

IBM DB2 Information Integrator



データ・ソース構成ガイド

バージョン 8

IBM DB2 Information Integrator



データ・ソース構成ガイド

バージョン 8

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、449 ページの『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

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 2003.6

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 - 2003. All rights reserved.

© Copyright IBM Japan 2003

目次

本書について	xi	チェックリスト: 連合オブジェクトのネーミング規則	26
本書の対象読者	xi	チェックリスト: 大文字小文字の区別をする値の保持	26
規則	xi	チェックリスト: データ・ソース統計	27
構文図の読み方	xi	チェックリスト: データ・タイプ・マッピング	27
第 1 章 データ・ソースへのアクセスの構成の概要	1	チェックリスト: ユーザー・マッピング	28
データ・ソースの構成の近道	1	チェックリスト: ラッパー	29
サポートされるデータ・ソース	3	第 4 章 DB2 ファミリー・データ・ソースへのアクセスの構成	31
データ・ソース・オブジェクトにニックネームを作成する	5	連合サーバーへの DB2 ファミリー・データ・ソースの追加	31
ニックネームの作成時に列オプションを組み込む	7	連合ノード・ディレクトリーへのノード項目のカタログ	32
ニックネームに対してニックネームを作成	7	連合システム・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	33
オプションの構成ステップ	8	DB2 ラッパーの登録	34
第 2 章 連合システムの概要	9	DB2 データ・ソースのサーバー定義の登録	35
ラッパーおよびラッパー・モジュール	9	CREATE SERVER ステートメント - DB2 ラッパーの例	37
サーバー定義およびサーバー・オプション	10	DB2 データ・ソースのユーザー・マッピングの作成	38
照合シーケンスとデータ・ソースの構成	11	CREATE USER MAPPING ステートメント - DB2 ラッパーの例	39
ユーザー・マッピング	11	DB2 データ・ソース・サーバーへの接続のテスト	40
ニックネームとデータ・ソース・オブジェクト	12	DB2 表およびビューのニックネームの登録	41
有効なデータ・ソース・オブジェクト	13	CREATE NICKNAME ステートメント - DB2 ラッパーの例	42
列オプション	14	DB2 ファミリー・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	43
データ・タイプ・マッピング	15	DB2_DJ_COMM 変数を設定してパフォーマンスを改善する (UNIX)	43
関数マッピング	16	第 5 章 Informix データ・ソースへのアクセスの構成	45
索引の指定	16	連合サーバーへの Informix データ・ソースの追加	45
第 3 章 連合データ・ソースの構成の計画	19		
連合オブジェクトのネーミング規則	19		
連合システムでの大文字小文字の区別をする値の保持	20		
データ・ソース統計の更新	22		
データ・タイプ・マッピングの計画	22		
関数マッピングの計画	24		
ユーザー・マッピングの計画	25		
正しいラッパーの選択	25		
連合システム構成の計画のためのチェックリスト	26		

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	46
Informix ラッパーの登録	48
Informix データ・ソースのサーバー定義の登録	49
CREATE SERVER ステートメント - Informix ラッパーの例	50
Informix データ・ソースのユーザー・マッピングの作成	52
CREATE USER MAPPING ステートメント - Informix ラッパーの例	53
Informix サーバーへの接続のテスト	54
Informix 表、ビュー、およびシノニムのニックネームの登録	55
CREATE NICKNAME ステートメント - Informix ラッパーの例	56
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	57
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	57
DB2_DJ_COMM 変数を設定してパフォーマンスを改善する (UNIX)	58

第 6 章 Oracle データ・ソースへのアクセスの構成	61
連合サーバーへの Oracle データ・ソースの追加	61
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	62
Oracle ラッパーの登録	63
Oracle ラッパーおよびライブラリー名	64
Oracle データ・ソースのサーバー定義の登録	66
CREATE SERVER ステートメント - Oracle ラッパーの例	67
Oracle データ・ソースのユーザー・マッピングの作成	68
CREATE USER MAPPING ステートメント - Oracle ラッパーの例	69
Oracle サーバーへの接続のテスト	70
Oracle 表およびビューのニックネームの登録	71
CREATE NICKNAME ステートメント - Oracle ラッパーの例	72
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	73

DB2_DJ_COMM 変数を設定してパフォーマンスを改善する (UNIX)	73
接続の問題	74

第 7 章 Sybase データ・ソースへのアクセスの構成	75
連合サーバーへの Sybase データ・ソースの追加	75
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	76
Sybase ラッパーの登録	78
Sybase データ・ソースのサーバー定義の登録	79
CREATE SERVER ステートメント - Sybase ラッパーの例	80
Sybase データ・ソースのユーザー・マッピングの作成	83
CREATE USER MAPPING ステートメント - Sybase ラッパーの例	83
Sybase サーバーへの接続のテスト	84
Sybase 表およびビューのニックネームの登録	86
CREATE NICKNAME ステートメント - Sybase ラッパーの例	87
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	87
DB2_DJ_COMM 環境変数を設定してパフォーマンスを改善する (UNIX)	88
DBLIB ではなく CTLIB を使用	88
sp_helpindex エラーを解決する	89

第 8 章 Microsoft SQL Server データ・ソースへのアクセスの構成	91
連合サーバーへの Microsoft SQL Server データ・ソースの追加	91
Microsoft SQL Server データ・ソースにアクセスするための連合サーバーとデータベースの準備	92
Microsoft SQL Server ラッパーの登録	94
Microsoft SQL Server データ・ソースのサーバー定義の登録	95
CREATE SERVER ステートメント - Microsoft SQL Server ラッパーの例	97
Microsoft SQL Server データ・ソースのユーザー・マッピングの作成	98
CREATE USER MAPPING ステートメント - Microsoft SQL Server ラッパーの例	99

Microsoft SQL Server リモート・サーバーへの接続のテスト	100
Microsoft SQL Server 表およびビューのニックネームの登録	101
CREATE NICKNAME ステートメント - Microsoft SQL Server ラッパーの例	103
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	103
DB2_DJ_COMM 変数を設定してパフォーマンスを改善する (UNIX)	104
ODBC トレースの入手	104
第 9 章 ODBC データ・ソースへのアクセスの構成	107
連合サーバーへの ODBC データ・ソースの追加	107
ODBC を通じてデータ・ソースにアクセスするための連合サーバーとデータベースの準備	109
ODBC ラッパーの登録	110
CREATE WRAPPER ステートメント - ODBC ラッパーの例	111
ODBC データ・ソースのサーバー定義の登録	112
CREATE SERVER ステートメント - ODBC ラッパーの例	113
ODBC データ・ソースのユーザー・マッピングの作成	114
CREATE USER MAPPING ステートメント - ODBC ラッパーの例	115
ODBC データ・ソース・サーバーへの接続のテスト	116
ODBC データ・ソース表およびビューのニックネームの登録	117
CREATE NICKNAME ステートメント - ODBC ラッパーの例	119
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	120
DB2_DJ_COMM 変数の設定によるパフォーマンスの改善	120
ODBC トレースの入手	121
第 10 章 Teradata データ・ソースへのアクセスの構成	123
連合サーバーへの Teradata データ・ソースの追加	123
Teradata サーバーへの接続のテスト	125

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)	126
Teradata ラッパーの登録	127
Teradata データ・ソースのサーバー定義の登録	128
CREATE SERVER ステートメント - Teradata ラッパーの例	130
Teradata データ・ソースのユーザー・マッピングの作成	131
CREATE USER MAPPING ステートメント - Teradata ラッパーの例	132
連合サーバーから Teradata サーバーへの接続のテスト	132
連合サーバーでの Teradata ニックネーム	134
Teradata 表およびビューのニックネームの登録	135
CREATE NICKNAME ステートメント - Teradata ラッパーの例	135
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	136
DB2_DJ_COMM 変数を設定して Teradata サーバーへのアクセスを改善する	136
Teradata アクセス・ロギングのチューニングと使用不可	137
libcliv2.so (AIX) のランタイム・リンクの使用可能化	137

第 11 章 OLE DB データ・ソースへのアクセスの構成	139
OLE DB データ・ソースを連合サーバーに追加	139
OLE DB ラッパーの登録	140
OLE DB データ・ソースのサーバー定義の登録	141
CREATE SERVER ステートメント - OLE DB ラッパーの例	142
OLE DB データ・ソースのユーザー・マッピングの作成	143
CREATE USER MAPPING ステートメント - OLE DB ラッパーの例	143

第 12 章 表構造ファイル・データ・ソースへのアクセスの構成	145
表構造ファイルとは?	145
表構造ファイルのタイプ	145
ソート・ファイル	146

ソートされていないファイル	146
DB2 Information Integrator が表構造ファイル を処理する方法	146
連合システムへの表構造ファイルの追加	148
表構造ファイル・ラッパーの登録	148
表構造ファイル・ラッパーの	
DB2_DJ_COMM DB2 プロファイル変数の設 定	149
表構造ファイルのサーバーの登録	150
表構造ファイルのニックネームの登録	151
CREATE NICKNAME ステートメント - 表 構造ファイル・ラッパーの例	152
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	153
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	153
表構造ファイル・ラッパーのメッセージ	154
第 13 章 Documentum データ・ソースへ のアクセスの構成	159
Documentum とは?	159
連合システムへの Documentum の追加	161
Documentum クライアント・ライブラリーの DB2 での使用可能化 (AIX および Solaris オ ペレーティング環境のみ)	162
Documentum のクライアント dmcl.ini ファイ ルの位置指定	163
Documentum ラッパーの登録	165
Documentum ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定	165
Documentum データ・ソースのサーバーの登 録	166
ユーザーのマッピング (Documentum ラッパ ー)	167
Documentum データ・ソースのニックネーム の登録	168
疑似列とは	168
CREATE NICKNAME ステートメント - Documentum ラッパーの例	172
Documentum データ・ソースのカスタム関数 の登録	174
カスタム関数のSTRING引き数の規則	175
照会でのカスタム関数の使用	175
カスタム関数表	176
Documentum データ・ソース - 照会の例	181

Documentum ラッパー用の CreateNicknameFile ユーティリティとは?	182
CreateNicknameFile ユーティリティのイン ストール (Documentum ラッパー)	183
CreateNicknameFile ユーティリティの構成 (Documentum ラッパー)	183
DM_ID オブジェクト・タイプの Documentum 登録済み表へのマッピング	184
繰り返し属性の二重の定義 (Documentum ラ ッパー)	185
Documentum ラッパーのアクセス・コントロ ール	186
Documentum ラッパーのメッセージ	186

第 14 章 Excel データ・ソースへのアクセ スの構成	195
Excel とは?	195
連合システムへの Excel の追加	197
Excel ラッパーの登録	197
Excel データ・ソースのサーバーの登録	198
Excel データ・ソースのニックネームの登録	199
Excel データ・ソース - 照会の例	199
Excel データ・ソース - サンプル・シナリオ	200
Excel ラッパーのファイル・アクセス・コン トロール・モデル	202
Excel ラッパーのメッセージ	203

第 15 章 BLAST データ・ソースへのアク セスの構成	211
BLAST とは?	211
連合システムへの BLAST の追加	216
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	217
BLAST デーモンの構成	217
BLAST デーモンの開始	220
BLAST ラッパーの登録	221
BLAST ラッパーの DB2_DJ_COMM DB2 プ ロファイル変数の設定	222
BLAST データ・ソースのサーバーの登録	223
BLAST データ・ソースのニックネームの登 録	223
定義行構文解析	224
固定列	224
CREATE NICKNAME ステートメント - BLAST ラッパーの例	230

BLAST ラッパーで作業するための TurboBlast のセットアップ	230
BLAST SQL 照会の作成	232
BLAST データ・ソース - 照会の例	232
BLAST ラッパーの最適化のヒント	234
BLAST ラッパーのメッセージ	234

第 16 章 XML データ・ソースへのアクセスの構成	239
XML とは?	239
連合システムへの XML の追加	243
XML ラッパーの登録	243
XML ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定	244
XML データ・ソースのサーバーの登録	245
ニックネームと XML 文書の間のデータ関連	246
XML ラッパーのコスト・モデル機能	248
XML コスト・モデル機能の最適化のヒント	249
XML データ・ソースのニックネームの登録	251
CREATE NICKNAME ステートメント - XML ラッパーの例	251
非ルート・ニックネームの連合ビューの作成 (XML ラッパー)	257
XML データ・ソース - 照会の例	258
XML ラッパーのメッセージ	259

第 17 章 Entrez データ・ソースへのアクセスの構成	271
Entrez とは?	271
Entrez の連合システムへの追加	272
Entrez ラッパーのカスタム関数の登録	273
Entrez ラッパーの登録	274
Entrez ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定	274
Entrez データ・ソースのサーバーの登録	275
Entrez データ・ソースのニックネームの登録	276
カスタム関数と Entrez 照会	278
Entrez ラッパーの関係述部	279
Entrez ラッパーで無効な WHERE 文節	280
スキーマ・データ・エレメントの単純化	280
項目リスト	280
名前	281
日付	281
Entrez データ・ソース - 照会の例	281
カスタム関数表 - Entrez ラッパー	283
PubMed および Nucleotide スキーマ表	283

Entrez ラッパーのメッセージ	291
-----------------------------	-----

第 18 章 Extended Search データ・ソースへのアクセスの構成	297
Extended Search とは?	297
Extended Search のデータ・ソース	298
Extended Search ラッパーがどのように機能するか	299
Extended Search のニックネーム	300
Extended Search の垂直表	302
連合サーバーへの Extended Search データ・ソースの追加	305
Extended Search ラッパーの登録	305
Extended Search データ・ソースのサーバーの登録	306
Extended Search データ・ソースのニックネームの登録	307
Extended Search データ・ソースのユーザー・マッピングの登録	308
Extended Search カスタム関数の登録	309
Extended Search ラッパー - 照会のガイドライン	310
Extended Search ラッパー - 照会の例	312
Extended Search ラッパー - 汎用照会言語	315
Extended Search ラッパーのメッセージ	317

第 19 章 HMMER データ・ソースへのアクセスの構成	323
HMMER とは?	323
HMMER の連合システムへの追加	327
正しいバージョンの hmmpfam 実行可能プログラムがインストールされていることの検査	328
HMMER デーモンの構成	328
HMMER デーモンの開始	331
HMMER ラッパーの登録	332
HMMER ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定	333
HMMER データ・ソースのサーバーの登録	334
引き数	334
オプション	334
HMMER データ・ソースのニックネームの登録	335
ニックネーム・オプション	336
固定列	336
CREATE NICKNAME ステートメント - HMMER ラッパーの例	339

HMMER データ・ソース - 完全指定の例 サンプルを使用した新規 HMMER 照会の構 成	339 341
HMMER ラッパーの最適化のヒント	342
HMMER ラッパーのメッセージ	342
第 20 章 ニックネームの変更	345
ニックネームの変更	345
データ・タイプの変更	345
ニックネーム・オプションの変更	346
第 21 章 DDL コマンド・リファレンス	347
CREATE FUNCTION ステートメント構文 - Extended Search ラッパー	347
CREATE NICKNAME ステートメント構文 - BLAST ラッパー	348
CREATE NICKNAME ステートメント構文 - Documentum ラッパー	350
CREATE NICKNAME ステートメント構文 - Excel ラッパー	353
CREATE NICKNAME ステートメント構文 - Extended Search ラッパー	355
CREATE NICKNAME ステートメントのオプ ション - Entrez ラッパー	358
CREATE NICKNAME ステートメント構文 - 表構造ファイル・ラッパー	358
CREATE NICKNAME ステートメント構文 - XML ラッパー	363
CREATE SERVER ステートメントの引き数 - BLAST ラッパー	368
CREATE SERVER ステートメントの引き数 とオプション - Documentum ラッパー	369
CREATE SERVER ステートメントの引き数 - Entrez ラッパー	370
CREATE SERVER ステートメントの引き数 - Excel ラッパー	370
CREATE SERVER ステートメント構文 - Extended Search ラッパー	371
CREATE USER MAPPING ステートメントの オプション - Documentum ラッパー	372
CREATE USER MAPPING ステートメント構 文 - Extended Search ラッパー	373
CREATE WRAPPER ステートメント構文 - Extended Search ラッパー	373

付録 A. 連合情報を含むグローバル・カタロ グ表内のビュー	375
---	-----

付録 B. 連合システムのサーバー・オプショ ン	379
---------------------------------------	-----

付録 C. 連合システムのユーザー・マッピン グ・オプション	391
---	-----

付録 D. 連合システムの列オプション	393
-------------------------------	-----

付録 E. 連合システムの関数マッピング・オ プション	395
--	-----

付録 F. SQL ステートメントで有効なサー バーのタイプ	397
CTLIB ラッパー	397
DBLIB ラッパー	397
DJXMSSQL3 ラッパー	397
DRDA ラッパー	398
Informix ラッパー	399
MSSQLODBC3 ラッパー	399
NET8 ラッパー	399
ODBC ラッパー	399
OLE DB ラッパー	400
SQLNET ラッパー	400
Teradata ラッパー	400

付録 G. デフォルトの順方向データ・タイ プ・マッピング	401
DB2 for z/OS and OS/390 データ・ソース	402
DB2 for iSeries データ・ソース	403
DB2 Server for VM and VSE データ・ソー ス	405
DB2 for Linux、DB2 for UNIX、および DB2 for Windows データ・ソース	406
Informix データ・ソース	408
Microsoft SQL Server データ・ソース	410
ODBC データ・ソース	413
Oracle NET8 データ・ソース	415
Oracle SQLNET データ・ソース	416
Sybase データ・ソース	417
Teradata データ・ソース	419

付録 H. デフォルトの逆方向データ・タイ プ・マッピング	421
DB2 for z/OS and OS/390 データ・ソース	422

DB2 for iSeries データ・ソース	423	DB2 Information Integrator インフォメーション・センターのトピックを見つける	439
DB2 for VM and VSE データ・ソース	425	DB2 HTML ドキュメンテーションの使用	441
DB2 for Linux、DB2 for UNIX、および DB2 for Windows データ・ソース	426	DB2 文書の検索	444
Informix データ・ソース	427	Netscape 4.x を使って DB2 資料を検索する場合のトラブルシューティング	445
Microsoft SQL Server データ・ソース	429	アクセシビリティ 447	
Oracle SQLNET データ・ソース	430	キーボードによる入力およびナビゲーション	447
Oracle NET8 データ・ソース	431	アクセスしやすい表示	447
Sybase データ・ソース	432	フォントの設定	447
Teradata データ・ソース	433	色覚への非依存	447
DB2 Information Integrator の技術資料 435		代替アラート・キュー	448
資料およびリリース情報へのアクセス	435	支援テクノロジーとの互換性	448
DB2 Information Integrator の資料	435	入手可能な資料	448
リリース情報およびインストール要件	438	特記事項 449	
DB2 Information Integrator ドキュメンテーション・フィックスパック	438	商標	451
DB2 Information Integrator インフォメーション・センターまたは DB2 HTML ドキュメンテーション CD を使ったトピックへのアクセス	439	索引 453	
DB2 Information Integrator インフォメーション・センターの機能	439	IBM と連絡を取る 461	
		製品情報	461

本書について

本書の内容は、次のとおりです。

- ラッパーを登録して連合システムにデータ・ソースを加える方法。ラッパーとは、ユーザーやアプリケーションが SQL ステートメントを使用してデータ・ソースと通信することを可能にするモジュールのことです。

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、連合データベース環境構築の任に当たる管理者、およびそのような環境においてアプリケーションを開発するアプリケーション・プログラマーを対象としています。

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 (GUI) コントロール (たとえば、フィールド名、フォルダー名、メニュー選択) を示します。

モノスペース体

コーディングやユーザーが入力するテキストの例を示します。

イタリック体

値をもって置きかえる変数を示します。イタリック体は、また資料の表題や強調語も示します。

英大文字体

SQL キーワードやオブジェクト名 (たとえば、表、ビュー、およびサーバー) を示します。

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。



オプション項目は、メインパスの下に示されます。

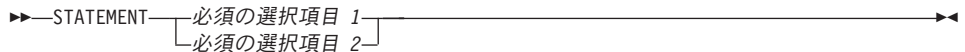


オプション項目がメインパスの上に示されている場合、その項目は読みやすくするために示されているのであり、ステートメントの実行には影響を与えません。



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。

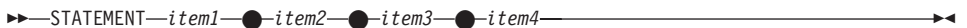
単一の変数が、いくつかのパラメーターのセットを表していることがあります。たとえば、下記の図で、変数 `parameter-block` は、**parameter-block** という見だしの下の内容で置きかえられます。



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 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、連合サーバーおよび連合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。アクセスを構成するステップは、どのデータ・ソースでも似ています。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 1. 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・ソース用の連合サーバーを準備する	クライアント構成アシスタント	DB2 ファミリー・データ・ソースの場合: ノードおよびリモート・データベースのカatalog Informix、Oracle、Sybase、Microsoft [®] SQL Server データ・ソースの場合: 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
2. ラッパーを作成する	DB2 コントロール・センター	

表 1. 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
3. サーバー定義を作成する	DB2 コントロール・センター	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せん。DB2 では、ノードは DB2 の実行コピーであるインスタンスと同じです。
4. ユーザー・マッピングを作成する	DB2 コントロール・センター	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する	DB2 コマンド・センター	DB2 コントロール・センターの「すべての表の表示 (Show All Tables)」パネルを使用して、接続を検査
6. ニックネームを作成する	DB2 コントロール・センター	

ただし、データ・ソースへのアクセスを構成するには、その前に連合サーバーが適切にセットアップされていることを確認する必要があります。特に、次の点が重要です。

- DB2 をクライアント・ソフトウェアにリンクする。これにより、連合サーバー上にデータ・ソース・ラッパー・ライブラリーが作成されます。
- データ・ソース環境変数をセットアップする。

関連概念:

- 8 ページの『オプションの構成ステップ』

関連タスク:

- 「*DB2 Information Integrator* インストール・ガイド」の『連合サーバーのセットアップをチェックする』

関連資料:

- 19 ページの『連合オブジェクトのネーミング規則』

サポートされるデータ・ソース

連合システムを使用すると、数多くのデータ・ソースにアクセスできます。サポートされるデータ・ソースは、次の表に示すとおりです。

表 2. 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、DB2 for UNIX、および DB2 for Windows®	7.1、7.2、8.1	DRDA®
DB2 Universal Database (z/OS™ および OS/390® 版)	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	DRDA
DB2 Universal Database (iSeries™ 版)	4.5 (またはそれ以降) で以下の APAR が適用 されているもの: <ul style="list-style-type: none">• SA95719• SE06003• SE06872• SI05990• SI05991	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
Informix™	7、8、9	Informix Client SDK
ODBC	3.x	データ・ソース用の ODBC ドライバー (Redbrick にア クセスするための Redbrick ODBC Driver など)。

表 2. 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
OLE DB		OLE DB 2.0 以上
Oracle	7.3.4、8.x、9.x	SQLNET または NET8 クライアント・ソフトウェア
Microsoft SQL Server	6.5、7.0、2000	Windows では、Microsoft SQL Server Client ODBC 3.0 以降のドライバー。 UNIX では、DataDirect Technologies (旧 MERANT) Connect ODBC 3.7 以降のドライバー。
Sybase	11.x、12.x	Sybase Open Client
Teradata	V2R3、V2R4	Teradata Call-Level Interface バージョン 2 (CLiV2) リリース 04.06 (またはそれ以降)
BLAST	2.x	BLAST デーモン (ラッパーと一緒に提供される)
Documentum	Documentum サーバー: EDMS 98 (バージョン 3 とも呼ばれる) および 4i。	Documentum Client API/Library
Entrez	1.0	なし
HMMER	2.2g	HMMER デーモン (ラッパーと一緒に提供される)
IBM Lotus Extended Search	4.0	Extended Search Client Library (ラッパーと一緒に提供される)
Microsoft Excel	97、2000	連合サーバーにインストールされた Excel 97 または 2000
表構造ファイル		なし
XML	1.0 仕様	なし

関連概念:

- 「連合システム・ガイド」の『データ・ソースとは?』

データ・ソース・オブジェクトにニックネームを作成する

ニックネームの作成作業は、一般に構成作業で最も複雑な作業です。ここでは、ニックネームの候補となるものを識別し、連合データ・ソース・オブジェクトのニックネームを登録する作業例を示します。

データ・ソース・オブジェクトには、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。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 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 サーチ・アルゴリズム、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。

連合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作成しません。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。

リモート・オブジェクト とは、次のものです。

- 連合サーバー上の別の DB2® データベース・インスタンス内の表およびビュー。これらのオブジェクトにはニックネームを作成する必要があります。
- 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作成する必要があります。

連合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 (対応付け) 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、連合サーバーがこれらのオブジェクトを参照するためのポインターです。

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連合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。

リレーショナル・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、リモート・サーバーからカタログ・データが検索され、連合グローバル・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・ソース情報のグローバル・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、索引を持つ表にニックネームを定義したとします。グローバル・カタログに提供されるメタデータは、索引に関する情報 (索引キー内の列の名前など) です。

ニックネームを作成するには、DB2 コントロール・センターを使用します。DB2 コマンド・センター、またはコマンド行プロセッサ (CLP) から `CREATE NICKNAME` ステートメントを実行することもできます。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。

`CREATE NICKNAME` ステートメントの例を以下に示します。

```
CREATE NICKNAME SYBSALES FOR SYBSERVER."salesdata"."europe"
```

ここで、

SYBSALES

これは、Sybase の表またはビューのユニークなニックネームです。

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許可 ID になります。ニックネームの長さは 128 文字までです。

SYBSERVER."salesdata"."europe"

これは、リモート・データ・ソース・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` ステートメントでデータ・ソース・サーバーに与えた名前です。
- **salesdata** は、オブジェクトが属するリモート・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・オブジェクトの名前です。この値は大文字小文字の区別があります。

ニックネームの作成時に、連合サーバー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。このため、サーバーはデータ・ソース・カタログの照会を試みません。接続が働かない場合、エラー・メッセージが出ます。

ニックネームの作成時に列オプションを組み込む

INDONESIA_SALES という表に INDSALES というニックネームを作成したいとします。この表にはデータ・タイプが CHAR の POSTAL_CODE という列があります。この列には数字のみが入ります。データ・ソースの照合シーケンスは、連合データベースの照合シーケンスとは異なります。通常、連合サーバーはこの列をデータ・ソースでソートしません。しかし、POSTAL_CODE 列には数字 ('0', '1', ..., '9') のみが入ります。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、データを連合サーバーに持ってくるというオーバーヘッドが避けられます。この情報を連合サーバーに提供するには、NUMERIC_STRING 列オプションを CREATE NICKNAME ステートメントに追加します。例:

```
CREATE NICKNAME INDSALES FOR SERVER44."sales"."INDONESIA_SALES"  
OPTIONS (POSTAL_CODE NUMERIC_STRING 'Y')
```

非リレーショナルのデータ・ソースの中には、ラッパーがデフォルトのタイプ・マッピングを含んでいないものもあります。ラッパーがデフォルトのタイプ・マッピングを含まない場合は、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 UNIX[®]、および DB2 for 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。例:

```
CREATE NICKNAME DRUGDATA1  
(DCODE INTEGER,DRUG CHAR(20),MANUFACTURER CHAR(20))  
FOR SERVER biochem_lab  
OPTIONS (FILE_PATH '/usr/pat/DRUGDATA1.TXT',  
COLUMN_DELIMITER ',', KEY_COLUMN 'Dcode', VALIDATE_DATA_FILE 'Y')
```

ニックネームに対してニックネームを作成

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。たとえば、AIX[®] を使用する連合サーバーと Windows を使用する連合サーバーがあり、この両方の連合サーバーから 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 を使用する連合サーバー上でしかサポートされません。AIX 連合サーバーから 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 連合サーバー上で、Excel データ・ソースにアクセスするようにサーバーをセットアップし、構成します。
2. Excel スプレッドシートにニックネームを作成します。
3. AIX 連合サーバー上で、DB2 ファミリー・データ・ソースにアクセスするようにサーバーをセットアップし、構成します。
4. Windows 連合サーバー上の Excel ニックネームにニックネームを作成します。

関連タスク:

- 『ニックネーム』
- 『ニックネームの作成の表とビューにフィルターをかける』
- 『ニックネーム作成のために表をフィルターに掛ける』
- 『ニックネームを作成する』

関連資料:

- 「*SQL* リファレンス 第 2 巻」の『CREATE NICKNAME ステートメント』
- 19 ページの『連合オブジェクトのネーミング規則』

オプションの構成ステップ

連合照会の中で使用するニックネームの列は、次のようにしてカスタマイズまたは調整できます。

- データ・ソースへのアクセスを当初に構成した時には索引を持たなかったオブジェクトに、索引を指定する。たとえば、表に新しい索引ができた時に、索引の指定を作成します。同様に、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することもあります。
- 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・マッピングを定義する。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。
- 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する。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。

第 2 章 連合システムの概要

以下の各セクションでは、連合システムについての概要を示します。

ラッパーおよびラッパー・モジュール

ラッパーとは、連合サーバーがデータ・ソースとやりとりするためのメカニズムです。連合サーバーは、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 と呼ばれる）を使用してラッパーをインプリメントします。これらのルーチンは、データ・ソースに接続し、そこからデータを繰り返し検索するための接続といった操作を、連合サーバーが実行できるようにします。通常、DB2® 連合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、ラッパーを連合データベースに登録します。

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。作成する必要があるのは 2 つのラッパーだけであり、その 1 つは DB2 データ・ソース・オブジェクト用、もう 1 つは Informix データ・ソース・オブジェクト用です。連合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、DB2 for UNIX®, および DB2 for 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。

サーバー定義とニックネームを使用して、各データ・ソース・オブジェクトを特定して識別（名前、ロケーションなど）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、ソースにユニークの照会言語に、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連合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、データ・タイプ・マッピングを作成すると、それによってデフォルトのデータ・タイプ・マッピングがオーバーライドされます。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連合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、連合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、関数マッピングを作成すると、それによってデフォルトの関数マッピングがオーバーライドされます。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、ラッパーを構成するため、または DB2 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。

関連概念:

- 10 ページの『サーバー定義およびサーバー・オプション』

関連資料:

- 「連合システム・ガイド」の『デフォルト・ラッパー名』

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、連合インスタンスの所有者はデータ・ソースを連合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。データ・ソースが RDBMS の場合、次のような情報が含まれます。

- RDBMS のタイプおよびバージョン
- RDBMS 上のデータ・ソースのデータベース名
- RDBMS に特有なメタデータ

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。定義には、どのデータベースに連合サーバーが接続できるかを指定する必要があります。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、連合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。そこで、Oracle データ・ソースの場合は、連合サーバー定義にデータベース名は含まれません。

インスタンス所有者が連合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・ソースはデータに対する要求に回答し、それ自体がサーバーです。

サーバー定義は、CREATE SERVER および ALTER SERVER ステートメントを使用して作成および変更します。

サーバー定義内の情報のあるものは、サーバー・オプションとして保管されます。サーバー定義の作成時に、サーバーに関して指定できるオプションを理解することが重要です。いくつかのサーバー・オプションはラッパーを構成し、あるオプションは DB2 がラッパーをどのように使用するかに影響します。

サーバー・オプションは、データ・ソースとの複数回の接続を通じて持続するように設定したり、1回の接続の間に関して設定したりできます。

関連概念:

- 11 ページの『ユーザー・マッピング』

関連資料:

- 379 ページの『付録 B. 連合システムのサーバー・オプション』

照合シーケンスとデータ・ソースの構成

連合データベースは、DB2 Information Integrator のインストール作業の一部として作成されたものです。その時点で、データ・ソースの照合シーケンスと一致する照合シーケンスが指定されました。サーバー定義を連合データベースに登録する時点で、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。この設定により、連合データベースに対し、連合データベースとデータ・ソース・サーバーの照合シーケンスが一致していることが通知されます。

ユーザー・マッピング

連合サーバーがデータ・ソースに要求をプッシュダウンする必要がある場合、サーバーは最初にデータ・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・ソースに関して、連合サーバーは、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・ソースとの接続にユーザー ID とパスワードが必要な場合は、連合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連合システムを使用するユーザー ID には、この関連付けを必ず作成しておかなければなりません。この関連付けは、ユーザー・マッピングと呼ばれます。

関連概念:

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、連合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、アクセスしたいデータ・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は、データ・ソース・オブジェクトと呼ばれます。

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、連合サーバーがこれらのオブジェクトを参照するためのポインターです。ニックネームは通常、CREATE NICKNAME ステートメントを使用して定義します。

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連合サーバーにサブミットする場合、その要求でデータ・ソースを指定する必要はありません。その代わりに、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にマップされます。それらのマッピング (対応付け) 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はありません。

たとえば、*NFX1.PERSON* という Informix™ データベース表を表すニックネーム *DEPT* を定義すると、`SELECT * FROM DEPT` というステートメントを連合サーバーから使用できます。しかし、連合サーバーから `SELECT * FROM NFX1.PERSON` というステートメントは使用できません (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、このメタデータおよび、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・タイプ間のマッピングが入っています。

現在のところ、ニックネームに対して DB2 のユーティリティー操作の一部 (REORG、REORGCHK、IMPORT、RUNSTATS など) を実行することはできません。

関連概念:

- 14 ページの『列オプション』

関連資料:

- 13 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

ニックネームは、アクセスするデータ・ソースにおいてオブジェクトを識別します。次の表は、連合システムの中でニックネームを作成できるオブジェクトの種類のリストです。

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、DB2 for UNIX、および DB2 for 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
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	Lotus Notes のデータベース、Microsoft Access、Microsoft Index Server、Web 検索エンジン、および LDAP ディレクトリ。
HMMER	HMMER の hmmpfam 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、階層 Markov モデルのライブラリー)。
Microsoft Excel	.xls ファイル (ワークブック内の最初のシートのみがアクセスされる)
表構造ファイル	.txt ファイル (非常に特定のフォーマットを持つテキスト・ファイル)
XML タグ・ファイル	XML 文書内の項目のセット

関連概念:

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

列オプション

ニックネームが付けられたオブジェクトについての追加のメタデータ情報を、グローバル・カタログに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ます。このメタデータは、列オプションと呼ばれるパラメーターに割り当てます。列オプションは、この列内のデータに通常の列とは異なる扱いをするように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。

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、列オプションは、ラッパーが XML 文書から列を解析する時に使用する XPath 式をラッパーに指示するために使用されます。

連合により 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、あたかもそれがローカル DB2 表であるかのように処理します。したがって、ニックネームを作成する任意のデータ・ソース・オブジェクトに対して、列オプションをセットすることができます。列オプションのいくつかは特定のタイプのデータ・ソース用に作られたものなので、それらのデータ・ソースにのみ適用できます。

仮に、データ・ソースが、連合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とします。連合サーバーは通常、データ・ソース側で文字データを含む列をソートすることはありません。データは連合データベースに戻され、ローカルで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') のみ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、データを連合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。

重要: NUMERIC_STRING 列オプションは、リレーショナル・データ・ソースに場合のみ有効です。

列オプションは CREATE NICKNAME および ALTER NICKNAME ステートメントで定義できます。

関連概念:

- 15 ページの『データ・タイプ・マッピング』

関連タスク:

- 「*連合システム・ガイド*」の『ニックネームを使用した操作』

関連資料:

- 393 ページの『付録 D. 連合システムの列オプション』

データ・タイプ・マッピング

連合サーバーがデータ・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・タイプと対応付けられて (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・マッピングの例を以下に示します。

- 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™ のタイプ DATE は、DB2 のタイプ 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® ラッパー内にあります。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、他も同様です。

非リレーショナルのデータ・ソースの場合、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、DB2 for UNIX®, および DB2 for Windows® のデータ・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・データの場合には、デフォルト・データ・タイプ・マッピングをオーバーライドしたり、デフォルト値がない場合にはマッピングを変更することがあります。たとえば、データ・ソース側で新しい組み込みタイプが使用可能になった場合、またはデータ・ソース側にマップしたいユーザー定義タイプがある場合に、タイプ・マッピングを作成できます。

重要: 新しいタイプ・マッピングを作成したり、デフォルトのタイプ・マッピングを変更したりすることは、ニックネームを作成する前に実行してください。タイプ・マッピングの変更より前にさされたニックネームには、新しいマッピングが反映されません。

関連概念:

- 「*連合システム・ガイド*」の『連合システムでのデータ・タイプ・マッピング』

関数マッピング

連合サーバーがデータ・ソース関数を認識するには、その関数が、DB2[®] for Linux、DB2 for UNIX[®]、および DB2 for Windows での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 では、既存の組み込みデータ・ソース関数と、それに対応する組み込み DB2 関数の間のデフォルトのマッピングが用意されています。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[®] 関数へのデフォルトの関数マッピングは、DRDA[®] ラッパーに含まれています。また、Sybase 関数へのデフォルトの関数マッピングは、CTLIB および DBLIB ラッパーに含まれている、というようになっています。

リレーショナル・データ・ソースの場合、連合サーバーに認識されていないデータ・ソース関数を使用する際に、関数マッピングを作成できます。そのマッピングは、データ・ソース関数と、連合データベースでの対応する DB2 関数との間に作成します。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。

関連概念:

- 「連合システム・ガイド」の『連合システムでの関数マッピング』
- 16 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、分散要求の処理を速くします。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。連合サーバーは、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、リモート・カタログに保管される索引情報はない)
- 連合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

あるいは、ある表に、ニックネームの作成時にあった索引に加えて、新しい索引が作られたとします。索引情報は、ニックネームが作成された時にグローバル・カタログに入るため、連合サーバーは新しい索引については認識しません。同様に、ビューにニックネームを作成すると、ビューの基になる表 (およびその索引) については、連合サーバーは認識しません。このような場合に、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。索引を持たない表に「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サーチするべきかを照会
オブティマイザーに示します。

関連概念:

- 「*連合システム・ガイド*」の『*連合システムでの索引の指定*』

第 3 章 連合データ・ソースの構成の計画

以下の各セクションでは、連合システムの計画作業に役立つ情報を示します。

連合オブジェクトのネーミング規則

他の DB2 オブジェクトと同様に、連合データベース・オブジェクトのネーミングにも規則があります。

連合データベース・オブジェクトには、以下のようなものが含まれます。

- 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

連合オブジェクト名は、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みます。(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)

連合オブジェクト名は、数字または下線文字で始めることはできません。

連合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)

連合オブジェクト名に、128 バイトを超える名前を使用す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、英大文字に変換されます。

関連概念:

- 「管理ガイド: プランニング」の『NLS 環境での命名規則』

- 「管理ガイド: プランニング」の『Unicode 環境での命名規則』

関連資料:

- 20 ページの『連合システムでの大文字小文字の区別をする値の保持』

連合システムでの大文字小文字の区別をする値の保持

データ・ソースで大文字小文字の区別がされるユーザー ID やパスワードなど値を連合システムで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 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。

サーバー・オプションとその有効な設定値についての情報は、別のトピックとして説明があります。

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:

連合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。

- たとえば、SQL ステートメントが二重引用符を含み、単一引用符を含まないとしします。その場合ステートメントを単一引用符で囲みます。たとえば、次の SQL ステートメントを挿入するとします。

```
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'
```

- たとえば、SQL ステートメントが単一引用符を含み、二重引用符を含まないとしします。その場合ステートメントを二重引用符で囲みます。たとえば、次の SQL ステートメントを挿入するとします。

```
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') "
```

- たとえば、SQL ステートメントが、単一引用符と二重引用符の両方を含んでいるとします。その場合、ステートメントを二重引用符で囲み、さらにステートメント中の二重引用符の前にバックスラッシュを入れます。たとえば、次の SQL ステートメントを挿入するとします。

```
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')"
```

注: 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、二重引用符の前にバックスラッシュを置くことはできません。

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 では、各引用符の前に円記号を置きます。たとえば、Microsoft SQL サーバー表用に NICK1 というニックネームを作りたいとします。表は、NORBASE データベースにあります。スキーマは、my_schema で表は、weekly_salary です。

連合サーバーの 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) から値を入力する場合、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、単一引用符もバックスラッシュも必要ありません。上記の例の場合、DB2 コマンド・プロンプトで次のように入力します。

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

関連資料:

- 379 ページの『付録 B. 連合システムのサーバー・オプション』

データ・ソース統計の更新

連合データベースは、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てデータ・ソース・オブジェクトにニックネームを作成した時に収集されます。連合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。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atalogから読み込まれ、連合サーバーのグローバル・カatalogに置かれます。照会オプティマイザーは、データ・ソースのカatalog情報の一部またはすべてを使用するので、ニックネームを作成する前にデータ・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。

オブジェクトのニックネームを作成する際に、連合データベースはそのオブジェクトのリモート統計情報を一度限り検索します。ニックネームの作成後に、リモート・ソースがリモート・オブジェクトのカatalog統計を更新した場合、変更された統計情報は連合サーバーのグローバル・カatalogには伝搬されません。連合サーバー上のグローバル・カatalogがリモート・オブジェクトの現行の統計を反映するためには、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。

アクション: 連合サーバーに含めたいデータ・ソース側のオブジェクトを決定します。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。これらのデータ・ソースのうちいずれの統計を更新できるかを判断し、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。

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、連合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・ソースのデータ・タイプと連合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。逆方向タイプ・マッピングは、透過的 DDL で使用されません。逆方向タイプ・マッピングでは、マッピングはローカル・タイプから対応するリモート・タイプへのマッピングです。この 2 つのデータ・タイプ・マッピングに関するこのほかの情報については、別のトピックで説明します。

DB2 for UNIX、および DB2 for Windows はデータ・タイプ・マッピングを使用して、データ・ソース・オブジェクトの列に DB2 でサポートされるどのデータ・タイプを定義するべきかを判別します。デフォルトのデータ・タイプ・マッピングは、データ・ソース・ラッパーに含まれています。

しかし、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。以下の場合に、デフォルトのマッピングをオーバーライドできます。

- 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、連合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、連合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・ソース表が別個のデータ・タイプまたはユーザー定義のデータ・タイプを含む場合、次の 2 つの選択があります。

- 連合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、連合サーバーはどのデータ・タイプをこれらの列にマップすべきかを知ることができます。これらの別個のまたはユーザー定義のデータ・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のデータ・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

列が関連付けられるか、基になる組み込みデータ・タイプにキャストするデータ・ソースのビューを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。プランニング・チェックリストに、データ・タイプ・マッピング表に作成したいデータ・ソースとタイプ・マッピングをリストします。

関連概念:

- 15 ページの『データ・タイプ・マッピング』
- 「連合システム・ガイド」の『照会処理のチューニング』

関連資料:

- 401 ページの『付録 G. デフォルトの順方向データ・タイプ・マッピング』
- 421 ページの『付録 H. デフォルトの逆方向データ・タイプ・マッピング』

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。

連合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、連合データベースにある対応する関数との間に作成します。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連合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・ソースの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:

- 16 ページの『関数マッピング』

ユーザー・マッピングの計画

連合サーバーがデータ・ソースに要求をプッシュダウンする必要がある場合、サーバーは最初にデータ・ソースに接続を確立する必要があります。サーバーは、有効なユーザー ID とパスワードを使用して、そのデータ・ソースに接続を行います。連合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連合システムを使用するユーザー ID には、この関連付けを必ず作成しておかなければなりません。この関連付けは、ユーザー・マッピングと呼ばれます。

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: 連合サーバーとデータ・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・マッピング表に連合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。

正しいラッパーの選択

データ・ソースの中には複数のラッパーを使用できるものがあります。どれを選択するかは、使用するデータ・ソースのクライアント・ソフトウェアのバージョンにより異なります。あるいは、連合サーバーで使用するオペレーティング・システムによる場合もあります。

たとえば、Oracle データ・ソースでは、SQLNET ラッパーと NET8 ラッパーという 2 つのラッパーを使用できます。たとえば、Oracle バージョン 8 を使用し、連合サーバーのオペレーティング・システムは Windows NT であるとしてみます。当初は、SQLNET ラッパーを作成しました。あとになって、SQLNET ラッパーは LOB データ・タイプをサポートせず、NET8 ラッパーが LOB をサポートすることがわかりました。NET8 ラッパーでの LOB サポートを活用するには、SQLNET ラッパーをドロップして NET8 ラッパーを作成する必要があります。

