Oracle Database 10*g* の SQL MODEL 句

オラクル・ホワイト・ペーパー

2003 年 8 月

ORACLE

Oracle Database 10*g* の SQL MODEL 句

Oracle Corporation 発行「The SQL Model Clause of Oracle Database 10*g*」の翻訳版です。

Oracle Database 10*g* の SQL MODEL 句

概要

SQL では、特に行間参照が関係する場合、計算が複雑になることがあります。多くの場合、その計算には SQL の複雑な JOIN 文や UNION 文を使用した、作成、メンテナンスおよび実行を効率よく行うことが困難なコードを必要とします。Oracle Database 10*g* では、複雑な SQL 計算に対して革新的なアプローチが導入されています。それが SQL MODEL 句です。

SQL の SELECT 文への拡張機能である MODEL 句を使用して、開発者はリレーショナル・データを多次元配列として扱い、それらの配列に対して式を定義できます。MODEL 句は、簡潔な構文で難しい計算の処理に必要な機能を提供します。MODEL 句は、自動的に式の依存関係を解決し、高度なアプリケーションにおける相互にリンクされた式の大規模なセットをサポートします。さらに、MODEL 句処理には、高度な最適化の方法およびデータ構造の採用により、ハイ・パフォーマンスを実現しています。MODEL 句の表現力とその使いやすさ、および Oracle データベースのスケーラビリティと管理性は、データベース・アプリケーションにとって大きな利点となります。

はじめに

ビジネスおよび専門分野の多くのデータベース・アプリケーションは、SQL には組み込みにくい計算を必要とします。製品階層または地域階層の様々なレベルにかかわる市場専有率計算やカスタマイズされた集計には、複雑な自己結合および UNION 演算が必要となる場合があります。時系列分析および連立方程式などの反復計算は、データを、データベースから外部計算エンジンに取り出すことが必要な場合もあります。論理的には、相互に依存する多くの式を伴う予算や売上予測などの財政モデルは、特に扱いにくいものです。

SQL の限界は、アプリケーション開発上およびメンテナンス上大きな負担となりますが、それ以上に大きな問題があります。データをデータベースから抽出して外部処理を行う場合、管理ワークロードは増大し、管理性は低下します。抽出したデータに基づいて、PC のスプレッドシートで実行される財政予測を検討してみます。多くのエンド・ユーザーが標準の予測シートのコピーを各自で実行する場合、計算方法の変更ごとに、すべてのマシンで更新処理を行う必要があります。1 つの更新に失敗しただけでも、財務計算にエラーが発生しますが、これは、今日のビジネス環境における主要なリスクとなっています。結果が一貫した式に基づいている場合でも、多くの PC へのタイムリーなネットワーク・アクセスを要する出力の整理という複雑なタスクが伴います。多くの PC への信頼性の高いアクセスは、ネットワークに接続されていないラップトップ・コンピュータへの転換

が進む中で、従来以上に困難な課題になっています。複雑な計算がデータベース内で一元化されると、ビジネス上の大きな効果があります。

Oracle Database 10*g* は、SQL の SELECT 文への強力な拡張機能である、SQL の MODEL 句を使用してこれらのニーズに対応します。MODEL 句は、あらゆるタイプのアプリケーションで使用される計算を簡素化し、一元化します。SQL の MODEL 句を使用すると、問合せの結果を多次元配列の形で参照でき、式を適用して新しい配列の値を計算できます。式は、行間、配列間の参照を伴う、高度な相互依存的計算の式の場合もあります。MODEL 句を使用して構築されたアプリケーションは、スプレッドシートやその他の外部計算エンジンのかわりに使用でき、より堅牢で効率的なソリューションを提供します。高度な計算をデータベースに統合することによって、パフォーマンス、スケーラビリティおよび管理性が、外部計算と比べて大幅に向上します。

- パフォーマンス – MODEL 句処理により、SQL の多くの JOIN 演算および UNION 演算が不要になります。MODEL 句により、高度なアルゴリズムおよびデータ構造の使用が可能になるため、パフォーマンスが最大化されます。最も基本的なレベルでは、MODEL 句を使用することにより、外部処理に必要な、データのラウンド・トリップ、すなわち別のアプリケーションへのデータのコピー、そのデータの処理および結果のデータベースへのロードが不要になります。
- スケーラビリティ – Oracle データベースのパラレル・クエリー機能は、PC のスプレッドシートなどの外部計算ツールでは実現できない企業レベルのスケーラビリティをサポートします。MODEL 句は、利用できるかぎりのシステム・リソースを効率的に使用して、Oracle データベースのパラレル処理機能を活用します。
- 管理性 – 計算がデータに近い場所で一元化されると、分散化した場所で計算が行われることによる非一貫性とセキュリティの問題が解消されます。また、様々な計算エンジンと互換性のないデータ構造が混在しているのではなく、アプリケーション間で共通のリレーショナルな環境を共有する場合、データの統合が簡素化されます。

