

IBM DB2 Cube Views



# ビジネス・モデリング・シナリオ

バージョン 8

**ご注意!**

本書および本書で紹介する製品をご使用になる前に、31 ページの『特記事項』に記載されている情報をお読みください。

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

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

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

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

(URL は、変更になる場合があります)

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

原典：	SC18-7803-00 IBM DB2 Cube Views Business Modeling Scenarios Version 8
発行：	日本アイ・ビー・エム株式会社
担当：	ナショナル・ランゲージ・サポート

第1刷 2003年 7月

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

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

© Copyright International Business Machines Corporation 2003. All rights reserved.

© Copyright IBM Japan 2003

---

# 目次

本書について . . . . .	v	詳細なシナリオ . . . . .	19
本書の対象読者 . . . . .	v	計測の作成ステップ . . . . .	20
<b>第 1 章 一定期間にわたる倉庫内の在庫の流れ と在庫の値の計算 . . . . .</b>	<b>1</b>	<b>第 5 章 販売額のランキング . . . . .</b>	<b>23</b>
詳細なシナリオ . . . . .	1	詳細なシナリオ . . . . .	23
計測の作成ステップ . . . . .	3	計測の作成ステップ . . . . .	25
<b>第 2 章 販売額に対する広告宣伝費の相関 . . . . .</b>	<b>7</b>	<b>第 6 章 ファクト表に保管された時間データ を使用した時間ディメンションの作成 . . . . .</b>	<b>27</b>
詳細なシナリオ . . . . .	7	詳細なシナリオ . . . . .	27
計測の作成ステップ . . . . .	7	属性とディメンションの作成ステップ . . . . .	29
<b>第 3 章 商店の利益と利ざやの計算 . . . . .</b>	<b>13</b>	<b>特記事項 . . . . .</b>	<b>31</b>
詳細なシナリオ . . . . .	13	商標 . . . . .	33
計測の作成ステップ . . . . .	13		
<b>第 4 章 インターネット・オーダー数のカウ ント . . . . .</b>	<b>19</b>		



---

## 本書について

本書は、「DB2 Cube Views セットアップおよびユーザーズ・ガイド」の内容を補足する資料であり、DB2 Cube Views メタデータを使って、よくある実際の業務シナリオをモデル化する方法を説明しています。

---

## 本書の対象読者

OLAP メタデータおよび DB2 Universal Database™ (DB2) を使用するデータベース管理者 (DBA) の方は、本書をお読みください。本書は、読者が以下の事項について理解されているものとして記述されています。

- (スタースキーマなどの) OLAP の概念。
- DB2 Cube Views メタデータ・オブジェクト (キューブ・モデル、ファクト・オブジェクト、ディメンション、結合、計測、および属性など)。



---

## 第 1 章 一定期間にわたる倉庫内の在庫の流れと在庫の値の計算

小売企業の XYZ Retail 社は、在庫品を、販売のために特定の店舗に発送するまで倉庫に保管しています。XYZ Retail は、一定期間にわたる倉庫での在庫状態にデータに関するデータを保守して、そのデータを分析したいと考えています。同社は特に、倉庫を以下の 2 つの側面から調べようとしています。

- 倉庫における商品の出庫と入庫の流れ。
- 任意の時点における倉庫内の商品の値。

第 1 の側面である商品の流れに関しては、ある一定期間中のデータを調べることも必要です。第 2 の側面では、ある特定の時点での倉庫のスナップショットが商品の値によってとられます。

---

### 詳細なシナリオ

XYZ Retail には、QUANTITY\_IN、QUANTITY\_OUT、CURRENT\_QUANTITY、PRODUCT\_VALUE、PRODUCT\_ID、および TIME\_ID という倉庫関連の列から成るファクト表があります。このデータは、週単位で表に入力されます。このデータベースにはまた、商品表と時間表もあります。例として、一連のファクト表のデータのサンプルを表 1 に示してあります。

表 1.

PRODUCT_ID	TIME_ID	QUANTITY_ID	QUANTITY_OUT	CURRENT_QUANTITY	PRODUCT_VALUE
1234	1	5	0	5	5
1234	2	20	10	15	5
1234	3	10	20	5	5

3 つのサンプル・データ入力項目のそれぞれの PRODUCT\_ID 値は同じです。それは、1 つの商品タイプを何度も倉庫から出し入れする可能性があるからです。

XYZ Retail の DBA は、以下の 3 種類の計測を作成する必要があります。

#### フロー・イン

倉庫に入庫される商品の流れをモデル化します。

#### フロー・アウト

倉庫からの出庫の流れをモデル化します。

**現行値** 任意の時点における商品の値をモデル化します。

DBA がフロー・インとフロー・アウトの最初の 2 つの計測を作成するには、それぞれ QUANTITY\_IN 列と QUANTITY\_OUT 列にマップされる計測を作成して、すべてのディメンションのデータを合計します。これは、完全加法計測と呼ばれます。それは、すべてのディメンションにわたって SUM 関数だけを使ってデータが集約されるからです。たとえば、表 2 は、1234 という PRODUCT\_ID の付いた商品の 3 か月間分の QUANTITY\_IN 列と QUANTITY\_OUT 列の一連のサンプル・データを示しています。フロー・インとフロー・アウトの計測では、このような月単位の値を合計して、四半期中に倉庫に出入庫した合計数量を計算します。

表 2. PRODUCT\_ID 1234 のフロー・インとフロー・アウトの完全加法計測でのサンプル・データの計算

	1 月	2 月	3 月	第 1 四半期
QUANTITY_IN	5	20	10	35
QUANTITY_OUT	0	10	20	30

完全加法計測は、最も簡単かつ最も一般的に作成される計測ですが、これは多くの場合、さらに複雑な計測の土台として使用されます。数値ソース・データをベースとする計測では、OLAP センターはデフォルトで完全加法計測を作成します。

DBA は、3 番目の計測 (現行値) を作成する際、PRODUCT\_VALUE に CURRENT\_QUANTITY を掛けて算出された計算済み計測を作成します。たとえば、PRODUCT\_ID=1234 の商品の値が 5 の場合、サンプル・データの現行値の計測は表 3 に示されているとおりになります。

表 3. PRODUCT\_ID 1234 の現行値計測でのサンプル・データの計算

	1 月	2 月	3 月
CURRENT_QUANTITY	5	10	20
現行値	25	50	100

次に、すべてのディメンションにわたってこのデータを集約する必要があります。ただしこの計測では、ある 1 つの時点における値が計算されるので、時間ディメンションをまたいで合計しても意味を成しません。よってこの集約では、商品ディメンションをまたいでデータが合計されて、時間の経過に沿ったデータの平均値が算出されます。これは、半加法計測と呼ばれます。集約のうちの一部にしか SUM 関数が関与しないからです。

スナップショット・データ、つまり、月単位の在庫データなどのある特定の時点を表わすデータを計算する計測は、多くの場合は半加法計測です。なぜなら、月の値を四半期の値に加えても無意味だからです。ある商品がある四半期の間ずっと倉庫にとどまっていた場合、その商品は、その四半期の 3 か月間、毎月の在庫の CURRENT\_QUANTITY

スナップショット・データに組み入れられます。一定期間中の CURRENT\_QUANTITY データが合計されるときには、倉庫内に 3 か月間とどまっている商品は 3 回カウントされます。表 4 に示されているとおり、第 1 四半期の値 25 には、倉庫のアクティビティに対する意義はまったくありません。倉庫内に 25 個の商品があったことはないので、25 個の商品の値を計算しても無駄であることをこの表は示します。

表 4. PRODUCT\_ID 1234 に対して時間ディメンションで SUM 関数を使った CURRENT\_QUANTITY 列のサンプル・データの計算

	1 月	2 月	3 月	第 1 四半期
SUM(CURRENT_QUANTITY)	5	15	5	25

すべてのディメンションにまたがって SUM 関数を使う代わりに、時間ディメンションに対して AVG、MIN、および MAX などの他の集約関数を実行することができます。たとえば、1 月、2 月、および 3 月の一連の同じサンプル・データに対して、表 5 に示されている時間ディメンションの別の集約関数を使って、この四半期の妥当な値を作成することができます。現行値計測を使って、この四半期中に倉庫に保管されていた商品の平均合計値を表したり、同じ四半期中の任意の時点における最大値または最小値を表したりすることができます。

