ASSIGNMENT-3

NAME: SHIVAM SINGH

TASK1:

Database Design:

1. Create the database named "HMBank"

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

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

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

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

1.

```
mysql> CREATE DATABASE HMBANK;
Query OK, 1 row affected (0.01 sec)
```

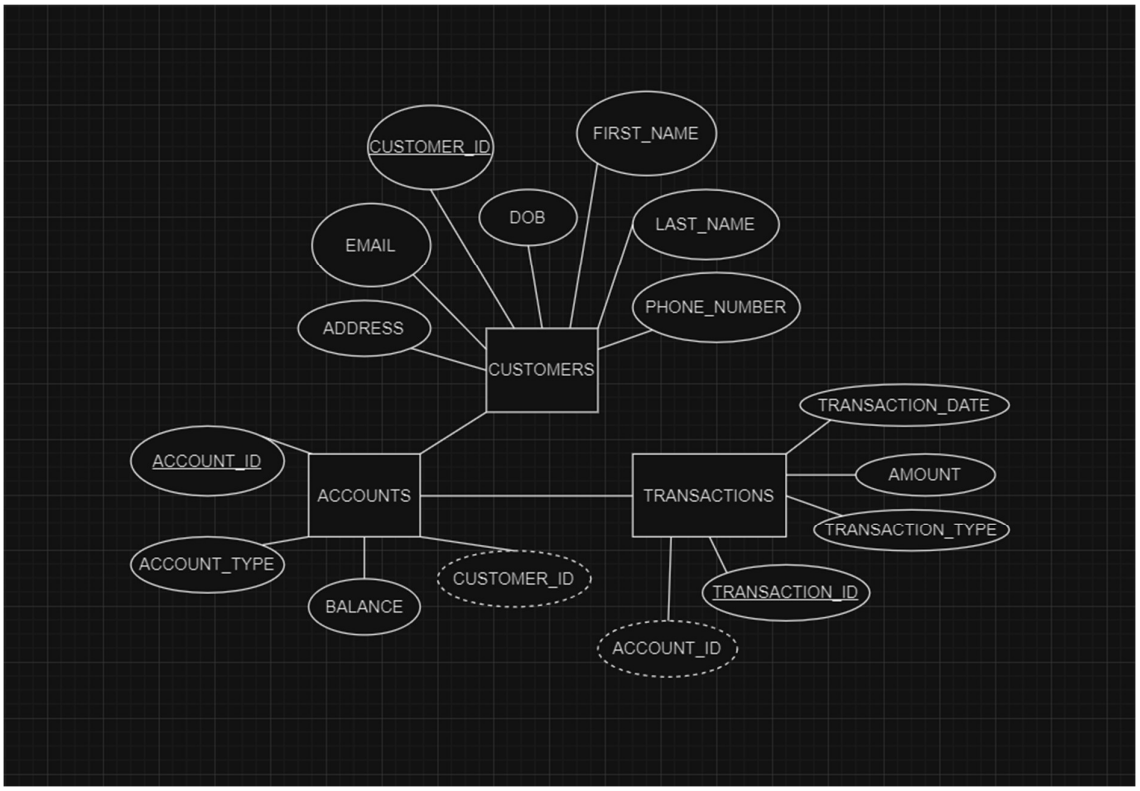
2.

```
mysql> CREATE TABLE Customers (
    ->     customer_id INT PRIMARY KEY,
    ->     first_name VARCHAR(50),
    ->     last_name VARCHAR(50),
    ->     DOB DATE,
    ->     email VARCHAR(100),
    ->     phone_number VARCHAR(20),
    ->     address VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> CREATE TABLE Accounts (
    ->     account_id INT PRIMARY KEY,
    ->     customer_id INT,
    ->     account_type VARCHAR(50),
    ->     balance DECIMAL(10, 2),
    ->     FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> CREATE TABLE Transactions (
    ->     transaction_id INT PRIMARY KEY,
    ->     account_id INT,
    ->     transaction_type VARCHAR(50),
    ->     amount DECIMAL(10, 2),
    ->     transaction_date DATE,
    ->     FOREIGN KEY (account_id) REFERENCES Accounts(account_id)
    -> );
Query OK, 0 rows affected (0.07 sec)
```

4.



6.

```
mysql> CREATE TABLE Customers (
    ->     customer_id INT PRIMARY KEY,
    ->     first_name VARCHAR(50),
    ->     last_name VARCHAR(50),
    ->     DOB DATE,
    ->     email VARCHAR(100),
    ->     phone_number VARCHAR(20),
    ->     address VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> CREATE TABLE Accounts (
    ->     account_id INT PRIMARY KEY,
    ->     customer_id INT,
    ->     account_type VARCHAR(50),
    ->     balance DECIMAL(10, 2),
    ->     FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> CREATE TABLE Transactions (
    ->     transaction_id INT PRIMARY KEY,
    ->     account_id INT,
    ->     transaction_type VARCHAR(50),
    ->     amount DECIMAL(10, 2),
    ->     transaction_date DATE,
    ->     FOREIGN KEY (account_id) REFERENCES Accounts(account_id)
    -> );
Query OK, 0 rows affected (0.07 sec)
```

TASK 2:

Select, Where, Between, AND, LIKE:

1. Insert at least 10 sample records into each of the following tables. • Customers • Accounts • Transactions

2. Write SQL queries for the following tasks:

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

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

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

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

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

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

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

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

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

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

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

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

1.

