

# Examen Parcial

## Asignatura: Base de Datos

Apellidos y Nombres: Mosquera Zevallos Valerio

Ciclo: 5to

Sección: 1A

Duración:

Docente: Mg. Ing. Raúl Fernández Bejarano

Código: R03398D

Fecha: 24.10.24

## Base de Datos en Microsoft Azure

### 1. Crear una Base de datos en un grupo de recursos

Microsoft Azure | Actualización | Buscar recursos, servicios y documentos (0+)

Inicio > Microsoft.SqlDatabase.newDatabaseExistingServer\_b682703fa8b4032 | Información general >

**Ventas (servidor-mosquera/Ventas)** Base de datos SQL

Buscar Copiar Restaurar Exportar Establecer firewall de servidor Eliminar Conectar con... Comentarios

**Información general**

- Registro de actividad
- Etiquetas
- Diagnosticar y solucionar problemas
- Editor de consultas (versión preliminar)
- Base de datos reflejada en Fabric (versión preliminar)
- Configuración
- Administración de datos
- Integraciones
- Power Platform
- Seguridad
- Rendimiento inteligente
- Supervisión
- Automation
- Ayuda

**Essentials**

- Grupo de recursos (mover): Recursos-Base-de-Datos-II
- Estado: Online
- Ubicación: Central US
- Suscripción (mover): Azure subscription 1
- Id. de suscripción: aac15e33-7d6c-4de4-a039-eb67fc4e1de4
- Etiquetas (editar): Agregar etiquetas

Nombre del servidor: servidor-mosquera.database.windows.net  
Cadenas de conexión: Mostrar las cadenas de conexión de la base de datos  
Plan de tarifa: Uso general - Sin servidor: Gen5, 1 vCore  
Retraso de pausa automática: 1 hora  
Punto de restauración más antiguo: No hay ningún punto de restauración disponible

**Empezar a trabajar con el panel Kanban**

Conéctese a la base de datos y empiece a trabajar con datos con unos pocos pasos sencillos. Más información >

- Configurar acceso**  
Configure el acceso de red al servidor SQL Server. Más información >  
Configurar
- Conectar con una aplicación**  
Use cadenas de conexión para conectarse a la base de datos SQL desde sus aplicaciones y herramientas favoritas.  
Ver cadenas de conexión
- Comenzar a desarrollar**  
Trabaje en la base de datos con herramientas para agregar, modificar y consultar datos. Comparar herramientas >  
Abrir Azure Data Studio  
Abrir en Visual Studio  
Apertura en Visual Studio Code
- Base de datos reflejada en Fabric**  
Replice las bases de datos existentes en Fabric y ayude a su equipo a conseguir unos objetivos simplificados de ETL y análisis operativo. Más información >