表 5. PRODUCT\_ID 1234 に対して時間ディメンションで、AVG、MAX、および MIN 関数を使った CURRENT\_QUANTITY 列のサンプル・データの計算

	1 月	2 月	3 月	第 1 四半期
AVG(CURRENT_QUANTITY)	5	15	5	8.3
MAX(CURRENT_QUANTITY)	5	15	5	15
MIN(CURRENT_QUANTITY)	5	15	5	5

## 計測の作成ステップ

以下のステップは、OLAP センターの「ファクト・プロパティ」ウィンドウを使って、既存のファクト・オブジェクト内にフロー・イン、フロー・アウト、および現行値の計測を作成する方法を説明しています。

1. 「ファクト・プロパティ」を開くには、OLAP センター・オブジェクト・ツリー内のファクト・オブジェクトを右マウス・ボタンでクリックしてから、「計測の編集」をクリックします。「ファクト・プロパティ」ウィンドウが開きます。
2. 次のようにして、フロー・イン計測を作成します。

- a. 「計測」ページで「**計算済み計測の作成**」をクリックして、フロー・イン計測を作成します。「SQL 式ビルダー」ウィンドウが開きます。
  - b. 「SQL 式ビルダー」ウィンドウで、「名前」フィールドに FLOW IN と入力します。
  - c. フロー・イン式を作成するには、以下のステップを行います。
    - 「列」フォルダーを拡張表示して、「データ」リスト内のファクト表を拡張表示します。
    - 「QUANTITY\_IN」列をダブルクリックして、式に追加します。
    - 「OK」をクリックして、「SQL 式ビルダー」ウィンドウを閉じます。「集約」ページで、デフォルトの集約関数 SUM を変更する必要はありません。SUM 関数は、フロー・イン計測のデフォルトです。それは、データ・ソースは数値であって、既存の計測だけでなく、列も参照するからです。
3. 次のようにして、フロー・アウト計測を作成します。
- a. 「計測」ページで「**計算済み計測の作成**」をクリックして、フロー・アウト計測を作成します。「SQL 式ビルダー」ウィンドウが開きます。
  - b. 「SQL 式ビルダー」ウィンドウで、「名前」フィールドに FLOW OUT と入力します。
  - c. フロー・アウト式を作成するには、以下のステップを行います。
    - 「列」フォルダーを拡張表示して、「データ」リスト内のファクト表を拡張表示します。
    - 「QUANTITY\_OUT」列をダブルクリックします。
  - d. 「OK」をクリックして、「SQL 式ビルダー」ウィンドウを閉じます。「集約」ページで、デフォルトの集約関数 SUM を変更する必要はありません。SUM 関数は、フロー・アウト計測のデフォルトです。それは、データ・ソースは数値であって、既存の計測だけでなく、列も参照するからです。
4. 次のようにして、現行値計測を作成します。
- a. 「計測」ページで「**計算済み計測の作成**」をクリックして、現行値計測を作成します。「SQL 式ビルダー」ウィンドウが開きます。
  - b. 「SQL 式ビルダー」ウィンドウで、「名前」フィールドに CURRENT VALUE と入力します。
  - c. 現行値式を作成するには、以下のステップを行います。
    - 「列」フォルダーを拡張表示して、「データ」リスト内のファクト表を拡張表示します。
    - 「データ」リストの「PRODUCT\_VALUE」列をダブルクリックします。
    - 「演算子」リストの \* 演算子をダブルクリックします。
    - 「データ」リストの「CURRENT\_QUANTITY」列をダブルクリックします。

図1は、「SQL 式ビルダー」ウィンドウで作成できる現行値式を示しています。

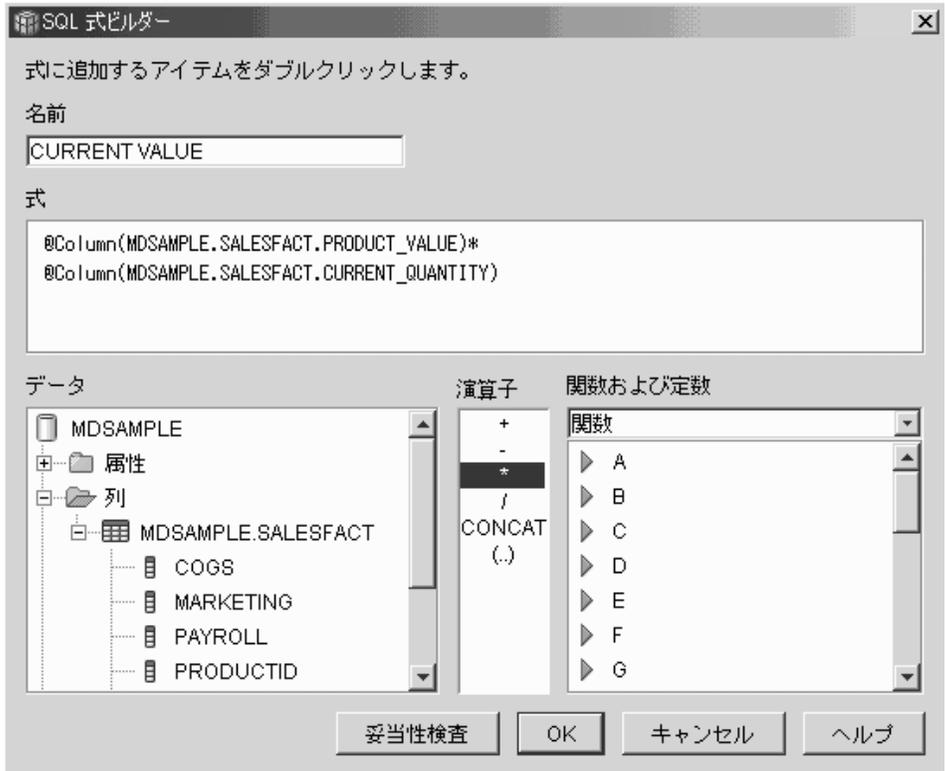


図1. 「SQL 式ビルダー」ウィンドウの完全な現行値式

- d. 「OK」をクリックして、「SQL 式ビルダー」ウィンドウを閉じます。
- e. 「集約」ページで、現行値計測の集約をクリックしてから、リストで「集約スクリプト」をクリックします。「集約スクリプト・ビルダー」ウィンドウが開きます。デフォルトの集約スクリプトには、どのディメンションにも使用される SUM 関数があります。
- f. 必要があれば、「Time (時間)」を選択して時間ディメンションを下に移動してから、 プッシュボタンをクリックして、このディメンションがスクリプトの最後にリストされるようにします。次に、「Time (時間)」ディメンションを選択状態にしたまま、「列関数」リストの **AVG** 関数をダブルクリックします。6 ページの図2 に示されている集約スクリプトは、データの平均をとるために、時間以外のすべてのディメンションからのデータを合計します。

