|  |  |
| --- | --- |
| Ciclo/Módulo | BASE DE DATOS (DAW/DAM) |
| Nombre y Apellidos | Jorge Wang Wang |
| Fecha | 22/02/2023 |

Dado el siguiente Modelo E/R

Diagrama

Descripción generada automáticamente con confianza baja

1.- Realice el script necesario para importar el modelo en MySQL. (2 puntos) .Incluya el siguiente conjunto de datos:

CREATE DATABASE examen;

use examen;

create table comercial(

id int(10) AUTO\_INCREMENT not null PRIMARY key,

nombre varchar(100) not null,

apellido1 varchar(100) not null,

apellido2 varchar(100),

ciudad varchar(100),

comision float

);

create table cliente(

id int(10) AUTO\_INCREMENT not null PRIMARY key,

nombre varchar(100) not null,

apellido1 varchar(100),

apellido2 varchar(100),

ciudad varchar(100),

categoria int(10)

);

create table pedido(

id int(10) AUTO\_INCREMENT not null PRIMARY key,

cantidad double not null,

fecha date,

id\_cliente int(10) not null,

id\_comercial int(10) not null,

FOREIGN key (id\_cliente) REFERENCES cliente (id),

FOREIGN key (id\_comercial) REFERENCES comercial (id)

on UPDATE CASCADE

on delete no ACTION

);  
/\*modifique los comandos de los insert de comando y comercial debido a que se desordenaban y no me introducían correctamente los datos además añadi ciudades a comerciales que eran datos que faltaban\*/

INSERT INTO cliente (id, nombre, apellido1,apellido2, ciudad,categoria) VALUES(1, 'Aarón', 'Rivero', 'Gómez', 'Almería', 100);

INSERT INTO cliente (id, nombre, apellido1,apellido2, ciudad,categoria) VALUES(2, 'Adela', 'Salas', 'Díaz', 'Granada', 200);

INSERT INTO cliente (id, nombre, apellido1,apellido2, ciudad,categoria) VALUES(3, 'Adolfo', 'Rubio', 'Flores', 'Sevilla', NULL);

INSERT INTO cliente (id, nombre, apellido1,apellido2, ciudad,categoria) VALUES(4, 'Adrián', 'Suárez', NULL, 'Jaén', 300);

INSERT INTO cliente (id, nombre, apellido1,apellido2, ciudad,categoria) VALUES(5, 'Marcos', 'Loyola', 'Méndez', 'Almería', 200);

INSERT INTO cliente (id, nombre, apellido1,apellido2, ciudad,categoria) VALUES(6, 'María', 'Santana', 'Moreno', 'Cádiz', 100);

INSERT INTO cliente (id, nombre, apellido1,apellido2, ciudad,categoria) VALUES(7, 'Pilar', 'Ruiz', NULL, 'Sevilla', 300);

INSERT INTO cliente (id, nombre, apellido1,apellido2, ciudad,categoria) VALUES(8, 'Pepe', 'Ruiz', 'Santana', 'Huelva', 200);

INSERT INTO cliente (id, nombre, apellido1,apellido2, ciudad,categoria)VALUES(9, 'Guillermo', 'López', 'Gómez', 'Granada', 225);

INSERT INTO cliente (id, nombre, apellido1,apellido2, ciudad,categoria) VALUES(10, 'Daniel', 'Santana', 'Loyola', 'Sevilla', 125);

INSERT INTO comercial (id,nombre,apellido1,apellido2, ciudad, comision) VALUES(1, 'Daniel', 'Sáez', 'Vega', 'Madrid', 0.15);

INSERT INTO comercial (id,nombre,apellido1,apellido2, ciudad, comision) VALUES(2, 'Juan', 'Gómez', 'López', 'Barcelona', 0.13);

INSERT INTO comercial (id,nombre,apellido1,apellido2, ciudad, comision) VALUES(3, 'Diego','Flores', 'Salas', 'Valencia',0.11);

INSERT INTO comercial (id,nombre,apellido1,apellido2, ciudad, comision) VALUES(4, 'Marta','Herrera', 'Gil', 'Madrid', 0.14);

INSERT INTO comercial (id,nombre,apellido1,apellido2, ciudad, comision) VALUES(5, 'Antonio','Carretero', 'Ortega', 'Bilbao',0.12);

