

RF-921

FORECASTING QUERIES IN ORACLE

Francisco Javier Moreno (Universidad Nacional de Colombia - Medellín, Colombia) -

fjmoreno@unalmed.edu.co

Anderson Franco García (Universidad Nacional de Colombia - Medellín, Colombia) -

afranco@unalmed.edu.co

Jaime Alberto Echeverri (Universidad de Medellín, Medellín, Colombia) -

jaecheverri@udem.edu.co

In order to make analysis and what if queries, Oracle has included to SQL language the MODEL clause. This type of queries can be posed in SQL standard too. In this article we compare both approaches. We present a set of queries through MODEL clause and SQL standard. The results evidence the advantages of the MODEL clause: the simplicity in the queries implementation and the computational saving that is reached avoiding joins, correlated subqueries and unions.

Keywords: model clause, sql, data warehouses, subqueries, joins.

CONSULTAS PARA PRONÓSTICOS EN ORACLE

Con el objetivo de hacer análisis y consultas de tipo *what-if* (qué pasaría si), Oracle ha adicionado al lenguaje SQL la cláusula MODEL. Este tipo de consultas se pueden plantear también en SQL estándar. En este artículo se comparan ambas aproximaciones. Se plantea un conjunto de consultas mediante la cláusula MODEL y mediante SQL estándar. Los resultados evidencian las ventajas de la cláusula MODEL: su simplicidad en el planteamiento de las consultas y el ahorro computacional que se logra al evitar reuniones (*joins*), subconsultas correlacionadas y uniones.

Palabras claves: cláusula MODEL, SQL, bodegas de datos, subconsultas, reuniones.

1. INTRODUCCIÓN

Muchas aplicaciones que accesan bodegas de datos (Inmon, 2002), (Kimball, 2002) requieren cálculos complejos, por ejemplo para hacer consultas y análisis hipotéticos o de tipo *what if* (qué pasaría si). Estos análisis sirven para pronosticar el comportamiento del negocio en un escenario hipotético generado a partir de datos históricos.

Generalmente los analistas formulan un posible escenario y suponen que se han adoptado determinadas políticas, lo que deriva en un mundo hipotético, que se puede explorar mediante consultas (Balmin, 2000).

Este tipo de consultas implican generalmente en SQL estándar (SQL:2003) (Eisenberg, 2004) auto-reuniones (*self-joins*), subconsultas correlacionadas y uniones. Su planteamiento es complejo y requiere un alto procesamiento computacional.

Con el objetivo de facilitar el planteamiento y mejorar el desempeño de este tipo de consultas, Oracle ha adicionado a SQL la cláusula MODEL. En este artículo se compara esta cláusula con consultas equivalentes en SQL estándar.

El artículo se estructura así: en la Sección 2 se expone la cláusula MODEL, en la Sección 3 se presenta un conjunto de consultas de tipo *what if* implementadas con la cláusula MODEL y con SQL estándar. Finalmente, en la Sección 4 se presentan conclusiones y trabajos futuros.

2. CLÁUSULA MODEL

La cláusula MODEL simplifica y facilita el planteamiento de consultas *what if*. Los datos de una tabla se acceden en forma de arreglos multidimensionales, donde los subíndices del arreglo se refieren a las *dimensiones* y el contenido de las celdas a las medidas de los *hechos*. Los hechos son fenómenos medibles de interés para una organización, por ejemplo las ventas. Las dimensiones son características que califican a los hechos. Por ejemplo, lugar (tienda), producto, cliente y tiempo (fecha); son dimensiones típicas de una venta.

A partir de una tabla de hechos, se pueden plantear consultas con el fin de hacer pronósticos. Estas consultas conllevan generalmente fórmulas complejas. MODEL facilita el planteamiento de tales fórmulas. Por ejemplo, pronosticar el total de ventas para los próximos cinco años a partir de las ventas de la última década con una fórmula determinada. Las fórmulas generan resultados que pueden incluso usarse para el cálculo de otras fórmulas en la misma sentencia, como en el ejemplo d) de la Sección 3.

