# Assignment 3

# BY SHREYASI REJA

## Banking System

### Tasks 1: Database Design:

1. Create the database named "HMBank"
   QUERY:-
   CREATE DATABASE HMBank;


2. Define the schema for the Customers, Accounts, and Transactions tables based on the provided schema.

- Customers:-

CREATE TABLE Customers (

-> CustomerID INT PRIMARY KEY,

-> FirstName VARCHAR(50),

-> LastName VARCHAR(50),

-> DOB DATE,

-> Email VARCHAR(100),

-> PhoneNumber VARCHAR(15),

-> Address VARCHAR(255)

-> );

```
mysql> CREATE TABLE Customers (
    -> CustomerID INT PRIMARY KEY,
    -> FirstName VARCHAR(50),
    -> LastName VARCHAR(50),
    -> DOB DATE,
    -> Email VARCHAR(100),
    -> PhoneNumber VARCHAR(15),
    -> Address VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.11 sec)

mysql> desc Customers;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| CustomerID  | int          | NO   | PRI | NULL    |       |
| FirstName   | varchar(50)  | YES  |     | NULL    |       |
| LastName    | varchar(50)  | YES  |     | NULL    |       |
| DOB         | date         | YES  |     | NULL    |       |
| Email       | varchar(100) | YES  |     | NULL    |       |
| PhoneNumber | varchar(15)  | YES  |     | NULL    |       |
| Address     | varchar(255) | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
7 rows in set (0.01 sec)

mysql>
```

• Accounts:-

CREATE TABLE Accounts (

-> AccountID INT PRIMARY KEY,

-> CustomerID INT,

-> AccountType VARCHAR(20),

-> Balance DECIMAL(10, 2),

-> FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

-> );

```
mysql> CREATE TABLE Accounts (
    ->     AccountID INT PRIMARY KEY,
    ->     CustomerID INT,
    ->     AccountType VARCHAR(20),
    ->     Balance DECIMAL(10, 2),
    ->     FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
    -> );
Query OK, 0 rows affected (0.10 sec)

mysql> desc Accounts;
+-------------+---------------+------+-----+---------+-------+
| Field       | Type          | Null | Key | Default | Extra |
+-------------+---------------+------+-----+---------+-------+
| AccountID   | int           | NO   | PRI | NULL    |       |
| CustomerID  | int           | YES  | MUL | NULL    |       |
| AccountType | varchar(20)   | YES  |     | NULL    |       |
| Balance     | decimal(10,2) | YES  |     | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql>
```

• Transactions:-

mysql> CREATE TABLE Transactions (

-> TransactionID INT PRIMARY KEY,

-> AccountID INT,

-> TransactionType VARCHAR(20),

-> Amount DECIMAL(10, 2),

-> TransactionDate DATE,

-> FOREIGN KEY (AccountID) REFERENCES Accounts(AccountID)

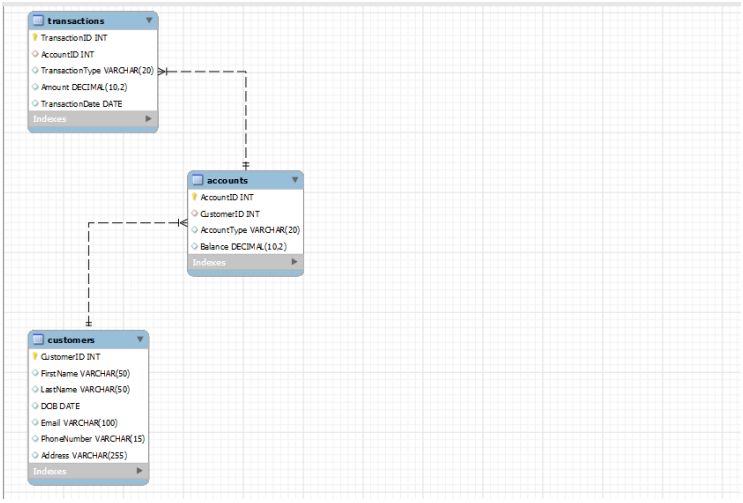
-> );

```
mysql> CREATE TABLE Transactions (
    ->     TransactionID INT PRIMARY KEY,
    ->     AccountID INT,
    ->     TransactionType VARCHAR(20),
    ->     Amount DECIMAL(10, 2),
    ->     TransactionDate DATE,
    ->     FOREIGN KEY (AccountID) REFERENCES Accounts(AccountID)
    -> );
Query OK, 0 rows affected (0.11 sec)

mysql> desc Transactions;
+-----------------+---------------+------+-----+---------+-------+
| Field           | Type          | Null | Key | Default | Extra |
+-----------------+---------------+------+-----+---------+-------+
| TransactionID   | int           | NO   | PRI | NULL    |       |
| AccountID       | int           | YES  | MUL | NULL    |       |
| TransactionType | varchar(20)   | YES  |     | NULL    |       |
| Amount          | decimal(10,2) | YES  |     | NULL    |       |
| TransactionDate | date          | YES  |     | NULL    |       |
+-----------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql>
```

4.Create an ERD (Entity Relationship Diagram) for the database



6. Write SQL scripts to create the mentioned tables with appropriate data types, constraints, and relationships.

• Customers • Accounts • Transactions

• Customers:-

CREATE TABLE Customers (

-> CustomerID INT PRIMARY KEY,

-> FirstName VARCHAR(50),

-> LastName VARCHAR(50),

-> DOB DATE,

-> Email VARCHAR(100),

-> PhoneNumber VARCHAR(15),

-> Address VARCHAR(255)

