

# REPORTE DE PRÁCTICA

## 1.3. Practica. Algebra relacional y SQL 2

ALUMNO: Cristian Cristobal Silverio  
Dr. Eduardo Cornejo-Velazquez



## 1. Introducción

En el ambito de las bases de datos, el manejo eficiente y controlado de la informaci´on es de suma importancia. Para ello, hacemos uso de diversas herramientas que permiten interactuar con los datos de manera efectiva. En esta practica, el ´algebra relacional, SQL y MySQL son herramientas las cuales nos facilitan la gesti´on y manipulaci´on de los datos.

## Marco teórico

### álgebra relacional

El álgebra relacional es un lenguaje formal basado en la teoría de conjuntos que se utiliza para describir las operaciones sobre las relaciones en una base de datos relacional. Este lenguaje proporciona un conjunto de operaciones básicas, como la selección, proyección, unión, diferencia, producto cartesiano y renombramiento, que permiten manipular y consultar los datos almacenados en las tablas.

### SQL

SQL es un lenguaje de programación estándar utilizado para gestionar y manipular bases de datos relacionales. Permite realizar diversas operaciones, como la creación de tablas, inserción de datos, actualización, eliminación y consultas complejas.

### MySQL

MySQL es un sistema de gestión de bases de datos relacional de código abierto que utiliza SQL como lenguaje de consulta. Es ampliamente utilizado debido a su robustez, escalabilidad y facilidad de uso. MySQL permite a los usuarios crear y gestionar bases de datos de manera eficiente, soportando una amplia gama de aplicaciones, desde pequeños proyectos hasta grandes sistemas empresariales.

### 3. Herramientas empleadas

1. SQL: Lenguaje de consulta estructurado utilizado para interactuar con la base de datos.
2. MySQL: Sistema de gesti3n de bases de datos relacional que permite la creaci3n y manipulaci3n de bases de datos.
3. MySQL Online Editor: Herramienta en l3nea para compilar y ejecutar sentencias SQL, accesible en OneCompiler

## 4. Desarrollo

### SENTENCIAS SQL

1. Obtener el tamaño del texto en todos los valores de la columna \First\_name":  
`SELECT LENGTH(First_name) FROM employee;`
2. Obtener el nombre de todos los empleados después de reemplazar 'o' con '#':  
`SELECT REPLACE(First_name, 'o', '#') FROM employee;`
3. Obtener el nombre y apellido de todos los empleados en una sola columna separados por \\_":  
`SELECT CONCAT (First_name, '_', Last_name) FROM employee;`
4. Obtener el año, mes y día de la columna \Joining\_date":  
`SELECT  
 YEAR(Joining_date),  
 MONTH(Joining_date),  
 DAY(Joining_date)  
FROM employee;`
5. Obtener todos los empleados en orden ascendente por nombre:  
`SELECT First_name FROM employee ORDER BY First_name ASC;`
6. Obtener todos los empleados en orden descendente por nombre:  
`SELECT First_name FROM employee ORDER BY First_name DESC;`
7. Obtener todos los empleados en orden ascendente por nombre y en orden descendente por salario:  
`SELECT First_name, Salary FROM employee ORDER BY First_name ASC, Salary DESC;`
8. Obtener todos los empleados con el nombre \Bob":  
`SELECT First_name FROM employee WHERE First_name = 'Bob';`
9. Obtener todos los empleados con el nombre \Bob" o \Alex":  
`SELECT First_name FROM employee WHERE First_name = 'Bob' OR First_name = 'Alex';`
10. Obtener todos los empleados que no tengan el nombre \Bob" o \Alex":  
`SELECT First_name FROM employee WHERE First_name NOT IN ('Bob', 'Alex');`
11. ¿Qué es una inyección SQL? es un tipo de vulnerabilidad en la que un atacante usa un trozo de código SQL para manipular una base de datos y acceder a información potencialmente valiosa.

## SENTENCIAS SQL A ALGEBRA RELACIONAL

$\pi_{\text{FirstName}}(\text{Employee})$

$\pi_{\text{FirstName}}(\text{Employee})$

$\pi_{\text{FirstName}, \text{LastName}}(\text{Employee})$

$\pi_{\text{JoiningDate}}(\text{Employee})$

$\pi_{\text{FirstName}}(\text{Employee})$

$\pi_{\text{FirstName}}(\text{Employee})$

$\pi_{\text{FirstName}, \text{Salary}}(\text{Employee})$

$\pi_{\text{FirstName}}(\sigma_{\text{FirstName} = \text{'Bob'}}(\text{Employee}))$

$\pi_{\text{FirstName}}(\sigma_{\text{FirstName} = \text{'Bob'}} \text{ OR } \text{FirstName} = \text{'Alex'}}(\text{Employee}))$

$\pi_{\text{FirstName}}(\sigma_{\text{NOT IN ('Bob', 'Alex')}}(\text{Employee}))$

## 5. Conclusiones

En esta actividad, hemos explorado diversas operaciones de algebra relacional y su implementacion en SQL. A traves de la creacion y manipulacion de tablas, hemos aprendido a realizar consultas basicas y avanzadas, como la selecci'on, proyeccion, renombramiento y eliminaci'on de duplicados. Ademias, hemos visto c'omo insertar registros y c'omo utilizar alias para mejorar la legibilidad de nuestras consultas. Esta actividad no solo nos ha permitido reforzar nuestros conocimientos en SQL y algebra relacional, sino que tambien nos ha brindado una comprension mas profunda de como se estructuran y manipulan los datos en una base de datos relacional. Estas habilidades son esenciales para cualquier profesional que trabaje con bases de datos, ya que permiten realizar consultas eficientes y precisa