

TRABAJO PRÁCTICO – Unidad 6

SGBDR (Aplicación de DDL y DML)

Alumno: Ignacio Figueroa – 45.406.120

Tecnicatura Universitaria en Programación – UTN

Materia: Base de Datos I

Comisión: 8

Introducción:

En este trabajo se implementa una base de datos relacional completa a partir de un modelo conceptual.

El objetivo es aplicar sentencias **DDL** (para crear estructuras de datos) y **DML** (para cargar y consultar datos), trabajando con **MySQL** mediante **XAMPP** y **Workbench**.

phpMyAdmin

The screenshot shows the phpMyAdmin configuration page. On the left, a sidebar lists databases: Nueva, information_schema, mysql, performance_schema, phpmyadmin, and test. The main area has four tabs: Configuraciones generales, Configuraciones de apariencia, Servidor de base de datos, and phpMyAdmin. The 'Configuraciones generales' tab shows the connection cotejamiento as utf8mb4_unicode_ci. The 'Configuraciones de apariencia' tab shows the language as Español - Spanish. The 'Servidor de base de datos' tab displays server details: Servidor: 127.0.0.1 via TCP/IP, Tipo de servidor: MariaDB, Conexión del servidor: No se está utilizando SSL, Versión del servidor: 10.4.32-MariaDB - mariadb.org binary distribution, Versión del protocolo: 10, Usuario: root@localhost, and Conjunto de caracteres del servidor: UTF-8 Unicode (utf8mb4). The 'phpMyAdmin' tab provides links to version information, documentation, and support.

Actividad 1: Base de Datos Librería

- Creación de tablas

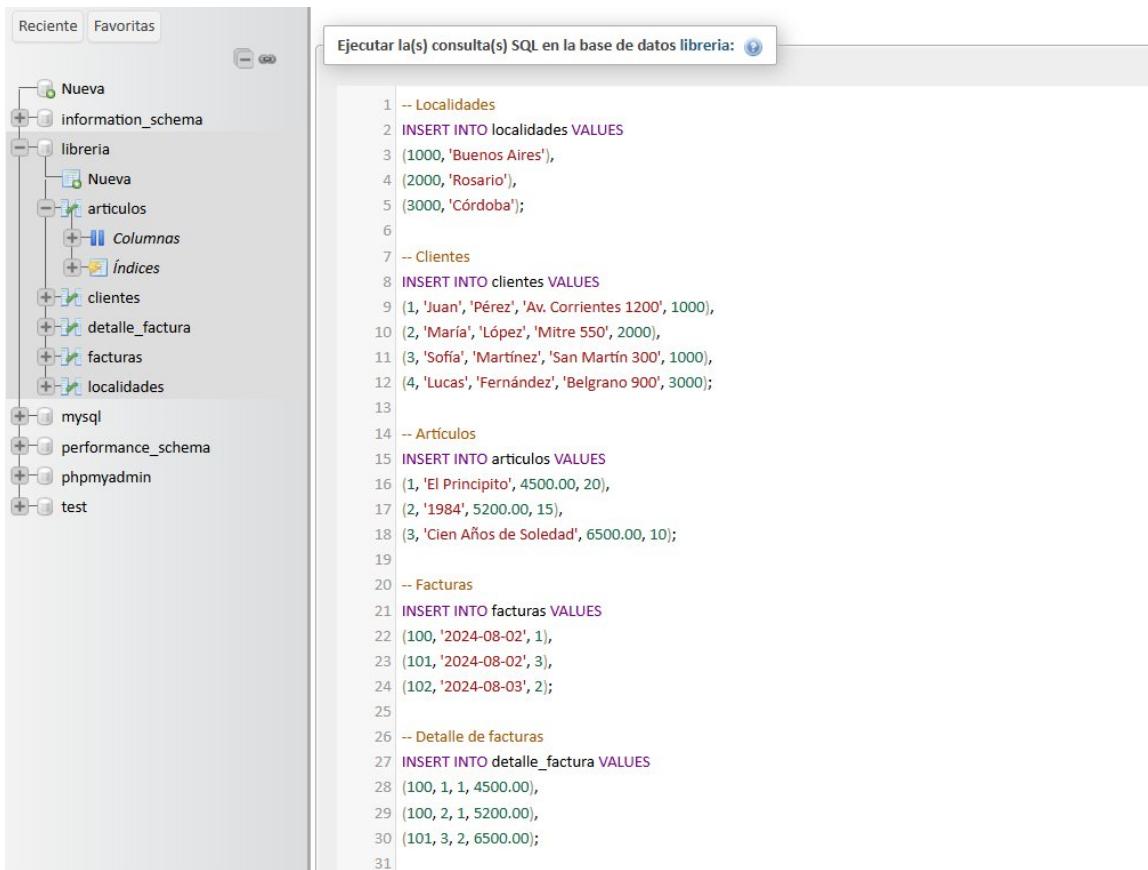
The screenshot shows the SQL editor in phpMyAdmin. The left sidebar shows the 'libreria' database selected. The SQL code pane contains the following SQL statements:

```

1 create table localidades (
2   CódigoPostal int primary key,
3   Nombre varchar(100) not null
4 ) ENGINE=INNODB;
5
6 create table clientes (
7   CódigoCliente int primary key,
8   Nombre varchar(50) not null,
9   Apellido varchar(50) not null,
10  Domicilio varchar(100),
11  CódigoPostal INT,
12  foreign key (CódigoPostal) references localidades(CódigoPostal)
13 ) ENGINE=InnoDB;
14
15 create table artículos (
16  CódigoArtículo int primary key,
17  Descripción varchar(100) not null,
18  Precio decimal(10,2) not null,
19  Stock int not null
20 ) ENGINE=InnoDB;
21
22 create table facturas (
23  NúmeroFactura int primary key,
24  Fecha date not null,
25  CódigoCliente int,
26  foreign key (CódigoCliente) references clientes(CódigoCliente)
27 ) ENGINE=InnoDB;
28
29 create table detalle_factura (
30  NúmeroFactura int,
31  CódigoArtículo int,
32  Cantidad int not null,
33  PrecioUnitario decimal(10,2) not null,
34  primary key (NúmeroFactura, CódigoArtículo),
35  foreign key (NúmeroFactura) references facturas(NúmeroFactura),
36  foreign key (CódigoArtículo) references artículos(CódigoArtículo)
37 ) ENGINE=InnoDB;

