

SQL
MODULE

LAB – 4

BY

SUSHRITHA V

Questions

Lab 1: Database Schema:

Consider a simple database with one table: Bank Account

Bank Account Table:

- Columns: account_id (Primary Key), account_holder_name, account_balance

Task 1: Insert Data

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

Task 2: Retrieving Data

Write an SQL SELECT statement to retrieve the account_holder_name and account_balance of all account holders from the Bank Account table.

Task 3: Filtering Data

Write an SQL SELECT statement to retrieve the account_holder_name and account_balance where the account_balance is more than 30,000.

Task 4: Updating Data

Write an SQL UPDATE statement to change the account_balance of the account holder whose ID is 101.

SOLUTIONS:

Lab 1: Database Schema:

Consider a simple database with one table: Bank Account

Bank Account Table:

- Columns: account_id (Primary Key), account_holder_name, account_balance

```
mysql> create database banking;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| banking  |
| information_schema |
| menu     |
| my_file  |
| mybackend|
| mysql    |
| performance_schema |
| student_management_system |
| sys      |
| world    |
+-----+
10 rows in set (0.00 sec)

mysql> use banking;
Database changed
```

```
mysql> Create Table BankAccount(
-> account_id INT PRIMARY KEY,
-> account_holder_name VARCHAR(50) NOT NULL,
-> account_balance DECIMAL(10,2) NOT NULL);
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> describe BankAccount;
```

| Field | Type | Null | Key | Default | Extra |
|---------------------|---------------|------|-----|---------|-------|
| account_id | int | NO | PRI | NULL | |
| account_holder_name | varchar(50) | NO | | NULL | |
| account_balance | decimal(10,2) | NO | | NULL | |

3 rows in set (0.00 sec)

Task 1: Insert Data

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

```
mysql> Insert into Bankaccount
-> values(201,'Ajay',10000),
-> (203,'Deepika',20000),
-> (202,'Sahana',25000),
-> (205,'Suman',50000),
-> (204,'Sushi',70000);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> select *from BankAccount;
+-----+-----+-----+
| account_id | account_holder_name | account_balance |
+-----+-----+-----+
| 201 | Ajay | 10000.00 |
| 202 | Sahana | 25000.00 |
| 203 | Deepika | 20000.00 |
| 204 | Sushi | 70000.00 |
| 205 | Suman | 50000.00 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

Task 2: Retrieving Data

Write an SQL SELECT statement to retrieve the account_holder_name and account_balance of all account holders from the Bank Account table.

```
mysql> select account_holder_name,account_balance from BankAccount;
+-----+-----+
| account_holder_name | account_balance |
+-----+-----+
| Ajay | 10000.00 |
| Sahana | 25000.00 |
| Deepika | 20000.00 |
| Sushi | 70000.00 |
| Suman | 50000.00 |
+-----+-----+
5 rows in set (0.00 sec)
```

Task 3: Filtering Data

Write an SQL SELECT statement to retrieve the account_holder_name and

account_balance where the account_balance is more than 30,000.

```
mysql> select account_holder_name,account_balance from BankAccount
-> where account_balance>30000;
+-----+-----+
| account_holder_name | account_balance |
+-----+-----+
| Sushi               | 70000.00       |
| Suman               | 50000.00       |
+-----+-----+
2 rows in set (0.00 sec)
```

Task 4: Updating Data

Write an SQL UPDATE statement to change the account_balance of the account holder whose ID is 101.

```
mysql> update BankAccount
-> set account_balance=125000
-> where account_id=201;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from BankAccount;
+-----+-----+-----+
| account_id | account_holder_name | account_balance |
+-----+-----+-----+
| 201       | Ajay                | 125000.00      |
| 202       | Sahana              | 25000.00       |
| 203       | Deepika             | 20000.00       |
| 204       | Sushi               | 70000.00       |
| 205       | Suman               | 50000.00       |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

ChatGPT Exercise

Using ChatGPT generates SQL queries of the below problem.

Scenario 1: In an employee database, you want to retrieve information about employees who belong to the "Sales" department and have a salary greater than 50,000.

