

REPORTE DE PRÁCTICA NO. 2

1.3. Práctica. Álgebra relacional y SQL (1)

ALUMNO:

KEVIN AXEL CHAVEZ QUIROZ



1. Introducción

Incluir una descripción de lo que trata la práctica.

En esta práctica, exploramos diversas operaciones y consultas en SQL, enfocándonos en la creación y manipulación de tablas dentro de una base de datos. Utilizamos dos tablas principales: **Employee** y **Reward**, para ilustrar cómo se pueden gestionar datos relacionados en un entorno de base de datos relacional.

Primero, aprendimos a crear tablas utilizando la sentencia **CREATE TABLE**, definiendo columnas con diferentes tipos de datos y estableciendo claves primarias y foráneas para mantener la integridad referencial. Luego, insertamos registros en estas tablas para poblarlas con datos de ejemplo.

A lo largo de la práctica, realizamos varias consultas SQL para obtener información específica de las tablas. Estas consultas incluyeron la selección de todos los registros, la obtención de valores únicos, el uso de alias para columnas, y la manipulación de cadenas de texto para mostrar datos en diferentes formatos. También exploramos funciones como **RTRIM()**, **LTRIM()**, **UPPER()**, **LOWER()**, **SUBSTRING()**, y **POSITION()** para trabajar con datos de texto.

Esta práctica nos permitió comprender mejor cómo interactuar con bases de datos SQL, realizar consultas complejas y asegurar la integridad de los datos mediante el uso de claves foráneas. Los conocimientos adquiridos son fundamentales para el desarrollo de aplicaciones que requieren el manejo eficiente de datos.

2. Herramientas empleadas

Describir qué herramientas se han utilizado...

1. MySQL Server. Esta herramienta se ocupo para la creacion de la de la base de datos y para la organizacion de esta creacion.

3. Desarrollo

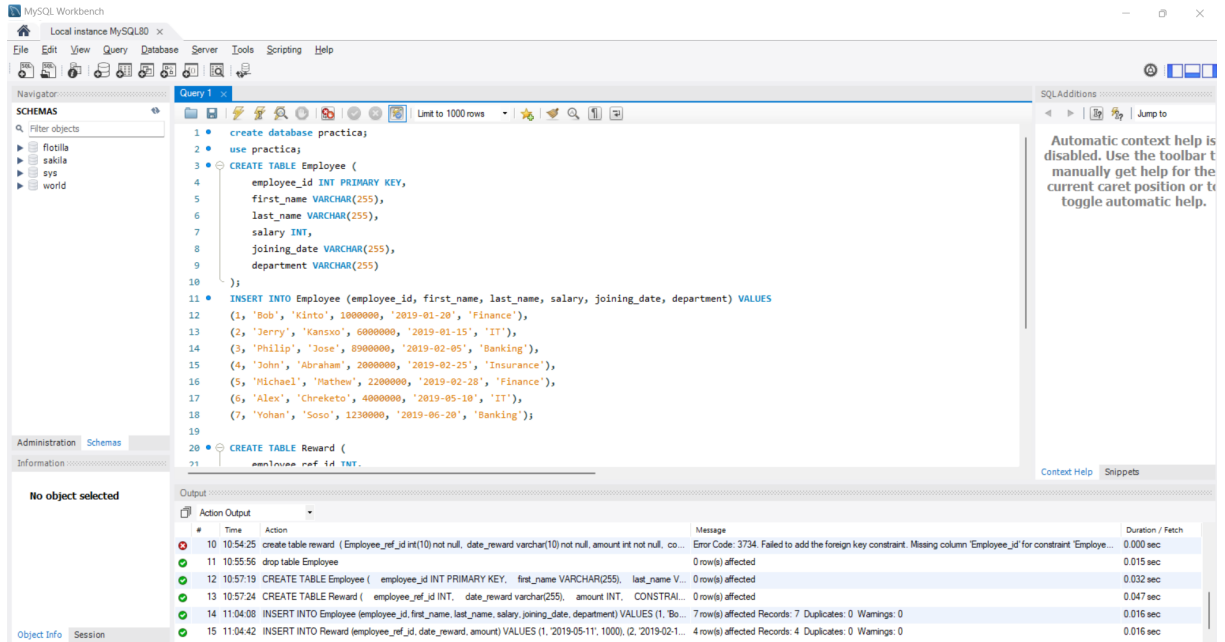


Figure 1: creacion de las tablas.

