

UNIDAD 6

MODELO FÍSICO DQL

BASES DE DATOS 22/23
CFGS DAW

BOLETÍN DQL. NIVELES BÁSICO, MEDIO Y AVANZADO

SOLUCIONES NIVEL AVANZADO

Revisado por:

Sergio Badal, Abelardo Martínez y Pau Miñana

Autores:

Raquel Torres

Paco Aldarias

Fecha: 16/02/23

Licencia Creative Commons



Reconocimiento - NoComercial - CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

ÍNDICE DE CONTENIDO

1. SOLUCIÓN Consulta 31.....	3
2. SOLUCIÓN Consulta 32.....	5
3. SOLUCIÓN Consulta 33.....	6
4. SOLUCIÓN Consulta 34.....	7
5. SOLUCIÓN Consulta 35.....	8
6. SOLUCIÓN Consulta 36.....	9
7. SOLUCIÓN Consulta 37.....	9
8. SOLUCIÓN Consulta 38.....	10

UD06. BOLETÍN MODELO FÍSICO DQL

1. SOLUCIÓN CONSULTA 31

Mostrar el nombre de todos aquellos clientes que hayan realizado al menos un pedido ordenado alfabéticamente.

OPCIÓN 1

Predicado EXISTS

```
mysql> SELECT C.NombreCliente AS 'Nombre Cliente'
-> FROM Clientes C
-> WHERE EXISTS
-> (SELECT CodigoPedido
-> FROM Pedidos P
-> WHERE P.CodigoCliente = C.CodigoCliente)
-> ORDER BY C.NombreCliente;

+-----+
| Nombre Cliente |
+-----+
| Agrojardin     |
| Beragua        |
| Camunas Jardines S.L. |
| Dardena S.A.   |
| DGPRODUCTIONS GARDEN |
| El Jardin Viviente S.L |
| Flores Marivi  |
| FLORES S.L.    |
| Gardening Associates |
| Gerudo Valley  |
| Golf S.A.      |
| Jardin de Flores |
| Jardineria Sara |
| Jardinerías Matías SL |
| Jardines y Mansiones CACTUS SL |
| Naturagua      |
| Sotogrande     |
| Tendo Garden   |
| Tutifruti S.A  |
+-----+
19 rows in set (0,00 sec)
```

OPCIÓN 2

Predicado IN

```
mysql> SELECT C.NombreCliente AS 'Nombre Cliente'
-> FROM Clientes C
-> WHERE C.CodigoCliente IN
-> (SELECT P.CodigoCliente
-> FROM Pedidos P
-> WHERE P.CodigoCliente = C.CodigoCliente)
-> ORDER BY C.NombreCliente;
+-----+
| Nombre Cliente |
+-----+
| Agrojardin     |
| Beragua        |
| Camunas Jardines S.L. |
| Dardena S.A.   |
| DGPRODUCTIONS GARDEN |
| El Jardin Viviente S.L |
| Flores Marivi  |
| FLORES S.L.    |
| Gardening Associates |
| Gerudo Valley  |
| Golf S.A.      |
| Jardin de Flores |
| Jardineria Sara |
| Jardinerías Matías SL |
| Jardines y Mansiones CACTUS SL |
| Naturagua      |
| Sotogrande     |
| Tendo Garden   |
| Tutifruti S.A. |
+-----+
19 rows in set (0,01 sec)
```

OPCIÓN 3

Utilizar GROUP BY

```
mysql> SELECT C.NombreCliente AS 'Nombre Cliente'
-> FROM Clientes C, Pedidos P
-> WHERE P.CodigoCliente = C.CodigoCliente
-> GROUP BY C.NombreCliente
-> ORDER BY C.NombreCliente;
+-----+
| Nombre Cliente |
+-----+
| Agrojardin     |
| Beragua        |
| Camunas Jardines S.L. |
| Dardena S.A.   |
| DGPRODUCTIONS GARDEN |
| El Jardin Viviente S.L |
| Flores Marivi  |
| FLORES S.L.    |
| Gardening Associates |
| Gerudo Valley  |
| Golf S.A.      |
| Jardin de Flores |
| Jardineria Sara |
| Jardinerías Matías SL |
| Jardines y Mansiones CACTUS SL |
| Naturagua      |
| Sotogrande     |
| Tendo Garden   |
| Tutifruti S.A. |
+-----+
19 rows in set (0,00 sec)
```

