

# UNIT 5

## PHYSICAL MODELING DATA QUERY LANGUAGE (DQL)

BASES DE DATOS 2022/2023  
CFGs DAW

### FULL DQL WORKSHOP (SELECTS) SOLUTIONS WORKSHOP C (ADVANCED LEVEL)

**Reviewed by:**

Sergio Badal

**Author:**

Paco Aldarias

Date: 03/14/23

License Creative Commons



**Acknowledgment - NonCommercial - ShareAlike (by-nc-sa):** A commercial use of the original work or possible derivative works is not allowed, the distribution of which must be done with a license equal to that which regulates the original work.

- SOLUTION QUERY 31

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

```
mysql> SELECT C.NOMBRECLIENTE
-> FROM CLIENTES C
-> WHERE EXISTS
-> <SELECT CODIGOPEDIDO
-> FROM PEDIDOS P
-> WHERE P.CODIGOCLIENTE = C.CODIGOCLIENTE>
-> ORDER BY C.NOMBRECLIENTE;
+-----+
| NOMBRECLIENTE |
+-----+
| Agrojardin    |
| Beragua       |
| Camunas Jardines S.L. |
| Dardena S.A.  |
| DGPRODUCTIONS GARDEN |
| El Jardin Viviente S.L |
| Flores Mariivi |
| 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 (0.02 sec)
```

**ALSO VALID:**

```
select c.nombrecliente
from clientes c inner join pedidos p
on c.codigocliente = p.codigocliente
group by c.codigocliente
having count(*)>=1
order by c.nombrecliente;
```

- SOLUTION QUERY 32

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

```
mysql> SELECT P.CODIGOPEDIDO,  
-> (SELECT SUM(CANTIDAD * PRECIOUNIDAD)  
-> FROM DETALLEPEDIDOS D  
-> WHERE D.CODIGOPEDIDO = P.CODIGOPEDIDO) AS TOTAL  
-> FROM PEDIDOS P, CLIENTES C  
-> WHERE P.CODIGOCLIENTE = C.CODIGOCLIENTE  
-> AND C.NOMBRECLIENTE = 'Beragua'  
-> ORDER BY P.CODIGOPEDIDO;  
  
+-----+-----+  
| CODIGOPEDIDO | TOTAL |  
+-----+-----+  
| 13 | 738.00 |  
| 14 | 829.00 |  
| 15 | 214.00 |  
| 16 | 234.00 |  
| 17 | 375.00 |  
+-----+-----+  
5 rows in set (0.00 sec)
```

**ALSO VALID:**

```
select p.codigopedido, sum(dp.cantidad*dp.preciounidad) as importe  
from clientes c inner join pedidos p inner join detallepedidos dp  
on c.codigocliente = p.codigocliente and p.codigopedido = dp.codigopedido  
where c.nombrecliente = "Beragua"  
group by p.codigopedido  
order by 1;
```

- SOLUTION QUERY 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(CANTIDAD * PRECIOUNIDAD)  
-> FROM DETALLEPEDIDOS D  
-> WHERE D.CODIGOPEDIDO = P.CODIGOPEDIDO> AS TOTAL  
-> FROM PEDIDOS P  
-> WHERE P.CODIGOPEDIDO BETWEEN 100 AND 110  
-> ORDER BY P.CODIGOPEDIDO;  
+-----+-----+  
: CODIGOPEDIDO : 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 (0.00 sec)
```

**ALSO VALID:**

```
select p.codigopedido, sum(dp.cantidad*dp.preciounidad) as importe  
from clientes c inner join pedidos p inner join detallepedidos dp  
on c.codigocliente = p.codigocliente and p.codigopedido = dp.codigopedido  
where p.codigopedido between 100 and 110  
group by p.codigopedido  
order by 1;
```

- SOLUTION QUERY 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.

#### ALSO VALID:

##### -- OPCION A (CON VISTAS)

```
create view vi_pedidos as
```

```
(SELECT dp.codigopedido, sum(dp.cantidad*dp.preciounidad) AS importe
```

```
FROM detallepedidos dp
```

```
GROUP BY dp.codigopedido);
```

```
select max(importe) AS MAXIMO, min(importe) AS MINIMO
```

```
from vi_pedidos TOTAL_PEDIDOS;
```

```
DROP VIEW vi_pedidos;
```

##### -- OPCION B(SIN VISTAS)

