

# REPORTE DE PRÁCTICA NO. 2

**NOMBRE DE LA PRÁCTICA —1.3. Práctica. Álgebra relacional y SQL (1)**

**Dr. Eduardo Cornejo-Velázquez**

**ALUMNO: Cesar Martinez Andrade**



# 1. Introducción

En esta práctica, exploraremos el fascinante mundo de las funciones en el contexto de MySQL. Las funciones son herramientas poderosas que nos permiten realizar operaciones específicas dentro de nuestras consultas SQL. Al utilizarlas correctamente, podemos simplificar nuestro código, mejorar la eficiencia y obtener resultados más precisos.

Para llevar a cabo esta práctica, utilizaremos el software MySQL Command Client, una herramienta que nos permite interactuar con bases de datos MySQL mediante líneas de código. Aquí, escribiremos nuestras consultas y las compilaremos para obtener los resultados deseados.

Además, aprovecharemos los conceptos del álgebra relacional que hemos estudiado previamente en clase. El álgebra relacional es una teoría matemática que nos proporciona un conjunto de operaciones para manipular relaciones (tablas) en bases de datos. Al combinar estas operaciones con las funciones de MySQL, podremos resolver problemas complejos de manera elegante y estructurada.

## Objetivos

1. Comprender el propósito y la sintaxis de las funciones en MySQL.
2. Familiarizarnos con el uso del software MySQL Command Client.
3. Aplicar el álgebra relacional para resolver problemas específicos.

## 2. Marco teórico

### Álgebra Relacional

El álgebra relacional de manera breve es un conjunto de operaciones matemáticas que se aplican a relaciones (tablas) en una base de datos. Fue propuesto por Edgar F. Codd[4], el padre de las bases de datos relacionales. Algunas de las operaciones más importantes del álgebra relacional son:

- Selección[3]: Permite filtrar filas de una tabla según una condición específica. Por ejemplo, podemos seleccionar todos los estudiantes que tienen una calificación superior a 8.
- Proyección: Nos permite elegir columnas específicas de una tabla. Por ejemplo, podemos proyectar solo los nombres y apellidos de los profesores.
- Unión: Combina dos tablas para obtener un conjunto de filas sin duplicados. Por ejemplo, podemos unir las tablas de estudiantes y profesores para obtener una lista completa de personas relacionadas con la institución.

### Funciones de SQL

Las funciones en SQL son operaciones predefinidas que se utilizan para procesar o transformar los datos de una base de datos. Estas funciones se combinan con la sentencia `SELECT` y otras sentencias para realizar diversas tareas. Aquí tienes una breve descripción de algunas funciones comunes:

1. Funciones de Agregado: Estas funciones realizan cálculos sobre grupos de filas y devuelven un solo valor. Ejemplos incluyen `SUM`, `AVG`, `COUNT`, `MIN` y `MAX`.
2. Funciones de Cadena: Manipulan cadenas de texto. Algunas funciones útiles son `CONCAT` (para concatenar cadenas), `SUBSTRING` (para extraer una parte de una cadena) y `UPPER/LOWER` (para convertir a mayúsculas o minúsculas).
3. Funciones de Fecha y Hora: Trabajan con fechas y horas. Ejemplos son `GETDATE()` (para obtener la fecha y hora actual), `DATEADD` (para agregar o restar días/meses/años) y `DATEDIFF` (para calcular la diferencia entre dos fechas).

4. Funciones Matemáticas: Realizan operaciones matemáticas. Por ejemplo, **ROUND** (redondeo), **ABS** (valor absoluto) y **POWER** (potencia).
5. Funciones de Conversión de Datos: Cambian el tipo de datos. Por ejemplo, **CAST** o **CONVERT** para convertir una cadena en un número.

### 3. Herramientas empleadas

Describir qué herramientas se han utilizado...