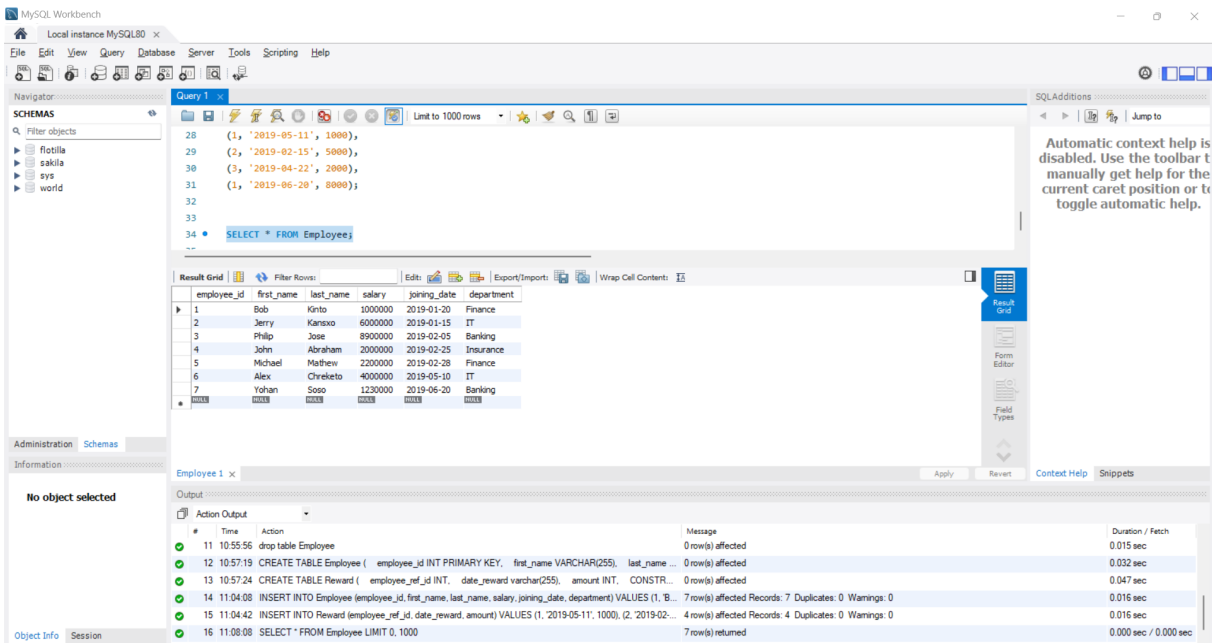


Figure 2: Obtener todos los empleados..

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS
Filter objects
flotilla
sakila
sys
world