### 2. Se agrega nuestra dirección IP

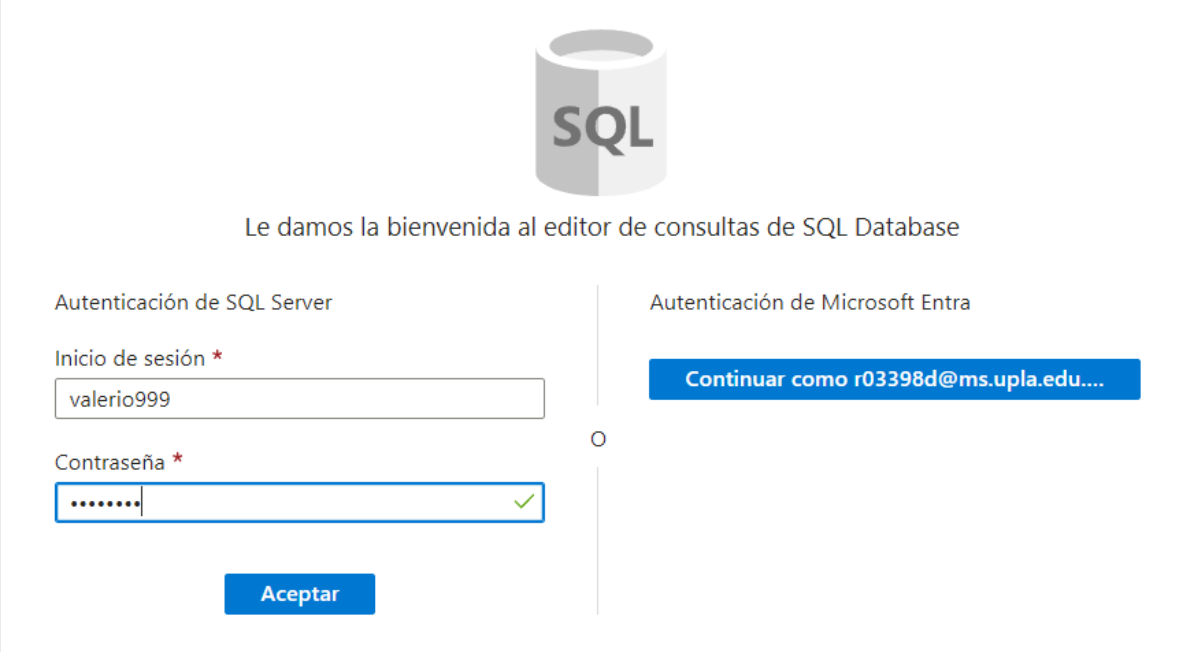
**Reglas de firewall**

Permita que determinadas direcciones IP de Internet públicas accedan a su recurso. Más información >

+ Agregar la dirección IPv4 del cliente (45.5.57.35) + Agregar una regla de firewall

Nombre de regla	Dirección IPv4 de inicio	Dirección IPv4 final	
MIDireccionIP	45.232.150.170	45.232.150.170	
query-editor-4d1ea5	201.230.41.206	201.230.41.206	
query-editor-804237	45.5.57.35	45.5.57.35	

3. Para acceder al editor de consultas debemos ingresar nuestras credenciales.



SQL

Le damos la bienvenida al editor de consultas de SQL Database

Autenticación de SQL Server

Inicio de sesión \*

valerio999

Contraseña \*

.....

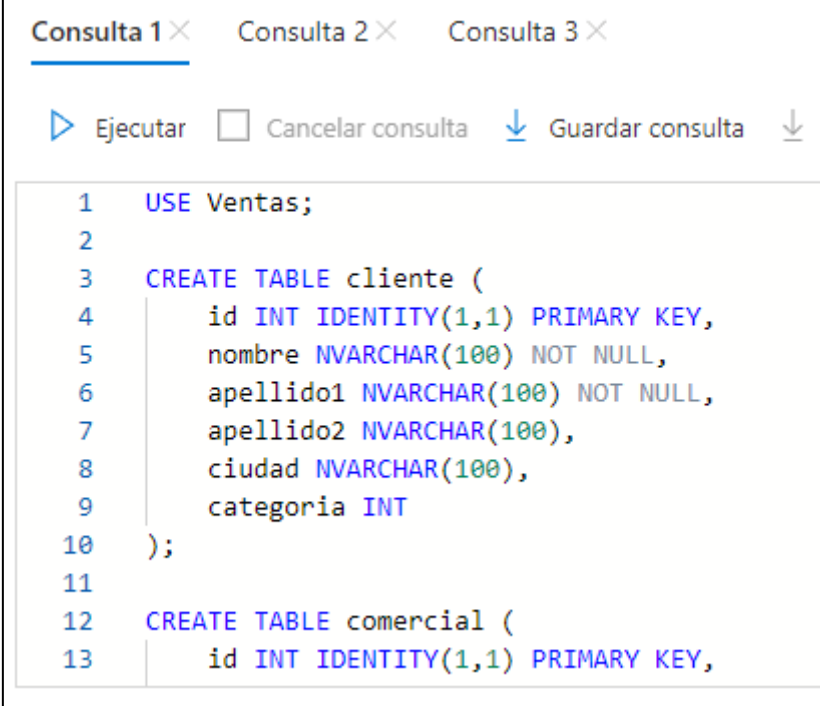
Aceptar

Autenticación de Microsoft Entra

Continuar como r03398d@ms.upla.edu....

O

4. Creamos 3 consultas para ordenar nuestro script (Ingresar tabla, Ingresar registros y hacer consultas). Ingresar tablas:





Consulta 1 ✕ Consulta 2 ✕ Consulta 3 ✕

Ejecutar Cancelar consulta Guardar consulta

```
1  USE Ventas;
2
3  CREATE TABLE cliente (
4      id INT IDENTITY(1,1) PRIMARY KEY,
5      nombre NVARCHAR(100) NOT NULL,
6      apellido1 NVARCHAR(100) NOT NULL,
7      apellido2 NVARCHAR(100),
8      ciudad NVARCHAR(100),
9      categoria INT
10 );
11
12 CREATE TABLE comercial (
13     id INT IDENTITY(1,1) PRIMARY KEY,
```


## 5. Ingresar registros.

Consulta 1 × Consulta 2 × Consulta 3 ×

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

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

Resultados Mensajes

 Buscar en elementos de filtro...

