Assignment -3

Banking System

Submitted by : Mridul Bhardwaj

Tasks 1: Database Design:

1. Create the database named "HMBank"

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

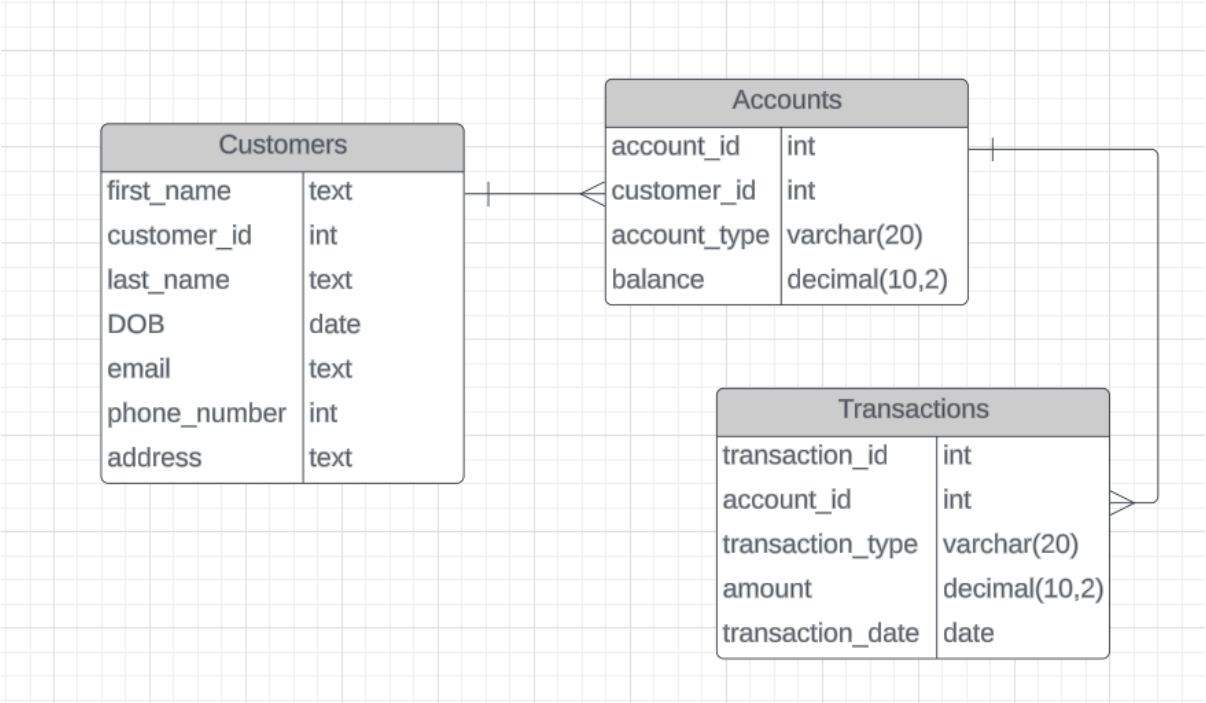
2. Define the schema for the Customers, Accounts, and Transactions tables based on the provided schema.

```
mysql> desc Customers;
+--------------+------+------+-----+---------+-------+
| Field        | Type | Null | Key | Default | Extra |
+--------------+------+------+-----+---------+-------+
| customer_id  | int  | NO   | PRI | NULL    |       |
| first_name   | text | YES  |     | NULL    |       |
| last_name    | text | YES  |     | NULL    |       |
| DOB          | date | YES  |     | NULL    |       |
| email        | text | YES  |     | NULL    |       |
| phone_number | int  | YES  |     | NULL    |       |
| address      | text | YES  |     | NULL    |       |
+--------------+------+------+-----+---------+-------+
7 rows in set (0.01 sec)
```

```
mysql> desc Accounts;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| account_id   | int          | NO   | PRI | NULL    |       |
| customer_id  | int          | YES  | MUL | NULL    |       |
| account_type | varchar(20)  | YES  |     | NULL    |       |
| balance      | decimal(10,2) | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

```
mysql> desc Transactions;
+------------------+---------------+------+-----+---------+-------+
| Field            | Type          | Null | Key | Default | Extra |
+------------------+---------------+------+-----+---------+-------+
| transaction_id   | int           | NO   | PRI | NULL    |       |
| account_id       | int           | YES  | MUL | NULL    |       |
| transaction_type | varchar(20)   | YES  |     | NULL    |       |
| amount           | decimal(10,2) | YES  |     | NULL    |       |
| transaction_date | date          | YES  |     | NULL    |       |
+------------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

