



ユニバーズデザインツールユーザガイド

■ SAP BusinessObjects Business Intelligence platform 4.1

2013-06-29

著作権

© 2013 SAP AG or an SAP affiliate company. All rights reserved.本書のいかなる部分も SAP AG の明示的許可なしに、いかなる形式、目的を問わず、複写、または送信することを禁じます。本書に記載された情報は、予告なしに変更されることがあります。SAP AGがライセンス、またはその頒布業者が頒布するソフトウェア製品には、他のソフトウェア会社の専有ソフトウェアコンポーネントが含まれています。製品仕様は、国ごとに変わる場合があります。これらの文書は SAP AG およびその関連会社（「SAP グループ」）が情報提供のためにのみ提供するもので、いかなる種類の表明および保証を伴うものではなく、SAP グループは文書に関する錯誤又は脱漏等に対する責任を負うものではありません。SAP グループの製品およびサービスに対する唯一の保証は、当該製品およびサービスに伴う明示的保証がある場合に、これに規定されたものに限られます。本書のいかなる記述も、追加の保証となるものではありません。SAP、および本書で言及されるその他 SAP の製品およびサービス、ならびにそれらのロゴは、ドイツおよびその他諸国における SAP AG の商標または登録商標です。商標に関する情報および表示の詳細については、<http://www.sap.com/japan/company/legal/copyright/index.epx>をご覧ください。

目次

第 1 章	ユニバースデザインツールの概要.....	15
1.1	ドキュメント履歴: ユニバースデザインツール.....	15
1.2	概要.....	15
1.3	ユニバースデザインツールとユニバースの基礎.....	16
1.3.1	ユニバースとは.....	16
1.3.2	ユニバースの役割.....	16
1.3.3	ユニバースに含まれるもの.....	17
1.3.4	ユニバースウィンドウについて.....	19
1.3.5	ユニバースデザインツールインストールルートパス.....	19
1.4	ユニバースデザインツールを使用したユニバースの作成方法.....	20
1.4.1	オブジェクトによる SQL の生成方法.....	20
1.4.2	サポートされるデータベーススキーマの種類.....	21
1.4.3	ユニバースの使い方.....	21
1.5	ユニバース作成者とは.....	22
1.5.1	必要とされるスキルと知識.....	23
1.5.2	ユニバースデザイナのタスク.....	23
1.6	ユニバースを作成するための基本手順.....	24
1.7	ユニバース開発プロセスの概要.....	24
1.7.1	ユニバースのデザイン手法.....	24
1.7.2	ユニバース運用サイクル.....	26
1.7.3	ユニバースの設計、作成時間の最適化.....	27
1.8	複数言語ユニバース.....	28
1.8.1	複数言語ユニバース.....	28
1.8.2	言語およびロケールの定義.....	29
1.8.3	さまざまなロケール.....	30
1.8.4	ユニバースデザインツールのユーザインタフェース用の製品言語の設定.....	30
1.8.5	多言語ユニバースの消費.....	31
1.8.6	リンクされたユニバースのフォールバックロケールの決定.....	31
1.8.7	トランスレーションマネジメントツール.....	31
1.8.8	多言語データ.....	32
1.9	ユニバースデザインツールのサンプル資料.....	32
1.9.1	デモデータベース.....	32
1.9.2	デモユニバース.....	33

1.10	インフォメーションデザインツールでのユニバースの使用.....	33
第 2 章	基本操作の実行.....	35
2.1	概要.....	35
2.2	ユニバースデザインツールの起動.....	35
2.2.1	ユニバースデザインツールを起動する.....	36
2.2.2	クイックデザインウィザードの使い方	37
2.3	Designer XI R3 を使用した XI R2 コネクションとユニバースの操作.....	37
2.4	クイックデザインウィザードを使用した基本的なユニバースの作成.....	38
2.4.1	クイックデザインウィザードの使用する方法.....	38
2.4.2	クイックデザインウィザードの使い方.....	38
2.4.3	クイックデザインウィザードで作成したユニバースの処理.....	45
2.5	ユニバースのインポート.....	45
2.5.1	ユニバースのリポジトリからのインポート.....	45
2.5.2	ユニバースを開くこととインポートすることとの違い.....	46
2.6	ユニバースを開く.....	46
2.7	ユニバースのエクスポート.....	47
2.7.1	リポジトリファイルシステムでのユニバースの構成.....	47
2.7.2	ユニバースのリポジトリへのエクスポート.....	48
2.7.3	エクスポートと保存の違い.....	49
2.8	ユニバースの保存.....	49
2.8.1	識別子としてのユニバースファイル名.....	50
2.8.2	ユニバースの保存.....	50
2.8.3	ユニバースの定義の PDF としての保存.....	50
2.9	ユニバースの終了.....	51
2.10	ワークグループ設計.....	52
2.10.1	ユニバースのロック.....	52
2.10.2	版数.....	52
2.11	ユニバースデザインツールのユーザインタフェースの使用.....	52
2.11.1	ユーザインタフェースの主要コンポーネント.....	53
2.11.2	ユニバースデザインツールのユーザインタフェース.....	54
2.11.3	ウィンドウの操作	54
2.11.4	ツールバーの使い方.....	55
2.11.5	ユニバースデザインツールでのアクションや処理の実行.....	56
2.12	検索と置換の使い方.....	58
2.12.1	検索機能.....	58
2.12.2	クイック検索の使い方.....	61
2.13	テーブル表示の整理.....	61
2.13.1	テーブルの表示方法.....	61
2.13.2	テーブルの操作.....	61
2.13.3	一覧モードの使用.....	62

2.13.4	テーブルの自動整列.....	64
2.13.5	テーブルの表示変更.....	64
2.14	スキーマの表示オプションの選択.....	66
2.14.1	構造枠にグラフィックオプションを設定.....	67
2.14.2	テーブルと列の値の表示.....	68
2.14.3	データベーステーブルの行数の表示.....	70
2.15	ユニバースの印刷.....	74
2.15.1	印刷オプションの設定.....	74
第 3 章	ユニバースの作成とユニバースパラメータの設定.....	77
3.1	ユニバースのパラメータ.....	77
3.2	新しいユニバースの作成.....	78
3.2.1	新しいユニバースを一から作成.....	78
3.3	ユニバースの要約情報の表示と入力.....	80
3.4	ユニバースのパラメータの設定.....	81
3.4.1	ユニバースの識別	81
3.4.2	接続の定義と編集.....	83
3.4.3	ユニバース要約パラメータの設定.....	91
3.4.4	要約情報を表示および入力する方法.....	91
3.4.5	ストラテジーの選択.....	92
3.4.6	リソースの制御.....	96
3.4.7	使用できるシステムリソースオプション.....	97
3.4.8	リソース制御情報を入力する.....	97
3.4.9	複数の SQL 文を生成するクエリでの最大実行時間.....	98
3.4.10	SQL の制限.....	98
3.4.11	ユニバースのリンクに指定するオプション.....	101
3.4.12	SQL 生成パラメータの設定.....	101
3.4.13	SQL 生成パラメータについて.....	103
3.4.14	ユーザインタフェースに設定する SQL パラメータ.....	103
3.4.15	PRM ファイルに設定する SQL パラメータ.....	124
第 4 章	テーブルと結合を使ったスキーマの作成.....	141
4.1	概要.....	141
4.2	スキーマとは.....	141
4.2.1	スキーマのデザインこそが成功するユニバースの鍵.....	142
4.2.2	スキーマのデザインとユニバースの作成プロセス.....	142
4.2.3	スキーマデザインの工程.....	143
4.3	テーブルの挿入.....	143
4.3.1	テーブル参照の使用.....	143
4.3.2	構造枠のテーブルの整列.....	146
4.4	派生テーブルの使い方.....	147

4.4.1	派生テーブルの追加、編集、および削除.....	148
4.5	ネストされた派生テーブル.....	150
4.5.1	派生テーブルエディタの使い方.....	150
4.5.2	ネストされた派生テーブルを作成する.....	151
4.5.3	ネストされた派生テーブルの名前の変更.....	152
4.6	入力列を含むテーブルの使い方.....	152
4.6.1	ハードコーディングされた値の一覧を定義する.....	153
4.6.2	ユーザが入力または選択できる値の一覧を定義する.....	153
4.7	結合の定義.....	154
4.7.1	結合とは.....	154
4.7.2	スキーマ内で結合を使用する意味.....	154
4.7.3	SQL内での結合.....	155
4.7.4	結合する必要があるテーブル.....	155
4.7.5	プライマリキーと外部キーの結合.....	156
4.7.6	結合のカーディナリティについて.....	157
4.7.7	結合の作成.....	157
4.7.8	結合のプロパティ.....	162
4.7.9	結合の編集.....	165
4.7.10	ユニバースの結合の ANSI 92 サポート.....	169
4.7.11	結合の削除.....	172
4.8	特定の結合の種類の定義.....	172
4.8.1	等結合の作成.....	173
4.8.2	シータ結合.....	177
4.8.3	外部結合.....	180
4.8.4	ショートカット結合.....	183
4.8.5	自己制限結合.....	184
4.9	カーディナリティの使い方.....	187
4.9.1	ユニバースデザインツールでのカーディナリティの使用方法.....	188
4.9.2	カーディナリティの手動設定.....	190
4.10	ユニバースのチェック.....	197
4.10.1	ユニバースの整合性の自動チェック.....	198
第 5 章	スキーマでの結合の問題の解決.....	203
5.1	概要.....	203
5.2	結合パスの問題.....	203
5.2.1	ルックアップテーブルとは.....	204
5.2.2	ファクトテーブルとは.....	204
5.2.3	不正確な結果を返す結合パスの種類.....	204
5.2.4	結合問題の検出と解決.....	205
5.3	エイリアスの定義.....	206
5.3.1	スキーマでのエイリアスの使い方.....	206

5.3.2	エイリアスの作成.....	207
5.4	コンテキストの定義.....	210
5.4.1	スキーマでのコンテキストの使い方.....	210
5.4.2	コンテキストの作成.....	211
5.4.3	コンテキストの編集.....	214
5.4.4	コンテキストの削除.....	215
5.4.5	コンテキストの更新.....	216
5.4.6	コンテキストの検出を妨げる結合パス.....	216
5.4.7	コンテキストがクエリに及ぼす影響.....	218
5.5	ループの解決.....	221
5.5.1	ループとは.....	221
5.5.2	ループによるクエリへの影響.....	222
5.5.3	ループの識別の目安.....	228
5.5.4	ループの自動的な識別と解決.....	229
5.5.5	ループの検出と解決のための ユニバースデザインツールの機能.....	229
5.5.6	ループ解決の例.....	238
5.6	キャズムトラップの解決.....	246
5.6.1	キャズムトラップとは.....	246
5.6.2	キャズムトラップの仕組み.....	248
5.6.3	キャズムトラップの検出.....	249
5.6.4	キャズムトラップの解決.....	250
5.7	ファントラップの解決.....	252
5.7.1	ファントラップとは.....	252
5.7.2	ファントラップの検出方法.....	254
5.7.3	ファントラップの解決方法.....	255
5.8	結合問題の視覚的検出.....	258
5.8.1	潜在的なキャズムトラップ.....	259
5.8.2	潜在的なファントラップ.....	260
5.9	ユニバースのチェック.....	261
5.9.1	ユニバースの整合性の自動チェック.....	262
5.9.2	ユニバースの整合性の手動チェック.....	263
5.9.3	ユニバース構造の最新表示.....	266
第 6 章	ユニバースの作成.....	267
6.1	概要.....	267
6.2	ユニバースの作成.....	267
6.2.1	オブジェクトとは.....	268
6.2.2	ユニバースで使用するオブジェクトの種類.....	269
6.2.3	クラスとは.....	269
6.2.4	クラスとオブジェクトの使い方.....	269
6.3	ユニバース枠の使い方.....	270

6.3.1	クラスとオブジェクトまたは条件の表示.....	270
6.4	クラス、オブジェクト、条件の基本操作.....	271
6.4.1	切り取り、コピー、貼り付け.....	271
6.4.2	クラス、オブジェクト、条件の移動.....	271
6.4.3	クラス、オブジェクト、条件の表示と非表示.....	271
6.5	クラスの定義.....	272
6.5.1	クラスの作成.....	273
6.5.2	クラスのプロパティ.....	275
6.5.3	クラスの変更.....	275
6.5.4	サブクラスの使い方.....	276
6.6	オブジェクトの定義.....	276
6.6.1	オブジェクトの作成.....	277
6.6.2	オブジェクトのプロパティ.....	278
6.6.3	オブジェクトの変更.....	279
6.6.4	オブジェクトの定義.....	280
6.6.5	プロパティ.....	283
6.6.6	詳細.....	284
6.6.7	インデックス認識の定義.....	286
6.6.8	ソース情報.....	290
6.6.9	SQL エディタによるオブジェクトの定義.....	291
6.6.10	オブジェクト書式の定義.....	294
6.6.11	オブジェクトの定義に使用したテーブルの表示.....	296
6.6.12	ディメンションの定義.....	296
6.6.13	詳細の定義.....	297
6.6.14	メジャーの定義.....	297
6.6.15	オブジェクトの制限の定義.....	303
6.6.16	条件オブジェクトの定義.....	308
6.6.17	自己結合を使った制限.....	313
6.6.18	複数のテーブルによる制限の適用.....	314
6.6.19	連結オブジェクト.....	316
6.7	階層の定義.....	317
6.7.1	多次元分析とは.....	317
6.7.2	階層の識別方法.....	318
6.7.3	階層の設定.....	319
6.8	階層のカスケード値の一覧の使用.....	322
6.8.1	カスケード値の一覧の作成.....	322
6.9	値の一覧の使い方.....	324
6.9.1	値の一覧の使い方.....	325
6.9.2	オブジェクトに関連付けられた値の一覧.....	326
6.9.3	値の一覧のプロパティとオプション.....	327
6.9.4	値の一覧の編集.....	332

6.9.5	値の一覧のエクスポート.....	335
6.9.6	値の一覧の最新表示.....	338
6.9.7	個人用データファイルからのデータの使用.....	338
6.9.8	ユニバース内の値の一覧の管理.....	340
6.9.9	.LOV ファイルの最適化とカスタマイズ.....	341
6.10	ユニバースのリンク.....	342
6.10.1	リンクされたユニバースとは.....	342
6.10.2	ユニバースをリンクする別の方法.....	344
6.10.3	ユニバースをリンクさせる利点.....	345
6.10.4	ユニバースのリンクの必要条件.....	346
6.10.5	ユニバースをリンクするときの制限.....	346
6.10.6	2 つのユニバース間のリンクの作成.....	347
6.10.7	派生ユニバースの編集.....	349
6.10.8	リンクの削除.....	350
6.10.9	コアユニバースの移動.....	351
6.10.10	派生ユニバースと値の一覧.....	351
6.10.11	コアユニバース内の順番に従ったオブジェクトの表示.....	352
6.11	ユニバースを他のユニバースに含める.....	352
6.11.1	コアユニバースの派生ユニバース内へのコピー.....	352
6.12	ストアドプロシージャユニバースの作成.....	353
6.12.1	Java Bean ユニバースのストアドプロシージャ.....	354
6.12.2	ストアドプロシージャに基づくユニバースの作成.....	354
6.13	ユニバースのテスト.....	358
6.13.1	クエリパネルでのオブジェクトのテスト.....	358
6.13.2	ユニバースの整合性のチェック	359
6.13.3	Web Intelligence でのユニバースのテスト.....	359
第 7 章	ユニバースの最適化.....	361
7.1	概要.....	361
7.2	集計テーブルの使い方.....	361
7.2.1	アグリゲート認識.....	362
7.2.2	アグリゲート認識のデータウェアハウスへの適用.....	362
7.2.3	アグリゲート認識の設定.....	362
7.2.4	オブジェクトの作成.....	364
7.2.5	集計オブジェクトのすべての組み合わせの確認.....	364
7.2.6	集計レベル順のオブジェクトの整列.....	364
7.2.7	@Aggregate_Aware 関数を使った集計オブジェクトの定義.....	365
7.2.8	非互換オブジェクトの指定.....	367
7.2.9	非互換オブジェクトの指定.....	370
7.2.10	集計テーブルのループの解決.....	372
7.2.11	アグリゲート認識のテスト.....	374

7.3	オブジェクトの SQL での @ 関数 の使用.....	375
7.3.1	オブジェクトへの @ 関数の挿入.....	376
7.3.2	@Aggregate_Aware.....	377
7.3.3	@Prompt.....	378
7.3.4	@Script.....	391
7.3.5	@Select.....	393
7.3.6	@Variable.....	394
7.3.7	@Where.....	400
7.4	外部ストラテジーを使用したユニバース作成のカスタマイズ.....	402
7.4.1	ユニバースデザインツールへの外部ストラテジーの移行.....	403
7.4.2	外部ストラテジーの概要.....	403
7.4.3	外部ストラテジーの概要.....	404
7.4.4	外部ストラテジーの Help テキストの作成.....	405
7.4.5	外部ストラテジーファイルの宣言の確認.....	407
7.4.6	サンプルの外部ストラテジーの使用.....	408
7.4.7	ストラテジーファイル(STG)の構造.....	408
7.4.8	ストラテジーの出力フォーマット.....	410
7.4.9	外部ストラテジーの作成.....	414
7.4.10	データのテキストファイルの作成.....	416
7.4.11	ユニバースデザインツールへの外部ストラテジーの適用.....	416
7.5	分析関数の使い方.....	417
7.5.1	分析関数とは.....	418
7.5.2	分析関数の利点.....	418
7.5.3	サポートされる分析関数ファミリ.....	418
7.5.4	ユニバースデザインツールでの分析関数の使用方法.....	419
7.5.5	IBM DB2 UDB および Oracle.....	420
7.5.6	RedBrick(RISQL関数).....	424
7.5.7	Teradata(OLAP 関数).....	427
7.5.8	SELECT 文への構文の自動読み込み.....	429
7.6	SQL プレフィックス関数.....	431
7.6.1	BEGIN_SQL ユニバースパラメータで SQL 文にプレフィックスを付ける.....	431
7.7	Array Fetch パラメータの最適化.....	432
7.7.1	Array Fetch/パラメータの変更.....	432
7.8	テーブルウェイトの設定.....	433
7.8.1	PRMファイル内のテーブルウェイトの変更.....	433
7.9	テーブルから返される行数の変更.....	433
7.9.1	返される行数の変更.....	434
7.10	ショートカット結合の使用.....	434
第 8 章	OLAP ユニバースの使用.....	435
8.1	OLAP ユニバースについて.....	435

8.1.1	OLAP ユニバースとは.....	435
8.1.2	ユニバース作成に使用できる OLAP データソース	436
8.2	OLAP データソースへの接続の定義.....	442
8.2.1	OLAP データソースへの接続.....	442
8.2.2	新規接続ウィザードを開始する.....	443
8.2.3	OLAP 接続用のデータベースミドルウェアを選択する.....	443
8.2.4	SAP BW OLAP 接続のログインパラメータ.....	444
8.2.5	MSAS OLAP 接続のログインパラメータ	446
8.2.6	Essbase 接続のログインパラメータの定義.....	447
8.2.7	OLAP 接続を行うためのソースキューブまたはクエリを選択する.....	448
8.2.8	OLAP 接続用の設定パラメータを定義する.....	449
8.2.9	Essbase 接続のカスタムパラメータの定義.....	450
8.3	OLAP ユニバースのカスタマイズ.....	451
8.3.1	追加パラメータを持つ OLAP ユニバースの作成.....	451
8.3.2	OLAP ユニバースの OLAP オプションの定義.....	452
8.3.3	OLAP ユニバースのオブジェクトの定義.....	453
8.3.4	ユニバースデザインツールの機能は OLAP ユニバースでサポートされています。.....	453
8.3.5	依頼されたデータベース投影関数.....	454
8.3.6	OLAP ユニバースの依頼されたメジャーの設定.....	456
8.3.7	メジャーに集計機能を設定.....	457
8.3.8	OLAP ユニバースの計算メジャー.....	458
8.3.9	キューブクエリの MDX 関数について.....	459
8.3.10	フィルタと WHERE 文のための XML 構文.....	460
8.3.11	OLAP ユニバースに含まれる定義済みの条件.....	461
8.3.12	OLAP ユニバースに含まれるオプションプロンプト.....	466
8.3.13	SAP NetWeaver BW ユニバースに対する特定のクエリのパフォーマンスを向上させる.....	466
8.4	OLAP ユニバースのライフサイクル管理.....	467
8.4.1	OLAP ユニバースのライフサイクル管理.....	467
8.4.2	概要: ユニバースオブジェクトのステータスと OLAP オブジェクトのステータスの関係	468
8.4.3	OLAP ユニバースを最新表示する.....	471
8.4.4	OLAP ユニバースのレベル 00 の再生成.....	474
8.4.5	レベル L00 の ALL への名前変更.....	474
8.4.6	OLAP ユニバースレベル前記号の置換.....	475
8.4.7	ユニバースと OLAP キューブの同期化.....	475
8.4.8	OLAP ユニバースの更新におけるディメンションの管理方法.....	476
8.4.9	OLAP ユニバースの更新における階層または特性の管理方法	481
8.4.10	OLAP ユニバースの更新におけるレベルの管理方法	488
8.4.11	OLAP ユニバースの更新における SAP 変数の管理方法	492
8.4.12	OLAP ユニバースの更新におけるキー数値またはメジャーの管理方法	496
8.4.13	OLAP ユニバースの更新における SAP キー日付の管理方法	501
8.5	さまざまな OLAP キューブをユニバースにマップする方法.....	503

8.5.1	ユニバースにおける SAP NetWeaver BW オブジェクトのマッピングおよび使用方法.....	503
8.5.2	Essbase キューブをユニバースコンポーネントにマップする方法.....	512
8.5.3	MSAS キューブをユニバースコンポーネントにマップする方法	513
第 9 章	メタデータソースからのユニバースの操作.....	515
9.1	メタデータソースからのユニバースの生成について.....	515
9.2	概要.....	515
9.3	ユニバース作成の概要.....	516
9.4	メタデータソースの選択.....	516
9.5	XML ソースからのユニバースの作成.....	517
9.5.1	XML メタデータソース.....	517
9.5.2	XML メタデータソースからユニバースを生成する	518
9.5.3	接続とユニバースのオプションの選択.....	519
9.5.4	XML メタデータソースからユニバースを更新する.....	520
9.6	ユニバースの DB2CV へのエクスポート.....	521
9.6.1	ユニバースをエクスポートする際の前提条件と制限事項.....	521
9.6.2	ユニバースメタデータの識別.....	522
9.6.3	DB2CV XML ファイルへのユニバースのエクスポート.....	523
9.6.4	DB2CV メタデータへのユニバースのマップ.....	523
9.6.5	特定の SQL 表現式のマップ.....	527
9.7	Oracle アナリティクスワークスペース.....	529
9.7.1	OLAP キューブから生成されたユニバースについて.....	529
9.7.2	Oracle OLAP 構造のユニバースコンポーネントへのマップ.....	530
9.7.3	リレーショナルビューの分析.....	530
9.7.4	ユニバースでショートカット結合を使用する目的.....	531
9.7.5	ユニバースコンポーネントにマップされた Oracle OLAP 構造について.....	533
9.7.6	ビューを作成してユニバースを生成.....	537
9.7.7	Oracle Analytic Workspace からユニバースを作成および表示するオプション.....	537
9.7.8	ビューの作成とユニバースの生成.....	538
9.7.9	Oracle Analytical Workspace からのみのビューの作成.....	539
9.7.10	既存のビューからのユニバースの生成.....	540
9.7.11	既存のビューからユニバースを生成する.....	540
第 10 章	ユニバースの展開.....	541
10.1	概要.....	541
10.2	ユニバースのデプロイ方法.....	541
10.2.1	リポジトリでのユニバースの識別.....	541
10.3	すべてのユーザへのユニバースのアクセス権の付与.....	542
10.4	ユニバースへのアクセス制限の設定.....	543
10.4.1	制限とは.....	543
10.4.2	ユニバースに適用できる制限.....	544

10.4.3	アクセス制限の管理.....	544
10.4.4	制限の作成.....	546
10.4.5	ユニバースへのアクセス制限の適用.....	548
10.4.6	ユニバースの利用可能なユーザの一覧へのユーザグループの追加.....	549
10.4.7	制限グループ優先順位の設定.....	550
10.4.8	ユーザとグループセキュリティ制限の表示.....	551
10.5	ユーザとログインの管理.....	552
10.5.1	ログインの管理.....	552
10.5.2	パスワードの管理.....	553
第 11 章	サンプル資料の使い方.....	555
11.1	サンプル資料の使い方 - 概要.....	555
11.2	Clubデータベース.....	555
11.2.1	テーブルの構成.....	555
付録 A	より詳しい情報.....	565
	索引.....	567

ユニバースデザインツールの概要

1.1 ドキュメント履歴: ユニバースデザインツール

以下の表は、最も重要なドキュメント変更の概要です。

バージョン	日付	説明
SAP BusinessObjects ユニバースデザインツール 4.0	2010 年 11 月 30 日	このドキュメントの初版です。 Universe Designer は「ユニバースデザインツール」という名前に変更されました。
SAP BusinessObjects ユニバースデザインツール 4.0 Service Pack 1	2011 年 2 月 25 日	
SAP BusinessObjects ユニバースデザインツール 4.0 Service Pack 2	2011 年 6 月 15 日	接続オブジェクトに “接続をローカルにダウンロード” というシステム管理者定義のセキュリティ権限が追加されました。
SAP BusinessObjects ユニバースデザインツール 4.0 機能パック 3	2012 年 2 月 20 日	

1.2 概要

この章では、ユニバースの作成に使用するツールであるユニバースデザインツールの概要について説明します。ユニバースとは何か、ユニバースに含まれる情報、ユニバースの作成方法、および実際のビジネス環境内でのユニバースの役割などが説明されています。

ユニバースの典型的な開発サイクル、推奨デザイン方法、およびこのリリースで出荷されるデモ用のデータベースとユニバースについて説明します。

この章では、ユニバースデザインツール、開発プロセス、およびユニバースに使用できる各種の言語について説明します。次の項目について説明します。

関連項目

- ・ 16 ページの[ユニバースデザインツールとユニバースの基礎](#)」
- ・ 20 ページの[ユニバースデザインツールを使用したユニバースの作成方法](#)」
- ・ 22 ページの[ユニバース作成者とは](#)」
- ・ 24 ページの[ユニバース開発プロセスの概要](#)」
- ・ 28 ページの[複数言語ユニバース](#)」

1.3 ユニバースデザインツールとユニバースの基礎

Business Objects ユニバースデザインツールは、Web Intelligence および Desktop Intelligence ユーザが使用するユニバースを作成するためのソフトウェアツールです。

1.3.1 ユニバースとは

ユニバースとは、次の定義を保存したファイルです。

- ・ 1 つまたは複数のデータベースミドルウェアへの接続パラメータ。
- ・ 列、テーブル、データベース関数といったデータベース内の実際のSQL構造に関連付けられた、オブジェクトと呼ばれるSQL構造オブジェクトはクラスにグループ化されます。オブジェクトとクラスはどちらも Web Intelligence ユーザーに表示されます。
- ・ データベース内で使われるテーブルと結合のスキーマオブジェクトは、スキーマに組み込まれたデータベース構造から作成されます。スキーマは、ユニバースデザインツール ユーザのみが使用できます。Web Intelligence ユーザーや Desktop Intelligence ユーザーには、スキーマは表示されません。

Web Intelligence ユーザーはユニバースに接続し、データベースに対するクエリーを実行します。ユニバース内に定義されたオブジェクトを使うことで、データベース内のデータ構造を意識せずにレポートを作成し、データを分析することができます。

1.3.2 ユニバースの役割

ユニバースの役割は、技術的な知識を持たない Web Intelligence ユーザーでも、簡単にデータベースにクエリーを実行し、レポートを作成してデータ分析ができるような、わかりやすく、使いやすいインターフェイスを提供することです。

ユニバースデザイナーの役割は、ユニバースデザインツールを使用して、たとえば列やデータベース関数など、ユーザがビジネス要件を満たすために必要な情報を取得するためにアクセスし照会するデータベース構造を表現したオブジェクトを作成することです。

ユニバース内に作成するオブジェクトは、エンドユーザのビジネス環境に密着した内容を使い慣れたビジネス用語で表現したものでなければなりません。オブジェクトは、データベース内の SQL 構造をビジネスの観点から表現し直したものだと言えます。

1.3.3 ユニバースに含まれるもの

ユニバースには、次の構造が含まれます。

- ・ クラス
- ・ オブジェクト

1.3.3.1 クラス

クラスとは、ユニバース内のオブジェクトを論理的にグループ化したものです。オブジェクトのカテゴリを表します。一般的にクラスには、そこに含まれるオブジェクトのカテゴリを反映した名前を付けます。また、クラスはさらにサブクラスに細分化できます。

1.3.3.2 オブジェクト

オブジェクトとは、データベース内のデータ、またはその派生データをマップした名前の付いたコンポーネントです。オブジェクトの名前には、それを使用するユーザグループの業務に応じた用語を使います。たとえば、製造管理担当者の使用するユニバースでは、“製品”、“製品寿命”、“製造日”などのオブジェクトが利用されます。財務分析担当者の場合は、“利幅”、“投資収益率”などのオブジェクト名を使うことが考えられます。

1.3.3.3 オブジェクトの種類

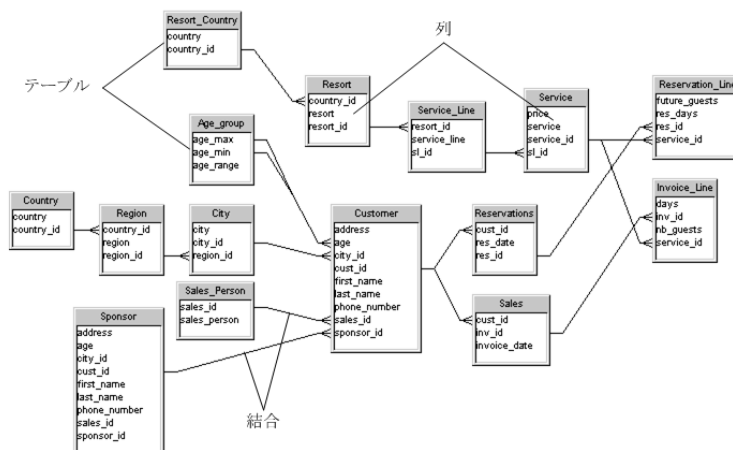
ユニバースデザインツールでは、オブジェクトにディメンション、詳細、メジャーのいずれかの資格を設定します。

オブジェクトの種類	説明
ディメンション	分析のパラメータディメンションオブジェクトはデータ分析のパラメータとなるもので、地理、商品、時間などの階層に対応したものとなります。例: "名前"、City_Id
詳細	詳細オブジェクトはディメンションオブジェクトの内容を説明するもので、データ分析では使用しません。例: 電話番号
メジャー	メジャーオブジェクトはディメンションオブジェクトの計測に使う数値データを持ったオブジェクトです。例: "売上げ"

1.3.3.4 スキーマの SQL 構造を投影したオブジェクト

Web Intelligence ユーザーにユニバースで表示されるオブジェクトでは、データベーススキーマに挿入された SQL 構造が生成されます。スキーマは、ユーザーのレポート作成とデータ分析に必要なデータを返すテーブルと結合をもとに、ユニバース作成者が作成します。

スキーマはユニバースファイルの一部ですが、スキーマを表示し、アクセスできるのはユニバースデザインツールだけです。スキーマの作成には、[ユニバース]ウィンドウの[構造]枠を使います。次のスキーマは、サンプルユニバースBeach.unvのものです。

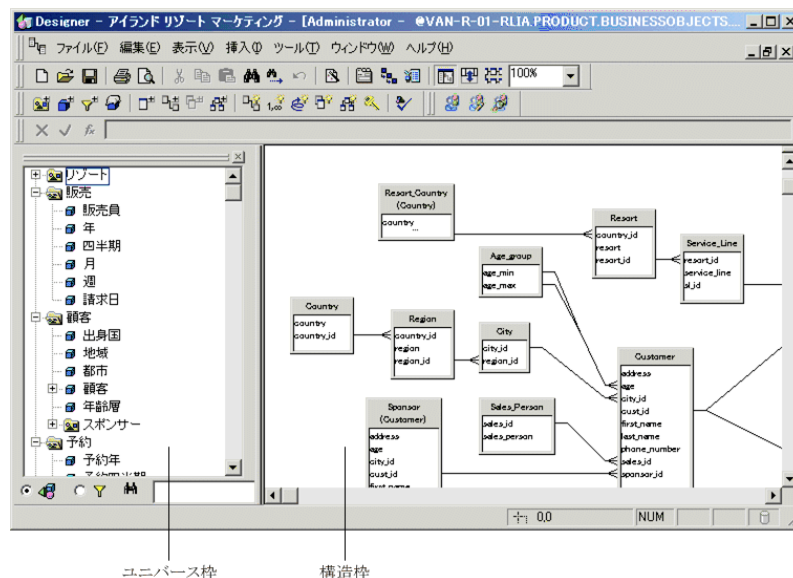


1.3.3.5 ユニバース内のオブジェクトの表示

オブジェクトは、[ユニバース]枠のツリー構造で表示されます。このオブジェクトエクスプローラを使って、クラスとオブジェクトを作成、削除、コピー、表示、移動します。

1.3.4 ユニバースウィンドウについて

ユニバースデザインツールの[ユニバースウィンドウ]を以下に示します。ウィンドウには、[ユニバース] ペイン (Web Intelligence でも表示) と[構造] ペイン (ユニバースデザインツールのみで表示) があります。



1.3.5 ユニバースデザインツールインストールルートパス

このガイドでは、\$INSTALLDIR 変数は、ユニバースデザインツールおよび Web Intelligence によって使用されるデータアクセスファイルのインストールルートパスを示します。これは、オペレーティングシステムサブディレクトリを伴う Business Objects のインストールパスで、ユニバースデザインツール実行ファイルとデータアクセスドライバを含みます。

Windows の場合、Windows\$INSTALLDIR は ¥¥...¥Business Objects¥BusinessObjects Enterprise 12.0¥win32_x86 です。

例: C:¥Program Files¥Business Objects¥BusinessObjects Enterprise 12.0¥win32_x86

1.4 ユニバースデザインツールを使用したユニバースの作成方法

ユニバースデザインツールには、データベースミドルウェアに接続可能な接続ウィザードが提供されています。ユニバースデザインツールを使用して複数の接続を作成できますが、各ユニバースに設定できる接続は 1 つだけです。このデータベース接続は、ユニバースと共に保存されます。

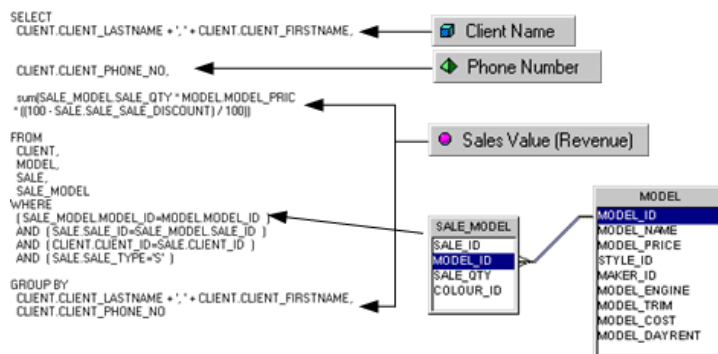
ユニバースデザインツールには、データベース内のテーブルを選択および表示するためのグラフィカルインタフェースがあります。データベース テーブルは、スキーマ内にテーブル シンボルを使って表示されます。このインタフェースを使用して、スキーマ内でのテーブルの操作、テーブルをリンクする結合の作成、エイリアス テーブルやコンテキストの作成、ループの解決を実行できます。Web Intelligence ユーザには、このスキーマは表示されません。

ユニバースデザインツールには、オブジェクトのツリー構造、つまりツリーエクスプローラもあります。ツリー エクスプローラを使用して、スキーマビューに表示される列や SQL 構造にマップされたオブジェクトを作成します。Web Intelligence ユーザーは、これらのオブジェクトを操作して、データベースに対するクエリーを実行します。

ユニバースデザインツールでは、Central Management System (CMS) リポジトリにユニバースをエクスポート/インポートすることでユニバースを配布します。

1.4.1 オブジェクトによる SQL の生成方法

Web Intelligence ユーザーは、オブジェクトを[クエリー]作業領域にドラッグしてクエリーを作成します。それぞれのオブジェクトの定義から SELECT 句が生成されます。クエリーを実行すると、すべてのオブジェクトについて SELECT 句と必要な WHERE 句が指定されたデータベースに対して実行されます。

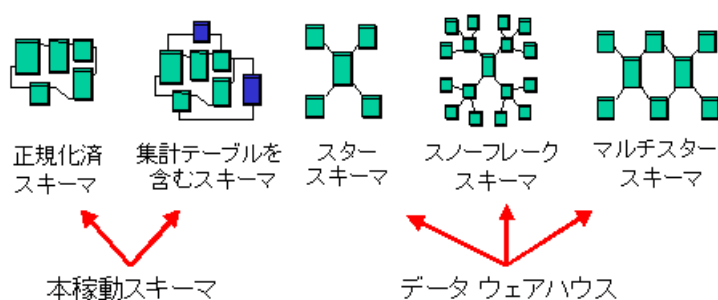


[クエリ]作業領域で、ユーザがディメンションオブジェクトあるいは詳細オブジェクトとメジャーオブジェクトを選択すると、ディメンションおよび詳細オブジェクトの内容を含んだ Group By 句が、自動的に SELECT 文に追加されます。

WHERE 句内の結合と From 句に含まれるテーブルは、[構造]枠内で構築したテーブルスキーマで定義されたものです。

1.4.2 サポートされるデータベーススキーマの種類

ユニバースデザインツールでは、下図に示すデータベーススキーマをすべてサポートしています。ユニバースデザインツールの使用にあたってデータベースの再定義、最適化の必要はありません。



1.4.3 ユニバースの使い方

ユニバースは、Web Intelligence のエンドユーザーが使用します。ユニバースは、Central Management Server (CMS)リポジトリに保存されます。エンドユーザは Web ブラウザからユニバースにアクセスします。

データベースへの接続はユニバース内に定義されているため、エンドユーザはユニバースに接続するだけで自動的にデータにアクセスできます。さらにデータへのアクセスは、ユニバース内に定義されたオブジェクトによって制限されます。これらのオブジェクトは、特定のユーザグループを対象にユーザのニーズに従ってユニバースデザイナーが作成したものです。

1.4.3.1 目標データの表現

ユニバースは、特定のアプリケーション、システム、ユーザグループが持つデータの必要性を表現するものです。たとえば、マーケティング部門、経理部門などの企業内の部署別のデータニーズをユニバースで表現することができます。

部署内のセクションとか、給与システムや在庫システムなどの一連の作業手順など、さらに小さな業務単位をユニバースで表現することもできます。

クラスの例として、従業員情報、就業情報、部署情報があります。

1.4.3.2 ユニバースとデータベーススキーマ

データベーススキーマを使用して、PERSONNEL、INVENTORY、および SALES の3つのユニバースを作成します。それぞれのユニバースにクラスとオブジェクトが定義されており、それぞれのオブジェクトはデータベース構造の特定の部分にマップされています。

1.4.3.3 ユニバースの使用者

ユニバースは、Web Intelligence のユーザーがレポートの作成と分析に使用します。したがってユニバースは、ユーザーの業務内容に対応したクラスとオブジェクトを提供する必要があります。

1.5 ユニバース作成者とは

ユニバースは、ユニバースデザイナーがユニバースデザインツールを使用して作成します。典型的なユニバースデザイナーのプロフィールというものはありません。企業内では一般的に、データベース管理、アプリケーション管理またはアプリケーション開発、プロジェクト管理などの担当者、または、他のユーザのためにユニバースを作成するのに必要なスキルを身に付けたレポート作成者がユニバースデザイナーとなります。

社内に複数のユニバースデザイナーがいることは珍しくありません。必要なデザイナーの人数はそれぞれの会社のデータニーズに応じて異なります。たとえば、デザイナーをアプリケーション、プロジェクト、部門、業務範囲別に1人ずつ配置することもできます。

複数のデザイナーがユニバースを作成する場合は、用語に関するルールセットまたはガイドラインを定義して、オブジェクトの表現に整合性が保たれるようにすることが重要です。

1.5.1 必要とされるスキルと知識

ユニバースデザイナーには、次のスキルと技術レベルが要求されます。

スキル/知識	説明
ユーザのニーズを分析する能力	ユニバースは、ユーザのデータニーズを満たすために作成されるものです。したがってユニバースデザイナーには、実際にエンドユーザが必要とする情報の種類を分析し、ユーザが使う言葉で表現したクラスとオブジェクトを作成して、エンドユーザのニーズに適応したユニバースを開発する能力が要求されます。つまり、エンドユーザが必要とするデータ分析に適応したクエリ結果とレポート作成を判断できる必要があります。
データベースに関する知識	ユニバースデザイナーには、社内のデータベース管理システム(DBMS)、その動作環境、データベースに保存されているデータの型と論理的構造についての十分な業務知識が要求されます。
SQL (Stuctured Query Language)	SQLについての業務知識が要求されます。

1.5.2 ユニバースデザイナーのタスク

ユニバースデザイナーは通常、次のタスクについての責任を負います。

- ・ ユーザのニーズ分析
- ・ ユニバースのデザインと作成
- ・ ユニバースの配布
- ・ ユニバースのメンテナンス

1.6 ユニバースを作成するための基本手順

ユニバースは、レポートツールを使ってレポートを作成するエンドユーザが使用する、.unv ファイルの一部です。ユニバースは、エンドユーザに表示される.unv ファイルの唯一の部分です。ユニバースを作成するには、以下の手順を実行します。

- 1 新しい(空の)ユニバースファイルを作成し、接続およびパラメータの詳細を設定します。
- 2 使用するデータベーステーブルのスキーマを作成し、テーブル間の結合を定義します。スキーマは、レポートツールを使用してレポートを作成するエンドユーザに表示されません。
- 3 スキーマ内の結合問題を解決します。
- 4 ユニバースを構成するクラスとオブジェクトを作成します。ユニバースは、レポートツールを使用してレポートを作成するエンドユーザが使用します。

関連項目

- ・ 35 ページの[基本操作の実行](#)
- ・ 77 ページの[ユニバースの作成とユニバースパラメータの設定](#)
- ・ 141 ページの[テーブルと結合を使ったスキーマの作成](#)
- ・ 203 ページの[スキーマでの結合の問題の解決](#)
- ・ 267 ページの[ユニバースの作成](#)
- ・ 361 ページの[ユニバースの最適化](#)

1.7 ユニバース開発プロセスの概要

下記のセクションでは、ユニバースを手動で作成する方法の概要と典型的なユニバース運用サイクルにおけるユニバース作成の位置付けを説明します。

1.7.1 ユニバースのデザイン手法

本ガイドで紹介するユニバースのデザイン手法は、準備段階と 3 つの実装段階で構成されています。

- ・ 業務問題の分析とユニバースの作成計画
- ・ スキーマの設計
- ・ ユニバースの作成
- ・ ユニバースのユーザへの配布

それぞれの段階に進むには、あらかじめ初期設計を行う必要があります。計画フェーズではユニバースデザインツールを使用する必要はありませんが、ユニバースの質に大きく影響するフェーズです。初期設計段階でユーザのレポート要求を十分に検討しなかったユニバースは、デザイン、作成、メンテナンスも難しく、ユーザにとっても使いにくいものとなります。

次に、それぞれの段階を説明します。

1.7.1.1 ユニバースデザインツールの使用を開始する前のユニバースの計画

第1段階に取り掛かる前に、ユニバースの作成プロジェクトに割り当てられた時間の80%までをユニバースの初期設計に当てます。ここでは、次のポイントに注意してください。

- ・ ユニバースの使用対象となるエンドユーザのレポート作成とデータ分析のニーズを十分に検討してください。スキーマの作成に使用するデータ構造は、関連するテーブルや列に含まれるデータに対して明確に定義されたユーザのニーズに基づいている必要があります。
- ・ ユニバースデザインツールを使用する前に、作成する必要があるオブジェクトについて明確な計画を立てる必要があります。データベース内の列を見ながらオブジェクトを作成するのではなく、ユーザニーズの分析によって確認したオブジェクトに一致する列を識別することが必要です。

1.7.1.2 スキーマの設計

ユニバースの元となるデータベース構造のスキーマを作成します。このスキーマには、データベースのテーブルと列、それらをリンクする結合が含まれます。構造内で発生するループ、キャズムトラップ、ファントラップなどの結合の問題をエイリアスやコンテキストを使って解決することが必要な場合もあります。構造全体の整合性をチェックすることも必要です。スキーマのデザインの詳細については、本ガイドの141 ページの「[テーブルと結合を使ったスキーマの作成](#)」と203 ページの「[スキーマでの結合の問題の解決](#)」の章を参照してください。

1.7.1.3 ユニバースの作成

作成したスキーマのコンポーネントを元に、SELECT 句を生成するオブジェクトを作成します。オブジェクトはクラス別にグループ化します。作成するオブジェクトは、ユーザのレポートニーズの分析結果にしたがって識別したオブジェクトです。さまざまな種類のオブジェクトを作成することで、ユーザに提供するレポート機能や多次元分析を拡張したり、クエリのパフォーマンスを最適化することができます。

ユニバース構造の整合性をチェックする必要があります。Web Intelligence でレポートを実行しテストをする必要もあります。

ユニバース構築の段階については、267 ページの「[ユニバースの作成](#)」章を参照してください。

1.7.1.4 ユニバースの配布

ユニバースは、Central Management System(CMS)リポジトリにエクスポートすることで、テストまたは業務を目的としてユーザに配布されます。この段階については、541 ページの「[ユニバースの展開](#)」の章を参照してください。

1.7.2 ユニバース運用サイクル

ユニバースの開発は、設計、デザイン、作成、配布、メンテナンスで構成されるサイクルプロセスです。ユニバースの設計と作成にはユニバースデザインツールを使用しますが、ユニバースの操作性は、開発サイクルの他のフェーズ間の相互の連携に関係します。

この節では、ユニバース開発プロジェクトの設計と実施に利用できるデザイン手法の概要を説明します。

次の表は、典型的なユニバース開発サイクルにおける主要ステップの概要をまとめたものです。

開発サイクル	説明
準備	<ul style="list-style-type: none">・ 対象となるデータソースを識別し、そのデータ構造を調べる。・ 対象となるそれぞれのデータベースの各テーブルに含まれるデータを認識する。・ 結合を理解する。・ カーディナリティを識別する。・ 可能性を検討する。
分析	<ul style="list-style-type: none">・ ユーザを識別し、その構成を調べる。たとえばユーザグループが部門別か、業務内容別か、など。・ ユーザが必要とする情報を識別する。・ ユーザが必要とする標準レポートを識別する。・ オブジェクトに適切な名前を付けられるよう、ユーザが実際に使用している用語を習得する。

開発サイクル	説明
設計	プロジェクトの戦略を識別する。たとえば、作成するユニバース数やどのユニバースをどのレベルにリンクさせるか、など。
実装	<ul style="list-style-type: none"> ユニバースデザインツールを使用してユニバースを作成する。このガイドではユニバース開発サイクルのこの部分での、ユニバースデザインツールの具体的な使用方法について説明しています。 作成プロセス中に、生成される SQL の有効性と信頼性を繰り返しテストする。
テスト	ユニバースから取得すべき情報を理解している少数の Web Intelligence パワーユーザのグループを形成する。ユニバースの実際の利用を想定した完全なテストを行うよう、ユーザに要請する。
デプロイ	対象となるエンドユーザがアクセスできる Central Management System(CMS)リポジトリに、ユニバースをエクスポートし、配布する。
発展	データソースとユーザニーズの変化や発展にしたがってユニバースのアップデートとメンテナンスを行う。

注

ユニバースのデザインでは、データソースの構造ではなく、ユーザの要求を第一に考えることが重要です。

1.7.3 ユニバースの設計、作成時間の最適化

ユーザ要求の分析とデザインは、開発プロセスの最重要ステップです。オブジェクト名にユーザのビジネス用語を使い、ユーザが必要とするデータに簡単にアクセスできるユニバースにするには、ユーザの開発プロセスへの参加が必要です。

設計までの初期ステップが正しく実行されれば、実装は簡単で、それにかかる時間も短くなります。

ユニバースの開発期間の80%までは、次の初期ステップに費やすべきです。

- ・ 準備
- ・ 分析

- ・ 設計

時間をかけてユニバースの基礎を固めることによって、計画と分析に必要な時間をかけなかった場合に比較して、実際にユニバースデザインツールを使用してユニバースを作成する残りの 20% の時間の生産性を向上できます。

1.8 複数言語ユニバース

1.8.1 複数言語ユニバース

SAP BusinessObjects Enterprise XI 4 の主要な機能の 1 つは、同じユニバースから多言語のメタデータおよびレポートを作成できる機能です。この機能により、ユーザは、ロケールに依存した、1 つのメタデータユニバースモデルによってサポートされる多言語のレポートをワンステップで作成できるようになり、完全な Unicode サポートが提供されます。レポートは、同じユニバースから構築した後で、ユーザ設定に基づいて複数の言語で表示できます。

翻訳できるユニバースメタデータは、次のとおりです。

- ・ ユニバース名
- ・ ユニバースの説明
- ・ クラス名
- ・ オブジェクト名
- ・ オブジェクト名、説明および書式
- ・ カスタム階層名
- ・ プロンプトおよび入力列の質問

注

ユニバースメタデータで定義されているプロンプトのみを翻訳できます。@Prompt 関数を使用して定義されているプロンプトは翻訳できません。

ユニバースには複数のロケールの翻訳を含めることができます。ユニバース上でレポートを作成する場合、ロケールに表示されるメタデータは、ユーザの優先表示ロケールによって異なります。

ユニバースは、ロケールがない場合に使用されるロケールである、フォールバックロケールも定義します。

ステータスが [使用可能] に設定されているロケールのみが、ユニバース上でレポートを作成するユーザに表示されます。

これらの表示可能なロケールで、ステータスが [翻訳を表示] カテゴリのメタデータのみが、ユニバース上でレポートを作成するユーザに表示されます。[翻訳を表示] カテゴリには、次のステータスのメタデータが含まれません。

- ・ NEEDS_REVIEW_TRANSLATION

- ・ NEEDS_REVIEW_LOCALIZATION
- ・ NEEDS_REVIEW_ADAPTATION
- ・ TRANSLATED
- ・ FINAL
- ・ SIGNED_OFF

ユニバースメタデータの翻訳、ユニバースロケールとメタデータステータスの設定は、トランスレーションマネジメントツールを使用して実行されます。翻訳とロケールパラメータは、追加の XML ストリームに保存され、.unv ファイル形式で保存されます。

ユニバースデザインツールのユーザインタフェースも多言語で表示できます。次のテキストは、ユニバースデザインツールの多言語ユニバース機能についての説明です。

1.8.2 言語およびロケールの定義

いくつかの言語は、複数の国に関連付けられています。たとえば、フランス語 (fr) は、フランス (FR)、ベルギー (BE)、およびスイス (CH) の各国で話されている言語です。この例の fr-FR、fr-BE、fr-CH は、フランス (FR)、ベルギー (BE)、およびスイス (CH) の言語としてのフランス語 (fr) を意味します。

言語	国
フランス語	フランス
フランス語	ベルギー
フランス語	スイス

同時に、1 つの国が複数の言語に関連付けられている場合もあります (fr-CH、de-CH、it-CH)。たとえば、スイスでは、ドイツ語、フランス語、およびイタリア語がすべて話されています。

言語	国
フランス語	スイス
ドイツ語	スイス
イタリア語	スイス

ロケールは、言語と地理的地域の組み合わせ、およびデータの並べ替え方法を定義します。日付と時刻は多くの場合、特定の書式に設定されます。たとえばフランス語とフランスなど、言語と国を組み合わせた場合、ロケールは、オペレーティングシステムまたはアプリケーションに応じて次のように表示されます。

オペレーティングシステム	ロケールの書式
Windows	フランス語 (フランス) ロケールはシステム設定 (国) から取得されます。
Java	fr_FR
Sun Solaris	fr_FR.ISO8859-1
Infoview 6.5.1	フランス語 (フランス)[fr-FR]

ドキュメントとアプリケーションを単純化するために、言語とロケールの両方の意味に「言語」という用語を使用します。

1.8.3 さまざまなロケール

用語	定義
製品言語	これは、ユニバースデザインツールのユーザインタフェースの言語です。この言語でメニューおよびメッセージが表示されます。
優先表示ロケール (PVL)	優先表示言語の設定です。この設定は、リソース (ドキュメントまたはユニバース) コンテンツまたは属性一覧を構成する文字列やテキスト、書式が、InfoView または Web Intelligence リッチクライアントのアプリケーションに表示されるとききのロケールを定義します。
置換 (フォールバック) ロケール	優先表示ロケールがない場合に使用されるロケールです。
ソース言語	ドキュメントが作成されたときのロケールです。

1.8.4 ユニバースデザインツールのユーザインタフェース用の製品言語の設定

ユニバースデザインツールの[ツール]の[一般設定]タブ > [オプション]設定で、使用できる言語の一覧から言語を選択します。これは、製品言語で、ユーザインタフェース言語(UIL)とも呼ばれます。ユニバースメタデータに対する影響はありません。オブジェクト名、コンテキスト名、およびクラスは、元のデータベース要素の言語で表示されます。ユニバースメタデータを翻訳するには、トランスレーションマネジメントツールを使用します。

1.8.5 多言語ユニバースの消費

ユーザが、Web Intelligence で多言語のユニバース上にレポートを作成する場合、たとえば、メタデータが表示されるロケールは、ロケールの可用性/ステータスとメタデータのカテゴリによって異なります。

- ・ ロケールでのメタデータの翻訳は、ステータスが[表示]カテゴリで、このロケールが[使用可能]と定義されている場合のみ表示されます。
- ・ メタデータの翻訳を表示するロケールは、次の優先順位に従います。
 - ・ ユーザの優先順位ロケール。
 - ・ 優先順位ロケールがない場合は、このユニバースで定義されているフォールバックロケール。
 - ・ このユニバースでのフォールバックロケールが定義されていない場合は、ユーザの優先表示ロケールで最優先されるロケール。
 - ・ ロケールがない場合は、元のコンテンツが表示されます。この元のコンテンツは、ユニバースデザインツールで定義されたメタデータです。

1.8.6 リンクされたユニバースのフォールバックロケールの決定

派生ユニバースは、異なるコアユニバースからのメタデータを再利用することがあります。派生ユニバースとコアユニバースが異なるフォールバックロケールを定義する場合、

- ・ 派生ユニバースレベルでフォールバックロケールが定義されている場合は、このフォールバックロケールを使用します。
- ・ 派生ユニバースレベルで定義されているフォールバックロケールがなく、派生ユニバースで定義されているコアユニバースがある場合は、最初のコアユニバースのフォールバックロケールを使用します。
- ・ コアユニバースに定義されているフォールバックロケールがない場合は、ユニバースにフォールバックロケールはありません。

1.8.7 トランスレーションマネジメントツール

ユニバースデザインツールは、ユニバースメタデータの翻訳またはメタデータの翻訳の表示には使用できません。ユニバースデザインツールでは、元のユニバースコンテンツのみが表示されます。BusinessObjects Enterprise スイートでは、翻訳作業のためのトランスレーションマネジメントツールが提供されています。このツールは、スタンドアロンアプリケーションで、Windows プラットフォーム用のみで提供されています。

トランスレーションマネジメントツールを使用すると、ユニバースの設計者は、次の操作を実行できます。

- ・ ユニバースに新しいロケールを追加し、[使用可能] に設定します。
- ・ ユニバースのフォールバックロケールを定義します。

- ・ インタフェースから直接、ユニバースメタデータを追加されたロケールに翻訳します。
- ・ 異なるロケールでメタデータの翻訳ステータスを設定します。
- ・ 外部翻訳用にユニバースメタデータを XLIFF ファイルにエクスポートし、アプリケーションにインポートします。

ユニバースメタデータは翻訳後に再保存が可能なため、多言語のレポートに役立ちます。詳細については、『トランスレーションマネジメントツールユーザガイド』を参照してください。

1.8.8 多言語データ

PREFERRED_VIEWING_LOCALE および DOMINANT_PREFERRED_VIEWING_LOCALE 変数によって、デザイナーは、クエリ時に多言語データをフィルタ処理してユーザの優先表示ロケールのデータのみを取得するようにユニバースをカスタマイズできます。これは @Variable 関数を使用して実行されます。

1.9 ユニバースデザインツールのサンプル資料

以下のサンプルが、ユニバースデザインツールで提供されています。

1.9.1 デモデータベース

本ガイドのサンプルの多くは、Microsoft Access 2000 で作成された Club データベースを基盤にしています。このデータベースは、架空の企業 Island Resorts のセールスマネージャが、売上げとマーケティングの分析のために使用します。データベースファイルの Club.mdb は、Business Objects のインストールフォルダの Databases サブフォルダにあります。

このデータベースの構造に関する詳しい説明は、本ガイドの付録を参照してください。

eFashion データベースも、このリリースに含まれています。このMS Access 2000データベースには、13の店舗で販売される211種類の製品(色違いを入れると663製品)の過去3年間にわたる販売データが保存されています。

データベースには、以下のデータが保存されています。

- ・ 週間単位で販売情報を記録した89,000行のメインファクトテーブル
- ・ セールスデータを記録した2次ファクトテーブル
- ・ 集計ナビゲーションを設定した 2 つの集計テーブル

1.9.2 デモユニバース

beach.unv と呼ばれるデモユニバースは、Business Objects のインストールパスの Samples フォルダの Universes サブフォルダにあります。このユニバースは、上記の Club データベースを元に作成されたものです。

このユニバースを使って、ユニバースデザインツールを使用した特定のオブジェクトやクラスの作成方法を学習できます。

ユニバースデザインツールでは、eFashion データベースを元に作成された eFashion ユニバースも提供されています。

1.10 インフォメーションデザインツールでのユニバースの使用

インフォメーションデザインツールを使用して、ユニバースデザインツール、ユニバースデザインツールデスクトップエディション、Universe Designer または Universe Designer Personal で作成した .unv 形式のユニバースの作業を実行できます。これらのファイルは、インフォメーションデザインツールから直接使用できないため、ファイルのバージョンに応じて、最初にこれらのファイルを変換するか、アップグレードおよび変換する必要があります。.unv ユニバースファイルの各種バージョンを使用するために実行するステップ、およびファイルを変換した後にサポートされる機能の詳細については、インフォメーションデザインツールを参照してください。

注

インフォメーションデザインツールで使用するために .unv ファイルを一度変換すると、そのファイルを作成するために使用したツールではそのファイルを開くことができません。

基本操作の実行

2.1 概要

この章では、ユニバースデザインツールでユニバースの作成、変更、更新を実行するための基本的な操作について説明します。次の項目について説明します。

- ・ 35 ページの [「ユニバースデザインツールの起動」](#)
- ・ 45 ページの [「ユニバースのインポート」](#)
- ・ 46 ページの [「ユニバースを開く」](#)
- ・ 47 ページの [「ユニバースのエクスポート」](#)
- ・ 49 ページの [「ユニバースの保存」](#)
- ・ 77 ページの [「ユニバースの作成とユニバースパラメータの設定」](#)
- ・ 52 ページの [「ユニバースデザインツールのユーザインタフェースの使用」](#)
- ・ 58 ページの [「検索と置換の使い方」](#)
- ・ 61 ページの [「テーブル表示の整理」](#)
- ・ 66 ページの [「スキーマの表示オプションの選択」](#)
- ・ 74 ページの [「ユニバースの印刷」](#)

2.2 ユニバースデザインツールの起動

ユニバースデザインツールは、Central Management System(CMS)リポジトリでのみ使用できます。ユニバースデザインツールを起動する前に、リポジトリにログインする必要があります。

初めて ユニバースデザインツールを起動して既存のユニバースで作業する場合、直接ユニバースを開き、セキュリティ接続でユニバースを保存し、リポジトリにエクスポートする必要があります。次に、ユニバースをインポートして最新表示し、更新済みのバージョンをエクスポートします。これで、CMS とローカルユニバースのバージョンが同期化されます。

ユニバースデザインツールを起動して、次のいずれかの方法でユニバースを開きます。

- ・ 新しいユニバースを作成する
- ・ CMS リポジトリからユニバースをインポートする
- ・ ファイルシステムから直接ユニバースを開く

ユニバースは、リポジトリにエクスポートされると、Web Intelligence ユーザーだけが使用できるようになります。ユニバースをインポートして変更し、次に更新されたユニバースをエクスポートするのが、ユニバースデザインツールでの一般的な操作手順です。これにより、CMS(リポジトリ)でのバージョンがファイルのバージョンと同期化されます。

注

ユニバースはファイルシステムに保存できます。これを実行するのは、ターゲット CMS に接続する権限のないその他のユーザと、ユニバースを共有する場合です。詳しい説明は、49 ページの「[ユニバースの保存](#)」を参照してください。

このリリースでインストールされている Business Objects 製品グループのツールアイコンをクリックして、タスクバーからユニバースデザインツールを起動します。ユニバースデザインツールを起動する前に、CMS にログインするように要求されます。

2.2.1 ユニバースデザインツールを起動する

ユニバースデザインツールを起動するには、以下の操作を実行します。

- 1 タスクバーで、[スタート]ボタンをクリックします。
- 2 [プログラム]メニューをポイントします。
- 3 [BusinessObjects]コマンドから、[ユニバースデザインツール]プログラムをクリックします。

CMS へのログインボックスが開きます。

- 4 次の情報を入力します。通常、こうした情報は BusinessObjects 管理者によって提供されます。

ログイン情報	説明
システム	CMS サーバ名
ユーザ名	リポジトリのユーザ名です。
パスワード	リポジトリのパスワードです。
認証	ユーザのセキュリティレベルを表します。

- 5 [OK]をクリックします。

ユニバースデザインツールの開始画面が表示され、空のセッションが開きます。

ユーザ名と CMS 名がタイトルバーに表示されます。

ユニバースデザインツールに設定されているオプションに従って、ユニバースデザインツールを起動すると、[クイックスタート]ユニバース設計ウィザードが自動的に開きます。[キャンセル]をクリックするとウィザードが終了します。その他のウィザードオプションを無効にする方法は、44 ページの「[クイックデザインウィザードの無効化](#)」を参照してください。クイックデザインウィザードを利用する場合は、38 ページの「[クイックデザインウィザードの使い方](#)」を参照してください。

2.2.2 クイックデザインウィザードの使い方

セッションを初めて開始するときには、デフォルトでクイックデザイン ウィザードが開きます。このウィザードを使用すると、簡単にユニバースを作成し、ユニバースデザインツールの操作に慣れることができますが、エンドユーザのレポート作成要件を満たす完全なユニバースの作成には適していません。

クイックデザイン ウィザードはユニバースデザインツールの機能を習得するためだけに使用し、ユニバースの設計には使用しないことをお勧めします。このガイドでは、ウィザードの使用方法を説明した38 ページの「[クイックデザインウィザードの使い方](#)」の章を除き、クイックデザインウィザードは使わないものとして、ユニバースのデザイン、構築、メンテナンスの説明と操作方法を解説しています。その他のクイックデザイン ウィザードオプションを無効にする方法は、44 ページの「[クイックデザインウィザードの無効化](#)」を参照してください。

2.3 Designer XI R3 を使用した XI R2 コネクションとユニバースの操作

ユニバースデザインツールのこのリリースでは、XI R2 コネクションにアクセスし、XI R2 CMS に保存されているユニバースを開いたり、インポートしたりすることができます。XI R2 ユニバースおよび接続を使用する際は、次の点に注意してください。

- ・ Desktop Intelligence XI R2 ユーザは、XI 3.1 ユニバースと XI R2 接続に基づいて Desktop Intelligence XI 3.1 で作成したドキュメントを最新表示できます。
- ・ Desktop Intelligence XI R2 ユーザは、XI 3.1 ユニバースと XI R2 接続に基づいてドキュメントを作成できます。
- ・ XI R2 接続を編集して保存する場合、その接続は XI 3.1 接続として保存されること、また、その接続を使った XI R2 レポートは最新表示できないことを知らせる警告メッセージが表示されます。
- ・ XI 3.1 Universe Designer で XI R2 ユニバースを開くことはできますが、以前のバージョンの Designer で XI 3.1 ユニバースを開くことはできません。

管理者は、Desktop Intelligence XI R2 と XI 3.1 のこの相互接続機能を利用することで、Desktop Intelligence XI R2 クライアントおよび XI 3.1 クライアントをサーバに接続したまま、XI 3.1 サーバにアップグレードすることができます。これは大規模なクライアント群のアップグレードを管理する上で、一時的に必要な段階です。

2.4 クイックデザインウィザードを使用した基本的なユニバースの作成

ユニバースデザインツールには、単純なリレーショナルスキーマから、デモンストレーション用やテスト用のユニバースを作成できる、基本的かつ完全なユニバースを作成できるクイックデザインウィザードが用意されています。作成されたユニバースをそのまますぐに使用することも、オブジェクトを編集してさらに複雑なユニバースを作成することもできます。基本的なユニバースに変更を加えることで、ユニバースの質と構造を徐々に洗練されたものにすることが可能です。

ただし、本稼動用ユニバースはクイックデザインウィザードを使わず、手動で作成してください。ユニバースを手動で作成する方法は、本ガイドの他のすべての章で説明されています。ユニバースの自動作成については、この節でのみ説明されています。

2.4.1 クイックデザインウィザードの使用方法

クイックデザインウィザードは、ユニバースの作成をアシストします。データベースとの接続、基本的なクラスとオブジェクトを対話形式で作成できるほか、オブジェクト、結合、カーディナリティの自動生成のための戦略が用意されています。ウィザードには、オブジェクト、結合、テーブルを自動作成するための戦略が内蔵されています。

クイックデザインを使用すると、次のような利点があります。

- ・ ユニバースデザインツールに慣れていない場合、ユーザインタフェースと基本的なユニバースのデザインに慣れるためのガイドとして使用できます。
- ・ デモ用のユニバースを作成している場合、デザインプロセスを自動化して時間を節約できます。ウィザードでユニバースの作業モデルを素早く設定し、そのあとで、デモの参加者に合わせてユニバースをカスタマイズできます。

2.4.2 クイックデザインウィザードの使い方

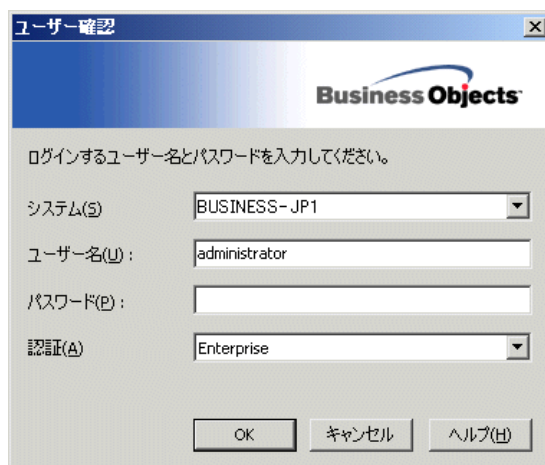
クイックデザインは、ユニバースを自動作成するときに使用するウィザードの名前です。ウィザードでの各ステップを、以下のセクションで説明していきます。

2.4.2.1 クイックデザインウィザードの起動

クイックデザインウィザードを開始する手順は、次の通りです。

- 1 ユニバースデザインツールを起動します。

[ユーザ確認]ダイアログボックスが表示されます。



- 2 [ユーザ確認]ダイアログボックスに、ユーザ名とパスワードを入力します。

- 3 [OK]をクリックします。

[クイックデザインウィザードへようこそ]画面が表示されます。

注

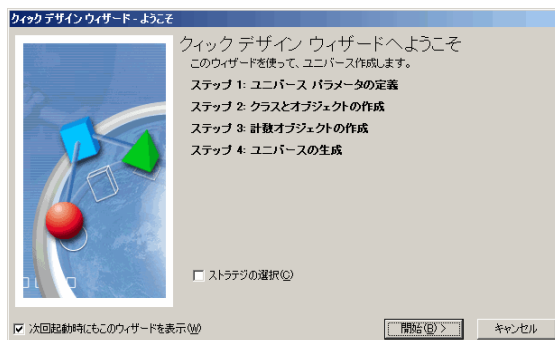
次のセッションの開始時にウィザードを起動する必要がある場合は、[次回起動時にもこのウィザードを表示]チェックボックスをオフにします。また、[オプション]ダイアログボックスの[一般設定]タブには、ウィザードを表示する2つのオプションがあります。[起動時にクイックデザインウィザードを表示]と[[ファイル]で[新規作成]を選択したらクイックデザインウィザードを起動]です([ツール]メニューの[オプション]コマンド)。

2.4.2.2 初期画面

最初の画面では、基本的なユニバースを作成するために必要な4つのステップが、簡単に説明されます。ダイアログボックスの下にある[ストラテジーの選択]チェックボックスをチェックしておくと、後のステップでストラテジーを選択できます。このチェックボックスをオンにしておくと、ユニバースの作成時にストラテジーを選択できます。オフにした場合は、ユニバースデザインツールのデフォルトのビルトインストラテジーが適用されます。

表示される各ダイアログボックスで、アクションを続行するために必要な情報を入力するよう指示するプロンプトが表示されます。

次のダイアログボックスに進むには[次へ]をクリックします。[戻る]をクリックすると前のステップに戻ります。[キャンセル]をクリックすると、いつでもクイックデザインウィザードを中止できます。



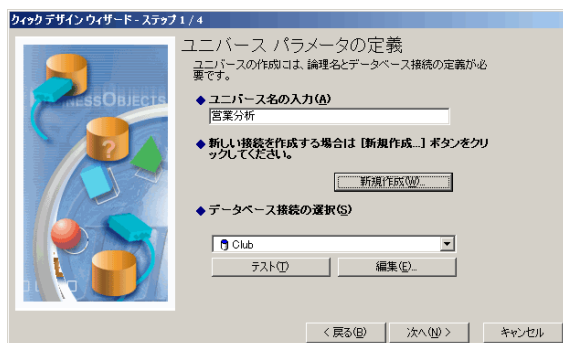
[ストラテジーの選択]チェックボックスを選択すると、ストラテジーを一覧表示したダイアログボックスが開きます。このダイアログボックスの詳細については、41 ページの「[ストラテジーの選択](#)」を参照してください。ストラテジーを選択、またはデフォルトのストラテジーを使用することができます。

[開始]をクリックして、作成プロセスを開始する。

2.4.2.3 ユニバースパラメータの定義

このステップではユニバースパラメータ(ユニバース名とデータベース接続)を定義します。

ユニバースには、半角英数字35文字まで(全角文字で17文字まで)の名前(ロングネーム)を付けられます。



接続は、作成することも、既存の接続を選択することもできます。新規に作成する場合は[新規作成]をクリックし、表示されるダイアログボックスに必要なパラメータを入力します。これらのダイアログボックスの詳細については、83 ページの「[接続の定義と編集](#)」を参照してください。

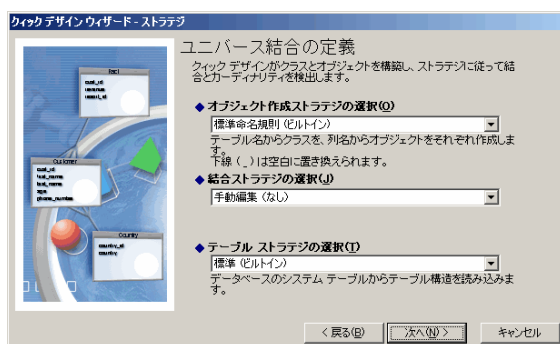
[テスト] ボタンをクリックして、接続が有効かどうかをチェックします。[編集]をクリックすると、選択した接続のパラメータを変更できます。

[次へ]をクリックして、次の手順に進みます。

2.4.2.4 ストラテジーの選択

ようこそ画面で [ストラテジーの選択] オプションを選択すると、オブジェクト、結合、テーブルを作成するストラテジーを指定するよう、クイックデザインでプロンプトが表示されます。

ストラテジーは、データベースやテキストファイルから自動的にデータ構造に関する情報を抽出するスクリプトです。ユニバースデザインツールでは、オブジェクト、結合およびテーブルの自動生成にこれらのスクリプトを使用します。



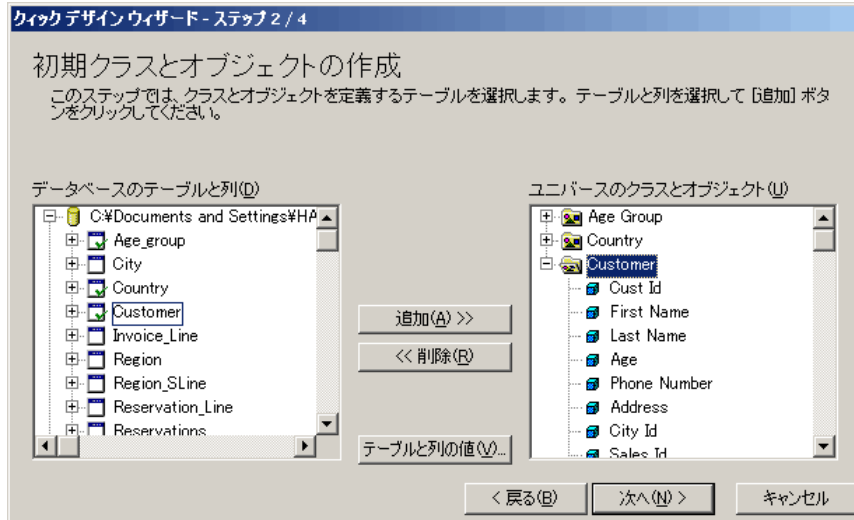
一覧ボックスで、別のストラテジーを選択することも、何も選択しないこともできます。現在のストラテジーについての簡単な説明がボックスの下に表示されます。

ユニバースデザインツールで提供されているビルトインの内部ストラテジーに加え、ユーザ定義の外部ストラテジーを作成することもできます。402 ページの「[外部ストラテジーを使用したユニバース作成のカスタマイズ](#)」を参照してください。

[次へ]をクリックして、次の手順に進みます。

2.4.2.5 初期クラスとオブジェクトの作成

データベース接続のパラメータを基に、ウィザードがデータベースのテーブルと列の一覧を表示します。左のボックスからテーブルと列を選択し、右の[ユニバースのクラスとオブジェクト]ボックスに追加して、初期クラスとオブジェクトを作成します。



デフォルトでは、左の枠にはテーブル名のみが表示されます。次の方法でファイルツリーを操作し、右の枠にクラスとオブジェクトを追加することができます。

- ・ テーブルの列を表示するには、テーブル名の左にある (+) 記号をクリックします。
- ・ テーブルまたは列のデータ値を表示するには、値をクリックして [テーブルと列の値] ボタンをクリックします。
- ・ 1 つのテーブルを選択するには、テーブルをクリックして [追加] ボタンをクリックします。
- ・ 複数の連続したテーブルを選択するには、[Shift] キーを押しながらブロックの最初と最後のテーブルをクリックします。選択したテーブルの間にあるテーブルがすべて反転表示されます。[追加] ボタンをクリックします。
- ・ 連続していない複数のテーブルを選択するには、Ctrl キーを押しながらテーブルを順番にクリックします。[追加] ボタンをクリックします。
- ・ テーブルを左の枠から右の枠にドラッグアンドドロップしても選択できます。

テーブルを挿入すると、ユニバースデザインツールに、テーブルのすべての列が含まれます。

右の枠で、クラス名がフォルダアイコンの横に表示されます。クラス名の横にある (+) 記号をクリックして、オブジェクトを表示します。クラスまたはオブジェクト名を変更するには、変更するクラスまたはオブジェクトをダブルクリックし、新しい名前をダイアログボックスに入力します。

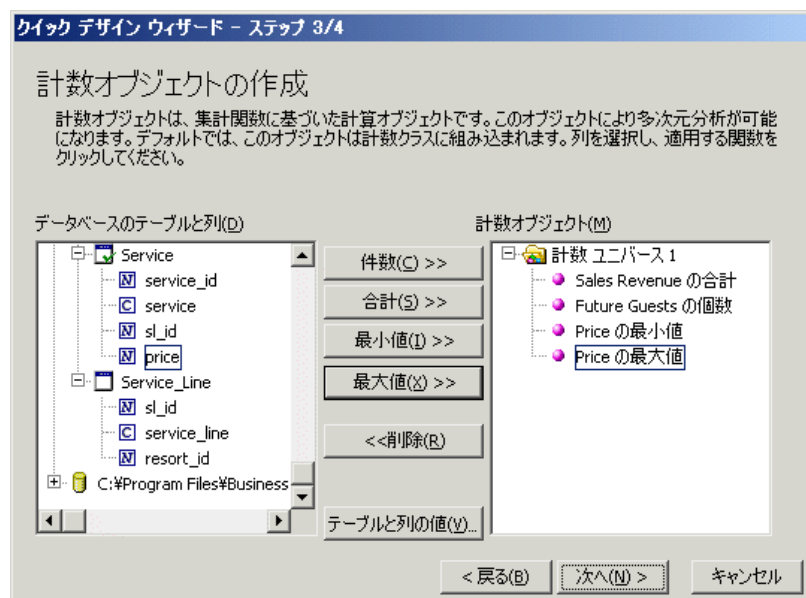
デフォルトでは、すべてのオブジェクトにはディメンションの資格が設定され、オブジェクト名の左側に立方体のアイコンが表示されます。

クラスまたはオブジェクトを削除するには、削除するクラスまたはオブジェクトをクリックして [削除] ボタンをクリックします。

[次へ] をクリックして次のステップに進みます。

2.4.2.6 メジャーオブジェクトの作成

メジャーオブジェクトは、集計関数(カウント、合計、最小値、最大値)から由来しています。この種類のオブジェクトは、数値情報を提供します。下の図の右のボックスには、いくつかのメジャーオブジェクトの例が表示されています。



オブジェクトに関連付けられたデータ値を表示するには、オブジェクトをクリックして[テーブルと列の値]ボタンをクリックします。

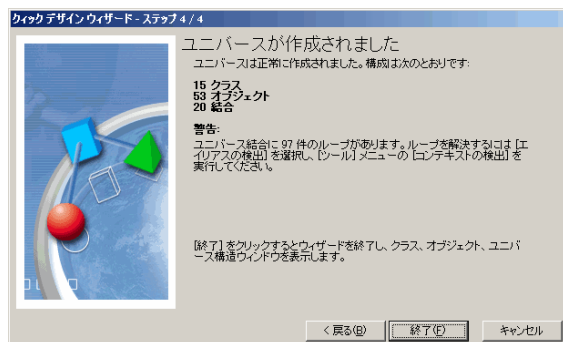
メジャーオブジェクトを作成するには、左の枠で適切なオブジェクトをクリックして[合計]ボタンをクリックします。作成したメジャーオブジェクトの名前を変更できます。

メジャーオブジェクトを1つまたは複数のメジャークラスにグループ化すると、ユニバースを整理できます。また、ユニバースのナビゲートも簡単になります。メジャーオブジェクトの詳細については、297ページの「[メジャーの定義](#)」を参照してください。

[次へ]をクリックすると、クイックデザインでユニバースの作成が開始されます。

2.4.2.7 ユニバースの生成

クイックデザインの各手順で指定したパラメータに基づいて、新しいユニバースが自動作成されます。作成されたユニバースに含まれるクラス、オブジェクト、結合の数が表示されます。



上記のダイアログボックスに、ユニバースの結合内にループが存在することを示すメッセージが表示されます。ユニバースデザインツールでは、エイリアスとコンテキストを使用してループを解決することができます。ループの解決に関しては、「スキーマのデザイン」の章を参照してください。

[終了] ボタンをクリックすると、新しく作成されたユニバースのユニバース枠と構造枠が表示されます。

2.4.2.8 ワークセッションの終了

[ファイル]>[名前を付けて保存]の順に選択してユニバースを保存し、[ファイル]>[閉じる]の順に選択してユニバースを閉じます。

ユニバースを保存すると、ユニバースデザインツールからファイル名を入力するよう要求されます。ユニバースファイル名に含められる最大文字数は、ご使用のオペレーティングシステムによって異なります。ファイルには .unv 拡張子が付きます。ユニバースデザインツールでは、これらのファイルは、デフォルトで、BusinessObjects フォルダの Universe サブフォルダに保存されます。Windows 2000 では、このフォルダはユーザプロファイルの Local Data フォルダの下にあります。

ユニバースデザインツールを終了するには、[ファイル]>[終了]を選択します。

2.4.2.9 クイックデザインウィザードの無効化

初めてセッションを開始すると、デフォルトでクイックデザインウィザードが開きます。新しいユニバースを作成する際に、クイックデザインウィザードが起動しないよう設定するには、次のようにオプションを変更します。

クイックデザインウィザードを無にする方法:

- 1 [[ツール]]>[[オプション]]の順に選択します。

[オプション]ダイアログボックスの[一般設定]ページが表示されます。

- 2 [起動時にクイックデザインウィザードを表示]チェックボックスをオフにします。(ウィザードの初期画面で[次回起動時にもこのウィザードを表示]チェックボックスをオフにした場合は、このオプションはすでにオフになっています。)
- 3 [[ファイル]で[新規作成]を選択したらクイックデザインウィザードを起動]チェックボックスをオフにします。
- 4 [OK]をクリックします。

[オプション]ダイアログボックスの[一般設定]タブにある上記オプションをチェックするだけで、いつでもクイックデザインウィザードを有効にできます。クイックデザインウィザードの使い方は、38 ページの[「クイックデザインウィザードの使い方」](#)で説明されています。

2.4.3 クイックデザインウィザードで作成したユニバースの処理

クイックデザインウィザードで基本的なユニバースを作成した後、結合を編集したり、エイリアスやコンテキストを使ってループを解消する必要がある場合があります。また、ユニバースデザインツールの豊富な機能を利用して、より複雑なコンポーネントを含むユニバースに拡張することもできます。ユニバースの編集や改良に関する操作や機能の説明は、本ガイドにある関連セクションを参照してください。

2.5 ユニバースのインポート

リポジトリのユニバースフォルダに保存されたユニバースをインポートすることができます(複数可)。インポートできるユニバースは、リポジトリにすでにエクスポートされているユニバースだけです。

ユニバースをインポートすると、リポジトリファイルシステムのユニバースのバージョンが CMS によってチェックされます。バージョンが同一であれば、ユニバースはユニバースデザインツールでも利用可能です。リポジトリファイルシステムのユニバースのバージョンが CMS のバージョンより新しい場合は、フォルダのユニバースを置換するかどうかを質問するメッセージボックスが表示されます。[はい]を選択すると、リポジトリファイルシステムのユニバースは、CMS のバージョンで上書きされます。

2.5.1 ユニバースのリポジトリからのインポート

2.5.1.1 ユニバースをリポジトリからインポートする

- 1 [ファイル] > [インポート]を選択します。
[ユニバースのインポート]ダイアログボックスが開きます。
- 2 ドロップダウンリストボックスからユニバースフォルダを選択します。
または
[参照]ボタンをクリックして、フォルダブラウザでユニバースを選択します。
ユニバースは、ここで指定したフォルダからインポートされます。
- 3 [ユニバース]リストボックスでユニバース名をダブルクリックすると、ユニバースがロックされ、
ロックされたユニバースの横には、ロックキー記号が表示されます。ロックを解除するには再度ダブルクリックします。
- 4 ユニバース名をクリックします。
クリックしたユニバースがインポートされます。
- 5 [インポートフォルダ]ボックスに表示されるファイルのパスを確認します。
ユニバースは、このフォルダにインポートされます。
- 6 [OK]をクリックします。

2.5.2 ユニバースを開くこととインポートすることとの違い

ユニバースは、ファイルシステムから直接開くことができます。そのユニバースを保存しても、ファイルシステムに保存されるだけです。CMS 内のバージョンは更新されません。ユニバースを更新しても、Web Intelligence ユーザーには利用できません。

ユニバースをインポートすると、リポジトリで利用できる現在のバージョンが、ユニバースデザインツールでも利用できるようになります。ユニバースの変更が終わったら、リポジトリにエクスポートします。CMS には最新の変更が反映されます。

2.6 ユニバースを開く

メニューコマンドまたは[開く]をクリックして、ユニバースを開きます。インポートせずにユニバースを直接開くと、CMS の最新バージョンとは一致していない可能性のある、ローカルファイルシステムのバージョンを開いていることになります。

2.6.1 ユニバースを直接開く

- 1 [ファイル] > [開く]を選択します。

[開く]ボックスに、ユニバースファイルのデフォルトの保存ディレクトリが表示されます。デフォルトのディレクトリは、[ツール] > [オプション] > [保存]で表示される[オプション]ダイアログボックスの[保存]ページで設定できます。

- 2 必要に応じて、ユニバースファイル (.UNV) を含んだディレクトリを参照します。
- 3 ユニバースファイルを選択し、[開く]をクリックします。

または

ユニバースファイルをダブルクリックします。

ユニバースが、現在のユニバースデザインツール ウィンドウに表示されます。

2.7 ユニバースのエクスポート

ユニバースを Web Intelligence ユーザーや他のデザイナーが利用できるようにするには、ユニバースをリポジトリにエクスポートします。

ユニバースをエクスポートする場合

- ・ リポジトリファイルシステム上で選択したユニバースフォルダに移動している。

および

- ・ セントラル管理システム(CMS)で作成されている。

ユニバースがリポジトリにエクスポートされるたびに、CMS のユニバースのバージョンは更新されます。Web Intelligence ユーザーが利用できるのは、このバージョンです。

注

ユニバースの保存とユニバースのエクスポートは、同じではありません。ユニバースを保存すると、ローカルのファイルシステムは更新されますが、CMS リポジトリのユニバースのバージョンは更新されません。詳しい説明は、49 ページの [「エクスポートと保存の違い」](#)を参照してください。

2.7.1 リポジトリファイルシステムでのユニバースの構成

リポジトリは、ローカルのフォルダシステムと CMS サーバにユニバースを保存します。ユーザは、ローカルファイルシステムのユニバースバージョンを操作します。このローカルファイルシステムは、ユニバースデザインツールがインストールされたサーバ上にあります。デフォルトで、ユニバースはユーザプロファイルで指定されたユニバースフォルダに保存されます。パスは次のとおりです。

¥¥Documents and Settings¥<ユーザ名>¥Application Data¥Business Objects¥Business Objects 12.0¥universes¥@<リポジトリ名>¥universe folder¥<ユニバース名>.unv

CMS サーバに保存されたユニバースは、バージョンの制御に使用されます。更新したユニバースをリポジトリにエクスポートすると、そのユニバースが CMS サーバに複写されます。

2.7.2 ユニバースのリポジトリへのエクスポート

2.7.2.1 ユニバースをリポジトリにエクスポートする

- 1 [ファイル] > [エクスポート]を選択します。
[ユニバースのエクスポート]ダイアログボックスが開きます。
- 2 フォルダのドロップダウンリストボックスからユニバースフォルダを選択します。
または
[参照]ボタンをクリックして、フォルダブラウザでユニバースフォルダを選択します。
ユニバースは、このフォルダにエクスポートされます。
- 3 [ユニバース] リストボックスでユニバース名をダブルクリックすると、ユニバースがロックされ、
ロックされたユニバースの横には、ロックキー記号が表示されます。ロックを解除するには再度ダブルクリックします。
- 4 [グループ]リストボックスで、グループをクリックします。このユーザグループがエクスポートするユニバースを使用します。
- 5 [ユニバース]リストボックスでユニバースをクリックします。[ユニバース]リストボックスには有効なユニバース名が表示されます。
- 6 開いていないユニバースをエクスポートする場合、[ユニバースの追加]ボタンをクリックし、希望するユニバースを参照します。
- 7 [OK]をクリックします。

2.7.3 エクスポートと保存の違い

ユニバースを保存すると、リポジトリファイルシステムのバージョンが更新されます。CMS のバージョンは更新されません。

ユニバースをエクスポートすると、リポジトリファイルシステムのバージョンが更新され、CMS 上のユニバースと同期化されます。

ユニバースを保存し、更新したバージョンをエクスポートしない場合、CMS のバージョンは更新されません。つまり、他のユーザは更新されたユニバースを使用できません。

リポジトリ内にある各ユニバースには、システムによって識別子が付加されます。詳細については、541 ページの「[リポジトリでのユニバースの識別](#)」を参照してください。

他のデザイナーがリポジトリでロックしているユニバースは、エクスポートできません。

ユニバースをエクスポートするには、ユニバースにセキュリティ接続が設定されている必要があります。

2.8 ユニバースの保存

作業セッション中は、定期的にユニバースを保存してください。ユニバースを保存すると、ユニバースデザインツールによって、拡張子 .unv の付いたファイルとして、ローカルファイルシステムに保存されます。

Web Intelligence では、ユーザーはユニバース名 (ロング ネーム) でユニバースを識別します。

ユニバースを保存しても、変更は CMS にあるバージョンには反映されません。ユニバースの更新の完了時に、そのユニバースを CMS にエクスポートする必要があります。

ユニバース名 (ロングネーム) と .unv に使用できる最大文字数は、次のとおりです。

名前の種類	使用できる最大文字数
ユニバース名	100
.unvファイル名	オペレーティングシステムの最大値

2.8.1 識別子としてのユニバースファイル名

ユニバースを使用したレポートの作成後に、そのユニバースのファイル名 (.unv) を変更することはできません。ファイル名が変更されると、古いファイル名のユニバースで作成されたレポートでユニバースを識別できなくなるからです。

2.8.2 ユニバースの保存

ユニバース名とファイル名 (.unv) は、異なっても構いません。

[名前を付けて保存]を使って新しい名前でもユニバースを保存すると、新しいユニバースは CMS のバージョンに関連付けられません。新しいユニバースのバージョンを作成するには、新規ユニバースを CMS にエクスポートする必要があります。

ユニバースを保存するには、次のいずれかの操作を行います。

- ・ メニューバーで、[ファイル] > [保存]の順に選択する。
- ・ 標準ツールバーの[保存]アイコンをクリックする。
- ・ キーボードから CTRL+S を入力する。

2.8.3 ユニバースの定義の PDF としての保存

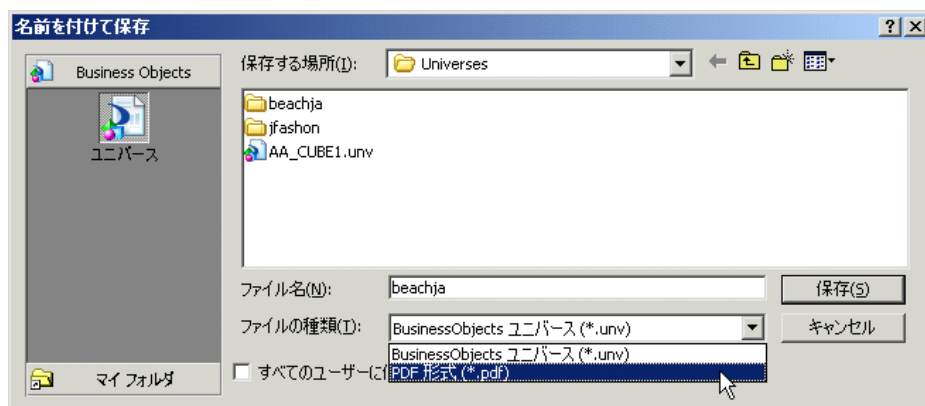
ユニバースの定義を Adobe PDF ファイルとして保存することができます。印刷時に出力できるユニバース情報と同じものが保存されます。これには、次の情報が含まれています。

- ・ 一般情報: パラメータ、リンクされたユニバース、テーブルスキーマ図
- ・ コンポーネントの一覧: オブジェクト、条件、階層、テーブル、結合、コンテキストなど、ユニバースに含まれるコンポーネントの一覧
- ・ コンポーネントの詳細: ユニバース内のオブジェクト、条件、階層、テーブル、結合、コンテキストの説明

PDFに表示するコンポーネントは、[オプション]ダイアログボックスの印刷オプション ([ツール] > [オプション] > [印刷/PDF]タブ) で選択できます。このオプションについては、74 ページの「[印刷オプションの設定](#)」を参照してください。

ユニバース情報を PDF として保存する手順は、次のとおりです。

- 1 [ファイル] > [別名で保存]を選択します。
- 2 [ファイルの種類]ドロップダウンリストボックスから[PDF 形式 (*.pdf)]を選択します。



- 3 [保存]をクリックします。

2.8.3.1 デフォルトの保存オプションの設定

デフォルトでは、ユニバースデザインツールはユニバースを Business Objects パスの Universe サブフォルダに保存します。別のデフォルト保存フォルダを次の手順で指定することもできます。

- 1 [[ツール]] > [[オプション]]の順に選択します。
[オプション]ダイアログボックスが表示されます。
- 2 [保存]タブをクリックします。
[保存]ページが表示されます。
- 3 [デフォルトのユニバースフォルダ]テキストボックスにファイルパスを入力します。
または
- 4 [参照]ボタンをクリックして、.unv ファイルが保存されているフォルダを指定します。
- 5 ファイルの自動保存間隔を指定する場合、[自動保存間隔]チェックボックスをオンにして、間隔をオプションの横にあるボックスで指定します。
- 6 [OK]をクリックします。

2.9 ユニバースの終了

次の方法でユニバースを終了できます。

ユニバースを閉じるには、次の操作を行います。

- ・ メニューバーで、[ファイル] > [閉じる]の順に選択します。
- ・ ユニバースウィンドウの右上にある[X]ボタンをクリックする。
- ・ キーボードで、CTRL+W を押す。

2.10 ワークグループ設計

ユニバースデザインツールは、複数のデザイナーがバージョン間の競合なしに同じユニバースで作業できるマルチユーザー環境を提供します。

ユニバースをロックして、一度に1人のデザイナーしかユニバースの変更を行えないようにすることができ、さらに変更履歴を維持するため、ユニバースにはバージョン番号が付けられます。

2.10.1 ユニバースのロック

universe フォルダに保存されると、ユニバースは必要なユーザ権限が与えられた複数のデザイナーによって共有されます。

ただし、同じユニバースは一度に1人のデザイナーしか編集できません。つまり、他のデザイナーがロックしていないユニバースだけを編集することができます。

注

ユニバースのロックは、[ユニバースのインポート]または[ユニバースのエクスポート]ダイアログボックスで設定します。ダイアログボックスでは、ロックしたユニバース名の横にはロックマーク(錠前)が表示され、他のデザイナーがロックしたユニバースでは、このマークがグレイ表示されます。

2.10.2 版数

ユニバースデザインツールでは、ユニバースを universe フォルダにエクスポートするたびに、ユニバースの版数が1つ加算されます。これにより、どれがユニバースの最新版かが判断できます。版数は、[ユニバースのパラメータ]の[要約]タブ([ファイル]>[ユニバースのパラメータ]>[要約])に、次のように表示されます。

2.11 ユニバースデザインツールのユーザインタフェースの使用

ユニバースデザインツール インタフェースは Microsoft Windows の標準に準拠しています。ウィンドウ、メニュー、ツールバー、ショートカットキー、オンラインヘルプを使用できます。

2.11.1 ユーザインタフェースの主要コンポーネント

各ユニバースは、ユニバースデザインツールのメインウィンドウ内の 1 つのユニバースウィンドウとして表示されます。

接続したデータベースで使用できるすべてのテーブルを表示する[テーブル参照]ウィンドウも使用できます。

2.11.1.1 ユニバースウィンドウ

[ユニバース]ウィンドウは、次の 2 つの枠に分かれています。

枠	表示
構造	ユニバースの元になっているターゲットデータベースをグラフィカルに表示しています。ユーザがクエリの実行に使用するオブジェクトをマップするテーブルと結合が含まれています。
ユニバース	ユニバースで定義されるクラスとオブジェクトです。ここには、Web Intelligence ユーザーがクエリーを作成する際に表示されるユニバースのコンポーネントが表示されます。

2.11.1.2 テーブル参照

[テーブル参照]には、接続したデータベースで使用できるテーブルが表示されます。[テーブル参照]で選択したテーブルは、[構造]枠に直接ドラッグするか、または[テーブル参照]でテーブルをダブルクリックして[構造]枠に挿入できます。

[テーブル参照]ウィンドウを開くには、次のいずれかの操作を行います。

- ・ [構造]枠の背景部分をダブルクリックする。

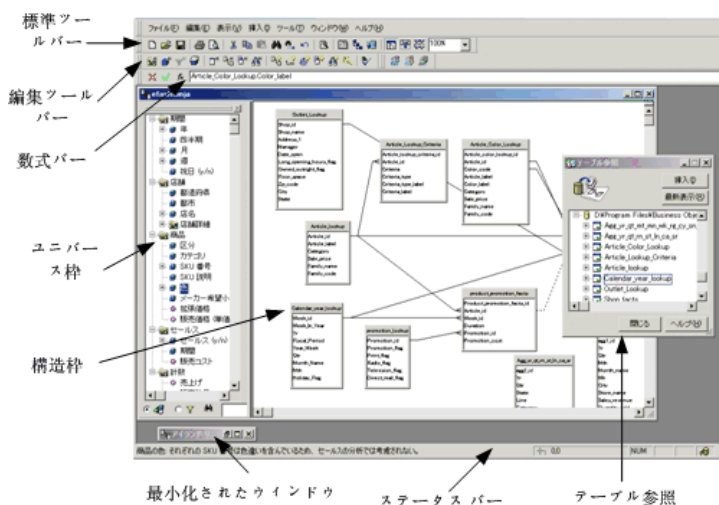
- ・ [構造]枠の背景部分を右クリックし、ポップアップメニューから[テーブルの挿入]を選択する。
- ・ [挿入] > [テーブル]を選択します。

注

[テーブル参照] ウィンドウの使い方は、「スキーマのデザイン」の章で詳しく説明します。

2.11.2 ユニバースデザインツールのユーザインタフェース

インタフェースの主なコンポーネントは次の通りです。



2.11.3 ウィンドウの操作

ユーザインタフェースの各ウィンドウは、次のように操作します。

- ・ ワークセッションでは、複数のユニバースで同時に作業できます。ユニバースデザインツールでは、各ユニバースは 1 つの[構造]枠と 1 つの[ユニバース]枠に表示されます。
- ・ 最近開いたユニバースは、[ファイル]メニューの下の方に表示されています。[ツール] > [オプション] > [一般設定]の順に選択して表示される[最近使ったファイル]オプションで、[ファイル]メニューに表示されるユニバースの数を設定できます。
- ・ ユニバースデザインツールウィンドウ内では、すべての作業ウィンドウを移動、サイズ変更、最小化することができます。

- ・ これらのウィンドウは、[ウィンドウ] > [整列]を選択して表示されるオプション、[重ねて表示][上下に並べて表示][左右に並べて表示]を使用して、最も作業しやすい位置に配置することができます。
- ・ [ウィンドウ] > [アイコンの整列]を選択して、ツールウィンドウで最小化されたウィンドウをすべて整列させることができます。

2.11.4 ツールバーの使い方

ユニバースデザインツール ウィンドウには、標準ツールバーと編集ツールバーの 2 種類のツールバーがあります。

両方のツールバーで選択できるボタンは、[ユニバース]枠または[構造]枠のどちらがアクティブになっているかによって異なります。使用できないボタンは、グレイ表示されます。

ツールバーは、表示と非表示を切り替えられます。ツールバーをドラッグして、ユニバースウィンドウの好きな位置に配置することができます。

2.11.4.1 ツールバーの移動

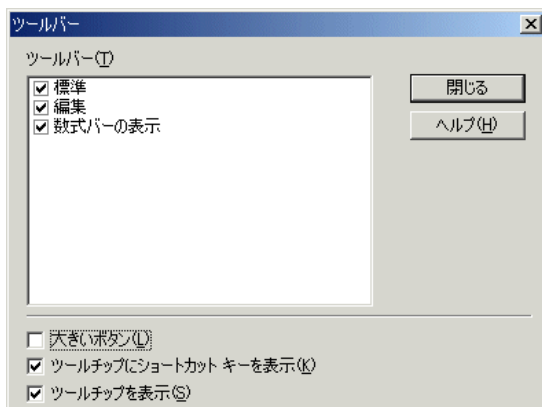
ツールバーを移動する手順は、次の通りです。

- 1 ツールバー内のボタン以外の場所をクリックします。
上記の図のように、両方のツールバー領域が表示されます。
- 2 左マウスボタンを押しながら、ツールバーを希望の位置までドラッグします。
- 3 マウスボタンを放します。
ツールバーが新しい位置に表示されます。

2.11.4.2 ツールバーの表示/非表示の切り替え

ツールバーの表示/非表示を切り替える手順は、次の通りです。

- 1 [表示] > [ツールバー]を選択します。
[ツールバー]ダイアログボックスが表示されます。



- 2 対応するツールバーのチェックボックスを選択または選択解除します。
- 3 ダイアログボックス下方にあるボタンのサイズ、ショートカットキーの表示、ツールヒントの表示に関するオプションを、希望にあわせて選択または選択解除します。
- 4 [OK]をクリックします。

2.11.5 ユニバースデザインツールでのアクションや処理の実行

ユニバースデザインツールでは、次の方法でアクションまたは処理を実行します。

- ・ メニューでコマンドを選択する。
- ・ Alt キーを押しながら、ショートカットキーを入力する。
- ・ ツールバーのボタンをクリックする。

2.11.5.1 マウスの使用

ユニバースデザインツールでは、マウスボタンに次の機能が割り当てられています。

シングルクリック

次のアクションを実行します。

- ・ 標準操作(コマンドの選択やボタンのクリック)の実行
- ・ [ユニバース]枠、[構造]枠または[テーブル参照]で要素を選択する。

- ユニバースデザインツール ウィンドウで 1 つまたは複数のコンポーネントを選択し、右ボタンを 1 回クリックしてポップアップメニューを表示する。メニューには、選択したコンポーネントに関するコマンドが含まれています。

ダブルクリック

次のユニバース構造をダブルクリックして、表示方法を変更したり、プロパティを編集したりできます。

ダブルクリック	結果
構造枠内の空のスペース	[テーブル参照]ダイアログボックスが表示されます。
構造枠のテーブル	テーブルの表示形式を変更します。テーブルとテーブル内の列の表示には、3種類の形式があります。詳細については、64 ページの「 テーブルの表示変更 」を参照してください。
[構造]枠の結合	[結合の編集]ダイアログボックスが表示されます。このダイアログボックスでは、結合のプロパティを変更できます。
ユニバース枠のクラス	[(クラス名の)プロパティの編集]ダイアログボックスが表示されます。このダイアログボックスでは、クラスのプロパティを変更できます。
ユニバース枠のオブジェクト	[(オブジェクト名の)プロパティの編集]ダイアログボックスが表示されます。このダイアログボックスでは、オブジェクトのプロパティを変更できます。
ユニバース枠の条件ビューに表示される条件オブジェクト	[(条件オブジェクトの)プロパティ編集]ダイアログボックスが表示されます。このダイアログボックスでは、オブジェクトのプロパティを変更できます。

2.11.5.2 アクションを元に戻す

実行したアクションは、次の 2 つの方法で元に戻します。

- [編集] > [元に戻す]を選択します。
- [元に戻す]をクリックします。

2.12 検索と置換の使い方

[検索]機能を使用して、ユニバースおよび構造枠に含まれる文字または文字列を検索することができます。[検索/置換]を使用すると、ユニバース構造内にある名前または説明に含まれる文字または文字列を検索して置換することができます。

2.12.1 検索機能

ユニバースおよび構造枠のユニバース構造に含まれるテキストを検索することができます。

2.12.1.1 検索オプションの設定

使用できる[検索]オプションは、[ユニバース]枠と[構造]枠のどちらから検索を実行したかで変わります。

次の検索オプションを設定して文字列を検索することができます。

オプション	条件	説明
検索対象	[ユニバース]枠または[構造]枠がアクティブ	この文字列を検索します。
大文字と小文字を区別する	[ユニバース]枠または[構造]枠がアクティブ	大文字と小文字を区別して文字を検索します。
単語単位で探す	[ユニバース]枠または[構造]枠がアクティブ	単語全体が一致するかを調べます。
以下も検索: 名前	[ユニバース]枠がアクティブ	このオプションを選択すると、クラス名、オブジェクト名、定義済みの条件名だけを探します。 選択しない場合は、クラス名、オブジェクト名、定義済みの条件名を検索範囲に含めません。

オプション	条件	説明
以下も検索: 説明	[ユニバース]枠がアクティブ	このオプションを選択すると、ユニバース構造の説明も検索します。
以下も検索: SQL	[ユニバース]枠がアクティブ	このオプションを選択すると、オブジェクト、結合、その他のユニバース構造のSQL定義も検索します。

2.12.1.2 ユニバース内での検索

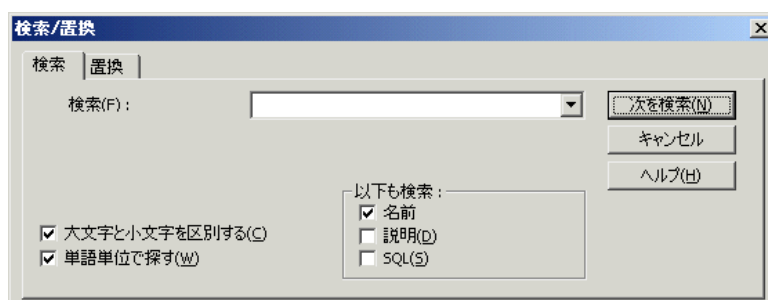
ユニバース内での検索手順は、次の通りです。

- 1 [ユニバース]枠または[構造]枠をクリックします。

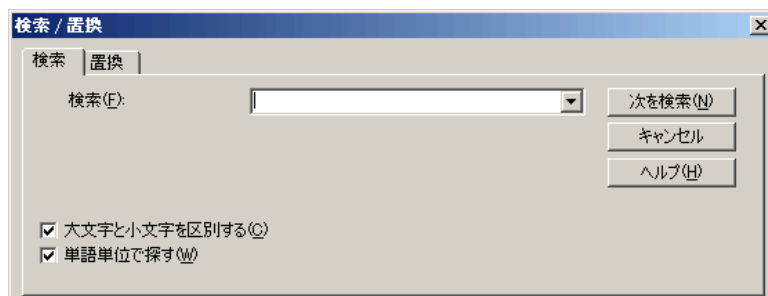
クリックした枠内の情報が検索されます。

- 2 [編集] > [検索]を選択します。

[検索/置換]ダイアログボックスが表示されます。開いている[ユニバース]枠のボックスは次のとおりです。



次の図は、[構造]枠がアクティブな場合です。



- 3 [検索]テキストボックスに、検索する文字または文字列を入力します。
- 4 適切な検索オプションを選択または選択解除します。
- 5 [次を検索]をクリックします。

検索文字(列) がユニバース枠で見つかった場合、オブジェクトが反転表示されます。検索文字(列) がオブジェクトの説明または SQL の定義で見つかった場合、文字(列) が反転表示された [オブジェクトのプロパティを編集] ボックスが自動的に表示されます。

- 6 [次を検索]をクリックすると、次の一致を検索します。
- 7 [検索/置換]ダイアログボックスを閉じるには、[キャンセル]をクリックします。

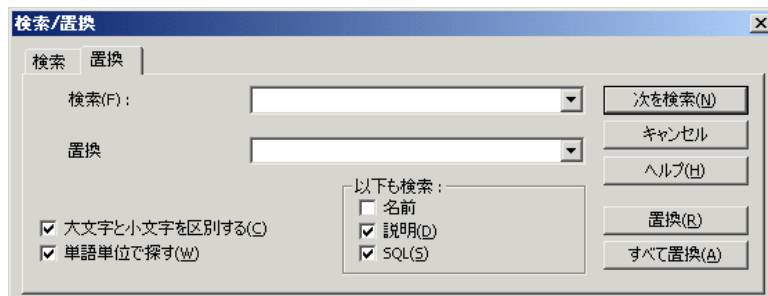
2.12.1.3 ユニバース内での検索置換

ユニバース内の文字または文字列を検索置換する手順は、次の通りです。

- 1 [編集] > [次を置換]の順に選択します。

[検索/置換]ダイアログボックスが表示されます。

- 2 [検索]テキストボックスに、検索する文字または文字列を入力します。



- 3 [置換]ボックスに置換する文字または文字列を入力します。検索文字(列)が見つかった場合、[検索対象]ボックスに入力する文字(列)で置き換えられます。
- 4 適切な検索オプションを選択または選択解除します。
- 5 検索結果が表示されるたびに、テキストを置換する場合は[置換]をクリックします。

または

[すべて置換]をクリックすると、ユニバース内で検索されたすべての文字または文字列が自動的に置換されます。

文字(列)を一つずつ順番に置換する場合、[オブジェクトのプロパティを編集]ボックスが自動的に開き、オブジェクトの説明に対象文字列が含まれる場合にアクティブになります。検索が終了するまで、[検索/置換]ボックスをクリックし続けて検索を実行します。

2.12.2 クイック検索の使い方

ユニバース枠の下にある検索ボックスに検索する文字列の最初の 1 字を入力すると、ユニバース枠内を検索できます。

ユニバース枠がアクティブな場合は、クラス名とオブジェクト名が検索されます。

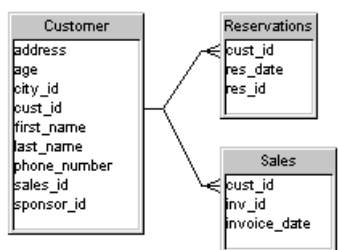
構造枠がアクティブな場合は、テーブル名が検索されます。

2.13 テーブル表示の整理

構造枠には、テーブルと列を表示、整理するための便利な機能があります。ここでは、その機能について説明します。スキーマをデザインする手法と、[構造]枠での効率的なスキーマの作成方法については、141 ページの「[テーブルと結合を使ったスキーマの作成](#)」の章を参照してください。

2.13.1 テーブルの表示方法

[構造]枠では、それぞれのテーブルは長方形のアイコンで表現されます。テーブル名は長方形の上部の枠に表示されます。長方形のアイコン内には、テーブルの列が一覧表示されます。テーブル間を接続する線は結合を示します。



2.13.2 テーブルの操作

[構造]枠のテーブルに対する操作を説明します。

2.13.2.1 テーブルの選択

テーブルを選択するには、次の操作を行います。

選択方法	操作
1つのテーブル	テーブルをクリックします。
複数のテーブル	<ul style="list-style-type: none">・ 左マウスボタンを押しながら選択するテーブルを四角形で囲みます。・ Shift キーを押しながら複数のテーブルを選択します。
すべてのテーブルを一度に選択	[編集] > [すべて選択]の順に選択します。

選択を解除するには、マウスポインタをテーブルの外に置き、もう一度クリックします。

2.13.2.2 テーブルの削除

テーブルを削除する手順は、次の通りです。

- 1 テーブルを選択します。
- 2 次のいずれかを行います。
 - ・ 標準ツールバーの[切り取り]をクリックします。
 - ・ [編集] > [切り取り]の順に選択します。
 - ・ Deleteキーを押します。

2.13.3 一覧モードの使用

一覧モードでは、アクティブユニバースで使用されているテーブル、結合、コンテキストをすべて表示することができます。ユニバースデザインツールの一覧モードでは、[構造]枠の上に 3 つの枠が追加されます。これらの枠は、[テーブル]枠、[結合]枠、[コンテキスト]枠として示されます。

次のように一覧モードを使用することができます。

アクション	結果
一覧モードの任意の枠でコンポーネントをクリックする。	コンポーネントが[構造]枠で反転表示されます。
テーブル、結合、またはコンテキストを構造枠で選択する。	対応するコンポーネントが[一覧]枠で反転して表示されます。
テーブル枠で、テーブル名をダブルクリックする。	[テーブルの名前を変更]ダイアログボックスが開きます。テーブルの名前だけでなく、データベースによっては、その所有者と資格を変更できます。
[結合]枠で、結合名をダブルクリックする。	[結合の編集]ダイアログボックスが表示されます。結合のプロパティを編集することができます。
[コンテキスト]枠で、コンテキスト名をダブルクリックする。	[コンテキストの編集]ダイアログボックスが表示されます。Ctrl キーを押しながら一覧にある結合をクリックして、結合を選択したコンテキストに追加できます。
コンポーネントをクリックし、2 つの一覧枠の間にある三角形記号をクリックする。	選択したコンポーネントに関連するコンポーネントだけを表示します。関係のないコンポーネントは表示されなくなります。
[一覧]枠と[構造]枠の間にあるセパレータをクリックし、上下にドラッグする。	ドラッグした方向により、[一覧]枠が拡大または縮小されます。

2.13.3.1 枠間にある三角形記号を使って表示されるコンポーネントにフィルタを適用する

各枠間にある三角形記号は、コンポーネント表示についてフィルタの役割を果たします。以下はその例です。

- ・ [テーブル]枠でテーブル名をクリックし、[結合]枠方向の三角形記号をクリックします。結合枠には、選択したテーブルの結合のみが表示されます。

- ・ [結合]枠で結合名をクリックし、テーブル枠方向の三角形記号をクリックします。[テーブル]枠には、選択した結合でリンクされているテーブルのみが表示されます。

2.13.3.2 一覧モードと一般表示モードを切り替える

一覧ビューと標準ビューは、次の 2 つの方法で切り替えることができます。

- ・ [ビュー] > [一覧モード]の順に選択する。
- ・ 一覧モードで、[一覧モード]ボタンをクリックする。

2.13.4 テーブルの自動整列

構造枠のテーブルは、次の 2 つの方法で自動配置することができます。

- ・ [表示] > [テーブルの整列]の順に選択する。
- ・ [配列]ボタンをクリックする。

2.13.5 テーブルの表示変更

テーブルを 3 通りの形式で表示することができます。各表示形式はフィルタの役目をし、テーブルに表示される情報量を調整します。

各表示形式は、次のとおりです。

テーブルの表示形式	説明
デフォルト	各テーブルの列を8列まで表示します。表示される列数を変更できます。詳細については、66 ページの 「スキーマの表示オプションの選択」 を参照してください。
名前のみ	テーブルに、テーブル名のみを表示します。この形式では、表示するテーブル数が多いときに[構造]枠内を有効に利用できます。

テーブルの表示形式	説明
結合列	テーブル間の結合に関する列のみが各テーブルに表示されます。通常、キー列が表示されます。

各表示形式は、次のとおりです。

2.13.5.1 デフォルトのテーブル表示

最初の8列を表示するテーブルシンボルを使って表示します。

Customer	
address	C
age	N
city_id	N
cust_id	N
first_name	C
last_name	C
phone_number	C
sales_id	N
...	

