

EJERCICIO BD

ASIGNATURA LINEA DE INVESTIGACIÓN III

JUAN MANUEL NAVARRO CORDOBA

561220150

**UNIVERSIDAD DE CUNDINAMARCA EXTENSIÓN CHIA PROGRAMA DE
INGENIERÍA DE SISTEMAS FACULTAD DE INGENIERÍA**

2025

Contenido

introducción	3
objetivo	3
1. Creación del contenedor	3
2. Ingresar al contenedor y correr mysql	4
3. Creación de la bd	5
4. Creación de las tablas	5
5. Insertar 10 registros en cada tabla	6
6. Listado de todos los pedidos, ordenados por fecha en orden descendente	8
7. Los dos pedidos de mayor valor	8
8. Identificadores de clientes que han realizado algún pedido, sin que aparezcan repetidos	9
9. La comisión más alta registrada en la tabla comercial	9
10. Listado de los nombres de los clientes que empiezan por a	10
11. Devuelve el id, nombre y apellidos de los clientes que han hecho pedidos, sin repetidos y en orden alfabético	11
12. Listado de todos los pedidos junto con los datos completos de los clientes, ordenando el resultado alfabéticamente por los datos del cliente	11
13. Todos los clientes, sus pedidos y los datos de los comerciales asociados a cada pedido 12	
14. Listado de todos los clientes, incluyendo los que no han realizado ningún pedido, junto con los datos de los pedidos (si existen)	13
15. La suma total de todas las cantidades registradas en la tabla pedido	14
16. Cantidad media (promedio) de todos los pedidos en la tabla pedido	15
17. Mayor cantidad registrada en la tabla pedido	15
18. Menor cantidad registrada en la tabla pedido	16
19. Digrama ER	16
conclusión	17
fuentes	17

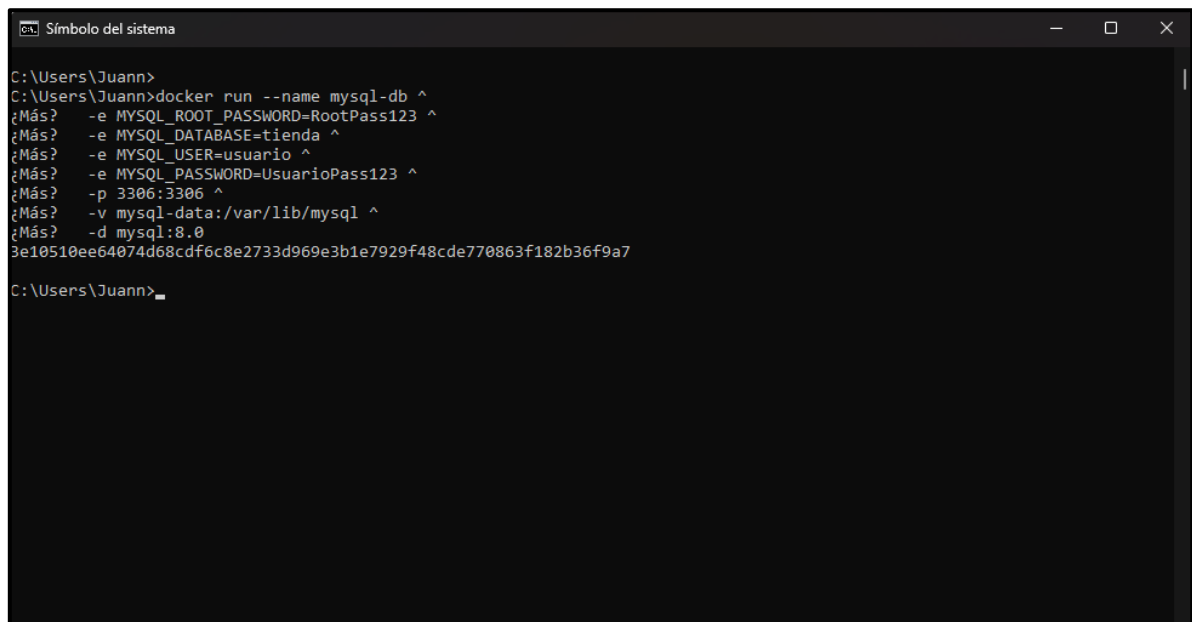
Introducción

El presente taller tiene como finalidad la construcción de un entorno de base de datos utilizando tecnologías de virtualización de contenedores. Para ello, se implementó un contenedor en Docker con el motor de base de datos MySQL, en el cual se configuró y desplegó la base de datos tienda. A partir de este entorno se crearon las tablas, relaciones y registros de prueba, garantizando la integridad referencial del modelo de datos. Asimismo, se ejecutaron diversas consultas SQL orientadas al manejo de información, permitiendo afianzar conocimientos sobre la creación, manipulación y consulta de bases de datos relacionales.