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



4. Create appropriate Primary Key and Foreign Key constraints for referential integrity.

```
mysql> CREATE TABLE Accounts (
    ->     account_id INT PRIMARY KEY,
    ->     customer_id INT,
    ->     account_type VARCHAR(20) CHECK (account_type IN ('savings', 'current', 'zero_balance')),
    ->     balance DECIMAL(10, 2),
    ->     FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
    -> );
Query OK, 0 rows affected (0.07 sec)
```

```
mysql> CREATE TABLE Transactions (
    ->     transaction_id INT PRIMARY KEY,
    ->     account_id INT,
    ->     transaction_type VARCHAR(20) CHECK (transaction_type IN ('deposit', 'withdrawal', 'transfer')),
    ->     amount DECIMAL(10, 2),
    ->     transaction_date TIMESTAMP,
    ->     FOREIGN KEY (account_id) REFERENCES Accounts(account_id)
    -> );
Query OK, 0 rows affected (0.09 sec)
```

```
mysql> alter table Customers add primary key(customer_id);
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

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

• Customers

```
mysql> create table Customers(customer_id int , first_name text, last_name text, DOB date, email text, phone_number int,
 address text);
Query OK, 0 rows affected (0.07 sec)
```

• Accounts

```
mysql> CREATE TABLE Accounts (
    ->     account_id INT PRIMARY KEY,
    ->     customer_id INT,
    ->     account_type VARCHAR(20) CHECK (account_type IN ('savings', 'current', 'zero_balance')),
    ->     balance DECIMAL(10, 2),
    ->     FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
    -> );
Query OK, 0 rows affected (0.07 sec)
```

• Transactions

```
mysql> create table Transactions (transaction_id int primary key,
    -> account_id int, transaction_type varchar(20) check(transaction_type in('deposit','withdrawal','transfer')), amount decimal(10,2),transaction_date dat
e, foreign key(account_id) references Accounts(account_id));
Query OK, 0 rows affected (0.08 sec)
```

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

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

• Customers

```
mysql> insert into Customers (customer_id,first_name,last_name,DOB,email,phone_number,address) values(
    -> 1,'Aman','Verma','2001-12-15','aman@abc.com',1234567890,'45 Saket'),
    -> (2,'Abhinav','Shukla','2000-04-26','abhinav@dfr.com',1245789630,'78 Pawal'),
    -> (3,'Abhishek','Parihar','2000-06-17','abhi@ert.com',1245780369,'88 Rohini'),
    -> (4,'Chanchal','Gupta','2002-09-04','chanchal@yui.com',1023457896,'56 Alwar'),
    -> (5,'Dimple','Singh','2002-06-08','dimple@wer.com',1045789632,'44 Malhar'),
    -> (6,'Rishi','Gill','2001-02-18','gill@ert.com',1203789654,'15 Omax'),
    -> (7,'Satya','Sehgal','2001-01-07','satya@wsd.com',1478520369,'45 Rohini'),
    -> (8,'Palak','Pal','2000-11-15','palak@aqw.com',1457896320,'48 Palika'),
    -> (9,'Tina','Noor','1999-12-06','tina@qwe.com',1452036987,'71 Tilak'),
    -> (10,'Tarun','Tal','2000-10-29','tarun@euio.com',1023457896,'55 Krishna');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from Customers;
+-------------+------------+-----------+------------+------------------+--------------+------------+
| customer_id | first_name | last_name | DOB        | email            | phone_number | address    |
+-------------+------------+-----------+------------+------------------+--------------+------------+
|           1 | Aman       | Verma     | 2001-12-15 | aman@abc.com     |   1234567890 | 45 Saket   |
|           2 | Abhinav    | Shukla    | 2000-04-26 | abhinav@dfr.com  |   1245789630 | 78 Pawal   |
|           3 | Abhishek   | Parihar   | 2000-06-17 | abhi@ert.com     |   1245780369 | 88 Rohini  |
|           4 | Chanchal   | Gupta     | 2002-09-04 | chanchal@yui.com |   1023457896 | 56 Alwar   |
|           5 | Dimple     | Singh     | 2002-06-08 | dimple@wer.com   |   1045789632 | 44 Malhar  |
|           6 | Rishi      | Gill      | 2001-02-18 | gill@ert.com     |   1203789654 | 15 Omax    |
|           7 | Satya      | Sehgal    | 2001-01-07 | satya@wsd.com    |   1478520369 | 45 Rohini  |
|           8 | Palak      | Pal       | 2000-11-15 | palak@aqw.com    |   1457896320 | 48 Palika  |
|           9 | Tina       | Noor      | 1999-12-06 | tina@qwe.com     |   1452036987 | 71 Tilak   |
|          10 | Tarun      | Tal       | 2000-10-29 | tarun@euio.com   |   1023457896 | 55 Krishna |
+-------------+------------+-----------+------------+------------------+--------------+------------+
10 rows in set (0.00 sec)
```

• Accounts

