



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO
INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA
LICENCIATURA EN CIENCIAS COMPUTACIONALES
BASES DE DATOS DISTRIBUIDAS

EQUIPO:

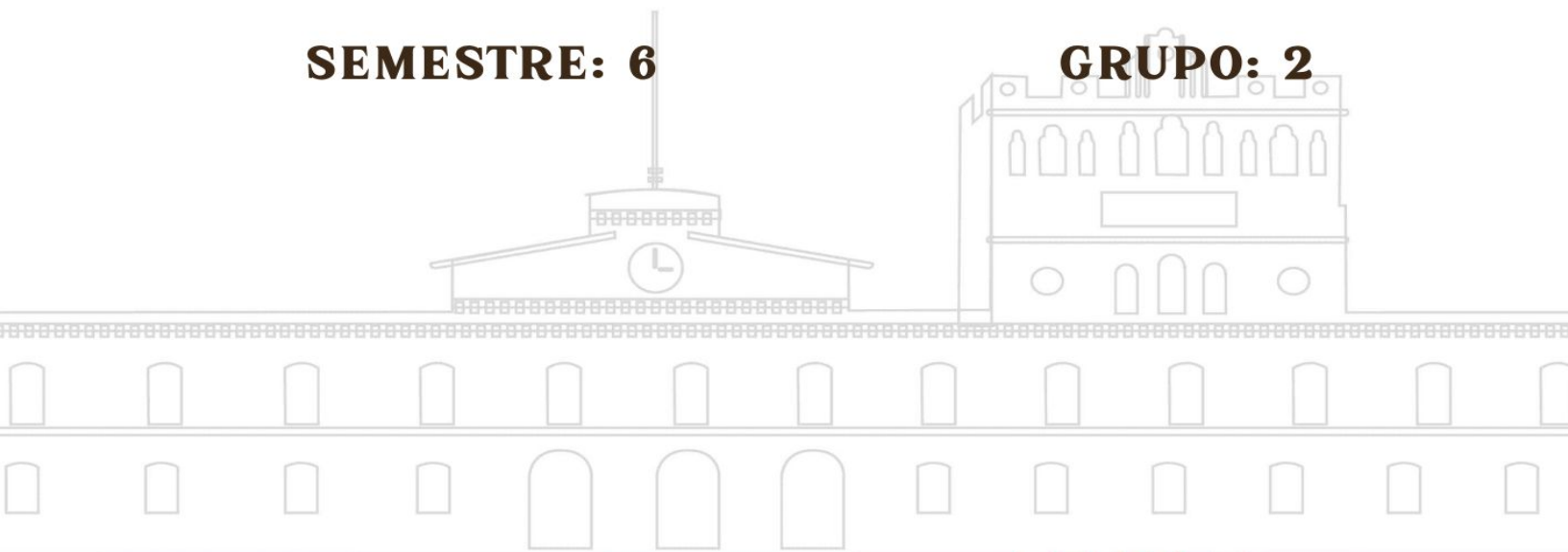
VILLEDA TREJO ADÁN

HERNANDEZ JANETT

EDUARDO CORNEJO VELAZQUEZ

SEMESTRE: 6

GRUPO: 2



Introducción

Este conjunto de ejercicios prácticos está diseñado para fortalecer los conocimientos y habilidades en el manejo de bases de datos distribuidas utilizando MySQL. A través de la manipulación de dos tablas fundamentales, Employee y Reward, se abordan operaciones básicas y avanzadas de consulta y manipulación de datos. Los ejercicios incluyen desde la creación de tablas y la inserción de registros, hasta la ejecución de consultas que permiten extraer información específica, aplicar funciones para transformar datos y utilizar alias para mejorar la legibilidad de los resultados. Este enfoque práctico facilita la comprensión de conceptos clave en bases de datos relacionales y el dominio de comandos SQL esenciales para la gestión eficiente de la información.

