



CDMX, 14 de Abril de 2025.

Actividad 7. Ejercicios

Base de datos relacionales.

Miranda Velazquez Mariles

Alfonso Roberto Gómez Rosales

Jose Emiliano Jauregui Guzmán



ACTIVIDAD VII:

EJERCICIO

1. Con base en el material consultado en la unidad resuelve el ejercicio que se plantea a continuación acerca de los siguientes temas:

- Lenguaje de Manipulación de Datos DML
- Consultas en MySQL Workbench

Ejercicio. Realización de consultas a una base de datos.

2. Utilizando tu ambiente de trabajo MySQL Workbench, el cual instalaste en la primera parte del proyecto integrador ejecuta el *script.sql* que se encuentra en el segmento **1.3.2** de la siguiente referencia:

Sánchez, J. (2020). *Ejercicios. Relización de consultas SQL* [Sitio Web]
Recuperado de <https://josejuansanchez.org/bd/ejercicios-consultas-sql/index.html>

El *script* corresponde con una base de datos para Gestión de Ventas, se creará la base de datos y se le ingresarán registros. Si encuentras errores al ejecutar corrígelos y documéntalos. Enseguida realiza las siguientes consultas:

- a) Listado con todos los pedidos que se han realizado. Los pedidos deben estar ordenados por el campo fecha.

```
SELECT *
FROM pedido
ORDER BY fecha;
DROP DATABASE IF EXISTS ventas;
CREATE DATABASE ventas CHARACTER SET utf8mb4;
USE 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),
  categoria 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)
);
```

```
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);
```



MySQL Workbench

Local instance MySQL92 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Administration
- Schemas

Information

No object selected

sys_pedido

Find

1 • SELECT *

2 FROM pedido

3 ORDER BY fecha;

4 DROP DATABASE IF EXISTS ventas;

Result Grid

id	total	fecha	id_cliente	id_comercial
10	250.45	2015-06-27	8	2
7	5760	2015-09-10	2	1
6	2400.6	2016-07-27	7	1
4	110.5	2016-08-17	8	3
11	75.29	2016-08-17	3	7
2	270.65	2016-09-10	1	5

Exported resultset to C:\Users\Miran\OneDrive\Documents\UVM\Sistemas operativos\tabla 1.csv

Output

#	Time	Action	Message	Duration / Fetch
89	10:45:03	INSERT INTO pedido VALUES(13, 545.75, '2019-01-25', 6, 1)	1 row(s) affected	0.000 sec
90	10:45:03	INSERT INTO pedido VALUES(14, 145.82, '2017-02-02', 6, 1)	1 row(s) affected	0.000 sec
91	10:45:03	INSERT INTO pedido VALUES(15, 370.85, '2019-03-11', 1, 5)	1 row(s) affected	0.016 sec
92	10:45:03	INSERT INTO pedido VALUES(16, 2389.23, '2019-03-11', 1, 5)	1 row(s) affected	0.000 sec

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Read Only Context Help Snippets

Object Info Session

Exported resultset to C:\Users\Miran\OneDrive\Documents\UVM\Sistemas operativos\tabla 1.csv

10:45 a. m. 10/04/2025

MySQL Workbench

Local instance MySQL92 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Administration
- Schemas

Information

No object selected

sys_pedido

Find

1 • SELECT *

2 FROM pedido

3 ORDER BY fecha;

4 DROP DATABASE IF EXISTS ventas;

Result Grid

id	total	fecha	id_cliente	id_comercial
10	250.45	2015-06-27	8	2
7	5760	2015-09-10	2	1
6	2400.6	2016-07-27	7	1
4	110.5	2016-08-17	8	3
11	75.29	2016-08-17	3	7
2	270.65	2016-09-10	1	5
9	2480.4	2016-10-10	8	3
14	145.82	2017-02-02	6	1
12	3045.6	2017-04-25	2	1
5	948.5	2017-09-10	5	2
1	150.5	2017-10-05	5	2
3	65.26	2017-10-05	2	1
8	1983.43	2017-10-10	4	6