テーブルにデフォルト表示列よりも多くの列がある場合、最後の列のあとに省略記号(...)が表示されます。テーブルを1回クリックすると、スクロールバーがテーブル右側に表示されます。テーブルは、下にドラッグしてテーブルを拡大することができます。

2.13.5.2 テーブル名のみ表示

テーブル名だけを表示するには、次の操作を行います。

- ・ テーブルをダブルクリックします。

テーブルの名前だけが表示されています。

2.13.5.3 結合列テーブルの表示

結合列のみを次のようにして表示できます。

- ・ [構造]で、すでに名前のみ表示に含まれているテーブルをダブルクリックします。テーブルには、結合列だけが表示されます。

2.13.5.4 すべてのテーブルの表示を変更する

選択したすべてのテーブルの表示を一度に変更するには、次のようにします。

- ・ [表示] > [テーブル表示の変更]の順に選択します。

2.14 スキーマの表示オプションの選択

[構造]枠でのテーブル、列、結合、カーディナリティの形状と表示方法をカスタマイズできます。

構造枠に表示するコンポーネントには、次の表示オプションが使用できます。

オプション	説明
結合表記	結合をいろいろな種類の直線、矢印やアリティなどのカーディナリティのインジケータを含んだ線など、さまざまな形式で表示することができます。
ベストサイド	このオプションを選択すると、2つのテーブルが自動的に最善の形で結合されます。つまり、テーブルの左右いずれか適切な側が選択され、2つのテーブルが最短距離で結合されます。
テーブル	テーブルの表示には、立体効果、エイリアス名表示、行数表示などのオプションを設定できます。行数を表示するには、[表示]>[テーブルの行数]の順に選択して行数を再度カウントする必要があります。詳細については、70 ページの「 データベーステーブルの行数の表示 」を参照してください。
列	列データの形式を列のすぐ横に表示することができます。キー列を下線付きで表示したり、列をテーブルの左揃えまたは中央揃えにすることができます。

オプション	説明
デフォルトの列数	テーブルに表示するデフォルトの列数を入力できます。テーブルの列数がデフォルト列数よりも多い場合、テーブルの最後の列のあとに省略記号 (...) が表示されます。テーブルを 1 回クリックすると、テーブルの横にスクロールバーが表示されます。
選択時に中央へ	[構造]枠は、計算された中央に従って表示されます。

2.14.1 構造枠にグラフィックオプションを設定

[構造]枠のコンポーネントのグラフィックオプションは、次のように設定します。

- 1 [[ツール]] > [[オプション]]の順に選択します。
[オプション]ダイアログボックスが表示されます。
- 2 [グラフィック]タブをクリックします。
[グラフィック]ページが表示されます。ここには、[構造]枠に設定できるグラフィックオプションがすべて表示されます。
- 3 希望の表示オプションを選択または入力します。
- 4 [OK]をクリックします。

2.14.1.1 表示オプションの設定例

ここでは、グラフィックオプション ([ツール] > [オプション] > [グラフィック]の順に選択) を使った構造枠のコンポーネントの表示例をいくつか紹介します。

エイリアス名の表示

このオプションを有効にして[構造]枠でエイリアステーブルを選択すると、テーブル名に加えて派生元のテーブル名が括弧内に表示されます。

行数と形式の表示

[行数を表示]オプションを使うと、テーブルに含まれる行数がテーブルシンボルの下に表示されます。行数を表示するには、[表示]>[テーブルの行数]の順に選択してデータベースのすべての行数をカウントしなおす必要があります。

[形式を表示]オプションを使うと、列のデータ型が記号で表示されます。それぞれのデータ型に対応する記号は、次の通りです。

- ・ Cは文字型
- ・ Dは日付型
- ・ Nは数値型
- ・ Tはロングテキスト型
- ・ L は BLOB (binary large object) 型

[構造]枠では、行数がテーブルシンボルの左下に、データタイプが列名の右端に表示されています。

2.14.2 テーブルと列の値の表示

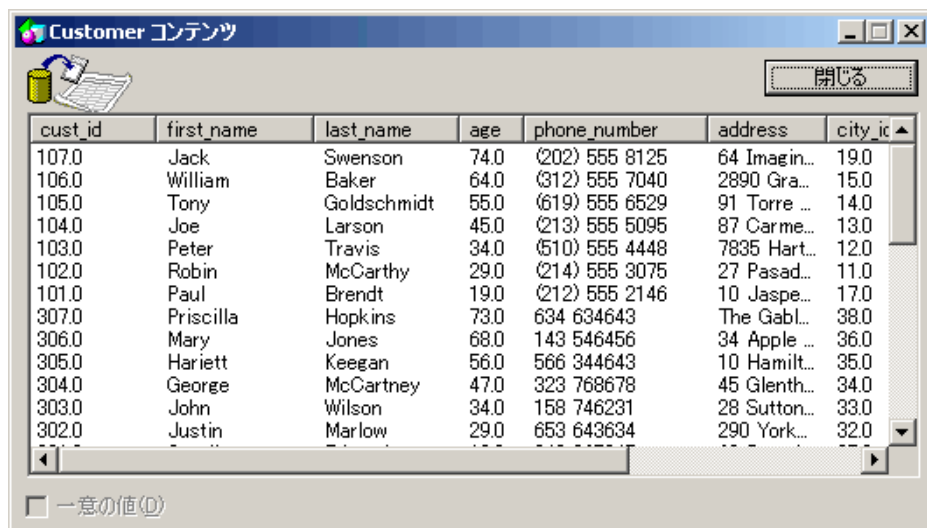
選択したテーブルまたは列の値を表示することができます。デフォルトで、テーブルに表示される行数は 100 行です。この値を変更し、必要に応じて行数を増やすことも減らすこともできます。

2.14.2.1 テーブルの値の参照

テーブルの値を表示する手順は、次の通りです。

- 1 値を参照するテーブルを[構造]枠でクリックします。
- 2 [表示]>[テーブル値]の順に選択します。

テーブルの各列の値を一覧表示した[(テーブル名) コンテンツ]ダイアログボックスが表示されます。



- 3 [一意の値]チェックボックスで、一意の値だけを表示することもできます。これをチェックすると、重複する値は表示されません。
- 4 [閉じる]をクリックします。

2.14.2.2 列の値の参照

列の値を表示する場合は、[表示] > [拡大表示]の順に選択して表示を拡大しておく、列の選択が容易になります。

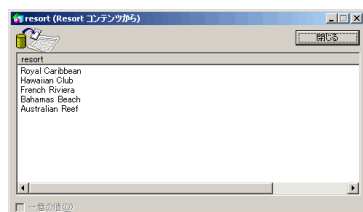
次のように、特定の列の値を表示します。

- 1 [構造]枠で、値を表示する列にマウスポインタを置きます。

ポインタが手の形に変わります。

- 2 マウスを右クリックし、ポップアップメニューから[列の値を表示]を選択します。

列の値を表示するダイアログボックスが開きます。



- 3 [一意の値]チェックボックスで、一意の値だけを表示することもできます。これをチェックすると、重複する値は表示されません。

- 4 [閉じる]をクリックします。

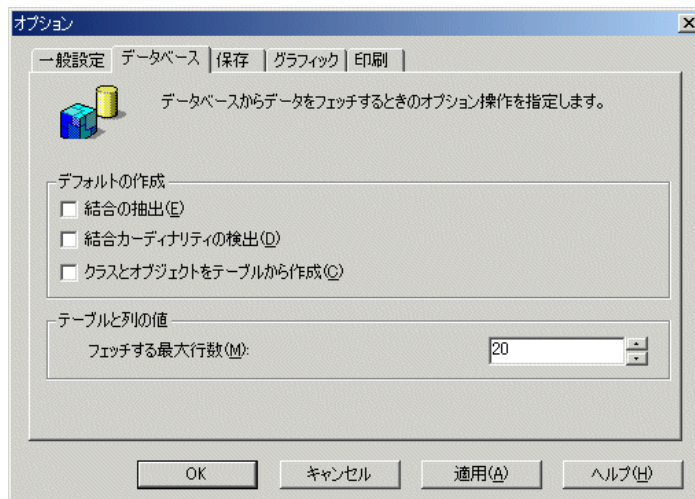
2.14.2.3 デフォルトの取得行数の変更

テーブルや列の値を見る場合に取得するデフォルトの行数を変更できます。テーブルに含まれる値のサンプルを見たい場合などは、デフォルト値を低くすることで取得行数を制限することができます。

各テーブルからフェッチする行数を変更する手順は、次の通りです。

- 1 [[ツール]] > [[オプション]]の順に選択します。
[オプション]ダイアログボックスが表示されます。
- 2 [データベース]タブをクリックします。
[データベース]ページが開きます。
- 3 [テーブルと列の値]グループのテキストボックスに値を直接入力するか、上下の矢印ボタンを使って設定します。

下の[データベース]ページでは、テーブルと列の値を表示する際、20 行だけを取得するように設定しています。



- 4 [OK]をクリックします。

2.14.3 データベーステーブルの行数の表示

各テーブルの行数を表示することができます。行数の表示は、次の2段階で行います。

- ・ [ツール] > [オプション] > [グラフィック]の順にクリックして、[行数を表示]オプションのチェックボックスをオンにする。
- ・ [表示] > [テーブルの行数]の順に選択して、すべてのテーブルの行数を最新表示する。

データベース内の各テーブルのすべての行数を表示できるだけでなく、特定のテーブルの行数を固定してクエリパフォーマンスを最適化できます。これにより、テーブルのウェイトにしたがって並べ替えられる From 句でのテーブル順を調整できます。詳細については、72 ページの「[テーブル行数の変更](#)」を参照してください。

注

テーブルの行数表示は、テーブルと列の値で取得する行数の設定とは関係がありません。

2.14.3.1 テーブルの行数の表示

各テーブルの行数を表示する手順は、次の通りです。

- 1 [[ツール]] > [[オプション]]の順に選択します。
[オプション]ダイアログボックスが表示されます。
- 2 [グラフィック]タブをクリックします。
[グラフィック]ページが表示されます。
- 3 [テーブル]グループボックスの[行数を表示]オプションをチェックします。
- 4 [OK]をクリックします。
- 5 行数を表示するテーブルを1つまたは複数選択します。

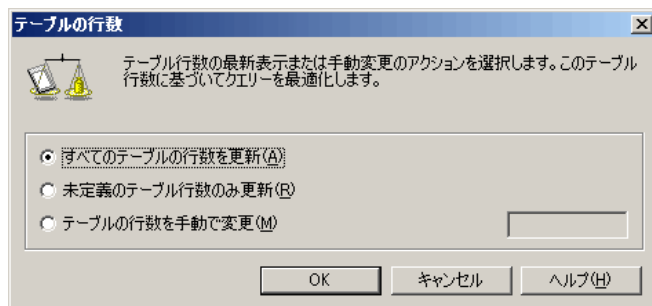
または

すべてのテーブルの行数を表示する場合は、[構造]枠の空白部分をクリックしてから、[編集] > [すべて選択]の順に選択します。

注

[構造]枠をクリックすると、[構造]枠内のコンポーネントに関連したメニューアイテムが使用可能になります。[構造]枠をクリックせずにメニューを選択すると、[ユニバース]枠に適用されるメニューアイテムしか使用できない恐れがあります。

- 6 [表示] > [テーブルの行数]の順に選択します。
[テーブルの行数]ダイアログボックスが表示されます。



このダイアログボックスのオプション内容は、次の通りです。

オプション	説明
すべてのテーブルの行数を更新	選択したテーブル、あるいは構造枠のすべてのテーブルの行数を更新します。
未定義のテーブル行数のみ更新	表示が選択されていないテーブルの行数を表示します。結果として[構造]枠のすべてのテーブルに行数が表示されます。
テーブルの行数を手動で変更	選択したテーブル、あるいは構造枠のすべてのテーブルの行数を変更します。オプションの横にあるテキストボックスに値を入力します。このオプションは、次のセクションで説明するクエリの最適化に使用します。

7 [すべてのテーブルの行数を更新]ラジオボタンをクリックします。

8 [OK]をクリックします。

選択したテーブルの行数が、[構造]枠のテーブルシンボルの左下に表示されます。

2.14.3.2 テーブル行数の変更

テーブルの行数設定を変更できます。変更する理由は、次の2つです。

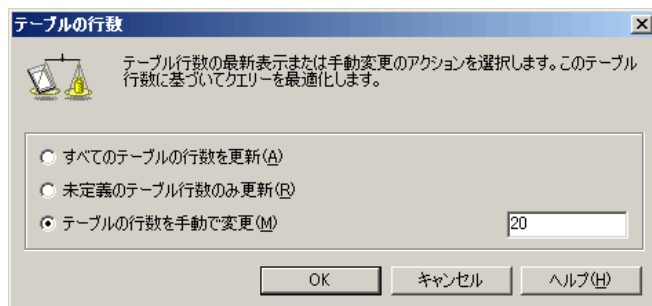
行数の変更理由	説明
クエリの最適化	クエリの最適化は、生成される SQL の FROM 句でのテーブル順に基づいています。テーブルは、行数の多い順に並べ替えられます。最適化機能を提供していないRDBMSでは、この順番が重要になります。 テーブルの行数設定を変更することで、From句でのテーブル順を変更できます。
データ量の変化に伴う行数の再設定	テーブルの行数設定を変更し、実際の行数がテーブルに設定した行数に反映されないようにできます。たとえば、テーブルが 50,000 行を含む場合でも 100 行のテストテーブルで作業することができます。

テーブルの行数を変更する手順は、次の通りです。

- 1 [[ツール]] > [[オプション]]の順に選択します。
[オプション]ダイアログボックスが表示されます。
- 2 [グラフィック]タブをクリックします。
[グラフィック]ページが表示されます。
- 3 [テーブル]グループボックスの[行数を表示]オプションをチェックします。
- 4 [OK]をクリックします。
- 5 行数を表示するテーブルを1つまたは複数選択します。

または

すべてのテーブルの行数を表示する場合は、[構造]枠の空白部分をクリックしてから、[編集] > [すべて選択]の順に選択します。
- 6 [表示] > [テーブルの行数]の順に選択します。
[テーブルの行数]ダイアログボックスが表示されます。
- 7 [テーブルの行数を手動で変更]ラジオボタンを選択します。
- 8 テーブルに設定する行数を入力します。



9 [OK]をクリックします。

選択したテーブルの行数が、[構造]枠のテーブルシンボルの左下に表示されます。

2.15 ユニバースの印刷

ユニバースデザインツールでは、すべてのWindows 標準印刷機能が提供されています。スキーマだけでなく、テーブル、列、結合などの[構造]枠のコンポーネント一覧を印刷することができます。また、印刷するコンポーネントと情報の書式を指定することもできます。

注

ユニバースの定義とスキーマは、ユニバースを PDF として保存して印刷することにより、プリントアウトにすることができます。詳しい説明は、50 ページの「[ユニバースの定義の PDF としての保存](#)」を参照してください。

2.15.1 印刷オプションの設定

[ツール] > [オプション] > [印刷/PDF]の順に選択して表示される[印刷/PDF]ページには、多くの印刷オプションがあります。ここで選択する印刷オプションは、ユニバースの定義をPDFとして保存する際、PDF ファイルに保存されるオプションにも適用されます。印刷および PDF オプションは、次の通りです。

印刷オプション	出力されるもの
一般情報	<p>次の情報を印刷します。</p> <ul style="list-style-type: none"> ユニバースのパラメータ リンクしたユニバース <p>[構造]枠のスキーマのグラフィック構造グラフィックの縮小率も設定できます。</p>

印刷オプション	出力されるもの
コンポーネント一覧	オブジェクト、条件、階層、テーブル、結合、コンテキストなどの種類別に、ユニバースのコンポーネントリストを印刷します。
完全説明	オブジェクト、条件、階層、テーブル、結合、コンテキストなどの各コンポーネントの詳細情報を印刷します。 印刷される内容は、各コンポーネントのプロパティ情報です。例えば、オブジェクトではSQL定義、資格、セキュリティアクセスレベル等が印刷されます。

ユニバースの印刷オプションを設定する手順は、次のとおりです。

- 1 [[ツール]] > [[オプション]]の順に選択します。
[オプション]ダイアログボックスが表示されます。
- 2 [印刷/PDF]タブをクリックします。
[印刷/PDF]ページが開きます。



- 3 設定する印刷オプションをチェックします。
- 4 [OK]をクリックします。

2.15.1.1 ページの設定

ページ設定オプションは、次の手順で設定します。

- 1 [ファイル] > [ページ]の順に選択します。

[ページ設定]ダイアログボックスが表示されます。

- 2 用紙のサイズと給紙方法、印刷の向き、余白に関するオプションを設定します。
- 3 [OK]をクリックします。

2.15.1.2 印刷プレビューの使い方

次のいずれかの操作で、ユニバースを印刷する前にプレビューを表示することができます。

- ・ [ファイル] > [印刷プレビュー]を選択します。
- ・ [印刷プレビュー]ボタンをクリックします。

2.15.1.3 ユニバースの印刷

ユニバースを印刷するには、次のどちらかの操作を行います。

- ・ [ファイル] > [印刷]を選択します。
- ・ [印刷]ボタンをクリックする。

ユニバースの作成とユニバースパラメータの設定

ユニバースを作成するには、まず新しいユニバースファイルを作成します。

新しいユニバースファイルを作成する際は、データベースミドルウェアへの接続パラメータを定義する必要があります。ユニバースデザインツールのオブジェクトの作成方法、現在のユニバースから別のユニバースへのリンク、クエリ制限などに関するパラメータも設定できます。

新しいユニバースを .unv ファイルとして保存します。このユニバースには、クラスとオブジェクトは一切含まれていません。ユニバースの開発プロセス中に、テーブルスキーマをデザインし、データベース構造にマップされたオブジェクトを作成することで、クラスとオブジェクトを作成します。

3.1 ユニバースのパラメータ

ユニバースのパラメータとは、ユニバースとそのデータベース接続を識別し、ユニバースを使って実行されるクエリのタイプを定義し、システムリソースの使用を制御するための一連の設定と制御を指します。

ユニバースのパラメータは、ユニバースを作成する際に表示される[ユニバースのパラメータ]ダイアログボックス([ファイル] > [パラメータ]) を使って設定します。データベース接続パラメータのみ、新しいユニバースを作成するときに手動選択または新規作成する必要があります。

パラメータはいつでも変更できます。ユニバースには、次のパラメータがあります。

パラメータ	説明
定義	ユニバース名、説明、接続パラメータ。これらのパラメータで、ユニバースを識別します。これらのパラメータの設定と変更についての説明は、81 ページの「 ユニバースの識別 」を参照してください。
要約	作成日時、更新日時、版数、デザイナーのコメント、および統計が表示されます。これらのパラメータの設定と変更についての説明は、80 ページの「 ユニバースの要約情報の表示と入力 」を参照してください。

パラメータ	説明
ストラテジー	ユニバースで使用するストラテジーが表示されます。ストラテジーは、データベースから構造情報を抽出するスクリプトです。これらのパラメータの設定と変更についての説明は、92 ページの「 ストラテジーの選択 」を参照してください。
制御	システムリソースの使用に関する制限が表示されます。これらのパラメータの設定と変更についての説明は、96 ページの「 リソースの制御 」を参照してください。
SQL	クエリパネルでエンドユーザが実行できるクエリの種類が表示されます。これらのパラメータの設定と変更についての説明は、96 ページの「 リソースの制御 」を参照してください。
リンク	リンクしているユニバースに関する設定が表示されます。これらのパラメータの設定と変更についての説明は、101 ページの「 ユニバースのリンクに指定するオプション 」を参照してください。

3.2 新しいユニバースの作成

ユニバースパラメータを定義してユニバースを保存することにより、新しいユニバースを一から作成する手順を、以下に説明します。[ユニバースのパラメータ]ダイアログボックスのすべてのタブの概要を紹介します。

それぞれのステップでの詳しい説明は、該当するパラメータのセクションを参照してください。

ユニバースの作成時に、すべてのパラメータを定義する必要はありません。接続を選択する必要がありますが、他のパラメータについてはデフォルト値をそのまま使用し、必要に応じて変更します。

3.2.1 新しいユニバースを一から作成

新しいユニバースを一から作成する手順は、次のとおりです。

- 1 [ファイル] > [新規]を選択します。

[ユニバースのパラメータ]ダイアログボックスの[定義]ページが表示されます。このページの詳しい説明は、81 ページの「[ユニバースの識別](#)」を参照してください。

注

ユニバースパラメータのオプションを選択すると、[ストアードプロシージャユニバースの選択]オプションが淡色表示されます。このオプションは選択も選択解除もできません。作成するユニバースの種類を変更するには、[キャンセル]をクリックして、やり直します。

- ・ ユニバース名と説明を入力します。
- ・ [接続]ドロップダウンリストから接続を選択 します。

または

- ・ ドロップダウンリストに適切な接続がない場合は[新規作成]ボタンをクリックし、新しい接続を作成します。新規接続の作成についての説明は、83 ページの「[接続の定義と編集](#)」を参照してください。

- 2 [要約]タブをクリックします。

[要約]ページが表示されます。このページの詳しい説明は、80 ページの「[ユニバースの要約情報の表示と入力](#)」を参照してください。

- ・ [コメント]ボックスに、ユニバースの情報を入力します。

- 3 [ストラテジー]タブをクリックします。

[ストラテジー]ページが表示されます。接続するデータソースに適用できるストラテジーが表示されます。このページの詳しい説明は、92 ページの「[ストラテジーの選択](#)」を参照してください。

- ・ [オブジェクト]、[結合]、[テーブル]のそれぞれに対し、ドロップダウンリストからストラテジーを選択します。

接続した RDBMS によって、ボックスに複数のストラテジーが選択肢として表示されることもあります。

- 4 [制御]タブをクリックします。

[制御]ページが表示されます。このページの詳しい説明は、96 ページの「[リソースの制御](#)」を参照してください。

- ・ [クエリ制限]ボックスで、適切なチェックボックスをオンまたはオフに設定します。
- ・ 選択したチェックボックスに、値を入力します。

- 5 [SQL]タブをクリックします。

[SQL]ページが表示されます。このページの詳しい説明は、98 ページの「[SQL の制限](#)」を参照してください。

- ・ 適切なチェックボックスをオンまたはオフにします。

- 6 新しいユニバースを既存のユニバースとリンクする場合は、[リンク]タブをクリックします。

[リンク]ページが開きます。このページの詳しい説明は、101 ページの「[ユニバースのリンクに指定するオプション](#)」を参照してください。

- ・ [リンクの追加]ボタンをクリックし、新規ユニバースとリンクするユニバースを選択します。

- 7 [パラメータ]タブをクリックします。

[パラメータ]ページが表示されます。SQL の生成を最適化するのに設定できる SQL パラメータが一覧表示されます。このページの詳しい説明は、101 ページの「[SQL 生成パラメータの設定](#)」を参照してください。

- 8 [OK]をクリックします。

ユニバースデザインツールでユニバース枠と構造枠が開きます。

- 9 [[ファイル]] > [[保存]]を選択します。

- ・ ユニバースのファイル名を入力します。
- ・ [保存]をクリックします。

3.3 ユニバースの要約情報の表示と入力

[要約]ページにユニバースの管理情報が表示されます。この情報を参考に、アクティブユニバースの構築過程を確認できます。

[要約]ページには、次の情報が表示されます。

情報	説明
作成	ユニバースの作成者と作成日時が表示されます。
更新	最後にユニバースを更新した日時と更新者名が表示されます。
版数	ユニバースが CMS にエクスポートされた回数を表します。
コメント	自分自身や他のデザイナーに対しての、ユニバースについてのコメントです。この情報は、ユニバースデザインツールでのみ提供されています。ユーザーに提供するコメントは、[ユーザ ID]ページの[説明]フィールドに入力します。
統計	ユニバースに含まれるクラス、オブジェクト、条件、テーブル、エイリアス、結合、コンテキスト、階層の数が表示されます。

3.4 ユニバースのパラメータの設定

ユニバースのパラメータは、以下を行うために設定できます。

- ・ 81 ページの「[ユニバースの識別](#)」
- ・ 83 ページの「[接続の定義と編集](#)」
- ・ 80 ページの「[ユニバースの要約情報の表示と入力](#)」
- ・ 92 ページの「[ストラテジーの選択](#)」
- ・ 98 ページの「[SQL の制限](#)」
- ・ 101 ページの「[ユニバースのリンクに指定するオプション](#)」
- ・ 101 ページの「[SQL 生成パラメータの設定](#)」

各種のパラメータは、[パラメータ]ダイアログボックス ([ファイル] > [パラメータ]) のページに含まれます。各パラメータグループは、以下の各セクションで説明されま

3.4.1 ユニバースの識別

それぞれのユニバースは、次のパラメータで識別されます。

識別子	適用
ファイル名 (8 文字)	ファイルシステム、Web Intelligence でのユニバース参照
ロングネーム (35 文字)	Web Intelligence ユーザ
説明	Web Intelligence ユーザー
一意の数値ID	CMS によるユニバースの識別この番号は、初めて CMS にエクスポートしたときにユニバースに割り当てられます。

名前と説明のパラメータは、ユニバースの作成時に[ユニバースのパラメータ]ダイアログボックスの[定義]タブで定義します。このユニバース識別パラメータは、いつでも変更できます。

データベース接続もこのタブで定義します。

新しい接続の作成方法は、83 ページの「[接続の定義と編集](#)」を参照してください。

次のユニバース識別パラメータを定義できます。

識別パラメータ	説明
名前	ユニバース名。Web Intelligence ユーザに表示されるユニバース名です。選択したリポジトリで利用できる文字は、ゼネラル Supervisor が設定します。(要修正)RDBMSにより、サポートされる文字は異なります。
説明	ユニバースの目的と内容の説明です。このフィールドはオプションです。この説明は、Web Intelligence ユーザーにも表示されますので、フィールドにユニバースの役割についての情報を入力しておくと便利です。
接続	名前の付いた一連のパラメータで、Web Intelligence からデータベース ファイル内のデータへアクセスする方法を定義します。ドロップダウンリストには、使用できるすべての接続が表示されます。新規に接続を作成することもできます。

3.4.1.1 ユニバースの識別パラメータの変更

ユニバースの識別パラメータを変更する手順は、次の通りです。

- 1 [ファイル] > [パラメータ]を選択します。

または

ツールバーの[ユニバースのパラメータ]ボタンをクリックします。

[ユニバースのパラメータ]ダイアログボックスの[定義]ページが表示されます。

- 2 名前と説明を入力します。
- 3 [接続]ドロップダウンリストから接続を選択します。
- 4 [テスト]をクリックし、接続が有効かどうかをチェックします。

サーバが応答していないという旨のメッセージが表示される場合、その接続は有効ではありません。[編集]をクリックして接続プロパティを編集し、接続パラメータを訂正してください。入力した情報がすべて正しい

にもかかわらずエラーメッセージが表示される場合は、RDBMS ドキュメンテーションのエラーメッセージに関する節を参照してください。

- 5 [OK]をクリックします。

3.4.2 接続の定義と編集

接続とは、Business Objectsアプリケーションからデータベース ファイル内のデータへのアクセス方法を定義した一連のパラメータに名前を付けたものです。接続によって、Web Intelligence はミドルウェアにリンクされます。データにアクセスするには、接続が必要です。

ユニバースを作成するときに、接続を選択または作成する必要があります。接続は、いつでも変更、削除、置き換えることができます。

注

接続オブジェクトには、“接続をローカルにダウンロード”という追加の管理者定義のセキュリティ権限があります。システム管理者は、慎重に扱うべき接続情報をローカルにダウンロードできるメンバーの定義など、接続関連のセキュリティを定義する必要があります。

注

接続の作成、変更、最適化の詳細については、『データアクセスガイド』を参照してください。

接続は、次の方法で管理できます。

接続の管理方法	説明
[接続パネル] から	メニュー内のコマンド: [ツール] > [接続]このパネルでは、アクセス可能な接続の一覧が表示されます。ログイン先の CMS 内にある [個人用]、[共有]、[セキュリティ] の各接続です。ユニバースデザインツールがスタンドアロンモードで起動された場合は、[個人用] 接続と [共有] 接続のみが表示されます。接続はフラットリスト、または階層形式で表示でき、後者の場合は接続はサブフォルダ内で表示されます。また、このダイアログボックスで新しい接続を作成、編集、削除することができます。
[ユニバースのパラメータ] ダイアログから	[ユニバースパラメータ] ダイアログの [定義] ページ ([ファイル] > [パラメータ] > [定義]) 新しい接続は、現在のユニバースで使用する適切な接続がない場合に作成します。接続を編集することもできます。

接続は、次の3つの要素で構成されます。

- ・ データアクセスドライバ
- ・ 接続パラメータとログインパラメータ
- ・ 接続の種類

次のセクションでは、それぞれの要素について説明します。

3.4.2.1 接続フォルダの管理の概要

ユニバースデザインツールを使用して接続フォルダを管理できます。セキュリティ接続が表示または管理できるあらゆる場所で、接続フォルダが表示されます。接続は、次の方法で管理できます。

接続の管理方法	説明
接続パネル	メニュー内のコマンド: [ツール] > [接続]このパネルでは、アクセス可能な接続の一覧が表示されます。ログイン先の CMS 内にある [個人用]、[共有]、[セキュリティ] の各接続です。ユニバースデザインツールがスタンドアロンモードで起動された場合は、[個人用] 接続と [共有] 接続のみが表示されます。接続はフラットリスト、または階層形式で表示でき、後者の場合は接続はサブフォルダ内で表示されます。また、このダイアログボックスで新しい接続を作成、編集、削除することができます。検証後、[接続パネル] から実行されるすべてのアクションは、CMS 内で自動的にコミットされます。
ユニバースパラメータダイアログ	[ユニバースパラメータ] ダイアログボックスの [定義] ページ ([ファイル] > [パラメータ] > [定義])新しい接続は、現在のユニバースで使用する適切な接続がない場合に作成します。接続を編集することもできます。

実行できるアクションはユーザ権利によって異なりますが、使用可能なアクションは次のとおりです。

- ・ 新規接続の作成
- ・ 新規接続フォルダの作成
- ・ 接続またはフォルダの編集
- ・ 接続または空のフォルダの削除
- ・ 接続またはフォルダの名前の変更
- ・ 接続に関する説明の編集
- ・ 接続のプロパティの表示
- ・ 編集モードでの切り取り、コピー、貼り付け

3.4.2.2 データアクセスドライバ

データアクセスドライバとは、ユニバースをミドルウェアに接続させるためのソフトウェアレイヤです。

データアクセスドライバは、Business Objects製品に含まれています。サポート対象のミドルウェアそれぞれに対し、個別のデータアクセスドライバが準備されています。ユニバースデザインツールをインストールする際に指定するデータアクセスキーによって、インストールされるドライバが決まります。

新しい接続を作成する際は、目的のRDBMSへの接続に使用するRDBMSミドルウェア用のデータアクセスドライバを選択してください。

3.4.2.3 接続およびログインパラメータ

次の接続およびログインパラメータを指定して、データアクセスドライバを設定します。

パラメータ	説明
種類	接続の種類: 個人、共有、またはセキュリティ
名前	接続名を表します。
ユーザ名	データベースのユーザ名です。通常、この名前はデータベース管理者から指定されます。
パスワード	そのデータベースのパスワード。通常、この名前はデータベース管理者から指定されます。
表示時のレポートの最新表示にシングルサインオンを使用する	このオプションを選択すると、CMS へのアクセスに使用したユーザ名とパスワードが、自動的にデータベースへのログインパラメータとして使用されます。シングルサインオンの設定については、『Business Objects Enterprise 管理者ガイド』を参照してください。
BusinessObjects ユーザアカウントに関連づけられたデータベース認証情報を使用する	このオプションを選択した場合、レポートを最新表示するために、BusinessObjects アカウントに関連付けられたデータベースユーザパスワードを入力する必要があります。これは、セントラル管理コンソールレベルで設定されます。このオプションの設定については、『BusinessObjects Enterprise 管理者ガイド』を参照してください。
データソース/サービス	データソース名またはデータベース名です。ODBC ドライバを使用する場合は、ターゲットデータベースをデータソース名で識別します。ネイティブドライバを使用している場合、データベース名でターゲットデータベースを識別します。

注

ODBC を利用した Excel ファイルおよび .csv 形式のテキストファイルへの接続を作成できます。ODBC を利用してアクセスしたテキストファイルまたは Excel ファイルに基づいてユニバースを Web Intelligence で使用するには、msjet.prm ファイルを接続用に編集する必要があります。

このファイルは、\$INSTALLDIR\$/BusinessObjects Enterprise 12.0/win32_x86/dataAccess/connectionserver/odbc フォルダにあります。ここで、\$INSTALLDIR\$ は、Business Objects アプリケーションがインストールされているディレクトリです。msjet.prm ファイルでは、DB_TYPE パラメータを次のように変更します。

変更前: <Parameter Name='DB_TYPE'>MS Jet Engine</Parameter>

変更後: <Parameter Name='DB_TYPE'>MS Jet</Parameter>

この変更を行った後で、Business Objects Enterprise サーバを停止して再起動する必要があります。注: Web Intelligence サーバと同じマシンでユニバースデザインツールを実行している場合、この値を変更した後でテキストまたは Excel ファイルに基づいて追加のユニバースを作成するには、この値を <Parameter Name='DB_TYPE'>MS Jet Engine</Parameter> にリセットする必要があります。

3.4.2.4 接続の種類

接続の種類で、データにアクセスする接続を使用できるユーザを決定します。ユニバースデザインツールでは、ワークセッション中にユーザが作成した接続がすべて自動的に保存されます。次にセッションを開始するとき、ユーザはこれらの接続を使用することができます。

ユニバースデザインツールで作成できる接続は、次の 3 種類です。

- ・ 個人
- ・ 共有
- ・ セキュリティ

接続の種類は、次のとおりです。

個人接続

データにアクセスできるのは、ユニバースを作成したコンピュータと作成者のみです。

接続パラメータは、たとえば次のようなユーザプロファイルディレクトリの Business Objects 12.0 フォルダにある LSI フォルダの PDAC.LSI ファイルに格納されます。

C:\Documents and Settings\<ユーザ名>\Application Data\Business Objects\Business Objects 12.0\lsi

これらのパラメータはスタティックで、更新できません。

Business Objects 製品のセキュリティ基準に照らし合わせると、この接続は安全とはいえません。

ユニバースを配布するときには、個人接続を使用しないでください。個人接続は、ローカルマシンで個人用データにアクセスするために使用します。

共有接続

すべてのユーザがデータにアクセスできます。Business Objects 製品のセキュリティ基準に照らし合わせると、この接続は安全とはいえません。

接続パラメータは、たとえば次のようなユーザプロファイルディレクトリの Business Objects 12.0 フォルダにある lsi フォルダの SDAC.LSI ファイルに格納されます。

C:\Documents and Settings\<ユーザ名>\Application Data\Business Objects\Business Objects 12.0\lsi

セキュリティ接続

- ・ データへのアクセスを集中管理します。もっとも安全な接続ですので、重要なデータへのアクセスにはこの接続を使用してください。
- ・ ユニバースデザインツールでは、セキュリティ接続を作成できます。
- ・ CMS 経由でユニバースを配布する場合は、セキュリティ接続が必要となります。
- ・ セキュリティ接続は、いつでも使用でき、また更新できます。

3.4.2.5 個人用接続と共有接続でのパスワードの設定

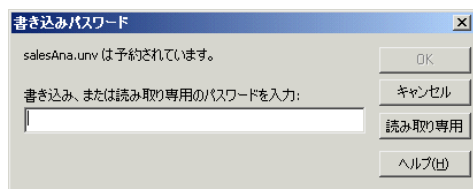
個人用接続および共有接続を持つユニバースには、パスワードを設定することができます。パスワードを使用することにより、リポジトリのない環境で不正なユーザからユニバースを保護することができます。

注

パスワードを忘れるとユニバースを復旧できなくなります。ユニバースパスワードのバックアップファイルを作成してください。

パスワードには、2 つの異なるオプションを設定することができます。

- ・ [読み取りパスワード]を設定すると、パスワードを入力するダイアログボックスが表示されます。パスワードが正しいと、ユニバースを開くことができます。
- ・ “書き込みパスワード”を設定すると、次のダイアログボックスが表示されます。



正しいパスワードを入力することにより、ユニバースを「読み取り専用」モードまたは「読み取り/書き込み」モードで開くことができます。

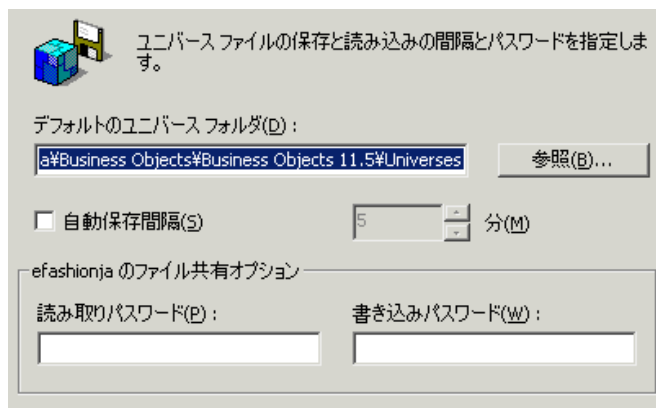
個人用接続と共有接続でパスワードを設定する手順は、次のとおりです。

- 1 [ツール] > [オプション]の順に選択します。

[オプション]ダイアログボックスが表示されます。

- 2 [保存]タブをクリックします。

[保存]ページが表示されます。



- 3 [保存]ページに表示される[読み取りパスワード]または[書き込みパスワード]ボックスに、パスワードを入力します。半角で最高40文字までの英数字を入力できます。
- 4 [OK]をクリックします。

3.4.2.6 ユーザの DBPass 変更後のデータベースへのアクセス

BusinessObjects 管理者は、データベースのユーザパスワードが変更された後も BusinessObjects ユーザログイン(名前とパスワード)で引き続きデータにアクセスできるようにすることができます。

以下のパラメータを設定すると、BusinessObjects ユーザは、データベースのパスワードが変更された後も BusinessObjects 管理者に連絡せずにデータへのアクセスを継続できます。

- ・ セントラル管理コンソールの[ログオン時に、ユーザのログオン情報をデータソースログオン情報として有効化、更新する]チェックボックスをオンにする。
- ・ ユニバースデザインツールの新規接続ウィザードの[新しい接続の定義]ページで、[BusinessObjects ユーザアカウントに関連したデータベース認証情報を使用する]チェックボックスおよび[表示時のレポートの最新表示にシングルサインオンを使用する]チェックボックスをオンにします。

上記のチェックボックスがオンの場合、更新された DBUser パラメータおよび DBPass パラメータは自動的に BusinessObjects のユーザアカウントに関連付けられます。

注

DBUser および DBPass は静的パラメータなので、セントラル管理コンソールで更新する必要があります。データベースのパスワードが変更された場合は、セントラル管理コンソールで各ユーザのパスワードを更新することをお勧めします。

3.4.2.7 新しい接続の定義

新しいデータベース接続は、新規接続ウィザードを使って作成します。このウィザードは、次の場所からアクセスできます。

- ・ [ユニバースのパラメータ]ダイアログボックスの[定義]ページ([ファイル] > [パラメータ] > [定義])普通、ユニバースでアクセスするデータに対する接続が作成されていない場合に、新しい接続を作成します。
- ・ [接続一覧]([ツール] > [接続])

接続ウィザードを使って、接続の詳細パラメータとカスタムパラメータを設定することができます。接続の作成、変更、最適化の詳細については、『データアクセスガイド』を参照してください。

[ユニバースのパラメータ]ダイアログボックスから接続を作成した場合は、[ユニバースのパラメータ]ダイアログボックスに戻り、[接続]ボックスに新しく作成した接続が表示されます。

[接続]ダイアログボックスから接続を作成した場合は、一覧に接続が表示されます。

関連項目

- ・ 443 ページの[新規接続ウィザードを開始する](#)

3.4.2.8 使用できる接続の一覧表示

[接続一覧] には、使用できるすべての接続が一覧表示されます。既存の接続を編集したり、新規接続を作成することができます。

接続の一覧は、次の手順で表示します。

- 1 [ツール] > [接続]を選択します。

[接続一覧]が表示されます。このダイアログボックスには、現在のユニバースで使用できるすべての接続が一覧表示されます。

- 2 ダイアログボックスを閉じるには、[キャンセル]をクリックします。

[接続]ダイアログボックスから既存の接続を編集することができます。

セキュリティ接続は、オンラインモードでしか編集できません。個人用接続と共有接続は、どちらのモードでも変更できます。

既存の接続の名前を変更することはできません。

3.4.2.9 接続の編集

接続を編集する手順は、次の通りです。

- 1 [ツール] > [接続]を選択します。

[接続一覧]が表示されます。

- 2 表示される接続の一覧から、編集する接続をクリックします。

- 3 [編集]をクリックします。

接続の[ログイン]ページが開きます。

- 4 必要に応じて、[データソース名]ボックスに新しいデータソース名またはデータベース名を入力します。

- 5 必要に応じてログインパラメータを変更します。

- 6 [次へ]をクリックします。

[テストの実行]ページが開きます。

- 7 [接続テスト]をクリックし、変更した接続をテストします。

- 8 [次へ]をクリックして[詳細パラメータ]と[カスタムパラメータ]ページに進みます。必要に応じてパラメータを変更します。デフォルトの値や既存の値をそのまま使用することができます。

- 9 [カスタムパラメータ]ページで[終了]をクリックし、変更を適用します。

3.4.2.10 接続の削除

[接続一覧]ダイアログボックスでは、既存の接続を削除することができます。セキュリティ接続は、オンラインモードでしか削除できません。個人接続と共有接続は、どちらのモードでも削除できます。

接続を削除する手順は、次の通りです。

- 1 [ツール] > [接続]を選択します。

[接続一覧]が表示されます。

- 2 表示される接続の一覧から、削除する接続をクリックします。

- 3 [削除]をクリックします。

削除を確認するダイアログボックスが表示されます。

- 4 [はい]をクリックします。

接続がリストから削除されました。

3.4.2.11 新しい接続の追加

[ツール] > [接続]で開く[接続]ページで[追加]をクリックすると、[新規接続]ウィザードを使って新しい接続を作成することができます。この接続ウィザードの使い方に関する説明は、91 ページの「[新しい接続の追加](#)」を参照してください。

3.4.3 ユニバース要約パラメータの設定

[ユニバースのパラメータ]ダイアログボックスの[要約]タブには、ユニバースの管理情報が表示されます。この情報を参考に、アクティブユニバースの構築過程を確認できます。

[要約] ページには、次の情報が表示されます。

情報	説明
作成	ユニバースの作成者と作成日時が表示されます。
更新	最後にユニバースを更新した日時と更新者名が表示されます。
版数	ユニバースをリポジトリにエクスポートした回数
コメント	ユーザやデザイナーに対してのユニバースについてのコメントが表示されます。
統計	ユニバースに含まれるクラス、オブジェクト、条件、テーブル、エイリアス、結合、コンテキスト、階層の数が表示されます。

3.4.4 要約情報を表示および入力する方法

- 1 [ファイル] > [パラメータ] を選択します。

または

[パラメータ]ボタンをクリックします。

[ユニバースのパラメータ] ダイアログボックスが表示されます。

- 2 [要約]タブをクリックします。
[要約]ページが開きます。
- 3 [コメント]ボックスにコメントを入力します。
- 4 [OK]をクリックします。

3.4.5 ストラテジーの選択

ストラテジーは、データベースやテキストファイルからデータ構造の情報を自動的に抽出するスクリプトです。ストラテジーの主要目的は、次の2つです。

- ・ 結合とカーディナリティの自動検出(結合ストラテジー)
- ・ クラス、オブジェクト、結合の自動作成(オブジェクトストラテジーと結合ストラテジー)

ストラテジーは、データベースの SQL 構造を自動的に検出し、それに従ってユニバース構造を自動的に作成する場合に役立ちます。

注

ユニバース構造を自動作成するストラテジーは、ユニバースのデザインと作成に不可欠なものではありません。ストラテジーは、データベースやデータベースデザインツールに含まれる既存のメタデータ情報を利用して、手早くユニバースを作成したい場合に便利です。ただし、ユーザニーズの分析から得たリレーションシップを基にオブジェクトと結合を作成する場合には、ストラテジーの提供する自動作成機能に頼るべきではありません。

ユニバースデザインツールでは、次の 2 種類のストラテジーを指定できます。

ストラテジー	説明
ビルトインストラテジー	ユニバースデザインツールで提供されているデフォルトのストラテジービルトインストラテジーはカスタマイズできません。
外部ストラテジー	ビルトインストラテジーと同じ種類の情報を含んだスクリプトで、データベースからの情報抽出を最適化できるようにカスタマイズしたユーザ定義のストラテジー

3.4.5.1 ストラテジーの選択

ストラテジーを選択する手順は、次の通りです。

- 1 [ファイル] > [パラメータ]を選択します。
または
[パラメータ]ボタンをクリックします。
[ユニバースのパラメータ]ダイアログボックスが表示されます。
- 2 [ストラテジー]タブをクリックします。
[ストラテジー]ページが表示されます。
- 3 [オブジェクト]、[結合]、[テーブル]のそれぞれに対し、ドロップダウンリストからストラテジーを選択します。
- 4 [OK]をクリックします。

3.4.5.2 ビルトインストラテジー

ビルトインストラテジーは、出荷時に設定されているユニバースデザインツールのデフォルトストラテジーです。サポートするすべてのデータベースに対して、それぞれビルトインストラテジーがあります。ビルトインストラテジーは変更できません。ドロップダウンリストでは、デフォルトでビルトインストラテジーが外部ストラテジーより前に表示されます。

ビルトインストラテジーは、次の目的で使します。

ストラテジー	目的
オブジェクト	テーブルスキーマでのテーブル作成時に、デフォルトのクラスとオブジェクトを自動的に作成*

ストラテジー	目的
結合	<ul style="list-style-type: none"> ・ テーブルスキーマでのテーブル作成時に、デフォルトの結合を自動的に抽出* ・ 結合作成時にカーディナリティを自動的に挿入* ・ テーブルスキーマ内の結合を自動的に検出: [ツール] > [自動検出] > [結合の検出]を選択すると、ユニバースデザインツールはストラテジーを使用して自動的に結合の候補を検出します。検出された結合を使用するかどうかは、デザイナーが決定します。 ・ テーブルスキーマにある結合のカーディナリティを自動的に検出し、挿入: [ツール] > [自動検出] > [カーディナリティの検出]を選択すると、ユニバースデザインツールはストラテジーを使用して、テーブルスキーマ内で選択された結合のカーディナリティを検出します。
テーブル	[テーブル参照]ダイアログボックスに表示されるテーブルの情報の制限

*これらの自動作成機能を使うには、[オプション]ダイアログボックスの[データベース]ページで、対応するオプションを有効にする必要があります。

オブジェクトストラテジーの使い方

オブジェクトストラテジーは、テーブルスキーマにテーブルを追加する際に自動的にクラスとオブジェクトを作成するためだけに使用されます。このストラテジーを使用するには、[オプション]ダイアログボックスの[データベース]ページで対応するストラテジーを有効にする必要があります。詳細は、95 ページの [「ストラテジーの自動作成機能の使い方」](#)を参照してください。

結合ストラテジーの使い方

選択した結合ストラテジーに従って、ユニバースデザインツールで実行される、テーブルスキーマ内のカーディナリティと結合の自動検出の方法が設定されます。

使用するデータベースによって、適用できる結合ストラテジーが一覧で表示されます。たとえば Oracle データベースを使う場合、列名の一致や列番号の一致などをもとにした自動検出の結合ストラテジーを選択できます。

ストラテジーを選択しなかった場合は、ユニバースデザインツールは列名に一致するデフォルト結合ストラテジーを使用して結合を検出します。選択した結合ストラテジーは、特に有効に設定する必要はありません。テーブルスキーマで結合やカーディナリティの検出を選択すると、指定したストラテジーが必ず使用されます。

結合ストラテジーは、結合を自動的に作成し、それにカーディナリティを設定するためにも使用されます。このストラテジーの自動デフォルト作成機能をするには、[オプション]ダイアログボックスの[データベース]ページで対応するストラテジーを有効にする必要があります。詳細は、95 ページの [「ストラテジーの自動作成機能の使い方」](#)を参照してください。

テーブルストラテジーの使い方

選択したテーブルストラテジーがデータベーステーブルの構造を読み取ります。ストラテジーによって、[テーブル参照]ダイアログボックスに表示される情報が変わります。表示される情報は、列のデータ型と説明などです。

3.4.5.3 ストラテジーの自動作成機能の使い方

ストラテジーの自動作成および挿入機能は、デフォルトでは無効になっています。これらの機能を使用するには、オブジェクトまたは結合の作成時に適用するストラテジーに対応する[デフォルトの作成]チェックボックスを選択する必要があります。これらのオプションは、[オプション]ダイアログボックスの[データベース]ページ([ツール] > [オプション] > [データベース])にあります。

[データベース]ページの[デフォルトの作成]オプションは、次のとおりです。

オプション	非選択時	選択時
結合の抽出	結合を手動で作成する必要があります。[ツール] > [自動検出] > [結合の検出]を選択すると、ユニバースデザインツールはストラテジーを使用して結合を検出し、結合の候補を表示します。検出された結合を使用するかどうかは、デザイナーが決定します。	選択した結合ストラテジーにしたがってテーブルをリンクする結合で、テーブルを受信します。
結合カーディナリティの検出	カーディナリティを手動で定義する必要があります。[ツール] > [自動検出] > [カーディナリティの検出]を選択すると、ユニバースデザインツールはストラテジーを使用して、選択された結合のカーディナリティを検出し、設定します。	カーディナリティの検出と適用は、結合の作成時に結合に含まれます。
デフォルトのクラスとオブジェクトをテーブルから作成	クラスとオブジェクトは、[ユニバース]枠で直接作成するか、または[構造]枠から[ユニバース]枠にテーブルまたは列をドラッグして手動で作成する必要があります。	デフォルトのクラスとオブジェクトは、テーブルを[構造]枠に追加するときにユニバース枠で自動作成されます。クラスはテーブル名に、オブジェクトは列名に対応しています。アンダースコア(_)はすべてスペースに置換されます。

ストラテジーのデフォルト作成オプションを選択する手順は、次の通りです。

- 1 [[ツール]] > [[オプション]]の順に選択します。
[オプション]ダイアログボックスが表示されます。
- 2 [データベース]タブをクリックします。
[データベース]ページが開きます。
- 3 ストラテジーに使用するデフォルトの作成オプションを選択します。
- 4 [OK]をクリックします。

3.4.5.4 表示する行数の設定

[オプション]ダイアログボックスの[データベース]ページで、データベースの各テーブルから表示する最大行数を指定することができます。データベースから受信する、物理的な行数を制限することはできませんが、この設定では、テーブルまたは列の値を表示する場合に一度に表示できる行数のデフォルト値を変更できます。この設定はユニバースデザインツールで返された行には適用されますが、Web Intelligence で実行されるクエリには適用されません。

表示する行数を設定する手順は、次のとおりです。

- ・ [フェッチする最大行数]テキストボックスに数値を入力します。上下の矢印ボタンを使って、デフォルトの値(100)を変更することもできます。

3.4.5.5 外部ストラテジーの使い方

外部ストラテジーとは、ユーザ定義の SQL スクリプトで、定義された出力構造に従って、カスタマイズされた自動ユニバース作成タスクを実行します。外部ストラテジーは、外部の XML ストラテジーファイル (<RDBMS>.STG) に格納されます。このファイルにある SQL スクリプトが、その他のストラテジーと一緒に、[ストラテジー] ページのドロップダウンリストに表示されます。

外部ストラテジーには、ビルトインストラテジーと同じ種類の情報が含まれていますが、多くの場合、ユニバースデザインツールが特定のデータベースの情報を抽出したり、データベースからの情報抽出を最適化したりするようにカスタマイズされています。

外部ストラテジーの定義の詳細については、402 ページの「[外部ストラテジーを使用したユニバース作成のカスタマイズ](#)」を参照してください。

3.4.6 リソースの制御

ユニバースデザインツールには、システムリソースの使用を制御するオプションがいくつか用意されています。

注

[制限のプレビュー]ダイアログボックスでこのタブが表示される場合、制限に適用されるパラメータで修正されたものは赤く表示されます。

3.4.7 使用できるシステムリソースオプション

次の制限をシステムリソースに指定することができます。

クエリ制限	説明
結果の最大行数	クエリの結果返される行数を、指定行数に制限します。返される行数は制限されるようになりますが、RDBMS ではクエリに含まれるすべての行が処理されます。RDBMS での処理結果が返される行数のみが制限されます。
最大実行時間	クエリの最大実行時間を指定することができます。このオプションによって、データが Web Intelligence に送信される時間を制限しますが、データベースでの処理は制限しません。
ロングテキスト型オブジェクトの最大の長さ	<p>ロングテキスト型オブジェクトの最大文字数を指定することができます。</p> <p>このチェックボックスをオンにしなければ、このパラメータは有効になりません。自動的に、デフォルトの最大値 (1000) に設定されます。デフォルトより大きい結果を得られるようにするには、このチェックボックスをオンにして、値を入力する必要があります。</p>

3.4.8 リソース制御情報を入力する

- 1 [ファイル]>[パラメータ]を選択します。
または
[パラメータ]ボタンをクリックします。
[ユニバースのパラメータ] ダイアログボックスが表示されます。

- 2 [制御]タブをクリックします。
[制御]ページが表示されます。
- 3 [クエリ制限]グループボックスで適用する制限のチェックボックスをオンにします。
選択したオプションのテキストボックスに適切な値を入力します。テキストボックスの横にある矢印をクリックして、入力した値を増減することもできます。
- 4 [OK]をクリックします。

3.4.9 複数の SQL 文を生成するクエリでの最大実行時間

クエリの実行時間に指定する制限時間は、クエリの総実行時間です。クエリに複数の SQL 文が含まれている場合、各 SQL 文に割り当てられる実行時間は、クエリの総実行時間を SQL 文の数で割った時間になり、各 SQL 文に同じ実行時間が割り当てられます。

したがって、クエリ内の特定の文がクエリ全体の実行時間の大半を占めるような場合は、本来その文の実行に必要な時間が確保できず、タスクが途中で終了してしまう恐れがあります。

複数の SQL 文を持つクエリの実行時間を制限する場合、最も時間のかかる文の実行時間を考慮し、その時間を SQL 文の数倍してください。

3.4.10 SQL の制限

エンド ユーザーが Web Intelligence の[クエリー]枠で作成できるクエリーの種類を制御することができます。

次の領域でのクエリ生成を制御することができます。

- ・ 個々のクエリでのサブクエリ、演算子、複合オペランドの使用
- ・ 複数の SQL 文の生成
- ・ デカルト積の使用不可または警告

各制御についての説明は、次のセクションに記載されています。

3.4.10.1 クエリの制御

個々のクエリで次の制御を設定できます。

オプション	説明
ユニオン、インターセクト、マイナス演算子の使用	データセット演算子 (ユニオン、インターセクト、マイナス) を使ってクエリを組み合わせ、結果セットを受信することができます。

3.4.10.2 複数の SQL 文の制御

次の制御を設定して、複数の SQL 文を処理する方法を指定できます。

オプション	説明
各コンテキストに対して複数の SQL 文を生成	コンテキストを使用するときに、複数の SQL 文を含んだクエリを作成できます。ユニバースにコンテキストが含まれる場合に、このオプションを選択します。
各メジャーに対して複数の SQL 文を生成	異なるテーブルにある列からのメジャーオブジェクトがクエリに含まれるときに、SQL を複数の文に分割します。このオプションの詳細については、251 ページの「 [各メジャーに対して複数の SQL 文を生成]オプションの使い方 」を参照してください。 メジャーオブジェクトが同じテーブルの列に基づいている場合、このオプションが選択されていても SQL 文は分割されません。
マルチコンテキストの選択	複数のコンテキストにあるオブジェクトでクエリを作成し、1 つの結果セットを生成できます。 コンテキストを使用して、ループ、キャズムトラップ、ファントラップ、その他の結合パス問題を解決する場合は、このチェックボックスを選択しないでください。

3.4.10.3 デカルト積の制御

デカルト積は、クエリに含まれる各テーブルの各行の可能な限りの組み合わせが含まれている結果セットです。ほとんどの場合、デカルト積の結果は間違っています。

デカルト積の生成に対して、次の制御を設定できます。

オプション	説明
使用不可	このオプションを選択すると、デカルト積を生成するクエリは実行されません。
警告	クエリ結果がデカルト積になることを、ユーザにメッセージで警告します。

3.4.10.4 SQL への制限の適用

SQLに制限を適用する手順は、次の通りです。

- 1 [ファイル] > [パラメータ]を選択します。

または

[パラメータ]ボタンをクリックします。

[ユニバースのパラメータ]ダイアログボックスが表示されます。

- 2 [SQL]タブをクリックします。

[SQL]ページが表示されます。

- 3 [クエリ]と[マルチパス]のそれぞれのグループボックスにあるオプションを設定します。

- 4 [デカルト積]グループボックスのラジオボタンを選択します。

- 5 [OK]をクリックします。

3.4.11 ユニバースのリンクに指定するオプション

[リンク]タブは、ユニバースの動的リンクに使用されます。動的リンクの詳細については、541 ページの「[ユニバースの展開](#)」の章を参照してください。

3.4.12 SQL 生成パラメータの設定

ユニバースデザインツールでは、多くの RDBMS に共通する SQL パラメータを動的に設定し、Web Intelligence 製品でユニバースを使用して生成される SQL を最適化できます。

3.4.12.1 旧バージョンの ユニバースデザインツールでのパラメータ(PRM)ファイルの使用

旧バージョンの Designer 6.5 では、ユニバースが使う SQL 生成パラメータは、パラメータ(PRM)ファイルと呼ばれる個別のファイルに記録され、自由に編集できました。PRM ファイルに設定された値は、接続に定義されたデータアクセスドライバを使って、すべてのユニバースに適用されていました。

現バージョンでは、クエリ生成を最適化する SQL パラメータの大多数が各ユニバースファイルで制御されるようになりました。ユニバースデザインツールで設定できるクエリ生成パラメータには PRM ファイルは使用されていませんが、データベース固有のパラメータにはまだ PRM ファイルが使用されています。

注

データアクセスドライバの PRM ファイルに関する詳しい説明は、『データアクセスガイド』を参照してください。
[ヘルプ] > [データアクセスガイド] を選択すると、『データアクセスガイド』にアクセスできます。

3.4.12.2 ユニバースデザインツールでの SQL パラメータの動的設定

サポートする大部分の RDBMS ミドルウェアに共通の多数のパラメータが、[ユニバースのパラメータ]ダイアログボックスの[パラメータ]タブ([ファイル] > [パラメータ] > [パラメータ])で編集できます。

これらのパラメータは、現在のユニバースだけに適用され、UNV ファイルに記録されます。ユニバースデザインツールでユニバースの SQL パラメータを変更すると、接続用データアクセスドライバの PRM ファイルに定義された値ではなく、ユニバースデザインツールで定義した値が使用されます。

3.4.12.3 SQL 生成パラメータの編集

SQL パラメータの値を変更し、ユニバースを使って SQL を生成する製品での SQL 生成方法を制御することができます。

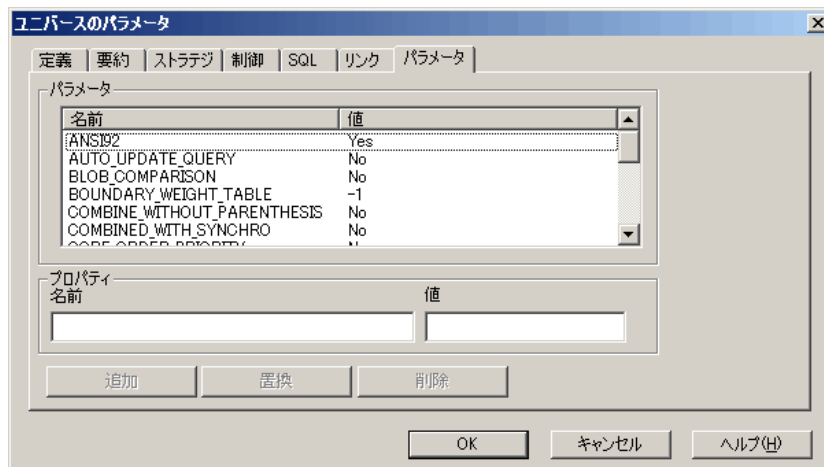
SQL生成パラメータは、次の手順で編集します。

- 1 [ファイル] > [パラメータ]を選択します。

[パラメータ]ダイアログボックスが開きます。

- 2 [パラメータ]タブをクリックします。

[パラメータ]ページが開きます。



- 3 次の要領でパラメータを編集、追加、削除します。

目的	操作
新しいパラメータの追加	<ul style="list-style-type: none"> ・ 一覧からパラメータをクリックします。 ・ [名前]ボックスにパラメータの名前を入力します。 ・ [値]ボックスに値を入力します。 ・ [追加]をクリックします。 ・ 新しい値が一覧の最後に追加されます。

目的	操作
名前または値の変更	<ul style="list-style-type: none"> 一覧のパラメータをクリックします。 [名前]ボックスに新しい名前を入力します。 [値]ボックスに新しい値を入力します。 [置換] をクリックします。 <p>変更された設定で値が保存されます。</p>
パラメータの削除	<ul style="list-style-type: none"> 削除するパラメータを一覧から選択します。 [削除]をクリックします。

4 [OK]をクリックします。

注

ユニバースに設定した SQL 生成パラメータの値は、そのユニバースを使う製品でしか使用できません。

3.4.13 SQL 生成パラメータについて

次のリファレンスに、クエリスクリプトの生成に影響するパラメータの説明を示します。これらのパラメータについて、2 つのグループ内でアルファベット順に説明します。

- ユニバースデザインツールまたはインフォメーションデザインツールのユーザインタフェースに設定する SQL パラメータ。これらは、ほとんどのデータアクセスドライバに共通する SQL パラメータです。各パラメータは、それが設定されているユニバースに対して有効です。
- PRM ファイルに設定する SQL パラメータ。これらは接続固有のパラメータであり、ターゲットのデータアクセスドライバのデータアクセスパラメータ (PRM) ファイルにリストされています。

関連項目

- 103 ページの[ユーザインタフェースに設定する SQL パラメータ](#)」
- 124 ページの[PRM ファイルに設定する SQL パラメータ](#)」

3.4.14 ユーザインタフェースに設定する SQL パラメータ

3.4.14.1 ANSI92

ANSI92 = Yes|No

値	Yes/No
デフォルト	No
説明	生成された SQL が ANSI92 標準に準拠するかどうかを指定します。 Yes: ANSI92 標準に準拠した SQL 生成を有効にします。 No: PRM パラメータ OUTER_JOIN_GENERATION に従って SQL を生成します。

3.4.14.2 AUTO_UPDATE_QUERY

AUTO_UPDATE_QUERY = Yes|No

値	Yes/No
デフォルト	No
説明	クエリのオブジェクトがユーザプロファイルで利用できない場合の動作を指定します。 Yes: クエリが更新され、オブジェクトがクエリから削除されます。 No: オブジェクトがクエリに残されます。

3.4.14.3 BACK_QUOTE_SUPPORTED

BACK_QUOTE_SUPPORTED = Yes|No

値	YES: SQL でバックスラッシュを使用します。 NO: SQL でバックスラッシュを使用しません。
デフォルト	YES OpenAccess データベースについては No
説明	SQL で、スペースまたは特殊文字を含むテーブル名または列名を囲むためにバックスラッシュを使用するかどうかを指定します。
結果	Table name=`My Table`

3.4.14.4 BEGIN_SQL

BEGIN_SQL = <文字列>

値	文字列
デフォルト	空の文字列

説明	<p>会計、優先順位付け、負荷管理に使用する SQL 文のプレフィックスとして使用します。このパラメータは、ドキュメント生成、LOV クエリを含むすべての SQL 生成に適用されます。</p> <p>このパラメータは、Web Intelligence、LiveOffice、および QaaWS でサポートされています。しかし、Desktop Intelligence および Crystal Reports では無視されます。</p> <p>Teradata の例</p> <pre>BEGIN_SQL=SET QUERY_BAND='string' for transaction;</pre> <p>このパラメータには、1 つ以上の名前と値のペアが含まれる文字列が必要です。これらのペアは、セミコロンで区切られ、単一引用符で囲まれています。すべての SQL 文は、BEGIN_SQL に続くこのパラメータがプレフィックスとして付いています。このパラメータに入力される名前と値のペアは、GetQueryBandPairs システムテーブルに入力されます。</p> <p>3 つの名前と値のペアの例:</p> <pre>BEGIN_SQL=SET QUERY_BAND='UserID=Jones;JobID=980;AppID=TRM' for transaction;</pre> <p>@Variable 関数を名前と値のペアの値として使用することもできます。戻り値は、単一引用符で囲まれます。BEGIN_SQL=SET QUERY_BAND='USER=@Variable('BOUSER');Document=@Variable('DPNAME');' for transaction;</p>
----	--

3.4.14.5 BLOB_COMPARISON

BLOB_COMPARISON = Yes|No

値	Yes/No
デフォルト	No
編集の可/不可	不可

説明	<p>BLOB ファイルが DISTINCT 句で使用されている場合、クエリを SELECT 句とともに生成するかどうかを指定します。クエリプロパティの No Duplicate Row の設定に関連しています。</p> <p>Yes: DISTINCT 句をクエリ内で使用できます。</p> <p>No: クエリ設定の No Duplicate Row がオンになっていても、クエリ内で DISTINCT 文を使用できません。</p>
----	--

3.4.14.6 BOUNDARY_WEIGHT_TABLE

BOUNDARY_WEIGHT_TABLE = Integer 32bits [0-9]

値	Integer 32bits [0 ～ 9、または負の整数]
デフォルト	-1
説明	<p>テーブルに多くの行がある場合、FROM 句の最適化を可能にします。</p> <p>テーブルのサイズ(行数)が入力値より大きい場合、テーブルはサブクエリとして宣言されます。</p> <p>FROM (SELECT col1, col2,....., coln,, FROM Table_Name WHERE simple condition)</p> <p>簡単な条件は、サブクエリをもたないものとして定義されます。</p> <p>-1、0、または任意の負の数は、この最適化が使用されていないことを意味します。</p>
制限	<p>最適化は以下の場合実装できません。</p> <ul style="list-style-type: none">・ OR 演算子がクエリ条件にある。・ 1 つのテーブルだけが SQL に関係している。・ クエリに外部結合が入っている。・ 最適化するテーブルで条件が指定されていない。・ 最適化するテーブルが派生テーブルである。

3.4.14.7 COLUMNS_SORT

COLUMNS_SORT = Yes|No

値	Yes/No
デフォルト	No
説明	列を[構造]枠のテーブルに表示する順序を指定します。 Yes: 列は文字順に表示されます。 No: 列はデータベースから取得された順序で表示されます。

3.4.14.8 COMBINE_WITHOUT_PARENTHESES

COMBINE_WITHOUT_PARENTHESES= Yes|No

値	Yes/No
デフォルト	No
説明	クエリに UNION、INTERSECT、または MINUS 演算子がある場合、クエリを かっこでカプセル化するかどうかを指定します。RedBrick で使用されます。 Yes: かっこを削除します。 No: かっこを残します。

3.4.14.9 COMBINED_WITH_SYNCHRO

COMBINED_WITH_SYNCHRO = Yes|No

値	Yes No
デフォルト	No
説明	<p>UNION, INTERSECTION、または EXCEPT 演算子を使用されており、各サブクエリのオブジェクトに互換性がないクエリを実行させるかどうか指定します。</p> <p>Yes: UNION, INTERSECTION、または EXCEPT 演算子を使用されており、各サブクエリのオブジェクトに互換性がないクエリを実行させるよう指定します。このタイプのクエリは、同期(レポート内の 2 つのブロック)を生成します。</p> <p>No: UNION, INTERSECTION、または EXCEPT 演算子を使用されており、各サブクエリのオブジェクトに互換性がないクエリを実行させないように指定します。そのようなクエリが実行されると、次のようなエラーメッセージが表示されます:「複雑すぎるクエリ:互換性のないオブジェクトを含んだサブクエリがあります。」これはデフォルトの値です。</p>

3.4.14.10 COMPARE_CONTEXTS_WITH_JOINS

COMPARE_CONTEXTS_WITH_JOINS = Yes|No

値	Yes No
デフォルト	Yes
説明	<p>コンテキストの比較方法を指定します。</p> <p>Yes: 各コンテキストで同じ結合が指定されているかどうかを検証します。</p> <p>No: 各コンテキストで同じテーブルの集合が指定されているかどうかを検証します。これはデフォルト値です。</p>

3.4.14.11 CORE_ORDER_PRIORITY

CORE_ORDER_PRIORITY = Yes|No

値	Yes No
デフォルト	No
説明	<p>このパラメータは、リンクされた派生ユニバースに追加したクラスまたはオブジェクトに適用されます。このパラメータは、コアユニバースまたは元の派生ユニバース内のクラスまたはオブジェクトには適用されません。このパラメータは、新しいクラスおよびオブジェクトのユニバースデザインツールでの整理方法を指定します。</p> <p>FIRST_LOCAL_CLASS_PRIORITY パラメータも参照してください。</p> <p>Yes: クラスおよびオブジェクトは次の順に整理されます。</p> <ul style="list-style-type: none"> 第 1 のコアユニバースクラス コアユニバースオブジェクト 第 1 のコアユニバースクラスに属するすべての派生ユニバースオブジェクト 第 2 のコアユニバースクラス コアユニバースオブジェクト 第 2 のコアユニバースクラスに属するすべての派生ユニバースオブジェクト 他のコアユニバースクラス... 派生ユニバースクラスおよびオブジェクト <p>No: クラスやオブジェクトが、派生ユニバースの元の順序に従うよう指定します。これはデフォルト値です。</p>

3.4.14.12 CORRECT_AGGREGATED_CONDITIONS_IF_DRILL

CORRECT_AGGREGATED_CONDITIONS_IF_DRILL = Yes|No

値	Yes No
デフォルト	No
説明	<p>Desktop Intelligence のみに適用されます。Desktop Intelligence で、メジャーをクエリや条件で集計するかどうかを指定します。</p> <p>Yes: クエリでドリルが有効になっている場合に、Desktop Intelligence で、メジャーをメインクエリと条件で個別に集計します。</p> <p>No: クエリでドリルが有効になっている場合に、Desktop Intelligence では、メジャーをメインクエリと条件で個別に集計できません。</p>

3.4.14.13 CUMULATIVE_OBJECT_WHERE

CUMULATIVE_OBJECT_WHERE = Yes|No

値	Yes No
デフォルト	No

説明	<p>このパラメータは、フィルタされたオブジェクトのみに適用されます。WHERE 句とこれらのオブジェクトのクエリ条件を組み合わせる方法を指定します。</p> <p>Yes: WHERE 句が、メインクエリの条件と AND 演算子で組み合わせられるように指定します。</p> <p>No: オブジェクトの WHERE 句が、このオブジェクトの条件と組み合わせられるように指定します。</p> <p>例</p> <p>John ではないフランス人のクライアントすべて、またはニューヨークではないアメリカの都市を探す場合、SQL は次のようになります。</p> <p>Yes:</p> <pre>(customer.first_name <> 'John') OR (city.city <> 'New York' AND customer_country.country = 'France' AND city_country.country = 'USA')</pre> <p>No:</p> <pre>(customer.first_name <> 'John' AND customer_country.country = 'France') OR (city.city <> 'New York' AND city_country.country = 'USA')</pre>
----	---

3.4.14.14 DECIMAL_COMMA

DECIMAL_COMMA = Yes|No

値	Yes No
デフォルト	No

説明	<p>Business Objects 製品で、必要に応じてカンマを小数点区切りとして挿入するかどうかを指定します。</p> <p>Yes: Business Objects 製品で、必要に応じて、カンマが小数点区切りとして挿入されます。</p> <p>No: Business Objects 製品は、小数点記号としてカンマを追加しません。これはデフォルト値です。</p>
----	---

3.4.14.15 DISABLE_ARRAY_FETCH_SIZE_OPTIMIZATION

DISABLE_ARRAY_FETCH_SIZE_OPTIMIZATION = Yes|No

値	Yes/No
デフォルト	No
説明	<p>最適化アルゴリズムを使用して、デフォルト設定を使用せずに、返される配列のサイズを最適化することができます。</p> <p>No: ユニバースで実行されるすべてのクエリが、最適化によるメリットを享受します。</p> <p>Yes: クエリは、デフォルト値のセットを使用します。</p> <p>このパラメータは、OLAP 接続にも当てはまります。</p>

3.4.14.16 DISTINCT_VALUES

DISTINCT_VALUES = GROUPBY|DISTINCT

値	GROUPBY DISTINCT
---	------------------

デフォルト	DISTINCT
説明	<p>[重複行は無視する]オプションが有効な場合、SQL を DISTINCT または GROUPBY 句と共に値の一覧やクエリパネルに生成するかどうかを指定します。</p> <p>DISTINCT: SQL が DISTINCT 句とともに生成されます。次はその例です。</p> <pre>SELECT DISTINCT cust_name FROM Customers</pre> <p>GROUPBY: SQL が GROUP BY 句とともに生成されます。次はその例です。</p> <pre>SELECT cust_name FROM Customers GROUP BY cust_name</pre>

3.4.14.17 END_SQL

END_SQL = 文字列

値	文字列
デフォルト	<空の文字列>
説明	このパラメータで指定した文が、各 SQL 文の末尾に追加されます。
例	<p>IMB DB2 データベースで、次のように使用できます。</p> <pre>END_SQL=FOR SELECT ONLY</pre> <p>サーバで、データブロックの読み取りが高速で行われます。</p> <p>次の例は別の宣言方法です。</p> <pre>END_SQL='write ' UNVID To Usage_Audit.Querieded_universe</pre> <p>監査テーブルにユニバース ID が書き込まれます。これは、ユーザ、クエリされたテーブルなど他のデータを記録するために使用されます。</p>

3.4.14.18 EVAL_WITHOUT_PARENTHESES

EVAL_WITHOUT_PARENTHESES = Yes|No

値	Yes No
デフォルト	No
説明	<p>デフォルトで、@Select(クラス¥オブジェクト) 関数が、かっこでくくられた <クラス¥オブジェクト> オブジェクトの SELECT 句で置き換えられます。</p> <p>たとえば、2 つの @Select 句を組み合わせた @Select(object1) *@Select(object2) は、</p> <p>SQL(object1) = A-B で SQL(object2) =C とすると、</p> <p>(A-B) * (C) となります。</p> <p>デフォルトではかっこが追加されますが、EVAL_WITHOUT_PARENTHESES = Yes に設定することによってこれを回避できます。この場合、A-B * C となります。</p> <p>Yes: かっこが SELECT(クラス¥オブジェクト) 関数の @Select 句から削除されます。</p> <p>No: かっこが @Select(クラス¥オブジェクト) 関数の SELECT 句を囲むように追加されます。</p>

3.4.14.19 FILTER_IN_FROM

FILTER_IN_FROM = Yes|No

値	Yes No
デフォルト	No

説明	<p>クエリ条件を FROM 句に含めるかどうかを指定します。この設定は、他のユニバースパラメータの設定 ANSI92 が Yes. に設定されている場合にのみ適用されます。</p> <p>Yes: 外部結合を編集する際、ユニバースデザインツールの[結合の詳細プロパティ]ダイアログボックスのドロップダウンリストボックスで選択したデフォルトの動作プロパティが[FROM 句のオブジェクトすべて]に設定されます。</p> <p>No: 外部結合を編集する際、ユニバースデザインツールの[結合の詳細プロパティ]ダイアログボックスのドロップダウンリストボックスで選択したデフォルトの動作プロパティが[FROM 句のオブジェクトなし]に設定されます。</p>
----	---

3.4.14.20 FIRST_LOCAL_CLASS_PRIORITY

FIRST_LOCAL_CLASS_PRIORITY = Yes|No

値	Yes No
デフォルト	No
説明	<p>このパラメータは、Desktop Intelligence のみに適用されます。</p> <p>CORE_ORDER_PRIORITY=Yes の場合のみ考慮されます。</p> <p>Yes: 派生ユニバースのクラスが最初に表示されます。</p> <p>No: 派生ユニバースのオブジェクトとサブクラスがコアユニバースのオブジェクトとサブクラスの後に表示されます。</p>

3.4.14.21 FORCE_SORTED_LOV

FORCE_SORTED_LOV = Yes|No

値	Yes No
---	--------

デフォルト	No
説明	並べ替えられた値の一覧を取得します。 Yes: 値の一覧を並べ替えられるよう指定します。 No: 値の一覧を並べ替えないよう指定します。

3.4.14.22 INNERJOIN_IN_WHERE

INNERJOIN_IN_WHERE = Yes|No

値	Yes No
デフォルト	No。パラメータを有効にするには、そのパラメータを手動で追加する必要があります。
説明	ANSI92 が Yes に設定されている場合、SQL 構文がすべての内部結合と共に WHERE 句に強制的に生成されるよう指定できます。この設定は、クエリに FULL OUTER、RIGHT OUTER、または LEFT OUTER 結合がなく、内部結合だけが含まれる場合に限り有効です。 Yes: クエリに内部結合だけが含まれる場合を除き、ANSI92 が Yes に設定されると、ANSI92 結合構文が FROM 句に生成されます。この場合、内部結合は WHERE 句に含まれます。 No: ANSI92 が Yes に設定されている場合、ANSI 句に FROM 92 結合構文が生成されます。

3.4.14.23 JOIN_BY_SQL

JOIN_BY_SQL = Yes|No

値	Yes No
デフォルト	No
説明	<p>複数の SQL 文をどのように処理するかを指定します。データベースで許可されている場合に限り、複数の SQL 文を組み合わせることができます。</p> <p>Yes: 複数の SQL 文が組み合わせられることを指定します。</p> <p>No: 複数の SQL 文が組み合わせられないことを指定します。これはデフォルト値です。</p>

3.4.14.24 MAX_INLIST_VALUES

MAX_INLIST_VALUES = [0-99]

値	整数: 最小 -1、最大は DB に依存
デフォルト	-1
説明	<p>IN LIST 演算子を使用する場合、条件に入力できる値の最大数を設定することができます。</p> <p>99 : IN LIST 演算子を使用して条件を作成するとき、値を 99 個まで入力できるよう指定します。</p> <p>入力できる最大値はデータベースに依存します。</p> <p>値が -1 の場合は、データベースによる制限を除いて、返される値の数に制限はありません。</p>

3.4.14.25 OLAP_UNIVERSE

OLAP_UNIVERSE = Yes|No

値	Yes No
デフォルト	デフォルト値はありません。
説明	<p>OLAP ユニバースが使用されているかどうかを示します。ユニバースデザインツールで OLAP ユニバースが使用されている場合、値は Yes に設定され、このパラメータが SQL パラメータの一覧に表示されます。ユニバースが OLAP ユニバースではない場合、このパラメータは SQL パラメータの一覧には表示されません。</p> <p>Yes: ユニバースは OLAP ユニバースです。</p> <p>No: ユニバースは OLAP ユニバースではありません。</p>

3.4.14.26 PATH_FINDER_OFF

パラメータは、デフォルトでは一覧表示されません。手作業でパラメータをリストに追加して、値を設定する必要があります。

PATH_FINDER_OFF= Yes|No

値	Yes No
デフォルト	デフォルト値はありません。パラメータを入力する必要があります。
説明	<p>結合の生成がデータベースで行われるため、HPIW に使用します。</p> <p>Yes: 結合がクエリで生成されません。</p> <p>No: 結合がクエリで生成されます。これはデフォルトの動作です。</p>

3.4.14.27 REPLACE_COMMA_BY_CONCAT

REPLACE_COMMA_BY_CONCAT= Yes|No

値	Yes No
デフォルト	No
説明	<p>旧バージョンのユニバースデザインツールでは、オブジェクトの SELECT 句の複数フィールドを区切るのにカンマを使用することができました。カンマは連結演算子として処理されました。すでにカンマがこのように使われているユニバースの場合、REPLACE_COMMA_BY_CONCAT を No に設定してこの動作を維持できます。現バージョンのユニバースデザインツールでは、デフォルトでこのパラメータが Yes に設定されます。その結果、カンマを複数フィールドの区切りとして使用する式は、自動的に連結構文を使用するように設定されます。</p> <p>Yes: 複数フィールドオブジェクトがある場合、カンマが連結式に置き換えられます。</p> <p>No: カンマをそのままにしておきます。</p>

3.4.14.28 SELFJOINS_IN_WHERE

SELFJOINS_IN_WHERE = Yes|No

値	Yes No
デフォルト	No
説明	<p>自己結合は通常、FROM 句に含まれます。このパラメータにより、自己結合のすべての条件を WHERE 句に指定した SQL 構文を生成することができます。このパラメータが考慮されるようにするには、ANSI92 パラメータを Yes に設定する必要があります。</p> <p>パラメータを有効にするには、そのパラメータを手動で一覧に追加する必要があります。</p> <p>Yes: 自己結合の条件は、SQL クエリの WHERE 句に含まれます。</p> <p>No: 自己結合の構文は ANSI 規格に基づいて生成され、自己結合の条件は SQL クエリの ON 句のテーブル結合定義の FROM 句に含まれます。</p>

3.4.14.29 SHORTCUT_BEHAVIOR

SHORTCUT_BEHAVIOR = Global|Successive

値	Global Successive
デフォルト	Successive
説明	<p>ショートカット結合の適用方法を指定します。このパラメータは、以前 GLOBAL_SHORTCUTS PRM ファイルで としてリストされていました。値は Global の場合が Yes、Successive の場合が No. に変更されました。</p> <p>Global: ショートカット結合を 1 つずつ検討します。ショートカット結合は、1 つ以上のテーブルをバイパスし、次に続くショートカット結合で使用される結合パスからテーブルを削除しない場合に限り適用されます。</p> <p>Successive: すべてのショートカット結合を適用するよう指定します。注: デカルト積を生成する場合、ショートカット結合は適用されません。</p>

3.4.14.30 SMART_AGGREGATE

SMART_AGGREGATE = Yes|No

値	Yes No
デフォルト	No

説明	<p>集計テーブルに基づくスマートメジャーに対する集計テーブルの使用方法を決定します。これにより、比率に基づくユニバースオブジェクトが正しく集計されるようになります。デフォルトでは、集計テーブルの計算済みの値を利用するため、これらのテーブルがすべての期間(異なる期間)を通じて整合性がない場合は、このパラメータを使用して最も詳細な集計テーブルが使用されるようにしてください。</p> <p>このパラメータは、ユニバースパラメータリストには表示されず、デフォルトでは無効になります。ユニバース設計者は、このパラメータを有効(値 Yes)にする前にパラメータリストに手動で挿入する必要があります。</p> <p>Yes: 追加のグループ設定クエリは、集計テーブルに基づくスマートメジャーに対する最初のクエリの集計テーブルを基盤とします。</p> <p>No: システムは、最も適切な集計テーブルを選択します。</p>
----	---

3.4.14.31 STORED_PROC_UNIVERSE

STORED_PROC_UNIVERSE = Yes|No

値	Yes No
デフォルト	No
説明	<p>ストアードプロシージャが含まれるユニバースを作成すると、この値は自動的に Yes に設定されます。この値は手動で変更しないでください。</p> <p>Yes: 作成中または編集集中のユニバースには、ストアードプロシージャが含まれています。</p> <p>No: ユニバースには、ストアードプロシージャは含まれていません。</p>

3.4.14.32 THOROUGH_PARSE

THOROUGH_PARSE = Yes|No

値	Yes No
デフォルト	No
説明	<p>クエリパネルのデフォルトの分析および個々のオブジェクトの分析に使用する手法を指定します。</p> <p>Yes: PREPARE, DESCRIBE および EXECUTE 文を使用して、オブジェクトの SQL を分析します。</p> <p>Prepare+DescribeCol+Execute</p> <p>No: PREPARE および DESCRIBE 文を使用して、オブジェクトの SQL を分析します。</p>

3.4.14.33 TRUST_CARDINALITIES

TRUST_CARDINALITIES = Yes|No

値	Yes No
デフォルト	No
説明	<p>結果が増大した場合に SQL を最適化できます。</p> <p>Yes: メジャーを含むクエリの場合は、メジャーを増大させ、また[結果オブジェクト]に表示されないすべての条件が、サブクエリに変換されます。これによって、メジャーに対して False の結果を返す可能性があるテーブルはクエリに含まれなくなります。</p> <p>No: 最適化は実装されません。</p>

3.4.14.34 UNICODE_STRINGS

UNICODE_STRINGS = Yes|No

値	Yes No
デフォルト	No
説明	<p>現在のユニバースで Unicode 文字列を操作できるかどうかを指定します。Microsoft SQL Server および Oracle 9 にのみ適用されます。SBO ファイルでデータベースの文字セットが Unicode に設定されていれば、SQL の生成を修正し、NCHAR と NVARCHAR のような特定の Unicode 列の型を処理できるようにする必要があります。</p> <p>Yes: 文字列に基づく条件が、UNICODE_PATTERN ファイルのパラメータ PRM の値に従って SQL で書式設定されます。たとえば、MS SQL Server(sql srv.prm) の場合、UNICODE_PATTERN=N\$ となります。</p> <p>条件 Customer_name='Arai' は</p> <p>Customer_name=N'Arai' となります。</p> <p>注: Unicode 値を基に @Prompt 構文を使って新しいプロンプトを作成すると、データ型は 'C' ではなく 'U' になります。</p> <p>No: 文字列に基づくすべての条件は、標準の SQL の書式で表現されます。たとえば、Customer_name='Arai' はそのまま Customer_name='Arai' となります。</p>

3.4.15 PRM ファイルに設定する SQL パラメータ

3.4.15.1 CASE_SENSITIVE

<Parameter Name="CASE_SENSITIVE">NO</Parameter>

説明	データベースで大文字と小文字が区別されるかどうかを指定します。このパラメータは Oracle で使用します。
----	--

値	YES: データベースで大文字と小文字が区別されます。 NO: データベースで大文字と小文字は区別されません。
デフォルト	NO

3.4.15.2 CHECK_OWNER_STATE

<Parameter Name="CHECK_OWNER_STATE">NO</Parameter>

説明	データベースが所有者名によるテーブル分類をサポートしているかどうかを SQL がチェックするか指定します。
値	YES: データベースが所有者名によるテーブル分類をサポートしているかどうかを SQL がチェックします。 NO: データベースが所有者名によるテーブル分類をサポートしているかどうかを SQL がチェックしません。
デフォルト	YES

3.4.15.3 CHECK_QUALIFIER_STATE

<Parameter Name="CHECK_QUALIFIER_STATE">NO</Parameter>

説明	データベースが修飾子によるテーブル分類をサポートしているかどうかを SQL がチェックするか指定します。
値	YES: データベースが修飾子によるテーブル分類をサポートしているかどうかを SQL がチェックします。 NO: データベースが修飾子によるテーブル分類をサポートしているかどうかを SQL がチェックしません。

デフォルト	YES
-------	-----

3.4.15.4 COMMA

<Parameter Name="COMMA">||' '</Parameter>

説明	次の構文で指定されたオブジェクトで、カンマに代わって使用されるデータベース連結演算子を指定します。 Tab.Col1, Tab.Col2 このパラメータは、すべてのデータアクセスドライバによって使用されます。
値	' ' +' '+
デフォルト	' '
結果	Tab.Col1 ' 'Tab.Col2

3.4.15.5 CONCAT

<Parameter Name="CONCAT">||</Parameter>

説明	連結演算子を指定します。このパラメータは、すべてのデータアクセスドライバで使用されます。
値	二重縦線 () またはプラス記号 (+)
デフォルト	

3.4.15.6 DATE_WITHOUT_QUOTE

<Parameter Name="DATE_WITHOUT_QUOTE">YES</Parameter>

説明	SQL 構文で日付を単一引用符で囲むかどうかを指定します。このパラメータは、MS Access で使用されます。
値	YES: 日付を単一引用符で囲みません。 NO: 日付を単一引用符で囲みます。
デフォルト	YES

3.4.15.7 DELIMIT_LOWERCASE

<Parameter Name="DELIMIT_LOWERCASE"></Parameter>

説明	小文字の識別子を引用符で区切るかどうかを指定します。
値	YES: 小文字の識別子を引用符で区切ります。 NO: 小文字の識別子を引用符で区切りません。

3.4.15.8 EXTERN_SORT_EXCLUDE_DISTINCT

<Parameter Name="EXTERN_SORT_EXCLUDE_DISTINCT">YES</Parameter>

説明	ORDER BY 句を含んだクエリで SELECT DISTINCT を生成するかどうかを指定します。
----	---

値	YES: クエリに ORDER BY 句が含まれている場合、SELECT DISTINCT を生成しません。 NO: クエリに ORDER BY 句が含まれている場合、SELECT DISTINCT を生成します。
デフォルト	YES

3.4.15.9 GROUPBY_WITH_ALIAS

<Parameter Name="GROUPBY_WITH_ALIAS">YES</Parameter>

説明	SELECT 文内にエイリアスを含む GROUP BY 句をデータベースで作成できるかどうかを指定します。
値	YES: SELECT 文内で GROUP BY 句を作成できます。 NO: SELECT 文内では、エイリアスを含む GROUP BY 句を作成できません。
デフォルト	YES

3.4.15.10 IDENTIFIER_DELIMITER

<Parameter Name="IDENTIFIER_DELIMITER">"</Parameter>

説明	<p>以下の機能を指定します。</p> <ul style="list-style-type: none">・ BACK_QUOTE_SUPPORTED パラメータが有効な場合、スペースあるいは特殊文字を含んだテーブル名と列名を引用符で囲みます。・ DELIMIT_IDENTIFIERS パラメータが有効な場合、テーブル名または列名に含まれる文字に関係なく、これらの名前を引用符で囲みます。 <p>このパラメータを使用するには、BACK_QUOTE_SUPPORTED または DELIMIT_IDENTIFIERS のいずれかを YES に設定する必要があります。これは、両方のパラメータのデフォルト値です。</p>
値	<p>” (二重引用符): スペースあるいは特殊文字を含んだテーブル名と列名は、二重引用符で囲まれます。</p> <p>’ (単一引用符): スペースあるいは特殊文字を含んだテーブル名と列名は、単一引用符で囲まれます。この値は、Microsoft Access でのみ使用できます。</p>
デフォルト	”
結果	Table name=“My Table”

3.4.15.11 IF_NULL

<Parameter Name=“IF_NULL”>NO</Parameter>

説明	2 つのパラメータを取る関数を指定します。最初のパラメータが NULL を返すと、2 つ目のパラメータ値が使用されます。
値	データベースによって異なります。
デフォルト	データベースによって異なります。

3.4.15.12 OUTERJOINS_COMPLEX

<Parameter Name="OUTERJOINS_COMPLEX"></Parameter>

説明	このパラメータは、OUTERJOINS_GENERATION とともに外部結合クエリの生成を管理します。
値	YES NO

3.4.15.13 OUTERJOINS_GENERATION

<Parameter Name="OUTERJOINS_GENERATION">ANSI92</Parameter>

このパラメータは、デフォルトの外部結合生成動作を制御します。次のように設定できます。

- ・ 外部結合生成は ANSI92 仕様に準拠しています。
- ・ 外部結合生成は、以前のバージョンのユニバースデザインツールと変わりありません。

注

PRM ファイルの OUTERJOINS_GENERATION パラメータは、ユニバースの ANSI92 設定に次のように関連します。

- ・ PRM ファイルの OUTERJOINS_GENERATION パラメータが ANSI92 に設定され、ユニバースの ANSI92 設定が NO に設定されている場合、PRM パラメータはユニバース設定を上書きし、外部結合は ANSI92 の動作に準拠します。
- ・ PRM ファイルの OUTERJOINS_GENERATION パラメータが USUAL に設定されている場合、ユニバースの ANSI92 設定が優先され、外部結合は、ユニバースの ANSI92 設定が YES か NO に応じて ANSI92 に準拠します。

留意点

ANSI92 値は REVERSE_TABLE_WEIGHT パラメータが SQL 生成の最適化に有用であることを妨げます。ANSI92 の動作に準拠する外部結合は、SQL 文のテーブルの順序を統制します。

説明	<p>外部結合のSQL構文を指定します。</p> <p>値が ANSI 92 の場合は、FROM 句に外部結合が生成されます。他の値では、WHERE 句に外部結合が生成されます。</p> <p>この設定を変更する場合、結合プロパティをチェックして外部結合式が有効であること、およびカーディナリティが正しいことを確認する必要があります。ANSI92 は、結合構文での手動でのカスタマイズをサポートしていません。</p>
----	--

値	<p>OUTERJOINS_GENERATION の主な値は次の通りです。</p> <ul style="list-style-type: none"> ANSI92: ユニバースデザインツールの ANSI92 パラメータ値に関係なく、デフォルトの外部結合動作は ANSI92 標準に準拠します。 No: 外部結合はサポートされません。 USUAL: デフォルトの外部結合動作は以前のバージョンのユニバースデザインツールと同じです。ユニバースデザインツールの ANSI92 パラメータが YES に設定されている場合、この動作は非優先になります。 <p>データベースによっては、他の設定も使用できます。次のデフォルトを参照してください。</p>
デフォルト	<p>ANSI_92: Oracle、MS SQL Server 2005、および Sybase のデフォルト値です。</p> <p>DB2: IBM DB2 のデフォルト値です。</p> <p>FULL_ODBC: Microsoft SQL Server のデフォルト値です。</p> <p>INFORMIX: IBM Informix のデフォルト値です。</p> <p>INGRES: Teradata のデフォルト値です。</p> <p>NO: ODBC のデフォルト値です。</p> <p>USUAL: HP Neoview、Netezza、IBM Red Brick および MS SQL Server 2000 のデフォルト値です。</p>

OUTERJOINS_GENERATION パラメータの設定例

設定 = USUAL:

```
FROM T1, T2
WHERE T1.col1(+) = T2.col2
```

設定 = DB2:

```
FROM T2 LEFT OUTER JOIN T1
ON T1.col1 = T2.col2
```

設定 = ODBC:

```
FROM {oj T1 LEFT OUTER JOIN T2 ON T1.col1=T2.col2}
Where (T2.col3 = T3.col1)
```

設定 = INFORMIX

```
FROM T2
OUTER T1
WHERE T1.col1=T2.col2
```

設定 = FULL-ODBC

```
FROM {oj T1 RIGHT OUTER JOIN T2 ON T2.col2=T1.col1
T2 INNER JOIN 3 on T2.col3 = T3.col1}
```

設定 = ANSI_92:

```
SELECT DISTINCT
  t1.col1,
  t2.col2
FROM
  (t1 RIGHT OUTER JOIN t2 ON (t1.col1=t2.col2))
```

Oracle での OUTERJOINS の使用

デフォルトの OUTERJOINS_GENERATION 設定は、ユニバースレベルでの ANSI92 パラメータの設定にかかわらず、既存のユニバースの動作に影響します。

既存の Oracle ユニバースが以前のユニバースデザインツールのバージョンと同様に動作するように設定するには、次の操作を行います。

- 1 PRM ファイルで、OUTERJOINS_GENERATION パラメータが USUAL に設定されていることを確認します。
- 2 PRM ファイルで、LEFT_OUTER パラメータおよび RIGHT_OUTER パラメータを \$(+) に設定します。

旧バージョンのユニバースデザインツールのユニバース SQL パラメータおよび PRM ファイルについては、『Designer ガイド』を参照してください。

3.4.15.14 OVER_CLAUSE

<Parameter Name="OVER_CLAUSE">YES</Parameter>

説明	SAP Business Objects アプリケーションが SQL を生成する際、RISQL 関数を含めることを許可します。データベースごとにサポートする RISQL 関数については、ANALYTIC_FUNCTIONS パラメータを参照してください。
値	YES: アプリケーションは SQL 生成時に RISQL 関数を含めることができます。 NO: アプリケーションは SQL 生成時に RISQL 関数を含めることができません。
デフォルト	YES

3.4.15.15 OWNER

<Parameter Name="OWNER">YES</Parameter>

説明	データベースで、テーブル名のプレフィックスとして所有者名をサポートするかどうかを指定します。
値	YES: データベースは、テーブル名の前への所有者名の追加をサポートします。 NO: データベースは、テーブル名の前への所有者名の追加をサポートしません。
デフォルト	YES

3.4.15.16 PREFIX_SYS_TABLE

<Parameter Name="PREFIX_SYS_TABLE">RBW_</Parameter>

<Parameter Name="PREFIX_SYS_TABLE">MSys</Parameter>

説明	ユニバースデザインツールにシステムテーブルを表示するかどうかを指定します。
値	MSys: ユニバースデザインツールのテーブルブラウザに MS Access システムテーブルを表示しません。 RBW_: Universe Designer のテーブルブラウザに IBM Red Brick システムテーブルを表示しません。 値なし: ユニバースデザインツールのテーブルブラウザにデータベースシステムテーブルを表示します。
デフォルト	MSys: MS Access のデフォルト値です。 RBW_: IBM Red Brick のデフォルト値です。

3.4.15.17 QUALIFIER

<Parameter Name="QUALIFIER">NO</Parameter>

説明	データベースで、テーブル名のプレフィックスとして修飾子をサポートするかどうかを指定します。
値	YES: データベースは、テーブル名の前への修飾子の追加をサポートします。 NO: データベースは、テーブル名の前への修飾子の追加をサポートしません。
デフォルト	RDBMS によって異なります。

3.4.15.18 QUOTE_OWNER

<Parameter Name="QUOTE_OWNER">YES</Parameter>

説明	所有者名を単一引用符で囲むかどうかを指定します。IBM Informix でのみ使用されます。
値	YES: テーブル名の先頭に、単一引用符に囲まれた所有者名が付けられます。これは ANSI 対応の IBM Informix データベースでは必須です。この設定を行わないと、IBM Informix は所有者名を大文字に変換します。 NO: テーブル名の先頭に、単一引用符に囲まれた所有者名を付けません。
デフォルト	YES
結果	SELECT Alias.col (<Alias> is a local Alias) FROM 'Owner'.table.col Alias

3.4.15.19 REFRESH_COLUMNS_TYPE

<Parameter Name="REFRESH_COLUMNS_TYPE">O</Parameter>

説明	列の更新方法を指定します。
値	O: 所有者名に従って列を更新します。 Q: 修飾子に従って列を更新します。 T: テーブル名に従って列を更新します。
デフォルト	O: Oracle のデフォルト値です。 Q: IBM Red Brick、Sybase、MS SQL Server および MS Access のデフォルト値です。

3.4.15.20 REMOVE_SEMICOLONS

<Parameter Name="REMOVE_SEMICOLONS"></Parameter>

説明	SAP BusinessObjects アプリケーションのクエリパネルに対して、フリーハンド SQL のセミコロンを削除するように命令します。
値	YES: クエリパネルはセミコロンを削除します。 NO: クエリパネルはセミコロンを削除しません。

3.4.15.21 REVERSE_TABLE_WEIGHT

<Parameter Name="REVERSE_TABLE_WEIGHT">YES</Parameter>

説明	<p>テーブルを生成する順序を指定します。このパラメータは Oracle で使用します。このパラメータは他のいくつかのデータベースでも使用されます。ただし、YES と NO が逆になる可能性があります。</p> <p>注 このパラメータは、Teradata ではサポートされていません。</p> <p>留意点 OUTERJOINS_GENERATION パラメータが ANSI92 に設定、またはユニバース ANSI92 設定が YES に設定されている場合、結果的に REVERSE_TABLE_WEIGHT パラメータは SQL 生成の最適化には影響しません。</p>
値	<p>YES: 小さい順にテーブルを生成します。</p> <p>NO: 大きい順にテーブルを生成します。</p>
デフォルト	YES

3.4.15.22 UNICODE_PATTERN

<Parameter Name="UNICODE_PATTERN">UNISTR(\$)</Parameter>

説明	<p>ユニバースの SQL 生成パラメータ UNICODE_STRINGS が YES に設定されている場合にのみ適用されます。適用される場合、文字列に基づくすべての条件は、この文字列値の書式で表現されます。このパラメータは、MS SQL Server と Oracle でのみ使用されます。</p>
値	<p>N\$: MS SQL Server の場合</p> <p>UNISTR(\$): Oracle の場合</p>

3.4.15.23 USER_INPUT_DATE_FORMAT

<Parameter Name="USER_INPUT_DATE_FORMAT">'dd-MM-yyyy HH:mm:ss'</Parameter>

説明	SQL 文の WHERE 句で生成するデフォルトの日付と時刻の書式を指定します。
値	<p>{¥d 'yyyy-mm-dd'}: ODBC のデフォルトの日付書式です。</p> <p>'DD-MM-YYYY HH:MM:SS': Oracle のデフォルトの日付と時刻の書式です。</p> <p>'MM/DD/YYYY': IBM Informix のデフォルトの日付書式です。</p> <p>'yyyy-mm-dd HH:mm:ss': MS SQL Server および大半の IBM DB2 サーバでのデフォルトの日付と時刻の書式です。</p> <p>'mm/dd/yyyy hh:m:s am/pm': Sybase のデフォルトの日付と時刻の書式です。</p> <p>'yyyy-mm-dd': Sybase ゲートウェイのデフォルトの日付書式です。</p> <p>注 ODBC で時間変数またはタイムスタンプ変数を使う場合は、odbc.sbo ファイルでデフォルトの日付書式値を {¥t 'hh:mm:ss'} または {¥t¥s 'yyyy-mm-dd hh:mm:ss'} に変更する必要があります。</p>
デフォルト	上の値を参照。

3.4.15.24 USER_INPUT_NUMERIC_SEPARATOR

<Parameter Name="USER_INPUT_NUMERIC_SEPARATOR">.</Parameter>

説明	SQL スクリプトの生成で使用するデフォルトの小数点の記号を指定します。
値	'.' (ピリオド)
デフォルト	'.'

3.4.15.25 DELIMIT_IDENTIFIERS

<Parameter Name="DELIMIT_IDENTIFIERS">YES</Parameter>

説明	データベース識別子を引用できるかどうかを指定します。識別子は、IDENTIFIER_DELIMITER パラメータで指定される区切り記号を使用して引用します。
値	YES: 識別子を引用できます。 NO: 識別子を引用できません。
デフォルト	YES
結果	Table name="my_table"

3.4.15.26 EXT_JOIN_INVERT

<Parameter Name="EXT_JOIN_INVERT">YES</Parameter>

説明	結合式での外部結合シンボルの表示方法を指定します。このパラメータは、IBM DB2、IBM Informix、Oracle、および Teradata で使用されます。
値	YES: Universe Designer の [結合の編集] ダイアログボックスで [外部結合] チェックボックスをクリックすると、結合式の外部結合シンボルが反転表示されます。 NO: Universe Designer の [結合の編集] ダイアログボックスで [外部結合] チェックボックスをクリックすると、外部結合を作成した側に外部結合シンボルが表示されます。
デフォルト	YES

3.4.15.27 KEY_INFO_SUPPORTED

<Parameter Name="KEY_INFO_SUPPORTED">YES</Parameter>

説明	データベースから 1 次キーと 2 次キーの定義を取り出せるかどうかを指定します。
値	YES: データベースから 1 次キーと 2 次キーの定義を取り出すことができます。このパラメータで Universe Designer の [構造] ウィンドウにキーを表示できます。 NO: データベースから 1 次キーと 2 次キーの定義を取り出すことはできません。
デフォルト	YES

3.4.15.28 ORDER_BY_STRINGS

<Parameter Name="ORDER_BY_STRINGS">YES</Parameter>

説明	データベースで、文字列に基づいて ORDER BY 句を正しく処理できるかどうかを指定します。このパラメータは、SAP BusinessObjects Data Federator の ORDERBYSTRINGS 機能に相当します。データベースで処理できない場合は、Data Federator クエリサーバによって並べ替えが実行されます。
値	YES: データベースで並べ替え処理を実行できます。 NO: データベースでは並べ替え処理を実行できません。

テーブルと結合を使ったスキーマの作成

4.1 概要

この章では、Web Intelligence ユーザーがレポートの作成に使用するオブジェクトを構築するために必要なすべての SQL 構造を含んだスキーマの作成方法を説明します。ここでいうSQL構造とは、テーブル、列、結合、データベース関数を指します。正確なスキーマを作成することが、エンドユーザのレポート作成ニーズに応えるためのベースとなります。

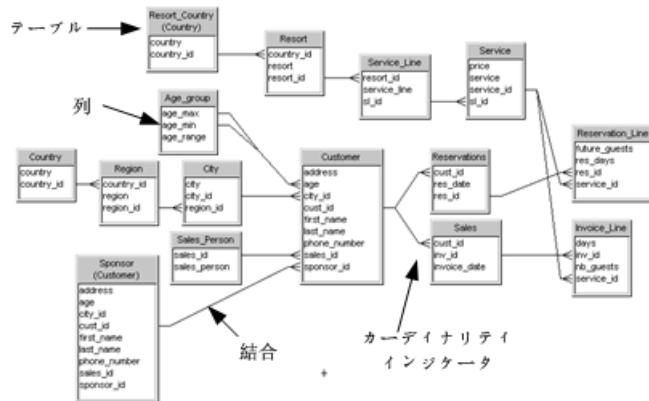
4.2 スキーマとは

スキーマとは、データベース構造をグラフィカルに表現したものです。ユニバースデザインツールで、ユニバースが表すデータベースの部分のスキーマを作成します。

スキーマにはテーブルと結合が定義されます。テーブルには、エンドユーザがレポートの作成に使用する列が含まれます。結合は、複数のテーブルに対して実行されるクエリが正しい結果を返すよう、テーブル間をリンクするものです。

スキーマをデザインするには、[テーブル参照]ウィンドウを使ってデータベースのテーブルを選択し、それを[構造]枠に挿入します。次に結合を作成して、テーブルをリンクします。ユニバーススキーマのデザインが終了したら、その整合性を自動的にチェックできます。

一例として、Beachユニバースのスキーマを次ページに紹介します。



4.2.1 スキーマのデザインこそが成功するユニバースの鍵

質の良いユニバースのデザインには、質の良いスキーマをデザインする必要があります。エンドユーザのレポート作成に必要なオブジェクトに対応する列を元に、スキーマにテーブルを挿入していきます。これらのオブジェクトは、ユーザニーズの分析結果をもとに定義したものでなければなりません。必要なオブジェクトを作成できるテーブルをデータベースから探し出します。

4.2.2 スキーマのデザインとユニバースの作成プロセス

スキーマの作成は、ユニバース開発サイクルの実装ステージの最初のステップです。ユーザニーズの分析と全体設計はユニバースデザインツールを使用せず実行できるため、スキーマの作成がユニバースデザインツールを使ったユニバース作成の最初の作業となります。

次のリストは、一般的なユニバース開発プロセスにおけるスキーマデザインの工程を明確にしたものです(実装、手順 1)。

- ・ 事前準備
 - 1 ユーザのニーズ分析
 - 2 設計
- ・ ユニバースデザインツールを使用した実装
 - 1 スキーマのデザインとテスト
 - 2 ユニバースオブジェクトの作成とテスト
 - 3 リポジトリを使用したユニバースのデプロイメント
- ・ メンテナンス
 - 1 ユーザ要件またはデータソースの変更に基づくユニバースの更新と保守

4.2.3 スキーマデザインの工程

この章では、スキーマデザインの次の工程を説明します。

- ・ テーブルの挿入と整列
- ・ 結合の作成とカーディナリティの設定
- ・ ループ、キャズムトラップ、ファントラップなどの結合の問題解決
- ・ スキーマの整合性のチェック

4.3 テーブルの挿入

スキーマデザインの最初の作業は、目的のデータベースにあるテーブルを選択し、そのテーブルを表すシンボルを[構造]枠に挿入することです。ユニバースデザインツールではこのテーブルシンボルを単にテーブルと呼びます。

テーブルを選択し、スキーマに挿入するには、[テーブル参照]ウィンドウを使います。[テーブル参照]ウィンドウは、データベースで使用できるテーブルをツリー表示する独立したウィンドウです。

注

テーブルを選択する前に、ユニバースを作成するために使用するストラテジーを指定してください。ストラテジーについての詳細は、92 ページの「[ストラテジーの選択](#)」を参照してください。

4.3.1 テーブル参照の使用

[テーブル参照]は、ターゲットデータベースのテーブルと列のツリーを表示するウィンドウです。[テーブル参照]でスキーマに挿入するデータベースのテーブルを表示し選択します。テーブル名の横にある[+]マークをクリックすると、テーブルに含まれる列を表示できます。

4.3.1.1 [テーブル参照]ウィンドウの表示

デフォルトでは、[テーブル参照]ウィンドウは非表示に設定されています。[構造]枠にテーブルを挿入するには、[テーブル参照]ウィンドウを開く必要があります。[テーブル参照]ウィンドウを開くには、次のいずれかの操作を行います。

[テーブル参照]ウィンドウを開く方法は次のとおりです。

- ・ [挿入] > [テーブル]を選択します。

または

- ・ [構造]枠の空白部分をダブルクリックします。

または

- ・ [テーブル参照]ボタンをクリックします。

[テーブル参照]ウィンドウが[構造]枠内に表示されます。

4.3.1.2 [テーブル参照]ウィンドウを使ったテーブルの挿入

[テーブル参照]ウィンドウから1つまたは複数のテーブルを挿入するには、次のいずれかの操作を行います。

1つのテーブルの挿入

テーブルを1つだけ挿入する手順は、次の通りです。

- ・ テーブルをクリックし[挿入]ボタンをクリックします。

または

- ・ テーブルを右クリックし、ポップアップメニューから[挿入]を選択します。

または

- ・ テーブルをダブルクリックします。

または

- ・ テーブルをクリックし、[構造]枠までドラッグします。

テーブルが挿入され、[構造]枠に表示されます。

複数のテーブルの挿入

複数のテーブルを挿入する

- 1 複数のテーブルをばらばらに選択するには、Ctrl キーを押しながらそれぞれのテーブルをクリックしていきます。

または

- 2 連続した複数のテーブルを選択するには、Shift キーを押しながら最初のテーブルと最後のテーブルをクリックします。

複数のテーブルが選択されました。

- 3 [挿入]ボタンをクリックします。

または

[構造]枠までドラッグします。

または

選択したテーブルを右クリックし、ポップアップメニューから[挿入]を選択します。

各テーブルが[構造]枠に表示されます。[テーブル参照]でユニバースに挿入したテーブルには、名前の横にチェックマークが付いて表示されます。

4.3.1.3 [テーブル参照]ウィンドウでのデータ表示

[テーブル参照]ウィンドウを使って、テーブルや列の値を表示することができます。

[テーブル参照]でデータを表示する方法

- 1 [テーブル参照]ウィンドウでテーブルを右クリックします。

または

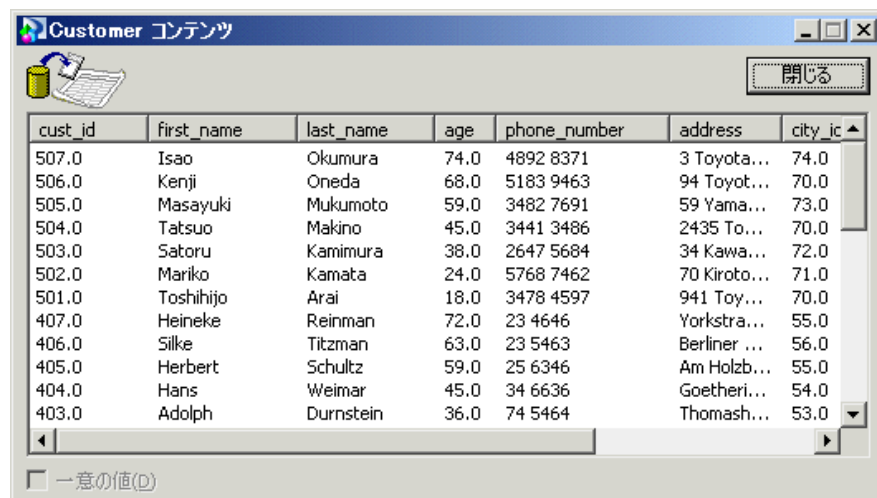
列の値を表示する場合は、テーブルを展開して列を右クリックします。

- 2 テーブルの場合はポップアップメニューから[テーブルの値を表示]を選択します。

または

列の場合はポップアップメニューから[列の値を表示]を選択します。

テーブルまたは列に含まれるデータを一覧表示するボックスが表示されます。



cust_id	first_name	last_name	age	phone_number	address	city_ic
507.0	Isao	Okumura	74.0	4892 8371	3 Toyota...	74.0
506.0	Kenji	Oneda	68.0	5183 9463	94 Toyot...	70.0
505.0	Masayuki	Mukumoto	59.0	3482 7691	59 Yama...	73.0
504.0	Tatsuo	Makino	45.0	3441 3486	2435 To...	70.0
503.0	Satoru	Kamimura	38.0	2647 5684	34 Kawa...	72.0
502.0	Mariko	Kamata	24.0	5768 7462	70 Kiroto...	71.0
501.0	Toshihijo	Arai	18.0	3478 4597	941 Toy...	70.0
407.0	Heineke	Reinman	72.0	23 4646	Yorkstra...	55.0
406.0	Silke	Titzman	63.0	23 5463	Berliner ...	56.0
405.0	Herbert	Schultz	59.0	25 6346	Am Holz...	55.0
404.0	Hans	Weimar	45.0	34 6636	Goetheri...	54.0
403.0	Adolph	Durnstein	36.0	74 5464	Thomash...	53.0

ヒント

列の幅が狭く、値がすべて表示されない場合は、Ctrl キー、Shift キー、テンキーの[+]を同時に押すと、列幅を広げることができます。

4.3.1.4 [テーブル参照]ウィンドウのパフォーマンスの最適化

[テーブル参照]ウィンドウから[構造]枠へのテーブル挿入に時間がかかる場合は、次の方法で最適化することができます。

原因	最適化の方法
データベースに多数のテーブルが存在します。ユニバースデザインツールでは、システムカタログに対するクエリが実行されるため、カタログが非常に大きいと、テーブルの取得に時間がかかることがあります。	別のデータベースアカウントを使って、挿入するテーブルだけで構成されたデータウェアハウスを構築します。作成したデータウェアハウスに対する接続を作成します。
テーブル挿入の際、自動的に結合を抽出し、カーディナリティをチェックするよう設定している。	<p>テーブルだけを挿入するようオプションを変更します。次のようにします。</p> <ol style="list-style-type: none"> [ツール] > [オプション]の順に選択します。 [オプション]ダイアログボックスが表示されます。 [データベース]タブをクリックします。 [データベース] ページが開きます。 [結合の抽出]チェックボックスおよび[結合カーディナリティの検出]チェックボックスをオフにします。 [OK]をクリックします。

4.3.2 構造枠のテーブルの整列

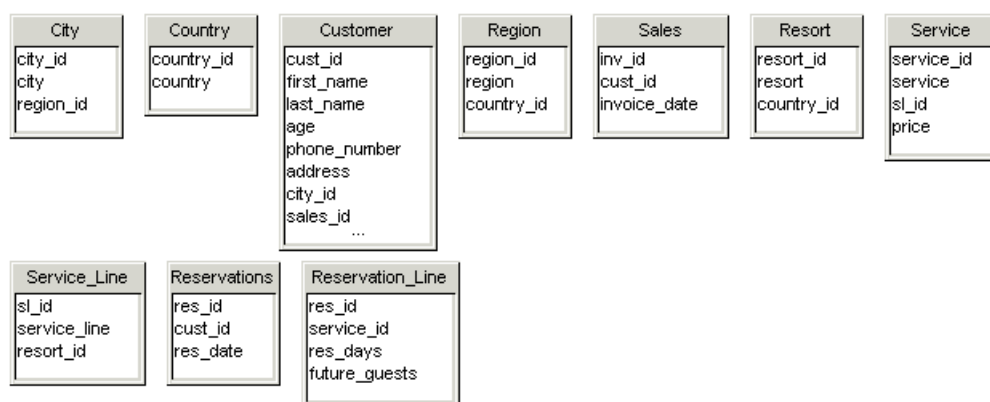
[構造]枠のテーブルを自動的に整列させ、初期スキーマを整理してから、結合作成のためにテーブルを配置し直すことができます。

4.3.2.1 構造枠のテーブルの自動整列

テーブルを自動整列する手順は、次の通りです。

- ・ [表示] > [テーブルの整列]の順に選択します。

テーブルが整列されて、表示されます。



4.4 派生テーブルの使い方

派生テーブルとは、ユニバーススキーマに定義したテーブルです。他のテーブルと同様、派生テーブルにもオブジェクトを作成します。派生テーブルはユニバースレベルの SQL クエリで定義され、ユニバースデザインツールでは論理テーブルとして使用されます。

派生テーブルには、次の利点があります。

- ・ ドキュメントに取得する分析用データの量を削減できる。

派生テーブルには、複雑な計算と関数を含めることができる。これらの操作は結果セットがドキュメントに返される前に行われるため、時間の節約となり、レポートレベルで大量のデータに複雑な分析を行う必要を減らすことができます。

- ・ データベース要約テーブルのメンテナンスを削減できる。

場合によっては、アグリゲート認識を使ってユニバースに組み込まれる複雑な計算の結果を保持する統計テーブルの代わりに派生テーブルを使用できる。集計テーブルを管理し頻繁に更新すると、費用がかかります。派生テーブルをでは、同じデータを使ってリアルタイムのデータ分析を行うことができます。

派生テーブルはデータベースのビューに似ていますが、派生テーブルの SQL にはプロンプトを含むことができるという利点があります。

4.4.1 派生テーブルの追加、編集、および削除

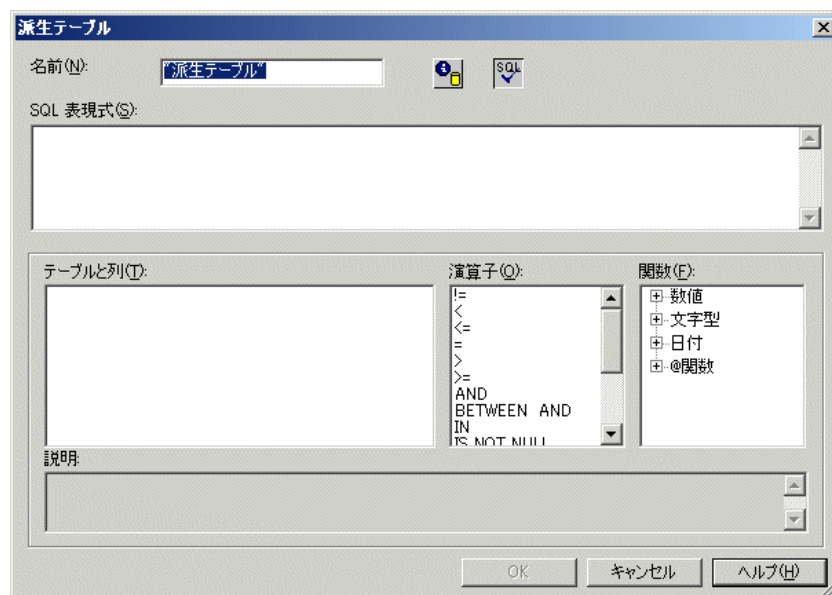
派生テーブルは、ユニバースデザインツールに通常のデータベーステーブルと全く同様に表示されますが、派生テーブルを作成するためのワークフローは異なります。派生テーブルの追加、編集、および削除について、以下のセクションで説明します。

4.4.1.1 派生テーブルの追加

派生テーブルを追加する手順は、次の通りです。

- 1 [挿入]メニューから[仮想テーブル]を選択します。

[派生テーブル]ダイアログボックスが表示されます。



- 2 [テーブル名]テキストボックスに作成するテーブルの名前を入力します。
- 3 [テーブル名]テキストボックスの下のボックスにテーブルSQLを入力します。

SQL を直接入力するか、[テーブルと列]、[演算子]、[関数]ボックスを使って SQL を作成することができます。

- 4 [OK]をクリックします。

作成した派生テーブルが、データベースの物理テーブルと一緒にスキーマに表示されます。

- 5 通常のテーブルと同様に、派生テーブルの列を使ってオブジェクトを作成します。

4.4.1.2 派生テーブルの編集

派生テーブルを編集する手順は、次のとおりです。

- 1 ユニバースデザインツール スキーマのテーブルを右クリックし、ショートカットメニューから[派生テーブルの編集]を選択します。
- 2 派生テーブルを編集し、[OK]をクリックします。

4.4.1.3 派生テーブルを削除する

- 1 スキーマ枠で、削除する派生テーブルを選択します。
- 2 Delete キーを押します。

4.4.1.4 例: 派生テーブルの作成

例 サーバ情報を返す派生テーブルの作成

この例では、ユーザがデータベースサーバの情報をレポートに追加できるオブジェクトを作成します。これには、SQL Server データベースに作成したユニバースのビルトイン変数である @@SERVERNAME と @VERSION の値を返す「サーバ名」と「バージョン」の 2 つのオブジェクトを作成します。

次の操作を行います。

- 1 [挿入]メニューから[派生テーブル]を選択します。
[派生テーブル]ダイアログボックスが表示されます。
- 2 [テーブル名]テキストボックスに[サーバ情報]と入力します。
- 3 [SQL表現式]ボックスに「Select @@SERVERNAME as サーバ名, @VERSION as バージョン」と入力します。

注

すべての派生列のエイリアスを SQL で指定する必要があります。ユニバースデザインツールでは、これらのエイリアスを使用して派生テーブルの列名が指定されます。

- 4 [OK]をクリックします。

派生テーブルのサーバ情報は、ユニバースデザインツールのスキーマに表示されます。

- 5 “サーバ情報”クラスを作成し、派生テーブル“サーバ情報”の“サーバ名”列と“バージョン”列をベースにした 2 つのディメンションオブジェクトを作成します。[サーバ情報]テーブルは、通常のデータベーステーブルと同様にテーブル一覧に表示され、その列も通常のテーブル列と同様に列一覧に表示されます。

これで、ユーザは“サーバ名”と“バージョン”オブジェクトが表示されたレポートを作成できるようになります。

例 各国の地域数を表示する

この例では、各国の地域数を示すテーブルを作成します。SQLは、次の通りです。

```
select country,
count (r.region_id) as number_of_regions
from country c,
region r
where r.country_id = c.country_id
group by country
```

この場合、計算を含む列のエイリアスを指定することが重要です。ユニバースデザインツールでは、これらのエイリアスが派生テーブルの列名として使用されます。派生テーブルには、country と number_of_regions の 2 つの列が作成されます。

4.5 ネストされた派生テーブル

ネストされた派生テーブル（「派生テーブルの派生テーブル」とも呼ばれています）は、最低 1 つの既存の派生テーブルから派生したテーブルです。またネストされた派生テーブルも、データベーステーブルを参照することができます。

ネストされた派生テーブルを作成するには、派生テーブル エディタを使用して SQL 式を入力し、派生テーブル(必要な場合、データベース内の物理テーブル)を選択します。派生テーブルの SQL 式は、レポートの生成時にネストされた派生テーブルの SQL 式に挿入されます。

4.5.1 派生テーブルエディタの使い方

派生テーブルまたはネストされた派生テーブルを定義するには、派生テーブル エディタを使用 します。エディタに SQL 式を入力してオブジェクト(テーブル、派生テーブル、列、関数)をダブルクリックすると、派生テーブルまたはネストされた派生テーブルの SQL 式が作成されます。

ネストされた派生テーブル内の派生テーブルを参照するには、@DerivedTable 関数を使用します。

- ・ 派生テーブルエディタでは、関数カタログに追加された @DerivedTable(Derived_table_name) 関数を使用できます。
- ・ 派生テーブルエディタの下部の中央の枠には、既存の派生テーブルおよびネストされた派生テーブルが表示されます。この枠は、ユニバースに派生テーブルが存在する場合にのみ表示されます。

[整合性のチェック]をクリックすると、派生テーブルとネストされた派生テーブルの両方に対して次のチェックが実行されます。

- ・ 参照されている派生テーブルが削除された場合の派生テーブルへの影響の検出
- ・ 循環参照のチェック
- ・ 許可されていないオブジェクト定義(@DerivedTable() および SELECT)内 の WHERE のチェック

4.5.2 ネストされた派生テーブルを作成する

派生テーブルの作成と同じ方法で、ネストされた派生テーブルを作成します。派生テーブルの追加および名前変更と同じ方法で、ネストされた派生テーブルの追加および名前変更を実行します。

ネストされた派生テーブルを作成する:

- 1 サンプルディレクトリ(Business Objects¥BusinessObjects Enterprise 12¥Samples¥en¥UniverseSamples)にあるユニバース(*.unv)を開きます。
- 2 [ユニバース構造]枠で右クリックして、コンテキストメニューの[派生テーブル]を選択します。
派生テーブル エディタが開いて、派生テーブル エディタの下部にある中央の枠に、使用可能な派生テーブルの一覧が表示されます。
- 3 ネストされた派生テーブルの名前を入力します。
- 4 SQL 式を入力します。テキスト全体を入力するか、エディタの入力補助機能を利用できます。
- 5 オブジェクト(テーブル、派生テーブル、列、関数)をダブルクリックします。
- 6 構文では @DerivedTable 関数を @DerivedTable(Derived_table_name) の形式で使用して、派生テーブルを選択します。
- 7 [構文のチェック]をクリックして派生テーブルの構文をチェックし、潜在的なエラーを修正したら、ネストされた派生テーブルを確認します。
ネストされた派生テーブルはユニバースに追加されます。
- 8 [OK]をクリックして、ネストされた派生テーブルを確認します。
ネストされた派生テーブルが[構造]枠に表示されます。派生テーブルとネストされた派生テーブルは、実際のデータベーステーブルを表すテーブルよりも薄い色になっています。

注

テーブル値を表示するには、異なるテーブルを右クリックします。

ユニバース内に、ネストされた派生テーブルを作成しました。

4.5.3 ネストされた派生テーブルの名前の変更

派生テーブルの名前を変更した場合、新しい名前は、そのテーブルを参照する他のすべての派生テーブルに反映され、更新されます。

4.6 入力列を含むテーブルの使い方

入力列を含むテーブルがユニバースに挿入されると、Web Intelligence ユーザーまたは Query as a Web Service ユーザーは、入力列を計算するために値を選択するか、入力する必要があります。入力列には、値の制限があります。入力列のデータは、必ずしも元のデータベースで使用できるわけではありません。データは、次のいずれかです。

- ・ ユニバースの作成時にハードコーディングされた値
- ・ プロンプト後にエンドユーザが入力したか、一覧から選択した値
- ・ 別のテーブルとの結合によって提供された値

入力列を含むテーブルは、データベース接続が Business Objects Data Federator サーバの場合のみサポートされます。

入力列の解決に適した結合を分析するときは、以下の点に注意します。

- ・ アルゴリズムの解決では、単純な結合のみを考慮する
- ・ 複数の左列または複数の右列を含む結合など複雑な結合は許可しない
- ・ equal オペランドまたは IN(NLIST)オペランドを含む結合のみを考慮するBetweenなどの演算子は、入力列の解決に使用できない

入力列を含むテーブルでは、[構造]枠の入力列の右側に矢印が表示されます。[テーブル参照]枠では、入力列が特定のアイコンで識別されます。

入力列を含むテーブルを挿入する場合は、入力列エディタを使用して設定を入力します。

注

ユニバースに追加するときは、テーブルの入力列のデフォルト値を割り当てる必要があります。

この機能は、次の製品およびコンポーネントで使用できます。

- ・ Web Intelligence
- ・ Query as a Web Service

関連項目

- ・ 153 ページの[ハードコーディングされた値の一覧を定義する](#)
- ・ 153 ページの[ユーザが入力または選択できる値の一覧を定義する](#)

4.6.1 ハードコーディングされた値の一覧を定義する

データベースには、1 つ以上の入力列のあるテーブルが少なくとも 1 つ含まれています。

ハードコーディングされた値の一覧は、テーブルの入力値として使用され、入力列の値を決定します。エンドユーザは、値を入力しません。値の一覧を定義するには、以下の手順に従ってください。

- 1 データベースからテーブルを選択し、ユニバースデザインツールの[構造]枠に追加します。
入力列エディタが表示されます。
- 2 入力列エディタでパラメータをクリックします。
- 3 [値]フィールドに、値または値の一覧を入力します。各値を二重引用符で囲み、セミコロン(;)で区切ります。
[値]列に値が表示されます。
- 4 [次回実行時]の一覧で、[この値を使用]を選択します。
[次回実行時]列に[この値を使用]が表示されます。
- 5 [OK]をクリックします。

ユニバースデザインツールの[構造]枠にテーブルが表示されます。入力列は矢印で識別されます。

4.6.2 ユーザが入力または選択できる値の一覧を定義する

データベースには、1 つ以上の入力列のあるテーブルが少なくとも 1 つ含まれています。

ユーザは、値を入力するか、テーブルで使用される値の一覧から値を選択して、入力列の値を決定することができます。スキーマで入力列テーブルの値を定義するには、次の手順に従ってください。

- 1 データベースからテーブルを選択し、ユニバースデザインツールの[構造]枠に追加します。
- 2 入力列エディタでパラメータをクリックします。
- 3 [次回実行時]の一覧で[別の値を入力]をクリックします。
Web Intelligence クエリーまたは Query as a Web Service クエリーの実行時に、ユーザーは関連付けられている値の一覧から値を選択するように求められます。
- 4 [プロンプト ラベルの編集]フィールドで、エンド ユーザーに表示されるデフォルトのプロンプトを編集します。
- 5 [ユニバースオブジェクトの参照]をクリックしてユニバースから値の一覧を選択します。
設定に追加した値の一覧からオブジェクトを削除する場合は、[選択されたオブジェクト]枠でオブジェクトをクリックし、[消去]をクリックします。
- 6 [OK]をクリックします。

ユニバースデザインツールの[構造]枠にテーブルが表示されます。入力列は矢印で識別されます。[テーブル参照]では、入力列が特定のアイコンで識別されます。

4.7 結合の定義

スキーマに複数のテーブルを挿入すると、リレーションにもとづいて結合を作成する必要があります。結合は複数のテーブルを組み合わせ、意味のあるデータを取り出す手段ですので、スキーマ内ではテーブルと同じく重要です。

4.7.1 結合とは

結合とは、関連する異なるテーブル間のデータをリンクする状態を意味します。通常、テーブルには親子関係があります。クエリに結合が含まれていない場合、データベースからクエリテーブルの行のすべての組み合わせを提示する結果セットが返されます。このようなクエリ結果はデカルト積と呼ばれ、役に立つデータとなることはほとんどありません。

たとえば、それぞれ 100 行と 50 行のデータを持つ 2 つのテーブルにクエリを実行すると、5000 行のデカルト積が返されます。より多くのテーブルを対象にしたクエリやデータ量の多いデータベースでは、デカルト積の結果はすぐに膨大なものとなります。ユニバースデザインツールでは、結合は、スキーマでテーブルを接続する線として表されます。

4.7.2 スキーマ内で結合を使用する意味

複数のテーブルに対するクエリが正しい結果を返すには、結合が必要となります。テーブル間の結合は、クエリに両方のテーブルが含まれた場合に、どのようにデータを取得するかを定義します。

スキーマのそれぞれのテーブルには、ユーザのニーズに対応した複数の列が含まれています。業務用ユニバースでは、Web Intelligence ユーザがあらゆるテーブルの組み合わせからデータを取得するように、さまざまなオブジェクト (列に対応) を組み合わせてクエリを実行できます。

スキーマ内のすべてのテーブルを結合を使ってリンクしておくことで、複数のテーブルの列データをクエリ内で組み合わせる方法を制限します。結合はテーブル間での列の組み合わせを共通列あるいは一致列に制限します。これにより、クエリ結果に意味のない情報が含まれることを防ぎます。

注

結合は、常に構造枠で作成します。オブジェクトに Where 句で手動で定義された結合など、構造枠から作成されなかった結合は、実行時に作成されるため、ユニバースデザインツールでは、整合性のチェックやコンテキスト検出の対象になりません。整合性のチェックとコンテキストの検出は、ユニバースのデザインに欠かせない作業です。詳しい説明は、この章の後半で説明します。

4.7.3 SQL内での結合

ユニバースデザインツールでは、デフォルトで、WHERE 句内でテーブル間の共通列または一致列を参照することで、結合を暗黙的に表現します。

通常は、2 つのテーブル間の結合は 1 つの WHERE 句で表現されます。従って、4 つのテーブルを組み合わせるには、少なくとも 3 つの条件が必要となります。

結合でリンクされた2つのテーブルを含んだクエリの結果は、組み合わせたテーブルのすべての列で構成される単一テーブルになります。このテーブルの各行には、共通列の値が一致するそれぞれのテーブルの行データが含まれます。

4.7.3.1 ANSI 92 サポート

ターゲットの RDBMS で ANSI 92 がサポートされる場合、ユニバースのパラメータ ANSI92 ([ファイル] > [パラメータ] > [パラメータ]) を[Yes]に設定して、スキーマで作成した結合で ANSI 92 のサポートを有効にすることができます。ユニバースで結合の ANSI 92 標準がサポートされる場合、新しく作成される結合は FROM 句に指定されます。また、列により結合されるオブジェクトを選択して FROM 句に含めることもできます。ANSI 92 のサポートは、169 ページの「[ユニバースの結合の ANSI 92 サポート](#)」を参照してください。

4.7.4 結合する必要がないテーブル

Web Intelligence でユニバースに対して実行するクエリに含まれるオブジェクトが生成する SQL の対象となるテーブルは、すべてスキーマで結合する必要があります。ただし、次の種類のテーブルだけは結合する必要はありません。

- それぞれの用途に応じて作成するエイリアスのベースとなるテーブルテーブル名や結合の問題の解決のため作成するエイリアスの元となるテーブルです。テーブルは通常、オブジェクトの定義では使用されません。
- アグリゲート認識の対象となるテーブル (ケースバイケース)たとえば、サンプルである eFashion ユニバースの集計テーブル (名前は "Agg_" で始まります) は、以下のとおり、スキーマ内のどのテーブルとも結合していません。

4.7.5 プライマリキーと外部キーの結合

結合は、通常、1つのテーブルのプライマリキーと別のテーブルのフォーリンキーの間に作成します。2つの一次キー間でも結合を作成できます。結合にプライマリキーが含まれていないことは、非常にまれなケースです。

それぞれのキーが、データベースでどのように設定されているかをよく理解しておく必要があります。マルチカラムキーの場合は、結合のカーディナリティの設定が影響を受け、それによりスキーマ内のコンテキストの設定方法も変わってきます。

コンテキストの検出と使い方の詳細については、205 ページの「[結合問題の検出と解決](#)」を参照してください。

4.7.5.1 キーの表示

構造枠に、すべてのテーブルのプライマリおよびフォーリンキーを表示することができます。キー列は、キーを含む各テーブルで下線表示されます。キーを表示するオプションを選択する場合、構造を最新表示するとキーが下線表示されます。

キー列を下線表示できるかどうかは、ターゲットデータベースで定義された一次キーによって異なります。

注

キー列を下線表示できる場合、情報は .UNV ファイルに保存されています。ユニバースを Central Management Server(CMS)リポジトリにエクスポートすると、この情報は失われます。ユニバースをインポートするたびに、キーを再表示する必要があります。

キーを表示する手順は、次の通りです。

- 1 [ツール]>[オプション]の順に選択します。

[オプション] ダイアログボックスの [一般設定] ページが表示されます。

- 2 [グラフィック]タブをクリックします。

[グラフィック] ページが表示されます。

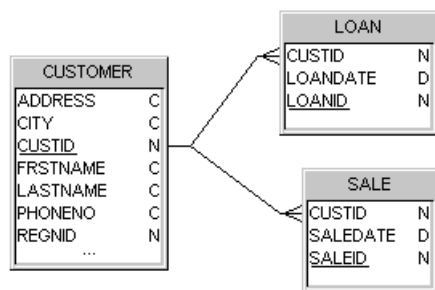
- 3 [列] ボックスで、[下線付きキー] オプションを選択します。

- 4 [OK]をクリックします。

キー列を下線表示するには、構造を最新表示する必要があります。

- 5 [表示] > [構造の最新表示]を選択します。

データベース構造が最新表示されます。スキーマ内のキー列に下線が付けられます。



4.7.6 結合のカーディナリティについて

カーディナリティとは、2つのテーブル間の結合を一致する行の数という観点から表現したものです。カーディナリティは結合の問題を検出し、コンテキストを作成してRDBMS構造の制限を補正するのに非常に重要となります。

スキーマで結合ごとにカーディナリティを設定してください。ユニバースデザインツールではカーディナリティが自動的に検出および設定されますが、結合されるキーの性質を考慮して、常に手動でカーディナリティをチェックしてください。

カーディナリティの設定と使い方については、187 ページの「[カーディナリティの使い方](#)」を参照してください。

4.7.7 結合の作成

ユニバースデザインツールでは、以下の方法で結合を作成できます。

- ・ スキーマ内で結合を手動でトレースする。
- ・ 結合のプロパティを直接定義する。
- ・ 自動検出された結合を選択する。
- ・ テーブル挿入時に結合を自動作成する。

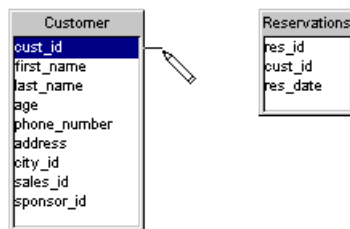
ここでは、それぞれの方法について説明します。

4.7.7.1 スキーマ内での結合の手動トレース

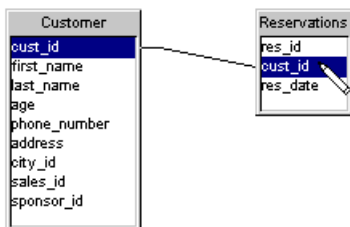
1つのテーブルの列から別のテーブルの対応する列にマウスで線を引いて直接個別に結合を作成することができます。

手動トレースによる結合の作成方法は、次の通りです。

- 1 結合するどちらかの列の上にマウスポインタを置きます。
ポインタが手の形になります。
- 2 マウスの左ボタンを押したままにします。
列が反転表示されます。
- 3 マウスを他のテーブルの結合先の列までドラッグします。
ドラッグ中は、マウスポインタが鉛筆の形で表示されます。



- 4 結合先の列の上に鉛筆ポインタを置きます。
結合先の列が反転表示されます。



- 5 マウスボタンを放します。
2つのテーブル間で結合が作成されました。
- 6 作成した結合をダブルクリックします。
[結合の編集]ダイアログボックスが開きます。結合のプロパティが表示されています。カーディナリティや結合の種類などの結合プロパティの設定については、162ページの「[結合のプロパティ](#)」を参照してください。
- 7 結合のプロパティを入力または選択します。
- 8 [OK]をクリックします。

4.7.7.2 結合プロパティの直接入力

[結合の編集]ダイアログボックスで、結合のプロパティを直接定義して結合を作成することができます。

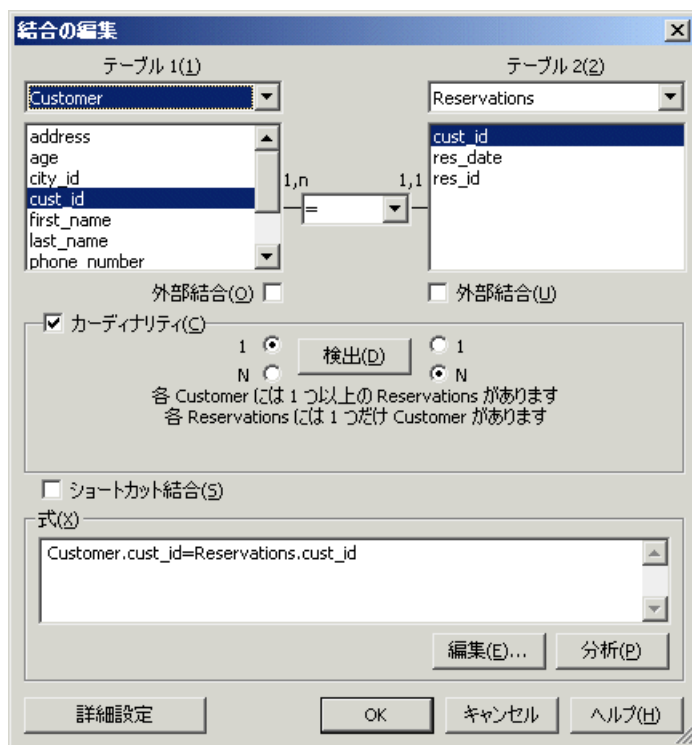
結合を直接定義する手順は、次の通りです。

- 1 [挿入]>[結合]の順に選択します。

または

[結合の挿入]ボタンをクリックします。

[結合の編集]ダイアログボックスが開きます。



- 2 [テーブル1]ドロップダウンリストからテーブルを選択します。

テーブル名の下ボックスに、選択したテーブルの列が表示されます。

- 3 新しい結合を作成する列の名前をクリックします。

- 4 [テーブル2]ドロップダウンリストからテーブルを選択します。

テーブル名の下ボックスに、選択したテーブルの列が表示されます。

- 5 新しい結合を作成するもう一方の列の名前をクリックします。

結合演算子、カーディナリティ、結合の種類など、結合に設定できるプロパティについては、162 ページの「[結合のプロパティ](#)」を参照してください。

6 結合のプロパティを入力または選択します。

7 [OK]をクリックします。

[結合の編集] ダイアログボックスで定義した 2 つのテーブルをリンクするスキーマに、新しい結合が表示されます。

4.7.7.3 自動検出された結合の選択

ユニバースデザインツールの[結合の検出]機能を使用すると、スキーマで選択した結合を自動的に検出できます。ユニバースデザインツールによって、ターゲットデータベース内の複数のテーブルにわたって列名が識別され、スキーマにテーブルの結合候補が表示されます。使用する候補を選択するか、すべての候補を挿入し、結合を作成できます。