1. **MySQL Command Client:** Esta herramienta nos permitirá interactuar con la base de datos MySQL mediante comandos SQL.
2. **Álgebra Relacional:** Aplicaremos los conceptos aprendidos en clase para modelar y resolver problemas relacionados con bases de datos.
3. **Bloc de notas o Notepad++:** Para agrupar y generar más rápidamente las tuplas necesarias usando las tabulaciones y comandos de copiado y pegado

### 4. Desarrollo

#### Bloque de Actividades solicitado

1. Escribe la sintaxis para crear la tabla "Employee".
2. Escribe la sintaxis para insertar 7 registros (de la imagen) a la tabla "Employee".
3. Escribe la sintaxis para crear la tabla "Reward".
4. Escribe la sintaxis para insertar 4 registros (en la imagen) a la tabla "Reward".
5. Obtener todos los empleados.
6. Obtener el primer nombre y apellido de todos los empleados.
7. Obtener todos los valores de la columna "First\_name" usando el alias "Nombre de empleado".
8. Obtener todos los valores de la columna "Last\_name" en minúsculas.
9. Obtener todos los valores de la columna "Last\_name" en mayúsculas.
10. Obtener los nombre únicos de la columna "Department".
11. Obtener los primeros 4 caracteres de todos los valores de la columna "First\_name".
12. Obtener la posición de la letra "h" en el nombre del empleado con First\_name = "Jhon".
13. Obtener todos los valores de la columna "First\_name" después de remover los espacios en blanco de la derecha.
14. Obtener todos los valores de la columna "First\_name" después de remover los espacios en blanco de la izquierda.

## Sentencias SQL

En el Listado 1 se presentaran todas las lineas de codigo para resolver el bloque de actividade solicitado anteriormente.

Listing 1: Crear base de datos competencia.

```
CREATE DATABASE PracticaAlgebraRelacionalySQL1;  
USE PracticaAlgebraRelacionalySQL1;  
  
CREATE TABLE Employee(Employee_id INT PRIMARY KEY,  
First_name VARCHAR(20),  
Last_name VARCHAR(20),  
Salary INT(15),  
Joining_date DATE,  
Departament VARCHAR(20));  
  
CREATE TABLE Reward(Employee_ref_id INT PRIMARY KEY,  
date_reward date,  
amount int(5));  
  
INSERT INTO Employee VALUES  
(1,"Bob","Kinto",1000000,"2019-01-20","Finance"),  
(2,"Jerry","Kansxo",6000000,"2019-01-15","IT"),  
(3,"Philip","Jose",8900000,"2019-02-05","Banking"),  
(4,"John","Abraham",2000000,"2019-02-25","Insurance"),  
(5,"Michael","Mathew",2200000,"2019-02-18","Finance"),  
(6,"Alex","chreketo",4000000,"2019-05-10","IT"),  
(7,"Yohan","Soso",1230000,"2019-06-20","Banking");  
  
INSERT INTO Reward VALUES  
(1,"2019-05-11",1000),  
(2,"2019-02-15",5000),  
(3,"2019-04-22",2000),  
(4,"2019-06-20",8000);  
  
SELECT * FROM Employee;  
SELECT first_name,last_name FROM Employee;  
SELECT first_name AS "Nombre-de-empleado" FROM Employee;  
SELECT lower(last_name) AS "Minusculas" FROM Employee;  
SELECT upper(last_name) AS "Mayusculas" FROM Employee;  
SELECT DISTINCT Departament AS "Departamentos-Unicos" FROM Employee;  
SELECT left(first_name,4) AS "Primeros-4-Caracteres" FROM Employee;  
SELECT LOCATE('h', First_name) AS "Posicion-de-la-letra-H"  
FROM Employee WHERE First_name = 'John';  
SELECT RTRIM(First_name) AS "Remover-Espacios-en-Blanco-de-la-Derecha"  
FROM Employee;  
SELECT LTRIM(First_name) AS "Remover-Espacios-en-Blanco-de-la-Izquierda"  
FROM Employee;
```