Objetivo

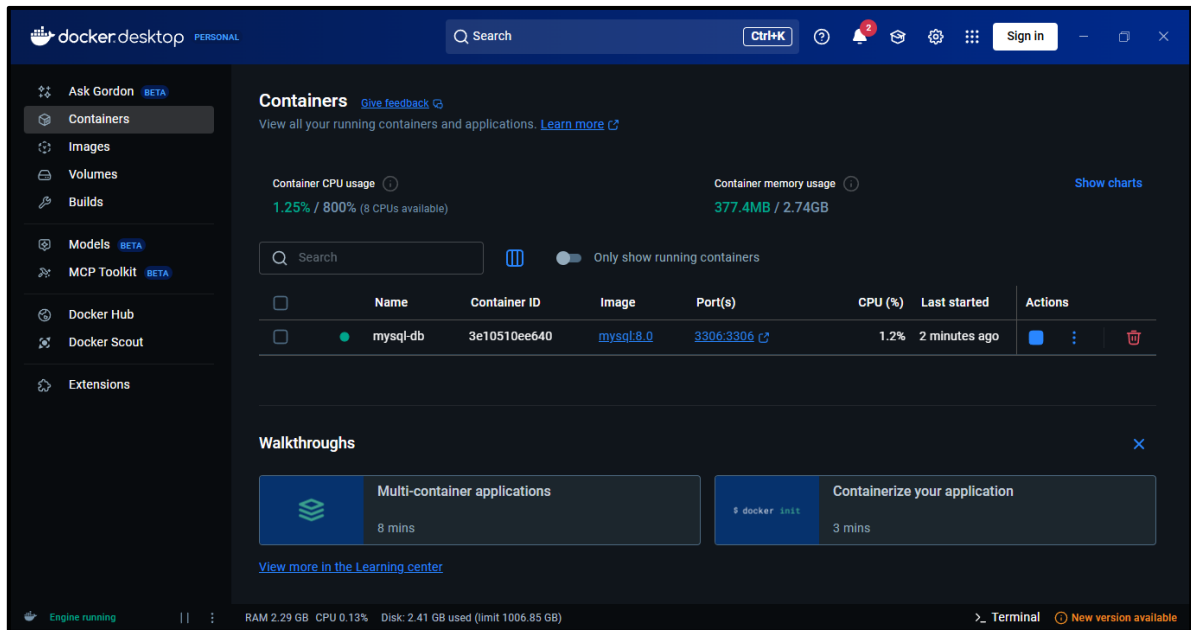
Implementar un entorno de gestión de bases de datos en MySQL dentro de un contenedor Docker, realizando la creación de la base de datos, las tablas y sus relaciones, así como la inserción de registros de prueba y la ejecución de consultas SQL que permitan validar el correcto funcionamiento del modelo de datos y fortalecer competencias en el manejo de información relacional.

1. Creación del contenedor



```
C:\Users\Juann> docker run --name mysql-db ^
¿Más? -e MYSQL_ROOT_PASSWORD=RootPass123 ^
¿Más? -e MYSQL_DATABASE=tienda ^
¿Más? -e MYSQL_USER=usuario ^
¿Más? -e MYSQL_PASSWORD=UsuarioPass123 ^
¿Más? -p 3306:3306 ^
¿Más? -v mysql-data:/var/lib/mysql ^
¿Más? -d mysql:8.0
3e10510ee64074d68cdf6c8e2733d969e3b1e7929f48cde770863f182b36f9a7

C:\Users\Juann>
```



2. Ingresar al contenedor y correr MySQL

```
Símbolo del sistema - docker exec -it mysql-db mysql -uroot -p

C:\Users\Juann>docker exec -it mysql-db mysql -uroot -p
# Cuando pida password: escribe RootPass123 (o tu contraseña)
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.43 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

3. Creación de la BD

```
Símbolo del sistema - docker exec -it mysql-db mysql -uroot -p
mysql> CREATE DATABASE IF NOT EXISTS tienda;
Query OK, 1 row affected, 1 warning (0.11 sec)

mysql> _
```

4. Creación de las tablas

```
Símbolo del sistema - docker exec -it mysql-db mysql -uroot -p

mysql> USE tienda;
Database changed
mysql> SHOW TABLES;
Empty set (0.00 sec)

mysql> CREATE TABLE comercial (
->   id INT(10) AUTO_INCREMENT PRIMARY KEY,
->   nombre VARCHAR(100) NOT NULL,
->   apellido1 VARCHAR(100),
->   apellido2 VARCHAR(100),
->   ciudad VARCHAR(100),
->   comision FLOAT
-> );
Query OK, 0 rows affected, 1 warning (0.17 sec)

mysql> CREATE TABLE cliente (
->   id INT(10) AUTO_INCREMENT PRIMARY KEY,
->   nombre VARCHAR(100) NOT NULL,
->   apellido1 VARCHAR(100),
->   apellido2 VARCHAR(100),
->   ciudad VARCHAR(100),
->   categoria INT(10)
-> );
Query OK, 0 rows affected, 2 warnings (0.11 sec)
```