結合の自動検出

結合は、[ユニバースのパラメータ]ダイアログボックスの[ストラテジー]ページ([ファイル]>[パラメータ]>[ストラテジー]の順に選択)で設定した結合ストラテジーをもとに検出されます。

ストラテジーは、データベースから構造情報を自動的に抽出するスクリプトファイルです。ユニバースデザインツールにはさまざまなビルトインストラテジーが用意されています。ストラテジーは、ダイアログボックスの [ストラテジー] ページのドロップダウンリストに一覧表示されています。

デフォルトの結合の自動検出ストラテジーは、キー情報を除外した列名のマッチングになっています。結合を自動検出に適用するストラテジーを、このドロップダウンリストから選択できます。

注

ストラテジーの使い方の詳細については、92 ページの「[ストラテジーの選択](#)」を参照してください。

結合の自動検出の適切な使い方

自動的に結合を検出することで、スキーマ内に簡単に結合を作成できます。ただし、スキーマをデザインする際は自動検出の限界についても認識しておく必要があります。

結合候補の検出に使用する結合ストラテジーでは、データベースの列名の一致を検査します。ターゲットのデータベースには、他のテーブルにプライマリー、フォーリンキー、その他の結合列と同じ名前が含まれていない場合もあります。ユニバースデザインツールでは、これらの列は無視されます。そのため、作成する各結合が自動検出されたかどうかを必ず手動で確認する必要があります。自動検出されなかった他の結合が必要な場合もあることに留意してください。

自動検出を使用して結合を作成する

1 [ユニバースのパラメータ] ダイアログボックスの [結合] ドロップダウンリストで、結合を検出する結合ストラテジーが選択されていることを確認してください。確認の手順は、次の通りです。

- ・ [ファイル]>[パラメータ]の順に選択し、[ストラテジー]タブをクリックします。
- ・ [結合] のドロップダウンリストから結合の検出に使用するストラテジーを選択し、[OK] をクリックします。

- 2 構造枠で複数のテーブルを選択します。

複数のテーブルは、[Shift] キーを押しながら各テーブルをクリックするか、または画面の空白部をクリックし、左マウスクリックした状態でマウスをドラッグしてカーソルで描く長方形で希望の複数のテーブルを囲みます。

- 3 [ツール]>[自動検出]>[結合の検出]の順に選択します。

または

[結合の検出]ボタンをクリックします。

[結合の候補] ダイアログボックスが表示されます。このダイアログボックスには、構造枠で選択したテーブル間に検出されたすべての結合の候補が表示されます。結合の候補は、構造枠で選択されたテーブル間の青い線による結合でも表示されています。



- 4 [挿入]をクリックするとすべての結合候補が作成されます。

- 5 または

特定の結合だけを作成するには、作成する結合候補を選択してから[挿入]をクリックします。

複数の結合候補を選択するには、Ctrlキーを押しながら結合をクリックします。連続した複数の結合候補を選択するには、Shiftキーを押しながら最初の結合と最後の結合をクリックします。

スキーマ内に結合が作成されます。

- 6 [閉じる]をクリックします。

4.7.7.4 関連テーブル間の結合の自動挿入

結合を使用するテーブルを構造枠に挿入するときに、結合も同時にスキーマに自動挿入することができます。結合の自動作成は、次の 2 つのプロセスで決定されます。

- ・ アクティブな結合戦略で、結合を検出する列情報を決定します。

- ・ 関連するテーブル間で結合を自動作成するには、デフォルトの作成オプション、[結合の抽出] が選択されている必要があります。このオプションは、[オプション] ダイアログボックスの [データベース] ページにあります。

結合を自動挿入するときの制約

結合の自動挿入を使用するとスキーマに結合を簡単に挿入できますが、スキーマのデザインに致命的な欠点をもたらす恐れもあります。結合はデータベース構造に基づいて挿入されるので、データベースで名前変更された複数のテーブルに共通な列は結合挿入時には考慮に入れられません。

そのため、製造ユニバースで結合を作成するときには、結合を自動挿入しないようにしてください。デモやユニバースを素早く構築する時にのみ、この方法を使用し、結合の挿入後、必ず各結合を注意深く確認してください。

テーブルと同時に結合を自動挿入する手順は、次の通りです。

- 1 [ユニバースのパラメータ] ダイアログボックスの [ストラテジー] ページで、結合を検出する結合ストラテジーが選択されていることを確認してください。
- 2 [ツール]>[オプション]の順に選択します。
[オプション]ダイアログボックスが開きます。
- 3 [データベース]タブをクリックします。
[データベース] ページが開きます。
- 4 [結合の抽出]オプションをチェックします。
- 5 [OK]をクリックします。

これで、構造枠にすでに挿入されているテーブルの他の列を参照する列をテーブルに挿入するとき、テーブル間の参照が関連テーブル間の結合として自動挿入されるようになりました。

4.7.8 結合のプロパティ

[結合の編集] ダイアログボックスで、結合のプロパティを定義します。次のプロパティを結合に定義できます。

プロパティ	説明
テーブル 1	結合の左端のテーブル選択したテーブルの列は、ドロップダウンリストボックスに表示されます。
テーブル 2	結合の右側のテーブル選択したテーブルの列は、ドロップダウンリストボックスに表示されます。

プロパティ	説明
演算子	テーブルの結合に使用する演算子: 結合に使用できる演算子については、163 ページの「 結合演算子 」を参照してください。
外部結合	このオプションを選択すると、外部結合を作成するときに、マッチしないデータを含むテーブルを決定します。外部結合についての説明は、180 ページの「 外部結合 」を参照してください。
カーディナリティ	このオプションを選択すると、結合のカーディナリティを定義できます。カーディナリティの定義と使い方については、187 ページの「 カーディナリティの使い方 」を参照してください。
ショートカット結合	結合をショートカット結合として定義します。ショートカット結合についての説明は、183 ページの「 ショートカット結合 」を参照してください。
式	クエリに両方のテーブルが含まれる場合、取得するデータを制限するために使用する Where 句
詳細	<p>ANSI 92 サポートがユニバースで有効になっている場合利用可能です。クリックすると、2 番目の結合プロパティボックスが表示されます。これは、結合にある 2 つのテーブルの列で作成されたオブジェクトを一覧表示します。オブジェクトを選択して FROM 句に含めることができます。</p> <p>結合構文の ANSI 92 サポートの有効化についての詳細は、169 ページの「ユニバースの結合の ANSI 92 サポート」を参照してください。</p>

4.7.8.1 結合演算子

テーブル 1 とテーブル 2 の間にあるボックスのドロップダウンリストで、結合に使用する演算子を選択します。指定した演算子にしたがって、結合列のデータの一致に使用する制限が変わります。

結合には次の演算子を設定できます。

演算子	説明
=	が次の値に等しい
!=	が次の値に等しくない
>	が次の値より大きい
<	が次の値より小さい
>=	が次の値以上 (\geq)
<=	が次の値以下 (\leq)
Between	が値の範囲内 (シータ結合)
Complex	複合関係

4.7.8.2 編集と分析

[結合の編集]ダイアログボックスには、結合のシンタックスを編集および確認するための機能が 2 つあります。

編集

[編集]ボタンをクリックすると、[SQLエディタ]が開きます。このグラフィックエディタを使用して、結合で使用するテーブル、列、演算子、関数のシンタックスを変更します。SQL エディタの使い方については、166 ページの「[結合 SQL エディタの使い方](#)」を参照してください。

分析

[分析]ボタンをクリックすると、結合式のSQL構文を分析できます。分析が成功すると、OK メッセージが表示されます。ユニバースデザインツールでエラーが発生すると問題の原因を指摘するメッセージが表示されます。

4.7.9 結合の編集

結合を編集する方法は、いくつかあります。

- ・ [結合の編集]ダイアログボックスで結合のプロパティを変更する。
- ・ 結合の SQL エディタを使用して、SQL シンタックスを直接編集する。
- ・ 数式バーを使用して、SQL シンタックスを直接編集する。

次に、それぞれの編集方法を説明します。

4.7.9.1 [結合の編集]ダイアログボックス

[結合の編集] ダイアログボックスは、結合プロパティの定義と編集に使用します。このダイアログボックスから SQLエディタを開いて、結合構文を編集することもできます。結合のプロパティについては、162 ページの「[結合のプロパティ](#)」を参照してください。

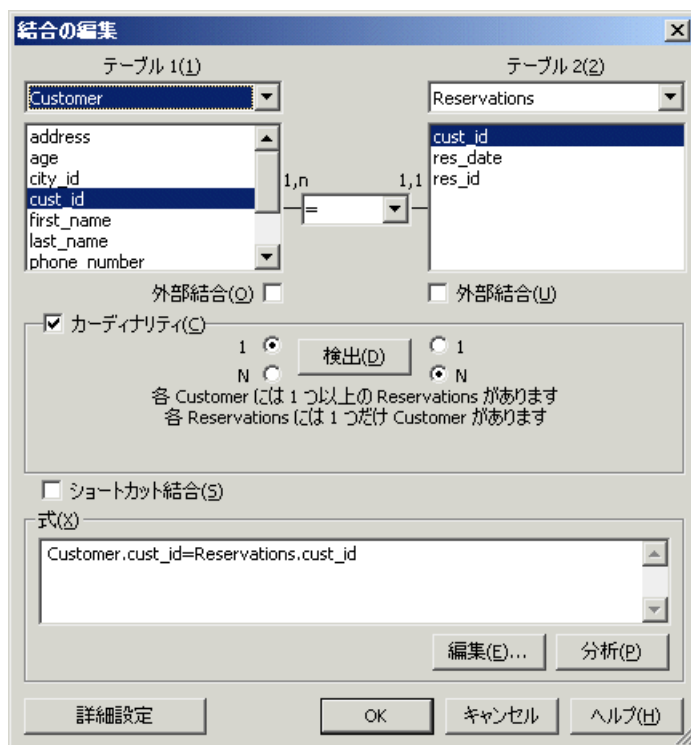
[結合の編集]ダイアログボックスを使って結合を編集する手順は、次の通りです。

- 1 構造枠で結合をダブルクリックします。

または

結合をクリックして[編集]>[結合]の順に選択します。

[結合の編集]ダイアログボックスが開きます。



- 2 テーブル間のドロップダウンリストから演算子を選択します。
- 3 必要に応じて、他のプロパティも選択します。
- 4 結合を ANSI 92 構文で定義する場合は、[詳細]ボタンをクリックします。
- 5 [OK]をクリックします。

ヒント

[編集]ボタンをクリックすると、SQLエディタで結合に使用されるSQLを直接編集できます。詳細については、166 ページの [「結合 SQL エディタの使い方」](#)を参照してください。

4.7.9.2 結合 SQL エディタの使い方

グラフィックエディタを使用して、結合の SQL 式を直接変更できます。このエディタには、[結合の編集] ダイアログボックスからアクセスします。

SQLエディタを使って結合を編集する手順は、次の通りです。

- 1 構造枠で結合をダブルクリックします。

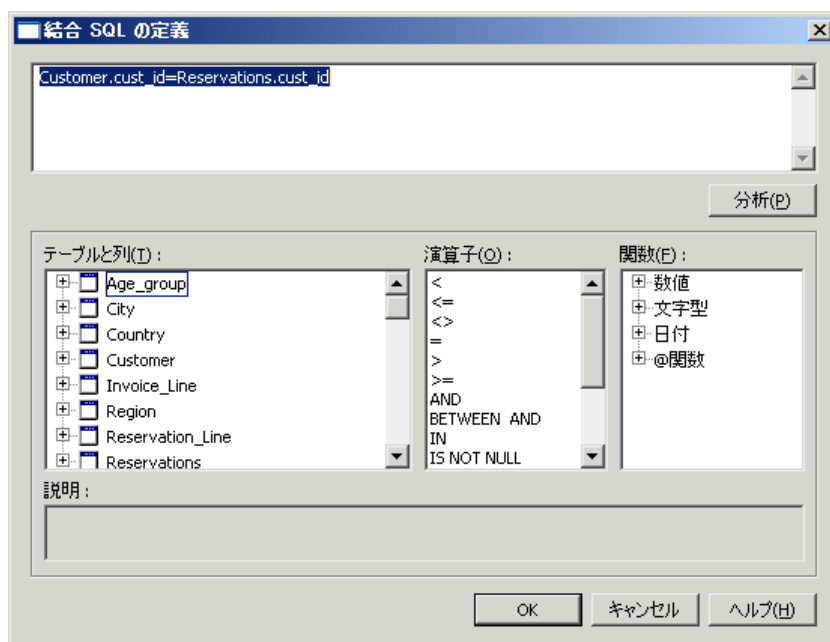
または

結合をクリックして[編集]>[結合]の順に選択します。

[結合の編集]ダイアログボックスが開きます。

- 2 [編集]ボタンをクリックします。

[結合 SQL の定義] ダイアログボックスが表示されます。結合の SQL 式がテキストボックスに表示されます。



- 3 結合式で SQL 構文を追加または編集する部分をクリックします。

SQL構文の変更、追加に使用できる編集機能は、次の通りです。




編集内容	操作
結合列の変更	<ul style="list-style-type: none"> ・ [テーブルと列]ボックスのテーブルを展開します。 ・ 列をダブルクリックします。
結合演算子の変更	[演算子]ボックスの演算子をダブルクリックします。
関数の適用	<ul style="list-style-type: none"> ・ [関数]ボックスの関数カテゴリを展開します。 ・ 使用する関数をダブルクリックします。

列、演算子、関数が結合の定義ボックスに表示されます。

- 4 [OK]をクリックします。

4.7.9.3 数式バーの使い方

数式バーは、[ユニバース]ウィンドウの上に表示されるテキストボックスで、ここには[構造]枠で選択した結合や[ユニバース]枠で選択したオブジェクトを定義する式が表示されます。数式バーの左には、次の3つの編集ボタンがあります。

編集ボタン	説明
	確定前の最後の変更内容をキャンセルします。表現式に複数の変更を加えた場合、確定前であれば、[キャンセル]ボタンで最初の状態に戻ります。変更作業を一つずつ取り消すには、[編集]>[元に戻す]の順に選択するか、[元に戻す]をクリックします。
	変更内容を確定します。これにより、結合式に加えた変更が有効となります。確定した変更を元に戻すには、[編集]>[元に戻す]の順に選択するか、[元に戻す]をクリックします。
	選択した結合の[結合の編集]ダイアログボックスを開きます。

数式バーを表示する手順は、次のとおりです。

- ・ [表示]>[数式バー]の順に選択します。

数式バーが[ユニバース]ウィンドウの上に表示されます。

数式バーを使って結合を編集する手順は、次のとおりです。

- 1 編集する結合をクリックします。
数式バーにその結合の表現式が表示されます。
- 2 数式バーに表示される結合式の変更箇所をクリックします。
- 3 必要に応じて式を変更します。
- 4 [適用]をクリックして、変更を適用します。
- 5 Return キーを押して数式バーを終了します。

または

数式バー以外の任意の場所をクリックします。

4.7.10 ユニバースの結合の ANSI 92 サポート

ユニバースデザインツールでは、結合に ANSI 92 構文がサポートされています。デフォルトで、ANSI 92 はサポートされません。サポートを有効にするには、SQL ユニバースパラメータ ANSI92 を[Yes]に設定する必要があります。このパラメータは、ユニバースパラメータダイアログボックス ([ファイル] > [パラメータ] > [パラメータ]) の [パラメータ] ページに表示されます。有効にすると、ユニバースの結合で ANSI 92 構文を使用することができます。

注

ANSI 92 の設定は、.prm ファイルにも宣言されます。.prm の設定が "usual" の場合、ユニバースデザインツールの設定が優先されます。.prm の設定が "ANSI92" の場合、ユニバースデザインツールレベルの設定が上書きされます。.prm ファイルと ANSI 92 の設定の詳細については、データアクセスガイドを参照してください。動作は、データベースのバージョンによって異なります。詳細については、お使いのデータベースの技術情報を参照してください。

結合でこの構文を使用する前に、ターゲットの RDBMS で ANSI 92 がサポートされていることを確認するようにしてください。

以下で、ユニバースでの ANSI 92 サポートの有効化、および ANSI 92 構文を使用した結合の定義について説明します。

4.7.10.1 例: デフォルトの結合構文と ANSI 92 構文の比較

以下は、2 つの結合の結合構文です。まず、WHERE 句で結合を定義した場合のデフォルトの動作を示します。次に、同じ結合を ANSI 92 標準を使用して FROM 句に含めた場合を示します。

デフォルトの結合構文

```
SELECT
  Resort.resort,
  'FY'+Format(Sales.invoice_date,'YYYY'),
  sum(Invoice_Line.days * Invoice_Line.nb_guests * Service.price)
FROM
  Resort,
  Sales,
  Invoice_Line,
  Service,
  Service_Line
WHERE
  ( Sales.inv_id=Invoice_Line.inv_id )
  AND ( Invoice_Line.service_id=Service.service_id )
  AND ( Resort.resort_id=Service_Line.resort_id )
  AND ( Service.sl_id=Service_Line.sl_id )
GROUP BY
  Resort.resort,
  'FY'+Format(Sales.invoice_date,'YYYY')
```

同じ結合で ANSI 92 標準を使用した場合

```
SELECT
  Resort.resort,
  'FY'+Format(Sales.invoice_date,'YYYY'),
```

```

sum(Invoice_Line.days * Invoice_Line.nb_guests * Service.price)
FROM
Resort INNER JOIN Service_Line ON (Resort.resort_id=Service_Line.resort_id)
INNER JOIN Service ON (Service.sl_id=Service_Line.sl_id)
INNER JOIN Invoice_Line ON (Invoice_Line.service_id=Service.service_id)
INNER JOIN Sales ON (Sales.inv_id=Invoice_Line.inv_id)

GROUP BY
Resort.resort,
'FY'+Format(Sales.invoice_date,'YYYY')

```

4.7.10.2 ユニバースでの ANSI 92 サポートの有効化

結合で ANSI 92 のサポートを有効化する手順は、次の通りです。

- 1 [ファイル]>[パラメータ]を選択します。

[ユニバースのパラメータ]ダイアログボックスが表示されます。

- 2 [パラメータ]タブをクリックします。

[パラメータ]ページが表示されます。ここで表示される SQL 生成パラメータをユニバースのレベルで設定すると、現在のユニバースでの SQL 生成を最適化することができます。これらのパラメータは、Business Objects 製品の旧バージョンで、ターゲットの RDBMS の PRM ファイルに含まれていたものです。特定の RDBMS 固有のパラメータはまだ PRM ファイルに含まれていますが、標準 SQL パラメータの多くは [パラメータ] ページに一覧表示されます。利用可能なパラメータの完全な一覧は、101 ページの [「SQL 生成パラメータの設定」](#) を参照してください。

- 3 一覧の ANSI92 パラメータをクリックします。

- 4 [値]ボックスに「Yes」と入力します。

- 5 [置換]ボタンをクリックします。

- 6 [OK]をクリックします。

これで、ANSI 92 標準を現在のユニバースの結合定義に適用できます。[結合の編集] ダイアログボックスの [詳細設定] ボタンをクリックすると、[結合の詳細プロパティ] ボックスが表示されます。フィルタを定義して、結合の FROM 句に含めるディメンションを指定できます。

4.7.10.3 ANSI 92 構文を使用した結合の定義

ANSI 92 構文を使用して、[結合の編集] ダイアログボックスから結合を定義できます。これは、詳細設定編集ボックスを使用して行います。このボックスでは、結合定義に含めるオブジェクトを選択できます。

ANSI 92 構文を使用して結合を定義する方法は、次の通りです。

- 1 ユニバースで ANSI 92 サポートを有効にします。詳しい説明は、170 ページの [「ユニバースでの ANSI 92 サポートの有効化」](#) を参照してください。

- 2 スキーマの結合をダブルクリックします。
[結合の編集]ダイアログボックスが表示されます。
- 3 [詳細設定]ボタンをクリックします。
[結合の詳細プロパティ]ダイアログボックスが開きます。



- 4 ドロップダウンリストから、次の FROM 句のいずれかのフィルタを選択します。

FROM 句のオプション	説明
デフォルトの動作	結合のデフォルト構文が適用されます。結合は WHERE 句で定義されます。
FROM 句のオブジェクトすべて	結合の左右にあるテーブルの列で定義されたすべてのオブジェクトが FROM 句に含まれます。
FROM 句のオブジェクトなし	FROM 句にオブジェクトは含まれません。
FROM で選択されたオブジェクト	結合テーブルの[結合の詳細プロパティ]ツリービューで選択したオブジェクトだけが FROM 句に含まれます。

- 5 [FROM で選択されたオブジェクト] を選択した場合、FROM 句に含めるオブジェクトを選択します。

- 6 [OK]をクリックします。
- 7 [結合の編集] ボックスにその他の結合パラメータを入力します。
- 8 [OK]をクリックします。

4.7.11 結合の削除

結合を削除する手順は、次の通りです。

- 1 削除する結合をクリックします。
結合が選択されます。
- 2 次のいずれかの操作を行います。
 - ・ キーボード上のBackspaceキーを押す。
 - ・ キーボード上のDeleteキーを押す。
 - ・ 結合を右クリックし、ポップアップメニューから[消去]を選択する。結合の削除を確認するメッセージが表示されます。
- 3 [はい]をクリックします。
結合が削除されました。

注

結合を削除する際には、スキーマとユニバースに対する影響を十分に考慮してください。結合を削除してもコンテキストに影響が出ないことを確認します。コンテキストで使用されている結合を削除しようすると、ユニバースデザインツールからの警告メッセージが表示されます。どのコンテキストが影響を受けるかを手動で確認し、ユニバースに対する影響を見極める必要があります。

4.8 特定の結合の種類の定義

ユニバースデザインツールでは、以下の種類の結合が定義できます。

結合の種類	説明
等結合 (複合等結合を含む)	1 つのテーブルの列の値ともう一方のテーブルの列の値の等価に基づいてテーブルをリンクします。両方のテーブルに同じ列があるので、結合で 2 つのテーブルを同期化します。 1 つの結合で複数列をリンクする複合等価結合を作成することもできます。
シータ結合(条件付結合)	2 つの列の等価ではなく、関係によってテーブルをリンクします。
外部結合の作成	一方のテーブルの行が他方のテーブルの共通列の値とは一致しないテーブル同士をリンクする結合
ショートカット結合	中間テーブルを経由せずに、双方向で同じ結果を生じる代替パスを提供する結合: 結合のパスをできるだけ短くすることにより、クエリ時間を最適化します。
自己制限結合	テーブルを制限するため、同じ 1 つのテーブルに適用する結合

次の節では、それぞれの結合について詳しく説明していきます。どの種類の結合でも作成方法は同じですが、[結合の編集]ダイアログボックスで定義するプロパティは異なります。

4.8.1 等結合の作成

等結合は、テーブル 1 とテーブル 2 の列にある共有の値によって、2 つのテーブルをリンクします。次の構文で制限が適用されます。

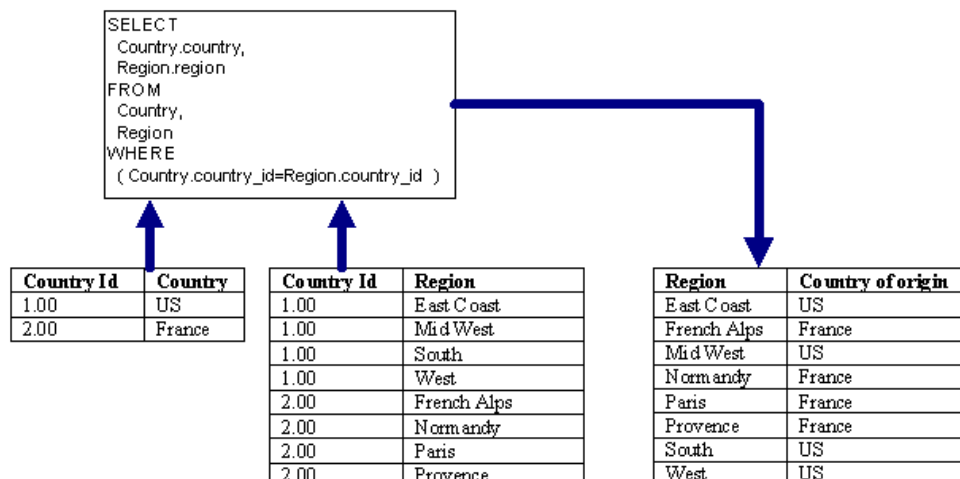
テーブル1.列_a = テーブル2.列_a

標準化されたデータベースでは通常、等結合に含まれる列は一方のテーブルの 1 次キーとほか方のテーブルの外部キーです。キーに関する詳しい説明は、156 ページの「[プライマリキーと外部キーの結合](#)」を参照してください。

新しい結合を作成すると、デフォルトでは等結合が作成されます。スキーマの結合のほとんどは、等価結合になります。

例 等結合によるデータの制限

下記の例ではSelect文が実行されると、Select句とFrom句にしたがってデカルト積が生成されます。しかしデータが返される前にWhere句による制限が適用され、両方のテーブルの[Country ID]列で値が一致する行だけが返されます。



4.8.1.1 新しい等結合の作成

新しい等結合を作成する手順は、次の通りです。

- ・ 2つのテーブル間に結合を作成します。

新しい結合はデフォルトで等結合になります。

ヒント

結合の作成方法についての説明は、157 ページの「[結合の作成](#)」を参照してください。

4.8.1.2 既存の結合からの等結合の作成

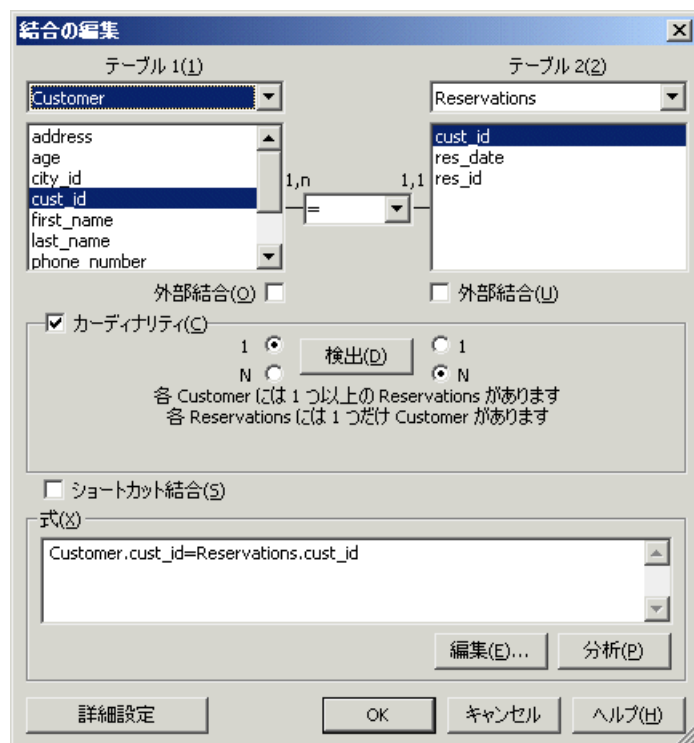
既存の結合から等結合を作成する手順は、次の通りです。

- 1 既存の結合をダブルクリックします。

[結合の編集] ダイアログボックスが表示されます。

- 2 [テーブル1]のリストボックスから列を選択します。
- 3 [テーブル2]のリストボックスから一致する列を選択します。
- 4 演算子ドロップダウンリストボックスから[=]を選択します。

次の[結合の編集]ダイアログボックスは、[Customer]と[Reservations]という2つのテーブル間に作成した等結合の例です。



注

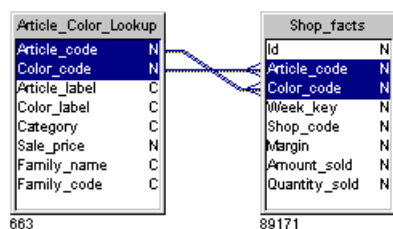
共通列に同じ名前が付けられているとは限りません。データベースでの1次キーと外部キーの列名を確認する必要があります。複数のテーブルで同じキー列が使われている場合でも、データベース内でのテーブルの役割によってそれぞれのテーブルで異なった列名となっていることもあります。

- 5 [分析]ボタンをクリックし、構文が正しいかチェックします。
エラーメッセージが表示される場合、2つのテーブルで列が共通かどうかを確認します。
- 6 [OK]をクリックします。

4.8.1.3 複合等結合の作成

複合等価結合を作成することもできます。これは、2 つのテーブル間にある複数の列をリンクする単一結合です。[結合の編集] ダイアログボックスにある Complex 演算子を使って、複合等価結合を作成できます。

サンプルユニバースのeFashionには、次の複合等結合が含まれています。



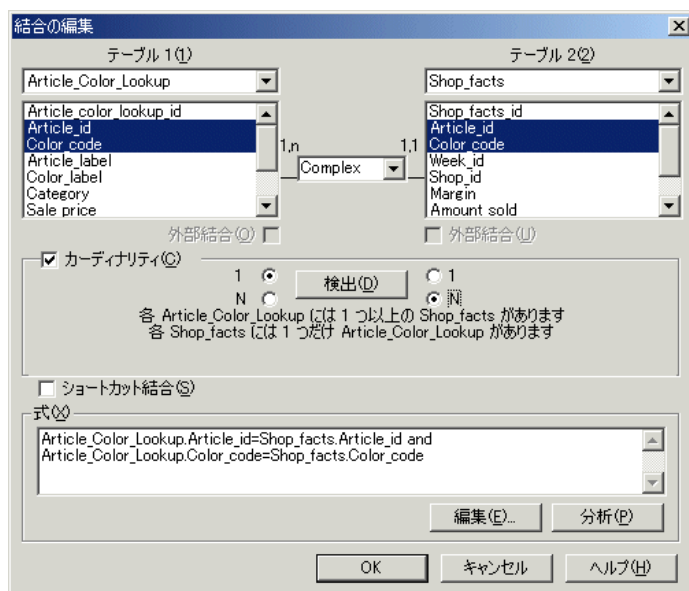
結合された列に対して、複数の単一等価結合ではなく、複合等価結合を使うと、次のような利点があります。

- ・ 1 つのカーディナリティのみを検出する。これは、時間の節約につながり、スキーマを読みやすい状態にまとめておくことができます。
- ・ [結合の編集] ダイアログボックスの [式] ボックスで、2 つのテーブル間にあるすべての結合の SQL を表示することができます。2 つのテーブル間で複数の単一等価結合を使用するとき、各結合に対して式が 1 つあります。

複合等結合を作成する手順は、次の通りです。

- 1 既存の結合をダブルクリックします。
[結合の編集] ダイアログボックスが表示されます。
- 2 [テーブル 1] ボックスで、複数の列を選択します。
- 3 [テーブル2]のリストボックスから一致する複数の列を選択します。
- 4 演算子のドロップダウンリストから "Complex" を選択します。

次の[結合の編集]ダイアログボックスは、[Article_Color_Lookup]と[Shop_facts]間の複合等結合の例です。



- 5 [分析]ボタンをクリックし、構文が正しいかチェックします。
エラーメッセージが表示される場合、2 つのテーブルで列が共通かどうかを確認します。
- 6 [OK]をクリックします。

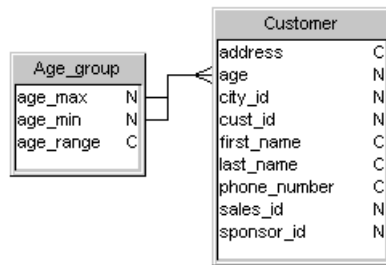
4.8.2 シータ結合

シータ結合は、2 つの列の等しくない関係に基づいてテーブルをリンクする結合です。シータ結合には、[=]以外の演算子を指定します。

次の例は、Between 演算子を使ったシータ結合の作成例です。

例 複合結合

次の[Age_Group]テーブルには、顧客の年令区分を分析するために使われる年令区分の情報が保存されています。

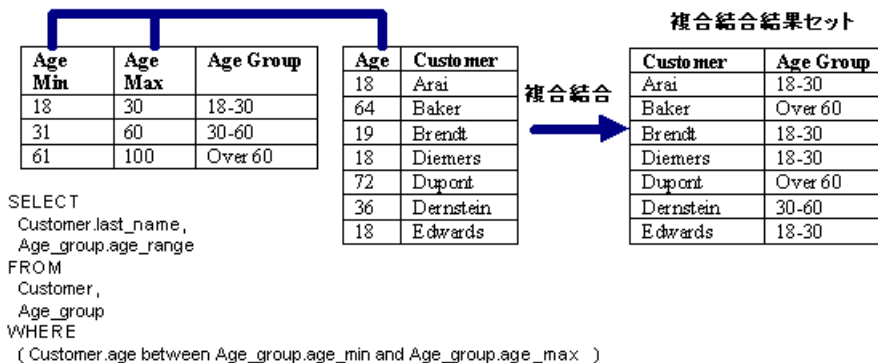


ユニバースにこのテーブルを含める必要がありますが、[Customer]テーブルと[Age_Group]テーブルには共通列がないため、等結合は設定できません。

したがって、最高年齢 (age_max) と最低年齢 (age_min) に Between 演算子を使ったシータ結合を作成します。複合結合を使うことで、[Customer]テーブルの[Age]列の値が、[Age_Group]テーブルの[age_max]と[age_min]の値の範囲に含まれるように定義する結合を作成できます。この結合は、次の式で表現されます。

Customer.age between Age_group.age_min and Age_group.age_max

次の図は、[Age_max]、[Age_min]、[Age]間の結合と、[Age_Group]と[Customer]の両方のテーブルを含むクエリで複合結合を使用した際に返される結果を示したものです。



4.8.2.1 複合結合の作成

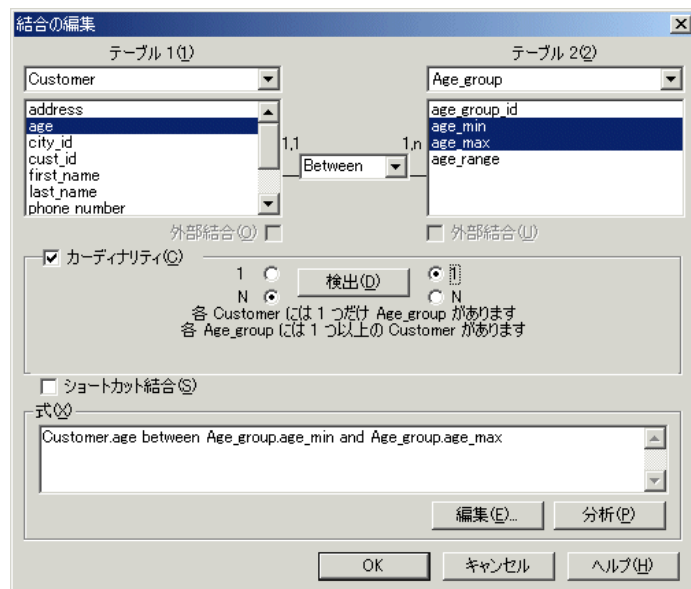
範囲列を使って複合結合を作成する手順は、次の通りです。

- 1 2つのテーブル間に結合を作成します。
デフォルトで、等価結合が作成されます。
- 2 作成した結合をダブルクリックします。

[結合の編集]ダイアログボックスが開きます。

- 3 テーブル 1 の列リストで列を選択します。
- 4 Ctrl キーを押しながら、テーブル 2 の列リストで列を 2 つ選択します。

次の図では、[Age_min]と[Age_max]の2列が選択されています。演算子ドロップダウンリストが自動的に [Between] に変わります。

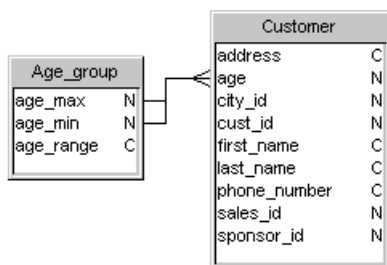


- 5 [分析]ボタンをクリックし、構文が正しいかチェックします。

エラーメッセージが表示される場合、選択した列をもう一度確認してください。

- 6 [OK]をクリックします。

構造枠で結合が作成されます。



4.8.3 外部結合

外部結合は、2つのテーブルをリンクしますが、片方のテーブルにはもう一方のテーブルの共通の列に相当しない列があります。

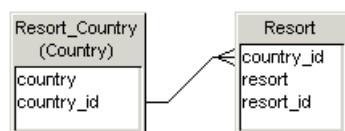
オリジナルの等価結合で外部テーブルになっているテーブルを指定して、外部結合を定義します。外部テーブルには、相当しない列があったとしても、すべての値を返す列が含まれています。選択した結合の[結合の編集]ダイアログボックスで、外部テーブルを指定します。

4.8.3.1 完全外部結合

デフォルトで、外部テーブルを結合の右または左に指定することにより、右外部結合または左外部結合を作成できます。また、ユニバースで結合のANSI 92サポートを有効化することにより、完全外部結合も作成できます。これは、ユニバースのSQLパラメータANSI 92([ファイル] > [パラメータ] > [パラメータ])を[Yes]に設定して行います。これにより、ユニバースの結合でANSI 92構文がサポートされ、結合の左右にあるテーブルを外部テーブルとして選択できます。完全外部結合の作成についての詳細は、182ページの[「完全外部結合の定義」](#)を参照してください。

例 外部結合

次の[Resort_Country]テーブルと[Resort]テーブルは、等結合でリンクされています。



それぞれのリゾートは1つの国に属していますが、それぞれの国にリゾートがあるわけではありません。等結合を使用すると、クエリ結果にはリゾートを持った国であるAustralia、France、USしか含まれなくなります。

国	リゾート
Australia	Australian Reef
France	French Riviera
US	Bahamas Beach
US	Hawaiian Club
US	Royal Caribbean

[Resort] テーブルの外部キーと一致する値とは無関係にすべての国を表示するには、Resort 列に対応する値がない国もクエリ結果に含まれるよう外部結合を定義します。

この外部結合の構文(Microsoft Access用)は、次の通りです。

```
SELECT
Resort_Country.country,
Resort.resort
FROM
Country Resort_Country,
Resort,
{ oj Resort_Country LEFT OUTER JOIN Resort ON Resort_Country.country_id=Resort.country_id }
```

注

上記の例では、Microsoft Accessを使用しています。[Resort]テーブルに続くすべての一对多の結合は、外部結合でなければなりません。外部結合としない場合、この外部結合で返されるNULL値は、続く結合に一致するNULL値が無ければ無視されます。外部結合は、RDBMS特定の結合です。詳細は、RDBMSのドキュメンテーションを参照してください。外部結合に関する制限についての詳しい説明は、183 ページの「[外部結合の使用に関する制限事項](#)」も参照してください。

4.8.3.2 外部結合の作成

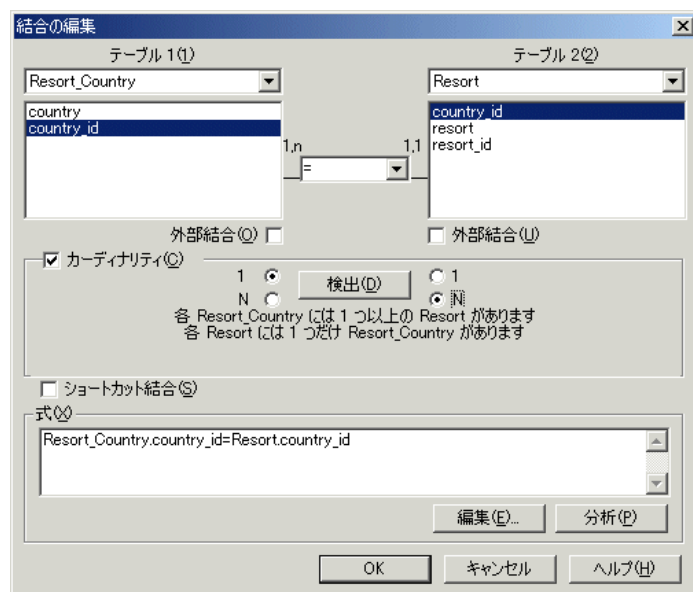
外部結合を作成する手順は、次の通りです。

- 1 既存の等結合をダブルクリックします。

[結合の編集]ダイアログボックスが開きます。

- 2 クエリですべての値を返すテーブルの [外部結合] チェックボックスを選択します。

次の例では、[Resort_Country]テーブルからすべての値を取得します。



- 3 [分析]ボタンをクリックし、構文が正しいかチェックします。

エラーメッセージが表示される場合、選択した列をもう一度確認してください。

- 4 [OK]をクリックします。

ユニバースデザインツールでは、構造枠に結合が表示されます。外部結合には、一致しない値を返すテーブルの反対側に小さな丸印が表示されます。



4.8.3.3 完全外部結合の定義

外部結合を定義する際、ANSI 92 標準を使用して定義することができます。これにより、完全外部結合を指定できます。外部結合に ANSI 92 標準を使用するには、ANSI 92 パラメータを [Yes] に設定する必要があります。このパラメータは、[パラメータ] ページ ([ファイル] > [パラメータ] > [パラメータ]) で利用できます。

注

ユニバースでのこのパラメータやその他の SQL 生成パラメータの設定については、101 ページの「[SQL 生成パラメータの設定](#)」を参照してください。

ANSI 92 パラメータを [Yes] に設定すると、結合の左右にあるテーブルを外部テーブルとして選択できます。このパラメータを設定する前に、ターゲットの RDBMS で外部結合の ANSI 92 構文がサポートされていることを確認する必要があります。

完全外部結合の定義には 2 段階あります。

- ・ ユニバースで外部結合の ANSI 92 サポートを有効にする。詳しい説明は、170 ページの「[ユニバースでの ANSI 92 サポートの有効化](#)」を参照してください。
- ・ [結合の編集] ダイアログボックスを使用して、完全外部結合を定義します。

完全外部結合を定義する手順は、次の通りです。

- 1 ユニバースで ANSI 92 サポートを有効にします。
- 2 スキーマの結合をダブルクリックします。

[結合の編集]ダイアログボックスが表示されます。

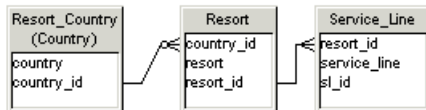
- 3 結合に含める両テーブルの[外部結合]チェックボックスをオンにします。
- 4 [OK]をクリックします。

ユニバースデザインツールでは、構造枠に結合が表示されます。完全外部結合は、テーブル間の結合リンクにある 2 つの円で示されます。

4.8.3.4 外部結合の使用に関する制限事項

外部結合は便利な機能ですが、次のようなパフォーマンスと実装の問題に注意してください。

- ・ パフォーマンスが低下する可能性があります。返される行が増え、データベースによっては外部結合が含まれるとインデックスを使用しないため、大量のデータによってクエリのパフォーマンスが劣化する。
- ・ データベースによる外部結合の使用制限すべてのデータベースが Where 句の外部結合を制御できるわけではありません。これは自己結合を使用する場合に必要となります。たとえば、[TYPE_CODE=10]の自己結合で[TYPE=10]と[Type=NULL]のすべての行を返すことはできませんが、外部結合によって NULL が返されるため[TYPE=10]が TRUE となることはありません。
- ・ 外部結合以降の不完全なクエリパスを避けられるよう、使用する RDBMS での外部結合の処理方法を確認してください。たとえば、Microsoft Access のサンプルデータベース Club.mdb では、結合パスに含まれる外部結合以降のすべての一対多結合は、すべて外部結合として定義する必要があります。外部結合としない場合、元の外部結合は無視されます。下記の例では、[Resort_Country]と[Resort]間の外部結合で返される NULL 値は、[Resort]と[Service_Line]間の結合では無視されます。この 3 つのテーブルを含んだクエリを実行すると、最初の結合を実行するクエリを別に作成し、そのクエリを SQL 文に入れるよう求めるデータベースエラーが返されます。この種のエラーはユーザを混乱させる元ですので、外部結合を使用しないか、パスを外部結合で完了させるようにしてください。



4.8.4 ショートカット結合

ショートカット結合は、2 つのテーブル間に代替パスを提供する結合です。ショートカット結合は、仲介テーブルを考慮せず、長い結合パスを短縮するので、クエリのパフォーマンスが向上します。

ショートカット結合の一般的な使用方法是、共有の lookup テーブルと結合パスの先の方に位置する別のテーブルをリンクすることです。結合パスは、同じコンテキストにある複数の異なるテーブルから成り立っています。

このような場合、対象となる値のテーブル内での階層レベルを下げ、同じ値が結合されるすべてのレベルに存在する場合にのみ、ショートカット結合を使用する価値があります。

指定されたコンテキストに "ショートカット" する結合パスがない場合は、ショートカット結合は無視されます。関連する Web Intelligence クエリに生成された SQL では、無効なショートカット結合は考慮されません。

注

ユニバースデザインツールでは、ループとコンテキストの自動検出にはショートカット結合は考慮されません。しかし、カーディナリティをショートカット結合に設定すると、コンテキストを検出するときに "一部のカーディナリティがまだ設定されていません。" という警告メッセージが表示されなくなります。

4.8.4.1 ショートカット結合の作成

ショートカット結合を作成する手順は、次の通りです。

- 1 直接リンクできる 2 つのテーブルを結合パスで見つけます。
- 2 2 つのテーブル間で結合を作成します。
- 3 作成した結合をダブルクリックします。

[結合の編集]ダイアログボックスが開きます。

- 4 [ショートカット結合]チェックボックスを選択します。
- 5 必要に応じて、他の結合プロパティを選択または入力します。
- 6 [OK]をクリックします。

2 つのテーブルを結合するショートカット結合が表示されます。ショートカット結合は、点線で表示されます。

注

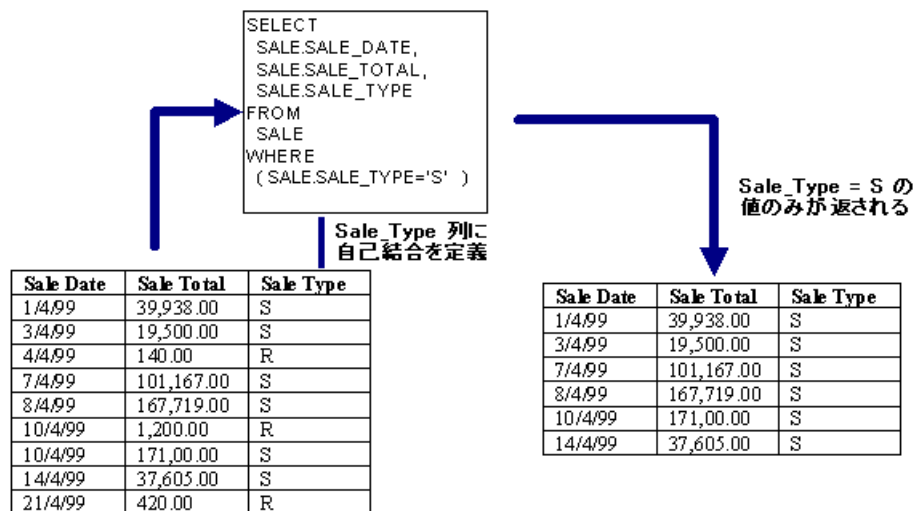
ショートカット結合には、それが置換する結合と同じカーディナリティを設定してください。

4.8.5 自己制限結合

自己制限結合は、正確には結合ではなく、単一のテーブルで自己制限を行います。自己結合を使うと、テーブルが返す値を特定の値だけに制限できます。

例 自己結合

次の[Sales]テーブルには、自動車の販売とレンタルの両方のデータが含まれています。取引の種類は、[Sale_Type]列の値(S=販売、R=レンタル)を使って区別されます。このテーブルに[Sale_Type = S]のデータだけを返す自己結合を設定すると、[Sales]テーブルから作成したオブジェクトと[Sales]テーブルを含む結合を使ったクエリの結果を自動車の販売取引だけに限定できます。



自己結合を設定しない場合は、[Sale_Type] 列の値に「S」か「R」にかかわらず、すべての行が返されます。

ヒント

自己制限結合にカーディナリティを設定すると、コンテキストを検出するときに “一部のカーディナリティがまだ設定されていません。” という警告メッセージが表示されなくなります。カーディナリティは設定されている限りどんな種類でも構いませんが、通常は一对一のカーディナリティを設定しておくことをお勧めします。

4.8.5.1 自己結合の作成

自己結合を作成する手順は、次の通りです。

- 1 [挿入]>[結合]の順に選択します。

[結合の編集]ダイアログボックスが開きます。

- 2 自己結合を設定するテーブルを[テーブル1]のドロップダウンリストから選択します。

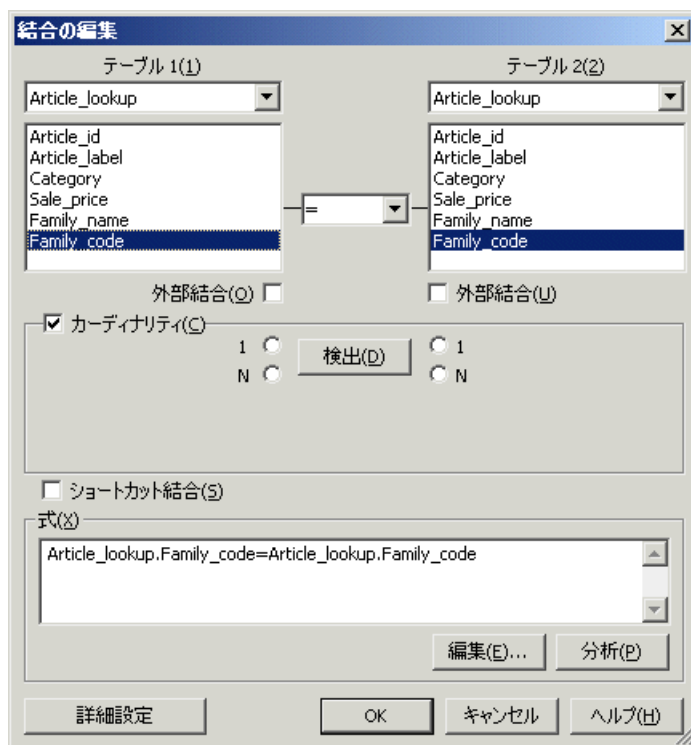
選択したテーブルの列が、テーブルの列ボックスに表示されます。

- 3 列のドロップダウンリストから制限を定義する列をクリックします。

- 4 テーブル 2 のドロップダウンリストで、先に選択したテーブルと同じテーブルを選択します。

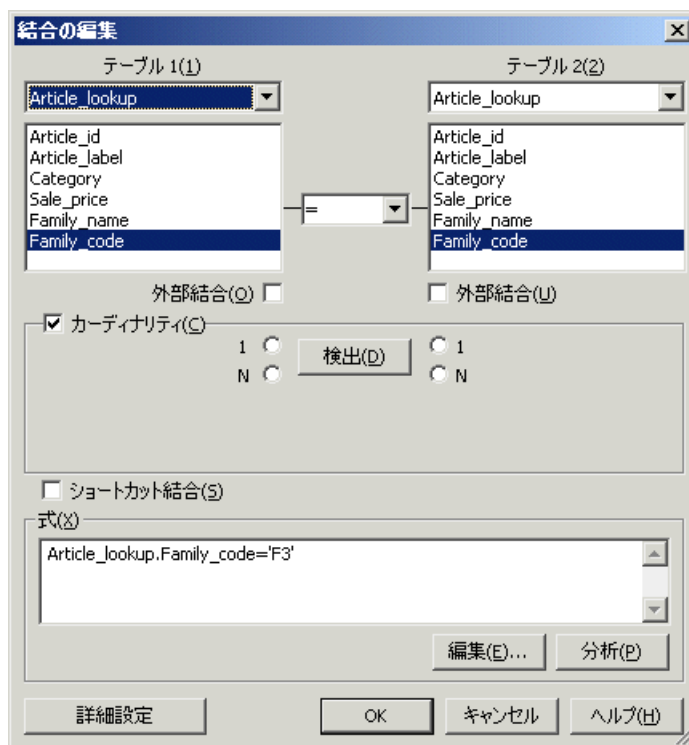
- 5 [テーブル1]で選択した列と同じ列を選択します。

[式]テキストボックスに結合式が表示されます。



- 6 結合式のオペランドを、結合列に設定する制限値に置き換えます。

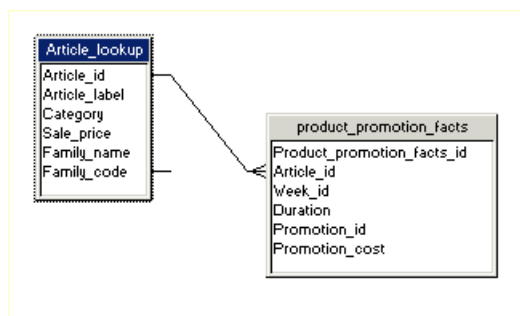
たとえば、[Family_code]列から取得する値を「F3」だけに制限する場合は、等記号の右側に表示される Article_lookup.Family_code を次のように「F3」に変更します。



7 [分析]ボタンをクリックしてシンタックスをチェックします。

8 [OK]をクリックします。

自己制限結合が、結合を定義した列の横に、短い線で表示されます。



4.9 カーディナリティの使い方

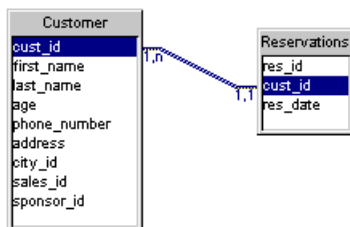
カーディナリティは、2 つのテーブル間でマッチする行数を説明する結合のプロパティです。

カーディナリティは、結合の片方の先の列で最小および最大行数として表示され、その列に対応する列が結合のもう一方の先のテーブルにも存在します。

最小および最大行数は、0、1、または n になります。結合は両方向の関係なので、カーディナリティは必ず 2 つ（各結合の先に 1 つ）に存在します。

例 結合のカーディナリティ

[Customer]と[Reservations]の2つのテーブル間の結合を例にとって説明します。



上の2つのテーブル間の結合におけるカーディナリティは、次のように表現できます。

説明	表記
各顧客[Customer]について、1つまたは複数の予約[Reservations]が可能です。	(1,N)
各予約データ[Reservations]は、特定の顧客[Customer]にだけ対応しています。	(1,1)

4.9.1 ユニバースデザインツールでのカーディナリティの使用方法

結合のカーディナリティには、クエリを実行する時に生成される SQL の役割はありません。ユニバースデザインツールでは、コンテキストと有効なクエリパスの設定にカーディナリティを使用します。

コンテキストは、有効なクエリパスを提供する結合の集合です。コンテキストを使用して、ターゲットデータベースでのテーブルのリンク方法によって多すぎるまたは少なすぎる行数を返す結合の問題を解消します。コンテキストの詳細については、205 ページの [「結合問題の検出と解決」](#)を参照してください。

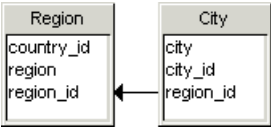
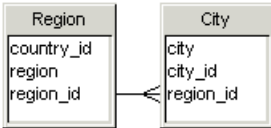

コンテキストを作成すると、クエリ用に生成される SQL で特定の結合パスが使用され、結合の問題が解決されます。

正しいコンテキストと有効な結合パスを定義するには、スキーマ内のすべての結合に正しいカーディナリティを設定する必要があります。

カーディナリティを設定することで、データベース内のテーブルのリレーションシップがより明確となり、結合パスの潜在的な問題点をスキーマ内でグラフィカルに識別できます。

4.9.1.1 カーディナリティの表示

次に記号を使用して構造枠にカーディナリティを表示することができます。

カーディナリティ記号	例	説明
矢印		矢印は、結合の「1」のサイドを示します。カーディナリティが 1、1 の場合は、矢印が結合の両サイドに表示されます。
アリティ		クローフット (山形模様) は、結合の「多」のサイドを示します。カーディナリティが 1、1 の場合、直線になります。
1、n		カーディナリティは、各結合の先で比率を表示します。

カーディナリティを表示する手順は、次の通りです。

- 1 [ツール]>[オプション]の順に選択します。
[オプション] ダイアログボックスの [一般設定] ページが表示されます。
- 2 [グラフィック]タブをクリックします。
[グラフィック] ページが表示されます。
- 3 [矢印]、[アリティ]、[1、n]ボタンをクリックします。
- 4 [OK]をクリックします。

4.9.1.2 結合に設定できるカーディナリティの種類

結合には次のカーディナリティを設定できます。

カーディナリティ	説明
1対1(1, 1)	テーブル1の各行に対して、それに対応する列がそれぞれ1行だけテーブル2に存在します。
1対多(1, n)	テーブル1の各行に対して、それに対応する列がそれぞれ1行または複数行テーブル2に存在します。
多対1(n, 1)	「1対多」と同じですが、カーディナリティの方向が逆になります。
多対多(n, n)	<p>テーブル1の各行または複数行に対して、それに対応する列が1行または複数行テーブル2に存在します。</p> <p>「多対多」カーディナリティは、リレーショナルデータベースではまれなケースで重複行を返します。その結果、パフォーマンス速度が落ち、不正確な結果が返される恐れもあります。「多対多」カーディナリティがある場合、相当する結合を再チェックし、テーブル間の関係を正確に把握してください。</p>

カーディナリティは、手動、またはユニバースデザインツールのカーディナリティの自動検出ツールを使って設定できます。次のセクションでは、この2種類の設定方法を説明します。

4.9.2 カーディナリティの手動設定

[結合の編集]ダイアログボックスでカーディナリティを定義して、結合にカーディナリティを手動設定することができます。

カーディナリティを手動設定する利点

カーディナリティを手動設定するときは、各結合を個別に考慮する必要があります。これにより、スキーマで起こりうる結合パス問題を予想、察知することができます。カーディナリティの自動検出を選択している場合は、このような問題を発見できない場合もあります。たとえば、結合パスの端に半端な一対一結合があったり、プライマリキーの超過によりすべての列で一意性が確保されない場合などがあります。

キーを理解する

各テーブルの一次キーと外部キーを評価して、結合へのカーディナリティを決定します。一次キーと外部キーの説明は、次のとおりです。

キー	説明
プライマリ	値でテーブルの各行を識別するテーブルの列または列の組み合わせ。一次キーで、テーブルの各行が一意であることを保証します。各テーブルには、一次キーが 1 つだけあります。
外部	他のテーブルのプライマリキーあるいは一意のキーに一致する値を持った1つあるいは複数の列の組合せ 外部キーで、「作成されていない顧客に売上を作成することはできない」といった制限を適用します。各テーブルには複数のフォーリンキーを設定できます。

カーディナリティを設定する基準

一次および外部キーの関係を評価して、結合のカーディナリティを次のように決定します。

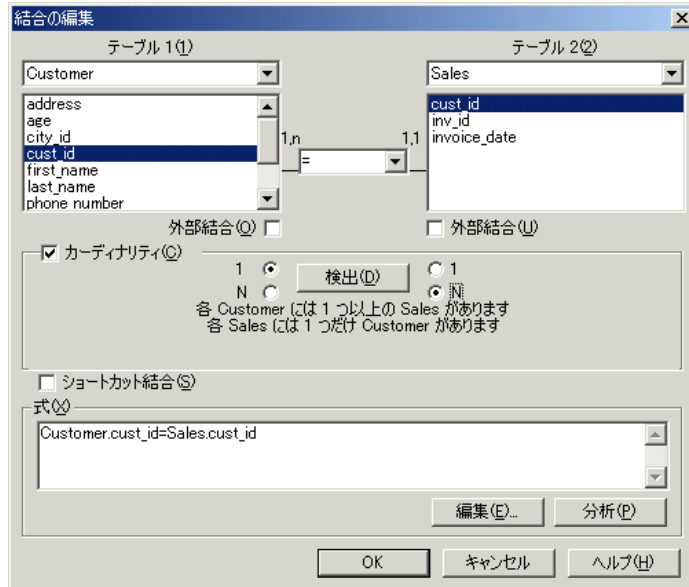
結合でリンクするもの	予想されるカーディナリティ
テーブル 1 の完全な一次キーと、テーブル 2 の完全なプライマリキー。以下はその例です。	1対1(1, 1) 各一次キー値に対し、各テーブルから 1 行のみが返されます。

結合でリンクするもの	予想されるカーディナリティ
テーブル 1 の 1 つのテーブルの完全な一次キーと、それに対応するテーブル 2 の外部キー。以下はその例です。	1 対多 (1、N) テーブルのフォーリンキー値は一意ではないこともあるので、オリジナルのテーブルの 1 つのプライマリキー値に対して多くのマッチする値が返されます。

結合でリンクするもの	予想されるカーディナリティ
テーブル 1 の完全な一次キーとテーブル 2 の一部のプライマリキー。以下はその例です。	1 対多 (1、N) 一次キーの一部しかマッチしていないので、オリジナルのテーブルの 1 つの一次キー値に対して多くのマッチする値が返されます。

4.9.2.1 カーディナリティを手動で設定する手順は、次の通りです。

- 1 結合をダブルクリックします。
または
結合をクリックして[編集] > [プロパティ]の順に選択します。
[結合の編集]ダイアログボックスが開きます。
- 2 [カーディナリティ]チェックボックスを選択します。
- 3 [テーブル 1]で、1 または N を選択します。
- 4 [テーブル 2]で、1 または N を選択します。



5 [OK]をクリックします。

4.9.2.2 カーディナリティの自動検出

ユニバースデザインツールの[カーディナリティの検出]機能を使って、以下の場合のカーディナリティを自動検出できます。

- ・ 選択した結合
- ・ すべての結合
- ・ 新規結合の作成時
- ・ [結合の編集]ボックスから

[カーディナリティの自動検出] を使用すると、カーディナリティが自動的に検出されます。

注

この機能は、適切に使用する必要があります。スキーマで検出されたすべてのカーディナリティを素早く収集することができますが、リレーショナルデータベースに多くの構造問題を継承して、不正確なカーディナリティの検出結果を招く恐れがあります。これには、不完全なプライマリ結合や、編集されすぎたプライマリキーなどが含まれます。これに関する詳しい説明は、196 ページの「[カーディナリティによるデータベース制限の解決](#)」を参照してください。

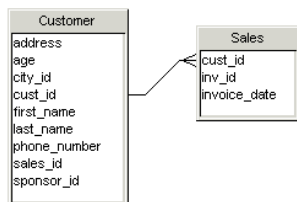
選択した結合のカーディナリティを自動検出する

選択した結合のカーディナリティを自動検出するには、次のどちらかの操作を行います。

- ・ 結合をクリックし、[ツール]>[カーディナリティの検出]の順に選択する。

- 結合を右クリックし、ドロップダウンメニューで[カーディナリティの検出]を選択します。

カーディナリティが、結合の先が複数に分かれた山模様で表示されます。



結合を選択せずに[ツール]>[カーディナリティの検出]の順に選択すると、「結合が選択されていません。すべての結合のカーディナリティを検出しますか?」というメッセージが表示されます。

すべての結合のカーディナリティを自動検出する手順は、次の通りです。

すべての結合のカーディナリティを自動検出するには、次の操作を行います。

- 1 [ツール]>[自動検出]>[カーディナリティの検出]の順に選択します。

または



[カーディナリティの検出]ボタンをクリックします。

すべての結合のカーディナリティを検出するかどうかを尋ねるメッセージが表示されます。

- 2 [はい]をクリックします。

構造枠のすべての結合がカーディナリティ付きで表示されます。

結合の作成時にカーディナリティを自動検出する

結合の作成時にカーディナリティを自動検出する手順は、次の通りです。

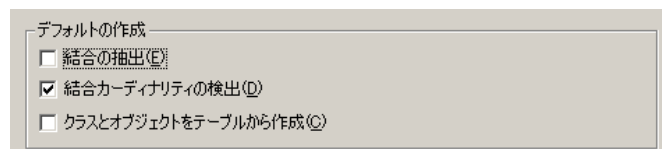
- 1 [ツール]>[オプション]の順に選択します。

[オプション] ダイアログボックスの[一般設定] ページが表示されます。

- 2 [データベース]タブをクリックします。

[データベース] ページが開きます。

- 3 [結合カーディナリティの検出]チェックボックスをチェックします。



- 4 [OK]をクリックします。

- 5 新しい結合を作成すると、カーディナリティが自動検出され、結合の上に表示されます。

[結合の編集]ダイアログボックスからカーディナリティを自動検出する

[結合の編集]ダイアログボックスからカーディナリティを自動検出する手順は、次の通りです。

- 1 結合をダブルクリックします。

[結合の編集]ダイアログボックスが開きます。

- 2 [カーディナリティ] チェックボックスを選択します。
- 3 [検出]ボタンをクリックします。

検出されたカーディナリティには、このボタンが自動的に選択されています。双方向のカーディナリティの内容が文章でも表示されます。



- 4 [OK]をクリックします。

4.9.2.3 カーディナリティの自動検出の最適化

使用する RDBMS に対応する PRM ファイルのパラメータを変更すると、カーディナリティ検出の応答時間を改善できます。これは、検出アルゴリズムが読み込むSQL文を3から2に減らし、パフォーマンスを改善するものです。

PRM ファイルは、ユニバース作成と Web Intelligence でのクエリ生成に使用するパラメータを記録したテキストファイルです。サポートするそれぞれのRDBMSに対し、専用のPRMファイルがあります。

PRM ファイルは、<INSTALLDIR>%win32_x86%dataAccess%ConnectionServer% の下の database フォルダにあります。

接続が使用するPRMファイルの確認

ユニバース接続が使用するPRMファイルを確認する手順は、次の通りです。

- 1 [ファイル]>[パラメータ]を選択します。

[パラメータ]ダイアログボックスが開きます。

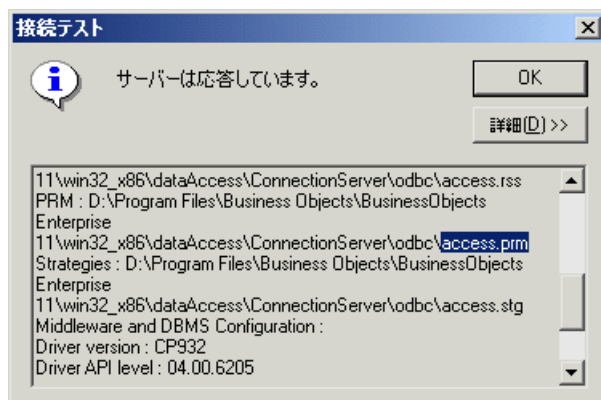
- 2 [テスト]ボタンをクリックします。

[接続テスト]メッセージボックスが表示されます。

- 3 [詳細]ボタンをクリックします。

接続の詳細情報が、ドロップダウンメッセージボックスに表示されます。

- 4 メッセージボックスを、PRMで始まる行までスクロールダウンします。
この行には、ユニバースが使用するPRMファイルの名前とパスが表示されます。



- 5 [OK]をクリックします。
[パラメータ]ダイアログボックスに戻ります。
- 6 [キャンセル]をクリックします。

PRMファイルを使ったカーディナリティ検出の最適化

PRMファイルを使ったカーディナリティ検出を最適化する手順は、次の通りです。

- 1 使用するデータベースに対応したPRMファイルをテキストエディタで開きます。
PRMファイルは、Business ObjectsパスのData Accessフォルダのサブフォルダ内にあります。次のパラメータを追加します。
- 2 LIGHT_DETECT_CARDINALITY = YES
- 3 PRMファイルを保存して、エディタを終了します。
次にユニバースを開いた時に、カーディナリティの自動検出が最適化されます。

4.9.2.4 カーディナリティによるデータベース制限の解決

次の基準にしたがってカーディナリティを設定し、スキーマデザインの失敗を導くような特殊な結合の問題の解決することができます。

問題	解決策
ルックアップテーブルの 1 次キーが 2 列で構成され、それぞれの列が別のファクトテーブルにリンクされている。どちらの結合も 1 次キーの不完全一致のため、多対多結合である。	<p>ルックアップテーブル側のカーディナリティを“多”から“1”に変更します。手順は、次の通りです。</p> <p>ルックアップテーブルに「lookup.pk_column = pk_column 値」のような自己結合(一対一)を追加します。これにより、ルックアップテーブル側の 1 次キーの値の一意性が確保され、ルックアップテーブル側の結合カーディナリティが“1”になります。</p>
1 次キーが超過しているため、一意性を確保するのに 1 次キーのすべての列が必要ではない。	<p>ターゲットデータベースの DBA (データベース管理者)の資格をもっている場合は、複数列の 1 次キーを英数字を使った単一系列の識別子に変更します。これで、複数列のプライマリキーでは難しかった、テーブルのカーディナリティを“1”にすることができます。DBA の資格をもっていない場合は、データベース管理者に相談してください。</p>

4.10 ユニバースのチェック

ユニバースのデザインが進むにつれ、その整合性を定期的にチェックすることが必要になります。ユニバースの整合性は、次の方法でチェックします。

チェック方法	説明
自動	ユニバースデザインツール オプションでは、定義の作成時、ユニバースのエクスポート時、ユニバースファイルを開いたときにユニバース構造の SQL 構文を自動的にチェックするよう設定できます。
手動	[整合性のチェック]コマンドで、選択したユニバース構造の整合性をチェックできます。

4.10.1 ユニバースの整合性の自動チェック

ユニバースデザインツールでは、次の整合性チェックオプションを使用して、定義の作成時、ユニバースのエクスポート時、ユニバースファイルを開いたときに SQL 構造を分析することができます。

自動チェックオプション	説明
定義の自動分析	ユニバースデザインツールでは、すべてのオブジェクト、条件、結合の作成時に、SQL 定義が自動的にチェックされます。[OK]をクリックすると、構造の作成時のチェックが有効になります。
エクスポート前に整合性チェックの警告を表示	ユニバースデザインツールでは、整合性をチェックしていないユニバースをエクスポートしようとする警告メッセージが表示されます。
開くときにユニバース構造をチェック	ユニバースを開く際に自動的に整合性をチェックします。

4.10.1.1 ユニバースの自動チェックオプションの設定

ユニバースの自動チェックオプションは、次の手順で設定します。

- 1 [ツール]>[オプション]の順に選択します。

[オプション] ダイアログボックスの [一般設定] ページが表示されます。

- 2 必要に応じ、[整合性]グループボックスの自動チェックオプションを選択します。

整合性

☒ 定義の自動分析 (P)

☐ エクスポート前に整合性チェックの警告を表示 (W)

☐ 開くときにユニバース構造をチェック (S)

- 3 [OK]をクリックします。

4.10.1.2 ユニバースの整合性の手動チェック

[整合性のチェック] オプションを使用して、アクティブユニバースのデザインが正確で最新のものであるかどうかを確認することができます。

「整合性のチェック」の目的は、以下の通りです。

- ・ ユニバースのオブジェクト、結合、カーディナリティのエラーを検出する。
- ・ 結合パスのループを検出する。
- ・ 必要なコンテキストを検出する。
- ・ ユニバースが接続するデータベースが変更されていないか検査する。

ユニバースの要素をデータベースと照らし合わせる前に、データベースへの接続が有効であるかどうかをこのオプションを使って確認します。接続が無効な場合は、チェックは継続されず、エラーメッセージが表示されます。

4.10.1.3 整合性のチェックで検出できるエラーの種類

整合性チェックでは、以下のエラーを検出します。

- ・ オブジェクト、条件、結合のSQL定義内の無効な構文
- ・ ループ
- ・ 孤立テーブル
- ・ 孤立結合
- ・ コンテキスト内のループ
- ・ 未設定あるいは間違って設定されているカーディナリティ

整合性チェックによるデータベース更新の検出

整合性チェックでは、まずデータベースに対してテーブルリストをリクエストします。次に、取得したリストをユニバースのテーブルと比較します。列に対しても同じ作業を行います。

リスト内のテーブルや列と一致しないテーブルと列は、構造枠内で使用できないテーブルとしてマークされます。こういったテーブルや列は、データベースで削除されたり、名前を変更されたテーブルや列であることがよくあります。詳しくは、202 ページの「[ユニバース構造の最新表示](#)」を参照してください。

注

データ量が多いと、カーディナリティのチェックに時間がかかる場合があります。データが曖昧、または不足していると、不正確な結果を得ることになります。データベースが非常に大きく、データ入力が不完全な場合、

[カーディナリティのチェック] オプションを使用しないでください。このオプションを使う場合は、PRMファイルをカスタマイズして自動検出を最適化できます。詳しい説明は、195 ページの「[カーディナリティの自動検出の最適化](#)」を参照してください。

4.10.1.4 整合性のチェック機能を使ったユニバースのチェック

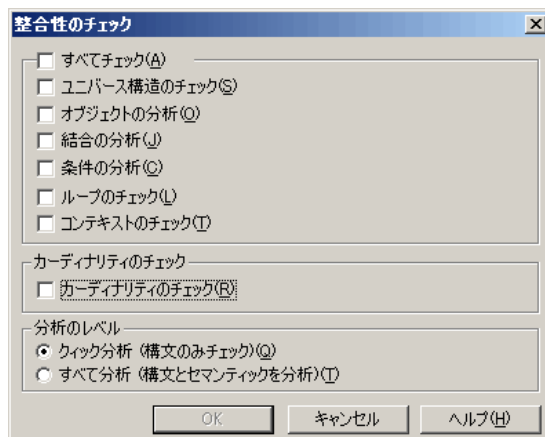
ユニバースの整合性をチェックする手順は、次の通りです。

- 1 [ツール]>[整合性のチェック]の順に選択します。

または

[整合性のチェック]ボタンをクリックします。

- 2 [整合性のチェック]ダイアログボックスが開きます。



- 3 チェックするコンポーネントを選択します。

注

[すべてチェック]オプションとは別に、[カーディナリティのチェック]オプションを選択することができます。これにより、データベースによっては長時間かかるカーディナリティをチェックせずにユニバース構造をチェックすることができます。

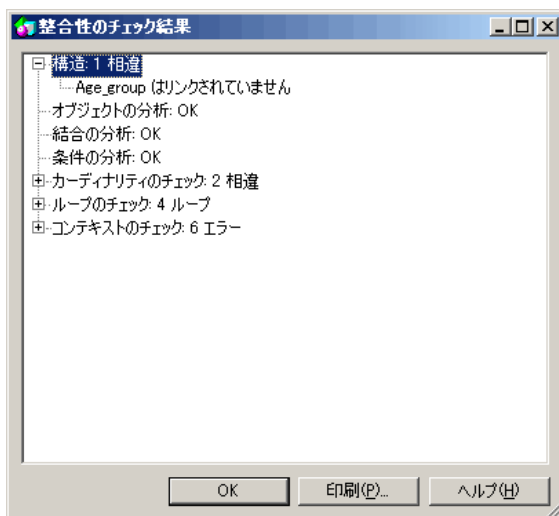
- 4 チェックしないコンポーネントを選択解除します。
- 5 [分析のレベル]グループボックスでは、[クイック分析]を選択すると、コンポーネントの構文のみをチェックします。

または

[すべて分析]を選択すると、コンポーネントの構文とセマンティックをチェックします。

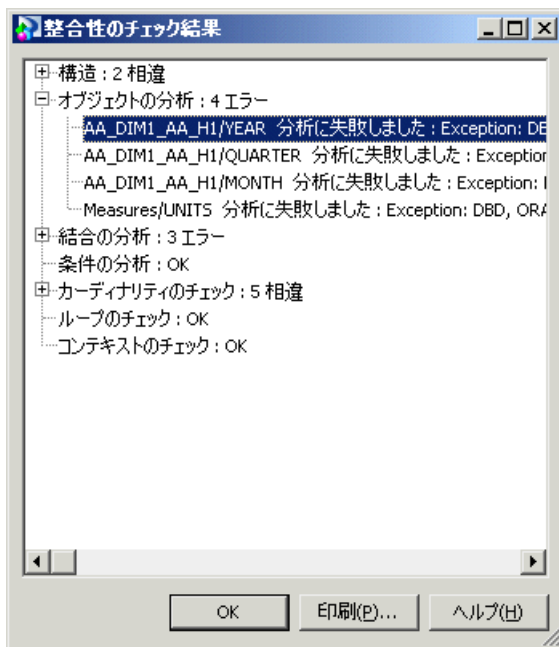
- 6 [OK]をクリックします。

[整合性のチェック結果]ダイアログボックスが開き、チェックの進行状況に合わせてチェック結果が表示されていきます。



エラーが検出されなかった場合は、各チェック項目の横に[OK]が表示されます。

- 7 左端の(+)記号をクリックして一覧を展開し、エラーの詳細を表示します。



一覧の項目をダブルクリックして、構造枠にある対応するコンポーネントを反転表示することができます。

- 8 [印刷]をクリックすると、このダイアログボックスの内容を印刷できます。
- 9 [OK]をクリックします。

注

[ループのチェック]オプションを使用する場合は、あらかじめすべてのカーディナリティが設定されていることを確認してください。そうでないと、結合のループが正しくチェックされません。

4.10.1.5 ユニバース構造の最新表示

整合性のチェックで、ユニバースの基となるデータベースが更新されていることがわかった場合は、[表示]メニューの[構造の最新表示]コマンドを使って、構造枠の内容を更新できます。

このコマンドで検出できるデータベースの更新と、それに対するユニバース構造の変更は次の通りです。

操作	ユニバースデザインツールで以下の項目が実行されます。
テーブルに列が追加された	ユニバースの対応するテーブルへ列を追加します。
テーブルから列が削除された	削除すべき列とそれに関連した結合を表示する警告メッセージを表示します。
データベースからテーブルが削除された	削除すべきテーブルとそれに関連した結合を示す警告メッセージを表示します。
データベース内のテーブル名が変更された	「選択されたテーブルが見つかりません」というメッセージを表示します。データベースのテーブル名に合わせ、ユニバースのテーブル名を変更する必要があります。名前の変更後も一致するテーブル名がデータベースに見つからない場合は、ユニバースデザインツールから「選択されたテーブルがデータベースに存在しません」というメッセージが返されます。
データベースは変更されていない	「更新の必要がありません」というメッセージを表示します。

ユニバース構造を最新表示する手順は、次のとおりです。

- ・ [表示]>[構造の最新表示]を選択します。
- ・ メッセージボックスが開き、データベースの更新内容か、更新の必要がないことを示すメッセージが表示されます。

スキーマでの結合の問題の解決

5.1 概要

この章では、ユニバースのスキーマをテーブルや結合で埋めていくにつれ、発生する結合の問題について説明します。そのような結合の問題を検出して解決し、ユニバースで実行するクエリからの結合パスから正確な結果が返されるようにする方法について説明していきます。ユニバースを作成する前に、結合の問題を解決する必要があります。

5.2 結合パスの問題

結合パスとは、結合によってリンクされたテーブルのデータにアクセスしてクエリを実行するための一連の結合です。

結合の問題は、リレーショナルデータベース内部でルックアップテーブルとファクトテーブルを関連付ける方法が制限されている場合に、しばしば発生します。スキーマのデザインで発生する結合パスの主要な問題は、次の3つです。

- ・ ループ
- ・ キャズムトラップ
- ・ ファントラップ

これらの問題はすべて、エイリアス(基になるテーブルのコピー)やコンテキスト(定義された結合パス)を作成し、クエリをメジャーとコンテキストに分離するユニバースデザインツールの機能を使用することで解決できます。

このセクションでは、ルックアップテーブルとファクトテーブルの定義を説明し、これらのテーブルで発生する結合パスの問題を解説します。さらに、エイリアス、コンテキスト、その他のユニバースデザインツールの機能を使用して、ユニバーススキーマの結合パスの問題を解決する方法について説明します。

ユニバースデザインツールでは通常、参照テーブルとファクトテーブル間に結合を作成します。

5.2.1 ルックアップテーブルとは

ルックアップ(またはディメンション)テーブルには、特定のアイテムや項目に関する情報が含まれます。たとえばルックアップテーブルには、顧客の名前や電話番号、年、国名などの、地理的観点に立った顧客情報が収められていることが考えられます。

ユニバースデザインツールでは、主に参照テーブルからディメンションオブジェクト、詳細オブジェクトを作成します。

5.2.2 ファクトテーブルとは

ファクトテーブルには、トランザクションに関する統計情報が収められます。たとえば、売り上げや利益などの数値が保存されます。

ユニバースではほとんどの場合、メジャーはファクトテーブルから定義します。

5.2.3 不正確な結果を返す結合パスの種類

リレーショナルデータベース内での結合方法が制限されていると、クエリは不正確な結果を返します。テーブルスキーマでのルックアップテーブルとファクトテーブルの関連付けの方法によって、クエリが返す行数が実際より少なかったり、多かったりする結合パスのインスタンスが作成されます。

次の種類の結合パスは、このような不正確な結果を返す恐れがあります。

結合パス	戻り値	説明
ループ	実際より少ない行数	ルックアップテーブル間に複数のパスがある結合
多対一結合の合流	実際より多い行数	ファクトテーブル間の複数の多対一結合が同じルックアップテーブルに合流する場合この問題は、キャズムトラップと呼ばれています。

結合パス	戻り値	説明
連続した多対一結合	実際より多い行数	一対多結合のリンク先のテーブルが、別の一対多結合でリンクされている場合一対多結合がこのように扇状に広がる問題は、ファントラップと呼ばれています。

5.2.4 結合問題の検出と解決

ユニバースデザインツールでは、結合の問題を検出し、解決するための機能が数多く提供されています。この節では、それぞれの機能を使い方を詳しく説明します。

結合パスの問題の検出と解決には、次の方法があります。

結合の問題	検出方法	解決方法
ループ	<ul style="list-style-type: none"> エイリアスの検出 コンテキストの検出 ループの検出 整合性のチェック テーブルスキーマを分析 	エイリアスとコンテキストを作成し、ループを解消する。
キャズムトラップ(多対一結合の合流)	テーブルスキーマを分析	<ul style="list-style-type: none"> コンテキストを作成する。 各メジャーに対して複数のSQL文を生成する。 複数のユニバースを作成する。(Web Intelligenceのみ)
ファントラップ(連続した多対一結合)	テーブルスキーマを分析	<ul style="list-style-type: none"> エイリアスを作成し、Aggregate Aware関数を適用する。 各メジャーに対して複数のSQL文を生成する。

ほとんどの結合の問題は、エイリアスカコンテキストを作成することで解決できます。ユニバースデザインツールのループ自動検出ツールを使ってスキーマ内のループを検出したり、コンテキストの自動検出機能を使用してキャズムトラップの発生個所を特定できます。ただし、ファントラップの解決にはスキーマを視覚分析し、手動でエイリアスあるいはコンテキストを作成する必要があります。

5.3 エイリアスの定義

エイリアスとは、スキーマ内にある既存のテーブルへの参照です。エイリアステーブルは、元のテーブル(ベーステーブル)の完全なコピーに別の名前を付けたものです。テーブル内のデータはベーステーブルとまったく同じですが、別の名前をつけることで、クエリのSQLでは2つの異なるテーブルとして扱われます。

Beach ユニバーススキーマには、Resort_Country (Country テーブルのエイリアス) と Sponsor (Customer テーブルのエイリアス) の 2 つのエイリアステーブルが含まれます。各エイリアステーブルについて、元のテーブルの名前がカッコ内に表示されます。

5.3.1 スキーマでのエイリアスの使い方

エイリアスを使う主な理由は、次の2つです。

- ・ クエリで同じテーブルを繰り返し指定できるようにする。ループやファントラップの解決を目的とした、エイリアスのもっとも重要な使い方です。サンプルのBeachユニバースには、CountryのエイリアスであるResort_Countryと、CustomerのエイリアスであるSponsorの2つのエイリアスがあります。
- ・ テーブル名を短縮し、SQL文を直接入力する際の手間を省く。

ヒント

エイリアスのもう一つの便利な使い方は、スキーマ内に挿入するすべてのテーブルにエイリアスを作成することです。元のベーステーブルをまとめてユニバース構造とは離れた場所に配置しておき、エイリアスだけでユニバースを作成します。これにより、テーブルにわかりやすい名前をつけることができるだけでなく、新たにベーステーブルのエイリアスが必要になった場合でも、ユニバースの主要構造を再作成する必要がなくなります。

5.3.1.1 エイリアスを使ったループの解決

ユニバース開発におけるエイリアスの最も一般的な使い方は、コモンテーブルでの潜在的なループを解決することです。ループとは、スキーマ内の複数のテーブルを経由しながら循環する一連の結合のことです。ループは、ルックアップテーブル間に複数の結合パスがある場合に発生します。

複数のクエリパスが使用するルックアップテーブルのエイリアスを作成することで、ループを解決できます。これに関する詳しい説明は、221 ページの「[ループの解決](#)」を参照してください。

5.3.1.2 エイリアスを使ったファントラップの解決

エイリアスは、潜在的なファントラップの解決にも使用されます。ファントラップとは、連続した複数の「一対多」結合が設定されている場合に、「多」側のテーブルで集計が実行されると不正確な結果を返すことをいいます。これに関する詳しい説明は、246 ページの [「キャズムトラップの解決」](#)を参照してください。

5.3.2 エイリアスの作成

エイリアスは手動で作成することも、結合パスのループを解消する可能性のあるエイリアスをユニバースデザインツールで自動検出することもできます。

ファントラップを解消するエイリアスを手動で作成する必要があります。エイリアスのみを使用し、ベーステーブルを使用しないスキーマを作成するときも、エイリアスを手動で作成します。

ループの解決を目的としたエイリアスの自動検出と作成に関する説明は、231 ページの [「エイリアスの検出と作成」](#)を参照してください。

5.3.2.1 エイリアスの手動作成

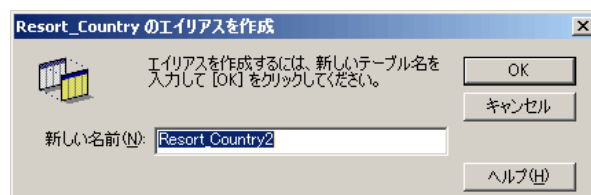
エイリアスを手動で作成する手順は、次の通りです。

- 1 エイリアスの作成に使用するテーブルをクリックします。
- 2 [挿入]>[エイリアス]の順に選択します。

または

[エイリアスの挿入]ボタンをクリックします。

[エイリアスを作成] ボックスが表示されます。新しいエイリアスの名前を入力します。



- 3 提案されるエイリアス名をそのまま使用するか、新しく名前を入力します。

注

入力するエイリアス名は、ベーステーブルと区別がつくようにエイリアスの役割に合った名前にしてください。たとえば、Country のエイリアスに Resort country という名前を付けます。Resort Country はリゾートのある国のデータを返すクエリに使用し、ベーステーブル Country は顧客の出身国のデータを返すクエリに使用します。

- 4 [OK]をクリックします。

構造枠にエイリアステーブルが表示されます。

- 5 スキーマのエイリアスと他のテーブル間で必要な結合を作成します。

ヒント

エイリアスとベーステーブルを混同しないよう、[ツール] > [オプション] > [グラフィック]をクリックし、[エイリアス名]チェックボックスを選択して、エイリアステーブルのタイトルにベーステーブルの名前を付けて表示することができます。

5.3.2.2 エイリアス名の変更

エイリアスの名前はいつでも変更できます。テーブルとエイリアスの命名方法は、RDBMS によって決まっています。エイリアス名は、テーブル名を直接変更するか、またはユニバースのエイリアスの一覧から変更できます。

エイリアス名の直接変更

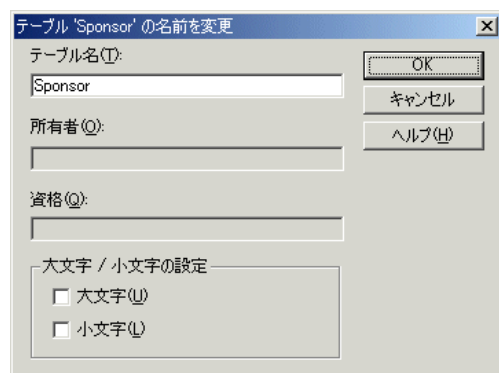
エイリアス名を直接変更する手順は、次の通りです。

- 1 テーブルをクリックして[編集]>[テーブル名の変更]の順に選択します。

または

テーブルを右クリックし、ポップアップメニューから[テーブル名の変更]を選択します。

[テーブル(テーブル名)の名前を変更]ダイアログボックスが開きます。



- 2 [テーブル名]ボックスに新しい名前を入力します。

[所有者] および [資格] ボックスは、データベースによって情報を入力できるかどうか異なります。アクティブな場合、必要に応じて情報を変更します。

- 3 エイリアス名をすべて大文字で表示する場合は、[大文字]チェックボックスをチェックします。

または

エイリアス名をすべて小文字で表示する場合は、[小文字]チェックボックスをチェックします。

- 4 [OK]をクリックします。

エイリアス名の一覧からの変更

エイリアス名を一覧から変更する手順は、次の通りです。

- 1 [ツール]>[エイリアス]の順に選択します。
- 2 [エイリアス一覧]ダイアログボックスが表示されます。ここには、作業中のユニバースにあるエイリアスがすべて表示されます。
- 3 変更するエイリアス名をクリックします。
- 4 新しい名前を[新しい名前]ボックスに入力します。
- 5 [適用]をクリックします。
- 6 [OK]をクリックします。

5.3.2.3 エイリアスの削除

エイリアスを削除するには、テーブルの削除と同じ操作を行います。エイリアスを削除する際は、そのエイリアスから作成したオブジェクトを削除するか、他のテーブルを使って定義し直す必要があります。

削除したエイリアスを参照するオブジェクトを変更せずに残しておく、そのオブジェクトを使ったクエリは Web Intelligence でエラーを発生します。

エイリアスを削除する手順は、次の通りです。

- 1 エイリアスをクリックし、[編集]>[消去]の順に選択します。

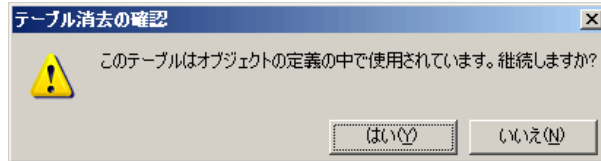
または

エイリアスを右クリックし、ポップアップメニューの[消去]コマンドを選択します。

または

エイリアスをクリックし、Delete キーを押します。

エイリアスを使っているオブジェクトがあると、次のメッセージが表示されます。



エイリアスを使っているオブジェクトがない場合は、メッセージは表示されずエイリアスはそのまま削除されます。

- 2 [はい]をクリックします。

指定したエイリアスが構造枠から削除されます。

5.4 コンテキストの定義

コンテキストとは、Web Intelligence に SQL を生成するための有効なクエリパスを提供する結合パスの集合です。

5.4.1 スキーマでのコンテキストの使い方

次の目的で、コンテキストをユニバーススキーマで使用できます。

- ・ ループの解消。
- ・ キャズムトラップの解消。
- ・ ファントラップ解消のヘルプ。
- ・ アグリゲート認識を使用した非互換オブジェクト検出のヘルプ。

5.4.1.1 コンテキストを使ったループの解決

コンテキストの最も一般的な使い方は、それぞれが他のファクトテーブルからのデータを返すようクエリパスを2つに分けることです。コンテキストは、複数のファクトテーブルを持ったスキーマで、結合パスを指定するために使用します。このようなスキーマには、エイリアスは適切ではありません。コンテキストの使い方についての説明は、221 ページの [「ループの解決」](#)を参照してください。

5.4.1.2 コンテキストを使ったキャズムトラップとファントラップの解決

コンテキストは、潜在的なキャズムトラップの解決にも使用されます。キャズムトラップは、2つの「多対一」結合が同じテーブルに合流すると発生します。同じディメンションに対して重複する行が返されるため、結果が増大します。コンテキストを使うと、ディメンションに対応した正しい行数を取得できるようクエリを分割できます。コンテキストをエイリアスと組み合わせ、ファントラップを解決することもできます。これに関する詳しい説明は、246 ページの「[キャズムトラップの解決](#)」を参照してください。

5.4.1.3 コンテキストを使ったアグリゲート認識の互換性確保

コンテキストを使用して、@AggregateAware 関数を使って定義されたオブジェクトと互換性のないオブジェクトがクエリに組み込まれるのを防ぐことができます。

5.4.2 コンテキストの作成

コンテキストは手動でも作成できますし、ユニバースデザインツールで自動検出することもできます。

コンテキストをループまたはキャズムトラップの解決に使用する場合は、必ずユニバースデザインツールのコンテキストの自動検出機能を使用する必要があります。一方、ファントラップの解決の使う場合は通常、コンテキストを手動で作成する必要があります。

ループの解決を目的としたコンテキストの自動検出についての説明は、221 ページの「[ループの解決](#)」を参照してください。

注

1 つまたは複数のコンテキストを作成する場合、すべての結合をそれらのコンテキストに含める必要があります。コンテキストに含まれていない結合でテーブルがリンクされている場合、クエリを実行する時にその結合は含まれません。

ここでは、コンテキストを自動で作成する方法と、手動で作成する方法を紹介します。

5.4.2.1 コンテキストの自動作成

コンテキストを自動作成する手順は、次の通りです。

- 1 [ツール]>[自動検出]>[コンテキストの検出]の順に選択します。

[コンテキストの候補]ダイアログボックスが開きます。スキーマのコンテキストの候補が表示されます。これらのコンテキストの候補はループまたはキャズムトラップを解消する必要がある場合があります。これは、キャズムトラップは 2 つのコンテキストが交わる部分に発生するからです。



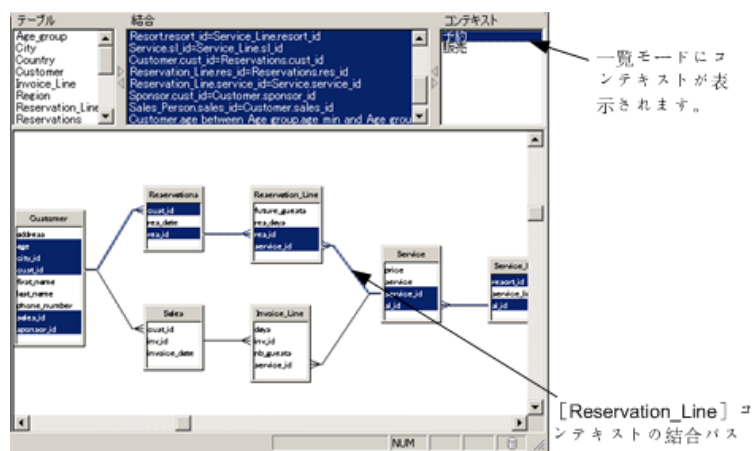
- 2 [コンテキストの検出]一覧でコンテキストをクリックし、[追加]ボタンをクリックします。
- 3 一覧にある各コンテキストの候補に手順 2 を繰り返します。

注

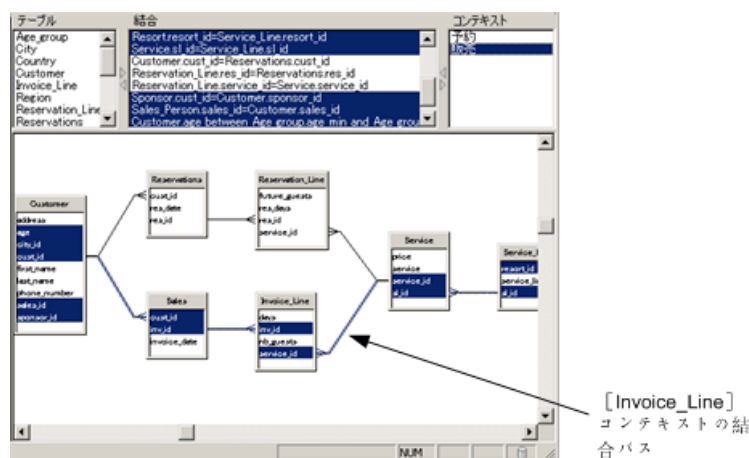
[コンテキストの候補]から[コンテキストの一覧]に追加したコンテキストの名前を変更する場合は、[コンテキストの一覧]でそのコンテキストをクリックし、[名前の変更]ボタンをクリックします。[コンテキスト名の変更]ボックスが表示されます。新しい名前を入力し、[OK]をクリックします。

- 4 [OK]をクリックします。

コンテキストは、一覧モード ([表示]>[一覧モード]) のコンテキスト枠に一覧表示されます。[予約]コンテキストは、次の通りです。



- 5 [販売]コンテキストは、次の通りです。



5.4.2.2 コンテキストの手動作成

コンテキストを手動作成する手順は、次の通りです。

- 1 [挿入]>[コンテキスト]の順に選択します。

または

[コンテキストの挿入]ボタンをクリックします。

[新規コンテキスト]ダイアログボックスが開きます。

The 'New Context' dialog box has the following fields and buttons:

- Context Name (N):
- Current Contexts List (L):
- Selection items only (S):
- Description (P):
- Find (D):
- Check (C):
- OK: [OK]
- Cancel: [キャンセル]
- Help (H): [ヘルプ(H)]

- 2 [コンテキスト名]ボックスに、コンテキストの名前を入力します。
- 3 コンテキストを定義するすべての結合を[現在のコンテキストの結合一覧]ボックスから選択します。

作成時には、次のオプションを使用できます。

- 4 [検出]ボタンをクリックして、コンテキストに含まれるすべての結合を反転表示する。
- 5 [選択項目のみ表示]を選択すると、選択された結合のみが表示されます。
- 6 [チェック]ボタンをクリックする。

ユニバースデザインツールによって、ループに対し選択された結合がチェックされます。

- 7 コンテキストから返されるデータの説明を入力します。このテキストは、コンテキストパスを含んだクエリを実行すると Web Intelligence ユーザに表示されるヘルプテキストになります。エンドユーザにもわかりやすい説明内容としてください。
- 8 [OK]をクリックします。

コンテキストが作成されました。

5.4.3 コンテキストの編集

コンテキストエディタを使用して、次のコンテキストのプロパティを編集できます。

- ・ 名前
- ・ コンテキストに含まれる結合
- ・ 説明

コンテキストに未解決のループがないかをチェックすることもできます。

5.4.3.1 コンテキストのプロパティの編集

コンテキストのプロパティを編集する手順は、次の通りです。

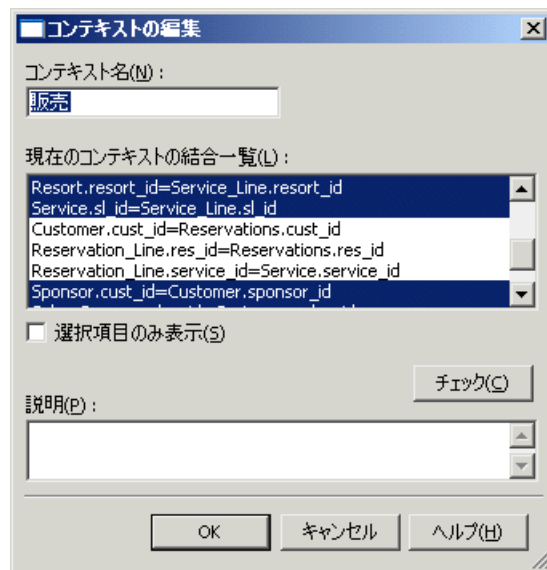
- 1 構造枠を一覧モードで表示していない場合は、[表示]>[一覧モード]の順に選択します。

構造枠の上に、一覧枠が表示されます。構造枠にあるすべてのテーブル、結合、コンテキストが一覧表示されます。

テーブル	結合	コンテキスト
Age_group	City.city_id=Customer.city_id	販売
City	City.region_id=Region.region_id	予約
Country	Country.country_id=Region.country_id	
Customer	Resort_Country.country_id=Resort.country_id	
Invoice_Line	Customer.cust_id=Sales.cust_id	
Region	Sales.inv_id=Invoice_Line.inv_id	
Reservation_Line	Invoice_Line.service_id=Service.service_id	

- 2 [コンテキスト]一覧枠から編集するコンテキストをダブルクリックします。

[コンテキストの編集]ダイアログボックスが表示されます。



- 3 コンテキスト名を変更する場合は、[コンテキスト名]ボックスに新しい名前を入力します。
- 4 [現在のコンテキストの結合一覧]リストでは、反転表示されている結合をクリックすると、その結合がコンテキストから除外されます。
または
逆に、通常表示の結合をクリックすると、コンテキストに追加されます。
- 5 コンテキストの説明を入力します。
- 6 [OK]をクリックします。
コンテキストへの変更が表示されます。

5.4.4 コンテキストの削除

コンテキストの一覧枠を使い、いつでもコンテキストを削除できます。コンテキストに含まれるテーブルや結合を削除したり、新しいテーブルや結合を追加する場合は、テーブルや結合を変更する前にコンテキストを削除してください。

変更が終了したら、キャズムトラップ解消用のコンテキストは手動で、ループ解消用のコンテキストはコンテキストの自動検出機能を使って作成します。詳細は、233 ページの「[コンテキストの検出と作成](#)」を参照してください。

5.4.4.1 コンテキストの一覧からのコンテキストの削除

コンテキストの一覧からコンテキストを削除する手順は、次の通りです。

- 1 構造枠を一覧モードで表示していない場合は、[表示]>[一覧モード]の順に選択して一覧モードに切り替えます。
- 2 コンテキスト一覧から削除するコンテキストを右クリックし、ポップアップメニューから[消去]を選択します。

または

コンテキストをクリックして選択し、[編集]>[消去]の順に選択します。

コンテキストが一覧から削除されます。

5.4.5 コンテキストの更新

ユニバース構造が変更されても、コンテキストは自動的に更新されません。テーブルや結合を追加、削除した場合は、すべてのコンテキストを更新する必要があります。

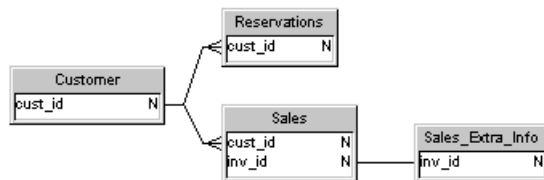
単純な変更であれば、[コンテキストの編集]ダイアログボックスやコンテキストの一覧枠を使って、それぞれのコンテキストに含まれる結合を手動で変更できます。一方、ユニバース構造を大きく変更した場合は、コンテキストを削除し、作成しなおす必要があります。

5.4.6 コンテキストの検出を妨げる結合パス

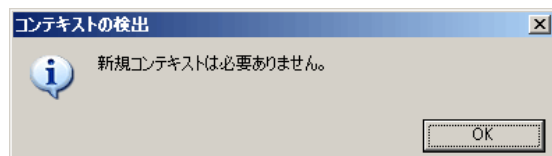
結合パスの末尾に一对一のカーディナリティがあると、ユニバースデザインツールでコンテキストを検出できない場合があります。この問題は、結合パス両端のテーブルのカーディナリティを「一对多」に設定することで解決できます。

例 コンテキストの検出を妨げる一对一のカーディナリティ

下のスキーマには、それぞれの販売実績に固有の情報を記録したテーブルSales_Extra_Infoが含まれています。このテーブルは、一对一の結合でSalesテーブルにリンクされています。



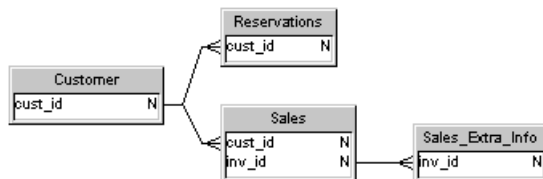
この結合パスをよく見ると、このスキーマにはReservationsとSalesの2つのコンテキストがあることがわかります。しかし、[ツール]>[自動検出]>[コンテキストの検出]の順に選択しても、次のメッセージが表示されます。



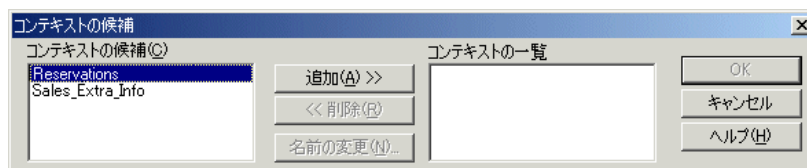
ユニバースデザインツールでは結合パスの末尾にある一対一結合は考慮されないため、2つのコンテキストがあることを検出できません。

5.4.6.1 コンテキストの検出を可能にするカーディナリティの変更

この問題を解決するには、Sales_Extra_InfoからSalesへの結合のカーディナリティを一対多に変更します。変更後のカーディナリティは、多対一でも構いません。重要なのは、結合パスが一対一以外のカーディナリティで終わっていることです。次のスキーマは、結合パスを一対多の結合で終わらせたものです。



このスキーマでコンテキストを検出すると、次のように2つのコンテキストが検出されます。



5.4.7 コンテキストがクエリに及ぼす影響

Web Intelligence ユーザに許可されたスキーマ構造上のオブジェクト定義によりますが、コンテキストによって次の 3 種類のクエリが実行されます。

- ・ 不明瞭なクエリ
- ・ 推量可能なクエリ
- ・ 非互換クエリ

Web Intelligence でこれらの種類のクエリを実行して、コンテキストによって生成される SQL をテストできます。エラーが発生したり、不正確な結果が返される場合は、関連する結合パスを分析する必要があります。

5.4.7.1 不明瞭なクエリ




エンドユーザに、どのクエリパスを使用するかを問い合わせます。このプロンプトは、どのコンテキストを使用するかを判断できないオブジェクトの組み合わせが選択されると、表示されます。

不明瞭なクエリでは、Web Intelligence によってダイアログボックスが表示され、ユーザに 2 つのコンテキストのどちらを選択するか質問します。ユーザの選択したコンテキストにしたがって、対応するテーブルと結合が SQL に追加されます。

例 不明瞭なクエリの実行

Web Intelligence ユーザが、次のクエリを実行したとします。

リゾート別年齢層別の利用サービス

 サービス  年齢層  リゾート

クエリが実行されると、ダイアログボックスが開いてユーザにコンテキストの選択を質問しますが、この場合のコンテキストは[予約]または[販売]になります。

ユーザは、年齢層別の予約サービスまたは販売サービスのどちらの情報を取得するかを選択します。[予約]コンテキストを選択した場合、次の SQL が生成されます。

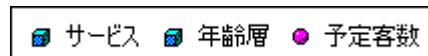
```
SELECT Service.service, Age_group.age_range, Resort.resort FROM Service, Age_group, Resort, Customer, Reservations, Reservation_Line, Service_Line
WHERE ( Resort.resort_id=Service_Line.resort_id ) AND ( Service.sl_id=Service_Line.sl_id ) AND ( Customer.age between Age_group.age_min and
Age_group.age_max ) AND ( Customer.cust_id=Reservations.cust_id ) AND ( Reservation_Line.res_id=Reservations.res_id ) AND ( Reservation_Line.ser
vice_id=Service.service_id )
```

もう1つのコンテキスト(販売)の結合パスは、このSQLには含まれていません。

5.4.7.2 推量可能なクエリ

Web Intelligence では、コンテキストの選択を要求するダイアログボックスが表示されずにクエリが実行されます。これは、オブジェクトの組み合わせに使用すべきコンテキストを判断するに足る十分な情報が含まれているためです。たとえば、エンドユーザが次のクエリを実行したとします。

サービス別年齢層別の予定客数



クエリを実行すると、コンテキストの選択を要求するダイアログボックスが表示されずにデータが返されます。[予定客数] オブジェクトは、[予約]コンテキストの一部である Reservation_Line テーブルの合計です。Web Intelligence では、クエリで [予約] コンテキストを使用するものとみなされます。

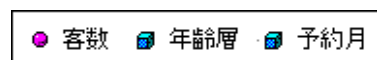
5.4.7.3 非互換クエリ

これは、異なるコンテキストに含まれるオブジェクトをクエリ内で組み合わせた場合です。2つのSELECT文が同期され、クエリ結果が異なるテーブルに表示されます。

例 互換性がないクエリの実行

Web Intelligence ユーザが、次のクエリを実行したとします。

年齢層別の滞在客数合計とその予約月



このクエリを実行すると、「販売」と「予約」の両方のコンテキストが使われるため、Web Intelligenceはコンテキストを問い合わせずにクエリを実行します。両方のコンテキストのSELECT文が次のように同期されます。

```
SELECT
  Age_group.age_range,
  sum(Invoice_Line.nb_guests)
FROM
  Age_group,
  Invoice_Line,
  Service_Line,
  Sales,
  Customer,
  Service
WHERE
  ( Customer.cust_id=Sales.cust_id )
  AND ( Invoice_Line.inv_id=Sales.inv_id )
  AND ( Invoice_Line.service_id=Service.service_id )
  AND ( Service.sl_id=Service_Line.sl_id )
  AND ( Customer.age between Age_group.age_min and Age_group.age_max )
  AND ( Service_Line.service_line = 'Accommodation' )
GROUP BY
  Age_group.age_range
```

このクエリは、次のように分割されています。

- ・ [年齢層] と [客数]
- ・ [予約月]

2つのクエリの結果が返されると、Web Intelligence は [年齢層] を使って結果を組み合わせ、同じレポート内の 2 つのテーブルに、次のように結果が表示されます。

18-30

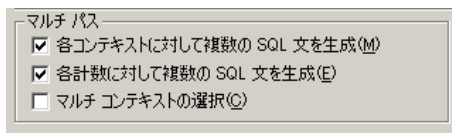
客数	予約月
451.00	Apr
	Feb
	Jan
	Jun
	May
	Nov

Web Intelligence で非互換クエリを実行するには、ユニバースデザインツールで [各コンテキストに対して複数の SQL 文を生成] オプションを選択する必要があります。次のセクションで、その方法を説明します。

5.4.7.4 [各コンテキストに対して複数の SQL 文を生成]オプションの選択

[各コンテキストに対して複数の SQL 文を生成]オプションを選択する手順は、次の通りです。

- 1 [ファイル]>[パラメータ]を選択します。
[ユニバースのパラメータ]ダイアログボックスが表示されます。
- 2 [SQL]タブをクリックします。
[SQL]ページが表示されます。
- 3 [各コンテキストに対して複数の SQL 文を生成]チェックボックスをオンにします。



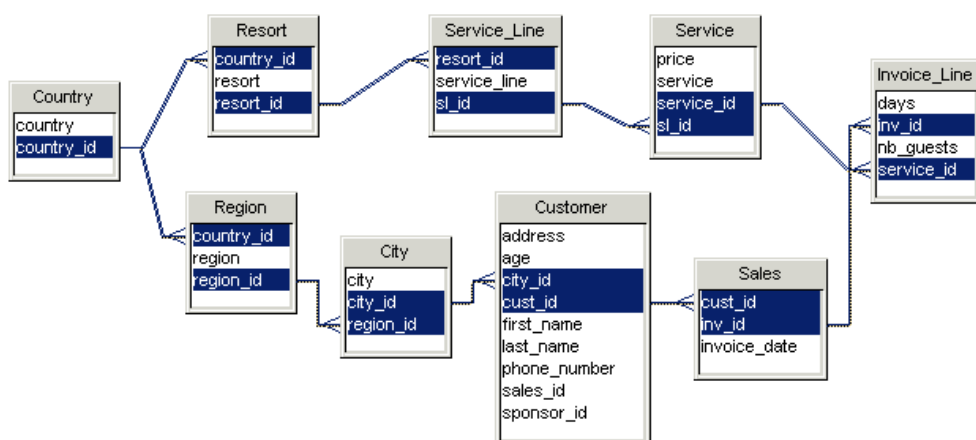
- 4 [OK]をクリックします。

5.5 ループの解決

リレーショナルデータベースでは、実際より少ない行数を返す結合パスを一般的にループと呼んでいます。

5.5.1 ループとは

ループとは、スキーマ内の複数のテーブルを経由しながら循環する一連の結合のことです。ループは、ルックアップテーブル間に複数の結合パスがある場合に発生します。ループは、スキーマでは次のように表示されます。



このスキーマには、リンクされた2種類の情報が含まれています。

対象	リンクされている情報
リゾート	利用できるサービス部門、各サービス部門のサービス、各サービスの請求内容、リゾートの所在国
顧客	顧客の都市、地域、国と顧客に対する売上げ、各販売取引の請求内容

この2種類の情報が、ループを形成する共通結合パスでリンクされています。ルックアップテーブルCountryは、リゾートの所在国と顧客の国の両方の情報を提供しています。

5.5.1.1 データベースではなく、ユニバーススキーマでループが発生する理由

データベースでは、テーブル間に複数の結合パスがあることは必ずしも誤りではなく、特定のユーザーニーズを満たすためにはむしろ必要な場合もあります。それぞれの結合パスをクエリ内で指定した場合は、それに合った異なる結果が得られます。

ただし、ユニバースデザインツールで作成するスキーマでは、リレーショナルデータベースでは扱えない、複数のパスを含んだクエリの使用を許可する必要がある場合が多いため、不正確な結果が返されることがあります。

返された行は、それぞれのパスを使って得られる結果のインターセクト部分になり、結果の行数は実際よりも少なくなります。このような結果を解析しても、満足な結果は得られません。

5.5.2 ループによるクエリへの影響

前述の構造を元にユニバースを作成すると、ループ内のテーブルに対して実行するクエリはすべて、リゾートの所在国と顧客の国籍が一致するデータだけを返します。共有ルックアップテーブルのCountryに対する二重制約により、実際よりも少ない行のデータしか取得できません。

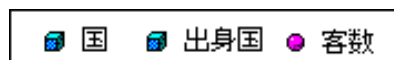
例 ループにより返される不正な結果

上記のループを含んだスキーマを使って、次のオブジェクトを作成します。



Web Intelligence からこのユニバースに、次のようなクエリを実行します。

それぞれのリゾートの所在国について、顧客の国籍別の滞在者数



ユーザーが期待する結果は、次のようなデータです。

France

出身国	客数
Germany	141.00
Japan	154.00
US	151.00

US

出身国	客数
Germany	329.00
Japan	345.00
US	431.00

リゾートのあるFranceとUSについて、ドイツ人、日本人、アメリカ人の滞在者数が表示されるはずでした。ところが、ループを含んだユニバースにクエリを実行すると、次の結果が返されます。

国	出身国	客数
US	US	431.00

これでは、USからの滞在客だけがUSのリゾートだけに滞在したことになり、別の国籍を持つ滞在客は、無視されてしまいます。

5.5.2.1 ループにより作成されるクエリ

構造枠の結合は、クエリのSQL内のWHERE句の作成に使用されます。結合の目的は、クエリが返すデータを制限することです。ループがあると、想定した以上の制限が適用され、返されるデータが正しくなくなります。

このループが作成するWHERE句は、次の通りです。

```
WHERE ( Country.country_id=Resort.country_id ) AND ( Resort.resort_id=Service_Line.resort_id ) AND ( Service_Line.sl_id=Service.sl_id ) AND ( Service.service_id=Invoice_Line.service_id ) AND ( Sales.inv_id=Invoice_Line.inv_id ) AND ( Customer.cust_id=Sales.cust_id ) AND ( City.city_id=Customer.city_id ) AND ( Region.region_id=City.region_id ) AND ( Country.country_id=Region.country_id ) AND ( Service_Line.service_line = 'Accommodation' )
```

次の2つの結合は、両方ともCountryテーブルの値を制限しています。

- ・ Country.country_id=Resort.country_id
- ・ Country.country_id=Region.country_id

Countryテーブルには、次の2つの用途があります。

- ・ リゾート所在国のルックアップ
- ・ 顧客の国籍のルックアップ

これによって、リゾートの国が顧客の国に一致するデータだけを取得するような制限が適用されてしまいます。結果としてレポートには、USのリゾートに滞在したUSからの顧客だけが滞在客数として表示されます。

ループの種類に従い、ユニバースデザインツールでループを解決できます。エイリアスを使って結合パスを分断するか、クエリでどちらか 1 つのパスだけが使用されるようコンテキストを使って結合パスを分離します。

5.5.2.2 エイリアスによるループの分断

エイリアスは、1つのクエリで同じテーブルを違う目的で使用することで、ループを分断します。エイリアスとは、ベーステーブルと同じテーブルに別の名前をつけたものです。エイリアスに含まれるデータはもとのテーブルのデータと同一ですが、別の名前を付けることでクエリのSQLでは2つの異なるテーブルとして扱われます。

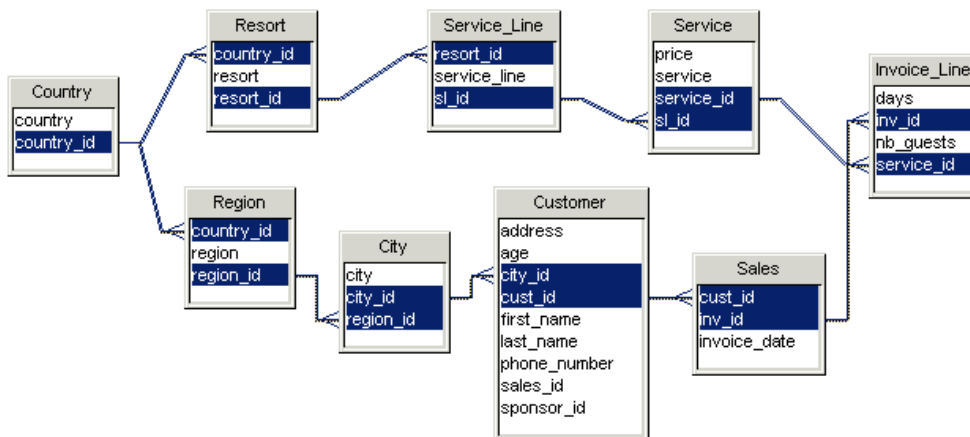
注

ここで取り上げる例のように、エイリアスを1つ作成するだけでループを解決できる場合もあります。Resort結合には本来のCountryテーブルを使用し、Region結合にはエイリアステーブルを使用します。また、ベーステーブルのそれぞれの結合についてエイリアスを作成することもできます。リレーショナルデータベースによっては、これが必要となるものもあります。

例 エイリアスでループを分断する

次のスキーマは、前のセクションで説明したループを含んでいるスキーマです。ルックアップテーブル Country は 2 つの結合の "1" 側だけでリンクされ、次の 2 つの目的で使われています。

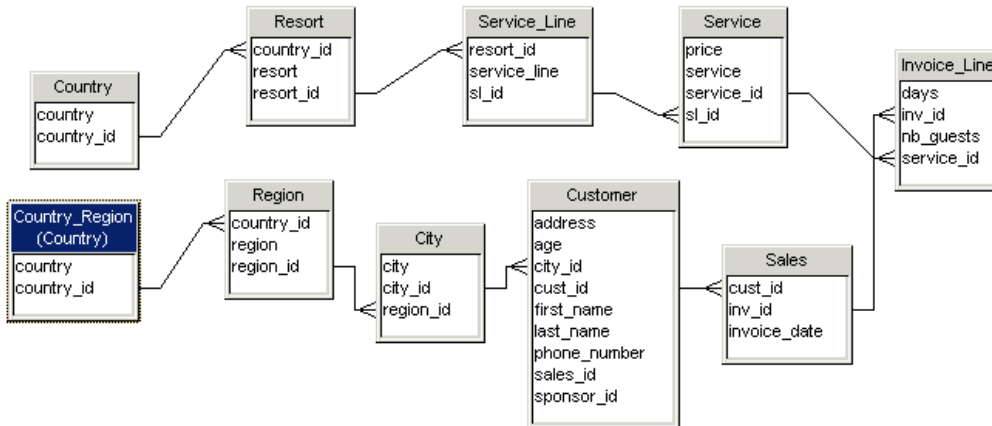
- ・ リゾートの所在国
- ・ 顧客の国



Countryテーブルのエイリアスを作成し、Country_Regionの名前を付けます。これで 2 つの "1" 側の結合を 2 つに分割できます。




- ・ CountryからResortテーブルの結合はそのまま
- ・ Country_RegionからRegionテーブルへの結合を作成

これで、スキーマが次のようになります。



前述の例で、実際よりも少ない行数を返したクエリを実行してみます。

それぞれのリゾートの所在国について、顧客の国籍別の滞在者数

	国		出身国		客数
---	---	---	-----	---	----

このクエリのWHERE句は、次の通りです。

```
WHERE ( City.city_id=Customer.city_id ) AND ( City.region_id=Region.region_id ) AND ( Country.country_id=Region.country_id ) AND ( Resort_Country.country_id=Resort.country_id ) AND ( Customer.cust_id=Sales.cust_id ) AND ( Invoice_Line.inv_id=Sales.inv_id ) AND ( Invoice_Line.service_id=Service.service_id ) AND ( Resort.resort_id=Service_Line.resort_id ) AND ( Service.sl_id=Service_Line.sl_id ) AND ( Service_Line.service_line = 'Accommodation' )
```

ここではCountryテーブルを制限する結合が1つ、Resort_Countryテーブルを制限する結合が1つになり、ループが分断されています。

クエリを実行すると、次のテーブルが返されます。

国	出身国	客数
France	Germany	141.00
France	Japan	154.00
France	US	151.00
US	Germany	329.00
US	Japan	345.00
US	US	431.00

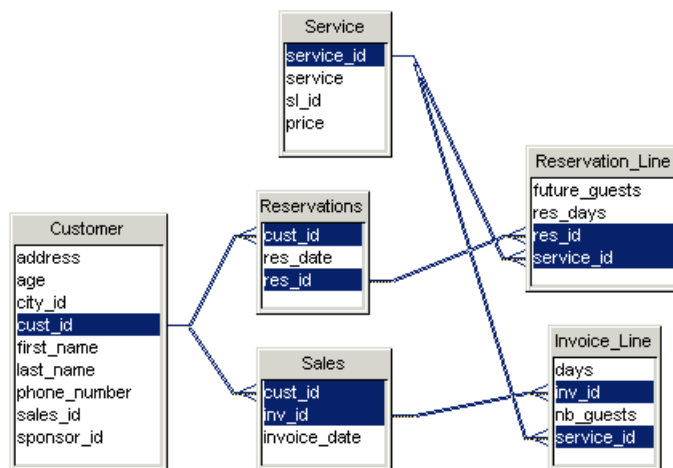
5.5.2.3 コンテキストによるループの解決

コンテキストは、ループ内のパスを特定の経路に限定する一連の結合を定義することで、ループを解決します。これにより、同じSQLクエリ内には異なるパスを使った結合は含まれなくなります。

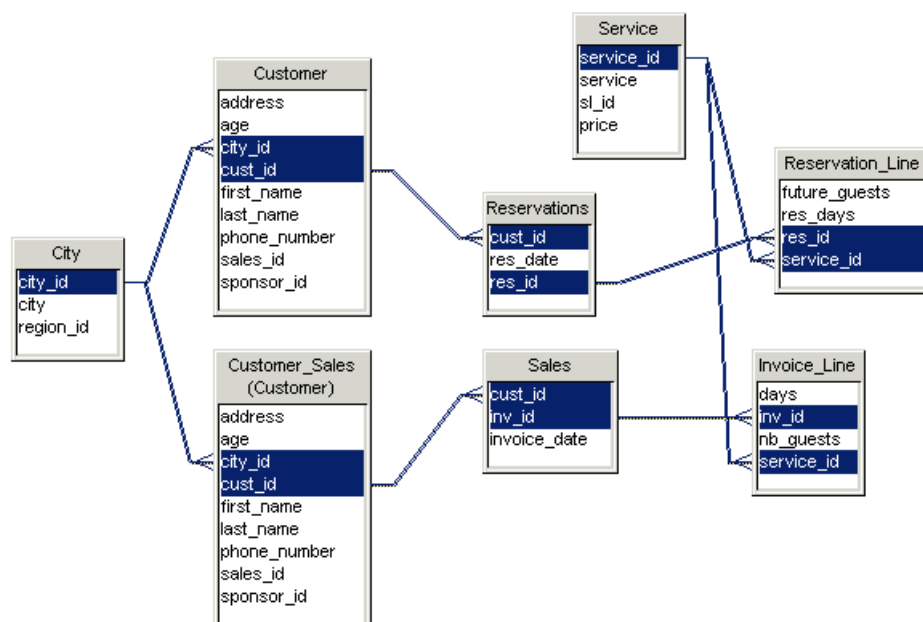
コンテキストは、ルックアップテーブルを共有している複数のファクトテーブル(マルチスター型)を含むスキーマで頻繁に使用されます。

例 コンテキストを使ってループを解決する

次のスキーマには、予約と販売実績に関する統計データが含まれています。トランザクションの種類別の統計データは、ファクトテーブルSalesとReservationsに保存されています。販売パスと予約パスの両方からサービス情報を引き出せるよう、スキーマにはループが含まれています。



Customerテーブルのエイリアスを作成し、CustomerからReservationへの結合と、Customer_SalesからSalesへの結合でループを分断することはできますが、スキーマにCityテーブルを追加するとループが再現されます。



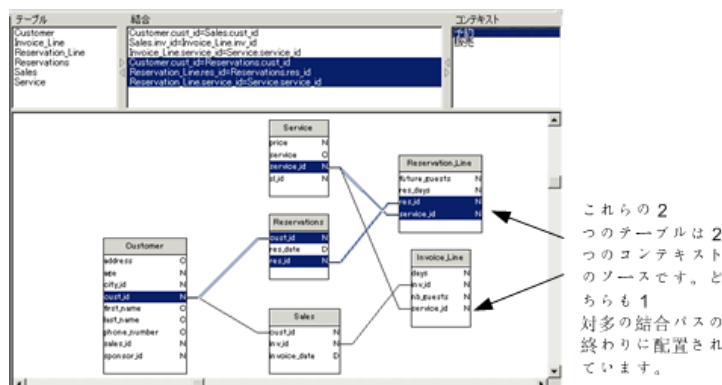
これでは、スキーマにテーブルを追加するたびにエイリアスを作成しなければならなくなりますので、メンテナンスも困難となり、それぞれのテーブルを使用する似通ったオブジェクトの数が増殖してしまいます。

このループを解決する唯一の方法は、ループをそのまま残し、スキーマ内で使用するパスを定義したコンテキストを作成することです。これにより、販売と予約のどちらの視点から顧客の情報が必要なのか、という疑問に答えられるクエリを実行できます。

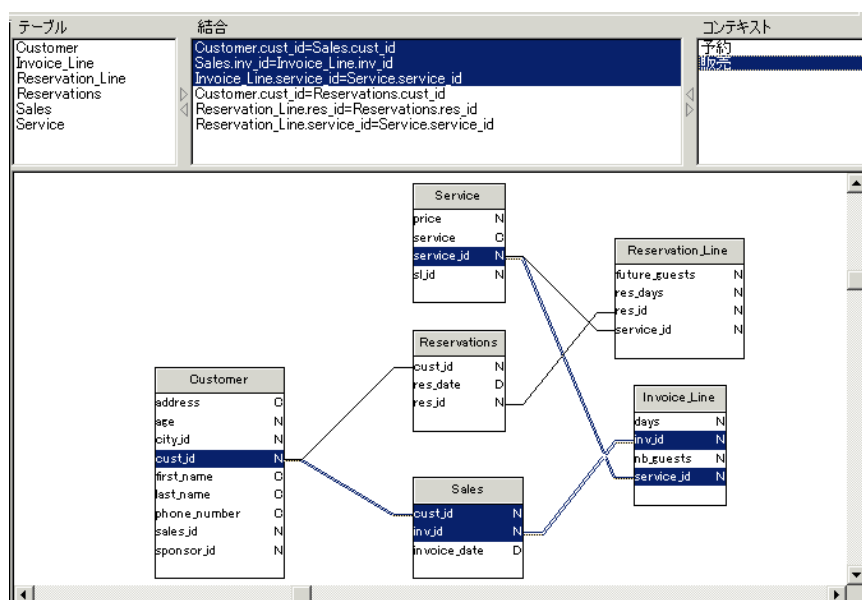
この例では、CustomerテーブルからServiceテーブルへのパスは2つあります。

パス	ユニバースデザインツールで、これらのコンテキストが検出されます。
ReservationsとReservation_Line	Reservation_Line
SalesとInvoice_Line	Invoice_Line

予約コンテキストは次の通りです。



販売コンテキストは次の通りです。



異なるコンテキストで使用する一連のオブジェクトをテーブルから作成します。ユーザーが選択したオブジェクトにしたがい、予約クエリまたは販売クエリが自動的に実行されます。

