
Consultas a la base de datos ventas

1.3 Gestión de ventas

1.3.3 Consultas sobre una tabla

01.- 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.

```
use ventas;
```

```
select *  
from pedido  
order by fecha desc;
```

```
+---+-----+-----+-----+-----+  
|id|total  |fecha      |id_cliente|id_comercial|  
+---+-----+-----+-----+-----+  
|15|370.85  |2019-03-11|1         |5           |  
|16|2389.23|2019-03-11|1         |5           |  
|13|545.75  |2019-01-25|6         |1           |  
|8 |1983.43|2017-10-10|4         |6           |  
|1 |150.5   |2017-10-05|5         |2           |  
|3 |65.26   |2017-10-05|2         |1           |  
|5 |948.5   |2017-09-10|5         |2           |  
|12|3045.6  |2017-04-25|2         |1           |  
|14|145.82  |2017-02-02|6         |1           |  
|9 |2480.4  |2016-10-10|8         |3           |  
|2 |270.65  |2016-09-10|1         |5           |  
|4 |110.5   |2016-08-17|8         |3           |  
|11|75.29   |2016-08-17|3         |7           |  
|6 |2400.6  |2016-07-27|7         |1           |  
|7 |5760    |2015-09-10|2         |1           |  
|10|250.45  |2015-06-27|8         |2           |  
+---+-----+-----+-----+-----+
```

02.- Devuelve todos los datos de los dos pedidos de mayor valor.

```
select *  
from pedido  
order by total desc  
limit 2;
```

id	total	fecha	id_cliente	id_comercial
7	5760	2015-09-10	2	1
12	3045.6	2017-04-25	2	1

03.- 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.

```
select distinct id_cliente
from pedido;
```

id_cliente
1
2
3
4
5
6
7
8

04.- Devuelve un listado de todos los pedidos que se realizaron durante el año 2017, cuya cantidad total sea superior a 500€.

```
select *
from pedido
where fecha between '2017-01-01' and '2017-12-31'
and total > 500;
```

O también:

```
select *
from pedido
where fecha >= '2017-01-01' and fecha <= '2017-12-31'
and total > 500;
```

id	total	fecha	id_cliente	id_comercial
5	948.5	2017-09-10	5	2
8	1983.43	2017-10-10	4	6
12	3045.6	2017-04-25	2	1

05.- Devuelve un listado con el nombre y los apellidos de los comerciales que tienen una comisión entre 0.05 y 0.11.

```
select nombre, apellido1, apellido2
from comercial
where comisión between 0.05 and 0.11;
```

O también:

```
select nombre, apellido1, apellido2
from comercial
where comisión >= 0.05 and comisión <= 0.11;
```

```
+-----+-----+-----+
| nombre | apellido1 | apellido2 |
+-----+-----+-----+
| Diego  | Flores    | Salas      |
| Antonio| Vega      | Hernández  |
| Alfredo| Ruiz      | Flores     |
+-----+-----+-----+
```

06.- Devuelve el valor de la comisión de mayor valor que existe en la tabla comercial.

```
select comisión
from comercial
order by comisión desc
limit 1;
```

O también:

```
select max(comisión)
from comercial;
```

```
+-----+
| comisión |
+-----+
| 0.15     |
+-----+
```

07.- 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.

```
select id, nombre, apellido1
from cliente
where apellido2 is not null
order by apellido1, apellido2, nombre;
```

```

+---+-----+-----+
|id|nombre  |apellido|
+---+-----+-----+
| 9 |Guillermo|López   |
| 5 |Marcos   |Loyola  |
| 1 |Aarón    |Rivero  |
| 3 |Adolfo   |Rubio   |
| 8 |Pepe     |Ruiz    |
| 2 |Adela    |Salas   |
|10|Daniel   |Santana |
| 6 |María    |Santana |
+---+-----+-----+

```

08.- 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.

```

select nombre
from cliente
where nombre like 'A%n'
or nombre like 'P%'
order by nombre;

```

```

+-----+
|nombre|
+-----+
|Aarón |
|Adrián|
|Pepe  |
|Pilar |
+-----+

```

09-. Devuelve un listado de los nombres de los clientes que no empiezan por A. El listado deberá estar ordenado alfabéticamente.

```

select nombre
from cliente
where nombre not like 'A%'
order by nombre;

```

```
+-----+
| nombre |
+-----+
| Daniel |
| Guillermo|
| Marcos |
| María |
| Pepe |
| Pilar |
+-----+
```

10.- 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 distinct nombre
from comercial
where nombre like '%el'
or nombre like '%o';
```

```
+-----+
| nombre |
+-----+
| Daniel |
| Diego |
| Antonio|
| Manuel |
| Alfredo|
+-----+
```

1.3.4 Consultas multitable (Composición interna)

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

01.- 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 c.id, c.nombre, c.apellido1, c.apellido2
from cliente as c join pedido as p
on c.id = p.id_cliente
order by c.apellido1, c.apellido2, c.nombre;
```

```

+---+-----+-----+-----+
|id|nombre|apellido1|apellido2|
+---+-----+-----+-----+
|5 |Marcos|Loyola   |Méndez   |
|1 |Aarón  |Rivero   |Gómez    |
|3 |Adolfo |Rubio    |Flores    |
|7 |Pilar  |Ruiz     |null      |
|8 |Pepe   |Ruiz     |Santana   |
|2 |Adela  |Salas    |Díaz      |
|6 |María  |Santana  |Moreno    |
|4 |Adrián |Suárez   |null      |
+---+-----+-----+-----+

```

02.- 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 p.*, c.*
from pedido as p join cliente as c
on p.id_cliente = c.id
order by c.apellido1, c.apellido2, c.nombre;

```

```

+---+-----+-----+-----+-----+-----+-----+-----+-----+
|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      |
|5 |948.5  |2017-09-10|5         |2           |5 |Marcos|Loyola   |Méndez   |Almería|200      |
|2 |270.65 |2016-09-10|1         |5           |1 |Aarón |Rivero   |Gómez    |Almería|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      |
|11|75.29  |2016-08-17|3         |7           |3 |Adolfo|Rubio    |Flores    |Sevilla|null     |
|6 |2400.6 |2016-07-27|7         |1           |7 |Pilar |Ruiz     |null      |Sevilla|300      |
|4 |110.5  |2016-08-17|8         |3           |8 |Pepe  |Ruiz     |Santana   |Huelva |200      |
|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      |
|3 |65.26  |2017-10-05|2         |1           |2 |Adela |Salas    |Díaz      |Granada|200      |
|7 |5760   |2015-09-10|2         |1           |2 |Adela |Salas    |Díaz      |Granada|200      |
|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      |
|8 |1983.43|2017-10-10|4         |6           |4 |Adrián|Suárez   |null      |Jaén   |300      |
+---+-----+-----+-----+-----+-----+-----+-----+-----+

```

03.- 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 p.*, c.*
from pedido as p join comercial as c
on p.id_comercial = c.id
order by c.apellido1, c.apellido2, c.nombre;

```

id	total	fecha	id_cliente	id_comercial	id_nombre	apellido1	apellido2	comisión
2	270.65	2016-09-10	1	5	Antonio	Carretero	Ortega	0.12
15	370.85	2019-03-11	1	5	Antonio	Carretero	Ortega	0.12
16	2389.23	2019-03-11	1	5	Antonio	Carretero	Ortega	0.12
8	1983.43	2017-10-10	4	6	Manuel	Dominguez	Hernández	0.13
4	110.5	2016-08-17	8	3	Diego	Flores	Salas	0.11
9	2480.4	2016-10-10	8	3	Diego	Flores	Salas	0.11
1	150.5	2017-10-05	5	2	Juan	Gómez	López	0.13
5	948.5	2017-09-10	5	2	Juan	Gómez	López	0.13
10	250.45	2015-06-27	8	2	Juan	Gómez	López	0.13
3	65.26	2017-10-05	2	1	Daniel	Sáez	Vega	0.15
6	2400.6	2016-07-27	7	1	Daniel	Sáez	Vega	0.15
7	5760	2015-09-10	2	1	Daniel	Sáez	Vega	0.15
12	3045.6	2017-04-25	2	1	Daniel	Sáez	Vega	0.15
13	545.75	2019-01-25	6	1	Daniel	Sáez	Vega	0.15
14	145.82	2017-02-02	6	1	Daniel	Sáez	Vega	0.15
11	75.29	2016-08-17	3	7	Antonio	Vega	Hernández	0.11

04.- 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 cl.*, p.*, co.*
from cliente as cl join pedido as p
on cl.id = p.id_cliente
join comercial as co
on p.id_comercial = co.id;
```

id_nombre	apellido1	apellido2	ciudad	categoria	id	total	fecha	id_cliente	id_comercial	id_nombre	apellido1	apellido2	comisión	
5	Marcos	Loyola	Ménde	Almería	200	1	150.5	2017-10-05	5	2	Juan	Gómez	López	0.13
1	Aarón	Rivero	Gómez	Almería	100	2	270.65	2016-09-10	1	5	Antonio	Carretero	Ortega	0.12
2	Adela	Salas	Díaz	Granada	200	3	65.26	2017-10-05	2	1	Daniel	Sáez	Vega	0.15
8	Pepe	Ruiz	Santana	Huelva	200	4	110.5	2016-08-17	8	3	Diego	Flores	Salas	0.11
5	Marcos	Loyola	Ménde	Almería	200	5	948.5	2017-09-10	5	2	Juan	Gómez	López	0.13
7	Pilar	Ruiz	null	Sevilla	300	6	2400.6	2016-07-27	7	1	Daniel	Sáez	Vega	0.15
2	Adela	Salas	Díaz	Granada	200	7	5760	2015-09-10	2	1	Daniel	Sáez	Vega	0.15
4	Adrián	Suárez	null	Jaén	300	8	1983.43	2017-10-10	4	6	Manuel	Dominguez	Hernández	0.13
8	Pepe	Ruiz	Santana	Huelva	200	9	2480.4	2016-10-10	8	3	Diego	Flores	Salas	0.11
6	Pepe	Ruiz	Santana	Huelva	200	10	250.45	2015-06-27	8	2	Juan	Gómez	López	0.13
3	Adolfo	Rubio	Flores	Sevilla	null	11	75.29	2016-08-17	3	7	Antonio	Vega	Hernández	0.11
2	Adela	Salas	Díaz	Granada	200	12	3045.6	2017-04-25	2	1	Daniel	Sáez	Vega	0.15
6	Maria	Santana	Moreno	Cádiz	100	13	545.75	2019-01-25	6	1	Daniel	Sáez	Vega	0.15
6	Maria	Santana	Moreno	Cádiz	100	14	145.82	2017-02-02	6	1	Daniel	Sáez	Vega	0.15
1	Aarón	Rivero	Gómez	Almería	100	15	370.85	2019-03-11	1	5	Antonio	Carretero	Ortega	0.12
1	Aarón	Rivero	Gómez	Almería	100	16	2389.23	2019-03-11	1	5	Antonio	Carretero	Ortega	0.12

05.- 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
from cliente join pedido
on cliente.id = pedido.id_cliente
where pedido.fecha between '2017-01-01' and '2017-12-31'
and total between 300 and 1000;
```

nombre	apellido1	apellido2
Marcos	Loyola	Ménde

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

```

select distinct co.nombre, co.apellido1, co.apellido2
from comercial as co join pedido as p
on co.id = p.id_comercial
join cliente as cl
on p.id_cliente = cl.id
where cl.nombre = 'María'
and cl.apellido1 = 'Santana'
and cl.apellido2 = 'Moreno';

```

```

+-----+-----+-----+
|nombre|apellido1|apellido2|
+-----+-----+-----+
|Daniel|Sáez    |Vega     |
+-----+-----+-----+

```

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

```

select distinct cl.nombre
from cliente as cl join pedido as p
on cl.id = p.id_cliente
join comercial as co
on p.id_comercial = co.id
where co.nombre = 'Daniel'
and co.apellido1 = 'Sáez'
and co.apellido2 = 'Vega';

```

```

+-----+
|nombre|
+-----+
|Adela  |
|Pilar  |
|María  |
+-----+

```

1.3.5 Consultas multitabla (Composición externa)

-- Resuelva todas las consultas utilizando las cláusulas LEFT JOIN y RIGHT JOIN.

--

-- 01.- 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 cl.nombre, cl.apellido1, p.*
from cliente as cl left join pedido as p
on cl.id = p.id_cliente
order by cl.apellido1, cl.apellido2, cl.nombre;

```


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

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

```
select co.apellido1, co.apellido2, co.nombre, p.*
from comercial as co left join pedido as p
on co.id = p.id_comercial
order by co.apellido1, co.apellido2, co.nombre;
```

apellido1	apellido2	nombre	id	total	fecha	id_cliente	id_comercial
Carretero	Ortega	Antonio	2	270.65	2016-09-10	1	5
Carretero	Ortega	Antonio	15	370.85	2019-03-11	1	5
Carretero	Ortega	Antonio	16	2389.23	2019-03-11	1	5
Domínguez	Hernández	Manuel	8	1983.43	2017-10-10	4	6
Flores	Salas	Diego	4	110.5	2016-08-17	8	3
Flores	Salas	Diego	9	2480.4	2016-10-10	8	3
Gómez	López	Juan	1	150.5	2017-10-05	5	2
Gómez	López	Juan	5	948.5	2017-09-10	5	2
Gómez	López	Juan	10	250.45	2015-06-27	8	2
Herrera	Gil	Marta	null	null	null	null	null
Ruiz	Flores	Alfredo	null	null	null	null	null
Sáez	Vega	Daniel	3	65.26	2017-10-05	2	1
Sáez	Vega	Daniel	6	2400.6	2016-07-27	7	1
Sáez	Vega	Daniel	7	5760	2015-09-10	2	1
Sáez	Vega	Daniel	12	3045.6	2017-04-25	2	1
Sáez	Vega	Daniel	13	545.75	2019-01-25	6	1
Sáez	Vega	Daniel	14	145.82	2017-02-02	6	1
Vega	Hernández	Antonio	11	75.29	2016-08-17	3	7

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

```
select cl.nombre, cl.apellido1, cl.apellido2
from cliente as cl left join pedido as p
on cl.id = p.id_cliente
where p.id is null;
```

```
+-----+-----+-----+
| nombre | apellido1 | apellido2 |
+-----+-----+-----+
| Guillermo | López | Gómez |
| Daniel | Santana | Loyola |
+-----+-----+-----+
```

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

```
select co.apellido1, co.apellido2, co.nombre
from comercial as co left join pedido as p
on co.id = p.id_comercial
where p.id is null;
```

```
+-----+-----+-----+
| apellido1 | apellido2 | nombre |
+-----+-----+-----+
| Herrera | Gil | Marta |
| Ruiz | Flores | Alfredo |
+-----+-----+-----+
```

-- 05.- Devuelve un listado con los clientes que no han realizado ningún pedido y de los comerciales que no han participado en ningún pedido. Ordene el listado alfabéticamente por los apellidos y el nombre. En el listado deberá diferenciar de algún modo los clientes y los comerciales.

```
select 'Cliente' as rol, concat_ws(' ', cl.apellido1, cl.apellido2,
cl.nombre) as nombre
from cliente as cl left join pedido as p
on cl.id = p.id_cliente
where p.id is null
union
select 'Comercial' as rol, concat_ws(' ', co.apellido1, co.apellido2,
co.nombre) as nombre
from comercial as co left join pedido as p
on co.id = p.id_comercial
where p.id is null

order by nombre;
```

```
+-----+-----+
| rol | nombre |
+-----+-----+
| Comercial | Herrera Gil Marta |
| Cliente | López Gómez Guillermo |
| Comercial | Ruiz Flores Alfredo |
| Cliente | Santana Loyola Daniel |
+-----+-----+
```

-- 06.- ¿Se podrían realizar las consultas anteriores con NATURAL LEFT JOIN o NATURAL RIGHT JOIN? Justifique su respuesta.

NO. LAS COLUMNAS A RELACIONAR TIENEN DISTINTOS NOMBRES.

1.3.6 Consultas resumen

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

```
select sum(total)
from pedido;
```

```
+-----+
| sum(total) |
+-----+
| 20992.829999999998 |
+-----+
```

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

```
select avg(total)
from pedido;
```

```
+-----+
| avg(total) |
+-----+
| 1312.0518749999999 |
+-----+
```

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

```
select count(distinct id_comercial)
from pedido;
```

```
+-----+
| count(distinct id_comercial) |
+-----+
| 6 |
+-----+
```

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

```
select count(*)
from cliente;
```

```
+-----+
| count(*) |
+-----+
| 10 |
+-----+
```

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

```
select max(total)
from pedido;
```

```
+-----+
|max(total)|
+-----+
| 5760      |
+-----+
```

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

```
select min(total)
from pedido;
```

```
+-----+
|min(total)|
+-----+
| 65.26     |
+-----+
```

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

```
select ciudad, max(categoría)
from cliente
group by ciudad;
```

```
+-----+-----+
|ciudad |max(categoría)|
+-----+-----+
|Almería|200           |
|Granada|225           |
|Sevilla|300           |
|Jaén   |300           |
|Cádiz  |100           |
|Huelva |200           |
+-----+-----+
```

-- 08.- 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 cl.id, concat_ws(' ', cl.nombre, cl.apellido1, cl.apellido2) as
cliente, p.fecha, max(p.total)
from cliente as cl join pedido as p
on cl.id = p.id_cliente
group by cl.id, cliente, p.fecha;
```

id	cliente	fecha	max(p.total)
5	Marcos Loyola Méndez	2017-10-05	150.5
1	Aarón Rivero Gómez	2016-09-10	270.65
2	Adela Salas Díaz	2017-10-05	65.26
8	Pepe Ruiz Santana	2016-08-17	110.5
5	Marcos Loyola Méndez	2017-09-10	948.5
7	Pilar Ruiz	2016-07-27	2400.6
2	Adela Salas Díaz	2015-09-10	5760
4	Adrián Suárez	2017-10-10	1983.43
8	Pepe Ruiz Santana	2016-10-10	2480.4
8	Pepe Ruiz Santana	2015-06-27	250.45
3	Adolfo Rubio Flores	2016-08-17	75.29
2	Adela Salas Díaz	2017-04-25	3045.6
6	María Santana Moreno	2019-01-25	545.75
6	María Santana Moreno	2017-02-02	145.82
1	Aarón Rivero Gómez	2019-03-11	2389.23

-- 09.- 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 cl.id, concat_ws(' ', cl.nombre, cl.apellido1, cl.apellido2) as
cliente, p.fecha, max(p.total)
from cliente as cl join pedido as p
on cl.id = p.id_cliente
where p.total > 2000
group by cl.id, cliente, p.fecha;
```

id	cliente	fecha	max(p.total)
7	Pilar Ruiz	2016-07-27	2400.6
2	Adela Salas Díaz	2015-09-10	5760
8	Pepe Ruiz Santana	2016-10-10	2480.4
2	Adela Salas Díaz	2017-04-25	3045.6
1	Aarón Rivero Gómez	2019-03-11	2389.23

-- 10.- Calcula el máximo valor de los pedidos realizados para cada uno de los comerciales durante la fecha 2016-08-17. Muestra el identificador del comercial, nombre, apellidos y total.

```
select co.id, concat_ws(' ', co.nombre, co.apellido1, co.apellido2) as
comercial, max(p.total)
from comercial as co join pedido as p
on co.id = p.id_comercial
where fecha = '2016-08-17'
group by co.id, comercial;
```

```

+---+-----+-----+
|id|comercial          |max(p.total)|
+---+-----+-----+
|3 |Diego Flores Salas  |110.5       |
|7 |Antonio Vega Hernández|75.29       |
+---+-----+-----+

```

-- 11.- Devuelve un listado con el identificador de cliente, nombre y apellidos y el número total de pedidos que ha realizado cada uno de clientes. Tenga en cuenta que pueden existir clientes que no han realizado ningún pedido. Estos clientes también deben aparecer en el listado indicando que el número de pedidos realizados es 0.

```

select cl.id, concat_ws(' ', cl.nombre, cl.apellido1, cl.apellido2) as
cliente, count(p.id_cliente)
from cliente as cl left join pedido as p
on cl.id = p.id_cliente
group by cl.id, cliente;

```

```

+---+-----+-----+
|id|cliente              |count(p.id_cliente)|
+---+-----+-----+
|1 |Aarón Rivero Gómez  |3                   |
|2 |Adela Salas Díaz    |3                   |
|3 |Adolfo Rubio Flores |1                   |
|4 |Adrián Suárez       |1                   |
|5 |Marcos Loyola Méndez|2                   |
|6 |María Santana Moreno|2                   |
|7 |Pilar Ruiz          |1                   |
|8 |Pepe Ruiz Santana   |3                   |
|9 |Guillermo López Gómez|0                   |
|10|Daniel Santana Loyola|0                   |
+---+-----+-----+

```

-- 12.- Devuelve un listado con el identificador de cliente, nombre y apellidos y el número total de pedidos que ha realizado cada uno de clientes durante el año 2017.

```

select cl.id, concat_ws(' ', cl.nombre, cl.apellido1, cl.apellido2) as
cliente, count(p.id_cliente)
from cliente as cl join pedido as p
on cl.id = p.id_cliente
where fecha between '2017-01-01' and '2017-12-31'
group by cl.id, cliente;

```

```

+---+-----+-----+
|id|cliente              |count(p.id_cliente)|
+---+-----+-----+
|5 |Marcos Loyola Méndez|2                   |
|2 |Adela Salas Díaz    |2                   |
|4 |Adrián Suárez       |1                   |
|6 |María Santana Moreno|1                   |
+---+-----+-----+

```

-- 13.- 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 cl.id, concat_ws(' ', cl.nombre, cl.apellido1) as cliente,
ifnull(max(p.total), 0)
from cliente as cl left join pedido as p
on cl.id = p.id_cliente
group by cl.id, cliente;
```

```
+---+-----+-----+
|id|cliente           |ifnull(max(p.total), 0)|
+---+-----+-----+
|1 |Aarón Rivero      |2389.23                 |
|2 |Adela Salas        |5760                    |
|3 |Adolfo Rubio       |75.29                   |
|4 |Adrián Suárez     |1983.43                 |
|5 |Marcos Loyola      |948.5                   |
|6 |María Santana     |545.75                  |
|7 |Pilar Ruiz         |2400.6                  |
|8 |Pepe Ruiz          |2480.4                  |
|9 |Guillermo López   |0                       |
|10|Daniel Santana    |0                       |
+---+-----+-----+
```

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

```
select year(fecha), max(total)
from pedido
group by year(fecha)
order by year(fecha);
```

```
+-----+-----+
|year(fecha)|max(total)|
+-----+-----+
|2015       |5760      |
|2016       |2480.4    |
|2017       |3045.6    |
|2019       |2389.23   |
+-----+-----+
```

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

```
select year(fecha), count(pedido.id)
from pedido
group by year(fecha)
order by year(fecha);
```

```
+-----+-----+
|year(fecha)|count(pedido.id)|
+-----+-----+
|2015       |2                |
|2016       |5                |
|2017       |6                |
|2019       |3                |
+-----+-----+
```

1.3.7 Subconsultas

-- 1.3.7.1 Con operadores básicos de comparación

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

```
use ventas;
select *
from pedido
where id_cliente = (select id
                    from cliente
                    where nombre = 'Adela'
                    and apellido1 = 'Salas'
                    and apellido2 = 'Díaz');
```

```
+---+-----+-----+-----+-----+
|id|total |fecha      |id_cliente|id_comercial|
+---+-----+-----+-----+-----+
| 3 |65.26 |2017-10-05|2         |1           |
| 7 |5760  |2015-09-10|2         |1           |
|12|3045.6|2017-04-25|2         |1           |
+---+-----+-----+-----+-----+
```

-- 02.- Devuelve el número de pedidos en los que ha participado el comercial Daniel Sáez Vega. (Sin utilizar INNER JOIN)

```
select count(p.id)
from pedido as p
where id_comercial = (select id
                     from comercial
                     where nombre = 'Daniel'
                     and apellido1 = 'Sáez'
                     and apellido2 = 'Vega');
```

```
+-----+
|count(p.id)|
+-----+
|6           |
+-----+
```

-- 03.- 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 year(fecha) = 2019));
```

```
+---+-----+-----+-----+-----+-----+
|id|nombre|apellido1|apellido2|ciudad |categoría|
+---+-----+-----+-----+-----+-----+
| 1 |Aarón |Rivero   |Gómez    |Almería|100      |
+---+-----+-----+-----+-----+-----+
```

-- 04.- Devuelve la fecha y la cantidad del pedido de menor valor realizado por

el cliente Pepe Ruiz Santana.

```
select fecha, total
from pedido
where total = (select min(total)
               from pedido
               where id_cliente = (select id
                                   from cliente
                                   where nombre = 'Pepe'
                                   and apellido1 = 'Ruiz'
                                   and apellido2 = 'Santana'));
```

```
+-----+-----+
| fecha      | total |
+-----+-----+
| 2016-08-17 | 110.5 |
+-----+-----+
```

-- 05.- Devuelve un listado con los datos de los clientes y los pedidos, de todos los clientes que han realizado un pedido durante el año 2017 con un valor mayor o igual al valor medio de los pedidos realizados durante ese mismo año.

```
select cl.*, p.*
from pedido as p join cliente as cl
on p.id_cliente = cl.id
where year(fecha) = 2017
and total >= (select avg(total)
              from pedido
              where year(fecha) = 2017);
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | nombre | apellido1 | apellido2 | ciudad | categoria | id | total | fecha      | id_cliente | id_comercial |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 4  | Adrián | Suárez    | null       | Jaén    | 300        | 8  | 1983.43 | 2017-10-10 | 4          | 6            |
| 2  | Adela  | Salas     | Díaz       | Granada | 200        | 12 | 3045.6  | 2017-04-25 | 2          | 1            |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

-- 1.3.7.2 Subconsultas con ALL y ANY

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

```
select *
from pedido
where total >= all (select total
                    from pedido);
```

```
+-----+-----+-----+-----+-----+
| id | total | fecha      | id_cliente | id_comercial |
+-----+-----+-----+-----+-----+
| 7  | 5760  | 2015-09-10 | 2          | 1            |
+-----+-----+-----+-----+-----+
```

-- 07.- 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|
+---+-----+-----+-----+-----+
| 9 |Guillermo|López    |Gómez    |Granada|225      |
|10|Daniel   |Santana  |Loyola    |Sevilla|125      |
+---+-----+-----+-----+-----+

```

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

```

select *
from comercial
where id <> all (select id_comercial
                 from pedido);

```

```

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

```

-- 1.3.7.3 Subconsultas con IN y NOT IN

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

```

select *
from cliente
where 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      |
+---+-----+-----+-----+-----+

```

-- 10.- 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|
+---+-----+-----+-----+-----+
| 4 |Marta   |Herrera  |Gil       |0.14    |
| 8 |Alfredo|Ruiz      |Flores    |0.05    |
+---+-----+-----+-----+-----+

```

-- 1.3.7.4 Subconsultas con EXISTS y NOT EXISTS

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

```

select *
from cliente
where not exists (select id_cliente
                  from pedido
                  where cliente.id = id_cliente);

```

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

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

```

select *
from comercial
where not exists (select id_comercial
                  from pedido
                  where comercial.id = id_comercial);

```

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