Tasks 1: Database Design:

1.Create the database named "HMBank"

```
mysql> CREATE DATABASE HMBANK;
Query OK, 1 row affected (0.07 sec)
```

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

```
mysql> CREATE TABLE IF NOT EXISTS Customers (customer_id INT PRIMARY KEY,
    ->     first_name VARCHAR(255),
    ->     last_name VARCHAR(255),
    ->     DOB DATE,
    ->     email VARCHAR(255),
    ->     phone_number VARCHAR(20),
    ->     address VARCHAR(255));
Query OK, 0 rows affected (0.12 sec)

mysql> CREATE TABLE IF NOT EXISTS 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 (0.11 sec)

mysql> CREATE TABLE IF NOT EXISTS 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 (0.08 sec)
```

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

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

```
mysql> INSERT INTO Customers (CUSTOMER_ID, first_name, last_name, DOB, email, phone_number, address)
    -> VALUES(1,"NISHITA", "MOHANTY", "2001-12-08", "NM@GMAIL.COM", "78966789", "BBSR"),
    -> (2, 'Priya', 'Verma', '1995-05-12', 'priya.verma@example.com', '876-543-2109', '456 Lotus Street'),
    ->     (3, 'Amit', 'Singh', '1989-11-25', 'amit.singh@example.com', '765-432-1098', '789 Ashoka Lane'),
    ->     (4, 'Neha', 'Sharma', '1998-03-18', 'neha.sharma@example.com', '654-321-0987', '101 Mahatma Road'),
    ->     (5, 'Suresh', 'Patel', '1991-09-30', 'suresh.patel@example.com', '543-210-9876', '202 Gandhi Avenue'),
    ->     (6, 'Kavita', 'Gupta', '1987-07-08', 'kavita.gupta@example.com', '432-109-8765', '303 Vedanta Lane'),
    ->
    -> (7, 'Rahul', 'Shah', '1993-06-22', 'rahul.shah@example.com', '987-654-3211', '456 Blossom Street'),
    ->     (8, 'Anjali', 'Desai', '1986-12-15', 'anjali.desai@example.com', '876-543-2110', '789 Sunshine Lane'),
    ->     (9, 'Sandeep', 'Kulkarni', '1996-04-28', 'sandeep.kulkarni@example.com', '765-432-1109', '101 Riverfront Road'),
    ->     (10, 'Meera', 'Joshi', '1990-10-03', 'meera.joshi@example.com', '654-321-1098', '202 Serenity Lane');
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> INSERT INTO Accounts (account_id, customer_id, account_type, balance)
    -> VALUES
    ->     (1, 1, 'savings', 5000.00),
    ->     (2, 2, 'current', 10000.00),
    ->     (3, 3, 'savings', 7500.00),
    ->     (4, 4, 'current', 12000.00),
    ->     (5, 5, 'zero_balance', 0.00),
    ->     (6, 6, 'current', 9000.00),
    ->     (7, 7, 'savings', 8000.00),
    ->     (8, 8, 'savings', 11000.00),
    ->     (9, 9, 'zero_balance', 0.00),
    ->     (10, 10, 'current', 13000.00);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> INSERT INTO Transactions (transaction_id, account_id, transaction_type, amount, transaction_date)
    -> VALUES
    ->     (1, 1, 'deposit', 1000.00, '2022-02-01'),
    ->     (2, 2, 'withdrawal', 500.00, '2022-02-05'),
    ->     (3, 3, 'transfer', 200.00, '2022-02-10'),
    ->     (4, 4, 'deposit', 1500.00, '2022-02-15'),
    ->     (5, 5, 'withdrawal', 300.00, '2022-02-20'),
    ->     (6, 6, 'transfer', 800.00, '2022-02-25'),
    ->     (7, 7, 'withdrawal', 700.00, '2022-03-01'),
    ->     (8, 8, 'deposit', 1200.00, '2022-03-05'),
    ->     (9, 9, 'withdrawal', 400.00, '2022-03-10'),
    ->     (10, 10, 'transfer', 1000.00, '2022-03-15');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

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 c.first_name, c.last_name, a.account_type, c.email
    -> FROM Customers c
    -> JOIN Accounts a ON c.customer_id = a.customer_id;
+------------+-----------+--------------+-------------------------------+
| first_name | last_name | account_type | email                         |
+------------+-----------+--------------+-------------------------------+
| NISHITA    | MOHANTY   | savings      | NM@GMAIL.COM                  |
| Priya      | Verma     | current      | priya.verma@example.com       |
| Amit       | Singh     | savings      | amit.singh@example.com        |
| Neha       | Sharma    | current      | neha.sharma@example.com       |
| Suresh     | Patel     | zero_balance | suresh.patel@example.com      |
| Kavita     | Gupta     | current      | kavita.gupta@example.com      |
| Rahul      | Shah      | savings      | rahul.shah@example.com        |
| Anjali     | Desai     | savings      | anjali.desai@example.com      |
| Sandeep    | Kulkarni  | zero_balance | sandeep.kulkarni@example.com  |
| Meera      | Joshi     | current      | meera.joshi@example.com       |
+------------+-----------+--------------+-------------------------------+
10 rows in set (0.01 sec)
```

