

# REPORTE DE PRÁCTICA

## ÁLGEBRA RELACIONAL Y BASES DE DATOS DISTRIBUIDAS

**ALUMNO:** Amuchategui Flores Braulio

Dr.Eduardo Cornejo-Velázquez

## Introducción

El álgebra relacional es un componente esencial en la teoría de bases de datos, proporcionando un marco matemático para la manipulación y consulta de datos. Este lenguaje formal es fundamental para el diseño, gestión y consulta de bases de datos relacionales, y se basa en operaciones que actúan sobre relaciones, que en términos prácticos corresponden a tablas en una base de datos. Los operadores del álgebra relacional, como selección, proyección, unión, diferencia y producto cartesiano, permiten transformar y combinar datos de manera precisa y estructurada.

En el contexto de bases de datos distribuidas, donde los datos están repartidos entre múltiples ubicaciones físicas o servidores, el álgebra relacional sigue desempeñando un papel crucial. Las bases de datos distribuidas presentan desafíos adicionales en comparación con las bases de datos centralizadas, como la necesidad de garantizar la consistencia y la integridad de los datos a través de diferentes nodos de red. El álgebra relacional ayuda a abordar estos desafíos al proporcionar una forma abstracta y matemática de describir las operaciones sobre datos distribuidos, facilitando así la integración y el procesamiento de información dispersa.

Por ejemplo, al aplicar una consulta sobre una base de datos distribuida, se pueden utilizar operaciones de álgebra relacional para combinar datos de diferentes nodos de manera eficiente. Operaciones como la unión y la intersección permiten consolidar datos provenientes de diversas fuentes, mientras que la selección y proyección ayudan a filtrar y refinar los datos relevantes. La capacidad de expresar consultas complejas de manera formal y precisa es especialmente valiosa en un entorno distribuido, donde la coordinación y la sincronización de datos pueden ser complicadas.

Además, el álgebra relacional en bases de datos distribuidas no solo se limita a la consulta y manipulación de datos, sino que también influye en el diseño de esquemas y en la optimización de consultas. Las técnicas de optimización se basan en transformar consultas algebraicas en planes de ejecución eficientes, lo cual es esencial para mejorar el rendimiento en sistemas distribuidos que manejan grandes volúmenes de datos y realizan operaciones complejas a través de múltiples nodos.

El álgebra relacional ofrece un marco poderoso para la consulta y manipulación de datos en bases de datos distribuidas. Su capacidad para formalizar operaciones sobre datos y su integración con técnicas de optimización aseguran que las consultas sean precisas y eficientes, facilitando la gestión de información distribuida y contribuyendo a la efectividad general de los sistemas de bases de datos modernos. Este enfoque formal y estructurado permite a los diseñadores de bases de datos y administradores manejar con éxito la complejidad de las bases de datos distribuidas, garantizando la integridad y el rendimiento en un entorno cada vez más dinámico y distribuido.

