

# UNIDAD 6

## MODELO FÍSICO DQL

**BASES DE DATOS 22/23**  
CFGS DAW

## BOLETÍN DQL. NIVELES BÁSICO, MEDIO Y AVANZADO

**Revisado por:**

Sergio Badal, Abelardo Martínez y Pau Miñana

**Autores:**

Raquel Torres

Paco Aldarias

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

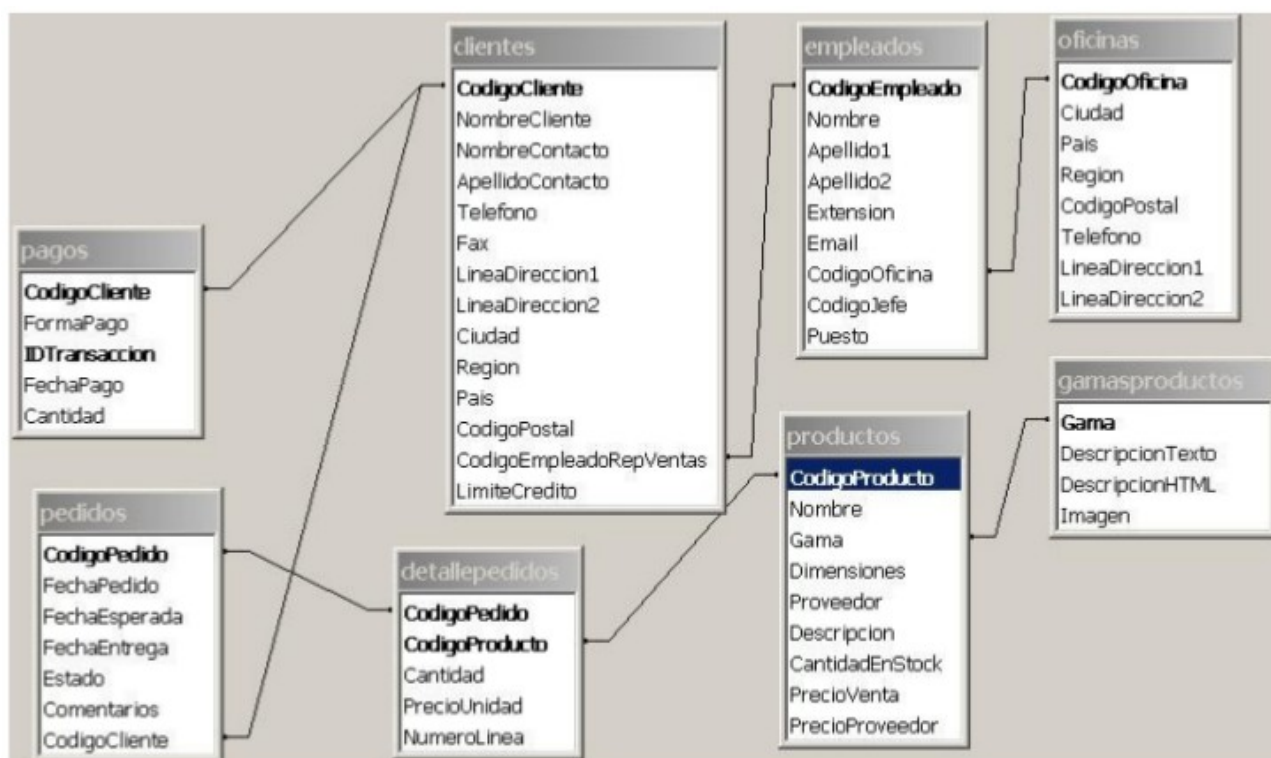
1. Consulta 11.....	5
2. Consulta 12.....	5
3. Consulta 13.....	5
4. Consulta 14.....	5
5. Consulta 15.....	5
6. Consulta 16.....	5
7. Consulta 17.....	5
8. Consulta 18.....	6
9. Consulta 19.1.....	6
10. Consulta 19.2.....	6
1. Consulta 21.....	7
2. Consulta 22.....	7
3. Consulta 23.....	7
4. Consulta 24.....	7
5. Consulta 25.....	7
6. Consulta 26.....	7
7. Consulta 27.....	7
8. Consulta 28.....	7
9. Consulta 29.1.....	8
10. Consulta 29.2.....	8
11. Consulta 29.3.....	8
12. Consulta 29.4.....	8
13. Consulta 29.5.....	8
1. Consulta 31.....	9
2. Consulta 32.....	9
3. Consulta 33.....	9
4. Consulta 34.....	9
5. Consulta 35.....	9
6. Consulta 36.....	9
7. Consulta 37.....	9
8. Consulta 38.....	9

## UD06. BOLETÍN MODELO FÍSICO DQL

Para realizar la práctica podemos utilizar cualquiera de los sistemas de bases de datos: MySQL, Oracle y PostgreSQL. Se puede hacer con uno y si se dispone de más tiempo hacerlo con el resto.