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

```
mysql> SELECT c.first_name, c.last_name, t.transaction_id, t.transaction_type, t.amount, t.transaction_date
    -> FROM Customers c
    -> JOIN Accounts a ON c.customer_id = a.customer_id
    -> JOIN Transactions t ON a.account_id = t.account_id;
+------------+-----------+----------------+------------------+---------+------------------+
| first_name | last_name | transaction_id | transaction_type | amount  | transaction_date |
+------------+-----------+----------------+------------------+---------+------------------+
| NISHITA    | MOHANTY   |              1 | deposit          | 1000.00 | 2022-02-01       |
| Priya      | Verma     |              2 | withdrawal       |  500.00 | 2022-02-05       |
| Amit       | Singh     |              3 | transfer         |  200.00 | 2022-02-10       |
| Neha       | Sharma    |              4 | deposit          | 1500.00 | 2022-02-15       |
| Suresh     | Patel     |              5 | withdrawal       |  300.00 | 2022-02-20       |
| Kavita     | Gupta     |              6 | transfer         |  800.00 | 2022-02-25       |
| Rahul      | Shah      |              7 | withdrawal       |  700.00 | 2022-03-01       |
| Anjali     | Desai     |              8 | deposit          | 1200.00 | 2022-03-05       |
| Sandeep    | Kulkarni  |              9 | withdrawal       |  400.00 | 2022-03-10       |
| Meera      | Joshi     |             10 | transfer         | 1000.00 | 2022-03-15       |
+------------+-----------+----------------+------------------+---------+------------------+
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 + 1000
    -> WHERE account_id = 1;
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from accounts
    -> where account_id =1;
+------------+-------------+--------------+---------+
| account_id | customer_id | account_type | balance |
+------------+-------------+--------------+---------+
|          1 |           1 | savings      | 6000.00 |
+------------+-------------+--------------+---------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT customer_id, CONCAT(first_name, ' ', last_name) AS full_name, email FROM Customers;
+-------------+-----------------+-----------------------------+
| customer_id | full_name       | email                       |
+-------------+-----------------+-----------------------------+
|           1 | NISHITA MOHANTY | NM@GMAIL.COM                |
|           2 | Priya Verma     | priya.verma@example.com     |
|           3 | Amit Singh      | amit.singh@example.com      |
|           4 | Neha Sharma     | neha.sharma@example.com     |
|           5 | Suresh Patel    | suresh.patel@example.com    |
|           6 | Kavita Gupta    | kavita.gupta@example.com    |
|           7 | Rahul Shah      | rahul.shah@example.com      |
|           8 | Anjali Desai    | anjali.desai@example.com    |
|           9 | Sandeep Kulkarni | sandeep.kulkarni@example.com |
|          10 | Meera Joshi     | meera.joshi@example.com     |
+-------------+-----------------+-----------------------------+
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 account_type = 'savings' AND balance = 0;
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 like 'bbsr';
+-------------+------------+-----------+------------+--------------+--------------+---------+
| customer_id | first_name | last_name | DOB        | email        | phone_number | address |
+-------------+------------+-----------+------------+--------------+--------------+---------+
|           1 | NISHITA    | MOHANTY   | 2001-12-08 | NM@GMAIL.COM | 78966789     | BBSR    |
+-------------+------------+-----------+------------+--------------+--------------+---------+
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 = 8;
+----------+
| balance  |
+----------+
| 11000.00 |
+----------+
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;
+------------+-------------+--------------+----------+
| account_id | customer_id | account_type | balance  |
+------------+-------------+--------------+----------+
|          2 |           2 | current      | 10000.00 |
|          4 |           4 | current      | 12000.00 |