```
Símbolo del sistema - docker exec -it mysql-db mysql -uroot -p
mysql> CREATE TABLE pedido (
->   id INT(10) AUTO_INCREMENT PRIMARY KEY,
->   cantidad DOUBLE,
->   fecha DATE,
->   id_cliente INT(10),
->   id_comercial INT(10),
->   FOREIGN KEY (id_cliente) REFERENCES cliente(id)
->     ON UPDATE CASCADE
->     ON DELETE RESTRICT,
->   FOREIGN KEY (id_comercial) REFERENCES comercial(id)
->     ON UPDATE CASCADE
->     ON DELETE RESTRICT
-> );
Query OK, 0 rows affected, 3 warnings (0.17 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_tienda |
+-----+
| cliente           |
| comercial          |
| pedido            |
+-----+
3 rows in set (0.01 sec)

mysql>
```

5. Insertar 10 registros en cada tabla

Tabla comercial

```
Símbolo del sistema - docker exec -it mysql-db mysql -uroot -p
mysql> INSERT INTO comercial (nombre, apellido1, apellido2, ciudad, comision) VALUES
-> ('Carlos', 'Perez', 'Lopez', 'Bogot', 5.5),
-> ('Lucia', 'Gomez', 'Martinez', 'Medelln', 4.2),
-> ('Andres', 'Ruiz', 'Castao', 'Cali', 6.0),
-> ('Paola', 'Sanchez', 'Moreno', 'Cartagena', 3.8),
-> ('Jorge', 'Torres', 'Gonzalez', 'Barranquilla', 7.1),
-> ('Beatriz', 'Mora', 'Diaz', 'Bucaramanga', 2.9),
-> ('Felipe', 'Lopez', 'Ramirez', 'Pereira', 5.0),
-> ('Diana', 'Castro', 'Sosa', 'Santa Marta', 4.7),
-> ('Miguel', 'Rojas', 'Alvarez', 'Tunja', 6.3),
-> ('Camila', 'Fernandez', 'Jimenez', 'Manizales', 3.5);
Query OK, 10 rows affected (0.42 sec)
Records: 10 Duplicates: 0 Warnings: 0

mysql>
```

Tabla cliente

```
Símbolo del sistema - docker exec -it mysql-db mysql -uroot -p

mysql> INSERT INTO cliente (nombre, apellido1, apellido2, ciudad, categoria) VALUES
-> ('Adrian', 'Mora', 'Perez', 'Bogot', 1),
-> ('Pedro', 'Lopez', 'Garcia', 'Medelln', 2),
-> ('Allan', 'Torres', 'Ruiz', 'Cali', 3),
-> ('Juan', 'Diaz', 'Martinez', 'Cartagena', 1),
-> ('Beatriz', 'Sosa', 'Alvarez', 'Barranquilla', 2),
-> ('Carolina', 'Ramos', 'Suarez', 'Bucaramanga', 1),
-> ('Santiago', 'Gomez', 'Muoz', 'Pereira', 2),
-> ('Laura', 'Vargas', 'Jimenez', 'Santa Marta', 3),
-> ('David', 'Castao', 'Mora', 'Tunja', 2),
-> ('Valentina', 'Moreno', 'Castro', 'Manizales', 1);
Query OK, 10 rows affected (0.05 sec)
Records: 10 Duplicates: 0 Warnings: 0

mysql> _
```

Tabla pedido

```
Símbolo del sistema - docker exec -it mysql-db mysql -uroot -p

mysql> INSERT INTO pedido (cantidad, fecha, id_cliente, id_comercial) VALUES
-> (250.00, '2025-09-01', 1, 1),
-> (180.50, '2025-09-02', 2, 2),
-> (300.00, '2025-09-03', 3, 3),
-> (75.00, '2025-09-04', 4, 4),
-> (500.00, '2025-09-05', 5, 5),
-> (220.75, '2025-09-06', 6, 6),
-> (150.00, '2025-09-07', 7, 7),
-> (90.00, '2025-09-08', 8, 8),
-> (350.00, '2025-09-09', 9, 9),
-> (410.00, '2025-09-10', 10, 10);
Query OK, 10 rows affected (0.03 sec)
Records: 10 Duplicates: 0 Warnings: 0

mysql> _
```

6. listado de todos los pedidos, ordenados por fecha en orden descendente

```
SELECT id, cantidad, fecha, id_cliente, id_comercial
FROM pedido
ORDER BY fecha DESC;
```