CUSTOMERS:

```
mysql>
mysql> SELECT*FROM CUSTOMERS;
+-------------+------------+-----------+------------+-------------------------+--------------+------------------+
| customer_id | first_name | last_name | DOB        | email                   | phone_number | address          |
+-------------+------------+-----------+------------+-------------------------+--------------+------------------+
|           1 | John       | Doe       | 1990-05-15 | john.doe@example.com    | 123-456-7890 | 123 Main St      |
|           2 | Jane       | Smith     | 1985-08-22 | jane.smith@example.com  | 987-654-3210 | 456 Oak Ave      |
|           3 | Mike       | Johnson   | 1995-02-10 | mike.johnson@example.com| 555-123-4567 | 789 Pine Rd      |
|           4 | Emily      | Brown     | 1982-11-28 | emily.brown@example.com | 111-222-3333 | 876 Elm Ln       |
|           5 | Chris      | Miller    | 1998-07-03 | chris.miller@example.com| 999-888-7777 | 543 Birch Blvd   |
|           6 | Sarah      | Clark     | 1993-04-18 | sarah.clark@example.com | 777-666-5555 | 234 Cedar St     |
|           7 | David      | Hall      | 1989-09-12 | david.hall@example.com  | 444-555-6666 | 321 Maple Ave    |
|           8 | Amy        | White     | 1991-12-30 | amy.white@example.com   | 333-444-5555 | 678 Pine Ln      |
|           9 | Brian      | Taylor    | 1987-06-25 | brian.taylor@example.com| 222-333-4444 | 987 Oak Blvd     |
|          10 | Laura      | Wilson    | 1996-03-08 | laura.wilson@example.com| 666-777-8888 | 876 Birch Rd     |
|          11 | Shivam     | Singh     | 2000-09-24 | shivasingh414@gmail.com | 8340508631   | Bokaro Steel City|
+-------------+------------+-----------+------------+-------------------------+--------------+------------------+
11 rows in set (0.00 sec)
```

ACCOUNTS & TRANSACTIONS:

```
mysql> SELECT*FROM ACCOUNTS;
+------------+-------------+--------------+----------+
| account_id | customer_id | account_type | balance  |
+------------+-------------+--------------+----------+
|          1 |           1 | savings      |  5000.00 |
|          2 |           2 | current      |  2500.50 |
|          3 |           3 | savings      | 10000.75 |
|          4 |           4 | zero_balance |     0.00 |
|          5 |           5 | current      |  7500.25 |
|          6 |           6 | savings      |  3000.50 |
|          7 |           7 | current      |  6000.75 |
|          8 |           8 | savings      | 12000.00 |
|          9 |           9 | savings      |  8000.25 |
|         10 |          10 | current      |  4000.50 |
|         11 |          11 | savings      |  2000.00 |
+------------+-------------+--------------+----------+
11 rows in set (0.00 sec)

mysql> SELECT*FROM TRANSACTIONS;
+----------------+------------+------------------+---------+------------------+
| transaction_id | account_id | transaction_type | amount  | transaction_date |
+----------------+------------+------------------+---------+------------------+
|              1 |          1 | deposit          | 1000.00 | 2024-01-01       |
|              2 |          2 | withdrawal       |  500.25 | 2024-01-02       |
|              3 |          3 | deposit          | 2000.50 | 2024-01-03       |
|              4 |          4 | withdrawal       |  100.75 | 2024-01-04       |
|              5 |          5 | transfer         | 1500.25 | 2024-01-05       |
|              6 |          6 | deposit          |  800.50 | 2024-01-06       |
|              7 |          7 | withdrawal       | 1200.75 | 2024-01-07       |
|              8 |          8 | transfer         | 3000.00 | 2024-01-08       |
|              9 |          9 | deposit          | 1000.25 | 2024-01-09       |
|             10 |         10 | withdrawal       |  400.50 | 2024-01-10       |
|             11 |         11 | deposit          |  500.75 | 2024-01-11       |
+----------------+------------+------------------+---------+------------------+
11 rows in set (0.00 sec)
```

2.

1.

```
mysql> SELECT
    ->     CONCAT(first_name, ' ', last_name) AS customer_name,
    ->     Accounts.account_type,
    ->     Customers.email
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id;
+---------------+--------------+---------------------------+
| customer_name | account_type | email                     |
+---------------+--------------+---------------------------+
| John Doe      | savings      | john.doe@example.com      |
| Jane Smith    | current      | jane.smith@example.com    |
| Mike Johnson  | savings      | mike.johnson@example.com  |
| Emily Brown   | zero_balance | emily.brown@example.com   |
| Chris Miller  | current      | chris.miller@example.com  |
| Sarah Clark   | savings      | sarah.clark@example.com   |
| David Hall    | current      | david.hall@example.com    |
| Amy White     | savings      | amy.white@example.com     |
| Brian Taylor  | savings      | brian.taylor@example.com  |
| Laura Wilson  | current      | laura.wilson@example.com  |
| Shivam Singh  | savings      | shivasingh414@gmail.com   |
+---------------+--------------+---------------------------+
11 rows in set (0.00 sec)
```

