

# REPORTE DE PRÁCTICA NO. ?

## ÁLGEBRA RELACIONAL Y BASES DE DATOS DISTRIBUIDAS

ALUMNO: Juan Diego Alvarado Moreno

Dr. Eduardo Cornejo-Velázquez

## Introducción

La presente práctica se centra en la aplicación del **álgebra relacional** en el contexto de **bases de datos distribuidas**. El álgebra relacional, un conjunto de operaciones matemáticas fundamentales, es esencial para manipular y consultar datos en **bases de datos relacionales**. Estas operaciones son la base teórica de lenguajes de consulta como **SQL** y son cruciales para la **fragmentación y distribución de datos** en sistemas distribuidos, asegurando la **consistencia y disponibilidad** de la información.

Las operaciones del álgebra relacional, incluyendo **selección, proyección, unión, intersección y diferencia**, permiten a los sistemas distribuidos ejecutar consultas de manera **paralela y optimizada**, mejorando así el rendimiento y la escalabilidad del sistema.

## Marco Teórico

### Álgebra Relacional

El álgebra relacional es un conjunto de operaciones matemáticas que permiten manipular y consultar datos en bases de datos relacionales. Estas operaciones son fundamentales para realizar consultas complejas de manera eficiente y constituyen la base teórica de lenguajes de consulta como SQL.

### SQL y MySQL

**SQL (Structured Query Language)** es un lenguaje de programación estándar utilizado para gestionar y manipular bases de datos relacionales. Permite a los usuarios definir, manipular y consultar datos de manera eficiente.

**MySQL** es un sistema de gestión de bases de datos relacional basado en SQL, ampliamente utilizado debido a su rendimiento, fiabilidad y facilidad de uso. Es especialmente popular en aplicaciones web y se destaca por su capacidad para manejar grandes volúmenes de datos y múltiples usuarios simultáneamente.

## Herramientas Empleadas

Para esta práctica, se utilizó **MySQL Workbench**, una herramienta de interfaz gráfica de usuario (GUI) diseñada específicamente para MySQL. Esta herramienta proporciona un entorno integrado para el diseño, modelado, generación y administración de bases de datos MySQL.

# Desarrollo de la Práctica

## Creación de Tablas e Inserción de Datos

### 4.1.1 Tabla "Employee"

```
CREATE TABLE Employee (  
    Employee_ID INT PRIMARY KEY,  
    First_name VARCHAR(50),  
    Last_name VARCHAR(50),  
    Department VARCHAR(50),  
    Salary DECIMAL(10, 2),  
    Joining_date DATE  
);
```

```
INSERT INTO Employee (Employee_ID, First_name, Last_name, Department, Salary, Joining_date)  
VALUES  
(1, 'John', 'Doe', 'HR', 60000.00, '2020-01-15'),  
(2, 'Jane', 'Smith', 'IT', 75000.00, '2019-03-12'),  
(3, 'Michael', 'Brown', 'Finance', 80000.00, '2018-07-23'),  
(4, 'Emily', 'Davis', 'Marketing', 50000.00, '2021-05-19'),  
(5, 'James', 'Wilson', 'Sales', 65000.00, '2017-10-05'),  
(6, 'Patricia', 'Garcia', 'HR', 62000.00, '2019-11-11'),  
(7, 'Robert', 'Martinez', 'IT', 72000.00, '2020-02-28');
```

#### 4.1.2 Tabla "Reward"

```
CREATE TABLE Reward (  
    Reward_ID INT PRIMARY KEY,  
    Employee_ID INT,  
    Reward_amount DECIMAL(10, 2),  
    Reward_date DATE,  
    FOREIGN KEY (Employee_ID) REFERENCES Employee(Employee_ID)  
);  
  
INSERT INTO Reward (Reward_ID, Employee_ID, Reward_amount, Reward_date)  
VALUES  
(1, 1, 1000.00, '2021-12-01'),  
(2, 2, 1500.00, '2020-11-20'),  
(3, 3, 1200.00, '2019-08-15'),  
(4, 4, 900.00, '2022-02-10');
```

## Consultas SQL

### 4.2.1 Consultas Básicas

— *Obtener todos los empleados*

**SELECT \* FROM** Employee;

Algebra Relacional:  $\pi^* (\text{Employee})$

1 • **SELECT \* FROM** Employee;