Scenario 2: An employee has resigned, and you need to remove their record from the

"employees" table. Write an SQL DELETE query for this.

Scenario 3: You want to delete all orders placed before '2022-01-01' that are still in the

'Pending' status. Write an SQL DELETE query for this.

Scenario 4: You want to remove all products from the "Discontinued" category as they

are no longer available. Write an SQL DELETE query for this.

Scenario 5: Employees in the "Sales" department are getting a bonus, and you want to

add 1000 to the bonus column for all employees in that department. Write an SQL UPDATE query for this

Scenario 1: In an employee database, you want to retrieve information about employees who belong to the "Sales" department and have a salary greater than 50,000.

```
mysql> CREATE TABLE Employee (
  ->     employee_id INT PRIMARY KEY,
  ->     first_name VARCHAR(50) NOT NULL,
  ->     last_name VARCHAR(50) NOT NULL,
  ->     department VARCHAR(50) NOT NULL,
  ->     salary DECIMAL(10, 2) NOT NULL,
  ->     hire_date DATE NOT NULL
  -> );
```

Query OK, 0 rows affected (0.05 sec)

```
mysql> desc employee;
```

| Field | Type | Null | Key | Default | Extra |
|-------------|---------------|------|-----|---------|-------|
| employee_id | int | NO | PRI | NULL | |
| first_name | varchar(50) | NO | | NULL | |
| last_name | varchar(50) | NO | | NULL | |
| department | varchar(50) | NO | | NULL | |
| salary | decimal(10,2) | NO | | NULL | |
| hire_date | date | NO | | NULL | |

6 rows in set (0.01 sec)

```
mysql> insert into employee
  -> values(1,'john','doe','sales',60000,'2020-01-15'),
  -> (2,'jane','smith','engineer',75000,'2019-06-01'),
  -> (3,'alice','johnson','sales',55000,'2021-03-20');
```

Query OK, 3 rows affected (0.01 sec)

Records: 3 Duplicates: 0 Warnings: 0

```
mysql> select * from employee;
```

| employee_id | first_name | last_name | department | salary | hire_date |
|-------------|------------|-----------|------------|----------|------------|
| 1 | john | doe | sales | 60000.00 | 2020-01-15 |
| 2 | jane | smith | engineer | 75000.00 | 2019-06-01 |
| 3 | alice | johnson | sales | 55000.00 | 2021-03-20 |

3 rows in set (0.00 sec)

```
mysql> SELECT *
  -> FROM Employee
  -> WHERE department = 'Sales' AND salary > 50000;
```

| employee_id | first_name | last_name | department | salary | hire_date |
|-------------|------------|-----------|------------|----------|------------|
| 1 | john | doe | sales | 60000.00 | 2020-01-15 |
| 3 | alice | johnson | sales | 55000.00 | 2021-03-20 |

2 rows in set (0.00 sec)

Scenario 2: An employee has resigned, and you need to remove their record from the

"employees" table. Write an SQL DELETE query for this.

```
mysql> DELETE FROM Employee
      -> WHERE employee_id = 3;
Query OK, 1 row affected (0.01 sec)

mysql> select * from employee;
```

| employee_id | first_name | last_name | department | salary | hire_date |
|-------------|------------|-----------|------------|----------|------------|
| 1 | john | doe | sales | 60000.00 | 2020-01-15 |
| 2 | jane | smith | engineer | 75000.00 | 2019-06-01 |

Scenario 3: You want to delete all orders placed before '2022-01-01' that are still in the

'Pending' status. Write an SQL DELETE query for this.

DELETE FROM Orders

WHERE order date < '2022-01-01' AND status = 'Pending';

Scenario 4: You want to remove all products from the "Discontinued" category as they

are no longer available. Write an SQL DELETE query for this.

DELETE FROM Products

WHERE category = 'Discontinued';

Scenario 5: Employees in the "Sales" department are getting a bonus, and you want to

add 1000 to the bonus column for all employees in that department. Write an SQL

UPDATE query for this

UPDATE Employee

SET bonus = bonus + 1000

WHERE department = 'Sales';