概念

MODEL 句は、問合せの列を、パーティション列、ディメンション列、メジャー列という 3 つのグループに対応付けることによって多次元配列を定義します。これらの要素が実行するタスクは次のとおりです。

- パーティションは、分析関数のパーティションと同様の方法（オラクル社『データ・ウェハウス・ガイド』参照）で結果セットのブロックを定義します。それぞれのパーティションは、式の側からは、独立した配列とみなされます。
- ディメンションは、パーティション内の各メジャー・セルを識別します。これらの列は、日付、地域および製品名などの識別特性です。
- メジャーは、スター・スキーマのファクト表のメジャーと類似しています。メジャーには、通常、販売単位やコストなどの数値が格納されています。

各セルは、ディメンションの組合せを省略なしに指定することにより、それぞれのパーティション内でアクセスされます。

これらの多次元配列に対する式の作成には、ディメンション値として表現された計算ルールを定義します。ルールは柔軟かつ簡潔なもので、表現力を高めるために、ワイルド・カードおよび FOR ループを使用できます。MODEL 句に基づく計算は、分析関数のデータベースへの統合、シンボリック参照による可読性の改善、スケーラビリティおよび非常に高い管理性を実現します。

次の図に、架空の売上表を使用したモデル概念の概要を示します。表には、国、製品、年度および売上数量の列があります。この図は 3 つの部分に分かれています。一番上のセグメントは、表をパーティション、ディメンション、メジャーの各列に分割するという概念を示しています。中央のセグメントは、2002 年度の Prod1 と Prod2 の値を計算する 2 つのルールを示しています。最後の 3 つ目のセグメントは、前述のルールを架空のデータが格納されている表に適用した際の問合せ結果を示しています。斜線が施されていない出力はデータベースから取得したデータで、斜線が施された出力は、ルールを適用して計算された行を示しています。なお、ルールは、それぞれのパーティション内で適用されます。

モデル・エンティティへの列の対応付け

国	製品	年度	Sales
パーティション	ディメンション	ディメンション	メジャー

↓ ルール

```
Sales (Prod1, 2002) = Sales (Prod1, 2000) + Sales (Prod1, 2001)
Sales (Prod2, 2002) = Sales (Prod2, 2000) + Sales (Prod2, 2001)
```

↓ MODEL 句の出力

国	製品	年度	Sales	
パーティション	ディメンション	ディメンション	メジャー	
A	Prod1	2000	10	元のデータ
A	Prod1	2001	15	
A	Prod2	2000	12	
A	Prod2	2001	16	
B	Prod1	2000	21	
B	Prod1	2001	23	
B	Prod2	2000	28	
B	Prod2	2001	29	
A	Prod1	2002	25	ルール適用結果
A	Prod2	2002	28	
B	Prod1	2002	44	
B	Prod2	2002	57	

表の既存データが、MODEL 句によって更新されることはありません。また、新しいデータが表に挿入されることもありません。表内のデータを変更するには、このモデルの結果を INSERT 文、UPDATE 文または MERGE 文に提供します。

技術的詳細

前述の項では、SQL の MODEL 句の概要を説明しました。ただし、その特長を明確に理解するためには、詳細な例が必要です。この文書の後半の項では、例を使用して、モデルの基本的概念およびキーワードについて、段階的に説明します。1 つの例ですべての特長が示されているわけではありませんが、少なくとも、すべての主要要素の簡単な説明は提供します。理解を助けるため、すべての例は Oracle Database 10*g* と同梱のサンプル・データを使用して実行できます。ここでは、サンプル・スキーマ SH（ビジネス・インテリジェンス・アプリケーションで広く使用されている、スター・スキーマ）を使用しています。SH には、海外での販売実績のある、消費者電子製品ベンダーの売上履歴が格納されています。

基本構文

この文書で説明する MODEL 句構文の基本要素は次のとおりです。構文アイテムはこの他にもありますが、ここで取り上げたセットが中核的な機能を代表しています。MODEL 句は、最後の ORDER BY 以外の、SELECT 文の他の句がすべて処理されてから処理されます。

```
<prior clauses of SELECT statement>
MODEL [main]
  [reference models]
  [PARTITION BY (<cols>)]
   DIMENSION BY (<cols>)
   MEASURES (<cols>)
    [IGNORE NAV] | [KEEP NAV]
  [RULES
    [UPSERT | UPDATE]
    [AUTOMATIC ORDER | SEQUENTIAL ORDER]
    [ITERATE (n) [UNTIL <condition>] ]
    ( <cell_assignment> = <expression> ... )
```

サンプル・データ

サンプルを簡潔にするために、Oracle 10*g* で提供されているサンプル・スキーマ・セットの SH（Sales History）を使用してビューを作成します。*sales_view* ビューは、国別の製品売上高の年間合計を、すべてのチャネルの集計結果として、ドルと数量で提示します。このビューは、100 万行のファクト表から構築されたもので、次のように定義されています。

```
CREATE VIEW sales_view AS
SELECT country_name country, prod_name prod,
  calendar_year year,
  SUM(amount_sold) sale, COUNT(amount_sold) cnt
FROM sales, times, customers, countries, products
WHERE sales.time_id = times.time_id AND
  sales.prod_id = products.prod_id AND
  sales.cust_id = customers.cust_id AND
  customers.country_id = countries.country_id
GROUP BY country_name, prod_name, calendar_year;
```