INSERT INTO comercial (id,nombre,apellido1,apellido2, ciudad, comision) VALUES(6, 'Manuel','Domínguez', 'Hernández', 'Sevilla', 0.13);

INSERT INTO comercial (id,nombre,apellido1,apellido2, ciudad, comision) VALUES(7, 'Antonio','Vega', 'Hernández', 'Barcelona', 0.11);

INSERT INTO comercial (id,nombre,apellido1,apellido2, ciudad, comision) VALUES(8, 'Alfredo','Ruiz', 'Flores', 'Vigo', 0.05);

INSERT INTO pedido VALUES(1, 150.5, '2017-10-05', 5, 2);

INSERT INTO pedido VALUES(2, 270.65, '2016-09-10', 1, 5);

INSERT INTO pedido VALUES(3, 65.26, '2017-10-05', 2, 1);

INSERT INTO pedido VALUES(4, 110.5, '2016-08-17', 8, 3);

INSERT INTO pedido VALUES(5, 948.5, '2017-09-10', 5, 2);

INSERT INTO pedido VALUES(6, 2400.6, '2016-07-27', 7, 1);

INSERT INTO pedido VALUES(7, 5760, '2015-09-10', 2, 1);

INSERT INTO pedido VALUES(8, 1983.43, '2017-10-10', 4, 6);

INSERT INTO pedido VALUES(9, 2480.4, '2016-10-10', 8, 3);

INSERT INTO pedido VALUES(10, 250.45, '2015-06-27', 8, 2);

INSERT INTO pedido VALUES(11, 75.29, '2016-08-17', 3, 7);

INSERT INTO pedido VALUES(12, 3045.6, '2017-04-25', 2, 1);

INSERT INTO pedido VALUES(13, 545.75, '2019-01-25', 6, 1);

INSERT INTO pedido VALUES(14, 145.82, '2017-02-02', 6, 1);

INSERT INTO pedido VALUES(15, 370.85, '2019-03-11', 1, 5);

INSERT INTO pedido VALUES(16, 2389.23, '2019-03-11', 1, 5);