La sintaxis esencial de MODEL es:

```
<cláusulas de la sentencia SELECT>
MODEL [RETURN UPDATED ROWS]
[PARTITION BY (<cols>)]
DIMENSION BY (<cols>)
MEASURES (<cols>)
[RULES
[AUTOMATIC ORDER | SEQUENTIAL ORDER]
( <asignación de celdas> = <expresión> ... )
<ORDER BY>
```

MODEL se ubica al final de la sentencia SELECT y antes de la cláusula ORDER BY (si la hay).

Las partes de la cláusula MODEL son:

- **PARTITION BY:** especifica las columnas que clasifican los datos en conjuntos (particiones). Cada conjunto es considerado por las fórmulas como un arreglo independiente.
- **DIMENSION BY:** especifica las columnas usadas como dimensiones. Éstas actúan como subíndices en cada uno de los arreglos especificados mediante PARTITION BY.
- **MEASURES:** identifica las columnas correspondientes a las celdas de los arreglos. Típicamente son valores numéricos como unidades vendidas o costos. Cada celda se accede por medio de una combinación de dimensiones (especificadas en DIMENSION BY).
- **RULES:** define las fórmulas para los cálculos. En el término <asignación de celdas> se especifican las celdas que se van a actualizar o insertar por medio de la fórmula especificada en <expresión>.

La opción RETURN UPDATED ROWS limita los resultados a aquellas filas que fueron creadas o actualizadas mediante la expresión <asignación de celdas> (Oracle, 2006). Las opciones AUTOMATIC ORDER y SEQUENTIAL ORDER se explican en el ejemplo d) de la Sección 3.

3. CONSULTAS *WHAT IF*

Sea una tabla de hechos ventas (*sales*), con dimensiones país (*country*), producto (*prod*) y año (*year*), y la medida total (*sale*) (Haydu, 2003)). Una muestra de datos se presenta en la Tabla 1.

COUNTRY	PROD	YEAR	SALE
Italia	bounce	1998	200
Italia	bounce	1999	100
Italia	bounce	2000	100
Italia	bounce	2001	200
Italia	bounce	2002	301
Italia	bounce	2003	300
Italia	mp	1998	98
Italia	mp	1999	99
Italia	mp	2000	100
Italia	mp	2001	101
Italia	mp	2002	303
Italia	mp	2003	302
Italia	ybox	1998	200
Italia	ybox	1999	100
Italia	ybox	2000	50
Italia	ybox	2001	80
Italia	ybox	2002	305
Italia	ybox	2003	304
Japón	bounce	1998	400
Japón	bounce	1999	410
Japón	bounce	2000	100
Japón	bounce	2001	200
Japón	bounce	2002	301
Japón	bounce	2003	300
Japón	mp	1998	410
Japón	mp	1999	450
Japón	mp	2000	460
Japón	mp	2001	470
Japón	mp	2002	303
Japón	mp	2003	302
Japón	ybox	1998	420
Japón	ybox	1999	430
Japón	ybox	2000	50
Japón	ybox	2001	80
Japón	ybox	2002	305
Japón	ybox	2003	304

Tabla 1. Tabla de hechos.

Sean las consultas:

a) Supóngase que se desea pronosticar el total de ventas del producto 'bounce' para 2005 en cada país así: el total será igual al valor máximo de las ventas del producto entre 1998 y 2002 en el respectivo país, más 100 unidades.

Consulta mediante MODEL:

```

SELECT country, prod, year, sales
FROM sales
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]);

```

Los resultados se muestran en la Tabla 2.

COUNTRY	PROD	YEAR	SALE
Italia	bounce	2005	401
Japón	bounce	2005	600

Tabla 2. Resultado de la consulta del ejemplo a).

Consulta en SQL estándar:

```

SELECT DISTINCT country, prod, 2005 AS year, 100 + (SELECT MAX(sale)
FROM sales AS s2
WHERE s2.prod = s1.prod AND s2.country = s1.country
AND s2.year BETWEEN 1998 AND 2002) AS sale
FROM sales AS s1
WHERE prod = 'bounce';

```

La consulta con MODEL a diferencia de la versión con SQL estándar, evita la subconsulta escalar correlacionada (que en el fondo conlleva una auto-reunión como se muestra a continuación). Además el planteamiento de la consulta con MODEL es más simple e intuitivo.