## 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');
```

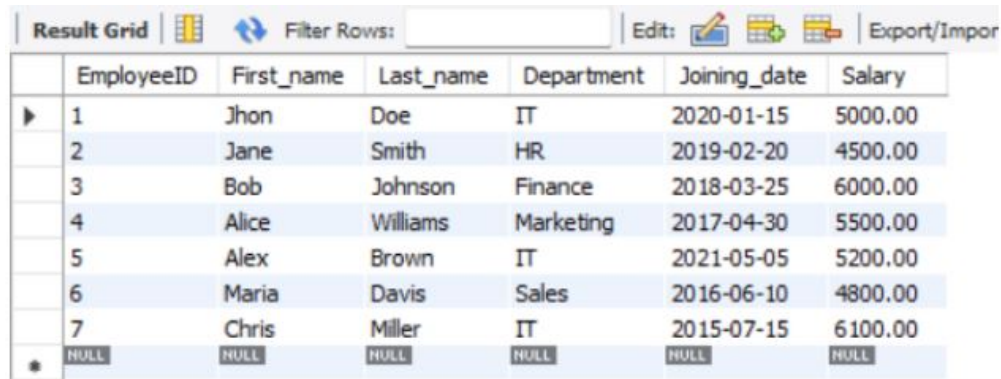
## Consultas SQL

### 4.2.1 Consultas Básicas

— *Obtener todos los empleados*

**SELECT \* FROM Employee;**

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



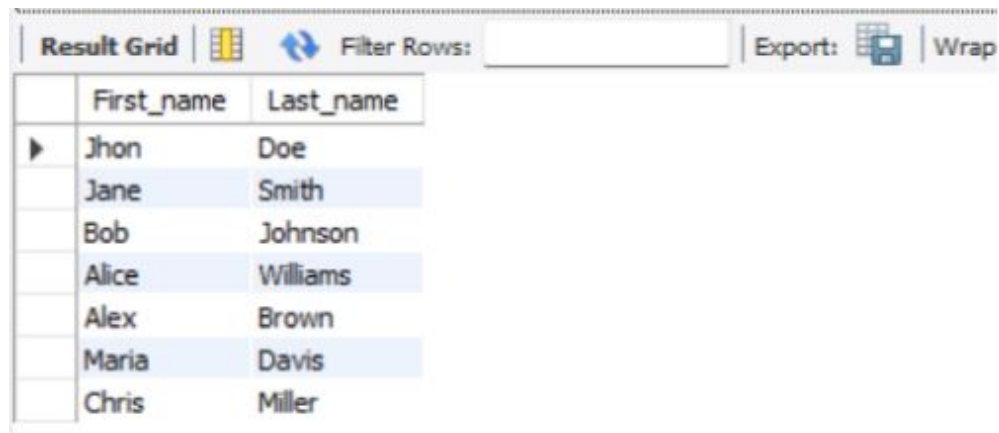
	EmployeeID	First_name	Last_name	Department	Joining_date	Salary
▶	1	Jhon	Doe	IT	2020-01-15	5000.00
	2	Jane	Smith	HR	2019-02-20	4500.00
	3	Bob	Johnson	Finance	2018-03-25	6000.00
	4	Alice	Williams	Marketing	2017-04-30	5500.00
	5	Alex	Brown	IT	2021-05-05	5200.00
	6	Maria	Davis	Sales	2016-06-10	4800.00
	7	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;

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



The image shows a screenshot of a database query result grid. At the top, there is a toolbar with the text 'Result Grid', a grid icon, a refresh icon, a 'Filter Rows:' text box, an 'Export:' button with a document icon, and a 'Wrap' button. Below the toolbar is a table with two columns: 'First\_name' and 'Last\_name'. The table contains eight rows of data. The first row is highlighted with a blue background. The first column of the table has a small triangle icon next to each row.

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

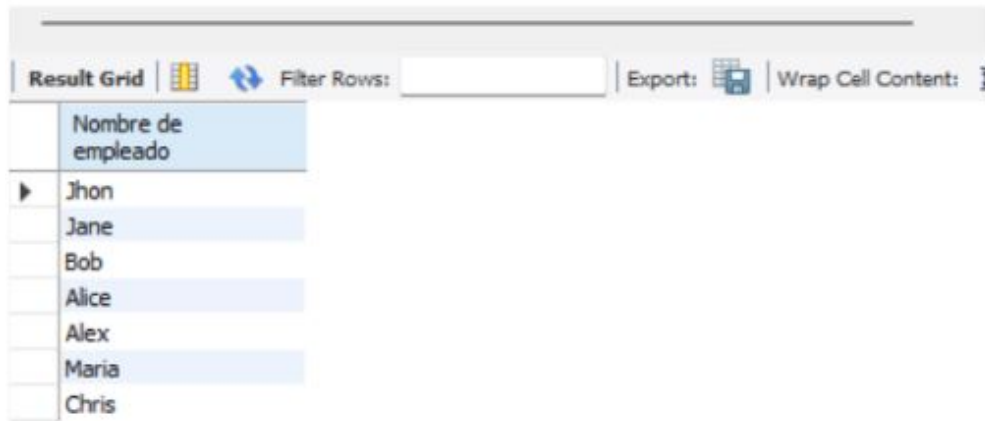
Figure 2: Resultado de la consulta para obtener nombres y apellidos

— Obtener todos los valores de la columna "First\_name" usando el alias "Nombre de empleado"

