

# REPORTE DE PRÁCTICA NO. 3

## 1.4. Práctica. Álgebra relacional y SQL (2)

ALUMNO:

KEVIN AXEL CHAVEZ QUIROZ



# 1. Introducción

Incluir una descripción de los que trata la práctica.

En esta práctica, hemos explorado varias operaciones fundamentales de SQL (Structured Query Language) para gestionar y manipular datos en una base de datos relacional. A continuación, se describen las principales actividades realizadas:

## 1. Creación de Tablas y Base de Datos:

- Definimos una base de datos llamada **Empresa** y creamos dos tablas: **Employee** y **Reward**.
- La tabla **Employee** contiene información sobre los empleados, como su ID, nombre, apellido, salario, fecha de ingreso y departamento.
- La tabla **Reward** registra las recompensas otorgadas a los empleados, incluyendo el ID de referencia del empleado, la fecha de la recompensa y el monto.

## 2. Inserción de Datos: Insertamos datos en ambas tablas utilizando sentencias `INSERT INTO`.

## 3. Consultas Básicas:

- Realizamos consultas para obtener información específica de las tablas, como la longitud de los nombres de los empleados y la concatenación de nombres y apellidos.
- Utilizamos funciones como `LENGTH` para calcular la longitud de los nombres y `REPLACE` para modificar los nombres reemplazando caracteres específicos.

## 4. Filtrado y Ordenación de Datos:

- Aplicamos cláusulas `WHERE` para filtrar empleados por nombre.
- Ordenamos los resultados de las consultas utilizando `ORDER BY` en orden ascendente y descendente, tanto por nombre como por salario.

## 5. Extracción de Componentes de Fechas: Extraímos el año, mes y día de la columna `Joining_date` utilizando funciones como `YEAR`, `MONTH` y `DAY`.

Estas operaciones son esenciales para cualquier trabajo con bases de datos, ya que permiten gestionar y analizar datos de manera eficiente. La práctica te ha proporcionado una base sólida para realizar consultas más complejas y gestionar bases de datos en proyectos futuros.

## 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

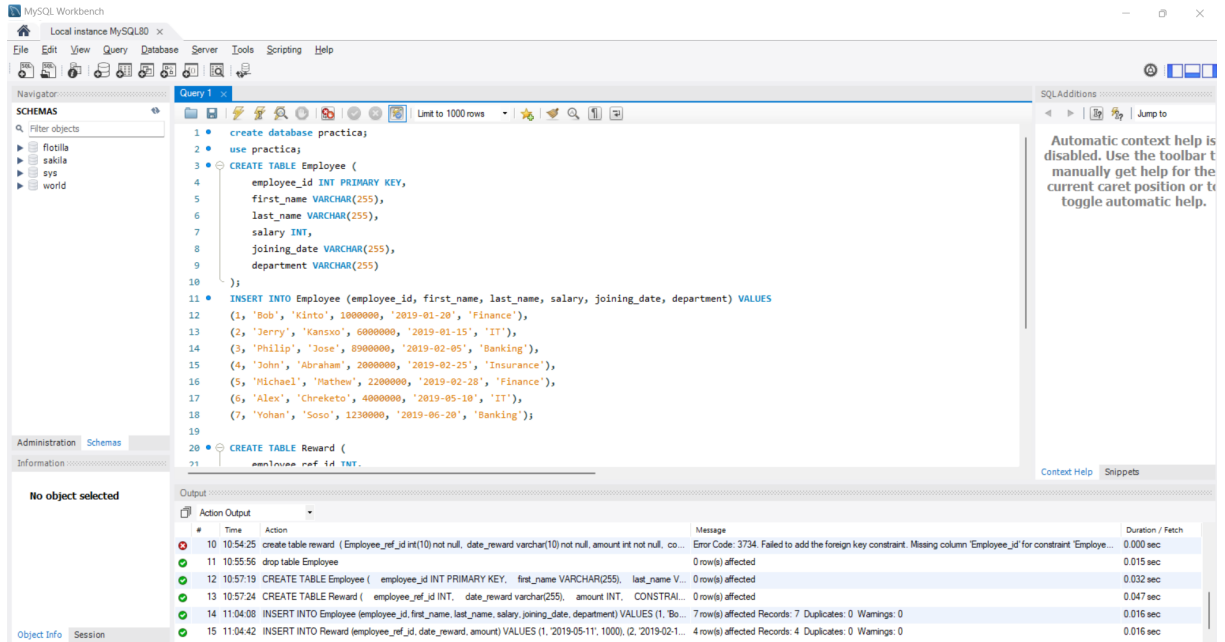


Figure 1: creacion de las tablas.

