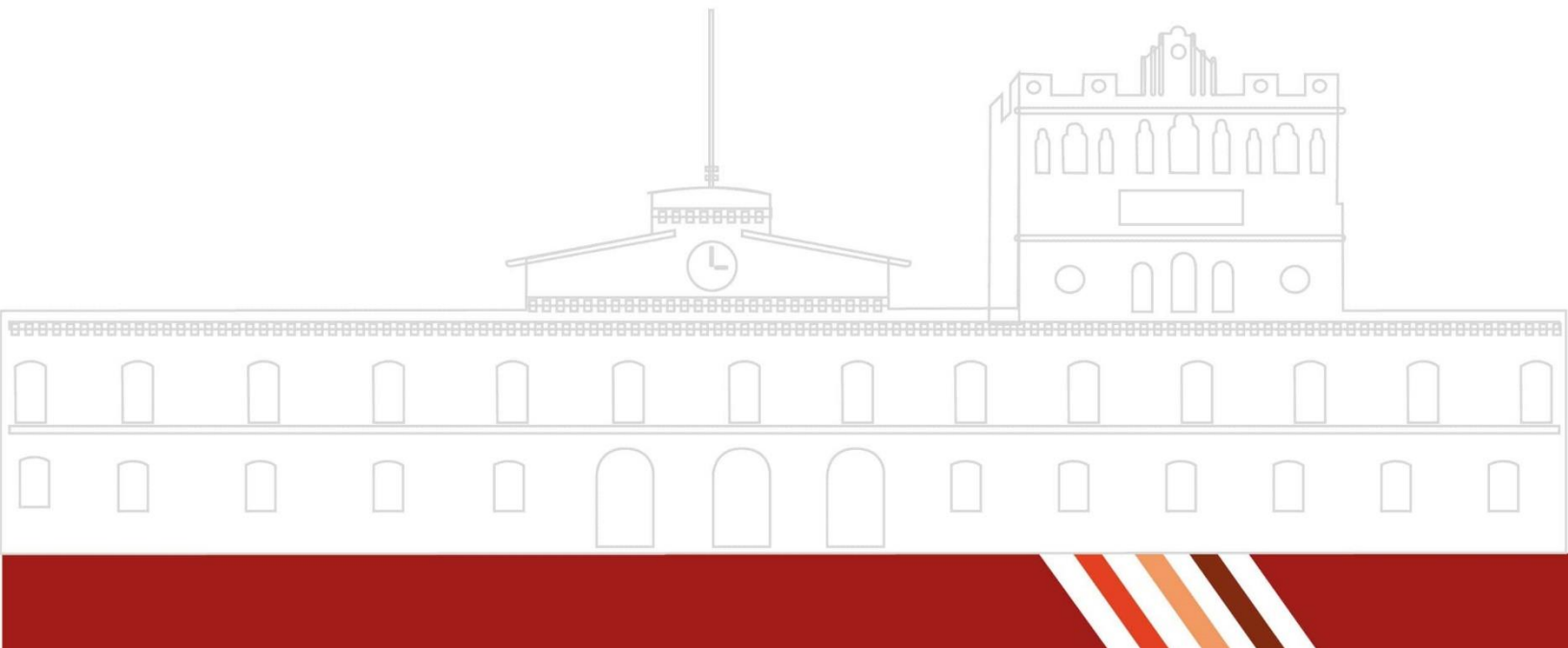


# REPORTE DE PRÁCTICA NO. 1.3

1.3. Práctica. Álgebra relacional y SQL (1)

ALUMNO: Sebastian Trejo Muñoz

Dr. Eduardo Cornejo-Velázquez



# 1. Introducción

En esta práctica, exploraremos el manejo y manipulación de bases de datos a través de ejercicios prácticos centrados en las tablas **Employee** y **Reward**. El objetivo es aplicar conocimientos de **SQL** para resolver una serie de ejercicios que ponen a prueba la capacidad de consultar, filtrar y modificar datos en estas tablas.