注: NET8 ラッパーは、SQLNET ラッパーよりも新しいバージョンの Oracle クライアントが必要になります。

ラッパーをドロップすると、連鎖的に重大な影響が生じます。以下のように、連合システムの他のオブジェクトが影響を受けます。

- ドロップされたサーバー定義に依存するすべてのサーバー定義、ユーザー定義関数マッピング、ユーザー定義データ・タイプ・マッピングもドロップされます。

- ドロップされたサーバー定義に依存する、すべてのユーザー定義関数マッピング、ニックネーム、ユーザー定義データ・タイプ・マッピング、およびユーザー・マッピングもドロップされます。
- ドロップされたニックネームに依存する索引の指定はすべてドロップされます。
- それらのニックネームに依存する連合ビューはすべて、「作動不能」とマークされます。
- ドロップされたオブジェクトおよび「作動不能」ビューに依存する、すべてのアプリケーションは無効にされます。

DB2 Relational Connect には、Oracle、Microsoft SQL Server、および Sybase データ・ソース用に複数のラッパーが用意されています。これらのラッパーの違いについては、データ・ソース別の構成トピックに記述されています。

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連合システム用に作成するラッパーを記入します。

関連概念:

- 9 ページの『ラッパーおよびラッパー・モジュール』

関連タスク:

- 「連合システム・ガイド」の『Microsoft SQL Server データ・ソースを連合サーバーに追加する』
- 「連合システム・ガイド」の『Oracle データ・ソースを連合サーバーに追加』
- 「連合システム・ガイド」の『Sybase データ・ソースを連合サーバーに追加する』

連合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、連合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、連合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: 連合オブジェクトのネーミング規則

連合オブジェクトのネーミング規則を知っていますか。

連合オブジェクトのネーミング規則についての情報は、このセクション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: 連合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	サーバー名 (サーバー定義の中の)	FOLD_ID サーバー・オプションの設定	FOLD_PW サーバー・オプションの設定

チェックリスト: データ・ソース統計

次の表に連合システムの一部となるデータ・ソースをリストします。どのデータ・ソースの統計を更新するかを示しておきます。

表5. プランニング・チェックリスト: 連合システム用に更新するデータ・ソース統計

データ・ソース	カタログ情報を保守? (Y/N)	このデータ・ソースの統計を更新? (Y/N)	統計を更新するために使用するデータ・ソース・ユーティリティーの名前
DB2 for UNIX および DB2 for 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のあるデータ・ソースのデータ・タイプと対応する連合サーバーのデータ・タイプを示します。

表6. プランニング・チェックリスト: 連合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	サーバー名 (サーバー定義の中の)	データ・ソース・データ・タイプ	DB2 for UNIX、および DB2 for Windows データ・タイプ

表6. プランニング・チェックリスト: 連合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	サーバー名 (サーバー定義 の中の)	データ・ソース・ データ・タイプ	DB2 for UNIX、および DB2 for Windows 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、連合サーバーのユーザー ID と連合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: 連合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 UNIX、お よび DB2 for Windows 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: 連合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	作成するラッパー
DB2 Universal Database™ DB2 for UNIX、 および DB2 for Windows® DB2 Universal Database for z/OS and OS/390® DB2 Universal Database for iSeries DB2 Server for VM and VSE	DRDA	
Informix	INFORMIX	
Oracle	SQLNet Net8	
Microsoft® SQL Server	DJXMSSQL3 MSSQLODBC3	
ODBC	なし	
OLE DB	OLEDB	
Sybase	CTLIB DBLIB	
BLAST	なし	
Documentum	なし	
Microsoft Excel	なし	
表構造ファイル	なし	
XML	なし	

第 4 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連合サーバーを構成する方法について説明します。使用できるデータベースとしては、DB2 for UNIX、および DB2 for Windows、DB2 for z/OS and OS/390、DB2 for iSeries、および DB2 Server for VM and VSE があります。

この章では、DB2 ファミリー・データ・ソースに合わせて構成をセットアップする際に実行する必要がある作業のリスト、発行する必要がある SQL ステートメントの例、および使用できる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。

連合サーバーへの DB2 ファミリー・データ・ソースの追加

連合サーバーが DB2 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 DB2 データ・ソースとオブジェクトの情報をサーバーに提供します。

DB2 コントロール・センター、DB2 コマンド・センター、または DB2 コマンド行プロセッサを使用して、DB2 データ・ソースへのアクセスを構成することができます。

DB2 コントロール・センターを使用する場合の利点は、各ステートメントやコマンドを入力する必要がないことです。DB2 コントロール・センターでは、DB2 データ・ソースへのアクセスを迅速に構成する最も容易な方法が提供されています。ただし、以下の構成タスクを開始するために DB2 コントロール・センターを使用することはできません。

- ノードをカタログする。
- リモート・データベースをカタログする。
- サーバー定義とユーザー・マッピングを妥当性検査するため、データ・ソース・サーバーへの接続をテストする。
- 列オプションを追加またはドロップする。

前提条件:

- DB2 コマンド・センターまたは DB2 コマンド行プロセッサへのアクセス。
- DB2 ファミリー・データ・ソースにアクセスするようにセットアップされている連合サーバーおよびデータベース。

制約事項:

DB2 for UNIX および DB2 for 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:

連合サーバーに DB2 データ・ソースを追加するには、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て、サーバー・オプションをセットします。
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. 表およびビューのニックネームを登録します。

関連概念:

- 1 ページの『データ・ソースの構成の近道』

関連タスク:

- 「*DB2 Information Integrator インストール・ガイド*」の『連合サーバーのセットアップをチェックする』
- 32 ページの『連合ノード・ディレクトリーへのノード項目のカタログ』
- 43 ページの『DB2 ファミリー・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』

連合ノード・ディレクトリーへのノード項目のカタログ

連合ノード・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・ソースを連合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、連合サーバーのノード・ディレクトリーに 1 つの項目をカタログします。連合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:

連合ノード・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実行します。
 - 通信プロトコルが Transmission Control Protocol/Internet Protocol (TCP/IP) の場合、**CATALOG TCPIP NODE** コマンドを実行します。

例:

CATALOG TCP/IP 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** コマンドを実行しません。

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。

ここでのタスクの順序では、次のタスクは『連合システム・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:

- 33 ページの『連合システム・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』

連合システム・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

連合システム・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・ソースを連合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・データベースを連合サーバー・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、連合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・データベースを連合サーバー・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 の連合サーバーの場合は、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、連合サーバー・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・エントリー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連合サーバー・システム・データベース・ディレクトリーのエントリー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。

ここでのタスクの順序では、次のタスクは『DB2 ラッパーの登録』です。

関連タスク:

- 32 ページの『連合ノード・ディレクトリーへのノード項目のカタログ』
- 34 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・ソースを連合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・ソースにアクセスするために使用するラッパーを指定するには、CREATE WRAPPER ステートメントを実行します。どの DB2 Server Edition (Enterprise、Personal、および Workgroup) にも、DRDA と呼ばれる DB2 ファミリー用のラッパーが 1 つ含まれています。

手順:

DB2 ファミリー・データ・ソースにアクセスするために使用するラッパーを指定するには、CREATE WRAPPER ステートメントを実行します。

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。デフォルト名を使用してラッパーを登録する場合、連合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。ラッパー名が連合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、指定した名前です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。仮に、AIX で稼働する連合サーバーがあり、デフォルト名とは異なるラッパー名と使用することに決めたとします。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。

例:

```
CREATE WRAPPER mywrapper LIBRARY 'libdb2drda.a'
```

mywrapper は、デフォルトのラッパー名の代わりに指定したラッパーの名前です。

以下の表には、オペレーティング・システムごとの DB2 のラッパー・ライブラリー名がリストされています。

表 9. DB2 ラッパー・ライブラリー名

連合サーバーの オペレーティング・システム	ラッパーのライブラリー名
AIX	libdb2drda.a
Solaris オペレーティング環境	libdb2drda.so
HP-UX	libdb2drda.sl
Linux	libdb2drda.so
Windows NT および Windows 2000	db2drda.dll

ここでのタスクの順序では、次のタスクは『DB2 データ・ソースのサーバー定義の登録』です。

関連タスク:

- 35 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 WRAPPER ステートメント』

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・ソースを連合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。

連合データベース内に、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、連合サーバーはその 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連合グローバル・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 ステートメントを実行します。

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ユニーク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。サポートされるバージョンは以下のとおりです。

- DB2 for UNIX、および DB2 for Windows、バージョン 6、バージョン 7.1、バージョン 7.2、およびバージョン 8.1
- DB2 for z/OS and OS/390、バージョン 5 以上
- DB2 for iSeries、バージョン 4 以上

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あればなりません。

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 ステートメントを実行すると、連合サーバーは 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削除したりできます。

ここでのタスクの順序では、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:

- 38 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』
- 37 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・ソース側での許可 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 が連合サーバーより 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:

- 35 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』
- 397 ページの『付録 F. 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・ソースを連合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、連合サーバーは、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。連合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 (ユーザー・マッピング) を定義しなければなりません。連合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、分散要求を送信します。

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 ステートメントを実行します。

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。

ここでのタスクの順序では、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:

- 40 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』
- 39 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存する単一の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存する単一の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:

- 38 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・ソースを連合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` ステートメントを実行します。

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エントリーをチェックします。
- DB2 連合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。
- サーバー定義をチェックします。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。

ここでのタスクの順序では、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。

関連概念:

- 10 ページの『サーバー定義およびサーバー・オプション』

関連タスク:

- 「*DB2 Information Integrator* インストール・ガイド」の『データ・ソース環境変数をチェックする』
- 41 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『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・ソースを連合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。

連合データベースは、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。連合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・Catalogから読み取られ、連合サーバー上のグローバル・Catalogに置かれます。オブティマイザーは、データ・ソースのCatalog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。

手順:

ニックネームを登録するには、`CREATE NICKNAME` ステートメントを実行します。

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、連合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』
- 42 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のニックネームです。

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。

DB2 ファミリー・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

DB2 データ・ソースに対する構成をセットアップした後、構成を変更してパフォーマンスを改善できます。たとえば *DB2_DJ_COMM* DB2 プロファイル変数を設定して、DB2 データ・ソースにアクセスするときのパフォーマンスを改善できます。

DB2_DJ_COMM 変数を設定してパフォーマンスを改善する (UNIX)

たとえば、DB2 データ・ソース・サーバーのアクセスに極端に時間がかかる場合、*DB2_DJ_COMM* プロファイル・レジストリー変数を設定して、パフォーマンスを改善することができます。*DB2_DJ_COMM* 変数を設定する場合、連合サーバーは、データ・ソースにアクセスしようとするときではなく、初期設定時にラッパーをロードします。

手順:

DB2_DJ_COMM 変数を設定するには、次のようにします。

1. 指定したラッパーに対応するラッパー・ライブラリーに、*DB2_DJ_COMM* 変数を設定します。次の表にあるコマンドを使用して、*DB2_DJ_COMM* 変数を設定します。

表 10. DB2 データ・ソース用に *DB2_DJ_COMM* 変数を設定するコマンド

連合サーバーの オペレーティング・システム	コマンド
AIX	<i>DB2_DJ_COMM</i> = 'libdb2drda.a'
Solaris オペレーティング環境	<i>DB2_DJ_COMM</i> = 'libdb2drda.so'
HP-UX	<i>DB2_DJ_COMM</i> = 'libdb2drda.sl'
Linux	<i>DB2_DJ_COMM</i> = 'libdb2drda.so'
Windows NT および Windows 2000	<i>DB2_DJ_COMM</i> = 'db2drda.dll'

DB2_DJ_COMM 変数を設定するには、**db2set** コマンドを使用します。たとえば、連合サーバーのオペレーティング・システムが AIX の場合、コマンドは次のようになります。

```
db2set DB2_DJ_COMM='libdb2drda.a'
```

2. DB2_DJ_COMM 変数をエクスポートします。例:

```
export DB2_DJ_COMM
```

3. DB2 インスタンスをリサイクルするには、次のコマンドを実行します。

```
db2stop  
db2start
```

DB2 インスタンスをリサイクルすると、変数の変更を DB2 インスタンスが確実に受け入れます。

関連タスク:

- 31 ページの『連合サーバーへの DB2 ファミリー・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 5 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連合サーバーを構成する方法について説明します。

この章では、Informix データ・ソースに合わせて構成をセットアップする際に実行する必要がある作業のリスト、発行する必要がある SQL ステートメントの例、および使用できる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。

連合サーバーへの Informix データ・ソースの追加

連合サーバーが Informix データ・ソースにアクセスするように構成するには、アクセスしたい Informix データ・ソースとオブジェクトの情報をサーバーに提供します。

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、DB2 コマンド・センター、または DB2 コマンド行プロセッサを使用して構成できます。

DB2 コントロール・センターを使用する場合の利点は、各ステートメントやコマンドを入力する必要がないことです。DB2 コントロール・センターでは、Informix データ・ソースへのアクセスを迅速に構成する最も容易な方法が提供されています。ただし、以下の構成タスクを開始するために DB2 コントロール・センターを使用することはできません。

- 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
- サーバー定義とユーザー・マッピングを妥当性検査するための、Informix サーバーへの接続のテスト
- 列オプションの追加またはドロップ

前提条件:

- DB2 コマンド・センターまたは DB2 コマンド行プロセッサへのアクセス。
- Informix データ・ソースにアクセスするようにセットアップされた連合サーバーおよびデータベース。
- 連合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。
- システム環境変数、db2dj.ini 変数 (コード・ページ変換変数を含む)、および DB2 Profile Registry (db2set) 変数の適切なセットアップ。変数には、`INFORMIXDIR`、`INFORMIXSERVER`、`CLIENT_LOCALE` (オプション)、`DB_LOCALE` (オプション)、`DBNLS` (オプション)、および `INFORMIXSQLHOSTS`

(オプション) などが含まれます。 `INFORMIXSQLHOSTS` 変数は、`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にのみ指定してください。

- AIX 連合サーバーでは、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:

連合サーバーに Informix データ・ソースを追加するには、次のようにします。

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. 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. Informix サーバーとの接続をテストします。
6. 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:

- 1 ページの『データ・ソースの構成の近道』

関連タスク:

- 「*DB2 Information Integrator* インストール・ガイド」の『データ・ソース環境変数をチェックする』
- 46 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』
- 57 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 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・ソースを連合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、連合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。

- UNIX では、デフォルト名は `$INFORMIXDIR/etc/sqlhosts` です。 `sqlhosts` ファイルは Informix クライアント SDK のインストールされているところにあります。
- 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 Setnet32 ユーティリティを使用して、sqlhosts ファイルを作成するかレジストリーをセットアップします。

sqlhosts ファイルまたはレジストリーは、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できます。

Informix サーバーに接続するため、連合サーバー上に Informix クライアント SDK を構成することもでき、これによって sqlhosts ファイルまたはレジストリーが作成されます。連合サーバーは、Informix SDK ディレクトリーまたは Windows レジストリーにある、sqlhosts を使用します。

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/sqlhosts 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せん。

```
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。

- a. 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。
- 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
```

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** ツールが連合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。

ここでのタスクの順序では、次のタスクは『Informix ラッパーの登録』です。

関連タスク:

- 48 ページの『Informix ラッパーの登録』

- 57 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 ラッパーの登録

Informix ラッパーの登録は、連合サーバーに 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使用するラッパーを指定するには、`CREATE WRAPPER` ステートメントを使用します。どの DB2 Server Edition (Enterprise、Personal、Workgroup) にも、`INFORMIX` と呼ばれる Informix 用のラッパーが 1 つ含まれています。

手順:

Informix データ・ソースにアクセスするために使用するラッパーを指定するには、`CREATE WRAPPER` ステートメントを実行します。

例:

```
CREATE WRAPPER INFORMIX
```

推奨事項: デフォルトのラッパー名 `INFORMIX` 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、連合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。ラッパー名が連合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、指定した名前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。

仮に、AIX で稼働する連合サーバーがあり、デフォルト名とは異なるラッパー名を使用することに決めたとします。`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。

例:

```
CREATE WRAPPER mywrapper LIBRARY 'libdb2informix.a'
```

Informix 用のラッパー・ライブラリー名は次のとおりです。

表 11. Informix ラッパー・ライブラリー名

連合サーバーのオペレーティング・システム	ラッパーのライブラリー名
AIX	<code>libdb2informix.a</code>
HP-UX	<code>libdb2informix.sl</code>
Linux	<code>libdb2informix.so</code>

表 11. Informix ラッパー・ライブラリー名 (続き)

連合サーバーの オペレーティング・システム	ラッパーのライブラリー名
Solaris オペレーティング環境	libdb2informix.so
Windows NT および Windows 2000	db2informix.dll

ここでのタスクの順序では、次のタスクは『Informix データ・ソースのサーバー定義の登録』です。

関連タスク:

- 49 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 WRAPPER ステートメント』

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、連合サーバーに 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、連合データベース内に定義する必要があります。最初に Informix 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:

Informix 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. 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 onscotcp 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、onscotcp 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 ステートメントを実行します。

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。

ここでのタスクの順序では、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:

- 52 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』
- 379 ページの『付録 B. 連合システムのサーバー・オプション』
- 50 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、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 連合システムが、アプリケーションの保管点ステートメントの検出または作成を実施する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、連合サーバーは、Informix データ・ソースのニックネームに対する INSERT、UPDATE、または DELETE ステートメントを許可します。

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。

アプリケーション保管点の実施は、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。

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・オプションがあります。

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、連合サーバーは、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。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組み合わせを試行しないで済みます。この設定により、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ということが少なくなります。

関連タスク:

- 49 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・ソースを連合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、連合サーバーは、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。それぞれの連合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 (ユーザー・マッピング) を定義しなければなりません。連合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 ステートメントを実行します。

例:

```
CREATE USER MAPPING FOR USERID SERVER INFORMIXSERVER
    OPTIONS (REMOTE_AUTHID 'remote_id', REMOTE_PASSWORD 'remote_password')
```

ここでのタスクの順序では、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:

- 54 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』
- 53 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、連合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 を使用方法を示す例が含まれます。

完全指定の例:

以下の例は、連合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存する単一の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存する単一の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:

- 52 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・ソースを連合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` ステートメントを実行します。

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 連合サーバー上の 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 連合変数の設定を調べ、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。

ここでのタスクの順序では、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。

関連概念:

- 10 ページの『サーバー定義およびサーバー・オプション』

関連タスク:

- 「*DB2 Information Integrator インストール・ガイド*」の『連合サーバーのセットアップをチェックする』
- 「*DB2 Information Integrator インストール・ガイド*」の『データ・ソース環境変数をチェックする』
- 55 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『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、連合サーバーに 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。

連合データベースは、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。連合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atalogから読み取られ、連合サーバー上のグローバル・カatalogに入れられます。オブティマイザーはデータ・ソースのカatalog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 ステートメントを実行します。

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』
- 56 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のニックネーム。

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。

連合サーバーは、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:

- 55 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。たとえば DB2_DJ_COMM プロファイル・レジストリー変数を設定して、Informix データ・ソースにアクセスする時のパフォーマンスを改善することができます。

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、連合サーバーは、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。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。

- 仮に、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 と FOLD_PW の値を L (単一引用符で区切る) にすれば、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')
```

- 連合サーバーは、ユーザー ID とパスワードの大文字、小文字の組み合わせをすべて試すため、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超過して ID がロックアウトされ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')
```

DB2_DJ_COMM 変数を設定してパフォーマンスを改善する (UNIX)

たとえば、Informix サーバーのアクセスに時間がかかる場合は、DB2_DJ_COMM DB2 プロファイル・レジストリー変数を設定すれば、ラッパーをデータ・ソースのアクセス時ではなく、連合サーバーの初期設定時にロードすることになり、パフォーマンスを改善できます。

手順:

DB2_DJ_COMM 変数を設定するには、次のようにします。

1. 指定したラッパーに対応するラッパー・ライブラリーに、DB2_DJ_COMM 変数を設定します。連合サーバーが AIX で稼働しているとします。DB2_DJ_COMM 変数を設定するコマンドは次のようになります。

```
db2set DB2_DJ_COMM='libdb2informix.a,libdb2informixF.a,libdb2informixU.a'
```

次の表は、各オペレーティング・システムごとの DB2_DJ_COMM コマンドと適切なライブラリー名のリストです。

表 12. Informix データ・ソース用に DB2_DJ_COMM 変数を設定するコマンド

連合サーバーの オペレーティング・システム	コマンド
AIX	DB2_DJ_COMM= 'libdb2informix.a'
HP-UX	DB2_DJ_COMM= 'libdb2informix.sl'
Linux	DB2_DJ_COMM= 'libdb2informix.so'
Solaris オペレーティング環境	DB2_DJ_COMM= 'libdb2informix.so'

2. DB2 インスタンスをリサイクルするには、次のコマンドを実行します。

```
db2stop
db2start
```

DB2 インスタンスをリサイクルすると、変数の変更を DB2 インスタンスが確実に受け入れます。

関連タスク:

- 45 ページの『連合サーバーへの 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 6 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連合サーバーを構成する方法について説明します。

この章では、Oracle データ・ソースに合わせて構成をセットアップする際に実行する必要がある作業のリスト、発行する必要がある SQL ステートメントの例、および使用できる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。

連合サーバーへの Oracle データ・ソースの追加

連合サーバーが Oracle 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 Oracle 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。

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、DB2 コマンド・センター、または DB2 コマンド行プロセッサを通して構成できます。

DB2 コントロール・センターを使用する場合の利点は、各ステートメントやコマンドを入力する必要がないことです。DB2 コントロール・センターでは、Oracle データ・ソースへのアクセスを迅速に構成する最も容易な方法が提供されています。ただし、以下の構成タスクを開始するために DB2 コントロール・センターを使用することはできません。

- 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
- サーバー定義とユーザー・マッピングを妥当性検査するための、Oracle サーバーへの接続のテスト
- 列オプションの追加またはドロップ

前提条件:

- DB2 コマンド・センターまたは DB2 コマンド行プロセッサへのアクセス。
- Oracle データ・ソースにアクセスするようセットアップされた連合サーバーおよびデータベース。
- Oracle クライアント・ソフトウェアが連合サーバー上にインストールされ、構成済みであること。
- 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。変数には、ORACLE_HOME、ORACLE_BASE、ORA_NLS、および TNS_ADMIN があります。

手順:

連合サーバーに Oracle データ・ソースを追加するには、次のようにします。

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. 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. Oracle サーバーとの接続をテストします。
6. Oracle 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:

- 1 ページの『データ・ソースの構成の近道』

関連タスク:

- 「*DB2 Information Integrator インストール・ガイド*」の『データ・ソース環境変数をチェックする』
- 62 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』
- 73 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 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・ソースを連合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、連合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、次のようにします。

1. Oracle クライアント・ソフトウェアに付属しているユーティリティーを使用します。

このユーティリティーの使用法の詳細は、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、連合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。

ここでのタスクの順序では、次のタスクは『Oracle ラッパーの登録』です。

関連タスク:

- 63 ページの『Oracle ラッパーの登録』
- 73 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、連合サーバーに 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使用するラッパーを指定するには、CREATE WRAPPER ステートメントを使用します。DB2 Information Integrator には、Oracle 用の 2 つのラッパー SQLNET および NET8 が付属しています。

手順:

Oracle データ・ソースへのアクセスに使用するラッパーを指定するには、CREATE WRAPPER ステートメントを使用します。

例:

```
CREATE WRAPPER NET8
```

CREATE WRAPPER ステートメントでどちらのラッパー (SQLNET または NET8) を使用するかを決める際は、このトピックの最後にある『関連資料』を参照してください。

SQLNET ラッパーは OCI 7 (Oracle コール・インターフェース) API 呼び出しを使用します。NET8 ラッパーは OCI 8 API 呼び出しを使用します。Oracle 8 または Oracle 9 クライアントがインストールされている場合は、NET8 ラッパーを使用するとパフォーマンスがよく、機能も強化されています。さらに、NET8 ラッパーには LOB サポートがあります。OCI 7 は LOB データ・タイプをサポートしないため、SQLNET ラッパーは Oracle LOB データ・タイプをサポートしません。

- SQLNET ラッパーは、Oracle LONG データ・タイプを DB2 for UNIX および DB2 for Windows の LOB データ・タイプにマップします。
- NET8 ラッパーは、Oracle LONG データ・タイプをサポートしていません。このラッパーは、Oracle LOB データ・タイプを DB2 for UNIX および DB2 for Windows の LOB データ・タイプにマップします。

推奨事項: デフォルトのラッパー名 (SQLNET または NET8) 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、連合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。ラッパー名が連合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、指定した名前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。

仮に、AIX で稼働する連合サーバーがあり、デフォルト名とは異なるラッパー名を使用することに決めたとします。SQLNET と NET8 用の CREATE WRAPPER ステートメントの例を次に示します。

```
CREATE WRAPPER mywrapper LIBRARY 'libdb2sqlnet.a'
```

```
CREATE WRAPPER mywrapper LIBRARY 'libdb2net8.a'
```

Oracle ラッパー・ライブラリー名のリストは、このトピックの最後にある『関連資料』を参照してください。

ここでのタスクの順序では、次のタスクは『Oracle データ・ソースのサーバー定義の登録』です。

関連タスク:

- 66 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』
- 64 ページの『Oracle ラッパーおよびライブラリー名』

Oracle ラッパーおよびライブラリー名

このトピックでは、Oracle データ・ソースにアクセスするラッパーを登録する場合に使用できる Oracle ラッパー名および Oracle ライブラリー名が提供されます。

以下の表は、使用している Oracle クライアント・バージョンおよびオペレーティング・システムに応じて使用する Oracle ラッパー名のリストです。

表 13. クライアント・バージョンおよびオペレーティング・システム別の Oracle ラッパー

Oracle クライアント	オペレーティング・システム	使用するラッパー
Oracle バージョン 7	AIX	SQLNET
	Windows NT および Windows 2000	SQLNET
	HP-UX、Linux、および Solaris オペレーティング環境	該当せず
Oracle バージョン 8	AIX	NET8
	Windows NT または Windows 2000	NET8 (推奨) または SQLNET
	HP-UX、Linux、および Solaris	NET8
Oracle バージョン 9	AIX	NET8
	Windows NT または Windows 2000	NET8 (推奨) または SQLNET
	HP-UX、Linux、および Solaris	NET8

以下の表は、連合サーバーのオペレーティング・システムに応じて使用する Oracle ラッパーのライブラリー名のリストです。

表 14. Oracle ラッパーのライブラリー名

連合サーバーのオペレーティング・システム	SQLNET のラッパー・ライブラリー名	NET8 のラッパー・ライブラリー名
AIX	libdb2sqlnet.a	libdb2net8.a
HP-UX	libdb2sqlnet.sl	libdb2net8.sl
Linux	libdb2sqlnet.so	libdb2net8.so
Solaris オペレーティング環境	libdb2sqlnet.so	libdb2net8.so
Windows NT および Windows 2000	db2sqlnet.dll	db2net8.dll

関連タスク:

- 66 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、連合サーバーに 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。

連合データベース内に、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.see1)))
```

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 ステートメントを実行します。

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。

ここでのタスクの順序では、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:

- 68 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』
- 379 ページの『付録 B. 連合システムのサーバー・オプション』
- 67 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にある場合は、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')
```

末尾の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。末尾にブランクを含まない VARCHAR 列がいくつかしかない場合は、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:

- 66 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・ソースを連合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、連合サーバーは、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。それぞれの連合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 (ユーザー・マッピング) を定義しなければなりません。連合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、Oracle 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 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 ステートメントを実行します。

例:

```
CREATE USER MAPPING FOR userid SERVER oraserver
      OPTIONS (REMOTE_AUTHID 'remote_id', REMOTE_PASSWORD 'remote_password')
```

ここでのタスクの順序では、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:

- 70 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』
- 69 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、連合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、連合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存する単一の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存する単一の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:

- 68 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・ソースを連合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` ステートメントを実行します。

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 連合サーバー上の Oracle クライアント・ソフトウェアを調べ、これが正しくインストールされ、Oracle 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 連合変数の設定を調べ、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。

ここでのタスクの順序では、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。

関連概念:

- 10 ページの『サーバー定義およびサーバー・オプション』

関連タスク:

- 「*DB2 Information Integrator インストール・ガイド*」の『連合サーバーのセットアップをチェックする』
- 「*DB2 Information Integrator インストール・ガイド*」の『データ・ソース環境変数をチェックする』
- 71 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『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、連合サーバーに 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。

連合データベースは、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。連合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、連合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 ステートメントを実行します。

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』
- 72 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のニックネーム。

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。

連合サーバーは、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:

- 71 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。たとえば DB2_DJ_COMM プロファイル・レジストリー変数を設定して、Oracle データ・ソースにアクセスする時のパフォーマンスを改善することができます。

DB2_DJ_COMM 変数を設定してパフォーマンスを改善する (UNIX)

Oracle サーバーのアクセスに時間がかかる場合、DB2_DJ_COMM 変数を設定して、パフォーマンスを改善できます。DB2_DJ_COMM を設定する場合、連合サーバーは、データ・ソースにアクセスしようとするときではなく、初期設定時にラッパーをロードします。

手順:

DB2_DJ_COMM 変数を設定するには、次のようにします。

1. 指定したラッパーに対応するラッパー・ライブラリーに、DB2_DJ_COMM 変数を設定します。仮に、連合サーバーが AIX で稼働していて、使用するラッパーが NET8 であるとする、DB2_DJ_COMM 変数を設定するコマンドは次のようになります。

```
db2set DB2_DJ_COMM= 'libdb2net8.a'
```

次の表は、有効な Oracle ライブラリー名のリストです。

表 15. Oracle ラッパーのライブラリー名

連合サーバーのオペレーティング・システム	SQLNET ラッパーのライブラリー名	NET8 ラッパーのライブラリー名
AIX	libdb2sqlnet.a	libdb2net8.a

表 15. Oracle ラッパーのライブラリー名 (続き)

連合サーバーのオペレーティング・システム	SQLNET ラッパーのライブラリー名	NET8 ラッパーのライブラリー名
HP-UX	libdb2sqlnet.sl	libdb2net8.sl
Linux	libdb2sqlnet.so	libdb2net8.so
Solaris オペレーティング環境	libdb2sqlnet.so	libdb2net8.so

2. DB2 インスタンスをリサイクルするには、次のコマンドを実行します。

```
db2stop
db2start
```

DB2 インスタンスをリサイクルすると、変数の変更を DB2 インスタンスが確実に受け入れます。

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、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りリモート・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・ホストのためのエントリーが必要です。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、ファイルのロケーションは、連合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 連合サーバーの場合

/etc/hosts ファイルを更新。

Windows 連合サーバーの場合

x:\winnt\system32\drivers\etc\hosts ファイルを更新。

関連タスク:

- 61 ページの『連合サーバーへの 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 7 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連合サーバーを構成する方法について説明します。

この章では、Sybase データ・ソースに合わせて構成をセットアップする際に実行する必要がある作業のリスト、発行する必要がある SQL ステートメントの例、および使用できる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。

連合サーバーへの Sybase データ・ソースの追加

連合サーバーが 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 Sybase データ・ソースとオブジェクトの情報をサーバーに提供します。

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、DB2 コマンド・センター、または DB2 コマンド行プロセッサを通して構成できます。

DB2 コントロール・センターを使用する場合の利点は、各ステートメントやコマンドを入力する必要がないことです。DB2 コントロール・センターでは、Sybase データ・ソースへのアクセスを迅速に構成する最も容易な方法が提供されています。ただし、以下の構成タスクを開始するために DB2 コントロール・センターを使用することはできません。

- 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
- サーバー定義とユーザー・マッピングを妥当性検査するための、Sybase サーバーへの接続のテスト
- 列オプションの追加またはドロップ

前提条件:

- DB2 コマンド・センターまたは DB2 コマンド行プロセッサへのアクセス。
- Sybase データ・ソースにアクセスするようにセットアップされた連合サーバーおよびデータベース。
- Sybase クライアント・ソフトウェアが連合サーバー上にインストールされ、構成済みであること。
- 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。変数には、SYBASE および SYBASE_OCS があります。

制約事項:

DBLIB と呼ばれる Sybase Open Client DB-Library ラッパーは、読み取り専用で、INSERT、UPDATE、または DELETE 操作をサポートしません。

手順:

連合サーバーに 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. 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. Sybase サーバーとの接続をテストします。
6. 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:

- 1 ページの『データ・ソースの構成の近道』

関連タスク:

- 「*DB2 Information Integrator インストール・ガイド*」の『データ・ソース環境変数をチェックする』
- 76 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』

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・ソースを連合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、連合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 DB2 連合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。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、連合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/interfaces` ディレクトリに作成されます。このユーティリティの使用法の詳細は、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 DB2 連合サーバー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 DB2 連合インスタンスの `$HOME/sqllib` ディレクトリにコピーします。
 - `ln` コマンドを使用して、`/sqllib` サブディレクトリからインスタンスの `$HOME/sqllib` ディレクトリ内の `interfaces` ファイルへのリンクを作成します。
例:

```
ln -s -f /home/sybase/interfaces /home/db2djinst1/sqllib
```
 - `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 (`isql` など) を使用してください。

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\sql.ini` ディレクトリに作成されます。このユーティリティの使用法の詳細は、Sybase の資料を参照してください。
2. この `sql.ini` ファイルを DB2 連合インスタンスの `c:\Program Files\IBM\SQLLIB` ディレクトリにコピーし、このファイルを DB2 連合サーバー・インスタンスに対してアクセス可能にします。

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。

ここでのタスクの順序では、次のタスクは『Sybase ラッパーの登録』です。

関連タスク:

- 78 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、連合サーバーに 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使用するラッパーを指定するには、`CREATE WRAPPER` ステートメントを使用します。DB2 Information Integrator には、Sybase 用に 2 つのラッパーが組み込まれています。1 つは `CTLIB` という Open Client Client-Library ラッパーで、もう 1 つは `DBLIB` という Open Client DB-Library ラッパーです。

手順:

Sybase データ・ソースへのアクセスに使用するラッパーを指定するには、`CREATE WRAPPER` ステートメントを使用します。

例:

```
CREATE WRAPPER CTLIB
```

連合サーバーで稼働しているオペレーティング・システムに関係なく、`CTLIB` または `DBLIB` ラッパーのいずれかを使用することができます。

推奨事項: デフォルトのラッパー名 (`CTLIB` または `DBLIB`) のいずれか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、連合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。ラッパー名が連合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、指定した名前でも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。仮に、AIX で稼働する連合サーバーがあり、デフォルト名とは異なるラッパー名を使用することに決めたとします。この場合、`CTLIB` および `DBLIB` 用の `CREATE WRAPPER` ステートメントの例は次のようになります。

```
CREATE WRAPPER mywrapper LIBRARY 'libdb2ctlib.a'
```

```
CREATE WRAPPER mywrapper LIBRARY 'libdb2dblib.a'
```

Sybase 用のラッパー・ライブラリー名は次のとおりです。

表 16. Sybase ラッパーのライブラリー名

連合サーバーのオペレーティング・システム	CTLIB ラッパーのライブラリー名	DBLIB ラッパーのライブラリー名
AIX	libdb2ctlib.a	libdb2dblib.a
HP-UX	libdb2ctlib.sl	libdb2dblib.sl
Linux	libdb2ctlib.so	libdb2dblib.so

表 16. Sybase ラッパーのライブラリー名 (続き)

連合サーバーのオペレーティング・システム	CTLIB ラッパーのライブラリー名	DBLIB ラッパーのライブラリー名
Solaris オペレーティング環境	libdb2ctlib.so	libdb2dblib.so
Windows NT および Windows 2000	db2ctlib.dll	db2dblib.dll

ここでのタスクの順序では、次のタスクは『Sybase データ・ソースのサーバー定義の登録』です。

関連タスク:

- 79 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 WRAPPER ステートメント』

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、連合サーバーに 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。

連合データベース内に、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。最初に Sybase 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:

Sybase 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. 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 ステートメントを実行します。

例:

```
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。

ここでのタスクの順序では、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:

- 83 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』
- 379 ページの『付録 B. 連合システムのサーバー・オプション』
- 80 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 および DBLIB 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:

サーバーを定義する際に、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 オペレーティング・システムで CTLIB ラッパーを使用してサーバーを登録する場合に、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', LOGIN_TIMEOUT '60', PACKET_SIZE '1024',
    IFILE '/home/sybase/interfaces')
```

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 オペレーティング・システムでサーバーを登録する場合に、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')
```

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。

IFILE

Sybase Open Client interfaces ファイルの絶対パスと名前を指定します。

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。

IGNORE_UDT

連合サーバーが、「強い型定義 (strong typing)」を持たない UDT の基となる組み込みタイプを判別するかどうかを指定します。

LOGIN_TIMEOUT

接続を試みた時に、DB2 Universal Database がログイン応答をどれだけの時間待つかを示す秒数を指定します。デフォルトでは、Sybase サーバーからの応答を無期限で待ちます。

PACKET_SIZE

Tabular Data Stream (TDS) パケットを送信する時に Client-Library が使用するパケット・サイズを決めます。アプリケーションが大量のテキスト、イメージ、またはバルク・データを送受信する必要がある場合は、パケット・サイズを大きくすると効率が向上する場合があります。

TIMEOUT

コマンドに対するサーバーの応答を DB2 Universal Database がどれだけの時間待つかを示す秒数を指定します。デフォルトでは、Sybase サーバーからの応答を無期限で待ちます。Sybase Open Client は、タイムアウトしきい値を使用して、長い時間実行されている照会や応答に、割り込みをかけます。

関連タスク:

- 79 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』

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・ソースを連合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、連合サーバーは、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。それぞれの連合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 (ユーザー・マッピング) を定義しなければなりません。連合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 ステートメントを実行します。

例:

```
CREATE USER MAPPING FOR userid SERVER SYBSERVER
    OPTIONS (REMOTE_AUTHID 'remote_id', REMOTE_PASSWORD 'remote_password')
```

ここでのタスクの順序では、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:

- 84 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』
- 83 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、連合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、連合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存する単一の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存する単一の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:

- 83 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・ソースを連合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 ステートメントを実行します。

例:

```
SET PASSTHRU local_server_name
SELECT count(*) FROM dbo.sysobjects
SET PASSTHRU RESET
```

ここで、*local_server_name* は、連合データベース・カタログにリモート・サーバーを登録するときに使用した名前です。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。
 - 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 連合サーバー上の 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 連合変数の設定を調べ、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。

ここでのタスクの順序では、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。

関連概念:

- 10 ページの『サーバー定義およびサーバー・オプション』

関連タスク:

- 「DB2 Information Integrator インストール・ガイド」の『連合サーバーのセットアップをチェックする』
- 「DB2 Information Integrator インストール・ガイド」の『データ・ソース環境変数をチェックする』
- 86 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『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、連合サーバーに 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。

連合データベースは、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。連合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atalogから読み取られ、連合サーバー上のグローバル・カatalogに入られます。オプティマイザーはデータ・ソースのカatalog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 ステートメントを実行します。

例:

```
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、DB2 は接続を使用してデータ・ソース・カatalog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』
- ・ 87 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ユニーク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許可 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。

連合サーバーは、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:

- 86 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。たとえば DB2_DJ_COMM 環境変数を設定して、Sybase データ・ソースにアクセスする時のパフォーマンスを改善することができます。

DB2_DJ_COMM 環境変数を設定してパフォーマンスを改善する (UNIX)

Sybase サーバーのアクセスに時間がかかる場合、DB2_DJ_COMM 環境変数を設定して、パフォーマンスを改善できます。DB2_DJ_COMM 環境変数を設定する場合、連合サーバーは、データ・ソースにアクセスしようとするときではなく、初期設定時にラッパーをロードします。

手順:

DB2_DJ_COMM 環境変数を設定するには、次のようにします。

1. 指定したラッパーに対応するラッパー・ライブラリーに、DB2_DJ_COMM 環境変数を設定します。ここで、連合サーバーが AIX で稼働し、使用するラッパーは CTLIB であるとしめます。DB2_DJ_COMM 環境変数を設定するコマンドは次のようになります。

```
db2set DB2_DJ_COMM= 'libdb2ctlib.a'
```

適切なライブラリー名については、次の表を参照してください。

表 17. Sybase ラッパーのライブラリー名

連合サーバーのオペレーティング・システム	CTLIB ラッパーのライブラリー名	DBLIB ラッパーのライブラリー名
AIX	libdb2ctlib.a	libdb2dblib.a
HP-UX	libdb2ctlib.sl	libdb2dblib.sl
Linux	libdb2ctlib.so	libdb2dblib.so
Solaris オペレーティング環境	libdb2ctlib.so	libdb2dblib.so

2. DB2 インスタンスをリサイクルするには、次のコマンドを実行します。

```
db2stop  
db2start
```

DB2 インスタンスをリサイクルすると、変数の変更を DB2 インスタンスが確実に受け入れます。

DBLIB ではなく CTLIB を使用

CT-Library はステートメントの動的な「準備/実行」をサポートします。これにより CT-Library アプリケーションは、ステートメントを一度準備すれば、いろいろな入力を使用してこのステートメントを何度も実行することができます。ステートメントの準備が一度で済めば、入力パラメーターが変わるたびにステートメントを再コンパイルする必要がなくなります。DB2 アプリケーションは動的 SQL を利用できませんが、リモート照会の連合照会処理は、動的 SQL のみを使用します。

sp_helpindex エラーを解決する

連合システムは 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:

- 75 ページの『連合サーバーへの 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 8 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連合サーバーを構成する方法について説明します。

この章では、Microsoft SQL Server データ・ソースに合わせて構成をセットアップする際に行う必要がある作業のリスト、発行する必要がある SQL ステートメントの例、および使用できる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。

連合サーバーへの 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連合サーバーを構成するには、アクセスしたい Microsoft SQL Server データ・ソースとオブジェクトの情報を連合サーバーに提供します。

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、DB2 コマンド・センター、または DB2 コマンド行プロセッサを通して構成できません。

DB2 コントロール・センターを使用する場合の利点は、各ステートメントやコマンドを入力する必要がないことです。DB2 コントロール・センターでは、Microsoft SQL Server データ・ソースへのアクセスを迅速に構成する最も容易な方法が提供されています。ただし、以下の構成タスクを開始するために DB2 コントロール・センターを使用することはできません。

- サーバー定義とユーザー・マッピングを検証するための、Microsoft SQL Server サーバーへの接続のテスト
- 列オプションの追加またはドロップ

前提条件:

- DB2 コマンド・センターまたは DB2 コマンド行プロセッサへのアクセス。
- Microsoft SQL Server データ・ソースにアクセスするようにセットアップされた連合サーバーおよびデータベース。
- Microsoft SQL Server ODBC ドライバーが連合サーバー上にインストール済みで、構成済みであること。
- 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。変数には、DJXODBCTRACE、DJX_ODBC_LIBRARY_PATH、ODBCINI、DB2LIBPATH、および DB2ENVLIST があります。

手順:

連合サーバーに Microsoft SQL Server データ・ソースを追加するには、以下のようになります。

1. 連合サーバーと連合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. 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. Microsoft SQL Server リモート・サーバーへの接続をテストします。
6. 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:

- 1 ページの『データ・ソースの構成の近道』

関連タスク:

- 「*DB2 Information Integrator* インストール・ガイド」の『データ・ソース環境変数をチェックする』
- 92 ページの『Microsoft SQL Server データ・ソースにアクセスするための連合サーバーとデータベースの準備』

Microsoft SQL Server データ・ソースにアクセスするための連合サーバーとデータベースの準備

Microsoft SQL Server データ・ソースにアクセスさせるための連合サーバーとデータベースの準備は、連合サーバーに 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の連合サーバーとデータベースの準備に必要なステップは、連合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:

連合サーバーとデータベースの準備は、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。
 - a. 「スタート」メニューから、「コントロール パネル」を開きます。
 - b. 「ODBC データ ソース」をダブルクリックし、「ODBC データ ソース アドミニストレータ」ウィンドウを開きます。
 - c. 「システム DSN」タブをクリックし、Microsoft SQL Server リモート・サーバーのエントリーを見つけます。

このエントリーは、連合データベースで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 システムの場合:

1. 連合サーバー上の `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。
3. 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 `$DJX_ODBC_LIBRARY_PATH/libodbcinst.sl` です。
 - Linux の場合、シンボリック・リンクは、`/usr/local/locale` から `$DJX_ODBC_LIBRARY_PATH/../locale` です。
 - Solaris オペレーティング環境の場合、シンボリック・リンクは `$HOME/sql1lib/locale` から `$DJX_ODBC_LIBRARY_PATH/../locale` 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。
4. DataDirect Connect ODBC **demoodbc** ツールを使用して、連合サーバーから Microsoft SQL Server データ・ソースへの接続をテストします。
 - a. 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実行します。

```
export ODBCINI=$HOME/.odbc.ini
```
 - b. `/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。
 - c. 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。

ここでのタスクの順序では、次のタスクは『Microsoft SQL Server ラッパーの登録』です。

関連タスク:

- 94 ページの『Microsoft SQL Server ラッパーの登録』

Microsoft SQL Server ラッパーの登録

Microsoft SQL Server ラッパーの登録は、連合サーバーに 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・ソースへのアクセスに使用するラッパーを指定するには、CREATE WRAPPER ステートメントを実行します。DB2 Information Integrator には、Microsoft SQL Server 用に 2 つのラッパーが組み込まれています。使用するラッパーは、連合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。

- UNIX システムの場合、デフォルトのラッパー名は、MSSQLODBC3 (DataDirect Connect ODBC 3.7 (またはそれ以降) 用) ドライバーです。
- Windows の場合、デフォルトのラッパー名は、DJXMSSQL3 (ODBC 3.0 (またはそれ以降) 用) ドライバーです。

手順:

Microsoft SQL Server データ・ソースへのアクセスに使用するラッパーを指定するには、CREATE WRAPPER ステートメントを実行します。

たとえば、Windows NT および Windows 2000 の場合、次のようにします。

```
CREATE WRAPPER DJXMSSQL3
```

推奨: いずれかのデフォルト・ラッパー名 (DJXMSSQL3 または MSSQLODBC3) 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、連合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。ラッパー名が連合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、指定した名前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。仮に、AIX で稼働する連合サーバーがあり、デフォルト名とは異なるラッパー名を使用することに決めたとします。この場合、次のような CREATE WRAPPER ステートメントを発行する必要があります。

```
CREATE WRAPPER mywrapper LIBRARY 'libdb2mssql3.a'
```

値 *mywrapper* は、デフォルトのラッパー名の代わりに指定したラッパーの名前です。

Microsoft SQL Server 用のラッパー・ライブラリー名は以下のとおりです。

表 18. Microsoft SQL Server ラッパーのライブラリー名

連合サーバーの オペレーティング・システム	ラッパーのライブラリー名
AIX	libdb2mssql3.a
HP-UX	libdb2mssql3.sl
Linux	libdb2mssql3.so
Solaris オペレーティング環境	libdb2mssql3.so
Windows	db2mssql3.dll

ここでのタスクの順序では、次のタスクは『Microsoft SQL Server データ・ソースのサーバー定義の登録』です。

関連タスク:

- 95 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 WRAPPER ステートメント』

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、連合サーバーに 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。

連合データベース内に、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. ノード名を見つけます。
 - 連合サーバーが Windows NT または Windows 2000 を使用する場合、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。
 - 連合サーバーが AIX、HP-UX、Linux、または Solaris オペレーティング環境を使用する場合、NODE は .odbc.ini ファイルに定義します。

次の例は、AIX での .odbc.ini ファイルです。

例: AIX の場合の .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 ファイルの最上部には、[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。各ノードには、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 ステートメントを実行します。

例:

```
CREATE SERVER server_name TYPE MSSQLSERVER VERSION 7.0 WRAPPER djxmssql3
  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。

ここでのタスクの順序では、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:

- 98 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』
- 379 ページの『付録 B. 連合システムのサーバー・オプション』
- 97 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 dxmssql3  
    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 *dxmssql3*

CREATE WRAPPER ステートメントで指定したラッパー名。

NODE *'sqlnode'*

Microsoft SQL Server リモート・サーバーが存在するノードの名前。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、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:

サーバーを定義する際に、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 djxmssql3
    OPTIONS (NODE 'sqlnode', DBNAME 'africa', COLLATING_SEQUENCE 'I')
```

COLLATING_SEQUENCE サーバー・オプションは、データ・ソースが連合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の区別をしません (例: 'STEWART' と 'StewART' は等しいと見なされます)。連合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示しています。

注: データ・ソースから戻される結果が連合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、連合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、連合サーバーは、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 または MAX)

関連タスク:

- 95 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、連合サーバーに 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、連合サーバーは、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。それぞれの連合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 (ユーザー・マッピング) を定義しなければなりません。連合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、分散要求を送信します。

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 ステートメントを実行します。

例:

```
CREATE USER MAPPING FOR userid SERVER sqlserver
    OPTIONS (REMOTE_AUTHID 'remote_id', REMOTE_PASSWORD 'remote_password')
```

ここでのタスクの順序では、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:

- 100 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』
- 99 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、連合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、連合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存する単一の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存する単一の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:

- 98 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、連合サーバーに 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 ステートメントを実行します。

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 のオプションが 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 連合サーバー上の ODBC ドライバーを調べ、これがインストールされ、Microsoft SQL Server リモート・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 連合変数の設定を調べ、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。

ここでのタスクの順序では、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。

関連概念:

- 10 ページの『サーバー定義およびサーバー・オプション』

関連タスク:

- 「*DB2 Information Integrator インストール・ガイド*」の『連合サーバーのセットアップをチェックする』
- 「*DB2 Information Integrator インストール・ガイド*」の『データ・ソース環境変数をチェックする』
- 101 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『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、連合サーバーに 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。

定義するそれぞれの Microsoft SQL Server 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。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。

連合データベースは、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。連合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ATALOGから読み取られ、連合サーバー上のグローバル・カATALOGに入れられます。オプティマイザーはデータ・ソースのカATALOG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 ステートメントを実行します。

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ATALOG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』
- 103 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 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のニックネーム。

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。

連合サーバーは、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:

- 101 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。たとえば DB2_DJ_COMM プロファイル・レジストリー変数を設定して、連合サーバーが Microsoft SQL Server データ・ソースにアクセスする時のパフォーマンスを改善したい場合があります。

DB2_DJ_COMM 変数を設定してパフォーマンスを改善する (UNIX)

Microsoft SQL Server リモート・サーバーのアクセスに時間がかかる場合、DB2_DJ_COMM DB2 プロファイル・レジストリー変数を設定して、パフォーマンスを改善できます。DB2_DJ_COMM 変数を設定する場合、連合サーバーは、データ・ソースにアクセスしようとするときではなく、初期設定時にラッパーをロードします。

手順:

DB2_DJ_COMM 変数を設定するには、次のようにします。

1. 指定したラッパーに対応するラッパー・ライブラリーに、DB2_DJ_COMM 変数を設定します。仮に、連合サーバーが AIX で稼働していて、使用するラッパーが MSSQLODBC3 であるとする、DB2_DJ_COMM 変数を設定するコマンドは次のようになります。

```
db2set DB2_DJ_COMM='libdb2mssql3.a'
```

以下の表には、サポートされているオペレーティング・システムごとに有効なライブラリー名がリストされています。

表 19. Microsoft SQL Server ラッパーのライブラリー名

連合サーバーのオペレーティング・システム	MSSQLODBC3 ラッパーのライブラリー名	DJXMSSQL3 ラッパーのライブラリー名
AIX	libdb2mssql3.a	なし
HP-UX	libdb2mssql3.sl	なし
Linux	libdb2mssql3.so	なし
Solaris オペレーティング環境	libdb2mssql3.so	なし
Windows NT および Windows 2000	なし	db2mssql3.dll

2. DB2 インスタンスをリサイクルするには、次のコマンドを実行します。

```
db2stop  
db2start
```

DB2 インスタンスをリサイクルすると、変数の変更を DB2 インスタンスが確実に受け入れます。

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。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。したがって、問題が解決したら、トレースはオフにしておいてください。

Windows 連合サーバーでは、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 連合サーバーでは、db2dj.ini ファイルに DJXODBCTRACE 変数を設定してください。例:

```
DJXODBCTRACE=/home/user1/trace_dir/filename.xxx
```

また、.odbc.ini ファイルでトレースをオンにセットする必要があります。たとえば、DataDirect ODBC 3.x ドライバーを使用しているとします。クライアント・ディレクトリー内に .odbc.ini ファイルの例があります。このファイルには、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==/opt/odbc/lib/odbcdrac.so
InstallDir=/opt/odbc
```

最初の行は、トレースを OFF にするなら Trace=0 とし、トレースを ON にするなら Trace=1 と設定されます。TraceFile は、この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。このパスおよびファイル名は、db2dj.ini ファイル内の行 DJXODBCTRACE=/home/user1/trace_dir/filename.xxx と一致している必要があります。

関連タスク:

- 91 ページの『連合サーバーへの 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 9 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連合サーバーを構成する方法について説明します。

この章では、ODBC データ・ソースに合わせて構成をセットアップする際に実行する必要がある作業のリスト、発行する必要がある SQL ステートメントの例、および使用できる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。

連合サーバーへの ODBC データ・ソースの追加

DB2 Information Integrator が提供するラッパーは、ODBC API を通してアクセスされる特定のデータ・ソースをサポートしています。これらのデータ・ソースには、Oracle、Microsoft SQL Server、および Microsoft Excel などがあります。これらのデータ・ソース用に特化して設計されたラッパーを使用すれば、よりよいパフォーマンスが得られます。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。

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。

ODBC ラッパーは、ODBC バージョン 3.x をサポートしています。

連合サーバーが ODBC 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 ODBC データ・ソースとオブジェクトの情報を連合サーバーに渡します。

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、DB2 コマンド・センター、または DB2 コマンド行プロセッサを通して構成できます。

DB2 コントロール・センターを使用する場合の利点は、各ステートメントやコマンドを入力する必要がないことです。DB2 コントロール・センターでは、ODBC データ・ソースへのアクセスを迅速に構成する最も容易な方法が提供されています。

ODBC ラッパーは、以下のオペレーティング・システムを使用する連合サーバーで使用できます。

- AIX
- HP-UX
- Linux on Intel オペレーティング・システム
- Solaris オペレーティング環境

- Windows NT、Windows 2000、Windows .NET

前提条件:

- DB2 コマンド・センターまたは DB2 コマンド行プロセッサへのアクセス。
- ODBC データ・ソースにアクセスするようセットアップされた連合サーバーおよびデータベース。
- ODBC ドライバーが連合サーバー上にインストール済みで、構成済みであること。
- 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 バージョン 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
 - リモート保管場所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
- 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、一部の複雑な検索済み UPDATE および DELETE ステートメントは、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、失敗します。
- ODBC ラッパーは、ほとんどのデータ・ソースでの読み取りおよび書き込み操作をサポートしています。

手順:

連合サーバーに ODBC データ・ソースを追加するには、次のようにします。

1. 連合サーバーと連合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:

- 1 ページの『データ・ソースの構成の近道』

関連タスク:

- 「DB2 Information Integrator インストール・ガイド」の『**連合サーバーのセットアップ**をチェックする』
- 109 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連合サーバーとデータベースの準備**』

ODBC を通してデータ・ソースにアクセスするための連合サーバーとデータベースの準備

ODBC を通してデータ・ソースにアクセスするための連合サーバーとデータベースの準備は、ODBC データ・ソースを連合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 を通してデータ・ソースにアクセスするための連合サーバーとデータベースの準備に必要なステップは、連合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、固有のライブラリー・パス・ロケーションがあります。

手順:

連合サーバーとデータベースの準備は、以下のように行います。

Windows の場合:

1. ODBC システム DSN が ODBC データ・ソースに接続するように設定されているかどうかを確認します。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。この設定は、「コントロール パネル」で確認します。
 - a. 「スタート」メニューから、「コントロール パネル」を開きます。
 - b. 「ODBC データ ソース」をダブルクリックし、「ODBC デバイス マネージャ」にアクセスします。
 - c. 「システム DSN」タブをクリックし、ODBC ドライバー用に定義したシステム DSN がリストに表示されることを確認します。

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。
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 ODBC データ・ソースへの接続をテストします。

UNIX システムの場合:

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が提供している資料を参照してください。

ここでのタスクの順序では、次のタスクは『ODBC ラッパーの登録』です。

関連タスク:

- 110 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、連合サーバーに 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 ラッパーを登録するには、CREATE WRAPPER ステートメントを実行する必要があります。

手順:

ODBC データ・ソースへのアクセスに使用するラッパーを指定するには、CREATE WRAPPER ステートメントを使用します。

例:

```
CREATE WRAPPER odbc
```

推奨事項: CREATE WRAPPER ステートメントを実行する際は、デフォルトのラッパー名 ODBC 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、連合サーバーは、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。ラッパー名が連合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、指定した名前でも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。仮に、AIX で稼働する連合サーバーがあり、デフォルト名とは異なるラッパー名を使用することに決めたとします。この場合、次のような CREATE WRAPPER ステートメントを発行する必要があります。

```
CREATE WRAPPER mywrapper  
LIBRARY 'libdb2rcodbc.a' OPTIONS (MODULE '/usr/lib/odbc.a')
```

MODULE '*/usr/lib/odbc.a*' は、ODBC ドライバー・マネージャーを含むライブラリーの絶対パスです。

アクセスする計画がある ODBC データ・ソースの数に関係なく、ODBC ラッパーを登録する必要があるのは、一度のみです。サーバー定義を登録する際には、データ・ソースのロケーションを指定します。また、ニックネームを登録する際には、正確なデータ・ソース・オブジェクトを指定します。

ここでのタスクの順序では、次のタスクは『ODBC データ・ソースのサーバー定義の登録』です。

関連タスク:

- 112 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 WRAPPER ステートメント』
- 111 ページの『CREATE WRAPPER ステートメント - ODBC ラッパーの例』

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* は連合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 ドライバー・マネージャー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* は連合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。

以下の表は、オペレーティング・システムごとの ODBC のラッパー・ライブラリー名のリストです。

表 20. ODBC ラッパーのライブラリー名

連合サーバーの オペレーティング・システム	ODBC ラッパーのライブラリー名
AIX	libdb2rcodbc.a
HP-UX	libdb2rcodbc.sl
Linux	libdb2rcodbc.so

表 20. ODBC ラッパーのライブラリー名 (続き)

Solaris オペレーティング環境	libdb2rcodbc.so
Windows	db2rcodbc.dll

関連タスク:

- 110 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、連合サーバーに 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。

連合データベース内に、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 ステートメントを実行します。

例:

```
CREATE SERVER server_name TYPE odbc
        VERSION 3.0 WRAPPER odbc_wrapper
        OPTIONS (NODE 'node_name')
```

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。

ここでのタスクの順序では、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:

- 114 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』
- 379 ページの『付録 B. 連合システムのサーバー・オプション』
- 113 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 server_name TYPE odbc
  VERSION 3.0 WRAPPER odbc_wrapper
  OPTIONS (NODE 'node_name', DBNAME 'venice')
```

server_name

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 *odbc*

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。ODBC ラッパーの場合、サーバー・タイプは *odbc* でなければなりません。

VERSION 3.0

アクセスしたい ODBC クライアントのバージョン。ODBC 標準バージョン 3 のすべてのリリースがサポートされています。

WRAPPER *odbc_wrapper*

CREATE WRAPPER ステートメントで指定したラッパー名。

NODE '*node_name*'

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 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 および DB2_ID_QUOTE_CHAR の使用方法を示す例です。

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・オプションを提供する必要があります。

- DB2_TABLE_QUOTE_CHAR ' '
- DB2_ID_QUOTE_CHAR ' '

例:

```
CREATE SERVER mysql_server TYPE odbc
      VERSION 3.0 WRAPPER odbc_wrapper
      OPTIONS (NODE 'mysql', DB2_TABLE_QUOTE_CHAR ' ',
              DB2_ID_QUOTE_CHAR ' ')
```

関連タスク:

- 112 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・ソースを連合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、連合サーバーは、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。それぞれの連合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 (ユーザー・マッピング) を定義しなければなりません。連合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、分散要求を送信します。

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 ステートメントを実行します。

例:

```
CREATE USER MAPPING FOR userid SERVER server_name
      OPTIONS (REMOTE_AUTHID 'remote_id', REMOTE_PASSWORD 'remote_password')
```

ここでのタスクの順序では、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:

- 116 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』
- 115 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存する単一の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存する単一の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:

- 114 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・ソースを連合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 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` ステートメントを実行します。

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 連合サーバー上の ODBC ドライバーを調べ、これがインストールされ、ODBC データ・ソース・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。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。

ここでのタスクの順序では、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。

関連概念:

- 10 ページの『サーバー定義およびサーバー・オプション』

関連タスク:

- 「*DB2 Information Integrator* インストール・ガイド」の『連合サーバーのセットアップをチェックする』
- 「*DB2 Information Integrator* インストール・ガイド」の『データ・ソース環境変数をチェックする』
- 117 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、連合サーバーに 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。

ODBC データ・ソースの表とビューのニックネームを作成できることに加えて、リモート・システムの表のニックネームも作成できます。

たとえば、Microsoft SQL Server の *italy* と呼ばれる表 (schema_name 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、連合サーバーから `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せん。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。

連合データベースは、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。連合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・Catalogから読み取られ、連合サーバー上のグローバル・Catalogに入れられます。オプティマイザーはデータ・ソースのCatalog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` ステートメントを実行します。

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Catalog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』
- 119 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のニックネーム。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:

- 117 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 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。たとえば DB2_DJ_COMM プロファイル・レジストリー変数を設定して、連合サーバーが ODBC データ・ソースにアクセスする時のパフォーマンスを改善したい場合があります。

DB2_DJ_COMM 変数の設定によるパフォーマンスの改善

ODBC リモート・サーバーのアクセスに時間がかかる場合、DB2_DJ_COMM プロファイル・レジストリー変数を設定して、パフォーマンスを改善できます。

DB2_DJ_COMM 変数を設定する場合、連合サーバーは、データ・ソースにアクセスしようとするときではなく、初期設定時にラッパーをロードします。

手順:

DB2_DJ_COMM 変数を設定するには、次のようにします。

1. 指定したラッパーに対応するラッパー・ライブラリーに、DB2_DJ_COMM 変数を設定します。

例:

ここで、連合サーバーが Windows NT を使用し、使用するラッパーは ODBC_WRAPPER であるとして、DB2_DJ_COMM 変数を設定するコマンドは次のようになります。

```
db2set DB2_DJ_COMM='db2rcodbc.dll'
```

DB2_DJ_COMM 変数は Windows レジストリーに追加されます。

以下の表には、サポートされているオペレーティング・システムごとに適切な ODBC ライブラリー名がリストされています。

表 21. ODBC ラッパーのライブラリー名

連合サーバーの オペレーティング・システム	ODBC ラッパーのライブラリー名
AIX	libdb2rcodbc.a
HP-UX	libdb2rcodbc.sl
Linux	libdb2rcodbc.so
Solaris オペレーティング環境	libdb2rcodbc.so
Windows NT、Windows 2000、および Windows .NET	db2rcodbc.dll、db2rcodbcF.dll、db2rcodbcU.dll

2. DB2 インスタンスをリサイクルするには、次のコマンドを実行します。


```
db2stop  
db2start
```

DB2 インスタンスをリサイクルすると、変更を DB2 インスタンスが確実に受け入れます。

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。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。したがって、問題が解決したら、トレースはオフにしておいてください。

Windows 連合サーバーでは、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 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。

関連タスク:

- 107 ページの『連合サーバーへの 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 10 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連合サーバーを構成する方法について説明します。

この章では、Teradata データ・ソースに合わせて構成をセットアップする際に実行する必要がある作業のリスト、発行する必要がある SQL ステートメントの例、および使用できる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。

連合サーバーへの Teradata データ・ソースの追加

Teradata データ・ソースに連合サーバーからアクセスするには、アクセスしたい Teradata データ・ソースとオブジェクトの情報を連合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、DB2 コマンド・センター、または DB2 コマンド行プロセッサを使用して、Teradata データ・ソースへのアクセスを構成することができます。

DB2 コントロール・センターを使用する場合の利点は、各ステートメントやコマンドを入力する必要がないことです。DB2 コントロール・センターでは、Teradata データ・ソースへのアクセスを迅速に構成する最も容易な方法が提供されています。ただし、SQL ステートメントを実行するために DB2 コントロール・センターを使用することはできません。

以下のサポートされる任意のオペレーティング・システムで、Teradata ラッパーを連合サーバーに追加できます。

- AIX バージョン 4.3 以上
- Windows NT および Windows 2000

前提条件:

- DB2 コマンド・センターまたは DB2 コマンド行プロセッサへのアクセス。
- Teradata データ・ソースにアクセスするようにセットアップされた連合サーバーおよびデータベース。
- Teradata Call-Level Interface バージョン 2 (CLIV2) リリース 04.06 以上をサポートし、連合サーバー上にインストールされて構成されている、Teradata クライアント・ソフトウェア。
- Teradata リリース V2R3 または V2R4 が実行されている、1 つ以上の Teradata サーバーへのアクセス。

- Teradata データ・ソースにアクセスできるように適切に設定されている、システム環境変数、db2dj.ini 変数、および DB2 Profile Registry (db2set) 変数。
適切にセットアップする必要のある db2dj.ini 変数には、以下のものが含まれます。

- COPLIB および COPERR 変数 (AIX のみ)。

例:

```
COPLIB=coplib_directory  
COPERR=coperr_directory
```

coplib_directory は、libcliv2.so ファイルが存在するディレクトリーの完全修飾パスです。*coperr_directory* は、errmsg.txt ファイルが存在するディレクトリーの完全修飾パスです。

デフォルトでは、インストール・プロセスにより libcliv2.so ファイルおよび errmsg.txt ファイルは同一ディレクトリーに置かれます。ただし、インストール・プロセスの際に、libcliv2.so ファイルおよび errmsg.txt ファイルを別個のディレクトリーに置くよう指定できます。

- Teradata NETRACE および COPANOMLOG 変数 (オプション)。

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。

例:

```
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。

手順:

Teradata データ・ソースを連合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チェックします (AIX)。
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. 連合サーバーから Teradata サーバーへの接続をテストします。
7. 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:

- 1 ページの『データ・ソースの構成の近道』

関連タスク:

- 125 ページの『Teradata サーバーへの接続のテスト』
- 「DB2 Information Integrator インストール・ガイド」の『データ・ソース環境変数をチェックする』

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・ソースを連合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 ステートメントを実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・セッションを開始し、Teradata サーバーにログオンします。
2. SQL コマンドを実行して、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、連合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。

ここでのタスクの順序では、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:

- 126 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・ソースを連合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 上の連合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 コマンドを実行します。

```
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 コマンドを実行してください。

```
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。

ここでのタスクの順序では、次のタスクは『Teradata ラッパーの登録』です。

関連タスク:

- 127 ページの『Teradata ラッパーの登録』

Teradata ラッパーの登録

Teradata ラッパーの登録は、Teradata データ・ソースを連合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。

連合サーバーに Teradata ラッパーを登録して、ラッパー・ライブラリーを識別するには、`CREATE WRAPPER` ステートメントを実行する必要があります。

DB2 Information Integrator には、TERADATA と呼ばれる Teradata ラッパーが組み込まれています。

手順:

Teradata データ・ソースにアクセスするために使用するラッパーを指定するには、`CREATE WRAPPER` ステートメントを実行します。

例:

```
CREATE WRAPPER TERADATA
```

推奨事項: `CREATE WRAPPER` ステートメントを実行する際、デフォルトのラッパー名 `TERADATA` 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、連合サーバーは、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。ラッパー名が連合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、指定した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。

例:

```
CREATE WRAPPER my_wrapper LIBRARY 'library_name'
```

my_wrapper 値はラッパーの名前で、*library_name* 値は使用しているオペレーティング・システムでの Teradata ラッパーのライブラリー名です。

AIX オペレーティング・システムの場合、 Teradata ラッパーのライブラリー名は `libdb2teradata.a` です。

Windows オペレーティング・システムの場合、Teradata ラッパーのライブラリー名は db2teradata.dll です。

ここでのタスクの順序では、次のタスクは『Teradata データ・ソースのサーバー定義の登録』です。

関連タスク:

- 128 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 WRAPPER ステートメント』

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・ソースを連合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。

連合データベース内に、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. ノード名を見つけてみます。
 - 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 ディレクトリー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、COPn 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 aaprwrtr
9.66.111.161  rtpscm11.data.xxx.com accesssm
```

この例では、**nodexyz** フィールドがノード名です。

2. CREATE SERVER ステートメントを実行します。

例:

```
CREATE SERVER server_name TYPE TERADATA VERSION 4 WRAPPER wrapper
OPTIONS (NODE 'node_name')
```

サーバー名を指定する必要があります。指定する名前はユニーク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 および Teradata V2R4 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 1 桁、または小数点のある 2 桁で指定できます。有効なバージョン番号の例には、3、3.0、3.5、4、4.0、4.4 などが含まれます。

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削除したりできます。

ここでのタスクの順序では、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:

- 131 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』
- 130 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 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 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 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。

- 連合サーバーの 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。
- 連合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:

- 128 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・ソースを連合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、連合サーバーは、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。それぞれの連合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 (ユーザー・マッピング) を定義しなければなりません。連合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:

連合ユーザー 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 ステートメントを実行します。

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。

ここでのタスクの順序では、次のタスクは『連合サーバーから Teradata サーバーへの接続のテスト』です。

関連タスク:

- 132 ページの『連合サーバーから 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』
- 132 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、ローカル連合ユーザー 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、ローカル連合ユーザー 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 にマップする連合ユーザー 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:

- 131 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』

連合サーバーから Teradata サーバーへの接続のテスト

連合サーバーから Teradata サーバーへの接続のテストは、Teradata データ・ソースを連合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、連合サーバーから 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 ステートメントを実行します。

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 連合サーバー上の 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 連合変数の設定を調べ、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 ステートメントを実行する場合、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。

ここでのタスクの順序では、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。

関連概念:

- 10 ページの『サーバー定義およびサーバー・オプション』

関連タスク:

- 135 ページの『Teradata 表およびビューのニックネームの登録』

- 「DB2 Information Integrator インストール・ガイド」の『**連合サーバーのセットアップ**をチェックする』
- 「DB2 Information Integrator インストール・ガイド」の『**データ・ソース環境変数**をチェックする』

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』

連合サーバーでの 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。

連合サーバーは、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。次に連合サーバーは、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。

連合データベースは、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。

連合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、連合サーバー上のグローバル・カタログに入られます。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発行できません。

関連タスク:

- 135 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』
- 135 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・ソースを連合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 ステートメントを実行します。

例:

```
CREATE NICKNAME TERANICKNAME FOR TERASERVER."remote_schema"."remote.table"
```

推奨事項: 連合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 ステートメントを実行する際には、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:

- 134 ページの『連合サーバーでの 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』
- 135 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のニックネームです。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:

- 134 ページの『[連合サーバーでの Teradata ニックネーム](#)』

関連タスク:

- 135 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。

DB2_DJ_COMM 変数を設定して Teradata サーバーへのアクセスを改善する

Teradata サーバーにアクセスするのに時間がかかる場合には、DB2_DJ_COMM DB2 プロファイル・レジストリー変数を設定してパフォーマンスを改善できます。

DB2_DJ_COMM 変数を設定する場合、連合サーバーは、データ・ソースにアクセスしようとするときではなく、初期設定時にラッパーをロードします。

手順:

DB2_DJ_COMM 変数を設定するには、次のようにします。

1. 指定したラッパーに対応するラッパー・ライブラリーに、DB2_DJ_COMM 変数を設定します。以下のいずれかのコマンドを使用して、DB2_DJ_COMM DB2 プロファイル・レジストリー変数を設定します。

AIX の場合

```
db2set DB2_DJ_COMM='libdb2teradata.a','libdb2teradataF.a','libdb2teradataU.a'
```

Windows の場合


```
db2set DB2_DJ_COMM='db2teradata.d11','db2teradataF.d11','db2teradataU.d11'
```

2. DB2 インスタンスをリサイクルするには、次のコマンドを実行します。

```
db2stop  
db2start
```

DB2 インスタンスをリサイクルすると、変数の変更を DB2 インスタンスが確実に受け入れます。

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・エントリ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、/sqllib/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 関数が連合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 コマンドを実行します。

```
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:

- 123 ページの『連合サーバーへの Teradata データ・ソースの追加』
- 126 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 11 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連合サーバーを構成する方法について説明します。

この章では、OLE DB データ・ソースに合わせて構成をセットアップする際に実行する必要のある作業のリスト、および発行する必要のある SQL ステートメントの例を示します。

OLE DB データ・ソースを連合サーバーに追加

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 Consumer と OLE DB Provider を定義しています。OLE DB Consumer 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 provider 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 Provider にアクセスできるようになります。

OLE DB ラッパーは、Windows NT または Windows 2000 オペレーティング・システムで稼働する DB2 連合サーバーでサポートされます。

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。

連合サーバーが 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 Provider の情報を連合サーバーに提供します。

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 コマンド行プロセッサを通して構成できます。

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 外部表関数を連合データベースに登録します。

前提条件:

- DB2 コマンド・センターまたは DB2 コマンド行プロセッサへのアクセス。

- OLE DB データ・ソースにアクセスするようにセットアップされた連合サーバーおよびデータベース。
- OLE DB 2.0 以降のドライバーおよび OLE DB Provider が連合サーバー上にインストールされ、構成済みである。

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:

連合サーバーに OLE DB 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. ユーザー・マッピングを作成します。

関連概念:

- 1 ページの『データ・ソースの構成の近道』

関連タスク:

- 「*DB2 Information Integrator* インストール・ガイド」の『連合サーバーのセットアップをチェックする』
- 140 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、連合サーバーに OLE 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ラッパーを登録するには、`CREATE WRAPPER` ステートメントを使用します。

手順:

ラッパーを登録するには、`CREATE WRAPPER` ステートメントを実行します。

例:

```
CREATE WRAPPER OLEDB
```

推奨事項: `CREATE WRAPPER` ステートメントを実行する際には、デフォルトのラッパー名 `OLEDB` 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、連合サーバーは、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。ラッパー名が連合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、指定した名前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。例:

```
CREATE WRAPPER mywrapper LIBRARY 'db2oledb.dll'
```

ここでのタスクの順序では、次のタスクは『OLE DB データ・ソースのサーバー定義の登録』です。

関連タスク:

- 141 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 WRAPPER ステートメント』

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、連合サーバーに 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。

連合データベース内に、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` ステートメントを実行します。

例:

```
CREATE SERVER server_name WRAPPER OLEDB
    OPTIONS (CONNECTSTRING 'Provider=Microsoft.Jet.OLEDB.4.0;
                          Data Source=c:%msdasdk%bin%oledb%wind.mdb')
```

ここでのタスクの順序では、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:

- 143 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』
- 142 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:

- 141 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・ソースを連合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、連合サーバーは、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。それぞれの連合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 (ユーザー・マッピング) を定義しなければなりません。連合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、分散要求を送信します。

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 ステートメントを実行します。

例:

```
CREATE USER MAPPING FOR userid SERVER server_name
    OPTIONS (REMOTE_AUTHID 'remote_id', REMOTE_PASSWORD 'remote_password')
```

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』
- 143 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:

- 143 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 12 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、表構造ファイルとは何か、また表構造ファイルをご使用の連合システムにデータ・ソースとして加える方法などについて解説し、さらに表構造ファイルのラッパーに関連したエラー・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) の固有の 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:

- 145 ページの『表構造ファイルのタイプ』
- 146 ページの『DB2 Information Integrator が表構造ファイルを処理する方法』
- 159 ページの『Documentum とは?』
- 195 ページの『Excel とは?』
- 211 ページの『BLAST とは?』
- 239 ページの『XML とは?』

関連タスク:

- 148 ページの『連合システムへの表構造ファイルの追加』

表構造ファイルのタイプ

表構造ファイルは、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。

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
```

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。

関連概念:

- 145 ページの『表構造ファイルとは?』
- 146 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:

- 148 ページの『連合システムへの表構造ファイルの追加』

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、表構造ファイルにあるデータを照会する 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 147 ページの図 1 に図示されています。

DB2 クライアント 連合データベース

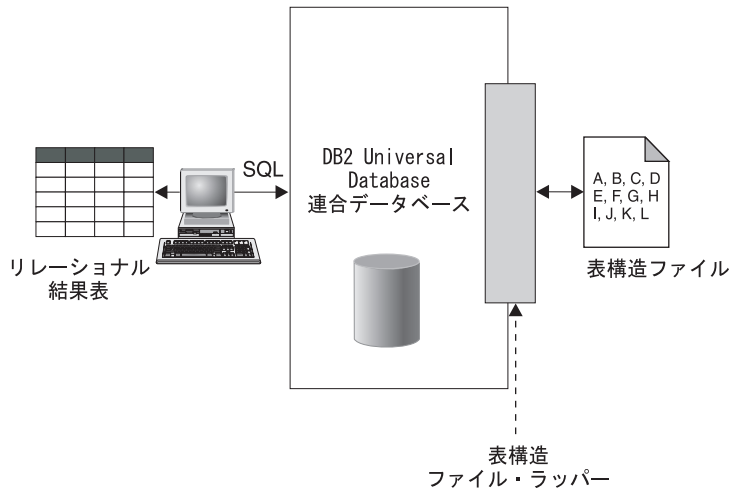


図1. 表構造ファイル・ラッパーがどのように機能するか

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	Manufacturer
234	DrugnameA	Manufacturer1
332	DrugnameB	Manufacturer2
333	DrugnameC	Manufacturer2

関連概念:

- 145 ページの『表構造ファイルとは?』
- 145 ページの『表構造ファイルのタイプ』

関連タスク:

- 148 ページの『連合システムへの表構造ファイルの追加』

連合システムへの表構造ファイルの追加

制約事項:

表構造ファイル・データ・ソースにアクセスするためにパススルー・セッションを使用することはできません。

手順:

表構造ファイルのデータ・ソースを連合サーバーに追加するには、以下を行う必要があります。

1. CREATE WRAPPER コマンドを使用してラッパーを登録します。CREATE WRAPPER コマンドを使用してラッパーを登録します。
2. オプション: 照会のパフォーマンスを向上させるために DB2_DJ_COMM 環境変数を設定します。
3. CREATE SERVER コマンドを使用してサーバーを登録します。
4. すべての表構造ファイルに対して CREATE NICKNAME コマンドを使用してニックネームを登録します。

コマンドは、DB2 コマンド行プロセッサから実行できます。

関連タスク:

- 148 ページの『表構造ファイル・ラッパーの登録』
- 149 ページの『表構造ファイル・ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定』
- 150 ページの『表構造ファイルのサーバーの登録』
- 151 ページの『表構造ファイルのニックネームの登録』

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、表構造ファイルを連合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。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、連合サーバーがデータ・ソースとの通信やデータ・ソースからのデータの検索に使用するメカニズムです。ラッパーは、ライブラリー・ファイルとして、システムにインストールされています。

手順:

ラッパーを登録するためには、CREATE WRAPPER ステートメントを使用して、表構造ファイルにアクセスするために使用するラッパーを指定します。

たとえば、AIX にラッパーを登録するには、以下に示すステートメントを実行します。

```
CREATE WRAPPER laboratory_flat_files LIBRARY 'libdb2lsfile.a'  
  OPTIONS(DB2_FENCED 'N');
```

この例で、laboratory_flat_files はラッパーに付けられた名前です。この名前は、ラッパーが登録されるデータベース内で固有でなければなりません。AIX の表構造ファイル・ラッパーに必要なライブラリー名は libdb2lsfile.a です。

ライブラリーはデフォルトで libdb2lsfile.a としてインストールされますが、インストール時にカスタマイズすることができます。正しい名前については、システム管理者に確認してください。

サポートされているプラットフォーム別の表構造ファイル・ラッパーのデフォルト・ライブラリー・ファイル名の表については、下記の関連タスク・セクションを参照してください。

ここでのタスクの順序では、次のタスクは『表構造ファイル・ラッパーの DB2_DJ_COMM プロファイル変数の設定』です。

関連タスク:

- 149 ページの『表構造ファイル・ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定』
- 「*DB2 Information Integrator* インストール・ガイド」の『非リレーショナル・ラッパー・ライブラリーをチェックする』

関連資料:

- 「*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・ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定

表構造ファイル・ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定は、表構造ファイルを連合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。表構造ファイルにアクセスする場合のパフォーマンスを向上させるために、DB2_DJ_COMM DB2 プロファイル変数を設定することができます。この変数は、初期設定時に連合サーバーがラッパーをロードするかどうかを決定します。

手順:

DB2_DJ_COMM DB2 プロファイル変数を設定するには、関連する CREATE WRAPPER ステートメントに指定したラッパーに対応するラッパー・ライブラリーを使用して db2set コマンドをサブミットします。

例:

```
db2set DB2_DJ_COMM='libdb2lsfile.a'
```

等号 (=) の両側にスペースを入れないようにしてください。

データベース始動時にラッパー・ライブラリーをロードすることに関連したオーバーヘッドがあります。このオーバーヘッドを回避するには、アクセスしたいライブラリーのみを指定します。

ここでのタスクの順序では、次のタスクは『表構造ファイルのサーバーの登録』です。

関連タスク:

- 150 ページの『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

表構造ファイルのサーバーの登録は、表構造ファイルを連合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:

表構造ファイル・サーバーを連合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。

ここでのタスクの順序では、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:

- 151 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、表構造ファイルを連合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。

行の最大の長さは 32768 です。

マルチバイト文字の入ったファイルはサポートされません。

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ファイルごとに `CREATE NICKNAME` を使用します。

ここでの作業は、これで終わりです。

関連タスク:

- 「*DB2 Information Integrator* インストール・ガイド」の『非リレーショナル・ラッパー・ライブラリーをチェックする』

関連資料:

- 358 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』
- 152 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 にできないよう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')
```

このオプションは大文字小文字の区別をします。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:

- 151 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:

- 358 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 (連合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、共通のパスを使用してすべてのノードから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:

- 186 ページの『Documentum ラッパーのアクセス・コントロール』
- 202 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:

- 234 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 22. 表構造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。	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す。(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す。(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。
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「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。

表 22. 表構造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す。(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す。(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 " <code><wrapper_name></code> " は、連合データベースに定義を試みているデータ・ソース (" <code><type></code> " " <code>''</code> ") の " <code>type</code> " 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 " <code><server_name></code> " から予期しないエラー・コード " <code>ERRNO = <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 " <code><server_name></code> " から予期しないエラー・コード " <code>Data 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「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 22. 表構造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 "<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・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 "<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「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、ファイルをソートされていないニックネームとして登録してください。
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「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 "<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「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 "<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・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。

表 22. 表構造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
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。	ニックネームがサポートされていないデータ・タイプに定義されています。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	"<option_name>" は "<object_name>" に対して有効な "<component>" 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実行してください。
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。1 文字にオプションを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。有効な値は "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>" に対して必須 "<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。

関連資料:

- 186 ページの『Documentum ラッパーのメッセージ』
- 203 ページの『Excel ラッパーのメッセージ』

- 234 ページの『BLAST ラッパーのメッセージ』
- 259 ページの『XML ラッパーのメッセージ』

第 13 章 Documentum データ・ソースへのアクセスの構成

この章では、Documentum とは何か、また Documentum データ・ソースをご使用の連合システムに加える方法などについて解説し、さらに Documentum ラッパーに関連したエラー・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 (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) の管理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 連合システムに追加できます。Documentum データ・ソースを連合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、連合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 つのタイプの 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。160 ページの図 2 は、Documentum ラッパーがどのように機能するかを示しています。

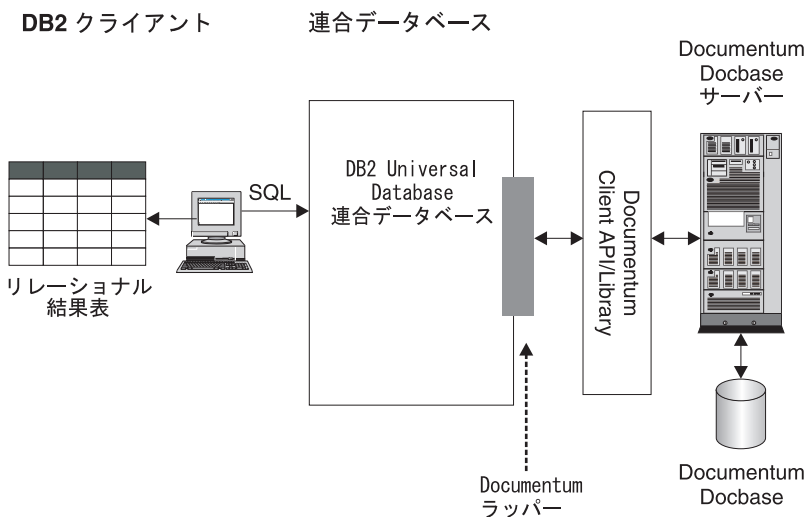


図2. Documentum ラッパーがどのように機能するか

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 23 は、Documentum Docbase のデフォルトの文書タイプである dm_document 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、これらの属性を連合データベース・システム内に接続とします。このデータのサブセットの名前は DrugAB_data としました。

表 23. DrugAB_data

Title (タイトル)	Subject (件名)	Authors (作成者)	Keywords (キーワード)
うさぎに対する Drug A の効果	Drug A	Curran, L.	うさぎ、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 161 ページの表 24 に示します。


```
SELECT title, authors
FROM drugAB_data
WHERE subject = 'Drug A'
```

表 24. 照会の結果

Title (タイトル)	Authors (作成者)
うさぎに対する Drug A の効果	Curran, L.
Drug A の毒性結果	Abelite, P., McMurtrey, K.

関連概念:

- 145 ページの『表構造ファイルとは?』
- 195 ページの『Excel とは?』
- 211 ページの『BLAST とは?』
- 239 ページの『XML とは?』

関連タスク:

- 161 ページの『連合システムへの Documentum の追加』

連合システムへの Documentum の追加

手順:

Documentum データ・ソースを連合サーバーに追加するには、以下のようにします。

1. Documentum クライアント・ライブラリーを DB2 で使用可能にします。
2. Documentum のクライアント dmcl.ini ファイルを指します。
3. CREATE WRAPPER ステートメントを使用してラッパーを登録します。CREATE WRAPPER ステートメントを使用してラッパーを登録します。
4. オプション：照会のパフォーマンスを向上させるために DB2_DJ_COMM 環境変数を設定します。
5. CREATE SERVER ステートメントを使用してサーバーを登録します。CREATE SERVER ステートメントを使用してサーバーを登録します。
6. CREATE USER MAPPING ステートメントを使用してユーザーにデータ・ソースに対するアクセス権を付与します。
7. CREATE NICKNAME ステートメントを使用してニックネームを登録します。CREATE NICKNAME ステートメントを使用してニックネームを登録します。
8. CREATE FUNCTION ステートメントを使用してカスタム関数を作成します。

ステートメントは、DB2 コマンド行プロセッサから実行できます。登録が完了すると、データ・ソースに対して照会ができます。

関連タスク:

- 162 ページの『Documentum クライアント・ライブラリーの DB2 での使用可能化 (AIX および Solaris オペレーティング環境のみ)』
- 163 ページの『Documentum のクライアント dmcl.ini ファイルの位置指定』
- 165 ページの『Documentum ラッパーの登録』
- 165 ページの『Documentum ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定』
- 166 ページの『Documentum データ・ソースのサーバーの登録』
- 167 ページの『ユーザーのマッピング (Documentum ラッパー)』
- 168 ページの『Documentum データ・ソースのニックネームの登録』
- 174 ページの『Documentum データ・ソースのカスタム関数の登録』

Documentum クライアント・ライブラリーの DB2 での使用可能化 (AIX および Solaris オペレーティング環境のみ)

Documentum クライアント・ライブラリーの DB2 での使用可能化 (AIX および Solaris オペレーティング環境のみ) は、Documentum を連合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、クライアント・ライブラリーが DB2 で使用可能で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・バージョン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 DB2 で使用可能にするには、クライアント・ライブラリーを適切なディレクトリーにコピーします。それぞれのサポートされているオペレーティング・システムのクライアント・ライブラリー名およびコピー先ディレクトリーに関しては、163 ページの表 25 を参照してください。また、クライアント・ライブラリーへのシンボリック・リンクを適切なディレクトリーに作成することもできます。

表 25. 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

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	sqllib/lib
Solaris オペレーティング環境	libdmcl.so	sqllib/lib
Windows	dmcl32.dll	x:%sqllib%bin

ここでのタスクの順序では、次のタスクは『Documentum のクライアント dmcl.ini ファイルの位置指定』です。

関連タスク:

- 163 ページの『Documentum のクライアント dmcl.ini ファイルの位置指定』

Documentum のクライアント dmcl.ini ファイルの位置指定

Documentum のクライアント dmcl.ini ファイルの位置指定は、Documentum を連合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。Documentum Docbases へのアクセスは、Documentum クライアントの dmcl.ini ファイルによって制御されます。DB2 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 クライアントの dmcl.ini ファイルに設定する必要があります。

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。

以下に、UNIX オペレーティング・システム上でこうした変数を参照する方法の例を示します。

