CENTRE ESPECÍFIC
D´EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA

TEMA6
DISEÑO FÍSICO. DQL
EJERCICIOS SOLUCIONADOS

Bases de Datos
CFGS DAW

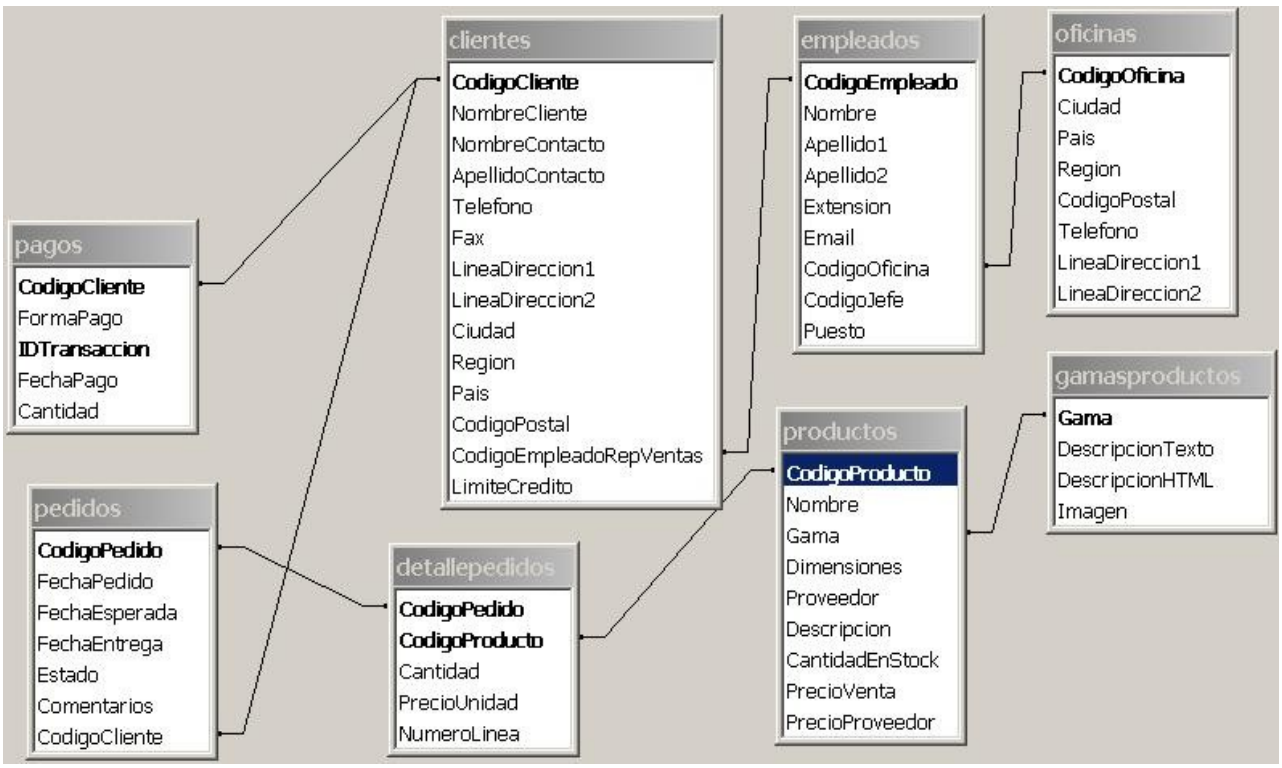
Raquel Torres
raquel.torres@ceedcv.es

Versión:180302.1339

UD06. DISEÑO FÍSICO. DQL

1. EJERCICIO 1

Disponemos del siguiente esquema de bbdd en el que se muestran las tablas que lo forman y cómo están relacionadas entre sí:



Por favor, 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 tanto en MySQL como en Oracle. 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.
```

1.1 Consulta 1

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).