Marco Teorico

Álgebra Relacional

El álgebra relacional es un lenguaje de consulta procedimental fundamental en las bases de datos relacionales. A través del álgebra relacional, el usuario puede expresar consultas que indican al sistema qué operaciones debe realizar sobre las relaciones (tablas) para obtener los resultados deseados. Este lenguaje utiliza operadores específicos que actúan sobre conjuntos de datos, permitiendo manipular y obtener información de manera precisa y estructurada. Las operaciones fundamentales incluyen:

Selección σ : Extrae filas que cumplen una condición.

Proyección Π : Extrae columnas específicas de una tabla.

Unión (\cup): Combina los resultados de dos relaciones.

Renombramiento (ρ): Cambia el nombre de una relación o atributos.

Diferencia ($-$): Obtiene los registros que están en una relación pero no en otra.

Producto cartesiano (\times): Combina todas las filas de dos relaciones.

SQL (Structured Query Language)

SQL es un lenguaje declarativo para la gestión y manipulación de bases de datos relacionales. A diferencia del álgebra relacional, SQL permite especificar qué información se desea obtener sin necesidad de indicar cómo se obtiene. SQL incluye instrucciones para la definición de datos (DDL), manipulación de datos (DML) y control de datos (DCL).

Entre las sentencias SQL utilizadas en los ejercicios prácticos destacan:

CREATE TABLE: Para definir tablas y sus estructuras.

INSERT INTO: Para insertar registros en las tablas.

SELECT: Para consultar datos, incluyendo el uso de alias (AS), funciones como LOWER(), UPPER(), y filtros.

Funciones de cadenas y manipulación de texto para transformar datos (como cambiar a mayúsculas/minúsculas).

Uso de operadores y cláusulas para filtrar, ordenar y limitar resultados.

SQL es esencial para interactuar con bases de datos relacionales, proporcionando una interfaz estándar compatible con múltiples sistemas gestores.

MySQL

MySQL es un sistema gestor de bases de datos relacional de código abierto muy utilizado para gestionar datos en entornos de desarrollo y producción. Permite almacenar, modificar y consultar datos mediante sentencias SQL. MySQL combina eficiencia, facilidad de uso y soporte amplio, siendo una herramienta común en aplicación web y de análisis de datos.

En los ejercicios vistos se utiliza MySQL para:

Crear estructuras de base de datos (CREATE TABLE).

Insertar múltiples registros (INSERT INTO).

Realizar consultas básicas y avanzadas con funciones de cadenas.

Trabajar con alias para mejorar la presentación de resultados.

MySQL también soporta el uso del álgebra relacional en la optimización interna de sus consultas, aunque los usuarios interactúan mediante SQL.

Herramientas Utilizadas y Desarrollo

Para el desarrollo de los ejercicios se empleó MySQL como sistema gestor y un entorno o consola para ejecutar sentencias SQL. Los ejercicios prácticos incluyen la creación de tablas Employee y Reward, la inserción de datos, y múltiples consultas para manipular y obtener información:

Mostrar todos los empleados.

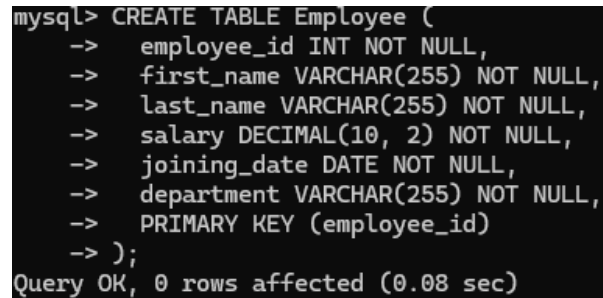
Extraer nombres con alias descriptivos.

Transformar cadenas de texto a mayúsculas o minúsculas.

Operar sobre los datos con funciones nativas SQL.