2.

```
mysql> SELECT
    ->     Transactions.transaction_id,
    ->     Customers.customer_id,
    ->     CONCAT(Customers.first_name, ' ', Customers.last_name) AS customer_name,
    ->     Transactions.transaction_type,
    ->     Transactions.amount,
    ->     Transactions.transaction_date
    -> FROM Transactions
    -> JOIN Accounts ON Transactions.account_id = Accounts.account_id
    -> JOIN Customers ON Accounts.customer_id = Customers.customer_id;
+----------------+-------------+---------------+------------------+---------+------------------+
| transaction_id | customer_id | customer_name | transaction_type | amount  | transaction_date |
+----------------+-------------+---------------+------------------+---------+------------------+
|              1 |           1 | John Doe      | deposit          | 1000.00 | 2024-01-01       |
|              2 |           2 | Jane Smith    | withdrawal       |  500.25 | 2024-01-02       |
|              3 |           3 | Mike Johnson  | deposit          | 2000.50 | 2024-01-03       |
|              4 |           4 | Emily Brown   | withdrawal       |  100.75 | 2024-01-04       |
|              5 |           5 | Chris Miller  | transfer         | 1500.25 | 2024-01-05       |
|              6 |           6 | Sarah Clark   | deposit          |  800.50 | 2024-01-06       |
|              7 |           7 | David Hall    | withdrawal       | 1200.75 | 2024-01-07       |
|              8 |           8 | Amy White     | transfer         | 3000.00 | 2024-01-08       |
|              9 |           9 | Brian Taylor  | deposit          | 1000.25 | 2024-01-09       |
|             10 |          10 | Laura Wilson  | withdrawal       |  400.50 | 2024-01-10       |
|             11 |          11 | Shivam Singh  | deposit          |  500.75 | 2024-01-11       |
+----------------+-------------+---------------+------------------+---------+------------------+
11 rows in set (0.00 sec)
```

3.

```
mysql> UPDATE Accounts
    -> SET balance = balance + 1000000.00
    -> WHERE account_id = 11;
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    |
+------------+-------------+--------------+------------+
|          1 |           1 | savings      |    5000.00 |
|          2 |           2 | current      |    2500.50 |
|          3 |           3 | savings      |   10000.75 |
|          4 |           4 | zero_balance |       0.00 |
|          5 |           5 | current      |    7500.25 |
|          6 |           6 | savings      |    3000.50 |
|          7 |           7 | current      |    6000.75 |
|          8 |           8 | savings      |   12000.00 |
|          9 |           9 | savings      |    8000.25 |
|         10 |          10 | current      |    4000.50 |
|         11 |          11 | savings      | 1002000.00 |
+------------+-------------+--------------+------------+
11 rows in set (0.00 sec)
```

4.

```
mysql> SELECT
    ->     customer_id,
    ->     CONCAT(first_name, ' ', last_name) AS full_name
    -> FROM Customers;
+-------------+--------------+
| customer_id | full_name    |
+-------------+--------------+
|           1 | John Doe     |
|           2 | Jane Smith   |
|           3 | Mike Johnson |
|           4 | Emily Brown  |
|           5 | Chris Miller |
|           6 | Sarah Clark  |
|           7 | David Hall   |
|           8 | Amy White    |
|           9 | Brian Taylor |
|          10 | Laura Wilson |
|          11 | Shivam Singh |
+-------------+--------------+
11 rows in set (0.00 sec)
```

5.

```
mysql> DELETE FROM Accounts
    -> WHERE balance = 0.00 AND account_type = 'savings';
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT*FROM ACCOUNTS;
+------------+-------------+--------------+------------+
| account_id | customer_id | account_type | balance    |
+------------+-------------+--------------+------------+
|          1 |           1 | savings      |    5000.00 |
|          2 |           2 | current      |    2500.50 |
|          3 |           3 | savings      |   10000.75 |
|          4 |           4 | zero_balance |       0.00 |
|          5 |           5 | current      |    7500.25 |
|          6 |           6 | savings      |    3000.50 |
|          7 |           7 | current      |    6000.75 |
|          8 |           8 | savings      |   12000.00 |
|          9 |           9 | savings      |    8000.25 |
|         10 |          10 | current      |    4000.50 |
|         11 |          11 | savings      | 1002000.00 |
+------------+-------------+--------------+------------+
11 rows in set (0.00 sec)
```

6.

```
mysql> SELECT *
    -> FROM Customers
    -> WHERE address LIKE '%BOKARO STEEL CITY%';
+-------------+------------+-----------+------------+-----------------------+--------------+------------------+
| customer_id | first_name | last_name | DOB        | email                 | phone_number | address          |
+-------------+------------+-----------+------------+-----------------------+--------------+------------------+
|          11 | Shivam     | Singh     | 2000-09-24 | shivasingh414@gmail.com | 8340508631   | Bokaro Steel City |
+-------------+------------+-----------+------------+-----------------------+--------------+------------------+
1 row in set (0.01 sec)
```

7.

```
mysql> SELECT balance
    -> FROM Accounts
    -> WHERE account_id = 4;
+----------+
| balance  |
+----------+
|     0.00 |
+----------+
1 row in set (0.00 sec)
```