```
mysql> select distinct ciudad from clientes;
+--------------------------+
| ciudad                   |
+--------------------------+
| San Francisco            |
| Miami                    |
| New York                 |
| Fuenlabrada              |
| Madrid                   |
| San Lorenzo del Escorial |
| Montornes del valles     |
| Santa cruz de Tenerife   |
| Barcelona                |
| Canarias                 |
|  Barcelona               |
| Sotogrande               |
| Humanes                  |
| Getafe                   |
| Fenlabrada               |
| Paris                    |
| Sydney                   |
| London                   |
+--------------------------+
18 rows in set (0.00 sec)
```

Fíjate en el resultado, como puedes observar a pesar de haber utilizado la cláusula distinct aparecen dos Barcelona, pero si lo observas bien, verás que el segundo está un poco separado de la columna. Esto es un error habitual, cuando se ha tecleado el dato se ha incluido sin querer un espacio en blanco delante del nombre, con lo cual la base de datos lo está tomando como dos ciudades distintas. Normalmente estas cosas deben estar controladas por el programa que permite la introducción de los datos, pero evidentemente éstos no son perfectos y a veces ocurren fallos como este. Si se observan anomalías de este tipo hay que corregirlas.

Para cambiar dicho registro y colocarlo adecuadamente tenemos varias opciones, yo voy a realizar una de ellas, pero espero que a vosotros al menos se os ocurran un par de ellas más.

```
mysql> update clientes set ciudad='Barcelona' where ciudad = ' Barcelona';
Query OK, 1 row affected (0.05 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Observa que en el filtro, después de la primera comilla, he dejado un espacio en blanco para reproducir el dato erróneo que hay en el campo. Si ahora vuelves a realizar la consulta anterior podrás observar que ya solo aparece una ciudad Barcelona.

Fíjate también que aparecen como ciudades Fuenlabrada y Fenlabrada, como es obvio, también es un error al teclear, se ha omitido una letra. Esto es más difícil de detectar en los programa si no hay una tabla de ciudades con la que comparar. Pues bien los datos deben ser revisados de forma periódica para detectar estos errores humanos.

Vamos a cambiarlo con un update.

```
mysql> update clientes set ciudad='Fuenlabrada' where ciudad='Fenlabrada';
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

1.2 Consulta 2

Mostrar cuántos clientes tenemos y mostrarlos en una columna denominada *Num_de_Clientes*.

```
mysql> Select count(*) as Num_de_Clientes from clientes;
+-----------------+
| Num_de_Clientes |
+-----------------+
|              36 |
+-----------------+
1 row in set (0.00 sec)
```

1.3 Consulta 3

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.

```
mysql> select count(*) as Num_de_Clientes, ciudad from clientes
    -> group by ciudad
    -> order by Num_de_Clientes DESC;
+-----------------+--------------------------+
| Num_de_Clientes | ciudad                   |
+-----------------+--------------------------+
|              11 | Madrid                   |
|               6 | Fuenlabrada              |
|               2 | Miami                    |
|               2 | Sydney                   |
|               2 | Humanes                  |
|               2 | Barcelona                |
|               2 | Paris                    |
|               1 | Sotogrande               |
|               1 | Santa cruz de Tenerife   |
|               1 | London                   |
|               1 | Getafe                   |
|               1 | Canarias                 |
|               1 | Montornes del valles     |
|               1 | San Francisco            |
|               1 | San Lorenzo del Escorial |
|               1 | New York                 |
+-----------------+--------------------------+
16 rows in set (0.00 sec)
```

1.4 Consulta 4

Mostrar el número de clientes que tenemos en cada ciudad de España en una columna denominada *Num_de_Clientes,* ordenado por la ciudad.