## 6. Hacer las consultas: Las consultas se muestran con más detalles en la sección “Enunciado 04”

Consulta 1 × Consulta 2 × Consulta 3 ×

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

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

Resultados Mensajes

 Buscar en elementos de filtro...

---

## Enunciado 04

### Eliminar dos filas de la tabla **pedido**

Para eliminar filas, primero necesitas identificar los **id** de los pedidos que deseas eliminar. Supongamos que deseas eliminar los pedidos con **id** 1 y 2:

```
DELETE FROM pedido WHERE id IN (1, 2);
```

1	-- 4. Eliminar dos filas de la tabla pedido
2	DELETE FROM pedido WHERE id IN (1, 2);
3	
4	
5	
6	
7	

Resultados

Mensajes

Consulta realizada correctamente: Affected rows: 2

## Enunciado 05

### Modificar dos filas de la tabla **pedido**

Para modificar filas, puedes usar la instrucción **UPDATE**. Supongamos que deseas cambiar el **total** de los pedidos con **id** 3 y 4:

```
UPDATE pedido SET total = 100.00 WHERE id = 3;  
UPDATE pedido SET total = 200.00 WHERE id = 4;
```

```

3
4  -- 5. Modificar dos filas de la tabla pedido
5  UPDATE pedido SET total = 100.00 WHERE id = 3;
6  UPDATE pedido SET total = 200.00 WHERE id = 4;
7
8

```

Resultados Mensajes

Consulta realizada correctamente: Affected rows: 2

## Enunciado 06

**a) Listado de todos los pedidos ordenados por fecha (más recientes primero)**

```
SELECT * FROM pedido ORDER BY fecha DESC;
```

```

8  -- 6. Consultas Básicas
9  -- a) Listado de todos los pedidos ordenados por fecha (más recientes primero)
10 SELECT * FROM pedido ORDER BY fecha DESC;
11
12 --b) Los dos pedidos de mayor valor
13 SELECT TOP 2 * FROM pedido ORDER BY total DESC;
14 --c) Identificadores de clientes que han realizado algún pedido (sin duplicados)
15 SELECT DISTINCT id_cliente FROM pedido;
16

```

Resultados Mensajes

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
3	100	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

Consulta realizada correctamente | 0s

**b) Los dos pedidos de mayor valor**

```
SELECT TOP 2 * FROM pedido ORDER BY total DESC;
```

```
--b) Los dos pedidos de mayor valor
SELECT TOP 2 * FROM pedido ORDER BY total DESC;

--c) Identificadores de clientes que han realizado algún pedido (sin duplicados)
SELECT DISTINCT id_cliente FROM pedido;

--d) Identificador, nombre y primer apellido de clientes con segundo apellido no NULL (ord
SELECT id, nombre, apellidos
```

**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) Identificadores de clientes que han realizado algún pedido (sin duplicados)**

```
SELECT DISTINCT id_cliente FROM pedido;
```

```

14
15 --c) Identificadores de clientes que han realizado algún pedido (sin duplicados)
16 SELECT DISTINCT id_cliente FROM pedido;
17
18 --d) Identificador, nombre y primer apellido de clientes con segundo apellido no NULL
19 SELECT id, nombre, apellido1
20 FROM cliente
21 WHERE apellido2 IS NOT NULL
22 ORDER BY apellido1, nombre;

```

Resultados   Mensajes

🔍 Buscar en elementos de filtro...

id_cliente
1
2
3
4
5
6
7
8

consulta realizada correctamente | 0s

**d) Identificador, nombre y primer apellido de clientes con segundo apellido no NULL (ordenados alfabéticamente)**

```

SELECT id, nombre, apellido1
FROM cliente
WHERE apellido2 IS NOT NULL
ORDER BY apellido1, nombre;

```

```

18 --d) Identificador, nombre y primer apellido de clientes con segundo apellido no NULL (ordenados alfabéticamente)
19 SELECT id, nombre, apellido1
20 FROM cliente
21 WHERE apellido2 IS NOT NULL
22 ORDER BY apellido1, nombre;
23
24 --e) Nombres de clientes que empiezan por A y terminan por n, o empiezan por P (ordenados alfabéticamente)
25 SELECT nombre
26 FROM cliente
27 WHERE (nombre LIKE 'A%n' OR nombre LIKE 'P%')
28 ORDER BY nombre;
29

```

**Resultados**   Mensajes

🔍 Buscar en elementos de filtro...