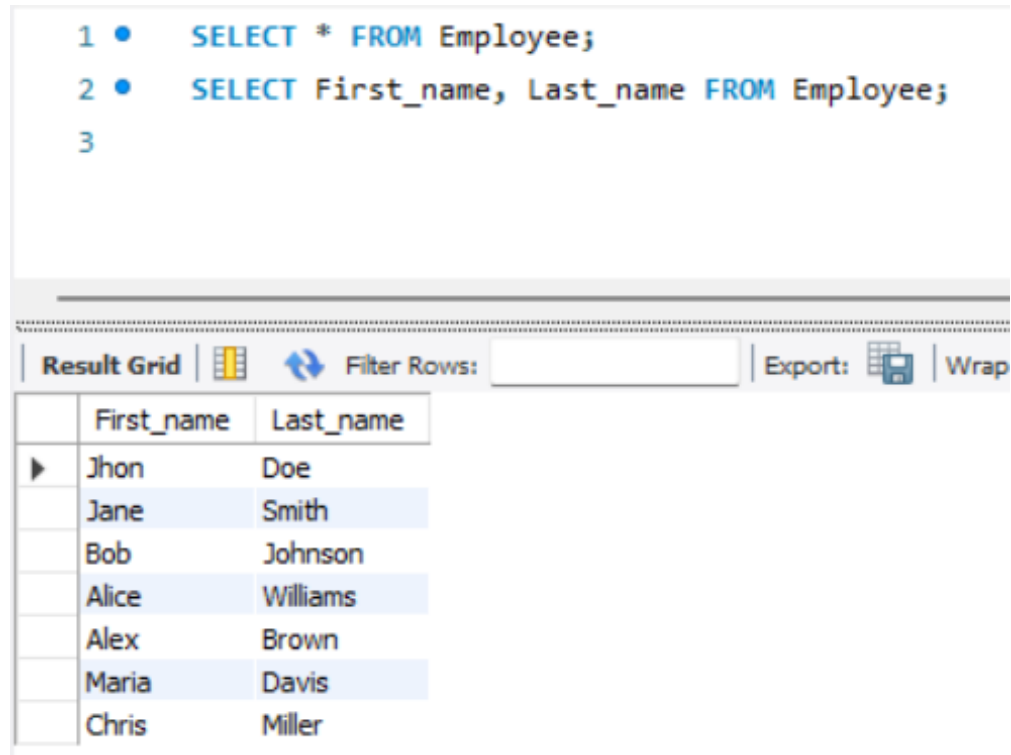
Result Grid						
Filter Rows: <input type="text"/>						
Edit:    Export/Import						
	EmployeeID	First_name	Last_name	Department	Joining_date	Salary
▶	1	Jhon	Doe	IT	2020-01-15	5000.00
	2	Jane	Smith	HR	2019-02-20	4500.00
	3	Bob	Johnson	Finance	2018-03-25	6000.00
	4	Alice	Williams	Marketing	2017-04-30	5500.00
	5	Alex	Brown	IT	2021-05-05	5200.00
	6	Maria	Davis	Sales	2016-06-10	4800.00
	7	Chris	Miller	IT	2015-07-15	6100.00
•	NULL	NULL	NULL	NULL	NULL	NULL

Figure 1: Resultado de la consulta para obtener todos los empleados

— *Obtener el primer nombre y apellido de todos los empleados*

**SELECT** First\_name, Last\_name **FROM** Employee;

Algebra Relacional: - {First\_name, Last\_name} (Employee)



The screenshot shows a database query interface. At the top, there is a list of three SQL queries. The first two are: 1. `SELECT * FROM Employee;` and 2. `SELECT First_name, Last_name FROM Employee;`. The third query is partially visible. Below the queries is a horizontal separator line. Underneath the line is a toolbar with the text "Result Grid", a grid icon, a double-headed arrow icon, the text "Filter Rows:", a text input field, the text "Export:", a floppy disk icon, and the text "Wrap". Below the toolbar is a table with two columns: "First\_name" and "Last\_name". The table contains eight rows of employee data.