2. SOLUCIÓN CONSULTA 32

Mostrar todos los pedidos del cliente “Beragua” con el importe total de cada pedido, ordenado por el número de pedido.

OPCIÓN 1

Subconsulta en el SELECT

```
mysql> SELECT P.CodigoPedido,
-> (SELECT SUM(D.Cantidad*D.PrecioUnidad)
-> FROM DetallePedidos D
-> WHERE D.CodigoPedido = P.CodigoPedido) AS Importe_Total
-> FROM Pedidos P, Clientes C
-> WHERE P.CodigoCliente = C.CodigoCliente
-> AND C.NombreCliente = 'Beragua'
-> ORDER BY P.CodigoPedido;
+-----+-----+
| CodigoPedido | Importe_Total |
+-----+-----+
|          13 |         738.00 |
|          14 |         829.00 |
|          15 |         214.00 |
|          16 |         234.00 |
|          17 |         375.00 |
+-----+-----+
5 rows in set (0,00 sec)
```

OPCIÓN 2

Utilizar GROUP BY

```
mysql> SELECT P.CodigoPedido, SUM(D.Cantidad*D.PrecioUnidad) AS Importe_Total
-> FROM DetallePedidos D, Pedidos P, Clientes C
-> WHERE D.CodigoPedido = P.CodigoPedido
-> AND P.CodigoCliente = C.CodigoCliente
-> AND C.NombreCliente = 'Beragua'
-> GROUP BY P.CodigoPedido
-> ORDER BY P.CodigoPedido;
+-----+-----+
| CodigoPedido | Importe_Total |
+-----+-----+
|          13 |         738.00 |
|          14 |         829.00 |
|          15 |         214.00 |
|          16 |         234.00 |
|          17 |         375.00 |
+-----+-----+
5 rows in set (0,00 sec)
```

3. SOLUCIÓN CONSULTA 33

Mostrar los pedidos con número comprendido entre 100 y 110, con el importe total de cada uno de ellos, ordenado por el número del pedido.

```
mysql> SELECT P.CodigoPedido,
-> (SELECT SUM(D.Cantidad*D.PrecioUnidad)
-> FROM DetallePedidos D
-> WHERE D.CodigoPedido = P.CodigoPedido) AS Importe_Total
-> FROM Pedidos P
-> WHERE P.CodigoPedido BETWEEN 100 AND 110
-> ORDER BY P.CodigoPedido;
```

CodigoPedido	Importe_Total
100	800.00
101	209.00
102	660.00
103	304.00
104	1760.00
105	1506.00
106	1077.00
107	3216.00
108	660.00
109	553.00
110	149.00

11 rows in set (0,01 sec)

4. SOLUCIÓN CONSULTA 34

Calcular el importe máximo de un pedido y el importe mínimo de un pedido de todos los pedidos realizados por los clientes.

```
mysql> SELECT MAX(Total) AS Importe_maximo, MIN(Total) AS Importe_minimo
-> FROM (SELECT SUM(Cantidad*PrecioUnidad) AS Total
-> FROM DetallePedidos
-> GROUP BY CodigoPedido) Totales_Pedidos;
```

Importe_maximo	Importe_minimo
73226.00	4.00

1 row in set (0,00 sec)

En este ejercicio hemos utilizado una tabla derivada a partir de la cual hemos calculado el máximo y el mínimo.

¿Y si queremos que aparezca el código del pedido?

