CENTRE ESPECÍFIC
D´EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA

TEMA 6 DISEÑO FÍSICO. DQL

EJERCICIOS COMPLEMENTARIOS BD TEORÍA RESUELTOS

Bases de Datos
CFGS DAW

Pau Miñana Climent
2020/2021

EJERCICIOS COMPLEMENTARIOS CON LA BASE DE DATOS DE LA TEORÍA DE LA UNIDAD RESUELTOS

Mostrar el precio del producto más barato

**Se muestran dos opciones para encontrar max/min puesto que cobran relevancia más adelante*.

```
mysql> SELECT MIN(precio) AS "Precio Minimo" FROM productosped;
+---------------+
| Precio Minimo |
+---------------+
|             3 |
+---------------+
1 row in set (0.00 sec)
mysql> SELECT precio FROM productosped WHERE precio<=ALL (SELECT precio FROM productosped);
+--------+
| precio |
+--------+
|      3 |
+--------+
1 row in set (0.00 sec)
```

SELECT MIN(precio) AS "Precio Minimo" FROM productosped;

SELECT precio FROM productosped WHERE precio<=ALL (SELECT precio FROM productosped);

Mostrar el nombre del producto más barato

```
mysql> SELECT nombreproducto FROM productosped WHERE precio=(SELECT MIN(precio) FROM productosped);
+--------------------+
| nombreproducto     |
+--------------------+
| NAIPES PETER PARKER |
+--------------------+
1 row in set (0.00 sec)

mysql> SELECT nombreproducto FROM productosped WHERE precio<=ALL (SELECT precio FROM productosped);
+--------------------+
| nombreproducto     |
+--------------------+
| NAIPES PETER PARKER |
+--------------------+
1 row in set (0.00 sec)
```

SELECT nombreproducto FROM productosped WHERE precio =
(SELECT MIN(precio) FROM productosped);

SELECT nombreproducto FROM productosped WHERE precio <= ALL
(SELECT precio FROM productosped);

Mostrar el nombre y el precio del producto más barato

```
mysql> SELECT nombreproducto,precio FROM productosped WHERE precio=(SELECT MIN(precio) FROM productosped);
+--------------------+--------+
| nombreproducto     | precio |
+--------------------+--------+
| NAIPES PETER PARKER |      3 |
+--------------------+--------+
1 row in set (0.00 sec)

mysql> SELECT nombreproducto,precio FROM productosped WHERE precio<= ALL (SELECT precio FROM productosped);
+--------------------+--------+
| nombreproducto     | precio |
+--------------------+--------+
| NAIPES PETER PARKER |      3 |
+--------------------+--------+
1 row in set (0.00 sec)
```

SELECT nombreproducto,precio FROM productosped WHERE precio =
(SELECT MIN(precio) FROM productosped);

SELECT nombreproducto,precio FROM productosped WHERE precio <= ALL
(SELECT precio FROM productosped);

En las funciones agregadas el resultado es un "cálculo" que no se corresponde con ninguna tupla. Esto es evidente en funciones como sum/count pero en otras como min/max el resultado es igual a una de las filas y aún así tampoco se corresponde realmente a esta. Son funciones agregadas y no devuelven realmente una tupla concreta. Por eso en sql estándar no se puede recurrir a consultas que a primera vista pueden parecer correctas, como:

SELECT nombreproducto,min(precio) FROM productosped;

```
mysql> SELECT nombreproducto,min(precio) FROM productosped;
ERROR 1140 (42000): In aggregated query without GROUP BY, expression #1 of SELECT list contains nonaggregat
ed column 'teoriaud6.productosped.NombreProducto'; this is incompatible with sql_mode=only_full_group_by
```

Mostrar el nombre del responsable del proyecto Repsol, S.A.

```
mysql> SELECT e.nombre FROM empleados e,proyectos p WHERE dni=responsable AND p.nombre="Repsol, S.A.";
+-------------+
| nombre      |
+-------------+
| Alberto Gil |
+-------------+
1 row in set (0.00 sec)
```

SELECT e.nombre FROM empleados e,proyectos p
WHERE dni=responsable AND p.nombre="Repsol, S.A.";

Mostrar el nombre del responsable del proyecto Repsol, S.A. y la ubicación de su departamento.

```
mysql> SELECT e.nombre, d.ubicacion FROM empleados e, proyectos p, departamentos d
    -> WHERE dni=responsable AND p.nombre="Repsol, S.A." AND e.dpto=coddpto;
+-------------+------------------+
| nombre      | ubicacion        |
+-------------+------------------+
| Alberto Gil | Planta quinta U1 |
+-------------+------------------+
1 row in set (0.00 sec)
```

