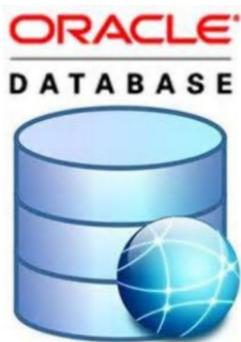


Tema 5.2 : Cont. Select



Ejemplos BD Jardinería

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

```
-----+
| empleados | CREATE TABLE `empleados` (
|           |   `CodigoEmpleado` int NOT NULL,
|           |   `Nombre` varchar(50) NOT NULL,
|           |   `Apellido1` varchar(50) NOT NULL,
|           |   `Apellido2` varchar(50) DEFAULT NULL,
|           |   `Extension` varchar(10) NOT NULL,
|           |   `Email` varchar(100) NOT NULL,
|           |   `CodigoOficina` varchar(10) NOT NULL,
|           |   `CodigoJefe` int DEFAULT NULL,
|           |   `Puesto` varchar(50) DEFAULT NULL,
|           |   PRIMARY KEY (`CodigoEmpleado`),
|           |   KEY `Empleados_OficinasFK` (`CodigoOficina`),
|           |   KEY `Empleados_EmpleadosFK` (`CodigoJefe`),
|           |   CONSTRAINT `Empleados_EmpleadosFK` FOREIGN KEY (`CodigoJefe`) REFERENCES `empleados`(`CodigoEmpleado`),
|           |   CONSTRAINT `Empleados_OficinasFK` FOREIGN KEY (`CodigoOficina`) REFERENCES `oficinas`(`CodigoOficina`)
|           |   ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+
```

Ejemplos BD Jardinería

codigoEmpleado	nombre	apellido1	apellido2	codigoJefe
1	Marcos	Magaña	Perez	NULL
2	Ruben	Lopez	Martinez	1
3	Alberto	Soria	Carrasco	2
4	Maria	Solis	Jerez	2
5	Felipe	Rosas	Marquez	3
6	Juan Carlos	Ortiz	Serrano	3
7	Carlos	Soria	Jimenez	3
8	Mariano	Lopez	Murcia	7
9	Lucio	Campoamor	Martin	7
10	Hilario	Rodriguez	Huertas	7
11	Emmanuel	Magaña	Perez	3
12	José Manuel	Martinez	De la Osa	11
13	David	Palma	Aceituno	11
14	Oscar	Palma	Aceituno	11
15	Francois	Fignon		3
16	Lionel	Narvaez		
17	Laurent	Serra		27 Larry
18	Michael	Bolton		28 John
19	Walter Santiago	Sanchez		29 Kevin
20	Hilary	Washington		30 Julian
21	Marcus	Paxton		31 Mariko
22	Lorena	Paxton		
23	Nei	Nishikori		
24	Narumi	Riko		
25	Takuma	Nomura		
26	Amy	Johnson		

Visualiza el código del empleado, su nombre, apellidos y el código de su jefe.

Ejemplos BD Jardinería

- Observamos que hay un empleado que en el campo codigojefe tiene un NULL, para listar sólo los que tienen jefe deberíamos hacer:
- `Select codigoEmpleado,nombre,apellido1, apellido2, codigoJefe from empleados where codigojefe is not null;`

Ejemplos BD Jardinería

```
mysql> -- VISUALIZAR EL CODIGO DE EMPLEADO QUE ES JEFE Y CUANTOS EMPLEADOS TIENE
mysql>
mysql> select codigojefe,count(*) from empleados group by codigojefe order by 2 desc;
+-----+-----+
| codigojefe | count(*) |
+-----+-----+
|      3 |      10 |
|      7 |       3 |
|     11 |       3 |
|      2 |       2 |
|     15 |       2 |
|     20 |       2 |
|     23 |       2 |
|     26 |       2 |
|     29 |       2 |
|    NULL |       1 |
|      1 |       1 |
|    18 |       1 |
+-----+-----+
12 rows in set (0.00 sec)
```

Ejemplos BD Jardinería

En la consulta anterior observamos que hay un empleado que no tiene jefe (NULL). Vamos a visualizar sólo los que tienen jefe.

```
mysql> select codigojefe,count(*) from empleados group by codigojefe having codigoJefe is not null order by 2 desc ;
+-----+-----+
| codigojefe | count(*) |
+-----+-----+
|      3    |     10 |
|      7    |      3 |
|     11    |      3 |
|      2    |      2 |
|     15    |      2 |
|     20    |      2 |
|     23    |      2 |
|     26    |      2 |
|     29    |      2 |
|      1    |      1 |
|     18    |      1 |
+-----+-----+
11 rows in set (0.00 sec)
```

Ejemplos BD Jardinería

Visualiza el código de oficina y cuántos empleados hay en cada una .

