ASSIGNMENT 3

TASK 1: Database Design

Create Database named “HMBank”

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

mysql> use HMBank;
Database changed
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| hmbank             |
| information_schema |
| mysql              |
| performance_schema |
| sakila             |
| sisdb              |
| sys                |
| techshop           |
| world              |
+--------------------+
9 rows in set (0.00 sec)
```

Create Tables:

- Customers
- Accounts
- Transactions

```
mysql> CREATE TABLE Customers (
    ->     customer_id INT PRIMARY KEY,
    ->     first_name VARCHAR(255),
    ->     last_name VARCHAR(255),
    ->     DOB DATE,
    ->     email VARCHAR(255),
    ->     phone_number VARCHAR(15),
    ->     address VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> desc Customers;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| customer_id  | int          | NO   | PRI | NULL    |       |
| first_name   | varchar(255) | YES  |     | NULL    |       |
| last_name    | varchar(255) | YES  |     | NULL    |       |
| DOB          | date         | YES  |     | NULL    |       |
| email        | varchar(255) | YES  |     | NULL    |       |
| phone_number | varchar(15)  | YES  |     | NULL    |       |
| address      | varchar(255) | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

```
mysql> CREATE TABLE Accounts (
    ->     account_id INT PRIMARY KEY,
    ->     customer_id INT,
    ->     account_type VARCHAR(20),
    ->     balance DECIMAL(10, 2),
    ->     FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
    -> );
Query OK, 0 rows affected (0.04 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> CREATE TABLE Transactions (
    ->     transaction_id INT PRIMARY KEY,
    ->     account_id INT,
    ->     transaction_type VARCHAR(20),
    ->     amount DECIMAL(10, 2),
    ->     transaction_date DATE,
    ->     FOREIGN KEY (account_id) REFERENCES Accounts(account_id)
    -> );
Query OK, 0 rows affected (0.04 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)
```

```
mysql> show tables;
+------------------+
| Tables_in_hmbank |
+------------------+
| accounts         |
| customers        |
| transactions     |
+------------------+
3 rows in set (0.00 sec)
```

Insert at least 10 values in each of the table

```
mysql> INSERT INTO Customers (customer_id, first_name, last_name, DOB, email, phone_number,address)
    -> VALUES
    -> (1, 'John', 'Doe', '1990-05-15', 'john.doe@email.com', '123-456-7890','Mumbai'),
    -> (2, 'Jane', 'Smith', '1985-08-22', 'jane.smith@email.com', '987-654-3210','Delhi'),
    -> (3, 'Bob', 'Johnson', '1995-02-10', 'bob.johnson@email.com', '555-123-4567','Chennai'),
    -> (4, 'Alice', 'Williams', '1988-11-30', 'alice.williams@email.com', '321-654-9870','Pune'),
    -> (5, 'Charlie', 'Brown', '1992-07-18', 'charlie.brown@email.com', '777-888-9999','Nagpur'),
    -> (6, 'Emma', 'Davis', '1982-04-05', 'emma.davis@email.com', '111-222-3333','Kolkata'),
    -> (7, 'Michael', 'Jones', '1993-09-25', 'michael.jones@email.com', '444-555-6666','Hydrabad'),
    -> (8, 'Sophia', 'Taylor', '1987-06-12', 'sophia.taylor@email.com', '999-888-7777','Chandigadh'),
    -> (9, 'David', 'Moore', '1997-03-08', 'david.moore@email.com', '666-333-1111','Kanpur'),
    -> (10, 'Olivia', 'Clark', '1980-12-20', 'olivia.clark@email.com', '222-444-5555','Jaypur');
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 | John       | Doe       | 1990-05-15 | john.doe@email.com      | 123-456-7890 | Mumbai     |
|           2 | Jane       | Smith     | 1985-08-22 | jane.smith@email.com    | 987-654-3210 | Delhi      |
|           3 | Bob        | Johnson   | 1995-02-10 | bob.johnson@email.com   | 555-123-4567 | Chennai    |
|           4 | Alice      | Williams  | 1988-11-30 | alice.williams@email.com | 321-654-9870 | Pune       |
|           5 | Charlie    | Brown     | 1992-07-18 | charlie.brown@email.com | 777-888-9999 | Nagpur     |
|           6 | Emma       | Davis     | 1982-04-05 | emma.davis@email.com    | 111-222-3333 | Kolkata    |
|           7 | Michael    | Jones     | 1993-09-25 | michael.jones@email.com | 444-555-6666 | Hydrabad   |
|           8 | Sophia     | Taylor    | 1987-06-12 | sophia.taylor@email.com | 999-888-7777 | Chandigadh |
|           9 | David      | Moore     | 1997-03-08 | david.moore@email.com   | 666-333-1111 | Kanpur     |
|          10 | Olivia     | Clark     | 1980-12-20 | olivia.clark@email.com  | 222-444-5555 | Jaypur     |
+-------------+------------+-----------+------------+-------------------------+--------------+------------+
10 rows in set (0.00 sec)
```

```
mysql> INSERT INTO Accounts (account_id, customer_id, account_type, balance)
    -> VALUES
    ->     (101, 1, 'savings', 5000.00),
    ->     (102, 2, 'current', 10000.00),
    ->     (103, 3, 'savings', 7500.00),
    ->     (104, 4, 'current', 12000.00),
    ->     (105, 5, 'savings', 3000.00),
    ->     (106, 6, 'current', 8000.00),
    ->     (107, 7, 'savings', 6000.00),
    ->     (108, 8, 'current', 9500.00),
    ->     (109, 9, 'savings', 4000.00),
    ->     (110, 10, 'current', 11000.00);
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      |  5000.00 |
|        102 |           2 | current      | 10000.00 |
|        103 |           3 | savings      |  7500.00 |
|        104 |           4 | current      | 12000.00 |
|        105 |           5 | savings      |  3000.00 |
|        106 |           6 | current      |  8000.00 |
|        107 |           7 | savings      |  6000.00 |
|        108 |           8 | current      |  9500.00 |
|        109 |           9 | savings      |  4000.00 |
|        110 |          10 | current      | 11000.00 |
+------------+-------------+--------------+----------+
10 rows in set (0.00 sec)
```

```
mysql> INSERT INTO Transactions (transaction_id, account_id, transaction_type, amount, transaction_date)
    -> VALUES
    ->     (201, 101, 'deposit', 1000.00, '2023-01-15'),
    ->     (202, 102, 'withdrawal', 500.00, '2023-02-20'),
    ->     (203, 103, 'transfer', 200.00, '2023-03-25'),
    ->     (204, 104, 'deposit', 1500.00, '2023-04-10'),
    ->     (205, 105, 'withdrawal', 800.00, '2023-05-05'),
    ->     (206, 106, 'transfer', 300.00, '2023-06-15'),
    ->     (207, 107, 'deposit', 700.00, '2023-07-20'),
    ->     (208, 108, 'withdrawal', 1000.00, '2023-08-12'),
    ->     (209, 109, 'transfer', 400.00, '2023-09-05'),
    ->     (210, 110, 'deposit', 1200.00, '2023-10-18');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from Transactions;
+----------------+------------+------------------+---------+------------------+
| transaction_id | account_id | transaction_type | amount  | transaction_date |
+----------------+------------+------------------+---------+------------------+
|            201 |        101 | deposit          | 1000.00 | 2023-01-15       |
|            202 |        102 | withdrawal       |  500.00 | 2023-02-20       |
|            203 |        103 | transfer         |  200.00 | 2023-03-25       |
|            204 |        104 | deposit          | 1500.00 | 2023-04-10       |
|            205 |        105 | withdrawal       |  800.00 | 2023-05-05       |
|            206 |        106 | transfer         |  300.00 | 2023-06-15       |
|            207 |        107 | deposit          |  700.00 | 2023-07-20       |
|            208 |        108 | withdrawal       | 1000.00 | 2023-08-12       |
|            209 |        109 | transfer         |  400.00 | 2023-09-05       |
|            210 |        110 | deposit          | 1200.00 | 2023-10-18       |
+----------------+------------+------------------+---------+------------------+
10 rows in set (0.00 sec)
```