Exported resultset to C:\Users\Miran\OneDrive\Documents\UVM\Sistemas operativos\tabla 1.csv

Output

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

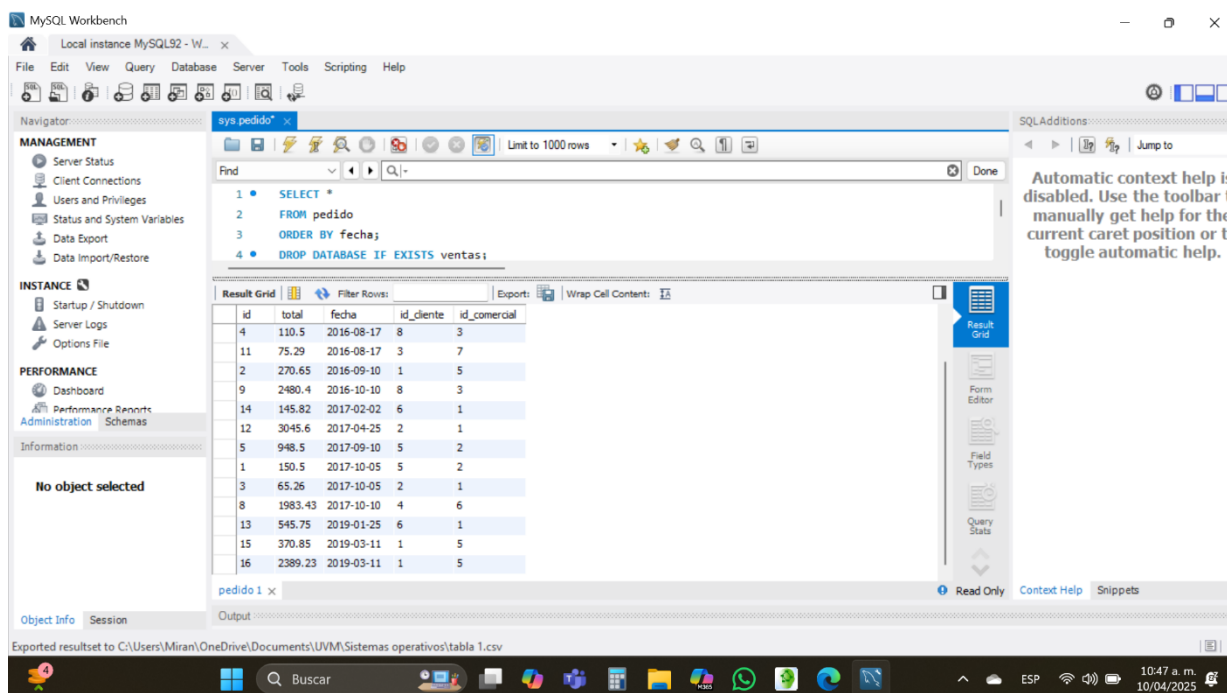
Read Only Context Help Snippets

Object Info Session

Exported resultset to C:\Users\Miran\OneDrive\Documents\UVM\Sistemas operativos\tabla 1.csv

10:47 a. m. 10/04/2025





b) Listado de todos los clientes.

```
SELECT *
FROM cliente;
DROP DATABASE IF EXISTS ventas;
CREATE DATABASE ventas CHARACTER SET utf8mb4;
USE 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),
  categoria 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)
);

```

```

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);

```



MySQL Workbench

Local instance MySQL92 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Administration
- Schemas

Information

No object selected

cliente 2 x

sys_pedido

Find

1 • SELECT *

2 • FROM cliente;

3 • DROP DATABASE IF EXISTS ventas;

4 • CREATE DATABASE ventas CHARACTER SET utf8mb4;