-> );

```
mysql> CREATE TABLE Customers (
    -> CustomerID INT PRIMARY KEY,
    -> FirstName VARCHAR(50),
    -> LastName VARCHAR(50),
    -> DOB DATE,
    -> Email VARCHAR(100),
    -> PhoneNumber VARCHAR(15),
    -> Address VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.11 sec)

mysql> desc Customers;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| CustomerID  | int          | NO   | PRI | NULL    |       |
| FirstName   | varchar(50)  | YES  |     | NULL    |       |
| LastName    | varchar(50)  | YES  |     | NULL    |       |
| DOB         | date         | YES  |     | NULL    |       |
| Email       | varchar(100) | YES  |     | NULL    |       |
| PhoneNumber | varchar(15)  | YES  |     | NULL    |       |
| Address     | varchar(255) | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
7 rows in set (0.01 sec)

mysql>
```

• Accounts:-

CREATE TABLE Accounts (

-> AccountID INT PRIMARY KEY,

-> CustomerID INT,

-> AccountType VARCHAR(20),

-> Balance DECIMAL(10, 2),

-> FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

-> );

```
mysql> CREATE TABLE Accounts (
    ->     AccountID INT PRIMARY KEY,
    ->     CustomerID INT,
    ->     AccountType VARCHAR(20),
    ->     Balance DECIMAL(10, 2),
    ->     FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
    -> );
Query OK, 0 rows affected (0.10 sec)

mysql> desc Accounts;
+-------------+---------------+------+-----+---------+-------+
| Field       | Type          | Null | Key | Default | Extra |
+-------------+---------------+------+-----+---------+-------+
| AccountID   | int           | NO   | PRI | NULL    |       |
| CustomerID  | int           | YES  | MUL | NULL    |       |
| AccountType | varchar(20)   | YES  |     | NULL    |       |
| Balance     | decimal(10,2) | YES  |     | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql>
```

• Transactions:-

mysql> CREATE TABLE Transactions (

-> TransactionID INT PRIMARY KEY,

-> AccountID INT,

-> TransactionType VARCHAR(20),

-> Amount DECIMAL(10, 2),

-> TransactionDate DATE,

-> FOREIGN KEY (AccountID) REFERENCES Accounts(AccountID)

-> );

```
mysql> CREATE TABLE Transactions (
    ->     TransactionID INT PRIMARY KEY,
    ->     AccountID INT,
    ->     TransactionType VARCHAR(20),
    ->     Amount DECIMAL(10, 2),
    ->     TransactionDate DATE,
    ->     FOREIGN KEY (AccountID) REFERENCES Accounts(AccountID)
    -> );
Query OK, 0 rows affected (0.11 sec)

mysql> desc Transactions;
+-----------------+---------------+------+-----+---------+-------+
| Field           | Type          | Null | Key | Default | Extra |
+-----------------+---------------+------+-----+---------+-------+
| TransactionID   | int           | NO   | PRI | NULL    |       |
| AccountID       | int           | YES  | MUL | NULL    |       |
| TransactionType | varchar(20)   | YES  |     | NULL    |       |
| Amount          | decimal(10,2) | YES  |     | NULL    |       |
| TransactionDate | date          | YES  |     | NULL    |       |
+-----------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql>
```

## Tasks 2: Select, Where, Between, AND, LIKE:

## 1. Insert at least 10 sample records into each of the following tables.

## Customers

```
mysql> select * from Customers;
+------------+-----------+----------+------------+--------------------------+-------------+---------+
| CustomerID | FirstName | LastName | DOB        | Email                    | PhoneNumber | Address |
+------------+-----------+----------+------------+--------------------------+-------------+---------+
|          1 | Raj       | Roy      | 1999-01-15 | rajroy2023@email.com     | 9786457445  | Kolkata |
|          2 | Rohit     | Ray      | 1999-01-16 | rohitray2023@email.com   | 9786457446  | Hooghly |
|          3 | Rahul     | Roy      | 1999-02-17 | rahulroy2023@email.com   | 9786457447  | Kolkata |
|          4 | Akash     | Dey      | 1999-02-18 | akashdey2023@email.com   | 9786457448  | Howrah  |
|          5 | Amit      | Mishra   | 1999-03-19 | amitmishra2023@email.com | 9786457449  | Delhi   |
|          6 | Arti      | Mishra   | 1999-04-20 | artimishra2023@email.com | 9786457441  | Odisha  |
|          7 | Piu       | Das      | 1999-04-09 | piudas2023@email.com     | 9786457442  | Bihar   |
|          8 | Shreya    | Sen      | 1999-04-13 | shreyassen2023@email.com | 9786457443  | Howrah  |
|          9 | Anik      | Singh    | 1999-06-11 | aniksingh2023@email.com  | 9786457444  | Mumbai  |
|         10 | Ayesha    | Singh    | 1999-09-12 | ayeshasingh2023@email.com | 9786457440 | Kolkata |
+------------+-----------+----------+------------+--------------------------+-------------+---------+
10 rows in set (0.00 sec)
```

## Accounts

```
mysql> select * from Accounts;
+-----------+------------+--------------+----------+
| AccountID | CustomerID | AccountType  | Balance  |
+-----------+------------+--------------+----------+
|       101 |          1 | Savings      |  5000.00 |
|       102 |          2 | Current      | 10000.00 |
|       103 |          3 | Zero Balance |     0.00 |
|       104 |          4 | Savings      |  7500.50 |
|       105 |          5 | Current      | 12000.75 |
|       106 |          6 | Savings      |  3000.25 |
|       107 |          7 | Current      |  8000.90 |
|       108 |          8 | Zero Balance |     0.00 |
|       109 |          9 | Savings      |  6000.00 |
|       110 |         10 | Current      |  9500.80 |
+-----------+------------+--------------+----------+
10 rows in set (0.00 sec)
```