Ejemplos BD Jardinería

Visualiza el código de oficina, ciudad, pais, la dirección 1 de cada oficina y cuántos empleados tiene.

5.9.- Consultas Reflexivas

- A veces, es necesario obtener información de relaciones reflexivas, por ejemplo un informe de empleados donde junto a su nombre y apellidos apareciera el nombre y apellidos de su jefe. Para ello hay que hacer un JOIN entre registros de la misma tabla:

```
mysql> use jardineria;
Database changed
mysql> describe empleados;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| CodigoEmpleado | int(11) | NO | PRI | NULL |       |
| Nombre | varchar(50) | NO |       | NULL |       |
| Apellido1 | varchar(50) | NO |       | NULL |       |
| Apellido2 | varchar(50) | YES |       | NULL |       |
| Extension | varchar(10) | NO |       | NULL |       |
| Email | varchar(100) | NO |       | NULL |       |
| CodigoOficina | varchar(10) | NO | MUL | NULL |       |
| CodigoJefe | int(11) | YES | MUL | NULL |       |
| Puesto | varchar(50) | YES |       | NULL |       |
+-----+-----+-----+-----+-----+-----+
rows in set (0.03 sec)

mysql>
mysql>
```

5.9.- Consultas Reflexivas

- VISUALIZA EL NOMBRE Y APELLIDO1 DE CADA EMPLEADO Y EL NOMBRE Y APELLIDO1 DE SU JEFE
- `select concat(e.nombre,
' , ',e.apellido1) as NEmpleado, concat (j.nombre,j.
apellido1) as NJefe from empleados
e, empleados j where e.codigojefe=j.codigoempleado;`

5.9.- Consultas Reflexivas

En la consulta anterior vemos que se está utilizando la tabla empleados dos veces una como Empleado (subordinado) y otra como Jefe. Por otro lado, se observa cómo sólo se visualizan únicamente los empleados que tienen jefe ya que es un inner join.

5.9.- Consultas Reflexivas

Visualiza el código del empleado, su nombre, apellidos de los empleados que son jefes y cuántos empleados Tienen.

5.10.- Consultas con Tablas Derivadas

- Las consultas con Tablas Derivadas o **inline views**, son aquellas que utilizan sentencias select en la cláusula FROM en lugar de nombres de tablas:

```
mysql> SELECT * FROM (SELECT CódigoEmpleado, Nombre FROM Empleados  
-> WHERE CódigoOficina='TAL-ES') as tabla_derivada;  
+-----+-----+  
| CódigoEmpleado | Nombre |  
+-----+-----+  
| 1 | Marcos |  
| 2 | Ruben |  
| 3 | Alberto |  
| 4 | Maria |  
| 5 | Felipe |  
| 6 | Juan Carlos |  
+-----+-----+  
6 rows in set (0.00 sec)  
mysql>
```

La tabla derivada (select CódigoEmpleado, nombre FROM Empleados) tiene un alias ‘tabla_derivada’. Es como una especie de tabla temporal resultado de ejecutar la consulta.

5.10.- Consultas con Tablas Derivadas

```
mysql> -- VISUALIZAR CUÁL ES EL MÁXIMO NÚMERO DE EMPLEADOS QUE PUEDE TENER UN JEFE
mysql> select max(numEmple) from (select codigojefe,count(*) numEmple from empleados group by codigojefe) tabla;
+-----+
| max(numEmple) |
+-----+
|      10      |
+-----+
1 row in set (0.00 sec)
```

5.10.- Consultas con Tablas Derivadas

- -- VISUALIZAR CUÁL ES EL MÁXIMO NUMERO DE EMPLEADOS POR OFICINA

5.10.- Consultas con Tablas Derivadas

- -- VISUALIZAR CUÁL ES LA OFICINA QUE TIENE EL MÁXIMO NUMERO DE EMPLEADOS

5.10.- Consultas con Tablas Derivadas

- Por ejemplo, para obtener la suma total de importes de la empresa, a partir de la suma de importes por código de pedido:
- `select sum(importe) from (select sum(cantidad*preciounidad) importe from detallepedidos group by codigopedido)t;`

```
MariaDB [jardineria]> select sum(importe) from (select sum(cantidad*preciounidad) importe from detallepedidos group by codigopedido)t;
+-----+
| sum(importe) |
+-----+
| 199538.00 |
+-----+
1 row in set (0.00 sec)
```

```
MariaDB [jardineria]> select sum(cantidad*preciounidad) from detallepedidos;
+-----+
| sum(cantidad*preciounidad) |
+-----+
| 199538.00 |
+-----+
1 row in set (0.00 sec)
```

