

IBM DB2 Web Query for IBM i

Version 2 Release 1.8

DN6000000.0514

Active Technologies、EDA、EDA/SQL、FIDEL、FOCUS、Information Builders、Information Builders のロゴ、iWay、iWay Software、Parlay、PC/FOCUS、RStat、Table Talk、Web390、WebFOCUS、WebFOCUS Active Technologies、および WebFOCUS Magnify は Information Builders, Inc. の登録商標であり、また DataMigrator および Hyperstage は同社の商標です。

本マニュアルの性質上、多くのハードウェア、ソフトウェア製品の商標が本文内で使用されています。ほとんどの場合、製品名はそれらの会社によって商標、登録商標として指定されています。したがって、弊社ではこれらの製品名を総称として使用する意図はありません。これらの製品名を、説明されている製品を参照する以外の目的で使用する場合、商標に関わる権利に関して十分注意が必要です。

Copyright © 2014 , by Information Builders, Inc. and iWay Software. All rights reserved. Patent Pending. このマニュアルの全部、または一部の転載、コピーは Information Builders Inc. の書面による承諾なしでは許可されません。

目次

はじめに.....	11
マニュアルの表記法.....	12
1. OLAP レポートによるデータ分析.....	15
日常の作業 - 一般的な DB2 Web Query の用途.....	16
OLAP レポートの要件.....	17
OLAP 有効データ.....	17
レポートの OLAP 有効化.....	18
OLAP 用語集.....	20
OLAP レポートの特性.....	21
OLAP データでの 3 通りの作業.....	24
レポート.....	24
選択パネル.....	25
OLAP コントロールパネル.....	27
ディメンションと基軸のドリルダウン.....	29
データのソート.....	29
基軸のソート.....	30
ディメンションのソート.....	34
数値データの分位グループ化.....	39
基軸上での計算の実行.....	43
データの制限.....	45
日付要素への選択条件の適用.....	50
傾向の視覚化.....	55
グラフとレポートの表示.....	56
OLAP グラフに追加されるタイトル.....	58
レポートでの基軸の表示のコントロール.....	62
基軸の積み重ね.....	63
基軸の列順変更.....	63
基軸の非表示と表示.....	64

ディメンションの追加と削除.....	65
OLAP レポートの保存.....	66
オプションの保存.....	67
OLAP で参照する統一されたフィールド名.....	71
OLAP レポートとグラフの他のフォーマットでの保存と表示.....	72
2. データ記述とアクセスの概要.....	77
データソースの用語について.....	78
アプリケーションのデータ解析方法.....	78
データアダプタによるシノニムの作成	79
シノニムの使用.....	81
シノニムの拡張.....	82
アプリケーションによるシノニムの使用法.....	82
3. シノニムエディタの使用.....	83
シノニムエディタの画面構成.....	84
シノニム属性の表示および編集.....	111
多言語のタイトルおよび説明の設定.....	121
モデル表示を使用したシノニムの拡張.....	125
データプロファイリング特性の表示.....	136
シノニムまたはセグメントのデータプロファイリング.....	137
データプロファイリングフィールド.....	138
クラスタ JOIN の作成.....	145
OLAP 分析のディメンション定義.....	154
シノニムエディタモデル表示でのディメンションビルダの使用.....	154
ビジネスビューの作成.....	158
シノニムへの一時項目 (DEFINE) の追加.....	167
シノニムフィルタの作成.....	171
シノニムへの一時項目 (COMPUTE) の追加.....	177
一時項目内 (DEFINE) に繰り返しフィールド数を格納.....	180
カスタムフィールドの属性および式の定義.....	185
シノニムへのグループフィールドの追加.....	194
データベース管理者セキュリティの使用.....	200
アクセスタイプの選択.....	206
セグメント、フィールド、フィールド値、非表示アクセス制限.....	206

複数ユーザへのセキュリティ制限の適用.....	209
DBA またはユーザパスワードの削除.....	210
4. メタデータとプロシジャの分析.....	213
インパクト分析ツールを使用したプロシジャの分析.....	214
データプロファイリング特性の表示.....	218
シノニムまたはセグメントのデータプロファイリング.....	219
単一フィールドのデータプロファイリング.....	223
5. 関数の使用.....	229
関数カテゴリ.....	230
文字列関数.....	230
ソースデータおよびデコード関数.....	232
日付時間関数.....	232
フォーマット変換関数.....	235
数値関数.....	236
システム関数.....	237
関数の引数指定.....	237
引数の種類.....	237
関数引数の増加.....	238
引数のフォーマット.....	238
一時項目での関数の使用.....	239
文字列関数.....	241
ARGLEN - 文字列の長さを取得.....	242
BITSON - ビットのオンとオフを返す.....	243
BYTVAL - 文字を 10 進数に変換.....	244
CHKFMT - 文字列のフォーマットを確認.....	245
CTRAN - 文字を他の文字に変換.....	246
CTRFLD - 文字列を中央揃え.....	248
EDIT - 文字を抽出または追加.....	249
GETTOK - 文字列 (トークン) を抽出.....	250
LCWORD - 文字列の先頭文字を大文字に変換.....	251
LCWORD2 - 文字列の先頭文字以外を小文字に変換.....	253
LCWORD3 - 文字列の先頭文字以外を小文字に変換.....	254
LJUST - 文字列を左揃え.....	255

LOCASE - テキストを小文字に変換.....	256
OVRLAY - 文字列を上書き.....	257
PARAG - テキストを行に分割.....	258
POSIT - 文字列の開始位置を検索.....	260
PTOA - パック 10 進数を文字に変換.....	261
REVERSE - 文字列の順序を入れ替え.....	262
RJUST - 文字列を右揃え.....	262
SOUNDEX - 文字列を音声的に比較.....	263
SPELLNM - ドルとセントの通貨表記を文字表記に書き替え.....	265
SUBSTR - 文字列を抽出.....	266
UPCASE - テキストを大文字に変換.....	268
データソースおよびデコード関数.....	269
DECODE - 値を置き換え.....	269
LAST - 前の値を抽出.....	272
WHERE 条件による値へのアクセス制限.....	273
日付時間関数.....	275
AYM - 基準となる日付に月数を加えて、新たな年月を求める.....	276
AYMD - 基準となる日付に日数を加えて、新たな日付を求める.....	277
CHGDAT - 日付文字列の表示を変更.....	278
DA 関数 - レガシー日付を整数に変換.....	281
DATEADD - 日付と日付単位の和または差を計算.....	282
DATECVT - 日付フォーマットを変換.....	284
DATEDIF - 2 つの日付の差を計算.....	285
DATEMOV - 日付を有効な位置に移動.....	287
日付構成要素を整数として取得.....	289
DATETRAN - 日付を国際フォーマットに変換.....	290
日付時間値の精度.....	298
マスターファイルの DATEPATTERN.....	302
DMY、MDY、YMD - 2 つの日付の差を計算.....	310
DOWK および DOWKL - 曜日を検索.....	311
DT 関数 - 整数を日付に変換.....	312
FIYR - 会計年度の取得.....	313
FIQTR - 会計四半期の取得.....	315
FIYYQ - カレンダー日付を会計日付に変換.....	318

GREGDT - ユリウス暦から太陽暦フォーマットに変換.....	320
HADD - 日付時間値を増加.....	321
HCONVRT - 日付時間値を文字フォーマットに変換.....	323
HDATE - 日付時間値の日付部分を日付フォーマットに変換.....	324
HDIFF - 2つの日付時間値の差単位数を計算.....	325
HDTTM - 日付値を日付時間値に変換.....	326
HGETC - 現在の日付および時間を日付時間フィールドに格納.....	327
HHMMSS - 現在の時間を取得.....	327
HINPUT - 文字列を日付時間値に変換.....	328
HMIDNT - 日付時間値の時間部分を午前零時に設定.....	329
HNAME - 日付時間構成要素を文字フォーマットで取得.....	330
HPART - 日付時間構成要素を数値フォーマットで取得.....	331
HSETPT - 日付時間値に構成要素を挿入.....	332
HTIME - 日付時間値の時間部分を数値に変換.....	333
JULDAT - 太陽暦からユリウス暦フォーマットに変換.....	334
TIMETOTS - 時間をタイムスタンプに変換.....	335
TODAY - 現在の日付を取得.....	336
YM - 経過月数を計算.....	337
フォーマット変換関数.....	338
ATODBL - 文字列を倍精度浮動小数点数フォーマットに変換.....	338
EDIT - フィールドのフォーマットを変換.....	339
FTOA - 数値を文字フォーマットに変換.....	341
HEXBYT - 10進数を文字に変換.....	342
ITONUM - 整数を倍精度小数点数フォーマットに変換.....	343
ITOPACK - 整数をパック 10進数フォーマットに変換.....	344
ITOZ - 数値をゾーン 10進数フォーマットに変換.....	345
PCKOUT - 指定した長さでパック 10進数を書き込み.....	346
数値関数.....	347
ABS - 絶対値を計算.....	348
BAR - 棒グラフを作成.....	348
CHKPCK - パック 10進数フィールドを検査.....	350
DMOD、FMOD、IMOD - 除算の剰余を計算.....	351
EXP - 「e」を N でべき乗.....	352
INT - 整数を検索.....	353

LOG - 自然対数を計算.....	354
MAX および MIN - 最大値または最小値を検索.....	354
PRDNOR および PRDUNI - 再生可能な乱数を生成.....	355
RDNORM および RDUNIF - 乱数を生成.....	358
SQRT - 平方根を計算.....	359
システム関数.....	360
FGETENV - 環境変数値を取得.....	360
GETUSER - ユーザ ID を取得.....	361
HHMMSS - 現在の時間を取得.....	362
TODAY - 現在の日付を取得.....	362
A. Java バッチ実行ユーティリティによる DB2 Web Query レポートの実行.....	363
Java バッチ実行ユーティリティ要件.....	364
B. フィールドの記述.....	369
フィールドの特性.....	370
フィールド名 - FIELDNAME.....	371
修飾フィールド名の使用.....	373
重複フィールド名の使用.....	375
修飾フィールド名評価時の規則.....	376
フィールドエイリアス - ALIAS.....	380
フィールドエイリアスの実装.....	381
表示データタイプ - USAGE.....	382
表示フォーマットの指定.....	383
データタイプのフォーマット.....	384
整数フォーマット.....	385
倍精度浮動小数点数フォーマット.....	386
単精度浮動小数点数フォーマット.....	387
パック 10 進数フォーマット.....	388
数値の表示オプション.....	389
拡張通貨記号の表示オプション.....	393
文字フォーマット.....	395
日付フォーマット.....	396
日付表示オプション.....	397
日付区切り記号の制御.....	402

日付の変換.....	403
日付フィールドの使用.....	404
数値日付リテラル.....	405
演算式の日付フィールド.....	405
日付フィールドの変換.....	406
日付フィールドの内部表現.....	407
標準外日付フォーマットの表示.....	408
日付フォーマットのサポート.....	409
日付表示オプションを使用した文字および数値フォーマット.....	409
日付時間フォーマット.....	410
日付時間フィールドの記述.....	413
文字フォーマット AnV.....	423
テキストフィールドのフォーマット.....	426
格納データタイプ - ACTUAL.....	428
ACTUAL 属性.....	428
ミッシング値 (Null 値) - MISSING.....	433
ミッシング値の使用.....	435
FML 階層の記述.....	436
データの確認 - ACCEPT.....	438
代替レポートフィールドタイトル - TITLE.....	440
フィールドの説明 - DESCRIPTION.....	441
多言語メタデータ.....	443
一時項目 (DEFINE) の記述 - DEFINE.....	449
一時項目 (DEFINE) の使用.....	451
一時項目 (COMPUTE) の記述 - COMPUTE.....	452
フィルタの記述 - FILTER.....	455
ソートオブジェクトの記述 - SORTOBJ.....	459
マスターファイル DEFINE による日付システム変数の使用.....	461
変数を使用したマスターファイルおよびアクセスファイルのパラメータ化.....	465
文字日付の 日付への変換.....	469
日付パターン変数の指定.....	470
日付パターン定数の指定.....	473
日付パターンサンプル.....	473

はじめに

このマニュアルは、IBM DB2 Web Query for IBM i で使用可能な機能について説明します。

マニュアルの構成

このマニュアルは、以下の章で構成されています。

	章/付録	内容
1	OLAP レポートによるデータ分析	OLAP を使用した場合の利点や、OLAP パネルおよび OLAP コントロールパネルでレポートをカスタマイズする方法について説明します。また、データを制限するための選択条件やソートなどの適用方法、および OLAP 有効レポートのトラブルシューティングについても説明します。さらに、レポートデータを動的に操作することでレポートから多くの手がかりを得るための多様な手段を、OLAP コントロールパネル (OCP) がどのように提供するかについても説明します。コントロールパネルからは、DB2 Web Query OLAP ユーザが使用可能なすべての機能を実行できます。
2	データ記述とアクセスの概要	データソースの記述の概要と、その使用方法について説明します。
3	シノニムエディタの使用	シノニムエディタを使用して、シノニムの表示および編集を行う方法について説明します。
4	メタデータとプロシジャの分析	インパクト分析を使用してプロシジャを分析する方法と、シノニムのフィールドのデータプロファイリングを表示する方法について説明します。

	章/付録	内容
5	関数の使用	関数の概要と種類について説明します。
A	Java バッチ実行ユーティリティによる DB2 Web Query レポートの実行	Java バッチ実行ユーティリティを使用して DB2 Web Query レポートを実行する方法について説明します。
B	フィールドの記述	データソースのフィールドレベルの情報を記述する方法について説明します。

マニュアルの表記法

このマニュアルの表記法は、次のとおりです。

表記	説明
<code>THIS TYPEFACE</code> または <code>this typeface</code>	構文を表します。表記どおりに入力してください。
<i>this typeface</i>	プレースホルダ (または変数)、クロスリファレンス、あるいは重要な用語を表します。
<u>underscore</u>	デフォルトの設定を表します。
this typeface	テキスト中で小文字で指定する必要があるファイル名およびコマンドを強調しています。
Key + Key	キーを同時に押すことを示します。
{ }	2 から 3 つの選択項目を示します。選択項目の 1 つを中括弧 ({ }) を含めずに入力します。
[]	任意指定のパラメータ群を示します。必須ではありませんが、この中から 1 つを選択することも可能です。パラメータのみを入力し、大括弧 ([]) は含めません。コマンド名や、ユーザーインターフェイスで使われている項目は、この記号で囲みます。

表記	説明
	構文中で、いずれか 1 つ選択する項目群を分離します。分離記号 () を含めずに、いずれか 1 つのみ入力します。
...	パラメータを複数回入力可能であることを示します。省略記号 (...) は含めずに、パラメータのみを入力します。
.	間に省略されているコマンドがあるか、後続するコマンドがある (場合も指定できる) ことを表します。

1 OLAP レポートによるデータ分析

OLAP のライセンスが有効になっている場合、DB2 Web Query OLAP (Online Analytical Processing) によって、ビジネス上の重要な決断を下すためのデータの表示および迅速な分析を行うことができます。

トピックス

- 日常の作業 - 一般的な DB2 Web Query の用途
- OLAP レポートの要件
- OLAP レポートの特性
- OLAP データでの 3 通りの作業
- ディメンションと基軸のドリルダウン
- データのソート
- 基軸上での計算の実行
- データの制限
- 傾向の視覚化
- グラフとレポートの表示
- レポートでの基軸の表示のコントロール
- ディメンションの追加と削除
- OLAP レポートの保存
- OLAP レポートとグラフの他のフォーマットでの保存と表示

日常の作業 - 一般的な DB2 Web Query の用途

ここでは、ニューヨークに小規模な会社を所有しており、カリフォルニア州オークランド市のある企業と提携することを検討しているというシナリオについて考察します。社長は月曜日午前中のミーティングに参加しなければならない、どのようにしてフライトを手配するかを決定する必要があります。

オンラインによる手配が一般的です。

まず、マイレージプログラムのポイントを所有している航空会社で、空席のある便を確認します。その航空会社では乗り継ぎが必要なため、他の航空会社の路線と運賃を検索します。

ニューヨークでは、ラガーディア、JFK、ロングアイランドのマッカーサーの各空港から搭乗できます。カリフォルニアでは、オークランドかサンフランシスコへ到着します。

日曜日に出発して、火曜の朝に戻るとすれば、運賃とスケジュールが都合に合えば、カリフォルニア行きは土曜発の便で、帰りは月曜夜の深夜便で戻ることも可能です。

航空会社の検索から始め、次に出発地と行き先、曜日、時間と運賃をオプションとして検索します。

別のアプローチとして、オンラインの航空運送事業者から検索を始め、出発できる時間を入力し、利用できる便と運賃を調べる方法もあります。

選択肢はさまざまありましたが 30 分で検索を終了し、すべての要因を考慮した適切な意思決定ができました。

アクセスする Web サイトは、問い合わせが円滑に行われるよう設計されています。さまざまなメニューや選択用ボックスで、質問の各行を展開しやすくなっています。必須および選択可能な情報は、ユーザに合わせて特定されます。選択肢の経路を前方にたどることも、後戻りして最初から別の経路をたどることも、また元の経路を異なる選択肢でたどり直すことも可能です。

答えが欲しい質問の経路をたどる必要性に加え、調査を容易にするにはサイトでの優れた設計が必要です。ほとんどのユーザにとって、この手順は直観的な処理が可能になりました。

OLAP 有効 DB2 Web Query レポートのデータを分析する際も、同じ手順が適用されます。

OLAP レポートの要件

トピックス

OLAP 有効データ

レポートの OLAP 有効化

OLAP 用語集

OLAP レポートでは、レポートに表示するデータと、レポート自体の両方にいくつかの準備が必要です。多くの場合、この準備はすべて透過的に行われ、ユーザが OLAP レポートの使用を開始する前に終了しています。ただし、OLAP 有効データとレポートの開発担当者や、パーソナルレポートを OLAP 有効レポートにする必要があり、その権限を与えられたユーザには、次に説明する概要が役立ちます。

OLAP 有効データ

DB2 Web Query OLAP レポートは、階層的なデータ構造に基づいて作成されます。たとえば、販売地区の通常の階層には、Region、State、City の各フィールド (降順) を含む GEOGRAPHY カテゴリがあるとします。この階層で最上位にある Region には、GEOGRAPHY 内で利用可能な全地域のリストが含まれます。階層で 2 番目に上位の State には、これらの地域内で利用可能な州のリストが含まれます。データはこのように階層化されています。

DB2 Web Query では、階層構造は通常データソースのマスターファイルとして構成され、そのデータソースを使用するすべてのレポートに対して有効になります。マスターファイルにデータを記述する責任を持った開発者または管理者は、その言語を使用できます。キーワード「WITHIN」で、階層内の各ディメンションの要素を定義します。

さらに、Developer Workbench で作業することにより、さまざまなグラフィカルツールにアクセスすることができます。これらのツールを使用して、フィールドをドラッグアンドドロップするだけで、配置、および階層の構成が容易に実行できます。階層は、すべてのプロシジャに対してグローバルである場合と、1つのプロシジャに対してローカルな場合があります。

- 複数のプロシジャで使用するマスターファイル内のグローバル階層を定義するには、ディメンションビルダを使用します。
- 特定プロシジャのコンポーネントとしてのローカル階層には、ディメンションツールを使用します。このツールで定義した階層は、ソースマスターファイルには影響しません。

レポートの OLAP 有効化

参照

OLAP レポートオプションの設定

OLAP 対応のデータの使用に加えて、レポートの OLAP 分析サポートを有効にする必要があります。OLAP でレポートを有効にする手順は、ユーザが OLAP データと対話する方法およびドリルダウンする方法を指定することで成立します。

主要な対話的操作は、レポート自体の内部で発生します。さらに、2つの補助的なツール、OLAP 選択パネルとコントロールパネルを表示するかどうかを選択できます。

参照

OLAP レポートオプションの設定**OLAP インターフェースオプション**

ユーザは OLAP インターフェースおよび次のドリルダウンオプションを制御することができます。

- **カラム (OLAP コントロールパネルなし) (OFF)** これを使用すると、OLAP コントロールパネルを開く [OLAP] ボタンが使用できます。
- **カラムのみ (Columns only)** OLAP コントロールパネルと OLAP 選択パネルのコントロールがオフになりますが、レポート自体の OLAP 機能は使用できます。右クリックメニューのオプションや、レポート内のフィールドのドラッグアンドドロップ、上下矢印を使用して、フィールドを昇順または降順にソートすることができます。このオプションがデフォルト設定で選択されています。
- **カラムとパネル (Columns with panel)** フィールド見出し左の四角いボタンから OLAP 選択パネルを表示できます。
- **フィルタを上に表示 (Show filters on top)** レポートの上に OLAP 選択パネルが表示されます。基軸、グラフ、およびディメンションの各コントロールが、OLAP、実行、およびリセットのボタンとともにレポート出力上に表示されます。選択パネルの [OLAP] ボタンをクリックすると、コントロールパネルが表示されます。
- **フィルタを下に表示 (Show filters on bottom)** レポートの下に OLAP 選択パネルが表示されます。基軸、グラフ、およびディメンションの各コントロールが、OLAP、実行、およびリセットのボタンとともにレポートの下に表示されます。選択パネルの [OLAP] ボタンをクリックすると、コントロールパネルが表示されます。
- **OLAP ボタンを表示 (Show Panel in Report)** OLAP 選択パネルが非表示の状態でも OLAP レポートが表示されます。レポートから各種の分析を行うことができます。[OLAP] ボタンの横に、選択条件が表示されます。
- **タブ表示 (Show Tabbed)** 複数のディメンションを含む OLAP レポートの場合は、このオプションにより、ディメンション名をラベルとするタブの下でディメンション要素をグループ化できます。

オートドリルダウンオプション

これらのオプションでは、選択したレポート列を上から下へ、または下から上へ即時にソートすることができます。

- **なし (None)**
オートドリルダウンを無効にします。

□ **ディメンション (Dimensions)**

レポートとグラフの両方でディメンションのオートドリルダウンを有効にします。

□ **ディメンションと基軸 (Dimensions and Measures)**

レポートとグラフ両方のディメンション、およびレポートの基軸で、オートドリルダウンを有効にします。

注意：スタイルシート (存在する場合) の明示的なドリルダウンは、OLAP 対応のリンクより優先されます。明示的なドリルダウンに関連付けされたリンクをクリックすると、動作は [オートドリルダウン] (AutoDrill) または [すべて] (All) ではなく、スタイルシートによって定義されます。

OLAP 用語集

下表は、DB2 Web Query OLAP ツールで作業する際に便利な OLAP 用語を示しています。用語のいくつかは直接 OLAP 選択パネルおよび OLAP コントロールパネルのインターフェースに反映されています。その他は有効なバックグラウンド情報です。

この表の左の列は用語、右の列はその定義を示します。

用語	定義
OLAP ディメンションツール	関連要素のグループまたはリスト。通常は階層別に構成されます。たとえば、Location ディメンションには、階層内に配置された要素 Country、Region、State および City が含まれます。ここでは Country が最上位レベルであり、City が基礎レベルです。ディメンションデータは通常、基軸項目を表現します。
階層	ディメンション内の要素の論理的な親子構造。
基軸	関連する別の要素の数量を指定する項目タイプ。基軸は一般に価格または数量を定義します。たとえば、Units、Revenue および Gross Margin は Account ディメンションの基軸であり、販売数量、売上金額、および利益幅をそれぞれ指定します。

用語	定義
ピボット	列から行へ、またはその逆にフィールド (またはフィールドグループ) を移動してレポートの表示を操作 (回転) します。

OLAP レポートの特性

OLAP 有効レポートには、その他の DB2 Web Query レポートとは異なる多くの機能があります。


基本の OLAP レポートは、下図のように表示されます。

QUARTER	Store Name:	PRODTYPE	Quantity:	Line Cost Of Goods Sold
Q1	AV VideoTown	Analog	18,449	3,969,296.00
		Digital	22,206	5,109,400.00
	Audio Expert	Analog	78,449	16,467,146.00
		Digital	105,983	25,092,678.00
	City Video	Analog	6,287	1,315,015.00
		Digital	7,196	1,607,513.00
	Consumer Merchandise	Analog	6,980	1,542,036.00
		Digital	14,957	3,251,090.00
	TV City	Analog	19,077	3,772,119.00
		Digital	41,307	10,128,967.00
Web Sales	Analog	545	124,366.00	
	Digital	829	190,201.00	
eMart	Analog	97,128	21,152,262.00	
	Digital	108,221	24,990,368.00	
Q2	AV VideoTown	Analog	11,781	2,863,655.00
		Digital	27,377	5,928,507.00
	Audio Expert	Analog	57,944	11,868,758.00

すべての OLAP ユーザは、OLAP レポートを構成する次の分析機能の利点を活用できます。

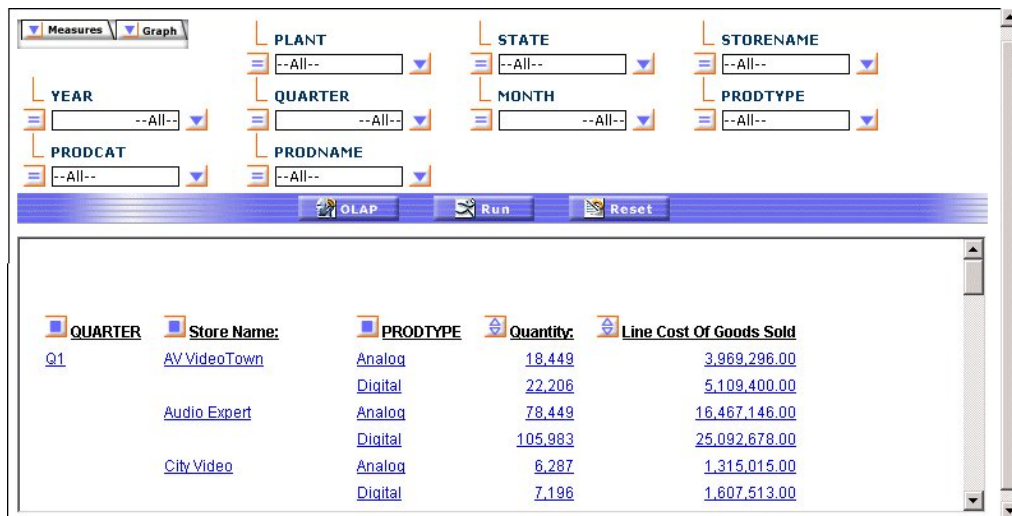
- **ハイパーリンク** OLAP レポートの値は、通常、関連情報へドリルダウンできるハイパーリンクになっています。

OLAP 設定に応じて、ハイパーリンクはディメンションフィールド (レポートをソートする) および基軸フィールド (数値データを表示する) の両方で有効な場合と、ディメンションフィールドのみで有効な場合があります。詳細は、18 ページの「[レポートの OLAP 有効化](#)」を参照してください。


- **コンテキストメニュー** 列タイトルのいずれかを右クリックして、分析オプションのメニューにアクセスできます。このオプションは、ディメンションおよび基軸に関連付けられたタスクに応じて若干異なります。
- **ソートボタン**  基軸 (レポートの本体を構成するフィールド) には、その横に青いひし形が表示されます。ひし形の上部または下部のいずれかをクリックすると、データを昇順または降順で即時にソートされます。
- **ディメンションおよび基軸のドラッグアンドドロップ機能**
 - ソートフィールドをドラッグアンドドロップして、縦 (BY) から横 (ACROSS) またはその逆に、ソート順を切り替えることができます。
 - 内側から外側 (またはその逆) へソートフィールドをドラッグして、ソート順を変更することができます。
 - ある位置から他へ基軸をドラッグして、データが表示される順序を変更できます。

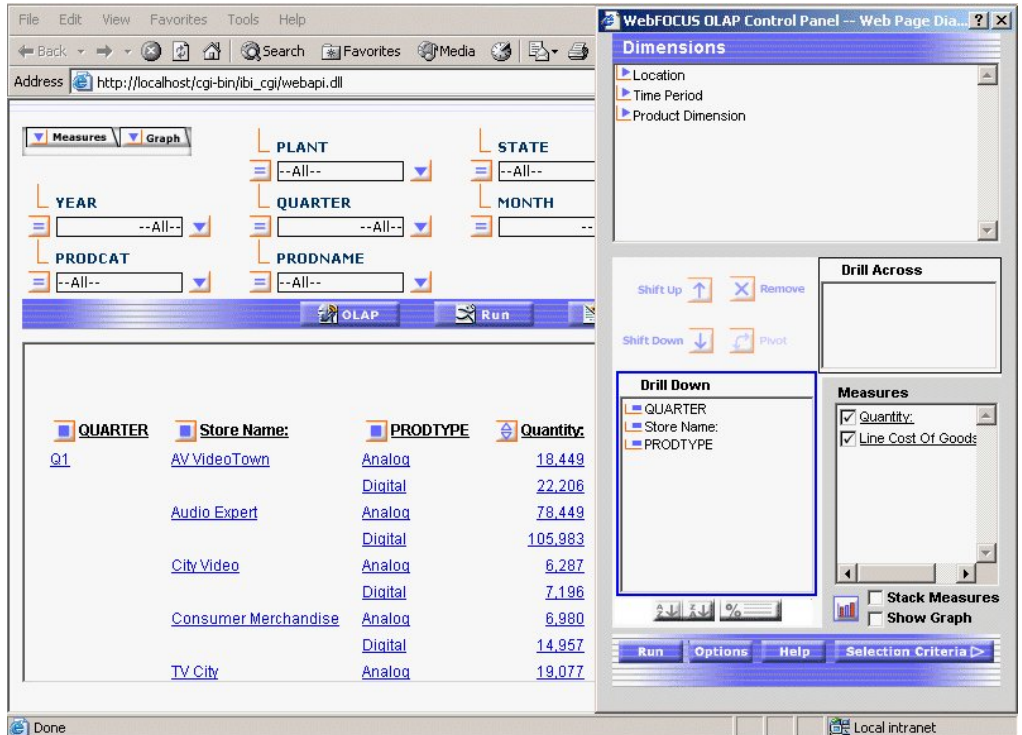
レポート自体の機能以上に、OLAP オプションは、特定のレポートに影響するインターフェースおよびドリルダウン設定による影響が大きくなります。これらの選択は、次のツールへのアクセス権限を有するかどうかによって決まります。

- **選択パネル** このツールを使用できる場合、パネルは下図のようにレポートの上、またはレポートの下に表示されます。OLAP ドロップダウンリストに入力可能な項目の制限値は、5000 です。詳細は、25 ページの「[選択パネル](#)」を参照してください。



QUARTER	Store Name:	PRODTYPE	Quantity:	Line Cost Of Goods Sold
Q1	AV VideoTown	Analog	18,449	3,969,296.00
		Digital	22,206	5,109,400.00
	Audio Expert	Analog	78,449	16,467,146.00
		Digital	105,983	25,092,678.00
	City Video	Analog	6,287	1,315,015.00
		Digital	7,196	1,607,513.00

- **OLAP** コントロールパネル このツールが使用可能な場合、ソートフィールド (BY または ACROSS) 横の  ボタンがアクティブになります。四角形または [OLAP] ボタンをクリックして、下図のようなコントロールパネルを開きます。詳細は、27 ページの「[OLAP コントロールパネル](#)」を参照してください。



QUARTER	Store Name:	PRODTYPE	Quantity:
Q1	AV VideoTown	Analog	18,449
	Audio Expert	Digital	22,206
	City Video	Analog	78,449
	Consumer Merchandise	Digital	105,983
	TV City	Analog	6,287
		Digital	7,196
		Analog	6,980
		Digital	14,957
		Analog	19,077

OLAP データでの 3 通りの作業

トピックス

レポート

選択パネル

OLAP コントロールパネル

OLAP データでの作業には 3 通りあります。レポート自体から、選択パネルから、およびコントロールパネルから、それぞれ作業できます。このマニュアルは、それぞれの位置から何を実行できるか、特定の OLAP 設定に最適かつ効果的な手法はどれかを理解できるよう構成されています。

レポート

レポート自体から、広い範囲の基本的な分析機能を実行できます。レポートに加えた変更は、即時に実装されます。すべての OLAP ユーザは次のタスクを実行できます。

- 基軸のデータを、昇順または降順のいずれかでソートする。
- 基軸、ディメンション、または両方をドリルダウンする (19 ページの「[OLAP レポートオプションの設定](#)」で説明する設定によって異なる)。
- 現在のレポートのフィールドを非表示にします。
- ディメンション階層で非表示のフィールドを表示し、それらをレポートに追加します。
- 縦 (BY) ソートフィールドを横 (ACROSS) ソートフィールドに、またはその逆に変更する。
- ソートフィールドを削除する。
- ピアグラフの列を追加する。数値データ (基軸) の傾向を視覚化する場合に役立ちます。
- 表形式レポートの上のフレームにデータをグラフで表現して表示します。

レポート出力 OLAP 分析の例については、16 ページの「[日常の作業 - 一般的な DB2 Web Query の用途](#)」を参照してください。

選択パネル

OLAP 選択パネルがオンになっている場合、階層のディメンションで特定の値を選択して、レポート内のデータを迅速に制限できます。ディメンションごとにドロップダウンリストがあります。1つ以上のディメンションリストから複数の値を選択して、レポート出力を絞り込むことができます。OLAP ドロップダウンリストに入力可能な項目の制限値は、5000 です。

レポートにディメンション要素を追加したい場合、それを選択パネルからレポートフレームにドラッグできます。受容可能な場所ではカーソルはプラス記号 (+) に変化します。

各ディメンションの左側に関係演算子のボタンがあります。このボタンでは、選択条件を迅速に定義できる基本の数値演算子の選択を切り替えられます。演算子には、「等しい」、「等しくない」、「より大きい」、「以上」、「より小さい」、「以下」、「含む」、「含まない」、「開始」、「開始以外」があります。詳細は、46 ページの「[選択条件の関係演算子](#)」を参照してください。

下図は、選択パネルの各ディメンションで、Equal to 演算子を選択した場合を示しています。

The screenshot shows the OLAP selection panel with the following dimensions and filters:

- Measures: Graph
- PLANT: --All--
- STATE: --All--
- STORENAME: eMail
- YEAR: --All--
- QUARTER: 2
- MONTH: 08
- PRODCAT: PDA Devices
- PRODNAME: --All--
- PRODTYPE: Digital

Buttons: OLAP, Run, Reset

Product Name:	Quantity:	Line Cost Of Goods Sold
ZT Digital PDA - Commercial	13,591	4,743,259.00
ZC Digital PDA - Standard	2,429	604,821.00

ディメンションフィールドの名前は、代わりに列タイトルが指定されている場合でも、マスターファイルで定義されたとおりに表示されることに注意してください。

さらに、選択パネルからはレポートでの基軸の表示をカスタマイズできます。パネルの左上にある [基軸] (Measures) または [グラフ] (Graphs) のいずれかの矢印をクリックして、基軸のリストを表示できます。

- [基軸] (Measures) の矢印で、選択した基軸を表示または非表示にできます。また、単純なピアグラフの列表示して傾向を明確にすることもできます。
- [グラフ] (Graphs) 矢印からは、グラフを作成する基軸を選択できます。また、7つの基本グラフタイプ、縦および横のピアグラフと線グラフ、エリアグラフおよび円グラフから 1つを指定できます。

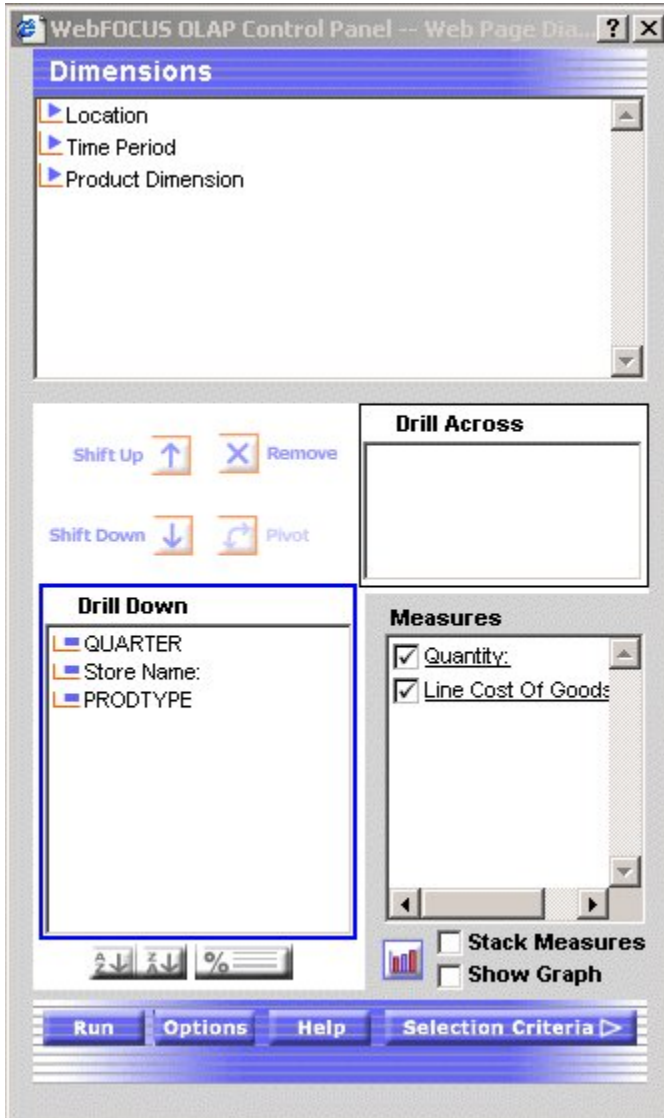
選択パネルはサイズを変更できることに注意してください。レポートウィンドウをサイズ変更すると、ディメンション、基軸およびグラフのコントロールが浮上し、フレーム内に表示し続けることができます。

選択パネルの下に、OLAP、実行、リセットの 3つのボタンが表示されます。

- **OLAP** OLAP コントロールパネル (OCP) が開きます。
- **実行 (Run)** 現在の設定を組み合わせるレポートを実行します。
- **リセット (Reset)** レポートのすべてのコントロールを以前の状態、つまり、現在の設定が選択される前のレポートが最後に実行された後の状態にリセットします。

OLAP コントロールパネル

コントロールパネルからは、下図のように、DB2 Web Query OLAP ユーザが使用可能なすべての機能を実行することができます。



コントロールパネルのメインウィンドウには、次のコンポーネントがあります。

- **ディメンション (Dimensions) ボックス** 現在のレポートが使用するソースデータの階層構造が反映されます。たとえば、Location ディメンションには Region、State および City フィールドが格納され、Region は複数の State で構成され、各 State には複数の City が格納されています。ディメンション名の左にある矢印をクリックすると、構成する要素が表示されます。ここに表示されるフィールドは、選択パネルにも表示されます。
- **ドリルダウン (Drill Down)/ドリルアクロス (Drill Across) ボックス** レポートをソートするフィールドが表示されます。[ドリルダウン] (Drill Down) フィールドを [ドリルアクロス] (Drill Across) フィールドに、またはその逆に「回転」して、レポート内の位置を入れ替えます。また、これらのタスクはレポート内のフィールドをドラッグしても実行できます。
- **基軸 (Measures) ボックス** レポートの本体 (通常は数値フィールド) があります。基軸の表示モードは、基軸の横のチェックボックスをクリックして変更できます。オプションとして、[表示] (Display)、[非表示] (Hide) および [グラフ表示] (Show Graph) があります。これは、選択パネルの [基軸] (Measures) コントロールから使用できるオプションと同等です。

最も頻繁に使用される機能は、OLAP レポートおよび選択パネルから直接使用できますが、いくつかの機能はコントロールパネルからのみ実行できます。

コントロールパネルからのみ実行可能な操作には、次のものがあります。

- ディメンションのソートオプションを使用する。降順と昇順 (A>Z または Z>A)、ソートフィールドの値を最上位または最下位のいずれかで指定した数値に制限する、またはレポートの各行に順位を割り当てます。詳細は、29 ページの「[データのソート](#)」を参照してください。
- 数値データをグループ化して、任意の数のグループ (たとえば、100 分位数、10 分位数、4 分位数など) をソートして表示するオプション。詳細は、39 ページの「[数値データの分位グループ化](#)」を参照してください。
- 省略された、または既存の文字、日付および範囲の使用に基づいた選択条件の定義。詳細は、45 ページの「[データの制限](#)」を参照してください。
- PDF または Excel フォーマットで OLAP 出力を保存する。詳細は、72 ページの「[OLAP レポートとグラフの他のフォーマットでの保存と表示](#)」を参照してください。
- 複数の基軸を積み重ねて、レポートの幅を制限する。詳細は、63 ページの「[基軸の積み重ね](#)」を参照してください。

ディメンションと基軸のドリルダウン

OLAP レポートおよびグラフのディメンション、さらにレポートの基軸でもドリルダウンできます。これらの設定は、次の必要なリンクで有効になります。

- **ディメンション (Dimensions)** レポートとグラフでディメンションのオートドリルダウンを有効にします。
- **ディメンションと基軸 (Dimensions and Measures)** レポートとグラフ両方のディメンションで、またレポートの基軸で、オートドリルダウンを有効にします。
- **なし (None)** オートドリルダウンを無効にします。このオプションがデフォルト設定で選択されています。

Developer Workbench では、[レポートオプション] (Report Options) の [フォーマット] (Format) タブからドリルダウンオプションを設定できます。この設定についての詳細は、19 ページの「[OLAP レポートオプションの設定](#)」を参照してください。

データのソート

トピックス

基軸のソート

ディメンションのソート

数値データの分位グループ化

階層内ディメンションの値、レポート本体を構成する量的基軸の値のいずれかまたは両方に基づいて、OLAP レポートのデータをソートすることができます。ソートオプションは、ソートデータの特性によって異なります。詳細は、30 ページの「[基軸のソート](#)」および 34 ページの「[ディメンションのソート](#)」を参照してください。

また、数値データをグループ化して、任意の数のグループ (100 分位数、4 分位数、10 分位数など) に分類することができます。詳細は、39 ページの「[数値データの分位グループ化](#)」を参照してください。

基軸のソート

手順/構文

OLAP レポート基軸を昇順または降順にソートするには
コントロールパネルから基軸を昇順/降順でソートするには
ソートした基軸のサブセットデータを表示するには
基軸のソート条件を削除するには
ソートフィールドを非表示にするには

OLAP レポート内の数値基軸に集計とソートを同時に適用し、データを降順または昇順にソートします。他のすべてのフィールドはこれに対応してソートされます。

ソート基軸では、指定した個数の最大値 (降順の場合) または最小値 (昇順の場合) に、レポートを限定することができます。


基軸をソートする場合、レポートの中間合計、中間見出し、または中間脚注は、特定のソートフィールドと関係付けられており、レポートが基軸フィールドの値で再度ソートされると意味がなくなるため、自動的に表示から除外されます。

注意：基軸によるソートは、基軸が積み重ねられているレポートでは使用できません。詳細は、64 ページの「[基軸の非表示と表示](#)」を参照してください。

手順

OLAP レポート基軸を昇順または降順にソートするには

基軸の値を降順でソートするには、次の手順を実行します。


-  ボタンを選択すると、[新規作成](New)、[開く](Open)、[保存](Save)、[実行](Run)、その他のオプションがポップアップウィンドウで表示されます。

または

- 基軸を右クリックして、メニューから [降順ソート] (Sort by Highest) を選択します。

レポートは自動的に実行されます。これにより、最高値が最初の行に表示されます。ひし形ボタンの上部は塗りつぶした青で表示され、現在のソート方向を示します。

基軸の値を昇順にソートするには、次の手順を実行します。

-  ボタンを選択すると、[新規作成](New)、[開く](Open)、[保存](Save)、[実行](Run)、その他のオプションがポップアップウィンドウで表示されます。

または

- 基軸を右クリックして、メニューから [昇順ソート] (Sort By Lowest) を選択します。最低値が最初の列に表示されます。ひし形の下部が塗りつぶした青で表示されます。

ヒント：基軸がソートされた後、ひし形ボタンのいずれかの半分をクリックすると、基軸のソートが反転します。ひし形の上下いずれかの半分にマウスを置くと、ひし形のその半分を次回クリックした場合のソート順を示すメッセージが表示されます。

手順

コントロールパネルから基軸を昇順/降順でソートするには

1. OLAP コントロールパネルを開きます。
2. [基軸] (Measures) ボックスの基軸をクリックして、[ソートオプション] (Sort Options) パネルを開きます。ここで、[基軸] (Measures) チェックボックスはクリックしないでください。このオプションは、基軸のソートではなく、表示を制御します。
[ソート] (Sort) ボックスが選択されていることを確認します。この設定は、選択した基軸にソート指定を適用するために必要です。
3. [降順] (High-to-Low) または [昇順] (Low-to-High) オプションを選択して、適用するソート順を指定します。デフォルトのソート順は、降順です。
4. [OK] をクリックします。
ソートパネルの代わりに [基軸] (Measures) ボックスが表示されます。ここで、基軸が青で表示され、定義済みのソート指定が存在することが示されます。
5. [実行] (Run) をクリックすると、基軸にソートが適用されたレポートが表示されます。
ソートした基軸の横にあるひし形ボタンは、ソート順を反映して変化します。降順の場合、ひし形の上半分が塗りつぶしの青、降順の場合は下半分が塗りつぶしの青で表示されます。

注意

- OLAP レポートの基軸をソートする場合、レポートは自動的に実行されます。ただし、コントロールパネルが開いていると、コントロールパネルの現在の変更はすべて適用されます。

- OLAP リクエストに横 (ACROSS) ソートフィールドがある場合、その基軸は、レポートに ACROSS 値ごとに複数回表示されます。基軸にソートを適用する場合、そのソートは基軸の最初の列で実行され、それに続くすべてのインスタンスに反映されます。ひし形ボタンの該当する半分は、最初のインスタンスにのみ塗りつぶしで表示されます。追加のソートを実行する場合は、基軸の最初の部分から行う必要があります。

手順

ソートした基軸のサブセットデータを表示するには

レポート内にあるレコードの合計数のサブセットのみを表示するよう選択できます。

1. OLAP コントロールパネルを開きます。
2. [基軸] (Measures) ボックスの基軸名をクリックして、[ソートオプション] (Sort Options) パネルを開きます ([基軸] (Measures) チェックボックスをクリックしないでください。これは基軸のソートではなく、表示の制御に使用します)。

[ソート] (Sort) チェックボックスが選択されていることを確認します。基軸にソート指定を適用するには、この設定が必要です。

3. [ランキング] (Rank) チェックボックスを選択し、レポートに含めるソートフィールドの数を指定します。入力ボックスの左側には、現在のソート順を示す [上位] (Highest) または [下位] (Lowest) が表示されます。

- [上位] (Highest) または [下位] (Lowest) 右側のスピンドットを使用して、ソートフィールドの数を増減します。

または

- テキストボックスにカーソルを置いて、値を入力します。

ソートフィールドのデフォルト数は 5 です。

4. [OK] をクリックします。

ソートパネルの代わりに [基軸] (Measures) ボックスが表示されます。ここで、基軸が青で表示され、定義済みのソート指定が存在することが示されます。

5. [実行] (Run) をクリックして、指定した数のソート値を持つレポートを表示します。

手順

基軸のソート条件を削除するには

表示、非表示に関わらず、基軸のソート指定は削除することができます。

1. OLAP コントロールパネルを開きます。

2. [基軸] (Measures) ボックスで、ソート指定を削除する基軸をクリックします。
3. [ソート] (Sort) チェックボックスの選択を解除します。
4. [OK] をクリックします。

手順

ソートフィールドを非表示にするには

1. Web Query 開発ツールを使用して、OLAP レポートを作成します。
2. OLAP コントロールパネルを開きます。
3. OLAP コントロールパネルの[ドリルダウン](Drill Down) ウィンドウで、[Country] フィールドをダブルクリックします。結果のウィンドウで、[非表示] (Hide) のチェックをオンにします。
4. [OK] をクリックします。

ソートフィールドのアイコンの色が反転表示されます。[ドリルダウン] (Drill Down) ウィンドウは、下図のように表示されます。



ディメンションのソート

手順/構文

ディメンションのソート順を変更するには

ソート値の表示を制限するには

縦にソートされたレポート行を順位付けするには

OLAP レポート内のソートフィールドを再配置するには

コントロールパネルからソートフィールドを再配置するには

OLAP レポート内の行と列を回転するには

コントロールパネルで行と列を回転するには

ソートフィールドを非表示にしてフィールドをソートするには


OLAP 階層のディメンションをソートするにはいくつかの方法があります。次のことが可能です。

- 昇順または降順でソートされたデータの順序を制御する。
- 最上位または最下位の値のいずれかで指定した数にソートフィールドの値を制限する。
- 縦にソートされたレポートの各行に順位を割り当てる。
- レポートのソートフィールド位置をシフトする。たとえば、[State] の次に [Product] でソートし、続いてその逆方向へ変更できます。
- 縦 (BY) ソートフィールドを横 (ACROSS) ソートフィールドに、またはその逆に回転する。
- 関連したソートが有効な間、レポートのソートフィールドを非表示にする。たとえば、[QUARTER] 列を表示せずに四半期でデータをソートします。
- 数値データを、任意の数の分位 (たとえば、100 分位数、10 分位数など) にグループ化する。

手順

ディメンションのソート順を変更するには

1. コントロールパネルを開きます。
2. [ドリルダウン] (Drill Down) または [ドリルアクロス] (Drill Across) ボックスからフィールドの 1 つを選択します。

3. [ソート](Sort)  ボタンを選択すると、[新規作成](New)、[開く](Open)、[保存](Save)、[実行](Run)、その他のオプションがポップアップウィンドウで表示されます。


[ソート](Sort) ウィンドウが開きます。

4. [ソート順](Sort Order) の下の [昇順](Low to High) または [降順](High to Low) オプションボタンを選択します (ディメンションのデフォルトは [昇順](Low to High) です)。
5. [OK] をクリックします。
メインの [コントロールパネル](Control Panel) ウィンドウが再度開きます。
6. [実行](Run) をクリックしてレポートを実行します。

手順

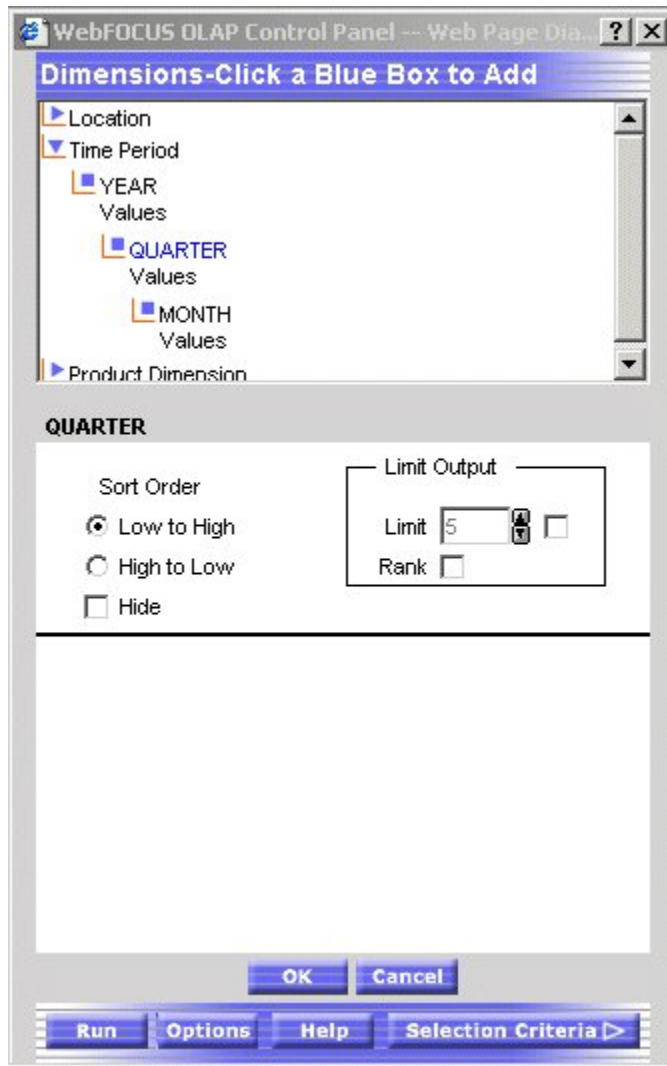
ソート値の表示を制限するには

ソートフィールドの値の表示を、上位または下位の特定の順位までに制限するには、次の手順を実行します。

1. OLAP コントロールパネルを開きます。
2. [ドリルダウン](Drill Down) リストボックスからフィールドを選択します。
3. [ソート](Sort)  ボタンを選択すると、[新規作成](New)、[開く](Open)、[保存](Save)、[実行](Run)、その他のオプションがポップアップウィンドウで表示されます。

ソートパネルが表示されます。

4. [ソート順] (Sort Order) の下の、下図のような OLAP コントロールパネルの [昇順] (Low to High) または [降順] (High to Low) オプションボタンを選択します。



5. [出力制限] (Limit Output) の下の [制限値] (Limit) チェックボックスをクリックし、入力エリアで値を選択するか、入力します。
6. [OK] をクリックします。
メインの [コントロールパネル] (Control Panel) ウィンドウが再度開きます。

7. [実行] (Run) をクリックして、レポートを実行します。

手順

縦にソートされたレポート行を順位付けするには

1. OLAP コントロールパネルを開きます。
2. [ドリルダウン] (Drill Down) リストボックスからフィールドを選択します。
3. [ソート] (Sort)  ボタンを選択すると、[新規作成] (New)、[開く] (Open)、[保存] (Save)、[実行] (Run)、その他のオプションがポップアップウィンドウで表示されます。
[ソート] (Sort) ウィンドウが開きます。
4. [ソート順] (Sort Order) の下の [昇順] (Low to High) または [降順] (High to Low) オプションボタンを選択します。
5. [順序] (Rank) チェックボックスをクリックします。
6. ランク付けするソートフィールドの数を制限するには、[制限値] (Limit) チェックボックスをクリックして、入力エリアで値を選択するか、入力します。
 - [降順] (High to Low) オプションボタンが選択されている場合、指定された値までの上位をランク付けできます。
 - [昇順] (Low to High) オプションボタンが選択されている場合、指定された値までの下位をランク付けできます。
7. [OK] をクリックします。
メインの [コントロールパネル] (Control Panel) ウィンドウが再度開きます。
8. [実行] (Run) をクリックして、レポートを実行します。

手順

OLAP レポート内のソートフィールドを再配置するには

- レポート内のデータのソート順や表示は、変更することができます。たとえば、[State] の次に [Product] でソートし、続いてその逆方向へ変更できます。再配置は次のように行います。
- 縦 (BY) ソートフィールドでは、新しい列位置にフィールドをドラッグアンドドロップします。

- 横 (ACROSS) ソートフィールドでは、下位のフィールドを上位へドラッグアンドドロップするか、その逆を実行します。

どちらの場合でも、フィールドをドロップできる位置では、カーソルがプラス記号 (+) に変わります。受け入れられない場所では、内部に斜線が引かれた円になります。

手順 **コントロールパネルからソートフィールドを再配置するには**

1. OLAP コントロールパネルを開きます。
2. [ドリルダウン] (Drill Down) または [ドリルアクロス] (Drill Across) ボックスからフィールドを選択します。
3. 必要な位置にフィールドが移動するまで、[シフトアップ] (Shift Up) または [シフトダウン] (Shift Down) 矢印をクリックします。
必要に応じて他のフィールドでもこの手順を繰り返します。
4. [実行] (Run) をクリックして、レポートを実行します。

手順 **OLAP レポート内の行と列を回転するには**

データの縦 (行方向) ソートフィールドを、横 (列方向) ソートフィールドに、すばやく変更することができます。または、その逆も可能です。変更は次のように行います。


- 縦 (BY) ソートフィールドから横 (ACROSS) ソートフィールドに変更するには、列タイトルの行の上へフィールドをドラッグアンドドロップします。
- 横 (ACROSS) ソートフィールドから縦 (BY) ソートフィールドに変更するには、列タイトルの行の適切な位置へフィールドをドラッグアンドドロップします。

どちらの場合でも、フィールドをドロップできる位置では、カーソルがプラス記号 (+) に変わります。(受け入れられない位置では、内部に斜線が引かれた円になります。)

手順 **コントロールパネルで行と列を回転するには**

データの縦 (行方向) ソートフィールドを、横 (列方向) ソートフィールドに変更することができます。または、その逆も可能です。


1. OLAP コントロールパネルを開きます。
2. [ドリルダウン] (Drill Down) または [ドリルアクロス] (Drill Across) ボックスで回転する行または列のタイトルを選択します。

3.  ボタンを選択すると、[新規作成] (New)、[開く] (Open)、[保存] (Save)、[実行] (Run)、その他のオプションがポップアップウィンドウで表示されます。タイトルが新しい場所に表示されます。
4. [実行] (Run) をクリックして、レポートを実行します。

手順

ソートフィールドを非表示にしてフィールドをソートするには

データのソートに使用するフィールドをレポートの表示から除外するには、次の手順を実行します。

1. OLAP コントロールパネルを開きます。
2. [ドリルダウン] (Drill Down) または [ドリルアクロス] (Drill Across) ボックスからフィールドを選択します。
3. [ソート] (Sort)  ボタンを選択すると、[新規作成] (New)、[開く] (Open)、[保存] (Save)、[実行] (Run)、その他のオプションがポップアップウィンドウで表示されます。

[ソート] (Sort) ウィンドウが開きます。

4. [ソート順] (Sort Order) の下の [非表示] (Hide) チェックボックスをクリックします。
5. [OK] をクリックします。

メインの [コントロールパネル] (Control Panel) ウィンドウが再度開きます。

6. [実行] (Run) をクリックしてレポートを実行します。

ヒント：非表示のソートフィールドを表示するには、この手順を繰り返して [非表示] (Hide) チェックボックスの選択を解除します。

数値データの分位グループ化

手順/構文

OLAP レポートでデータをグループ化して並べて表示するには

表形式のレポートでは、数値データを 100 分位数、10 分位数、4 分位数などのグループに配分して表示することができます。たとえば、生徒のテストの点数を 10 分位数でグループ化し、生徒のどれがクラスの上位 10 パーセントに入るかを判断できます。または、販売員のどれが総売上高に基づく全販売員の上位半分に入るかを判断できます。

グループ化は、選択された縦 (BY) フィールドの値に基づいて実行され、データは指定した数のグループに可能な限り均等に割り当てられます。

データをグループに配分して表示する場合、次の処理が実行されます。

- 新しいフィールド (デフォルト設定で TILE とラベルが付く) がレポート出力に追加され、TILE フィールドの各インスタンスに割り当てられた分位番号を表示します。OLAP コントロールパネルの [分位] (Tiles) セクションで列タイトルを変更できます。
- 分位は、リクエストにあるすべての上位のソートフィールド内で計算され、TILE フィールドより上位のソートフィールドの値が変更されるたびに再計算されます。
- インスタンスの個数は、TILE フィールドを基に集計されます。リクエストに下位セグメントからのフィールドが表示された場合、TILE フィールドのインスタンス 1 つに複数のレポート行が表示される可能性があります。
- TILE フィールドの値が等しいインスタンスは、同一のグループに配分されます。たとえば、次のように 3 つのグループに配分されるデータについて考察します。

1
5
5
5
8
9

この場合、インスタンスを等しい数のレコードを含むグループに分けて、次のテーブルを生成します。

グループ	データ値
1	1,5
2	5,5
3	8,9

ここでは、等しいデータ値はすべて同一グループに入れる必要があるため、グループ 2 の 5 は、グループ 1 に移動され、グループ 2 は空になります。最後のグループは、次のテーブルのように表示されます。

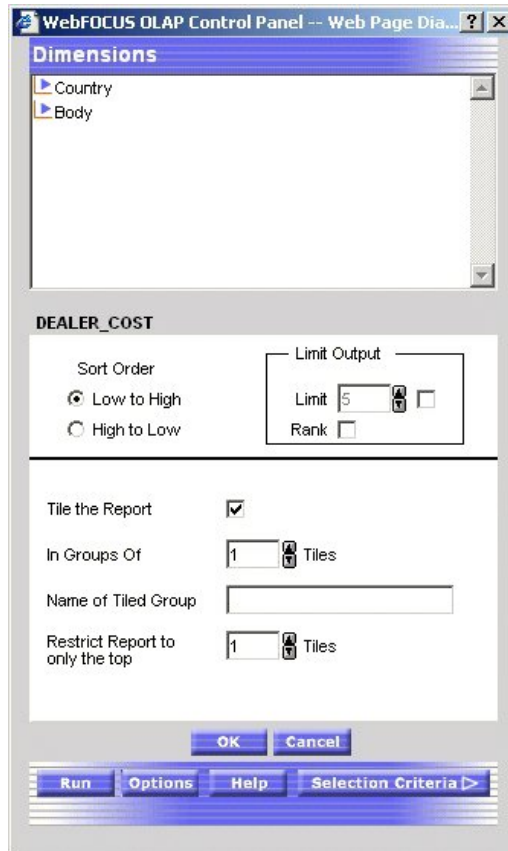
分位番号	データ値
1	1,5,5,5
2	
3	8,9

手順

OLAP レポートでデータをグループ化して並べて表示するには

1. OLAP コントロールパネルを開きます。
2. [ドリルダウン](Drill Down) リストボックスから数値または日付フィールドを選択します。
3. [分位](Tiles) ボタンをクリックします。

下図のように、OLAP コントロールパネルの下部に [分位] (Tiles) パネルが開きます。



4. [レポート分位] (Tile the Report) のチェックをオンにします。
5. [対象グループ] (In Groups Of) テキストボックスで、データのグループ化に使用する分位数を選択します。たとえば、100 を選択すると 100 分位数、10 を選択すると 10 分位数、というようになります。
6. [分位グループ名] (Name of Tiled Group) 入力ボックスに、タイルカラムの名前を入力します。
7. [レポート制限] (Restrict Report to only the Top) 入力エリアで、レポートに表示するタイルグループの数を選択します。
8. オプションとして、[ソート順] (Sort Order) オプションボタンを選択します。

- [降順] (High to Low) を選択すると、データが降順でソートされ、最上位のデータ値が分位 1 に配置されます。
 - [昇順] (Low to High) を選択すると、データが昇順でソートされ、最下位のデータ値が分位 1 に配置されます。このオプションがデフォルト設定で選択されています。
9. レポートに表示する値に最上位のタイル値を指定する場合、[制限値] (Limit) 入力エリアから値を選択します。たとえば、制限値に「3」と入力した場合、分位番号 3 以上のグループに割り当てられたデータ行はレポートに表示されません。
 10. [OK] をクリックして選択を受け入れ、メインの [コントロールパネル] (Control Panel) ウィンドウに戻ります。
 11. [実行] (Run) をクリックしてレポートを再実行し、表示します。

基軸上での計算の実行

手順/構文

基軸へ計算を適用するには

参照

基軸上で実行できる計算

OLAP レポートの基軸にある数値データに関する平均、パーセント、集計などの標準的な計算を実行できます。

手順

基軸へ計算を適用するには

1. レポートを実行します。
2. OLAP コントロールパネルを開きます。
3. [基軸] (Measures) ボックスの基軸をクリックします。

[ソートオプション] (Sort Options) パネルが表示されます。[基軸] (Measures) チェックボックスをクリックしないでください。ここでは基軸のソートではなく、表示をコントロールします。

4. [基軸演算] (Measure Calculations) の下の矢印をクリックし、ドロップダウンリストから計算式を選択します。

[なし] (None) がデフォルト値です。詳細は、44 ページの「[基軸上で実行できる計算](#)」を参照してください。

5. [OK] をクリックします。

ソートパネルに代わって [基軸] (Measures) ボックスが表示されます。ここでは、基軸の先頭に選択した計算が表示されます。

6. [実行] (Run) をクリックします。

適用した計算が、列タイトルに追加されます。

参照

基軸上で実行できる計算

下表の左の列は計算のタイプを示し、右の列はその機能の説明です。

計算式	機能
Average Sum of Squares	統計分析において標準偏差の平方和の平均を計算します。
Average	フィールドの平均値を計算します。
Count	フィールドの件数を計算します。
Count Distinct	-REMOTE を使用する場合に、フィールド内で種類の個数を集計します。他の演算モードの場合、これは合計と同様に動作します。
Maximum	フィールドの最大値を出力します。
Minimum	フィールドの最小値を出力します。
Percent	フィールドの合計値に基づいて、そのフィールドのパーセントを計算します。パーセントは、[集計] (Summary) フィールド同様、[PRINT] にも使用できます。
Percent Count	あるフィールドのパーセンテージを、件数に基づいて計算します。

計算式	機能
Row	フィールドの行全体の合計値に基づいて、そのフィールドのパーセントを計算します。
Summarize	フィールドの件数を集計します。
Total	フィールドの個数を集計して、それを見出しに使用します (脚注、中間見出し、中間脚注にも使用可)。

データの制限

トピックス

日付要素への選択条件の適用

手順/構文

選択パネルから選択条件を適用するには

コントロールパネルから選択条件を適用するには

コントロールパネルで選択条件を変更するには

コントロールパネルから選択条件を削除するには

参照

選択条件の関係演算子

OLAP レポートでは、ディメンション階層の親カテゴリに属す値が制限されます。レポートに表示されるデータをさらに制限する方法がいくつかあります。

選択パネルまたはコントロールパネル ディメンション値および関係演算子 (=、>、< など) を選択して、OLAP レポートのデータを明示的に制限できます。関係演算子についての詳細は、46 ページの「[選択条件の関係演算子](#)」を参照してください。

- 選択パネルからは、最も簡単なアプローチができます。数回のマウスクリックでディメンション値と関連演算子を選択し、レポートを完全に表示できます。
- コントロールパネルからは、選択パネルでは使用できないいくつかのオプションを使用できます。オプションには、日付または日付範囲、特定の文字のある/なしに基づいたレコード選択があります。

選択パネルで行った変更は、コントロールパネル (閉じていても) にも即時に実装されることに注意してください。ただし、コントロールパネルで行った変更は、レポートが実行されるまで、選択パネルに反映されることはありません。







レポート 基軸とディメンションをドリルダウンして情報のサブセットを絞り込み、間接的にデータを制限できます。詳細は、29 ページの「[ディメンションと基軸のドリルダウン](#)」を参照してください。



参照

選択条件の関係演算子

いくつかの関係演算子を使用して、選択条件を定義できます。そのほとんどは選択パネルおよびコントロールパネルでサポートされています。下表は、コントロールパネルでのみサポートされている演算子の説明です。

左の列が演算子、右の列はその説明です。



演算子	表示されるレコード
 - 等しい (Equal to)	指定した条件と等しい。 デフォルトで選択される演算子です。
 - 等しくない (Not Equal to)	指定した条件と等しくない。
 - 以下 (Less than or equal to)	指定した条件以下。
 - より小さい (Less than)	指定した条件より小さく、かつ等しくない。
 - 以上 (Greater than or equal to)	指定した条件以上。
 - より大きい (Greater than)	指定した条件より大きく、かつ等しくない。

演算子	表示されるレコード
 - 文字列を含む (Contains)	指定した条件を含む。 注意： この演算子は文字フィールドでのみ使用可能であり、コントロールパネルでのみサポートされています。
 - 文字列を含まない (Not contain)	指定した条件を含まない。 注意： この演算子は文字フィールドでのみ使用可能であり、コントロールパネルでのみサポートされています。

注意：同じ関係演算子を使用すると、1 つ以上の値を選択できます。

次の関係演算子は、日付の範囲を選択することができます。これらはコントロールパネルからのみ使用できます。

下表の最初の列は関係演算子、2 番目の列はその説明を示しています。

演算子	表示するレコード
 - 範囲内 (Within range)	指定された日付フィールド内の値は、指定した範囲内にある。 注意： この関連演算子を使用するには、[日付選択] (Date Selection) パネルの [指定範囲] (Range) チェックボックスを選択する必要があります。
 - 範囲外 (Not within range)	指定された日付フィールド内の値は、指定した範囲内がない。 注意： この関連演算子を使用するには、[日付選択] (Date Selection) パネルの [指定範囲] (Range) チェックボックスを選択する必要があります。

手順

選択パネルから選択条件を適用するには

選択パネルがオンの場合、OLAP 階層にある各ディメンションに対してコントロール (ドロップダウンリスト) が 1 つあります。ディメンションフィールドの名前は、代わりに列タイトルが指定されている場合でも、マスターファイルで定義されたとおりに表示されることに注意してください。

レポートを含むディメンションのデータを制限する手順は次のとおりです。

1. ディメンションの右にある矢印をクリックして、値のリストを表示します。
2. リストから 1 つ以上の値を選択します (デフォルト値は [すべて] (All) です)。
複数の値を選択するには、キーボードで Ctrl キーを押しながら必要な値をクリックします。
3. ディメンションの左にあるボタンから関係演算子を選択し、選択の標準を示します。
「等しい (=)」がデフォルト値です。
演算子のリストを切り替えることができます。詳細は、46 ページの「[選択条件の関係演算子](#)」を参照してください。
4. 値を制限する必要があるディメンションごとに手順 1 から 3 を繰り返します。
5. 選択パネルの下の帯にある [実行] (Run) ボタンをクリックします。

ヒント：選択条件を変更または削除するには、値のリストを再度開いて他の値かまたは [すべて] (All) を選択します。

手順

コントロールパネルから選択条件を適用するには

1. コントロールパネルを開きます。
2. ウィンドウ右下の [選択条件] (Selection Criteria) ボタンをクリックします。
[選択条件] (Selection Criteria) パネルが開きます。
3. [選択条件] (Selection Criteria) パネル上の [ディメンション] (Dimensions) ボックスで、ディメンションを展開して [値] (Values) をクリックします。
2 番目のウィンドウが開きます。1 つ以上の値を選択します (複数選択には Ctrl キーを押します)。
4. [OK] をクリックして [選択条件] (Selection Criteria) パネルに戻ります。ここでは、ドロップダウンリストに選択した値が表示されます。
 - 開発者が OLAP レポートを作成したレポートオブジェクトに選択条件を適用している場合、選択したフィールドの受容可能な値のみが表示されます。
 - 選択条件が適用されていない場合、フィールドのすべての値がドロップダウンリストに表示されます。

5. [選択条件] (Selection Criteria) パネルでは、ディメンションの横にある関係演算子をクリックして、選択のベースとなる関係を指定します。たとえば、=、> または < を指定します。完全なリストは、46 ページの「[選択条件の関係演算子](#)」を参照してください。
6. 値を制限する別のディメンションでも同じ手順を繰り返します。
7. [実行] (Run) をクリックして、レポートを実行します。

手順

コントロールパネルで選択条件を変更するには

ヒント：選択パネルにアクセスする場合、選択条件を調整または削除する最も簡単な方法が使用されます。詳細は、47 ページの「[選択パネルから選択条件を適用するには](#)」を参照してください。

コントロールパネルの [選択条件] (Selection Criteria) パネルから次を実行します。

1. 変更するディメンション値の横にある [選択] (Select) ボタンをクリックします。

2 番目のウィンドウが開きます。

値を変更する テキストボックスに新しい値を入力するか、リストから 1 つ以上の値を選択します。この値はデータソースの値と同じ大文字小文字で入力する必要があります。

テキストボックスに入力できる値は 1 つのみです。リストから 1 つ以上の値を選択した場合、最初の値のみが表示されます。ただし、レポートにはすべての値が表示されます。

値の選択を解除 Ctrl キーを押したまま値をクリックします。


2. [OK] をクリックして [選択条件] (Selection Criteria) パネルに戻ります。ここで変更した値の確認や、必要に応じて関係演算子の変更が可能です。
3. 再度 [OK] をクリックして選択を確認し、メインの [コントロールパネル] (Control Panel) ウィンドウに戻ります。
4. [実行] (Run) をクリックして、レポートを実行します。

手順

コントロールパネルから選択条件を削除するには

ヒント：選択パネルにアクセスする場合、選択条件を調整または削除する最も簡単な方法が使用されます。詳細は、47 ページの「[選択パネルから選択条件を適用するには](#)」を参照してください。

コントロールパネルの [選択条件] (Selection Criteria) パネルから次を実行します。

1. 削除する条件を選択します。
2. [削除] (Delete)  ボタンを選択すると、[新規作成] (New)、[開く] (Open)、[保存] (Save)、[実行] (Run)、その他のオプションがポップアップウィンドウで表示されます。
選択したカテゴリがリストから削除されます。
3. [実行] (Run) をクリックし、すべての値を使用してレポートを実行します。

日付要素への選択条件の適用

手順/構文

日付フィールドに選択条件を適用するには

日付範囲に選択条件を適用するには

リストボックスに日付を追加するには

リストボックスから日付を削除するには

参照



日付フォーマットの制限

選択条件は、他のタイプの要素に適用するように、日付要素にも適用できます。結果は選択した日付で限定されます。たとえば、特定の日付に関連付けられたデータを表示したり、特定の日付のデータを除外したりすることができます。

注意：別のディメンション要素と同様に、日付フィールドは開発者がマスターファイルで定義しておく必要があります。マスターファイルでは、選択条件で使用できる日付フォーマットを指定します。

この機能は、コントロールパネルからのみサポートされています。ここでは、日付フォーマットの適切なコントロールを有する [日付選択] (Date Selection) パネルから選択条件を選択します。

また、[開始] (From) および [終了] (To) の日付を指定して、指定した年の日付範囲を選択できます。次の 2 つの関係演算子で、日付の範囲を選択することができます。

- [範囲内] (Within range)  演算子では、示された日付フィールドの値が指定された範囲内の場合にレコードが表示されます。
- [範囲外] (Not within range)  演算子では、示された日付フィールドの値が指定された範囲外の場合にレコードが表示されます。

詳細は、54 ページの「[日付フォーマットの制限](#)」を参照してください。

手順

日付フィールドに選択条件を適用するには

1. コントロールパネルを開きます。
2. [選択条件] (Selection Criteria) ボタンをクリックします。
[選択条件] (Selection Criteria) パネルが開きます。
3. [選択条件] (Selection Criteria) パネル上の [ディメンション] (Dimensions) ボックスで、日付フィールドのあるディメンションを展開し [値] (Values) ボタンをクリックします。

2 番目のウィンドウが開き、ディメンションの日付フォーマットのコントロールが表示されます。たとえば、日付フォーマットが **YYM** の場合、年と月のコントロールのみが表示されます。フォーマットが **YYMD** の場合、年、月および日のコントロールが表示されます。

注意：[日付選択] (Date Selection) パネルは、サポートされている日付フォーマットがある場合にのみ表示されます。詳細は、54 ページの「[日付フォーマットの制限](#)」を参照してください。
4. スピンコントロール、ドロップダウンリストを使用して、または値を入力して日付を指定します。

日付フォーマットに **Y.M.D** などのマスクが含まれている場合、日付はバックスラッシュ付きで [日付選択] (Date Selection) リストボックス、[選択条件] (Selection Criteria) パネルおよびレポート下部のドロップダウンリストに表示されます。ただし、日付は、レポート本文内で指定されたとおりに表示されます。
5. [追加] (Add) をクリックして [選択] (Selections) リストボックスに日付を表示します。
6. [OK] をクリックして [選択条件] (Selection Criteria) パネルに戻り、選択した日付を確認します。

7. [選択条件] (Selection Criteria) パネルで、日付フィールドの横にある関係ボタン (たとえば、=、> または <) をクリックして、レコード選択のベースを示します。
8. オプションとして、手順 2 から 7 を繰り返して追加の日付選択条件を定義します。
9. [実行] (Run) をクリックして、レポートを実行します。

手順

日付範囲に選択条件を適用するには

1. OLAP コントロールパネルを開きます。
2. [選択条件] (Selection Criteria) ボタンをクリックします。
[選択条件] (Selection Criteria) パネルが開きます。
3. [選択条件] (Selection Criteria) パネル上の [ディメンション] (Dimensions) ボックスで、日付フィールドのあるディメンションを展開し [値] (Values) ボタンをクリックします。



2 番目のウィンドウが開き、ディメンションの日付フォーマットのコントロールが表示されます。たとえば、日付フォーマットが YYM の場合、年と月のコントロールのみが表示されます。フォーマットが YYMD の場合、年、月および日のコントロールが表示されます。

注意：[日付選択] (Date Selection) パネルは、サポートされている日付フォーマットがある場合にのみ表示されます。詳細は、54 ページの「[日付フォーマットの制限](#)」を参照してください。

4. [指定範囲] (Range) チェックボックスをクリックします。
[含む] (Inclusive) および [含まない] (Exclusive) オプションボタンが表示されます。
 - [含む] (Inclusive) を選択すると、指定した日付を含む範囲が表示されます。
 - [含まない] (Exclusive) を選択すると、指定した日付を排除した範囲が表示されます。

注意

 - 一度に選択できる日付範囲は 1 つのみです。
 - 日付フォーマットに年が含まれている場合のみ、選択条件を日付範囲に適用できます。詳細は、54 ページの「[日付フォーマットの制限](#)」を参照してください。

[開始] (From) および [終了] (To) ドロップダウンリストに、選択可能なオプションがすべて表示されます。デフォルトでは現在の日付が表示されます。
5. スピンコントロールとドロップダウンリストを使用して、[開始] (From) の日付と [終了] (To) の日付を指定します。
6. [OK] をクリックして [選択条件] (Selection Criteria) パネルに戻ります。
7. 選択した日付範囲の [開始] (From) および [To] を両方表示するには、ドロップダウンリストの下矢印をクリックします。
8. [選択条件] (Selection Criteria) パネルの日付要素の左にある関係演算子をクリックして、以下を実行します。
 - [範囲内] (Within range)  演算子を選択して、値が指定された範囲内の場合にレコードを表示します。
 - [範囲外] (Not within range)  演算子を選択して、値が指定された範囲外の場合にレコードを表示します。
9. [実行] (Run) をクリックして、レポートを実行します。

手順

リストボックスに日付を追加するには

1. コントロールパネルを開きます。
2. [選択条件] (Selection Criteria) をクリックして、[選択条件] (Selection Criteria) パネルを開きます。
3. [選択] (Select) ボタンをクリックして、[日付選択] (Date Selection) パネルを開きます。

4. スピンボタン、ドロップダウンリストを使用して、または値を入力して追加する日付を指定します。
5. [追加] (Add) をクリックします。
リストボックス内に日付が表示されます。
6. [OK] をクリックして [選択条件] (Selection Criteria) パネルに戻ります。

手順

リストボックスから日付を削除するには

1. コントロールパネルを開きます。
2. [選択条件] (Selection Criteria) をクリックして、[選択条件] (Selection Criteria) パネルを開きます。
3. [選択] (Select) をクリックして、[日付選択] (Date Selection) パネルを開きます。
4. リストボックスから削除する日付を 1 つ以上選択します。
5. [削除] (Delete) をクリックします。
日付が削除されます。
6. [OK] をクリックして [選択条件] (Selection Criteria) パネルに戻ります。

参照

日付フォーマットの制限

日付要素に選択条件を適用する場合、次の制限があります。

- ❑ [日付選択] (Date Selection) パネルはユリウス暦の日付をサポートしていません。ただし、ユリウス暦の日付を使用する場合も、日付コントロールは表示されます。
- ❑ 日付のフォーマット (D、I2D、A2D) のみを持つ日付は、[日付選択] (Date Selection) パネルからサポートしています。代わりに、値のリストはデータソースから取得されます。
- ❑ 日付フォーマットに次のフォーマットの 1 つが含まれている場合、[日付選択] (Date Selection) パネルの [指定範囲] (Range) チェックボックスを使用できます。
 - ❑ DATE フォーマット - たとえば YMD、MDY、YYMD、MDYY、Q、M など。
 - ❑ A4YY
 - ❑ I4YY
 - ❑ I8YYMD

- A8YYMD
- I6YYM
- A6YYM

傾向の視覚化

手順/構文

数値基軸のピアグラフ列を追加するには

レポートをより強力にするため、選択したデータのグラフをレポート出力に直接挿入することができます。これらの視覚的表現は、数値データの隣に縦または横のピアグラフ列として表示され、データ間の関係や傾向をより明確にします。

次の場所から、選択した基軸にピアグラフを適用できます。

- レポート自体のコンテキストメニュー。
これは、数値基軸にピアグラフを適用する最も簡単な方法です。
- 選択パネルの [基軸] (Measures) コントロール。
- コントロールパネルの [基軸] (Measures) ボックスを選択します。

ピアグラフについての詳細は、「[vistrnd_new.xml#WSE14B1D54-627E-46e0-AF5C-B5354C2222E5](#)」を参照してください。

手順

数値基軸のピアグラフ列を追加するには

ピアグラフを最も簡単に適用するには、レポート自体から次の手順を実行します。

1. 基軸フィールドのタイトルを右クリックします。
2. メニューから [ピアグラフの表示] (Visualize) を選択します。

レポートが自動的に実行され、選択した基軸列の横にピアグラフの列が表示されます。

ヒント：ピアグラフを削除するには、基軸の列タイトルを右クリックし、メニューから [ピアグラフの非表示] (Remove Visualization) を選択します。

列にピアグラフを適用する別の方法についての詳細は、「[vistrnd_new.xml#WSE14B1D54-627E-46e0-AF5C-B5354C2222E5](#)」を参照してください。

グラフとレポートの表示

トピックス

OLAP グラフに追加されるタイトル

手順/構文

選択パネルから基軸をグラフ化するには

コントロールパネルから基軸をグラフ化するには

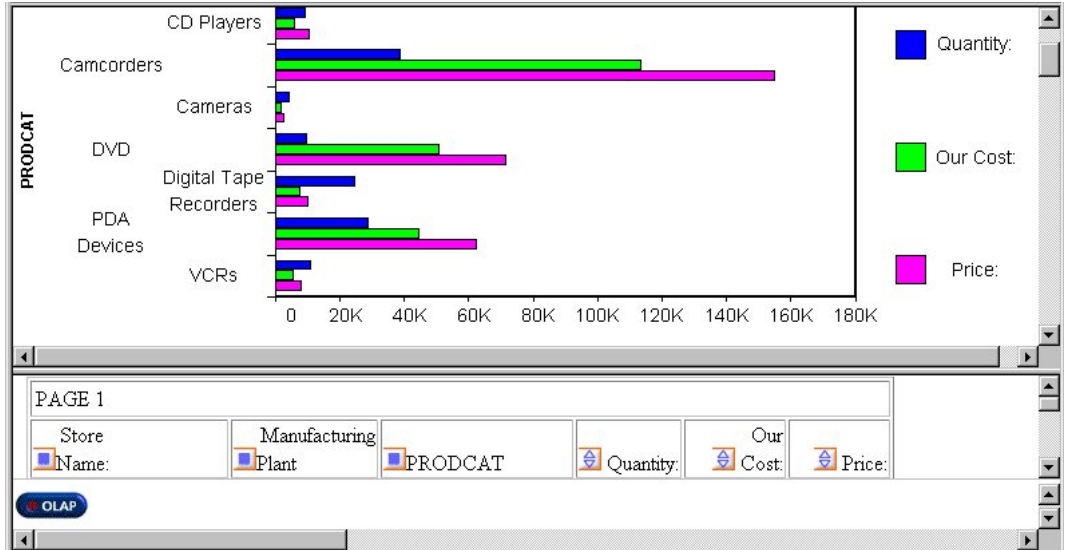
参照

OLAP グラフでのグラフスタイルと基軸スタイルの組み合わせ

OLAP レポートの基軸をグラフ化する場合、特定のデータ要素を選択して表形式のレポートに含めて表示します。同時に、同じ情報のグラフ表現がスプリットウィンドウにも表示されます。グラフは比較しやすいようにウィンドウの上半分のフレームに表示されます。

グラフ化を可能にするには、レポートのデータに最低 1 つの数値基軸と、1 つのソートフィールド (BY または ACROSS) を含める必要があります。これらの基本要件が満たされると、選択パネルまたはコントロールパネルのグラフコントロールが有効になります。

下図のように、ここでは 3 つのソートフィールド (Product Category、Store Name、Manufacturing Plant) と 3 つの数値基軸 (Quantity、Our Cost、Price) が含まれており、簡単に比較できるよう横の棒グラフとして表示されています。



グラフは OLAP レポート、選択パネルまたはコントロールパネルからリクエストできます。

- OLAP レポート** 選択した基軸のデータを表す縦のピアグラフを作成できます。
- 選択パネルまたはコントロールパネル** 次の 7 タイプの異なるグラフを作成し、1 つ以上の基軸に適用できます。
 - 縦の棒グラフ (デフォルト設定のグラフタイプです。)
 - 縦の線グラフ
 - 縦の面グラフ
 - 横の棒グラフ
 - 横の線グラフ
 - 横の面グラフ
 - 円グラフ

複数の基軸をグラフ化することを選択した場合、各列のデータに適合する異なるグラフタイプを適用できますが、次の制限があります。

- 基軸のコントロールグラフスタイルとして縦または横の棒、折れ線、面の各グラフを選択した場合、これらのスタイルのいずれかの組み合わせを、別の基軸に適用することができます。たとえば、最初の基軸は棒グラフとして、2番目の基軸は折れ線グラフとして、3番目の基軸は面グラフとして表示することができます。すべての基軸の方向は、同一（縦または横）にする必要があります。
- コントロールするグラフスタイルとして円グラフを選択した場合、別の基軸に使用できるのも円グラフのみになります。

サポートされている組み合わせについての詳細は、58 ページの「[OLAP グラフでのグラフスタイルと基軸スタイルの組み合わせ](#)」を参照してください。

注意：レポートのディメンションにドリルダウン機能を使用できる場合、同じ機能は自動的にグラフにも有効になります。結果として、データの 1 つのグラフ表現から別のグラフ表現へドリルダウンできます。

OLAP グラフに追加されるタイトル

複数のソートフィールドが設定された OLAP で、グラフ機能を使用すると、複数のグラフが生成されます。グラフのそれぞれにタイトルが追加されます。このタイトルは、外側のソートフィールドの値と等しくなります。

参照

[OLAP グラフでのグラフスタイルと基軸スタイルの組み合わせ](#)

下表は、各グラフスタイルと、使用可能な組み合わせを示しています。

グラフスタイルのコントロール	利用可能な基軸スタイル
縦の棒グラフ (デフォルト設定)	縦の棒グラフ (デフォルト設定)
	縦の線グラフ
	縦の面グラフ

グラフスタイルのコントロール	利用可能な基軸スタイル
縦の線グラフ	縦の線グラフ (デフォルト設定)
	縦の棒グラフ
	縦の面グラフ
縦の面グラフ	縦の面グラフ (デフォルト設定)
	縦の棒グラフ
	縦の線グラフ
横の棒グラフ	横の棒グラフ (デフォルト設定)
	横の線グラフ
	横の面グラフ
横の線グラフ	横の線グラフ (デフォルト設定)
	横の棒グラフ
	横の面グラフ
横の面グラフ	横の面グラフ (デフォルト設定)
	横の線グラフ
	横の面グラフ
円グラフ	円グラフ

手順

選択パネルから基軸をグラフ化するには

1. グラフコントロールの左にある下矢印をクリックして、現在のレポートにあるすべての数値基軸を持つパネルを開きます。

各基軸の左にチェックボックス、右にグラフボタンがあります。すべてのチェックボックスはデフォルト設定でオフになっており、すべてのグラフボタンはデフォルト設定で灰色で表示されています (無効)。

2. 基軸と関連したチェックボックスを選択します。

基軸の右にあるグラフボタンが有効になります。デフォルトのグラフスタイルは、縦のピアグラフです。

3. 選択した基軸に適用するスタイルに到達するまで、7つのグラフスタイルアイコンを切り替えます。

4. グラフ化する他のすべての基軸に対して、手順 2 と手順 3 を繰り返します。

定義できるグラフタイプのリストについては、58 ページの「[OLAP グラフでのグラフスタイルと基軸スタイルの組み合わせ](#)」を参照してください。

5. 選択パネルの下の帯にある [実行] (Run) ボタンをクリックします。

レポートと選択パネルの上のフレームにグラフが表示されます。

手順

コントロールパネルから基軸をグラフ化するには

1. コントロールパネルを開きます。

2. [基軸] (Measures Properties) ボックスの下にある [グラフ表示] (Show Graph) チェックボックスを選択します。

[ドリルダウン] (Drill Down) および [ドリルアクロス] (Drill Across) ボックスの内容によって、X軸フィールドが異なることに注意してください。複数のドリル(X軸)フィールドがある場合、複数のグラフが縦に積み重ねられ、同じフレームに表示されます。基軸はグラフ上のY軸フィールドとして表示されます。

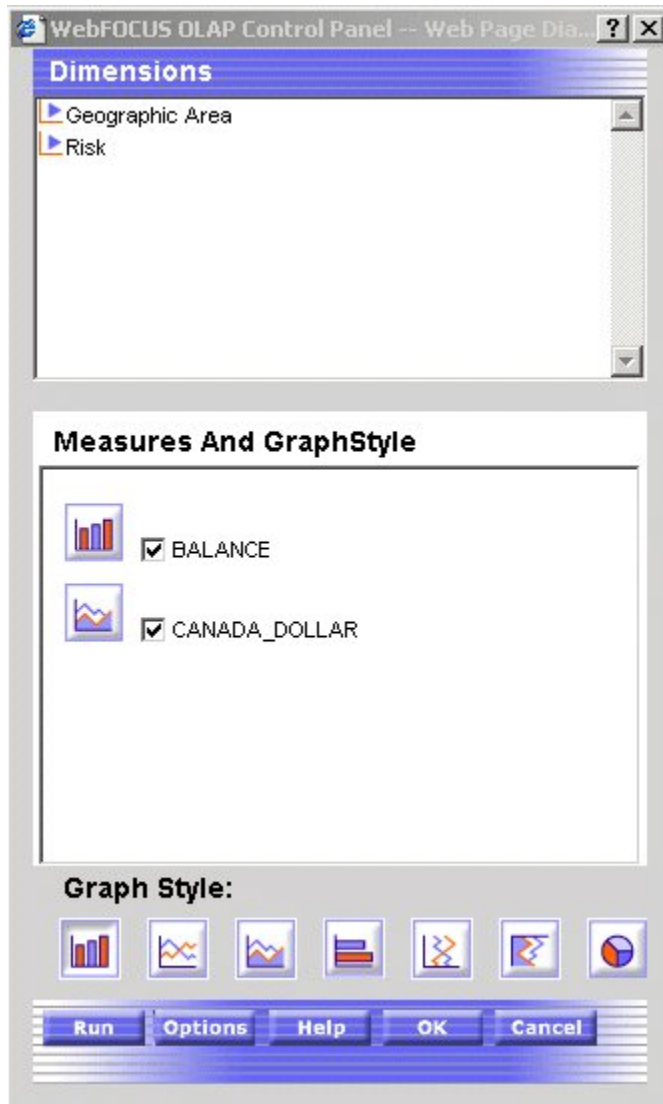
3. [グラフ表示] (Show Graph) チェックボックスと隣接する [グラフ] (Graph) アイコンをクリックします。

[基軸] (Measures) および [グラフスタイル] (GraphStyle) パネルが開きます。

利用可能な基軸と関連したチェックボックスは、デフォルトで選択が解除されています。

4. ウィンドウ下部にある 7つのアイコンの 1つをクリックして、コントロールするグラフタイプを設定します。

5. グラフ化する基軸のチェックボックスを選択します。複数の選択も可能です。
選択した各基軸の横に、コントロールするグラフスタイルに応じたグラフアイコンが表示されます。
6. 下図のように、基軸の横のアイコンをクリックして、サポートされている組み合わせから異なるグラフスタイルを選択します。



7. [OK] をクリックして、メインの [コントロールパネル] (Control Panel) ウィンドウに戻ります。すべてのグラフ設定が保持されたままです。
8. [実行] (Run) をクリックして、スプリットウィンドウにグラフと表形式のレポートを表示します。

注意

- [グラフ表示] (Show Graph) チェックボックスを選択し、コントロールするグラフスタイルを選択せずに [実行] (Run) をクリックした場合、デフォルトのスタイル (縦のピアグラフ) が適用されます。
- [グラフ表示] (Show Graph) チェックボックスを選択せずに [実行] (Run) をクリックした場合、グラフなしで、表形式のレポートのみが表示されます。
- [基軸] (Measures) および [グラフスタイル] (GraphStyles) パネルで最低 1 つの基軸を選択し、[グラフ表示] (Show Graph) チェックボックスを選択しなかった場合、[OK] をクリックすると、システムによって自動的に [グラフ表示] (Show Graph) チェックボックスが選択されます。表形式のレポートがグラフ付きで表示されます。
- 文字またはデータフィールドのグラフ化は選択できません。数値基軸がない場合、[グラフ表示] (Show Graph) チェックボックスおよび [グラフ] (Graph) ボタンは使用できません (灰色で表示されます)。

レポートでの基軸の表示のコントロール

トピックス

基軸の積み重ね

基軸の列順変更

基軸の非表示と表示

OLAP レポートでは、元のレポートリクエストに戻らずには新しい基軸を追加できませんが、レポート内での基軸の表示は複数の方法で調整できます。次のことが可能です。

- 基軸を行に積み重ねる。
- 基軸列の順序を変更する。
- 基軸の表示と非表示を切り替える。
- 任意の数値基軸の後にピアグラフ列を追加する。

基軸の積み重ね

手順/構文

積み重ねた基軸を表示するには

OLAP レポートに複数の基軸がある場合、基軸を同じ列内の別の行に積み重ねて、レポートの幅を小さくすることができます。

積み重ねた基軸には、ピアグラフを適用できません。

手順

積み重ねた基軸を表示するには

1. コントロールパネルを開きます。
2. [積み重ね基軸] (Stack Measures) のチェックをオンにして、1 列下の異なる行に基軸を表示します。
3. [実行] (Run) をクリックして、レポートを実行します。

ヒント：標準の表示をリストアするには、[積み重ね基軸] (Stack Measures) チェックボックスの選択を解除して、レポートを再度実行します。

基軸の列順変更

手順/構文

OLAP レポートの基軸列を再配置するには

レポート内で、基軸列の表示順序を変更することができます。

手順

OLAP レポートの基軸列を再配置するには

数値列を再配置するには、フィールドを新しい列の位置にドラッグアンドドロップします。

フィールドをドロップできる位置では、カーソルがプラス記号 (+) に変わります。受け入れられない位置では、内部に斜線が引かれた円の形で示されます。

基軸の非表示と表示

手順/構文

レポートから基軸を非表示にするまたは表示するには
選択パネルから基軸を非表示にするまたは表示するには
コントロールパネルから基軸を表示するまたは非表示にするには

OLAP レポート、選択パネルまたはコントロールパネルから基軸を非表示または表示にできません。

手順

レポートから基軸を非表示にするまたは表示するには

基軸列を非表示にする 列タイトルを右クリックし、メニューから [非表示] (Hide) を選択します。列は自動的に表示から除かれます。

基軸列を表示する 列タイトルを右クリックし、メニューから [表示] (Unhide) を選択します。非表示の基軸は 2 番目のメニューにすべて表示されます。

レポートに再度表示する基軸を選択します。

ヒント: このレポートに新しい基軸を追加する場合、元のリクエストに戻り、そこでフィールドを追加する必要があります。

手順

選択パネルから基軸を非表示にするまたは表示するには

1. [基軸] (Measures) コントロールの左にある下矢印をクリックして、レポート内の基軸のリストを表示します。
2. 基軸の横のチェックボックスをクリックして、それを表示または非表示にします。チェックボックスでは、3 つの位置を切り替えられます。
 - **基軸を非表示にする** ブランクになるまでチェックボックスをクリックします。
 - **非表示の基軸を表示する** チェックマークが出るまでチェックボックスをクリックします。

ヒント: 同じチェックボックスを使用して、数値基軸のピアグラフの列を表示できます。この設定は、チェックボックスにグラフとして現れます。詳細は、55 ページの「[傾向の視覚化](#)」を参照してください。

手順 **コントロールパネルから基軸を表示するまたは非表示にするには**

1. コントロールパネルを開きます。
2. [基軸](Measures) ボックスで、基軸の横のチェックボックスをクリックして、それを表示または非表示にします。チェックボックスでは、3つの位置を切り替えられます。
 - **基軸を非表示にする** ブランクになるまでチェックボックスをクリックします。
 - **非表示の基軸を表示する** チェックマークが出るまでチェックボックスをクリックします。

ヒント：同じチェックボックスを使用して、数値基軸のピアグラフの列を表示できます。この設定は、チェックボックスにグラフとして現れます。詳細は、55 ページの「[傾向の視覚化](#)」を参照してください。
3. [実行](Run) をクリックして、レポートを実行します。

ディメンションの追加と削除

手順/構文

コントロールパネルからディメンション要素を追加するには
 レポートからディメンション要素を削除するには
 コントロールパネルからディメンション要素を削除するには

ディメンション階層にあるすべての値は Developer Workbench の OLAP レポートで使用できるため、元のレポートリクエストに戻らずに、いつでもディメンションを OLAP レポートに追加することができます。ディメンションは次の場所から追加できます。

- OLAP レポート
- コントロールパネル

手順 **コントロールパネルからディメンション要素を追加するには**

1. コントロールパネルを開きます。
2. レポートのレイアウトボックス (ドリルダウンまたはドリルアクロス) を選択し、レポート内で使用する新しいソート方向を指定します。

3. ウィンドウ上部にある [ディメンション] (Dimensions) ボックスでディメンションを展開し、指定したレイアウトボックスに追加するディメンション要素をクリックします。リストの最後に新しいディメンションが追加されます。
4. 新しいソートフィールドの位置を変更する場合は、上向き矢印をクリックして再配置します。
5. [実行] (Run) をクリックして、新しい設定でレポートを実行します。