## Transactions

```
mysql> select * from Transactions;
+---------------+-----------+-----------------+---------+-----------------+
| TransactionID | AccountID | TransactionType | Amount  | TransactionDate |
+---------------+-----------+-----------------+---------+-----------------+
|          1001 |       101 | Deposit         |  500.00 | 2024-01-16      |
|          1002 |       102 | Withdrawal      | 1000.00 | 2024-01-17      |
|          1003 |       103 | Deposit         |  250.50 | 2024-01-18      |
|          1004 |       104 | Withdrawal      |  100.25 | 2024-01-19      |
|          1005 |       105 | Transfer        | 1500.75 | 2024-01-20      |
|          1006 |       106 | Withdrawal      |  500.00 | 2024-01-21      |
|          1007 |       107 | Transfer        |  800.90 | 2024-01-22      |
|          1008 |       108 | Deposit         |  200.00 | 2024-01-23      |
|          1009 |       109 | Withdrawal      |  600.00 | 2024-01-24      |
|          1010 |       110 | Withdrawal      |  700.80 | 2024-01-25      |
+---------------+-----------+-----------------+---------+-----------------+
10 rows in set (0.01 sec)

mysql>
```

2.Write SQL queries for the following tasks:

1. Write a SQL query to retrieve the name, account type and email of all customers.

```
select  Accounts.CustomerID , FirstName , LastName , email , AccountType
   -> from Customers
   -> join accounts on Accounts.CustomerID = Customers.customerID;
```

```
+------------+-----------+----------+--------------------------+--------------+
| CustomerID | FirstName | LastName | email                    | AccountType  |
+------------+-----------+----------+--------------------------+--------------+
|          1 | Raj       | Roy      | rajroy2023@email.com     | Savings      |
|          2 | Rohit     | Ray      | rohitray2023@email.com   | Current      |
|          3 | Rahul     | Roy      | rahulroy2023@email.com   | Zero Balance |
|          4 | Akash     | Dey      | akashdey2023@email.com   | Savings      |
|          5 | Amit      | Mishra   | amitmishra2023@email.com | Current      |
|          6 | Arti      | Mishra   | artimishra2023@email.com | Savings      |
|          7 | Piu       | Das      | piudas2023@email.com     | Current      |
|          8 | Shreya    | Sen      | shreyassen2023@email.com | Zero Balance |
|          9 | Anik      | Singh    | aniksingh2023@email.com  | Savings      |
|         10 | Ayesha    | Singh    | ayeshasingh2023@email.com | Current     |
+------------+-----------+----------+--------------------------+--------------+
10 rows in set (0.01 sec)

mysql>
```

2.Write a SQL query to list all transaction corresponding customer.

```
SELECT Transactions.TransactionID , Transactions.AccountID , Accounts.CustomerID , FirstName , LastName
   -> FROM Transactions
   -> JOIN Accounts ON Transactions.AccountID = Accounts.AccountID
   -> JOIN Customers ON Accounts.CustomerID = Customers.CustomerID;
```

```
+---------------+-----------+------------+-----------+----------+
| TransactionID | AccountID | CustomerID | FirstName | LastName |
+---------------+-----------+------------+-----------+----------+
|          1001 |       101 |          1 | Raj       | Roy      |
|          1002 |       102 |          2 | Rohit     | Ray      |
|          1003 |       103 |          3 | Rahul     | Roy      |
|          1004 |       104 |          4 | Akash     | Dey      |
|          1005 |       105 |          5 | Amit      | Mishra   |
|          1006 |       106 |          6 | Arti      | Mishra   |
|          1007 |       107 |          7 | Piu       | Das      |
|          1008 |       108 |          8 | Shreya    | Sen      |
|          1009 |       109 |          9 | Anik      | Singh    |
|          1010 |       110 |         10 | Ayesha    | Singh    |
+---------------+-----------+------------+-----------+----------+
10 rows in set (0.01 sec)

mysql>
```

3. Write a SQL query to increase the balance of a specific account by a certain amount.
   update accounts set Balance = 1.1 * Balance where AccountType like 'savings';

```
mysql> SELECT * FROM Account;
ERROR 1146 (42S02): Table 'hmbank.account' doesn't exist
mysql> SELECT * FROM Accounts;
+-----------+------------+--------------+----------+
| AccountID | CustomerID | AccountType  | Balance  |
+-----------+------------+--------------+----------+
|       101 |          1 | Savings      |  5500.00 |
|       102 |          2 | Current      | 10000.00 |
|       103 |          3 | Zero Balance |     0.00 |
|       104 |          4 | Savings      |  8250.55 |
|       105 |          5 | Current      | 12000.75 |
|       106 |          6 | Savings      |  3300.28 |
|       107 |          7 | Current      |  8000.90 |
|       108 |          8 | Zero Balance |     0.00 |
|       109 |          9 | Savings      |  6600.00 |
|       110 |         10 | Current      |  9500.80 |
+-----------+------------+--------------+----------+
10 rows in set (0.00 sec)

mysql>
```

4. Write a SQL query to Combine first and last names of customers as a full_name.
   SELECT CONCAT(FirstName, ' ', LastName) AS FullName
   -> FROM Customers;

```
mysql> SELECT CONCAT(FirstName, ' ', LastName) AS FullName
    -> FROM Customers;
+--------------+
| FullName     |
+--------------+
| Raj Roy      |
| Rohit Ray    |
| Rahul Roy    |
| Akash Dey    |
| Amit Mishra  |
| Arti Mishra  |
| Piu Das      |
| Shreya Sen   |
| Anik Singh   |
| Ayesha Singh |
+--------------+
10 rows in set (0.06 sec)

mysql>
```

5. Write a SQL query to remove accounts with a balance of zero where the account type is savings.
   DELETE FROM Accounts
   WHERE balance = 0 AND account_type = 'savings';

```
mysql> /*
   /*> DELETE FROM Accounts
   /*> WHERE balance = 0 AND account_type = 'savings';
   /*> */
mysql>
```

6. Write a SQL query to Find customers living in a specific city.
   select CONCAT(FirstName ,' ' , LastName) as fullname , Address
   -> from Customers where Address like 'Mumbai';

```
mysql> select CONCAT(FirstName ,' ' , LastName) as fullname , Address
    -> from Customers where Address like 'Mumbai';
+------------+---------+
| fullname   | Address |
+------------+---------+
| Anik Singh | Mumbai  |
+------------+---------+
1 row in set (0.00 sec)

mysql>
```

7.Write a SQL query to Get the account balance for a specific account.

