

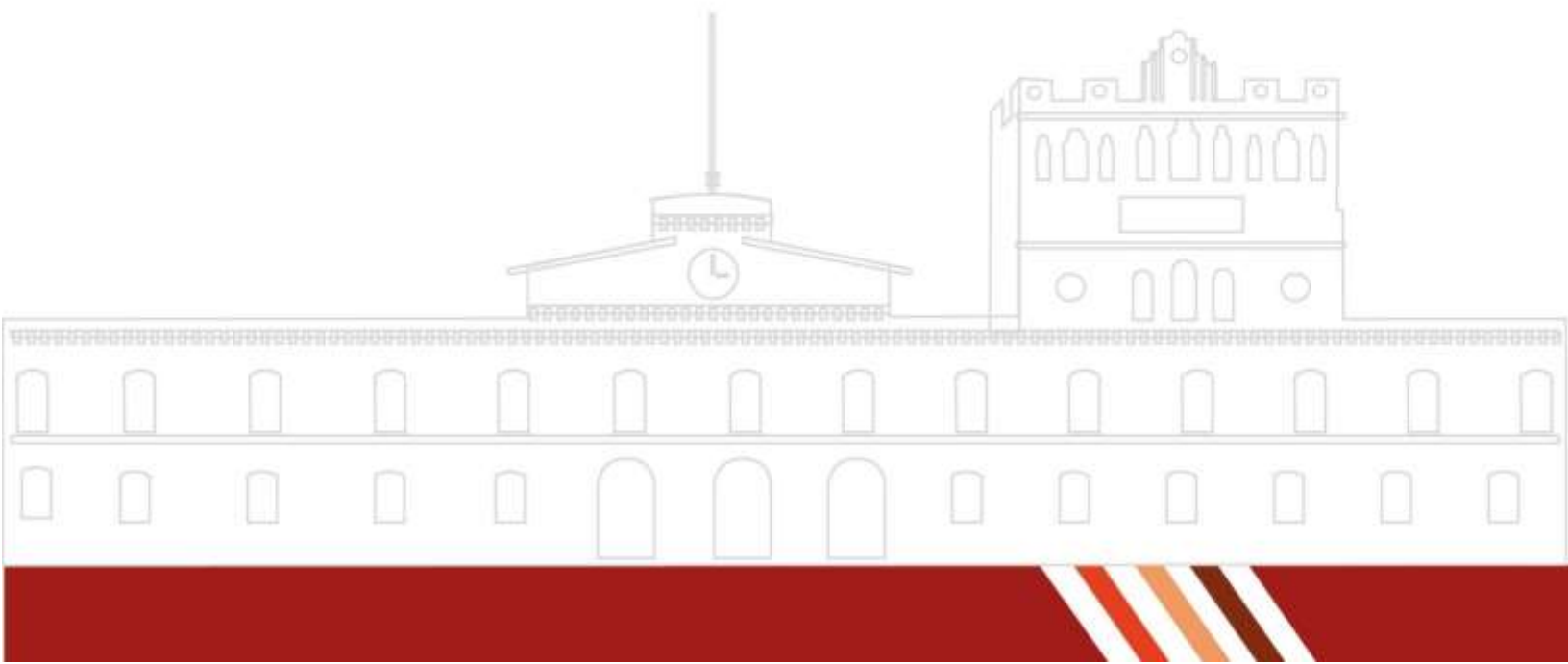
Reporte de Practica No. 1.3

Nombre de la Practica:

EJERCICIOS 1

**Alumno: Angel Amaya
Zumaya**

**Dr. Eduardo Cornejo
Velázquez**



Introducción

En el reporte se desarrollan ejercicios prácticos utilizando **álgebra relacional** y su implementación en **SQL con MySQL**, aplicados a las tablas *Employee* y *Reward*. El objetivo es reforzar los conceptos teóricos y la práctica de consultas básicas, creación e inserción de datos, y manipulación de cadenas.

2. Marco teórico

□ Álgebra Relacional

- Definición: modelo teórico de operaciones sobre relaciones (proyección, selección, unión, producto cartesiano, etc.).
- Ejemplo: π nombre, apellido (Empleado).

□ SQL (Structured Query Language)

- Lenguaje estándar para manejo de bases de datos relacionales.
- Sentencias DDL (CREATE, ALTER, DROP) y DML (INSERT, SELECT, UPDATE, DELETE).

□ MySQL

- Sistema de gestión de bases de datos relacional muy utilizado.
- Compatible con SQL estándar, soporta consultas, funciones de cadenas, agregados, etc.

□ Herramientas utilizadas

- MySQL Server.
- MySQL Workbench para ejecutar consultas y mostrar resultados.