Consulta en SQL estándar con auto-reunión:

```

SELECT s1.country, s1.prod, 2005 AS year, 100 +
MAX(s2.sale) AS sale
FROM sales AS s1, sales AS s2
WHERE s1.prod = 'bounce' AND
s2.country = s1.country AND
s2.prod = s1.prod AND
s2.year BETWEEN 1998 AND 2002
GROUP BY s1.country, s1.prod;

```

En la Figura 1 se muestra el plan de ejecución en Oracle para las tres sentencias. Las sentencias en SQL implican dos accesos a la tabla *sales* y operaciones de reunión o agregación; mientras que la consulta con MODEL sólo hace un acceso a la tabla. Además, la columna *cost*, que da una idea del costo general de una consulta, presenta un menor valor para la consulta con MODEL.

Plan de ejecución de la consulta mediante la cláusula MODEL

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		28	588	3 (0)
1	SQL MODEL ORDERED		28	588	
2	TABLE ACCESS FULL	SALES	28	588	3 (0)

Plan de ejecución de la consulta en SQL estándar

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		10	130	4 (25)
1	SORT AGGREGATE		1	21	
* 2	TABLE ACCESS FULL	SALES	4	84	3 (0)
3	HASH UNIQUE		10	130	4 (25)
* 4	TABLE ACCESS FULL	SALES	10	130	3 (0)

Plan de ejecución de la consulta en SQL estándar con auto-reunión

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		13	442	8 (25)
1	HASH GROUP BY		13	442	8 (25)
* 2	HASH JOIN		13	442	7 (15)
* 3	TABLE ACCESS FULL	SALES	8	168	3 (0)
* 4	TABLE ACCESS FULL	SALES	10	130	3 (0)

Figura 1. Plan de ejecución para las tres sentencias.

b) En este ejemplo se presenta una consulta que calcula el total de las ventas del producto 'bounce' en cada país entre 1998 y 2001, basadas en las ventas de los productos 'mp' y 'ybox' para el respectivo país.

En la consulta se usa la función **CV()** que "copia" valores de celdas referenciadas en el lado izquierdo al lado derecho de la fórmula. Aquí se utiliza la expresión **CV(year)** para copiar el respectivo año (entre 1998 y 2001) del lado izquierdo al lado derecho de la fórmula.

Consulta mediante MODEL:

```
SELECT country, prod, year, sales
FROM sales
MODEL RETURN UPDATED ROWS
PARTITION BY (country)
DIMENSION BY (prod, year)
MEASURES (sale sales)
RULES (
sales['bounce', year BETWEEN 1998 AND 2001] = sales['mp', CV(year)] +
sales['ybox', CV(year)] )
ORDER BY country, prod, year;
```

Los resultados se muestran en la Tabla 3.

COUNTRY	PROD	YEAR	SALE
Italia	bounce	1998	298
Italia	bounce	1999	199
Italia	bounce	2000	150
Italia	bounce	2001	181
Japón	bounce	1998	860
Japón	bounce	1999	880
Japón	bounce	2000	510
Japón	bounce	2001	550

Tabla 3. Resultado de la consulta del ejemplo b).

Consulta en SQL estándar:

```

SELECT country, prod, year, (SELECT sale
                             FROM sales
                             WHERE prod = 'mp' AND year = s.year AND
                             country = s.country) +
                             (SELECT sale
                              FROM sales
                              WHERE prod = 'ybox' AND year = s.year AND country =
                              s.country) AS total
FROM sales AS s
WHERE prod = 'bounce' AND
      year BETWEEN 1998 AND 2001
ORDER BY country, prod, year;

```

En este ejemplo, MODEL evita dos auto-reuniones (camufladas en las dos subconsultas escalares correlacionadas). Las subconsultas equivalen a las reglas implementadas en la sección RULES. Aquí la columna *cost* del plan de ejecución generó un valor de 3 unidades para ambas soluciones; sin embargo la columna *operation* presentó sólo un acceso a la tabla *sales* para la consulta con MODEL, mientras que en la consulta en SQL estándar presentó tres accesos, lo que sugiere un mejor rendimiento en la consulta planteada con MODEL.