SELECT  AccountID , Balance

-> FROM Accounts

-> WHERE AccountID = 104;

```
mysql> SELECT  AccountID , Balance
    -> FROM Accounts
    -> WHERE AccountID = 104;
+-----------+---------+
| AccountID | Balance |
+-----------+---------+
|       104 | 8250.55 |
+-----------+---------+
1 row in set (0.00 sec)

mysql>
```

8. Write a SQL query to List all current accounts with a balance greater than $1,000.

select * from Accounts

-> where AccountType like 'current'

-> and Balance > 83124 ; -- $1000 is equal to Rs.83124

```
mysql> select * from Accounts
    -> where AccountType like 'current'
    -> and Balance > 83124 ; -- $1000 is equal to Rs.83124
Empty set (0.01 sec)

mysql>
```

9. Write a SQL query to Retrieve all transactions for a specific account.

select * from Transactions

-> having AccountID = 105;

```
mysql> select * from Transactions
    -> having AccountID = 105;
+---------------+-----------+-----------------+---------+-----------------+
| TransactionID | AccountID | TransactionType | Amount  | TransactionDate |
+---------------+-----------+-----------------+---------+-----------------+
|          1005 |       105 | Transfer        | 1500.75 | 2024-01-20      |
+---------------+-----------+-----------------+---------+-----------------+
1 row in set (0.00 sec)

mysql>
```

10. Write a SQL query to Calculate the interest accrued on savings accounts based on a given interest rate.

SELECT AccountID,Balance as CurrentBalance , (Balance * 1.04)- Balance AS AccruedInterest ,

-> Balance * 1.04 as BalanceAfterInterest

-> FROM Accounts

-> WHERE AccountType = 'savings';

```
mysql> SELECT AccountID,Balance as CurrentBalance , (Balance * 1.04)- Balance AS AccruedInterest ,
    -> Balance * 1.04 as BalanceAfterInterest
    -> FROM Accounts
    -> WHERE AccountType = 'savings';
+-----------+----------------+-----------------+---------------------+
| AccountID | CurrentBalance | AccruedInterest | BalanceAfterInterest |
+-----------+----------------+-----------------+---------------------+
|       101 |        5500.00 |        220.0000 |            5720.0000 |
|       104 |        8250.55 |        330.0220 |            8580.5720 |
|       106 |        3300.28 |        132.0112 |            3432.2912 |
|       109 |        6600.00 |        264.0000 |            6864.0000 |
+-----------+----------------+-----------------+---------------------+
4 rows in set (0.01 sec)

mysql>
```

11. Write a SQL query to Identify accounts where the balance is less than a specified overdraft limit.

SET @overdraftLimit = 50000.00;

SELECT *

-> FROM Accounts

-> WHERE balance < @overdraftLimit and AccountType not like 'ZeroBalance' ;

```
mysql> SET @overdraftLimit = 50000.00;
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT *
    -> FROM Accounts
    -> WHERE balance < @overdraftLimit and AccountType not like 'ZeroBalance' ;
+-----------+------------+--------------+----------+
| AccountID | CustomerID | AccountType  | Balance  |
+-----------+------------+--------------+----------+
|       101 |          1 | Savings      |  5500.00 |
|       102 |          2 | Current      | 10000.00 |
|       103 |          3 | Zero Balance |     0.00 |
|       104 |          4 | Savings      |  8250.55 |
|       105 |          5 | Current      | 12000.75 |
|       106 |          6 | Savings      |  3300.28 |
|       107 |          7 | Current      |  8000.90 |
|       108 |          8 | Zero Balance |     0.00 |
|       109 |          9 | Savings      |  6600.00 |
|       110 |         10 | Current      |  9500.80 |
+-----------+------------+--------------+----------+
10 rows in set (0.01 sec)

mysql>
```

12. Write a SQL query to Find customers not living in a specific city.

select * from Customers

-> where Address not IN('Kolkata','Mumbai','Bihar');

```
mysql> select * from Customers
    -> where Address not IN('Kolkata','Mumbai','Bihar');
+------------+-----------+----------+------------+-------------------------+-------------+---------+
| CustomerID | FirstName | LastName | DOB        | Email                   | PhoneNumber | Address |
+------------+-----------+----------+------------+-------------------------+-------------+---------+
|          2 | Rohit     | Ray      | 1999-01-16 | rohitray2023@email.com  | 9786457446  | Hooghly |
|          4 | Akash     | Dey      | 1999-02-18 | akashdey2023@email.com  | 9786457448  | Howrah  |
|          5 | Amit      | Mishra   | 1999-03-19 | amitmishra2023@email.com| 9786457449  | Delhi   |
|          6 | Arti      | Mishra   | 1999-04-20 | artimishra2023@email.com| 9786457441  | Odisha  |
|          8 | Shreya    | Sen      | 1999-04-13 | shreyassen2023@email.com| 9786457443  | Howrah  |
+------------+-----------+----------+------------+-------------------------+-------------+---------+
5 rows in set (0.00 sec)

mysql>
```