```
mysql> insert into Accounts(account_id,customer_id,account_type,balance) values
    -> (101,1,'savings',1200),
    -> (102,2,'current',15000),
    -> (103,3,'savings',5000),
    -> (104,4,'savings',2400),
    -> (105,5,'zero_balance',0),
    -> (106,6,'savings',4500),
    -> (107,7,'current',7800),
    -> (108,8,'zero_balance',500),
    -> (109,9,'savings',4800),
    -> (110,10,'current',9000);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from Accounts;
+------------+-------------+--------------+----------+
| account_id | customer_id | account_type | balance  |
+------------+-------------+--------------+----------+
|        101 |           1 | savings      |  1200.00 |
|        102 |           2 | current      | 15000.00 |
|        103 |           3 | savings      |  5000.00 |
|        104 |           4 | savings      |  2400.00 |
|        105 |           5 | zero_balance |     0.00 |
|        106 |           6 | savings      |  4500.00 |
|        107 |           7 | current      |  7800.00 |
|        108 |           8 | zero_balance |   500.00 |
|        109 |           9 | savings      |  4800.00 |
|        110 |          10 | current      |  9000.00 |
+------------+-------------+--------------+----------+
10 rows in set (0.00 sec)
```

- Transactions

```
mysql> insert into Transactions (transaction_id,account_id,transaction_type,amount,transaction_date) values
    -> (501,101,'deposit',5000,'2024-01-05'),
    -> (502,102,'deposit',8500,'2024-12-20'),
    -> (503,103,'withdrawal',200,'2024-01-10'),
    -> (504,104,'deposit',4500,'2024-01-06'),
    -> (505,105,'transfer',100,'2024-01-08'),
    -> (506,106,'deposit',1500,'2023-12-26'),
    -> (507,107,'withdrawal',250,'2024-01-02'),
    -> (508,108,'transfer',140,'2024-12-03'),
    -> (509,109,'deposit',2600,'2024-01-16'),
    -> (510,110,'withdrawal',200,'2024-01-11');
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from Transactions;
+----------------+------------+------------------+---------+------------------+
| transaction_id | account_id | transaction_type | amount  | transaction_date |
+----------------+------------+------------------+---------+------------------+
|            501 |        101 | deposit          | 5000.00 | 2024-01-05       |
|            502 |        102 | deposit          | 8500.00 | 2024-12-20       |
|            503 |        103 | withdrawal       |  200.00 | 2024-01-10       |
|            504 |        104 | deposit          | 4500.00 | 2024-01-06       |
|            505 |        105 | transfer         |  100.00 | 2024-01-08       |
|            506 |        106 | deposit          | 1500.00 | 2023-12-26       |
|            507 |        107 | withdrawal       |  250.00 | 2024-01-02       |
|            508 |        108 | transfer         |  140.00 | 2024-12-03       |
|            509 |        109 | deposit          | 2600.00 | 2024-01-16       |
|            510 |        110 | withdrawal       |  200.00 | 2024-01-11       |
+----------------+------------+------------------+---------+------------------+
10 rows in set (0.00 sec)
```

2. Write SQL queries for the following tasks:
   1. Write a SQL query to retrieve the name, account type and email of all customers.

```
mysql> select Customers.first_name, Accounts.account_type,Customers.email from Customers join Accounts on Customers.customer_id=Accounts.customer_id;
+------------+--------------+-----------------+
| first_name | account_type | email           |
+------------+--------------+-----------------+
| Aman       | savings      | aman@abc.com    |
| Abhinav    | current      | abhinav@dfr.com |
| Abhishek   | savings      | abhi@ert.com    |
| Chanchal   | savings      | chanchal@yui.com|
| Dimple     | zero_balance | dimple@wer.com  |
| Rishi      | savings      | gill@ert.com    |
| Satya      | current      | satya@wsd.com   |
| Palak      | zero_balance | palak@aqw.com   |
| Tina       | savings      | tina@qwe.com    |
| Tarun      | current      | tarun@euio.com  |
+------------+--------------+-----------------+
10 rows in set (0.00 sec)
```

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