```
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。

Documentum の dmcl.ini ファイルのロケーションのデフォルト・パスは、/pkgs/documentum です。両方の行が含まれている場合、DMCL_CONFIG が使用され

ます。Windows オペレーティング・システムでは、dmcl.ini ファイルのロケーションを定義するのに、スラッシュではなく円記号 (¥) が使用されます。

AIX および Solaris オペレーティング環境では、db2dj.ini ファイルは \$HOME/sqllib/cfg にあります。

Windows では、db2dj.ini ファイルは x:%sqllib¥cfg にあります (ここで **x:** は sqllib ディレクトリーのあるドライブを表します)。

図 3 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 3. docbroker 名が指定されている dmcl.ini ファイルのサンプル

2. DB2 インスタンスをリサイクルするには、以下のコマンドを出します。

```
db2stop
db2start
```

ここでのタスクの順序では、次のタスクは『Documentum ラッパーの登録』です。

関連タスク:

- 165 ページの『Documentum ラッパーの登録』

Documentum ラッパーの登録

Documentum ラッパーの登録は、Documentum を連合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。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、連合サーバーがデータ・ソースとの通信やデータ・ソースからのデータの検索に使用するメカニズムです。ラッパーは、ライブラリー・ファイルとして、システムにインストールされています。

手順:

Documentum ラッパーを登録するには、CREATE WRAPPER ステートメントをサブミットします。

たとえば、デフォルト・ライブラリー・ファイル libdb21sdctm.a から Dctm_Wrapper という AIX 上の Documentum ラッパーを作成するには、次のステートメントをサブミットします。

```
CREATE WRAPPER Dctm_Wrapper LIBRARY 'libdb21sdctm.a'  
  OPTIONS(DB2_FENCED 'N');
```

サポートされているプラットフォーム別の Documentum ラッパーのデフォルト・ライブラリー・ファイル名の表については、関連タスク・セクションを参照してください。

ここでのタスクの順序では、次のタスクは『Documentum ラッパーの DB2_DJ_COMM プロファイル変数の設定』です。

関連タスク:

- 165 ページの『Documentum ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定』
- 「*DB2 Information Integrator インストール・ガイド*」の『非リレーショナル・ラッパー・ライブラリーをチェックする』

関連資料:

- 「*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 DB2_DJ_COMM DB2 プロファイル変数の設定

Documentum ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定は、Documentum を連合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際のパフォーマンスを向上させるために、DB2_DJ_COMM DB2 プロファイル変数を設定します。この変数は、初期設定時に連合サーバーがラッパーをロードするかどうかを決定します。

手順:

DB2_DJ_COMM DB2 プロファイル変数を設定するには、関連する CREATE WRAPPER ステートメントに指定したラッパーに対応するラッパー・ライブラリーを使用して db2set コマンドをサブミットします。

例:

```
db2set DB2_DJ_COMM='libdb21sdctm.a'
```

等号 (=) の両側にスペースを入れないようにしてください。

データベース始動時にラッパー・ライブラリーをロードすることに関連したオーバーヘッドがあります。このオーバーヘッドを回避するには、アクセスしたいライブラリーのみを指定します。

ここでのタスクの順序では、次のタスクは『Documentum データ・ソースのサーバーの登録』です。

関連タスク:

- 166 ページの『Documentum データ・ソースのサーバーの登録』

Documentum データ・ソースのサーバーの登録

Documentum データ・ソースのサーバーの登録は、Documentum を連合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。

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用する必要があります。

手順:

Documentum サーバーを連合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。

たとえば、関連する CREATE WRAPPER ステートメントで作成された Dctm_Wrapper ラッパーに対して Dctm_Server1 というサーバーがあるとします。また、サーバーに AIX 上で実行する Docbase が含まれており、データの保管に Oracle を使用するとします。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 Dctm_Server1
  TYPE DCTM
  VERSION 3
```

```
WRAPPER Dctm_Wrapper
OPTIONS( NODE 'Dctm_Docbase',
        OS_TYPE 'AIX',
        RDBMS_TYPE 'ORACLE');
```

ここでのタスクの順序では、次のタスクは『ユーザーのマッピング』です。

関連タスク:

- 167 ページの『ユーザーのマッピング (Documentum ラッパー)』

関連資料:

- 「SQL リファレンス 第2巻」の『CREATE SERVER ステートメント』
- 369 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』

ユーザーのマッピング (Documentum ラッパー)

ユーザーのマッピング (Documentum ラッパー) は、Documentum を連合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。以前に定義したサーバーにユーザーをマップして、データ・ソースへのアクセス権をそれらユーザーに付与する必要があります。

手順:

連合サーバーへユーザーをマップするには、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');
```

独自のユーザー・マッピングを定義することもできます。以下の例では、USER は、USER という名前のユーザーではなく、現行のユーザーを意味するキーワードです。

```
CREATE USER MAPPING FOR USER SERVER Dctm_Server1
OPTIONS(REMOTE_AUTHID 'Lisa', REMOTE_PASSWORD 'Lisa_pw')
```

ここでのタスクの順序では、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:

- 168 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』

- 372 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は、Documentum を連合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、ユーザーをそのサーバーへマップした後、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 Documentum データ・ソースを参照する場合に使用されません。

制約事項:

- パススルー・セッションはサポートされていません。
- 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 サーバーから参照されるニックネームは 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。

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て、各オブジェクト・タイプまたは主要な登録済みの表の Docbase ごとにニックネームを作成します。

疑似列とは

CREATE NICKNAME ステートメントは、また 6 つ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 26 にリストされています。

表 26. 疑似列名および定義

疑似列名	定義
GET_FILE	VARCHAR (255)
GET_FILE_DEL	VARCHAR (255)
GET_RENDITION	VARCHAR (255)
GET_RENDITION_DEL	VARCHAR (255)

表 26. 疑似列名および定義 (続き)

疑似列名	定義
HITS	INTEGER
SCORE	DOUBLE

表 27 は、SELECT 文節の疑似列をリストしたものです。

表 27. 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" というストリングです。</p> <p>例:</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" というストリングです。</p>

表 27. 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>例:</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>

171 ページの表 28 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 28. 検索文節を含む、照会においての *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>例:</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>例:</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>

ここでのタスクの順序では、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。

関連概念:

- 182 ページの『Documentum ラッパー用の CreateNicknameFile ユーティリティとは?』

関連タスク:

- 174 ページの『Documentum データ・ソースのカスタム関数の登録』

関連資料:

- 350 ページの『CREATE NICKNAME ステートメント構文 - Documentum ラッパー』
- 172 ページの『CREATE NICKNAME ステートメント - Documentum ラッパーの例』

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 29 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 29. `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 30 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 30. *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 attached_task_status 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。
r_content_size	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。
owner_name	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)。

172 ページの表 29 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 を使用して連合システムを照会することができます。連合システム内で、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。

関連概念:

- 182 ページの『Documentum ラッパー用の CreateNicknameFile ユーティリティとは?』

関連タスク:

- 168 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:

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

Documentum データ・ソース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 を連合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、照会内での完全テキストの検索や文書内容の検索などのいくつかの Documentum 固有の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 176 ページの表 31 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・パーティーの完全テキスト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 DB2 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 DB2 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string引き数の規則

string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stringは単一引用符で囲みます。
- string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string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 31 は、述部のカスタム関数をリストしたものです。

表 31. 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>

表 31. 述部のカスタム関数 (続き)

関数名	説明
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>
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 custom 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>

表 31. 述部のカスタム関数 (続き)

関数名	説明
ANY_LIKE(arg1, arg2)	<p>指定した値と同種の任意の値の反復リストをテストします。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リストをテストします。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>
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>

表 31. 述部のカスタム関数 (続き)

関数名	説明
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 data-bbox="592 392 1072 465">... 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 data-bbox="592 753 1072 805">... 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 data-bbox="592 968 999 1020">... 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 data-bbox="592 1177 1134 1229">... 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 data-bbox="592 1392 1072 1444">... DCTM.FOLDER('/Tools/Drills')=1 OR DCTM.FOLDER_TREE('/Animals/Horses')=1</pre>

表 31. 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>

ここでの作業は、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。

Documentum サーバーと DB2 とでは、LIKE 述部の処理方法が違います。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 '%'
```

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
```

関連資料:

- 199 ページの『Excel データ・ソース - 照会の例』

Documentum ラッパー用の CreateNicknameFile ユーティリティーとは?

無料でダウンロードできる Docbasic ユーティリティー (CreateNicknameFile) を使用して、Docbase オブジェクトまたは登録されている表の完全な定義の入った ASCII ファイルを作成できます。この出力ファイルを編集して、以下のことを行えます。

- 列および属性ごとにカスタム・ローカル名を定義します。ローカルおよびリモート名は、Docbase で認識される最初の名前です。
- 必要のない列および属性を削除します。事前定義されている Documentum 文書タイプ (dm_document) には、EDMS98 で 59 属性、4i で 76 属性しか含めることができません。これらのほとんどは、低レベルの文書管理およびアプリケーション開発のためのメタデータです。それら必要のない属性を削除することによって、パフォーマンスに影響を与えることなく、SELECT * SQL ステートメントを一層効果的なものに行うことができます。
- FOLDERS オプションの値を追加して、特定の Documentum フォルダーへの、このニックネームに対する検索を制限します。
- 必要であれば、DATE から TIMESTAMP へのマッピングを変更します。ユーティリティーは、最も効果的であると考えられるため、DQL DATE から DB2® DATE へのマッピングを生成します。
- アプリケーションに応じて、CHAR から VARCHAR への、またその逆のマッピングを変更します。

Docbase にユーティリティーをインストールして、Documentum Windows® グラフィカル・ユーザー・インターフェースから実行する必要があります。ユーティリティーが生成するファイルは、そのユーティリティーがインストールされている Docbase に固有のものであります。

関連タスク:

- 183 ページの『CreateNicknameFile ユーティリティーのインストール (Documentum ラッパー)』
- 183 ページの『CreateNicknameFile ユーティリティーの構成 (Documentum ラッパー)』
- 184 ページの『DM_ID オブジェクト・タイプの Documentum 登録済み表へのマッピング』

CreateNicknameFile ユーティリティのインストール (Documentum ラッパー)

CreateNicknameFile ユーティリティは、Documentum データ・ソースの CREATE NICKNAME ステートメントを作成する場合の助けとなります。

手順:

ユーティリティをインストールするには、以下の手順に従います。

1. DB2 Information Integrator 製品の Web サイトのダウンロード・セクションから、CreateNicknameFile ユーティリティをダウンロードします。
2. EDMS98 Workspace グラフィカル・ユーザー・インターフェースまたは 4i Desktop Client を使用して、CreateNicknameFile.txt という名前のユーティリティをインポートします。ユーティリティをプロシージャ・タイプとして、Docbase キャビネットまたはフォルダーにインポートし、自由に名前を指定できます。
3. 新しくインポートした CreateNicknameFile.txt オブジェクトのプロパティ・ダイアログで、「ユーザーによって実行可能 (Can be run by user)」ボックスにチェックマークを付けます。

関連概念:

- 182 ページの『Documentum ラッパー用の CreateNicknameFile ユーティリティとは?』

関連タスク:

- 183 ページの『CreateNicknameFile ユーティリティの構成 (Documentum ラッパー)』
- 184 ページの『DM_ID オブジェクト・タイプの Documentum 登録済み表へのマッピング』

CreateNicknameFile ユーティリティの構成 (Documentum ラッパー)

CreateNicknameFile ユーティリティは、Documentum データ・ソースの CREATE NICKNAME ステートメントを作成する場合の助けとなります。

前提条件:

これを構成するには、まず CreateNicknameFile ユーティリティをインストールする必要があります。

手順:

ユーティリティをインストールした後、構成するには以下の手順に従います。

1. ユーティリティのアイコンをダブルクリックして、実行します。

2. Documentum の文書/オブジェクト・タイプ名を入力します。デフォルトは dm_document です。

登録されている表にニックネームを作成する必要がある場合、名前として dm_registered を指定してください。 dm_registered を指定すると、 <owner>.<table_name> の形式で完全修飾表名を入力するように要求されます。表が Docbase の所有者によって所有されている (一般) 場合、所有者名に use dm_dbo を使用します。

ユーティリティーは、登録されている表のニックネームの名前の命名規則を想定しています。その規則では、「登録されている表 (registered table)」であることを示すために、表名の接頭部として "rt_" を付けます。この規則を使用したくない場合、ユーティリティーが設定するニックネームを変更できます。

3. 作成するニックネームに関連したサーバー名を入力します。
4. ニックネームの名前を入力します。

ニックネームの名前は、注釈的で、DB2 インスタンス内で固有のものである必要があります。複数のサーバーに対して同じ <object_type> が定義される必要があるため、ユーティリティーは <server_name>.<object_type> の命名規則を想定します。この規則を使用したくない場合、ユーティリティーが設定するニックネームを変更できます。

5. 出力ファイルの名前を入力します。

デフォルトは C:\Temp\nickname.txt です。出力ファイルを受け取るディレクトリはすでに存在していて、ユーザーによって書き込み可能である必要があります。

表示されるプロンプトに答えると、ニックネーム・ファイルが作成され、テキスト・エディターで表示されます。

関連概念:

- 182 ページの『Documentum ラッパー用の CreateNicknameFile ユーティリティーとは?』

関連タスク:

- 168 ページの『Documentum データ・ソースのニックネームの登録』
- 183 ページの『CreateNicknameFile ユーティリティーのインストール (Documentum ラッパー)』

DM_ID オブジェクト・タイプの Documentum 登録済み表へのマッピング

CreateNicknameFile ユーティリティーによって作成される列定義は、Documentum ラッパーの要件 (各データ・タイプを対応する DB2 データ・タイプの正しいマッピングも含む) に準拠しています。唯一の例外は、Documentum が登録済み表の DM_ID データ・タイプをサポートしない点です。ユーティリティーは、登録済み表の列が 16 文字長のストリングおよび「_id」で終了する名前のストリングとして定義される場合に、そ

の列がオブジェクト ID を含むために使用されると想定します。DM_ID データ・タイプの場合、ユーティリティーは列を DB2 CHAR(16) データ・タイプにマップします。それ以外のすべての場合、すべての string/varchar 列は DB2 VARCHAR データ・タイプにマップされます。

手順:

正しいデータ・タイプのマッピングを確実なものにするためには、以下のように行います。

1. CreateNicknameFile ユーティリティーによって作成された出力ファイルの中の列データ・タイプ定義を調べます。
2. ユーティリティーが Documentum 列のデータ・タイプを間違った DB2 データ・タイプにマップした場合は、そのファイルを使用してニックネームを DB2 に登録する前に DB2 データ・タイプを変更します。

関連概念:

- 182 ページの『Documentum ラッパー用の CreateNicknameFile ユーティリティーとは?』

関連タスク:

- 183 ページの『CreateNicknameFile ユーティリティーのインストール (Documentum ラッパー)』
- 183 ページの『CreateNicknameFile ユーティリティーの構成 (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、データ・タイプが VARCHAR ではない、同じリモートの繰り返し属性のための 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。

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、照会の結果に含められます。

関連資料:

- 153 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』
- 202 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 32. `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せん。(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。

表 32. 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せん。(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せん。(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。
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せん。(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せん。(理由「予期しない列の数が要求された」。)	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。

表 32. 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せん。(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せん。(理由「サポートされていない列の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せん。(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せん。(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。
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「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。

表 32. 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せん。(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せん。(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せん。(理由「不整合な列の数」。)	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せん。(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せん。(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。

表 32. 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せん。(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。
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「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せん。(理由「開始トランザクションが失敗」。)	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せん。(理由「入力パラメーターが NULL でなかった」。)	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。
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せん。(理由「Dctm 関数は DCTM.function(...) =1 のようであればならない」。)	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。

表 32. 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せん。(理由「無効な列数が要求された」。)	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 " <code><column-name></code> " の有効な "COLUMN" オプションではありません。	DELIMITER オプションが列 <code><column-name></code> 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、" <code><server-name></code> " の " <code><option-value></code> " に設定することはできません。	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、" <code><server-name></code> " の " <code><option-value></code> " に設定することはできません。	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。
SQL1882N	"NICKNAME" オプション "IS_REG_TABLE" は、" <code><nickname></code> " の " <code><option-value></code> " に設定することはできません。	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、" <code><nickname></code> " の " <code><option-value></code> " に設定することはできません。	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、" <code><server-name></code> " の " <code><option-value></code> " に設定することはできません。	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。

表 32. 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 "FOLDERS" は、"<nickname>" の "<option-value>" に設定することはできません。	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>" の "<option-value>" に設定することはできません。	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'。さらに、VERSIONS 'Y' は、IS_REG_TABLE が 'Y' の表に指定することはできません。
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 = 「db2dj.ini に DOCUMENTUM または DMCL_CONFIG env var が欠落」	必須の環境変数が設定されていません。それらを db2dj.ini ファイルに設定してください。
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。

表 32. 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
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。

関連資料:

- 154 ページの『表構造ファイル・ラッパーのメッセージ』
- 203 ページの『Excel ラッパーのメッセージ』
- 234 ページの『BLAST ラッパーのメッセージ』
- 259 ページの『XML ラッパーのメッセージ』

第 14 章 Excel データ・ソースへのアクセスの構成

この章では、Excel とは何か、また Excel データ・ソースをご使用の連合システムに加える方法などについて解説し、さらに Excel ラッパーに関連したエラー・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® (MS) 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、および Excel 2000 のスプレッドシートをサポートします。図 4 は、Excel ラッパーがスプレッドシートと連合システムを接続する方法を示しています。

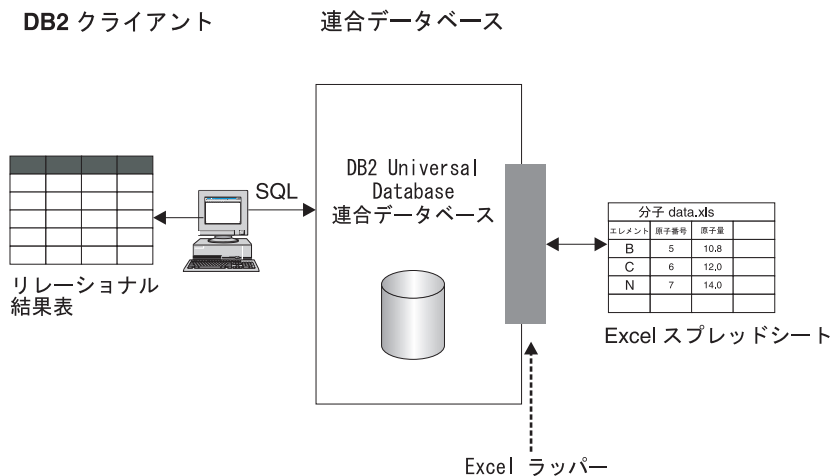


図 4. Excel ラッパーがどのように機能するか

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 連合システム内の列にマップします。表 33 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 33. 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_A	1.23	367	テスト済み

表 33. *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_G		210	
3	compound_F	0.000425536	174	テスト済み
4	compound_Y	1.00256		テスト済み
5	compound_Q		1024	
6	compound_B	33.5362		
7	compound_S	0.96723	67	テスト済み
8				
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、分子の数が 100 より大きいすべての複合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 34 に示されています。

表 34. 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:

- 145 ページの『表構造ファイルとは?』
- 159 ページの『Documentum とは?』
- 211 ページの『BLAST とは?』
- 239 ページの『XML とは?』

関連タスク:

- 197 ページの『連合システムへの Excel の追加』

連合システムへの Excel の追加

手順:

Excel データ・ソースを連合システムに追加するには、以下のようにします。

1. CREATE WRAPPER ステートメントを使用してラッパーを登録します。CREATE WRAPPER ステートメントを使用してラッパーを登録します。
2. CREATE SERVER ステートメントを使用してサーバーを登録します。CREATE SERVER ステートメントを使用してサーバーを登録します。
3. アクセスしたい Excel スプレッドシートごとに、CREATE NICKNAME ステートメントを使用してニックネームを登録します。

コマンドは、DB2 コマンド行プロセッサから実行できます。

関連タスク:

- 197 ページの『Excel ラッパーの登録』
- 198 ページの『Excel データ・ソースのサーバーの登録』
- 199 ページの『Excel データ・ソースのニックネームの登録』

Excel ラッパーの登録

Excel ラッパーの登録は、Excel を連合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。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、連合サーバーがデータ・ソースとの通信やデータ・ソースからのデータの検索に使用するメカニズムです。ラッパーは、ライブラリー・ファイルとして、システムにインストールされています。

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。
- MS 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:

Excel データ・ソース・ラッパーを登録するには、CREATE WRAPPER ステートメントをサブミットします。

ライブラリー・ファイル db21sx1s.dll を使用して、Excel_9x_Wrapper という Excel 97 用の Excel ラッパーを作成するには、以下のステートメントをサブミットします。

```
CREATE WRAPPER Excel_9x_Wrapper LIBRARY 'db21sx1s.dll'  
  OPTIONS(DB2_FENCED 'N');
```

ここでのタスクの順序では、次のタスクは『Excel データ・ソースのサーバーの登録』です。

関連タスク:

- 198 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 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 を連合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:

Excel サーバーを連合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。

たとえば、CREATE WRAPPER ステートメントを使用して作成した Excel_2000_Wrapper ラッパーのサーバーを登録する biochem_node1 というノード名の biochem_lab というサーバーを作成するには、以下のステートメントをサブミットします。

```
CREATE SERVER biochem_lab WRAPPER Excel_2000_Wrapper;
```

ここでのタスクの順序では、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:

- 199 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』
- 370 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 を連合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。

制約事項:

ラッパーがサポートする DATE データ・タイプの日付範囲は、1970 年 1 月 1 日から 2038 年 1 月 18 日までです。

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、連合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:E5');
```

ここでの作業は、これで終わりです。

関連資料:

- 353 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。

以下の照会は、分子の質量が 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、複合名または分子の質量が 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、複合名にストリング ase が含まれており、分子の質量が 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:

- 181 ページの『Documentum データ・ソース - 照会の例』
- 200 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 スプレッド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、それらはスプレッド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 コマンド行プロセッサを使用して入力されます。ラッパーが登録された後、スプレッドシートに対して照会ができます。

シナリオは、Compound_Master.xls という複合スプレッド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 35 に示されています。

表 35. サンプル・スプレッドシート *Compound_Master.xls*

	A	B	C	D
1	compound_A	1.23	367	テスト済み
2	compound_G		210	
3	compound_F	0.000425536	174	テスト済み
4	compound_Y	1.00256		テスト済み
5	compound_Q		1024	

表 35. サンプル・スプレッドシート *Compound_Master.xls* (続き)

	A	B	C	D
6	compound_B	33.5362		
7	compound_S	0.96723	67	テスト済み
8				
9	compound_O	1.2		テスト済み

手順:

Excel ラッパーを使用して Excel 2000 スプレッド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'
      OPTIONS(DB2_FENCED 'N')
```

2. サーバーを登録します。

```
db2 => CREATE SERVER biochem_lab WRAPPER Excel_2000
```

3. Excel スプレッド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・ソースは連合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: "Give me all the compound data where mol_count is greater than 100"

```
SELECT * FROM compound_master WHERE mol_count > 100
```

結果: All fields for rows 1, 2, 3, 5, and 7.

- サンプル SQL 照会: "Give me the compound_name and mol_count for all compounds where the mol_count has not yet been determined."

```
SELECT compound_name, mol_count FROM compound_master
WHERE mol_count IS NULL
```

結果: Fields compound_name & mol_count of rows 4, 6, and 9 from the spreadsheet.

- サンプル SQL 照会: "Count the number of compounds that have not been tested and the weight is greater than 1."

```
SELECT count(*) FROM compound_master
WHERE was_tested IS NULL AND weight > 1
```

結果: The record count of 1 which represents the single row 6 from the spreadsheet that meets the criteria.

- サンプル SQL 照会: "Give me the compound_name and mol_count for all compounds where the mol_count has been determined and is less than the average mol_count"

```
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 36 を戻します。

表 36. 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:

- 197 ページの『[連合システムへの Excel の追加](#)』

関連資料:

- 199 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:

- 153 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](#)』
- 186 ページの『[Documentum ラッパーのアクセス・コントロール](#)』

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 37. 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、連合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。

表 37. 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 "-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。
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。

表 37. 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 "-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。
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。

表 37. 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 "-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。

表 37. 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 "-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> オプションは除去できない」です。	指定したオプションは必須オプションのため、除去できません。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-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。

表 37. 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 "-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。
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。

表 37. 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 "-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 出カ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 照会のとき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。
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。

関連資料:

- 154 ページの『表構造ファイル・ラッパーのメッセージ』
- 186 ページの『Documentum ラッパーのメッセージ』
- 234 ページの『BLAST ラッパーのメッセージ』
- 259 ページの『XML ラッパーのメッセージ』

第 15 章 BLAST データ・ソースへのアクセスの構成

この章では、BLAST とは何か、また BLAST データ・ソースをご使用の連合システムに加える方法などについて解説し、さらに BLAST ラッパーに関連したエラー・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 38 に示されています。

表 38. 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リージョンと同種の領域をもつシーケンス検出します。
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に変換されます。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 38. 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。

213 ページの図 5 は BLAST が実際の連合システムでどう機能するかを示しています。

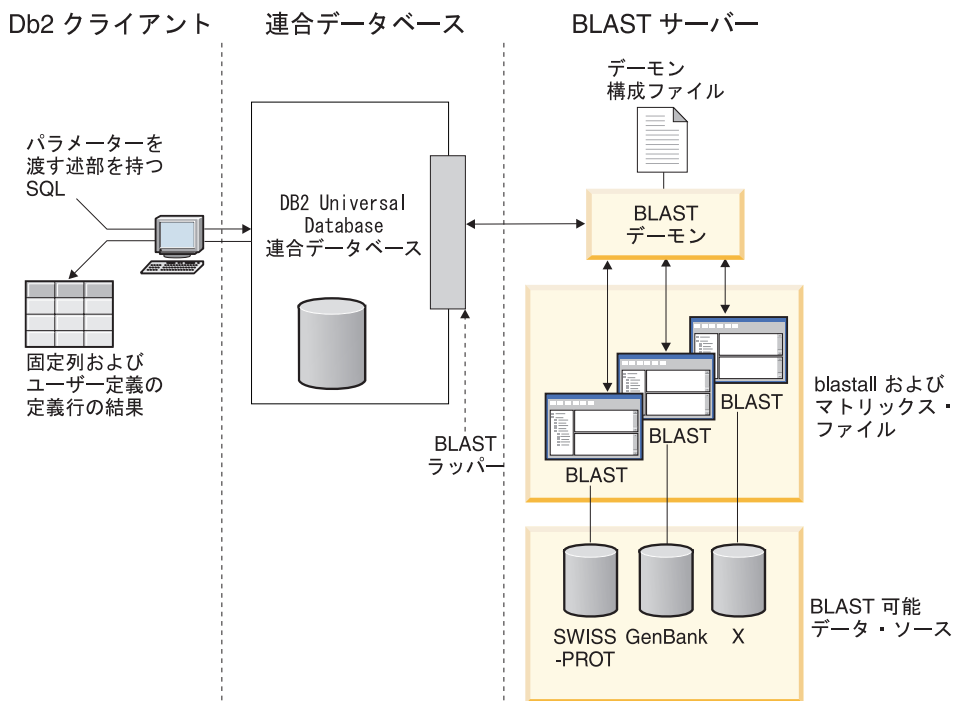


図5. BLAST ラッパーがどのように機能するか

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 特有のパラメーターを渡す述部を持つ 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 連合データベース・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。このサーバーは連合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、連合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。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、および

- ユーザー構成の定義情報。

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、そして実際の連合システムの中の他のデータ・ソースと結合可能なリレーショナル表へと移動します。

図 6 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
GTTCTTCCAGTGCCCAAGTCCATTCTGACATCAATGAAGAAGGTAAATCCCTGCGTGATCCCTCTGCC
AAGATGTCGAAATCAGACCCGGATAAACTAGCTGCTGTCAGAATAACAGACAGCCGGAGGAGATCGTGC
AGAAGTCCCGAAGGCTGTGACGGACTTACCTCGGAGGTCACCTACGACCCGGCCAGGCGAGGAGGCGT
GTCCAACCTTGGTGGCCATCCACGCGGCAGTGACCGGACTCCCGGTGGAGGAGGTGGTCCGCCGAAGTGCT
GGCATCAACACCCTGGCTACAAGTTGGTGGTGGCGGAGGCTGTGATTGAGAGATTTGCACCAATTAAGA
GTGAAATTGAAAACTGAAGAGGAACAAGGACCACCTAGAGAAGGTTTTACAAGTTGGTCCGCAAAAGC
CAAAGAATTAGCATATCCCGTGTGCCAGGAGGTGAAGAAATTGGTGGGGTTTCTATAGGCAGTCTCACCT
AGTCCCAGAAAATGTTTTTATCTTGTGGTCTGCTTGCACACTCAGTCTAATAAAGGCAGCTTTCCTAAG
ACGCCAACAAATCCAGTTTGGGGATGCTTAGTTACT
>8:9747 PMON5699
AAGAAGTCTTGTAGAACTTCCACCTCCGGCTTCCCTCCACCTCTCTTACTGTCCCAACCTTCTGAG
ACGCTTTTTCTCTCCCGAGGATTTATCTTTCTCTCTCTCTCTCTCTCTTTTTTTTTTTTCCCT
TTTTCCCCCAGGCTGGTTTTGCTTTGGGGAGGGGGGTTTTTTAAAGGGGCCGGGGGGCCCTTT
CTCCCCCTAATGGGGTAAATTAATAATGGGGGGGGGGTTTTTTTTTTTTAAACCCCTATTTGGTCCGG
CCCGGGGATTTCCCCCCCCCTTGCCTCCGGGTTCCGGGGCCGGAGGAGGGGGAAAAAGGGCGGGAA
CCTTTGGTAGTTTTCCCTCGGAAAAAATTTTTCGGGGGGAAAAACCTCCCT
>13:6512 PMON5498
GATAAGAGGCAGAATAGAAGACTGGACTACTTCTCTCTAAAAACACATTTAAACTAAGCCTGAGCAAT
CTCCACCCAAATGGACCGGAAACCTTAAAAAAGAATCCTACTCTGAAAGAAAAAGAGGAGGACACATCAA
GAGGTAGAAGGGGCGATTTTCATGATATAAAACAACCCATACCTCCAGAGTGGGAAGCTCCACAGACTGAA
AACTAACTGGTTACAGAACTCACCTACAGGAGTGAGCCACATCAAACCTCGAATGTGGGGATCTG
GCACTGGTAGAAAGAGCCCTGGAGCATCTGGCATTGAAGGCCAGTGGGGCTTGTGTGCAGGAGATCCAC
AGGACTAGGGGAAACGGAGACCCCATCTTAAAAGGTGCACACAGACTTTTACGTGCACTGGGTCCCAG
TGCAAAGCAAAGTCTCCATAGGAATCTGGGTCAAACCTGACTGCAGTTCTTGGAGGACCTCCTGGGAAAG
CAAGGGTGAATGTGGCTTCTTGTGGGAAAGGACATTGGAAGCAAAGCTCTTGGGAATATTATCAGTGT
GC
>15:8924 PMON5426
GGAGAACTGACTCCTGAGCAGCTGCAATTCATGCGGCAGGTGCAGCTCGCCAGTGGCAGAAGACGCTG
CCACAGCGCGGACCCGGAACATCGTGACCGGCCTGGGCATCGGGCGCTGGTGTGGCAATTTGTATCC
GTTTGGACTGTAGACTCAGGGAGACCGCATTTAGGGGAACAGGAAGGGCAGCAGGGGCGTGTAGGAGGGC
AGTGTGGGGGTGGTAGAAGGAGCCGAGATATGAAAACCTTGGCTCCTTTAACTCTGAATCAAGCGTTT
GGGTACCTTACGTTGCTATTTAAAGGTGATTTTAGTATAATTGATTAATGATTACGGAGTCCGGTGA
GGGCTCCCAGGAGCAGACGGCAGAAGATCGAATTTGGGAGGATGATCAGCAGCGGTGGTTGAGCAAGTGT
GGGAAAAAGGAAATGCCACATTCACGTTGTTTCTGAACCCACCTCCCAGATGGTTACACCTTCTACT
CGGTGCCAGGAGCGTTTCTTGGATGAGCTGGAGGATGAGGCCAAAGCTGCTC
```

図 6. fasta ファイルのサンプル、*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、登録されている連合システムを通じて、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 = 'ACATTCTTATAGAGTATTGCTACTCCTCCAGGATAGAGTCATCTCT
GGTCTCCAGAGCCACCGCTGGCTACAAGTTGGTGGTGGCGGAGGCTGTGATTGAGAGATTGG
CACCAATACAGAAACTCACCTACAGGAGTGAGCGGGTGGTAGAAGGAGCCCGAGATATGAAA
ACCTGTTTCAAGACCCATTGTCACCGGGG';
```

照会の結果は、BLAST ラッパーによって表 39 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 39. BLAST は、実際の連合システムに組み込まれているときは、結果をリレーショナル表フォームで戻します。

固有の 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・ソースから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、連合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ったような洞察を得ることができます。

関連概念:

- 145 ページの『表構造ファイルとは?』
- 159 ページの『Documentum とは?』
- 195 ページの『Excel とは?』

- 239 ページの『XML とは?』

関連タスク:

- 216 ページの『連合システムへの BLAST の追加』

連合システムへの BLAST の追加

手順:

BLAST データ・ソースを連合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。BLAST デーモンを構成します。
3. BLAST デーモンを開始します。BLAST デーモンを開始します。
4. CREATE WRAPPER ステートメントを使用してラッパーを登録します。CREATE WRAPPER ステートメントを使用してラッパーを登録します。
5. オプション: 照会のパフォーマンスを向上させるために DB2_DJ_COMM 環境変数を設定します。
6. CREATE SERVER ステートメントを使用してサーバーを登録します。
7. CREATE NICKNAME ステートメントを使用してニックネームを登録します。

ステートメントは、DB2 コマンド行プロセッサから実行できます。BLAST ラッパーが連合システムに追加されたら、BLAST データ・ソースに対して照会ができます。

関連タスク:

- 217 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』
- 217 ページの『BLAST デーモンの構成』
- 220 ページの『BLAST デーモンの開始』
- 221 ページの『BLAST ラッパーの登録』
- 222 ページの『BLAST ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定』
- 223 ページの『BLAST データ・ソースのサーバーの登録』
- 223 ページの『BLAST データ・ソースのニックネームの登録』

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 を連合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。インストールされていない場合は、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 検索を実行し、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。

ここでのタスクの順序では、次のタスクは『BLAST デーモンの構成』です。

関連タスク:

- 217 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 を連合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 ラッパーは、DB2 Universal Database 連合システムから TCP/IP でアクセス可能な UNIX ベースのマシン上で、BLAST デーモンが稼働していることを必要とします。デーモンは、ラッパーと DB2 Universal Database とは別に実行され、ラッパーからの BLAST ジョブ要求を listen します。デーモン実行可能ファイル db2blast_daemon は、BLAST サーバー・マシンのどのディレクトリーにあってもかまいません。

DB2 Universal Database インストールの際、デーモン実行可能ファイルは、連合サーバーがインストールされるマシンで、AIX の場合は /usr/opt/db2_08_01/bin ディレクトリーに、その他の UNIX プラットフォームの場合は /opt/IBM/db2/V8.1/bin ディレク

トリーに入れられます。ご使用の環境で、BLAST が別のマシンで実行されている場合は、そのマシン上の選択した場所にデーモンをコピーする必要があります。

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。

BLAST デーモンは構成ファイルを必要とします。BLAST_DAEMON.config という名前のサンプル・デーモン構成ファイルが、ディレクトリー DB2PATH/samples/lifesci にあります。ここで、DB2PATH は DB2 Universal Database がインストールされているディレクトリーです。BLAST_DAEMON.config は、そのファイルのデフォルトの名前です。

構成ファイルをデーモンにアクセス可能な任意の場所へコピーします。必要であれば名前を変更します。そして、データ・ソースを処理するために編集します。デフォルトにより、blast_daemon は、開始された作業ディレクトリーの中での blast_daemon の構成情報を探します。

手順:

デーモンを構成するには、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、ネットワーク接続経由でラッパーへ戻されます。この時点で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マシン上の 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。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構成ファイルの CREATE NICKNAME ステートメントの DATASOURCE オプションで使用する各項目名を指定する必要があります。

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。

```
entry_name = path to BLAST-able_data_source
```

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
```

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ファイル名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・ファイル名をもっていなければなりません。

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。そのようにしないと、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。備えられている構成ファイルのサンプルでは、最終行は改行で終了していません。正常に実行するには、最終行を改行で終了することが必要となります。

例:

以下の例は、必須オプションおよび GenBank と SWISS-PROT についての BLAST 可能データ・ソース指定のついたサンプル構成ファイルの内容を示しています。

```
=
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
```

ここでのタスクの順序では、次のタスクは『BLAST デーモンの開始』です。

関連タスク:

- 220 ページの『BLAST デーモンの開始』

関連資料:

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

BLAST デーモンの開始

BLAST デーモンの開始は、BLAST を連合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 デーモンが実行されてい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:

デーモン・インストール・ディレクトリーにあり、デーモン構成ファイルの名前を変更せず、また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、BLAST デーモンを開始するには、コマンド行に以下のコマンドを入力します。

```
db2blast_daemon
```

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。

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、デーモン構成ファイルが配置されているディレクトリーにない場合、BLAST デーモンを開始するには、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・デーモンがその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
```

ここでのタスクの順序では、次のタスクは『BLAST ラッパーの登録』です。

関連タスク:

- 221 ページの『BLAST ラッパーの登録』

BLAST ラッパーの登録

BLAST ラッパーの登録は、BLAST を連合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。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、連合サーバーがデータ・ソースとの通信やデータ・ソースからのデータの検索に使用するメカニズムです。ラッパーは、ライブラリー・ファイルとして、システムにインストールされています。

手順:

BLAST ラッパーを登録するには、`CREATE WRAPPER` ステートメントをサブミットします。

たとえば、デフォルト・ライブラリー・ファイル `libdb2lsblast.a` から `my_blast` という AIX 上の BLAST ラッパーを作成するには、次のステートメントをサブミットします。

```
CREATE WRAPPER my_blast LIBRARY 'libdb21sblast.a'  
  OPTIONS(DB2_FENCED 'N');
```

サポートされているプラットフォーム別の BLAST ラッパーのデフォルト・ライブラリー・ファイル名の表については、関連タスク・セクションを参照してください。

ここでのタスクの順序では、次のタスクは『BLAST ラッパーの DB2_DJ_COMM プロファイル変数の設定』です。

関連タスク:

- 222 ページの『BLAST ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定』
- 「*DB2 Information Integrator* インストール・ガイド」の『非リレーショナル・ラッパー・ライブラリーをチェックする』

関連資料:

- 「*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 DB2_DJ_COMM DB2 プロファイル変数の設定

BLAST ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定は、BLAST を連合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・ソースにアクセスする際のパフォーマンスを向上させるために、DB2_DJ_COMM DB2 プロファイル変数を設定します。この変数は、初期設定時に連合サーバーがラッパーをロードするかどうかを決定します。

手順:

DB2_DJ_COMM DB2 プロファイル変数を設定するには、関連する CREATE WRAPPER ステートメントに指定したラッパーに対応するラッパー・ライブラリーを使用して db2set コマンドをサブミットします。

例:

```
db2set DB2_DJ_COMM='libdb21sblast.a'
```

等号 (=) の両側にスペースを入れないようにしてください。

データベース始動時にラッパー・ライブラリーをロードすることに関連したオーバーヘッドがあります。このオーバーヘッドを回避するには、アクセスしたいライブラリーのみを指定します。

ここでのタスクの順序では、次のタスクは『BLAST データ・ソースのサーバーの登録』です。

関連タスク:

- 223 ページの『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

BLAST データ・ソースのサーバーの登録は、BLAST を連合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 サーバーを連合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')
```

ここでのタスクの順序では、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:

- 223 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 SERVER ステートメント』
- 368 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は、BLAST を連合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。

ニックネームは、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。定義行構文解析の詳細については、『定義行構文解析』を参照してください。固定列の詳細については、『固定列』を参照してください。

定義行構文解析

定義行 (define と呼ばれます) 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。

定義行情報をユーザーの結果表に入れたい場合は、CREATE NICKNAME ステートメントに定義行列を指定する必要があります。各列の指定は、INDEX オプションを指定する必要があります。各列に対して DELIMITER オプションを指定する必要があります。ただし、その列に定義行の残りを入れたい場合に指定した最後の列を除きます。

定義行フィールドは、integer、float、double、または varchar のタイプのものでなければなりません。

BLAST ヒットの Accession 番号フィールドの中にデータが検出された場合、これらのデータは、その BLAST ヒットの Definition フィールドの中のデータの前に挿入されます。Accession 番号データとそれに続く Definition フィールド・データを含んでいる結果の定義行は、ラッパーによって構文解析されます。

固定列

CREATE NICKNAME ステートメントは、自動的に固定列を作成します。固定列は、CREATE NICKNAME ステートメントの中には表れません。しかし、ニックネーム定義の一部であり、SQL 照会で参照できます。固定列には入力と出力の 2 つのタイプがあります。

入力固定列

入力固定列は、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。入力固定列は、照会選択リストの中で参照することもできます。そして、結果表の一部として戻すことができます。入力固定列は、225 ページの表 40 にリストされています。

表 40. 入力固定列

名前	データ・タイプ	使用できる演算子	説明
BlastSeq	varchar(32000)	=	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と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。
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 検索タイプだけが、この述部を使用します。

表 40. 入力固定列 (続き)

名前	データ・タイプ	使用できる演算子	説明
NMatchReward	integer	=	同種領域の中の一一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。

表 40. 入力固定列 (続き)

名前	データ・タイプ	使用できる演算子	説明
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。
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。

各入力固定列に対するサポートされている BLAST 検索タイプとスイッチについては、表 41 にリストされています。

表 41. 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	Y	N/A
E_Value	n、p、x、tn、tx	-e	N	10
QueryStrands	n	S	N	3
GapAlign	n、p、x、tn、tx	-g	N	T
Matrix	p、x、tn、tx	-n	N	BLOSUM62
NMismatchPenalty	n	-q	N	-3
NMatchReward	n	-r	N	1
FilterSequence	n、p、x、tn、tx	-F	N	T
NumberOfAlignments	n、p、x、tn、tx	-b	N	250
GapCost	n、p、x、tn、tx	-G	N	11
ExtendedGapCost	n、p、x、tn、tx	-E	N	1