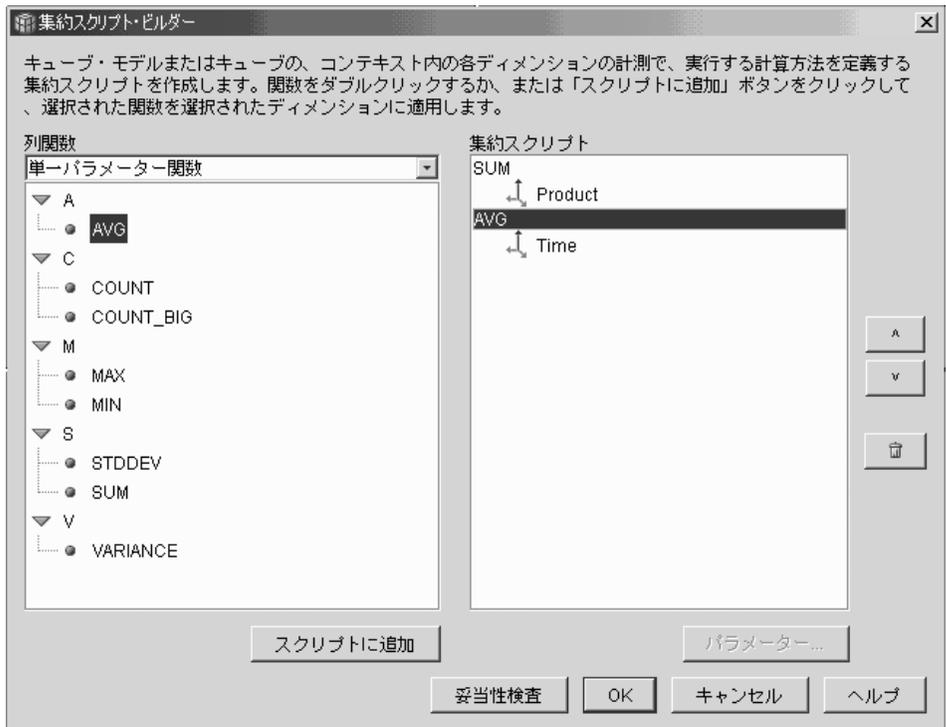


図2. 現行値計測の集約スクリプト

- g. 「集約スクリプト・ビルダー」ウィンドウで「妥当性検査」をクリックして、集約スクリプトが有効であることを検証します。「OK」をクリックして集約スクリプトを保管してから、ウィンドウを閉じます。
5. 「OK」をクリックして、ファクト・オブジェクトに加えた変更を保管し、「ファクト・プロパティ」ウィンドウを閉じます。

これで、倉庫の在庫に関する 3 通りの計算済みの計測が作成されました。これらの計測を使って、倉庫に出入りする商品の流れのパターンを分析することができます。

---

## 第 2 章 販売額に対する広告宣伝費の相関

ある自動車ディーラーが、自社の広告宣伝費の増額を検討しています。このディーラーは、妥当な決断を下すために、まず広告費と販売額のこれまでの関係を分析したいと考えています。ディーラーは、広告宣伝費の増減によって販売額が影響を受けたことがあるかどうかという点、特に、広告宣伝費の増額が販売額の増加と密接な関係にあるのかどうかを確かめたいと思っています。

---

### 詳細なシナリオ

このディーラーのデータベースには、「販売額」および「広告宣伝費」という列から成るファクト表があります。またこのデータベースには他に、いくつかのディメンション表もあります。DBA は、DB2 CORRELATION 関数を用いる計測を作成して、経費と販売額の相関計算を実行することができます。CORRELATION 関数は、2 つの入力パラメーターを必要とする複数パラメーター関数です。この場合、DBA は、「販売額」列と「広告宣伝費列」を 2 つの入力パラメーターとして使用します。

DBA は、最初に集約スクリプト内で複数パラメーター集約関数を用いる必要があります。複数パラメーター関数は、すべてのディメンションに対して適用できますが、時間ディメンション以外のすべてのディメンションにまず適用した後に MAX 関数などの別の関数を時間関数に対して適用することもできます。DBA は、計測用の SQL 式を定義して、それが「広告宣伝費」列に直接マップされるようにします。この SQL 式は、複数パラメーター関数で使用する 2 つのパラメーターのうちの最初のもので、DBA は「販売額」列に直接マップされる SQL 式として、2 番目のパラメーターを定義します。CORRELATION 関数は、唯一の集約関数として定義されます。それは、すべてのディメンションにまたがって、広告宣伝費と販売額結果の統計上の相関関係を計測で計算できるようにするためです。

---

### 計測の作成ステップ

以下のステップは、OLAP センターの「ファクト・プロパティ」ウィンドウを使って、既存のファクト・オブジェクト内に広告宣伝/販売額の関係の計測を作成する方法を説明しています。

1. OLAP センター・オブジェクト・ツリー内のファクト・オブジェクトを右マウス・ボタンでクリックしてから、「計測の編集」をクリックして、「ファクト・プロパティ」を開きます。
2. 「計算済み計測の作成」プッシュボタンをクリックします。「SQL 式ビルダー」ウィンドウが開きます。

3. 「SQL 式ビルダー」ウィンドウで、「名前」フィールドに ADVERTISING-SALES CORRELATION と入力します。
4. 計測の式を定義します。なおこの式は、集約スクリプト内で CORRELATION 関数の最初のパラメーターとしても使用されます。この式を定義するには、「データ」リストの「計測」フォルダーを拡張表示して、「AD COSTS」計測をダブルクリックし、これを「式」リストに追加します。図3は、「SQL 式ビルダー」ウィンドウで作成できる現行値式を示しています。

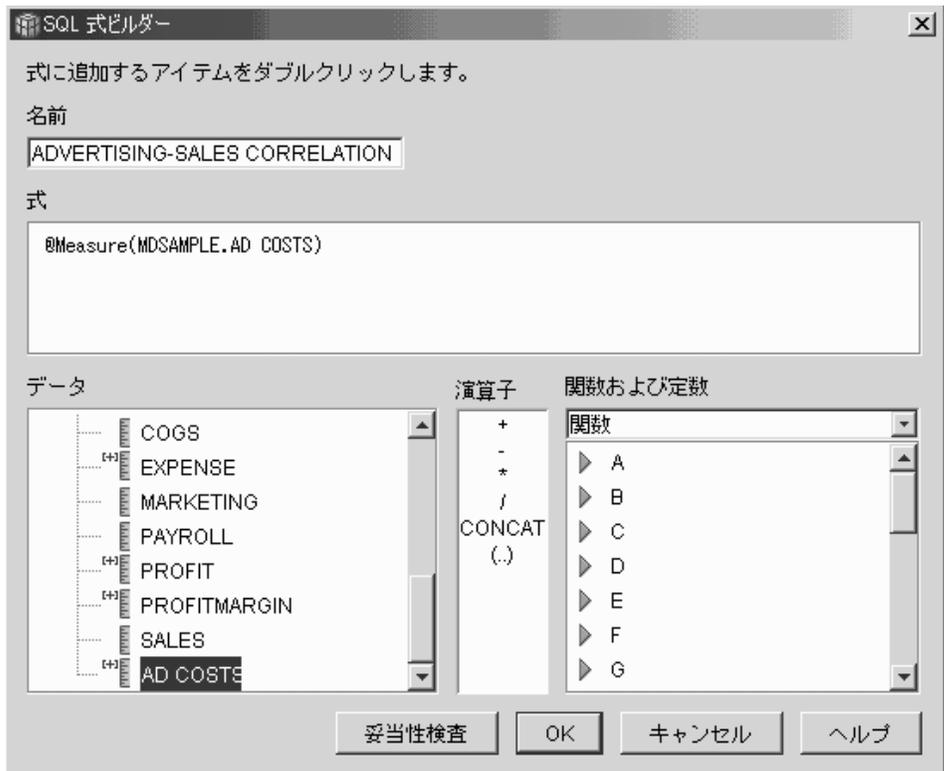


図3. 「SQL 式ビルダー」ウィンドウの完全な広告宣伝/販売額の相関式

5. 9 ページの図4 に示されているとおり、「集約」ページで、**ADVERTISING-SALES CORRELATION** 計測の集約をクリックしてから、リストで「集約スクリプト」をクリックします。「集約スクリプト・ビルダー」ウィンドウが開きます。



図4. 「ファクト・プロパティー」ウィンドウの「集約」ページ

- 「列関数」フィールドで、「複数パラメーター関数」を選択します。複数パラメーター関数のリストで **CORRELATION** 関数を選択してから、「スクリプトに追加」をクリックします。10 ページの図5 は、スクリプトのディメンション・リストの先頭に置かれた **CORRELATION** 関数を示しています。