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

Consulta realizada correctamente | 0s

**e) Nombres de clientes que empiezan por A y terminan por n, o empiezan por P (ordenados alfabéticamente)**

```

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

```



```
23
24 --e) Nombres de clientes que empiezan por A y terminan por n, o empiezan por P (ordenados
25 SELECT nombre
26 FROM cliente
27 WHERE (nombre LIKE 'A%n' OR nombre LIKE 'P%')
28 ORDER BY nombre;
29
30 --f) Nombres de clientes que no empiezan por A (ordenados alfabéticamente)
31 SELECT nombre
32 FROM cliente
33 WHERE nombre NOT LIKE 'A%'
34 ORDER BY nombre;
```

**Resultados** Mensajes

🔍 Buscar en elementos de filtro...

nombre
Aarón
Aarón
Adrián
Pepe
Pilar

**f) Nombres de clientes que no empiezan por A (ordenados alfabéticamente)**

```
SELECT nombre
FROM cliente
WHERE nombre NOT LIKE 'A%'
ORDER BY nombre;
```

```

29
30 --f) Nombres de clientes que no empiezan por A (ordenados alfabéticamente)
31 SELECT nombre
32 FROM cliente
33 WHERE nombre NOT LIKE 'A%'
34 ORDER BY nombre;
35
36 --g) Nombres de comerciales que terminan por "o" o "o" (sin duplicados)
37 SELECT DISTINCT nombre
38 FROM comercial
39 WHERE nombre LIKE '%o'
40 ORDER BY nombre;

```

**Resultados** Mensajes

🔍 Buscar en elementos de filtro...

nombre

Daniel

Guillermo

Marcos

María

Pepe

Pilar

#### g) Nombres de comerciales que terminan por "o" o "o" (sin duplicados)

```

SELECT DISTINCT nombre
FROM comercial
WHERE nombre LIKE '%o'
ORDER BY nombre;

```

```

35
36 --g) Nombres de comerciales que terminan por "o" o "o" (sin duplicados)
37 SELECT DISTINCT nombre
38 FROM comercial
39 WHERE nombre LIKE '%o'
40 ORDER BY nombre;

```

**Resultados** Mensajes

🔍 Buscar en elementos de filtro...

