CONSULTAS 2 Erick Llanos Ríos 1º DAW

1.- Crea la siguiente base de datos en tu SGBD con las siguientes tablas y los siguientes registros.

```
DROP DATABASE IF EXISTS ventas;
CREATE DATABASE ventas CHARACTER SET utf8mb4;
USE ventas;
CREATE TABLE cliente (
id INT UNSIGNED AUTO INCREMENT PRIMARY KEY,
nombre VARCHAR(100) NOT NULL,
apellido1 VARCHAR(100) NOT NULL,
apellido2 VARCHAR(100),
ciudad VARCHAR(100),
categoría INT UNSIGNED
);
CREATE TABLE comercial (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
nombre VARCHAR(100) NOT NULL,
apellido1 VARCHAR(100) NOT NULL,
apellido2 VARCHAR(100),
comisión FLOAT
);
CREATE TABLE pedido (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
total DOUBLE NOT NULL,
fecha DATE,
id_cliente INT UNSIGNED NOT NULL,
id comercial INT UNSIGNED NOT NULL,
FOREIGN KEY (id_cliente) REFERENCES cliente(id),
FOREIGN KEY (id_comercial) REFERENCES comercial(id)
);
```

```
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);

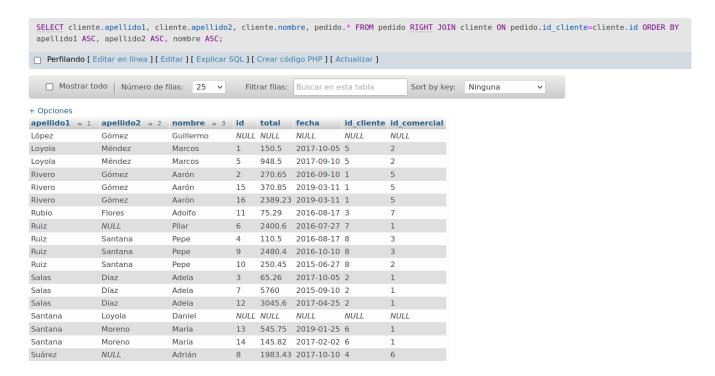
2.- Devuelve un listado con el identificador, nombre y los apellidos de todos los clientes que han realizado algún pedido. El listado debe estar ordenado alfabéticamente y se deben eliminar los elementos repetidos.

+ Opciones						
id	nombre 🔺 1	apellido1	apellido2			
1	Aarón	Rivero	Gómez			
2	Adela	Salas	Díaz			
3	Adolfo	Rubio	Flores			
4	Adrián	Suárez	NULL			
10	Daniel	Santana	Loyola			
9	Guillermo	López	Gómez			
5	Marcos	Loyola	Méndez			
6	María	Santana	Moreno			
8	Pepe	Ruiz	Santana			
7	Pilar	Ruiz	NULL			

3.- 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.



4.- Devuelve un listado con todos los clientes junto con los datos de los pedidos que han realizado. Este listado también debe incluir los clientes que no han realizado ningún pedido. El listado debe estar ordenado alfabéticamente por el primer apellido, segundo apellido y nombre de los clientes.



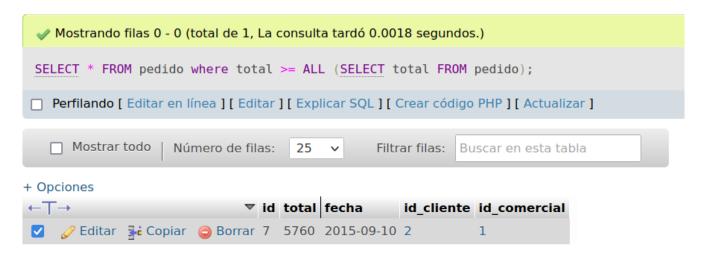
5.- Devuelve un listado que solamente muestre los comerciales que no han realizado ningún pedido.



6.- Devuelve un listado con todos los pedidos que ha realizado Adela Salas Díaz. (Sin utilizar INNER JOIN).



7.- Devuelve el pedido más caro que existe en la tabla pedido si hacer uso de MAX, ORDER BY ni LIMIT.



8.- Devuelve un listado de los clientes que no han realizado ningún pedido. (Utilizando IN o NOT IN).



9.- Devuelve un listado de los clientes que no han realizado ningún pedido. (Utilizando EXISTS o NOT EXISTS).

select * from comercial where NOT EXISTS

(SELECT id_comercial from pedido where pedido.id_comercial = comercial.id);

id	nombre	apellido1	apellido2	comisión
4	Marta	Herrera	Gil	0.14
8	Alfredo	Ruiz	Flores	0.05

10.- Borra la clave foránea de la tabla pedido: <u>FOREIGN</u> KEY (id_cliente) REFERENCES cliente(id). Crea una clave foránea en la tabla pedido: FOREIGN KEY (id_cliente) REFERENCES cliente(id), de forma que cuando elimine un registro o se actualice se haga en cascada.