Cada ejercicio requerirá la redacción de una sentencia **SQL** que resuelva el problema planteado. Para demostrar la comprensión y correcta aplicación de los conceptos, se deberá incluir una captura de pantalla que muestre tanto la sentencia **SQL** utilizada como el resultado de su ejecución. Esta documentación permitirá verificar que la consulta fue ejecutada con éxito y que el resultado cumple con los requerimientos del ejercicio.

## 2. Marco teórico

### Tablas en Bases de Datos Relacionales:

En un sistema de bases de datos relacional, la información se organiza en tablas, que consisten en filas y columnas. Cada tabla representa una entidad específica, y cada fila en la tabla representa un registro único de esa entidad. Las columnas, por su parte, definen los atributos o características de la entidad. Por ejemplo, la tabla `Employee` podría contener columnas como `EmployeeID`, `Name`, y `Position`, mientras que la tabla `Reward` podría incluir `RewardID`, `EmployeeID`, y `RewardDate`.

### Consultas SQL:

SQL permite a los usuarios interactuar con la base de datos mediante comandos que pueden consultar (`SELECT`), insertar (`INSERT`), actualizar (`UPDATE`), y eliminar (`DELETE`) datos. Las consultas SQL son el pilar de la interacción con los datos, permitiendo filtrar, ordenar, agrupar, y sumarizar la información almacenada. En el contexto de las tablas `Employee` y `Reward`, las consultas SQL se utilizan para extraer información específica, como la relación entre empleados y las recompensas recibidas.

### Relaciones entre Tablas:

En un esquema relacional, las tablas pueden estar vinculadas entre sí mediante claves foráneas (Foreign Keys), lo que permite establecer relaciones entre diferentes conjuntos de datos. Por ejemplo, la relación entre la tabla `Employee` y `Reward` puede estar definida a través de la columna `EmployeeID`, donde cada recompensa está asociada con un empleado específico. Estas relaciones son fundamentales para realizar consultas que combinen información de múltiples tablas.

### Filtrado y Manipulación de Datos:

A través de las consultas SQL, es posible aplicar filtros y condiciones que permiten extraer conjuntos de datos específicos basados en ciertos criterios. Operadores como `WHERE`, `AND`, `OR`, y `LIKE` son comunes en el filtrado de datos, permitiendo la selección de registros que cumplan con ciertas condiciones.

### 3. Herramientas utilizadas

#### 1. MySQL Server

MySQL Server es un sistema de gestión de bases de datos relacional de código abierto, usado para almacenar y manejar datos. Permite realizar consultas **SQL**, gestionar la seguridad y control de acceso, y es conocido por su velocidad y escalabilidad. Es popular en el desarrollo web para manejar bases de datos en aplicaciones como sitios de comercio electrónico y sistemas de gestión de contenido.

#### 2. Overleaf

Overleaf es una plataforma en línea para crear y colaborar en documentos **LaTeX**. Permite la edición colaborativa en tiempo real y la compilación automática del documento. Ofrece plantillas y se integra con herramientas de referencias bibliográficas y control de versiones. Es ampliamente usado en entornos académicos y científicos para escribir artículos, tesis y presentaciones de alta calidad.

## 4. Desarrollo

### Sentencias SQL

1. Escribe la sintaxis para crear la tabla “Employee”.

```
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));
```

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

```
INSERT INTO Employee (Employee_id, First_name, Last_name, Salary, Joining_date,  
Departament)
```

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');
```

3. Escribe la sintaxis para crear la tabla “Reward”.

