Nombre: Hernandez Sanchez Juan German

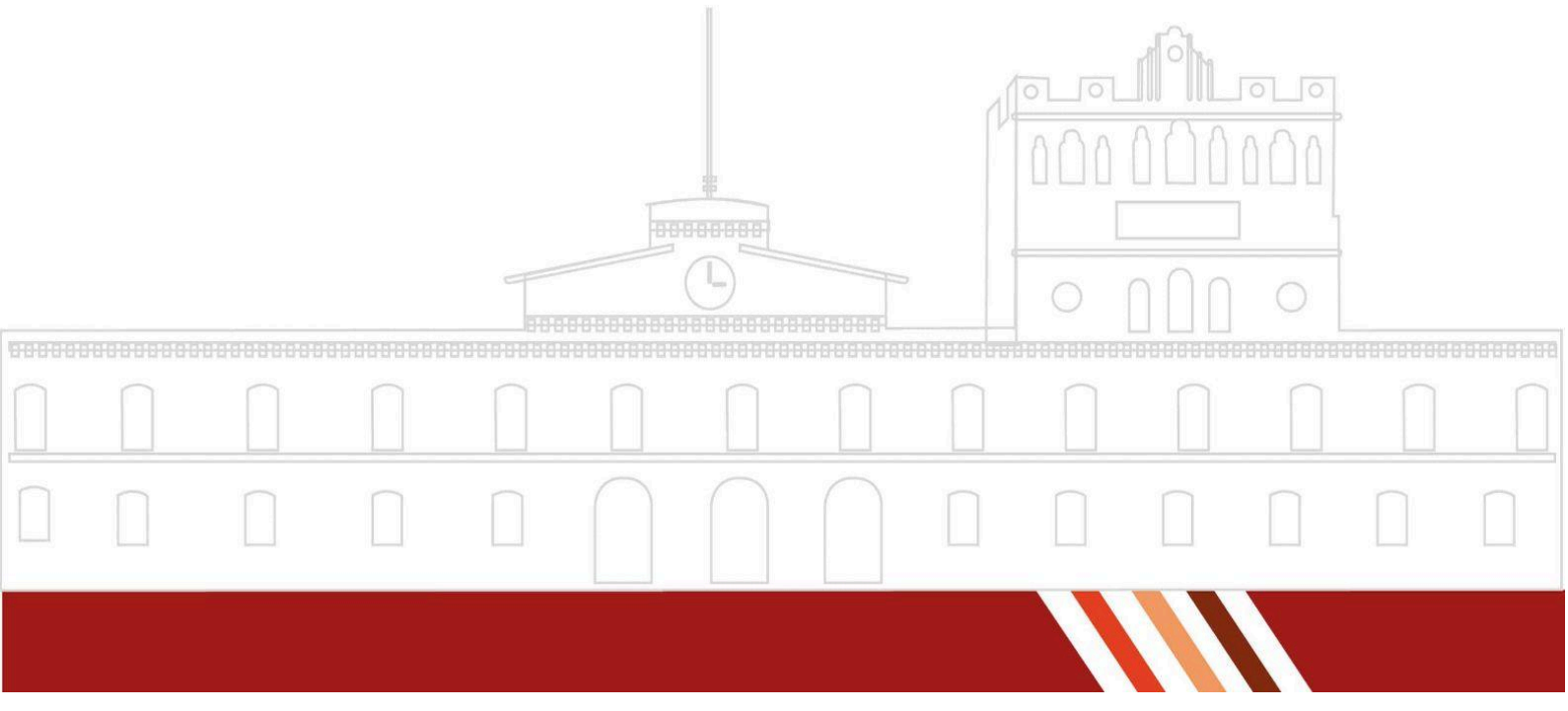
Materia: Base de datos distribuidas

No. De Cuenta: 472549

Catedrático: Eduardo Cornejo Velazquez

Carrera: Licenciatura en Ciencias Computacionales

Semestre: Sexto Grupo: “2”



Introducción

En esta solución de ejercicios llevaremos a cabo e implementaremos varias herramientas las cuales serán de mucha utilidad para resolver dichos ejercicios, el algebra relacional asi como el SQL y el MySQL serán nuestras herramientas a utilizar y a continuación les redactare un poquito de ellas.

Marco teórico

Herramientas empleadas

Como había comentado anteriormente se utilizaron varias herramientas y aqui les explico cuales y cual fue es su funcionamiento.

LaTeX es un sistema de preparación de documentos de alta calidad, especialmente adecuado para documentos que contienen fórmulas matemáticas complejas. Es ampliamente utilizado en la academia y en la publicación científica.