Query 1
Limit to 1000 rows

```

29 (2, '2019-02-15', 5000),
30 (3, '2019-04-22', 2000),
31 (1, '2019-06-20', 8000);
32
33
34 • SELECT * FROM Employee;
35 • SELECT first_name, last_name FROM Employee;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Contents: [IA](#)

first_name	last_name
Bob	Kinto
Jerry	Kansxo
Philip	Jose
John	Abraham
Michael	Mathew
Alex	Chreketo
Yohan	Soso

Administration Schemas

Information

No object selected

Employee 2 x

Output

Action Output

#	Time	Action	Message
✓ 12	10:57:19	CREATE TABLE Employee (employee_id INT PRIMARY KEY, first_name VARCHAR(255), last_name ...	0 row(s) affected
✓ 13	10:57:24	CREATE TABLE Reward (employee_ref_id INT, date_reward varchar(255), amount INT, CONSTR...	0 row(s) affected
✓ 14	11:04:08	INSERT INTO Employee (employee_id, first_name, last_name, salary, joining_date, department) VALUES (1, 'B...	7 row(s) affected Records: 7 Duplicates: 0 Warnings: 0
✓ 15	11:04:42	INSERT INTO Reward (employee_ref_id, date_reward, amount) VALUES (1, '2019-05-11', 1000), (2, '2019-02-...	4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0
✓ 16	11:08:08	SELECT * FROM Employee LIMIT 0, 1000	7 row(s) returned
✓ 17	11:09:20	SELECT first_name, last_name FROM Employee LIMIT 0, 1000	7 row(s) returned

Object Info Session

Figure 3: Obtener el primer nombre y apellido de todos los empleados.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

Filter objects

flotilla
sakila
sys
world

Query 1 x

Limit to 1000 rows

```

30 (3, '2019-04-22', 2000),
31 (1, '2019-06-20', 8000);
32
33
34 • SELECT * FROM Employee;
35 • SELECT first_name, last_name FROM Employee;
36 • SELECT first_name AS "Nombre de empleado" FROM Employee;
--

```

Result Grid

Nombre de empleado

Bob
Jerry
Philip
John
Michael
Alex
Yohan

Administration Schemas

Information

No object selected

Employee 3 x

Output

Action Output

#	Time	Action	Message
✓ 13	10:57:24	CREATE TABLE Reward (employee_ref_id INT, date_reward varchar(255), amount INT, CONSTR...	0 row(s) affected
✓ 14	11:04:08	INSERT INTO Employee (employee_id, first_name, last_name, salary, joining_date, department) VALUES (1, 'B...	7 row(s) affected Records: 7 Duplicates: 0 Warnings: 0
✓ 15	11:04:42	INSERT INTO Reward (employee_ref_id, date_reward, amount) VALUES (1, '2019-05-11', 1000), (2, '2019-02-...	4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0
✓ 16	11:08:08	SELECT * FROM Employee LIMIT 0, 1000	7 row(s) returned
✓ 17	11:09:20	SELECT first_name, last_name FROM Employee LIMIT 0, 1000	7 row(s) returned
✓ 18	11:10:24	SELECT first_name AS "Nombre de empleado" FROM Employee LIMIT 0, 1000	7 row(s) returned

Object Info Session

Figure 4: Obtener todos los valores de la columna “First_name” usando el alias “Nombre de empleado”.

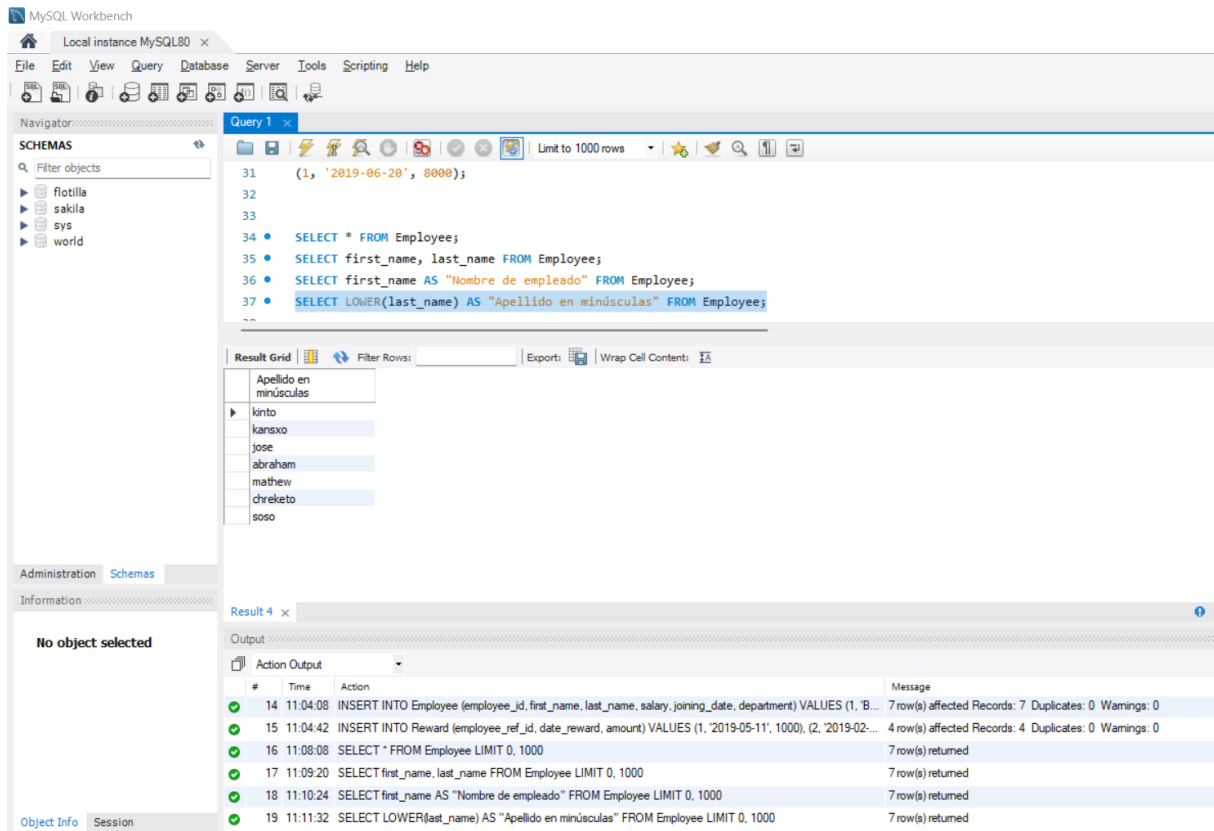


Figure 5: . Obtener todos los valores de la columna “Last_name” en minúsculas

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

- flotilla
- sakila
- sys
- world

Query 1