このサンプルを Oracle システムで最短の実行時間で実行するためには、この文のマテリアライズド・ビューを作成します。サンプル・コードは、Oracle データベースのサマリー管理機能によって、マテリアライズド・ビューの利点を活用できるよう自動的に書きなおされます。

MODEL 句の最初の例

モデルの最初の例として、次の文を検討します。この文では、2 つの製品の売上高値を計算して、この 2 つの製品のデータに基づいて新規製品の売上高を定義します。

```
SELECT SUBSTR(country,1,20) country, SUBSTR(prod,1,15) prod,
year, sales
FROM sales_view
WHERE country IN ('Italy','Japan')
MODEL RETURN UPDATED ROWS
  PARTITION BY (country)
  DIMENSION BY (prod, year)
  MEASURES (sale sales)
  RULES (
    sales['Bounce', 2002] = sales['Bounce', 2001] +
      sales['Bounce', 2000],
    sales['Y Box', 2002] = sales['Y Box', 2001],
    sales['2_Products', 2002] = sales['Bounce', 2002] +
sales['Y Box', 2002])
ORDER BY country, prod, year;
```

結果は次のとおりです。

```
COUNTRY              PROD            YEAR       SALES
-------------------- --------------- ---------- ----------
Italy                2_Products      2002       92613.16
Italy                Bounce          2002       9299.08
Italy                Y Box           2002       83314.08
Japan                2_Products      2002       103816.6
Japan                Bounce          2002       11631.13
Japan                Y Box           2002       92185.47
```

この文では、データを国別に分割しているため、式は国ごとに適用されます。この売上高のファクト・データは 2001 年度で終了するため、2002 年度の値を定義するルールでは新しいセルが挿入されます。最初のルールは、Bounce というビデオ・ゲームの 2002 年度の売上高を、2000 年度と 2001 年度の売上高の合計として定義します。第 2 のルールは、2002 年度の Y Box の売上高を、2001 年度と同じ値として定義します。第 3 のルールは、2_Products という製品を定義します。これは単純に、Bounce と Y Box との 2002 年度の値の合計です。2_Products の値は先行する 2 つの式から派生したもので、Bounce と Y Box のルールは、2_Products のルール前に実行する必要があります。

前述の例には、次のような特性があります。

- MODEL キーワードに続く RETURN UPDATED ROW 句は、この問合せにおいてのみ作成または更新された行に対して返される結果を制限します。新規に計算された値のみに結果セットを制限する場合にこの句を使用すると有効です。ここでの例では、一貫して RETURN UPDATED ROWS 句を使用します。

- ここの例で、式の開始時に必ず使用されている RULES キーワードはオプションですが、読みやすさを考慮して使用します。
- 同様に、例の多くは、「国」列では ORDER BY は不要ですが、例に手を加えて複数の国を使用するユーザーの便宜を考慮してこの指定も含めています。

セルと値の参照

この項では、SQL モデルでセルおよび値を参照する方法を検討します。セル参照上のマテリアルは、SQL の MODEL 句の大きな機能を理解する上で必須です。

位置ベースのセル参照: シングル・セル・アクセスとアップサート

2000 年の製品 Bounce のイタリアにおける既存の売上高の値を更新して、10 に設定する場合を考えます。その場合、次のような問合せで、既存のセルをその値に更新できます。

```
SELECT SUBSTR(country,1,20) country, SUBSTR(prod,1,15) prod,
  year, sales
FROM sales_view
WHERE country='Italy'
MODEL RETURN UPDATED ROWS
  PARTITION BY (country)
  DIMENSION BY (prod, year)
  MEASURES (sale sales)
  RULES (
    sales['Bounce', 2000] = 10 )
ORDER BY country, prod, year;

COUNTRY              PROD            YEAR       SALES
-------------------- --------------- ---------- ----------
Italy                Bounce          2000       10
```

前述の問合せの式では、位置ベースのセル参照を使用しています。セル参照の値は、式の中での位置に基づいて適切なディメンションに適合されます。モデルの DIMENSION BY 句によって、各ディメンションに割り当てられる位置が決まります。この場合、最初の位置は製品（prod）で、第 2 の位置は年度です。

2005 年度の製品 Bounce のイタリアにおける売上高の予測値を作成して、20 に設定する場合を考えます。次の問合せを使用します。

```
SELECT SUBSTR(country,1,20) country, SUBSTR(prod,1,15) prod,
  year, sales
FROM sales_view
WHERE country='Italy'
MODEL RETURN UPDATED ROWS
    PARTITION BY (country)
    DIMENSION BY (prod, year)
    MEASURES (sale sales)
    RULES (
      sales['Bounce', 2005] = 20 )
ORDER BY country, prod, year;

COUNTRY              PROD            YEAR       SALES
-------------------- --------------- ---------- ----------
Italy                Bounce          2005       20
```