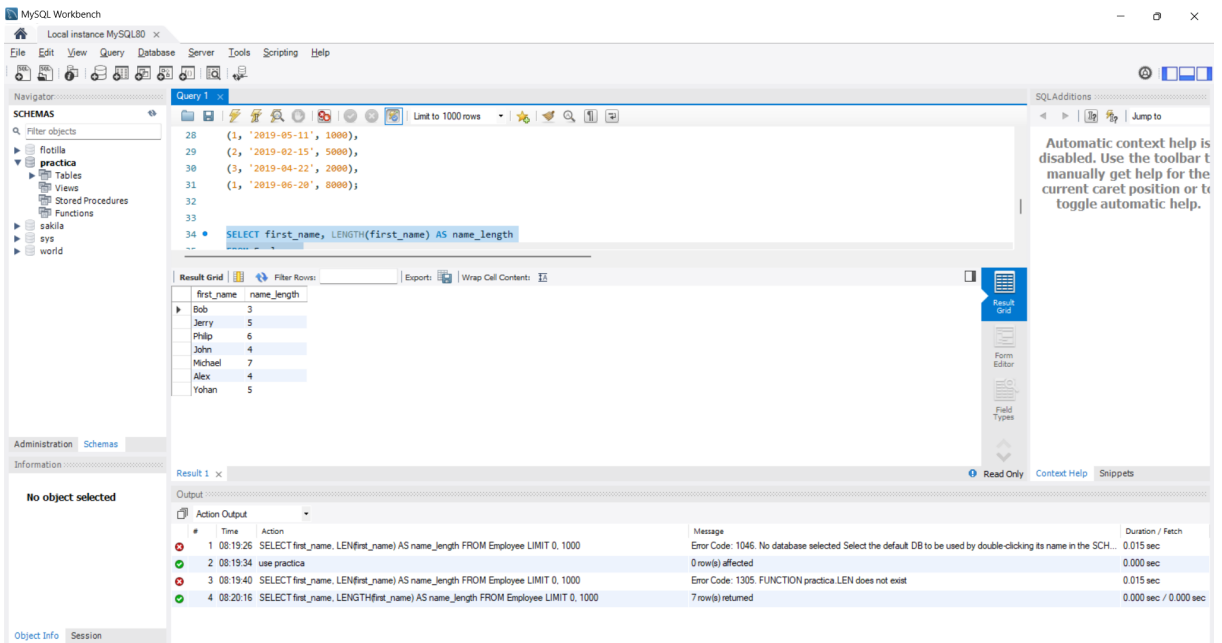


Figure 2: Obtener el tamaño del texto en todos los valores de la columna “First\_name”.