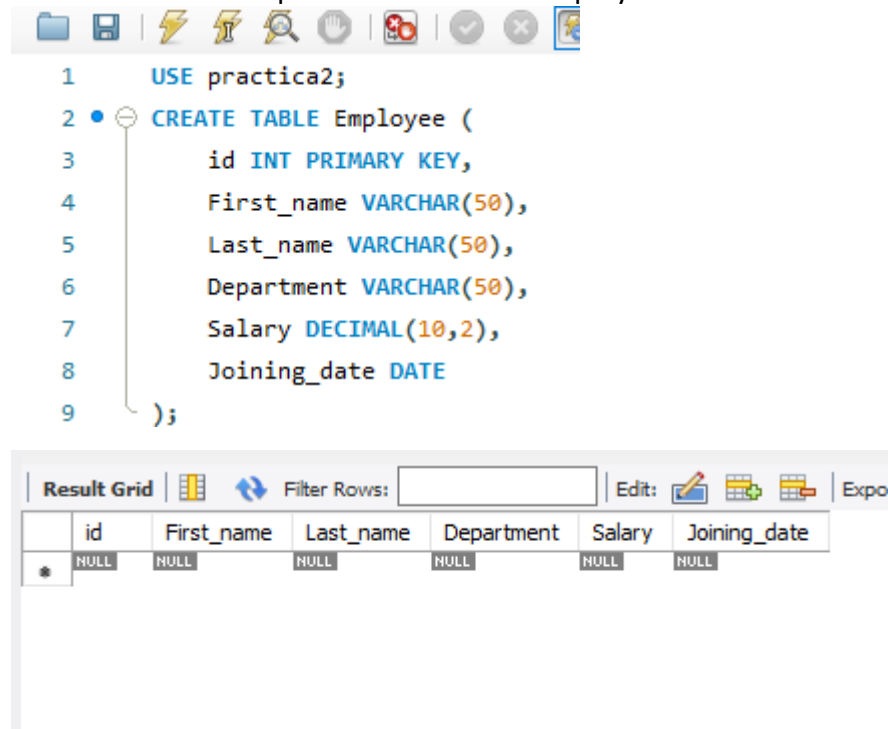
3. Metodología de diseño e implementación

Las herramientas que se utilizó en esta práctica fue el Sistema Gestor de Bases de Datos (SGBD) MySQL Workbench, en el cual ya hemos estado familiarizados y vamos a realizar el uso de álgebra relacional.

4. Desarrollo (MySQL)

EJERCICIOS.

1. Escribe la sintaxis para crear la tabla "Employee".



```
1  USE practica2;
2  CREATE TABLE Employee (
3      id INT PRIMARY KEY,
4      First_name VARCHAR(50),
5      Last_name VARCHAR(50),
6      Department VARCHAR(50),
7      Salary DECIMAL(10,2),
8      Joining_date DATE
9  );
```

Result Grid

	id	First_name	Last_name	Department	Salary	Joining_date
*	NULL	NULL	NULL	NULL	NULL	NULL

2. Escribe la sintaxis para insertar 7 registros (de la imagen) a la tabla "Employee".

```
INSERT INTO Employee (id, First_name, Last_name, Department, Salary, Joining_date) VALUES
(1, 'Jhon', 'Doe', 'HR', 50000, '2020-01-10'),
(2, 'Jane', 'Smith', 'IT', 60000, '2019-03-15'),
(3, 'Michael', 'Johnson', 'Finance', 70000, '2021-06-20'),
(4, 'Emily', 'Davis', 'IT', 55000, '2018-11-25'),
(5, 'William', 'Brown', 'HR', 52000, '2022-05-12'),
(6, 'Sophia', 'Wilson', 'Finance', 75000, '2017-08-30'),
(7, 'Daniel', 'Miller', 'IT', 58000, '2019-09-01');
```

Result Grid

	id	First_name	Last_name	Department	Salary	Joining_date
▶	1	Jhon	Doe	HR	5000...	2020-01-10
	2	Jane	Smith	IT	6000...	2019-03-15
	3	Michael	Johnson	Finance	7000...	2021-06-20
	4	Emily	Davis	IT	5500...	2018-11-25
	5	William	Brown	HR	5200...	2022-05-12
	6	Sophia	Wilson	Finance	7500...	2017-08-30
	7	Daniel	Miller	IT	5800...	2019-09-01

3. Escribe la sintaxis para crear la tabla "Reward".

```
CREATE TABLE Reward (  
    Employee_id INT,  
    Date_reward DATE,  
    Amount DECIMAL(10,2),  
    FOREIGN KEY (Employee_id) REFERENCES Employee(id)  
);
```