```
SELECT First_name AS "Nombre-de-empleado" FROM Employee;
```

lgebra Relacional: - {First\_name "Nombre-de-empleado"} (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

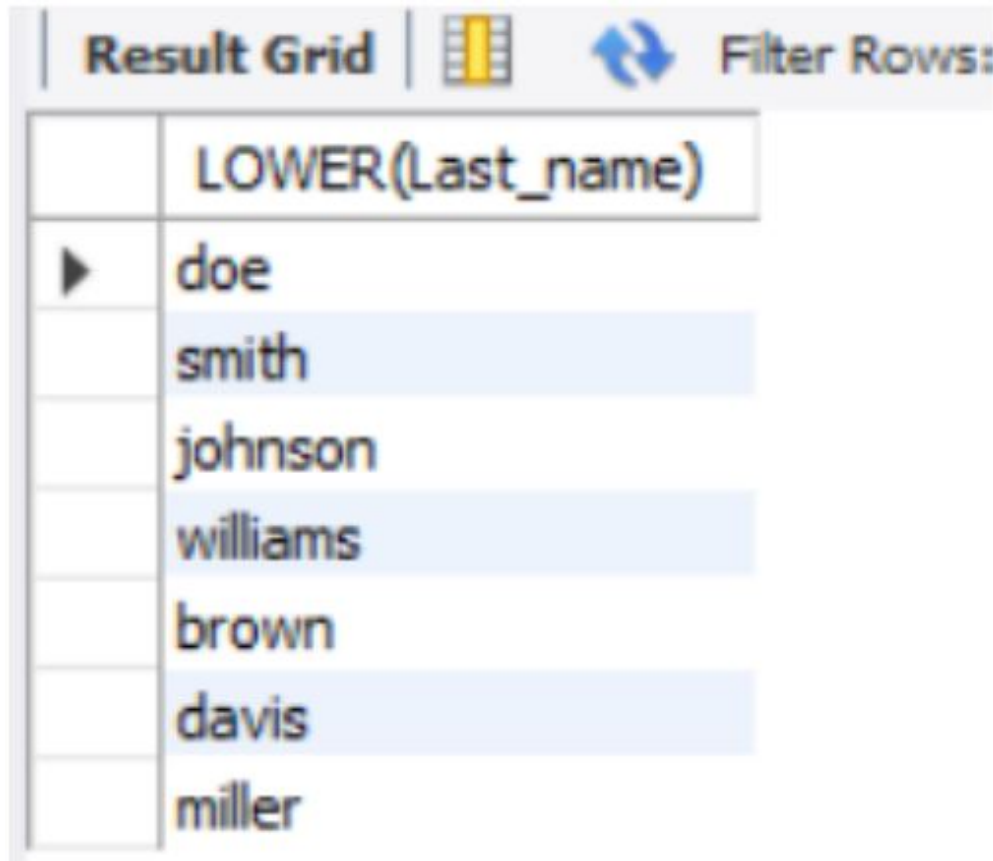
Figure 3: Resultado de la consulta con alias "Nombre de empleado"

#### 4.2.2 Manipulación de Cadenas

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

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

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



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

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;

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

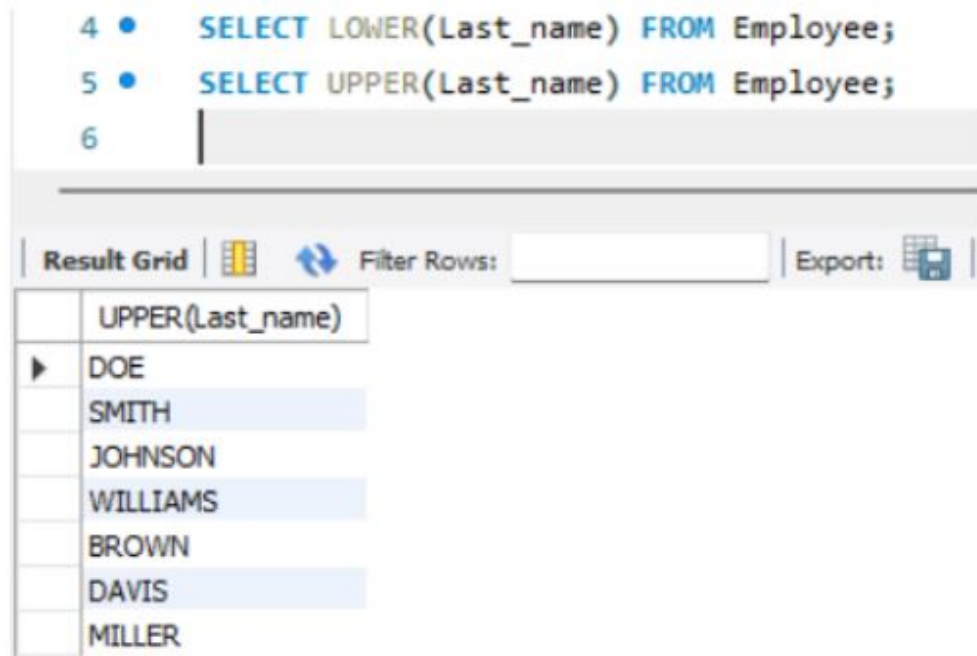
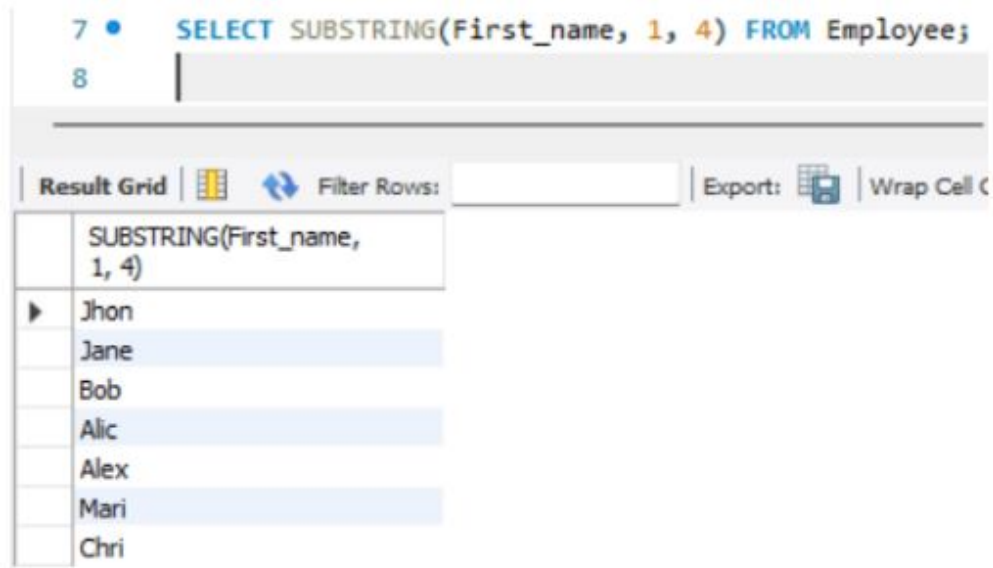


Figure 5: Resultado de la consulta para apellidos en mayúsculas

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

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

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



The screenshot shows a database query interface. At the top, the SQL query is entered: `SELECT SUBSTRING(First_name, 1, 4) FROM Employee;`. Below the query editor, there is a toolbar with options like 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell'. The 'Result Grid' is active, displaying 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';  