	First_name	Last_name
▶	Jhon	Doe
	Jane	Smith
	Bob	Johnson
	Alice	Williams
	Alex	Brown
	Maria	Davis
	Chris	Miller

Figure 2: Resultado de la consulta para obtener nombres y apellidos



— Obtener todos los valores de la columna "First\_name" usando el alias "Nombre de empleado"  
**SELECT** First\_name **AS** "Nombre-de-empleado" **FROM** Employee;  
Algebra Relacional:  $\pi_{\text{First\_name}}(\sigma_{\text{First\_name} = \text{"Nombre-de-empleado"}}(\text{Employee}))$

```
1 • SELECT * FROM Employee;  
2 • SELECT First_name, Last_name FROM Employee;  
3 • SELECT First_name AS "Nombre de empleado" FROM Employee;  
4 |
```

	Nombre de empleado
▶	Jhon
	Jane
	Bob
	Alice
	Alex
	Maria
	Chris

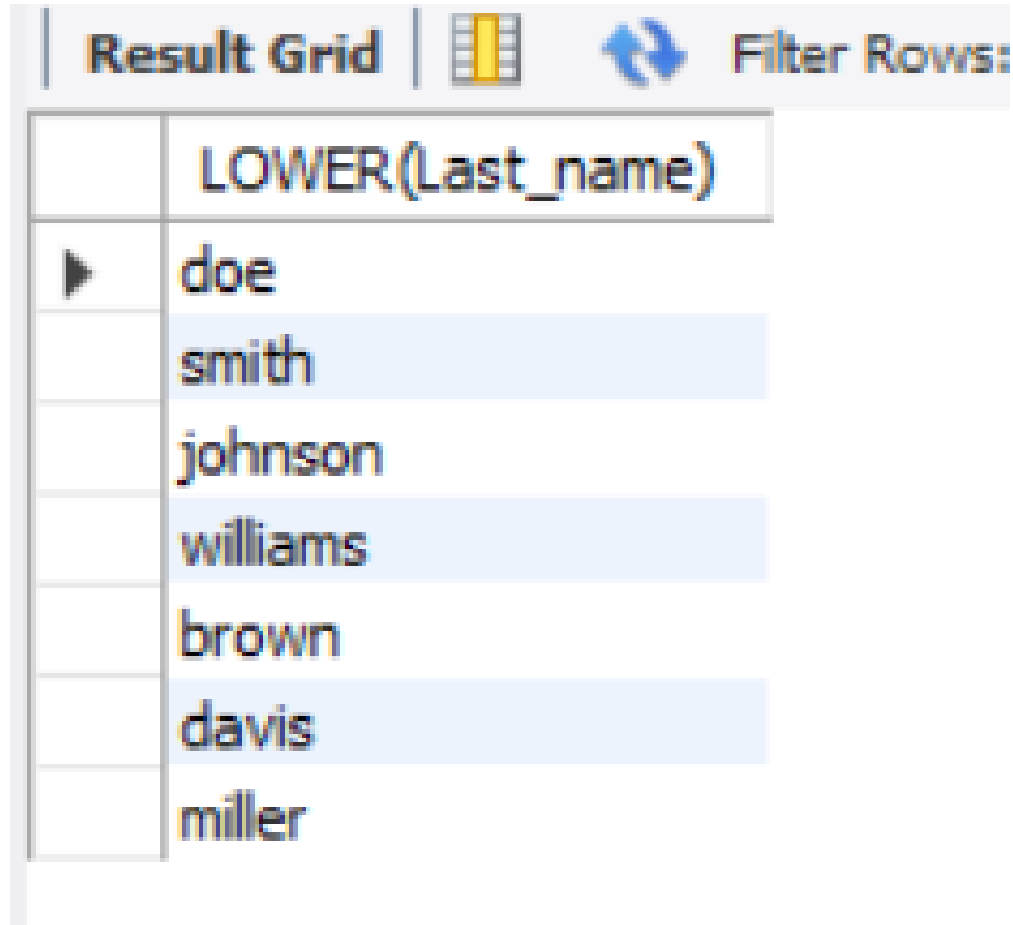
Figure 3: Resultado de la consulta con alias "Nombre de empleado"

#### 4.2.2 Manipulación de Cadenas

— *Obtener todos los valores de la columna 'Last\_name' en min sculas*

**SELECT LOWER**(Last\_name) **FROM** Employee;

Algebra Relacional: - {**LOWER**(Last\_name)} (Employee)