Entonces necesitamos realizar una consulta más compleja.

```
mysql> SELECT DP1.CodigoPedido, SUM(DP1.PrecioUnidad * DP1.Cantidad) AS Importe_maximo_y_minimo
-> FROM DetallePedidos DP1
-> GROUP BY DP1.CodigoPedido
-> HAVING SUM(DP1.PrecioUnidad * DP1.Cantidad) >= ALL
-> (SELECT SUM(DP2.PrecioUnidad * DP2.Cantidad)
-> FROM DetallePedidos DP2
-> WHERE DP2.CodigoPedido <> DP1.CodigoPedido
-> GROUP BY DP2.CodigoPedido)
-> UNION ALL
-> SELECT DP1.CodigoPedido, SUM(DP1.PrecioUnidad * DP1.Cantidad) AS Importe_maximo_y_minimo
-> FROM DetallePedidos DP1
-> GROUP BY DP1.CodigoPedido
-> HAVING SUM(DP1.PrecioUnidad * DP1.Cantidad) <= ALL
-> (SELECT SUM(DP2.PrecioUnidad * DP2.Cantidad)
-> FROM DetallePedidos DP2
-> WHERE DP2.CodigoPedido <> DP1.CodigoPedido
-> GROUP BY DP2.CodigoPedido);
+-----+-----+
| CodigoPedido | Importe_maximo_y_minimo |
+-----+-----+
|          33 |          73226.00 |
|          42 |           4.00 |
+-----+-----+
2 rows in set (0,00 sec)
```

5. SOLUCIÓN CONSULTA 35

Mostrar el nombre del producto y el total de unidades pedidas, de los productos de los cuales se hayan pedido más de 450 unidades ordenados de mayor a menor por el número de unidades.

```
mysql> SELECT P.Nombre, SUM(D.Cantidad) AS Total_unidades
-> FROM Productos P, DetallePedidos D
-> WHERE P.CodigoProducto = D.CodigoProducto
-> GROUP BY P.Nombre
-> HAVING SUM(D.Cantidad) > 450
-> ORDER BY SUM(D.Cantidad) DESC;
+-----+-----+
| Nombre | Total_unidades |
+-----+-----+
| Thymus Vulgaris |          961 |
| Thymus Citriodra (Tomillo limón) |          455 |
+-----+-----+
2 rows in set (0,00 sec)
```


6. SOLUCIÓN CONSULTA 36

Mostrar el nombre del cliente y la suma total del importe de todos los pedidos realizados por él, ordenado por el nombre del cliente.

```
mysql> SELECT C.NombreCliente AS Nombre_Cliente, SUM(D.Cantidad*D.PrecioUnidad) AS Importe_Total_Pedidos
-> FROM Clientes C, Pedidos P, DetallePedidos D
-> WHERE C.CodigoCliente = P.CodigoCliente
-> AND P.CodigoPedido = D.CodigoPedido
-> GROUP BY C.NombreCliente
-> ORDER BY C.NombreCliente;
```

Nombre_Cliente	Importe_Total_Pedidos
Agrojardin	8489.00
Beragua	2390.00
Camunas Jardines S.L.	2246.00
Dardena S.A.	4160.00
DGPRODUCTIONS GARDEN	6165.00
El Jardin Viviente S.L	1171.00
Flores Marivi	4399.00
Gardening Associates	10926.00
Gerudo Valley	81849.00
Golf S.A.	232.00
Jardin de Flores	12081.00
Jardineria Sara	7863.00
Jardinerías Matías SL	10972.00
Jardines y Mansiones CACTUS SL	18279.00
Naturagua	929.00
Sotogrande	272.00
Tendo Garden	23794.00
Tutifrutí S.A	3321.00

18 rows in set (0,00 sec)

7. SOLUCIÓN CONSULTA 37

Mostrar el nombre del producto y el precio de venta del producto más caro que tengamos.