```
mysql> SELECT customers.customer_id, customers.first_name, transactions.transaction_id,
    ->  transactions.amount, transactions.transaction_date
    -> FROM customers
    -> join Accounts on Customers.customer_id = Accounts.customer_id
    -> join Transactions on Accounts.account_id = Transactions.account_id;
+-------------+------------+----------------+---------+------------------+
| customer_id | first_name | transaction_id | amount  | transaction_date |
+-------------+------------+----------------+---------+------------------+
|           1 | Aman       |            501 | 5000.00 | 2024-01-05       |
|           2 | Abhinav    |            502 | 8500.00 | 2024-12-20       |
|           3 | Abhishek   |            503 |  200.00 | 2024-01-10       |
|           4 | Chanchal   |            504 | 4500.00 | 2024-01-06       |
|           5 | Dimple     |            505 |  100.00 | 2024-01-08       |
|           6 | Rishi      |            506 | 1500.00 | 2023-12-26       |
|           7 | Satya      |            507 |  250.00 | 2024-01-02       |
|           8 | Palak      |            508 |  140.00 | 2024-12-03       |
|           9 | Tina       |            509 | 2600.00 | 2024-01-16       |
|          10 | Tarun      |            510 |  200.00 | 2024-01-11       |
+-------------+------------+----------------+---------+------------------+
10 rows in set (0.01 sec)
```

3. Write a SQL query to increase the balance of a specific account by a certain amount.

```
mysql> update Accounts
    -> set  balance = balance+500
    -> where account_id = 105;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Accounts;
+-------------+-------------+--------------+----------+
| account_id  | customer_id | account_type | balance  |
+-------------+-------------+--------------+----------+
|         101 |           1 | savings      |  1200.00 |
|         102 |           2 | current      | 15000.00 |
|         103 |           3 | savings      |  5000.00 |
|         104 |           4 | savings      |  2400.00 |
|         105 |           5 | zero_balance |   500.00 |
|         106 |           6 | savings      |  4500.00 |
|         107 |           7 | current      |  7800.00 |
|         108 |           8 | zero_balance |   500.00 |
|         109 |           9 | savings      |  4800.00 |
|         110 |          10 | current      |  9000.00 |
+-------------+-------------+--------------+----------+
10 rows in set (0.00 sec)
```

4. Write a SQL query to Combine first and last names of customers as a full_name.

```
mysql> select concat(first_name,' ', last_name) as full_name from Customers;
+-----------------+
| full_name       |
+-----------------+
| Aman Verma      |
| Abhinav Shukla  |
| Abhishek Parihar|
| Chanchal Gupta  |
| Dimple Singh    |
| Rishi Gill      |
| Satya Sehgal    |
| Palak Pal       |
| Tina Noor       |
| Tarun Tal       |
+-----------------+
10 rows in set (0.00 sec)
```

5. Write a SQL query to remove accounts with a balance of zero where the account type is savings.

```
mysql> delete from Accounts where balance=0 and account_type = 'savings';
Query OK, 0 rows affected (0.01 sec)
```

6. Write a SQL query to Find customers living in a specific city.

```
mysql> select * from Customers where address = '55 Krishna';
+-------------+------------+-----------+------------+---------------+--------------+------------+
| customer_id | first_name | last_name | DOB        | email         | phone_number | address    |
+-------------+------------+-----------+------------+---------------+--------------+------------+
|          10 | Tarun      | Tal       | 2000-10-29 | tarun@euio.com |   1023457896 | 55 Krishna |
+-------------+------------+-----------+------------+---------------+--------------+------------+
1 row in set (0.00 sec)
```

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

```
mysql> select balance from Accounts where account_id =102;
+-----------+
| balance   |
+-----------+
| 15000.00 |
+-----------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT *
    -> FROM Accounts
    -> WHERE account_type = 'current' AND balance > 1000.00;
+------------+-------------+--------------+-----------+
| account_id | customer_id | account_type | balance   |
+------------+-------------+--------------+-----------+
|        102 |           2 | current      | 15000.00 |
|        107 |           7 | current      |  7800.00 |
|        110 |          10 | current      |  9000.00 |
+------------+-------------+--------------+-----------+
3 rows in set (0.00 sec)
```

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

```
mysql> select * from Transactions where account_id = 105;
+----------------+------------+------------------+--------+------------------+
| transaction_id | account_id | transaction_type | amount | transaction_date |
+----------------+------------+------------------+--------+------------------+
|            505 |        105 | transfer         | 100.00 | 2024-01-08       |
+----------------+------------+------------------+--------+------------------+
1 row in set (0.01 sec)
```

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

```
mysql> -- interest rate is 5%
mysql> select account_id,balance as current_balance, (balance*1.05) - balance as accrued_interest,
    -> balance*1.05 as balance_after_interest from Accounts where account_type = 'savings';
+------------+-----------------+------------------+-----------------------+
| account_id | current_balance | accrued_interest | balance_after_interest |
+------------+-----------------+------------------+-----------------------+
|        101 |         1200.00 |          60.0000 |             1260.0000 |
|        103 |         5000.00 |         250.0000 |             5250.0000 |
|        104 |         2400.00 |         120.0000 |             2520.0000 |
|        106 |         4500.00 |         225.0000 |             4725.0000 |
|        109 |         4800.00 |         240.0000 |             5040.0000 |
+------------+-----------------+------------------+-----------------------+
5 rows in set (0.01 sec)
```

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

