SQL LAB -4

Submitted by: Thummala Pavani

AF ID : AF0366728

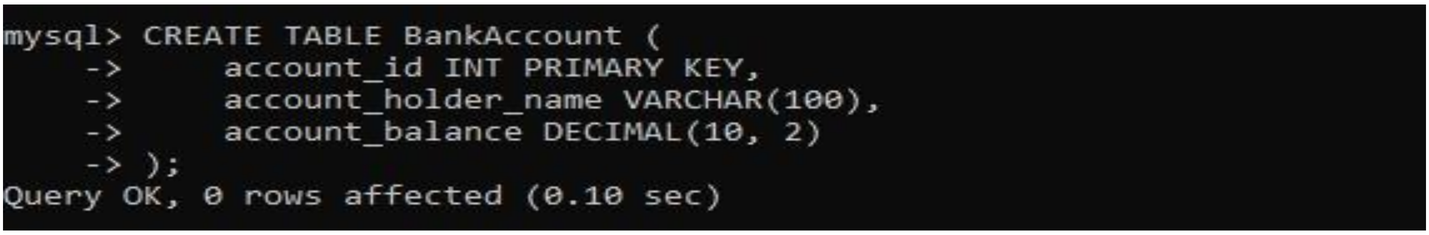
Batch ID : ANP-C7281

Lab 1: Database Schema:

Consider a simple database with one tables: BankAccount BankAccount Table:

● Columns: account_id (Primary Key), account_holder_name,

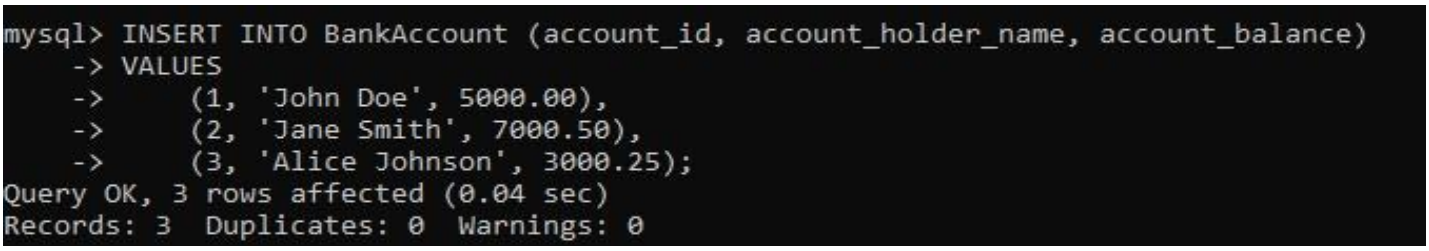
account_balance



```
mysql> CREATE TABLE BankAccount (
    ->     account_id INT PRIMARY KEY,
    ->     account_holder_name VARCHAR(100),
    ->     account_balance DECIMAL(10, 2)
    -> );
Query OK, 0 rows affected (0.10 sec)
```

Task 1: Insert Data

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

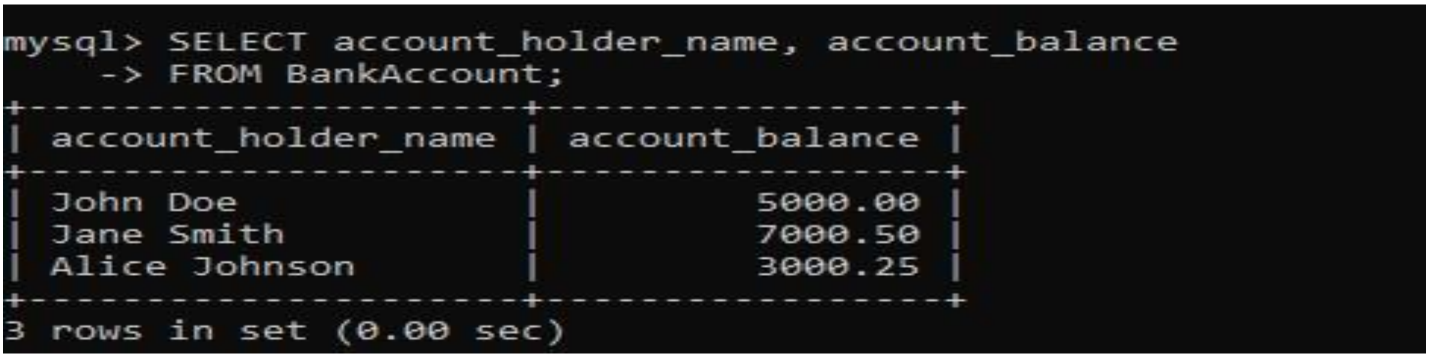


```
mysql> INSERT INTO BankAccount (account_id, account_holder_name, account_balance)
    -> VALUES
    ->     (1, 'John Doe', 5000.00),
    ->     (2, 'Jane Smith', 7000.50),
    ->     (3, 'Alice Johnson', 3000.25);
Query OK, 3 rows affected (0.04 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Task 2: Retrieving Data

Write an SQL SELECT statement to retrieve the account_holder_name and

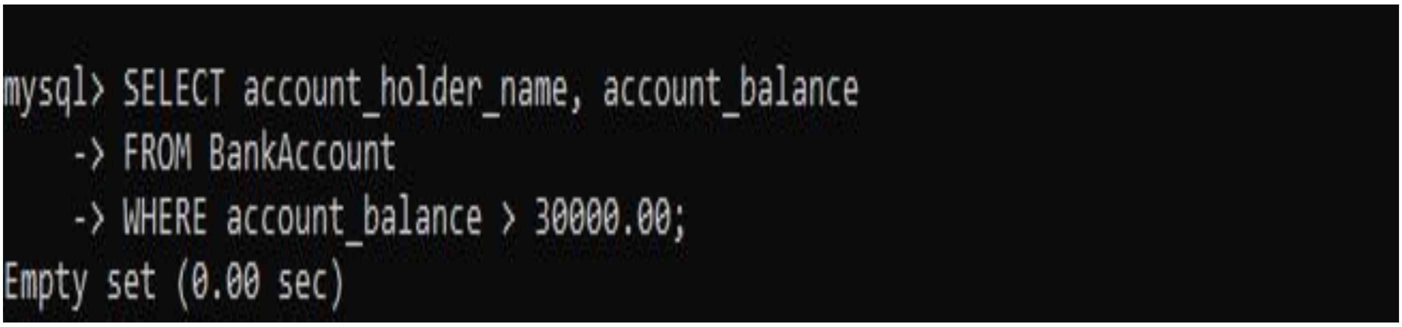
account_balance of all account holders from the BankAccount table.