SELECT e.nombre, d.ubicacion FROM empleados e, proyectos p, departamentos d
WHERE dni=responsable AND p.nombre="Repsol, S.A." AND e.dpto=coddpto;

Mostrar todos los pedidos y el precio total de cada uno de ellos.

```
mysql> SELECT Numpedido,SUM(cantidad*precio) AS Total FROM productospedido a,productosped b
    -> WHERE a.refeproducto=b.refeproducto GROUP BY numpedido;
+-----------+--------------------+
| Numpedido | Total              |
+-----------+--------------------+
|         1 |                411 |
|         2 |              337.5 |
|         3 |  653.0000019073486 |
|         4 | 1218.9000091552734 |
|         5 |  525.0000038146973 |
+-----------+--------------------+
5 rows in set (0.00 sec)
```

SELECT Numpedido,SUM(cantidad*precio) AS Total FROM productospedido a,productosped b
WHERE a.refeproducto=b.refeproducto GROUP BY numpedido;

Mostrar los pedidos que superen los 600 euros y su precio.

```
mysql> SELECT Numpedido,SUM(cantidad*precio) AS Total FROM productospedido a,productosped b
    -> WHERE a.refeproducto=b.refeproducto GROUP BY numpedido HAVING sum(cantidad*precio)>600;
+-----------+--------------------+
| Numpedido | Total              |
+-----------+--------------------+
|         3 |  653.0000019073486 |
|         4 | 1218.9000091552734 |
+-----------+--------------------+
2 rows in set (0.00 sec)
```

SELECT Numpedido,SUM(cantidad*precio) AS Total FROM productospedido a,productosped b WHERE a.refeproducto=b.refeproducto GROUP BY numpedido HAVING sum(cantidad*precio)>600;

Mostrar el precio total del pedido más caro

```
mysql> SELECT MAX(Total2) as Total FROM
    -> (SELECT SUM(cantidad*precio) AS Total2 FROM productospedido a,productosped b
    ->  WHERE a.refeproducto = b.refeproducto GROUP BY numpedido) totalpedidos;
+--------------------+
| Total              |
+--------------------+
| 1218.9000091552734 |
+--------------------+
1 row in set (0.00 sec)

mysql> SELECT SUM(cantidad*precio) AS Total FROM productospedido a,productosped b
    -> WHERE a.refeproducto=b.refeproducto GROUP BY numpedido HAVING Total >= ALL
    -> (SELECT SUM(cantidad*precio) FROM productospedido a,productosped b
    ->  WHERE a.refeproducto = b.refeproducto GROUP BY numpedido);
+--------------------+
| Total              |
+--------------------+
| 1218.9000091552734 |
+--------------------+
1 row in set (0.00 sec)
```

SELECT MAX(Total2) as Total FROM
(SELECT SUM(cantidad*precio) AS Total2 FROM productospedido a,productosped b
WHERE a.refeproducto = b.refeproducto GROUP BY numpedido) totalpedidos;

SELECT SUM(cantidad*precio) AS Total FROM productospedido a,productosped b
WHERE a.refeproducto=b.refeproducto GROUP BY numpedido HAVING Total >= ALL
(SELECT SUM(cantidad*precio) FROM productospedido a,productosped b
WHERE a.refeproducto = b.refeproducto GROUP BY numpedido);

Se puede observar que la primera propuesta es más simple, pero con las limitaciones anteriormente comentadas de las funciones agregadas el máximo no se corresponde con su tupla y, por tanto, para mostrar más datos correspondientes a esta "tupla máxima" es necesario replantear la consulta y la complejidad se iguala. Esto quedará patente en las próximas consultas que se realizan.

Por otro lado, con la segunda opción se selecciona directamente la tupla con el máximo valor, así pues, sólo hay que añadir los campos deseados y acaba resultando quizás una aproximación más fácil de comprender.

Aún así ambas opciones son perfectamente válida y usar una u otra se trata de una elección puramente personal.