## Tasks 3: Aggregate functions, Having, Order By, GroupBy and Joins:

1. Write a SQL query to Find the average account balance for all customers.

select Customers.CustomerID , FirstName , LastName , avg(Balance) as AvgBalance

-> from Accounts

-> join Customers where Customers.CustomerID = Accounts.CustomerID

-> group by CustomerID;

```
mysql> select Customers.CustomerID , FirstName , LastName , avg(Balance) as AvgBalance
    -> from Accounts
    -> join Customers where Customers.CustomerID = Accounts.CustomerID
    -> group by CustomerID;
+------------+-----------+----------+--------------+
| CustomerID | FirstName | LastName | AvgBalance   |
+------------+-----------+----------+--------------+
|          1 | Raj       | Roy      |  5500.000000 |
|          2 | Rohit     | Ray      | 10000.000000 |
|          3 | Rahul     | Roy      |     0.000000 |
|          4 | Akash     | Dey      |  8250.550000 |
|          5 | Amit      | Mishra   | 12000.750000 |
|          6 | Arti      | Mishra   |  3300.280000 |
|          7 | Piu       | Das      |  8000.900000 |
|          8 | Shreya    | Sen      |     0.000000 |
|          9 | Anik      | Singh    |  6600.000000 |
|         10 | Ayesha    | Singh    |  9500.800000 |
+------------+-----------+----------+--------------+
10 rows in set (0.05 sec)

mysql>
```

2. Write a SQL query to Retrieve the top 10 highest account balances.

select * from Accounts

-> order by Balance desc

-> limit 10;

```
mysql> select * from Accounts
    -> order by Balance desc
    -> limit 10;
+-----------+------------+--------------+----------+
| AccountID | CustomerID | AccountType  | Balance  |
+-----------+------------+--------------+----------+
|       105 |          5 | Current      | 12000.75 |
|       102 |          2 | Current      | 10000.00 |
|       110 |         10 | Current      |  9500.80 |
|       104 |          4 | Savings      |  8250.55 |
|       107 |          7 | Current      |  8000.90 |
|       109 |          9 | Savings      |  6600.00 |
|       101 |          1 | Savings      |  5500.00 |
|       106 |          6 | Savings      |  3300.28 |
|       103 |          3 | Zero Balance |     0.00 |
|       108 |          8 | Zero Balance |     0.00 |
+-----------+------------+--------------+----------+
10 rows in set (0.00 sec)

mysql>
```

3. Write a SQL query to Calculate Total Deposits for All Customers in specific date.

```
select TransactionID , AccountID , sum(Amount) as TotalDeposit
    -> from Transactions
    -> where TransactionType like 'deposit'
    -> and TransactionDate = '24-01-03'
    -> group by TransactionID;
```

```
mysql> select TransactionID , AccountID , sum(Amount) as TotalDeposit
    -> from Transactions
    -> where TransactionType like 'deposit'
    -> and TransactionDate = '24-01-03'
    -> group by TransactionID;
Empty set (0.05 sec)

mysql>
```

4. Write a SQL query to Find the Oldest and Newest Customers.

```
select min(DOB) as OldestCustomer ,
    -> max(DOB) as NewestCustomer
    -> from Customers;
```

```
mysql> select min(DOB) as OldestCustomer ,
    -> max(DOB) as NewestCustomer
    -> from Customers;
+----------------+----------------+
| OldestCustomer | NewestCustomer |
+----------------+----------------+
| 1999-01-15     | 1999-09-12     |
+----------------+----------------+
1 row in set (0.00 sec)

mysql>
```

5.Write a SQL query to Retrieve transaction details along with the account type.

```
SELECT Transactions.*, Accounts.AccountType
    -> FROM Transactions
    -> JOIN Accounts ON Transactions.AccountID =
Accounts.AccountID;
```

```
mysql> SELECT Transactions.*, Accounts.AccountType
    -> FROM Transactions
    -> JOIN Accounts ON Transactions.AccountID = Accounts.AccountID;
+---------------+-----------+-----------------+---------+-----------------+--------------+
| TransactionID | AccountID | TransactionType | Amount  | TransactionDate | AccountType  |
+---------------+-----------+-----------------+---------+-----------------+--------------+
|          1001 |       101 | Deposit         |  500.00 | 2024-01-16      | Savings      |
|          1002 |       102 | Withdrawal      | 1000.00 | 2024-01-17      | Current      |
|          1003 |       103 | Deposit         |  250.50 | 2024-01-18      | Zero Balance |
|          1004 |       104 | Withdrawal      |  100.25 | 2024-01-19      | Savings      |
|          1005 |       105 | Transfer        | 1500.75 | 2024-01-20      | Current      |
|          1006 |       106 | Withdrawal      |  500.00 | 2024-01-21      | Savings      |
|          1007 |       107 | Transfer        |  800.90 | 2024-01-22      | Current      |
|          1008 |       108 | Deposit         |  200.00 | 2024-01-23      | Zero Balance |
|          1009 |       109 | Withdrawal      |  600.00 | 2024-01-24      | Savings      |
|          1010 |       110 | Withdrawal      |  700.80 | 2024-01-25      | Current      |
+---------------+-----------+-----------------+---------+-----------------+--------------+
10 rows in set (0.00 sec)

mysql>
```

