



Crea la siguiente Base de datos usando el esquema de la imagen, luego ejecuta el script para llenar los datos. (Es importante que crees la base de datos exactamente como se ve en la imagen, sino lanzara errores).

```

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);
  
```

### 1.3.3 Consultas sobre una tabla

1. Devuelve un listado con todos los pedidos que se han realizado. Los pedidos deben estar ordenados por la fecha de realización, mostrando en primer lugar los pedidos más recientes.
2. Devuelve todos los datos de los dos pedidos de mayor valor.
3. Devuelve un listado con los identificadores de los clientes que han realizado algún pedido. Tenga en cuenta que no debe mostrar identificadores que estén repetidos.
4. Devuelve un listado de todos los pedidos que se realizaron durante el año 2017, cuya cantidad total sea superior a 500€.
5. Devuelve un listado con el nombre y los apellidos de los comerciales que tienen una comisión entre 0.05 y 0.11.
6. Devuelve el valor de la comisión de mayor valor que existe en la tabla `comercial`.
7. Devuelve el identificador, nombre y primer apellido de aquellos clientes cuyo segundo apellido **no** es NULL. El listado deberá estar ordenado alfabéticamente por apellidos y nombre.
8. Devuelve un listado de los nombres de los clientes que empiezan por A y terminan por n y también los nombres que empiezan por P. El listado deberá estar ordenado alfabéticamente.
9. Devuelve un listado de los nombres de los clientes que **no** empiezan por A. El listado deberá estar ordenado alfabéticamente.
10. Devuelve un listado con los nombres de los comerciales que terminan por e1 o o. Tenga en cuenta que se deberán eliminar los nombres repetidos.

### 1.3.4 Consultas multitable (Composición interna)

Resuelva todas las consultas utilizando la sintaxis de SQL1 y SQL2.

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.
2. 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.

3. 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.
4. 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.
5. Devuelve un listado de todos los clientes que realizaron un pedido durante el año 2017, cuya cantidad esté entre 300 € y 1000 €.
6. Devuelve el nombre y los apellidos de todos los comerciales que ha participado en algún pedido realizado por María Santana Moreno.
7. Devuelve el nombre de todos los clientes que han realizado algún pedido con el comercial Daniel Sáez Vega.