Sentencia SQL

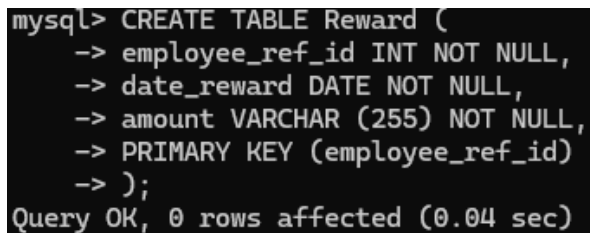
```
CREATE TABLE Employee(  
  employeeid INT NOT NULL,  
  firstname VARCHAR(255) NOT NULL,  
  lastname VARCHAR(255) NOT NULL,  
  salary DECIMAL(10, 2) NOT NULL,  
  joiningdate DATE NOT NULL,  
  department VARCHAR(255) NOT NULL,  
  PRIMARY KEY (employeeid)  
);
```



```
mysql> CREATE TABLE Employee (  
->  employee_id INT NOT NULL,  
->  first_name VARCHAR(255) NOT NULL,  
->  last_name VARCHAR(255) NOT NULL,  
->  salary DECIMAL(10, 2) NOT NULL,  
->  joining_date DATE NOT NULL,  
->  department VARCHAR(255) NOT NULL,  
->  PRIMARY KEY (employee_id)  
-> );  
Query OK, 0 rows affected (0.08 sec)
```

Figure 1: Sentencia en MSQL

```
CREATE TABLE Reward (  
  employeerefid INT NOT NULL,  
  datereward DATE NOT NULL,  
  amount VARCHAR (255) NOT NULL,  
  PRIMARY KEY (employeerefid)  
);
```



```
mysql> CREATE TABLE Reward (  
->  employee_ref_id INT NOT NULL,  
->  date_reward DATE NOT NULL,  
->  amount VARCHAR (255) NOT NULL,  
->  PRIMARY KEY (employee_ref_id)  
-> );  
Query OK, 0 rows affected (0.04 sec)
```

Figure 2: Sentencia en MSQL

SENTANCIAS INSERT INTO

INSERT INTO Employee values

```
(1, 'John', 'Doe', 50000.00, '2020-01-15', 'Marketing'),  
(2, 'Jane', 'Smith', 55000.00, '2019-03-23', 'HR'),  
(3, 'Michael', 'Brown', 60000.00, '2018-07-11', 'Finance'),  
(4, 'Emily', 'Davis', 65000.00, '2021-05-19', 'IT'),  
(5, 'David', 'Wilson', 52000.00, '2022-08-01', 'Marketing'),  
(6, 'Sarah', 'Johnson', 48000.00, '2023-01-09', 'Sales'),  
(7, 'Paul', 'Martinez', 53000.00, '2020-11-22', 'HR');
```

```
mysql> INSERT INTO Employee values  
-> (1, 'John', 'Doe', 50000.00, '2020-01-15', 'Marketing'),  
-> (2, 'Jane', 'Smith', 55000.00, '2019-03-23', 'HR'),  
-> (3, 'Michael', 'Brown', 60000.00, '2018-07-11', 'Finance'),  
-> (4, 'Emily', 'Davis', 65000.00, '2021-05-19', 'IT'),  
-> (5, 'David', 'Wilson', 52000.00, '2022-08-01', 'Marketing'),  
-> (6, 'Sarah', 'Johnson', 48000.00, '2023-01-09', 'Sales'),  
-> (7, 'Paul', 'Martinez', 53000.00, '2020-11-22', 'HR');  
Query OK, 7 rows affected (0.06 sec)  
Records: 7 Duplicates: 0 Warnings: 0
```

Figure 3: Ilustracion

INSERT INTO Reward VALUES

```
(1, '2023-05-10', 500.00),  
(2, '2023-06-15', 750.00),  
(3, '2023-07-01', 600.00),  
(4, '2023-08-05', 800.00);
```

```
mysql> INSERT INTO Reward VALUES  
-> (1, '2023-05-10', 500.00),  
-> (2, '2023-06-15', 750.00),  
-> (3, '2023-07-01', 600.00),  
-> (4, '2023-08-05', 800.00);  
Query OK, 4 rows affected (0.02 sec)  
Records: 4 Duplicates: 0 Warnings: 0
```