表 41. 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 (続き)

名前	BLAST 検索タイプ	BLAST スイッチ	必須 ?	デフォルト
WordSize (Blastn の場合、7 より小さい値は無効)	n、p、x、tn、tx	-W	N	11 -BLASTn
				3 -BLASTp
ThresholdEx	n、p、x、tn、tx	-f	N	0

出力固定列

出力固定列は、照会結果表の中に戻され、述部として使用できます。出力固定列は表 42 にリストされています。

表 42. 出力固定列

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。
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。
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・シーケンスの間に一致した、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。

表 42. 出力固定列 (続き)

名前	データ・タイプ	説明
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。
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。
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。
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。

ここでの作業は、これで終わりです。

関連資料:

- 348 ページの『CREATE NICKNAME ステートメント構文 - BLAST ラッパー』
- 230 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` を使用して連合システムで照会することができます。連合システムで、`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:

- 223 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:

- 348 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。プ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 NT/2000 オペレーティング・システムをサポートしています。BLAST デーモンは、Windows NT/2000 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 NT/2000 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 デーモンのあるマシン上に 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、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. プラスト可能データベース仕様エントリに、TurboBlast にプラスト可能データベース名をアップロードするのに使用したプラスト可能データベース名を指定します。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。プラスト可能データ・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。プラスト・デーモンは、`blastall` ではなく `tblastall` を呼び出し、プ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プラスト・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。

BLAST SQL 照会の作成

BLAST データ・ソースの SQL には、標準の BLAST スイッチを blastall 実行可能ファイルへ渡すために使用される特別な入力述部のみが含まれていなければなりません。

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的です。

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:

- 223 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:

- 232 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るために、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 tblx.tblx_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:

- 181 ページの『Documentum データ・ソース - 照会の例』
- 199 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

同一サーバー上でラッパーとデーモンの両方を実行すると、ネットワーク通信のボトルネックの可能性を除去することができます。

関連資料:

- 153 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』

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 43. 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せん。(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、連合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、連合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。
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。

表 43. 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「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>" の有効な "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 43. 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>" の "<option-value>" に設定するこ とはできません。	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。

関連資料:

- 154 ページの『表構造ファイル・ラッパーのメッセージ』
- 186 ページの『Documentum ラッパーのメッセージ』
- 203 ページの『Excel ラッパーのメッセージ』
- 259 ページの『XML ラッパーのメッセージ』

第 16 章 XML データ・ソースへのアクセスの構成

この章では、XML とは何か、また XML データ・ソースをご使用の連合システムに加える方法などについて解説し、さらに XML ラッパーに関連したエラー・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 '>') の括弧に入れた語) を使用します。サンプル XML 文書が 図 7 に示されています。

```
<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 7. 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

図 8 に、XML ラッパーによる連合システムの処理方法を示します。

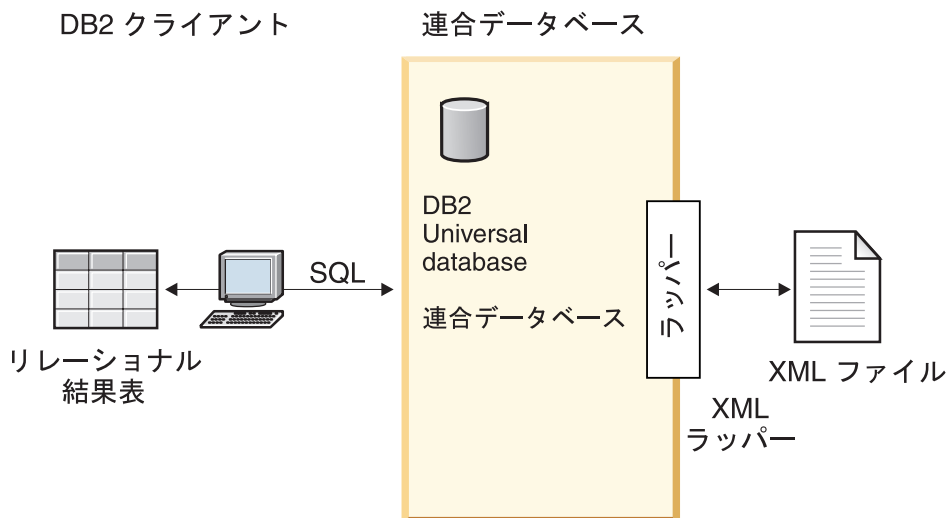


図 8. XML ラッパーがどのように機能するか

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、239 ページの図 7 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 文書がいったん連合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 9 に示されています。

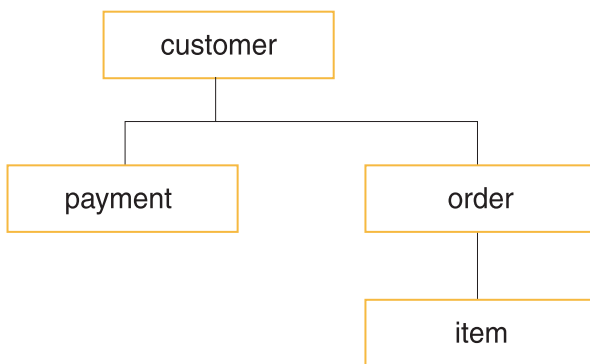


図 9. サンプル XML 文書のツリー構造

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 //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:

- 145 ページの『表構造ファイルとは?』
- 159 ページの『Documentum とは?』
- 195 ページの『Excel とは?』
- 211 ページの『BLAST とは?』
- 246 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:

- 243 ページの『[連合システムへの XML の追加](#)』

連合システムへの XML の追加

XML ラッパーを登録すると、連合サーバーで XML データ・ソースを使用できます。XML ラッパーを登録してから、対応するサーバーおよびニックネームを登録して、連合サーバーが XML データを検索して処理できるようにします。

手順:

XML データ・ソースを連合サーバーに追加するには、以下のようにします。

1. CREATE WRAPPER ステートメントを使用してラッパーを登録します。CREATE WRAPPER ステートメントを使用してラッパーを登録します。
2. オプション: 照会のパフォーマンスを向上させるために DB2_DJ_COMM 環境変数を設定します。
3. CREATE SERVER ステートメントを使用してサーバーを登録します。
4. CREATE NICKNAME ステートメントを使用してニックネームを使用します。
5. 非ルート・ニックネームの連合ビューを作成します。

ルート・ニックネームは、XML 文書の最上位にあるエレメントを識別します。非ルート・ニックネームは、XML 文書の最下位にあるエレメントを識別します。

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 ラッパーが連合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。

関連タスク:

- 243 ページの『[XML ラッパーの登録](#)』
- 244 ページの『[XML ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定](#)』
- 245 ページの『[XML データ・ソースのサーバーの登録](#)』
- 251 ページの『[XML データ・ソースのニックネームの登録](#)』
- 257 ページの『[非ルート・ニックネームの連合ビューの作成 \(XML ラッパー\)](#)』

XML ラッパーの登録

XML ラッパーの登録は、XML を連合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。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、連合サーバーがデータ・ソースとの通信やデータ・ソースからのデータの検索に使用するメカニズムです。ラッパーは、ライブラリー・ファイルとして、システムにインストールされています。

次のオペレーティング・システムで XML ラッパーを使用できます。

- AIX
- HP-UX
- Linux
- Solaris オペレーティング環境
- Windows NT

手順:

XML ラッパーを登録するには、CREATE WRAPPER ステートメントをサブミットします。

たとえば、AIX 上でデフォルト・ライブラリー・ファイル libdb2lxml.a から my_xml という XML ラッパーを登録するには、次のステートメントを実行します。

```
CREATE WRAPPER my_xml LIBRARY 'libdb2lxml.a';
```

ここでのタスクの順序では、次のタスクは『XML ラッパーの DB2_DJ_COMM プロファイル変数の設定』です。

関連タスク:

- 244 ページの『XML ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定』
- 「*DB2 Information Integrator* インストール・ガイド」の『非リレーショナル・ラッパー・ライブラリーをチェックする』

関連資料:

- 「*SQL* リファレンス 第 2 巻」の『CREATE WRAPPER ステートメント』

XML ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定

XML ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定は、XML を連合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 文書にアクセスする際のパフォーマンスを高めるため、DB2_DJ_COMM DB2 プロファイル変数をオプションで設定できます。この変数は、初期設定時に連合サーバーがラッパーをロードするかどうかを決定します。

手順:

DB2_DJ_COMM DB2 プロファイル変数を設定するには、関連する CREATE WRAPPER ステートメントに指定したラッパーに対応するラッパー・ライブラリーを使用して **db2set** コマンドをサブミットします。

例:

```
db2set DB2_DJ_COMM='libdb2lxml.a'
```

等号 (=) の両側にスペースを入れないようにしてください。

データベースの始動時に連合サーバーがラッパー・ライブラリーをロードすると、システムのプロセッサ使用量が増加します。過度の使用を避けるには、アクセスしたいライブラリーのみを指定してください。

ここでのタスクの順序では、次のタスクは『XML データ・ソースのサーバーの登録』です。

関連概念:

- 「管理ガイド: インプリメンテーション」の『環境変数およびプロファイル・レジストリー』

関連タスク:

- 245 ページの『XML データ・ソースのサーバーの登録』

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』

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 を連合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、連合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 サーバーを連合システムに登録するには、CREATE SERVER ステートメントを実行します。

例:

```
CREATE SERVER xml_server WRAPPER my_xml;
```

ここでのタスクの順序では、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。

関連タスク:

- 251 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』

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、連合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、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ことで、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す。

関連概念:

- 239 ページの『XML とは?』
- 248 ページの『XML ラッパーのコスト・モデル機能』
- 249 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:

- 251 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:

- 363 ページの『CREATE NICKNAME ステートメント構文 - XML ラッパー』
- 251 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・ニックネームの要素のうち行を生成するルート・要素 (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・要素 (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・要素および行生成非ルート・要素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検索するための XPath 式を評価するのに必要な時間 (ミリ秒) です。表示される XPath 式には、実際の行を検索するための XPath 式に加えて、それらの行の中に含まれる列値を検索する XPath 式も含まれます。

関連概念:

- 239 ページの『XML とは?』
- 246 ページの『ニックネームと XML 文書の間のデータ関連』
- 249 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:

- 363 ページの『CREATE NICKNAME ステートメント構文 - XML ラッパー』
- 251 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します。

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:

- 239 ページの『XML とは?』

- 246 ページの『ニックネームと XML 文書の間のデータ関連』
- 248 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 を連合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、親ニックネームの元素内でネストされている元素に対応します。

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 ステートメントを実行します。

ここでのタスクの順序では、次のタスクは『非ルート・ニックネームの連合ビューの作成 (XML ラッパー)』です。

関連タスク:

- 257 ページの『非ルート・ニックネームの連合ビューの作成 (XML ラッパー)』

関連資料:

- 363 ページの『CREATE NICKNAME ステートメント構文 - XML ラッパー』
- 251 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。

完全指定の例:

以下に示すのは、図 10 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。

```
<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 10. サンプル XML ファイル

親ニックネーム `customers` を作成するには、以下のステートメントを実行します。

```
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)  OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER xml_server
OPTIONS(DIRECTORY_PATH '/home/db2user',
       XPATH '//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 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。

これで、customers ニックネームの子のニックネーム (orders、payments、および items) を作成することができます。

以下のニックネーム・ステートメントを実行して orders 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)  OPTIONS(FOREIGN_KEY 'CUSTOMERS'))
FOR SERVER xml_server
OPTIONS( XPATH './order');
```

以下のニックネーム・ステートメントを実行して payments 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)  OPTIONS(FOREIGN_KEY 'CUSTOMERS'))
FOR SERVER xml_server
OPTIONS( XPATH './payment');
```

以下のニックネーム・ステートメントを実行して items 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)  OPTIONS(FOREIGN_KEY 'ORDERS'))
FOR SERVER xml_server
OPTIONS( XPATH './item');
```

列オプションの例:

以下の列オプションの例は、DOCUMENT 列オプションを使用してニックネームを作成する方法を示しています。

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)   OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER xml_server
OPTIONS(XPATH './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)   OPTIONS(PRIMARY_KEY 'YES')
  FOR SERVER xml_server
  OPTIONS(XPATH '/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 文書を検索します。

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)   OPTIONS(PRIMARY_KEY 'YES')
  FOR SERVER xml_server
  OPTIONS(XPATH '/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 '/employee');
```

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:

以下のビューの例は、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ます。これらの例では 256 ページの図 11 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。

```
<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 11. 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.amount, 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 it.quantity, it.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:

- 251 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:

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

非ルート・ニックネームの連合ビューの作成 (XML ラッパー)

非ルート・ニックネームの連合ビューの作成 (XML ラッパー) は、XML を連合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連合ビューを定義できます。連合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連合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:

- 251 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 - 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 '/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:

- 363 ページの『CREATE NICKNAME ステートメント構文 - XML ラッパー』
- 251 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 44. 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 ステートメントの中の列のデータ・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 ステートメントの中の列のデータ・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。

表 44. 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「<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。
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。

表 44. 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「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。

表 44. 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「<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。
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 ラッパー」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。

表 44. 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 ラッパー」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「列のデータ・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 ラッパー」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せん。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 ラッパー」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 ラッパー」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 列で正しくない述部を使用」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 ラッパー」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 列で正しくない述部を使用」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 ラッパー」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 列で正しくない述部を使用」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 44. 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 ラッパー」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 オプションと DOCUMENT オプションは互換性がない」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 ラッパー」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 オプションと FOREIGN_KEY オプションは互換性が 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 ラッパー」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 オプションと PRIMARY_KEY オプションは互換性が 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 ラッパー」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 オプションと FOREIGN_KEY オプションは互換性が 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 ラッパー」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 オプションと PRIMARY_KEY オプションは互換性が 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。

表 44. 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 ラッパー」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 オプションと PRIMARY_KEY オプションは互換性がない」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 ラッパー」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。
SQL1822N	データ・ソース「XML ラッパー」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 ラッパー」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 ラッパー」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 ラッパー」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。

表 44. 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 ラッパー」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 ラッパー」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 ラッパー」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 オプションと DOCUMENT オプションは互換性がない」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。
SQL1822N	データ・ソース「XML ラッパー」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 オプションと SOAP オプションは互換性がない」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 ラッパー」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 オプションと SOAP オプションは互換性がない」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。

表 44. 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 ラッパー」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 オプションと DIRECTORY_PATH オプションは互換性がない」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 ラッパー」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 オプションと STREAMING オプションは互換性がない」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 ラッパー」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 オプションと FOREIGN_KEY オプションは互換性がない」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 ラッパー」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 オプションと FOREIGN_KEY オプションは互換性がない」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。
SQL1822N	データ・ソース「XML ラッパー」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。

表 44. 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 ラッパー」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 ラッパー」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 ラッパー」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 データへの参照が欠落」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 ラッパー」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 列オプションの値 'SOAP' 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。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 ラッパー」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。SOAP オプションを指定する必要があります。

表 44. 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 ラッパー」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。
SQL1822N	データ・ソース「XML ラッパー」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 オプションはルート・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 ラッパー」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 オプションはルート・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>" に対して有効な "<option_type>" 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。

表 44. 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
SQL1881N	"DIRECTORY_PATH" は " <code><object_name></code> " の有効な "NICKNAME" オプションではありません。	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ーは、ルート・ディレクトリー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、" <code><object_name></code> " の " <code><option_value></code> " に設定することはできません。	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。
SQL1882N	"nickname" オプション "STREAMING" は、" <code><object_name></code> " の " <code><option_value></code> " に設定することはできません。	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	" <code><option_name></code> " は " <code><object_name></code> " に対して必須 " <code><option_type></code> " 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。

関連資料:

- 154 ページの『表構造ファイル・ラッパーのメッセージ』
- 186 ページの『Documentum ラッパーのメッセージ』
- 203 ページの『Excel ラッパーのメッセージ』
- 234 ページの『BLAST ラッパーのメッセージ』
- 363 ページの『CREATE NICKNAME ステートメント構文 - XML ラッパー』

第 17 章 Entrez データ・ソースへのアクセスの構成

この章では、Entrez とは何か、また Entrez データ・ソースを連合システムに加える方法について解説し、さらに Entrez ラッパーに関連したエラー・メッセージのリストを示します。

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。

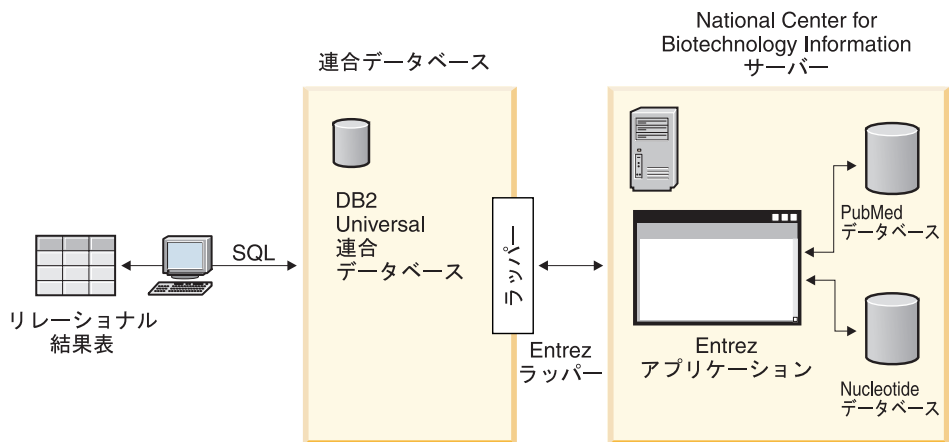


図 12. Entrez ラッパーがどのように機能するか

Entrez ラッパーを構成するエレメントの多くは、全データベースに共通です。そのようなエレメントとしては、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 コーティリティー
- 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

関連概念:

- 145 ページの『表構造ファイルとは?』
- 159 ページの『Documentum とは?』
- 195 ページの『Excel とは?』
- 211 ページの『BLAST とは?』
- 239 ページの『XML とは?』

関連タスク:

- 272 ページの『Entrez の連合システムへの追加』

Entrez の連合システムへの追加

ステートメントは、DB2 コマンド行プロセッサから実行できます。Entrez ラッパーを連合システムに追加した後、Entrez データ・ソースに対して照会を実行できます。

手順:

Entrez データ・ソースを連合サーバーに追加するには、以下のようにします。

1. CREATE FUNCTION ステートメントを実行して、カスタム関数を登録します。
2. CREATE WRAPPER ステートメントを実行して、ラッパーを登録します。

3. オプション: 照会のパフォーマンスを向上させるために DB2_DJ_COMM 環境変数を設定します。
4. CREATE SERVER ステートメントを実行して、サーバーを登録します。
5. CREATE NICKNAME ステートメントを実行して、ニックネームを登録します。

関連タスク:

- 273 ページの『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

Entrez ラッパーのカスタム関数の登録は、Entrez を連合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しなければなりません。
- 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、AS TEMPLATE キーワードを使用してCREATE FUNCTION ステートメントを実行しなければなりません。

各関数の完全修飾名は、Entrez.<function-name> です。

以下の例では、1 つのバージョンの CONTAINS 関数を登録します。

```
CREATE FUNCTION entrez.contains (varchar(), varchar())  
  RETURNS INTEGER AS TEMPLATE;
```

カスタム関数を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samples/lifesci/entrez 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、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カスタム関数を登録することができます。

ここでのタスクの順序では、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』
- 278 ページの『カスタム関数と Entrez 照会』
- 283 ページの『カスタム関数表 - Entrez ラッパー』

Entrez ラッパーの登録

Entrez ラッパーの登録は、Entrez を連合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。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、連合サーバーがデータ・ソースとの通信やデータ・ソースからのデータの検索に使用するメカニズムです。ラッパーは、ライブラリー・ファイルとして、システムにインストールします。

手順:

Entrez ラッパーを登録するには、CREATE WRAPPER ステートメントを実行します。

たとえば、AIX 上でデフォルト・ライブラリー・ファイル libdb2lsentrez.a から entrez_wrapper という Entrez ラッパーを作成するには、次のステートメントをサブミットします。

```
CREATE WRAPPER entrez_wrapper LIBRARY 'libdb2lsentrez.a'  
  OPTIONS(EMAIL 'jeff@someplace.com', DB2_FENCED 'N');
```

ラッパーを登録する際には、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組み込まれ、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。

ここでのタスクの順序では、次のタスクは『Entrez ラッパーの DB2_DJ_COMM プロファイル変数の設定』です。

関連タスク:

- 274 ページの『Entrez ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定』
- 「*DB2 Information Integrator* インストール・ガイド」の『非リレーショナル・ラッパー・ライブラリーをチェックする』

関連資料:

- 「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 DB2_DJ_COMM DB2 プロファイル変数の設定

Entrez ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定は、Entrez を連合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。Entrez データ・ソースにアクセスする際のパフォーマンスを向上させるために、DB2_DJ_COMM DB2 プロファイル変数を設定します。この変数は、初期設定時に連合サーバーがラッパーをロードするかどうかを決定します。

手順:

DB2_DJ_COMM プロファイル変数を設定するには、関連する CREATE WRAPPER ステートメントに指定したラッパーに対応するラッパー・ライブラリーを使用して **db2set** コマンドを実行します。

```
db2set DB2_DJ_COMM='libdb21sentrez.a'
```

等号 (=) の両側にスペースを入れないようにしてください。

データベースの始動時に連合サーバーがラッパー・ライブラリーをロードすると、プロセスの使用量が増加します。過度の使用を避けるには、アクセスしたいライブラリーのみを指定してください。

ここでのタスクの順序では、次のタスクは『Entrez データ・ソースのサーバーの登録』です。

関連タスク:

- 275 ページの『Entrez データ・ソースのサーバーの登録』

Entrez データ・ソースのサーバーの登録

Entrez データ・ソースのサーバーの登録は、Entrez を連合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。

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 サーバーを連合システムに登録するには、CREATE SERVER ステートメントを実行します。

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実行します。

```
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実行します。

```
CREATE SERVER nucleotid_server1
TYPE NUCLEOTIDE
VERSION 1.0
WRAPPER entrez_wrapper;
```

ここでのタスクの順序では、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:

- 276 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:

- 370 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 を連合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、ルート表と呼ばれます。データベース内の他の表すべては、この親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 ステートメントを実行します。

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;
CREATE NICKNAME PMURL 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 nuc1 OPTIONS (REMOTE_OBJECT 'GBSEQ');
CREATE NICKNAME NewFeatures FOR SERVER nuc1
  OPTIONS (REMOTE_OBJECT 'GBFEATURES', PARENT 'NEWSEQ');
CREATE NICKNAME NewIntervals FOR SERVER nuc1
  OPTIONS (REMOTE_OBJECT 'GBINTERVALS', PARENT 'NEWFEATURES');
CREATE NICKNAME NewQualifiers FOR SERVER nuc1
  OPTIONS (REMOTE_OBJECT 'GBQUALIFIERS', PARENT 'NEWFEATURES');
CREATE NICKNAME NewReference FOR SERVER nuc1
  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 NEWPMESH 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');
CREATE NICKNAME NEWPMURL FOR SERVER pubmed_server
  OPTIONS (REMOTE_OBJECT 'PMURL', PARENT 'NEWPMARTICLES');
```

ここでのタスクの順序では、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。

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE NICKNAME ステートメント』
- 283 ページの『PubMed および Nucleotide スキーマ表』

カスタム関数と Entrez 照会

連合環境は 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 構文で必要とされる 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[tit1])
```

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[tit1]) OR (drosophila[all] OR "fruit fly"[all]))
```

関連資料:

- 283 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:

- 280 ページの『Entrez ラッパーで無効な WHERE 文節』

関連タスク:

- 281 ページの『Entrez データ・ソース — 照会の例』
- 273 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:

- 279 ページの『Entrez ラッパーの関係述部』

関連タスク:

- 281 ページの『Entrez データ・ソース — 照会の例』
- 273 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、エントリーに 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 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、姓付きで表示したり姓なしで表示したりできます。姓なしの場合、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。姓は、(<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を使用されます。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を使用されます。

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:

- 279 ページの『Entrez ラッパーの関係述部』
- 280 ページの『Entrez ラッパーで無効な WHERE 文節』

関連タスク:

- 273 ページの『Entrez ラッパーのカスタム関数の登録』

カスタム関数表 - Entrez ラッパー

表 45. Entrez ラッパーのカスタム関数

関数名	説明
CONTAINS (col VARCHAR(), term VARCHAR()), CONTAINS (col INTEGER, term VARCHAR()), CONTAINS (col SMALLINT, term VARCHAR()), CONTAINS (col REAL, term VARCHAR()), CONTAINS (col DOUBLE, term VARCHAR()), CONTAINS (col DATE, term VARCHAR()), CONTAINS (col TIME, term VARCHAR()), CONTAINS (col CHAR(), term VARCHAR()), CONTAINS (col TIMESTAMP(), term VARCHAR())	与えられた式を使用してタグ付きの列を検索します。 col タグ付きの列 term 検索語
SEARCH_TERM (col VARCHAR(), term VARCHAR())	Entrez 検索エンジンに Entrez 検索語を直接渡します。 col タグ付きの列 term 検索語

PubMed および Nucleotide スキーマ表

ここでは、PubMed および Nucleotide スキーマの表を提供します。

PubMed スキーマ

このスキーマは、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。スキーマは、関連する複数のニックネームで構成されています。次の表の中で、Q 列はフィールド・タグです。検索可能な有効なタグのリストについては、<http://www.ncbi.nlm.nih.gov/entrez/query/static/help/pmhhelp.html#SearchFieldDescriptionsandTags> を参照してください。F 列は、ニックネーム列が、指定されたフェッチ・キーかどうかを示します。フェッチ・キーを使うことで、処理が円滑になることがあります。

表 46. PubMed PMArticles ニックネーム

列名	データ・タイプ	説明	Q	F
PMID	VARCHAR(10) NOT NULL	PubMed ID PRIMARY_KEY Y	UID	Y
MedlineID	VARCHAR(10)	Medline ID	UID	Y
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。		
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。		
DateCreated	DATE NOT NULL			
DateCompleted	DATE			
DateRevised	DATE			
ArticleTitle	VARCHAR(250) NOT NULL		TI	
Pagination	VARCHAR(32)			
Abstract	VARCHAR(32000)		TIAB	
Affiliation	VARCHAR(250)	第 1 著者の姓と住所	AD	
AuthorList	VARCHAR(3200)	著者のリスト; 標準化済み	AU	
LanguageList	VARCHAR(250) NOT NULL	セミコロン区切りリスト	LA	
PublicationTypeList	VARCHAR(250) NOT NULL	セミコロン区切りリスト	PT	

表 46. PubMed PMArticles ニックネーム (続き)

列名	データ・タイプ	説明	Q	F
VernacularTitle	VARCHAR(250)			
DateOfElectronicPublication	VARCHAR(32)	NCBI スキーマでは、この列の構造は指定されていない		
Country	VARCHAR(128)			
MedlineTA	VARCHAR(250) NOT NULL			TA
NlmUniqueID	VARCHAR(32)	NlmUniqueID が指定されていない場合には MedlineCode		
GeneSymbolList	VARCHAR(250)	セミコロン区切りリスト; 1996 年以來使用されていない		
NumberOfReferences	INTEGER			
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	
KeywordList	VARCHAR(3200)	セミコロン区切りリスト		
SpaceFlightMissionList	VARCHAR(250)	セミコロン区切りリスト		
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		
PublicationStatus	VARCHAR(32)			
ProviderID	VARCHAR(32)			
CitationSubsetList	VARCHAR(250)	セミコロン区切りリスト		SB
AllFields	VARCHAR(1)	疑似列; 常に NULL を戻す		ALL
TextWords	VARCHAR(1)	疑似列; 常に NULL を戻す		TW

表 46. PubMed PMArticles ニックネーム (続き)

列名	データ・タイプ	説明	Q	F
PubDate	DATE	雑誌および書籍の 出版日付 + medline 日付	DP	
PubDateString	VARCHAR(32)	雑誌および書籍の 出版日付 + medline 日付	DP	
Title	VARCHAR(250)	書籍または雑誌の タイトル	TA	
Journal_ISSN	CHAR(9)		TA	
Journal_Volume	VARCHAR(10)		VI	
Journal_Issue	VARCHAR(10)		IP	
Journal_Coden	VARCHAR(32)			
Journal_ISOAbbreviation	VARCHAR(32)			
Book_Publisher	VARCHAR(128)			
Book_Authors	VARCHAR(250)	その他の著者のリ ストとして標準化 済み		
Book_CollectionTitle	VARCHAR(128)			
Book_Volume	VARCHAR(10)			

表 47. PubMed PMAccession ニックネーム

列名	データ・タイプ	説明	Q
PMID	VARCHAR(10) NOT NULL	FOREIGN_KEY PMARTICLES	
DataBankName	VARCHAR(250) NOT NULL		SI
Accession	VARCHAR(32) NOT NULL		SI

表 48. PubMed PMChemical ニックネーム

列名	データ・タイプ	説明	Q
PMID	VARCHAR(10) NOT NULL	FOREIGN_KEY PMARTICLES	
NameOfSubstance	VARCHAR(128) NOT NULL		NM

表 48. PubMed PMChemical ニックネーム (続き)

列名	データ・タイプ	説明	Q
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	

表 49. PubMed PMMeSHHeading ニックネーム

列名	データ・タイプ	説明	Q
PMID	VARCHAR(10) NOT NULL	FOREIGN_KEY	ID PMARTICLES
DescriptorOrName	VARCHAR(128) NOT NULL		MH (照会に述部 "DescriptorIsMajor = Y" が含まれている場合は、検索語は MAJR)
DescriptorIsMajor	CHAR NOT NULL	記述子がメジャーの場合 Y	
QualifierOrSubhead	VARCHAR(128)		SH
QSIIsMajor	CHAR	修飾語または副題がメジャーの場合 Y	

表 50. PubMed PMComments ニックネーム

列名	データ・タイプ	説明	Q
PMID	VARCHAR(10) NOT NULL	FOREIGN_KEY	PMARTICLES
RefSource	VARCHAR(128) NOT NULL		
Type	VARCHAR(32) NOT NULL	CommentOn, CommentIn, ErratumIn, ErratumFor, RepublishedFrom, RepublishedIn, RetractionOf, RetractionIn, UpdateIn, UpdateOf, SummaryForPatents, OriginalReportIn	

表 50. PubMed PMComments ニックネーム (続き)

列名	データ・タイプ	説明	Q
Note	VARCHAR(3200)		

表 51. PubMed PMArticleID ニックネーム

列名	データ・タイプ	説明	Q
PMID	VARCHAR(10) NOT NULL	FOREIGN_KEY PMARTICLES	
ArticleID	VARCHAR(32) NOT NULL		
IdType	VARCHAR(8) NOT NULL	doi, pii, pmcpid, pmpid, sici, pubmed, medline, pmcid	

表 52. PubMed PMURL ニックネーム

列名	データ・タイプ	説明	Q
PMID	VARCHAR(10) NOT NULL	FOREIGN_KEY PMARTICLES	
URL	VARCHAR(250) NOT NULL		
Language	CHAR(2)	ISO 言語コード	
Type	CHAR(1)	全文の場合 F、サマ リーの場合 S	

Nucleotide スキーマ

詳細については、http://www.ncbi.nlm.nih.gov/entrez/query/static/help/Summary_Matrices.html#Search_Fields_and_Qualifiers を参照してください。

表 53. Nucleotide GBSeq ニックネーム

列名	データ・タイプ	説明	Q	F
PrimaryAccession	VARCHAR(16) NOT NULL	主 ACCESSION 番号	PACC	Y
SequenceKey	VARCHAR(32) NOT NULL	PRIMARY_KEY Y		
LocusName	VARCHAR(16) NOT NULL		ACCN	

表 53. Nucleotide GBSeg ニックネーム (続き)

列名	データ・タイプ	説明	Q	F
SeqLength	INTEGER NOT NULL		SLEN	
Strandedness	VARCHAR(32)	not-set, single-stranded, double-stranded, mixed-stranded		
MoleculeType	VARCHAR(16)	nucleic-acid, dna, rna, trna, rrna, mrna, urna, snrna, snorna, peptide	PROP	
Topology	VARCHAR(16)	linear, circular		
Division	CHAR(3) NOT NULL		PROP	
UpdateDate	DATE NOT NULL		MDAT	
CreateDate	DATE NOT NULL			
Definition	VARCHAR(7000) NOT NULL		TITL	
Version	INTEGER			
GI	VARCHAR(16)	FETCH_KEY Y	UID	
KeywordList	VARCHAR(7000)	セミコロン区 切りリスト	KYWD	
Segment	VARCHAR(250)			
Source	VARCHAR(200) NOT NULL		ORGN	
Organism	VARCHAR(7000) NOT NULL		ORGN	
Taxonomy	VARCHAR(7000) NOT NULL			
Comment	VARCHAR(7000)			
Primary	VARCHAR(7000)			
SourceDB	VARCHAR(250)			
Sequence	CLOB			
AllText	VARCHAR(1)	疑似列、常に NULL を戻す	ALL	

表 53. *Nucleotide GBSeq* ニックネーム (続き)

列名	データ・タイプ	説明	Q	F
FreeText	VARCHAR(1)	疑似列、常に NULL を戻す	WORD	

表 54. *GBReference* ニックネーム

列名	データ・タイプ	説明	Q
SequenceKey	VARCHAR(32) NOT NULL	FOREIGN_KEY Y	
ReferenceNum	INTEGER NOT NULL	GBReference_reference から構文解析される	
RangeLow	INTEGER NOT NULL	参照の下限 (GBReference_reference から構文解析)	
RangeHigh	INTEGER NOT NULL	参照の上限 (GBReference_reference から構文解析)	
Authors	VARCHAR(3200)	GenBank 形式によ る氏名のセミコロン 区切りリスト	AUTH
Consortium	VARCHAR(250)		
Title	VARCHAR(250)		WORD
Journal_Title	VARCHAR(250) NOT NULL		JOUR
MedlineID	INTEGER		
PubMedID	INTEGER		
Remarks	VARCHAR(3200)		

表 55. *Nucleotide GBFeatures* ニックネーム

列名	データ・タイプ	説明	Q
SequenceKey	VARCHAR(32) NOT NULL	FOREIGN_KEY GBSEQ	
FeatureJoinKey	VARCHAR(32) NOT NULL	PRIMARY_KEY Y	
FeatureKey	VARCHAR(20) NOT NULL		FKEY

表 55. *Nucleotide GBFeatures* ニックネーム (続き)

列名	データ・タイプ	説明	Q
FeatureLocation	VARCHAR(200) NOT NULL		

表 56. *Nucleotide GBIntervals* ニックネーム

列名	データ・タイプ	説明	Q
FeatureJoinKey	VARCHAR(32) NOT NULL	FOREIGN_KEY GBFEATURES	
IntervalFrom	INTEGER		
IntervalTo	INTEGER		
IntervalPoint	INTEGER		
IntervalAccession	VARCHAR(32) NOT NULL		

表 57. *Nucleotide GBQualifiers* ニックネーム

列名	データ・タイプ	説明	Q
FeatureJoinKey	VARCHAR(32) NOT NULL	FOREIGN_KEY GBFEATURES	
QualifierName	VARCHAR(50)		
QualifierValue	VARCHAR(32000)		

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 58. *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。

表 58. *Entrez* のラッパーによって出されるメッセージ (続き)

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す。(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す。(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 58. *Entrez* のラッパーによって出されるメッセージ (続き)

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」。)	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「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す。(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す。(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 " <code><wrapper_name></code> " は、連合サーバーに定義を試みているデータ・ソース (" <code><server_type></code> ", " <code><server_version></code> ") の " <code>version</code> " 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。

表 58. *Entrez* のラッパーによって出されるメッセージ (続き)

SQL1816N	ラッパー "<wrapper_name>" は、連合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、連合データベースに定義したいデータ・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。
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>" に対して有効な "<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>" の "<option_value>" に設定することはできません。	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。

表 58. Entrez のラッパーによって出されるメッセージ (続き)

SQL1883N	"<option_name>" は "<option_name>" に対して必 須 "<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	"FOREIGN_KEY" ("COLUMN" オプション) が複数回指定されました。	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	"PRIMARY_KEY" ("COLUMN" オプション) が複数回指定されました。	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。
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。

表 58. *Entrez* のラッパーによって出されるメッセージ (続き)

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。

第 18 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 とは何か、Extended Search データ・ソースを連合システムに加える方法、また、SQL を使用して Extended Search データ・ソースを検索する方法について説明します。さらに、Extended Search ラッパーに関連するエラー・メッセージのリストも示します。

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検索し、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 製品に付属のドキュメンテーション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、その後を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。インター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、ノート 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、連合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・ソースが連合データベース・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。

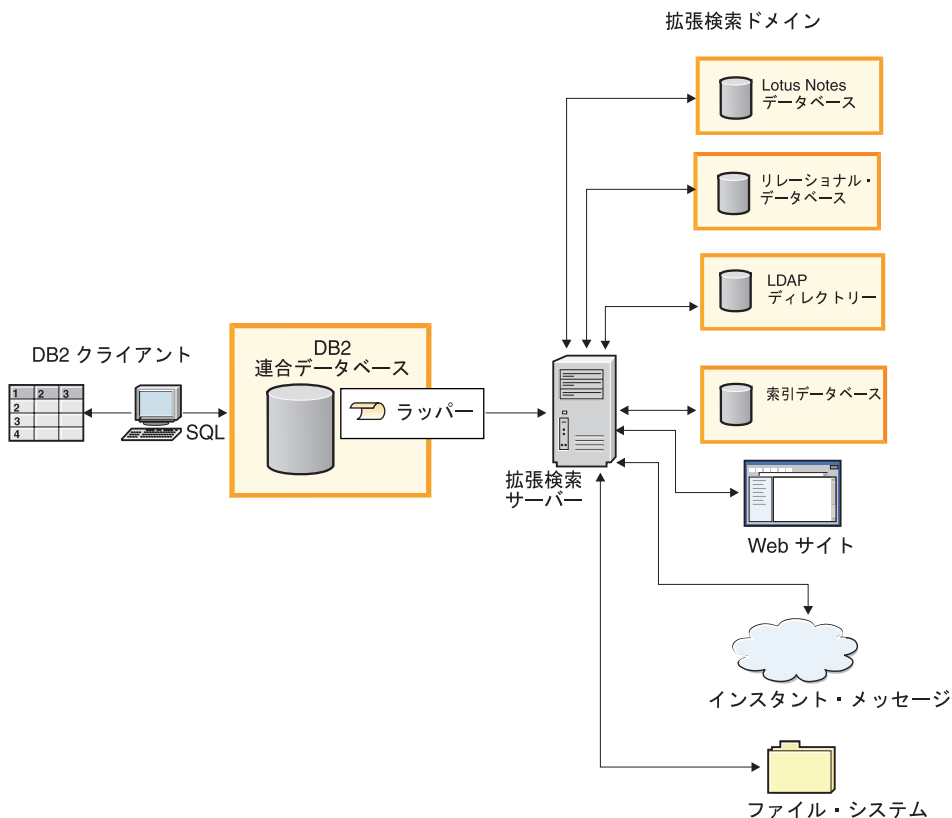


図 13. Extended Search ラッパーがどのように機能するか

関連タスク:

- 305 ページの『連合サーバーへの 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ることで 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るとください。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、またはそれを削除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ようになります。

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、結果データを連合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:

- 302 ページの『Extended Search の垂直表』

関連タスク:

- 307 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:

- 312 ページの『Extended Search ラッパー - 照会の例』
- 355 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ふ可能性があります。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、固有の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となります。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、連合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。

列名	データ・タイプ	説明
以下の 3 つの列は、どのニックネームについても常に戻されます。		
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作成されます。		
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。

列名	データ・タイプ	説明
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 484 DECIMAL 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、連合データベース・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:

- 300 ページの『Extended Search のニックネーム』

関連タスク:

- 307 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:

- 312 ページの『Extended Search ラッパー - 照会の例』
- 355 ページの『CREATE NICKNAME ステートメント構文 - Extended Search ラッパー』

連合サーバーへの Extended Search データ・ソースの追加

Extended Search ラッパーは、Microsoft Windows NT、Microsoft Windows 2000、および IBM AIX オペレーティング・システムでインストールできます。このラッパーを使用すれば、Windows、AIX、Sun Solaris、および Red Hat Linux for Intel オペレーティング・システムに存在する Extended Search サーバーを検索できます。

前提条件:

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・ソースを連合システムに追加するには、次のようにします。

1. Extended Search ラッパーを登録する。
2. ラッパーで使用したい Extended Search サーバーを登録する。
3. 検索したい Extended Search ソースのニックネームを登録する。
4. Extended Search ソースを検索する必要がある DB2 ユーザーのユーザー・マッピングを登録する。
5. 検索オプションを指定する Extended Search カスタム関数を登録する。

関連タスク:

- 305 ページの『Extended Search ラッパーの登録』
- 306 ページの『Extended Search データ・ソースのサーバーの登録』
- 307 ページの『Extended Search データ・ソースのニックネームの登録』
- 308 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』
- 309 ページの『Extended Search カスタム関数の登録』

Extended Search ラッパーの登録

この作業は、Extended Search データ・ソースを連合システムに追加するためのメインタスクの一部です。Extended Search データ・ソースを検索するには、最初に Extended Search ラッパーを登録しなければなりません。

手順:

Extended Search ラッパーを登録するには、DB2 コマンド行プロセッサから CREATE WRAPPER ステートメントを実行します。

たとえば、Windows システムで NotesDBwrapper 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実行します。

```
CREATE WRAPPER NotesDBwrapper LIBRARY 'db2uies.dll'
```

AIX システムで myESwrapper 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実行します。

```
CREATE WRAPPER myESwrapper LIBRARY 'libdb2uies.a'
```

次の作業は、『Extended Search データ・ソースのサーバーの登録』です。

関連タスク:

- 306 ページの『Extended Search データ・ソースのサーバーの登録』

関連資料:

- 373 ページの『CREATE WRAPPER ステートメント構文 - Extended Search ラッパー』

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・ソースを連合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、連合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 ステートメントを実行します。

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実行します。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実行します。

```
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:

- 307 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:

- 371 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・ソースのニックネームの登録

この作業は、Extended Search データ・ソースを連合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 ステートメントを実行します。

たとえば、以下のステートメントを実行して、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 カテゴリー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実行して、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 エントリーに拡張して、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:

- 300 ページの『Extended Search のニックネーム』
- 302 ページの『Extended Search の垂直表』

関連タスク:

- 308 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:

- 312 ページの『Extended Search ラッパー - 照会の例』
- 355 ページの『CREATE NICKNAME 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・ソースを連合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・マッピングを連合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 ステートメントを実行します。

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:

- 309 ページの『Extended Search カスタム関数の登録』

関連資料:

- 373 ページの『CREATE USER MAPPING ステートメント構文 - Extended Search ラッパー』

Extended Search カスタム関数の登録

この作業は、Extended Search データ・ソースを連合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 ES_SEARCH 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:

- ES_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 (ES_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。
- ES_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。
- ES_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_N1, ES_N2
WHERE ESWRAPPER.ES_SEARCH(ES_N1.DOC_RANK, 'IBM') = 1 AND
      ESWRAPPER.ES_SEARCH(ES_N2.DOC_RANK, 'IBM') = 1
```

手順:

Extended Search カスタム関数を登録するには、DB2 コマンド行プロセッサから以下の CREATE FUNCTION ステートメントを実行します。

```
CREATE FUNCTION ESWRAPPER.ES_SEARCH(INTEGER, VARCHAR(1024))
  RETURNS INTEGER AS TEMPLATE
```

関連資料:

