

Josue David Gelvez Jurado-02210131011

## Taller 3 P3 SGBD

### Gestión de Ventas

- 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.

```
SELECT * FROM pedido p
```

```
order by fecha DESC
```

Enter a SQL expression to filter results (use Ctrl+Space)						
Grilla	id	total	fecha	id_cliente	id_comercial	
1	16	2.389,23	2019-03-11	1	5	
2	15	370,85	2019-03-11	1	5	
3	13	545,75	2019-01-25	6	1	
4	8	1.983,43	2017-10-10	4	6	
5	1	150,5	2017-10-05	5	2	
6	3	65,26	2017-10-05	2	1	
7	5	948,5	2017-09-10	5	2	
8	12	3.045,6	2017-04-25	2	1	
9	14	145,82	2017-02-02	6	1	
10	9	2.480,4	2016-10-10	8	3	
11	2	270,65	2016-09-10	1	5	
12	11	75,29	2016-08-17	3	7	
13	4	110,5	2016-08-17	8	3	
14	6	2.400,6	2016-07-27	7	1	
15	7	5.760	2015-09-10	2	1	
16	10	250,45	2015-06-27	8	2	

2. Devuelve todos los datos de los dos pedidos de mayor valor.

```
select *
```

```
FROM
```

```
pedido p
```

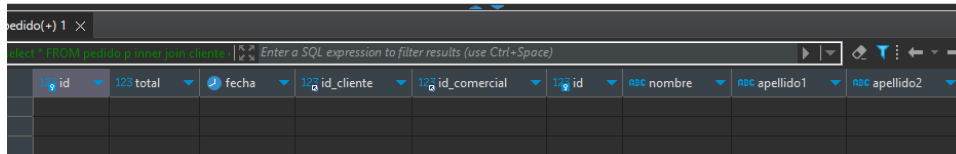
```
order by total DESC
```

```
limit 2
```

Enter a SQL expression to filter results (use Ctrl+Space)						
Grilla	id	total	fecha	id_cliente	id_comercial	
1	7	5.760	2015-09-10	2	1	
2	12	3.045,6	2017-04-25	2	1	

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.

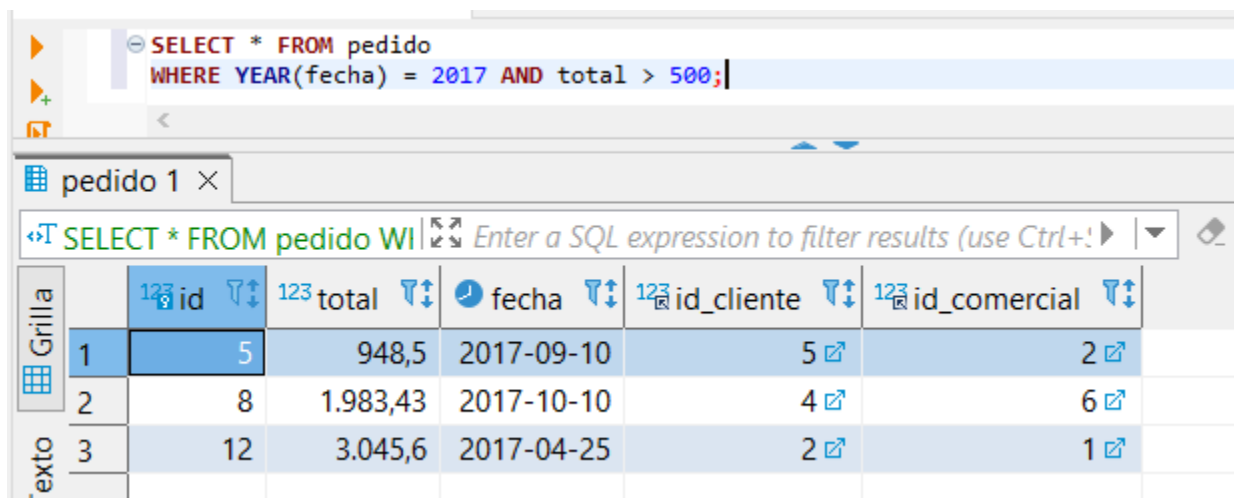
```
select *  
FROM pedido p  
inner join cliente c on p.id_cliente = c.id  
where p.id_cliente is null
```



id	total	fecha	id_cliente	id_comercial	id	nombre	apellido1	apellido2
----	-------	-------	------------	--------------	----	--------	-----------	-----------

4. 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 YEAR(fecha) = 2017 AND total > 500;
```



id	total	fecha	id_cliente	id_comercial
5	948,5	2017-09-10	5	2
8	1.983,43	2017-10-10	4	6
12	3.045,6	2017-04-25	2	1

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

```
SELECT c.nombre, c.apellido1, c.apellido2  
FROM comercial c  
WHERE c.comisión BETWEEN 0.05 AND 0.11;
```

SELECT c.nombre, c.apellido1, c.apellido2  
FROM comercial c  
WHERE c.comisión BETWEEN 0.05 AND 0.11;

comercial 1 ×

SELECT c.nombre, c.apellido1

	nombre	apellido1	apellido2
1	Diego	Flores	Salas
2	Antonio	Vega	Hernández
3	Alfredo	Ruiz	Flores

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

```
SELECT MAX(c.comisión) as mayor_valor_comision
from comercial c
```

SELECT MAX(c.comisión) as mayor\_valor\_comision  
from comercial c

Resultados 1 ×

SELECT MAX(c.comisión) as

	mayor_valor_comision
1	0,150000006

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.

```
SELECT c.id , c.nombre ,c.apellido1
from cliente c
where c.apellido2 is not NULL
order by c.apellido1, c.nombre
```

cliente 1 x

SELECT c.id , c.nombre , c.a Enter a SQL expression to fil

Grilla	id	nombre	apellido1
1	9	Guillermo	López
2	5	Marcos	Loyola
3	1	Aarón	Rivero
4	3	Adolfo	Rubio
5	8	Pepe	Ruiz
6	2	Adela	Salas
7	10	Daniel	Santana
8	6	María	Santana

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.

```
SELECT c.nombre
FROM cliente c
WHERE (c.nombre LIKE 'A%n') OR (c.nombre LIKE 'P%')
ORDER BY c.nombre;
```

cliente 1 x

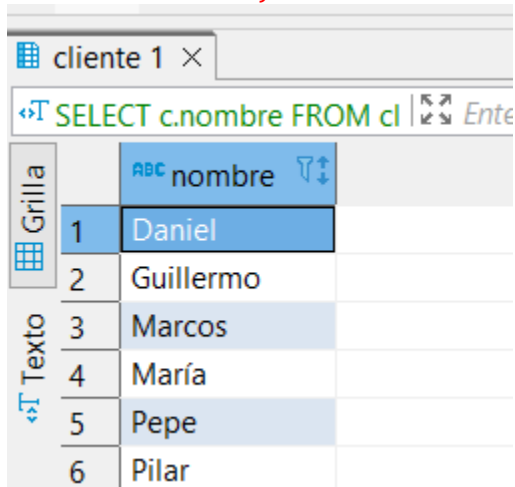
SELECT c.nombre FROM cl Enter a SQL

Grilla	nombre
1	Aarón
2	Adrián
3	Pepe
4	Pilar

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

```
SELECT c.nombre
FROM cliente c
WHERE NOT (c.nombre LIKE 'A%')
```

**ORDER BY** c.nombre;

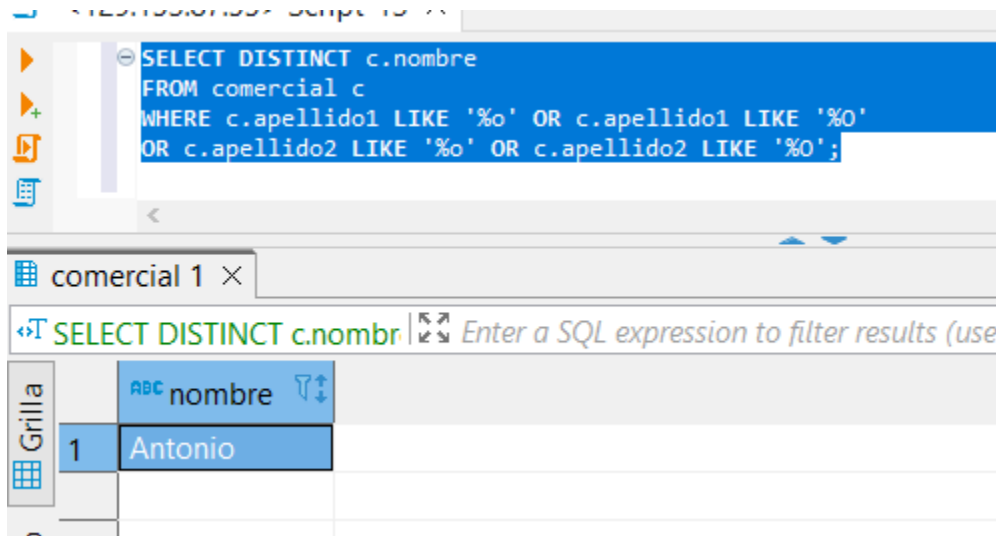


The screenshot shows a database client window titled 'cliente 1'. The SQL query entered is 'SELECT c.nombre FROM cl'. The result is displayed in a table with 6 rows. The first row is highlighted in blue.

	nombre
1	Daniel
2	Guillermo
3	Marcos
4	María
5	Pepe
6	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 c.nombre
FROM comercial c
WHERE c.apellido1 LIKE '%o' OR c.apellido1 LIKE '%O'
OR c.apellido2 LIKE '%o' OR c.apellido2 LIKE '%O';
```



The screenshot shows a database client window titled 'comercial 1'. The SQL query entered is 'SELECT DISTINCT c.nombre FROM comercial c WHERE c.apellido1 LIKE '%o' OR c.apellido1 LIKE '%O' OR c.apellido2 LIKE '%o' OR c.apellido2 LIKE '%O';'. The result is displayed in a table with 1 row. The first row is highlighted in blue.

	nombre
1	Antonio

### Consultas multitable

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 c.id, c.nombre, c.apellido1, c.apellido2
FROM cliente c
inner JOIN pedido p ON c.id = p.id_cliente
ORDER BY c.apellido1, c.nombre;
```

129.155.07.55 Script 13

```
SELECT DISTINCT c.id, c.nombre, c.apellido1, c.apellido2
FROM cliente c
inner JOIN pedido p ON c.id = p.id_cliente
ORDER BY c.apellido1, c.nombre;
```

cliente 1 x

SELECT DISTINCT c.id, c.no Enter a SQL expression to filter results (use Ctrl+.)

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

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.

```
SELECT c.id AS cliente_id, c.nombre AS cliente_nombre, c.apellido1 AS
cliente_apellido1,
c.apellido2 AS cliente_apellido2, p.id AS pedido_id, p.total AS pedido_total,
p.fecha AS pedido_fecha
FROM cliente c
INNER JOIN pedido p ON c.id = p.id_cliente
ORDER BY c.nombre, p.fecha;
```

SELECT c.id AS cliente\_id, c.nombre AS cliente\_nombre, c.apellido1 AS cliente\_apellido1, c.apellido2 AS cliente\_apellido2, p.id AS pedido\_id, p.total AS pedido\_total, p.fecha AS pedido\_fecha  
FROM cliente c  
INNER JOIN pedido p ON c.id = p.id\_cliente  
ORDER BY c.nombre, p.fecha;

cliente(+) 1 x

SELECT c.id AS cliente\_id, Enter a SQL expression to filter results (use Ctrl+Space)

	cliente_id	cliente_nombre	cliente_apellido1	cliente_apellido2	pedido_id	pedido_total	pedido_fecha
1	1	Aarón	Rivero	Gómez	2	270,65	2016-09-10
2	1	Aarón	Rivero	Gómez	16	2.389,23	2019-03-11
3	1	Aarón	Rivero	Gómez	15	370,85	2019-03-11
4	2	Adela	Salas	Díaz	7	5.760	2015-09-10
5	2	Adela	Salas	Díaz	12	3.045,6	2017-04-25
6	2	Adela	Salas	Díaz	3	65,26	2017-10-05
7	3	Adolfo	Rubio	Flores	11	75,29	2016-08-17
8	4	Adrián	Suárez	[NULL]	8	1.983,43	2017-10-10
9	5	Marcos	Loyola	Méndez	5	948,5	2017-09-10
10	5	Marcos	Loyola	Méndez	1	150,5	2017-10-05
11	6	María	Santana	Moreno	14	145,82	2017-02-02
12	6	María	Santana	Moreno	13	545,75	2019-01-25
13	8	Pepe	Ruiz	Santana	10	250,45	2015-06-27
14	8	Pepe	Ruiz	Santana	4	110,5	2016-08-17
15	8	Pepe	Ruiz	Santana	9	2.480,4	2016-10-10
16	7	Pilar	Ruiz	[NULL]	6	2.400,6	2016-07-27

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.

```
SELECT pedido.id, pedido.total, pedido.fecha, c.nombre, c.apellido1,
c.apellido2, c2.nombre, c2.apellido1, c2.apellido2, c2.comisión
FROM pedido
INNER JOIN cliente c ON pedido.id_cliente = c.id
INNER JOIN comercial c2 ON pedido.id_comercial = c2.id
ORDER BY c2.nombre, c2.apellido1, c2.apellido2
```

SELECT pedido.id, pedido.total, pedido.fecha, c.nombre, c.apellido1, c.apellido2, c2.nombre, c2.apellido1, c2.apellido2, c2.comisión  
FROM pedido  
INNER JOIN cliente c ON pedido.id\_cliente = c.id  
INNER JOIN comercial c2 ON pedido.id\_comercial = c2.id  
ORDER BY c2.nombre, c2.apellido1, c2.apellido2

pedido(+) 1 ×

SELECT pedido.id, pedido. Enter a SQL expression to filter results (use Ctrl+!)

	id	total	fecha	nombre	apellido1	apellido2	nombre
1	2	270,65	2016-09-10	Aarón	Rivero	Gómez	Antonio
2	15	370,85	2019-03-11	Aarón	Rivero	Gómez	Antonio
3	16	2.389,23	2019-03-11	Aarón	Rivero	Gómez	Antonio
4	11	75,29	2016-08-17	Adolfo	Rubio	Flores	Antonio
5	3	65,26	2017-10-05	Adela	Salas	Díaz	Daniel
6	6	2.400,6	2016-07-27	Pilar	Ruiz	[NULL]	Daniel
7	7	5.760	2015-09-10	Adela	Salas	Díaz	Daniel
8	12	3.045,6	2017-04-25	Adela	Salas	Díaz	Daniel
9	13	545,75	2019-01-25	María	Santana	Moreno	Daniel
10	14	145,82	2017-02-02	María	Santana	Moreno	Daniel
11	4	110,5	2016-08-17	Pepe	Ruiz	Santana	Diego
12	9	2.480,4	2016-10-10	Pepe	Ruiz	Santana	Diego
13	1	150,5	2017-10-05	Marcos	Loyola	Méndez	Juan
14	5	948,5	2017-09-10	Marcos	Loyola	Méndez	Juan
15	10	250,45	2015-06-27	Pepe	Ruiz	Santana	Juan
16	8	1.983,43	2017-10-10	Adrián	Suárez	[NULL]	Manuel

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.

```
SELECT cliente.id, cliente.nombre, cliente.apellido1, cliente.apellido2,
pedido.id as id_pedido, pedido.total, pedido.fecha, comercial.nombre as
nombre_comercial,
comercial.apellido1 as apellido1_comercial, comercial.apellido2 as
apellido2_comercial,comercial.comisión
FROM cliente
LEFT JOIN pedido ON cliente.id = pedido.id_cliente
LEFT JOIN comercial ON pedido.id_comercial = comercial.id
ORDER BY cliente.nombre, cliente.apellido1, cliente.apellido2, pedido.fecha DESC
```



SELECT cliente.id, cliente.n							
	id	nombre	apellido1	apellido2	id_pedido	total	fecha
1	1	Aarón	Rivero	Gómez	15	370,85	2019-0
2	1	Aarón	Rivero	Gómez	16	2.389,23	2019-0
3	1	Aarón	Rivero	Gómez	2	270,65	2016-0
4	2	Adela	Salas	Díaz	3	65,26	2017-1
5	2	Adela	Salas	Díaz	12	3.045,6	2017-0
6	2	Adela	Salas	Díaz	7	5.760	2015-0
7	3	Adolfo	Rubio	Flores	11	75,29	2016-0
8	4	Adrián	Suárez	[NULL]	8	1.983,43	2017-1
9	10	Daniel	Santana	Loyola	[NULL]	[NULL]	[N
10	9	Guillermo	López	Gómez	[NULL]	[NULL]	[N
11	5	Marcos	Loyola	Méndez	1	150,5	2017-1
12	5	Marcos	Loyola	Méndez	5	948,5	2017-0
13	6	María	Santana	Moreno	13	545,75	2019-0
14	6	María	Santana	Moreno	14	145,82	2017-0
15	8	Pepe	Ruiz	Santana	9	2.480,4	2016-1
16	8	Pepe	Ruiz	Santana	4	110,5	2016-0
17	8	Pepe	Ruiz	Santana	10	250,45	2015-0
18	7	Pilar	Ruiz	[NULL]	6	2.400,6	2016-0

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

```
SELECT c.id, c.nombre, c.apellido1, c.apellido2
FROM cliente c
INNER JOIN pedido p ON c.id = p.id_cliente
WHERE YEAR(p.fecha) = 2017 AND p.total BETWEEN 300 AND 1000
ORDER BY c.apellido1, c.apellido2, c.nombre;
```

SELECT c.id, c.nombre, c.ap				
	id	nombre	apellido1	apellido2
1	5	Marcos	Loyola	Méndez

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

```
SELECT DISTINCT c.nombre, c.apellido1, c.apellido2
FROM comercial c
```

```

INNER JOIN pedido p ON c.id = p.id_comercial
INNER JOIN cliente cl ON p.id_cliente = cl.id
WHERE cl.nombre = 'María' AND cl.apellido1 = 'Santana' AND cl.apellido2 = 'Moreno'

```

comercial 1 x

SELECT DISTINCT c.nombre | Enter a SQL expression to filter result

	ABC nombre	ABC apellido1	ABC apellido2
1	Daniel	Sáez	Vega

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

```

SELECT DISTINCT c.nombre
FROM cliente c
inner JOIN pedido p ON c.id = p.id_cliente
inner JOIN comercial co ON co.id = p.id_comercial
WHERE co.nombre = 'Daniel' AND co.apellido1 = 'Sáez' AND co.apellido2 = 'Vega';

```

cliente 1 x

SELECT DISTINCT c.nombre | Enter c

	ABC nombre
1	Adela
2	Pilar
3	María

8. 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 c.nombre, c.apellido1, c.apellido2, p.id, p.total, p.fecha
FROM cliente c

```

```
LEFT JOIN pedido p ON c.id = p.id_cliente
ORDER BY c.apellido1, c.apellido2, c.nombre
```

cliente(+) 1 ×

</

9. 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 c.id, c.nombre, c.apellido1, c.apellido2, p.id as id_pedido, p.total, p.fecha,
cl.nombre as nombre_cliente, cl.apellido1 as apellido1_cliente, cl.apellido2 as
apellido2_cliente
FROM comercial c
LEFT JOIN pedido p ON p.id_comercial = c.id
LEFT JOIN cliente cl ON cl.id = p.id_cliente
ORDER BY c.apellido1, c.apellido2, c.nombre
```

LEFT JOIN cliente cl ON cl.id = p.id\_cliente

comercial(+) 1 x

SELECT c.id, c.nombre, c.apellido1, c.apellido2, p.id\_pedido, total, fecha

	id	nombre	apellido1	apellido2	id_pedido	total	fecha
1	5	Antonio	Carretero	Ortega	2	270,65	2016-0
2	5	Antonio	Carretero	Ortega	15	370,85	2019-0
3	5	Antonio	Carretero	Ortega	16	2.389,23	2019-0
4	6	Manuel	Domínguez	Hernández	8	1.983,43	2017-1
5	3	Diego	Flores	Salas	4	110,5	2016-0
6	3	Diego	Flores	Salas	9	2.480,4	2016-1
7	2	Juan	Gómez	López	1	150,5	2017-1
8	2	Juan	Gómez	López	5	948,5	2017-0
9	2	Juan	Gómez	López	10	250,45	2015-0
10	4	Marta	Herrera	Gil	[NULL]	[NULL]	[N
11	8	Alfredo	Ruiz	Flores	[NULL]	[NULL]	[N
12	1	Daniel	Sáez	Vega	3	65,26	2017-1
13	1	Daniel	Sáez	Vega	6	2.400,6	2016-0
14	1	Daniel	Sáez	Vega	7	5.760	2015-0
15	1	Daniel	Sáez	Vega	12	3.045,6	2017-0
16	1	Daniel	Sáez	Vega	13	545,75	2019-0
17	1	Daniel	Sáez	Vega	14	145,82	2017-0
18	7	Antonio	Vega	Hernández	11	75,29	2016-0

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

```
SELECT c.id, c.nombre, c.apellido1, c.apellido2
FROM cliente c
LEFT JOIN pedido p ON c.id = p.id_cliente
WHERE p.id IS NULL
ORDER BY c.apellido1, c.apellido2, c.nombre;
```

cliente 1 x

SELECT c.id, c.nombre, c.apellido1, c.apellido2

	id	nombre	apellido1	apellido2
1	9	Guillermo	López	Gómez
2	10	Daniel	Santana	Loyola

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

```
SELECT id, nombre, apellido1, apellido2, comisión
FROM comercial
WHERE id NOT IN (SELECT DISTINCT id_comercial FROM pedido)
ORDER BY apellido1, apellido2, nombre;
```

comercial 1 ×

Enter a SQL expression to filter results (use Ctrl+!) |

	123 id	ABC nombre	ABC apellido1	ABC apellido2	123 comisión
1	4	Marta	Herrera	Gil	0,14
2	8	Alfredo	Ruiz	Flores	0,05

12. 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 tipo, id, nombre, apellido1, apellido2
FROM cliente
WHERE id NOT IN (SELECT DISTINCT id_cliente FROM pedido)
UNION SELECT 'Comercial' AS tipo, id, nombre, apellido1, apellido2
FROM comercial
WHERE id NOT IN (SELECT DISTINCT id_comercial FROM pedido)
ORDER BY apellido1, apellido2, nombre;
```

Resultados 1 ×

Enter a SQL expression to filter results (use Ctrl+!) |

	ABC tipo	123 id	ABC nombre	ABC apellido1	ABC apellido2
1	Comercial	4	Marta	Herrera	Gil
2	Cliente	9	Guillermo	López	Gómez
3	Comercial	8	Alfredo	Ruiz	Flores
4	Cliente	10	Daniel	Santana	Loyola

#### • Consultas resumen

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

```
SELECT SUM(total) AS total_valor_pedidos
FROM pedido;
```

Resultados 1 x

SELECT SUM(total) AS tota | Enter a SQL ex

Grilla	123 total_valor_pedidos
1	20.992,83

Texto

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

```
SELECT AVG(total) AS media
FROM pedido;
```

Resultados 1 x

SELECT AVG(total) AS med | En

Grilla	123 media
1	1.312,051875

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

```
SELECT COUNT(DISTINCT id_comercial) AS total_comer
FROM pedido;
```

Resultados 1 x

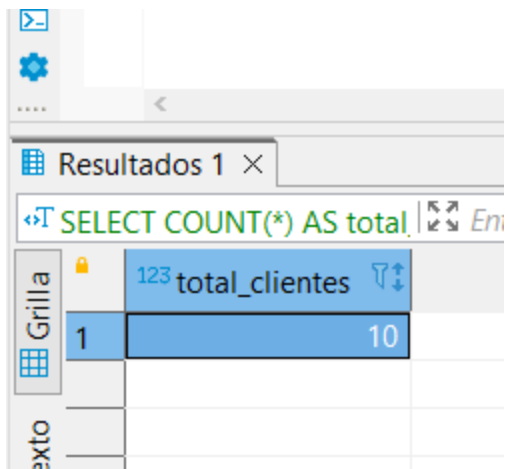
SELECT COUNT(DISTINCT i |

Grilla	123 total_comer
1	6

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

```
SELECT COUNT(*) AS total_clientes
```

```
FROM cliente;
```



Resultados 1

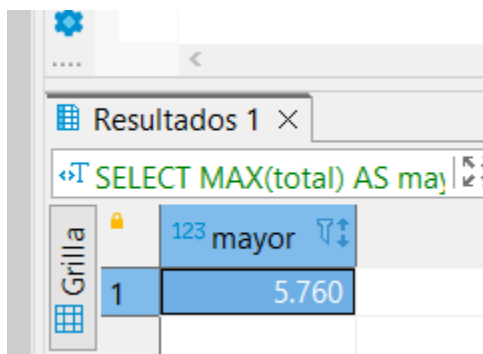
SELECT COUNT(\*) AS total

Grilla	total_clientes
1	10

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

SELECT MAX(total) AS mayor

FROM pedido;



Resultados 1

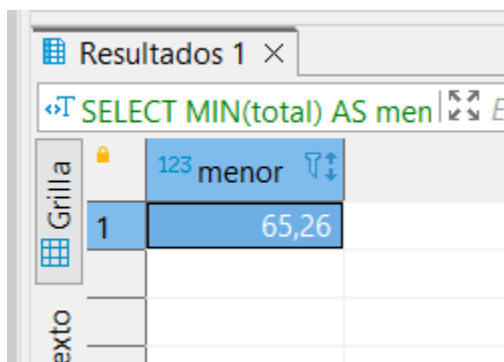
SELECT MAX(total) AS mayor

Grilla	mayor
1	5.760

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

SELECT MIN(total) AS menor\_cantidad

FROM pedido;



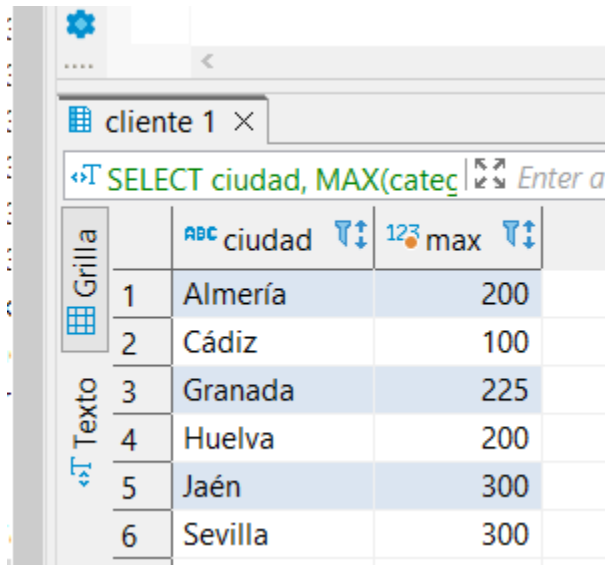
Resultados 1

SELECT MIN(total) AS menor

Grilla	menor
1	65,26

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

```
SELECT ciudad, MAX(categoría) AS max
FROM cliente
GROUP BY ciudad;
```



The screenshot shows a database application window titled 'cliente 1'. The query editor displays the SQL query: `SELECT ciudad, MAX(categoría) AS max`. Below the editor, a table with 3 columns is shown. The first column is a numeric index (1-6), the second column is 'ciudad' (Almería, Cádiz, Granada, Huelva, Jaén, Sevilla), and the third column is 'max' (200, 100, 225, 200, 300, 300). The table is displayed in a grid view with a sidebar on the left showing 'Grilla' and 'Texto' options.

	ciudad	max
1	Almería	200
2	Cádiz	100
3	Granada	225
4	Huelva	200
5	Jaén	300
6	Sevilla	300

8. 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 c.id AS id_cliente, c.nombre, c.apellido1, c.apellido2, p.fecha, MAX(p.total) AS
maximo_pedido
FROM cliente c
LEFT JOIN pedido p ON c.id = p.id_cliente
GROUP BY c.id, p.fecha;
```



cliente(+) 1 x

SELECT c.id AS id\_cliente, c.nombre, c.apellido1, c.apellido2, p.fecha, MAX(p.total) AS maximo\_pedido

	id_cliente	nombre	apellido1	apellido2	fecha	maximo_pedido
1	1	Aarón	Rivero	Gómez	2016-09-10	2
2	1	Aarón	Rivero	Gómez	2019-03-11	2.3
3	2	Adela	Salas	Díaz	2015-09-10	
4	2	Adela	Salas	Díaz	2017-04-25	3.
5	2	Adela	Salas	Díaz	2017-10-05	
6	3	Adolfo	Rubio	Flores	2016-08-17	
7	4	Adrián	Suárez	[NULL]	2017-10-10	1.9
8	5	Marcos	Loyola	Méndez	2017-09-10	
9	5	Marcos	Loyola	Méndez	2017-10-05	
10	6	María	Santana	Moreno	2017-02-02	1
11	6	María	Santana	Moreno	2019-01-25	5
12	7	Pilar	Ruiz	[NULL]	2016-07-27	2.
13	8	Pepe	Ruiz	Santana	2015-06-27	2
14	8	Pepe	Ruiz	Santana	2016-08-17	
15	8	Pepe	Ruiz	Santana	2016-10-10	2.
16	9	Guillermo	López	Gómez	[NULL]	[↑
17	10	Daniel	Santana	Loyola	[NULL]	[↑

9. 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 p.id_cliente, c.nombre, c.apellido1, c.apellido2, p.fecha, MAX(p.total) AS
max_valor
FROM pedido p
JOIN cliente c ON p.id_cliente = c.id
WHERE (p.fecha, p.id_cliente, p.total) IN (
    SELECT fecha, id_cliente, MAX(total)
    FROM pedido
    WHERE total > 2000
    GROUP BY fecha, id_cliente
)
GROUP BY p.fecha, p.id_cliente;
```

SELECT p.id_cliente, c.nom							
Grilla	id_cliente	nombre	apellido1	apellido2	fecha	max_valor	
1	2	Adela	Salas	Díaz	2015-09-10	5.760	
2	7	Pilar	Ruiz	[NULL]	2016-07-27	2.400,6	2
3	8	Pepe	Ruiz	Santana	2016-10-10	2.480,4	Dicti
4	2	Adela	Salas	Díaz	2017-04-25	3.045,6	Val.
5	1	Aarón	Rivero	Gómez	2019-03-11	2.389,23	1

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 p.id_comercial, c.nombre, c.apellido1, c.apellido2, MAX(p.total) AS max_valor
FROM pedido p
JOIN comercial c ON p.id_comercial = c.id
WHERE p.fecha = '2016-08-17'
GROUP BY p.id_comercial;
```

SELECT p.id_comercial, c.n							
id_comercial	nombre	apellido1	apellido2	max_valor			
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 c.id, c.nombre, c.apellido1, c.apellido2, COUNT(p.id) AS num_pedidos
FROM cliente c
LEFT JOIN pedido p ON c.id = p.id_cliente
GROUP BY c.id, c.nombre, c.apellido1, c.apellido2;
```

cliente 1 x

SELECT c.id, c.nombre, c.apellido1, c.apellido2, COUNT(p.id) AS num\_pedidos

	id	nombre	apellido1	apellido2	num_pedidos
1	1	Aarón	Rivero	Gómez	3
2	2	Adela	Salas	Díaz	3
3	3	Adolfo	Rubio	Flores	1
4	4	Adrián	Suárez	[NULL]	1
5	5	Marcos	Loyola	Méndez	2
6	6	María	Santana	Moreno	2
7	7	Pilar	Ruiz	[NULL]	1
8	8	Pepe	Ruiz	Santana	3
9	9	Guillermo	López	Gómez	0
10	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 c.id, c.nombre, c.apellido1, c.apellido2, COUNT(p.id) AS num_pedidos
FROM cliente c
LEFT JOIN pedido p ON c.id = p.id_cliente
WHERE YEAR(p.fecha) = 2017
GROUP BY c.id, c.nombre, c.apellido1, c.apellido2;
```

cliente 1 x

SELECT c.id, c.nombre, c.apellido1, c.apellido2, COUNT(p.id) AS num\_pedidos

	id	nombre	apellido1	apellido2	num_pedidos
1	2	Adela	Salas	Díaz	2
2	4	Adrián	Suárez	[NULL]	1
3	5	Marcos	Loyola	Méndez	2
4	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 c.id, c.nombre, c.apellido1, IFNULL(MAX(p.total), 0) AS max_cantidad
FROM cliente c
LEFT JOIN pedido p ON c.id = p.id_cliente
GROUP BY c.id, c.nombre, c.apellido1;
```

cliente 1 x

SELECT c.id, c.nombre, c.apellido1, c.apellido2, c.max\_cantidad

	id	nombre	apellido1	max_cantidad
1	1	Aarón	Rivero	2.389,23
2	2	Adela	Salas	5.760
3	3	Adolfo	Rubio	75,29
4	4	Adrián	Suárez	1.983,43
5	5	Marcos	Loyola	948,5
6	6	María	Santana	545,75
7	7	Pilar	Ruiz	2.400,6
8	8	Pepe	Ruiz	2.480,4
9	9	Guillermo	López	0
10	10	Daniel	Santana	0

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

```
SELECT YEAR(fecha) AS año, MAX(total) AS maximo
FROM pedido
GROUP BY YEAR(fecha);
```

Resultados 1 x

SELECT YEAR(fecha) AS año, MAX(total) AS maximo

	año	maximo
1	2015	5.760
2	2016	2.480,4
3	2017	3.045,6
4	2019	2.389,23

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

```
SELECT YEAR(fecha) AS año, COUNT(*) AS total_pedidos
```

```
FROM pedido
GROUP BY YEAR(fecha);
```

Resultados 1 x

SELECT YEAR(fecha) AS añ | Enter a SQL express

Grilla	añ	total_pedidos
1	2.015	2
2	2.016	5
3	2.017	6
4	2.019	3

### Subconsultas

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

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

pedido 1 x

SELECT \* FROM pedido WI | Enter a SQL expression to filter results (use Ctrl+!) |

Grilla	id	total	fecha	id_cliente	id_comercial
1	3	65,26	2017-10-05	2	1
2	7	5.760	2015-09-10	2	1
3	12	3.045,6	2017-04-25	2	1

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

```
SELECT COUNT(*)FROM pedido
WHERE id_comercial = (
    SELECT id
    FROM comercial
```

**WHERE** nombre = 'Daniel' **AND** apellido1 = 'Sáez' **AND** apellido2 = 'Vega')

	COUNT(*)
1	6

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

```
SELECT id, nombre, apellido1, apellido2 FROM cliente
WHERE id = (SELECT id_cliente
FROM pedido
WHERE total = (
SELECT MAX(total)
FROM pedido
WHERE YEAR(fecha) = 2019))
```

	id	nombre	apellido1	apellido2
1	1	Aarón	Rivero	Gómez

4. Devuelve la fecha y la cantidad del pedido de menor valor realizado por el cliente Pepe Ruiz Santana.

```
SELECT fecha, total FROM pedido
WHERE id_cliente = (SELECT id
FROM cliente
WHERE nombre = 'Pepe'
AND apellido1 = 'Ruiz'
AND apellido2 = 'Santana')
ORDER BY total ASC
LIMIT 1;
```

ORDER BY total ASC  
LIMIT 1;

pedido 1 ×

SELECT fecha, total FROM

	fecha	total
1	2016-08-17	110,5

5. 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 c.id, c.nombre, c.apellido1, c.apellido2, p.id AS id_pedido, p.total, p.fecha
FROM cliente c, pedido p
WHERE c.id = p.id_cliente AND YEAR(p.fecha) = 2017
AND p.total >= (SELECT AVG(total) FROM pedido WHERE YEAR(fecha) = 2017)
ORDER BY c.id, p.fecha;
```

cliente(+) 1 ×

SELECT c.id, c.nombre, c.apellido1, c.apellido2, p.id AS id\_pedido, p.total, p.fecha

	id	nombre	apellido1	apellido2	id_pedido	total	fecha
1	2	Adela	Salas	Díaz	12	3.045,6	2017-0
2	4	Adrián	Suárez	[NULL]	8	1.983,43	2017-1

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

```
SELECT id, total
FROM pedido WHERE total >= ALL (SELECT total FROM pedido);
```

pedido 1 ×

SELECT id, total FROM ped

	id	total
1	7	5.760

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

```
SELECT id, nombre, apellido1, apellido2
FROM cliente
WHERE id <> ALL (SELECT DISTINCT id_cliente FROM pedido)
```

cliente 1 ×

SELECT id, nombre, apellid

	id	nombre	apellido1	apellido2
1	9	Guillermo	López	Gómez
2	10	Daniel	Santana	Loyola

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

```
SELECT * FROM comercial
WHERE NOT EXISTS (
  SELECT 1 FROM pedido
  WHERE pedido.id_comercial = comercial.id
);
```

comercial 1 ×

SELECT \* FROM comercial

	id	nombre	apellido1	apellido2	comisión
1	4	Marta	Herrera	Gil	0,14
2	8	Alfredo	Ruiz	Flores	0,05



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

```
SELECT id, nombre, apellido1, apellido2
FROM cliente
WHERE id NOT IN (SELECT id_cliente FROM pedido);
```

SELECT id, nombre, apellido1, apellido2 | Enter a SQL expression to filter results (use Ctrl+)

	id	nombre	apellido1	apellido2
1	9	Guillermo	López	Gómez
2	10	Daniel	Santana	Loyola

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

```
SELECT id, nombre, apellido1, apellido2
FROM comercial
WHERE id NOT IN (SELECT DISTINCT id_comercial FROM pedido);
```

comercial 1 x

SELECT id, nombre, apellido1, apellido2 | Enter a SQL expression to filter results (use Ctrl+)

	id	nombre	apellido1	apellido2
1	4	Marta	Herrera	Gil
2	8	Alfredo	Ruiz	Flores

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

```
SELECT id, nombre, apellido1, apellido2
FROM cliente WHERE NOT EXISTS (
    SELECT * FROM pedido
    WHERE cliente.id = pedido.id_cliente
);
```

cliente 1 ×					
SELECT id, nombre, apellido   Enter a SQL expression to filter results (use Ctrl-					
Grilla	id	nombre	apellido1	apellido2	
1	9	Guillermo	López	Gómez	
2	10	Daniel	Santana	Loyola	
exto					

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

```
SELECT id, nombre, apellido1, apellido2, comisión
FROM comercial c
WHERE NOT EXISTS (
    SELECT 1 FROM pedido p
    WHERE p.id_comercial = c.id
);
```

comercial 1 ×						
SELECT id, nombre, apellido   Enter a SQL expression to filter results (use Ctrl+!   ▾ 🔍						
Grilla	id	nombre	apellido1	apellido2	comisión	
1	4	Marta	Herrera	Gil	0,14	
2	8	Alfredo	Ruiz	Flores	0,05	
exto						