|          6 |           6 | current      |  9000.00 |
|         10 |          10 | current      | 13000.00 |
+------------+-------------+--------------+----------+
4 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 = 3;
+---------------+------------+------------------+--------+------------------+
| transaction_id | account_id | transaction_type | amount | transaction_date |
+---------------+------------+------------------+--------+------------------+
|             3 |          3 | transfer         | 200.00 | 2022-02-10       |
+---------------+------------+------------------+--------+------------------+
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> SELECT
    ->     account_id,
    ->     balance,
    ->     (0.05 * balance) AS interest_accrued
    -> FROM
    ->     Accounts
    -> WHERE
    ->     account_type = 'savings';
+------------+----------+------------------+
| account_id | balance  | interest_accrued |
+------------+----------+------------------+
|          1 |  5000.00 |         250.0000 |
|          3 |  7500.00 |         375.0000 |
|          7 |  8000.00 |         400.0000 |
|          8 | 11000.00 |         550.0000 |
+------------+----------+------------------+
4 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 < 1000 AND balance >= 0;
+------------+-------------+--------------+---------+
| account_id | customer_id | account_type | balance |
+------------+-------------+--------------+---------+
|          5 |           5 | zero_balance |    0.00 |
|          9 |           9 | zero_balance |    0.00 |
+------------+-------------+--------------+---------+
2 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 NOT address LIKE 'BBSR';
+-------------+------------+-----------+------------+-----------------------------+--------------+---------------------+
| customer_id | first_name | last_name | DOB        | email                       | phone_number | address             |
+-------------+------------+-----------+------------+-----------------------------+--------------+---------------------+
|           2 | Priya      | Verma     | 1995-05-12 | priya.verma@example.com     | 876-543-2109 | 456 Lotus Street    |
|           3 | Amit       | Singh     | 1989-11-25 | amit.singh@example.com      | 765-432-1098 | 789 Ashoka Lane     |
|           4 | Neha       | Sharma    | 1998-03-18 | neha.sharma@example.com     | 654-321-0987 | 101 Mahatma Road    |
|           5 | Suresh     | Patel     | 1991-09-30 | suresh.patel@example.com    | 543-210-9876 | 202 Gandhi Avenue   |
|           6 | Kavita     | Gupta     | 1987-07-08 | kavita.gupta@example.com    | 432-109-8765 | 303 Vedanta Lane    |
|           7 | Rahul      | Shah      | 1993-06-22 | rahul.shah@example.com      | 987-654-3211 | 456 Blossom Street  |
|           8 | Anjali     | Desai     | 1986-12-15 | anjali.desai@example.com    | 876-543-2110 | 789 Sunshine Lane   |
|           9 | Sandeep    | Kulkarni  | 1996-04-28 | sandeep.kulkarni@example.com | 765-432-1109 | 101 Riverfront Road |
|          10 | Meera      | Joshi     | 1990-10-03 | meera.joshi@example.com     | 654-321-1098 | 202 Serenity Lane   |
+-------------+------------+-----------+------------+-----------------------------+--------------+---------------------+
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 |
+-----------------+
|     7550.000000 |
+-----------------+
1 row in set (0.01 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  |
+------------+-------------+--------------+----------+
|         10 |          10 | current      | 13000.00 |
|          4 |           4 | current      | 12000.00 |
|          8 |           8 | savings      | 11000.00 |
|          2 |           2 | current      | 10000.00 |
|          6 |           6 | current      |  9000.00 |
|          7 |           7 | savings      |  8000.00 |
|          3 |           3 | savings      |  7500.00 |
|          1 |           1 | savings      |  5000.00 |
|          5 |           5 | zero_balance |     0.00 |
|          9 |           9 | zero_balance |     0.00 |
+------------+-------------+--------------+----------+
10 rows in set (0.01 sec)
```

