

SQL Analítico - PIVOT

Laboratorio de Bases de Datos

PIVOT

Este boletín contén varias exercicios para practicar a expresión PIVOT. Para iso, usarase o almacén de datos sobre *ventas*.

1. Pivot

1. Dada a seguinte consulta, define unha nova que devolva o mesmo resultado onde os meses de xaneiro (1), febreiro (2) e marzo (3) aparezan como columnas.

```
01 | SELECT calendar_month_number, calendar_year, COUNT(*)
02 | FROM   dwsales s JOIN dwtimes t
03 |       ON s.time_id = t.time_id
04 | WHERE  calendar_month_number BETWEEN 1 AND 3
05 | GROUP BY calendar_month_number, calendar_year
```

Solución:

```
01 | SELECT *
02 | FROM (SELECT calendar_month_number, calendar_year
03 |       FROM   dwsales s JOIN dwtimes t
04 |             ON s.time_id = t.time_id)
05 | PIVOT (COUNT(*) FOR calendar_month_number IN (1, 2, 3))
```

2. Modifica a consulta, aplicando alias, para que as columnas se chamen 'Xaneiro', 'Febreiro' e 'Marzo', respectivamente.

Solución:

```
01 | SELECT *
02 | FROM (SELECT calendar_month_number, calendar_year
03 |       FROM   dwsales s JOIN dwtimes t
04 |             ON s.time_id = t.time_id)
05 | PIVOT (COUNT(*) FOR calendar_month_number
06 |       IN (1 "Xaneiro", 2 "Febreiro", 3 "Marzo"))
```

3. Deseña unha consulta que mostre os ingresos (*amount_sold*) medios por cada trimestre (*calendar_quarter_number*) de cada ano (*calendar_year*). Cada trimestre correspóndese a unha columna. O resultado debe estar ordenado por ano.

Solución:

```
01 | SELECT *
02 | FROM (SELECT calendar_quarter_number, calendar_year,
03 |           amount_sold
04 |           FROM dwsales s JOIN dwtimes t
05 |           ON s.time_id = t.time_id)
06 | PIVOT (AVG(amount_sold) FOR calendar_quarter_number
07 |       IN (1 "T1", 2 "T2", 3 "T3", 4 "T4"))
08 | ORDER BY calendar_year DESC
```

4. Modifica a consulta anterior para que, ademais dos ingresos, mostre a suma de cantidade (*quantity_sold*) de produtos vendidos (igualmente por cada trimestre de cada ano).

Solución:

```
01 | SELECT *
02 | FROM (SELECT calendar_quarter_number, calendar_year,
03 |           amount_sold, quantity_sold
04 |           FROM dwsales s JOIN dwtimes t
05 |           ON s.time_id = t.time_id)
06 | PIVOT (AVG(amount_sold) AS "Media",
07 |       SUM(quantity_sold) AS "Cantidade"
08 |       FOR calendar_quarter_number
09 |       IN (1 "T1", 2 "T2", 3 "T3", 4 "T4"))
10 | ORDER BY calendar_year
```

5. Mostra por cada cliente (debe visualizarse o identificador, nome e apelido) o importe total gastado cada ano entre os anos 1998 e 2000. Os anos deben aparecer en columnas. Ordenar o resultado polo apelidos e despois polo nome.

Solución:

```
01 | SELECT *
02 | FROM (SELECT amount_sold, calendar_year,
03 |           dc.cust_id,
04 |           dc.cust_first_name, dc.cust_last_name
05 |           FROM dwsales ds JOIN dwtimes dt
06 |           ON ds.time_id = dt.time_id
07 |           JOIN dwcustomers dc
08 |           ON ds.cust_id = dc.cust_id
09 |           WHERE calendar_year BETWEEN 1998 AND 2000)
10 | PIVOT (SUM(amount_sold) FOR calendar_year
11 |       IN (1998, 1999, 2000))
12 | ORDER BY cust_last_name, cust_first_name
```

6. Mostra a cantidade media de produtos vendidos polas categorías (de produto) 'Women', 'Men', 'Girls' e 'Boys' en cada categoría da promoción (*promo_category*). Excluir aquelas vendas onde o importe foi inferior a 1.000 euros e aquelas que non teñen promoción (*NO PROMOTION*). As categorías de produto debe mostrar cada unha nunha columna diferente. Ordenar o resultado pola categoría da promoción.

Solución:

```
01 | SELECT *
02 | FROM (SELECT quantity_sold, prod_category, promo_category
03 |       FROM dwsales ds JOIN dwproducts dp
04 |                       ON ds.prod_id = dp.prod_id
05 |                       JOIN dwpromotions dpr
06 |                       ON ds.promo_id = dpr.promo_id
07 |       WHERE amount_sold < 1000 AND ds.promo_id != 999)
08 | PIVOT (avg(quantity_sold) FOR prod_category
09 |       IN ('Women', 'Men', 'Boys', 'Girls'))
10 | ORDER BY promo_category
```

2. Consultas avanzadas con PIVOT

7. Modifica a consulta 5 para que as columnas se chamen *ano1*, *ano2* e *ano3* para 1988, 1999 e 2000. Ademais, engade unha nova columna “Total” que devolva o importe total gastado neses 3 anos por cada cliente.

Solución:

```
01 | SELECT cust_first_name, cust_last_name,
02 |       ano1, ano2, ano3, ano1+ano2+ano3 AS "Total"
03 | FROM (SELECT amount_sold, calendar_year,
04 |           dc.cust_id,
05 |           dc.cust_first_name, dc.cust_last_name
06 |       FROM dwsales ds JOIN dwtimes dt
07 |                       ON ds.time_id = dt.time_id
08 |                       JOIN dwcustomers dc
09 |                       ON ds.cust_id = dc.cust_id
10 |       WHERE calendar_year BETWEEN 1998 AND 2000)
11 | PIVOT (SUM(amount_sold) FOR calendar_year
12 |       IN (1998 ano1, 1999 ano2, 2000 ano3))
13 | ORDER BY cust_last_name, cust_first_name
```