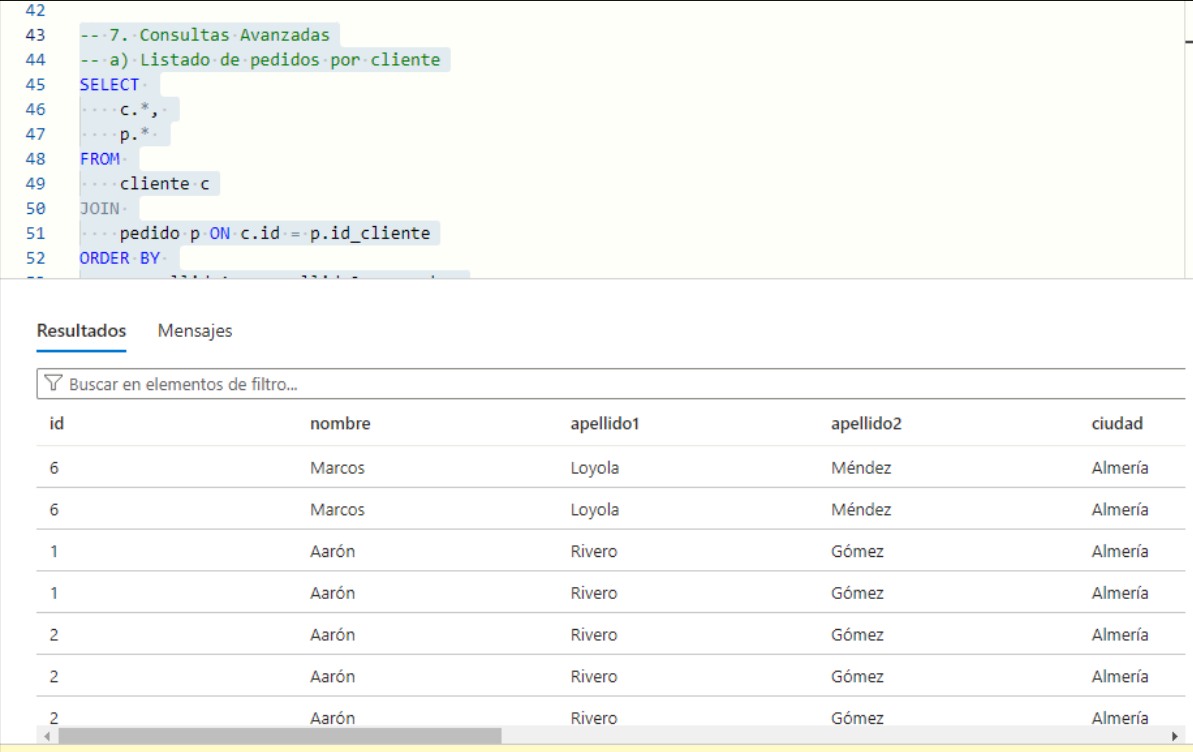
nombre

Carretero

## Enunciado 07

### a) Listado de pedidos por cliente

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



The screenshot shows a SQL query execution interface. The query is as follows:

```
-- 7.- Consultas Avanzadas
-- a) Listado de pedidos por cliente
SELECT
    c.*,
    p.*
FROM
    cliente c
JOIN
    pedido p ON c.id = p.id_cliente
ORDER BY
    c.apellido1, c.apellido2, c.nombre;
```

The results are displayed in a table with the following columns: id, nombre, apellido1, apellido2, and ciudad. The table contains 8 rows of data.

id	nombre	apellido1	apellido2	ciudad
6	Marcos	Loyola	Méndez	Almería
6	Marcos	Loyola	Méndez	Almería
1	Aarón	Rivero	Gómez	Almería
1	Aarón	Rivero	Gómez	Almería
2	Aarón	Rivero	Gómez	Almería
2	Aarón	Rivero	Gómez	Almería
2	Aarón	Rivero	Gómez	Almería
2	Aarón	Rivero	Gómez	Almería

Consulta realizada correctamente | 0s

### b) Listado de pedidos por comercial

```
sql
Copiar código
SELECT
    com.*,
    p.*
FROM
    comercial com
JOIN
```

```

        pedido p ON com.id = p.id_comercial
ORDER BY
        com.apellido1, com.apellido2, com.nombre;

```

55 -- b) Listado de pedidos por comercial  
56 SELECT  
57 .... com.\*,  
58 .... p.\*,  
59 FROM  
60 .... comercial com  
61 JOIN  
62 .... pedido p ON com.id = p.id\_comercial  
63 ORDER BY  
64 .... com.apellido1, com.apellido2, com.nombre;  
65  
66 -- c) Listado de clientes con pedidos y comerciales

Resultados Mensajes

🔍 Buscar en elementos de filtro...

id	apellido1	nombre	apellido2	comision
7	Antonio	Vega	Hernández	0,11
5	Antonio	Carretero	Ortega	0,12
5	Antonio	Carretero	Ortega	0,12
1	Daniel	Sáez	Vega	0,15
1	Daniel	Sáez	Vega	0,15
1	Daniel	Sáez	Vega	0,15
1	Daniel	Sáez	Vega	0,15

Consulta realizada correctamente 10s

### c) Listado de clientes con pedidos y comerciales

sql

Copiar código

```

SELECT
    c.*,
    p.*,
    com.*
FROM
    cliente c
LEFT JOIN
    pedido p ON c.id = p.id_cliente
LEFT JOIN
    comercial com ON p.id_comercial = com.id
ORDER BY
    c.apellido1, c.apellido2, c.nombre;

```

66	-- c) Listado de clientes con pedidos y comerciales
67	SELECT
68	... c.*,
69	... p.*,
70	... com.*
71	FROM
72	... cliente c
73	LEFT JOIN
74	... pedido p ON c.id = p.id_cliente
75	LEFT JOIN
76	... comercial com ON p.id_comercial = com.id

Resultados		Mensajes		
<input type="text"/> Buscar en elementos de filtro...				
id	nombre	apellido1	apellido2	ciudad
10	Guillermo	López	Gómez	Granada
6	Marcos	Loyola	Méndez	Almería
6	Marcos	Loyola	Méndez	Almería
1	Aarón	Rivero	Gómez	Almería
1	Aarón	Rivero	Gómez	Almería
2	Aarón	Rivero	Gómez	Almería
2	Aarón	Rivero	Gómez	Almería