### ESQUEMA DE LA BBDD

Disponemos del siguiente esquema de la BD o diseño físico, en el que se muestran las tablas que lo forman y cómo están relacionadas entre sí:



Dedícale unos minutos a revisar los nombres de las tablas, los campos que las forman y cómo están relacionadas entre ellas. Es fundamental conocer las tablas para realizar después las consultas de forma adecuada.

El siguiente paso será crear las bases de datos correspondientes. Recordemos cómo se hacía.

Para MySQL haremos lo siguiente:

```
mysql> CREATE DATABASE jardineria CHARACTER SET utf8 COLLATE utf8_spanish_ci;
Query OK, 1 row affected (0.00 sec)

mysql> use jardineria;
Database changed
```

El archivo que contiene el script con la base de datos lo puedes descargar de la plataforma. Lo ejecutaremos y comprobaremos que se han creado las tablas correctamente.

```
mysql> show tables;
+-----+
| Tables_in_jardineria |
+-----+
| clientes              |
| detallepedidos        |
| empleados             |
| gamasproductos        |
| oficinas              |
| pagos                 |
| pedidos               |
| productos             |
+-----+
8 rows in set (0.00 sec)
```

Para Oracle haremos lo siguiente:

```
SQL> conn as sysdba
Enter user-name: sys
Enter password:
Connected.
SQL> show user;
USER is "SYS"
SQL> create user jardineria identified by curso;
User created.
SQL> grant connect, resource to jardineria;
Grant succeeded.
SQL> connect jardineria
Enter password:
Connected.
SQL> show user;
USER is "JARDINERIA"
SQL>
```

De la misma manera, el archivo que contiene el script con la base de datos lo puedes descargar de la plataforma, lo ejecutaremos y comprobaremos que se han creado las tablas correctamente.

```
SQL> select table_name from user_tables;
TABLE_NAME
-----
OFICINAS
EMPLEADOS
GAMASPRODUCTOS
CLIENTES
PEDIDOS
PRODUCTOS
DETALLEPEDIDOS
PAGOS
8 rows selected.
```

## NIVEL BÁSICO

### 1. CONSULTA 11

Mostrar las distintas ciudades en las que la empresa de jardinería tiene clientes. Fíjate en el resultado obtenido y si detectas errores de algún tipo, corrígelos (pista: hay dos errores).

```
SELECT DISTINCT ciudad FROM clientes; UPDATE clientes SET ciudad = "Barcelona" WHERE ciudad = " Barcelona"; UPDATE clientes SET ciudad = "Fuenlabrada" WHERE ciudad = "Fenlabrada";
```

### 2. CONSULTA 12

Mostrar cuántos clientes tenemos y mostrarlos en una columna denominada Num\_de\_Clientes.

```
SELECT COUNT(codigocliente) AS Num_de_Clientes FROM clientes;
```

### 3. CONSULTA 13

Mostrar el nombre, la cantidad en almacén y el precio de compra (proveedor) de los productos de la gama Herramientas ordenado por el nombre del producto.

```
SELECT nombre, cantidadenstock, precioproveedor FROM productos WHERE gama = "Herramientas" ORDER BY nombre;
```

### 4. CONSULTA 14

Mostrar la valoración del almacén de cada producto de la gama Herramientas (en una columna denominada Importe) ordenado por el importe obtenido y su nombre. (La valoración del almacén se realiza multiplicando los productos en stock por su precio de compra)

```
SELECT precioproveedor*cantidadenstock AS Valoracion, nombre FROM productos WHERE gama = "Herramientas" ORDER BY Valoracion, nombre;
```

### 5. CONSULTA 15

Mostrar el beneficio obtenido en la venta de cada producto de la gama Herramientas (en una columna denominada Beneficio). (El beneficio se calculará como el precio de venta menos el precio del proveedor)

```
SELECT precioventa-precioproveedor AS Beneficio, nombre FROM productos WHERE gama = "Herramientas";
```

### 6. CONSULTA 16

Mostrar cuál es el beneficio máximo (en una columna denominada Beneficio) que se puede obtener con la venta de un producto de los que tenemos en Stock (si no tiene stock no cuenta).

```
SELECT (precioventa-precioproveedor)*cantidadenstock AS Beneficio_Max, nombre FROM productos WHERE cantidadenstock <> 0;
```

### 7. CONSULTA 17

Mostrar el Código del pedido, su fecha, el código del cliente y la fecha esperada del pedido para todos aquellos cuya fecha de entrega haya sido posterior a la esperada. El resultado debe estar ordenado por la fecha de pedido.

```
SELECT codigopedido, fechapedido, codigocliente, fechaesperada FROM pedidos WHERE fechaentrega > fechaesperada ORDER BY fechapedido;
```

## 8. CONSULTA 18

Obtener cuántos pedidos nos ha realizado el cliente que tiene el código 30.

