

SOLUCIONS EXERCICIS BD JARDINERIA

- 1) Por cada cliente (CodigoCliente): número de pedidos que ha hecho, fecha del su primer pedido y fecha del último.

```
select CodigoCliente, count(CodigoPedido), MIN(FechaPedido), MAX(FechaPedido)
from pedidos
group by CodigoCliente
```

Por cada cliente (CodigoCliente y NombreCliente): número de pedidos que ha hecho, fecha del su primer pedido y fecha del último.

```
select c.CodigoCliente, NombreCliente, count(CodigoPedido), MIN(FechaPedido),
MAX(FechaPedido)
from Pedidos p INNER JOIN Clientes c ON p.CodigoCliente=c.CodigoCliente
group by c.CodigoCliente, NombreCliente
```

Nota: Aunque no hubiéramos querido proyectar CodigoCliente deberíamos de todas formas haberlo incluido como criterio del GROUP BY porque NombreCliente no tiene restricción de único y podría darse el caso de que 2 clientes se llamaran igual.

- 2) Por cada gama: precio máximo, mínimo y medio de sus productos

```
select gama, max(PrecioVenta), min(PrecioVenta), avg(PrecioVenta)
from productos
group by Gama
```

- 3) Importe total de cada pedido

```
select sum(cantidad*PrecioUnidad), CodigoPedido
from detallepedidos
group by CodigoPedido
```

- 4) Nombre completo del empleado en una sola columna y número de clientes que tiene a su cargo

```
select concat(Nombre , ' ', Apellido1, ' ', Apellido2) , count(*)  
from empleados inner join clientes on CodigoEmpleado=CodigoEmpleadoRepVentas  
group by concat(Nombre , ' ', Apellido1, ' ', Apellido2),CodigoEmpleado
```

```
select concat(Nombre , ' ', Apellido1, ' ', Apellido2) , count(*)  
from empleados inner join clientes on CodigoEmpleado=CodigoEmpleadoRepVentas  
group by Nombre , Apellido1, Apellido2, CodigoEmpleado
```

Nota: El CodigoEmpleado se incluye en el GROUP BY porque podría darse el caso de 2 empleados que se llamaran igual.

- 5) Nombre y dirección de cada oficina y número de empleados que tienen de las oficinas que no sean de España o Spain.

```
select oficinas.CodigoOficina, oficinas.LineaDireccion1, count(*)  
from empleados inner join oficinas on empleados.CodigoOficina=  
oficinas.CodigoOficina  
where Pais not like 'Spain' and Pais not like 'España'  
group by oficinas.CodigoOficina, LineaDireccion1
```

```
select oficinas.CodigoOficina, oficinas.LineaDireccion1, count(*)  
from empleados inner join oficinas on empleados.CodigoOficina=  
oficinas.CodigoOficina  
where Pais NOT IN('Spain', 'España')  
group by oficinas.CodigoOficina, LineaDireccion1
```

Nota: En MySql hubiéramos obtenido un resultado análogo aunque no hubiéramos incluido LineaDireccion1 en los criterios ya que CodigoOficina es Clave Primaria de la tabla a la que pertenece el campo LineaDireccion1. Pero, como hemos comentado en los apuntes, esto no es sintácticamente correcto y podría dar error en otros GBD.

- 6) Importe total de los pagos efectuados por cada cliente en cada año.
- ```
select clientes.CodigoCliente, NombreCliente, year(FechaPago), sum(Cantidad)
from pagos inner join clientes on clientes.CodigoCliente= pagos.CodigoCliente
group by clientes.CodigoCliente, NombreCliente, year(FechaPago)
```

Nota: Idem 6: En MySql no haría falta incluir NombreCliente en los criterios.

- 7) Nombre del jefe y número de empleados que tiene a su cargo
- ```
select j.CodigoEmpleado, concat(j.nombre,' ',j.apellido1), count(*)
from empleados em inner join empleados j on em.CodigoJefe=j.CodigoEmpleado
group by j.CodigoEmpleado, concat(j.nombre,' ',j.apellido1)
```

HAVING

- 8) Número de pedidos que ha hecho cada cliente viendo el nombre y la dirección del cliente, de los clientes que han hecho más de 3 pedidos.
- ```
select count(*), clientes.NombreCliente, clientes.LineaDireccion1
from pedidos inner join clientes on clientes.CodigoCliente= pedidos.CodigoCliente
group by clientes.CodigoCliente, clientes.NombreCliente, clientes.LineaDireccion1
having count(*) >3
```
- 9) Número de pedidos que ha hecho cada cliente viendo el nombre y la dirección del cliente, de los clientes que han hecho pedidos por un importe (total de todos los pedidos que haya hecho) mayor que 1000€.
- ```
select count(DISTINCT p.CodigoPedido), NombreCliente, LineaDireccion1
from pedidos p inner join clientes c on c.CodigoCliente= p.CodigoCliente inner join
DetallePedidos d on d.CodigoPedido=p.CodigoPedido
group by c.CodigoCliente, NombreCliente, LineaDireccion1
having sum(Cantidad*PrecioUnidad)>1000
```

- 10) Por cada gama, precio máximo, mínimo y medio de sus productos. De las categorías con menos de 10 productos

```
select gama, min(precioventa), max(precioventa), avg(precioventa)
from productos
group by gama
having count(*)<10
```

- 11) Fecha del primer pedido de cada cliente que haya hecho algún pedido en el 2018 o después.

```
select min(FechaPedido), NombreCliente, max(FechaPedido)
from pedidos inner join clientes on pedidos.CodigoCliente = clientes.CodigoCliente
group by clientes.CodigoCliente, NombreCliente
having max(year(FechaPedido))>=2018
```

- 12) Nombre completo del empleado y número de clientes que tiene a su cargo de los empleados que trabajen en oficinas de Madrid o Barcelona y tengan al menos tres clientes.

```
select concat(Nombre, ' ', Apellido1) NombreCompleto, count(*) 'nº de clientes'
from empleados inner join clientes on empleados.CodigoEmpleado =
clientes.CodigoEmpleadoRepVentas
inner join oficinas on empleados.CodigoOficina = oficinas.codigoOficina
where oficinas.ciudad like 'Madrid' or oficinas.ciudad like 'Barcelona'
group by CodigoEmpleado, concat(Nombre, ' ', Apellido1)
having count(*)>2
```

- 13) Nombre y dirección de cada oficina y número de empleados que tiene de las oficinas que no sean de España o Spain y tengan menos de 10 empleados

```
select oficinas.CodigoOficina, LineaDireccion1, count(*)
from empleados inner join oficinas on empleados.CodigoOficina=
oficinas.CodigoOficina
where Pais not like 'Spain' and Pais not like 'España'
group by oficinas.CodigoOficina, LineaDireccion1
having count(CodigoEmpleado)<10
```

- 14) Importe total de los pagos efectuados por cada cliente en cada año a partir del año 2017

```
select sum(Cantidad), CodigoCliente, year(FechaPago)
from Pagos
group by CodigoCliente, year(FechaPago)
having year(FechaPago)>=2017
```

(la restricción también podría haberse hecho en el WHERE)

```
select sum(Cantidad), CodigoCliente, year(FechaPago)
from Pagos
where year(FechaPago)>=2017
group by CodigoCliente, year(FechaPago)
```