

CONSULTAS HIPOTÉTICAS: LA CLÁUSULA MODEL EN SQL

Hypotetical queries: MODEL clause in SQL

RESUMEN

Con el objetivo de hacer análisis históricos y predictivos mediante consultas de tipo *what-if* (qué pasaría si), Oracle ha adicionado al lenguaje SQL la cláusula MODEL. Aunque este tipo de consultas se pueden plantear también en SQL estándar, su planteamiento se facilita con esta cláusula. En este artículo se comparan ambas aproximaciones. Para ello se plantea un conjunto de consultas planteadas con la cláusula MODEL y con 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, dimensiones, medidas, subconsultas, reuniones.

ABSTRACT

In order to make historical and predictive analysis using what if queries, Oracle has included to the SQL language the MODEL clause. Although this type of queries can be expressed in SQL standard, their formulation is facilitated with this clause. In this article we compare both approaches. We present a set of queries formulated with the MODEL clause and with SQL standard. The results show the advantages of the MODEL clause: the simplicity in the queries formulation and the computational saving that is reached avoiding joins, correlated subqueries and unions.

KEYWORDS: MODEL clause, SQL, data warehouses, dimensions, measures, subqueries, joins.

1. INTRODUCCIÓN

Muchas aplicaciones que accesan bodegas de datos [1], [2] requieren cálculos complejos, por ejemplo para hacer análisis históricos y predictivos mediante consultas de tipo *what-if* (qué pasaría si). Estos análisis ayudan en el pronóstico del comportamiento de determinadas actividades del negocio, en escenarios hipotéticos generados a partir de datos históricos.

Usualmente 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 [3].

Este tipo de consultas implican usualmente en SQL estándar (SQL:2008) [4] auto-reuniones (*self-joins*), subconsultas correlacionadas y uniones, cuyo 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 hipotéticas (*what if queries*). 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 de la venta); son dimensiones típicas de una venta.

A partir de una tabla de hechos, se pueden plantear consultas con el fin de hacer pronósticos. Por ejemplo, pronosticar el total de ventas de un producto para los próximos cinco años a partir de las ventas de la última década con una fórmula determinada (ver ejemplos en la Sección 3). Estas consultas requieren usualmente fórmulas complejas. La cláusula MODEL facilita el planteamiento de tales fórmulas. Las fórmulas generan resultados que se pueden usar incluso en otras fórmulas

FRANCISCO JAVIER MORENO

Universidad Nacional de Colombia,
Sede Medellín
fjmoreno@unalmed.edu.co

JAIME ALBERTO ECHEVERRI

Universidad de Medellín,
jaecheverri@udem.edu.co

ANDERSON FRANCO GARCÍA

Universidad Nacional de Colombia,
Sede Medellín
afranco@unalmed.edu.co

en la misma sentencia, como en el ejemplo d) de la Sección 3.