### 9. CONSULTA 19.1

```
SELECT COUNT(codigopedido) FROM pedidos  
WHERE codigocliente = 30;
```

Se ha detectado que hay errores en nuestros datos. Se han encontrado pedidos con fecha de entrega nula y estado Entregado. Se desea encontrar esas inconsistencias y mostrarlas ordenadas por la fecha de pedido.

### 10. CONSULTA 19.2

```
SELECT codigopedido, fecha_pedido, estado FROM pedidos  
WHERE fechaentrega IS NULL AND estado = "Entregado"  
ORDER BY fechapedido;
```

Mostrar las líneas del pedido 10 ordenadas por el número de línea.

```
SELECT numerolinea, codigoproducto, cantidad, preciounidad FROM detallepedidos  
WHERE codigopedido = 10  
ORDER BY numerolinea;
```

**NIVEL MEDIO****1. CONSULTA 21**

Mostrar el número de clientes que tenemos en cada ciudad en una columna denominada Num\_de\_Clientes ordenado por el número de clientes de mayor a menor.

```
SELECT ciudad, COUNT(*) AS Num_de_Clientes FROM clientes
GROUP BY ciudad
ORDER BY Num_de_Clientes DESC;
```

**2. CONSULTA 22**

Mostrar el número de clientes que tenemos en cada ciudad de España en una columna denominada Num\_de\_Clientes, ordenado por la ciudad.

```
SELECT ciudad, COUNT(*) AS Num_de_Clientes FROM clientes
WHERE pais = 'España'
GROUP BY ciudad
ORDER BY ciudad;
```

**3. CONSULTA 23**

Mostrar el número de clientes que tenemos en cada ciudad de España con más de un cliente en una columna denominada Num\_de\_Clientes, ordenado de mayor a menor por el número de clientes.

```
SELECT ciudad, COUNT(*) AS Num_de_Clientes FROM clientes
WHERE pais = 'España'
GROUP BY ciudad
HAVING COUNT(*) > 1
ORDER BY Num_de_Clientes desc;
```

**4. CONSULTA 24**

Mostrar cuál es el beneficio máximo que se puede obtener con la venta de un producto de los que tenemos en Stock en cada una de las gamas que tenemos. Ordena el resultado por el beneficio de mayor a menor.

```
SELECT MAX(PrecioVenta - PrecioProveedor), gama FROM productos
WHERE CantidadEnStock > 0
GROUP BY gama
ORDER BY MAX(PrecioVenta - PrecioProveedor) desc;
```

**5. CONSULTA 25**

Obtener cuántos pedidos ha realizado cada cliente, ordenado por el número de pedidos, de mayor a menor número de pedidos.

```
SELECT CodigoCliente, COUNT(*) AS N_Pedidos FROM pedidos
GROUP BY CodigoCliente
ORDER BY COUNT(*) DESC;
```

**6. CONSULTA 26**

Mostrar cuántos pedidos ha rechazado cada uno de nuestros clientes, ordenado por el número de rechazo.

```
SELECT CodigoCliente, COUNT(*) AS Pedidos_rechazados FROM pedidos
WHERE estado = 'Rechazado'
GROUP BY CodigoCliente
ORDER BY COUNT(*) DESC;
```

**7. CONSULTA 27**

Mostrar el importe total del pedido número 10.

```
SELECT CodigoPedido, sum(preciounidad*cantidad) as ImporteTotal FROM detallepedidos
WHERE CodigoPedido = 10;
```

**8. CONSULTA 28**

Obtener la máxima cantidad de un producto solicitada en un pedido siempre que ésta sea mayor

o igual a 100. Mostrar el resultado ordenado por la Cantidad pedida.

#### 9. CONSULTA 29.1

```
SELECT codigopedido, MAX(cantidad) FROM detallepedidos
WHERE cantidad >= 100
GROUP BY codigopedido
ORDER BY max(cantidad);
```

Mostrar el código del producto y el importe total pedido de cada producto cuyo importe total esté situado entre los 800 y los 1000 euros ordenado por el total obtenido.