6. Write a SQL query to Get a list of customers along with their account details.

   select Customers.* , Accounts.*

   -> from Customers

   -> join Accounts on Accounts.CustomerID = Customers.CustomerID;

```
mysql> select Customers.* , Accounts.*
    -> from Customers
    -> join Accounts on Accounts.CustomerID = Customers.CustomerID;
+------------+-----------+----------+------------+-------------------------+-------------+---------+-----------+------------+--------------+----------+
| CustomerID | FirstName | LastName | DOB        | Email                   | PhoneNumber | Address | AccountID | CustomerID | AccountType  | Balance  |
+------------+-----------+----------+------------+-------------------------+-------------+---------+-----------+------------+--------------+----------+
|          1 | Raj       | Roy      | 1999-01-15 | rajroy2023@email.com    | 9786457445  | Kolkata |       101 |          1 | Savings      |  5500.00 |
|          2 | Rohit     | Ray      | 1999-01-16 | rohitray2023@email.com  | 9786457446  | Hooghly |       102 |          2 | Current      | 10000.00 |
|          3 | Rahul     | Roy      | 1999-02-17 | rahulroy2023@email.com  | 9786457447  | Kolkata |       103 |          3 | Zero Balance |     0.00 |
|          4 | Akash     | Dey      | 1999-02-18 | akashdey2023@email.com  | 9786457448  | Howrah  |       104 |          4 | Savings      |  8250.55 |
|          5 | Amit      | Mishra   | 1999-03-19 | amitmishra2023@email.com| 9786457449  | Delhi   |       105 |          5 | Current      | 12000.75 |
|          6 | Arti      | Mishra   | 1999-04-20 | artimishra2023@email.com| 9786457441  | Odisha  |       106 |          6 | Savings      |  3300.28 |
|          7 | Piu       | Das      | 1999-04-09 | piudas2023@email.com    | 9786457442  | Bihar   |       107 |          7 | Current      |  8000.90 |
|          8 | Shreya    | Sen      | 1999-04-13 | shreyassen2023@email.com| 9786457443  | Howrah  |       108 |          8 | Zero Balance |     0.00 |
|          9 | Anik      | Singh    | 1999-06-11 | aniksingh2023@email.com | 9786457444  | Mumbai  |       109 |          9 | Savings      |  6600.00 |
|         10 | Ayesha    | Singh    | 1999-09-12 | ayeshasingh2023@email.com| 9786457440 | Kolkata |       110 |         10 | Current      |  9500.80 |
+------------+-----------+----------+------------+-------------------------+-------------+---------+-----------+------------+--------------+----------+
10 rows in set (0.00 sec)

mysql>
```

7. Write a SQL query to Retrieve transaction details along with customer information for a specific account.

   SELECT Transactions.*, Customers.*

   -> FROM Transactions

   -> JOIN Accounts ON Transactions.AccountID = Accounts.AccountID

   -> JOIN Customers ON Accounts.CustomerID = Customers.CustomerID

   -> WHERE Accounts.customerID = 4;

```
mysql> SELECT Transactions.*, Customers.*
    -> FROM Transactions
    -> JOIN Accounts ON Transactions.AccountID = Accounts.AccountID
    -> JOIN Customers ON Accounts.CustomerID = Customers.CustomerID
    -> WHERE Accounts.customerID = 4;
+---------------+-----------+-----------------+--------+-----------------+------------+-----------+----------+------------+-------------------------+-------------+---------+
| TransactionID | AccountID | TransactionType | Amount | TransactionDate | CustomerID | FirstName | LastName | DOB        | Email                   | PhoneNumber | Address |
+---------------+-----------+-----------------+--------+-----------------+------------+-----------+----------+------------+-------------------------+-------------+---------+
|          1004 |       104 | Withdrawal      | 100.25 | 2024-01-19      |          4 | Akash     | Dey      | 1999-02-18 | akashdey2023@email.com  | 9786457448  | Howrah  |
+---------------+-----------+-----------------+--------+-----------------+------------+-----------+----------+------------+-------------------------+-------------+---------+
1 row in set (0.00 sec)

mysql>
```

8. Write a SQL query to Identify customers who have more than one account.

```
SELECT CustomerID
    -> FROM Accounts
    -> GROUP BY CustomerID
    -> HAVING COUNT(*) > 1;
```

```
mysql> SELECT CustomerID
    -> FROM Accounts
    -> GROUP BY CustomerID
    -> HAVING COUNT(*) > 1;
Empty set (0.00 sec)

mysql>
```

9. Write a SQL query to Calculate the difference in transaction amounts between deposits and withdrawals.

```
SELECT SUM(CASE WHEN TransactionType = 'deposit' THEN
Amount ELSE -Amount END) AS NetTransactionAmount
    -> FROM Transactions;
```

```
mysql> SELECT SUM(CASE WHEN TransactionType = 'deposit' THEN Amount ELSE -Amount END) AS NetTransactionAmount
    -> FROM Transactions;
+----------------------+
| NetTransactionAmount |
+----------------------+
|            -4252.20 |
+----------------------+
1 row in set (0.00 sec)

mysql>
```

10. Write a SQL query to Calculate the average daily balance for each account over a specified period.

```
select Transactions.AccountID , avg(Balance) as DailyAverage
    -> from Accounts
    -> join Transactions on Transactions.AccountID =
Accounts.AccountID
    -> where TransactionDate between '2024-01-01' and '2024-
01-20'
    -> group by AccountID;
```

```
mysql> select Transactions.AccountID , avg(Balance) as DailyAverage
    -> from Accounts
    -> join Transactions on Transactions.AccountID = Accounts.AccountID
    -> where TransactionDate between '2024-01-01' and '2024-01-20'
    -> group by AccountID;
+-----------+--------------+
| AccountID | DailyAverage |
+-----------+--------------+
|       101 |  5500.000000 |
|       102 | 10000.000000 |
|       103 |     0.000000 |
|       104 |  8250.550000 |
|       105 | 12000.750000 |
+-----------+--------------+
5 rows in set (0.00 sec)

mysql>
```

11. Calculate the total balance for each account type.

```
SELECT AccountType, SUM(Balance) AS TotalBalance
    -> FROM Accounts
    -> GROUP BY AccountType;
```

```
mysql> SELECT AccountType, SUM(Balance) AS TotalBalance
    -> FROM Accounts
    -> GROUP BY AccountType;
+--------------+--------------+
| AccountType  | TotalBalance |
+--------------+--------------+
| Savings      |     23650.83 |
| Current      |     39502.45 |
| Zero Balance |         0.00 |
+--------------+--------------+
3 rows in set (0.00 sec)

mysql>
```