前述の問合せでは、式により年度の値が 2005 に設定されるため、配列に新しいセルが作成されます。

注意: 将来の売上高予測などの新しいセルを作成する場合、位置ベースの参照または FOR ループ（後述）を使用する必要があります。すなわち、位置ベースの参照では更新と配列への挿入の両方が可能です。これはアップサート・プロセスと呼ばれます。

シンボリック・セル参照: マルチセル・アクセスと更新

1999 年度以降の全年度の製品 Bounce の売上高を、既存の値から更新する場合を考えます。この場合も、イタリアの値を変更して 10 に設定します。この操作には、次の問合せを使用します。

```
SELECT SUBSTR(country,1,20) country, SUBSTR(prod,1,15) prod,
  year, sales
FROM sales_view
WHERE country='Italy'
MODEL RETURN UPDATED ROWS
  PARTITION BY (country)
  DIMENSION BY (prod, year)
  MEASURES (sale sales)
  RULES (
    sales[prod='Bounce', year>1999] = 10 )
ORDER BY country, prod, year;

COUNTRY              PROD            YEAR       SALES
-------------------- --------------- ---------- ----------
Italy                Bounce          2000       10
Italy                Bounce          2001       10
```

前述の問合せの式では、シンボリック・セル参照を使用しています。シンボリック・セル参照を使用する場合は、標準の SQL 条件を使用して、式の一部を構成するセルを決定します。<,>、IN および BETWEEN などの条件を使用できます。この例では、式は製品値が Bounce で、年度値が 1999 より大きいすべてのセルに適用されます。例では、1 つの式が複数のセルにアクセスすることを示しています。

注意: シンボリック参照は非常に効果的ですが、既存のセル更新のみに使用されます。将来の売上高予測などの新規セルの作成はできません。セル参照が、そのディメンションのいずれかでシンボリックな表記法を使用する場合は、その式は更新のみを実行します。後半で、1 つの式から複数のセル作成に簡潔な方法である、MODEL 句における FOR ループについて説明します。

単一の問合せでの位置ベースの参照およびシンボリック・セル参照

単一の問合せを使用して、複数の国における数年間のいくつかの製品の売上高を更新する場合を考えます。また、新規セルも挿入するものとします。いくつかの式を 1 つの問合せに配置すると、データのアクセス回数が減少するため、式を 1 つ持つ複数の問合せより効率的です。また、その方法を使用した場合、SQL がより簡潔な表現となり、開発者の生産性の向上にもつながります。次に、2 つの既存製品の値を更新し、新規の製品を挿入する例を示します。表記法の説明に、この問合せでは 3 行のインライン・コメントが使用されています。

```
SELECT SUBSTR(country,1,20) country, SUBSTR(prod,1,15) prod,
  year, sales
FROM sales_view WHERE country IN ('Italy','Japan')
MODEL RETURN UPDATED ROWS
  PARTITION BY (country)
  DIMENSION BY (prod, year)
  MEASURES (sale sales)
  RULES (
    sales['Bounce', 2002] = sales['Bounce', year = 2001] ,
      --positional notation: can insert new cell
    sales['Y Box', year>2000] = sales['Y Box', 1999],
      --symbolic notation: can update existing cell
    sales['2_Products', 2005] = sales['Bounce', 2001] +
      sales['Y Box', 2000] )
      --positional notation: permits creation of new cell
      --for new product
  ORDER BY country, prod, year;


COUNTRY              PROD            YEAR       SALES
-------------------- --------------- ---------- ----------
Italy                2_Products      2005       34579.63
Italy                Bounce          2002       4928.65
Italy                Y Box           2001       15177.7
Japan                2_Products      2005       52563.55
Japan                Bounce          2002       6443.77
Japan                Y Box           2001       22297.76
```

この例のデータには、2001 年度までの値しかないため、2002 年度以降についてのルールには、新規セルの挿入が必要です。同様のことが、ここで定義する新規製品名にも当てはまります。3 つ目の計算では、2005 年度について新規製品 2_Products を定義しているため、その製品用にセルが 1 つ挿入されます。最初の、2002 年度の Bounce についてのルールは位置表記法なので、新しいセルが挿入されます。第 2 の、Y Box についてのルールではシンボリックな表記法が使用されていますが、2001 年度については Y Box の値が既存するため、それらの値が更新されます。第 3 の、2005 年度の 2_Products についてのルールは位置ベースです。したがって、新規のセルが挿入され、出力にも表示されます。

式の右辺でのマルチセル参照

これまでの例では、マルチセル参照は式の左辺のみで使用されていました。式の右辺にある複数のセルの参照を考えます。マルチセル参照は式の右辺でも使用できますが、その場合は、集計関数を適用してそれらを単一の値に変換する必要があります。OLAP 集計（逆分散関数など）や統計集計も含めて、すべての既存の集計関数およびユーザー定義の集計関数を使用できます。