```
Símbolo del sistema - docker exec -it mysql-db mysql -uroot -p
-> (90.00, '2025-09-08', 8, 8),
-> (350.00, '2025-09-09', 9, 9),
-> (410.00, '2025-09-10', 10, 10);
Query OK, 10 rows affected (0.03 sec)
Records: 10 Duplicates: 0 Warnings: 0

mysql> SELECT id, cantidad, fecha, id_cliente, id_comercial
-> FROM pedido
-> ORDER BY fecha DESC;
+-----+-----+-----+-----+-----+
| id | cantidad | fecha      | id_cliente | id_comercial |
+-----+-----+-----+-----+-----+
| 10 | 410      | 2025-09-10 | 10         | 10           |
| 9  | 350      | 2025-09-09 | 9          | 9            |
| 8  | 90       | 2025-09-08 | 8          | 8            |
| 7  | 150      | 2025-09-07 | 7          | 7            |
| 6  | 220.75   | 2025-09-06 | 6          | 6            |
| 5  | 500      | 2025-09-05 | 5          | 5            |
| 4  | 75       | 2025-09-04 | 4          | 4            |
| 3  | 300      | 2025-09-03 | 3          | 3            |
| 2  | 180.5    | 2025-09-02 | 2          | 2            |
| 1  | 250      | 2025-09-01 | 1          | 1            |
+-----+-----+-----+-----+-----+
10 rows in set (0.02 sec)

mysql>
```

7. los dos pedidos de mayor valor

```
SELECT id, cantidad, fecha, id_cliente, id_comercial
FROM pedido
ORDER BY cantidad DESC
LIMIT 2;
```

```
Símbolo del sistema - docker exec -it mysql-db mysql -uroot -p

mysql> SELECT id, cantidad, fecha, id_cliente, id_comercial
-> FROM pedido
-> ORDER BY cantidad DESC
-> LIMIT 2;
+-----+-----+-----+-----+-----+
| id | cantidad | fecha      | id_cliente | id_comercial |
+-----+-----+-----+-----+-----+
| 5  | 500      | 2025-09-05 | 5          | 5            |
| 10 | 410      | 2025-09-10 | 10         | 10           |
+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql>
```


8. identificadores de clientes que han realizado algún pedido, sin que aparezcan repetidos

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

```
Símbolo del sistema - docker exec -it mysql-db mysql -uroot -p

mysql> SELECT DISTINCT c.id, c.nombre, c.apellido1, c.apellido2
-> FROM cliente c
-> JOIN pedido p ON c.id = p.id_cliente;
+-----+-----+-----+-----+
| id | nombre | apellido1 | apellido2 |
+-----+-----+-----+-----+
| 1 | Adrian | Mora | Perez |
| 2 | Pedro | Lopez | Garcia |
| 3 | Allan | Torres | Ruiz |
| 4 | Juan | Diaz | Martinez |
| 5 | Beatriz | Sosa | Alvarez |
| 6 | Carolina | Ramos | Suarez |
| 7 | Santiago | Gomez | Muoz |
| 8 | Laura | Vargas | Jimenez |
| 9 | David | Castao | Mora |
| 10 | Valentina | Moreno | Castro |
+-----+-----+-----+-----+
10 rows in set (0.03 sec)

mysql> _
```

9. la comisión más alta registrada en la tabla comercial

```
SELECT MAX(comision) AS mayor_comision
FROM comercial;
```

```
Símbolo del sistema - docker exec -it mysql-db mysql -uroot -p

mysql> SELECT MAX(comision) AS mayor_comision
-> FROM comercial;
+-----+
| mayor_comision |
+-----+
| 7.1 |
+-----+
1 row in set (0.01 sec)