Employee_id	Date_reward	Amount
-------------	-------------	--------

4. Escribe la sintaxis para insertar 4 registros (en la imagen) a la tabla "Reward".

```
INSERT INTO Reward VALUES(1,'2019-05-11',1000), (3, '2022-01-20', 1500),  
(5, '2022-03-10', 1200),  
(7, '2021-07-05', 800);
```

Employee_id	Date_reward	Amount
1	2019-05-11	1000.00
3	2022-01-20	1500.00
5	2022-03-10	1200.00
7	2021-07-05	800.00

5. Obtener todos los empleados.

```
28 • SELECT * FROM Employee;
```

29

Result Grid					
Filter Rows:					
id	First_name	Last_name	Department	Salary	Joining_date
1	Jhon	Doe	HR	50000.00	2020-01-10
2	Jane	Smith	IT	60000.00	2019-03-15
3	Michael	Johnson	Finance	70000.00	2021-06-20
4	Emily	Davis	IT	55000.00	2018-11-25
5	William	Brown	HR	52000.00	2022-05-12
6	Sophia	Wilson	Finance	75000.00	2017-08-30
7	Daniel	Miller	IT	58000.00	2019-09-01

6. Obtener el primer nombre y apellido de todos los empleados.

```
28 • SELECT First_name, Last_name FROM Employee;  
29
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
First_name	Last_name			
Jhon	Doe			
Jane	Smith			
Michael	Johnson			
Emily	Davis			
William	Brown			
Sophia	Wilson			
Daniel	Miller			

7. Obtener todos los valores de la columna "First_name" usando el alias "Nombre de empleado".

```
29 • SELECT First_name AS "Nombre de empleado" FROM Employee;  
30
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Nombre de empleado				
Jhon				
Jane				
Michael				
Emily				
William				
Sophia				

8. Obtener todos los valores de la columna "Last_name" en minúsculas.

```
29 • SELECT LOWER(Last_name) FROM Employee;  
30
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
LOWER(Last_name)				
doe				
smith				
johnson				
davis				
brown				
wilson				
miller				

9. Obtener todos los valores de la columna "Last_name" en mayúsculas.

```
30 • SELECT upper(Last_name) FROM Employee;
31
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content
	upper(Last_name)			
▶	DOE			
	SMITH			
	JOHNSON			
	DAVIS			
	BROWN			
	WILSON			
	MILLER			

10. Obtener los nombre únicos de la columna "Departament".

```
31 • SELECT DISTINCT Department FROM Employee;
32
```

Result Grid		Filter Rows:	Export:	Wi
	Department			
▶	HR			
	IT			
	Finance			

11. Obtener los primeros 4 caracteres de todos los valores de la columna "First_name".

```
32 • SELECT SUBSTRING(First_name, 1, 4) FROM Employee;
33
34
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	SUBSTRING(First_name, 1, 4)			
▶	Jhon			
	Jane			
	Mich			
	Emil			
	Will			
	Soph			
	Dani			

12. Obtener la posición de la letra “h” en el nombre del empleado con First_name = “Jhon”.

```
33 • SELECT POSITION('h' IN First_name)
34 FROM Employee
35 WHERE First_name = 'Jhon';
36
37
```

Result Grid		Filter Rows:	Export:
POSITION('h' IN First_name)			
▶	2		

13. Obtener todos los valores de la columna “First_name” después de remover los espacios en blanco de la derecha.

```
36 • SELECT RTRIM(First_name) FROM Employee;
37
38
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
RTRIM(First_name)				
▶	Jhon			
	Jane			
	Michael			
	Emily			
	William			
	Sophia			
	Daniel			

14. Obtener todos los valores de la columna “First_name” después de remover los espacios en blanco de la izquierda.

```
37 • SELECT LTRIM(First_name) FROM Employee;
38
39
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
LTRIM(First_name)				
▶	Jhon			
	Jane			
	Michael			
	Emily			
	William			
	Sophia			
	Daniel			

5. Conclusiones

Los ejercicios permitieron aplicar de manera práctica el álgebra relacional utilizando SQL en MySQL. Se comprobó cómo las operaciones de proyección y selección en álgebra relacional se traducen directamente en consultas SQL mediante SELECT y WHERE. Además, se reforzó el uso de funciones de cadenas, alias y operadores.

Bibliografía

- El manual de MySQL: <https://dev.mysql.com/doc/>
- El libro: Elmasri, R. & Navathe, S. (2015). *Fundamentals of Database Systems*. Pearson.
- Silberschatz, A., Korth, H., & Sudarshan, S. (2006). *Database System Concepts*. McGraw-Hill.