#### d) Clientes con pedidos en 2017 entre 300 € y 1000 €

sql

Copiar código

SELECT

... c.\*

FROM

... cliente c

JOIN

... pedido p ON c.id = p.id\_cliente

WHERE

YEAR(p.fecha) = 2017 AND p.total BETWEEN 300 AND 1000;

```

79
80 --d) Clientes con pedidos en 2017 entre 300 € y 1000 €
81 SELECT
82   ... c.*
83 FROM
84   ... cliente c
85 JOIN
86   ... pedido p ON c.id = p.id_cliente
87 WHERE
88   ... YEAR(p.fecha) = 2017 AND p.total BETWEEN 300 AND 1000;
89

```

Resultados   Mensajes

🔍 Buscar en elementos de filtro...

id	nombre	apellido1	apellido2	ciudad	categoria
5	Adrián	Suárez		Jaén	300

### e) Clientes y pedidos (incluye clientes sin pedidos)

sql

Copiar código

```

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

```

```

90 -- e) Clientes y pedidos (incluye clientes sin pedidos)
91 SELECT
92     c.*,
93     p.*
94 FROM
95     cliente c
96 LEFT JOIN
97     pedido p ON c.id = p.id_cliente
98 ORDER BY
99     c.apellido1, c.apellido2, c.nombre;
100

```

Resultados    Mensajes

Buscar en elementos de filtro...

id	nombre	apellido1	apellido2	ciudad
10	Guillermo	López	Gómez	Granada
6	Marcos	Loyola	Méndez	Almería
6	Marcos	Loyola	Méndez	Almería
1	Aarón	Rivero	Gómez	Almería
1	Aarón	Rivero	Gómez	Almería
2	Aarón	Rivero	Gómez	Almería
2	Aarón	Rivero	Gómez	Almería

Consulta realizada correctamente 10s

## f) Comerciales y pedidos (incluye comerciales sin pedidos)

sql

Copiar código

```

SELECT
    com.*,
    p.*
FROM
    comercial com
LEFT JOIN
    pedido p ON com.id = p.id_comercial
ORDER BY
    com.apellido1, com.apellido2, com.nombre;

```

id	apellido1	nombre	apellido
8	Alfredo	Ruiz	Flores
7	Antonio	Vega	Hernández
5	Antonio	Carretero	Ortega
5	Antonio	Carretero	Ortega
1	Daniel	Sáez	Vega
1	Daniel	Sáez	Vega

Consulta realizada correctamente | 1s

### g) Total de todos los pedidos

sql

Copiar código

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

Resultados	Mensajes
<input type="text"/> Buscar en elementos de filtro...	
Total_Pedidos	
20695,92	

### h) Cantidad media de todos los pedidos

sql

Copiar código

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

Resultados	Mensajes
<input type="text"/> Buscar en elementos de filtro...	
Media_Pedidos	
1478,28	

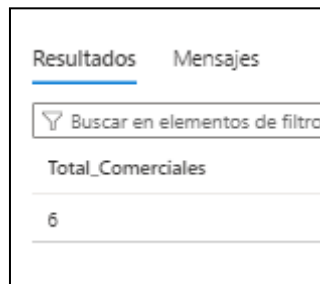


### i) Número total de comerciales distintos

sql

Copiar código

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



The screenshot shows a web interface with two tabs: 'Resultados' (active) and 'Mensajes'. Below the tabs is a search bar with a funnel icon and the text 'Buscar en elementos de filtro...'. The main content area displays a table with one column header 'Total\_Comerciales' and one row with the value '6'.

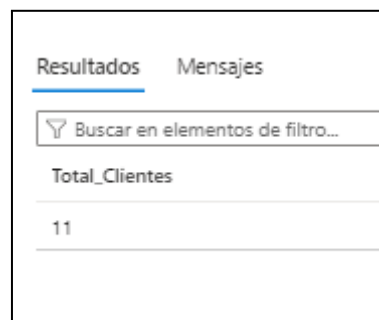
Total_Comerciales
6

### j) Número total de clientes

sql

Copiar código

```
SELECT
    COUNT(*) AS Total_Clientes
FROM
    cliente;
```



The screenshot shows a web interface with two tabs: 'Resultados' (active) and 'Mensajes'. Below the tabs is a search bar with a funnel icon and the text 'Buscar en elementos de filtro...'. The main content area displays a table with one column header 'Total\_Clientes' and one row with the value '11'.