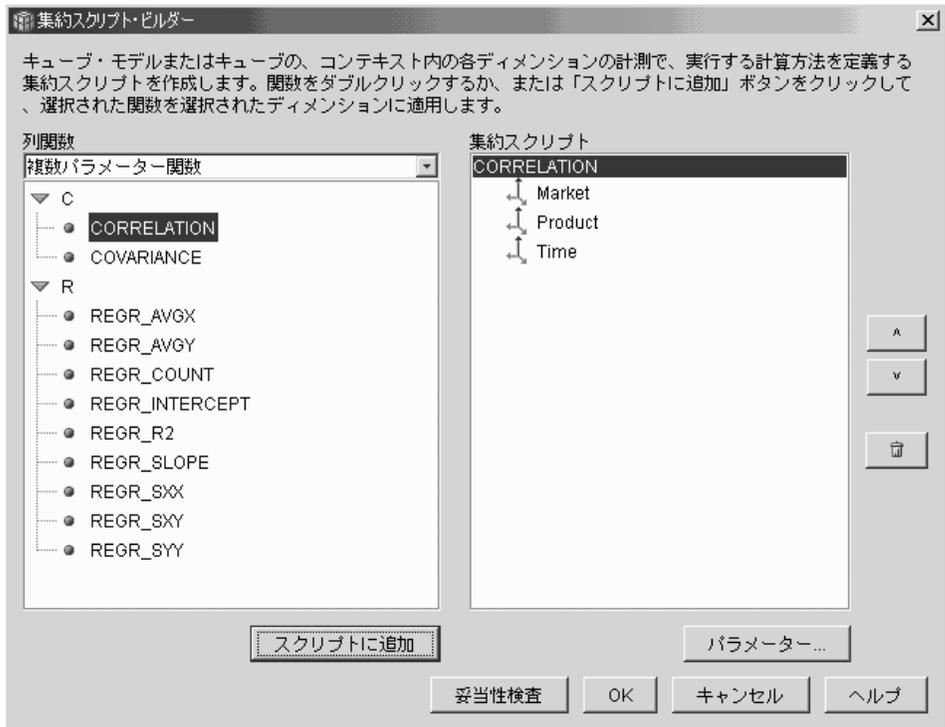


図 5. 広告宣伝/販売額の関係の計測の集約スクリプト

7. 「パラメーター」プッシュボタンをクリックして、CORRELATION 関数の 2 番目のパラメーターを指定します。「関数パラメーター」ウィンドウが開きます。「既存の計測の使用」を選択してから、「SALES (販売額)」を選択します。11 ページの図 6 は、2 番目のパラメーターとして販売額計測が指定された「関数パラメーター」ウィンドウを示しています。「OK」をクリックして選択内容を保管し、「関数パラメーター」ウィンドウを閉じます。

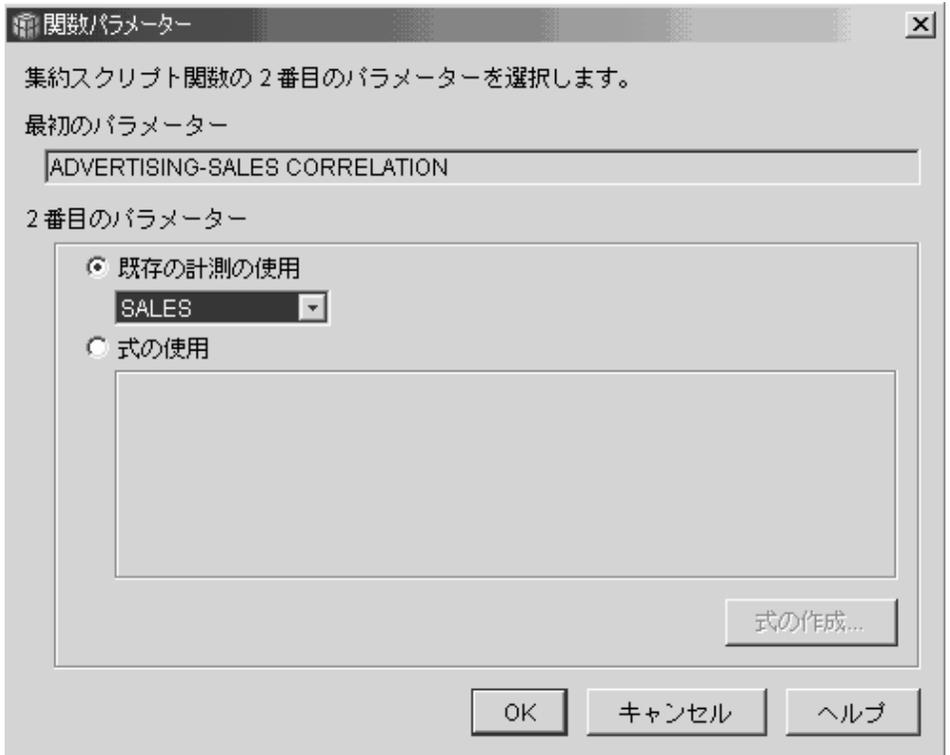


図 6. 2 番目のパラメーターとして販売額計測が指定された「関数パラメーター」ウィンドウ

- 「集約スクリプト・ビルダー」ウィンドウで「**妥当性検査**」押しボタンをクリックして、集約スクリプトが有効であることを検証します。「**OK**」をクリックして集約スクリプトを保管してから、ウィンドウを閉じます。
- 「**OK**」をクリックして、ファクト・オブジェクトに加えた変更を保管し、「ファクト・プロパティ」ウィンドウを閉じます。

これで、データベース内で 2 つのタイプのデータを相互に関連付ける計測が作成されました。この計測を使って、各種の結果におけるこれまでの傾向に基づいて今後の広告宣伝費に関する決断を下すことができます。



---

## 第 3 章 商店の利益と利ざやの計算

あるおもちゃ店の店長が、年間のいずれかの時節や商品のタイプなどの、さまざまな要因が利益と利ざやにどのように影響を与えるかを分析したいと考えています。より進んだ分析を実行するために、そのおもちゃ店の DBA は、先に利益と利ざやの計測を作成する必要があります。そうすればその DBA は、さまざまな要因を、利益と利ざやの計測に対して相互に関連付けて比較することができます。

---

### 詳細なシナリオ

そのおもちゃ店のデータベースには、販売額、販売商品コスト (COGS)、および経費の各列と、それに対応した、いくつかのディメンション表別の外部キー列から成るファクト表があります。DBA は、販売額、COGS、および経費の列にそれぞれマップされる販売額、COGS、および経費の計測をすでに作成しています。利益および利ざやの計測は、それらの既存の計測だけから作成することができます。

DBA は、利益計測を作成するために、SQL 式で  $SALES-(COGS+EXPENSE)$  を計算して、算出されたデータをすべてのディメンションにわたって合計するための計測を作成します。利益計測を作成するには、既存の計測または列、またはこの 2 つの組み合わせを参照します。

DBA は、利益計測を作成し終わったら、利ざや計測を作成することができます。利ざや計測とは、 $(利益/販売額)*100$  というように、既存の 2 つの計測の比率をパーセントで表わしたものであり、独自の集約関数を必要としません。この計測が集約関数を必要としない理由は、すでに集約済みのデータをもつ他の計測しか参照しないからです。DBA が、他の計測を参照するだけの計測である複合計測を使用して比率を計算する場合、さらに別の集約を定義する必要はありません。SUM などの集約関数の大半にとって、比率は無意味です。たとえばおもちゃ店が、連続した 4 つの四半期中に、40%、32%、28%、および 37% の利ざやを得た場合にその期間中の比率を合計すると、年間の利ざやは 137% になりますが、これではつじつまが合いません。

---

### 計測の作成ステップ

以下のステップは、OLAP センターの「ファクト・プロパティ」ウィンドウを使って、既存のファクト・オブジェクト内に利益および利ざやの計測を作成する方法を説明しています。

