

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);
```

```
MariaDB [ventas]> show tables;
+-----+
| Tables_in_ventas |
+-----+
| cliente          |
| comercial        |
| pedido           |
+-----+
3 rows in set (0.001 sec)
```

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.

✓ Mostrando filas 0 - 9 (total de 10, La consulta tardó 0.0030 segundos.) [nombre: AARÓN... - PILAR...]

```
select distinct cliente.id, cliente.nombre, cliente.apellido1, cliente.apellido2 from cliente INNER JOIN pedido on cliente.id
order by nombre ASC;
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

+ Opciones

id	nombre	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.

✓ Mostrando filas 0 - 15 (total de 16, La consulta tardó 0.0019 segundos.)											
<code>SELECT pedido.*, cliente.* FROM pedido INNER JOIN cliente ON pedido.id_cliente = cliente.id;</code>											
<input type="checkbox"/> Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]											
<input type="checkbox"/> Mostrar todo Número de filas: <input type="text" value="25"/> v Filtrar filas: <input type="text" value="Buscar en esta tabla"/> Sort by key: <input type="text" value="Nir"/>											
+ Opciones											
id	total	fecha	id_cliente	id_comercial	id	nombre	apellido1	apellido2	ciudad	categoria	
1	150.5	2017-10-05	5	2	5	Marcos	Loyola	Méndez	Almería	200	
2	270.65	2016-09-10	1	5	1	Aarón	Rivero	Gómez	Almería	100	
3	65.26	2017-10-05	2	1	2	Adela	Salas	Díaz	Granada	200	
4	110.5	2016-08-17	8	3	8	Pepe	Ruiz	Santana	Huelva	200	
5	948.5	2017-09-10	5	2	5	Marcos	Loyola	Méndez	Almería	200	
6	2400.6	2016-07-27	7	1	7	Pilar	Ruiz	NULL	Sevilla	300	
7	5760	2015-09-10	2	1	2	Adela	Salas	Díaz	Granada	200	
8	1983.43	2017-10-10	4	6	4	Adrián	Suárez	NULL	Jaén	300	
9	2480.4	2016-10-10	8	3	8	Pepe	Ruiz	Santana	Huelva	200	
10	250.45	2015-06-27	8	2	8	Pepe	Ruiz	Santana	Huelva	200	
11	75.29	2016-08-17	3	7	3	Adolfo	Rubio	Flores	Sevilla	NULL	
12	3045.6	2017-04-25	2	1	2	Adela	Salas	Díaz	Granada	200	
13	545.75	2019-01-25	6	1	6	María	Santana	Moreno	Cádiz	100	
14	145.82	2017-02-02	6	1	6	María	Santana	Moreno	Cádiz	100	
15	370.85	2019-03-11	1	5	1	Aarón	Rivero	Gómez	Almería	100	
16	2389.23	2019-03-11	1	5	1	Aarón	Rivero	Gómez	Almería	100	

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.

```
SELECT cliente.apellido1, cliente.apellido2, cliente.nombre, pedido.* FROM pedido RIGHT JOIN cliente ON pedido.id_cliente=cliente.id ORDER BY apellido1 ASC, apellido2 ASC, nombre ASC;
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Sort by key: Ninguna

+ Opciones

apellido1	apellido2	nombre	id	total	fecha	id_cliente	id_comercial
López	Gómez	Guillermo	NULL	NULL	NULL	NULL	NULL
Loyola	Méndez	Marcos	1	150.5	2017-10-05	5	2
Loyola	Méndez	Marcos	5	948.5	2017-09-10	5	2
Rivero	Gómez	Aarón	2	270.65	2016-09-10	1	5
Rivero	Gómez	Aarón	15	370.85	2019-03-11	1	5
Rivero	Gómez	Aarón	16	2389.23	2019-03-11	1	5
Rubio	Flores	Adolfo	11	75.29	2016-08-17	3	7
Ruiz	NULL	Pilar	6	2400.6	2016-07-27	7	1
Ruiz	Santana	Pepe	4	110.5	2016-08-17	8	3
Ruiz	Santana	Pepe	9	2480.4	2016-10-10	8	3
Ruiz	Santana	Pepe	10	250.45	2015-06-27	8	2
Salas	Díaz	Adela	3	65.26	2017-10-05	2	1
Salas	Díaz	Adela	7	5760	2015-09-10	2	1
Salas	Díaz	Adela	12	3045.6	2017-04-25	2	1
Santana	Loyola	Daniel	NULL	NULL	NULL	NULL	NULL
Santana	Moreno	María	13	545.75	2019-01-25	6	1
Santana	Moreno	María	14	145.82	2017-02-02	6	1
Suárez	NULL	Adrián	8	1983.43	2017-10-10	4	6

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

✓ Mostrando filas 0 - 1 (total de 2, La consulta tardó 0.0016 segundos.)

```
SELECT * FROM comercial LEFT JOIN pedido ON comercial.id=pedido.id_comercial WHERE pedido.id is null;
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Sort by key: Ninguna

+ Opciones

id	nombre	apellido1	apellido2	comisión	id	total	fecha	id_cliente	id_comercial
4	Marta	Herrera	Gil	0.14	NULL	NULL	NULL	NULL	NULL
8	Alfredo	Ruiz	Flores	0.05	NULL	NULL	NULL	NULL	NULL

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

✓ Mostrando filas 0 - 2 (total de 3, La consulta tardó 0.0027 segundos.)

```
SELECT pedido.* FROM pedido WHERE pedido.id_cliente = (SELECT id FROM cliente WHERE nombre='Adela' AND apellido1 = 'Salas' AND apellido2='Díaz');
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Sort by key: Ninguna

+ Opciones

				id	total	fecha	id_cliente	id_comercial
<input type="checkbox"/>	Editar	Copiar	Borrar	3	65.26	2017-10-05	2	1
<input type="checkbox"/>	Editar	Copiar	Borrar	7	5760	2015-09-10	2	1
<input type="checkbox"/>	Editar	Copiar	Borrar	12	3045.6	2017-04-25	2	1

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

✓ Mostrando filas 0 - 0 (total de 1, La consulta tardó 0.0018 segundos.)

```
SELECT * FROM pedido where total >= ALL (SELECT total FROM pedido);
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

+ Opciones

				id	total	fecha	id_cliente	id_comercial
<input checked="" type="checkbox"/>	Editar	Copiar	Borrar	7	5760	2015-09-10	2	1

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

```
SELECT * from cliente where cliente.id not in (SELECT id_cliente from pedido);
```

id	nombre	apellido1	apellido2	ciudad	categoría
9	Guillermo	López	Gómez	Granada	225
10	Daniel	Santana	Loyola	Sevilla	125

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: FOREIGN 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.