**INSERT** **INTO** cliente **VALUES**(1, 'Aarón', 'Rivero', 'Gómez', 'Almería', 100);

**INSERT** **INTO** cliente **VALUES**(2, 'Adela', 'Salas', 'Díaz', 'Granada', 200);

**INSERT** **INTO** cliente **VALUES**(3, 'Adolfo', 'Rubio', 'Flores', 'Sevilla', **NULL**);

**INSERT** **INTO** cliente **VALUES**(4, 'Adrián', 'Suárez', **NULL**, 'Jaén', 300);

**INSERT** **INTO** cliente **VALUES**(5, 'Marcos', 'Loyola', 'Méndez', 'Almería', 200);

**INSERT** **INTO** cliente **VALUES**(6, 'María', 'Santana', 'Moreno', 'Cádiz', 100);

**INSERT** **INTO** cliente **VALUES**(7, 'Pilar', 'Ruiz', **NULL**, 'Sevilla', 300);

**INSERT** **INTO** cliente **VALUES**(8, 'Pepe', 'Ruiz', 'Santana', 'Huelva', 200);

**INSERT** **INTO** cliente **VALUES**(9, 'Guillermo', 'López', 'Gómez', 'Granada', 225);

**INSERT** **INTO** cliente **VALUES**(10, 'Daniel', 'Santana', 'Loyola', 'Sevilla', 125);

**INSERT** **INTO** comercial **VALUES**(1, 'Daniel', 'Sáez', 'Vega', 0.15);

**INSERT** **INTO** comercial **VALUES**(2, 'Juan', 'Gómez', 'López', 0.13);

**INSERT** **INTO** comercial **VALUES**(3, 'Diego','Flores', 'Salas', 0.11);

**INSERT** **INTO** comercial **VALUES**(4, 'Marta','Herrera', 'Gil', 0.14);

**INSERT** **INTO** comercial **VALUES**(5, 'Antonio','Carretero', 'Ortega', 0.12);

**INSERT** **INTO** comercial **VALUES**(6, 'Manuel','Domínguez', 'Hernández', 0.13);

**INSERT** **INTO** comercial **VALUES**(7, 'Antonio','Vega', 'Hernández', 0.11);

**INSERT** **INTO** comercial **VALUES**(8, 'Alfredo','Ruiz', 'Flores', 0.05);

**INSERT** **INTO** pedido **VALUES**(1, 150.5, '2017-10-05', 5, 2);

**INSERT** **INTO** pedido **VALUES**(2, 270.65, '2016-09-10', 1, 5);

**INSERT** **INTO** pedido **VALUES**(3, 65.26, '2017-10-05', 2, 1);

**INSERT** **INTO** pedido **VALUES**(4, 110.5, '2016-08-17', 8, 3);

**INSERT** **INTO** pedido **VALUES**(5, 948.5, '2017-09-10', 5, 2);

**INSERT** **INTO** pedido **VALUES**(6, 2400.6, '2016-07-27', 7, 1);

**INSERT** **INTO** pedido **VALUES**(7, 5760, '2015-09-10', 2, 1);

**INSERT** **INTO** pedido **VALUES**(8, 1983.43, '2017-10-10', 4, 6);

**INSERT** **INTO** pedido **VALUES**(9, 2480.4, '2016-10-10', 8, 3);

**INSERT** **INTO** pedido **VALUES**(10, 250.45, '2015-06-27', 8, 2);

**INSERT** **INTO** pedido **VALUES**(11, 75.29, '2016-08-17', 3, 7);

**INSERT** **INTO** pedido **VALUES**(12, 3045.6, '2017-04-25', 2, 1);

**INSERT** **INTO** pedido **VALUES**(13, 545.75, '2019-01-25', 6, 1);

**INSERT** **INTO** pedido **VALUES**(14, 145.82, '2017-02-02', 6, 1);

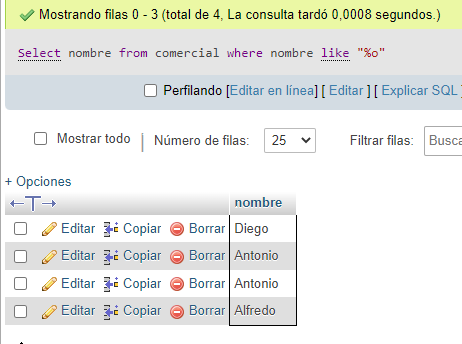
**INSERT** **INTO** pedido **VALUES**(15, 370.85, '2019-03-11', 1, 5);

**INSERT** **INTO** pedido **VALUES**(16, 2389.23, '2019-03-11', 1, 5);

2.- Realice las siguientes consultas. (Cada Consulta tendrá un valor de 0,3 puntos)

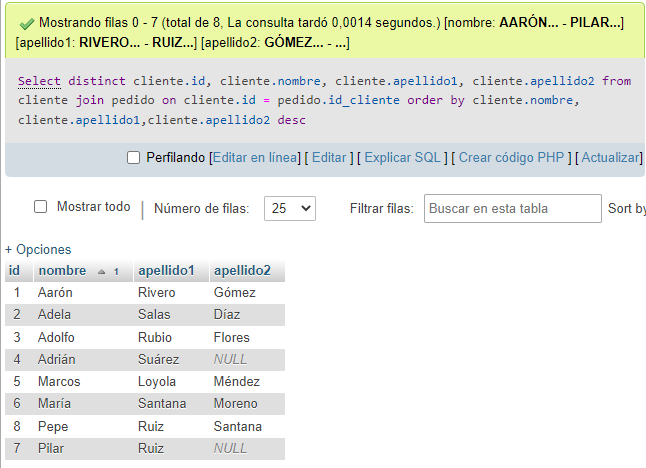
1. Devuelve un listado con los nombres de los comerciales que terminan por el o o. Tenga en cuenta que se deberán eliminar los nombres repetidos.

**Select nombre from comercial where nombre like "%o"**



1. Devuelve un listado con el identificador, nombre y los apellidos de todos los clientes que han realizado algún pedido. El listado debe estar ordenado alfabéticamente y se deben eliminar los elementos repetidos.

