

## Laboratorio 1

**PRESENTADO POR: DANIELA HURTADO USECHE**

**CC: 1113681157**

### *Parte 1: Fundamentos*

- **Breve explicación de: Qué es una base de datos.**

**R/** es un sistema de colección organizada de datos, en donde se puede recopilar, almacenar, acceder, gestionar y actualizar la información fácilmente.

Es utilizada para almacenar información de manera estructurada, permitiendo la fácil recuperación y manipulación de esta.

- **Elementos clave: tablas, registros, campos, relaciones.**

**R/ Tablas:** estructura que almacena datos organizados en filas y columnas

**Registros:** es una fila dentro de una tabla

**Campos:** es una columna de la tabla

**Relaciones:** conecta datos entre dos o más tablas

- **Diferencia entre bases de datos relacionales y no relacionales (introducción breve a modelos no relacionales).**

**R/ Relacionales:** Usan tablas con relaciones entre ellas

**No Relacionales:** Usan estructuras como documentos, claves-valor, grafos, etc.

### *Parte 2: Diseño de la base de datos*

- Diseñar un modelo para una base de datos de una librería que incluya:
- Tabla Libros: contiene ID, Título, Autor, Género, Año de publicación.
- Tabla Clientes: contiene ID, Nombre, Correo, Teléfono.

- Tabla Prestamos: contiene ID, ID del Cliente, ID del Libro, Fecha de Préstamo, Fecha de Devolución.

Bases de datos

SQL

Estado actual

Cuentas

Ejecute la(s) consulta(s) SQL en el servidor "127.0.0.1": ?

1 CREATE DATABASE LIBRERIA;

phpMyAdmin

Reciente Favoritas

Nueva

biblioteca

empresa

information\_schema

libreria

mysql

performance\_schema

phpmyadmin

test

Mostrar ventana de consultas SQL

✓ MySQL ha devuelto un conjunto de valores vacío (es decir, cero columnas) (La consulta tardó 0,0008 segundos.)

CREATE DATABASE LIBRERIA;

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

Error: #1046 Base de datos no seleccionada

Examinar

Estructura

SQL

Buscar

Insertar

Exportar

Importar

Privilegios

Operaciones

Ejecutar la(s) consulta(s) SQL en la tabla libreria.clientes: ?

1 use libreria;

2 CREATE TABLE Libros (

3 Id\_Libro INT PRIMARY KEY AUTO\_INCREMENT, Titulo VARCHAR(250) NOT NULL, Autor VARCHAR(150), Genero VARCHAR(150),

4 Año\_Publicacion INT, Disponible BOOLEAN DEFAULT TRUE

5 );

6 CREATE TABLE Clientes (

7 Id\_Cliente INT PRIMARY KEY AUTO\_INCREMENT, Nombre VARCHAR(100) NOT NULL, Correo VARCHAR(255), Telefono VARCHAR(20)

8 );

9 CREATE TABLE Prestamos (

10 Id\_Prestamo INT PRIMARY KEY AUTO\_INCREMENT, Id\_Cliente INT, Id\_Libro INT, Fecha\_Prestamo DATE, Fecha\_Devolucion DATE,

11 FOREIGN KEY (Id\_Cliente) REFERENCES Clientes(Id\_Cliente),

12 FOREIGN KEY (Id\_Libro) REFERENCES Libros(Id\_Libro)

13 );

14

```
✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0002 segundos.)

use libreria;

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0004 segundos.)

CREATE TABLE Libros ( Id_Libro INT PRIMARY KEY AUTO_INCREMENT, Titulo VARCHAR(250) NOT NULL, Autor VARCHAR(150), Genero VARCHAR(150), Año_Publicacion INT, Disponible BOOLEAN DEFAULT TRUE );

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0009 segundos.)

CREATE TABLE Clientes ( Id_Cliente INT PRIMARY KEY AUTO_INCREMENT, Nombre VARCHAR(100) NOT NULL, Correo VARCHAR(255), Telefono VARCHAR(20) );


[ Editar en línea ] [ Editar ] [ Crear código PHP ]

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0005 segundos.)

CREATE TABLE Prestamos ( Id_Prestamo INT PRIMARY KEY AUTO_INCREMENT, Id_Cliente INT, Id_Libro INT, Fecha_Prestamo DATE, Fecha_Devolucion DATE, FOREIGN KEY (Id_Cliente) REFERENCES Clientes(Id_Cliente), FOREIGN KEY (Id_Libro) REFERENCES Libros(Id_Libro) );
```