```
32
33
34 • SELECT * FROM Employee;
35 • SELECT first_name, last_name FROM Employee;
36 • SELECT first_name AS "Nombre de empleado" FROM Employee;
37 • SELECT LOWER(last_name) AS "Apellido en minúsculas" FROM Employee;
38 • SELECT UPPER(last_name) AS "Apellido en mayúsculas" FROM Employee;
```

Result Grid

Apellido en mayúsculas
KINTO
KANSXO
JOSE
ABRAHAM
MATHEW
CHREKETO
SOSO

Administration Schemas

Information

No object selected

Result 5

Output

Action Output

#	Time	Action	Message
✓ 15	11:04:42	INSERT INTO Reward (employee_ref_id, date_reward, amount) VALUES (1, '2019-05-11', 1000), (2, '2019-02-...	4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0
✓ 16	11:08:08	SELECT * FROM Employee LIMIT 0, 1000	7 row(s) returned
✓ 17	11:09:20	SELECT first_name, last_name FROM Employee LIMIT 0, 1000	7 row(s) returned
✓ 18	11:10:24	SELECT first_name AS "Nombre de empleado" FROM Employee LIMIT 0, 1000	7 row(s) returned
✓ 19	11:11:32	SELECT LOWER(last_name) AS "Apellido en minúsculas" FROM Employee LIMIT 0, 1000	7 row(s) returned
✓ 20	11:12:41	SELECT UPPER(last_name) AS "Apellido en mayúsculas" FROM Employee LIMIT 0, 1000	7 row(s) returned

Object Info Session

Figure 6: .Obtener todos los valores de la columna “Last_name” en mayúsculas.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- flotilla
- sakila
- sys
- world

Query 1 x

Limit to 1000 rows