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

```
mysql> SELECT
    ->     c.first_name,
    ->     c.last_name,
    ->     SUM(t.amount) AS total_deposits
    -> FROM
    ->     Customers c
    -> JOIN
    ->     Accounts a ON c.customer_id = a.customer_id
    -> JOIN
    ->     Transactions t ON a.account_id = t.account_id
    -> WHERE
    ->     t.transaction_type = 'deposit' AND
    ->     t.transaction_date = '2022-02-01'
    -> GROUP BY
    ->     c.customer_id;
+------------+-----------+----------------+
| first_name | last_name | total_deposits |
+------------+-----------+----------------+
| NISHITA    | MOHANTY   |        1000.00 |
+------------+-----------+----------------+
1 row in set (0.03 sec)
```

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

```
mysql> -- Oldest Customer
mysql> SELECT *
    -> FROM Customers
    -> ORDER BY DOB ASC
    -> LIMIT 1;
+-------------+------------+-----------+------------+-------------------------+--------------+-------------------+
| customer_id | first_name | last_name | DOB        | email                   | phone_number | address           |
+-------------+------------+-----------+------------+-------------------------+--------------+-------------------+
|           8 | Anjali     | Desai     | 1986-12-15 | anjali.desai@example.com | 876-543-2110 | 789 Sunshine Lane |
+-------------+------------+-----------+------------+-------------------------+--------------+-------------------+
1 row in set (0.00 sec)

mysql>
mysql> -- Newest Customer
mysql> SELECT *
    -> FROM Customers
    -> ORDER BY DOB DESC
    -> LIMIT 1;
+-------------+------------+-----------+------------+-------------+--------------+---------+
| customer_id | first_name | last_name | DOB        | email       | phone_number | address |
+-------------+------------+-----------+------------+-------------+--------------+---------+
|           1 | NISHITA    | MOHANTY   | 2001-12-08 | NM@GMAIL.COM | 78966789     | BBSR    |
+-------------+------------+-----------+------------+-------------+--------------+---------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT
    ->     t.transaction_id,
    ->     t.account_id,
    ->     t.transaction_type,
    ->     t.amount,
    ->     t.transaction_date,
    ->     a.account_type
    -> FROM
    ->     Transactions t
    -> JOIN
    ->     Accounts a ON t.account_id = a.account_id;
+----------------+------------+------------------+----------+------------------+--------------+
| transaction_id | account_id | transaction_type | amount   | transaction_date | account_type |
+----------------+------------+------------------+----------+------------------+--------------+
|              1 |          1 | deposit          | 1000.00 | 2022-02-01       | savings      |
|              2 |          2 | withdrawal       |  500.00 | 2022-02-05       | current      |
|              3 |          3 | transfer         |  200.00 | 2022-02-10       | savings      |
|              4 |          4 | deposit          | 1500.00 | 2022-02-15       | current      |
|              5 |          5 | withdrawal       |  300.00 | 2022-02-20       | zero_balance |
|              6 |          6 | transfer         |  800.00 | 2022-02-25       | current      |
|              7 |          7 | withdrawal       |  700.00 | 2022-03-01       | savings      |
|              8 |          8 | deposit          | 1200.00 | 2022-03-05       | savings      |
|              9 |          9 | withdrawal       |  400.00 | 2022-03-10       | zero_balance |
|             10 |         10 | transfer         | 1000.00 | 2022-03-15       | 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
    ->     c.customer_id,
    ->     c.first_name,
    ->     c.last_name,
    ->     c.email,
    ->     c.phone_number,
    ->     c.address,
    ->     a.account_id,
    ->     a.account_type,
    ->     a.balance