Mostrar el pedido más caro y su precio

```
mysql> SELECT Numpedido,SUM(cantidad*precio) AS Total FROM productospedido a,productosped b
    -> WHERE a.refeproducto=b.refeproducto GROUP BY numpedido HAVING Total =
    ->    (SELECT MAX(Total2) FROM
    ->         (SELECT SUM(cantidad*precio) AS Total2 FROM productospedido a,productosped b
    ->          WHERE a.refeproducto=b.refeproducto GROUP BY numpedido) totalpedidos);
+-----------+--------------------+
| Numpedido | Total              |
+-----------+--------------------+
|         4 | 1218.9000091552734 |
+-----------+--------------------+
1 row in set (0.00 sec)

mysql> SELECT Numpedido,SUM(cantidad*precio) AS Total FROM productospedido a,productosped b
    -> WHERE a.refeproducto=b.refeproducto GROUP BY numpedido HAVING Total >= ALL
    ->    (SELECT SUM(cantidad*precio) FROM productospedido a,productosped b
    ->     WHERE a.refeproducto =b.refeproducto GROUP BY numpedido);
+-----------+--------------------+
| Numpedido | Total              |
+-----------+--------------------+
|         4 | 1218.9000091552734 |
+-----------+--------------------+
1 row in set (0.00 sec)
```

SELECT Numpedido,SUM(cantidad*precio) AS Total FROM productospedido a,productosped b
WHERE a.refeproducto=b.refeproducto GROUP BY numpedido HAVING Total =
(SELECT MAX(Total2) FROM
(SELECT SUM(cantidad*precio) AS Total2 FROM productospedido a,productosped b
WHERE a.refeproducto=b.refeproducto GROUP BY numpedido) totalpedidos);

SELECT Numpedido,SUM(cantidad*precio) AS Total FROM productospedido a,productosped b
WHERE a.refeproducto=b.refeproducto GROUP BY numpedido HAVING Total >= ALL
(SELECT SUM(cantidad*precio) FROM productospedido a,productosped b
WHERE a.refeproducto =b.refeproducto GROUP BY numpedido);

Mostrar el pedido más caro,su precio y el nombre de su proveedor

```
mysql> SELECT a.Numpedido, nombreproveedor, SUM(cantidad*precio) AS Total
    -> FROM productospedido a, productosped b, pedidos, proveedores
    -> WHERE a.refeproducto = b.refeproducto AND a.numpedido = pedidos.numpedido
    -> AND proveedor = codproveedor GROUP BY numpedido HAVING Total =
    ->               (SELECT MAX(Total) FROM
    ->               (SELECT SUM(cantidad*precio) AS Total FROM productospedido a, productosped b
    ->                WHERE a.refeproducto = b.refeproducto GROUP BY numpedido) totalpedidos);
+-----------+-----------------+--------------------+
| Numpedido | nombreproveedor | Total              |
+-----------+-----------------+--------------------+
|         4 | JUGUETOS, S.A.  | 1218.9000091552734 |
+-----------+-----------------+--------------------+
1 row in set (0.00 sec)

mysql> SELECT a.Numpedido, nombreproveedor, SUM(cantidad*precio) AS Total
    -> FROM productospedido a, productosped b, pedidos, proveedores
    -> WHERE a.refeproducto = b.refeproducto AND a.numpedido = pedidos.numpedido
    -> AND proveedor = codproveedor GROUP BY numpedido HAVING Total >= ALL
    ->               (SELECT SUM(cantidad*precio) FROM productospedido a, productosped b
    ->                WHERE a.refeproducto = b.refeproducto GROUP BY numpedido);
+-----------+-----------------+--------------------+
| Numpedido | nombreproveedor | Total              |
+-----------+-----------------+--------------------+
|         4 | JUGUETOS, S.A.  | 1218.9000091552734 |
+-----------+-----------------+--------------------+
1 row in set (0.00 sec)
```

```
SELECT a.Numpedido, nombreproveedor, SUM(cantidad*precio) AS Total
FROM productospedido a, productosped b, pedidos, proveedores
WHERE a.refeproducto = b.refeproducto AND a.numpedido = pedidos.numpedido
AND proveedor = codproveedor GROUP BY numpedido HAVING Total =
        (SELECT MAX(Total) FROM
        (SELECT SUM(cantidad*precio) AS Total FROM productospedido a, productosped b
         WHERE a.refeproducto = b.refeproducto GROUP BY numpedido) totalpedidos);
```

```
SELECT a.Numpedido, nombreproveedor, SUM(cantidad*precio) AS Total
FROM productospedido a, productosped b, pedidos, proveedores
WHERE a.refeproducto = b.refeproducto AND a.numpedido = pedidos.numpedido
AND proveedor = codproveedor GROUP BY numpedido HAVING Total >= ALL
        (SELECT SUM(cantidad*precio) FROM productospedido a, productosped b
         WHERE a.refeproducto = b.refeproducto GROUP BY numpedido);
```

VISTAS

Como se ha podido observar, el uso de subconsultas puede llegar a extender bastante las instrucciones, dificultando su manejo y organización, sobre todo en entornos no gráficos como las consolas de comandos. Además es posible que algunas de estas subconsultas se vayan a usar en repetidas ocasiones. En estos casos el uso de las vistas puede ayudar a simplificar las expresiones y agilizar el proceso.