## Capturas de Pantalla



```
MySQL 9.0 Command Line Cli  X  +  v  -  □  X

Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 9.0.1 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE PracticaAlgebraRelacionalySQL1;
Query OK, 1 row affected (0.01 sec)

mysql> USE PracticaAlgebraRelacionalySQL1;
Database changed
mysql> CREATE TABLE Employee(Employee_id INT PRIMARY KEY,
-> First_name VARCHAR(20),
-> Last_name VARCHAR(20),
-> Salary INT(15),
-> Joining_date DATE,
-> Departament VARCHAR(20));
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> CREATE TABLE Reward(Employee_ref_id INT PRIMARY KEY, date_reward date, amo
unt int(5));
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> INSERT INTO Employee VALUES
-> (1,"Bob","Kinto",1000000,"2019-01-20","Finance"),
-> (2,"Jerry","Kansxo",6000000,"2019-01-15","IT"),
-> (3,"Philip","Jose",8900000,"2019-02-05","Banking"),
-> (4,"John","Abraham",2000000,"2019-02-25","Insurance"),
-> (5,"Michael","Mathew",2200000,"2019-02-18","Finance"),
-> (6,"Alex","chreketo",4000000,"2019-05-10","IT"),
-> (7,"Yohan","Soso",1230000,"2019-06-20","Banking");
Query OK, 7 rows affected (0.00 sec)
Records: 7 Duplicates: 0 Warnings: 0

mysql>
mysql> INSERT INTO Reward VALUES
-> (1,"2019-05-11",1000),
-> (2,"2019-02-15",5000),
-> (3,"2019-04-22",2000),
-> (4,"2019-06-20",8000);
Query OK, 4 rows affected (0.00 sec)
Records: 4 Duplicates: 0 Warnings: 0

mysql> |
```

Figure 1: Creacion y poblado de tablas

```
MySQL 9.0 Command Line Cli x + v - □ x

mysql> SELECT upper(last_name) AS "Mayusculas" FROM Employee;
+-----+
| Mayusculas |
+-----+
| KINTO      |
| KANSXO     |
| JOSE       |
| ABRAHAM    |
| MATHEW     |
| CHREKETO   |
| SOSO       |
+-----+
7 rows in set (0.00 sec)

mysql> SELECT DISTINCT Department AS "Departamentos Unicos" FROM Employee;
+-----+
| Departamentos Unicos |
+-----+
| Finance              |
| IT                   |
| Banking              |
| Insurance            |
+-----+
4 rows in set (0.00 sec)

mysql> SELECT left(first_name,4) AS "Primeros 4 Caracteres" FROM Employee;
+-----+
| Primeros 4 Caracteres |
+-----+
| Bob                   |
| Jerr                  |
| Phil                  |
| John                  |
| Mich                  |
| Alex                  |
| Yoha                  |
+-----+
7 rows in set (0.00 sec)

mysql> SELECT LOCATE('h', First_name) AS "Posicion de la letra H" FROM Employee W
HERE First_name = 'John';
+-----+
| Posicion de la letra H |
+-----+
| 3                       |
+-----+
1 row in set (0.00 sec)

mysql> SELECT RTRIM(First_name) AS "Remover Espacios en Blanco de la Derecha" FRO
M Employee;
+-----+
| Remover Espacios en Blanco de la Derecha |
+-----+
```

Figure 2: Consultas solicitadas.