1. 「ファクト・プロパティ」を開くには、OLAP センター・オブジェクト・ツリー内のファクト・オブジェクトを右マウス・ボタンでクリックしてから、「計測の編集」をクリックします。「ファクト・プロパティ」ウィンドウが開きます。

2. 次のようにして、利益計測を作成します。
  - a. 「計測」ページで「計算済み計測の作成」プッシュボタンをクリックします。「SQL 式ビルダー」ウィンドウが開きます。
  - b. 「SQL 式ビルダー」ウィンドウで、「名前」フィールドに PROFIT と入力します。
  - c. 利益式を作成するには、「データ」リストの「計測」フォルダーを拡張表示して、以下のステップを行います。
    - 「データ」リストの「SALES (販売額)」計測をダブルクリックして、それを式に追加します。
    - 「演算子」リストの - 演算子をダブルクリックします。
    - 「データ」リストの COGS 計測をダブルクリックします。
    - 「演算子」リストの + 演算子をダブルクリックします。
    - 「データ」リストの「EXPENSE (経費)」計測をダブルクリックします。
    - 「式」フィールドで、@Measure(MDSAMPLE.COGS)+@Measure(MDSAMPLE.EXPENSE) と表記された式部分を強調表示にしてから、「演算子」リストの (..) 演算子をダブルクリックし、式で選択した部分を括弧で囲みます。

15 ページの図 7 は、「SQL 式ビルダー」ウィンドウで作成できる利益式を示しています。

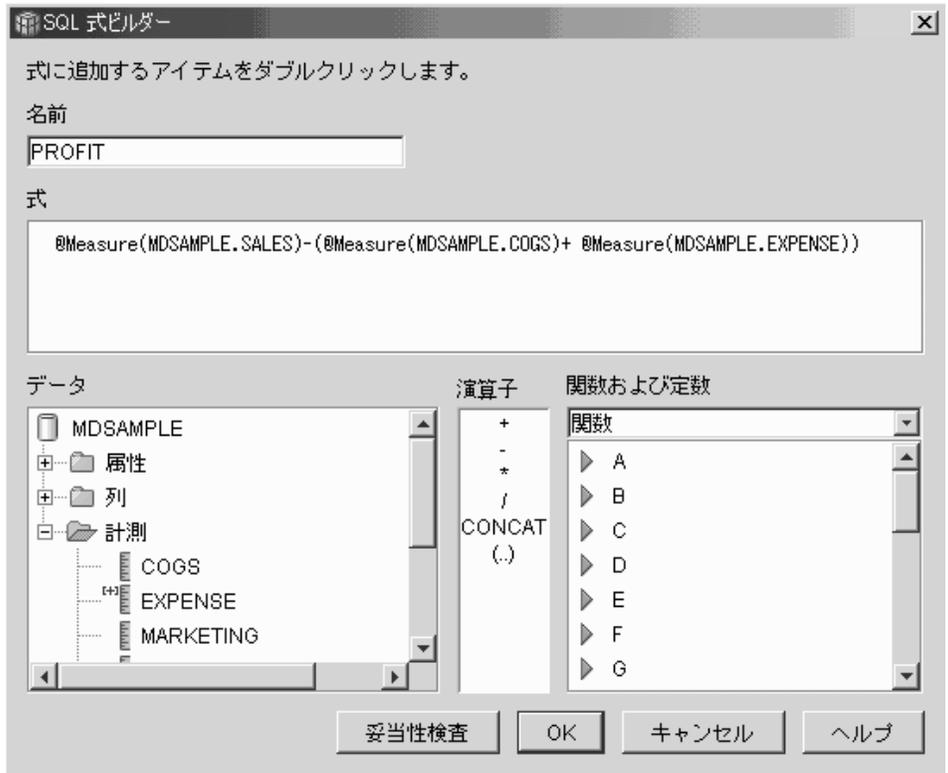


図7. 「SQL 式ビルダー」ウィンドウの完全な利益式

- d. 「OK」をクリックして利益計測を作成し、「SQL 式ビルダー」ウィンドウを閉じます。
  - e. 「集約」ページで、利益計測の集約をクリックしてから、SUM 関数を選択します。これで利益計測は完成しました。
3. 次のようにして、利ざや計測を作成します。
    - a. 「計測」ページで「計算済み計測の作成」をクリックします。「SQL 式ビルダー」ウィンドウが開きます。
    - b. 「SQL 式ビルダー」ウィンドウで、「名前」フィールドに PROFIT MARGIN と入力します。
    - c. 利ざや式を作成するには、「データ」リストの「計測」フォルダーを拡張表示して、以下のステップを行います。
      - 「データ」リストの「PROFIT (利益)」計測をダブルクリックして、それを式に追加します。
      - 「演算子」リストの / 演算子をダブルクリックします。
      - 「データ」リストの「SALES (販売額)」計測をダブルクリックします。

- 「式」フィールドへの入力によって、式全体を括弧で囲みます。
- 式の末尾にカーソルを置いて、「演算子」リストの \* 演算子をダブルクリックします。
- 「式」フィールドで、式の末尾に 100 と入力します。