c) El siguiente ejemplo calcula el crecimiento porcentual entre 1998 y 2001 de las ventas de los productos 'ybox' y 'bounce' en cada país. Éste se calcula así para 1998:

$$\text{growth_pct}(1998) = 100 * (\text{sales}(1998) - \text{sales}(1997)) / \text{sales}(1997).$$

De forma similar se hace el cálculo para los otros años.

Consulta mediante MODEL:

```

SELECT country, prod, year, sales, growth_pct
FROM sales
MODEL RETURN UPDATED ROWS
PARTITION BY (country)
DIMENSION BY (prod, year)
MEASURES (sale sales, 0 growth_pct)
RULES (
growth_pct[prod IN ('bounce', 'ybox'),
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;

```

Los resultados se muestran en la Tabla 4.

COUNTRY	PROD	YEAR	SALE	GROWTH_PCT
Italia	bounce	1998	200	
Italia	bounce	1999	100	-50
Italia	bounce	2000	100	0
Italia	bounce	2001	200	100
Italia	ybox	1998	200	
Italia	ybox	1999	100	-50
Italia	ybox	2000	50	-50
Italia	ybox	2001	80	60
Japón	bounce	1998	400	
Japón	bounce	1999	410	2.5
Japón	bounce	2000	500	21.9512195
Japón	bounce	2001	200	-60
Japón	ybox	1998	420	
Japón	ybox	1999	430	2.38095238
Japón	ybox	2000	50	-88.372093
Japón	ybox	2001	80	60

Tabla 4. Resultado de la consulta del ejemplo c).

Consulta en SQL estándar:

```
SELECT country, prod, year, sale sales, 100 * (sale -
      (SELECT sale
        FROM sales
        WHERE prod = s.prod AND
        year = s.year - 1 AND
        country = s.country))/
      (SELECT sale
        FROM sales
        WHERE prod = s.prod AND
        year = s.year - 1 AND
        country = s.country)
      AS growth_pct
FROM sales AS s
WHERE year BETWEEN 1998 AND 2001 AND prod IN ('bounce', 'ybox')
ORDER BY country, prod, year;
```

En este ejemplo se crea una medida llamada *growth_pct*. Ésta se calcula a partir de las ventas del producto correspondientes al mismo año y al año anterior. Para 1998, de cada uno de los productos se obtienen nullos, ya que no hay datos para 1997. De nuevo la consulta con MODEL es más simple y eficiente. Por ejemplo, las fórmulas para hacer los pronósticos se visualizan directamente en la sección RULES, mientras que en la versión en SQL estándar las fórmulas se camuflan en medio de las subconsultas escalares. De forma similar al ejemplo b), el plan de ejecución presenta en la columna *cost* para ambas soluciones un valor de 4 unidades, sin embargo la columna *operation* muestra un solo acceso a la tabla *sales* en la consulta con MODEL y tres accesos en la consulta con SQL estándar.

d) La siguiente consulta calcula las ventas en Italia y Japón, para el producto 'bounce' en 2002 como la suma de las ventas del 2001 y 2000 del mismo producto respectivamente en Italia y Japón. Las ventas del producto 'ybox' en el 2002 en Italia y Japón se igualan al valor de las ventas en 2001 de 'ybox' en Italia y Japón respectivamente. Además se calcula el valor para 2002 en Italia y Japón, de un nuevo producto llamado '2_products', que se calcula como la suma de los valores recién generados para los productos 'bounce' y 'ybox'.

Consulta mediante MODEL:

```
SELECT country, prod, year, sales
FROM sales
WHERE country IN ('Italia','Japón')
MODEL RETURN UPDATED ROWS
PARTITION BY (country)
DIMENSION BY (prod, year)
MEASURES (sale sales)
RULES AUTOMATIC ORDER (
sales['2_Products', 2002] = sales['bounce', 2002] +
sales['ybox', 2002],
sales['bounce', 2002] = sales['bounce', 2001] +
sales['bounce', 2000],
sales['ybox', 2002] = sales['ybox', 2001])
ORDER BY country, prod, year;
```

Los resultados se muestran en la Tabla 5.