La sintaxis esencial de la cláusula 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>
```

La cláusula MODEL se coloca al final de la sentencia SELECT y antes de la cláusula ORDER BY (si la hay). La cláusula MODEL posee las siguientes partes:

- **PARTITION BY:** especifica las columnas que clasifican los datos en conjuntos (particiones). Cada conjunto se trata como un arreglo.
- **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 tales 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 crear por medio de la fórmula especificada en <expresión>.

La opción RETURN UPDATED ROWS limita los resultados a aquellas filas que fueron actualizadas o creadas mediante la expresión <asignación de celdas> [5]. 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*) [6]. 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	Desk	1998	98
Italia	Desk	1999	99
Italia	Desk	2000	100
Italia	Desk	2001	101
Italia	Desk	2002	303
Italia	Desk	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	Desk	1998	410
Japón	Desk	1999	450
Japón	Desk	2000	460
Japón	Desk	2001	470
Japón	Desk	2002	303
Japón	Desk	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 *Sales*.

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 de ese 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 fórmula 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 estándar 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, presentó un menor valor (3) 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		1	130	3 (0)
1	SQL MODEL ORDERED		28	588	1
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	1
* 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, basado en las ventas de los productos 'Desk' y 'Ybox' para el respectivo país.

En la consulta se usa la función *CV* (*current value*) que "copia" valores de celdas referenciadas en el lado izquierdo al lado derecho de la fórmula. Aquí se usa 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['Desk', 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 = 'Desk'
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 la consulta con MODEL y un valor de 5 unidades para la consulta en SQL estándar. Además la columna *operation* presentó sólo un acceso a la tabla *sales* para la consulta con MODEL, mientras que 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 valor se obtiene 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 nulos, ya que no hay datos para 1997. De nuevo la consulta con MODEL es más simple y eficiente. Además las fórmulas para expresar 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 presentó en la columna *cost* un menor valor para la consulta con MODEL (6 vs. 4 unidades). Además 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.

El planteamiento de esta consulta en SQL estándar es complejo. En el Apéndice A se presenta una solución que usa la cláusula WITH [7]; 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 8 unidades en la consulta con SQL estándar.

En esta consulta con MODEL se observa cómo se pueden usar valores recién calculados en una regla en otra regla. Aquí por ejemplo se usan los valores calculados en las dos últimas reglas y se aplican en la primera regla.

Respecto al orden de evaluación de las reglas, en este ejemplo, las dos últimas reglas se deben evaluar antes de la primera. El orden de evaluación de las reglas se puede especificar de dos formas [8]:

RULES SEQUENTIAL ORDER

Ésta es la opción predeterminada. Implica que las reglas en la cláusula son evaluadas en el orden en que fueron escritas (de arriba hacia abajo), 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. Como en el ejemplo anterior la primera regla que aparece escrita se debe evaluar luego de las otras dos reglas, es necesario definir el orden automático.

e) El siguiente ejemplo usa ciclos FOR para pronosticar el valor de las ventas de los productos ‘Desk’, ‘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 [9].

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 ('Desk', '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	Desk	2004	301,0033
Italia	Desk	2005	300,00989
Italia	Desk	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	Desk	2004	301,0033
Japón	Desk	2005	300,00989
Japón	Desk	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 la cláusula 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 [10].

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

Agradecimientos.

Este trabajo se desarrolla en el marco del Doctorado en Ingeniería de Sistemas de la misma universidad, auspiciado por Colciencias, del que el primer autor es becario.

5. BIBLIOGRAFÍA

- [1] W. H. Inmon, Building the Data Warehouse, Wiley, 2005.
- [2] R. Kimball, M. Ross, W. Thornthwaite, J. Mundy, B. Becker, The Data Warehouse Lifecycle Toolkit, Wiley, 2008.
- [3] A. Balmin, Y. Papakonstantinou, T. Papadimitriou, Optimization of Hypothetical Analysis in OLAP Environment, 16th International Conference on Data Engineering, San Diego, 2000.
- [4] International Organization for Standardization ISO, SQL:2008 ISO/IEC 9075(1-4,9-11,13,14), Documento Oficial Estándar Internacional, Nueva York, 2008. Disponible en:
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=45498
- [5] Oracle Corp., Using the SQL MODEL Clause to Define Interrow Calculations, Oracle Tutorial, 2006. Disponible en:
<http://www.oracle.com/technology/obe/obe10gdb/bidw/sqlmodel/sqlmodel.htm>
- [6] Oracle Corp., The SQL Model Clause of Oracle Database 10g, Oracle White Paper, 2003. Disponible en:
http://www.oracle.com/technology/products/bi/pdf/10gr1_twp_bi_dw_sqlmodel.pdf
- [7] A. Danchenkov, D. Burlison, Oracle Tuning: the Definitive Reference, Rampant Techpress, 2006.
- [8] Oracle Corp., Oracle Database Data Warehousing Guide 10g Release 2 (10.2), Capítulo 22: SQL for Modeling, 2005. Disponible en:
http://download-east.oracle.com/docs/cd/B19306_01/server.102/b14223/sqlmodel.htm
- [9] J. Gennick, Announcing the New MODEL, Oracle Technology Network, 2004. Disponible en:
http://www.oracle.com/technology/oramag/oracle/04-jan/o14tech_sql.html
- [10] Oracle Corp., Oracle Database Performance Tuning Guide 10g Release 2 (10.2), Capítulo 20: Using Application Tracing Tools, 2005. Disponible en:
http://download.oracle.com/docs/cd/B19306_01/server.102/b14211/toc.htm

APÉNDICE A

```
WITH parcial AS (
SELECT DISTINCT country, prod, 2002 AS 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 AS 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' AS prod,
                2002 AS year,
                (SELECT SUM(valor)
                 FROM parcial AS p
                 WHERE p.country = s2.country) AS valor
FROM sales AS s2
WHERE country IN ('Japón', 'Italia');
```