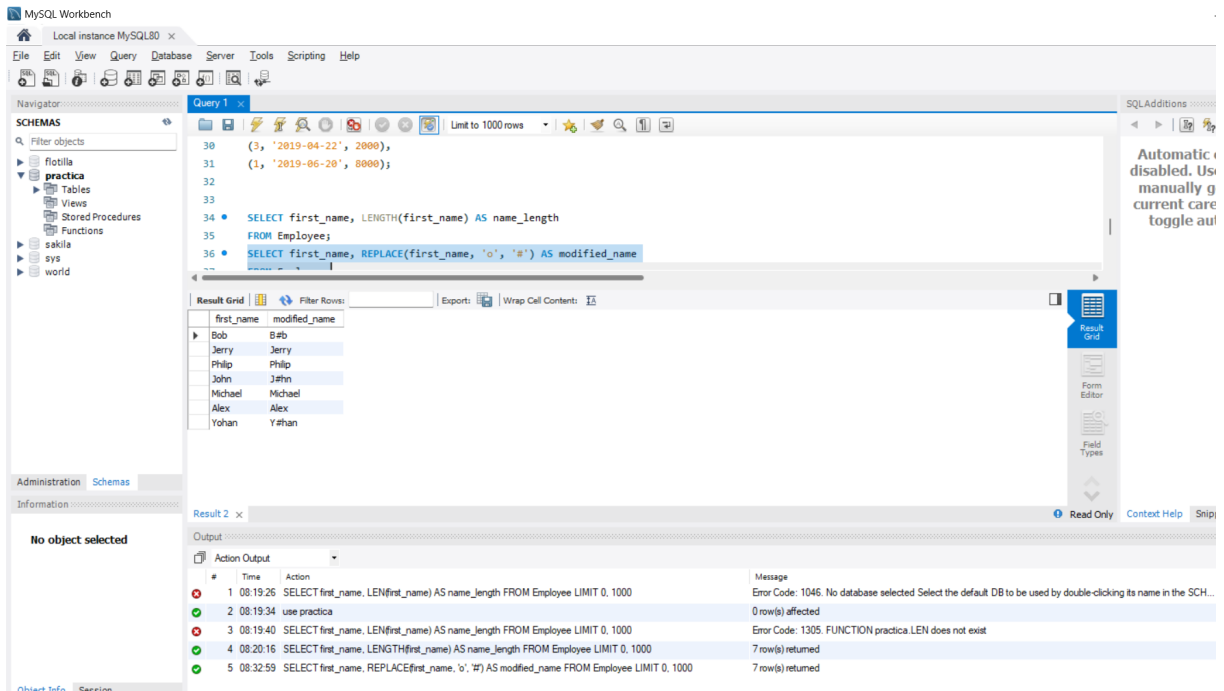


Figure 3: Obtener el nombre de todos los empleados después de reemplazar 'o' con '#'.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

flotilla

practica

Tables

Views

Stored Procedures

Functions

sakila

sys

world

Query 1 x

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 first_name, LENGTH(first_name) AS name_length
35 FROM Employee;
36 SELECT first_name, REPLACE(first_name, 'o', '#') AS modified_name
37 FROM Employee;
38 SELECT CONCAT(first_name, '_', last_name) AS full_name
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content: 12

full\_name

Bob\_Kinto

Jerry\_Kansio

Philo\_Jose

John\_Abraham

Michael\_Mathew

Alex\_Chreketo

Yohan\_Soso

Administration Schemas

Information

No object selected

Result 3 x

Read Only Context Help Snip

Output

Action Output

#	Time	Action	Message
1	08:19:26	SELECT first_name, LEN(first_name) AS name_length FROM Employee LIMIT 0, 1000	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...
2	08:19:34	use practica	0 row(s) affected
3	08:19:40	SELECT first_name, LEN(first_name) AS name_length FROM Employee LIMIT 0, 1000	Error Code: 1305. FUNCTION practica.LEN does not exist
4	08:20:16	SELECT first_name, LENGTH(first_name) AS name_length FROM Employee LIMIT 0, 1000	7 row(s) returned
5	08:32:59	SELECT first_name, REPLACE(first_name, 'o', '#') AS modified_name FROM Employee LIMIT 0, 1000	7 row(s) returned
6	08:35:09	SELECT CONCAT(first_name, '_', last_name) AS full_name FROM Employee LIMIT 0, 1000	7 row(s) returned

Object Info Session

Figure 4: Obtener el nombre y apellido de todos los empleados en una sola columna separados por “\_”

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHMAS

- flotilla
- practica
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sakila
- sys
- world

Query 1 x

```
39 FROM Employee;
40
41 SELECT
42     first_name,
43     last_name,
44     YEAR(Joining_date) AS year,
45     MONTH(Joining_date) AS month,
46     DAY(Joining_date) AS day
47 FROM Employee;
48
```

Result Grid

	first_name	last_name	year	month	day
▶	Bob	Kinto	2019	1	20
	Jerry	Kansoxo	2019	1	15
	Philip	Jose	2019	2	5
	John	Abraham	2019	2	25
	Michael	Mathew	2019	2	28
	Alex	Chrekeeto	2019	5	10
	Yohan	Soso	2019	6	20

Administration Schemas

Information

No object selected

Result 1 x

Output

Action Output

#	Time	Action	Message
1	08:37:13	SELECT first_name, last_name, YEAR(joining_date) AS year, MONTH(joining_date) AS month, D...	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...
2	08:37:18	use practica	0 row(s) affected
3	08:37:25	SELECT first_name, last_name, YEAR(joining_date) AS year, MONTH(joining_date) AS month, D...	7 row(s) returned

Object Info Session

SQLAdditions

Automatic disabled. U manually t current car toggle at

Figure 5: Obtener el año, mes y día de la columna “Joining\_date”.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

flotilla

practica

Tables

Views

Stored Procedures

Functions

sakila

sys

world

Query 1

Limit to 1000 rows

```
41 SELECT
42     first_name,
43     last_name,
44     YEAR(joining_date) AS year,
45     MONTH(joining_date) AS month,
46     DAY(joining_date) AS day
47 FROM Employee;
48
49 SELECT *
```