```

- Inserción de datos (DML)



```

Reciente | Favoritas

Nueva
information_schema
libreria
    Nueva
        articulos
            Columnas
            Índices
        clientes
        detalle_factura
        facturas
        localidades
mysql
performance_schema
phpmyadmin
test

Ejecutar la(s) consulta(s) SQL en la base de datos libreria: ⚙️

1 -- Localidades
2 INSERT INTO localidades VALUES
3 (1000, 'Buenos Aires'),
4 (2000, 'Rosario'),
5 (3000, 'Córdoba');
6
7 -- Clientes
8 INSERT INTO clientes VALUES
9 (1, 'Juan', 'Pérez', 'Av. Corrientes 1200', 1000),
10 (2, 'María', 'López', 'Mitre 550', 2000),
11 (3, 'Sofía', 'Martínez', 'San Martín 300', 1000),
12 (4, 'Lucas', 'Fernández', 'Belgrano 900', 3000);
13
14 -- Artículos
15 INSERT INTO articulos VALUES
16 (1, 'El Principito', 4500.00, 20),
17 (2, '1984', 5200.00, 15),
18 (3, 'Cien Años de Soledad', 6500.00, 10);
19
20 -- Facturas
21 INSERT INTO facturas VALUES
22 (100, '2024-08-02', 1),
23 (101, '2024-08-02', 3),
24 (102, '2024-08-03', 2);
25
26 -- Detalle de facturas
27 INSERT INTO detalle_factura VALUES
28 (100, 1, 1, 4500.00),
29 (100, 2, 1, 5200.00),
30 (101, 3, 2, 6500.00);
31

```

Consulta pedida en el práctico:

Clientes que compraron el 02/08/2024

```

1 SELECT DISTINCT c.Nombre, c.Apellido
2 FROM clientes c
3 JOIN facturas f ON c.CodigoCliente = f.CodigoCliente
4 WHERE f.Fecha = '2024-08-02';

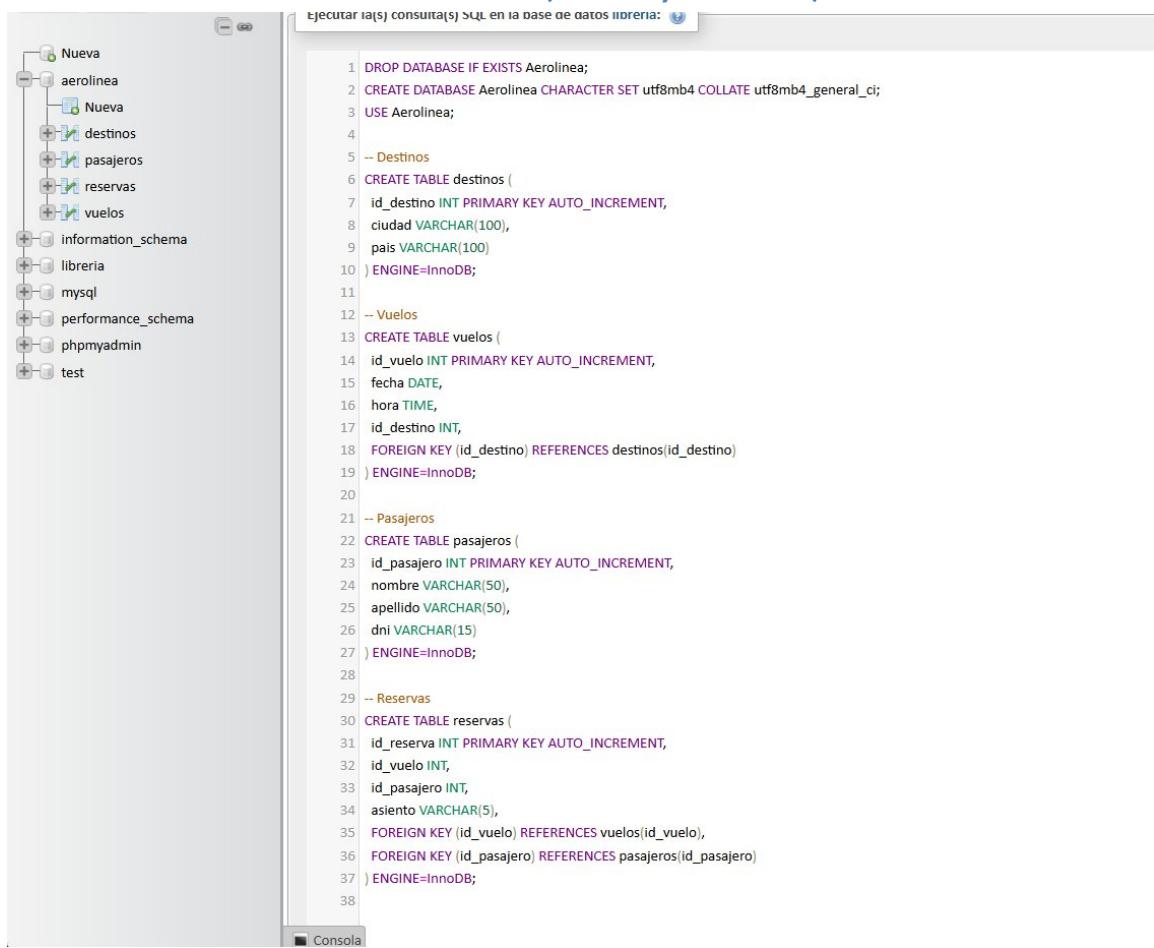
```

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** A tree view on the left lists schemas: Nueva, information_schema, libreria, articulos, facturas, localidades, mysql, performance_schema, phpmyadmin, and test.
- Query Results:** The main window displays a query result set with two rows of data. The columns are labeled "Nombre" and "Apellido". The data is as follows:

Nombre	Apellido
Juan	Pérez
Sofía	Martínez
- SQL Statement:** The SQL statement used is: `SELECT DISTINCT c.Nombre, c.Apellido FROM clientes c JOIN facturas f ON c.CodigoCliente = f.CodigoCliente WHERE f.Fecha = '2024-08-02';`
- Buttons and Options:** The interface includes buttons for "Perfilendo" (Profile), "Explicar SQL" (Explain SQL), "Crear código PHP" (Create PHP code), and "Actualizar" (Update). It also features filters for "Mostrar todo" (Show all), "Número de filas" (Number of rows), "Filtrar filas" (Filter rows), "Buscar en esta tabla" (Search in this table), and "Ordenar según la clave" (Sort by key). There are also buttons for "Imprimir" (Print), "Copiar al portapapeles" (Copy to clipboard), "Exportar" (Export), "Mostrar gráfico" (Show graphic), and "Crear vista" (Create view).
- Save Options:** At the bottom, there are buttons for "Guardar esta consulta en favoritos" (Save this query as a favorite) and "Etiqueta:" (Label:).