手順


レポートからディメンション要素を削除するには

削除するディメンション列を右クリックして、メニューから [削除] (Delete) を選択します。

レポートは自動的に実行されます。

手順

コントロールパネルからディメンション要素を削除するには

1. [ドリルダウン] (Drill Down) または [ドリルアクロス] (Drill Across) ボックスから要素を選択します。ボックス上のボタンが有効になります。
2. [削除] (Remove)  をクリックします。[ドリルダウン] (Drill Down) または [ドリルアクロス] (Drill Across) ボックスから要素が削除されます。
3. [実行] (Run) をクリックして新しいレポートを表示します。

OLAP レポートの保存

トピックス

オプションの保存

OLAP で参照する統一されたフィールド名

DB2 Web Query には、OLAP レポートの保存機能がいくつか用意されています。

- レポートを保存するダイアログでは、[マイレポート] (Reports) の下のユーザの選択したサブフォルダに保存できます。サブフォルダは、OLAP ツールからも作成できます。
- ユーザはレポートを OLAP 機能を削除して保存できますが、そうすると OLAP ツールを単純なレポート作成ツールとして使用できます。

- 参照されるフィールド名は、OLAP 製品全般に共通です。たとえば、AS または TITLE 句は、OLAP 選択パネルまたは OLAP コントロールパネル (OCP) を使用して生成されたレポートに表示されます。

注意：OLAP レポートを保存する場合、新しく保存したレポートを表示するには、ドメインのリフレッシュを実行する必要があります。リフレッシュを実行しないと、新しいレポートはドメインツリーに表示されません。

オプションの保存

手順/構文

OLAP のカスタムレポートを作成するには

OLAP 対応のレポートを OLAP 機能のないプレーンなレポートとして保存するには

OLAP 選択パネルで作業する場合、ツールバーの [保存] (Save) ボタンを使用してレポートを保存できます。OCP では [オプション] (Options) ボタンを選択して、他の保存オプションにもアクセスできます。

手順

OLAP のカスタムレポートを作成するには

1. OLAP 選択パネルで [保存] (Save) ボタンをクリックします。

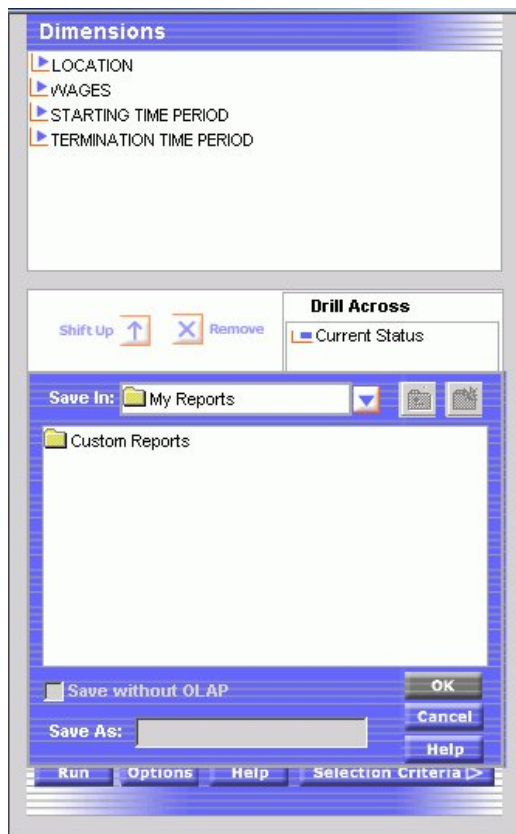
下図に示すように、5 つのボタンを持つ OLAP 選択パネルが表示されます。[OLAP]、[実行] (Run)、[リセット] (Reset)、[保存] (Save) および [ヘルプ] (Help) ボタンがあります。



または

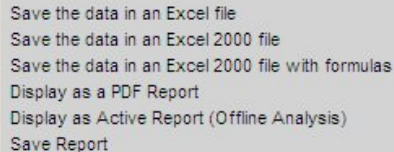
OLAP コントロールパネル (OCP) の [オプション] (Options) ボタンをクリックします。

下図に示すように、OLAP コントロールパネルでは、パネルの下部に [オプション] (Options) ボタンが表示されます。



ここまでいくつかの保存オプションについて説明しました。

下図のように、[オプション](Options)メニューには、[Excelファイルでデータを保存](Save the data in an Excel file)、[Excel 2000ファイルでデータを保存](Save the data in an Excel 2000 file)、[式とともにExcel 2000ファイルでデータを保存](Save the data in an Excel 2000 file with formulas)、[PDFレポートで表示](Display as a PDF Report)、[active report形式で表示(オフライン分析)](Display as active report (Offline Analysis))、[レポートの保存](Save Report)の6つのオプションがあります。

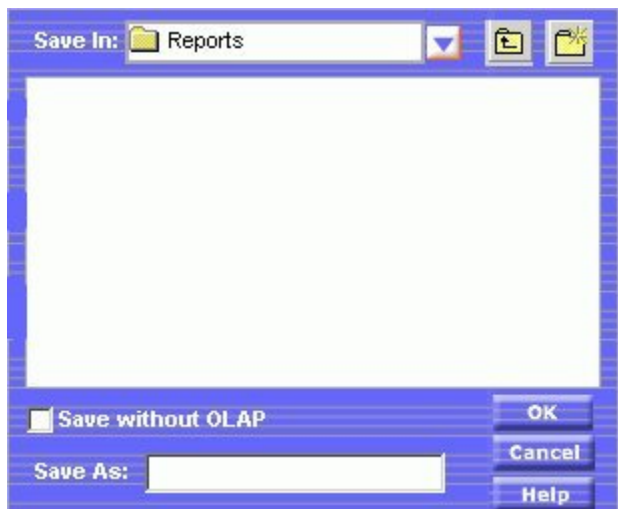


Save the data in an Excel file
Save the data in an Excel 2000 file
Save the data in an Excel 2000 file with formulas
Display as a PDF Report
Display as Active Report (Offline Analysis)
Save Report

2. [レポートを保存](Save Report)をクリックすると、下図のように、新しいダイアログボックスが開きます。



3. [レポート] (Reports) をクリックすると、下図のように、右上の [新規フォルダ] (New Folder) アイコンが使用可能になります。



4. コンテキストメニューが開き、新しいフォルダの作成が要求されます。下図の例のように「OLAP Reports」という名前のサブフォルダを作成します。



5. [名前を付けて保存] (Save As) ボックスでレポートに名前を付け、[OK] をクリックしてレポートに戻ります。

手順

OLAP 対応のレポートを OLAP 機能のないプレーンなレポートとして保存するには

ユーザはレポートを OLAP 機能を削除して保存できますが、そうすると OLAP ツールを単純なレポート作成ツールとして使用できます。

1. 67 ページの「[OLAP のカスタムレポートを作成するには](#)」の手順に従い、サブフォルダを作成します。
2. サブフォルダを選択した後、[OLAP 以外を保存] (Save without OLAP) ボックスを選択します。
3. [名前を付けて保存] (Save As) ボックスにレポートの新しい名前を入力し、[OK] をクリックします。

OLAP 機能が除外された、新しいレポートが生成されます。下図は、従業員の給与レベルを示す通常のレポートです。従業員の姓でソートされます。

Last Name	Pay,Level							
	1	2	3	4	5	6	7	8
ADAMKIEWICZ	5	.	.	.
ADAMS	6	.	.
ALBOR	.	2
ALIBUDBUD	7	.
ALSMAN	5	.	.	.
AMBERSON	.	.	.	4
ANDERSON	5	.	.	.
ARNOLD	.	2
AUSTIN	.	2

OLAP で参照する統一されたフィールド名

フィールドの参照に関する、開発者によるレポート設計が OCP および OLAP 選択パネルの両方に適用されます。フィールドの参照は、レポートと OCP および OLAP 選択パネルで違いはありません。AS、TITLE またはフィールド名によるフィールド参照は、レポート出力および OLAP コントロールで共通です。

OLAP レポートとグラフの他のフォーマットでの保存と表示

手順/構文

OLAP レポートとグラフを PDF フォーマットで表示するには

OLAP レポートとグラフを Excel ファイルとして保存するには

OLAP レポートとグラフをレポートフォルダに保存するには

OLAP レポートおよびグラフは、使用しているブラウザに HTML フォーマットで表示されます。レポートと対応するグラフを PDF および Excel フォーマットで表示することができます。

下図のようなポップアップメニューに、保存や表示に利用可能なフォーマットのリストが表示されます。

The screenshot displays the WebFOCUS OLAP Control Panel interface. The main area shows a data table with the following content:

Product Name:	Quantity:	Line Cost Of Goods Sold
ZT Digital PDA - Commercial	12,433	4,339,117.00
ZC Digital PDA - Standard	4,327	1,077,423.00

On the right side, a context menu is open, listing the following options:

- Save the data in an Excel file
- Save the data in an Excel 2000 file
- Display as a PDF Report
- Save the Report

- PDF (Portable Document Format) は、ユーザのブラウザやプリンタのタイプに関わらず、レポートやグラフの外観やレイアウトを保持する場合に役立ちます。

PDF フォーマットを選択すると、レポートは Adobe Reader で表示され、グラフはそのままブラウザのウィンドウ上に表示されます。Adobe Reader から印刷する場合、レポートのみが印刷されます。
- Excel は、大容量のデータベースをスプレッドシートに変換したり、レポートやグラフを一般に使用されている Office ツールに保存する場合に役立ちます。Excel フォーマットは次の 2 つを使用できます。
- Excel 2000 では、ほとんどのスタイルシートの属性をサポートしているので、完全なレポートのフォーマット設定が可能です。レポートを表示するコンピュータには、Microsoft Excel 2000 以降がインストールされている必要があります。

Excel 2000 を選択した場合、レポートとグラフは、Excel オプションを使用してデータを操作できる、同じツールで表示されます。Excel からはレポートとグラフの両方を印刷できます。

Excel 2000 フォーマットで保存した場合、明示的なドリルダウン (ベースレポートからドリルダウンレポートに渡されるパラメータに基づく) のみが動作を継続します。ディメンションと基軸のオートドリルダウンは、Excel ではサポートされていません。
- Excel フォーマットは、限定されたフォーマットをサポートするバイナリ表示フォーマットです。レポートを表示するコンピュータに、Microsoft Excel がインストールされている必要があります。

ドリルダウンは、いずれの種類もサポートされていません。

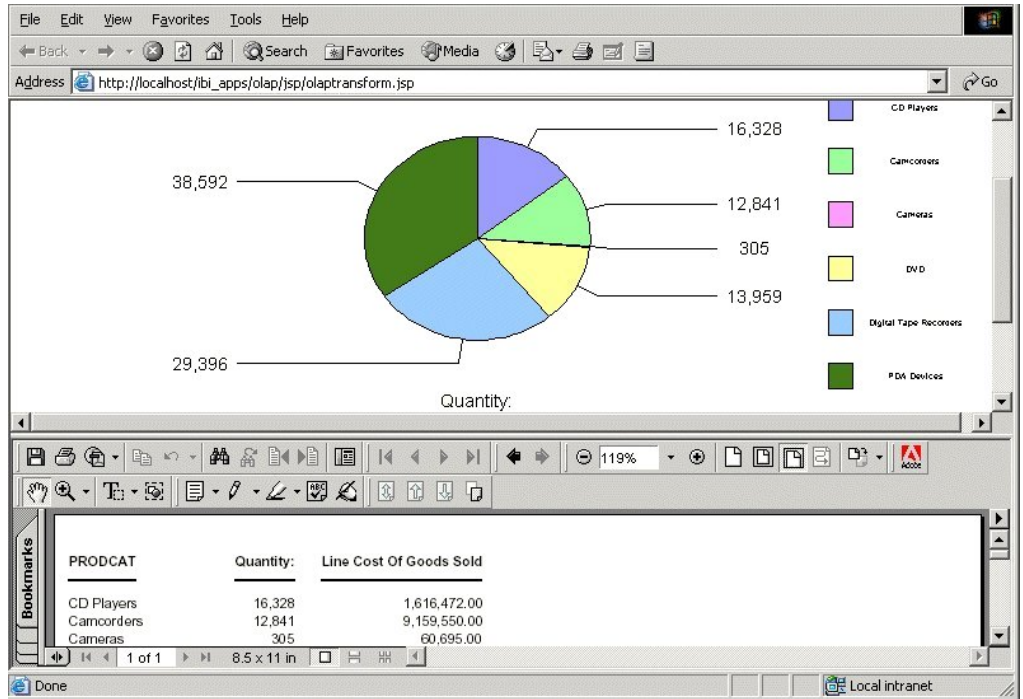
ユーザは [レポート] (Report) フォルダに HTML 出力を保存することができます。

手順

OLAP レポートとグラフを PDF フォーマットで表示するには

1. コントロールパネルを開きます。
2. ウィンドウ下部の [オプション] (Options) ボタンをクリックします。
3. [PDF レポートで表示] (Display as a PDF Report) を選択します。

下図のように、グラフはレポートの上のブラウザに表示されます。2 番目のブラウザが開き、Adobe Reader を起動してレポート出力を表示します。



ヒント：必要な場合は、Adobe Reader から PDF レポートを保存し、印刷することができます。

手順

OLAP レポートとグラフを Excel ファイルとして保存するには

1. コントロールパネルを開きます。
2. ウィンドウ下部の [オプション] (Options) ボタンをクリックします。
3. [Excel ファイルでデータを保存] (Save the data in an Excel file) または [Excel 2000 ファイルでデータを保存] (Save the data in an Excel 2000 file) を選択します。
4. 指示に従って、データをエクスポートします。

手順

OLAP レポートとグラフをレポートフォルダに保存するには

1. コントロールパネルを開きます。

2. ウィンドウ下部の [オプション] (Options) ボタンをクリックします。
3. [マイレポートとして保存] (Save as) を選択します。
2 番目のウィンドウが開きます。
4. 説明的な名前を入力し、[OK] をクリックしてグラフと表形式のレポートを保存します。

注意：グラフ凡例のラベルに文字数の制限はありませんが、長いラベルは切り取られる場合があります。

2 データ記述とアクセスの概要

DB2 Web Query および Developer Workbench が提供するグラフィカル ツールセットを使用して、次のようなさまざまなタイプのデータソースを記述し、それらにアクセスすることができます。

- リレーショナル (例、DB2 および Microsoft SQL Server)
- プロシジャ (Query/400)
- 複数メンバーファイルフォーマット (複数の DB Heritage File)
- ERP (JDE World および JDE EnterpriseOne)

これらのグラフィカルツールを使用して、次の作業が行えます。

- DB2 Web Query による読み込みおよびレポートの作成が可能なメタデータに、データソーススキーマを変換する。
- オプション製品の Developer Workbench を使用して、生成後のメタデータを拡張する。その際、DB2 Web Query データ記述言語の知識は必要ありません。

トピックス

- データソースの用語について
- アプリケーションのデータ解析方法
- データアダプタによるシノニムの作成
- シノニムの使用
- シノニムの拡張
- アプリケーションによるシノニムの使用法

データソースの用語について

データソースのタイプにより、同じ概念に対して異なる用語が使用されます。たとえば、データの有意な最小要素は、階層データベース管理システムおよびインデックスデータアクセスメソッドの多くでは「フィールド」と呼ばれますが、リレーショナルデータベース管理システムでは「カラム」(または「列」)と呼ばれます。

このほかにも、異なる用語で識別される一般的な概念が多数存在します。このため、このマニュアルでは説明を簡素化するために用語を標準化しています。たとえば、データソースのタイプに関わらず、データの有意な最小要素は「フィールド」と呼ばれます。ただし、明確にする必要がある場合は、データソースに特化した用語が使用されます。新しい標準用語が使用される際、その用語の定義と異なるデータソースタイプでの名称を記述します。

アプリケーションのデータ解析方法

アプリケーションがデータソースにアクセスする場合、アクセス先のデータをアプリケーション自体が解釈する必要があります。データを解釈するためにアプリケーションはを読み取ります。シノニムとは、特定のデータソースに関連付けられた生成済みメタデータを示す用語です。

アプリケーションについて、以下の知識が必要です。

- データの全体的構造。たとえば、リレーショナルデータ、階層データ、マルチディメンションデータ、シーケンシャルデータなどのデータのタイプ。構造による違い(配列およびインデックス)など。
- 特定のデータ要素。たとえば、データソースに格納されているフィールド名や各フィールドのデータタイプ(例、文字、日付、整数)。

シノニムは、データソースのエイリアスをサーバに提供して、テーブルの記述方法および検索方法を指示します。

シノニムの主要コンポーネントはマスターファイルです。マスターファイルは、データソースの構造とそのフィールド(例、マスターファイルにはフィールド名やデータタイプなどの情報)が含まれます。

データソースのタイプによっては、アクセスファイルがマスターファイルを補足します。アクセスファイルには、データソース記述を補足する追加情報が記述されています。これには、データソースのフルネームやフルパス名などがあります。アクセスファイルの情報の性質は、データソースによって異なります。データソースを記述するには、マスターファイルが1つと、データソースによってはアクセスファイルが1つ必要です。

データアダプタによるシノニムの作成

手順/構文

ウィザードを使用せずにアダプタを構成するには

参照

アダプタの動作

シノニムを生成するには、DB2 Web Query の [メタデータ] (Metadata) メニューのいずれかを使用します。これにより、DBMS カタログを参照し、シノニムを作成するオブジェクトを選択することができます。ツールは特定のデータソースのシノニムの作成に必要な情報の指定を要求し、生成されたシノニムをサーバに格納します。

レポートを作成するには、データの使用権限が必要です。また、アクセスするデータタイプに対してアダプタが構成されている必要があります。シノニム作成の開始時に、Developer Workbench では、[構成するアダプタを選択するか、シノニムを作成する接続を選択] (Select adapter to configure or Select connection to create synonym) ウィンドウが開きます。シノニムを作成するオプションは、アダプタが正しく構成された後に有効になります。

アダプタは多くのデータソースで使用可能です。各アダプタの構成はアクセスするデータソースに対して特有であるため、SQL または DB2 Web Query とデータソースのデータ管理言語 (DML) 間の変換が可能になります。アダプタは、製品の多様性に対するソリューションを提供します。製品の多様性には、構文、機能、スキーマ、データタイプ、カタログ、データ表現、メッセージの処理、アンサセットの取得においての相違が含まれます。シノニムの作成プロセスの管理は、アダプタにより行われます。詳細は、79 ページの「[アダプタの動作](#)」を参照してください。

参照

アダプタの動作

アダプタは、データインターフェースとデータソース間の通信を管理します。たとえば、データ管理リクエストをデータソースに送信し、アンサセットまたはメッセージをリクエストの発信者に返します。これらの機能を実行するため、アダプタは次の処理を行います。

- リクエストを該当する DML に変換する。
- 標準接続コールを使用して、ターゲットデータソースに接続する。
- 処理を担当するデータソースに、リクエストを送信する。

- 以降の処理がクライアントで行えるよう、結果またはエラーの状態をクライアントアプリケーションに返送する。

手順

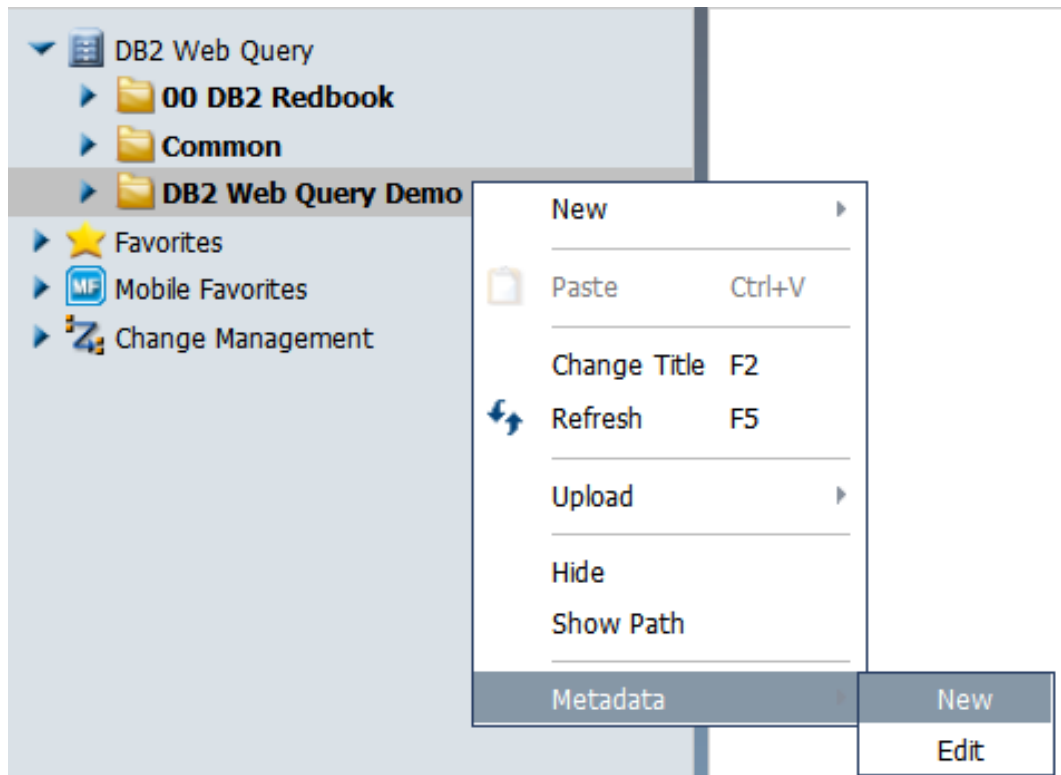
ウィザードを使用せずにアダプタを構成するには

1. Web Query 管理者 (WebQueryAdministrator グループのメンバーであるユーザ ID) または DBA 権限を所有するユーザ (最上位フォルダの folder-dba グループのメンバーである任意のユーザ ID) として DB2 Web Query にログインします。

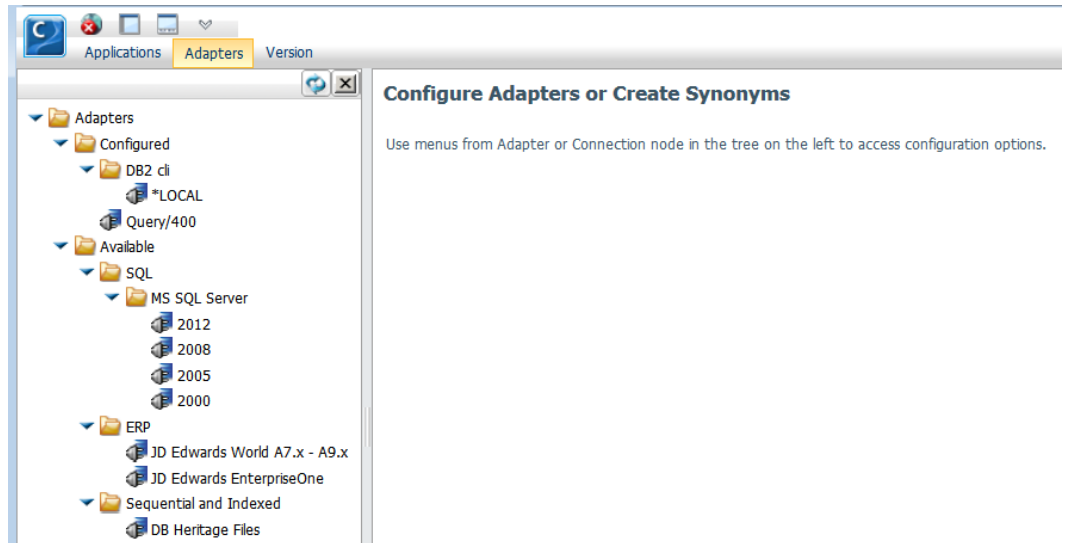
これらのタイプのユーザは、アダプタの管理および構成を行えますが、それ以外のユーザには、アダプタの管理と構成は許可されていません。

2. 任意のフォルダを右クリックし、[メタデータ] (Metadata)、[編集] (Edit) を選択します。

下図は、Reporting Server で [アダプタの構成] (Configuring Adapter) ウィンドウを開く手順を示しています。



3. [利用可能](Available) フォルダを展開し、利用可能なアダプタのリストを表示します。
 下図は、Reporting Server で展開された [利用可能](Available) フォルダを示しています。



4. 構成するアダプタを選択し、接続パラメータ値を入力します。
 5. Reporting Server のヘルプを参照し、選択したアダプタを構成します。

シノニムの使用

シノニムの生成後、DB2 Web Query のレポートツールのすべてを使用してレポートを作成することができます。多くの場合、データへのアクセスおよびレポートとグラフの作成には、構成済みアダプタと生成済みシノニムのみが必要です。

ただし、DB2 Web Query のデータ記述言語でサポートされている特定の機能を実装するためにシノニムの拡張が必要な場合があります。その場合、オプション製品の Developer Workbench が必要です。この製品にはシノニムエディタが含まれています。

シノニムエディタを使用する場合、グラフィカルツールによりシノニムコンポーネントの表示および編集可能な属性がすべて表示されるため、データ記述言語の知識は必要ありません。生成済みシノニムに変更を行うと、シノニムエディタは変更内容を検証し、データ記述言語の構文に違反している場合はメッセージを表示します。

シノニムの拡張

ここでは、データアクセスおよびレポート機能を拡張するために追加することができる属性を、いくつか紹介します。次のことが可能です。

- 使用可能なシノニムを結合し、複数セグメント (複数テーブル) ファイルを作成することで、クラスタ JOIN ビューを作成する。
- OLAP 解析用のディメンションを追加する。
- 一時項目 (DEFINE) および集計フィールド (一時項目 (COMPUTE)) を追加する。
- フィルタを追加し、データ選択条件を指定する。
- グループをサポートするデータソースにグループ定義を追加する。
- 多言語のバリエーションを含む、分かりやすいタイトルや説明を追加する。
- 文字フィールドのサイズ、日付フィールドのフォーマットなど、フィールドのフォーマットを変更する。
- ビジネスビューを参照する任意のデータ検索リクエストが利用できるフィールドを制限し、アプリケーションでの役割に基づいてフィールドをグループ化するためにメタデータのビジネスビューを作成する。
- キューブデータソースの親子階層を定義する。

シノニムエディタについての詳細は、83 ページの「[シノニムエディタの使用](#)」を参照してください。

アプリケーションによるシノニムの使用法

シノニムは、関連付けられたデータソースとは別に格納されます。アプリケーションはシノニムを使用し、次の手順に従って、データソースを解析します。

- 1.** リクエストに指定されたデータソースのマスターファイルを識別して特定した後、読み込みます。
- 2.** アクセスファイルが必要なデータソースタイプである場合、リクエストに指定されたデータソースをアクセスファイルで検索し、読み込みます。
- 3.** データソースを検索し、読み込みます。

データソースのコンテンツは、シノニム内の情報に基づいて解析されます。

3 | シノニムエディタの使用

シノニムは、データソースを記述する一連の属性で構成されています。シノニムはデータソースのメタデータを提供し、この情報によって、アダプタは対応するデータにアクセスし、そのデータを処理します。シノニムエディタは、シノニムの作成、表示、変更などの操作が可能なグラフィカルインターフェースを提供します。

注意：DB2 Web Query V2R1M0 では、ブラウザベースのシノニムエディタが追加されています。ブラウザベースバージョンのシノニムエディタについての詳細は、『新機能ガイド』を参照してください。この章では、DB2 Web Query Developer Workbench からアクセス可能なバージョンについて説明しています。

トピックス

- ❑ シノニムエディタの画面構成
- ❑ シノニム属性の表示および編集
- ❑ 多言語のタイトルおよび説明の設定
- ❑ モデル表示を使用したシノニムの拡張
- ❑ データプロファイリング特性の表示
- ❑ クラスタ JOIN の作成
- ❑ OLAP 分析のディメンション定義
- ❑ ビジネスビューの作成
- ❑ シノニムへの一時項目 (DEFINE) の追加
- ❑ シノニムフィルタの作成
- ❑ シノニムへの一時項目 (COMPUTE) の追加
- ❑ 一時項目内 (DEFINE) に繰り返しフィールド数を格納
- ❑ カスタムフィールドの属性および式の定義
- ❑ シノニムへのグループフィールドの追加
- ❑ データベース管理者セキュリティの使用

シノニムエディタの画面構成

参照

シノニムエディタの主要属性

シノニムエディタツールバー

シノニムエディタのオプション設定

シノニムエディタのフォーマットオプション設定

シノニムエディタのカラム管理設定

シノニムエディタのトレースオプション設定

シノニムエディタの実行オプション設定

シノニムエディタ - セグメントコンテキストメニュー

シノニムエディタ - カラムおよびフィールドショートカットメニュー

シノニムエディタ - フィールド表示タブ

シノニムエディタ - セグメント表示タブ

シノニムエディタ - リスト表示タブ

シノニムエディタ - モデル表示タブ

シノニムエディタ - テキスト表示タブ

シノニムエディタ - アクセスファイルテキスト表示タブ

シノニムエディタは、DB2 Web Query Developer Workbench から使用することができます。シノニムエディタには、[フィールド表示] (Field View)、[セグメント表示] (Segment View)、[リスト表示] (List View)、[モデル表示] (Modeling View)、[テキスト表示] (Text View)、[アクセスファイルテキスト表示] (Access File Text View) のタブがあり、これらを使用して、シノニムを管理できます。

参照

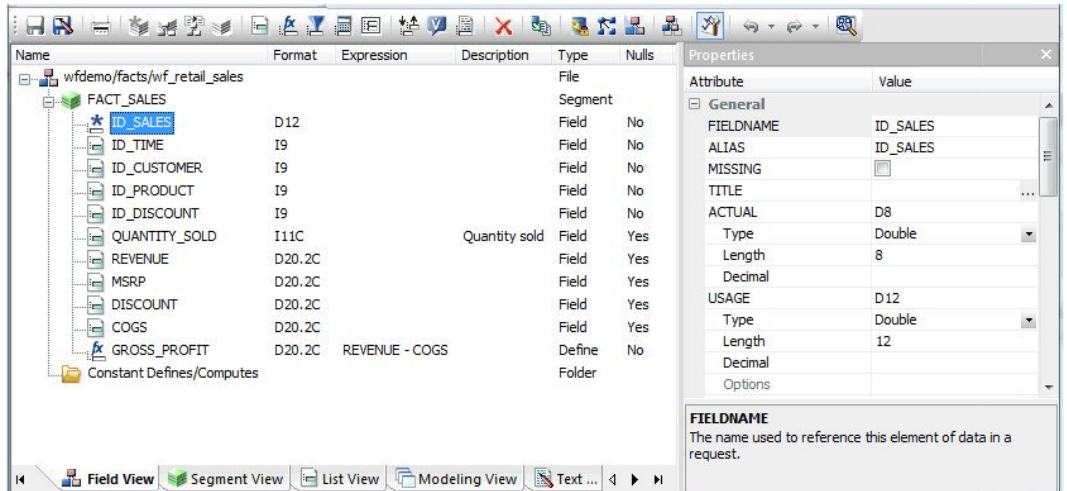
シノニムエディタの主要属性

[フィールド表示] (Field View) タブ、[セグメント表示] (Segment View) タブの左側には、セグメントとフィールドの階層が表示され、デフォルト設定でこのツールの右下に表示される [プロパティ] (Properties) ウィンドウには、選択した項目の属性と値が表示されます。

注意：利用可能な属性は、シノニムのタイプにより異なります。




[プロパティ] (Properties) ウィンドウなど、表示されるウィンドウ (ビジネスビュー、ディメンションビルダ、DBA) は、ツールバーのアイコンを使用して、サイズや表示エリアを変更したり、非表示にすることができます。



下図は、キーフィールドが選択された SQL データソースの例を示しています。




[プロパティ] (Properties) ウィンドウの下部には、現在選択されている属性についての情報が表示されます。ここでは、FIELDNAME 属性の説明が表示されています。

タブには、次のオブジェクトが表示されます。

オブジェクト	名前	機能
	シノニム	シノニムを示す一般的なアイコンです。[フィールド表示] (Field View) タブの左側に表示されます。使用されているファイル名とそのファイル内にあるアプリケーションについての情報を提供します。
	セグメント	シノニム内のセグメントです。ルート (親) セグメントはツリーの先頭に表示されます。
	仮想セグメント (クロスリファレンス)	仮想セグメント (クロスリファレンスセグメント) を示します。このアイコンは、薄い色で表示されます。







オブジェクト	名前	機能
	キーフィールド	キーフィールドを示します。
	フィールド	一般のフィールドを示します。
	一時項目 (DEFINE)	一時項目 (DEFINE) を示します。 一時項目 (DEFINE) についての詳細は、167 ページの「シノニムへの一時項目 (DEFINE) の追加」を参照してください。
	インデックス フィールド	ネイティブ DBMS に、このフィールド値の取得を高速化するためのインデックスが存在することを示します。
	フィルタ	マスターファイルのフィルタを示します。
	一時項目 (COMPUTE)	一時項目 (COMPUTE) を示します。
	グループ	グループを示します。 このオプションは、[ツール] (Tools) メニューから [オプション] (Options) を選択した場合に、有効になります。[オプション] (Options) ダイアログボックスで、[拡張オプションのサポート] (Support extended options) を選択します。
	ソートオブジェクト	ソートオブジェクトを示します。 このオプションは、[ツール] (Tools) メニューから [オプション] (Options) を選択した場合に、有効になります。[オプション] (Options) ダイアログボックスで、[拡張オプションのサポート] (Support extended options) を選択します。

オブジェクト	名前	機能
	スタイル	スタイルオブジェクトを示します。 このオプションは、[ツール] (Tools) メニューから [オプション] (Options) を選択した場合に、有効になります。[オプション] (Options) ダイアログボックスで、[拡張オプションのサポート] (Support extended options) を選択します。







参照

シノニムエディタツールバー

シノニムエディタのツールバーには、使用頻度の高い機能のボタンが含まれており、これらの機能にすばやくアクセスすることができます。ボタンの動作は選択するオブジェクトにより異なります。したがって、ツールバーボタンは無効の状態の場合があります。

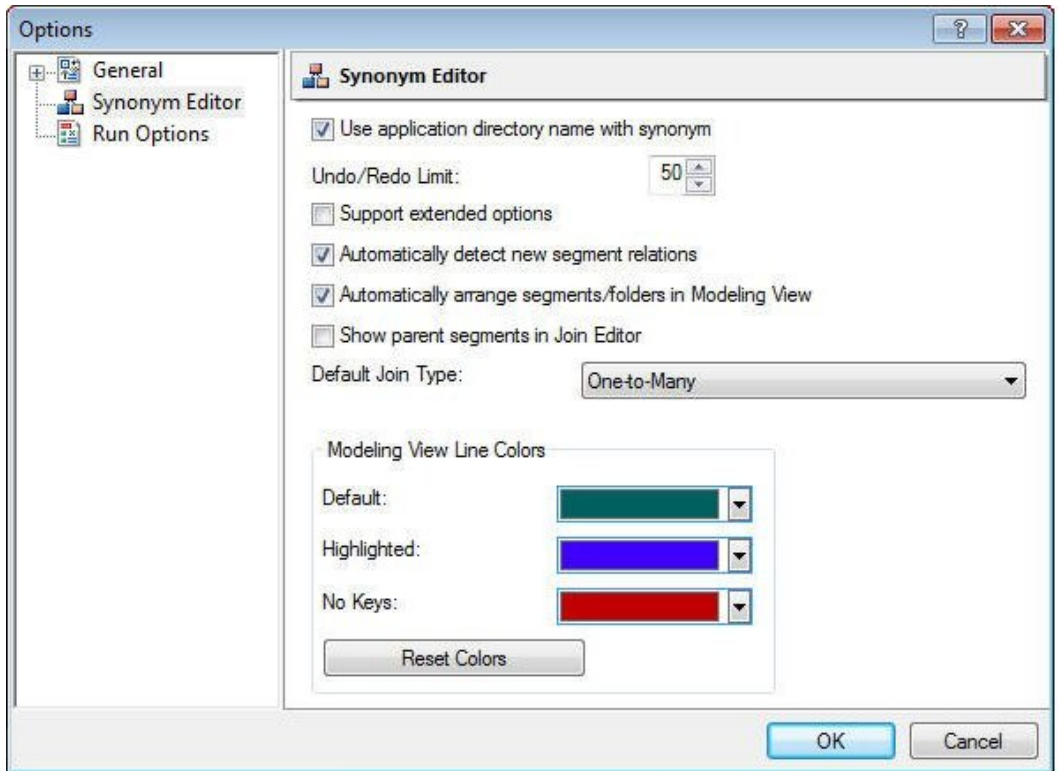
ボタン	説明
	編集したシノニムを保存します。
	シノニムに加えた変更を、新しいファイルに保存します (名前を付けて保存)。
	現在の表示を印刷することができます。
	既存のシノニムのセグメントを参照します。
	既存のシノニムからセグメントを追加することができます。
	シノニム作成ツールの [インポートしたメタデータからセグメントを追加] (Segment via Metadata Import) オプションを使用して、インポートしたメタデータからセグメントを追加することができます。このツールは、シノニムを作成し、ツールを起動したシノニム内のセグメントとして追加します。

ボタン	説明
	<p>セグメントを手動で追加し、シノニムエディタ内のセグメント属性フィールドに値を割り当てることができます。</p> <p>この方法は、Web Query データソースの場合のように、マスターファイルを最初から記述する場合にのみ使用します。</p>
	SUM/PRINT フィールドを追加します。
	<p>一時項目 (DEFINE) フィールドを追加します。</p> <p>一時項目 (DEFINE) についての詳細は、167 ページの「シノニムへの一時項目 (DEFINE) の追加」を参照してください。</p>
	<p>マスターファイルを追加します。</p> <p>フィルタについての詳細は、171 ページの「シノニムフィルタの作成」を参照してください。</p>
	<p>一時項目 (COMPUTE) を追加します。</p> <p>一時項目 (COMPUTE) についての詳細は、177 ページの「シノニムへの一時項目 (COMPUTE) の追加」を参照してください。</p>
	<p>グループを追加します。</p> <p>グループについての詳細は、194 ページの「シノニムへのグループフィールドの追加」を参照してください。</p>
	セグメントに変数を挿入します。
	選択項目を削除します。
	オブジェクトのサンプルデータの表示と更新を行います。
	DBA によるデータソースのアクセスが有効になります。

ボタン	説明
	ディメンションビルダで従属階層を有効にします。
	<p>ビジネスビューおよびカスタムマスターファイルを作成し、元のシノニムからフィールドを選択するために使用することができます。また、フィールド名、タイトル、説明をカスタマイズすることもできます。</p> <p>ビジネスビューについての詳細は、158ページの「ビジネスビューの作成」を参照してください。</p>
	マスターファイルの既存セグメントとフィールドに基づいて、[ビジネスビュー] (Business View) および [ディメンションビルダ] (Dimension Builder) ウィンドウのデフォルトツリー階層を作成します。
	属性のプロパティを記述します。
	操作を元に戻すことややり直すことができます。
	シノニムの概要を表示します。[モデル表示] (Modeling View) タブのみで使用可能です。

参照 シノニムエディタのオプション設定

メニューバーの [ツール] (Tools) をクリックして [オプション] (Options) を選択すると、下図のような [オプション] (Options) ファ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を使用して、シノニムエディタの設定と外観のカスタマイズが行えます。



シノニムエディタの設定ウィンドウには、次の項目とオプションがあります。

シノニムでアプリケーションディレクトリ名を使用する (Use application directory name with synonym)

このチェックをオンにすると、既存のシノニムの参照および db_lookup による変換の両方で、シノニム名の選択時に、アプリケーションディレクトリ名が使用されます。

元に戻す/やり直しの最大回数 (Undo/Redo Limit)

シノニムエディタで許可する元に戻す/やり直し操作の最大回数を指定します。

拡張オプションのサポート (Support extended options)

このチェックをオンにすると、シノニムの編集の際に、[ソートオブジェクト] (Sort Objects) および [スタイル] (Styles) フォルダの両方が表示されます。

新規セグメント関係を自動的に検出する (Automatically detect new segment relations)

シノニムエディタのモデル表示で新しいセグメントを作成する場合に、[新規セグメント関係を自動的に検出する] (Automatically detect new segment relations) のチェックをオンにしてテーブルを作業領域(モデル表示)にドラッグまたは選択します。これにより、アクセスファイルのキー情報に基づいて、関係 (JOIN) が自動的に作成されます。

モデル表示のセグメント/フォルダを自動整列 (Automatically arrange segments/folders in Modeling View)

モデル表示で複数または個別のテーブルを選択またはドラッグすると、これらが自動的に整列されます。

JOIN エディタに親セグメントを表示 (Show parent segments in Join Editor)

選択した場合、クラスタ JOIN または他のシノニムを参照する (または含む) シノニムの親セグメント内のフィールドを JOIN エディタに表示するかどうかは制御されます。

デフォルト JOIN タイプ (Default Join Type)

デフォルト JOIN タイプを設定します。

- 1 対 n (One-to-Many)** 複数インスタンス (1 対 n) の JOIN を示します。実行時に、各ホストレコードには、クロスリファレンスファイルの一致レコードを多数含めることができます。デフォルト設定では、[すべて JOIN] (Join All) が選択されています。
- 1 対 1 (One-to-One)** 単一インスタンス (1 対 1) の JOIN を示します。実行時に、各ホストレコードには、クロスリファレンスファイルの一致レコードを 1 つのみ含めることができます。

モデル表示線色 (Modeling View Line Colors) リストボックス

モデル表示のプロセスフローコネクタの色を設定します。

デフォルト (Default) モデル表示の線の色を変更します。

ハイライト (Highlighted) モデル表示でハイライトされた線の色を変更します。

キーなし (No keys) モデル表示でキーが存在しない線の色を変更します。

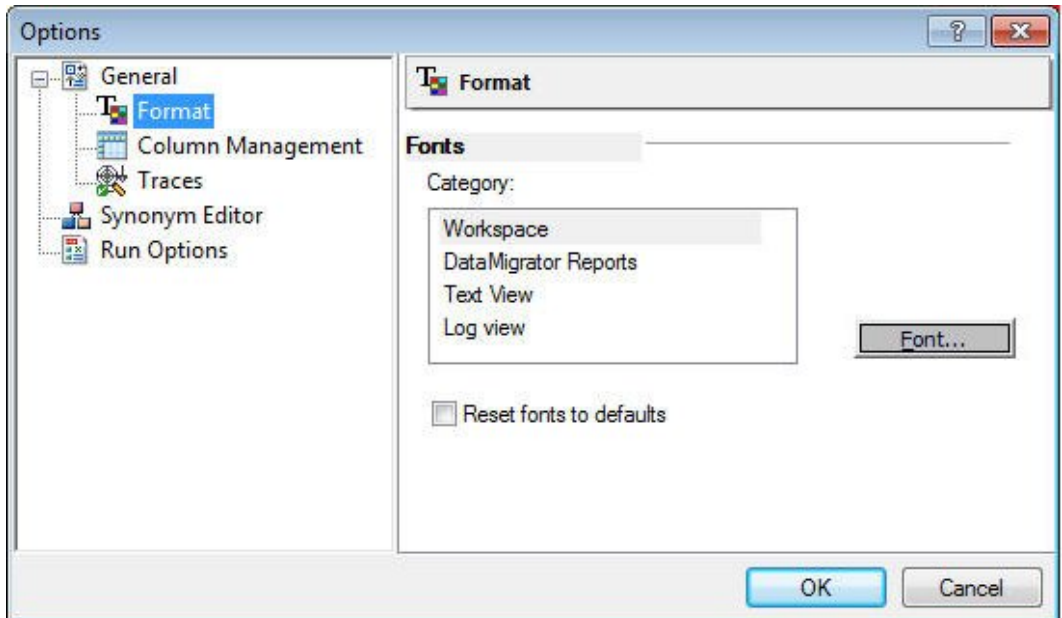
色をリセットしてデフォルトに戻す (Reset Colors)

デフォルト色に戻します。

参照

シノニムエディタのフォーマットオプション設定

[フォーマット](Format) ウィンドウには、[オプション](Options) ダイアログボックスからアクセスします。[オプション](Options) ダイアログボックスを開くには、[ツール](Tools) メニューから [オプション](Options) を選択します。下図のように、[全般](General) ノードを展開し、[フォーマット](Format) を選択します。



[フォント](Fonts) セクションには、次の項目とオプションがあります。

カテゴリ (Category)

プロセスフロー、ワークスペース、レポート、テキスト表示、ログ表示で使用するフォントを選択することができます。

フォント (Font)

フォントの設定を指定するダイアログボックスを表示します。

フォントをリセットしてデフォルトに戻す (Reset font to default)

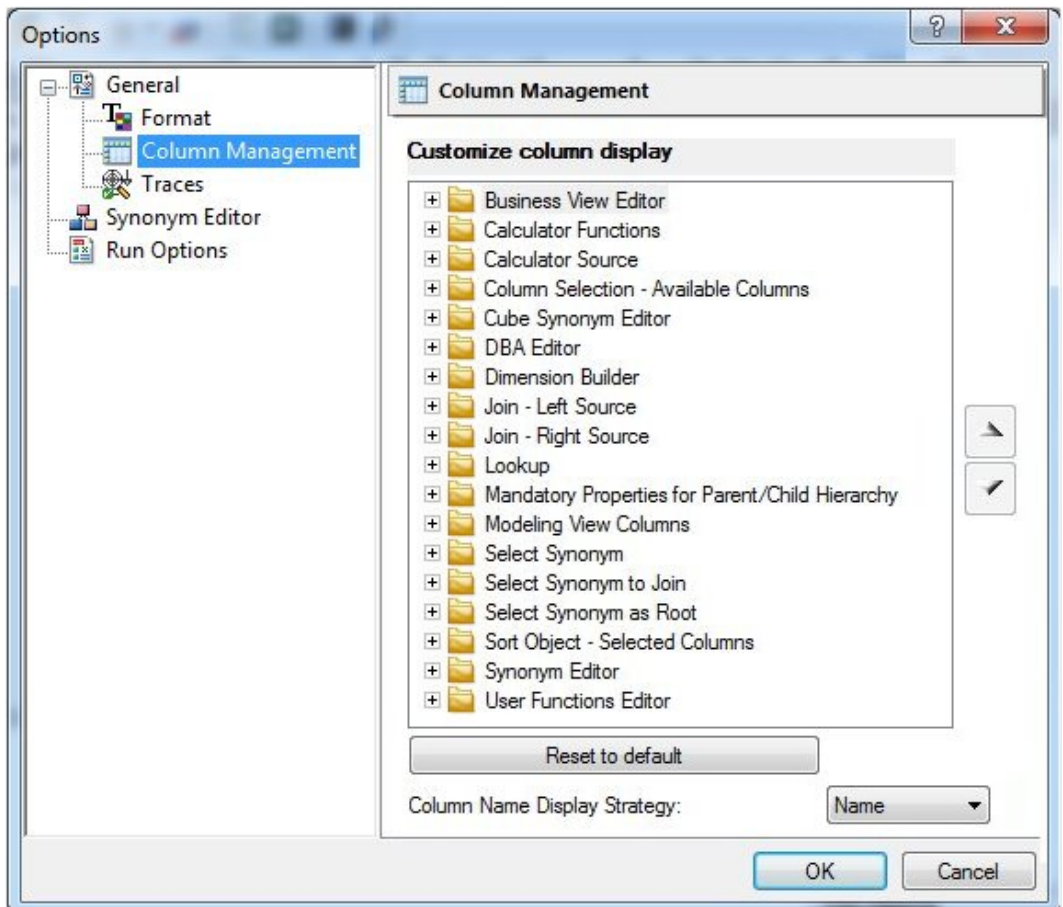
デフォルトフォントに戻します。

参照

シノニムエディタのカラム管理設定

カラム管理の設定を変更し、[変換] (Transform)、[カラムの選択] (Select Columns)、[JOIN]、[ソート] (Sort) ウィンドウ内に表示するカラムを選択することができます。ここでは列の表示順序を変更することも可能です。この設定は列の表示に関連するすべてのダイアログボックスで有効です。

[カラムの管理] (Column Management) ウィンドウへは、[オプション] (Options) ダイアログボックスからアクセスします。[オプション] (Options) ダイアログボックスを開くには、[ツール] (Tools) メニューから [オプション] (Options) を選択します。下図のように、[全般] (General) ノードを展開し、[カラムの管理] (Column Management) を選択します。



[カラムの管理] (Column Management) ウィンドウには、次の項目とオプションがあります。

カラム表示のカスタマイズ (Customize column display)

カスタマイズ可能なカラムに関連するダイアログボックスを表示します。フォルダを展開して表示されるチェックボックスを使用して、各ダイアログボックスにカラムを追加することができます。

デフォルトに戻す (Reset to default)

デフォルト値に戻します。

カラム名の表示条件 (Column Name Display Strategy)

ツリーおよびリストに表示する情報を制御します。選択可能なオプションには、[名前] (Name)、[タイトル] (Title)、[説明] (Description)、[エイリアス] (Alias) があります。タイトル、説明、エイリアスが存在しない場合は、デフォルト設定により名前別に表示されます。

注意： ビジネスビューで作業する場合は、[タイトル] (Title) の使用をお勧めします。

使用可能な任意の [カスタマイズ] (Customize) 列を展開すると、オプションのデフォルト設定を確認することができます。これらの列には、リストによっては、表示されないものもあります。次の列を追加することができます。

エイリアス (Alias)

フィールドの別名です。DBMS シノニムでは、実フィールド名です。

アプリケーション (Application)

シノニムの格納先アプリケーションを示します。

所属するセグメント (Belongs To Segment)

親セグメントを表示します。

接続 (Connection)

使用するアダプタ接続名を示します。

ホストデータ (Data Origin)

シノニムの作成日を示します。

更新日時 (Date Modified)

シノニムの最新の更新日時を示します。

説明 (Description)

フィールドの説明です。

式 (Expression)

フィールドの演算式です。

拡張子 (Extension)

シノニムの拡張子 (データソースタイプ) を示します。

フィールドタイプ (Field Type)

インデックスフィールドを (I)、読み取り専用フィールドを (R) で示します。

フォーマット (Format)

フィールドのデータが格納されている状態でのフィールドのタイプと長さです。

関数 (Function)

関数名を示します。

外部キー付き (Has Foreign Keys)

シノニムに外部キーが含まれていることを示します。

インデックス (Index)

インデックスフィールドを示します。

JOIN 条件 (Join conditions)

JOIN の条件を示します。

JOIN の従属元 (Join Parent)

JOIN の従属元を示します。

JOIN 方法 (Join Strategy)

JOIN の方法を示します。

キー (Keys)

シノニムのキーを示します。

長さ (Length)

フィールドの長さです。

NULL (Nulls)

フィールドへのミッシングデータの格納を許可するかどうかを指定します。

セグメント数 (Number of Segments)

シノニムのセグメント数を示します。

順序 (Order)

セグメント内でのフィールドの順序を示します。

接頭語 (Prefix)

フィールドの接頭語を示します。

主キーテーブル (Primary Key Tables)

シノニムの主キーを示します。

プロパティ (Properties)

フィールドが属性と基軸のどちらであるかを示します。

実テーブル名 (Real Table Name)

テーブルの実名または物理ファイル名を示します。

参照 (Reference)

フィールドのインデックスフィールドへの参照を示します。

小数部 (Scale)

小数点以下の最大桁数です。

SCD タイプ (SCD Type)

SCD (緩やかに変化するディメンション) 処理で使用します。

セグメント (Segment)

親セグメントを示します。

サイズ (Size)

シノニムのサイズを示します。

ソース (Source)

シノニムのソースを示します。

SQL 変換の注意 (SQL Conversion Notes)

SQL が変換されたことを示します。

テーブル (Table)

フィールドを含むシノニム名です。

タイトル (Title)

一般に使用するフィールド名の代わりとしてレポートに表示するタイトルを入力します。

タイプ (Type)

アプリケーションディレクトリ内のオブジェクトのタイプです。

次のフィールドは、各カラム表示オプションで使用可能です。デフォルトカラムは太字で表示されています。

 ビジネスビューエディタ (Business View Editor)

従属元セグメント (Belongs to Segment)、**フォーマット (Format)**、**式 (Expression)**、**説明 (Description)**、**NULL (Nulls)**、セグメント (Segment)、タイプ (Type)、長さ (Length)、小数部 (Scale)、エイリアス (Alias)、タイトル (Title)、SCD タイプ (SCD Type)、フィールドタイプ (Field Type)、インデックス (Index)、順序 (Order)、プロパティ (Property)、参照 (Reference)

 演算関数 (Calculator Functions)

フォーマット (Format)、**説明 (Description)**、SQL 変換の注意 (SQL Conversion Notes)

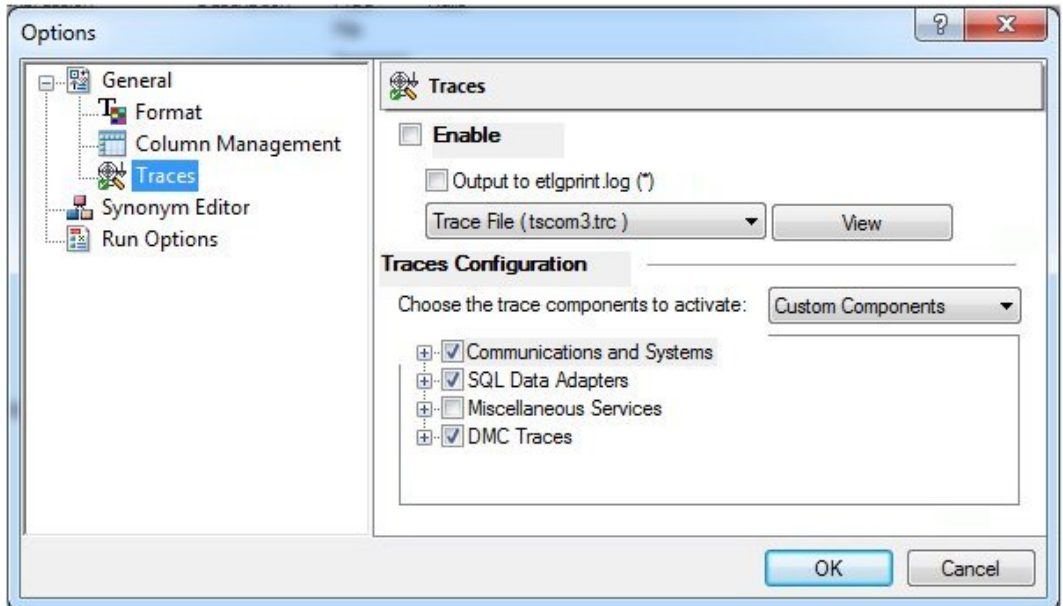
 演算ソース (Calculator Source)

テーブル (Table)、**フォーマット (Format)**、**説明 (Description)**、**NULL (Nulls)**、接頭語 (Prefix)、セグメント (Segment)、タイプ (Type)、式 (Expression)、長さ (Length)、小数部 (Scale)、エイリアス (Alias)、タイトル (Title)、SCD タイプ (SCD Type)、フィールドタイプ (Field Type)、インデックス (Index)、順序 (Order)、プロパティ (Property)、参照 (Reference)

必要に応じて、上下の矢印を使用してフィールドの表示順序を変更します。

参照 シノニムエディタのトレースオプション設定

下図のように、トレースオプションの設定により、サーバのトレースを構成することができます。

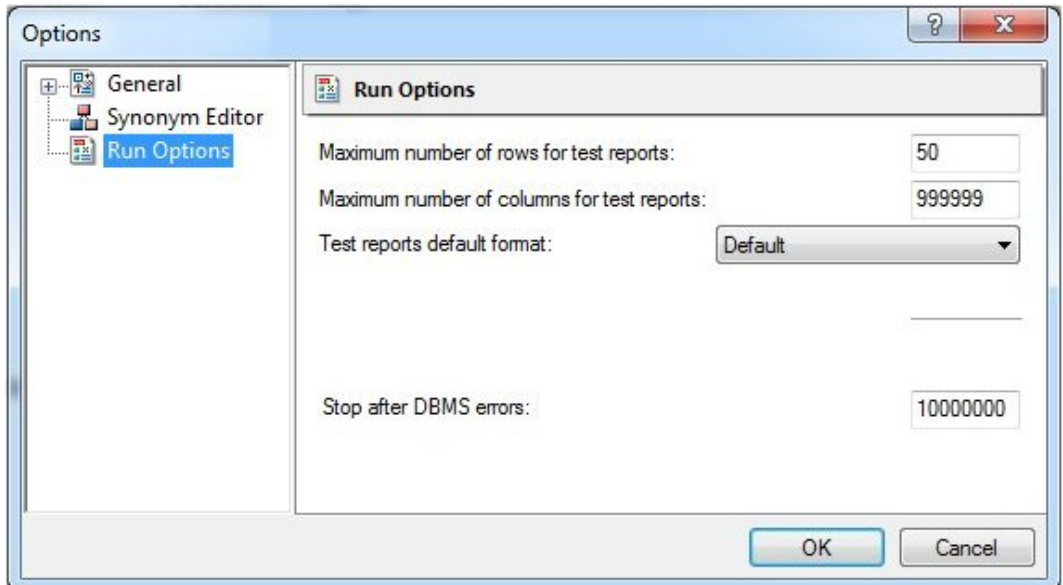


トレースオプションは、すべてのコンポーネント、デフォルトコンポーネント、カスタムコンポーネントセットに対して有効にすることができます。トレース出力は、Web コンソールのログ (etlprint.log)、tscom3.trc、または connection.trc トレースファイルに格納することができます。

参照

シノニムエディタの実行オプション設定

[実行オプション] (Run Options) ユーザ設定には、取得する行数とフィールド数、およびサンプルデータを取得する際のデフォルトフォーマットが含まれています。



[実行オプション] (Run Options) 設定ウィンドウには、次の項目とオプションがあります。

テストレポートの最大行数 (Maximum number of rows for test reports)

変換やSQLのテストを実行する際に、サンプルデータとして抽出する行数を設定します。デフォルト値は 50 です。

テストレポートの最大カラム数 (Maximum number of columns for test reports)

変換やSQLのテストを実行する際に、サンプルデータとして抽出するフィールド数を設定します。デフォルト設定ではすべてのフィールド (最大値は 999999) が抽出されます。

テストレポートのデフォルトフォーマット (Test reports default format)

変換やSQLのテストを実行する際に、サンプルデータとして抽出するレポートのフォーマットを設定します。デフォルト値は [デフォルト] (Default) です。レポートフォーマットには次の 7 種類があります。

- **デフォルト (Default)** シノニムの編集オプションに基づいて、数値フィールドおよび日付フィールドのフォーマットを設定します。

- **HTML** HTML 形式のレポートを生成します。
- **HTML - テキスト (HTML - Plain Text)** テキスト形式のレポートを作成します。
- **active report** HTML active report を生成します。このフォーマットは、オフライン分析を実行することができます。
- **Excel** Excel 形式のレポートを生成します。
- **PDF** PDF 形式のレポートを生成します。
- **フォーマットなし (Unformatted)** 数値フィールドと日付フィールドに、フォーマットを適用しません。

停止までの DBMS エラー回数 (Stop after DBMS error)

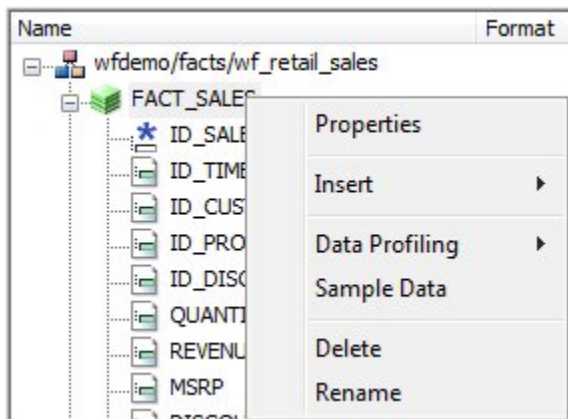
サーバがプロシジャの実行を停止するまでに許可する DBMS に関連するエラーの回数を設定します。

参照

シノニムエディタ - セグメントコンテキストメニュー

シノニムエディタでセグメントを右クリックすると、次のコンテキストメニューが表示されます。このメニューは、[フィールド表示](Field View) および [セグメント表示](Segment View) で使用可能です。

下図は、ルートセグメントが選択された SQL データソースの例を示しています。



次のオプションがあります。

プロパティ (Properties)

選択したセグメントのプロパティを表示するウィンドウを開きます。

挿入 (Insert)

次のいずれかを挿入することができます。

既存シノニムのセグメント参照を挿入 (Reference to Existing Synonym)

現在のシノニムから既存のシノニムへの参照ポインタを追加します。選択リストには、候補のシノニムが表示されます。結果としてソースシノニムを変更する場合は、作業中のシノニムを再度開くことで、それらの変更が反映されます。

既存スノーflakeシノニムの参照 (Reference to Existing Synonym with Snowflake)

現在のシノニムから既存のシノニムへの参照ポインタを追加します。選択リストには、参照シノニムとともに、候補のシノニムが表示されます。

既存シノニムのセグメントコピー (Copy of Existing Synonym)

既存のシノニムの静的なコピーを、作業中のシノニムに挿入します。結果としてソースシノニムを変更する場合は、作業中のシノニムには、変更は反映されません。

インポートメタデータのセグメント (Segment via Metadata Import)

シノニム作成ツールを使用し、新しいシノニムを作成して現在のシノニムに追加することができます。

手動でセグメントを追加 (Segment Manually)

コードを直接記述することにより、シノニムを挿入します。

フィールド (Field)

セグメントに一般的なフィールドを挿入します。

一時項目 (DEFINE) (Define)

シノニムに一時項目 (DEFINE) を挿入します。

フィルタ (Filter)

セグメントにフィルタを挿入します。

一時項目 (COMPUTE) (Compute)

ファイルに一時項目 (COMPUTE) を挿入します。

グループ (Group)

セグメントにグループを挿入します。

変数 (Variable)

シノニム/アクセスファイルパラメータで使用されている変数名を挿入します。

データプロファイリング (Data Profiling)

セグメントデータの特性を表示します。

[データプロファイリング] (Data Profiling) は、マスターファイル階層ツリーのすべてのフィールドの右クリックで表示されるコンテキストメニューから利用することができます。

サンプルデータ (Sample Data)

ワークスペースに、選択したセグメントまたはシノニムのサンプルデータを表示します。

サンプルデータは、シノニムエディタツールバーから使用することができます。

下図は、セグメントのサンプルデータの例を示しています。

	ID_SALES	ID_TIME	ID_CUSTOMER	ID_PRODUCT	ID_DISCOUNT	Quantity Sold	Revenue
1	1	1461	26735	2207	100	4	676.00
2	2	82	30480	2003	104	1	149.99
3	3	426	16018	2106	101	1	620.99
4	4	1237	4349	1112	100	1	379.99
5	5	390	21915	2003	100	1	199.99
6	6	917	4947	4006	100	2	1,599.98
7	7	589	40645	2003	102	1	169.99
8	8	23	4245	2053	100	4	399.96
9	9	102	73945	1018	104	1	397.49

注意：サンプルデータは、シノニムのフィールドデータのテスト、取得されるレコードのタイプを確認する方法として適しています。また、JOINの実行やデータソース接続のテストなどにも役立ちます。

親キー付きサンプルデータ (Sample Data with parent key)

親セグメントに結合されている、選択したセグメントのサンプルデータを表示します。このオプションは、複数セグメントのシノニムで子セグメントを選択した場合にのみ使用可能です。

削除 (Delete)

セグメントを削除します。

名前の変更 (Rename)

セグメント名を変更します。

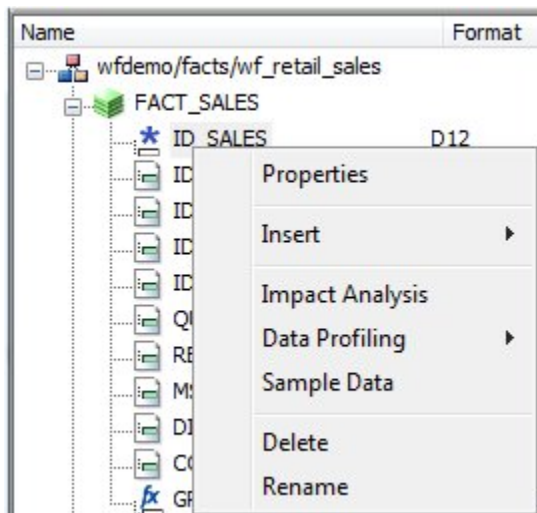
JOIN のプロパティ (Join Properties)

JOIN エディタを開いて、選択したセグメントの親セグメントの結合方法を指定します。このオプションは、複数セグメントのシノニムで子セグメントを選択した場合にのみ使用可能です。

参照**シノニムエディタ - カラムおよびフィールドショートカットメニュー**

シノニムエディタでフィールドを右クリックすると、次のコンテキストメニューが表示されます。このメニューは、[フィールド表示] (Field View)、[セグメント表示] (Segment View)、[リスト表示] (List View) で使用可能です。

下図は、フィールドが選択された SQL データソースの例を示しています。



注意：使用可能なオプションは、データソースによって異なります。

次のオプションがあります。

プロパティ (Properties)

選択したフィールドのプロパティを表示するウィンドウを開きます。

挿入 (Insert)

次のいずれかを挿入することができます。

フィールド (Field)

シノニムに一般的なフィールドを挿入します。

一時項目 (DEFINE) (Define)

シノニムに一時項目 (DEFINE) を挿入します。

フィルタ (Filter)

セグメントにフィルタを挿入します。

一時項目 (COMPUTE) (Compute)

ファイルに一時項目 (COMPUTE) を挿入します。

グループ (Group)

シノニムにグループフィールドを挿入します。

変数 (Variable)

シノニム/アクセスファイルパラメータで使用されている変数名を挿入します。

インパクト分析 (Impact Analysis)

ワークスペースの特定フィールドのインパクト分析レポートを表示します。インパクト分析を使用すると、マスターファイルまたは、マスターファイル内のフィールドにアクセスするフィールドを識別することができます。インパクト分析についての詳細は、213 ページの「[メタデータとプロシジャの分析](#)」を参照してください。

データプロファイリング (Data Profiling)

フィールドデータの特性を表示します。[データプロファイリング](Data Profiling) は、マスターファイル階層ツリーのすべてのフィールドの右クリックで表示されるコンテキストメニューから利用することができます。

サンプルデータ (Sample Data)

ワークスペースにサンプルデータを表示します。

下図は、フィールドのサンプルデータの例を示しています。

	Revenue
1	676.00
2	149.99
3	620.99
4	379.99
5	199.99
6	1,599.98
7	169.99
8	399.96
9	397.49
10	2,249.99
11	350.99

注意：サンプルデータは、シノニムのフィールドデータのテスト、取得されるレコードのタイプを確認する方法として適しています。また、JOINの実行やデータソース接続のテストなどにも役立ちます。

日付の分解 (**Decompose Date**)

日付フィールドを分解し、年、四半期、月、日の一時項目 (DEFINE) を作成します。

注意：[日付の分解] (Decompose Date) は日付フィールドでのみ表示されます。

削除 (**Delete**)

フィールドを削除します。

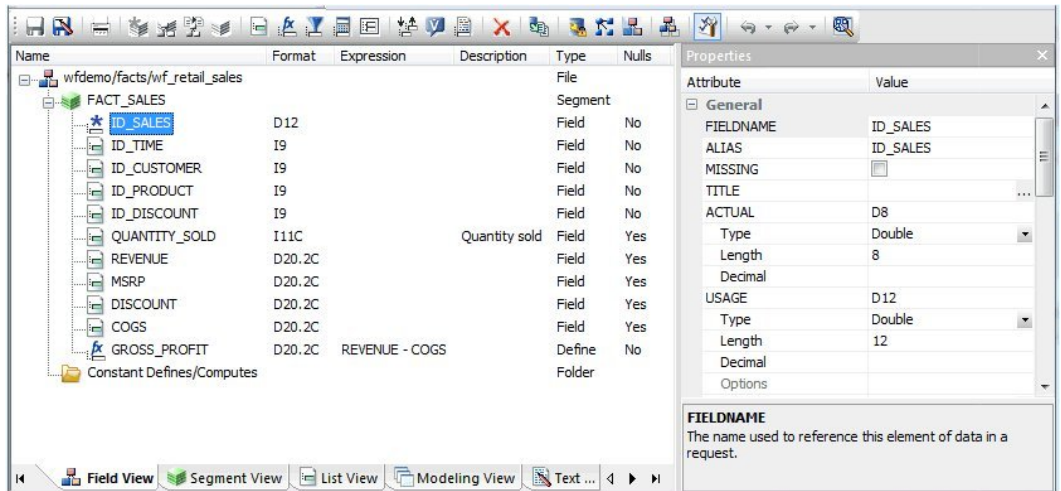
名前の変更 (**Rename**)

フィールド名を変更します。

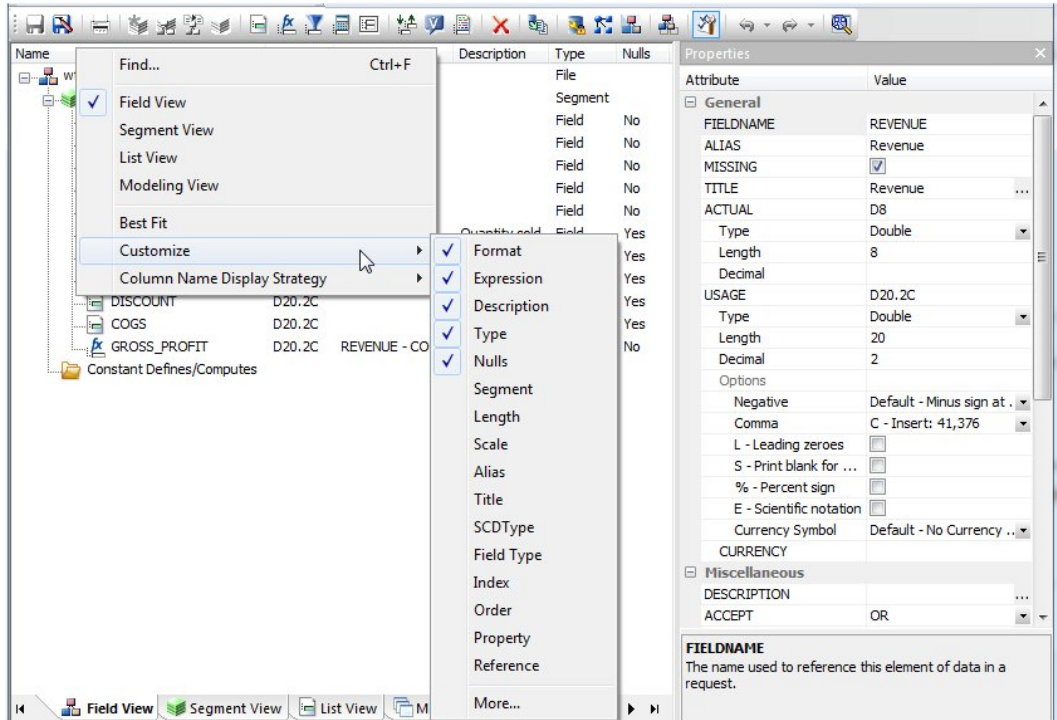
参照 シノニムエディタ - フィールド表示タブ

[フィールド表示] (Field View) タブには、使用可能なフィールドおよびセグメントのテーブルが表示されます。

下図は、[フィールド表示] (Field View) タブ内の Microsoft SQL Server データソースの例を示しています。



ツールバーメニューの下部に表示される行 (名前、フォーマット、式、説明、NULL)は、右クリックとドラッグにより、左右に移動することや、表示枠内での順序を変更することができます。[名前] (Name) 列を移動することはできません。下図のように、この行を右クリックして [カスタマイズ] (Customize) を選択すると、この行に表示する列をカスタマイズすることができます。

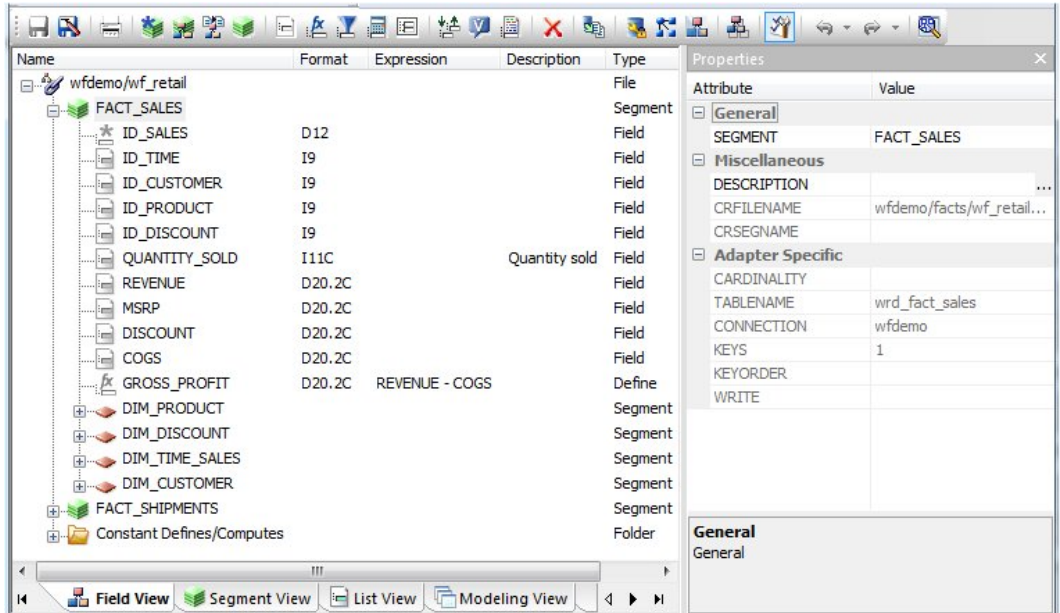


このメニューは、ワークスペースでブランクの領域を右クリックして表示することもできます。このメニューからは、表示を切り替えることもできます。さらに、最後のオプションを選択すると、[オプション] (Options) ダイアログボックスの [カラムの管理] (Column Management) セクションを開くことができます。詳細は、93 ページの「[シノニムエディタのカラム管理設定](#)」を参照してください。

参照 シノニムエディタ - セグメント表示タブ

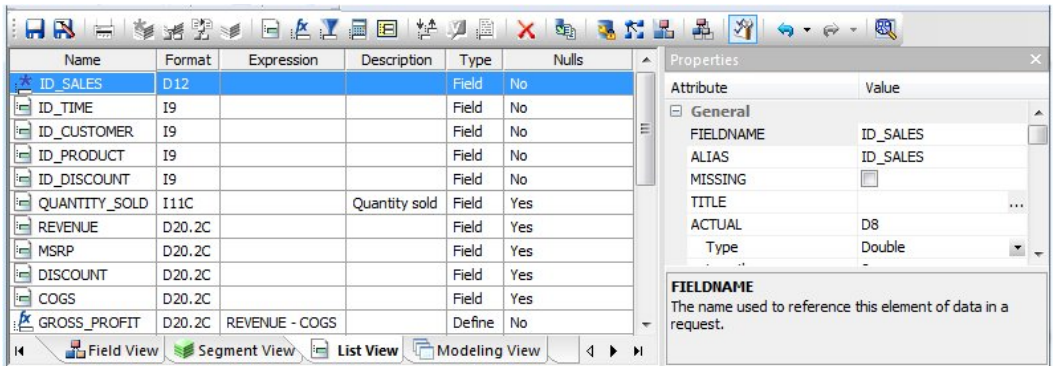
[セグメント表示] (Segment View) タブには、シノニムに格納されているセグメントが表示されます。

下図は、[セグメント表示] (Segment View) タブ内の Microsoft SQL Server データソースの例を示しています。



参照 シノニムエディタ - リスト表示タブ

下図のように、[リスト表示] (List View) タブの左側にはオブジェクトのリストが表示され、右側には選択した項目の属性および値が表示されます。

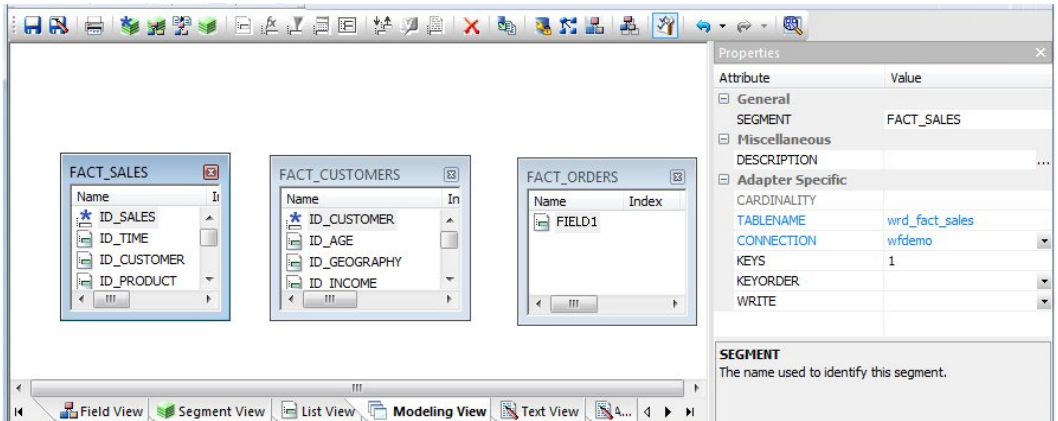


注意：[リスト表示] (List View) で列見出しをクリックすると、フィールド値の表示を降順または昇順でソートするオプションを提供するコンテキストメニューが表示されます。表示は、任意のフィールドを基準にソートすることができます。

参照 シノニムエディタ - モデル表示タブ

[モデル表示] (Modeling View) タブをクリックすると、シノニムが図示されます。[モデル表示] (Modeling View) を使用して、クラスタ JOIN の作成、JOIN プロパティの表示、セグメントの追加および編集が行えます。

下図は、[モデル表示] (Model View) タブ内の SQL データソースの例を示しています。



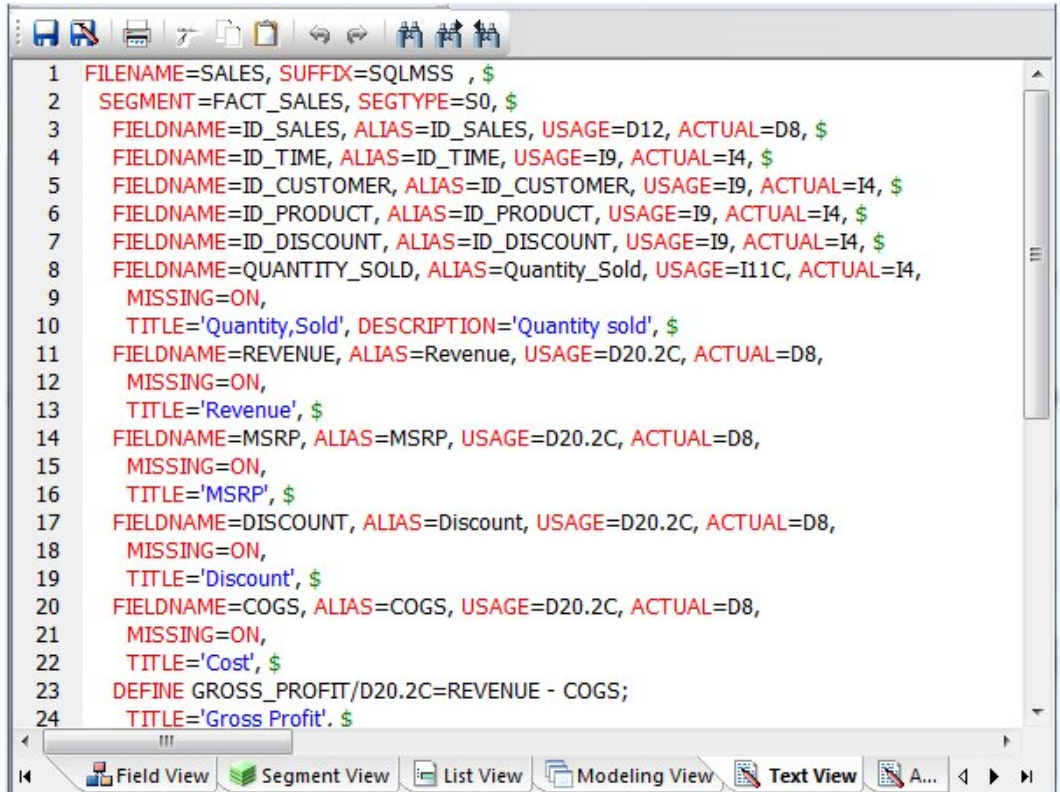
モデル表示では、ワークスペースでブランクの領域を右クリックすると、このビューで許可されたタスクが表示されます。たとえば、セグメントの挿入や、シノニムの自動整列などが行えます。モデル表示についての詳細は、125 ページの「モデル表示を使用したシノニムの拡張」を参照してください。

参照 シノニムエディタ - テキスト表示タブ

この表示は、シノニムをエディタで開きます。

[ファイル] (File) メニューから [印刷] (Print) を選択して、マスターファイルのコードを印刷することや、検索の実行が可能なほか、コードに変更を加えることもできます。

下図は、[テキスト表示] (Text View) タブ内の Microsoft SQL Server データソースの例を示しています。



```
1 FILENAME=SALES, SUFFIX=SQLMSS , $
2 SEGMENT=FACT_SALES, SEGTYPE=S0, $
3 FIELDNAME=ID_SALES, ALIAS=ID_SALES, USAGE=D12, ACTUAL=D8, $
4 FIELDNAME=ID_TIME, ALIAS=ID_TIME, USAGE=I9, ACTUAL=I4, $
5 FIELDNAME=ID_CUSTOMER, ALIAS=ID_CUSTOMER, USAGE=I9, ACTUAL=I4, $
6 FIELDNAME=ID_PRODUCT, ALIAS=ID_PRODUCT, USAGE=I9, ACTUAL=I4, $
7 FIELDNAME=ID_DISCOUNT, ALIAS=ID_DISCOUNT, USAGE=I9, ACTUAL=I4, $
8 FIELDNAME=QUANTITY_SOLD, ALIAS=Quantity_Sold, USAGE=I11C, ACTUAL=I4,
9 MISSING=ON,
10 TITLE='Quantity_Sold', DESCRIPTION='Quantity sold', $
11 FIELDNAME=REVENUE, ALIAS=Revenue, USAGE=D20.2C, ACTUAL=D8,
12 MISSING=ON,
13 TITLE='Revenue', $
14 FIELDNAME=MSRP, ALIAS=MSRP, USAGE=D20.2C, ACTUAL=D8,
15 MISSING=ON,
16 TITLE='MSRP', $
17 FIELDNAME=DISCOUNT, ALIAS=Discount, USAGE=D20.2C, ACTUAL=D8,
18 MISSING=ON,
19 TITLE='Discount', $
20 FIELDNAME=COGS, ALIAS=COGS, USAGE=D20.2C, ACTUAL=D8,
21 MISSING=ON,
22 TITLE='Cost', $
23 DEFINE GROSS_PROFIT/D20.2C=REVENUE - COGS;
24 TITLE='Gross Profit'. $
```

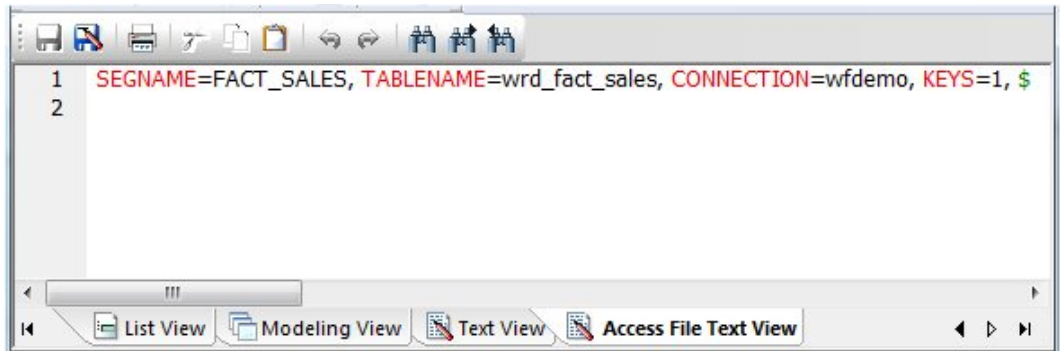
参照

シノニムエディタ-アクセスファイルテキスト表示タブ

[アクセスファイルテキスト表示] (Access File Text View) タブをクリックすると、データベースへのアクセスに使用する、シノニムのアクセスファイルの記述が表示されます。

注意：必要な場合は、[ファイル] (File) メニューから [印刷] (Print) を選択して、マスターファイルのコードを印刷することや、検索の実行が可能のほか、コードに変更を加えることもできます。

下図は、[アクセスファイルテキスト表示] (Access File Text View) タブ内の SQL データソースの例を示しています。



シノニム属性の表示および編集

手順/構文

シノニムの属性を表示、編集するには

参照

ファイル属性の概要

セグメント属性の概要

フィールド属性の概要

シノニムエディタを使用して、シノニムの属性の表示および編集が行えます。

手順

シノニムの属性を表示、編集するには

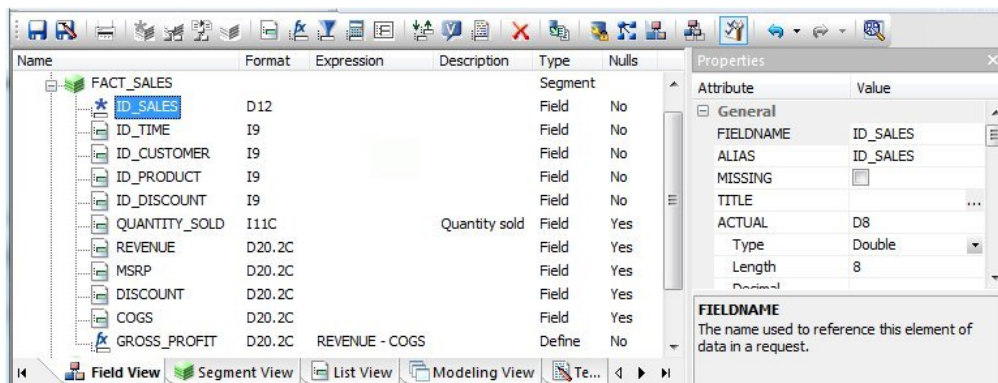
シノニムの属性を表示、編集するには、次の手順を実行します。

1. [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。

デフォルト設定では、シノニムエディタは前回使用した表示モードで開き、[フィールド表示] (Field View) タブの左側に、セグメントとフィールドの階層、右側に選択した項目の属性と値が表示されます。

注意：利用可能な属性は、シノニムのタイプにより異なります。

下図は、キーフィールドが選択された SQL データソースの例を示しています。



注意：フィールドとともに、フォーマット、式、説明、NULL 値を階層に表示することができます。これらの値を編集するには、シノニムエディタの右側でそれぞれに対応する属性フィールドを使用します。

2. 新しい値を入力するか、ドロップダウンリストおよびチェックボックスを使用することで、属性値を変更します。

注意：シノニムエディタは、マスターファイルを使用不可能にする変更を禁止しています。したがって、灰色表示されたフィールド値を編集することはできません。また、変更正しい構文またはフォーマットが適用されていない場合は、フィールドのテキストが赤色で表示されます。保存しようとしているファイルにエラーが含まれている場合、メッセージおよび警告が表示されます。

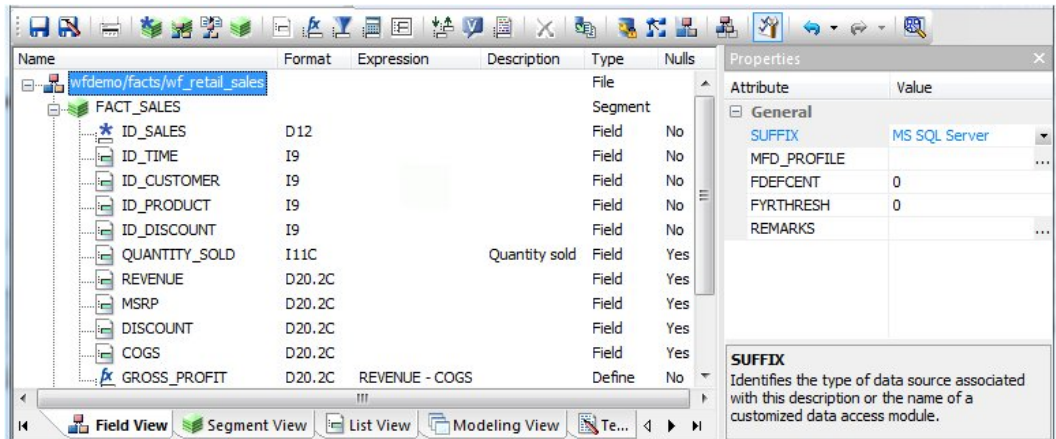
3. [ファイル] (File) メニューから [保存] (Save) を選択し、変更を保存します。
4. シノニムエディタを閉じるには、[ファイル] (File) メニューから [閉じる] (Close) を選択するか、画面右上隅のコントロールボタンをクリックします。

注意：変更したシノニムエディタを保存せずに閉じようとする、保存を確認するダイアログボックスが表示されます。

参照

ファイル属性の概要

下図は、シノニム名が選択された SQL データソースの例を示しています。



注意：属性リストの下部には、現在選択されている属性についての情報が表示されます。ここでは、SUFFIX 属性の説明が表示されています。

シノニムには、次のファイル属性を含めることができます。

全般 (General)

SUFFIX

シノニムまたはデータソースのタイプを示します。

MFD_PROFILE

マスターファイルを含むリクエストの前に実行するプロシジャ名です。

FDFCENT

4 桁の西暦年の日付を処理する際の、デフォルトの上位 2 桁の値を定義します。

注意：上位 2 桁の値が以前のものである (例、19xx) データを抽出する場合を除き、デフォルト設定 (0) を使用します。

FYRTHRESH

基準とする年を定義します。この値は、西暦年の上位 2 桁の値を適用する (FDFCENT) 最下位の年を示します。

注意：上位 2 桁の値が以前のものである (例、19xx) データを抽出する場合を除き、デフォルト設定 (0) を使用します。

REMARKS

ファイルレベルの説明を格納することができます。シノニムの多言語による説明を指定することができます。REMARKS は、レポートの実行時に、ファイル名とともに表示されます。

多言語の説明についての詳細は、121 ページの「[多言語のタイトルおよび説明の設定](#)」を参照してください。

DATASET

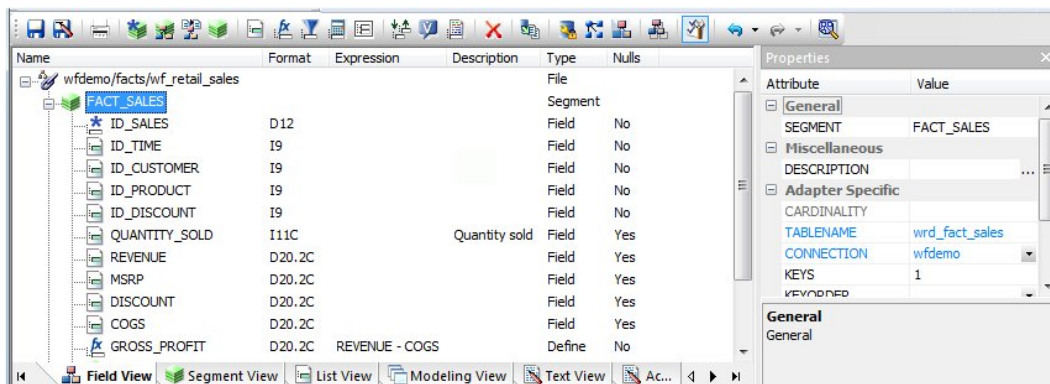
ファイル名に使用するデータソースの物理パスを、拡張子およびデータファイルのパスとともに指定します。

注意：利用可能な属性は、シノニムのタイプにより異なります。

参照

セグメント属性の概要

下図は、セグメントが選択された SQL データソースの例を示しています。



注意：属性リストの下部には、現在選択されている属性についての情報が表示されます。ここでは、SEGMENT 属性の説明が表示されます。

シノニム内のセグメントには、次の属性を含めることができます。

全般 (General)

SEGMENT

セグメント名です。

SEGTYPE

セグメントのタイプまたは親セグメントとの関係を指定し、セグメント内のキーフィールドおよびソート順を示します。

タイプ (Type) [タイプ] (Type) ドロップダウンリストから、セグメントタイプおよびソートオプションを選択します。

キー (Keys) レコードはデータソース内のキーフィールドでソートされます。ソートに使用するキーフィールド数を入力します。たとえば、複数の従業員に同一の社員番号が割り当てられることはないため、社員番号フィールドをキーとして使用することができます。1つのセグメントインスタンスは、複数のキーフィールドを持つことができます。したがって、複数のフィールド値を使用してレコードを区別することができます。

SEGSUF

SEGSUF はマスターファイルによりデータソースの一部が記述され、そのデータソースが構造全体を記述するのではなく別のデータソースに記述される場合に使用されます。

注意：SEGSUF はセグメント、および (存在する場合) その子孫のデータソースタイプであり、そのタイプは SUFFIX 値とは異なります。

その他 (Miscellaneous)**DESCRIPTION**

セグメントの説明 (コメント) を格納します。

多言語の説明についての詳細は、121 ページの「[多言語のタイトルおよび説明の設定](#)」を参照してください。

CRFILENAME

クロスリファレンスデータソースの名前です。

CRSEGNAME

クロスリファレンスセグメント名です。

CRKEY

クロスリファレンスセグメントの共有 JOIN フィールドを示します。

アダプタ依存 (Adapter Specific)

注意：[アダプタ依存] (Adapter Specific) の項目は、シノニムとともにアクセスファイルコンポーネントが生成された場合に表示されます。

CARDINALITY

シノニムを生成したときの元のデータソースに存在するレコード数を定義します。

TABLENAME

テーブル名またはビュー名です。テーブル名のほかに、オーナー ID が含まれる場合もあります。シノニムによっては、データソース名が含まれる場合もあります。

CONNECTION

使用するアダプタ接続名を示します。

KEYS

主キーを構成するフィールドの数です。

KEYORDER

主キーに基づいてデータをソートする際の論理的ソート順です。

WRITE

テーブルに対して書き込み操作が可能かどうかを指定します。

DBSPACE

テーブルが存在するストレージ領域です。

PERSISTENCE

テーブルの持続タイプと関連したテーブルプロパティを指定します。これは、一時テーブルをサポートする DBMS ではオプションです。それ以外では必須です。

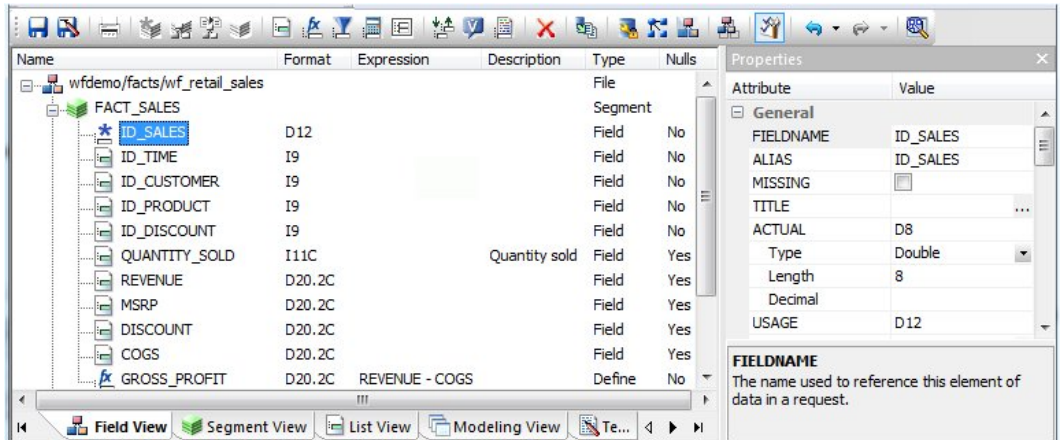
注意：利用可能な属性は、シノニムのタイプにより異なります。

参照

フィールド属性の概要

下図は、キーフィールドが選択された SQL データソースの例を示しています。

注意：利用可能な属性は、シノニムのタイプにより異なります。



注意：属性リストの下部には、現在選択されている属性についての情報が表示されます。ここでは、FIELDNAME 属性の説明が表示されています。

シノニム内のフィールドには、次の属性を含めることができます。

全般 (General)**FIELDNAME**

フィールド名です。

ALIAS

フィールドの別名です。DBMS シノニムでは、実フィールド名です。

MISSING

トランザクションで値がない場合の空 (NULL) データの処理方法です。

TITLE

一般に使用するフィールド名の代わりとしてレポートに表示するタイトルを入力します。これにより、フィールドに多言語のタイトルを指定することができます。

多言語のタイトルについての詳細は、121 ページの「[多言語のタイトルおよび説明の設定](#)」を参照してください。

ACTUAL

実際にデータソースに格納されているデータのデータタイプおよび長さを記述します。

USAGE

フィールドのデータタイプおよびフォーマット(使用フォーマットまたは表示フォーマット)を記述します。

注意：その他の属性の DEFCENT および YRTHRESH は、USAGE フィールドが日付、時間、または日付時間(タイムスタンプ)フォーマットに設定されている場合に、利用することができます。これらの属性は、フィールドの西暦年の上位 2 桁および年のしきい値を入力するために使用します。

- **タイプ (Type)** フィールドの値を固定文字、可変文字、テキスト、整数、浮動、倍精度、実数、パック 10 進数、日付、時間、日付時間(タイムスタンプ)、BLOB としてとして設定することができます。
- **長さ (Length)** 文字の長さを設定することができます。
- **オプション (Options)** 負の整数の表示方法の設定、カンマ(,)の表示と非表示の設定、先頭の 0(ゼロ)の表示設定、0(ゼロ)のブランクの表示設定、パーセント記号の設定、通貨記号の選択が行えます。