Result Grid

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

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Output

Action Output

#	Time	Action	Message	Duration / Fetch
128	10:48:07	INSERT INTO pedido VALUES(11, 75.29, '2016-08-17', 3, 7)	1 row(s) affected	0.000 sec
129	10:48:07	INSERT INTO pedido VALUES(12, 3045.6, '2017-04-25', 2, 1)	1 row(s) affected	0.015 sec
130	10:48:07	INSERT INTO pedido VALUES(13, 545.75, '2019-01-25', 6, 1)	1 row(s) affected	0.000 sec
131	10:48:07	INSERT INTO pedido VALUES(14, 145.82, '2017-02-02', 6, 1)	1 row(s) affected	0.000 sec
132	10:48:07	INSERT INTO pedido VALUES(15, 370.85, '2019-03-11', 1, 5)	1 row(s) affected	0.016 sec

Read Only Context Help Snippets

Active schema was cleared

10:48 a. m. 10/04/2025

MySQL Workbench

Local instance MySQL92 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Administration
- Schemas

Information

No object selected

cliente 2 x

sys_pedido

Find

1 • SELECT *

2 • FROM cliente;

3 • DROP DATABASE IF EXISTS ventas;

4 • CREATE DATABASE ventas CHARACTER SET utf8mb4;

5 • USE ventas;

6

7 • CREATE TABLE cliente (

Result Grid

id	nombre	apellido1	apellido2	ciudad	categoria
1	Aarón	Rivero	Gómez	Almería	100
2	Adela	Salas	Díaz	Granada	200
3	Adolfo	Rubio	Flores	Sevilla	100
4	Adrián	Suárez		Jaén	300
5	Marcos	Loyola	Méndez	Almería	200
6	Maria	Santana	Moreno	Cádiz	100
7	Pilar	Ruiz		Sevilla	300
8	Pepe	Ruiz	Santana	Huelva	200
9	Guillermo	López	Gómez	Granada	225
10	Daniel	Santana	Loyola	Sevilla	125

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Output

Read Only Context Help Snippets

Active schema was cleared

10:48 a. m. 10/04/2025



- c) Listado con el identificador, nombre y los apellidos de todos los clientes que han realizado algún pedido.

Listado con el identificador, nombre y los apellidos de todos los clientes que han realizado algún pedido.
<pre>SELECT DISTINCT c.id, c.nombre, c.apellido1, c.apellido2 FROM cliente c JOIN pedido p ON c.id = p.id_cliente;</pre>

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

- d) Listado que muestre todos los clientes, con todos los pedidos que han realizado y con nombre y apellido de los comerciales asociados.

Listado que muestre todos los clientes, con todos los pedidos que han realizado y con nombre y apellido de los comerciales asociados.
<pre>SELECT clientes.nombre AS nombre_cliente, clientes.apellido1 AS apellido1_cliente, clientes.apellido2 AS apellido2_cliente, pedidos.id_pedido, pedidos.total, pedidos.fecha, comerciales.nombre AS nombre_comercial, comerciales.apellido1 AS apellido1_comercial, comerciales.apellido2 AS apellido2_comercial FROM pedidos INNER JOIN clientes ON pedidos.id_cliente = clientes.id INNER JOIN comerciales ON pedidos.id_comercial = comerciales.id ORDER BY clientes.id, pedidos.id_pedido;</pre>



