Universidad Autónoma del Estado de Hidalgo

ICBI

REPORTE DE PRÁCTICA

1.3. Práctica. Álgebra relacional y SQL

ALUMNO: Cristian Cristobal Silverio

Dr. Eduardo Cornejo-Velázquez

1. Introducción

En el ámbito de las bases de datos, el manejo eficiente y controlado de la información es de suma importancia. Para ello, hacemos uso de diversas herramientas que permiten interactuar con los datos de manera efectiva. En esta practica, el álgebra relacional, SQL y MySQL son herramientas las cuales nos facilitan la gestión y manipulación de los datos.

2. 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 (Structured Query Language)

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

Para el desarrollo de esta practica utlizamos varias herramientas las cuales presentamos a continuacion.

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

4. Desarrollo

Sentencias Utilizadas en los Ejercicios

1. Crear la tabla "Employee":

```
CREATE TABLE Employee (
Employee_id INT,
First_name VARCHAR(255),
Last_name VARCHAR(255),
Salary INT,
Joining_date DATE,
Department VARCHAR(255)
);
```

2. Escribir la sintaxis para insertar 7 registros a la tabla "Employee":

```
INSERT INTO Employee (Employee_id, First_name, Last_name, Salary, Join
(1, 'Bob', 'Kinto', 1000000, '2019-01-20', 'IT'),
(2, 'Jerry', 'Nasko', 600000, '2019-02-15', 'Finance'),
(3, 'John', 'Michael', 900000, '2019-02-25', 'Banking'),
(4, 'Philip', 'Mathew', 2000000, '2020-03-10', 'Insurance'),
(5, 'Alexi', 'Cherokee', 800000, '2020-04-18', 'Finance'),
(6, 'Michael', 'Bamettu', 1200000, '2020-05-12', 'Banking'),
(7, 'Yohan', 'Soso', 1230000, '2020-06-20', 'Banking');
```

3. Escribir la sintaxis para crear la tabla "Reward":

```
CREATE TABLE Reward (
Employee_ref_id INT,
Date_rewarded DATE,
Amount INT
);
```

4. Escribir la sintaxis para insertar 4 registros a la tabla "Reward":

```
INSERT INTO Reward (Employee_ref_id, Date_rewarded, Amount) VALUES
(1, '2019-05-11', 10000),
(2, '2019-02-15', 5000),
(3, '2019-04-22', 20000),
(5, '2019-06-20', 15000);
```

5. Obtener todos los empleados:

```
SELECT * FROM Employee;
```

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

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

```
SELECT First_name, Last_name FROM Employee;
```

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

7. Obtener todos los valores de la columna "First-name" usando el alias "Nombre de empleado":

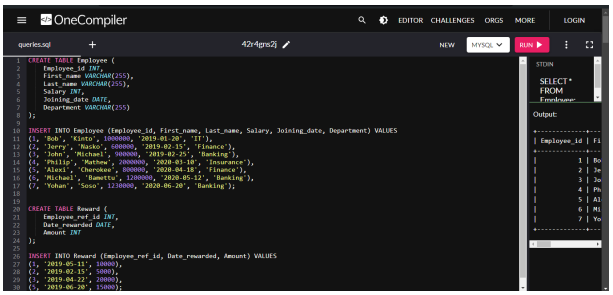

Figure 1: Consulta de todos los empleados.

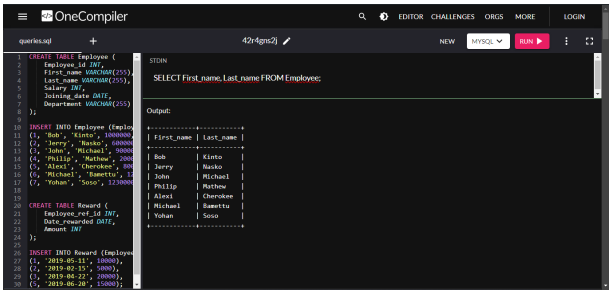

Figure 2: Consulta del primer nombre y apellido de todos los empleados.

```
SELECT First_name AS "Nombre-de-empleado" FROM Employee;
```

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

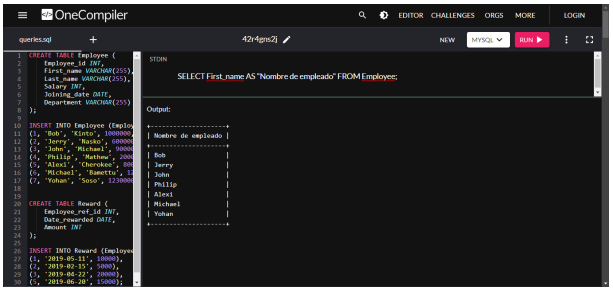

Figure 3: Consulta con alias en la columna First_name.