- 312 ページの『Extended Search ラッパー - 照会の例』
- 315 ページの『Extended Search ラッパー - 汎用照会言語』
- 347 ページの『CREATE FUNCTION ステートメント構文 - Extended Search ラッパー』
- 310 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 - 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:

- 309 ページの『Extended Search カスタム関数の登録』

関連資料:

- 312 ページの『Extended Search ラッパー - 照会の例』
- 315 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ーが含まれています。

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:

- 300 ページの『Extended Search のニックネーム』
- 302 ページの『Extended Search の垂直表』

関連タスク:

- 307 ページの『Extended Search データ・ソースのニックネームの登録』
- 309 ページの『Extended Search カスタム関数の登録』

関連資料:

- 315 ページの『Extended Search ラッパー - 汎用照会言語』
- 347 ページの『CREATE FUNCTION ステートメント構文 - Extended Search ラッパー』
- 355 ページの『CREATE NICKNAME ステートメント構文 - Extended Search ラッパー』
- 310 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 で次の照会を実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実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

```

GQL 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:

- 309 ページの『Extended Search カスタム関数の登録』

関連資料:

- 312 ページの『Extended Search ラッパー - 照会の例』
- 347 ページの『CREATE FUNCTION ステートメント構文 - Extended Search ラッパー』
- 310 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 59. 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す。(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 59. *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 <i>Extended Search</i> 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。 <i>Extended Search</i> 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>" に存在するデータ・タイプ・マッピングがありません。	CREATE NICKNAME ステートメントまたは ALTER NICKNAME ステートメントにある列が、 <i>Extended Search</i> 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 ステートメントを連合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 <i>Extended Search</i> 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、 <i>Extended Search</i> 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	Connection to remote <i>Extended Search</i> server "<host_name>" on port "<port_number>" could not be established or was terminated.	<i>Extended Search</i> ラッパーは指定されたポートでリモート <i>Extended Search</i> サーバーに接続しようとしたましたが、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 <i>Extended Search</i> サーバーのホスト名とポート番号を検査し、 <i>Extended Search</i> 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。

表 59. *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
SQL1834N	User-defined column " <code><column_name></code> " is identical to a fixed column for wrapper " <code><wrapper_name></code> " but uses a different data type.	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 <i>Extended Search</i> ラッパーの固定列と同じ名前を持つが異なるデータ・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。
SQL1835N	Extended Search object " <code><object_name></code> " of type " <code><object_type></code> " could not be found on the remote Extended Search server " <code><host_name></code> ".	指定された <i>Extended Search</i> オブジェクトは、指定されたりモート <i>Extended Search</i> 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 <i>Extended Search</i> 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	No column mapping exists between user-defined column " <code><column_name></code> " and a field name on the remote Extended Search server " <code><host_name></code> ".	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ーに属するデータ・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。

表 59. *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
SQL1837N	The required option " <code><option_name></code> " of type " <code><object_type></code> " on wrapper " <code><wrapper_name></code> " cannot be dropped.	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 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	The search statement " <code><option_name></code> " is not a valid Extended Search query.	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	One or more search parameters are not valid.	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。
SQL1881N	" <code><option_name></code> " は " <code><object_name></code> " に対して有効な " <code><option_type></code> " 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。Extended Search 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。

表 59. *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
SQL1882N	The "<option_type>" option "<option_name>" cannot be set to "<option_value>" for "<object_name>".	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。Extended Search 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>" に対して必須 "<option_type>" オプションです。	Extended Search 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ていませんでした。Extended Search 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>

関連資料:

- 「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 19 章 HMMER データ・ソースへのアクセスの構成

この章では、HMMER とは何か、また HMMER データ・ソースを連合システムに加える方法について解説し、さらに HMMER ラッパーに関連したエラー・メッセージのリストを示します。

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ツールです。HMMER は、<http://hmmer.wustl.edu/> から無料でダウンロードできます。HMMER は、元々 BLAST の検索の機能向上を目的として開発されたものです。HMMER ラッパーのこのバージョンでは、遺伝子列を使用することによりモデルのデータベースを検索し、テスト遺伝子列がどのファミリーに属するかを決定します。

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できるよう、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 ソフトウェア・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 ラッパーでは、**hmmpfam** プログラムの使用がサポートされています。このプログラムは、PFAM などのプロファイル HMM データベースから、特定の遺伝子列を検索するためのものです。

HMMER ラッパーは **hmmpfam** ユーティリティーを始動します。このユーティリティーは、プロファイル HMM を使用することによって、たんぱく質または核酸の列のファミリーの主要な構造に関するコンセンサスをモデル化します。324 ページの表 60 を参照してください。

表 60. HMMER ユーティリティー

HMMER ユーティリティー	説明
hmmpfam	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。
hmmalign	複数の遺伝子列を配置して 1 つのプロファイル HMM を構成します。
hmmbuild	複数の遺伝子列配置からプロファイル HMM を作成します。
hmmcalibrate	データベース検索実行の前に、プロファイル HMM の該当する統計的有意性パラメーターを決定します。
hmmconvert	HMMER プロファイル HMM を、Genetics Computer Group (GCG) プロファイルなどの他のフォーマットに変換します。
hmmemit	プロファイル HMM に基づいて確率モデルを使用する遺伝子列を生成します。
hmmfetch	HMM データベースから HMM を検索します。
hmmindex	HMM データベースのためのバイナリー・サーバー・サイド・インクルード (server-side includes, SSI) を作成します。
hmmsearch	遺伝子列データベースからプロファイル HMM を検索し、モデル化ファミリーの付加的な同種物を探します。

クライアントにおいてユーザーまたはアプリケーションは、`hmmpfam` コマンド行オプションにマップされる HMMER 固有の述部を含む SQL ステートメントを実行します。それらの SQL ステートメントおよび述部は、HMMER ラッパーを含む連合データベース・サーバーに送られます。

HMMER ラッパーは、HMMER アプリケーションが解釈できる形式に照会を変換した後、その照会を実行するために `hmmpfam` ユーティリティーを始動します。 `hmmpfam` を実行するサーバーには、連合データベース・サーバーを含むシステムとは別個のシステムを使用できます。特別なデーモン・プログラムが HMMER 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、連合データベース・サーバーからの照会要求を受信し、それを HMMER アプリケーションへ送信します。その後、プロファイル・データベースに対して HMMER アプリケーションが実行されます。

325 ページの図 14 に、連合システムでの HMMER の動作を示します。

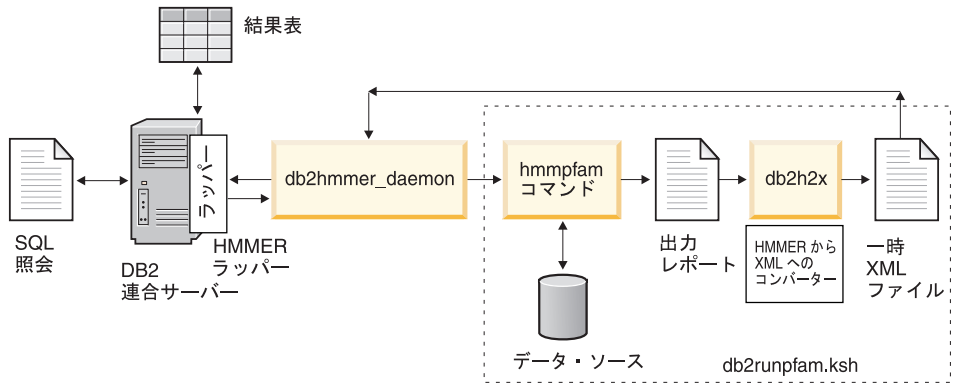


図 14. HMMER ラッパーがどのように機能するか

デーモンは、HMMER の結果を 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 ユーティリティーによって構成され、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。

326 ページの図 15 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 = 'MTMFQQNVVDHQSDEQDKQAKGAAPT KRLNISFNVKI AVNVNTKMTTTH
INQQAPGTSSSSNSQNASPSKI VVRQSSSFDLRQQLARLGRQLASGQDGHGGISTILI INLLLL
ILLSICCDVCRSHNYTVHQSPPEVSKDQMRLLRPKLDSDVVEKVAIWHKHAAPPSIVEGIAISS
RPQSTMAHHPDDRDRDRDPSEEQHGVDERMVLERVTRDCVQRCIVEEDLFLDFEQCEKADNGEK
CYKTRCTKGAQWYRALKEL ESCQEACLSLQFYPYDMPCIGACEMAQRDYWHLQRLAISHLVERTQ
PQLERAPRADGGSTPLTIRWAMHFPEHYLASRPFNIQYQFVDHGHGEELDEQEDQDASGETGSSAW
FNLADYDCDEYYMCEILEALIPYTQYRFRFELPFGENRDEVLYSPATPAYQTPPEGAPISAPVIEH
LMGLDDSHLAVHWHHPGRFTNGPIEGYRLRLSSSEGNATSEQLVPAGRGSYIFSQLQAGTNYTLALS
MINKQEGEPVAKGFVQTHSARNEKPAKDLTESVLLVGRRAVMWQSLEPAGENSMIYQSQEELADIA
WSKREQLLWLLNVHGLRSLKFESGQMVSPAQQLKLDLGNISSGRWVPRRSLFDWLHHRLYFAMES
PERNQSSFQIISTDLLGESAQKVGESFDLPVEQLEVDALNGWIFWRNEESLWRQDLHGRMIHRLLR
IRQPGWFLVQPQHFIIHMLPQEGKFLEISYDGGFKHPLPLPPPSNGAGNGPASSHWQSFALLGRS
LLL PDSGQLILVEQQGAASPSASWPLKNLPDCWAVILLVPESQPLTSAGGKPHSLKALLGAQAAK
ISWKEPERNPYQSADAARSWSYELEVLVDVASQSAF SIRNIRGPIFGLQRLQPDNLYLQVRRAINVD
GEPGEWTEPLAARTWPLGPHRLRWASRQGSVIHTNELGEGLEVQEQLERLPGPMTMVNESVGYVY
TGDGLLHCINLVHSQWGPCISEPLQHVGSVTYDWRGGRVYWTDLARNCVVRMDPWSGSELLPVFE
ANFLALDPRQGHLYYATSSQLSRHGSTPDEAVTYRVNGLGEGIASFVLDLTDQDQLFWLVKSGAL
RLYRAPLTAGGDSLQMIQQIKGVFQAVPDSLQLLRPLGALLWLERSGRRARLVRLAAPLDMELPT
PDQASPASALQLLDPQLPRRDEGVIPTMLVLPDSVRLDDGHWDHFVWRWQPSSTGGNHSVSYRLL
EFGQRLQTLDLSTPFARLTQLPQAQLKISITPRTAWRSGDTRVQLTTPPVAPSQPRRLRVFVE
RLATALQEANVSAVLRWDAPEQQQEAPMQALEYHISCWVGSSELHEELRLNQSALEARVEHLQPDQT
YHFQVEARVAATGAAAGAASHALHVAPEVQAVPRVLYANA EFIGELDLDRNRRLVHTASPVHEL
VGI EGQRLLWVNEHVELLTHVPGSAPAKLARMRAEVLALAVDWIQRIVYWAELDAPQAAIIYR
LDLCNFEGKILQGERVWSTPRGRLLKDLVALPQAQSLIWLEYEQGSPRNGSLRGRNLT D GSELEWA
TVQPLIRLHAGSLEPGSETLNLVDNQKLCVYDVARQLCTASALRAQLNLLGEDSIAGQLAQDSGY
LYAVKNWSIRAYGRRRQQL EYTVLEPEEVRLLQAHNYQAYPPKNCLLPPSSGGSLKATDCEEQR
CLLNLPMITASEDCPLPIPGVRYQLNLTARGPGSEEHDHGVEPLGQWLLGAGESLNLTDLLPFTR
YRVSGILSSFYQKLLALPTLVLAPLELLTASATPSPPRNF SVRVLSPRELEVSWLPPEQLRSESVY
YTLHWQQLDGENVQDRREWEAHERRELETAGTHRLTGIKPGSGYSLWVQAHATPTKSNSSERLHVR
SFAELPELQLELGPYSLSLTWAGTPDPLGSLQLECRSSAEQLRRNVAGNHTKMVVEPLQPRTRYQ
CRLLLGYAATPGAPLYHGTAEVYETLGDAPSQPGKPLEHIAEEVFRVWTWAARNGGAPIALYNLE
ALQARSDIRRRRRRRRRRNSGGSLQPLWAEPPVVVEDQWLDFCNTTELSCIVKSLHSSRLLLFRVR
ARSLHGWGPYSEESERVAEPFVSPKRGSLVLAIIAPAAIVSSCVLALVLRVKVQKRRLRACKLL
QQSRPISIWSNLSTLQTQQQLMAVRNRAFSTT LSDADIALLPQINWSQLKLLRFLGSGAFGEVYEGQ
LKTEDSEEPQRVAIKSLRKGA SEFAELLQEAQLMSNFKHENIVRLVIGICFDTESISLIMEHMEAGD
LLSYLRAARATSTQEPQPTAGLSLSELLAMCIDVANGCSYLEDMHFVHRDLACRNCLVTESTGSTD
RRRTVKIGDFGLARDIYKSDYYRKEGEGLLPVRWMPSPESLVDGLFTTQSDVWAFVGLCWEILTLGQ
QPYAARNNFVLAHVKEGGRLQPPMCTEKLYSLLLLCWRTDPWERPSFRRCYNTLHAISTDLRRT
QMASATADTVVSCSRPEFKVRFDGQPLEEHREHNERPEDENLTLREVPLKDKQLYANEGVSRL'
```

図 15. 7LES_DROME データに対する照会の例

HMMER ラッパーは、照会結果を、327 ページの表 61 に示すリレーショナル表に変換します。

表 61. HMMER ラッパーが連合システムに統合されると、HMMER は結果をリレーショナル表の形で戻す

Model	ModelScore	DomainNumber	DomainScore
pkinase	+3.041000000000000E+002	1	+3.041000000000000E+002
fn3	+1.763000000000000E+002	1	+4.900000000000000E+001
fn3	+1.763000000000000E+002	2	+1.360000000000000E+001
fn3	+1.763000000000000E+002	3	+1.620000000000000E+001
fn3	+1.763000000000000E+002	4	+6.350000000000000E+001
fn3	+1.763000000000000E+002	5	+1.460000000000000E+001
fn3	+1.763000000000000E+002	6	+1.940000000000000E+001
rrm	-4.450000000000000E+001	1	-4.450000000000000E+001

これでデータは完全に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。

関連概念:

- 145 ページの『表構造ファイルとは?』
- 159 ページの『Documentum とは?』
- 195 ページの『Excel とは?』
- 211 ページの『BLAST とは?』
- 239 ページの『XML とは?』
- 271 ページの『Entrez とは?』
- 297 ページの『Extended Search とは?』

HMMER の連合システムへの追加

手順:

HMMER データ・ソースを連合サーバーに追加するには、以下のようになります。

1. 適切なバージョンの hmmpfam 実行可能ファイル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。HMMER デーモンを構成します。
3. HMMER デーモンを開始します。HMMER デーモンを開始します。
4. CREATE WRAPPER ステートメントを実行してラッパーを登録します。CREATE WRAPPER ステートメントを実行してラッパーを登録します。
5. オプション: 照会のパフォーマンスを向上させるために DB2_DJ_COMM 環境変数を設定します。

6. CREATE SERVER ステートメントを実行してサーバーを登録します。CREATE SERVER ステートメントを実行してサーバーを登録します。
7. CREATE NICKNAME ステートメントを実行してニックネームを登録します。CREATE NICKNAME ステートメントを実行してニックネームを登録します。

ステートメントは、DB2 コマンド行プロセッサから実行できます。HMMER ラッパーを連合システムに追加した後、HMMER データ・ソースに対して照会を実行できません。

関連タスク:

- 328 ページの『正しいバージョンの hmmpfam 実行可能プログラムがインストールされていることの検査』

正しいバージョンの hmmpfam 実行可能プログラムがインストールされていることの検査

正しいバージョンの hmmpfam 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 を連合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。以下の手順で、ご使用の HMMER サーバーに最新バージョンの hmmpfam 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を検査してください。

手順:

現在の hmmpfam 実行可能プログラムのバージョンをチェックするには、次のようになります。

1. コマンド行から以下を実行し、出力ファイルに示されるバージョン番号を確認します。

```
hmmpfam -h
```

2. hmmpfam 実行可能プログラムが最新バージョン (HMMER 2.2 以上) 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。

ここでのタスクの順序では、次のタスクは『HMMER デーモンの構成』です。

関連タスク:

- 328 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 を連合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。HMMER ラッパーでは、HMMER デーモンが AIX ベースのマシン上で実行される必要があります。また、連合サーバーから TCP/IP を介して HMMER デーモンにアクセスできることが必要です。デーモンは、ラッパーと DB2

Universal Database とは別に実行され、ラッパーからの HMMER ジョブ要求を listen します。デーモン実行可能ファイル `db2hmmmer_daemon` は、HMMER サーバーのどのディレクトリーにあってもかまいません。

DB2 Universal Database のインストールの際、デーモン実行可能ファイルは、連合サーバーと同じコンピューターに置かれます。AIX プラットフォームでは、デーモン実行可能ファイルが置かれるディレクトリーは `/usr/opt/db2_08_01/bin` です。HMMER と連合サーバーを同じコンピューター上にインストールしなかった場合は、HMMER をインストールしたコンピューター上の任意のロケーションにデーモン実行可能ファイルをコピーする必要があります。また、構成ファイル `HMMER_DAEMON.config`、提供されている変換ユーティリティー `db2h2x`、およびシェル・スクリプト `db2runpfam.ksh` もコピーしてください。

プログラムは、実行可能ファイルでなければなりません。実行可能ファイルでない場合は、ターゲット・システム上で AIX コマンド `chmod a+x db2hmmmer_daemon db2h2x db2runpfam.ksh` を実行し、プログラムを実行可能ファイルにしてください。

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、`hmmpfam` 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。

HMMER デーモンは構成ファイルを必要とします。`HMMER_DAEMON.config` という名前のサンプル・デーモン構成ファイルが、ディレクトリー `DB2PATH/samples/lifesci` にあります。ここで、`DB2PATH` は DB2 Universal Database がインストールされているディレクトリーです。`HMMER_DAEMON.config` は、そのファイルのデフォルトの名前です。

構成ファイルをデーモンにアクセス可能な任意の場所へコピーします。必要であれば名前を変更します。そして、データ・ソースを処理するために編集します。デフォルトにより、HMMER デーモンは、開始された作業ディレクトリーの中での `blast_daemon` の構成情報を探します。

手順:

デーモンを構成するには、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 HMMER 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 デーモンでは、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、HMMER 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、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、デーモンが 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、ネットワーク接続経由でラッパーへ戻されます。この時点でデー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、提供されている db2runpfam.ksh シェル・スクリプトの完全修飾名です。

HMPFAM_PATH

これは、デーモンを実行するマシン上の hmmpfam 実行可能ファイルの完全修飾名です。

H2X_PATH

これは、デーモンに付属している db2h2x (HMMER から XML への) 変換プログラムの完全修飾名です。

データベース指定項目

プロファイル・データベースのロケーションを指定します。CREATE NICKNAME ステートメントでデータ・ソースのニックネームを作成する際は、構成ファイルの DATASOURCE オプションで使用しているエントリー名をメモしておいてください。デーモンが正常に機能するには、このエントリー名を指定する必要があります。

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、次のフォーマットで含まれている必要があります。

```
entry_name = path to profile_database
```

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
```

構成ファイルは改行文字で終わらなければなりません。

例:

次の例は、PFAM の必須オプションとプロファイル・データベース仕様を含むサンプル構成ファイルの内容を示すものです。

```
=
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
H2X_PATH=/home/user_id/sql1lib/bin/db2h2x
myhmms=/home/user_id/hmmer/tutorial/myhmms
pfam1s=/home/user_id/hmmer/pfam/Pfam_1s
```

1. 必ず、最初の行は等号にしてください。これをしないと、デーモンは開始しません。DAEMON_PORT を指定しない場合は、エラー・メッセージが戻されます。
2. 構成ファイルの最後の行は、必ず改行文字にしてください。最後が改行されていないと、最後の行にリストされているデータ・ソースを使用する HMMER 照会の実行時にエラー・メッセージが出されます。

ここでのタスクの順序では、次のタスクは『HMMER デーモンの開始』です。

関連タスク:

- 331 ページの『HMMER デーモンの開始』

HMMER デーモンの開始

HMMER デーモンの開始は、連合システムに HMMER 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:

まず、以下の条件が成立している必要があります。

- 現行ディレクトリーがデーモンのインストール・ディレクトリーである。
- デーモン構成ファイルの名前を変更していない。
- 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。

これらの条件が成立するなら、コマンド行で以下のコマンドを入力して HMMER デーモンを開始します。

```
db2hmmmer_daemon
```

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。

構成ファイルの名前やディレクトリーのロケーションを変更してしまった場合は、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に新しい名前あるいはロケーションを指示してください。

たとえば、以下のコマンドは、ラッパー・デーモンがその構成情報をサブディレクトリー `cfg` 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
```

ここでのタスクの順序では、次のタスクは『HMMER ラッパーの登録』です。

関連タスク:

- 332 ページの『HMMER ラッパーの登録』

HMMER ラッパーの登録

HMMER ラッパーの登録は、連合システムに HMMER 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。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、連合サーバーがデータ・ソースとの通信やデータ・ソースからのデータの検索に使用するメカニズムです。コンピューターは、ラッパーをライブラリー・ファイルとしてインストールします。

手順:

HMMER ラッパーを登録するには、`CREATE WRAPPER` ステートメントを実行します。

たとえば、デフォルト・ライブラリー・ファイル `libdb21shmmmer.a` から `my_hmmmer` という AIX 上の HMMER ラッパーを作成するには、次のステートメントをサブミットします。

```
CREATE WRAPPER my_hmmmer LIBRARY 'libdb21shmmmer.a'  
OPTIONS(DB2_FENCED 'N');
```

Windows では、libdb2lshmmr.a の代わりに db2lshmmr.dll を使用してください。

ここでのタスクの順序では、次のタスクは『HMMER ラッパーの DB2_DJ_COMM プロファイル変数の設定』です。

関連タスク:

- 333 ページの『HMMER ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE WRAPPER ステートメント』

HMMER ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定

HMMER ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定は、連合システムに HMMER を追加するという、より大きな作業の中のオプション・タスクです。HMMER データ・ソースにアクセスする際のパフォーマンスを向上させるためには、DB2_DJ_COMM DB2 プロファイル変数を設定します。この変数は、初期設定時に連合サーバーがラッパーをロードするかどうかを決定します。

手順:

DB2_DJ_COMM DB2 プロファイル変数を設定するには、関連する CREATE WRAPPER ステートメントに指定したラッパー・ライブラリーを使用して、db2set コマンドをサブミットします。

例:

```
db2set DB2_DJ_COMM=libdb2lshmmr.a,libdb2lshmmrF.a,libdb2lshmmrU.a
```

ここで、libdb2lshmmr.a、libdb2lshmmrF.a、および libdb2lshmmrU.a は、それぞれ、AIX プラットフォーム上のメイン・ライブラリー、fenced ライブラリー、unfenced ライブラリーの名前を表します。

等号 (=) の両側にスペースを入れないようにしてください。

データベース始動時にラッパー・ライブラリーをロードする際、不要なオーバーヘッドが発生するのを避けるため、アクセスする見込みのあるライブラリーだけを指定してください。

ここでのタスクの順序では、次のタスクは『HMMER データ・ソースのサーバーの登録』です。

関連タスク:

- 334 ページの『HMMER データ・ソースのサーバーの登録』

HMMER データ・ソースのサーバーの登録

HMMER データ・ソースのサーバーの登録は、連合システムに HMMER 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。

手順:

HMMER サーバーを連合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。

hmmpfam 実行可能ファイルとデーモン・インスタンスを使用する HMMER 検索を実行する各サーバーを登録してください。

たとえば、hmmpfam 検索に CREATE WRAPPER ステートメントを使用する my_hmmer というラッパーの場合は、次のステートメントで hmmer_server1 サーバーを登録します。

```
CREATE SERVER hmmer_server1
  TYPE pfam
  VERSION 2.2
  WRAPPER my_hmmer
  OPTIONS (NODE 'someserver.someschool.edu', DAEMON_PORT '4098')
```

引き数

TYPE 必須: 指定されているサーバーを使用して実行される検索のタイプを決定します。このリリースでは、値を PFAM に設定しておく必要があります。

VERSION

必須: 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 hmmpfam 実行可能ファイルのバージョン (HMMER 2.2 以上) と一致している必要があります。

WRAPPER

必須: CREATE WRAPPER ステートメントを実行したときに登録したラッパーの名前を指定します。

オプション

サーバー・オプション値は、単一引用符で囲む必要があります。

NODE 必須: 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。

DAEMON_PORT

オプション: デーモンが 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。デフォルトは 4098 です。

PROCESSORS

オプション: HMMER プログラムが使用するプロセッサの数を指定します。このオプションは、`hmmpfam` コマンドの `--cpu` オプションと同等です。例: `PROCESSORS '2'`。

HMMPFAM_OPTIONS

オプション: オプションと列名をマップする参照表に対応する列名を持たない、`--null2`、`--pvm`、`--xnu` などの `hmmpfam` オプションを指定します。例: `HMMPFAM_OPTIONS '--xnu --pvm'`。この例の場合は、`hmmpfam` オプションを使用する代わりに、SQL 照会の `WHERE` 文節で適切な列名を使用することもできます。

ここでのタスクの順序では、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:

- 335 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:

- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』

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、連合システムに HMMER 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` ステートメントを使用します。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。

HMMER の場合の `CREATE NICKNAME` ステートメントの構文は、次のようになります。

```
► CREATE NICKNAME—nickname—FOR SERVER—server-name—OPTIONS—(—————►
► DATASOURCE—'data_source_name'—[ ,—TIMEOUT—'timeout_duration' ]—————►
```

```
┌,—HMMTYPE—'hmmer_type'┐
└──────────────────────────┘
──────────────────────────┘
──────────────────────────┘
```

例:

```
CREATE NICKNAME myhmms
FOR SERVER hmmer_server
OPTIONS(DATASOURCE 'myhmms',TIMEOUT '60')
```

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。

DATASOURCE

必須: HMMER 検索を実行するデータ・ソースの名前。ここで使用される正確なストリングが、HMMER デーモンの構成ファイル内にも存在する必要があります。

TIMEOUT

オプション: HMMER ラッパーがデーモンからの結果を待機する最大時間 (分表示)。デフォルト値は 60 です。

HMMTYPE

オプション: モデルと遺伝子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。デフォルト値は PROTEIN です。

固定列

CREATE NICKNAME ステートメントは、自動的に固定列を作成します。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。固定列は、CREATE NICKNAME ステートメントの中には表れません。固定列には入力と出力の 2 つのタイプがあります。

入力固定列

入力固定列は、SQL 照会でパラメーター受け渡し述部として使用します。これらの固定列は、標準の hmmpfam オプションを HMMER に渡します。次に、これらの hmmpfam オプションを使用する指定されたデータ・ソースで HMMER が実行されます。なお、入力固定列は、結果表の一部として戻される照会選択リストの中でも参照できます。337 ページの表 62 は、入力固定列のリストです。

表 62. 入力固定列

名前	データ・タイプ	説明	使用できる演算子	hmmpfam オプション	戻り値
HmmQSeq	varchar(32000)	検索に使用される入力遺伝子列。	=		入力と同じ。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>	入力と同じ。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	入力と同じ。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。

表 62. 入力固定列 (続き)

名前	データ・タイプ	説明	使用できる演算子	hmmpfam オプション	戻り値
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	入力と同じ。デフォルトは 'N' です。

出力固定列

出力固定列は、照会結果表に戻され、述部として使用できます。表 63 は、出力固定列のリストです。

表 63. 出力固定列

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。

表 63. 出力固定列 (続き)

名前	データ・タイプ	説明
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。

ここでの作業は、これで終わりです。

関連資料:

- 339 ページの『HMMER データ・ソース - 完全指定の例』
- 339 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

以下の CREATE NICKNAME ステートメントは、ニックネーム myhmms を定義します。

```
CREATE NICKNAME myhmms
FOR SERVER hmmer_server
OPTIONS(DATASOURCE 'myhmms',TIMEOUT '60')
```

CREATE NICKNAME ステートメントを発行した後、ニックネーム myhmms を使用して連合システムを照会することができます。連合システムで、myhmms 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。

ここでの作業は、これで終わりです。

HMMER データ・ソース - 完全指定の例

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 hmmpfam 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。

制約事項:

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的です。

手順:

HMMER 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。

次の完全指定の例は、HmmQSeq を検索シーケンスとして使用する照会を作成して実行するのに必要なすべてのステートメントを示しています。

```
CREATE WRAPPER hmmer_wrapper
LIBRARY 'libdb2lshmmer.a'
OPTIONS (DB2_FENCED 'N');

CREATE SERVER hmmer_serv
TYPE pfam VERSION 2.2
WRAPPER hmmer_wrapper
OPTIONS(NODE 'HMMERserv.MyCompany.com');

CREATE NICKNAME myhmms
FOR SERVER hmmer_serv
OPTIONS(DATASOURCE 'myhmms', TIMEOUT '1');

-- Run the 7LES_DROME gene sequence on the myhmms 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 myhmms
WHERE HmmQSeq =
      'MTMFWQQNVVDHQSDEQDKQAKGAAPT KRLNISFNVKIAVNVNTKMTTTHINQQAPGTSS...';
```

関連タスク:

- 335 ページの『HMMER データ・ソースのニックネームの登録』
- 341 ページの『サンプルを使用した新規 HMMER 照会の構成』

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、各ニックネームに使用される名前は、HMMER 検索およびデータ・ソースのタイプを記述します。こうして記述名を使用することにより、各サンプル照会で登録ステートメントをリストする必要がなくなります。また、いくつかの例は、他のデータ・ソースと結合した場合の HMMER の動作を示すものとなっています。

照会 1

```
SELECT Model, ModelScore, ModelEValue, DomainNumber, DomainScore, DomainEValue
FROM myhmms
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 myhmms の hmmpfam 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 HMMER 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 Model, ModelScore, ModelEValue
FROM myhmms
WHERE HmmQSeq = 'MTMFWQQNVDHQSDEQDKQAKGAAPT KRLNISFNVKIAVNVNTKMTTTHINQ... '
AND ModelScore > 0
```

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 myhmms 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 (337 ページの表 62 を参照) を渡します。ラッパーは、SELECT の後にリストされた 3 つの列を戻します。

照会 3

```
SELECT Model, DomainNumber, DomainScore, DomainEValue
FROM myhmms
WHERE HmmQSeq = 'MTMFWQQNVDHQSDEQDKQAKGAAPT KRLNISFNVKIAVNVNTKMTTTHINQ... '
AND ModelEValue < 1
ORDER BY DomainScore DESC
```

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 myhmms 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 (337 ページの表 62 を参照) を渡します。ラッパーは、SELECT の後にリストされている 4 つの列を戻し、結果を DomainScore 順に、高いほうからソートします。

関連タスク:

- 281 ページの『Entrez データ・ソース — 照会の例』

関連資料:

- 181 ページの『Documentum データ・ソース - 照会の例』
- 199 ページの『Excel データ・ソース - 照会の例』
- 312 ページの『Extended Search ラッパー - 照会の例』

HMMER ラッパーの最適化のヒント

同一サーバー上でラッパーとデーモンの両方を実行すると、ネットワーク通信のボトルネックの可能性を除去することができます。

関連タスク:

- 333 ページの『HMMER ラッパーの DB2_DJ_COMM DB2 プロファイル変数の設定』

関連資料:

- 153 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』
- 234 ページの『BLAST ラッパーの最適化のヒント』

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 64. 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とその列に対して有効であるかどうか検査してください (337 ページの表 62 を参照)。

表 64. HMMER ラッパーのメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	Unexpected error code "Unspecified Error" received from data source "Hmmer wrapper". Associated text and tokens are "Unable to resolve NODE host name".	CREATE SERVER で指定されている TCP/IP ノード名は無効です。
SQL1822N	Unexpected error code "Unspecified Error" received from data source "Hmmer wrapper". 関連するテキストおよびトークンは "Unable to connect to daemon" です。	hmmer_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	Unexpected error code "Unspecified Error" received from data source "Hmmer wrapper". 関連するテキストおよびトークンは "Unknown error from the hmmer daemon" 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ていない可能性があります。
SQL1822N	Unexpected error code "Unspecified Error" received from data source "Hmmer wrapper". Associated text and tokens are "FATAL: No such option "--cut_TC".	CutMode 述部は小文字で指定する必要があります。例: WHERE CutMode = 'tc'

関連資料:

- 154 ページの『表構造ファイル・ラッパーのメッセージ』
- 186 ページの『Documentum ラッパーのメッセージ』
- 203 ページの『Excel ラッパーのメッセージ』
- 234 ページの『BLAST ラッパーのメッセージ』
- 259 ページの『XML ラッパーのメッセージ』
- 291 ページの『Entrez ラッパーのメッセージ』
- 317 ページの『Extended Search ラッパーのメッセージ』

第 20 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ための ALTER NICKNAME ステートメントの使用法を説明します。

ニックネームの変更

ALTER NICKNAME ステートメントを使用して、データ・ソースまたはビューの連合データベースの表現を変更することができます。

制約事項:

ALTER NICKNAME ステートメントは、BLAST、Documentum、ラッパーのいずれの列名を変更するためにも使用することはできません。ALTER NICKNAME ステートメントは、表構造ファイルおよび XML ラッパーの列名を変更するために使用できます。

手順:

ニックネーム列値を変更するには、ALTER NICKNAME ステートメントを使用して、以下のことを行う必要があります。

- これらの列のローカル・データ・タイプの変更
- これらの列のオプションの追加、変更、または削除

関連タスク:

- 345 ページの『データ・タイプの変更』
- 346 ページの『ニックネーム・オプションの変更』

データ・タイプの変更

ALTER NICKNAME ステートメントを使用して、列のデータ・タイプを変更することができます。

手順:

列のデータ・タイプを変更するには、ALTER NICKNAME ステートメントを使用します。

たとえば、以下の ALTER NICKNAME ステートメントは、DRUG 列のローカル・データ・タイプを CHAR(30) に変更します。DRUG 列は最初は、CREATE NICKNAME ス

ステートメントを使用して CHAR(20) として定義されていました。ニックネーム DRUGDATA1 は、drugdata1.txt というローカルの表構造ファイルです。

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

関連タスク:

- 345 ページの『ニックネームの変更』
- 346 ページの『ニックネーム・オプションの変更』

ニックネーム・オプションの変更

ALTER NICKNAME ステートメントを使用して、ニックネーム・オプションを変更することができます。

手順:

ニックネーム・オプションを変更するには、ALTER NICKNAME ステートメントを使用します。

たとえば、以下の ALTER NICKNAME ステートメントは、表構造ファイル drugdata1.txt の完全修飾パスを変更します。パスは最初は、CREATE NICKNAME ステートメントを使用して '/user/pat/drugdata1.txt' として定義されていました。ニックネーム DRUGDATA1 は、drugdata1.txt というローカルの表構造ファイルです。

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

関連タスク:

- 345 ページの『ニックネームの変更』
- 345 ページの『データ・タイプの変更』

第 21 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、まずステートメントのアルファベット順に、そしてラッパーのアルファベット順に列挙されています。

CREATE FUNCTION ステートメント構文 - Extended Search ラッパー

```
► CREATE FUNCTION ESWRAPPER.ES_SEARCH (—INTEGER—, —VARCHAR(1024)—) ►
► RETURNS INTEGER AS TEMPLATE ◀
```

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)

参照式を定義します。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ストリングを指定する必要があります。

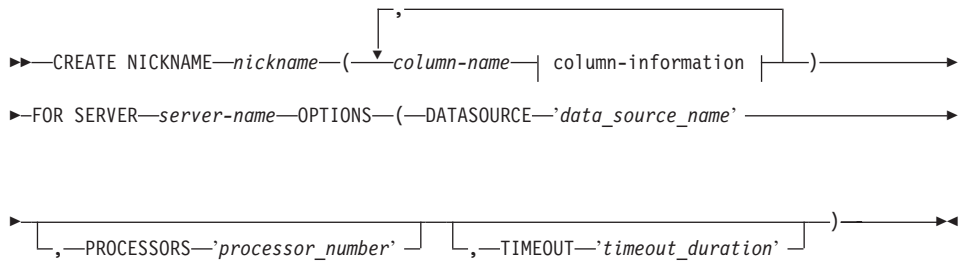
関連タスク:

- 309 ページの『Extended Search カスタム関数の登録』

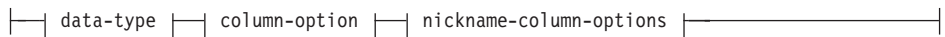
関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE FUNCTION (ソースまたはテンプレート) ステートメント』
- 312 ページの『Extended Search ラッパー - 照会の例』
- 315 ページの『Extended Search ラッパー - 汎用照会言語』

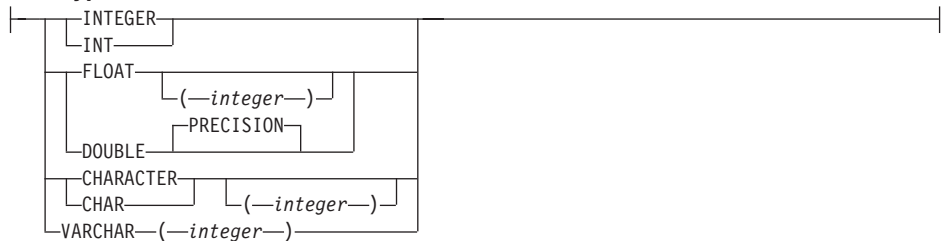
CREATE NICKNAME ステートメント構文 - BLAST ラッパー



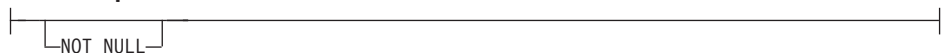
column-information:



data-type:



column-option:



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。

INDEX このオプションが定義行列のグループの中で表示される列の序数。このオプションは必須です。

DELIMITER

このオプションが表示される列の定義行情報の終点を決定するために使用する

べき区切り文字。このオプションの値に複数の文字がある場合は、それらの文字のうちのいずれかが最初に現れる位置がこのフィールドの情報の終端を表します。デフォルトは行の終わりです。このオプションは必須です。ただし、その列に定義行の残りを入れたい場合に指定した最後の列を除きます。

DEFAULT

以下の入力固定列の新規デフォルト値を指定します。

- E_value
- QueryStrands
- GapAlign
- NMisMatchPenalty
- NMatchReward
- Matrix
- FilterSequence
- NumberOfAlignments
- GapCost
- ExtendedGapCost
- WordSize
- ThresholdEx

この新しい値は事前設定の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、与えられた列に示された値と同じタイプのものでなければなりません。このオプションは任意です。

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。

DATASOURCE

BLAST 検索実行の対象のデータ・ソース名。ここで使用される正確なストリングが BLAST デーモンの構成ファイルに存在している必要があります。このオプションは必須です。

PROCESSORS

BLAST 照会の評価時に使用するプロセッサの数を指定します。これは、BLAST の `blastall -a` オプションに対応します。このオプションは任意です。デフォルト値は 1 です。

TIMEOUT

BLAST ラッパーがデーモンからの結果を待機する最大時間 (分表示)。デフォルトは 60 です。このオプションは任意です。

関連タスク:

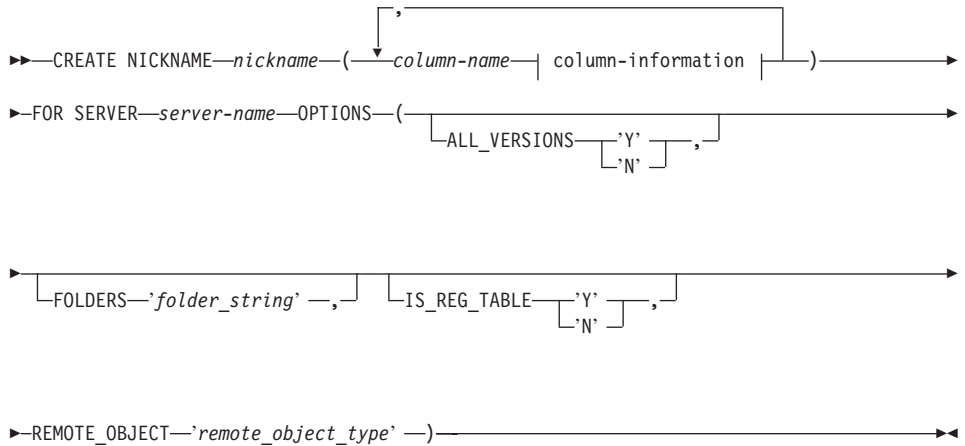
- 217 ページの『BLAST デーモンの構成』
- 223 ページの『BLAST データ・ソースのニックネームの登録』

関連資料:

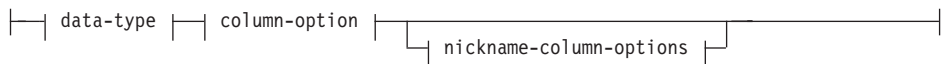
- 『SQL リファレンス 第2巻』の『CREATE NICKNAME ステートメント』
- 230 ページの『CREATE NICKNAME ステートメント - BLAST ラッパーの例』

CREATE NICKNAME ステートメント構文 - Documentum ラッパー

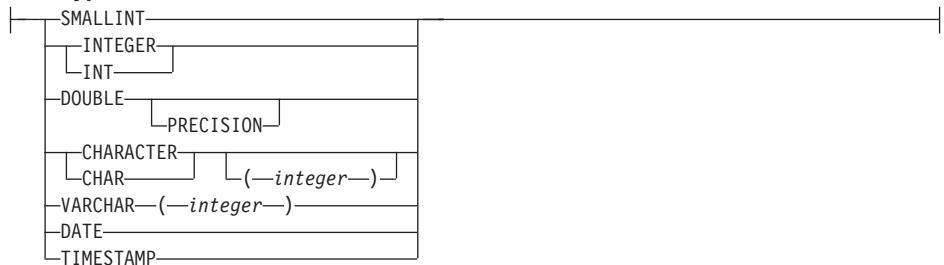
Documentum の CREATE NICKNAME ステートメントの構文は以下のとおりです。



column-information:



data-type:



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

全オブジェクト・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 ステートメントによってこのオプションを変更することにより、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・タイプかまたは登録済みの表になります。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:

- 168 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』
- 172 ページの『CREATE NICKNAME ステートメント - Documentum ラッパーの例』

CREATE NICKNAME ステートメント構文 - Excel ラッパー

```
▶ CREATE NICKNAME nickname ( column-name | data-type | column-option )
▶ )
  FOR SERVER server-name OPTIONS ( ( FILE_PATH 'path'
  , RANGE 'range' ) )
```

data-type:

```
|
|  INTEGER
|  |
|  |  INT
|  |
|  FLOAT
|  |
|  |  ( integer )
|  |
|  VARCHAR ( integer )
|  DATE
|
```

column-option:

```
|
|  NOT NULL
|
```

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、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。

右下指定は範囲から省略できます。この場合、右下の有効な行が使用されます。左上の値が省略される場合、その値は *A1* を取ります。範囲がスプレッドシートの中に実際にあるよりも多くの行を指定する場合、実際の行数が使用されます。