Se usa la última consulta, opción 2, como ejemplo práctico y se decide crear una vista con los datos agrupados de cada pedido incluyendo precio total, datos del proveedor y fecha del pedido, llamado facturas para poder consultarlo. Se deja fuera los productos puesto que evitan poder agrupar los pedidos en una línea y tener el precio total pero se incluyen otros datos que no se usan en esta consulta para generalizar y permitir otros usos de la vista.

```
mysql> CREATE VIEW facturas AS SELECT a.Numpedido,codproveedor, nombreproveedor, codpostal, fecha,
    -> SUM(cantidad*precio) AS Total  FROM productospedido a, productosped b, pedidos, proveedores
    -> WHERE a.refeproducto = b.refeproducto AND a.numpedido = pedidos.numpedido
    -> AND proveedor = codproveedor
    -> GROUP BY numpedido;
Query OK, 0 rows affected (0.28 sec)

mysql> select * from facturas;
+-----------+-------------+--------------------+-----------+------------+--------------------+
| Numpedido | codproveedor | nombreproveedor    | codpostal | fecha      | Total              |
+-----------+-------------+--------------------+-----------+------------+--------------------+
|         1 | TO342       | JUGUETOS, S.A.     | 45600     | 2013-06-10 |                411 |
|         2 | MA280       | TOYPLAY, S.A.      | 28005     | 2013-06-10 |              337.5 |
|         3 | BA843       | CARMELO DIAZ, S.L. | 06004     | 2013-06-12 |  653.0000019073486 |
|         4 | TO342       | JUGUETOS, S.A.     | 45600     | 2013-06-14 | 1218.9000091552734 |
|         5 | MA280       | TOYPLAY, S.A.      | 28005     | 2013-06-14 |  525.0000038146973 |
+-----------+-------------+--------------------+-----------+------------+--------------------+
5 rows in set (0.00 sec)
```

```
CREATE VIEW facturas AS SELECT a.Numpedido, codproveedor, nombreproveedor, codpostal, fecha,
SUM(cantidad*precio) AS Total  FROM productospedido a, productosped b, pedidos, proveedores
WHERE a.refeproducto = b.refeproducto AND a.numpedido = pedidos.numpedido
AND proveedor = codproveedor
GROUP BY numpedido;
```

Con esto la consulta queda simplificada a:

```
mysql> SELECT numpedido, nombreproveedor, Total
    -> FROM facturas
    -> WHERE Total>= ALL (Select Total from facturas);
+-----------+-----------------+--------------------+
| numpedido | nombreproveedor | Total              |
+-----------+-----------------+--------------------+
|         4 | JUGUETOS, S.A.  | 1218.9000091552734 |
+-----------+-----------------+--------------------+
1 row in set (0.00 sec)
```

SELECT numpedido, nombreproveedor, Total
FROM facturas
WHERE Total>= ALL (Select Total from facturas);

Cabe recordar que las vistas sólo almacenan la instrucción de la consulta y no los datos de la "tabla" creada, así pues, no hacen que las consultas se ejecuten más rápido,pero tampoco gastan un espacio de almacenamiento significativo al no contener realmente los datos.
Que no os confunda el hecho de que, por ejemplo, en mysql se muestren en la lista de tablas; si intentamos eliminarlas se puede observar como la tabla en realidad no existe.

```
mysql> show tables;
+---------------------+
| Tables_in_teoriaud6 |
+---------------------+
| dep_inf             |
| departamentos       |
| empleados           |
| empleados2          |
| facturas            |
| pedidos             |
| productosped        |
| productospedido     |
| proveedores         |
| proyectos           |
+---------------------+
10 rows in set (0.01 sec)

mysql> drop table facturas;
ERROR 1051 (42S02): Unknown table 'teoriaud6.facturas'
```

Licencia