イタリアにおける 2005 年度の Bounce の売上高予測を、1999 年度から 2001 年度までの売上高の最高値を 100 上回る値に設定する場合は、次の問合せを行います。

```
SELECT SUBSTR(country,1,20) country, SUBSTR(prod,1,15) prod,
  year, sales
FROM sales_view
WHERE country='Italy'
MODEL RETURN UPDATED ROWS
  PARTITION BY (country)
  DIMENSION BY (prod, year)
  MEASURES (sale sales)
  RULES (
    sales['Bounce', 2005] = 100 + MAX(sales)['Bounce',
      year BETWEEN 1998 AND 2002] )
ORDER BY country, prod, year;

COUNTRY              PROD            YEAR       SALES
-------------------- --------------- ---------- ----------
Italy                Bounce          2005       5028.65
```

前述の問合せでは、BETWEEN 条件を使用して、式の右辺で複数のセルを指定しています。MAX()関数によってこれらのセルが 1 つの値に集計されます。

注意: 集計関数は、式の右辺でのみ使用できます。集計関数で使用される引数は、定数、バインド変数、MODEL 句のメジャーまたはそれらが使用される式です。

CV()関数: 右辺の計算での左辺の値の使用

CV()関数は非常に効果的なツールで、式の右辺で使用されて、複数のセルを参照する左辺の指定をコピーします。この方法により、非常に簡潔で柔軟なマルチセル式が可能となります。CV()関数は、SQL の結合演算に相当しますが、それ以上に簡潔で可読性の高い関数です。

イタリアにおける複数の年度にわたる Bounce の売上高を、式を使用して更新する場合を考えます。ただし、各年度の売上高はその年の Mouse Pad の売上高の合計に、同年の Y Box の売上高の 20%を加算したものとします。そのためには、次の問合せを使用します。

```
SELECT SUBSTR(country,1,20) country, SUBSTR(prod,1,15) prod,
  year, sales
FROM sales_view
WHERE country='Italy'
MODEL RETURN UPDATED ROWS
  PARTITION BY (country)
  DIMENSION BY (prod, year)
  MEASURES (sale sales)
  RULES (
    sales['Bounce', year BETWEEN 1995 AND 2002] =
      sales['Mouse Pad', CV(year)] +
      0.2 * sales['Y Box', CV(year)]
  )
ORDER BY country, prod, year;

COUNTRY              PROD            YEAR       SALES
-------------------- --------------- ---------- ----------
Italy                Bounce          1999       7681.51
Italy                Bounce          2000       9586.286
Italy                Bounce          2001       21587.916
```

式で使用されている 2 つの CV（）関数は、現在参照されている、左辺のセルの年度ディメンションの値を返します。前述の式の左辺が Bounce と 1999 のセルを参照する場合、右辺の式は次のように解決されます。

```
sales['Mouse Pad', 1999] + 0.2 * sales['Y Box', 1999].
```

同様に、左辺が Bounce と 2000 のセルを参照する場合、評価の対象となる右辺の式は次のようになります。

```
sales['Mouse Pad', 2000] + 0.2 * sales['Y Box', 2000].
```

CV（）関数は、引数としてディメンション・キーを受け取ります。また、CV（）は、位置ベースの参照をもたらす CV()のように、引数なしに使用することもできます。

したがって、前述の式は次のように記述できます。

```
s ['Bounce', year BETWEEN 1995 AND 2002] =
  s['Mouse Pad', CV()] + 0.2 * s['Y Box', CV()]
```

この条件は 1995 年度から 2002 年度の範囲の任意の値を受け入れることができるにもかかわらず、前述の式の結果としては、1999 年度から 2001 年度までの値のみが表示されます。これは、例の表の場合、1999 年度から 2001 年度のみのデータしか存在しないためです。式では、その柔軟性を示す例証として、広い時間範囲を使用しています。

行間計算の式での CV（）の使用方法

CV（）を使用すると、非常に柔軟性の高い式となります。たとえば、CV（年度）値からの減算によって、データ・セット内の他の行を参照できます。セル参照に CV（year）–2 という式がある場合、2 年前のデータにアクセスできます。

イタリアにおける Y Box、Bounce および Mouse Pad の売上高について、対前年比の伸び率（%）を計算する場合を考えます。このタスクの問合せは次のとおりです。

```
SELECT SUBSTR(country,1,10) country, SUBSTR(prod,1,10) prod,
year, sales, growth_pct
FROM sales_view
WHERE country='Italy'
MODEL RETURN UPDATED ROWS
  PARTITION BY (country)
  DIMENSION BY (prod, year)
  MEASURES (sale sales, 0 growth_pct)
  RULES (
    growth_pct[prod IN ('Bounce','Y Box','Mouse Pad'),
               year BETWEEN 1998 and 2001] =
        100* (sales[CV(prod), CV(year)] –
          sales[CV(prod), CV(year) -1] ) /
          sales[CV(prod), CV(year) -1] )
ORDER BY country, prod, year;
```

```
COUNTRY    PROD       YEAR       SALES      GROWTH_PCT
---------- ---------- ---------- ---------- ----------
Italy      Bounce     1999       2,474.78
Italy      Bounce     2000       4,333.69    75.11
Italy      Bounce     2001       4,846.30    11.83
Italy      Mouse Pad  1998       3,055.69
Italy      Mouse Pad  1999       4,663.24    52.61
```

```
Italy      Mouse Pad  2000       3,662.83    -21.45
Italy      Mouse Pad  2001       4,747.90    29.62
Italy      Y Box      1999       15,215.16
Italy      Y Box      2000       29,322.89   92.72
Italy      Y Box      2001       81,207.55   176.94
```