8.

```
mysql> SELECT *
    -> FROM Accounts
    -> WHERE account_type = 'current' AND balance > 1000.00;
+------------+-------------+--------------+---------+
| account_id | customer_id | account_type | balance |
+------------+-------------+--------------+---------+
|          2 |           2 | current      | 2500.50 |
|          5 |           5 | current      | 7500.25 |
|          7 |           7 | current      | 6000.75 |
|         10 |          10 | current      | 4000.50 |
+------------+-------------+--------------+---------+
4 rows in set (0.00 sec)
```

9.

```
mysql> SELECT *
    -> FROM Transactions
    -> WHERE account_id = 11;
+----------------+------------+------------------+--------+------------------+
| transaction_id | account_id | transaction_type | amount | transaction_date |
+----------------+------------+------------------+--------+------------------+
|             11 |         11 | deposit          | 500.75 | 2024-01-11       |
+----------------+------------+------------------+--------+------------------+
1 row in set (0.00 sec)
```

10.

```
ERROR 1054 (42S22): Unknown column 'interest_rate' in 'field list'
mysql> SELECT
    ->     account_id,
    ->     account_type,
    ->     balance,
    ->     3.5 AS interest_rate, -- Fixed interest rate
    ->     (balance * 3.5 / 100) AS interest_accrued
    -> FROM Accounts
    -> WHERE account_type = 'savings';
+------------+--------------+------------+---------------+------------------+
| account_id | account_type | balance    | interest_rate | interest_accrued |
+------------+--------------+------------+---------------+------------------+
|          1 | savings      |    5000.00 |           3.5 |      175.0000000 |
|          3 | savings      |   10000.75 |           3.5 |      350.0262500 |
|          6 | savings      |    3000.50 |           3.5 |      105.0175000 |
|          8 | savings      |   12000.00 |           3.5 |      420.0000000 |
|          9 | savings      |    8000.25 |           3.5 |      280.0087500 |
|         11 | savings      | 1002000.00 |           3.5 |    35070.0000000 |
+------------+--------------+------------+---------------+------------------+
5 rows in set (0.00 sec)
```

11.

```
mysql> SELECT* FROM ACCOUNTS
    -> WHERE BALANCE > 10000;
+-------------+-------------+--------------+------------+
| account_id  | customer_id | account_type | balance    |
+-------------+-------------+--------------+------------+
|           3 |           3 | savings      |   10000.75 |
|           8 |           8 | savings      |   12000.00 |
|          11 |          11 | savings      | 1002000.00 |
+-------------+-------------+--------------+------------+
3 rows in set (0.00 sec)
```

12.

```
mysql> SELECT *
    -> FROM Customers
    -> WHERE LOWER(address) NOT LIKE '%bokaro steel city%';
+-------------+------------+-----------+------------+-------------------------+--------------+---------------+
| customer_id | first_name | last_name | DOB        | email                   | phone_number | address       |
+-------------+------------+-----------+------------+-------------------------+--------------+---------------+
|           1 | John       | Doe       | 1990-05-15 | john.doe@example.com    | 123-456-7890 | 123 Main St   |
|           2 | Jane       | Smith     | 1985-08-22 | jane.smith@example.com  | 987-654-3210 | 456 Oak Ave   |
|           3 | Mike       | Johnson   | 1995-02-10 | mike.johnson@example.com| 555-123-4567 | 789 Pine Rd   |
|           4 | Emily      | Brown     | 1982-11-28 | emily.brown@example.com | 111-222-3333 | 876 Elm Ln    |
|           5 | Chris      | Miller    | 1998-07-03 | chris.miller@example.com| 999-888-7777 | 543 Birch Blvd|
|           6 | Sarah      | Clark     | 1993-04-18 | sarah.clark@example.com | 777-666-5555 | 234 Cedar St  |
|           7 | David      | Hall      | 1989-09-12 | david.hall@example.com  | 444-555-6666 | 321 Maple Ave |
|           8 | Amy        | White     | 1991-12-30 | amy.white@example.com   | 333-444-5555 | 678 Pine Ln   |
|           9 | Brian      | Taylor    | 1987-06-25 | brian.taylor@example.com| 222-333-4444 | 987 Oak Blvd  |
|          10 | Laura      | Wilson    | 1996-03-08 | laura.wilson@example.com| 666-777-8888 | 876 Birch Rd  |
+-------------+------------+-----------+------------+-------------------------+--------------+---------------+
10 rows in set (0.04 sec)
```

TASK 3:

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

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

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

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

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

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

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

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

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

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

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

11. Calculate the total balance for each account type.

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

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

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

1.

```
mysql> SELECT AVG(balance) AS average_balance
    -> FROM Accounts;
+-----------------+
| average_balance |
+-----------------+
|    96363.954545 |
+-----------------+
1 row in set (0.00 sec)
```

2.

```
mysql> SELECT *
    -> FROM Accounts
    -> ORDER BY balance DESC
    -> LIMIT 10;
+------------+-------------+--------------+------------+
| account_id | customer_id | account_type | balance    |
+------------+-------------+--------------+------------+
|         11 |          11 | savings      | 1002000.00 |
|          8 |           8 | savings      |   12000.00 |
|          3 |           3 | savings      |   10000.75 |
|          9 |           9 | savings      |    8000.25 |
|          5 |           5 | current      |    7500.25 |
|          7 |           7 | current      |    6000.75 |
|          1 |           1 | savings      |    5000.00 |
|         10 |          10 | current      |    4000.50 |
|          6 |           6 | savings      |    3000.50 |
|          2 |           2 | current      |    2500.50 |
+------------+-------------+--------------+------------+
10 rows in set (0.00 sec)
```