5.5.3 ループの識別の目安

次のガイドラインを元にスキーマを分析すると、ループの解決にエイリアスとコンテキストのどちらが適切かを判断する目安となります。この分析はスキーマの理解に役立ちますが、正式なループの識別と解決には、[ソー

ル]メニューの[エイリアスと検出]コマンドと[コンテキストの検出]コマンドを使ってください。詳しい説明は、231 ページの「[エイリアスの検出と作成](#)」と233 ページの「[コンテキストの検出と作成](#)」とを参照してください。

ループに含まれるテーブル	ループの解決方法
ルックアップテーブルが1つだけ	エイリアス
複数の結合の"1"側で終わるルックアップテーブル	エイリアス
複数のファクトテーブル	コンテキスト

5.5.4 ループの自動的な識別と解決

ユニバースデザインツールを使ってループを自動検出し、提案されるエイリアスとコンテキストの候補をスキーマに挿入してループを解決することができます。

5.5.4.1 ループ検出の前にカーディナリティを設定

ループの自動検出と解決機能を使う前に、スキーマ内のすべての結合のカーディナリティを設定しておく必要があります。

設計時には、カーディナリティを手動で定義するか、ユニバースデザインツールが自動検出したカーディナリティを手動で検証することをお勧めします。

カーディナリティは、次の2種類の方法で設定できます。

- ・ 手動: 詳細については、187 ページの「[カーディナリティの使い方](#)」を参照してください。
- ・ カーディナリティの自動検出: 詳細については、187 ページの「[カーディナリティの使い方](#)」を参照してください。

5.5.5 ループの検出と解決のための ユニバースデザインツールの機能

以下のユニバースデザインツール機能を使用して、ループを検出し解決できます。

ループの識別と解決方法	説明
エイリアスの検出	<p>エイリアスによって構造内のループを解決できるテーブルを検出し、各テーブルのエイリアス候補を提案します。表示される候補から直接エイリアスを挿入し、エイリアス名を変更できます。</p> <p>作成するエイリアスがコンテキストに含まれるよう、コンテキストを検出する前にエイリアスを検出する必要があります。</p> <p>ファントラップの解決に必要なエイリアスは検出できません。</p>
コンテキストの検出	<p>構造内のループを解決できるコンテキストを検出し、コンテキスト候補を提案します。表示される候補から直接コンテキストを挿入し、コンテキスト名を変更できます。</p> <p>新規エイリアスがコンテキストに含まれるよう、エイリアスを検出してからコンテキストを検出する必要があります。</p> <p>キャズムトラップの解決に必要なコンテキストは検出できない場合があります。検出できない場合は、手動でコンテキストを識別する必要があります。</p>
ループの検出	<p>構造内のループを検出し、反転表示します。ループの解決に必要なエイリアスまたはコンテキストを提案します。提案されたエイリアスまたはコンテキストは、[ループの検出]ダイアログボックスから直接挿入できます。</p> <p>ループの検出は、スキーマのクイックチェックやループの視覚確認に使用してください。エイリアス候補は挿入前に編集したり、目視確認できませんので、ループの解決には使用しないことをお勧めします。</p>

5.5.5.1 一般的なループ識別方法と解決方法

ここでは、一般的なループの検出と解決方法を紹介します。それぞれのセクションでは、各ステップの詳しい内容も説明します。

- 1 すべてのカーディナリティが設定されていることを確認します。
詳しくは、187 ページの「[カーディナリティの使い方](#)」を参照してください。
- 2 エイリアスを自動検出し、ループの解決にエイリアスが必要かどうかをチェックします。
詳しい説明は、231 ページの「[エイリアスの検出と作成](#)」を参照してください。
- 3 [エイリアスの候補]ダイアログボックスに表示されるエイリアス候補を挿入します。
- 4 コンテキストを自動検出し、エイリアスだけでは解決できないループの解決にコンテキストが必要かどうかをチェックします。
詳しい説明は、233 ページの「[コンテキストの検出と作成](#)」を参照してください。
- 5 [コンテキストの候補]ダイアログボックスに表示されるコンテキスト候補を作成します。
- 6 オブジェクトを作成し、クエリを実行してループが解決されたか確認します。
オブジェクトの作成とユニバース構造のテストの詳細については、267 ページの「[ユニバースの作成](#)」の章を参照してください。

注

エイリアスを作成したテーブルからすでにオブジェクトが作成されている場合、ベーステーブルではなくエイリアスを使ってオブジェクトを再定義する必要があります。

5.5.5.2 エイリアスの検出と作成

[エイリアスの検出]オプションを使って、アクティブユニバースでループの原因となるテーブルを自動検出することができます。提案されるエイリアスの候補は、その場で編集、挿入することができます。

注

[エイリアスの検出]オプションを使用する前に、スキーマのテーブルがすべて結合でリンクされ、カーディナリティが設定されていることを確認してください。

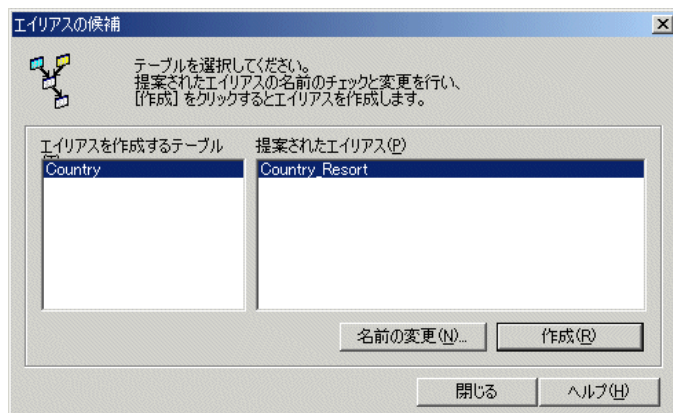
エイリアスを検出し、作成する方法

- 1 [ツール] > [自動検出] > [エイリアスの検出]の順に選択します。

または

[エイリアスの検出]ボタンをクリックします。

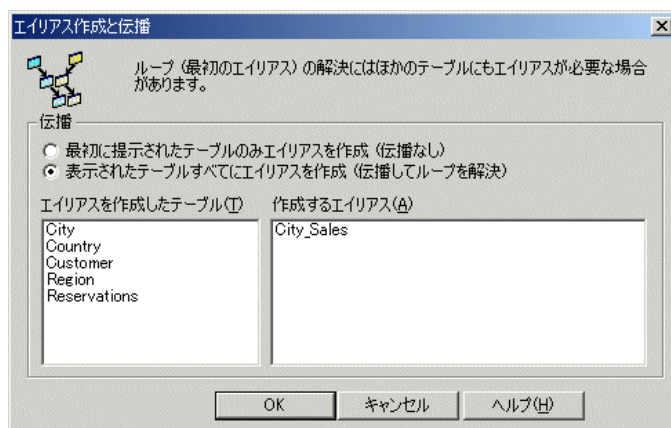
[エイリアスの候補]ダイアログボックスが開きます。左の枠に、エイリアスの必要なテーブル (複数可) が表示されます。右の枠に、ループを分割するために挿入できるエイリアスが表示されます。



- 2 左枠からエイリアスを作成するテーブルを選択します。
右の枠に候補エイリアスの名前が表示されます。
- 3 名前を変更するには、[名前の変更]ボタンをクリックし、[名前の変更]ボックスに新しい名前を入力します。
- 4 [作成]をクリックします。
エイリアスの作成を確認するプロンプトが表示されます。
- 5 [OK]をクリックします。
構造枠にエイリアスが表示されます。
- 6 残りのテーブルに手順 2 から 5 までを繰り返します。
- 7 [閉じる]をクリックします。

5.5.5.3 複数のエイリアスの検出と作成

エイリアスを作成すると、新しい結合パスに対応した別のエイリアスが必要となることがあります。[エイリアスの検出]を使用する際、さらに別のエイリアスが必要になった場合、ユニバースデザインツールで[作成]ボタンをクリックすると次のダイアログボックスが表示されます。



ここでは、次の 2 つのオプションを利用できます。

- ・ 提案された最初のテーブルだけエイリアスを作成する。
- ・ 表示されたテーブルすべてにエイリアスを作成する。

5.5.5.4 コンテキストの検出と作成

コンテキストの自動検出で、コンテキストが必要かどうかをチェックできます。提案されるコンテキストの候補は、その場で編集、挿入することができます。

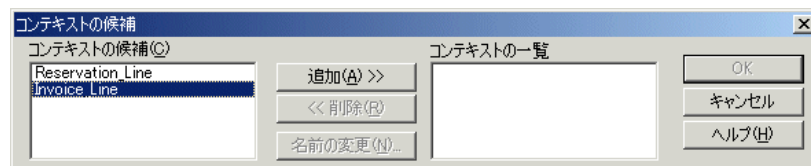
コンテキストを検出し、作成する

- 1 [ツール]>[自動検出]>[コンテキストの検出]の順に選択します。

または

[コンテキストの検出]ボタンをクリックします。

[コンテキストの候補]ダイアログボックスが開きます。左枠には、コンテキストの候補が一覧表示されます。



- 2 左枠から作成するコンテキストをクリックします。

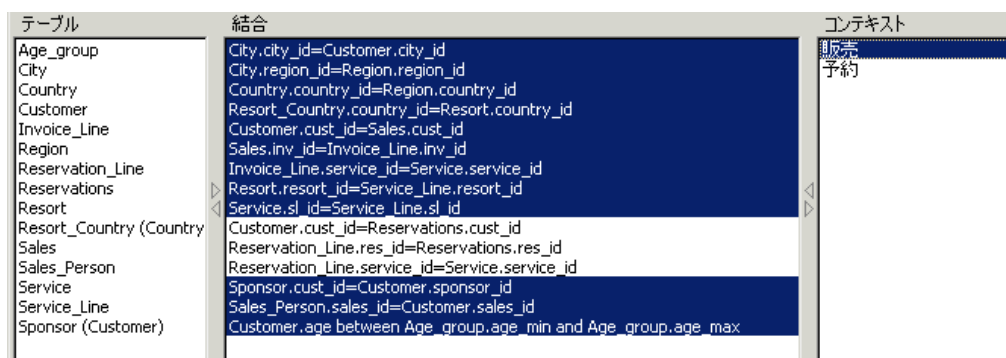
選択したコンテキストに含まれるすべてのテーブルが、スキーマ内で反転表示されます。

- 3 [追加]ボタンをクリックします。

選択したコンテキストが右枠の[コンテキストの一覧]に移動します。右枠からコンテキストを削除するには、コンテキストを選択して[削除]をクリックします。

- 4 作成するコンテキストを追加する場合は、ステップ3と4を繰り返します。
- 5 コンテキストの名前を変更するには、右枠からコンテキストを選択して[名前の変更]ボタンをクリックします。
[コンテキスト(コンテキスト名)の名前を変更]ダイアログボックスが開きます。名前を入力します。
- 6 [OK]をクリックします。

作成したコンテキストが、ユニバースウィンドウのコンテキストボックスに表示されます。



注

ユーザにとってあいまいなループがユニバースに存在する場合は、ループを解決するコンテキストにわかりやすい名前を付けてください。そのコンテキストによってどのような情報が取得できるのかを、Web Intelligence ユーザに明解に示す必要があります。

5.5.5.5 ループの自動検出

[ループの検出] を使ってユニバースでループを検出することができます。この機能はスキーマ内のループを自動検出し、ループの解決に必要なエイリアスやコンテキストを提案します。

この機能を使用すると、スキーマにループがあるかどうかを素早くチェックできます。検出したループを解決するためのエイリアスやコンテキストも自動的に提案されますが、エイリアスの自動検出やコンテキストの自動検出と比べると、エイリアスやコンテキストの作成順を完全に制御できないという難点があります。

推奨するループ解決の手順については、230 ページの「[一般的なループ識別方法と解決方法](#)」を参照してください。

注

整合性のチェックでは、結合、カードィナリティ、ループを含むユニバース構造のエラーを自動的にチェックできます。整合性のチェックでは、発見されたエラーの解決策も提案されます。詳しい説明は、263 ページの「[ユニバースの整合性の手動チェック](#)」を参照してください。

スキーマ内のループを検出する

- 1 スキーマにあるすべての結合にカーディナリティが設定されていることを確認してください。
- 2 [ツール]>[自動検出]>[ループの検出]の順に選択します。

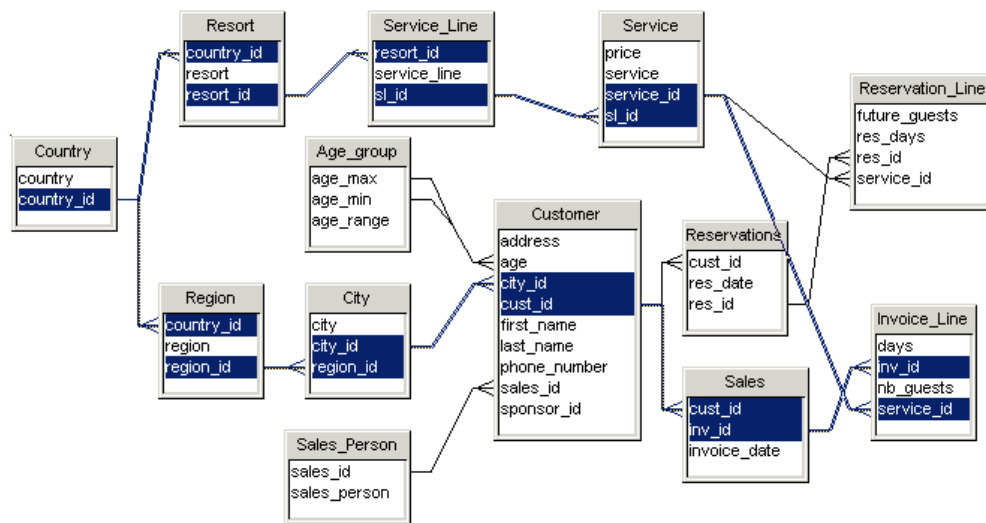
または

[ループの検出]ボタンをクリックします。

[ループの検出] ボックスが表示されます。ここに、検出されたループ数と、解消方法が表示されます。



ループを形成する結合パスが、次のように構造枠内で反転表示されます。



- 3 [次へ]ボタンをクリックして次のループとその解消策を表示します。ユニバースデザインツールで検出された各ループの結合パスが、構造枠で強調表示されます。
- 4 [閉じる]をクリックします。

5.5.5.6 エイリアスとコンテキストの自動作成

ループの検出を実行すると、ユニバースデザインツールによって、ループの解決に必要なエイリアスまたはコンテキストの候補が提案されます。提案されるエイリアスやコンテキストの候補は、その場で挿入、作成することができます。

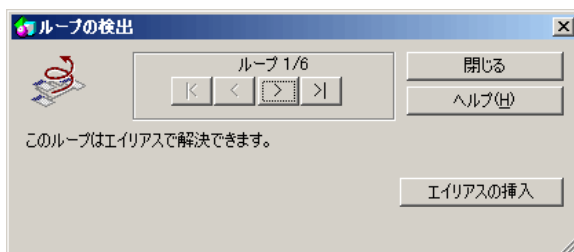
ループの検出機能を使ったエイリアスの作成

- 1 [ツール]>[自動検出]>[ループの検出]の順に選択します。

[ループの検出]ダイアログボックスが開きます。このダイアログボックスにはスキーマに含まれるループと、各ループに対してそれを解決するエイリアスやコンテキストの候補が提案されます。

- 2 次のメッセージが表示されるまで、右向きの矢印ボタンをクリックします。

このループはエイリアスで解決できます



- 3 [エイリアスの挿入]ボタンをクリックします。

エイリアスが構造枠に挿入されます。挿入されたエイリアスは、ループの原因となっていたテーブルに自動的に結合されます。

5.5.5.7 ループの検出機能を使ったコンテキストの作成

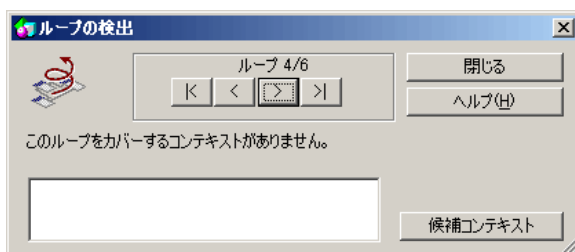
ループの検出機能を使ってコンテキストを作成する手順は、次の通りです。

- 1 [ツール]>[自動検出]>[ループの検出]の順に選択します。

[ループの検出]ダイアログボックスが開きます。このダイアログボックスにはスキーマに含まれるループと、各ループに対してそれを解決するエイリアスやコンテキストの候補が提案されます。

- 2 次のメッセージが表示されるまで、右向きの矢印ボタンをクリックします。

このループをカバーするコンテキストがありません

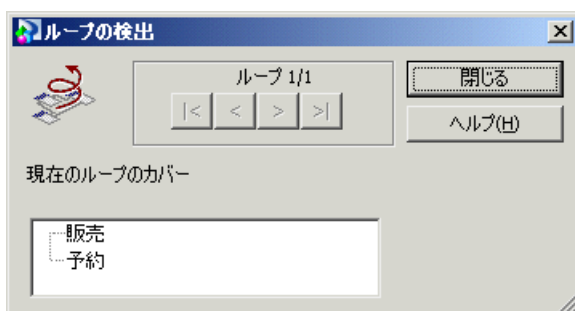


- 3 [候補コンテキスト]ボタンをクリックします。
[コンテキストの候補]ダイアログボックスが開きます。



- 4 左枠から作成するコンテキストをクリックします。
選択したコンテキストに含まれるすべてのテーブルが、スキーマ内で反転表示されます。
- 5 [追加]ボタンをクリックします。
選択したコンテキストが右枠の[コンテキストの一覧]に移動します。右枠からコンテキストを削除するには、コンテキストを選択して[削除]をクリックします。
- 6 作成するコンテキストを追加する場合は、ステップ3と4を繰り返します。
- 7 [OK]をクリックします。

コンテキストを確認するダイアログボックスが表示されます。



- 8 [閉じる]をクリックします。
作成したコンテキストが、ユニバースウィンドウのコンテキストボックスに表示されます。

5.5.6 ループ解決の例

ここでは、ループ解決の実例をいくつか紹介します。

- ・ エイリアスを作成し、共有フレキシブルルックアップテーブルが原因となるループを解決する
- ・ エイリアスを作成し、共有フレキシブルルックアップテーブルが原因となるループを解決する
- ・ ループの解決にエイリアスが適切でない場合を見分ける
- ・ コンテキストを作成し、ループを解決する
- ・ エイリアスとコンテキストの両方を作成し、ループを解決する

ここで使用するスキーマは、Beach ユニバースに基づいていません。ここでは輸送会社のスキーマを想定し、この章で紹介したBeachユニバースでのループの解決例とは違った角度から、ループの解決方法を見ていくことにします。

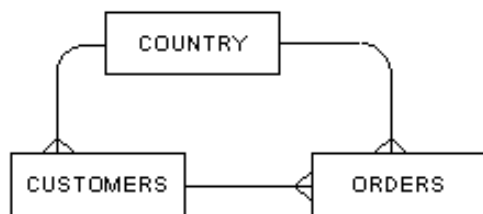
5.5.6.1 エイリアスを作成し、共有フレキシブルルックアップテーブルが原因となるループを解決する

営業データベースに、世界中の顧客に対する製品の販売情報が収められているとします。取引には、次の特徴があります。取引には、次の特徴があります。

- ・ 顧客は世界中に分散している。
- ・ 顧客は在住国からオーダーを出す。
- ・ 納品先の国は自由に指定できる。

たとえば英国に住む顧客が、ブラジルを納品先として車を1台オーダーするといった取引が記録されています。

このデータベースのスキーマは、次のようになります。



このスキーマは、次のように解釈されます。

- ・ それぞれの顧客は、特定の国に在住する。

- ・ それぞれの顧客は、複数の製品に対し複数のオーダーを出すことができる。
- ・ 納品先の国は顧客の在住国とは限らない。

この状況では、各テーブルとその列の内容は次のようになります。

country_id	country
1	USA
2	UK
3	France
4	Germany
5	Spain

cust_id	last_name	loc_country
100	COLTRANE	1
101	MULLIGAN	1
102	WALDRON	3
103	HANCOCK	4
104	DAVIS	2
105	BARBIERI	5
106	STREATS	5

order_id	cust_id	order_date	ship_country
12345	100	1/1/95	2
12346	101	1/6/95	1
12347	101	2/6/95	3
12348	102	8/4/95	5
12349	103	10/3/95	4
12350	104	15/8/95	2
12351	105	6/2/95	5
12352	106	7/3/95	4

次の情報を取得するクエリを実行するとします。

- ・ 顧客名
- ・ 顧客の在住国
- ・ 各オーダーの日付
- ・ オーダー品の発送先国

上記のデータを抽出するSQLは次のようになります。

```
SELECT CUSTOMERS.LAST_NAME, COUNTRY.COUNTRY, ORDERS.ORDER_ID, ORDERS.ORDER_DATE, COUNTRY.COUNTRY FROM
CUSTOMERS, ORDERS, COUNTRY WHERE
(CUSTOMERS.CUST_ID=ORDERS.CUST_ID) AND
(ORDERS.SHIP_COUNTRY=COUNTRY.COUNTRY_ID) AND
(CUSTOMER.LOC_COUNTRY=COUNTRY.COUNTRY_ID)
```

このSQLを実行すると、クエリ結果には自国を納品先として指定した顧客しか含まれません。つまり、自分の在住国以外の国を納品先に指定した顧客の情報は取得できません。

これは、顧客の在住国と納品国とのインターセクト部分だけが結果として取得されたためです。予測した結果は、次の表に示す通りです。

last_name	country	order_id	order_date	country
COLTRANE	USA	12345	1/1/95	UK
MULLIGAN	USA	12346	1/6/95	USA
MULLIGAN	USA	12347	2/6/95	France
WALDRON	France	12348	8/4/95	Spain
HANCOCK	Germany	12349	10/3/95	Germany
DAVIS	UK	12350	15/8/95	UK
BARBIERI	Spain	12351	6/2/95	Spain
STREATS	Spain	12352	7/3/95	Germany

生成された SQL は次の結果を返します。

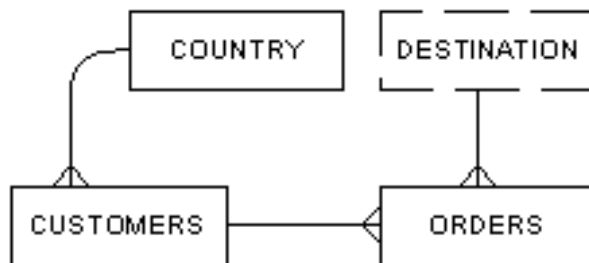
last_name	country	order_id	order_date	country
MULLIGAN	USA	12346	1/6/95	USA
HANCOCK	Germany	12349	10/3/95	Germany
DAVIS	UK	12350	15/8/95	UK
BARBIERI	Spain	12351	6/2/95	Spain

エイリアスを挿入することで、このループを分断できます。エイリアス作成の第1段階は、データベース構造で複数の用途に使われているルックアップテーブルを特定することです。次のセクションで、その方法を説明します。

5.5.6.2 多目的のルックアップテーブルを見つける

テーブル[COUNTRY]は、顧客の所在地と納品先の両方で参照されています。このようなテーブルを共有ルックアップテーブルと呼びます。

ループを解消するには、次の図のようにエイリアス[DESTINATION]を作成します。



作成されたエイリアス[DESTINATION]によって既存の3つの結合を残したままループが解消され、循環する結合パスがなくなっています。

5.5.6.3 共有ルックアップテーブルとエイリアスをFrom句で参照する

このスキーマの変更を反映させるには、From句で同じテーブルを2回、つまり本来のテーブル名で1回、エイリアス(本来のテーブル名にエイリアス名を続ける)でもう1回参照します。

SQLは、次のように変わります。

```
SELECT CUSTOMER.NAME, COUNTRY.NAME, ORDERS.ORDER_DATE DESTINATION.NAME FROM CUSTOMER, ORDERS, COUNTRY,
COUNTRY DESTINATION WHERE (CUSTOMER.CUST_ID=ORDERS.CUST_ID) AND (ORDERS.SHIP_DEST_ID= DESTINATION.COUNTRY_ID)
AND (CUSTOMER.CUST_LOC_ID=COUNTRY.COUNTRY_ID)
```

5.5.6.4 エイリアスを作成し、共有フレキシブルルックアップテーブルが原因となるループを解決する

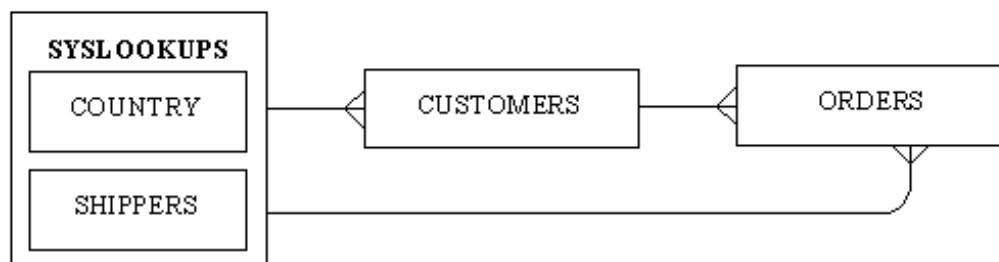
営業データベースに異なる国々に居住する顧客のデータが収められており、顧客がオーダーした品が、異なる輸送会社からさまざまな輸送能

データベース上では、国のデータと輸送会社のデータは正規化され、それぞれルックアップテーブルに収められています。正規化は、データの重複を回避することで、テーブル間の関連性を調整する処理です。

構造上の理由から、2つのルックアップテーブルを作成する代わりに、コード(code)、タイプ(type)、説明(description)の3フィールドで構成される1つのルックアップテーブル[SYSLOOKUPS]だけが作成されているとします。タイプフィールドには、国や輸送会社などのデータの種類を特定する情報が指定されています。

このようなテーブルはフレキシブルルックアップテーブルと呼ばれ、CASE ツールなどで自動生成されるスキーマでよく見られるテーブルです。

このシステムのスキーマとテーブル内容は、次のようになります。



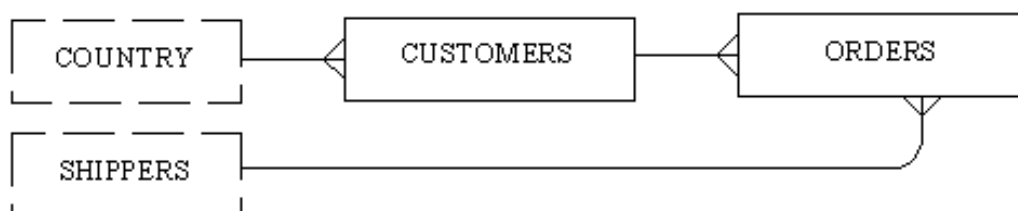
cust_id	last_name	loc_country
100	COLTRANE	1
101	MULLIGAN	1
102	WALDRON	3
103	HANCOCK	4
104	DAVIS	2
105	BARBIERI	5
106	STREETS	5

order_id	cust_id	order_date	ship_id
12345	100	1/1/95	2
12346	101	1/6/95	1
12347	101	2/6/95	3
12348	102	8/4/95	5
12349	103	10/3/95	4
12350	104	15/8/95	2
12351	105	6/2/95	5
12352	106	7/3/95	4

type	code	description
CTRY	1	USA
CTRY	2	UK
CTRY	3	France
CTRY	4	Germany
CTRY	5	Spain
SHIP	1	Man With A Van
SHIP	2	'Cut You Up' Couriers
SHIP	3	Parcel Fun
SHIP	4	Boggit & Leggit Couriers
SHIP	5	Deliveries 'R Us
SHIP	6	Sky Nut

[SYSLOOKUPS]テーブルは複数の用途に使用されているため、テーブルのドメインの数([type]フィールドの値の種類)だけエイリアスを作成する必要があります。この例では、[SYSLOOKUPS]テーブルの用途にしたがい、[COUNTRY]と[SHIPPERS]のエイリアスを作成します。

次の図は変更後のスキーマです。



ユニバースデザインツールでは、顧客の国を COUNTRY.DESCRPTION、輸送会社を SHIPPERS.DESCRPTION として定義したオブジェクトを作成できます。

関連する結合は、次のようになります。

CUSTOMERS.LOC_COUNTRY=COUNTRY.CODE

ORDERS.SHIP_ID=SHIPPERS.CODE

自己結合を使って結果を制限する

オブジェクトの定義が終了したら、それぞれのエイリアスが異なるドメインのデータを返さないように制限を加える必要があります。自己結合の作成の詳細については、184 ページの「[自己制限結合](#)」を参照してください。

たとえば、顧客101へ2件のオーダーを処理した輸送会社名を取得するとします。

SQL は次のようになります。

```
SELECT ORDERS.ORDER_ID, ORDERS.CUST_ID, ORDERS.ORDER_DATE, SHIPPERS.DESCRPTION SHIPPER FROM ORDERS, SYSLOOKUPS SHIPPERS WHERE (ORDERS.SHIP_ID=SHIPPERS.CODE)
```

このSQLで次の結果が得られます。

order_id	cust_id	order_date	shipper
12346	101	1/6/95	Man With A Van
12346	101	1/6/95	USA
12347	101	2/6/95	Parcel Fun
12347	101	2/6/95	France

このクエリでは、国名と輸送会社名の両方が返されます。これは、[Man With a Van]と[USA]が[code]の値[1]を、[Parcel Fun]と[France]が[code]の値[3]を共有しているためです。

このエラーは、次の方法で修正します。

- ・ エイリアステーブル[SHIPPERS]に新しい自己結合を設定する。[結合の編集]ダイアログボックスで、[テーブル1]と[テーブル2]の両方に[SHIPPERS]を指定し、SHIPPERS.TYPE='SHIP'のSQL式を設定します。
- ・ エイリアステーブル[COUNTRY]に新しい自己結合を設定する。[結合の編集]ダイアログボックスで、[テーブル1]と[テーブル2]の両方に[COUNTRY]を指定し、COUNTRY.TYPE='CTRY'のSQL式を設定します。

制限の適用に関する問題

オブジェクトのWHERE句に制限を追加したり、エイリアスと[CUSTOMERS]あるいは[ORDERS]テーブル間の結合に制限を追加した場合は、次の問題が発生します。

- ・ オブジェクトのWHERE句に制限を加えた場合は、エイリアスから作成するすべてのオブジェクトに同じ制限を設定する必要があります。エイリアスから多数のオブジェクトを作成する場合、ユニバースのメンテナンス上重大な問題が生じる恐れがあります。
- ・ エイリアスとほかのテーブル間の結合に制限を加えた場合は、結合が実際に使用される場合にだけ制限が有効になりますので、[SHIPPERS]オブジェクトだけを含んだクエリでは[ORDERS]テーブルは参照されず、[SHIPPERS]エイリアスはすべての行（不要な[COUNTRY]行を含め）を返します。つまり、結合は使用されず、それに設定した制限は適用されません。

要約

この例では、共有フレキシブルルックアップテーブルのスキーマについて説明し、次の作業を行いました。

- 1 共有ルックアップテーブル用にCOUNTRYとSHIPPERSエイリアスを作成する。
- 2 エイリアスに自己結合をつくり、行を制限する。

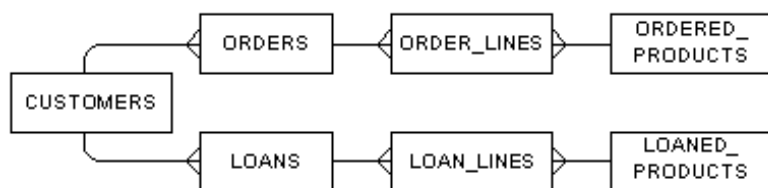
1つの複合ルックアップテーブルから2つのエイリアスを作成し、さらにエイリアスに制限(自己結合)を設定しました。つまり、テーブル構造によっては作成したエイリアスを編集したり、制限を追加する必要があるということになります。

5.5.6.5 ループの解決にエイリアスが適切でない場合を見分ける

上記のループの解消にエイリアスを作成しても、適切な解決策とはなりません。下記の理由から、このケースではコンテキストを使用するのが最良だといえます。次の例は、エイリアスが不適切で、コンテキストが最適なケースを紹介したものです。

このループを解決するため、複数の用途に使用されるルックアップテーブルを識別しようとしても、それが[PRODUCTS]テーブルであるのか、[CUSTOMERS]テーブルであるのか判断がつかかねます。

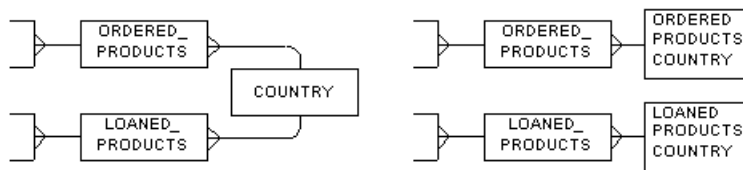
とりあえず、次のように[PRODUCTS]テーブルにエイリアスを作成した場合を考えてみます。



[PRODUCTS]テーブルのエイリアスを作成し、[ORDERED_PRODUCTS]と[LOANED_PRODUCTS]テーブルとして定義すると、ユーザに混乱を招く恐れがあります。一般的に、ユーザは製品が購入されたかレンタルされたかにかかわらず、あくまで製品全体を一つの情報として考えるのが普通だと考えられます。

さらに、製品の製造国データを[COUNTRY]テーブルとして追加した場合、このルックアップテーブルを[PRODUCTS]テーブルと直接結合する必要があります。

このテーブルを追加した場合、スキーマは次のようになります。



左のスキーマでは、再びループが発生するため[ORDERED_PRODUCTS_COUNTRY]と[LOANED_PRODUCTS_COUNTRY]のエイリアスを作成しなければなりません。このように、エイリアスの作成は本質的な解決法とならないだけでなく、ユニバースを複雑にし、その管理を困難にします。

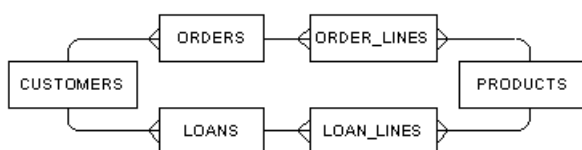
したがって、コンテキストを作成しループを解消する必要があります。

5.5.6.6 コンテキストを作成し、ループを解決する

データベースに、製品を購入またはレンタルした顧客の情報が収められているとします。顧客と製品のリレーションシップは、次の2つの異なるモデルで表現できます。

- ・ 顧客がオーダー(購入)した製品に基づいたモデル
- ・ 顧客がレンタルした製品に基づいたモデル

このデータベースは次のスキーマで表現することができます。



[ORDER]と[ORDER_LINES]テーブルを使用して、単純な顧客名と製品名の一覧を取得するクエリを実行できます。製品をオーダーした顧客別の製品一覧が取得されます。

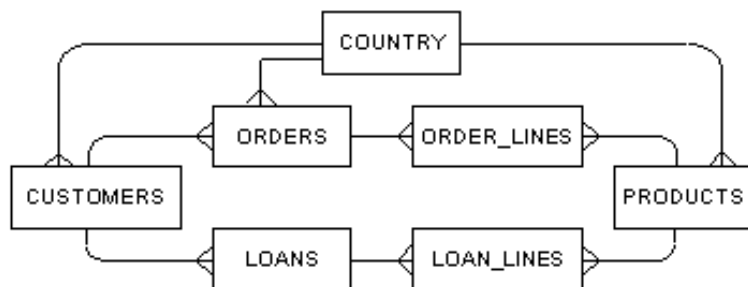
逆に、[LOANS]と[LOAN_LINES]テーブルを使用すると、製品をレンタルした顧客別の製品一覧だけを取得できます。

このスキーマにはループがあるため、6つの結合をすべて含むクエリの結果には、販売とレンタルの両方の実績がある製品しか含まれません。つまり、それぞれの顧客について、購入はしたがレンタルはしていない製品と、レンタルはしたが購入はしていない製品は、結果から除外されてしまいます。

エイリアスとコンテキストの両方を作成し、ループを解決する

コンテキストとエイリアスの両方を使用して、ユニバースのループを解決することができます。次の例は、ループの解決にコンテキストとエイリアスの両方を使用するものです。

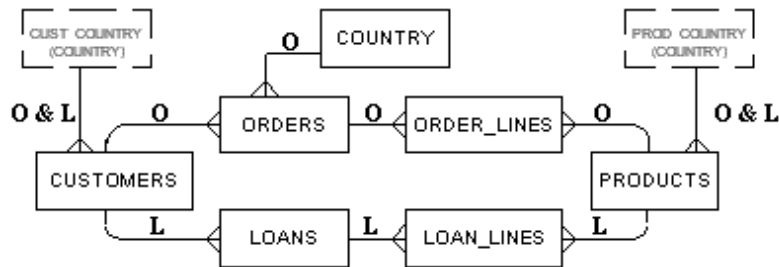
次のようなスキーマを持つユニバースを考えてみます。



このユニバースのループを解消するには、次のエイリアスとコンテキストを使う必要があります。

- ・ [COUNTRY]テーブルに2つのエイリアス([CUST_COUNTRY]と[PROD_COUNTRY])を作成する。
- ・ [CUSTOMERS]と[PRODUCTS]間のループを解決するために、コンテキスト([ORDERS]と[LOANS])を作成する。
- ・ [CUSTOMERS]と[CUST_COUNTRY]、[PRODUCTS]と[PROD_COUNTRY]の2つの結合が両方のコンテキストに含まれていることを確認する。

操作の結果、スキーマが次のように変更されます。



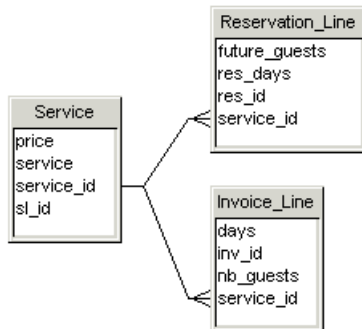
5.6 キャズムトラップの解決

キャズムトラップとは、実際よりも多い行数を返す結合パスがあるリレーショナルデータベースのスキーマで発生する一般的な問題です。

5.6.1 キャズムトラップとは

キャズムトラップとは、3つのテーブル間の結合パスの一種で、2つの「多対一」結合が同じテーブルに集中し、それを分割するコンテキストが定義されていない状況を指します。

次の例は、Beach ユニバーススキーマの一部です。3つのテーブルがスキーマ全体から分割され、キャズムトラップが表示されています。データとしては同じ Club 接続を使用していますが、[Service] テーブルが2つの「多対一」結合の「一」側でリンクされています。



このスキーマでは、次のすべての条件が満たされた場合には、正しい情報を取得できません。

- ・ユニバース構造の3つのテーブル間に「多対一対多」リレーションシップがあること
- ・それぞれの結合の「多」側にある2つのテーブルを元にしたオブジェクトがクエリに含まれること
- ・1つのディメンションに対し、複数の行が返されること

上記のすべての条件が満たされた場合に、クエリがデカルト積を返す例を次に紹介します。

例 キャズムトラップは警告を出さずに結果行数を増殖する

上記のスキーマを使って Web Intelligence ユーザーがクエリーを実行すると、次の結果が返されます。

クエリー	結果				
<div> <div>サービス ● 客数</div> <div>サービス 等しい (=) 'スポーツ'</div> </div>	<table> <tr> <th>サービス</th><th>客数</th></tr> <tr> <td>Sports</td><td>145.00</td></tr> </table>	サービス	客数	Sports	145.00
サービス	客数				
Sports	145.00				
<div> <div>サービス ● 予定客数</div> <div>サービス 等しい (=) 'スポーツ'</div> </div>	<table> <tr> <th>サービス</th><th>予定客数</th></tr> <tr> <td>Sports</td><td>8.00</td></tr> </table>	サービス	予定客数	Sports	8.00
サービス	予定客数				
Sports	8.00				

次に、すでに滞在を終了した客と予約客を含めたクエリを実行したとします。

<div> <div>サービス ● 客数 ● 予定客数</div> <div>サービス 等しい (=) 'スポーツ'</div> </div>

次の結果が返されます。

サービス	客数	予定客数
Sports	188.00	96.00

Sports サービスを利用した顧客数と予約した顧客数が大幅に増えています。デカルト積が生成されたため、不正確な結果が返されています。このような問題を検出できないと、重大な影響が出る恐れがあります。この例では、アイランドリゾートの管理者が、実際よりも大幅にスポーツサービスが利用されていると勘違いすることになってしまいます。

5.6.2 キャズムトラップの仕組み

キャズムトラップがあると、クエリはそれぞれのメジャーで可能なすべての行を組み合わせた結果を返します。上記の例では、次の情報が組み合わせられています。

- ・ [客数]のトランザクション*[予定客数]のトランザクション
- ・ [予定客数]のトランザクション*[客数]のトランザクション

次の例では、キャズムトラップがデカルト積を返す仕組みを詳しく見てみましょう。

例 キャズムトラップのデカルト積を調べる

データを集計するクエリが返す行を調べる必要があります。ここでは、ディメンション”請求日”と”予約日”をクエリに追加し、それぞれのトランザクションの詳細を見ることにします。

滞在客数のレポートは次のように表示されます。

サービス	請求日	客数
Sports	3.00	4.00
Sports	4.00	133.00
Sports	6.00	8.00

予定客数のレポートは次のように表示されます。

サービス	予約日	予定客数
Sports	1.00	3.00
Sports	2.00	7.00

これらのレポートから、次のトランザクション回数がわかります。

- ・ 滞在客数 = トランザクション回数3
- ・ 予定客数 = トランザクション回数3

2つのディメンションを同じクエリに入れると、次の結果が返されます。

サービス	請求日	客数	予約日	予定客数
Sports	3.00	4.00	1.00	3.00
Sports	3.00	4.00	2.00	1.00
Sports	4.00	129.00	1.00	4.00
Sports	4.00	35.00	2.00	9.00
Sports	6.00	8.00	1.00	6.00
Sports	6.00	8.00	2.00	2.00
	合計:	188.00	合計:	96.00

このクエリは、[予定客数]のすべての行と[客数]のすべての行を組み合わせた結果を返しています。[客数]のトランザクションが3回、[予定客数]のトランザクションが3回繰り返してカウントされています。

取得したデータを合計しても、もちろん正しい結果は得られません。

ユニバースデザインツールでは、ルートとは異なり、キャズムトラップを自動的に検出することはできませんが、[ツール] > [コンテキストの検出]を選択してコンテキストを自動検出し、提案されるコンテキストの候補をスキーマで使うことができます。

コンテキストを検出して、スキーマ内にある「多対一」結合を調べることができます。また、コンテキストを検出することにより、「多対一」結合が集中するテーブルを取得し、テーブルで実行されるクエリを分割するコンテキストを提案します。これが、スキーマにキャズムトラップが含まれていないかどうかを確認する最も効率的な方法です。

スキーマに表示される「一対多」結合を目視で分析して、キャズムトラップを検出することもできます。

スキーマでキャズムトラップを見分けられない場合は、詳細データの表示行を丹念に調べる以外に問題を発見する方法はありません。これらの方法で問題を発見できない場合でも、キャズムトラップの危険を示す兆候は一切ありません。

5.6.3 キャズムトラップの検出

コンテキストの自動検出機能を使って必要なコンテキストを検出し、2つのコンテキストが合流する場所を探すことで、キャズムトラップを発見することができます。コンテキストの重なるポイントが、キャズムトラップの原因です。

同じルックアップテーブルに到達する「多対一」結合を持った、2つのファクトテーブルがある場合、キャズムトラップの恐れがあります。

ヒント

結合の問題を検出するための、テーブルスキーマの編集方法については、258 ページの「[結合問題の視覚的検出](#)」を参照してください。

5.6.4 キャズムトラップの解決

キャズムトラップを解決するには、2つの独立したクエリを実行し、結果を組み合わせる必要があります。ファクトテーブルに定義されたオブジェクトの種類とエンドユーザ環境にしたがい、次の解決方法があります。

- ・ それぞれのファクトテーブルについてコンテキストを作成する。この解決策は、すべてのユニバースに適用できます。
- ・ ユニバースの SQL パラメータを変更し、各メジャーに対して独立した SQL クエリを生成する。この解決策は、メジャーオブジェクトにのみ適用できます。この方法では、ディメンションや詳細オブジェクトには複数のクエリを生成できません。

次のセクションでは、それぞれの解決方法について説明します。

5.6.4.1 コンテキストを使ったキャズムトラップの解決

結合の「多」側にある各ファクトテーブルに、コンテキストを定義できます。この例では、[SERVICE]から[RESERVATION_LINE]と[SERVICE]から[INVOICE_LINE]のそれぞれにコンテキストを定義します。

両方のコンテキストに含まれるオブジェクトを使ってクエリを実行すると、デカルト積を避けるため2つのSELECT文が生成され、Web Intelligence 内に2つのテーブルが作成されます。

5.6.4.2 コンテキストが有効なケース

コンテキストを作成すると、キャズムトラップは必ずユニバースで解決されます。ファクトテーブルのどちらか、または両方にディメンションオブジェクトがある場合は、コンテキストを使用する必要があります。

5.6.4.3 コンテキストを使ったキャズムトラップの解決

コンテキストを使ってキャズムトラップを解決する手順は、次の通りです。

- 1 スキーマ内で「一対多対一」の結合パスを分析し、潜在的なキャズムトラップを識別します。
- 2 [ツール]>[コンテキストの検出]の順に選択します。

[コンテキストの候補]ダイアログボックスが表示されます。



- 3 左枠から作成するコンテキストをクリックし、[追加]ボタンをクリックして、右枠の[コンテキストの一覧]に追加します。
- 4 すべてのコンテキスト候補を追加します。

[OK]をクリックします。作成したコンテキストが、ユニバースウィンドウ(一覧モード)のコンテキストボックスに表示されます。

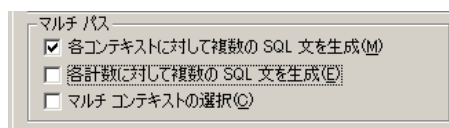
- 5 [ファイル]>[パラメータ]を選択します。

[ユニバースのパラメータ]ダイアログボックスが表示されます。

- 6 [SQL]タブをクリックします。

[SQL]ページが表示されます。

- 7 [各コンテキストに複数の SQL 文を生成]チェックボックスを選択します。



- 8 [OK]をクリックします。

キャズムトラップ内のテーブルにクエリを実行すると、クエリがメジャーとディメンションに分割されて実行されます。

5.6.4.4 [各メジャーに対して複数の SQL 文を生成]オプションの使い方

両方のファクトテーブルにメジャーオブジェクトしか定義されていない場合は、ユニバースパラメータの [各メジャーに対して複数の SQL 文を生成] オプションを使用できます。この選択により、クエリパネルに表示されるそれぞれのメジャーに、独立した SQL クエリが生成されます。

この解決策は、ディメンションと詳細オブジェクトには適用できません。

[各メジャーに対して複数の SQL 文を生成]オプションが有効な場合と無効な場合を次の表にまとめます。

選択	状況
[各メジャーに対して複数の SQL 文を生成]オプションを使う。	メジャーオブジェクトだけを含むユニバースでは、両方のファクトテーブルに定義されます。複数のSQL文を生成することで、メンテナンスが必要なコンテキストを使用せずに済みます。
[各メジャーに対して複数の SQL 文を生成]オプションを使わない。	<p>どちらかのファクトテーブルにディメンションオブジェクトまたは詳細オブジェクトが定義されている場合は、このオプションは使用すべきではありません。このオプションが設定されたユニバースに対して、ディメンションオブジェクトまたは詳細オブジェクトを含んだクエリを実行すると、デカルト積が返されます。</p> <p>このオプションを設定するとクエリの実行時間が長くなりますので、クエリのパフォーマンスが大幅に低下するようであれば、コンテキストの作成やユニバースの分割でキャズムトラップを解決する必要があります。</p>

[各メジャーに対して複数の SQL 文を生成]オプションを有効にする手順は、次のとおりです。

- 1 [ファイル]>[パラメータ]の順に選択します。
[ユニバースのパラメータ]ダイアログボックスが表示されます。
- 2 [SQL]タブをクリックします。
- 3 [マルチパス]グループボックスにある[各メジャーに対して複数の SQL 文を生成]オプションをチェックします。
- 4 [OK]をクリックします。

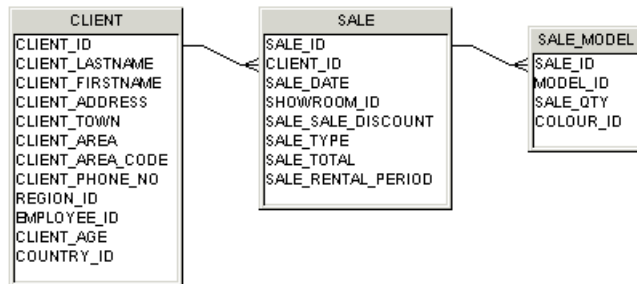
5.7 ファントラップの解決

リレーショナルデータベースのスキーマでは、ファントラップはキャズムトラップほど一般的な問題ではありません。キャズムトラップと同様、ファントラップも実際よりも多い行数を返します。

5.7.1 ファントラップとは

ファントラップは3つのテーブル間の結合パスの一種で、「一対多」結合が他の「一対多」結合を持つテーブルにリンクされている状況を指します。両方のテーブルから作成されたオブジェクトを含んだクエリを実行すると、「一対多」結合の連鎖によって不正確な結果が返されます。

次の図は、ファントラップの簡単な例です。



特定の顧客に対して、モデルごとの販売車両台数の合計を問い合わせるクエリを実行すると、結合の「多」側にある結合を解除せずに、結合の「一」側のテーブルに集合関数を適用するため、不正確な結果が返されます。

例 ファントラップは警告を出さずに結果行数を増殖する

前ページのスキーマを元に Web Intelligence ユーザが次のクエリを実行したとします。

ClientName SaleValue SaleQTY
ClientName 等しい (=) WendyCraig

次の結果が返されます。

ClientName	SaleQTY	SaleValue
WendyCraig	2.00	57,092.00

この結果に問題はありません。ユーザが次のようにディメンション[モデルID]をクエリに追加したとします。

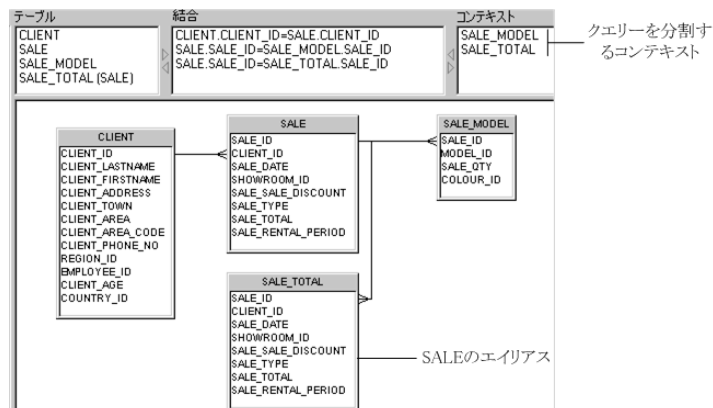
ClientName SaleValue Model Id SaleQTY
ClientName Equal to 'WendyCraig'

次の結果が表示されます。

WendyCraig		
Model Id	SaleValue	SaleQTY
1,034.00	57,092.00	1.00
1,081.00	57,092.00	1.00
Sum:	114,184.00	2.00

このレポートでは、SaleValue が 2 倍になっています。Model_ID の各インスタンスは 1 倍です。これらの結果がレポートで集約されると、合計が不正確になります。ファントラップがデカルト積を返したのが理由です。顧客の WendyCraig は、車 2 台を合計 \$57,092.00 で購入したのであり、レポートに表示される \$114,184.00 は正しい数値ではありません。クエリに Model_ID を追加したことで、SaleValue が ModelId の値の数だけ繰り返し集計されています。

クエリ内にディメンションオブジェクトのあるファントラップは、エイリアスとコンテキストを使って解決できます。次のスキーマは、このファントラップスキーマの解決方法です。



WendyCraig に対しデカルト積を返したクエリを再度実行すると、今度は次のテーブルが表示されます。

WendyCraig

Sale Qty	Model Id	Sale Total
1.00	1,034.00	57,092.00
1.00	1,081.00	

5.7.2 ファントラップの検出方法

ファントラップを自動的に検出することはできません。テーブルスキーマに表示されるカーディナリティを、目視で検討する必要があります。

メジャーオブジェクトが参照するテーブルが 2 つあり、これらのテーブルに連続した多対一結合が設定されている場合、ファントラップの恐れがあります。

結合の問題を検出するためのテーブルスキーマの編集方法については、258 ページの「[結合問題の視覚的検出](#)」を参照してください。

5.7.3 ファントラップの解決方法

ファントラップの解決方法は、2種類あります。

- ・ 最初の集計を行うテーブルにエイリアスを作成し、コンテキストを自動検出([ツール]>[コンテキストの検出])して、ベーステーブルとエイリアステーブルのコンテキストを挿入する方法これは、ファントラップの問題を解決する一番効率的な方法です。
- ・ ユニバースのSQLパラメータを変更する方法: この方法は、メジャーオブジェクトにしか利用できません。

次に、それぞれの解決方法を説明します。

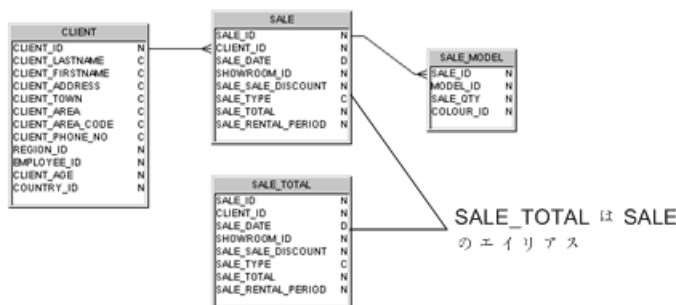
5.7.3.1 エイリアスとコンテキストを使ったファントラップの解決

集計データを提供するテーブルのエイリアスを作成し、コンテキストを検出して挿入することで、クエリを分割します。次のようにして、それを行います。

エイリアスとコンテキストを使ったファントラップの解決方法は、次の通りです。

- 1 スキーマ内で「一対多対一対多」の関係を持った結合パスを分析し、潜在的なファントラップを識別します。
- 2 集計データを提供するテーブルにエイリアスを作成します。

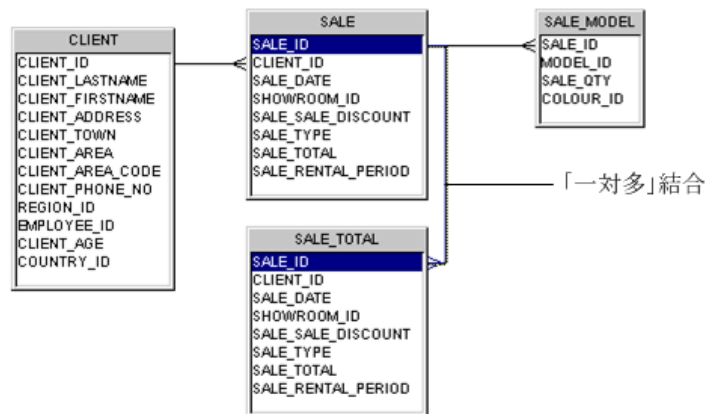
たとえば、ファントラップの例で取り上げた[販売価格]オブジェクトは、[Sale]テーブルの[Sale_Total]列を集計するオブジェクトです。したがって、[Sale]テーブルのエイリアスを[Sale_Total]という名前で作成します。



- 3 ベーステーブルとエイリアステーブルに結合を作成します。

「一対一」結合を作成すると、ユニバースデザインツールによってコンテキストを検出できないため、コンテキストを手動で作成する必要があります。通常は、「一対多」結合を作成すると、コンテキストを自動検出して挿入することができます。

次のように [Sale] と [Sale_Total] 間に「一対多」結合を作成します。



- 4 エイリアステーブルを元に集計を行うオブジェクトを作成します。

ここでは、[販売価格]オブジェクトの定義

sum(SALE.SALE_TOTAL) を新しい定義

sum(Sale_Total.SALE_TOTAL) に変更します。

- 5 [ツール]>[コンテキストの検出]の順に選択します。

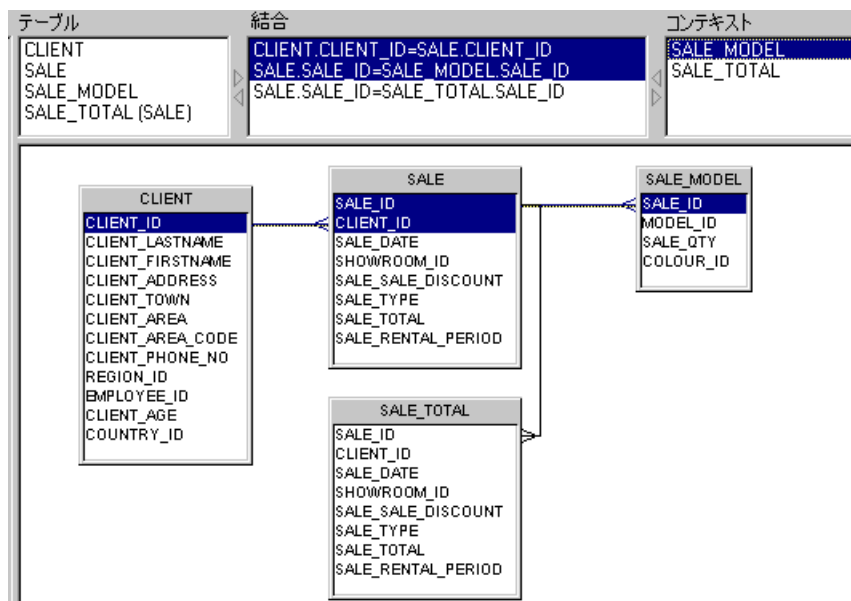
[コンテキストの候補]ダイアログボックスが表示されます。[コンテキストの候補]ダイアログボックスが開き、ベーステーブル用コンテキストとエイリアステーブル用コンテキストが提案されます。

注

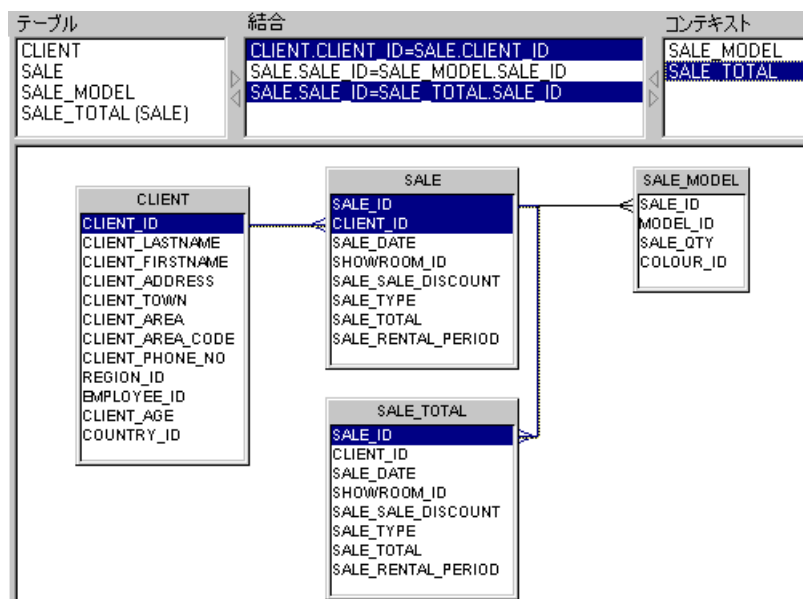
エイリアステーブルとベーステーブルを「一対一」結合でリンクした場合は、コンテキストを手動で作成する必要があります。

- 6 コンテキスト候補をクリックし、[追加]ボタンをクリックします。
- 7 もう一つのコンテキスト候補も同様に追加します。
- 8 [OK]をクリックします。

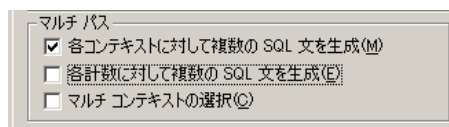
コンテキストが作成され、スキーマに表示されます。一覧モード([表示]>[一覧モード])の場合は、作成したコンテキストの名前をコンテキスト枠で確認できます。CLIENT>SALE>SALE_MODELのコンテキストパスは、次の通りです。



CLIENT>SALE>SALE_TOTAL のコンテキストパスは、次の通りです。



- 9 [ファイル]>[パラメータ]を選択します。
[パラメータ]ダイアログボックスが開きます。
- 10 [SQL]タブをクリックします。
[SQL]ページが表示されます。
- 11 [各コンテキストに対して複数のSQL文を生成]オプションをチェックします。



- 12 [OK]をクリックします。
- 13 クエリを実行し、ファントラップが解決されたことを確認します。

5.7.3.2 [各メジャーに対して複数の SQL 文を生成]オプションの使い方

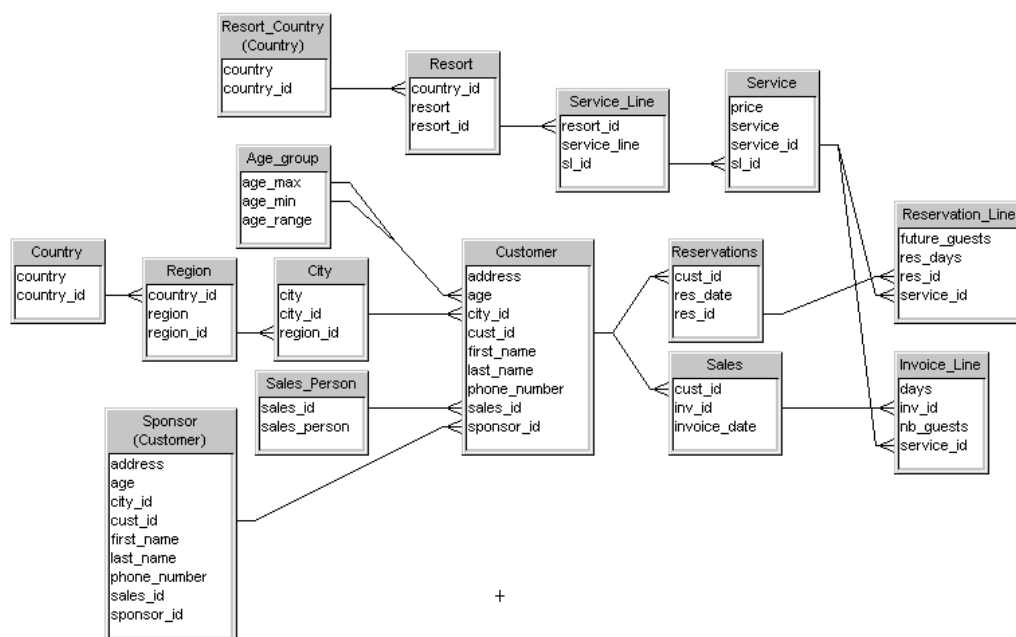
連続した「一対多」結合の「多」側の2つのテーブルにメジャーオブジェクトしか定義していない場合は、[各コンテキストに対して複数の SQL を使う] オプションを使ってファントラップを解決できます。この選択により、クエリパネルに表示されるそれぞれのメジャーに、独立した SQL クエリが生成されます。

ただし、このオプションではディメンションに対して複数の SQL 文は生成できません。エンドユーザがこれらのメジャーを参照するディメンションオブジェクトをクエリに含めることが予想される場合は、エイリアスとコンテキストを作成してファントラップを解決してください。

このオプションの詳しい説明と設定方法は、258 ページの [「\[各メジャーに対して複数の SQL 文を生成\]オプションの使い方」](#)を参照してください。

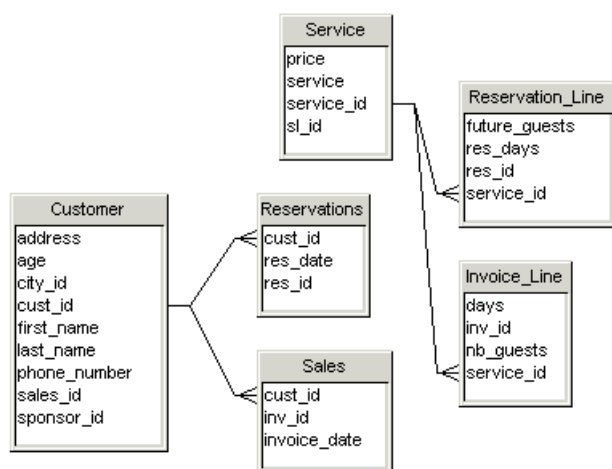
5.8 結合問題の視覚的検出

[構造] 枠に表示されるテーブルを再構成し、潜在的なキャズムトラップとファントラップがないかを視覚的に確認できます。これには、「多」側の結合が枠の片側に、「一」側の結合が反対側にくるようにテーブルの配置を変更します。次の Beach ユニバースのスキーマは、「一対多」の関係が左から右に並ぶように、テーブルスキーマを再構成したものです。



5.8.1 潜在的なキャズムトラップ

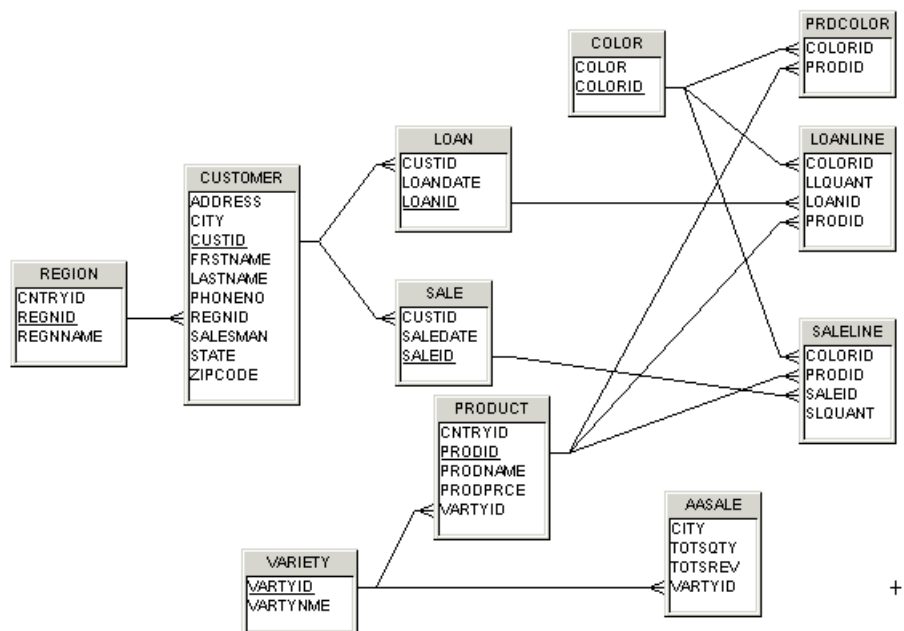
潜在的なキャズムトラップの例です。



最終的なユニバースでは、どちらの結合パスもSalesとReservationsの2つのコンテキストに分割されています。

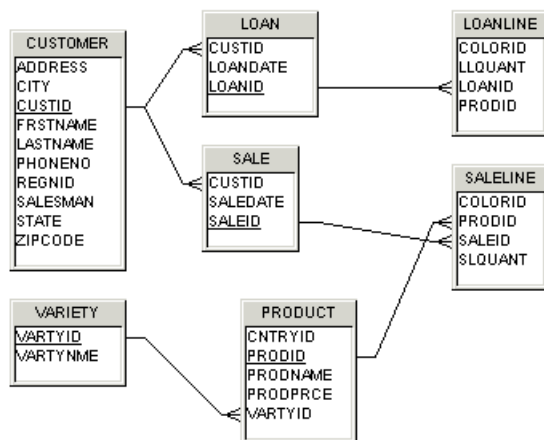
5.8.2 潜在的なファントラップ

次の例は、車両販売データベースのユニバーススキーマです。



次の結合パスに、潜在的なファントラップが考えられます。

- ・ CUSTOMER → LOAN → LOANLINE
- ・ CUSTOMER → SALES → SALELINE
- ・ VARIETY → PRODUCT → SALELINE



ヒント

スキーマに必要なテーブルをすべて挿入し終わっても、すぐにオブジェクトの作成には取り掛からないでください。テーブルを移動して、すべての「一対多」結合を同じ方向に設定します。ユニバースデザインツールは、グラフィックツールであるため、製品の視覚機能を活用してユニバースを設計できます。1時間程度、テーブルの再配置と視覚分析を行うことで、実際のデザインにかかる時間を大幅に節約できます。

5.9 ユニバースのチェック

ユニバースのデザインが進むにつれ、その整合性を定期的にチェックすることが必要になります。ユニバースの整合性は、次の方法でチェックします。

チェック方法	説明
自動	ユニバースデザインツール オプションでは、定義の作成時、ユニバースのエクスポート時、ユニバースファイルを開いたときにユニバース構造の SQL 構文を自動的にチェックするよう設定できます。
手動	[整合性のチェック]コマンドで、選択したユニバース構造の整合性をチェックできます。

5.9.1 ユニバースの整合性の自動チェック

ユニバースデザインツールでは、次の整合性チェックオプションを使用して、定義の作成時、ユニバースのエクスポート時、ユニバースファイルを開いたときに SQL 構造を分析することができます。

自動チェックオプション	説明
定義の自動分析	ユニバースデザインツールでは、すべてのオブジェクト、条件、結合の作成時に、SQL 定義が自動的にチェックされます。[OK]をクリックすると、すぐに設定が有効になります。
エクスポート前に整合性チェックの警告を表示	ユニバースデザインツールでは、整合性をチェックしていないユニバースをエクスポートしようとすると警告メッセージが表示されます。
開くときにユニバース構造をチェック	ユニバースを開く際に自動的に整合性をチェックします。

5.9.1.1 ユニバースの自動チェックオプションの設定

ユニバースの自動チェックオプションは、次の手順で設定します。

- 1 [ツール]>[オプション]の順に選択します。
[オプション] ダイアログボックスの [一般設定] ページが表示されます。
- 2 必要に応じ、[整合性]グループボックスの自動チェックオプションを選択します。

整合性

☒ 定義の自動分析(P)

☐ エクスポート前に整合性チェックの警告を表示(W)

☐ 開くときにユニバース構造をチェック(S)

- 3 [OK]をクリックします。

5.9.2 ユニバースの整合性の手動チェック

[整合性のチェック] オプションを使用して、アクティブユニバースのデザインが正確で最新のものであるかどうかを確認することができます。

「整合性のチェック」の目的は、以下の通りです。

- ・ ユニバースのオブジェクト、結合、カーディナリティのエラーを検出する。
- ・ 結合パスのループを検出する。
- ・ 必要なコンテキストを検出する。
- ・ ユニバースが接続するデータベースが変更されていないか検査する。

ユニバースの要素をデータベースと照らし合わせる前に、データベースへの接続が有効であるかどうかをこのオプションを使って確認します。接続が無効な場合は、チェックは継続されず、エラーメッセージが表示されます。

5.9.2.1 整合性のチェックで検出できるエラーの種類

整合性チェックでは、以下のエラーを検出します。

- ・ オブジェクト、条件、結合のSQL定義内の無効な構文
- ・ ループ
- ・ 孤立テーブル
- ・ 孤立結合
- ・ コンテキスト内のループ
- ・ 未設定あるいは間違って設定されているカーディナリティ

5.9.2.2 整合性チェックによるデータベース更新の検出

整合性チェックでは、まずデータベースに対してテーブルリストをリクエストします。次に、取得したリストをユニバースのテーブルと比較します。列に対しても同じ作業を行います。

リスト内のテーブルや列と一致しないテーブルと列は、構造枠内で使用できないテーブルとしてマークされます。こういったテーブルや列は、データベースで削除されたり、名前を変更されたテーブルや列であることがよくあります。詳しくは、266 ページの「[ユニバース構造の最新表示](#)」を参照してください。

注

データ量が多いと、カーディナリティのチェックに時間がかかる場合があります。データが曖昧、または不足していると、不正確な結果を得ることになります。データベースが非常に大きく、データ入力が不完全な場合、[カーディナリティのチェック] オプションを使用しないでください。このオプションを使う場合は、PRMファイルをカスタマイズして自動検出を最適化できます。詳しい説明は、195 ページの「[カーディナリティの自動検出の最適化](#)」を参照してください。

5.9.2.3 整合性のチェック機能を使ったユニバースのチェック

ユニバースの整合性をチェックする手順は、次の通りです。

- 1 [ツール]>[整合性のチェック]の順に選択します。

または

[整合性のチェック]ボタンをクリックします。

[整合性のチェック]ダイアログボックスが開きます。

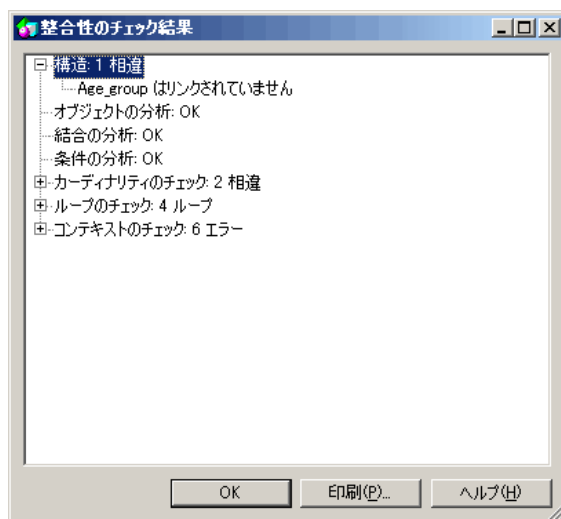
- 2 チェックするコンポーネントを選択します。
- 3 チェックしないコンポーネントを選択解除します。
- 4 [分析のレベル]グループボックスでは、[クイック分析]を選択すると、コンポーネントの構文のみをチェックします。

または

[すべて分析]を選択すると、コンポーネントの構文とセマンティックをチェックします。

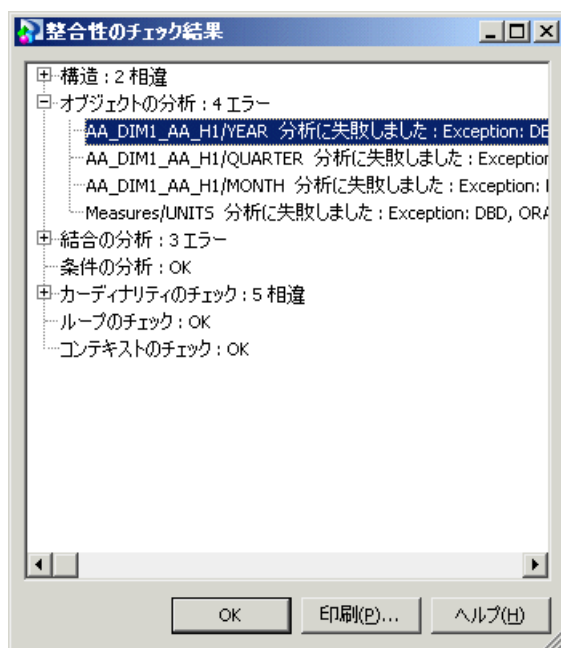
- 5 [OK]をクリックします。

[整合性のチェック結果]ダイアログボックスが開き、チェックの進行状況に合わせてチェック結果が表示されていきます。



エラーが検出されなかった場合は、各チェック項目の横に[OK]が表示されます。

- 6 左端の(+)記号をクリックして一覧を展開し、エラーの詳細を表示します。



一覧の項目をダブルクリックして、構造枠にある対応するコンポーネントを反転表示することができます。

- 7 [印刷]をクリックすると、このダイアログボックスの内容を印刷できます。
8 [OK]をクリックします。

注

[ループのチェック]オプションを使用する場合は、あらかじめすべてのカーディナリティが設定されていることを確認してください。そうでないと、結合のループが正しくチェックされません。

5.9.3 ユニバース構造の最新表示

整合性のチェックで、ユニバースの基となるデータベースが更新されていることがわかった場合は、[表示]メニューの[構造の最新表示]コマンドを使って、構造枠の内容を更新できます。

このコマンドで検出できるデータベースの更新と、それに対するユニバース構造の変更は次の通りです。

操作	ユニバースデザインツールで以下の項目が実行されます。
テーブルに列が追加された	ユニバースの対応するテーブルへ列を追加します。
テーブルから列が削除された	削除すべき列とそれに関連した結合を表示する警告メッセージを表示します。
データベースからテーブルが削除された	削除すべきテーブルとそれに関連した結合を示す警告メッセージを表示します。
データベース内のテーブル名が変更された	「選択されたテーブルが見つかりません」というメッセージを表示します。データベースのテーブル名に合わせ、ユニバースのテーブル名を変更する必要があります。名前の変更後も一致するテーブル名がデータベースに見つからない場合は、ユニバースデザインツールから「選択されたテーブルがデータベースに存在しません」というメッセージが返されます。
データベースは変更されていない	「更新の必要がありません」というメッセージを表示します。

5.9.3.1 ユニバースの最新表示

ユニバース構造を最新表示する手順は、次のとおりです。

- ・ [表示]>[構造の最新表示]を選択します。

メッセージボックスが開き、データベースの更新内容か、更新の必要がないことを示すメッセージが表示されます。

ユニバースの作成

スキーマを作成し、整合性のチェックを実行して、ループの問題を解決したら、レポートツールで使用するユニバースを作成できます。

6.1 概要

この章では、Web Intelligence ユーザーがクエリーの実行とレポートの作成で使用するクラスとオブジェクトの作成方法を説明します。エンドユーザーのレポート作成機能を拡張するためのオブジェクト定義の最適化とユニバースの最適化の方法も紹介します。

これまでの章では、ユニバースの設計方法およびテーブル、列、結合などのユニバースのデータベース構造を使ってテーブル スキーマを作成する方法を説明しました。結合パスのループを解決する方法についても解説しました。

作成したスキーマは、Web Intelligence ユーザーには表示されません。データベース構造の作成が終了すると、[ユニバース]枠に表示されるクラスとオブジェクトを開発し、ユーザーがクエリーを実行してドキュメントとレポートを生成することになります。

6.2 ユニバースの作成

ユニバースの作成は、ユニバース開発サイクルではオブジェクトの作成段階にあたります。オブジェクトは、ユーザーの分析結果に基づき、結合パスの問題を解決済みの健全なスキーマデザインを使って作成しなければなりません。

次のリストは、一般的なユニバース開発プロセスにおけるユニバース作成(およびテスト)作業の工程を明確にしたものです(実装、手順 2)。

- ・ 事前準備
 - 1 ユーザーのニーズ分析
 - 2 設計
- ・ 実装
 - 1 スキーマのデザインとテスト
 - 2 ユニバースオブジェクトの作成とテスト
 - 3 リポジトリを使用したユニバースのデプロイメント

- ・ メンテナンス

- 1 ユーザ要件またはデータソースの変更に基づくユニバースの更新と保守

6.2.1 オブジェクトとは

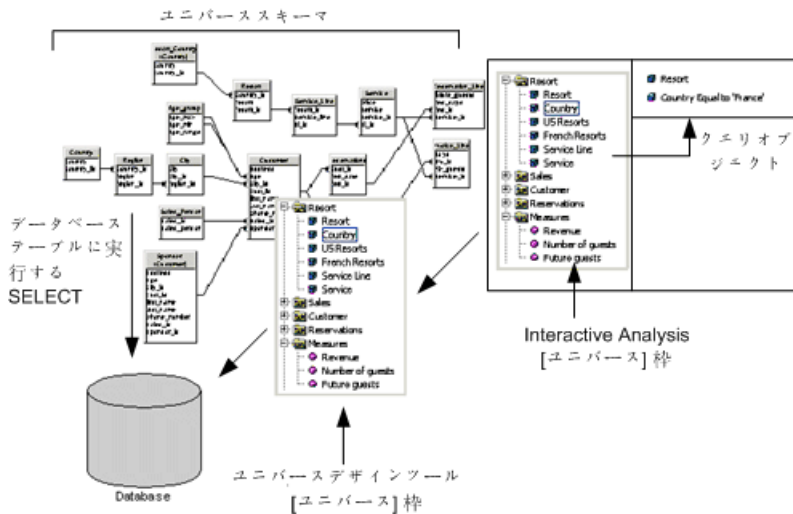
Business Objects製品では、データベースの列あるいは関数を表すユニバースのコンポーネントをオブジェクトと言います。

オブジェクトは[ユニバース]枠にアイコンで表示されます。それぞれのオブジェクトは、エンドユーザがビジネス環境で実際に扱う情報やデータを表します。ユニバースデザインツールの[ユニバース]枠で作成したオブジェクトがエンドユーザに表示され、エンドユーザはそれを使用してレポートを作成します。また、ユニバースデザインツールだけで使用するオブジェクトを作成し、[ユニバース] ペインでオブジェクトを Interactive Analysis ユーザに表示しないようにできます。

Web Intelligence ユーザは、オブジェクトを [ユニバース] ペインから [クエリ] ペインにドラッグし、クエリを実行し、取得したデータでレポートを作成します。

それぞれのオブジェクトは、データベースの列または関数にマッピングされており、[クエリ]枠で使用すると、対応する SELECT 文が生成されます。クエリパネルや Web パネルに複数のオブジェクトが挿入されると、それぞれのオブジェクトに関連付けられた SELECT 文とデフォルトの WHERE 句を含んだ SQL が実行されます。






次の図は、Web Intelligence の [ユニバース] ペインのオブジェクトと ユニバースデザインツールの [ユニバース] ペインのオブジェクトを示したものです。ユニバースデザインツールの [ユニバース] ペインにある各オブジェクトはユニバーススキーマの列にマップされ、クエリの実行時に使用される SELECT 文と関連付けられています。



ユニバースデザイナーは ユニバースデザインツールを使用して、Web Intelligence ユーザがクエリを実行するために [クエリ] ペインに含めるオブジェクトを作成します。

6.2.2 ユニバースで使用するオブジェクトの種類

オブジェクトには、次の3種類の資格があります。

オブジェクトの資格	例	説明
ディメンション	 都道府県  都市  店名	クエリによる分析の焦点となるオブジェクトです。ディメンションは、クエリのキー要素となる、データベース内の1つあるいは複数の列または関数にマップされます。
詳細	 店名  郵便番号  住所	ディメンションについての補足データを提供します。詳細オブジェクトは常にディメンションに従属します。ディメンションに関連付けられた詳細情報を提供する、データベース内の1つあるいは複数の列または関数にマップされます。
メジャー	 売上げ  販売数量  利益	データベースの統計値にマップされた集計関数が定義されています。

オブジェクトを作成するときは、それがクエリ内で果たす役割に応じて資格を設定します。この役割に従って、オブジェクトが[クエリ]枠で使われたときに生成される SELECT 文が決まります。

6.2.3 クラスとは

クラスとは、オブジェクトのコンテナです。クラスは、Windowsのフォルダと同じものです。作成したクラスには、ユニバース内で共通の目的を持ったオブジェクトを格納します。

6.2.4 クラスとオブジェクトの使い方

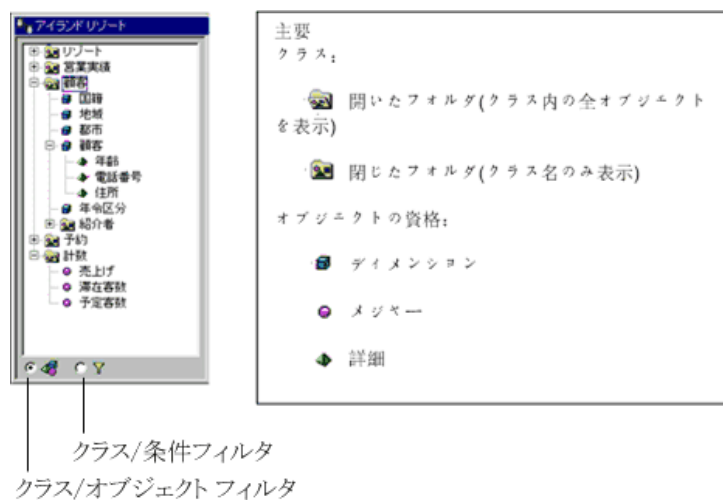
ユニバース枠内に配置するクラスとオブジェクトは、Web Intelligence ユーザーがオブジェクトによって表される情報を取り扱う方法に対応して配置します。

6.3 ユニバース枠の使い方

ユニバース内のクラスとオブジェクトは、[ユニバース] 枠を使って作成します。

[ユニバース]枠には、クラスとオブジェクトが階層表示されます。[ユニバース]枠は、クラスとオブジェクトの表示、作成、編集、再構成に使用します。

次の図は、[ユニバース]枠の例です。クラス名はフォルダアイコンの隣に、オブジェクト名はオブジェクトの資格アイコンの隣にそれぞれ表示されます。



6.3.1 クラスとオブジェクトまたは条件の表示

枠の下にある 2 つのラジオボタンを使い、クラスとオブジェクトを表示するか、条件オブジェクトを表示するかを切り替えられます。条件オブジェクトは、1 つまたは複数の SELECT 文内で使用される定義済みの WHERE 句です。

ユニバース枠の表示形式は、次の 2 種類です。

表示	オプション	表示されるコンポーネント
クラス/オブジェクト	左のラジオボタン	すべてのクラスとオブジェクト

表示	オプション	表示されるコンポーネント
クラス/条件	右のラジオボタン	すべてのクラスと各クラスのオブジェクトに適用される条件

関連項目

- ・ 303 ページの[オブジェクトの制限の定義](#)

6.4 クラス、オブジェクト、条件の基本操作

[ユニバース]枠での操作には、クラス、オブジェクト、条件に共通の基本操作があります。

6.4.1 切り取り、コピー、貼り付け

Windowsの標準コマンドで、選択部分の切り取り、コピー、貼り付けができます。

6.4.2 クラス、オブジェクト、条件の移動

ドラッグアンドドロップで、コンポーネントを他の場所に移動できます。

6.4.3 クラス、オブジェクト、条件の表示と非表示

ユニバース枠では、クラス、オブジェクト、条件の表示 / 非表示を切り替えることができます。これらは、Web Intelligence ユーザには表示されませんが、ユニバースデザインツールでは表示されたままになります。

コンポーネントの非表示機能は、次のような場合に便利な機能です。

- ・ リンクされたユニバースでのコンポーネントが、作業中のユニバースで必要ない場合
- ・ SQL構文を最適化するためだけに使用するオブジェクトをエンドユーザから隠す場合

- ・ 開発のために使用したコンポーネントで、[クエリ]枠には表示したくない場合
- ・ コンポーネントを削除せずに一時的に無効にする場合

6.4.3.1 クラス、オブジェクト、条件を非表示にする

クラス、オブジェクト、条件を非表示にする手順は、次の通りです。

- 1 非表示にするコンポーネントをユニバース枠でクリックします。
- 2 [編集] > [項目を隠す]の順に選択します。

または

[編集]ツールバーの[項目の表示切り替え]ボタンをクリックします。

[ユニバース]枠では、そのコンポーネント名が斜体で表示されます。

6.4.3.2 クラス、オブジェクト、条件の再表示

非表示にしたコンポーネント名は斜体で表示されます。

クラス、オブジェクト、条件を再表示する手順は、次の通りです。

- 1 再表示するコンポーネントを[ユニバース]枠でクリックします。
- 2 [編集] > [項目を表示]の順に選択します。

コンポーネント名が標準に戻り、エンドユーザにも表示されるようになります。エンドユーザには、スキーマは表示されません。

6.5 クラスの定義

クラスとは、オブジェクトのコンテナです。ユニバースのすべてのオブジェクトは、いずれかのクラスに所属している必要があります。クラスは、関連したオブジェクトをグループ化するために使用します。クラスを使うことで、エンドユーザはより簡単にオブジェクトを見つけることができます。新規クラスを作成し、既存クラスのプロパティを編集することができます。クラスは、ユニバース枠内の階層ツリーにフォルダとして表示されます。

ヒント

クラスを有効に使うには、関連するディメンションと詳細オブジェクトをそれぞれグループ化してクラスにまとめ、メジャーは別のクラスに入れてください。サブクラスを作成して、オブジェクトをより細かく分類することもできます。サブクラスについての詳しい説明は、276 ページの「[サブクラスの使い方](#)」を参照してください。

6.5.1 クラスの作成

ユニバース枠にクラスを作成する方法は2種類あります。

- ・ クラスを手動で定義する。
- ・ テーブルスキーマからテーブルをドラッグし、自動作成する。

次に、それぞれの方法を説明します。

6.5.1.1 クラスの手動作成

ユニバース枠では、手動でクラスを作成することができます。ユーザニーズの分析結果に従って、すでに作成すべきオブジェクトをリストアップし、グループ化している場合は、そのリストを使ってクラスを手動作成することが、エンドユーザのニーズに対応したユニバース構造を確保する最良の方法です。

ユニバース枠に初めてクラスを作成する手順は、次のとおりです。

- 1 [挿入]>[クラス]の順に選択します。

または

[クラスの挿入]ボタンをクリックします。

[クラスのプロパティを編集]ダイアログボックスが開きます。

- 2 [クラス名]テキストボックスにクラス名を入力します。
- 3 [説明]テキストボックスに、そのクラスについてのコメントや説明を入力します。
- 4 [OK]をクリックします。

作成したクラスがユニバース枠に表示されます。

ヒント

[OK]の代わりに[適用]をクリックすると、ダイアログボックスを開いたままクラス名と説明だけを確定できます。他のクラスを作成する場合、同じダイアログボックスを使って新しいクラスのプロパティを入力できます。これにより、1つの[プロパティを編集]ダイアログボックスを使って一連のクラスを作成することができます。新しいクラスを作成するたびに別の[プロパティを編集]ダイアログボックスが表示されないので、不必要にクリックを繰り返すことなく、時間の節約にもつながります。

6.5.1.2 既存のクラスのあるユニバース枠でのクラスの作成

すでにクラスのあるユニバース枠にクラスを作成する手順は、次の通りです。

- 1 ユニバース枠で、作成するクラスを挿入する場所のすぐ上にあるクラスをクリックし、[挿入]>[クラス]の順に選択します。新しいクラスが、その下に挿入されます。

または

ユニバース枠で、作成するクラスを挿入する場所のすぐ上にあるクラスをクリックし、[クラスの挿入]ボタンをクリックします。新しいクラスが、その下に挿入されます。

[クラスのプロパティを編集]ダイアログボックスが開きます。

- 2 [クラス名]テキストボックスにクラス名を入力します。
- 3 [OK]をクリックします。

作成したクラスがユニバース枠に表示されます。

6.5.1.3 テーブルスキーマからのクラスの自動作成

テーブルスキーマのテーブルをユニバース枠までドラッグするだけで、自動的にクラスを作成できます。デフォルトでは、テーブル名がそのままクラス名になります。新しいオブジェクトもクラスの下に自動的に作成されます。テーブルに含まれる列が、自動的にクラス内のオブジェクトとして作成されます。

作成されたクラスやオブジェクトのプロパティを編集し、エンドユーザーにわかりやすい名前に変更してください。オブジェクトプロパティの編集方法は、276 ページの「[オブジェクトの定義](#)」を参照してください。

オブジェクトは、[ユニバースのパラメータ] ダイアログボックスの [ストラテジー] ページで選択されたオブジェクト用のストラテジーに従って自動的に作成されます。このページにアクセスするには、[ファイル]>[パラメータ]>[ストラテジー] の順にクリックします。ストラテジーは変更できます。新しいストラテジーを作成し、クラスとオブジェクトの作成プロセスをカスタマイズすることもできます。ストラテジーの詳細については、402 ページの「[外部ストラテジーを使用したユニバース作成のカスタマイズ](#)」および92 ページの「[ストラテジーの選択](#)」を参照してください。

注

クラスとオブジェクトを自動的に作成するということは、データベース構造から直接ユニバースコンポーネントを作成するということですが、クラスとオブジェクトはあくまでユーザニーズの分析結果に基づいて作成し、データベースで利用可能なテーブルと列を直接コピーして作成しないようにしてください。ユーザニーズに基づいたユニバース設計の詳細については、24 ページの「[ユニバースのデザイン手法](#)」を参照してください。

テーブルスキーマから自動的にクラスを作成する手順は、次のとおりです。

- 1 テーブルスキーマでクラスの元になるテーブルを選択します。

- 2 テーブルをユニバース枠までドラッグし、クラス階層の希望する場所にドロップします。

新しいクラスが階層内に作成されます。ユニバース枠にドラッグしたテーブルに含まれる列がオブジェクトとしてクラス内に作成されます。デフォルトでは、クラス名にはテーブル名、オブジェクトには列名がそれぞれ適用されます。

6.5.2 クラスのプロパティ

次のプロパティにクラスを定義できます。

プロパティ	説明
名前	特殊文字を含めることができます。ユニバースで同一名は使用できません。クラス名には、大文字小文字の区別があります。クラスはいつでも名前変更できます。
説明	クラスを説明するコメントです。ここに入力した説明は、クエリパネルに表示されます。このフィールドには、ユーザが使い慣れたビジネス用語を使用して、クエリにふさわしい説明を入力してください。Ctrl キーと Enter キーを押して、改行することもできます。

6.5.3 クラスの変更

クラス名と説明は、[クラスのプロパティを編集] ダイアログボックスでいつでも変更することができます。[(クラス名)のプロパティを編集]ダイアログボックスを開くには、いくつかの方法があります。

- ・ クラスフォルダをダブルクリックする。
- ・ クラスフォルダを右クリックし、[編集]>[クラスのプロパティ]の順に選択する。
- ・ クラスフォルダをクリックし、[編集]>[クラスのプロパティ] の順に選択します。

注

上記の操作でクリックする個所は、クラスフォルダのアイコンでも、クラス名でも構いません。

6.5.4 サブクラスの使い方

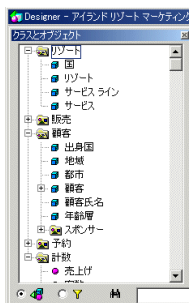
サブクラスとはクラスに含まれるクラスのことです。サブクラスを使うと、関連したオブジェクトをさらに細かくまとめることができます。サブクラスには、さらに下位レベルのサブクラスやオブジェクトを含めることができます。

6.5.4.1 サブクラスの作成

サブクラスを作成する手順は、次の通りです。

- ・ クラスフォルダかクラス名をクリックし、[挿入]>[サブクラス]の順に選択する。
- ・ クラスフォルダかクラス名を右クリックし、ポップアップメニューから[サブクラスの挿入]コマンドを選択する。

次のユニバース枠では、[顧客]クラスの下に[スポンサー]サブクラスが作成されています。



6.6 オブジェクトの定義

オブジェクトとは、ユニバースデータベーススキーマのテーブルにある列をマップするユニバースコンポーネントです。オブジェクトを列に定義された関数にマップすることもできます。

それぞれのオブジェクトは、マップする列あるいは関数に対応したSELECT文を生成します。Web Intelligence ユーザが、クエリパネルでこれらのオブジェクトを使ってクエリを作成すると、各オブジェクトに対応した列や関数を表す SELECT 句が生成され、SELECT 文に組み込まれます。

6.6.1 オブジェクトの作成

[ユニバース]枠内でオブジェクトを作成します。Web Intelligence ユーザは、オブジェクトの名前と資格でオブジェクトを識別します。オブジェクトは、ユニバースペインで手動で作成するか、対応するデータベース構造を構造ペインからユニバースペインにドラッグして自動で作成します。

6.6.1.1 オブジェクトの手動作成

オブジェクトを手動で作成するには、ユニバース枠にオブジェクトを挿入し、そのプロパティを定義します。オブジェクトはいずれかのクラスに属している必要があります。

オブジェクトを手動で作成する手順は、次の通りです。

- 1 ユニバース枠でオブジェクトを作成するクラスあるいはそのクラス内のオブジェクトを右クリックし、ポップアップメニューから[オブジェクトの挿入]を選択します。

または

クラスをクリックしてから、[挿入]メニューの[オブジェクト]を選択するか、編集ツールバーの[オブジェクトの挿入]ボタンをクリックしても構いません。

オブジェクトが選択したクラス内に作成され、[オブジェクトのプロパティを編集]ダイアログボックスが開きます。

- 2 [名前]ボックスにオブジェクトの名前を入力します。

オブジェクトには常に、エンドユーザが日常使用するビジネス用語の名前を付けてください。オブジェクトは、それが表すデータベーススキーマ内の列名にはとらわれずに命名してください。

- 3 [プロパティ]タブをクリックし、オブジェクトのプロパティを選択します。

- 4 [SELECT句]テキストボックスにSELECT句を入力するか、[>>]ボタンをクリックしてSQLエディタを開き、SELECT句を作成します。

- 5 [OK]をクリックします。

関連項目

- ・ [294 ページのSQL エディタの使用](#)
- ・ [278 ページのオブジェクトのプロパティ](#)

6.6.1.2 オブジェクトの自動作成

構造枠のテーブルから列を選択し、それをユニバース枠までドラッグすることで自動的にオブジェクトを作成できます。オブジェクトは、列をドロップした場所にあるクラス内に作成されます。デフォルトのオブジェクト名は、列名になります。アンダースコア(_)は、すべてスペースに置き換えられます。デフォルトのオブジェクトのデータ型は、列のデータ型に対応したものになります。データ型は、[(オブジェクト名)のプロパティを編集]ダイアログボックスの[定義]タブにある[型]のドロップダウンリストボックスを使って変更できます。

作成されたクラスやオブジェクトのプロパティを編集し、エンドユーザのニーズに対応したわかりやすい名前に変更してください。オブジェクトプロパティの編集方法は、276 ページの「[オブジェクトの定義](#)」を参照してください。

オブジェクトは、[ユニバースのパラメータ] ダイアログボックスの [ストラテジー] ページで選択されたオブジェクト用のストラテジーに従って自動的に作成されます。このページにアクセスするには、[ファイル] > [パラメータ] > [ストラテジー] の順にクリックします。ストラテジーは変更できます。新しいストラテジーを作成し、クラスとオブジェクトの作成プロセスをカスタマイズすることもできます。

ストラテジーの使い方の詳細については、402 ページの「[外部ストラテジーを使用したユニバース作成のカスタマイズ](#)」と92 ページの「[ストラテジーの選択](#)」を参照してください。

注

クラスとオブジェクトを自動的に作成するということは、データベース構造から直接ユニバースコンポーネントを作成するということですが、クラスとオブジェクトはあくまでユーザニーズの分析結果に基づいて作成するものですので、データベースにあるテーブルと列を単にコピーするべきではありません。ユーザニーズに基づいたユニバース設計の詳細については、24 ページの「[ユニバースのデザイン手法](#)」を参照してください。

オブジェクトを自動的に作成する手順は、次のとおりです。

- 1 構造枠でテーブルの列をクリックします。
- 2 列をユニバース枠までドラッグし、クラス階層の希望する場所にドロップします。オブジェクトは、既存のクラス内にしか作成できません。

新しいオブジェクトが階層内に作成されます。デフォルトでは、オブジェクトには列名がつけられます。

オブジェクトには常に、エンドユーザが日常使用するビジネス用語の名前を付けてください。オブジェクトは、それが表すデータベーススキーマ内の列名にはとらわれずに命名してください。

6.6.2 オブジェクトのプロパティ

[(オブジェクト) のプロパティを編集] ダイアログボックスでは、次のオブジェクトプロパティを編集できます。

編集できるプロパティ	プロパティ
<p>定義</p> <p>オブジェクトの定義プロパティの詳細は、280 ページの「オブジェクトの定義」を参照してください。</p>	<ul style="list-style-type: none"> ・ 名前 ・ データ型 ・ 説明 ・ SELECT 句 ・ WHERE 句 <p>このページからSQLエディタを開いてSELECT句とWHERE句を定義できます。</p>
<p>プロパティ</p> <p>オブジェクトのプロパティの詳細は、283 ページの「プロパティ」を参照してください。</p>	<ul style="list-style-type: none"> ・ オブジェクトの資格 ・ 値の一覧の関連付け
<p>詳細</p> <p>オブジェクトの詳細プロパティの詳細は、284 ページの「詳細」を参照してください。</p>	<ul style="list-style-type: none"> ・ セキュリティ ・ オブジェクトの使用に関するユーザ権限 ・ 日付形式
<p>キー</p> <p>オブジェクトのキープロパティに関する詳細は、286 ページの「インデックス認識の定義」を参照してください。</p>	<ul style="list-style-type: none"> ・ キーの種類 ・ SELECT 句 ・ WHERE 句 ・ 有効
<p>ソース情報</p> <p>このタブの使い方については、290 ページの「ソース情報」を参照してください。</p>	<ul style="list-style-type: none"> ・ 技術情報 ・ マップ ・ リネージ

オブジェクトのプロパティは、いつでも編集できます。上記の各オブジェクトのプロパティの詳しい説明は、279 ページの「[オブジェクトの変更](#)」を参照してください。

6.6.3 オブジェクトの変更

作成時に定義したオブジェクトのプロパティは、いつでも編集することができます。オブジェクトのプロパティは、[(オブジェクト名)のプロパティを編集]ダイアログボックスで定義します。このダイアログボックスを開くには、オブ

ジェクトをダブルクリックするか、右クリック> [オブジェクトのプロパティ]を選択します。ここでは、[(オブジェクト名)のプロパティを編集]ダイアログボックスのそれぞれのタブで設定できるプロパティについて説明していきます。

6.6.4 オブジェクトの定義

[定義]ページは次のように表示されます。

The screenshot shows a dialog box titled 'Edit Properties' with a 'Definition' tab. It contains the following fields and controls:

- 名前(N):** A text box containing '国'.
- 型(T):** A dropdown menu set to '文字型'.
- 説明(D):** A text box containing 'リゾートが存在する国'.
- SELECT 句(S):** A text box containing 'Resort_Country.country'.
- WHERE 句(W):** An empty text box.
- Navigation buttons (up, down, left, right) are present next to the SELECT and WHERE clauses.
- Buttons at the bottom: 'テーブル(E)...' and '分析(P)'.

[プロパティの編集]ダイアログボックスの[定義]ページで、次のプロパティを定義することができます。

プロパティ	説明	必須/オプション
名前	オブジェクトの名前です。特殊文字とスペースを含む半角英数字を使用できます。名前には、大文字と小文字の区別があります。同じクラス内で、同一のオブジェクト名は使用できません。クラスが異なる場合には、同じ名前を使用できます。	必須

プロパティ	説明	必須/オプション
型	<p>オブジェクトのデータタイプです。次の4つの型のいずれかになります。</p> <ul style="list-style-type: none"> ・ 文字 ・ 日付 ・ ロングテキスト ・ 数値 <p>現バージョンのユニバースデザインツールでは、BLOB はサポートされていません。</p>	必須
説明	<p>オブジェクトのコメントです。ここでの説明はクエリパネルに表示されるので、エンドユーザに有益なオブジェクトに関する情報を入力してください。Ctrl キーと Return キーを押して、ポインタを次の行に移動できます。</p>	オプション
SELECT 文	<p>オブジェクトで指定する SELECT 文です。SQL エディタを使って SELECT 文を作成できます。詳しくは、283 ページの「プロパティ」を参照してください。*</p>	必須
WHERE 句	<p>オブジェクトで指定する SELECT 文の WHERE 句です。WHERE 句で、クエリで返される行数を制限します。SQL エディタを使って WHERE 句を作成できます。*</p>	オプション

* Select 文または Where 句で @Prompt の追加または編集を行うことができます。Select 文または Where 句を右クリックすると、@Prompt が文中にない場合は、ショートカットメニューに[@Prompt の新規作成]が表示されます。既存の @Prompt がある場合は、その @Prompt 内をクリックすると[@Prompt の編集]がショートカットメニューに表示されます。[@Prompt]エディタが開きます。

[テーブル]ボタン

[テーブル] ボタンをクリックすると、スキーマで使用するテーブルの一覧が表示されます。この一覧で、オブジェクトの定義に含める他のテーブルの他の列を選択できます。これにより、オブジェクトで SELECT 文に複数の

テーブルからの列を指定することができます。詳細については、314 ページの「[複数のテーブルによる制限の適用](#)」を参照してください。

[分析]ボタン

[分析]ボタンをクリックすると、オブジェクトの SELECT 文が分析されます。構文エラーが検出されると、エラーを説明するメッセージボックスが表示されます。

関連項目

- ・ 294 ページの[SQL エディタの使用](#)
- ・ 458 ページの[OLAP ユニバースの計算メジャー](#)
- ・ 379 ページの[@Prompt エディタ](#)

6.6.4.1 オブジェクト定義の編集

オブジェクトの定義を編集する手順は、次の通りです。

- 1 オブジェクトをダブルクリックします。
[プロパティの編集]ダイアログボックスの[定義]ページが表示されます。
- 2 必要に応じてオブジェクトの定義とプロパティを入力または選択します。
- 3 [OK]をクリックします。

6.6.4.2 動的ハイパーリンクとしてのオブジェクトの定義

セル内のテキストをハイパーリンクとして定義できます。この方法は、レポートで動的ハイパーリンクを使用するのに役立ちます。この場合、列内のセルテキストが、結果オブジェクトに依存する特定のリソースへのハイパーリンクとして表示されます。

オブジェクトの SELECT 文を編集してハイパーリンク宣言を含め、オブジェクトの[ハイパーリンクとして読み込む]書式プロパティを選択します。

6.6.4.3 動的ハイパーリンクとしてオブジェクトを定義する

この方法を使用すると、オブジェクトによって動的ハイパーリンクが結果のレポートに作成されます。

- 1 オブジェクトを右クリックし、[オブジェクトのプロパティ]を選択します。

[プロパティを編集]ダイアログが表示されます。

- 2 SELECT 文を入力し、適切なハイパーリンクを含めます。
- 3 新規のプロパティを保存します。
- 4 オブジェクトを右クリックし、[オブジェクトの書式]を選択します
[オブジェクトの書式]ペインが表示されます。
- 5 [ハイパーリンクとして読み込む]オプションを選択します。
- 6 [OK]をクリックして、書式設定を保存します。
- 7 レポートツールを使用して、レポートを作成し、リンクをテストします。

レポート内でオブジェクトを使用すると、結果の列にハイパーリンクが含まれます。

例 カレンダー情報へのハイパーリンクの使用

以下の SELECT 文を使用すると、列のセル内にある年の値に応じて timeanddate.com から年のカレンダー情報が取得されます。宣言では、文字列の右から 4 文字(年)までが取得されます。FY(会計年度)は、ターゲット URL で認識できないため削除されることに注意してください。

```
'<a href=http://www.timeanddate.com/calendar/?year=>' +right(@Select(Reservations¥Reservation Year),4)
+ '</a>'
```

6.6.5 プロパティ

[プロパティの編集]ダイアログボックスの[プロパティ]ページで、値の一覧に次のオブジェクトの資格とプロパティを指定することができます。

プロパティ	説明
資格	<p>クエリパネルで使用するときにオブジェクトが実行する定義済みの役割です。オブジェクトには、次の3種類の資格があります。</p> <ul style="list-style-type: none"> ・ デイメンション ・ 詳細 ・ メジャー <p>オブジェクトの資格に関する詳しい説明は、269ページの「ユニバースで使用するオブジェクトの種類」を参照してください。</p>

プロパティ	説明
値の一覧の関連付け	このオプションを選択すると、データ値を含んだファイルとオブジェクトが関連付けられます。デフォルトで選択されています。詳細については、324 ページの「 値の一覧の使い方 」を参照してください。

6.6.5.1 オブジェクトの資格と値の一覧プロパティを設定する

オブジェクトの資格と値の一覧プロパティを設定する手順は、次の通りです。

- 1 オブジェクトをダブルクリックします。

オブジェクトの[プロパティの編集]ダイアログボックスが表示されます。

- 2 [プロパティ]タブをクリックします。

[プロパティ]ページが表示されます。

- 3 [資格]オプションボタンをクリックして、オブジェクトがディメンション、詳細、メジャーのどれであるかを決定します。

返された値の一覧とオブジェクトを関連付けるには、[値の一覧の関連付け]オプションを選択します。

値の一覧の作成と使い方に関する詳しい説明は、324 ページの「[値の一覧の使い方](#)」を参照してください。

- 4 [OK]をクリックします。

6.6.6 詳細

[詳細]ページは次のように表示されます。

セキュリティアクセス レベル(L)
このオブジェクトを使用できるのは次のアクセス レベル以上の権限を持つユーザーだけです。

パブリック ▼

次の項目で使用可能

☒ 結果(R)
☒ 条件(C)
☒ 並べ替え(S)

データベース形式(D)
デフォルトでは、地域設定は以下の形式です。オブジェクトデータの読み込みにほかの形式を指定することもできます。

[プロパティの編集] ダイアログボックスの [詳細] ページで、次のプロパティを定義することができます。

プロパティ	説明
セキュリティアクセスレベル	<p>オブジェクトのセキュリティアクセスレベルを定義します。セキュリティレベルを選択し、ユーザのオブジェクト使用を制限する適切なセキュリティレベルを設定します。</p> <p>設定できるセキュリティアクセスレベルは次の通りです。</p> <ul style="list-style-type: none"> ・ パブリック ・ コントロール ・ リストリクト ・ コンフィデンシャル ・ プライベート <p>パブリックレベルでは、すべてのユーザがそのオブジェクトを使用できます。[リストリクト] を割り当てると、リストリクトまたはそれ以上の権限をもつユーザだけがオブジェクトを表示し、使用できます。</p>
結果項目で使用可能	このオプションを選択すると、オブジェクトはクエリで使用できるようになります。
条件項目で使用可能	このオプションを選択すると、オブジェクトは条件で設定できるようになります。

プロパティ	説明
並べ替え項目で使用可能	このオプションを選択すると、取得した値を並べ替えられるようになります。
データベース形式	<p>日付型のオブジェクトで利用できるオプションです。</p> <p>デフォルトで、オブジェクトの日付書式は MS Windows のコントロールパネルの[地域のオプション]ダイアログボックスで定義されています。ターゲットデータベース形式の日付に適応するように、書式を変更することもできます。たとえば、日付書式をアメリカ形式やヨーロッパ形式にすることもできます。これに関する詳しい説明は、294 ページの「オブジェクト書式の定義」を参照してください。</p>

6.6.6.1 オブジェクトのセキュリティおよびユーザアクセス権の設定

オブジェクトにセキュリティおよびユーザアクセス権を設定する手順は、次のとおりです。

- 1 オブジェクトをダブルクリックします。
オブジェクトの [プロパティの編集] ダイアログボックスが表示されます。
- 2 [詳細設定] タブをクリックします。
[詳細設定] ページが開きます。
- 3 [セキュリティアクセスレベル] ボックスで、適切なアクセスレベルを選択します。
- 4 [次のアイテムで使用可能] ボックスで、適切な選択肢を選択します。
- 5 デフォルトの日付書式を変更する場合は、[データベース形式] ボックスに新しい日付書式を入力します。
- 6 [OK] をクリックします。

6.6.7 インデックス認識の定義

[キー] タブを使用すると、オブジェクトのキー認識を定義できます。キー認識とは、キー列のインデックスを利用して、データの取得を高速化する機能です。

ユニバースデザインツールで作成するオブジェクトは、エンドユーザにとって意味のあるデータベース列を基に作成されます。たとえば[顧客]オブジェクトは、顧客名を記録したフィールドのデータを取得します。こういった場合は通常、Customer テーブルに主キー(たとえば整数)が設定されています。この 1 次キーは、エンドユーザには不要ですが、データベースパフォーマンスには重要です。ユニバースデザインツールでインデックス認識を設定すると、どのデータベース列を 1 次キーおよび外部キーにするか設定できます。これにより、クエリパフォーマンスが大幅に改善されます。

- ・ ユニバースデザインツールでは、キー列のインデックスを活用してデータの取得処理を高速化できます。
- ・ ユニバースデザインツールは、最も効率的なフィルタ処理を行う SQL を生成できます。スター配列スキーマのデータベースでは、このことが特に重要となります。ディメンションテーブルの値にフィルタを適用するクエリを作成すると、ユニバースデザインツールは、ディメンションテーブルの外部キーを使ってファクトテーブルに直接フィルタを適用します。これにより、ディメンションテーブルの不要な結合を排除できます。

ユニバースデザインツールでは、インデックス認識の重複は無視されません。同じ名前の顧客が 2 人いる場合、それぞれに個別の 1 次キーが設定されていなければ、ユニバースデザインツールは 1 人のデータしか取得しません。

例 都市リストからの顧客の検索

この例では、アイランドリゾートマーケティングユニバースを使って、Houston、Dallas、San Francisco、San Diego、Los Angeles の顧客別の売上げを表示するレポートを作成します。これには、クエリパネルで[顧客]と[売上げ]オブジェクトを[結果オブジェクト]枠にドラッグし、[都市]オブジェクトを[条件]枠にドラッグして一覧に表示する都市を制限します。

インデックス認識が設定されていない場合、ユニバースデザインツールでは、以下の SQL が生成されます。

```
SELECT
  Customer.last_name,
  sum(Invoice_Line.days * Invoice_Line.nb_guests * Service.price)
FROM
  Customer,
  Invoice_Line,
  Service,
  City,
  Sales
WHERE
  ( City.city_id=Customer.city_id )
  AND ( Customer.cust_id=Sales.cust_id )
  AND ( Sales.inv_id=Invoice_Line.inv_id )
  AND ( Invoice_Line.service_id=Service.service_id )
  AND (
    City.city IN ('Houston', 'Dallas', 'San Francisco', 'Los Angeles', 'San Diego')
  )
GROUP BY
  Customer.last_name
```

この例では、ユニバースデザインツールで City テーブルへの結合が作成され、取得する都市が制限されています。

インデックス認識を設定すると、ユニバースデザインツールに、city_id が City テーブルの 1 次キーであることが設定され、Customer テーブルの外部キーとしても認識されます。この情報を使い、ユニバースデザインツールは City テーブルに結合を追加せずに都市を制限できます。SQLは、次の通りです。

```
SELECT
  Customer.last_name,
  sum(Invoice_Line.days * Invoice_Line.nb_guests * Service.price)
FROM
  Customer,
  Invoice_Line,
  Service,
  Sales
WHERE
```

```

( Customer.cust_id=Sales.cust_id )
AND ( Sales.inv_id=Invoice_Line.inv_id )
AND ( Invoice_Line.service_id=Service.service_id )
AND (
  Customer.city_id IN (10, 11, 12, 13, 14)
)
GROUP BY
  Customer.last_name

```

この例では、ユニバースデザインツールは、外部キー city_id の値にフィルタを適用するだけで、都市を制限する SQL を生成できます。

6.6.7.1 プライマリキーのインデックス認識の設定

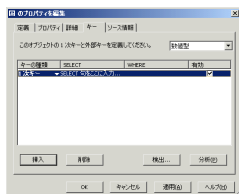
プライマリキーのインデックス認識を設定する手順は、次の通りです。

- 1 インデックス対応とするオブジェクトをクリックし、ポップアップメニューから[オブジェクトのプロパティ]を選択します。

[(オブジェクト名)のプロパティを編集]ダイアログボックスが開きます。

- 2 [キー]タブを選択します。
- 3 [挿入]をクリックします。

[1 次キー]行が[キー]ページに挿入されます。



- 4 以下の操作を行って、1 次キーのインデックスを作成します。

- ・ [キーの種類] から [1 次キー] を選択します。

[SELECT]フィールドの[...]ボタンをクリックし、SQL編集ダイアログボックスを開きます。

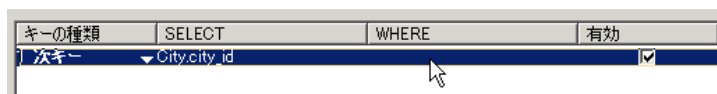


SQL エディタが表示されます。

- ・ SQL エディタを使って 1 次キー SQL SELECT 句を作成するか、直接入力します。たとえば、[都市]オブジェクトの 1 次キー SQL は、"City.city_id" です。

SQL エディタについての詳しい説明は、294 ページの「[SQL エディタの使用](#)」を参照してください。

- ・ 1 次キーのデータ型をドロップダウンリストから選択します。
- 5 WHERE 句を追加する場合、以下の操作を行います。
- ・ [Where]列の下を行内をクリックします。



- ・ [WHERE]フィールドの[...]ボタンをクリックし、SQL 編集ダイアログボックスを開きます。

SQL エディタが表示されます。

- ・ SQL エディタを使って 1 次キー SQL WHERE 句を作成するか、直接入力します。上の例には WHERE 句はありません。
 - ・ キーの型のドロップダウンリストから、[数値型]を選択します。
- 6 [有効]チェックボックスをオンにします。
- 7 [OK]をクリックします。

注

プライマリキーに複数の列を追加するには、複数の列を連結してプライマリキーを定義できます。これらの列は同じテーブルに属し、同じデータ型である必要があります。

たとえば、サンプルデータベース "club.mdb" では、「リゾート」テーブルに Country_id と Resort_id に基づいた複数列のプライマリキーがあります。

したがって、〈リゾート〉オブジェクトにインデックス認識を定義するため、Country_id と Resort_id を連結してプライマリキーを定義できます。

Resort.country_id & Resort.resort_id

& は、MS Access 連結演算子です。

6.6.7.2 外部キー認識の設定

外部キー認識を設定する手順は、次の通りです。

- 1 キー認識を設定するオブジェクトを右クリックします。
メニューから [オブジェクトのプロパティ] を選択します。
[(オブジェクト名)のプロパティを編集]ダイアログボックスが開きます。
- 2 [キー]タブを選択します。
- 3 [挿入]をクリックします。

[キー]ページにキー行が挿入されます。

- 4 以下の動作を行って、この外部キーのキー認識を作成します。
 - ・ [キーの種類] から [外部キー] を選択します。
 - ・ [SELECT]フィールドの[...]ボタンをクリックし、SQL編集ダイアログボックスを開きます。

SQL エディタが表示されます。

- ・ SQL エディタを使って外部キー SQL SELECT 句を作成するか、直接入力します。
- ・ 外部キーのデータ型をドロップダウンリストから選択します。

- 5 外部キーを構成するすべての列について、ステップ 3 から 4 を繰り返します。

- 6 WHERE 句を追加する場合、以下の操作を行います。
 - ・ [WHERE]列の下の反転表示された行をクリックします。
 - ・ [WHERE]フィールドの[...]ボタンをクリックし、SQL 編集ダイアログボックスを開きます。

SQL エディタが表示されます。

- ・ SQL エディタを使って外部キー SQL WHERE 句を作成するか、直接入力します。
- ・ キーの型のドロップダウンリストから、[数値型]を選択します。

- 7 [有効]チェックボックスをオンにします。

- 8 外部キーのすべての列で上のステップを繰り返します。

『インデックス認識の定義』の例では、[キー]タブは、次のようになります。

このオブジェクトの 1 次キーと外部キーを定義してください。 数値型

キーの種類	SELECT	WHERE	有効
1 次キー	City.city_id		<input checked="" type="checkbox"/>
外部キー	▼ Customer.city_id		<input checked="" type="checkbox"/>

挿入 削除 検出... 分析(P)

6.6.8 ソース情報

[ソース情報]ページは、Data Integrator で生成されたユニバースによって使用されます。[ソース情報]タブは次のように表示されます。

The screenshot shows a software window with a tabbed interface. The 'Source Information' (ソース情報) tab is active. It contains three sections, each with a text area and a vertical scrollbar:

- 技術情報 (Technical Information):** The top section.
- マップ (Mapping):** The middle section.
- 系列 (Series):** The bottom section.