    -> FROM
    ->     Customers c
    -> JOIN
    ->     Accounts a ON c.customer_id = a.customer_id;
+-------------+------------+-----------+-------------------------+--------------+-------------------+------------+--------------+----------+
| customer_id | first_name | last_name | email                   | phone_number | address           | account_id | account_type | balance  |
+-------------+------------+-----------+-------------------------+--------------+-------------------+------------+--------------+----------+
|           1 | NISHITA    | MOHANTY   | NM@GMAIL.COM            | 78966789     | BBSR              |          1 | savings      |  5000.00 |
|           2 | Priya      | Verma     | priya.verma@example.com | 876-543-2109 | 456 Lotus Street  |          2 | current      | 10000.00 |
|           3 | Amit       | Singh     | amit.singh@example.com  | 765-432-1098 | 789 Ashoka Lane   |          3 | savings      |  7500.00 |
|           4 | Neha       | Sharma    | neha.sharma@example.com | 654-321-0987 | 101 Mahatma Road  |          4 | current      | 12000.00 |
|           5 | Suresh     | Patel     | suresh.patel@example.com| 543-210-9876 | 202 Gandhi Avenue |          5 | zero_balance |     0.00 |
|           6 | Kavita     | Gupta     | kavita.gupta@example.com| 432-109-8765 | 303 Vedanta Lane  |          6 | current      |  9000.00 |
|           7 | Rahul      | Shah      | rahul.shah@example.com  | 987-654-3211 | 456 Blossom Street|          7 | savings      |  8000.00 |
|           8 | Anjali     | Desai     | anjali.desai@example.com| 876-543-2110 | 789 Sunshine Lane |          8 | savings      | 11000.00 |
|           9 | Sandeep    | Kulkarni  | sandeep.kulkarni@example.com | 765-432-1109 | 101 Riverfront Road |      9 | zero_balance |     0.00 |
|          10 | Meera      | Joshi     | meera.joshi@example.com | 654-321-1098 | 202 Serenity Lane |         10 | current      | 13000.00 |
+-------------+------------+-----------+-------------------------+--------------+-------------------+------------+--------------+----------+
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
    ->     c.customer_id,
    ->     c.first_name,
    ->     c.last_name,
    ->     c.email,
    ->     c.phone_number,
    ->     c.address,
    ->     t.transaction_id,
    ->     t.transaction_type,
    ->     t.amount,
    ->     t.transaction_date
    -> FROM
    ->     Customers c
    -> JOIN
    ->     Accounts a ON c.customer_id = a.customer_id
    -> JOIN
    ->     Transactions t ON a.account_id = t.account_id
    -> WHERE
    ->     a.account_id = 4;
+-------------+------------+-----------+-------------------------+--------------+-----------------+----------------+------------------+---------+---------
--------+
| customer_id | first_name | last_name | email                   | phone_number | address         | transaction_id | transaction_type | amount  | transacti
on_date |
+-------------+------------+-----------+-------------------------+--------------+-----------------+----------------+------------------+---------+---------
--------+
|           4 | Neha       | Sharma    | neha.sharma@example.com | 654-321-0987 | 101 Mahatma Road |              4 | deposit          | 1500.00 | 2022-02-1
5       |
+-------------+------------+-----------+-------------------------+--------------+-----------------+----------------+------------------+---------+---------
--------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT
    ->     c.customer_id,
    ->     c.first_name,
    ->     c.last_name,
    ->     c.email,
    ->     c.phone_number,
    ->     c.address,
    ->     COUNT(a.account_id) AS num_accounts
    -> FROM
    ->     Customers c
    -> JOIN
    ->     Accounts a ON c.customer_id = a.customer_id
    -> GROUP BY
    ->     c.customer_id
    -> HAVING
    ->     COUNT(a.account_id) > 1;
Empty set (0.00 sec)
```