図 8 は、「SQL 式ビルダー」ウィンドウで作成できる利ざや式を示しています。

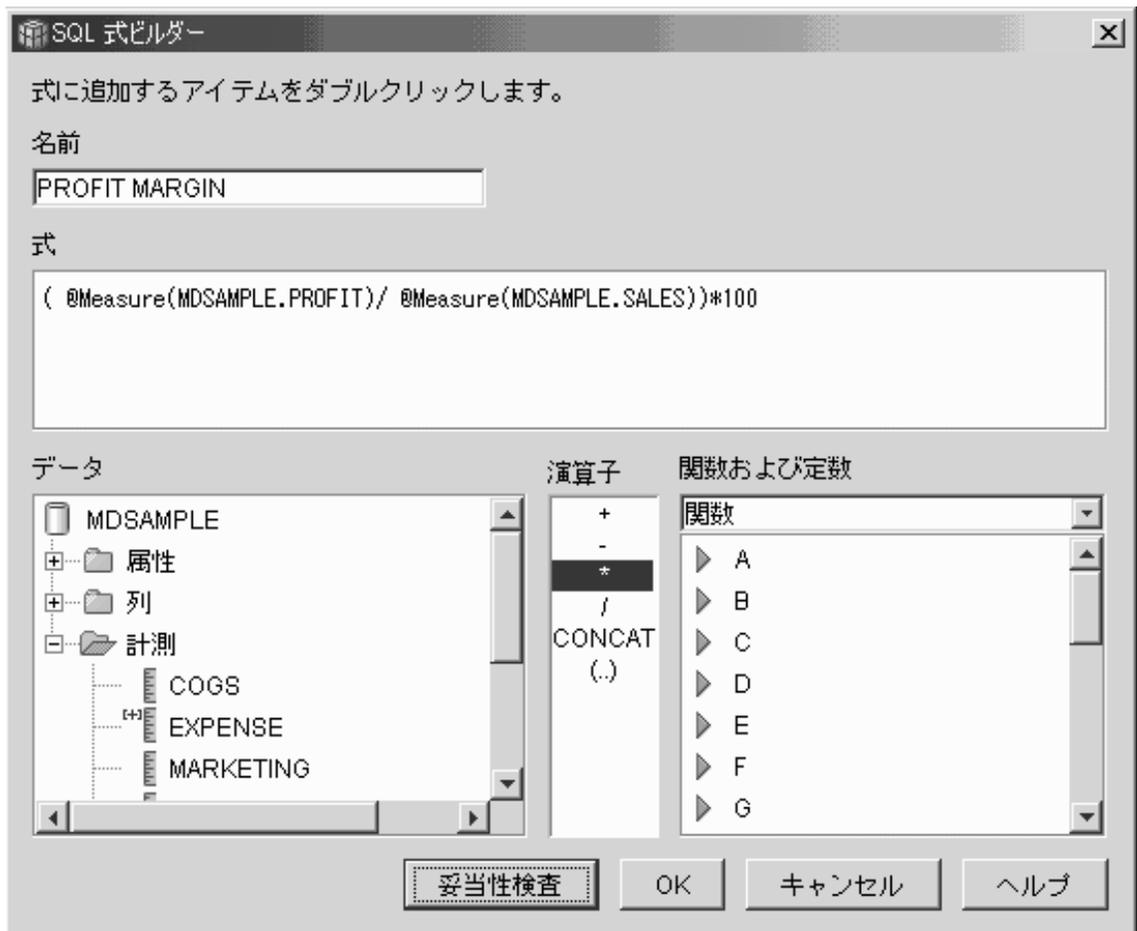


図 8. 「SQL 式ビルダー」ウィンドウの完全な利ざや式

- d. 「OK」をクリックして利ざや計測を作成し、「SQL 式ビルダー」ウィンドウを閉じます。

「集約」ページでは、OLAP センターによって複合計測に対してデフォルトで集約関数が NONE に設定されるので、集約関数を変更する必要はありません。

4. 「OK」をクリックして、「ファクト・プロパティ」ウィンドウを閉じ、ファクト・オブジェクトに追加した 2 つの新しい計測を保管します。

DBA は、この 2 つの計測を作成し終わったら、これらの重要な計測に関わるさらに別の分析を実行することができます。



---

## 第 4 章 インターネット・オーダー数のカウント

ある小売り企業が、数年前にインターネット販売を開始して、業務拡張を図りました。現在、同社は、インターネット販売の効果を分析したいと考えています。同社が最初に行うべき計算の 1 つは、インターネットを通して完了したオーダーの数です。

---

### 詳細なシナリオ

この会社のデータベースには、ORDER\_ID、PRODUCT\_ID、QUANTITY、および TIME\_ID の各列から成るインターネット用のファクト表があります。PRODUCT\_ID 列には、対応するオーダーで販売された各商品が入り、QUANTITY 列には、そのオーダーで購入された商品の数量が入ります。複数の商品のオーダーの場合、そのオーダーで販売された各商品の数と同じ数の行エントリーが設けられます。たとえば、表 6 は 3 件のオーダーを示していますが、この場合のオーダー 1 は、商品 A が 3 つ、商品 O が 1 つ、および商品 G が 1 つで構成されます。

表 6. ファクト表の内容の一部

ORDER_ID	PRODUCT_ID	QUANTITY
1	A	3
1	O	1
1	G	1
2	L	1
2	Q	2
3	P	5

DBA は、ORDER\_ID 列内の個々のエントリーをカウントするオーダー・カウント計測を作成することができます。オーダー・カウント計測を定義するには、SQL 式内で DISTINCT キーワードを使用し、すべてのディメンションにわたる集約用の COUNT 関数を使用します。この計測の SQL 式は、集約時にカウントされる個々のオーダーのリストを作成します。この計測には合計は関与しないので、非加法計測と呼ばれます。

非加法計測はまた、カウントしたい文字データやその他のデータに対しても役立ちます。たとえば、非加法計測を使って、商品の出荷先の郵便番号の数をカウントすることができます。

この例では、DBA は、ORDER\_ID 列に直接マップされるオーダー ID 計測を定義することを決意します。ただし、選択によっては、同じ方法で ORDER\_ID 列を使用することもできます。デフォルトの集約は、列または計測が SQL 式内で使用されるかどうか

かに基づいて異なりますが、どちらが使用されても、『計測の作成ステップ』に説明されているとおり、デフォルトの集約を COUNT 関数に変更する必要があります。

---

## 計測の作成ステップ

以下のステップは、OLAP センターの「ファクト・プロパティ」ウィンドウを使って、既存のファクト・オブジェクト内にオーダー・カウント計測を作成する方法を説明しています。

1. 「ファクト・プロパティ」を開くには、OLAP センター・オブジェクト・ツリー内のファクト・オブジェクトを右マウス・ボタンでクリックしてから、「計測の編集」をクリックします。「ファクト・プロパティ」ウィンドウが開きます。
2. 「計測」ページで「計算済み計測の作成」プッシュボタンをクリックします。「SQL 式ビルダー」ウィンドウが開きます。
3. 「SQL 式ビルダー」ウィンドウで、「名前」フィールドに ORDER COUNT と入力します。
4. オーダー・カウント式を作成するには、「データ」リストの「計測」フォルダーを拡張表示して、以下のステップを行います。
  - 「関数および定数」フィールドで、「その他」を選択します。その他の関数および定数のリストで、**DISTINCT** キーワードをダブルクリックします。
  - 「データ」リストの「オーダー ID (ORDER ID)」計測をダブルクリックします。

21 ページの図 9 は、「SQL 式ビルダー」ウィンドウで作成できるオーダー・カウント式を示しています。



図9. 「SQL 式ビルダー」ウィンドウの完全なオーダー・カウント式

5. 「OK」をクリックして、「SQL 式ビルダー」ウィンドウを閉じます。
6. 「集約」ページで、オーダー・カウント計測の集約をクリックしてから、**COUNT** 関数を選択します。
7. 「OK」をクリックして、ファクト・オブジェクトに加えた変更を保管し、「ファクト・プロパティ」ウィンドウを閉じます。

これで、個々のオーダー ID 行値の数をカウントする計測が作成されました。この計測を他の計測とともに使用して、さらに深くデータを分析することができます。



---

## 第 5 章 販売額のランキング

あるオフィス用品販売チェーン・ストアが、ここ数年で急速に発展をとげました。同社の経営陣は、営業成績の最も不振な店舗をいくつか閉店して、コストの削減と利益率の増大をもくろんでいます。閉店を決断する際は、営業成績ヒストリーが重要な要因となります。分析担当者が分析を完了するには、販売額の数字をランク付けして、すべてのディメンションを対象としてドリルダウンする必要があります。

---

### 詳細なシナリオ

このオフィス用品販売店のデータベースには、他の列に加えて、販売額列から成るファクト表があります。またこのデータベースには、いくつかのディメンション表もあります。DBA は、RANK 関数を使用する販売額ランク計測を作成することができます。なおこの関数は、DB2 Universal Database (DB2 UDB) に備えられている OLAP 関数です。

DB2 Cube Views は、DB2 UDB に備えられている以下の OLAP 関数をサポートしません。

#### RANK

行を並べて、各行にランキングを割り当てます。このランクは、順序付けによって確定された直前の行の番号に 1 を加えたものと定義されます。複数の行において行値の重複があるために、それらの行の相対順序を判別できない場合、同じランク番号が割り当てられます。行値の重複がある場合、ランキングの結果の番号と番号の間にギャップが生じる可能性があります。24 ページの表 7 は、一連の行値サンプルの場合の RANK 関数のランキング結果の例を示しています。

RANK 関数の標準的な構文は次のとおりです。

```
RANK ( ) OVER (ORDER BY sort-key-expression expression-order)
```

ただし *sort-key-expression* は、ランク付けされる一連のデータ、また *expression-order* は、*sort-key-expression* の値を昇順または降順に並べるためのキーワード **ASC** または **DESC** です。DB2 Cube Views では、*sort-key-expression* は、列や属性ではなく、既存の計測でなければなりません。また、DB2 Cube Views は、この関数とともに DB2 UDB に用意された PARTITION BY 文節をサポートしません。RANK 関数の詳細については、DB2 UDB インフォメーション・センターを参照してください。

#### DENSERANK

行を並べて、各行にランキングを割り当てます。行のランクは、正確に直前の行数に 1 を加えたものと定義されます。したがって、ランキングの結果は順次ランクとなり、ランクの番号付けの間にギャップが生じることはありません。表 7 は、一連の行値サンプルの場合の DENSERANK 関数のランキング結果の例を示しています。

DENSERANK 関数の標準的な構文は次のとおりです。

DENSERANK ( ) OVER (ORDER BY *sort-key-expression expression-order*)