Data Integrator から生成されたユニバースの場合、このタブにはソーステーブルからターゲットテーブルへの計算に使用される式と技術的な説明が表示されます。Web Intelligence ユーザは、この情報を利用できるようになります。

[ソース情報]タブでは、次の種類の情報を指定できます。

- ・ 技術的な説明。Data Integrator から生成されたユニバースで利用できる、技術的な説明です。
- ・ マッピングの情報。Data Integrator 内で適用される、ソーステーブルとターゲットテーブルのマッピングです。目標は、マッピングの式を提供するためではなく、説明的なコメントとして表示されます。これにより、オブジェクト定義で使用するソース列のユーザに通知します。
- ・ データリネージ情報。ターゲット列に関連するソース列のリスト。この情報を使って、Data Integrator および Web Intelligence レポートのインパクト分析を簡単に実行できます。

6.6.9 SQL エディタによるオブジェクトの定義

SQL エディタは、オブジェクトに対する SELECT 文または WHERE 句の定義や、OLAP ユニバースオブジェクトに対する MDX 演算子および関数の挿入に使用できます。SQL エディタは、テーブル、列、オブジェクト、演算子、関数をつリー構造で表示するグラフィックエディタです。表示されている構造をダブルクリックして、それを SELECT 句または WHERE 句ボックスに入力できます。

SQL エディタでは、次の編集オプションを使用できます。

編集オプション	説明
テーブルと列	<p>構造枠に表示されるすべてのテーブルとその列。</p> <p>注 このオプションは、リレーショナルユニバース専用で、OLAP ユニバースには利用できません。</p>
クラスとオブジェクト	ユニバース枠に表示されるすべてのクラスとそのオブジェクト。
演算子	SELECT 句の SQL 構造で組み合わせる、または WHERE 句で条件を設定するときに使用できる演算子。
関数	<ul style="list-style-type: none"> 数値関数、文字関数、日付関数などのデータベース関数 Business Objects 固有の @ 関数 <p>使用できる関数は、接続するデータベース用パラメータファイル (.PRM) の Functions セクションに設定されている関数です。サポートされているデータベースごとに .prm ファイルがあります。ファイルは、BusinessObjects パスの Data Access フォルダに格納されています。.PRM ファイルを編集して、使用できる関数を編集したり、追加したりすることができます。</p> <p>PRM ファイルの編集については、『データアクセスガイド』で説明されています。</p>
オブジェクト SQL の表示	このオプションを選択すると、SELECT 句、または WHERE 句ボックスに表示されるオブジェクトの SQL 文が表示されます。
分析	クリックすると、シンタックスを分析します。シンタックスが無効の場合、問題を説明するメッセージボックスが表示されます。
説明	選択したオブジェクトまたは関数の説明が表示されます。

関連項目

- 459 ページの[キューブクエリの MDX 関数について](#)
- 294 ページの[SQL エディタの使用](#)

6.6.9.1 キューブクエリの MDX 関数について

MDX エディタを使用して、キューブクエリを定義します。

新規オブジェクトまたは定義済みのフィルタを OLAP ユニバースに追加する際に、特定のデータソース接続に対応した、サポートされている MDX 式の一覧を使用できます。

使用できる式のライブラリが、.prm 接続ファイルに保存されています。オブジェクトの[プロパティの編集]ペインを開き、クエリの[SELECT 句の編集]ペインを開くと、使用できる式が[関数]ペインに表示されます。SELECT 句または WHERE 句に式を挿入するには、対象の句を、式を挿入する位置でクリックし、適切な式をダブルクリックします。

OLAP ユニバース MDX 辞書 - 関数一覧(PRM ファイル)

新規のオブジェクトまたは定義済みのフィルタを OLAP ユニバースに追加する際に、式で使用する MDX 関数(主にメンバー関数)と MDX 演算子の明示的な一覧が、適切な OLAP 接続(SAP または MSAS)のオブジェクトエディタおよびフィルタエディタで使用可能になります。SAP または mySQL(sap.prm, sqlsrv_as.prm)の接続の設定方法については、『データアクセスガイド』を参照してください。使用できる関数と演算子は、ユニバースの接続の種類によって異なります。この関数一覧は、接続ごとに PRM ファイルで提供されます。サポートされている全関数の一覧が提供されるのではなく、特に頻繁に使用される関数のみが提供されます。

以下の MDX 演算子がクエリで使用できます。

- ・ Equal
- ・ NotEqual
- ・ InList
- ・ NotInList
- ・ Greater
- ・ GreaterOrEqual
- ・ Less
- ・ LessOrEqual
- ・ Between
- ・ NotBetween
- ・ Like
- ・ NotLike

以下の一覧は、条件の編集に使用できる MDX フォルダ関数の例です。使用できる関数は、基盤となるデータベースによって異なります。

- ・ 設定関数(ADDCALCULATEDMEMBERS, ALLMEMBERSなど)
- ・ 統計/数値関数(AGGREGATE, AVG など)
- ・ ナビゲーション/メンバー関数(ANCESTOR, ASCENDANTS など)
- ・ メタデータ関数(AXIS, HIERARCHY など)

6.6.9.2 SQL エディタの使用

SQL エディタを使用して、オブジェクト定義に SQL 式および MDX 式を挿入できます。また、SQL 文を右クリックして、[@Prompt の新規作成]を選択し、SQL に @Prompt 式を追加するか、または[@Prompt の編集]を選択し、既存の @Prompt 式を編集することもできます。これによって、@Prompt エディタが開きます。

SQLエディタの使用手順は、次の通りです。

- 1 オブジェクトをダブルクリックします。

オブジェクトの[プロパティを編集]ダイアログボックスが表示されます。

- 2 [SELECT 句]あるいは[WHERE 句]ボックスの横にある[>>]ボタンをクリックします。

[(オブジェクト名)の SELECT 句の編集]あるいは[(オブジェクト名)の WHERE 句の編集]ダイアログボックスが表示されます。

- 3 SELECT 句または WHERE 句の構造に構文を追加する位置をクリックします。ボックスが空の場合、ボックス内をクリックします。カーソルがボックスの左上に表示されます。
- 4 テーブルノードを拡張して、列を表示します。
- 5 列をダブルクリックすると、SELECT 句あるいは WHERE 句内のカーソル位置にその列の定義が挿入されます。

ヒント

選択した列に値の一覧から 1 つまたは複数の値を選択するには、列を右クリックして[値の一覧]を選択します。

- 6 クラスノードを拡張して、オブジェクトを表示します。
- 7 オブジェクトをダブルクリックして、SELECT 句または WHERE 句に @Select または @Where 関数を挿入します。これらの関数で、現在のオブジェクトに選択したオブジェクトの SELECT 句または WHERE 句を使用するように指示します。@関数についての詳しい説明は、375 ページの「[オブジェクトの SQL での @ 関数の使用](#)」を参照してください。
- 8 演算子をダブルクリックして、編集ボックスに演算子を挿入します。
- 9 関数ノードを拡張して、使用できる関数を表示します。
- 10 関数をダブルクリックして、編集ボックスに関数を挿入します。
- 11 [分析]ボタンをクリックしてシンタックスをチェックします。
- 12 [OK]をクリックします。

6.6.10 オブジェクト書式の定義

選択したオブジェクトのデータ値の表示書式を設定できます。設定した書式は、Web Intelligence レポートのセルに表示された関連するデータ値に適用されます。

[オブジェクトの書式]ダイアログボックスには、[数値]、[配置]、[フォント]、[罫線]、[網かけ]の5つのタブがあります。

たとえば、整数値をデフォルト書式の 1,000.00 ではなく、\$1,000 と表示したり、目立たせる値を赤で表示することができます。

[数値]ページにある[数値]、[通貨]、[指数]、[パーセンテージ]のカテゴリは、数値型の変数とオブジェクトだけに、[日付/時刻]カテゴリは、日付型の変数とオブジェクトだけに適用できます。

書式に関する情報は、ユニバースと一緒にインポート/エクスポートされます。

定義した書式を元に戻すには、[オブジェクトの書式のリセット]コマンドを使います。

関連項目

- ・ 459 ページの[キューブクエリの MDX 関数について](#)

6.6.10.1 オブジェクトの書式の設定

オブジェクトに書式を設定する手順は、次のとおりです。

- 1 書式を設定するオブジェクトを右クリックします。
- 2 ポップアップメニューから[オブジェクトの書式]を選択します。
[オブジェクトの書式]ダイアログボックスが開きます。
- 3 対応するタブをクリックし、書式を選択あるいは入力します。
- 4 [OK]をクリックします。

6.6.10.2 オブジェクト書式のリセット

オブジェクトに設定した書式は、いつでも元に戻すことができます。

オブジェクトの書式を解除する方法は、次のとおりです。

- ・ オブジェクトを選択し、[編集]>[オブジェクトの書式のリセット]の順に選択する。
または
- ・ オブジェクトを右クリックし、ポップアップメニューから、[オブジェクトの書式のリセット]を選択する。

6.6.11 オブジェクトの定義に使用したテーブルの表示

ユニバース枠のオブジェクトの定義に使用した、構造枠にあるテーブルを表示できます。これは、テーブルを簡単に識別できるので、オブジェクト名からオブジェクトが使用しているテーブルをすぐに判断できないときに便利な機能です。

6.6.11.1 オブジェクトが使用しているテーブルの表示

オブジェクトが使用しているテーブルを表示する手順は、次の通りです。

- 1 ユニバース枠で、オブジェクトを右クリックします。
ポップアップメニューが表示されます。
- 2 ポップアップメニューから[関連付けられたテーブルの表示]を選択します。
構造枠で、関連付けられたテーブルが反転表示されます。

6.6.12 ディメンションの定義

ディメンションは、クエリによる分析の焦点となるオブジェクトです。ディメンションは、クエリのキー要素となる、データベース内の1つあるいは複数の列または関数にマップされます。一般的に“国”、“販売員”、“製品”、“製品区分”などがディメンションになります。

オブジェクトの作成時に、デフォルトでディメンションの資格が設定されます。オブジェクトの資格は、いつでも変更できます。

オブジェクトをディメンションとして定義する手順は、次のとおりです。

- 1 オブジェクトをダブルクリックします。
オブジェクトの[プロパティを編集]ダイアログボックスが表示されます。
- 2 [プロパティ]タブをクリックします。
[プロパティ]ページが表示されます。
- 3 [資格]グループボックスから[ディメンション]ラジオボタンをクリックします。
- 4 [OK]をクリックします。

6.6.13 詳細の定義

詳細オブジェクトは、ディメンションについての補足データを提供します。詳細オブジェクトは常にディメンションに従属します。ディメンションに関連付けられた詳細情報を提供する、データベース内の 1 つあるいは複数の列または関数にマップされます。

詳細オブジェクトを定義するには、オブジェクトの資格を[詳細]に設定し、それが従属するディメンションを指定する必要があります。

オブジェクトを詳細として定義する手順は、次のとおりです。

- 1 オブジェクトをダブルクリックします。

オブジェクトの[プロパティを編集]ダイアログボックスが表示されます。

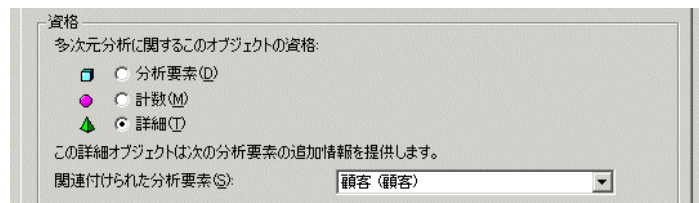
- 2 [プロパティ]タブをクリックします。

[プロパティ]ページが表示されます。

- 3 [資格]グループボックスから[詳細]ラジオボタンをクリックします。

ユニバース内のすべてのディメンションオブジェクトのドロップダウンリストを持った[関連付けられたディメンション]が表示されます。

- 4 ドロップダウンリストからディメンションを選択します。選択したディメンションの資格またはプロパティの詳細が説明されます。



- 5 [OK]をクリックします。

6.6.14 メジャーの定義

メジャーオブジェクトを定義するには、オブジェクトの資格を[メジャー]に設定します。メジャーオブジェクトは大変柔軟なオブジェクトで、動的に変化します。メジャーオブジェクトが返す値は、同じクエリに含まれる他のディメンションと詳細オブジェクトによって変わります。たとえば、“国”オブジェクトを含んだクエリで返されるメジャー“売上げ”の値は、“国”と“地域”を含んだクエリで返される“売上げ”の値と同じではありません。

メジャーオブジェクトは、ディメンションや詳細よりも複雑でパワフルなオブジェクトですので、次のセクションで詳しく説明します。

6.6.14.1 メジャーが返すデータの種類

メジャーオブジェクトは、数値情報を返します。メジャーは、集計関数を使って作成します。最もよく使われる集計関数は、次の5つです。

- ・ Sum
- ・ Count
- ・ Average
- ・ Minimum
- ・ Maximum

6.6.14.2 ディメンション、詳細とメジャーとの違い

メジャーは、次の点でディメンションや詳細オブジェクトとは異なっています。

- ・ メジャーは動的なオブジェクトである。
- ・ メジャーは集計を考慮できる。

この2つの性質を詳しく説明します。

6.6.14.3 動的なオブジェクトとしてのメジャー

メジャーオブジェクトが返す値は、同じクエリに含まれる他のディメンションと詳細オブジェクトによって変わります。

次の図は、ディメンションの異なる2つのクエリに含まれる“売上げ”メジャーが異なる値を返す例です。

年	売上げ
FY1998	287,929.00
FY1999	307,400.00
FY2000	376,115.00

同じメジャーでも、異なる値を返します。

サービス	売上げ
Activities	22,400.00
Bungalow	59,040.00
Excursion	15,300.00
Fast Food	5,840.00
Hotel Room	60,288.00
Hotel Suite	115,632.00
Poolside Bar	14,280.00
Restaurant	75,335.00
Sports	8,000.00

6.6.14.4 メジャーが生成する Group By 句

メジャーオブジェクトを他の資格のオブジェクトと組み合わせてクエリを実行すると、SELECT 文に自動的に Group By 句が挿入されます。

Group By句は、次のSQL規則に従って生成されます。

SELECT句に集計関数が含まれる場合、集計関数以外のすべてのオブジェクトをGroup By句で指定する。

この規則に従い、メジャーと同じクエリに含まれるすべてのディメンションと詳細オブジェクトは、生成される Group By 句に自動的に挿入されます。クエリが正しい結果を返すには、ディメンションと詳細オブジェクトの定義に集計関数が含まれていないことが必要です。

次の例では、“リゾート”、“サービスライン”、“年”のディメンションオブジェクトが SELECT 句と Group By 句の両方で指定されています。

結果オブジェクト

レポート サービスライン 年 売上げ

条件

レポート 選択 (0) Bahamas Beach

Bahamas Beach

年	サービスライン	売上げ
FY1998	Accommodation	213,464.00
FY1998	Food & Drinks	35,865.00
FY1998	Recreation	38,600.00
FY1999	Accommodation	225,240.00
FY1999	Food & Drinks	38,360.00
FY1999	Recreation	43,800.00
FY2000	Accommodation	234,960.00
FY2000	Food & Drinks	95,455.00
FY2000	Recreation	45,700.00

GROUP BY で指定されたディメンション

```

SELECT
  Resort.resort,
  Service_Line.service_line,
  FY*Format(Sales.invoice_date,'YY'),
  sum(Invoice_Line.days * Invoice_Line.nb_guests * Service.price)
FROM
  Resort,
  Service_Line,
  Sales,
  Invoice_Line,
  Service
WHERE
  (Invoice_Line.inv_id=Sales.inv_id )
  AND (Invoice_Line.service_id=Service.service_id )
  AND (Resort.resort_id=Service_Line.resort_id )
  AND (Service.sl_id=Service_Line.sl_id )
  AND (
    Resort.resort = 'Bahamas Beach'
  )
GROUP BY
  Resort.resort,
  Service_Line.service_line,
  FY*Format(Sales.invoice_date,'YY')

```

結果は最低レベルの“リゾート”で集計され、次に“サービスライン”、“年”で集計されています。

注

クエリにメジャーオブジェクトしか指定されていない場合は、Group By 句は生成されません。

6.6.14.5 メジャーに集計機能を設定

メジャーを作成する際は、集計関数をレポートにどのように投影するかを設定する必要があります。

メジャーオブジェクトが返す値は、クエリ処理の次の 2 つのレベルで集計されます。

- ・ クエリレベル: 生成される SELECT 文を使ってデータを集計します。
- ・ マイクロキューブからレポートのブロックレベル: データをマイクロキューブからレポート内のブロックに投影する際、データはマイクロキューブ内でローカル集計されます。

注

マイクロキューブとは、レポートに投影する前のクエリ結果の状態を表す概念で、BusinessObjects レポーティング製品がメモリ内に保持する取得データを意味します。ブロックレベルとは、取得したデータを使ってユーザーが作成する平面(二次元)レポートを意味します。レポートの作成には、マイクロキューブ内のすべてのデータを使うこともできますし、一部のデータだけを使うこともできます。マイクロキューブ内のデータに集計関数を適用(ローカル集計)してレポート内に新しいデータを作成することもできます。

次の図は、クエリにおける 2 レベルの集計処理を図解したものです。

- ・ ユーザーは Web Intelligence でクエリーを作成する。
- ・ Web Intelligence がクエリーに対応する SQL を生成し、データベースに SELECT 文を送信する。
- ・ データがマイクロキューブに返される。これが第 1 の集計レベルです。
- ・ マイクロキューブが集計されたデータをレポートに投影する。データはクエリパネルでさらに再集計されます。これが第 2 の集計レベルです。

クエリを実行すると、SELECT 文の結果セットがマイクロキューブに保存され、マイクロキューブ内のすべてのデータがブロックに投影されます。データは、マイクロキューブに保存された最下位レベルに合わせて投影されますので、この時点では集計は行われません。

クエリパネルを使用してユーザがレポートを変更し、マイクロキューブの最下位レベルのデータを表示する必要がなくなると、データが上位レベルで集計されるよう、ローカル集計が行われます。

たとえば前ページの例では、ブロックから[年]オブジェクトを削除すると各年度の売り上げを表示する3行が不要となり、データがローカルで集計(この場合は合計)されてレポートの通算売り上げが表示されます。

ローカル集計の種類は、メジャーの[プロパティを編集]ダイアログボックスの[プロパティ]ページで指定します。このページを開くには、オブジェクトを右クリック>[オブジェクトのプロパティ]>[プロパティ]タブをクリックします。

ローカル集計は、SELECT 集計とは異なりますのでご注意ください。

関連項目

- 454 ページの[依頼されたデータベース投影関数](#)

6.6.14.5.1 依頼されたデータベース投影関数

ユニバースでは、メジャーに投影関数(Sum、Min、Max、Count、および Avg)を含めることはできません。投影関数は、レポートに表示されるディメンション数がクエリの結果セットに含まれるディメンション数よりも小さい場合、Web Intelligence でローカルにメジャーを集計するために使用されます。

比率、平均、重量などの非追加メジャーは、クエリの結果セットと同じ集計レベルに表示されるだけです。したがって、非追加メジャーの投影関数は、通常、ユニバースで[なし]に設定されます。

依頼されたデータベース投影関数を使用すると、非追加メジャーの集計をデータベースサーバに委任できます。これらは、Web Intelligence ではスマートメジャーと呼ばれています。スマートメジャーの投影関数は、オブジェクトのプロパティの [プロパティ] ページで [依頼されたデータベース] に設定されます。Web Intelligence でのこれらの関数およびその他の関数の使用方法について詳しくは、『Web Intelligence における式、関数、計算の使い方』の『スマート メジャーを使用した値の計算』の節を参照してください。

注

MSAS または Essbase データソースに基づくOLAP ユニバースについては、スマートメジャーはすべてユニバースに作成され、投影関数はデフォルトで依頼されたデータベースに設定されます。

注

アグリゲート認識セットを含むメジャーに基づくスマートメジャーを使用する場合、次の制約に留意してください。メジャー定義で使用される集計テーブルのデータに整合性がある、つまり集計値が詳細値に関して正確であることを確認することを強くお勧めします。データに整合性がない場合、スマートメジャーで不整合なデータが生成されることがあります。たとえば、スマートメジャーに対して年集計テーブルと日集計テーブルが使用される場合、過去の年については年集計テーブルと日集計テーブルに整合性があります。しかし、現在の年について日テーブルは日次ベースで正確ですが、年テーブルは空白である可能性があります。この場合、現在の年テーブルおよび日テーブルに基づくスマートメジャーを使用するレポートは、整合性のない結果をもたらす可能性があります。

例 スマートメジャー

この例では、クエリに2つのディメンション(“国”および“地域”)、3つのメジャー(“注文金額”、“納品金額”、“納品割合”)が含まれています。

L01 地域	納品金額	注文金額	納品割合
Reg1	497,318,880	497,332,680	99.997
Reg2	199,463,776	199,466,536	99.998
Reg3	198,927,552	198,933,072	99.997
		合計:	299.992

納品割合の合計は、納品割合の列が合計されているので、誤りです。

このメジャーの投影関数が[依頼されたデータベース]にユニバースで設定されている場合、ユーザがレポートを最新表示した際に、Web Intelligence がデータベースに接続して正しい値を計算します。

L01 地域	納品金額	注文金額	納品割合
Reg1	497,318,880	497,332,680	99.997
Reg2	199,463,776	199,466,536	99.998
Reg3	198,927,552	198,933,072	99.997
		合計:	299.992
		合計:	99.997

注

Ratio 関数(Average)など一部の関数の使用には、注意が必要です。列から平均を計算する場合、この関数を正しく設定しないと予期しない計算が行われることがあります。

たとえば、 $\text{sum}(\text{Shop_facts.Margin})/\text{sum}(\text{Shop.facts.Quantity_sold})$ という SQL 関数では、予期しない結果が返されることがあります。正しく設定しなかった場合、各セルの平均が計算され、これらの平均の合計が返されます。修正するには、次の手順に従って関数のパラメータ化を実行する必要があります。

- 1 該当する関数の[プロパティの編集]に移動します。

- 2 [このメジャーは次の方法で集計されます]オプションで、[関数]ドロップダウンリストから[Db delegated]関数を選択します。
 - 3 変更を保存します。
-

関連項目

- ・ 457 ページの[メジャーに集計機能を設定](#)」

6.6.14.6 メジャーの定義

メジャーを定義する

- 1 オブジェクトをダブルクリックします。
オブジェクトの[プロパティを編集]ダイアログボックスが表示されます。
- 2 [プロパティ]タブをクリックします。
[プロパティ]ページが表示されます。
- 3 [資格]グループボックスから[メジャー]ラジオボタンをクリックします。
設定できる集計関数の一覧を持った[関数]ドロップダウンリストボックスが表示されます。
- 4 関数を選択します。
- 5 [OK]をクリックします。

関連項目

- ・ 457 ページの[メジャーに集計機能を設定](#)」
- ・ 454 ページの[依頼されたデータベース投影関数](#)」

6.6.15 オブジェクトの制限の定義

オブジェクトの制限とは、クエリが返すデータを制限するための基準を定義するSQL内の条件です。

オブジェクトの制限は、ユーザが使用できるデータを制限するために設定します。この制限は、あくまでそのオブジェクトを使用するユーザのデータニーズに基づいたものでなければなりません。ユーザは、必ずしもオブジェクトが返すすべてのデータを必要としているわけではありません。また、セキュリティ上の理由から、特定のデータへのアクセスを制限する場合があります。

ユニバースでは、次の2種類の制限を設定できます。

制限の種類	説明
強制	オブジェクトの WHERE 句で定義する制限: ユーザは制限されたデータにはアクセスできず、Web Intelligence でこの制限を回避することはできません。
オプション	ユーザが任意にクエリに挿入できる条件オブジェクト: 条件オブジェクトとは、クエリパネルのオブジェクトで指定された SELECT 文に挿入できる、定義済みの WHERE 句です。

注

Web Intelligence では、ユーザはクエリパネルで条件を適用できます。ユーザレベルで簡単に作成できるような制限を定義した条件オブジェクトを準備する必要はありません。単純な条件は、ユーザ自身が必要に応じて設定できます。

6.6.15.1 オブジェクトの WHERE 句の定義

[(オブジェクト名の)プロパティを編集]ダイアログボックスの[定義]ページにある[WHERE 句]ボックスに条件を追加することで、オブジェクトに制限を適用することができます。

条件はオブジェクトの作成時だけでなく、いつでも追加することができます。

ユニバースでは、SQL文に含めるWHERE句を使ってクエリが返すデータを制限する方法が2つあります。

- ・ スキーマ内のテーブルをリンクする結合に従って生成される WHERE 句は、自動的にオブジェクトの SELECT 文に挿入されます。結合は通常、テーブル間の等結合です。テーブルの結合を使って取得データを制限することで、デカルト積を回避します。
- ・ オブジェクトに WHERE 句を設定して、条件を追加できます。これは、結合によって生成される WHERE 句に追加される条件となります。たとえば、ユーザが取得できる情報をデータのサブ集合だけに限定する場合などは、オブジェクトに WHERE 句を設定してクエリが返すデータを制限します。

例 オブジェクトのデフォルトのWhere句(結合のみ)を変更する

次のレポートは、すべての国の販売員を制限なしに取得したものです。

販売員	出身国
Barrot	France
Carlin	France
Edwood	UK
Fischer	Germany
Galagers	US
Ishimoto	Japan
Nagata	Japan

このクエリのSQLは次の通りです。WHERE句には、テーブル[Customer]、[City]、[Region]、[Sales_Person]間の結合による制限だけが含まれています。

```
SELECT
  Sales_Person.sales_person, Country.country
FROM
  Sales_Person,
  Country,
  Region,
  City,
  Customer
WHERE
  ( City.city_id=Customer.city_id )
  AND ( City.region_id=Region.region_id )
  AND ( Country.country_id=Region.country_id )
  AND ( Sales_Person.sales_id=Customer.sales_id )
```

これを変更し、[出身国] が「France」のデータだけに制限するには、[出身国] オブジェクトに WHERE 句の条件を追加します。次のレポートは、フランス人の顧客を持つ販売員だけを表示した例です。

販売員	出身国
Barrot	France
Carlin	France

このクエリのSQLは次の通りです。

```
SELECT
  Sales_Person.sales_person,
  Country.country
FROM
  Sales_Person,
  Country,
  Region,
  City,
  Customer
WHERE
  ( City.city_id=Customer.city_id )
  AND ( City.region_id=Region.region_id )
  AND ( Country.country_id=Region.country_id )
  AND ( Sales_Person.sales_id=Customer.sales_id )
  AND ( Country.country = 'France' )
```

WHERE 句に 1 行追加されています。この追加された行が、[出身国]オブジェクトに追加した WHERE 句による制限です。

注

WHERE 句では、自己結合以外の結合は作成しないでください。WHERE 句に定義した結合は、コンテキストの(自動)検出機能や集計テーブルの非互換検出機能では考慮されません。すべての結合は、[構造]枠で設定してください。これにより、すべての結合がユニバースデザインツールの自動検出機能の対象になります。

6.6.15.2 WHERE句の定義

WHERE句を定義する手順は、次の通りです。

- 1 オブジェクトをダブルクリックします。

[プロパティの編集]ダイアログボックスの[定義]ページが表示されます。

- 2 [WHERE句]ボックスに直接構文を入力します。

または

[>>]ボタンをクリックして SQL エディタを開きます。

- 3 SQL 構造と機能の一覧に表示される列、オブジェクト、演算子、関数をダブルクリックします。

ヒント

WHERE 句の値の選択方法: [テーブルと列]の一覧で列を右クリックし、[値の一覧]を選択します。列の値の一覧が表示されます。IN 演算子を使用する場合などは、複数の値を選択して WHERE 句に挿入できます。

- 4 [OK]をクリックしてエディタを閉じます。

次の図は、WHERE 句を使って取得するデータを「France」だけに制限する[出身国]オブジェクトの例です。

The screenshot shows a dialog box titled 'Define WHERE Clause'. It contains the following fields and controls:

- 名前(N):** 出身国
- 型(T):** 文字型
- 説明(D):** 販売員の出身国
- SELECT 句(S):** Country.country
- WHERE 句(W):** Country.country = 'France'

There are '>>' and '<<' buttons between the SELECT and WHERE clauses. At the bottom, there are buttons for 'テーブル(B)...' and '分析(P)'.

- 5 [OK]をクリックします。

6.6.15.3 WHERE 句に関する注意事項

WHERE句はデータの制限に効果的な方法ですが、ユニバースに次の問題が発生しないよう注意して使用する必要があります。

問題	説明	解決策
類似したオブジェクトが多数存在する。	WHERE 句を使って取得するデータを制限する一連のオブジェクトを作成すると、複数の類似した名前のオブジェクトが作成されてしまいます。たとえば“顧客(日本)”、“顧客(アメリカ)”、“顧客(フランス)”など、類似したオブジェクトが多数あるとユーザが混乱します。	それぞれの制限に対応した条件オブジェクトを作成します。
階層の作成が困難である。	同じデータに対するWHERE句を持ったオブジェクトが多数あると、ドリルダウンで使用するデフォルト論理階層の構築が難しくなります。	それぞれの制限に対応した条件オブジェクトを作成します。
オブジェクト名から制限の内容を判断しにくい。	オブジェクト名をかなり工夫して付けない限り、ユーザがオブジェクト名から制限された内容を判断することは難しくなります。ユーザは、クエリのSQLを表示してWHERE句の内容を確認することもできますが、クエリの実行前にSQLを確認することをユーザに期待するのは無理があります。	<ul style="list-style-type: none"> それぞれの制限に対応した条件オブジェクトを作成します。 制限の内容が明確なオブジェクト名を付けます。
WHERE句の競合が発生する。	同じクエリに、制限内容の類似した複数のオブジェクトが含まれると、WHERE句の競合が発生し、取得できるデータが無くなる恐れがあります。	それぞれの制限に対応した条件オブジェクトを作成し、レポートレベルでクエリを同期するか、クエリ結果を結合するようユーザへの指示を徹底します。

条件オブジェクトを作成することで、類似オブジェクト、階層の作成、不明瞭なオブジェクト名の問題は解決できます。

WHERE句の競合は、条件オブジェクトを作成し、レポートレベルで同期処理やユニオン演算子を使ってクエリ結果を結合する必要があることをユーザに知らせることで、解決できます。

オブジェクトの定義にWHERE句を使用することで発生する潜在的な問題を考慮し、オブジェクトにWHERE句を設定する代わりに、できる限り条件オブジェクトを作成してください。条件オブジェクトを使用すると、ハードコード化されたWHERE句で発生する問題を軽減できます。

注

条件オブジェクトでは、自己結合以外の結合は作成しないでください。条件オブジェクトに結合を定義すると、再利用可能なWHERE句に結合を定義することになりますので、コンテキストの検出機能や集計テーブルの非

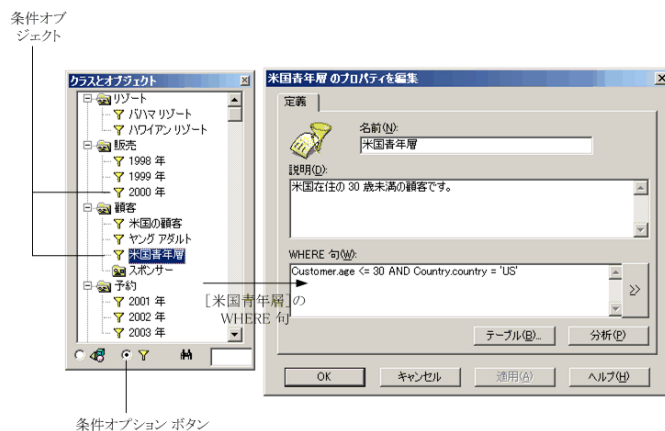
互換検出機能では考慮されません。すべての結合は、構造枠で設定してください。これにより、すべての結合が自動検出機能の対象になります。

6.6.16 条件オブジェクトの定義

条件オブジェクトとは、クエリパネルのオブジェクトで指定された SELECT 文に挿入できる、定義済みの WHERE 句です。

条件オブジェクトは、ユニバース枠の [条件] 表示に表示されます。ユニバース枠の左下にある [条件] オプションボタンをクリックして、[条件] ビューにアクセスします。

次の図は、「アイランドリゾート」ユニバースの条件オブジェクトの一覧と、[米国青年層] の WHERE 句を示したものです。



6.6.16.1 条件オブジェクトの利点と欠点

条件オブジェクトの使用には、次のような利点があります。

- ・ 複雑または頻繁に使用する条件がすでに定義されている。
- ・ 適用する条件が選択肢として一覧表示される。
- ・ オブジェクトが重複しない。
- ・ 条件オブジェクトは、ユニバース枠のクラスとオブジェクトの表示を変更しません。

注

クエリパネルやWebパネルで条件オブジェクトを表示するようユーザを指導する必要があります。

条件オブジェクトの唯一の欠点は、データの一部にしかアクセスできないような制限をユーザに強制できないことです。制限を強制する場合は、オブジェクトにWHERE句を定義する必要があります。

6.6.16.2 条件オブジェクトでは WHERE 句の競合が解決されない

条件オブジェクトでは、WHERE句の競合によってフェッチするデータがなくなる問題は解決できません。同じデータにアクセスする2つの条件オブジェクトを含んだクエリが実行されると、2つの条件がAND演算子で結合されるため、両方の条件を満足するデータがない限り、情報を取得できません。各条件オブジェクトのクエリおよび複合クエリの2つのクエリを作成しているユーザは、レポートレベルでこの問題を解決することができます。

6.6.16.3 必須フィルタ

必須フィルタには、次の2種類があります。

- ・ ユニバース: ユニバース必須フィルタは、所属するクラスに依存しません。ユニバース必須フィルタは、クエリに含まれるオブジェクト(ディメンション、メジャー、および詳細)とは無関係に、クエリに含まれます。

NetWeaver Business Warehouse(BW)でOLAPユニバースが生成される場合、大部分のSAP NetWeaverBW変数は、ユニバース必須フィルタとして作成されます。

- ・ クラス: クラス必須フィルタは、オブジェクトのクラスのアイテムがクエリで使用された場合のみ表示されます。

クラス必須フィルタは、ユーザが次の操作を行った場合に呼び出されます。

- ・ オブジェクト(ディメンション、メジャー、または詳細)をWeb Intelligenceの[クエリー]パネルの[結果]枠に追加した場合。
- ・ 同じクラスに属するオブジェクトが[結果]枠で選択されていなくても、定義済みのユニバースフィルタを[クエリー]パネルの[フィルタ]に追加した場合。
- ・ 必須フィルタが適用されたクラスに属するオブジェクト(ディメンション、メジャー、または詳細)でフィルタを作成した場合。

必須フィルタは、デフォルト値を指定したり、値の一覧に関連付けたりすることができます。

必須フィルタは表示されません。Web Intelligenceの[クエリー]パネルで選択することもできません。ユニバースデザインツールでは、クエリでフィルタを必須に設定すると、そのフィルタは自動的に表示されなくなり、[項目を表示]コマンドが無効になります。必須オプションのチェックを外すと、フィルタは表示されるようになります。[項目を隠す]コマンドが有効になります。

エンドユーザクエリには、複数の必須フィルタを含めることができます。デフォルトでは、クエリ内ですべての必須フィルタはAND演算子で結合されます。

すべてのサブクラスは、親クラスから必須フィルタを継承します。ただし、以下の点に注意してください。

- ・ @SELECT 関数を含む別のオブジェクトを参照するオブジェクト(ディメンション、メジャー、詳細)は、参照オブジェクトのクラス必須フィルタを継承しません。
- ・ @WHERE 関数を含む別のオブジェクトの WHERE 句を参照するオブジェクトの WHERE 句は、参照オブジェクトのクラス必須フィルタを継承しません。
- ・ 別の定義済みのフィルタ、または @WHERE 関数を含むオブジェクトの WHERE 句を参照する定義済みのフィルタは、参照オブジェクトのクラス必須フィルタを継承しません。

例 OLAP ユニバースの必須フィルタ

次のフィルタ(XML コードで表示)では、プロンプトでユーザが入力したコードが認証されます。

```
<FILTER KEY="[BCOMUSI]">
  <CONDITION OPERATORCONDITION="InList">
    <CONSTANT TECH_NAME=
      "@Prompt('CO_CODE Char User MultiSingle Man Def',
        'A','Company code¥Lov[BCOMUSI]Base',
        multi,primary_key)"/>
    </CONDITION>
  </FILTER>
```

関連項目

- ・ 310 ページの[必須フィルタの例](#)
- ・ 311 ページの[必須フィルタと値の一覧](#)

6.6.16.3.1 必須フィルタの例

ユニバースの必須フィルタの使用方法を次の例に示します。

テーブルに保存されたログインでユーザが入力したログインを確認する。

```
1 = (Select 1 from Club.dbo.Login
where Login = @Variable('BOUSER')
AND Password = @Prompt('Password?', 'A', ,mono,free) )
```

ユニバースの使用時間を午前 9 時～午後 6 時に制限する。

```
1 = (select 1
where datepart(HH,getdate()) between 9 and 18)
```

クラス必須フィルタの例を次に示します。

国/地方/都市/顧客を含むクラスに定義され、クエリを特定期間の販売情報に制限し、期間についてユーザにプロンプトを表示する。

```
Club.dbo.Customer.cust_id in
(Select cust_id from Club.dbo.Sales
where @Select(Sales¥Year) in
@Prompt('Sales Periods?', 'A',
'Sales¥Year',multi,constrained))
```

関連項目

- ・ 309 ページの[必須フィルタ](#)

6.6.16.4 必須フィルタと値の一覧

必須フィルタには、値の一覧を関連付けることができます。値の一覧を関連付けるには、フィルタを適用するオブジェクトの[オブジェクトのプロパティ]ページで値の一覧のオプションを明示的に選択する必要があります。

ユニバース必須フィルタは、カスケード値の一覧に関連付けることができます。

クラス必須フィルタは、対応するクラスの少なくとも 1 つのオブジェクトがカスケード値の一覧に含まれる場合、カスケード値の一覧に関連付けることができます。これは、カスケード値の一覧グループオブジェクトが別のクラスの場合でも同じです。

推奨事項

必須フィルタは、カスケード値の一覧の最上位レベルのみで生成します。

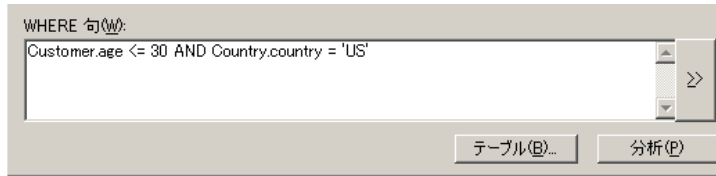
カスケード値の一覧は、プロンプトを含む必須フィルタに関連付けしないでください。Web Intelligence は、カスケード値の一覧のプロンプトをサポートしていません。

6.6.16.5 条件オブジェクトの作成

条件オブジェクトを作成する手順は、次の通りです。

- 1 [ユニバース]枠の下にある[条件]ラジオボタンをクリックします。
[ユニバース]枠の[条件]ビューが表示されます。ユニバースのすべてのクラスがツリー構造で表示されます。
- 2 テーブルを右クリックし、ポップアップメニューから[挿入]を選択します。
または
クラスをクリックし、[条件の挿入]ボタンをクリックします。
[プロパティの編集]ダイアログボックスが表示されます。[名前]ボックスにデフォルト名が表示されています。
[WHERE 句]は空です。
- 3 条件名を入力します。
- 4 [WHERE 句]ボックスに直接 WHERE 句の構文を入力します。
または
[Where 句]の隣にある[>>]ボタンをクリックして Where 句エディタを開きます。
- 5 [SQL 構造]と[機能]の一覧に表示される列、オブジェクト、演算子、関数をダブルクリックします。
- 6 [OK]をクリックしてエディタを閉じます。

次の図は、[USAのヤング]という条件オブジェクトの設定内容です。この条件オブジェクトは、国籍が「US」の 30 歳以下の顧客のデータだけを返す制限を適用するものです。

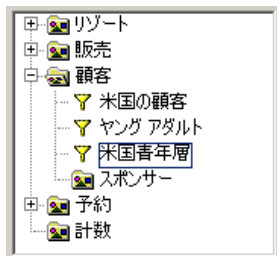


- 7 [分析]をクリックすると、クエリの構文を確認できます。
- 8 フィルタを強制フィルタとして定義するには、[強制フィルタ]チェックボックスを選択します。
デフォルトでは、強制フィルタはクラスに適用され、値の一覧には適用されません。
- 9 ラジオボタンを選択し、強制フィルタがクラスまたはユニバースに適用されるよう定義します。
- 10 強制フィルタを値の一覧に適用するには、[値の一覧に適用]を選択します。
- 11 [OK]をクリックします。

新しい条件オブジェクトが[ユニバース]枠の[条件]ビューに表示されます。

注

事前定義されたフィルタエディタを使用して条件オブジェクトを編集するには、[>>]をクリックします。



関連項目

- ・ [309 ページの必須フィルタ](#)
- ・ [311 ページの必須フィルタと値の一覧](#)
- ・ [461 ページのOLAP ユニバースに含まれる定義済みの条件](#)
- ・ [466 ページのOLAP ユニバースに含まれるオプションプロンプト](#)

6.6.16.6 同じクエリでの複数の条件オブジェクトの使い方

同じオブジェクトに対して 2 つの条件オブジェクトが定義されており、これらを 1 つのクエリで指定すると、両方の WHERE 句の条件を満足するデータがないため、データを取得できない場合があります。ハードコード化さ

れた WHERE 句を回避できるよう、できる限りオブジェクトの WHERE 句は定義しないことが重要ですが、条件オブジェクトを使う場合でも潜在的な問題があることをユーザに認識してもらう必要があります。

それぞれの条件のオブジェクトを使った 2 つのクエリーを結合することで、この問題は解決できます。

注

Web Intelligence ユーザーが、競合する 2 つの条件オブジェクトを 1 つのクエリーに含めることを防止するため、条件オブジェクト“X”の説明に「オブジェクト“Y”と一緒に使用できません。」といったコメントを付けておくことができます。

6.6.16.7 複数の WHERE 句があるとデータを取得できない理由

オブジェクトに WHERE 句を定義すると、その制限はテーブル結合による制限に AND 演算子を使って追加されます。同じデータを制限する 2 つの条件オブジェクトをクエリに含めると、2 つの WHERE 句が AND で接続されます。このようなクエリは、両方の条件を満たすデータがないため、データを返しません。

たとえば、Bahamas Beach と Hawaiian Club で利用できるサービスを取得するため、Bahamas リゾートと Hawaiian リゾートに条件オブジェクトを使用し、次のクエリを実行します。

このクエリの SQL は次の通りです。

```
SELECT Service.service, Resort.resort FROM Service, Resort, Service_Line WHERE ( Resort.resort_id=Service_Line.resort_id ) AND ( Service.sl_id=Service_Line.sl_id ) AND ( ( Resort.resort = 'Bahamas Beach' ) AND ( Resort.resort = 'Hawaiian Club' ) )
```

WHERE 条件の最後の 2 つの WHERE 句が AND 句で結ばれています。

クエリを実行すると、1 つのクエリで 2 つの条件（「Bahamas Beach」かつ「Hawaiian Club」であるリゾート）を満足するデータがないため、取得できるデータはないと判断されます。「フェッチするデータがありません。」というメッセージボックスが表示されます。

2 つのクエリを作成した制限の組み合わせ

この問題は、それぞれの条件オブジェクトを使った 2 つのクエリを実行し、結果をユニオン演算子で結合することで、解決できます。

6.6.17 自己結合を使った制限

自己結合を使い、テーブルの列を示すフラグによって参照する列を切り替えることで、データを制限することができます。この場合、1 列めか 2 列めかを示す値が、フラグとして 3 列めに存在している必要があります。

自己結合の制限の作成と使い方の詳細については、184 ページの [「自己制限結合」](#) を参照してください。

6.6.18 複数のテーブルによる制限の適用

オブジェクトに返されるデータを、別のテーブルの値と一致するオブジェクトで指定されたテーブルの値に制限することができます。

たとえば、“国籍”というオブジェクトは、Country テーブルを参照します。次の値を返します。

出身国
Australia
France
Germany
Holland
Japan
UK
US

このオブジェクトを“販売員”クラスで使用し、販売員が活動する国だけ取得するには、オブジェクト名を“販売員の国”に変更し、Sales_Person テーブルの販売員の国のデータだけを返すよう Country テーブルに制限を加えます。

“販売員の国”オブジェクトの SQL 定義は、次のとおりです。

```
SELECT Country.country FROM Country, Sales_Person, Customer, City, Region WHERE ( City.city_id=Customer.city_id ) AND ( City.region_id=Region.region_id ) AND ( Country.country_id=Region.country_id ) AND ( Sales_Person.sales_id=Customer.sales_id )
```

このオブジェクトは、次のデータを返します。

販売員の国
France
Germany
Japan
UK
US

[販売員の国]オブジェクトがクエリに挿入されたら、SELECT文のFrom句に[Sales_Person]テーブルを加えるよう指定することで、データを制限します。

これで、Sales_Person クラスの下にある Country オブジェクトは、販売員が働いている国のデータのみを返すようになります。オブジェクトの定義シートにある [テーブル] ボタンを使用して、制限を適用します。

Country テーブルは、等価結合のみを使用した仲介結合で Sales_Person テーブルに結合する必要があります。

注

SELECT 文にテーブル制限が記述された SQL を変更した場合、ユニバースデザインツールによって、オブジェクトの SELECT 文と WHERE 句に必要なテーブルが自動的に再設定されます。参照するテーブルが入れ替わって SQL が上書きされても、メッセージによる通知はありませんので注意してください。

6.6.18.1 複数のテーブルによる条件の適用

複数のテーブルを参照してオブジェクトを制限する手順は、次のとおりです。

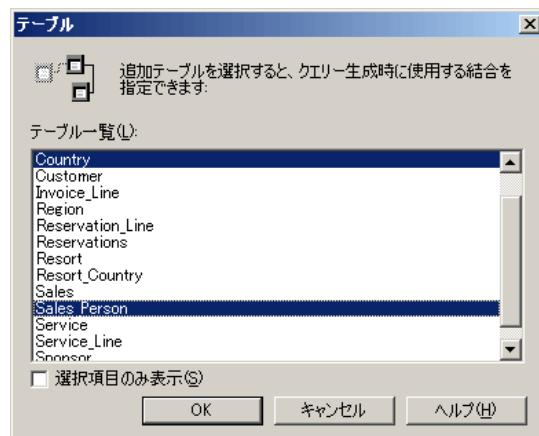
- 1 オブジェクトをダブルクリックします。

オブジェクトの [プロパティを編集] ダイアログボックスが表示されます。

- 2 [テーブル] ボタンをクリックします。

ユニバースのテーブルが一覧表示されます。

- 3 オブジェクトで現在のテーブルに追加するテーブルを選択します (複数可)。[Ctrl] キーを押しながら複数のテーブルを選択することができます。下の図では、[Country] テーブルと [Sales_Person] テーブルを選択しています。



- 4 各ダイアログボックスで [OK] をクリックします。
- 5 Web Intelligence でクエリを実行し、テーブル制限を確認します。

6.6.18.2 データ制限方法の適用基準

ユニバースに制限を設定する場合は、次のガイドラインを参考にしてください。

- ・ オブジェクトの定義では WHERE 句を使用しないでください。どうしても WHERE 句を使用せざるを得ない場合は、類似オブジェクトの問題や WHERE 句の競合の十分注意を払ってください。
- ・ 条件オブジェクトを作成し、ユーザが任意に使用できる定義済みの条件を提供すると、クラスとオブジェクトの構成を変えずに類似オブジェクトの問題を回避できます。

- ・ SQL内でのテーブルの参照場所に関係なく、複数のテーブルに制限を適用する場合は、自己結合を使用してください。この方法は、複数の種類のデータの切り替えにフラグを使っているテーブルに最適です。
- ・ ユニバース内のルックアップテーブルを多目的に使っている場合は、結合を追加してください。

6.6.19 連結オブジェクト

連結オブジェクトは、2つのテーブルの列を組み合わせて作成するオブジェクトです。たとえば、“顧客クラス”に Last Name と First Name 列を連結した “フルネーム” というオブジェクトを作成できます。

6.6.19.1 連結オブジェクトの作成

連結オブジェクトは次の要領で作成します。

- 1 オブジェクトを作成します。

たとえば、“顧客” クラスに新しいオブジェクトを作成し、“フルネーム” という名前を付けます。オブジェクトに「顧客の姓と名前を連結したものです」といった説明を入力するのを忘れないでください。



- 2 オブジェクトをダブルクリックします。

[(オブジェクト名)のプロパティを編集]ダイアログボックスが開きます。

- 3 [SELECT句]ボックスに、連結オブジェクトのSQL定義を入力します。

たとえば MS Access ならば、[フルネーム]には次の構文を入力します。

```
rtrim (Customer.first_name + ' ' + Customer.last_name)
```

rtrim関数は文字列の後ろの空白を削除する関数で、2つの引用符は姓と名前の間にスペースを挿入するために必要です。

定義 | プロパティ | 詳細 | キー

名前(N): フルネーム 型(T): 文字型

説明(D): 顧客の姓と名を連結したものです。

SELECT 句(S): rtrim(Customer.last_name + ' ' + Customer.first_name)

WHERE 句(W):

テーブル(T)... 分析(P)

OK キャンセル 適用(A) ヘルプ(H)

注

[編集]ボタンをクリックしてSQLエディタを開き、エディタのグラフィックツールを使ってオブジェクトのSQL構文を定義することもできます。このエディタに関する詳しい説明は、「スキーマのデザイン」の章を参照してください。

- 4 各ダイアログボックスで[OK]をクリックします。

Full Name オブジェクトに対するクエリを実行すると、ファーストネームのアルファベット順に一覧表示されたフルネームが返されます。

6.7 階層の定義

ユーザが多次元分析を行えるよう、オブジェクトに階層を設定します。

6.7.1 多次元分析とは

多次元分析とは、階層構造に配列されたディメンションを分析することです。

エンドユーザは、多次元分析でさまざまな角度からデータを分析し、データの全体的な傾向や例外を見つけることができます。

階層とは、関連したディメンションを順番に配列したものです。たとえば“地理”階層は、“国”、“地方”、“都市”ディメンションによって構成されます。

Web Intelligence では、ドリルダウンまたはドリルアップ機能を使用して、多次元分析を実行できます。

6.7.1.1 ドリル

ドリルは、階層レベルを上下にナビゲートして分析を行うものです。ユーザは階層を「ドリルアップ」または「ドリルダウン」することができます。

たとえば、数年にわたる予約データを細かく追跡する必要のあるユーザがいるとします。ユニバースデザイナーは、「予約年」、「予約四半期」、「予約月」、「予約日」を含む「予約」階層を準備することができます。

これにより、ユーザは「予約四半期」などの総合レベルから、「予約月」、「予約日」などのより詳しい情報へとドリルダウンできます。また「予約四半期」から「予約年」へと、よりまとまった情報にドリルアップすることもできます。

6.7.2 階層の識別方法

階層は、どんな形式でも構いません。典型的な階層の例には、次のようなものがあります。

- ・ 地理: 大陸 - 国 - 地方 - 都市
- ・ 製品: カテゴリ - ブランド - 製品
- ・ 時間: 年 - 四半期 - 月 - 週 - 日

さらに、次のように階層を組み合わせることもできます。

地理/製品: 大陸 - 国 - カテゴリ - ブランド - 製品

階層を構成するデータは、データベース内での保存方法やデータの種類による制限を受けます。ユーザグループの分析要求に最適な特定のシステムで階層を構成するには、データを細心の注意で分析する必要があります。

データの階層構造を一律に決定する、簡単で確実な規則というものはありませんが、データベース内の「一対多 (1-N)」のリレーションシップは、潜在的な階層を示す手掛かりとなります。

下の図では、テーブル間の「一対多」のリレーションシップが地理的な階層と対応しています。



6.7.3 階層の設定

ユニバースデザインツールでは、デフォルトで、多次元分析のためのデフォルト階層が提供されています。これらは、ユニバース枠に表示される順番にしたがって表示されるクラスとオブジェクトです。オブジェクトを作成するとき、ユーザにとって意味のあるデフォルト階層を作成する必要があります。

別のクラスにあるオブジェクトも含めたユーザ定義の階層を作成する必要がある場合もあります。この場合、新しい階層を作成します。

階層エディタを使用すると、デフォルト階層を表示し、ユーザ定義階層を作成することができます。このグラフィックエディタで、ユニバース階層を管理できます。

注

カスタム階層を定義すると、デフォルト階層はアクティブでなくなり、エンドユーザが使用できなくなります。デフォルト階層をアクティブにする場合は、[階層構造エディタ]でデフォルト階層を明示的に選択し、カスタム階層リストに追加する必要があります。

6.7.3.1 階層の表示

階層を表示するには、次の操作を行います。

6.7.3.1.1 ユニバースの階層を表示する

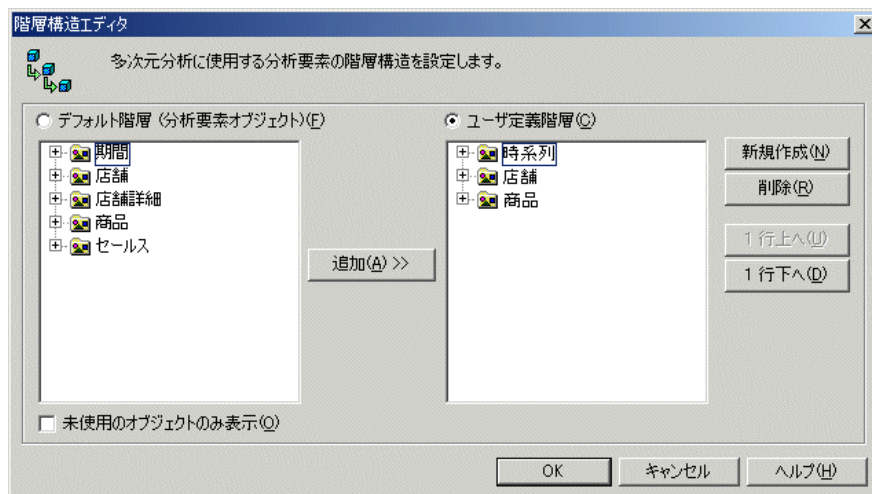
- 1 [ツール]>[階層]の順に選択します。

または

[階層]ボタンをクリックします。

[階層構造エディタ]が開きます。ユニバースデザインツールでは、階層はフォルダアイコンで、ディメンションは立方体アイコンで表されます。

左の枠には、アクティブユニバースのディメンションオブジェクトを含んだデフォルト階層がすべて表示されています。右の枠には、ユーザが作成したカスタム階層がすべて表示されています。



- 2 階層ノード([+]記号)をクリックすると、階層別に整理されたディメンションが表示されます。
- 3 [キャンセル]をクリックします。

6.7.3.2 階層の設定

新しい階層を作成するには、[カスタム階層]枠に新しいフォルダを作成し、そこにディメンションを階層順に追加していきます。

階層またはディメンションを選択して [削除] ボタンをクリックすると、階層またはディメンションが削除されます。

6.7.3.2.1 新しい階層を作成する

- 1 階層構造エディタを開き、[新規作成]をクリックします。

または

階層構造エディタの左枠でクラスを選択し、そのまま右枠までドラッグします。

階層を示すフォルダが右の枠に表示されます。

- 2 階層名を入力します。
 - 3 Enter キーを押して、名前を適用します。
 - 4 作成した階層をクリックします。
- 階層が反転表示されます。
- 5 左枠で、デフォルト階層の[+]記号をクリックします。
- 展開するのは、新しいカスタム階層に追加するディメンションを持った階層です。

- 6 ディメンションをクリックします。複数のディメンションオブジェクトを選択するには、Ctrl キーを押しながら、ディメンションをクリックしていきます。

1 つまたは複数のディメンションが反転表示されます。

- 7 [追加]ボタンをクリックします。

左枠で選択したディメンションオブジェクトが、右枠で選択した階層に挿入されます。

注

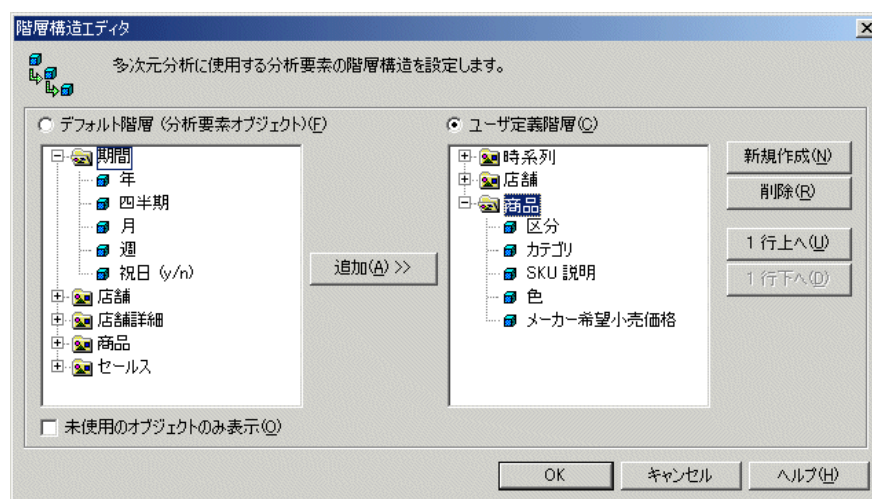
[未使用のオブジェクトのみ表示] オプションを選択すると、まだ階層に含めていないディメンションのみが表示されます。

6.7.3.3 階層とディメンションの再編成

階層内でディメンションを表示する順序を変更することができます。オブジェクトを移動するには、オブジェクトをクリックし、[1 行上へ] または [1 行下へ] ボタンをクリックします。同様に階層の順序も変更することができます。

ディメンションまたは階層をドラッグアンドドロップして移動することもできます。

階層とディメンションオブジェクトの例を、次に示します。



この階層構造エディタでは、[時系列]、[店舗]、[商品]の3つのカスタム階層が設定されています。“商品”階層は“区分”、“カテゴリ”、“SKU 説明”、“色”、“メーカー希望小売価格”のディメンションオブジェクトで構成されています。

6.8 階層のカスケード値の一覧の使用

デフォルトまたはカスタム階層を値の一覧と関連付けることができます。これは、カスケード値の一覧と呼ばれます。

注

“値の一覧(LOV)”とは、オブジェクトに関連付けられたデータ値を記録したリストです。値の一覧についての説明は、324 ページの「[値の一覧の使い方](#)」を参照してください。

カスケード値の一覧は、ユニバース内の階層と関連付けられた値の一覧の配列です。各階層レベルに、その階層の値の一覧を返すプロンプトが定義されます。

カスケード値の一覧に関連付けられた階層を含むレポートを最新表示すると、階層が表示され、レベルの選択を要求するプロンプトが表示されるので、クエリを実行する前に値の一覧から 1 つ以上の値を選択します。

たとえば、“予約”四半期は“年度”階層と関連付けられます。“予約”四半期の月が使用される場合、“年度”階層が表示されます。ユーザは、クエリを実行する前に、その四半期の年度を選択するように求められます。

6.8.1 カスケード値の一覧の作成

カスケード値の一覧は、デフォルト階層またはカスタム階層に対して作成できます。各レベルにつき 1 つの LOV ファイルが作成されます。クエリが実行されると、要求された階層レベルの LOV のみが返されます。

注

このように階層の値の一覧を繰り返し使用することは、値の一覧の階層を作成することとは異なり、階層のすべてのレベルのすべての値の一覧がマイクロキューブに返されます。カスケード値の一覧を使用すると、レベルが入力されるまで LOV はマイクロキューブに返されず、レベルの LOV のみが返されます。

6.8.1.1 カスケード値の一覧を作成する

- 1 [ツール] > [値の一覧] > [カスケード値の一覧を作成]の順に選択します。
[カスケード値の一覧を作成]ダイアログボックスが開きます。



次のオプションが使用できます。

カスケード LOV オプション	説明
デフォルト階層 カスタム階層	どちらかを選択すると、ユニバースに定義された、対応するデフォルトまたはカスタム階層が[利用可能]枠に表示されます。これらの階層の種類の詳細については、319 ページの「 階層の設定 」を参照してください。
階層表示	選択すると、階層は[クエリパネル]にツリー表示されます。これにより、階層間の移動が容易になります。レベルがクリックされると、[クエリパネル]の右の枠に値の一覧が表示されます。
オブジェクト	ディメンションの階層レベル。
プロンプトのテキスト	そのレベルの値の一覧のプロンプトに表示されるテキスト。

- 2 [デフォルト階層]または[カスタム階層]オプションボタンをクリックします。

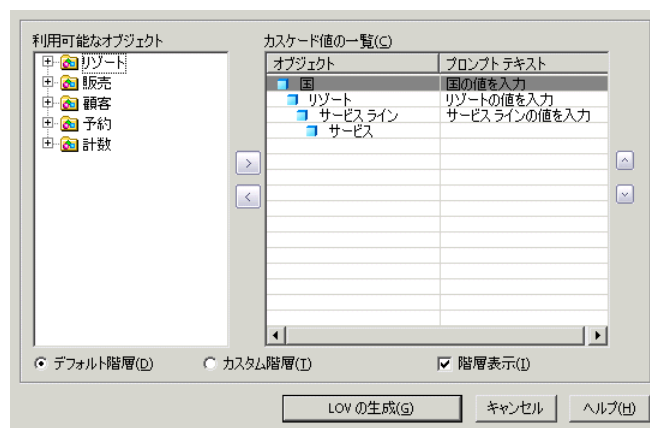
ユニバースで使用できる、対応する階層のリストが表示されます。

- 3 1つのクラスをクリックするか、クラスを展開して、1つ以上のオブジェクトを選択します。
- 4 右向きの矢印をクリックします。

クラスにあるすべてのオブジェクトがオブジェクトの一覧に表示されます。

または

選択したオブジェクトが[オブジェクト]の一覧に表示されます。



- 5 各オブジェクトにプロンプトテキストを入力します。
- 6 [カスケード値の一覧]内でのオブジェクトの位置を変更する場合は、オブジェクトをクリックし、上向きおよび下向きの矢印を使用して一覧の中を上下させます。

オブジェクトを削除する場合は、そのオブジェクトをクリックして左向きの矢印をクリックします。

- 7 [階層表示]チェックボックスをオンまたはオフにします。
- 8 [LOVの生成]をクリックします。

[カスケード値の一覧を作成]ダイアログボックスが閉じられます。カスケード値の一覧の各レベルに対してLOVが作成されます。各.LOVファイルは、ファイルシステム上のユニバースのサブフォルダに保存されます。たとえば、C:\Documents and Settings\ユーザー名\Application Data\Business Objects\Business Objects 12.0\Universes\CMS名\beachXI3.0\サブフォルダに保存されます。

注

オブジェクトごとの値の一覧の編集、CMSへのエクスポート、および作成については、324ページの[「値の一覧の使い方」](#)を参照してください。

6.9 値の一覧の使い方

値の一覧とは、オブジェクトに関連付けられたデータ値を記録したリストです。値の一覧には、次の2種類のデータソースからのデータが含まれています。

データソース	説明
データベース	<p>オブジェクトを作成すると、ユニバースデザインツールにより、自動的にそのオブジェクトに値の一覧が割り当てられます。値の一覧は、ユーザまたはデザイナーがクエリパネルでそのオブジェクトの値の一覧を表示しない限り、作成されません。値の一覧が要求されると、オブジェクトが参照する列にSELECT DISTINCTクエリが実行されます。</p> <p>取得されるデータは、拡張子 .LOV の付いたファイルとして、ユニバースファイルを保存している同じフォルダの下にある、ユニバースのサブフォルダに保存されます。.LOVファイルは、一覧の値のソースとして使用されます。</p>
外部ファイル	<p>値の一覧は、テキストファイルやExcelファイルなどの個人データにも関連付けられます。</p> <p>外部ファイルを参照する値の一覧では、値が固定されます。外部ファイルへの動的リンクは設定できません。外部ファイルが変更された場合は、.LOVファイルを最新表示する必要があります。</p>

6.9.1 値の一覧の使い方

Web Intelligence では、ユーザが [クエリ] ペインでクエリを作成する際、[値の一覧を表示] オペランドを使って条件を設定できます。

注

オブジェクトが参照する列の値を制限するような条件が[クエリ]枠のオブジェクトに適用された場合も .LOV ファイルが作成されます。

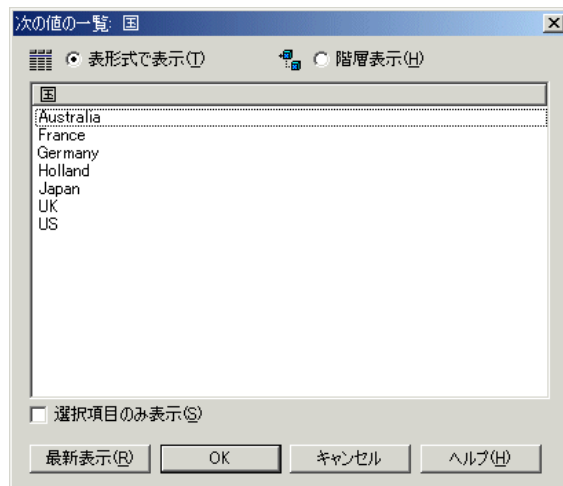
オブジェクトの値の一覧には、そのオブジェクトで利用できる値が表示され、ユーザはそこから条件とする値を選択します。値の一覧を初めて使用した場合は、データが .LOV ファイルとしてファイルシステムのユニバースのサブフォルダに保存されます。従って、オブジェクトに対するSELECT DISTINCTクエリは一度しか実行されません。

このフォルダには、ユニバース作成者がオブジェクトのデータへのアクセスを制限するために使用した.LOVファイルも保存されています。ユニバースデザインツールから作成するこれらの.LOVファイルは、値の一覧を特定の値に制限するために使用されます。

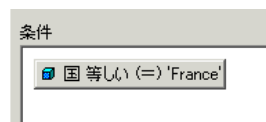
例 [国]の値の一覧を使う

“国”オブジェクトには、{COUNTRY.COUNTRY_NAME} の SELECT 句が定義されています。デフォルトでオブジェクトに関連付けられた値の一覧には、[COUNTRY_NAME]列のすべての値が含まれています。[国]オブジェクトをクエリの条件にした場合は、この値の一覧が表示されます。

ユーザがクエリで取得するデータを France だけに限定するには、Country テーブルにあるすべての国を表示する値の一覧から“France”を選択し、これを条件とすることができます。



一覧から[France]が選択されると、[クエリ]枠の[条件]枠に次の条件が表示されます。



これで、クエリはフランスに対するデータだけを返すようになります。

6.9.2 オブジェクトに関連付けられた値の一覧

ユニバースデザインツールでディメンションオブジェクトや詳細オブジェクトを作成すると、自動的に値の一覧が割り当てられます。ただし、オブジェクトを作成した時点ですぐに一覧が作成されるわけではありません。クエリパネルでオブジェクトが使われたときに、データベースにクエリを実行して値の一覧を取得できるという設定をデフォルトでオブジェクトに定義するだけです。

注

メジャーオブジェクトには、値の一覧は割り当てられません。

ユニバースデザインツールのクエリ枠のオブジェクトに条件を設定すると、値の一覧が初めて要求された時点で、SELECT DISTINCT 文がオブジェクトが参照する列に実行され、値の一覧が返されます。

.LOV ファイルが自動的に作成され、ユニバースのサブフォルダに保存されます。以降は、ユニバースデザインツールで同じ値の一覧が要求されると、データベースからではなく、.LOV ファイルから値が取得されます。

6.9.2.1 値の一覧の管理に関するデザイナの役割

ユニバースデザイナは、一覧に表示するデータと一覧用に取得するデータの種類と量を定義します。

オブジェクトには、値の一覧に関する次のプロパティを設定できます。

- ・ オブジェクトに値の一覧を関連付けるかどうか。
- ・ 一覧を最新表示するタイミング。
- ・ 値の一覧の取得に使用するSELECT DISTINCTクエリに設定する条件を定義する。条件を設定したクエリは、オブジェクトのプロパティとして保存されます。
- ・ 値の一覧を単純な一覧として表示するか、階層表示するか。
- ・ 値の一覧の取得元を列の値にするか、Excelなどの外部ファイルにするか。

値を固定した値の一覧を作成し、リポジトリにエクスポートすることもできます。この.LOVファイルは最新表示されず、常にそのオブジェクトの値の一覧として使われます。更新は行われません。

6.9.3 値の一覧のプロパティとオプション

次のオブジェクトプロパティを設定して、関連付けられた値の一覧を Web Intelligence でどのように使用するかを制御できます。

プロパティ	説明
値の一覧の関連付け	<ul style="list-style-type: none"> ・ このオプションを選択すると、オブジェクトに値の一覧を関連付けることができます。デフォルトで選択されています ・ 選択しないと、オブジェクトと値の一覧は関連付けられません。 ・ ディメンションオブジェクトと詳細オブジェクトではデフォルトで選択されますが、メジャーオブジェクトでは選択されません。
一覧名	取得した一覧データを保存する .LOV ファイルの名前です。8 文字までの英数字で指定します。

プロパティ	説明
ユーザによる値の一覧の編集	<ul style="list-style-type: none"> このオプションを選択すると、ユーザは Web Intelligence で LOV ファイルを編集できるようになります。 選択しないと、ユーザは一覧を編集できません。 <p>注 これは、Excel スプレッドシートなどの個人用データファイルには当てはまりません。これらはリポジトリにエクスポートされず、そのままローカルマシンに残ります。ユーザはローカルファイルを編集するか、他のローカルデータファイルで対象となる値の一覧を変更できます。</p> <p>値の一覧の目的は、ユーザが使用できる値を制限することです。編集を許可すると、ユーザが選択できる値を管理できなくなります。通常、個人用データファイルを値の一覧のソースとして使用していない場合、このオプションをオフにして、ユーザによって値の一覧が編集されないようにします。</p>
使用前に自動更新(BusinessObjects のみ)	<ul style="list-style-type: none"> このオプションを選択すると、クエリパネルにオブジェクトの値の一覧を表示する際に、必ず一覧の値が更新されます。毎回、.LOV ファイルが最新表示されるので、パフォーマンスに影響を及ぼします。このオプションは、Web Intelligence レポートには適用されません。 選択しないと、ユーザのログイン時に一度だけ一覧が更新されます。 <p>一覧に含まれる値の更新頻度が高い場合は、このオプションを選択する必要がありますが、パフォーマンスへの影響も十分に考慮してください。</p> <p>一覧の内容が一定の場合は、このオプションを選択しないでください。</p>
階層表示	Web Intelligence でカスケード値の一覧を階層で表示するには、[階層表示] プロパティを選択します。
ユニバースと一緒にエクスポート	<ul style="list-style-type: none"> このオプションを選択すると、オブジェクトに関連付けられた .LOV ファイルがユニバースと一緒にリポジトリにエクスポートされます。 エクスポートするには、あらかじめオブジェクトに関連付けられた値の一覧を作成しておく必要があります。値の一覧は、.LOV ファイルとして保存されます。 選択しないと、オブジェクトの .LOV ファイルはリポジトリにエクスポートされません。 <p>一覧を定期的カスタマイズする場合は、このオプションを選択してください。変更内容がユニバースと共にエクスポート、インポートされます。</p>

プロパティ	説明						
依頼検索	Web Intelligence ユーザが値の一覧に返される値の数を制限するには、[依頼検索] プロパティを選択します。[依頼検索] プロパティを選択すると、Web Intelligence でクエリの実行時に、値の一覧ボックスが空の状態が表示されます。ユーザは値を入力して、値の一覧をフィルタ処理する際の検索条件を定義します。						
	多くのデータソースではワイルドカード文字がサポートされており、データベースを容易に検索できます。Web Intelligence では、次のワイルドカードをサポートしています。						
	<table><tr><td>*</td><td>ゼロ文字を含む任意の数の文字数に一致</td></tr><tr><td>?</td><td>1 つの文字に一致</td></tr><tr><td>¥</td><td>次の文字をエスケープして、ワイルドカード文字を検索</td></tr></table>	*	ゼロ文字を含む任意の数の文字数に一致	?	1 つの文字に一致	¥	次の文字をエスケープして、ワイルドカード文字を検索
	*	ゼロ文字を含む任意の数の文字数に一致					
	?	1 つの文字に一致					
¥	次の文字をエスケープして、ワイルドカード文字を検索						
依頼検索オプションには、次の制限があります。							
<ul style="list-style-type: none">・ カスケード値の一覧では、依頼検索はサポートされていません。・ 依頼検索は、タイプ文字のオブジェクトの値の一覧のみで実行できます。・ 値の一覧に対してユーザ定義の SQL が入力されると、依頼検索は実行できません。・ 値の一覧に対して[階層表示]オプションが使用されている場合、依頼検索は実行できません。							

次のボタンを使って、値の一覧の編集、表示、デフォルト名の復元を行うことができます。

オプション	説明
デフォルト値を復元	オブジェクトの作成時に割り当てられた .LOV ファイルのデフォルト名を復元します。
編集	一覧に表示する値を編集します。エディタを使用して、クエリパネルで表示される値を制限できます。

オプション	説明
表示	オブジェクトの値の一覧を表示します。値の一覧を作成し、ユニバースとともにリポジトリにエクスポートする場合は、[表示]ボタンをクリックして .LOV ファイルを作成する必要があります。作成されたファイルを編集し、値を固定することができます。

6.9.3.1 値の一覧のプロパティとオプションの設定

値の一覧(.LOV)ファイルのプロパティとオプションを設定する

- 1 オブジェクトをダブルクリックします。
[プロパティの編集]ダイアログボックスの[定義]ページが表示されます。
- 2 [プロパティ]タブをクリックします。
[プロパティ]ページが表示されます。
- 3 ページの下部にある[値の一覧]グループボックスから、必要なオプションを選択します。
- 4 [一覧名]ボックスに、関連付ける .LOV ファイル名を入力します。
- 5 一覧の値を制限する場合は、[編集]ボタンをクリックします。
- 6 クエリパネルを使って、一覧の値を取得するクエリを作成します。
- 7 値の一覧を表示するには、[表示]ボタンをクリックします。

このボタンをクリックすると、オブジェクトが参照するデータベースの列に SELECT DISTINCT クエリが実行されます。このクエリは、レポートینگ製品でオブジェクトの .LOV ファイルを作成する場合にも使用されます。

- 8 [OK]をクリックします。

6.9.3.2 オブジェクトに関連付けられた値の一覧の表示

ユニバースデザインツールで、オブジェクトに関連付けられた値の一覧を表示することができます。値の一覧を表示すると、取得したデータを使ってデフォルトの .LOV ファイルが作成され、Business Objects パスの UserDocs ディレクトリに保存されます。デフォルトでは、値の一覧を表示すると自動的に .LOV ファイルが作成されます。

値の一覧の表示形式には、表形式と階層表示の2種類があります。

値の一覧を表示する手順は、次の通りです。

- 1 オブジェクトをダブルクリックします。

[プロパティの編集]ダイアログボックスの[定義]ページが表示されます。

- 2 [プロパティ]タブをクリックします。

[プロパティ]ページが表示されます。

- 3 [表示]ボタンをクリックします。

[値の一覧(オブジェクト名)]ダイアログボックスに、オブジェクトに関連付けられたすべての値が表示されます。



- 4 [キャンセル]をクリックします。

6.9.3.3 値の一覧の作成

.LOV ファイルを作成するには、次の操作を行います。

- 1 オブジェクトの値の一覧を表示します。
- 2 [OK]をクリックします。

ユニバースデザインツールでは値の一覧(.LOV)ファイルは、ユニバースファイルが保存されたフォルダの下にユニバースサブフォルダに保存されます。サブフォルダの名前は.LOVファイルの作成に使用したオブジェクトのあるユニバースのファイル名と同じです。

.LOVファイルが作成されると、一覧を編集して表示されるデータを制限したり、データの表示形式を変更したりすることができます。

6.9.4 値の一覧の編集

値の一覧の編集には、次の2種類があります。

- ・ 一覧を生成するSELECT DISTINCTクエリに条件を適用する。たとえば“リゾート”オブジェクトの値の一覧に表示されるリゾートを、特定の予約客数を上回るリゾートだけに限定する。
- ・ ユーザが一覧から値を選択しやすくなるよう、階層を作成する。これは一覧に含まれる値が多い場合に、非常に効果的な方法です。

6.9.4.1 値の一覧に条件を適用

値の一覧に条件を適用する手順は、次の通りです。

- 1 オブジェクトをダブルクリックします。

[(オブジェクト名)のプロパティを編集]ダイアログボックスが開きます。

- 2 [プロパティ]タブをクリックします。

[プロパティ]ページが表示されます。

- 3 [値の一覧の関連付け]チェックボックスをオンにします。

- 4 一覧の名前を変更する場合は、[一覧名]ボックスに .LOV ファイル名を入力します。

☒ 値の一覧の関連付け(L)
 一覧名(N):

☐ ユーザーによる値の一覧の編集(U)
☐ 使用前に自動更新(B)
☐ 階層表示
☒ ユニバースと一緒にエクスポート(X)

- 5 [編集]ボタンをクリックします。

クエリパネルが開きます。編集中のオブジェクトが、[結果オブジェクト]枠に表示されます。

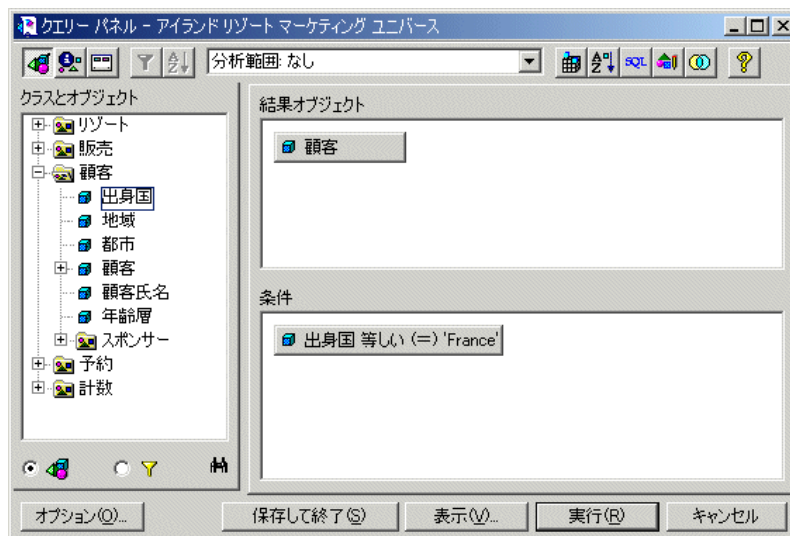
- 6 値の一覧に適用する条件に使用するオブジェクトを[条件]枠までドラッグします。

- 7 [演算子]枠から、使用する演算子をダブルクリックします。

- 8 [オペランド]枠から、使用するオペランドをダブルクリックします。

- 9 必要に応じて、値を選択あるいは入力します。

たとえば、次のクエリではFranceの顧客だけが取得されます。



- 10 [OK]をクリックします。
- 11 [表示]ボタンをクリックします。
空の一覧が表示されます。
- 12 [最新表示]をクリックします。
- 13 値の一覧が表示されます。



- 14 各ダイアログボックスで[OK]をクリックします。

6.9.4.2 値の一覧への階層の作成

値の一覧に階層を作成する手順は、次のとおりです。

- 1 オブジェクトをダブルクリックします。

[(オブジェクト名)のプロパティを編集]ダイアログボックスが開きます。

- 2 [プロパティ]タブをクリックします。

[プロパティ]ページが表示されます。

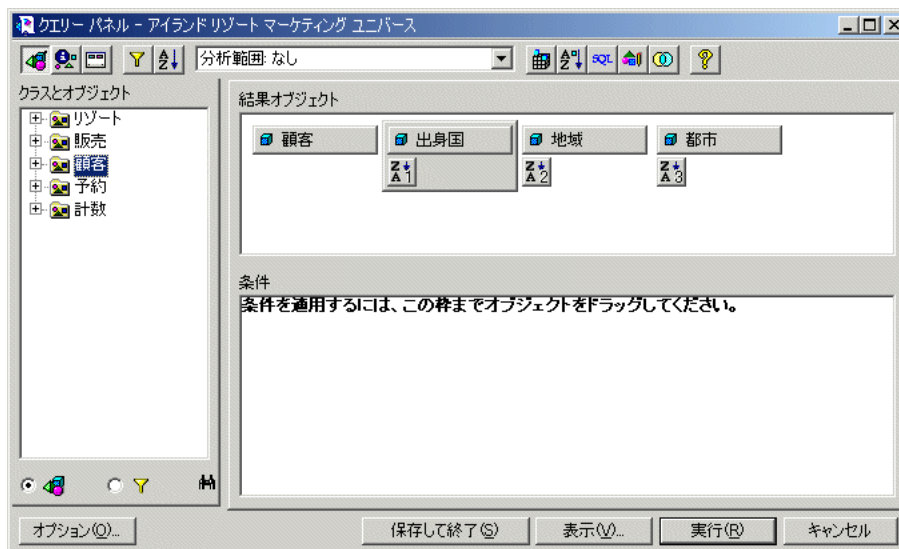
- 3 [値の一覧の関連付け]チェックボックスをオンにします。

- 4 一覧の名前を変更する場合は、[一覧名]ボックスに .LOV ファイル名を入力します。

- 5 [編集]ボタンをクリックします。

クエリパネルが開きます。編集中のオブジェクトが、[結果オブジェクト]枠に表示されます。

- 6 次の図のように、階層を構成するオブジェクトを[結果オブジェクト]にあるメインオブジェクトの右側にドラッグします。



- 7 [OK]をクリックします。

- 8 [表示]ボタンをクリックします。

空の一覧が表示されます。

- 9 [最新表示]をクリックします。

値の一覧が表示されます。



10 各ダイアログボックスで[OK]をクリックします。

6.9.5 値の一覧のエクスポート

値の一覧は、ユニバースと一緒に CMS にエクスポートすることができます。ファイル システム上で関連付けられた .LOV ファイルは、ユニバース ファイルが保存されたのと同じフォルダにあるユニバース サブ ディレクトリに複写されます。

6.9.5.1 Web Intelligence でのエクスポートされた .LOV ファイルの使い方

Web Intelligence で、ユニバースデザインツールからエクスポートされた .LOV ファイルに関連付けられているオブジェクトを使ってクエリが実行されると、次のどちらかの方法で決定された値の一覧が返されます。

- ・ .LOVファイルに含まれるデータ
- ・ .LOVファイルに設定されたSELECT DISTINCTクエリ

ユニバースデザインツールで条件を作成してオブジェクトが返す値を制限した場合、すべての値を持った値の一覧ではなく、制限された値の一覧が返されます。値の一覧には、ユニバースデザインツールで設定されたすべての条件と書式設定が反映されます。

.LOVファイルをユニバースと一緒にエクスポートしなかった場合は、条件や書式設定を考慮しない、デフォルトの値の一覧が返されます。この値を使って、デフォルトの.LOVファイルが作成されます。

6.9.5.2 一覧のエクスポートにデータを含めるか、含めないか

値の一覧を Central Management Server(CMS)リポジトリへエクスポートする方法は、2 つあります。

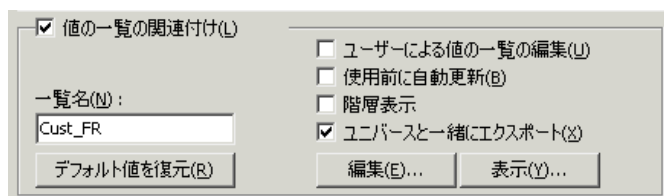
エクスポートする内容	説明
クエリ定義のみ(データなし)	一覧の値を返すSELECT DISTINCTクエリの定義と一緒に.LOVファイルをエクスポートします。これには、ユニバースデザインツールのクエリ枠を使って .LOV ファイルに設定したすべての条件が含まれます。エクスポートされる .LOV ファイルにはデータがないため、クエリパネルで初めてオブジェクトを使用してデータを取得した際に、自動的に値が取り込まれます。頻繁に更新されるデータやデータ量の多い値の一覧には、この方法を使用してください。
データあり	ユニバースデザインツールで値の一覧を表示、編集したときに返されたすべてのデータを含む .LOV ファイルがエクスポート、またはインポートされます。 .LOV ファイルのデータが一定している場合には、この方法を使用してください。頻繁に更新されるデータやデータ量の多い値の一覧をデータと一緒にエクスポートすると、エクスポート処理が遅くなりますので、この方法は選択しないでください。

値の一覧の定義のエクスポート

値の一覧の定義だけ(データなし)をエクスポートする手順は、次のとおりです。

- 1 オブジェクトの値の一覧を作成します。
- 2 [(オブジェクト名)のプロパティを編集]ダイアログボックスの[プロパティ]ページにある[ユニバースと一緒にエクスポート]チェックボックスをオンにします。

次の図には、フランス人顧客だけを返す .LOV ファイル「Cust_FR」が設定されています。



- 3 [ツール]>[値の一覧]の順に選択します。

[値の一覧]ダイアログボックスが開きます。このダイアログボックスには、ユニバースにあるクラスとオブジェクトの一覧とそれぞれのオブジェクトの値の一覧を管理するオプションが表示されます。

- 4 クラスを展開し、リポジトリにエクスポートする.LOVファイルを持ったオブジェクトを選択します。



- 5 [削除]ボタンをクリックします。

オブジェクトの.LOVファイルからデータが削除されます。これで.LOVファイルには、値の一覧のクエリ定義だけが保存されたことになります。

- 6 [OK]をクリックします。

- 7 [ファイル]>[エクスポート]の順に選択します。

[ユニバースのエクスポート]ダイアログボックスが開きます。

- 8 [ユニバース]の一覧から、エクスポートするユニバースを選択します。

- 9 [OK]をクリックします。

ユニバースのエクスポートが正常に終了したことを知らせるメッセージボックスが表示されます。

値の一覧とデータのエクスポート

値の一覧をデータと共にエクスポートする手順は、次の通りです。

- 1 オブジェクトの値の一覧を作成します。

- 2 [(オブジェクト名)のプロパティを編集]ダイアログボックスの[プロパティ]ページにある[ユニバースと一緒にエクスポート]チェックボックスをオンにします。
- 3 [表示]ボタンをクリックします。
値の一覧が表示されます。
- 4 一覧が空白の場合は、[最新表示]ボタンをクリックして、一覧にデータを取り込みます。
- 5 各ダイアログボックスで[OK]をクリックします。
- 6 [ファイル]>[エクスポート]の順に選択します。
[ユニバースのエクスポート]ダイアログボックスが開きます。
- 7 [ユニバース]の一覧から、エクスポートするユニバースを選択します。
- 8 [OK]をクリックします。
ユニバースのエクスポートが正常に終了したことを知らせるメッセージボックスが表示されます。

6.9.6 値の一覧の最新表示

ユニバースデザインツールで値の一覧のデータを最新表示する方法は、2 種類あります。

- ・ オブジェクトの値の一覧を表示し、[最新表示]ボタンをクリックする。
- ・ [ツール]>[値の一覧]の順に選択し、[値の一覧]ダイアログボックスでオブジェクトを選択して[最新表示]ボタンをクリックする。

6.9.7 個人用データファイルからのデータの使用

データベースサーバから取得した会社用データの代わりに、個人用データから作成した値の一覧をオブジェクトに割り当てることができます。

個人用データとは、Microsoft Excel、Lotus 1-2-3、dBASEなどのデータやテキストファイルなどのフラットファイルに保存されたデータを指します。

個人用データの利用には、次の利点があります。

- ・ 個人用データファイルからのデータの読み取りは、会社用データベースにアクセスしてデータを取得するよりも短時間で処理できる。
- ・ 必要な値がデータベースから取得できない場合でも、それを指定できる。
- ・ ユーザに表示される一覧の値を管理できる。

個人用データファイルを使うことの欠点は、データが固定されるということです。値を変更する場合は、手動でデータを更新する必要があります。

6.9.7.1 個人用データファイルからの値の一覧の作成

個人用データファイルから値の一覧を作成する手順は、次の通りです。

- 1 [ツール]>[値の一覧]の順に選択します。

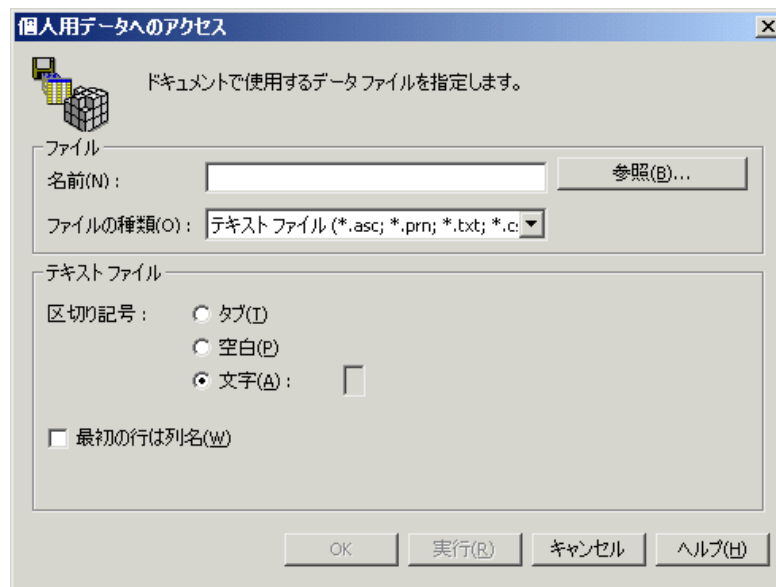
[値の一覧]ダイアログボックスが開きます。

- 2 クラスを展開し、個人用データを割り当てるオブジェクトをクリックします。
- 3 [プロパティ]グループボックスの[個人用データ]オプションボタンをクリックします。

値の一覧の種類を変更することを確認するメッセージボックスが表示されます。

- 4 [OK]をクリックします。

[個人用データへのアクセス]ダイアログボックスが開きます。表示されるオプションは、選択したファイルの種類によって異なります。



- 5 [参照]をクリックして LOV として使用するファイルを指定します。

または

[名前]ボックスにファイル名を直接入力します。

- 6 [ファイルの種類]リストボックスでファイル形式を指定します。
- 7 選択できるファイル形式は、次の通りです。

- ・ テキストファイル (*.asc;*.prn;*.txt;*.csv)
- ・ Microsoft Excel ファイル (*.xls)
- ・ dBASE ファイル (*.dbf)
- ・ Microsoft Excel 97 ファイル (*.xls)

注

Excel 97 で作成したファイルを使用する場合、[Microsoft Excel ファイル (*.xls)]ではなく、[Microsoft Excel 97 ファイル (*.xls)]を選択してください。

- 8 必要に応じて、オプションを指定します。

テキストファイルでは、1 行が 1 つのデータ行に対応します。テキストファイルでは、列の区切り文字 (タブ、空白、文字) を指定します。[文字]を選択した場合は、テキストボックスに区切り文字として使用する文字を入力します。

- 9 [OK]をクリックします。

6.9.8 ユニバース内の値の一覧の管理

[ツール]>[値の一覧]の順に選択して表示される[値の一覧]ダイアログボックスを使い、ユニバース内のすべての値の一覧を管理できます。ダイアログボックスの中央の枠には、作業中のユニバースのすべてのクラスとオブジェクトがツリー表示され、すべてのオブジェクトの値の一覧にアクセスできます。このダイアログボックスには、次のボタンがあります。

オプション	説明
編集	選択したオブジェクトにクエリを定義するために使用するクエリパネルを表示します。既存のクエリを編集、再定義できます。
表示	選択したオブジェクトの現在の値の一覧を表示します。
削除	選択したオブジェクトの値の一覧から値を消去します。
最新表示	値の一覧を最新表示します。

6.9.8.1 値の一覧の管理ツールへのアクセス

値の一覧の管理ツールにアクセスする手順は、次の通りです。

- 1 [ツール]>[値の一覧]>[値の一覧の編集]の順に選択します。

[値の一覧]ダイアログボックスが開きます。



- 2 クラスを展開し、オブジェクトを選択します。
- 3 必要に応じ、ボタンをクリックするか、またはオプションを選択します。
- 4 [OK]をクリックします。

6.9.9 .LOV ファイルの最適化とカスタマイズ

値の一覧の最適化とカスタマイズの一般的な方法は、次の通りです。

方法	説明
一覧の値の取得先のテーブルを変更する。	デフォルトでは、LOVは、それが関連付けられたオブジェクトと同じ列を参照します。この列の行数が多い場合には、.LOVの最新表示にも時間がかかります。同じ値を返すテーブルが他にもあり、その行数が少ない場合には、そのテーブルからデータを取得するよう、LOVを編集します。
コード番号と説明を組み合わせる。	典型的な、LOVのカスタマイズは、コード番号と説明を組み合わせることです。ユーザにとってあまり意味を持たない「製品コード」を返すオブジェクトがある場合、このオブジェクトの .LOV を編集し、「製品説明」と組み合わせて表示することで、わかりやすくなります。逆に、「製品説明」に「製品コード」を組み合わせる表示することもできます。

6.10 ユニバースのリンク

1つまたは複数のユニバースを動的にリンクすることができます。

6.10.1 リンクされたユニバースとは

リンクユニバースとは、パラメータ、クラス、オブジェクト、結合を共通コンポーネントとして共有するユニバースです。

2つのユニバースをリンクするとき、片方がコアユニバースの役割を果たし、もう一方が派生ユニバースとなります。コアユニバースが編集されると、その変更が派生ユニバースにも自動的に反映されます。

注

リンクユニバースの展開についての詳細は、351 ページの [「派生ユニバースと値の一覧」](#) を参照してください。

6.10.1.1 コアユニバースとは

コアユニバースとは、別のユニバースがリンクしているユニバースです。コアユニバースには、それにリンクしている別のユニバースと共通のコンポーネントが含まれています。リンクしているユニバースを派生ユニバースと呼びます。コアユニバースは、再利用可能な部品ライブラリと考えることができます。

コアユニバースは、派生ユニバースでのコアユニバースのコンポーネントの使用方法により、カーネルまたはマスタユニバースと呼ばれます。カーネルおよびマスタユニバースの詳細については、347 ページの「[2 つのユニバース間のリンクの作成](#)」を参照してください。

6.10.1.2 派生ユニバースとは

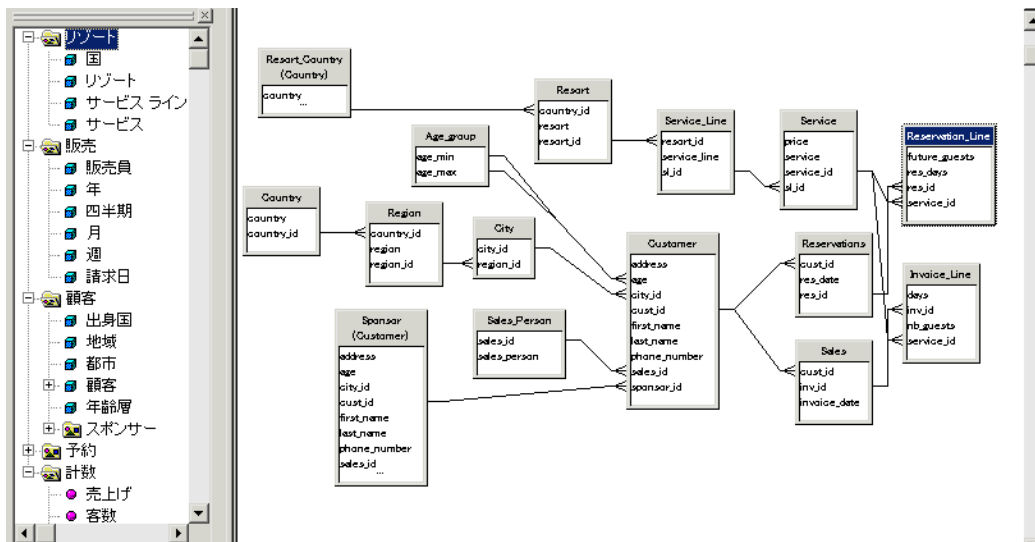
派生ユニバースとは、コアユニバースにリンクされているユニバースです。リンクされていることにより、派生ユニバースはコアユニバースと共通のコンポーネントを共有しています。

- ・ リンクされているコアユニバースがカーネルユニバースの場合、コンポーネントを派生ユニバースに追加することができます。
- ・ リンクされているコアユニバースがマスタユニバースの場合、派生ユニバースにはコアユニバースのすべてのコンポーネントが含まれています。クラスとオブジェクトは、派生ユニバースに追加できません。これらは、派生ユニバースを使用するユーザのニーズに合わせて、非表示になっていることもあります。

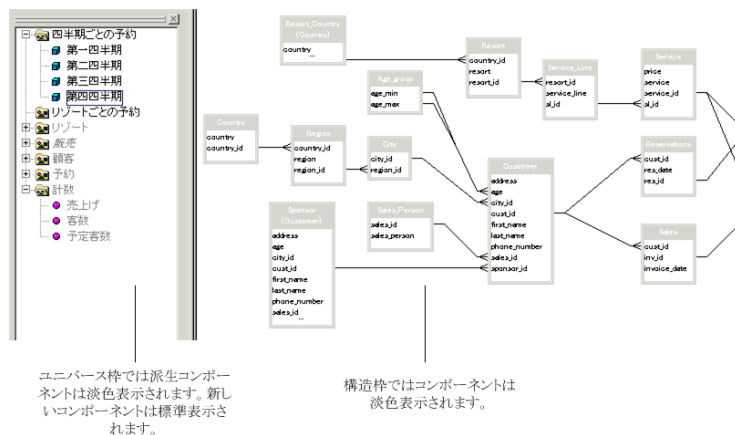
例 リンクされたコアユニバースと派生ユニバース

この例では、リンクされた 2 つのユニバースが表示されています。1 つは共通コンポーネントを持ったコアユニバース、もう一方はコアユニバースの構造を使用するが、さらに独自の新しいクラスとオブジェクトも追加された派生ユニバースです。

コアユニバースは Beach.unv です。アイランドリゾートのセールスマネージャがマーケティングを分析するために使用しています。このユニバースは、このリリースで提供されるデモユニバースの 1 つです。このユニバースの内容を下に示します。



このコアユニバースから、[予約]に焦点を当てた派生ユニバースを作成します。



コアユニバースからの派生コンポーネントはグレイ表示されます。マネージャは2つの新しいクラス、四半期別の予約とレポート別の予約を作成します。これらのクラスとオブジェクトは通常どおり表示されます。[予約]ユニバースには必要のない[販売]クラスを非表示にします。コアユニバースに加えられた変更は、派生ユニバースに自動的に反映されます。

6.10.2 ユニバースをリンクする別の方法

ユニバースをリンクするには、次のいずれかのアプローチを使用します。

- ・ カーネルとして使用
- ・ マスタとして使用
- ・ コンポーネントとして使用

これらの3つの方法を個別に、または組み合わせて使用することができます。

6.10.2.1 カーネルとして使用

カーネルを使用する場合、ユニバースにはコアコンポーネントが含まれます。コアコンポーネントとは、すべてのユニバースに共通のコンポーネントです。このカーネルユニバースから作成したリンクユニバースには、コアコンポーネントとそのユニバース独自の追加コンポーネントが含まれます。

カーネルユニバースへの変更は、そのユニバースにリンクしたすべてのユニバースに自動的に反映されます。

6.10.2.2 マスタとして使用

マスタの使用は、リンクユニバースの共通コンポーネントを自動化するもう 1 つの方法です。

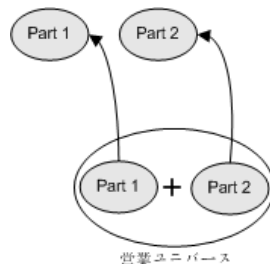
マスタユニバースには、すべてのコンポーネントが含まれています。マスタユニバースからの派生ユニバースでは、派生ユニバースを使用するユーザのプロファイルにしたがって非表示に設定されているコンポーネントもあります。

派生ユニバースで表示されるコンポーネントは、マスタユニバースのサブセットです。派生ユニバースに新しいコンポーネントを独自に追加することはできません。下の例では、マスタユニバースから派生した人事と営業のユニバースが示されています。それぞれのユニバースではマスタユニバースのコンポーネントの一部が隠されています。

マスタユニバースへの変更は、そのマスタユニバースから派生したすべてのユニバースに自動的に反映されます。

6.10.2.3 コンポーネントとして使用

この使用方法では、複数のユニバースを 1 つのユニバースに統合します。営業ユニバースは、第 1 部および第 2 部の 2 つのユニバースを統合して作成されています。



6.10.3 ユニバースをリンクさせる利点

リンクされたユニバースを使用する利点は、次の通りです。

- ・ ダイナミックリンクによりユニバースの開発時間とメンテナンス時間を短縮できます。コアユニバース内のコンポーネントを変更すると、ユニバースデザインツールによって、すべての派生ユニバースの同じコンポーネントに変更が反映されます。

- ・ 新規ユニバースの作成でも、共通コンポーネントを繰り返し定義する必要がなくなり、コアユニバースで一元管理することができます。
- ・ ユニバースをリンクすることでカスタマイズがより簡単になります。基本コアユニバースを設定するデータベース管理者と、具体的な業務分野に基づいてより詳細なユニバースを作成するデザイナー間で開発工程を分業することができます。

6.10.4 ユニバースのリンクの必要条件

次の必要条件を満たす場合には、作業中のユニバースをコアユニバースにリンクできます。

- ・ コアユニバースと派生ユニバースが同じデータアカウントまたはデータベース、およびRDBMSから作成されている。コアユニバースと派生ユニバースの両方に同じ接続を使用することにより、ユニバースの管理が簡単になります。これは、いつでも変更できます。
- ・ コアユニバースと派生ユニバースが同じリポジトリにある。
- ・ コアユニバースが、少なくとも一度はエクスポートされ、再インポートされている。派生ユニバースは、リンクを作成する前にエクスポートされている必要はありません。
- ・ エクスポートされた派生ユニバースは、コアユニバースと同じユニバースドメインにある。
- ・ ユニバースをリンクする権限がある。

6.10.5 ユニバースをリンクするときの制限

ユニバースをリンクするときは、次の制限事項に留意してください。

- ・ ストアドプロシージャを使用したユニバースにリンクできない。
- ・ ユニバースのリンクは 1 レベルだけです。つまり、派生ユニバースから、さらに派生ユニバースを作成することはできません。
- ・ すべてのクラスおよびオブジェクトが、カーネルユニバースおよびリンクされたユニバースで一意的に存在します。そうでない場合、競合が発生します。
- ・ 2 つのユニバースの構造は、両方のユニバースのテーブル間で結合が作成できる構造である必要があります。そうでない場合、両方のユニバースからのオブジェクトを使用してクエリを実行するときにデカルト積が作成されます。
- ・ 派生ユニバースでは、コアユニバースのテーブルスキーマ、クラス、オブジェクトのみが使用できます。コンテキストは、派生ユニバースで再検出する必要があります。
- ・ 派生ユニバースをコアユニバースと同じ構造でエクスポートするときには、コアユニバースに関連した値の一覧は保存されません。

6.10.6 2つのユニバース間のリンクの作成

アクティブユニバースを別のユニバースにリンクすることができます。その場合、作業中のユニバースが派生ユニバースになり、リンクされたユニバースがコアユニバースになります。コアユニバースからのコンポーネントは、派生ユニバースに継承されます。

ユニバースをコアユニバースにリンクするには、コアユニバースがリポジトリにエクスポートされている必要があります。

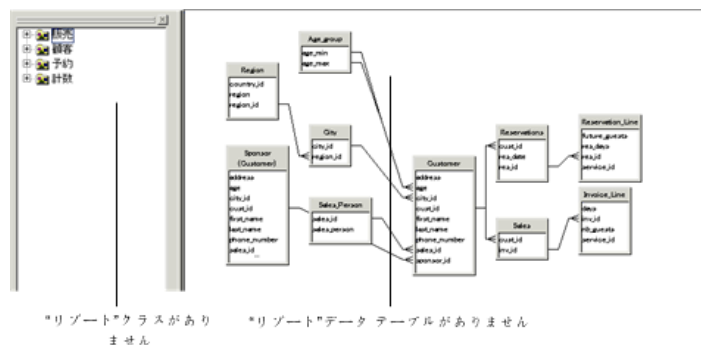
注

ユニバースをリンクする場合、リンクを壊さないようにコアユニバースを同じリポジトリ内に移動させます。これで、派生ユニバースへのリンクが維持されるため、コアユニバースを別のリポジトリフォルダにエクスポートできます。

6.10.6.1 派生ユニバースとコアユニバース間のリンクを作成する手順は、次のとおりです。

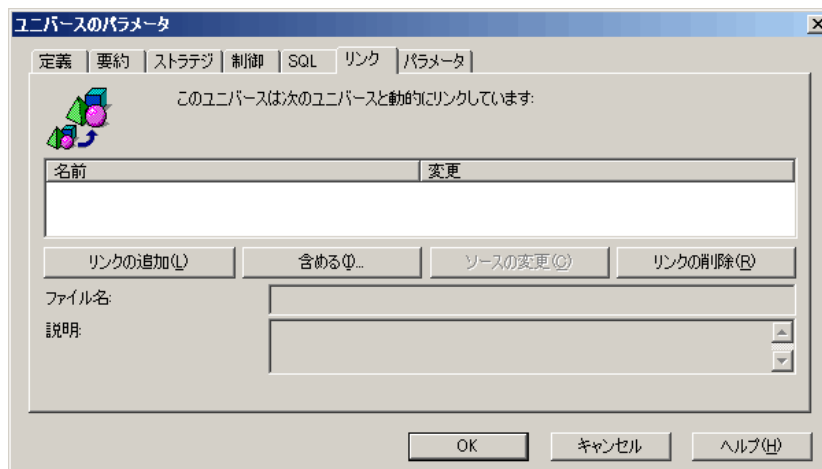
- 1 コアユニバースにリンクするユニバースが、現在開かれていることを確認してください。

たとえば、下の“Beach”ユニバースには、各国の売上げ情報は含まれていますが、リゾートのデータは含まれていません。この売上げユニバースを、リゾートのデータを含んだ“resort”ユニバースにリンクします。売上げ“Beach”ユニバースが派生ユニバース、“resort”ユニバースがコアユニバースになります。



- 2 [編集] > [リンク] の順に選択します。

[ユニバースのパラメータ]ダイアログボックスの[リンク]ページが開きます。

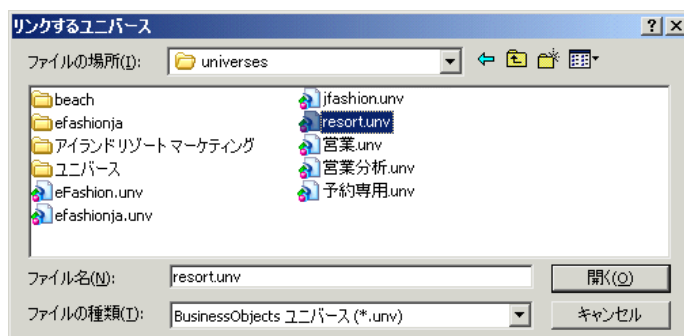


- 3 [リンクの追加]ボタンをクリックします。

[リンクするユニバース] ダイアログボックスが開きます。ドメインで利用できるユニバースが一覧表示されます。

- 4 リンクするユニバースを選択します。リンクするユニバースは、作業中のユニバースで使用するコンポーネントを含んだコアユニバースです。

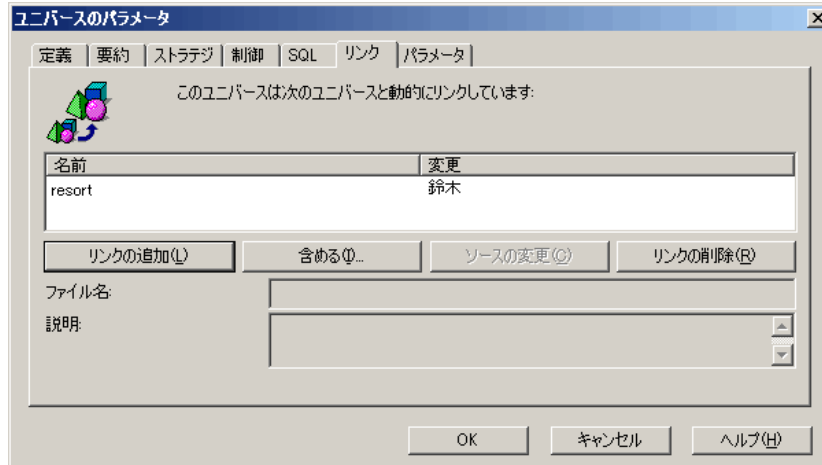
ここでは、[resort]ユニバースを選択します。



選択したユニバースが一度もエクスポートされていない場合は、エラーメッセージが表示されます。リンクを作成する前に、ユニバースが最低1度はエクスポートされている必要があります。

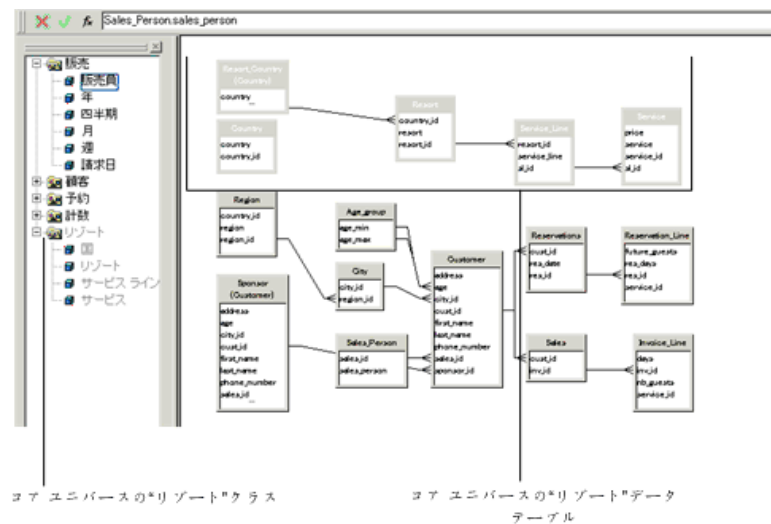
- 5 [開く]をクリックします。

選択したユニバースが一覧に表示されます。



6 [OK]をクリックします。

リンクが作成されました。コアユニバースは、作業中のユニバース内でグレイ表示されています。



6.10.7 派生ユニバースの編集

コアテーブルと派生ユニバーステーブル間に結合を作成して、リンクが作成されました。現在のコンテキストをすべて削除し、新しい構造のコンテキストを再検出する必要があります。

注

派生ユニバース内にあるコアユニバースからの構造、クラス、オブジェクトは変更することができません。

6.10.7.1 派生ユニバースの編集

派生ユニバースを編集する手順は、次のとおりです。

- 1 2つのユニバースの構造間で結合を作成します。

結合を作成することにより、クエリに含まれるオブジェクトに両方の構造からのデカルト積が返されないようにします。

- 2 既存のコンテキストを削除します。
- 3 エイリアスを検出します。
- 4 コンテキストを検出します。
- 5 必要に応じて、新しいオブジェクトを非表示にしたり、作成したりします。

注

コンポーネントの非表示の詳細については、271 ページの「[クラス、オブジェクト、条件の表示と非表示](#)」を参照してください。

6.10.8 リンクの削除

派生ユニバースにコアコンポーネントまたはコアコンポーネントへの結合に基づいたオブジェクトが含まれていない場合にのみ、コアユニバースへのリンクを削除することができます。

6.10.8.1 ユニバースからのリンクの削除

ユニバースからリンクを削除する手順は、次の通りです。

- 1 派生ユニバースを開きます。
- 2 [編集] > [リンク] の順に選択します。
[ユニバースのパラメータ]ダイアログボックスの[リンク]ページが表示されます。
- 3 一覧からコアユニバースの名前をクリックします。
- 4 [リンクの削除]ボタンをクリックします。
- 5 [OK]をクリックします。

コアユニバースからのコンポーネントがアクティブユニバースから削除されました。

6.10.9 コアユニバースの移動

コアユニバースを移動した場合、リンクを維持するには新しい場所を指定する必要があります。

6.10.9.1 移動したコアユニバースへのリンクの更新

移動したコアユニバースへのリンクを更新する手順は、次のとおりです。

- 1 派生ユニバースを開きます。
- 2 [編集] > [リンク] の順に選択します。
- 3 一覧でリンクしているコアユニバースをクリックします。
- 4 [ソースの変更]をクリックします。
[リンクするユニバース] ダイアログボックスが開きます。
- 5 コアユニバースの新しい場所を指定します。
- 6 [開く]をクリックします。
新しいコアユニバースがリンクの一覧に表示されます。

6.10.10 派生ユニバースと値の一覧

コアオブジェクトに関連している値の一覧は、派生ユニバースをリポジトリにエクスポートするときに一緒に保存されません。

コアオブジェクトに関連している値の一覧を保存する手順は、次の通りです。

- 1 派生ユニバースと一緒にリポジトリへエクスポートする値の一覧を含んだオブジェクトと同じ定義を使って、新しいオブジェクトを作成します。
- 2 新しいオブジェクトに、コアオブジェクトと同じ値の一覧を指定します。
- 3 新しいオブジェクトを非表示にします。

非表示オブジェクトには値の一覧を保持する機能があるので、このようにすると値の一覧を派生ユニバースと一緒にインポートまたはエクスポートすることができます。

6.10.11 コアユニバース内の順番に従ったオブジェクトの表示

デフォルトでは、ユーザに表示されるオブジェクトの順番は、派生ユニバースで設定した順番に従います。したがって、コアユニバースでオブジェクトの並び順を変更しても、ユーザへの表示には影響はありません。派生ユニバースのオブジェクト順をコアユニバースと常に同期させるには、使用するデータベースの *.PRM ファイルに次のパラメータを設定する必要があります。

CORE_ORDER_PRIORITY = Y

適切な *.PRM ファイルにパラメータを設定する方法の詳細については、『データアクセスガイド』([ヘルプ]>[データアクセスガイド]) を参照してください。

6.11 ユニバースを他のユニバースに含める

コアユニバースのコンポーネントを派生ユニバースにコピーすることができます。派生ユニバースにコピーされたコンポーネントは、コアユニバースのコンポーネントに依存しません。コピーされたコンポーネントは、コアユニバースにはリンクしません。コアユニバースに変更を加えても、派生ユニバースには継承されません。

6.11.1 コアユニバースの派生ユニバース内へのコピー

コアユニバースを派生ユニバース内にコピーすると、派生ユニバースにコピーされたコンポーネントは、コアユニバースのコンポーネントに依存しません。コピーされたコンポーネントは、コアユニバースにはリンクしません。コアユニバースに変更を加えても、派生ユニバースには継承されません。

コアユニバースを派生ユニバース内にコピーする理由は、次のうちのいずれかです。

- ・ ユニバースの内容を作業中のユニバースにコピーする。
- ・ 2 つのユニバース間の動的なリンクを終了する。

注

2 つのユニバース間に動的なリンクが存在する場合、上記の操作でリンクは削除されます。

6.11.1.1 コアユニバースの派生ユニバース内へのコピー

コアユニバースを派生ユニバースにコピーする手順は、次の通りです。

- 1 ユニバースを開きます。
- 2 [編集] > [リンク] の順に選択します。
[ユニバースのパラメータ]ダイアログボックスの[リンク]ページが表示されます。
- 3 [リンクの追加]ボタンをクリックします。
[リンクするユニバース] ダイアログボックスが開きます。ドメインで利用できるユニバースが一覧表示されます。
- 4 リンクするユニバースを選択します。リンクするユニバースは、作業中のユニバースで使用するコンポーネントを含んだコアユニバースです。
- 5 [含める]をクリックします。
- 6 [OK]をクリックします。

コアユニバースのコンポーネントが、現在のユニバース内に表示されます。

6.12 ストアドプロシージャユニバースの作成

ストアド プロシージャ ユニバースは、Web Intelligence ユーザーがデータベースに常駐するストアド プロシージャにアクセスできる特殊なユニバースです。これが Web Intelligence ユーザーがストアド プロシージャにアクセスできる唯一の方法です。Web Intelligence ユーザーは、ストアド プロシージャ ユニバースを使用して、データベースのストアド プロシージャに基づくレポートを作成できます。

ストアド プロシージャは、コンパイルされた SQL プログラムで、1つ 以上の SQL 文からできており、ターゲットデータベースに常駐して実行されます。

レポートがストアド プロシージャに基づいている場合、Web Intelligence では Desktop Intelligence によって作成されたレポートを開くことはできません。つまり、Web Intelligence ユーザーがストアド プロシージャにアクセスしてレポートを作成するためには、特別なストアド プロシージャ ユニバースを使用する必要があります。

ストアドプロシージャは次の利点を提供します。

- ・ ストアドプロシージャを使えば、コードをカプセル化できます。ストアドプロシージャを使えば、アプリケーションソースの中でデータベースの操作を行うのは1度だけですみ、何度も行う必要がありません。このことにより、アプリケーションのソースのデバッグ作業の効率と安全性が向上します。
- ・ データベーススキーマに変更が加えられたとき、ソースコードの中で影響を受ける部分は、ストアドプロシージャを呼び出している1か所だけですみます。データベーススキーマにどのような変更が加えられても、データベース管理者がそれに対応する作業を行うだけですみ、ソースコード更新の必要が生じません。
- ・ ストアドプロシージャはサーバに置かれているため、高度なセキュリティ制限を設定することができます。これによって、きちんと保護されたストアドプロシージャに対しては、より信用度の高いアクセス許可を与えることができます。
- ・ ストアドプロシージャは、コンパイルを行った後、クライアントのアプリケーションの外に配置することができるため、パスワードや個人データといった慎重な扱いが求められる変数でさえも、SQL構文に使用することができます。

- ・ ストアドプロシージャ使用によるネットワークトラフィックの削減

BusinessObjects XI Release 3.0 では、ストアドプロシージャを Desktop Intelligence や ユニバースデザインツールのユニバースと併用できます。ストアドプロシージャを含むユニバースは、Crystal Reports や Web Intelligence にも利用できます。

ストアドプロシージャユニバースには、次の制限が適用されます。

- ・ ストアドプロシージャユニバースのオブジェクト間で結合はできません。
- ・ ストアドプロシージャユニバースでフィルタを使用することはできません。
- ・ ストアドプロシージャユニバースを標準のユニバースにリンクすることはできません。
- ・ ストアドプロシージャが入っているデータベースやアカウントへのアクセスは、Web Intelligence スーパーバイザが許可します。
- ・ すべてのRDBMSがストアドプロシージャをサポートしているわけではありません。お使いのRDBMSのデータベースガイドを参照して、ストアドプロシージャがサポートされているかを確認してください。
- ・ ストアドプロシージャに含まれるCOMPUTE、PRINT、OUTPUT、STATUSステートメントは実行されません。

ストアドプロシージャを使用したレポート作成についての詳しい説明は『Desktop Intelligenceガイド』を参照してください。

6.12.1 Java Bean ユニバースのストアドプロシージャ

BusinessObjects XI Release 3.0 は、Java Beans でのユニバースの作成をサポートしています。Java Beans ベースのユニバースは、構築されたユニバースのエンティティ関係から、結果セットを返します。

Java Beans ベースのユニバースでは、ストアドプロシージャベースのユニバースと同じワークフローを使用し、同じ利点が得られます。また、以下の制限も共通しています。

- ・ 結合が許可されない。
- ・ ユニバースでフィルタを使用できない。

Java Beans の使用方法に関する詳細は、『データアクセスガイド』を参照してください。

6.12.2 ストアドプロシージャに基づくユニバースの作成

Business Objects は、次のストアドプロシージャをサポートしています。

- ・ パラメータを使用しないストアドプロシージャ
- ・ パラメータを使用した(IN)
- ・ 複数の結果セットを使用するストアドプロシージャ
- ・ 複数のステートメント(SELECT 以外の SQL 文)を使用するストアドプロシージャ

パラメータを使用するストアドプロシージャを作成するには、クイックデザインウィザードアイコンをクリックし、パラメータを使用するストアドプロシージャに基づくユニバースの作成の手順に従います。

関連項目

- ・ 355 ページの[ストアドプロシーダを使用する」](#)
- ・ 356 ページの[入力パラメータを使用するストアドプロシーダに基づくユニバースの作成」](#)
- ・ 357 ページの[複数の結果セットを持つストアドプロシーダ」](#)

6.12.2.1 スタドプロシーダのクラスとオブジェクト

- ・ ユニバースデザインツールは、選択されたストアドプロシーダごとにテーブルを 1 つ(結果セットが複数の場合は複数)生成し、ストアドプロシーダによって返される列ごとにオブジェクトを 1 つ作成します。
- ・ 結果セット構造は、ユーザが関数を記述する際に決定されます。

6.12.2.2 スタドプロシーダを使用する

パラメータを使用しない 1 つまたは複数のストアドプロシーダに基づいてユニバースを作成できます。ツールバーから利用できるクイックデザインウィザードを使用します。

- 1 [クイックデザインウィザード]ツールバーボタンをクリックします。
初期枠が表示されます。
- 2 枠の下部にある[ストアドプロシーダユニバースの選択]チェックボックスをクリックします。
- 3 [開始]をクリックします。
[ユニバースパラメータの定義]パネルが開きます。
- 4 [ユニバース名の入力]フィールドにユニバース名を入力します。
- 5 [データベース接続の選択]のドロップダウンリストからデータベース接続を選択します。
- 6 [次へ]をクリックします。
[初期クラスとオブジェクトの作成]パネルが開きます。
- 7 スタドプロシーダをクリックします。
- 8 [追加]をクリックします。
[ユニバースクラスおよびオブジェクト]枠にストアドプロシーダが作成されます。
- 9 [次へ]をクリックします。
- 10 [完了]をクリックします。[完了]パネルが表示されます。

6.12.2.2.1 ユニバースのストアドプロシーダパラメータ

パフォーマンス改善のため、同じデータソースに対し、同一ユニバースに基づく複数のストアドプロシーダが使用できます。クイックデザインウィザードを使用するか、[挿入]>[ストアドプロシーダ]>[更新]をクリックします。

ユニバースパラメータで、STORED_PROC_UNIVERSE が YES に設定されていることを確認します。これは、現在のユニバースがストアードプロシージャに基づいていることを示しています。

ストアードプロシージャ列での分析エラーを回避するには、複雑な SQL に基づくエイリアスを結果列に割り当てることをお勧めします。たとえば、SUM や COUNT などの集計関数を使用します。エイリアスを持つオブジェクトの作成は制約されません。

制限

ストアードプロシージャは、OUT または動的な結果セットパラメータをサポートしていません。

6.12.2.3 入力パラメータを使用するストアードプロシージャに基づくユニバースの作成

プロシージャによって、データベースのリストから値を選択するよう促すプロンプトが表示される場合は、宣言されたオブジェクトがすでに存在します。

1 つ以上のパラメータを必要とするストアードプロシージャに基づいて、ユニバースを作成できます。入力された値に応じて、プロシージャは適切なファクトテーブルからその値のファクトデータを返します。

- 1 [クイックデザインウィザード]ツールバーボタンをクリックします。
初期枠が表示されます。
- 2 枠の下部にある[ストアードプロシージャユニバースの選択]チェックボックスをクリックします。
- 3 [開始]をクリックします。
[ユニバースパラメータの定義]パネルが開きます。
- 4 [ユニバース名の入力]フィールドにユニバース名を入力します。
- 5 [データベース接続の選択]のドロップダウンリストからデータベース接続を選択します。
- 6 [次へ]をクリックします。
[初期クラスとオブジェクトの作成]パネルが開きます。
- 7 ストアドプロシージャをクリックします。
- 8 [追加]をクリックします。
ストアードプロシージャで入力パラメータが要求される場合は、ストアードプロシージャエディタが表示されます。
- 9 パラメータの一覧からパラメータをクリックします。
- 10 [値]フィールドにパラメータの値を入力するか、プロンプトに入力します。
- 11 [この値を使用]または[別の値を入力]を選択します。
値を入力した場合は、プロシージャの実行時にプロシージャが列を検索して結果が構造に設定される際に、この値がプロシージャに渡されます。
- 12 [別の値を入力]を選択した場合は、プロンプトに値を入力します。
プロンプトメッセージを入力するか、テーブルの subscriber ID 一覧など、既存のオブジェクトに移動して選択します。
- 13 [OK]をクリックします。
- 14 [次へ]をクリックします。

15 [完了]をクリックします。[完了]パネルが表示されます。

6.12.2.3.1 値のプロンプト

ストアードプロシージャの実行時には、プロンプトを使用してパラメータの値を定義します。

デフォルトでは、ストアードプロシージャ構造から取得されたストアードプロシージャパラメータ名がプロンプトに表示されます。

セマンティックを調整し、値の一覧をこのプロンプトに関連付けておくと、一覧から値を追加するのに役立ちます。

ストアードプロシージャパラメータダイアログボックスの各パラメータの前にあるボタンをクリックすると、[詳細] ダイアログボックスが開きます。

6.12.2.4 ストアドプロシージャのプロンプトの値一覧の使い方

ストアードプロシージャを動的パラメータで定義する場合、値の一覧(lov)をプロンプト定義に関連付けると、標準テーブルに基づいたユニバースオブジェクトを選択できます。値の一覧は、標準テーブルから取得する必要があります。この方法は、ユニバースオブジェクトの値の一覧をユーザに提案する場合に便利です。

値の一覧に含めることができるのは、単純値のみです。ユーザ定義の値の一覧を編集または作成することはできません。

注

ストアードプロシージャの定義に含まれないクラスまたはオブジェクトを挿入すると、それらのクラスまたはオブジェクトは非表示になります。これらを表示するようにステータスを変更することはできません。

6.12.2.5 複数の結果セットを持つストアードプロシージャ

例: 複数の結果セットを返すストアードプロシージャ設計時、同じストアードプロシージャに基づいて、ユニバース構造に複数のテーブルが作成されます。

```
CREATE PROCEDURE qaputel.sp.getcustomer_2results
@location varchar(10)
AS
SELECT customer_key as KEYID, CUST_LNAME as Lname
FROM CUSTOMER
WHERE ADDRESS_LINE1 like @location
SELECT PREFIX as PREFIX, GENDER as GENDER, BIRTH_DT as BirthDATE
FROM CUSTOMER
```

複数の結果セット(RS)を処理するには、次のような方法があります。

Stored procedure RS1: a, b + RS2: b, d, e

Table A1: A, B

Table A2: B, D, E

このサンプル式では、同じ ID に基づいて 2 つのテーブルが生成されます。ユニバースデザインツール モジュールの Table A1 を編集する際、Table A2 も編集できます。

列は、ストアードプロシージャの結果セット構造に従って配置されます。2 つのテーブルは、同じストアードプロシージャに基づいて生成されます。この例で、2 つのフローの結果セット構造は同じです。ユニバースデザインツールは、ストアードプロシージャの 2 番目の結果セットで示されるテーブル名を変更します。ユニバースデザインツールでは、ビジネスエレメントを編集できます。

ユニバースデザインツールは、各結果セットに対してユニバースにテーブルを 1 つ作成し、それぞれのテーブルに対応するオブジェクトをいくつか作成します。これらのオブジェクトは互いに独立しています。ユーザは、このユニバースを通常のストアードプロシージャと同じように変更できます。

6.13 ユニバースのテスト

ユニバース内のクラスとオブジェクトの整合性は、[ツール] > [整合性のチェック] の順に選択して定期的に確認してください。また、Web Intelligence で実際にオブジェクトをテストしてください。オブジェクトがクエリに生成する SQL を表示するには、クエリパネルを使ってユニバースオブジェクトでクエリを作成し、[SQL の表示] ボタンをクリックします。

6.13.1 クエリパネルでのオブジェクトのテスト

クエリの SQL をクエリパネルで表示するには、次の手順を実行します。

- 1 [ツール]>[クエリパネル]の順に選択します。

クエリパネルが開きます。

- 2 オブジェクトを右の[結果]枠にドラッグします。
- 3 [SQL]ボタンをクリックします。



- 4 クエリの SQL が表示されます。
- 5 [OK]をクリックし、表示されたら[キャンセル]をクリックしてクエリパネルを閉じます。

6.13.2 ユニバースの整合性のチェック

ユニバースのクラスやオブジェクトを作成したり編集した場合は、必ずユニバースの整合性をチェックしてください。整合性のチェックの詳細については、262 ページの「[ユニバースの整合性の自動チェック](#)」を参照してください。

6.13.3 Web Intelligence でのユニバースのテスト

Web Intelligence でテストクエリを実行し、オブジェクトをテストしてください。オブジェクトのテストでは、次のポイントをチェックします。

- ・ オブジェクトが表示されるか?されない場合は、最終の変更後にユニバースを保存したか?
- ・ SQLは正しいか?
- ・ クエリの結果は正しいか?

結合をテストするには、正しい結果が取得できるかをチェックし、[整合性のチェック]コマンドでスキーマのコンポーネントをチェックします。

ユニバースの最適化

ユニバースを最適化することでクエリ時間が大幅に短縮されるケースがよくあります。ユニバースを最適化する方法は、いくつか考えられます。

- ・ ユニバースのArray Fetchパラメータを最適化する。
- ・ 各テーブルにウェイトを設定する。
- ・ ショートカット結合を使用する。
- ・ データベースに集計テーブルを作成し、これを利用する。

ここでは、それぞれの方法について説明します。

7.1 概要

ユニバースを最適化するには、次のテクニックを使用します。

- ・ 361 ページの [「集計テーブルの使い方」](#)
- ・ 375 ページの [「オブジェクトの SQL での @ 関数 の使用」](#)
- ・ 402 ページの [「外部ストラテジーを使用したユニバース作成のカスタマイズ」](#)
- ・ 417 ページの [「分析関数の使い方」](#)

7.2 集計テーブルの使い方

ユニバースデザインツールには、オブジェクトに SELECT 文を定義して、ベーステーブルではなく、データベースの集計テーブルに対してクエリを実行する機能があります。条件を設定して、クエリが最適化される場合は集計テーブルに対してクエリを実行し、そうでない場合はベーステーブルに対して実行するようにもできます。集計テーブルを使用してクエリを最適化することを、アグリゲート認識といいます。

この章では、アグリゲート認識をユニバースに設定する方法について説明します。

7.2.1 アグリゲート認識

アグリゲート認識とは、データベース内の集計テーブルを利用するようユニバースに設定できる機能です。集計テーブルには、計算済みのデータが含まれています。オブジェクトの SELECT 句で @Aggregate_Aware という関数を使うことによって、未集計のデータを含むテーブルではなく、集計テーブルに対してクエリが実行されます。

集計テーブルを使用するとクエリの実行がスピードアップされるため、SQL トランザクションのパフォーマンスが向上します。

ユニバースのアグリゲート認識の信頼性と効率は、集計テーブルの正確さによって異なります。集計テーブルは、すべてのファクトテーブルと同時に最新表示する必要があります。

集計テーブルを基にした代替定義を持つオブジェクトが少なくとも1つはあるユニバースには、アグリゲート認識が設定されていることになります。これらの定義は、集計のレベルに相当します。たとえば、“売上” オブジェクトを月別、四半期別、年別に集計することができるといった具合です。こういったオブジェクトを集計オブジェクトと呼びます。

集計オブジェクトを持つユニバースに作成されたクエリは、速やかに適切なレベルの集計情報を返します。

7.2.2 アグリゲート認識のデータウェアハウスへの適用

アグリゲート認識は、データウェアハウスを使用する場合は特に便利な機能です。時間、地理、製品の3つのディメンションで構成されたデータウェアハウスを例に説明します。

このデータウェアハウスの一番下のレベルには、顧客と製品に関する日単位の情報が保存されます。つまり、各顧客の一日の製品購入が各行に保存されていて、その行数を割り出す計算式は次のようになります。

365日 x 100都市 x 10製品 = 365,000行!