Result Grid

	employee_id	first_name	last_name	salary	joining_date	department
6	Alex	Chelato	4000000	2019-05-10	IT	
1	Bob	Kinto	1000000	2019-01-20	Finance	
2	Jerry	Kanexo	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	

Administration Schemas

Information

No object selected

Employee 2 x

Output

#	Time	Action	Message
1	08:37:13	SELECT first_name, last_name, YEAR(joining_date) AS year, MONTH(joining_date) AS month, DAY(joining_date) AS day FROM Employee;	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...
2	08:37:18	use practica	0 row(s) affected
3	08:37:25	SELECT first_name, last_name, YEAR(joining_date) AS year, MONTH(joining_date) AS month, DAY(joining_date) AS day FROM Employee;	7 row(s) returned
4	08:39:38	SELECT * FROM Employee ORDER BY first_name ASC LIMIT 0, 1000	7 row(s) returned

Object Info Session

Automatic disabled. Use manually current care toggle au

Figure 6: .Obtener todos los empleados en orden ascendente por nombre.



MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

flotilla

practica

Tables

Views

Stored Procedures

Functions

sakila

sys

world

Query 1 x

```
44 YEAR(joining_date) AS year,
45 MONTH(joining_date) AS month,
46 DAY(joining_date) AS day
47 FROM Employee;
48
49 • SELECT *
50 FROM Employee
51 ORDER BY first_name ASC;
52
53 • SELECT *
```

Result Grid

	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	

Administration Schemas

Information

No object selected

Employee 3 x

Output

Action Output

#	Time	Action	Message
1	08:37:13	SELECT first_name, last_name, YEAR(joining_date) AS year, MONTH(joining_date) AS month, D...	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...
2	08:37:18	use practica	0 row(s) affected
3	08:37:25	SELECT first_name, last_name, YEAR(joining_date) AS year, MONTH(joining_date) AS month, D...	7 row(s) returned
4	08:39:38	SELECT * FROM Employee ORDER BY first_name ASC LIMIT 0, 1000	7 row(s) returned
5	08:40:35	SELECT * FROM Employee ORDER BY first_name DESC LIMIT 0, 1000	7 row(s) returned

Object Info Session

SQLAdditions

Automatic c disabled. Use manually ge current care! toggle aut

Figure 7: Obtener todos los empleados en orden descendente por nombre.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHMAS

- flotilla
- practica
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sakila
- sys
- world

Query 1 x

Limit to 1000 rows

```

51 ORDER BY first_name ASC;
52
53 • SELECT *
54 FROM Employee
55 ORDER BY first_name DESC;
56
57 • SELECT *
58 FROM Employee
59 ORDER BY first_name ASC, Salary DESC;
60
--

```

Result Grid

employee_id	first_name	last_name	salary	joining_date	department
6	Alex	Chynketo	4000000	2019-05-10	IT
1	Bob	Kinto	1000000	2019-01-20	Finance
2	Jerry	Kansvo	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

Administration Schemas

Information

No object selected

Employee 4 x

Output

Action Output

#	Time	Action	Message
1	08:37:13	SELECT first_name, last_name, YEAR(joining_date) AS year, MONTH(joining_date) AS month, D...	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...
2	08:37:18	use practica	0 row(s) affected
3	08:37:25	SELECT first_name, last_name, YEAR(joining_date) AS year, MONTH(joining_date) AS month, D...	7 row(s) returned
4	08:39:38	SELECT * FROM Employee ORDER BY first_name ASC LIMIT 0, 1000	7 row(s) returned
5	08:40:35	SELECT * FROM Employee ORDER BY first_name DESC LIMIT 0, 1000	7 row(s) returned
6	08:42:27	SELECT * FROM Employee ORDER BY first_name ASC, Salary DESC LIMIT 0, 1000	7 row(s) returned

Object Info Session

SQLAdditions

Automatic disabled. Use manually (current can toggle au

Figure 8: Obtener todos los empleados en orden ascendente por nombre y en orden descendente por salario.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- flotilla
- practica
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sakila
- sys
- world

Query 1 x

Limit to 1000 rows

```
52
53 • SELECT *
54 FROM Employee
55 ORDER BY first_name DESC;
56
57 • SELECT *
58 FROM Employee
59 ORDER BY first_name ASC, Salary DESC;
60
61 • SELECT *
```

Result Grid

	employee_id	first_name	last_name	salary	joining_date	department
▶ 1	Bob	Kinto	1000000	2019-01-20	Finance	
•	NULL	NULL	NULL	NULL	NULL	NULL