```
mysql> select * from Accounts where balance < 10000;
+------------+-------------+--------------+---------+
| account_id | customer_id | account_type | balance |
+------------+-------------+--------------+---------+
|        101 |           1 | savings      | 1200.00 |
|        103 |           3 | savings      | 5000.00 |
|        104 |           4 | savings      | 2400.00 |
|        105 |           5 | zero_balance |  500.00 |
|        106 |           6 | savings      | 4500.00 |
|        107 |           7 | current      | 7800.00 |
|        108 |           8 | zero_balance |  500.00 |
|        109 |           9 | savings      | 4800.00 |
|        110 |          10 | current      | 9000.00 |
+------------+-------------+--------------+---------+
9 rows in set (0.00 sec)
```

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

```
mysql> select * from Customers where address <> '45 Saket';
+-------------+------------+-----------+------------+------------------+--------------+------------+
| customer_id | first_name | last_name | DOB        | email            | phone_number | address    |
+-------------+------------+-----------+------------+------------------+--------------+------------+
|           2 | Abhinav    | Shukla    | 2000-04-26 | abhinav@dfr.com  |   1245789630 | 78 Pawal   |
|           3 | Abhishek   | Parihar   | 2000-06-17 | abhi@ert.com     |   1245780369 | 88 Rohini  |
|           4 | Chanchal   | Gupta     | 2002-09-04 | chanchal@yui.com |   1023457896 | 56 Alwar   |
|           5 | Dimple     | Singh     | 2002-06-08 | dimple@wer.com   |   1045789632 | 44 Malhar  |
|           6 | Rishi      | Gill      | 2001-02-18 | gill@ert.com     |   1203789654 | 15 Omax    |
|           7 | Satya      | Sehgal    | 2001-01-07 | satya@wsd.com    |   1478520369 | 45 Rohini  |
|           8 | Palak      | Pal       | 2000-11-15 | palak@aqw.com    |   1457896320 | 48 Palika  |
|           9 | Tina       | Noor      | 1999-12-06 | tina@qwe.com     |   1452036987 | 71 Tilak   |
|          10 | Tarun      | Tal       | 2000-10-29 | tarun@euio.com   |   1023457896 | 55 Krishna |
+-------------+------------+-----------+------------+------------------+--------------+------------+
9 rows in set (0.00 sec)
```

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

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

```
mysql> select avg(balance) as average_balance from Accounts;
+-----------------+
| average_balance |
+-----------------+
|     5070.000000 |
+-----------------+
1 row in set (0.00 sec)
```

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

```
mysql> select * from Accounts order by balance desc limit 10;
+------------+-------------+--------------+-----------+
| account_id | customer_id | account_type | balance   |
+------------+-------------+--------------+-----------+
|        102 |           2 | current      | 15000.00  |
|        110 |          10 | current      |  9000.00  |
|        107 |           7 | current      |  7800.00  |
|        103 |           3 | savings      |  5000.00  |
|        109 |           9 | savings      |  4800.00  |
|        106 |           6 | savings      |  4500.00  |
|        104 |           4 | savings      |  2400.00  |
|        101 |           1 | savings      |  1200.00  |
|        105 |           5 | zero_balance |   500.00  |
|        108 |           8 | zero_balance |   500.00  |
+------------+-------------+--------------+-----------+
10 rows in set (0.01 sec)
```

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

```
mysql> SELECT Customers.customer_id, Customers.first_name, SUM(Transactions.amount)
 AS total_deposits
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id
    -> JOIN Transactions ON Accounts.account_id = Transactions.account_id
    -> WHERE Transactions.transaction_type = 'deposit' AND Transactions.transaction
_date = '2024-01-16'
    -> GROUP BY Customers.customer_id, Customers.first_name;
+-------------+------------+----------------+
| customer_id | first_name | total_deposits |
+-------------+------------+----------------+
|           9 | Tina       |        2600.00 |
+-------------+------------+----------------+
1 row in set (0.01 sec)
```

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