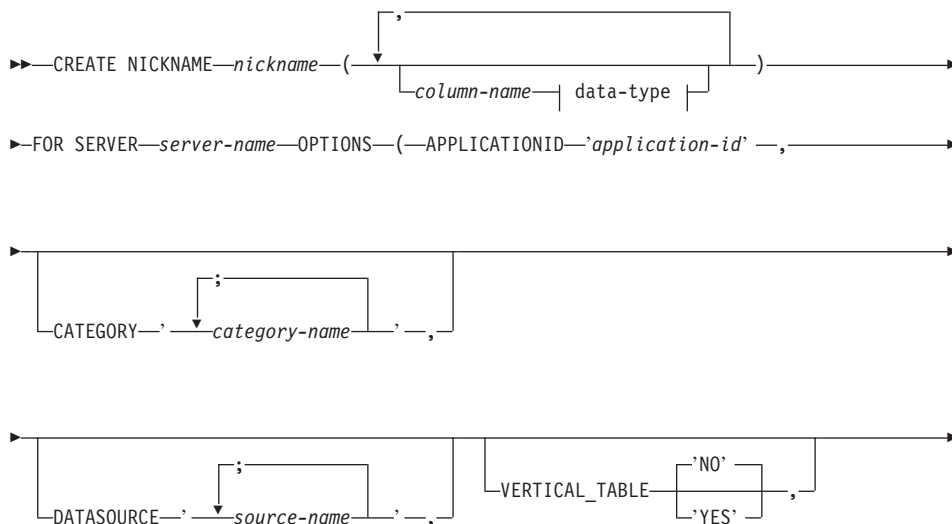
関連タスク:

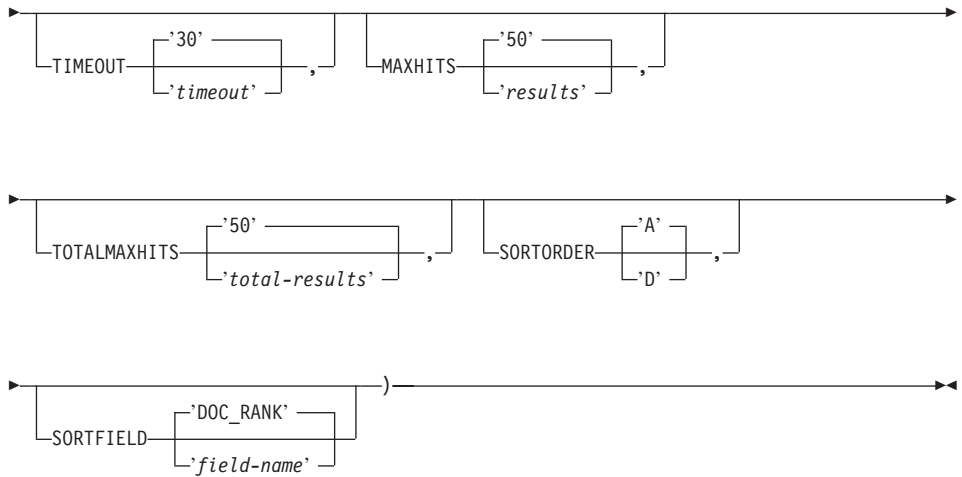
- 199 ページの『Excel データ・ソースのニックネームの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE NICKNAME ステートメント』

CREATE NICKNAME ステートメント構文 - Extended Search ラッパー





data-type:



NICKNAME

この Extended Search ニックネーム表の固有の名前を指定します。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

検索したい 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する必要があります。このパラメーターは必須です。

CATEGORY

検索したい 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・ソース名を指定する必要があります。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する必要があります。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。

MAXHITS

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。

TOTALMAXHITS

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。ラッパーはこれらの結果を単一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、つまり昇順 (A) か降順 (D) を指定します。デフォルト値は A です。

SORTFIELD

検索結果をソートする必要のある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かめてください。

関連概念:

- 300 ページの『Extended Search のニックネーム』
- 302 ページの『Extended Search の垂直表』

関連タスク:

- 307 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』
- 312 ページの『Extended Search ラッパー - 照会の例』

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。この名前は、ニックネームのスキーマおよび NCBI データベース、さらに他のニックネームとの関係を決定します。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:

- 276 ページの『Entrez データ・ソースのニックネームの登録』

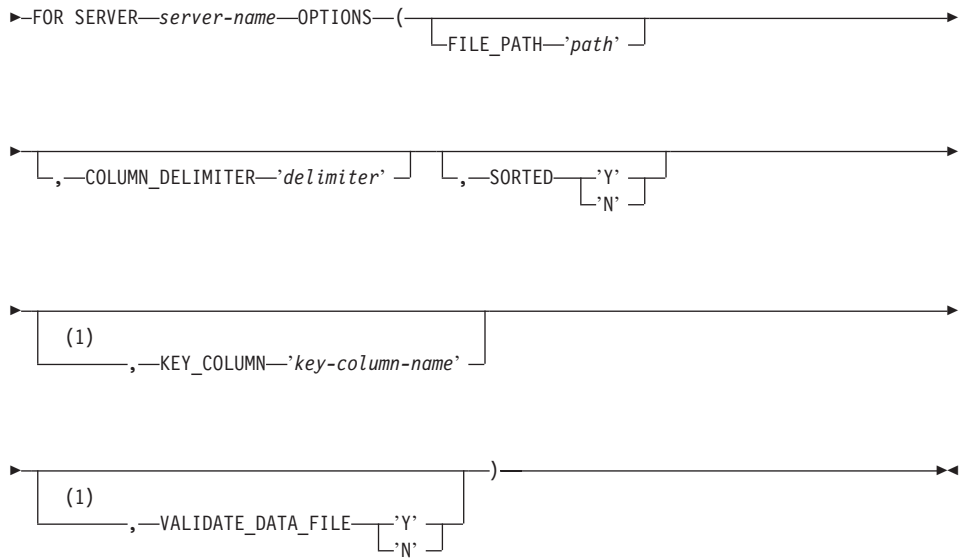
関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE NICKNAME ステートメント』

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。

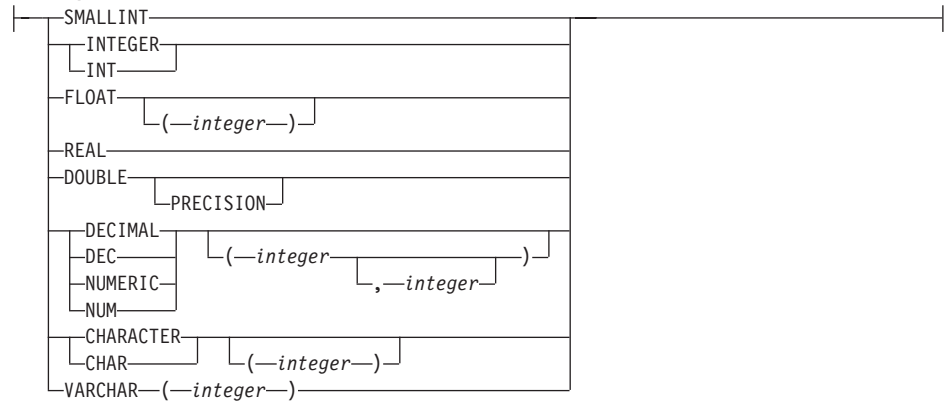
```
►► CREATE NICKNAME nickname ( column-name | column-information | )
```



column-information:



data-type:



column-option:



nickname-column-options:

|—OPTIONS—(—DOCUMENT—'FILE' —)|

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。

nickname

アクセスされる表構造ファイルの固有のニックネーム。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、VARCHAR、SMALLINT、INTEGER、FLOAT、DOUBLE、REAL、および DECIMAL です。

SMALLINT

短整数。

INTEGER または INT

長精度整数。

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。

REAL 単精度浮動小数点。

DOUBLE または DOUBLE PRECISION

倍精度浮動小数点。

FLOAT 倍精度浮動小数点。

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。

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。

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ているのか、ソートされていないのかを指定します。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 各国語サポート区分の設定によって定義されている現在の地域の照合順序に従って、昇順でソートされます。

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)。単一の桁区切り文字のみが許可されています。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しています。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。NULL フィールドが行の末尾にある場合、区切り文字に行終了文字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保管することができません。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

ファイルをソートする場合のキーとなる、ファイル内の列の名前 (単一引用符で囲まれる)。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によって指定された列は、キー列として指定しないでください。

単一系列のキーのみがサポートされます。複数列キーは使用できません。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 にすることができないことを指定されることをお勧めします。

このオプションは大文字小文字の区別をします。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、チェックは行われません。

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:

- 150 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』
- 152 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』

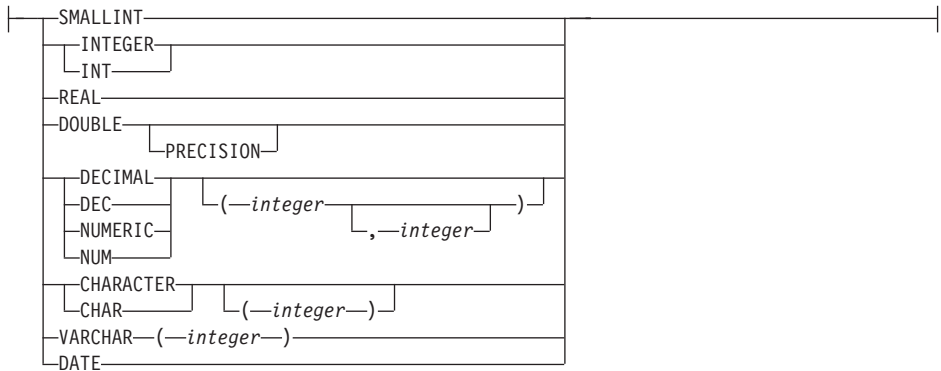
CREATE NICKNAME ステートメント構文 - XML ラッパー

```
►► CREATE NICKNAME nickname ( column-name | Column structure | ) ►►
► FOR SERVER server-name OPTIONS ( ( FILE_PATH 'path' —, —
                                     DIRECTORY_PATH 'path' —, — ) )
► XPATH 'xpath_expression' | Nickname parameters | ) ►►
```

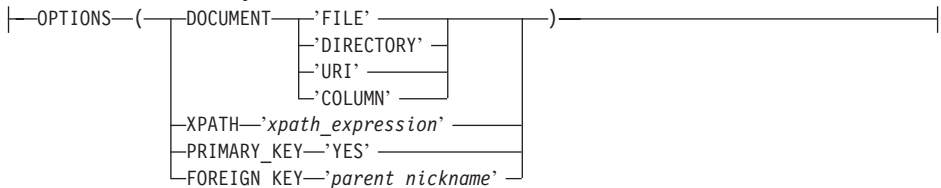
Column structure:

```
| Data type options | [ NOT NULL ] | Nickname column options |
```

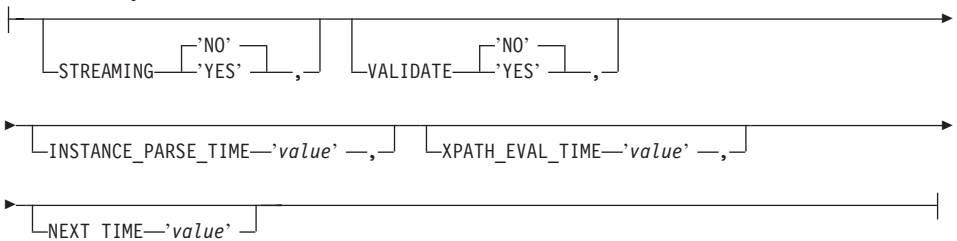
Data type options:



Nickname column options:



Nickname parameters:



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。このオプションを使用して、複数の XML ソース・ファイルに関する単一のニックネームを作成します。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せん。

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・データのタイプを表します。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 オプションに関連する列は 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 文書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・ロケーションを表します。

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。ニックネームは、多くても 1 つの **PRIMARY_KEY** 列オプションしかもてません。'YES' が唯一の有効な値です。このオプションによって指定された列は、ラッパーが生成したキーを保持します。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にだけ使用できます。

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。ニックネームは、多くても 1 つの **FOREIGN_KEY** 列オプションしかもてません。このオプションの値は、大文字小文字の区別をします。このオプションによって指定された列は、ラッパーが生成したキーを保持します。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にだけ使用できます。

親ニックネームが別のスキーマ名をもつ場合、**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・ドキュメント (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 に設定する場合、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 に設定する場合、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、大容量または複雑な 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、大容量または複雑な 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ルートではない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・エレメントの位置指定を行うのに必要な時間 (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、大容量または複雑な 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ルートではない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 に設定する場合、エラー・メッセージを受け取ります。

関連タスク:

- 251 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』
- 251 ページの『CREATE NICKNAME ステートメント - XML ラッパーの例』

CREATE SERVER ステートメントの引き数 - BLAST ラッパー

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。

オプション

NODE BLAST デーモン・プロセスを実行中の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。

DAEMON_PORT

BLAST ジョブ要求をデーモンが listen するネット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。デフォルトは 4007 です。このオプションは任意です。

関連タスク:

- 223 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』

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:

- 166 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 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:

- 275 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』

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。

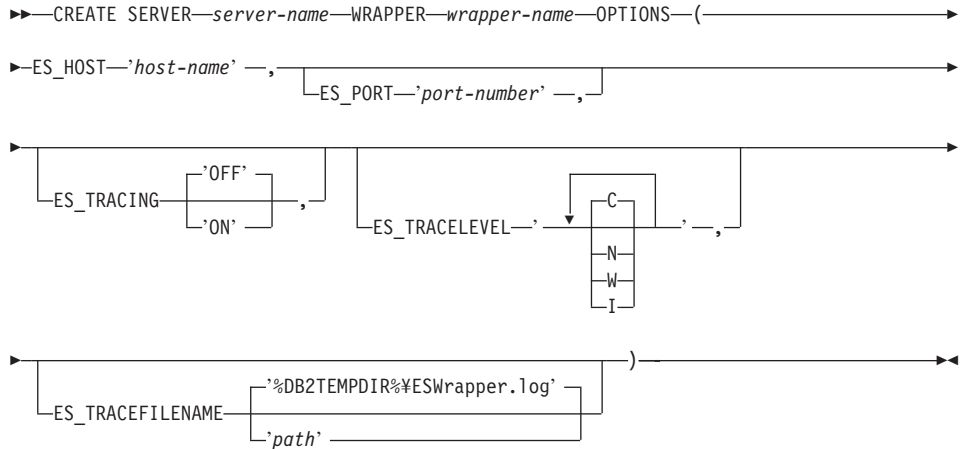
関連タスク:

- 198 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 SERVER ステートメント構文 - Extended Search ラッパー



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い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

検索したい 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対するトレースを使用可能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

トレースが使用可能になっている場合、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、独立して使用可能および使用不可にできます。

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

トレースが使用可能になっている場合、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する場合、デフォルト値は DB2 temp ディレクトリー (%DB2TEMPDIR%\ESWrapper.log または %DB2TEMPDIR%\ESWrapper.log) にある ESWrapper.log ファイルです。

関連タスク:

- 306 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:

- 167 ページの『ユーザーのマッピング (Documentum ラッパー)』

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 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・ソースに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:

- 308 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 WRAPPER ステートメント構文 - Extended Search ラッパー

```
▶—CREATE WRAPPER—wrapper-name—LIBRARY—'library-name' —▶
```

WRAPPER

この Extended Search ラッパーの固有の名前を指定します。

LIBRARY

以下のプラットフォームに依存するライブラリー名の 1 つを指定します。

- Windows: db2uies.dll
- AIX: libdb2uies.a

関連タスク:

- 305 ページの『Extended Search 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』

付録 A. 連合情報を含むグローバル・カタログ表内のビュー

連合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、および 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、連合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 (Linux、UNIX、および Windows 版) 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 操作を実行すると、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、連合情報が入っている 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 65. 通常、連合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.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.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	連合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。

表 65. 通常、連合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.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。
SYSCAT.REVTYPEMAPPINGS	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、ローカル DB2 データ・タイプからデータ・ソースのデータ・タイプへの対応付け。これらのマッピングは、リモートの (意識されない) 表にのみ使用される。
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.TABLES	作成したそれぞれのローカル DB2 表、連合ビュー、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、DB2 SQL ステートメントを使用してデータ・ソースを照会する時に使用される。
SYSCAT.USEROPTIONS	連合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	作成したローカル連合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、連合情報が入っている 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せん。

表 66. 更新可能な、連合グローバル・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ない。
SYSSTAT.FUNCTIONS	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ない。
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.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、同義語、および別名の情報は含まれていない。タイプされた表の場合、表の階層のルート表のみがビューに含まれる。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ない。

付録 B. 連合システムのサーバー・オプション

サーバー・オプションは、CREATE SERVER ステートメントでデータ・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・オプションの中には、すべてのデータ・ソースについて使用可能なものと、データ・ソース固有のものがあります。

非リレーショナル・ラッパーには、非常に独特のサーバー・オプションがあり、それらについてはデータ・ソース構成情報の中で文書化されています。

リレーショナル・データ・データ・ソースに共通の連合サーバー・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
- 日付時刻オプション。DATEFORMAT、TIMEFORMAT、TIMESTAMPFORMAT
- ロケーション・オプション。CONNECTSTRING、DBNAME、IFILE
- セキュリティ・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

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サーバー・オプションのリストを、次の表に示します。

表 67. 使用可能なサーバー・オプション

データ・ソース	COLLATING_SEQUENCE	COMM_RATE	CONNECTSTRING	CPU_RATIO	DATEFORMAT	DB2_MAXIMAL_PUSHDOWN	DBNAME	FOLD_ID	FOLD_PW	IFILE	IGNORE_UDT	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 for iSeries	X	X		X		X	X	X	X				X	X				X		X				X
DB2 for z/OS and OS/390	X	X		X		X	X	X	X				X	X				X		X				X
DB2 for VM and VSE	X	X		X		X	X	X	X				X	X				X		X				X
DB2 for Linux、DB2 for UNIX、および DB2 for Windows	X	X		X		X	X	X	X				X	X				X		X				X
Informix	X	X		X		X	X	X	X		X	X	X	X		X		X		X				
Microsoft SQL Server	X	X		X		X	X	X	X			X	X	X		X		X		X				
ODBC	X	X		X	X	X	X	X	X			X	X	X		X		X		X		X	X	
OLE DB	X		X			X																		
Oracle	X	X		X		X	X	X	X			X			X		X	X	X	X				X
Sybase	X	X		X		X	X	X	X	X		X		X	X	X	X	X	X	X	X			
Teradata	X	X		X		X						X	X		X					X				

次の表に、各サーバー・オプションの説明と、それぞれの有効な設定値およびデフォルト設定値を示します。

表 68. サーバー・オプションとその設定値

オプション	説明と有効な設定値	デフォルト の設定値
COLLATING_SEQUENCE	<p>データ・ソースが連合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する。</p> <p>'Y' データ・ソースは DB2 連合データベースと同じ照合シーケンスを持つ。</p> <p>'N' データ・ソースは DB2 連合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つ。</p> <p>'I' データ・ソースは DB2 連合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ない (例: 'STEWART' と 'StewART' は等しいと見なされる)。</p>	'N'
COMM_RATE	<p>連合サーバーとデータ・ソース・サーバー間の通信レートを指定する。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、1x1023 より小さい値です。値は、任意の有効な倍精度表記 (例: 123E10、123、または 1.21E4 など) で表現できます。</p> <p>値は整数または浮動小数点数。</p>	'2'
CONNECTSTRING	<p>OLE DB Provider への接続に必要な初期化プロパティを指定する。</p>	なし
CPU_RATIO	<p>データ・ソースの CPU が連合サーバーの CPU より、どれほど速いまたは遅いかを表す。</p> <p>有効な値は 0 より大きく、1x1023 より小さい値です。値は、任意の有効な倍精度表記 (例: 123E10、123、または 1.21E4 など) で表現できます。</p> <p>値は整数または浮動小数点数。</p> <p>1 に設定した場合、それは DB2 連合の CPU 速度とデータ・ソースの CPU 速度が同じであるということ (1:1 の比) です。0.5 に設定した場合、それは DB2 連合の CPU 速度が、データ・ソースの CPU 速度の 50% の遅い速度であるということです。2 に設定した場合、それは DB2 連合の CPU 速度が、データ・ソースの CPU 速度の 2 倍の速い速度であるということです。</p>	'1.0'

表 68. サーバー・オプションとその設定値 (続き)

オプション	説明と有効な設定値	デフォルト の設定値
DATEFORMAT	<p>データ・ソースで使用されている日付の形式。形式は、'DD'、'MM'、および 'YY' または 'YYYY' を使用して日付の数値形式を表記することにより指定します。スペースやコンマなどの区切り文字も指定してください。たとえば、'2003-01-01' の日付形式は 'YYYY-MM-DD' になります。このフィールドは、NULL 可能です。</p>	なし
(この表の最後にある注 5 を参照)		
DB2_MAXIMAL_PUSHDOWN	<p>照会オプティマイザーがアクセス・プランを閉じる際に使用される主要な基準。照会オプティマイザーは、リモート・データ・ソースが可能な限り多くの照会処理を実行できるよう、コストまたはユーザー要件に基づいてアクセス・プランを選択できます。</p> <p>'Y' 照会オプティマイザーは、他のプランよりも多くの照会操作をデータ・ソースにプッシュダウンすることになるアクセス・プランを選択します。複数のアクセス・プランのプッシュダウン量が同じなら、照会オプティマイザーはコストが最低のプランを選択します。</p> <p>連合サーバー上のマテリアライズ照会表 (MQT) が照会の一部または全部を処理できる場合には、マテリアライズ照会表を含むアクセス・プランが使用されます。結果がカルテシアン積になる照会は連合データベースによって処理され、プッシュダウンされません。</p> <p>'N' 照会オプティマイザーは、コストに基づいてアクセス・プランを選択します。</p>	'N'
DBNAME	<p>連合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 の場合、この値は 1 つのインスタンス内の特定のデータベースに対応するか、または DB2 for z/OS and OS/390 では、データベースの LOCATION 値に対応します。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。</p>	なし。

表 68. サーバー・オプションとその設定値 (続き)

オプション	説明と有効な設定値	デフォルト の設定値
FOLD_ID	<p>連合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 に適用される。有効な値は以下のとおり。</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' 連合サーバーは、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' 連合サーバーは、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' 連合サーバーは、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、連合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>連合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る。有効な値は以下のとおり。</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' 連合サーバーは、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' 連合サーバーは、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' 連合サーバーは、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、連合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 68. サーバー・オプションとその設定値 (続き)

オプション	説明と有効な設定値	デフォルト の設定値
IFILE	Sybase Open Client インターフェース・ファイルのパスと名前を指定する。Windows NT 連合サーバーでのデフォルトは %DB2PATH%\%interfaces。UNIX 連合サーバーでのデフォルトのパスと名前値は、 \$DB2INSTANCE/sqlib/interfaces。	なし。
IGNORE_UDT	<p>CTLIB または DBLIB ラッパーを使用してアクセスされるデータ・ソースでのユーザー定義タイプ (UDT) を、連合サーバーで使用するかどうか。有効な値は以下のとおり。</p> <p>'Y' UDT のユーザー定義の指定を無視する。</p> <p>'N' UDT のユーザー定義の指定を無視しない。</p>	'N'
INFORMIX_LOCK_MODE	<p>Informix データ・ソースについてロック・モードを設定することを指定する。Informix ラッパーは、Informix データ・ソースとの接続確立直後に 'SET LOCK MODE' コマンドを発行します。有効な値は以下のとおり。</p> <p>'W' Informix ロック・モードを WAIT に設定します。ロックされている表または行にラッパーがアクセスしようとする、Informix はロックが解放されるまで待機します。</p> <p>'N' Informix ロック・モードを NOWAIT に設定します。ロックされている表または行にラッパーがアクセスしようとする、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。ロックされている表または行にラッパーがアクセスしようとした際に、指定された秒数内にロックが解放されないなら、Informix はエラーを戻します。</p>	'W'

表 68. サーバー・オプションとその設定値 (続き)

オプション	説明と有効な設定値	デフォルト の設定値
IO_RATIO	<p>データ・ソースの入出力システムが連合サーバーの入出力システムより、どれほど速いまたは遅いかを表す。</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、任意の有効な倍精度表記 (例: 123E10、123、または 1.21E4 など) で表現できます。</p> <p>1 に設定した場合、それは DB2 連合の入出力速度とデータ・ソースの入出力速度が同じであるということ (1:1 の比) です。0.5 に設定した場合、それは DB2 連合の入出力速度が、データ・ソースの入出力速度の 50% の遅い速度であるということです。2 に設定した場合、それは DB2 連合の入出力速度が、データ・ソースの入出力速度の 2 倍の速い速度であるということです。</p>	'1.0'
IUD_APP_SVPT_ ENFORCE	<p>DB2 連合システムが、アプリケーションの保管点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。 SET SERVER OPTION ステートメントを使用してこのサーバー・オプションを設定する場合、静的 SQL ステートメントでは無効です。</p> <p>'Y' エラーが発生し、データ・ソースでアプリケーション保管場所ステートメントが実行されるようになっていないなら、連合サーバーは挿入、更新、または削除の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 (SQL1476) が戻されます。</p> <p>'N' エラーが検出されても、連合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、アプリケーション側で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 連合サーバーが待つ秒数を指定する。デフォルト値は TIMEOUT と同じです。</p>	'0'
NODE	<p>データ・ソースを RDBMS へのインスタンスとして定義するための名前。</p>	なし。

表 68. サーバー・オプションとその設定値 (続き)

オプション	説明と有効な設定値	デフォルト の設定値
PACKET_SIZE	<p>Sybase インターフェース・ファイルのバケット・サイズをバイト単位で指定する。指定したバケット・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 (例: 大きな表に行を挿入する場合など) にバケット・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。</p>	
PLAN_HINTS	<p>プラン・ヒント を使用可能にするかどうかを指定する。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・オプティマイザに対する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 データ・ソースの場合のみ使用可能です。</p>	'N'
PUSHDOWN	<p>'Y' DB2 はデータ・ソースに操作を評価させることを考慮します。</p> <p>'N' DB2 は、データ・ソースに対して、列名を指定した SELECT しか含まれていない SQL ステートメントを送信します。データ・ソースに送られる SQL には、述部 (WHERE= など) 列とスカラー関数 (MAX や MIN など)、ソート (ORDER BY や GROUP BY など)、および結合が含まれません。</p>	'Y'
TIMEFORMAT	<p>データ・ソースで使用されている時刻の形式。 'hh12'、 'hh24'、 'mm'、 'ss'、 'AM'、または 'A.M' を使用して入力します。たとえば、 '16:00:00' の場合の時刻形式は 'hh24:mm:ss' になります。 '8:00:00 AM' の時刻形式は 'hh12:mm:ss AM' です。このフィールドは、NULL 可能です。</p>	なし

表 68. サーバー・オプションとその設定値 (続き)

オプション	説明と有効な設定値	デフォルト の設定値
TIMESTAMPFORMAT (この表の最後にある注 5 を 参照)	データ・ソースで使用されるタイム・スタンプの形式。日付と時刻の形式の後、'n' (0.1 秒単位)、'nn' (0.01 秒単位)、'nnn' (ミリ秒単位)、そして最大で 'nnnnnn' (マイクロ秒単位) を指定できます。たとえば、'2003-01-01-24:00:00.000000' の場合のタイム・スタンプの形式は、'YYYY-MM-DD-hh24:mm:ss.nnnnnn' です。このフィールドは、NULL 可能です。	なし
TIMEOUT	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 連合サーバーが待つ秒数を指定する。 <i>seconds</i> 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は 0 であり、これは DB2 が無期限に応答を待つことを意味します。	'0'

表 68. サーバー・オプションとその設定値 (続き)

オプション	説明と有効な設定値	デフォルト の設定値
VARCHAR_NO_ TRAILING_BLANKS	<p>このオプションは、データ・ソースに変長長の文字データ・タイプがあり、その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。</p> <p>ある種のデータ・ソース (Oracle など) の文字比較セマンティクスは、ブランク埋め込み文字比較のセマンティクス、つまり DB2 for Linux、DB2 for UNIX、および DB2 for Windows の比較セマンティクスと同じ結果を戻すセマンティクスではありません。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せん。</p> <p>DB2 ファミリー・データ・ソースの場合、有効な設定値は 'Y' のみです。</p> <p>'Y' Yes - これらの VARCHAR 列に末尾ブランクはない。</p> <p> このデータ・ソースの文字比較セマンティクスは、ブランク埋め込み文字比較セマンティクスであり、それは連合サーバーとは異なります。文字比較操作は連合サーバー上で処理され、データ・ソースにはプッシュダウンされません。</p> <p>'N' No - これらの VARCHAR 列には末尾ブランクがある。</p> <p> このデータ・ソースの文字比較セマンティクスは、ブランク埋め込み文字比較セマンティクスであり、それは連合サーバーの文字比較セマンティクスと類似しています。文字比較操作は、データ・ソースにプッシュダウンされて処理されます。</p>	'N'

この表に関する注

1. 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。
2. DB2 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。
3. 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ので、大文字小文字の区別は意味をなしません。

4. 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 は複数回試行することになるので、便利に思えるかもしれませんが、パフォーマンスが悪化する可能性があります (DB2 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)。
5. このオプションが使用されるのは、 SERVER_TYPE の値が GENERIC である場合のみです。 SERVER_TYPE の値がそれ以外なら、このオプションは無視されます。

関連概念:

- 「*連合システム・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*』
- 「*連合システム・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*』

関連資料:

- 「*SQL リファレンス 第 2 巻*」の『*CREATE SERVER ステートメント*』

付録 C. 連合システムのユーザー・マッピング・オプション

ユーザー・マッピング・オプションは、連合サーバーとデータ・ソース間のユーザー・マッピングのための、許可および会計情報ストリングを提供します。これらのオプションは、ユーザー ID とパスワード権限をサポートする任意のデータ・ソースで使用できます。

これらのオプションは CREATE USER MAPPING ステートメントで使用します。

表 69. ユーザー・マッピング・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
ACCOUNTING_STRING	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
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。	連合データベースとの接続に使用される ID。
REMOTE_DOMAIN	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。有効な設定値は、任意の有効な Windows NT ドメイン・ネームです。このオプションを指定しない場合は、Documentum データ・ソースはそのデータベースのデフォルトの認証ドメインを使用して認証します。	なし
REMOTE_PASSWORD	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。 このオプションを指定しないなら、データ・ソース・サーバーとの接続にパスワードは使用されません。サーバーとの接続にパスワードが必要な場合、接続は失敗します。	なし

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』

付録 D. 連合システムの列オプション

列情報は、CREATE NICKNAME または ALTER NICKNAME ステートメントで列オプションと呼ばれるパラメーターを使用して、指定することができます。

列オプションの主目的は、SQL コンパイラーにニックネーム列についての情報を提供することです。1 つまたは複数の列の列オプションを 'Y' にセットすると、SQL コンパイラーは、評価操作を実行する述部について、追加のプッシュダウンの可能性を考慮します。これは、コンパイラーがグローバルな最適化を実施するのに役立ちます。これらの値は、大文字でも小文字でも指定できます。

重要: 非リレーショナル・ラッパーの場合、付加的な列オプションを使用できます。

表 70. 列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
NUMERIC_STRING	'Y' Yes の場合、この列には数字のストリング '0'、'1'、'2'、... '9' が入る。ブランクは含まない。重要: この列に、後にブランクが付いた数値ストリングのみが入る場合、'Y' を指定することはお勧めできません。 'N' No の場合、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。	'N'

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るブランクがこの列には含まれないことを、 옵ティマイザーに知らせることになります。このオプションは、データ・ソースの照合シーケンスが DB2 の照合シーケンスとは異なる場合に役立ちます。このオプションでマークされた列は、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。

表 70. 列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
VARCHAR_NO_ TRAILING_BLANKS	<p>このオプションは、データ・ソースに変長長の文字データ・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。</p> <p>ある種のデータ・ソース (Oracle など) の文字比較セマンティクスは、ブランク埋め込み文字比較のセマンティクス、つまり DB2 for Linux、UNIX、および Windows の比較セマンティクスと同じ結果を戻すセマンティクスではありません。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 列だけにこれを適用したい場合に設定してください。</p> <p>‘Y’ Yes - この VARCHAR 列に末尾ブランクはない。</p> <p>このデータ・ソースの文字比較セマンティクスは、ブランク埋め込み文字比較セマンティクスであり、それは連合サーバーとは異なります。文字比較操作は連合サーバー上で処理され、データ・ソースにはプッシュダウンされません。</p> <p>‘N’ No - この VARCHAR 列には末尾ブランクがある。</p> <p>このデータ・ソースの文字比較セマンティクスは、ブランク埋め込み文字比較セマンティクスであり、それは連合サーバーの文字比較セマンティクスと類似しています。文字比較操作は、データ・ソースにプッシュダウンされて処理されます。</p>	‘N’

関連概念:

- 「連合システム・ガイド」の『プッシュダウン分析』

関連タスク:

- 「連合システム・ガイド」の『グローバルな最適化』

付録 E. 連合システムの関数マッピング・オプション

DB2 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。連合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、連合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 71. 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見積もり	'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見積もり	'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見積もり	'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見積もり	'450'

表 71. 関数マッピング・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
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テージの見積もり	'100'
REMOTE_NAME	データ・ソース関数の名前。	ローカル名

付録 F. 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、ベンダーにより、目的により、またオペレーティング・システムにより変化します。サポートされる値は使用するラッパーにより異なります。

CREATE SERVER ステートメントには有効なサーバー・タイプを指定する必要があります。

CTLIB ラッパー

Sybase 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

DBLIB ラッパー

DBLIB クライアント・ソフトウェアによりサポートされる、 Sybase または Microsoft SQL Server データ・ソース。

サーバー・タイプ	データ・ソース
SYBASE	Sybase

DJXMSSQL3 ラッパー

ODBC 3.0 以上のドライバーによりサポートされる、 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

DRDA ラッパー

DB2 ファミリー

表 72. DB2 for Linux、UNIX、および 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 V1.2
DB2/HP	IBM DB2 for HP-UX
DB2/NT	IBM DB2 for Windows NT
DB2/EEE	IBM DB2 Enterprise-Extended Edition
DB2/CS	IBM DB2 for Common Server
DB2/SUN	IBM DB2 for Solaris V1 and V1.2
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 73. DB2 for iSeries (and AS/400)

サーバー・タイプ	データ・ソース
DB2/400	IBM DB2 for iSeries and AS/400

表 74. 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 75. DB2 Server for VM および 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

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 ドライバーによりサポートされる 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. またはこれ以上

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
必要なし	任意の OLE DB provider

SQLNET ラッパー

Oracle SQL*Net V1 または V2 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 V7.3. またはこれ以上

Teradata ラッパー

Teradata V2R3 および V2R4 クライアント・ソフトウェアによりサポートされる Teradata データ・ソース。

サーバー・タイプ	データ・ソース
TERADATA	Teradata

付録 G. 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・オブジェクトにニックネームを作成すると、DB2 for Linux、DB2 for UNIX、および DB2 for Windows は、表についての情報をグローバル・カタログに入れます。

この情報は、それぞれの列のリモートのデータ・タイプおよび、対応する DB2 for Linux、DB2 for UNIX、および DB2 for Windows のデータ・タイプを含んでいます。DB2 for Linux、DB2 for UNIX、および DB2 for Windows データ・タイプは、ローカル・データ・タイプと呼ばれます。

連合データベースは、データ・タイプ・マッピングを使用して、データ・ソース・オブジェクトの列にどのような DB2 for Linux、DB2 for UNIX、および DB2 for Windows データ・タイプを定義するべきかを判別します。

連合サーバーがデータ・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 for Linux、DB2 for UNIX、および DB2 for Windows のデータ・タイプと対応付けられて (マッピングされて) いなければなりません。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 for Linux、DB2 for UNIX、および DB2 for Windows 連合サーバーは、以下のローカル・データ・タイプのマッピングをサポートしません。

- DATALINK
- ユーザー定義タイプ

データ・ソースのデータ・タイプと連合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。

次の表は、DB2 for Linux、DB2 for UNIX、および DB2 for 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。

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。

重要: データ・ソースから DB2 for Linux、DB2 for UNIX、および DB2 for 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 連合スキーマは SYSIBM です。

DB2 for z/OS and OS/390 データ・ソース

表 76. 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	-	-	-
FLOAT	8	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	0	-
ROWID	-	-	-	-	Y	-	VARCHAR	40	-	Y
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-

表 76. 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
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 for iSeries データ・ソース

表 77. 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

表 77. 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
CHAR	255	32672	-	-	Y	-	VARCHAR	-	0	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
FLOAT	4	-	-	-	-	-	REAL	-	-	-
FLOAT	8	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0	N
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 78. 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	-	-	-
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、DB2 for UNIX、および DB2 for Windows データ・ソース

表 79. DB2 for Linux、DB2 for UNIX、および DB2 for 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

表 79. DB2 for Linux、DB2 for UNIX、および DB2 for 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) (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
VARGRAPH	-	-	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	0	N

Informix データ・ソース

表 80. 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	32	-	-	-	-	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	-	-	-

表 80. 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
REAL	-	-	-	-	-	-	REAL	4	-	-
SERIAL	-	-	-	-	-	-	INTEGER	4	-	-
SERIAL8	-	-	-	-	-	-	BIGINT	-	-	-
SMALLFLOAT	-	-	-	-	-	-	REAL	4	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-
TEXT	-	-	-	-	-	-	CLOB	2147483647	-	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-

注:

- Informix DATETIME データ・タイプでは、DB2 UNIX および Windows 連合サーバーは、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 81. Microsoft SQL Server 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
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	-	-	-
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	-

表 81. Microsoft SQL Server 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
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_BINARY	1	254	-	-	-	-	CHARACTER	-	-	Y
SQL_BINARY	255	8000	-	-	-	-	VARCHAR	-	-	Y
SQL_BIT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_CHAR	1	254	-	-	-	-	CHAR	-	-	N
SQL_CHAR	255	8000	-	-	-	-	VARCHAR	-	-	N
SQL_DATE	-	-	-	-	-	-	DATE	4	-	-
SQL_DECIMAL	1	31	0	31	-	-	DECIMAL	-	-	-
SQL_DECIMAL	32	38	0	38	-	-	DOUBLE	8	-	-
SQL_DECIMAL	32	32	0	31	-	-	DOUBLE	8	-	-
SQL_DOUBLE	-	-	-	-	-	-	DOUBLE	8	-	-

表 81. Microsoft SQL Server 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
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
sysname	-	-	-	-	-	-	VARCHAR	30	-	Y
sysname	1	254	-	-	-	-	CHAR	-	-	N
text	-	-	-	-	-	-	CLOB	-	-	N
timestamp	-	-	-	-	-	-	VARCHAR	8	-	Y
tinyint	-	-	-	-	-	-	SMALLINT	2	-	-
固有 ID ²	1	4000	-	-	Y	-	VARCHAR	16	-	Y
varbinary	1	8000	-	-	-	-	VARCHAR	-	-	Y
varchar	1	8000	-	-	-	-	VARCHAR	-	-	N

表 81. Microsoft SQL Server 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:

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。

ODBC データ・ソース

表 82. 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

表 82. 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) (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_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 83. Oracle NET8 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	0	0	0	0	-	¥0	BLOB	2147483647	0	Y
CHAR	1	254	0	0	-	¥0	CHAR	0	0	N
CHAR	255	32672	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	63	0	0	-	¥0	REAL	0	0	N
FLOAT	64	126	0	0	-	¥0	DOUBLE	0	0	N
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
RAW	1	254	0	0	-	¥0	CHAR	0	0	Y
RAW	255	32672	0	0	-	¥0	VARCHAR	0	0	Y
ROWID	0	0	0	NULL	-	¥0	CHAR	18	0	N
VARCHAR2	1	32672	0	0	-	¥0	VARCHAR	0	0	N

Oracle SQLNET データ・ソース

表 84. Oracle SQLNET 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
CHAR	1	254	0	0	-	¥0	CHAR	0	0	N
CHAR	255	32672	0	0	-	¥0	VARCHAR	0	0	N
DATE	0	0	0	0	-	¥0	TIMESTAMP	0	0	N
FLOAT	1	63	0	0	-	¥0	REAL	0	0	N
FLOAT	64	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
RAW	1	254	0	0	-	¥0	CHAR	0	0	Y
RAW	255	32672	0	0	-	¥0	VARCHAR	0	0	Y
ROWID	0	0	0	NULL	-	¥0	CHAR	18	0	N
VARCHAR2	1	32672	0	0	-	¥0	VARCHAR	0	0	N

Sybase データ・ソース

表 85. Sybase CTLIB および DBLIB 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
binary	1	254	-	-	-	-	CHAR	-	-	Y
binary	255	16384	-	-	-	-	VARCHAR	-	-	Y
bit	-	-	-	-	-	-	SMALLINT	-	-	-
char	1	254	-	-	-	-	CHAR	-	-	N
char	255	16384	-	-	-	-	VARCHAR	-	-	N
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

表 85. Sybase CTLIB および DBLIB 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
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
real	-	-	-	-	-	-	REAL	-	-	-
smalldatetime	-	-	-	-	-	-	TIMESTAMP	-	-	-
smallint	-	-	-	-	-	-	SMALLINT	-	-	-
smallmoney	-	-	-	-	-	-	DECIMAL	10	4	-
sysname	1	254	-	-	-	-	CHAR	-	-	N
text	-	-	-	-	-	-	CLOB	-	-	-
timestamp	-	-	-	-	-	-	VARCHAR	8	-	Y
tinyint	-	-	-	-	-	-	SMALLINT	-	-	-
varbinary	1	16384	-	-	-	-	VARCHAR	-	-	Y
varchar	1	16384	-	-	-	-	VARCHAR	-	-	N

Teradata データ・ソース

表 86. 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
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	-	-	-

表 86. 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
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-
VARCHAR	32673	64000	-	-	-	-	CLOB	-	-	-
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	-	-
VARGRAPHIC	16337	32000	-	-	-	-	DBCLOB	-	-	-

関連概念:

- 「連合システム・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 H. 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・タイプと連合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 で使用されません。

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 ラッパーにあります。

DB2 連合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、DB2 for UNIX、および DB2 for 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、DB2 for UNIX、および DB2 for Windows 連合サーバーは、ローカル・データ・タイプ 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 7 では Informix 表の C2 のマッピングは DECIMAL となり、Informix バージョン 8 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、DB2 for UNIX、および DB2 for 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・タイプの間、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 87. 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	-	-	-
DOUBLE	-	8	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	8	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	REAL	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP	-	10	-	-	-	-	TIMESTAMP	-	-	-

表 87. 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) (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	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	N

DB2 for iSeries データ・ソース

表 88. 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	-	-	-

表 88. 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) (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
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 89. 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	-	-	-
REAL	-	4	-	-	-	-	REAL	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP	-	10	-	-	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	-
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPH	-	-	-	-	-	-	VARGRAPH	-	-	N

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

表 90. DB2 for Linux、DB2 for UNIX、および DB2 for 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 91. Informix 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	-	-	-	-	-	-	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	-	-	-
LONG VARCHAR	-	32700	-	-	N	-	TEXT	-	-	-
LONG VARCHAR	-	32700	-	-	Y	-	BYTE	-	-	-
REAL	-	4	-	-	-	-	SMALLFLOAT	-	-	-
SMALLINT	-	2	-	-	-	-	INTEGER	-	-	-
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	-	-	-

表 91. Informix 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

注:

1. このタイプ・マッピングは Informix サーバーのバージョン 7 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 8 以上でのみ有効。

Informix DATETIME データ・タイプでは、DB2 UNIX および Windows 連合サーバーは、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 datetime.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 92. 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
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	-	-	-
VARCHAR	8001	32672	-	-	Y	-	image	-	-	-

Oracle SQLNET データ・ソース

表 93. Oracle SQLNET 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	0	2147483647	0	0	Y	¥0	LONG RAW	0	0	Y
CHARACTER	1	254	0	0	N	¥0	CHAR	0	0	N
CHARACTER	0	0	0	0	Y	¥0	RAW	0	0	Y
CLOB	0	2147483647	0	0	N	¥0	LONG	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
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
VARCHAR	1	2000	0	0	Y	¥0	RAW	0	0	Y
VARCHAR	1	4000	0	0	N	¥0	VARCHAR2	0	0	N

注: DB2 for Linux、DB2 for UNIX、および DB2 for Windows の BIGINT データ・タイプは、透過的 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。