3.

```
mysql> SELECT
    ->     Customers.customer_id,
    ->     CONCAT(Customers.first_name, ' ', Customers.last_name) AS customer_name,
    ->     SUM(CASE WHEN Transactions.transaction_type = 'deposit' AND Transactions.transaction_date = '
2024-01-05' THEN Transactions.amount ELSE 0 END) AS total_deposits
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id
    -> JOIN Transactions ON Accounts.account_id = Transactions.account_id
    -> GROUP BY Customers.customer_id, customer_name;
+-------------+---------------+----------------+
| customer_id | customer_name | total_deposits |
+-------------+---------------+----------------+
|           1 | John Doe      |           0.00 |
|           2 | Jane Smith    |           0.00 |
|           3 | Mike Johnson  |           0.00 |
|           4 | Emily Brown   |           0.00 |
|           5 | Chris Miller  |           0.00 |
|           6 | Sarah Clark   |           0.00 |
|           7 | David Hall    |           0.00 |
|           8 | Amy White     |           0.00 |
|           9 | Brian Taylor  |           0.00 |
|          10 | Laura Wilson  |           0.00 |
|          11 | Shivam Singh  |           0.00 |
+-------------+---------------+----------------+
```

4.

```
mysql> SELECT
    ->     customer_id,
    ->     first_name,
    ->     last_name,
    ->     DOB
    -> FROM Customers
    -> ORDER BY DOB ASC
    -> LIMIT 1;
+-------------+------------+-----------+------------+
| customer_id | first_name | last_name | DOB        |
+-------------+------------+-----------+------------+
|           4 | Emily      | Brown     | 1982-11-28 |
+-------------+------------+-----------+------------+
1 row in set (0.00 sec)

mysql> SELECT
    ->     customer_id,
    ->     first_name,
    ->     last_name,
    ->     DOB
    -> FROM Customers
    -> ORDER BY DOB DESC
    -> LIMIT 1;
+-------------+------------+-----------+------------+
| customer_id | first_name | last_name | DOB        |
+-------------+------------+-----------+------------+
|          11 | Shivam     | Singh     | 2000-09-24 |
+-------------+------------+-----------+------------+
1 row in set (0.00 sec)
```

5.

```
ysql> SELECT
   ->     Transactions.transaction_id,
   ->     Transactions.account_id,
   ->     Transactions.transaction_type,
   ->     Transactions.amount,
   ->     Transactions.transaction_date,
   ->     Accounts.account_type
   -> FROM Transactions
   -> JOIN Accounts ON Transactions.account_id = Accounts.account_id;
+---------------+------------+------------------+---------+------------------+--------------+
| transaction_id | account_id | transaction_type | amount  | transaction_date | account_type |
+---------------+------------+------------------+---------+------------------+--------------+
|             1 |          1 | deposit          | 1000.00 | 2024-01-01       | savings      |
|             2 |          2 | withdrawal       |  500.25 | 2024-01-02       | current      |
|             3 |          3 | deposit          | 2000.50 | 2024-01-03       | savings      |
|             4 |          4 | withdrawal       |  100.75 | 2024-01-04       | zero_balance |
|             5 |          5 | transfer         | 1500.25 | 2024-01-05       | current      |
|             6 |          6 | deposit          |  800.50 | 2024-01-06       | savings      |
|             7 |          7 | withdrawal       | 1200.75 | 2024-01-07       | current      |
|             8 |          8 | transfer         | 3000.00 | 2024-01-08       | savings      |
|             9 |          9 | deposit          | 1000.25 | 2024-01-09       | savings      |
|            10 |         10 | withdrawal       |  400.50 | 2024-01-10       | current      |
|            11 |         11 | deposit          |  500.75 | 2024-01-11       | savings      |
+---------------+------------+------------------+---------+------------------+--------------+
```