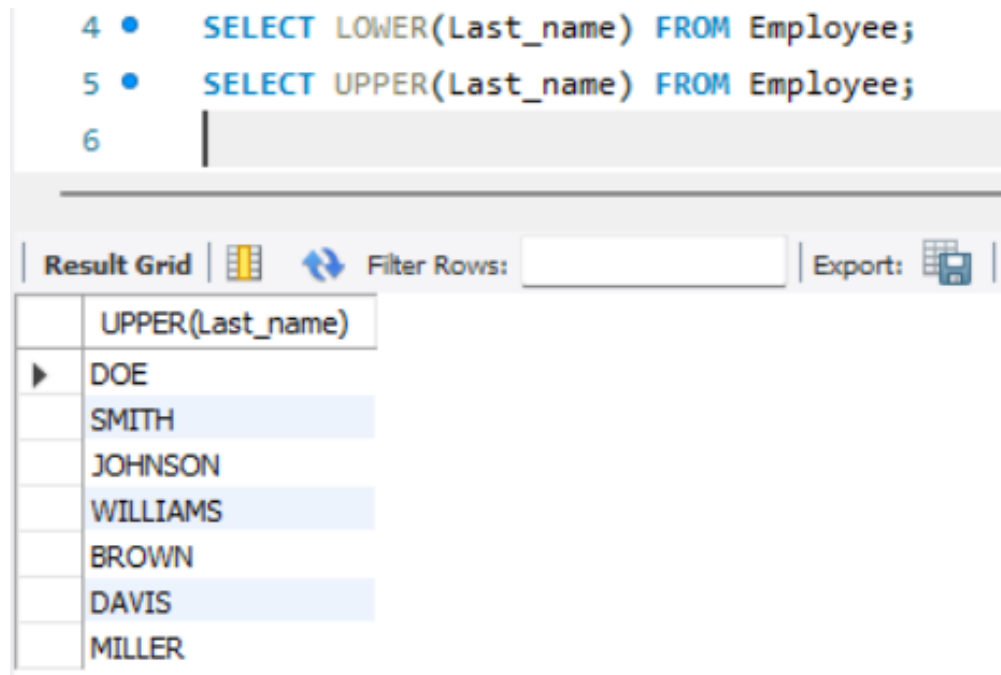
	LOWER(Last_name)
▶	doe
	smith
	johnson
	williams
	brown
	davis
	miller

Figure 4: Resultado de la consulta para apellidos en minúsculas

— *Obtener todos los valores de la columna 'Last\_name' en may sculas*

**SELECT UPPER**(Last\_name) **FROM** Employee;

Algebra Relacional: - {UPPER(Last\_name)} (Employee)



```
4 • SELECT LOWER(Last_name) FROM Employee;
5 • SELECT UPPER(Last_name) FROM Employee;
6
```

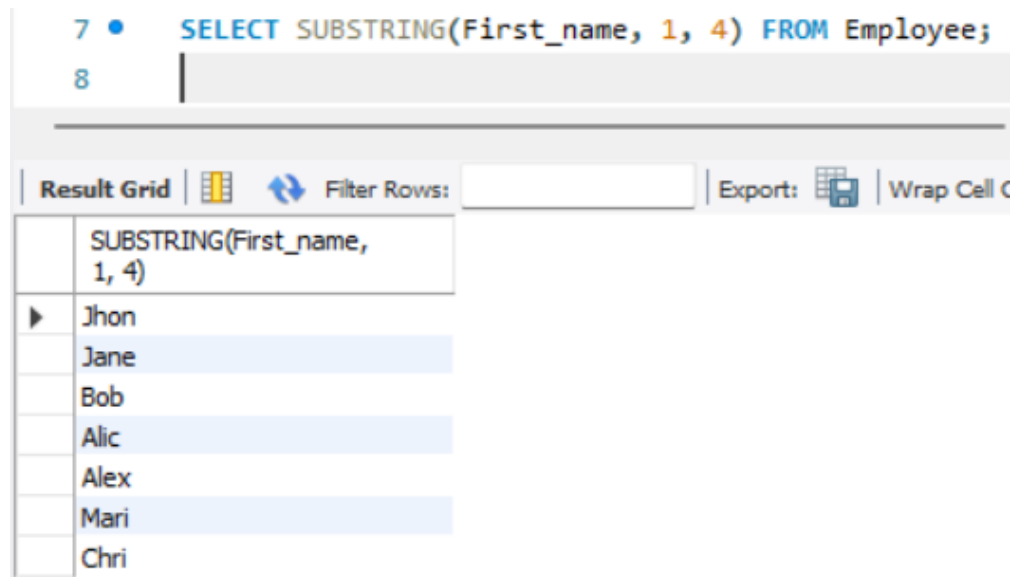
UPPER(Last_name)
DOE
SMITH
JOHNSON
WILLIAMS
BROWN
DAVIS
MILLER