mysql> _
```

10. Listado de los nombres de los clientes que empiezan por A

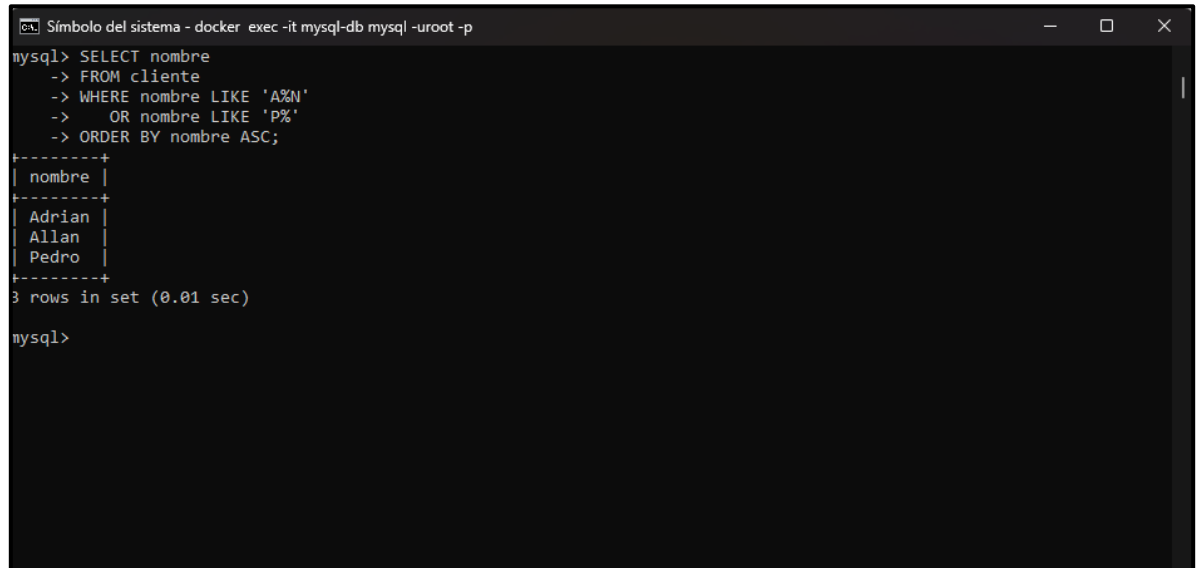
SELECT nombre, apellido1, apellido2

FROM cliente

WHERE nombre **LIKE** 'A%N'

OR nombre **LIKE** 'P%'

ORDER BY nombre **ASC**;



The screenshot shows a terminal window titled "Símbolo del sistema - docker exec -it mysql-db mysql -uroot -p". The prompt is "mysql>". The user enters the following SQL query:

```
mysql> SELECT nombre
-> FROM cliente
-> WHERE nombre LIKE 'A%N'
-> OR nombre LIKE 'P%'
-> ORDER BY nombre ASC;
```

The output is a table with one column named "nombre". The results are:

nombre
Adrian
Allan
Pedro

Below the table, it says "3 rows in set (0.01 sec)". The prompt "mysql>" is visible again at the bottom.

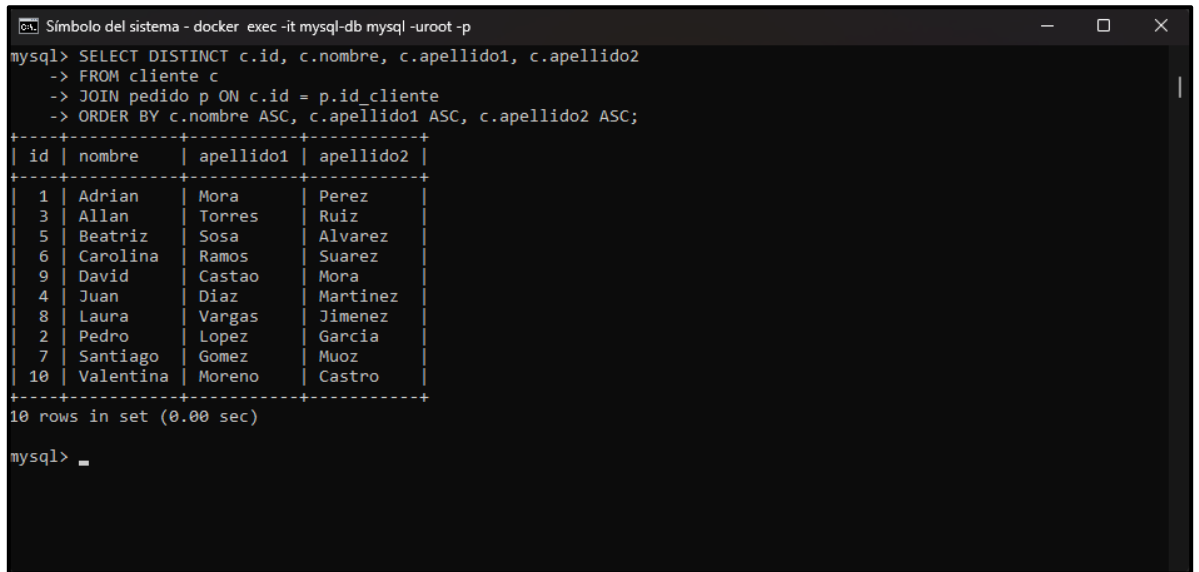
11. Devuelve el id, nombre y apellidos de los clientes que han hecho pedidos, sin repetidos y en orden alfabético

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

FROM cliente c

JOIN pedido p ON c.id = p.id_cliente

ORDER BY c.nombre ASC, c.apellido1 ASC, c.apellido2 ASC;
```