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

```
mysql> SELECT
    ->     t.account_id,
    ->     SUM(CASE WHEN t.transaction_type = 'deposit' THEN t.amount ELSE -t.amount END) AS transaction_difference
    -> FROM
    ->     Transactions t
    -> GROUP BY
    ->     t.account_id;
+------------+------------------------+
| account_id | transaction_difference |
+------------+------------------------+
|          1 |                1000.00 |
|          2 |                -500.00 |
|          3 |                -200.00 |
|          4 |                1500.00 |
|          5 |                -300.00 |
|          6 |                -800.00 |
|          7 |                -700.00 |
|          8 |                1200.00 |
|          9 |                -400.00 |
|         10 |               -1000.00 |
+------------+------------------------+
10 rows in set (0.00 sec)
```

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

```
mysql> SELECT
    ->     account_id,
    ->     AVG(balance) AS average_daily_balance
    -> FROM
    ->     (
    ->         SELECT
    ->             a.account_id,
    ->             a.balance,
    ->             t.transaction_date,
    ->             ROW_NUMBER() OVER (PARTITION BY a.account_id ORDER BY t.transaction_date DESC) AS row_num
    ->         FROM
    ->             Accounts a
    ->         LEFT JOIN
    ->             Transactions t ON a.account_id = t.account_id
    ->         WHERE
    ->             t.transaction_date >= '2022-02-15' AND t.transaction_date <= '2022-03-15'
    ->     ) subquery
    -> WHERE
    ->     row_num = 1
    -> GROUP BY
    ->     account_id;
+------------+----------------------+
| account_id | average_daily_balance |
+------------+----------------------+
|          4 |          12000.000000 |
|          5 |              0.000000 |
|          6 |           9000.000000 |
|          7 |           8000.000000 |
|          8 |          11000.000000 |
|          9 |              0.000000 |
|         10 |          13000.000000 |
+------------+----------------------+
7 rows in set (0.02 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      |      31500.00 |
| current      |      44000.00 |
| zero_balance |          0.00 |
+--------------+---------------+
3 rows in set (0.00 sec)
```

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

```
mysql> SELECT
    ->     account_id,
    ->     COUNT(transaction_id) AS num_transactions
    -> FROM
    ->     Transactions
    -> GROUP BY
    ->     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 |
+------------+------------------+
10 rows in set (0.01 sec)
```

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

```
mysql> SELECT
    ->     c.customer_id,
    ->     c.first_name,
    ->     c.last_name,
    ->     a.account_type,
    ->     SUM(a.balance) AS total_balance
    -> FROM
    ->     Customers c
    -> JOIN
    ->     Accounts a ON c.customer_id = a.customer_id
    -> GROUP BY
    ->     c.customer_id, a.account_type
    -> ORDER BY
    ->     total_balance DESC;
+-------------+------------+-----------+--------------+---------------+
| customer_id | first_name | last_name | account_type | total_balance |
+-------------+------------+-----------+--------------+---------------+
|          10 | Meera      | Joshi     | current      |      13000.00 |
|           4 | Neha       | Sharma    | current      |      12000.00 |
|           8 | Anjali     | Desai     | savings      |      11000.00 |
|           2 | Priya      | Verma     | current      |      10000.00 |
|           6 | Kavita     | Gupta     | current      |       9000.00 |
|           7 | Rahul      | Shah      | savings      |       8000.00 |
|           3 | Amit       | Singh     | savings      |       7500.00 |
|           1 | NISHITA    | MOHANTY   | savings      |       5000.00 |
|           5 | Suresh     | Patel     | zero_balance |          0.00 |
|           9 | Sandeep    | Kulkarni  | zero_balance |          0.00 |
+-------------+------------+-----------+--------------+---------------+
10 rows in set (0.01 sec)
```

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

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

Tasks 4: Subquery and its type:

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

```
mysql> SELECT
    ->     c.customer_id,
    ->     c.first_name,
    ->     c.last_name,
    ->     MAX(a.balance) AS highest_balance
    -> FROM
    ->     Customers c
    -> JOIN
    ->     Accounts a ON c.customer_id = a.customer_id
    -> GROUP BY
    ->     c.customer_id, c.first_name, c.last_name
    -> ORDER BY
    ->     highest_balance DESC
    -> LIMIT 1;
+-------------+------------+-----------+-----------------+
| customer_id | first_name | last_name | highest_balance |
+-------------+------------+-----------+-----------------+
|          10 | Meera      | Joshi     |        13000.00 |
+-------------+------------+-----------+-----------------+
1 row in set (0.00 sec)
```