Figure 5: Resultado de la consulta para apellidos en mayúsculas

— Obtener los primeros 4 caracteres de todos los valores de la columna "First\_name"

**SELECT SUBSTRING**(First\_name, 1, 4) **FROM** Employee;

Algebra Relacional: - {SUBSTRING(First\_name, 1, 4)} (Employee)



The screenshot shows a database query editor with the following SQL query:

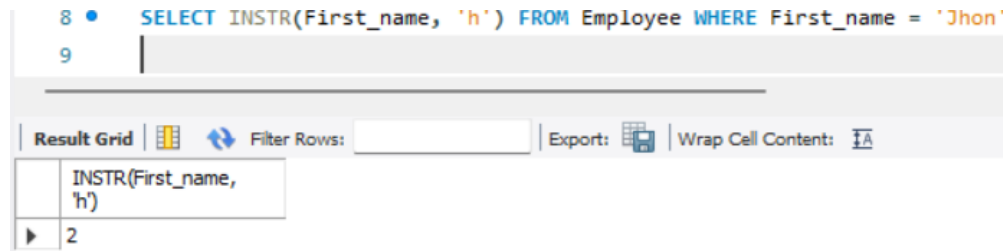
```
7 • SELECT SUBSTRING(First_name, 1, 4) FROM Employee;
8
```

Below the query editor is a toolbar with options: Result Grid, Filter Rows, Export, and Wrap Cell. The results are displayed in a grid with the following data:

	SUBSTRING(First_name, 1, 4)
▶	Jhon
	Jane
	Bob
	Alic
	Alex
	Mari
	Chri

Figure 6: Resultado de la consulta para los primeros 4 caracteres de los nombres

— *Obtener la posición de la letra "h" en el nombre del empleado con First\_name = "John"*  
**SELECT** INSTR(First\_name, 'h') **FROM** Employee **WHERE** First\_name = 'John';  
 Algebra Relacional:  $\pi_{\{ \text{POSITION('h' IN First\_name)} \}} ( \sigma_{\{ \text{First\_name} = \text{'Jhon'} \}} (\text{Employee}))$



The screenshot shows a SQL query editor with the following query:

```
8 • SELECT INSTR(First_name, 'h') FROM Employee WHERE First_name = 'Jhon'
```

Below the query editor, there is a toolbar with options: "Result Grid", "Filter Rows:", "Export:", and "Wrap Cell Content:". Below the toolbar, a result grid is displayed with the following data:

	INSTR(First_name, 'h')
▶	2

Figure 7: Resultado de la consulta para la posición de 'h' en "John"

### 4.2.3 Consultas Avanzadas

— *Obtener los nombres únicos de la columna "Department"*

**SELECT DISTINCT** Department **FROM** Employee;

Algebra Relacional:  $\pi_{\{\text{Department}\}}(\text{Employee})$

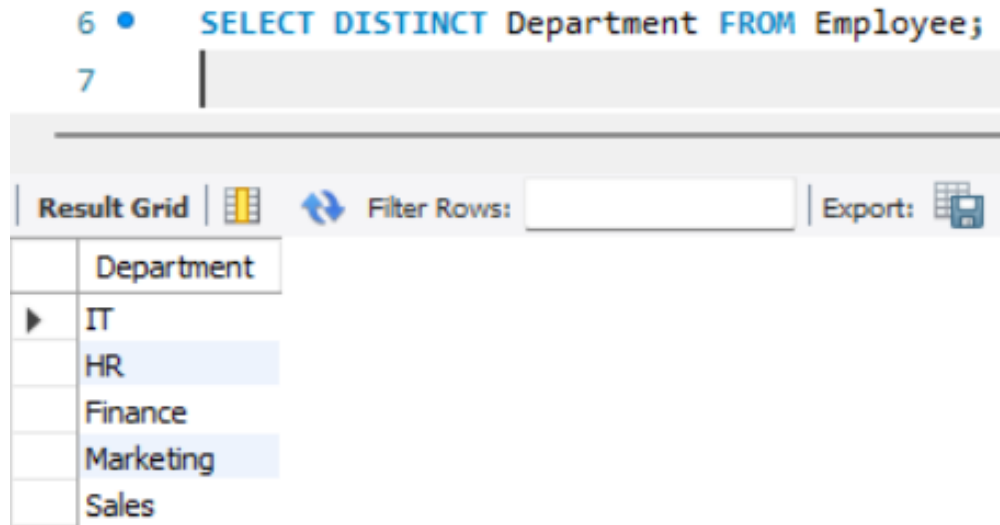





Figure 8: Resultado de la consulta para departamentos únicos

— Obtener todos los valores de la columna "First\_name" después de remover los espacios en blanco.  
**SELECT** RTRIM(First\_name) **FROM** Employee;  
Algebra Relacional:  $\pi_{\{RTRIM(First\_name)\}} (Employee)$