```
mysql> select count(*) as Num_de_Clientes, ciudad from clientes
    -> where pais = 'España'
    -> group by ciudad
    -> order by ciudad;
+-----------------+-------------------------+
| Num_de_Clientes | ciudad                  |
+-----------------+-------------------------+
|               2 | Barcelona               |
|               1 | Canarias                |
|               4 | Fuenlabrada             |
|               1 | Getafe                  |
|               2 | Humanes                 |
|              10 | Madrid                  |
|               1 | Montornes del valles    |
|               1 | Santa cruz de Tenerife  |
|               1 | Sotogrande              |
+-----------------+-------------------------+
9 rows in set (0.00 sec)
```

1.5 Consulta 5

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.

Analicemos el enunciado:

- Mostrar el número de clientes -> count(*)
- En cada ciudad -> agrupado por ciudad
- De España -> donde el país es España
- Con más de 1 clientes -> número de clientes > 1
- Ordenado de mayor a menor -> por número de clientes descendente.

Ahora creamos la instrucción SQL que tiene en cuenta todo esto.

```
mysql> select count(*) as Num_de_Clientes, Ciudad from clientes
    -> Where pais = 'España'
    -> Group By Ciudad
    -> Having count(*) > 1
    -> Order By Num_de_Clientes DESC;
+-----------------+-------------+
| Num_de_Clientes | Ciudad      |
+-----------------+-------------+
|              10 | Madrid      |
|               4 | Fuenlabrada |
|               2 | Barcelona   |
|               2 | Humanes     |
+-----------------+-------------+
4 rows in set (0.00 sec)
```

1.6 Consulta 6

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.

```
mysql> select nombre,cantidadenstock,precioproveedor from productos
    -> where gama='Herramientas'
    -> Order by Nombre;
+----------------------+-----------------+-----------------+
| nombre               | cantidadenstock | precioproveedor |
+----------------------+-----------------+-----------------+
| Azadón               |              15 |           11.00 |
| Pala                 |              15 |           13.00 |
| Rastrillo de Jardín  |              15 |           11.00 |
| Sierra de Poda 400MM |              15 |           11.00 |
+----------------------+-----------------+-----------------+
4 rows in set (0.02 sec)
```

1.7 Consulta 7

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)

```
mysql> select Nombre, (CantidadenStock * PrecioProveedor) as Importe
    -> from productos
    -> where gama = 'Herramientas'
    -> Order by Importe, Nombre;
+----------------------+---------+
| Nombre               | Importe |
+----------------------+---------+
| Azadón               |  165.00 |
| Rastrillo de Jardín  |  165.00 |
| Sierra de Poda 400MM |  165.00 |
| Pala                 |  195.00 |
+----------------------+---------+
4 rows in set (0.01 sec)
```

1.8 Consulta 8

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)

```
mysql> Select Nombre, (PrecioVenta - PrecioProveedor) as Beneficio
    -> From productos
    -> where gama = 'Herramientas';
+----------------------+-----------+
| Nombre               | Beneficio |
+----------------------+-----------+
| Sierra de Poda 400MM |      3.00 |
| Pala                 |      1.00 |
| Rastrillo de Jardín  |      1.00 |
| Azadón               |      1.00 |
+----------------------+-----------+
4 rows in set (0.00 sec)
```

1.9 Consulta 9

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).

```
mysql> select max(PrecioVenta - PrecioProveedor) as Beneficio
    -> from productos
    -> where CantidadenStock > 0;
+-----------+
| Beneficio |
+-----------+
|     93.00 |
+-----------+
1 row in set (0.00 sec)
```

1.10 **Consulta 10**

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.

```
mysql> select max(PrecioVenta - PrecioProveedor) as Beneficio, Gama
    -> from productos
    -> where CantidadenStock > 0
    -> Group By Gama
    -> Order By Beneficio DESC;
+-----------+--------------+
| Beneficio | Gama         |
+-----------+--------------+
|     93.00 | Ornamentales |
|     20.00 | Frutales     |
|      3.00 | Herramientas |
|      1.00 | Aromáticas   |
+-----------+--------------+
4 rows in set (0.02 sec)
```

1.11 **Consulta 11**

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.