結果の中の空白のセルは NULL である点が重要です。この式の結果として、2 年前のデータに製品の値が存在しない場合は、NULL 値が返されます。どの製品も、1998 年度には値がないため、それぞれの製品について、1999 年の伸び率の計算結果は NULL となります。

ワイルド・カードと ANY キーワードの併用

ワイルド・カード演算子は、セルの指定に非常に有効です。モデルには、この目的のために ANY キーワードが用意されています。次に示すとおり、先の例の 1998 年から 2001 年までの年度という指定のかわりにこのキーワードを使用できます。

```
SELECT SUBSTR(country,1,10) country, SUBSTR(prod,1,10) prod,
year, sales, growth_pct
FROM sales_view
WHERE country='Italy'
MODEL RETURN UPDATED ROWS
  PARTITION BY (country)
  DIMENSION BY (prod, year)
  MEASURES (sale sales, 0 growth_pct)
  RULES (
    growth_pct[prod IN ('Bounce','Y Box','Mouse Pad'),
               ANY] =
        100* (sales[CV(prod), CV(year)] –
          sales[CV(prod), CV(year) -1] ) /
          sales[CV(prod), CV(year) -1] )
ORDER BY country, prod, year;


COUNTRY    PROD       YEAR       SALES       GROWTH_PCT
---------- ---------- ---------- ----------- ----------
Italy      Bounce     1999       2,474.78
Italy      Bounce     2000       4,333.69    75.11
Italy      Bounce     2001       4,846.30    11.83
Italy      Mouse Pad  1998       3,055.69
Italy      Mouse Pad  1999       4,663.24    52.61
Italy      Mouse Pad  2000       3,662.83    -21.45
Italy      Mouse Pad  2001       4,747.90    29.62
Italy      Y Box      1999       15,215.16
Italy      Y Box      2000       29,322.89   92.72
Italy      Y Box      2001       81,207.55   176.94
```

この問合せは、データ・セットが 1998 年から 2001 年までのものであり、それが先の問合せで指定された範囲のため、先の問合せと同じ結果を返します。

ANY をセル参照で使用して、NULL を含むすべてのディメンションをインクルードできます。シンボリック参照の表記法では、IS ANY というフレーズを使用します。ワイルド・カードの ANY を位置表記法またはシンボリック表記法で使用すると、セルの挿入ができなくなる点に注意してください。

FOR ループ: 新規セルを指定する簡潔な方法

FOR コンストラクトは、式の左辺で使用されるワイルド・カードと同様に動作し、単一の式によって新規のセルを挿入できるようになります。（FOR コンストラクトは式の左辺でのみ使用できます。）FOR の例として、一部の製品の 2005 年度の売上高を 2001 年度の売上高より 30%多く推定する次の式を検討します。

```
RULES
(  sales['Mouse Pad', 2005] = 1.3 * sales['Mouse Pad', 2001],
   sales['Bounce', 2005] = 1.3 * sales['Bounce', 2001],
   sales['Y Box', 2005] = 1.3 * sales['Y Box', 2001] )
```

位置表記法を式の左辺で使用することによって、これらの製品の 2005 年度のセルが、まだ配列に存在しない場合は挿入されることが保証されます。この方法では、製品と同数の式が必要なため、大きな式となります。何十もの製品を対象とするタスクの場合は、非常に扱いにくいアプローチとなります。FOR を使用すると、この計算の表現を変更して、簡潔で、しかも同様に動作する式が得られます。

```
SELECT SUBSTR(country,1,20) country, SUBSTR(prod,1,15) prod,
year, sales
FROM sales_view
WHERE country='Italy'
MODEL RETURN UPDATED ROWS
  PARTITION BY (country)
  DIMENSION BY (prod, year)
  MEASURES (sale sales)
  RULES (
    sales[FOR prod IN ('Mouse Pad', 'Bounce', 'Y Box'),
      2005] = 1.3 * sales[CV(prod), 2001] )
ORDER BY country, prod, year;


COUNTRY              PROD            YEAR       S
-------------------- --------------- ---------- ----------
Italy                Bounce          2005       6407.245
Italy                Mouse Pad       2005       6402.63
Italy                Y Box           2005       108308.304
```