その他 (Miscellaneous)

DESCRIPTION

フィールドの説明(コメント)を格納します。

多言語の説明についての詳細は、121 ページの「[多言語のタイトルおよび説明の設定](#)」を参照してください。

ACCEPT

データ確認の条件を指定します。

OR 受容可能な値を指定します。

FROM-TO 使用可能な値のフィールド範囲を指定します。

FIND ファイルおよびファイル名を入力し、DB2 Web Queryでデータソースおよび受容値リストを検索する場所を指定します。確認条件が割り当てられたデータフィールドのフィールド名、フィールドが検索されるターゲット DB2 Web Query データソースのフィールド名、確認条件を含むターゲットデータフィールドのフィールド名を入力します。

注意：FIND は、DB2 Web Query データソースのみで利用することができ、OLAP を有効にしたマスターファイルには適用されません。

WITHIN

ディメンションに追加するフィールド名を格納します。

これらの WITHIN ステートメントは、ディメンションビルダでシノニムに追加され、リレーショナルテーブルの OLAP を有効にします。WITHIN ステートメントにより、OLAP コントロールパネルを使用する OLAP の分析が可能になります。

プロパティ (Properties)

フィールドのプロパティを設定します。

参照 (Reference)

別のデータソースを参照することができます。

FIELDTYPE

インデックスフィールドです。FIELDTYPE にインデックスを付けるには、インデックス (Index) のチェックをオンにします。

注意：FIELDTYPE=R は読み取り専用フィールドを示します。この設定は、RDBMS によってフィールドに値が自動的に割り当てられる場合に役立ちます。

ACCESS_PROPERTY

フィールドデータのアクセスオプションを指定します。

- ❑ **INTERNAL** サンプルデータまたは利用可能なカラムリストに表示しないフィールドを定義します。レポートツールのフィールドリストに表示するフィールドを制限します。
- ❑ **NEED_VALUE** データアクセスに値が必要なフィールドを定義します。
- ❑ **選択基準** 値、範囲、複数值による定義します。

AUTHRESP

認証処理の結果を記述するフィールドを定義します。正しいレスポンス値を ACCEPT 属性に入力します (複数值を許可する場合、OR により記述します)。

AUTHTOKEN

実行する処理へ入力値として渡す、レスポンストークンを含むフィールドを定義します。

HELPMESSAGE

フィールドにヘルプメッセージを追加します。

SCD タイプ (SCD Type)

SCD を設定します。このオプションは、既存のリレーショナルターゲットでのみ使用することができ、ターゲットオブジェクトでSCDが有効になっている場合にのみフローで使用します。

サロゲートキー (surrogate key) テーブルのキーフィールドで、SCD タイプは [ブランク] (blank) です。これ以外で SCD タイプが [ブランク] (blank) のフィールドに対しては、SCD 処理は行われません。シノニムでは、テーブルの先頭フィールドではない場合も、常にこのフィールドが最初に表示されます。

注意：サロゲートキーには、整数を指定する必要があります。シノニムで、フィールドが読み取り専用 (Readonly) および自動増加 (Autoincrement) として識別された場合、サロゲートキー値の割り当てに、データベースの自動増加または識別処理を使用します。これらの識別が行われない場合、DataMigrator の処理により、このフィールドの値が割り当てられ、新しい行が追加されるたびに、その値が増加されます。

論理キーフィールド (Logical Key Field) データベースキーです。

アクティベーションフラグ (Activation Flag) 行が最新であることを示します。

開始日/終了日 (Begin Date/End Date) 行の値に対する日付範囲です。終了日がブランクの場合、行が最新であることを示します。開始日と終了日を使用する場合は、[終了日] (End Date) フィールドを NULLABLE (NULL 値可) として作成する必要があります。

タイプ I (Type I) (履歴を上書き) データベース値を新しい値で上書きするフィールドを示します。

タイプ II (Type II) (履歴を保存) データベース行に非アクティブフラグが設定されている、または終了日が割り当てられているフィールドを示します。新しい値の新しい行が挿入されます。

ブランク (blank) (恒久フィールド) データベース値が変化しないことを示します。

多言語のタイトルおよび説明の設定

手順/構文

多言語のタイトルおよび説明を設定するには

シノニムエディタでシノニムを開き、多言語のタイトル、キャプション、説明テキストを入力することができます。これらの説明は、シノニムに対して生成されたレポートで、指定した言語で表示されます。

[多言語] (Multilingual Titles) ダイアログボックスは、シノニムエディタの REMARKS、TITLE、DESCRIPTION 属性値で使用することができます。

注意：利用可能な属性は、シノニムのタイプにより異なります。

手順

多言語のタイトルおよび説明を設定するには

1. [データサーバ] (Data Servers) エリアで [マスターファイル] (Master File) をダブルクリックするか、[ファイル] (File) メニューから [シノニムエディタで編集] (Edit in Synonym Editor) を選択します。

シノニムエディタの [フィールド表示] (Field View) タブに、マスターファイルが開きます。

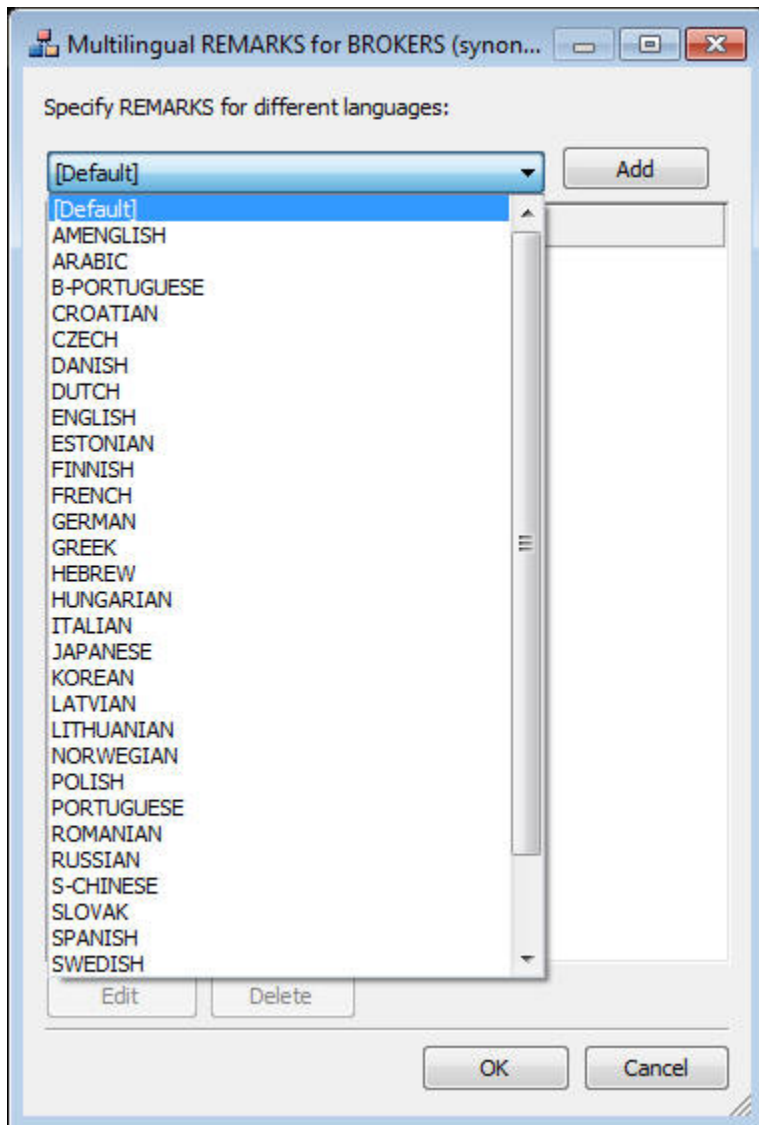
2. 多言語テキストを追加するには、次の手順を実行します。
 - TITLE - 左側のマスターファイル階層で、フィールドの 1 つをクリックします。
 - REMARKS - 左側のマスターファイル階層で、ルートレベル (アプリケーション/ファイル名) をクリックします。
 - DESCRIPTION - 左側のマスターファイル階層フィールドで、フィールド、セグメント、またはカスタムフィールドをクリックします。

関連する属性および値が右側に表示されます。

3. REMARKS、TITLE、または DESCRIPTION の値フィールド末尾の [...] (参照) ボタンをクリックします。

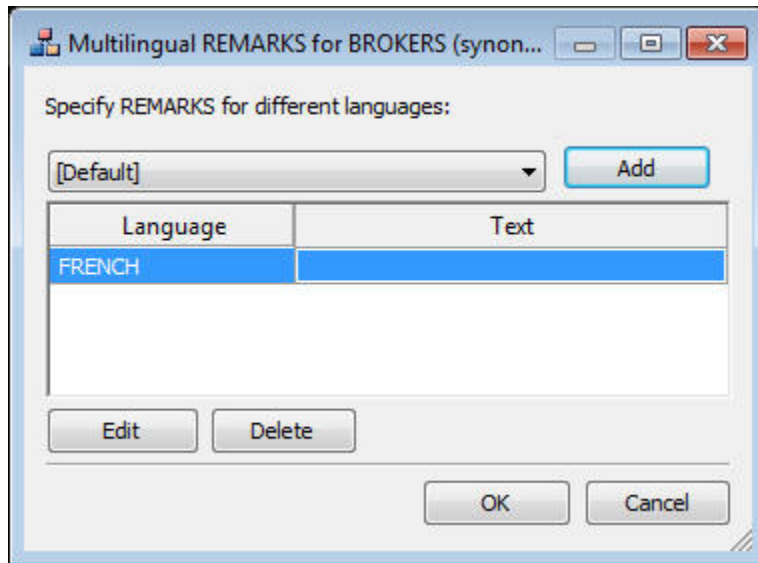
[多言語] (Multilingual) ダイアログボックスが開きます。

4. [異なる言語の TITLE/DESCRIPTION を入力] (Specify Titles/Descriptions for different languages) ドロップダウンリストから、備考(説明)またはタイトルに表示する言語を選択します。



5. [追加] (Add) をクリックします。

選択した言語は、デフォルト言語 (コードページ選択により設定) の下に追加されます。



6. [テキスト] (Text) テキストボックスに、説明またはタイトルを入力します。

- 7.** その他のタイトルまたは説明を追加、編集、削除することもできます。
その他の言語を追加するには、次の手順を実行します。

 - a.** 追加する言語をドロップダウンリストから選択します。
 - b.** [追加] (Add) をクリックします。
 - c.** フィールドのタイトルまたは説明を入力します。

入力済みのタイトルまたは説明を編集するには、次の手順を実行します。

 - a.** タイトルまたは説明を選択し、[編集] (Edit) をクリックします。
タイトルをダブルクリックして編集することもできます。
 - b.** タイトルまたは説明を直接入力します。
タイトルを削除するには、次の手順を実行します。
 - a.** 削除するタイトルまたは説明、および言語を選択します。
 - b.** [削除] (Delete) をクリックします。
- 8.** [OK] をクリックし、[多言語 TITLE] (Multilingual Titles) ダイアログボックスを閉じます。
- 9.** [ファイル] (File) メニューから [保存] (Save) を選択し、シノニムを保存します。
- 10.** シノニムを閉じるには、[ファイル] (File) メニューから [閉じる] (Close) を選択するか、画面右上隅のコントロールボタンをクリックします。

モデル表示を使用したシノニムの拡張

手順/構文

モデル表示を使用してシノニムを拡張するには

モデル表示を使用してシノニムを編集するには

参照

モデル表示コンテキストメニュー

JOIN エディタダイアログボックス

シノニムエディタの [モデル表示] (Modeling View) タブは、アクセスファイルに保持および保存される画面構成をシノニム用に視覚表示します。[モデル表示] (Modeling View) を使用して、OLAP 分析のディメンションの定義、JOIN のプロパティの表示、クラスタ JOIN の作成、セグメントの追加と編集を行うことができます。ディメンションビルダモデル表示の使用についての詳細は、154 ページの「[OLAP 分析のディメンション定義](#)」を参照してください。

注意：モデル表示は、キューブデータソースでは利用できません。

手順

モデル表示を使用してシノニムを拡張するには

モデル表示を使用し、セグメントを追加してシノニムを拡張します。

1. [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。

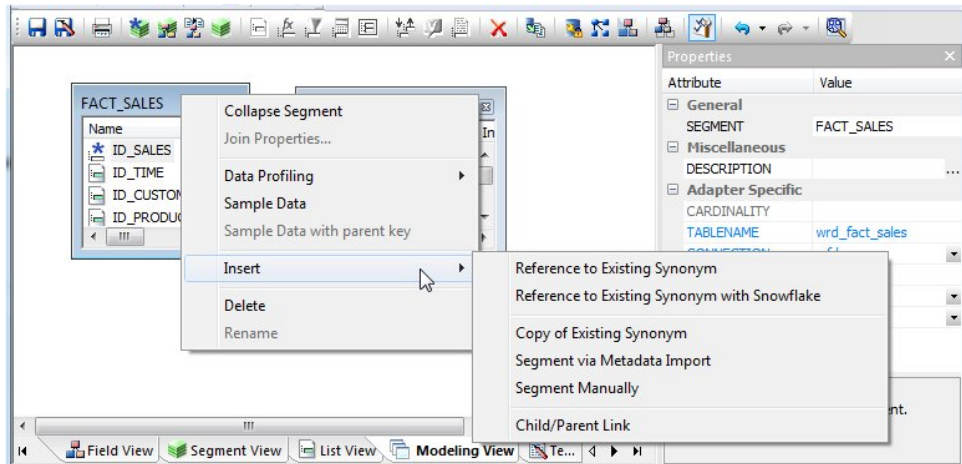
シノニムエディタでシノニムが開きます。

2. [モデル表示] (Modeling View) タブをクリックします。

ワークスペースにシノニムエディタの [モデル表示] (Modeling View) タブが表示されます。

3. ワークスペースで、セグメントを右クリックします。

次のコンテキストメニューが表示されます。



4. 次のいずれかの方法で、セグメントを挿入します。

- ❑ 既存シノニムのセグメント参照を挿入 (Insert Reference to Existing Synonym) - 既存のシノニムを現在のシノニムのセグメントとして追加します。
- ❑ 既存スノーフレイクシノニムの参照を挿入 (Insert Reference to Existing Synonym with Snowflake) - 既存のシノニムを現在のシノニムのセグメントとして追加します。選択リストには、参照シノニムとともに、候補のシノニムが表示されます。
- ❑ 既存のシノニムからセグメントのコピーを挿入 (Insert Copy of Existing Synonym) - 既存のシノニムのテーブルをセグメントとして追加します。
- ❑ インポートしたメタデータからセグメントを挿入 (Insert Segment Via metadata import) - [シノニムの作成] (Create Synonym) ツールを使用して、セグメントを追加します。このツールは、シノニムを作成し、ツールを起動したシノニム内のセグメントとして追加します。
- ❑ 手動でセグメントを挿入 (Insert Segments Manually) - 現在のシノニムに手動でセグメントを追加します。

注意：この方法は、DB2 Web Query データソースの場合のように、マスターファイルを最初から記述する場合にのみ使用します。

手順

モデル表示を使用してシノニムを編集するには

モデル表示を使用してシノニムを編集するには、次の手順を実行します。

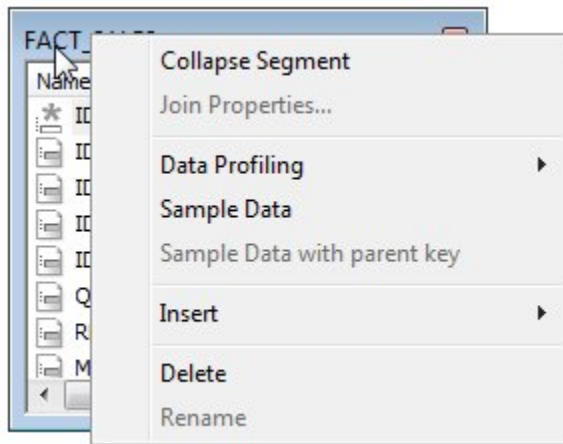
1. [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。

シノニムエディタでシノニムが開きます。

2. [モデル表示] (Modeling View) タブをクリックします。
3. [セグメント] (Segment) アイコンを右クリックします。

コンテキストメニューが表示されます。ここにはセグメントを追加するためのオプションが表示されます。

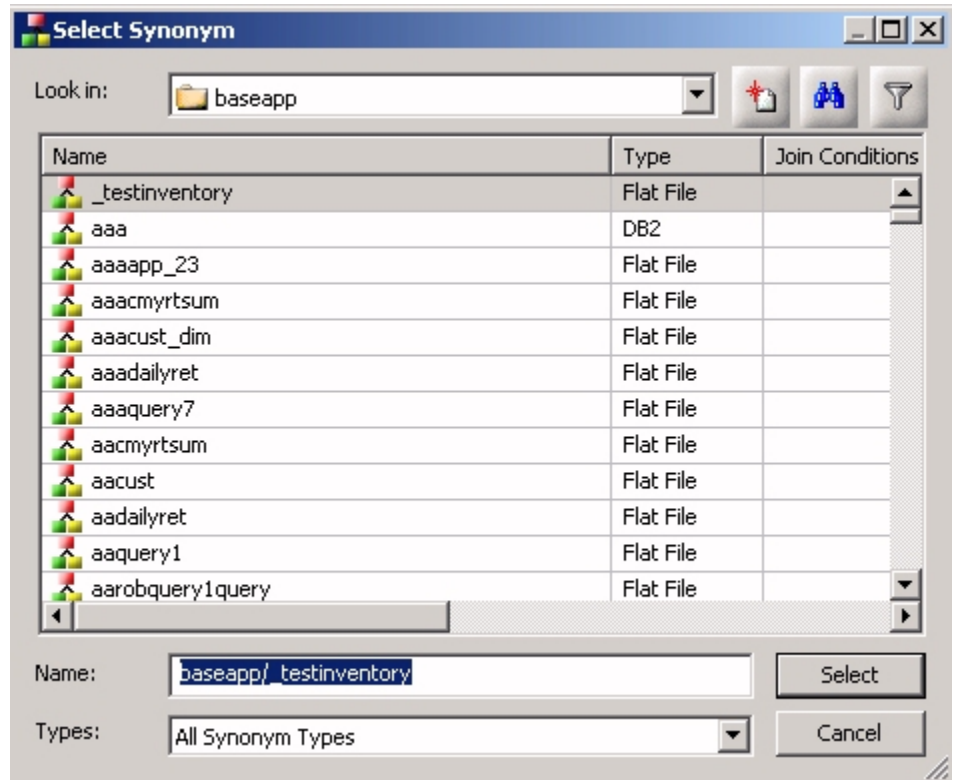
注意：セグメントを追加するオプションは、ワークスペース上部のアイコンから選択することもできます。



既存シノニムからセグメントを挿入するには、次の手順を実行します。

- a. [挿入] (Insert) を選択し、コンテキストメニューから [既存シノニムのセグメントを参照] (Reference to Existing Synonym) を選択します。

[シノニムの選択] (Select Synonym) ダイアログボックスが開きます。



列は、カスタマイズやソートが行えます。このダイアログボックスのシノニムは、Reporting Server のアプリケーションパスの構成に基づいて表示されます。

- b.** シノニムをクリックし、[選択] (Select) をクリックします。

シノニムにセグメントが追加されます。

既存のスノーflakeシノニムからセグメントを挿入するには、次の手順を実行します。

- a.** [挿入] (Insert)、[既存スノーflakeシノニムの参照を挿入] (Reference to Existing Synonym with Snowflake) を選択します。

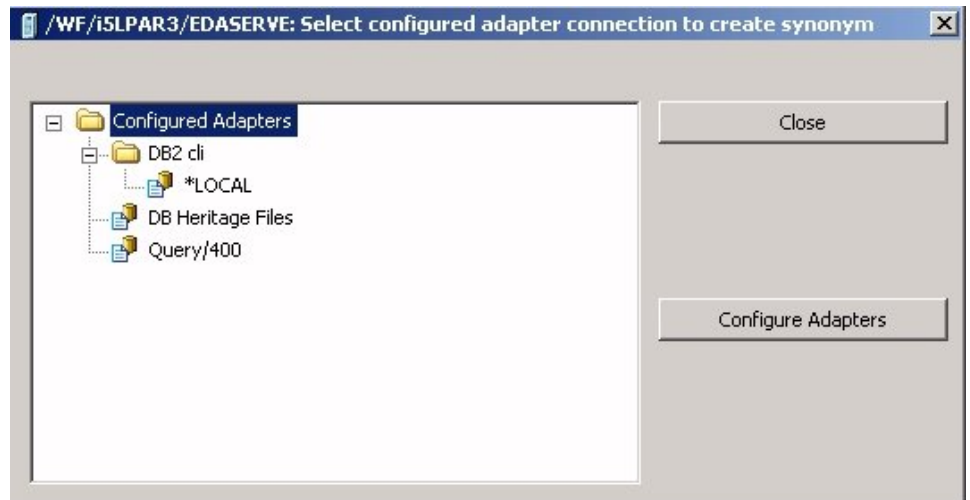
この手順は、既存シノニムの参照を挿入する場合と同様ですが、選択リストに候補のシノニムと参照シノニムが表示される点が異なります。

インポートしたメタデータからセグメントを挿入するには、次の手順を実行します。

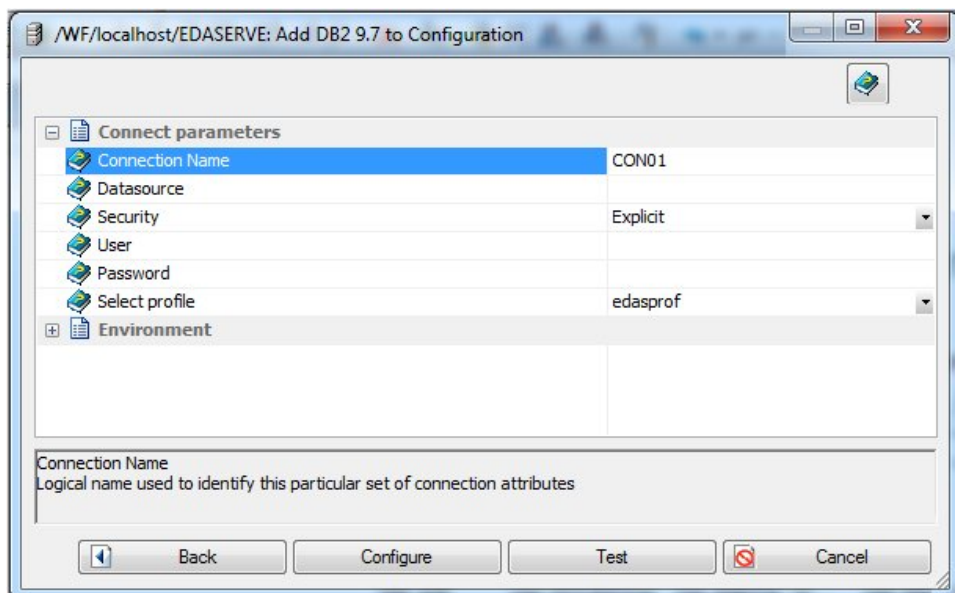
- a. コンテキストメニューから、[挿入](Insert)、[インポートしたメタデータからセグメントを挿入](Segment via Metadata Import) を選択します。

この方法により、シノニム作成ツールを起動してシノニムを作成し、このシノニムをインポート実行元のシノニムの新しいセグメントに関連付けることができます。

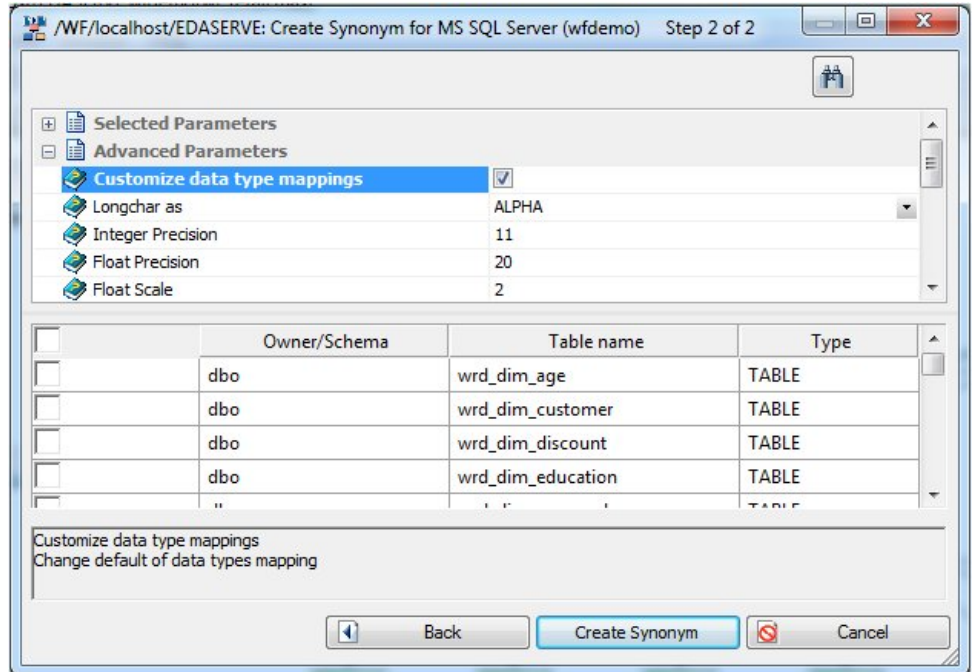
- b. 下図のように、このオプションを選択すると、最初に [アダプタの選択] (Select Adapter) ダイアログボックスが表示され、構成済みアダプタ構成を選択して続行することや、必要な場合は、新しいアダプタを構成することができます。



下図は、表示される画面の例を示しています。ここで、接続パラメータの情報を
入力します。



下図は、シノニムを作成するテーブルを選択する画面の例を示しています。

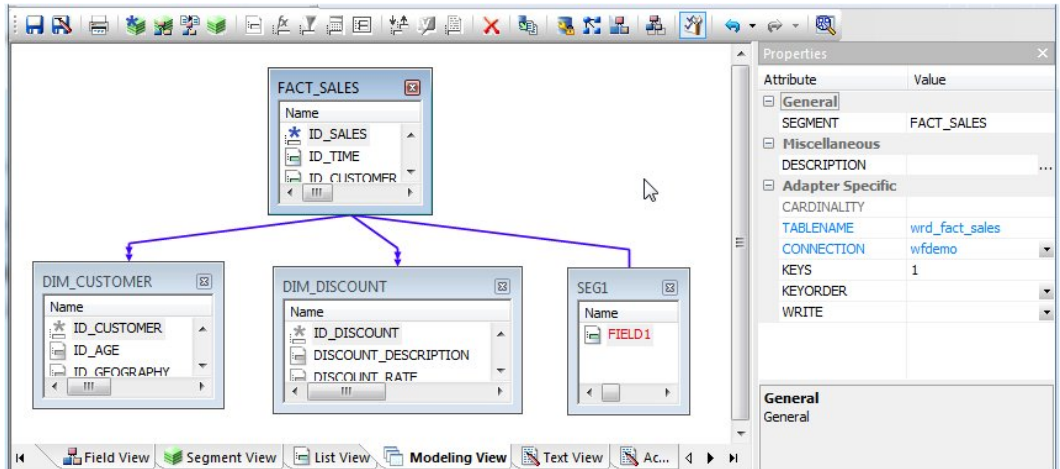


シノニムにセグメントが追加されます。

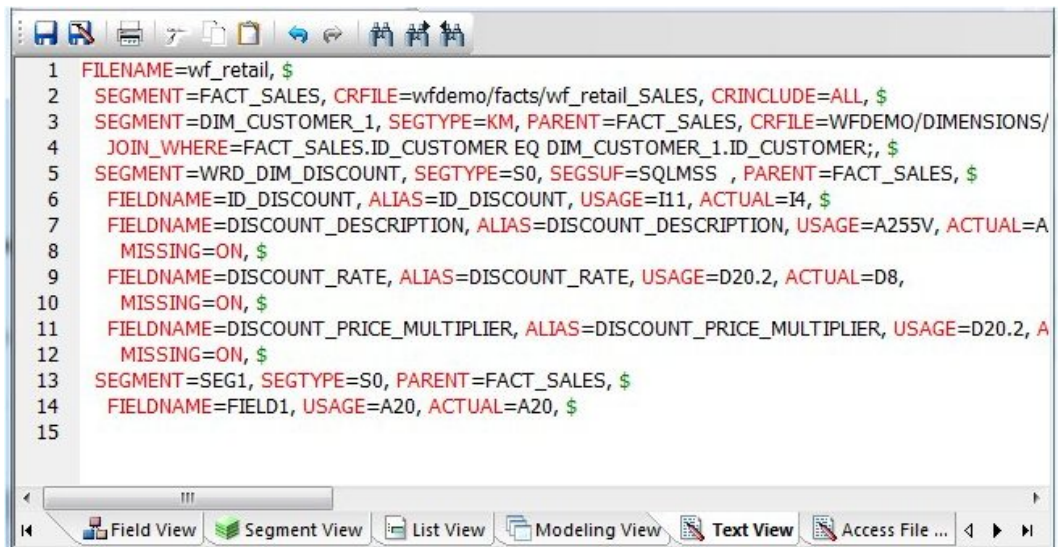
セグメントを手動で挿入するには、次の手順を実行します。

- a. [挿入] (Insert)、[手動でセグメントを追加] (Segment Manually) を選択します。
モデル表示にシノニムが追加され、デフォルトのセグメント名を使用してデフォルトのフィールドが作成されます。
- b. 新しいセグメントの属性は、[モデル表示] (Modeling View) タブまたはその他のタブの、[プロパティ] (Properties) ウィンドウで変更することができます。

下図は、[モデル表示] (Modeling View) タブでいくつかのセグメントが追加された状態を示しています。

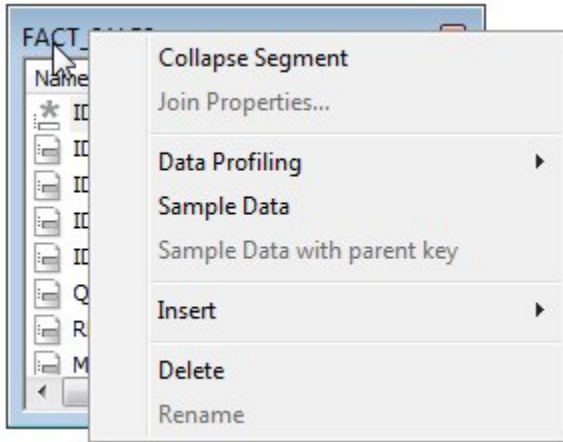


結果のセグメント属性は、[シノニムテキスト表示] (Synonym Text View) または [アクセスファイルテキスト表示] (Access File Text View) タブで確認することができます。



参照 モデル表示コンテキストメニュー

シノニムエディタの [モデル表示] (Modeling View) タブでシノニムまたはセグメントを右クリックすると、次のコンテキストメニューが表示されます。



コンテキストメニューには、次のコマンドがあります。

セグメントの折りたたみ (Collapse Segment)

[ファイル] (File) アイコンと表形式表示を切り替え、フィールド、サンプルデータ、サンプルデータの親キーを表示することができます。

ヒント：リスト表示を開くには、[ファイル] (File) アイコンをダブルクリックすることもできます。リスト表示を閉じるには、リストのタイトルバーをダブルクリックするか、タイトルバーの [X] ボタンをクリックします。

JOIN のプロパティ (Join Properties)

[JOIN のプロパティ] (Join Properties) ウィンドウを表示します。

データプロファイリング (Data Profiling)

セグメントデータの特徴を表示します。

サンプルデータ (Sample Data)

ワークスペースにサンプルデータを表示します。

親キー付きサンプルデータ (Sample Data with parent key)

ワークスペースに親キー付きサンプルデータを表示します。

挿入 (Insert)

- **既存シノニムの参照 (Reference to Existing Synonym)** 既存のシノニムを参照し、現在のシノニムにセグメントを挿入することができます。
- **既存スノーflakeシノニムの参照 (Reference to Existing Synonym with Snowflake)** 既存のシノニムを参照し、現在のシノニムにセグメントを挿入することができます。選択リストには、参照シノニムとともに、候補のシノニムが表示されます。
- **既存シノニムのセグメントコピー (Copy of Existing Synonym)** 既存のシノニムを現在のシノニムにコピーします。
- **インポートメタデータのセグメント (Segment via Metadata Import)** シノニム作成ツールで新しいシノニムを作成し、これを現在のシノニムに追加することができます。
- **手動でセグメントを追加 (Segment Manually)** 現在のシノニムにセグメントを追加します。このセグメントは手動で記述する必要があります。

削除 (Delete)

セグメントを削除します。

名前の変更 (Rename)

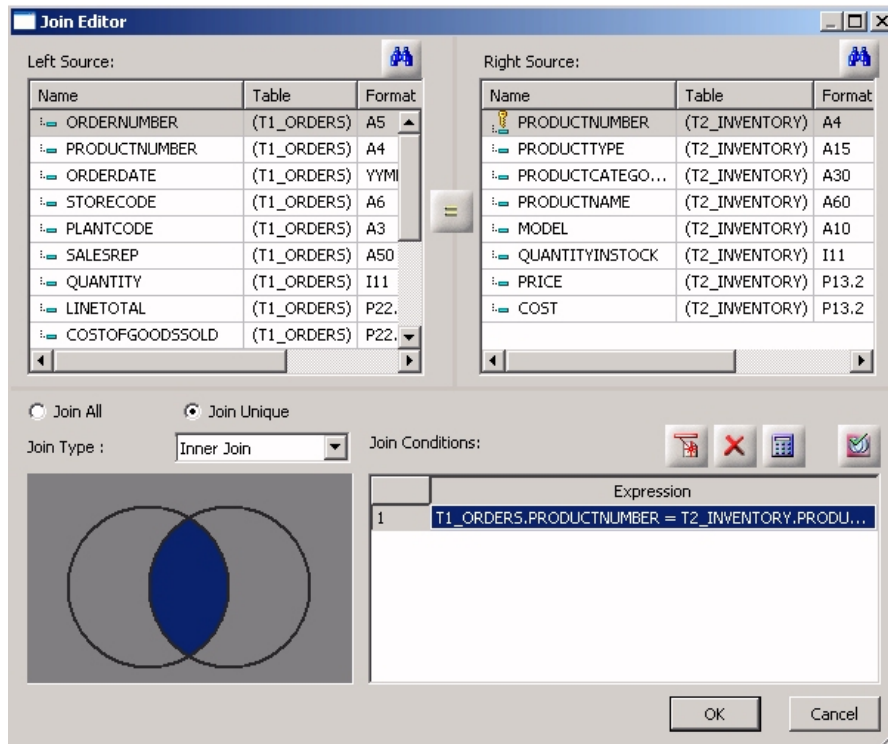
セグメント名を変更します。

参照

JOIN エディタダイアログボックス

シノニムエディタの [モデル表示] (Modeling View) タブで、セグメントから [JOIN のプロパティ] (Join Properties) を選択すると、[JOIN エディタ] (Join Editor) ダイアログボックスが表示されます。

[JOIN エディタ] (Join Editor) ダイアログボックスには、[左ソースカラム] (Left Source Column)、[右ソースカラム] (Right Source Column)、[JOIN タイプ] (Join Type)、[JOIN 条件] (Join Condition)、および [1 対 n] (One-to-Many) と [1 対 1] (One-to-One) の選択肢オプションがあります。[左ソースカラム] (Left Source Column) および [右ソースカラム] (Right Source Column) を使用して、JOIN のマッピングを作成し、サンプルデータを表示することができます。



データプロファイリング特性の表示

トピックス

シノニムまたはセグメントのデータプロファイリング
データプロファイリングフィールド

シノニム内のフィールドのデータ特性は、データプロファイリングとして表示することができます。シノニムまたはセグメント内のすべてのフィールドの特性を表示することも、特定のフィールドの特性を個別に表示することもできます。

文字フィールドの場合、データプロファイリングで生成される情報には、セグメント、フォーマット、件数(種類)、件数、パターン数、(長さの) 最大値、(長さの) 最小値、(長さ) 平均、NULL 数があります。パターン数には、文字フィールド別に検索されたパターン件数が表示されます。

数値フィールドの場合、データプロファイリングが提供する情報には、セグメント、フォーマット、件数(種類)、件数、パターン数、最大値、最小値、平均値、NULL 数があります。

各フィールドのデータプロファイリングからは、統計、パターン、値、異常値の各レポートへアクセスすることができます。

シノニムまたはセグメントのデータプロファイリング

手順/構文

シノニムまたはセグメントのデータプロファイリングを表示するには

データプロファイリングには、シノニムまたはセグメント内のすべてのフィールド情報が表示されます。また、シノニムまたはセグメントのデータプロファイリングレポートからは、個々のフィールドの値レポートまたはパターンレポートにドリルダウンすることができます。

Segment	Name	Format	Count	Distinct Count	Distinct Percent	Patterns Count	Minimum Value	Maximum Value	Minimum Length	Maximum Length	Average Length	Nulls Count	Nulls Percent	Duplicate Count	Duplicate Percent	
1	ORDERS	PRODUCTNUMBER	A4	32283	75	.23	1	1001	5015	4	4	4	0	.00	32208	99.77

手順

シノニムまたはセグメントのデータプロファイリングを表示するには

シノニムまたはセグメントのデータプロファイリング情報を表示するには、次の手順を実行します。

1. [プロジェクト] (Projects) エリアまたは [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。

デフォルト設定では、シノニムエディタは、前回使用した表示で開きます。

2. シノニムまたはセグメントの名前を右クリックし、[データプロファイリング] (Data Profiling)、[統計] (Statistics) を順に選択します。

ワークスペースにデータプロファイリング情報が表示されます。ここでは最後の4つのカラムが表示されています。それ以外の情報は説明のためのサンプルです。実際のレポートは、ワークスペースの横方向に表示されます。

データプロファイリングの結果のツールバーを使用し、サーバメッセージの表示、レポートの印刷、データのテキストとしてのコピー、レポートのエクスポートを実行することができます。

3. 必要に応じて、[名前](Name) または (文字フィールドの)[パターン数](Patterns Count) をクリックして、値レポートまたはパターンレポートにドリルダウンすることができます。

下図は、[名前](Name) をクリックして生成された値レポートの一部を示しています。

パターン解析では、「9」は数字、「A」は任意の大文字、「a」は任意の小文字を表します。表示可能なすべての特殊文字はそのまま表示されますが、表示不可の文字は「X」で表示されます。

データプロファイリングフィールド

手順/構文

データプロファイリングの統計を表示するには

データプロファイリングのパターンを表示するには

データプロファイリングの値を表示するには

データプロファイリングの値グラフを表示するには

データプロファイリングの円グラフを表示するには

データプロファイリングの重複値を表示するには

データプロファイリングの異常値を表示するには

データプロファイリングは、シノニム内のすべてのフィールド、セグメント、に対または個別のフィールドに対して実行することができます。Shift キーまたは Ctrl キーを押しながらフィールド選択すると、複数のフィールドを選択することができます。単一フィールドのデータプロファイリングでは、次の4つのレポートにアクセスすることができます。

- **統計** シノニムまたはセグメントのデータプロファイリングレポートと同一の情報を表示します。

文字フィールドの場合、統計レポートで生成される情報には、セグメント、フォーマット、件数(種類)、件数、パターン数、(長さの)最大値、(長さの)最小値、(長さ)平均、NULL 数があります。

数値フィールドの場合、統計レポートで生成される情報には、セグメント、フォーマット、件数(種類)、件数、パターン数、(長さの)最大値、(長さの)最小値、(長さ)平均値、NULL 数があります。

- **パターン** 文字フィールドに対してのみ使用することができます。文字、数字、特殊文字のパターンおよび件数とそのパーセントを表示します。
- **値** ユニーク値およびそのパーセントを表示します。
- **値グラフ** 文字フィールドのグラフを表示します。
- **値円グラフ** 文字フィールドの円グラフを表示します。
- **値の重複** 重複する値およびそのパーセントを表示します。
- **異常値** 上位および下位それぞれ 10 個の値とその件数を表示します。

これらのレポートを表示するには、シノニムエディタでフィールドの 1 つを右クリックし、[データプロファイリング] (Data Profiling) を選択します。

手順

データプロファイリングの統計を表示するには

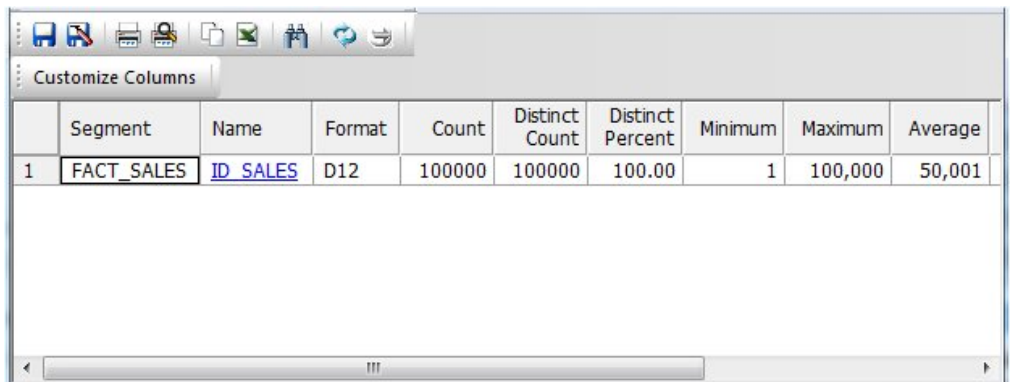
単一フィールドの統計データプロファイリング情報を表示するには、次の手順を実行します。

1. [プロジェクト] (Projects) エリアまたは [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。

デフォルト設定では、シノニムエディタは、前回使用した表示で開きます。

2. フィールドを右クリックし、[データプロファイリング] (Data Profiling)、[統計] (Statistics) を選択します。

下図のように、ワークスペースにデータプロファイリング統計情報が表示されます。



	Segment	Name	Format	Count	Distinct Count	Distinct Percent	Minimum	Maximum	Average
1	FACT_SALES	ID_SALES	D12	100000	100000	100.00	1	100,000	50,001

- 必要に応じて、[名前] (Name) または (文字フィールドの) [パターン数] (Patterns Count) をクリックして、値レポートまたはパターンレポートにドリルダウンすることができます。

手順

データプロファイリングのパターンを表示するには

データプロファイリングパターンには、文字、数字、特殊文字、件数が表示されます。これは、文字フィールドでのみ使用できます。

単一フィールドのパターンのデータプロファイリング情報を表示するには、次の手順を実行します。

- [プロジェクト] (Projects) エリアまたは [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。

デフォルト設定では、シノニムエディタは、前回使用した表示で開きます。

- フィールドを右クリックし、[データプロファイリング] (Data Profiling)、[パターン] (Patterns) を選択します。

下図のように、パターンデータプロファイリングが表示されます。

	Pattern	Count	Percent
1	Aaaaaa	21936	29.25
2	Aaaaa	19071	25.43
3	Aaaaaaa	13790	18.39
4	Aaaa	9556	12.74
5	Aaaaaaaa	5297	7.06
6	Aaa	2561	3.41
7	Aaaaaaaaaa	2178	2.90
8	Aaaaaaaaaa	352	.47
9	Aa	177	.24
10	Aaaaaaaaaa	82	.11

パターン解析では、「9」は数字、「A」は任意の大文字、「a」は任意の小文字を表します。表示可能なすべての特殊文字はそのまま表示されますが、表示不可の文字は「X」で表示されます。

手順

データプロファイリングの値を表示するには

データプロファイリング値には、ユニーク値が表示されます。

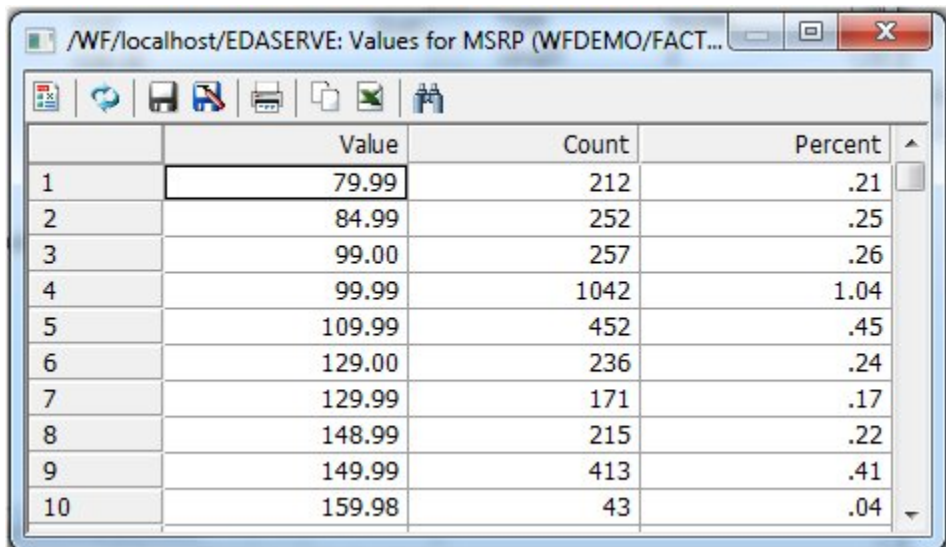
単一フィールドの値のデータプロファイリング情報を表示するには、次の手順を実行します。

1. [プロジェクト] (Projects) エリアまたは [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。

デフォルト設定では、シノニムエディタは、前回使用した表示で開きます。

2. フィールドを右クリックし、[データプロファイリング] (Data Profiling)、[値] (Values) を選択します。

下図のように、値データプロファイリング情報が表示されます。



The screenshot shows a window titled "/WF/localhost/EDASERVE: Values for MSRP (WFDEMO/FACT...". The window contains a table with the following data:

	Value	Count	Percent
1	79.99	212	.21
2	84.99	252	.25
3	99.00	257	.26
4	99.99	1042	1.04
5	109.99	452	.45
6	129.00	236	.24
7	129.99	171	.17
8	148.99	215	.22
9	149.99	413	.41
10	159.98	43	.04

手順

データプロファイリングの値グラフを表示するには

データプロファイリングの値グラフは、棒グラフとして表示されます。

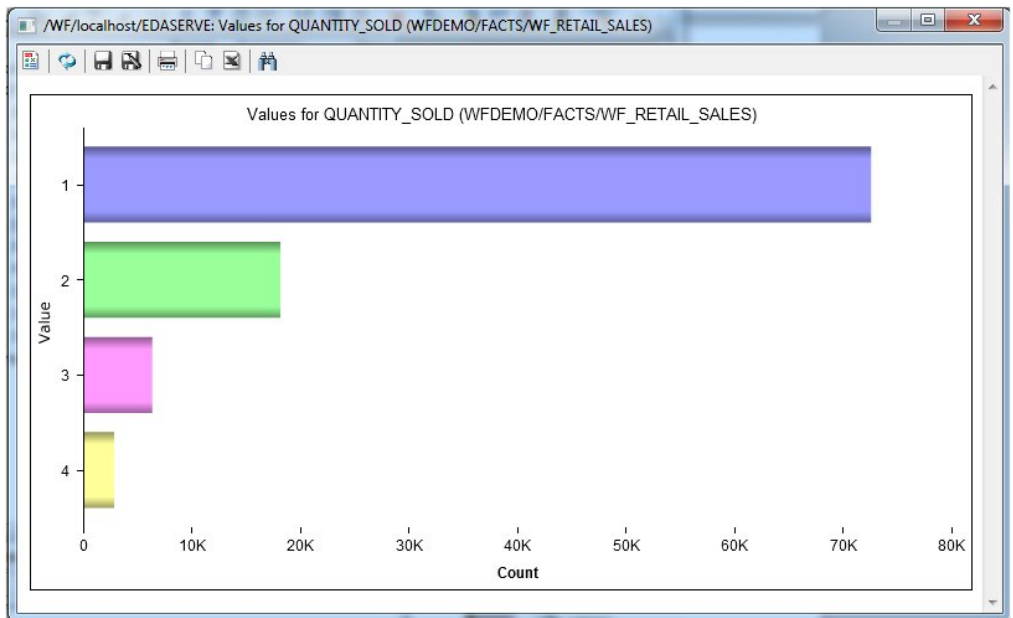
単一フィールドの値グラフのデータプロファイリング情報を表示するには、次の手順を実行します。

1. [プロジェクト] (Projects) エリアまたは [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。

デフォルト設定では、シノニムエディタは、前回使用した表示で開きます。

2. フィールドを右クリックし、[データプロファイリング] (Data Profiling)、[値グラフ] (Values Graph) を選択します。

下図のように、値グラフのデータプロファイリングが表示されます。



手順

データプロファイリングの円グラフを表示するには

データプロファイリングの値円グラフは、円グラフとして表示されます。

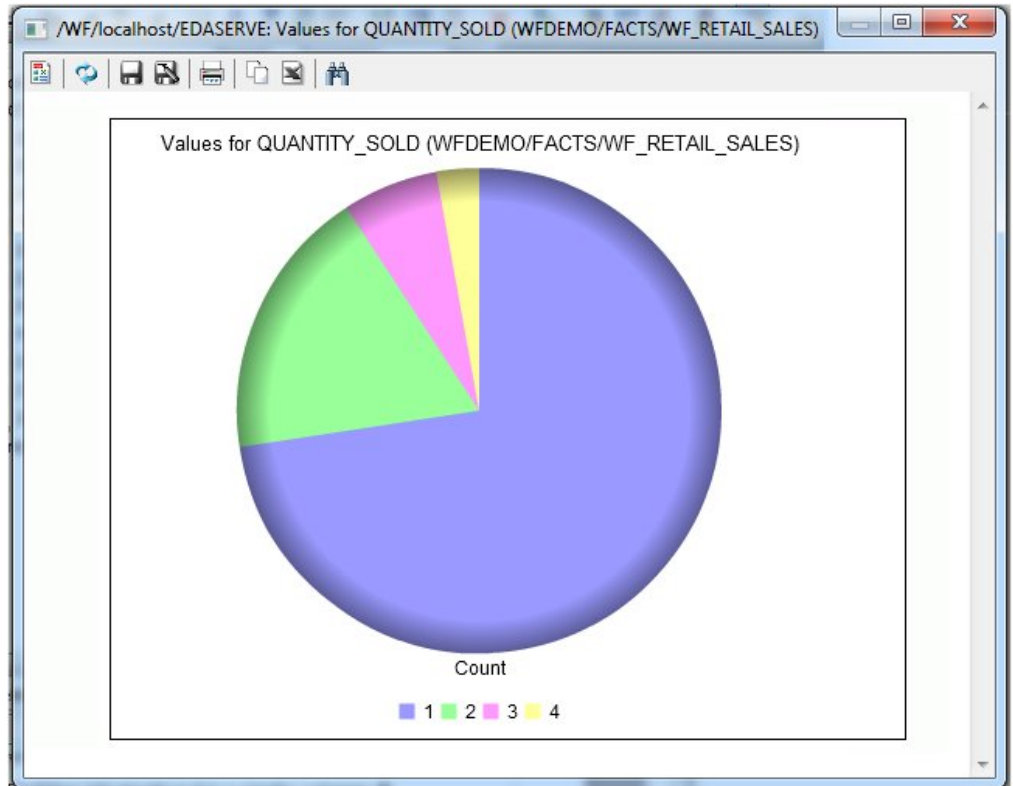
単一フィールドの値円グラフのデータプロファイリング情報を表示するには、次の手順を実行します。

1. [プロジェクト] (Projects) エリアまたは [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。

デフォルト設定では、シノニムエディタは、前回使用した表示で開きます。

2. フィールドを右クリックし、[データプロファイリング] (Data Profiling)、[値円グラフ] (Values Pie Graph) を選択します。

下図のように、値円グラフのデータプロファイリングが表示されます。



手順

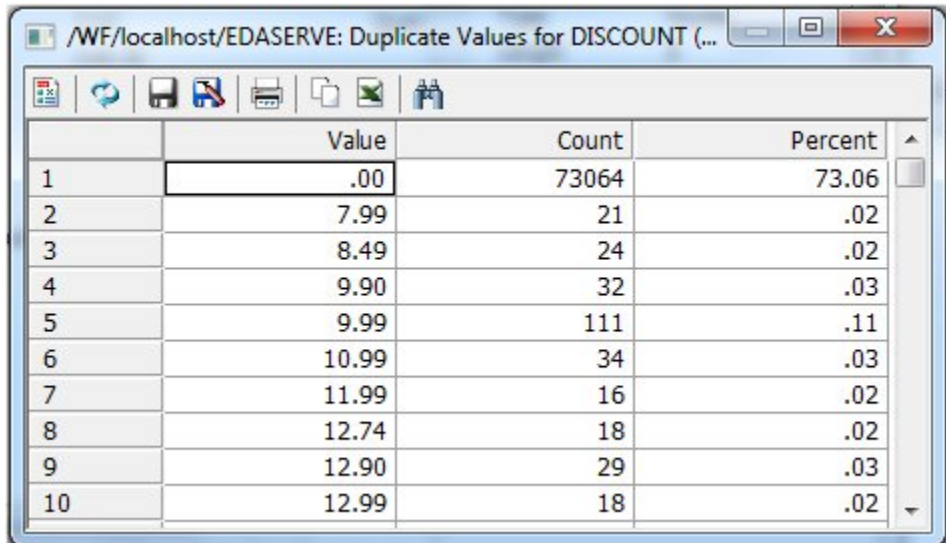
データプロファイリングの重複値を表示するには

[データプロファイリング] (Data Profile) の [値の重複] (Duplicate Values) には、重複する値が表示されます。

単一フィールドの重複値のデータプロファイリング情報を表示するには、次の手順を実行します。

1. [プロジェクト] (Projects) エリアまたは [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。
2. フィールドを右クリックし、[データプロファイリング] (Data Profiling)、[値の重複] (Duplicate Values) を選択します。

下図のように、重複する値のデータプロファイリングが表示されます。



	Value	Count	Percent
1	.00	73064	73.06
2	7.99	21	.02
3	8.49	24	.02
4	9.90	32	.03
5	9.99	111	.11
6	10.99	34	.03
7	11.99	16	.02
8	12.74	18	.02
9	12.90	29	.03
10	12.99	18	.02

手順

データプロファイリングの異常値を表示するには

データプロファイリングの異常値には、上位および下位それぞれ 10 個の DISTINCT 値が表示されます。

単一フィールドの異常値のデータプロファイリング情報を表示するには、次の手順を実行します。

1. [プロジェクト] (Projects) エリアまたは [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。

デフォルト設定では、シノニムエディタは、前回使用した表示で開きます。

2. フィールドを右クリックし、[データプロファイリング] (Data Profiling)、[異常値] (Outliers) を選択します。

下図のように、異常値のデータプロファイリングが表示されます。

	High	Count	Low	Count
1	15,999.40	89	79.99	212
2	15,996.00	6	84.99	252
3	13,996.00	13	99.00	257
4	13,596.00	10	99.99	1042
5	11,999.55	197	109.99	452
6	11,997.00	22	129.00	236
7	10,497.00	27	129.99	171
8	10,197.00	57	148.99	215
9	8,999.96	65	149.99	413
10	7,999.70	490	159.98	43

注意：異常値では、存在する場合、上位および下位それぞれ 10 個の DISTINCT 値が表示されます。

クラスタ JOIN の作成

手順/構文

既存シノニムを拡張してクラスタ JOIN を作成するには

新しいシノニムを使用してクラスタ JOIN を作成するには

クラスタ JOIN を使用すると、同一または混在データソースを使用した複数のリレーショナルテーブルの既存シノニムを結合することにより、新しいファイル構造を作成することができます。たとえば、DB2 と Oracle のテーブルを結合することなどできます。クラスタ JOIN を使用して物理テーブルにリンクすることにより、メタデータ内に新しいビューを作成し、作成した新しいビューおよび構造に対するレポートを容易に作成することができます。クラスタ JOIN は、シノニムエディタのモデル表示を使用して作成することができます。

作成されるマスターファイルでは、結合されたテーブルのフィールドが 1 つのファイルにまとめられます。結合されたファイルのアクセスファイルには、データソースの実際の場所と JOIN 情報が格納されます。また、テーブルのリンク状況も表示されます。

ツールに追加できるテーブル数の合計は 64 (JOIN を 63 個使用) であり、結果として最大 64 個のセグメントが含まれた新しいマスターファイルが作成されます。

クラスタ JOIN ツールを使用して、1つのファクトテーブルから多数のディメンションテーブルを参照するスタースキーマを作成することができます。必要に応じて、複数のファクトテーブルで構成されるファクトテーブルのビューを作成することもできます。

手順

既存シノニムを拡張してクラスタ JOIN を作成するには

セグメントを追加することにより、既存シノニムを拡張するには、モデル表示を使用します。

1. [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。

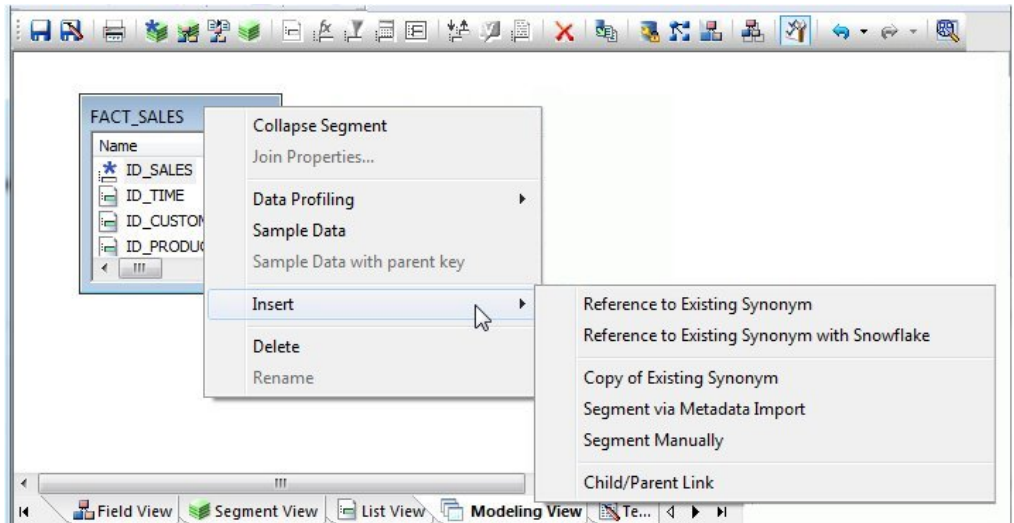
シノニムエディタでシノニムが開きます。

2. [モデル表示] (Modeling View) タブをクリックします。

ワークスペースにシノニムエディタの [モデル表示] (Modeling View) タブが表示されます。

3. ワークスペースでセグメントを右クリックし、[挿入] (Insert) を選択します。

下図のように、コンテキストメニューが表示されます。

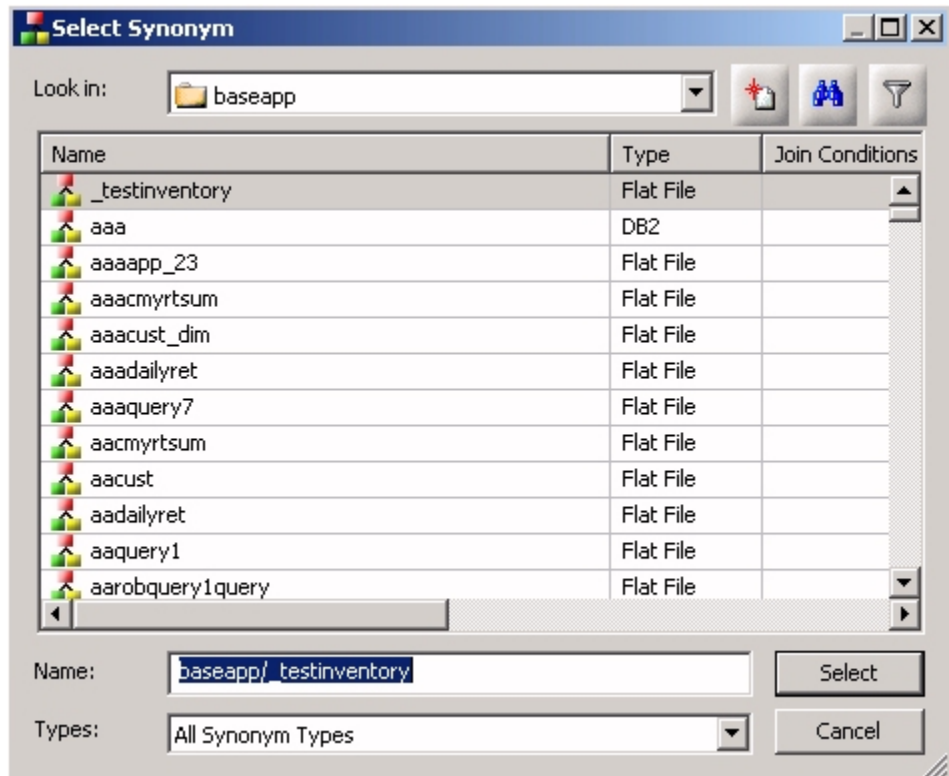


4. 次のいずれかの方法でテーブル (セグメント) を挿入します。

既存シノニムからセグメントを挿入するには、次の手順を実行します。

- a. [挿入] (Insert)、[既存シノニムのセグメントを参照] (Reference to Existing Synonym) を選択します。

下図のように、[シノニムの選択] (Select Synonym) ダイアログボックスが開きます。



- b. 挿入するシノニムを選択し、[選択] (Select) をクリックします。

注意：既存のテーブルまたはシノニムでクラスタ JOIN を作成する場合、この方法を使用します。

ヒント：元のシノニムを変更しない場合、[モデル表示] (Modeling View) の [ファイル] (File) メニューから [名前を付けて保存] (Save As) を選択します。

既存のスノーflakeシノニムからセグメントを挿入するには、次の手順を実行します。

[挿入] (Insert)、[既存スノーflakeシノニムの参照] (Reference to Existing Synonym with Snowflake) を順に選択します。

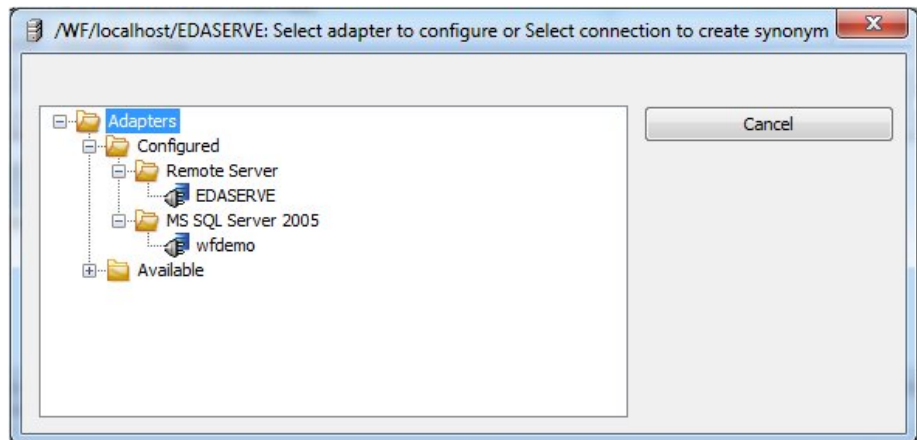
この手順は、既存シノニムの参照を挿入する場合と同様ですが、選択リストに候補のシノニムと参照シノニムが表示される点が異なります。

インポートしたメタデータからセグメントを挿入するには、次の手順を実行します。

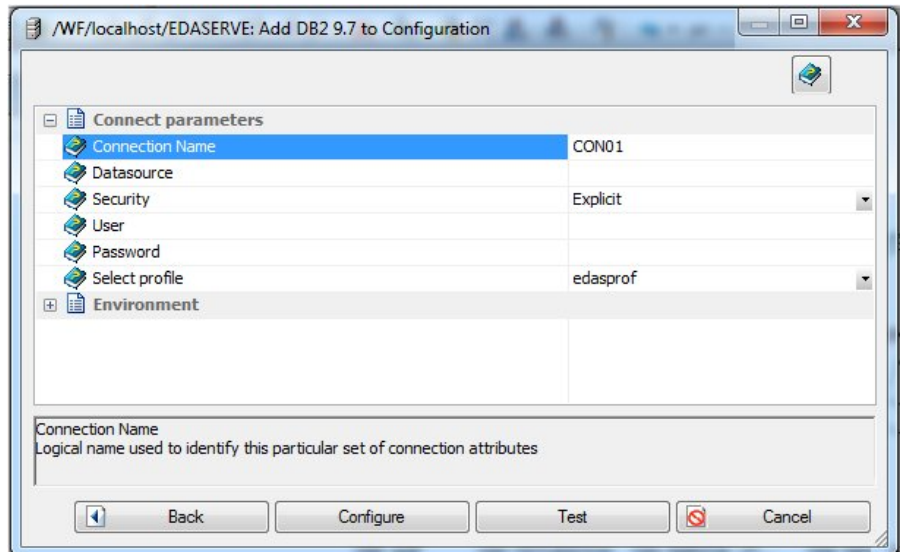
- a. [挿入] (Insert)、[インポートメタデータのセグメント] (Segment via Metadata Import) を選択します。これにより、[シノニムの作成] (Create Synonym) ツールを使用したセグメントの追加が可能になります。このツールは、シノニムを作成し、ツールを起動したシノニム内のセグメントとして追加します。

注意：クラスタ JOIN を作成し、存在しないシノニムを使用する必要がある場合、この方法を使用します。このオプションにより、シノニムを作成後、クラスタ JOIN の作成を続行することができます。

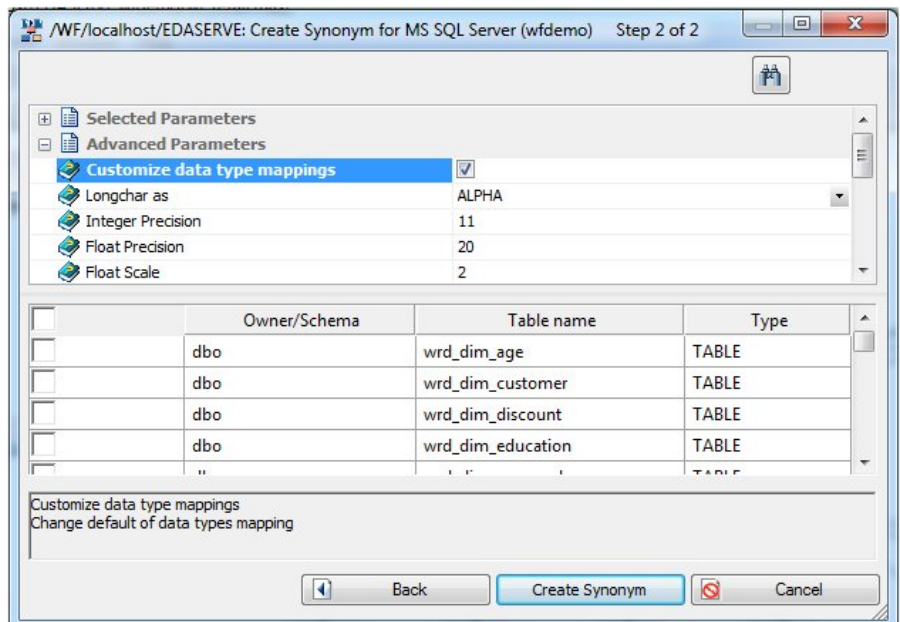
- b. このオプションを選択した場合、最初に [アダプタ] (Adapter) ダイアログボックスが表示されます。ここで、構成済みのアダプタ接続を選択して次へ進むことも、必要に応じて新しいアダプタを構成することもできます。下図は、[アダプタ] (Adapter) ダイアログボックスを示しています。



下図は、表示される画面の例を示しています。ここで、接続パラメータの情報を入力します。



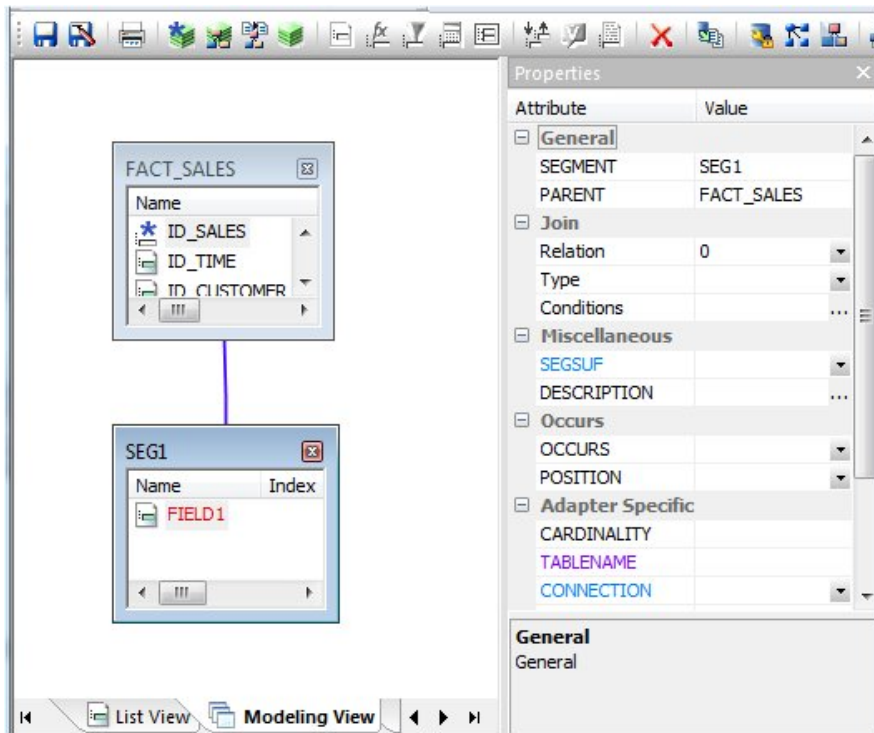
下図は、シノニムを作成するテーブルを選択する画面の例を示しています。



選択したシノニムが作成され、モデル表示に追加されます。

セグメントを手動で挿入するには、次の手順を実行します。

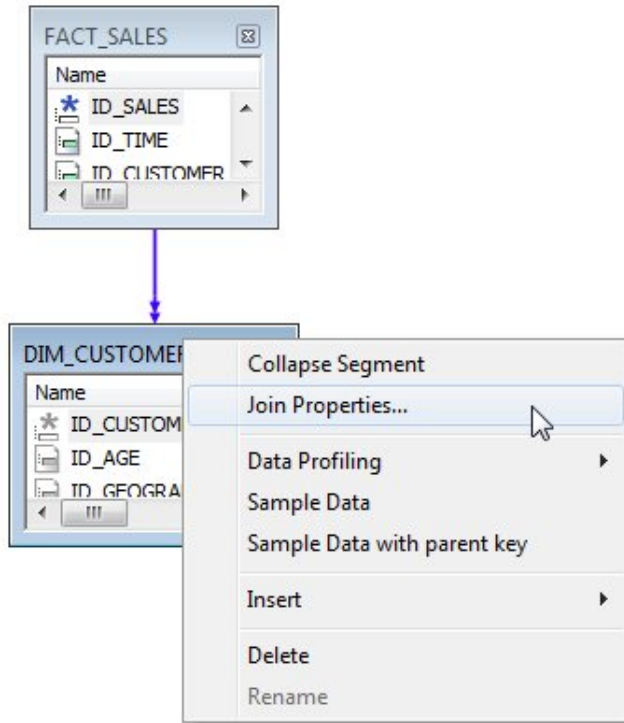
[挿入] (Insert)、[手動でセグメントを追加] (Segment Manually) を選択します。これにより、シノニムエディタでセグメント属性フィールドに値を割り当てることが可能になります。



注意： Web Query データソースの場合のように、マスターファイルを最初から記述する場合は、この方法を使用します。

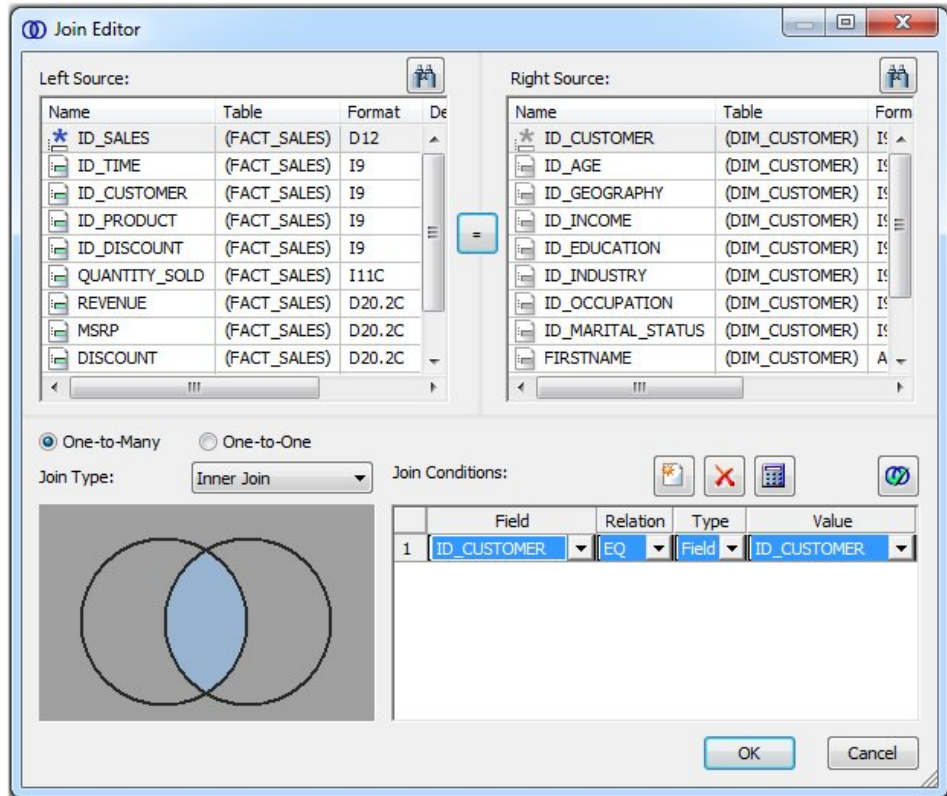
モデル表示にシノニムが追加されます。

5. 下図のように、セグメントを右クリックし、[JOIN のプロパティ] (Join Properties) を選択します。



注意：このオプションは、リレーショナルテーブルを使用する場合にのみ使用可能です。

下図のような [JOIN エディタ] (Join Editor) ダイアログボックスが表示されます。



6. [1 対 n] (One-to-Many) (JOIN ALL) または [1 対 1] (One-to-One) (JOIN UNIQUE) のラジオボタンを選択します。
 - **1 対 n (One-to-Many)** 複数インスタンスの JOIN を示します。実行時に、各ホストレコードには、クロスリファレンスファイルの一致レコードを多数含めることができます。
 - **1 対 1 (One-to-One)** 単一インスタンスの JOIN を示します。実行時に、各ホストレコードには、クロスリファレンスファイルの一致レコードを 1 つのみ含めることができます。
7. ドロップダウンリストから、[JOIN タイプ] (Join Type) の 1 つを選択します。選択肢には、[INNER JOIN] (Inner)、[OUTER JOIN] (Outer)、[CROSS JOIN] (Cross Join) があります。

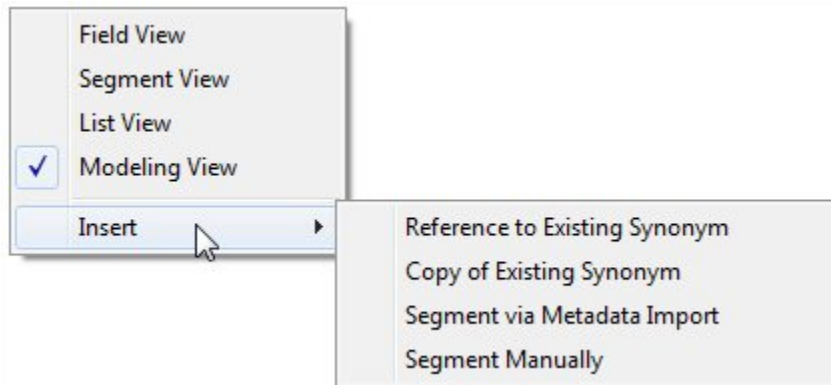
両方のセグメントに同一のフィールドが存在する場合、[JOIN 条件] (Join Condition) フィールドに JOIN が自動的に作成されます。

手順

新しいシノニムを使用してクラスタ JOIN を作成するには

クラスタ JOIN を作成するには、ブランクのシノニムから開始する方法もあります。

1. [マスターファイル] (Master Files) フォルダを右クリックし、[新規作成] (New)、[シノニムエディタでシノニムを編集] (Synonym via Synonym Editor) を選択します。
[新規マスターファイル] ダイアログボックスが開きます。
2. [ファイル名] (File Name) テキストボックスに一意のファイル名を入力します。
3. [作成] (Create) をクリックします。
シノニムエディタでシノニムが開きます。
4. [モデル表示] (Modeling View) タブをクリックします。
5. ワークスペース内を右クリックし、利用可能なオプションのいずれかを選択して新しいビューの作成を開始します。



OLAP 分析のディメンション定義

トピックス

シノニムエディタモデル表示でのディメンションビルダの使用

シノニムを変更し、OLAP (オンライン分析処理) を有効にすることができます。シノニムエディタには、OLAP 階層およびディメンションを作成するツールが用意されています。OLAP を使用すると、階層データのドリルダウンや集約、列から行 (またはその逆) へのフィールドのピボット、指定された条件としきい値に基づくデータソースのフィルタ設定およびクエリによる情報の選別が可能になります。

シノニムエディタを使用してマスターファイルの OLAP を有効にし、フィールドレベルでディメンションを作成後、各ディメンションにフィールドを関連付けます。

注意：OLAP はレポート機能であり、データ保守プロジェクトとは関係ありません。

シノニムエディタモデル表示でのディメンションビルダの使用

手順/構文

親子階層を追加するには

ディメンションを削除するには

レベル階層を追加するには

参照


ディメンションビルダツールバー

ディメンションビルダを使用すると、メタデータを手動で編集することなく、エンタープライズデータ (リレーショナルデータソースまたはレガシーデータソース) に基づいて、マルチディメンション分析のための論理ビューを作成することができます。ディメンションビルダはリレーショナルおよび非リレーショナルデータソースで動作します。ディメンションビルダを有効にするには、ツールバーまたは [ツール] (Tools) メニューから、[ディメンションビルダ] (Dimension Builder) を選択します。


手順

親子階層を追加するには

1. シノニムエディタで [モデル表示] (Modeling View) タブをクリックします。

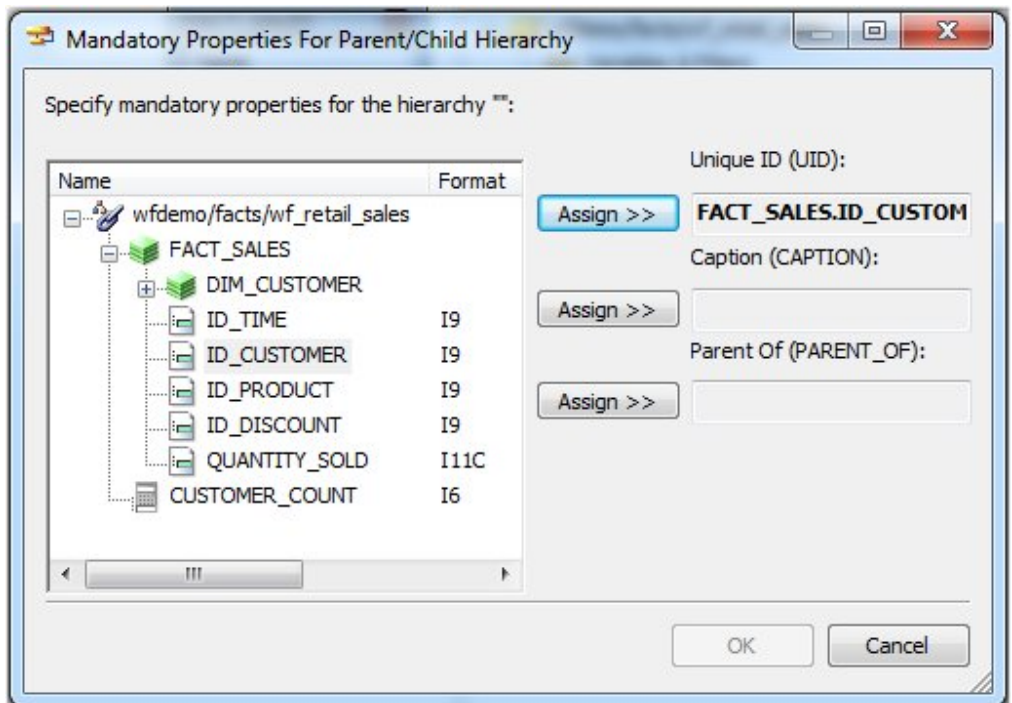
2. ツールバーの [ディメンションビルダ] (Dimension Builder)  アイコンをクリックします。

ディメンションビルダが開きます。

3. [ディメンションビルダ] (Dimension Builder) ツールバーの [従属階層] (Parent/Child)  アイコンをクリックします。

[従属階層の必須プロパティ] (Mandatory Properties for Parent/Child Hierarchy) ダイアログボックスが開きます。

4. フィールドツリー からフィールドを選択して [割り当て] (Assign) をクリックし、[ユニーク ID (UID)] (Unique ID) を割り当てます。



5. [キャプション (CAPTION)] および [従属元 (PARENT_OF)] に対しても、手順 4 を繰り返します。

6. [OK] をクリックし、[従属階層の必須プロパティ] (Mandatory Properties for Parent/Child Hierarchy) ダイアログボックスを閉じます。

ディメンションビルダに必須プロパティが追加されます。

7. コンテキストメニューを使用し、ディメンション名の変更、プロパティの表示、またはディメンションのサンプルデータの表示を行います。
8. [ファイル] (File) メニューから [保存] (Save) を選択し、ディメンションを保存します。

手順

ディメンションを削除するには

ディメンションを削除するには、ディメンションの 1 つを右クリックして [削除] (Delete) を選択するか、ディメンションビルダのツールバーの [削除] (Delete) ボタンをクリックします。

手順

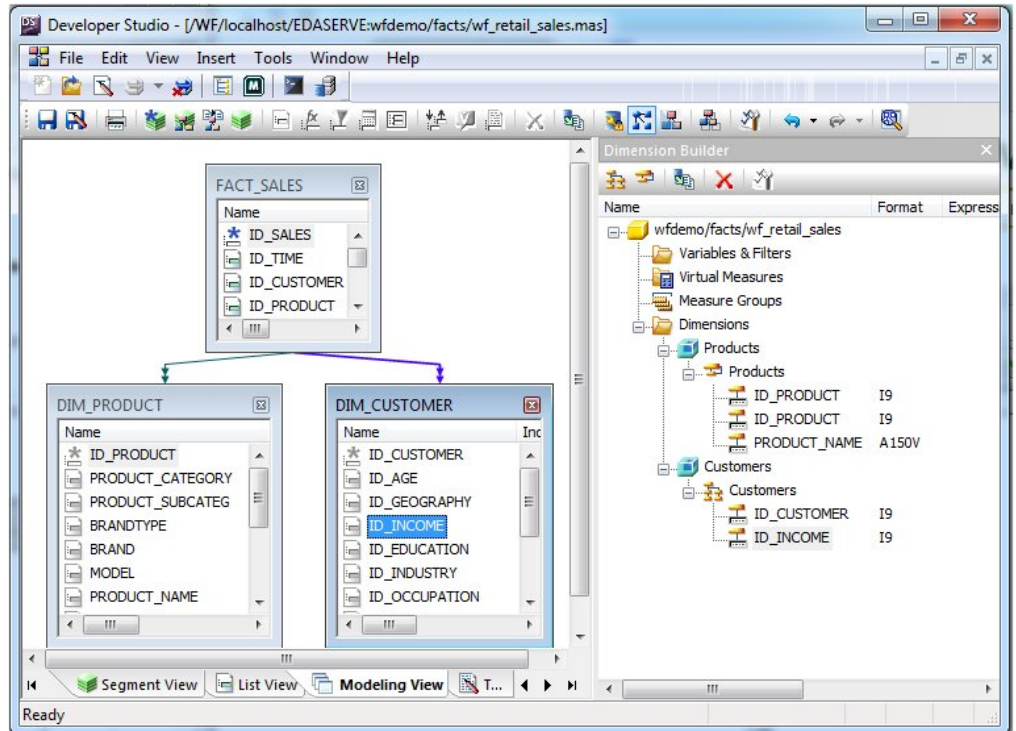
レベル階層を追加するには

この手順により、既存のマスターファイルの編集、テーブルの追加、およびディメンションの生成と変更をすることができます。

1. シノニムエディタで [モデル表示] (Modeling View) タブをクリックします。
2. ディメンションビルダのツールバーから [レベル階層の追加] (Levels Hierarchy) ボタンを選択します。

階層にレベルが追加されます。コンテキストメニューを使用してディメンション名を変更するか、デフォルト名をそのまま使用します。


3. セグメントでフィールドを選択し、ディメンションビルダの階層フォルダにドラッグします。







4. [ファイル](File) メニューから [保存](Save) を選択し、ディメンションを保存します。マスターファイルにディメンションが保存されます。

参照 **ディメンションビルダツールバー**

ディメンションビルダのツールバーからは、次のコマンドにアクセスすることができます。

ボタン	定義
	レベル階層を追加します。

ボタン	定義
	親子階層を作成し、階層に必須プロパティを割り当てることができます。
	選択項目を削除します。
	選択フィールドのサンプルデータを表示、更新することができます。
	階層を編集することができます。

ビジネスビューの作成

手順/構文

シノニムエディタを使用してビジネスビューを作成するには
シノニムエディタを使用してビジネスビューを作成するには (代替方法)

参照

ビジネスビュー使用上の注意
ビジネスビューマスターファイルの使用

マスターファイルのビジネスビューを定義すると、マスターファイルの代替ビューを作成して、使用可能なフィールドを制限することや、元のマスターファイルから、フィールドのサブセットを作成することができます。フィールドをフォルダにグループ化し、ビジネスビューごとにフィールド名、タイトル、説明をカスタマイズすることができます。

ビジネスビューのフィールドはフォルダ単位で分類されます。各フォルダには、フィールド群が格納されます。フォルダ内のフィールドは、元のマスターファイルの異なるセグメントから取得することもできます。ビジネスビューには既存のフィールドを格納することができるほか、一時項目 (DEFINE)、一時項目 (COMPUTE)、フィルタに既存のカスタムフィールドを含めることもできます。カスタムフィールドは、元のマスターファイルの特定のセグメントに関連付けられ、実フィールドと同一の規則が適用されます。元のマスターファイルの単一パスに存在する場合、複数のフォルダからのフィールドをレポートで参照することができます。

シノニムエディタでマスターファイルを開き、ツールバーの [ビジネスビュー] (Business View) ボタンをクリックすると、シノニムエディタ内に [ビジネスビュー] (Business View) ウィンドウが開きます。このウィンドウでは、ビジネスビューの属性を追加することができます。この時点でビジネスビューの属性を追加すると、マスターファイルの末尾にビジネスビューの属性が追加されます。このファイルを保存すると、ファイルをレポート作成に使用する際に、ビジネスビューはアクティブビューになります。

マスターファイルをビジネスビューに変更したくない場合は、ビジネスビューを別名で保存することをお勧めします。または、より推奨される方法として、シノニムエディタのオプションで [新規作成] (New)、[シノニム] (Synonym) を選択し、新しいマスターファイルシノニムを作成する方法があります。さらに、既存シノニムのセグメントの参照を挿入し、ビジネスビューの作成を続行します。この場合、ビジネスビューは、クロスリファレンスのマスターファイルを参照し、ツール (JOIN や DEFINE など) およびレポートで使用される際は、実フィールドは、すべて参照ファイルから取得されます。

注意：インパクト分析では、DB2 Web Query プロシジャのほかに、ビジネスビューも検索されます。インパクト分析により、元のマスターファイルの変更が、ビジネスビューに使用されるフィールドに影響を与えるかどうか表示することができます。インパクト分析についての詳細は、213 ページの「[メタデータとプロシジャの分析](#)」を参照してください。

既存のマスターファイルのビジネスビューは、シノニムエディタを使用して作成することができます。

手順

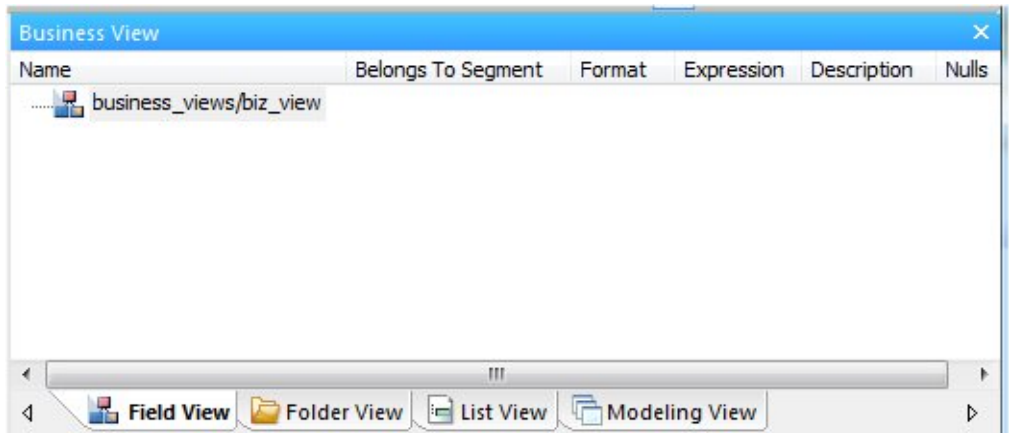
シノニムエディタを使用してビジネスビューを作成するには

1. [データサーバ] (Data Servers) エリアから、ビジネスビューを作成する [マスターファイル] (Master Files) フォルダに移動します。
2. [マスターファイル] (Master Files) フォルダを右クリックし、[新規作成] (New)、[シノニムエディタでシノニムを編集] (Synonym via Synonym Editor) を選択します。
3. 新しいファイルの一意の名前を指定し、[開く] (Open) ([データサーバ] (Data Servers) エリアの場合は [作成] (Create)) をクリックします。
4. シノニムエディタの [セグメントフィールド表示] (Field View) タブで、左上隅のファイル名を右クリックし、[挿入] (Insert)、[既存シノニムのセグメントを参照] (Reference to Existing Synonym) を選択します。
5. [既存シノニムの参照を挿入] (Insert Reference to Existing Synonym) ダイアログボックスから、ビジネスビューを作成するシノニムを選択します。

参照されたファイルが新しいシノニムに追加され、そのフィールドが左側のフレームに表示されます。

6. [ツール] (Tools) メニューから [ビジネスビュー] (Business View) を選択し、[ビジネスビュー] (Business View) ウィンドウを開きます。シノニムエディタで [ビジネスビュー] (Business View) アイコンをクリックすることもできます。

注意：ビジネスビューマスターファイルには、ルートフォルダは1つだけ指定することができます。



ヒント：[ツール] (Tools) メニューから [プロパティ] (Properties) を選択して [プロパティ] (Properties) ウィンドウを開き、[ビジネスビュー] (Business View) ウィンドウで選択した項目の追加情報を表示することができます。[プロパティ] (Properties) ウィンドウを使用して、タイトル、説明、またはフィールド名を変更します。灰色表示されている項目を編集することはできません。

7. [ビジネスビュー] (Business View) ウィンドウでファイル名を右クリックし、[デフォルトビジネスビューの作成] (Create Default Business View) または [新規フォルダ] (New Folder) を選択します。

[デフォルトビジネスビューの作成] (Create Default Business View) オプションは、マスターファイルに存在するセグメントとフィールドの複製を作成します。必要に応じてビューを再編成することができます。[新規フォルダ] (New Folder) オプションは、ルートフォルダを作成します。左側ウィンドウでフィールドを選択してこのフォルダ内にドラッグすると、マスターファイルのフィールドを追加することができます。追加のフォルダを作成し、必要な構造を作成することもできます。

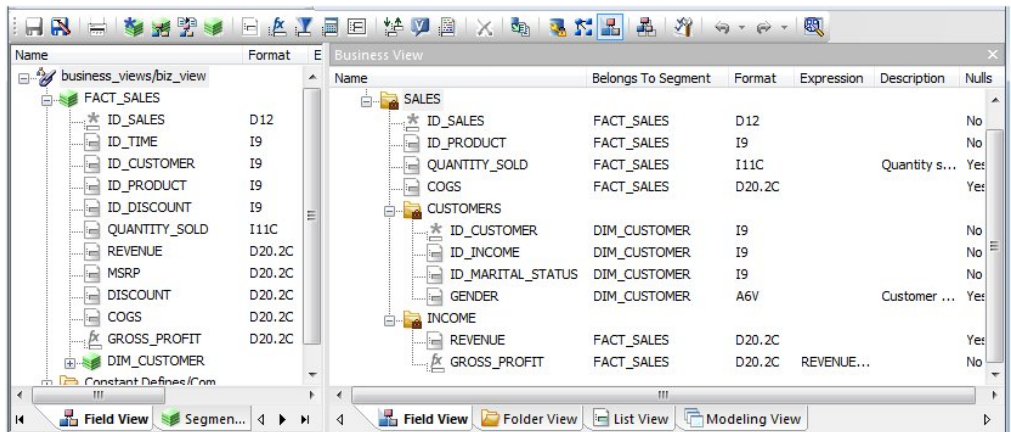
8. ビジネスビューに追加のフォルダを追加するには、[ビジネスビュー] (Business View) ウィンドウで作成されたビジネスビューのルートフォルダを右クリックし、[新規フォルダ] (New Folder) を選択します。

注意：複数のサブフォルダを作成し、作成したフォルダは、編成の目的でブランクにしておくことができます。

9. 左側ウィンドウでマスターファイルからフィールドを選択し、[ビジネスビュー] (Business View) ウィンドウの適切なフォルダにドラッグします。Shift キーまたは Ctrl キーを押しながら、複数のフィールドを選択します。

注意：必要な場合は、複数のフォルダに重複するフィールドを格納することができますが、各フォルダに格納可能な同一フィールドは、1 つだけです。

選択したフィールドが [ビジネスビュー] (Business View) ウィンドウに表示されます。



10. [保存] (Save) または [名前を付けて保存] (Save As) アイコンを選択し、ビジネスビューマスターファイルとしてビジネスビューを保存します。

注意：ビジネスビューマスターファイルは、元のマスターファイルとは別のアプリケーションに保存することができます。

手順

シノニムエディタを使用してビジネスビューを作成するには (代替方法)

シノニムエディタを使用してビジネスビューを作成する場合、159 ページの「シノニムエディタを使用してビジネスビューを作成するには」の方法に従うことが推奨されます。ここでは、シノニムエディタを使用してビジネスビューを作成する別の方法について説明します。

1. [データサーバ] (Data Servers) エリアで、ビジネスビューを作成する [マスターファイル] (Master Files) フォルダに移動し、変更するマスターファイルを選択します。

注意：ビジネスビューは、マスターファイルが選択可能な任意の場所で作成することができます。

2. マスターファイルをダブルクリックするか、[ファイル] (File) メニューから [シノニムエディタで編集] (Edit in Synonym Editor) を選択します。

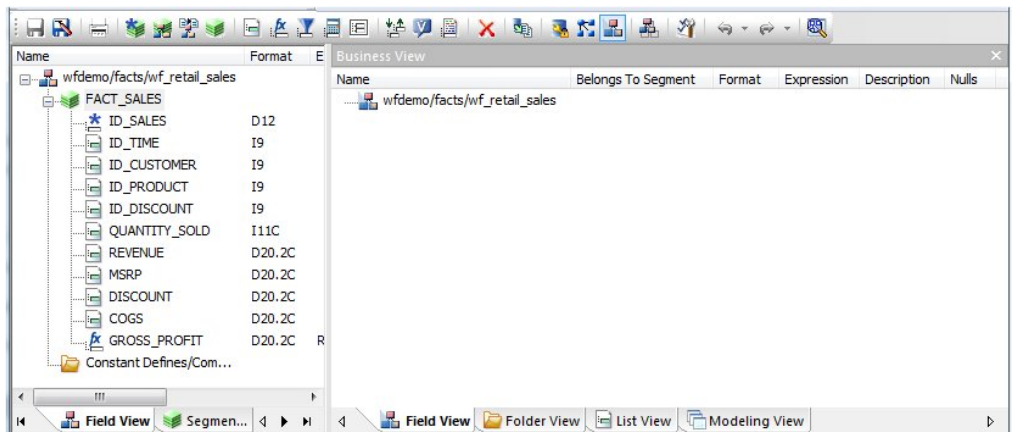
シノニムエディタの [フィールド表示] (Field View) タブに、マスターファイルが開きます。

注意：シノニムエディタでマスターファイルを開く場合、前回アクセスしたときに最後に選択されたタブが開きます。

3. [ツール] (Tools) メニューから [ビジネスビューの作成] (Business View) を選択します。シノニムエディタで [ビジネスビュー] (Business View) アイコンをクリックすることもできます。

[ビジネスビュー] (Business View) ウィンドウが開きます。

注意：ビジネスビューマスターファイルには、ルートフォルダは1つだけ指定することができます。



ヒント：[ツール] (Tools) メニューから [プロパティ] (Properties) を選択して [プロパティ] (Properties) ウィンドウを開き、[ビジネスビュー] (Business View) ウィンドウで選択した項目の追加情報を表示することができます。[プロパティ] (Properties) ウィンドウを使用して、タイトル、説明、またはフィールド名を変更します。灰色表示されている項目を編集することはできません。