id_pedido	total	fecha	id_cliente	id_comercial	nombre_cliente	apellido1_cliente	apellido2_cliente	nombre_comercial	apellido1_comercial	apellido2_comercial
2	270.65	2016-09-10	1	5	Aarón	Rivero	Gómez	Antonio	Carretero	Ortega
15	370.85	2019-03-11	1	5	Aarón	Rivero	Gómez	Antonio	Carretero	Ortega
16	2389.23	2019-03-11	1	5	Aarón	Rivero	Gómez	Antonio	Carretero	Ortega
3	65.26	2017-10-05	2	1	Adela	Salas	Díaz	Daniel	Sáez	Vega
7	5760	2015-09-10	2	1	Adela	Salas	Díaz	Daniel	Sáez	Vega
12	3045.6	2017-04-25	2	1	Adela	Salas	Díaz	Daniel	Sáez	Vega
11	75.29	2016-08-17	3	7	Adolfo	Rubio	Flores	Antonio	Vega	Hernández
8	1983.43	2017-10-10	4	6	Adrián	Suárez	NULL	Manuel	Domínguez	Hernández
5	948.5	2017-09-10	5	2	Marcos	Loyola	Méndez	Juan	Gómez	López
13	545.75	2019-01-25	6	1	María	Santana	Moreno	Daniel	Sáez	Vega
14	145.82	2017-02-02	6	1	María	Santana	Moreno	Daniel	Sáez	Vega
6	2400.6	2016-07-27	7	1	Pilar	Ruiz	NULL	Daniel	Sáez	Vega
4	110.5	2016-08-17	8	3	Pepe	Ruiz	Santana	Diego	Flores	Salas
9	2480.4	2016-10-10	8	3	Pepe	Ruiz	Santana	Diego	Flores	Salas
10	250.45	2015-06-27	8	2	Pepe	Ruiz	Santana	Juan	Gómez	López

e) Devuelve un listado con todos los pedidos que ha realizado “Adela Salas Díaz”

```
SELECT
p.id AS pedido_id,
p.total AS pedido_total,
p.fecha AS pedido_fecha,
c.nombre AS cliente_nombre,
c.apellido1 AS cliente_apellido1,
c.apellido2 AS cliente_apellido2
FROM pedido p
INNER JOIN cliente c ON p.id_cliente = c.id
WHERE c.nombre = 'Adela'
AND c.apellido1 = 'Salas'
AND c.apellido2 = 'Díaz';
```



The screenshot shows a SQL IDE interface with a query editor and a results grid. The query is as follows:

```

93 p.total AS pedido_total,
94 p.fecha AS pedido_fecha,
95 co.nombre AS comercial_nombre,
96 co.apellido1 AS comercial_apellidomaterno,
97 co.apellido2 AS comercial_apellidopaterno
98 FROM cliente c
99 INNER JOIN pedido p ON c.id = p.id_cliente
100 INNER JOIN comercial co ON p.id_comercial = co.id;
101
102 SELECT
103 p.id AS pedido_id,
104 p.total AS pedido_total,
105 p.fecha AS pedido_fecha,
106 c.nombre AS cliente_nombre,
107 c.apellido1 AS cliente_apellido1,
108 c.apellido2 AS cliente_apellido2
109 FROM pedido p
110 INNER JOIN cliente c ON p.id_cliente = c.id
111 WHERE c.nombre = 'Adela'
112 AND c.apellido1 = 'Salas'
113 AND c.apellido2 = 'Díaz';

```

The results grid shows the following data:

pedido_id	pedido_to...	pedido_fecha	cliente_nomb...	cliente_apellid...	cliente_apellid...
3	65.26	2017-10-05	Adela	Salas	Díaz
7	5760	2015-09-10	Adela	Salas	Díaz
12	3045.6	2017-04-25	Adela	Salas	Díaz

The status bar at the bottom indicates "Query Completed" and "16 row(s) returned".

f) Devuelve el número de pedidos en los que ha participado el comercial Daniel Sáez Vega

```

SELECT
    co.nombre AS comercial_nombre,
    co.apellido1 AS comercial_apellidomaterno,
    co.apellido2 AS comercial_apellidopaterno,
    COUNT(*) AS total_pedidos
FROM pedido p
JOIN comercial co ON p.id_comercial = co.id
WHERE co.nombre = 'Daniel'
    AND co.apellido1 = 'Sáez'
    AND co.apellido2 = 'Vega';