```

33
34 • SELECT * FROM Employee;
35 • SELECT first_name, last_name FROM Employee;
36 • SELECT first_name AS "Nombre de empleado" FROM Employee;
37 • SELECT LOWER(last_name) AS "Apellido en minúsculas" FROM Employee;
38 • SELECT UPPER(last_name) AS "Apellido en mayúsculas" FROM Employee;
39 • SELECT DISTINCT department FROM Employee;

```

Result Grid

department
Finance
IT
Banking
Insurance

Administration Schemas

Information

No object selected

Employee 6 x

Output

Action Output

#	Time	Action	Message
✓ 16	11:08:08	SELECT * FROM Employee LIMIT 0, 1000	7 row(s) returned
✓ 17	11:09:20	SELECT first_name, last_name FROM Employee LIMIT 0, 1000	7 row(s) returned
✓ 18	11:10:24	SELECT first_name AS "Nombre de empleado" FROM Employee LIMIT 0, 1000	7 row(s) returned
✓ 19	11:11:32	SELECT LOWER(last_name) AS "Apellido en minúsculas" FROM Employee LIMIT 0, 1000	7 row(s) returned
✓ 20	11:12:41	SELECT UPPER(last_name) AS "Apellido en mayúsculas" FROM Employee LIMIT 0, 1000	7 row(s) returned
✓ 21	11:13:40	SELECT DISTINCT department FROM Employee LIMIT 0, 1000	4 row(s) returned

Object Info Session

Figure 7: Obtener los nombre únicos de la columna “Departament”.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- flotilla
- sakila
- sys
- world

Query 1 x