4. [ビジネスビュー] (Business View) ウィンドウでファイル名を右クリックし、[デフォルトビジネスビューの作成] (Create Default Business View) または [新規フォルダ] (New Folder) を選択します。

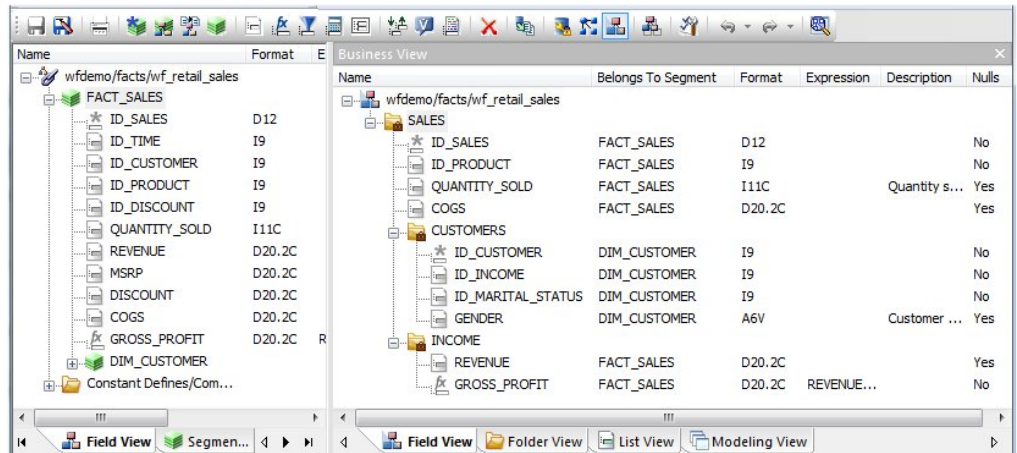
[デフォルトビジネスビューの作成] (Create Default Business View) オプションは、マスターファイルに存在するセグメントとフィールドの複製を作成するため、必要に応じてビューを再編成することができます。[新規フォルダ] (New Folder) オプションは、ルートフォルダを作成します。左側ウィンドウでフィールドを選択してこのフォルダ内にドラッグアンドドロップすると、マスターファイルのフィールドを追加することができます。追加のフォルダを作成し、必要な構造を作成することもできます。

5. ビジネスビューに追加のフォルダを追加するには、[ビジネスビュー] (Business View) ウィンドウで作成されたビジネスビューのルートフォルダを右クリックし、[新規フォルダ] (New Folder) を選択します。

注意：複数のサブフォルダを作成し、作成したフォルダは、編成の目的でブランクにしておくことができます。

6. 左側ウィンドウでマスターファイルからフィールドを選択し、[ビジネスビュー] (Business View) ウィンドウの適切なフォルダにドラッグアンドドロップします。Shift キーまたは Ctrl キーを押しながら、複数のフィールドを選択します。

選択したフィールドが [ビジネスビューツリー] (Business View Tree) タブに表示されます。



注意：必要な場合は、複数のフォルダに重複するフィールドを格納することができますが、各フォルダに格納可能な同一フィールドは、1 つだけです。

7. [保存] (Save) または [名前を付けて保存] (Save As) アイコンを選択し、ビジネスビューマスターファイルとしてビジネスビューを保存します。

[保存] (Save) オプションを使用すると、ビジネスビューは、現在のマスターファイル内に保存され、このマスターファイルをレポートツールで開いた場合に、ビジネスビューが表示されます。

注意：ビジネスビューマスターファイルは、元のマスターファイルとは別のアプリケーションに保存することができます。

8. [ファイル] (File) メニューから [閉じる] (Close) を選択し、シノニムエディタを閉じます。

参照

ビジネスビュー使用上の注意

- 参照マスターファイルを使用してビジネスビューを作成する際は、次のことに注意してください。

- USAGE および ACTUAL フォーマット、インデックスなどの詳細情報は、参照マスターファイルで保持されます。
- クラスタマスターファイルの情報は、すべて参照マスターファイルで保持されません。
- 参照マスターファイルで指定される DBA 属性は、ビジネスビューで使用されます。
- 複数のファイルを結合した場合など、マスターファイルに記述した複数のフィールドが同一名の場合、ビジネスビューで参照されているフィールド名のインスタンスは BELONGS_TO_SEGMENT 属性で識別されます。
- フォルダは分類の目的でブランクにすることができます。たとえば、Region に「North」、「South」、「East」、「West」と呼ばれるブランクのフォルダを含めることができます。
- ビジネスビューに対して SQL SELECT コマンドを発行することができます。ただし、ダイレクト SQL パススルーリクエストはサポートされません。
- ビジネスビューは、代替ファイルビューおよび完全修飾フィールド名をサポートしません。
- ビジネスビューフォルダに対する SEG. 演算子は、実セグメントのすべてのフィールドではなく、そのフォルダのすべてのフィールドを表示します。
- ビジネスビューに対するリクエストで、ビジネスビュー内に存在しないフィールドやセグメントを参照することはできません。
- すべての HOLD フォーマットは、ビジネスビューでサポートされます。
- すべてのアダプタは、ビジネスビューに対する検索リクエストをサポートします。
- ビジネスビューは、MAINTAIN や REBUILD などのデータソース保守コマンドではサポートされません。
- 参照マスターファイル、つまりビジネスビューの作成で現在使用中のマスターファイルには、一時項目 (DEFINE)、一時項目 (COMPUTE)、フィルタ、その他のフィールドを含めることができます。

参照

ビジネスビューマスターファイルの使用

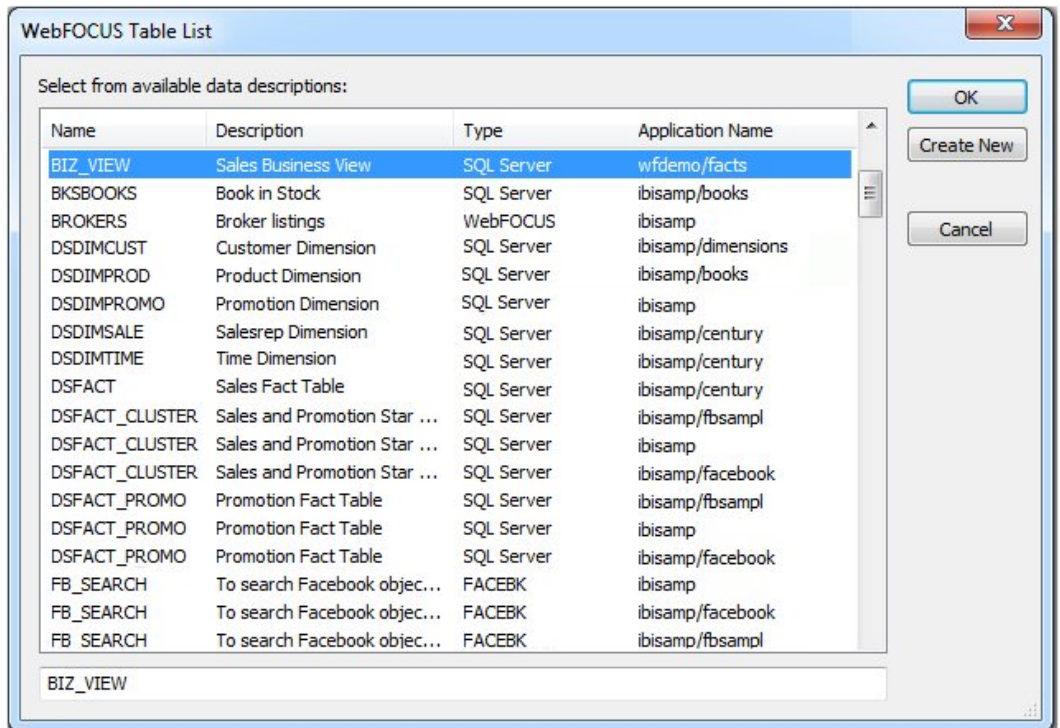
DB2 Web Query ツールおよびレポートでビジネスビューマスターファイルを使用する際は、タイトルと説明をビジネスビューから取得する設定を行わない場合、フィールドフォーマット、説明、タイトルは、元のマスターファイルから取得されます。

Developer Workbench のすべてのエリアからの ビジネスビューマスターファイルにアクセスすることができます。

データサーバ (Data Server)、またはリポジトリ (Repository) エリア

マスターファイルの一覧には、ビジネスビューを含む、利用可能なマスターファイルすべてが表示されます。[説明] (Description) 列には、マスターファイルまたはビジネスビューファイルの説明が表示されます。

下図は、マスターファイル一覧に表示されるビジネスビューマスターファイルの例を示しています。これは [データサーバ] (Data Servers) エリアでレポートを作成する際に表示されます。



これはプロジェクトエリアでも利用できますが、プロジェクトエリアの場合は、マスターファイルの一覧に備考は表示されません。

InfoAssist

InfoAssist でビジネスビューを使用する場合は、フィールドリストにはビジネスビューのフィールドのみが表示されます。

下図は、InfoAssist のビジネスビューマスターファイルの例を示しています。

シノニムへの一時項目 (DEFINE) の追加

手順/構文

シノニムの一時項目 (DEFINE) を作成するには

シノニムエディタのカスタムフィールドとして、一時項目 (DEFINE) を作成することができます。カスタムフィールドをリクエストで使用する場合、データソースの実フィールドのように扱うことができます。一時項目 (DEFINE) を使用できるのは、データソースを使用してレポートを作成する場合に限られます。

一時項目 (DEFINE) には、式、定数、フィールド名を含めることができます。

- 一時項目 (DEFINE) の式が複雑になる場合、[一時項目の演算] (Virtual Column Calculator) を実行して演算を作成するか、この [式] (Expression) テキストボックスに直接式を入力します。
- 一時項目 (DEFINE) の式が定数値 1 つのように単純である場合、[式] (Expression) テキストボックスに直接入力します。

一時項目 (DEFINE) は、次のアイコンにより識別できます。



一時項目 (DEFINE) の作成後、シノニムを右クリックし、[サンプルデータ] (Sample Data) を選択することでテストすることができます。別のダイアログボックスに、サンプルデータが表示されます。

手順

シノニムの一時項目 (DEFINE) を作成するには

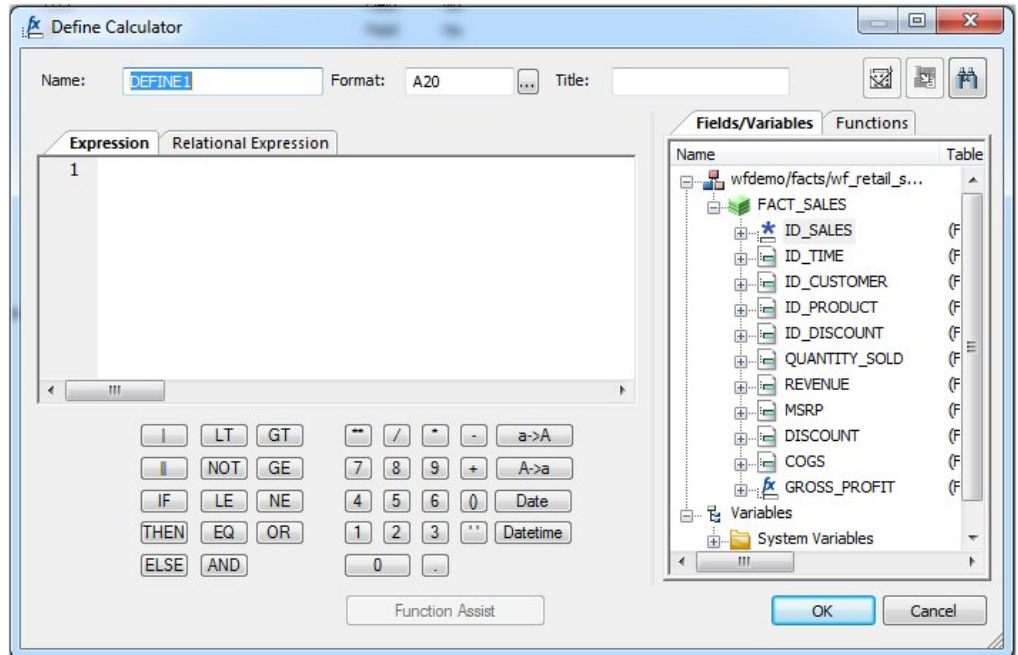
シノニムの一時項目 (DEFINE) を作成するには、次の手順を実行します。

1. [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。

シノニムエディタでシノニムが開きます。

2. セグメント (またはフィールド) を右クリックし、[挿入] (Insert)、[一時項目 (DEFINE)] (Define) を選択します。

下図のように、[一時項目 (DEFINE) の演算] (Define Calculator) が開きます。




3. [一時項目 (DEFINE) の演算] (Define Calculator) の [名前] (Name) テキストボックスにフィールド名を入力するか、デフォルトの DEFINE 名を使用します。
4. [タイトル] (Title) テキストボックスに、一時項目 (DEFINE) のタイトルを入力することもできます。

ヒント：シノニムエディタでは、[TITLE] および [DESCRIPTION] の [値] (Value) テキストボックス右側にある [...] (参照) ボタンをクリックし、多言語のタイトルを指定します。

5. [式] (Expression) タブおよび演算ボタンを使用し、一時項目 (DEFINE) の式を作成します。

または

[関係式] (Relational Expression) タブを使用し、条件式を作成します。

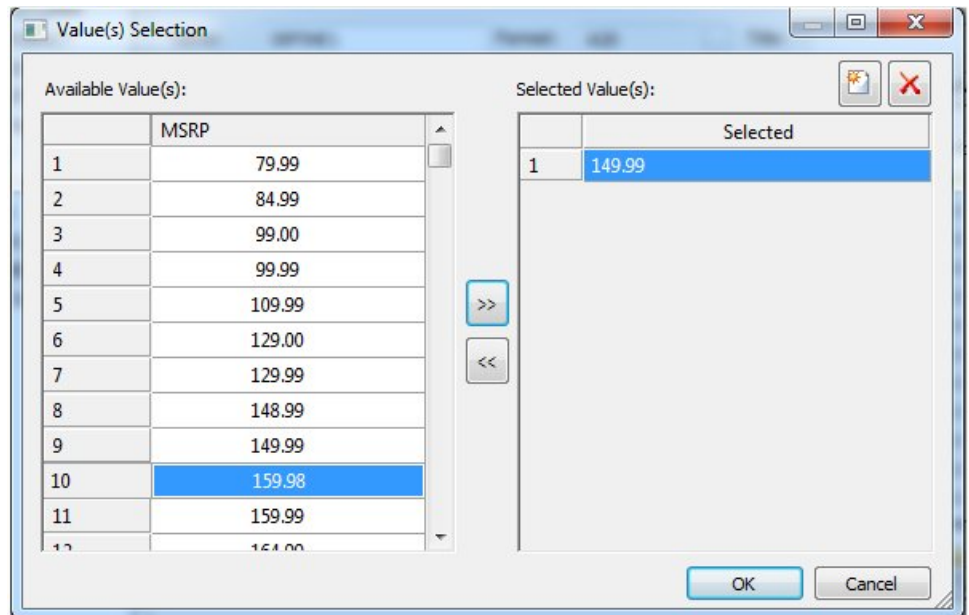
- a. [関係式] (Relational Expression) タブで、[新しい行の追加] (Add New Row)  ボタンをクリックし、ドロップダウンリストを使用してフィルタカラム、関係、タイプを選択します。

注意：パラメータはサポートされません。

- b. [値] (Value) テキストボックスの右側の [...] (参照) ボタンをクリックします。



下図のような [値の選択] (Value Selection) ダイアログボックスが表示されます。



- c. 利用可能な値を選択し、矢印を使用して値を追加または削除します。
- d. [OK] をクリックして [値の選択] (Value Selection) ダイアログボックスを閉じ、[関係式] (Relational Expression) タブに戻ります。

選択した項目が [値] (Value) フィールドに追加されます。

- e. 別の式を追加するには、ダイアログボックス右側の [カラム/変数] (Columns/Variables) タブでフィールドまたは変数をダブルクリックします。

[関係式] (Relational Expression) タブに、新しい行が追加されます。ここで、式を作成することができます。

- f. 式を削除するには、式の番号またはフィールド列を選択し、[削除] (Delete) ボタンを押すか [選択した行の削除] (Delete selected rows) をクリックします。式が [関係式] (Relational Expression) タブから削除されます。
6. ここで、演算のダイアログボックス右上の [式の確認] (Check Expression) ボタンおよび [サンプルデータ] (Sample Data) ボタンを選択し、式が有効かを確認するとともに、フィルタのサンプルデータを表示することもできます。
7. [OK] をクリックして演算のダイアログボックスを閉じ、シノニムエディタに戻ります。

注意： DEFINE、タイトル、または式を編集するには、シノニムエディタの [プロパティ] (Properties) セクションから直接変更するか、[EXPRESSION] の [値] (Value) テキストボックスの右側にある [...] (参照) ボタンをクリックして一時項目の演算を再起動します。

8. 式にフィールドが使用されていない、または一時項目 (DEFINE) が定義されていない場合、WITH オプションを使用して、定義済みの演算の論理ホームを指定することができます。WITH オプションにより、一時項目の論理ホームを通常よりも下位のセグメントに割り当てることもできます (下位セグメントのインスタンスの計算など)。

ヒント： [フィールド表示] (Field View) タブで一時項目 (DEFINE) をドラッグアンドドロップし、別のセグメントに移動することができます。その場合、セグメントの関連付けも変更されます。

9. NULL データを許可するフィールドに対しては、[ミッシングデータ] (Missing Data) オプションを選択します。すべてのミッシングデータを許可することができます。
10. [ファイル] (File) メニューから [保存] (Save) を選択し、シノニムを保存します。
11. シノニムエディタを閉じるには、[ファイル] (File) メニューから [閉じる] (Close) を選択するか、画面右上隅のコントロールボタンをクリックします。

式および一時項目 (DEFINE) の属性についての詳細は、185 ページの「[カスタムフィールドの属性および式の定義](#)」を参照してください。

シノニムフィルタの作成

手順/構文

シノニムのフィルタを作成するには

シノニムエディタでマスターファイルのフィルタを作成し、ビジネスビューファイルまたはレポートツールで使用することができます。また、フィルタを使用して他のデータ確認を実行したり、作成した条件に基づいてデータをソートしたりすることもできます。

フィルタは特定のセグメントに対して作成され、デフォルト設定では、選択したセグメントに関連付けられます。フィルタは関連情報を持たないセグメントに対しても作成できます。

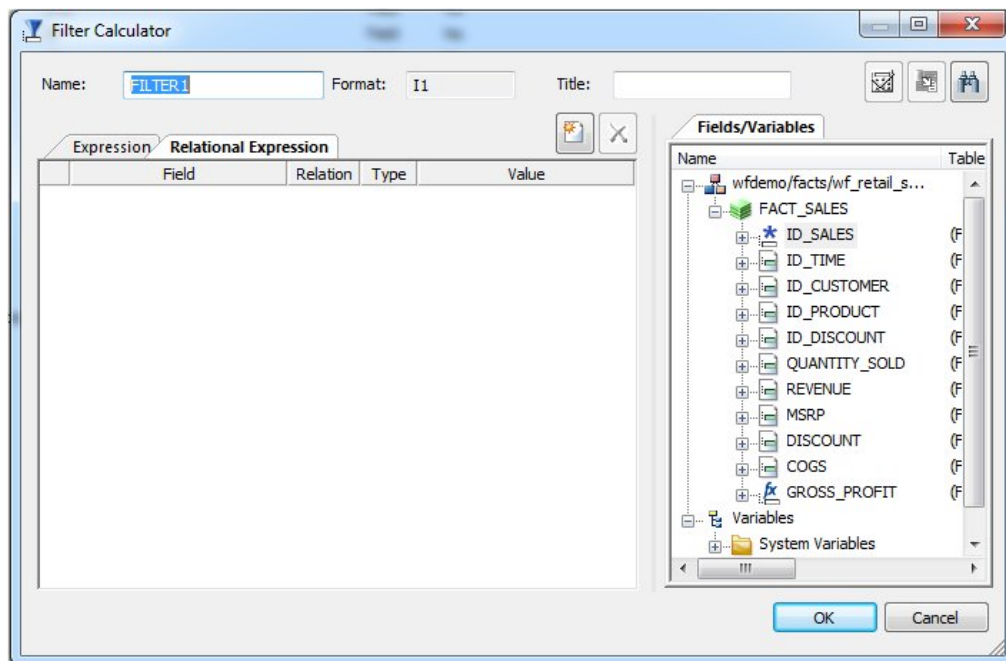
手順 シノニムのフィルタを作成するには

1. [データサーバ] (Data Servers) エリアの [マスターファイル] (Master File) フォルダでシノニムをダブルクリックするか、シノニムを右クリックして [シノニムエディタで編集] (Edit in Synonym Editor) を選択します。

シノニムエディタでシノニムが開きます。

2. セグメントまたはフィールドを右クリックし、[挿入] (Insert)、[フィルタ] (Filter) を選択します。

下図のように、[フィルタの演算] (Filter Calculator) ダイアログボックスが開きます。



3. [フィルタの演算] (Filter Calculator) の [名前] (Name) テキストボックスにフィールド名を入力するか、デフォルトのフィルタ名を使用します。

注意：レポート実行時のフィルタの動作を識別しやすくするため、フィルタに分かりやすい名前を付けることをお勧めします。

4. [フォーマット] (Format) テキストボックスにはデフォルト値の「I1」が表示されます。

注意：このフォーマットを変更することはできません。フィルタの値は false の場合 0 (ゼロ)、true の場合 1 を返します。


5. [TITLE] テキストボックスにフィルタのタイトルを入力することもできます。

ヒント：シノニムエディタの [プロパティ] (Properties) ウィンドウでは、[TITLE] および [DESCRIPTION] の [値] (Value) テキストボックスの右側にある [...] (参照) ボタンをクリックし、多言語のタイトルを指定します。

6. [式] (Expression) タブおよび演算ボタンを使用し、フィルタの式を作成します。

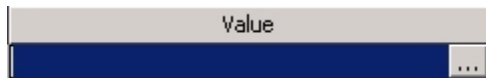
または

[関係式] (Relational Expression) タブを使用し、条件式を作成します。

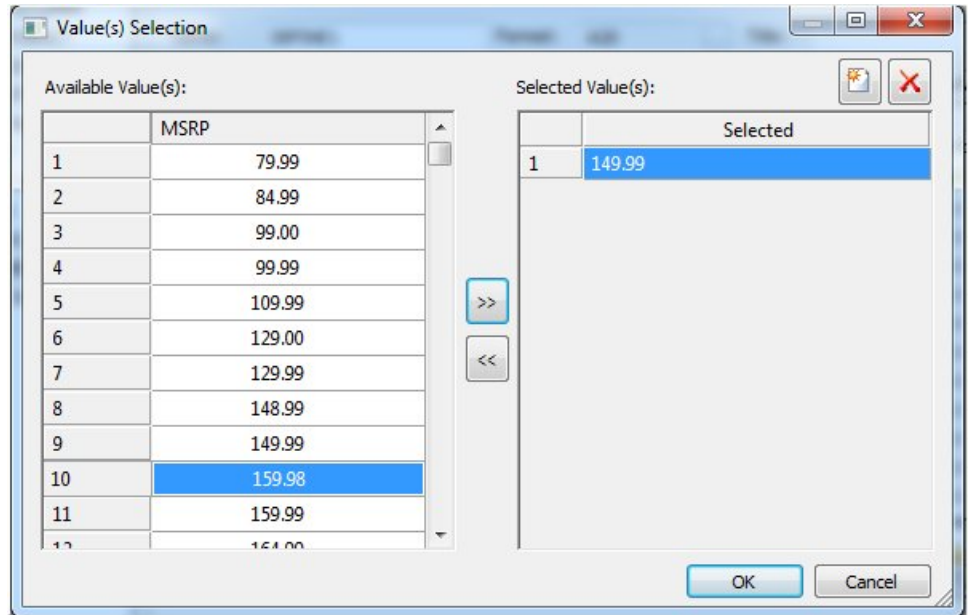
- a. [関係式] (Relational Expression) タブで、[新しい行の追加] (Add New Row)  [関係式] (Relational Expression) タブから、ドロップダウンリストを使用してフィルタを設定する [カラム] (Column)、[関係] (Relation)、[タイプ] (Type) を選択します。

注意：パラメータはマスターファイルフィルタには使用できません。

- b. [値] (Value) テキストボックスの右側の [...] (参照) ボタンをクリックします。

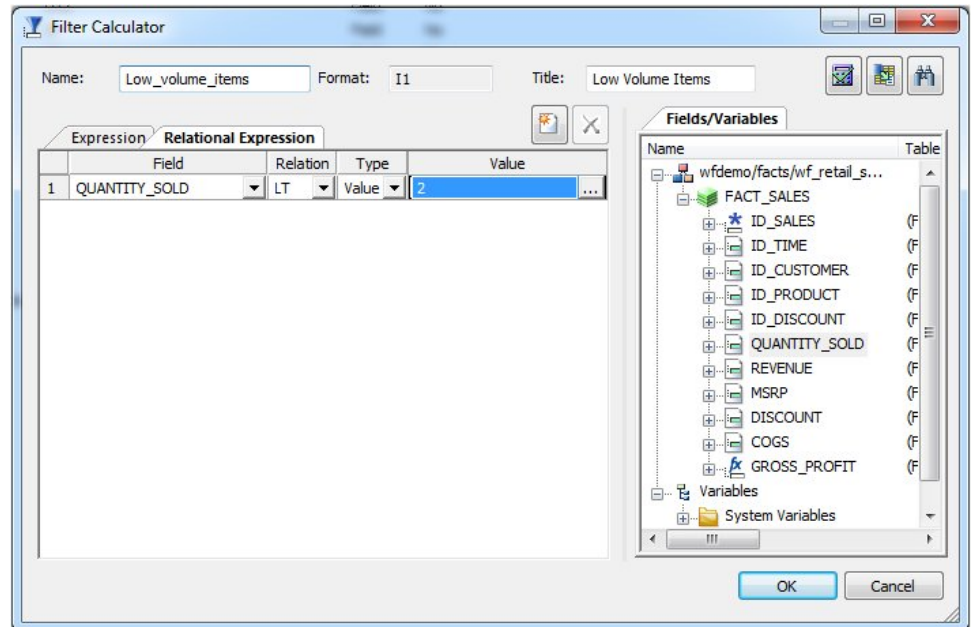


下図のような [値の選択] (Value Selection) ダイアログボックスが表示されます。



- c. 利用可能な値を選択し、矢印を使用して値を追加または削除します。
- d. [OK] をクリックして [値の選択] (Value Selection) ダイアログボックスを閉じ、[関係式] (Relational Expression) タブに戻ります。

値フィールドに式が追加されます。

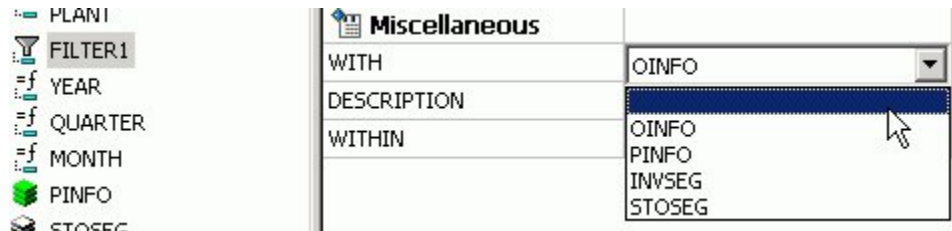


- e. フィルタを追加するには、[フィルタの演算] (Filter Calculator) 右側の [フィールド/変数] (Fields/Variables) タブで、フィールドまたは変数をダブルクリックします。
フィルタは [関係式] (Relational Expression) タブに追加されます。このタブでは、式の値を追加することができます。
- f. 式を削除するには、式の番号またはフィールド列を選択し、[削除] (Delete) ボタンを押すか [選択した行の削除] (Delete selected rows) をクリックします。
式が [関係式] (Relational Expression) タブから削除されます。
- 7. ここで、[フィルタの演算] (Filter Calculator) の右上の [式の確認] (Check expression) ボタンおよび [サンプルデータ] (Sample Data) ボタンを選択し、式が有効かを確認するとともに、フィルタのサンプルデータを表示することもできます。
- 8. [OK] をクリックし、[一時項目 (COMPUTE) の演算] (Compute Calculator) を閉じ、シノニムエディタに戻ります。

注意：フィルタ、タイトル、または式を編集するには、シノニムエディタの [プロパティ] (Properties) ウィンドウから直接実行するか、[EXPRESSION] の [値] (Value) テキストボックス右側にある [...] (参照) ボタンをクリックして [フィルタの演算] (Filter Calculator) を再起動します。

9. セグメントとの関連付けを行わないフィルタを作成するには、[WITH] ドロップダウンリストからブランクセグメントを選択します。

フィルタは特定のセグメントに対して作成され、デフォルト設定では、選択したセグメントに関連付けられます。フィルタは関連情報を持たないセグメントに対しても作成できます。



注意：複数のセグメントに存在するフィールドを使用する場合、関連付ける WITH セグメントは最下位セグメントである必要があります。そうでない場合は、エラーを回避するためにブランクにしておきます。

また、シノニムのフィールドが式や一時項目 (COMPUTE) にどれも使用されていない場合、WITH オプションを使用して、フィルタ演算の論理ホームを指定することができます。WITH オプションを使用して、フィルタフィールドの論理ホームを通常よりも下位のセグメントに移動することもできます (下位セグメントのインスタンスの計算など)。

10. [ファイル] (File) メニューから [保存] (Save) を選択し、シノニムを保存します。
フィルタはシノニムの一部として保存されます。
11. シノニムエディタを閉じるには、[ファイル] (File) メニューから [閉じる] (Close) を選択するか、画面右上隅のコントロールボタンをクリックします。

シノニムへの一時項目 (COMPUTE) の追加

手順/構文

シノニムエディタで一時項目 (COMPUTE) を作成するには

シノニムエディタでは、カスタムフィールドとして、一時項目 (COMPUTE) を作成することができます。カスタムフィールドを追加する手順は、DEFINE フィールドの追加と同様です。一時項目 (COMPUTE) は「マスターファイルの COMPUTE フィールド」として DEFINE フィールドやその他のフィールドと区別されます。

手順

シノニムエディタで一時項目 (COMPUTE) を作成するには

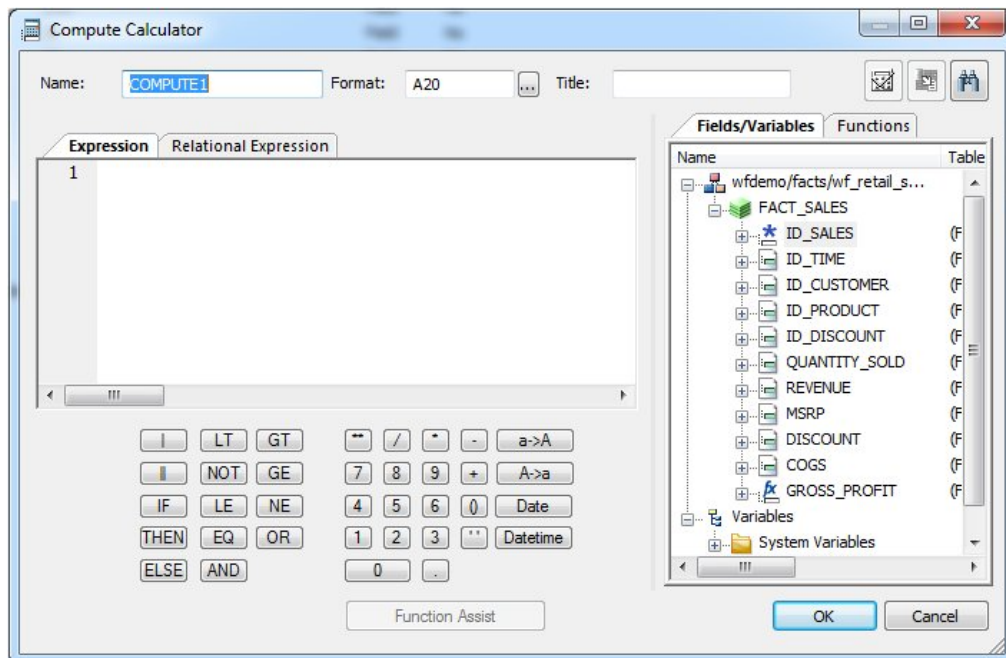
シノニムに 一時項目 (COMPUTE) を作成するには、次の手順を実行します。

1. [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。

シノニムエディタでシノニムが開きます。

2. セグメント (またはフィールド) を右クリックし、[挿入] (Insert)、[一時項目 (COMPUTE)] (Compute) を選択します。

[一時項目 (COMPUTE) の演算] (Compute Calculator) ダイアログボックスが表示されます。




3. [一時項目 (COMPUTE) の演算] (Compute Calculator) の [カラム] (Column) テキストボックスにフィールド名を入力するか、デフォルトのフィールド名を使用します。
4. [タイトル] (Title) テキストボックスに一時項目 (COMPUTE) のタイトルを入力することもできます。

ヒント：シノニムエディタでは、[TITLE] および [DESCRIPTION] の [値] (Value) テキストボックス右側にある [...] (参照) ボタンをクリックし、多言語のタイトルを指定します。

5. [式] (Expression) タブと演算用のボタンをクリックし、一時項目 (COMPUTE) の式を作成します。

または

[関係式] (Relational Expression) タブを使用し、条件式を作成します。

- a. [関係式] (Relational Expression) タブで、[新しい行の追加] (Add New Row)  [関係式] (Relational Expression) タブから、ドロップダウンリストを使用してフィルタを設定する [カラム] (Column)、[関係] (Relation)、[タイプ] (Type) を選択します。

注意：パラメータはサポートされません。

- b. [値] (Value) テキストボックスの右側の [...] (参照) ボタンをクリックします。
- c. 利用可能な値を選択し、矢印を使用して値を追加または削除します。
- d. [OK] をクリックして [値の選択] (Value Selection) ダイアログボックスを閉じ、[関係式] (Relational Expression) タブに戻ります。

値フィールドに式が追加されます。

- e. フィルタを追加するには、[一時項目 (COMPUTE) の演算] (Compute Calculator) の右側の [カラム/変数] (Columns/Variables) タブでフィールドまたは変数をダブルクリックします。

フィルタは [関係式] (Relational Expression) タブに追加されます。このタブでは、式の値を追加することができます。

- f. 式を削除するには、式の番号またはフィールド列を選択し、[削除] (Delete) ボタンを押すか [選択した行の削除] (Delete selected rows) をクリックします。式が [関係式] (Relational Expression) タブから削除されます。

6. ここで、[一時項目 (COMPUTE) の演算] (Compute Calculator) 右上の [式の確認] (Check expression) ボタンおよび [サンプルデータ] (Sample Data) ボタンをクリックし、式が有効かを確認するとともに、フィルタのサンプルデータを表示することもできます。
7. [OK] をクリックし、[一時項目 (COMPUTE) の演算] (Compute Calculator) を閉じ、シノニムエディタに戻ります。

注意： COMPUTE、タイトル、または式を編集するには、シノニムエディタの [プロパティ] (Properties) セクションから直接実行するか、 [式] (Expression) 右側の [値] (Value) テキストボックス右側にある [...] (参照) ボタンをクリックして [一時項目 (COMPUTE) の演算] を再起動します。

8. NULL データを許可するフィールドに対しては、 [ミッシングデータ] (Missing Data) オプションを選択します。すべてのミッシングデータを許可することができます。
9. [ファイル] (File) メニューから [保存] (Save) を選択し、シノニムを保存します。
10. シノニムエディタを閉じるには、 [ファイル] (File) メニューから [閉じる] (Close) を選択するか、画面右上隅のコントロールボタンをクリックします。

式および COMPUTE 属性についての詳細は、185 ページの「[カスタムフィールドの属性および式の定義](#)」を参照してください。

一時項目内 (DEFINE) に繰り返しフィールド数を格納

手順/構文

一時項目 (DEFINE) を使用した OCCURS 指定

参照

OCCURS で一時項目 (DEFINE) を使用する際の注意

マスターファイルの OCCURS 属性は、データソースの繰り返しフィールド、または繰り返しフィールドグループを記述します。フィールドの繰り返しグループは、マスターファイルで子孫セグメントとして記述され、そのセグメントの OCCURS 属性は、繰り返し数を決定する方法を指定します。

繰り返しの数は、各レコードインスタンスで同一である必要はありません。繰り返しの数は、データソースのフィールドから計算される場合があります。その場合、マスターファイルで、各レコードの繰り返し数を示す一時項目 (DEFINE) を作成して、その一時項目を OCCURS 属性の値として使用します。

構文 一時項目 (DEFINE) を使用した OCCURS 指定

```

SEGNAME = parent, SEGTYPE = segtype,$
.
.
.
DEFINE definefield/In = expression;
SEGNAME = osegname, SEGTYPE=S0, PARENT = parent,
  OCCURS = definefield ,$
FIELDNAME = rfield, ALIAS = ralias,
  USAGE = rufmt, ACTUAL = rafmt,$
.
.
.
[FIELDNAME = orderfield, ALIAS = ORDER,
  USAGE = In, ACTUAL = I4,$]

```

説明

parent

親セグメント名です。

segtype

親セグメントの SEGTYPE です。

definefield

繰り返しされるフィールドまたはフィールドグループの数を示す一時項目 (DEFINE) です。このフィールドは、繰り返しフィールドが格納されているセグメントの上位セグメントに定義する必要があります。

n

繰り返しの数を記述する一時項目 (DEFINE) のフォーマットです。整数フォーマットにする必要があります。

expression

各レコードインスタンスの繰り返しの数を取得する有効な式です。

osegname

下位 OCCURS セグメントの名前です。

rfield

OCCURS セグメント内の繰り返しフィールドの名前です。

ralias

OCCURS セグメント内の繰り返しフィールドのエイリアスです。

rufmt

OCCURS セグメント内の繰り返しフィールドの表示フォーマットです。

rafmt

OCCURS セグメント内の繰り返しフィールドの実際のフォーマットです。

orderfield

内部カウンタフィールドの名前です。OCCURS セグメントの最後のフィールドとして指定することができます。orderfield フィールドは、各繰り返しフィールドに連続番号を関連付け、繰り返しデータの順序が重要な場合に役立ちます。たとえば、月単位または四半期単位の値を示しても、レコード自体でデータに対応する月または四半期が明示的に指定されていない場合があります。USAGE フォーマットは、整数にする必要があります。また、ACTUAL フォーマットは 14 です。

参照

OCCURS で一時項目 (DEFINE) を使用する際の注意

OCCURS の値として使用する一時項目 (DEFINE) は、マスターファイルまたはそれ以外で、再定義することはできません。

例

OCCURS セグメントでの一時項目 (DEFINE) の使用

EMPLOYEE データソースを使用した次のリクエストは、繰り返しフィールドが存在する固定フォーマットのシーケンシャルファイルを作成します。以下は、このリクエストについての説明です。

- 各従業員の FICA (米国連邦保険拠出法) 控除項目数を計算します。
- 各従業員の FICA 控除項目フィールドすべての長さを格納する一時項目 (COMPUTE) フィールドを作成します。
- レコードのそれぞれに、計算した従業員の控除項目フィールドの長さ、従業員の識別情報、従業員の FICA 控除のすべてを格納する HOLD ファイルを作成します。

控除項目数は、従業員ごとに異なります。レコードで控除項目を含む部分が OCCURS セグメントを構成します。繰り返しの数は、TABLE リクエストで作成した長さのフィールドに基づいて取得されます。

繰り返される控除項目フィールドが格納されたファイルを作成するプロシジヤは、次のとおりです。

```

DEFINE FILE EMPLOYEE
  CTR/I5 WITH DED_AMT = IF EMP_ID NE LAST EMP_ID THEN 1 ELSE LAST CTR + 1;
END
TABLE FILE EMPLOYEE
  SUM CNT.DED_AMT NOPRINT EMP_ID LAST_NAME FIRST_NAME CURR_SAL
  COMPUTE DEDLEN/I5 = 12 * CNT.DED_AMT;
  BY EMP_ID NOPRINT
  SUM DED_AMT
  BY EMP_ID NOPRINT
  ACROSS CTR NOPRINT
  WHERE DED_CODE EQ 'FICA'
  ON TABLE SET HOLDLIST PRINTONLY
  ON TABLE HOLD AS OCCURS1 FORMAT ALPHA
END

```

OCCURS1 ファイルには、従業員ごとに異なる数の DED_AMT フィールドが格納された 1 件のレコードがあります。DED_AMT の実際の件数の長さ合計は、「DEDLEN」という名前のフィールドに格納されます。HOLD コマンドが生成したマスターファイルには、10 個の DED_AMT フィールドが記述されます。

```

FILENAME=OCCURS1 , SUFFIX=FIX , $
SEGMENT=OCCURS1, SEGTYPE=S0, $
  FIELDNAME=EMP_ID, ALIAS=E01, USAGE=A9, ACTUAL=A09, $
  FIELDNAME=LAST_NAME, ALIAS=E02, USAGE=A15, ACTUAL=A15, $
  FIELDNAME=FIRST_NAME, ALIAS=E03, USAGE=A10, ACTUAL=A10, $
  FIELDNAME=CURR_SAL, ALIAS=E04, USAGE=D12.2M, ACTUAL=A12, $
  FIELDNAME=DEDLEN, ALIAS=E05, USAGE=I5, ACTUAL=A05, $
  FIELDNAME=DED_AMT, ALIAS=E06, USAGE=D12.2M, ACTUAL=A12, $
  FIELDNAME=DED_AMT, ALIAS=E07, USAGE=D12.2M, ACTUAL=A12, $
  FIELDNAME=DED_AMT, ALIAS=E08, USAGE=D12.2M, ACTUAL=A12, $
  FIELDNAME=DED_AMT, ALIAS=E09, USAGE=D12.2M, ACTUAL=A12, $
  FIELDNAME=DED_AMT, ALIAS=E10, USAGE=D12.2M, ACTUAL=A12, $
  FIELDNAME=DED_AMT, ALIAS=E11, USAGE=D12.2M, ACTUAL=A12, $
  FIELDNAME=DED_AMT, ALIAS=E12, USAGE=D12.2M, ACTUAL=A12, $
  FIELDNAME=DED_AMT, ALIAS=E13, USAGE=D12.2M, ACTUAL=A12, $
  FIELDNAME=DED_AMT, ALIAS=E14, USAGE=D12.2M, ACTUAL=A12, $
  FIELDNAME=DED_AMT, ALIAS=E15, USAGE=D12.2M, ACTUAL=A12, $

```

このマスターファイルを編集して、繰り返される DED_AMT フィールドを OCCURS セグメントで記述することができます。「NUMOCC」という名前の一時的項目 (DEFINE) は、DEDLEN フィールドの件数を計算します。ORDER フィールドは、実際にはファイルに存在しません。これは、DB2 Web Query によってデータが入力された、内部カウンタです。

```
FILENAME=OCCURS1 , SUFFIX=FIX      , $
  SEGMENT=OCCURS1, SEGTYPE=S0, $
    FIELDNAME=EMP_ID, ALIAS=E01, USAGE=A9, ACTUAL=A09, $
    FIELDNAME=LAST_NAME, ALIAS=E02, USAGE=A15, ACTUAL=A15, $
    FIELDNAME=FIRST_NAME, ALIAS=E03, USAGE=A10, ACTUAL=A10, $
    FIELDNAME=CURR_SAL, ALIAS=E04, USAGE=D12.2M, ACTUAL=A12, $
    FIELDNAME=DEDLEN, ALIAS=E05, USAGE=I5, ACTUAL=A05, $
    DEFINE NUMOCC/I2 = DEDLEN/12;,$
  SEGNAME=DEDUCTION, SEGTYPE=S0, PARENT=OCCURS1, OCCURS=NUMOCC,$
    FIELDNAME=DED_AMT, ALIAS=E06,  USAGE=D12.2M, ACTUAL=A12, $
    FIELDNAME=ORDER,  ALIAS=ORDER, USAGE=I2      , ACTUAL=I4 , $
```

次のリクエストは、ORDER フィールドを使用して、各従業員の繰り返しフィールドの 1 件目を選択して表示します。すべての従業員には、少なくとも 1 件の控除項目が存在することから、レポート出力には、すべての従業員が表示されます。

```
TABLE FILE OCCURS1
  PRINT NUMOCC LAST_NAME CURR_SAL DED_AMT
  WHERE ORDER EQ 1
END
```

出力結果は次のとおりです。

NUMOCC	LAST_NAME	CURR_SAL	DED_AMT
10	STEVENS	\$11,000.00	\$64.17
8	SMITH	\$13,200.00	\$100.10
4	JONES	\$18,480.00	\$247.94
8	SMITH	\$9,500.00	\$60.96
1	BANNING	\$29,700.00	\$519.75
8	IRVING	\$26,862.00	\$626.78
4	ROMANS	\$21,120.00	\$317.62
1	MCCOY	\$18,480.00	\$161.70
5	BLACKWOOD	\$21,780.00	\$444.67
7	MCKNIGHT	\$16,100.00	\$187.88
4	GREENSPAN	\$9,000.00	\$52.50
10	CROSS	\$27,062.00	\$631.40

繰り返しフィールドの 10 件目を表示すると、レポート出力には、2 名の従業員のみが表示されます。

```
TABLE FILE OCCURS1
  PRINT NUMOCC LAST_NAME CURR_SAL DED_AMT
  WHERE ORDER EQ 10
END
```

出力結果は次のとおりです。

NUMOCC	LAST_NAME	CURR_SAL	DED_AMT
10	STEVENS	\$11,000.00	\$58.33
10	CROSS	\$27,062.00	\$526.20

カスタムフィールドの属性および式の定義

参照

カスタムフィールド属性

カスタムフィールドの演算

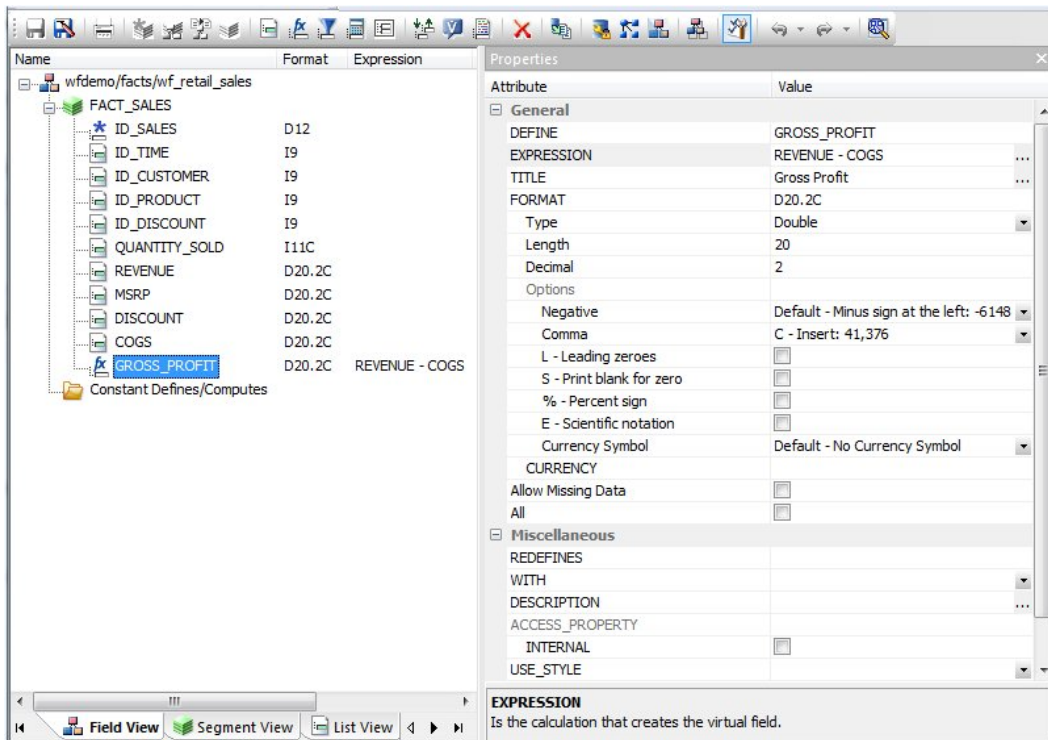
カスタムフィールドは、その値がデータソースに格納されているのではなく、データソースにあるデータから計算されるフィールドです。一時項目 (DEFINE)、マスターファイルフィルタ (FILTER)、または一時項目 (COMPUTE) を追加することにより、シノニムにカスタムフィールドを作成することができます。このフィールドは、関連付けられたデータソースにレポートツールからアクセスするときに使用できます。

シノニムエディタを使用して、属性値を定義し、カスタムフィールドの式を作成することができます。

参照 カスタムフィールド属性

以下の属性は、シノニムエディタのカスタムフィールド (DEFINE、FILTERS、COMPUTE) に使用することができます。

注意：利用可能な属性は、シノニムおよび選択するカスタムフィールドのタイプにより異なります。下図は、一時項目 (DEFINE) が選択された SQL データソースの例を示しています。



カスタムフィールド (DEFINE、FILTERS、COMPUTE) で一般的な属性には、次のものがあります。

全般 (General)

DEFINE

一時項目 (DEFINE) 名です。

注意：この属性は、一時項目 (DEFINE) が選択されたときにのみ表示されます。

FILTER

マスターファイルのフィルタフィールド名です。

注意：この属性は、仮想フィルタフィールドが選択されたときにのみ表示されます。

COMPUTE

一時項目 (COMPUTE) 名です。

注意：この属性は、一時項目 (COMPUTE) が選択されたときにのみ表示されます。

EXPRESSION

一時項目 (DEFINE) を作成するための演算式です。

TITLE

フィールド名の代わりにレポートで一般に使用するタイトルを指定します。これにより、一時項目 (DEFINE) に多言語のタイトルを指定することができます。

FORMAT

一時項目 (DEFINE) のデータタイプとフォーマットです。

注意：この属性は、一時項目 (DEFINE および COMPUTE) カスタムフィールドのみに表示されます。

ミッシングを許可 (Allow Missing Data)

ミッシングデータを許可します。それ以外の場合、トランザクション値が提供されません。

注意：この属性は、一時項目 (DEFINE および COMPUTE) カスタムフィールドのみに表示されます。

すべて (All)

すべてのミッシングデータを許可します。それ以外の場合、トランザクション値が提供されます。

注意：この属性は、一時項目 (DEFINE および COMPUTE) カスタムフィールドのみに表示されます。

その他 (Miscellaneous)

WITH

式にフィールドが使用されていない、または一時項目 (DEFINE) が定義されていない場合、WITH オプションを使用して、定義済みの演算の論理ホームを指定することができます。WITH オプションにより、一時項目の論理ホームを通常よりも下位のセグメントに割り当てすることもできます (下位セグメントのインスタンスの計算など)。

注意：この属性は、DEFINE および FILTER カスタムフィールドにのみ表示されます。

DESCRIPTION

一時項目 (DEFINE) についての説明 (コメント) を格納します。

WITHIN

ディメンションに追加するフィールド名を格納します。

これらの WITHIN ステートメントは、ディメンションビルダでシノニムに追加され、リレーショナルテーブルの OLAP を有効にします。WITHIN ステートメントにより、OLAP コントロールパネルを使用する OLAP の分析が可能になります。

注意：この属性は、DEFINE および FILTER カスタムフィールドにのみ表示されます。

SCD タイプ (SCD Type)

SCD を設定します。このオプションは、既存のリレーショナルターゲットにのみ使用可能です。

サロゲートキー (surrogate key) テーブルの先頭フィールドで、SCD タイプはブランクです。これ以外で SCD タイプが [ブランク] (blank) のフィールドに対しては、SCD 処理は行われません。

論理キーフィールド (Logical Key Field) データベースキーです。

アクティベーションフラグ (Activation Flag) 行が最新であることを示します。

開始日/終了日 (Begin Date/End Date) 行の値に対する日付範囲です。終了日がブランクの場合、行が最新であることを示します。

タイプ I (Type I) (履歴を上書き) データベース値を新しい値で上書きするフィールドを示します。

タイプ II (Type II) (履歴を保存) データベース行に非アクティブフラグが設定されている、または終了日が割り当てられているフィールドを示します。新しい値の新しい行が挿入されます。

ブランク (blank) (キーフィールドではないフィールド) データベース値が変化しないことを示します。

USE_STYLE

フィールドに適用するスタイルシート名です。

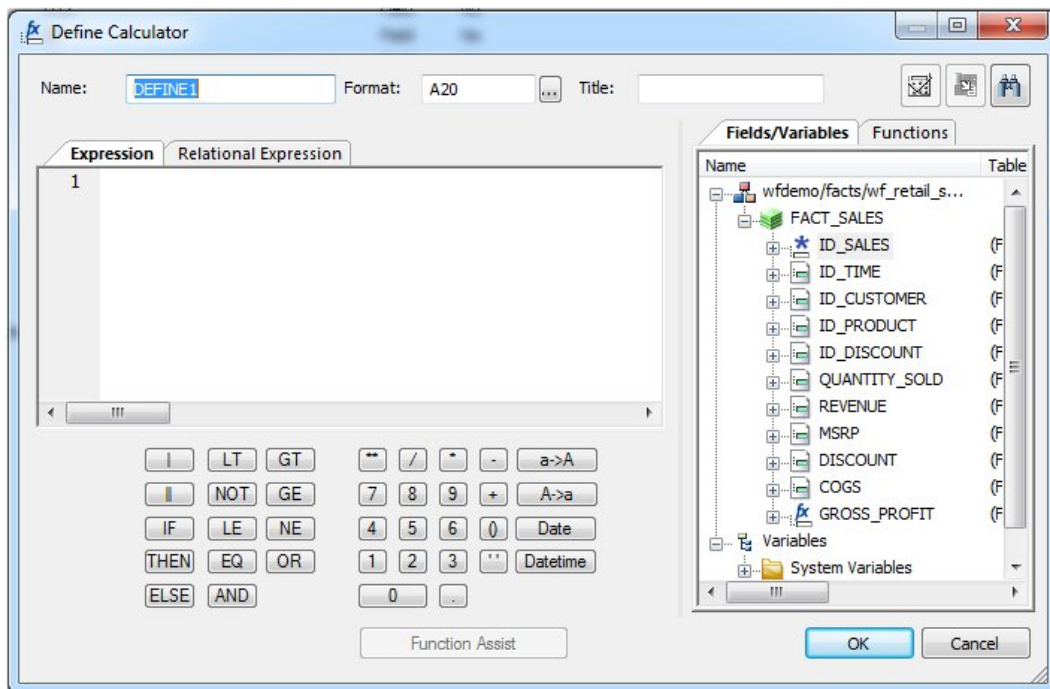
注意：利用可能な属性は、シノニムのタイプにより異なります。

参照 カスタムフィールドの演算

[一時項目 (DEFINE) の演算] (Define Calculator)、[フィルタの演算] (Filter Calculator)、または [一時項目 (COMPUTE) の演算] (Compute Calculator) を起動するには、シノニムエディタの [EXPRESSION] の [値] (Value) テキストボックス右側にある [...] (参照) ボタンをクリックします。



作成するカスタムフィールドのタイプにより異なるダイアログボックスが開きます。



演算のダイアログボックスには、次のフィールドとオプションがあります。

名前 (Name)

作成されるオブジェクト (一時項目 (DEFINE)、フィルタ、一時項目 (COMPUTE)) の名前です。

フォーマット (Format)

ファイルフォーマットです。

式 (Expression) タブ

式を入力する場所です。[カラム/変数] (Columns/Variables) タブからデータソースフィールドを、[関数] (Functions) タブから関数を、[演算] (Calculator) から数字および演算子を追加することができます。

関係式 (Relational Expression) タブ

フィルタ作成ウィンドウを表示します。ここで、フィールドの追加と削除、関係とタイプの選択、フィルタ値の選択を行います。

フィールド/変数 (Fields/Variables) タブ

利用可能なソースフィールドの階層リスト、および条件式の作成に使用可能なシステム変数フォルダを表示します。

関数 (Functions) タブ

関数は、値を返すプログラムです。このタブには、一時項目の値を取得するビルトイン関数が一覧表示されます。

関数アシスト (Function Assist) ボタン

変換を作成または編集する際に、ダイアログボックスを表示します。このダイアログボックスで、関数のパラメータを指定することができます。

数字/演算子

数字と演算子を入力することができます。

次の演算子があります。

| (縦棒)

2つの値を連結します。1つ目の値の末尾に空白があれば、これらは保持されます。たとえば、FIRST_NAME と LAST_NAME がともに A15 フォーマットの場合、次の式で連結します。

```
FULL_NAME = FIRST_NAME | LAST_NAME
```

次のようなフィールドが生成されます。

```
MICHAEL      SMITHSONJ  
ANE          JONES  
.  
.  
.
```

|| (2つの縦棒)

2つの値を連結します。1つ目の値の末尾に空白があれば、これらはすべて削除されます。たとえば、次の式は氏名を作成し、カンマ (,) を挿入します。

```
FULL_NAME = LAST_NAME || (', ' | FIRST_NAME)
```

次のようなフィールドが生成されます。

```
SMITHSON, MICHAEL  
JONES, JANE  
.  
.  
.
```

括弧で囲まれた部分の連結を最初に行い (フィールドの後の空白は保持)、結果は LAST_NAME と連結し、LAST_NAME の末尾の空白は削除します。

IF

条件分岐を成立させます。

THEN

条件分岐の結果が TRUE の場合の動作を指定します。

ELSE

条件分岐の結果が FALSE の場合の動作を指定します。

LT

左辺の値が右辺の値より小さい場合、TRUE を返します。

NOT

被演算子が false の場合、TRUE を返します。

LE

左辺の値が右辺の値以下である場合、TRUE を返します。

EQ

左辺の値が右辺の値と等しい場合、TRUE を返します。

AND

被演算子がいずれも TRUE である場合、TRUE を返します。

GT

左辺の値が右辺の値より大きい場合、TRUE を返します。

GE

左辺の値が右辺の値以上である場合、True を返します。

NE

左辺の値が右辺の値と等しくない場合、True を返します。

OR

被演算子のいずれかが True の場合、True を返します。

べき算します。

()

括弧記号を入力します。

"

二重引用符 (") を入力します。引用符の間に、文字の値を入力します。

a->A

選択した文字列を大文字に変換します。

A->a

選択した文字列を小文字に変換します。

式の確認 (Check expression) ボタン

式が正しいかを確認します。

サンプルデータ (Sample data) ボタン

式のサンプルデータを生成します。

シノニムへのグループフィールドの追加

手順/構文

セグメントにグループフィールドを追加するには

グループフィールドに新しいフィールドを追加するには

既存のフィールドをグループフィールドに追加するには

セグメントからグループフィールドを削除するには

グループフィールドからフィールドを削除するには

参照

グループフィールド属性

グループをサポートするデータソースでは、一意の名前を複数フィールドに割り当てて、グループフィールドを作成することができます。グループフィールドは、物理的に隣接する複数の文字フィールドにより作成されます。グループフィールドは、類似するフィールドまたは論理的に接続されているフィールドをグループ化し、別々のセグメントではなく1つの単位としてアクセスするための効率的な手段を提供します。

注意： MAINTAIN 環境では、グループキーを除いたマスターファイルのグループフィールドがサポートされます。グループキーは、VSAM でサポートされる実フィールドのグループです。MAINTAIN 環境でグループを作成する場合、グループは表示されず、グループを構成するフィールドのみが表示されます。

手順

セグメントにグループフィールドを追加するには

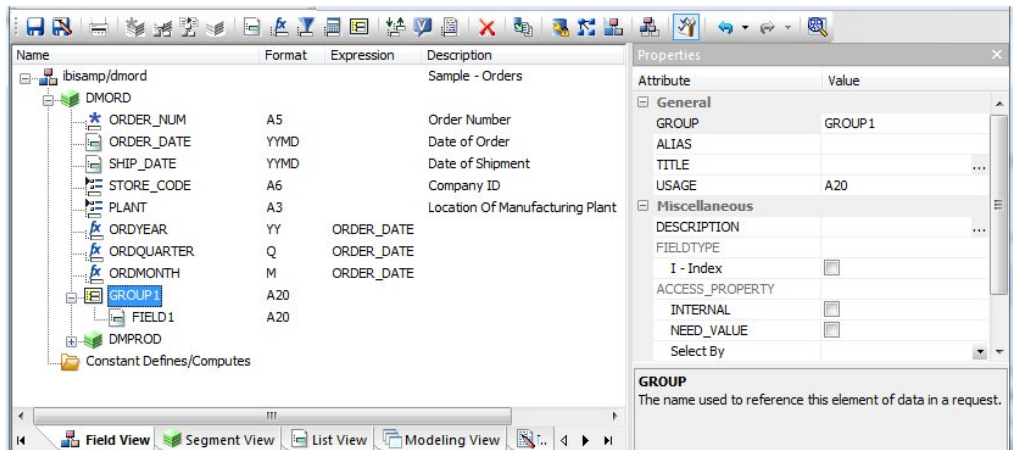
1. [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。

シノニムエディタでシノニムが開きます。

2. セグメント (またはフィールド) を右クリックし、[挿入] (Insert)、[グループ] (Group) を選択します。

注意：データソースによっては、グループオプションを利用できない場合があります。

シノニムにグループが追加され、この属性と値がツリー右側に表示されます。



3. [GROUP] テキストボックスにグループ名を入力します。
4. グループの USAGE タイプを選択します。グループの長さ、小数点以下の桁数 (必要な場合)、表示オプションを指定します。
5. グループにインデックスを付ける場合は、[I インデックス] (I-Index) のチェックをオンにします。
6. ACCESS_PROPERTY の値を選択し、グループのデータにアクセスオプションを指定します。
7. 必要に応じて、TITLE および DESCRIPTION 表示オプションを指定することもできます。
8. [ファイル] (File) メニューから [保存] (Save) を選択し、シノニムを保存します。

9. シノニムエディタを閉じるには、[ファイル] (File) メニューから [閉じる] (Close) を選択するか、画面右上隅のコントロールボタンをクリックします。

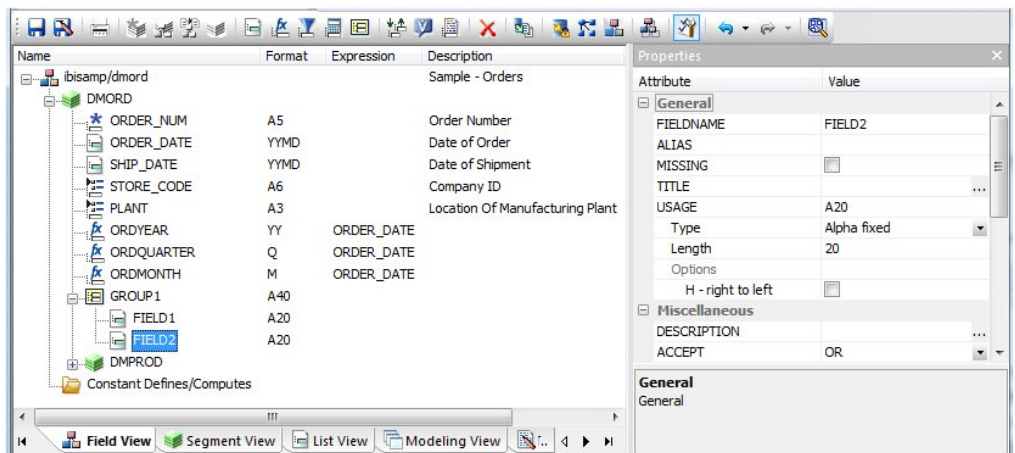
注意：グループを追加する場合、フィールドはそのグループに自動的に追加されます。

手順

グループフィールドに新しいフィールドを追加するには

1. 新しいフィールドを追加するグループフィールドを右クリックし、[挿入] (Insert)、[フィールド] (Field) を選択します。

下図のように、グループにフィールドが追加され、この属性と値がツリーの右側に表示されます。



2. グループフィールドに必要な情報を入力します。フィールド属性についての詳細は、111 ページの「[シノニム属性の表示および編集](#)」を参照してください。

手順

既存のフィールドをグループフィールドに追加するには

1. グループフィールドに追加するフィールドをクリックします。
2. マウスの左ボタンを押しながら、フィールドをグループフィールド名にドラッグアンドドロップします。

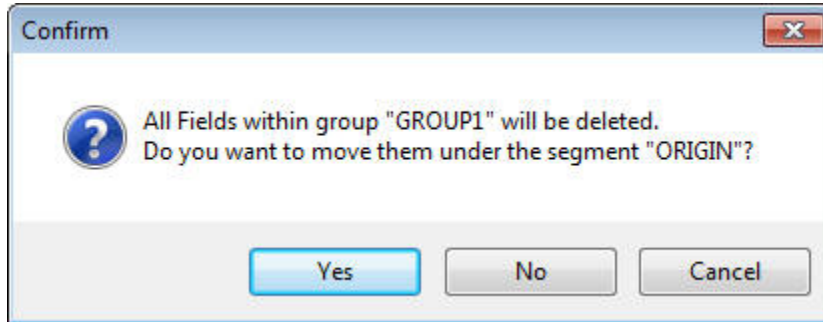
これにより、フィールドがグループフィールドに追加されます。

手順

セグメントからグループフィールドを削除するには

1. グループを右クリックし、[削除] (Delete) を選択します。

グループ内のすべてのフィールドの削除を確認するメッセージが表示されます。



2. グループを削除してシノニムのルートセグメントにあるフィールド(グループ内)を移動するには、[はい] (Yes) をクリックします。

グループおよびグループ内のすべてのフィールドを削除するには、[いいえ] (No) をクリックします。

削除を確認するダイアログボックスを閉じて、シノニムエディタに戻るには、[キャンセル] (Cancel) をクリックします。

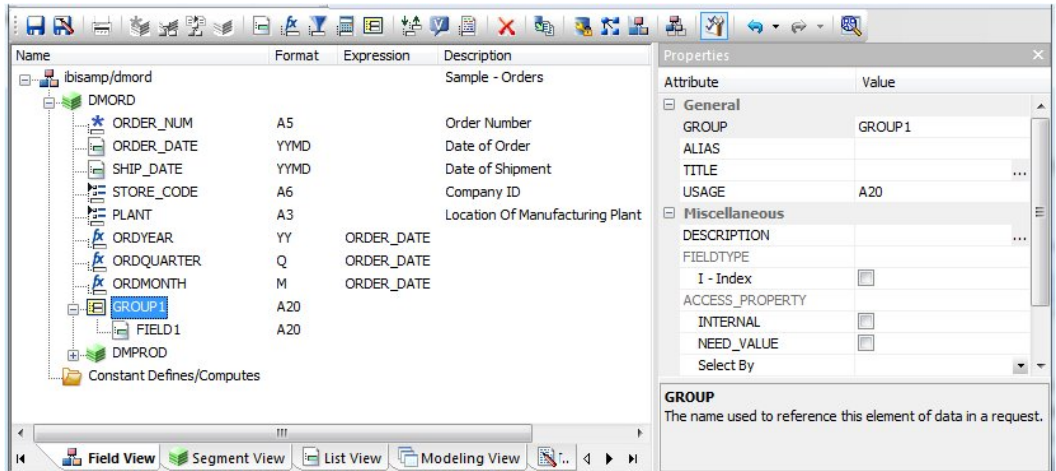
手順

グループフィールドからフィールドを削除するには

フィールドを右クリックし、[削除] (Delete) を選択します。

参照 グループフィールド属性

シノニム内のグループフィールドには、次の属性を含めることができます。



全般 (General)

GROUP

グループ名です。

ALIAS

グループに別名を割り当てます。

レポートを作成すると、グループに別のタイトルを指定していない限り、列見出しとしてグループ名が表示されます。ALIAS を列タイトルとして使用することはできません。

TITLE

グループ名の代わりに使用するタイトルを入力します。TITLE は通常、レポートで使用され、グループの多言語のタイトルを指定することができます。

USAGE

グループフィールドのフォーマットです。グループフィールドは、いくつかのフィールドを連結して構成するため、適切なフォーマットはシノニムエディタが特定します。たとえば、グループフィールドが2つの文字フィールドを持ち、各フィールドの長さが20バイト(A20)の場合、グループフィールドは長さが40バイトの文字フィールド(A40)になります。個々のフィールドのタイプに関わらず、グループフィールドは常に文字フィールドです。

その他 (Miscellaneous)

DESCRIPTION

グループについての説明(コメント)を格納します。この説明は、フィールドリストおよびステータスバーに表示されます。

OLAP 有効レポートでは、フィールドの説明がツールヒントとして表示されます。説明を入力しない場合は、ツールヒントにはフィールド名 (列タイトル) が表示されます。

FIELDTYPE

インデックスグループを示します。フィールドの値にインデックスを付けることで、データ検索のパフォーマンスを向上させることができます。そのためには、フィールドを追加してデータを追加する前に、[インデックス](Index)のチェックをオンにします。インデックスは内部的に格納、管理されるデータ値と位置情報のテーブルで、データ検索のパフォーマンスを向上させます。マスターファイルには、いくつかのインデックスを関連付けることができますが、インデックスとセグメントの最大合計数は 64 です。

注意：FIELDTYPE=R は読み取り専用フィールドを示します。この設定は、RDBMS によってフィールドに値が自動的に割り当てられる場合に役立ちます。

ヒント：データをフィールドに追加後、インデックスをオンにすることができますが、インデックスを作成するには、リビルド機能のインデックス再定義オプションを使用する必要があります。

ACCESS_PROPERTY

フィールドデータのアクセスオプションを指定します。

INTERNAL - サンプルデータまたは利用可能なフィールドリストに表示されないフィールドを定義します。レポートツールのフィールドリストに表示するフィールドを制限します。

NEED_VALUE - データにアクセスする際に値が必要になるフィールドを定義します。レポートリクエストで選択 (WHERE 条件) が必要なことを示します。

選択基準 (Select By) - 値、範囲、複数值でフィールドを定義します。

- [値] (Value) のチェックがオンの場合、レポートリクエストで1つの値のみの選択を定義する必要があります。
- [範囲] (Range) のチェックがオンの場合、レポートリクエストで選択範囲を定義する必要があります。

- [複数値](Multivalued)のチェックがオンの場合、レポートリクエストで複数の値を選択することができます。

USE_STYLE

グループに適用するスタイルシート名です。

注意：利用可能な属性は、シノニムのタイプにより異なります。

データベース管理者セキュリティの使用

トピックス

アクセスタイプの選択

セグメント、フィールド、フィールド値、非表示アクセス制限

複数ユーザへのセキュリティ制限の適用

DBA またはユーザパスワードの削除

手順/構文

データベース管理者のセキュリティを設定するには

ユーザのセキュリティを設定するには

参照

DBA のガイドライン

DBA ウィンドウ

マスターファイルは、ファイル単位でセキュリティを設定することができます。各データソースは、レベルの異なる 2 つのセキュリティで保守することができます。

- **データベース管理者レベル** データソースにデータベース管理者 (DBA) パスワードを設定します。DBA はマスターファイルとデータソースに制限なくアクセスできます。また、各ユーザのセキュリティ制限を設定および変更することができます。

- **ユーザレベル** データソースに DBA パスワードおよびユーザパスワードを設定します。ユーザパスワードは、データソースにアクセスが可能なユーザを示します。ユーザパスワードを設定する際は、少なくともファイルアクセスのタイプ(読み取り、書き込み、読み取りおよび書き込み)を設定する必要があります。各ユーザのセグメント、フィールド、またはフィールド値へのアクセスをさらに制限し、セキュリティを強化することができます。詳細は、206 ページの「[セグメント、フィールド、フィールド値、非表示アクセス制限](#)」を参照してください。ユーザパスワードを設定した後、同一の制限を複数のユーザに適用することができます。詳細は、209 ページの「[複数ユーザへのセキュリティ制限の適用](#)」を参照してください。


注意：シノニム作成処理の実行中に、データベース管理者(DBA)パスワードを設定することはできません。シノニムエディタを使用する必要があります。

セキュリティが設定されている場合、データソースにアクセスするには、データベース管理者(DBA)またはユーザのパスワードを入力する必要があります。DBA またはユーザがデータソースへのアクセス権限が不要になった場合は、それらのセキュリティを削除することができます。

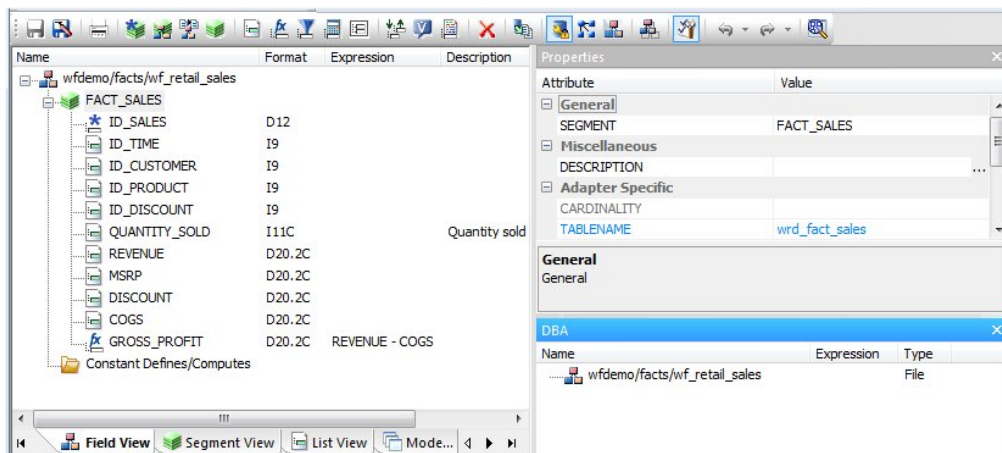
データソースにセキュリティを設定する前に、データベース管理者は DBA のガイドラインを認識する必要があります。詳細は、203 ページの「[DBA のガイドライン](#)」を参照してください。

手順

データベース管理者のセキュリティを設定するには

1. シノニムエディタで、[ツール] (Tools) メニューから [DBA] を選択するか、[シノニム] (Synonym) ツールバーの [DBA]  ボタンをクリックします。

下図のように、ワークスペースに [DBA] ウィンドウが表示されます。



2. [DBA] ウィンドウでファイル名を右クリックし、[挿入] (Insert)、[DBA] を選択します。

マスターファイルのデフォルト DBA パスワードが作成されます。この値は、変更や削除が行えるほか、ファイル制限を指定するためのユーザや、現在のデータソースへのデータソースに特化した制限を指定するためのファイル名を追加することができます。また、DBA セキュリティ制限が格納された別の DBA ファイルを指定することもできます。

注意：パスワードの作成中、カーソルがこのボックス内に表示されている状態で右クリックすると、[編集] (Edit) オプションが表示されます。これらのオプションを使用して、パスワードに対する操作 ([元に戻す] (Undo)、[すべて選択] (Select All)、[切り取り] (Cut)、[貼り付け] (Paste)、または [削除] (Delete)) を実行することができます。

手順

ユーザのセキュリティを設定するには

1. [DBA] ウィンドウで、[DBA] アイコンを右クリックし、ユーザ制限を挿入するか、DBA ファイルを指定します。
2. ユーザの追加後、ユーザのフィールドを右クリックして [挿入] (Insert) を選択し、ファイルアクセス制限の挿入を続行することができます。

3. アクセスのタイプとして、[読み取り] (Read)、[書き込み] (Write)、[読み取り/書き込み] (Read/Write) [更新] (Update) のいずれかを選択します。
4. [フィールド制限] (Restriction to Field)、[値] (Value)、[セグメント] (Segment)、[非表示] (Noprint)、[同一制限] (Same Restriction) のオプションそれぞれに、制限のタイプを指定します。
注意：[同一制限] (Same Restriction) オプションは、複数ユーザが存在するときに有効になります。
5. [OK]をクリックすることでユーザパスワードを含むマスターファイルを保存します。

参照

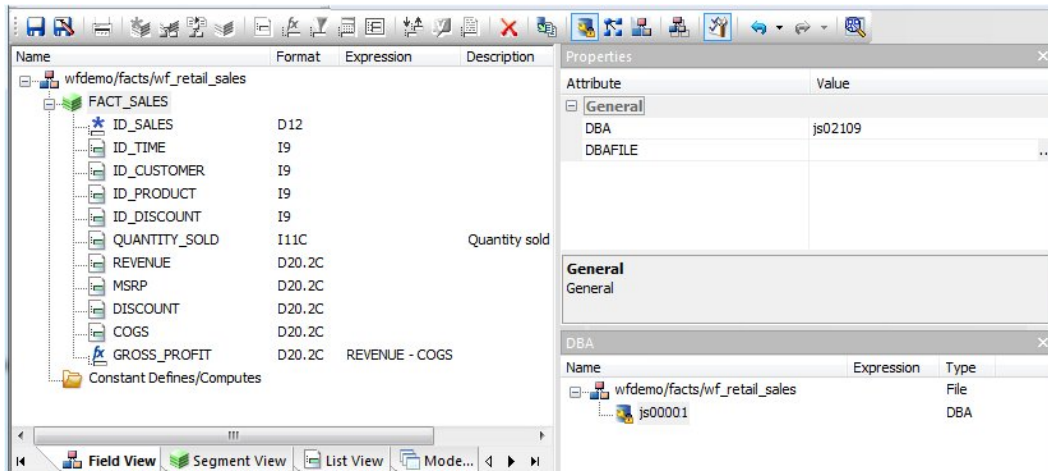
DBA のガイドライン

以下のガイドラインに従うと、マスターファイルに適用するセキュリティ制限を正しく実行することができます。

- アクセス制限を適用するファイルすべてに、DBA パスワードを設定する必要があります。
- セグメント、フィールド、またはフィールド値の制限は、どれもデータベース管理者レベルでは設定されません。データベース管理者は、データソースおよびクロスリファレンスデータソースへは無制限にアクセスできます。
- セキュリティ制限の適用後、データベース管理者は、データソースを使用する前にすべての制限を十分にテストする必要があります。特に、フィールド値がエラーを含んでいないか確認することは重要です。エラーを含んでいる場合、フィールドデータへのユーザアクセス制限は、不適切に設定されます。
- クロスリファレンスデータソースのすべてのグループは、セキュリティ制限を同一にする必要があります。
- データベース管理者は、すべてのセキュリティ制限を変更することができます。
- アクセスレベルは、ユーザがアクセスできるフィールドに影響します。データベース管理者は、ユーザがどのコマンドを使用するか考慮する必要があります。ユーザがアクセス権限を持たない場合、そのユーザはエラーメッセージを受信します。

参照

DBA ウィンドウ



DBA パスワードを選択した場合、[DBA] ウィンドウでは、次のオプションが使用可能です。

DBA パスワード (DBA Password)

デフォルト設定では、DBA パスワードは Reporting Server への接続に使用するユーザ ID と同一です。DBA パスワードのコンテキストメニューで、[名前の変更] (Rename) オプションを使用して、64 バイトまでの異なるパスワードを入力することができます。これは、現在のデータソースを作成、管理するデータベース管理者のパスワードです。

DBA にはソースデータおよび対応するマスターファイルへのフルアクセス権があり、他のユーザのアクセス権を制御します。

DBAFILE

DBA セキュリティ制限を含むマスターファイル名を選択します。このマスターファイルの DBA セキュリティ制限は、他のマスターファイルで使用することができます。

ファイル名の挿入 (Insert Filename)

ユーザセキュリティを適用するマスターファイル名を入力します。このオプションは、データソースに特化した制限を現在のデータソースに追加する場合に使用します。これにより、選択したマスターファイルに FILENAME 属性が追加されます。リファレンスマスターファイルの FILENAME 属性は、現在のデータソースの DBA セクションにある FILENAME 属性と同一である必要があります。

ユーザの挿入 (Insert User)

現在のデータソースへのアクセス権限を許可するユーザ名を、64 バイト以内で入力します。

ファイルアクセス (File Access)

ユーザのアクセス権限として、次のオプションのいずれかを選択します。

- 表示のためのフルアクセス権限を許可するには、[読み取り] (Read) アクセスを選択します。
- データソースの追加および変更を許可するには、[書き込み] (WRITE) アクセスを選択します。
- 上記の両方を許可するには、[読み取り/書き込み] (READ/WRITE) を選択します。
- フィールド値の変更を許可するには、[更新] (UPDATE) を選択します。

制限 (Restrictions) - セグメント (Segment)、フィールド (Field)、値 (Value)、非表示 (Noprint)、同一 (Same)

ファイルアクセスを選択した場合、適用する制限のタイプを選択することができます。

- すべてまたは個別のセグメントへのアクセスを許可するには、[セグメント] (Segment) を選択します。
- すべてまたは個別のフィールドへのアクセスを許可するには、[フィールド] (Field) を選択します。
- テスト条件に一致する値へのアクセスを制限するには、[値] (Value) を選択します。206 ページの「[セグメント、フィールド、フィールド値、非表示アクセス制限](#)」を参照してください。
- レポートで非表示にするフィールドを指定するには、[非表示] (Noprint) を選択します。
- 設定済みの他のユーザと同一の制限を適用するには、[同一] (Same) を選択します。

アクセス制限 (Access Restrictions)

- ユーザ (User)** マスターファイルに書き込むユーザ名です。
- 名前 (Name)** 選択されたマスターファイルコンポーネント名 (例、セグメントまたはフィールド名) です。
- アクセス (Access)** アクセス制限のタイプです。

- **制限 (Restrict)** ファイルアクセス制限のオプションです。
- **値 (Value)** アクセスを制限する値です。

アクセスタイプの選択

ユーザパスワードを割り当てるときは、ファイルアクセス、およびファイル制限オプションが使用できます。少なくとも、ユーザがデータソースへのアクセス権限を持つアクセスタイプを設定する必要があります。ファイルアクセスタイプは、[DBA] ウィンドウの[ファイルアクセス] (File Access) グループで指定することができます。このグループには、4つのファイルアクセスオプションがあります。

- **読み取り (Read)** ユーザにデータソースの読み取り (表示) のみを許可します。
- **書き込み (Write)** ユーザにデータソースの書き込み (追加または変更) のみを許可します。
- **読み取り/書き込み (Read/Write)** ユーザにデータソースの読み取りと書き込みを許可します。

- **更新 (Update)** ユーザに既存のフィールド値の更新 (変更) を許可します。

ファイルアクセスタイプは、ユーザのデータソース全体に対する権限を決定します。

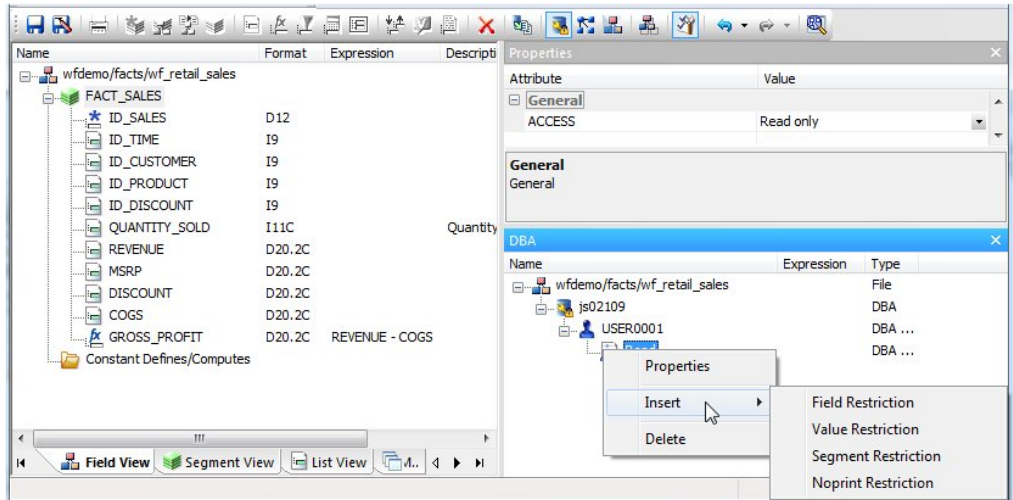
- ファイルアクセスタイプのみを指定する場合は、指定されたユーザのアクセス権限は、データソース全体に適用されます。
- セグメント、フィールド、フィールド値へのアクセス制限をすることにより、制限を強化することができます。206 ページの「[セグメント、フィールド、フィールド値、非表示アクセス制限](#)」を参照してください。

セグメント、フィールド、フィールド値、非表示アクセス制限

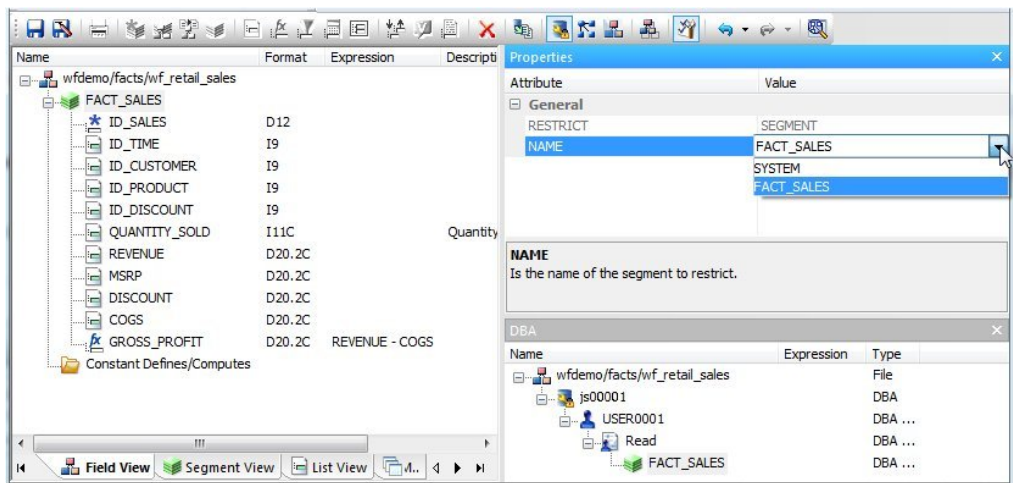
ユーザのアクセス制限を指定することにより、マスターファイルのセグメント、フィールド、フィールド値、NOPRINT フィールドへのアクセスを制限することができます。制限する項目 (セグメント、フィールド、フィールド値) を指定し、次に制限するアクセスタイプを設定します。

ファイルアクセス制限を右クリックし、コンテキストメニューから [セグメント制限] (Segment Restriction)、[フィールド制限] (Field Restriction)、[値制限] (Value Restriction)、[非表示制限] (Noprint Restriction) オプションのいずれかを選択します。

- **セグメント (Segment)** 下図のように、個別セグメントのアクセスタイプを指定します。

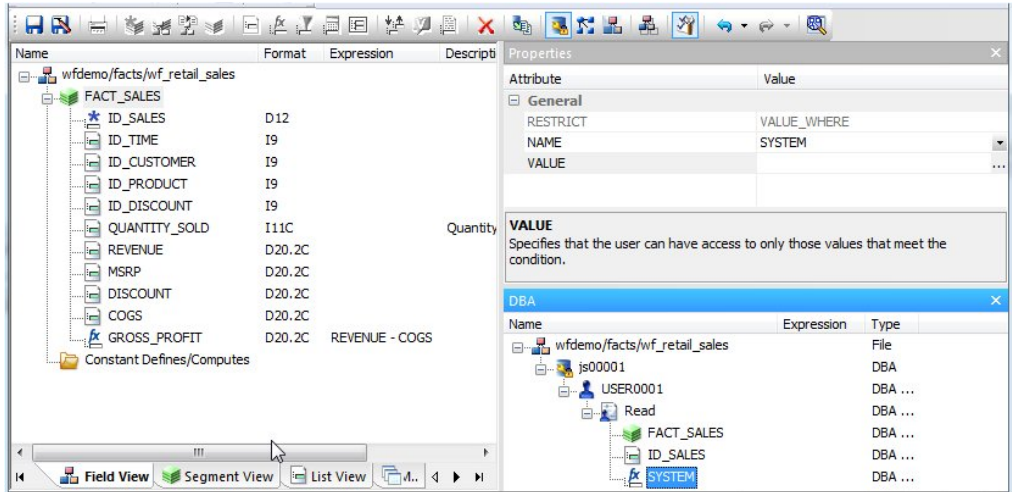


下図は、ユーザがセグメント名を変更する方法を示しています。

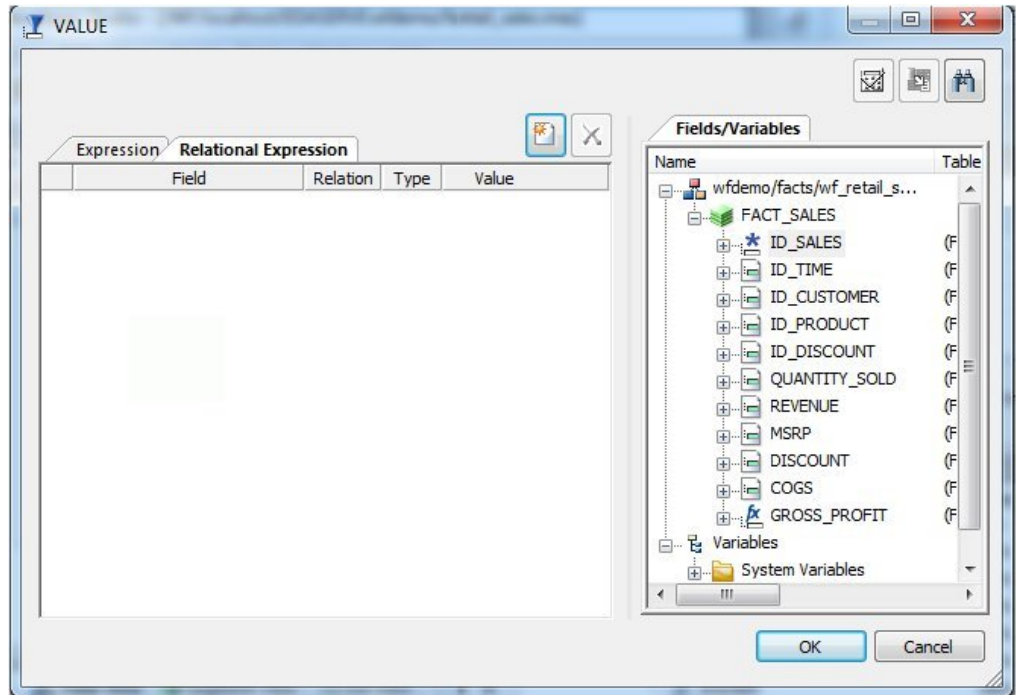


- **フィールド (Field)** 個別のフィールドのアクセスタイプを指定します。
- **値 (Value)** アクセスタイプ (読み取りまたは書き込み) およびテスト条件を指定します。ユーザが使用できる値は、テスト条件に一致するものに制限されます。

下図は、値フィールドで使用するフィールド名を、ユーザが変更する方法を示しています。



下図は、条件の作成方法を示しています。このダイアログボックスは、[値] (Value) テキストボックス横の [...] (参照) ボタンを押した後に取得されています。



- **非表示 (Noprint)** このフィールドに [非表示] (Noprint) を指定して、データを非表示にすることがもできます。フィールドを [非表示] (Noprint) に指定すると、ユーザが検索した場合、文字フォーマットデータはブランクとして表示され、数字フォーマットデータは 0 (ゼロ) として表示されます。

複数ユーザへのセキュリティ制限の適用

手順/構文

定義済みの制限を他のユーザに適用するには

1名のユーザに適用した制限を他のユーザに適用することができます。これは、同一の制限をグループのユーザに適用するとき役に立ちます。

手順

定義済みの制限を他のユーザに適用するには

1. [DBA] ウィンドウで、[DBA パスワード] (DBA password) を右クリックし、[挿入] (Insert)、[ユーザ] (User) を選択します。
2. 新しく追加したユーザを右クリックして [挿入] (Insert) を選択し、適用するアクセス制限のタイプを指定します。
使用可能なアクセスタイプには、[読み取りアクセス] ()、[書き込みアクセス] (Write Access)、[読み取り/書き込み] (Read/Write Access) アクセス、[更新アクセス] (Update Access) があります。
3. アクセスタイプの 1 つを右クリックし、[挿入] (Insert)、[同一制限] (Same Restriction) を選択します。
注意：[同一制限] (Same Restriction) オプションは、複数のユーザが存在する場合にのみ使用可能です。[プロパティ] (Properties) ウィンドウの NAME 属性で、ドロップダウンリストが有効になります。
4. NAME 属性横のドロップダウンリストボックスで、矢印キーをクリックし、新しいユーザに適用するセキュリティ制限を所有するユーザを選択します。
ドロップダウンリストで選択したユーザのセキュリティ制限が、新しいユーザに適用されます。他のユーザにセキュリティ制限を適用する場合は、手順 1 から 4 を繰り返します。
注意：複数のユーザにセキュリティ制限を適用するには、少なくとも 1 つのユーザセキュリティ制限を作成しておく必要があります。

DBA またはユーザパスワードの削除

手順/構文

DBA パスワードを削除するには

DBA パスワードを削除するには

ユーザのセキュリティや DBA パスワードが不要になった場合は、削除することができます。

手順

DBA パスワードを削除するには

1. [DBA] ウィンドウで、削除するユーザのパスワードを選択します。

2. 右クリックして [削除] (Delete) を選択するか、キーボードの Delete キーを押します。
その他のユーザのセキュリティ制限の割り当て元のユーザを削除する場合は、そのユーザに関連付けられたすべてのユーザのセキュリティ制限を再設定する必要があります。

手順

DBA パスワードを削除するには

DBA セキュリティを削除すると、そのデータソースのユーザすべてのセキュリティが削除されます。

[DBA] ウィンドウで、[DBA パスワード] (DBA password) を選択し、右クリックして [削除] (Delete) を選択するか、キーボードの Delete キーを押します。

すべてのセキュリティ情報が削除されます。

4 | メタデータとプロシジャの分析

インパクト分析を使用してプロシジャを分析する方法と、シノニムのフィールドのデータプロファイリングを表示する方法について説明します。

注意：インパクト分析には、DB2 Web Query Developer Workbench 経由でのみアクセスすることができます。

トピックス

- インパクト分析ツールを使用したプロシジャの分析
- データプロファイリング特性の表示

インパクト分析ツールを使用したプロシジャの分析

手順/構文

シノニムエディタからのインパクト分析結果を表示するには
インターフェースからインパクト分析を使用するには

インパクト分析ツールを使用すると、特定のマスターファイルやマスターファイル内のフィールドにアクセスするプロシジャを識別するレポートを生成することができます。このツールにより、マスターファイルやフィールドの変更および削除による影響を分析することができます。インパクト分析ツールは、データ分析や検索条件の制御、レポートの保存が可能のほか、検索条件に基づいてプロシジャを開いて編集する作業を対話的に行うことが可能です。

インパクト分析では、Web Query プロシジャのほかにビジネスビューも検索されます。インパクト分析により、元のマスターファイルの変更が、ビジネスビューに使用されるフィールドに影響を与えるかどうか表示することができます。

インパクト分析ツールには、Developer Workbench からアクセスすることや、シノニムエディタおよびビジネスビュー内のツールからアクセスすることができます。インパクト分析をシノニムエディタまたはビジネスビューから起動した場合は、ファイルは Reporting Server のアプリケーションパスに基づいて検索されます。インパクト分析をインターフェースから起動した場合は、検索対象のアプリケーションおよびドメインを選択することができます。

手順

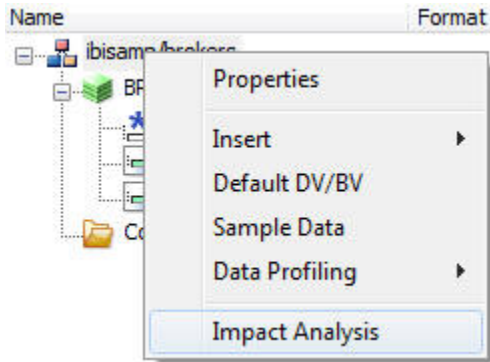
シノニムエディタからのインパクト分析結果を表示するには

インパクト分析をシノニムエディタから起動する際は、ファイルは Reporting Server のアプリケーションパスに基づいて検索されます。

1. [データサーバ] (Data Servers) エリアでマスターファイルをダブルクリックするか、[ファイル] (File) メニューから [シノニムエディタで編集] (Edit in Synonym Editor) を選択します。

シノニムエディタの [ツリー表示] (Tree View) タブに、マスターファイルが開きます。

2. 下図のように、シノニム名またはカラムフィールドを右クリックし、ショートカットメニューから [インパクト分析] (Impact Analysis) を選択します。



注意：インパクト分析は [ビジネスビュー] (Business View) タブから使用することもできます。

レポートスプレッドシートにインパクト分析結果が表示されます。

3. 結果のツールバーを使用して、サーバメッセージの表示、レポートの印刷、データのテキストとしてのコピー、レポートのエクスポートを実行することができます。

	Found In	Found In Application	Found In Type	Usage	Line	Adapter	Table Name Dataset	Syn
1	dblsamp	ibisamp	Procedure	USE	21	Focus/FDS		
2	ldbroke	ibisamp	Procedure	USE	21	Focus/FDS		
3	ldbroke	ibisamp	Procedure	CREATE	27	Focus/FDS		
4	ldbroke	ibisamp	Procedure	MODIFY	30	Focus/FDS		

4. [ファイル] (File) メニューから [保存] (Save) を選択し、インパクト分析結果を保存します。
5. [ファイル] (File) メニューから [閉じる] (Close) を選択し、インパクト分析結果ウィンドウを閉じ、シノニムエディタに戻ります。

手順

インターフェースからインパクト分析を使用するには

1. [データサーバ] (Data Servers) エリアでマスターファイルフォルダからマスターファイルを選択し、[ファイル] (File) メニューから [インパクト分析] (Impact Analysis) を選択します。