```
mysql> select CodigoPedido,fechapedido,CodigoCliente, FechaEsperada
    -> from pedidos
    -> where FechaEsperada < fechaentrega
    -> order by fechapedido;
+--------------+-------------+---------------+----------------+
| CodigoPedido | fechapedido | CodigoCliente | FechaEsperada  |
+--------------+-------------+---------------+----------------+
|           32 | 2007-01-07  |             4 | 2007-01-19     |
|           96 | 2008-03-20  |            35 | 2008-04-12     |
|           49 | 2008-07-12  |            26 | 2008-07-22     |
|           31 | 2008-09-04  |            13 | 2008-09-30     |
|          113 | 2008-10-28  |            36 | 2008-11-09     |
|          128 | 2008-11-10  |            38 | 2008-12-10     |
|          115 | 2008-11-29  |            36 | 2009-01-26     |
|           55 | 2008-12-10  |            14 | 2009-01-10     |
|           60 | 2008-12-22  |             1 | 2008-12-27     |
|            9 | 2008-12-22  |             1 | 2008-12-27     |
|           18 | 2009-01-05  |             9 | 2009-01-06     |
|           16 | 2009-01-06  |             7 | 2009-01-07     |
|           17 | 2009-01-08  |             7 | 2009-01-09     |
|           22 | 2009-01-11  |             9 | 2009-01-11     |
|           13 | 2009-01-12  |             7 | 2009-01-14     |
|          103 | 2009-01-15  |            30 | 2009-01-20     |
|          123 | 2009-01-15  |            30 | 2009-01-20     |
|          114 | 2009-01-15  |            36 | 2009-01-29     |
|           28 | 2009-02-10  |             3 | 2009-02-17     |
|           68 | 2009-02-10  |             3 | 2009-02-17     |
|           38 | 2009-03-05  |            19 | 2009-03-06     |
|          112 | 2009-03-05  |            36 | 2009-04-06     |
|           39 | 2009-03-06  |            19 | 2009-03-07     |
|           40 | 2009-03-09  |            19 | 2009-03-10     |
|           42 | 2009-03-22  |            19 | 2009-03-23     |
|           43 | 2009-03-25  |            23 | 2009-03-26     |
|           44 | 2009-03-26  |            23 | 2009-03-27     |
|           45 | 2009-04-01  |            23 | 2009-03-04     |
|           46 | 2009-04-03  |            23 | 2009-03-04     |
|           92 | 2009-04-19  |            27 | 2009-04-30     |
|          106 | 2009-05-13  |            30 | 2009-05-15     |
|          126 | 2009-05-13  |            30 | 2009-05-15     |
+--------------+-------------+---------------+----------------+
32 rows in set (0.00 sec)
```

1.12 **Consulta 12**

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

```
mysql> select count(*)
    -> from pedidos
    -> where codigocliente=30;
+----------+
| count(*) |
+----------+
|       10 |
+----------+
1 row in set (0.00 sec)
```

1.13 **Consulta 13**

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

```
mysql> select codigocliente, count(*) as cuantos_pedidos
    -> from pedidos
    -> GROUP by codigocliente
    -> order by cuantos_pedidos desc;
+---------------+-----------------+
| codigocliente | cuantos_pedidos |
+---------------+-----------------+
|             1 |              11 |
|            16 |              10 |
|            30 |              10 |
|             3 |               9 |
|            13 |               5 |
|            14 |               5 |
|            15 |               5 |
|            19 |               5 |
|            23 |               5 |
|            26 |               5 |
|            27 |               5 |
|            28 |               5 |
|            35 |               5 |
|            36 |               5 |
|             4 |               5 |
|            38 |               5 |
|             5 |               5 |
|             7 |               5 |
|             9 |               5 |
+---------------+-----------------+
19 rows in set (0.00 sec)
```

1.14 **Consulta 14**

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.