Actividad 2: Base de Datos Aerolínea (Vuelos y Reservas)



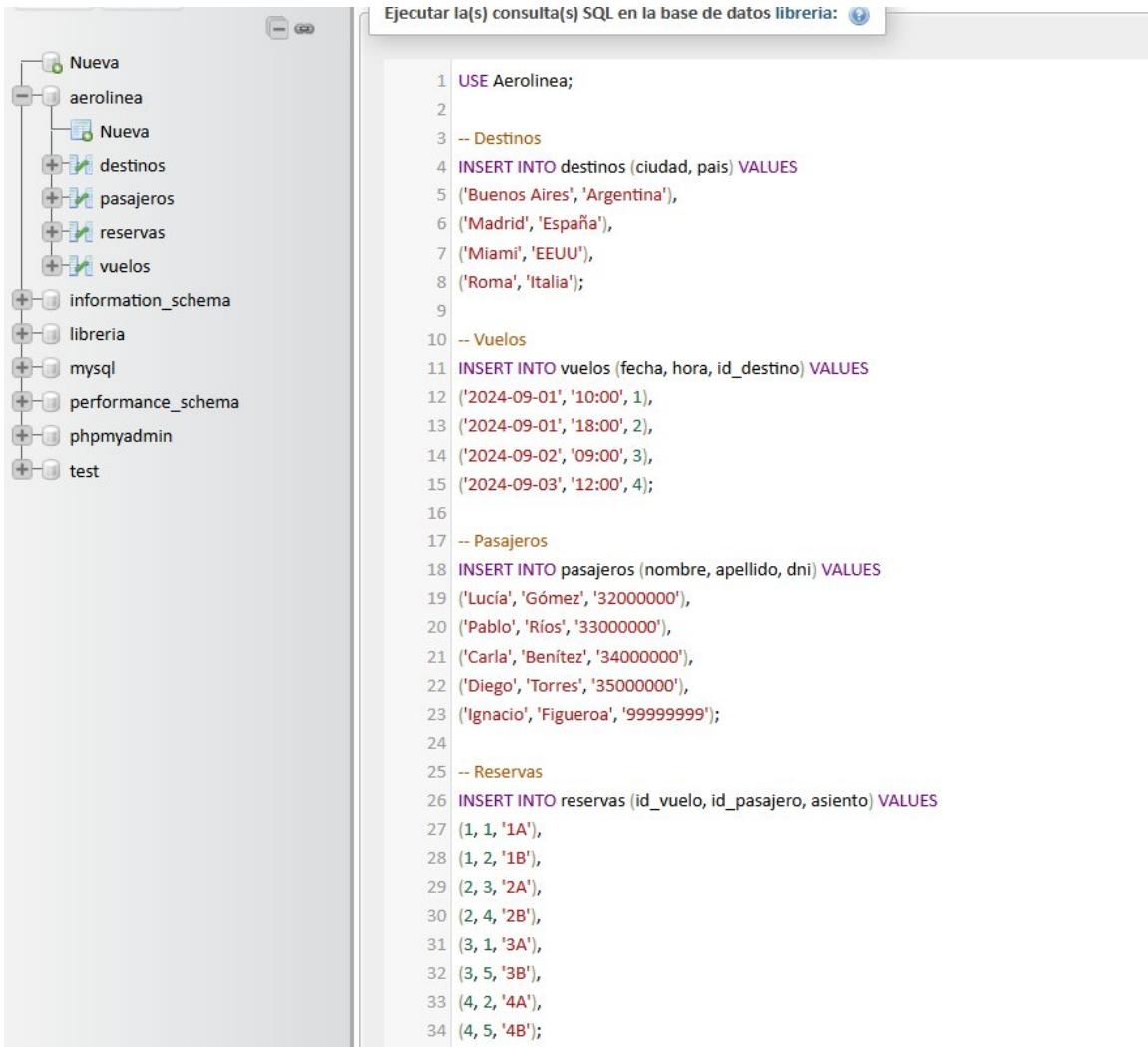
The screenshot shows the MySQL Workbench interface. On the left, the database tree displays the schema 'aerolinea' with tables 'destinos', 'pasajeros', 'reservas', and 'vuelos'. On the right, the 'Consola' tab contains the following SQL code:

```

1 DROP DATABASE IF EXISTS Aerolinea;
2 CREATE DATABASE Aerolinea CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
3 USE Aerolinea;
4
5 -- Destinos
6 CREATE TABLE destinos (
7     id_destino INT PRIMARY KEY AUTO_INCREMENT,
8     ciudad VARCHAR(100),
9     pais VARCHAR(100)
10 ) ENGINE=InnoDB;
11
12 -- Vuelos
13 CREATE TABLE vuelos (
14     id_vuelo INT PRIMARY KEY AUTO_INCREMENT,
15     fecha DATE,
16     hora TIME,
17     id_destino INT,
18     FOREIGN KEY (id_destino) REFERENCES destinos(id_destino)
19 ) ENGINE=InnoDB;
20
21 -- Pasajeros
22 CREATE TABLE pasajeros (
23     id_pasajero INT PRIMARY KEY AUTO_INCREMENT,
24     nombre VARCHAR(50),
25     apellido VARCHAR(50),
26     dni VARCHAR(15)
27 ) ENGINE=InnoDB;
28
29 -- Reservas
30 CREATE TABLE reservas (
31     id_reserva INT PRIMARY KEY AUTO_INCREMENT,
32     id_vuelo INT,
33     id_pasajero INT,
34     asiento VARCHAR(5),
35     FOREIGN KEY (id_vuelo) REFERENCES vuelos(id_vuelo),
36     FOREIGN KEY (id_pasajero) REFERENCES pasajeros(id_pasajero)
37 ) ENGINE=InnoDB;
38