The screenshot shows a terminal window with a MySQL prompt. The query executed is: `SELECT DISTINCT c.id, c.nombre, c.apellido1, c.apellido2 FROM cliente c JOIN pedido p ON c.id = p.id_cliente ORDER BY c.nombre ASC, c.apellido1 ASC, c.apellido2 ASC;`. The results are displayed in a table with 10 rows and 4 columns: id, nombre, apellido1, and apellido2. The rows are ordered alphabetically by name and then by surnames.

id	nombre	apellido1	apellido2
1	Adrian	Mora	Perez
3	Allan	Torres	Ruiz
5	Beatriz	Sosa	Alvarez
6	Carolina	Ramos	Suarez
9	David	Castao	Mora
4	Juan	Diaz	Martinez
8	Laura	Vargas	Jimenez
2	Pedro	Lopez	Garcia
7	Santiago	Gomez	Muoz
10	Valentina	Moreno	Castro

10 rows in set (0.00 sec)

12. listado de todos los pedidos junto con los datos completos de los clientes, ordenando el resultado alfabéticamente por los datos del cliente

```
SELECT c.id AS id_cliente,

       c.nombre AS nombre_cliente,

       c.apellido1 AS apellido1_cliente,

       c.apellido2 AS apellido2_cliente,

       c.ciudad AS ciudad_cliente,

       c.categoria AS categoria_cliente,

       p.id AS id_pedido,

       p.cantidad,

       p.fecha,

       p.id_comercial

FROM cliente c

JOIN pedido p ON c.id = p.id_cliente

ORDER BY c.nombre ASC, c.apellido1 ASC, c.apellido2 ASC;
```

```

Simbolo del sistema - docker exec -it mysql-db mysql -uroot -p
-> c.ciudad AS ciudad_cliente,
-> c.categoria AS categoria_cliente,
-> p.id AS id_pedido,
-> p.cantidad,
-> p.fecha,
-> p.id_comercial
-> FROM cliente c
-> JOIN pedido p ON c.id = p.id_cliente
-> ORDER BY c.nombre ASC, c.apellido1 ASC, c.apellido2 ASC;

```

id_cliente	nombre_cliente	apellido1_cliente	apellido2_cliente	ciudad_cliente	categoria_cliente	id_pedido	cantidad	fecha	id_comercial
1	Adrian	Mora	Perez	Bogot	1	1	250	2025-09-01	1
3	Allan	Torres	Ruiz	Cali	3	3	300	2025-09-03	3
5	Beatriz	Sosa	Alvarez	Barranquilla	2	5	500	2025-09-05	5
6	Carolina	Ramos	Suarez	Bucaramanga	1	6	220.75	2025-09-06	6
9	David	Castao	Mora	Tunja	2	9	350	2025-09-09	9
4	Juan	Diaz	Martinez	Cartagena	1	4	75	2025-09-04	4
8	Laura	Vargas	Jimenez	Santa Marta	3	8	90	2025-09-08	8
2	Pedro	Lopez	Garcia	Medelln	2	2	180.5	2025-09-02	2
7	Santiago	Gomez	Muoz	Pereira	2	7	150	2025-09-07	7
10	Valentina	Moreno	Castro	Manizales	1	10	410	2025-09-10	10

```

10 rows in set (0.01 sec)

mysql> _

```

13. Todos los clientes, sus pedidos y los datos de los comerciales asociados a cada pedido

```

SELECT c.id AS id_cliente,

       c.nombre AS nombre_cliente,

       c.apellido1 AS apellido1_cliente,

       c.apellido2 AS apellido2_cliente,

       c.ciudad AS ciudad_cliente,

       c.categoria AS categoria_cliente,

       p.id AS id_pedido,

       p.cantidad,

       p.fecha,

       com.id AS id_comercial,

       com.nombre AS nombre_comercial,

       com.apellido1 AS apellido1_comercial,

       com.apellido2 AS apellido2_comercial,

       com.ciudad AS ciudad_comercial,

       com.comision AS comision_comercial

FROM cliente c

JOIN pedido p ON c.id = p.id_cliente

JOIN comercial com ON p.id_comercial = com.id

```

ORDER BY c.nombre ASC, c.apellido1 ASC, c.apellido2 ASC;

```

mysql> SELECT c.id AS id_cliente,
-> c.nombre AS nombre_cliente,
-> c.apellido1 AS apellido1_cliente,
-> c.apellido2 AS apellido2_cliente,
-> c.ciudad AS ciudad_cliente,
-> c.categoria AS categoria_cliente,
-> p.id AS id_pedido,
-> p.cantidad,
-> p.fecha,
-> com.id AS id_comercial,
-> com.nombre AS nombre_comercial,
-> com.apellido1 AS apellido1_comercial,
-> com.apellido2 AS apellido2_comercial,
-> com.ciudad AS ciudad_comercial,
-> com.comision AS comision_comercial
-> FROM cliente c
-> JOIN pedido p ON c.id = p.id_cliente
-> JOIN comercial com ON p.id_comercial = com.id
-> ORDER BY c.nombre ASC, c.apellido1 ASC, c.apellido2 ASC;