OPCIÓN 1

Subconsulta en WHERE

```
mysql> SELECT PR1.Nombre, PR1.PrecioVenta
-> FROM Productos PR1
-> WHERE PR1.PrecioVenta = (SELECT MAX(PR2.PrecioVenta)
-> FROM Productos PR2);
```

Nombre	PrecioVenta
Trachycarpus Fortunei	462.00

1 row in set (0,00 sec)

OPCIÓN 2

Predicado ALL

```
mysql> SELECT PR1.Nombre, PR1.PrecioVenta
-> FROM Productos PR1
-> WHERE PR1.PrecioVenta >= ALL
->       (SELECT PR2.PrecioVenta
->        FROM Productos PR2);
+-----+-----+
| Nombre          | PrecioVenta |
+-----+-----+
| Trachycarpus Fortunei |      462.00 |
+-----+-----+
1 row in set (0,01 sec)
```

OPCIÓN 3

Tabla derivada

```
mysql> SELECT PR1.Nombre, PR1.PrecioVenta
-> FROM Productos PR1, (SELECT MAX(PR2.PrecioVenta) AS MaxPrecio
->                      FROM Productos PR2) TD_Maximoprecio
-> WHERE PR1.PrecioVenta = TD_Maximoprecio.MaxPrecio;
+-----+-----+
| Nombre          | PrecioVenta |
+-----+-----+
| Trachycarpus Fortunei |      462.00 |
+-----+-----+
1 row in set (0,01 sec)
```

8. SOLUCIÓN CONSULTA 38

Mostrar el nombre del cliente, el número de pedido, la base imponible del pedido, el importe del IVA (21%) y el total del pedido, para los pedidos 100, 103, 106 y 109.

OPCIÓN 1

Agrupando por cabecera de pedido.

```
mysql> SELECT C.NombreCliente, P.CodigoPedido,
-> SUM(D.Cantidad*D.PrecioUnidad) AS Base_Imponible,
-> SUM(D.Cantidad*D.PrecioUnidad)*0.21 AS Importe_IVA,
-> SUM(D.Cantidad*D.PrecioUnidad)*1.21 AS Importe_Total
-> FROM DetallePedidos D, Clientes C, Pedidos P
-> WHERE C.CodigoCliente = P.CodigoCliente
-> AND P.CodigoPedido = D.CodigoPedido
-> AND P.CodigoPedido IN (100, 103, 106, 109)
-> GROUP BY P.CodigoPedido, C.NombreCliente
-> ORDER BY C.NombreCliente, P.CodigoPedido;
```

NombreCliente	CodigoPedido	Base_Imponible	Importe_IVA	Importe_Total
El Jardin Viviente S.L	109	553.00	116.1300	669.1300
Flores Marivi	100	800.00	168.0000	968.0000
Jardineria Sara	103	304.00	63.8400	367.8400
Jardineria Sara	106	1077.00	226.1700	1303.1700

4 rows in set (0,00 sec)

OPCIÓN 2

Agrupando por detalle de pedido.

```
mysql> SELECT C.NombreCliente, D.CodigoPedido,
-> SUM(D.Cantidad*D.PrecioUnidad) AS Base_Imponible,
-> SUM(D.Cantidad*D.PrecioUnidad)*0.21 AS Importe_IVA,
-> SUM(D.Cantidad*D.PrecioUnidad)*1.21 AS Importe_Total
-> FROM DetallePedidos D, Clientes C, Pedidos P
-> WHERE C.CodigoCliente = P.CodigoCliente
-> AND P.CodigoPedido = D.CodigoPedido
-> AND P.CodigoPedido IN (100, 103, 106, 109)
-> GROUP BY D.CodigoPedido, C.NombreCliente
-> ORDER BY C.NombreCliente, D.CodigoPedido;
```

NombreCliente	CodigoPedido	Base_Imponible	Importe_IVA	Importe_Total
El Jardin Viviente S.L	109	553.00	116.1300	669.1300
Flores Marivi	100	800.00	168.0000	968.0000
Jardineria Sara	103	304.00	63.8400	367.8400
Jardineria Sara	106	1077.00	226.1700	1303.1700

4 rows in set (0,00 sec)