Total_Clientes
11

### k) Mayor cantidad en la tabla de pedidos

sql

Copiar código

```
SELECT
    MAX(total) AS Mayor_Pedido
FROM
    pedido;
```

Resultados	Mensajes
<input type="text" value="Buscar en elementos de filtro..."/>	
Mayor_Pedido	
5760	

## 1) Máximo valor de pedidos por cliente en el mismo día

sql

Copiar código

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

Resultados

Mensajes

Buscar en elementos de filtro...

cliente_id	nombre	apellido1	apellido2	fecha
8	Pilar	Ruiz		2015-06-27
2	Aarón	Rivero	Gómez	2015-09-10
7	María	Santana	Moreno	2016-07-27
3	Adela	Salas	Díaz	2016-08-17
8	Pilar	Ruiz		2016-08-17
8	Pilar	Ruiz		2016-10-10

Resultados ordenados por: nombre | 2

**m) Máximo valor de los pedidos realizados para cada comercial el 2016-08-17**

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

Resultados Mensajes			
🔍 Buscar en elementos de filtro...			
id_comercial	nombre	apellido1	max_total
3	Flores	Diego	200
7	Vega	Antonio	75,29

**n) Listado de clientes con el número total de pedidos realizados**

```
SELECT cl.id AS id_cliente, cl.nombre, cl.apellido1, COUNT(p.id) AS  
total_pedidos  
FROM cliente cl  
LEFT JOIN pedido p ON cl.id = p.id_cliente  
GROUP BY cl.id, cl.nombre, cl.apellido1;
```

Resultados Mensajes			
🔍 Buscar en elementos de filtro...			
id_cliente	nombre	apellido1	total_pedidos
1	Aarón	Rivero	2
2	Aarón	Rivero	3
3	Adela	Salas	1
4	Adolfo	Rubio	1
5	Adrián	Suárez	1
6	Marcos	Loyola	2

**o) Listado de clientes con el número total de pedidos durante 2017**

```
SELECT cl.id AS id_cliente, cl.nombre, cl.apellido1, COUNT(p.id) AS  
total_pedidos  
FROM cliente cl  
LEFT JOIN pedido p ON cl.id = p.id_cliente AND YEAR(p.fecha) = 2017  
GROUP BY cl.id, cl.nombre, cl.apellido1;
```

Resultados

Mensajes

Buscar en elementos de filtro...

id_cliente	nombre	apellido1	total_pedidos
1	Aarón	Rivero	0
2	Aarón	Rivero	2
3	Adela	Salas	0
4	Adolfo	Rubio	1
5	Adrián	Suárez	1
6	Marcos	Loyola	1

**p) Clientes con pedidos en 2017 con valor mayor o igual al promedio**

```

WITH promedio AS (
    SELECT AVG(total) AS media_pedidos
    FROM pedido
    WHERE YEAR(fecha) = 2017
)
SELECT cl.*, p.*
FROM cliente cl
JOIN pedido p ON cl.id = p.id_cliente
WHERE YEAR(p.fecha) = 2017 AND p.total >= (SELECT media_pedidos FROM
promedio);

```

Resultados	Mensajes
<input type="text" value="Buscar en elementos de filtro..."/>	
media_pedidos	
1244,67	

**q) Listado de clientes que no han realizado ningún pedido**

```

SELECT cl.id AS id_cliente, cl.nombre, cl.apellido1
FROM cliente cl
WHERE NOT EXISTS (
    SELECT 1 FROM pedido p WHERE p.id_cliente = cl.id
);

```

Resultados Mensajes		
🔍 Buscar en elementos de filtro...		
id_cliente	nombre	apellido1
9	Pepe	Ruiz
10	Guillermo	López
11	Daniel	Santana

**r) Listado de comerciales que no han realizado ningún pedido**

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

Resultados Mensajes		
🔍 Buscar en elementos de filtro...		
id_comercial	nombre	apellido1
4	Herrera	Marta
8	Ruiz	Alfredo