ヒント：マスターファイルの 1 つを右クリックし、コンテキストメニューから [インパクト分析] (Impact Analysis) を選択することもできます。このインターフェースでは、検索するアプリケーションまたはフォルダを選択することができます。

インパクト分析ツールが表示されます。[新規レポート] (New Report) タブが表示された状態で、この中には選択した (マスター) ファイルの名前およびデフォルトの検索パスが表示されています。

2. 特定のマスターファイルまたはフィールドをアクセスしているすべてのプロシジャを検索するには、次のいずれかを実行します。

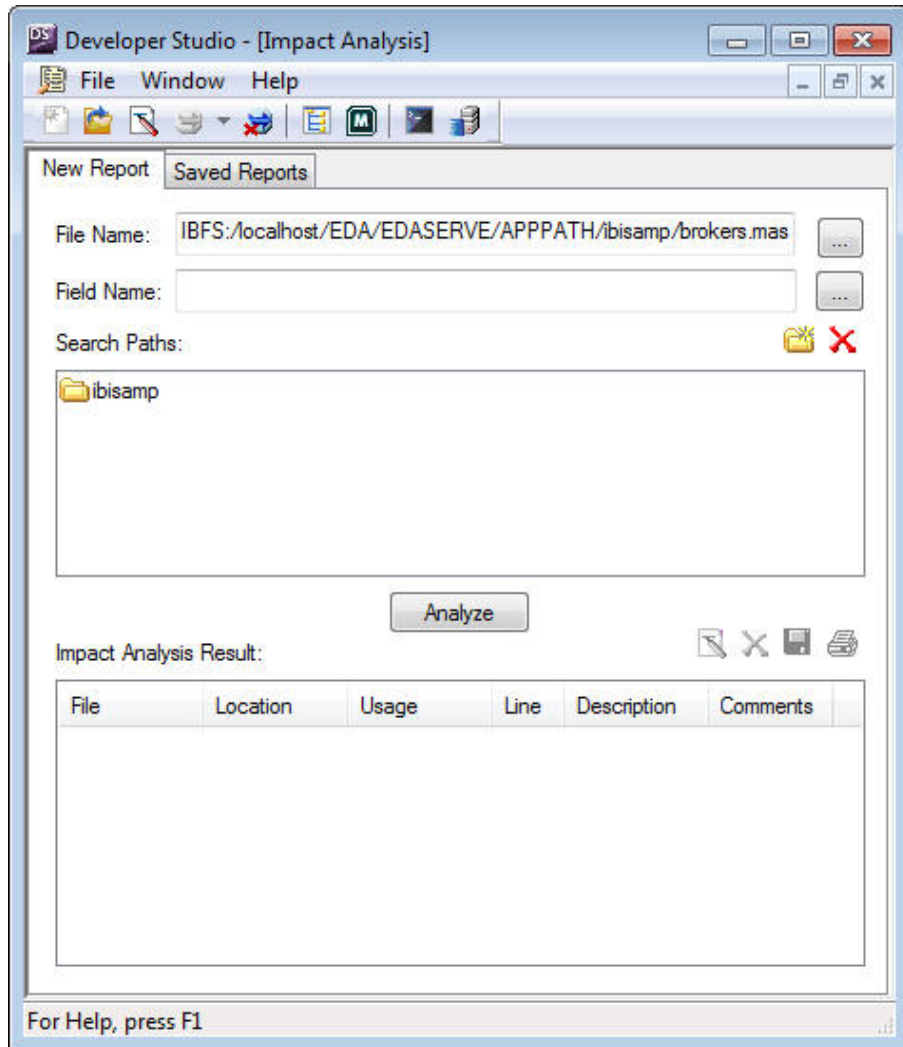
- **選択済みのマスターファイル** デフォルト設定では、インパクト分析ツールを起動する際に選択したマスターファイルが検索されます。手順 3 へ進みます。
- **異なるマスターファイル** [ファイル名] (File Name) 検索ボックス右側の [...] (参照) ボタンをクリックして [開く] (Open) ダイアログボックスを表示し、この中から異なるマスターファイルを選択します。
- **マスターファイル内の単一フィールド** マスターファイルの選択後、[フィールド名] (Field Name) 検索ボックス右側の [...] (参照) ボタンをクリックし、[マスターファイル] (Master File) コンテキストメニューのフィールド名をダブルクリックします。

3. [検索パス] (Search Path) ボックス内にデフォルト設定では表示されていませんが、これ以外のディレクトリパス内のプロシジャを検索する必要がある場合は、[検索パス] (Search Path) ボックス上部のフォルダアイコンをクリックし、表示される [フォルダの参照] (Browse for Folder) ダイアログボックスの中からフォルダを選択することで、検索するフォルダ追加することも可能です。

注意：検索パスを選択し、[検索パス] (Search Path) ボックス上部の [削除] (Delete) アイコンをクリックすることで、検索パスを削除することもできます。

4. [分析] (Analyze) ボタンをクリックし、[インパクト分析の結果] (Impact Analysis Result) ウィンドウにレポートを表示します。

下図は、分析を実行した後の [インパクト分析] (Impact Analysis) ダイアログボックスの [新規レポート] (New Report) タブを示しています。ファイル名および複数検索パスが入力され、[インパクト分析の結果] (Impact Analysis Result) ウィンドウにレポートが表示されています。



分析の実行後、プロシジャの実行、レポートの印刷、レポートからの項目の削除、レポートのエクスポートが可能です。

レポートは XML フォーマットでエクスポートされます。このファイルには .IAR という拡張子が付けられ、他の場所を指定しない限り、次のデフォルトディレクトリに保存されます。

```
drive:¥ ibi¥ DevStudioreleasenumber¥ bin
```

5. [保存済みレポート] (Saved Reports) タブをクリックすることで、以前に作成したレポートすべてにアクセスすることができます。

インパクト分析レポートは、すべて次の XML フォーマットファイルに自動的に保存されます。

```
drive:¥  
ibi¥  
DevStudioreleasenumber¥  
bin¥  
IARepository.xml
```

新しい分析レポートを実行するたびに、情報はこのファイルに追加されます。レポート操作のオプションでは、レポートの保存、以前に保存したレポートを開く、レポートの削除が可能です。

データプロファイリング特性の表示

トピックス

シノニムまたはセグメントのデータプロファイリング

単一フィールドのデータプロファイリング

シノニム内のフィールドのデータ特性は、データプロファイリングとして表示することができます。シノニムまたはセグメント内のすべてのフィールドの特性を表示することも、特定のフィールドの特性を個別に表示することもできます。

注意：アダプタを適切に構成しないと、データプロファイリングは使用できない場合があります。

文字フィールドの場合、データプロファイリングで生成される情報には、セグメント、フォーマット、件数 (種類)、件数、パターン数、(長さの) 最大値、(長さの) 最小値、(長さ) 平均、NULL 数があります。パターン数には、文字フィールド別に検索されたパターン件数が表示されます。

数値フィールドの場合、データプロファイリングが提供する情報には、セグメント、フォーマット、件数 (種類)、件数、パターン数、最大値、最小値、平均値、NULL 数があります。

各フィールドのデータプロファイリングからは、統計、パターン、値、異常値の各レポートへアクセスすることができます。

シノニムまたはセグメントのデータプロファイリング

手順/構文

シノニムまたはセグメントのデータプロファイリングを表示するには

データプロファイリングには、シノニムまたはセグメント内のすべてのフィールド情報が表示されます。また、シノニムまたはセグメントのデータプロファイリングレポートからは、個々のフィールドの値レポートまたはパターンレポートにドリルダウンすることができます。

手順

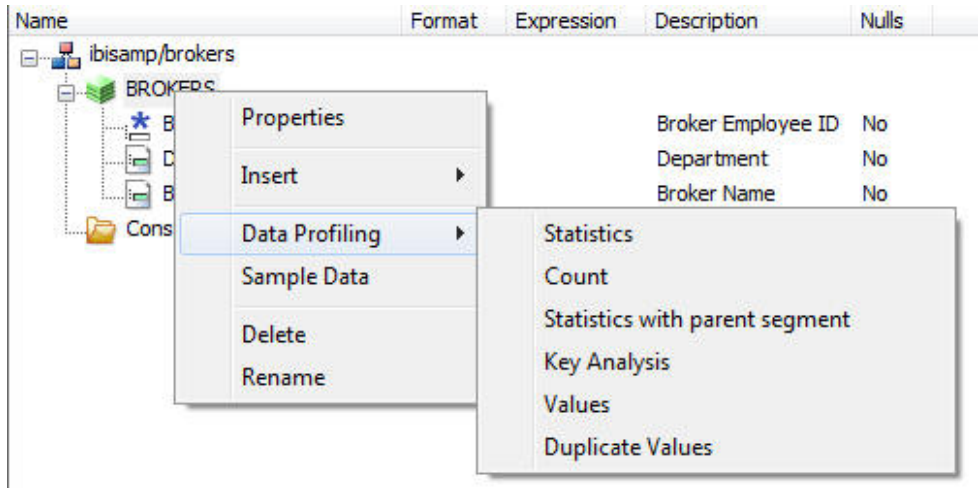
シノニムまたはセグメントのデータプロファイリングを表示するには

シノニムまたはセグメントのデータプロファイリング情報を表示するには、次の手順を実行します。

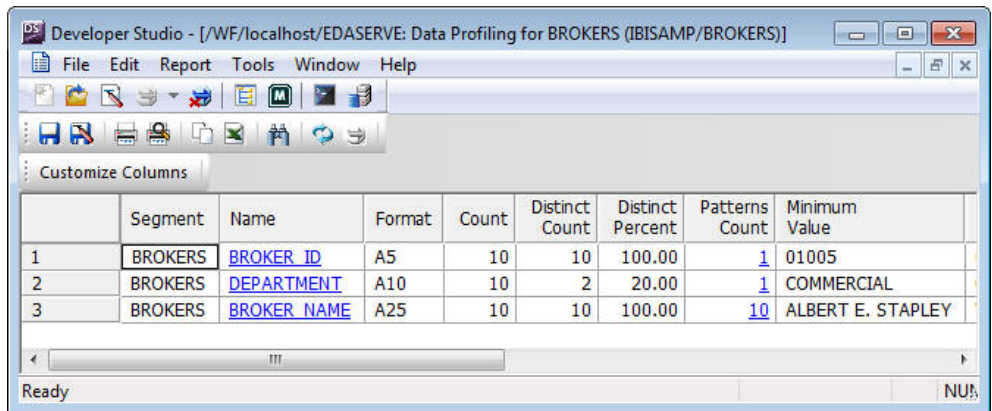
1. [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。

シノニムエディタの [フィールド表示] (Field View) タブが表示されます。

- 下図のように、シノニムまたはセグメント名を右クリックして [データプロファイリング] (Data Profiling) を選択し、[統計] (Statistics)、[件数] (Count)、[親セグメント付き統計] (Statistics with parent segment)、[キー分析] (Key Analysis)、[値] (Values)、[値の重複] (Duplicate Values) のいずれかを選択します。



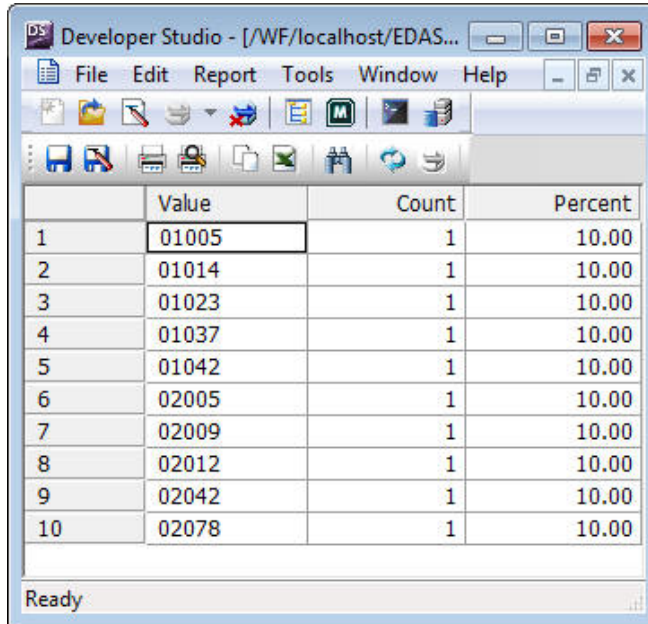
ワークスペースにデータプロファイリング情報が表示されます。



データプロファイリングの結果のツールバーを使用し、サーバメッセージの表示、レポートの印刷、データのテキストとしてのコピー、レポートのエクスポートを実行することができます。

- 必要に応じて、[名前] (Name) または (文字フィールドの) [パターン数] (Patterns Count) をクリックして、値レポートまたはパターンレポートにドリルダウンすることができます。

下図は、[名前] (Name) をクリックして生成された値レポートの一部を示しています。

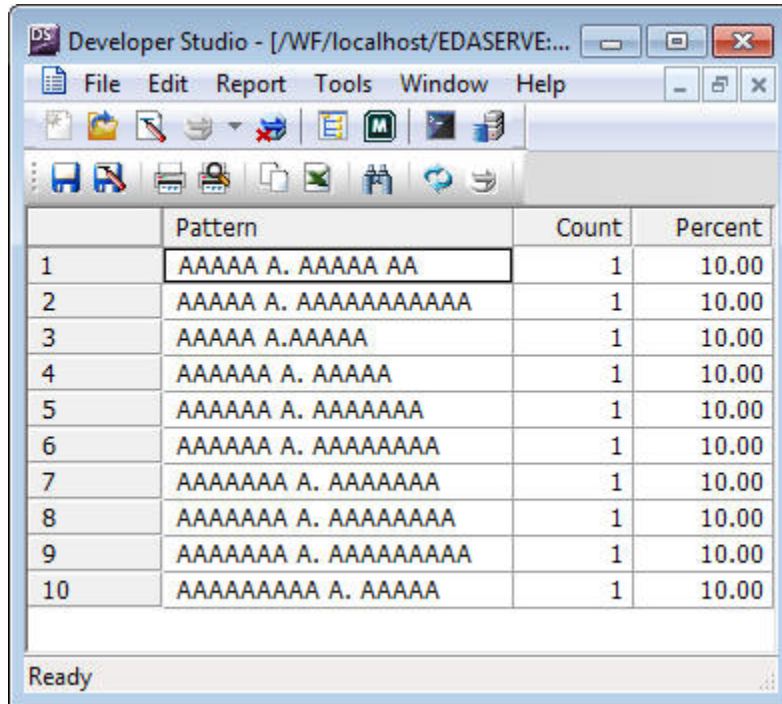


The screenshot shows a window titled "Developer Studio - [/WF/localhost/EDAS...]" with a menu bar (File, Edit, Report, Tools, Window, Help) and a toolbar. Below the toolbar is a table with the following data:

	Value	Count	Percent
1	01005	1	10.00
2	01014	1	10.00
3	01023	1	10.00
4	01037	1	10.00
5	01042	1	10.00
6	02005	1	10.00
7	02009	1	10.00
8	02012	1	10.00
9	02042	1	10.00
10	02078	1	10.00

The status bar at the bottom of the window displays "Ready".

下図は、Address フィールドの例を示しています。



The screenshot shows a window titled "Developer Studio - [/WF/localhost/EDASERVE...]" with a menu bar (File, Edit, Report, Tools, Window, Help) and a toolbar. Below the toolbar is a table with the following data:

	Pattern	Count	Percent
1	AAAAA A. AAAAA AA	1	10.00
2	AAAAA A. AAAAAAAAAAAAA	1	10.00
3	AAAAA A.AAAAA	1	10.00
4	AAAAAA A. AAAAA	1	10.00
5	AAAAAA A. AAAAAAA	1	10.00
6	AAAAAA A. AAAAAAAA	1	10.00
7	AAAAAA A. AAAAAAA	1	10.00
8	AAAAAA A. AAAAAAAA	1	10.00
9	AAAAAA A. AAAAAAAA	1	10.00
10	AAAAAA A. AAAAA	1	10.00

Ready

パターン解析では、「9」は数字、「A」は任意の大文字、「a」は任意の小文字を表します。表示可能なすべての特殊文字はそのまま表示されますが、表示不可の文字は「X」で表示されます。

単一フィールドのデータプロファイリング

手順/構文

データプロファイリングの統計を表示するには

データプロファイリングのパターンを表示するには

データプロファイリングの値を表示するには

データプロファイリングの値グラフを表示するには

データプロファイリングの重複値を表示するには

データプロファイリングの異常値を表示するには

単一フィールドのデータプロファイリングでは、次の 4 つのレポートにアクセスすることができます。

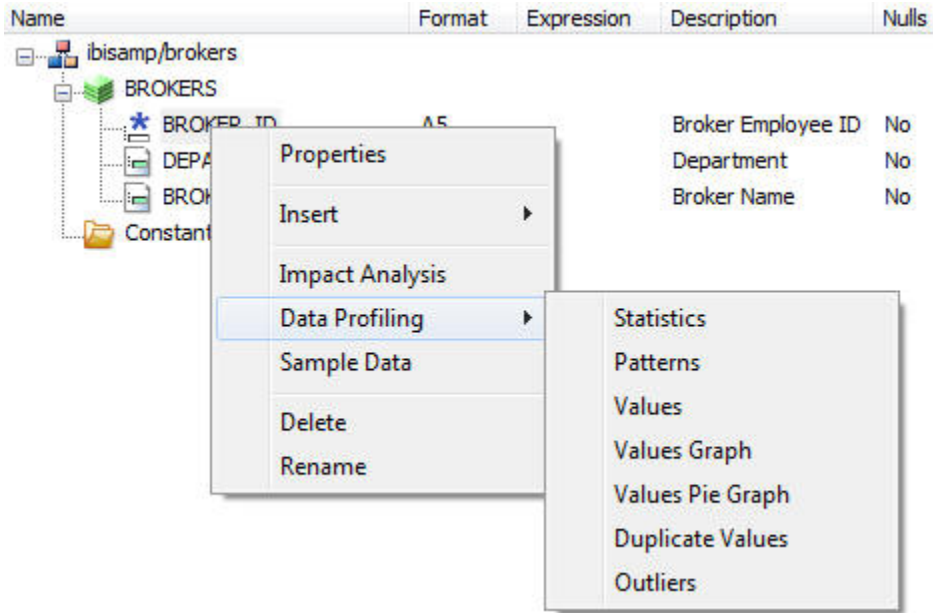
- **統計** シノニムまたはセグメントのデータプロファイリングレポートと同一の情報を表示します。

文字フィールドの場合、統計レポートで生成される情報には、セグメント、フォーマット、件数(種類)、件数、パターン数、(長さの)最大値、(長さの)最小値、(長さ)平均、NULL 数があります。

数値フィールドの場合、統計レポートは、セグメント、フォーマット、件数(種類)、件数、パターン数、最大値、最小値、平均値、NULL 数を表示します。

- **パターン** 文字フィールドのみで使用可能です。文字、数字、特殊文字のパターンおよび件数とそのパーセントを表示します。
- **値** ユニーク値およびそのパーセントを表示します。
- **値グラフ** 文字フィールドのグラフを表示します。
- **値の重複** 重複する値を表示します。
- **異常値** 上位および下位それぞれ 10 個の値とその件数を表示します。

下図のように、これらのレポートを表示するには、シノニムエディタでフィールドの1つを右クリックし、[データプロファイリング] (Data Profiling) を選択します。



手順

データプロファイリングの統計を表示するには

単一フィールドの統計データプロファイリング情報を表示するには、次の手順を実行します。

1. [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。
シノニムエディタの [フィールド表示] (Field View) タブが表示されます。
2. フィールドの1つを右クリックし、[データプロファイリング] (Data Profiling) のコンテキストメニューから [統計] (Statistics) を選択します。

下図のように、ワークスペースに統計データプロファイリング情報が表示されます。

	Segment	Name	Format	Count	Distinct Count	Distinct Percent	Patterns Count
1	BROKERS	BROKER_ID	A5	10	10	100.00	1

- 必要に応じて、[名前] (Name) または (文字フィールドの) [パターン数] (Patterns Count) をクリックして、値レポートまたはパターンレポートにドリルダウンすることができます。

手順

データプロファイリングのパターンを表示するには

データプロファイルパターンには、文字、数字、特殊文字、件数、パーセントが表示されます。

単一フィールドのパターンのデータプロファイリング情報を表示するには、次の手順を実行します。

- [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。

シノニムエディタの [フィールド表示] (Field View) タブが表示されます。

- フィールドの 1 つを右クリックし、[データプロファイリング] (Data Profiling) のコンテキストメニューから [パターン] (Patterns) を選択します。

ワークスペースにパターンデータプロファイリング情報が表示されます。

- 必要に応じて、パターンの件数をクリックし、実際のパターンを表示することができます。

パターン解析では、「9」は数字、「A」は任意の大文字、「a」は任意の小文字を表します。表示可能なすべての特殊文字はそのまま表示されますが、表示不可の文字は「X」で表示されます。

手順

データプロファイリングの値を表示するには

データプロファイリング値には、ユニーク値が表示されます。

単一フィールドの値のデータプロファイリング情報を表示するには、次の手順を実行します。

1. [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。
シノニムエディタの [フィールド表示] (Field View) タブが表示されます。
2. フィールドの 1 つを右クリックし、[データプロファイリング] (Data Profiling) のコンテキストメニューから [値] (Value) を選択します。
値のデータプロファイリング情報が表示されます。

手順

データプロファイリングの値グラフを表示するには

[データプロファイリング値グラフ] (Data Profile Values Graph) が表示されます。

1. [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。
シノニムエディタの [フィールド表示] (Field View) タブが表示されます。
2. フィールドの 1 つを右クリックし、[データプロファイリング] (Data Profiling) のコンテキストメニューから [値グラフ] (Values Graph) を選択します。
値グラフデータプロファイリング情報が表示されます。

手順

データプロファイリングの重複値を表示するには

[データプロファイリング] (Data Profile) の [値の重複] (Duplicate Values) には、重複する値が表示されます。

単一フィールドの重複値のデータプロファイリング情報を表示するには、次の手順を実行します。

1. [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。
シノニムエディタの [フィールド表示] (Field View) タブが表示されます。

2. フィールドの 1 つを右クリックし、[データプロファイリング] (Data Profiling) のコンテキストメニューから [値の重複] (Duplicate Values) を選択します。

重複値のデータプロファイリング情報が表示されます。

手順

データプロファイリングの異常値を表示するには

データプロファイリングの異常値には、上位および下位それぞれ 10 個の DISTINCT 値が表示されます。

単一フィールドの異常値のデータプロファイリング情報を表示するには、次の手順を実行します。

1. [データサーバ] (Data Servers) エリアで、[マスターファイル] (Master Files) フォルダ内のマスターファイルをダブルクリックしてシノニムを開きます。

シノニムエディタの [フィールド表示] (Field View) タブが表示されます。

2. フィールドの 1 つを右クリックし、[データプロファイリング] (Data Profiling) のコンテキストメニューから [異常値] (Outliers) を選択します。

異常値のデータプロファイリング情報が表示されます。

注意：異常値には、存在する場合、上位および下位それぞれ最大 10 個の DISTINCT 値が表示されます。

5 関数の使用

ここでは、関数の概要および使用方法について説明します。関数は、計算および操作をする上での手段を提供します。

ここで説明する関数を使用するには、後述する一時項目を作成する必要があります。

トピックス

- ❑ 関数カテゴリ
- ❑ 関数の引数指定
- ❑ 一時項目での関数の使用
- ❑ 文字列関数
- ❑ データソースおよびデコード関数
- ❑ 日付時間関数
- ❑ フォーマット変換関数
- ❑ 数値関数
- ❑ システム関数

関数カテゴリ

トピックス

文字列関数

ソースデータおよびデコード関数

日付時間関数

フォーマット変換関数

数値関数

システム関数

アクセス可能な関数の種類は次のとおりです。

- **文字列関数** 文字フィールドまたは文字列を操作します。
- **データソースおよびデコード関数** データソースの値を抽出し、入力フィールドの値に基づいて値を割り当てます。
- **日付および日付時間関数** 日付および時間を操作します。
- **フォーマット変換関数** フィールドのフォーマットを変換します。
- **数値関数** 数値定数と数値フィールドの計算を実行します。
- **システム関数** オペレーティングシステムを呼び出して、オペレーティング環境についての情報を取得します。

文字列関数

次の関数は、文字フィールドまたは文字列を操作します。

ARGLEN

フィールド内の末尾のブランクを除いた文字列の長さを取得します。

BITSON

文字列内の特定のビットを評価し、オンかオフかを返します。

BYTVAL

文字列を ASCII または EBCDIC の 10 進数に変換します。

CHKFMT

文字列内の文字または文字種が正しいかを確認します。

CTRAN

文字列内の文字を 10 進数に基づいて他の文字に変換します。

CTRFLD

文字列をフィールド内で中央揃えにします。

EDIT

文字列から文字を抽出、または文字列に文字を追加します。

GETTOK

文字列を「トークン」と呼ばれる区切り文字で分割します。

LCWORD

文字列内の先頭文字を大文字、それ以外を小文字に変換します。

LCWORD2

文字列の先頭の文字以外をすべて小文字に変換します。

LCWORD3

文字列内の文字を先頭大文字に変換します。この関数で、各文字列の先頭文字が大文字に、その他すべての文字が小文字に変換されます。

LJUST

文字列をフィールド内で左揃えにします。

LOCASE

文字列を小文字に変換します。

OVLAY

文字列を他の文字列で上書きします。

PARAG

テキスト行を区切り文字で分割します。

POSIT

文字列内の特定の文字列の開始位置を見つけます。

PTOA

数値形式を文字形式に変換します。数値の小数点の位置を保持し、先頭に空白を挿入することにより、数値を右揃えします。PTOA によって変換された数値には、編集オプションを追加することができます。

REVERSE

文字列内の文字を逆にします。

RJUST

文字列を右揃えにします。

SOUNDEX

文字列を、綴りとは無関係に音声で検索します。

SPELLNUM

小数点以下 2 桁の文字または数値をドルとセントの文字表記に書き替えます。

SUBSTR

文字列内の開始位置および長さに基づいて、文字列を抽出します。

UPCASE

文字列を大文字に変換します。

ソースデータおよびデコード関数

次の関数は、データソースの値を取得し、値を割り当てます。

DECODE

コード化された入力フィールドの値に基づいて値を割り当てます。

LAST

フィールドの前の値を取得します。

日付時間関数

トピックス

標準日付時間関数

次の関数は、日付および時間を操作します。

標準日付時間関数

AYM

年月フォーマットの日付と指定した月数の和または差を計算します。

AYMD

年月日フォーマットの日付と指定した日数の和または差を計算します。

DAT

文字フォーマットの日付の年月日部分を再配置し、長いフォーマットと短いフォーマット間の変換を実行します。

DA

日付を 1899 年 12 月 31 日から数えた経過日数に変換します。

DADMY - 日月年フォーマットに変換します。

DADYM - 日年月フォーマットに変換します。

DAMDY - 月日年フォーマットに変換します。

DAMYD - 月年日フォーマットに変換します。

DAYDM - 年日月フォーマットに変換します。

DAYMD - 年月日フォーマットに変換します。

DATEADD

日付フォーマットに単位を追加、または日付フォーマットから単位を削除します。

DATECVT

日付フォーマットを変換します。

DATEDIF

2つの日付の差を単位で返します。

DATEMOV

日付を特定の日付に移動します。

DATEPATTERN

日付の値を文字フォーマットで格納します。特別な標準は存在せず、年、四半期、月などの構成要素の任意の組み合わせ、および任意の区切り文字を使用します。

DATETRAN

日付を国際フォーマットに変換します。

DMY、MDY、YMD

2つの日付の差を計算します。

DOWK、DOWKL

日付に対応する曜日を検索します。

DT

1899年12月31日から数えた経過日数を日付に変換します。

DTDMY - 数値を日月年の日付に変換します。

DTDYM - 数値を日年月の日付に変換します。

DTMDY - 数値を月日年の日付に変換します。

DTMYD - 数値を月年日の日付に変換します。

DTYDM - 数値を年日月の日付に変換します。

DTYMD - 数値を年月日の日付に変換します。

FIYR

会計年度の開始日および会計年度の算定方式に基づいて、特定のカレンダー日付に対応する会計年度を返します。

FIQTR

会計年度の開始日および会計年度の算定方式に基づいて、特定のカレンダー日付に対応する会計四半期を返します。

FIYYQ

指定したカレンダー日付に対応する会計日付を返します。この日付には、会計年度および会計四半期が含まれます。

GREGDT

ユリウス暦の日付を年月日フォーマットに変換します。

HADD

日付時間フィールドを指定した単位数、増加します。

HCNVRT

日付時間フィールドを文字列に変換します。

HDATE

日付時間フィールドの日付部分を抽出し、日付フォーマットに変換後、結果を YYMD フォーマットで返します。

HDIFF

2つの日付時間値の差を単位数で計算します。

HDTTM

日付フィールドを日付時間フィールドに変換します。時間部分は午前零時に設定されます。

HGETC

現在の日付と時間を日付時間フォーマットで格納します。

HMMSS

システムから現在の時間を取得します。

HINPUT

文字列を日付時間値に変換します。

HMIDNT

日付時間フィールドの時間部分を午前零時 (すべてゼロ) に変更します。

HNAME

日付時間フィールドから特定の構成要素を抽出し、文字フォーマットで返します。

HPART

日付時間フィールドから特定の構成要素を抽出し、数値フォーマットで返します。

HSETPT

指定した構成要素の数値を日付時間値に挿入します。

HTIME

日付時間フィールドの時間部分をミリ秒またはマイクロ秒に変換します。

JULDAT

年月日フォーマットの日付をユリウス暦 (年月) フォーマットに変換します。

TIMETOTS

時間をタイムスタンプに変換します。

TODAY

システムから現在の日付を取得します。

YM

日付から日付までの経過月数を計算します。日付は年月フォーマットである必要があります。

フォーマット変換関数

次の関数は、フィールドのフォーマットを変換します。

ATODBL

文字フォーマットの数値を倍精度フォーマットに変換します。

EDIT

数値を含む文字フィールドを数値フォーマットに変換、または数値フィールドを文字フォーマットに変換します。

FTOA

数値フォーマットの数値を文字フォーマットに変換します。

HEXBYT

ASCII または EBCDIC の 10 進コードに対応する文字を取得します。

ITONUM

データソースの整数を倍精度フォーマットに変換します。

ITOPACK

データソースの整数をパック 10 進数フォーマットに変換します。

ITOZ

数値フォーマットの数値をゾーン 10 進数フォーマットに変換します。

PCKOUT

抽出ファイルに指定した長さでパック 10 進数を書き込みます。

数値関数

次の関数は、数値定数または数値フィールドに対する計算を実行します。

ABS

数値の絶対値を返します。

BAR

横棒グラフを生成します。

CHKPCK

フィールド内のデータがパック 10 進フォーマットかどうかを確認します。

DMOD、FMOD、IMOD

除算の剰余を計算します。

EXP

値「e」を指数でべき乗します。

INT

数値の整数部分を返します。

LOG

数値の自然対数を返します。

MAX、MIN

値リストから最大値、最小値をそれぞれ返します。

PRDNOR、PRDUNI

再生可能な乱数を生成します。

RDNORM、RDUNIF

乱数を生成します。

SQRT

数値の平方根を計算します。

システム関数

次の関数は、オペレーティングシステムを呼び出して、オペレーティング環境についての情報を取得します。

FGETENV

オペレーティングシステムの環境変数値を取得し、文字列として返します。

GETUSER

接続ユーザの ID を取得します。

HHMMSS

システムから現在の時間を取得します。

TODAY

システムから現在の日付を取得します。

関数の引数指定**トピックス**

引数の種類

関数引数の増加

引数のフォーマット

関数に引数を指定する際は、使用可能な引数の種類、フォーマットと長さ、数と順序を理解する必要があります。

引数の種類

関数では、次の引数を使用できます。

- 数値定数 (例、6 や 15 など)。
- 日付定数 (例、022802)。
- 日付 (文字、数値、日付、AnV フォーマット)。
- 文字リテラル (例、STEVENS、NEW YORK NY)。リテラルは一重引用符 (') で囲みます。
- 文字フォーマットの数値。

- フィールド名 (例、FIRST_NAME や HIRE_DATE)。フィールドには、データソースフィールドまたは一時項目も使用できます。フィールドは 66 バイト以内のフィールド名または完全修飾名、一意の省略名、あるいはエイリアスです。
- 式 (例、数値、日付、または文字で構成される式)。式には算術演算子と連結記号 (||) が使用できます。有効な式の例は次のとおりです。

```
CURR_SAL * 1.03
```

```
FN || LN
```

- 一重引用符 (!) で囲んだ出力値フォーマット。
- 他の関数。

関数引数の増加

ユーザが記述するサブルーチンでサポートされる引数の数が、28 個から 200 個に増加しています。ユーザが記述するサブルーチンの引数に関するその他の規則に、変更はありません。

引数のフォーマット

引数のフォーマットは関数ごとに異なります。フォーマットには、文字、数値、日付があります。引数のフォーマットに誤りがあると、エラーが発生したり、正しい結果が得られない原因になります。有効なフォーマットを取得するには、ツールバーで [フォーマット] (Format) ボタンをクリックします。有効なタイプと長さが表示されます。引数フォーマットの種類は次のとおりです。

- **文字引数** 内部的には 1 バイトにつき 1 英数文字として格納されます。文字引数には、リテラル、文字フィールド、文字フォーマットで格納された数値または日付、文字式、文字フィールドのフォーマットがあります。
- **数値引数** 内部的には 2 進数またはパック 10 進数として格納されます。数値引数のフォーマットには、整数 (I)、単精度浮動小数点数 (F)、倍精度浮動小数点数 (M)、パック 10 進数 (P) などがあります。数値引数は、数値定数フィールド、フィールド、式、または数値フィールドのフォーマットをとることもできます。数値引数は、関数で使用する際、すべて倍精度浮動小数点数に変換されますが、結果は出力フィールドで指定したフォーマットで返されます。

- **日付引数** 日付引数のフォーマットは、文字、数値、日付のいずれかです。関数で使用可能なフォーマットの種類は、個々の関数の引数リストで指定します。日付引数のフォーマットは、文字、数値、日付のいずれかです。日付フィールド、式、または日付フィールドのフォーマットをとることもできます。引数に 2 桁の年を指定すると、世紀の値は、DATEFNS、YRTHRESH、DEFCENT パラメータ設定に基づいて割り当てられます。

一時項目での関数の使用

ここで説明する関数を使用するには、後述する一時項目を作成する必要があります。

一時項目とは、その値自体はデータソースに保存されていないが、既存のデータから計算を行ったり、絶対値を割り当てたりできるフィールドです。一時項目を作成する場合、式を記述してその値を生成します。1つの値を生成するために、式にはフィールド、定数、演算子を組み合わせて使用することができます。たとえば、給与額および控除額で構成されるデータがある場合、「控除額/給与額」という式を使用して給与額に対する控除の割合を計算することができます。

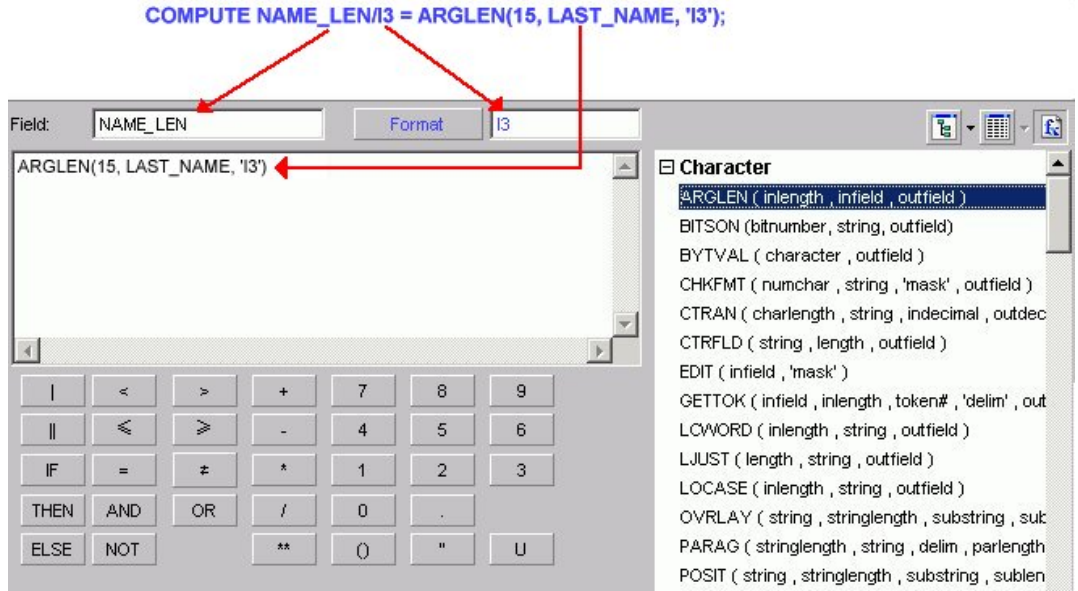
ユーザが独自に式を指定することもできますが、特定の計算や操作を行うために用意されたさまざまな関数の中から必要なものを選択することもできます。さらに、単純な式および関数を構成要素としてさらに複雑な式を作成したり、特定の一時項目を使用してその他の一時項目の評価を行ったりすることも可能です。[フィールドの選択] (Field selection) タブで一時項目を作成するには、DEFINE ツールまたは COMPUTE ツールを選択します。

この章で紹介する例では、テキストエディタで表示した場合と同様に、一時項目 (COMPUTE) の完全な構文を記載しています。以下はその例です。COMPUTE NAME_LEN/I3 = ARGLEN(15, LAST_NAME, 'I3');

COMPUTE ツールを使用して一時項目を作成する場合は、次の規則が適用されます。

- 式に COMPUTE を含める必要はありません。この語句は暗黙的に追加されます。
- 式の終了を示すセミコロン (;) を含める必要はありません。セミコロンは暗黙的に追加されます。
- 複数の式を作成する場合に、AND を含める必要はありません。AND は暗黙的に追加されます。COMPUTE 式は、それぞれ別の一時項目として作成する必要があります。

下図は、InfoAssist で ARGLEN 関数を使用した場合の例を、完全な構文で示しています。COMPUTE ツールで、フィールド名に NAME_LEN、フォーマットに I3 が指定され、[式] (Expression) ウィンドウには「ARGLEN(15, LAST_NAME, 'I3）」と記述されています。



文字列関数

トピックス

ARGLEN - 文字列の長さを取得

BITSON - ビットのオンとオフを返す

BYTVAL - 文字を 10 進数に変換

CHKFMT - 文字列のフォーマットを確認

CTRAN - 文字を他の文字に変換

CTRFLD - 文字列を中央揃え

EDIT - 文字を抽出または追加

GETTOK - 文字列 (トークン) を抽出

LCWORD - 文字列の先頭文字を大文字に変換

LCWORD2 - 文字列の先頭文字以外を小文字に変換

LCWORD3 - 文字列の先頭文字以外を小文字に変換

LJUST - 文字列を左揃え

LOCASE - テキストを小文字に変換

OVLAY - 文字列を上書き

PARAG - テキストを行に分割

POSIT - 文字列の開始位置を検索

PTOA - パック 10 進数を文字に変換

REVERSE - 文字列の順序を入れ替え

RJUST - 文字列を右揃え

SOUNDEX - 文字列を音声的に比較

SPELLNM - ドルとセントの通貨表記を文字表記に書き替え

SUBSTR - 文字列を抽出

UPCASE - テキストを大文字に変換

文字列関数は、文字と数字を含むフィールドと文字列を操作します。

ARGLEN - 文字列の長さを取得

手順/構文

文字列の長さを取得

ARGLEN 関数は、フィールド内の末尾のブランクを除いた文字列の長さを取得します。マスターファイル内のフィールドフォーマットでは、末尾のブランクを含めた長さを指定します。

構文 **文字列の長さを取得**

```
ARGLEN(inlength, infield, 'outfield')
```

説明

inlength

整数

文字列を含むフィールドの長さです。長さを含むフィールドを指定することもできます。

infield

文字

文字列を含むフィールド名です。

outfield

整数

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 **文字列の長さを取得**

ARGLEN は LAST_NAME の文字列の長さを取得し、結果を NAME_LEN に格納します。

```
COMPUTE NAME_LEN/I3 = ARGLEN(15, LAST_NAME, 'I3');
```

BITSON - ビットのオンとオフを返す

手順/構文

ビットのオンとオフを返す

BITSON 関数は、文字列内の特定のビットを評価し、オンかオフかを返します。BITSON は、ビットがオンの場合は値 1 を返し、オフの場合は値 0 (ゼロ) を返します。この関数は、マルチパンチデータの解釈に役立ちます。ここで、各パンチは情報の 1 項目を表します。

構文 **ビットのオンとオフを返す**

```
BITSON(bitnumber, string, 'outfield')
```

説明

bitnumber

整数

評価するビット数です。ビット数は文字列の最も左にあるビットから数えられます。

string

文字

評価する文字列です。文字列は一重引用符 (') で囲みます。文字列を含むフィールドを指定することもできます。文字列は、複数の 8 ビットブロックで構成されます。

outfield

整数

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 **フィールド内のビットを評価**

BITSON は LAST_NAME の 24 ビット目を評価し、結果を BIT_24 に格納します。

```
COMPUTE BIT_24/I1 = BITSON(24, LAST_NAME, 'I1');
```

BYTVAL - 文字を 10 進数に変換

手順/構文

文字を変換

BYTVAL 関数は、文字列をオペレーティングシステムに対応する ASCII または EBCDIC の 10 進数に変換します。

構文

文字を変換

```
BYTVAL(character, 'outfield')
```

説明

character

文字

変換される文字です。文字を含むフィールド、または一重引用符 (') で囲んで文字自体を指定することができます。複数の文字を指定すると、先頭の文字が評価されません。

outfield

整数

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例

フィールドの 1 文字目を変換

BYTVAL 関数は、LAST_NAME の 1 文字目に対応する ASCII または EBCDIC の 10 進数に変換し、結果を LAST_INIT_CODE に格納します。入力文字列には複数の文字が含まれているため、BYTVAL 関数は 1 文字目を評価します。

```
COMPUTE LAST_INIT_CODE/I3 = BYTVAL(LAST_NAME, 'I3');
```

CHKFMT - 文字列のフォーマットを確認

手順/構文

文字列のフォーマットを確認

CHKFMT 関数は、文字列内の文字または文字種が正しいかを確認します。第 1 文字列を第 2 文字列 (mask) と比較し、第 1 文字列の各文字を mask 内の対応する各文字と比較します。文字列内のすべての文字が mask 内の文字列または文字種と一致する場合、値 0 (ゼロ) を返します。それ以外の場合、CHKFMT は mask と一致しない文字列の先頭文字の位置と同一の値を返します。

mask が文字列よりも短い場合、この関数は mask に対応する部分の文字列のみを確認します。たとえば、4 バイトの mask を使用して 9 バイトの文字列をテストする場合、最初の 4 バイトのみが確認され、それ以外は不一致と見なされ、不一致部分の先頭位置が返されます。

構文

文字列のフォーマットを確認

```
CHKFMT(numchar, string, 'mask', 'outfield')
```

説明

numchar

整数

mask と比較する文字のバイト数です。

string

文字

確認する文字列です。文字列は一重引用符 (') で囲みます。文字列を含むフィールドを指定することもできます。

mask

文字

比較に使用する mask 文字列です。mask 文字列は一重引用符 (') で囲みます。

mask 内には、文字種のみを表す汎用文字を含めることができます。文字列内の文字の 1 つがこれらの文字列の 1 つと比較され、文字種が同一である場合、文字は一致したと見なされます。汎用文字は次のとおりです。

A - A から Z (大文字または小文字)

9 - 0 から 9 の任意の数字

x - A から Z の文字または 0 から 9 の数字

§ - 任意の文字

mask 内のその他の文字は、その文字自体を表します。たとえば、mask 内の 3 つ目の文字が「B」の場合、文字列内の 3 つ目の文字が一致するためには「B」である必要があります。

outfield

整数

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 フィールドのフォーマットを変換

CHKFMT 関数は、「11」で始まる 9 バイトの EMP_ID を検証し、結果を CHK_ID に格納します。

```
COMPUTE CHK_ID/I3 = CHKFMT(9, EMP_ID, '119999999', 'I3');
```

CTRAN - 文字を他の文字に変換

手順/構文

文字を他の文字に変換

CTRAN 関数は、文字列内の文字を 10 進数に基づいて他の文字に変換します。この関数は、利用できない文字、入力が難しい文字、またはキーボードにない文字を置換文字列によって入力するときに役立ちます。

CTRAN を使用するには、対応する内部マシンの 10 進表現の知識が必要です。

Unicode 構成の場合、この関数は次の範囲の値を使用します。

- 1 バイト文字 - 0 (ゼロ) から 255
- 2 バイト文字 - 256 から 65535
- 3 バイト文字 - 65536 から 16777215
- 4 バイト文字 - 16777216 から 4294967295 (主として EBCDIC 用)

構文

文字を他の文字に変換

```
CTRAN(charlen, string, decimal, decvalue, 'outfield')
```

説明

charlen

整数

文字列のバイト数です。長さを含むフィールドを指定することもできます。

string

文字

変換する文字列です。文字列は一重引用符 (') で囲みます。文字列を含むフィールドを指定することもできます。

decimal

整数

変換する文字の ASCII または EBCDIC の 10 進コードです。

decvalue

整数

decimal の代替文字として使用する文字の ASCII または EBCDIC の 10 進コードです。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例

空白をアンダースコア () に変換 (EBCDIC プラットフォーム)

CTRAN 関数は、ADDRESS_LN3 (EBCDIC 10 進数は 64) の空白をアンダースコア (EBCDIC 10 進数は 109) に変換し、ALT_ADDR に格納します。

```
COMPUTE ALT_ADDR/A20 = CTRAN(20, ADDRESS_LN3, 64, 109, 'A20');
```

CTRFLD - 文字列を中央揃え

手順/構文

文字列を中央揃え

CTRFLD 関数は、文字列をフィールド内で中央揃えにします。先頭の空白の数は末尾の空白の数と同一またはそれよりも 1 つ少ない数になります。

フィールドの内容または埋め込みフィールドのみで構成される見出しを中央揃えにするとときに役立ちます。HEADING CENTER は、各フィールド値を末尾の空白を含めて中央揃えにします。末尾の空白を含めずに中央揃えにするには、CTRFLD を使用します。

制限事項：スタイルシートを使用するレポートで CTRFLD を使用する場合、対象項目を中央揃え要素としてスタイル設定しない限り、CTRFLD は無効になります。また、デフォルトフォントがプロポーショナルであるプラットフォームで CTRFLD 関数を使用する場合、固定フォントを使用するか、リクエストの実行前に SET STYLE=OFF を発行します。

構文

文字列を中央揃え

```
CTRFLD(string, length, 'outfield')
```

説明

string

文字

文字列です。文字列は一重引用符 (') で囲みます。文字列を含むフィールドを指定することもできます。

length

整数

文字列および *outfield* のバイト数です。長さを含むフィールドを指定することもできます。この引数には正の数を指定します。*length* に負の数を指定すると、予期しない結果の原因になります。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 フィールドを中央揃え

CTRLFD 関数は、LAST_NAME を中央揃えにし、結果を CENTER_NAME に格納します。

```
COMPUTE CENTER_NAME/A12 = CTRLFD(LAST_NAME, 12, 'A12');
```

EDIT - 文字を抽出または追加

手順/構文

文字列を抽出または追加

EDIT 関数は、文字列から文字を抽出、または文字列に文字を追加します。文字列の任意の部分から文字列を抽出することができます。また、文字列から抽出した文字列を他の文字列に挿入することもできます。たとえば、文字列の先頭の 2 バイトと末尾の 2 バイトを抽出し、単一文字列を生成することができます。

EDIT 関数は、mask 内の文字とソースフィールド内の文字を比較します。mask 内で「9」が見つかると、EDIT はソース文字列の対応する文字を新しいフィールドにコピーします。mask 内にドル記号 (\$) が見つかると、EDIT はソースフィールド内の対応する文字を無視します。mask 内でその他の文字が見つかると、EDIT は新しいフィールドの対応する位置に、その文字をコピーします。EDIT 関数では、結果は文字に限定され、サイズは mask 引数の値が決定するため、outfield 引数を指定する必要はありません。

EDIT 関数は、フィールドのフォーマットを変換することもできます。EDIT 関数によるフィールドの変換についての詳細は、339 ページの「[EDIT-フィールドのフォーマットを変換](#)」を参照してください。

構文 文字列を抽出または追加

```
EDIT(fieldname, 'mask');
```

説明

fieldname

文字

ソースフィールドです。

文字列の抽出元の文字列です。mask と同一またはそれよりも長い文字列を指定します。

mask

文字

文字列です。文字列は一重引用符 (') で囲みます。mask の長さ (9 と \$ を除く) により出力フィールドの長さが決定されます。

例 フィールドから文字を抽出/フィールドに文字を追加

EDIT 関数は FIRST_NAME フィールドの 1 文字目を抽出し、FIRST_INIT に格納します。また、EMP_ID フィールドにハイフン (-) を追加し、結果を EMPIDEDIT に格納します。

```
COMPUTE FIRST_INIT/A1 = EDIT(FIRST_NAME, '9$$$$$$$$$'); AND  
COMPUTE EMPIDEDIT/A11 = EDIT(EMP_ID, '999-99-9999');
```

GETTOK - 文字列 (トークン) を抽出

手順/構文

文字列 (トークン) を抽出

GETTOK 関数は、ソース文字列を解析し、「トークン」と呼ばれる文字列を検索します。「区切り文字」と呼ばれる文字列内の特定の文字により、文字列をトークンに分割します。GETTOK 関数は、token_number により指定されたトークンを返します。文字列内の先頭と末尾の空白は無視されます。

たとえば、文字列の 4 つ目の単語を抽出することを想定します。区切り文字に空白を使用します。token_number には 4 を指定します。GETTOK は、この区切り文字により文を単語に分割し、4 つ目の単語を抽出します。文字列が区切り文字で分割されていない場合は、PARAG を使用します。

構文 文字列 (トークン) を抽出

```
GETTOK(infield, inlen, token_number, 'delim', outlen, 'outfield')
```

説明

infield

文字

元の文字列を含むフィールドです。

inlen

整数

ソース文字列の長さをバイト数で指定します。この引数が 0 (ゼロ) 以下の場合、空白が返されます。

token_number

整数

抽出するトークン数です。この引数が正の数の場合、トークンは左から右へ数えられます。この引数が負の数の場合、トークンは右から左へ数えられます。たとえば、-2 の場合、右から 2 つ目のトークンが抽出されます。この引数が 0 (ゼロ) の場合、空白が返されます。先頭と末尾の空白のトークンは無視されます。

delim

文字

一重引用符 (') で囲まれた元の文字列の区切り文字です。複数の文字を指定した場合、先頭の文字のみが使用されます。

outlen

整数

抽出するトークンのサイズです。この引数が 0 (ゼロ) 以下の場合、空白が返されます。この引数よりも長い場合、トークンは切り捨てられます。短い場合には、トークンの末尾に空白が追加されます。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。抽出されたトークンに、区切り文字は含まれません。

例**フィールドからトークンを抽出**

GETTOK 関数は、ADDRESS_LN3 から最後のトークンを抽出し、LAST_TOKEN に格納します。

```
COMPUTE LAST_TOKEN/A10 = GETTOK(ADDRESS_LN3, 20, -1, ' ', 10, 'A10');
```

LCWORD - 文字列の先頭文字を大文字に変換**手順/構文**

文字列の先頭文字を大文字に変換

LCWORD 関数は、文字列内の先頭文字を大文字、それ以外を小文字に変換します。単語の先頭文字、および一重引用符 (') と二重引用符 (") の後の先頭文字を除き、すべてのアルファベットを小文字に変換します。たとえば、「O'CONNOR」は「O'Connor」に変換され、「JACK'S」は「Jack'S」に変換されます。

LCWORD 関数は、文字列内の数字を大文字として処理するため、その後続く文字は小文字に変換されます。その結果、先頭文字が大文字で残りの文字が小文字の文字列が生成されます。

構文 **文字列の先頭文字を大文字に変換**

```
LCWORD(length, string, 'outfield')
```

説明

length

整数

変換される文字列またはフィールドの長さです。長さを含むフィールドを指定することもできます。

string

文字

変換される文字列です。文字列は一重引用符 (') で囲みます。文字列を含むフィールドを指定することもできます。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。このフィールドの長さは、*length* フィールドの値以上である必要があります。

例 **文字列の先頭文字以外を小文字に変換**

LCWORD 関数は、LAST_NAME フィールド内の先頭文字を大文字、それ以外を小文字に変換し、MIXED_CASE に格納します。

```
COMPUTE MIXED_CASE/A15 = LCWORD(15, LAST_NAME, 'A15');
```

LCWORD2 - 文字列の先頭文字以外を小文字に変換

手順/構文

文字列の先頭文字以外を小文字に変換

LCWORD2 関数は、文字列の先頭の文字以外をすべて小文字に変換します。LCWORD2 では、単一の一重引用符 (') が検出された場合、その次の文字は、小文字に変換されます。たとえば、「SMITH」は「Smith」に、「JACK'S」は「Jack's」に、それぞれ変換されます。

構文 **文字列の先頭文字以外を小文字に変換**

```
LCWORD2(length, string, 'outfield')
```

説明

length

整数

変換される文字列またはフィールドの長さです。長さを含むフィールドを指定することもできます。

string

文字

変換する文字列です。文字列を含む一時項目を指定することもできます。

outfield

文字

結果を格納するフィールド名、または出力フォーマットです。フォーマットは一重引用符 (') で囲みます。このフィールドの長さは、*length* フィールドの値以上である必要があります。

例 **文字列の先頭文字以外を小文字に変換**

Developer Workbench のレポート作成ツールまたはシノニムエディタで、一時項目 (COMPUTE) あるいは一時項目 (DEFINE) のダイアログボックスを使用して、変換を実行します。これにより、LCWORD2 関数によって、文字列「O'CONNOR」は先頭大文字に変換されます。

```
MYVAL1/A10='O'CONNOR'S';  
LC2/A10 = LCWORD2(10, MYVAL1, 'A10');
```

これらのフィールドのレポート出力は、次のとおりです。

```
MYVAL1      LC2
-----    ---
O'CONNOR'S  O'Connor's
```

LCWORD3 - 文字列の先頭文字以外を小文字に変換

手順/構文

LCWORD3 - 文字列を先頭大文字に変換

LCWORD3 関数は、文字列内の文字を先頭大文字に変換します。この関数で、各文字列の先頭文字が大文字に、その他すべての文字が小文字に変換されます。また、一重引用符 (') の後に空白が続く場合および入力文字列の最後の文字が一重引用符の場合を除いて、一重引用符に続く文字が大文字に変換されます。

たとえば、「SMITH」は「Smith」に、「JACK'S」は「Jack's」に、それぞれ変換されます。

構文

LCWORD3 - 文字列を先頭大文字に変換

`LCWORD3(length, string, output)`

説明

length

整数

変換される文字列またはフィールドの長さです。長さを含むフィールドを指定することもできます。

string

文字

変換する文字列、または文字列を含むフィールドです。

output

文字

結果を格納するフィールド名、または出力フォーマットです。フォーマットは一重引用符 (') で囲みます。このフィールドの長さは、*length* フィールドの値以上である必要があります。

例 文字列の先頭文字以外を小文字に変換 LCWORD3 の使用

Developer Workbench のレポート作成ツールまたはシノニムエディタで、一時項目 (COMPUTE) あるいは一時項目 (DEFINE) のダイアログボックスを使用して、変換を実行します。これにより、LCWORD3 関数によって、文字列「O'CONNOR'S」および「o'connor's」は、先頭大文字に変換されます。

```
MYVAL1/A10='O'CONNOR'S';
MYVAL2/A10='o'connor's';
LC1/A10 = LCWORD3(10, MYVAL1, 'A10');
LC2/A10 = LCWORD3(10, MYVAL2, 'A10');
```

出力結果では、最初の一重引用符 (') に続く文字「C」は大文字で出力されています。これは、この文字の後の文字が空白でもなく、入力文字列の最後の文字でもないためです。2つ目の一重引用符 (') に続く文字「s」は小文字で出力されています。これは、この文字が入力文字列の最後の文字であるためです。

MYVAL1	LC1	MYVAL2	LC2
-----	---	-----	---
O'CONNOR'S	O'Connor's	o'connor's	O'Connor's

LJUST - 文字列を左揃え

手順/構文

文字列を左揃え

LJUST 関数は、文字列を左揃えにします。左側の空白はすべて右側に移動します。

スタイルシート (SET STYLE=ON) を使用するレポートでは、項目を中央揃えにしない限り、LJUST 関数の視覚効果は有効になりません。

構文 文字列を左揃え

```
LJUST(length, string, 'outfield')
```

説明

length

整数

string および *outfield* の長さです。長さを含むフィールドを指定することもできます。

string

文字

左揃えされる文字列です。文字列を含むフィールドを指定することもできます。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 フィールドを左揃え

LJUST 関数は XNAME フィールドを左揃えに変更し、結果を YNAME に格納します。

```
COMPUTE YNAME/A25 = LJUST(15, XNAME, 'A25');
```

LOCASE - テキストを小文字に変換

手順/構文

テキストを小文字に変換

LCASE 関数は、テキストを小文字に変換します。

構文 テキストを小文字に変換

```
LOCASE(length, string, 'outfield')
```

説明

length

整数

string および *outfield* の長さです。長さを含むフィールドを指定することもできます。長さは 1 以上で、*string* と *outfield* に同一の値が指定されている必要があります。それ以外の値が指定されている場合はエラーが発生します。

string

文字

変換される文字列です。文字列は一重引用符 (') で囲みます。文字列を含むフィールドを指定することもできます。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。フィールド名には *string* と同一の値を指定することもできます。

例 フィールドを小文字に変換

LOCASE 関数は、LAST_NAME フィールドを小文字に変換し、結果を LOWER_NAME に格納します。

```
COMPUTE LOWER_NAME/A15 = LOCASE(15, LAST_NAME, 'A15');
```

OVERLAY - 文字列を上書き

手順/構文

文字列を上書き

OVERLAY 関数は、文字列を他の文字列で上書きします。

構文 文字列を上書き

```
OVERLAY(string1, stringlen, string2, sublen, position, 'outfield')
```

説明

string1

文字

元の文字列です。

stringlen

整数

string1 および *outfield* の長さです。長さを含むフィールドを指定することもできます。引数に 0 (ゼロ) 以下を指定すると、予期しない結果が発生します。

string2

文字

string1 を上書きする文字列です。

sublen

整数

string2 の長さです。長さを含むフィールドを指定することもできます。この引数が 0 (ゼロ) 以下の場合、空白が返されます。

position

整数

`source_string` 内で上書きを開始する位置です。この引数が 0 (ゼロ) 以下の場合、空白が返されます。この引数が `stringlen` よりも大きい場合、この関数は元の文字列を返します。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。上書き文字列が出力フィールドよりも長い場合、文字列はフィールドの長さに切り捨てられます。

例 文字列を上書き

OVRLAY 関数は、EMP_ID の末尾の 3 バイトを CURR_JOBCODE の値で置換して新しいセキュリティ ID を作成し、結果を NEW_ID に格納します。

```
COMPUTE NEW_ID/A9 = OVRLAY(EMP_ID, 9, CURR_JOBCODE, 3, 7, 'A9');
```

PARAG - テキストを行に分割

手順/構文

テキストを行に分割

PARAG 関数は、区切り文字を使用して 1 行の文字列を複数の文字列に分割します。行の開始位置から特定のバイト数をスキャンし、その文字列グループ末尾の空白を区切り文字で置換します。その後、区切り文字から開始して、行の次の文字列グループをスキャンし、このグループ末尾の空白を 2 つ目の区切り文字と置換することにより、2 つ目のトークンを作成します。行の末尾に到達するまで、この処理を繰り返します。

区切り文字で分割された各文字列グループは個別の行になります。その後、GETTOK 関数がこれらの行を異なるフィールドに指定します。PARAG 関数は、スキャンしたグループ内に空白が見つからない場合、グループの先頭文字を区切り文字で置換します。このため、すべての文字列がスキャンされたバイト数以下であることが重要です。

入力された行の長さがほぼ同じである場合、均等に表示されるように分割行の長さを指定することができます。たとえば、文字列が 120 バイトの場合、各行を 60 バイト (2 行) または 40 バイト (3 行) に分割することができます。この方法を使用すると、テキスト行を段落形式で表示することが可能です。

ただし、行を等しいサイズに分割する場合、意図する数よりも多くの行が作成される可能性があります。たとえば、120 バイトのテキスト行を最大 60 バイトの 2 行に分割しようとして、1 つ目の行が 50 バイト、2 つ目の行が 55 バイトに分割されたことを想定します。分割後の 3 つ目の行は 15 バイトになります。これを修正するには、3 つ目の行の先頭に空白を挿入 (弱連結を使用) し、この行を 2 つ目の行末に追加 (強連結を使用) します。

構文 **テキストを行に分割**

```
PARAG(length, string, 'delim', subsize, 'outfield')
```

説明

length

整数

string および *outfield* の長さです。長さを含むフィールドを指定することもできます。

string

文字

文字列です。文字列は一重引用符 (') で囲みます。文字列を含むフィールドを指定することもできます。

delim

文字

一重引用符 (') で囲まれた区切り文字です。テキストで使用されていない文字を選択します。

subsize

整数

各行の長さの最大値です。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 **テキストを行に分割**

PARAG 関数は、カンマ (,) を区切り文字として使用し、ADDRESS_LN2 を 10 バイト以下の行に分割します。その結果を PARA_ADDR に格納します。

```
COMPUTE PARA_ADDR/A20 = PARAG(20, ADDRESS_LN2, ',', 10, 'A20');
```

POSIT - 文字列の開始位置を検索

手順/構文

文字列の開始位置を検索

POSIT 関数は、文字列の中から特定の文字列の開始位置を見つけます。たとえば、文字列 PRODUCTION 内の文字列 DUCT の開始位置は 4 です。指定した文字列が元の文字列内に存在しない場合、この関数は値 0 (ゼロ) を返します。

構文 **文字列の開始位置を検索**

```
POSIT(parent, inlength, substring, sublength, 'outfield')
```

説明

parent

文字

文字列です。文字列は一重引用符 (') で囲みます。文字列を含むフィールドを指定することもできます。

inlength

整数

元の文字列のバイト数です。長さを含むフィールドを指定することもできます。この引数が 0 (ゼロ) 以下の場合、0 (ゼロ) が返されます。

substring

文字

位置を検索する文字列です。一重引用符 (') で囲まれた文字列またはその文字列を含むフィールドを指定することもできます。

sublength

整数

substring の長さです。引数が 0 (ゼロ) 以下の場合、または *inlength* よりも大きいときは、0 (ゼロ) が返されます。

outfield

整数

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 **文字列の開始位置を検索**

POSIT 関数は、LAST_NAME の 1 つ目の大文字「I」の位置を決定し、結果を I_IN_NAME に格納します。

```
COMPUTE I_IN_NAME/I2 = POSIT(LAST_NAME, 15, 'I', 1, 'I2');
```

PTOA - パック 10 進数を文字に変換

PTOA 関数は、数値を数値フォーマットから文字フォーマットに変換します。数値の小数点の位置を保持し、先頭に空白を挿入することにより、数値を右揃えします。PTOA によって変換された数値には、編集オプションを追加することができます。

PTOA を使用して 10 進数を含む数値を文字列に変換するときは、数値の整数部分と小数点以下を格納するために十分な大きさの文字フォーマットを指定する必要があります。たとえば、P12.2C フォーマットは A14 に変換されます。出力フォーマットの長さが不十分な場合、右端の文字が切り捨てられます。

参照 **パック 10 進数を文字に変換**

```
PTOA(number, '(format)', output)
```

説明

number

数値 P (パック 10 進数)、F または D (単精度または倍精度浮動小数点数) 変換される数値です。

format

文字

数値フォーマットを括弧で囲んで指定します。

output

文字

この引数の長さは数値の長さよりも大きくする必要があります。編集オプションおよび負の符号が追加される可能性も考慮します。たとえば、パック 10 進数から文字フォーマットへの変換は、PGROSS をパック 10 進数から文字フォーマットに変換します。

```
PTOA(PGROSS, '(P12.2)', 'A14');
```

REVERSE - 文字列の順序を入れ替え

手順/構文

文字列の順序を入れ替え

REVERSE 関数は、文字列内の文字の順序を逆にします。

構文

文字列の順序を入れ替え

```
REVERSE(length, string, 'outfield')
```

説明

length

整数

string および *outfield* の長さです。長さを含むフィールドを指定することもできます。

string

文字

文字列です。文字列は一重引用符 (') で囲みます。文字列を含むフィールドを指定することもできます。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例

文字列の順序を入れ替え

REVERSE 関数は、PRODCAT 内の文字列の順序を入れ替え、結果を REVERSE_NAME に格納します。

```
COMPUTE REVERSE_NAME/A15 = REVERSE(15, PRODCAT, 'A15');
```

RJUST - 文字列を右揃え

手順/構文

文字列を右揃え

RJUST 関数は、文字列を右揃えにします。末尾の空白は、すべて先頭の空白になります。数値を含む文字フィールドを表示するときに役立ちます。

スタイルシート (SET STYLE=ON) を使用するレポートでは、項目を中央揃えにしない限り、RJUST 関数の視覚効果は有効になりません。また、スタイルシートがデフォルト設定でオンに設定されているプラットフォームで RJUST 関数を使用する場合は、リクエストの実行前に SET STYLE=OFF を発行します。

構文

文字列を右揃え

```
RJUST(length, string, 'outfield')
```

説明

length

整数

string および *outfield* の長さです。長さを含むフィールドを指定することもできます。文字揃えの問題を回避するため、必ず同一の長さを指定します。

string

文字

文字列です。一重引用符 (') で囲まれた文字列を含むフィールドを指定することもできます。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例

フィールドを右揃え

RJUST 関数は、LAST_NAME フィールドを右揃えに変更し、結果を RIGHT_NAME に格納します。

```
COMPUTE RIGHT_NAME/A15 = RJUST(15, LAST_NAME, 'A15');
```

SOUNDEX - 文字列を音声的に比較

手順/構文

文字列を音声的に比較

SOUNDEX 関数は、綴りに関わらず、文字列を音声的に解析します。文字列を 4 バイトのコードに変換します。先頭の文字は、常に文字列内の先頭文字です。残りの 3 バイトは、文字列内で音声的に有意な次の 3 バイトを表します。

音声検索を実行するには、次の手順を実行します。

1. SOUNDEX 関数を使用して、検索するフィールドのデータ値を音声コードに変換します。
2. SOUNDEX 関数を使用して、最も妥当と想定されるターゲット文字列を音声コードに変換します。ターゲット文字列の綴りは正確である必要はありませんが、先頭文字は一致する必要があります。
3. WHERE 条件または IF 条件を使用して、手順 1 で作成した一時項目と手順 2 で作成した一時項目を比較します。

構文

文字列を音声的に比較

```
SOUNDEX(inlength, string, 'outfield')
```

説明

inlength

文字 (2 バイト)

string のバイト数です。長さを含むフィールドを指定することもできます。一重引用符 (') で囲まれた数値、または数値を含むフィールドを指定することもできます。01 から 99 の 2 桁の数値 (例、'01') を指定する必要があります。99 よりも大きい数値を指定すると、出力結果としてアスタリスク (*) が返されます。

string

文字

文字列です。文字列は一重引用符 (') で囲みます。文字列を含むフィールドを指定することもできます。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例

文字列を音声的に比較

以下のリクエストは次の 3 つのフィールドを作成します。

- PHON_NAME には、従業員の LAST_NAME の音声コードが含まれます。
- PHON_COY には、想定した音声コード「MICOY」が含まれます。

- PHON_MATCH 関数は、想定した値を音声コードと比較します。

```
COMPUTE PHON_NAME/A4 = SOUNDEX('15', LAST_NAME, 'A4'); AND
COMPUTE PHON_COY/A4 WITH LAST_NAME = SOUNDEX('15', 'MICOY', 'A4'); AND
COMPUTE PHON_MATCH/A3 = IF PHON_NAME IS PHON_COY THEN 'YES' ELSE 'NO';
```

SPELLNM - ドルとセントの通貨表記を文字表記に書き替え

手順/構文

ドルとセントの通貨表記を文字表記に書き替え

SPELLNM 関数は、2 桁の英数文字列または小数点以下 2 桁の数値をドルとセントの文字表記に変換します。たとえば、「32.50」という値は THIRTY TWO DOLLARS AND FIFTY CENTS に変換されます。

構文

ドルとセントの通貨表記を文字表記に書き替え

```
SPELLNM(outlength, number, 'outfield')
```

説明

outlength

整数

outfield の長さです。長さを含むフィールドを指定することもできます。

数値の最大値が分かっている場合、下表に従って *outlength* の値を決定します。

最大値 (より小さい)	<i>outlength</i> の値
\$10	37
\$100	45
\$1,000	59
\$10,000	74
\$100,000	82
\$1,000,000	96

number

文字または数値 (9.2)

文字に変換する数字です。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例

ドルとセントの通貨表記を文字表記に書き替え

SPELLNM 関数は CURR_SAL の値の綴りを書き出し、結果を AMT_IN_WORDS に格納します。

```
COMPUTE AMT_IN_WORDS/A82 = SPELLNM(82, CURR_SAL, 'A82');
```

SUBSTR - 文字列を抽出

手順/構文

文字列を抽出

SUBSTR 関数は、文字列内の開始位置および長さに基づいて、文字列を抽出します。SUBSTR 関数は、他のフィールドの値に基づいて文字列の位置を変更することができます。

構文

文字列を抽出

```
SUBSTR(inlength, parent, start, end, sublength, 'outfield')
```

説明

inlength

整数

元の文字列の長さです。長さを含むフィールドを指定することもできます。

parent

文字

一重引用符 (') で囲まれた文字列です。文字列を含むフィールドを指定することもできます。

start

整数

`parent` 内の抽出文字列の開始位置です。この引数が 0 (ゼロ) 以下の場合、空白が返されます。

end

整数

抽出する文字列の終了位置です。この引数が `start` より小さい、または `inlength` より大きい場合、この関数は空白を返します。

sublength

整数

文字列の長さです。この値は通常、`end - start + 1` (`end` と `start` の値の差に 1 を加えたもの) です。`sublength` が `end - start + 1` よりも長い場合、抽出文字列の末尾に空白が追加されます。短い場合は、抽出文字列は切り捨てられます。この値は `outfield` で宣言した長さと同一にします。`sublength` の長さ分の文字のみが処理されます。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例

文字列を抽出

`POSIT` 関数は、`LAST_NAME` の 1 つ目の「I」の位置を決定し、結果を `I_IN_NAME` に格納します。その後、`SUBSTR` 関数は、`LAST_NAME` から「I」で開始する 3 バイトを抽出し、結果を `I_SUBSTR` に格納します。

```
COMPUTE I_IN_NAME/I2 = POSIT(LAST_NAME, 15, 'I', 1, 'I2'); AND
COMPUTE I_SUBSTR/A3 = SUBSTR(15, LAST_NAME, I_IN_NAME, I_IN_NAME+2, 3,
'A3');
```

UPCASE - テキストを大文字に変換

手順/構文

テキストを大文字に変換

UPCASE 関数は、文字列を大文字に変換します。大文字と小文字が混在する値、および大文字のみの値が混在するフィールドをソートする際に役立ちます。大文字と小文字が混在するフィールドをソートすると、誤った結果が発生します。これは、EBCDIC によるソートは常に大文字の前に小文字を配置し、ASCII によるソートは常に小文字の前に大文字を配置するためです。正しい結果を取得するには、値すべてを大文字にした新しいフィールドを定義し、そのフィールドをソートします。

構文

テキストを大文字に変換

```
UPCASE(length, input, 'outfield')
```

説明

length

整数

input および *outfield* のバイト数です。

input

文字

文字列です。文字列は一重引用符 (') で囲みます。文字列を含むフィールドを指定することもできます。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例

大文字と小文字が混在するフィールドを大文字に変換

UPCASE 関数は、LAST_NAME フィールドを小文字に変換し、結果を LOWER_NAME に格納します。

```
COMPUTE LAST_NAME_UPPER/A15 = UPCASE(15, LAST_NAME_MIXED, 'A15') ;
```

データソースおよびデコード関数

トピックス

DECODE - 値を置き換え

LAST - 前の値を抽出

WHERE 条件による値へのアクセス制限

データソースおよびデコード関数は、データソースの値を抽出し、入力フィールドの値に基づいて値を割り当てます。

DECODE - 値を置き換え

手順/構文

関数に値を指定

ファイルから値を読み込み

参照

ファイルからの値の読み込みについてのガイドライン

DECODE 関数は、コード化された入力フィールドの値に基づいて値を割り当てます。コード化されたフィールドの値に、より意味のある値を指定する場合に役立ちます。たとえば、フィールド GENDER は、女性従業員に F コード、男性従業員用に M コードを割り当てることにより、効率的にデータを格納します (女性従業員を表す 6 バイト分の文字 female の代わりに 1 バイト)。DECODE は、これらの値を置き換え、レポート上に正しく表示します。

DECODE 関数を使用するには、値を関数内に直接指定、または個別のファイルから読み取ります。

構文

関数に値を指定

```
DECODE fieldname(code1 result1 code2 result2...[ELSE default]);
```

説明

fieldname

文字または数値

入力フィールド名です。

code

文字または数値

DECODE 関数が検索するコード化された値です。この値に空白、カンマ (,)、または他の特殊文字が埋め込まれている場合、一重引用符 (') で囲みます。DECODE が指定した値を検出すると、対応する結果が返されます。

result

文字または数値

コードに割り当てられた値です。この値に空白またはカンマ (,) が埋め込まれている場合、あるいは負の数が含まれるときは、一重引用符 (') で囲みます。

default

文字または数値

コードが見つからない場合に割り当てられる値です。デフォルト値を省略する場合、DECODE は不一致コードに空白または 0 (ゼロ) を割り当てます。

DECODE 関数のコードと結果の組み合わせには、通常 40 行まで、ELSE キーワードを含める場合は、39 行まで使用することができます。結果からコードを区別するには、カンマ (,) または空白を使用します。

例

関数に値を指定

まず、EDIT 関数が CURR_JOBCODE フィールドの 1 文字目を抽出します。次に、DECODE 関数は、抽出された値に基づいて ADMINISTRATIVE または DATA PROCESSING を返します。

```
COMPUTE DEPX_CODE/A1 = EDIT(CURR_JOBCODE, '9$$'); AND  
COMPUTE JOB_CATEGORY/A15 = DECODE DEPX_CODE(A 'ADMINISTRATIVE'  
B 'DATA PROCESSING') ;
```

構文

ファイルから値を読み込み

```
DECODE fieldname(ddname [ELSE default]);
```

説明

fieldname

文字または数値

入力フィールド名です。

ddname

文字

置き換えた値を含む物理ファイルを指定する論理名または短縮名です。

default

文字または数値

コードが見つからない場合に割り当てられる値です。デフォルト値を省略する場合、DECODE はブランクまたは 0 (ゼロ) を不一致コードに割り当てます。

参照

ファイルからの値の読み込みについてのガイドライン

- ファイル内の各レコードには、カンマ (,) またはブランクで区切られた要素の組み合わせが含まれていることが想定されます。
- ファイル内の各レコードが 1 つの要素のみで構成されている場合、この要素はコードとして解釈され、結果としてブランクまたは 0 (ゼロ) が生成されます。

これにより、ファイルに次の選別条件で参照するリテラルを含めることができます。

```
IF field IS (filename)
```

また、このファイルを計算式で指定する IF 条件のリテラルファイルとして使用することもできます。以下はその例です。

```
TAKE = DECODE SELECT (filename ELSE 1);
VALUE = IF TAKE IS 0 THEN... ELSE...;
```

TAKE は、リテラルファイルに SELECT 値が見つかった場合には 0 (ゼロ)、それ以外の場合には 1 です。式の計算と同様に VALUE の計算が実行されます。

```
IF SELECT (filename) THEN... ELSE...;
```

- ファイルの最大容量は、32767 バイトです。
- 先頭と末尾のブランクは無視されます。
- 各レコードの残りの部分は無視されますが、これらはコメントや他のデータに使用可能です。この規則は、ファイル名が HOLD である場合以外に適用されます。ファイル名が HOLD の場合、ファイルはフィールドに内部フォーマットで書き込む HOLD コマンドにより作成されたと見なされ、DECODE の組はそれに基づいて解釈されます。この場合、レコードの残りの部分は無視されます。

例 ファイルから値を読み込み

DECODE 関数は、EMP_ID が HOLD ファイルに存在する従業員に値 0 (ゼロ) を、存在しない従業員に値 1 を、それぞれ割り当てます。

```
COMPUTE NOT_IN_LIST/I1 = DECODE EMP_ID(HOLD ELSE 1);
```

LAST - 前の値を抽出

手順/構文

前の値を抽出

LAST 関数は、フィールドの前の値を抽出します。

LAST の影響は、DEFINE コマンドと COMPUTE コマンドのどちらに使用するかによって異なります。

- ❑ DEFINE コマンドでは、LAST 値は、ソートを実行する前に、データソースから抽出した以前のレコードに適用されます。
- ❑ COMPUTE コマンドでは、LAST 値は、内部マトリックスの前の行のレコードに適用されます。

構文

前の値を抽出

```
LAST fieldname
```

説明

fieldname

文字または数値

フィールド名です。

例 前の値を抽出

LAST は、DEPARTMENT フィールドの前の値を抽出し、部署ごとの給与合計を再計算するかどうかを決定します。前の値が現在の値と同一である場合、CURR_SAL が RUN_TOT に追加され、各部署の給与の合計が生成されます。

```
COMPUTE RUN_TOT/D12.2M = IF DEPARTMENT EQ LAST DEPARTMENT THEN  
    (RUN_TOT + CURR_SAL) ELSE CURR_SAL ;
```


WHERE 条件による値へのアクセス制限

手順/構文

WHERE 条件による値へのアクセス制限

DBA セキュリティを使用して、データソース内の特定の値へのアクセスを制限することができます。RESTRICT=VALUE 属性は、IF 句でサポートされている条件で使用することができます。RESTRICT=VALUE_WHERE 属性は、WHERE 句でサポートされているすべての条件で使用することができます (例、フィールド間での比較、関数の使用)。WHERE 式は、可能な場合に構成済みのアダプタに渡されます。

構文

WHERE 条件による値へのアクセス制限

```
USER=password, ACCESS={R|RW|W},
  RESTRICT=VALUE_WHERE, NAME=name,
  VALUE=expression; , $
```

説明

password

最大長が 64 バイトのユーザパスワードです。

ACCESS={R|RW|W}

ユーザに許可するアクセス権限のタイプです (R=読み取り、RW=読み取り/書き込み、W=書き込み)。

RESTRICT=VALUE_WHERE

ユーザは VALUE 属性で指定された式を満たす値にのみアクセスすることができます。

name

制限の対象とするフィールドまたはセグメントの名前です。値のテストでのみ使用可能な NAME=SYSTEM は、下位のセグメントを含めてデータソース内のすべてのセグメントが制限の対象になります。制限の対象として複数のフィールドまたはセグメントを指定するには、1 名のユーザに対して RESTRICT 属性を複数回発行します。

VALUE=*expression*

ユーザにアクセス権限を与える値を記述する式を指定します。式には、TABLE リクエストの WHERE テストでサポートされるすべての要素を使用することができます。

日付時間関数

トピックス

AYM - 基準となる日付に月数を加えて、新たな年月を求める

AYMD - 基準となる日付に日数を加えて、新たな日付を求める

CHGDAT - 日付文字列の表示を変更

DA 関数 - レガシー日付を整数に変換

DATEADD - 日付と日付単位の和または差を計算

DATECVT - 日付フォーマットを変換

DATEDIF - 2 つの日付の差を計算

DATEMOV - 日付を有効な位置に移動

日付構成要素を整数として取得

DATETRAN - 日付を国際フォーマットに変換

日付時間値の精度

マスターファイルの DATEPATTERN

DMY、MDY、YMD - 2 つの日付の差を計算

DOWK および DOWKL - 曜日を検索

DT 関数 - 整数を日付に変換

FIYR - 会計年度の取得

FIQTR - 会計四半期の取得

FIYYQ - カレンダー日付を会計日付に変換

GREGDT - ユリウス暦から太陽暦フォーマットに変換

HADD - 日付時間値を増加

HCVRT - 日付時間値を文字フォーマットに変換

HDATE - 日付時間値の日付部分を日付フォーマットに変換

HDIFF - 2 つの日付時間値の差単位数を計算

HDTTM - 日付値を日付時間値に変換

HGETC - 現在の日付および時間を日付時間フィールドに格納

HHMMSS - 現在の時間を取得

HINPUT - 文字列を日付時間値に変換

HMIDNT - 日付時間値の時間部分を午前零時に設定

HNAME - 日付時間構成要素を文字フォーマットで取得

HPART - 日付時間構成要素を数値フォーマットで取得

HSETPT - 日付時間値に構成要素を挿入

HTIME - 日付時間値の時間部分を数値に変換

JULDAT - 太陽暦からユリウス暦フォーマットに変換

TIMETOTS - 時間をタイムスタンプに変換

TODAY - 現在の日付を取得

YM - 経過月数を計算

日付と時間の値を操作する日付および時間関数について説明します。

標準日付時間関数を使用する際、これらの関数の動作を変更する設定、使用可能なフォーマット、値の指定方法について理解する必要があります。

営業日とそれ以外の曜日を定義することで、日付時間関数の動作を変更することができます。これにより、営業日に対して日付関数を使用すると、営業日以外の日付は無視されません。

AYM - 基準となる日付に月数を加えて、新たな年月を求める

手順/構文

基準となる日付に日数を加えて、新たな日付を求める

AYM 関数は、年月フォーマットの日付と指定した月数の和または差を計算します。CHGDATE または EDIT 関数を使用することで、日付をこのフォーマットに変換することができます。

構文

基準となる日付に日数を加えて、新たな日付を求める

```
AYM(indate, months, 'outfield')
```

説明

indate

整数 (I4、I4YM、I6、I6YYM)

年月フォーマットの元の日付、日付を含むフィールド名、日付を返す式のいずれかを指定します。日付が有効でない場合、この関数は 0 (ゼロ) を返します。

months

整数

日付に加算、または日付から減算する月数です。月を減算するには、負の値を使用します。

outfield

整数 (I4YM、I6YYM)

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

ヒント：入力日付が整数年月日フォーマット (I6YMD または I8YYMD) の場合、日付を 100 で除算することにより、年月フォーマットに変換し、結果を整数に設定します。これにより日付の日の部分が削除されます。日は、小数部分で表されます。

例 日付に月を追加

COMPUTE コマンドは、HIRE_DATE の日付を年月日から年月フォーマットに変換し、結果を HIRE_MONTH に格納します。AYM は、HIRE_MONTH に 6 か月加算し、結果を AFTER6MONTHS に格納します。

```
COMPUTE HIRE_MONTH/I4YM = HIRE_DATE/100; AND
COMPUTE AFTER6MONTHS/I4YM = AYM(HIRE_MONTH, 6, 'I4YM');
```

AYMD - 基準となる日付に日数を加えて、新たな日付を求める

手順/構文

基準となる日付に日数を加えて、新たな日付を求める

AYMD 関数は、年月フォーマットの日付と指定した日数の和または差を計算します。CHGDAT または EDIT 関数を使用することで、日付をこのフォーマットに変換することができます。

日の加算または減算により、世紀が前後に変更される場合、出力年の世紀の桁は調整されます。

構文 基準となる日付に日数を加えて、新たな日付を求める

```
AYMD(indate, days, 'outfield')
```

説明

indate

整数 (I6、I6YMD、I8、I8YYMD)

日付が有効でない場合、この関数は 0 (ゼロ) を返します。

days

整数

indate に加算または *indate* から減算する日数です。日を減算するには、負の値を使用します。

outfield

整数 (I6、I6YMD、I8、I8YYMD)

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。*indate* がフィールドの場合、*outfield* とフォーマットが同一である必要があります。

例 日付に日数を追加

AYMD 関数は、HIRE_DATE フィールド内の各値に 35 日加算し、結果を AFTER35DAYS に格納します。

```
COMPUTE AFTER35DAYS/I6YMD = AYMD(HIRE_DATE, 35, 'I6YMD');
```

CHGDAT - 日付文字列の表示を変更

手順/構文

日付文字列の表示を変更

参照

日付の長さを変換

CHGDAT 関数使用上の注意

CHGDAT 関数は、日付を表す入力文字列の年、月、日の部分を再編成します。この関数により、長い日付入力文字列を短く、短い日付入力文字列を長くすることもできます。長い表現には、3つの日付構成要素である年、月、日がすべて含まれ、短い表現では、年、月、日の日付構成要素の1つまたは2つが省略されます。入力および出力日付文字列は、日付文字列内の構成要素(年、月、日)の順序、および年に2桁と4桁のどちらを使用するか(例、97 または 1997)を記述します。CHGDAT は、入力日付文字列を読み取り、同一の日付を異なる方法で表示する出力日付文字列を作成します。

注意：CHGDAT には、実際の日付ではなく入力文字列としての日付が必要です。CHGDAT を適用する前に、入力対象を日付文字列に変換しておきます。この変換には EDIT 関数や DATECVT 関数などが使用できます。

日付文字列内の日付構成要素の順序は、次の文字で構成される表示オプションで記述します。

文字	説明
D	日 (01 から 31)
M	月 (01 から 12)
Y[Y]	年。Y - 2 桁の年を示します (例、94)。YY - 4 桁の年を示します (例、1994)。

結果文字列に数字ではなく月名を表示するには、結果文字列の表示オプションに次の文字を追加します。

文字	説明
T	月を 3 バイトの略語で表示します。
X	月を完全な名前を表示します。

表示オプションには、表示する文字を 5 バイト以下で指定することができます。これらの表示オプション以外の文字列は、無視されます。

たとえば、表示オプション「DMYY」は、2 桁の日、2 桁の月、4 桁の年を指定します。

注意：表示オプションは、日付フォーマットとは異なります。

参照

日付の長さを変換

下表のように、日付を短い表現から長い表現 (例、年月から年月日) に変換すると、短い表現内に存在しない日付部分は、この関数により指定されます。

存在しない日付部分	関数により指定
日 (例、YM から YMD)	月の最終日
月 (例、Y から YM)	年の最終月 (12 月)
年 (例、MD から YMD)	99 年
2 桁の年から 4 桁の年への変換 (例、YMD から YYMD)	DATEFNS=ON の場合、世紀は DEFCENT および YRTHRESH で定義する 100 年で決定されます。 DATEFNS=OFF の場合、19xx 年指定されます。ここで、xx は年の末尾の 2 桁です。

構文 日付文字列の表示を変更

```
CHGDAT('in_display_options', 'out_display_options', date_string, 'outfield')
```

説明

in_display_options

文字 (A1 から A5)

date_string のレイアウトを記述する 5 バイト以内の一連の表示オプションです。これらのオプションは、文字フィールドに格納するか、リテラルとして指定します。リテラルは一重引用符で囲みます。

out_display_options

文字 (A1 から A5)

変換後の文字列のレイアウトを記述する 5 バイト以内の一連の表示オプションです。これらのオプションは、文字フィールドに格納するか、リテラルとして指定します。リテラルは一重引用符で囲みます。

date_string

文字 (A2 から A8)

入力日付文字列です。日付構成要素の順序は *in_display_options* で指定します。

元の日付が数値フォーマットの場合、日付文字列に変換する必要があります。*date_string* の日付表現が正しくない (日付が無効) 場合、関数はブランクを返します。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

out_display_options で指定する日付文字列を格納するために十分なバイト数を指定します。A17 を指定すると、最長の日付文字列を格納することができます。

参照 CHGDAT 関数使用上の注意

CHGDAT は、日付ではなく日付文字列を使用し、17 バイト以下の日付文字列を返すため、EDIT 関数、DATECVT 関数、またはそれ以外の手段により、日付文字列に変換、あるいは日付文字列から変換する必要があります。

例 日付表示を YMD から MDYYX に変換

EDIT 関数は、HIRE_DATE の表示を数値から文字フォーマットに変換します。CHGDAT 関数は、ALPHA_HIRE の各値の構成要素の表示を YMD から MDYYX に変換し、結果を A17 フォーマットで HIRE_MDY に格納します。出力値のオプション X により、完全な月名が表示されます。

```
COMPUTE ALPHA_HIRE/A17 = EDIT(HIRE_DATE); AND
COMPUTE HIRE_MDY/A17 = CHGDAT('YMD', 'MDYYX', ALPHA_HIRE, 'A17');
```

DA 関数 - レガシー日付を整数に変換

手順/構文

レガシー日付を整数に変換

DA 関数は、日付を 1899 年 12 月 31 日との日数の差に変換します。日付を日数に変換することにより、日付の和または差の計算を実行し、日付の間隔を計算することができます。

DA 関数は 6 つあります。各関数では、異なるフォーマットの日付が使用可能です。

構文 レガシー日付を整数に変換

```
function(indate, 'outfield')
```

説明

function

次のいずれかです。

DADMY - 日月年フォーマットの日付を変換します。

DADYM 日年月フォーマットの日付を変換します。

DAMDY - 月日年フォーマットの日付を変換します。

DAMYD - 月年日フォーマットの日付を変換します。

DAYDM - 年日月フォーマットの日付を変換します。

DAYMD - 年月日フォーマット日付を変換します。

indate

整数またはパック 10 進数

日付表示オプションを含む I または P フォーマットです。

変換される日付、または日付を含むフィールド名のいずれかを指定します。日付は、変換前に整数に切り捨てられます。

年を指定するには、最後の 2 桁のみを入力します。世紀構成要素は、関数により入力されます。日付が有効でない場合、この関数は 0 (ゼロ) を返します。

outfield

整数

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。返される日付のフォーマットは、関数により異なります。

例 **日付を変換し差を計算**

DAYMD 関数は、DAT_INC および HIRE_DATE フィールドの値を 1899 年 12 月 31 日との日数の差に変換します。その後、その最小値と他の値との差が計算されます。

```
COMPUTE DAYS_HIRED/I8 = DAYMD(DAT_INC, 'I8') - DAYMD(HIRE_DATE, 'I8');
```

DATEADD - 日付と日付単位の和または差を計算

手順/構文

日付単位数を日付に加算または日付から減算

DATEADD 関数は、日付フォーマットとの単位の和または差を計算します。単位は次のいずれかです。

- 年
- 月 月単位を使用する計算が無効な日付を作成する場合、DATEADD は、この日付を月の最終日に修正します。たとえば、10 月 31 日に 1 か月加算すると、11 月は 30 日間であるため、11 月 30 日になります。
- 日付
- 平日 平日単位を使用すると、DATEADD は土曜日と日曜日を計算しません。たとえば、金曜日に 1 日加算すると、結果は月曜日になります。
- 営業日 営業日単位を使用すると、DATEIF は BUSDAYS パラメータおよび祝日ファイルにより、営業日を決定します。これ以外は無視されます。月曜日が営業日でない場合、日曜日の次の営業日は火曜日になります。

日を基準としない日付 (例、YM または YQ) の和または差の計算は、DATEADD 関数を使用せずに、直接実行してください。

DATEADD 関数は、年月日を含む完全な日付形式でのみ有効です。

構文

日付単位数を日付に加算または日付から減算

```
DATEADD(date, 'unit', #units)
```

説明

date

日付

年月日を含む完全な日付形式です。

unit

文字

次のいずれかを一重引用符 (') で囲んで指定します。

Y - 構成要素「年」を示します。

M - 構成要素「月」を示します。

D - 構成要素「日」を示します。

WD - 構成要素「平日」を示します。

BD - 構成要素「営業日」を示します。

#units

整数

date の値との和または差が計算される日付単位数です。単位数と異なる場合、次に大きい整数に切り捨てられます。

例

日付に平日を追加

DATEADD は、平日 3 日を NEW_DATE に追加します。3 日追加すると HIRE_DATE_PLUS_THREE が週末になることがあるため、4 日以上追加する場合があります。

```
COMPUTE NEW_DATE/YYMD = HIRE_DATE; AND  
COMPUTE HIRE_DATE_PLUS_THREE/YYMD = DATEADD(NEW_DATE, 'WD', 3);
```

DATECVT - 日付フォーマットを変換

手順/構文

日付フォーマットを変換

DATECVT 関数は、中間の計算を実行せずに、アプリケーションの日付のフォーマットを変換します。無効なフォーマットを指定すると、DATECVT は 0 (ゼロ) またはブランクを返します。

DATECVT は、日付フォーマットの A8YYMD、A8MDYY、A8DMYY、I8YYMD、I8MDYY、I8DMYY、P8YYMD、P8MDYY、P8DMYY を使用する場合に最適化されます。現在、これらの日付を変換するには、最適化済みの SQL が DB2 エンジンに渡されるようになっています。

注意：この関数を呼び出す代わりに、単純な割り当てを使用することができます。

構文

日付フォーマットを変換

```
DATECVT(date, 'infmt', 'outfmt')
```

説明

date

日付

変換される日付です。無効な日付を指定すると、0 (ゼロ) が返されます。変換を実行する際、レガシー日付は、このフィールドに指定された DEFCENT および YRTHRESH パラメータの設定に従います。

infmt

文字

日付のフォーマットです。フォーマットは一重引用符 (') で囲みます。次のいずれかです。

- 標準、非レガシー、または日付フォーマット (例、YYMD、YQ、M、DMY、JUL)。
- レガシー日付フォーマット (例、I6YYMD、A8MDYY)。
- 非日付フォーマット (例、I8、A6)。YYMD フィールドの基準日 (1900/12/31) のオフセットとしての *infmt* 関数の非日付フォーマットです。

outfmt

文字

出力のフォーマットです。フォーマットは一重引用符 (') で囲みます。次のいずれかです。

- 標準、非レガシー、または日付フォーマット (例、YYMD、YQ、M、DMY、JUL)。
- レガシー日付フォーマット (例、I6YMD、A8MDYY)。
- 非日付フォーマット (例、I8、A6)。YYMD フィールドの基準日 (1900/12/31) のオフセットとしての *infmt* 関数の非日付フォーマットです。

例 日付フォーマットを YYMD から DMY に変換

DATECVT 関数は、19991231 を 311299 に変換し、結果を CONV_FIELD に格納します。

```
COMPUTE CONV_FIELD/DMY = DATECVT(19991231, 'I8YYMD', 'DMY');
```

または

```
COMPUTE CONV_FIELD/DMY = DATECVT('19991231', 'A8YYMD', 'DMY');
```

例 レガシー日付を日付フォーマットに変換

DATECVT 関数は、HIRE_DATE のフォーマットを I6YMD レガシー日付フォーマットから YYMD 日付フォーマットに変換します。

```
COMPUTE NEW_HIRE_DATE/YYMD = DATECVT(HIRE_DATE, 'I6YMD', 'YYMD');
```

DATEDIF - 2 つの日付の差を計算

手順/構文

2 つの日付の差を計算

DATEDIF 関数は、2 つの日付の差を単位で返します。単位は次のいずれかです。

- **年** DATEDIF で年単位を使用すると、DATEADD の逆の結果が返されます。日付 X から 1 年を減算して日付 Y を作成する場合、X と Y の年差は 1 です。2 月 29 日から 1 年を減算すると、結果は 2 月 28 日になります。
- **月** DATEDIF で月単位を使用すると、DATEADD の逆の結果が返されます。日付 X から 1 か月を減算して日付 Y を作成する場合、X と Y の月差は 1 です。to_date が月の最終日である場合、逆数計算規則を保障するために、月差は絶対数で切り上げられる可能性があります。

入力日付の1つまたは両方が月の最終日である場合、この規則が適用されます。これにより、1月31日と4月30日の月差は、2か月ではなく3か月になります。

- **日付**
- **平日** 平日単位を使用すると、DATEDIFは日付の計算から土曜日と日曜日を除外します。これにより、金曜日と月曜日の差は1日になります。
- **営業日** 営業日単位を使用すると、DATEIFはBUSDAYSパラメータおよび祝日ファイルにより、営業日を決定します。これ以外は無視されます。これにより、月曜日が営業日でない場合、金曜日と火曜日の差は1日になります。

DATEDIFは、整数を返します。2つの日付の差が整数でない場合、DATEDIFは値を切り捨て、次の最大整数値を返します。たとえば、2001年3月2日と2002年3月1日の年差は0(ゼロ)です。終了日が開始日よりも前である場合、DATEDIFは負の値を返します。

日を基準としない日付(例、YMまたはYQ)の差の計算は、DATEDIF関数を使用せずに、直接実行してください。

構文

2つの日付の差を計算

```
DATEDIF(from_date, to_date, 'unit')
```

説明

from_date

日付

差を計算する開始日です。年月日を含む完全な日付形式です。

to_date

日付

差を計算する終了日です。

unit

文字

次のいずれかを一重引用符 (') で囲んで指定します。

Y - 構成要素「年」を示します。

M - 構成要素「月」を示します。

D - 構成要素「日」を示します。

WD - 構成要素「平日」を示します。

BD - 構成要素「営業日」を示します。

例 2つの日付の差に基づき平日日数を計算

DATECVT 関数は、HIRE_DATE と DAT_INC 内のレガシー日付を日付フォーマット YYMD に変換します。DATEDIF 関数は、これらの日付フォーマットで、NEW_HIRE_DATE と NEW_DAT_INC の差から平日日数を計算します。

```
COMPUTE NEW_HIRE_DATE/YYMD = DATECVT(HIRE_DATE, 'I6YMD', 'YYMD'); AND
COMPUTE NEW_DAT_INC/YYMD = DATECVT(DAT_INC, 'I6YMD', 'YYMD'); AND
COMPUTE WDAYS_HIRED/I8 = DATEDIF(NEW_HIRE_DATE, NEW_DAT_INC, 'WD');
```