5.10.- Consultas con Tablas Derivadas

- Por ejemplo, en la BD jardinería, si se quiere obtener el importe del pedido de menor coste de todos los pedidos. Primero se obtiene el total de todos los pedidos y luego el pedido de menor coste con la función de columna MIN:
- `select min(importe) from (select codigoPedido, nombre, precioVenta, cantidad, precioventa*cantidad Importe from detallePedidos p, productos pr where p.codigoProducto=pr.codigoProducto order by 2) as tabla;`

5.10.- Consultas con Tablas Derivadas

```
mysql>
mysql> select SUM(Cantidad*PrecioUnidad) as total,CodigoPedido
-> FROM DetallePedidos
-> Group BY CodigoPedido;
```

CodigoPedido	total
104	1682.00
105	1072.00
106	3216.00
107	660.00
108	520.00
109	264.00
110	154.00
111	61.00

89 rows in set (0.00 sec)

```
mysql>
```

```
MariaDB [jardineria]> select min(importe) from (select sum(cantidad*preciounidad) importe from detallepedidos group by codigo pedido )t;
+-----+
| min(importe) |
+-----+
| 4.00 |
+-----+
1 row in set (0.00 sec)
```

t es el alias de la tabla derivada, es obligatorio utilizar siempre un alias.

5.10.- Consultas con Tablas Derivadas

- Para determinar cuál es el código de pedido que le corresponde al importe menor de todos los pedidos:

Actividad 5.1: BD Jardinería

- Codifica en mySql sentencias para obtener la siguiente información:
 - 1.- El código de oficina y la ciudad donde hay oficinas.

```
mysql> select codigoOficina,Ciudad from Oficinas;
+-----+-----+
| codigoOficina | Ciudad |
+-----+-----+
| BCN-ES       | Barcelona |
| BOS-USA      | Boston    |
| LON-UK       | Londres   |
| MAD-ES       | Madrid    |
| PAR-FR       | Paris     |
| SFC-USA      | San Francisco |
| SYD-AU       | Sydney    |
| TAL-ES       | Talavera de la Reina |
| TOK-JP       | Tokyo     |
+-----+-----+
9 rows in set (0.00 sec)
```

Actividad 5.1

2.- Cuántos empleados hay en la compañía?

3.- Cuántos Clientes tiene cada país?

Actividad 5.1

4.- Cuál fue el pago medio en 2008 (usar la función YEAR)

5.- Cuántos pedidos están en cada estado ordenado descendente por el número de pedidos?

Actividad 5.1

6.- Precio del producto más caro y del más barato

Actividad 5.1

7.- El nombre del producto más caro.

Actividad 5.2

1.- Visualizar el nombre del representante de ventas de cada cliente.

Actividad 5.2

2.- Nombre del cliente con mayor límite de crédito

3.- Nombre, apellido1 y cargo de los empleados que no representen a ningún cliente

Actividad 5.3

1.-Listado en el que figure el nombre de cada cliente, y el nombre y apellidos de su representante de ventas

Actividad 5.3

2.- Mostrar el nombre de los clientes que no hayan realizado pagos junto con el nombre de sus representantes de ventas.

Actividad 5.3

3.-Listado de las ventas totales de los productos que hayan facturado más de 3000 euros. Se mostrará el nombre, unidades vendidas, total facturado y total facturado con impuestos (18% IVA)

Actividad 5.3

4/ Listar la dirección de las oficinas que tengan clientes en Fuenlabrada

Actividad 5.4

- Visualizar un listado en el que figure el nombre y apellido de cada empleado junto con el de su jefe

Actividad 5.5

- Visualizar la información de los empleados que pertenecen a la oficina ‘TAL_ES’.

Actividad 5.6

- Sacar el importe del pedido de menor coste de todos los pedidos.

Actividad 5.7

1/ Cliente que hizo el pedido de mayor cuantía

Actividad 5.7

- Con tablas derivadas: