

Enunciado

1. Crear la Base de Datos “ventas” utilizando

- a) Google FireBase o
- b) Microsoft Azure

Microsoft Azure

Actualización

Buscar recursos, servicios y documentación

Inicio > Servidores SQL Server > serverbasebrayan >

Crear base de datos SQL

Microsoft

Detalles de la base de datos

Indique la configuración necesaria para esta base de datos, incluida la selección de un servidor lógico y la configuración de los recursos de proceso y almacenamiento.

Nombre de la base de datos *

ventasbrayan

Servidor

serverbasebrayan (Brazil South)

¿Quiere usar un grupo elástico de SQL?

Sí

No

Entorno de carga de trabajo

Implementación

Producción

Configuración predeterminada proporcionada para las cargas de trabajo de Producción. Las configuraciones se pueden modificar según sea necesario.

Proceso y almacenamiento *

Uso general

Serie estándar (Gen 5), 2 Núcleos virtuales, Almacenamiento: 32 GB, redundancia de zona deshabilitada

Configurar base de datos

Redundancia del almacenamiento de copias de seguridad

Elija el modo de replicación de las copias de seguridad de PITR y LTR. La restauración geográfica o la posibilidad de recuperación tras una interrupción regional solo están disponibles si se ha seleccionado el almacenamiento con redundancia geográfica.

Redundancia de almacenamiento de copia de seguridad

Almacenamiento de copias de seguridad con redundancia local

Almacenamiento de copias de seguridad con redundancia de zona

Almacenamiento de copias de seguridad con redundancia regional

Revisar y crear

Siguiente: Redes >

Inicio > Servidores SQL Server > serverbasebrayan >

Crear base de datos SQL

Microsoft

Identidad administrada asignada por el usuario en el nivel de base de datos

No configurado

Advanced Data Security

Ahora no

Always encrypted with secure enclaves

No configurado

Libro de contabilidad Sql(base de datos)

Deshabilitado

Almacenamiento de resumen

Deshabilitado

Configuración adicional

Usar datos existentes

Blank

Intercalación

SQL_Latin1_General_CP1_CI_AS

Ventana de mantenimiento

Valor predeterminado del sistema (de 17:00 a 8

Etiquetas

SQL

Resumen del costo

Básico (Basic)

Costo por DTU (en USD)

Selección: DTU

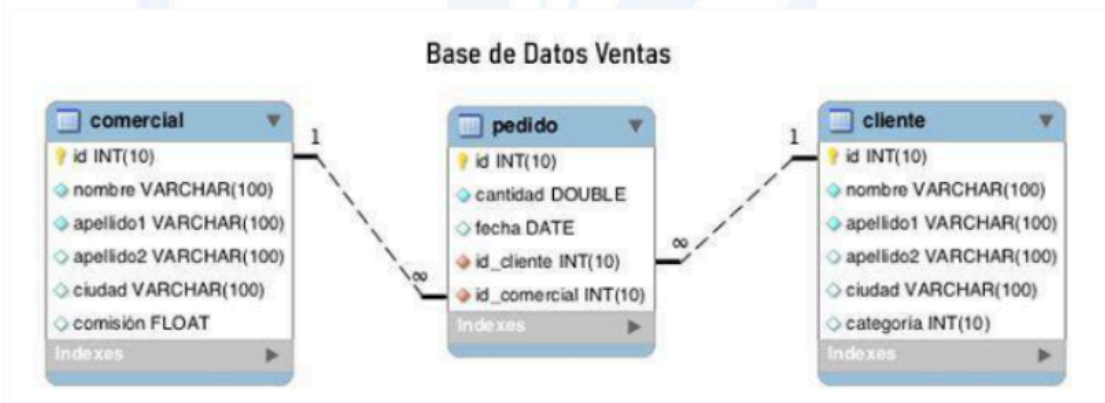
COSTO MENSUAL ESTIMADO

1.19

x 5

5.93 USD

2. Crear las tablas Comercial, pedido y cliente para la base de datos "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),  
    comision 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)  
);
```

Consulta 1 ✕ Consulta 2 ✕

Ejecutar

☐ Cancelar consulta

Guardar consulta

Exportar datos como ▾

Mostrar solo editor

Abrir Copilot

```
1  --CREAR TABLAS
2  -- Crear la tabla cliente
3  CREATE TABLE cliente (
4      id INT IDENTITY(1,1) PRIMARY KEY, -- id del cliente, clave primaria y autoincremental
5      nombre VARCHAR(100) NOT NULL, -- nombre del cliente, no puede ser nulo
6      apellido1 VARCHAR(100) NOT NULL, -- primer apellido, obligatorio
7      apellido2 VARCHAR(100), -- segundo apellido, opcional
8      ciudad VARCHAR(100), -- ciudad del cliente, opcional
9      categoria INT -- categoría del cliente
10 );
11
-- Crear la tabla comercial
CREATE TABLE comercial (
    id INT IDENTITY(1,1) PRIMARY KEY, -- id del comercial, clave primaria y autoincremental
    nombre VARCHAR(100) NOT NULL, -- nombre del comercial, no puede ser nulo
    apellido1 VARCHAR(100) NOT NULL, -- primer apellido, obligatorio
    apellido2 VARCHAR(100), -- segundo apellido, opcional
    comision FLOAT -- comisión del comercial, valor de tipo flotante
);

-- Crear la tabla pedido
CREATE TABLE pedido (
    id INT IDENTITY(1,1) PRIMARY KEY, -- id del pedido, clave primaria y autoincremental
    total FLOAT NOT NULL, -- total del pedido, obligatorio
    fecha DATE, -- fecha del pedido
    id_cliente INT NOT NULL, -- id del cliente que realizó el pedido, referencia a cliente
    id_comercial INT NOT NULL, -- id del comercial que gestionó el pedido, referencia a comercial
    FOREIGN KEY (id_cliente) REFERENCES cliente(id), -- clave foránea que relaciona con la tabla cliente
    FOREIGN KEY (id_comercial) REFERENCES comercial(id) -- clave foránea que relaciona con la tabla comercial
);
```

3. Insertar elementos a las tablas creadas anteriormente

```
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);
```

myan) | Editor de consultas (versión preliminar) ☆ ...

↑ Abrir consulta Comentarios Introducción

Consulta 1 × Consulta 2 ×

Ejecutar ☐ Cancelar consulta Guardar consulta Exportar datos como Mostrar solo editor Abrir Copilot

```
1 -- Insertar valores en la tabla cliente
2 INSERT INTO cliente (nombre, apellido1, apellido2, ciudad, categoria) VALUES
3 ('Aarón', 'Rivero', 'Gómez', 'Almería', 100),
4 ('Adela', 'Salas', 'Díaz', 'Granada', 200),
5 ('Adolfo', 'Rubio', 'Flores', 'Sevilla', NULL),
6 ('Adrián', 'Suárez', NULL, 'Jaén', 300),
7 ('Marcos', 'Loyola', 'Méndez', 'Almería', 200),
8 ('María', 'Santana', 'Moreno', 'Cádiz', 100),
9 ('Pilar', 'Ruiz', NULL, 'Sevilla', 300),
10 ('Pepe', 'Ruiz', 'Santana', 'Huelva', 200),
11 ('Guillermo', 'López', 'Gómez', 'Granada', 225),
12 ('Daniel', 'Santana', 'Loyola', 'Sevilla', 125);
```

Consulta 1 ✕ Consulta 2 ✕

Ejecutar

Cancelar consulta

Guardar consulta

Exportar datos como

Mostrar solo editor

Abrir Cop

```
14 -- Insertar valores en la tabla comercial
15 INSERT INTO comercial (nombre, apellido1, apellido2, comision) VALUES
16 ('Daniel', 'Sáez', 'Vega', 0.15),
17 ('Juan', 'Gómez', 'López', 0.13),
18 ('Diego', 'Flores', 'Salas', 0.11),
19 ('Marta', 'Herrera', 'Gil', 0.14),
20 ('Antonio', 'Carretero', 'Ortega', 0.12),
21 ('Manuel', 'Domínguez', 'Hernández', 0.13),
22 ('Antonio', 'Vega', 'Hernández', 0.11),
23 ('Alfredo', 'Ruiz', 'Flores', 0.05);
```

Consulta 1 ✕ Consulta 2 ✕

Ejecutar

Cancelar consulta

Guardar consulta

Exportar datos como

Mostrar solo editor

Abrir Copilot

```
25 -- Insertar valores en la tabla pedido
26 INSERT INTO pedido (total, fecha, id_cliente, id_comercial) VALUES
27 (150.5, '2017-10-05', 5, 2),
28 (270.65, '2016-09-10', 1, 5),
29 (65.26, '2017-10-05', 2, 1),
30 (110.5, '2016-08-17', 8, 3),
31 (948.5, '2017-09-10', 5, 2),
32 (2400.6, '2016-07-27', 7, 1),
33 (5760, '2015-09-10', 2, 1),
34 (1983.43, '2017-10-10', 4, 6),
35 (2480.4, '2016-10-10', 8, 3),
36 (250.45, '2015-06-27', 8, 2),
37 (75.29, '2016-08-17', 3, 7),
38 (2045.6, '2017-04-25', 2, 1)
```

Ir consulta Comentarios Introducción

Consulta 1 ✕ Consulta 2 ✕ Consulta 3 ✕

Ejecutar

Cancelar consulta

Guardar consulta

Exportar datos como

Mostrar solo editor

Abrir Copilot

```
1 -- Mostrar todo el contenido de la tabla cliente
2 SELECT *
3 FROM cliente; -- Selecciona todas las filas y columnas de la tabla cliente
4 -- Mostrar todo el contenido de la tabla comercial
5 SELECT *
6 FROM comercial; -- Selecciona todas las filas y columnas de la tabla comercial
7 -- Mostrar todo el contenido de la tabla pedido
8 SELECT *
9 FROM pedido; -- Selecciona todas las filas y columnas de la tabla pedido
10
```

Resultados Mensajes

Buscar en elementos de filtro...

id	nombre	apellido1	apellido2	ciudad	categoría
1	Aarón	Rivero	Gómez	Almería	100
2	Adela	Salas	Díaz	Granada	200
3	Adolfo	Rubio	Flores	Sevilla	
4	Adrián	Suárez		Jaén	300
5	Marcos	Loyola	Méndez	Almería	200
6	María	Santana	Moreno	Cádiz	100

rayan | Editor de consultas (version preliminar)

Abrir consulta

Comentarios

Introducción

Consulta 1 ×

Consulta 2 ×

Consulta 3 ×

Ejecutar

Cancelar consulta

Guardar consulta

Exportar datos como

Mostrar solo editor

Abrir Copilot

```
1 -- Mostrar todo el contenido de la tabla cliente
2 SELECT *
3 FROM cliente; -- Selecciona todas las filas y columnas de la tabla cliente
4 -- Mostrar todo el contenido de la tabla comercial
5 SELECT *
6 FROM comercial; -- Selecciona todas las filas y columnas de la tabla comercial
7 -- Mostrar todo el contenido de la tabla pedido
8 SELECT *
9 FROM pedido; -- Selecciona todas las filas y columnas de la tabla pedido
10
```

Resultados

Mensajes

Buscar en elementos de filtro...

id	total	fecha	id_cliente	id_comercial
1	150.5	2017-10-05	5	2
2	270.65	2016-09-10	1	5
3	65.26	2017-10-05	2	1
4	110.5	2016-08-17	8	3
5	948.5	2017-09-10	5	2
6	2400.6	2016-07-27	7	1
7	5760	2015-06-10	2	1

```
12 DELETE FROM pedido
13 WHERE id IN (1, 2); -- Especifica los IDs de la
14
```

Consulta realizada correctamente: Affected rows: 2

5. Modificar dos filas de la tabla pedido

```
-- Actualizar la tabla pedido para dos filas específicas
UPDATE pedido
SET total = 500.00, -- Nuevo valor para la columna 'total'
    fecha = '2020-05-15' -- Nuevo valor para la columna 'fecha'
WHERE id IN (3, 4); -- Especifica las filas a modificar mediante sus IDs
```

ltados Mensajes

ulta realizada correctamente: Affected rows: 2

6. Realizar las siguientes Consultas Básicas en una tabla

- a) 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.

```
22 -- Listado de todos los pedidos realizados
23 SELECT *
24 FROM pedido;
25
```

Resultados Mensajes

Buscar en elementos de filtro...				
id	total	fecha	id_cliente	id_comercial
3	500	2020-05-15	2	1
4	500	2020-05-15	8	3
5	948,5	2017-09-10	5	2
6	2400,6	2016-07-27	7	1
7	5760	2015-09-10	2	1
8	1983,43	2017-10-10	4	6
9	2480,4	2016-10-10	8	3
10	250,45	2015-06-27	8	2

b) Devuelve todos los datos de los dos pedidos de mayor valor.

ayan) | Editor de consultas (versión preliminar) ☆ ...

Abrir consulta Comentarios Introducción

Consulta 1 Consulta 2 **Consulta 3**

Ejecutar Cancelar consulta Guardar consulta Exportar datos como Mostrar solo editor Abrir Copilot

```
19 WHERE id IN (3, 4); -- Especifica las filas a modificar mediante sus IDs
20
21
22 -- Listado de todos los pedidos realizados
23 SELECT *
24 FROM pedido;
25
26
27 -- Listado de los dos pedidos de mayor valor
28 SELECT *
29 FROM pedido
30 ORDER BY total DESC -- Ordena por el valor total en orden descendente
31 OFFSET 0 ROWS FETCH NEXT 2 ROWS ONLY; -- Devuelve solo las dos primeras filas
32
```

Resultados Mensajes

Buscar en elementos de filtro...				
id	total	fecha	id_cliente	id_comercial
7	5760	2015-09-10	2	1
12	3045,6	2017-04-25	2	1

- c) 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

```
5 -- Listado de los identificadores de clientes que han realizado algún pedido
6 SELECT DISTINCT id_cliente
7 FROM pedido;
8
9
```

Resultados Mensajes

🔍 Buscar en elementos de filtro...

id_cliente

1
2
3
4
5
6

- d) 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

```
19 -- Listado de identificador, nombre y primer apellido de clientes cuyo segundo apellido no es NULL
20
21 SELECT id, nombre, apellido1
22 FROM cliente
23 WHERE apellido2 IS NOT NULL;
24
25
```

Resultados Mensajes

🔍 Buscar en elementos de filtro...

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

- e) 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.

```
45
46 -- Listado de los nombres de los clientes que empiezan por 'A' y terminan por
47 SELECT nombre
48 FROM cliente
49 WHERE (nombre LIKE 'A%n' OR nombre LIKE 'P%');
50
51
```

Resultados Mensajes

🔍 Buscar en elementos de filtro...

nombre

Aarón
Adrián
Pilar
Pepe

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


```

52 -- Listado de los nombres de los clientes que no empiezan por 'A'
53 SELECT nombre
54 FROM cliente
55 WHERE nombre NOT LIKE 'A%';
56
57

```

Resultados Mensajes

Buscar en elementos de filtro...
nombre
Marcos
María
Pilar
Pepe
Guillermo
Daniel

- g) 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.

```

57 -- Listado de nombres de los comerciales que terminan por 'o' o 'O'
58 SELECT nombre
59 FROM comercial
60 WHERE nombre LIKE '%o' OR nombre LIKE '%O';
61
62

```

Resultados Mensajes

Buscar en elementos de filtro...
nombre
Diego
Antonio
Antonio
Alfredo

7. Realizar las siguientes consultas avanzadas en varias tablas

- 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.
- 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.

a)

```

63 -- Listado de todos los pedidos realizados por cada cliente, mostrando todos los datos de pedidos y clientes, ordenados alfabéticamente
64 SELECT cliente.id, cliente.nombre, cliente.apellido1, cliente.apellido2, cliente.ciudad, cliente.categoria,
65 pedido.id AS id_pedido, pedido.total, pedido.fecha, pedido.id_comercial
66 FROM pedido
67 JOIN cliente ON pedido.id_cliente = cliente.id -- Relaciona los pedidos con los clientes
68 ORDER BY cliente.nombre, cliente.apellido1, cliente.apellido2; -- Ordena los resultados alfabéticamente por nombre y apellidos del cliente
69

```

Resultados Mensajes

Buscar en elementos de filtro...

id	nombre	apellido1	apellido2	ciudad	categoria	id_pedido
1	Aarón	Rivero	Gómez	Almería	100	15
1	Aarón	Rivero	Gómez	Almería	100	16
2	Adela	Salas	Díaz	Granada	200	3
2	Adela	Salas	Díaz	Granada	200	7
2	Adela	Salas	Díaz	Granada	200	12
3	Adolfo	Rubio	Flores	Sevilla		11
4	Adrián	Suárez		Jaén	300	8
5	Marcos	Loyola	Méndez	Almería	200	5

B)