```
mysql> SELECT account_holder_name, account_balance
    -> FROM BankAccount;
+---------------------+-----------------+
| account_holder_name | account_balance |
+---------------------+-----------------+
| John Doe            |         5000.00 |
| Jane Smith          |         7000.50 |
| Alice Johnson       |         3000.25 |
+---------------------+-----------------+
3 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.

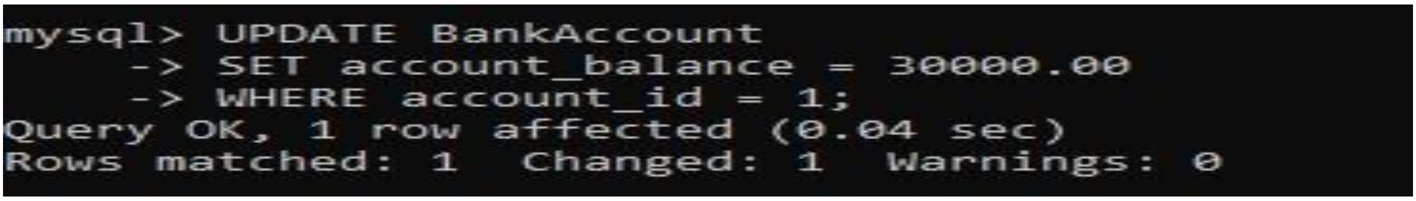


```
mysql> SELECT account_holder_name, account_balance
    -> FROM BankAccount
    -> WHERE account_balance > 30000.00;
Empty set (0.00 sec)
```

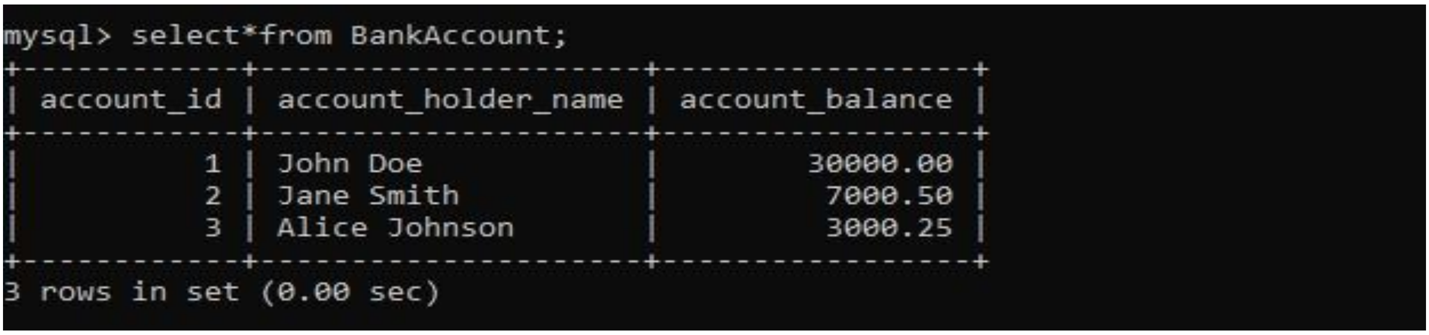
Empty set because there is no account entry with balance more than 30000.

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 = 30000.00
    -> WHERE account_id = 1;
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```



```
mysql> select*from BankAccount;
+------------+---------------------+-----------------+
| account_id | account_holder_name | account_balance |
+------------+---------------------+-----------------+
|          1 | John Doe            |        30000.00 |
|          2 | Jane Smith          |         7000.50 |
|          3 | Alice Johnson       |         3000.25 |
+------------+---------------------+-----------------+
3 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: Retrieve information about employees who belong to the "Sales" department and have a salary greater than 50,000.

```
SELECT *
FROM employees
WHERE department = 'Sales' AND salary > 50000;
```

Scenario 2: Remove the record of an employee who has resigned from the "employees" table.

```
DELETE FROM employees
WHERE employee_status = 'Resigned';
```

Scenario 3: Delete all orders placed before '2022-01-01' that are still in the 'Pending' status.

```
DELETE FROM orders
WHERE order_date < '2022-01-01' AND order_status = 'Pending';
```

Scenario 4: Remove all products from the "Discontinued" category.

```
DELETE FROM products
WHERE category = 'Discontinued';
```

Scenario 5: Add 1000 to the bonus column for all employees in the "Sales" department.

```
UPDATE employees
SET bonus = bonus + 1000
WHERE department = 'Sales';
```