```

34 SELECT * FROM Employee;
35 SELECT first_name, last_name FROM Employee;
36 SELECT first_name AS "Nombre de empleado" FROM Employee;
37 SELECT LOWER(last_name) AS "Apellido en minúsculas" FROM Employee;
38 SELECT UPPER(last_name) AS "Apellido en mayúsculas" FROM Employee;
39 SELECT DISTINCT department FROM Employee;
40 SELECT SUBSTRING(first_name, 1, 4) AS "Primeros 4 caracteres" FROM Employee;
--

```

Result Grid

Primeros 4 caracteres
Bob
Jerr
Phil
John
Mich
Alex
Yoha

Administration Schemas

Information

No object selected

Result 7 x

Output

Action Output

#	Time	Action	Message
✓ 17	11:09:20	SELECT first_name, last_name FROM Employee LIMIT 0, 1000	7 row(s) returned
✓ 18	11:10:24	SELECT first_name AS "Nombre de empleado" FROM Employee LIMIT 0, 1000	7 row(s) returned
✓ 19	11:11:32	SELECT LOWER(last_name) AS "Apellido en minúsculas" FROM Employee LIMIT 0, 1000	7 row(s) returned
✓ 20	11:12:41	SELECT UPPER(last_name) AS "Apellido en mayúsculas" FROM Employee LIMIT 0, 1000	7 row(s) returned
✓ 21	11:13:40	SELECT DISTINCT department FROM Employee LIMIT 0, 1000	4 row(s) returned
✓ 22	11:15:18	SELECT SUBSTRING(first_name, 1, 4) AS "Primeros 4 caracteres" FROM Employee LIMIT 0, 1000	7 row(s) returned

Object Info Session

Figure 8: Obtener los primeros 4 caracteres de todos los valores de la columna "First_name".

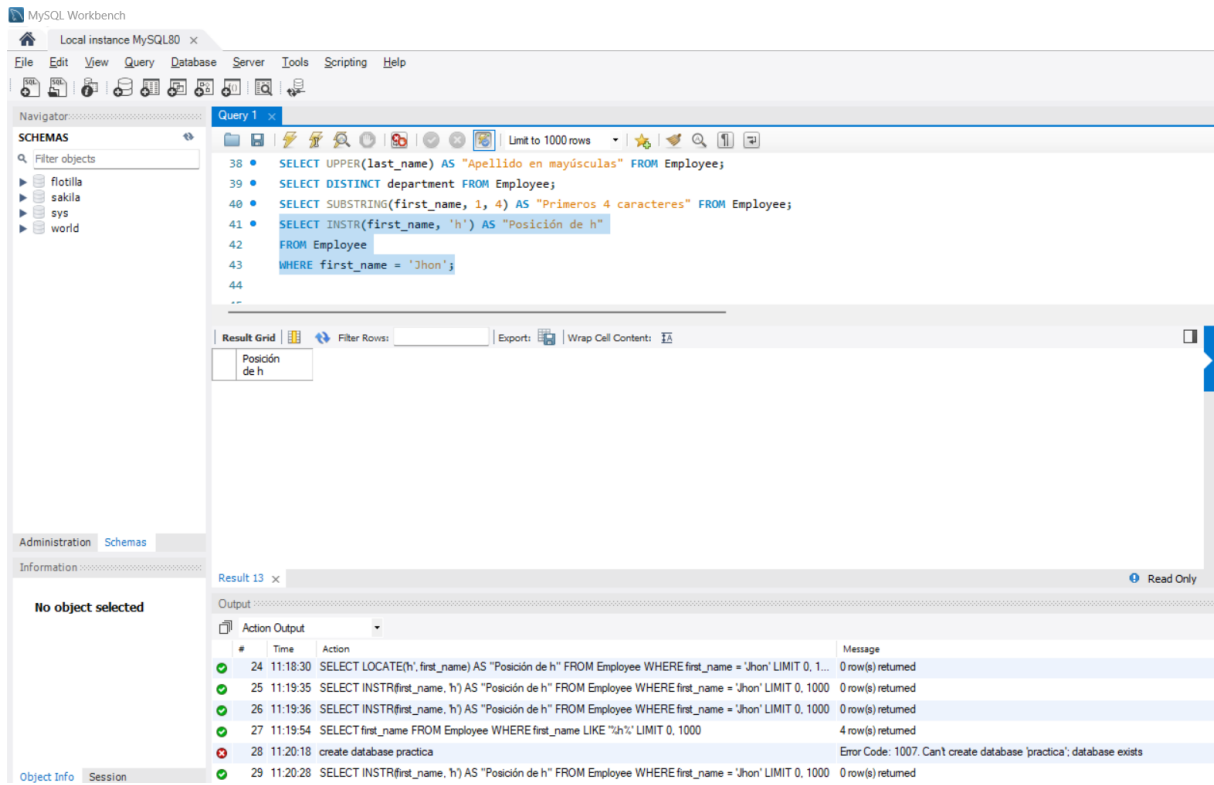


Figure 9: Obtener la posición de la letra “h” en el nombre del empleado con First_name = “Jhon”

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

- flotilla
- sakila
- sys
- world

Query 1 x

Limit to 1000 rows

```
36 • SELECT first_name AS "Nombre de empleado" FROM Employee;
37 • SELECT LOWER(last_name) AS "Apellido en minúsculas" FROM Employee;
38 • SELECT UPPER(last_name) AS "Apellido en mayúsculas" FROM Employee;
39 • SELECT DISTINCT department FROM Employee;
40 • SELECT SUBSTRING(first_name, 1, 4) AS "Primeros 4 caracteres" FROM Employee;
41 • SELECT INSTR(first_name, 'h') AS "Posición de h" FROM Employee WHERE first_name = 'Jhon';
42 • SELECT RTRIM(first_name) AS "Nombre sin espacios a la derecha" FROM Employee;
```

Result Grid

	Nombre sin espacios a la derecha
▶	Bob
	Jerry
	Philip
	John
	Michael
	Alex
	Yohan

Administration Schemas

Information

No object selected

Result 14 x

Output

#	Time	Action	Message
✓	25	11:19:35	SELECT INSTR(first_name, 'h') AS "Posición de h" FROM Employee WHERE first_name = 'Jhon' LIMIT 0, 1000 0 row(s) returned
✓	26	11:19:36	SELECT INSTR(first_name, 'h') AS "Posición de h" FROM Employee WHERE first_name = 'Jhon' LIMIT 0, 1000 0 row(s) returned
✓	27	11:19:54	SELECT first_name FROM Employee WHERE first_name LIKE "h%" LIMIT 0, 1000 4 row(s) returned
✗	28	11:20:18	create database practica Error Code: 1007. Can't create database 'practica'; database exists
✓	29	11:20:28	SELECT INSTR(first_name, 'h') AS "Posición de h" FROM Employee WHERE first_name = 'Jhon' LIMIT 0, 1000 0 row(s) returned
✓	30	11:21:35	SELECT RTRIM(first_name) AS "Nombre sin espacios a la derecha" FROM Employee LIMIT 0, 1000 7 row(s) returned

Object Info Session

Figure 10: Obtener todos los valores de la columna “First_name” después de remover los espacios en blanco de la derecha.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- flotilla
- sakila
- sys
- world

Query 1 x

Limit to 1000 rows