```

Carga de datos:



The screenshot shows the MySQL Workbench interface. On the left, a tree view displays the database schema:

- Nueva aerolinea
 - Nueva destinos
 - Nueva pasajeros
 - Nueva reservas
 - Nueva vuelos
- information_schema
- libreria
- mysql
- performance_schema
- phpmyadmin
- test

In the main pane, titled "Ejecutar la(s) consulta(s) SQL en la base de datos libreria:", the following SQL script is displayed:

```

1 USE Aerolinea;
2
3 -- Destinos
4 INSERT INTO destinos (ciudad, pais) VALUES
5 ('Buenos Aires', 'Argentina'),
6 ('Madrid', 'España'),
7 ('Miami', 'EEUU'),
8 ('Roma', 'Italia');
9
10 -- Vuelos
11 INSERT INTO vuelos (fecha, hora, id_destino) VALUES
12 ('2024-09-01', '10:00', 1),
13 ('2024-09-01', '18:00', 2),
14 ('2024-09-02', '09:00', 3),
15 ('2024-09-03', '12:00', 4);
16
17 -- Pasajeros
18 INSERT INTO pasajeros (nombre, apellido, dni) VALUES
19 ('Lucía', 'Gómez', '32000000'),
20 ('Pablo', 'Ríos', '33000000'),
21 ('Carla', 'Benítez', '34000000'),
22 ('Diego', 'Torres', '35000000'),
23 ('Ignacio', 'Figueroa', '99999999');
24
25 -- Reservas
26 INSERT INTO reservas (id_vuelo, id_pasajero, asiento) VALUES
27 (1, 1, '1A'),
28 (1, 2, '1B'),
29 (2, 3, '2A'),
30 (2, 4, '2B'),
31 (3, 1, '3A'),
32 (3, 5, '3B'),
33 (4, 2, '4A'),
34 (4, 5, '4B');