前述の例のような指定を FOR キーワードなしに記述すると、既存のセルの更新のみが行われ、新規のセルは挿入されません。例のデータの場合は、行が 1 つも返されないことになります。次にキーワードなしの問合せを示します。

```
SELECT SUBSTR(country,1,20) country, SUBSTR(prod,1,15) prod,
year, sales
FROM sales_view
WHERE country='Italy'
MODEL RETURN UPDATED ROWS
  PARTITION BY (country)
  DIMENSION BY (prod, year)
  MEASURES (sale sales)
  RULES (
    sales[prod IN ('Mouse Pad', 'Bounce', 'Y Box'), 2005] =
      1.3 * sales[CV(prod), 2001] )
ORDER BY country, prod, year;


no rows selected
```

FOR コンストラクトは、1 つの式で位置ベースの参照を持つ複数の式を生成して、新規セルの作成を可能にする（UPSERT 動作）ためのツールとみなすことができます。

値のシーケンスにわたる FOR ループ

セル参照に必要なディメンション値が決まった間隔でシーケンスから得られる場合は、別の形式の FOR コンストラクトを使用できます。

```
FOR dimension FROM <value1> TO <value2>
   [INCREMENT | DECREMENT] <value3>
```

この指定により、value1 と value2 の間のディメンション値が、value1 から始まり、value3 の増分で大きくなる（または小さくなる）形で作成されます。FOR ループで値の範囲を使用することによって、多くの既存セルに適用される指定や多くの新規セルを作成する指定を非常に簡潔なものにできます。

その他のセル処理機能

式の評価順序

デフォルトでは、式は MODEL 句に現れる順に評価されます。SEQUENTIAL ORDER キーワードを MODEL 句で指定することにより、そのような評価順序を明示できます。モデルの計算で、式の依存関係が正しい順序で処理するためには、AUTOMATIC ORDER キーワードを使用します。多くの式がモデルに含まれている場合、式が論理的に正しい順序でリストされているか、手動で確認するよりも、AUTOMATIC ORDER オプションを使用する方が効率的です。自動順序機能を使用することにより、モデルの開発およびメンテナンスの効率が向上します。

NULL メジャーと欠損セル

SQL モデルを使用するアプリケーションは、セルのメジャーとして 2 つの形式の非決定論的値を処理できる必要があります。それは、配列内に存在するが NULL 値を割り当てられたセルと配列内に存在しないセルです。セル参照と呼ばれるが配列内に存在しないセルは、欠損セルと呼ばれます。MODEL 句のデフォルトの NULL の処理方法は、他の SQL タスクの場合と同じで、欠損セルはデフォルトでは NULL として処理されます。モデルには、そのようなセルの処理方法として別の方法も用意されています。IGNORE NAV キーワード（NAV は Non-Available Value の頭文字で「使用できない値」の意）をモデル・レベルまたは個々のルール・レベルで追加できます。このフレーズを使用すると、NULL 値および欠損セルを式の数値計算では 0 として、文字処理では空の文字列として処理できます。

参照モデル

既存のセルが更新され、新セルが追加されたマルチ・ディメンショナルな配列は、SQL メイン・モデルと呼ばれます。メイン・モデルとともに、MODEL 句によって 1 つまたは複数の、参照モデルという読込み専用のマルチディメンショナル配列を定義できます。参照モデルは検索表として機能します。参照モデルを使用すると、異なるディメンションの配列を式で参照できます。たとえば、収益予測では、税金参照配列とコスト参照配列を参照できますが、税金は国ごとにディメンション化され、コストは製品ごとにディメンション化されています。もう 1 つの例は、為替計算です。この場合、換算率は参照モデルとして扱われます。

参照モデルを結合する式、ワイルド・カードの ANY および CV（）関数は非常に高い柔軟性を備えています。

反復モデル

MODEL 句の ITERATE オプションを使用すると、式を指定した回数繰り返して評価できます。反復モデルを使用して、独立した式を含むモデルを計算できます。すなわち、この機能を使用して連立方程式を解くことができます。反復する回数は、ITERATE 句に対する引数として指定されます。必要に応じて、早期中止条件を指定して、最大反復回数に達する前に式の評価を中止することも可能です。この条件は、ITERATE の UNTIL 副次句で指定され、反復処理の最後にチェックされます。終了条件を、反復処理全体を通じて、セルの値の変更に基づいて指定できると便利な場合もあります。そこで、UNTIL 条件での現在の反復処理の前後のセル値へのアクセスを通して、そのような条件を指定するメカニズムが用意されています。また、計算に使用する現在の反復回数にアクセスできます。

結論

複雑な行間計算を SQL で指定する、という複雑なタスクは、従来、他の言語によるプログラムや様々なタイプの計算エンジンなどの、データベースの外部ソリューションによって対処されてきました。Oracle Database 10*g* は、SQL の MODEL 句を使用することにより、これらの課題を解決するための新しい手段を提供します。MODEL 句は、SELECT 文への拡張機能であり、リレーショナル・データを多次元配列として処理し、各セルに対して、簡潔で柔軟な表記法によるアクセスを可能にします。その結果、複雑な SQL の結合や和結合は不要となり、処理は最適化されます。MODEL 句は、式間の論理的依存関係を自動的に処理し、よって、計算開発およびメンテナンスを一段と簡素化します。MODEL 句によって活用される Oracle データベースのパラレル・クエリー処理機能は、エンタープライズ・レベルのスケーラビリティを実現します。