```

id_cliente	nombre_cliente	apellido1_cliente	apellido2_cliente	ciudad_cliente	categoria_cliente	id_pedido	cantidad	fecha	id_comercial	nombre_comercial	apellido1_comercial	apellido2_comercial	ciudad_comercial	comision_comercial
1	Adrian	Mora	Perez	Bogot	5.5	1	1	2025-09-01	1	Carlos	Perez	Lopez	Bogot	5.5
3	Allan	Torres	Ruiz	Calli	6	3	3	2025-09-03	3	Andres	Ruiz	Castao	Calli	6
5	Beatriz	Sosa	Alvarez	Barranquilla	7.1	2	5	2025-09-05	5	Jorge	Torres	Gonzalez	Barranquilla	7.1
6	Carolina	Ramos	Suarez	Bucaramanga	2.9	1	6	220.75	2025-09-06	6	Beatriz	Mora	Diaz	Bucaramanga
9	David	Castao	Mora	Tunja	6.3	2	9	350	2025-09-09	9	Higuel	Rojas	Alvarez	Tunja
4	Juan	Diaz	Martinez	Cartagena	3.8	1	4	75	2025-09-04	4	Paola	Sanchez	Moreno	Cartagena
8	Laura	Vargas	Jimenez	Santa Marta	4.7	3	8	90	2025-09-08	8	Diana	Castro	Sosa	Lopez
2	Pedro	Lopez	Garcia	Medelln	4.2	2	2	180.5	2025-09-02	2	Lucia	Gomez	Martinez	Medelln
7	Santiago	Gomez	Muoz	Pereira	5	2	7	150	2025-09-07	7	Felipe	Lopez	Ramirez	Pereira

14. listado de todos los clientes, incluyendo los que no han realizado ningún pedido, junto con los datos de los pedidos (si existen)

```

SELECT c.id AS id_cliente,

c.nombre AS nombre_cliente,

c.apellido1 AS apellido1_cliente,

c.apellido2 AS apellido2_cliente,

c.ciudad AS ciudad_cliente,

c.categoria AS categoria_cliente,

p.id AS id_pedido,

p.cantidad,

p.fecha,

p.id_comercial

FROM cliente c

LEFT JOIN pedido p ON c.id = p.id_cliente

ORDER BY c.apellido1 ASC, c.apellido2 ASC, c.nombre ASC;

```

```
Símbolo del sistema - docker exec -it mysql-db mysql -uroot -p

mysql> SELECT c.id AS id_cliente,
-> c.nombre AS nombre_cliente,
-> c.apellido1 AS apellido1_cliente,
-> c.apellido2 AS apellido2_cliente,
-> c.ciudad AS ciudad_cliente,
-
-> p.fecha,
-> p.id_comercial
-> FROM cliente c
-> LEFT JOIN pedido p ON c.id = p.id_cliente
-> ORDER BY c.apellido1 ASC, c.apellido2 ASC, c.nombre ASC;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id_cliente | nombre_cliente | apellido1_cliente | apellido2_cliente | ciudad_cliente | categoria_cliente | id_pedido | cantidad | fecha | id_comercial |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 9 | David | Castao | Mora | Tunja | 2 | 9 | 350 | 2025-09-09 | 9 |
| 4 | Juan | Díaz | Martínez | Cartagena | 1 | 4 | 75 | 2025-09-04 | 4 |
| 7 | Santiago | Gomez | Muñoz | Pereira | 2 | 7 | 150 | 2025-09-07 | 7 |
| 2 | Pedro | Lopez | García | Medellín | 2 | 2 | 180.5 | 2025-09-02 | 2 |
| 1 | Adrian | Mora | Perez | Bogotá | 1 | 1 | 250 | 2025-09-01 | 1 |
| 10 | Valentina | Moreno | Castro | Manizales | 1 | 10 | 410 | 2025-09-10 | 10 |
| 6 | Carolina | Ramos | Suarez | Bucaramanga | 1 | 6 | 220.75 | 2025-09-06 | 6 |
| 5 | Beatriz | Sosa | Alvarez | Barranquilla | 2 | 5 | 500 | 2025-09-05 | 5 |
| 3 | Allan | Torres | Ruiz | Cali | 3 | 3 | 300 | 2025-09-03 | 3 |
| 8 | Laura | Vargas | Jimenez | Santa Marta | 3 | 8 | 90 | 2025-09-08 | 8 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.01 sec)

mysql>
```

15. la suma total de todas las cantidades registradas en la tabla pedido

**SELECT SUM(cantidad) AS total_pedidos
FROM pedido;**

```
Símbolo del sistema - docker exec -it mysql-db mysql -uroot -p

mysql> SELECT SUM(cantidad) AS total_pedidos
-> FROM pedido;
+-----+
| total_pedidos |
+-----+
| 2526.25 |
+-----+
1 row in set (0.01 sec)

mysql> _
```