8. Obtener todos los valores de la columna “Last-name” en minúsculas:

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

9. Obtener todos los valores de la columna “Last-name” en mayúsculas:

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

10. Obtener los nombres únicos de la columna “Department”:

```
SELECT DISTINCT Department FROM Employee;
```

$$\delta(\pi_{\text{Department}}(\text{Employee}))$$

Figure 4: Consulta de la columna Last$_n ameenminúsculas$.

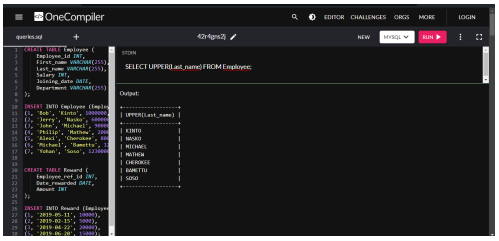

Figure 5: Consulta de la columna Last$_n ameenmayúsculas$.

11. Obtener los primeros 4 caracteres de todos los valores de la columna “First-name”:

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

12. Obtener la posición de la letra “h” en el nombre del empleado con First-name = “John”:

```
SELECT POSITION('h' IN First_name) FROM Employee WHERE First_name = 'John';
```

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;
```

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;
```

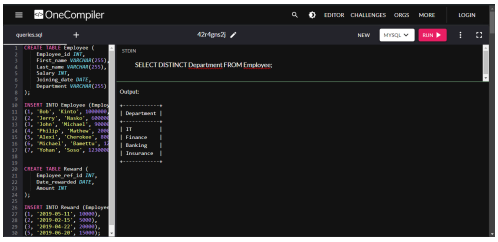

Figure 6: Consulta de nombres únicos en la columna Department.

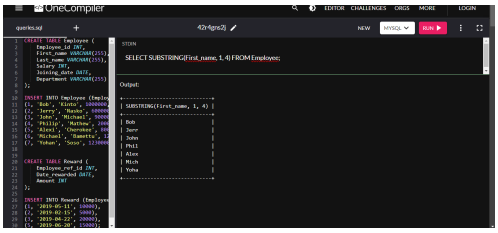

Figure 7: Consulta de los primeros 4 caracteres de la columna First$_name$.

5. Conclusiones

En esta actividad, hemos explorado diversas operaciones de álgebra relacional y su implementación en SQL. A través de la creación y manipulación de tablas, hemos aprendido a realizar consultas básicas y avanzadas, como la selección, proyección, renombramiento y eliminación de duplicados. Además, hemos visto cómo insertar registros y cómo utilizar alias para mejorar la legibilidad de nuestras consultas.

Esta actividad no solo nos ha permitido reforzar nuestros conocimientos en SQL y álgebra relacional, sino que también nos ha brindado una comprensión más profunda de cómo 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

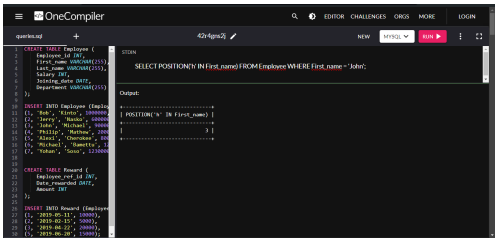

Figure 8: Consulta de la posición de la letra "h" en el nombre John.

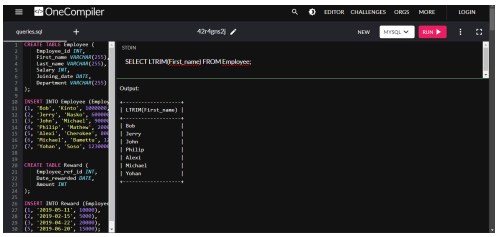

Figure 9: Consulta después de remover los espacios en blanco de la izquierda.

Referencias Bibliográficas

References

[1] http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro14/33$_algebra_relacional.html$