MODEL 句と、Oracle データベースの以前のリリースにおける SQL 分析関数の拡張機能の機能により、Oracle データベースは、高度な計算に対応できる堅牢なプラットフォームとなっています。高度な計算機能を SQL に統合することによって、データの DBMS からの抽出、外部処理、結果の Oracle データベースへの再挿入などが不要となるため、管理性が高まり、データベース統合が簡素化されます。この重要な新機能は、データベース・アプリケーションの全分野のアプリケーション、特にビジネス・インテリジェンスの分野のアプリケーションで活用できます。

付録: MODEL 句の EXPLAIN PLAN

EXPLAIN PLAN オペレーションの実行により、作成されたモデルを Oracle データベースが評価する際に採用しているアルゴリズムを検証できます。このモデルに SEQUENTIAL ORDER 式が含まれている場合は、ORDERED と表示されます。AUTOMATIC ORDER モデルの場合は、EXPLAIN PLAN ツールに、モデル内の式に周期的依存関係が存在するかどうかによって、ACYCLIC または CYCLIC と表示されます。さらに、EXPLAIN PLAN ツールの出力には、ORDERED モデルの場合は FAST という注釈が含まれます。また、左辺のセル参照がすべて単一のセル参照で、式の右辺に集計関数があり、それらが SUM、COUNT、AVG などの単純な算術計算を実行する非弁別的集計関数である場合は、ACYCLIC アルゴリズムが含まれます。この場合の式評価は、非常に効率的なため、FAST というラベルが付与されます。

ORDERED 処理を使用するプラン例

```
EXPLAIN PLAN FOR
SELECT SUBSTR(country,1,10) country,
  SUBSTR(prod,1,10) product, year, sales
FROM sales_view
WHERE country IN ('Italy','Brazil')
MODEL RETURN UPDATED ROWS
  PARTITION BY (country) DIMENSION BY (prod, year)
  MEASURES (sale sales)
  RULES SEQUENTIAL ORDER
  (
   sales['Bounce', 2003] =
     AVG(sales)[ANY, 2001] * 1.24,
   sales[prod != 'Y Box', 2000] =
     sales['Y Box', 2000] * 1.25
  );
```

この問合せに対する結果の最初の 5 行は次のとおりです。

```
COUNTRY              PRODUCT         YEAR       SALES
-------------------- --------------- ---------- ----------
Italy                Bounce          2000       36653.6125
Italy                Mouse Pad       2000       36653.6125
Italy                Music CD-R      2000       36653.6125
Italy                Fly Fishin      2000       36653.6125
Italy                Deluxe Mou      2000       36653.6125
```

この問合せは次の行で始まる EXPLAIN PLAN を作成します。

```
SELECT STATEMENT
  SQL MODEL ORDERED
```

第 2 の式の左辺はマルチセル参照なため、Oracle データベースは、この問合せに対しては、FAST メソッドを選択しません。

ACYCLIC FAST 処理を使用するプラン例

```
EXPLAIN PLAN FOR
SELECT SUBSTR(country,1,10) country,
  SUBSTR(prod,1,10) product, year, sales
FROM sales_view
WHERE country IN ('Italy','Brazil')
MODEL RETURN UPDATED ROWS
  PARTITION BY (country) DIMENSION BY (prod, year)
  MEASURES (sale sales)
  RULES AUTOMATIC ORDER
  (
    sales['Bounce', 2004] =
      sales['Y Box', 2001] * 0.25,
    sales['Mouse Pad', 2004] =
      sales['Mouse Pad', 2001] / SUM(sales)[ANY, 2001] *
      2 * sales['All Products', 2004],
    sales['All Products', 2004] = 200000
  );
```

この問合せは以下の結果を返します。

```
COUNTRY              PRODUCT         YEAR       SALES
-------------------- --------------- ---------- ----------
Italy                All Produc      2004       200000
Italy                Bounce          2004       20301.8875
Italy                Mouse Pad       2004       1342.86407
Brazil               All Produc      2004       200000
Brazil               Bounce          2004
Brazil               Mouse Pad       2004       2344.7976
```

この問合せは、次の行で始まる EXPLAIN PLAN を作成します。

```
SELECT STATEMENT
  SQL MODEL ACYCLIC FAST
```

このモデルで使用されている式は周期的ではないため、EXPLAIN PLAN ツールには ACYCLIC と表示されます。この場合、FAST メソッドが選択されたのは、FAST メソッドにより、付録の冒頭で述べた 2 つの要件が満足されるためです。

Oracle Database 10*g* の SQL MODEL 句
2003 年 8 月
著書: John Haydu

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

海外からのお問合せ窓口:
電話: +1.650.506.7000
ファックス: +1.650.506.7200
www.oracle.com