Administration Schemas

Information

No object selected

Employee 5 x

Output

Action Output

#	Time	Action	Message
✓ 2	08:37:18	use practica	0 row(s) affected
✓ 3	08:37:25	SELECT first_name, last_name, YEAR(joining_date) AS year, MONTH(joining_date) AS month, ...	7 row(s) returned
✓ 4	08:39:38	SELECT * FROM Employee ORDER BY first_name ASC LIMIT 0, 1000	7 row(s) returned
✓ 5	08:40:35	SELECT * FROM Employee ORDER BY first_name DESC LIMIT 0, 1000	7 row(s) returned
✓ 6	08:42:27	SELECT * FROM Employee ORDER BY first_name ASC, Salary DESC LIMIT 0, 1000	7 row(s) returned
✓ 7	08:44:47	SELECT * FROM Employee WHERE first_name = 'Bob' LIMIT 0, 1000	1 row(s) returned

Object Info Session

Figure 9: Obtener todos los empleados con el nombre “Bob”.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

Flotilla

practica

Tables

Views

Stored Procedures

Functions

sakila

sys

world

Query 1

Limit to 1000 rows

```
59 ORDER BY first_name ASC, Salary DESC;
60
61 SELECT *
62 FROM Employee
63 WHERE first_name = 'Bob';
64
65 SELECT *
66 FROM Employee
67 WHERE first_name IN ('Bob', 'Alex');
68
```

Result Grid

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

Administration Schemas

Information

No object selected

Employee 6 x

Output

Action Output

#	Time	Action	Message
4	08:39:38	SELECT * FROM Employee ORDER BY first_name ASC LIMIT 0, 1000	7 row(s) returned
5	08:40:35	SELECT * FROM Employee ORDER BY first_name DESC LIMIT 0, 1000	7 row(s) returned
6	08:42:27	SELECT * FROM Employee ORDER BY first_name ASC, Salary DESC LIMIT 0, 1000	7 row(s) returned
7	08:44:47	SELECT * FROM Employee WHERE first_name = 'Bob' LIMIT 0, 1000	1 row(s) returned
8	08:46:24	FROM Employee WHERE first_name IN ('Bob', 'Alex')	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...
9	08:46:29	SELECT * FROM Employee WHERE first_name IN ('Bob', 'Alex') LIMIT 0, 1000	2 row(s) returned

Object Info Session

SQLAdditions

Automati disabled. t manually current ca toggle a

Figure 10: Obtener todos los empleados con el nombre “Bob” o “Alex”