ただし *sort-key-expression* は、ランク付けされる一連のデータ、また *expression-order* は、*sort-key-expression* の値を昇順または降順に並べるためのキーワード **ASC** または **DESC** です。DB2 Cube Views では、*sort-key-expression* は、列や属性ではなく、既存の計測でなければなりません。また、DB2 Cube Views は、この関数とともに DB2 UDB に用意された PARTITION BY 文節をサポートしません。DENSERANK 関数の詳細については、DB2 UDB インフォメーション・センターを参照してください。

## ROWNUMBER

行の順次行番号を、最初の行を 1 とする順序付けを使って計算します。ORDER BY 文節を指定しないと、行番号は任意の順序で行に割り当てられます。

ROWNUMBER 関数の標準的な構文は次のとおりです。

ROWNUMBER ( ) OVER ([ORDER BY *sort-key-expression expression-order*])

ただし *sort-key-expression* は、ランク付けされる一連のデータ、また *expression-order* は、*sort-key-expression* の値を昇順または降順に並べるためのキーワード **ASC** または **DESC** です。DB2 Cube Views では、列や属性ではなく、既存の計測をこの関数のデータ・ソースとして使用しなければなりません。また、DB2 Cube Views は、この関数とともに DB2 UDB に用意された PARTITION BY 文節をサポートしません。ROWNUMBER 関数の詳細については、DB2 UDB インフォメーション・センターを参照してください。

これらの OLAP 関数は、SQL 式ビルダー関数および定数のリストには載っていません。

表 7. RANK 関数と DENSERANK 関数を使った一連のサンプル値のランキング結果

行値	順序付け	RANK 関数での ランキング結果	DENSERANK 関数 でのランキング結果
100	1	1	1
35	2	2	2
23	3	3	3
8	4	4	4

表 7. RANK 関数と DENSERANK 関数を使った一連のサンプル値のランキング結果 (続き)

行値	順序付け	RANK 関数での ランキング結果	DENSERANK 関数 でのランキング結果
8	4	4	5
6	5	6	6

## 計測の作成ステップ

以下のステップは、OLAP センターの「ファクト・プロパティ」ウィンドウを使って、既存のファクト・オブジェクト内に販売額ランク計測を作成する方法を説明しています。

1. 「ファクト・プロパティ」を開くには、OLAP センター・オブジェクト・ツリー内のファクト・オブジェクトを右マウス・ボタンでクリックしてから、「計測の編集」をクリックします。「ファクト・プロパティ」ウィンドウが開きます。
2. 「計測」ページで「計算済み計測の作成」をクリックして、販売額ランク計測を作成します。「SQL 式ビルダー」ウィンドウが開きます。
3. 「SQL 式ビルダー」ウィンドウで、「名前」フィールドに SALES RANK と入力します。
4. 販売額ランク式を作成するには、以下のステップを行います。
  - 以下の関数構文を「式」フィールドに入力します。RANK ( ) OVER (ORDER BY measure DESC)
  - 「データ」リストの「計測」フォルダーを拡張表示します。
  - 「式」フィールドの関数構文内の語 measure を強調表示してから、「SALES (販売額)」をダブルクリックして、販売額計測を式に追加します。

最終的な式は、26 ページの図 10 に示されています。

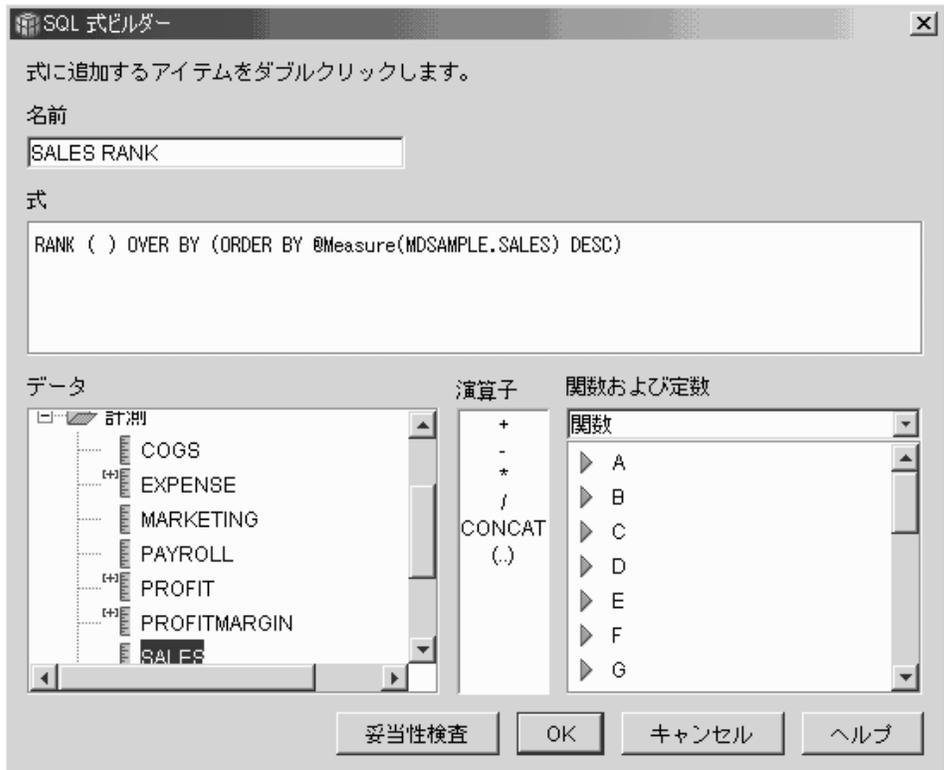


図 10. 「SQL 式ビルダー」ウィンドウの完全な販売額ランク式

5. 「妥当性検査」をクリックして、式が有効であることを確認します。「OK」をクリックして、「SQL 式ビルダー」ウィンドウを閉じます。

「集約」ページで、デフォルト集約 None を変更する必要はありません。None オプションは、販売額ランク計測のデフォルトです。それは、データ・ソースは数値であって、既存の計測のみを参照するからです。

分析担当者は、計測の式内で RANK 関数を使って販売額列を降順に並べることで、他のすべてのディメンション全体をドリルダウンし、特定の商品ラインに関して、またはデータベースに保管されている他のディメンションのデータをふまえて、ここ 1 年間で営業成績の最も不振な店舗を判別することができます。

---

## 第 6 章 ファクト表に保管された時間データを使用した時間ディメンションの作成

小売企業の XYZ Retail 社は、販売トランザクション・データを DB2 Cube Views においてモデル化して、社内データの分析をさらに効率化しようとしています。しかし、そのデータのトランザクションに関わる特性が原因で、利用できる時間情報は、各トランザクションに関連した日付だけです。時間情報（時間ディメンションでモデル化されます）が必要なのは、四半期ごとの販売傾向の分析や、週ごとの平均在庫値などの、共通した多数の計算にコンテキストを付加するためです。

たいていの DBA は、トランザクションの時間データを日付やタイム・スタンプで保管しないようにしています。なぜなら、1 日でもトランザクションのない日があると、データに穴があいてしまって、データの正確な集約と表示という点で問題を生じる可能性があるからです。通常は、時間表内で時間データをモデル化するほうが望ましいやり方です。しかし XYZ Retail の DBA は、少なくとも 1 つのトランザクションが毎日必ずあると確信して、現在のデータ構造を維持することを決定しました。

---

### 詳細なシナリオ

XYZ Retail には、販売額、コスト、販売数量、および日付などの、各トランザクションに関する計測可能なデータの入ったファクト表があります。それ以外に、データベースには、地域ディメンション表と商品ディメンション表が置かれています。問題は、時間データが、別個のディメンション表内に保管されているのではなく、ファクト表に入っていることにあります。DBA は、ファクト・オブジェクト内の日付データに基づいたディメンション・オブジェクトを作成する必要があります。