```



The screenshot shows a MySQL IDE interface. The top panel displays a SQL query with line numbers 107 to 125. The query involves an INNER JOIN between a 'cliente' table and a 'pedido' table, filtering for a specific client and then counting the total orders for a specific commercial entity.

```

107 c.apellido1 AS cliente_apellido1,
108 c.apellido2 AS cliente_apellido2
109 FROM pedido p
110 INNER JOIN cliente c ON p.id_cliente = c.id
111 WHERE c.nombre = 'Adela'
112 AND c.apellido1 = 'Salas'
113 AND c.apellido2 = 'Díaz';
114
115 SELECT
116 co.nombre AS comercial_nombre,
117 co.apellido1 AS comercial_apellidomaterno,
118 co.apellido2 AS comercial_apellidopaterno,
119 COUNT(*) AS total_pedidos
120 FROM pedido p
121 JOIN comercial co ON p.id_comercial = co.id
122 WHERE co.nombre = 'Daniel'
123 AND co.apellido1 = 'Sáez'
124 AND co.apellido2 = 'Vega';
125

```

The bottom panel shows the 'Result Grid' with the following data:

comercial_nombre	comercial_apellidomaterno	comercial_apellidopaterno	total_pedidos
Daniel	Sáez	Vega	6

Below the result grid, the 'Action Output' panel shows a log of SQL actions and their execution times.

	Time	Action	Response	Duration / Fetch Time
1	23:54:24	SELECT c.id AS cliente_id, c.nombre AS cliente_nombre, c.apellido1 AS cliente_apellidomaterno, c.apellido2 AS...	16 row(s) returned	0.00094 sec / 0.000...
2	00:01:23	SELECT p.id AS pedido_id, p.total AS pedido_total, p.fecha AS pedido_fecha, c.nombre AS cliente_nombre, c.a...	3 row(s) returned	0.0011 sec / 0.00001...
3	00:03:16	SELECT * FROM pedido LIMIT 0, 1000	16 row(s) returned	0.0011 sec / 0.00001...
4	00:06:43	SELECT p.id AS pedido_id, p.total AS pedido_total, p.fecha AS pedido_fecha, c.nombre AS cliente_nombre, c.a...	3 row(s) returned	0.00081 sec / 0.0000...
5	00:17:44	SELECT COUNT(*) AS total_pedidos FROM pedido p INNER JOIN comercial c ON p.id_comercial = c.id WHERE c.no...	1 row(s) returned	0.0036 sec / 0.00001...

Genera el *script.sql* para cada una de tus consultas y toma captura de pantalla del resultado obtenido.

Si tienes algún problema con los comandos los siguientes materiales pueden servirte de apoyo además de la bibliografía revisada con anterioridad:

Tutoriales Ya (s.f.) *MySQL Ya desde cero* [Sitio Web]. Recuperado de <https://www.tutorialesprogramacionya.com/mysql/ya/>

MySQL AB, (s.f.) *Tutorial Básico MySQL* [Archivo PDF]. Recuperado de <http://www.v-espino.com/~chema/daw1/tutoriales/tutorial-mysql.pdf>

3. Desarrolla un reporte en el que incluyas los siguientes puntos:

- Planteamiento: Explica la funcionalidad de tu base de datos.
- Procedimiento: Documenta tu procedimiento desde la creación de base de datos, llenado de tablas y consultas realizadas. Para las consultas integra, enunciado de la consulta, consulta en *sql* y captura de pantalla del resultado obtenido.

4. Redacta una conclusión en la que describas la utilidad de las consultas.



5. Elabora el reporte en este mismo documento.
6. Genera una carpeta .ZIP donde integres tu reporte y tu *script.sql* que contenga todo el código desarrollado.
7. Al finalizar esta actividad, vuelve a la plataforma y sigue los pasos que se indican para enviar tu trabajo.

* * *

Conclusiones

Miranda Velázquez Mariles:

Las consultas realizadas son muy útiles porque nos permiten obtener información clara y organizada sobre los pedidos y los clientes de la empresa, al listar los pedidos por fecha, podemos ver cómo han ido cambiando las ventas con el tiempo, identificar los momentos de mayor actividad y hacer un seguimiento más fácil de cada operación; por otro lado, tener un listado completo de todos los clientes nos ayuda a conocer mejor a las personas con las que trabajamos, lo cual es importante para mantener una buena relación con ellos y ofrecerles un mejor servicio. En general, este tipo de consultas hacen que sea más fácil tomar decisiones, llevar un buen control del negocio y mejorar la organización de la información.