9 • **SELECT** RTRIM(First\_name) **FROM** Employee;  
10

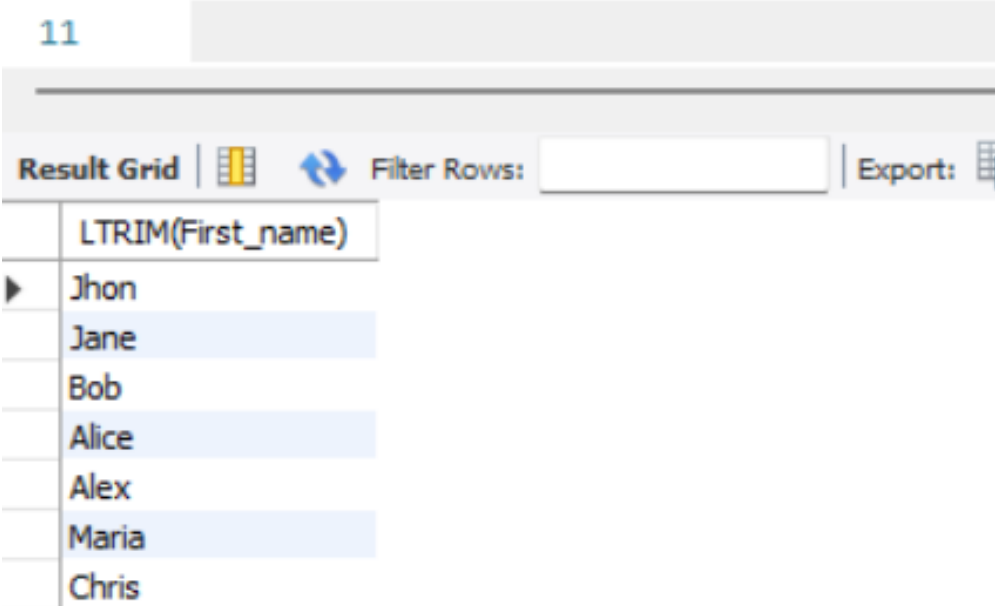
Result Grid |   Filter Rows:  | Export: 

	RTRIM(First_name)
▶	Jhon
	Jane
	Bob
	Alice
	Alex
	Maria
	Chris

Figure 9: Resultado de la consulta con RTRIM en nombres

— Obtener todos los valores de la columna "First\_name" después de remover los espacios en blanco  
**SELECT** LTRIM(First\_name) **FROM** Employee;  
Algebra Relacional:  $\pi_{\{LTRIM(First\_name)\}} (Employee)$

```
10 • SELECT LTRIM(First_name) FROM Employee;
11
```



LTRIM(First_name)
Jhon
Jane
Bob
Alice
Alex
Maria
Chris

Figure 10: Resultado de la consulta con LTRIM en nombres



## Conclusiones

A través de esta práctica, hemos profundizado en la comprensión y aplicación del álgebra relacional en el contexto de bases de datos MySQL. Hemos aprendido a:

- Aplicar operaciones del álgebra relacional para realizar consultas complejas.
- Optimizar el rendimiento de las bases de datos mediante el uso eficiente de consultas SQL.
- Configurar y manipular bases de datos MySQL, obteniendo una mejor comprensión de su arquitectura y funcionalidades.

Esta experiencia práctica ha reforzado la importancia del álgebra relacional en el diseño y manejo de bases de datos distribuidas, proporcionando una base sólida para futuros desarrollos en este campo.

## Referencias Bibliográficas

- Costal Costa, D. (2007). *El modelo relacional y el álgebra relacional*.
- Definición Wiki. (2023). *Definición de Álgebra Relacional en MySQL: según Autor, Ejemplos, qué es, Concepto, Significado*.
- Universitat Jaume I. (2018). *BASES DE DATOS (IG18 Semipresencial) El Modelo Relacional Álgebra*.
- Platzi. (2023). *Álgebra relacional y Bases de Datos*.