この方法で年間販売に関する情報を取得した場合、データベースエンジンに大量の行を追加することになります。しかし、実際には会社の年間売り上げを取得するには、それほど多くの行は必要ありません。

3年 x 3カ国 x 3社 = 27行

つまり、テーブルから27行分のデータを取得すればよいことになりますので、これらの行を集計テーブルにあらかじめまとめておけば、より効果的だということです。

7.2.3 アグリゲート認識の設定

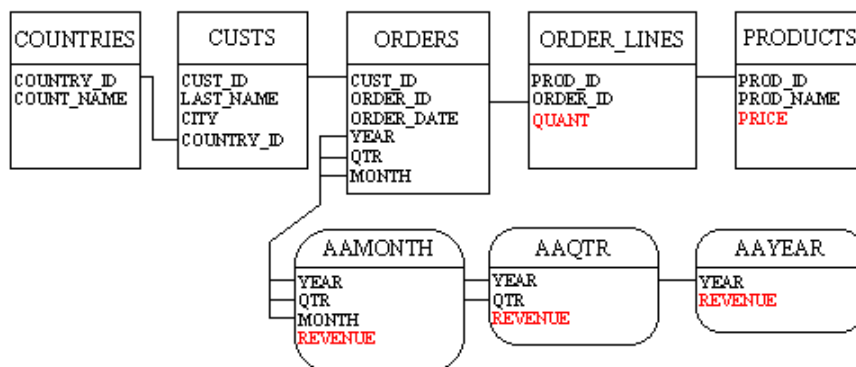
ユニバースのアグリゲート認識の設定は、4段階に分かれます。主な手順は、下図のとおりです。

- ・ オブジェクトの作成
 - 1 オブジェクトのすべての定義 (テーブル/列の組み合わせ) を識別
 - 2 集計レベルにしたがいオブジェクトを整理
 - 3 @Aggregate_Awareness 関数を使ってオブジェクトを作成
- ・ 非互換オブジェクトの指定
 - 1 オブジェクト/集計マトリックスの作成
 - 2 最初の集計テーブルについて、各オブジェクトが互換性があるか (同一集計レベルまたはそれ以上の集計レベル)、または互換性がないか (同一レベルより低い集計レベル) どうかを決定します。
 - 3 このテーブルでは、非互換オブジェクトのみチェック
 - 4 その他の集計テーブルについて、手順 1 ～ 3 を繰り返します。
- ・ 必要なコンテキストの定義

集計レベル毎にコンテキストを1つ定義

- ・ 結果のテスト
 - 1 クエリを数回実行
 - 2 結果を比較

上記の各ステップの詳細は、次のセクションで説明します。説明には、次のスキーマを例として使用します。



このスキーマには、あらかじめ定義された[AAMONTH]、[AAQTR]、[AAYEAR]の3つの集計テーブルが含まれています。

注

上記のスキーマは非常に簡単な例で、このデザインは実際のアプリケーションの典型的な例を紹介するものではありません。一般的に集計テーブルは、時系列を表す1種類のディメンションだけでなく、複数の種類のディメンションを組み合わせたものになります。また、時系列ディメンション(年、四半期、月)は、集計テーブルではなくマスタテーブルで定義されているのが普通です。

7.2.4 オブジェクトの作成

ユニバースにアグリゲート認識を設定する最初のステップは、アグリゲート認識を組み込むオブジェクトを決定することです。メジャーオブジェクトまたはディメンションオブジェクトを使用できます。

上記スキーマをベースに、次の定義を使って“売上高”オブジェクトを作成したとします。

PRODUCTS.PRICE*ORDER_LINES.QUANT

非集計テーブルを使ってデータを集計する代わりに、データベースの集計テーブルを使うよう、このオブジェクトの定義を変更することになります。

“売上高”オブジェクトをアグリゲート認識オブジェクトとして再定義していく過程のそれぞれのステップでは、集計テーブルを使用する他のオブジェクトにも同じ処理を行う必要があります。

7.2.5 集計オブジェクトのすべての組合わせの確認

次のステップでは、さまざまなテーブルのオブジェクトの組合わせをすべて識別します。以下の方法で“売上げ”オブジェクトを定義できます。

- ・ AAMONTH.REVENUE
- ・ AAYEAR.REVENUE
- ・ AAQTR.REVENUE
- ・ PRODUCTS.PRICE*ORDER_LINES.QUANT

7.2.6 集計レベル順のオブジェクトの整列

オブジェクトのすべての組合わせを確認したら、集計のレベルにしたがってそれらを整列します。

- ・ 最高レベルの集計は[AAYEAR.REVENUE]です。
- ・ 次のレベルは[AAQTR.REVENUE]です。
- ・ 次のレベルは[AAMONTH.REVENUE]です。
- ・ 最低レベルの集計は[PRODUCTS.PRICE*ORDER_LINES.QUANT]です。

7.2.7 @Aggregate_Aware 関数を使った集計オブジェクトの定義

次に@Aggregate_Aware関数を使って、アグリゲート認識を設定するすべてのオブジェクトを定義します。
@Aggregate_Aware関数を使うと、まず、そのパラメータに指定されたすべての集計テーブルにクエリが実行されます。集計テーブルが適切でない場合は、非集計テーブルを使った本来の集計クエリが実行されます。@関数の詳細については、375 ページの「[オブジェクトの SQL での @ 関数の使用](#)」を参照してください。

@Aggregate_Aware 関数を使った“売上高”オブジェクトの SELECT 文は、次のとおりです。

@Aggregate_Aware関数の構文は、次の通りです。

```
@Aggregate_Aware(sum(agg_table_1), ... sum(agg_table_n))
```

[agg_table_1]が最高レベルの集計で、最低レベルは[agg_table_n]です。

引数には、すべての集計テーブルの名前を入力する必要があります。テーブル名の順序にも注意してください。優先順位は左から右です。

7.2.7.1 @Aggregate_Aware関数を使ってオブジェクトを定義する

@Aggregate_Aware関数を使ったオブジェクトの再定義は、次の要領で行います。

- 1 オブジェクトをダブルクリックします。

オブジェクトの[プロパティを編集]ダイアログボックスが表示されます。

- 2 [SELECT 句]ボックスの横にある[>>]ボタンをクリックします。

[SELECT 句の編集]ダイアログボックスが開きます。

- 3 SELECT 句の先頭をクリックします。

または

オブジェクトに SELECT 句が定義されていない場合は、[Select]ボックスの任意の場所をクリックします。

カーソルがボックスの左上に表示されます。

- 4 [関数]枠の[@ 関数]の左にある[+]記号をクリックします。

使用できる @ 関数が表示されます。



- 5 [@Aggregate_Aware]をダブルクリックします。

SELECT句に@Aggregate_Aware関数の構文が自動的に挿入されます。構文の説明は、ダイアログボックスの下方にある[説明]ボックスに表示されます。この説明を参考にして、@関数のパラメータを入力します。

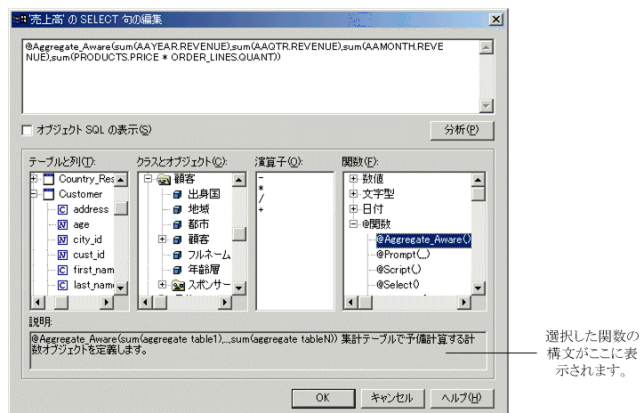
- 6 @Aggregate_Aware 関数の () 内に集計を、上位レベルから下位レベルの順に挿入します。

- 7 それぞれの集計レベルはカンマで区切ります。[売上げ高]オブジェクトの場合は、次の構文になります。

```
@Aggregate_Aware(sum (AAYEAR.REVENUE), sum(AAQTR.REVENUE), sum (AAMONTH.REVENUE),
sum(PRODUCTS.PRICE*ORDER_LINES.QUANT))
```

- 8 [分析]をクリックすると、クエリの構文を確認できます。

売上高に対する SQL エディタの SELECT 文SQL エディタは、次のように表示されます。



9 各ダイアログボックスで[OK]をクリックします。

この例では、ディメンションオブジェクトの[Year]と[Quarter]も@Aggregate_Aware関数を使って再定義する必要があります。

7.2.8 非互換オブジェクトの指定

次のステップでは、ユニバースの各集計テーブルに対する非互換オブジェクトを指定します。指定した一連の非互換オブジェクトにしたがい、SQLを生成する際に無視すべき集計テーブルが決定されます。

オブジェクトは、それぞれの集計テーブルに対して互換または非互換になります。互換性の規則は以下の通りです。

- ・ オブジェクトがテーブルと同じレベルか上位レベルにある場合は、テーブルと互換であるといえます。
- ・ オブジェクトがテーブルより下位レベルにある場合(またはテーブルと関連性のない場合)は、テーブルと非互換であるといえます。

7.2.8.1 マトリックスを使ったオブジェクトの分析

オブジェクトや集計テーブルの互換性を分析するには、マトリックスを構築すると便利です。マトリックスの最初の2列には、クラスとオブジェクトを入れます。次に、ユニバース内の各集計テーブルに対するヘッダ列を入れます。この例での空白マトリックスは以下のようになります。

クラス	オブジェクト	AAYEAR	AAQTR	AAMONTH
顧客	顧客のコード番号 (CUSTOMER.CUST_ID)			
	顧客名 (CUSTOMER.LAST_NAME)			
	顧客の在住都市 (CUSTOMER.CITY)			
	顧客の国籍 (COUNTRIES.COUNT_NAME)			
製品	製品のコード番号 (PRODUCT.PROD_ID)			
	製品名 (PRODUCT.PROD_NAME)			
注文	注文年 (AAYEAR.PROD_NAME)			
	注文四半期 (AAQTR.QTR)			
	注文月 (AAMONTH.MONTH)			
	注文日 (ORDERS.ORDER_DATE)			

クラス	オブジェクト	AAYEAR	AAQTR	AAMONTH
売上げ メジャー	売上げ (@Aggregate_Aware(...))			

各テーブルに対して、オブジェクトが非互換であれば X を入力します。

この例では、マトリックスは以下のようになります。

クラス	オブジェクト	AAYEAR	AAQTR	AAMONTH
顧客	顧客のコード番号 (CUSTOMER.CUST_ID)	X (n)	X (n)	X (n)
	顧客名 (CUSTOMER.LAST_NAME)	X (n)	X (n)	X (n)
	顧客の在住都市 (CUSTOMER.CITY)	X (n)	X (n)	X (n)
	顧客の国籍 (COUNTRIES.COUNT_NAME)	X (n)	X (n)	X (n)
製品	製品のコード番号 (PRODUCT.PROD_ID)	X (n)	X (n)	X (n)
	製品名 (PRODUCT.PROD_NAME)	X (n)	X (n)	X (n)
注文	注文年 (AAYEAR.PROD_NAME)	- (s)	- (h)	- (h)
	注文四半期 (AAQTR.QTR)	X (l)	- (s)	- (h)

クラス	オブジェクト	AAYEAR	AAQTR	AAMONTH
	注文月 (AAMONTH.MONTH)	X (l)	3 (l)	- (s)
	注文日 (ORDERS.ORDER_DATE)	X (l)	X (l)	X (l)
売上げ メジャー	売上げ (@Aggregate_Aware(...))	-	-	-

X (n): このオブジェクトは集計テーブルとは関係ありません。互換オブジェクトです。

X (l): オブジェクトは集計テーブルより下位の集計レベルです。情報の派生には使用できません。非互換オブジェクトです。

- (s): オブジェクトは集計テーブルと同じ集計レベルです。情報の派生に使用できます。互換オブジェクトです。

- (h): オブジェクトは集計テーブルより上位の集計レベルです。情報の派生に使用できます。互換オブジェクトです。

7.2.9 非互換オブジェクトの指定

次に非互換オブジェクトを指定します。非互換オブジェクトは、[集計テーブルの自動参照]ダイアログボックス([ツール] > [集計テーブルの自動参照])で指定します。

集計テーブルに対する非互換オブジェクトは、[集計テーブルの自動参照]ダイアログを使用して以下の要領で指定します。

- 1 [ツール] > [集計テーブルの自動参照]の順に選択します。

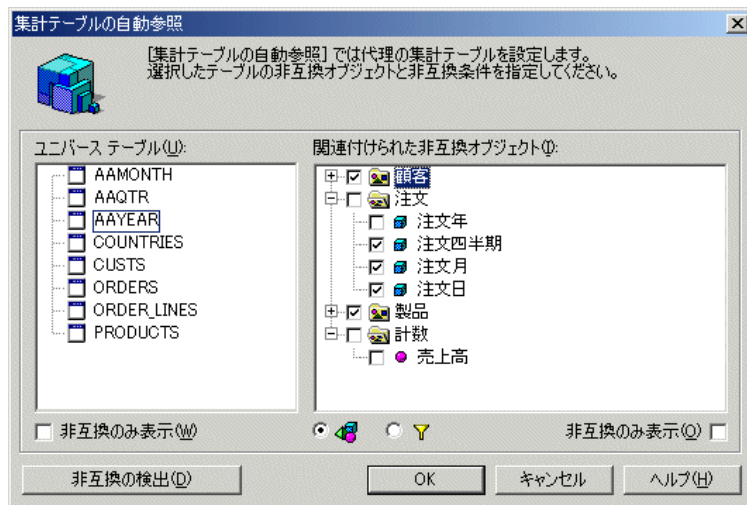
[集計テーブルの自動参照]ダイアログボックスが表示されます。このダイアログボックスは、次の2つの枠で構成されています。

- ・ ユニバースのすべてのテーブルを表示する [ユニバーステーブル]
- ・ ユニバースのすべてのオブジェクトを表示する [関連付けられた非互換オブジェクト]

- 2 左のボックスで、集計テーブルをクリックします。

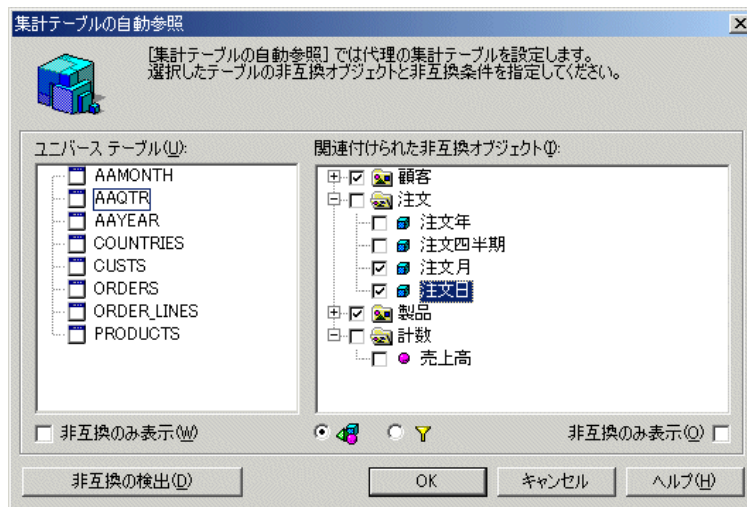
- 3 右のボックスで、各非互換オブジェクトのチェックボックスを選択します。

たとえばマトリックスに従うと、AAYEARテーブルでは[顧客]クラスのすべてのオブジェクトが非互換オブジェクトです。クラス名の横にあるチェックボックスを次のようにチェックします。

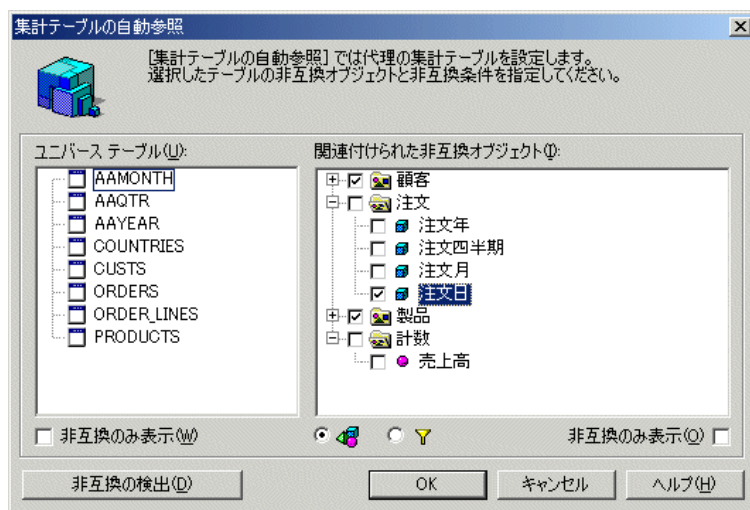


- 4 ユニバースの各集計テーブルで上記の手順を繰り返します。

AQTRテーブルに対する非互換オブジェクトは、次の通りです。



AAMONTHテーブルに対する非互換オブジェクトは、1つだけです。



5 すべてのテーブルですべての非互換オブジェクトが指定されたら、[OK]をクリックします。

注

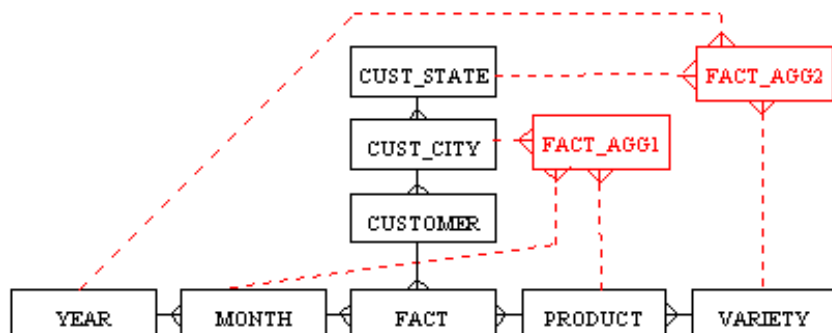
このダイアログボックスには、非互換オブジェクトの指定をサポートする[非互換の検出]ボタンがあります。テーブルを選択してこのボタンをクリックすると、ユニバースデザインツールにより非互換と見なされたオブジェクトが自動的に検出されます。[非互換の検出]で検出された非互換オブジェクトは、あくまでも目安であり、最終的な判断はユーザ自身で行ってください。

7.2.10 集計テーブルのループの解決

データベースに集計テーブルが複数含まれる場合、コンテキストを使ってすべてのループを解消しなければなりません。

例 集計テーブルのデータベースにコンテキストを使ってループを解決する

集計テーブルのある簡単なスキーマを例にとってみましょう。



次のポイントに注意してください。

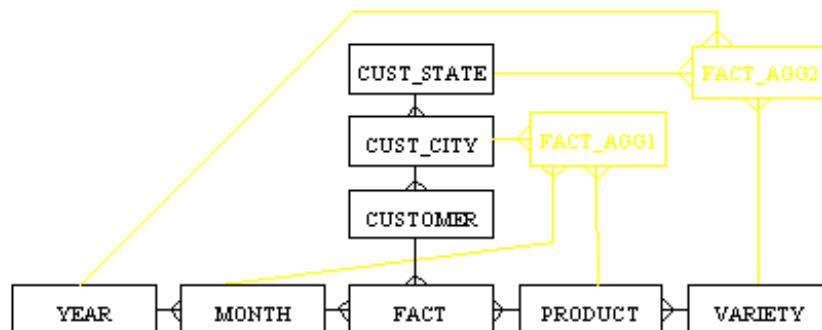
- ・ [FACT_AGG1]は、[FACT]とほとんど同一の集計テーブルです。含まれるのは、[CUST_CITY]、[PRODUCT]、[MONTH]別に集計されるメジャーと、[CUSTOMER]の City キー、Product キー、Month キーです。
- ・ [FACT_AGG2] もほとんど [FACT] テーブルと同じ集計テーブルです。メジャーが[Customer State]、[Product]、[Year]で集計されています。
- ・ このスキーマでは、メジャー(つまりキーパーフォーマンス指標)がすべてのファクトテーブルにあります。[Sales Revenue]は、[FACT_AGG1]、[FACT_AGG2]、[FACT]に格納され、各テーブルのそれぞれのレベルごとに集計されます。

[Sales Revenue]と[CUST_STATE]を使ってクエリを作成すると、[CUST_STATE]と[CUST_CITY]の結合ではなく、[CUST_STATE]と[FACT_AGG2]の結合が使用されます。

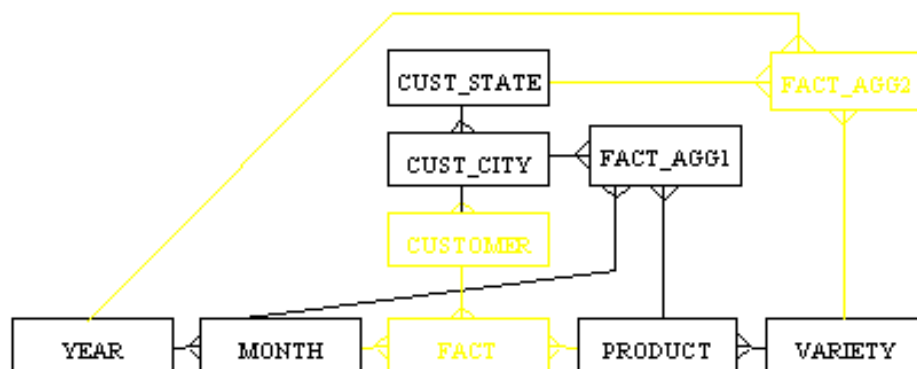
このクエリを実行するには、[FACT]、[FACT_AGG1]、[FACT_AGG2]の 3 つのコンテキストを定義しておく必要があります。コンテキストを別の名前に変更する必要はありません。ユーザにとって明白であるからです。

3つのコンテキストに含まれる結合を、次ページに示します。それぞれのスキーマではコンテキストを黒の結合で表示しています。

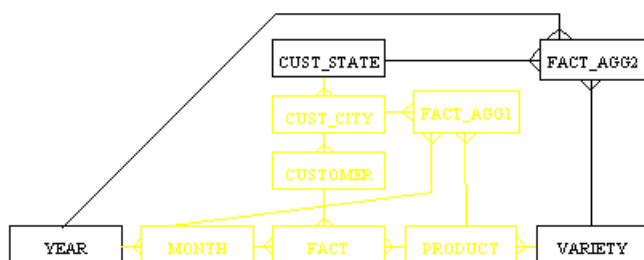
FACTコンテキスト



FACT_AGG1コンテキスト



FACT_AGG2コンテキスト



7.2.11 アグリゲート認識のテスト

アグリゲート認識設定の最終ステップでは、Web Intelligence で結果をテストします。

最初の例に基づいて以下のクエリー実行し、結果を比較します。

- ・ 売上げに対する注文年
- ・ 売上げに対する注文四半期
- ・ 売上げに対する注文月
- ・ 売上げに対する顧客
- ・ 売上げに対する製品

7.3 オブジェクトの SQL での @ 関数の使用

@ 関数は、オブジェクトに SQL を指定するときにさらにフレキシブルなメソッドを提供する特別な関数です。@ 関数は、オブジェクトの[SELECT 句の編集]ボックスの[関数]枠で使います。

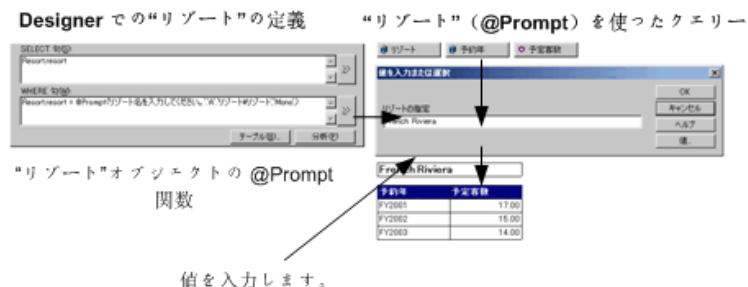
オブジェクトの SELECT 文または WHERE 句に、1 つまたは複数の @ 関数を挿入することができます。次の @関数を使用できます。

@ 関数	説明	通常の使用箇所
@Aggregate_Aware	ディメンションの値と集計データを持った列をオブジェクトに組み込みます。	SELECT 句
@Prompt	この句によって、SQL にプロンプトが挿入されます。ユーザがクエリを実行するときに、@Prompt 関数を使用するオブジェクトがクエリに含まれているたびに、制限値を入力するように求めるプロンプトが表示されます。	SELECT 句 WHERE 句
@Script	@Script 関数を使用するオブジェクトがクエリに含まれるたびに、スクリプトを実行します。	WHERE 句
@Select	別のオブジェクトの SELECT 句を使用できるようになります。	SELECT 句
@Variable	メモリに保存された変数の値を呼び出します。	WHERE 句
@Where	別のオブジェクトのWHERE句を再利用できるようになります。	WHERE 句

例 @Prompt 関数を使って指定された値だけを返す

@Prompt 関数は、ユニバースデザインツールで使用できる @Functionsの 1 つです。@Prompt 関数を設定したオブジェクトが Web Intelligence クエリーで実行されると、メッセージ ボックスが表示されます。

このメッセージボックスはユーザにオブジェクトの値を入力するよう要求します。クエリは、下記のように入力された値に対応するデータだけを返します。



7.3.1 オブジェクトへの @ 関数の挿入

オブジェクトの SQL 定義に @ 関数を挿入する手順は、次のとおりです。

- 1 [ユニバース]枠のオブジェクトをダブルクリックします。

オブジェクトの[プロパティを編集]ダイアログボックスが表示されます。

- 2 [SELECT 句]ボックスの横にある[>>]ボタンをクリックします。

または

[WHERE 句]ボックスの横にある[>>]ボタンをクリックします。

[SELECT 句の編集]または[WHERE 句の編集]ダイアログボックスが表示されます。

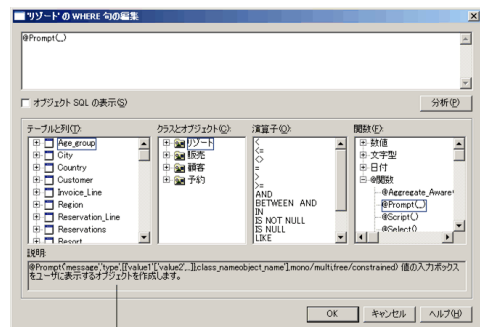
- 3 Select 句または Where 句の @関数を追加する位置をクリックします。まだSQL構文を定義していない場合は、空白のボックス内のどこかをクリックすると、カーソルがボックスの左上に表示されます。カーソルがボックスの左上に表示されます。
- 4 Click the @Functions node in the Functions pane.

使用できる @ 関数の一覧が表示されます。



- 5 @ 関数をダブルクリックします。

@ 関数の構文が、オブジェクトの SELECT 文または WHERE 句に追加されます。構文の説明は、ダイアログボックスの下方にある[説明]ボックスに表示されます。この説明を参考にして、@ 関数のパラメータを入力します。



@ 関数構文の説明

- 6 ダイアログボックスの上の枠に必要なパラメータを入力します。
- 7 [分析]をクリックすると、クエリの構文を確認できます。
- 8 各ダイアログボックスで[OK]をクリックします。

7.3.2 @Aggregate_Aware

@Aggregate_Aware 関数を使うと、データベース内の集計テーブルを利用してオブジェクトを定義できます。データベースに集計テーブルがある場合、集計データを取得するクエリでは、ファクトテーブルやイベントテーブルよりも、集計テーブルに SELECT 文を実行する方がはるかに簡単です。集計テーブルとの互換性がないことが宣言されたオブジェクトは、集計テーブルを使用することはできませんが、クエリに対してベーステーブルを使用できます。

@Aggregate_Aware 関数を使って、ユニバースにアグリゲート認識を設定することもできます。これには、@Aggregate_Aware 関数の使用に関連した一連の操作が必要となります。

- ・ 各集計テーブルに対する非互換オブジェクトを指定します。
- ・ 集計テーブルのループをすべて解決します。
- ・ 集計テーブルをテストして、正しい結果が返されることを確認します。

7.3.2.1 @Aggregate_Aware 関数の構文

@Aggregate_Aware 関数の構文は、次のとおりです。

```
@Aggregate_Aware(sum(agg_table_1), ...
sum(agg_table_n))
```

引数には、すべての集計テーブルの名前を入力する必要があります。テーブル名の順序にも注意してください。優先順位は左から右です。

構文	説明
agg_table_1	最高レベルの集計です。
agg_table_n	最低レベルの集計です。

例

```
@Aggregate_Aware( R_Country.Revenue,
  R_Region.Revenue,
  R_City.Revenue,
  R_Customer.Revenue,
  R_Age_Range.Revenue,
  sum(Invoice_Line.days * Invoice_Line.nb_guests * Service.price)
)
```

この例では、この関数を使用するオブジェクトがどのテーブルとも非互換であると宣言された場合、これらのテーブルは無視されます。R_Country テーブルおよび R_Region テーブルと非互換のオブジェクトでは、使用される SQL が次のようになります。

```
@Aggregate_Aware( R_City.Revenue,
  R_Customer.Revenue,
  R_Age_Range.Revenue,
  sum(Invoice_Line.days * Invoice_Line.nb_guests * Service.price)
)
```

7.3.3 @Prompt

@Prompt 関数を使用して、クエリにプロンプトを挿入します。プロンプトを使用すると、ユーザがレポートを作成するときにデータを制限したり、大きい値オブジェクトの使用を容易にしたりすることができます。@Prompt 関数は、オブジェクトの SELECT 文または WHERE 句に定義します。この関数は、そのオブジェクトをクエリで使用するとユーザに 1 つまたは複数の制限値を入力するよう要求したり、値または値の一覧を選択をするように要求します。クエリを実行すると、値を問い合わせるメッセージボックスが表示されます。

@Prompts は、条件の値を表示せずに、生成される SQL に強制的に制限を適用する場合に使用します。

オプションで、プロンプトにデフォルト値を定義できます。デフォルト値を含むプロンプトの動作は、デフォルト値を使用する Web Intelligence プロンプトと同じです。

@Prompt の定義は、次の方法で編集できます。

- ・ @Prompt エディタを使用する。
- ・ 条件に対する[プロパティの編集]ダイアログの[定義]枠にある[SELECT]フィールドまたは[WHERE]フィールドに定義を入力する。
- ・ [プロパティの編集]ダイアログの[詳細設定]編集枠に定義を入力する。

注

- ・ 異なる一次キーなどの、使用方法以外は非常に類似している2つのプロンプトを使用する場合は、同じ質問(プロンプトテキスト)を使用しないでください。これは、システムで2つのプロンプトが識別できなくなり、適切でない応答が返される場合があるためです。
- ・ @Prompt の定義は潜在的に複雑であるため、特に値の一覧を入力するときは、@Prompt エディタを使用することをお勧めします。
- ・ @Prompt 関数が単一値である場合、同一のクエリ内で @Variable 関数と @Prompt 関数を統合できます。

関連項目

- ・ 379 ページの[@Prompt エディタ](#)
- ・ 384 ページの[既存の @Prompt 式の編集](#)
- ・ 391 ページの[@Prompt 関数の構文](#)
- ・ 385 ページの[手動による SQL 文の @Prompt 関数の定義](#)

7.3.3.1 @Prompt エディタ

@Prompt エディタでは、Web Intelligence ユーザーまたは Desktop Intelligence ユーザーがリレーショナル ユニバースか OLAP ユニバースのいずれかに対してクエリーを実行するときに表示される、プロンプトの定義や編集を行うことができます。エディタにより、プロンプトの定義や編集のプロセスが簡略化されます。プロンプトの定義は枠の下部に表示され、プロンプトに対して別の値を定義すると自動的に更新されます。既存の @Prompt 式を右クリックすると、@Prompt 文字列の構文が正しい場合にのみ[[@Prompt の編集](#)]メニューアイテムが有効化されます。

@Prompt の定義は、[プロパティの編集]ダイアログボックスの[SELECT]枠または[WHERE]枠で直接入力することもできます。

関連項目

- ・ 391 ページの[@Prompt 関数の構文](#)
- ・ 385 ページの[手動による SQL 文の @Prompt 関数の定義](#)
- ・ 382 ページの[プロンプトに対する静的な値の一覧の定義](#)
- ・ 383 ページの[プロンプトに対する値の一覧としてのユニバースオブジェクトの選択](#)
- ・ 383 ページの[プロンプトに対するカスタムの値の一覧の選択](#)
- ・ 383 ページの[プロンプトに対する値の一覧としてのユニバースオブジェクトの選択](#)
- ・ 384 ページの[既存の @Prompt 式の編集](#)

7.3.3.2 @Prompt エディタ用の @Prompt 式のプロパティ

プロンプトに対する次のプロパティを編集できます。

プロパティ	説明
メッセージ	<p>ユーザに表示されるプロンプトメッセージ。たとえば、「国を選択してください」と入力できます。プロンプトテキスト(質問)。</p> <p>デフォルト: Enter values</p>
値の型	<p>ユーザが入力または選択するデータタイプ。この設定によって、ユーザが確実に適切なタイプのデータを入力または選択するようにします。次の中から選択します。</p> <ul style="list-style-type: none"> Alphanumeric(A) Numeric(N) Date(D) <p>デフォルト: Alphanumeric</p>
キーの種類	<p>選択モードとして一次キーが選択されている場合は、ユーザが入力または選択するキーの種類を設定します。次の中から選択します。</p> <ul style="list-style-type: none"> Alphanumeric(A) Numeric(N) Date(D) <p>デフォルト: なし</p>
複数選択の許可	<p>このオプションが選択されている場合、ユーザは複数の値を入力または選択することができます。</p> <p>デフォルト: 選択されていない(ユーザは1つの値のみを選択または入力できる)</p>
選択モード	<p>プロンプトに必要な値をユーザがどのように選択するかを定義します。以下を選択できます。</p> <ul style="list-style-type: none"> Free: ユーザは任意の値を入力できます。 Constrained: ユーザは提案された値から選択する必要があります。 Primary_key*: クエリに一次キー値を使用すると、応答時間が大幅に増加します。ユニバースのオブジェクトに対する一次キーを選択または入力します。プロンプト時に、ユーザはオブジェクト名を選択しますが、クエリには実際に、対応する一次キー値が使用されます。*データベーステーブルに対してインデックス認識が設定されている場合のみ、一次キーの設定を使用できます。 <p>デフォルト: Free</p>
最後に選択した値を維持	<p>このオプションが選択されている場合、最後に使用した値が、次のクエリ実行時に提案されます。このオプションが選択されていない場合は、常に、デフォルト値が使用されます。</p> <p>デフォルト: 選択されていない</p>

プロパティ	説明
表示値	<p>ユーザは、値の一覧から 1 つまたは複数の項目を選択する必要があります。値の一覧は、次の方法で定義できます。</p> <ul style="list-style-type: none"> ・ [値の一覧]ボックスに値の一覧を入力する。これが静的一覧になります。 ・ ユニバースからオブジェクトを選択する ・ ファイル入力ウィザードを使用してファイルをインポートする <p>デフォルト: Static</p>
デフォルト値	プロンプトのデフォルト値を指定できます。

関連項目

- ・ 391 ページの[@Prompt 関数の構文](#)」
- ・ 385 ページの[手動による SQL 文の @Prompt 関数の定義](#)」
- ・ 382 ページの[プロンプトに対する静的な値の一覧の定義](#)」
- ・ 383 ページの[プロンプトに対する値の一覧としてのユニバースオブジェクトの選択](#)」
- ・ 383 ページの[プロンプトに対するカスタムの値の一覧の選択](#)」
- ・ 383 ページの[プロンプトに対する値の一覧としてのユニバースオブジェクトの選択](#)」
- ・ 384 ページの[既存の @Prompt 式の編集](#)」

7.3.3.3 プロンプトエディタを使用した @Prompt 式の作成

ユニバースをユニバースデザインツールで開き、オブジェクトを作成または編集します。

- 1 条件の[SELECT]フィールドまたは[WHERE]フィールドで、プロンプトを追加する式内を右クリックし、ショートカットメニューから[@Prompt エディタ]を選択します。
- 2 [メッセージ]ボックスに、ユーザに表示するメッセージを入力します。
- 3 以前のレポートに使用した値を提案するプロンプトにするには、[最後に選択した値を維持]を選択します。レポートを初めて実行するときは、デフォルト値が存在する場合はデフォルト値が提案されます。
- 4 ユーザが複数の値を入力または選択できるようにするには、[複数値の選択を認める]を選択します。
- 5 [選択モード]を設定します。[フリー]選択の場合、ユーザは許容範囲の任意の値を入力できます。[制限]選択の場合、ユーザは値の一覧から値を選択する必要があります。[一次キー]の場合、ユーザはオブジェクト名を選択しますが、クエリにはオブジェクトのプライマリキーが使用されます。[一次キー]の設定は、インデックス認識を設定している場合に限り使用できます。

注

[Constrained]オプションが選択されていて、値の一覧が指定されていない場合は、対応するタブの色が赤になり、[OK]ボタンが無効化されます。強調表示されたタブにカーソルを移動すると、コンテキストに応じたツールヒントにより、問題の修正方法を示すメッセージが表示されます。

- 6 選択モードに対する[値の型]を[英数字]、[数値]、または[日付]から選択します。
- 7 選択モードに[一次キー]を選択している場合は、[キーの種類]に[英数字]、[数値]、または[日付]を設定します。
- 8 複数選択を使用している場合は、値の一覧を定義します。値の一覧を入力またはインポートするか、ユニバースオブジェクトを選択してください。
- 9 [デフォルト値]タブをクリックし、デフォルト値を定義します。複数選択を許可する場合は、複数のデフォルト値を設定できます。
- 10 [OK]をクリックします。
@Prompt エディタが閉じ、[プロパティの編集]枠の条件文にプロンプト式が表示されます。
- 11 プロンプトを検証し、[解析]をクリックします。
構文に誤りがあると、SQL フラグメントにエラーがあることを示す「分析に失敗しました」というメッセージが表示されます。

関連項目

- ・ 391 ページの[@Prompt 関数の構文](#)
- ・ 385 ページの[手動による SQL 文の @Prompt 関数の定義](#)
- ・ 382 ページの[プロンプトに対する静的な値の一覧の定義](#)
- ・ 383 ページの[プロンプトに対する値の一覧としてのユニバースオブジェクトの選択](#)
- ・ 383 ページの[プロンプトに対するカスタムの値の一覧の選択](#)
- ・ 383 ページの[プロンプトに対する値の一覧としてのユニバースオブジェクトの選択](#)
- ・ 384 ページの[既存の @Prompt 式の編集](#)

7.3.3.4 プロンプトに対する静的な値の一覧の定義

@Prompt エディタの下部には、静的な値の一覧の定義に使用するテーブル枠があります。上矢印と下矢印を使用して、入力する値の位置を変更できます。[キャプション]タイトルテキストの色が赤の場合は、値の入力または値の修正のいずれかを行う必要があります。

- 1 [キャプション]フィールドに最初の値を入力します。[一次キー]を[選択モード]として選択している場合は、2 番目のフィールドにインデックス値を入力します。
- 2 [+]をクリックして、静的な値の一覧テーブルに値を挿入します。
- 3 テーブルにさらに値を入力して、値の一覧を完成させます。
- 4 [デフォルト値]枠にデフォルト値を入力します。

デフォルト値は、ユーザがレポートを実行するときに提案されます。[最後に選択した値を維持]が設定されていると、レポートを最後に実行したときに使用した値が提案されます。このオプションが設定されていない場合は、レポートを実行するたびにデフォルト値が提案されます。

注

- ・ 静的な一覧の値を編集するには、静的な一覧の値を選択し、次にテーブルの値を編集して、[更新]ボタンをクリックします。
- ・ 値を削除するには、その値を選択して[-]をクリックします。

関連項目

- ・ 391 ページの[@Prompt 関数の構文](#)」
- ・ 385 ページの[手動による SQL 文の @Prompt 関数の定義](#)」
- ・ 383 ページの[プロンプトに対する値の一覧としてのユニバースオブジェクトの選択](#)」
- ・ 383 ページの[プロンプトに対するカスタムの値の一覧の選択](#)」
- ・ 383 ページの[プロンプトに対する値の一覧としてのユニバースオブジェクトの選択](#)」
- ・ 384 ページの[既存の @Prompt 式の編集](#)」

7.3.3.5 プロンプトに対する値の一覧としてのユニバースオブジェクトの選択

プロンプトに対する値の一覧として、ユニバースオブジェクトを選択できます。

- 1 @Prompt エディタで、値の一覧枠から[ユニバースオブジェクト]を選択します。
- 2 プロンプトで使用するユニバースオブジェクトに移動し、選択します。
- 3 オブジェクトをダブルクリックします。
@Prompt エディタが閉じ、[プロパティの編集]枠の条件文にプロンプト式が表示されます。

関連項目

- ・ 391 ページの[@Prompt 関数の構文](#)」
- ・ 385 ページの[手動による SQL 文の @Prompt 関数の定義](#)」
- ・ 382 ページの[プロンプトに対する静的な値の一覧の定義](#)」
- ・ 383 ページの[プロンプトに対するカスタムの値の一覧の選択](#)」
- ・ 383 ページの[プロンプトに対する値の一覧としてのユニバースオブジェクトの選択](#)」
- ・ 384 ページの[既存の @Prompt 式の編集](#)」

7.3.3.6 プロンプトに対するカスタムの値の一覧の選択

データファイル入力ウィザードを使用すると、@Prompt 関数に挿入するカスタムの値の一覧を @Prompt エディタの "キャプション" テーブルにインポートできます。Excel ファイルとテキストファイルのみがサポートされています。

- 1 @Prompt エディタで、値の一覧枠から[ファイルの入力]を選択します。
データファイル入力ウィザードが表示されます。
- 2 [参照]をクリックして、使用する入力ファイルを参照します。
- 3 [開く]をクリックします。
- 4 [次へ]をクリックします。

- 5 値の一覧ファイルの最初の行が一覧の列のヘッダまたはタイトルの値として使用されている場合は、[最初の行が列のヘッダを指定する]を選択します。
- 6 [ファイルのエンコード]タイプを選択します。
- 7 入力ファイル内のデータを区切るために使用する[区切り記号]を設定します。
- 8 [次へ]をクリックします。
- 9 [一意の列値を取得する]をクリックして、一意の値だけが提案されるようにします。
- 10 [上位レコードの数を取得する]をクリックして、ユーザに提案される値の数を定義します。
- 11 [列マップ]枠を使用して、キャプションおよび一次キー値に使用する列を選択します。
- 12 [列での並べ替え]をクリックして選択した列を並べ替え、[昇順]を選択して列を昇順に並べ替えます。[昇順]が選択されていない場合、列は降順でソートされます。
- 13 [完了]をクリックします。
列と行の追加に関する確認メッセージが表示されます。[はい]をクリックして選択内容を確定します。
- 14 カスタムの値の一覧が @Prompt エディタの値の一覧テーブルに挿入され、値が @Prompt の定義に挿入されます。
- 15 [OK]をクリックして SELECT 句または WHERE 句に @Prompt を挿入します。

関連項目

- ・ 391 ページの[@Prompt 関数の構文](#)」
- ・ 385 ページの[手動による SQL 文の @Prompt 関数の定義](#)」
- ・ 382 ページの[プロンプトに対する静的な値の一覧の定義](#)」
- ・ 383 ページの[プロンプトに対する値の一覧としてのユニバースオブジェクトの選択](#)」
- ・ 383 ページの[プロンプトに対する値の一覧としてのユニバースオブジェクトの選択](#)」
- ・ 384 ページの[既存の @Prompt 式の編集](#)」

7.3.3.7 既存の @Prompt 式の編集

ユニバースはユニバースデザインツールで開きます。選択したオブジェクト式に、@Prompt 関数が含まれています。

オブジェクトまたは条件の定義または where 句にすでに存在している @Prompt 関数を編集します。編集は、@Prompt エディタを介して行います。

- 1 @Prompt 関数を右クリックし、ショートカットメニューから[プロンプトの編集]を選択します。
- 2 @Prompt エディタを使用して、プロンプト式を編集します。
- 3 [解析]をクリックして、@Prompt の構文を確認します。

関連項目

- ・ 391 ページの[@Prompt 関数の構文](#)」
- ・ 385 ページの[手動による SQL 文の @Prompt 関数の定義](#)」
- ・ 382 ページの[プロンプトに対する静的な値の一覧の定義](#)」

- ・ [383 ページの プロンプトに対する値の一覧としてのユニバースオブジェクトの選択](#)
- ・ [383 ページの プロンプトに対するカスタムの値の一覧の選択](#)
- ・ [383 ページの プロンプトに対する値の一覧としてのユニバースオブジェクトの選択](#)

7.3.3.8 手動による SQL 文の @Prompt 関数の定義

注

以前に @Prompt 関数を定義したことがない場合、@Prompt エディタの使用をお勧めします。@Prompt 関数の構文は複雑なため、構文の定義のページを参照してください。

オブジェクトの @Prompt 関数を手動で定義するには、次の手順に従います。

- 1 [プロパティの編集]ウィンドウを開き、ユニバース枠内のクラス、オブジェクト、または条件をダブルクリックするか、オブジェクトを右クリックして[オブジェクトのプロパティ]を選択するか、[編集] > [プロパティ]を選択します。
- 2 [定義]枠の[SELECT]ダイアログ内または[WHERE]ダイアログ内をクリックし、必要な構文に従って @Prompt 値を定義します(下のリンクを参照)。
- 3 [解析]をクリックして、SQL の構文を確認します。

関連項目

- ・ [391 ページの @Prompt 関数の構文](#)
- ・ [379 ページの @Prompt エディタ](#)
- ・ [384 ページの 既存の @Prompt 式の編集](#)
- ・ [382 ページの プロンプトに対する静的な値の一覧の定義](#)
- ・ [383 ページの プロンプトに対する値の一覧としてのユニバースオブジェクトの選択](#)

7.3.3.9 手動でプロンプトを定義するための @Prompt 式のプロパティ

@Prompt の構文は、次のとおりです。

```
@Prompt('message','type','lov',Mono|Multi,free|constrained|primary_key
,persistent|not_persistent,('default value'))
```

例は次のとおりです。

```
@Prompt('Displayed text ','A','Store¥City',Mono,constrained,Persistent,('Paris'))
```

@Prompt 式の値のプロパティについて、次の表で説明します。

プロパティ	説明
'message'	<p>必須</p> <p>メッセージボックスに表示するテキストを、'地域を選択してください'、'対象期間'、'店舗の指定'のように単一引用符で囲んで指定します。ユーザがクエリを実行すると、プロンプトボックスにテキストが表示されます。</p>
'type'	<p>必須ですが、空にしておくことができます。デフォルトでは、'A'に設定されています。</p> <p>3 番目のパラメータのデータタイプです。次の中から指定します。</p> <ul style="list-style-type: none">・ 'A' - 英数字・ 'N' - 数字・ 'D' - 日付 <p>データ型は、単一引用符で囲んで指定します</p> <p>値のペアを示すハードコード化された一覧を使用する場合、構文は 'value_type:key_type' となり、たとえば、'A:N' のようになります。ここで、最初の値はエンドユーザに表示されるキャプションであり、2 番目の値はクエリを速度を上げるためにクエリで実際に使用される一次キー値です。上記のとおり、各タイプ(キャプションおよび一次キー)は A、N、または D になります。例: 'A:A' または 'A:N'。この場合、次のパラメータ 'lov' には、パラメータのペアの一覧が含まれます。同様に、'default_value' パラメータには、値のペアが含まれています。一次キーを使用する際は、インデックス認識を設定する必要があります。</p>

プロパティ	説明
lov	<p>必須ですが、空にしておくこともできます。一覧が空の場合は、カンマが必須です。このパラメータがユニバース オブジェクトである場合、5 番目のパラメータ(selection mode= free constrained primary key)は primary_key とし、ユニバースでインデックス認識を設定する必要があります。</p> <p>次のどちらかの値の一覧を指定できます。</p> <ul style="list-style-type: none"> 既存ユニバース オブジェクトの値の一覧。[クラスとオブジェクト] パネルから目的のオブジェクトをダブルクリックして指定できます。クラスとオブジェクト間の区切り記号は「¥」です。全体を単一引用符で囲みます。例: '顧客¥出身国' キーを使用してオブジェクトのキー値を返す場合は、5 番目の値を primary_key に設定します。 ハードコーディングされた単一の値または値のペアの一覧。ペアの値は、コロンで区切ります。各値は、単一引用符で囲みます。値のペアは、カンマで区切ります。一覧全体は中括弧でくくります。制約を primary_key に設定します。 <p>単一デフォルト値の構文は次のようになります。</p> <p>{'値'}</p> <p>複数の単一デフォルト値の構文は次のようになります。</p> <p>{'値 1','値 2',...,'最後の値'}</p> <p>デフォルト値のペアを定義できます。</p> <p>デフォルト値のペアの構文は、{'値':'キー'} です。</p> <p>値とキーの区切り文字はコロン(:)です。</p> <p>デフォルト値のペアの構文は次のようになります。</p> <p>{'値 1':'キー 1','値 2':'キー 2',...,'最後の値':'最後のキー'}</p> <p>たとえば、{'Australia':'A','France':'F','Germany':'G','Japan':'J','Spain':'S','United Kingdom':'UK'} です。</p>
Mono Multi	<p>必須ですが、空にしておくことができます。デフォルトでは、Mono に設定されています。カンマは必須です。</p> <p>ユーザーが値の一覧から値を 1 つだけ選択できる場合、Mono を使用します。</p> <p>ユーザーが値の一覧から複数の値を選択できる場合、Multi を使用します。</p>

プロパティ	説明
free constrained primary_key	<p>必須ですが、空にしておくことができます。デフォルトでは、free に設定されています カンマは必須です。</p> <p>ユーザーが値を入力するか、値の一覧から選択できる場合、free を使用します。</p> <p>ユーザーが値の一覧から値を選択する必要がある場合、constrained を使用します。</p> <p>ユニバースでキーを設定した場合、primary_key パラメータを使用します。入力された値でも表示されている値でもなく、オブジェクトに関連付けられているキー値が使用されます。lov パラメータを使用する場合、primary_key は必須です。</p>
persistent not_persistent	<p>オプションの要素です。このパラメータを設定した場合、引数の末尾にはカンマを付けます。このパラメータを設定せず、7 番目のパラメータ (デフォルト値) を設定している場合でも、カンマを設定する必要があります。</p> <p>注 このパラメータが Desktop Intelligence に影響を及ぼすことはありません。</p> <p>ドキュメントを最新表示すると、メッセージボックスで最後に使用された値がデフォルトで表示される場合には、デフォルト値が定義されていても persistent を使用します。</p> <p>ドキュメントを最新表示すると、メッセージボックスで使用された値がデフォルトで表示されない場合、not_persistent を使用します。</p>

プロパティ	説明
'default value'	<p>オプションの要素です。'default value' パラメータを使用して、ユーザーに表示されるデフォルト値を定義します。ハードコーディングされた一覧を使用する場合、ここで入力されるデフォルト値は、[lov] の一覧に含まれている必要があります。</p> <p>たとえば、単一の値は次のようになります。</p> <pre>{'France'}</pre> <p>値のペアは次のようになります。</p> <pre>{'France':'F'}</pre> <p>2 つの値のペアの場合は次のようになります。</p> <pre>{'France':'F','Germany':'G'}</pre> <p>ドキュメントを最新表示すると、これらの値がデフォルトで表示されますが、persistent オプションが設定されている場合は、デフォルト値ではなくメッセージボックスで最後に使用された値が使用されます。</p> <p>単一の値または値のペアを設定できます。</p> <p>メッセージボックスの定義に primary_key パラメータを指定した場合は、キー値を入力する必要があります。</p>

7.3.3.10 @Prompt 関数の使用例

次に、@Prompt 構文の例を示します。

@Prompt 関数の最小使用例:

```
@Prompt('Displayed text ','A',,,)
```

デフォルト値を持たない LOV を使用した @Prompt の使用例:

```
@Prompt('Displayed text ','A',{'Paris','London','Madrid'},,,)
```

LOV と 1 つのデフォルト値を使用した @Prompt の使用例:

```
@Prompt('Displayed text ','A',{'Paris','London','Madrid'},,,,'Paris')
```

オブジェクトとデフォルト値を使用した @Prompt の使用例:

```
@Prompt('Displayed text ','A','StoreCity',,,,'Paris')
```

すべての使用可能な設定での @Prompt の使用例:

```
@Prompt('Displayed text ','A','Store¥City',Mono,Constrained,Persistent,{'Paris'})
```

値のペアを含むが、デフォルト値を持たない LOV を使用した @Prompt の使用例:

```
@Prompt('Displayed text ','A:N',{'Paris':'12','London':'7','Madrid':'15'},,)
```

値のペアと 1 つのデフォルト値のペアを含む LOV を使用した @Prompt の使用例:

```
@Prompt('Displayed text ','A:N',{'Paris':'12','London':'7','Madrid':'15'},,{'Paris':'12'})
```

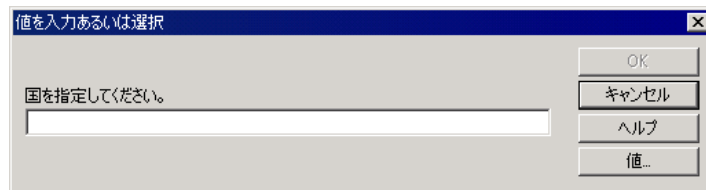
例 @Prompt を使って、デフォルト値を持たない[国]の値を制限する

[国]オブジェクトは、リゾートのある国を返します。返される値を 1 つの国だけに限定するため、それぞれのリゾート国に対応したオブジェクトをユニバースに追加することもできますが、@Prompt を使えば次のオブジェクトだけでこの要求を満足できます。

```
Country.country=@prompt('Choose a country',
'A','Customer¥Country of origin',Mono,primary_key,,)
```

このオブジェクトを含んだクエリーを実行すると、ユーザーにメッセージボックスが表示され、そこで指定された国のデータだけが返されます。

Web Intelligence でクエリーを実行すると、次のプロンプトボックスが表示されます。



例 @Prompt 構文とデフォルト値

```
@prompt('Enter value(s) for Customer with IA:',
'A','Customer¥Customer with I A',
Multi,primary_key,,{'Baker','Brendt'})
```

例 ハードコーディングされた値の一覧を使用する @Prompt 構文

次の例では、国の一覧とデフォルト値を表示する方法を示します。オブジェクトがキーで、制約が primary_key に設定されている場合、デフォルト値は {'England: 21', 'Scotland:39'} など一対のセット(値, キー)で表すことができます。一地域のみを選択して単一のパラメータを設定する必要があります。デフォルト値は、値の一覧に表示される必要があります。

```
SELECT dbo.region.sales_region
FROM dbo.region
WHERE dbo.region.region_id = @Prompt('Choose a region','A:N',
{'England':'21','Scotland':'39','Wales':'14'},
Mono, primary_key, Persistent, {'Scotland':'39'})
```

この機能を使用すると、CASE WHEN ELSE 句がサポートされていないデータベース(特に OLAP データベース)で CASE WHEN ELSE 句の動作を実行することもできます。

例 @Prompt 構文を使って、パターンと一致プロンプトで事前定義条件を作成する

次の例では、クライアント名の最初の文字を入力することによってクライアント名を選択できます。Web Intelligence ユーザーが「H%」と入力すると、姓が H で始まるすべてのクライアントがレポートに返されます。

```
(@Select(Client¥Client Name)
  LIKE (@Prompt('enter','A',,,)+%))
```

クライアントに大文字または小文字のいずれかを使用できるようにするには、次の構文を使用します。

```
(@Select(Client¥Client Name)
  LIKE lower(@Prompt('enter','A',,,)+%) OR
  (@Select(Client¥Client Name)
  LIKE upper(@Prompt('enter','A',,,)+%))
```

7.3.3.11 @Prompt 関数の構文

@Prompt 関数には反転性があるため、構文は複雑です。プロンプトメッセージを作成し、データタイプ、データが単一値か複数值か、およびデータが永続的であるかどうかを指定します。また、デフォルトも指定できます。以下に構文を示します。

```
@Prompt('message','type',[lov],Mono|Multi,
free|constrained|primary_key,persistent|not_persistent,[default_values])
```

関連項目

- ・ 378 ページの [@Prompt](#)」
- ・ 385 ページの [手動による SQL 文の @Prompt 関数の定義](#)」
- ・ 385 ページの [手動でプロンプトを定義するための @Prompt 式のプロパティ](#)」

7.3.4 @Script

@Script 関数は、VBA(Visual Basic for Applications)マクロの結果を返します。VBA マクロは Windows 環境でしか実行できません。@Script 関数を使用すると、オブジェクトを含んだクエリを最新表示または実行するたびに、指定した VBA マクロが実行されます。

@Script 関数は、プロンプトボックス(@Prompt 関数)よりも複雑な処理が要求される場合に、オブジェクトの WHERE 句に指定します。VBA マクロは、BusinessObjects レポートファイル(.REP)に保存されます。レポート

ファイルのデフォルトの保存場所は、BusinessObjects パスの UserDocs フォルダです。.REP ファイルの保存場所は、自由に変更できます。

注

@Script は、ユニバースデザインツールと、Desktop Intelligence または Desktop Intelligence 3-tier モードのクライアントバージョンでのみサポートされます。Desktop Intelligence のクライアントバージョン以外では、@Script 関数を使用しないことをお勧めします。この関数は、Desktop Intelligence レポートの公開やスケジュール用に InfoView で使用できる Desktop Intelligence のサーバーバージョン、および Web Intelligence ではサポートされていません。Web Intelligence の場合、対話型オブジェクトには @Script 関数を使用せず、@Prompt 関数を使ったシンプルなデザインを心がけてください。

7.3.4.1 @Script 関数の構文

@Script 関数の構文は、次のとおりです。

```
@Script('var_name', ['var_type'], 'script_name')
```

注

2番目の引数は省略できますが、引数を指定しない場合でも区切り記号のカンマを入力する必要があります。パラメータの内容は、次の通りです。

構文	説明
'var_name'	マクロ内に宣言された変数名: この名前で実行したマクロの結果を、オブジェクトの SQL 定義内に取り込みますので、VBAマクロとオブジェクトのSQL定義で同じ変数名を指定する必要があります。
'var_type'	このパラメータはオプションです。関数から返されるデータタイプを示します。次の中から指定します。 <ul style="list-style-type: none"> ・ 'A' - 英数字 ・ 'N' - 数字 ・ 'D' - 日付 データ型は、単一引用符で囲んで指定します
'script_name'	実行する VBA マクロ名

7.3.5 @Select

@Select関数を使うと、既存のオブジェクト定義に使用した SELECT 句を再利用できます。オブジェクトの SELECT 句に @Select 関数を使用する場合は、@Select 関数のパラメータとしてユニバース内の他のオブジェクトを “クラス名¥オブジェクト名” の形式で指定します。関数が、参照するオブジェクトの SELECT 句へのポインタの役目を果たします。

@Select 関数を使用した場合、次のような利点があります。

- ・ SQL コードのインスタンスを作成せずに済む。
- ・ コードの整合性を確保できる。

注

@Select 関数や @Where 関数を使うと、そのオブジェクトはユニバース内の他のオブジェクトに依存することになり、オブジェクトに依存関係を設定したことになります。参照するオブジェクトを削除した場合は、@Select 関数または @Where 関数を使った定義を手動で更新する必要があります。

7.3.5.1 @Select 関数の構文

@Select 関数の構文は、次のとおりです。

```
@Select(Classname¥Objectname)
```

構文	説明
Classname	参照オブジェクトのあるクラスの名前。
Objectname	参照するオブジェクトの名前。

7.3.5.2 @Select 関数の使用例

例 @Select 関数を使った Service_line Select 句の再利用

各リポートで行われたキャンペーンの対象となったサービス部門を取得するため、[キャンペーン]クラスを作成し、そこに[キャンペーンサービスライン]というオブジェクトを作成することになります。このオブジェクトには、@Select関数を使って既存の[サービスライン]のSELECT句を再利用できます。このオブジェクトには、@Select関数を使って Service_Line オブジェクトの既存の SELECT 句を再利用できます。

キャンペーンサービスラインの SELECT 句が次のように表示されます。

名前(N): キャンペーン サービスライン 型(T): 文字型

説明(D):

SELECT 句(S): @Select(リポート*サービスライン)

WHERE 句(W):

テーブル(B)... 分析(P)

7.3.6 @Variable

@Variable 関数は、WHERE 句の中などで次のいずれかの種類の変数に割り当てられた値を参照するために使用します。

- ・ BusinessObjects システム変数
- ・ レポート変数
- ・ 言語(ロケール)変数
- ・ オペレーティングシステム変数
- ・ Desktop Intelligence のカスタム変数

多くの場合、@Variable 関数は、[(オブジェクト名)のプロパティを編集]ダイアログボックスの[定義]ページにある[WHERE 句]ボックスの条件のオペランド側(右側)に指定します。このクエリによって、変数の値が取得されます。

注

- ・ @Variable は、単一値関数であり、IN または INLIST 演算子と一緒に使用できません。
- ・ 1つのクエリで同じ @Variable 関数を複数回実行すると、プロンプトは 1 度だけ表示されます。
- ・ @Variable 関数は、次のように設定された単一値の @Prompt 関数と同等です。

```
@Prompt('Question','A',,mono,free)
```

@Prompt 関数が単一値である場合、同一のクエリ内で @Variable 関数と @Prompt 関数を統合できます。

関連項目

- ・ 395 ページの[@Variable 関数の構文](#)」
- ・ 395 ページの[@Variable プロパティの説明](#)」

7.3.6.1 @Variable 関数の構文

@Variable 関数の構文は、次のとおりです。

```
@Variable(<VariableName>')
```

注

名前は、必ず単一引用符で囲みます。

例 BOUSER 値を返す @Variable 構文

```
@Variable('BOUSER')
```

関連項目

- ・ 394 ページの[@Variable](#)」
- ・ 395 ページの[@Variable プロパティの説明](#)」

7.3.6.2 @Variable プロパティの説明

いずれの場合も、変数名は必ず単一引用符で囲みます。

変数名	説明
BusinessObjects システム変数 <ul style="list-style-type: none">・ BOUSER: ユーザログイン・ DBUSER: データベースユーザ名・ DBPASS: データベースユーザのパスワード	BusinessObjects システム変数の値。対応する BusinessObjects ユーザのログインに従い、返されるデータが変わります。 宣言された BusinessObjects データベースユーザの値。

変数名	説明
レポート変数 <ul style="list-style-type: none"> ・ DOCNAME: ドキュメント名 ・ DPNAME: データプロバイダの名前 ・ DPTYPE: データプロバイダの種類 ・ UNVNAME: ユニバース名 ・ UNVID: 使用されたユニバースの ID 	これらの変数は、たとえば SELECT 句の前に実行される Begin_SQL パラメータで参照できます。これは、最も頻繁に使用されるレポートクエリまたはユニバースの確認など、データベースの使用に関する監査目的で使用できます。
言語変数 <ul style="list-style-type: none"> ・ PREFERRED_VIEWING_LOCALE ・ DOMINANT_PREFERRED_VIEWING_LOCALE 	言語変数 <ul style="list-style-type: none"> ・ PREFERRED_VIEWING_LOCALE: ユーザの優先表示ロケール。これは、ユニバースメタデータを表示するためにユーザが選択するロケールと同じです。 ・ DOMINANT_PREFERRED_VIEWING_LOCALE: ユーザの優先表示ロケールの優先ロケール。この変数は、ユーザがデータをすべてのロケール (fr_FR、fr_BE、fr_CA など) で翻訳することを防止します。fr_FR の翻訳を使用でき、ユーザロケールが fr_BE または fr_CA の場合には、同じ優先ロケールが共有されるので、fr_FR の翻訳を再使用できます。
オペレーティングシステム変数	インストール環境に関する情報を取得するために、Windows 環境変数を入力できます。
カスタム変数	Desktop Intelligence では、事前定義されたテキストファイルを使用して固定された変数の一覧を提供できます。

関連項目

- ・ 394 ページの[@Variable](#)」
- ・ 395 ページの[@Variable 関数の構文](#)」

7.3.6.3 BusinessObjectsシステム変数の使い方

@Variable 関数を使って BusinessObjects システム変数を参照し、ログインする BusinessObjects ユーザのプロファイルを使ってデータを制限することができます。

注

BusinessObjects のログインパラメータとデータベースのログインパラメータは同じでなければなりません。

それぞれの BusinessObjects ユーザに割り当てられるユーザ名は、次の BusinessObjects システム変数に保存されます。

- ・ BOUSER: ユーザ名

この変数は、Business Objects 製品にログインする際にユーザに表示される[ユーザ確認]ダイアログボックスのフィールドに対応するものです。

@Variable 関数を WHERE 句に定義すると、クエリにそのオブジェクトが含まれた場合、ユーザのデータベースプロファイルに従ってデータへのアクセスが制限されます。

@Variable 関数は、[(オブジェクト名)のプロパティを編集]ダイアログボックスの[定義]ページにある[WHERE 句]ボックスの条件のオペランド側(右側)に指定します。

例

@Variable

人事課のデータベースから構築されたユニバースに[従業員氏名]というオブジェクトがあるとします。このオブジェクトが返すデータを、各従業員がデータベースで許可された値だけに制限することになります。このオブジェクトが返すデータを、各従業員がデータベースで許可された値だけに制限することになります。それぞれのユーザがアクセスできる従業員情報はデータベースプロファイルに定義されていますので、取得できるデータを従業員ごとに制御することができます。

次のように、@Variable 関数を WHERE 句に挿入します。

```
Employees.Employee_Name = @Variable('BOUSER')
```

“従業員氏名”を含むクエリを実行すると、BOUSER の値に一致するデータだけが返されます。

関連項目

- ・ 394 ページの[@Variable](#)」
- ・ 395 ページの[@Variable 関数の構文](#)」
- ・ 395 ページの[@Variable プロパティの説明](#)」

7.3.6.4 ロケール変数の使用

Web Intelligence によってレポートが取得され、適切なロケールで情報が表示されるように、@Variable 関数のロケール変数を使用してロケール設定を定義します。データベース テーブルには、データの翻訳を保持する行の言語を定義する列が必要です。ロケールは、言語と地理的地域、データの並べ替え方法、日付の書式、およびその他の特定の書式を定義します。@Variable 関数は、オブジェクトの WHERE 句に定義します。この

関数は、そのオブジェクトをクエリで使用する際に、ロケールを選択するようにユーザーに要求します。ユーザーがクエリを実行すると、ロケールの入力を要求するプロンプトボックスが表示されます。トランスレーションマネジメントツールのガイドに、ロケールコードと優先ロケールコードの一覧が記載されています。

定義できる設定は、次のとおりです。

- ・ @Variable('PREFERRED_VIEWING_LOCALE')
- ・ @Variable('DOMINANT_PREFERRED_VIEWING_LOCALE')

例

次の PRODUCT テーブルは、多数の言語に翻訳されています。ユーザーは、特定のロケールで製品名が表示されることを求めます。

表 7-13: PRODUCT テーブル

製品 ID	ロケール	製品名
DC1212	en_GB	Digital camera
DC1212	fr_FR	Appareil photo numérique
DC1212	de_DE	Digitalkamera
DC1212	es_ES	Cámaras digitales
...

```
SELECT Product_Name
```

```
FROM PRODUCT
```

```
WHERE PRODUCT.LOCALE = @Variable('PREFERRED_VIEWING_LOCALE')
```

クエリの実行時には、ユーザーが変数を適切なロケールに置き換えます。すると、Web Intelligence でそのロケールの情報が取得されます。

関連項目

- ・ [394 ページの@Variable」](#)
- ・ [395 ページの@Variable 関数の構文」](#)
- ・ [395 ページの@Variable プロパティの説明」](#)

7.3.6.5 レポート変数の使用

@Variable 関数をオブジェクトの WHERE 句で使うことで、リクエストにレポート変数を含めることができます。

これらの変数は、SELECT 句の前に実行される Begin_SQL で参照できます。これは、最も頻繁に使用されるレポートクエリまたはユニバースの確認など、データベースの使用に関する監査目的で使用できます。

変数は、以下で参照できます。

- ・ SELECT 句、WHERE 句などのオブジェクトの定義
- ・ フィルタ
- ・ 結合式
- ・ Begin_SQL パラメータ

関連項目

- ・ 394 ページの[@Variable](#)」
- ・ 395 ページの[@Variable 関数の構文](#)」
- ・ 395 ページの[@Variable プロパティの説明](#)」

7.3.6.6 オペレーティングシステム変数の使い方

インストール環境に関する情報を取得するために、Windows 環境変数を入力できます。たとえば、NUMBER_OF_PROCESSORS、USERNAME などです。

例

次の @Variable(NUMBER_OF_PROCESSORS) をクエリ含めると、使用マシンのプロセッサ数が応答に含まれます。

関連項目

- ・ 394 ページの[@Variable](#)」
- ・ 395 ページの[@Variable 関数の構文](#)」
- ・ 395 ページの[@Variable プロパティの説明](#)」

7.3.6.7 Desktop Intelligence でのカスタム変数の使い方

Desktop Intelligence では、オブジェクトの WHERE 句で @Variable 関数を使用して、指定したテキストファイル内の変数を参照することができます。これを利用して、オブジェクトの特定の条件を定義することができます。

この変数を使うには、コマンドラインから -vars パラメータを指定して Business Objects を起動する必要があります。この機能を使うすべてのマシンで、Windows ショートカットのコマンドラインを変更してください。

注

BusinessObjects をコマンドラインから起動しなければならないため、ユーザ数の多い環境では @Variable 関数のメンテナンスが難しくなります。ユーザ数が多い場合や、地理的に分散している場合は、@Variable 関数によるテキストファイル参照を使ったデータの制限は使用しないでください。

@Variable 関数を使ってテキストファイル変数を参照することの長所

@Variable 関数を使ってテキストファイル変数を参照することの主な利点は、ユニバースを変更せずに、テキストファイルで変数の値を更新できることです。

@Variable 関数を使ってテキストファイル変数を参照することの短所

- ・ すべてのクライアントのポストで、コマンドラインに `-vars <テキストファイル名.txt>` 引数を追加する必要があります。
- ・ テキストファイルへのアクセスと変更が比較的簡単に行えますので、セキュリティ上の問題があります。

このように @Variable 関数を使ったテキストファイル変数の参照には潜在的な問題が多く含まれていますので、Business Objects 製品を業務で使用する場合は、Supervisor のセキュリティオプションを使ってデータへのアクセスを制限してください。

関連項目

- ・ 400 ページの[Desktop Intelligence でのカスタム変数の使い方](#)」
- ・ 394 ページの[@Variable](#)」
- ・ 395 ページの[@Variable 関数の構文](#)」
- ・ 395 ページの[@Variable プロパティの説明](#)」

7.3.6.7.1 Desktop Intelligence でのカスタム変数の使い方

次の手順に従って 1 つまたは複数の定義済み変数値を使用します。

- 1 一連の変数とその値を定義したテキストファイルを作成します。書式は “変数名 = 値” です。
- 2 BusinessObjects の起動に使用するコマンドラインに `Busobj.exe -vars <textfile.txt>` を追加します。たとえば、Bovars.txt というファイルがある場合は、`C:\¥BusinessObjects¥Busobj.exe -vars Bovars.txt` と入力します。-vars 構文は、BusinessObjects で使用するテキストファイルをメモリに読み込むよう、オペレーティングシステムに指示するスイッチです。
- 3 テキスト変数の参照を設定するオブジェクトをダブルクリックし、[(オブジェクト名の)プロパティ編集]ダイアログボックスを開きます。
- 4 WHERE 句のオペランド側(右側)に @Variable 関数を入します。例: `COUNTRY.COUNTRY_NAME = @Variable('国')Country` は、テキストファイル内に記録されている変数名です。この変数名は、引用符で囲む必要があります。
- 5 [OK]をクリックし、ユニバースを保存します。

7.3.7 @Where

@Where 関数を使うと、既存のオブジェクト定義に使用した WHERE 句を再利用できます。オブジェクトの WHERE 句に @Where 関数を使用する場合は、@Where 関数のパラメータとしてユニバース内の他のオブジェクトを“クラス名¥オブジェクト名”の形式で指定します。関数が、参照するオブジェクトの WHERE 句へのポイントの役目を果たします。

@Where 関数を使うとオブジェクト間に動的なリンクが作成されます。参照元のオブジェクトの WHERE 句が変更されると、自動的に参照先の WHERE 句も変更されます。

@Where 関数では既存のコードを利用できるため、次の利点があります。

- ・ SQL コードのインスタンスを作成せずに済む。
- ・ コードの整合性を確保できる。

@Select 関数や @Where 関数を使うと、そのオブジェクトはユニバース内のほかのオブジェクトに依存することになり、オブジェクトに依存関係を設定したことになります。参照するオブジェクトを削除した場合は、@Select 関数または @Where 関数を使った定義を手動で更新する必要があります。

7.3.7.1 @Where 関数の構文

この関数の構文は、次の通りです。

```
@Where(Classname¥Objectname)
```

構文	説明
Classname	クラスの名前。
Objectname	参照するオブジェクトの名前。

7.3.7.2 @Where 関数を使用した WHERE 句 の再利用例

例 @Where 関数を使った“リゾート”の WHERE 句の再利用

各リゾートで利用できるサービス部門を返す[リゾートのサービス部門]オブジェクトを作成することにします。“リゾート”オブジェクトに定義した @Prompt 関数を再利用して、クエリの実行時にどのリゾートのサービス部門のデータを取得するかをユーザに問い合わせます。

[リゾート]オブジェクト(参照するオブジェクト)のSQLは、次のように定義されています。

名前(N): 型(T):

説明(D):
リゾートの名前

SELECT 句(S):
Resortresort

WHERE 句(W):
Resortresort = @Prompt(リゾート名を入力してください。','A','リゾート\\リゾート','MONO')

テーブル(B)... 分析(P)

次のように、新しく作成する“リゾートのサービス部門”オブジェクトでは WHERE 句に @Prompt 関数を使って“リゾート”のプロンプトを参照しています。

SELECT 句(S):
Resortresort & ' の ' & Service_Line.service_line

WHERE 句(W):
@Where(リゾート\\リゾート)

テーブル(B)... 分析(P)

ユーザが[リゾートのサービス部門]オブジェクトを使ってクエリを実行すると、リゾート名を入力するよう要求するプロンプトが表示されます。“リゾート”の WHERE 句を変更すると、“リゾートのサービス部門”にもその変更が自動的に反映されます。

7.4 外部ストラテジーを使用したユニバース作成のカスタマイズ

ユニバースデザインツールでは、組み込みの自動ルーチンを使用して、データベースの構造に基づきユニバースコンポーネントが自動的に作成されます。これらのルーチンはストラテジーと呼ばれ、[パラメータ]ダイアログボックスの[ストラテジー]ページ([ファイル] > [パラメータ] > [ストラテジー])で利用できます。これらのストラテジーは、ユニバースデザインツールに組み込まれています。アクセスしたり変更したりすることはできません。ストラテジーの使用および有効化についての詳細は、92 ページの「[ストラテジーの選択](#)」を参照してください。

また、定義された出力構造に従った SQL スクリプトを作成して、カスタマイズされたユニバース自動作成タスクを実行することもできます。これらのスクリプトは、その他のストラテジーとともに[ストラテジー]ページで選択できます。ユーザにより定義およびカスタマイズされたこれらのスクリプトを外部ストラテジーと呼びます。

この節では、外部ストラテジーとその使用について説明します。

7.4.1 ユニバースデザインツールへの外部ストラテジーの移行

Universe Designer 6.5 より前のバージョンのユニバースデザインツールの外部ストラテジーは `st<xxxx>.txt` ファイルという外部テキストファイルで定義されていました。このファイルは、現在はサポートされていません。

注

Universe Designer 6.5 から移行する場合、外部ストラテジーは、ユニバースデザインツールでの処理と同様に処理されます。

旧バージョンでユーザによりカスタマイズおよび定義された外部ストラテジーをユニバースデザインツールで利用できるようにするには、以下の操作を実行します。

- ・ 新しい外部ストラテジーファイル(<RDBMS>.STG)を次のように編集します。
 - ・ ターゲットの RDBMS の外部ストラテジーファイルを XML エディタで開きます。
 - ・ 各ストラテジーの新しいエントリを作成します。
 - ・ 各ストラテジーで、SQL タグを使用して STG ファイルに SQL スクリプトを直接コピーします。

または

- ・ FILE タグを使用して、外部テキストファイルのデータを参照するファイルパスを入力します。

これら 2 つの方法についての説明は、414 ページの「[外部ストラテジーの作成](#)」をご覧ください。

- ・ 2 番目の XML ファイル(<RDBMS><言語>.STG)に Help テキストをコピーします。詳細については、405 ページの「[外部ストラテジーの Help テキストの作成](#)」を参照してください。
- ・ 外部ストラテジーファイルの宣言が、一般パラメータファイル(SBO)にあることを確認してください。旧バージョンの Universe Designer のパラメータファイル(PRM)ではないことに注意してください。詳細については、407 ページの「[外部ストラテジーファイルの宣言の確認](#)」を参照してください。

7.4.2 外部ストラテジーの概要

以下の表で、外部ストラテジーを作成および管理する際に使用されるファイルとその役割について概説します。

外部ストラテジー管理プロセスでの役割およびファイル	説明
外部ストラテジーファイルに作成および格納される外部ストラテジー(<RDBMS>.STG)	<p>XML ファイルに、外部ストラテジー名、種類、SQL スクリプト、またはデータの入った外部テキストファイルへの参照が含まれます。ファイルはここに保存されます。\$INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/<RDBMS>.stg 各 RDBMS に 1 つのファイルがあります。strategy.dtd ファイルはここで使用します。\$INSTALLDIR/dataAccess/RDBMS/connectionServer/strategy.dtd 関連トピック:</p> <ul style="list-style-type: none"> 408 ページの「ストラテジーファイル(STG)の構造」 414 ページの「外部ストラテジーの作成」
外部ストラテジー言語ファイルで作成および格納される外部ストラテジーの Help テキスト (<RDBMS><言語>.STG)	<p>XML ファイルに、外部ストラテジーファイルにある各外部ストラテジーの Help テキストが含まれます。これは、[ストラテジー]ページで外部ストラテジーを選択するとその下に表示されるテキストです。ファイルはここに保存されます。\$INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/<RDBMS><language>.stg strategy_localization.dtd ファイルはここに保存されます。\$INSTALLDIR/dataAccess/RDBMS/connectionServer/strategy_localization.dtd 関連トピック: 405 ページの「外部ストラテジーの Help テキストの作成」</p>
ターゲット RDBMS の汎用データアクセスファイル (SBO)で宣言される外部ストラテジーファイル	<p>XML ファイルに、ターゲット RDBMS の汎用データアクセスパラメータが含まれます。この外部ストラテジーファイルの名前は、デフォルトでパラメータ External Strategies の値として設定されます。関連トピック: 407 ページの「外部ストラテジーファイルの宣言の確認」</p>

7.4.3 外部ストラテジーの概要

外部ストラテジーとは、.UNV ファイルの外部に保存される SQL スクリプトです。これは、ユニバースデザインツールで、ユニバースでのオブジェクトまたは結合の作成やテーブル検出タスクを自動化できるよう構築されています。外部ストラテジーは、拡張子 STG の付いた外部ストラテジーファイルに保存されます。外部ストラテジーファイルは、XML 形式です。サポートされる各 RDBMS に対して 1 つのファイルがあります。

外部ストラテジーファイルは次のディレクトリに格納されます。

```
$INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/<rdbms>.stg
```

注

外部ストラテジーファイルは、XML エディタを使用して編集する必要があります。

7.4.3.1 外部ストラテジーへのアクセス

外部ストラテジーは、[ストラテジー] ページのビルトインストラテジーを一覧表示するドロップダウンリストボックスに表示されます。各ドロップダウンリストボックスは、XML ファイルのストラテジーの種類のカテゴリに対応します。外部ストラテジーは、次のように External Strategy というプレフィクスにその名前が付いてリストに表示されます。

```
External Strategy:<ストラテジー名>
```

たとえば、ストラテジーファイル内で Constraints という名前の結合作成用の外部ストラテジーは、[ストラテジー] ページの[結合]ドロップダウンリストに External Strategy:Constraints と表示されます。

7.4.4 外部ストラテジーの Help テキストの作成

[ストラテジー] ページで、選択したストラテジーの下にメモが表示されます。これがストラテジーの Help テキストです。ビルトインストラテジーの場合、Help テキストへのアクセスまたは編集はできません。外部ストラテジーの場合、Help テキストにアクセスして編集できます。

注

旧バージョンのユニバースデザインツールでは、Help テキストはストラテジーファイルの[HELP]セクションに含まれていました。このバージョンでは、このテキストが別のファイルに格納されるようになりました。次に、このファイル、外部ストラテジー言語ファイルについて説明します。

7.4.4.1 別のファイルに格納される外部ストラテジー Help テキスト

外部ストラテジーの Help テキストは、<RDBMS><言語>.stg という別の外部ストラテジー言語ファイルに格納されています。たとえば、oracleen.stg は、oracle.stg ファイルのストラテジー用の Help テキストファイルです。

Help テキストのエントリは、編集およびカスタマイズできます。Help テキストには、ストラテジーについてよく知らないデザイナーを支援するため、そのストラテジーの動作について簡単な説明があります。

外部ストラテジー言語ファイルには、外部ストラテジーファイルに表示される個々の外部ストラテジーに対応する Help テキストのエントリを作成するようにします。

インストールした ユニバースデザインツールの各言語バージョンにストラテジー言語ファイルがあります。外部ストラテジー言語ファイルは、外部ストラテジーファイルを同じディレクトリにあります。たとえば、ユニバースデザインツールのフランス語バージョンがインストールされている場合、Oracle の外部ストラテジー言語ファイルは oraclefr.stg です。英語版は oracleen.stg です。

新しい外部ストラテジーを外部ストラテジーファイルに作成する場合、外部ストラテジー言語ファイルに Help テキストも作成します。これにより、ユニバースを使用するほかのデザイナーに外部ストラテジーに関する情報を提供します。

例 Oracle データアクセスドライバで出荷されたストラテジーの Help テキストエントリ

oracleja.stg ファイルに表示されるクラスとオブジェクトストラテジーの Help テキストは以下のとおりです。これは、oracle.stg ファイルに定義された、クラスとストラテジー外部ストラテジーの Help テキストです。

```
<Strategy Name="Classes_and_Objects">
  <Message id="Help">This strategy reads the database structure. It associates tables with classes, and columns with objects.</Message>
  <Message id="Name">External Strategy: Classes and Objects</Message>
```

7.4.4.2 外部ストラテジーの Help エントリの作成

外部ストラテジーの Help エントリを作成する手順は、次の通りです。

- 1 ターゲットの RDBMS の外部ストラテジー言語ファイルを XML エディタで開きます。ターゲットの RDBMS の外部ストラテジー言語ファイルの保存場所は次の通りです。

```
$INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/<RDBMS><言語>.stg.
```

例:

```
$INSTALLDIR/dataAccess/RDBMS/connectionServer/oracle/oracleja.stg.
```

- 2 新しい Name 要素を作成します。
- 3 ストラテジーの名前を入力します。これは、Help テキストを作成するストラテジーの名前です。
- 4 Help という Message id を作成します。このタグに、Help テキストが含まれます。
- 5 Help テキストを入力します。
- 6 Name という Message id を作成します。このタグには、外部ストラテジーを選択したときにストラテジーのドロップダウンリストに表示する名前が含まれます。
- 7 ストラテジーの名前を入力します。

確認してから、保存してファイルを閉じます。

次にユニバースデザインツールを起動したときに、選択した外部ストラテジーの下に Help テキストが表示されます。

ヒント

新しい Name 要素のパラメータを簡単に作成および設定するには、既存の Name 要素をコピーして、新しいストラテジーの新しい値を入力する方法もあります。

7.4.5 外部ストラテジーファイルの宣言の確認

外部ストラテジーファイルは、ターゲット EDBMS の一般パラメータ (SBO) ファイルの Parameter セクションで宣言します。たとえば、Oracle の外部ストラテジーファイルは oracle.stg です。これには、次のように oracle.sbo ファイルに oracle という値が入っています。

Parameter (10)		
	Name	Text
1	Family	Oracle
2	SQL External File	oracle
3	SQL Parameter File	oracle
4	Description File	oracle
5	Strategies File	oracle
6	Driver Level	31
7	Array Fetch Available	True
8	Array Bind Available	True
9	Binary Slice Size	32000
10	CharSet Table	oracle

oracle は Oracle 用の外部ストラテジーファイルの名前です。この名前は oracle.sbo ファイルで宣言されます。

7.4.5.1 ストラテジーファイルでの SBO ファイル宣言の確認

外部ストラテジーファイルが正しく宣言されていることを確認する手順は、次の通りです。

- 1 ターゲットの RDBMS の SBO ファイルを開きます。
- 2 パラメータ Strategies Name が外部ストラテジーファイルの名前に設定されていることを確認します。これは、デフォルト設定です。
- 3 名前が正しく設定されていない場合、外部ストラテジーファイルの名前を正しく入力します。
- 4 変更箇所がある場合、保存してからファイルを閉じます。

または

- 5 何も変更していない場合、保存せずにファイルを閉じます。

注

旧バージョンのユニバースデザインツールの外部ストラテジーは、PRM ファイルに宣言されています。Universe Designer 6.5 ではこれが変更されました。SBO ファイルのパラメータ Strategies File は、デフォルトでターゲット RDBMS の外部ストラテジーファイルの名前に設定されます。外部ストラテジーの Universe Designer 6.5 への移行に関する詳細は、404 ページの「[外部ストラテジーの概要](#)」を参照してください。

7.4.6 サンプルの外部ストラテジーの使用

ストラテジーファイルには、Business Objects製品が提供する既存のストラテジーが含まれています。外部ストラテジーファイルには、オブジェクト作成ストラテジー、結合作成ストラテジー、テーブル参照ストラテジーを1つずつ指定することもできますし、複数のストラテジーを準備しておくこともできます。

サンプルファイルは、カスタマイズしたり、新しい外部ストラテジーのベースとしてそのまま使用したりできます。既存のストラテジーをカスタマイズすることも、独自のストラテジーを作成することもできます。

変更する前に、各ファイルのコピーを保存します。

7.4.7 ストラテジーファイル(STG)の構造

サポートされる各 RDBMS に、XML 形式の外部ストラテジーファイル(STG)が用意されています。このファイルに、既存の外部ストラテジーを移行するか、新しい外部ストラテジーを作成します。すべての外部ストラテジーファイルは、次のディレクトリにあるストラテジー DTD ファイル(<RDBMS>.dtd)を使用します。

`$INSTALLDIR/dataAccess/RDBMS/connectionServer`

外部ストラテジー XML ファイルの要素は、この外部ストラテジー DTD ファイルに定義されています。XML SPY のような特定の XML エディタを使用すると、新しいストラテジー要素を作成するとき、ドロップダウンリストに利用可能なパラメータが一覧表示されます。

外部ストラテジーファイルには、Strategies というメインのセクションがあります。すべての外部ストラテジーがこの節に定義されます。Strategies セクションには、次の要素とパラメータがあります。

ファイル要素	説明
Strategy	メインの要素です。すべての外部ストラテジーがこの要素内に作成されます。
Name	外部ストラテジーの名前です。この名前が、[ストラテジー]ページのドロップダウンリストに表示されます。デフォルトの要素です。

ファイル要素	説明
Type	<p>外部ストラテジーが表示される[ストラテジー]ページのリストです。3つの値があります。</p> <ul style="list-style-type: none"> ・ JOIN: 結合ストラテジーが[結合]リストに表示されます。 ・ OBJECT: クラスとオブジェクトストラテジーが[クラスとオブジェクト]リストに表示されます。 ・ STRUCT: テーブル検出ストラテジーが[テーブル]リストに表示されます。
SQL	<p>スクリプトの SQL コードです。これは、このストラテジーを選択したときにユニバースデザインツールで実行される SQL スクリプトです。この SQL スクリプトは、オブジェクトと結合の作成ルーチンやテーブル検出のルーチンを正しく実行するために、特定の出力フォーマットに従う必要があります。外部ストラテジーの SQL を作成する方法についての詳細は、411 ページの「オブジェクトストラテジー (OBJECT) の出力書式」を参照してください。</p>
Connection	<p>データベース接続を指定します。接続の種類は「個人」である必要があります。</p>
SkipMeasures	<p>Y に設定すると、クイックデザインウィザードのデモンション作成に関連する画面をスキップします。</p>
File	<p>ユニバースを自動的に作成する特定の出力フォーマットで構成されたデータをもつ外部テキストファイルのファイルパスです。詳しい説明は、416 ページの「データのテキストファイルの作成」を参照してください。</p>

例 oracle.stg のクラスとオブジェクト外部ストラテジー

Oracle の外部ストラテジーファイルは oracle.stg で、
 \$INSTALLDIR/dataAccess/RDBMS/connectionServer/oracle/oracle.stg. ディレクトリに格納されています。
 このファイルには、ユニバースデザインツールで提供されている外部ストラテジーの例が多数含まれています。
 これらのストラテジーは、カスタマイズしたり、新しいストラテジーのテンプレートとして使用したりできます。

自動的にテーブルをクラスに、列をオブジェクトに関連付ける oracle.stg ファイルの外部ストラテジーを次に示します。

```
<Strategy Name="Classes_and_Objects">
  <Type>OBJECT</Type>
  <SQL>SELECT
    U1.table_name,'|',
    U1.column_name,'|',
    translate(initcap(U1.table_name),' ',''),'|',
    translate(initcap(U1.column_name),' ',''),'|',
    U1.table_name||'|'||U1.column_name,'|',
    ',','|',
    decode(SUBSTR(U1.DATA_TYPE,1,1),'N','N','F','N','D','D','C'),'|',
    SUBSTR(U2.comments,1,474),'|',
    'O','|'
  FROM USER_TAB_COLUMNS U1,USER_COL_COMMENTS U2
  WHERE
    U1.table_name=U2.table_name
  and U1.column_name=U2.column_name
  UNION
  SELECT
    S.SYNONYM_NAME,'|',
    U1.column_name,'|',
    translate(initcap(S.SYNONYM_NAME),' ',''),'|',
    translate(initcap(U1.column_name),' ',''),'|',
    S.SYNONYM_NAME||'|'||U1.column_name,'|',
    ',','|',
    decode(SUBSTR(U1.DATA_TYPE,1,1),'N','N','F','N','D','D','C'),'|',
    SUBSTR(U2.comments,1,474),'|',
    'O','|'
  FROM ALL_TAB_COLUMNS U1, ALL_COL_COMMENTS U2, ALL_OBJECTS O, USER_SYNONYMS S
  WHERE
    S.table_owner=O.owner
  AND  S.table_name=O.object_name
  AND  (O.OBJECT_TYPE='TABLE' OR O.OBJECT_TYPE='VIEW')
  AND  O.owner=U1.owner
  AND  O.object_name=U1.table_name
  AND  U1.owner=U2.owner
  AND  U1.table_name=U2.table_name
  AND  U1.column_name=U2.column_name</SQL>
</Strategy>
```

7.4.8 ストラテジーの出力フォーマット

SQL スクリプトは、外部ストラテジーファイルの <SQL> タグ内に記述またはコピーします。SQL スクリプトが返す情報の順序およびタイプは、作成するストラテジーがオブジェクト、結合、テーブルのいずれであるかに依存します。ユニバースデザインツールでは、ストラテジーのタイプにより異なる情報が必要です。

ストラテジーの SQL スクリプトを作成するとき、そのスクリプトの出力が以下に説明される出力フォーマットと一致していることを確認する必要があります。

スクリプトの出力は、一連の列としてフォーマットされます。各列は、オブジェクト、結合、またはテーブルなどのコンポーネントの作成に使用される生成情報の単位に対応します。

この節では、次の各ストラテジーに応じた出力フォーマットを紹介します。

- ・ オブジェクト作成ストラテジー
- ・ 結合作成ストラテジー
- ・ テーブル参照ストラテジー

7.4.8.1 オブジェクトストラテジー (OBJECT) の出力書式

オブジェクトストラテジーの出力ファイルには 9 つの列があります。値が NULL の場合でも、列数は常に 9 列必要です。返される値はすべてパイプ文字 '|' で区切られ、末尾にも必ずパイプ文字が付きます。

列番号	列の内容	説明
1	テーブル	テーブル名の書式は[修飾子].[所有者.]です。テーブルの各名称は、英数半角で 35 文字までです。この列を空白にしておくと、SELECT句(第5列)とWHERE句(第6列)によってテーブルが取得されます。
2	列名	列の名前です。
3	クラス名	クラスの名前です。クラス名¥サブクラス名形式をサポートしています。
4	オブジェクト名	オブジェクトの名前または条件です。オブジェクト名を空白にしておくと、クラスとその説明が作成されます。
5	Select	SELECT 文
6	Where	Select 列を空白にし WHERE 句を指定すると、定義済みの条件とその説明が生成されます。
7	型	C (文字)、N (数値)、D (日付)、T (ロングテキスト) のいずれかを指定します。指定しないと、デフォルトで C (文字) が選択されます。
8	説明	オブジェクトの説明です。

列番号	列の内容	説明
9	資格	D (ディメンション)、M (メジャー)、I (詳細) のいずれかを指定します。指定しないと、デフォルトでD (ディメンション) が選択されます。

例 列のコメントをオブジェクトの説明にコピーする外部オブジェクトストラテジー

次のサンプルには WHERE 句がありません。WHERE 句の出力列は空になります。

<Strategies>

<Strategy Name="Read Column descriptions">

<Type>OBJECT</Type>

<SQL>Select

	列	説明
Table_name, ' ',	1	テーブル名
Column_name, ' ',	2	列名
Replace (Table_name, '_', ' '), ' ',	3	テーブル名のアンダースコアをクラス名の空白に置き換えます。
Replace (Column_name, '_', ' '), ' ',	4	列名のアンダースコアをオブジェクト名の空白に置き換えます。
Table_name '.' Column_name, ' ',	5	テーブル名と列名をピリオドをはさんで連結します。これは、SELECT 句です。
, ' ',	6	WHERE 句はありません。
Column_type, ' ',	7	システムテーブルから列の型を取得します。

Column_Desc,' ',	8	システムテーブルから列の説明を取得します。
' ',' '	9	オブジェクトのタイプが NULL の場合、デフォルトでディメンションが選択されます。

</SQL>

7.4.8.2 結合ストラテジー (JOIN) の出力書式

結合ストラテジーの出力書式は、次の通りです。

列番号	列の内容	説明
1	テーブル 1	結合の最初のテーブルの名前です。
2	テーブル 2	結合の 2 番目のテーブルの名前です。
3	結合定義	テーブル1.列1=テーブル2.列2 という書式で表現される、結合の実際の定義です。
4	外部結合の種類	外部結合の種類です。L は外部左結合、R は外部右結合です。空白にしておくと、[外部結合なし]になります。
5	カーディナリティ (オプション)	有効な値は 11、1N、N1 です。

7.4.8.3 テーブル参照ストラテジー (STRUCT) の出力フォーマット

テーブル参照ストラテジーの出力書式は、次の通りです。

列番号	列の内容	説明
1	修飾子	使用データベースに依存: テーブル修飾子は、データベース名、またはその他の識別子です。
2	所有者	使用データベースに依存
3	テーブル	テーブル、ビュー、またはシノニムの名称
4	列	列の名称
5	データ型	C (文字)、N (数値)、D (日付)、T (ロングテキスト) のいずれかを指定します。指定しないと、デフォルトでC (文字)が選択されます。
6	NULL 可/不可 Y (はい)またはN (いいえ)のいずれかを指定します。	列に NULL 値を許容するかどうかの指定です。

7.4.9 外部ストラテジーの作成

外部ストラテジーを作成するには 2 通りの方法があります。

外部ストラテジーの作成方法	XML ファイルのタグ	説明
SQL スクリプトを直接入力	SQL	ストラテジーの SQL スクリプトを、SQL タグを使用して外部ストラテジーファイルに直接挿入します。
外部ファイルのデータを参照する	FILE	ストラテジーのデータが入った外部テキストファイルのファイルパスと名前を入力します。

次の手順では、この2種類の設定方法を説明します。

7.4.9.1 外部ストラテジーの作成

外部ストラテジーを直接作成する手順は、次の通りです。

- 1 ターゲットの RDBMS の外部ストラテジーファイルを XML エディタで開きます。ターゲットの RDBMS の外部ストラテジーファイルの保存場所は次の通りです。

`$INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/<RDBMS>.stg`.

- 2 新しいストラテジー要素を作成します。

これは新しいストラテジーです。XML Spy などの XML エディタを使用している場合、ストラテジーの Name、Type、SQL などの要素が自動的に作成されます。

- 3 ストラテジーの名前を入力します。

ここで指定したストラテジー名は [ユニバースパラメータ] ダイアログボックスの [ストラテジー] タブ、またはクイックデザインウィザードで表示されます。

- 4 [TYPE=] パラメータで、ストラテジーの種類 (OBJECT、JOIN、STRUCT) を定義します。

例: TYPE=OBJECT

- 5 このストラテジーのSQL文を定義します。SQL 文のフォーマットは、410 ページの「[ストラテジーの出力フォーマット](#)」を参照してください。

または

データの入ったテキストファイルを参照する場合、SQL 要素を File 要素で置き換えます。
「C:\Path\Filename.txt」のように、データファイルのファイルパスを入力します。

- 6 必要に応じて、オプションの要素を追加し、値を設定します。
- 7 XML ファイルが有効であることを確認し、保存してからファイルを閉じます。

- 8 この外部ストラテジーファイルが、ターゲット RDBMS の汎用データアクセスファイル (<RDBMS>.SBO) で宣言されていることを確認します。手順は、次の通りです。

- ・ 次のディレクトリにある汎用データアクセスファイル (SBO) を開きます。

```
$INSTALLDIR/dataAccess/RDBMS/connectionServer/<RDBMS>/
```

- ・ Strategies File 要素が外部ストラテジーファイルの名前に設定されていることを確認します。これはデフォルト値です。
- ・ SBO ファイルを変更した場合、保存してファイルを閉じます。

この外部ストラテジーは、[パラメータ] ダイアログボックスの [ストラテジー] ページにある、[結合]、[オブジェクト]、[テーブル] のドロップダウンリストに表示されます。新しく作成した外部ストラテジーが表示されるようにするには、ユニバースデザインツールを終了して再起動する必要があります。

注

この外部ストラテジーを [ストラテジー] ページで選択したときに表示される Help テキストを追加する場合、外部ストラテジーファイルとして同じディレクトリに格納された、<RDBMS><言語>.STG ファイルにそのテキストを追加します。外部ストラテジーの Help テキストの追加については、405 ページの [「外部ストラテジーの Help テキストの作成」](#)を参照してください。

7.4.10 データのテキストファイルの作成

外部ストラテジーのデータを含むテキストファイルを作成できます。外部ストラテジーを作成する際、直接 SQL を挿入する代わりに、テキストファイルのパスと名前を入力できます。外部ストラテジーファイルに FILE 要素を挿入し、その値にファイルパスと名前を設定します。

SQL スクリプトの出力には、ストラテジー、オブジェクト、結合、またはテーブルの各タイプに適切なフォーマットを使用することが必要です。出力フォーマットについては、410 ページの [「ストラテジーの出力フォーマット」](#)を参照してください。

ただし、いずれのフォーマットでもデフォルトの区切り文字にはタブ記号が使われています。

7.4.11 ユニバースデザインツールへの外部ストラテジーの適用

外部ストラテジーの適用手順は、次の通りです。

- 1 使用する外部ストラテジーが、[パラメータ] ダイアログボックスの [ストラテジー] ページで選択されていることを確認します。

次はその例です。

- ・ オブジェクトストラテジーを使用してオブジェクトを抽出し、挿入するには、[挿入]メニューから[候補オブジェクト]コマンドを選択します。

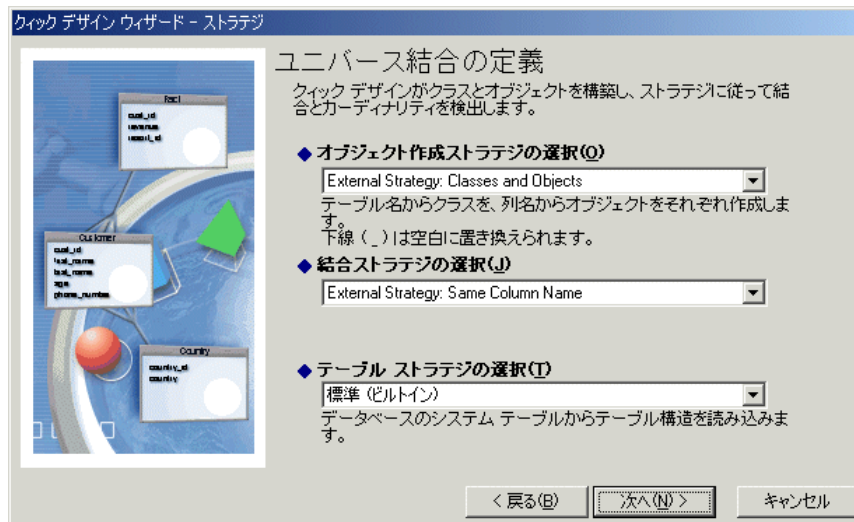
- ・ 結合戦略を使用して結合を派生させ、挿入するには、[ツール]メニューから[結合の検出]コマンドを選択します。
- ・ テーブル参照戦略を使用してテーブルを抽出し、挿入するには、[挿入]メニューから、[テーブル]を選択します。

注

結合戦略を選択すると、ユニバースデザインツールでその戦略を使用して候補の結合とカーディナリティが検出されます。提案された結合またはカーディナリティを適用するよう選択できます。候補の結合やカーディナリティを選択した戦略に基づいて自動的に適用する場合、[オプション]ダイアログボックスの[データベース] ページ ([ツール] > [オプション] > [データベース]) にある作成オプションで対応するものを選択する必要があります。詳しい説明は、95 ページの「[戦略の自動作成機能の使い方](#)」を参照してください。

7.4.11.1 クイックデザインウィザードでの戦略の選択

クイックデザインウィザードで、外部戦略を選択することができます。これには、ウィザードの最初の画面で、[戦略の選択] オプションをクリックする必要があります。



7.5 分析関数の使い方

ユニバースデザインツールでは、RDBMS 固有の分析関数の使用がサポートされています。分析関数は、RedBrickではRISQL関数、TeradataではOLAP関数とも呼ばれます。ユニバースデザインツールを使用してユニバース内のオブジェクトの分析関数を定義できます。

Web Intelligence ユーザは分析関数も使用できるため、通常は InfoView のレポート機能に提供されないデータ分析を実行することができます。Web Intelligence でのこれらの関数およびその他の関数の使用方法について詳しくは、『Web Intelligence における式、関数、計算の使い方』の『スマートメジャーを使用した値の計算』の節を参照してください。

この節では次の RDBMS について、ユニバース内のオブジェクトに分析関数、RISQL 関数、OLAP 関数を定義する方法を説明します。

- ・ 420 ページの [「IBM DB2 UDB および Oracle」](#)
- ・ 424 ページの [「RedBrick\(RISQL関数\)」](#)
- ・ 427 ページの [「Teradata\(OLAP 関数\)」](#)

7.5.1 分析関数とは

分析関数とは、整列された行グループあるいはパーティションに分割された結果集合に分析処理を行う関数です。

ユニバースデザインツールでは、オブジェクトに分析関数を定義して区分内の順位、累計、比率を計算することができます。使用するRDBMSによっては、パーティション内で分析を適用する行範囲を指定することもできます。

分析関数の詳しい説明は、使用するRDBMSに付属の資料を参照してください。

7.5.2 分析関数の利点

ユニバースデザインツールで分析関数を使ってオブジェクトを定義すると、Web Intelligence ユーザは次の利点を得られます。

- ・ 作業の軽減。分析関数が定義されたオブジェクトがあると、通常ならばレポートレベルで拡張構文を使わなければならないようなデータ分析が可能になります。
- ・ 関数の追加。移動平均の計算や特殊集計の適用など、InfoViewでは使用できない分析処理ができるようになります。分析関数を定義したオブジェクトを使用することで、Web Intelligence ユーザはこれまで不可能だった詳細データ分析が可能になります。
- ・ クエリパフォーマンスの改良: 計算はサーバ上で行われます。

7.5.3 サポートされる分析関数ファミリー

次のファミリーの分析関数を使用できます。

- ・ 順位
- ・ 累計
- ・ 比率、対レポート比率、レポート内集計

7.5.4 ユニバースデザインツールでの分析関数の使用方法

分析関数を使うと、オブジェクトのSELECT文に分析関数を定義できます。

各パラメータファイル(PRM)のRDBMSセクションには、SELECT文で利用できる分析関数の一覧が指定されています。この一覧には、分析関数をサポートするRDBMSのそれぞれの関数ファミリーにあるすべての分析関数が含まれているわけではありません。

7.5.4.1 PRM ファイルとは

PRM ファイルは、ユニバース作成と Web Intelligence 製品での SQL クエリ生成に使用されるパラメータを記録したテキストファイルです。サポートするそれぞれのRDBMSに対し、専用のPRMファイルがあります。PRMファイルは次のフォルダに保存されています。

<INSTALLDIR>%dataAccess%RDBMS%connectionServer%<rdbs>%

パラメータファイルの内容変更の詳細については、『データアクセスガイド』を参照してください。

分析関数を使用する前に、その関数がPRMファイルに記載されているかどうかを確認してください。記述されていない場合は、一覧に関数名を追加できます。追加すると、オブジェクトに対するSELECT文内でのその関数の使用がユニバースデザインツールでサポートされます。詳しい説明は、421 ページの「[PRMファイル内の分析関数の確認と追加](#)」を参照してください。

7.5.4.2 それぞれのRDBMSでの分析関数の使い方

次のそれぞれのRDBMSについて、分析関数の使い方を説明します。

- ・ SELECT文での分析関数、RISQL関数、OLAP関数の構文
- ・ 分析関数を追加するためのPRMファイルの変更方法と確認方法
- ・ 分析関数に関するRDBMS固有の規則と制限

- ・ SELECT文の編集で、自動的に分析関数の構文を読み込む方法

7.5.5 IBM DB2 UDB および Oracle

この 2 つの RDBMS では、分析関数の構文は同じです。

7.5.5.1 DB2、UDB および Oracle RDBMS の SELECT 文の定義

分析関数は、オブジェクトの SELECT 文に定義します。[SELECT句]ボックスあるいはSQLエディタで分析関数の構文を入力します。

注

SQLエディタの[関数]一覧に分析関数を追加し、その構文を自動的に読み込むことができます。[関数]一覧に関数を追加するには、.prm ファイルの[FUNCTIONS]セクションに分析関数を追加します。詳しい説明は、429 ページの「[SELECT 文への構文の自動読み込み](#)」を参照してください。

分析関数には、必ずOVERというキーワードが使われます。

```
RANK() OVER (PARTITION BY calender.cal_year ORDER BY SUM(telco_facts.total_billed_rev)DESC)
```

キーワードOVERに続く部分で、パーティションと結果テーブルでの行の並べ替えを定義します。

次の表は、ファミリーごとに分析関数の構文をまとめたものです。

関数ファミリー	構文	説明
順位	<code>RANK() OVER(PARTITION BY arg1 ORDER BY arg2 ASC/DESC)</code>	<ul style="list-style-type: none"> ・ arg1は省略できます。引数を指定しない場合は、デフォルトで結果集合全体がパーティションとなります。 ・ arg2は省略できません。この引数の値に従って順位を付けます。 ・ ASC/DESCで昇順/降順を指定します。デフォルトはASC (昇順)です。
Windows Aggregate	<code>SUM(arg1) OVER(PARTITION BY arg2 ORDER BY arg3)</code>	<ul style="list-style-type: none"> ・ arg1は、累計対象を指定する引数です。 ・ arg2はリセット条件です。この引数は省略できます。 ・ arg3はグループ条件です。この引数は省略できます。
レポート作成 Aggregate	<code>RATIO_TO_REPORT(arg1) OVER(PARTITION BY arg2)</code>	<ul style="list-style-type: none"> ・ arg1は、比率計算対象を指定する引数です。 ・ arg2はリセット条件です。この引数は省略できます。

Window句の使い方

ウィンドウ内集計ファミリーでは、arg3に続けてウィンドウのサイズを定義する<window句>を定義することができます。例:

```
<window frame units> ::= ROW | RANGE <window frame start> ::= UNBOUNDED PRECEDING | <window frame preceding> | CURRENT ROW <window frame between>
```

BETWEEN句とウィンドウのサイズ定義の構文については、使用するRDBMSに付属の資料を参照してください。

7.5.5.2 PRMファイル内の分析関数の確認と追加

分析関数をサポートできるようIBM DB2 UDBとOracleのPRMファイルが更新されました。

ただし、RDBMSで利用できるすべての分析関数が含まれているとは限りません。分析関数を使用する場合は、その関数がPRMファイルの[RDBMS]セクションに定義されているか確認し、必要に応じて関数名を追加してください。

次のようにして、それを行います。

Oracle と IBM DB2 の PRM ファイルに分析関数を追加する手順は、次のとおりです。

- 1 Business Objects パスの DataAccess フォルダに移動します。
- 2 使用するRDBMSに対応するPRMファイルをテキストエディタで開きます。
- 3 [RDBMS]セクションまでスクロールします。
- 4 次のパラメータが指定されていることを確認します。

PRMファイル内のパラメータと値	説明
OVER_CLAUSE = Y	適切な SQL(OVER_CLAUSE)を生成します。
RISQL_FUNCTIONS = <使用する関数の一覧>	使用できる分析関数

- 5 一覧にない分析関数を使用するには、一覧の最後に関数名を追加します。たとえばRATIO_TO_REPORT関数を追加するには、次のように一覧の最後にこれを追加します。

```
[RDBMS]
{GENERAL}
...
OVER_CLAUSE=Y
RISQL_FUNCTIONS=RANK,SUM,AVG,COUNT,MIN,MAX,
VARIANCE,STDDEV,RATIO_TO_REPORT
```

- 6 ファイルを保存して、エディタを終了します。

PRM ファイルに行った変更を反映するには、ユニバースデザインツールを再起動する必要があります。

7.5.5.3 DB2、UDB および Oracle RDBMS 分析関数の使用ルール

DB2 UDBとOracle 8.1.6で分析関数を使う場合は、次の規則に従ってください。

規則	説明
Group By句では分析関数を使用しない。	SUMのような集計関数が定義された分析関数は Group By句でも使用できますが、RANKのような分析関数は使用できません。 分析関数を Group By句で使用できないよう、PRMファイルのRISQL FUNCTIONSパラメータに続けて、分析関数がリストアップされます。これに先立つ OVER_CLAUSEは、デフォルト設定の Y に設定する必要があります。
分析関数で Group By句を生成してはならない。	SQL エディタの[関数]一覧に表示するため、PRMファイルの[FUNCTIONS]セクションに分析関数を追加した場合は、その GROUP CLAUSE パラメータを Nに設定する必要があります。これで、GROUP BY句が生成されなくなります。詳しい説明は、429 ページの「 SELECT 文への構文の自動読み込み 」を参照してください。
集計関数を持つ分析関数では、分析関数が使用するすべてのディメンションを Group By 句で指定しなければならない。	たとえば、「RANK()OVER(PARTITION BY年ORDER BY SUM(売上げ))」では、クエリ内で RANK 関数を単独で使用している場合でも、Group By 句でディメンション“年”を指定する必要があります。

7.5.5.4 OracleとDB2での分析関数の使用制限

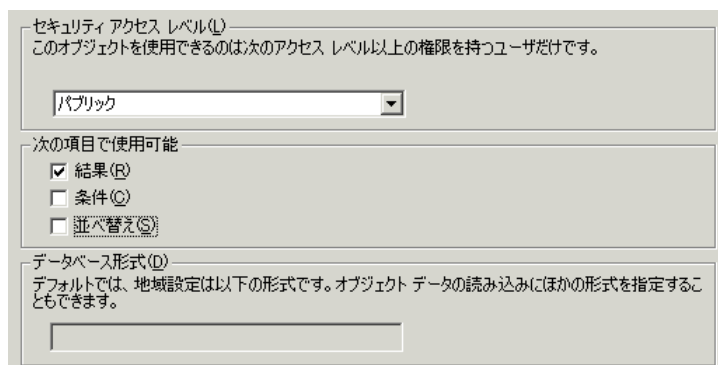
IBM DB2 UDB v7.1とOracle 8.1.6での分析関数の使用には、次の制約があります。

- 分析関数を定義したオブジェクトには、@Prompt 関数と @Variable 関数は使用できない。
- ユーザオブジェクトに分析関数を使用できない。SQLエディタの[関数]一覧に表示するため、PRMファイルの[FUNCTIONS]セクションに分析関数を追加した場合は、そのIN MACROパラメータをNに設定する必要があります。
- 分析関数を定義したオブジェクトは、条件で使用したり、並べ替えを適用することができない。エンドユーザがこのオブジェクトを使って条件を定義すると、SQLエラーメッセージが表示されます。エンドユーザが、このオブジェクトを使って条件を定義したり、条件に並べ替えを適用できないよう設定するには、オブジェクトのプロパティを次のように変更します。

条件や並べ替えでの分析オブジェクトの使用の禁止

条件や並べ替えでの分析関数の使用を禁止する手順は、次の通りです。

- 1 ユニバースデザインツールでオブジェクトを右クリックします。
- 2 ポップアップメニューから[オブジェクトのプロパティ]を選択します。
[プロパティの編集]ダイアログボックスが表示されます。
- 3 [詳細]タブをクリックし、[次の項目で使用可能]グループボックスにある[条件]と[並べ替え]チェックボックスをオフにします。



- 4 [OK]をクリックします。

7.5.6 RedBrick(RISQL関数)

ここでは、ユニバースデザインツールでの RISQL 関数の使用方法について説明します。

7.5.6.1 RedBrick RISQL 関数の SELECT 文の定義

分析関数は、オブジェクトの SELECT 文に定義します。[SELECT句]ボックスあるいはSQLエディタで分析関数の構文を入力します。

注

SQLエディタの[関数]一覧にRISQL関数を追加し、その構文を自動的に読み込むことができます。[関数]一覧にRISQL関数を追加するには、PRMファイルの[FUNCTIONS]セクションにRISQL関数を追加します。詳しい説明は、429 ページの「[SELECT 文への構文の自動読み込み](#)」を参照してください。

次の表は、ファミリーごとにRISQL関数の構文をまとめたものです。

関数ファミリー	構文	説明
順位(RANK)	RANK(arg1) 例: RANK(SUM(telco_facts.total_billed_rev))	arg1は、省略できません。この引数の値に従って順位を付けます。
集計ファミリー(CUME、MOVINGAVG、MOVINGSUM)	MOVINGSUM(arg1,Number) 例: MOVINGSUM (COUNT(complaints.id),2)	<ul style="list-style-type: none"> arg1は、省略できません。この引数の値に従って累計します。 行数は省略できます。指定した行数分を集計します。
比率 (RATIOTOREPORT)	RATIOTOREPORT(arg1) 例: RATIOTOREPORT (SUM(telco_facts.total_billed_rev))	arg1は、省略できません。この引数の値に従って比率を計算します。

7.5.6.2 PRMファイルでのRISQL関数サポートの確認と追加

PRMファイルには、RDBMSで利用できるすべてのRISQL関数が含まれているとは限りません。RISQL 関数を使用する場合は、その関数が PRM ファイルの[RDBMS]セクションに定義されているか確認し、必要に応じて関数名を追加してください。次のようにして、それを行います。

RedBrickのPRMファイルにRISQL関数を追加する手順は、次の通りです。

- 1 Business Objects パスの DataAccess フォルダに移動します。
- 2 使用するRDBMSに対応するPRMファイルをテキストエディタで開きます。
- 3 [RDBMS]セクションまでスクロールします。
- 4 次のパラメータが指定されていることを確認します。

PRMファイル内のパラメータと値	説明
OLAP_CLAUSE = WHEN	条件の適用
RISQL_FUNCTIONS = <使用する関数の一覧>	使用できる分析関数

設定例は、次の通りです。

```
[RDBMS]
(GENERAL)
...
OVER_CLAUSE=WHEN
RISQL_FUNCTION= RANK, CUME, MOVINGSUM, MOVINGAVG, RATIOTOREPORT, TERTILE
```

- 5 一覧にないOLAP関数を使用するには、一覧の最後に関数名を追加します。
- 6 ファイルを保存して、エディタを終了します。

PRM ファイルに行った変更を反映するには、ユニバースデザインツールを再起動する必要があります。

7.5.6.3 RISQL関数の使用規則

RISQL関数を使う場合は、次の規則に従ってください。

規則	説明
Group By句ではRISQL関数を使用しない。	SUMのような集計関数が定義されたRISQL関数はGroup By句でも使用できますが、RANKのような分析関数は使用できません。 RISQL関数をGroup By句で使用できないよう、PRMファイルのRISQL FUNCTIONSパラメータに続けて、RISQL関数をリストアップします。RISQL FUNCTIONSの直前にあるOVER_CLAUSEは、WHENに設定する必要があります。これは、デフォルト設定です。
RISQL関数でGroup By句を生成してはならない。	SQL エディタの[関数]一覧に表示するため、PRMファイルの[FUNCTIONS]セクションに RISQL 関数を追加した場合は、その GROUP CLAUSE パラメータを N に設定する必要があります。この設定では、関数は Group By 句を生成できません。詳しい説明は、429 ページの「 SELECT 文への構文の自動読み込み 」を参照してください。
RISQL関数は、条件にも使用できる。	When句が生成されます。

7.5.6.4 RedBrickでの分析関数の使用制限

RISQL関数の使用には、次の制約があります。

- ・ RESET BY 句はサポートされません。
- ・ SORT BY句はサポートされません。オブジェクトプロパティを編集することによって、エンドユーザがオブジェクトをソートに使用できないようにする方法については、『Oracle と DB2 でのアナリティクス関数の使用に関する制限』を参照してください。

7.5.7 Teradata(OLAP 関数)

ここでは、ユニバースデザインツールでの OLAP 関数の使用方法について説明します。

7.5.7.1 Teradata OLAP 関数の SELECT 文の定義

Teradata V2R3では、Ratio関数は使用できません。OLAP関数は、オブジェクトのSelect句に定義します。
[SELECT句]ボックスあるいはSQLエディタで分析関数の構文を入力します。

関数一覧で関数を使用可能にし、構文入力を自動化する方法については、『Oracle と DB2 でのアナリティクス関数の使用に関する制限』を参照してください。

次の表は、ファミリーごとにOLAP関数の構文をまとめたものです。

関数ファミリー	構文	説明
順位(RANK)	RANK(arg1 DESC/ASC) 例: RANK(invoice_line.nb_guests)	<ul style="list-style-type: none"> ・ arg1は、省略できません。この引数の値に従って順位を付けます。指定できるのは、オブジェクトかオブジェクトのリストです。 <p>注</p> <p>arg1 には、集計関数(sum、avg、min、count)を使ったオブジェクトは指定できません。</p> <ul style="list-style-type: none"> ・ ASC/DESCで昇順/降順を指定します。デフォルトはASC (昇順)です。

関数ファミリ	構文	説明
集計ファミリ(CSUM、MAVG、MDIFF、MLINREG、MSUM)	CSUM(arg1 DESC/ASC) 例: CSUM(invoice_line.nb_guests)	<ul style="list-style-type: none"> arg1は、省略できません。この引数の値に従って累計します。指定できるのは、オブジェクトかオブジェクトのリストです。 ASC/DESCで昇順/降順を指定します。デフォルトはASC (昇順)です。

7.5.7.2 PRM ファイル内の OLAP 関数の確認と追加

OLAP関数をサポートできるようTeradataのPRMファイルが更新されました。ただし、RDBMSで利用できるすべてのOLAP関数が含まれているとは限りません。OLAP 関数を使用する場合は、その関数が PRM ファイルの [RDBMS]セクションに定義されているか確認し、必要に応じて関数名を追加してください。次のようにして、それを行います。

TeradataのPRMファイルにOLAP関数を追加する手順は、次の通りです。

- 1 Business Objects パスの DataAccess フォルダに移動します。
- 2 使用するRDBMSに対応するPRMファイルをテキストエディタで開きます。
- 3 [RDBMS]セクションまでスクロールします。
- 4 次のパラメータが指定されていることを確認します。

PRMファイル内のパラメータと値	説明
OLAP_CLAUSE = QUALIFY	条件の適用
RISQL_FUNCTIONS = <使用する関数の一覧>	使用できる分析関数

設定例は、次の通りです。

```
[RDBMS]
(GENERAL)
...
OVER_CLAUSE=QUALIFY
RISQL_FUNCTION= RANK, CSUM, MAVG, MDIFF, MLINREG, MSUM, QUANTILE
```

- 5 一覧にないOLAP関数を使用するには、一覧の最後に関数名を追加します。

- 6 ファイルを保存して、エディタを終了します。

PRM ファイルに行った変更を反映するには、ユニバースデザインツールを再起動する必要があります。

7.5.7.3 RISQL関数の使用規則

OLAP関数を使う場合は、次の規則に従ってください。

- ・ Group By句ではOLAP関数を使用しない。OLAP関数をGroup By句で使用できないよう、PRMファイルのRISQL FUNCTIONSパラメータに続けて、分析関数がリストアップします。RISQL FUNCTIONS の直前にある OVER_CLAUSE は、QUALIFY に設定する必要があります。これは、デフォルト設定です。
- ・ 同じクエリ内では、集計関数を使って定義されたオブジェクトとOLAP関数を使って定義されたオブジェクトを組み合わせることはできない。
- ・ OLAP関数は条件で使用できる。QUALIFY句が生成されます。
- ・ OLAP関数は並べ替えで使用できる。SORT BY句が生成されます。

7.5.7.4 Teradataでの分析関数の使用制限

OLAP関数の使用には、次の制約があります。

- ・ RESET BY 句はサポートされない。
- ・ サブクエリでは、OLAP関数を使用できない。
- ・ 同じSelect文で2つのOLAP関数は使用できない。
- ・ OLAP関数内では他の関数を参照できない。
- ・ ユーザオブジェクトに OLAP 関数を使用できない。

7.5.8 SELECT 文への構文の自動読み込み

SQLエディタ([(オブジェクト名)のSELECT句の編集]ダイアログボックス)の[関数]一覧に分析関数を追加し、その構文を自動的に読み込むことができます。

[関数]一覧に分析関数を追加するには、使用するRDBMSに対応したPRMファイルの[FUNCTIONS]セクションに分析関数を追加します。

PRMファイルに追加された関数は、SQLエディタ[(オブジェクト名)のSELECT句の編集]ダイアログボックスの[関数]一覧に表示されます。一覧から関数をダブルクリックすると、その構文が自動的に編集ボックスに挿入されます。

PRMファイルに分析関数を追加する場合は、次のパラメータを設定する必要があります。

パラメータ	説明
GROUP = N	分析関数、RISQL関数、OLAP関数がGroup By句を生成しないよう設定する必要があります。このパラメータをNに設定すると、分析関数はGroup By句を生成できなくなります。
IBM DB2 UDB バージョン 7.1 と ORACLE 8.1.6 のみ: IN_MACRO = N	DB2 UDB と Oracle の分析関数が、ユーザオブジェクトとして使用できないように設定します。RedBrickとTeradataでは、このパラメータはYに設定できます。

PRMファイルの[FUNCTIONS]セクションに分析関数を追加する手順は、次の通りです。

PRMファイルに分析関数を追加する手順は、次の通りです。

- 1 Business Objects パスの DataAccess フォルダに移動します。
- 2 使用するRDBMSに対応するPRMファイルをテキストエディタで開きます。
- 3 [FUNCTION]セクションまでスクロールします。
- 4 既存の関数をどれかコピーし、一覧の最後に貼り付けます。
- 5 関数番号を変更して一意の番号にし、追加する分析関数に合わせて内容を変更します。
- 6 GROUPの値をNにします。

IBM DB2 UDBおよびORACLEを使う場合は、IN_MACROの値をNにします。

以下はその例です。

```
(n)
NAME: RANK
TRAD:
HELP: Return the rank of
TYPE=N
IN_MACRO=N
GROUP=N
SQL=
```

- 7 PRMファイルを保存して、エディタを終了します。

変更を反映するには、ユニバースデザインツールを再起動する必要があります。

注

ユニバースデザインツールを再起動すると、追加した分析関数の構文が、対応する種類[数値]、[文字型]、[日付])の下に表示されます。

7.6 SQL プレフィックス関数

SQL プレフィックス関数は、Begin_SQL ユニバースパラメータを使用して、BusinessObjects によって生成される SQL 文の前に SQL 命令を挿入します。これによって、生成されたすべての SQL 文の前に命令を実行できます。この機能は、SELECT 文の前にパラメータを渡すことをサポートするすべてのデータベースで使用できます。いくつかの例を次に示します。

- ・ Teradata: トランザクションに 'QUERY_BAND' を使用します。Teradata のドキュメンテーションを参照してください。
- ・ Oracle: 読み取り専用トランザクションを開始します。
- ・ Netezza: 最適化オプションを開始します。

SQL プレフィックス関数を設定するには、ユニバースに対して SQL_prefix パラメータを設定します。

関連項目

- ・ 103 ページの[ユーザインタフェースに設定する SQL パラメータ](#)

7.6.1 BEGIN_SQL ユニバースパラメータで SQL 文にプレフィックスを付ける

お使いのデータベースは、SELECT 文の前にパラメータを渡すことをサポートします。

BEGIN_SQL パラメータでは、SQL 文が生成されるたびに同じパラメータを SQL 文の前に付けることができます。

- 1 [ユニバースのパラメータ]ダイアログボックスが開きます。
- 2 [パラメータ]タブをクリックします。
- 3 パラメータの一覧で BEGIN_SQL パラメータを選択し、使用するプレフィックスコマンドを入力します。
- 4 設定を保存します。
- 5 ユニバースを保存します。

例

この例では、Teradata で BEGIN_SQL パラメータを使用します。クエリには、レポーティングで使用するためにクエリにバインディングされているユーザ ID とアプリケーション IDが含まれています。[ユニバースのパラメータ]ダイアログボックスの[パラメータ]枠で、BEGIN_SQL パラメータが次のように設定されます。

```
BEGIN_SQL = SET QUERY_BAND = 'UserId=DG12234;AppId=TRM;' FOR TRANSACTION;
```

クエリを実行すると、次の 2 つの SQL 文が実行されます。

1)BEGIN_SQL 文

```
SET QUERY_BAND = 'UserId=DG12234;Appld=TRM;' FOR TRANSACTION;
```

2)メインクエリの SQL 結果文

```
SELECT  
  RESORT_COUNTRY.COUNTRY, sum(INVOICE_LINE.DAYS * INVOICE_LINE.NB_GUESTS  
    * SERVICE.PRICE)  
FROM  
  COUNTRY RESORT_COUNTRY, INVOICE_LINE, RESORT_COUNTRY.COUNTRY
```

7.7 Array Fetch パラメータの最適化

CS.CFG ファイル内の Array Fetch パラメータは、FETCH プロセスに許可する最大行数を設定するパラメータです。CFG ファイルは、Business Objects 製品からデータベースに対して実行されるクエリの特定のパラメータのデフォルト値を指定する XML ファイルです。

Array Fetch パラメータによってネットワーク上のパケットサイズが決まります。たとえば、Array Fetch の値を 20 に設定すると、100 行のデータを受け取るのに 5 回のフェッチ動作が実行されます。

データソースによっては FETCH サイズを変更できないこともあります。この場合はすべての行が 1 回の FETCH で返されます。BLOB 型のデータを取得する場合は、Array Fetch サイズを 1 にする必要があります。

大きな Array Fetch を送信可能なネットワークの場合は、Array Fetch サイズをより大きい値に変更 (1 ~ 999 の間) できます。この変更により FETCH プロセスがスピードアップし、クエリの処理時間が短縮できます。

7.7.1 Array Fetch パラメータの変更

Array Fetch パラメータを変更する手順は、次の通りです。

- 1 XML エディタで CS.CFG ファイルを開きます。
CFG ファイルは、次のディレクトリに保存されています。
<INSTALDIR>%dataAccess%\RDBMS%\connectionServer.
- 2 Array Fetch パラメータを検索します。
- 3 パラメータ値を設定します。CFG ファイルを保存して閉じます。
- 4 ユニバースデザインツールを再起動します。

7.8 テーブルウェイトの設定

テーブルウェイトとは、テーブルに含まれる行数を計測したものです。ウェイトの軽いテーブルは、重いテーブルよりも少ない行を含みます。デフォルトでは、BusinessObjectsは、テーブルを軽いものから重いものへと(行数の少ないものから多いものへと)並べ替えます。これにしたがってSQL文のFrom句内のテーブル順が決まります。

データベースレベルでのテーブルの並べ替え順は、データベースによって異なります。たとえば、Sybase で使用される順序は BusinessObjectsと同じですが、Oracle では逆になります。したがって生成されるSQLも、Oracleを除いたデータベースでは最適化されています。

Oracle データベースで使用する場合は、BusinessObjects のテーブル並べ替え順を逆転して SQL を最適化します。これには、対応するデータベースのPRMファイル内のパラメータを変更します。

7.8.1 PRMファイル内のテーブルウェイトの変更

PRM ファイル内のテーブルウェイトを変更する手順は、次のとおりです。

- 1 使用するデータベースに対応した PRM ファイルを XML エディタで開きます。

PRMファイルは、次のフォルダに保存されています。

`<INSTALLDIR>%dataAccess%RDBMS%connectionServer%<rdbs>%`

たとえば、Oracle のファイル、oracle.prm は次の場所にあります。

`<INSTALLDIR>%dataAccess%RDBMS%connectionServer%oracle%oracle.prm`

- 2 このファイルの Configuration セクションで REVERSE_TABLE_WEIGHT パラメータを探します。
- 3 YをNに変更します。

REVERSE_TABLE_WEIGHT=N となったことを確認します。

このパラメータが指定されていない場合のデフォルト値はYです。

- 4 パラメータをNに設定すると、BusinessObjectsはテーブルを行数の多い順に並べ替えます。
- 5 PRMファイルを保存し、エディタを終了します。
- 6 ユニバースデザインツールを再起動し、.PRM ファイルに変更を適用します。

7.9 テーブルから返される行数の変更

ユニバースデザインツールでは、任意のテーブルの行数を手動で変更することもできます。テーブルの行数を確認するには、[表示]>[テーブルの行数]の順に選択します。テーブルシンボルの左下にそのテーブルの行数が表示されます。行数を変更する手順は、次の通りです。

7.9.1 返される行数の変更

テーブルから返される行数を変更する手順は、次のとおりです。

- 1 ユニバースデザインツールでユニバースを開きます。
- 2 行数を変更するテーブルを右クリックします。
- 3 ポップアップメニューから[テーブルの行数]を選択します。
[テーブルの行数]ダイアログボックスが開きます。
- 4 [テーブルの行数を手動で変更]ラジオボタンを選択します。
ダイアログボックスの左にテキストボックスが開きます。
- 5 テキストボックスに行数を入力します。この値がテーブルの行数として指定されます。
- 6 [OK]をクリックし、ユニバースを保存します。

7.10 ショートカット結合の使用

ショートカット結合は、共通のパス内ですでに結合されている2つのテーブルをリンクする結合です。ショートカット結合を使うことで、クエリが使用するテーブルの数を減らすことができます。詳細については、183 ページの「[ショートカット結合](#)」を参照してください。

注

ショートカット結合では、ループは作成されません。

OLAP ユニバースの使用

8.1 OLAP ユニバースについて

8.1.1 OLAP ユニバースとは

OLAP ユニバースは、OLAP キューブまたはクエリから生成された BusinessObjects ユニバースです。OLAP ユニバースは、選択した OLAP データソースへの接続から自動的に生成されます。

作成されたユニバースは、他のユニバースと同様に Central Management Server(CMS)にエクスポートできます。CMS にエクスポートすると、Web Intelligence ユーザは、そのユニバースを使用してクエリを実行し、レポートを作成できるようになります。

OLAP ユニバースの生成および管理は、次の手順にて行います。

- ・ まず最初に、OLAP データソースを選択して、OLAP ユニバースを生成します。

注

OLAP データソースへのセキュリティ接続では、ユニバースを生成したり、構造を最新表示したりする必要のあるすべてのユーザが、接続へのダウンロード権限を付与されている必要があります。この権限は、管理者が CMC で設定します。

- ・ 新規接続ウィザードを使って、データソース接続を定義し、新規ユニバースの接続を選択します。

ユニバースデザインツールによって自動的にユニバースが生成されます。OLAP 構造は、ユニバース内のクラス、メジャー、ディメンション、詳細、およびフィルタに直接マップされます。[ユニバース]枠にユニバース構造が表示されます。

- ・ OLAP ユニバースを保存して、CMS にエクスポートすることができます。
- ・ OLAP ユニバースコンポーネントはすべて変更可能です。
- ・ OLAP ユニバースの更新ウィザードを使用すると、OLAP ユニバースのライフサイクルが管理できます。ウィザードを使用すると、OLAP データソースで行った変更がユニバース構造に自動更新されます。このウィザードは、生成されたオブジェクトと手動で追加または変更されたオブジェクトを区別できます。このため、ユニバースデザインツール内での手動による変更を保存できます。