```
select max(importe) AS MAXIMO, min(importe) AS MINIMO
```

```
from
```

```
(SELECT dp.codigopedido, sum(dp.cantidad*dp.preciounidad) as importe
```

```
FROM detallepedidos dp
```

```
GROUP BY dp.codigopedido) TOTAL_PEDIDOS;
```

- SOLUTION QUERY 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.CODIGOPRODUCTO
-> HAVING SUM(D.CANTIDAD) > 450
-> ORDER BY TOTAL_UNIDADES DESC;

+-----+-----+
| NOMBRE                                | TOTAL_UNIDADES |
+-----+-----+
| Thymus Vulgaris                       | 961            |
| Thymus Citriodora (Tomillo limón)     | 455            |
+-----+-----+
2 rows in set (0.00 sec)
```

**ALSO VALID:**

```
select p.nombre as "producto", sum(dp.cantidad) as "total unidades"
```

```
from productos p inner join detallepedidos dp
```

```
on p.codigoproducto = dp.codigoproducto
```

```
group by dp.codigoproducto
```

```
having sum(dp.cantidad)>450
```

```
order by 2 desc;
```

- SOLUTION QUERY 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,
-> SUM(D.CANTIDAD * D.PRECIOUNIDAD) AS 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;
```

NOMBRECLIENTE	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 Mariivi	4399.00
Gardening Associates	10926.00
Gerudo Valley	81849.00
Golf S.A.	232.00
Jardin de Flores	12081.00
Jardinerías Matías SL	10972.00
Jardineria Sara	7863.00
Jardines y Mansiones CACTUS SL	18279.00
Naturagua	929.00
Sotogrande	272.00
Tendo Garden	23794.00
Tutifruti S.A	3321.00

18 rows in set (0.00 sec)

**ALSO VALID:**

```
select c.nombrecliente, sum(dp.cantidad*dp.preciounidad) as importe
from clientes c inner join pedidos p inner join detallepedidos dp
on c.codigocliente = p.codigocliente and p.codigopedido = dp.codigopedido
group by p.codigocliente
order by c.nombrecliente;
```

- SOLUTION QUERY 37

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

```
mysql> SELECT NOMBRE, PRECIOVENTA
-> FROM PRODUCTOS
-> WHERE PRECIOVENTA = (SELECT MAX(PRECIOVENTA)
->                        FROM PRODUCTOS);
+-----+-----+
| NOMBRE                | PRECIOVENTA |
+-----+-----+
| Trachycarpus Fortunei |      462.00 |
+-----+-----+
1 row in set (0.00 sec)
```

**ALSO VALID:**

```
select p1.nombre as "producto", p1.precioventa as "precio venta"
from productos p1
where not exists
(select p2.CodigoProducto from productos p2 where
p1.CodigoProducto<>p2.codigoproducto
and p2.precioventa>p1.precioventa);
```



- SOLUTION QUERY 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.

```
mysql> SELECT C.NOMBRECLIENTE, D.CODIGOPEDIDO,
-> SUM(D.CANTIDAD * D.PRECIOUNIDAD) AS BASE,
-> SUM(D.CANTIDAD * D.PRECIOUNIDAD) * 0.21 AS IVA,
-> SUM(D.CANTIDAD * D.PRECIOUNIDAD) * 1.21 AS 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 C.NOMBRECLIENTE, D.CODIGOPEDIDO
-> ORDER BY C.NOMBRECLIENTE, D.CODIGOPEDIDO;
```

NOMBRECLIENTE	CODIGOPEDIDO	BASE	IVA	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)

**ALSO VALID:**

```
select c.nombrecliente as "cliente", dp.codigopedido as "número de pedido",
sum(dp.cantidad*dp.preciounidad) as "base imponible",
sum(dp.cantidad*dp.preciounidad)*0.21 as "IVA",
(sum(dp.cantidad*dp.preciounidad)+(sum(dp.cantidad*dp.preciounidad)*0.21)) as
"total"
from clientes c inner join pedidos p inner join detallepedidos dp
on c.codigocliente = p.codigocliente and p.codigopedido = dp.codigopedido
where dp.codigopedido in (100,103,106, 109)
group by dp.codigopedido;
```