**Select distinct cliente.id, cliente.nombre, cliente.apellido1, cliente.apellido2 from cliente join pedido on cliente.id = pedido.id\_cliente order by cliente.nombre, cliente.apellido1,cliente.apellido2 desc**



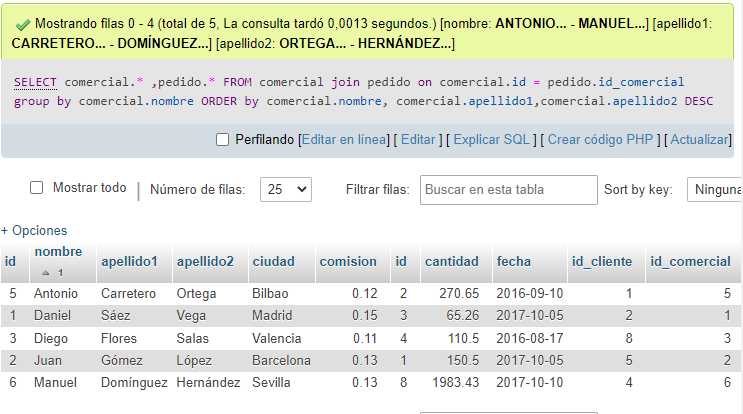
1. Devuelve un listado que muestre todos los pedidos que ha realizado cada cliente. El resultado debe mostrar todos los datos de los pedidos y del cliente. El listado debe mostrar los datos de los clientes ordenados alfabéticamente.

SELECT pedido.\* ,cliente.\* from pedido join cliente on cliente.id = pedido.id\_cliente group by cliente.nombre order by cliente.nombre DESC



1. Devuelve un listado que muestre todos los pedidos en los que ha participado un comercial. El resultado debe mostrar todos los datos de los pedidos y de los comerciales. El listado debe mostrar los datos de los comerciales ordenados alfabéticamente.

SELECT comercial.\* ,pedido.\* FROM comercial join pedido on comercial.id = pedido.id\_comercial group by comercial.nombre ORDER by comercial.nombre, comercial.apellido1,comercial.apellido2 DESC



1. Devuelve un listado que muestre todos los clientes, con todos los pedidos que han realizado y con los datos de los comerciales asociados a cada pedido.

SELECT cliente.\*, pedido.\* ,comercial.\* FROM `cliente` join pedido on cliente.id = pedido.id\_cliente join comercial on comercial.id = pedido.id\_comercial group by cliente.id



1. Devuelve un listado de todos los clientes que realizaron un pedido durante el año 2017, cuya cantidad esté entre 300 € y 1000 €.

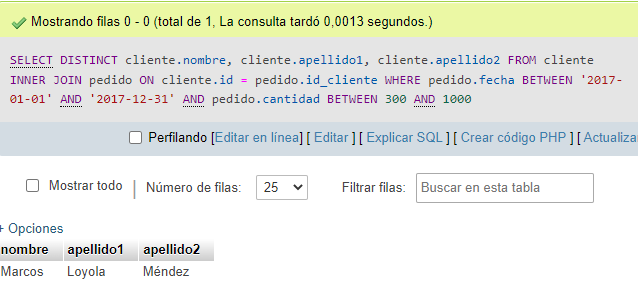
SELECT DISTINCT cliente.nombre, cliente.apellido1, cliente.apellido2

FROM cliente

INNER JOIN pedido ON cliente.id = pedido.id\_cliente

WHERE pedido.fecha BETWEEN '2017-01-01' AND '2017-12-31'

AND pedido.cantidad BETWEEN 300 AND 1000;



1. Devuelve el nombre y los apellidos de todos los comerciales que ha participado en algún pedido realizado por María Santana Moreno.

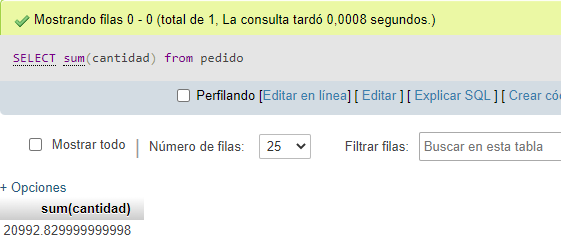
Select nombre, apellido from comercial inner join pedido on id=id inner join comercial on --- --- where cliente.nombre=”maría” and apellido =”santana” and apellido2 = “moreno”

1. Devuelve el nombre de todos los clientes que han realizado algún pedido con el comercial Daniel Sáez Vega.

Igual que el 7

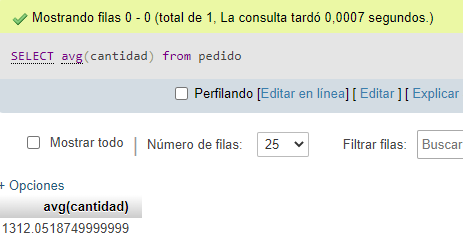
1. Calcula la cantidad total que suman todos los pedidos que aparecen en la tabla pedido.