関連項目

- ・ 436 ページの[ユニバース作成に使用できる OLAP データソース](#)」
- ・ 442 ページの[OLAP データソースへの接続](#)」
- ・ 453 ページの[ユニバースデザインツールの機能は OLAP ユニバースでサポートされています。」](#)
- ・ 467 ページの[OLAP ユニバースのライフサイクル管理](#)」

8.1.2 ユニバース作成に使用できる OLAP データソース

次の OLAP データソースから自動的に OLAP ユニバースを作成できます。

- ・ SAP NetWeaver Business Warehouse(BW)
- ・ Microsoft Analysis Services(MSAS)2000
- ・ Microsoft Analysis Services(MSAS)2005
- ・ Hyperion Essbase

注

ユニバースデザインツール、Web Intelligence リッチクライアント、および Web Intelligence などの SAP BusinessObjects OLAP 製品から、Essbase OLAP データソースに接続するには、これらの SAP BusinessObjects OLAP 製品をホストするマシン上に Essbase クライアントミドルウェアが正しくインストール、設定されていることを確認します。具体的には、Essbase クライアント環境変数 ARBORPATH と ESSBASEPATH が作成され、(Windows ユーザ環境変数ではなく) Windows システム環境変数として設定されていることを確認します。

ユニバースは、キューブまたはクエリーから生成されます。OLAP ユニバースは、ユニバース内で 1 つのキューブしかサポートしません。

関連項目

- ・ 436 ページの[SAP NetWeaver Business Warehouse\(BW\)データソース](#)」
- ・ 503 ページの[ユニバースにおける SAP NetWeaver BW オブジェクトのマッピングおよび使用方法](#)」
- ・ 440 ページの[OLAP ユニバース向けにサポートされる MSAS 機能](#)」
- ・ 513 ページの[MSAS キューブをユニバースコンポーネントにマップする方法](#)」
- ・ 441 ページの[OLAP ユニバースでサポートされる Essbase の機能](#)」
- ・ 512 ページの[Essbase キューブをユニバースコンポーネントにマップする方法](#)」

8.1.2.1 SAP NetWeaver Business Warehouse(BW)データソース

NetWeaver BW データソースに基づいて OLAP ユニバースを作成する場合、InfoCube/MultiCube に直接基づいて、または任意の InfoProvider の一番上の有効な BEx クエリに基づいて、OLAP ユニバースを作成できます。インフォプロバイダは、次のいずれかです。

- ・ インフォキューブ

- ・ MultiCube または Multi-InfoProvider
- ・ オペレーショナルデータストア(ODS)
- ・ インフォセット

関連項目

- ・ 437 ページの[データソースとしての SAP NetWeaver Business Warehouse\(BW\) InfoCube」](#)
- ・ 437 ページの[データソースとしての SAP NetWeaver BW クエリ」](#)
- ・ 438 ページの[推奨されるデータソースとしてのクエリ」](#)

8.1.2.1.1 データソースとしての SAP NetWeaver Business Warehouse(BW) InfoCube

OLSP ユニバースの作成に使用するデータソースとしては、次の種類のインフォキューブがサポートされています。

- ・ 標準およびトランザクション InfoCube: データおよびメタデータは、同じ SAP NetWeaver Business Warehouse(BW)システムに物理的に格納されます。
- ・ リモート InfoCube: データは、リモートシステムに物理的に格納されます。

注

リモートインフォキューブは完全にサポートされていますが、アドホッククエリ、レポートिंग、および分析を使用するシナリオでは、リモートインフォキューブでのユニバースの作成およびデプロイメントはお勧めできません。このようなアーキテクチャは、通常、対話型クエリで期待されるクエリパフォーマンスに応えることが期待できません。

- ・ MultiCube および Multi-InfoProvider

注

マルチキューブまたはマルチインフォプロバイダの最上位での Business Objects ユニバースの作成およびデプロイメントは、インフォキューブの最上位でのユニバースの作成およびデプロイメントと同じです。

インフォキューブ内のすべての特性、階層、キー値(時間、単位など)は、ユニバースに表示されます。

8.1.2.1.2 データソースとしての SAP NetWeaver BW クエリ

SAP NetWeaver BW ユーザは、BEx クエリを使用して SAP ビジネスエクスペローラのフロントエンドにアクセスします。

注

データソースとして機能し、OLAP インタフェースから Business Objects ユニバースを使用できるようにするには、OLAP の OLE DB に対して BEx クエリを解放する必要があります。[クエリプロパティ] ダイアログボックスの [拡張モード] タブで、SAP NetWeaver BW クエリデザイナの BEx クエリへの外部アクセスを許可します。

ユニバースには、行、列、および自由特性として選択された BEx クエリのすべての InfoObject が表示されます。これには、特性、階層、キー数値、構造、および変数が含まれます。

InfoSet およびオペレーショナルデータストア (ODS) は、どちらも BEx クエリ経由でユニバースに公開できません。

ODS に基づくクエリ

ODS は、BEx クエリ経由でユニバースに公開できます。

ODS オブジェクトは、多くの場合、インフォキューブに集計される前に、詳細なトランザクションレベルのデータ管理に使用されます。NetWeaver データストアの設計に ODS オブジェクトを含めると、InfoCube のサイズを最小化し、ロードおよびクエリパフォーマンスを向上できます。

注

ODS は、通常、大規模で詳細なリレーショナル構造です。OLAP BAPI インタフェースから ODS へアクセスしても、理想的なクエリパフォーマンスは得られません。迅速なレポート配布を求めるエンドユーザの期待に応えるために、次の代案を検討してください。

- ・ BAPI 呼び出しから ODS への直接アクセスを作成する
- ・ Web Intelligence でダイレクト SQL を使用して ODS テーブルにアクセスする

InfoSet に基づくクエリ

InfoSet は、BEx クエリ経由でユニバースに公開できます。

InfoSet は、SAP NetWeaver BW で定義され、マスタデータをレポートすることもあります。

注

マスタデータは、インフォキューブ上のユニバースを基にすることによってレポートできます。これによって InfoSet および BEx クエリを通過するための要件を回避できます。これら 2 つの方法の主な違いは、インフォキューブを使用せずにレポートされたマスタデータでは、有効なトランザクションにデータが制限されることです。

関連項目

- ・ 438 ページの[推奨されるデータソースとしてのクエリ](#)

8.1.2.1.3 推奨されるデータソースとしてのクエリ

BEx クエリは、次の理由から Business Objects ユニバースを生成するためのデータソースとして推奨されます。

- ・ 次の表に概要が示されているように、すべての SAP NetWeaver BW メタデータ機能を InfoCube レベルに取得できるわけではありません。

NetWeaver BW メタデータの機能	SAP OLAP Business Application Programming Interface(BAPI)のサポートレベル
特性(時間と単位を含む)	インフォキューブ/BEx クエリ
階層	インフォキューブ/BEx クエリ
基本的なキー数値	インフォキューブ/BEx クエリ
ナビゲーション属性	BEx クエリのみ
表示属性	インフォキューブ/BEx クエリ
計算キー数値/式	BEx クエリのみ

NetWeaver BW メタデータの機能	SAP OLAP Business Application Programming Interface(BAPI)のサポートレベル
制限キー数値	BEx クエリのみ
カスタム構造	BEx クエリのみ
変数	BEx クエリのみ

- ・ BEx クエリは、データモデリング環境を柔軟に拡張します。インフォキューブでの変更は困難です。
- ・ BEx クエリには、エンドユーザの要求を満たすカスタマイズデータソースを作成するために非常に便利な機能が用意されています。

BEx クエリにはデータソースとしての長所がありますが、すべてのレポートに BEx クエリが必要なわけではありません。また、既存の BW クエリごとにユニバースが必要なわけでもありません。メンテナンスコストを最小化するためには、すべてのアドホッククエリとレポートニングニーズを満たすために必要な BEx クエリとユニバースの最終的な数に実装戦略を限定します。必要なユニバースの数を削減するために、次の点に注意してください。

- ・ Web Intelligence がフロントエンドツールの場合、BEx クエリの出力書式によって制限されません。
- ・ 大規模な BEx クエリから作成した OLAP ユニバースを使用する場合、直接的なパフォーマンス上の影響はありません。Web Intelligence クエリに挿入されていない OLAP ユニバースオブジェクトは、クエリパフォーマンスに直接影響を及ぼしません。

注

アドホッククエリおよびレポートニングの範囲内の InfoCube または MultiCube ごとにいくつか (1 ～少数) の BEx クエリを用意することをお勧めします。次に、これらの各 BEx クエリの上にユニバースを作成します。

8.1.2.1.4 SAP NetWeaver BW の複数言語ユニバース

Web Intelligence では、SAP NetWeaver BW の複数言語機能を使用できます。複数言語環境を実装するには、NetWeaver BW システムに複数言語メタデータおよび複数言語データが含まれている必要があります。

ソリューションでサポートされる各言語のユニバースを作成する必要があります。ユニバース接続が作成された言語によって、ユニバースが生成される言語が決定されます。

ユーザの SAP 認証によって、クエリに返されるデータの言語が決定されます。ユーザは、SAP 認証を使用して InfoView にログインし、SAP サーバから返される結果について希望する言語を指定します。

結果セットの言語は、SAP の Unicode サポートに依存します。希望する言語でのデータが SAP システムに含まれていない場合、Web Intelligence ではこの言語でデータを使用できません。NetWeaver BW で説明が翻訳されない場合、Web Intelligence は、説明の代わりに技術名称の表示に戻します。

8.1.2.1.5 ユニバースデザインツールで SAP NetWeaver BW を使用するための前提条件

SAP NetWeaver BW データソースからユニバースを作成する場合、表示用に SSO(シングルサインオン)を有効にできます。SSO では、SAP 認証情報で SAP BusinessObjects Enterprise にログオンしたり、SAP 認証を活用したりできます。

SAP の最上位の OLAP ユニバースの SSO を有効にするには、SAP Integration をインストールして、SAP Security Plug In を設定する必要があります。

SAP のセキュリティ統合が設定されると、SAP 認証情報を使用してユニバースデザインツールを起動できます。BusinessObjects Enterprise のユーザ名を、SAP ユーザ ID とのセキュリティ統合を設定する際に定義した SAP システム ID と SAP クライアント ID を連結したものに設定します。

詳細については、『BusinessObjects XI Integration for SAP インストールガイド』および『BusinessObjects XI Integration for SAP ユーザガイド』を参照してください。

8.1.2.2 OLAP ユニバース向けにサポートされる MSAS 機能

次のテーブルには、MSAS データソースから生成されたユニバース向けの MSAS 機能に対するサポートレベルの概要が示されています。

MSAS メタデータ機能	OLAP ユニバースのサポートレベル
キューブ	サポートされます
ローカルキューブ	サポートされます
仮想キューブ(MSAS 2000)	サポートされます
パースペクティブ (MSAS 2000)	サポートされます
ディメンション	サポートされます
仮想ディメンション(MSAS 2000)	サポートされます
階層	サポートされます
レベル	サポートされます
レベルプロパティ	サポートされます
属性(MSAS 2005)	サポートされます
メジャー	サポートされます
メジャーグループ(MSAS 2005)	サポートされます
計算メジャー	サポートされます
表示フォルダ(MSAS 2005)	サポートされます
KPI (MSAS 2005)	サポートしていません

MSAS メタデータ機能	OLAP ユニバースのサポートレベル
アクション	サポートしていません
データベースの並べ替え順序	Web Intelligence でカスタム並べ替え順序を定義する必要あり
ライトバック	サポートしていません

関連項目

- 513 ページの [MSAS キューブをユニバースコンポーネントにマップする方法](#)」

8.1.2.3 OLAP ユニバースでサポートされる Essbase の機能

次の表では、Hyperion Essbase データソースから生成されたユニバースを対象として、Essbase 機能のサポートレベルを示します。

Essbase メタデータ機能	OLAP ユニバースのサポートレベル
ブロックストレージモード	サポートされます
集計ストレージモード	サポートされます
ハイブリッドモード	サポートしていません
エイリアステーブル	サポートされます
ディメンション	サポートされます
属性のディメンション	サポートされます
メンバーの複写	サポートされます
世代	サポートされます
レベル	サポートしていません
ユーザ定義属性(UDA)	サポートしていません
動的時間系列(DTS)	サポートしていません
Essbase Integration Services(EIS)のドリルスルー	サポートしていません
置換変数	サポートしていません
リンクされたパーティション	サポートしていません

Essbase メタデータ機能	OLAP ユニバースのサポートレベル
Linked Reporting Objects(LRO)	サポートしていません
データベースの並べ替え順序	Web Intelligence でカスタム並べ替え順序を定義する必要があります
ライトバック	サポートしていません

関連項目

- 512 ページの [Essbase キューブをユニバースコンポーネントにマップする方法](#)

8.2 OLAP データソースへの接続の定義

8.2.1 OLAP データソースへの接続

OLAP ユニバースを生成するには、最初に OLAP データソースへの接続を定義する必要があります。ユニバースの作成に使用するキューブまたはクエリごとに 1 つの接続を定義します。

新規接続ウィザードを使用して接続を定義します。このウィザードでは、接続の作成で次の手順を行います。

- 新規接続ウィザードを ユニバースデザインツールで起動します。
- 接続に名前を付け、データベースミドルウェアを選択する。
- 接続のログインパラメータを定義する。これらのパラメータは、選択したデータベースミドルウェアに応じて異なります。
- ユニバースを作成するために使用するキューブまたはクエリを選択する。
- 接続の期限を定義する。
- カスタムパラメータを定義する。これらのパラメータは、選択したデータベースミドルウェアに応じて異なります。

接続の定義は、OLAP ユニバースの作成で行う最初の手順です。接続を定義すると、ユニバースデザインツールで自動的にユニバースが生成されます。

注

[ツール]メニューの接続リストで接続を定義する場合、別の手順でユニバースを作成する必要があります。

関連項目

- 443 ページの [新規接続ウィザードを開始する](#)
- 443 ページの [OLAP 接続用のデータベースミドルウェアを選択する](#)
- 444 ページの [SAP BW OLAP 接続のログインパラメータ](#)

- ・ 446 ページの[MSAS OLAP 接続のログインパラメータ](#)」
- ・ 447 ページの[Essbase 接続のログインパラメータの定義](#)」
- ・ 448 ページの[OLAP 接続を行うためのソースキューブまたはクエリを選択する](#)」
- ・ 449 ページの[OLAP 接続用の設定パラメータを定義する](#)」
- ・ 450 ページの[Essbase 接続のカスタムパラメータの定義](#)」

8.2.2 新規接続ウィザードを開始する

新規接続ウィザードを開始するには、次のいずれかの操作を実行します。

起動方法	操作
[新規ユニバース]アイコン	[新規ユニバース]アイコンをクリックし、次に[ユニバースパラメータ]ボックスの[定義]ページにある[新規]ボタンをクリックします。
ファイルメニュー[ファイルメニュー]	空のセッションから、[ファイル] > [パラメータ]を選択し、次に[ユニバースのパラメータ]ボックスの[定義]ページにある[新規...]をクリックします。
クイックデザインウィザード	クイックデザインウィザードが有効に設定されている場合、ユニバースデザインツールを起動すると同時にクイックデザインウィザードも自動的に開始します。[新規]ボタンをクリックして、クイックデザインウィザードを開始します。 注 ウィザードが無効になっている場合は、[ファイル] > [新規作成]を選択します。[ファイル] > [新規作成]を選択してもクイックデザインウィザードが開始しない場合は、[ツール] > [オプション]を選択します。[オプション]ダイアログボックスの[全般]ページにて、[[ファイル]で[新規作成]を選択したらクイックデザインウィザードを起動]チェックボックスをオンにします。[OK]をクリックし、[ファイル] > [新規作成]を選択します。
ツールメニュー	[ツール] > [接続]を選択します。[ウィザード接続]ダイアログボックスで、[追加]をクリックします。

8.2.3 OLAP 接続用のデータベースミドルウェアを選択する

新規接続ウィザードの[データベースミドルウェアの選択]ページで、接続の名前を入力し、接続のタイプおよびデータベースミドルウェアを選択します。

[データベースミドルウェアの選択]パラメータ	説明
接続タイプ	<p>接続へのアクセスを制限するには、セキュリティ保護を選択します(推奨)。</p> <p>すべてのユーザによる接続へのアクセスを無制限に許可するには、共有を選択します。</p> <p>ユニバースの作成者にアクセスを限定するには、個人を選択します。ローカルコンピュータ上にある個人データにアクセスするには、個人接続を使用します。</p>
接続名	接続の名前を入力します。
ストアドプロシージャネットワークレイヤのフィルタ	ストアドプロシージャネットワークレイヤのフィルタパラメータは、OLAP 接続には使用されません。
利用可能なデータアクセスドライバの一覧です。	<p>このページは、使用しているデータアクセスドライバキーに対応したデータベースとミドルウェアの一覧が表示されます。</p> <p>ターゲットデータベースのノードを展開して、そのデータベース用としてサポートされているミドルウェアを表示します。</p> <p>ミドルウェアノードを展開して、OLAP ミドルウェアの Business Objects データアクセスドライバを表示します。</p> <p>データアクセスドライバを選択します。</p>

8.2.4 SAP BW OLAP 接続のログインパラメータ

[新しい接続] ウィザードの [ログインパラメータ] ダイアログボックスには、次のパラメータが含まれています。

ログインパラメータ	説明
認証モード	<ul style="list-style-type: none"> 指定したユーザ名とパスワードを使用する: ログインの詳細を認証として使用します。 BusinessObjects 認証マップを使用する: レポートを最新表示するには、BusinessObjects アカウントに関連付けられたデータベースユーザパスワードが要求されます。これは dbuser パラメータおよび dbpass パラメータを使用して設定します。これらは管理レベルで設定されます。このオプションの設定については、『BusinessObjects Enterprise 管理者ガイド』を参照してください。 表示時のレポートの最新表示にシングルサインオンを使用する: このオプションを選択すると、CMS へのアクセスに使用したユーザ名とパスワードが、自動的にデータベースへのログインパラメータとして使用されます。シングルサインオン (SSO) の設定については、『BusinessObjects Enterprise 管理者ガイド』を参照してください。
使用可能であれば SNC を使用します	SNC を使用する場合、このチェックボックスをオンにしてください。
クライアント	SAP NetWeaver BW システムのクライアントの識別に使用する番号 (必須)。
ユーザ名	[認証モード] が [指定したユーザ名とパスワードを使用する] の場合、OLAP サーバへのアクセスに使用するユーザ名。
パスワード	[認証モード] が [指定したユーザ名とパスワードを使用する] の場合、OLAP サーバへのアクセスに使用するパスワード。
言語	<p>接続に使用する言語。</p> <p>注 接続言語は、ユニバースを生成する言語を決定します。</p>
言語の保存	<p>次のうち接続に使用する言語。</p> <ul style="list-style-type: none"> [言語の保存] をオンにすると、[言語] フィールドの値が使用されます。 [言語の保存] をオフにすると、ユーザのセッションからの値が使用されます。
ログインモードまたはサーバの種類	<p>負荷分散を使用せずに SAP サーバに直接接続するには、[アプリケーションサーバ] を選択します。</p> <p>SAP 負荷分散機能の利点を活用するには、[メッセージサーバ] を選択します。</p>

ログインパラメータ	説明
アプリケーションサーバ	SAP アプリケーションサーバの名前または IP アドレスを選択するか、入力します。
システム番号	システム番号を入力します (たとえば、00)。アプリケーションサーバのログインモードには必須です。
システム ID	[メッセージサーバ] ログインモードを使用する場合は、[メッセージサーバ]と[ログオングループ]を入力し、必要な場合は[システム ID]も入力します。
ログオングループ	
メッセージサーバ	

8.2.5 MSAS OLAP 接続のログインパラメータ

[新しい接続] ウィザードの [ログインパラメータ] ダイアログボックスには、次のパラメータが含まれています。

ログインパラメータ	説明
認証モード	<ul style="list-style-type: none"> 指定したユーザ名とパスワードを使用する: ログインの詳細を認証として使用します。 BusinessObjects 認証マップを使用する: レポートを最新表示するには、BusinessObjects アカウントに関連付けられたデータベースユーザパスワードが要求されます。これは dbuser パラメータおよび dbpass パラメータを使用して設定します。これらは管理レベルで設定されます。このオプションの設定については、『SAP Business Objects Business Intelligence platform 管理者ガイド』を参照してください。 表示時のレポートの最新表示にシングルサインオンを使用する: このオプションを選択すると、CMS へのアクセスに使用したユーザ名とパスワードが、自動的にデータベースへのログインパラメータとして使用されます。シングルサインオン (SSO) の設定については、『SAP Business Objects Business Intelligence platform 管理者ガイド』を参照してください。

ログインパラメータ	説明
サーバ	<p>次のいずれかを入力します。</p> <ul style="list-style-type: none"> MSAS サーバで公開して設定した MSAS ライブラリの URL MSAS データソースのサーバ名 MSAS キューブファイルのフルパスファイル名。次のように、二重引用符で囲んだフルパスファイル名を入力します。 <p>"Z:¥All cubes¥test.cub"</p> <p>注 キューブファイルが SAP BusinessObjects ホスト以外のホストシステムに配置されている場合、SAP BusinessObjects ホストとそのホストは接続を共有している必要があります。キューブファイルへの接続は、SAP BusinessObjects ホスト上で直接作成する必要があります。</p>
ユーザ名	[認証モード] が [指定したユーザ名とパスワードを使用する] の場合、OLAP サーバへのアクセスに使用するユーザ名。
パスワード	[認証モード] が [指定したユーザ名とパスワードを使用する] の場合、OLAP サーバへのアクセスに使用するパスワード。
言語	接続に使用する言語。

8.2.6 Essbase 接続のログインパラメータの定義

新規接続ウィザードの[ログインパラメータ]ページでは、Essbase データベースへの接続のためにログインの詳細情報を指定します。

ログインパラメータ	説明
認証モード	<p>[指定したユーザ名とパスワードを使用する]を選択すると、ユーザは接続の使用時にログイン情報の入力を要求されるようになります。Essbase セキュリティと BusinessObjects Enterprise を同期する場合は、[ユーザ名]と[パスワード]に「Essbase DBuser」と「Essbase DBpass」を入力します。</p> <p>接続にユーザの BusinessObjects Enterprise のログイン認証情報を使用する場合は、[BusinessObjects 認証マップを使用する]を選択します。</p> <p>注 Essbase 接続では、[表示時のレポートの最新表示にシングルサインオンを使用する]オプションはサポートされません。</p>
ユーザ名	「Essbase Dbuser」を入力します。
パスワード	「Essbase DBpass」を入力します。
サーバ	Essbase サーバ名を入力します。

8.2.7 OLAP 接続を行うためのソースキューブまたはクエリを選択する

キューブブラウザには、ターゲットサーバで使用できる OLAP キューブが表示されます。

キューブノードを展開して、使用可能なキューブとクエリを表示します。ブラウザには、検索に役立つ次のツールがあります。

キューブブラウザツール	説明
お気に入り	<p>すばやくアクセスできるように、選択したキューブへのリンクを格納しておくフォルダです。キューブを[お気に入り]に追加するには、OLAP キューブブラウザ内でキューブを右クリックし、ショートカットメニューで[お気に入りに追加]を選択します。</p>

キューブブラウザツール	説明
検索	テキスト文字列で使用可能なキューブまたはクエリの名前を検索します。テキストボックスに文字列を入力し、[検索]をクリックします。検出された各インスタンスは強調表示されます。検索を続けるには、[検索]をクリックします。
\$INFOCUBE フォルダ	SAP NetWeaver BW のデータソースについては、InfoCube と MultiCube が \$INFOCUBE フォルダにグループ化されます。

ユニバースを作成するために使用するキューブまたはクエリを選択します。

8.2.8 OLAP 接続用の設定パラメータを定義する

新規接続ウィザードの[設定パラメータ]ページでは、接続持続時間を管理するために接続パラメータを定義します。接続を作成するときはデフォルトの設定を使用して、後からその設定を変更できます。

接続パラメータ	説明
接続プールモード	<p>接続持続時間は、[接続プールモード]および[プールタイムアウト]の各パラメータで定義できます。</p> <p>デフォルトでは [接続プールモード] は [接続維持時間] に設定され、[プールタイムアウト] はデフォルトの 10 分になります。</p> <p>注</p> <p>デフォルトの接続持続時間を維持することをお勧めします。トランザクションが終了するたびに [接続プールモード] が切断に設定される場合、ユニバース作成プロセスの速度が大幅に低下します。トランザクションごとに切断すると、階層的な値の一覧の操作などエンドユーザの主なワークフローにも影響を及ぼします。</p> <p>SAP NetWeaver BW を使用する場合、接続持続時間は大きな影響を及ぼします。</p> <p>しかし、接続持続時間は、BEx クエリの変更で既存のユニバースを更新する場合にも影響を及ぼすことがあります。これは、SAP NetWeaver BW への接続が確立されるたびに、OLAP BAPI インタフェースがクライアント側にメタデータキャッシュを作成するからです。このキャッシュは、接続が閉じられたときのみ空になります。</p> <p>SAP BEx クエリとメタデータキャッシュが非同期になるリスクを最小化するために、[プールタイムアウト] を 10 分から 1 分に変更できます。</p> <p>BW クエリの編集と新しいユニバースのこれらのクエリへのマッピングを並行して行う場合、ユニバース接続も閉じてメタデータキャッシュを空にするようにユニバースデザインツールを閉じてから、BEx クエリ側で行われたばかりの変更を考慮するために新しいユニバースを作成することをお勧めします。</p>
プールタイムアウト	
配列フェッチサイズ	
配列バインドサイズ	
ログインタイムアウト	[配列バインドサイズ] および [ログインタイムアウト] パラメータは、OLAP 接続では使用されません。

8.2.9 Essbase 接続のカスタムパラメータの定義

新規接続ウィザードの[カスタムパラメータ]ページでは、エイリアステーブルを指定し、ユニバースの生成時にメジャーディメンションとして使用するディメンションを選択します。

ログインパラメータ	説明
エイリアステーブル	デフォルト以外のエイリアステーブルでユニバースを生成するには、一覧からエイリアステーブルを選択します。
メジャーディメンション	メジャーディメンションとして使用するディメンションを選択します。ユニバースデザインツールは、ユニバースでメジャーとして選択したディメンションのメンバーを生成します。

8.3 OLAP ユニバースのカスタマイズ

8.3.1 追加パラメータを持つ OLAP ユニバースの作成

これは OLAP ユニバースのみに適用される機能で、MSAS、SAP NetWeaver Business Warehouse(BW)、または Essbase で OLAP ユニバースを作成する際に追加のメタデータパラメータを定義できます。

OLAP ユニバースを作成する際に、以下のパラメータを定義できます。

汎用 OLAP オプション	説明
詳細としての技術名の生成	ユニバースでディメンションの詳細オブジェクトとして技術名を生成するように、アプリケーションを設定できます。ユニバースの生成時に、技術名を示す詳細オブジェクトが作成されます。

SAP OLAP オプション	説明
メジャーの集計を依頼に設定	依頼されたデータベースにメジャーの集計関数を設定するように、アプリケーションを設定できます。
前置記号 L00、L01 などを置換	ユニバースレベル前置記号は、オブジェクトの階層でのレベルを示します。レベル L00 は最上位またはルートレベル、L01 は次の下位レベルです。[新規ユニバースウィザード]で、OLAP ユニバースレベル前置記号を別の前置記号と置換することができます。レベルの番号付けは維持されますが、たとえば、前置記号 L を Level に置換することができます。[新しい前置記号]フィールドにユーザ独自の前置記号を入力してください。この前置記号は、OLAP ユニバース全体のすべてのレベルに付け加えられます。
レベル 00 を ALL に名前変更	このオプションは、[レベル 00 の生成]オプションが[いいえ]に設定されている場合は無効になっています。ユニバースの次回生成時に、最上位レベル(ルートレベル)の L00 を ALL に名前変更することができます。
レベル 00 の生成	このオプションは SAP 特性にのみ適用されます。特性と階層に対してのみ、このオプションを無効化することができます。レベル 00 は、常に階層変数に対して生成されます。ユニバースを生成または更新する際にレベル番号(L00、L01、L02 など)を再生成できます。レベル番号は、レベル名に付加されます (Monthly Sales_L01 など)。これは、ALL レベルを使用してクエリーの結果を集計する Web Intelligence レポートに役立ちます。これにより、Web Intelligence レポートで集計フィールドを作成する必要がなくなります。

注

無効な [レベル 00 の生成] でユニバースが作成される場合、階層にルートレベルは生成されません。

8.3.2 OLAP ユニバースの OLAP オプションの定義

OLAP オプションを使用して、OLAP ソースから特定のユニバースメタデータを生成する方法を定義します。[オプション] ダイアログボックス([ツール] > [オプション] > [OLAP])の[OLAP]ページ から、OLAP オプションを選択します。選択したオプションに応じて、OLAP ソースの全コンテンツがユニバースで抽出され、作成されます。以下の OLAP ユニバース生成オプションを選択できます。

汎用 OLAP オプション	説明
詳細としての技術名の生成	ユニバースの技術名をプロパティとして生成するようにアプリケーションを設定できます。ユニバースの生成時に、技術名をポイントするオブジェクトが生成されます。

SAP OLAP オプション	説明
メジャーの集計を依頼に設定	集計関数を使用するメジャー用に、依頼されたメジャーを生成するようにアプリケーションを設定できます。ユニバースが生成される際、集計関数を使用するメジャーは依頼されたデータベースに設定されます。
前置記号 L00、L01 などを置換	ユニバースレベル前置記号は、オブジェクトの階層でのレベルを示します。レベル L00 は最上位またはルートレベル、L01 は次の下位レベルです。[新規ユニバースウィザード]で、OLAP ユニバースレベル前置記号を別の前置記号と置換することができます。レベルの番号付けは維持されますが、たとえば、前置記号 L を Level に置換することができます。[新しい前置記号]フィールドにユーザ独自の前置記号を入力してください。この前置記号は、OLAP ユニバース全体のすべてのレベルに付け加えられます。
レベル 00 を ALL に名前変更	レベル 00 生成オプションが[いいえ]に設定されている場合、このオプションは無効になっています。ユニバースが次に生成される際に、最上位レベル(ルートレベル)の L00 を ALL に名前変更できます。これは、ALL レベルを使用してクエリーの結果を集計する Web Intelligence レポートに役立ちます。これにより、Web Intelligence レポートで集計フィールドを作成する必要がなくなります。
レベル 00 の生成	このオプションは SAP 特性にのみ適用されます。特性に対してのみ、このオプションを無効化することができます。レベル 00 は、常に階層および階層変数に対して生成されます。 ユニバースを生成または更新する際にレベル番号(L00、L01、L02 など)を再生成できます。レベル番号は、レベル名に付け加えられます(Monthly Sales_L01 など)。

8.3.3 OLAP ユニバースのオブジェクトの定義

SQL エディタは、オブジェクトに対する SELECT 文または WHERE 句の定義や、OLAP ユニバースオブジェクトに対する MDX 演算子および関数の挿入に使用できます。このオプションおよび機能は、基礎となるデータベースに応じて SQL エディタで利用できます。

8.3.4 ユニバースデザインツールの機能は OLAP ユニバースでサポートされています。

OLAP ユニバースは、自動的に作成されます。OLAP ユニバースを作成すると、任意のユニバースコンポーネントを変更できます。

これらの ユニバースデザインツール機能は、生成された OLAP ユニバースでサポートされています。

- ・ クラスおよびオブジェクト(ディメンション、詳細、およびメジャー)の非表示、複製、名前変更
- ・ 新しいクラスおよびオブジェクト(ディメンション、詳細、およびメジャー)の挿入
- ・ オブジェクトの書式の編集
- ・ オブジェクトのデータタイプの編集
- ・ 一次キーと外部キーの定義
- ・ ディメンション、詳細、およびメジャーオブジェクトの MDX 構文の解析
- ・ ユニバースの整合性のチェック
- ・ 階層の編集
- ・ カスケード値の一覧の作成
- ・ 値一覧の依頼検索の定義とユーザによるクエリ実行時の値一覧のロード制限
- ・ 変数のデフォルト値の使用
- ・ 依頼されたデータベースの投影機能によるメジャーの定義(スマートメジャー)
- ・ ユニバース構造の更新

さらに、OLAP ユニバースでは次の独自機能を使用できます。

- ・ 計算メジャーの作成(SAP Netweaver BW および MSAS のみ)
- ・ 定義済み条件の作成
- ・ オプションプロンプトの定義

OLAP ユニバースに基づくすべてのオブジェクトは、キーを使用して生成されます。オブジェクトの階層に重複値が存在する場合、キーを使用して値の一覧内の不整合が解消されます。たとえば、1 回は親のフランスで、もう1 回は親のテキサスでというように、階層内でパリが 2 出現する場合を考えます。ユーザが、フランスのパリを選択すると、フランスのパリの行のみが返されます。

以下の ユニバースデザインツール機能は OLAP ユニバースでサポートされていません。

- ・ OLAP ユニバースでは、行レベルのセキュリティ認証を設定できません。
- ・ OLAP ユニバースでは、値の一覧を編集できません。
- ・ OLAP ユニバースにスキーマは生成されないで、ユニバースのエンティティ関係スキーマを表示、編集することはできません。

関連項目

- ・ 458 ページの[OLAP ユニバースの計算メジャー](#)
- ・ 461 ページの[OLAP ユニバースに含まれる定義済みの条件](#)
- ・ 466 ページの[OLAP ユニバースに含まれるオプションプロンプト](#)
- ・ 454 ページの[依頼されたデータベース投影関数](#)

8.3.5 依頼されたデータベース投影関数

ユニバースでは、メジャーに投影関数(Sum、Min、Max、Count、および Avg)を含めることはできません。投影関数は、レポートに表示されるディメンション数がクエリの結果セットに含まれるディメンション数よりも小さい場合、Web Intelligence でローカルにメジャーを集計するために使用されます。

比率、平均、重量などの非追加メジャーは、クエリの結果セットと同じ集計レベルに表示されるだけです。したがって、非追加メジャーの投影関数は、通常、ユニバースで[なし]に設定されます。

依頼されたデータベース投影関数を使用すると、非追加メジャーの集計をデータベースサーバに委任できます。これらは、Web Intelligence ではスマートメジャーと呼ばれています。スマートメジャーの投影関数は、オブジェクトのプロパティの [プロパティ] ページで [依頼されたデータベース] に設定されます。Web Intelligence でのこれらの関数およびその他の関数の使用方法について詳しくは、『Web Intelligence における式、関数、計算の使い方』の『スマートメジャーを使用した値の計算』の節を参照してください。

注

MSAS または Essbase データソースに基づく OLAP ユニバースについては、スマートメジャーはすべてユニバースに作成され、投影関数はデフォルトで依頼されたデータベースに設定されます。

注

アグリゲート認識セットを含むメジャーに基づくスマートメジャーを使用する場合、次の制約に留意してください。メジャー定義で使用される集計テーブルのデータに整合性がある、つまり集計値が詳細値に関して正確であることを確認することを強くお勧めします。データに整合性がない場合、スマートメジャーで不整合なデータが生成されることがあります。たとえば、スマートメジャーに対して年集計テーブルと日集計テーブルが使用される場合、過去の年については年集計テーブルと日集計テーブルに整合性があります。しかし、現在の年について日テーブルは日次ベースで正確ですが、年テーブルは空白である可能性があります。この場合、現在の年テーブルおよび日テーブルに基づくスマートメジャーを使用するレポートは、整合性のない結果をもたらす可能性があります。

例 スマートメジャー

この例では、クエリに 2 つのディメンション(“国”および“地域”)、3 つのメジャー(“注文金額”、“納品金額”、“納品割合”)が含まれています。

L01 地域	納品金額	注文金額	納品割合
Reg1	497,318,880	497,332,680	99.997
Reg2	199,463,776	199,466,536	99.998
Reg3	198,927,552	198,933,072	99.997
		合計:	299.992

納品割合の合計は、納品割合の列が合計されているので、誤りです。

このメジャーの投影関数が [依頼されたデータベース] にユニバースで設定されている場合、ユーザがレポートを最新表示した際に、Web Intelligence がデータベースに接続して正しい値を計算します。

L01 地域	納品金額	注文金額	納品割合
Reg1	497,318,880	497,332,680	99.997
Reg2	199,463,776	199,466,536	99.998
Reg3	198,927,552	198,933,072	99.997
		合計:	299.992
		合計:	99.997

注

Ratio 関数(Average)など一部の関数の使用には、注意が必要です。列から平均を計算する場合、この関数を正しく設定しないと予期しない計算が行われることがあります。

たとえば、 $\text{sum(Shop.facts.Margin)}/\text{sum(Shop.facts.Quantity_sold)}$ という SQL 関数では、予期しない結果が返されることがあります。正しく設定しなかった場合、各セルの平均が計算され、これらの平均の合計が返されます。修正するには、次の手順に従って関数のパラメータ化を実行する必要があります。

- 1 該当する関数の[プロパティの編集]に移動します。
- 2 [このメジャーは次の方法で集計されます]オプションで、[関数]ドロップダウンリストから[Db delegated]関数を選択します。
- 3 変更を保存します。

関連項目

- ・ 457 ページの[メジャーに集計機能を設定](#)

8.3.6 OLAP ユニバースの依頼されたメジャーの設定

集計関数を使用するメジャー用に、依頼されたメジャーを生成するようにアプリケーションを設定できます。ユニバースが生成される際に、集計関数を使用するメジャーは依頼されたデータベースに設定されます。

関連項目

- ・ 475 ページの[OLAP ユニバースレベル前記号の置換](#)
- ・ 474 ページの[OLAP ユニバースのレベル 00 の再生成](#)
- ・ 474 ページの[レベル L00 の ALL への名前変更](#)

8.3.7 メジャーに集計機能を設定

メジャーを作成する際は、集計関数をレポートにどのように投影するかを設定する必要があります。

メジャーオブジェクトが返す値は、クエリ処理の次の 2 つのレベルで集計されます。

- ・ クエリレベル: 生成される SELECT 文を使ってデータを集計します。
- ・ マイクロキューブからレポートのブロックレベル: データをマイクロキューブからレポート内のブロックに投影する際、データはマイクロキューブ内でローカル集計されます。

注

マイクロキューブとは、レポートに投影する前のクエリ結果の状態を表す概念で、BusinessObjects レポーティング製品がメモリ内に保持する取得データを意味します。ブロックレベルとは、取得したデータを使ってユーザが作成する平面(二次元)レポートを意味します。レポートの作成には、マイクロキューブ内のすべてのデータを使うこともできますし、一部のデータだけを使うこともできます。マイクロキューブ内のデータに集計関数を適用(ローカル集計)してレポート内に新しいデータを作成することもできます。

次の図は、クエリにおける 2 レベルの集計処理を図解したものです。

- ・ ユーザーは Web Intelligence でクエリーを作成する。
- ・ Web Intelligence がクエリーに対応する SQL を生成し、データベースに SELECT 文を送信する。
- ・ データがマイクロキューブに返される。これが第 1 の集計レベルです。
- ・ マイクロキューブが集計されたデータをレポートに投影する。データはクエリパネルでさらに再集計されます。これが第 2 の集計レベルです。

クエリを実行すると、SELECT 文の結果セットがマイクロキューブに保存され、マイクロキューブ内のすべてのデータがブロックに投影されます。データは、マイクロキューブに保存された最下位レベルに合わせて投影されますので、この時点では集計は行われません。

クエリパネルを使用してユーザがレポートを変更し、マイクロキューブの最下位レベルのデータを表示する必要がなくなると、データが上位レベルで集計されるよう、ローカル集計が行われます。

たとえば前ページの例では、ブロックから[年]オブジェクトを削除すると各年度の売り上げを表示する 3 行が不要となり、データがローカルで集計(この場合は合計)されてリゾートの通算売り上げが表示されます。

ローカル集計の種類は、メジャーの[プロパティを編集]ダイアログボックスの[プロパティ]ページで指定します。このページを開くには、オブジェクトを右クリック > [オブジェクトのプロパティ] > [プロパティ]タブをクリックします。

ローカル集計は、SELECT 集計とは異なりますのでご注意ください。

関連項目

- ・ 454 ページの[依頼されたデータベース投影関数](#)

8.3.8 OLAP ユニバースの計算メジャー

ユニバースでは、計算メジャーを作成してクエリを制限できます。OLAP ユニバースの計算メジャーは、非 OLAP ユニバースのメジャーオブジェクトの定義のようなものです。ただし、SQL ではなく、XML タグに埋め込まれた MDX 関数を使用して制限を定義します。

計算メジャーは、次の OLAP データソースで使用できます。

- ・ SAP NetWeaver Business Warehouse (BW)
- ・ MSAS 2000 および 2005

計算メジャーは、フィルタまたは WHERE 句で使用できます。

計算メジャー式の構文

計算メジャーの構文は、<EXPRESSION></EXPRESSION> タグに埋め込まれた計算で構成されます。

ユニバースデザインツール機能は、以下のような計算メジャー式で使用できます。

- ・ @Select
- ・ @Prompt
- ・ @Variable
- ・ @Where

注

計算メジャー式に @Aggregate_Aware 関数を含めることはできません。整合性のチェック機能は、XML 構文と、MDX 文に挿入されているものも含めた、上記に一覧されている @ 関数を検証します。ただし、MDX 文は解析されません。

"10"、"ABC" などの定数は使用できます。

計算メジャーは、次の OLAP メタデータを参照できます。

- ・ メジャー
- ・ デイメンション
- ・ デイメンションレベル
- ・ MDX 式

計算メジャー式に関する推奨事項

メジャー定義ではなく、@Select (メジャー名) を使用してください。これは、次の理由からです。

- ・ @Select は、クエリ時に解決されます。
- ・ 計算メジャーが @Select 関数内に含まれる場合、計算メジャーは、他の計算メジャーを参照できます。
- ・ @Select 関数内のオブジェクトの整合性は、チェックされます。

各オブジェクト定義について、キーを生成し、設定します。

オブジェクトへの参照、またはレベルや属性の一意の名前や技術名を参照する定義の詳細を使用します。

例 計算メジャー式

```
<EXPRESSION>@Select(Key Figures¥Order Amount)*@Select(Key Figures¥Order Quantity)</EXPRESSION>
```

関連項目

- 459 ページの[OLAP ユニバースで計算メジャーを作成する](#)

8.3.8.1 OLAP ユニバースで計算メジャーを作成する

OLAP ユニバースで計算メジャーを作成する手順は、次のとおりです。

- 1 ユニバースデザインツールで、OLAP ユニバースを開きます。
- 2 ユニバースに新しいメジャーオブジェクトを挿入します。
- 3 [WHERE 句]ボックスに、XML/MDX 式としてオブジェクトの定義を入力するか貼り付けます。
- 4 [解析]をクリックして、オブジェクトの定義を確認し、エラーがある場合は修正します。
- 5 [OK]をクリックしてオブジェクトの定義を保存します。
- 6 [ツール] > [整合性のチェック]の順に選択します。

整合性チェックは、XML 構文と、ユニバースデザインツール @FUNCTIONS を検証します。

関連項目

- 458 ページの[OLAP ユニバースの計算メジャー](#)

8.3.9 キューブクエリの MDX 関数について

MDX エディタを使用して、キューブクエリを定義します。

新規オブジェクトまたは定義済みのフィルタを OLAP ユニバースに追加する際に、特定のデータソース接続に対応した、サポートされている MDX 式の一覧を使用できます。

使用できる式のライブラリが、.prm 接続ファイルに保存されています。オブジェクトの[プロパティの編集]ペインを開き、クエリの[SELECT 句の編集]ペインを開くと、使用できる式が[関数]ペインに表示されます。SELECT 句または WHERE 句に式を挿入するには、対象の句を、式を挿入する位置でクリックし、適切な式をダブルクリックします。

OLAP ユニバース MDX 辞書 - 関数一覧(PRM ファイル)

新規のオブジェクトまたは定義済みのフィルタを OLAP ユニバースに追加する際に、式で使用する MDX 関数(主にメンバー関数)と MDX 演算子の明示的な一覧が、適切な OLAP 接続(SAP または MSAS)のオブジェ

クエリエディタおよびフィルタエディタで使用可能になります。SAP または mySQL(sap.prm、sqlsrv_as.prm)の接続の設定方法については、『データアクセスガイド』を参照してください。使用できる関数と演算子は、ユニバースの接続の種類によって異なります。この関数一覧は、接続ごとに PRM ファイルで提供されます。サポートされている全関数の一覧が提供されるのではなく、特に頻繁に使用される関数のみが提供されます。

以下の MDX 演算子がクエリで使用できます。

- ・ Equal
- ・ NotEqual
- ・ InList
- ・ NotInList
- ・ Greater
- ・ GreaterOrEqual
- ・ Less
- ・ LessOrEqual
- ・ Between
- ・ NotBetween
- ・ Like
- ・ NotLike

以下の一覧は、条件の編集に使用できる MDX フォルダ関数の例です。使用できる関数は、基盤となるデータベースによって異なります。

- ・ 設定関数(ADDCALCULATEDMEMBERS, ALLMEMBERSなど)
- ・ 統計/数値関数(AGGREGATE, AVG など)
- ・ ナビゲーション/メンバー関数(ANCESTOR, ASCENDANTS など)
- ・ メタデータ関数(AXIS, HIERARCHY など)

8.3.10 フィルタと WHERE 文のための XML 構文

この節では、WHERE 句またはフィルタ文を OLAP ユニバースで定義するための XML 構文について説明します。FILTER または FILTER EXPRESSION タグを手動で追加し、手動またはユニバースデザインツールの MDX エディタを使用してタグの間に式を入力します。

- ・ 単一のオブジェクト定義を使用する場合は、<FILTER= "your_object_definition"> を使用します。オブジェクト定義を二重引用句で囲みます。
- ・ 1 つ以上のオブジェクトを含む複雑な MDX 式を使用する場合は、<FILTER EXPRESSION= "yourcomplexMDX_expression"> を使用します。式を二重引用句で囲みます。

単一フィルタオブジェクトのための構文は次のとおりです。

```
<FILTER = "your_object_definition"><CONDITION OPERATORCONDITION="yourOperator"><CONSTANT VALUE="your_Value"/></CONDITION></FILTER>
```

この場合は、以下のとおりです。

- ・ yourMDX_expression は、二重引用句で囲んだ単一オブジェクト定義です。

- ・ CONSTANTVALUE は、CONSTANT CAPTION または CONSTANT TECH_NAME のいずれかです。
- ・ yourOperator は、フィルタ式演算子(equals、inlistなど)です。Inlist 演算子を使用する場合は、CONSTANT CAPTION、または CONSTANT TECH_NAME 要素を、リスト内の各アイテムごとに挿入する必要があります。
- ・ your_Value は、CONSTANT CAPTION を使用している場合は定義済みフィルタ値、または CONSTANT TECH_NAME を使用している場合はオブジェクト識別子です。

3 つの国がリストにあるとします。InList 演算子を使用している単一フィルタオブジェクトの構文は、以下のようになります。

```
<FILTER= "your_object_definition "><CONDITION OPERATORCONDITION="InList"><CONSTANT CAPTION="英国"/><CONSTANT CAPTION="フランス"/><CONSTANT CAPTION="ドイツ"/></CONDITION></FILTER>
```

複雑なフィルタ式のための構文およびフィルタされた値の TECH_NAME は次のとおりです。

```
<FILTER EXPRESSION="yourComplex_MDX_Expression"><CONDITION OPERATORCONDITION="Equal"><CONSTANT TECH_NAME="1"/></CONDITION></FILTER>
```

例 フィルタ式の計算されたメンバーを使用したフィルタ

```
<FILTER EXPRESSION="IIF([0CALYEAR].CurrentMember > "2000", 1,0)"><CONDITION OPERATORCONDITION="Equal"><CONSTANT CAPTION="1"/></CONDITION></FILTER>
```

8.3.11 OLAP ユニバースに含まれる定義済みの条件

OLAP ユニバースに含まれる定義済みの条件は、非 OLAP ユニバースに含まれる条件と同様です。ただし、SQL ではなく XML を使用して WHERE 句を定義します。フィルタは手動で、または定義済みフィルタエディタを使用して宣言できます。

8.3.11.1 定義済みフィルタオプションの XML 構文

定義済み条件の構文

1 つの定義済み条件には、AND 演算子および OR 演算子で結合された複数のフィルタが含まれる場合があります。デフォルトでは、すべてのフィルタは AND 演算子で結合されます。OR を使用してフィルタを含めるには、AND タグと OR タグを使用する必要があります。

定義済みのフィルタの定義では、@Select、@Prompt、および @Variable の各関数を使用できます。

定義済みのフィルタには、1 つまたは複数のプロンプトを設定できます。プロンプトには、必須のものとオプションのものがあります。

例 定義済み条件における AND タグと OR タグの使い方

```
<OPERATOR VALUE="AND">
  <FILTER "[Level Object definition]">
    <CONDITION OPERATORCONDITION="Operator">
      <CONSTANT Level Attribute="Value"/>
    </CONDITION>
  </FILTER>
  <OPERATOR VALUE="OR">
    <FILTER "[Level Object definition]">
      <CONDITION OPERATORCONDITION="Operator">
        <CONSTANT Level Attribute="Value"/>
      </CONDITION>
    </FILTER>
    <FILTER "[Level Object definition]">
      <CONDITION OPERATORCONDITION="Operator">
        <CONSTANT Level Attribute="Value"/>
      </CONDITION>
    </FILTER>
  </OPERATOR>
</OPERATOR>
```

8.3.11.2 OLAP ユニバースでの定義済み条件の手動作成

定義済み条件を作成する

- 1 ユニバースデザインツールで、OLAP ユニバースを開き、[ユニバース] 枠の下部にある条件ラジオボタンをクリックします。
[ユニバース] 枠の条件ビューが表示されます。ユニバースのクラスがツリービューで表示されます。
- 2 クラスを右クリックし、コンテキストメニューから、[条件] を選択します。
- 3 [Where 句] ボックスで、XML テンプレートフィルタを編集します。

テンプレートフィルタの書式は、次のとおりです。

```
<FILTER "[Level Object definition]">
  <CONDITION OPERATORCONDITION="Operator">
    <CONSTANT Level Attribute="Value"/>
  </CONDITION>
</FILTER>
```

テンプレートの要素を次のように変更します。

テンプレート要素	指定できる値
Level Object definition	フィルタで使用するディメンションレベルまたはメジャーを入力します。オブジェクト名ではなく、オブジェクト定義を入力します。
Operator	次のいずれかを入力します。 <ul style="list-style-type: none"> Equal NotEqual Greater Less GreaterOrEqual LessOrEqual Between NotBetween InList NotInList Like NotLike
Level Attribute	次のいずれかを入力します。 <ul style="list-style-type: none"> NAME CAPTION TECH_NAME DESCRIPTION
Value	値またはプロンプトを入力します。CONSTANT タグごとに 1 つの値を定義します。

編集された定義済み条件の例

```

<FILTER KEY="[0D_DIV].[LEVEL01]">
  <CONDITION OPERATORCONDITION="InList">
    <CONSTANT CAPTION="Internal"/>
    <CONSTANT CAPTION="Service"/>
  </CONDITION>
</FILTER>

```

- 4 [解析] をクリックして、構文を確認し、エラーがある場合は修正します。
- 5 [OK] をクリックして条件を保存します。

関連項目

- 461 ページの[OLAP ユニバースに含まれる定義済みの条件](#)
- 466 ページの[OLAP ユニバースに含まれるオプションプロンプト](#)

8.3.11.3 定義済みフィルタエディタについて

定義済みフィルタエディタは、OLAP ユニバースで定義済みのフィルタを編集するためのものです。フィルタエディタを使用して、オブジェクト、演算子、値の一覧、プロンプト、関数、および OLAP ユニバース用のフィルタの定義に使用できる他のオプション要素を選択することができます。

フィルタの条件プロパティパネルで、フィルタ式を手動で入力するか、[>>] をクリックして定義済みフィルタエディタを開きます。エディタが開いたら、フィルタ式に @Prompt を挿入します。フィルタ式の適切な箇所を右クリックして、ショートカットメニューから、[@Prompt の新規作成] を選択します。定義済みフィルタエディタにより、クエリ/オブジェクト定義にフィルタ式が挿入されます。

例 国をカナダに制限するための国レベルでの“顧客”ディメンションの制限

```
<FILTER KEY="[顧客].[国].[国]"> <CONDITION OPERATORCONDITION="Equal"> <CONSTANT
CAPTION="カナダ" /> </CONDITION> </FILTER>
```

関連項目

- ・ 464 ページの[定義済みフィルタエディタのオプションについて](#)
- ・ 465 ページの[定義済みフィルタエディタを使用した定義済みフィルタの編集](#)
- ・ 459 ページの[キューブクエリの MDX 関数について](#)

8.3.11.4 定義済みフィルタエディタのオプションについて

定義済みフィルタエディタを使用すると、OLAP ユニバース用のユニバースフィルタを簡単に定義できます。次のオプションを選択できます。

オプション	説明
演算子を選択	使用可能な一覧から演算子を選択します。デフォルトは、[Equal] です。
フィルタ基準の指定	既存のユニバースオブジェクトまたはフリー定義に基づいてフィルタします (例: [メジャー].[インターネット売上金額])。デフォルトは、[ユニバースオブジェクト] です。
LoV の選択	既存のオブジェクトに基づいてフィルタする際に、現在のユニバースのオブジェクト一覧を選択します。デフォルト選択は、オブジェクト一覧のルートクラスです。
比較値	オブジェクト/式を比較する値を定義します。選択した演算子に応じて、入力する値セットが 1 つまたは 2 つあります。値は、静的な値の場合と、プロンプトに基づく値場合があります。デフォルトは、[静的な値] です。

オプション	説明
プロンプトの追加	プロンプトを手動で編集するか、@Prompt エディタを使用します。[>>] をクリックして、@Prompt エディタを開きます。
インデックス認識の設定	インデックス認識関数を有効化します。これを正常に機能させるには、一次キーを宣言する必要があります。インデックス認識がユニバースデザインツールに設定されている場合、一次キー列と外部キー列を使用してすばやくデータを取得し、ユニバースデザインツールで SQL フィルタを効率的に生成することができます。デフォルトは、未選択です。
計算式を使用	選択すると、<EXPRESSION> </EXPRESSION> タグ内のフィルタ式が括弧で囲まれます。デフォルトは、未選択です。
省略可能	現在のフィルタ式をオプションとして設定します。これは、フィルタエディタの現在のフィルタ式にのみ適用され、定義済みの条件オブジェクト全体には適用されません。デフォルトは、未選択です。

注

“オプション” タグは、Web Intelligence の事前定義済みフィルタには使用できません。これらのタグが使用されると、クエリの必須部分として処理されるため、クエリが実行されません。

関連項目

- 465 ページの [定義済みフィルタエディタを使用した定義済みフィルタの編集](#)

8.3.11.5 定義済みフィルタエディタを使用した定義済みフィルタの編集

OLAP ユニバースでフィルタを編集します。

定義済みフィルタエディタは、値を選択または入力したときに更新されます。フィルタ式を右クリックして、フィルタ式に @Prompt 式を挿入できます。右クリックし、[@Prompt の新規作成] を選択すると、プロンプト エディタが開きます。

- 条件 (フィルタ) 枠のプロパティ枠で、[>>] をクリックします。
定義済みフィルタエディタが表示されます。
- フィルタをユニバースオブジェクトに基づかせるには、[ユニバースオブジェクト] を選択し、[使用できるオブジェクト] 枠からオブジェクトを選択します。作成した式を定義済みフィルタで使用するには、[フリー定義] を選択し、[使用できるオブジェクト] 枠に式を入力します。
- [演算子] の一覧から演算子を選択します。In List および Not In List 演算子についてのみ、複数の値 (右オペランド) を指定できます。
- [静的な値] を選択して 1 つ以上の固定値を定義するか、[プロンプト] を選択してプロンプト式を挿入します。

[プロンプト]を選択すると[編集]ボタンが有効になります。[編集]をクリックして @Prompt エディタを開き、必要に応じてプロンプト式を定義します。

- 5 [OK] をクリックして、フィルタの定義を検証します。

パーサーにより、構文エラーがチェックされます。整合性についてもチェックされます。エラーが検出されると、警告メッセージがエラーメッセージとともに表示されます。エラーが検出されなかった場合は、新しい条件オブジェクトがフィルタ定義とともにユニバースに追加されます。

関連項目

- ・ 464 ページの[定義済みフィルタエディタのオプションについて](#)
- ・ 464 ページの[定義済みフィルタエディタについて](#)

8.3.12 OLAP ユニバースに含まれるオプションプロンプト

OLAP データソースから生成されたユニバースでは、オプションプロンプトがサポートされます。

SAP NetWeaver BW オプション変数では、オプション条件付きのフィルタがユニバースで自動的に生成されます。

定義済みの条件またはオブジェクトの WHERE 句でプロンプトをオプションにするには、2 つの XML タグ <OPTIONAL> と </OPTIONAL> との間に XML フィルタ式を埋め込みます。

例 定義済み条件に含まれるオプションプロンプト

```
<OPTIONAL>
  <FILTER KEY="[Products].[Family]" >
    <CONDITION OPERATORCONDITION="InList" >
      <CONSTANT CAPTION="@prompt('Enter value(s) for Product family:','A','Products¥Family',Multi,primary_key,persistent)"/>
    </CONDITION>
  </FILTER>
</OPTIONAL>
```

関連項目

- ・ 462 ページの[OLAP ユニバースでの定義済み条件の手動作成](#)

8.3.13 SAP NetWeaver BW ユニバースに対する特定のクエリのパフォーマンスを向上させる

ディメンションのキーおよび中間名の詳細オブジェクトだけを含む、SAP NetWeaver BW ユニバースに対するクエリについては、生成されているオブジェクトの構文を変更してクエリパフォーマンスを向上させることができます。

構文を変更するには、次の操作を行います。

- 1 ユニバースデザインツールでユニバースを開きます。
- 2 変更するキー詳細オブジェクトをダブルクリックします。
- 3 [プロパティの編集]ダイアログボックスの[定義]タブにある[SELECT 句]テキストボックスで、SAP 特性の NAME 属性を参照するように構文を変更します。

たとえば、L01 Customer Key というオブジェクトの場合、次の SELECT 構文が生成されています。

```
[Z_CUSTOM].[LEVEL01].[2Z_CUSTOM].[Value]
```

この構文を、NAME 属性を参照するように変更します。

```
[Z_CUSTOM].[LEVEL01].[NAME]
```

- 4 [OK]をクリックして変更を保存します。
- 5 名前オブジェクトに対して、同じ手順を繰り返します。SAP 特性の DESCRIPTION 属性を参照するように、構文を変更します。

たとえば、L01 Customer Medium Name というオブジェクトの場合、次の SELECT 構文が生成されています。

```
[Z_CUSTOM].[LEVEL01].[5Z_CUSTOM].[Value]
```

この構文を、DESCRIPTION 属性を参照するように変更します。

```
[Z_CUSTOM].[LEVEL01].[DESCRIPTION]
```

8.4 OLAP ユニバースのライフサイクル管理

8.4.1 OLAP ユニバースのライフサイクル管理

注

XIR3.1 SP2 より前のバージョンの Universe Designer で作成したユニバースを開く場合は、ユニバースまたは OLAP ソースで変更を加える前にユニバースを最新表示して保存する必要があります。

OLAP ユニバースは、SAP NetWeaver BEx クエリや MSAS 2005 キューブなどの OLAP データソースから自動的に生成されます。ユニバースデザインツールでは、既存の OLAP ユニバースのオブジェクトを作成し、変更できます。

[OLAP ユニバースの更新]ウィザードでは、OLAP データソースに加えられた変更に基づいて、OLAP ユニバースの構造を自動的に更新できます。このウィザードは、ユニバースと更新されたデータソースとを比較します。このウィザードは、生成されたオブジェクトと手動で追加または変更されたオブジェクトを区別できます。このため、ユニバースデザインツール内での手動による変更を保存できます。ウィザードでは、ユニバースデザインツールで手動で追加されたオブジェクトは更新されません。

次の表に示すとおり、何を検出でき、更新できるかは、項目とデータソースによって異なります。

ウィザードが検出できる項目	新しい項目を検出できる場所	変更された項目を検出できる場所	削除された項目を検出できる場所
ディメンション	すべてのデータソース	すべてのデータソース	すべてのデータソース
階層	SAP NetWeaver BW および MSAS のみ	すべてのデータソース	すべてのデータソース
レベル	すべてのデータソース	すべてのデータソース	すべてのデータソース
プロパティ	MSAS のみ	MSAS のみ	MSAS のみ
メジャー	すべてのデータソース	すべてのデータソース	すべてのデータソース
SAP NetWeaver BW 変数	SAP NetWeaver BW のみ	SAP NetWeaver BW のみ	SAP NetWeaver BW のみ
サブクラス	すべてのデータソース	すべてのデータソース	すべてのデータソース

注

Universe Designer の XIR3.1 SP2 よりも前のバージョンを使用して作成されたユニバースを更新する際、ディメンション名が SAP キューブで変更されている場合は、ディメンションの最新表示は機能しません。ディメンションがユニバースで重複しているためです。ユニバースのクラスを手動で更新する必要があります。

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)」
- ・ 468 ページの[概要: ユニバースオブジェクトのステータスと OLAP オブジェクトのステータスの関係](#)」
- ・ 476 ページの[OLAP ユニバースの更新におけるディメンションの管理方法](#)」
- ・ 481 ページの[OLAP ユニバースの更新における階層または特性の管理方法](#)」
- ・ 488 ページの[OLAP ユニバースの更新におけるレベルの管理方法](#)」
- ・ 492 ページの[OLAP ユニバースの更新における SAP 変数の管理方法](#)」
- ・ 496 ページの[OLAP ユニバースの更新におけるキー数値またはメジャーの管理方法](#)」
- ・ 501 ページの[OLAP ユニバースの更新における SAP キー日付の管理方法](#)」

8.4.2 概要: ユニバースオブジェクトのステータスと OLAP オブジェクトのステータスの関係

以下の表は、SAP OLAP オブジェクトのステータスとユニバースオブジェクトのステータスの関係の概要について簡単に示したものです。ほとんどのアクションには固有の注があります。詳しくは、この章の詳細な節を参照してください。

OLAP メタデータ	ユニバースオブジェクトのステータス				
	変更なし	更新状況*	削除	移動	非表示
ディメンション	ユニバースの "クラス"				

OLAP メタデータ		ユニバースオブジェクトのステータス				
		変更なし	更新状況*	削除	移動	非表示
ステータス	変更なし	NoC	Upd	NoC	NoC	NoC
	更新状況*	Upd	Upd	NoC	Upd	Upd
	削除	Del/Ob	Del/Ob	NoC	Del/Ob	NoC
	移動	Move	NoC	NoC	NoC	Move
	特性作成済み	CreS	CreS	N/A	CreS	CreS
	作成	Cre	Cre	N/A	Cre	Cre
階層または特性		ユニバースの“サブクラス”				
ステータス	変更なし	NoC	Upd	NoC	NoC	NoC
	更新状況*	Upd	Upd	NoC	Upd	Upd
	変更済み	UpdMDX	UpdMDX	NoC	UpdMDX	UpdMDX
	表示属性	Cre	Cre	Cre	Cre	Cre
	ナビゲーション属性	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	削除	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	移動	Move	Move	NoC	Move	Move
	新規	Cre	Cre	Cre	Cre	Cre
レベル		ユニバースの“レベル”				
ステータス	変更なし	NoC	NoC	NoC	NoC	NoC
	更新状況*	Upd	Upd	NoC	Upd	Upd
	削除	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	移動	Move	Move	NoC	Move	Move
	新規	Cre	Cre	Cre	Cre	Cre
変数		ユニバースの“フィルタ”				

OLAP メタデータ		ユニバースオブジェクトのステータス				
		変更なし	更新状況*	削除	移動	非表示
ステータス	変更なし	NoC	NoC	NoC	NoC	NoC
	更新状況*	Upd	Upd	Cre	Upd	Upd
	削除	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	新規	Cre	Cre	Cre	Cre	Cre
キー数値		ユニバースの “メジャー”				
ステータス	変更なし	NoC	NoC	NoC	NoC	NoC
	更新状況*	Upd	Upd	NoC	Upd	Upd
	削除	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	移動	Move	Move	NoC	Move	Move
	新規	Cre	Cre	Cre	Cre	Cre
キー日付		ユニバースの “パラメータ”				
ステータス	変更なし	NoC	N/A	Cre	N/A	N/A
	削除	Del	N/A	N/A	N/A	N/A
	新規	Cre	N/A	Cre	N/A	N/A

凡例

- ・ *: オブジェクトプロパティ(名前、説明など)の 1 つが変更されました
- ・ Cre: 同等のオブジェクトを作成します
- ・ CreS: 同等のサブクラスオブジェクトを作成します
- ・ Del/Ob: 削除されているか、使用されていません(使用されていないオブジェクトは非表示になり、名前の先頭に ## が付けられています)
- ・ Move: オブジェクトは移動されています
- ・ N/A: 適用されません
- ・ NoC: 変更がありません
- ・ Upd: 更新されています
- ・ UpdMDX: MDX 定義を更新します

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)」
- ・ 476 ページの[OLAP ユニバースの更新におけるディメンションの管理方法](#)」
- ・ 481 ページの[OLAP ユニバースの更新における階層または特性の管理方法](#)」
- ・ 488 ページの[OLAP ユニバースの更新におけるレベルの管理方法](#)」

- ・ 492 ページの[OLAP ユニバースの更新における SAP 変数の管理方法](#)」
- ・ 496 ページの[OLAP ユニバースの更新におけるキー数値またはメジャーの管理方法](#)」
- ・ 501 ページの[OLAP ユニバースの更新における SAP キー日付の管理方法](#)」

8.4.3 OLAP ユニバースを最新表示する

OLAP ユニバース構造の最新表示を行うには。

- ・ ユニバースデザインツールで、更新するユニバースを開きます。
- ・ [表示] > [構造の最新表示]を選択します。
OLAP ユニバースの更新ウィザードが表示されます。
- ・ [開始]をクリックします。

更新済みのメタデータ定義を示すページが表示されます。

手動で行われたユニバース内のオブジェクトへの変更を維持する場合は、保持オプションを選択します。デフォルトでは、すべての保持オプションが選択されています。次の属性の保持を選択できます。

オプション	説明
ビジネス名を保持	クラス名、ディメンション名、メジャー名、詳細名、および条件名。
オブジェクトの種類を保持	ユニバースのオブジェクトが変更された場合(たとえば、詳細がディメンションに変更され、最初のオブジェクトタイプが更新により再導入されない場合)。これに関係するのは、ディメンション、メジャー、および詳細です。
オブジェクトの説明を保持	これが選択されていて、説明が OLAP ソースで更新される場合は、ユニバースはこの情報では更新されません。
オブジェクトのデータ型を保持する	文字、数値、日付、および可変長文字列のテキストオブジェクト。
オブジェクトの値の一覧のオプションを保持	最初に設定したオプションを保持できます。 <ul style="list-style-type: none"> ・ 値の一覧の関連付け ・ 使用前に自動更新 ・ 階層表示 ・ ユニバースと一緒にエクスポート ・ 依頼検索

オプション	説明
オブジェクトの詳細オプションを保持	オプションは次のとおりです。 セキュリティアクセスレベル オブジェクトは、以下で使用できます。 <ul style="list-style-type: none">・ 結果で使用・ 条件で使用・ ソートで使用
使用されていないオブジェクトを削除する	データソースに存在しない項目がユニバースから削除されます。
使用されていないオブジェクトを非表示にする	キューブに存在しない項目がユニバースで非表示にされ、先頭に /##/ が付けられます。

- ・ 必要なオプションを選択して、[次へ]をクリックします。

[管理結果の変更]ページが表示され、追加、削除、または非表示の各オブジェクトが表示されます。非表示のオブジェクトはユニバース内の別のクラスに移動され、先頭に /##/ の付いた斜体で表示されます。

- ・ 追加されたメタデータオプションパネルで、追加されたメタデータの処理方法に関するオプションを設定します。

汎用 OLAP オプション	説明
詳細としての技術名の生成	ユニバースの技術名をプロパティとして生成するようにアプリケーションを設定できます。ユニバースの生成時に、技術名をポイントするオブジェクトが生成されます。
手動で削除されたすべてのオブジェクトが再生成されました	手動で削除されたユニバースオブジェクトが、再生成されます。

SAP OLAP オプション	説明
メジャーの集計を依頼に設定	依頼されたデータベースにメジャーの集計関数を設定するように、アプリケーションを設定できます。
前置記号 L00、L01 などを置換	ユニバースレベル前置記号は、オブジェクトの階層でのレベルを示します。レベル L00 は最上位またはルートレベル、L01 は次の下位レベルです。[新規ユニバースウィザード]で、OLAP ユニバースレベル前置記号を別の前置記号と置換することができます。レベルの番号付けは維持されますが、たとえば、前置記号 L を Level に置換することができます。[新しい前置記号]フィールドにユーザ独自の前置記号を入力してください。この前置記号は、OLAP ユニバース全体のすべてのレベルに付け加えられます。
レベル 00 を ALL に名前変更	このオプションは、[レベル 00 の生成]オプションが[いいえ]に設定されている場合は無効になっています。ユニバースの次回生成時に、最上位レベル(ルートレベル)の L00 を ALL に名前変更することができます。
レベル 00 の生成	このオプションは SAP 特性にのみ適用されます。特性に対してのみ、このオプションを無効化することができます。レベル 00 は、常に階層および階層変数に対して生成されます。ユニバースを生成または更新する際にレベル番号(L00、L01、L02 など)を再生成できます。レベル番号は、レベル名に付加されます (Monthly Sales_L01 など)。これは、ALL レベルを使用してクエリーの結果を集計する Web Intelligence レポートに役立ちます。これにより、Web Intelligence レポートで集計フィールドを作成する必要がなくなります。

- ・ [管理結果の変更]ページで、次のいずれかのオプションを選択します。

オプション	説明
OK	結果に問題がなければ、[OK]をクリックして、ユニバースを保存したり、エクスポートしたりせずに閉じることができます。
エクスポート	変更内容に問題がなければ、[エクスポート]をクリックして保存し、更新したユニバースを CMS にエクスポートします。
整合性のチェック	整合性のチェックを行うには、[整合性のチェック]をクリックします。構造のチェック、オブジェクトの解析、結合の解析、条件の解析、およびカーディナリティのチェックが行われます。チェックが終了すると、[整合性のチェック結果]ページが表示されます。チェックの結果は、このページから印刷できます。

表示されるはずのユニバースに対する変更がすべて表示されていない場合は、ユニバースデザインツールを終了、再起動してからもう一度更新してみます。これによって、データソースへの新しい接続が確立し、キャッシュがクリアされます。

関連項目

- ・ 475 ページの[ユニバースと OLAP キューブの同期化](#)
- ・ 467 ページの[OLAP ユニバースのライフサイクル管理](#)

8.4.4 OLAP ユニバースのレベル 00 の再生成

ユニバースを生成または更新する際にレベル番号(L00、L01、L02 など)を再生成できます。レベル番号は、レベル名に付け加えられます(Monthly Sales_L01 など)。

関連項目

- ・ 456 ページの[OLAP ユニバースの依頼されたメジャーの設定](#)
- ・ 475 ページの[OLAP ユニバースレベル前記号の置換](#)
- ・ 474 ページの[レベル L00 の ALL への名前変更](#)

8.4.5 レベル L00 の ALL への名前変更

ユニバースが次に生成される際に、最上位レベル(ルートレベル)の L00 を ALL に名前変更できます。これは、ALL レベルを使用してクエリの結果を集計する SAP BusinessObjects Web Intelligence レポートに役立ちます。これにより、Web Intelligence レポートで集計フィールドを作成する必要がなくなります。

関連項目

- ・ 456 ページの[OLAP ユニバースの依頼されたメジャーの設定](#)

- ・ 475 ページの[OLAP ユニバースレベル前記号の置換](#)
- ・ 474 ページの[OLAP ユニバースのレベル 00 の再生成](#)

8.4.6 OLAP ユニバースレベル前記号の置換

ユニバースレベル前置記号は、オブジェクトの階層でのレベルを示します。レベル L00 は最上位またはルートレベル、L01 は次の下位レベルです。新規ユニバースウィザードでは、OLAP ユニバースレベル前置記号を異なる前置記号と置換することができます。レベルの番号付けは維持されますが、たとえば、前置記号 L を Level に置換することができます。[新しい前置記号]フィールドにユーザ独自の前置記号を入力してください。この前置記号は、OLAP ユニバース全体のすべてのレベルに付け加えられます。

関連項目

- ・ 456 ページの[OLAP ユニバースの依頼されたメジャーの設定](#)
- ・ 474 ページの[OLAP ユニバースのレベル 00 の再生成](#)
- ・ 474 ページの[レベル L00 の ALL への名前変更](#)

8.4.7 ユニバースと OLAP キューブの同期化

ユニバースを更新する際、ユニバースのオブジェクトは OLAP キューブのオブジェクトと比較されます。この比較により、キューブで加えられた変更がユニバースに悪影響を与えないことが保証されます。そのため、ユニバースで使用される(さらには削除される)どのオブジェクトも、常に利用できる必要があります。OLAP キューブのどの新しいオブジェクトも、ユニバースで利用できるようになります。変更によって異なるオブジェクトがどのように影響を受けるかについては、次のリンクを参照してください。

オブジェクトプロパティが更新されると、ユニバースでは特定のプロパティのみが更新され、他のプロパティは変更されません。次の表に結果を示します。

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)
- ・ 476 ページの[OLAP ユニバースの更新におけるディメンションの管理方法](#)
- ・ 481 ページの[OLAP ユニバースの更新における階層または特性の管理方法](#)
- ・ 488 ページの[OLAP ユニバースの更新におけるレベルの管理方法](#)
- ・ 492 ページの[OLAP ユニバースの更新における SAP 変数の管理方法](#)
- ・ 496 ページの[OLAP ユニバースの更新におけるキー数値またはメジャーの管理方法](#)
- ・ 501 ページの[OLAP ユニバースの更新における SAP キー日付の管理方法](#)

8.4.8 OLAP ユニバースの更新におけるディメンションの管理方法

この方法は、SAP、MSAS、Essbase の各データソースに適用されます。ユニバースのクラスは OLAP ディメンションと同等です。OLAP オブジェクトに関するユニバースオブジェクトの管理方法は、変更の種類によって異なります。特定の OLAP オブジェクトの変更によってユニバースオブジェクトがどのような影響を受けるかについては、下記のトピックを参照してください。

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)」
- ・ 476 ページの[ディメンションに変更がない場合](#)」
- ・ 477 ページの[ディメンションが更新される場合 \(名前、説明\)](#)」
- ・ 478 ページの[ディメンションが削除される場合](#)」
- ・ 479 ページの[ディメンションが移動される場合](#)」
- ・ 480 ページの[階層または特性が作成される場合](#)」
- ・ 480 ページの[ディメンションが新規作成される場合](#)」

8.4.8.1 ディメンションに変更がない場合

以下の表は、ディメンションに変更がない場合に、さまざまな状況において同等のユニバースクラスに発生する処理を示しています。

ユニバースクラスの状況	ユニバースクラスに発生する処理
変更なし	ユニバースクラスは変更されません。
更新された	[ビジネス名を保持] オプションが選択されていない場合は、ビジネス名が更新されます。 [オブジェクトの説明を保持する] オプションが選択されていない場合は、説明が更新されます。 これらのオプションが選択されている場合は、変更なしのままです。
削除された	ユニバースクラスは変更されません。 [手動で削除されたオブジェクトを再生成する] オプションが選択されている場合は、オブジェクトが作成されます。削除されていない子は再生成されません。
移動された	ユニバースクラスは変更されません。

ユニバースクラスの状況	ユニバースクラスに発生する処理
非表示	ユニバースクラスは変更されません。

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)」
- ・ 477 ページの[ディメンションが更新される場合 \(名前、説明\)](#)」
- ・ 478 ページの[ディメンションが削除される場合](#)」
- ・ 479 ページの[ディメンションが移動される場合](#)」
- ・ 480 ページの[階層または特性が作成される場合](#)」
- ・ 480 ページの[ディメンションが新規作成される場合](#)」

8.4.8.2 ディメンションが更新される場合 (名前、説明)

以下の表は、ディメンションの名前または説明が変更される場合、さまざまな状況において、同等のユニバースクラスに発生する処理を示しています。

ユニバースクラスの状況	ユニバースクラスに発生する処理
変更なし	<p>[ビジネス名を保持] オプションが選択されていない場合は、ビジネス名が更新されます。</p> <p>[オブジェクトの説明を保持する] オプションが選択されていない場合は、説明が更新されます。</p> <p>これらのオプションが選択されている場合は、変更なしのままです。</p>
更新された	<p>[ビジネス名を保持] オプションが選択されていない場合は、ビジネス名が更新されます。</p> <p>[オブジェクトの説明を保持する] オプションが選択されていない場合は、説明が更新されます。</p> <p>これらのオプションが選択されている場合は、変更なしのままです。</p>
削除された	<p>ユニバースクラスは変更されません。</p> <p>[手動で削除されたオブジェクトを再生成する] オプションが選択されている場合は、作成されます。</p> <p>削除されていない子は再生成されません。</p>

ユニバースクラスの状況	ユニバースクラスに発生する処理
移動された	<p>[ビジネス名を保持] オプションが選択されていない場合は、ビジネス名が更新されます。</p> <p>[オブジェクトの説明を保持する] オプションが選択されていない場合は、説明が更新されます。</p> <p>これらのオプションが選択されている場合は、変更なしのままです。</p>
非表示	<p>[ビジネス名を保持] オプションが選択されていない場合は、ビジネス名が更新されます。</p> <p>[オブジェクトの説明を保持する] オプションが選択されていない場合は、説明が更新されます。</p> <p>これらのオプションが選択されている場合は、変更なしのままです。</p>

関連項目

- ・ 471 ページの [OLAP ユニバースを最新表示する](#)」
- ・ 476 ページの [ディメンションに変更がない場合](#)」
- ・ 478 ページの [ディメンションが削除される場合](#)」
- ・ 479 ページの [ディメンションが移動される場合](#)」
- ・ 480 ページの [階層または特性が作成される場合](#)」
- ・ 480 ページの [ディメンションが新規作成される場合](#)」

8.4.8.3 ディメンションが削除される場合

以下の表は、ディメンションが削除される場合に、さまざまな状況において、同等のユニバースクラスに発生する処理を示しています。

ユニバースクラスの状況	ユニバースクラスに発生する処理
変更なし	<p>[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除されます。[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが使用できなくなります。クラスにカスタムオブジェクトが含まれている場合は、削除されません。</p>
更新された	<p>[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除されます。[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが使用できなくなります。クラスにカスタムオブジェクトが含まれている場合は、削除されません。</p>

ユニバースクラスの状況	ユニバースクラスに発生する処理
削除された	ユニバースクラスは変更されません。
移動された	[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除されます。[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが使用できなくなります。クラスにカスタムオブジェクトが含まれている場合は、削除されません。
非表示	ユニバースクラスは変更されません。

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)」
- ・ 476 ページの[ディメンションに変更がない場合](#)」
- ・ 477 ページの[ディメンションが更新される場合 \(名前、説明\)](#)」
- ・ 479 ページの[ディメンションが移動される場合](#)」
- ・ 480 ページの[階層または特性が作成される場合](#)」
- ・ 480 ページの[ディメンションが新規作成される場合](#)」

8.4.8.4 ディメンションが移動される場合

以下の表は、ディメンションが移動される場合に、さまざまな状況において同等のユニバースクラスに発生する処理を示しています。

ユニバースクラスの状況	ユニバースクラスに発生する処理
変更なし	適宜、クラスが移動されます。
更新された	変動なし
削除された	変更なし [手動で削除されたオブジェクトを再生成する] オプションが [はい] の場合は、作成されます。 削除されていない子は再生成されません。
移動された	変動なし
非表示	適宜、クラスが移動されます。

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)」
- ・ 476 ページの[ディメンションに変更がない場合](#)」
- ・ 477 ページの[ディメンションが更新される場合 \(名前、説明\)](#)」
- ・ 478 ページの[ディメンションが削除される場合](#)」
- ・ 480 ページの[階層または特性が作成される場合](#)」
- ・ 480 ページの[ディメンションが新規作成される場合](#)」

8.4.8.5 階層または特性が作成される場合

階層は MSAS データソースまたは Essbase データソースに適用され、特性は SAP データソースに適用されます。以下の表は、SAP 特性が作成される場合に、さまざまな状況において同等のユニバースクラスに発生する処理を示しています。

ユニバースクラスの状況	ユニバースクラスに発生する処理
変更なし	サブクラスが作成されます。
更新された	サブクラスが作成されます。
削除された	適用されません。
移動された	サブクラスが作成されます。
非表示	サブクラスが作成されます。

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)」
- ・ 476 ページの[ディメンションに変更がない場合](#)」
- ・ 477 ページの[ディメンションが更新される場合 \(名前、説明\)](#)」
- ・ 478 ページの[ディメンションが削除される場合](#)」
- ・ 479 ページの[ディメンションが移動される場合](#)」
- ・ 480 ページの[ディメンションが新規作成される場合](#)」

8.4.8.6 ディメンションが新規作成される場合

ディメンションの作成時にユニバースクラスが作成されます。

関連項目

- ・ [471 ページのOLAP ユニバースを最新表示する」](#)
- ・ [476 ページのディメンションに変更がない場合」](#)
- ・ [477 ページのディメンションが更新される場合 \(名前、説明\) 」](#)
- ・ [478 ページのディメンションが削除される場合 」](#)
- ・ [479 ページのディメンションが移動される場合 」](#)
- ・ [480 ページの階層または特性が作成される場合 」](#)

8.4.9 OLAP ユニバースの更新における階層または特性の管理方法

この節は、MSAS データソースと Essbase データソースの階層と SAP データソースの特性に適用されます。ユニバースのサブクラスは OLAP 特性と同等です。OLAP オブジェクトに関するユニバースオブジェクトの管理方法は、変更の種類によって異なります。特定の OLAP オブジェクトの変更によってユニバースオブジェクトがどのような影響を受けるかについては、下記のトピックを参照してください。

関連項目

- ・ [471 ページのOLAP ユニバースを最新表示する」](#)
- ・ [481 ページの階層または特性に変更がない場合」](#)
- ・ [482 ページの特性ビジネス名または説明が更新される場合 」](#)
- ・ [483 ページの特性のアクティブな階層が変更される場合 」](#)
- ・ [484 ページの特性の表示属性がナビゲーション属性に変更される場合」](#)
- ・ [486 ページの階層または特性が削除される場合 」](#)
- ・ [487 ページの階層または特性が移動される場合 」](#)
- ・ [488 ページの階層または特性が新規作成される場合」](#)

8.4.9.1 階層または特性に変更がない場合

以下の表は、階層または特性に変更がない場合に、さまざまな状況において同等のユニバースサブクラスに発生する処理を示しています。

ユニバースサブクラスの状況	ユニバースサブクラスに発生する処理
変更なし	変動なし
更新された	変動なし

ユニバースサブクラスの状態	ユニバースサブクラスに発生する処理
削除された	変更なし [手動で削除されたオブジェクトを再生成する]オプションが[はい]の場合は、作成されます。 削除されていない子レベルは再生成されません。
移動された	変動なし
非表示	変動なし

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)」
- ・ 482 ページの[特性ビジネス名または説明が更新される場合](#)」
- ・ 483 ページの[特性のアクティブな階層が変更される場合](#)」
- ・ 484 ページの[特性の表示属性がナビゲーション属性に変更される場合](#)」
- ・ 486 ページの[階層または特性が削除される場合](#)」
- ・ 487 ページの[階層または特性が移動される場合](#)」
- ・ 488 ページの[階層または特性が新規作成される場合](#)」

8.4.9.2 特性ビジネス名または説明が更新される場合

以下の表は、特性が更新される場合に、さまざまな状況において同等のユニバースサブクラスに発生する処理を示しています。

ユニバースサブクラスの状態	ユニバースサブクラスに発生する処理
変更なし	[ビジネス名を保持] オプションが選択されていない場合は、ビジネス名が更新されます。 [オブジェクトの説明を保持する] オプションが選択されていない場合は、説明が更新されます。 これらのオプションが選択されている場合は、変更なしのままです。

ユニバースサブクラスの状態	ユニバースサブクラスに発生する処理
更新された	<p>[ビジネス名を保持] オプションが選択されていない場合は、ビジネス名が更新されます。</p> <p>[オブジェクトの説明を保持する] オプションが選択されていない場合は、説明が更新されます。</p> <p>これらのオプションが選択されている場合は、変更なしのままです。</p>
削除された	<p>変更なし</p> <p>[手動で削除されたオブジェクトを再生成する] オプションが[はい]の場合は、作成されます。</p> <p>削除されていない子レベルは再生成されません。</p>
移動された	<p>[ビジネス名を保持] オプションが選択されていない場合は、ビジネス名が更新されます。</p> <p>[オブジェクトの説明を保持する] オプションが選択されていない場合は、説明が更新されます。</p> <p>これらのオプションが選択されていない場合は、変更なしのままです。</p>
非表示	<p>[ビジネス名を保持] オプションが選択されていない場合は、ビジネス名が更新されます。</p> <p>[オブジェクトの説明を保持する] オプションが選択されていない場合は、説明が更新されます。</p> <p>これらのオプションが選択されている場合は、変更なしのままです。</p>

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)
- ・ 481 ページの[階層または特性に変更がない場合](#)
- ・ 483 ページの[特性のアクティブな階層が変更される場合](#)
- ・ 484 ページの[特性の表示属性がナビゲーション属性に変更される場合](#)
- ・ 486 ページの[階層または特性が削除される場合](#)
- ・ 487 ページの[階層または特性が移動される場合](#)
- ・ 488 ページの[階層または特性が新規作成される場合](#)

8.4.9.3 特性のアクティブな階層が変更される場合

これは、SAP データソースにのみ適用されます。以下の表は、特性のアクティブな階層が変更されるさまざまな状況において、同等のユニバースサブクラスに発生する処理を示しています。

ユニバースサブクラスの状況	ユニバースサブクラスに発生する処理
変更なし	新規のアクティブな階層を参照するようにサブクラスで既存のオブジェクトの MDX 定義が更新されます。 最新表示前に作成されたレポートが引き続き使用されます。
更新された	新規のアクティブな階層を参照するようにサブクラスで既存のオブジェクトの MDX 定義が更新されます。 最新表示前に作成されたレポートが引き続き使用されます。
削除された	変更なし [手動で削除されたオブジェクトを再生成する] オプションが [はい] の場合は、作成されます。 削除されなかった子レベルは再生成されません。
移動された	新規のアクティブな階層を参照するようにサブクラスで既存のオブジェクトの MDX 定義が更新されます。 最新表示前に作成されたレポートが引き続き使用されます。
非表示	新規のアクティブな階層を参照するようにサブクラスで既存のオブジェクトの MDX 定義が更新されます。

関連項目

- ・ 471 ページの [OLAP ユニバースを最新表示する](#)
- ・ 481 ページの [階層または特性に変更がない場合](#)
- ・ 482 ページの [特性ビジネス名または説明が更新される場合](#)
- ・ 484 ページの [特性の表示属性がナビゲーション属性に変更される場合](#)
- ・ 486 ページの [階層または特性が削除される場合](#)
- ・ 487 ページの [階層または特性が移動される場合](#)
- ・ 488 ページの [階層または特性が新規作成される場合](#)

8.4.9.4 特性の表示属性がナビゲーション属性に変更される場合

これは、SAP データソースにのみ適用されます。以下の表は、特性の表示属性がナビゲーション属性に変更される場合に、さまざまな状況において同等のユニバースサブクラスに発生する処理を示しています。

ユニバースサブクラスの状況	ユニバースサブクラスに発生する処理
変更なし	作成
更新された	作成
削除された	作成
移動された	作成
非表示	作成

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)
- ・ 481 ページの[階層または特性に変更がない場合](#)
- ・ 482 ページの[特性ビジネス名または説明が更新される場合](#)
- ・ 483 ページの[特性のアクティブな階層が変更される場合](#)
- ・ 486 ページの[階層または特性が削除される場合](#)
- ・ 487 ページの[階層または特性が移動される場合](#)
- ・ 488 ページの[階層または特性が新規作成される場合](#)

8.4.9.5 特性のナビゲーション属性が表示属性に変更される場合

これは、SAP データソースにのみ適用されます。以下の表は、階層または特性のナビゲーション属性が表示属性に変更される場合に、さまざまな状況において同等のユニバースサブクラスに発生する処理を示しています。

ユニバースサブクラスの状況	ユニバースサブクラスに発生する処理
変更なし	[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除されます。[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが非表示になります。クラスにカスタムオブジェクトが含まれている場合は、削除されません。
更新された	[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除されます。[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが非表示になります。クラスにカスタムオブジェクトが含まれている場合は、削除されません。
削除された	変動なし

ユニバースサブクラスの状況	ユニバースサブクラスに発生する処理
移動された	[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除されます。[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが非表示になります。クラスにカスタムオブジェクトが含まれている場合は、削除されません。
非表示	[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除されます。[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが非表示になります。クラスにカスタムオブジェクトが含まれている場合は、削除されません。

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)」
- ・ 481 ページの[階層または特性に変更がない場合](#)」
- ・ 482 ページの[特性ビジネス名または説明が更新される場合](#)」
- ・ 483 ページの[特性のアクティブな階層が変更される場合](#)」
- ・ 486 ページの[階層または特性が削除される場合](#)」
- ・ 487 ページの[階層または特性が移動される場合](#)」
- ・ 488 ページの[階層または特性が新規作成される場合](#)」

8.4.9.6 階層または特性が削除される場合

以下の表は、階層または特性が削除される場合に、さまざまな状況において同等のユニバースサブクラスに発生する処理を示しています。

ユニバースサブクラスの状況	ユニバースサブクラスに発生する処理
変更なし	[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除されます。[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが使用できなくなります。サブクラスにカスタムオブジェクトが含まれている場合は、削除されません。
更新された	[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除され、[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが使用できなくなります。サブクラスにカスタムオブジェクトが含まれている場合は、削除されません。
削除された	変動なし

ユニバースサブクラスの状況	ユニバースサブクラスに発生する処理
移動された	[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除され、[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが使用できなくなります。サブクラスにカスタムオブジェクトが含まれている場合は、削除されません。
非表示	[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除され、[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが使用できなくなります。サブクラスにカスタムオブジェクトが含まれている場合は、削除されません。

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)」
- ・ 481 ページの[階層または特性に変更がない場合](#)」
- ・ 482 ページの[特性ビジネス名または説明が更新される場合](#)」
- ・ 483 ページの[特性のアクティブな階層が変更される場合](#)」
- ・ 484 ページの[特性の表示属性がナビゲーション属性に変更される場合](#)」
- ・ 487 ページの[階層または特性が移動される場合](#)」
- ・ 488 ページの[階層または特性が新規作成される場合](#)」

8.4.9.7 階層または特性が移動される場合

特性が同一ディメンション内で移動される場合は、変更はありません。以下の表は無視してください。以下の表は、階層または特性が他のディメンションに移動される場合に、さまざまな状況において同等のユニバースサブクラスに発生する処理を示しています。

ユニバースサブクラスの状況	ユニバースサブクラスに発生する処理
変更なし	適宜、サブクラスが移動されます。
更新された	適宜、サブクラスが移動されます。
削除された	変更なし [手動で削除されたオブジェクトを再生成する] オプションが [はい] の場合は、作成されます。 削除されていない子レベルは再生成されません。
移動された	変更なし

ユニバースサブクラスの状態	ユニバースサブクラスに発生する処理
非表示	適宜、サブクラスが移動されます。

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)」
- ・ 481 ページの[階層または特性に変更がない場合](#)」
- ・ 482 ページの[特性ビジネス名または説明が更新される場合](#)」
- ・ 483 ページの[特性のアクティブな階層が変更される場合](#)」
- ・ 484 ページの[特性の表示属性がナビゲーション属性に変更される場合](#)」
- ・ 486 ページの[階層または特性が削除される場合](#)」
- ・ 488 ページの[階層または特性が新規作成される場合](#)」

8.4.9.8 階層または特性が新規作成される場合

階層または特性の作成時にユニバースサブクラスが作成されます。

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)」
- ・ 481 ページの[階層または特性に変更がない場合](#)」
- ・ 482 ページの[特性ビジネス名または説明が更新される場合](#)」
- ・ 483 ページの[特性のアクティブな階層が変更される場合](#)」
- ・ 484 ページの[特性の表示属性がナビゲーション属性に変更される場合](#)」
- ・ 486 ページの[階層または特性が削除される場合](#)」
- ・ 487 ページの[階層または特性が移動される場合](#)」

8.4.10 OLAP ユニバースの更新におけるレベルの管理方法

注

ユニバースでは、レベルを他の階層に移動しないでください。レベルを移動する場合は、レベルをコピーして新規の階層に貼り付けます。

ユニバースレベルまたはディメンションオブジェクトは、OLAP レベルと同等です。OLAP オブジェクトに関するユニバースオブジェクトの管理方法は、変更の種類によって異なります。特定の OLAP オブジェクトの変更によってユニバースオブジェクトがどのような影響を受けるかについては、下記のトピックを参照してください。

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)」
- ・ 489 ページの[レベルに変更がない場合](#)」
- ・ 489 ページの[レベルの名前または説明が更新される場合](#)」
- ・ 491 ページの[レベルが削除される場合](#)」
- ・ 491 ページの[レベルが移動される場合](#)」
- ・ 492 ページの[レベルが新規作成される場合](#)」

8.4.10.1 レベルに変更がない場合

以下の表は、レベルに変更がない場合に、さまざまな状況においてユニバースレベルに発生する処理を示しています。

ユニバースレベルの状況	ユニバースレベルに発生する処理
変更なし	変動なし
更新された	変動なし
削除された	変更なし[手動で削除されたオブジェクトを再生成する]オプションが[はい]の場合は、作成されます。
移動された	変動なし
非表示	変動なし

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)」
- ・ 489 ページの[レベルの名前または説明が更新される場合](#)」
- ・ 491 ページの[レベルが削除される場合](#)」
- ・ 491 ページの[レベルが移動される場合](#)」
- ・ 492 ページの[レベルが新規作成される場合](#)」

8.4.10.2 レベルの名前または説明が更新される場合

以下の表は、レベルの名前または説明が更新される場合に、さまざまな状況においてユニバースレベルに発生する処理を示しています。

ユニバースレベルの状況	ユニバースレベルに発生する処理
変更なし	<p>[ビジネス名を保持] オプションが選択されていない場合は、ビジネス名が更新されます。</p> <p>[オブジェクトの説明を保持する] オプションが選択されていない場合は、説明が更新されます。</p> <p>これらのオプションが選択されている場合は、変更なしのままです。</p>
更新された	<p>[ビジネス名を保持] オプションが選択されていない場合は、ビジネス名が更新されます。</p> <p>[オブジェクトの説明を保持する] オプションが選択されていない場合は、説明が更新されます。</p> <p>これらのオプションが選択されている場合は、変更なしのままです。</p>
削除された	<p>変更なし[手動で削除されたオブジェクトを再生成する] オプションが [はい] の場合は、作成されます。</p>
移動された	<p>[ビジネス名を保持] オプションが選択されていない場合は、ビジネス名が更新されます。</p> <p>[オブジェクトの説明を保持する] オプションが選択されていない場合は、説明が更新されます。</p> <p>これらのオプションが選択されている場合は、変更なしのままです。</p>
非表示	<p>[ビジネス名を保持] オプションが選択されていない場合は、ビジネス名が更新されます。</p> <p>[オブジェクトの説明を保持する] オプションが選択されていない場合は、説明が更新されます。</p> <p>これらのオプションが選択されている場合は、変更なしのままです。</p>

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)」
- ・ 489 ページの[レベルに変更がない場合](#)」
- ・ 491 ページの[レベルが削除される場合](#)」
- ・ 491 ページの[レベルが移動される場合](#)」
- ・ 492 ページの[レベルが新規作成される場合](#)」

8.4.10.3 レベルが削除される場合

以下の表は、レベルが削除される場合に、さまざまな状況においてユニバースレベルに発生する処理を示しています。

ユニバースレベルの状況	ユニバースレベルに発生する処理
変更なし	[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除されます。[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが使用できなくなります。
更新された	[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除されます。[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが使用できなくなります。
削除された	変更なし
移動された	[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除されます。[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが使用できなくなります。
非表示	[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除されます。[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが使用できなくなります。

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)
- ・ 489 ページの[レベルに変更がない場合](#)
- ・ 489 ページの[レベルの名前または説明が更新される場合](#)
- ・ 491 ページの[レベルが移動される場合](#)
- ・ 492 ページの[レベルが新規作成される場合](#)

8.4.10.4 レベルが移動される場合

以下の表は、レベルが移動される場合に、さまざまな状況においてユニバースレベルに発生する処理を示しています。

ユニバースレベルの状況	ユニバースレベルに発生する処理
変更なし	適宜、(同一階層内で)レベルが移動されます。
更新された	適宜、(同一階層内で)レベルが移動されます。
削除された	変更なし[手動で削除されたオブジェクトを再生成する] オプションが [はい] の場合は、作成されます。
移動された	変更なし[手動で削除されたオブジェクトを再生成する] オプションが [はい] の場合は、作成されます。
非表示	適宜、(同一階層内で)レベルが移動されます。

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)
- ・ 489 ページの[レベルに変更がない場合](#)
- ・ 489 ページの[レベルの名前または説明が更新される場合](#)
- ・ 491 ページの[レベルが削除される場合](#)
- ・ 492 ページの[レベルが新規作成される場合](#)

8.4.10.5 レベルが新規作成される場合

OLAP レベルの作成時にユニバースレベルが作成されます。

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)
- ・ 489 ページの[レベルに変更がない場合](#)
- ・ 489 ページの[レベルの名前または説明が更新される場合](#)
- ・ 491 ページの[レベルが削除される場合](#)
- ・ 491 ページの[レベルが移動される場合](#)

8.4.11 OLAP ユニバースの更新における SAP 変数の管理方法

この節に関係するのは、SAP データソースのみです。ユニバースフィルタと関連する値一覧のオブジェクトは、OLAP 変数と同等です。OLAP オブジェクトに関するユニバースオブジェクトの管理方法は、変更の種類によっ

て異なります。特定の OLAP オブジェクトの変更によってユニバースオブジェクトがどのような影響を受けるかについては、下記のトピックを参照してください。

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)」
- ・ 493 ページの[SAP 変数に変更がない場合](#)」
- ・ 493 ページの[SAP 変数の名前または説明が更新される場合](#)」
- ・ 494 ページの[SAP 変数が削除される場合](#)」
- ・ 495 ページの[SAP 変数が新規作成される場合](#)」

8.4.11.1 SAP 変数に変更がない場合

以下の表は、SAP ソース変数に変更がない場合の、さまざまな状況におけるユニバースフィルタの管理方法を示しています。

ユニバースフィルタの状況	ユニバースフィルタに発生する処理
変更なし	変動なし
更新された	変動なし
削除された	作成されます。変数に関係付けられた特性がユニバースにない場合は、その特性のサブクラスも作成されます。
移動された	変動なし
非表示	変動なし

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)」
- ・ 493 ページの[SAP 変数の名前または説明が更新される場合](#)」
- ・ 494 ページの[SAP 変数が削除される場合](#)」
- ・ 495 ページの[SAP 変数が新規作成される場合](#)」

8.4.11.2 SAP 変数の名前または説明が更新される場合

以下の表は、SAP ソース変数の名前または説明のいずれかが更新される場合の、さまざまな状況におけるユニバースフィルタの管理方法を示しています。

ユニバースフィルタの状況	ユニバースフィルタに発生する処理
変更なし	<p>[ビジネス名を保持] オプションが選択されていない場合は、ビジネス名が更新されます。</p> <p>[オブジェクトの説明を保持する] オプションが選択されていない場合は、説明が更新されます。</p> <p>これらのオプションが選択されている場合は、変更なしのままです。</p>
更新された	<p>[ビジネス名を保持] オプションが選択されていない場合は、ビジネス名が更新されます。</p> <p>[オブジェクトの説明を保持する] オプションが選択されていない場合は、説明が更新されます。</p> <p>これらのオプションが選択されている場合は、変更なしのままです。</p>
削除された	作成されます。変数に関係付けられた特性がユニバースにない場合は、その特性のサブクラスも作成されます。
移動された	<p>[ビジネス名を保持] オプションが選択されていない場合は、ビジネス名が更新されます。</p> <p>[オブジェクトの説明を保持する] オプションが選択されていない場合は、説明が更新されます。</p> <p>これらのオプションが選択されている場合は、変更なしのままです。</p>
非表示	<p>[ビジネス名を保持] オプションが選択されていない場合は、ビジネス名が更新されます。</p> <p>[オブジェクトの説明を保持する] オプションが選択されていない場合は、説明が更新されます。</p> <p>これらのオプションが選択されている場合は、変更なしのままです。</p>

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)」
- ・ 493 ページの[SAP 変数に変更がない場合](#)」
- ・ 494 ページの[SAP 変数が削除される場合](#)」
- ・ 495 ページの[SAP 変数が新規作成される場合](#)」

8.4.11.3 SAP 変数が削除される場合

以下の表は、SAP 変数が削除される場合の、さまざまな状況におけるユニバースフィルタの管理方法を示しています。

ユニバースフィルタの状況	ユニバースフィルタに発生する処理
変更なし	[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除されます。[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが非表示になります。無効にする場合: クエリでの自動適用を避けるために必須からオプションへの変更も行われます。
更新された	[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除されます。[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが非表示になります。無効にする場合: クエリでの自動適用を避けるために必須からオプションへの変更も行われます。
削除された	変更なし
移動された	[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除されます。[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが非表示になります。無効にする場合: クエリでの自動適用を避けるために必須からオプションへの変更も行われます。
非表示	[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除されます。[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが非表示になります。無効にする場合: クエリでの自動適用を避けるために必須からオプションへの変更も行われます。

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)」
- ・ 493 ページの[SAP 変数に変更がない場合](#)」
- ・ 493 ページの[SAP 変数の名前または説明が更新される場合](#)」
- ・ 495 ページの[SAP 変数が新規作成される場合](#)」

8.4.11.4 SAP 変数が新規作成される場合

以下の表は、SAP 変数が新規作成される場合の、さまざまな状況におけるユニバースフィルタの管理方法を示しています。

ユニバースフィルタの状況	ユニバースフィルタに発生する処理
変更なし	作成されます。変数に関係付けられた特性がユニバースにない場合は、その特性のサブクラスも作成されます。
更新された	作成されます。変数に関係付けられた特性がユニバースにない場合は、その特性のサブクラスも作成されます。
削除された	作成されます。変数に関係付けられた特性がユニバースにない場合は、その特性のサブクラスも作成されます。
移動された	作成されます。変数に関係付けられた特性がユニバースにない場合は、その特性のサブクラスも作成されます。
非表示	作成されます。変数に関係付けられた特性がユニバースにない場合は、その特性のサブクラスも作成されます。

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)」
- ・ 493 ページの[SAP 変数に変更がない場合](#)」
- ・ 493 ページの[SAP 変数の名前または説明が更新される場合](#)」
- ・ 494 ページの[SAP 変数が削除される場合](#)」

8.4.12 OLAP ユニバースの更新におけるキー数値またはメジャーの管理方法

SAP データソースではキー数値が使用され、MSAS データソースおよび Essbase データソースではメジャーが使用されます。ユニバースのメジャーは、OLAP キー数値と同等です。OLAP オブジェクトに関するユニバースオブジェクトの管理方法は、変更の種類によって異なります。特定の OLAP オブジェクトの変更によってユニバースオブジェクトがどのような影響を受けるかについては、下記のトピックを参照してください。

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)」
- ・ 497 ページの[キー数値またはメジャーに変更がない場合](#)」
- ・ 497 ページの[キー数値またはメジャーの名前、説明、またはデータタイプが更新される場合](#)」
- ・ 499 ページの[キー数値またはメジャーが削除される場合](#)」
- ・ 500 ページの[キー数値またはメジャーが移動される場合](#)」
- ・ 500 ページの[キー数値またはメジャーが新規作成される場合](#)」

8.4.12.1 キー数値またはメジャーに変更がない場合

以下の表は、SAP キー数値または MSAS/Essbase メジャーに変更がないさまざまな状況において、ユニバースメジャーに発生する処理を示しています。

ユニバースメジャーの状況	ユニバースメジャーに発生する処理
変更なし	変動なし
更新された	変動なし
削除された	変更なし[手動で削除されたオブジェクトを再生成する]オプションが[はい]の場合は、作成されます。
移動された	変動なし
非表示	変動なし

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)
- ・ 497 ページの[キー数値またはメジャーの名前、説明、またはデータタイプが更新される場合](#)」
- ・ 499 ページの[キー数値またはメジャーが削除される場合](#)」
- ・ 500 ページの[キー数値またはメジャーが移動される場合](#)」
- ・ 500 ページの[キー数値またはメジャーが新規作成される場合](#)」

8.4.12.2 キー数値またはメジャーの名前、説明、またはデータタイプが更新される場合

以下の表は、SAP キー数値または MSAS/Essbase メジャーが更新される場合に、さまざまな状況においてユニバースメジャーに発生する処理を示しています。

ユニバースメジャーの状況	ユニバースメジャーに発生する処理
変更なし	<p>[ビジネス名を保持] オプションが選択されていない場合は、ビジネス名が更新されます。</p> <p>[オブジェクトの説明を保持する] オプションが選択されていない場合は、説明が更新されます。</p> <p>[オブジェクトのデータ型を保持する] オプションが選択されていない場合は、データ型が更新されます。</p> <p>これらのオプションが選択されている場合は、変更なしのままです。</p>
更新された	<p>[ビジネス名を保持] オプションが選択されていない場合は、ビジネス名が更新されます。</p> <p>[オブジェクトの説明を保持する] オプションが選択されていない場合は、説明が更新されます。</p> <p>[オブジェクトのデータ型を保持する] オプションが選択されていない場合は、データ型が更新されます。</p> <p>これらのオプションが選択されている場合は、変更なしのままです。</p>
削除された	<p>変更なし[手動で削除されたオブジェクトを再生成する]オプションが[はい]の場合は、作成されます。</p>
移動された	<p>[ビジネス名を保持] オプションが選択されていない場合は、ビジネス名が更新されます。</p> <p>[オブジェクトの説明を保持する] オプションが選択されていない場合は、説明が更新されます。</p> <p>[オブジェクトのデータ型を保持する] オプションが選択されていない場合は、データ型が更新されます。</p> <p>これらのオプションが選択されている場合は、変更なしのままです。</p>
非表示	<p>[ビジネス名を保持] オプションが選択されていない場合は、ビジネス名が更新されます。</p> <p>[オブジェクトの説明を保持する] オプションが選択されていない場合は、説明が更新されます。</p> <p>[オブジェクトのデータ型を保持する] オプションが選択されていない場合は、データ型が更新されます。</p> <p>これらのオプションが選択されている場合は、変更なしのままです。</p>

関連項目

- 471 ページの[OLAP ユニバースを最新表示する](#)

- ・ [497 ページのキー数値またはメジャーに変更がない場合](#)」
- ・ [499 ページのキー数値またはメジャーが削除される場合](#)」
- ・ [500 ページのキー数値またはメジャーが移動される場合](#)」
- ・ [500 ページのキー数値またはメジャーが新規作成される場合](#)」

8.4.12.3 キー数値またはメジャーが削除される場合

以下の表は、SAP キー数値または MSAS/Essbase メジャーが削除される場合に、さまざまな状況においてユニバースメジャーに発生する処理を示しています。

ユニバースメジャーの状況	ユニバースメジャーに発生する処理
変更なし	[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除されます。[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが使用できなくなります。
更新された	[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除されます。[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが使用できなくなります。
削除された	変更なし
移動された	[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除されます。[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが使用できなくなります。
非表示	[使用されていないオブジェクトを削除する]オプションが選択されている場合は、削除されます。[使用されていないオブジェクトを非表示にする]オプションが選択されている場合は、サブクラスが使用できなくなります。

関連項目

- ・ [471 ページのOLAP ユニバースを最新表示する](#)」
- ・ [497 ページのキー数値またはメジャーに変更がない場合](#)」
- ・ [497 ページのキー数値またはメジャーの名前、説明、またはデータタイプが更新される場合](#)」
- ・ [500 ページのキー数値またはメジャーが移動される場合](#)」
- ・ [500 ページのキー数値またはメジャーが新規作成される場合](#)」

8.4.12.4 キー数値またはメジャーが移動される場合

以下の表は、SAP キー数値または MSAS/Essbase メジャーが移動されるさまざまな状況において、ユニバースメジャーに発生する処理を示しています。

ユニバースメジャーの状況	ユニバースメジャーに発生する処理
変更なし	適宜、オブジェクトが移動されます。
更新された	適宜、オブジェクトが移動されます。
削除された	変更なし[手動で削除されたオブジェクトを再生成する]オプションが[はい]の場合は、作成されます。
移動された	変更なし
非表示	適宜、オブジェクトが移動されます。

関連項目

- ・ [471 ページのOLAP ユニバースを最新表示する](#)
- ・ [497 ページのキー数値またはメジャーに変更がない場合](#)
- ・ [497 ページのキー数値またはメジャーの名前、説明、またはデータタイプが更新される場合](#)
- ・ [499 ページのキー数値またはメジャーが削除される場合](#)
- ・ [500 ページのキー数値またはメジャーが新規作成される場合](#)

8.4.12.5 キー数値またはメジャーが新規作成される場合

OLAP キー数値またはメジャーの作成時にユニバースメジャーが作成されます。

関連項目

- ・ [471 ページのOLAP ユニバースを最新表示する](#)
- ・ [497 ページのキー数値またはメジャーに変更がない場合](#)
- ・ [497 ページのキー数値またはメジャーの名前、説明、またはデータタイプが更新される場合](#)
- ・ [499 ページのキー数値またはメジャーが削除される場合](#)
- ・ [500 ページのキー数値またはメジャーが移動される場合](#)

8.4.13 OLAP ユニバースの更新における SAP キー日付の管理方法

この節は、SAP データソースにのみ適用されます。ユニバースパラメータは、OLAP キー日付と同等です。OLAP オブジェクトに関するユニバースオブジェクトの管理方法は、変更の種類によって異なります。特定の OLAP オブジェクトの変更によってユニバースオブジェクトがどのような影響を受けるかについては、下記のトピックを参照してください。

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)
- ・ 501 ページの[SAP キー日付に変更がない場合](#)
- ・ 502 ページの[SAP キー日付が削除される場合](#)
- ・ 502 ページの[SAP キー日付が新規作成される場合](#)

8.4.13.1 SAP キー日付に変更がない場合

ユニバースパラメータは、OLAP キー日付と同等です。以下の表は、SAP キー日付に変更がない場合に、さまざまな状況においてユニバースパラメータに発生する処理を示しています。

ユニバースパラメータの状況	ユニバースパラメータに発生する処理
変更なし	変動なし
更新された	適用外
削除された	適用外
移動された	適用外
非表示	適用外

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)
- ・ 502 ページの[SAP キー日付が削除される場合](#)
- ・ 502 ページの[SAP キー日付が新規作成される場合](#)

8.4.13.2 SAP キー日付が削除される場合

ユニバースパラメータは、OLAP キー日付と同等です。以下の表は、SAP キー日付が削除される場合に、さまざまな状況においてユニバースパラメータに発生する処理を示しています。

ユニバースパラメータの状況	ユニバースパラメータに発生する処理
変更なし	削除
更新された	適用外
削除された	適用外
移動された	適用外
非表示	適用外

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する](#)
- ・ 501 ページの[SAP キー日付に変更がない場合](#)
- ・ 502 ページの[SAP キー日付が新規作成される場合](#)

8.4.13.3 SAP キー日付が新規作成される場合

ユニバースパラメータは、OLAP キー日付と同等です。以下の表は、SAP キー日付が新規作成される場合に、さまざまな状況においてユニバースパラメータに発生する処理を示しています。

ユニバースパラメータの状況	ユニバースパラメータに発生する処理
変更なし	作成
更新された	適用外
削除された	作成
移動された	適用外
非表示	適用外

関連項目

- ・ 471 ページの[OLAP ユニバースを最新表示する」](#)
- ・ 501 ページの[SAP キー日付に変更がない場合」](#)
- ・ 502 ページの[SAP キー日付が削除される場合」](#)

8.5 さまざまな OLAP キューブをユニバースにマップする方法

8.5.1 ユニバースにおける SAP NetWeaver BW オブジェクトのマッピングおよび使用方法

InfoCube または BEx クエリのいずれかからユニバースを作成すると、ユニバースデザインツールによって SAP NetWeaver BW OLAP 構造がユニバースの同等のクラスおよびオブジェクトにマップされます。

行、列、自由特性、およびフィルタとして設定されている BEx クエリのすべての InfoObjects は、ユニバースで公開されます。これには、特性、階層、キー数値、構造、および変数が含まれます。

階層がマップされ、Web Intelligence ユーザが BW 階層に従ってドリルダウンできるようになります。

インフォキューブでは、すべてのディメンション、キー数値、および階層がマップされます。

次の表に、各 BW オブジェクトに対して作成されるユニバースオブジェクトを示します。

NetWeaver BW オブジェクト:	作成されるユニバースオブジェクト
ディメンショングループ	クラス
特性	ディメンションおよび詳細オブジェクトを含むサブクラス
階層を含む特性	<p>データソースが BEx クエリの場合: 現在定義されている階層内の各階層レベルに対するディメンションおよび詳細オブジェクトを含むサブクラス</p> <p>データソースがインフォキューブの場合: 特性に対して定義されているすべての階層の各階層レベルに対するディメンションおよび詳細オブジェクトを含むサブクラス</p>

NetWeaver BW オブジェクト:	作成されるユニバースオブジェクト
特性に基づく構造 (BEx クエリのみ)	構造に対して単一のディメンションオブジェクトを含むクラス
ナビゲーション属性	ディメンションおよび詳細オブジェクトを含むサブクラス(特性と同様)
表示属性	ディメンションの詳細オブジェクト
キー数値構造	クラス
キー数値	単位/通貨用のディメンションオブジェクトを含むキー数値構造に対するクラスのメジャーオブジェクト
計算キー数値 (BEx クエリのみ)	メジャーおよびディメンションオブジェクト (キー数値と同様)
制限キー数値 (BEx クエリのみ)	メジャーおよびディメンションオブジェクト (キー数値と同様)
変数 (BEx クエリのみ)	クエリに必須のフィルタ 変数が適用されるディメンションのクラスでは、2 つのディメンションオブジェクト(1 つはキャプション用、もう 1 つは説明用)が値の一覧をサポートしています。
キー日付変数 (BEx クエリのみ)	ユニバースでキー日付変数を定義するユニバースパラメータ

BEx クエリのフィルタセクションの特性はマップされません。しかし、フィルタリングはユニバースに適用されます。フィルタに固定値が設定されている場合、フィルタは Web Intelligence クエリの実行中に透過的に適用されます。特性に変数が定義されている場合、変数は次の制限と共にマップされます。

- ・ 変数は、常に必須変数として作用する
- ・ 階層バージョン変数を除き、階層および階層ノード変数がサポートされています。

これらの制限を回避するために、BEx クエリのフィルタセクションからフリーセクションへ特性を移動することをお勧めします。

関連項目

- ・ 505 ページの[特性のマップと使用方法](#)」

- ・ 505 ページの[キー数値のマップと使用方法](#)」
- ・ 506 ページの[階層のマップと使用方法](#)」
- ・ 506 ページの[ユニバースにおける変数のサポート方法](#)」
- ・ 508 ページの[ユニバースへの変数のマップ方法](#)」

8.5.1.1 特性のマップと使用方法

BEx クエリまたはインフォキューブで、特性に階層が定義されていない場合、ユニバースデザインツールにより、2 つのディメンションオブジェクト ("レベル 00" および "レベル 01") として特性を含むクラスが作成されます。"レベル 00" ディメンションは、すべてのメンバーが選択された場合 (NetWeaver から返されるメンバーはすべてのメンバー)、特性の集計を表します。"レベル 01" ディメンションには、特性のすべてのメンバーが値のフラットリストとして含まれます。

各ディメンションオブジェクトについて、ユニバースデザインツールは、キーに詳細オブジェクト、説明に最大 3 つの詳細オブジェクト(短い、中間、長い説明)、各表示属性に詳細オブジェクトを作成します。

SELECT 句は特性の技術名称を使用して定義されます。

BW クエリで定義されるナビゲーション属性は、特性と同様の方法で親オブジェクトクラスでマップされます。

注

ユニバースで多数のナビゲーション属性を定義すると、Web Intelligence におけるクエリのパフォーマンスに悪影響を与えます。

特性に基づく BEx クエリに定義された構造は、構造要素をディメンションメンバーとして含む単一のディメンションオブジェクトとしてユニバースに含まれます。

8.5.1.2 キー数値のマップと使用方法

InfoCube 内または BEx クエリで定義されたすべてのキー数値は、キー数値と呼ばれる単一のオブジェクトクラスのユニバースに格納されます。

ほとんどのキー数値は NetWeaver BW で定義され、通貨特性または単位特性のいずれかが設定されます。各キー数値に対し、ユニバースデザインツールによって次のオブジェクトが生成されます。

- ・ 単位を除くキー数値に対応する数値形式のメジャーオブジェクト。
- ・ 単位または通貨を含む文字形式のディメンションオブジェクト。たとえば、USD、¤、km です。
- ・ SAP サーバに設定されたユーザ設定に基づく単位(書式付きの値)、およびキー数値を含む文字形式のディメンションオブジェクト。たとえば、200 USD、345 ¤、25 km です。

キー数値クラスには、BEx クエリで定義された制限キー数値および計算キー数値が含まれます。元の計算および制限はクエリに適用されますが、ユニバースでは公開されません。

8.5.1.3 階層のマッピングと使用方法

階層をマッピングすると、Web Intelligence ユーザは、カスタマイズのユニバース階層と同様に、SAP NetWeaver BW 階層をドリルダウンできます。

注

Web Intelligence の [ドキュメントのプロパティ] ダイアログボックスの [クエリドリルの使用] オプションを使用すると、ドリルダウンパフォーマンスが大幅に向上します。

BEx クエリの特性に階層が定義されている場合、ユニバースデザインツールによって、ユニバースに 1 つの階層構造と、その階層の各レベルにサブクラスが作成されます。構造は、現在の BEx クエリ定義に応じて異なります。

- ・ BEx クエリで階層が定義されている場合、ユニバースデザインツールは、ユニバースにこの階層構造を作成します。
- ・ 実行時に階層の選択をユーザに許可する階層変数が BEx クエリに定義されている場合、ユニバースデザインツールは、ユニバースに汎用階層を作成します。この構造には、特性に使用できる階層構造用に定義された最大数のレベルが含まれています。

インフォキューブの最上位のユニバースを作成する場合、特性に定義されるすべての階層は、作成されたユニバースで公開されます。ユニバースデザインツールは、各階層構造に対してサブクラスを作成します。これらの各サブクラスは、階層のレベルに対して作成されます。

ユニバースにおいて“レベル 00”階層は、構造の最上位ノードを表します。階層構造に複数の最上位ノードが存在する場合、“レベル 00”ディメンションには値の一覧としてすべての最上位ノードが含まれます。階層属性が非フィルタ未割り当てノードに設定されている場合は、未割り当てメンバーの最上位ノードを“レベル 00”を含める必要があります。未割り当てメンバーは、階層の最下位レベルでグループ化されます。

注

ほとんどの場合、SAP NetWeaver BW 階層の最上位ノードは、1 つのみです。デフォルトのユニバースから“レベル 00”オブジェクトを削除して、使用するユニバースを単純化することができます。一般に、“レベル 00”を維持する必要があるのは、未割り当てメンバーのクエリ/レポートが必要な場合のみです。

BEx クエリで階層内のレベル数に変更された場合、ユニバースを更新する必要があります。

関連項目

- ・ 467 ページの [OLAP ユニバースのライフサイクル管理](#)

8.5.1.4 ユニバースにおける変数のサポート方法

SAP 変数は、BW クエリに定義されるユーザプロンプトとして解釈できます。変数は、必須の場合もオプションの場合もあります。デフォルト値を設定することもできます。

特性の変数は、特性の値をフィルタするために使用されます。変数は、クエリを実行するときに値と共に取得されます。BW 変数には、特性値、階層、階層ノード、テキストおよび式要素を格納できます。

NetWeaver BW 変数が適用されるのは、BEx クエリのみです。

注

SAP NetWeaver BW 変数ウィザードの [その他の設定] ダイアログボックスでクエリデザイナの変数を定義する際に、[入力可能] オプションを選択する必要があります。

ユニバースでは、次の種類の SAP NetWeaver BW 変数がサポートされています。

- ・ 特性変数
- ・ 階層バージョン変数を除く、階層変数
- ・ 階層ノード変数
- ・ 通貨変数
- ・ 式変数
- ・ テキスト変数(置換パスとして)
- ・ キー日付変数

次の表には、ユーザ入力の BW 変数に対するユニバースのサポート状況が示されています。ユーザ入力変数は、必須の場合もオプションの場合もあります。デフォルト値を設定することもできます。

変数の種類		サポートレベル
特性(キー日付および通貨を含む)	単一値プロンプト	サポートされます
	複数の単一値プロンプト	サポートされます
	間隔プロンプト	サポートされます 単一値変数であるキー日付変数に対してはサポートされていません。
	選択オプションプロンプト	間隔プロンプトとしてサポート 単一値変数であるキー日付変数を求める間隔プロンプトとしてはサポートされていません。
	計算済みの値セット	サポートされていません。
テキスト		サポートされます
式		Price 変数、Quota 変数および数値のみサポート
階層		Version 変数以外をサポート
階層ノード		サポートされます

次の表に、BW 変数の他の処理タイプに対するユニバースのサポート状況を示します。

変数の種類	処理の種類			
	置換パス	権限	顧客の終了	SAP の終了
特性	サポートされます	サポートされます	サポートされていますが、ユニバースでプロンプトは作成されません。	サポートされます
テキスト	サポートされます	該当なし	サポートされます	該当なし
式	サポートされます	該当なし	サポートされます	ユーザ入力なしでサポート
階層	該当なし	該当なし	サポートされます	サポートされます
階層ノード	該当なし	該当なし	サポートされます	ユーザ入力なしでサポート

Exclude 演算子はサポートされていますが、Web Intelligence では選択した値がクエリから除外されるように指定されません。Less than、Greater than などの他の演算子は、選択オプション入力タイプのみで使用できます。選択オプションタイプは、Web Intelligence のプロンプトの間隔に変換されます。

注

Web Intelligence で BW 変数を処理するには、Web Intelligence クエリに少なくとも 1 つのメジャーを含める必要があります。

関連項目

- ・ 508 ページの[ユニバースへの変数のマップ方法](#)」
- ・ 510 ページの[ユニバースにおけるキー日付変数のサポート方法](#)」
- ・ 511 ページの[ユニバースにおける階層変数と階層ノード変数のサポート方法](#)」

8.5.1.4.1 ユニバースへの変数のマップ方法

ディメンションが結果セットで使用されておらず、ユーザが結果セットを制限できる場合でも、ユーザには、すべてのオプション変数および必須変数の入力を求めるプロンプトが表示される必要があります。したがって、対応する特性がクエリに含まれていなくても、BEx クエリに定義されている変数はマップされます。

ユーザは、変数が必須なのか、オプションなのかを把握し、オプション変数を無視できるようにする必要があります。オプション変数はユニバースでオプションとして定義され、Web Intelligence でオプションプロンプトとなります。必須変数は、Web Intelligence で強制プロンプトとなります。

特性変数については、ユニバースデザインツールによってユニバースに必須フィルタが作成されます。必須フィルタは、定義済みのクエリフィルタオブジェクトです。Web Intelligence ユーザには表示されませんが、ユニバースに作成したすべての Web Intelligence クエリに自動的かつ透過的に適用されます。

変数の種類	マップ先
通貨および式変数を含む特性変数	ユニバースの必須フィルタ
階層変数	ユニバースの必須フィルタ
階層ノード変数	クラスの必須フィルタ
キー日付変数	ユニバースのパラメータ

各必須フィルタには、@Prompt 関数が期待される値の一覧を表示するための参照オブジェクトとして、2 つのディメンションオブジェクトが作成されます。ユニバースでは、値の一覧ディメンションが非表示にされています。これらのディメンションは、プロンプトが正しく機能するために必要なもので、削除しないでください。移動や変更は、慎重に行なう必要があります。

変数のデフォルト値は、一次キー、persistent/not persistent、デフォルト値の各パラメータを使用してフィルタの @Prompt 関数で定義されます。@Prompt 関数構文は、ユニバースのフィルタの[プロパティ]ページに表示できます。

BW 変数と Web Intelligence ユーザによって定義されたフィルタとの競合を回避するために、SAP 変数定義に含まれるオブジェクトは、オブジェクトプロパティの [詳細] ページの [条件項目で使用可能] オプションを解除して生成します。これによって、Web Intelligence ユーザが、[フィルタ] ペインで SAP 変数に含まれるディメンションを含めることが制限されます。

例 SAP BW 変数に対して生成された WHERE 句

この例は、ディメンションオブジェクト Customer2 の BW 変数に対して生成された WHERE 句を示しています。変数に対して生成された WHERE 句の構文は、フィルタのプロパティページに表示されます。

```
<FILTER KEY="[Z_VAR002]">
  <CONDITION OPERATORCONDITION="Equal">
    <CONSTANT TECH_NAME="@Prompt(
      'Customer Variable Single Value Mandatory',
      'A',
      'Customer2¥LovCustomer Variable Single Value MandatoryBase',
      mono,
      primary_key)" />
    <CONDITION>
  </FILTER>
```

プロンプトテキストは、BW 変数名から生成されます。テキストを編集して、さらに説明的にすることができます。

"Customer2¥LovCustomer Variable Single Value MandatoryBase"は、値の一覧の作成に使用された非表示ユニバースオブジェクトの名前です。

注

クラスの名前を変更したり、値の一覧 オブジェクトを別のフォルダに移動した場合、フィルタキーの構文を更新する必要があります。

8.5.1.4.2 変数と値の一覧のサポート方法

BEx クエリには、10 超の変数を含めることができます。つまり、10 以上の値の一覧をロードできます。値の一覧のロードおよび最新表示は、パフォーマンスに重大な影響を及ぼす場合があります。変数を含むクエリのクエリパフォーマンスを向上するために、次のオプションを使用できます。

- ・ キー日付を除くすべての SAP BW 変数は、ユニバースの生成時に必須フィルタにマップされます。デフォルトでは、階層ノード変数以外のフィルタオブジェクトは値の一覧に関連付けられません。オブジェクトのプロパティページで、値の一覧を明示的に関連付ける必要があります。
- ・ オプション変数は、オプションプロンプトとして生成されます。オプションプロンプトは、クエリ実行時に値の一覧を自動的にロードしません。
- ・ 値の一覧プロパティの依頼検索オプションによって、クエリの実行時に空の値の一覧が表示されます。ユーザは、値の一覧に返される値の数を制限する検索条件を入力します。

値の一覧の依頼検索オプションを有効にするには、値の一覧を適用するオブジェクトの[オブジェクトのプロパティ]ページで値の一覧プロパティを編集します。

注

カスケード値の一覧 では、依頼検索はサポートされていません。

関連項目

- ・ 466 ページの[OLAP ユニバースに含まれるオプションプロンプト](#)

8.5.1.4.3 ユニバースにおけるキー日付変数のサポート方法

BEx クエリのキー日付変数では、時間依存データの日付を指定できます。キー日付は、取得したディメンションのデータに影響を及ぼすことができます。たとえば、時間の経過に伴って製品説明を変更できます。キー日付は、階層構造に影響を及ぼすことができます。たとえば、ある年には特定のコストセンタを“レベル 01”に、別の年には“レベル 02”に配置することができます。