16. cantidad media (promedio) de todos los pedidos en la tabla pedido

**SELECT AVG(cantidad) AS promedio_pedidos
FROM pedido;**

```
Símbolo del sistema - docker exec -it mysql-db mysql -uroot -p
mysql> SELECT AVG(cantidad) AS promedio_pedidos
-> FROM pedido;
+-----+
| promedio_pedidos |
+-----+
|          252.625 |
+-----+
1 row in set (0.00 sec)

mysql>
```

17. mayor cantidad registrada en la tabla pedido

**SELECT MAX(cantidad) AS mayor_cantidad
FROM pedido;**

```
Símbolo del sistema - docker exec -it mysql-db mysql -uroot -p
mysql> SELECT MAX(cantidad) AS mayor_cantidad
-> FROM pedido;
+-----+
| mayor_cantidad |
+-----+
|           500 |
+-----+
1 row in set (0.01 sec)

mysql>
```

18. menor cantidad registrada en la tabla pedido

SELECT MIN(cantidad) AS menor_cantidad

FROM pedido;

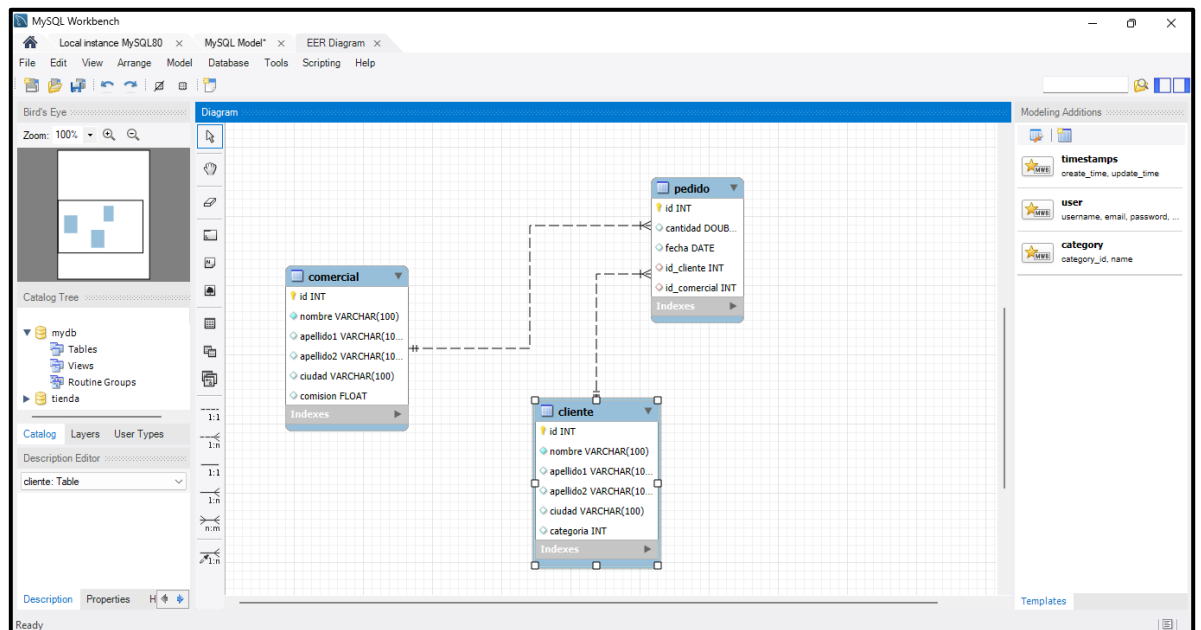
```
Símbolo del sistema - docker exec -it mysql-db mysql -uroot -p

mysql> SELECT MIN(cantidad) AS menor_cantidad
-> FROM pedido;

+-----+
| menor_cantidad |
+-----+
|              75 |
+-----+
1 row in set (0.01 sec)

mysql> _
```

19. Digrama ER



Conclusión

El desarrollo del taller permitió comprender y aplicar los fundamentos de la virtualización de servicios con Docker y la administración de información con MySQL. La construcción del modelo de base de datos y la ejecución de consultas demuestran la importancia de diseñar correctamente las estructuras y relaciones entre entidades para garantizar la integridad y consistencia de los datos. Adicionalmente, el uso de contenedores ofrece ventajas en términos de portabilidad, escalabilidad y facilidad de despliegue, consolidándose como una práctica recomendada en entornos académicos y profesionales de gestión de bases de datos.

Fuentes

<https://docs.docker.com/>

<https://dev.mysql.com/doc/>

<https://www.redhat.com/en/topics/containers/what-is-docker>