El álgebra relacional es un conjunto de operaciones matemáticas que se utilizan para manipular y consultar datos en una base de datos relacional. Las operaciones básicas incluyen:

- Selección (σ): Filtra filas que cumplen con una condición específica.
- Proyección (π): Selecciona columnas específicas de una tabla.
- Unión (∪): Combina las filas de dos tablas.
- Intersección (∩): Devuelve las filas comunes a dos tablas.
- Diferencia (-): Devuelve las filas que están en una tabla pero no en la otra.
- Producto Cartesiano (×): Combina todas las filas de dos tablas.
- Join (⋈): Combina filas de dos tablas basadas en una condición común.

SQL es un lenguaje estándar para gestionar y manipular bases de datos relacionales. Algunas de las operaciones más comunes en SQL incluyen:

- SELECT: Recupera datos de una o más tablas.
- INSERT: Inserta nuevos datos en una tabla.
- UPDATE: Modifica datos existentes en una tabla.
- DELETE: Elimina datos de una tabla.
- JOIN: Combina filas de dos o más tablas basadas en una condición relacionada.

MySQL es un sistema de gestión de bases de datos relacional (RDBMS) que utiliza SQL para gestionar los datos. Es conocido por su rendimiento, fiabilidad y facilidad de uso. MySQL es ampliamente utilizado en aplicaciones web y empresariales.

Desarrollo

Sentencias SQL

Presentar las sentencias para crear la base de datos y tablas. Adem`as incluir las sentencias para insertar registros.
En el Listado 1 se presenta la sentencia SQL para crear la base de datos competencia.

EJERCICIOS.

1. Obtener el tamaño del texto en todos los valores de la columna “First_name”.

```
mysql> SELECT LENGTH(First_name) AS text_size FROM Employee;
+-----------+
| text_size |
+-----------+
|         3 |
|         5 |
|         6 |
|         4 |
|         7 |
|         4 |
|         5 |
+-----------+
7 rows in set (0.00 sec)
```

2. Obtener el nombre de todos los empleados después de reemplazar ‘o’ con ‘#’.

```
mysql> SELECT REPLACE(First_name, 'o', '#') AS replaced_name FROM Employee;
+---------------+
| replaced_name |
+---------------+
| B#b           |
| Jerry         |
| Philip        |
| J#hn          |
| Michael       |
| Alex          |
| Y#han         |
+---------------+
7 rows in set (0.01 sec)
```

3.- Obtener el nombre y apellido de todos los empleados en una sola columna separados por “_”.

```
mysql> SELECT CONCAT(First_name, '_', Last_name) AS full_name FROM Employee;
+-----------------+
| full_name       |
+-----------------+
| Bob_Kinto       |
| Jerry_Kansxo    |
| Philip_Jose     |
| John_Abraham    |
| Michael_Mathew  |
| Alex_chreketo   |
| Yohan_Soso      |
+-----------------+
7 rows in set (0.01 sec)
```

4. Obtener el año, mes y día de la columna “Joining_date”.

```
mysql> SELECT YEAR(Joining_date) AS year, MONTH(Joining_date) AS month, DAY(Joining_date) AS day FROM Employee;
+------+-------+------+
| year | month | day  |
+------+-------+------+
| 2019 |     1 |   20 |
| 2019 |     1 |   15 |
| 2019 |     2 |    5 |
| 2019 |     2 |   25 |
| 2019 |     2 |   28 |
| 2019 |     5 |   10 |
| 2019 |     6 |   20 |
+------+-------+------+
7 rows in set (0.01 sec)
```

5. Obtener todos los empleados en orden ascendente por nombre.

```
mysql> SELECT * FROM Employee ORDER BY First_name ASC;
+-------------+------------+-----------+---------+--------------+------------+
| Employee_id | First_name | Last_name | Salary  | Joining_date | Department |
+-------------+------------+-----------+---------+--------------+------------+
|           6 | Alex       | chreketo  | 4000000 | 2019-05-10   | IT         |
|           1 | Bob        | Kinto     | 1000000 | 2019-01-20   | Finance    |
|           2 | Jerry      | Kansxo    | 6000000 | 2019-01-15   | IT         |
|           4 | John       | Abraham   | 2000000 | 2019-02-25   | Insurance  |
|           5 | Michael    | Mathew    | 2200000 | 2019-02-28   | Finance    |
|           3 | Philip     | Jose      | 8900000 | 2019-02-05   | Banking    |
|           7 | Yohan      | Soso      | 1230000 | 2019-06-20   | Banking    |
+-------------+------------+-----------+---------+--------------+------------+
7 rows in set (0.00 sec)
```