12. Identify accounts with the highest number of transactions order by descending order.

```
SELECT AccountID, COUNT(TransactionID) AS TransactionCount
    -> FROM Transactions
    -> GROUP BY AccountID
    -> ORDER BY TransactionCount DESC;
```

```
mysql> SELECT AccountID, COUNT(TransactionID) AS TransactionCount
    -> FROM Transactions
    -> GROUP BY AccountID
    -> ORDER BY TransactionCount DESC;
+-----------+------------------+
| AccountID | TransactionCount |
+-----------+------------------+
|       101 |                1 |
|       102 |                1 |
|       103 |                1 |
|       104 |                1 |
|       105 |                1 |
|       106 |                1 |
|       107 |                1 |
|       108 |                1 |
|       109 |                1 |
|       110 |                1 |
+-----------+------------------+
10 rows in set (0.00 sec)

mysql>
```

13. List customers with high aggregate account balances, along with their account types.

```
SELECT Customers.CustomerID, SUM(Balance) AS
AggregateBalance, GROUP_CONCAT(AccountType) AS
AccountTypes
    -> FROM Customers
    -> JOIN Accounts ON Customers.CustomerID =
Accounts.CustomerID
    -> GROUP BY Customers.CustomerID
    -> HAVING AggregateBalance > 10000;
```

```
mysql> SELECT Customers.CustomerID, SUM(Balance) AS AggregateBalance, GROUP_CONCAT(AccountType) AS AccountTypes
    -> FROM Customers
    -> JOIN Accounts ON Customers.CustomerID = Accounts.CustomerID
    -> GROUP BY Customers.CustomerID
    -> HAVING AggregateBalance > 10000;
+------------+------------------+--------------+
| CustomerID | AggregateBalance | AccountTypes |
+------------+------------------+--------------+
|          5 |         12000.75 | Current      |
+------------+------------------+--------------+
1 row in set (0.00 sec)

mysql>
```

14. Identify and list duplicate transactions based on transaction amount, date, and account.

```
SELECT Amount, TransactionDate, AccountID
   -> FROM Transactions
   -> WHERE (Amount, TransactionDate, AccountID) IN (
   -> SELECT Amount, TransactionDate, AccountID
   -> FROM Transactions
   ->     GROUP BY Amount, TransactionDate, AccountID
   ->  HAVING COUNT(*) > 1
   -> );
```

```
mysql> SELECT Amount, TransactionDate, AccountID
    -> FROM Transactions
    -> WHERE (Amount, TransactionDate, AccountID) IN (
    -> SELECT Amount, TransactionDate, AccountID
    -> FROM Transactions
    ->     GROUP BY Amount, TransactionDate, AccountID
    ->  HAVING COUNT(*) > 1
    -> );
Empty set (0.01 sec)

mysql>
```

Tasks 4: Subquery and its type:

1. Retrieve the customer(s) with the highest account balance.

```
SELECT *
   -> FROM Customers
   -> WHERE CustomerID = (
   -> SELECT CustomerID
   -> FROM Accounts
   ->     ORDER BY Balance DESC
   ->  LIMIT 1
   -> );
```

```
mysql> SELECT *
    -> FROM Customers
    -> WHERE CustomerID = (
    -> SELECT CustomerID
    -> FROM Accounts
    ->     ORDER BY Balance DESC
    ->  LIMIT 1
    -> );
+------------+-----------+----------+------------+-------------------------+-------------+---------+
| CustomerID | FirstName | LastName | DOB        | Email                   | PhoneNumber | Address |
+------------+-----------+----------+------------+-------------------------+-------------+---------+
|          5 | Amit      | Mishra   | 1999-03-19 | amitmishra2023@email.com | 9786457449  | Delhi   |
+------------+-----------+----------+------------+-------------------------+-------------+---------+
1 row in set (0.00 sec)

mysql>
```

2. Calculate the average account balance for customers who have more than one account.

```
SELECT AVG(Balance) AS AvgBalance
   -> FROM Accounts
   -> WHERE CustomerID IN (
   -> SELECT CustomerID
   -> FROM Accounts
   ->     GROUP BY CustomerID
   ->  HAVING COUNT(*) > 1
   -> );
```

```
mysql> SELECT AVG(Balance) AS AvgBalance
    -> FROM Accounts
    -> WHERE CustomerID IN (
    -> SELECT CustomerID
    -> FROM Accounts
    ->     GROUP BY CustomerID
    ->  HAVING COUNT(*) > 1
    -> );
+------------+
| AvgBalance |
+------------+
|       NULL |
+------------+
1 row in set (0.01 sec)

mysql>
```

3. Retrieve accounts with transactions whose amounts exceed the average transaction amount.

```
SELECT *
   -> FROM Accounts
   -> WHERE EXISTS (
   -> SELECT 1
   ->     FROM Transactions
   ->  WHERE Transactions.AccountID = Accounts.AccountID
   ->  AND Amount > (SELECT AVG(Amount) FROM
Transactions)
   -> );
```

```
mysql> SELECT *
    -> FROM Accounts
    -> WHERE EXISTS (
    -> SELECT 1
    ->     FROM Transactions
    ->  WHERE Transactions.AccountID = Accounts.AccountID
    ->  AND Amount > (SELECT AVG(Amount) FROM Transactions)
    -> );
+-----------+------------+-------------+----------+
| AccountID | CustomerID | AccountType | Balance  |
+-----------+------------+-------------+----------+
|       102 |          2 | Current     | 10000.00 |
|       105 |          5 | Current     | 12000.75 |
|       107 |          7 | Current     |  8000.90 |
|       110 |         10 | Current     |  9500.80 |
+-----------+------------+-------------+----------+
4 rows in set (0.01 sec)

mysql>
```