```
CREATE TABLE Reward(Employee_ref_id INT PRIMARY KEY, date_reward date, amount  
int(5));
```

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

```
INSERT INTO Reward (Employee_ref_id, date_reward, amount)
```

VALUES

```
(1, '2019-05-11', 1000),  
(2, '2019-02-15', 5000),  
(3, '2019-04-22', 2000),  
(4, '2019-06-20', 8000);
```

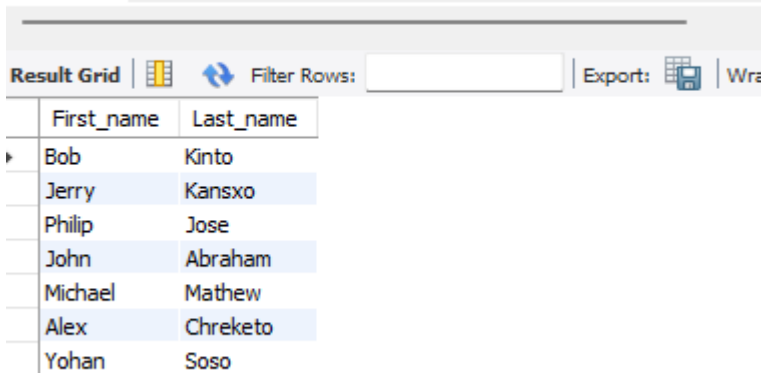
5. Obtener todos los empleados.