```

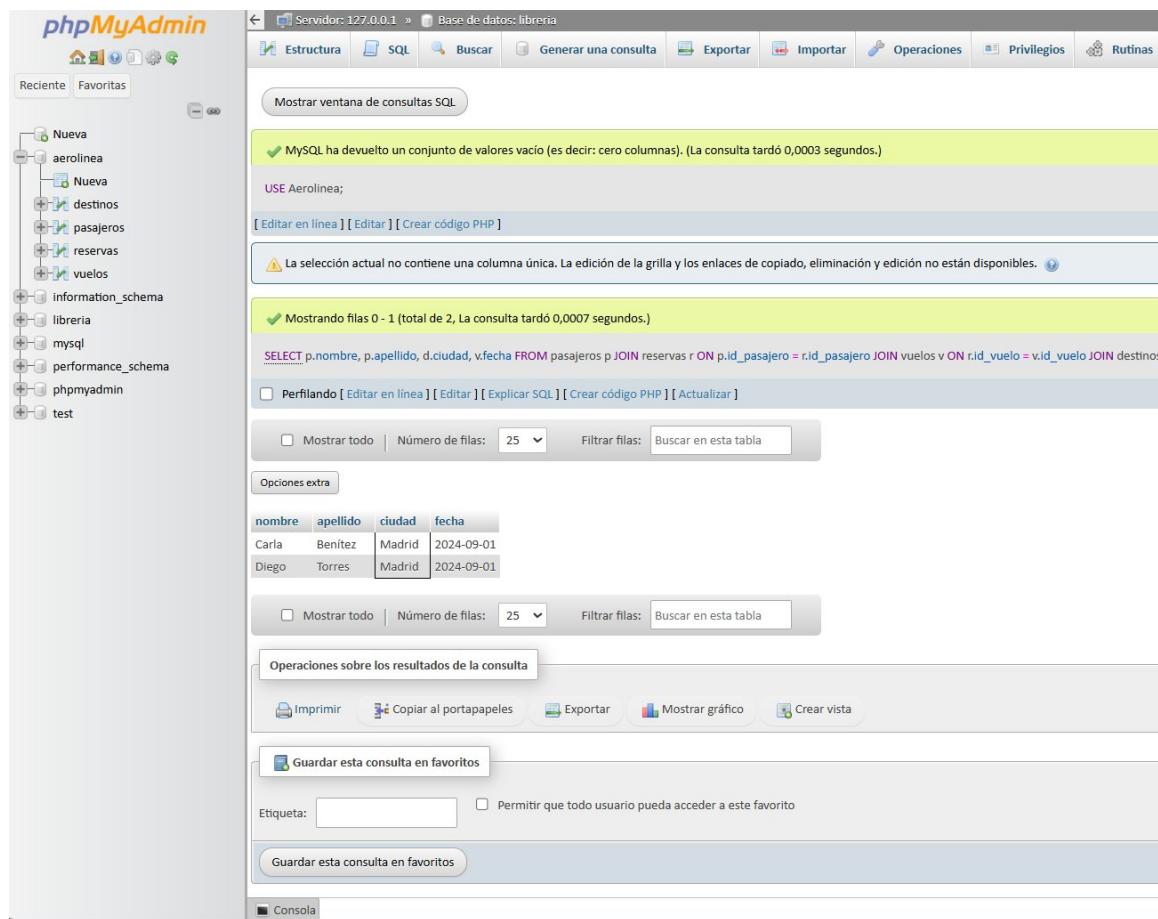
Consultas pedidas:

- a) Vuelos y pasajeros con destino “Madrid”

```

1 USE Aerolinea;
2
3 SELECT p.nombre, p.apellido, d.ciudad, v.fecha
4 FROM pasajeros p
5 JOIN reservas r ON p.id_pasajero = r.id_pasajero
6 JOIN vuelos v ON r.id_vuelo = v.id_vuelo
7 JOIN destinos d ON v.id_destino = d.id_destino
8 WHERE d.ciudad = 'Madrid';
9

```



The screenshot shows the phpMyAdmin interface with the following details:

- Servidor:** 127.0.0.1
- Base de datos:** libreria
- SQL Query:**

```
USE Aerolinea;
SELECT p.nombre, p.apellido, d.ciudad, v.fecha
FROM pasajeros p
JOIN reservas r ON p.id_pasajero = r.id_pasajero
JOIN vuelos v ON r.id_vuelo = v.id_vuelo
JOIN destinos d ON v.id_destino = d.id_destino
WHERE d.ciudad = 'Madrid';
```
- Execution Result:**
 - MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0003 segundos.)
 - Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0007 segundos.)
- Table Data:**

nombre	apellido	ciudad	fecha
Carla	Benitez	Madrid	2024-09-01
Diego	Torres	Madrid	2024-09-01
- Operations:**
 - Imprimir, Copiar al portapapeles, Exportar, Mostrar gráfico, Crear vista
 - Guardar esta consulta en favoritos (Etiqueta: [empty], Permitir que todo usuario pueda acceder a este favorito)

- b) Crear índice para ordenar más rápido

```
1 USE Aerolinea;  
2  
3 CREATE INDEX idx_apellido_nombre ON pasajeros(apellido, nombre);  
4
```

Verificar

```
4  
5 EXPLAIN SELECT * FROM pasajeros ORDER BY apellido, nombre;  
6 |
```

Crear vista ordenada:

```
6  
7 CREATE OR REPLACE VIEW vista_pasajeros_orden AS  
8 SELECT * FROM pasajeros ORDER BY nombre;
```

Conclusión:

En este trabajo se aplicaron comandos **DDL** y **DML** para crear y manejar bases de datos en **MySQL con phpMyAdmin**, utilizando claves primarias, foráneas, índices y vistas.

Además, se reforzó el uso de relaciones entre tablas mediante claves foráneas y consultas con JOIN.