

EXAMEN PRÁCTICO 2doTrimestre

Erick Llanos 1º DAW

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 INNER JOIN pedido ON cliente.id ORDER BY nombre ASC;
```

+ 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



2.- 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 pedido.*, cliente.* FROM pedido INNER JOIN cliente ON pedido.id_cliente = cliente.id;
```

id	total	fecha	id_cliente	id_comercial	id	nombre	apellido1	apellido2	ciudad	categoría
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

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

```
SELECT cliente.nombre, cliente.apellido1, cliente.apellido2, cliente.id FROM cliente WHERE cliente.id = (SELECT id_cliente FROM pedido WHERE YEAR(fecha)=2017 && TOTAL BETWEEN 300 AND 1000);
```

←T→	nombre	apellido1	apellido2	id
<input type="checkbox"/> Editar  Copiar  Borrar	Marcos	Loyola	Méndez	5

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;
```

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

```
SELECT DISTINCT comercial.nombre, comercial.apellido1, comercial.apellido2 FROM cliente INNER JOIN pedido ON
pedido.id_cliente=cliente.id INNER JOIN comercial on comercial.id=pedido.id_comercial WHERE cliente.nombre='María' AND
cliente.apellido1='Santana' AND cliente.apellido2='Moreno';
```

nombre	apellido1	apellido2
Daniel	Sáez	Vega

6.- Devuelve un listado que solamente muestre los clientes que no han realizado ningún pedido.

```
SELECT * FROM cliente LEFT JOIN pedido ON cliente.id=pedido.id_cliente WHERE pedido.id is null;
```

id	nombre	apellido1	apellido2	ciudad	categoría	id	total	fecha	id_cliente	id_comercial
9	Guillermo	López	Gómez	Granada	225	NULL	NULL	NULL	NULL	NULL
10	Daniel	Santana	Loyola	Sevilla	125	NULL	NULL	NULL	NULL	NULL

7.- Devuelve los datos del cliente que realizó el pedido más caro en el año 2019. (Sin utilizar INNER JOIN)

```
SELECT cliente.* FROM cliente WHERE id = (SELECT id_cliente FROM pedido WHERE total = (SELECT MAX(total) FROM pedido WHERE
FECHA LIKE '%2019%'));
```

	id	nombre	apellido1	apellido2	ciudad	categoría
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1	Aarón	Rivero	Gómez	Almería	100

8.- Devuelve un listado de los clientes que no han realizado ningún pedido. (Utilizando ANY o ALL).

```
SELECT * FROM cliente WHERE id!= ALL(SELECT id_cliente FROM pedido);
```

	id	nombre	apellido1	apellido2	ciudad	categoría
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	9	Guillermo	López	Gómez	Granada	225
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	10	Daniel	Santana	Loyola	Sevilla	125

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

```
SELECT * FROM comercial WHERE id NOT IN (SELECT id_comercial FROM pedido);
```

				id	nombre	apellido1	apellido2	comisión			
<input type="checkbox"/>		Editar		Copiar		Borrar	4	Marta	Herrera	Gil	0.14
<input type="checkbox"/>		Editar		Copiar		Borrar	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.

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0119 segundos.)

```
ALTER TABLE pedido DROP CONSTRAINT pedido_ibfk_1;
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0305 segundos.)

```
ALTER TABLE pedido ADD CONSTRAINT key_id_cliente FOREIGN KEY(id_cliente) REFERENCES cliente (id) ON UPDATE CASCADE ON DELETE CASCADE;
```

```
-----+
| Table | Create Table
|
+-----+
-----+
| pedido | CREATE TABLE `pedido` (
| `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
| `total` double NOT NULL,
| `fecha` date DEFAULT NULL,
| `id_cliente` int(10) unsigned NOT NULL,
| `id_comercial` int(10) unsigned NOT NULL,
| PRIMARY KEY (`id`),
| KEY `id_comercial` (`id_comercial`),
| KEY `key_id_cliente` (`id_cliente`),
| CONSTRAINT `key_id_cliente` FOREIGN KEY (`id_cliente`) REFERENCES `cliente` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,
| CONSTRAINT `pedido_ibfk_2` FOREIGN KEY (`id_comercial`) REFERENCES `comercial` (`id`)
| ) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=utf8mb4 |
+-----+
-----+
1 row in set (0.001 sec)

MariaDB [ventas]> |
```

```
MariaDB [ventas]>
```