```
mysql> select customer_id, first_name, DOB from Customers
    -> order by DOB asc
    -> limit 1;
+-------------+------------+------------+
| customer_id | first_name | DOB        |
+-------------+------------+------------+
|           9 | Tina       | 1999-12-06 |
+-------------+------------+------------+
1 row in set (0.00 sec)

mysql> select customer_id, first_name, DOB from Customers
    -> order by DOB desc
    -> limit 1;
+-------------+------------+------------+
| customer_id | first_name | DOB        |
+-------------+------------+------------+
|           4 | Chanchal   | 2002-09-04 |
+-------------+------------+------------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT transactions.transaction_id, transactions.amount, transactions.transaction_date,accounts.account_type
    -> FROM transactions
    -> JOIN accounts ON transactions.account_id = accounts.account_id;
+----------------+---------+------------------+--------------+
| transaction_id | amount  | transaction_date | account_type |
+----------------+---------+------------------+--------------+
|            501 | 5000.00 | 2024-01-05       | savings      |
|            502 | 8500.00 | 2024-12-20       | current      |
|            503 |  200.00 | 2024-01-10       | savings      |
|            504 | 4500.00 | 2024-01-06       | savings      |
|            505 |  100.00 | 2024-01-08       | zero_balance |
|            506 | 1500.00 | 2023-12-26       | savings      |
|            507 |  250.00 | 2024-01-02       | current      |
|            508 |  140.00 | 2024-12-03       | zero_balance |
|            509 | 2600.00 | 2024-01-16       | savings      |
|            510 |  200.00 | 2024-01-11       | current      |
+----------------+---------+------------------+--------------+
10 rows in set (0.00 sec)
```

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

```
mysql> SELECT Customers.customer_id, Customers.first_name, Customers.email, Accounts.account_id, Accounts.account_type, Accounts.balance
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id;
+-------------+------------+-----------------+------------+--------------+----------+
| customer_id | first_name | email           | account_id | account_type | balance  |
+-------------+------------+-----------------+------------+--------------+----------+
|           1 | Aman       | aman@abc.com    |        101 | savings      |  1200.00 |
|           2 | Abhinav    | abhinav@dfr.com |        102 | current      | 15000.00 |
|           3 | Abhishek   | abhi@ert.com    |        103 | savings      |  5000.00 |
|           4 | Chanchal   | chanchal@yui.com|        104 | savings      |  2400.00 |
|           5 | Dimple     | dimple@wer.com  |        105 | zero_balance |   500.00 |
|           6 | Rishi      | gill@ert.com    |        106 | savings      |  4500.00 |
|           7 | Satya      | satya@wsd.com   |        107 | current      |  7800.00 |
|           8 | Palak      | palak@aqw.com   |        108 | zero_balance |   500.00 |
|           9 | Tina       | tina@qwe.com    |        109 | savings      |  4800.00 |
|          10 | Tarun      | tarun@euio.com  |        110 | current      |  9000.00 |
+-------------+------------+-----------------+------------+--------------+----------+
10 rows in set (0.00 sec)
```

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

```
mysql> SELECT Customers.customer_id, Customers.first_name, Customers.email,
    -> Transactions.transaction_id, Transactions.transaction_type, Transactions.amount, Transactions.transaction_date
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id
    -> JOIN Transactions ON Accounts.account_id = Transactions.account_id;
+-------------+------------+-----------------+----------------+------------------+---------+------------------+
| customer_id | first_name | email           | transaction_id | transaction_type | amount  | transaction_date |
+-------------+------------+-----------------+----------------+------------------+---------+------------------+
|           1 | Aman       | aman@abc.com    |            501 | deposit          | 5000.00 | 2024-01-05       |
|           2 | Abhinav    | abhinav@dfr.com |            502 | deposit          | 8500.00 | 2024-12-20       |
|           3 | Abhishek   | abhi@ert.com    |            503 | withdrawal       |  200.00 | 2024-01-10       |
|           4 | Chanchal   | chanchal@yui.com|            504 | deposit          | 4500.00 | 2024-01-06       |
|           5 | Dimple     | dimple@wer.com  |            505 | transfer         |  100.00 | 2024-01-08       |
|           6 | Rishi      | gill@ert.com    |            506 | deposit          | 1500.00 | 2023-12-26       |
|           7 | Satya      | satya@wsd.com   |            507 | withdrawal       |  250.00 | 2024-01-02       |
|           8 | Palak      | palak@aqw.com   |            508 | transfer         |  140.00 | 2024-12-03       |
|           9 | Tina       | tina@qwe.com    |            509 | deposit          | 2600.00 | 2024-01-16       |
|          10 | Tarun      | tarun@euio.com  |            510 | withdrawal       |  200.00 | 2024-01-11       |
+-------------+------------+-----------------+----------------+------------------+---------+------------------+
10 rows in set (0.00 sec)
```

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

```
mysql> SELECT Customers.customer_id, Customers.first_name, Customers.email, COUNT(Accounts.account_id) AS num_accounts
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id
    -> GROUP BY Customers.customer_id, Customers.first_name, Customers.email
    -> HAVING COUNT(Accounts.account_id) > 1;
Empty set (0.01 sec)
```

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