COUNTRY	PROD	YEAR	SALE
Italia	bounce	2002	300
Italia	ybox	2002	80
Italia	2_Products	2002	380
Japón	bounce	2002	700
Japón	ybox	2002	80
Japón	2_Products	2002	780

Tabla 5. Resultado de la consulta del ejemplo d).

Consulta en SQL estándar:

Véase el Apéndice A.

En la consulta con MODEL se observa como se usan valores recién calculados en las reglas en otras reglas en la misma sentencia. Aquí se usan los valores calculados en las dos primeras reglas y se aplican en la tercera regla.

El planteamiento de esta consulta en SQL estándar es complejo. En el Apéndice A se presenta una solución que usa la cláusula WITH (Burlison, 2006) (también se pudo haber creado una vista); allí se requieren varias subconsultas (que camuflan auto-reuniones) lo que dificulta su planteamiento. Además el rendimiento es bajo al compararlo con la versión de la consulta con MODEL. En el

plan de ejecución la columna *cost* presentó un valor de 4 unidades en la consulta con MODEL y un valor de 7 unidades en la consulta con SQL estándar.

El orden de evaluación de las reglas puede ser especificado de dos formas:

RULES SEQUENTIAL ORDER

Ésta es la opción predeterminada. Implica que las reglas en la cláusula sean evaluadas en el orden en que fueron escritas, sin tener en cuenta las dependencias lógicas entre los valores.

RULES AUTOMATIC ORDER

Esta opción permite que Oracle determine el orden de evaluación de las reglas teniendo en cuenta la dependencia lógica de los valores.

En el ejemplo anterior la primera regla debe evaluarse después de las dos siguientes, por lo que es necesario definir el orden automático ya que no fueron escritas en un orden lógico (Oracle, 2005).

e) El siguiente ejemplo usa ciclos FOR para pronosticar el valor de las ventas de los productos 'mp', 'bounce' y 'ybox' del 2004 al 2006. Los valores para cada año se calculan con base en el incremento que se dio en los dos años anteriores y para cada producto respectivamente (Gennick, 2004).

La fórmula para calcular el valor de las ventas para 2004 es:

$$\text{sales}(2004) = ((\text{sales}(2003) - \text{sales}(2002)) / \text{sales}(2002)) * \text{sales}(2003) + \text{sales}(2003).$$

De forma similar se calcula el valor para 2005 y 2006.

Consulta mediante MODEL:

```
SELECT country, prod, year, sales
FROM sales
MODEL RETURN UPDATED ROWS
PARTITION BY (country)
DIMENSION BY (prod, year)
MEASURES (sale sales)
RULES (
sales[FOR prod IN ('mp', 'bounce', 'ybox') ,FOR year FROM 2004 TO 2006
INCREMENT 1] = ((sales[CV(),CV() - 1] - sales[CV(),CV() - 2]) / sales[CV(),CV() - 2]) * sales[CV(),CV() - 1] + sales[CV(),CV() - 1])
ORDER BY country, prod, year;
```

Los resultados se muestran en la Tabla 6.

COUNTRY	PROD	YEAR	SALE
Italia	bounce	2004	299.003322
Italia	bounce	2005	298.009956
Italia	bounce	2006	297.019889
Italia	mp	2004	301.0033
Italia	mp	2005	300.00989
Italia	mp	2006	299.019758
Italia	ybox	2004	303.003279
Italia	ybox	2005	302.009825
Italia	ybox	2006	301.019629
Japón	bounce	2004	299.003322
Japón	bounce	2005	298.009956
Japón	bounce	2006	297.019889
Japón	mp	2004	301.0033
Japón	mp	2005	300.00989
Japón	mp	2006	299.019758
Japón	ybox	2004	303.003279
Japón	ybox	2005	302.009825
Japón	ybox	2006	301.019629

Tabla 6. Resultado de la consulta del ejemplo e).