```

37 • SELECT LOWER(last_name) AS "Apellido en minúsculas" FROM Employee;
38 • SELECT UPPER(last_name) AS "Apellido en mayúsculas" FROM Employee;
39 • SELECT DISTINCT department FROM Employee;
40 • SELECT SUBSTRING(first_name, 1, 4) AS "Primeros 4 caracteres" FROM Employee;
41 • SELECT INSTR(first_name, 'h') AS "Posición de h" FROM Employee WHERE first_name = 'Jhon';
42 • SELECT RTRIM(first_name) AS "Nombre sin espacios a la derecha" FROM Employee;
43 • SELECT LTRIM(first_name) AS "Nombre sin espacios a la izquierda" FROM Employee;

```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	Nombre sin espacios a la izquierda
▶	Bob
	Jerry
	Philip
	John
	Michael
	Alex
	Yohan

Administration Schemas

Information

No object selected

Result 15 x

Output

Action Output

#	Time	Action	Message
✓ 26	11:19:36	SELECT INSTR(first_name, 'h') AS "Posición de h" FROM Employee WHERE first_name = 'Jhon' LIMIT 0, 1000	0 row(s) returned
✓ 27	11:19:54	SELECT first_name FROM Employee WHERE first_name LIKE "jh%" LIMIT 0, 1000	4 row(s) returned
✗ 28	11:20:18	create database practica	Error Code: 1007. Can't create database 'practica';
✓ 29	11:20:28	SELECT INSTR(first_name, 'h') AS "Posición de h" FROM Employee WHERE first_name = 'Jhon' LIMIT 0, 1000	0 row(s) returned
✓ 30	11:21:35	SELECT RTRIM(first_name) AS "Nombre sin espacios a la derecha" FROM Employee LIMIT 0, 1000	7 row(s) returned
✓ 31	11:22:48	SELECT LTRIM(first_name) AS "Nombre sin espacios a la izquierda" FROM Employee LIMIT 0, 1000	7 row(s) returned

Object Info Session

Figure 11: Obtener todos los valores de la columna “First_name” después de remover los espacios en blanco de la izquierda.

Sentencias SQL

Presentar las sentencias para crear la base de datos y tablas. Además incluir las sentencias para insertar registros.

En el Listado 1 se presenta la sentencia SQL para crear la base de datos competencia.

Listing 1: Crear base de Practical.