DATEMOV - 日付を有効な位置に移動

手順/構文

日付を指定の位置に移動

DATEMOV 関数は、日付を有効な位置に移動します。

DATEMOV 関数は、年月日を含む完全な日付形式でのみ有効です。

構文 日付を指定の位置に移動

```
DATEMOV(date, 'move-point')
```

説明

date

日付

年月日を含む完全な日付形式です。移動する日付です。

move-point

文字

日付を移動する有効な位置です。一重引用符 (') で囲みます。無効な位置を指定すると、リターンコード 0 (ゼロ) が返されます。有効な値には、次のものがあります。

EOM - 月の終わりです。

BOM - 月のはじめです。

EOQ - 四半期の終わりです。

BOQ - 四半期のはじめです。

EOY - 年の終わりです。

BOY - 年のはじめです。

EOW - 週の終わりです。

BOW - 週のはじめです。

NWD - 次の平日です。

NBD - 次の営業日です。

PWD - 先週の平日です。

PBD - 前回の営業日です。

WD- - 平日またはそれ以前です。

BD- - 営業日またはそれ以前です。

WD+ - 平日またはそれ以降です。

BD+ - 営業日またはそれ以降です。

営業日は **BUSDAYS** および **HDAY** パラメータの設定により計算されます。

例 週の最終日を計算

DATEMOV 関数は、**NEW_DATE** 内の各日付の週の最終日を決定し、結果を **EOW** に格納します。

```
COMPUTE NEW_DATE/YYMDWT = DATECVT(HIRE_DATE, 'I6YMD', 'YYMDWT'); AND  
COMPUTE EOW/YYMDWT = DATEMOV(NEW_DATE, 'EOW');
```


日付構成要素を整数として取得

手順/構文

日付構成要素を抽出して整数を取得

DPART 関数は、日付フィールドから特定の構成要素を抽出し、それを数値フォーマットで返します。

構文 **日付構成要素を抽出して整数を取得**

```
DPART(datevalue, 'component', outfield)
```

説明

datevalue

日付

年月日を含む完全な日付形式です。

component

文字

取得される構成要素名です。文字列は一重引用符 (') で囲みます。有効な値には、次のものがあります。

年 - YEAR、YY

月 - MONTH、MM

日 - DAY、DAY-OF-MONTH

四半期 - QUARTER、QQ

outfield

整数

結果を格納するフィールド名、または出力値の整数フォーマットです。フォーマットの場合は一重引用符 (') で囲みます。

例 日付構成要素を整数フォーマットで抽出

次のリクエストは、VIDEOTRK データソースが使用され、DPART 関数によって TRANSDATE フィールドから年、月、日付コンポーネントを抽出します。

```
DEFINE FILE
  VIDEOTRK
  YEAR/I4 = DPART(TRANSDATE, 'YEAR', 'I4');
  MONTH/I4 = DPART(TRANSDATE, 'MM', 'I4');
  DAY/I4 = DPART(TRANSDATE, 'DAY', 'I4');
END

TABLE FILE VIDEOTRK
PRINT TRANSDATE YEAR MONTH DAY
BY LASTNAME BY FIRSTNAME
WHERE LASTNAME LT 'DIAZ'
END
```

出力結果は次のとおりです。

LASTNAME	FIRSTNAME	TRANSDATE	YEAR	MONTH	DAY
-----	-----	-----	----	-----	----
ANDREWS	NATALIA	91/06/19	1991	6	19
		91/06/18	1991	6	18
BAKER	MARIE	91/06/19	1991	6	19
		91/06/17	1991	6	17
BERTAL	MARCIA	91/06/23	1991	6	23
		91/06/18	1991	6	18
CHANG	ROBERT	91/06/28	1991	6	28
		91/06/27	1991	6	27
		91/06/26	1991	6	26
COLE	ALLISON	91/06/24	1991	6	24
		91/06/23	1991	6	23
CRUZ	IVY	91/06/27	1991	6	27
DAVIS	JASON	91/06/24	1991	6	24

DATETRAN - 日付を国際フォーマットに変換

手順/構文

日付を国際フォーマットに変換

参照

DATETRAN 関数使用上の注意

DATETRAN 関数は、日付を国際フォーマットに変換します。

構文

日付を国際フォーマットに変換

```
DATETRAN (indate, '(intype)', '([formatops]')', 'lang', outlen, 'outfield')
```

説明

indate

フォーマットを設定する入力日付です。日付フォーマットに、日付表示オプション付きの文字または数値フォーマットを使用することはできません。

intype

次のいずれかの文字列は、入力日付構成要素とその表示順序を示します。文字列は、一重引用符 (') で囲みます。

入力構成要素が 1 つ

1つの構成要素の入力タイプ	説明
'(W)'	曜日構成要素のみ (元のフォーマットは「W」のみ)
'(M)'	月構成要素のみ (元のフォーマットは「M」のみ)

入力構成要素が 2 つ

2つの構成要素の入力タイプ	説明
'(YYM)'	4 桁の西暦年、月
'(YM)'	2 桁の西暦年、月
'(MY)'	月、2 桁の西暦年
'(MYY)'	月、4 桁の西暦年

入力構成要素が 3 つ

3つの構成要素の入力タイプ	説明
'(YYMD)'	4桁の西暦年、月、日
'(YMD)'	2桁の西暦年、月、日
'(DMYY)'	日、月、4桁の西暦年
'(DMY)'	日、月、2桁の西暦年
'(MDYY)'	月、日、4桁の西暦年
'(MDY)'	月、日、2桁の西暦年
'(MD)'	月、日 (年月日の日付から抽出。西暦年は無視)
'(DM)'	日、月 (年月日の日付から抽出。西暦年は無視)

formatops

0 (ゼロ)、またはフォーマットオプションを表す文字列です。文字列は括弧および一重引用符 (') で囲みます。括弧と引用符 (') は、フォーマットオプションを指定しない場合でも必要です。フォーマットオプションは、次のとおりです。

- 月または日の数値の先頭の 0 (ゼロ) を非表示にするオプション。
- 月または日構成要素を完全な名前または略名に変換するオプション。変換先の文字は、すべて大文字に指定することも、言語のデフォルト値 (先頭大文字、またはすべて小文字) に指定することも可能です。
- 日付の区切り文字オプション、および日付にカンマ (,) を付けるオプション。

先行する 0 (ゼロ) を非表示にするオプションには次のものがあります。

フォーマットオプション	説明
m	月部分の 0 (ゼロ) を省略 (1 月から 9 月を 01 から 09 ではなく 1 から 9 で表示) します。

フォーマットオプション	説明
d	1桁の日を01から09ではなく、1から9として表示します。
dp	1桁の日を01から09ではなく、1から9として表示します。数値の後にはピリオド(.)が追加されます。
do	1桁の日を1から9として表示します。英語(言語コードEN)でのみ、数値の後に序数を表す接尾語(st、nd、rd、th)が追加されます。

有効な月名および日付オプションは次のとおりです。

フォーマットオプション	説明
T	月の略名がピリオド(.)なしで表示されます。すべて大文字です。
TR	完全な月名が表示されます。すべて大文字です。
Tp	月の略名が末尾にピリオド(.)を伴って表示されます。すべて大文字です。
t	月の略名がピリオド(.)なしで表示されます。言語コードによって、すべて小文字または先頭大文字で表示されます。
tr	完全な月名が表示されます。言語コードによって、すべて小文字または先頭大文字で表示されます。
tp	月の略名が末尾にピリオド(.)を伴って表示されます。名前の大文字、小文字は選択する言語のデフォルト値が使用されます(例、フランス語およびスペイン語ではすべて小文字、英語およびドイツ語では先頭大文字)。
W	日付の先頭に曜日の略名が表示されます。すべて大文字で、ピリオド(.)は使用されません。

フォーマットオプション	説明
WR	日付の先頭に完全な曜日名が表示されます。すべて大文字です。
Wp	日付の先頭に曜日の略名が表示されます。すべて大文字で、末尾にはピリオド (.) が追加されます。
w	日付の先頭に曜日の略名が区切り記号なしで表示されます。ピリオド (.) は使用されません。名前の大文字、小文字は選択する言語のデフォルト値が使用されます (例、フランス語およびスペイン語ではすべて小文字、英語およびドイツ語では先頭大文字)。
wr	日付の先頭に完全な曜日名が表示されます。名前の大文字、小文字は選択する言語のデフォルト値が使用されます (例、フランス語およびスペイン語ではすべて小文字、英語およびドイツ語では先頭大文字)。
wp	日付先頭に曜日の略名が表示されます。末尾にはピリオド (.) が追加されます。名前の大文字、小文字は選択する言語のデフォルト値が使用されます (例、フランス語およびスペイン語ではすべて小文字、英語およびドイツ語では先頭大文字)。
X	日付の末尾に曜日の略名が表示されます。すべて大文字で、ピリオド (.) は使用されません。
XR	日付の末尾に完全な曜日名が表示されます。すべて大文字です。
Xp	日付の末尾に曜日の略名が表示されます。すべて大文字で、末尾にはピリオド (.) が追加されます。
x	日付の末尾に曜日の略名がピリオド (.) なしで表示されます。名前の大文字、小文字は選択する言語のデフォルト値が使用されます (例、フランス語およびスペイン語ではすべて小文字、英語およびドイツ語では先頭大文字)。

フォーマットオプション	説明
<code>xr</code>	日付の末尾に完全な曜日名が表示されます。名前の大文字、小文字は選択する言語のデフォルト値が使用されます (例、フランス語およびスペイン語ではすべて小文字、英語およびドイツ語では先頭大文字)。
<code>xp</code>	日付の末尾に曜日の略名がピリオド (.) を伴って表示されます。名前の大文字、小文字は選択する言語のデフォルト値が使用されます (例、フランス語およびスペイン語ではすべて小文字、英語およびドイツ語では先頭大文字)。

有効な日付の区切り文字オプションは、次のとおりです。

フォーマットオプション	説明
<code>B</code>	構成要素の区切り文字に空白を 1 つ使用します。このオプションは、月名および曜日が文字に変換されている場合、またはカンマ (,) が使用されている場合のデフォルト値です。
<code>.</code>	構成要素の区切り文字にピリオド (.) を使用します。
<code>-</code>	構成要素の区切り文字にマイナス記号 (-) を使用します。このオプションは、空白がデフォルトの区切り文字として使用できない場合のデフォルト値です。
<code>/</code>	構成要素の区切り文字にスラッシュ (/) を使用します。
<code> </code>	構成要素の区切り文字を省略します。
<code>K</code>	構成要素の区切り文字に適切なアジア言語の文字を使用します。

フォーマットオプション	説明
c	月の末尾にカンマ (,) を追加します (T、Tp、TR、t、tp、tr の後ろに追加)。 日の末尾にカンマ (,) とブランクを 1 つずつ追加します (W、Wp、WR、w、wp、wr の後ろに追加)。 日の前にカンマ (,) とブランクを 1 つずつ追加します (X、XR、x、xr の後ろに追加)。
e	スペイン語やポルトガル語の「de」または「DE」を日と月の間、および月と年の間に表示します。大文字と小文字の使用は、月名に一致します。月名が大文字の場合は「DE」、小文字の場合は「de」が表示されます。DMY、DMYY、MY、MYY フォーマットで役立ちます。
D	日と指定された区切り文字の間にカンマ (,) を挿入します。
Y	年と指定された区切り文字の間にカンマ (,) を挿入します。

lang

日付が変換される言語の 2 バイトの標準 ISO コードです。文字は一重引用符 (') で囲みます。以下は、有効な言語コードです。

'AR' アラビア語

'CS' チェコ語

'DA' デンマーク語

'DE' ドイツ語

'DU' オランダ語

'EN' 英語

'ES' スペイン語

'FI' フィンランド語

'FR' フランス語

'EL' ギリシャ語

'IW' ヘブライ語
'IT' イタリア語
'JA' 日本語
'KO' 韓国語
'LT' リトアニア語
'NO' ノルウェー語
'PO' ポーランド語
'PT' ポルトガル語
'RU' ロシア語
'SV' スウェーデン語
'TH' タイ語
'TR' トルコ語
'TW' 中国語 (繁体字)
'ZH' 中国語 (簡体字)

outlen

数値

出力フィールドの長さをバイト数で指定します。フィールドの長さが値よりも小さい場合は、すべてがブランクの結果が返されます。長さが大きすぎる場合は、右側にブランクが挿入されます。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

参照

DATETRAN 関数使用上の注意

- 月名および曜日名の長さは異なるため、タイプ A 出力フィールドにはさまざまな長さの情報が格納されます。また、0 (ゼロ) を非表示にするオプションを選択した場合、月および日付の数字の長さは、1 バイト、2 バイトのいずれかになります。使用されないバイトの部分には、ブランクが挿入されます。

- 入力が無効または不整合の場合は、0 (ゼロ) が出力されます。データが欠落している場合も空白が出力されます。
- 基準日 (1900-12-31 および 1900-12、または 1901-01) は、DATEDISPLAY 設定が ON である状態として処理され、自動的に空白として表示されません。基準日 (内部整数値 0 を含む) の出力を非表示にするには、DATETRAN 関数を呼び出す前に 0 (ゼロ) をテストします。以下はその例です。

```
RESULT/A40 = IF DATE EQ 0 THEN ' ' ELSE  
              DATETRAN (DATE, '(YYMD)', '(.t)', 'FR', 40, 'A40');
```

- 変換後の日付構成要素の有効値は、「DTLNGIng」という名前のファイルに含まれています。Ing は、言語を指定する 3 バイトのコードです。これらのファイルは、日付の変換先の各言語からアクセス可能である必要があります。

例 DATETRAN 関数の使用

DATETRAN 関数は、フランス語でデフォルトの大文字小文字を使用して、曜日を表示します。

```
COMPUTE OUT/A8=DATETRAN(DATEW, '(W)', '(wr)', 'FR', 8 , 'A8') ;
```

日付時間値の精度

手順/構文

日付時間値の精度

以前のバージョンでは、日付時間値の秒コンポーネントは 0 (ゼロ)、3、6 桁で表示することができました。現在のバージョンでは、0 (ゼロ) 桁から 9 桁で表示することができません。

日付時間フォーマットの秒コンポーネントを制御するには、フォーマットに 1 から 9 までの桁数を入力します。

構文 日付時間値の精度

3 桁 (ミリ秒、s)、6 桁 (マイクロ秒、m)、9 桁 (ナノ秒、n) 用に、特殊なフォーマットコンポーネントが用意されています。

- **時間値のみ** フォーマットに複数の時間表示コンポーネントが含まれる場合、次のように指定します。

- 時間、分、秒、ミリ秒、マイクロ秒の順にコンポーネントを指定します。
- 最初のコンポーネントには、時、分、秒のいずれかを指定する必要があります。
- 間のコンポーネントを省略することはできません。時を指定した場合、次のコンポーネントは分にしなければならず、秒にすることはできません。
- 秒(S)のフォーマットの後に表示桁数を入力します。ここで、ミリ秒(s)、マイクロ秒(m)、ナノ秒(n)コンポーネントを1つかそれ以上指定することもできます。
- **日付コンポーネントと時間コンポーネントの両方** 時間コンポーネントは1文字で、表示する時間の最小単位を指定します。この文字は次のいずれかです。
 - 秒コンポーネントに表示する桁数を指定する1から9までの数字。
 - サポートされる時間コンポーネントのいずれか。この場合、上位の時間コンポーネントもすべて表示されます。

例

日付時間値精度の指定

この変換を実行するには、Developer Workbench のシノニムエディタまたはレポート作成ツールで、[一時項目 (COMPUTE)] (COMPUTE) または [一時項目 (DEFINE)] (DEFINE) ダイアログボックスを使用します。日付が 1999 年 2 月、時間が午前 02:05:25.123456789 であることを想定します。次の一時項目 (DEFINE) は、表示フォーマットおよび関数呼び出しにナノ秒のコンポーネントまたは精度を使用しています。

```

TRANSDT/HYYMDn      = DT(19990205 02:05:25.123456789);
O_HSsmn/HSsmn      = TRANSDT;
O_HHIS2/HHIS2      = TRANSDT;
O_HYYMDn/HYYMDn    = TRANSDT;
O_HYYMD1/HYYMD1    = TRANSDT;
O_HADD/HYYMD9      = HADD(TRANSDT, 'NS', 2, 12, 'HYYMD9');
O_HCNVRT/A26       = HCNVRT(TRANSDT, '(H23)', 23, 'A26');
O_HDIFF/D12.2      = HDIFF(O_HADD, TRANSDT, 'NS', 'D12.2');
TRANSDATE_DATE/YYMD = HDATE(TRANSDT, 'YYMD');
O_HDTTM/HYYMDn     = HDTTM(TRANSDATE_DATE,12, 'HYYMDn');
O_HEXTR/HHIS9      = HEXTR(TRANSDT, 'n',12, 'HHIS9');
O_HGETC/HYYMDn     = HGETC(12, 'HYYMDn');
O_HINPUT/HYYMDn    = HINPUT(14, O_HCNVRT,12, 'HYYMDn');
O_HMASK/HYYMDn     = HMASK(O_HEXTR, 'HSsmn', TRANSDT, 12, 'HYYMDn');
O_HMIDNT/HYYMDn    = HMIDNT(TRANSDT,12, 'HYYMDn');
O_HNAME/A10        = HNAME(TRANSDT, 'NANOSECOND', 'A10');
O_HPART/I10        = HPART(TRANSDT, 'NANOSECOND', 'I10');
O_HSETPT/HYYMDn    = HSETPT(TRANSDT, 'NS', 28, 12, 'HYYMDn');
O_HTIME/P20.2C     = HTIME(12,TRANSDT, 'D16');

```

レポートで、これらの一時項目 (DEFINE) フィールドは、次のように出力されます。

```
TRANSDT    1999/02/05 02:05:25.123456789
O_HSsmn    25.123456789
O_HHIS2     02:05:25.12
O_HYYMDn   1999/02/05 02:05:25.123456789
O_HYYMD1   1999/02/05 02:05:25.1
O_HADD      1999/02/05 02:05:25.123456791
O_HCNVRT    19990205020525123456789
O_HDIFF                    2.00
O_HDTTM     1999/02/05 00:00:00.000000000
O_HEXTR     00:00:00.000000789
O_HGETC     2008/06/25 10:26:52.343644000
O_HINPUT    1999/02/05 02:05:25.000000000
O_HMASK     1999/02/05 00:00:00.000456789
O_HMIDNT    1999/02/05 00:00:00.000000000
O_HNAME     123456789
O_HPART                    123456789
O_HSETPT    1999/02/05 02:05:25.000000028
O_HTIME                    7,525,123,456,789.00
```

注意

- O_HSsmn フィールドには、TRANSDT の秒 (S)、ミリ秒 (s)、マイクロ秒 (m)、ナノ秒 (n) が表示されます。
- O_HHIS2 フィールドには、TRANSDT の時間 (H)、分 (I)、秒 (S) が表示されます。秒は 2 桁で表示されます (2)。
- O_HYYMDn フィールドには、TRANSDT の YYMD フォーマットの日付とナノ秒 (n) までの時間が表示されます。
- O_HYYMD1 フィールドには、TRANSDT の YYMD フォーマットの日付と 1 桁の秒までの時間が表示されます。
- O_HADD フィールドは、HADD 関数を呼び出し、TRANSDT の日付時間値に 2 つのナノ秒を加算して作成されます。
- O_HCNVRT フィールドは、HCNVRT 関数を呼び出し、TRANSDT の日付時間値を文字フォーマットに変換して作成されます。
- O_HDIFF フィールドは、HDIFF 関数を呼び出し、O_HADD の日付時間値から TRANSDT の日付時間値を減算して作成されます。
- O_HDTTM フィールドは、HDTTM 関数を呼び出して TRANSDATE_DATE から日付を取り出し、時間コンポーネントを「0 (ゼロ)」に設定して作成されます。

- O_HEXTR フィールドは、HEXTR 関数を呼び出して TRANSDT からナノ秒 (9 桁のうち下 3 桁) を抽出し、残りのコンポーネントを 0 (ゼロ) に設定して作成されます。
- HGETC フィールドは、HGETC 関数を呼び出して現在の日付時間を取り出し、秒コンポーネントに 9 桁で表示して作成されます。オペレーティングシステムの種類によっては、9 桁すべてを返すとは限らないため、その場合、返されなかった桁は「0」と表示されます。
- O_HINPUT フィールドは、HINPUT 関数を呼び出して O_HCNVRT フィールドに保存された日付時間文字列を日付時間値に変換し、ナノ秒まで表示して作成されます。
- O_HMASK フィールドは、HMASK 関数を呼び出して O_HEXTR から時間、分、秒、ミリ秒、ナノ秒を抽出し、残りのコンポーネントを TRANSDT から抽出して作成されます。
- O_HMIDNT フィールドは、HMIDNT 関数を呼び出して TRANSDT から日付を抽出し、時間を午前零時に設定して作成されます。
- O_HNAME フィールドは、HNAME 関数を呼び出し、TRANSDT から文字フォーマットのナノ秒コンポーネントを抽出して作成されます。
- O_HPART フィールドは、HPART 関数を呼び出し、TRANSDT から文字フォーマットのナノ秒コンポーネントを抽出して作成されます。
- O_HPART フィールドは、HSETPT 関数を呼び出し、ナノ秒コンポーネントの値を 28 に設定して作成されます。
- O_HTIME フィールドは、HTIME 関数を呼び出し、TRANSDT の時間部分をナノ秒の数値に変換して作成されます。

参照

ナノ秒日付時間フォーマットコンポーネント使用上の注意

- 日付フィールドでは、ACTUAL フォーマットの最大値は H12 です。USAGE フォーマットの最大値は H23 です。
- 日付時間関数 HADD、HDIFF、HNAME、HPART、HSETPT は、引数としてコンポーネント名をとることができます。これらの日付時間関数で使用されるナノ秒のコンポーネント名は「nanosecond」で、「ns」という省略形を使用することもできます。さらに、HEXTR 関数と HMASK 関数には、9 桁のうち下 3 桁を表す「n」という新しいコンポーネントが追加されています。
- 次の関数には、日付時間フォーマットの精度によって長さが決定される引数がありません。

- HADD、HDIFF、HINPUT、HMIDNT、HSETPT、HTIME は長さの引数をとります。長さは 8、10、または 12 で、秒の値が 6 桁を超える場合は、必ず 12 を指定します。
- HDTTM と HGETC は日付時間値の長さ引数をとり、8、10、または 12 を指定することが可能ですが、秒の値が 6 桁を超える場合は、必ず 12 を指定します。
- HCNVRT は、変換される日付時間フィールドのフォーマットと長さを指定する引数を 1 つずつとります。これらの引数の長さの上限は 23 です。出力フォーマットには、返される文字すべてを格納するために十分な長さを指定する必要があります。
- 最初の引数が 12 である場合、HTIME は日付時間値の時間部分をナノ秒に変換します。

マスターファイルの DATEPATTERN

トピックス

日付パターン変数の指定

日付パターン定数の指定

データソースには、日付の値が文字フォーマットで格納され、特別な規格が存在せず、年、四半期、月などの構成要素の任意の組み合わせ、および任意の区切り文字を使用するものがあります。レポートにソートが設定されている場合、このようなデータはアルファベット順にソートされ、結果は実務上の意味を持ちません。データフィールドのソート、集計、レポート実行を適切に行うため、DB2 Web Query では、「DATEPATTERN」というマスターファイル属性で指定する変換パターンを使用して、文字フォーマットの日付を標準 DB2 Web Query 日付フォーマットに変換することができます。

パターンの各要素は、実際に入力される特定の文字、または日付構成要素を表す変数です。マスターファイルの USAGE 属性を編集して、日付構成要素の日付パターンを記述する必要があります。DATEPATTERN 文字列の最大長は 64 バイトです。

日付パターン変数の指定

手順/構文

- 日付パターンに年を指定
- 日付パターンに月を表す数値を指定
- 日付パターンに月名を指定
- 日付パターンに曜日を指定
- 日付パターンにユリウス暦の日付を指定
- 日付パターンに曜日を指定
- 日付パターンに四半期を指定

有効な日付構成要素 (変数) は、年、四半期、月、日、曜日です。日付パターン内の変数は、大括弧 ([]) で囲みます。これらの大括弧は、入力または出力の一部ではありません。データに大括弧が含まれる場合は、日付パターン内でエスケープ文字を使用して、データの大括弧を変数を囲む大括弧と区別する必要があります。

構文

日付パターンに年を指定

[YYYY]

4 桁の年を指定します。

[YY]

2 桁の年を指定します。

[yy]

0 (ゼロ) を非表示にする 2 桁の年を指定します (例、2008 年の場合は 8)。

[by]

空白でパディングされた 2 桁の年を指定します。

構文

日付パターンに月を表す数値を指定

[MM]

2 桁の月を表す数値を指定します。

[mm]

0 (ゼロ) を非表示にする月を表す数値を指定します。

[bm]

空白でパディングされた月を表す数値を指定します。

構文 日付パターンに月名を指定

[MON]

3 バイトの月名を大文字で指定します。

[mon]

3 バイトの月名を小文字で指定します。

[Mon]

3 バイトの月名を先頭大文字で指定します。

[MONTH]

月の完全名を大文字で指定します。

[month]

月の完全名を小文字で指定します。

[Month]

月の完全名を先頭大文字で指定します。

構文 日付パターンに曜日を指定

[DD]

2 桁の月単位の日付を指定します。

[dd]

0 (ゼロ) を非表示にする月単位の日付を指定します。

[bd]

空白でパディングされた月単位の日付を指定します。

構文 日付パターンにユリウス暦の日付を指定

[DDD]

3 桁の年単位の日付を指定します。

[ddd]

0 (ゼロ) を非表示にする年単位の日付を指定します。

[bdd]

空白でパディングされた年単位の日付を指定します。

構文 日付パターンに曜日を指定

[WD]

1 桁の曜日を指定します。

[DAY]

3 バイトの曜日を大文字で指定します。

[day]

3 バイトの曜日を小文字で指定します。

[Day]

3 バイトの曜日を先頭大文字で指定します。

[WDAY]

曜日の完全名を大文字で指定します。

[wday]

曜日の完全名を小文字で指定します。

[Wday]

曜日の完全名を先頭大文字で指定します。

曜日の場合、WEEKFIRST の設定で、週の開始日を定義します。

構文 日付パターンに四半期を指定

[Q]

1 桁の四半期番号を指定します (1、2、3、4 のいずれか)。

Q2 や Q02 などの文字列の場合は、[Q] の前に定数を使用します (例、Q0[Q])。

日付パターン定数の指定

変数の間に、任意の定数値を挿入することができます。

通常は変数の一部として解釈される文字を挿入する場合は、円記号 (¥) を使用します。以下はその例です。

□ 定数値として左大括弧 (l) を指定するには、¥[を使用します。

□ 定数値として円記号 (¥) を指定するには、¥¥ を使用します。

一重引用符 (') の場合は、連続する 2 つの一重引用符 (") を使用します。

例 日付パターンサンプル

データソースの日付が「CY 2001 Q1」の形式の場合、DATEPATTERN 属性は次のとおりです。

```
DATEPATTERN = 'CY [YYYY] Q[Q]'
```

データソースの日付が「Jan 31, 01」の形式の場合、DATEPATTERN 属性は次のとおりです。

```
DATEPATTERN = '[Mon] [DD], [YY]'
```

データソースの日付が「APR-06」の形式の場合、DATEPATTERN 属性は次のとおりです。

```
DATEPATTERN = '[MON]-[YY]'
```

データソースの日付が「APR-06」の形式の場合、DATEPATTERN 属性は次のとおりです。

```
DATEPATTERN = '[MON] - [YY]'
```

データソースの日付が「APR '06」の形式の場合、DATEPATTERN 属性は次のとおりです。

```
DATEPATTERN = '[MON] ''[YY]'
```

データソースの日付が「APR[06]」の形式の場合、DATEPATTERN 属性は次のとおりです。

```
DATEPATTERN = '[MON] ¥  
[[YY]¥  
] (or '[MON] ¥  
[[YY]])'
```

右大括弧 (]) はエスケープ文字を表し、必須ではありません。

例 文字の日付によるソート

次の例で、.ftmDATE1 は、以下のデータが格納されたシーケンシャルファイルです。

```

June 1, '02
June 2, '02
June 3, '02
June 10, '02
June 11, '02
June 12, '02
June 20, '02
June 21, '02
June 22, '02
June 1, '03
June 2, '03
June 3, '03
June 10, '03
June 11, '03
June 12, '03
June 20, '03
June 21, '03
June 22, '03
June 1, '04
June 2, '04
June 3, '04
June 4, '04
June 10, '04
June 11, '04
June 12, '04
June 20, '04
June 21, '04
June 22, '04

```

DATE1 マスターファイルの DATE1 フィールドには、USAGE フォーマットおよび ACTUAL フォーマットが設定されます。これらは両者とも文字 (A18) フォーマットです。

```

FILENAME=DATE1, SUFFIX=FIX,
  DATASET = c:¥
tst¥
datel.ftm, $
  SEGMENT=FILE1, SEGTYPE=S0, $
  FIELDNAME=DATE1, ALIAS=E01, USAGE=A18, ACTUAL=A18, $

```

次のリクエストは、DATE1 FIELD でソートします。

```

TABLE FILE DATE1
PRINT DATE1 NOPRINT
BY DATE1
ON TABLE SET PAGE NOPAGE
END

```

出力で、文字の日付は、日付順ではなくアルファベット順にソートされています。

```
DATE1
-----
June 1, '02
June 1, '03
June 1, '04
June 10, '02
June 10, '03
June 10, '04
June 11, '02
June 11, '03
June 11, '04
June 12, '02
June 12, '03
June 12, '04
June 2, '02
June 2, '03
June 2, '04
June 20, '02
June 20, '03
June 20, '04
June 21, '02
June 21, '03
June 21, '04
June 22, '02
June 22, '03
June 22, '04
June 3, '02
June 3, '03
June 3, '04
June 4, '04
```

日付を正しくソートするため、マスターファイルに DATEPATTERN 属性を追加して、DB2 Web Query によって日付が Web Query の日付フィールドに変換されるようにします。また、USAGE フォーマットを編集し、DB2 Web Query の日付フォーマットにする必要があります。適切なパターンを構成するため、格納されている日付のすべての要素を記述する必要があります。文字の日付には、次の変数と定数があります。

- 変数 - 先頭大文字の月の完全名 [Month]。
- 定数 - ブランク。
- 変数 - ゼロを非表示にした月を表す数値 [dd]。
- 定数 - カンマ (,)、ブランク、アポストロフィ (') (パターンでは 2 つのアポストロフィとして記述) の順に入力。

□ 変数 - 2 桁の年 [YY]。

編集後のマスターファイルは、次のようになります。DEFCENT 属性を追加して、2 桁の年を 4 桁の年に変換します。

```
FILENAME=DATE1, SUFFIX=FIX,
  DATASET = c:¥
tst¥
date1.ftm, $
  SEGMENT=FILE1, SEGTYPE=S0, $
  FIELDNAME=DATE1, ALIAS=E01, USAGE=A18, ACTUAL=A18, DEFCENT=20,
  DATEPATTERN = '[Month] [dd], ''[YY]', $
```

同一のリクエストを発行すると、出力は次のようになります。DATE1 は、USAGE で指定した MtrDYY フォーマットの DB2 Web Query の日付に変換されています。

```
DATE1
-----
June  1, 2002
June  2, 2002
June  3, 2002
June 10, 2002
June 11, 2002
June 12, 2002
June 20, 2002
June 21, 2002
June 22, 2002
June  1, 2003
June  2, 2003
June  3, 2003
June 10, 2003
June 11, 2003
June 12, 2003
June 20, 2003
June 21, 2003
June 22, 2003
June  1, 2004
June  2, 2004
June  3, 2004
June  4, 2004
June 10, 2004
June 11, 2004
June 12, 2004
June 20, 2004
June 21, 2004
June 22, 2004
```

DMY、MDY、YMD - 2つの日付の差を計算

手順/構文

2つの日付の差を計算

DMY、MDY、YMD 関数は、数値フォーマットの 2 つのレガシー日付の差を計算します。

構文

2つの日付の差を計算

```
function(begin, end)
```

説明

function

次のいずれかです。

DMY - 日月年フォーマットの 2 つの日付の差を計算します。

MDY - 月日年フォーマットの 2 つの日付の差を計算します。

YMD - 年月日フォーマットの 2 つの日付の差を計算します。

begin

整数、パック 10 進数、または文字フォーマット

日付表示オプションを含む I、P、A フォーマットです。

開始日、または日付を含むフィールド名のいずれかを指定します。

end

整数、パック 10 進数、または文字フォーマット

日付表示オプションを含む I、P、A フォーマットです。

終了日、または日付を含むフィールド名のいずれかを指定します。

例

2つの日付の日数差を計算

YMD 関数は、HIRE_DATE と DATE_INC の日付の日数の差を計算します。

```
COMPUTE DIFF/I4 = YMD(HIRE_DATE, FST.DAT_INC);
```

DOWK および DOWKL - 曜日を検索

手順/構文

曜日を検索

DOWK および DOWKL 関数は、対応する曜日を検索します。DOWK は曜日の 3 バイトの省略形を返し、DOWKL は完全な曜日名を表示します。

構文

曜日を検索

```
{DOWK|DOWKL}(indate, 'outfield')
```

説明

indate

整数 (I6YMD、I8 YMD)

年月日フォーマットのレガシー日付です。日付が無効の場合、関数はブランクを返します。日付が 2 桁の年を指定し、DEFCENT および YRTHRESH の値が設定されていない場合は、20 世紀と見なされます。

outfield

DOWK - 文字

DOWKL - 文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例

曜日を検索

DOWK 関数は、HIRE_DATE フィールドの値に対応する曜日を決定し、結果を DATED に格納します。

```
COMPUTE DATED/A3 = DOWK(HIRE_DATE, 'A3');
```

DT 関数 - 整数を日付に変換

手順/構文

整数を日付に変換

DT 関数は、1899 年 12 月 31 日との日数の差を表す整数を対応する日付に変換します。この関数は、日付を日数に変換して、算術演算を実行する場合に役立ちます。DT 関数は、結果を日付に再変換します。

DT 関数は 6 つあります。それぞれの関数は、数値を異なるフォーマットの日付に変換します。

注意：USERFNS が LOCAL に設定されると、DT 関数は 6 桁の日付のみを表示します。

構文

整数を日付に変換

```
function(number, 'outfield')
```

説明

function

次のいずれかです。

DTDMY - 数値を日月年の日付に変換します。

DTDYM - 数値を日年月の日付に変換します。

DTMDY - 数値を月日年の日付に変換します。

DTMYD - 数値を月年日の日付に変換します。

DTYDM - 数値を年日月の日付に変換します。

DTYMD - 数値を年月日の日付に変換します。

number

整数

1899 年 12 月 31 日以降の日数です。数字は整数に切り捨てられます。

outfield

整数

I6xxx フォーマットです。xxx の部分は、上記の DTxxx に対応します。

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。出力フォーマットは、使用されている関数により異なります。

例 整数を日付に変換

DTMDY 関数は、NEWF フィールド (DAYMD によりに数に変換されたフィールド) を対応する日付に変換し、結果を NEW_HIRE_DATE に格納します。

```
COMPUTE NEWF/I8 WITH EMP_ID = DAYMD(HIRE_DATE, NEWF); AND
COMPUTE NEW_HIRE_DATE/I8MDYY WITH EMP_ID = DTMDY(NEWF, NEW_HIRE_DATE);
```

FIYR - 会計年度の取得

手順/構文

会計年度の取得

FIYR 関数は、会計年度の開始日および会計年度の算定方式に基づいて、特定のカレンダー日付に対応する会計年度を返します。

構文 会計年度の取得

```
FIYR(inputdate, lowcomponent, startmonth, startday, yrnumbering, output)
```

説明

inputdate

日付

会計年度を取得する日付です。この日付には、基準日からのオフセットとして格納される標準の日付を指定する必要があります。

会計年度が月の初日以外から始まる場合、日付の要素は Y(Y)、M、D または Y(Y)、JUL で構成する必要があります (JUL は YJUL と同等)。会計年度が月の初日から始まる場合、日付の要素は Y(Y)、M または Y(Y)、Q で構成する必要があります。

lowcomponent

文字

次のいずれかです。

- D - 日付に D または JUL 構成要素が含まれている場合。

- M - 日付に M 構成要素は含まれているが、D 構成要素が含まれていない場合。
- Q - 日付に Q 構成要素が含まれている場合。

startmonth

数値

1 から 12 までの数字を使用して、会計年度の開始月を表します (例、1 は 1 月、12 は 12 月)。下位構成要素が Q の場合、開始月には 1、4、7、10 のいずれかを指定する必要があります。

startday

数値

開始月の開始日です。通常は 1 を指定します。下位構成要素が M または Q の場合、1 を指定する必要があります。

yrnumbering

文字

有効な値には、次のものがあります。

FYE - 会計年度の終了日を基準にする方式を使用します。会計年度値は、会計年度の最終日のカレンダー一年になります。たとえば、会計年度が 2008 年 10 月 1 日から始まる場合、「2008 年 11 月 1 日」という日付は会計年度 2009 年の第 1 四半期に分類されます。これは、この日付が 2009 年 9 月 30 日に終了する会計年度の範囲内にあるためです。

FYS - 会計年度の開始日を基準にする方式を使用します。この会計年度値は、会計年度の開始日のカレンダー一年になります。たとえば、会計年度が 2008 年 4 月 6 日から始まる場合、「2008 年 7 月 6 日」という日付は会計年度 2008 年の第 2 四半期に分類されます。これは、この日付が 2008 年 4 月 6 日に始まる会計年度の範囲内にあるためです。

output

I、Y、YY

結果は、整数フォーマット、あるいは Y または YY になります。この関数は、年の値を返します。エラーが発生した場合は、0 (ゼロ) が返されます。

注意：会計年度の開始日として 2 月 29 日を使用することはできません。

例 会計年度の取得

次の例は、特定の会計期間(PERIOD フィールド、YYM フォーマット)に対応する会計年度を取得し、サポートされているフォーマット(Y、YY、I4)で値を返します。

```
FISCALYY/YY=FIYR(PERIOD,'M', 4,1,'FYE',FISCALYY);
FISCALY/Y=FIYR(PERIOD,'M', 4,1,'FYE',FISCALY);
FISCALI/I4=FIYR(PERIOD,'M', 4,1,'FYE',FISCALI);
END
```

出力結果では、2002 年 4 月 (2002/04) は会計年度 2003 年に分類されています。これは、開始月が 4 月であり、会計年度の算定方式として FYE が使用されているためです。

Account	PERIOD	FISCALYY	FISCALY	FISCALI
1000	2002/01	2002	02	2002
	2002/02	2002	02	2002
	2002/03	2002	02	2002
	2002/04	2003	03	2003
	2002/05	2003	03	2003
	2002/06	2003	03	2003
2000	2002/01	2002	02	2002
	2002/02	2002	02	2002
	2002/03	2002	02	2002
	2002/04	2003	03	2003
	2002/05	2003	03	2003
	2002/06	2003	03	2003

FIQTR - 会計四半期の取得

手順/構文

会計四半期の取得

FIQTR 関数は、会計年度の開始日および会計年度の算定方式に基づいて、特定のカレンダー日付に対応する会計四半期を返します。

構文 会計四半期の取得

```
FIQTR(inputdate, lowcomponent, startmonth, startday, yrnumbering, output)
```

説明

inputdate

日付

会計年度を取得する日付です。この日付には、基準日からのオフセットとして格納される標準の日付を指定する必要があります。

会計年度が月の初日以外から始まる場合、日付の要素は Y(Y)、M、D または Y(Y)、JUL で構成する必要があります (JUL は YJUL と同等)。会計年度が月の初日から始まる場合、日付の要素は Y(Y)、M または Y(Y)、Q で構成する必要があります。

lowcomponent

文字

次のいずれかです。

- D - 日付に D または JUL 構成要素が含まれている場合。
- M - 日付に M 構成要素は含まれているが、D 構成要素が含まれていない場合。
- Q - 日付に Q 構成要素が含まれている場合。

startmonth

数値

1 から 12 までの数字を使用して、会計年度の開始月を表します (例、1 は 1 月、12 は 12 月)。下位構成要素が Q の場合、開始月には 1、4、7、10 のいずれかを指定する必要があります。

startday

数値

開始月の開始日です。通常は 1 を指定します。下位構成要素が M または Q の場合、1 を指定する必要があります。

yrnumbering

文字

有効な値には、次のものがあります。

FYE - 会計年度の終了日を基準にする方式を使用します。会計年度値は、会計年度の最終日のカレンダー年になります。たとえば、会計年度が 2008 年 10 月 1 日から始まる場合、「2008 年 11 月 1 日」という日付は会計年度 2009 年の第 1 四半期に分類されます。これは、この日付が 2009 年 9 月 30 日に終了する会計年度の範囲内にあるためです。

FYS - 会計年度の開始日を基準にする方式を使用します。この会計年度値は、会計年度の開始日のカレンダー一年になります。たとえば、会計年度が 2008 年 4 月 6 日から始まる場合、「2008 年 7 月 6 日」という日付は会計年度 2008 年の第 2 四半期に分類されます。これは、この日付が 2008 年 4 月 6 日に始まる会計年度の範囲内にあるためです。

output

I または Q

結果は、整数フォーマット、または Q になります。この関数は 1 から 4 までの値を返します。エラーが発生した場合は、0 (ゼロ) が返されます。

注意：会計年度の開始日として 2 月 29 日を使用することはできません。

例

会計四半期の取得

次の例は、特定の従業員の開始日 (START_DATE フィールド、YYMD フォーマット) に対応する会計四半期を取得し、サポートされているフォーマット (Q および I1) で値を返します。

```
FISCALQ/Q=FIQTR (START_DATE, 'D', 10, 1, 'FYE', FISCALQ);
FISCALI/I1=FIQTR (START_DATE, 'D', 10, 1, 'FYE', FISCALI);
```

出力結果では、1998 年 11 月 12 日 (1998/11/12) は第 1 四半期 (Q1) に分類されています。これは、開始月が 10 月であるためです。

Last Name	First Name	Starting Date	FISCALQ	FISCALI
----	-----	-----	-----	-----
CHARNEY	ROSS	1998/09/12	Q4	4
CHIEN	CHRISTINE	1997/10/01	Q1	1
CLEVELAND	PHILIP	1996/07/30	Q4	4
CLINE	STEPHEN	1998/11/12	Q1	1
COHEN	DANIEL	1997/10/05	Q1	1
CORRIVEAU	RAYMOND	1997/12/05	Q1	1
COSSMAN	MARK	1996/12/19	Q1	1
CRONIN	CHRIS	1996/12/03	Q1	1
CROWDER	WESLEY	1996/09/17	Q4	4
CULLEN	DENNIS	1995/09/05	Q4	4
CUMMINGS	JAMES	1993/07/11	Q4	4
CUTLIP	GREGG	1997/03/26	Q2	2

FIYYQ - カレンダー日付を会計日付に変換

手順/構文

カレンダー日付を会計日付に変換

FIYYQ 関数は、指定したカレンダー日付に対応する会計日付を返します。この日付には、会計年度および会計四半期が含まれます。返された会計日付は、会計年度の開始日および会計年度の算定方式に基づいています。

構文 **カレンダー日付を会計日付に変換**

```
FIYYQ(inputdate, lowcomponent, startmonth, startday, yrnumbering, output)
```

説明

inputdate

日付

会計年度を取得する日付です。この日付には、基準日からのオフセットとして格納される標準の日付を指定する必要があります。

会計年度が月の初日以外から始まる場合、日付の要素は Y(Y)、M、D または Y(Y)、JUL で構成する必要があります (JUL は YJUL と同等)。会計年度が月の初日から始まる場合、日付の要素は Y(Y)、M または Y(Y)、Q で構成する必要があります。

lowcomponent

文字

次のいずれかです。

- D - 日付に D または JUL 構成要素が含まれている場合。
- M - 日付に M 構成要素は含まれているが、D 構成要素が含まれていない場合。
- Q - 日付に Q 構成要素が含まれている場合。

startmonth

数値

1 から 12 までの数字を使用して、会計年度の開始月を表します (例、1 は 1 月、12 は 12 月)。下位構成要素が Q の場合、開始月には 1、4、7、10 のいずれかを指定する必要があります。

startday

数値

開始月の開始日です。通常は 1 を指定します。下位構成要素が M または Q の場合、1 を指定する必要があります。

yrnumbering

文字

有効な値には、次のものがあります。

FYE - 会計年度の終了日を基準にする方式を使用します。会計年度値は、会計年度の最終日のカレンダー一年になります。たとえば、会計年度が 2008 年 10 月 1 日から始まる場合、「2008 年 11 月 1 日」という日付は会計年度 2009 年の第 1 四半期に分類されます。これは、この日付が 2009 年 9 月 30 日に終了する会計年度の範囲内にあるためです。

FYS - 会計年度の開始日を基準にする方式を使用します。この会計年度値は、会計年度の開始日のカレンダー一年になります。たとえば、会計年度が 2008 年 4 月 6 日から始まる場合、「2008 年 7 月 6 日」という日付は会計年度 2008 年の第 2 四半期に分類されます。これは、この日付が 2008 年 4 月 6 日に始まる会計年度の範囲内にあるためです。

output

Y[Y]Q または QY[Y]

エラーが発生した場合は、0 (ゼロ) が返されます。

注意：会計年度の開始日として 2 月 29 日を使用することはできません。

例 **カレンダー日付を会計日付に変換**

次の例は、各従業員の開始日 (START_DATE フィールド、YYMD フォーマット) を、年および四半期構成要素を含む会計日付に変換し、サポートされているすべてのフォーマット (YQ、YYQ、QY、QYY) で値を返します。

```
FISYQ/YQ=FIYYQ(START_DATE,'D',10,1,'FYE',FISYQ);
FISYYQ/YYQ=FIYYQ(START_DATE,'D',10,1,'FYE',FISYYQ);
FISQY/QY=FIYYQ(START_DATE,'D',10,1,'FYE',FISQY);
FISQYY/QYY=FIYYQ(START_DATE,'D',10,1,'FYE',FISQYY);
```

出力結果では、1998年11月12日(1998/11/12)が第1四半期(Q1)に変換されています。これは、開始月が10月であり、会計年度の算定方式としてFYEが使用されているためです。

Last Name	First Name	Starting Date	FISYQ	FISYYQ	FISQY	FISQYY
CHARNEY	ROSS	1998/09/12	98 Q4	1998 Q4	Q4 98	Q4 1998
CHIEN	CHRISTINE	1997/10/01	98 Q1	1998 Q1	Q1 98	Q1 1998
CLEVELAND	PHILIP	1996/07/30	96 Q4	1996 Q4	Q4 96	Q4 1996
CLINE	STEPHEN	1998/11/12	99 Q1	1999 Q1	Q1 99	Q1 1999
COHEN	DANIEL	1997/10/05	98 Q1	1998 Q1	Q1 98	Q1 1998
CORRIVEAU	RAYMOND	1997/12/05	98 Q1	1998 Q1	Q1 98	Q1 1998
COSSMAN	MARK	1996/12/19	97 Q1	1997 Q1	Q1 97	Q1 1997
CRONIN	CHRIS	1996/12/03	97 Q1	1997 Q1	Q1 97	Q1 1997
CROWDER	WESLEY	1996/09/17	96 Q4	1996 Q4	Q4 96	Q4 1996
CULLEN	DENNIS	1995/09/05	95 Q4	1995 Q4	Q4 95	Q4 1995
CUMMINGS	JAMES	1993/07/11	93 Q4	1993 Q4	Q4 93	Q4 1993
CUTLIP	GREGG	1997/03/26	97 Q2	1997 Q2	Q2 97	Q2 1997

GREGDT - ユリウス暦から太陽暦フォーマットに変換

手順/構文

ユリウス暦から太陽暦フォーマットに変換

参照

GREGDT の DATEFNS 設定

GREGDT 関数は、日付をユリウス暦フォーマットから太陽暦フォーマット(年月日)に変換します。

ユリウス暦フォーマットの日付は、5桁または7桁の数値です。先頭の2桁または4桁は年、末尾の3桁は1月1日から数えた日数です。たとえば、ユリウス暦フォーマットの1999年1月1日は、99001と1999001のいずれかです。

参照

GREGDT の DATEFNS 設定

GREGDT 関数は、ユリウス暦の日付を YMD または YYMD フォーマットに変換します。必要に応じて、DEFCENT および YRTHRESH パラメータ設定を使用して世紀を決定します。GREGDT 関数は、日付を次のように返します。

DATEFNS 設定	16 または 17 フォーマット	18 フォーマット、またはそれ以上
ON	YMD	YYMD
OFF	YMD	YMD

構文

ユリウス暦から太陽暦フォーマットに変換

```
GREGDT(indate, 'outfield')
```

説明

indate

整数 (15、17)

ユリウス暦の日付です。変換前に整数に切り捨てられます。切り捨て後、各値は 5 桁または 7 桁の数値である必要があります。日付が有効でない場合、この関数は 0 (ゼロ) を返します。

outfield

整数 (16、18、16YMD、18YYMD)

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例

ユリウス暦から太陽暦フォーマットに変換

GREGDT 関数は、JULIAN フィールドを YYMD (太陽暦) フォーマットに変換します。

```
COMPUTE GREG_DATE/I8 = GREGDT(JULIAN, 'I8');
```

HADD - 日付時間値を増加

手順/構文

日付時間値を増加

HADD 関数は、指定した単位数分、日付時間値を増加します。

構文 日付時間値を増加

```
HADD(value, 'component', increment, length, 'outfield')
```

説明

value

日付時間

増加される日付時間値です。値を含む日付時間フィールド、または値を返す式を指定することもできます。

component

文字

増加される構成要素名です。文字列は一重引用符 (') で囲みます。

注意：WEEKDAY は、HADD で有効な構成要素ではありません。

increment

整数

構成要素の増加に使用する単位数です。値を含む数値フィールド名、または値を返す式を指定することもできます。

length

整数

返された日付時間値の長さです。有効な値には、次のものがあります。

8 - ミリ秒を含む時間値です。

10 - マイクロ秒を含む時間値です。

outfield

日付時間

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 日付時間フィールドの月構成要素を増加

HADD 関数は、TRANSDATE の各値に 2 か月加算し、結果を ADD_MONTH に格納します。必要に応じて、日の部分が調整されます。

```
COMPUTE ADD_MONTH/HYYMDS = HADD(TRANSDATE, 'MONTH', 2, 8, 'HYYMDS');
```

HCNVRT - 日付時間値を文字フォーマットに変換

手順/構文

日付時間値を文字フォーマットに変換

HCNVRT 関数は、日付時間値を演算子 EDIT、CONTAINS、LIKE などを使用する文字フォーマットに変換します。

構文 **日付時間値を文字フォーマットに変換**

```
HCNVRT(value, 'fmt', length, 'outfield')
```

説明

value

日付時間

変換される日付時間値です。値を含む日付時間フィールド名、または値を返す式を指定することもできます。

fmt

文字

日付時間フィールドのフォーマットです。フォーマットは一重引用符 (') で囲みます。

length

整数

返される文字フィールド内のバイト数です。実際の値、値を含む文字フィールド名、値を返す式のいずれかを指定します。*length* の値が文字フィールドを表示するために必要なバイト数よりも小さい場合、関数はブランクを返します。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 日付時間フィールドを文字フォーマットに変換

HCVRT 関数は、TRANSDATE フィールドを文字フォーマットに変換します。1 つ目の関数には、フィールドの日付時間表示オプションは含まれません。このオプションは 2 つ目の関数で指定します。入力フィールドの秒の表示も指定します。

```
COMPUTE ALPHA_DATE_TIME1/A20 = HCVRT(TRANSDATE, '(H17)', 17, 'A20'); AND  
COMPUTE ALPHA_DATE_TIME2/A20 = HCVRT(TRANSDATE, '(HYMDS)', 20, 'A20');
```

HDATE - 日付時間値の日付部分を日付フォーマットに変換

手順/構文

日付時間値の日付部分を日付フォーマットに変換

HDATE 関数は、日付時間値の日付の部分を日付フォーマット YYMD に変換します。この結果は、別の日付フォーマットに変換することができます。

構文 日付時間値の日付部分を日付フォーマットに変換

```
HDATE(value, 'YYMD')
```

説明

value

日付時間

変換される日付時間値です。値を含む日付時間フィールド名、または値を返す式を指定することもできます。

YYMD

日付

出力日付フォーマットです。値は常に YYMD です。YYMD は定数値で、この構文では変更できません。このフォーマットは、DEFINE および COMPUTE を実行して変更することができます。

例 日付時間フィールドの日付部分を日付フォーマットに変換

HDATE 関数は、TRANSDATE フィールドの日付部分を日付フォーマット YYMD に変換します。

```
COMPUTE TRANSDATE_DATE/YYMD = HDATE(TRANSDATE, 'YYMD');
```

HDIFF - 2つの日付時間値の差単位数を計算

手順/構文

2つの日付時間値の差を計算

HDIFF 関数は、2つの日付時間値の単位での日数差を計算します。

構文

2つの日付時間値の差を計算

```
HDIFF(value1, value2, 'component', 'outfield')
```

説明

value1

日付時間

終了日付時間値、値を含む日付時間フィールド名、値を返す式のいずれかを指定します。

value2

日付時間

開始日付時間値、値を含む日付時間フィールド名、値を返す式のいずれかを指定します。

component

文字

計算に使用される構成要素名です。文字は一重引用符 (') で囲みます。構成要素が週の場合、計算に WEEKFIRST パラメータの設定が使用されます。

outfield

倍精度または単精度浮動小数点数

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例

2つの日付時間値の単位数を計算

HDIFF 関数は、TRANSDATE と ADD_MONTH フィールドの日数の差を計算し、結果を D12.2 フォーマットで DIFF_PAYS に格納します。

```
COMPUTE ADD_MONTH/HYYMDS = HADD(TRANSDATE, 'MONTH', 2, 8, 'HYYMDS'); AND
COMPUTE DIFF_DAYS/D12.2 = HDIFF(ADD_MONTH, TRANSDATE, 'DAY', 'D12.2');
```

HDTTM - 日付値を日付時間値に変換

手順/構文

日付値を日付時間値に変換

HDTTM 関数は、日付値を日付時間フィールドに変換します。時間部分は午前零時に設定されます。

構文 **日付値を日付時間値に変換**

```
HDTTM(date, length, 'outfield')
```

説明

date

日付

変換される日付値、値を含む日付フィールド名、値を返す式のいずれかを指定します。

length

整数

返された日付時間値の長さです。有効な値には、次のものがあります。

8 - ミリ秒を含む時間値です。

10 - マイクロ秒を含む時間値です。

outfield

日付時間

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 **日付フィールドを日付時間フィールドに変換**

HDTTM 関数は、日付フィールド TRANSDATE_DATE を日付時間フィールドに変換します。

```
COMPUTE TRANSDATE_DATE/YYMD = HDATE(TRANSDATE, 'YYMD'); AND  
COMPUTE DT2/HYYMDIA = HDTTM(TRANSDATE_DATE, 8, 'HYYMDIA');
```

HGETC - 現在の日付および時間を日付時間フィールドに格納

手順/構文

現在の日付および時間を日付時間フィールドに格納

HGETC 関数は、現在の日付と時間を日付時間フォーマットのフィールドに格納します。オペレーティングシステム環境で、ミリ秒やマイクロ秒値が利用できない場合、これらの構成要素には、0 (ゼロ) が取得されます。

構文 **現在の日付および時間を日付時間フィールドに格納**

```
HGETC(length, 'outfield')
```

説明

length

整数

返された日付時間値の長さです。有効な値には、次のものがあります。

8 - ミリ秒を含む時間値です。

10 - マイクロ秒を含む時間値です。

outfield

日付時間

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 **現在の日付および時間を日付時間フィールドに格納**

HGETC 関数は、現在の日付および時間を DT2 へ格納します。

```
COMPUTE DT2/HYYMDm = HGETC(10, 'HYYMDm');
```

HHMMSS - 現在の時間を取得

手順/構文

現在の時間を取得

HHMMSS 関数は、オペレーティングシステムから現在の時間を取得します。時間は、時、分、秒をピリオド (.) で区切った 8 バイトの文字列として取得されます。

構文 **現在の時間を取得**

```
HHMMSS('outfield')
```

説明

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。フィールドフォーマットには、A8 以上を指定します。

例 **現在の時間を取得**

HHMMSS 関数は、現在の時間を取得します。

```
COMPUTE NOWTIME/A8 = HHMMSS('A8');
```

HINPUT - 文字列を日付時間値に変換

手順/構文

文字列を日付時間値に変換

HINPUT 関数は、文字列を日付時間値に変換します。

構文 **文字列を日付時間値に変換**

```
HINPUT(inputlength, 'inputstring', length, 'outfield')
```

説明

inputlength

整数

変換される文字列の長さです。実際の値、値を含む文字フィールド名、値を返す式のいずれかを指定します。

inputstring

文字

変換される文字列です。文字列は一重引用符 (') で囲みます。文字列を含む文字フィールド名、または文字列を返す式を指定することもできます。この文字列には、入力値として有効な任意の日付時間の組み合わせを使用することができます。

length

整数

返された日付時間値の長さです。有効な値には、次のものがあります。

8 - ミリ秒を含む時間値です。

10 - マイクロ秒を含む時間値です。

outfield

日付時間

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 文字列を日付時間値に変換

HCVRT 関数は、TRANSDATE フィールドを文字フォーマットに変換します。その後、HINPUT 関数が文字列を日付時間値に変換します。

```
COMPUTE ALPHA_DATE_TIME/A20 = HCVRT(TRANSDATE, '(H17)', 17, 'A20'); AND
COMPUTE DT_FROM_ALPHA/HYYMDS = HINPUT(14, ALPHA_DATE_TIME, 8, 'HYYMDS');
```

HMIDNT - 日付時間値の時間部分を午前零時に設定

手順/構文

日付時間値の時間部分を午前零時に設定

HMIDNT 関数は、日付時間値の時間部分を午前零時 (デフォルト値はすべて 0 (ゼロ)) に変更します。これにより、日付フィールドを日付時間フィールドと比較することができます。

構文 日付時間値の時間部分を午前零時に設定

```
HMIDNT(value, length, 'outfield')
```

説明

value

日付時間

時間部分が午前零時に設定される日付時間値です。値を含む日付時間フィールド名、または値を返す式を指定することもできます。

length

整数

返された日付時間値の長さです。有効な値には、次のものがあります。

8 - ミリ秒を含む時間値です。

10 - マイクロ秒を含む時間値です。

outfield

日付時間

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 時間を午前零時に設定

HMIDNT 関数は、TRANSDATE フィールドの時間部分を午前零時に、最初は 24 時間制、次に 12 時間制で設定します。

```
COMPUTE TRANSDATE_MID_24/HYYMDS = HMIDNT(TRANSDATE, 8, 'HYYMDS'); AND  
COMPUTE TRANSDATE_MID_12/HYYMDSA = HMIDNT(TRANSDATE, 8, 'HYYMDSA');
```

HNAME - 日付時間構成要素を文字フォーマットで取得

手順/構文

日付時間構成要素を文字フォーマットで取得

HNAME 関数は、日付時間値から指定された構成要素を文字フォーマットで抽出します。

構文 日付時間構成要素を文字フォーマットで取得

```
HNAME(value, 'component', 'outfield')
```

説明

value

日付時間

構成要素が抽出される日付時間値、値を含む日付時間フィールド名、値を返す式のいずれかを指定します。

component

文字

取得される構成要素名です。文字列は一重引用符 (') で囲みます。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。フィールドフォーマットには、A2 以上を指定します。

関数は、他のすべての構成要素を数値にのみ変換します。年は常に 4 桁であり、時間は 24 時間制と見なされます。

例 週構成要素を文字フォーマットで取得

HNAME 関数は、TRANSDATE フィールドから週を文字フォーマットで取得します。WEEKFIRST パラメータ設定の変更は、構成要素の値を変更します。

```
COMPUTE WEEK_COMPONENT/A10 = HNAME(TRANSDATE, 'WEEK', 'A10');
```

例 日構成要素を文字フォーマットで取得

HNAME 関数は、TRANSDATE フィールドから日を文字フォーマットで取得します。

```
COMPUTE DAY_COMPONENT/A2 = HNAME(TRANSDATE, 'DAY', 'A2');
```

HPART - 日付時間構成要素を数値フォーマットで取得

手順/構文

日付時間構成要素を数値フォーマットで取得

HPART 関数は、指定した構成要素を日付時間値から抽出し、数値フォーマットで返します。

構文 日付時間構成要素を数値フォーマットで取得

```
HPART(value, 'component', 'outfield')
```

説明

value

日付時間

日付時間値、値を含む日付時間フィールド名、値を返す式のいずれかを指定します。

component

文字

取得される構成要素名です。文字列は一重引用符 (') で囲みます。

outfield

整数

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 日構成要素を数値フォーマットで取得

HPART 関数は、TRANSDATE フィールドから日を整数フォーマットで取得します。

```
COMPUTE DAY_COMPONENT/I2 = HPART(TRANSDATE, 'DAY', 'I2');
```

HSETPT - 日付時間値に構成要素を挿入

手順/構文

日付時間値に構成要素を挿入

HSETPT 関数は、指定した構成部分の数値を日付時間値に挿入します。

構文 日付時間値に構成要素を挿入

```
HSETPT(dtfield, 'component', value, length, 'outfield')
```

説明

dtfield

日付時間

日付時間値、値を含む日付時間フィールド名、値を返す式のいずれかを指定します。

component

文字

挿入される構成要素名です。文字列は一重引用符 (') で囲みます。

value

整数

要求された構成要素に挿入される数値、値を含む数値フィールド名、値を返す式のいずれかを指定します。

length

整数

返された日付時間値の長さです。有効な値には、次のものがあります。

8 - ミリ秒を含む時間値です。

10 - マイクロ秒を含む時間値です。

outfield

日付時間

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 日付時間フィールドに日構成要素を挿入

HSETPT 関数は、日の数値「28」を ADD_MONTH フィールドに挿入し、結果を INSERT_DAY に格納します。

```
COMPUTE ADD_MONTH/HYYMDS = HADD(TRANSDATE, 'MONTH', 2, 8, 'HYYMDS'); AND
COMPUTE INSERT_DAY/HYYMDS = HSETPT(ADD_MONTH, 'DAY', 28, 8, 'HYYMDS');
```

HTIME - 日付時間値の時間部分を数値に変換

手順/構文

日付時間値の時間部分を数値に変換

HTIME 関数は、1 つ目の引数が 8 の場合、日付時間値の時間部分をミリ秒の数値に変換し、1 つ目の引数が 10 の場合、マイクロ秒の数値に変換します。マイクロ秒を含めるには、入力日付時間値は 10 バイトでなければなりません。

構文 日付時間値の時間部分を数値に変換

```
HTIME(length, value, 'outfield')
```

説明

length

整数

入力日付時間値の長さです。有効な値には、次のものがあります。

8 - ミリ秒を含む時間値です。

10 - マイクロ秒を含む時間値です。

value

日付時間

時間に変換される日付時間値、値を含む日付時間フィールド名、値を返す式のいずれかを指定します。

outfield

倍精度または単精度浮動小数点数

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 日付時間フィールドの時間部分を数値に変換

HTIME 関数は、TRANSDATE フィールドの時間部分をミリ秒数に変換します。

```
COMPUTE MILLISEC/D12.2 = HTIME(8, TRANSDATE, 'D12.2');
```

JULDAT - 太陽暦からユリウス暦フォーマットに変換

手順/構文

太陽暦からユリウス暦フォーマットに変換

参照

JULDAT の DATEFNS 設定

JULDAT 関数は、日付を太陽暦フォーマット (年月日) からユリウス暦フォーマット (年日) に変換します。ユリウス暦フォーマットの日付は、5 桁または 7 桁の数値です。先頭の 2 桁または 4 桁は年、末尾の 3 桁は 1 月 1 日から数えた日数です。たとえば、ユリウス暦フォーマットの 1999 年 1 月 1 日は、99001 と 1999001 のいずれかです。

参照

JULDAT の DATEFNS 設定

JULDAT 関数は、太陽暦の日付を YYNNN または YYYYNNN フォーマットに変換します。必要に応じて、DEFCENT および YRTHRESH パラメータ設定を使用して世紀を決定します。

JULDAT 関数は、次のデータを返します。

DATEFNS 設定	16 または 17 フォーマット	18 フォーマット、またはそれ以上
ON	YYNNN	YYYYNNN
OFF	YYNNN	YYNNN

構文 **太陽暦からユリウス暦フォーマットに変換**

```
JULDAT(indate, 'outfield')
```

説明

indate

整数 (I6、I8、I6YMD、I8YYMD)

日付、または年月日フォーマット (YMD または YYMD) の日付を含むフィールド名のいずれかを指定します。

outfield

整数 (I5、I7)

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 **太陽暦からユリウス暦フォーマットに変換**

JULDAT 関数は、HIRE_DATE フィールドをユリウス暦フォーマットに変換します。

```
COMPUTE JULIAN/I7 = JULDAT(HIRE_DATE, 'I7');
```

TIMETOTS - 時間をタイムスタンプに変換**手順/構文**

時間をタイムスタンプに変換

TIMETOTS 関数は、時間をタイムスタンプに変換します。値の日付構成要素には、現在の日付が使用されます。1 つ目の引数は、H (日付時間) フォーマットである必要があります。日付構成要素は、現在の日付に設定されます。

構文 **時間をタイムスタンプに変換**

```
TIMETOTS (time, length, 'outfield')
```

説明

time

日付時間

日付時間フォーマットで表される時間です。

length

整数

結果の長さです。次のいずれかを指定することができます。

8 - ミリ秒を含む時間値です。

10 - マイクロ秒を含む入力時間値です。

outfield

日付時間

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 時間をタイムスタンプに変換

TIMETOTS 関数は、時間引数をタイムスタンプに変換します。

```
COMPUTE TSTMPSEC/HYYMDS = TIMETOTS(TMSEC, 8, 'HYYMDS'); AND  
COMPUTE TSTMPMILLI/HYYMDm = TIMETOTS(TMMILLI, 10, 'HYYMDm');
```

TODAY - 現在の日付を取得

手順/構文

現在の日付を取得

TODAY 関数は、MM/DD/YY または MM/DD/YYYY フォーマットでオペレーティングシステムから現在の日付を取得します。常に現在の日付が返されます。このため、深夜にアプリケーションを実行する場合は、TODAY を使用することをお勧めします。デフォルト設定で挿入されたスラッシュ記号 (/) を削除するには、EDIT 関数を使用します。

構文 現在の日付を取得

```
TODAY('outfield')
```

説明

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。フィールドフォーマットには、A8 以上を指定します。以下のように適用します。

- DATEFNS=ON でフォーマットが A8 または A9 の場合、TODAY は 2 桁の年を返します。
- DATEFNS=ON でフォーマットが A10 以上の場合、TODAY は 4 桁の年を返します。

- DATEFNS=OFF の場合、*outfield* のフォーマットに関わらず、TODAY は 2 桁の年を返します。

例 現在の日付を取得

TODAY は、現在の日付を取得し、DATE フィールドに格納します。

```
COMPUTE DATE/A10 = TODAY('A10');
```

YM - 経過月数を計算

手順/構文

経過月数を計算

YM 関数は、日付から日付までの経過月数を計算します。日付は年月フォーマットである必要があります。CHGDAT または EDIT 関数を使用することにより、日付をこのフォーマットに変換することができます。

構文 経過月数を計算

```
YM(fromdate, todate, 'outfield')
```

説明

fromdate

整数 (I4YM、I6YYM)

年月日の開始日付です (例、I4YM)。日付が有効でない場合、この関数は 0 (ゼロ) を返します。

todate

整数 (I4YM、I6YYM)

整数年月フォーマットのレガシー日付です。日付が有効でない場合、この関数は 0 (ゼロ) を返します。

outfield

整数

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

ヒント：*fromdate* または *todate* が整数の年月日フォーマット (I6YMD または I8YYMD) の場合、年月フォーマットを変換し、結果を整数に設定するには、100 で除算します。これにより日付の日の部分が削除されます。日は、小数部分で表されます。

例 経過月数を計算

COMPUTE コマンドは、年月日の日付を年月フォーマットに変換します。その後、YM は HIRE_DATE/100 と DAT_INC/100 フィールドの値の差を計算します。

```
COMPUTE HIRE_MONTH/I4YM = HIRE_DATE/100; AND  
COMPUTE MONTH_INC/I4YM = DAT_INC/100; AND  
COMPUTE MONTHS_HIRED/I3 = YM(HIRE_MONTH, MONTH_INC, 'I3');
```

フォーマット変換関数

トピックス

ATODBL - 文字列を倍精度浮動小数点数フォーマットに変換

EDIT - フィールドのフォーマットを変換

FTOA - 数値を文字フォーマットに変換

HEXBYT - 10 進数を文字に変換

ITONUM - 整数を倍精度小数点数フォーマットに変換

ITOPACK - 整数をパック 10 進数フォーマットに変換

ITOZ - 数値をゾーン 10 進数フォーマットに変換

PCKOUT - 指定した長さでパック 10 進数を書き込み

フォーマット変換関数は、フィールドのフォーマットを変換します。

ATODBL - 文字列を倍精度浮動小数点数フォーマットに変換

手順/構文

文字列を倍精度浮動小数点数フォーマットに変換

ATODBL 関数は、文字列を実数 (倍精度浮動小数点数) フォーマットに変換します。

構文 文字列を倍精度浮動小数点数フォーマットに変換

```
ATODBL(string, length, 'outfield')
```

説明

string

文字

変換される文字列です。文字列を含むフィールドを指定することもできます。

length

文字

infield の 2 バイトの文字列です。数値定数、または値を含むフィールドです。数値定数を指定する場合は、一重引用符 (') で囲みます (例、'12')。最大値は 15 です。

outfield

倍精度浮動小数点数

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 文字列を倍精度浮動小数点数フォーマットに変換

ATODBL 関数は、EMP_ID フィールドを倍精度小数点数フォーマットに変換し、結果を D_EMP_ID に格納します。

```
COMPUTE D_EMP_ID/D12.2 = ATODBL(EMP_ID, '09', 'D12.2');
```

EDIT - フィールドのフォーマットを変換

手順/構文

フィールドのフォーマットを変換

EDIT 関数は、数値を含む文字フィールドを数値フォーマットに変換、または数値フィールドを文字フォーマットに変換します。特定のフォーマットが必要なコマンドを使用してフィールドを操作する場合に役立ちます。

EDIT で変換後の値を新しいフィールドに割り当てるときは、新しいフィールドのフォーマットが返された値のものと同一である必要があります。たとえば、EDIT で数値フィールドを文字フォーマットに変換する場合、新しいフィールドは文字フォーマットである必要があります。

```
DEFINE ALPHAPRICE/A6 = EDIT(PRICE);
```

EDIT 関数は、特殊文字を次のように処理します。

- 文字フィールドを数値フィールドに変換する場合、フィールドの符号や小数点は受容可能で、数値フィールドに格納されます。
- 浮動小数点数またはパック 10 進数フィールドを文字フォーマットに変換するときは、符号、小数点、小数点以下のすべての数字は削除されます。さらに残りの数字を右揃えし、指定されたフィールドの長さに達するまで、先頭に 0 (ゼロ) を追加します。浮動小数点数またはパック 10 進数フォーマットの 10 桁以上の数値を変換すると、結果に誤りが生じる可能性があります。

また、EDIT 関数を使用して、文字列から文字を抽出したり、文字列に文字を追加したりすることもできます。詳細は、249 ページの「[EDIT - 文字を抽出または追加](#)」を参照してください。

構文 **フィールドのフォーマットを変換**

```
EDIT(fieldname);
```

説明

fieldname

文字または数値

フィールド名です。

例 **数値を文字フォーマットに変換**

EDIT 関数は、HIRE_DATE (レガシー日付フォーマット) を文字フォーマットに変換します。これにより、文字フォーマットをとる CHGDAT 関数でこのフィールドが使用できるようになります。

```
COMPUTE ALPHA_HIRE/A17 = EDIT(HIRE_DATE); AND  
COMPUTE HIRE_MDY/A17 = CHGDAT('YMD', 'MDYYX', ALPHA_HIRE, 'A17');
```

FTOA - 数値を文字フォーマットに変換

手順/構文

数値を文字フォーマットに変換

FTOA 関数は、16 桁以内の数値を文字フォーマットに変換します。数値の小数点の位置を保持し、先頭に空白を追加して右揃えします。FTOA で変換する数値には、編集オプションを追加することができます。

FTOA を使用して小数部を含む値を文字列に変換するときは、数値の整数部分と小数点以下を格納するために十分な大きさの文字フォーマットを指定する必要があります。たとえば、D12.2 は、A14 に変換されます。出力フォーマットの大きさが十分でない場合、小数点以下は切り捨てられます。

構文

数値を文字フォーマットに変換

```
FTOA(number, '(format)', 'outfield')
```

説明

number

数値 F または D (単精度および倍精度浮動小数点数)

変換される数値です。数値を含むフィールド名を指定することもできます。

format

文字

数値の出力フォーマットです。文字列は一重引用符 (') および括弧で囲みます。浮動小数点数は、単精度および倍精度フォーマットのみがサポートされています。出力に表示する編集オプションをすべて含めます。D (倍精度浮動小数点数) フォーマットを指定すると、カンマ (,) が自動的に追加されます。

この引数のフィールド名を使用する場合、引用符や括弧を使用せずに名前を入力します。フォーマットを指定する場合、括弧で囲む必要があります。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。この引数の長さは数値の長さよりも大きくする必要があります。編集オプションおよび負の符号が追加される可能性も考慮します。

例 数値を文字フォーマットに変換

FTOA 関数は、GROSS フィールドのフォーマットを倍精度浮動小数点数フォーマットから文字フォーマットに変換し、結果を ALPHA_GROSS に格納します。

```
COMPUTE ALPHA_GROSS/A15 = FTOA(GROSS, '(D12.2)', 'A15');
```

HEXBYT - 10 進数を文字に変換

手順/構文

10 進数を文字に変換

HEXBYT 関数は、10 進数の整数に対応する ASCII、EBCDIC、Unicode のいずれかの文字を取得します。取得される文字は、構成およびオペレーティングシステムにより異なります。この関数は、ASCII、EBCDIC、Unicode 文字セットのいずれかで、単一文字を返します。この関数を使用することにより、CTRAN 関数と同様、使用するキーボードにはない文字を生成することができます。

Unicode 構成の場合、この関数は次の範囲の値を使用します。

- 1 バイト文字 - 0 (ゼロ) から 255
- 2 バイト文字 - 256 から 65535
- 3 バイト文字 - 65536 から 16777215
- 4 バイト文字 - 16777216 から 4294967295 (主として EBCDIC 用)

特殊文字の表示は、ソフトウェアとハードウェアにより異なります。特殊文字には表示されないものがあります。

構文 10 進数を文字に変換

```
HEXBYT(input, 'outfield')
```

説明

input
整数

文字に変換される 10 進数です。Unicode 以外の環境では、255 より大きい値は、*input* の値を 256 で除算した剰余として扱われます。

outfield
文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 10 進数を文字に変換

HEXBYT 関数は、LAST_INIT_CODE を対応する文字に変換し、結果を LAST_INIT に格納します。

```
COMPUTE LAST_INIT_CODE/I3 = BYTVAL(LAST_NAME, 'I3'); AND
COMPUTE LAST_INIT/A1 = HEXBYT(LAST_INIT_CODE, 'A1');
```

ITONUM - 整数を倍精度小数点数フォーマットに変換

手順/構文

整数を倍精度浮動小数点数フォーマットに変換

ITONUM 関数は、データソースの整数を倍精度フォーマットに変換します。プログラム言語やデータベースには、整数フォーマットを使用するものがあります。ただし、(5 バイト長以上の) 整数はマスターファイルではサポートされません。このため、倍精度フォーマットへの変換が必要になります。

入力フィールド内の最も右側から数えた有効バイト数を指定する必要があります。結果は、8 バイトの倍精度浮動小数点数フィールドです。

構文 整数を倍精度浮動小数点数フォーマットに変換

```
ITONUM(maxbytes, infield, 'outfield')
```

説明

maxbytes

数値

2 進数符号を含めた有効数値データの 8 バイトの infield の最大数です。有効な値には、次のものがあります。

- 5 - 最も左側の 3 バイトを無視します。
- 6 - 最も左側の 2 バイトを無視します。
- 7 - 最も左側のバイトを無視します。

infield

文字

2 進数を含むフィールドです。このフィールドの USAGE および ACTUAL フォーマットは、どちらも A8 である必要があります。

outfield

倍精度浮動小数点数

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。フォーマットは Dn である必要があります。

例 整数を倍精度小数点数フォーマットに変換

ITONUN 関数は、BINARYFLD を倍精度フォーマットに変換します。

```
COMPUTE MYFLD/D14 = ITONUM(6, BINARYFLD, 'D14');
```

ITOPACK - 整数をパック 10 進数フォーマットに変換

手順/構文

整数をパック 10 進数フォーマットに変換

ITOPAK 関数は、データソースの整数をパック 10 進数フォーマットに変換します。プログラム言語やデータベースには、倍長整数フォーマットを使用するものがあります。5 バイト長以上の整数はマスターファイルではサポートされていません。このため、パック 10 進数フォーマットへの変換が必要になります。

入力フィールド内の最も右側から数えた有効バイト数を指定する必要があります。結果は、有効桁数が 15 桁以内の 8 バイトのパック 10 進数フィールド (例、P15 または P16.2) です。

制限事項：「PIC 9(15) COMP」として定義したフィールド、またはこれに準ずるフィールド (有効桁数 15 桁) において、変換可能な最大数は、167,744,242,712,576 です。

構文 整数をパック 10 進数フォーマットに変換

```
ITOPACK(maxbytes, infield, 'outfield')
```

説明

maxbytes

数値

2 進数符号を含めた有効数値データの 8 バイトの `infield` の最大数です。

有効な値には、次のものがあります。

5 - 最も左側の 3 バイトを無視します (11 桁以内の有効な位置)。

6 - 最も左側の 2 バイトを無視します (14 桁以内の有効な位置)。

7 - 最も左側のバイトを無視します (15 桁以内の有効な位置)。

`infield`

文字

2 進数を含むフィールドです。このフィールドの USAGE および ACTUAL フォーマットは、どちらも A8 である必要があります。

`outfield`

パック 10 進数

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。フォーマットは `Pn` または `Pn.d`。

例 整数をパック 10 進数フォーマットに変換

ITOPACK 関数は、BINARYFLD をパック 10 進数フォーマットに変換します。

```
COMPUTE PACKFLD/P14.4 = ITOPACK(6, BINARYFLD, 'P14.4');
```

ITOZ - 数値をゾーン 10 進数フォーマットに変換

手順/構文

ゾーン 10 進数フォーマットに変換

ITOZ 関数は、数値フォーマットの数をゾーン 10 進数フォーマットに変換します。リクエストはゾーン 10 進数を処理することはできませんが、ゾーン 10 進数フィールドを抽出ファイルに書き込み、外部プログラムで使用することができます。

構文 ゾーン 10 進数フォーマットに変換

```
ITOZ(outlength, number, 'outfield')
```

説明

`outlength`

整数

number のバイト数です。最大バイト数は 15 です。末尾バイトには、符号が含まれません。

number
数値

変換される数値です。数値を含むフィールドを指定することもできます。数値は変換前に切り捨てられて整数になります。

outfield
文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 **数値をゾーン 10 進数フォーマットに変換**

ITOEZ 関数は、CURR_SAL フィールドをゾーン 10 進数フォーマットに変換します。

```
COMPUTE ZONE_SAL/A8 = ITOEZ(8, CURR_SAL, 'A8');
```

PCKOUT - 指定した長さでパック 10 進数を書き込み

手順/構文

指定した長さでパック 10 進数を書き込み

PCKOUT 関数は、抽出ファイルに指定した長さでパック 10 進数を書き込みます。リクエストが抽出ファイルにパック 10 進数を保存する際、フォーマットの指定に関わらず、通常 8 バイトまたは 16 バイトのフィールドとして書き込みます。PCKOUT を使用することで、フィールドを 1 から 16 バイトの指定した長さに変更することができます。

構文 **指定した長さでパック 10 進数を書き込み**

```
PCKOUT(infield, outlength, 'outfield')
```

説明

infield
数値

数値を含むフィールドです。このフィールドには、パック 10 進数、整数、浮動小数点数、倍精度フォーマットが使用できます。フィールドが整数フォーマットではない場合、端数処理されて、値は最も近い整数になります。

outlength

数値

outfield の長さ (1 から 16 バイト) です。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。フィールドはパック 10 進数を含んでいますが、この関数はフィールドを文字として返します。

例 指定した長さでパック 10 進数を書き込み

PCKOUT 関数は、CURR_SAL フィールドを 5 バイトのパック 10 進数フィールドに変換し、結果を SHORT_SAL に格納します。

```
COMPUTE SHORT_SAL/A5 = PCKOUT(CURR_SAL, 5, 'A5');
```

数値関数

トピックス

ABS - 絶対値を計算

BAR - 棒グラフを作成

CHKPCK - パック 10 進数フィールドを検査

DMOD、FMOD、IMOD - 除算の剰余を計算

EXP - 「e」を N でべき乗

INT - 整数を検索

LOG - 自然対数を計算

MAX および MIN - 最大値または最小値を検索

PRDNOR および PRDUNI - 再生可能な乱数を生成

RDNORM および RDUNIF - 乱数を生成

SQRT - 平方根を計算

数値関数は、数値定数と数値フィールドの計算を実行します。

ABS - 絶対値を計算

手順/構文

絶対値を計算

ABS 関数は、数値の絶対値を返します。

構文

絶対値を計算

`ABS(argument)`

説明

argument

数値

最大値または最小値を返す値、値を含むフィールド名、あるいは値を返す式です。式を指定する場合は、評価の順序を正しくするため、必要に応じて括弧を使用します。

例

絶対値を計算

1 つ目の COMPUTE コマンドは、DIFF フィールドを作成します。次に、ABS 関数は DIFF の絶対値を計算します。

```
COMPUTE DIFF/I5 = DELIVER_AMT - UNIT_SOLD; AND  
COMPUTE ABS_DIFF/I5 = ABS(DIFF);
```

BAR - 棒グラフを作成

手順/構文

棒グラフを作成

BAR 関数は、横棒グラフを作成します。棒には、繰り返し文字が使用されます。必要に応じて、棒グラフを明確にするために目盛りを作成することができます。これには、棒を含む列タイトルを目盛りで置き換えます。

構文

棒グラフを作成

```
BAR(barlength, infield, maxvalue, 'char', 'outfield')
```

説明

barlength

数値

棒の長さの最大値をバイト数で指定します。この値が0(ゼロ)以下の場合、関数は棒グラフを返しません。

infield

数値

棒グラフとして描くデータフィールドです。

maxvalue

数値

棒グラフの最大値です。この値は、*infield*に格納された最大値より大きくなければなりません。*infield*の値が*maxvalue*の値よりも大きい場合、関数は*maxvalue*を使用し、最大長の棒グラフを返します。

char

文字

棒グラフを作成する繰り返し文字です。文字列は一重引用符(')で囲みます。複数の文字を指定した場合、先頭の文字のみが使用されます。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符(')で囲みます。出力フィールドには、*barlength*に定義された最大値の長さを持つ棒グラフの表示に十分な長さが必要です。

例

棒グラフを作成

BAR 関数は、CURR_SAL フィールドの棒グラフを作成し、出力結果を SAL_BAR に格納します。作成された棒の長さは 30 バイト以下で、表す値は 30,000 以下である必要があります。

```
COMPUTE SAL_BAR/A30 = BAR(30, CURR_SAL, 30000, '=', SAL_BAR);
```

CHKPCK - パック 10 進数フィールドを検査

手順/構文

パック 10 進数フィールドを検査

CHKPCK 関数は、プラットフォームで使用可能な場合、パック 10 進数フィールドとして記述されるフィールド内のデータを検査します。この関数は、リクエストがフィールドを読み取る際に、有効なパック 10 進数が含まれていることを期待して実際に含まれていない場合に、データ例外が発生することを防止します。

CHKPCK の使用方法は、次のとおりです。

1. マスターファイル (USAGE および ACTUAL 属性) では、フィールドがパック 10 進数ではなく、文字として定義します。これにより、フィールドデータは変更されずにパック 10 進数のままになりますが、リクエストによるデータの読み取り時にデータ例外の発生を防止することができます。
2. CHPCK を呼び出してフィールドを検査します。この関数は、パック 10 進数として定義したフィールドに出力結果を返します。検査する値が有効なパック 10 進数の場合、関数は値を返します。パック 10 進数ではない場合は、エラーコードを返します。

構文

パック 10 進数フィールドを検査

```
CHKPCK(inlength, infield, error, 'outfield')
```

説明

inlength

数値

引数の長さです。1 から 16 バイトの値を指定します。

infield

文字

入力フィールド名です。フィールドは、パック 10 進数ではなく、文字として記述されます。

error

数値

値がパック 10 進数ではない場合に関数が返すエラーコードです。データの範囲外のエラーコードを選択します。エラーコードは、整数に切り捨てられた後、パック 10 進数フォーマットに変換されます。ただし、出力フィールドのフォーマットによっては、レポートに小数点付きで表示される場合があります。

outfield

パック 10 進数

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例

パック 10 進数を検査

CHKPCK 関数は、PACK_SAL フィールドの値を検査し、結果を GOOD_PACK フィールドに格納します。フォーマットがパック 10 進数以外の値は、エラーコード -999 を返します。パック 10 進数フォーマットの値は正しく表示されます。

```
COMPUTE GOOD_PACK/P8CM = CHKPCK(8, PACK_SAL, -999, GOOD_PACK);
```

DMOD、FMOD、IMOD - 除算の剰余を計算

手順/構文

除算の剰余を計算

MOD 関数は、除算の剰余を計算します。各関数は、異なるフォーマットで剰余を返します。

関数は次の公式を使用します。

$$\text{remainder} = \text{dividend} - \text{INT}(\text{dividend}/\text{divisor}) * \text{divisor}$$

- DMOD - 剰余を単精度浮動小数点数で返します。
- FMOD - 剰余を倍精度浮動小数点数で返します。
- IMOD - 剰余を整数で返します。

INT 関数についての詳細は、353 ページの「[INT - 整数を検索](#)」を参照してください。

構文 除算の剰余を計算

```
function(dividend, divisor, 'outfield')
```

説明

function

次のいずれかです。

DMOD - 剰余を単精度浮動小数点数で返します。

FMOD - 剰余を倍精度浮動小数点数で返します。

IMOD - 剰余を整数で返します。

dividend

数値

被除数です。

divisor

数値

除数です。

outfield

数値

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。フォーマットは、特定の関数により返された結果により決定されます。

例 除算の剰余を計算

IMOD 関数は、ACCTNUMBER を 1000 で除算し、その剰余を LAST3_ACCT に返します。

```
COMPUTE LAST3_ACCT/I3L = IMOD(ACCTNUMBER, 1000, LAST3_ACCT);
```

EXP - 「e」を N でべき乗

手順/構文

「e」を N でべき乗

EXP 関数は、値「e」(およそ 2.72) を指数でべき乗します。この関数は、引数の対数を返す LOG 関数の逆です。

EXP は、無限級数の項の和を計算します。項が合計に加算する値が 0.000001 パーセントよりも小さくなったところで、関数は計算を終了し、結果を倍精度小数点数で返します。

構文 「e」をNでべき乗

```
EXP(power, 'outfield')
```

説明

power

数値

「e」をべき乗する指数です。

outfield

倍精度または単精度浮動小数点数

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 EXP - 「e」をNでべき乗

EXP 関数は、「e」を 2 乗します。

```
COMPUTE E2/D12.2 = EXP(2, 'D12.2');
```

INT - 整数を検索

手順/構文

整数を検索

INT 関数は、数値の整数構成要素を返します。

構文 整数を検索

```
INT(argument)
```

説明

argument

数値

整数構成要素が返される値です。値を含むフィールド名または値を返す式を指定することもできます。式を指定する場合は、評価の順序を正しくするため、必要に応じて括弧を使用します。

例 **整数を検索**

INT は DED_AMT フィールドの最大整数を検索し、結果を INT_DED_AMT に格納します。

```
COMPUTE INT_DED_AMT/I9 = INT(DED_AMT);
```

LOG - 自然対数を計算

手順/構文

自然対数を計算

LOG 関数は、数値の自然対数を返します。

構文 **自然対数を計算**

`LOG(argument)`

説明

argument

数値

自然対数が計算される値です。値を含むフィールド名、または値を返す式を指定することもできます。式を指定する場合は、評価の順序を正しくするため、必要に応じて括弧を使用します。 *argument* の値が 0 (ゼロ) 以下の場合、LOG は 0 (ゼロ) を返しません。

例 **自然対数を計算**

LOG 関数は、CURR_SAL フィールドの対数を計算します。

```
COMPUTE LOG_CURR_SAL/D12.2 = LOG(CURR_SAL);
```

MAX および MIN - 最大値または最小値を検索

手順/構文

最大値または最小値を検索

MAX および MIN 関数は、値リストからそれぞれ最大値と最小値を返します。

構文 最大値または最小値を検索

```
{MAX|MIN}(argument1, argument2, ...)
```

説明

MAX

最大値を返します。

MIN

最小値を返します。

argument1, argument2

数値

最大値または最小値を返す値、値を含むフィールド名、あるいは値を返す式です。式を指定する場合は、評価の順序を正しくするため、必要に応じて括弧を使用します。

例 最小値を抽出

MIN 関数は、ED_HRS フィールドの値と定数 30 のうちの、小さい方の値を返します。

```
COMPUTE MIN_EDHRS_30/D12.2 = MIN(ED_HRS, 30);
```

PRDNOR および PRDUNI - 再生可能な乱数を生成

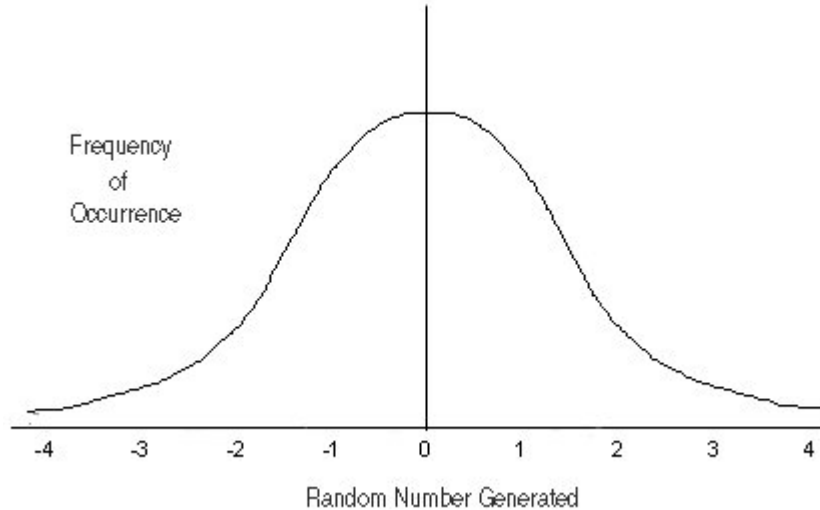
手順/構文

再生可能な乱数を生成

PRDNOR 関数および PRDUNI 関数は、再生可能な乱数を生成します。

- PRDNOR は、相加平均が 0 (ゼロ) で標準偏差が 1 の、正規分布を持つ再生可能乱数を倍精度浮動小数点数で生成します。PRDNOR 関数が数値を多数生成した場合、これらの数値は次のようになります。

- これらの数値は、下図のように釣鐘曲線で表されます。釣鐘曲線の最高点は 0 (ゼロ) です。これは、0 (ゼロ) に近い数値が 0 (ゼロ) から遠い数値よりも多いことを示します。



- 数値の平均値は 0 (ゼロ) に近似します。
- 数値の大きさに制限はありませんが、ほとんどの数値は 3 から -3 までの値になります。
- PRDUNI は、0 (ゼロ) と 1 の間に均等に分布する再生可能な乱数を生成します。0 (ゼロ) と 1 の間の数値は、すべて等しい確率で生成されます。

構文

再生可能な乱数を生成

```
{PRDNOR|PRDUNI}(seed, 'outfield')
```

説明

PRDNOR

相加平均が 0 (ゼロ) で標準偏差が 1 の、正規分布を持つ再生可能な乱数を倍精度浮動小数点数で生成します。

PRDUNI

0 (ゼロ) と 1 の間に均等に分布する再生可能な乱数を倍精度浮動小数点数で生成します。

seed

数値

9 バイト以内のシード、またはシードを含むフィールドです。シードは整数に切り捨てられます。

outfield

倍精度浮動小数点数

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例

再生可能な乱数を生成

PRDNOR 関数は、乱数を割り当て、RAND に格納します。シードは 40 です。異なる数値セットを生成するには、シードを変更します。

```
COMPUTE RAND/D12.2 = PRDNOR(40, 'D12.2');
```

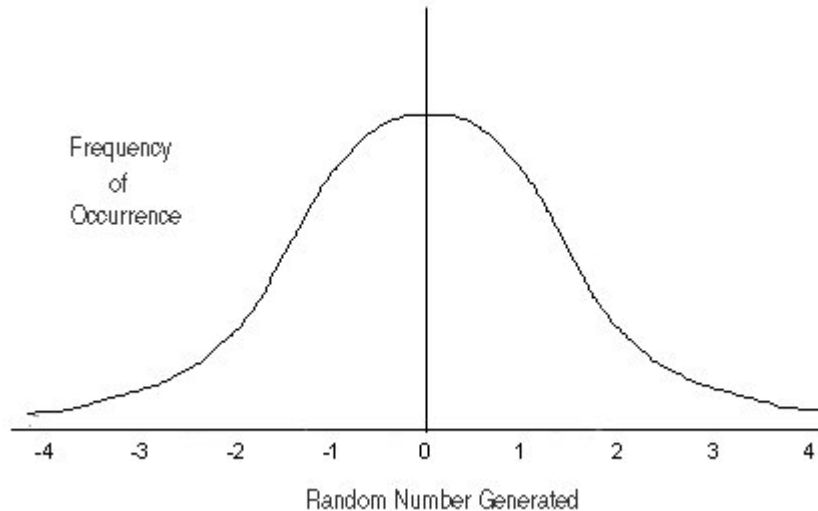
RDNORM および RDUNIF - 乱数を生成

手順/構文

乱数を生成

RDNORM および RDUNIF 関数は、乱数を生成します。

- RDNORM は、相加平均が 0 (ゼロ) で標準偏差が 1 の、正規分布を持つ乱数を倍精度浮動小数点数で生成します。RDNORM 関数は、数値 (1 から 32768) を多数生成します。
- これらの数値は、下図のように釣鐘曲線で表されます。釣鐘曲線の最高点は 0 (ゼロ) です。これは、0 (ゼロ) に近い数値が 0 (ゼロ) から遠い数値よりも多いことを示します。



- 数値の平均値は 0 (ゼロ) に近似します。
- 数値の大きさに制限はありませんが、ほとんどの数値は 3 から -3 までの値になります。
- RDUNIF は、0 と 1 の間に均等に分布する乱数を生成します。0 と 1 の間の数値は、すべて等しい確率で生成されます。

構文 **乱数を生成**

```
{RDNORM|RDUNIF}('outfield')
```

説明**RDNORM**

相加平均が 0 (ゼロ) で標準偏差が 1 の、正規分布を持つ乱数を倍精度浮動小数点数で生成します。

RDUNIF

0 と 1 の間に均等に分布する乱数を倍精度浮動小数点数で生成します。

outfield

倍精度浮動小数点数

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 **乱数を生成**

RDNORM 関数は、乱数を割り当て、RAND に格納します。

```
COMPUTE RAND/D12.2 = RDNORM('D12.2');
```

SQRT - 平方根を計算**手順/構文**

平方根を計算

SQRT 関数は、数値の平方根を計算します。

構文 **平方根を計算**

```
SQRT(argument)
```

説明*argument*

数値

平方根が計算される値です。値を含むフィールド名、または値を返す式を指定することもできます。式を指定する場合は、評価の順序を正しくするため、必要に応じて括弧を使用します。負の数値を指定すると、結果は 0 (ゼロ) になります。

例 平方根を計算

SQRT 関数は、LISTPR の平方根を計算します。

```
COMPUTE SQRT_LISTPR/D12.2 = SQRT(LISTPR);
```

システム関数

トピックス

FGETENV - 環境変数値を取得

GETUSER - ユーザ ID を取得

HHMMSS - 現在の時間を取得

TODAY - 現在の日付を取得

システム関数は、オペレーティングシステムを呼び出して、オペレーティング環境についての情報を取得します。

FGETENV - 環境変数値を取得

手順/構文

環境変数の値を取得

FGETENV 関数は、オペレーティングシステムの環境変数値を取得し、文字列として返します。

構文 環境変数の値を取得

```
FGETENV(varlength, 'varname', outfieldlen, 'outfield')
```

説明

varlength

整数

環境変数名の長さです。

varname

文字

環境変数名です。

outfieldlen

整数

環境変数の値が格納されているフィールドの長さです。

outfield

文字

出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。

例 言語ロケールを取得

LANG 環境変数を使用して、FGETENV 関数は言語ロケールのオブジェクトロケーションを取得します。

```
COMPUTE LANG_LOCALE/A40 = FGETENV(4, 'LANG', 40, 'A40');
```