 lgebra Relacional: - {POSITION('h' IN First\_name)} ( - {First\_name = 'John'}(Employee))



Result Grid	
	INSTR(First_name, 'h')
▶	2

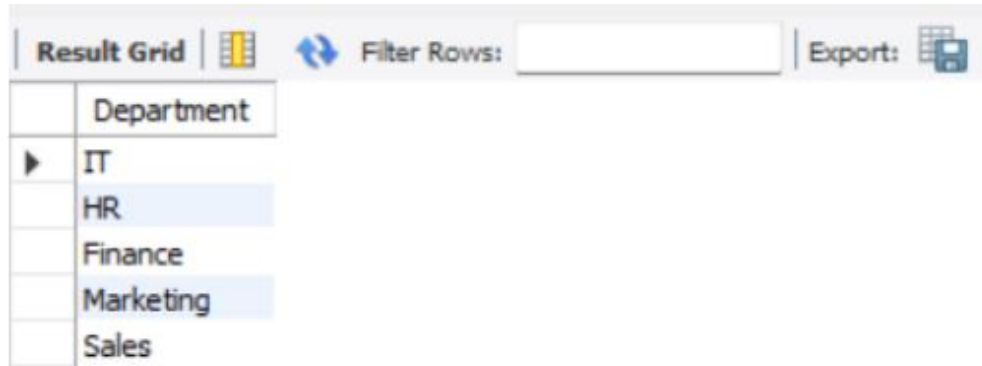
Figure 7: Resultado de la consulta para la posición de 'h' en "John"

### 4.2.3 Consultas Avanzadas

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

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

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



The screenshot shows a software interface for a database query result. At the top, there is a toolbar with a 'Result Grid' tab, a 'Filter Rows' input field, and an 'Export' button. Below the toolbar is a table with a single column labeled 'Department'. The table contains five rows with the values 'IT', 'HR', 'Finance', 'Marketing', and 'Sales'. The rows for 'HR', 'Marketing', and 'Sales' are highlighted with a light blue background.

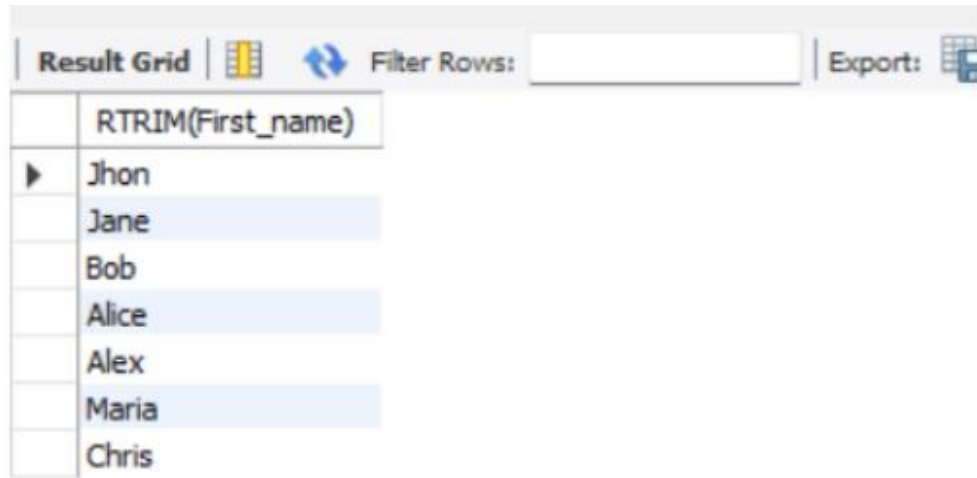
Department
IT
HR
Finance
Marketing
Sales

Figure 8: Resultado de la consulta para departamentos únicos

— Obtener todos los valores de la columna "First\_name" después de remover los espacios en