```

71 -- Listado de todos los pedidos en los que ha participado un comercial, mostrando todos los datos de los pedidos y los comerciales, ordenados alfabéticamente
72 SELECT comercial.id AS id_comercial, comercial.nombre, comercial.apellido1, comercial.apellido2, comercial.comision,
73 pedido.id AS id_pedido, pedido.total, pedido.fecha, pedido.id_cliente
74 FROM pedido
75 JOIN comercial ON pedido.id_comercial = comercial.id -- Relaciona los pedidos con los comerciales
76 ORDER BY comercial.nombre, comercial.apellido1, comercial.apellido2; -- Ordena los resultados alfabéticamente por nombre y apellidos del comercial
77
78

```

Resultados Mensajes

Buscar en elementos de filtro...

id_comercial	nombre	apellido1	apellido2	comision	id_pedido	total
5	Antonio	Carretero	Ortega	0,12	15	370,85
5	Antonio	Carretero	Ortega	0,12	16	2389,23
7	Antonio	Vega	Hernández	0,11	11	75,29
1	Daniel	Sáez	Vega	0,15	3	500
1	Daniel	Sáez	Vega	0,15	6	2400,6
1	Daniel	Sáez	Vega	0,15	7	5760
1	Daniel	Sáez	Vega	0,15	12	3045,6
1	Daniel	Sáez	Vega	0,15	13	545,75

- c) 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.
- d) Devuelve un listado de todos los clientes que realizaron un pedido durante el año 2017, cuya cantidad esté entre 300 € y 1000 €.
- e) 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.
- f) 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.
- g) Calcula la cantidad total que suman todos los pedidos que aparecen en la tabla pedido.
- h) Calcula la cantidad media de todos los pedidos que aparecen en la tabla pedido.
- i) Calcula el número total de comerciales distintos que aparecen en la tabla pedido.
- j) Calcula el número total de clientes que aparecen en la tabla cliente.
- k) Calcula cuál es la mayor cantidad que aparece en la tabla pedido
- l) 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.

C)

```
-- Listado de todos los clientes, junto con todos los pedidos realizados y los datos de los comerciales asociados
SELECT cliente.id AS id_cliente, cliente.nombre, cliente.apellido1, cliente.apellido2, cliente.ciudad, cliente.categoria,
       pedido.id AS id_pedido, pedido.total, pedido.fecha,
       comercial.id AS id_comercial, comercial.nombre AS nombre_comercial, comercial.apellido1 AS apellido1_comercial, comercial.apellido2 AS apellido2_comercial,
FROM cliente
LEFT JOIN pedido ON cliente.id = pedido.id_cliente --- Relaciona los clientes con los pedidos
LEFT JOIN comercial ON pedido.id_comercial = comercial.id --- Relaciona los pedidos con los comerciales
ORDER BY cliente.nombre, cliente.apellido1, cliente.apellido2; -- Ordena los resultados alfabéticamente por cliente
```

Resultados Mensajes

* Buscar en elementos de filtro...

id_cliente	nombre	apellido1	apellido2	ciudad	categoria	id_pedido
	Aarón	Rivero	Gómez	Almería	100	15
	Aarón	Rivero	Gómez	Almería	100	16
	Adela	Salas	Díaz	Granada	200	3
	Adela	Salas	Díaz	Granada	200	7
	Adela	Salas	Díaz	Granada	200	12
	Adolfo	Rubio	Flores	Sevilla		11
	Adrián	Suárez		Jaén	300	8
0	Daniel	Santana	Loyola	Sevilla	125	

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

```

88
89 -- Listado de clientes que realizaron un pedido en 2017 con un total entre 300 € y 1000 €
90 SELECT DISTINCT cliente.id, cliente.nombre, cliente.apellido1, cliente.apellido2
91 FROM cliente
92 JOIN pedido ON cliente.id = pedido.id_cliente --- Relaciona clientes con pedidos
93 WHERE YEAR(pedido.fecha) = 2017 --- Filtra por el año 2017
94 AND pedido.total BETWEEN 300 AND 1000; --- Filtra por total entre 300 y 1000
95
96

```

Resultados Mensajes

Buscar en elementos de filtro...

id	nombre	apellido1	apellido2
5	Marcos	Loyola	Méndez

e) 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:

```

97 -- Listado de todos los clientes junto con los datos de los pedidos realizados, incluyendo clientes sin pedidos
98 SELECT cliente.id AS id_cliente, cliente.nombre, cliente.apellido1, cliente.apellido2,
99 --- pedido.id AS id_pedido, pedido.total, pedido.fecha
100 FROM cliente
101 LEFT JOIN pedido ON cliente.id = pedido.id_cliente --- Relaciona clientes con pedidos, incluyendo clientes sin pedidos
102 ORDER BY cliente.nombre; -- Ordena por nombre del cliente
103
104

```

Resultados Mensajes

Buscar en elementos de filtro...

id_cliente	nombre	apellido1	apellido2	id_pedido	total	fecha
1	Aarón	Rivero	Gómez	15	370,85	2019-03-11
1	Aarón	Rivero	Gómez	16	2389,23	2019-03-11
2	Adela	Salas	Díaz	3	500	2020-05-15
2	Adela	Salas	Díaz	7	5760	2015-09-10
2	Adela	Salas	Díaz	12	3045,6	2017-04-25
3	Adolfo	Rubio	Flores	11	75,29	2016-08-17
4	Adrián	Suárez		8	1983,43	2017-10-10

f) 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:

```

106 -- Listado de todos los comerciales junto con los datos de los pedidos realizados, incluyendo comerciales sin pedidos
107 SELECT comercial.id AS id_comercial, comercial.nombre, comercial.apellido1, comercial.apellido2,
108 --- pedido.id AS id_pedido, pedido.total, pedido.fecha
109 FROM comercial
110 LEFT JOIN pedido ON comercial.id = pedido.id_comercial --- Relaciona comerciales con pedidos, incluyendo comerciales sin pedidos
111 ORDER BY comercial.nombre; -- Ordena por nombre del comercial
112

```

Resultados Mensajes

Buscar en elementos de filtro...

id_comercial	nombre	apellido1	apellido2	id_pedido	total	fecha
8	Alfredo	Ruiz	Flores			
7	Antonio	Vega	Hernández	11	75,29	2016-08-17
5	Antonio	Carretero	Ortega	15	370,85	2019-03-11
5	Antonio	Carretero	Ortega	16	2389,23	2019-03-11
1	Daniel	Sáez	Vega	3	500	2020-05-15
1	Daniel	Sáez	Vega	6	2400,6	2016-07-27
1	Daniel	Sáez	Vega	7	5760	2015-09-10
1	Daniel	Sáez	Vega	12	3045,6	2017-04-25

g) Calcula la cantidad total que suman todos los pedidos que aparecen en la tabla pedido:

```
114 -- Calcula la suma total de todos los pedidos
115 SELECT SUM(total) AS total_pedidos
116 FROM pedido; -- Suma el total de la columna 'total' en la tabla 'pedido'
117
118
```

Resultados Mensajes

Buscar en elementos de filtro...

total_pedidos

21395,92

h) Calcula la cantidad media de todos los pedidos que aparecen en la tabla

```
117
118 -- Calcula la cantidad media de todos los pedidos
119 SELECT AVG(total) AS cantidad_media
120 FROM pedido; -- Calcula el promedio de la columna 'total' en la tabla 'pedido'
121
122
```

Resultados Mensajes

Buscar en elementos de filtro...

cantidad_media

1528,28

i) Calcula el número total de comerciales distintos que aparecen en la tabla

```
123 -- Calcula el número total de comerciales distintos en la tabla 'pedido'
124 SELECT COUNT(DISTINCT id_comercial) AS total_comerciales
125 FROM pedido; -- Cuenta la cantidad de comerciales únicos que aparecen en la columna 'id_comercial'
126
127
```

Resultados Mensajes

Buscar en elementos de filtro...

total_comerciales

6

j) Calcula el número total de clientes que aparecen en la tabla cliente:

```

127  -- Calcula el número total de clientes en la tabla 'cliente'
128  SELECT COUNT(*) AS total_clientes
129  FROM cliente; -- Cuenta el número total de filas en la tabla 'cliente'
130
131

```

Resultados Mensajes

Buscar en elementos de filtro...

total_clientes

10

k) Calcula cuál es la mayor cantidad que aparece en la tabla pedido:

```

132  -- Calcula la mayor cantidad de un pedido en la tabla 'pedido'
133  SELECT MAX(total) AS mayor_cantidad
134  FROM pedido; -- Encuentra el valor máximo en la columna 'total'
135
136

```

Resultados Mensajes

Buscar en elementos de filtro...

mayor_cantidad

5760

l) Calcula cuál es el máximo valor de los pedidos realizados durante el mismo día para cada uno de los clientes:

```

137  -- Calcula el valor máximo de los pedidos realizados el mismo día para cada cliente
138  SELECT id_cliente, fecha, MAX(total) AS maximo_valor
139  FROM pedido
140  GROUP BY id_cliente, fecha; -- Agrupa por cliente y fecha, y devuelve el pedido de mayor valor para cada grupo
141
142

```

Resultados Mensajes

Buscar en elementos de filtro...

id_cliente	fecha	maximo_valor
8	2015-06-27	250,45
2	2015-09-10	5760
7	2016-07-27	2400,6
3	2016-08-17	75,29
8	2016-10-10	2480,4
6	2017-02-02	145,82
2	2017-04-25	3045,6
5	2017-09-10	948,5

- m) 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.
- n) 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.

M)

```

142 -- Máximo valor de los pedidos realizados el 2016-08-17 para cada comercial
143 SELECT comercial.id, comercial.nombre, comercial.apellido1, comercial.apellido2, MAX(pedido.total) AS max_total
144 FROM comercial
145 JOIN pedido ON comercial.id = pedido.id_comercial
146 WHERE pedido.fecha = '2016-08-17'
147 GROUP BY comercial.id, comercial.nombre, comercial.apellido1, comercial.apellido2;
148

```

Resultados Mensajes

Buscar en elementos de filtro...

id	nombre	apellido1	apellido2	max_total
7	Antonio	Vega	Hernández	75,29

N)

```

149
150 -- Listado de clientes con el número total de pedidos, incluyendo los que no tienen pedidos
151 SELECT cliente.id, cliente.nombre, cliente.apellido1, cliente.apellido2, COUNT(pedido.id) AS total_pedidos
152 FROM cliente
153 LEFT JOIN pedido ON cliente.id = pedido.id_cliente
154 GROUP BY cliente.id, cliente.nombre, cliente.apellido1, cliente.apellido2;
155

```

Resultados Mensajes

Buscar en elementos de filtro...

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

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

O)

```
155
156 -- Listado de clientes y número total de pedidos realizados en 2017
157 SELECT cliente.id, cliente.nombre, cliente.apellido1, cliente.apellido2, COUNT(pedido.id) AS total_pedidos
158 FROM cliente
159 JOIN pedido ON cliente.id = pedido.id_cliente
160 WHERE YEAR(pedido.fecha) = 2017
161 GROUP BY cliente.id, cliente.nombre, cliente.apellido1, cliente.apellido2;
162
163
```

Resultados Mensajes

Buscar en elementos de filtro...				
id	nombre	apellido1	apellido2	total_pedidos
2	Adela	Salas	Díaz	1
4	Adrián	Suárez		1
5	Marcos	Loyola	Méndez	1
6	María	Santana	Moreno	1

P)

```
163
164 -- listado de clientes que realizaron un pedido en 2017 con un valor mayor o igual al promedio de los pedidos en 2017
165 SELECT cliente.id, cliente.nombre, cliente.apellido1, cliente.apellido2, pedido.total, pedido.fecha
166 FROM cliente
167 JOIN pedido ON cliente.id = pedido.id_cliente
168 WHERE YEAR(pedido.fecha) = 2017
169 AND pedido.total >= (SELECT AVG(total) FROM pedido WHERE YEAR(fecha) = 2017);
170
171
```

Resultados Mensajes

Buscar en elementos de filtro...					
id	nombre	apellido1	apellido2	total	fecha
4	Adrián	Suárez		1983,43	2017-10-10
2	Adela	Salas	Díaz	3045,6	2017-04-25

Q)

```
3
4 -- Listado de clientes que no han realizado ningún pedido utilizando NOT IN
5 SELECT cliente.id, cliente.nombre, cliente.apellido1, cliente.apellido2
6 FROM cliente
7 WHERE cliente.id NOT IN (SELECT id_cliente FROM pedido);
8
9
```

Resultados Mensajes

Buscar en elementos de filtro...			
id	nombre	apellido1	apellido2
9	Guillermo	López	Gómez
10	Daniel	Santana	Loyola

R)

```
178
179 -- Listado de comerciales que no han realizado ningún pedido utilizando NOT IN
180 SELECT comercial.id, comercial.nombre, comercial.apellido1, comercial.apellido2
181 FROM comercial
182 WHERE comercial.id NOT IN (SELECT id_comercial FROM pedido);
183
184
```

Resultados Mensajes

Buscar en elementos de filtro...			
id	nombre	apellido1	apellido2
4	Marta	Herrera	Gil
8	Alfredo	Ruiz	Flores