6. Obtener todos los empleados en orden descendente por nombre.

```
mysql> SELECT * FROM Employee ORDER BY First_name DESC;
+-------------+------------+-----------+---------+--------------+------------+
| Employee_id | First_name | Last_name | Salary  | Joining_date | Department |
+-------------+------------+-----------+---------+--------------+------------+
|           7 | Yohan      | Soso      | 1230000 | 2019-06-20   | Banking    |
|           3 | Philip     | Jose      | 8900000 | 2019-02-05   | Banking    |
|           5 | Michael    | Mathew    | 2200000 | 2019-02-28   | Finance    |
|           4 | John       | Abraham   | 2000000 | 2019-02-25   | Insurance  |
|           2 | Jerry      | Kansxo    | 6000000 | 2019-01-15   | IT         |
|           1 | Bob        | Kinto     | 1000000 | 2019-01-20   | Finance    |
|           6 | Alex       | chreketo  | 4000000 | 2019-05-10   | IT         |
+-------------+------------+-----------+---------+--------------+------------+
7 rows in set (0.00 sec)
```

7. Obtener todos los empleados en orden ascendente por nombre y en orden descendente por salario.

```
mysql> SELECT * FROM Employee ORDER BY First_name ASC, Salary DESC;
+-------------+------------+-----------+---------+--------------+------------+
| Employee_id | First_name | Last_name | Salary  | Joining_date | Department |
+-------------+------------+-----------+---------+--------------+------------+
|           6 | Alex       | chreketo  | 4000000 | 2019-05-10   | IT         |
|           1 | Bob        | Kinto     | 1000000 | 2019-01-20   | Finance    |
|           2 | Jerry      | Kansxo    | 6000000 | 2019-01-15   | IT         |
|           4 | John       | Abraham   | 2000000 | 2019-02-25   | Insurance  |
|           5 | Michael    | Mathew    | 2200000 | 2019-02-28   | Finance    |
|           3 | Philip     | Jose      | 8900000 | 2019-02-05   | Banking    |
|           7 | Yohan      | Soso      | 1230000 | 2019-06-20   | Banking    |
+-------------+------------+-----------+---------+--------------+------------+
7 rows in set (0.01 sec)
```

8. Obtener todos los empleados con el nombre “Bob”.

```
mysql> SELECT * FROM Employee WHERE First_name = 'Bob';
+-------------+------------+-----------+---------+--------------+------------+
| Employee_id | First_name | Last_name | Salary  | Joining_date | Department |
+-------------+------------+-----------+---------+--------------+------------+
|           1 | Bob        | Kinto     | 1000000 | 2019-01-20   | Finance    |
+-------------+------------+-----------+---------+--------------+------------+
1 row in set (0.00 sec)
```

9. Obtener todos los empleados con el nombre “Bob” o “Alex”.

```
mysql> SELECT * FROM Employee WHERE First_name = 'Bob' OR First_name = 'Alex';
+-------------+------------+-----------+---------+--------------+------------+
| Employee_id | First_name | Last_name | Salary  | Joining_date | Department |
+-------------+------------+-----------+---------+--------------+------------+
|           1 | Bob        | Kinto     | 1000000 | 2019-01-20   | Finance    |
|           6 | Alex       | chreketo  | 4000000 | 2019-05-10   | IT         |
+-------------+------------+-----------+---------+--------------+------------+
2 rows in set (0.00 sec)
```

10. Obtener todos los empleados que no tengan el nombre “Bob” o “Alex”.

```
mysql> SELECT * FROM Employee WHERE First_name != 'Bob' AND First_name != 'Alex';
+-------------+------------+-----------+---------+--------------+------------+
| Employee_id | First_name | Last_name | Salary  | Joining_date | Department |
+-------------+------------+-----------+---------+--------------+------------+
|           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 | 2019-02-28   | Finance    |
|           7 | Yohan      | Soso      | 1230000 | 2019-06-20   | Banking    |
+-------------+------------+-----------+---------+--------------+------------+
5 rows in set (0.00 sec)
```

11. ¿Qué es una inyección SQL?

Una inyección SQL (SQL Injection) es un tipo de vulnerabilidad de seguridad que permite a un atacante interferir con las consultas que una aplicación hace a su base de datos. Esto se logra insertando código SQL malicioso en una entrada de usuario que luego es ejecutada por la base de datos.

¿Cómo funciona?

La inyección SQL ocurre cuando una aplicación no valida o desinfecta adecuadamente la entrada del usuario antes de incluirla en una consulta SQL. Por ejemplo, si una aplicación web permite a los usuarios ingresar su nombre de usuario y contraseña para iniciar sesión, un atacante podría ingresar un código SQL malicioso en lugar de un nombre de usuario legítimo.

Conclusiones

La creación e implementación de esta base de datos aborda muchos puntos y dará muchas ventajas, asi mismo también implementar herramientas nos ayuda para poder hacer la base de datos y solucionar ejercicios de una forma más sencilla.

Referencias

Álgebra relacional _ AcademiaLab. (s. f.). https://academia-lab.com/enciclopedia/algebra-relacional/

SEAS, Estudios Superiores Abiertos. (2022, 16 mayo). *Diferencia entre SQL y MySQL | Blog SEAS*. Blog de SEAS. https://www.seas.es/blog/informatica/diferencia-entre-sql-y-mysql/