Al igual que en el ejemplo d), la consulta equivalente en SQL es muy compleja y no se presenta por razones de espacio.

4. CONCLUSIONES Y TRABAJOS FUTUROS

Se demostraron las ventajas de la cláusula MODEL frente a consultas equivalentes en SQL estándar para el planteamiento de consultas de tipo *what if*.

Este tipo de consultas requieren en SQL estándar subconsultas correlacionadas o reuniones y en algunas ocasiones uniones y vistas auxiliares, que se evitan mediante MODEL.

Aunque se presentaron métricas de rendimiento propias de Oracle que soportan las conclusiones enunciadas, hacia el futuro se planea hacer un estudio más completo al respecto, mediante herramientas como TKPROF (Oracle, 2001).

Igualmente se planea diseñar un mecanismo para transformar, bajo determinadas condiciones, una consulta que implique subconsultas correlacionadas que camuflen auto-reuniones, en una versión equivalente con MODEL.

Agradecimientos.

Este trabajo es parte del proyecto “Modelo Multidimensional Espacio-Temporal y su correspondiente Lenguaje de Consulta”, de la Convocatoria Nacional de Investigación 2007, Modalidad 2, Universidad Nacional de Colombia Sede Medellín y se desarrolla en el marco del Doctorado en Ingeniería de Sistemas de la misma universidad, auspiciado por Colciencias, del que el segundo autor es becario.

5. BIBLIOGRAFÍA

Balmin, Y. Papakonstantinou, Papadimitriou, T. Hypothetical Analysis in OLAP Environment, 2000.

Burleson, D. Oracle Tuning The Definitive Reference. Abril 2006.

Eisenberg, A., Melton, J., Krishna, K., Kulkarni, G., Michels, J., Zemke, F. SQL: 2003 has been published. SIGMOD Record 33(1): 119-126. 2004.

Gennick, J. Announcing the New MODEL. Enero 2004.
http://www.oracle.com/technology/oramag/oracle/04-jan/o14tech_sql.html

Haydu, J. The SQL Model Clause of Oracle Database 10g. Agosto. 2003

Inmon, W. Building the Data Warehouse, Marzo. 2002.

Kimball, R. The Data Warehouse Toolkit, Abril. 2002.

Oracle. Using the SQL MODEL Clause to Define Interrow Calculations, tutorial.2006.
<http://www.oracle.com/technology/obe/obe10gdb/bidw/sqlmodel/sqlmodel.htm>

Oracle. Oracle Database Data Warehousing Guide 10g Release 2 (10.2).
Capítulo 22 SQL for Modeling. Diciembre 2005
http://download-east.oracle.com/docs/cd/B19306_01/server.102/b14223/sqlmodel.htm

Oracle. Oracle9i Database Performance Guide and Reference Release 1 (9.0.1).
Capítulo 10 Using SQL Trace and TKPROF. Junio 2001

APÉNDICE A

```

WITH parcial AS (
SELECT DISTINCT country, prod, 2002 year,
      (SELECT sale
       FROM sales AS s2
       WHERE s2.prod = s1.prod AND s2.country = s1.country AND
s2.year = 2001) +
      (SELECT sale
       FROM sales AS s2
       WHERE s2.prod = s1.prod and s2.country = s1.country and
s2.year = 2000) AS valor
FROM sales AS s1
WHERE country in ('Japón', 'Italia') AND
      prod = 'bounce'
UNION
SELECT DISTINCT country, prod, 2002 year,
      (SELECT sale
       FROM sales AS s2
       WHERE s2.prod = s1.prod AND s2.country = s1.country
AND s2.year = 2001) AS valor
FROM sales AS s1
WHERE country IN ('Japón', 'Italia') AND
      prod = 'ybox')
SELECT *
FROM parcial
UNION
SELECT DISTINCT country, '2_products' prod, 2002 year, (SELECT SUM(valor)
FROM parcial AS p
WHERE p.country = s2.country) AS valor
FROM sales AS s2
WHERE country IN ('Japón', 'Italia');

```