ファクト表内の単一の列の日付データに基づいて時間ディメンションを作成する場合、以下の 2 通りの要件が関わってきます。

- 有効なキューブ・モデル内のすべてのディメンション・オブジェクトは、ファクト・オブジェクトに結合される必要があります。しかも時間ディメンション・オブジェクトとファクト・オブジェクトは、同一のファクト表をベースにするので、ファクト表を自身に結合するための自己結合を使って、時間ディメンション・オブジェクトをファクト・オブジェクトに結合する必要があります。
- DBA は、週、月、四半期、および年などの分かりやすいレベルに分けて日付データを集約するための、計算済み属性を作成しなければなりません。

自己結合とは、表を自身に結合するタイプの結合のことであり、この例では、表はファクト表になります。自己結合では、ファクト表内の任意の行を固有識別できる 1 つ以上の列をまとめて結合しなければなりません。主キーを使うのが、最も望ましいやり方

です。なお、主キーが定義されていない場合の主キーの有望な候補は、ファクト表をディメンション表に結合するのに使用する一連の列になります。キューブ・モデルの最適化のためには、主キーを定義する必要があります。結合カーディナリティーは 1:1 でなければならず、結合タイプは内部結合でなければなりません。

図 11 は、ファクト・オブジェクト、ファクト表に基づいたディメンション、およびファクトからディメンションへの結合を、どのように 1 つのファクト表にマップできるかを示しています。

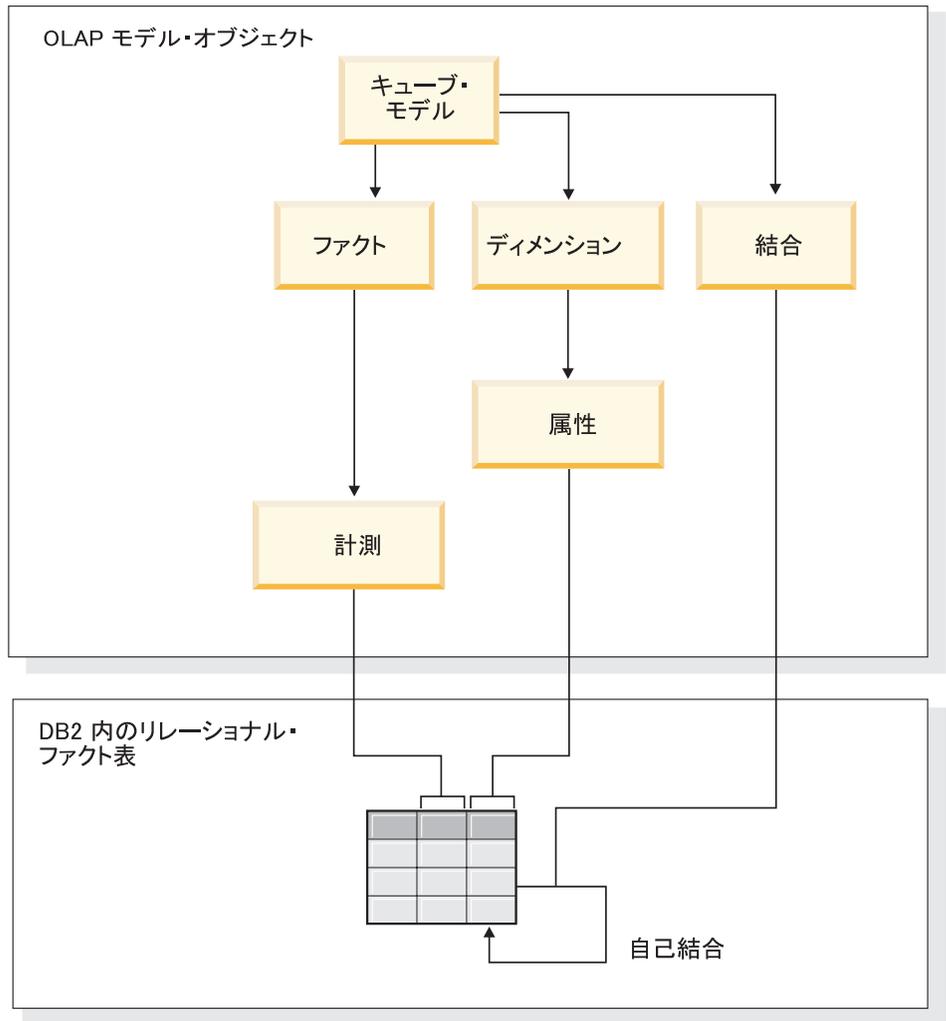


図 11. 自己結合を使って表を自身に結合する方法

## 属性とディメンションの作成ステップ

以下のステップは、OLAP センターの「ディメンション」ウィザードを使って、ファクト表に基づいた時間ディメンションと計算済み属性を作成する方法を説明しています。

1. 「ディメンション」ウィザードを開くには、OLAP センター・オブジェクト・ツリー内のキューブ・モデルを右マウス・ボタンでクリックしてから、「**ディメンションの作成**」をクリックします。「ディメンション」ウィザードが開きます。
2. 「名前」ページで、「名前」フィールドに Time と入力します。選択によっては、業務名を変更したり、コメントを入力したりすることもできます。「次へ」をクリックします。
3. キューブ・モデルのファクト表を選択します。「次へ」をクリックします。ディメンション内には表は 1 つしかないので、ディメンション結合を指定する必要はありません。「ディメンション結合」ページで、「次へ」をクリックします。
4. 「ディメンション属性」ページで、「**タイム・スタンプ (Timestamp)**」列を選択します。
5. オプション: 月、四半期、および年などのもっと大きい単位にタイム・スタンプ・データを集約する追加の計算済み属性を作成します。計算済み属性を作成するには、「**計算済み属性の作成**」プッシュボタンをクリックして「SQL 式ビルダー」を開いてから、ソース・タイム・スタンプ列を月、四半期、および年に分けて計算する式を各属性ごとに定義します。各計算済み属性を定義し終わったら、「**妥当性検査**」プッシュボタンをクリックして、式が有効であることを確認してから、「**OK**」をクリックして「SQL 式ビルダー」を閉じ、「ディメンション」ウィザードに戻ります。必要な属性をすべて選択および作成した後で、「次へ」をクリックします。
6. 「ディメンション・タイプ」ページで、「**Time (時間)**」を選択します。「次へ」をクリックします。
7. 「ファクト・ディメンション結合」ページで、「**結合の作成**」をクリックします。開かれた「結合」ウィザードで、自己結合を作成します。名前を入力して、「次へ」をクリックします。左右両方の属性ごとに、主キーなどの、ファクト表内の任意の行を固有定義する列または一連の列を選択します。一度にペアを 1 つずつ選択して、「**追加**」をクリックし、その属性ペアを結合に追加します。内部結合タイプと 1:1 のカーディナリティーを選択します。必要な属性ペアを追加し終わったら、「完了」をクリックします。「結合」ウィザードが閉じます。
8. 「ファクト・ディメンション結合」ページで、「完了」をクリックします。

XYZ Retail 社は、キューブ・モデルに時間ディメンションを定義したので、社内のデータ分析はさらに意義のあるものになります。これで同社は、在庫を含めて、時間対応の分析を実行できるようになりました。



---

## 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものであり、米国以外の国においては本書で述べる製品、サービス、またはプログラムを提供しない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒106-0032  
東京都港区六本木 3-2-31  
IBM World Trade Asia Corporation  
Licensing

**以下の保証は、国または地域の法律に沿わない場合は、適用されません。** IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation  
J46A/G4  
555 Bailey Avenue  
San Jose, CA 95141-1003  
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

---

## 商標

以下は、IBM Corporation の商標です。

DB2  
DB2 Universal Database  
IBM  
Office Connect  
Redbooks

以下は、それぞれ各社の商標または登録商標です。

Microsoft、Windows および Windows NT は、Microsoft Corporation の米国およびその他の国における商標です。

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

UNIX は、The Open Group の米国およびその他の国における登録商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。