## Algebra Relacional

### 4.4.1 Sintaxis para insertar registros

$$Employee \leftarrow Employee \cup \{1, "Bob", "Kinto", 1000000, "2019 - 01 - 20", "Finance"\} \quad (1)$$

$$Employee \leftarrow Employee \cup \{2, "Jerry", "Kansxo", 6000000, "2019 - 01 - 15", "IT"\} \quad (2)$$

$$Employee \leftarrow Employee \cup \{3, "Philip", "Jose", 8900000, "2019 - 02 - 05", "Banking"\} \quad (3)$$

$$Employee \leftarrow Employee \cup \{4, "John", "Abraham", 2000000, "2019 - 02 - 25", "Insurance"\} \quad (4)$$

$$Employee \leftarrow Employee \cup \{5, "Michael", "Mathew", 2200000, "2019 - 02 - 18", "Finance"\} \quad (5)$$

$$Employee \leftarrow Employee \cup \{6, "Alex", "chreketo", 4000000, "2019 - 05 - 10", "IT"\} \quad (6)$$

$$Employee \leftarrow Employee \cup \{7, "Yohan", "Soso", 1230000, "2019 - 06 - 20", "Banking"\} \quad (7)$$

$$Reward \leftarrow Reward \cup \{1, "2019 - 05 - 11", 1000\} \quad (8)$$

$$Reward \leftarrow Reward \cup \{2, "2019 - 02 - 15", 5000\} \quad (9)$$

$$Reward \leftarrow Reward \cup \{3, "2019 - 04 - 22", 2000\} \quad (10)$$

$$Reward \leftarrow Reward \cup \{4, "2019 - 06 - 20", 8000\} \quad (11)$$

### 4.4.2 Consultas

Obtener todos los empleados.

$$\sigma * (Employee) \quad (12)$$

Obtener el primer nombre y apellido de todos los empleados.

$$\pi Firstname, Lastname (Employee) \quad (13)$$

Obtener todos los valores de la columna "Firstname" usando el alias "Nombre de empleado"

$$\rho Nombredeempleado / \pi Firstname (Employee) \quad (14)$$

Obtener todos los valores de la columna "Lastname" en minúsculas.

$$lower(\pi Lastname (Employee)) \quad (15)$$

Obtener los nombre únicos de la columna "Departament"

$$Distinct(\pi Departement (Employee)) \quad (16)$$

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

$$\pi left(Firstname, 4) (Employee) \quad (17)$$

Obtener la posición de la letra "h" en el nombre del empleado con Firstname = "Jhon".

$$Locate("h", \pi Firstname = "Jhon" (Employee)) \quad (18)$$

Obtener todos los valores de la columna "Firstname" después de remover los espacios en blanco de la derecha.

$$\pi rtrimfirstname (Employee) \quad (19)$$

Obtener todos los valores de la columna "Firstname" después de remover los espacios en blanco de la izquierda.

$$\pi ltrimfirstname (Employee) \quad (20)$$

## 5. Conclusiones

En el fascinante mundo de las bases de datos, el álgebra relacional se alza como una herramienta indispensable. Durante esta práctica, he internalizado su relevancia al momento de insertar datos y he explorado algunas de las funciones que ofrece el lenguaje SQL. Entre estas funciones, encontramos joyas como LEFT o LOCATE, que son solo una pequeña muestra de las numerosas palabras reservadas que me permitirán afinar y optimizar mis consultas en el futuro.

## 6. Referencias

### References

- [1] Algebra relacional. (n.d.). <https://basesdatosdistribuidas.blogspot.com/2012/11/algebra-relacional.html>
- [2] Platzi. (n.d.). *Álgebra relacional y Bases de Datos*. <https://platzi.com/tutoriales/1566-bd/5823-algebra-relacional-y-bases-de-datos/>
- [3] VanMSFT. (n.d.). *Ejemplos de SELECT (Transact-SQL) - SQL Server*. Microsoft Learn. <https://learn.microsoft.com/es-es/sql/t-sql/queries/select-examples-transact-sql?view=sql-server-ver16>
- [4] colaboradores de Wikipedia. (2024, July 14). *Edgar Frank Codd*. Wikipedia, La Enciclopedia Libre. [https://es.wikipedia.org/wiki/Edgar\\_Frank\\_Codd](https://es.wikipedia.org/wiki/Edgar_Frank_Codd)