Oracle NET8 データ・ソース

表 94. Oracle NET8 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	0	2147483647	0	0	Y	¥0	BLOB	0	0	Y
CHARACTER	1	254	0	0	N	¥0	CHAR	0	0	N
CHARACTER	0	0	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
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
VARCHAR	1	4000	0	0	N	¥0	VARCHAR2	0	0	N
VARCHAR	1	2000	0	0	Y	¥0	RAW	0	0	Y

注: DB2 for Linux、DB2 for UNIX、および DB2 for Windows の BIGINT データ・タイプは、透過的 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 95. Sybase CTLIB および DBLIB 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	-	-	-	-	-	-	decimal	19	0	-
BLOB	-	-	-	-	-	-	image	-	-	-
CHARACTER	-	-	-	-	N	-	char	-	-	-
CHARACTER	-	-	-	-	Y	-	binary	-	-	-
CLOB	-	-	-	-	-	-	text	-	-	-
DATE	-	-	-	-	-	-	datetime	-	-	-
DECIMAL	-	-	-	-	-	-	decimal	-	-	-
DOUBLE	-	-	-	-	-	-	float	-	-	-
INTEGER	-	-	-	-	-	-	integer	-	-	-
REAL	-	-	-	-	-	-	real	-	-	-
SMALLINT	-	-	-	-	-	-	smallint	-	-	-
TIME	-	-	-	-	-	-	datetime	-	-	-
TIMESTAMP	-	-	-	-	-	-	datetime	-	-	-
VARCHAR	1	255	-	-	N	-	varchar	-	-	-
VARCHAR	256	32672	-	-	N	-	text	-	-	-
VARCHAR	1	255	-	-	Y	-	varbinary	-	-	-
VARCHAR	256	32672	-	-	Y	-	image	-	-	-

Teradata データ・ソース

表 96. 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	-	-	-
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	-	-	-

表 96. 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	

- 注:
1. Teradata VARBYTE データ・タイプに含めることができるのは、指定された長さ (1 ~ 64000) の DB2 BLOB データ・タイプのみです。
 2. Teradata VARCHAR データ・タイプに含めることができるのは、指定された長さ (1 ~ 64000) の DB2 CLOB データ・タイプのみです。
 3. Teradata VARGRAPHIC データ・タイプに含めることができるのは、指定された長さ (1 ~ 32000) の DB2 DBCLOB データ・タイプのみです。
-

関連概念:

- 「連合システム・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』

DB2 Information Integrator の技術資料

続くいくつかのトピックでは、以下の実行方法を説明します。

- 資料およびリリース情報にアクセスする (資料の印刷および注文も含む)
- DB2 Information Integrator インフォメーション・センターまたは DB2 HTML ドキュメンテーション CD を使用してトピックにアクセスする

資料およびリリース情報へのアクセス

DB2 Information Integrator の技術情報は、以下の形式で利用可能です。

- ブック (PDF およびハードコピー)。DB2 Information Integrator ライブラリー内の各ブックの説明は、www.ibm.com/shop/publications/order の IBM Publications Center にあります。
- インフォメーション・センター (HTML 形式)。
- DB2 データベース・ツールのヘルプ (HTML 形式)。

DB2 Information Integrator の資料

DB2 Information Integrator 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 Information Integrator 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: $x:\backslash\text{doc}\backslash\%L$
- 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 CD-ROM ドライブ名を表します。
- $cdrom$ は CD-ROM 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)。

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	1
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。

以下の表に、DB2 Information Integrator 用の利用可能な資料を示します。

表 97. DB2 Information Integrator 資料

資料名	資料番号	インストール・ カテゴリ	PDF ファイル名
「 <i>IBM DB2 Information Integrator</i> ソリューション・ガイド」	SC88-9563	getting_started	iiyisx80
「 <i>IBM DB2 Information Integrator</i> インストール・ガイド」	GC88-9562	getting_started	iiyigx80
「 <i>IBM DB2 Information Integrator</i> マイグレーション・ガイド」	SC88-9610	getting_started	iiymgx80
「 <i>IBM DB2 Information Integrator</i> 連合システム・ガイド」	SC88-9614	admin	iiyfp80
「 <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	optional	iiylsx80
「 <i>IBM DB2 Information Integrator</i> 開発者向けガイド」	SC88-9609	ad	iiyfsx80

PDF ファイルからの資料の印刷方法

DB2 Information Integrator PDF ドキュメンテーション CD に収録されている DB2 Information Integrator 資料の PDF ファイルを印刷することができます。 Adobe Acrobat Reader を使用して、資料全体、一定範囲のページ、または特定のページを印刷できます。

前提条件:

Adobe Acrobat Reader がインストールされていることを確認してください。このプログラムは、Adobe 社の Web サイト (www.adobe.com) から入手できます。

手順:

PDF ファイルから DB2 Information Integrator 資料を印刷するには以下のようにします。

1. DB2 Information Integrator PDF ドキュメンテーション CD をドライブに挿入します。 UNIX オペレーティング・システムの場合、CD をマウントします。
2. Adobe Acrobat Reader を起動します。
3. 以下に示すいずれかの場所から PDF ファイルを開きます。
 - 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 CD-ROM ドライブ名を表します。
 - `cdrom` は CD-ROM のマウント・ポイントを表します (UNIX)。
 - `%L` は印刷する資料のロケールです (en_US など)。
4. 「ファイル」->「印刷」をクリックします。
5. 「印刷」ウィンドウで、すべてのページを印刷するか、現在のページを印刷するか、ページ範囲を印刷するかを指定します。
6. 「OK」をクリックします。

印刷資料の注文方法

ご使用の DB2 Information Integrator 製品用の Doc Pack (ドキュメンテーション・パッケージ) を IBM 販売店に注文することによって、印刷された DB2 Information Integrator マニュアルを入手できます。 Doc Pack とは、DB2 Information Integrator ライブラリーのマニュアルのサブセットです。 Doc Pack は、購入した DB2 製品を使い始めるのに役立つように作られています。

Doc Pack のマニュアルは、DB2 Information Integrator 製品に付属の DB2 Information Integrator PDF ドキュメンテーション CD に収録されているマニュアルと同じものです。

以下の方法で、個々の資料を注文することもできます。

- IBM 営業担当員または認定販売業者に連絡する。お客様がお住まいの地域の IBM 担当員の情報については、お手数ですが IBM の Web サイト (www.ibm.com/planetwide) の IBM Worldwide Directory of Contacts で確認してください。
- IBM Publications Center (www.ibm.com/shop/publications/order) にアクセスする。

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。リリース情報は、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要約も提供します。

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-ROM から入手できます。

- Windows オペレーティング・システムの場合: $x:\backslash\text{doc}\backslash\%L$
- UNIX オペレーティング・システムの場合: $/\text{cdrom}/\text{doc}/\%L/$

各部分の意味は以下のとおりです。

- x は Windows CD-ROM ドライブ名を表します。
- cdrom は CD-ROM 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)。

表 98. リリース情報

資料名	ファイル名	位置
「DB2 Information Integrator リリース情報」	ReleaseNotes	<ul style="list-style-type: none">• 製品 CD-ROM• DB2 Information Integrator インフォメーション・センター• DB2 Information Integration Installation Launchpad
「DB2 Information Integrator インストール要件」	Prereqs	<ul style="list-style-type: none">• 製品 CD-ROM• DB2 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 は定期的にドキュメンテーション・フィックスパックを提供しています。ドキュメンテーション・フィックスパックによって、新しい情報が入手可能になった場合に、DB2 HTML ドキュメンテーション CD からインストールした情報をアップデートすることができます。

ドキュメンテーション・フィックスパックは累積的なものです。たとえば、バージョン 8.1 用のドキュメンテーションをインストールしてからバージョン 8.1.2 を適用すると、フィックスパック 1 のアップデートとバージョン 8.1.2 のアップデートがドキュメンテーションに適用されます。

ドキュメンテーション・フィックスパックをインストールすると、HTML ドキュメンテーションには、製品の印刷版マニュアルおよびオンライン PDF マニュアルよりも新しい情報が記載されることになります。

DB2 Information Integrator インフォメーション・センターまたは DB2 HTML ドキュメンテーション CD を使ったトピックへのアクセス

DB2 Information Integrator インフォメーション・センターを使用すると、DB2 Information Integrator をビジネスで活用するために必要な情報にアクセスできます。

DB2 Information Integrator インフォメーション・センターの機能

DB2 Information Integrator インフォメーション・センターには、以下の機能があります。

統合されたナビゲーション・ツリー

1 つのナビゲーション・ツリーから、DB2 Information Integrator ライブラリー内のすべてのトピックを探し出すことができます。

検索 ナビゲーション・ツールバーの「**検索**」をクリックすることによって、ワークステーション上のすべてのトピックを検索できます。

マスター索引

1 つのマスター・インデックスからトピックや Tools ヘルプの情報にアクセスできます。この索引には、DB2 Information Integrator ライブラリー全体の項目が入っています。

マスター用語集

マスター用語集は、DB2 Information Integrator ライブラリーで使用されている用語を定義しています。

資料の定期的な更新

更新された HTML トピックをダウンロードすることによって、トピックを最新の状態に保つことができます。

DB2 Information Integrator インフォメーション・センターのトピックを見つける

DB2 Information Integrator インフォメーション・センターは、次の主なエレメントから構成されています。

ナビゲーション・ツリー

ナビゲーション・ツリーは、ブラウザー・ウィンドウの左側のフレームにあり

ます。ツリーを拡張したり縮小したりして、DB2 Information Integrator インフォメーション・センター内のトピック・リンクや、用語集、マスター索引などを表示したり、隠したりできます。

ナビゲーション・ツールバー

ナビゲーション・ツールバーは、ブラウザ・ウィンドウの上部右側のフレームにあります。ナビゲーション・ツールバー各種のプッシュボタンを使って、DB2 Information Integrator インフォメーション・センターを検索したり、ナビゲーション・ツリーを隠したり、ナビゲーション・ツリーに現在表示されているトピックを検索したりできます。

内容フレーム

内容フレームは、ブラウザ・ウィンドウの下部右側のフレームにあります。ナビゲーション・ツリーのリンクをクリックしたり、検索結果をクリックしたり、別のトピックやマスター索引からのリンクをクリックしたりすると、内容フレームに該当するトピックが表示されます。

前提条件:

ブラウザから DB2 Information Integrator インフォメーション・センターにアクセスするには、以下のいずれかのブラウザを使用する必要があります。

- Microsoft Explorer バージョン 5 以上
- Netscape Navigator バージョン 6.1 以上

制約事項:

DB2 Information Integrator インフォメーション・センターには、DB2 HTML ドキュメンテーション CD から選択してインストールしたトピックだけが入っています。トピックへのリンクをクリックするとき、Web ブラウザーが「ファイルが見つかりません」というエラーを戻す場合、DB2 HTML ドキュメンテーション CD から追加のトピックをインストールしてください。

手順:

キーワードを指定してトピックを検索するには以下のようにします。

1. ナビゲーション・ツールバーで、「**検索**」をクリックします。
2. 「**検索**」ウィンドウの一番上のテキスト入力フィールドに、参照したいトピックに関係する用語を 2 つ以上入力して、「**検索**」をクリックします。「**結果**」フィールドにトピックのリストが表示されます。入力した検索ストリングに最も近似のトピックがリストの最上部に表示されます。

入力する用語が多いほど、検索の精度が上がり、結果のトピック数を減らすことができます。

3. 「**結果**」フィールドで、参照したいトピックのタイトルをクリックします。そうすると、そのトピックが内容フレームに表示されます。

ナビゲーション・ツリー内のトピックを検索するには以下のようにします。

1. ナビゲーション・ツリーで、参照したい内容に関係したトピックのカテゴリのとなりのブック・アイコンをクリックします。サブカテゴリのリストがアイコンの下に表示されます。
2. 参照したいトピックの入ったカテゴリが見つかるまで、ブック・アイコンをクリックします。トピックにリンク接続されているカテゴリは、そのカテゴリ・タイトルの上にカーソルを置くと、リンクとして表示されます。ナビゲーション・ツリーの中で、ページ・アイコンはトピックを識別するために使用されています。
3. トピック・リンクをクリックします。そうすると、そのトピックが内容フレームに表示されます。

マスター索引内のトピックを検索するには以下のようにします。

1. ナビゲーション・ツリーで、「索引」をクリックします。索引が拡張され、50 音順のリンクのリストが表示されます。
2. ナビゲーション・ツリーで、探しているサブジェクトの最初の文字をクリックします。その文字で始まる項目のリストが内容フレームに表示されます。ブック・アイコンが表示される場合、そのサブジェクトには複数の索引項目があります。
3. 参照したいサブジェクトに対応したブック・アイコンをクリックします。クリックした用語の下に、関係するトピックのリストが表示されます。
4. 参照したいトピックのリストをクリックします。そうすると、そのトピックが内容フレームに表示されます。

DB2 HTML ドキュメンテーションの使用

このトピックでは、DB2 HTML ドキュメンテーション CD にあるドキュメンテーションをインストール、表示、およびコピーする方法と、インストール後にそれらを更新する方法を説明しています。

DB2 HTML ドキュメンテーションのインストール

DB2 HTML ドキュメンテーション CD のインストール・ディレクトリーは、情報のカテゴリーごとに異なります。以下のとおりです。

htmlcdpath/doc/htmlcd/%L/category

htmlcdpath

DB2 HTML ドキュメンテーション CD がインストールされるディレクトリーです。

%L 使用する資料のロケールです (en_US など)。

category

カテゴリー ID。たとえば、インストール情報は `getting_started` です。

DB2 HTML ドキュメンテーション CD から技術資料を直接参照する

すべての HTML トピックは DB2 HTML ドキュメンテーション CD から表示できません。

制約事項:

オンライン・ヘルプを表示するには、DB2 製品をインストールする必要があります。

手順:

DB2 HTML ドキュメンテーション CD から HTML ドキュメンテーションを表示するには、以下の手順に従います。

1. DB2 HTML ドキュメンテーション CD をドライブに挿入します。
2. Web ブラウザーを起動して、以下のファイルを開きます。
 - Windows オペレーティング・システムの場合
`x:\Program Files\IBM\sqllib\doc\htmlcd\%L\index.*`
 - UNIX オペレーティング・システムの場合
`/cdrom/Program Files/IBM/sqllib/doc/htmlcd/%L/index.htm`

各部分の意味は以下のとおりです。

- *x* は Windows CD-ROM ドライブ名を表します。
- *cdrom* は CD-ROM 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)。

DB2 HTML ドキュメンテーション CD から Web サーバーへファイルをコピーする

すべての DB2 ライブラリーは DB2 HTML ドキュメンテーション CD に収録されています。ライブラリーを有効に活用するため、Web サーバーにインストールすることができます。

手順:

ファイルを DB2 HTML ドキュメンテーション CD から Web サーバー上の適切なパスにコピーします (デフォルトのパスを以下に示します)。

- Windows オペレーティング・システムの場合: `x:\Program Files\IBM\sqllib\doc\htmlcd\%L*`
- UNIX オペレーティング・システムの場合: `/cdrom/Program Files/IBM/sqllib/doc/htmlcd/%L`

各部分の意味は以下のとおりです。

- *x* は Windows CD-ROM ドライブ名を表します。
- *cdrom* は CD-ROM 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)。

コンピューター上の HTML 資料をアップデートする

IBM がアップデートを提供した場合、DB2 HTML ドキュメンテーション CD からインストールした HTML ファイルを、以下のいずれかの方法でアップデートできます。

- インフォメーション・センターを使用する (DB2 管理グラフィカル・ユーザー・インターフェイス・ツールをインストールしてある場合)。
- DB2 HTML ドキュメンテーション・フィックスパックをダウンロードして適用する。

これらの手順を実行しても、DB2 のコードがアップデートされるわけではありません。

前提条件:

必要な場合にはアップデート・プログラムが IBM サーバーから最新のドキュメンテーション・フィックスパックをダウンロードするので、コンピューターがインターネットに接続されていることを確認します。インターネットに接続するには、プロキシ情報を提供することが必要な場合もあります。

手順:

インフォメーション・センターを使用して、マシンにインストールされている HTML ドキュメンテーションをアップデートするには以下のようにします。

1. DB2 インフォメーション・センターを開始します。
 - グラフィカル管理ツールから、ツールバーの「インフォメーション・センター」アイコンをクリックします。
 - コマンド行で、db2ic と入力します。
2. 「インフォメーション・センター」 -> 「Update Local Documentation (ローカル・ドキュメンテーションのアップデート)」をクリックして、アップデートを開始します。

ドキュメンテーション・アップデートが入手可能であれば、ダウンロードされて、適用されます。

手操作でドキュメンテーション・アップデートをダウンロードして適用するには、以下のようにします。

1. Web ブラウザーで DB2 サポート・ページ www.ibm.com/software/data/db2/udb/winos2unix/support を開きます。
2. 「DB2 Version 8」をクリックして、ご使用のオペレーティング・システム用のドキュメンテーション・フィックスパックのリンクを探します。
3. ドキュメンテーション・フィックスパックのレベルと、マシンにインストールされている文書のレベルを比較して、ローカルな DB2 文書が更新されているかどうかを確認します。

4. さらに新しいバージョンの文書がある場合、ご使用のオペレーティング・システムに対応するフィックスパックをダウンロードします。すべての Windows オペレーティング・システム用に 1 つ、およびすべての UNIX オペレーティング・システムに 1 つのフィックスパックがあります。
5. 以下のようにしてフィックスパックを適用します。
 - Windows オペレーティング・システムの場合、フィックスパックは自己抽出型の zip ファイルとなっています。空のディレクトリーにドキュメンテーション・フィックスパックをダウンロードし、`unzip` します。そのディレクトリーで `setup` コマンドを実行して、ドキュメンテーション・フィックスパックをインストールします。
 - UNIX オペレーティング・システムの場合、ドキュメンテーション・フィックスパックは圧縮された tar.Z となっています。tar ファイルが解凍・展開されると、`delta_install` というディレクトリーが作成されます。そのディレクトリー内でスクリプト `installdocfix` を実行して、ドキュメンテーション・フィックスパックをインストールします。

DB2 文書の検索

DB2 資料を検索するには、Netscape バージョン 6.1 以上または Microsoft Internet Explorer バージョン 5 以上を使用してください。ブラウザの Java サポートが有効になっていることを確認してください。

ブラウザから DB2 Information Integrator インフォメーション・センターにアクセスして、ナビゲーション・ツールバーの検索アイコンをクリックすると、検索ウィンドウが開きます。初めて検索機能を実行する場合、検索ウィンドウのロードに 1 分程度かかる場合があります。

制約事項:

文書検索を実行するときには、以下の制約事項があります。

- ブール検索はサポートされていません。ブール検索修飾子 *and* および *or* は、検索において無視されます。たとえば、以下の検索の結果は同じになります。
 - サブレット *and* Bean
 - サブレット *or* Bean
- ワイルドカード検索はサポートされていません。 *java** を検索しても、リテラル・ストリング *java** が検出され、 *javadoc* などは検出されません。

一般に、単一の語を検索するよりも句を検索したほうが、より良い検索結果が得られません。

手順:

DB2 文書を検索するには以下のようにします。

1. ナビゲーション・ツールバーで、「検索」をクリックします。
2. 「検索」ウィンドウの一番上のテキスト入力フィールドに、参照したいトピックに関する用語を 2 つ以上入力して、「検索」をクリックします。「結果」フィールドに、正確さのランクに従ってトピックのリストが表示されます。
入力する用語が多いほど、検索の精度が上がり、結果のトピック数を減らすことができます。
3. 「結果」フィールドで、参照したいトピックのタイトルをクリックします。そうすると、そのトピックが内容フレームに表示されます。

検索を実行するときには、最初の結果が自動的にブラウザー・フレームにロードされます。他の検索結果の内容を表示するには、リストから結果をクリックします。

Netscape 4.x を使って DB2 資料を検索する場合のトラブルシューティング

検索の問題のほとんどは、Web ブラウザーの Java サポートに関係しています。このトピックでは、可能な解決策を記載しています。

手順:

Netscape 4.x に共通している問題は、セキュリティー・クラスが欠落しているか、配置が間違っていることです。以下に示す解決策を実行してみてください (特にブラウザーの Java コンソールに以下の行がある場合は必ず実行してください)。

```
Cannot find class java/security/InvalidParameterException
```

以下のファイルを、DB2 HTML ドキュメンテーション CD から、Netscape ブラウザーがインストールされているディレクトリー内にある `java\classes\java\security\` ディレクトリーにコピーします。`java\security\` サブディレクトリーを作成することが必要な場合もあります。

- Windows オペレーティング・システムの場合

```
x:Program Files\IBM\sqllib\doc\htmlcd\%L\InvalidParameterException.class
```

- UNIX オペレーティング・システムの場合

```
/cdrom/Program Files/IBM/sqllib/doc/htmlcd/%L  
/InvalidParameterException.class
```

各部分の意味は以下のとおりです。

- `x` は Windows CD-ROM ドライブ名を表します。
- `cdrom` は CD-ROM 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)。

上記の対処策を実行しても Netscape ブラウザーで検索入力ウィンドウが表示されない場合は、以下の処置を実行してください。

- Netscape ブラウザーのすべてのインスタンスを停止して、コンピューター上で Netscape コードが実行されていないことを確認します。その後、Netscape ブラウザーの新しいインスタンスを開き、再度検索を開始します。
- ブラウザーのキャッシュを削除します。
- Netscape の別のバージョンまたは別のブラウザーを使用します。

アクセシビリティ

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）ユーザーは、アクセシビリティ機能を使用することによって、ソフトウェア製品を十分活用できます。DB2 Information Integrator バージョン 8 に備わっている主なアクセシビリティ機能は以下のとおりです。

- すべての機能をマウスの代わりにキーボードを使用して操作できます。
- フォントのサイズと色をカスタマイズできます。
- アラートを表示にするか音声にするかを指定できます。
- DB2 は、Java™ Accessibility API を使用するアクセシビリティ・アプリケーションをサポートします。
- DB2 の資料は、アクセスしやすい形式で提供されています。

キーボードによる入力およびナビゲーション

キーボードだけを使用して、コントロール・センター、データウェアハウス・センター、レプリケーション・センターなどの、DB2 データベース・ツールを操作できます。マウスの代わりに複数のキーまたはキーの組み合わせを使用してほとんどの操作を実行できます。

UNIX ベースのシステムでは、キーボード・フォーカスの置かれている位置が強調表示されます。この強調表示によって、アクティブなウィンドウ領域が示されます。そのウィンドウ領域が、ユーザーのキー・ストロークの対象となります。

アクセスしやすい表示

DB2 データベース・ツールには、視力の弱いユーザーのためにユーザー・インターフェースを拡張し、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

DB2 データベース・ツールでは、「ツール設定」ノートブックを使用して、メニューおよび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。

色覚への非依存

この製品の機能を使用するために、ユーザーは色を識別する必要はありません。

代替アラート・キュー

「ツール設定」ノートブックを使用して、アラートを音声にするか、表示にするかを指定できます。

支援テクノロジーとの互換性

DB2 Information Integrator インターフェースは、身体に障害を持つ人々によって使用されているスクリーン・リーダーおよび他の支援テクノロジーに採用されている Java Accessibility API をサポートしています。

入手可能な資料

DB2 ファミリー製品の資料は HTML 形式で入手可能です。資料は、ご使用のブラウザーに設定されている表示設定に従って表示することができます。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。

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すが、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が、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

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。

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. _西暦年_. All rights reserved.

商標

以下は、IBM Corporation の商標です。

AIX
DB2
Domino
IBM
Informix
Lotus
Lotus Notes
QuickPlace
WebSphere

以下は、それぞれ各社の商標または登録商標です。

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。

UNIX は、The Open Group がライセンスしている米国およびその他の国における登録商標です。

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または登録商標です。

他の会社名、製品名およびサービス名などはそれぞれ各社の商標または登録商標です。

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]

大文字小文字の区別

大文字小文字の区別の保持 20

オブジェクト

ネーミング 19

[カ行]

開始

HMMER

デーモン 331

カスタム関数

Extended Search 309

カタログ

「グローバル・カタログ」を参照
375

環境変数

HMMER ラッパー 333

関数テンプレート

Extended Search 309

関数マッピング

説明 16

プラン 24

関数マッピング・オプション

有効な設定値 395

逆方向タイプ・マッピング

デフォルト・マッピング 421

グローバル・カタログ

連合情報を含むビュー 375

構成

連合データ・ソース

概要 8

HMMER デーモン 328

コマンド・センター

データ・ソースの構成 1

コントロール・センター

データ・ソースの構成 1

[サ行]

サーバー定義

説明 10

サーバー・オプション

一時 10

説明 10

有効な設定値 379

サーバー・タイプ

有効な連合タイプ 397

最適化

表構造ファイル 153

BLAST 234

HMMER

ラッパー 342

索引の指定

説明 16

サマリー表

ニックネームの作成 5

サンプル

照会

BLAST 223, 232

Documentum 181

Excel 199

Extended Search 310, 312

XML 258

HMMER

照会 339

順方向タイプ・マッピング

デフォルト・マッピング 401

垂直表

Extended Search 302

[タ行]

チューニング

データ・ソース構成

DB2 ファミリー・データ・ソース 43

Informix 57

Microsoft SQL Server 103

ODBC 120

Oracle 73

Sybase 87

Teradata 136

追加

HMMER、連合 327

データ・ソース

オプションの構成ステップ 8

構成 1

有効なサーバー・タイプ 397

データ・ソース・オブジェクト

説明 12

有効なオブジェクト・タイプ 13

リモート 5

ローカル 5

データ・タイプ

サポートされない 15

データ・タイプ・マッピング

逆方向 421

順方向 401

説明 15

プラン 22

統計

データ・ソースの更新 22

統計の更新の計画 22

登録

HMMER

ラッパー 332

server

HMMER データ・ソース 334

トラブルシューティング

Microsoft SQL Server 構成 103

ODBC 構成 120

Oracle 構成 73

トラブルシューティング (続き)

Sybase 構成 87

Teradata 構成 136

[ナ行]

ニックネーム

作成の概要 5

サマリー表の 5

説明 12

データ・タイプの変更 345

ニックネームの 5

ニックネーム・オプションの変更
346

変更 345

有効なデータ・ソース・オブジェ
クト 13

列オプション、例 5

DB2 ファミリー・データ・ソー
スの場合の作成 41

Informix のための作成 55

Microsoft SQL Server のための作
成 101

ODBC のための作成 117

Oracle のための作成 71

Sybase のための作成 86

Teradata の場合の作成 135

XML の場合の作成 251

ニックネームの登録

HMMER

データ・ソース 335

[ハ行]

パスワード

値の大文字小文字が区別される
20

汎用照会言語

Extended Search 315

表構造ファイル

概要 145

最適化 153

サポートされるバージョン 3

タイプ 145

ニックネーム、有効なオブジェ
クト 13

表構造ファイル (続き)

ファイル・アクセス制御モデル
153

メッセージ 154

例 145

連合システムへの追加

サーバーの登録 150

ニックネームの登録 151

ラッパーの登録 148

DB2 Information Integrator による
アクセス 146

非リレーショナル・データ・ソース
データ・タイプ・マッピング、指
定 15

フラット・ファイル

「表構造ファイル」を参照 3

[マ行]

マップ・フィールド

Extended Search 302

命名規則

連合データベース・オブジェクト
の 19

メッセージ

表構造ファイル・ラッパー 154

BLAST ラッパー 234

Documentum ラッパー 186

Excel ラッパー 203

Extended Search ラッパー 317

HMMER ラッパー 342

XML ラッパー 234

[ヤ行]

ユーザー ID

値の大文字小文字が区別される
20

ユーザー定義タイプ (UDT)

サポートされないデータ・タイプ
15

ユーザーのマッピング

Extended Search 308, 373

ユーザー・マッピング

説明 11

プラン 25

ユーザー・マッピング (続き)

Extended Search 308, 373

ユーザー・マッピング・オプション
説明 11

有効な設定値 391

[ラ行]

ラッパー

説明 9

プラン 25

Teradata の場合の作成 127

XML の場合の作成 243

リモート・オブジェクト

説明 5

列オプション

説明 14

有効な設定値 393

例 5

ローカル・オブジェクト

説明 5

A

ACCOUNTING_STRING ユーザー・
オプション

有効な設定値 391

B

BLAST

サポートされるバージョン 3

説明 211

ニックネーム、有効なオブジェ
クト 13

メッセージ 234

連合システムへの追加

サーバーの登録 223

正しい blastall executable がイ
ンストールされていることを
検査 217

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 217

ニックネームの登録 223

ラッパーの登録 221

BLAST (続き)

連合システムへの追加 (続き)

- BLAST 構成ファイル 217
- BLAST デーモンの開始 220
- BLAST デーモンの設定と構成 217
- CREATE NICKNAME ステートメント 223
- CREATE SERVER ステートメント 223
- CREATE WRAPPER ステートメント 221

C

COLLATING_SEQUENCE サーバー・オプション

有効な設定値 379

COMM_RATE サーバー・オプション
有効な設定値 379

CONNECTSTRING サーバー・オプション

有効な設定値 379

CPU_RATIO サーバー・オプション
有効な設定値 379

CREATE FUNCTION ステートメント

- Documentum 174
- Extended Search 309, 347

CREATE NICKNAME ステートメント

- 表構造ファイル 151
- 例 5
- BLAST 223
- DB2 ファミリー・データ・ソース 41, 42
- Documentum 168
- Excel ファイル 199
- Extended Search 307, 355
- Informix 55, 56
- Microsoft SQL Server 101, 103
- ODBC 117, 119
- Oracle 71, 72
- Sybase 86, 87
- Teradata 135
- XML 251, 363

CREATE SERVER ステートメント
表構造ファイル 150

- BLAST 223
- DB2 ファミリー・データ・ソース 37
- Documentum 166
- Excel ファイル 198
- Extended Search 306, 371
- Informix 50
- Microsoft SQL Server 97
- ODBC 113
- OLE DB 142
- Oracle 67
- Sybase 80
- Teradata 128, 130
- XML 245

CREATE USER MAPPING ステートメント

- DB2 ファミリー・データ・ソース 38, 39
- Documentum 167
- Extended Search 308, 373
- Informix 52
- Microsoft SQL Server 98, 99
- ODBC 114, 115
- OLE DB 143
- Oracle 68, 69
- Sybase 83
- Teradata 131, 132

CREATE WRAPPER ステートメント
表構造ファイル 148

- BLAST 221
- Documentum 165
- Excel ファイル 197
- Extended Search 305, 373
- ODBC 111
- XML 243

CreateNicknameFile ユーティリティ
一、Documentum

- インストール 183
- 構成 183
- 説明 182
- DM_ID オブジェクト・タイプのマッピング 184

D

DATALINK データ・タイプ
サポートされない 15

DATEFORMAT サーバー・オプション
有効な設定値 379

DB2 for iSeries

- サポートされるバージョン 3
- デフォルトの逆方向タイプ・マッピング 421
- デフォルトの順方向タイプ・マッピング 401
- ニックネーム、有効なオブジェクト 13
- 有効なサーバー・タイプ 397

DB2 for Linux, DB2 for UNIX, および DB2 for Windows

- サポートされるバージョン 3
- デフォルトの逆方向タイプ・マッピング 421
- デフォルトの順方向タイプ・マッピング 401
- ニックネーム、有効なオブジェクト 13
- 有効なサーバー・タイプ 397

DB2 for VM and VSE

- サポートされるバージョン 3
- デフォルトの逆方向タイプ・マッピング 421
- デフォルトの順方向タイプ・マッピング 401
- ニックネーム、有効なオブジェクト 13
- 有効なサーバー・タイプ 397

DB2 for z/OS and OS/390

- サポートされるバージョン 3
- デフォルトの逆方向タイプ・マッピング 421
- デフォルトの順方向タイプ・マッピング 401
- ニックネーム、有効なオブジェクト 13
- 有効なサーバー・タイプ 397

DB2 ファミリー・データ・ソース
サーバー定義の登録 35

DB2 ファミリー・データ・ソース
(続き)
チューニング 43
ノード・エントリーのカタログ
32
へのアクセスの構成 31
ラッパーの登録 34
リモート・データベースのカタ
ログ 33
連合システムへの追加 31
接続のテスト 40
ニックネームの登録 41
ユーザー・マッピングの作成
38
CREATE NICKNAME ステ
ートメント 42
CREATE SERVER ステートメ
ント 37
CREATE USER MAPPING ス
テートメント 39

DB2_DJ_COMM 環境変数 149, 165,
222, 244

DB2 ファミリー・データ・ソー
スのチューニング 43
Informix、チューニング 57
Microsoft SQL Server、チューニ
ング 103
ODBC、チューニング 120
Oracle、チューニング 73
Sybase、チューニング 87
Teradata のチューニング 136

DB2_MAXIMAL_PUSHDOWN サー
バー・オプション
有効な設定値 379

DBNAME サーバー・オプション
有効な設定値 379

DISABLE function 関数マッピング・
オプション
有効な設定値 395

Documentum
繰り返し属性の二重の定義 185
サポートされるバージョン 3
資料へのユーザー・アクセス
186
説明 159

Documentum (続き)
ニックネーム、有効なオブジェク
ト 13
メッセージ 186
例 159
連合システムへの追加
カスタム関数の登録 174
サーバーの登録 166
ニックネームの登録 168
ユーザーのマッピング 167
ラッパーの登録 165
CREATE FUNCTION ステ
ートメント 174
CREATE NICKNAME ステ
ートメント 168
CREATE SERVER ステートメ
ント 166
CREATE USER MAPPING ス
テートメント 167
CREATE WRAPPER ステート
メント 165
CreateNicknameFile ユーティリ
ティ 182
Documentum クライアント・
ライブラリーを DB2 で使用
可能にする (AIX および
Solaris オペレーティング環
境のみ) 162
Documentum のクライアント
dmcl.ini ファイルの位置指定
163
CreateNicknameFile ユーティリテ
ィー 182

E

Entrez
サポートされるバージョン 3
ニックネーム、有効なオブジェク
ト 13

Excel ファイル
サポートされるバージョン 3
サンプル・ユーザーのシナリオ
200
説明 195

Excel ファイル (続き)
ニックネーム、有効なオブジェク
ト 13
ファイル・アクセス制御モデル
202
メッセージ 203
連合システムへの追加
サーバーの登録 198
ニックネームの登録 199
ラッパーの登録 197
CREATE NICKNAME ステ
ートメント 199
CREATE SERVER ステートメ
ント 198

Extended Search
検索可能データ・ソース 297
サポートされるバージョン 3
サンプルの照会 312
照会のガイドライン 310
照会のカスタマイズ 309
垂直表 302
説明 297
ニックネーム、有効なオブジェク
ト 13
汎用照会言語 315
複数のニックネームの作成 300
マップ・フィールド 302
メッセージ 317
連合システムへの追加
関数テンプレートの登録 309
サーバーの登録 306
手順の要約 305
ニックネームの登録 307
ユーザーのマッピング 308
ユーザー・マッピングの登録
308
ラッパーの登録 305
CREATE FUNCTION ステ
ートメント 309, 347
CREATE NICKNAME ステ
ートメント 307, 355
CREATE SERVER ステートメ
ント 306, 371
CREATE USER MAPPING ス
テートメント 308, 373

Extended Search (続き)

連合システムへの追加 (続き)

CREATE WRAPPER ステートメント 305, 373

F

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
20

有効な設定値 379

Informix データ・ソースでの設定
57

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
20

有効な設定値 379

Informix データ・ソースでの設定
57

H

HMMER

サポートされるバージョン 3
照会

構成 341

ニックネーム、有効なオブジェクト 13

ラッパー 323

I

IFILE サーバー・オプション

有効な設定値 379

IGNORE_UDT サーバー・オプション

有効な設定値 379

Informix

クライアント構成ファイルのセットアップ 46

クライアント構成ファイルのテスト 46

サーバー定義の登録 49

サポートされるバージョン 3

チューニング 57

デフォルトの逆方向タイプ・マッピング 421

Informix (続き)

デフォルトの順方向タイプ・マッピング 401

ニックネーム、有効なオブジェクト 13

へのアクセスの構成 45

有効なサーバー・タイプ 397

ラッパーの登録 48

連合システムへの追加 45

接続のテスト 54

ニックネームの登録 55

ユーザー・マッピングの作成
52

CREATE NICKNAME ステートメント 56

CREATE SERVER ステートメント 50

CREATE USER MAPPING ステートメント 53

INFORMIX_LOCK_MODE サーバー・オプション

有効な設定値 379

INITIAL_INSTS 関数マッピング・オプション

有効な設定値 395

INITIAL_IOS 関数マッピング・オプション

有効な設定値 395

INSTS_PER_ARGBYTE 関数マッピング・オプション

有効な設定値 395

INSTS_PER_INVOC 関数マッピング・オプション

有効な設定値 395

IOS_PER_ARGBYTE 関数マッピング・オプション

有効な設定値 395

IOS_PER_INVOC 関数マッピング・オプション

有効な設定値 395

IO_RATIO サーバー・オプション

有効な設定値 379

IUD_APP_SVPT_ENFORCE サーバー・オプション

有効な設定値 379

L

LOGIN_TIMEOUT サーバー・オプション

有効な設定値 379

M

Microsoft Excel

「Excel ファイル」を参照 3

Microsoft SQL Server

サーバー定義の登録 95

サポートされるバージョン 3

チューニング 103

デフォルトの逆方向タイプ・マッピング 421

デフォルトの順方向タイプ・マッピング 401

トラブルシューティング 103

ニックネーム、有効なオブジェクト 13

へのアクセスの構成 91

有効なサーバー・タイプ 397

ラッパーの登録 94

連合サーバーの準備 92

連合システムへの追加

概要 91

接続のテスト 100

ニックネームの登録 101

ユーザー・マッピングの作成
98

CREATE NICKNAME ステートメント 103

CREATE SERVER ステートメント 97

CREATE USER MAPPING ステートメント 99

N

NODE サーバー・オプション

有効な設定値 379

NUMERIC_STRING 列オプション

有効な設定値 393

O

ODBC

- サーバー定義の登録 112
- サポートされるバージョン 3
- チューニング 120
- デフォルトの順方向タイプ・マッピング 401
- トラブルシューティング 120
- ニックネーム、有効なオブジェクト 13
- へのアクセスの構成 107
- 有効なサーバー・タイプ 397
- ラッパーの登録 110
- 連合サーバーの準備 109
- 連合システムへの追加
 - 概要 107
 - 接続のテスト 116
 - ニックネームの登録 117
 - ユーザー・マッピングの作成 114
- CREATE NICKNAME ステートメント 119
- CREATE SERVER ステートメント 113
- CREATE USER MAPPING ステートメント 115

OLE DB

- サーバー定義の登録 141
- サポートされるバージョン 3
- へのアクセスの構成 139
- 有効なサーバー・タイプ 397
- ラッパーの登録 140
- 連合システムへの追加
 - 概要 139
 - ユーザー・マッピングの作成 143
- CREATE SERVER ステートメント 142
- CREATE USER MAPPING ステートメント 143

Oracle

- クライアント構成ファイルのセットアップ 62
- クライアント構成ファイルのテスト 62

Oracle (続き)

- サーバー定義の登録 66
- チューニング 73
- デフォルトの逆方向タイプ・マッピング 421
- デフォルトの順方向タイプ・マッピング 401
- トラブルシューティング 73
- ニックネーム、有効なオブジェクト 13
- へのアクセスの構成 61
- ラッパーの登録 63
- 連合システムへの追加
 - 概要 61
 - 接続のテスト 70
 - ニックネームの登録 71
 - ユーザー・マッピングの作成 68
- CREATE NICKNAME ステートメント 72
- CREATE SERVER ステートメント 67
- CREATE USER MAPPING ステートメント 69

P

PACKET_SIZE サーバー・オプション

- 有効な設定値 379

PASSWORD サーバー・オプション

- 有効な設定値 379

PERCENT_ARGBYTES 関数マッピング・オプション

- 有効な設定値 395

PLAN_HINTS サーバー・オプション

- 有効な設定値 379

PUSHDOWN サーバー・オプション

- 有効な設定値 379

R

REMOTE_AUTHID ユーザー・オプション

- 有効な設定値 391

REMOTE_DOMAIN ユーザー・オプション

- 有効な設定値 391

REMOTE_NAME 関数マッピング・オプション

- 有効な設定値 395

REMOTE_PASSWORD ユーザー・オプション

- 有効な設定値 391

S

SET SERVER OPTION ステートメント

- 一時的にオプションを設定する 10

Sybase

- クライアント構成ファイルのセットアップ 76
- クライアント構成ファイルのテスト 76
- サーバー定義の登録 79
- サポートされるバージョン 3
- チューニング 87
- デフォルトの逆方向タイプ・マッピング 421
- デフォルトの順方向タイプ・マッピング 401
- トラブルシューティング 87
- ニックネーム、有効なオブジェクト 13
- へのアクセスの構成 75
- 有効なサーバー・タイプ 397
- ラッパーの登録 78
- 連合システムへの追加
 - 概要 75
 - 接続のテスト 84
 - ニックネームの登録 86
 - ユーザー・マッピングの作成 83
- CREATE NICKNAME ステートメント 87
- CREATE SERVER ステートメント 80
- SYSCAT カタログ・ビュー 375
- SYSTAT カタログ・ビュー 375

T

Teradata

- アクセス・ロギング 136
 - 実行時リンクの使用可能化
(AIX) 126, 136
 - チューニング 136
 - デフォルトの逆方向タイプ・マッピング 421
 - デフォルトの順方向タイプ・マッピング 401
 - トラブルシューティング 136
 - ニックネーム、有効なオブジェクト 13
 - へのアクセスの構成 123
 - 有効なサーバー・タイプ 397
 - 連合サーバーでのニックネームの概要 134
 - 連合システムへの追加
 - 概要 123
 - サーバー定義の登録 128
 - 実行時リンクの検査
(AIX) 126
 - 接続のテスト 125, 132
 - ニックネームの登録 135
 - ユーザー・マッピングの作成
131
 - ラッパーの登録 127
 - CREATE NICKNAME ステートメント 135
 - CREATE SERVER ステートメント 130
 - CREATE USER MAPPING ステートメント 132
- TIMEFORMAT サーバー・オプション
- 有効な設定値 379
- TIMEOUT サーバー・オプション
- 有効な設定値 379
- TIMESTAMPFORMAT サーバー・オプション
- 有効な設定値 379

U

- UDT (ユーザー定義タイプ)
- サポートされない 15

V

- VARCHAR_NO_TRAILING_
BLANKS サーバー・オプション
- 有効な設定値 379
- VARCHAR_NO_TRAILING_
BLANKS 列オプション
- 有効な設定値 393

X

XML

- コスト・モデル機能 248, 249
 - サポートされるバージョン 3
 - 説明 239
 - ソース文書およびニックネーム
246
 - ニックネーム、有効なオブジェクト 13
 - メッセージ 259
- 連合システムへの追加
- 概要 243
 - サーバーの登録 245
 - 照会の実行 258
 - ニックネームの登録 246, 251
 - 非ルート・ニックネームのビュー連合の作成 257
 - ラッパーの登録 243
- CREATE NICKNAME ステートメント 251, 363
- CREATE SERVER ステートメント 245
- CREATE WRAPPER ステートメント 243
- DB2_DJ_COMM 環境変数の設定 244

IBM と連絡を取る

お住まいの国または地域の IBM 事業所を探すには、www.ibm.com/planetwide の IBM Directory of Worldwide Contacts をお調べください。

製品情報

DB2 Information Integrator に関する情報は、電話または Web で入手可能です。

Web をご利用の場合は、www.ibm.com/software/data/integration にアクセスしてください。このサイトには、技術ライブラリー、資料の注文方法、クライアント・ダウンロード、ニュースグループ、フィックスバック、ニュース、および Web リソースへのリンクに関する最新情報が掲載されています。

お住まいの国または地域の IBM 事業所を探すには、www.ibm.com/planetwide の IBM Directory of Worldwide Contacts をお調べください。



Printed in Japan

日本アイ・ビー・エム株式会社

〒106-8711 東京都港区六本木3-2-12