4. Identify customers who have no recorded transactions.

   SELECT SUM(Balance) AS TotalBalanceNoTransactions
   -> FROM Accounts
   -> WHERE NOT EXISTS (
   ->  SELECT 1
   ->  FROM Transactions
   ->  WHERE Transactions.AccountID = Accounts.AccountID
   -> );

```
mysql> SELECT SUM(Balance) AS TotalBalanceNoTransactions
    -> FROM Accounts
    -> WHERE NOT EXISTS (
    ->     SELECT 1
    ->     FROM Transactions
    ->     WHERE Transactions.AccountID = Accounts.AccountID
    -> );
+----------------------------+
| TotalBalanceNoTransactions |
+----------------------------+
|                       NULL |
+----------------------------+
1 row in set (0.00 sec)

mysql>
```

5. Calculate the total balance of accounts with no recorded transactions.

   SELECT Transactions.*
   -> FROM Transactions
   -> JOIN Accounts ON Transactions.AccountID = Accounts.AccountID
   -> WHERE Accounts.Balance = (SELECT MIN(Balance) FROM Accounts);

```
mysql> SELECT Transactions.*
    -> FROM Transactions
    -> JOIN Accounts ON Transactions.AccountID = Accounts.AccountID
    -> WHERE Accounts.Balance = (SELECT MIN(Balance) FROM Accounts);
+---------------+-----------+-----------------+--------+-----------------+
| TransactionID | AccountID | TransactionType | Amount | TransactionDate |
+---------------+-----------+-----------------+--------+-----------------+
|          1003 |       103 | Deposit         | 250.50 | 2024-01-18      |
|          1008 |       108 | Deposit         | 200.00 | 2024-01-23      |
+---------------+-----------+-----------------+--------+-----------------+
2 rows in set (0.00 sec)

mysql>
```

6. Retrieve transactions for accounts with the lowest balance.

```
SELECT Customers.*
    -> FROM Customers
    -> WHERE NOT EXISTS (
    ->     SELECT 1
    ->     FROM Accounts
    ->     JOIN Transactions ON Accounts.AccountID = Transactions.AccountID
    ->  WHERE Accounts.CustomerID = Customers.CustomerID
    -> );
```

```
mysql> SELECT Customers.*
    -> FROM Customers
    -> WHERE NOT EXISTS (
    ->     SELECT 1
    ->     FROM Accounts
    ->     JOIN Transactions ON Accounts.AccountID = Transactions.AccountID
    ->  WHERE Accounts.CustomerID = Customers.CustomerID
    -> );
Empty set (0.00 sec)

mysql>
```

7. Identify customers who have accounts of multiple types.

```
SELECT Customers.*
    -> FROM Customers
    -> WHERE EXISTS (
    ->     SELECT 1
    ->     FROM Accounts
    ->     WHERE Accounts.CustomerID = Customers.CustomerID
    ->  GROUP BY AccountType
    ->  HAVING COUNT(DISTINCT AccountType) > 1
    -> );
```

```
mysql> SELECT Customers.*
    -> FROM Customers
    -> WHERE EXISTS (
    ->     SELECT 1
    ->     FROM Accounts
    ->     WHERE Accounts.CustomerID = Customers.CustomerID
    ->  GROUP BY AccountType
    ->  HAVING COUNT(DISTINCT AccountType) > 1
    -> );
Empty set (0.00 sec)

mysql>
```

8. Calculate the percentage of each account type out of the total number of accounts.

SELECT AccountType, COUNT(*) * 100.0 / (SELECT COUNT(*) FROM Accounts) AS Percentage

-> FROM Accounts

-> GROUP BY AccountType;

```
mysql> SELECT AccountType, COUNT(*) * 100.0 / (SELECT COUNT(*) FROM Accounts) AS Percentage
    -> FROM Accounts
    -> GROUP BY AccountType;
+--------------+------------+
| AccountType  | Percentage |
+--------------+------------+
| Savings      |   40.00000 |
| Current      |   40.00000 |
| Zero Balance |   20.00000 |
+--------------+------------+
3 rows in set (0.04 sec)

mysql>
```

9. Retrieve all transactions for a customer with a given customer_id.

SELECT *

-> FROM Transactions

-> join Accounts on Transactions.AccountID = Accounts.AccountID

-> WHERE CustomerID = (SELECT CustomerID FROM Customers WHERE CustomerID = 1);

```
mysql> SELECT *
    -> FROM Transactions
    -> join Accounts on Transactions.AccountID = Accounts.AccountID
    -> WHERE CustomerID = (SELECT CustomerID FROM Customers WHERE CustomerID = 1);
+---------------+-----------+-----------------+--------+-----------------+-----------+------------+-------------+---------+
| TransactionID | AccountID | TransactionType | Amount | TransactionDate | AccountID | CustomerID | AccountType | Balance |
+---------------+-----------+-----------------+--------+-----------------+-----------+------------+-------------+---------+
|          1001 |       101 | Deposit         | 500.00 | 2024-01-16      |       101 |          1 | Savings     | 5500.00 |
+---------------+-----------+-----------------+--------+-----------------+-----------+------------+-------------+---------+
1 row in set (0.00 sec)

mysql>
```

10. Calculate the total balance for each account type, including a subquery within the SELECT clause.

SELECT AccountType, SUM(Balance) AS TotalBalance

-> FROM Accounts

-> GROUP BY AccountType,

-> (SELECT 1);

```
mysql> SELECT AccountType, SUM(Balance) AS TotalBalance
    -> FROM Accounts
    -> GROUP BY AccountType,
    ->  (SELECT 1);
+--------------+--------------+
| AccountType  | TotalBalance |
+--------------+--------------+
| Savings      |     23650.83 |
| Current      |     39502.45 |
| Zero Balance |         0.00 |
+--------------+--------------+
3 rows in set (0.00 sec)

mysql>
```