```
mysql> select CodigoCliente,FechaPedido, Estado
    -> from pedidos
    -> where FechaEntrega is Null AND Estado = 'Entregado'
    -> order by fechapedido;
+---------------+-------------+-----------+
| CodigoCliente | FechaPedido | Estado    |
+---------------+-------------+-----------+
|            28 | 2009-01-24  | Entregado |
|             3 | 2009-02-07  | Entregado |
|             3 | 2009-02-07  | Entregado |
+---------------+-------------+-----------+
3 rows in set (0.00 sec)
```

1.15 **Consulta 15**

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

```
mysql> select CodigoCliente, Count(*)
    -> from pedidos where estado='Rechazado'
    -> group by CodigoCliente
    -> order by 2 DESC;
+---------------+----------+
| CodigoCliente | Count(*) |
+---------------+----------+
|            13 |        2 |
|            16 |        2 |
|            30 |        2 |
|            38 |        2 |
|             1 |        2 |
|             3 |        2 |
|             4 |        2 |
|             5 |        2 |
|            14 |        1 |
|            15 |        1 |
|            19 |        1 |
|            23 |        1 |
|            28 |        1 |
|            36 |        1 |
|             7 |        1 |
|             9 |        1 |
+---------------+----------+
16 rows in set (0.00 sec)
```

1.16 Consulta 16

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

```
mysql> select NumeroLinea, CodigoProducto, Cantidad, PrecioUnidad
    -> from detallepedidos
    -> where CodigoPedido = 10
    -> order by NumeroLinea;
+-------------+----------------+----------+--------------+
| NumeroLinea | CodigoProducto | Cantidad | PrecioUnidad |
+-------------+----------------+----------+--------------+
|           1 | FR-91          |       30 |        75.00 |
|           2 | FR-82          |        5 |        70.00 |
|           3 | OR-234         |        5 |        64.00 |
+-------------+----------------+----------+--------------+
3 rows in set (0.00 sec)
```

1.17 Consulta 17

Mostrar el importe total del pedido número 10.

```
mysql> select sum(Cantidad * PrecioUnidad) as total
    -> FROM detallepedidos
    -> WHERE CodigoPedido = 10;
+---------+
| total   |
+---------+
| 2920.00 |
+---------+
1 row in set (0.00 sec)
```

1.18 Consulta 18

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.

```
mysql> select CodigoProducto, max(Cantidad)
    -> from detallepedidos
    -> group by CodigoProducto
    -> having max(Cantidad) >= 100
    -> order by 2;
+----------------+---------------+
| CodigoProducto | max(Cantidad) |
+----------------+---------------+
| AR-002         |           110 |
| OR-157         |           113 |
| FR-29          |           120 |
| FR-48          |           120 |
| 30310          |           143 |
| OR-247         |           150 |
| OR-177         |           150 |
| AR-006         |           180 |
| FR-57          |           203 |
| OR-214         |           212 |
| AR-009         |           290 |
| FR-17          |           423 |
| AR-008         |           450 |
+----------------+---------------+
13 rows in set (0.00 sec)
```

Fijaos en la forma de hacer el *Order by*, obviamente hubiera sido equivalente a hacer:

...
order by max(Cantidad);

1.19 Consulta 19

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.

```
mysql> select CodigoProducto, sum(Cantidad * PrecioUnidad) as Total_Producto
    -> from detallepedidos
    -> group BY CodigoProducto
    -> having sum(Cantidad * PrecioUnidad) BETWEEN 800 and 1000
    -> Order by Total_Producto;
+----------------+----------------+
| CodigoProducto | Total_Producto |
+----------------+----------------+
| OR-225         |         840.00 |
| FR-17          |         846.00 |
| OR-208         |         884.00 |
| FR-79          |         946.00 |
| OR-218         |         950.00 |
| OR-237         |         950.00 |
| FR-29          |         960.00 |
| OR-217         |         975.00 |
| FR-82          |         980.00 |
| AR-009         |         986.00 |
| 22225          |         996.00 |
+----------------+----------------+
11 rows in set (0.02 sec)
```

De la misma manera podríamos haber puesto:

...
having Total_Producto between 800 and 1000

...

1.20 Consulta 20

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.