```
mysql> SELECT account_id,
    ->        SUM(CASE WHEN transaction_type = 'deposit' THEN amount ELSE 0 END) AS total_deposits,
    ->        SUM(CASE WHEN transaction_type = 'withdrawal' THEN amount ELSE 0 END) AS total_withdrawals,
    ->        SUM(CASE WHEN transaction_type = 'deposit' THEN amount ELSE -amount END) AS difference
    -> FROM Transactions
    -> GROUP BY account_id;
+------------+----------------+-------------------+------------+
| account_id | total_deposits | total_withdrawals | difference |
+------------+----------------+-------------------+------------+
|        101 |        5000.00 |              0.00 |    5000.00 |
|        102 |        8500.00 |              0.00 |    8500.00 |
|        103 |           0.00 |            200.00 |    -200.00 |
|        104 |        4500.00 |              0.00 |    4500.00 |
|        105 |           0.00 |              0.00 |    -100.00 |
|        106 |        1500.00 |              0.00 |    1500.00 |
|        107 |           0.00 |            250.00 |    -250.00 |
|        108 |           0.00 |              0.00 |    -140.00 |
|        109 |        2600.00 |              0.00 |    2600.00 |
|        110 |           0.00 |            200.00 |    -200.00 |
+------------+----------------+-------------------+------------+
10 rows in set (0.01 sec)
```

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

```
mysql> select Transactions.account_id, avg(balance) as DailyAvgBalance from Accounts
    -> join transactions on Transactions.account_id = Accounts.account_id
    -> where transaction_date between '2024-01-02' and '2024-01-09'
    -> group by account_id;
+------------+-----------------+
| account_id | DailyAvgBalance |
+------------+-----------------+
|        101 |     1200.000000 |
|        104 |     2400.000000 |
|        105 |      500.000000 |
|        107 |     7800.000000 |
+------------+-----------------+
4 rows in set (0.01 sec)
```

11. Calculate the total balance for each account type.

```
mysql> select account_type,sum(balance) as total_balance from Accounts group by account_type;
+--------------+---------------+
| account_type | total_balance |
+--------------+---------------+
| savings      |      17900.00 |
| current      |      31800.00 |
| zero_balance |       1000.00 |
+--------------+---------------+
3 rows in set (0.01 sec)
```

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

```
mysql> SELECT account_id, COUNT(transaction_id) AS transaction_count
    -> FROM Transactions
    -> GROUP BY account_id
    -> ORDER BY transaction_count DESC;
+------------+-------------------+
| account_id | transaction_count |
+------------+-------------------+
|        101 |                 1 |
|        102 |                 1 |
|        103 |                 1 |
|        104 |                 1 |
|        105 |                 1 |
|        106 |                 1 |
|        107 |                 1 |
|        108 |                 1 |
|        109 |                 1 |
|        110 |                 1 |
+------------+-------------------+
10 rows in set (0.01 sec)
```

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

```
mysql> SELECT
    ->     Customers.customer_id,
    ->     Customers.first_name,
    ->     Accounts.account_type,
    ->     SUM(Accounts.balance) AS total_balance
    -> FROM
    ->     Customers
    -> JOIN
    ->     Accounts ON Customers.customer_id = Accounts.customer_id
    -> GROUP BY
    ->     Customers.customer_id, Customers.first_name, Accounts.account_type
    -> ORDER BY
    ->     total_balance DESC;
+-------------+------------+--------------+---------------+
| customer_id | first_name | account_type | total_balance |
+-------------+------------+--------------+---------------+
|           2 | Abhinav    | current      |      15000.00 |
|          10 | Tarun      | current      |       9000.00 |
|           7 | Satya      | current      |       7800.00 |
|           3 | Abhishek   | savings      |       5000.00 |
|           9 | Tina       | savings      |       4800.00 |
|           6 | Rishi      | savings      |       4500.00 |
|           4 | Chanchal   | savings      |       2400.00 |
|           1 | Aman       | savings      |       1200.00 |
|           5 | Dimple     | zero_balance |        500.00 |
|           8 | Palak      | zero_balance |        500.00 |
+-------------+------------+--------------+---------------+
10 rows in set (0.01 sec)
```

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

```
mysql> SELECT
    ->     amount,
    ->     transaction_date,
    ->     account_id,
    ->     COUNT(transaction_id) AS duplicate_count
    -> FROM
    ->     Transactions
    -> GROUP BY
    ->     amount, transaction_date, account_id
    -> HAVING
    ->     COUNT(transaction_id) > 1;
Empty set (0.01 sec)
```

Tasks 4: Subquery and its type:

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

```
mysql> SELECT
    ->     Customers.customer_id,
    ->     Customers.first_name,
    ->     MAX(Accounts.balance) AS highest_balance
    -> FROM
    ->     Customers
    -> JOIN
    ->     Accounts ON Customers.customer_id = Accounts.customer_id
    -> GROUP BY
    ->     Customers.customer_id, Customers.first_name
    -> ORDER BY
    ->     highest_balance DESC
    -> LIMIT 1;
+-------------+------------+-----------------+
| customer_id | first_name | highest_balance |
+-------------+------------+-----------------+
|           2 | Abhinav    |        15000.00 |
+-------------+------------+-----------------+
1 row in set (0.01 sec)
```

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