キー日付変数は、特殊な SAP BW 変数です。これは、ユーザが入力した日付値が BW クエリのディメンションに含まれないからです。キー日付は、クエリのプロパティです。

BEx クエリでは、キー日付変数は、次の 2 種類の使用目的で定義できます。

- ・ 特定階層に有効な日付を指定して、その階層のみに影響を及ぼす。
- ・ 完全なクエリの日付を指定する。この場合、クエリに設定されたキー日付は、以下に影響を及ぼします。
 - ・ 時間依存マスターデータ
 - ・ 為替レート
 - ・ 階層の一覧
 - ・ 時間依存階層構造

注

ユニバースでは、キー日付の使用はユニバース全体に限定されています。したがって、ユニバースで生成されたキー日付は、その他すべての SAP 変数およびデータに影響を及ぼします。

SAP NetWeaver BW がサポートするキー日付変数は、BW クエリごとに 1 つのみです。したがって、ユニバースに含まれるキー日付変数は、1 つのみです。

キー日付変数には、必須のものとオプションのものがあります。デフォルト値を設定することもできます。デフォルト値が定義されておらず、ユーザが値を入力しない場合、クエリは、現在のシステム日付を使用します。

クエリのキー日付の変数プロパティは、次の表で説明されているように、5 つのユニバースパラメータにマップされます。

パラメータ	説明
KEYDATE_ENABLED	ユニバースでキー日付を有効にする場合、[はい]に設定します。
KEYDATE_NAME	キー日付変数の技術名称です。
KEYDATE_CAPTION	ユーザに値の入力を要求する際に表示される、キー日付変数のキャプションです。
KEYDATE_DEFAULT_VALUE	キー日付のデフォルト値が存在する場合、デフォルト値です。
KEYDATE_MANDATORY	ユーザによる値の入力が必須の場合、またはデフォルト値を使用する場合、[はい]に設定します。

クエリを実行すると、Web Intelligence がすべてのクエリに対して同じキー日付を提案します。ユーザは、キー日付を変更できます。使用するキー日付の管理には、[キー日付のプロパティ]ダイアログボックスを使用できます。ユーザは、その他の種類の変数プロンプトが表示される前に、キー日付を入力するように要求されます。

8.5.1.4.4 ユニバースにおける階層変数と階層ノード変数のサポート方法

階層変数は、クエリで使用する階層についてユーザにプロンプトを表示するために使用されます。Web Intelligence ユーザは、クエリおよびレポートを作成して、任意の階層からメンバーを抽出および表示できます。

階層変数がオプションで、ユーザがプロンプトを空のままにした場合、レポートでは階層は使用されません。

レポートには、選択した階層に依存しない、階層レベルの最大数が含まれています。結果セットで返されない階層レベルは、レポートで空です。

階層ノード変数は、クエリで階層の最上位ノードとして定義されるべきノードについてユーザにプロンプトを表示します。

クエリに階層と階層ノード変数の両方が含まれている場合、Web Intelligence ユーザは、まず使用できる階層の一覧で階層を選択する必要があります。次に、階層ノードを選択します。使用できる階層ノードの一覧には、すべての階層の階層ノードが表示されます。この一覧は、選択した階層に基づいてフィルタが適用されていません。ユーザは、正しい階層からノードを選択する責任を負います。異なる階層から階層ノードを選択した場合、レポートが空になる可能性があります。

関連項目

- 506 ページの[階層のマップと使用方法](#)

8.5.2 Essbase キューブをユニバースコンポーネントにマップする方法

ユニバースデザインツールは、Essbase アウトラインを同等のクラスとオブジェクトにマップすることにより、Essbase キューブからユニバースを作成します。接続を作成するときに、キューブデータソースを特定できます。

Essbase エイリアステーブルには、アウトライン内のディメンション、レベル、およびメンバーに関する一連の代用名が定義されています。ユニバースデザインツールは、Essbase データソースへの接続を作成したときに選択されたエイリアステーブルに基づいた名前を使用して、ユニバースを生成します。

Essbase アウトラインでは、メジャーはディメンションとして定義されています。Essbase データソースへの接続を作成するときに、メジャーディメンションとして使用するディメンションを選択します。ユニバースデザインツールは、ディメンションのメンバーをユニバース内のメジャーとして生成します。

任意のディメンションにおいて、複数のレベルの階層がサポートされています。階層の最大値は、ディメンションごとに定義できます。

次のテーブルには、各 Essbase アウトライン要素に対して、ユニバース内で作成されるオブジェクトが示されています。

Essbase オブジェクト	作成されるユニバース オブジェクト
ディメンション	ディメンションの世代を含むクラス
世代	2 つの詳細オブジェクト(1 つはキャプション用、もう 1 つは名前用)を含むディメンション クラス内のオブジェクトです。
メジャー ディメンション	ユニバース接続内のメジャー ディメンションとして選択されたディメンションに基づいて名前の付けられたクラス。通常は、メジャー クラスまたはアカウント クラスです。
メジャー	メジャー クラスまたはサブクラス内にあるメジャー オブジェクトです。メジャーは、Essbase アウトライン内の構造に一致するクラスおよびサブクラスの構造を使用して作成されます。

メジャーは、集計投影関数をデフォルトで委任したデータベースに設定して生成されます。Web Intelligence レポートを更新するとき、メジャーの集計はデータベースサーバに委任されます。

関連項目

- 442 ページの[OLAP データソースへの接続](#)
- 454 ページの[依頼されたデータベース投影関数](#)

8.5.3 MSAS キューブをユニバースコンポーネントにマップする方法

ユニバースデザインツールは、MSAS 構造を同等のクラスとオブジェクトにマップすることにより、MSAS キューブからユニバースを作成します。接続を作成するときに、キューブデータソースを指定します。

次のテーブルには、各 MSAS アウトラインエレメントに対して、ユニバース内で作成されるオブジェクトが示されています。このマッピングは、MSAS 仮想キューブとローカルキューブ(.cub ファイル)だけでなく、MSAS 標準キューブにも適用されます。

MSAS オブジェクト	作成されるユニバース オブジェクト
ディメンション	ディメンションのオブジェクトを含むクラスです。
表示フォルダ(MSAS 2005)	ディメンションクラス内にあるサブクラスです。
階層	対応するディメンションクラス内にあるサブクラス、または対応する表示フォルダクラス内にあるサブクラスです。
属性(MSAS 2005)	対応するディメンションクラス内にあるサブクラス、または対応する表示フォルダクラス内にあるサブクラスです。
メジャー	すべてのメジャーオブジェクトを含むメジャークラスです。メジャーオブジェクトは、メジャークラス内またはメジャーグループのサブクラス内で作成されます。
メジャーグループ(MSAS 2005)	メジャークラス内にあるサブクラスです。
レベル	ディメンションクラスまたはサブクラス内にあるオブジェクトとすべてのサブレベルの集合体を代表する“すべてをレベル”オブジェクトです。
レベルプロパティ	適用先のレベルオブジェクトの詳細です。

メジャーは、集計投影関数をデフォルトで委任したデータベースに設定して生成されます。Web Intelligence レポートを更新するとき、メジャーの集計はデータベースサーバに委任されます。

関連項目

- ・ 442 ページの [OLAP データソースへの接続](#)
- ・ 454 ページの [依頼されたデータベース投影関数](#)

メタデータソースからのユニバースの操作

9.1 メタデータソースからのユニバースの生成について

ユニバースデザインツールの[メタデータエクスチェンジ]によって、他のデータウェアハウス製品で生成された XML ファイルからユニバースを作成することができます。XML ファイルの内容を分析し、メタデータ情報を抽出して、それをクラス、オブジェクト、テーブル、列、カスタム階層、結合を含んだ BusinessObjects メタデータに変換します。次に、新しい BusinessObjects ユニバースを作成します。また、他のメタデータソースからユニバースを作成することもできます。

ユニバースデザインツールを使用して、次のメタデータデータソースからユニバースを作成できます。

メタデータソース	名前
次の規格に準拠した XML ファイル	<ul style="list-style-type: none">Common Warehouse Model(CWM 1.0)Common Warehouse Model OLAP(CWM OLAP)Oracle Warehouse BuilderBusinessObjects Data IntegratorIBM DB2 Data Warehouse CenterIBM DB2 Cube Views
データベースビュー	Oracle アナリティクスワークスペース

ユニバースデザインツールを使用して、特定の XML メタデータソースを使用するユニバースを更新し、ユニバースを DB2 Cube Views(DB2CV)XML 形式にエクスポートすることもできます。

9.2 概要

メタデータ形式を選択するには、[メタデータエクスチェンジ]パネル([ファイル]>[メタデータエクスチェンジ])を使用します。この形式は、ターゲットメタデータソースファイルによって使用されます。詳細については、メタデータソースの選択に関する節を参照してください。

形式を選択したら、Universe Builder ウィザードに従ってターゲットデータベースを選択し、ユニバースの作成に使用する構造を選択します。次に、ターゲット接続を選択し、ユニバースを生成します。

ユニバース作成のプロセスは、すべての XML メタデータソースで同じです。XML メタデータソースからのユニバースの作成については、517 ページの「[XML ソースからのユニバースの作成](#)」を参照してください。

Oracle アナリティクスワークスペースデータソースに対するユニバースの作成プロセスは異なります。接続を選択すると、選択したメタデータソースに固有のユニバース作成パネルが表示されます。使用できる各メタデータソースについては、それぞれの節で説明されています。

メタデータソースからユニバースを作成すると、他のユニバースの場合と同様に、任意のユニバースコンポーネントを変更できます。

ユニバースを保存して Central Management Server にエクスポートします。CMS にエクスポートすると、Web Intelligence ユーザはそのユニバースを使用してクエリとレポートを作成できるようになります。

9.3 ユニバース作成の概要

メタデータ形式を選択するには、メタデータエクスチェンジパネル([ファイル]>[メタデータエクスチェンジ])を使用します。この形式は、ターゲットメタデータソースファイルによって使用されます。詳細については、516 ページの「[メタデータソースの選択](#)」を参照してください。

形式を選択したら、Universe Builder ウィザードに従ってターゲットデータベースを選択し、ユニバースの作成に使用する構造を選択します。次に、ターゲット接続を選択し、ユニバースを生成します。

ユニバース作成のプロセスは、すべての XML メタデータソースで同じです。XML メタデータソースからのユニバースの作成については、517 ページの「[XML ソースからのユニバースの作成](#)」を参照してください。

Oracle アナリティクスワークスペースデータソースからのユニバースの作成プロセスは異なります。接続を選択すると、Oracle アナリティクスワークスペースに固有のユニバース作成パネルが表示されます。データベースでビューを作成し、そのビューからユニバースを作成します。

メタデータソースからユニバースを作成すると、他のユニバースの場合と同様に、任意のユニバースコンポーネントを変更できます。

ユニバースを保存して Central Management Server にエクスポートします。CMS にエクスポートすると、Web Intelligence ユーザはそのユニバースを使用してクエリとレポートを作成できるようになります。

9.4 メタデータソースの選択

メタデータエクスチェンジパネル([ファイル]>[メタデータエクスチェンジ])で、ユニバースを作成または更新するためのメタデータソースを選択します。また、DB2CV XML 形式にエクスポートするユニバースも選択できます。

メタデータエクスチェンジパネルでは、次のオプションを使用できます。

メタデータエクスチェンジのオプション	説明
ビューからのユニバースの作成	ドロップダウンリストからメタデータソース形式を選択します。ユニバースの作成に使用するソース XML ファイルまたはデータベースビューです。ユニバースの作成ウィザードを使用すると、メタデータソースの接続の選択、ユニバースにマップするメタデータコンポーネントの選択、および最終的なユニバースの生成という作業を手順に沿って進めることができます。
ユニバースの更新元	更新済みのメタデータソースを選択します。ユニバースの作成に使用されたメタデータソースです。このソースは更新されているため、同じ修正を適用してユニバースを更新する必要があります。ユニバースの更新ウィザードを使用すると、ユニバースの更新に必要な作業を手順に沿って進めることができます。
ユニバースのエクスポート先	ユニバースのエクスポート先のメタデータ形式を選択します。たとえば、DB2CV XML 規格を選択して、その形式でユニバースを保存できます。

9.5 XML ソースからのユニバースの作成

メタデータエクスチェンジ([ファイル]>[メタデータエクスチェンジ])から使用できる OLAP Universe Builder ウィザードに従って、XML メタデータソースからユニバースを作成します。ユニバースを生成する前に、ユニバース接続と生成のオプションを設定できます。

関連項目

- 517 ページの[XML メタデータソース](#)

9.5.1 XML メタデータソース

ユニバースは、次のデータソース規格に準拠した XML ファイルから作成できます。

- Common Warehouse Model(CWM Relational 1.0)
- Common Warehouse Model OLAP(CWM OLAP)

- ・ Oracle Warehouse Builder(Oracle WB)
- ・ Data Integrator
- ・ IBM DB2 Data Warehouse Center(IBM DB2 DWC)
- ・ IBM DB2 Cube Views

メタデータエクスチェンジ([ファイル]>[メタデータエクスチェンジ])から使用できる OLAP Universe Builder ウィザードに従って、XML メタデータソースからユニバースを作成します。

9.5.2 XML メタデータソースからユニバースを生成する

- 1 [ファイル]>[メタデータエクスチェンジ]を選択します。
メタデータエクスチェンジパネルが表示されます。
- 2 [ユニバースの作成元]ドロップダウンリストボックスからメタデータの形式を選択します。
[OK]をクリックします。
Universe Builder ウィザードが開始されます。
[次へ]をクリックします。
[XML ファイルソース]ページが表示されます。
- 3 [参照]ボタンをクリックして XML ソースファイルを選択します。これは、ユニバースの生成に使用するファイルです。
[次へ]をクリックします。
[データベースの選択]ページが表示されます。
- 4 ソースデータベースをクリックします。
[次へ]をクリックします。
[ユニバース要素]ページが表示されます。利用可能なデータベースのテーブルと列が左枠に一覧表示されます。
- 5 1 つまたは複数のテーブルおよび列を選択して右向き矢印をクリックすると、右枠に追加されます。右枠のテーブルと列は、生成されたユニバースに含まれます。矢印ボタンを使用して、必要に応じてユニバース枠にテーブルを追加、または削除できます。
[次へ]をクリックします。
[接続とユニバースのプロパティ]ページが表示されます。ユニバースデザインツールで使用する接続が一覧表示されます。
- 6 [接続]リストで接続をクリックします。この接続は、ユニバースがデータの取得に使用するデータソースへの接続になります。
ユニバース名を入力します。

オプションのチェックボックスをオンまたはオフにします。[詳細設定]ボタンをクリックして、トレースログファイルと XML ソースファイルのオプションを設定します。

[次へ]をクリックします。

[ユニバース生成の要約]ページが表示されます。このページには、ウィザードで選択したオプションの要約が一覧表示されます。

[完了]をクリックします。

生成されたユニバースが、ユニバースデザインツールのユニバース枠と構造枠に表示されます。

9.5.3 接続とユニバースのオプションの選択

メタデータ Universe Builder ウィザードの[接続とユニバースの作成]ページには、次のようなオプションがあります。

ウィザードのページ	ユニバースのオプション	説明
ユニバース作成の設定	接続の選択	一覧には、ユニバースデザインツールで使用できる接続が表示されます。これは、ターゲット RDBMS への接続です。
	ユニバース名	生成されるユニバースの名前です。
	ユニバースの自動保存	オンにすると、ユニバースの作成時に保存されます。
	既存のユニバースの置換	オンにすると、[ユニバースの自動保存]がオンで同じ名前のユニバースがすでに存在するときに、既存のユニバースが新しいユニバースによって置き換えられます。
詳細設定	[一般]タブ トレース	トレースフォルダへのパスです。このフォルダには、ユニバースが生成されときのログファイルが格納されます。フォルダを参照して選択できます。

ウィザードのページ	ユニバースのオプション	説明
	[ファイルの場所]タブ デフォルトの XML ソースファイル フォルダ	ユニバースの作成に使用される XML ファイルを格納するデフォルトフォルダへのパスです。フォルダを参照して選択できます。
	[ファイルの場所]タブ パラメータファイル	パラメータファイルを格納するデフォルトフォルダへのパスです。これらのファイルは、ユニバースの作成時に作成されます。これらのファイルは選択されたメタデータを格納および参照し、他のユニバースの作成と更新に再利用します。パラメータファイルは、選択したメタデータ自体を格納するわけではありません。元の XML ファイルを介して、選択したメタデータへの橋渡しを行うフィルタです。フォルダを参照して選択できます。

9.5.4 XML メタデータソースからユニバースを更新する

- 1 [ファイル]>[メタデータエクスチェンジ]を選択します。メタデータエクスチェンジパネルが表示されます。
- 2 [ユニバースの更新元]ドロップダウンリストからメタデータ形式を選択します。[OK]をクリックします。Universe Builder ウィザードが起動します。[次へ]をクリックします。XML ソースファイルのページが表示されます。
- 3 [参照]ボタンをクリックして XML ソースファイルを選択します。これは、ユニバースの更新に使用するファイルです。[次へ]をクリックします。[データベースの選択]ページが表示されます。
- 4 ソースデータベースをクリックします。[次へ]をクリックします。[ユニバースファイル]ページが表示されます。[参照]ボタンをクリックして、ユニバースを選択します。このユニバースを、選択した XML メタデータソースによって更新します。[次へ]をクリックします。[ユニバース要素]ページが表示されます。使用可能なデータベースのテーブルと列が左枠に一覧表示されます。追加または修正されたテーブルには、赤いチェックマークが付加されます。
- 5 1 つまたは複数のテーブルおよび列を選択して右向き矢印をクリックすると、右枠の修正済みのテーブルに追加されます。右枠のテーブルと列は、生成されたユニバースに含まれます。矢印ボタンを使用して、必要に応じてユニバース枠にテーブルを追加、または削除できます。[次へ]をクリックします。接続およびユニバースのプロパティページが表示されます。ユニバースデザインツールで使用できる接続が一覧表示されます。それぞれについて、519 ページの「[接続とユニバースのオプションの選択](#)」で説明します。
- 6 [接続]リストで接続をクリックします。この接続は、ユニバースがデータの取得に使用するデータソースへの接続になります。ユニバース名を入力します。オプションのチェックボックスをオンまたはオフにします。[詳

細設定]ボタンをクリックして、トレースログファイルとXML ソースファイルのオプションを設定します。[次へ]をクリックします。[ユニバース生成の要約]ページが表示されます。このページには、ウィザードで選択したオプションの要約が一覧表示されます。[完了]をクリックします。更新されたユニバースは、ユニバースデザインツールのユニバース枠と構造枠に表示されます。

9.6 ユニバースの DB2CV へのエクスポート

ユニバースは、IBM DB2 Cube Views XML 形式ファイルにエクスポートできます。

まず、ユニバースの定義が IBM DB2 Cube Views XML 形式でコンパイルする XML ファイルにエクスポートされます。このファイルはその後、API または OLAP Center ツールによって IBM DB2 Cube Views へとロードされます。IBM DB2 Cube Views は XML ファイルからメタデータを読み取り、次のクエリ最適化に適した自動サマリー表(AST)を推奨します。

関連項目

- ・ 521 ページの [ユニバースをエクスポートする際の前提条件と制限事項](#)
- ・ 522 ページの [ユニバースメタデータの識別](#)

9.6.1 ユニバースをエクスポートする際の前提条件と制限事項

ユニバースをXMLファイルへとエクスポートするために必要な要件は次のとおりです。

ユニバースレベルの制限事項

- ・ 各ユニバースは各キューブモデルにエクスポートされる。
- ・ ユニバースは、1 つのファクトテーブルを持つ単独のスノーフレーク型スキーマと一致させる必要がある。
- ・ ユニバースには少なくとも1つのメジャーが必要。
- ・ ユニバース間のリンクはサポートされない。
- ・ コンテキストは考慮されず、エクスポートもされない。
- ・ カスタム階層レベルは、同じクラスにグループ化する必要がある。

クラスとオブジェクト

- ・ @ 関数のなかでサポートされるのは @SELECT 関数のみ。他の @ 関数はエクスポートでマップされません。
- ・ オブジェクトの定義の WHERE フィールドで指定した条件はエクスポートされない。注: 条件は最適化では使用されないため、DB2 Cube Views オブジェクトではサポートされません。
- ・ 複数パラメータの集計関数はエクスポートされない。

- ・ 各クラスに、同じディメンションテーブルで作成されたオブジェクトを含める必要がある。
- ・ IBMDB2CV 内の同じディメンションに参照されているすべてのオブジェクト(IBMDB2CV の属性)は、ユニバースの同じクラスにグループ化される必要がある。他のクラスの他のメジャーは、自動的に IBMDB2CV のファクトオブジェクトに追加されます。

結合

結合の右列または左列がユニバースのオブジェクトに一致しない場合、この列の属性が自動的に作成され、列テーブルを含んでいるディメンション(またはファクト)に追加されます。

9.6.2 ユニバースメタデータの識別

この節では、一致する多次元スキーマがIBM DB2 Cube Viewsに存在しないオブジェクトを、ユニバース定義をXMLファイルにエクスポートする際にどのように識別し、処理するかについて説明します。

リレーショナルメタデータを含むユニバース

ユニバースは、多次元のデザイン制約を持たないリレーショナルメタデータをベースとしています。ユニバース内のすべてのオブジェクトが対応するIBM DB2 Cube Views オブジェクトに一致している必要はなく、IBM DB2 Cube Views の多次元ルールには準拠しません。

リレーショナル構造と正しく一致させるため、BusinessObjects UMB で自動検出機能プロセスを実行し、IBM DB2 Cube Viewsで必要かつ適切なメタデータを識別し、定義してください。影響を受ける多次元オブジェクトは次のとおりです。

ファクト

IBM DB2 Cube Views ファクトオブジェクトは、ユニバースで見つかったメジャーセットから自動的に作成されます。

ディメンション

ファクトとして識別されていないテーブルはディメンションテーブルと仮定されます。IBM DB2 Cube Views ディメンションは、BusinessObjectsクラスから直接引用されます。

クラス内にあるオブジェクトはIBM DB2 Cube Views ディメンションの属性を決定します。このクラス内にあるBusinessObjectsオブジェクトで参照されたテーブルは、オブジェクトのSELECTフィールドを分析することによって検出されます。

属性

属性は、ユニバース内にあるテーブルの列から直接引用されるわけではありません。まず候補属性が検出され、次の情報をもとに識別されます。

- ・ クラス内のBusinessObjectsオブジェクト
- ・ @SELECT 構文により他の BusinessObjects オブジェクトの SELECT フィールドで参照されている BusinessObjects オブジェクト
- ・ 結合に含まれる列

属性の関係

ユニバース内の詳細とディメンションの関係は、IBM DB2 Cube Views では機能依存の属性関係型(Functional Dependency)へと変換されます。

結合

結合とそのプロパティは、ユニバース構造から直接読み取られます。

メジャー

メジャーオブジェクト用にすべてのクラスが検索されます。メジャーが同じファクトテーブルで作成されたわけではない場合、このメジャーは無視されます。

階層

DB2 Cube Views 内の階層はディメンションオブジェクトにリンクしており、レベルはすべてこのディメンションのメンバですが、ユニバース側では同様ではありません。ユニバースでは、Business Objects のあらゆるクラスから派生するレベルをカスタム 階層に含めることができます。階層は次のように処理されます。

- ・ ユニバースがデフォルトの階層だけを使用している場合、IBM DB2 Cube Views へのエクスポート用の階層は、各クラス内のオブジェクトを使い、オブジェクトから推定されます。
- ・ ユニバースにカスタム 階層が含まれている場合、この階層は変更されずにそのままエクスポートされます。

9.6.3 DB2CV XML ファイルへのユニバースのエクスポート

BusinessObjectsユニバースを IBM DB2 Cube Views XML ファイルにエクスポートする方法は次のとおりです。

9.6.4 DB2CV メタデータへのユニバースのマップ

この節では、ユニバース構造とIBM DB2 Cube Views構造間のマップについて詳しく説明します。

次の各節では、ユニバースを XML ファイルにエクスポートする際、ユニバースから IBM DB2 Cube Views へとマップされる構造について、詳しく説明します。

ユニバースからキューブモデルへ

ユニバースからキューブモデルへのマップは次のとおりです。

ユニバースのプロパティ	キューブプロパティ
ショートネーム(ファイル名)	キューブファイル名

ユニバースのプロパティ	キューブプロパティ
ユニバース名(ロングネーム)	ビジネス名 デフォルトではユニバースのショートネーム(<universe short name>)です
説明	コメント
ファクトテーブル名	factsRef
クラスの一覧	dimensionRef
ファクトテーブルを含んでいる結合の一覧	joinRef

クラスからディメンションへ

クラスからディメンションへのマップは次のとおりです。

クラスのプロパティ	ディメンションのプロパティ
名前	名前とビジネス名
説明	コメント
ディメンションオブジェクトと詳細オブジェクトの一覧。 メジャーのマップについては、次のメジャーからメジャーへの表を参照してください。	attributeRef
クラスから継承されたディメンションテーブル間の結合	joinRef
階層	階層がユーザ定義である場合、IBM DB2 Cube Views から要求されているものと同じディメンションにある階層レベルがすべて取得されるよう、ディメンションが変更されます。 このディメンションのプロパティは hierarchyRef となります。

ファクトテーブルからファクトへ

ファクトテーブルからファクトへのマップは次のとおりです。

ファクトテーブルのプロパティ	ファクトのプロパティ
ファクトテーブル名 この名前は[ユニバース]-[エクスポート]パネルで表示される[ファクト]ボックスに手動で入力します。デフォルトで入力されている名前 Facts_<universe name> を使用することもできます。	名前とビジネス名
テーブルの説明	コメント
ユニバースにあるメジャーの一覧	measureRef
メジャーが参照している列とオブジェクトの一覧	attributeRef

メジャーからメジャーへ

メジャーからメジャーへのマップは次のとおりです。

メジャーのプロパティ	メジャーのプロパティ
名前	名前とビジネス名
説明	コメント
SELECT 句で参照される列とオブジェクト	sqlExpression 列
SELECT 句の形式	sqlExpression テンプレート
集計関数	集計関数

ディメンションオブジェクトと詳細オブジェクトから属性へ

ディメンションオブジェクトと詳細オブジェクトから属性へのマップは次のとおりです。

ディメンションオブジェクトと詳細オブジェクト	属性
名前	名前とビジネス名
説明	コメント
SELECT 句で参照される列とオブジェクト	sqlExpression 列
SELECT 句の形式	sqlExpression テンプレート

ディメンションと詳細の関係から属性の関係へ

ディメンションと詳細の関係から属性の関係へのマップは次のとおりです。

ディメンションと詳細の関係	属性の関係
ディメンション名 + 詳細名 結合文字は _ です。	名前とビジネス名
ディメンション	左の属性
詳細	右の属性

デフォルトの階層から階層へ

デフォルトの階層から階層へのマップは次のとおりです。

デフォルトの階層	階層
名前	名前とビジネス名
オブジェクトの一覧 階層の一部として詳細オブジェクトを使用することはできません。	AttributeRef

注

カスタム 階層が存在しない場合、階層としてクラスが使用されます。

カスタム 階層から階層へ

カスタム 階層から階層へのマップは次のとおりです。

カスタム 階層	階層
名前	名前とビジネス名
オブジェクトの一覧	attributeRef

結合から結合へ

結合から結合へのマップは次のとおりです。

結合	結合
左のテーブル名 + 右のテーブル名 結合文字は _ です。	名前とビジネス名
左列	左の属性
右列	右の属性
複雑な表現式の場合、単純な表現式ではそれぞれ、 左列と右列が識別されます。 単純な表現式はそれぞれ左列と右列で同一です	単純な表現式はそれぞれ1組の属性ペアにマップされます。

9.6.5 特定の SQL 表現式のマップ

SQL 表現式には、独自のエクスポートの方式でマップされるものがあります。ここでは、次の SQL 表現式について説明します。

- ・ メジャー用の SELECT 表現式
- ・ @AggregateAware 関数
- ・ 複雑な結合の表現式
- ・ シータ結合

- ・ ショートカット結合

メジャー用の SELECT 表現式

BusinessObjects UMB は、メジャーの SELECT 表現式から次の情報を取得します。

- ・ メジャーに含まれているテーブルと列の検出、および sqlExpression:column へのマップ
- ・ 集計関数の識別
- ・ 表現式の決定と sqlExpression:template へのマップ

@AggregateAware 関数

オブジェクトが @AggregateAware 関数を含んでいる場合、考慮されるのはこの関数の最後のパラメータだけです。これは、この関数で使用する集計の最低レベルを含んでいる表現式です。以下はその例です。

たとえば、ユニバース内のメジャー表現の @AggregateAware 関数は次のようになります。

```
@AggregateAware(
sum(AggregatedTable1.Sales_revenue),
sum(AggregatedTable2.Sales_revenue),
sum(Fact_Table.Amount_sold))
```

IBM DB2 Cube Views にマップされる表現式は次のとおりです。

```
sum(Fact_Table.Amount_sold))
```

複雑な結合の表現式

ユニバースの複雑な結合の表現式には、次の形式を使用することができます。

```
LeftTable.Column=RightTable.Column
```

複雑な結合の場合、これらの種類の表現式はAND演算子でリンクされます。BusinessObjects UMB は、複雑な結合内の各表現式を、同じ結合内でIBM DB2 Cube Viewsの属性ペアへとマップします。

シータ結合

シータ結合は、2 つの IBM DB2 Cube Views 結合へと分割されます。この IBM DB2 Cube Views 結合で、BETWEEN 演算子は <= および >= に置き換えられます。以下はその例です。

ユニバースの結合に次の表現式があるとしたします。

```
Customer.age between Age_group.age_min and Age_group.age_max
```

この結合は、次の表現式を持つ2つの結合に分割されます。

```
Join1: Customer.age >= Age_group.age_min
Join2: Customer.age <= Age_group.age_max
```

ショートカット結合

IBM DB2 Cube Viewsにはエクスポートされません。ユニバースのショートカット結合は代替パスを意味し、中間テーブルを考慮しないことでクエリのパフォーマンスを改善します。ショートカット結合は、キューブモデル内でループを作成するため、エクスポートされません。

9.7 Oracle アナリティクスワークスペース

Oracle OLAP ユニバースビルダウィザードの手順に従って、ユニバースを作成できます。メタデータエクスチェンジパネル([ファイル]>[メタデータエクスチェンジ])から、Oracle OLAP ユニバースビルダウィザードに接続します。

次に、Oracle OLAP Universe Builder ウィザードによるユニバースの作成方法の概要を示します。

メタデータエクスチェンジを起動して、[ユニバースの作成元]ドロップダウンリストから[Oracle OLAP]を選択します。

Oracle OLAP ユニバースビルダウィザードが起動します。次の手順を実行します。

- ・ ユニバースの作成に使用する InfoProvider に接続する。
- ・ データベースを選択する。
- ・ ターゲットメタデータソースにするキューブを選択する。
- ・ キューブメタデータに基づいたビューを作成する。
- ・ ビューを基にするユニバースを生成します。

既存のビューからユニバースを作成することもできます。

9.7.1 OLAP キューブから生成されたユニバースについて

Oracle OLAP Universe Builder でユニバースを作成すると、SQL が Oracle アナリティクスワークスペースにアクセスするよう、自動的に設定されます。BusinessObjects Oracle OLAP Universe Builder は、次の主なタスクを実行します。

- ・ ユニバースに、リレーショナルファクトビューを実際のビューまたは派生テーブルとして挿入する。
- ・ ディメンションレベルと階層を示すエイリアスを追加する。
- ・ リレーショナルビューを、標準結合とショートカット結合を使用してディメンションテーブルに結合する。結合の式は、このソリューションに特有です。
- ・ 各キューブディメンションにオブジェクトのクラスを、ディメンションの各レベルにオブジェクトを作成する。
- ・ ディメンションに複数の階層がある場合は、各階層のサブクラスを作成する。ビュー定義とユニバースでは、複数階層ディメンションを使用できます。
- ・ 複数階層ディメンションの結果として生じたオブジェクトの非互換性を解決するために、集計ナビゲーションを定義する。
- ・ 集計ナビゲーションを処理するために AggregateAware 関数を使用するオブジェクト式を定義する。

- ・ 実際のディメンションメンバー(識別子)をマップするオブジェクトを、メンバー記述を示すオブジェクトの詳細に変換する。
- ・ メジャーオブジェクトを作成する。

9.7.2 Oracle OLAP 構造のユニバースコンポーネントへのマップ

この節では、Oracle OLAP キューブ構造からユニバースがどのように作成されるかを説明します。ここでは、生成されるユニバースの構造について説明し、マッピングプロセスに関する一般的な質問に回答します。

9.7.3 リレーショナルビューの分析

BusinessObjects Oracle OLAP Universe Builder は、OLAP_TABLE 関数を呼び出すビューを生成し、ビュー列をディメンションの階層とキューブのメジャーにマップします。生成されたスクリプトは次のような形式です。

```
CREATE VIEW BOBJ_FK_UNITS_CUBE_VIEW AS SELECT * FROM TABLE(OLAP_TABLE('GLOBAL_AW2.TEST DURATION session','",&LIMIT_MAP'
```

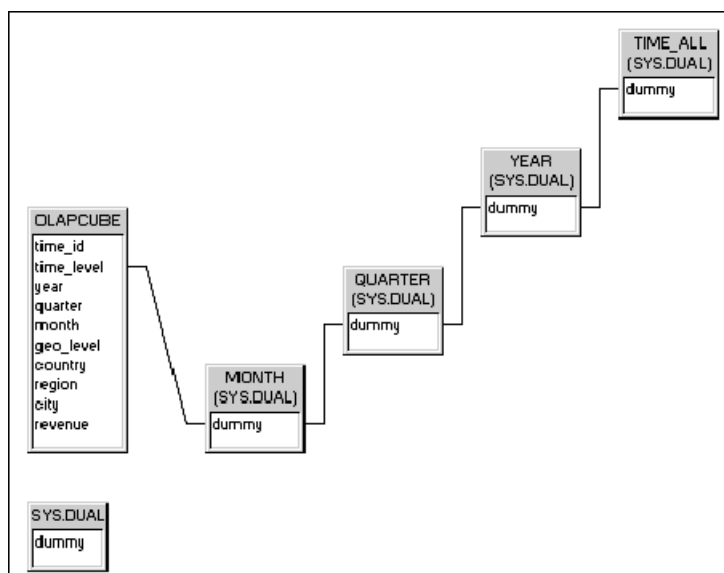
LIMIT_MAP は、OLAP_TABLE の limit_map パラメータのテキストを格納する変数です。このテキストは、Oracle OLAP Universe Builder によって生成されます。limit_map パラメータの例は次のとおりです。

```
DIMENSION GLOBAL_AW2.TEST!FK_TIME WITH
HIERARCHY GLOBAL_AW2.TEST!FK_TIME_PARENTREL (FK_TIME_HIERLIST ¥"CALENDAR¥")
LEVELREL FK_TIME_YEAR,FK_TIME_QUARTER,FK_TIME_MONTH
FROM GLOBAL_AW2.TEST!FK_TIME_FAMILYREL USING GLOBAL_AW2.TEST!FK_TIME_LEVELLIST
LEVELREL FK_TIME_YEAR_DESC,FK_TIME_QUARTER_DESC,FK_TIME_MONTH_DESC
FROM GLOBAL_AW2.TEST!FK_TIME_FAMILYREL USING GLOBAL_AW2.TEST!FK_TIME_LEVELLIST
LABEL GLOBAL_AW2.TEST!FK_TIME_LONG_DESCRIPTION
ATTRIBUTE FK_TIME_LEVEL FROM GLOBAL_AW2.TEST!FK_TIME_LEVELREL
DIMENSION GLOBAL_AW2.TEST!FK_CUSTOMER WITH
HIERARCHY GLOBAL_AW2.TEST!FK_CUSTOMER_PARENTREL (FK_CUSTOMER_HIERLIST ¥"MARKET_SEGMENT¥")
INHIERARCHY GLOBAL_AW2.TEST!FK_CUSTOMER_INHIER
LEVELREL null,null,null,FK_CUSTOMER_TOTAL_MARKET,FK_CUSTOMER_MARKET_SEGMENT,
FK_CUSTOMER_ACCOUNT,FK_CUSTOMER_SHIP_TO
FROM GLOBAL_AW2.TEST!FK_CUSTOMER_FAMILYREL USING GLOBAL_AW2.TEST!FK_CUSTOMER_LEVELLIST
LEVELREL null,null,null,FK_CUSTOMER_TOTAL_MARKET_DESC,FK_CUSTOMER_MARKET_SEGMENT_D01,
FK_CUSTOMER_ACCOUNT_DESC,FK_CUSTOMER_SHIP_TO_DESC
FROM GLOBAL_AW2.TEST!FK_CUSTOMER_FAMILYREL USING GLOBAL_AW2.TEST!FK_CUSTOMER_LEVELLIST
LABEL GLOBAL_AW2.TEST!FK_CUSTOMER_LONG_DESCRIPTION
HIERARCHY GLOBAL_AW2.TEST!FK_CUSTOMER_PARENTREL (FK_CUSTOMER_HIERLIST ¥"SHIPMENTS¥")
INHIERARCHY GLOBAL_AW2.TEST!FK_CUSTOMER_INHIER
LEVELREL null,null,null,FK_CUSTOMER_ALL_CUSTOMERS,
FK_CUSTOMER_REGION,FK_CUSTOMER_WAREHOUSE,null
FROM GLOBAL_AW2.TEST!FK_CUSTOMER_FAMILYREL USING GLOBAL_AW2.TEST!FK_CUSTOMER_LEVELLIST
LEVELREL null,null,null,FK_CUSTOMER_ALL_CUSTOMERS_DESC,
FK_CUSTOMER_REGION_DESC,FK_CUSTOMER_WAREHOUSE_DESC,null
FROM GLOBAL_AW2.TEST!FK_CUSTOMER_FAMILYREL USING GLOBAL_AW2.TEST!FK_CUSTOMER_LEVELLIST
LABEL GLOBAL_AW2.TEST!FK_CUSTOMER_LONG_DESCRIPTION
ATTRIBUTE FK_CUSTOMER_LEVEL FROM GLOBAL_AW2.TEST!FK_CUSTOMER_LEVELREL
MEASURE FK_UNITS_CUBE_UNITS AS NUMBER FROM GLOBAL_AW2.TEST!FK_UNITS_CUBE_UNITS
ROW2CELL OLAP_CALC
```

9.7.4 ユニバースでショートカット結合を使用する目的

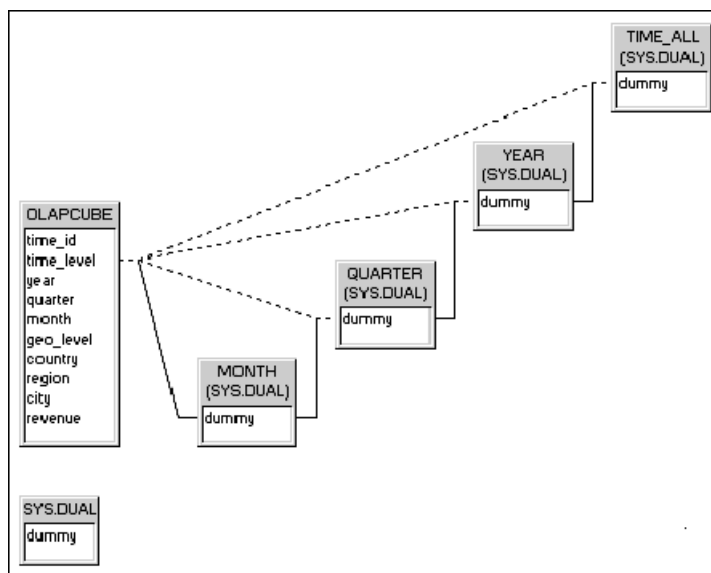
ショートカット結合を使用することにより、BusinessObjects は、各オブジェクトではなく各オブジェクトの組み合わせに対して SQL を生成します。

BusinessObjects は、クエリからテーブルを除外し、ある階層内で直接リンクされていない 2 つのテーブル間に“ショートカット”を設定できる場合にショートカット結合を使用します。たとえば、次のスキーマを基にします。



ショートカット結合が QUARTER および OLAPCUBE テーブルの間に定義される場合、BusinessObjects は、四半期ごとの収入を取得するために MONTH テーブル経由で結合する必要はありません。

最下位レベルのテーブルを除く時間階層内の各テーブルは、次に示されるようにショートカット結合によって OLAPCUBE.time_level に結合する必要があります。



結合式には OLAPCUBE から返された行を制限する式が含まれます。QUARTER の場合は、OLAPCUBE.time_level = 'QTR' がこれにあたります。ユニバースデザインツールが結合を許可するには、式は MONTH テーブルも参照する必要があります。これは生成しようとしている実際の結合式の一部ではないため、コメントの中に記述されます。完全な結合式は次のようになります。

```
/* QUARTER.DUMMY */ OLAPCUBE.time_level = 'QTR'
```

サンプルの時間階層のショートカット結合式の完全な一覧を次に示します。

結合テーブル	式
MONTH、OLAPCUBE	/* MONTH.DUMMY */ OLAPCUBE.time_level = 'MONTH'
QUARTER、OLAPCUBE	/* QUARTER.DUMMY */ OLAPCUBE.time_level = 'QTR'
YEAR、OLAPCUBE	/* YEAR.DUMMY */ OLAPCUBE.time_level = 'YEAR'
TIME_ALL、OLAPCUBE	/* TIME_ALL.DUMMY */ OLAPCUBE.time_level = 'ALL'

9.7.5 ユニバースコンポーネントにマップされた Oracle OLAP 構造について

目的のユニバースを取得して設定するために、Oracle OLAP Universe Builder はユニバースの次のようなオブジェクトを追加および設定します。

表示

Oracle OLAP Universe Builder は、ユニバースに、Oracle テーブル `sys.dual` だけでなくリレーショナルビューもテーブルとして挿入します。派生テーブルを使用する場合、派生テーブルはビューの定義(OLAP_TABLE 関数の選択部分)と共に挿入されます。

階層テーブル

リレーショナルビューに表示される各階層では、階層の各レベルに `sys.dual` のエイリアスが作成されます。エイリアスの名前はレベル名です。たとえば、ALL、YEAR、MONTH、および QUARTER の 4 つのレベルがある TIME ディメンションには、ALL、YEAR、MONTH、および QUARTER の 4 つのエイリアスを作成します。

複数階層テーブル

注

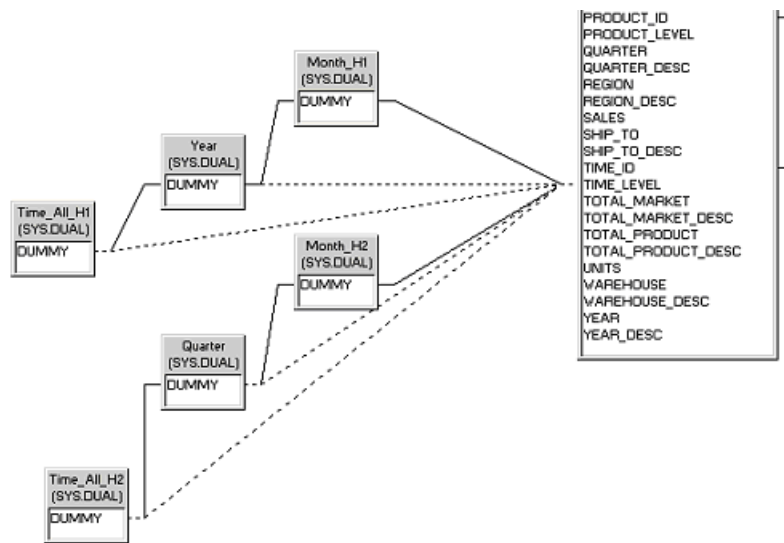
複数階層は特別なケースです。詳細については、この章の最終節を参照してください。

1 つのディメンションに複数の階層がある場合、一部の階層が同じレベルを共有する場合でも、各階層に異なるテーブルセットが作成されます。これは、共有レベルとして、階層と同じ数のエイリアスが作成されることを意味します。そのようなエイリアスに名前を付ける場合は、レベル名と階層名を連結します。以下はその例です。

ディメンション時間には、H1(All_Time、Year、Month)および H2(All_Time、Quarter、Month)の 2 種類の階層があります。

All_Time と Month は両方の階層で共有されるので、All_Time のエイリアスは All_Time_H1 と All_Time_H2 の 2 つになります。

また、Month のエイリアスは、Month_H1 と Month_H2 の 2 つになります。



ディメンション結合

- レベルを示すテーブルのそれぞれは、同じ階層内のすぐ下のレベルと結合されます。結合式は次のようになります。

```
/* Alias1.DUMMY=Alias2.DUMMY */ 1=1
```

ここで、Alias1 は 1 つのレベルを示し、Alias2 はその階層内の直上のレベルを示します。例

```
/* Quarter.DUMMY=Year.DUMMY */ 1=1
```

- 結合が標準になっている最下位レベル以外では、各テーブルはショートカット結合型を使用してビューに結合されます。結合式はビューから返される行をフィルタする値を定義し、型は次のようになります。

```
/* Alias.DUMMY */
```

```
VIEW.levelColumn = 'level_value'
```

ここで、Alias はエイリアス名、levelColumn はビュー内のレベルを示す列、level_value はレベル名に一致する列の値です。

例: MYVIEW は OLAP キューブを示すビュー、レベルを含む列は time_level、レベルの値は ALL、YEAR、QTR、MONTH です。

結合されたテーブルの式

```
MONTH, MYVIEW /* MONTH.DUMMY */ MYVIEW.time_level = 'MONTH'
QUARTER, MYVIEW /* QUARTER.DUMMY */ MYVIEW.time_level = 'QTR'
YEAR, MYVIEW /* YEAR.DUMMY */ MYVIEW.time_level = 'YEAR'
TIME_ALL, MYVIEW /* TIME_ALL.DUMMY */ MYVIEW.time_level = 'ALL'
```

クラスとオブジェクトのマッピング

ブリッジは、各 OLAP ディメンションのクラスと各レベルのオブジェクトを作成する必要があります。クラスとオブジェクトのプロパティは次のようにマップされます。

ユニバースのアイテム	プロパティ	OLAP 項目からのマップ先
クラス (この章の最後の特別なケースも参照してください)	名前	ディメンション名
	オブジェクト	レベルと属性
ディメンション	名前	現在のレベル名 (ビューのフィールド名によって識別)
	SELECT 文	ビュー フィールド (たとえば、MYVIEW.YEAR)
	テーブル	適切な結合を使用できるようにする追加テーブル <ul style="list-style-type: none"> ・ 現在のビューのテーブル ・ 表示 ・ すべてのディメンションで最上位レベルのテーブル
詳細(オプション)	親ディメンション 注: その他のすべてのプロパティは、上記のディメンションの場合と同様です。	ID フィールドに関連する説明 フィールドから作成されたディメンションオブジェクト。たとえば、ID フィールドは YEAR、説明フィールドは YEAR_DESC です。オブジェクト YEAR は YEAR_DESC の詳細オブジェクトです。
メジャー	名前	キューブのメジャー名
	選択(ドリルスルーなし)	ビューフィールド(たとえば、MYVIEW.SALES)
	テーブル	適切な結合を使用できるようにする追加テーブル <ul style="list-style-type: none"> ・ 表示 ・ すべてのディメンションで最上位レベルのテーブル

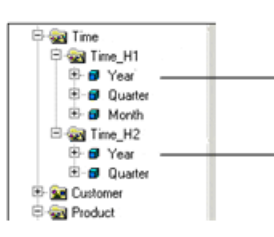
ユニバースのアイテム	プロパティ	OLAP 項目からのマップ先
	集計関数	なし

複数階層の特別なケース(複数階層ディメンションの使用)

複数階層ディメンションをサポートするには、ユニバース内で次の操作を実行します。

- この節の冒頭で説明されているとおり、各階層に対して複数のテーブルを作成する。
- ディメンションに対してクラスを、各階層に対してサブクラスを作成する。サブクラスに名前を付ける場合は、ディメンション名と階層名を連結します。
- 階層内のレベルに対応する各オブジェクトについて、SELECT 式に集計関数を追加する。現在の階層のエイリアスを除く上位レベルのエイリアスは、すべてコメントとして参照されます。以下はその例です。

```
@Aggregate_Aware (glb_dnorm_fact_mktseg_view.YEAR/*Year_H1.DUMMY
Channel_All.dummy Customer_All.dummy Product_All.dummy Time_All_H1.dummy
glb_dnorm_fact_mkseg_view.dummy*/)
```



```
@Aggregate_Aware (glb_dnorm_fact_mktseg_view.YEAR/*Year_H2.DUMMY
Channel_All.dummy Customer_All.dummy Product_All.dummy Product_All2.dummy
glb_dnorm_fact_mkseg_view.dummy*/)
```

- 集計テーブルの自動参照を設定して、サブクラスのオブジェクト(階層)を別の階層に対応しているテーブルと非互換にする。これにより、エンドユーザは、異なる階層に属するレベルを示すオブジェクトをレポート内で使用することがなくなります。

たとえば、H1 階層のテーブル Year_H1 は H2 階層のオブジェクトとは非互換です。



また H2 階層のテーブル Year_H2 は H1 階層のオブジェクトとは非互換です。



9.7.6 ビューを作成してユニバースを生成

最初にアナリティクスワークスペースキューブメタデータを使用してビューを定義し、次にユニバース作成オプションを設定して新しいユニバースを生成することによって、Oracle OLAP ユニバースを生成します。

9.7.7 Oracle Analytic Workspace からユニバースを作成および表示するオプション

ユニバースとビューは次のようにして作成できます。

[ユニバースの作成]オプション	説明
ビューを作成してユニバースを生成	ビューを作成し、ユニバースを作成するためにマップされる構造を選択します。
Oracle Analytical Workspace からのみのビューの作成	ユニバースを作成しない場合は、ビューを作成し、そのビューを保存できます。ビューはビューリストに追加され、ユニバースを作成するためにいつでも使用できます。

[ユニバースの作成]オプション	説明
既存のビューからのユニバースの生成	既存のビューを選択し、このビューから直接ユニバースを生成します。

関連項目

- ・ 537 ページの[ビューを作成してユニバースを生成](#)」
- ・ 539 ページの[Oracle Analytical Workspace からのみのビューの作成](#)」
- ・ 540 ページの[既存のビューからのユニバースの生成](#)」

9.7.8 ビューの作成とユニバースの生成

最初にアナリティクスワークスペースキューブメタデータを使用してビューを定義し、次にユニバース作成オプションを設定して新しいユニバースを生成することによって、Oracle OLAP ユニバースを生成します。

ビューを作成してユニバースを生成する

- 1 [ファイル]>[メタデータエクスチェンジ]を選択します。

Metadata Bridge パネルが表示されます。

[ユニバースの作成元]ドロップダウンリストから[Oracle OLAP]を選択します。

Oracle OLAP Universe Builder ウィザードが起動します。

- 2 [ビューの作成とユニバースの生成]を選択して[次へ]をクリックします。
- 3 接続を選択し、ユーザ名とパスワードを入力して、[完了]をクリックします。

その接続で利用できるアナリティクスワークスペースキューブを示したアナリティクスワークスペース(AW)キューブパネルが表示されます。

- 4 キューブノードをクリックします。

その接続で利用できるアナリティクスワークスペース(AW)が表示されます。

- 5 AW ノードを展開し、AW で利用できるキューブを表示します。

- 6 キューブを選択して[次へ]をクリックします。

- 7 ステータスボックスには、選択したキューブからロードされているメタデータの進行状況が表示されます。

[ビューの作成]ページが表示されます。このページには、そのキューブで利用できるディメンションとメジャーの一覧が表示されます。

- 8 必要に応じて、データ型とデータ長の値を変更します。手順は、次の通りです。

* データ型またはデータ長値をダブルクリックする。

* ドロップダウンリストボックスからデータ型を選択する。

- 9 [次へ]をクリックします。

[階層レベル]ページが表示されます。このページには、データ型と値と共に階層レベルが一覧表示されます。

- 10 必要に応じて階層値を編集し、[次へ]をクリックします。

[ビューとユニバースのプロパティ]ページが表示されます。

- 11 ビューの名前を入力し、ビューとユニバースのオプションを選択します。ビューのプロパティとユニバースのオプションは、次のとおりです。

ビューの名前を表示: このフィールドは編集できます。

OLAP_EXPRESSION の列を作成: これを選択すると、ビューに Raw(32)型の特別な列が作成され、ユニバースで OLAP_EXPRESSION 関数が使用できるようになります。

識別子の列を作成: これを選択すると、ディメンションメンバー(識別子)を示す列が作成されます。

既存のデータベースオブジェクトを置換: これを選択すると、データベース内の既存の型とビューの構造が置換されます。

派生テーブルを使用: これを選択すると、ユニバースは、データベースに物理的に作成されたビューからではなく、キューブ構造を参照する派生テーブルを使用して作成されます。派生テーブルは、データベース構造を参照するユニバースにのみ存在する仮想テーブルです。データベースには作成されません。これは、ユーザがビューを作成する権限を持たない場合や、データベースのビューを蓄積しない場合に便利です。派生テーブルの使用方法については、『ユニバースデザインツールユーザガイド』を参照してください。

オブジェクト ID を詳細に変換: ビューの識別子オプションで[作成]列が選択された場合のみアクティブになります。これを選択すると、オブジェクト ID は生成されたユニバースで詳細オブジェクトに変換されます。

- 12 [次へ]をクリックします。

SQL の検証ページが表示されます。

- 13 ビューの SQL を検証し、[次へ]をクリックします。

ユニバース情報の要約ページが表示されます。

- 14 ユニバース情報を検証し、[完了]をクリックします。

ユニバースデザインツールが開始され、新しく生成されたユニバースが開きます。

9.7.9 Oracle Analytical Workspace からのみのビューの作成

アナリティクスワークスペースキューブメタデータを使用するビューを作成できます。保存されたビューはビューの一覧に表示されます。一度ビューを作成すると、後でそのビューを選択し、ユニバースを生成できます。ビューだけを作成するには、537 ページの「[ビューを作成してユニバースを生成](#)」と同じ手順を実行します。ただし、Oracle OLAP ユニバースビルダウィザードの開始時に[ビューのみを作成]オプションボタンを選択します。

ビューはターゲットデータベースで作成されます。いつでもこのビューに接続して、ユニバースを作成できます。ビューを使用してユニバースを作成する手順については、540 ページの「[既存のビューからのユニバースの生成](#)」を参照してください。

9.7.10 既存のビューからのユニバースの生成

既存のビューからユニバースを生成することができます。既存のビューはリストに表示されます。リストからビューを選択し、ユニバースを生成します。

9.7.11 既存のビューからユニバースを生成する

- 1 [ファイル]>[メタデータエクスチェンジ]を選択します。メタデータエクスチェンジパネルが表示されます。[ユニバースの作成元]ドロップダウンリストから[Oracle OLAP]を選択して、[OK]をクリックします。Oracle OLAP Universe Builder ウィザードが起動します。
- 2 Oracle OLAP Universe Builder ウィザードの開始ページから[ビューからのユニバースの生成]オプションボタンを選択します。[次へ]をクリックします。[接続パラメータ]ボックスが表示されます。
- 3 接続を選択し、ユーザ名とパスワードを入力し、[次へ]をクリックします。その接続で使用するアナリティクスワークスペースキューブを示したキューブパネルが表示されます。
- 4 キューブノードをクリックします。その接続で使用するアナリティクスワークスペース(AW)が表示されます。
- 5 AW ノードを展開し、AW で使用できるキューブを表示します。キューブを選択して[次へ]をクリックします。キューブで定義されている使用可能なビューの一覧が表示されます。
- 6 一覧からビュー名をクリックし、[次へ]をクリックします。ステータスボックスには、選択したキューブからロードされているメタデータの進行状況が表示されます。[ユニバースの作成]ページが表示されます。このページには、ユニバースの作成に使用できるビューに定義されているディメンション、メジャー、および階層レベルが一覧表示されます。
- 7 必要に応じて列名または階層レベルを変更します。次の手順で行います。列名またはレベル値をダブルクリックします。適切な名前を選択、または入力します。
- 8 [完了]をクリックします。ユニバースデザインツールが開始され、新しく生成されたユニバースが開きます。

ユニバースの展開

10.1 概要

この章では、ユニバースのデプロイメントおよび管理方法について説明します。次の内容について説明します。

- ・ 541 ページの [「ユニバースのデプロイ方法」](#)
- ・ 543 ページの [「ユニバースへのアクセス制限の設定」](#)
- ・ 552 ページの [「ユーザとログインの管理」](#)

10.2 ユニバースのデプロイ方法

ユニバースの展開とは、Web Intelligence ユーザまたは他のデザイナーがユニバースを利用可能な状態にする操作を意味します。ユニバースを Central Management Server (CMS) にエクスポートすることで、ユニバースを展開できます。

ユニバースをテストリポジトリにエクスポートして Web Intelligence でテストを実行し、それから本稼動用リポジトリにユニバースをエクスポートしてください。

ユニバースの設計、作成、およびテスト段階が終了してから、ユニバースを Web Intelligence ユーザにデプロイするようにしてください。

リポジトリからのインポート/エクスポートによるユニバースの展開の詳細については、次の節を参照してください。

- ・ 45 ページの [「ユニバースのインポート」](#)
- ・ 47 ページの [「ユニバースのエクスポート」](#)

10.2.1 リポジトリでのユニバースの識別

それぞれのユニバースは、次のパラメータで識別されます。

識別子	説明
ファイル名	最大半角 100 文字まで (全角で 50 文字) の名前と拡張子「.unv」
ロングネーム	半角35文字(全角で17文字)までの名前を付けることができます。ユニバースの目的を詳しく説明する名前です。Web Intelligence では、この名前を使ってユニバースを識別します。
一意のシステム識別子	CMS に割り当てられた識別子

10.2.1.1 ユニバース識別子のルール

リポジトリの universe フォルダに保存されたユニバース識別子には、次のルールが適用されます。

- ・ ユニバース識別子は、CMS 内で一意である。
- ・ ファイル名とフォルダの場所(パス)を組み合わせることにより、ユニバースは親フォルダで一意である。

10.3 すべてのユーザへのユニバースのアクセス権の付与

ユニバースをワークグループモードで保存することで、ワークグループモード、エンタープライズモードにかかわらず、すべてのユニバースデザインツールユーザがアクセスできるよう設定できます。ユニバースの接続は、セキュリティ接続にすることはできません。すべてのユーザがユニバースにアクセスできるようにするには、ユニバースを安全ではない接続 (セキュリティ接続以外の接続) で保存する必要があります。

すべての ユニバースデザインツール ユーザがユニバースにアクセスできるようにするには、次の手順を実行します。

- 1 使用可能にするユニバースにセキュリティ接続が設定されていないことを確認します。
- 2 セキュリティ接続は、ユニバースを CMS にエクスポートするときに必要です。ユニバースがセキュリティ接続になっている場合、新しい共有接続を選択または作成します。詳しい説明は、83 ページの [「接続の定義と編集」](#)を参照してください。
- 3 [ファイル] > [別名で保存]を選択します。

[名前を付けて保存]ダイアログボックスが表示されます。

- 4 [すべてのユーザに保存]オプションを選択します。
- 5 [OK]をクリックします。

10.4 ユニバースへのアクセス制限の設定

ユニバースを使用する定義済みのユーザとグループに、制限を適用することができます。

ユニバースセキュリティは、2つのレベルで管理されます。

セキュリティレベル	説明
CMS	Central Management Console では、Central Management Server(CMS)に保存されているユニバースに適用する制限を設定できます。ユーザがアクセスできるユニバースを設定し、ユーザグループに定義された権限に応じて、ユニバースでの表示、編集、削除などの操作を制限することができます。このガイドは、CMSレベルでの制限の設定については説明していません。セントラル管理システムの使い方の詳細については、『BusinessObjects Enterprisea 管理者ガイド』を参照してください。
ユニバース	ユニバースの使用を許可されたユーザに制限を定義できます。制限には、オブジェクトアクセス、行アクセス、クエリと SQL 生成の制御、および接続の制御などが含まれます。このガイドでは、ユニバースで定義できる制御の種類を説明します。

10.4.1 制限とは

制限は、ユニバースに適用される、名前の付いた制限グループです。制限は、ユニバースで選択されたグループまたはユーザアカウントに設定することができます。ユーザがユニバース、オブジェクト、行、クエリの種類に接続すると、それぞれに適用された制限によってユニバースで使用される接続が決まります。

制限は BusinessObjects ユーザまたはグループに割り当てます。これにより、ユーザグループのプロファイルに基づいて、ユニバースオブジェクトやリソースへのアクセスが制限されます。

10.4.2 ユニバースに適用できる制限

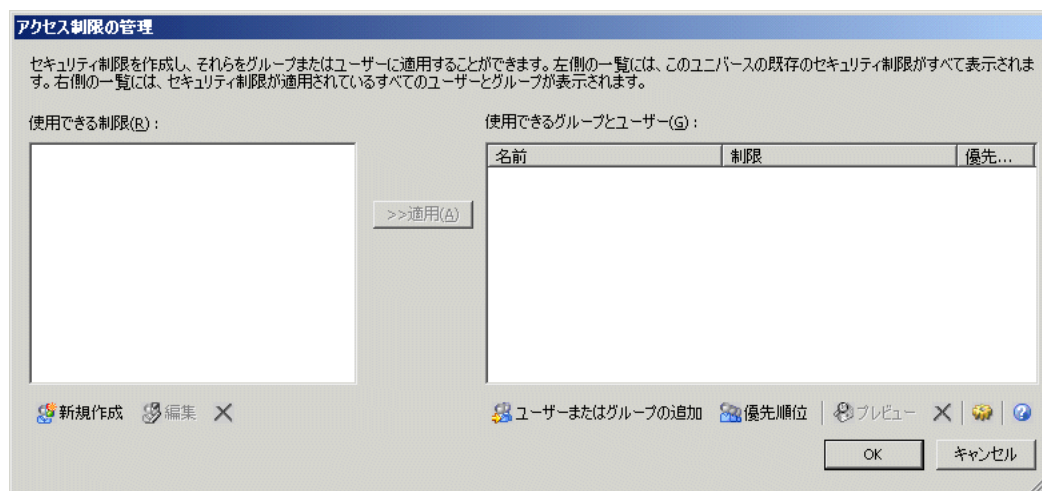
ユーザグループに適用されるアクセス制限は、制限内に定義します。ユニバースには、複数の制限を定義できます。制限はいつでも編集または削除することができます。

次の種類の制限を定義して、ユニバースの選択されたユーザまたはグループに適用できます。

制限の種類	説明
接続	ユニバースのデータソースへの接続です。ユニバースには代替の接続を選択できます。接続の作成と編集の詳細については、83 ページの「 接続の定義と編集 」を参照してください。
クエリの制御	クエリの結果のサイズや実行時間を制限するオプションです。詳しい説明は、98 ページの「 SQL の制限 」を参照してください。
SQL 生成オプション	クエリの SQL 生成を制御するオプションです。詳しい説明は、101 ページの「 SQL 生成パラメータの設定 」を参照してください。
オブジェクトアクセス	選択したオブジェクトをユニバースで利用できないようにすることができます。
行アクセス	行へのアクセスを制限し、クエリで返される結果を制限する WHERE 句を定義できます。
代替テーブルへのアクセス	ユニバースの参照先であるテーブルを、データベース上のほかのテーブルに置き換えることができます。

10.4.3 アクセス制限の管理


アクセス制限は、[アクセス制限の管理]ダイアログボックスで管理します。このボックスを表示するには、[ツール]>[セキュリティの管理]>[アクセス制限の管理]の順に選択します。以下のダイアログボックスが表示されます。





現在ユニバースで使用できる制限が[使用できる制限]枠に一覧表示されます。

各制限に定義されたユーザおよびユーザグループが[利用可能なグループとユーザ]枠に表示されます。

アクセス制限の管理で使用できるオプションは次のとおりです。

制限オプション	説明
新規作成	新しい制限を定義します。
編集	既存の制限を変更します。
選択された制限の削除 	一覧から制限を削除します。
ユーザまたはグループの追加	BusinessObjects システムに定義された BusinessObjects ユーザおよびグループの一覧からユーザまたはグループを追加します。
優先順位	1 つ以上のユーザグループに優先順位のレベルを設定できます。
プレビュー	BusinessObjects システムに定義されたすべてのユーザおよびグループを表示できます。

制限オプション	説明
選択したユーザまたはグループからセキュリティ制限を削除 	選択したユーザまたはグループに対して設定されたすべての制限を削除します。
制限オプション 	行制限を AND 演算子や OR 演算子を使用して実装するかどうかを選択できます。

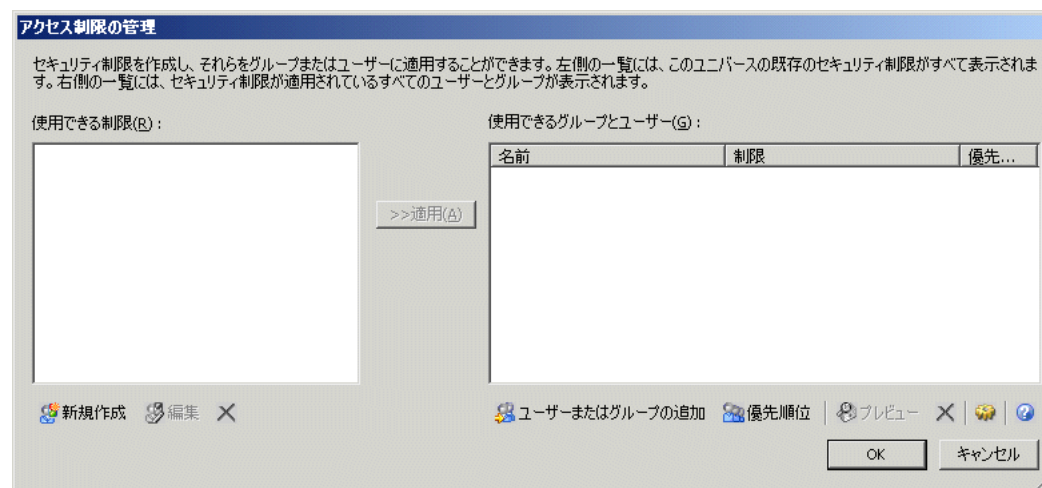
10.4.4 制限の作成

ユニバースが CMS にインポートされると、いつでも制限を作成、編集、削除できます。

適用先となるユーザグループのクエリの必要性に応じて、複数の制限を作成することも可能です。

10.4.4.1 制限を作成する

- 1 [ツール]>[セキュリティの管理]>[アクセス制限の管理]の順に選択します。
[アクセス制限の管理]ダイアログボックスが開きます。

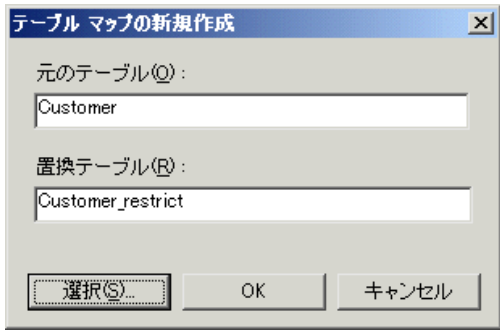


2 [新規作成] をクリックします。

[制限の編集] ダイアログボックスが表示されます。ユニバースの接続に対する置換接続を選択するか、[接続] ダイアログの隣にあるボタンを使用して、接続のプロパティを編集することができます。

3 次のいずれか、または複数の操作を実行します。

制限セット	操作
新規接続	[接続] リストボックスから接続名を選択します。
クエリ制御	<ul style="list-style-type: none"> ・ [制御] タブをクリックします。 ・ クエリオプションを選択し、各オプションに値を入力します。
SQL 生成オプション	<ul style="list-style-type: none"> ・ [SQL] タブをクリックします。 ・ クエリ、複数のパス、デカルト積オプションに適切なチェックボックスを選択します。
オブジェクトのアクセス制限	<ul style="list-style-type: none"> ・ [オブジェクト] タブをクリックします。 ・ [追加] をクリックします。[制限オブジェクト] ボックスが表示されます。 ・ [選択] をクリックします。オブジェクトのブラウザが表示されます。 ・ 制限するオブジェクトを選択します。 ・ [OK] をクリックしてオブジェクトのブラウザと[制限オブジェクト] ボックスを閉じます。
行アクセスの制限	<ul style="list-style-type: none"> ・ [行] タブをクリックします。 ・ [追加] をクリックします。 ・ [テーブル] ボックスの横にある[ブラウザ] ボタンをクリックします。 ・ テーブル名をクリックし、[OK] をクリックします。 ・ [WHERE 句] ボックスの横にある[ブラウザ] ボタンをクリックします。 ・ 定義ボックスに WHERE 句を入力します。または SQL エディタから列、演算子、関数を選択して WHERE 句を構築します。エディタの詳細については、166 ページの「結合 SQL エディタの使い方」を参照してください。

制限セット	操作
別のテーブルへの参照	<ul style="list-style-type: none"> ・ [テーブルマップ]タブをクリックします。 ・ [追加]をクリックします。[テーブルマップの新規作成]ダイアログボックスが開きます。 ・ [元のテーブル]ボックスにカーソルを置き、[選択]をクリックします。テーブル参照が表示されます。 ・ テーブルを選択し、[OK]をクリックします。 ・ [置換テーブル]ボックスにカーソルを置き、[選択]をクリックします。 ・ テーブル参照でテーブルを選択し、[OK]をクリックします。 

- 4 [OK]をクリックします。
新しい制限が一覧に表示されます。
- 5 [OK]をクリックします。

10.4.5 ユニバースへのアクセス制限の適用

ユニバースへのアクセス制限を設定するには、1 つ以上のユーザまたはユーザグループに制限を適用します。

10.4.5.1 ユニバースのユーザグループに制限を適用する

ユニバースの接続の置換接続を選択できます。


- 1 [ツール]>[セキュリティの管理]>[アクセス制限の管理]の順に選択します。
[アクセス制限の管理]ダイアログボックスが開きます。
- 2 [使用できる制限]枠の制限をクリックします。
- 3 [使用できるグループとユーザ]枠のユーザまたはグループをクリックします。
または
複数のユーザまたはグループを選択する場合は、Ctrl キーを押しながら、複数のユーザまたはグループをクリックします。
- 4 [適用]をクリックします。
- 5 [OK]をクリックします。

10.4.6 ユニバースの利用可能なユーザの一覧へのユーザグループの追加

BusinessObjects システムに定義されたユーザグループに制限を適用します。これらのユーザは、BusinessObjects Administration Console の[グループとユーザアカウントの管理]セクションで定義されます。BusinessObjects システムのユーザおよびグループの設定については、『BusinessObjects Enterprise XI R 3.0 管理者ガイド』を参照してください。

[使用できるグループとユーザ]枠に含まれていないユーザグループに制限を適用する必要がある場合は、そのユーザグループを次のようにして一覧に追加できます。

10.4.6.1 [使用できるグループとユーザ]枠にユーザグループを追加する

- 1 [アクセス制限の管理]ボックス ([ツール]>[セキュリティの管理]>[アクセス制限の管理]) で、[ユーザまたはグループの追加]アイコン  をクリックします。

[ユーザとグループの選択]ダイアログボックスが開きます。BusinessObjects システムへのアクセス権限がある BusinessObjects Administration Console に定義されたすべてのユーザグループが一覧表示されています。ユーザのリストが大きすぎてターゲットユーザまたはグループを容易に検索できない場合は、リストを次の方法で検索できます。

- ・ ドロップダウンリストから[名前]または[説明]を選択します。
- ・ [対象テキスト] ボックスにテキスト文字列を入力し、上で選択した“名前”または“説明”フィールドのいずれかで、リストからそのテキスト文字列を検索します。
- ・ [検索]アイコンをクリックして検索を開始します。

リストをフィルタするために、[グループ]または[ユーザ]チェックボックスをオンにして、リストにグループまたはユーザのみを表示することもできます。

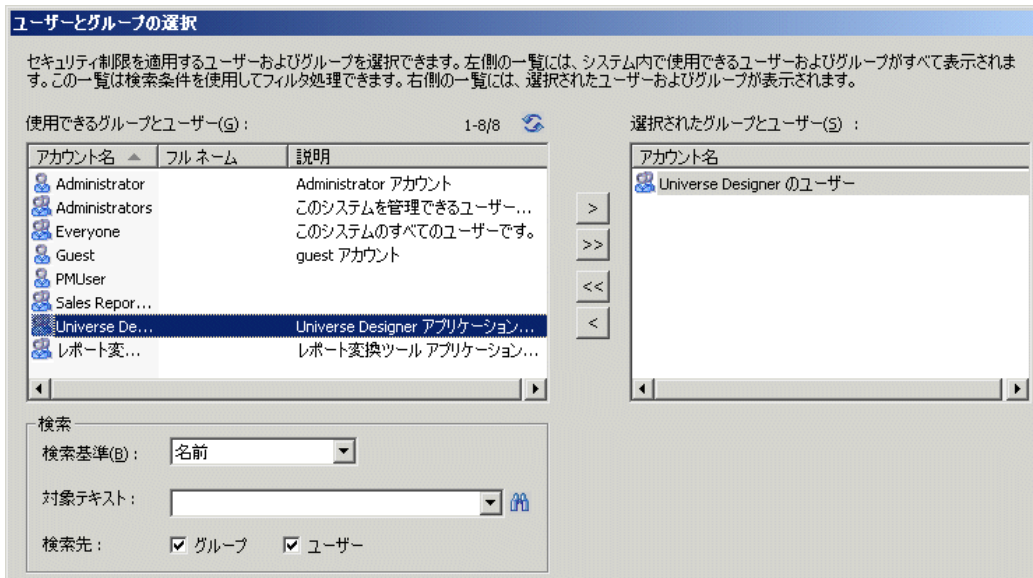
- 2 ユーザまたはグループをクリックします。

または

Ctrl キーを押しながら、複数のユーザまたはグループをクリックします。

- 3 右向き矢印をクリックします。

ダイアログボックスの右側の[選択したグループとユーザ]枠にユーザまたはグループが表示されます。



- 4 [OK]をクリックします。

これで、ユーザまたはグループが[アクセス制限の管理]ダイアログボックスの[使用できるグループとユーザ]リストに表示されます。

10.4.7 制限グループ優先順位の設定

複数のグループに属するユーザにどの制限を適用するかを、ユニバースを使用して指定することができます。たとえば、ユーザが 2 つのグループに属していて、販売グループでは 5,000 行のデータを表示する制限が、マーケティンググループでは 10,000 行のデータを表示する制限が設定されているとします。ユーザのレポート更新時には、グループの最下位レベルに関連付けられた制限が適用されます。上の例では、販売グループがレベル 1、マーケティンググループがレベル 2 であった場合、マーケティング(10,000 行)の制限が適用されます。

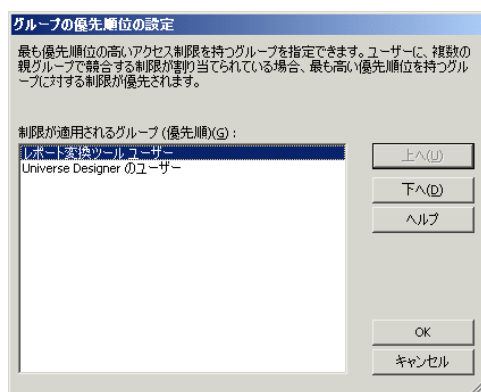
ユーザグループには順位を設定できます。順位に従って最下位のグループの制限が使用されます。

注

順位の設定は、接続、テーブルマッピング、SQL 制御など、排他的な制限にのみ適用されます。オブジェクト制限が両方のグループに設定されている場合は、両方に順位を適用できます。

10.4.7.1 複数の限定が使用されているユーザグループに優先順位を設定する

- 1 [ツール]>[セキュリティの管理]>[アクセス制限の管理]の順に選択します。
[アクセス制限の管理]ダイアログボックスが開きます。
- 2 [使用できるグループとユーザ]枠のユーザまたはグループをクリックします。
- 3 [優先順位]アイコンをクリックします。
[グループの優先順位の設定]ダイアログボックスが開きます。



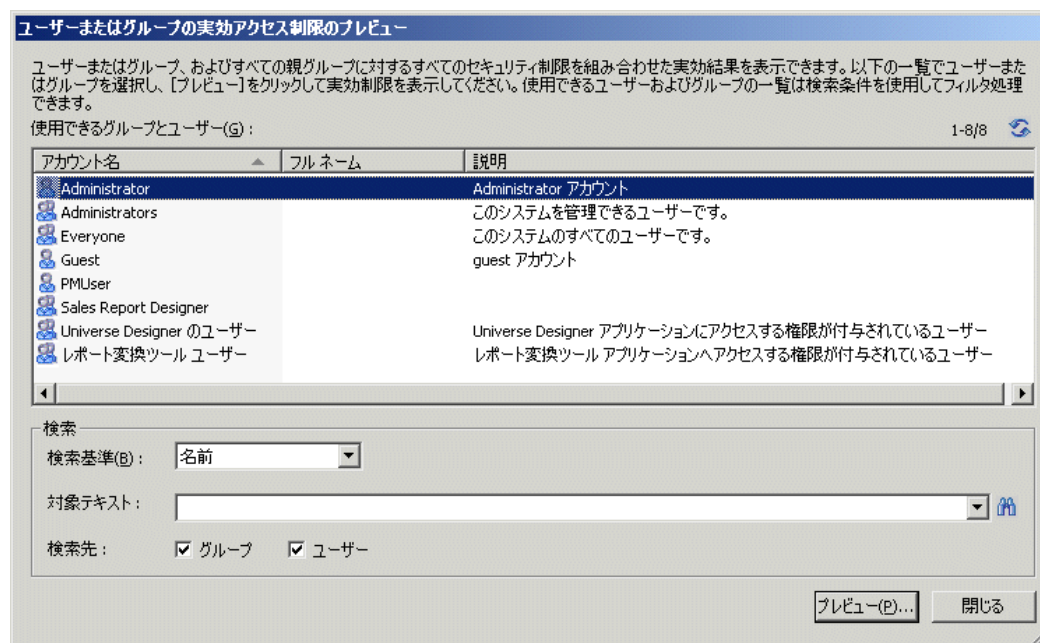
- 4 ユーザまたはグループを選択し、[上へ]または[下へ]ボタンをクリックして優先順位のレベルを変更します。
- 5 [OK]をクリックします。

10.4.8 ユーザとグループセキュリティ制限の表示

すべてのユーザとグループに適用された制限を表示することができます。

10.4.8.1 すべてのユニバースユーザとグループに設定された制限を表示する

- 1 [ツール]>[セキュリティ制限のプレビュー]の順に選択します。
ユーザとグループをプレビューするダイアログボックスが開きます。



2 一覧でユーザアカウント名をクリックします。

3 [プレビュー]をクリックします。

そのユーザアカウントに適用されたセキュリティ制限が表示されます。赤く表示されたパラメータとオプションは、その制限に特別に適用され、変更されたことを示します。

4 [OK]をクリックしてダイアログボックスを閉じます。

10.5 ユーザとログインの管理

ユニバースデザインツールに別のユーザとしてログインし、ログインを変更することもできます。ユーザアカウントはターゲットのリポジトリで有効なアカウントである必要があります。

スタンドアロンモードのユニバースデザインツールにログインすることもできます。ユニバースデザインツールを使用してユニバース、個人用接続および共有接続を作成できますが、CMS 内で接続およびユニバースにアクセスすることはできません。

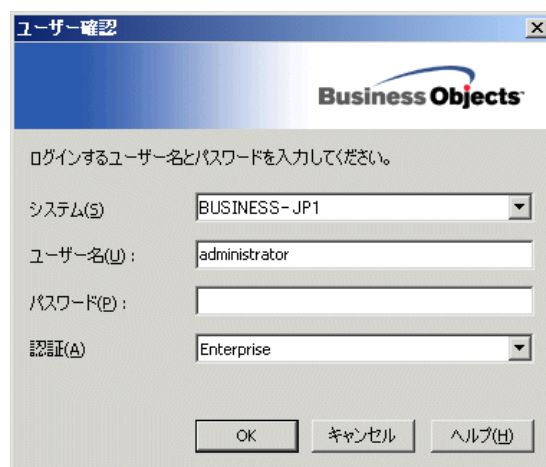
10.5.1 ログインの管理

作業セッションを終了せずに、別のユーザとしてユニバースデザインツールにログインし直すことができます。ユーザアカウントはターゲットのリポジトリで有効である必要があります。他のユーザの有効なユーザ名とパスワードを知っている場合のみ、そのユーザとしてログインできます。

10.5.1.1 他のユーザとしてログインする

- 1 [ツール]>[別の名前でのログイン]の順に選択します。

開いているユニバースがあると、ユニバースデザインツールによって自動的に閉じられます。[ユーザ確認]ダイアログボックスが表示されます。



ユーザー確認

Business Objects

ログインするユーザー名とパスワードを入力してください。

システム(S): BUSINESS-JP1

ユーザー名(U): administrator

パスワード(P):

認証(A): Enterprise

OK キャンセル ヘルプ(H)

- 2 [ユーザ名]ボックスに有効なユーザ名を入力します。
- 3 新しいユーザ名とパスワードを入力します。
- 4 [OK]をクリックします。

他のユーザとしてログインすると、自動的にそのユーザの持つすべてのアクセス権が与えられます。ただし、ユーザプロファイルで制限が設定されている場合は、一部の操作に制限が適用されます。

10.5.2 パスワードの管理

ユーザアカウントにパスワードの変更権限が付与されている場合、セッション中にログインに使用したパスワードを変更することができます。ただし、ユーザ名は変更できません。

10.5.2.1 パスワードを変更する

- 1 [ツール]>[パスワードの変更]の順に選択します。
[パスワードの変更]ダイアログボックスが開きます。
- 2 [古いパスワード入力]テキストボックスに、現在のパスワードを入力します。
- 3 [新しいパスワード入力]テキストボックスに、新しいパスワードを入力します。
- 4 [パスワードの確認]テキストボックスに、再度パスワードを入力して新しいパスワードを確認します。
- 5 [OK]をクリックします。
これでパスワードが変更されました。

サンプル資料の使い方

11.1 サンプル資料の使い方 – 概要

この付録では、Microsoft Accessを使って作成されたClubデータベースの詳細な情報を提供します。このデータベースは、本ガイド内で使用する例およびイラストに引用したデータベースです。

データベースファイルの Club.mdb は、Businss Objects フォルダの ¥Samples¥<language>¥Databases サブフォルダにあります。eFashion デモデータベースも、このフォルダに保存されています。

11.2 Clubデータベース

本ガイドで紹介されたほとんどの例は、Clubデータベースを使用しています。

11.2.1 テーブルの構成

Club データベースは、架空のパッケージツアー会社、アイランドリゾートのセールスマネージャが使用するデータベースです。セールスマネージャは、このデータベースの情報に基づいてセールス分析やマーケティング分析を行います。データベースは、次のテーブルから構成されています。

- ・ Age_group
- ・ City
- ・ Country
- ・ Customer
- ・ Invoice_Line
- ・ Region
- ・ Region_Sline
- ・ Reservation_Line

- ・ Reservations
- ・ Resort
- ・ Sales
- ・ Sales_Person
- ・ Service
- ・ Service_Line

次に、上記のテーブルとそれぞれの列について説明します。

11.2.1.1 Age_groupテーブル

Age_group テーブルは、顧客の年齢層の情報を保管します。

列名	説明
age_min	年齢層の下限
age_max	年齢層の上限
age_range	顧客の年齢層

11.2.1.2 Cityテーブル

City テーブルは、顧客の居住している都市の情報を保管します。

列名	説明
city_id	システムが自動的につける都市番号
city	顧客の居住する都市(Albertville, Amsterdam, Augsburg...Versailles, Washington D.C., Yokohama)

列名	説明
region_id	システムが自動的につける地域番号

11.2.1.3 Countryテーブル

Country テーブルは、顧客の居住している国の情報を保管します。

列名	説明
country_id	システムが自動的につける国番号
country	顧客の居住する国名 (Australia, France, Germany, Holland, Japan, UK, US)

11.2.1.4 Customerテーブル

Customer テーブルは、氏名および住所などの顧客に関連した情報を保管します。

列の名称	説明
cust_id	システムが自動的につける顧客番号
first_name	顧客の名称
last_name	顧客の姓
age	顧客の年齢

列の名前	説明
phone_number	顧客の電話番号
address	顧客の住所のはじめの一行
city_id	システムが自動的につける都市番号
sales_id	営業担当者番号(パッケージツアーの販売担当者)
sponsor_id	システムが自動的に付ける紹介者番号(オプション)

11.2.1.5 Invoice_Line テーブル

このテーブルには、請求書の情報が保管されます。この情報は、顧客へ請求書を送付するために使用されます。

列名	説明
inv_id	システムが自動的につける請求書番号
service_id	システムが自動的につけるサービス番号
days	(3-15)の数で、リゾート滞在日数を表します。顧客の滞在期間を15日まで記録します。15日を超える場合、超えた日数は新規滞在として日数を計算し直します。
nb_guests	請求書作成のための顧客数

11.2.1.6 Regionテーブル

Region テーブルは、顧客の居住地の地域情報を保管します。

列名	説明
region_id	システムが自動的につける地域番号
region	その顧客が居住する地域(Bavaria, East Coast, East Germany...Wales, West, West Japan)
country_id	システムが自動的につける国番号

11.2.1.7 Region_Sline テーブル

このテーブルで、ユニバースの営業売り上げを集計します。アグリゲート認識については、第5章を参照してください。

列名	説明
sl_id	システムが自動的に付けるサービスライン番号(サービスラインの情報は、Service_Lineテーブルに保管されます。)
region_id	システムが自動的につける地域番号
sales_revenue	その地域の営業売上げ合計

11.2.1.8 Reservation_Lineテーブル

Reservation_Line テーブルは、顧客の予約関連情報を保管します。

列名	説明
res_id	システムが自動的につける予約番号
service_id	システムが自動的につけるサービス番号
res_days	予約された曜日(1 - 7)
future_guests	予約人数(1 - 5)

11.2.1.9 Reservationテーブル

Reservationテーブルは、予約情報を保管します。

列名	説明
res_id	システムが自動的につける予約番号
cust_id	システムが自動的につける顧客番号
res_date	顧客の予約した日付

11.2.1.10 Resortテーブル

Resort テーブルは、各リゾート情報を保管します。

列名	説明
resort_id	システムが自動的につけるリゾート番号
resort	リゾート名 (Australian Reef, Bahamas Beach, French Riviera, Hawaiian Club, Royal Caribbean)
country_id	システムが自動的につける国番号

11.2.1.11 Salesテーブル

Sales テーブルは、売上げ情報を保管します。

列名	説明
inv_id	システムが自動的につける請求書番号
cust_id	システムが自動的につける顧客番号
invoice_date	請求書の日付

11.2.1.12 Sales_Personテーブル

Sales_Person テーブルは、アイランドリゾートの営業担当者の情報を保管します。

列名	説明
sales_id	システムが自動的に付けるセールス担当者番号

列名	説明
sales_person	セールス担当者名 (Andersen, Barrot, Bauman... Moore, Nagata, Schmidt)

11.2.1.13 Serviceテーブル

Service テーブルは、指定されたリゾートでの利用可能なサービスの種類および価格を保管します。

列名	説明
service_id	システムが自動的につけるサービス番号
service	リゾートで利用可能なサービス(後述のクエリ結果を参照)
sl_id	システムが自動的に付けるサービスライン番号(サービスライン情報は、次の一覧表に示されています。)
price	サービスの価格

11.2.1.14 Service_Lineテーブル

Service_Lineテーブルは、リゾートのサービスライン情報を保管します。サービスラインとは、サービスのカテゴリのことです。

列名	説明
sl_id	システムが自動的につけるサービスライン番号

列名	説明
service_line	サービスラインには、accommodation、food and drinks、recreationの3つの値があります。
resort_id	システムが自動的に付けるリゾート番号(値1 - 5)

より詳しい情報

情報リソース	場所
SAP BusinessObjects 製品情報	http://www.sap.com
SAP ヘルプ ポータル	<p>http://help.sap.com/businessobjects/ へアクセスし、[SAP BusinessObjects Overview] サイドパネルから [All Products] をクリックします。</p> <p>SAP ヘルプ ポータルでは、すべての SAP BusinessObjects 製品とそのデプロイメントについて扱った最新のドキュメンテーションにアクセスできます。PDF 版またはインストール可能な HTML ライブラリのダウンロードが可能です。</p> <p>一部のガイドは SAP サービス マーケットプレイスに格納されており、SAP ヘルプ ポータルからは入手できません。ヘルプ ポータルのガイド一覧で、そのようなガイドには SAP サービス マーケットプレイスへのリンクが付いています。保守契約を締結されたお客様には、このサイトにアクセスするための正規ユーザー ID が付与されます。ID の入手方法については、お客様担当のカスタマー サポート担当者までお問い合わせください。</p>
SAP サービス マーケットプレイス	<p>http://service.sap.com/bosap-support > ドキュメンテーション</p> <ul style="list-style-type: none"> ・ インストール ガイド: https://service.sap.com/bosap-instguides ・ リリース ノート: http://service.sap.com/releasenotes <p>SAP サービス マーケットプレイスには、一部のインストール ガイド、アップグレードおよび移行ガイド、デプロイメント ガイド、リリース ノート、サポート対象プラットフォームに関するドキュメントが格納されています。保守契約を締結されたお客様には、このサイトにアクセスするための正規ユーザー ID が付与されます。ID の入手方法については、お客様担当のカスタマー サポート担当者までお問い合わせください。SAP ヘルプ ポータルから SAP サービス マーケットプレイスにリダイレクトされた場合は、左側のナビゲーション ペインのメニューを使用して、アクセスするドキュメンテーションが含まれているカテゴリを探します。</p>
Docupedia	<p>https://cw.sdn.sap.com/cw/community/docupedia</p> <p>Docupedia は追加のドキュメンテーションリソース、協調的なオーサリング環境、および対話型のフィードバックチャネルを提供します。</p>

情報リソース	場所
開発者向けリソース	https://boc.sdn.sap.com/ https://www.sdn.sap.com/irj/sdn/businessobjects-sdklibrary
SAP Community Network 上の SAP BusinessObjects に関する記事	https://www.sdn.sap.com/irj/boc/businessobjects-articles これらの記事は、以前はテクニカル ペーパーという名称でした。
ノート	https://service.sap.com/notes これらのノートは、以前はナレッジ ベース記事という名称でした。
SAP Community Network 上のフォーラム	https://www.sdn.sap.com/irj/scn/forums
トレーニング	http://www.sap.com/services/education 弊社では、従来のクラス型の学習から目標を定めた eラーニング セミナーまで、学習ニーズや好みの学習スタイルに合わせたトレーニング パッケージを提供しています。
オンライン カスタマー サポート	http://service.sap.com/bosap-support SAP サポート ポータルには、カスタマー サポート プログラムとサービスに関する情報が含まれています。また、さまざまなテクニカル情報およびダウンロードへのリンクも用意されています。保守契約を締結されたお客様には、このサイトにアクセスするための正規ユーザー ID が付与されます。ID の入手方法については、お客様担当のカスタマー サポート担当者までお問い合わせください。
コンサルティング	http://www.sap.com/services/bysubject/businessobjectsconsulting コンサルタントは、初期の分析段階からデプロイメントプロジェクトの実現まで一貫したサポートを提供します。リレーショナル データベースと多次元データベース、接続、データベース設計ツール、カスタマイズされた埋め込みテクノロジーなどのトピックに関する専門的なサポートを行います。

索引

記号

@ 関数 375
@Aggregate_Aware 365, 377
@Prompt 378
@Prompt エディタ 379
@Select 393
@Variable 394
@Variable ロケール 397
@Where 401
\$INSTALLDIR
 ガイドで使用する変数 15

A

ANSI 92
 完全外部結合の作成 182
 結合のサポート 155, 169
 結合の定義 170
 ユニバースパラメータ 104
ANSI92 SQL 構文 130
Array Fetch
 最適化 432
AUTO_UPDATE_QUERY
 ユニバースパラメータ 104

B

BACK_QUOTE_SUPPORTED
 ユニバースパラメータ 105
Beach ユニバース 32
BEGIN_SQL
 ユニバースパラメータ 105
BLOB_COMPARISON
 ユニバースパラメータ 106
BOUNDARY_WEIGHT_TABLE
 ユニバースパラメータ 107

C

CASE_SENSITIVE 124
CHECK_OWNER_STATE 125
CHECK_QUALIFIER_STATE 125
Club データベース 32, 555
 Age_group テーブル 556
 City テーブル 556
 Country テーブル 557
 Customer テーブル 557
 Invoice テーブル 558
 Region テーブル 559

Club データベース (続き)
 Region_Sline テーブル 559
 Reservation_Line テーブル 560
 Resort テーブル 561
 Sales テーブル 561
 Sales_Person テーブル 561
 Service テーブル 562
 Service_Line テーブル 562
 テーブルの構成 555
COLUMNS_SORT
 ユニバースパラメータ 108
COMBINE_WITHOUT_PARENTHESES
 ユニバースパラメータ 108
COMBINED_WITH_SYNCHRO
 ユニバースパラメータ 109
COMMA 126
COMPARE_CONTEXTS_WITH_JOINS
 ユニバースパラメータ 109
CONCAT 126
CORE_ORDER_PRIORITY
 ユニバースパラメータ 110
CORRECT_AGGREGATED_CONDITIONS_IF_DRILL
 ユニバースパラメータ 110
CUMULATIVE_OBJECT_WHERE
 ユニバースパラメータ 111

D

DATE_WITHOUT_QUOTE 127
DECIMAL_COMMA
 ユニバースパラメータ 112
DELIMIT_IDENTIFIERS 138
DELIMIT_LOWERCASE 127
DISABLE_ARRAY_FETCH_SIZE_OPTIMIZATION
 ユニバースパラメータ 113
DISTINCT_VALUES
 ユニバースパラメータ 113

E

eFashion
 データベース 555
 ユニバース 32
END_SQL
 ユニバースパラメータ 114
Essbase キューブ
 オブジェクトからユニバースへのマップ
 ピング 512
 接続の定義 442
 ユニバース内でサポートされる機能
 441

EVAL_WITHOUT_PARENTHESES
 ユニバースパラメータ 115
EXT_JOIN_INVERT 138
EXTERN_SORT_EXCLUDE_DISTINCT
 127

F

FILTER_IN_FROM
 ユニバースパラメータ 115
FIRST_LOCAL_CLASS_PRIORITY
 ユニバースパラメータ 116
FORCE_SORTED_LOV
 ユニバースパラメータ 116

G

Group 句
 メジャーで生成 299
GROUPBY_WITH_ALIAS 128

I

IBM DB2
 分析関数 420
IDENTIFIER_DELIMITER 128
IF_NULL 129
INNERJOIN_IN_WHERE
 ユニバースパラメータ 117

J

Java Bean ユニバース 353, 354
Java Bean ユニバースのストアプロシージャ 353, 354
JOIN_BY_SQL
 ユニバースパラメータ 117

K

KEY_INFO_SUPPORTED 139

L

LOV
 ストアプロシージャ
 LOV の使用 357
LOV (「値の一覧」を参照) 283

M

MAX_INLIST_VALUES
ユニバースパラメータ 118
MDX 関数 291, 293, 459
MSAS キューブ
オブジェクトからユニバースへのマッ
ピング 513
接続の定義 442
ユニバース内でサポートされる機能
440

O

OLAP
Oracle OLAP からユニバースへの
マッピング 530
OLAP 関数 417
Teradata 427
OLAP 定義済みフィルタエディタ 464
OLAP の追加パラメータ 451
OLAP ユニバース
Essbase オブジェクトのマッピング 512
MDX 関数 293, 459
MSAS オブジェクトのマッピング 513
SAP NetWeaver BW オブジェクトの
マッピング 503
オプション 452
作成 435
サポートされているデータソース 436
接続の定義 442
定義済みフィルタエディタ 464
変更 453
OLAP ユニバースの同期化 475
OLAP_UNIVERSE
ユニバースパラメータ 118
Oracle
分析関数 420
Oracle アナリティクスワークスペース
OLAP からユニバースへのマッピング
530
階層テーブル 533
クラスとオブジェクトのマッピング 533
ディメンション結合 533
ビューからユニバースを生成 540
ビューの作成とユニバースの生成 537
表示 533
複数階層テーブル 533
ユニバース内のショートカット結合の
使用 531
ユニバースの作成 529, 537
ORDER_BY_STRINGS 139
OUTERJOINS_COMPLEX 130
OUTERJOINS_GENERATION 130
OVER_CLAUSE 132
OWNER 133

P

PATH_FINDER_OFF
ユニバースパラメータ 119
PDF
名前を付けて保存 50
PREFIX_SYS_TABLE 133
PRMファイル
CASE_SENSITIVE 124
CHECK_OWNER_STATE 125
CHECK_QUALIFIER_STATE 125
COMMA 126
CONCAT 126
DATE_WITHOUT_QUOTE 127
DELIMIT_IDENTIFIERS 138
EXT_JOIN_INVERT 138
EXTERN_SORT_EXCLUDE_DISTINCT
127
GROUPBY_WITH_ALIAS 128
IDENTIFIER_DELIMITER 128
KEY_INFO_SUPPORTED 139
OUTERJOINS_GENERATION 130
OVER_CLAUSE 132
OWNER 133
PREFIX_SYS_TABLE 133
QUALIFIER 134
QUOTE_OWNER 134
REFRESH_COLUMNS_TYPE 135
REVERSE_TABLE_WEIGHT 135
UNICODE_PATTERN 136
USER_INPUT_DATE_FORMAT 136
USER_INPUT_NUMERIC_SEPARATOR
137
PRM ファイル 419
DELIMIT_LOWERCASE 127
IF_NULL 129
ORDER_BY_STRINGS 139
OUTERJOINS_COMPLEX 130
REMOVE_SEMICOLONS 135
関数の追加 429

Q

QUALIFIER 134
QUOTE_OWNER 134

R

RedBrick
RISQL 関数 424
REFRESH_COLUMNS_TYPE 135
REMOVE_SEMICOLONS 135
REPLACE_COMMA_BY_CONCAT
ユニバースパラメータ 119
REVERSE_TABLE_WEIGHT 135

RISQL 関数 417
RedBrick 424

S

SAP NetWeaver BW
OLAP ユニバースのデータソース 436
OLAP ユニバースへのマッピング 503
接続の定義 442
複数言語ユニバース 439
ユニバース内でサポートされる機能
438
SELECT 句 266
SELFJOINS_IN_WHERE
ユニバースパラメータ 120
SHORTCUT_BEHAVIOR
ユニバースパラメータ 121
SMART_AGGREGATE
ユニバースパラメータ 121
SQL
エディタ 291
外部ストラテジーのテキストファイルの
作成 416
クエリ制御の設定 98
複数の文 99
SQL エディタ
結合の編集 166
SQL 生成オプション
制限 429
SQL パラメータ
動的 101
STG
ファイルパラメータ 408
STORED_PROC_UNIVERSE
ユニバースパラメータ 122

T

Teradata
OLAP 関数 427
THOROUGH_PARSE
ユニバースパラメータ 122
TRUST_CARDINALITIES
ユニバースパラメータ 123

U

UNICODE_PATTERN 136
UNICODE_STRINGS
ユニバースパラメータ 123
USER_INPUT_DATE_FORMAT 136
USER_INPUT_NUMERIC_SEPARATOR
137

W

WHERE 句
 オブジェクト 266
 競合 309, 313
 注意事項 306
 定義 304
 データを取得できない 313
 変更 304

あ

アクション
 元に戻す 57
 アクセス
 外部ストラテジー 405
 すべてのユーザがユニバースにアク
 セス 542
 アグリゲート認識 362
 オブジェクトの識別 364
 オブジェクトの定義 365
 設定 363
 データウェアハウス 362
 テーブルの参照 370
 非互換オブジェクトの参照 370
 非互換オブジェクトの指定 367
 ユニバースのテスト 374

値

テーブルの表示 68
 列の表示 68

値の一覧 324

SAP NetWeaver BW 変数 510
 エクスポート 335
 オブジェクトへの関連付け 283
 オプション 266, 283
 階層の作成 334
 カスケード 322
 カスタマイズ 341
 管理 340
 個人用データファイル 338
 最新表示 266, 338
 最適化 341
 削除 266
 作成 331
 消去 266
 条件の適用 332
 定義 326
 表示 266, 330
 プロパティ 327
 プロパティの設定 284
 変更 332
 編集 266, 332
 レポートでの使用 325

い

移行
 XI への外部ストラテジー 403
 外部ストラテジー Help テキスト 405
 一次キー
 キー 286
 一覧モード
 有効 62
 移動
 オブジェクト 271
 クラス 271
 ツールバー 55
 依頼検索
 値の一覧 327
 印刷
 オプションの設定 74
 整合性チェックの結果 200, 264
 プレビュー 76
 ページ設定 76
 ユニバース 74
 インターセクト
 許可 33
 使用 98
 インタフェース
 コンポーネント 54
 インタフェース言語 28, 30
 インフォメーションデザインツール
 .unv ユニバースの使用 33
 インポート
 ユニバース 45
 ユニバースのロック 52

う

ウィザード
 クイックデザイン 38
 ウィンドウ
 操作 54

え

エイリアス
 Essbase テーブル 512
 検出 229, 230, 231
 削除 209
 作成 207, 230, 231, 236
 スキーマでの役割 206
 定義 206
 名前 207, 208
 ファントラップの解決 255
 複数 232
 不適切な場合 244
 ループの解決 224

エクスポート

XMLメタデータファイルへのユニバー
 ス 521
 値の一覧 335
 制限 521
 前提条件 521
 ユニバース 47
 ユニバースのロック 52

エディタ

SQL 166

エラー

整合性のチェック 199, 263

演算子

結合 163

お

オブジェクト 17, 266, 268
 SELECT 句 266
 SQL の生成 20
 WHERE 句 266
 アグリゲート認識の定義 365
 値の一覧の関連付け 283
 移動 271
 階層 317
 階層順序の変更 321
 キー認識オプション 286
 結果で使用 266
 コメント 266
 作成 277, 278
 資格 17, 283
 資格の設定 284
 種類 266, 269
 条件で使用 266
 詳細 269
 詳細オプション 284
 書式設定 295
 書式のリセット 295
 ストラテジー 93
 生成される SQL の概要 18
 セキュリティ 286
 セキュリティアクセス 266
 ソース情報オプション 291
 定義 276
 ディメンション 269
 [テーブル]ボタン 280
 デフォルトの作成 33
 名前 266
 非互換 367
 日付書式 266
 非表示 271
 表示 19, 270
 プロパティ 278
 [分析]ボタン 280
 変更 279
 編集 279

オブジェクト (続き)
メジャー 269
役割の概要 268
ユーザアクセス 286
連結 316

オブジェクトアクセス
制限 429

オプション
値の一覧の関連付け 266
使用前に自動更新 266
ユーザによる値の一覧の編集 266
ユニバースと一緒にエクスポート 266

オプションプロンプト
OLAP ユニバース内 466

オフライン
スタンドアロンモード 552

か

カーディナリティ 216
Designer での使い方 188
キー 190
結合に設定 190
検出 33, 193
検出の最適化 195
最適化 195
手動設定 190
定義 187
データベース制限の解決 196
表示 189

カーネルとして使用
ユニバースのリンク 344

カーネルユニバース
変更 351
リンクの削除 350

解決
エイリアスとループ 224
キャズムトラップ 246, 250
結合パスの問題 205
コンテキストでループを解決 226
ファントラップ 252, 255
ループ 221, 229

階層

OLAP ユニバースにおけるマッピング
(NetWeaver BW) 506
値の一覧 334
エディタ 319
オブジェクトの再編成 321
カスタム 319
作成 317, 319, 320
識別 318
設定 319, 320
デフォルト階層の表示 319
ドリル 318
変数 (NetWeaver BW) 511

階層テーブル

Oracle アナリティクスワークスペース
ユニバースでの使用 533

外部キー 156
キー 286
認識の設定 289

外部結合
ANSI 92 182
完全 182
作成 180
制限 183
定義 76

外部ストラテジー 402
Designer で適用 416
Designer でのアクセス 405
Help テキストの移行 405
Help テキストの作成 405
SQL テキストファイルの作成 416
SQL に直接入力 414
STG ファイルパラメータ 408
XI への移行 403
オブジェクト作成ストラテジーの出力
書式 411
外部ストラテジーファイルの宣言 407
結合ストラテジーの出力書式 413
作成 414
サンプルの使い方 408
出力形式 410
使用 402
選択 92
定義 404
テーブル参照ストラテジー出力書式
414
表示する行数の設定 96
ファイルとプロセスの概要 403
リファレンステキストファイル 414

拡張 PRM 128

CASE_SENSITIVE 124
CHECK_OWNER_STATE 125
CHECK_QUALIFIER_STATE 125
COMMA 126
CONCAT 126
DATE_WITHOUT_QUOTE 127
DELIMIT_IDENTIFIERS 138
DELIMIT_LOWERCASE 127
EXT_JOIN_INVERT 138
EXTERN_SORT_EXCLUDE_DISTINCT
127
GROUPBY_WITH_ALIAS 128
IF_NULL 129
KEY_INFO_SUPPORTED 139
ORDER_BY_STRINGS 139
OUTERJOINS_COMPLEX 130
OUTERJOINS_GENERATION 130
OVER_CLAUSE 132
OWNER 133

拡張 PRM (続き)

PREFIX_SYS_TABLE 133
QUALIFIER 134
QUOTE_OWNER 134
REFRESH_COLUMNS_TYPE 135
REMOVE_SEMICOLONS 135
REVERSE_TABLE_WEIGHT 135
UNICODE_PATTERN 136
USER_INPUT_DATE_FORMAT 136
USER_INPUT_NUMERIC_SEPARATOR
137

確認

ユニバース 197, 261

カスケード値の一覧
作成 322

カスタマイズ
値の一覧 341

型

オブジェクト 266

関数

PRM ファイルに追加 429
関数一覧に表示 429

完全外部結合
作成 182

管理
値の一覧 340

き

キー

OLAP ユニバース内 453
カーディナリティ 190
外部キーの設定 289
使用 286
定義 286
認識 286
表示 156
プライマリキー 156
プライマリキーの設定 288

キー数値

OLAP ユニバース内でのマッピング
505

キータブ

キー認識オプション 286

キー認識

外部キー認識の設定 289
プライマリキー認識の設定 288

キー日付変数

OLAP ユニバース内 510

起動

ユニバースデザインツール 35, 36

キャズムトラップ 246

解決 246, 250
検出 249
コンテキストの使用 250
視覚的な検出 258

キャズムトラップ (続き)

- 識別 249
- 複数の SQL 251

行

- 行数の表示 71
- 行数の変更 71, 72
- 最大表示行数の設定 96
- 取得行数の変更 70

行アクセス

- 制限 429

行数

- クエリの最適化 33
- データ量に合わせて調整 33
- 表示 67

共有

- 接続 86

切り取り 271

く

クイックデザイン

- ウィザード 38
- ウィザードの無効 37
- 表示オプション 39

クエリ

- インターセクト 98
- 最大実行時間 98
- サブクエリの許可 98
- 条件オブジェクトの組み合わせ 313
- 条件オブジェクトの使用 312
- 推量 218
- 制御の設定 97, 98
- 非互換 218
- 複合条件 98
- 不明瞭 218
- ユニオン 98
- ループ 222

クエリの制御

- 制限 429

クラス 17, 269

- 移動 271
- 作成 273, 274
- サブクラス 276
- 定義 272
- デフォルトの作成 33
- 非表示 271
- プロパティ 275
- 変更 275
- 編集 275

クラスとオブジェクト

- OLAP ユニバース内での変更 453
- Oracle アナリティクスワークスペース 533
- ストアドプロシージャ 355
- クラスとオブジェクトのマッピング
- Essbase OLAP ユニバース 512

クラスとオブジェクトのマッピング (続き)

- MSAS OLAP ユニバース 513
- NetWeaver BW OLAP ユニバース 503

グラフィック

- 結合の作成 158
- 結合パスの問題の検出 258
- テーブル 61
- ループの識別 229

クリップボード

- 操作 271

グループセキュリティ

- 制限の表示 551

け

計算メジャー

- OLAP ユニバース内 458

形式

- データ型の表示 67

結果の増殖

- キャズムトラップ 246
- ファントラップ 252

結合

- ANSI 92 構文を使用した定義 170
- ANSI 92 サポート 155, 169
- 演算子 163
- カーディナリティの検出 33
- カーディナリティの設定 190
- 外部キー 156
- 外部結合 172, 180
- 検出 160
- 削除 172
- 作成 157, 158, 159
- サポートされる種類 172
- シート結合 172, 177
- 自己結合 172, 184
- ショートカット結合 172, 183
- 数式バーで編集 168
- ストラテジー 93
- 定義 154
- テーブルも挿入 161
- 等結合 172
- プライマリキー 156
- プロパティ 162
- 分析 164
- 変更 165
- 編集 164, 165, 166
- リンクされたテーブルの表示 33

結合パス

- エイリアスの定義 206
- キャズムトラップ 205, 246
- ファクトテーブルの役割 204
- ファントラップ 205
- 不正確な結果 204
- 問題の解決 205

結合パス (続き)

- 問題の概要 203
- 問題の検出 205, 258
- ループ 205
- ルックアップテーブル 204

言語 30

- 言語の定義 29, 30

検索

- スキーマのループ 229
- ユニバースでの検索 58
- ユニバース内 58
- ユニバースのクイック検索 61

検出

- エイリアス 229, 230, 231
- カーディナリティ 193
- カーディナリティの最適化 195
- キャズムトラップ 249
- 結合 160
- 結合のカーディナリティ 33
- 結合パスの問題 258
- コンテキスト 229, 233
- 整合性エラー 199, 263
- ファントラップ 254
- ユニバースのエラー 199, 263
- ループ 229, 234

検証

- ユニバース 197, 261

こ

更新

- OLAP ユニバース 467
- コンテキスト 216

構造

- STG ファイル 408

構造枠 53

- 最新表示 202, 266
- 表示オプション 67

構文

- SELECT 句に自動挿入 429
- 連結オブジェクト 316

個人

- 接続 86

コピー 271

コメント

- オブジェクト 266
- ユニバース 80

コンテキスト

- キャズムトラップの解決 250
- 検出 229, 233
- 検出の問題 216
- 更新 216
- 削除 215
- 作成 211, 233, 236
- スキーマでの役割 210
- 定義 210

コンテキスト (続き)
 非互換クエリ 218
 ファントラップの解決 255
 複数の SQL 文 99
 不明瞭なクエリ 218
 変更 214
 編集 214
 ループの解決 226
 コンテキスト検出の問題 216
 コンポーネントとして使用
 ユニバースのリンク 345

さ

最新表示
 OLAP ユニバース 467
 値の一覧 266, 338
 構造 202, 266
 最適化
 値の一覧 341
 テーブル参照 146
 削除
 SQL パラメータ 102
 エイリアス 209
 結合 172
 コンテキスト 215
 接続 90
 テーブル 61
 作成
 OLAP ソースからのユニバース 435
 値の一覧 331
 値の一覧の階層 334
 エイリアス 207, 230, 231
 オブジェクト 277, 278
 階層 317, 319
 外部ストラテジー 414
 カスケード値の一覧 322
 クラス 273, 274
 結合 157, 158, 159, 160
 コンテキスト 211, 233
 サブクラス 276
 シータ結合 177
 自己結合 184
 条件オブジェクト 308
 詳細 297
 制限セット 546
 接続 83, 89
 接続 OLAP 442
 ディメンション 296
 デフォルトのクラスとオブジェクト 33
 等結合 173
 動的 SQL パラメータ 102
 ファイルの値の一覧 338
 複合結合 175
 メジャー 297

作成 (続き)
 メタデータソースからのユニバース
 517
 ユニバース 77, 78
 リンク 347
 サブクラス
 作成 276
 参照
 テーブル 53
 サンプル
 ユニバースとデータベース 32

し

シータ結合
 作成 177
 定義 76
 資格
 オブジェクト 283, 284
 識別
 階層 318
 キャズムトラップ 249
 集計レベル 364
 ファントラップ 254
 ユニバース 81, 541
 ループ 229
 自己結合
 作成 184
 定義 76
 データの制限 313
 システム
 ログイン 35
 自動
 エイリアスの作成 236
 オブジェクト作成 278
 カーディナリティの検出 193
 クラスの作成 274
 結合の挿入 161
 コンテキストの作成 236
 テーブルの整列 64
 ユニバースのチェック 198, 262
 ループの検出 234
 集計
 テーブル 362
 メジャーに設定 300, 457
 出力
 外部ストラテジーのフォーマット 410
 手動
 オブジェクト作成 277
 カーディナリティの設定 190
 ユニバースのチェック 199, 263
 手法
 ユニバース設計 24
 消去
 値の一覧 266

条件
 値の一覧に適用 332
 オブジェクト(「条件オブジェクト」を参照) 308
 表示 270
 複数のテーブルを参照 314
 条件オブジェクト
 OLAP ユニバース内での定義 461
 WHERE 句の競合 309
 移動 271
 クエリでの使用 312
 クエリで必須 309
 作成 308
 定義 308
 非表示 271
 詳細
 オブジェクトオプション 284
 作成 297
 定義 297
 ショートカット結合
 Oracle アナリティクスワークスペース
 ユニバースでの使用 531
 作成 183
 定義 76
 書式設定
 オブジェクト 295
 リセット 295
 シングルサインオン (SSO)
 SAP NetWeaver BW 439
 シングルサインオン(SSO)
 MSAS 2005 446

す

推量可能なクエリ 218
 数式バー
 結合の編集 168
 表示 168
 数値
 ユニバースのバージョン 52
 スキーマ
 エイリアスの使用 206
 カーディナリティの使用 188
 行数の表示 67
 結合パスの問題の検出 258
 コンテキストの使用 210
 最新表示 202, 266
 定義 141
 データ型の表示 67
 テーブルの操作 143
 デザイン工程 143
 表示 66
 スタンドアロン
 CMS を使わない使用 552
 ストアドプロシージャ
 クラスとオブジェクト 355

ストアドプロシージャ (続き)
 動的な値 357
 複数結果セット 357
 ストアドプロシージャのオブジェクト 355
 ストアドプロシージャの動的な値 357
 ストラテジー
 オブジェクト 93
 外部 (「外部ストラテジー」を参照) 402
 クイックデザインウィザードでの選択
 417
 結合 93
 出力フォーマット 410
 選択 92
 テーブル 93
 スマートメジャー 301, 454

せ

正規化 241
 制御
 ユニバース 543
 制限
 SQL 生成オプション 429
 WHERE 句 304
 WHERE 句での注意事項 306
 オブジェクトアクセス 429
 ガイドライン 315
 行アクセス 429
 クエリの実行時間 98
 クエリの制御 429
 自己結合の使用 313
 接続 429
 代替テーブルへのアクセス 429
 複数のテーブル 314
 制限セット
 グループ優先順位の設定 550
 作成 546
 定義 544
 ユーザグループへの適用 548
 整合性
 自動チェック 198, 262
 手動チェック 199, 263
 ユニバースのチェック 197, 261
 整合性のチェック 359
 Designer の開始時 76, 202
 エラーの種類 199, 263
 結果の印刷 200, 264
 自動分析 76, 202
 送信オプション 76, 202
 データベースの変更 202, 266
 生成
 動的 SQL パラメータ 102
 製品言語 30
 整理
 テーブルの表示 61, 146

整列
 自動整列 146
 セキュリティ
 CMS レベル 543
 オブジェクト 286
 オブジェクトアクセス 266
 制限セット 543
 接続 86
 ユニバースレベル 543
 接続 84
 OLAP ユニバース 442
 XI R2 の使用 37
 一覧表示 89
 共有 86
 個人 86
 削除 90
 新規 89
 新規作成 89
 制限 429
 セキュリティ 86
 追加 91
 定義 83
 データベースエンジン 85
 名前 85
 パスワード 85, 87
 パラメータの概要 OLAP 442
 変更 82, 83
 ユニバースパラメータ 81
 接続パネル 84
 設定
 カーディナリティ 190
 階層 320
 行数 71
 最大表示行数 96
 スキーマの表示オプション 67
 テーブルウェイト 433
 デフォルトの保存 51
 動的 SQL パラメータ 101
 保存オプション 51
 ユニバースのリンク 347
 要約パラメータ 91
 リソースの制御 97
 設定オプション
 OLAP ユニバース内 452
 説明
 変更 82
 ユニバース 81
 宣言
 外部ストラテジー 407
 選択
 スキーマの表示オプション 66
 ストラテジー 92
 テーブル 61

そ

挿入
 @ 関数 375
 最適化 146
 テーブル 143, 144
 ソース言語 30
 ソース情報
 オブジェクトオプション 291

た

代替テーブルへのアクセス
 制限 429
 代用言語 30
 多次元分析 317
 階層の作成 320
 種類 317

ち

チェック
 ユニバース 197, 198, 199, 261, 262,
 263
 置換
 文字列または文字 58
 抽出
 テーブルの結合 33

つ

追加
 接続 91
 接続 OLAP 442
 テーブル 144
 追加 OLAP パラメータ 451
 ツールバー
 移動 55
 使用 55

て

定義 246
 .PRM ファイル 419
 @ 関数 375
 OLAP ユニバース内の条件オブジェ
 クト 461
 WHERE 句 304
 アグリゲート認識オブジェクト 365
 値の一覧 326
 オブジェクト 276
 カーディナリティ 187
 外部ストラテジー 404
 キャズムトラップ 246
 クラス 272

定義 (続き)

- コンテキスト 210
- シート結合 177
- 自己結合 184
- 条件オブジェクト 308
- 詳細 297
- ショートカット結合 183
- 接続 89
- 接続 OLAP 442
- ディメンション 296
- 動的 SQL パラメータ 101
- ファントラップ 252
- 複合等結合 175
- メジャー 297
- ユニバースパラメータ 77
- ループ 221

定義済みフィルタエディタ 464

ディメンション

- 作成 296
- 定義 296

ディメンション結合

- Oracle アナリティクスワークスペース
ユニバースでの使用 533

データ

- 値の一覧ファイル 338
- 空のデータ 313
- ドリル 318
- 表示 145

データ型

- 表示 67

データベース

- 値の一覧の依頼検索 327
- サポートされるスキーマ 21
- テーブルの表示 143

データベースエンジン

- 接続 85

テーブル

- 値の表示 68
- 返される行数の変更 434
- 画像表示 61
- 行数の表示 71
- 結合の抽出 33
- 結合も挿入 161
- 削除 61
- 参照 (「テーブル参照」を参照) 143
- 自動整列 64
- 集計 362
- 集計テーブルのループ 372
- スキーマの操作 143
- ストラテジー 93
- 整理 61
- 整理して表示 146
- 整列 146
- 選択 61
- 操作 61
- 挿入 143, 144

テーブル (続き)

- 挿入の最適化 146
- 追加 144
- デフォルトのクラスとオブジェクトの作成 33
- 派生 147
- 表示変更 64
- ファクトの定義 204
- 複数のテーブルを参照 314
- ルックアップ 204

テーブルウェイト

- 設定 433

テーブル参照 53

- 最適化 146
- 使用 143
- データの表示 145
- 有効 143

テーブルの整列 64

[テーブル]ボタン 280

デカルト積

- 回避 100
- 警告 100

適用

- 外部ストラテジー 416

デザイン

- スキーマ 143
- デザインウィザード
- 無効 37

テスト

- ユニバース 359

デフォルト

- クラスとオブジェクト 33
- 保存オプション 51
- 保存オプションの変更 51

デフォルト階層

- カスタム階層と共に使用 319

デフォルト値

- プロンプトの構文 378

デモ

- データベース 32
- ユニバース 32

と

統計

- ユニバース 80

等結合

- 作成 173
- 定義 76
- 複合 175

動的

- SQL パラメータ 102

ドキュメント言語 30

特性

- OLAP ユニバース内でのマッピング
505

閉じる

- ユニバース 51
- トラブルシューティング
整合性のチェック 200, 264
- トランスレーションマネジメントツール 31
- ドリル 318

な

名前

- エイリアス 207, 208
- オブジェクト 266
- 接続 85
- ユニバース 81

に

入力列 152

認証

- Essbase 接続 447
- MSAS 接続 446
- SAP NetWeaver BW 接続 444
- ログイン 35

ね

ネストされた派生テーブル 150

- 作成 151

ネストされた派生テーブルの作成 151

は

ハイパーリンク

- オブジェクト作成 282
- オブジェクトの書式 282
- 読み込み形式 282

配布

- ユニバース 541

パスワード

- 接続 85, 87
- 変更 553
- ログイン 35

派生テーブル

- 使用 147

派生テーブルエディタ 150

派生テーブルの派生テーブル 150

派生ユニバース

- オブジェクト順 352
- リンクの作成 347

パラメータ

- リソースの制御を設定 97

パラメーター化された列 152

パラメータファイル

- 定義 419

貼り付け 271

版数 52

ひ

非互換オブジェクト 367

日付型

データベース形式 266

必須フィルタ

値の一覧 311

定義 309

変数 (SAP NetWeaver BW) 508

非表示

オブジェクト 271

クラス 271

条件オブジェクト 271

ビュー

条件 270

表示

Oracle アナリティクスワークスペース
ユニバースでの使用 533

値の一覧 330

一覧モード 62

オブジェクト 19, 270

カーディナリティ 189

キー 156

行数 67, 71

グループセキュリティ制限 551

数式バー 168

スキーマ 67

スキーマオプション 66

接続 89

データ型 67

データベーステーブル 143

テーブル参照のデータ 145

テーブルの行数 71

テーブルの整理 61

テーブルの変更 64

ユニバース枠の条件 270

表示言語 30

開く

ユニバース 46

ふ

ファイル

値の一覧の作成 338

ファクトテーブル

定義 204

ファントラップ

エイリアスとコンテキストの使用 255

解決 252, 255

結果の増殖 252

検出 254

視覚的な検出 258

識別 254

定義 252

ファントラップ (続き)

複数の SQL 258

フィルタ

クエリで必須 309

クラスと条件 270

フォルダ

リポジトリファイルシステム 47

複合クエリ

許可 33

複合結合

作成 175

複合条件

使用 98

複数階層テーブル

Oracle アナリティクスワークスペース
ユニバースでの使用 533

複数言語ユニバース 28

SAP NetWeaver BW 439

複数の SQL

キャズムトラップ 251

キャズムトラップの解決に使用 251

ファントラップ 258

複数のエイリアス 232

不正確な結果

キャズムトラップ 246

ファントラップ 252

ループ 222

プライマリキー 156

認識の設定 288

フレキシブルルックアップテーブル 241

プレビュー

ユニバース 76

プロパティ

ユニバース 77

プロンプト

オプション(OLAP ユニバース) 466

デフォルト値の構文 378

プロンプトエディタ 379

分析

結合 164

分析関数 417

IBM DB2 420

Oracle 420

RedBrick 424

Teradata 427

関数一覧に表示 429

サポートされる種類 419

利点 418

[分析]ボタン 280

へ

ページ

設定 76

ヘルプ

外部ストラテジーに作成 405

変更

Array Fetch 432

WHERE 句 304

値の一覧 332

オブジェクト 279

オブジェクトの書式 295

行数 71, 72

クラス 275

結合 165, 166

コンテキスト 214

取得行数 70

スキーマの表示 66

接続 82, 83

説明 82

テーブルの行数 434

テーブルの表示 64

デフォルトの保存オプション 51

パスワード 553

ユニバースの定義パラメータ 82

ユニバース名 82

編集

SQL エディタ 291

値の一覧 332

オブジェクト 279

階層 319

クラス 275

結合 164, 165, 166

コンテキスト 214

数式バーの使用 168

接続 83

動的 SQL パラメータ 102

変数

OLAP ユニバース (NetWeaver BW)
506

ほ

保存

PDF ファイル 50

デフォルト 51

デフォルトの変更 51

ユニバース 49

ま

マイナス

許可 33

マウス

アクション 56

マスタとして使用

ユニバースのリンク 345

む

無効

デザインウィザード 37

め

メジャー

Group 句 299

依頼されたデータベースの投影機能
301, 454

計算 458

作成 297

集計関数 298

集計機能 300, 457

定義 297

動的なオブジェクト 298

複数の文 99

メタデータ

接続オプション 519

ユニバースのオプション 519

メタデータユニバース

Oracle アナリティクスワークスペース
からの作成 529, 537

作成 517

ソースの選択 516

ユニバースの XML 形式へのエクス
ポート 521

も

文字

検索または置換 58

文字列

検索置換 58

元に戻す

アクション 57

ゆ

有効

一覧モード 62

テーブル参照 143

ユーザ

オブジェクトへのアクセス 286

ユニバースへのアクセス 542

ログイン 35, 552

ユーザインタフェース言語 30

ユーザグループ

制限セットの適用 548

制限の優先順位の設定 550

ユニオン

許可 33

使用 98

ユニバース

.unv ファイル拡張子 49

OLAP 435

OLAP の更新 467

OLAP の最新表示 467

印刷 74

インポート 45

ウィンドウの概要 19

エクスポート 47

エクスポートの制限事項 521

開発サイクル 24, 26

概要 16, 80

クイックデザインウィザード 38

コメント 80, 91

作成 77, 78

作成方法 20

識別 81, 541

識別子 541

条件の表示 270

ストアドプロシージャの使用 354

すべてのユーザがアクセス 542

整合性のチェック 197, 261

接続 81

接続の作成 83

接続の定義 83

接続の編集 83

説明 81, 91

使い方の概要 21

デザインプロファイル 22

テスト 359

デフォルトのクラスとオブジェクトの作
成 33

統計 80

動的リンク 347

閉じる 51

名前 81, 541

名前変更 82

配布 541

派生ユニバースでのオブジェクト順
352

パラメータの定義 77, 81

版数 52

開く 46

ファイル名 541

他のユニバースに含める 352

保存 49

保存オプション 51

役割 16

ユニバースのリンク 101

要約パラメータ 91

リソースの制御 97

リソースの制御を設定 97

ロック 52

ロングネーム 49, 541

ワークグループ設計 52

枠 53, 270

ユニバースセキュリティ

制御の設定 543

ユニバース設計

開発サイクル 26

手法 24

初期設計 25

ユニバースデザインツール

インタフェースコンポーネント 54

開始 35, 36

構造枠 53

ユーザインタフェース 53

ユニバースウィンドウ 53

ユニバース枠 53

ユニバースの作成

ストアドプロシージャの使用 355

ユニバースの整合性のチェック 359

ユニバースのライフサイクル (OLAP) 467

ユニバースファイル

インフォメーションデザインツールで
の使用 33

よ

要約

ユニバースの情報 80

要約パラメータ 91

り

リセット

オブジェクトの書式 295

リソース

ユニバースに設定 97

リポジトリ

フォルダの構造 47

ユニバースのエクスポート 47

ユニバースのデプロイ 541

リンク

作成 347

動的 347

ユニバース 101

リンクされたユニバース 342

CORE_ORDER_PRIORITY 352

オブジェクト順 352

カーネルとして使用 344

コンポーネントとして使用 345

システム要件 346

制限 346

設定 347

動的リンク 347

他のユニバースに含める 352

マスタとして使用 345

利点 345

リンクの削除 350

リンク方法 344

る

ループ

- エイリアスで解決 224
- 解決 221, 229
- クエリへの影響 222
- 検出 229, 234
- コンテキストで解決 226
- 識別 229
- 定義 221
- 例 238

ロックアップテーブル

- 共有 240
- 定義 204
- フレキシブル 241

れ

列

- 値の表示 68
- 連結オブジェクト 316
- 構文 316
- 作成 316

ろ

ログイン

- 管理 552
- システム 35
- 認証 35
- パスワード 35

ログイン (続き)

- 他のユーザとして 552
- ユーザ名 35
- ロケール 30, 397
- ロケールの定義 29
- ロック
- ユニバース 52

わ

ワークグループ

- ユニバース設計 52
- 割り当て
- パスワード 87