```
mysql> select CodigoProducto, sum(Cantidad * PrecioUnidad) as Total_Producto
    -> from detallepedidos
    -> WHERE PrecioUnidad BETWEEN 50 and 100
    -> group BY CodigoProducto
    -> having sum(Cantidad * PrecioUnidad) BETWEEN 800 and 1000
    -> Order by CodigoProducto;
+----------------+----------------+
| CodigoProducto | Total_Producto |
+----------------+----------------+
| FR-82          |         980.00 |
| OR-217         |         975.00 |
+----------------+----------------+
2 rows in set (0.00 sec)
```

1.21 Consulta 21

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

```
mysql> SELECT C.CODIGOCLIENTE, C.NOMBRECLIENTE, P.CODIGOPEDIDO
    -> FROM  EMPLEADOS E, CLIENTES C, PEDIDOS P
    -> WHERE E.CODIGOEMPLEADO = C.CODIGOEMPLEADOREPVENTAS
    -> AND   C.CODIGOCLIENTE = P.CODIGOCLIENTE
    -> AND   E.NOMBRE = 'Emmanuel'
    -> ORDER BY C.CODIGOCLIENTE;
+---------------+---------------+--------------+
| CODIGOCLIENTE | NOMBRECLIENTE | CODIGOPEDIDO |
+---------------+---------------+--------------+
|             7 | Beragua       |           14 |
|             7 | Beragua       |           16 |
|             7 | Beragua       |           13 |
|             7 | Beragua       |           15 |
|             7 | Beragua       |           17 |
|             9 | Naturagua     |           22 |
|             9 | Naturagua     |           19 |
|             9 | Naturagua     |           21 |
|             9 | Naturagua     |           18 |
|             9 | Naturagua     |           20 |
+---------------+---------------+--------------+
10 rows in set (0.00 sec)
```

1.22 Consulta 22

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.

```
mysql> SELECT E.NOMBRE, COUNT(P.CODIGOPEDIDO) CUANTOS_PEDIDOS
    -> FROM CLIENTES C, PEDIDOS P, EMPLEADOS E
    -> WHERE C.CODIGOCLIENTE = P.CODIGOCLIENTE
    -> AND E.CODIGOEMPLEADO = C.CODIGOEMPLEADOREPVENTAS
    -> GROUP BY E.NOMBRE
    -> ORDER BY CUANTOS_PEDIDOS;
+-----------------+-----------------+
| NOMBRE          | CUANTOS_PEDIDOS |
+-----------------+-----------------+
| Michael         |               5 |
| Lorena          |              10 |
| Josú Manuel     |              10 |
| Mariano         |              10 |
| Mariko          |              10 |
| Emmanuel        |              10 |
| Julian          |              10 |
| Lucio           |              10 |
| Walter Santiago |              20 |
| Felipe          |              20 |
+-----------------+-----------------+
10 rows in set (0.00 sec)
```

1.23 Consulta 23

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 Marivi                |
| FLORES S.L.                  |
| Gardening Associates         |
| Gerudo Valley                |
| Golf S.A.                    |
| Jardin de Flores             |
| Jardineria Sara              |
| Jardinerýas Matýas SL        |
| Jardines y Mansiones CACTUS SL |
| Naturagua                    |
| Sotogrande                   |
| Tendo Garden                 |
| Tutifruti S.A                |
+------------------------------+
19 rows in set (0.02 sec)
```

1.24 Consulta 24

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)
```

1.25 Consulta 25

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)
```

1.26 Consulta 26

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.

1.27 Consulta 27

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 Citriodra (Tomillo limón) |             455 |
+---------------------------------+-----------------+
2 rows in set (0.00 sec)
```

1.28 Consulta 28

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 Marivi                   |       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)
```

1.29 Consulta 29

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)
```

1.30 Consulta 30

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)
```