6.

```
mysql> SELECT
    ->     Customers.customer_id,
    ->     Customers.first_name,
    ->     Customers.last_name,
    ->     Customers.DOB,
    ->     Customers.email,
    ->     Customers.phone_number,
    ->     Customers.address,
    ->     Accounts.account_id,
    ->     Accounts.account_type,
    ->     Accounts.balance
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id;
+-------------+------------+-----------+------------+-------------------------+--------------+------------------+------------+--------------+------------+
| customer_id | first_name | last_name | DOB        | email                   | phone_number | address          | account_id | account_type | balance    |
+-------------+------------+-----------+------------+-------------------------+--------------+------------------+------------+--------------+------------+
|           1 | John       | Doe       | 1990-05-15 | john.doe@example.com    | 123-456-7890 | 123 Main St      |          1 | savings      |    5000.00 |
|           2 | Jane       | Smith     | 1985-08-22 | jane.smith@example.com  | 987-654-3210 | 456 Oak Ave      |          2 | current      |    2500.50 |
|           3 | Mike       | Johnson   | 1995-02-10 | mike.johnson@example.com| 555-123-4567 | 789 Pine Rd      |          3 | savings      |   10000.75 |
|           4 | Emily      | Brown     | 1982-11-28 | emily.brown@example.com | 111-222-3333 | 876 Elm Ln       |          4 | zero_balance |       0.00 |
|           5 | Chris      | Miller    | 1998-07-03 | chris.miller@example.com| 999-888-7777 | 543 Birch Blvd   |          5 | current      |    7500.25 |
|           6 | Sarah      | Clark     | 1993-04-18 | sarah.clark@example.com | 777-666-5555 | 234 Cedar St     |          6 | savings      |    3000.50 |
|           7 | David      | Hall      | 1989-09-12 | david.hall@example.com  | 444-555-6666 | 321 Maple Ave    |          7 | current      |    6000.75 |
|           8 | Amy        | White     | 1991-12-30 | amy.white@example.com   | 333-444-5555 | 678 Pine Ln      |          8 | savings      |   12000.00 |
|           9 | Brian      | Taylor    | 1987-06-25 | brian.taylor@example.com| 222-333-4444 | 987 Oak Blvd     |          9 | savings      |    8000.25 |
|          10 | Laura      | Wilson    | 1996-03-08 | laura.wilson@example.com| 666-777-8888 | 876 Birch Rd     |         10 | current      |    4000.50 |
|          11 | Shivam     | Singh     | 2000-09-24 | shivasingh414@gmail.com | 8340508631   | Bokaro Steel City|         11 | savings      | 1002000.00 |
+-------------+------------+-----------+------------+-------------------------+--------------+------------------+------------+--------------+------------+
11 rows in set (0.00 sec)

mysql>
```

7.

```
mysql> SELECT
    ->     Customers.customer_id,
    ->     Customers.first_name,
    ->     Customers.last_name,
    ->     Customers.DOB,
    ->     Customers.email,
    ->     Customers.phone_number,
    ->     Customers.address,
    ->     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
    -> WHERE Accounts.account_id = 5;
+-------------+------------+-----------+------------+-------------------------+--------------+---------------+----------------+------------------+---------+------------------+
| customer_id | first_name | last_name | DOB        | email                   | phone_number | address       | transaction_id | transaction_type | amount  | transaction_date |
+-------------+------------+-----------+------------+-------------------------+--------------+---------------+----------------+------------------+---------+------------------+
|           5 | Chris      | Miller    | 1998-07-03 | chris.miller@example.com| 999-888-7777 | 543 Birch Blvd|              5 | transfer         | 1500.25 | 2024-01-05       |
+-------------+------------+-----------+------------+-------------------------+--------------+---------------+----------------+------------------+---------+------------------+
1 row in set (0.00 sec)
```

8.

```
mysql> SELECT
    ->     Customers.customer_id,
    ->     Customers.first_name,
    ->     Customers.last_name,
    ->     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.last_name
    -> HAVING COUNT(Accounts.account_id) > 1;
Empty set (0.00 sec)
```

9.

```
mysql> SELECT
    ->     Transactions.account_id,
    ->     SUM(CASE WHEN Transactions.transaction_type = 'deposit' THEN Transactions.amount ELSE 0 END)
AS total_deposits,
    ->     SUM(CASE WHEN Transactions.transaction_type = 'withdrawal' THEN Transactions.amount ELSE 0 EN
D) AS total_withdrawals,
    ->     (SUM(CASE WHEN Transactions.transaction_type = 'deposit' THEN Transactions.amount ELSE 0 END)
 -
    ->      SUM(CASE WHEN Transactions.transaction_type = 'withdrawal' THEN Transactions.amount ELSE 0 E
ND)) AS difference
    -> FROM Transactions
    -> GROUP BY Transactions.account_id;
+------------+----------------+-------------------+------------+
| account_id | total_deposits | total_withdrawals | difference |
+------------+----------------+-------------------+------------+
|          1 |        1000.00 |              0.00 |    1000.00 |
|          2 |           0.00 |            500.25 |    -500.25 |
|          3 |        2000.50 |              0.00 |    2000.50 |
|          4 |           0.00 |            100.75 |    -100.75 |
|          5 |           0.00 |              0.00 |       0.00 |
|          6 |         800.50 |              0.00 |     800.50 |
|          7 |           0.00 |           1200.75 |   -1200.75 |
|          8 |           0.00 |              0.00 |       0.00 |
|          9 |        1000.25 |              0.00 |    1000.25 |
|         10 |           0.00 |            400.50 |    -400.50 |
+------------+----------------+-------------------+------------+
```

10.

```
mysql> SELECT
    ->     Transactions.account_id,
    ->     AVG(balance) AS average_daily_balance
    -> FROM (
    ->     SELECT
    ->         account_id,
    ->         transaction_date,
    ->         SUM(CASE WHEN transaction_type = 'deposit' THEN amount ELSE -amount END) AS balance
    ->     FROM Transactions
    ->     GROUP BY account_id, transaction_date
    -> ) AS Transactions
    -> GROUP BY Transactions.account_id;
+------------+----------------------+
| account_id | average_daily_balance |
+------------+----------------------+
|          1 |          1000.000000 |
|          2 |          -500.250000 |
|          3 |          2000.500000 |
|          4 |          -100.750000 |
|          5 |         -1500.250000 |
|          6 |           800.500000 |
|          7 |         -1200.750000 |
|          8 |         -3000.000000 |
|          9 |          1000.250000 |
|         10 |          -400.500000 |
+------------+----------------------+
```