```
CREATE DATABASE Practica.

use practica;

CREATE TABLE Employee (
    employee-id INT PRIMARY KEY,
    first_name VARCHAR(255),
    last_name VARCHAR(255),
    salary INT,
    joining_date VARCHAR(255),
    department VARCHAR(255)
);

INSERT INTO Employee (employee-id, first-name, last-name, salary, joining-date, department)
(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, '2019-02-28', 'Finance'),
(6, 'Alex', 'Chreketo', 4000000, '2019-05-10', 'IT'),
(7, 'Yohan', 'Soso', 1230000, '2019-06-20', 'Banking');

CREATE TABLE Reward (
    employee-ref-id INT,
    date-reward varchar(255),
    amount INT,
    CONSTRAINT fk-employee FOREIGN KEY (employee-ref-id) REFERENCES Employee(employee-id)
);

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),
(1, '2019-06-20', 8000);

SELECT * FROM Employee;

SELECT first-name, last-name FROM Employee;
SELECT first-name AS "Nombre-de-empleado" FROM Employee;
SELECT LOWER(last-name) AS "Apellido-en-min sculas" FROM Employee;
SELECT UPPER(last-name) AS "Apellido-en-may sculas" FROM Employee;
SELECT DISTINCT department FROM Employee;
SELECT SUBSTRING(first-name, 1, 4) AS "Primeros-4-caracteres" FROM Employee;
SELECT INSTR(first-name, 'h') AS "Posici n-de-h" FROM Employee WHERE first_name = 'Jhon';
SELECT RTRIM(first_name) AS "Nombre-sin-espacios-a-la-derecha" FROM Employee;
SELECT LTRIM(first_name) AS "Nombre-sin-espacios-a-la-izquierda" FROM Employee;
```

4. Conclusiones

En esta práctica, hemos explorado y aplicado diversas operaciones y consultas en SQL, utilizando dos tablas principales: **Employee** y **Reward**. A lo largo de la práctica, hemos aprendido a:

1. **Crear Tablas:** Utilizamos la sentencia `CREATE TABLE` para definir la estructura de las tablas **Employee** y **Reward**, especificando tipos de datos y estableciendo claves primarias y foráneas para mantener la integridad referencial.
2. **Insertar Datos:** Insertamos registros en las tablas utilizando la sentencia `INSERT INTO`, lo que nos permitió poblar las tablas con datos de ejemplo.
3. **Realizar Consultas:** Ejecutamos diversas consultas SQL para obtener información específica de las tablas. Estas consultas incluyeron la selección de todos los registros, la obtención de valores únicos, el uso de alias para columnas, y la manipulación de cadenas de texto para mostrar datos en diferentes formatos.
4. **Manipulación de Cadenas:** Utilizamos funciones como `RTRIM()`, `LTRIM()`, `UPPER()`, `LOWER()`, `SUBSTRING()`, y `POSITION()` para trabajar con datos de texto, demostrando cómo SQL puede manejar y transformar datos de manera eficiente.

Esta práctica nos ha permitido comprender mejor cómo interactuar con bases de datos SQL, realizar consultas complejas y asegurar la integridad de los datos mediante el uso de claves foráneas. Los conocimientos adquiridos son fundamentales para el desarrollo de aplicaciones que requieren el manejo eficiente de datos. Además, hemos visto cómo SQL puede ser una herramienta poderosa y flexible para gestionar y analizar información en un entorno de base de datos relacional.

En resumen, esta práctica ha sido una excelente oportunidad para consolidar nuestros conocimientos de SQL y prepararnos para enfrentar desafíos más avanzados en el manejo de bases de datos.

Referencias Bibliográficas

References

- [1] arzibyte. (2024, 19 julio). *Agregar columna llave foránea (fk) en MySQL*. Parzibyte's Blog. <https://parzibyte.me/blog/2019/11/12/agregar-columna-llave-foranea-fk-mysql/>
- [2] uintana, A. (s. f.). *Insertar y Consultar datos en MySQL / Mariadb*. Styde.net. <https://styde.net/insertar-y-consultar-datos-en-mysql-mariadb/#:~:text=El%20comando%20SELECT%20va%20seguido,la%20tabla%20que%20deseas%20consultar.>
- [3] iñolo, A. C. (2010, 19 octubre). *MySQL - Sensibilidad a mayúsculas/minúsculas de los nombres de las tablas - Adictos al trabajo*. Adictos Al Trabajo. <https://adictosaltrabajo.com/2010/10/19/mysql-table-names-uppercase/>