GETUSER - ユーザ ID を取得**手順/構文****ユーザ ID を取得**

GETUSER 関数は、接続ユーザの ID を取得します。

構文 ユーザ ID を取得

```
GETUSER('outfield')
```

説明

outfield

文字

A8 以上の出力値のフォーマットです。フォーマットは一重引用符 (') で囲みます。長さは、関数を発行するプラットフォームに依存します。プラットフォームで必要な長さを指定します。十分な長さを準備しないと、出力結果は切り捨てられます。

例 ユーザ ID を取得

GETUSER 関数は、リクエストを実行しているユーザの ID を取得します。

```
COMPUTE USERID/A8 = GETUSER('A8');
```

HHMMSS - 現在の時間を取得

HHMMSS関数は、オペレーティングシステムから現在の時間を取得します。時間は、時、分、秒をピリオド(.)で区切った8バイトの文字列として取得されます。この関数の使用方法についての詳細は、327ページの「[HHMMSS - 現在の時間を取得](#)」を参照してください。

TODAY - 現在の日付を取得

TODAY関数は、システムから現在の日付を取得します。この関数の使用方法についての詳細は、336ページの「[TODAY - 現在の日付を取得](#)」を参照してください。

A Java バッチ実行ユーティリティによる DB2 Web Query レポートの実行

Java バッチ実行ユーティリティ (RUNWEBQRY) により、コマンドラインから DB2 Web Query レポートを実行することができます。これにより、DB2 Web Query にログインすることなくリクエストをバッチキューに送信することができます。ユーティリティは、DB2 Web Query のレポートプロシジャを 1 つ受容し、プロシジャは、Java プログラム経由で実行されます。

トピックス

- Java バッチ実行ユーティリティ要件

Java バッチ実行ユーティリティ要件

手順/構文

Java バッチ実行ユーティリティを呼び出すには

参照

RUNWEBQRY コマンド入力パラメータの取得

以下は、ユーティリティの実行における要件です。

- DB2 Web Query は、Reporting Server および Application Server が稼働中のシステムにインストールし、実行すること。
- プロシジャが DB2 Web Query 環境に存在すること。
- ユーザ ID が DB2 Web Query のライセンスを所有すること。

手順

Java バッチ実行ユーティリティを呼び出すには

このユーティリティをコマンドラインから呼び出すには、次の手順を実行します。

1. 5250 端末エミュレータ経由で IBM i システムにログインします。
2. コマンドラインで、次のコマンドを実行します。

```
RUNWEBQRY
```

3. F4 キーを押します。

下図のような Web Query Java バッチ実行 (Run Java batch) (RUNWEBQRY) 画面が表示されます。

```

Run Web Query Report (RUNWEBQRY)

Type choices, press Enter.

Domain HREF . . . . . 'untitled' Character value
Folder HREF . . . . . Character value
File Name (FEX) . . . . .

-----

Output Format . . . . . WP_ DOC, WP
Output Queue . . . . . QPRINT Character value
Run Report with User ID . . . . . *CURRENT Character value

-----

Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

```

- 画面上で、次の入力パラメータ値を入力します。

ドメインの *HREF* (Domain href)

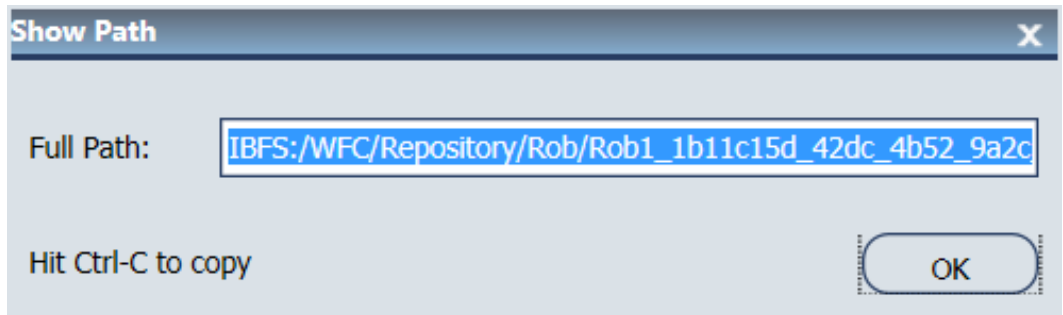
ダッシュボードまたは HREF のドメイン名です。HREF はオブジェクトの内部名です。HREF はダッシュボードメニューの [プロパティ] (Properties) オプションで表示することができます。このメニューはマウスを右クリックすると表示されます。名前が正確に 8 文字の場合は、HREF を使用する必要はありません。HREF は、名前が 7 文字以下、または 9 文字以上の場合に使用します。入力パラメータは、レポート名を右クリックし、コンテキストメニューから [プロパティ] (Properties) を選択することで、[プロパティ] (Properties) ページから取得できます。

フォルダの HREF (Folder href)

ダッシュボードまたは HREF のフォルダ名です。HREF はオブジェクトの内部名です。HREF はダッシュボードメニューの [プロパティ] (Properties) オプションで表示することができます。このメニューはマウスを右クリックすると表示されます。名前が正確に 12 文字の場合は、HREF を使用する必要はありません。HREF は、名前が 11 文字以下、または 13 文字以上の場合に使用します。入力パラメータは、レポート名を右クリックし、コンテキストメニューから [プロパティ] (Properties) を選択することで、[プロパティ] (Properties) ページから取得できます。

ファイル名 (File Name (fex))

BI Portal ツリーに表示されるレポートプロシジャ (fex) の名前またはフルパスです。かすのように、この入力名は、レポート名を右クリックし、コンテキストメニューから [パスの表示] (Show Path) を選択することで、[パスの表示] (Show Path) オプションから取得することができます。プロシジャのフルパスおよびレポート名をコピーするには、Ctrl+C を押します。



出力フォーマット (Output format)

レポートのフォーマットです。DOC、WP のいずれかの値を入力します。デフォルト値は WP です。

出力キュー (Output Queue)

IBMi システム上で、レポートの送信先となる出力キュー名です。デフォルト値は QPRINT です。

ユーザ ID で実行 (Run Report with User ID)

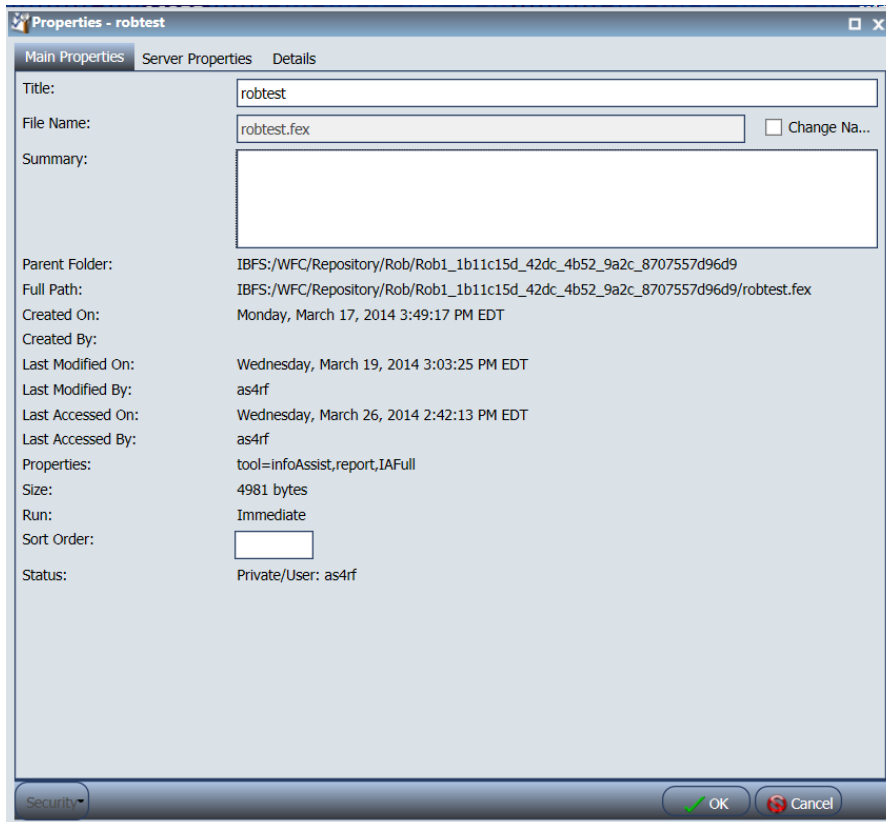
別のユーザ ID で実行することができます。デフォルト値は *CURRENT です。これは、ジョブの実行に現在のユーザ ID を使用することを意味します。

注意：複数のリクエストを実行する場合は、SBMJOBを使用して、ジョブストリームの一部として RUNWEBQRY コマンドを使用することができます。

参照

RUNWEBQRY コマンド入力パラメータの取得

下図は、パラメータ情報を取得するための [プロパティ] (Properties) ページの例を示しています。このページにアクセスするには、レポートフォルダのレポート名の 1 つを右クリックし、コンテキストメニューから [プロパティ] (Properties) を選択します。



下図は、サンプルプロパティページに入力パラメータが入力された状態を示しています。

```
Run Web Query Report (RUNWEBQRY)

Type choices, press Enter.

Domain HREF . . . . . 'untitled' Character value
Folder HREF . . . . . _____ Character value
File Name (FEX) . . . . . > IBFS:/WFC/Repository/Rob/Rob1_1b11c15d_42dc_
4b52_9a2c_8707557d96d9/robtest.fex_
_____
_____
_____
_____
Output Format . . . . . WP_ DOC, WP
Output Queue . . . . . QPRINT Character value
Run Report with User ID . . . . . *CURRENT Character value

Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys
```


B フィールドの記述

フィールドは、データソースに格納されたデータの有意な最小単位ですが、フィールドごとにさまざまな特性を表すことができます。これらの特性を記述するには、マスターファイルの属性を使用します。

トピックス

- ❑ フィールドの特性
- ❑ フィールド名 - FIELDNAME
- ❑ フィールドエイリアス - ALIAS
- ❑ 表示データタイプ - USAGE
- ❑ 格納データタイプ - ACTUAL
- ❑ ミッシング値 (Null 値) - MISSING
- ❑ FML 階層の記述
- ❑ データの確認 - ACCEPT
- ❑ 代替レポートフィールドタイトル - TITLE
- ❑ フィールドの説明 - DESCRIPTION
- ❑ 多言語メタデータ
- ❑ 一時項目 (DEFINE) の記述 - DEFINE
- ❑ 一時項目 (COMPUTE) の記述 - COMPUTE
- ❑ フィルタの記述 - FILTER
- ❑ ソートオブジェクトの記述 - SORTOBJ
- ❑ マスターファイル DEFINE による日付システム変数の使用
- ❑ 変数を使用したマスターファイルおよびアクセスファイルのパラメータ化
- ❑ 文字日付の 日付への変換

フィールドの特性

マスターファイルでは、次のフィールド特性を記述します。

- FIELDNAME 属性で識別するフィールド名。
- フィールドの別名。これは、ネイティブデータ管理システムで定義された元の名前、特定のデータソースタイプで独自に選択した別名、特別な場合としてフィールドを解釈するために事前に定義された値のいずれかです。これらの別名をリクエストで使用することができます。この別名は ALIAS 属性で定義します。
- フィールドがデータを格納して表示する方法。これらの特性は、ACTUAL、USAGE、MISSING 属性で指定します。

ACTUAL 属性は、データソースにデータを実際に格納する際のデータタイプおよび長さを指定します。たとえば、長さが 15 バイトの文字フィールドとしてフィールドを定義します。FOCUS データソースでは、ACTUAL 属性は使用されず、データフォーマットの記述には、USAGE 属性を使用します。は、データストレージを処理します。

USAGE 属性は、フィールドをレポートに表示する際のフィールドのフォーマットを指定します。この属性のエイリアスは FORMAT です。また、日付フォーマット、通貨記号表示 (浮動)、ゼロサプレス (ゼロを非表示) などの編集オプションを指定することもできます。

MISSING 属性は、FOCUS データソースや多くのリレーショナルデータソースなどの Null データをサポートするデータソースで、Null 値のフィールドへの書き込みおよびフィールドからの読み取りを有効にします。

- 一時項目を作成するオプション。一時項目は、データソースに格納されている値ではなく、データソースに存在する情報から算出された値を持つフィールドです。一時項目 (DEFINE) は DEFINE 属性で指定します。
- オプションとして使用する開発者向けのフィールドの説明。この説明は DESCRIPTION 属性で指定します。
- 受容可能なフィールドのデータ入力値。この特性は ACCEPT 属性で指定します。
- フィールドの代替レポートフィールドタイトル。この特性は TITLE 属性で指定します。
- フィールドに格納された下 2 桁の西暦年に上 2 桁の世紀の値を割り当てる 100 年。この時間枠は DEFCENT および YRTHRESH の 2 つの属性で定義します。

フィールド名 - FIELDNAME

トピックス

修飾フィールド名の使用

重複フィールド名の使用

修飾フィールド名評価時の規則

手順/構文

フィールド名の識別

参照

FIELDNAME 属性使用時の注意

フィールド名の制限事項

FIELDNAME 属性を使用してフィールドを識別します。この属性は、マスターファイルのフィールド宣言で最初に指定する属性です。フィールドには、ネイティブデータソースでの名前に関係なく、任意の名前を割り当てることができます。同様に、FOCUS データソースでは、新しいデータソースのフィールドに任意の名前を割り当てることができます。

レポートを生成すると、デフォルト設定でレポートのフィールドタイトルにはフィールド名が表示されるため、レポートを閲覧するユーザに分かりやすい名前を付けておくと役立ちます。別の方法として、レポートリクエストに AS 句を使用してレポートに表示する別のフィールドタイトルを指定することもできます。また、マスターファイルに TITLE 属性を使用して、すべてのレポートに別のフィールドタイトルを指定することもできます。詳細は、440 ページの「[代替レポートフィールドタイトル-TITLE](#)」を参照してください。

構文 **フィールド名の識別**

`FIELD[NAME] = field_name`

説明

field_name

このフィールドに適用する名前です。名前の最大長は 512 バイトです。名前が 12 バイトを超える場合、いくつかの制限事項が適用されます。詳細は、373 ページの「[フィールド名の制限事項](#)」を参照してください。名前には 1 つ以上の文字を含める必要があります。それ以外は、文字、数字、アンダースコア (_) を任意に組み合わせて使用することができます。異なるオペレーティングシステム環境や、数式を計算する際に問題を起こす可能性があるため、これ以外の文字は使用しないことをお勧めします。

また、タイプを表す Cn、En、Xn は、レポートフィールド、HOLD ファイルフィールド、その他の特別なオブジェクトを参照する際に使用されるため、フィールド名には使用しないことをお勧めします (ここで、n は任意の 1 桁または 2 桁の連続番号)。

フィールドのレポートフィールドタイトルに特殊文字を使用する必要がある場合は、マスターファイルの TITLE 属性でタイトルを指定する方法もあります。詳細は、440 ページの「[代替レポートフィールドタイトル - TITLE](#)」を参照してください。

参照 **FIELDNAME 属性使用時の注意**

FIELDNAME 属性の使用時には次の規則が適用されます。

- **エイリアス** FIELDNAME 属性のエイリアスは FIELD です。
- **変更** FOCUS データソースでは、フィールドにインデックスが作成されている場合 (INDEX=I 属性)、データソースを再構築しない限り、フィールド名を変更することはできません。これ以外の場合は名前を変更することができます。

参照

フィールド名の制限事項

12 バイトを超えるフィールド名およびエイリアスには、次の制限事項が適用されます。

- クロスリファレンスファイルが FOCUS データベースの場合、12 バイトを超えるフィールド名を使用して、JOIN コマンドにクロスリファレンスフィールドを指定することはできません。
- FOCUS データソースのインデックスフィールドおよびテキストフィールドには 12 バイトを超える長さのフィールド名を含めることはできません。XFOCUS マスターファイルのテキストフィールドおよびインデックスフィールドには 12 バイトの制限は適用されません。長い名前のエイリアスは両タイプのデータソースでサポートされます。
- 代替ファイルビューで指定するフィールド名として、修飾名を使用することはできません。
- CHECK FILE コマンドの PICTURE および HOLD オプションは、表示された図または HOLD ファイルに長い名前の先頭の 11 バイトを表示します。12 バイト目の位置にキャレット (>) が表示された場合、これは実際の名前が表示された部分より長いことを意味します。
- ?FF、? HOLD、? DEFINE

これらのコマンドは、名前の 31 バイト分に加えて、長いフィールド名を示すキャレット (>) を 32 バイト目に表示します。

修飾フィールド名の使用

手順/構文

リクエストでの修飾フィールド名の指定

修飾文字の変更

リクエストでは参照されたすべてのフィールド名をファイル名またはセグメント名で修飾することができます。この方法は、重複したフィールド名がマスターファイルのセグメントや結合されたデータソースに存在する場合に役立ちます。

FOCUSマスターファイルでは、テキストフィールドおよびインデックスフィールドのフィールド名は 12 バイトに制限されます。XFOCUS マスターファイルでは、テキストフィールドおよびインデックスフィールド名には 12 バイトの制限は適用されません。ただし、テキストフィールドおよびインデックスフィールドのエイリアスの最大長は 512 バイトです。TITLE 属性または AS 句が使用されていない場合は、最大で 512 バイトのフィールド名が TABLE レポートのフィールドタイトルに表示されます。

SET FIELDNAME コマンドのデフォルト値である SET FIELDNAME=NEW により、長いフィールド名および修飾フィールド名は有効になっています。構文についての詳細は、

構文 **リクエストでの修飾フィールド名の指定**

```
[filename.][segname.]fieldname
```

説明

filename

マスターファイルまたはタグの名前です。タグ名は JOIN および COMBINE コマンドに使用します。

segname

フィールドが存在するセグメントの名前です。

fieldname

フィールド名です。

例 **フィールド修飾名の指定**

EMPLOYEE データソースの EMPINFO セグメントに存在する EMP_ID フィールドの完全修飾名は次のとおりです。

```
EMPLOYEE.EMPINFO.EMP_ID
```

構文 **修飾文字の変更**

```
SET QUALCHAR = qualcharacter
```

デフォルトの修飾文字はピリオド (.) です。

重複フィールド名の使用

1つのフィールド名またはエイリアスで複数のフィールドを参照できる場合、重複したフィールド名と見なされます。次のような場合にフィールドの重複が発生します。

- 1つの名前がマスターファイル内で複数回出現する場合。
- 複数のマスターファイルで定義された JOIN、または再帰的 JOIN。
- 接頭語を指定せずに COMBINE を発行した場合。

同一セグメント内で重複フィールドを使用することはできません (同一のフィールド名およびエイリアスを持つフィールド)。2 回目に出現したフィールドにはアクセスできないため、CHECK および CREATE FILE を発行すると次のメッセージが生成されます。

```
(FOC1829) WARNING. FIELDNAME IS NOT UNIQUE WITHIN A SEGMENT: fieldname
```

マスターファイルでは、異なるセグメントに重複フィールド名を含めることができます。リクエストでそのフィールドを取得するには、フィールド名をセグメント名で修飾する必要があります。マスターファイルに複数回出現するフィールドをリクエストで修飾しない場合は、マスターファイルで最初に見つかったフィールドが取得されます。

注意：マスターファイルに実フィールドまたは一時項目 (DEFINE) の重複フィールド名が含まれる場合、フィールドの取得時には次のロジックが適用されます。

- 一時項目 (DEFINE) のみが重複している場合は、最後の一時項目 (DEFINE) が取得されます。
- 実フィールドのみが重複している場合は、最初の実フィールドが取得されます。
- マスターファイルに同一名の実フィールドと 1つまたは複数の一時項目 (DEFINE) が存在する場合は、最後の一時項目 (DEFINE) が取得されます。
- マスターファイルの外部で定義したフィールドの名前がマスターファイルの一時項目または実フィールドの名前と同一の場合は、マスターファイルの外部で定義した最後の一時項目が取得されます。

レポートには修飾名をフィールドタイトルとして含めることができます。レポートに重複フィールド名の修飾フィールドタイトルを表示するかどうかを指定するには SET QUALTITLES コマンドを使用します。このコマンドについての詳細は、SET QUALTITLES=ON に設定すると、リクエストで修飾名を指定しなくても、重複フィールド名の修飾フィールドタイトルが表示されます。デフォルト値は OFF で、修飾フィールドタイトルの表示は無効になっています。

修飾フィールド名評価時の規則

修飾フィールド名の評価には次の規則が適用されます。

- 最大の修飾フィールド名は filename.segname.fieldname です。以下はその例です。

```
TABLE FILE EMPLOYEE
PRINT EMPLOYEE.EMPINFO.EMP_ID
END
```

ここでは、完全修飾フィールドとして EMP_ID が指定されています。ファイル名の EMPLOYEE およびセグメント名の EMPINFO はフィールド修飾子です。

重複した修飾子名を使用することもできます。以下はその例です。

```
FILENAME=CAR, SUFFIX=FOC
  SEGNAME=ORIGIN, SEGTYPE=S1
    FIELDNAME=COUNTRY, COUNTRY, A10, $
  SEGNAME=COMP, SEGTYPE=S1, PARENT=ORIGIN
    FIELDNAME=CAR, CARS, A16, $
    .
    .
    .
TABLE FILE CAR
PRINT CAR.COMP.CAR
END
```


このリクエストは、「CAR」というエイリアス付きでフィールドを出力します。この場合、ファイル名とフィールド名の両方が CAR です。

フィールド名は、そのファイル名、セグメント名のいずれか一方の修飾子で修飾することができます。以下はその例です。

```
FILENAME=CAR, SUFFIX=FOC
  SEGNAME=ORIGIN, SEGTYPE=S1
    FIELDNAME=COUNTRY, COUNTRY, A10, $
  SEGNAME=COMP, SEGTYPE=S1, PARENT=ORIGIN
    FIELDNAME=CAR, CARS, A16, $
    .
    .
    .
TABLE FILE CAR
PRINT COMP.CAR AND CAR.CAR
END
```

このリクエストは、「CAR」というエイリアス付きでフィールドを 2 回出力します。

修飾子が 1 つの場合は、セグメント名がファイル名より優先されます。そのため、ファイル名とセグメント名が同一の場合は、セグメント名で修飾された方のフィールドが取得されます。

- フィールド名の先頭文字が演算接頭語と同一の場合、リクエストの参照先がそのフィールド名全体を指しているのか、接頭語演算子が適用されたフィールド名を指しているのかを判断できない場合があります。この場合は、演算接頭語をフィールドに適用して計算した値ではなく、全体をフィールド名とした値が取得されます。次の例では、修飾されていない「CNT.COUNTRY」というフィールド名および「COUNTRY」というフィールド名が指定されています。

```
FILENAME=CAR, SUFFIX=FOC
  SEGNAME=ORIGIN, SEGTYPE=S1
    FIELDNAME=CNT.COUNTRY, ACNTRY, A10, $
    FIELDNAME=COUNTRY, BCNTRY, A10, $
TABLE FILE CAR
SUM CNT.COUNTRY
END
```

このリクエストの「CNT.COUNTRY」という文字列は、演算接頭語の CNT. が適用された「COUNTRY」というフィールドを指しているのではなく、「CNT.COUNTRY」というフィールドを指しているものと解釈されます。そのため、このリクエストではエイリアスが ACNTRY のフィールドの方が集計されます。フィールド名の CNT.COUNTRY にはピリオド(.)が含まれていますが、これは修飾されていないフィールド名です。これは修飾名でもフィールド名に付ける演算接頭語でもありません。両者ともマスターファイルで使用することはできません。このリクエストでは、エイリアスが BCNTRY のフィールドのインスタンス数は集計されません。

- マスターファイルで定義したファイル名、セグメント名のいずれかが演算接頭語と同一の場合でも、リクエストはフィールドに演算接頭語を適用して値を計算するのではなく、セグメント内のフィールド自体の値を取得します。

以下はその例です。

```
FILENAME=CAR, SUFFIX=FOC
  SEGNAME=ORIGIN, SEGTYPE=S1
    FIELDNAME=COUNTRY, COUNTRY, A10, $
  SEGNAME=PCT, SEGTYPE=S1, PARENT=ORIGIN
    FIELDNAME=CAR, CARS, I2, $
TABLE FILE CAR
SUM PCT.CAR PCT.PCT.CAR
BY COUNTRY
END
```

このリクエストは、最初にエイリアスが CAR のフィールドを集計し、次に COUNTRY に対する CAR のパーセント値を計算します。

- 修飾フィールド名が2つの修飾レベルのどちらかでも成り立つ場合は、上位にある修飾フィールド名が優先されます。

次の例では、修飾されていないフィールド名 (ORIGIN セグメントの ORIGIN.COUNTRY フィールド) とセグメント名で修飾されているフィールド名 (ORIGIN セグメントの COUNTRY フィールド) のどちらも成り立ちます。ここでは、セグメント名が修飾されているフィールドが取得されます。

```
FILENAME=CAR, SUFFIX=FOC
  SEGNAME=ORIGIN, SEGTYPE=S1
    FIELDNAME=ORIGIN.COUNTRY, OCNTRY, A10, $
    FIELDNAME=COUNTRY, CNTRY, A10, $
TABLE FILE CAR
PRINT ORIGIN.COUNTRY
END
```

このリクエストは、エイリアスが CNTRY のフィールドを表示します。エイリアスが OCNTRY のフィールドを取得するには、次のようにフィールド名の ORIGIN.COUNTRY をそのセグメント名の ORIGIN で修飾する必要があります。

```
PRINT ORIGIN.ORIGIN.COUNTRY
```

- 修飾フィールド名が修飾レベルの等しい2つのフィールド名のどちらでも成り立つ場合は、基本フィールド名の長さが短い方のフィールド名が優先されます。以下はその例です。

```
FILENAME=CAR, SUFFIX=FOC
  SEGNAME=CAR, SEGTYPE=S1
    FIELDNAME=CAR.CAR, CAR1, A10, $
  SEGNAME=CAR.CAR, SEGTYPE=S1, PARENT=CAR
    FIELDNAME=CAR, CAR2, A10, $
TABLE FILE CAR
PRINT CAR.CAR.CAR
END
```

この例では、「CAR.CAR.CAR」という文字列が、CAR セグメントの CAR.CAR というフィールドを指しているのか、CAR.CAR セグメントの CAR というフィールドを指しているのかを判断することができません。この場合、両者とも「CAR.CAR」という名前はピリオド(.)を含む修飾されていない名前で、修飾名ではありません。修飾名をマスターファイルで使用することはできません。

いずれの場合も修飾フィールド名は同一で、修飾レベルでの明確な差はありません。

この場合は、エイリアスが CAR2 のフィールドの基本フィールド名の方が短いため、CAR2が表示されます。この例は、2つの修飾レベルで比較した前回の例とは異なります。CAR1 フィールドを取得するには、そのエイリアスを指定する必要があります。

フィールドエイリアス - ALIAS

トピックス

フィールドエイリアスの実装

すべてのフィールドに別名 (エイリアス) を割り当てることができます。フィールドのエイリアスには、そのネイティブデータソースで定義された元の名前、ユーザが独自に選択した任意の名前、特別な場合として定義済みの値のいずれかを割り当てることができます。エイリアスの割り当て方法は、データソースのタイプおよび特別な場合としてデータソースでのフィールドの役割により異なります。エイリアスを割り当てると、通常のフィールド名の別名としてエイリアスをリクエストに使用することができます。エイリアスを割り当てするには、ALIAS 属性を使用します。

例

フィールドエイリアスの使用

EMPLOYEE データソースでは、FIELDNAME 属性を使用してフィールドに「CURR_SAL」という名前を割り当て、ALIAS 属性を使用してこのフィールドに「CSAL」というエイリアスを割り当てています。

```
FIELDNAME = CURR_SAL, ALIAS = CSAL, USAGE = D12.2M, $
```

リクエストでは、有効な名前として両者を同等に使用することができます。次の 2 つの TABLE リクエストは、両者が同一のフィールドを参照し、機能的に同等で、同一の結果を生成することを示しています。

```
TABLE FILE EMPLOYEE  
PRINT CURR_SAL BY EMP_ID  
END
```

```
TABLE FILE EMPLOYEE  
PRINT CSAL BY EMP_ID  
END
```

注意：抽出ファイル (HOLD および PCHOLD) のフィールドの識別には、エイリアスではなく、フィールド名が使用されます。

フィールドエイリアスの実装

ALIAS 属性に値を割り当てる場合、特記されていない限り、FIELDNAME 属性と同一の規則に従って名前を付ける必要があります。次の方法で各データソースタイプの ALIAS 属性に値を割り当てします。

- **リレーショナルデータソース** ALIAS 属性には、リレーショナルテーブルで定義された元のフィールド名と同一の名前を指定します。
- **シーケンシャルデータソース** ALIAS 属性は、リクエストでフィールドの識別に使用するシノニムまたは別名を記述します。エイリアスには、任意の名前を割り当てることができます。一般的に、フィールドの実名よりも短い名前を選択します。たとえば、フィールド名が LAST_NAME の場合、エイリアスは LN にします。ALIAS 属性は、マスターファイルで必須ですが、ブランクの値を指定することもできます。

なお、連続した繰り返しフィールドには ALIAS の値を ORDER に指定してこの属性を別の方法で使用します。また、データソースに複数のレコードタイプが含まれている場合に RECTYPE および MAPVALUE フィールドにも使用します。

- **FOCUS データソース** ALIAS 属性は、リクエストでフィールドの識別に使用するシノニムまたは別名を記述します。エイリアスには、任意の名前を割り当てることができます。一般的に、フィールドの実名よりも短い名前を選択します。たとえば、フィールド名が LAST_NAME の場合、エイリアスは LN にします。ALIAS 属性は、マスターファイルで必須ですが、ブランクの値を指定することもできます。エイリアスは、データソースを再構築せずに変更することができます。エイリアスが他のデータソースで参照されている場合、そのマスターファイルでも同様の変更が必要な場合があります。

表示データタイプ - USAGE

トピックス

表示フォーマットの指定

データタイプのフォーマット

整数フォーマット

倍精度浮動小数点数フォーマット

単精度浮動小数点数フォーマット

パック 10 進数フォーマット

数値の表示オプション

拡張通貨記号の表示オプション

文字フォーマット

日付フォーマット

日付表示オプション

日付区切り記号の制御

日付の変換

日付フィールドの使用

数値日付リテラル

演算式の日付フィールド

日付フィールドの変換

日付フィールドの内部表現

標準外日付フォーマットの表示

日付フォーマットのサポート

日付表示オプションを使用した文字および数値フォーマット

日付時間フォーマット

日付時間フィールドの記述

文字フォーマット AnV

テキストフィールドのフォーマット

USAGE 属性は、レポートに表示するフィールドまたは計算に使用するフィールドのフォーマットを記述します。この属性は、「FORMAT 属性」とも呼ばれます。

表示フォーマットの指定

手順/構文

表示フォーマットの指定

参照

USAGE 属性使用時の注意

FOCUS データソースでは ACTUAL 属性を使用しないため、USAGE 属性を使用してフィールドの格納方法も指定します。他のタイプのデータソースでは、ACTUAL 値に対応する USAGE 値を割り当てて、フィールドをデータソースに格納する場合のデータタイプと同一のデータタイプでフィールドを識別します。データが文字として格納される場合、レポートでのフィールドの表示方法に基づいて、USAGE を割り当てます。変換は、自動的に実行されます。USAGE 値に対応する ACTUAL 値についての詳細は、使用するデータアダプタのマニュアルを参照してください。

フィールドのデータタイプおよび長さを選択する以外に、日付フォーマット、通貨記号表示 (浮動)、ゼロサプレス (ゼロを非表示) などの表示オプションを指定することもできます。これらのオプションを使用して、レポートにフィールドを表示する方法をカスタマイズします。

構文

表示フォーマットの指定

USAGE = t1[d]

説明

t

データタイプです。有効値には、A (文字)、F (単精度浮動小数点数)、D (倍精度浮動小数点数)、I (整数)、P (パック 10 進数)、TX (テキスト) があります。また、D、W、M、Q、Y (日付) の有効な組み合わせを使用することもできます。

l

長さの指定です。指定はデータタイプにより異なります。詳細は、各データタイプのセクションを参照してください。なお、日付フォーマットのフィールドには長さは指定しません。

d

1 つまたは複数の表示オプションです。データタイプごとに表示オプションが異なります。詳細は、各データタイプのセクションを参照してください。

USAGE 値の最大長は 8 バイトです。

フィールドのタイプおよび長さを指定する値は、フィールドを表示または格納する際に割り当てられる出力位置の値に反映されます。表示オプションは、フィールドを表示または印刷する場合にのみ影響します。抽出ファイルなどの表示に関係しないフィールドの取得時には表示オプションは無効です。

注意：指定した USAGE フォーマットで数値フィールドを表示できない場合は、アスタリスク (*) が表示されます (例、集計結果の桁数が大きすぎる場合)。

詳細は、各フォーマットタイプのセクションを参照してください。

参照

USAGE 属性使用時の注意

USAGE 属性の使用時には次の規則が適用されます。

- **エイリアス** USAGE 属性のエイリアスは FORMAT です。
- **変更** ほとんどのデータソースでは、USAGE 属性で指定するタイプおよび長さは、そのフィールドの ACTUAL 属性で有効なタイプおよび長さの範囲内でのみ変更することができます。表示オプションはいつでも変更することができます。

FOCUS データソースでは、タイプの指定を変更することはできません。フィールドのデータタイプが I、F、D、P の場合、長さの指定は格納時には影響せず、表示のみに反映されるため、長さの指定を変更することができます。P フィールドでは長さ指定の小数部を変更することはできません。A (文字) フィールドの長さ指定は、REBUILD 機能を使用する場合にのみ変更することができます。表示オプションはいつでも変更することができます。

データタイプのフォーマット

次のフォーマットタイプを指定することができます。

- **数値** 数値フォーマットには、整数、単精度浮動小数点数、倍精度浮動小数点数、パック 10 進数の 4 つのタイプがあります。詳細は、389 ページの「[数値の表示オプション](#)」を参照してください。
- **文字** 文字フォーマットは、数字、文字、およびその他の文字で構成され、一連の文字として解釈されるすべての値に使用することができます。

- **日付** 日付フォーマットを使用すると、年、四半期、月、日、曜日などの日付要素を定義して、次の操作を実行することができます。
 - 日付によるソート
 - 日付の比較および計算
 - トランザクションでの日付の自動検証

たとえば、DECODE 関数を使用して日付値を割り当てるようなアプリケーションでは、部分的な日付機能を備えた日付表示オプションを使用して文字、整数、パック 10 進数フィールドを使用することもできます。
- **日付時間** 多くのリレーショナルデータソースで使用するタイムスタンプデータタイプと同様に、日付時間フォーマットは日付と時間の両方をサポートします。日付時間フィールドは、8 バイト、10 バイト、または 12 バイトで格納されます。日付には 4 バイト、時間にはマイクロ秒とナノ秒のどちらを使用するかにより、6 バイトまたは 8 バイトが使用されます。計算は、データタイプ内で直接割り当てた場合にのみ実行可能です。その他の演算はすべて、一連の日付時間関数で実行されます。
- **テキスト** テキストフィールドを使用して、大量のデータを格納したり、そのデータに改行を追加して表示したりすることができます。

整数フォーマット

整数に使用します。整数は、0 (ゼロ) から 9 までの値で構成される任意の数値で、小数部を含みません。

整数フィールドと日付表示オプションを使用して、限定的な日付サポートを提供することもできます。整数フィールドの使用についての詳細は、409 ページの「[日付表示オプションを使用した文字および数値フォーマット](#)」を参照してください。

整数の USAGE タイプは I です。詳細は、389 ページの「[数値の表示オプション](#)」を参照してください。長さ指定のフォーマットは次のとおりです。

n

説明

n

表示する桁数です。長さの最大値は 11 で、これには小数部、およびフィールドに負の値が格納される場合は、先頭のマイナス符号 (-) も含まれます。小数部 (最大で *n* マイナス 1) の桁数を指定することもでき、小数部を指定した場合、数値は、その桁数に先頭の小数点追加されて表示されます。

以下はその例です。

フォーマット	表示
I6	4316
I6.2	43.16
I2	22
I4	-617

倍精度浮動小数点数フォーマット

倍精度浮動小数点数フォーマットは、0 から 9 までの数字およびオプションの小数点で構成された任意の値に対して使用することができます。

倍精度浮動小数点数の USAGE タイプは D です。表示オプションについての詳細は、389 ページの「[数値の表示オプション](#)」を参照してください。長さ指定のフォーマットは次のとおりです。

`t[.s]`

説明

`t`

表示する桁数です。この桁数には、オプションの小数点、負の値を持つフィールド先頭のマイナス符号 (-) を含めて、33 バイト以内の数字を指定します。サポートされる有効桁数は、オペレーティングシステムにより異なります。

`s`

小数点の後に続く桁数です。t より小さい 31 桁までの数字を指定します。

以下はその例です。

フォーマット	表示
D8.2	3,187.54
D8	416

たとえば D8.2 の場合、数字の 8 は小数点および小数点以下の桁数を含めた全体の最大桁数を表します。数字の 2 は、全体の 8 桁のうち、小数点以下の桁数を表します。カンマ (,) は自動的に表示され、桁数には含まれません。

単精度浮動小数点数フォーマット

単精度浮動小数点数フォーマットは、小数部を含む数値を含めて、任意の数値に使用することができます。この数値は、0 から 9 までの数字、およびオプションの小数点で構成されます。倍精度浮動小数点数フォーマットとは異なり、このフォーマットは長さが 9 桁を超えることができないため、主に桁数の少ない小数に使用します。

単精度浮動小数点数の USAGE タイプは F です。表示オプションについての詳細は、389 ページの「[数値の表示オプション](#)」を参照してください。長さ指定のフォーマットは次のとおりです。

`t[.s]`

説明

`t`

表示する桁数です。この桁数には、最大 33 桁の数字、オプションの小数点、負の値を持つフィールド先頭のマイナス符号 (-) が含まれます。サポートされる有効桁数は、オペレーティングシステムにより異なります。

`s`

小数点の後に続く桁数です。t より小さい 31 桁までの数字を指定します。

以下はその例です。

フォーマット	表示
F5.1	614.2
F4	318

パック 10 進数フォーマット

パック 10 進数フォーマットは、小数点数を含む任意の数値で使用することができます。小数点数は、オプションの小数点、および 0 (ゼロ) から 9 までの任意の数字で構成されます。

パック 10 進数フィールドを日付表示オプションとともに使用して、限定的な日付サポートを活用することもできます。詳細は、409 ページの「[日付表示オプションを使用した文字および数値フォーマット](#)」を参照してください。

パック 10 進数の USAGE タイプは P です。表示オプションについての詳細は、389 ページの「[数値の表示オプション](#)」を参照してください。

長さ指定のフォーマットは次のとおりです。

`t[.s]`

説明

`t`

表示する桁数です。この桁数は、最大 31 桁の数字、オプションの小数点、負の値を持つフィールド先頭のマイナス符号 (-) を含めて、最大で 33 バイトを指定することができます。

`s`

小数点の後に続く桁数です。t より小さい 31 桁までの数字を指定します。

以下はその例です。

フォーマット	表示
P9.3	4168.368
P7	617542

数値の表示オプション

表示オプションを使用して、数値フォーマットを編集することができます。これらのオプションは、フィールドのデータを印刷または画面に表示する場合にのみ反映され、データソースに保存する場合には影響しません。

編集オプション	説明	効果
-	マイナス符号	負の数値データの右側にマイナス符号 (-) を表示します。 注意： フォーマットオプションの B、E、R、T、DMY、MDY、YMD ではサポートされません。
%	パーセント記号	数値データの横にパーセント記号 (%) を表示します。パーセント計算は実行されません。
A	負の符号の非表示	数値の絶対値を表示しますが、格納されている値には影響しません。 <ul style="list-style-type: none"> ❑ 負の符号を非表示にする USAGE 属性を含むフィールドを、HOLD ファイルに出力すると、HOLD ファイルには、符号付きの値が格納されます。また、負の符号を非表示にする USAGE 属性を HOLD ファイルに指定し、HOLD ファイルに対するリクエストを実行した場合、レポート出力では、マイナス符号 (-) は非表示になります。 ❑ 負の符号を非表示にするオプションを、次の表示オプションと併用することはできません。 <ul style="list-style-type: none"> ❑ B (ブラケットネガティブ) ❑ R (クレジットネガティブ) ❑ - (右側負符号)
B	ブラケットネガティブ	負の値を括弧で囲みます。

編集オプション	説明	効果
c	カンマサプレス	カンマ (,) を非表示にします。 数値フォーマットオプションの M (ドル記号、浮動) および N (ドル記号、固定)、データフォーマットの D (倍精度浮動小数点数) と組み合わせて使用します。
C	カンマ挿入	有効数字 3 桁ごとにカンマ (,) を挿入します。また、コンチネンタル 10 進数表記を適用している場合はピリオド (.) を挿入します。
DMY	日-月-年	日/月/年の形式で文字または整数データを日付として表示します。
E	指数表記	有効数字のみを表示します。
L	リーディングゼロ	先頭に 0 (ゼロ) を追加します。
M	通貨記号表示 (浮動) (US コードページの場合は \$)	有効数字の最上位桁の左側に浮動通貨記号を表示します。デフォルトの通貨記号は、コードページにより異なります。SET CURRSYMB=symbol コマンドを使用して、通貨記号として任意の文字または次の通貨コードのいずれかを指定することができます。 USD または '\$' - 米ドルを指定します。 GBP - イギリスポンドを指定します。 JPY - 日本円または中国元を指定します。 EUR - ユーロを指定します。
MDY	月-日-年	月/日/年の形式で文字または整数データを日付として表示します。

編集オプション	説明	効果
N	固定 \$ (US コード ページの場合)	<p>フィールドの左側に通貨記号を表示します。通貨記号は、各ページの最初の詳細行にのみ表示されます。デフォルトの通貨記号は、コードページにより異なります。SET CURRSYMB=<i>symbol</i> コマンドを使用して、通貨記号として任意の文字または次の通貨コードのいずれかを指定することができます。</p> <p>USD または '\$' - 米ドルを指定します。</p> <p>GBP - イギリスポンドを指定します。</p> <p>JPY - 日本円または中国元を指定します。</p> <p>EUR - ユーロを指定します。</p>
R	クレジット (CR) ネガティブ	負の値の後に「CR」を表示します。
S	ゼロサプレス	データ値が 0 (ゼロ) の場合、その位置にブランクを表示します。
T	月の変換	月を 3 文字の省略形で表示します。
YMD	年-月-日	年/月/日の形式で文字または整数データを日付として表示します。

例

数値表示オプションの使用

下表は、数値フィールドで使用可能な表示オプションの例を示しています。

オプション	フォーマット	データ	表示
マイナス符号	I2- D7- F7.2-	-21 -6148 -8878	21- 6148- 8878.00-

表示データタイプ - USAGE

オプション	フォーマット	データ	表示
パーセント記号	I2% D7% F3.2%	21 6148 48	21% 6,148% 48.00%
カンマサプレス	D6c D7Mc D7Nc	41376 6148 6148	41376 \$6148 \$ 6148
カンマ挿入	I6C	41376	41,376
ゼロサプレス	D6S	0	
ブラケットネガティブ	I6B	-64187	(64187)
クレジット (CR) ネガティブ	I8R	-3167	3167 CR
リーディングゼロ	F4L	31	0031
通貨記号表示 (浮動)	D7M	6148	\$6,148
通貨記号表示 (固定)	D7N	5432	\$ 5,432
指数表記	D12.5E	1234.5	0.12345D+04
年/月/日	I6YMD I8YYMD	980421 19980421	98/04/21 1998/04/21
月/日/年	I6MDY I8MDYY	042198 04211998	04/21/98 04/21/1998
日/月/年	I6DMY I8DMYY	210498 21041998	21/04/98 21/04/1998
月の変換	I2MT	07	JUL

次のように、複数の表示オプションを組み合わせることができます。

フォーマット	データ	表示
I5CB	-61874	(61,874)

すべてのオプションは、任意の順序で指定することができます。数値フォーマットオプションの M (ドル記号、浮動) および N (ドル記号、固定)、データフォーマットの D (倍精度浮動小数点数) を指定した場合は、自動的にオプション C (カンマ) が有効になります。オプションの L と S を同時に使用することはできません。文字または整数の USAGE 指定に表示オプションの M (月) が含まれている場合は、その指定の任意の場所にオプションの T (変換) を挿入することができます。浮動小数点数フィールドで使用できない日付表示オプション (D、M、T、Y) についての詳細は、409 ページの「[日付表示オプションを使用した文字および数値フォーマット](#)」を参照してください。

拡張通貨記号の表示オプション

参照

拡張通貨記号フォーマット

国際言語サポート (NLS) でデフォルトの通貨記号を構成した場合でも、レポート出力に表示する通貨記号を任意に選択することができます。その場合は、表示オプションの [通貨記号表示 (浮動) - M] (Floating dollar (M)) または [通貨記号表示 (固定) - N] (Non-floating dollar (N)) の代わりに拡張通貨記号フォーマットを使用します。表示オプションの [通貨記号表示 (浮動) - M] (Floating dollar (M)) または [通貨記号表示 (固定) - N] (Non-floating dollar (N)) を使用する場合は、デフォルトのコードページに関連付けられた通貨記号が表示されます。たとえば、英語 (米国) コードページを使用する場合は、ドル記号 (\$) が表示されます。

一方、拡張通貨記号フォーマットを使用すると、ドル記号 (\$) 以外の記号を表示することができます。たとえば米国ドル、イギリスポンド、日本円、中国元、またはユーロの記号を表示することができます。拡張通貨記号は、数値フォーマット (I、D、F、P) でサポートされます。

拡張通貨記号フォーマットは、数値表示フォーマットの末尾に 2 文字を組み合わせて指定します。この組み合わせの最初の文字には、感嘆符 (!) またはコロン (:) のいずれかを使用することができます。コロン (:) は ASCII および EBCDIC のすべてのコードページで正しく動作するため、コロン (:) を使用することを推奨します。感嘆符 (!) は、すべての EBCDIC コードページで共通する記号でないため、使用するコードページによっては感嘆符 (!) が別の記号に変換されて予期しない動作が発生する場合があります。下表は、サポートされる拡張通貨表示オプションの一覧です。

表示オプション	説明	例
:d または !d	ドル記号 (固定)	D12.2:d
:D または !D	ドル記号 (浮動)	D12.2:D
:e または !e	ユーロ記号 (固定)	F9.2:e
:E または !E	ユーロ記号 (浮動) (左側)	F9.2:E
:F または !F	ユーロ記号 (浮動) (右側)	F9.2:F
:l または !l	イギリスポンド記号 (固定)	D12.1:l
:L または !L	イギリスポンド記号 (浮動)	D12.1:L
:y または !y	日本円または中国元 (固定)	I9:y
:Y または !Y	日本円または中国元 (浮動)	I9:Y

参照

拡張通貨記号フォーマット

次のガイドラインが適用されます。

- フォーマット指定の最大長は 8 バイトです。
- 拡張通貨オプションは、フォーマットの最後のオプションとして指定する必要があります。
- 拡張通貨記号フォーマットに、表示オプションの [通貨記号表示 (浮動) - M] (Floating (M)) または [通貨記号表示 (固定) - N] (Non-floating (N)) を含めることはできません。

- 固定通貨記号は、レポートページの最初の行にのみ表示されます。(次の例のように) レポートの列に複数の通貨記号を表示するために、フィールドベースのフォーマット再設定を使用する場合、その 1 行目に関連した記号のみが表示されます。この場合は、固定通貨記号を使用することはできません。
- 端末 I/O プロシジャにより、小文字は大文字として送信されます。そのため、固定の拡張通貨記号はプロシジャでのみ指定することができます。
- 拡張通貨記号フォーマットは、単精度浮動小数点数、倍精度浮動小数点数、パック 10 進数、整数のフォーマットが設定されたフィールドに使用することができます。文字フォーマットおよび可変長文字フォーマットを使用することはできません。

文字フォーマット

参照

4 キロバイト文字フィールド使用時の注意

文字フォーマットは、数字、文字、およびその他の文字で構成され、一連の文字として解釈されるすべての値に使用することができます。

文字フィールドと日付表示オプションを使用して、限定的な日付サポートを利用することもできます。文字フィールドの使用方法についての詳細は、409 ページの「[日付表示オプションを使用した文字および数値フォーマット](#)」を参照してください。

文字の USAGE タイプは A です。長さ指定のフォーマットは n です。ここで、n はフィールドの最大バイト数を表します。文字フィールドの長さの最大値は、FOCUS ファイルセグメントでは 3968 バイト、XFOCUS ファイルセグメントでは 4096 バイトです。固定フォーマットのシーケンシャルデータソースでは、最大長は 4095 バイトです。フィールドの長さは、マスターファイル、DEFINE FILE コマンド、COMPUTE コマンドで定義することができます。

以下はその例です。

フォーマット	表示
A522	The minutes of today's meeting were submitted...
A2	B3
A24	127-A429-BYQ-49

標準の数値表示オプションは、文字データフォーマットに使用することはできません。ただし、実行時にパターンを指定して、文字データの表示を制御することができます。たとえば、製品コードを複数の区分で表示し、各区分をハイフン (-) で分割する場合、DEFINE コマンドに次の記述を含めます。

```
PRODCODE/A11 = EDIT (fieldname,'999-999-999') ;
```

説明

fieldname

既存のフィールド名です。新しく定義したフィールド名ではありません。

たとえば、値が 716431014 の場合、PRODCODE には 716-431-014 と表示されます。詳細は、[こちら](#) を参照してください。

参照

4 キロバイト文字フィールド使用時の注意

- 長い文字フィールドにインデックスを付けることはできません。
- FOCUS データソースでは、セグメントを 4 キロバイトページに収める必要があります。そのため、文字フィールドの最大長は、そのセグメント内の他のフィールドの長さにより異なります。
- 長い文字フィールドを印刷またはホールドすることはできますが、オンラインで参照することはできません。
- 長い文字フィールドはキーとして使用することができます。

日付フォーマット

日付フォーマットを使用すると、日付フィールドを定義した後、フィールド値を操作して正しい日付の値として表示することができます。日付フォーマットを使用すると、次の操作を実行することができます。

- 日付構成要素 (例、年、四半期、月、日、曜日) を定義し、日付フィールドから日付構成要素を簡単に抽出する。
- 日付の表示方法に関係なく、レポートを日付順でソートする。
- 特別な日付処理関数を使用せずに、日付演算を実行して日付を比較する。
- 表示または編集フォーマットを考慮せずに、一般的な表記方法で日付を指定する (例、JAN 1 1995)。
- トランザクションの日付を自動検証する。

日付表示オプション

参照

フィールドフォーマット Y、YY、M、W の格納方法

日付リテラルの解析表

日付フォーマットではタイプや長さは指定しません。その代わりに、日付構成要素オプション (D、W、M、Q、Y、YY) および表示オプションを指定します。下表は、これらのオプションの一覧です。

表示オプション	説明	効果
D	日	日の値を 1 から 31 の範囲で表示します。
M	月	月の値を 1 から 12 の範囲で表示します。
Y	年	年を 2 桁で表示します。
YY	4 桁の年	年を 4 桁で表示します。
T	月または日の変換	USAGE 設定に M が含まれる場合、月の 3 文字の省略形を大文字で表示します。それ以外の場合は、曜日を表示します。
t	月または日の変換	上記の大文字の T の場合と同様に機能します。ただし、月または日付の最初の文字が大文字で、後に続く文字は小文字になります。*
TR	月または日の変換	上記の大文字の T の場合と同様に機能します。ただし、省略名の代わりに月または日の完全名が表示されます。
tr	月または日の変換	上記の小文字の t の場合と同様に機能します。ただし、省略名の代わりに月または日の完全名が表示されます。*
Q	四半期	Q1 から Q4 を表示します。

表示オプション	説明	効果
W	曜日	USAGE 設定に他の日付構成要素オプションとともに使用して、3 文字の曜日の省略名を大文字で表示します。USAGE 設定にこの日付構成要素オプションのみを指定した場合は、曜日を数字で表示します (1 から 7 までの数字、月曜日が 1)。
w	曜日	上記の大文字の W の場合と同様に機能します。ただし、最初の文字が大文字で、後に続く文字が小文字になります。*
WR	曜日	上記の大文字の W の場合と同様に機能します。ただし、省略名の代わりに曜日の完全名が表示されます。
wr	曜日	上記の小文字の w の場合と同様に機能します。ただし、省略名の代わりに曜日の完全名が表示されます。*
J[UL]or JULIAN	ユリウス暦	ユリウス暦フォーマットで日付を表示します。
YYJ[UL]	ユリウス暦	ユリウス暦フォーマットの日付を YYYYDDD フォーマットで表示します。この 7 桁のフォーマットは、4 桁の年に続けて 1 月 1 日からの通算日を表示します。たとえば、ユリウス暦フォーマットの 2001 年 1 月 3 日は 2001003 になります。

*** 注意：**これらの表示オプションを使用する場合は、マスターファイルで実際にフィールドが小文字で保存されていることを確認してください。

次の日付構成要素の組み合わせは、日付フォーマットではサポートされません。

I2M, A2M, I2MD, A2MD

参照

フィールドフォーマット Y、YY、M、W の格納方法

Y、YY、M は SmartDate フォーマットではありません。SmartDate フォーマットの YMD および YYMD は、「12/31/1900」という基準日からのオフセットとして格納されます。また、SmartDate フォーマットの YM、YQ、YYM、YYQ は、「01/1901」という基準日からのオフセットとして格納されます。W フォーマットは、曜日を表す 1 から 7 までの 1 桁の整数で格納されます。Y、YY、M フォーマットは、整数として格納されます。Y および M の表示上の長さは 2 です。YY の表示上の長さは 4 です。Y および YY フィールドフォーマットを使用する場合は、次の 2 点に注意する必要があります。

- Y フォーマットは、DEFCENT および YRTHRESH 設定に基づいてソートされません。Y フォーマットのフィールドは、位置をずらした数字ではなく 4 桁の整数のため、YY フィールドとは異なります。
- DEFCENT および YRTHRESH を使用して、フィールドフォーマットを Y から YY に変換することができます。

参照

日付リテラルの解析表

下表は、日付フォーマットの動作を示しています。日付フォーマットの入力桁数がそれぞれの列に示されています。フィールドの USAGE またはフォーマットがそれぞれの行に示されています。行と列の交差部は、入力およびフォーマットの結果を示しています。

日付フォーマット	1	2	3	4
YYMD	*	*	CC00/0m/dd	CC00/mm/dd
MDYY	*	*	*	*
DMYY	*	*	*	*
YMD	*	*	CC00/0m/dd	CC00/mm/dd
MDY	*	*	*	*
DMY	*	*	*	*
YYM	CC00/0m	CC00/mm	CC0y/mm	CCyy/mm
MYM	*	*	*	*
YM	CC00/0m	CC00/mm	CC0y/mm	CCyy/mm

日付フォーマット	1	2	3	4
MY	*	*	0m/CCyy	mm/CCyy
M	0m	mm	*	*
YYQ	CC00/q	CC0y/q	CCyy/q	0yyy/q
QYY	*	*	q/CCyy	*
YQ	CC00/q	CC0y/q	CCyy/q	0yyy/q
QY	*	*	q/CCyy	*
Q	q	*	*	*
JUL	00/00d	00/0dd	00/ddd	0y/ddd
YYJUL	CC00/00d	CC00/0dd	CC00/ddd	CC0y/ddd
YY	000y	00yy	0yyy	yyyy
Y	0y	yy	*	*
D	0d	dd	*	*
W	w	*	*	*

日付フォーマット	5	6	7	8
YYMD	CC0y/mm/dd	CCyy/mm/dd	0yyy/mm/dd	yyyy/mm/dd
MDYY	0m/dd/CCyy	mm/dd/CCyy	0m/dd/yyyy	mm/dd/yyyy
DMYY	0d/mm/CCyy	dd/mm/CCyy	0d/mm/yyyy	dd/mm/yyyy
YMD	CC0y/mm/dd	CCyy/mm/dd	0yyy/mm/dd	yyyy/mm/dd
MDY	0m/dd/CCyy	mm/dd/CCyy	0m/dd/yyyy	mm/dd/yyyy
DMY	0d/mm/CCyy	dd/mm/CCyy	0d/mm/yyyy	dd/mm/yyyy
YYM	0yyy/mm	yyyy/mm	*	*

日付フォーマット	5	6	7	8
MY	0m/yyyy	mm/yyyy	*	*
YM	0yyy/mm	yyyy/mm	*	*
MY	0m/yyyy	mm/yyyy	*	*
M	*	*	*	*
YYQ	yyyy/q	*	*	*
QYY	q/yyyy	*	*	*
YQ	yyyy/q	*	*	*
QY	q/yyyy	*	*	*
Q	*	*	*	*
JUL	yy/ddd	*	*	*
YYJUL	CCyy/ddd	0yyy/ddd	yyyy/ddd	*
YY	*	*	*	*
Y	*	*	*	*
D	*	*	*	*
W	*	*	*	*

注意

- CC は、DFC/YRT 設定で指定された世紀に相当する 2 桁の値を表します。
- アスタリスク (*) は、メッセージ FOC177 (無効な日付定数) を表します。
- 日付リテラルは右から左に読み取られます。日付リテラルおよび日付フィールドは計算式に使用することができます。詳細は、

日付区切り記号の制御

日付を表示する際に、日付の区切り記号を制御することができます。基本的な日付フォーマット (例、YMD、MDYY) では、日付構成要素をスラッシュ記号 (/) で区切ります。年月フォーマットでも同様に、年と四半期を空白で区切ります (例、94 Q3、Q3 1994)。単一コンポーネントのフォーマットでは、1つの数字または名前のみが表示されます。

区切り記号としてピリオド (.)、ダッシュ (-)、空白を指定したり、区切り記号を完全に除外したりすることができます。下表は、区切り記号を変更する際の USAGE 設定を示しています。

フォーマット	表示
YMD	93/12/24
Y.M.D	93.12.24
Y-M	93-12
YBMBD	93 12 24 (文字 B は空白を表します)
Y M D	931224 (連結記号「 」は区切り記号を省略します)

注意

- 日付区切り記号の変更が可能な日付フォーマットには、YYMD、MDYY、DMYY、YMD、MDY、DMY、YYM、MYY、YM、MY、YYQ、QYY、YQ、QY があります。
- 日付変換オプションを含むフォーマットで日付区切り記号を変更することはできません。
- 日付表示オプション (例、I8YYMD) を含む文字または数値フォーマットで日付区切り記号のスラッシュ (/) を変更することはできません。

日付の変換

数値で表示した月および曜日は、JAN、January、Wed、Wednesday などの文字に変換することができます。変換した月または曜日には、3文字の省略名または完全名を使用することができます。名前は大文字、小文字のいずれかで表示することができます。また、曜日 (例、Monday) は日付の前後いずれかに追加することができます。これらのすべてのオプションはそれぞれ独立して機能します。

変換	表示
MT	JAN
Mt	Jan
MTR	JANUARY
Mtr	January
WR	MONDAY
wr	Monday

例

日付フォーマットの使用

下表は、FOCUS 以外のデータソースに格納したデータの USAGE および ACTUAL フォーマットのサンプルを示しています。[値] 列には実際のデータ値、[表示] 列には表示時のデータ値が示されています。

USAGE	ACTUAL	値	表示
wrMtrDYY	A6YMD	990315	Monday, March 15 1999
YQ	A6YMD	990315	99 Q1
QYY	A6YMD	990315	Q1 1999
YMD	A6	990315	99/03/15
MDYY	A6YMD	990315	03/15/1999

なお、ACTUAL フォーマットの日付属性は、FOCUS 以外のデータソースに日付を格納する際の順序を指定します。ACTUAL フォーマットで月、日、年の順序を指定しない場合、USAGE フォーマットの指定から推定されます。

日付フィールドの使用

日付フォーマットに指定したフィールドは、日付の入力時に値が自動検証されます。日付の値として、一般表記の日付リテラル (例、JAN 12 1999) を入力することも、数値の日付リテラル (例、011299) を入力することもできます。

一般表記の日付リテラルを使用すると、日付構成要素間にブランクを挿入したり、月の省略名を使用したりして、簡単で分かりやすい方法で日付を指定することができます。たとえば、1999 年 4 月 25 日は次のいずれかの方法で一般表記の日付リテラルを指定することができます。

```
APR 25 1999
25 APR 1999
1999 APR 25
```

一般表記の日付リテラルは、すべての日付計算およびデータソースの更新機能で使用することができます。次のコードは、その一例を示しています。

```
In WHERE screening           WHERE MYDATE IS 'APR 25 1999'
In arithmetic expressions    MYDATE - '1999 APR 25'
In computational date comparisons  IF MYDATE GT '25 APR 1999' In
comma-delimited data        ...,MYDATE = APR 25 1999, ...
```

下表は、一般表記の日付リテラルのフォーマットを示しています。

リテラル	フォーマット
年-月-日	4 桁の年、大文字 3 文字の省略名、または完全名の月、1 桁または 2 桁の日付で構成されます (例、1999 APR 25 または APRIL 25 1999)。
年-月	年および月は上記のとおりです。
年-四半期	年は上記のとおりです。四半期には Q および四半期数が追加されます (例、1999 Q3)。
月	月は上記のとおりです。
四半期	四半期は上記のとおりです。

リテラル	フォーマット
曜日	曜日を表す大文字 3 文字の省略名または完全名です (例、MON、MONDAY)。

一般表記の日付けリテラルの日付構成要素は、ターゲットフィールドの USAGE 設定で指定した順序に関係なく、任意の順序で指定することができます。複数の日付構成要素は、1 つまたは複数のブランクで区切ります。

たとえば、日付フィールドの USAGE 設定が YM の場合、フィールドに入力する一般表記の日付リテラルには、年および月を任意の順序で記述することができます。MAY 1999 および 1990 APR はいずれも有効なリテラルです。

数値日付リテラル

数値日付リテラルは、単純な数字の配列という点で、一般表記の日付リテラルとは異なります。数値日付リテラルで入力する日付構成要素の順序は、それに対応する USAGE 設定の日付構成要素の順序に一致させる必要があります。また、数値日付リテラルには、USAGE 設定に含まれているすべての日付構成要素を含める必要があります。たとえば、USAGE 設定が DMY の場合、1999 年 4 月 25 日は次のように記述する必要があります。

250499

数値日付リテラルは、すべての日付計算およびデータソースの更新機能で使用することができます。

演算式の日付フィールド

演算式で日付フィールドを操作する際の一般的な規則として、1 つの演算式に複数の日付フィールドを使用する場合は、共通の日付構成要素を指定する必要があります。日付構成要素は任意の順序で指定することができます。この場合、表示オプションは無視されません。Y、YY のいずれか、および Q、M、W、D は有効な構成要素です。

なお、四半期、月、曜日に割り当てた演算式は、それぞれ 4、12、7 で剰余計算されるため、第 5 四半期や 13 番目の月などの異常な出力結果が防止されます。

たとえば、NEWQUARTER と THISQUARTER の両方に「Q」という USAGE が指定され、THISQUARTER の値が 2 の場合、次のステートメントの NEWQUARTER の値は、1 になります (つまり、5 を 4 で除算した剰余である 1)。

NEWQUARTER = THISQUARTER + 3

日付フィールドの変換

手順/構文

日付フィールドの変換

日付フィールドは、フォーマット変換および日付構成要素変換という 2 つの方法で変換することができます。フォーマット変換では、日付フォーマットフィールドの値を日付表示オプションを使用する文字フィールドまたは整数フィールドに割り当てることができます (詳細は、次のセクションを参照)。逆の変換も可能です。

日付構成要素変換では、USAGE で日付構成要素を指定するフィールドを、異なる日付構成要素を指定するもう 1 つのフィールドに割り当てることができます。

たとえば、「REPORTDATE (DMY)」という値を ORDERDATE (Y) に割り当てることができます。この場合は、REPORTDATE から年が抽出されます。REPORTDATE が Apr 27 99 の場合、ORDERDATE は 99 です。

ORDERDATE の値を REPORTDATE に割り当てすることもできます。ORDERDATE の値が 99 の場合、REPORTDATE の値は Jan 1 99 です。この場合、ミッシング日付構成要素の値が REPORTDATE に適用されます。

構文

日付フィールドの変換

```
field1/format = field2;
```

説明

field1

日付フォーマットフィールド、または日付表示オプションを使用する文字あるいは整数フォーマットフィールドです。

format

field1 (ターゲットフィールド) の USAGE (または FORMAT) 設定です。

field2

日付フォーマットフィールド、または日付表示オプションを使用する文字あるいは整数フォーマットフィールドです。*field1* および *field2* のフォーマットタイプ (文字、整数、日付) と日付構成要素 (YY、Y、Q、M、W、D) は一致する必要はありません。

日付フィールドの内部表現

日付フィールドは、内部的には 4 バイトのバイナリ整数で表現されます。これは、日付フォーマットの基準日からの経過時間を表します。各フィールドでは、経過時間の単位としてフィールドの最小日付構成要素が使用されます。

たとえば、REPORTDATE の USAGE 設定が MDY の場合、経過時間は日単位で測定され、内部的にはフィールドに入力した日付と基準日との間の経過日数が格納されます。たとえば、February 13, 1964 の数値リテラル (つまり 021364) を入力し、そのフィールドを出力すると、02/13/64 と表示されます。これを式に使用すると次のようになります。

```
NEWDATE = 'FEB 28 1964' - REPORTDATE ;  
DAYS/D = NEWDATE ;
```

DAYS の値は 15 になります。ただし、REPORTDATE の内部表現は、December 31, 1900 と February 13, 1964 との間の日数を表す 4 バイトのバイナリ整数になります。

経過時間と同様に、基準日もフィールドの最小日付構成要素の単位に基づいて測定されます。たとえば、YQ フィールドの場合、経過時間は四半期単位で測定され、基準日は 1901 年の第 1 四半期になります。YM フィールドの場合、経過時間は月単位で測定され、基準日は 1901 年の最初の月になります。

レポートにブランクまたは実際の基準日を表示するには、SET DATEDISPLAY コマンドを使用します。詳細は、[こちら](#)を参照してください。デフォルト値の OFF は、日付が基準日と一致する場合にブランクを表示します。この値を ON に設定すると、実際の基準日の値が表示されます。

通常は、日付フォーマットの内部表現を意識する必要はありません。ただし、基準日に設定したすべての日付はブランクとして表示され、ブランクまたはすべて 0 (ゼロ) と入力したすべての日付フィールドは確認時に受容されて基準日として解釈されます。これらのフィールドはブランクとして表示されますが、日付計算および式では基準日として解釈されます。

標準外日付フォーマットの表示

手順/構文

ALLOWCVTERR の呼び出し

デフォルト設定では、FOCUS 以外のデータソースの日付フィールドに無効な日付が含まれている場合、メッセージが表示され、そのレコード全体がレポートに表示されません。たとえば、A6 の ACTUAL および YMD の USAGE の日付フィールドに '980450' が含まれている場合、そのフィールドを含むレコードは表示されません。無効な日付を含むレコードの残りの部分を表示するには、SET ALLOWCVTERR コマンドを使用します。

注意：ALLOWCVTERR パラメータは、一時項目 (DEFINE) ではサポートされません。

構文

ALLOWCVTERR の呼び出し

```
SET ALLOWCVTERR = {ON|OFF}
```

説明

ON

無効な日付を含むフィールドを表示します。

OFF

無効な日付を検知した場合に診断メッセージを生成し、その日付を含むレコードを表示しません。デフォルト値は OFF です。

無効な日付が検知された場合、ALLOWCVTERR は MISSING=ON の設定に基づいて、フィールドの値を MISSING、基準日のいずれかに設定します。

下表は、ALLOWCVTERR=ON に設定された状態で無効な日付が存在すると仮定した場合の DATEDISPLAY および MISSING の間の相互作用の結果を示しています。

	MISSING=OFF	MISSING=ON
DATEDISPLAY=ON	基準日の 19001231 または 1901/1 を表示します。	.
DATEDISPLAY=OFF	ブランクを表示します。	.

DATEDISPLAY は、基準日の表示方法にのみ影響します。

日付フォーマットのサポート

日付フォーマットフィールドは、次の機能では特別な方法で使用されます。

- **ダイアログマネージャ** 変数を一般表記の日付リテラルに設定すると、その変数は日付フィールドとして機能します。以下はその例です。

```
-SET &NOW = 'APR 25 1960' ;
-SET &LATER = '1990 25 APR' ;
-SET &DELAY = &LATER - &NOW ;
```

この場合、&DELAY の値は、2 つの日付間の日数 (10,957) になります。

- **抽出ファイル** SAVB ファイルおよびフォーマットが設定されていない HOLD ファイルの日付フィールドは、フィールドの実際の値と標準基準日との差を表す 4 バイトのバイナリ整数として格納されます。SAVE ファイルおよびフォーマットが設定された HOLD ファイルの日付フィールド (例、USAGE WP) は、表示オプションなしで格納されます。
- **グラフ** 日付フィールドは、ACROSS および BY 句のソートフィールドとしてサポートされません。
- **FML** 日付フィールドは、RECAP ステートメントではサポートされません。

日付表示オプションを使用した文字および数値フォーマット

標準の日付フォーマットを使用して日付を表す以外に、文字、整数、パック 10 進数フィールドで日付表示オプション (D、M、Y、T) を使用することもできます。ただし、標準の日付フォーマットで使用可能な日付サポートの一部は使用することができません。

文字および整数フィールドで日付表示オプションを使用すると、特別な日付関数を使用した場合に、限定的な日付機能が提供されます。詳細は、

日付表示オプションを使用して文字または数値フィールドを日付として表現する場合は、年、月、日を指定することができます。これらの 3 つの要素をすべて指定すると、日付は 6 桁 (年を 4 桁で表示する場合は 8 桁) になり、USAGE には次のフォーマットを設定することができます。

フォーマット	表示
I6MDY	04/21/98
I6YMD	98/04/21
P6DMY	21/04/98

フォーマット	表示
I8DMYY	21/04/1998

フォーマットの M の直後に T を追加して、月の値 (1 から 12) をそれぞれ対応する月の名前に変換することができます。以下はその例です。

フォーマット	データ	表示
I6MTDY	05/21/98	MAY 21 98
I4MTY	0698	JUN 98
I2MT	07	JUL

日付の要素が月のみの場合、I2MT フォーマットでは、値 4 は APR と表示されます。この方法は、レポートで行や列を月別にソートする場合に役立ちます。これらの値は、文字ではなく数値に基づいてソートされるため、JAN、FEB、MAR のように日付順に正しく表示されます。なお、T 表示オプションを使用しない場合、I2M は通貨記号表示 (浮動) の整数として解釈されます。

日付時間フォーマット

手順/構文

ISO 標準日付時間表記の有効化

多くのリレーショナルデータソースで使用するタイムスタンプデータタイプと同様に、日付時間フォーマットは日付と時間の両方をサポートします。

日付時間フィールドは、8 バイト、10 バイト、または 12 バイトで格納されます。日付には 4 バイト、時間にはマイクロ秒とナノ秒のどちらを使用するかにより、6 バイトまたは 8 バイトが使用されます。

演算処理では、同一データタイプ間の直接代入、つまり、文字から文字へ、数値から数値へ、日付から日付へ、日付時間から日付時間への代入処理のみを行えます。その他の演算はすべて、一連の日付時間関数で実行されます。日付時間フィールドを操作するサブルーチンについての詳細は、『関数リファレンス』を参照してください。

日付時間フォーマットは、ISO 8601:2000 日付時間表記標準に準拠した出力値の生成および入力値の受容もサポートします。この表記を有効にするには、SET パラメータおよび特定のフォーマットオプションを使用します。

構文

ISO 標準日付時間表記の有効化

```
SET DTSTANDARD = {OFF|ON|STANDARD|STANDARDU}
```

説明

OFF

ISO 8601:2000 標準日付時間表記の準拠をサポートしません。デフォルト値は OFF です。

ON|STANDARD

ISO 標準フォーマットの認識および出力を有効にします。これを有効にすると、日付と時間の区切り文字として T を使用すること、秒を桁数で表すときの区切り文字にピリオド (.) およびカンマ (,) を使用すること、世界時の末尾に Z を使用すること、タイムゾーン情報付きの入力の受容が可能になります。STANDARD は ON の同義語です。

STANDARDU

ISO 標準フォーマットを (STANDARD と同様に) 有効にし、可能な場合は入力文字列を対応する世界時の時間に変換します。世界時は、以前は「グリニッジ標準時」として知られていました。このオプションにより、アプリケーションで一貫した日付時間値を格納することが可能になりました。

例 SET DTSTANDARD の使用

次のリクエストは、入力の日付時間値に ISO 8601:2000 日付時間標準フォーマットを使用します。SET DTSTANDARD=OFF に設定すると、リクエストは「(FOC177)：無効な日付データです：」で終了します。

```
SET DTSTANDARD = &STAND
DEFINE FILE EMPLOYEE
- * The following input is format YYYY-MM-DDThh:mm:ss.STZD
DT1/HYYMds = DT(2004-06-01T19:20:30.45+01:00);
- * The following input has comma as the decimal separator
DT2/HYYMds = DT(2004-06-01T19:20:30,45+01:00);
DT3/HYYMds = DT(20040601T19:20:30,45);
DT4/HYYMDUs = DT(2004-06-01T19:20:30,45+01:00);
END
TABLE FILE EMPLOYEE
HEADING CENTER
"DTSANDARD = &STAND "
" "
SUM CURR_SAL NOPRINT DT1 AS 'DT1: INPUT = 2004-06-01T19:20:30.45+01:00'
OVER DT2 AS 'DT2: INPUT = 2004-06-01T19:20:30,45+01:00'
OVER DT3 AS 'DT3: INPUT = 20040601T19:20:30,45'
OVER DT4 AS 'DT4: OUTPUT FORMAT HYYMDUs'
END
```

DTSTANDARD= STANDARD に設定すると、出力は入力値が受容されたことを示しますが、DT1、DT2、DT4 のタイムゾーンのオフセット (+01:00) は、出力で無視されます。DT4 のフォーマットの文字 U により、日付と時間の区切り文字として T が使用されます。

DTSANDARD = STANDARD

```
DT1: INPUT = 2004-06-01T19:20:30.45+01:00 2004-06-01 19:20:30.450
DT2: INPUT = 2004-06-01T19:20:30,45+01:00 2004-06-01 19:20:30.450
DT3: INPUT = 20040601T19:20:30,45 2004-06-01 19:20:30.450
DT4: OUTPUT FORMAT HYYMDUs 2004-06-01T19:20:30.450
```

DTSTANDARD= STANDARDU に設定すると、出力では、タイムゾーンのオフセット (+01:00) が差し引かれて、DT1、DT2、DT4 の値は標準時間に変換されます。

DTSANDARD = STANDARDU

```
DT1: INPUT = 2004-06-01T19:20:30.45+01:00 2004-06-01 18:20:30.450
DT2: INPUT = 2004-06-01T19:20:30,45+01:00 2004-06-01 18:20:30.450
DT3: INPUT = 20040601T19:20:30,45 2004-06-01 19:20:30.450
DT4: OUTPUT FORMAT HYYMDUs 2004-06-01T18:20:30.450
```

日付時間フィールドの記述

手順/構文

表示オプションを使用しない数値日付時間値の記述

時間値のみの記述

日付時間値の記述

参照

時間値のみの表示オプション

日付時間フィールドの日付構成要素に使用する表示オプション

日付時間フィールドの時間構成要素に使用する表示オプション

日付時間フォーマット使用時の注意

マスターファイルの USAGE (または FORMAT) 属性を使用して、日付時間フィールド値をレポート出力およびフォームに表示する方法、これらの値が式および関数で動作する方法を指定します。FOCUS データソースでは、この属性で値を格納する方法も指定します。

日付時間フィールドはフォーマットタイプの H で指定します。日付時間フィールドの USAGE 属性に H フォーマットコードを設定して、フィールドの長さまたは関連する日時表示オプションのいずれかを指定することができます。

日付時間フィールドの MISSING 属性には、ON または OFF を指定することができます。この属性を OFF に設定すると、日付時間フィールドに値が存在しない場合にデフォルトのブランクが表示されます。

構文

表示オプションを使用しない数値日付時間値の記述

このフォーマットは、文字 HOLD ファイルまたはトランザクションファイルでの使用に適しています。

USAGE = Hn

説明

n

1 から 23 バイトまでのフィールドの長さです。この長さには、日付の表示に最大で 8 バイト、時間の表示に最大で 9、12、または 15 バイトが含まれます。長さが 20 バイト未満の場合、日付は右側で切り捨てられます。

8 バイトの日付 YYYYMMDD には、年に 4 桁、月に 2 桁、日付に 2 桁が含まれます。

9 バイトの時間 HHMMSSsss には、時間に 2 桁、分に 2 桁、秒に 2 桁、ミリ秒に 3 桁が含まれます。ミリ秒構成要素には、秒の小数部が 3 桁で表されます。

12 バイトの時間 HHMMSSsssmmm には、時間に 2 桁、分に 2 桁、秒に 2 桁、ミリ秒に 3 桁、マイクロ秒に 3 桁が含まれます。ミリ秒構成要素には、秒の値の小数部が 3 桁で表されます。マイクロ秒構成要素には、ミリ秒の値に加えてさらに 3 桁の小数部が表されます。

15 バイトの時間 HHMMSSsssmmmnnn には、2 桁の時間、2 桁の分、2 桁の秒、3 桁のミリ秒に、ミリ秒のマイクロ秒が含まれます。ミリ秒構成要素には、秒の値の小数部が 3 桁で表されます。マイクロ秒構成要素には、ミリ秒の値に加えてさらに 3 桁の小数部が表されます。ナノ秒構成要素には、ミリ秒の値に加えてさらに 3 桁の小数部が表されます。

このフォーマットでは、日付と時間構成要素間の空白、構成要素内の小数点、空白、区切り記号はありません。時間は 24 時間制で入力します。たとえば、19991231225725333444 の値は 1999/12/31 10:57:25.333444PM を表します。

構文

時間値のみの記述

USAGE = *Htimefmt1*

説明

timefmt1

時間のみを表示するための USAGE フォーマットです。時間、分、秒の各構成要素は常にコロン (:) で区切り、空白は挿入しません。時間値には、午前午後 (am/pm) を表す記号の直前に、1つの空白を含めることができます。詳細は、415 ページの「[時間値のみの表示オプション](#)」を参照してください。

参照

時間値のみの表示オプション

下表は、時間のみの USAGE 属性に使用する有効な時間表示オプションの一覧です。ここでは、時間の値として 2:05:27.123456444 a.m. を想定します。

オプション	説明	効果
H	<p>時間 (2 桁)。</p> <p>フォーマットにこのオプションの a、A、または B が含まれる場合、時間の値は 01 から 12 です。</p> <p>それ以外の場合、時間の値は 00 から 23 になり、00 が午前零時を表します。</p>	<p>2 桁の時間を表示します。以下はその例です。</p> <p>USAGE = HH は 02 を表示します。</p>
h	<p>ゼロサプレス (ゼロを非表示) を使用した時間。</p> <p>フォーマットにこのオプションの a、b、A、B のいずれかが含まれる場合、時間の値は 1 から 12 です。</p> <p>それ以外の場合は、時間は 0 (ゼロ) から 23 です。</p>	<p>ゼロサプレス (ゼロを非表示) を使用して時間を表示します。以下はその例です。</p> <p>USAGE = Hh は 2 を表示します。</p>
I	<p>分 (2 桁)。</p> <p>分の値は 00 から 59 です。</p>	<p>2 桁の分を表示します。以下はその例です。</p> <p>USAGE = HHI は 02:05 を表示します。</p>
i	<p>ゼロサプレス (ゼロを非表示) を使用した分。</p> <p>分の値は 0 から 59 です。</p>	<p>ゼロサプレス (ゼロを非表示) を使用して分を表示します。これを時間フォーマット (H または h) とともに使用することはできません。以下はその例です。</p> <p>USAGE = Hi は 5 を表示します。</p>

オプション	説明	効果
S[x]	<p>秒 (2 桁)。秒の値の後には、x オプションを使用して、9 桁まで指定することができます。ここで、x は 1 から 9 までの数字を表します。そのほかに、s、m、n フォーマットを使用して、3 桁、6 桁、9 桁を表示することができます。</p> <p>S: 00 から 59</p>	<p>2 桁の秒を表示します。以下はその例です。</p> <p>USAGE = HHIS は 02:05:27 を表示します。</p> <p>USAGE = HHIS1 は 02:05:27.1 を表示します。</p>
s	<p>ミリ秒 (秒の小数点以下 3 桁)。</p> <p>000 から 999</p>	<p>秒を小数点以下 3 桁に表示します。以下はその例です。</p> <p>USAGE = HHISs は 02:05:27.123 を表示します。</p>
m	<p>マイクロ秒 (ミリ秒以下の追加の 3 桁)。</p> <p>000 から 999</p>	<p>秒を小数点以下 6 桁に表示します。以下はその例です。</p> <p>USAGE = HSsm は 27.123456 を表示します。</p>
n	<p>ナノ秒 (マイクロ秒以下の追加の 3 桁)。</p> <p>000 から 999</p>	<p>秒を小数点以下 9 桁に表示します。以下はその例です。</p> <p>USAGE = HSsn は 27.123456444 を表示します。</p>
A	<p>大文字の AM または PM を含む 12 時間制表示。</p>	<p>01 から 12 の時間に AM または PM を続けて表示します。以下はその例です。</p> <p>USAGE = HHISA は 02:05:27AM を表示します。</p>

オプション	説明	効果
a	小文字の am または pm を含む 12 時間制表示。	01 から 12 の時間に am または pm を続けて表示します。以下はその例です。 USAGE = HHISa は 02:05:27am を表示します。
B	大文字の AM または PM、およびその前に 1 つの空白を含む 12 時間制表示。	01 から 12 の時間に 1 つの空白、AM または PM を続けて表示します。以下はその例です。 USAGE = HHISB は 02:05:27 AM を表示します。
b	小文字の am または pm、およびその前に 1 つの空白を含む 12 時間制表示。	01 から 12 の時間に 1 つの空白、am または pm を続けて表示します。以下はその例です。 USAGE = HHISb は 02:05:27 am を表示します。
Z	ユニバーサル時間を示す Z を含めた 24 時間表示。Z を AM/PM とともに指定することはできません。	01 から 24 の時間に Z を続けて表示します。以下はその例です。 USAGE = HHISZ は 14:30[:20.99]Z を表示します。

フォーマットに複数の時間表示オプションを含める場合は、次の規則が適用されます。

- 時間、分、秒、ミリ秒、マイクロ秒、ナノ秒の順にコンポーネントを指定します。
- 最初のオプションには、時間、分、秒のいずれかを指定する必要があります。
- 中間の構成要素を省略することはできません。時間を指定した場合、次のオプションは分にする必要があります。秒にすることはできません。

注意：AM/PM 時間表示オプションのいずれかを指定しない限り、時間構成要素では 24 時間制が使用されます。

構文 **日付時間値の記述**

USAGE = Hdatefmt [separator] [timefmt2]

説明

datefmt

日付時間フィールドの日付の部分を表示するための USAGE フォーマットです。詳細は、418 ページの「[日付時間フィールドの日付構成要素に使用する表示オプション](#)」を参照してください。

separator

日付構成要素間の区切り文字です。デフォルトの区切り文字はスラッシュ (/) です。その他の有効な区切り文字には、ピリオド (.)、ハイフン (-)、空白 (B)、なし (N) があります。変換された月では、これらの区切り文字は k オプションの未使用時のみ指定可能です。

STANDARD と STANDARDU の設定では、日付の区切り文字は常にハイフン (-) です。日付と時間のデフォルトの区切り文字は空白です。ただし、文字 U を区切り文字オプションとして指定する場合、日付と時間は文字 T で区切ります。

timefmt2

日付の後に続く時間のフォーマットです。時間と日付は空白で区切ります。時間構成要素間はコロン (:) で区切ります。時間のみを使用するフォーマットと異なり、日付フォーマットに続く時間フォーマットは最大 2 文字で構成されます。最初の文字が表示する時間構成要素のすべてを表し、2 つ目の文字が AM/PM オプションを表します。詳細は、421 ページの「[日付時間フィールドの時間構成要素に使用する表示オプション](#)」を参照してください。

参照 **日付時間フィールドの日付構成要素に使用する表示オプション**

下表は、日付フォーマットに使用可能な表示オプションの有効な組み合わせを示しています。ここでは、日付として February 5, 1999 を想定します。

オプション	説明	例
Y	2 桁の年	99
YY	4 桁の年	1999

オプション	説明	例
M	2桁の月 (01 から 12 まで)	02
MT	月の完全名	February
Mt	月の短縮名	Feb
D	2桁の日	05
d	ゼロサプレス (ゼロを非表示) を使用した日ゼロは、ブランクスペースで置き換えられます。	5
e	ゼロを削除した日。日付の数字およびそれよりも右側に存在する構成要素は、左側に移動されます。 日付区切り文字が必要です。	5
o	ゼロを削除した月。ゼロを削除した日の e オプションが、自動的に実装されます。月および日の数字、およびそれよりも右側に存在する構成要素は、左側に移動されます。 日付区切り文字が必要です。	5
k	月または日の後に年が続き、月を短縮名または完全名に変換するフォーマットを使用する場合は、k を指定してカンマ (,) およびブランクで年と日を区切ります。それ以外の場合、区切り文字はブランクになります。	USAGE = HMtDkYY prints Feb 05, 1999

注意：AM/PM 時間表示オプションのいずれかを指定しない限り、時間構成要素には 24 時間制が使用されます。

例 **日付時間、月、日付を表す数値でのゼロ削除の使用**

次のリクエストは、「01/01/2013」という日付時間値を作成します。次に、この値を次のフォーマットで表示します。

- 月と日付の通常の表示 - HMDYY フォーマット
- 月のゼロ削除、日付のゼロ削除 - HoeYY フォーマット
- 月のゼロ削除、日付のゼロ削除なし (日付にゼロ削除が強制的に適用される) - HodYY フォーマット
- 日付のゼロ削除、月のゼロ削除なし - HMeYY フォーマット。この場合、日付のゼロ削除により、月のゼロ削除は強制的に適用されません。

```
DEFINE FILE GGSALES
DATE1A/HMDYY = DT(01/01/2013);
DATE1B/HoeYY = DATE1A;
DATE1C/HodYY = DATE1A;
DATE1D/HMeYY = DATE1A;
END
TABLE FILE GGSALES
SUM DOLLARS NOPRINT
DATE1A AS 'HMDYY'
DATE1B AS 'HoeYY'
DATE1C AS 'HodYY'
DATE1D AS 'HMeYY'
ON TABLE SET PAGE NOPAGE
END
```

出力結果は次のとおりです。

HMDYY	HoeYY	HodYY	HMeYY
-----	-----	-----	-----
01/01/2013	1/1/2013	1/1/2013	01/1/2013

例 **ゼロサプレスとゼロ削除の比較**

次のリクエストは、日付時間フォーマットで2つの日付を作成します。この日構成要素には、リーディングゼロ (01) が含まれています。1つ目の日付では、日構成要素が先頭の構成要素で、左端に表示されます。2つ目の日付では、日構成要素が2番目の構成要素で、中央に表示されます。このリクエストは、これらの日付を次のフォーマットで表示します。

- すべてのゼロを表示 - HDMYY フォーマット
- 日構成要素にゼロサプレスを使用 - HdMYYY フォーマット

□ 日構成要素にゼロ削除を使用 - HeMYY フォーマット

```

DEFINE FILE GGSales
DATE1A/HDMYY = DT(01/12/2012);
DATE2A/HMDYY = DT(12/01/2012);
DATE1B/HdMYy = DATE1A;
DATE2B/HMdYY = DATE2A;
DATE1C/HeMYy = DATE1A;
DATE2C/HMeYY = DATE2A;
END
TABLE FILE GGSales
SUM DOLLARS NOPRINT
DATE1A AS 'HDMYY'
DATE2A AS '' OVER
DATE1B AS 'HdMYy'
DATE2B AS '' OVER
DATE1C AS 'HeMYy'
DATE2C AS ''
ON TABLE SET PAGE NOPAGE

```

出力結果では、1行目のすべての日付に0(ゼロ)が表示されています。2行目では、日付を表す数値にゼロサプレスが適用されています。ここでは、0(ゼロ)がブランクに置き換えられ、すべての構成要素の位置が、1行目の構成要素の位置と一致しています。3行目では、ゼロ削除が適用されています。日付を表す数値から0(ゼロ)が削除され、残りの数値がすべて左側へ移動しています。

```

HDMYY 01/12/2012 12/01/2012
HdMYy 1/12/2012 12/ 1/2012
HeMYy 1/12/2012 12/1/2012

```

参照

日付時間フィールドの時間構成要素に使用する表示オプション

下表は、有効な表示オプションの一覧です。ここでは、日付および時間をそれぞれ February 5, 1999 および 02:05:25.444555333 a.m. と想定します。

オプション	説明	例
H	時間を表示します。	USAGE = HYYMDH は 1999/02/05 02 を表示します。
I	時間：分を表示します。	USAGE = HYYMDI は 1999/02/05 02:05 を表示します。

オプション	説明	例
S[x]	時間：分：秒を表示します。xオプションを指定する場合 (x は 1 から 9 の数字)、秒は x 桁で表示されます。そのほかに、s、m、n オプションを指定して、3、6、9 桁の秒を表示することもできます。	USAGE = HYYMDS は 1999/02/05 02:05:25 を表示します。 USAGE = HYYMDS1 は 1999/02/05 02:05:25.4 を表示します。
s	時間：分：秒：ミリ秒を表示します。	USAGE = HYYMDS は 1999/02/05 02:05:25.444 を表示します。
m	時間：分：秒：ミリ秒：マイクロ秒を表示します。	USAGE = HYYMDm は 1999/02/05 02:05:25.444555 を表示します。
n	時間：分：秒：ナノ秒を表示します。	USAGE = HYYMDn は 1999/02/05 02:05:25.444555333 を表示します。
A	AM または PM を表示します。このオプションでは 12 時間制が使用され、時間をゼロサプレスで表示します (ゼロを非表示)。	USAGE = HYYMDSA は 1999/02/05 2:05:25AM を表示します。
a	am または pm を表示します。このオプションでは 12 時間制が使用され、時間をゼロサプレスで表示します (ゼロを非表示)。	USAGE = HYYMDSa は 1999/02/05 2:05:25am を表示します。
B	1 つの空白に続けて AM または PM を表示します。このオプションでは 12 時間制が使用され、時間をゼロサプレスで表示します (ゼロを非表示)。	USAGE = HYYMDSB は 1999/02/05 2:05:25 AM を表示します。
b	1 つの空白に続けて am または pm を表示します。このオプションでは 12 時間制が使用され、時間をゼロサプレスで表示します (ゼロを非表示)。	USAGE = HYYMDSb は 1999/02/05 2:05:25 am を表示します。

オプション	説明	例
Z	ユニバーサル時間を示す Z を表示します。このオプションでは 12 時間制が使用されます。Z を AM/PM とともに指定することはできません。	USAGE = HHISZ は 14:30[:20.99]Z を表示します。

日付構成要素で指定可能な組み合わせおよび順序は次のとおりです。

- 先頭に年を指定する組み合わせは Y、YY、YM、YYM、YMD、YYMD です。
- 先頭に月を指定する組み合わせは M、MD、MY、MYY、MDY、MDYY です。
- 先頭に日を指定する組み合わせは D、DM、DMY、DMYY です。

参照 **日付時間フォーマット使用時の注意**

- 時間構成要素を使用するには、日付構成要素を指定する必要があります。
- k オプションを使用する場合は、日付区切り文字を変更することはできません。

文字フォーマット AnV

手順/構文

マスターファイルでの AnV フィールドの指定

参照

AnV フォーマット使用時の注意

HOLD ファイルへの AnV フィールドの継承

文字フォーマット AnV は、FOCUS、XFOCUS、リレーショナルデータソースのマスターファイルでサポートされます。このフォーマットは、リレーショナルデータベース管理システムによりサポートされる VARCHAR (可変長文字) データタイプを表すために使用します。

リレーショナルデータソースでは、AnV は VARCHAR フィールドの実際の長さの情報を保持します。この情報は、異なる RDBMS の VARCHAR フィールドに値を入力する際に重要です。この情報は、文字列連結時に末尾の空白を保持するかどうかに影響します。また Oracle の場合は、文字列の比較に影響します。なお、Oracle 以外のリレーショナルエンジンでは、文字列連結の末尾の空白は無視されます。

FOCUS、XFOCUS データソースでは、AnV は、実際の変長文字列サポートを提供するわけではありません。これは、フィールドに格納されたデータの実際の長さを保持するために、2つの先行バイトが追加された固定長文字フィールドです。この長さは、2バイトを占有する短整数値として格納されます。AnV フィールドの一部として入力した末尾の空白もフィールドの長さとして計上されます。

注意：2 バイトのオーバーヘッドおよびそれらの削除に必要な追加処理のために、非リレーショナルデータソースで AnV フォーマットは使用しないことをお勧めします。

構文

マスターファイルでの AnV フィールドの指定

```
FIELD=name, ALIAS=alias, USAGE=AnV [,ACTUAL=AnV] , $
```

説明

n

フィールドのサイズ (最大長) です。1 から 4093 までの値を指定することができます。なお、この長さの格納には追加の 2 バイトが使用されるため、実際の A4093V フィールドの長さは 4095 バイトになります。サイズゼロ (0、AOV) はサポートされません。フィールドのインスタンスの長さは 0 (ゼロ) にすることができます。

注意：HOLD FORMAT ALPHA は、マスターファイル内に AnW の ACTUAL フォーマットを作成します。詳細は、426 ページの「[HOLD ファイルへの AnV フィールドの継承](#)」を参照してください。

例

マスターファイルでの AnV フォーマットの指定

次の例は、サイズが 200 の DB2 データソースのマスターファイルで指定した VARCHAR フィールドを示しています。

```
$ VARCHAR FIELD USING AnV
  FIELD=VARCHAR200, ALIAS=VC200, USAGE=A200V, ACTUAL=A200V, MISSING=ON , $
```

次の例は、サイズが 200 の FOCUS データソースのマスターファイルで指定した AnV フィールドを示しています。

```
FIELD=ALPHAV, ALIAS=AV200, USAGE=A200V, MISSING=ON , $
```


データソースに AnV フィールドがある場合は、次のように記述して長さを指定せずに HOLD FORMAT ALPHA ファイルを作成します。

```
FIELD=ALPHA, USAGE=A25, ACTUAL=A25V, $
```

または

```
DEFINE ...
  ALPHA/A25 = VARCHAR ;
END
```

または

```
COMPUTE ALPHA/A25 = VARCHAR ;
```

マスターファイルを変更または作成して AnV を追加するには、データを変換して、フィールドの先頭にその長さを追加する必要があります。たとえば、次のようにフィールドを記述する場合に HOLD コマンドを発行します。

```
FIELD=VARCHAR, ,USAGE=A25V, ACTUAL=A25, $
```

または

```
DEFINE ...
  VARCHAR/A25V = ALPHA ;
END
```

または

```
COMPUTE VARCHAR/A25V = ALPHA ;
```

参照

AnV フォーマット使用時の注意

- ここに記載された制限を除いて、AnV は An が使用可能な任意の場所で使用することができます。
- このデータタイプでは、リレーショナルデータソースの CREATE FILE を含めて、FOCUS および SQL のすべての演算がサポートされます。
- JOIN は、An と AnV フィールドの間ではサポートされません。
- DBCS 文字はサポートされます。An フォーマットと同様に、文字数は 4 キロバイトデータ領域内に収める必要があります。
- COMPUTE および DEFINE は、左辺で指定されたデータタイプを生成します。
- AnV と TX フィールド間の変換はサポートされません。
- AnV フィールドに日付表示オプションを含めることはできません。

参照 **HOLD** ファイルへの **AnV** フィールドの継承

HOLD FORMAT ALPHA コマンドを使用して **AnV** フィールドをシーケンシャルデータソースに継承する場合、2 バイトの整数長は 6 桁の文字長に変換されます。**HOLD** ファイルのフィールドは、文字データが続くこの 6 桁の数で構成されます。このフィールドのフォーマット属性は次のとおりです。

```
... USAGE=AnV, ACTUAL=AnW
```

AnW は、**HOLD FORMAT ALPHA** の結果として生成され、必要に応じて読み取りおよび入力に使用することができます。**HOLD** ファイルで、このフィールドが占有するバイト数は $6+n$ です。

例 **HOLD** ファイルへの **AnV** フィールドの継承

たとえば、「TITLEV」という **A39V** フィールドは、次のように **HOLD** ファイルに継承されます。

```
FIELDNAME = TITLEV ,E03 ,A39V ,A39W ,$
```

バイナリ **HOLD** ファイルでは **ACTUAL** フォーマットには完全な 4 バイトワードになるまで空白が追加されますが、**USAGE** および **ACTUAL** フォーマットは **AnV** です。**HOLD** ファイルで、このフィールドが占有するバイト数は $2+n$ です。

AnV フィールドをデータソースに入力する場合、設定した長さを超えて入力したすべてのバイトは無視されます。これらのバイトは入力処理の一部として空白に設定されます。

HOLD FORMAT sqlengine コマンドを使用してリレーショナルデータソースを作成すると、**AnV** フィールドによりそのリレーショナルデータソース内に **VARCHAR** フィールドが生成されます。

たとえば、名前が **TITLEV** の **A39V** フィールドは、**HOLD FORMAT DB2** ファイルに次のように継承されます。

```
FIELDNAME = 'TITLEV', 'TITLEV', A39V, A39V ,$
```

テキストフィールドのフォーマット

手順/構文

マスターファイルでのテキストフィールドの指定

任意のテキストの組み合わせを、テキストフィールドとして格納することができます。

構文 マスターファイルでのテキストフィールドの指定

```
FIELD = fieldname, ALIAS = aliasname, USAGE = TXn[F], $
```

説明

fieldname

テキストフィールドに割り当てる名前です。

aliasname

フィールド名の別名です。

n

TABLE 内で、テキストフィールドの出力表示の長さです。表示する長さの範囲は 1 から 256 バイトです。

すべての文字、数字、特殊文字は、このフォーマットで格納することができます。下表は、テキストフィールドフォーマットのサンプルを示しています。

フォーマット	表示
TX50	This course provides the DP professional with the skills needed to create, maintain, and report from FOCUS data sources.
TX35	This course provides the DP professional with the skills needed to create, maintain, and report from FOCUS data sources.