11.

```
mysql> SELECT
    ->     account_type,
    ->     SUM(balance) AS total_balance
    -> FROM Accounts
    -> GROUP BY account_type;
+--------------+---------------+
| account_type | total_balance |
+--------------+---------------+
| savings      |    1040001.50 |
| current      |      20002.00 |
| zero_balance |          0.00 |
+--------------+---------------+
3 rows in set (0.00 sec)
```

12.

```
mysql> SELECT
    ->     Accounts.account_id,
    ->     COUNT(Transactions.transaction_id) AS num_transactions
    -> FROM Accounts
    -> JOIN Transactions ON Accounts.account_id = Transactions.account_id
    -> GROUP BY Accounts.account_id
    -> ORDER BY num_transactions DESC;
+------------+------------------+
| account_id | num_transactions |
+------------+------------------+
|          1 |                1 |
|          2 |                1 |
|          3 |                1 |
|          4 |                1 |
|          5 |                1 |
|          6 |                1 |
|          7 |                1 |
|          8 |                1 |
|          9 |                1 |
|         10 |                1 |
|         11 |                1 |
+------------+------------------+
11 rows in set (0.00 sec)
```

13.

```
mysql> SELECT
    ->     Customers.customer_id,
    ->     Customers.first_name,
    ->     Customers.last_name,
    ->     Customers.DOB,
    ->     Customers.email,
    ->     Customers.phone_number,
    ->     Customers.address,
    ->     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,
    ->     Customers.last_name,
    ->     Customers.DOB,
    ->     Customers.email,
    ->     Customers.phone_number,
    ->     Customers.address,
    ->     Accounts.account_type
    -> ORDER BY total_balance DESC;
+-------------+------------+-----------+------------+-------------------------+--------------+------------------+--------------+---------------+
| customer_id | first_name | last_name | DOB        | email                   | phone_number | address          | account_type | total_balance |
+-------------+------------+-----------+------------+-------------------------+--------------+------------------+--------------+---------------+
|          11 | Shivam     | Singh     | 2000-09-24 | shivasingh414@gmail.com | 8340508631   | Bokaro Steel City | savings      |    1002000.00 |
|           8 | Amy        | White     | 1991-12-30 | amy.white@example.com   | 333-444-5555 | 678 Pine Ln      | savings      |      12000.00 |
|           3 | Mike       | Johnson   | 1995-02-10 | mike.johnson@example.com | 555-123-4567 | 789 Pine Rd      | savings      |      10000.75 |
|           9 | Brian      | Taylor    | 1987-06-25 | brian.taylor@example.com | 222-333-4444 | 987 Oak Blvd     | savings      |       8000.25 |
|           5 | Chris      | Miller    | 1998-07-03 | chris.miller@example.com | 999-888-7777 | 543 Birch Blvd   | current      |       7500.25 |
|           7 | David      | Hall      | 1989-09-12 | david.hall@example.com  | 444-555-6666 | 321 Maple Ave    | current      |       6000.75 |
|           1 | John       | Doe       | 1990-05-15 | john.doe@example.com    | 123-456-7890 | 123 Main St      | savings      |       5000.00 |
|          10 | Laura      | Wilson    | 1996-03-08 | laura.wilson@example.com | 666-777-8888 | 876 Birch Rd     | current      |       4000.50 |
|           6 | Sarah      | Clark     | 1993-04-18 | sarah.clark@example.com | 777-666-5555 | 234 Cedar St     | savings      |       3000.50 |
|           2 | Jane       | Smith     | 1985-08-22 | jane.smith@example.com  | 987-654-3210 | 456 Oak Ave      | current      |       2500.50 |
|           4 | Emily      | Brown     | 1982-11-28 | emily.brown@example.com | 111-222-3333 | 876 Elm Ln       | zero_balance |          0.00 |
+-------------+------------+-----------+------------+-------------------------+--------------+------------------+--------------+---------------+
11 rows in set (0.00 sec)
```

14.

```
mysql> SELECT
    ->     transaction_id,
    ->     account_id,
    ->     transaction_type,
    ->     amount,
    ->     transaction_date
    -> FROM Transactions
    -> WHERE (amount, transaction_date, account_id) IN (
    ->     SELECT amount, transaction_date, account_id
    ->     FROM Transactions
    ->     GROUP BY amount, transaction_date, account_id
    ->     HAVING COUNT(*) > 1
    -> )
    -> ORDER BY amount, transaction_date, account_id, transaction_type;
Empty set (0.00 sec)
```

TASK 4:

Subquery and its type:

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

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

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

4. Identify customers who have no recorded transactions.

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

6. Retrieve transactions for accounts with the lowest balance.

7. Identify customers who have accounts of multiple types.

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

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

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

1.

```
mysql> SELECT
    ->     Customers.customer_id,
    ->     Customers.first_name,
    ->     Customers.last_name,
    ->     Customers.DOB,
    ->     Customers.email,
    ->     Customers.phone_number,
    ->     Customers.address,
    ->     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,
    ->     Customers.last_name,
    ->     Customers.DOB,
    ->     Customers.email,
    ->     Customers.phone_number,
    ->     Customers.address
    -> ORDER BY highest_balance DESC
    -> LIMIT 1;
+-------------+------------+-----------+------------+-----------------------+--------------+-----------------+-----------------+
| customer_id | first_name | last_name | DOB        | email                 | phone_number | address         | highest_balance |
+-------------+------------+-----------+------------+-----------------------+--------------+-----------------+-----------------+
|          11 | Shivam     | Singh     | 2000-09-24 | shivasingh414@gmail.com | 8340508631   | Bokaro Steel City |      1002000.00 |
+-------------+------------+-----------+------------+-----------------------+--------------+-----------------+-----------------+
1 row in set (0.00 sec)
```