Figure 4: Ilustracion

Obtener todos los empleados

SELECT *FROM EMPLOYEE;

```
mysql> SELECT * FROM Employee;  
+-----+-----+-----+-----+-----+-----+  
| employee_id | first_name | last_name | salary | joining_date | department |  
+-----+-----+-----+-----+-----+-----+  
| 1 | John | Doe | 50000.00 | 2020-01-15 | Marketing |  
| 2 | Jane | Smith | 55000.00 | 2019-03-23 | HR |  
| 3 | Michael | Brown | 60000.00 | 2018-07-11 | Finance |  
| 4 | Emily | Davis | 65000.00 | 2021-05-19 | IT |  
| 5 | David | Wilson | 52000.00 | 2022-08-01 | Marketing |  
| 6 | Sarah | Johnson | 48000.00 | 2023-01-09 | Sales |  
| 7 | Paul | Martinez | 53000.00 | 2020-11-22 | HR |  
+-----+-----+-----+-----+-----+-----+  
7 rows in set (0.02 sec)
```

Figure 5: Ilustracion

Obtener el primer nombre y apellido de todos los empleados

SELEC firstname, lastname FROM Employee;

```
mysql> SELECT first_name, last_name FROM Employee;
+-----+-----+
| first_name | last_name |
+-----+-----+
| John      | Doe       |
| Jane      | Smith     |
| Michael   | Brown     |
| Emily     | Davis     |
| David     | Wilson    |
| Sarah     | Johnson   |
| Paul      | Martinez  |
+-----+-----+
7 rows in set (0.01 sec)
```

Figure 6: Ilustracion

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

SELECT firstname AS 'Nombre de employee' FORM Employee;

```
mysql> SELECT first_name AS 'Nombre de empleado' FROM Employee;
+-----+
| Nombre de empleado |
+-----+
| John               |
| Jane               |
| Michael            |
| Emily              |
| David              |
| Sarah              |
| Paul               |
+-----+
7 rows in set (0.00 sec)
```

Figure 7: Ilustracion

Obtener todos los valores de la columna “Lastname” en mayúsculas

SELECT lower(lastname) AS ”Nombre de emplado” FROM Employee

```
mysql> SELECT UPPER(last_name) AS last_name_may FROM Employee;
+-----+
| last_name_may |
+-----+
| DOE           |
| SMITH         |
| BROWN         |
| DAVIS         |
| WILSON        |
| JOHNSON       |
| MARTINEZ      |
+-----+
7 rows in set (0.01 sec)
```

Figure 8: Ilustracion

Obtener todos los valores de la columna “Lastname” en minúsculas

SELECT upper(lastname) AS "Nombre de empleado" FROM Employee

```
mysql> SELECT LOWER(last_name) AS last_name_min FROM Employee;
+-----+
| last_name_min |
+-----+
| doe           |
| smith         |
| brown         |
| davis         |
| wilson        |
| johnson       |
| martinez      |
+-----+
7 rows in set (0.04 sec)
```

Figure 9: Ilustracion

CONCLUSIONES

El álgebra relacional es la base teórica fundamental para el manejo y consulta de bases de datos relacionales. Su conjunto de operaciones permite estructurar de forma clara y precisa cómo extraer y manipular datos. SQL es el lenguaje estándar que traduce esas operaciones teóricas en instrucciones concretas que los sistemas gestores de bases de datos pueden procesar, facilitando el trabajo con datos mediante comandos intuitivos y versátiles.

MySQL, como sistema gestor ampliamente utilizado, proporciona un entorno práctico para aplicar estos conceptos, permitiendo crear, modificar y consultar bases de datos de manera eficiente y segura.

La combinación del conocimiento teórico y la práctica con sentencias SQL específicas fortalece la capacidad para diseñar y gestionar bases de datos que respondan a necesidades reales de información.

Estos fundamentos son esenciales para cualquier profesional que desee trabajar con bases de datos, garantizando un manejo adecuado y optimizado de los datos almacenados.