```
SELECT * FROM Employee;
```

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

SELECT First\_name, Last\_name FROM Employee;

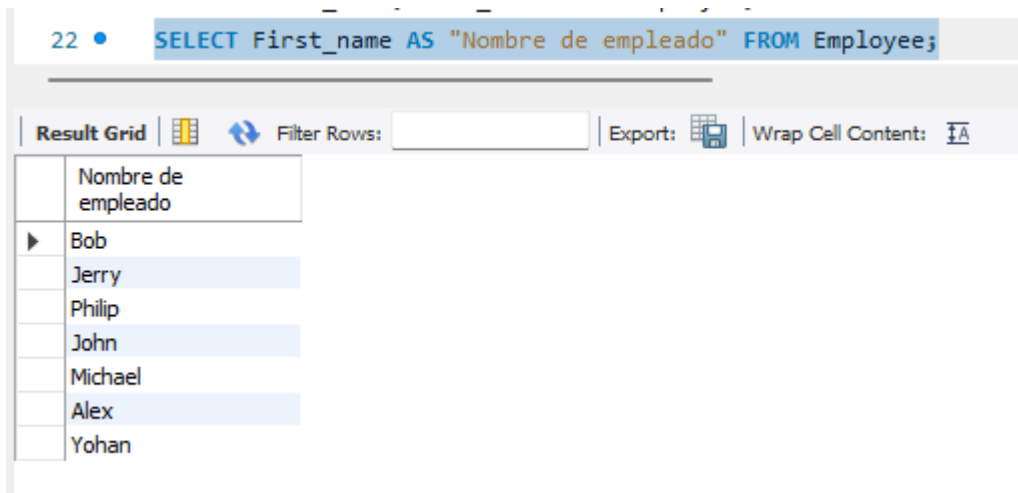
```
21 • SELECT First_name, Last_name FROM Employee;  
22 • SELECT First_name AS "Nombre de empleado" FROM Employee;
```



First_name	Last_name
Bob	Kinto
Jerry	Kansxo
Philip	Jose
John	Abraham
Michael	Mathew
Alex	Chreketo
Yohan	Soso

7. Obtener todos los valores de la columna “First\_name” usando el alias “Nombre de empleado”.

SELECT First\_name AS "Nombre de empleado" FROM Employee;



Nombre de empleado
Bob
Jerry
Philip
John
Michael
Alex
Yohan

8. Obtener todos los valores de la columna “Last\_name” en minúsculas.

SELECT LOWER(Last\_name) FROM Employee;

23 • `SELECT LOWER(Last_name) FROM Employee;`

LOWER(Last_name)
kinto
kansxo
jose
abraham
mathew
chreketo
soso

9. Obtener todos los valores de la columna “Last\_name” en mayúsculas.

`SELECT UPPER(Last_name) FROM Employee;`

24 • `SELECT UPPER(Last_name) FROM Employee;`

UPPER(Last_name)
KINTO
KANSXO
JOSE
ABRAHAM
MATHEW
CHREKETO
SOSO

10. Obtener los nombre únicos de la columna “Department”.

`SELECT DISTINCT Department FROM Employee;`

25 • `SELECT DISTINCT Department FROM Employee;`

Department
Finance
IT
Banking
Insurance

11. Obtener los primeros 4 caracteres de todos los valores de la columna “First\_name”.

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

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

Result Grid		Filter Rows:	Export:	Wrap Cell Cont
	SUBSTRING(First_name, 1, 4)			
▶	Bob			
	Jerr			
	Phil			
	John			
	Mich			
	Alex			
	Yoha			

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

`SELECT INSTR(First_name, 'h') FROM Employee WHERE First_name = 'John';`

27 • `SELECT INSTR(First_name, 'h') FROM Employee WHERE First_name = 'John';`




Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	INSTR(First_name, 'h')			
▶	3			

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

`SELECT RTRIM(First_name) FROM Employee;`



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

Result Grid |   Filter Rows:  | Export: 

	RTRIM(First_name)
▶	Bob
	Jerry
	Philip
	John
	Michael
	Alex
	Yohan

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

`SELECT LTRIM(First_name) FROM Employee;`

29 • `SELECT LTRIM(First_name) FROM Employee;`

Result Grid

Filter Rows:

Export:

LTRIM(First\_name)

▶ Bob

Jerry

Philip

John

Michael

Alex

Yohan

## Algebra relacional

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

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

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

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

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

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

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

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

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

$\text{Reward} \leftarrow \text{Reward} \cup \{1, \text{"2019-05-11"}, 1000\}$

$\text{Reward} \leftarrow \text{Reward} \cup \{2, \text{"2019-02-15"}, 5000\}$

$\text{Reward} \leftarrow \text{Reward} \cup \{3, \text{"2019-04-22"}, 2000\}$

$\text{Reward} \leftarrow \text{Reward} \cup \{4, \text{"2019-06-20"}, 8000\}$

5. Obtener todos los empleados.

$\sigma_{true}(\text{Employee})$

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

$\pi_{\text{First\_name}, \text{Last\_name}}(\text{Employee})$

7. Obtener todos los valores de la columna “First\_name” usando el alias “Nombre de empleado”.

$\pi_{\text{First\_name} \rightarrow \text{Nombre\_de\_empleado}}(\text{Employee})$

8. Obtener todos los valores de la columna “Last\_name” en minúsculas.

$\pi_{\text{Last\_name} \rightarrow \text{Nombre\_de\_empleado}}(\text{Employee})$

9. Obtener todos los valores de la columna “Last\_name” en mayúsculas.

$\pi_{\text{Last\_name} \rightarrow \text{Nombre\_de\_empleado}}(\text{Employee})$

10. Obtener los nombre únicos de la columna “Departament”.

$\delta_{Department}(Employee)$

## 5. Conclusión

En esta práctica, se llevó a cabo una serie de ejercicios utilizando SQL para la creación y manipulación de bases de datos relacionales. A través de la creación de las tablas Employee y Reward, se estableció una estructura básica para almacenar información relevante sobre empleados y sus recompensas. Posteriormente, se realizaron varias consultas que ilustraron cómo extraer y manipular datos de estas tablas, desde la simple recuperación de todos los registros hasta operaciones más específicas como la obtención de valores únicos y la transformación de texto.

Esta práctica no solo reforzó los conceptos fundamentales del manejo de bases de datos relacionales, sino que también demostró la flexibilidad y el poder de SQL como herramienta para manejar grandes volúmenes de datos de manera eficiente. Al comprender y aplicar estas técnicas, se establecen las bases para desarrollar habilidades avanzadas en la gestión de datos, esenciales en entornos de trabajo que requieren un análisis de información preciso y confiable.