## En SQL Sever Management Studio

```
USE Ventas;

CREATE TABLE cliente (
    id INT IDENTITY(1,1) PRIMARY KEY,
    nombre NVARCHAR(100) NOT NULL,
    apellido1 NVARCHAR(100) NOT NULL,
    apellido2 NVARCHAR(100),
    ciudad NVARCHAR(100),
    categoria INT
);

CREATE TABLE comercial (
    id INT IDENTITY(1,1) PRIMARY KEY,
    apellido1 NVARCHAR(100) NOT NULL,
    nombre NVARCHAR(100) NOT NULL,
    apellido2 NVARCHAR(100),
    comision FLOAT
);

CREATE TABLE pedido (
    id INT IDENTITY(1,1) PRIMARY KEY,
    total FLOAT NOT NULL,
    fecha DATE,
    id_cliente INT NOT NULL,
    id_comercial INT NOT NULL,
    FOREIGN KEY (id_cliente) REFERENCES cliente(id),
    FOREIGN KEY (id_comercial) REFERENCES comercial(id)
);
```

### 7. Insertar registros para las tablas

```
-- Inserciones en la tabla cliente
INSERT INTO cliente VALUES ('Aarón', 'Rivero', 'Gómez', 'Almería', 100);
INSERT INTO cliente VALUES ('Adela', 'Salas', 'Díaz', 'Granada', 200);
INSERT INTO cliente VALUES ('Adolfo', 'Rubio', 'Flores', 'Sevilla', NULL);
INSERT INTO cliente VALUES ('Adrián', 'Suárez', NULL, 'Jaén', 300);
INSERT INTO cliente VALUES ('Marcos', 'Loyola', 'Méndez', 'Almería', 200);
INSERT INTO cliente VALUES ('María', 'Santana', 'Moreno', 'Cádiz', 100);
INSERT INTO cliente VALUES ('Pilar', 'Ruiz', NULL, 'Sevilla', 300);
INSERT INTO cliente VALUES ('Pepe', 'Ruiz', 'Santana', 'Huelva', 200);
INSERT INTO cliente VALUES ('Guillermo', 'López', 'Gómez', 'Granada', 225);
INSERT INTO cliente VALUES ('Daniel', 'Santana', 'Loyola', 'Sevilla', 125);
```

```
-- Inserciones en la tabla comercial
INSERT INTO comercial VALUES ('Daniel', 'Sáez', 'Vega', 0.15);
INSERT INTO comercial VALUES ('Juan', 'Gómez', 'López', 0.13);
INSERT INTO comercial VALUES ('Diego', 'Flores', 'Salas', 0.11);
INSERT INTO comercial VALUES ('Marta', 'Herrera', 'Gil', 0.14);
INSERT INTO comercial VALUES ('Antonio', 'Carretero', 'Ortega', 0.12);
INSERT INTO comercial VALUES ('Manuel', 'Domínguez', 'Hernández', 0.13);
INSERT INTO comercial VALUES ('Antonio', 'Vega', 'Hernández', 0.11);
INSERT INTO comercial VALUES ('Alfredo', 'Ruiz', 'Flores', 0.05);
```

```
-- Inserciones en la tabla pedido
INSERT INTO pedido VALUES (150.5, '2017-10-05', 5, 2);
INSERT INTO pedido VALUES (270.65, '2016-09-10', 1, 5);
INSERT INTO pedido VALUES (65.26, '2017-10-05', 2, 1);
INSERT INTO pedido VALUES (110.5, '2016-08-17', 8, 3);
INSERT INTO pedido VALUES (948.5, '2017-09-10', 5, 2);
INSERT INTO pedido VALUES (2400.6, '2016-07-27', 7, 1);
INSERT INTO pedido VALUES (5760, '2015-09-10', 2, 1);
INSERT INTO pedido VALUES (1983.43, '2017-10-10', 4, 6);
INSERT INTO pedido VALUES (2480.4, '2016-10-10', 8, 3);
INSERT INTO pedido VALUES (250.45, '2015-06-27', 8, 2);
INSERT INTO pedido VALUES (75.29, '2016-08-17', 3, 7);
INSERT INTO pedido VALUES (3045.6, '2017-04-25', 2, 1);
INSERT INTO pedido VALUES (545.75, '2019-01-25', 6, 1);
INSERT INTO pedido VALUES (145.82, '2017-02-02', 6, 1);
INSERT INTO pedido VALUES (370.85, '2019-03-11', 1, 5);
INSERT INTO pedido VALUES (2389.23, '2019-03-11', 1, 5);
```

8.

9.