### Parte 3: Creación e instrucciones SQL

- Usar un gestor de bases de datos (ejemplo: SQLite o MySQL) para realizar:
  - Creación de las tablas mencionadas.
  - Inserción de datos ficticios (ejemplo: 5 libros, 3 clientes y 3 préstamos).



```
1 use libreria;
2
3 INSERT INTO Libros (Titulo, Autor, Genero, Año_Publicacion) VALUES
4   ('Cien Años de Soledad', 'Gabriel Garcia Marquez', 'Novela', 1967), ('Harry Potter y la Piedra Filosofal', 'JK ROWLING',
5   'Fantasia', 1997), ('1984', 'George Orwell', 'Ciencia ficción', 1949), ('La metamorfosis', 'Franz Kafka', 'Ficción', 1915),
6   ('Orgullo y prejuicio', 'Jane Austen', 'Novela', 1813);
7
8 INSERT INTO clientes (Nombre, Correo, Telefono) VALUES
9   ('Ana Perez', 'anap@yahoo.es', '3259684'), ('Luis Pinto', 'lp@hotmail.com', '35268974'), ('Lina Mora',
10  'morita@gmail.com', '32589741');
11
12 INSERT INTO prestamos (Id_Cliente, Id_Libro, Fecha_Prestamo, Fecha_Devolucion) VALUES (1, 3, '2024-08-01', '2024-08-15'),
13   (2, 2, '2024-08-02', '2024-08-16'), (3, 4, '2025-03-01', NULL);
```

```
✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0002 segundos.)

use libreria;

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

✓ 5 filas insertadas.
La id de la fila insertada es: 5 (La consulta tardó 0.0782 segundos.)

INSERT INTO Libros (Titulo, Autor, Genero, Año_Publicacion) VALUES ('Cien Años de Soledad', 'Gabriel Garcia Marquez', 'Novela', 1967), ('Harry Potter y la Piedra Filosofal',
'JK ROWLING', 'Fantasia', 1997), ('1984', 'George Orwell', 'Ciencia ficción', 1949), ('La metamorfosis', 'Franz Kafka', 'Ficción', 1915), ('Orgullo y prejuicio', 'Jane
Austen', 'Novela', 1813);

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

✓ 3 filas insertadas.
La id de la fila insertada es: 3 (La consulta tardó 0.0873 segundos.)

INSERT INTO clientes (Nombre, Correo, Telefono) VALUES ('Ana Perez', 'anap@yahoo.es', '3259684'), ('Luis Pinto', 'lp@hotmail.com', '35268974'), ('Lina Mora',
'morita@gmail.com', '32589741');

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

✓ 3 filas insertadas.
La id de la fila insertada es: 3 (La consulta tardó 0.2550 segundos.)

INSERT INTO prestamos (Id_Cliente, Id_Libro, Fecha_Prestamo, Fecha_Devolucion) VALUES (1, 3, '2024-08-01', '2024-08-15'), (2, 2, '2024-08-02', '2024-08-16'), (3, 4, '2025-03-
01', NULL);
```

- Consultas básicas en SQL:

- Mostrar todos los libros disponibles.

Ejecutar la(s) consulta(s) SQL en la tabla **libreria.clientes**: ?

```
1 SELECT * FROM Prestamos WHERE Fecha_Devolucion < CURRENT_DATE;
```

Opciones extra

						Id_Prestamo	Id_Cliente	Id_Libro	Fecha_Prestamo	Fecha_Devolucion	
<input type="checkbox"/>		Editar		Copiar		Borrar	1	1	3	2024-08-01	2024-08-15
<input type="checkbox"/>		Editar		Copiar		Borrar	2	2	2	2024-08-02	2024-08-16

- Listar los préstamos realizados por un cliente específico.

Ejecutar la(s) consulta(s) SQL en la tabla **libreria.Prestamos**: ?

```
1 SELECT * FROM Prestamos
2 JOIN Clientes ON Prestamos.Id_Cliente = Clientes.Id_Cliente
3 WHERE clientes.Nombre = 'Ana Perez';
```

Id_Prestamo	Id_Cliente	Id_Libro	Fecha_Prestamo	Fecha_Devolucion	Id_Cliente	Nombre	Correo	Telefono
1	1	3	2024-08-01	2024-08-15	1	Ana Perez	anap@yahoo.es	3259684

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

- Obtener los libros más prestados.

Ejecutar la(s) consulta(s) SQL en la tabla **libreria.Prestamos**: ?

```
1 SELECT Libros.Titulo, COUNT(*) AS VecesPrestado FROM Prestamos JOIN Libros ON Prestamos.Id_Libro = Libros.Id_Libro
2 GROUP BY Libros.Titulo
3 ORDER BY VecesPrestado DESC;
4
```

Titulo	VecesPrestado
La metamorfosis	1
1984	1
Harry Potter y la Piedra Filosofal	1

- Consultar los clientes que no han devuelto un libro.

Ejecutar la(s) consulta(s) SQL en la tabla libreria.Prestamos: ?

```
1 SELECT * FROM Prestamos
2 JOIN Clientes ON Prestamos.Id_Cliente = Clientes.Id_Cliente
3 WHERE Prestamos.Fecha_Devolucion IS null;
```

Id_Prestamo	Id_Cliente	Id_Libro	Fecha_Prestamo	Fecha_Devolucion	Id_Cliente	Nombre	Correo	Telefono
3	3	4	2025-03-01	NULL	3	Lina Mora	morita@gmail.com	32589741

Ejecutar la(s) consulta(s) SQL en la tabla libreria.Prestamos: ?

```
1 SELECT clientes.Nombre FROM Prestamos
2 JOIN Clientes ON Prestamos.Id_Cliente = Clientes.Id_Cliente
3 WHERE Prestamos.Fecha_Devolucion IS null;
```

**Nombre**

Lina Mora

## Entregable Final

Un archivo comprimido que contenga:

1. Archivo SQL con las instrucciones para crear y poblar las tablas (incluyendo consultas realizadas).
2. Capturas de pantalla que muestren la base de datos funcional y el resultado de las consultas realizadas.
3. Una breve reflexión escrita (máximo 150 palabras) sobre las ventajas y limitaciones de usar bases de datos relacionales en comparación con modelos no relacionales.

**R/** Las bases de datos relacionales ofrecen una estructura clara y organizada, las cuales se emplean cuando los datos están bien definidos y requieren integridad, como en sistemas bancarios. Gracias al uso de claves y relaciones, se puede mantener la consistencia entre tablas y realizar consultas complejas con SQL. Sin embargo, pueden volverse lentas o poco flexibles cuando se manejan grandes volúmenes de datos no estructurados o cambiantes. Ahí es donde los modelos no relacionales tienen ventaja, ya que permiten escalar fácilmente y adaptarse a distintos tipos de información, como documentos, grafos o datos en tiempo real. Aun así, suelen sacrificar integridad o requerir más lógica desde la aplicación.