SELECT sum(cantidad) from pedido



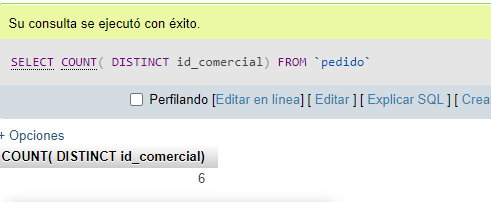
1. Calcula la cantidad media de todos los pedidos que aparecen en la tabla pedido.

SELECT avg(cantidad) from pedido



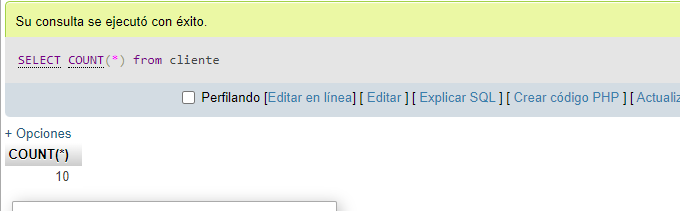
1. Calcula el número total de comerciales distintos que aparecen en la tabla pedido.

SELECT COUNT( DISTINCT id\_comercial) FROM `pedido`



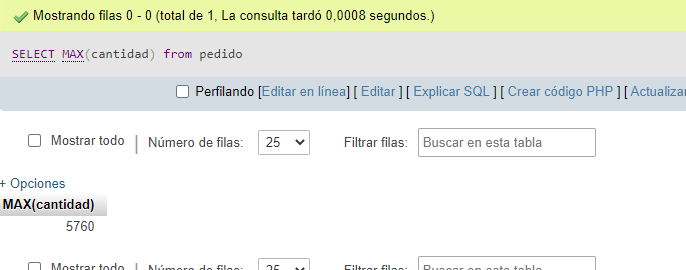
1. Calcula el número total de clientes que aparecen en la tabla cliente.

SELECT COUNT(\*) from cliente



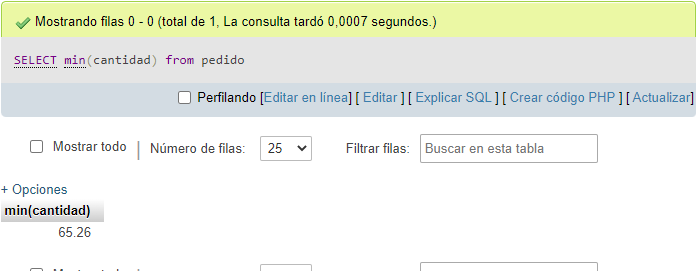
1. Calcula cuál es la mayor cantidad que aparece en la tabla pedido.

SELECT MAX(cantidad) from pedido



1. Calcula cuál es la menor cantidad que aparece en la tabla pedido.

SELECT min(cantidad) from pedido



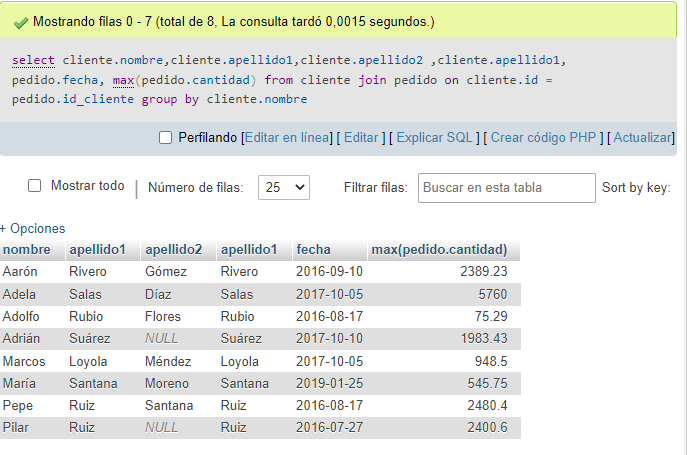
1. Calcula cuál es el valor máximo de categoría para cada una de las ciudades que aparece en la tabla cliente.