Para poder concluir con esta actividad realmente me entretiene mucho el estar pensando que va y que no, me resulta aún un poco confuso cual es la funcionalidad de cada sentencia y considero que se desarrolla más conocimiento en la práctica por lo que me gustaría seguir desarrollando estos conocimientos ya que si me dejan sin mis apuntes difícilmente podría desarrollar yo solito en una hoja en blanco todo el procedimiento que se acabó de realizar, por lo que me queda muy claro que tengo que ser más autodidacta y más en este ámbito laboral en el que vamos a concluir en esta materia. Además, al observar la clase veo que hay compañeros muy jóvenes que tienen bastante conocimiento en el manejo de estas sentencias por lo que hay camino por recorrer. Este ejercicio mediante el uso de los filtros que nos permitió unir información de distintas tablas creadas me resulto bastante similar con las tablas dinámicas en Excel, que que buscamos información específica para un resultado en concreto.



Jose Emiliano Jauregui Guzman:

Para poder concluir con esta actividad realmente me entretiene mucho el estar pensando que va y que no, me resulta aún un poco confuso cual es la funcionalidad de cada sentencia y considero que se desarrolla más conocimiento en la práctica por lo que me gustaría seguir desarrollando estos conocimientos ya que si me dejan sin mis apuntes difícilmente podría desarrollar yo solito en una hoja en blanco todo el procedimiento que se acabó de realizar, por lo que me queda muy claro que tengo que ser más autodidacta y más en este ámbito laboral en el que vamos a concluir en esta materia. Además, al observar la clase veo que hay compañeros muy jóvenes que tienen bastante conocimiento en el manejo de estas sentencias por lo que hay camino por recorrer. Este ejercicio mediante el uso de los filtros que nos permitió unir información de distintas tablas creadas me resulto bastante similar con las tablas dinámicas en Excel, que que buscamos información específica para un resultado en concreto.

Alfonso Roberto Gómez Rosales:

En un mundo cada vez más impulsado por los datos, dominar un mayor número de consultas en SQL no es solo una habilidad técnica, sino una herramienta esencial para tomar decisiones informadas. Mientras más consultas conocemos, más formas tenemos de interactuar con la información, analizarla desde distintos ángulos y descubrir patrones que de otra manera pasarían desapercibidos. Esto nos permite ahorrar tiempo, automatizar tareas repetitivas y, sobre todo, obtener respuestas precisas a preguntas complejas.

Además, ampliar nuestro repertorio de consultas fortalece nuestra capacidad de adaptarnos a diferentes escenarios y bases de datos. No se trata solo de saber “extraer” información, sino de entender cómo está estructurada, cómo se relaciona y cómo puede transformarse para generar valor. Por eso, seguir aprendiendo y practicando nuevas consultas en SQL es clave para crecer profesionalmente, resolver problemas reales y mantenernos vigentes en un entorno digital cada vez más competitivo.



Referencias APA

- Sánchez, J. (2020). *Ejercicios. Relización de consultas SQL* [Sitio Web] Recuperado de <https://josejuansanchez.org/bd/ejercicios-consultas-sql/index.html>
- Virtualab ITP (15 de Octubre de 2014). Tutorial Consultas en MySQL Workbench. Recuperado de <https://www.youtube.com/watch?v=YYSxy40WdAQ>
- Pulido, E., Escobar, O., Núñez, J. (2019). *Base de Datos*. <https://es.scribd.com/document/728742385/BaseDatos-Pulido-Escobar-Nunez-eBook>
- García Velázquez, G. (2023). Estructura básica de una consulta. Universidad Contemporánea de las Américas. Recuperado de <https://uniclanet.unicla.edu.mx/assets/tareas/6709220240211134422.pdf>
- Universidad Autónoma del Estado de Hidalgo. (s.f.). 5.4 Formulación de Consultas con SQL. Recuperado de https://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro14/54_formulacin_de_consultas_con_sql.html