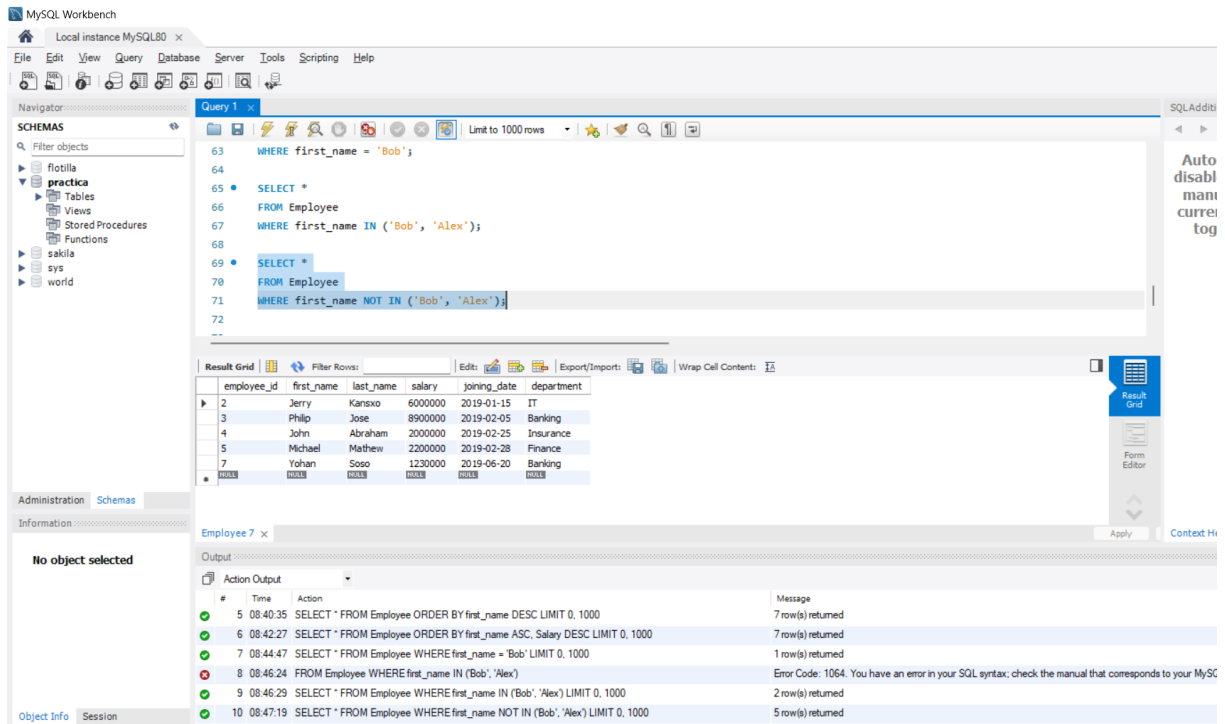


Figure 11: Obtener todos los empleados que no tengan el nombre “Bob” o “Alex”.

### ¿Qué es una inyección SQL?

La inyección SQL (SQL Injection) es una vulnerabilidad de seguridad que permite a un atacante manipular las consultas SQL que una aplicación envía a su base de datos.

Esta técnica se utiliza para insertar código malicioso en los campos de entrada de una aplicación web, con el objetivo de acceder a datos sensibles, modificar información o incluso tomar el control de la base de datos.

## 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 Practica1.

```
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 first_name, LENGTH(first_name) AS name_length
FROM Employee;

SELECT first_name, REPLACE(first_name, 'o', '#') AS modified_name
FROM Employee;

SELECT CONCAT(first_name, '_', last_name) AS full_name
FROM Employee;

SELECT
    first_name ,
```

```
    last_name ,  
    YEAR(Joining_date) AS year ,  
    MONTH(Joining_date) AS month ,  
    DAY(Joining_date) AS day  
FROM Employee;
```

```
SELECT *  
FROM Employee  
ORDER BY first_name ASC;
```

```
SELECT *  
FROM Employee  
ORDER BY first_name DESC;
```

```
SELECT *  
FROM Employee  
ORDER BY first_name ASC, Salary DESC;
```

```
SELECT *  
FROM Employee  
WHERE first_name = 'Bob';
```

```
SELECT *  
FROM Employee  
WHERE first_name IN ( 'Bob', 'Alex' );
```

```
SELECT *  
FROM Employee  
WHERE first_name NOT IN ( 'Bob', 'Alex' );
```

## 4. Conclusiones

En esta práctica, hemos explorado y aplicado diversas operaciones fundamentales de SQL para la gestión y manipulación de datos en una base de datos relacional. A través de la creación de tablas, la inserción de datos y la ejecución de consultas, hemos adquirido una comprensión más profunda de cómo interactuar con bases de datos de manera eficiente.

Principales aprendizajes:

1. **Creación y gestión de tablas:** Aprendimos a definir estructuras de tablas y a insertar datos en ellas, lo cual es esencial para organizar y almacenar información de manera estructurada.
2. **Consultas básicas y avanzadas:** Realizamos consultas para extraer y manipular datos, utilizando funciones como `LENGTH`, `REPLACE`, `CONCAT`, y operadores de filtrado y ordenación. Estas habilidades son cruciales para obtener información específica y relevante de la base de datos.
3. **Filtrado y ordenación de datos:** Aplicamos técnicas para filtrar y ordenar datos, lo que nos permite presentar la información de manera clara y ordenada según criterios específicos.
4. **Extracción de componentes de fechas:** Utilizamos funciones para extraer el año, mes y día de las fechas, lo que es útil para análisis temporales y reportes.
5. **Seguridad en bases de datos:** Discutimos la importancia de prevenir vulnerabilidades como la inyección SQL, subrayando la necesidad de validar y sanitizar las entradas de usuario.

Esta práctica nos ha proporcionado una base sólida en el uso de SQL, preparándonos para enfrentar desafíos más complejos en el manejo de bases de datos. La capacidad de realizar consultas eficientes y seguras es una habilidad valiosa en el campo de la informática y será fundamental en futuros proyectos y aplicaciones.



## 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] ñoio, 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/>
- [4] tackHowTo. (n.d.). *MySQL Practice Exercises with Solutions - Part 2*. StackHowTo. Recuperado de <https://stackhowto.com/mysql-practice-exercises-with-solutions-part-2/>