SQL LAB-3
RDBMS, MYSQL

NAME: Seelam Vijayalakshmi

ID: AF0366867

QUESTIONS

Lab 2.Database Schema:

Consider a simple database with one tables: Employee

Employee Table:

• Columns: emp_id (Primary Key), first_name, last_name, age, email

Task 1: Insert Data

Write an SQL INSERT statement to insert data into the Employee table.

Task 2: Retrieving Data

Write an SQL SELECT statement to retrieve the first_name and last_name of all employees from the Employee table.

Task 3: Filtering Data

Write an SQL SELECT statement to retrieve the first_name, last_name, and age of employees who are older than 30 years.

Task 4: Updating Data

Write an SQL UPDATE statement to increase the age of employees by 1 year for all employees older than 25.

ChatGPT Exercise

Using ChatGPT generates SQL queries to update the Employee salary.

Scenario:

Due to a pricing adjustment, the company decided to increase the salary of all employees by 10%. Create an SQL update query to apply this change selectively to employees with a specific job title, say 'Manager'

Consider a simple database with one tables: Employee

Employee Table:

• Columns: emp_id (Primary Key), first_name, last_name, age, email

Code:-

```
mysql> CREATE TABLE Employee (
    ->     emp_id INT PRIMARY KEY, -- Unique identifier for each employee
    ->     first_name VARCHAR(50), -- First name of the employee
    ->     last_name VARCHAR(50), -- Last name of the employee
    ->     age INT, -- Age of the employee
    ->     email VARCHAR(100) -- Email address of the employee
    -> );
Query OK, 0 rows affected (0.03 sec)
```

Output:-

```
mysql> desc Employee;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| emp_id     | int          | NO   | PRI | NULL    |       |
| first_name | varchar(50)  | YES  |     | NULL    |       |
| last_name  | varchar(50)  | YES  |     | NULL    |       |
| age        | int          | YES  |     | NULL    |       |
| email      | varchar(100) | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

Task 1: Insert Data

Write an SQL INSERT statement to insert data into the Employee table.

Code:-

```
mysql> INSERT INTO Employee (emp_id, first_name, last_name, age, email) -- Inserting data into the Employee table
    -> VALUES
    ->     (1, 'John', 'Doe', 30, 'john.doe@example.com'), -- Inserting data for the first employee
    ->     (2, 'Jane', 'Smith', 25, 'jane.smith@example.com'), -- Inserting data for the second employee
    ->     (3, 'Michael', 'Johnson', 35, 'michael.johnson@example.com'); -- Inserting data for the third employee
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Output:-

```
mysql> select *from Employee;
+--------+------------+-----------+------+---------------------------+
| emp_id | first_name | last_name | age  | email                     |
+--------+------------+-----------+------+---------------------------+
|      1 | John       | Doe       |   30 | john.doe@example.com      |
|      2 | Jane       | Smith     |   25 | jane.smith@example.com    |
|      3 | Michael    | Johnson   |   35 | michael.johnson@example.com |
+--------+------------+-----------+------+---------------------------+
3 rows in set (0.00 sec)
```

Task 2: Retrieving Data

Write an SQL SELECT statement to retrieve the first_name and last_name of all

employees from the Employee table.

```
mysql> SELECT first_name, last_name -- Retrieving first_name and last_name of all employees
    -> FROM Employee;
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| John       | Doe       |
| Jane       | Smith     |
| Michael    | Johnson   |
+------------+-----------+
3 rows in set (0.00 sec)
```

Task 3: Filtering Data

Write an SQL SELECT statement to retrieve the first_name, last_name, and age of employees who are older than 30 years.

```
mysql> SELECT first_name, last_name, age -- Retrieving first_name, last_name, and age of employees older than 30 years
    -> FROM Employee
    -> WHERE age > 30;
+------------+-----------+------+
| first_name | last_name | age  |
+------------+-----------+------+
| Michael    | Johnson   |   35 |
+------------+-----------+------+
1 row in set (0.00 sec)
```

Task 4: Updating Data

Write an SQL UPDATE statement to increase the age of employees by 1 year for all employees older than 25.

Code:-

```
mysql> UPDATE Employee
    -> SET age = age + 1 -- Increasing the age of employees by 1 year
    -> WHERE age > 25;
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2  Changed: 2  Warnings: 0
```

Output:-

```
mysql> Select *from Employee;
+--------+------------+-----------+------+---------------------------+
| emp_id | first_name | last_name | age  | email                     |
+--------+------------+-----------+------+---------------------------+
|      1 | John       | Doe       |   31 | john.doe@example.com      |
|      2 | Jane       | Smith     |   25 | jane.smith@example.com    |
|      3 | Michael    | Johnson   |   36 | michael.johnson@example.com |
+--------+------------+-----------+------+---------------------------+
3 rows in set (0.00 sec)
```

ChatGPT Exercise

Using ChatGPT generates SQL queries to update the Employee salary.

Scenario:

Due to a pricing adjustment, the company decided to increase the salary of all employees by 10%. Create an SQL update query to apply this change selectively to employees with a specific job title, say 'Manager'

Initial Employee Table:-

```
mysql> Select *from Employee;
+--------+------------+-----------+------+-----------------------------+------------+--------+
| emp_id | first_name | last_name | age  | email                       | job_title  | salary |
+--------+------------+-----------+------+-----------------------------+------------+--------+
|      1 | John       | Doe       |   31 | john.doe@example.com        | Manager    |  60000 |
|      2 | Jane       | Smith     |   25 | jane.smith@example.com      | Supervisor |  50000 |
|      3 | Michael    | Johnson   |   36 | michael.johnson@example.com | Analyst    |  55000 |
+--------+------------+-----------+------+-----------------------------+------------+--------+
3 rows in set (0.00 sec)
```

Code:-

```
mysql> -- Increasing the salary of all employees with the job title 'Manager' by 10%
mysql> UPDATE Employee
    -> SET salary = salary * 1.10 -- Apply a 10% increase to the current salary
    -> WHERE job_title = 'Manager';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Output:-

```
mysql> Select *from Employee;
+--------+------------+-----------+------+-----------------------------+------------+--------+
| emp_id | first_name | last_name | age  | email                       | job_title  | salary |
+--------+------------+-----------+------+-----------------------------+------------+--------+
|      1 | John       | Doe       |   31 | john.doe@example.com        | Manager    |  66000 |
|      2 | Jane       | Smith     |   25 | jane.smith@example.com      | Supervisor |  50000 |
|      3 | Michael    | Johnson   |   36 | michael.johnson@example.com | Analyst    |  55000 |
+--------+------------+-----------+------+-----------------------------+------------+--------+
3 rows in set (0.00 sec)
```