```
mysql> SELECT
    ->     Customers.customer_id,
    ->     Customers.first_name,
    ->     AVG(Accounts.balance) AS average_balance
    -> FROM
    ->     Customers
    -> JOIN
    ->     Accounts ON Customers.customer_id = Accounts.customer_id
    -> GROUP BY
    ->     Customers.customer_id, Customers.first_name
    -> HAVING
    ->     COUNT(Accounts.account_id) > 1;
Empty set (0.01 sec)
```

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

```
mysql> SELECT accounts.account_id, accounts.account_type, transactions.transaction_id, transactions.amount
    -> FROM accounts
    -> JOIN transactions ON accounts.account_id = transactions.account_id
    -> WHERE transactions.amount > (SELECT AVG(amount) FROM transactions);
+------------+--------------+----------------+---------+
| account_id | account_type | transaction_id | amount  |
+------------+--------------+----------------+---------+
|        101 | savings      |            501 | 5000.00 |
|        102 | current      |            502 | 8500.00 |
|        104 | savings      |            504 | 4500.00 |
|        109 | savings      |            509 | 2600.00 |
+------------+--------------+----------------+---------+
4 rows in set (0.01 sec)
```

4. Identify customers who have no recorded transactions.

```
mysql> SELECT Customers.customer_id,Customers.first_name
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id
    -> join Transactions on Accounts.account_id = Transactions.account_id
    -> WHERE Transactions.transaction_id IS NULL;
Empty set (0.00 sec)
```

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

```
mysql> SELECT a.account_id, a.account_type, a.balance
    -> FROM Accounts a
    -> WHERE NOT EXISTS (SELECT 1 FROM Transactions t WHERE t.account_id = a.account_id);
Empty set (0.00 sec)
```

6. Retrieve transactions for accounts with the lowest balance.

```
mysql> SELECT t.transaction_id, t.account_id, t.amount
    -> FROM Transactions t
    -> JOIN (SELECT account_id FROM Accounts ORDER BY balance LIMIT 1) AS lowest_balance_accounts ON t.account
_id = lowest_balance_accounts.account_id;
+----------------+------------+--------+
| transaction_id | account_id | amount |
+----------------+------------+--------+
|            505 |        105 | 100.00 |
+----------------+------------+--------+
1 row in set (0.01 sec)
```

7. Identify customers who have accounts of multiple types.

```
mysql> SELECT
    ->     customer_id,
    ->     first_name
    -> FROM Customers
    -> WHERE
    ->     customer_id IN (SELECT
    ->                 customer_id
    ->             FROM Accounts
    ->             GROUP BY customer_id
    ->             HAVING COUNT(DISTINCT account_type) > 1
    ->     );
Empty set (0.01 sec)
```

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

```
mysql> SELECT
    ->     account_type,
    ->     COUNT(*) AS account_count,
    ->     (COUNT(*) * 100.0 / (SELECT COUNT(*) FROM Accounts)) AS percentage
    -> FROM
    ->     Accounts
    -> GROUP BY
    ->     account_type;
+--------------+---------------+------------+
| account_type | account_count | percentage |
+--------------+---------------+------------+
| savings      |             5 |   50.00000 |
| current      |             3 |   30.00000 |
| zero_balance |             2 |   20.00000 |
+--------------+---------------+------------+
3 rows in set (0.03 sec)
```

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

```
mysql> SELECT transactions.transaction_id, transactions.amount, transactions.transaction_date
    -> FROM transactions
    -> join accounts on accounts.account_id = transactions.account_id
    -> join customers ON accounts.customer_id = customers.customer_id
    -> WHERE customers.customer_id = '2';
+----------------+---------+------------------+
| transaction_id | amount  | transaction_date |
+----------------+---------+------------------+
|            502 | 8500.00 | 2024-12-20       |
+----------------+---------+------------------+
1 row in set (0.00 sec)
```

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

```
mysql> use HMBank;
Database changed
mysql> SELECT
    ->     account_type,
    ->     (SELECT SUM(balance) FROM accounts a2 WHERE a2.account_type = a1.account_type) as total_balance
    -> FROM
    ->     accounts a1
    -> GROUP BY
    ->     account_type;
+--------------+---------------+
| account_type | total_balance |
+--------------+---------------+
| savings      |      17900.00 |
| current      |      31800.00 |
| zero_balance |       1000.00 |
+--------------+---------------+
3 rows in set (0.05 sec)
```