SELECT ciudad, max(categoria) from cliente GROUP by ciudad



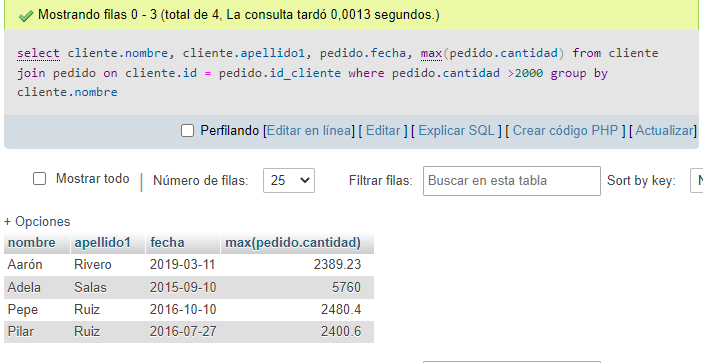
1. Calcula cuál es el máximo valor de los pedidos realizados durante el mismo día para cada uno de los clientes. Es decir, el mismo cliente puede haber realizado varios pedidos de diferentes cantidades el mismo día. Se pide que se calcule cuál es el pedido de máximo valor para cada uno de los días en los que un cliente ha realizado un pedido. Muestra el identificador del cliente, nombre, apellidos, la fecha y el valor de la cantidad.

select cliente.nombre,cliente.apellido1,cliente.apellido2 ,cliente.apellido1, pedido.fecha, max(pedido.cantidad) from cliente join pedido on cliente.id = pedido.id\_cliente group by cliente.nombre



1. Calcula cuál es el máximo valor de los pedidos realizados durante el mismo día para cada uno de los clientes, teniendo en cuenta que sólo queremos mostrar aquellos pedidos que superen la cantidad de 2000 €.

select cliente.nombre, cliente.apellido1, pedido.fecha, max(pedido.cantidad) from cliente join pedido on cliente.id = pedido.id\_cliente where pedido.cantidad >2000 group by cliente.nombre



1. Devuelve un listado que muestre el identificador de cliente, nombre, primer apellido y el valor de la máxima cantidad del pedido realizado por cada uno de los clientes. El resultado debe mostrar aquellos clientes que no han realizado ningún pedido indicando que la máxima cantidad de sus pedidos realizados es 0. Puede hacer uso de la función IFNULL.

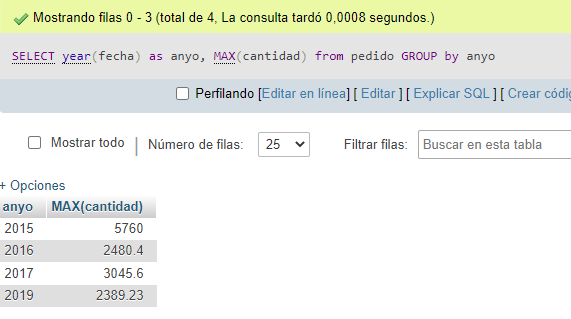
SELECT cliente.id, cliente.nombre, cliente.apellido1, IFNULL(MAX(pedido.cantidad), 0) AS max\_cantidad

FROM cliente LEFT JOIN pedido ON cliente.id = pedido.id\_cliente

GROUP BY cliente.id

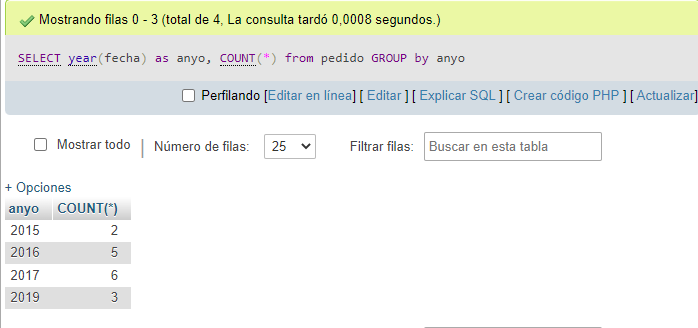


1. Devuelve cuál ha sido el pedido de máximo valor que se ha realizado cada año.

SELECT year(fecha) as anyo, MAX(cantidad) from pedido GROUP by anyo 

1. Devuelve el número total de pedidos que se han realizado cada año.

SELECT year(fecha) as anyo, COUNT(\*) from pedido GROUP by anyo



3.- Realice una aplicación informática que permita realizar las cinco primeras consultas. No es necesario realizar un menú. (2 puntos).