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

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

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

```
mysql> SELECT
    ->     a.account_id,
    ->     a.balance,
    ->     t.transaction_id,
    ->     t.amount,
    ->     t.transaction_date
    -> FROM
    ->     Accounts a
    -> JOIN
    ->     Transactions t ON a.account_id = t.account_id
    -> WHERE
    ->     t.amount > (SELECT AVG(amount) FROM Transactions);
+------------+----------+----------------+---------+------------------+
| account_id | balance  | transaction_id | amount  | transaction_date |
+------------+----------+----------------+---------+------------------+
|          1 |  5000.00 |              1 | 1000.00 | 2022-02-01       |
|          4 | 12000.00 |              4 | 1500.00 | 2022-02-15       |
|          6 |  9000.00 |              6 |  800.00 | 2022-02-25       |
|          8 | 11000.00 |              8 | 1200.00 | 2022-03-05       |
|         10 | 13000.00 |             10 | 1000.00 | 2022-03-15       |
+------------+----------+----------------+---------+------------------+
5 rows in set (0.02 sec)
```

4. Identify customers who have no recorded transactions.

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

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

```
mysql> SELECT
    ->     SUM(a.balance) AS total_balance_no_transactions
    -> FROM
    ->     Accounts a
    -> LEFT JOIN
    ->     Transactions t ON a.account_id = t.account_id
    -> WHERE
    ->     t.account_id IS NULL;
+-------------------------------+
| total_balance_no_transactions |
+-------------------------------+
|                          NULL |
+-------------------------------+
1 row in set (0.01 sec)
```

6. Retrieve transactions for accounts with the lowest balance.

```
mysql> SELECT
    ->     t.transaction_id,
    ->     t.account_id,
    ->     t.amount,
    ->     t.transaction_date
    -> FROM
    ->     Transactions t
    -> JOIN (
    ->     SELECT
    ->         a.account_id
    ->     FROM
    ->         Accounts a
    ->     ORDER BY
    ->         a.balance ASC
    ->     LIMIT 1
    -> ) min_balance_accounts ON t.account_id = min_balance_accounts.account_id;
+----------------+------------+--------+------------------+
| transaction_id | account_id | amount | transaction_date |
+----------------+------------+--------+------------------+
|              5 |          5 | 300.00 | 2022-02-20       |
+----------------+------------+--------+------------------+
1 row in set (0.01 sec)
```

7. Identify customers who have accounts of multiple types.

```
mysql> SELECT
    ->     c.customer_id,
    ->     c.first_name,
    ->     c.last_name
    -> FROM
    ->     Customers c
    -> JOIN
    ->     Accounts a ON c.customer_id = a.customer_id
    -> GROUP BY
    ->     c.customer_id
    -> HAVING
    ->     COUNT(DISTINCT a.account_type) > 1;
Empty set (0.00 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(*) / (SELECT COUNT(*) FROM Accounts)) * 100 AS percentage
    -> FROM
    ->     Accounts
    -> GROUP BY
    ->     account_type;
+--------------+---------------+------------+
| account_type | account_count | percentage |
+------+-------+---------------+------------+
| savings      |             4 |    40.0000 |
| current      |             4 |    40.0000 |
| zero_balance |             2 |    20.0000 |
+--------------+---------------+------------+
3 rows in set (0.04 sec)
```

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

```
mysql> SELECT
    ->     t.transaction_id,
    ->     t.account_id,
    ->     t.amount,
    ->     t.transaction_date
    -> FROM
    ->     Transactions t
    -> JOIN
    ->     Accounts a ON t.account_id = a.account_id
    -> WHERE
    ->     a.customer_id = 6;
+----------------+------------+--------+------------------+
| transaction_id | account_id | amount | transaction_date |
+----------------+------------+--------+------------------+
|              6 |          6 | 800.00 | 2022-02-25       |
+----------------+------------+--------+------------------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT
    ->     account_type,
    ->     (SELECT SUM(balance) FROM Accounts a2 WHERE a2.account_type = a.account_type) AS total_balance
    -> FROM
    ->     Accounts a
    -> GROUP BY
    ->     account_type;
+--------------+---------------+
| account_type | total_balance |
+--------------+---------------+
| savings      |      31500.00 |
| current      |      44000.00 |
| zero_balance |          0.00 |
+--------------+---------------+
3 rows in set (0.05 sec)
```