テキストフィールドフォーマットでは、標準の編集オプションを使用することはできません。

参照 テキストフィールドフォーマット使用時の注意

- ❑ DEFINE コマンドおよび COMPUTE コマンドでは、テキストフィールドと文字フィールド間の変換がサポートされます。
- ❑ セグメントの任意の位置での複数テキストフィールドがサポートされます。

格納データタイプ - ACTUAL

トピックス

ACTUAL 属性

ACTUAL 属性は、データを実際にデータソースに格納する際のデータタイプおよび長さを記述します。文字データタイプなどのいくつかのデータタイプがユニバーサルであるのに対し、他のデータタイプはデータソースのタイプにより異なります。ユニークデータタイプをサポートするデータソースもあります。そのため、ACTUAL 属性に割り当て可能な値は、データソースのタイプにより異なります。

ACTUAL 属性

手順/構文

ACTUAL 属性の指定

参照

ACTUAL から USAGE への変換

COBOL PICTURE から USAGE フォーマットへの変換

この属性は、データが実際のデータソースに存在する際のデータのタイプおよび長さを記述します。この情報は、データソースの現在の記述に基づきます (例、COBOL FD ステートメント)。ACTUAL 属性は、FOCUS 以外のデータソースのマスターファイルに記述する特別な特性の 1 つです。この属性は FOCUS 以外のデータ構造のフォーマットを記述する場合にのみ使用するため、FOCUS データ構造のマスターファイルでは使用されません。

データソースに日付が文字フィールドとして格納され、レポートでソートや集計の目的での日付への変換が必要な場合は、マスターファイルで DATEPATTERN 属性を使用することができます。は、指定されたパターンを使用して、文字日付を の日付に変換します。

構文

ACTUAL 属性の指定

ACTUAL = *format*

説明

format

下表のいずれかの値で構成されます。この表には、読み取り可能なデータタイプのコードが示されています。

ACTUAL タイプ	説明
DATE	入力する日付と日付フォーマットの基準日との差を表す内部的な 4 バイトの整数フォーマットです。
An	ここで、 <i>n</i> の値は固定フォーマットシーケンシャルおよび VSAM データソースでは 1 から 4095、FOCUS 以外のその他のデータソースでは 1 から 256 です。A から Z、0 から 9 の英数文字、および EBCDIC 表示モードの特殊文字です。 An は、すべての日付時間文字列フォーマットおよび Hn 表示フォーマットを受容します。ACTUAL=An は、文字 HOLD ファイルまたは SAVE ファイルに存在する日付時間フィールドも受容します。
D8	内部的に 8 バイトで格納された倍精度浮動小数点数です。
F4	内部的に 4 バイトで格納された単精度浮動小数点数です。
Hn	H8、H10、H12 は、バイナリ HOLD ファイルまたは SAVB ファイルに存在する日付時間フィールドを受容します。
In	バイナリ整数は次のとおりです。 I1 = 1 バイトのバイナリ整数。 I2 = 2 バイトのバイナリ整数。 I4 = 4 バイトのバイナリ整数。 I8 = 8 バイトのバイナリ整数。 注意： USAGE は P または D を指定する必要があります。小数点数の指定は、USAGE の P または D に従います。

ACTUAL タイプ	説明
Pn	<p>ここで、$n = 1-16$。パック 10 進数の内部フォーマット。n はバイト数を表し、末尾のバイトに格納する 1 桁と符号 (+ または -) を除いて、それぞれのバイトに 2 桁ずつ格納されます。たとえば、P6 は 11 桁と 1 つの符号を表します。</p>
Zn	<p>ここで、n は 1 から 31 です。ゾーン 10 進数の内部フォーマットです。n は桁数を表し、それぞれ 1 バイトの格納領域を使用します。末尾の桁には、1 桁と符号が格納されます。</p> <p>フィールドに想定小数点が含まれる場合は、Zn の ACTUAL フォーマットおよび Pm.d の USAGE フォーマットを使用してフィールドを表します。ここで、m は表示する桁数と想定小数点の合計、d は小数部の桁数を表します。また、m には n より少なくとも 1 つ大きい値を指定する必要があります。たとえば、ACTUAL=Z5 で小数点以下 1 桁のフィールドには、USAGE=P6.1 または P7.1 以上が必要です。</p>

注意

- データソースをプログラムで作成した場合を除いて、すべての文字はタイプ A (文字)、タイプ Z (ゾーン 10 進数) のいずれかになります。
- 日付時間値でサポートされる ACTUAL フォーマットは、An、H8、H10、H12 です。An は、すべての日付時間文字列および Hn USAGE 表示フォーマットを受容します。ACTUAL=H8、H10、H12 は、バイナリ HOLD ファイルまたは SAVB ファイルに存在する日付時間フィールドを受容します。ACTUAL=An は、文字 HOLD ファイルまたは SAVE ファイルに存在する日付時間フィールドも受容します。
- 日付時間フィールドを含むデータソースからバイナリ HOLD ファイルを作成すると、そのフィールドの ACTUAL フォーマットは Hn になります。日付時間フィールドを含むデータソースから文字 HOLD ファイルを作成すると、そのフィールドの ACTUAL フォーマットは An になります。

参照

ACTUAL から USAGE への変換

次の ACTUAL フォーマットから USAGE (表示) フォーマットへの変換は、自動的に処理されるため、関数を呼び出す必要はありません。

ACTUAL	USAGE
A	A, D, F, I, P, date format, date-time format
D	D
DATE	date format
F	F
H	H
I	I, date format
P	P, date format
Z	D, F, I, P

参照

COBOL PICTURE から USAGE フォーマットへの変換

下表は、COBOL、FORTRAN、PL1 Assembler のフィールド記述に使用可能な USAGE および ACTUAL フォーマットを示しています。

COBOL USAGE FORMAT	COBOL PICTURE のバイト	INTERNAL STORAGE	ACTUAL FORMAT	USAGE FORMAT
DISPLAY	X(4)	4	A4	A4
DISPLAY	S99	2	Z2	P3
DISPLAY	9(5)V9	6	Z6.1	P8.1
DISPLAY	99	2	A2	A2
COMP	S9	4	I2	I1
COMP	S9(4)	4	I2	I4
COMP*	S9(5)	4	I4	I5
COMP	S9(9)	4	I4	I9
COMP-1**	—	4	F4	F6
COMP-2***	—	8	D8	D15
COMP-3	9	8	P1	P1
COMP-3	S9V99	8	P2	P5.2
COMP-3	9(4)V9(3)	8	P4	P8.3
FIXED BINARY(7) (COMP-4)	B or XL1	8	I4	I7

* FORTRAN の INTEGER、PL/1 の FIXED BINARY(31)、Assembler の F と同等です。

** FORTRAN の REAL、PL/1 の FLOAT(6)、Assembler の E と同等です。

*** FORTRAN の DOUBLE PRECISION または REAL*8、PL/1 の FLOAT(16)、Assembler の D と同等です。

注意

1. ここでは USAGE の長さの最小値が示されています。必要に応じてこの値を大きくすることができます。また、他の編集オプションを追加することもできます。
2. USAGE フォーマットで負の値の使用が予想される場合は、マイナス符号 (-) を配置する場所が別途必要になります。
3. PICTURE 句を内部的な浮動点項目に使用することはできません。
4. USAGE の長さには、予想される最大桁数を確保する必要があります。
5. USAGE フォーマットでは、小数点を配置する場所が別途必要になります。

なお、FOCUS データソースは ACTUAL 属性を使用せず、代わりに USAGE 属性を使用してフィールドの格納およびフォーマット方法を指定します。

ミッシング値 (Null 値) - MISSING

トピックス

ミッシング値の使用

手順/構文

ミッシング値の指定

参照

MISSING 属性使用時の注意

セグメントインスタンスは存在するが、セグメント内のフィールドのいずれかにデータが入力されていない場合、そのフィールドは値が欠落した状態になります。いくつかのデータソースタイプではデータの欠落をブランクまたは 0 (ゼロ) として表示しますが、他のデータソースタイプではこのデータの欠落を Null インジケータまたは特別な Null 値として明示的に指定します。Null 値は「ミッシングデータ」とも呼ばれ、レポートアプリケーションで平均値の計算のような集計関数を実行する場合に重要です。

FOCUS データソースおよびほとんどのリレーショナルデータソースのように、ミッシングデータをサポートするデータソースタイプを使用する場合は、オプションの MISSING 属性を使用してフィールドに Null 値を入力したり、フィールドから Null 値を読み取ったりすることができます。次のような場合に MISSING 属性を使用すると役立ちます。

- **新しいセグメントインスタンスを作成する場合** マスターファイルあるいは一時項目 (DEFINE または COMPUTE) で MISSING を ON に設定したフィールドの値が指定されない場合、そのフィールドにはミッシング値が割り当てられます。
- **レポートを生成する場合** Null 値を含むフィールドを取得する場合、そのフィールド値は平均や合計などの集計計算に使用されません。このフィールドの値を呼び出してレポートに表示すると、ミッシング値を示す特殊文字が表示されます。デフォルト文字はピリオド (.) ですが、SET NODATA コマンドを使用して任意の文字に変更することができます。詳細は、

構文

ミッシング値の指定

```
MISSING = {ON|OFF}
```

説明

ON

新しいセグメントインスタンスの作成時およびレポートの実行時に、意図的に入力したブランクまたは 0 (ゼロ) とミッシング値を区別します。

OFF

新しいセグメントインスタンスの作成時およびレポートの実行時に、ブランクまたは 0 (ゼロ) とミッシング値を区別しません。デフォルト値は OFF です。

参照

MISSING 属性使用時の注意

MISSING 属性の使用時には次の規則が適用されます。

- **エイリアス** MISSING 属性にはエイリアスはありません。
- **値** フィールドに Null 属性が事前定義されている場合は、MISSING 属性をその属性に一致させることをお勧めします。つまり、データソースの作成時にこの特性が明示的に設定されているか、デフォルトの状態を設定されているかなどです。たとえば、リレーショナルテーブルフィールドがミッシングデータを受容するように作成されている場合は、このフィールドの MISSING 属性を ON に設定してミッシング値を正しく解釈するように記述します。

FOCUS データソースでは MISSING=ON もサポートされます。この設定にすると、数値フィールドでは NULL 値に「-9998998」が割り当てられ、文字フィールドでは「」が割り当てられます。

- **変更** MISSING 属性はいつでも変更することができます。なお、MISSING 属性を変更しても、以前の設定で入力して実際に格納されているデータ値には影響を与えません。ただし、データの解釈方法には影響します。MISSING=ON の設定でミッシングデータを入力した後、MISSING=OFF の設定に変更すると、最初にミッシング値として入力したデータはブランク (文字フィールド) または 0 (ゼロ) (数値フィールド) として解釈されます。

ミッシング値の使用

ここでは、次の 4 つのレコードで示されたフィールド値について考察します。

		1	3
--	--	---	---

フィールド宣言に MISSING 属性を指定せずにミッシング値の平均を計算すると、2 つのブランクレコードに対して 0 (ゼロ) の値が自動的に入力されます。そのため、この 4 つのレコードの平均は $(0+0+1+3)/4$ で 1 になります。MISSING 属性を ON に設定すると、この 2 つのブランクレコードが計算に使用されないため、平均は $(1+3)/2$ で 2 になります。

ユニークセグメントのミッシング値の場合でも、MISSING 属性の設定に基づいて 0 (ゼロ)、ブランク、ミッシング値のいずれかが自動的に入力されます。ユニークセグメントのミッシング値が他の値と異なる点は、ミッシング値が格納されないことです。個数または平均値を計算するユニークセグメントのフィールドには MISSING 属性を指定する必要はありません。

レポートに Null 値 (ミッシングデータ) を使用方法についての詳細は、を参照してください。このマニュアルでは、WHERE 句に MISSING 選択演算子を使用したり、DEFINE FILE コマンドに SOME または ALL 句を使用して一時項目 (DEFINE) を作成したりして、レポートでこれらの値を区別する別の方法について説明しています。

FML 階層の記述

手順/構文

マスターファイルでのフィールド間の階層の指定

階層フィールド値への説明キャプションの割り当て

FML (Financial Modeling Language) では、階層データ構造に対する動的なレポート作成がサポートされます。

FML を使用して、マスターファイルのフィールド間に階層関係を定義し、これらのフィールドを自動的に表示することができます。また、指定した階層フィールドの値の代わりに、説明キャプションをレポートに表示することもできます。

2つのフィールド間に階層関係を定義するには、マスターファイルで `PROPERTY=PARENT_OF` および `REFERENCE=hierarchyfld` 属性を指定します。

親フィールドおよび子フィールドには、同一の `FORMAT` または `USAGE` を指定し、さらに階層関係を指定する必要があります。親フィールドおよび子フィールドの両方のフォーマットは、数値、文字のいずれかに設定する必要があります。

構文

マスターファイルでのフィールド間の階層の指定

```
FIELD=parentfield,...,PROPERTY=PARENT_OF, REFERENCE=[seg.]hierarchyfld, $
```

説明

parentfield

階層の親フィールドです。

`PROPERTY=PARENT_OF`

このフィールドを、階層内で参照されるフィールドの親として指定します。

これらの属性は、フィールドごとに指定することができます。そのため、1つのマスターファイルに複数の階層を定義することができます。ただし、各フィールドに指定できる親はそれぞれ1つです。同一階層フィールドの `CAPTION` 属性が複数のフィールドで指定されている場合は、その階層構造を上から下、左から右に検索して最初に見つかった親がこの場合の親として使用されます。

seg

階層フィールドのセグメントの位置です。 `hierarchyfield` フィールドが複数のセグメントに存在する場合に必要です。

hierarchyfld

階層の子フィールドです。

PARENT_OF は、マスターファイルの一時項目 (DEFINE) にも使用することができます。

```
DEFINE name/fmt=expression; ,PROPERTY=PARENT_OF,REFERENCE=hierarchyfld,$
```

構文

階層フィールド値への説明キャプションの割り当て

マスターファイルで階層フィールドにキャプションを割り当てるには、次の属性を使用します。

```
FIELD=captionfield,..., PROPERTY=CAPTION, REFERENCE=[seg.]hierarchyfld, $
```

説明

captionfield

階層フィールドの説明キャプションを追加するフィールド名です。たとえば、従業員 ID が階層フィールドの場合、従業員 ID の代わりに従業員の名前 (姓) を説明テキストとしてレポートに表示することができます。

PROPERTY=CAPTION

このフィールドに、階層フィールド値の代わりにレポートに表示する説明キャプションが指定されていることを示します。

キャプションはフィールドごとに指定することができますが、各フィールドに追加できるキャプションは 1 つです。同一階層フィールドの CAPTION 属性が複数のフィールドで指定されている場合は、その階層構造を上から下、左から右に検索して最初に見つかった親がキャプションとして使用されます。

seg

階層フィールドのセグメントの位置です。hierarchyfield フィールドが複数のセグメントに存在する場合に必要です。

hierarchyfld

階層フィールドです。

CAPTION は、マスターファイルの一時項目 (DEFINE) にも使用することができます。

```
DEFINE name/format=expression; ,PROPERTY=CAPTION,REFERENCE=hierarchyfld,$
```

例 マスターファイルでの階層の定義

CENTGLマスターファイルには、勘定科目表の階層が格納されています。GL_ACCOUNT_PARENT は、階層の親フィールドです。GL_ACCOUNT フィールドは、階層フィールドです。次の GL_ACCOUNT_CAPTION フィールドは、階層フィールドの説明キャプションとして使用することができます。

```
FILE=CENTGL          ,SUFFIX=FOC
SEGNAME=ACCOUNTS,SEGTYPE=S01
FIELDNAME=GL_ACCOUNT,          ALIAS=GLACCT,   FORMAT=A7,
                                TITLE='Ledger,Account', FIELDTYPE=I, $
FIELDNAME=GL_ACCOUNT_PARENT,   ALIAS=GLPAR,   FORMAT=A7,
                                TITLE=Parent,
                                PROPERTY=PARENT_OF, REFERENCE=GL_ACCOUNT, $
FIELDNAME=GL_ACCOUNT_TYPE,     ALIAS=GLTYPE,  FORMAT=A1,
                                TITLE=Type,$
FIELDNAME=GL_ROLLUP_OP,        ALIAS=GLROLL,  FORMAT=A1,
                                TITLE=Op, $
FIELDNAME=GL_ACCOUNT_LEVEL,    ALIAS=GLLEVEL, FORMAT=I3,
                                TITLE=Lev, $
FIELDNAME=GL_ACCOUNT_CAPTION,  ALIAS=GLCAP,   FORMAT=A30,
                                TITLE=Caption,
                                PROPERTY=CAPTION,   REFERENCE=GL_ACCOUNT, $
FIELDNAME=SYS_ACCOUNT,        ALIAS=ALINE,   FORMAT=A6,
                                TITLE='System,Account,Line', MISSING=ON, $
```

データの確認 - ACCEPT

手順/構文

データの確認

参照

ACCEPT 属性使用時の注意

オプションとして指定する ACCEPT 属性は、MODIFY 処理からフィールドに入力されるデータの確認に使用します。ACCEPT テストは、FIXFORM の処理が完了した際に即時に適用されます。その後、後続く COMPUTE ステートメントで値を操作することができます。フィールド宣言に ACCEPT 属性を指定して、許容フィールド値のリストまたは範囲を定義することができます。リレーショナル用語では、ドメインを定義することを意味します。

構文 データの確認

```
ACCEPT = list
ACCEPT = value1 TO value2
ACCEPT = FIND (field [AS name] IN file)
```

説明

list

許容値のリストです。構文は次のとおりです。

```
value1 OR value2 OR value3...
```

たとえば、ACCEPT = RED OR WHITE OR BLUE のように記述します。項目の区切り文字として空白を使用することもできます。許容値のリストが1行に収まらない場合は、そのまま次の行に続けます。リストはカンマ (,) で終了します。

value1 TO value2

許容値の範囲を設定します。たとえば、ACCEPT = 150 TO 1000 のように記述します。

FIND

他のインデックスフィールドの値に対して、入力データを確認します。このオプションは FOCUS データソースにのみ使用可能です。

ACCEPT 属性に埋め込み空白が含まれる場合 (例、Great Britain)、その値は一重引用符 (') で囲む必要があります。

フィールド宣言で ACCEPT 属性を指定し、SET コマンドパラメータである ACCBLN の値が OFF の場合、空白および 0 (ゼロ) の値は ACCEPT 属性に明示的にコーディングされている場合のみ使用することができます。SET ACCBLN についての詳細は、

例 埋め込み空白を含むリストの指定

```
ACCEPT = SPAIN OR ITALY OR FRANCE OR 'GREAT BRITAIN'
```

参照 ACCEPT 属性使用時の注意

ACCEPT 属性の使用時には次の規則が適用されます。

- **エイリアス** ACCEPT 属性にはエイリアスはありません。
- **変更** ACCEPT 属性の情報はいつでも変更することができます。
- **一時項目 (DEFINE)** ACCEPT 属性を使用して、DEFINE 属性で作成した一時項目を確認することはできません。

- **HOLD ファイル** HOLD ファイルのマスターファイルに ACCEPT 属性を継承するには、SET HOLDATTR コマンドを使用します。HOLD ファイルについての詳細は、レポート言語のマニュアルを参照してください。
- ACCEPT 属性は MODIFY 処理でのみ使用します。複数のプロシジャで共用する確認リストを1つ作成しておく役立ちます。FIND 関数は、値のリストが大規模な場合や変更頻度が高い場合に役立ちます。

代替レポートフィールドタイトル - TITLE

手順/構文

代替タイトルの指定

参照

TITLE 属性使用時の注意

レポートを作成すると、デフォルト設定でレポートの各フィールドタイトルにはフィールド名が表示されます。フィールドの TITLE 属性オプションを指定して、デフォルトのフィールドタイトルを変更することができます。

また、レポートリクエストに AS 句を使用して、レポートごとに異なるフィールドタイトルを指定することもできます。詳細は、

なお、AVE. などの演算接頭語をフィールドに使用した場合、TITLE 属性の指定はレポートに反映されません。演算接頭語をフィールドに使用した場合は、AS 句を使用して代替フィールドタイトルを指定することができます。

マスターファイルは TITLE 属性を多言語でサポートします。詳細は、443 ページの「[多言語メタデータ](#)」を参照してください。

構文

代替タイトルの指定

```
TITLE = 'text'
```

説明

テキスト

最大で 512 バイトの任意の文字列です。テキストをカンマ (,) で区切ることにより、最大で5つのタイトル行に分割することができます。フィールドタイトルの末尾にブランクを含める場合は、そのブランクの位置にスラッシュ (/) を配置します。カンマ (,) または先頭にブランクを含む文字列は、一重引用符 (') で囲む必要があります。

例 **デフォルトフィールドタイトルの置換**

たとえば、次のように FIELD 宣言を記述します。

```
FIELD = LNAME, ALIAS = LN, USAGE = A15, TITLE = 'Client,Name', $
```

これにより、デフォルトフィールドタイトルの LNAME が次のタイトルに置換されます。

```
Client
Name
-----
```

参照 **TITLE 属性使用時の注意**

TITLE 属性の使用時には次の規則が適用されます。

- **エイリアス** TITLE 属性にはエイリアスはありません。
- **変更** TITLE 属性の情報はいつでも変更することができます。TITLE 属性を上書きするには、リクエストで AS 句を指定するか、SET TITLES=OFF コマンドを使用して TITLE 属性をオフにします。
- **一時項目 (DEFINE)** DEFINE 属性で作成した一時項目に TITLE 属性を使用する場合は、DEFINE 式を終了するセミコロン (;) を TITLE キーワードと同一行に配置する必要があります。
- **HOLD ファイル** HOLD ファイルのマスターファイル内に TITLE 属性を継承するには、SET HOLDATTR コマンドを使用します。HOLD ファイルについての詳細は、レポート言語のマニュアルを参照してください。

フィールドの説明 - DESCRIPTION**手順/構文**

フィールド説明の指定

参照

DESCRIPTION 属性使用時の注意

オプションの DESCRIPTION 属性を使用して、マスターファイル内のフィールドにコメントおよび他の説明を追加することができます。追加するコメントは、最大で 2 キロバイト (2048 バイト) です。

なお、フィールド宣言、セグメント宣言、ファイル宣言にもコメントを追加することができます。その場合は、フィールドにコメントを入力し、終了を表すドル記号(\$)を配置します。また、コメント専用の行を作成するには、宣言の後に新しい行を挿入し、その行の先頭にドル記号(\$)を配置します。

FOCUS データソースに使用する DESCRIPTION 属性は、データソースを再構築せずにも変更することができます。

マスターファイルでは、多言語の DESCRIPTION 属性がサポートされます。詳細は、443 ページの「[多言語メタデータ](#)」を参照してください。

構文 **フィールド説明の指定**

```
DESC[RIPTION] = text
```

説明

DESCRIPTION

短縮形の DESC を使用することもできます。キーワードを短縮しても、機能面での影響はありません。

テキスト

最大で 2 キロバイト (2048 バイト) の任意のテキストです。テキストにカンマ (,) が含まれる場合は、テキスト全体を一重引用符 (') で囲む必要があります。

例 **フィールド説明の指定**

次のように FIELD 宣言を記述して DESCRIPTION 属性を指定します。

```
FIELD=UNITS,ALIAS=QTY,USAGE=I6, DESC='QUANTITY SOLD, NOT RETURNED', $
```

参照 **DESCRIPTION 属性使用時の注意**

DESCRIPTION 属性の使用時には次の規則が適用されます。

- **エイリアス** DESCRIPTION 属性のエイリアスは DEFINITION です。
- **変更** DESCRIPTION 属性はいつでも変更することができます。
- **一時項目 (DEFINE)** DEFINE 属性で作成した一時項目に DESCRIPTION 属性を使用することができます。

多言語メタデータ

手順/構文

マスターファイルでの多言語メタデータの指定

言語使用の有効化

参照

言語および言語コードの略名

多言語メタデータ使用時の注意

マスターファイルは、多言語のフィールドタイトルをサポートします。使用されるタイトルまたは説明は、LANG パラメータの値およびマスターファイルで指定された TITLE_ *ln* または DESC_ *ln* 属性の有無により異なります。ここで、*ln* はフィールドタイトルまたは説明に適用される言語を識別します。

マスターファイルでは、フィールドタイトルは次の項目から取得されます。

1. AS 句を使用してレポートリクエストに指定されたタイトル
2. マスターファイルの TITLE 属性 (リクエストに AS 句が指定されておらず、SET TITLES=ON の場合)
3. マスターファイルで指定したフィールド名 (AS 句と TITLE 属性のいずれも指定されておらず、SET TITLES=OFF の場合)

構文

マスターファイルでの多言語メタデータの指定

```
FIELDNAME = field, ...
.
.
.
TITLE= default_column_headingTITLE_ ln = column_heading_for_ ln .
.
.
DESC= default_descDESC_ ln = desc_for_ ln .
.
.
```

説明

field

マスターファイル内のフィールドです。

default_column_heading

このフィールドタイトルは、SET TITLES=ON の設定で、LANG パラメータがサーバのデフォルト言語に設定されている場合、またはその他の言語に設定されているがマスターファイルのフィールドに対応する TITLE_In 属性がない場合に使用されます。このフィールドタイトルは、In 値が無効な場合にも使用されます。

default_desc

この説明テキストは、LANG パラメータがサーバのデフォルトの言語に設定されている場合、またはその他の言語に設定されているがマスターファイルのフィールドに対応する DESC_In 属性がない場合に使用されます。この説明テキストは、In 値が無効な場合にも使用されます。

TITLE_In = *column_heading_for_In*

フィールドタイトルに使用する言語、およびその言語でのフィールドタイトルのテキストを指定します。このフィールドタイトルは、SET TITLES=ON の設定で、LANG パラメータがサーバのデフォルト以外の言語に設定され、マスターファイルに対応する TITLE_In 属性がある場合に使用されます。ここで、In は LANG パラメータで指定された 2 バイトの ISO 639 言語コードの略名です。詳細は、444 ページの「[言語および言語コードの略名](#)」を参照してください。

DESC_In = *desc_for_In*

説明に使用する言語、およびその言語での説明テキストを指定します。この説明は、LANG パラメータがサーバのデフォルト言語以外の言語に設定され、マスターファイルに対応する DESC_In 属性がある場合に使用されます。In 有効な値は、2 バイトの ISO 639 言語コードの略名です。

参照

[言語および言語コードの略名](#)

言語名	2 文字の言語コード	3 文字の言語の略名
アラビア語	ar	ARB
バルト語	lt	BAL
中国語 (簡体字)	zh	PRC
中国語 (繁体字)	tw	ROC
チェコ語	cs	CZE

言語名	2文字の言語コード	3文字の言語の略名
デンマーク語	da	DAN
オランダ語	nl	DUT
英語 (米国)	en	AME または ENG
英語 (英国)	uk	UKE
フィンランド語	fi	FIN
フランス語 (カナダ)	fc	FRE
フランス語 (フランス)	fr	FRE
ドイツ語 (オーストリア)	at	GER
ドイツ語 (ドイツ)	de	GER
ギリシャ語	el	GRE
ヘブライ語	iw	HEW
イタリア語	it	ITA
日本語 (ASCII: Shift-JIS(cp942) EBCDIC: cp939)	ja	JPN
日本語 (UNIX では、ASCII: EUC(cp10942))	je	JPE
韓国語	ko	KOR
ノルウェー語	no	NOR
ポーランド語	pl	POL
ポルトガル語 (ブラジル)	br	POR
ポルトガル語 (ポルトガル)	pt	POR
ロシア語	ru	RUS

言語名	2文字の言語コード	3文字の言語の略名
スペイン語	es	SPA
スウェーデン語	sv	SWE
タイ語	th	THA
トルコ語	tr	TUR

構文

言語使用の有効化

サポートされるプロファイル、プロシ ज्याで次の構文を発行します。

```
SET LANG = lng
```

または

```
SET LANG = ln
```

説明

lng

言語の 3 文字の略名です。

ln

2 文字の ISO 言語コードです。

注意：SET LANG がプロシ ज्या内に使用されている場合、この値は nlscfg.err ファイルまたはすべてのプロファイルに設定されている値を上書きします。

参照

NLS 構成ファイルでの言語の有効化

構成ファイルで、次のコマンドを発行します。

```
LANG = lng
```

参照

多言語メタデータ使用時の注意

- ❑ 文字を正しく生成するには、使用するすべての言語が、指定されたコードページと一致する必要があります。
- ❑ マスターファイルは、格納しておく必要があります。

- 多言語の説明は、一時項目 (DEFINE および COMPUTE) を含むマスターファイルに記述されたすべてのフィールドでサポートされます。
- SET HOLDATTR=ON で HOLD コマンドを発行すると、HOLD マスターファイルには TITLE 属性が 1 つだけ継承されます。この値は、レポートの出力に表示されるフィールドタイトルです。

例

マスターファイルでの多言語説明の使用

次の例では、CENTINV データソースのマスターファイルで、PROD_NUM および PRODNAME フィールドの説明にフランス語 (DESC_FR)、スペイン語 (DESC_ES)、デフォルト言語 (DESC) が指定されています。

```
FILE=CENTINV, SUFFIX=FOC, FDFC=19, FYRT=00
SEGNAME=INVINFO, SEGTYPE=S1, $
FIELD=PROD_NUM, ALIAS=PNUM, FORMAT=A4, INDEX=I,
  DESCRIPTION='Product Number'
  DESC='Product Number',
  DESC_ES='Numero de Producto',
  DESC_FR='Nombre de Produit', $
FIELD=PRODNAME, ALIAS=PNAME, FORMAT=A30,
  WITHIN=PRODCAT,
  DESCRIPTION='Product Name'
  DESC_FR='Nom de Produit',
  DESC_ES='Nombre de Producto', $
FIELD=QTY_IN_STOCK, ALIAS=QIS, FORMAT=I7,
  DESCRIPTION='Quantity In Stock', $
FIELD=PRICE, ALIAS=RETAIL, FORMAT=D10.2,
  TITLE='Price:',
  DESCRIPTION=Price, $
```

例 リクエストでの複数言語タイトルの使用

次の例では、CENTINV データソースのマスターファイルで、PROD_NUM および PRODNAME フィールドのタイトルにフランス語 (TITLE_FR)、スペイン語 (TITLE_ES)、デフォルト言語 (TITLE) が指定されています。

```
FILE=CENTINV, SUFFIX=FOC, FDFC=19, FYRT=00
SEGNAME=INVINVO, SEGTYPE=S1, $
FIELD=PROD_NUM, ALIAS=PNUM, FORMAT=A4, INDEX=I,
  TITLE='Product,Number:',
  TITLE_FR='Nombre,de Produit:',
  TITLE_ES='Numero,de Producto:',
  DESCRIPTION='Product Number', $
FIELD=PRODNAME, ALIAS=PNAME, FORMAT=A30,
  WITHIN=PRODCAT,
  TITLE='Product,Name:',
  TITLE_FR='Nom,de Produit:',
  TITLE_ES='Nombre,de Producto:'
  DESCRIPTION='Product Name', $
FIELD=QTY_IN_STOCK, ALIAS=QIS, FORMAT=I7,
  TITLE='Quantity,In Stock:',
  DESCRIPTION='Quantity In Stock', $
FIELD=PRICE, ALIAS=RETAIL, FORMAT=D10.2,
  TITLE='Price:',
  DESCRIPTION=Price, $
```

そのため、次のリクエストは TITLE 属性を使用して、すべてのフィールドタイトルを英語で作成します。

```
TABLE FILE CENTINV
PRINT PROD_NUM PRODNAME PRICE
WHERE PRICE LT 200
END
```

出力結果は次のとおりです。

Product Number:	Product Name:	Price:
-----	-----	-----
1004	2 Hd VCR LCD Menu	179.00
1008	DVD Upgrade Unit for Cent. VCR	199.00
1026	AR3 35MM Camera 10 X	129.00
1028	AR2 35MM Camera 8 X	109.00
1030	QX Portable CD Player	169.00
1032	R5 Micro Digital Tape Recorder	89.00

次のコマンドを発行して言語をスペイン語に設定し、上記のリクエストを実行します。

```
SET LANG = SPA
```


出力結果には、PROD_NUM および PRODNAME フィールドに存在する TITLE_ES 属性がフィールドタイトルとして表示されます。PRICE フィールドにはスペイン語のタイトルが存在しないため、TITLE 属性のフィールドタイトルが表示されます。

Numero	Nombre	Price:
de Producto:	de Producto:	-----
-----	-----	-----
1004	2 Hd VCR LCD Menu	179.00
1008	DVD Upgrade Unit for Cent. VCR	199.00
1026	AR3 35MM Camera 10 X	129.00
1028	AR2 35MM Camera 8 X	109.00
1030	QX Portable CD Player	169.00
1032	R5 Micro Digital Tape Recorder	89.00

一時項目 (DEFINE) の記述 - DEFINE

トピックス

一時項目 (DEFINE) の使用

手順/構文

一時項目 (DEFINE) の記述

参照

マスターファイルの一時項目 (DEFINE) 使用時の注意

オプションの DEFINE 属性を使用して、レポートの作成時に使用する一時項目 (DEFINE) を作成します。一時項目 (DEFINE) の値は、データソースに存在する情報、つまり永続フィールドから取得することができます。一時項目 (DEFINE) の一般的な使用例は次のとおりです。

- データレコードに存在しない新しい数値を計算する。
- 他の文字列から新しい文字列を計算する。
- データ値を範囲またはグループに分類する。
- 計算内でサブルーチンを呼び出す。

一時項目 (DEFINE) は、データソースを使用してレポートを作成する際に常に使用することができます。

構文 一時項目 (DEFINE) の記述

```
DEFINE fieldname/format [REDEFINES field2] = expression;
[,TITLE='title',]
[TITLE_ln='titleln', ... ,]
[,DESC[CRPTION]='desc',]
[DESC_ln='descln', ... ,]$,
```

説明

fieldname

一時項目 (DEFINE) の名前です。この名前は、FIELDNAME 属性を使用して割り当てる名前の規則に従います。FIELDNAME 属性についての詳細は、371 ページの「[フィールド名 - FIELDNAME](#)」を参照してください。

format

ファイルフォーマットです。このフォーマットは、USAGE 属性を使用して割り当てるフォーマットと同様の方法で指定します。詳細は、382 ページの「[表示データタイプ - USAGE](#)」を参照してください。特定のフォーマットを指定しない場合は、デフォルトの D12.2 が割り当てられます。

field2

フィールド名が複数のセグメントに存在する場合に、そのフィールドの再定義または再計算を行えるようにします。

expression

有効な式です。式はセミコロン (;) で終了する必要があります。式についての詳細は、なお、一時項目 (DEFINE) の式に IF-THEN 句を使用する場合は、ELSE 句を含める必要があります。

`TITLE='title'`

一時項目 (DEFINE) のデフォルト言語のフィールドタイトルです。

`TITLE_ln='titleln'`

言語コード「ln」で指定する言語の、一時項目 (DEFINE) のフィールドタイトルです。

`DESC[CRPTION]='desc'`

一時項目 (DEFINE) のデフォルト言語の説明です。

`DESC_ln='descln'`

言語コード「ln」で指定する言語の、一時項目 (DEFINE) の説明です。

そのセグメントのすべてのフィールド設営の後に各 DEFINE 属性を配置します。

例 一時項目 (DEFINE) の記述

次の例は、CAR セグメントに「PROFIT」という一時項目 (DEFINE) を記述する方法を示しています。

```
SEGMENT = CARS ,SEGTYPE = S1 ,PARENT = CARREC, $
  FIELDNAME = DEALER_COST ,ALIAS = DCOST ,USAGE = D7, $
  FIELDNAME = RETAIL_COST ,ALIAS = RCOST ,USAGE = D7, $
DEFINE PROFIT/D7 = RETAIL_COST - DEALER_COST; $
```

参照 マスターファイルの一時項目 (DEFINE) 使用時の注意

DEFINE 属性の使用時には次の規則が適用されます。

- **エイリアス** DEFINE 属性にはエイリアスはありません。
- **変更** 一時項目 (DEFINE) の宣言は、いつでも変更することができます。
- **DEFINE FILE コマンド**は、マスターファイルに同名で記述した一時項目 (DEFINE) より優先されます。
- 一時項目 (DEFINE) の値を取得する式が関数を呼び出す場合、USERFCHK パラメータを FULL に設定していないとパラメータの個数とタイプの確認は実行されません。

一時項目 (DEFINE) の使用

DEFINE 属性では、修飾フィールド名を式の左辺に使用することはできません。左辺に WITH 句を使用して、選択した任意の実フィールドと同一セグメントに一時項目 (DEFINE) を配置します。これにより、DEFINE 式を評価する位置が決定されます。

DEFINE の右辺の式は、同一パスにある任意セグメントのフィールドを参照することができます。マスターファイルの DEFINE ステートメントの右辺の式には、修飾フィールド名を使用することができます。

マスターファイルの DEFINE 属性は、そのパスにあるフィールドのみを参照することができます。複数のパスにあるフィールドから値を取得する一時項目 (DEFINE) を作成するには、レポートリクエストを発行する前に、代替ビューを使用して DEFINE FILE を使用して一時項目 (DEFINE) を作成する必要があります。詳細は、DEFINE FILE コマンドは、1 回だけ使用する一時項目 (DEFINE) を作成し、マスターファイルにその一時項目 (DEFINE) の宣言を追加したくない場合に使用すると役立ちます。

マスターファイルに記述した一時項目 (DEFINE) は、実際に格納されたフィールドのように扱い、データソースを使用する際に常に使用することができます。そのため、マスターファイルに記述した一時項目 (DEFINE) はレポートリクエストでクリアすることはできません。

一時項目 (DEFINE) は、JOIN 内のクロスリファレンスとして使用することはできません。ただし、これらを JOIN のホストフィールドとして使用することができます。

注意： MAINTAIN では、定数値の DEFINE 属性はサポートされません。MAINTAIN プロシジャでそのようなフィールドを使用すると、次のメッセージが表示されます。

```
(FOC03605) name is not recognized.
```

一時項目 (COMPUTE) の記述 - COMPUTE

手順/構文

マスターファイルへの COMPUTE コマンドの追加

参照

マスターファイル COMPUTE 使用時の注意

COMPUTE コマンドをマスターファイルに含めて、後に続く TABLE リクエストで参照することができます。これにより、式を 1 回だけ作成して、複数のリクエストで再利用することができます。

構文

マスターファイルへの COMPUTE コマンドの追加

```
COMPUTE fieldname/fmt=expression;  
[ ,TITLE='title', ]  
[TITLE_ln='titleln', ... , ]  
[ ,DESC[CRPTION]='desc', ]  
[DESC_ln='descln', ... , ]$
```

説明

fieldname

一時項目 (COMPUTE) の名前です。

fmt

一時項目 (COMPUTE) のフォーマットおよび長さです。

expression

一時項目 (COMPUTE) の計算に使用する式です。

`TITLE='title'`

一時項目 (COMPUTE) のデフォルト言語のフィールドタイトルです。

`TITLE_ln='titleln'`

言語コード「ln」で指定する言語の、一時項目 (COMPUTE) のフィールドタイトルです。

`DESC[CRPTION]='desc'`

一時項目 (COMPUTE) のデフォルト言語の説明です。

`DESC_ln='descln'`

言語コード「ln」で指定する言語の、一時項目 (COMPUTE) の説明です。

参照

マスターファイル COMPUTE 使用時の注意

すべてのインスタンスにおいて、マスターファイルで指定する COMPUTE には一時項目 (COMPUTE) と同一の機能および制限が適用されます。特に、マスターファイルの COMPUTE フィールドは次の規則に従う必要があります。

- JOIN、DEFINE、ACROSS 句で使用することはできません。また、演算接頭語とともに使用することはできません。
- 選択条件として使用する場合、構文は IF TOTAL フィールドまたは WHERE TOTAL フィールドです。
- ソートフィールドとして使用する場合、構文は「BY TOTAL COMPUTE field」です。
- 一時項目 (COMPUTE) を見出し、脚注に挿入するには、HEADING または FOOTING コマンドより前に参照する必要があります。

注意： MAINTAIN では、マスターファイルでの一時項目 (COMPUTE) の使用は、現在サポートされていません。また、これらの一時項目 (COMPUTE) は、アップデートアシストウィザードには表示されません。

例

一時項目 (COMPUTE) のマスターファイルでのコーディングおよびアクセス

一時項目 (COMPUTE) をマスターファイルにコーディングするには、標準の COMPUTE 構文を使用します。これにより、以降の TABLE リクエストで、その一時項目 (COMPUTE) を参照してアクセスすることができます。動詞オブジェクトとして使用する場合 (次の例を参照)、構文は「SUM (または PRINT) COMPUTE field」です。

次の例は、SALESTES マスターファイル (埋め込まれた COMPUTE で変更された SALES FILE) です。

```

FILENAME=SALESTES, SUFFIX=FOC,
SEGNAME=STOR_SEG, SEGTYPE=S1,
    FIELDNAME=STORE_CODE, ALIAS=SNO, FORMAT=A3, $
    FIELDNAME=CITY, ALIAS=CTY, FORMAT=A15, $
    FIELDNAME=AREA, ALIAS=LOC, FORMAT=A1, $

SEGNAME=DATE_SEG, PARENT=STOR_SEG, SEGTYPE=SH1,
    FIELDNAME=DATE, ALIAS=DTE, FORMAT=A4MD, $

SEGNAME=PRODUCT, PARENT=DATE_SEG, SEGTYPE=S1,
    FIELDNAME=PROD_CODE, ALIAS=PCODE, FORMAT=A3, FIELDTYPE=I, $
    FIELDNAME=UNIT_SOLD, ALIAS=SOLD, FORMAT=I5, $
    FIELDNAME=RETAIL_PRICE, ALIAS=RP, FORMAT=D5.2M, $
    FIELDNAME=DELIVER_AMT, ALIAS=SHIP, FORMAT=I5, $
    FIELDNAME=OPENING_AMT, ALIAS=INV, FORMAT=I5, $
    FIELDNAME=RETURNS, ALIAS=RTN, FORMAT=I3, MISSING=ON, $
    FIELDNAME=DAMAGED, ALIAS=BAD, FORMAT=I3, MISSING=ON, $

COMPUTE REVENUE/D12.2M=UNIT_SOLD*RETAIL_PRICE;
    
```

次の TABLE リクエストは REVENUE フィールドを使用します。

```

TABLE FILE SALESTES
HEADING CENTER
"NEW YORK PROFIT REPORT"
" "
SUM UNIT_SOLD AS 'UNITS,SOLD' RETAIL_PRICE AS 'RETAIL_PRICE'
COMPUTE REVENUE;
BY PROD_CODE AS 'PROD,CODE'
WHERE CITY EQ 'NEW YORK'
END
    
```

出力結果は次のとおりです。

```

NEW YORK PROFIT REPORT

PROD  UNITS
CODE  SOLD   RETAIL_PRICE      REVENUE
----  -
B10   30     $ .85             $25.50
B17   20     $1.89            $37.80
B20   15     $1.99            $29.85
C17   12     $2.09            $25.08
D12   20     $2.09            $41.80
E1    30     $ .89            $26.70
E3    35     $1.09            $38.15
    
```

フィルタの記述 - FILTER

手順/構文

マスターファイルのフィルタ宣言

リクエストでのマスターファイルフィルタの使用

参照

マスターファイルフィルタ使用時の注意

ブール一時項目 (DEFINE) (True または False を評価する DEFINE フィールド) を、レコード選択条件として使用することができます。一時項目 (DEFINE) を主としてレコード選択に使用する場合、DEFINE ではなく FILTER を使用して式を保存することにより、マスターファイルでこの目的を明確にし、一時項目 (DEFINE) を分類することができます。フィルタは次の機能を提供します。

- 頻繁に使用する選択条件をマスターファイルに分類、保存しておき、複数のリクエストおよびツールで再利用することができます。
- VSAM、ISAM などのデータソースでは、フィルタ式を WHERE または IF 句に直接挿入できるものがあり、ブール一時項目 (DEFINE) に比べて最適化を向上させることができます。

構文

マスターファイルのフィルタ宣言

```
FILTER filtername = expression;  
[, DESC[CRIPITION]='desc',]  
[DESC_ln='descln', ... ,]$,
```

説明

filtername

フィルタに割り当てる名前です。フィルタには内部的に 11 フォーマットが割り当てられ、変更することはできません。

expression

True (1 をフィルタフィールドに割り当て) または False (0 をフィルタフィールドに割り当て) を評価する論理式です。その他の式のフィールドは、すべてマスターファイルの標準数値一時項目 (DEFINE) になります。ダイアログマネージャ変数は、標準のマスターファイル DEFINE と同様、フィルタ式で使用することができます。

```
DESC[CRIPITION]='desc'
```

ソートオブジェクトのデフォルト言語の説明です。

```
DESC_ln='descln'
```

言語コード「ln」で指定する言語の、ソートオブジェクトの説明です。

構文 リクエストでのマスターファイルフィルタの使用

```
TABLE FILE filename .  
.  
.  
{WHERE|IF} expression_using_filters
```

説明

expression_using_filters

フィルタを参照する論理式です。WHERE 句では、論理式で 1 つまたは複数のフィルタおよび一時項目 (DEFINE) を参照することができます。

参照 マスターファイルフィルタ使用時の注意

- ❑ フィルタフィールド名には内部的に 11 フォーマットが割り当てられ、変更することはできません。
- ❑ フィルタは、標準の数値一時項目 (DEFINE) としてレポートリクエストの任意の位置に使用することができますが、WHERE TOTAL テストではサポートされません。

例 マスターファイルフィルタの定義と使用

ここでは、MOVIES マスターファイルに次のフィルタ宣言を追加する場合について考察します。

```
FILTER G_RATING = RATING EQ 'G' OR 'PG'; $
```


次のリクエストは G_RATING を適用します。

```
TABLE FILE MOVIES
HEADING CENTER
"Rating G and PG"
PRINT TITLE CATEGORY RATING
WHERE G_RATING
ON TABLE SET PAGE NOPAGE
ON TABLE SET GRID OFFON TABLE SET STYLE *
type=report, style=bold, color=black, bgcolor=yellow, $
type=data, bgcolor=aqua, $
ENDSTYLE
END
```

出力結果は次のとおりです。

Rating G and PG		
TITLE	CATEGORY	RATING
JAWS	ACTION	PG
CABARET	MUSICALS	PG
BABETTE'S FEAST	FOREIGN	G
SHAGGY DOG, THE	CHILDREN	G
REAR WINDOW	MYSTERY	PG
VERTIGO	MYSTERY	PG
BACK TO THE FUTURE	COMEDY	PG
GONE WITH THE WIND	CLASSIC	G
AIRPLANE	COMEDY	PG
ALICE IN WONDERLAND	CHILDREN	G
ANNIE HALL	COMEDY	PG
FIDDLER ON THE ROOF	MUSICALS	G
BIG	COMEDY	PG
TOP GUN	ACTION	PG
FAMILY, THE	FOREIGN	PG
BAMBI	CHILDREN	G
DEATH IN VENICE	FOREIGN	PG

ソートオブジェクトの記述 - SORTOBJ

手順/構文

マスターファイルのフィルタ宣言

リクエストでのソートオブジェクトの参照

参照

マスターファイルの一時項目 (DEFINE) 使用時の注意

マスターファイルでソート句と属性を定義しておく、そのマスターファイルを使用するリクエストで、それらを名前参照することができます。ソートオブジェクトのテキスト全体は、ソートオブジェクトが TABLE で参照される時点で、置き換えられます。ソートオブジェクトのソート句は、この置き換えの前には確認されません。確認されるのは、マスターファイルの SORTOBJ レコードのソートオブジェクト名と等号 (=) のみです。

参照

マスターファイルの一時項目 (DEFINE) 使用時の注意

- ソートオブジェクトは、最初の SEGNAME/SEGMENT の後に、任意の位置に記述することができます。ただし、マスターファイルの一時項目 (DEFINE) を含むすべてのフィールドの後に記述する必要があります。
- ソートオブジェクトには、マスターファイルフィールドとローカルの一時項目 (DEFINE) の両方を使用することができます。
- マスターファイルにソートオブジェクトの記述の数に制限はありませんが、TABLE リクエストが参照する数の結果が、リクエストのソート句の最大値を超えることはできません。
- ソートオブジェクトの記述の後に、オプションの属性を含めることがあります。
- ソートオブジェクトと同名のフィールドが存在し、その名前がリクエストで参照された場合は、ソートオブジェクトが使用されます。

構文 マスターファイルのフィルタ宣言

```
FILE= ...
SEG= ...
FIELD= ...
SORTOBJ sortname = {BY|ACROSS} sortfield1 [attributes]
  [{BY|ACROSS} sortfield2 ... ];
  [,DESC[CRPTION]='desc',]
  [DESC_ln='descln', ... ,]$\
```

説明

sortname

テンプレートの名前です。

sortfield1, sortfield2 ..

レポートリクエストのソートに使用するマスターファイルのフィールドまたはローカルの一時項目 (DEFINE) です。

attributes

有効な任意のソート属性です。

;

ソートオブジェクトの式の末尾を区切るために必要な構文です。

DESC[CRPTION]='desc'

ソートオブジェクトのデフォルト言語の説明です。

DESC_ln='descln'

言語コード「ln」で指定する言語の、ソートオブジェクトの説明です。

構文 リクエストでのソートオブジェクトの参照

```
TABLE FILE ...
.
.
.
BY sortname .
.
.
END
```

説明

sortname

リクエストに挿入するソートオブジェクトです。

例 ソートオブジェクトの宣言と参照

次のソートオブジェクトは、GGSALES マスターファイルに「CRSORT」という名前で記述されています。次の 2 つのソート句が定義されています。

- SKIP-LINE 属性付きの BY REGION
- ACROSS CATEGORY

```
SORTOBJ CRSORT = ACROSS CATEGORY BY REGION SKIP-LINE ; , $
```

次のリクエストは、CRSORT ソートオブジェクトを参照します。

```
TABLE FILE GGSALES
SUM DOLLARS
BY CRSORT
ON TABLE SET PAGE NOPAGE
END
```

出力結果は次のとおりです。

Region	Category		
	Coffee	Food	Gifts
Midwest	4178513	4404483	2931349
Northeast	4201057	4445197	2848289
Southeast	4435134	4308731	3037420
West	4493483	4204333	2977092

マスターファイル DEFINE による日付システム変数の使用

参照

マスターファイルの DEFINE での日付システム変数用のメッセージ

マスターファイルの DEFINE フィールドでは、ダイアログマネージャのシステム日付変数を使用して、リクエストでマスターファイルを解析するたびにシステムの日付を取得することができます。

日付変数の戻り値のフォーマットは、変数名で指定されたフォーマットと同一です。たとえば、&DATEYYMD の日付戻り値のフォーマットは YYMD です。例外として、変数名の &DATE および &TOD の場合は文字の値を返すため、文字フォーマットのフィールドに割り当てる必要があります。また、DEFINE 式では変数名の &DATE および &TOD を一重引用符 (') で囲む必要があります。

以下は、マスターファイルの DEFINE でサポートされる変数です。

- ❑ &DATE
- ❑ &TOD
- ❑ &DATEMDY
- ❑ &DATEDMY
- ❑ &DATEYMD
- ❑ &DATEMDYY
- ❑ &DATEDMY Y
- ❑ &DATEYYMD
- ❑ &DMY
- ❑ &YMD
- ❑ &MDY
- ❑ &YYMD
- ❑ &MDYY
- ❑ &DMYY

上記以外の変数はマスターファイルではサポートされません。

例

マスターファイルの DEFINE での日付変数 &DATE の使用

次の EMPLOYEE マスターファイルには、「TDATE」という一時項目 (DEFINE) が追加されています。TDATE のフォーマットは A12 で、&DATE の値を取得します。戻り値は文字であり、この変数は一重引用符 (') で囲む必要があります。

```
FILENAME=EMPLOYEE, SUFFIX=FOC
SEGNAME=EMPINFO,  SEGTYPE=S1
FIELDNAME=EMP_ID,    ALIAS=EID,      FORMAT=A9,      $
FIELDNAME=LAST_NAME, ALIAS=LN,      FORMAT=A15,     $
FIELDNAME=FIRST_NAME, ALIAS=FN,      FORMAT=A10,     $
FIELDNAME=HIRE_DATE,  ALIAS=HDT,     FORMAT=I6YMD,   $
FIELDNAME=DEPARTMENT, ALIAS=DPT,     FORMAT=A10,     $
FIELDNAME=CURR_SAL,   ALIAS=CSAL,    FORMAT=D12.2M,  $
FIELDNAME=CURR_JOBCODE, ALIAS=CJC,    FORMAT=A3,      $
FIELDNAME=ED_HRS,     ALIAS=OJT,     FORMAT=F6.2,    $
DEFINE TDATE/A12     ='&DATE';, $
.
.
.
```

次のリクエストは TDATE の値を表示します。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME FIRST_NAME HIRE_DATE TDATE AS 'TODAY''S,DATE'
WHERE LAST_NAME EQ 'BANNING'
END
```

出力結果は次のとおりです。

				TODAY'S
<u>LAST NAME</u>	<u>FIRST NAME</u>	<u>HIRE DATE</u>	<u>DATE</u>	
BANNING	JOHN	82/08/01	06/17/04	

例 マスターファイルの DEFINE での日付変数 &YYMD の使用

次の EMPLOYEE マスターファイルには、「TDATE」という一時項目 (DEFINE) が追加されています。TDATE のフォーマットは YYMD で、&YYMD の値を取得します。

```
FILENAME=EMPLOYEE, SUFFIX=FOC
SEGNAME=EMPINFO, SEGTYPE=S1
  FIELDNAME=EMP_ID, ALIAS=EID, FORMAT=A9, $
  FIELDNAME=LAST_NAME, ALIAS=LN, FORMAT=A15, $
  FIELDNAME=FIRST_NAME, ALIAS=FN, FORMAT=A10, $
  FIELDNAME=HIRE_DATE, ALIAS=HDT, FORMAT=I6YMD, $
  FIELDNAME=DEPARTMENT, ALIAS=DPT, FORMAT=A10, $
  FIELDNAME=CURR_SAL, ALIAS=CSAL, FORMAT=D12.2M, $
  FIELDNAME=CURR_JOBCODE, ALIAS=CJC, FORMAT=A3, $
  FIELDNAME=ED_HRS, ALIAS=OJT, FORMAT=F6.2, $
DEFINE TDATE/YYMD = &YYMD ;, $
.
.
.
```

次のリクエストは TDATE の値を表示します。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME FIRST_NAME HIRE_DATE TDATE AS 'TODAY' 'S,DATE'
WHERE LAST_NAME EQ 'BANNING'
END
```

出力結果は次のとおりです。

```

                                     TODAY'S
LAST_NAME FIRST NAME HIRE DATE DATE
BANNING   JOHN           82/08/01 2004/06/17
```

参照 マスターファイルの DEFINE での日付システム変数用のメッセージ

マスターファイルの一時項目 (DEFINE) でサポートされていない変数を使用すると、次のメッセージが表示されます。

```
(FOC104) DEFINE IN MASTER REFERS TO A FIELD OUTSIDE ITS SCOPE: var
```


変数を使用したマスターファイルおよびアクセスファイルのパラメータ化

手順/構文

マスターファイル変数の作成

参照

マスターファイルおよびアクセスファイルの属性での変数サポート

マスターファイルでグローバル変数を定義し、その変数を使用してマスターファイルおよびそれに対応するアクセスファイル内の特定の属性をパラメータ化することができます。たとえば、アクセスファイル内の接続属性をマスターファイルで定義した変数でパラメータ化し、実行時に実際の接続名を指定することができます。

構文

マスターファイル変数の作成

マスターファイルの FILE 宣言の前に、次の変数定義を追加します。

```
VARIABLE NAME=[&&]var, USAGE=Aln, [DEFAULT=defvalue,][QUOTED={OFF|ON},] $
```

説明

`[&&]var`

グローバル変数に割り当てる名前です。マスターファイルまたはアクセスファイル内の変数を参照するときは、2つのアンパサンド (&) を名前の先頭に追加する必要があります。ただし、変数を定義する場合は、これらのアンパサンドはオプションとして追加します。

`ln`

変数値の最大の長さです。

`defvalue`

指定した変数のデフォルト値です。実行時に値が設定されていない場合は、この値が使用されます。

`QUOTED = {OFF|ON}`

ON を指定すると、変数に割り当てられた文字列が一重引用符 (') で囲まれます。文字列内の一重引用符は、2つの一重引用符に変換されます。デフォルト値は OFF です。

参照 マスターファイルおよびアクセスファイルの属性での変数サポート

マスターファイルでは、次の属性を変数でパラメータ化することができます。POSITION、OCCURS、REMARKS、DESCRIPTION、TITLE、HELPMESSAGE。

マスターファイルの DBA セクションで、USER 属性と VALUE 属性をパラメータ化することができます。

アクセスファイルでは、次の属性を変数でパラメータ化することができます。CONNECTION、TABLENAME、START、CHKPT_SAVE、CHKPT_FILE、POLLING、TIMEOUT、MAXLUWS、ACTION、MSGLIMIT、DIRECTORY、NAME、EXTENSION、DATA_ORIGIN、MAXFILES、MAXRECS、PICKUP、TRIGGER、DISCARD、ARCHIVE。

注意：複数の変数を連結して、1つの属性値を作成することができます。

例 マスターファイルおよびアクセスファイルでの属性のパラメータ化

次のリクエストは、EMPLOYEE という名前の FOCUS データソースから、ORAEMP という名前の Oracle テーブルを作成します。

```
TABLE FILE EMPLOYEE
SUM LAST_NAME FIRST_NAME CURR_SAL CURR_JOBCODE DEPARTMENT
BY EMP_ID
ON TABLE HOLD AS ORAEMP FORMAT SQLORA
END
```

このリクエストで作成されるマスターファイルは次のとおりです。

```
FILENAME=ORAEMP , SUFFIX=SQLORA , $
SEGMENT=SEG01, SEGTYPE=S0, $
  FIELDNAME=EMP_ID, ALIAS=EID, USAGE=A9, ACTUAL=A9, $
  FIELDNAME=LAST_NAME, ALIAS=LN, USAGE=A15, ACTUAL=A15, $
  FIELDNAME=FIRST_NAME, ALIAS=FN, USAGE=A10, ACTUAL=A10, $
  FIELDNAME=CURR_SAL, ALIAS=CSAL, USAGE=D12.2M, ACTUAL=D8, $
  FIELDNAME=CURR_JOBCODE, ALIAS=CJC, USAGE=A3, ACTUAL=A3, $
  FIELDNAME=DEPARTMENT, ALIAS=DPT, USAGE=A10, ACTUAL=A10, $
```

このリクエストで作成されるアクセスファイルは次のとおりです。

```
SEGNAME=SEG01, TABLENAME=ORAEMP, KEYS=01, WRITE=YES, $
```

次の変数定義をマスターファイルに追加して、アクセスファイル内の TABLENAME 属性およびマスターファイル内の EMP_ID フィールドの TITLE 属性をパラメータ化します。

```
FILENAME=ORAEMP, SUFFIX=SQLORA , $
VARIABLE NAME=table, USAGE=A8, DEFAULT=EDUCFILE, $
VARIABLE NAME=emptitle, USAGE=A30, DEFAULT=empid,$
```

マスターファイルで、EMP_ID の FIELD 宣言に TITLE 属性を追加します。

```
FIELDNAME=EMP_ID, ALIAS=EID, USAGE=A9, ACTUAL=A9,
      TITLE='&&emptitle', $
```

アクセスファイルで、TABLENAME 属性の値を変数名に置換します。

```
SEGNAME=SEG01, TABLENAME=&&table, KEYS=01, WRITE=YES, $
```

次のリクエストは、変数の値を設定し、TABLE リクエストを発行します。

```
-SET &&table = ORAEMP;
-SET &&emptitle = 'Id,number';
TABLE FILE ORAEMP
PRINT EMP_ID LAST_NAME FIRST_NAME DEPARTMENT
END
```

ここで、-SET コマンド内の &&emptitle の値が一重引用符 (') で囲まれています。これは、この値に特殊文字のカンマ (,) が含まれているためです。一重引用符は文字列の一部ではなく、またレポート出力には表示されません。変数定義に QUOTED=ON 属性を含めた場合は、フィールドタイトルが一重引用符に囲まれた状態で表示されます。

レポート出力では、EMP_ID フィールドのフィールドタイトルに、&&emptitle で設定された値が表示されます。また、このリクエストがアクセスするテーブルは、この例の最初の手順で作成した ORAEMP テーブルになります。

```
Id
number      LAST_NAME      FIRST_NAME  DEPARTMENT
-----
071382660   STEVENS          ALFRED      PRODUCTION
112847612   SMITH            MARY        MIS
117593129   JONES            DIANE       MIS
119265415   SMITH            RICHARD     PRODUCTION
119329144   BANNING          JOHN        PRODUCTION
123764317   IRVING           JOAN        PRODUCTION
126724188   ROMANS           ANTHONY     PRODUCTION
219984371   MCCOY            JOHN        MIS
326179357   BLACKWOOD        ROSEMARIE   MIS
451123478   MCKNIGHT         ROGER       PRODUCTION
543729165   GREENSPAN        MARY        MIS
818692173   CROSS            BARBARA     MIS
```

例

変数の連結による属性値の作成

次の例の TABLENAME 属性には、データベース名、オーナー ID、テーブル接頭語、および変数接尾語を含む固定テーブル名で構成される複数部分名を指定する必要があります。ここでは、各部分の変数を別々に定義し、それらの変数を連結します。

最初に、各部分の変数を別々に定義します。

```
VARIABLE NAME=db,USAGE=A8,DEFAULT=mydb,$
VARIABLE NAME=usr,USAGE=A8,DEFAULT=myusrid,$
VARIABLE NAME=tprf,USAGE=A4,DEFAULT=test_,$
VARIABLE NAME=tsuf,USAGE=YYM,$
```

アクセスファイルで、これらの変数を連結して TABLENAME 属性を作成します。各部分の区切り文字はピリオド (.) ですが、変数名を連結してピリオドを保持するためには、2 つのピリオドを使用する必要があります。

```
TABLENAME=&db..&usr..&tprf.table&tsuf,
```

デフォルト値に基づいて、TABLENAME は次のようになります。

```
TABLENAME=mydb.myusrid.test_table
```

リクエストでは、それぞれの変数に次の値を設定します。

```
I-SET &&db=db1;
-SET &&tprf=prod_;
-SET &&tsuf=200801;
```

これらの値により、使用される TABLENAME は次のようになります。

```
TABLENAME=db1.myusrid.prod_table200801
```

文字日付の日付への変換

トピックス

日付パターン変数の指定

日付パターン定数の指定

日付パターンサンプル

参照

DATEPATTERN 使用時の注意

データソースには、日付の値が文字フォーマットで格納され、特別な規格が存在せず、年、四半期、月などの構成要素の任意の組み合わせ、および任意の区切り文字を使用するものがあります。レポートにソートが設定されている場合、このようなデータはアルファベット順にソートされ、結果は実務上の意味を持ちません。データフィールドのソート、集計、レポート実行を適切に行うため、では、「DATEPATTERN」というマスターファイル属性で指定する変換パターンを使用して、文字フォーマットの日付を標準日付フォーマットに変換することができます。

パターンの各要素は、実際に入力される特定の文字、または日付構成要素を表す変数です。マスターファイルの USAGE 属性を編集して、日付構成要素の日付パターンを記述する必要があります。DATEPATTERN 文字列の最大長は 64 バイトです。

参照

DATEPATTERN 使用時の注意

- 元の日付に USAGE フォーマットに相当する要素が含まれていない場合、変換後の日付は、元のデータと同一になります。この場合、元のデータを表示するには、OCCURS セグメントを使用して元の文字フォーマットのフィールドを再定義し、リクエストでそのフィールドを表示することもできます。
- DATEPATTERN では、ACTUAL を USAGE フォーマットに変換する必要があります。そのため、SUFFIX=FOC および SUFFIX=XFOC のデータソースでは、DATEPATTERN はサポートされません。

日付パターン変数の指定

手順/構文

- 日付パターンに年を指定
- 日付パターンに月を表す数値を指定
- 日付パターンに月名を指定
- 日付パターンに曜日を指定
- 日付パターンにユリウス暦の日付を指定
- 日付パターンに曜日を指定
- 日付パターンに四半期を指定

有効な日付構成要素 (変数) は、年、四半期、月、日、曜日です。日付パターン内の変数は、大括弧 ([]) で囲みます。これらの大括弧は、入力または出力の一部ではありません。データに大括弧が含まれる場合は、日付パターン内でエスケープ文字を使用して、データの大括弧を変数を囲む大括弧と区別する必要があります。

構文 **日付パターンに年を指定**

[YYYY]

4 桁の年を指定します。

[YYYY]

4 桁の年を指定します。

[YY]

2 桁の年を指定します。

[yy]

0 (ゼロ) を非表示にする 2 桁の年を指定します (例、2008 年の場合は 8)。

[by]

空白でパディングされた 2 桁の年を指定します。

構文 **日付パターンに月を表す数値を指定**

[MM]

2桁の月番号を指定します。

[mm]

0 (ゼロ) を非表示にする月を表す数値を指定します。

[bm]

ブランクでパディングされた月を表す数値を指定します。

構文 **日付パターンに月名を指定**

[MON]

3文字の大文字の月名を指定します。

[mon]

3文字の子文字の月名を指定します。

[Mon]

3文字の先頭大文字の月名を指定します。

[MONTH]

大文字の月の完全名を指定します。

[month]

小文字の月の完全名を指定します。

[Month]

先頭大文字の月の完全名を指定します。

構文 **日付パターンに曜日を指定**

[DD]

2桁の月単位の日付を指定します。

[dd]

0 (ゼロ) を非表示にする月単位の日付を指定します。

[bd]

空白でパディングされた月単位の日付を指定します。

構文 **日付パターンにユリウス暦の日付を指定**

[DDD]

3 桁の年単位の日付を指定します。

[ddd]

0 (ゼロ) を非表示にする年単位の日付を指定します。

[bdd]

空白でパディングされた年単位の日付を指定します。

構文 **日付パターンに曜日を指定**

[WD]

1 桁の曜日を指定します。

[DAY]

3 文字の大文字の曜日名を指定します。

[day]

3 文字の小文字の曜日名を指定します。

[Day]

3 文字の先頭大文字の曜日名を指定します。

[WDAY]

大文字の曜日の完全名を指定します。

[wday]

小文字の曜日の完全名を指定します。

[Wday]

先頭大文字の曜日の完全名を指定します。

曜日の場合、WEEKFIRST の設定で、週の開始日を定義します。

構文 日付パターンに四半期を指定

[Q]

1 桁の四半期番号を指定します (1、2、3、4 のいずれか)。

Q2 や Q02 などの文字列の場合は、[Q] の前に定数を使用します (例、Q0[Q])。

日付パターン定数の指定

変数の間に、任意の定数値を挿入することができます。

通常は変数の一部として解釈される文字を挿入する場合は、円記号 (¥) を使用します。以下はその例です。

□ 定数値として左大括弧 (()) を指定するには、¥[を使用します。

□ 定数値として円記号 (¥) を指定するには、¥¥ を使用します。

一重引用符 (') の場合は、連続する 2 つの一重引用符 (") を使用します。

日付パターンサンプル

データソースの日付が「CY 2001 Q1」の形式の場合、DATEPATTERN 属性は次のとおりです。

```
DATEPATTERN = 'CY [YYYY] Q[Q]'
```

データソースの日付が「Jan 31, 01」の形式の場合、DATEPATTERN 属性は次のとおりです。

```
DATEPATTERN = '[Mon] [DD], [YY]'
```

データソースの日付が「APR-06」の形式の場合、DATEPATTERN 属性は次のとおりです。

```
DATEPATTERN = '[MON]-[YY]'
```

データソースの日付が「APR -06」の形式の場合、DATEPATTERN 属性は次のとおりです。

```
DATEPATTERN = '[MON] - [YY]'
```

データソースの日付が「APR '06」の形式の場合、DATEPATTERN 属性は次のとおりです。

```
DATEPATTERN = '[MON] '['[YY]'
```

データソースの日付が「APR[06]」の形式の場合、DATEPATTERN 属性は次のとおりです。

```
DATEPATTERN = '[MON] ¥  
[[YY]¥  
' (or '[MON] ¥  
[[YY]]'
```

右大括弧に、エスケープ文字を追加する必要はありません。

例 文字の日付によるソート

次の例で、は、以下のデータが格納されたシーケンシャルファイルです。

```
June 1, '02  
June 2, '02  
June 3, '02  
June 10, '02  
June 11, '02  
June 12, '02  
June 20, '02  
June 21, '02  
June 22, '02  
June 1, '03  
June 2, '03  
June 3, '03  
June 10, '03  
June 11, '03  
June 12, '03  
June 20, '03  
June 21, '03  
June 22, '03  
June 1, '04  
June 2, '04  
June 3, '04  
June 4, '04  
June 10, '04  
June 11, '04  
June 12, '04  
June 20, '04  
June 21, '04  
June 22, '04
```

DATE1 マスターファイルの DATE1 フィールドには、USAGE フォーマットおよび ACTUAL フォーマットが設定されます。これらは両者とも文字 (A18) フォーマットです。

```
FILENAME=DATE1 , SUFFIX=FIX ,  
DATASET = , $  
SEGMENT=FILE1, SEGTYPE=S0, $  
FIELDNAME=DATE1, ALIAS=E01, USAGE=A18, ACTUAL=A18, $
```

次のリクエストは、DATE1 FIELD でソートします。

```
TABLE FILE DATE1
PRINT DATE1 NOPRINT
BY DATE1
ON TABLE SET PAGE NOPAGE
END
```

出力で、文字の日付は、日付順ではなくアルファベット順にソートされています。

```
DATE1
-----
June 1, '02
June 1, '03
June 1, '04
June 10, '02
June 10, '03
June 10, '04
June 11, '02
June 11, '03
June 11, '04
June 12, '02
June 12, '03
June 12, '04
June 2, '02
June 2, '03
June 2, '04
June 20, '02
June 20, '03
June 20, '04
June 21, '02
June 21, '03
June 21, '04
June 22, '02
June 22, '03
June 22, '04
June 3, '02
June 3, '03
June 3, '04
June 4, '04
```

日付を正しくソートするため、マスターファイルに DATEPATTERN 属性を追加して、によって日付が の日付フィールドに変換されるようにします。また、USAGE フォーマットを編集し、 の日付フォーマットにする必要もあります。適切なパターンを構成するため、格納されている日付のすべての要素を記述する必要があります。文字の日付には、次の変数と定数があります。

- 変数 - 先頭大文字の月の完全名 [Month]。

- 定数 - ブランク。
- 変数 - ゼロを非表示にした月を表す数値 [dd]。
- 定数 - カンマ (,)、ブランク、アポストロフィ (') (パターンでは 2 つのアポストロフィとして記述) の順に入力。
- 変数 - 2 桁の年 [YY]。

編集後のマスターファイルは、次のようになります。DEFCENT 属性を追加して、2 桁の年を 4 桁の年に変換します。

```
FILENAME=DATE1      , SUFFIX=FIX ,  
DATASET =           , $  
SEGMENT=FILE1, SEGTYPE=S0, $  
    FIELDNAME=DATE1, ALIAS=E01, USAGE=MtrDYY, ACTUAL=A18,  
    DEFCENT=20,  
    DATEPATTERN = '[Month] [dd], ''[YY]'' , $
```

同一のリクエストを発行すると、出力は次のようになります。DATE1 は、USAGE で指定した MtrDYY フォーマットの の日付に変換されています。

```
DATE1
-----
June  1, 2002
June  2, 2002
June  3, 2002
June 10, 2002
June 11, 2002
June 12, 2002
June 20, 2002
June 21, 2002
June 22, 2002
June  1, 2003
June  2, 2003
June  3, 2003
June 10, 2003
June 11, 2003
June 12, 2003
June 20, 2003
June 21, 2003
June 22, 2003
June  1, 2004
June  2, 2004
June  3, 2004
June  4, 2004
June 10, 2004
June 11, 2004
June 12, 2004
June 20, 2004
June 21, 2004
June 22, 2004
```


インデックス

数字

- 100 分位 39, 41
- 10 進数 244
- 10 進数整数変換 342
- 4 分位 39, 41

A

- ABS 関数 348
- ACCBLN パラメータ 439
- ACCEPTBLANK パラメータ 439
- ACCEPT 属性 438, 439
- ACCEPT リスト 48
- ACTUAL 属性 370, 428, 429, 433
- ALIAS 属性 370, 380, 381
- ALLOWCVTERR パラメータ 408
- ARGLEN 関数 242
- ATODBL 関数 338
- AYMD 関数 277
- AYM 関数 276
- A データタイプ 395

B

- BAR 関数 348

BITSON 関数 243

BYTVAL 関数 244

C

- CHECK FILE コマンド
 - 長いフィールド名 373
- CHGDAT 関数 278
- CHKFMT 関数 245
- CHKPCK 関数 350
- COMPUTE オブジェクトの参照 452
- COMPUTE の演算 190
- CTRAN 関数 246
- CTRFLD 関数 248

D

- DADMY 関数 281
- DADYM 関数 281
- DAMDY 関数 281
- DAMYD 関数 281
- DATEADD 関数 283
- DATECVT 関数 284
- DATEDIF 関数 285
- DATEDISPLAY パラメータ 407, 408
 - ALLOWCVTERR 408

DATEFNS パラメータ 320, 334

DATEMOV 関数 287

DATEPATTERN 属性 469

DATETRAN 関数 290

DAYDM 関数 281

DAYMD 関数 281

DA 関数 281

DBA ウィンドウ 204

DBA (データベース管理) 200

DECODE 関数 269, 270, 271

DEFCENT 属性 370

DEFCENT 値 54

DEFINE 属性 370, 449, 450, 451

DEFINE の演算 190

DEFINITION 属性 441, 442

DESCRIPTION 属性 370, 441, 442

DESC 属性 441, 442

DMOD 関数 351

DMY 関数 310

DOWKL 関数 311

DOWK 関数 311

DTDMY 関数 312

DTDYM 関数 312

DTMDY 関数 312

DTMYD 関数 312

DTSTANDARD パラメータ 411

DTYDM 関数 312

DTYMD 関数 312

DT 関数 312

D データタイプ 386

E

EDIT 関数 249, 340

EXP 関数 352

F

FGETENV 関数 360

FIELDNAME 属性 370, 371, 372, 373

FIELD 属性 371, 372

FIQTR 関数 315

FIYR 関数 313

FIYYQ 関数 318

FML 階層 436, 438

データの記述 436

マスターファイル 438

要件 436

FMOD 関数 351

FORMAT 属性 383, 384

FTOA 関数 341

F データタイプ 387

G

GETTOK 関数 250

GETUSER 関数 361

GREGDT 関数 320

H

HADD 関数 321

HCVRT 関数 323

HDATE 関数 324

HDIFF 関数 325

HDTTM 関数 326

HELPMESSAGE 属性 370

HEXBYT 関数 342

HGETC 関数 327

HHMMSS 関数 327, 362

HINPUT 関数 328

HMIDNT 関数 329

HNAME 関数 330

HOLD ファイル 429, 430
日付時間データタイプ 430

HPART 関数 331

HSETPT 関数 332

HTIME 関数 333

H データタイプ 410, 413

I

IMOD 関数 351

INT 関数 353

ISO 標準日付時間フォーマット 411

ITONUM 関数 343

ITOPACK 関数 344

ITOZ 関数 345

I データタイプ 385

J

JOIN コマンド
長いフィールド名 373

JULDAT 関数 334

L

LAST 関数 272

LCWORD3 関数 254

LCWORD2 関数 253, 255

LCWORD 関数 252

LJUST 関数 255

LOCASE 関数 256

LOG 関数 354

M

MAX 関数 354

MDY 関数 310

MIN 関数 354

MISSING 属性 370, 408, 433, 434, 435
ALLOWCVTERR パラメータ 408

N

NLS 443

NOPRINT 基軸 64

Null 値 433, 434, 435

O

OLAP 15, 66, 154

レポートの保存 66

OLAP ディメンション 154

OLAP フィールド名参照 71

OLAP インターフェースオプション 19

OLAP 階層 17, 20

プロシジャ定義 17

マスターファイル定義 17

OLAP 機能の削除 71

OLAP グラフ 56

OLAP グラフ作成 56

OLAP グラフ表示 56

OLAP コントロールパネル 19, 23, 26, 27, 31,
34, 38, 48, 50, 60

グラフウィンドウ 60

選択条件ウィンドウ 48

データ順序の指定 34

データ要素の回転 38

データ要素のソート 34

日付選択ウィンドウ 50

表示 19, 23, 26

分析 27

レポートリクエスト結果のグラフ化 60

OLAP コントロールパネル特有の機能 28

OLAP 選択条件適用 45, 50
日付 50

選択パネル 22

OLAP 選択パネル 22, 25, 47
条件の適用 47

OLAP ソートオプション 28

OLAP ツール 18, 19, 22, 23, 27
コントロールパネル 27
選択 18, 19

OLAP ツールの選択 18, 19

OLAP ディメンション要素 20, 34, 65
OLAP パネルから追加 65
データ順の入れ替え 34

OLAP データ 17, 29

OLAP データ選択オプション 28

OLAP データのソート 29

OLAP データのソート表示 32

OLAP ハイパーリンク 20
スタイルシート 20

OLAP パネル
表示 19, 26

OLAP フィールド参照 71

OLAP フィールド名 71

OLAP レポートフィールド値
積み重ね表示 63

OLAP 保存オプション 28

OLAP ボタン 26

OLAP 有効データソース 17

OLAP 有効レポート 18, 19

OLAP レポート 20, 24, 29, 30, 31, 32, 34, 38,
39, 48, 55, 60, 63, 65, 66, 72, 73, 74
Excel 形式で保存 74

- OLAP レポート ((続く))
 - PDF フォーマット表示 73
 - グラフ表示 60
 - 選択条件の適用 48
 - ソート 31, 32, 34
 - ソート順の変更 34
 - ソート条件の削除 32
 - データの回転 20, 38
 - データの削除 65, 66
 - データの追加 65
 - ピアグラフ 55
 - 非表示フィールドによるソート 39
 - フィールド値 29, 30, 31, 32, 63
 - フィールドの非表示 39
 - 保存 72, 73, 74
 - OLAP レポート基軸の積み重ね 28
 - OLAP レポートグラフ表示 24
 - OLAP レポート構文概要 17
 - OLAP レポート操作 24
 - OLAP レポートソートフィールドのピボット 20, 24, 38
 - OLAP レポートのディメンションコントロール 25
 - OLAP レポートディメンション要素 34, 65
 - ソート 34
 - OLAP レポートデータ制限 45
 - OLAP レポートのグラフ化 24, 56
 - OLAP レポートのソート 24, 30, 32, 39
 - OLAP レポートのディメンション要素
 - コントロールパネルから削除 66
 - レポートから削除 66
 - OLAP レポートのデータ順序の入れ替え 34
 - OLAP レポートのドリルダウン 24
 - OLAP レポートの日付範囲 46, 51, 52
 - 選択 46
 - 選択条件の適用 52
 - 適用 46
 - OLAP レポートの日付要素
 - 選択条件の適用 50
 - 追加 53
 - 日付範囲の適用 50, 52
 - OLAP レポートのフィールド値
 - ソート 30, 31, 32
 - OLAP レポートの保存 67, 72, 73, 74
 - OLAP レポート日付フィールド 51
 - 選択条件の指定 51
 - OLAP レポート日付要素 50, 54
 - 削除 54
 - OLAP レポート列 38
 - 回転 38
 - 再配置 38
 - OLAP レポート列の再表示 24
 - OLAP レポート列の非表示 24
 - OLAP レポートを Excel ファイルとして保存 74
 - OVERLAY 関数 257
- P**
- PARAG 関数 258
 - PCKOUT 関数 346
 - POSIT 関数 260
 - PRDNOR 関数 355
 - PRDUNI 関数 355
 - PTOA 関数 261
 - P データタイプ 388

Q

QUALCHAR パラメータ 374

R

RDNORM 関数 358

RDUNIF 関数 358

REVERSE 関数 262

RJUST 関数 262

S

SAVE ファイル 429, 430

SCD 120

SET パラメータ 370, 371, 373, 374, 407, 408,
411, 439

ACCBLN 439

ACCEPTBLANK 439

ALLOWCVTERR 408

DATEDISPLAY 407

DTSTANDARD 411

FIELDNAME 370, 371, 373

QUALCHAR 374

SmartDate 54, 399

SOUNDEX 関数 263

SPELLNUM 関数 265

SQL トランスレータと長いフィールド名 373

SUBSTR 関数 266

T

TILE フィールド 39, 41

TIMETOTS 関数 335, 362

TITLE 属性 370, 440, 441

TODAY 関数 336, 362

TX データタイプ 427

U

UPCASE 関数 268

USAGE 属性 370, 383, 384

USAGE フォーマット 431, 432

W

Web 検索 15, 16

Web サイトの検索 16

Web のクエリ 16

Y

YMD 関数 310

YM 関数 337

YRTHRESH 属性 370

YRTHRESH 値 54

あ

アクセス制限 206

アクセスの制限
 データ 206
 アクセスファイル
 定義 78
 アクセスファイルテキスト表示タブ 111
 値 269
 値のデコード 269, 270, 271
 値フィールド 138
 アダプタ 79, 80
 ウィザードを使用しない構成 80

い

異常値 138, 227
 一時項目 178, 373, 449, 450, 451, 461
 作成 450, 451
 修飾フィールド名 451
 長いフィールド名 373, 450
 一時項目 (COMPUTE) 177
 一時項目 (COMPUTE) の追加 178
 一時項目 (DEFINE)
 修飾フィールド名 451
 長いフィールド名 450
 一時項目 (DEFINE) の OCCURS 180
 一時項目 (DEFINE) の OCCURS 繰り返し 180
 一時項目の作成 178
 一時項目の属性 185
 一時フィールド 178
 インパクト分析 214
 インパクト分析 (Impact Analysis) 216

え

円グラフ 58
 演算接頭語 44

お

オートドリルダウン 18, 19, 29
 大文字と小文字の変換 268
 オンライン分析処理 (OLAP) 15

か

階層 20
 階層データ構造 436
 外部関数 229
 拡張通貨記号 393, 394
 フォーマット 394
 カラムの管理 93
 環境変数 360
 環境変数値の取得 360
 関係演算子 44, 46, 48
 関係ボタン 47, 48, 50
 勘定科目の階層 438
 勘定科目表の階層 438
 関数 229, 313, 315, 318, 359
 FIQTR 315
 FIYR 313
 FIYYQ 318
 関数カテゴリ 230, 269, 347
 関数の引数 237

き

- 基軸コントロール
 - 選択パネル 47, 64
- 基軸制御 25
- 基軸ソート 22, 30
- 基軸チェックボックス 31
- 基軸に基づくソート 30, 31, 32
 - ソート条件の削除 32
 - ランキング値 31
- 基軸の積み重ね 63
- 基軸のドリルダウン 18, 19, 29
- 基軸の表示 63
 - 積み重ね 63
- 基準日 407
- 既存フィールドのグループフィールドへの追加 196
- 行 38
- 共通フィールド名参照 71

く

- クラスタ JOIN 82, 145
 - 作成 145
- グラフ
 - OLAP 56
 - OLAP 表示 56, 58, 60
- グラフウィンドウ 60
- グラフコントロール
 - コントロールパネル 56
 - 選択ウィンドウ 56
- グラフスタイル 58, 60
 - OLAP での制御 58, 60

- グラフ制御 25
- グラフタイプ
 - OLAP 56, 58
 - 円 58
 - 縦折れ線 58
 - 縦棒 58
 - 縦面 58
 - 横折れ線 58
 - 横棒 58
 - 横面 58
- グラフ表示チェックボックス 60
- グラフボタン 60
- グループフィールドからのフィールド削除 197
- グループフィールド属性 198
- グループフィールドの追加 194
- グループフィールドへの新規フィールド追加 196
- グローバル変数 465

け

- 経過時間 337
- 傾向 55
 - グラフ表示 55
- 傾向の視覚化 55
- 言語 443
- 現在時間の取得 327, 362
- 現在日付の取得 362

こ

- 降順ソート
 - 基軸 30, 31

降順ソート (続く)
 ディメンション 34

昇順ソート 30, 31
 基軸 30, 31

降順ソートラジオボタン 31

国際日付フォーマット 290

小文字変換 256

コントロールパネル 23, 27, 31, 34, 48, 50, 60
 グラフウィンドウ 60
 選択条件ウィンドウ 48
 日付選択ウィンドウ 50
 分析 27
 レポートリクエスト結果のグラフ化 60

コントロールパネル特有の機能 28

さ

最小値の検索 354

最大整数 353

最大値の検索 354

再配置
 ディメンション 38

財務レポート 438

サブタイトル非表示 30, 31

サブルーチン 229

し

システム関数 237, 360

自然対数 354

指定範囲チェックボックス 50, 54

自動ソート 30

シノニム 81, 83, 125, 218

シノニムエディタ 83, 84, 85, 87, 106, 108, 109,
 110, 111, 145, 153, 214

 画面構成 84

 クラスタ JOIN 153

シノニムエディタツールバー 87

シノニムエディタによる一時項目の作成 167

シノニムエディタの一時項目 (COMPUTE) 177

シノニムエディタのシノニム属性 111

シノニムエディタの多言語タイトルと説明 121

シノニム属性 111

シノニム属性の表示と編集 111

シノニムの拡張 82

シノニムの作成 79, 125

シノニムのデータプロファイリング 136, 218

修飾子 374

修飾フィールド名 373, 374, 375, 376, 379, 451
 一時項目 451
 一時項目 (DEFINE) 451
 修飾のレベル 379

剰余計算 351

す

数値 347, 348

数値関数 236, 347, 348, 350, 351, 352, 353, 354

数値データ
 分位グループ 39, 41

数値データタイプ 384, 385, 386, 387, 388, 389, 391

整数 385

単精度浮動小数点数 387

倍精度浮動小数点数 386

パック 10 進数 388

表示オプション 389, 391

数値引数 238

数値表示オプション 389, 391

数値フォーマット 341

数値列 230

スケール 348

スタイルシート 20

OLAP ハイパーリンク 20

スノーフレイク 147

すべてのフィールド 138, 219, 223

せ

世紀をまたがる日付 54

整数データタイプ 385

セグメントからのグループ削除 197

セグメント属性概要 114

セグメント表示タブ 108

絶対値 348

選択条件 25, 46, 47, 48, 50, 51, 52

OLAP 25

OLAP 選択パネルからの適用 47

OLAP レポートへの適用 48

削除 50

データ範囲への適用 46, 48

適用 46, 48

選択条件 (続く)

日付範囲への適用 52

日付要素への適用 50, 51

選択条件パネル 48

選択リストボックス 53, 54

日付の削除 54

先頭大文字変換 252

そ

ソースデータ 77, 200, 206

ソート 29

ソート順

数値データグループ化 39, 41

ソートチェックボックス 31

ゾーン 10 進数変換 345

た

ダイアログマネージャ 409, 461

マスターファイルの DEFINE の変数 461

代替ファイルビュー

長いフィールド名 373

タイムスタンプデータタイプ 410, 413

タイムスタンプ変換 335

太陽暦日付 54

太陽暦日付の変換 334

多言語の指定 443

多言語のメタデータ 443

多言語メタデータ

使用時の注意 443

縦グラフ 58

単一フィールド 138, 224, 226, 227

単精度浮動小数点数データタイプ 387

ち

抽出ファイル

日付データタイプ 409

重複したフィールド名 373, 374, 375, 376

つ

通貨記号 393, 394

拡張通貨記号 394

通貨のフォーマット設定 393

通貨表記の書き替え 265

積み重ね基軸の表示 63

て

定義 78

ディメンション 20, 34, 38, 50, 65, 66

昇順ソート 34

再配置 38

再表示 65

削除 66

定義 50

ディメンション階層 48

ディメンションの削除 156

ディメンションビルダ 154, 157

ツールバー 157

データ記述

フィールド宣言 370

データ順序の指定 34

データ制限 45

データセキュリティ 200, 206

データセキュリティ設定 200, 206

データソース 17, 269, 370

OLAP 17

フィールドの記述 370

データソース関数 232, 269, 272

データソース記述 77

データソース値 269

データソース値の抽出 272

データソースの解釈 78, 82

データソースの書き込みアクセス 206

データソースの記述 77, 370, 436

FML 階層 436

フィールド宣言 370

データソースのパス 200

データソースの読み取り/書き込みアクセス 206

データソースの読み取りアクセス 206

データタイプ 383, 384, 385, 386, 387, 388,
389, 391, 395, 396, 399, 403, 404,
405, 409, 410, 413, 427, 428, 429,
431, 432, 433

数値表示オプション 389, 391

整数 385

単精度浮動小数点数 387

テキスト 427

倍精度浮動小数点数 386

パック 10 進数 388

日付 396, 403, 404, 405, 409

日付時間 410, 413

日付の格納 399

文字 395, 396

データタイプの用語 78
データの視覚化
棒グラフ 55
データプロファイリング 136, 138, 218, 219,
223, 224, 226, 227
統計 138
データプロファイリング値 226
データベース記述
フィールド宣言 370
テキスト大文字変換 268
テキストデータタイプ 373, 427
長いフィールド名 373
テキスト表示タブ 110
テキスト分割 258
デコード関数 232, 269, 270, 271

と

統計 224

な

内部関数 229
内部表現
日付 407
長いフィールド名 373, 374, 375, 376, 450
CHECK FILE コマンド 373
一時項目 374, 450
インデックスフィールド 373
代替フィールドビュー 373
長い文字フィールド 396
長さの計算 242
ナノ秒単位の日付時間値 298

名前
フィールド 371, 372, 373, 374

は

倍精度フォーマット変換 343
倍精度浮動小数点数データタイプ 386, 387, 388
パスワード 200, 210
パスワードの削除 210
パターン列 138, 223
パック 10 進数数値変換 346
パック 10 進数データタイプ 388
パック 10 進数フィールド 388
パック 10 進数フィールドの検査 350
パック 10 進数変換 344

ひ

引数の種類 237
引数のフォーマット 238
ビジネスビュー 158, 162
ビジネスビューの作成 162
ビジネスビューマスターファイル 166
日付 53
OLAP レポートでの指定 53
日付計算 405
日付差計算 285, 310
日付時間関数 232, 276
日付時間構成要素の取得 330, 331

- 日付時間精度 298
 - 日付時間値の格納 327
 - 日付時間値の差 276, 277
 - 日付時間値の差計算 325
 - 日付時間値の精度 298
 - 日付時間値の設定 329
 - 日付時間値の増加 321
 - 日付時間値の挿入 332
 - 日付時間値の追加 276, 277
 - 日付時間値の変換 333
 - 日付時間値フォーマット変換 334
 - 日付時間データタイプ 410, 429, 430
 - HOLD ファイル 429, 430
 - SAVE ファイル 429, 430
 - 日付選択ウィンドウ 54
 - 日付タイプデータの格納 399
 - 日付値の変換 406
 - 日付データタイプ 396, 397, 399, 402, 403, 404, 405, 406, 407, 408, 409
 - RECAP コマンド 409
 - 値の変換 406
 - 格納 399
 - 区切り記号 402
 - グラフでの考慮事項 409
 - 計算 405
 - ダイアログマネージャ 409
 - 抽出ファイル 409
 - 内部表現 407
 - 表示オプション 397
 - 標準外のフォーマット 408
 - 変換 403
 - 日付データタイプ (続く)
 - リテラル 399, 404, 405
 - 日付の移動 287
 - 日付の区切り記号 402
 - 日付の取得 336
 - 日付の変換 403
 - 日付のリテラル 399, 404, 405
 - 日付範囲の指定 50
 - 日付引数 238
 - 日付表示オプション 397
 - 日付フィールドの計算 405
 - 日付フォーマット 50, 51, 54, 283
 - SmartDate 54
 - 指定 50
 - 制限 54
 - ユリウス暦 54
 - 日付フォーマット変換 284, 324
 - 日付変換 403, 406
 - 日付文字列表示 278
 - 日付リテラル 399, 404, 405
 - 表示オプション 383, 384, 389, 391, 397
 - 数値 389, 391
 - 日付 397
 - 標準日付時間関数 232, 276
- ふ
- ファイル記述 77, 78, 370
 - フィールド宣言 370
 - ファイル属性の概要 113

- フィールド 44, 216, 369, 370, 371, 372, 373, 374, 375, 376, 441, 442
 - OLAP レポート 演算 44
 - 説明 441, 442
 - 名前 371, 372, 373, 374, 375, 376
 - 記述 370
- フィールドエイリアス 380, 381
- フィールド演算 44
- フィールド属性概要 117
- フィールドタイトルの置換 440, 441
- フィールド値 206, 438, 439
 - 確認 438, 439
- フィールド値の確認 438, 439
- フィールド値へのアクセス制限設定 206
- フィールドのエイリアス 380, 381
- フィールドの説明 441, 442
- フィールドの表示フォーマット 383, 384, 385, 386, 387, 388, 389, 391, 395, 396, 397, 399, 403, 404, 405, 409, 410, 413, 427, 428, 431, 432, 433
 - 数値表示オプション 389, 391
 - 整数 385
 - 単精度浮動小数点数 387
 - テキスト 427
 - 倍精度浮動小数点数 386
 - パック 10 進数 388
 - 日付 396, 403, 404, 405, 409
 - 日付時間 410, 413
 - 日付の格納 399
 - 日付表示オプション 397
 - 文字 395, 396
- フィールド表示タブ 85, 106
- フィールドフォーマット 383, 384, 385, 386, 387, 388, 389, 391, 395, 396, 397, 399, 403, 404, 405, 409, 410, 413, 427, 428, 429, 431, 432, 433
 - 数値表示オプション 389, 391
 - 整数 385
 - 単精度浮動小数点数 387
 - テキスト 427
 - 倍精度浮動小数点数 386
 - パック 10 進数 388
 - 日付 396, 403, 404, 405, 409
 - 日付時間 410, 413
 - 日付の格納 399
 - 日付表示オプション 397
 - 文字 395, 396
- フィールドフォーマット変換 340
- フィールドプロシジャの特定 216
- フィールド名 371, 372, 373, 374, 375, 376
 - 修飾 373, 374, 375, 376
- フィールド名参照 71
- フィルタ 171
- フィルタの演算 190
- フィルタの作成 171
- フォーマット 338
- フォーマット変換 278, 324, 326, 338, 341, 342
- フォーマット変換関数 235
- 複数グラフ
 - OLAP 60
- プロシジャ OLAP 階層定義 17
- プロシジャの暗号化
 - マスターファイル 216
- 分位フィールド 39, 41

へ

- 平方根 359
- べき乗 352
- 別のフィールドタイトル 440, 441
- 編集オプション 383, 384, 389, 391, 397
 - 数値 389, 391
 - 日付 397
- 変数 461
 - マスターファイル 461
 - マスターファイルの DEFINE 461

ほ

- 棒グラフ 348

ま

- マスターファイル 17, 50, 54, 78, 110, 111, 153, 156, 200, 214, 216, 370, 436, 438, 443, 455
 - DEFCENT 値と YRTHRESH 値の指定 54
 - Y2K 属性 370
 - 階層 436
 - 新規作成 153
 - 多言語の記述 443
 - ディメンションの定義 17, 50
 - 日付フォーマットの指定 50
 - フィルタ 455
 - プロシジャの識別 214
- マスターファイル WITHIN 属性 17
- マスターファイルでの階層の定義 17
- マスターファイル内の COMPUTE 452
- マスターファイル内のフィルタ 455

- マスターファイルの COMPUTE 452
- マスターファイルの Y2K 属性 370
- マスターファイルの印刷 110, 111
- マスターファイルのグローバル変数 465
- マスターファイルのセキュリティ設定 200
- マスターファイルの編集 156

み

- ミッシング値 433, 434, 435

も

- 文字 249, 250
- 文字関数
 - LCWORD2 253, 255
 - LCWORD3 254
- 文字検索 260
- 文字データタイプ 395, 396
- 文字引数 238
- 文字日付 469
- 文字日付の変換 469
- 文字フィールド 230
- 文字フォーマット 323, 330, 341
- 文字変換 244, 246, 342
- 文字列 242, 338
- 文字列音声比較 263
- 文字列関数 230, 242
- 文字列の位置揃え 255, 262

文字列の一部抽出 249, 250, 266

文字列の上書き 257

文字列の順序入れ替え 262

文字列の中央揃え 248

文字列の抽出 249

文字列の分割 258

文字列の右揃え 262

文字列ビット 243

文字列フォーマット 245

文字列変換 252, 328, 338

文字列を別の文字で上書き 257

モデル表示タブ 109, 125, 154

ゆ

ユーザ ID 361

ユーザ ID の取得 361

よ

曜日検索 311

横グラフ 58

ら

ランキングチェックボックス 31

乱数の生成 355, 358

れ

レポート

 OLAP 67

 OLAP 有効 18, 19

レポートデータの検索 15

レポートデータの分析 24

レポートの OLAP 分析 24

レポートの保存 66, 71

 OLAP 機能の除外 71

ろ

論理ビューの作成 154