2.

```
mysql> SELECT
    ->     Customers.customer_id,
    ->     Customers.first_name,
    ->     Customers.last_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,
    ->     Customers.last_name
    -> HAVING COUNT(Accounts.account_id) > 1;
Empty set (0.00 sec)
```

3.

```
mysql> WITH AvgTransaction AS (
    ->     SELECT
    ->         AVG(amount) AS avg_amount
    ->     FROM Transactions
    -> )
    ->
    -> SELECT
    ->     Accounts.account_id,
    ->     Accounts.customer_id,
    ->     Transactions.transaction_id,
    ->     Transactions.transaction_type,
    ->     Transactions.amount,
    ->     Transactions.transaction_date
    -> FROM Accounts
    -> JOIN Transactions ON Accounts.account_id = Transactions.account_id
    -> JOIN AvgTransaction ON 1=1  -- Cross join to get the average transaction amount
    -> WHERE Transactions.amount > AvgTransaction.avg_amount;
+------------+-------------+----------------+------------------+---------+------------------+
| account_id | customer_id | transaction_id | transaction_type | amount  | transaction_date |
+------------+-------------+----------------+------------------+---------+------------------+
|          3 |           3 |              3 | deposit          | 2000.50 | 2024-01-03       |
|          5 |           5 |              5 | transfer         | 1500.25 | 2024-01-05       |
|          7 |           7 |              7 | withdrawal       | 1200.75 | 2024-01-07       |
|          8 |           8 |              8 | transfer         | 3000.00 | 2024-01-08       |
```

4.

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

5.

```
mysql> SELECT
    ->     SUM(Accounts.balance) AS total_balance_no_transactions
    -> FROM Accounts
    -> LEFT JOIN Transactions ON Accounts.account_id = Transactions.account_id
    -> WHERE Transactions.transaction_id IS NULL;
+-------------------------------+
| total_balance_no_transactions |
+-------------------------------+
|                          NULL |
+-------------------------------+
1 row in set (0.00 sec)
```

6.

```
mysql> WITH MinBalanceAccounts AS (
    ->     SELECT
    ->         account_id,
    ->         MIN(balance) AS min_balance
    ->     FROM Accounts
    ->     GROUP BY account_id
    -> )
    ->
    -> SELECT
    ->     Transactions.transaction_id,
    ->     Transactions.account_id,
    ->     Transactions.transaction_type,
    ->     Transactions.amount,
    ->     Transactions.transaction_date
    -> FROM Transactions
    -> JOIN MinBalanceAccounts ON Transactions.account_id = MinBalanceAccounts.account_id
    -> WHERE Transactions.amount = MinBalanceAccounts.min_balance;
Empty set (0.00 sec)
```

7.

```
mysql> SELECT
    ->     Customers.customer_id,
    ->     Customers.first_name,
    ->     Customers.last_name
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id
    -> GROUP BY
    ->     Customers.customer_id,
    ->     Customers.first_name,
    ->     Customers.last_name
    -> HAVING COUNT(DISTINCT Accounts.account_type) > 1;
Empty set (0.00 sec)
```

8.

```
mysql> SELECT
    ->     account_type,
    ->     COUNT(*) AS num_accounts,
    ->     (COUNT(*) * 100.0 / (SELECT COUNT(*) FROM Accounts)) AS percentage
    -> FROM Accounts
    -> GROUP BY account_type;
+--------------+--------------+------------+
| account_type | num_accounts | percentage |
+--------------+--------------+------------+
| savings      |            6 |   54.54545 |
| current      |            4 |   36.36364 |
| zero_balance |            1 |    9.09091 |
+--------------+--------------+------------+
3 rows in set (0.01 sec)
```

9.

```
mysql> SELECT
    ->     Transactions.transaction_id,
    ->     Transactions.account_id,
    ->     Transactions.transaction_type,
    ->     Transactions.amount,
    ->     Transactions.transaction_date
    -> FROM Transactions
    -> JOIN Accounts ON Transactions.account_id = Accounts.account_id
    -> WHERE Accounts.customer_id = 2;
+----------------+------------+------------------+--------+------------------+
| transaction_id | account_id | transaction_type | amount | transaction_date |
+----------------+------------+------------------+--------+------------------+
|              2 |          2 | withdrawal       | 500.25 | 2024-01-02       |
+----------------+------------+------------------+--------+------------------+
1 row in set (0.00 sec)
```

10.

```
mysql> SELECT
    ->     account_type,
    ->     (SELECT SUM(balance) FROM Accounts WHERE account_type = a.account_type) AS total_balance
    -> FROM Accounts a
    -> GROUP BY account_type;
+--------------+---------------+
| account_type | total_balance |
+--------------+---------------+
| savings      |    1040001.50 |
| current      |      20002.00 |
| zero_balance |         0.00 |
+--------------+---------------+
```