```
SELECT RTRIM(First_name) FROM Employee;
```

álgebra Relacional:  $\{ \text{RTRIM(First\_name)} \} \text{ (Employee)}$



The screenshot shows a database interface with a 'Result Grid' tab. The query executed is `SELECT RTRIM(First_name) FROM Employee;`. The result grid displays a single column titled `RTRIM(First_name)` with the following values: Jhon, Jane, Bob, Alice, Alex, Maria, and Chris. The rows for Jane, Alice, and Maria are highlighted in blue. The interface also includes a 'Filter Rows' search bar and an 'Export' button.

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;  
álgebra Relacional:  $\pi_{\{LTRIM(First\_name)\}} (Employee)$

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

Result Grid		Filter Rows:	Export:
	LTRIM(First_name)		
▶	Jhon		
	Jane		
	Bob		
	Alice		
	Alex		
	Maria		
	Chris		

Figure 10: Resultado de la consulta con LTRIM en nombres

## Conclusiones

El álgebra relacional es un marco teórico fundamental en la base de datos relacionales. Proporciona un conjunto de operaciones que permiten manipular y consultar datos almacenados en tablas (o relaciones). Estas operaciones incluyen la selección, proyección, unión, diferencia, producto cartesiano y las operaciones de combinación, como la intersección y la unión natural.

El álgebra relacional proporciona una base matemática sólida para la consulta de bases de datos, lo que garantiza que las operaciones sobre los datos sean precisas y consistentes. Además, es independiente del lenguaje SQL, lo que permite que diferentes sistemas de bases de datos puedan implementar estas operaciones de manera uniforme.

## 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*.