#### 10. CONSULTA 29.2

```
SELECT codigoproducto, SUM(cantidad*preciounidad) FROM detallepedidos
WHERE cantidad*preciounidad BETWEEN 800 AND 1000
ORDER BY SUM(cantidad*preciounidad);
```

Mostrar el código del producto y el importe total pedido de cada producto, de los productos con un precio mayor o igual a 50 euros y menor o igual a 100 y cuyo importe total esté situado entre los 800 y los 1000 euros, ordenado por el código del producto.

#### 11. CONSULTA 29.3

```
SELECT codigoproducto, cantidad*preciounidad FROM detallepedidos
WHERE preciounidad BETWEEN 50 AND 100 AND cantidad*preciounidad BETWEEN 800 AND 1000
ORDER BY codigoproducto;
```

Mostrar el código del cliente, su nombre y los números de los pedidos que han realizado los clientes del representante cuyo nombre es Emmanuel.

#### 12. CONSULTA 29.4

```
SELECT c.codigocliente, c.nombrecliente, p.codigopedido, e.nombre
FROM clientes c, pedidos p, empleados e
WHERE c.codigoempleadorepventas = e.codigoempleado
AND e.nombre = 'Emmanuel'
AND c.codigocliente = p.codigocliente;
```

Mostrar el nombre de los empleados y el número de pedidos realizados por todos sus clientes ordenado de menor a mayor por el número de pedidos.

#### 13. CONSULTA 29.5

```
SELECT empleados.nombre, COUNT(pedidos.codigopedido) FROM empleados, pedidos, clientes
WHERE empleados.codigoempleado = clientes.codigoempleadorepventas
AND clientes.codigocliente = pedidos.codigocliente
GROUP BY empleados.nombre
ORDER BY COUNT(pedidos.codigopedido);
```

Mostrar cuál es el beneficio máximo (en una columna denominada Beneficio) que se puede obtener con la venta de un producto de los que tenemos en Stock (si no tiene stock no cuenta). Necesitamos saber también a qué producto pertenece ese beneficio.

```
SELECT nombre, MAX(precioventa-precioproveedor) FROM productos
WHERE cantidadenstock > 0
GROUP BY nombre
ORDER BY MAX(precioventa-precioproveedor) DESC LIMIT 1;
```



**NIVEL AVANZADO****1. CONSULTA 31**

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

```
SELECT c.nombrecliente, COUNT(p.codigocliente) AS n_pedidos FROM clientes c, pedidos p
WHERE p.codigocliente = c.codigocliente
GROUP BY c.nombrecliente
HAVING COUNT(p.codigocliente) > 0 -----> no haría falta porque la condición es > 0
ORDER BY c.nombrecliente;
```

**2. CONSULTA 32**

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

```
SELECT d.codigopedido, SUM(d.cantidad * d.preciounidad) AS Precio_Total
FROM detallepedidos d, pedidos p, clientes c
WHERE d.codigopedido = p.codigopedido
AND p.codigocliente = c.codigocliente
AND c.nombrecliente = 'Beragua'
GROUP BY d.codigopedido
ORDER BY d.codigopedido;
```

**3. 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.

```
SELECT d.codigopedido, SUM(d.cantidad * d.preciounidad) AS Precio_total
FROM detallepedidos d
WHERE d.codigopedido BETWEEN 100 AND 110
GROUP BY d.codigopedido
ORDER BY d.codigopedido;
```

**4. CONSULTA 34**

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

```
SELECT MAX(importe_total) AS importe_maximo, MIN(importe_total) AS importe_minimo
FROM (
  SELECT SUM(preciounidad * cantidad) AS importe_total
  FROM detallepedidos
  GROUP BY codigopedido
) importepedido;
```

**5. 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.

```
SELECT p.nombre, SUM(d.cantidad) FROM detallepedidos d, productos p
WHERE p.codigoproducto = d.codigoproducto
GROUP BY p.nombre
HAVING SUM(d.cantidad) > 450
ORDER BY SUM(d.cantidad) desc;
```

**6. 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.

```
SELECT c.nombrecliente, SUM(d.preciounidad * d.cantidad)
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;
```

**7. CONSULTA 37**

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

```
SELECT p.nombre, MAX(p.precioventa) AS precio FROM productos p
GROUP BY p.nombre
ORDER BY MAX(p.precioventa) desc
LIMIT 1;
```

**8. 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.

```
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 Total_Pedido
FROM clientes c, pedidos p, detallepedidos d
WHERE c.codigocliente = p.codigocliente
AND p.codigopedido = d.codigopedido
AND p.codigopedido IN (100, 103, 106, 109)
GROUP BY p.codigopedido
ORDER BY p.codigopedido;
```