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

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

```
mysql>  SELECT
    ->       CONCAT(C.first_name,' ',C.last_name) AS customer_name,
    ->       A.account_type,
    ->       C.email
    -> FROM
    ->       Customers C
    -> JOIN
    ->       Accounts A ON C.customer_id=A.customer_id;
+----------------+--------------+--------------------------+
| customer_name  | account_type | email                    |
+----------------+--------------+--------------------------+
| John Doe       | savings      | john.doe@email.com       |
| Jane Smith     | current      | jane.smith@email.com     |
| Bob Johnson    | savings      | bob.johnson@email.com    |
| Alice Williams | current      | alice.williams@email.com |
| Charlie Brown  | savings      | charlie.brown@email.com  |
| Emma Davis     | current      | emma.davis@email.com     |
| Michael Jones  | savings      | michael.jones@email.com  |
| Sophia Taylor  | current      | sophia.taylor@email.com  |
| David Moore    | savings      | david.moore@email.com    |
| Olivia Clark   | current      | olivia.clark@email.com   |
+----------------+--------------+--------------------------+
10 rows in set (0.00 sec)
```

Write SQL query to list all transaction corresponding customer.

```
mysql> SELECT
    ->     T.transaction_id,
    ->     T.account_id,
    ->     T.transaction_type,
    ->     T.amount,
    ->     T.transaction_date,
    ->     CONCAT(C.first_name,' ',C.last_name) AS customer_name,
    ->     C.email
    -> FROM
    ->     Transactions T
    -> JOIN
    ->     Accounts A ON T.account_id = A.account_id
    -> JOIN
    ->     Customers C ON A.customer_id = C.customer_id;
+----------------+------------+------------------+---------+------------------+----------------+--------------------------+
| transaction_id | account_id | transaction_type | amount  | transaction_date | customer_name  | email                    |
+----------------+------------+------------------+---------+------------------+----------------+--------------------------+
|            201 |        101 | deposit          | 1000.00 | 2023-01-15       | John Doe       | john.doe@email.com       |
|            202 |        102 | withdrawal       |  500.00 | 2023-02-20       | Jane Smith     | jane.smith@email.com     |
|            203 |        103 | transfer         |  200.00 | 2023-03-25       | Bob Johnson    | bob.johnson@email.com    |
|            204 |        104 | deposit          | 1500.00 | 2023-04-10       | Alice Williams | alice.williams@email.com |
|            205 |        105 | withdrawal       |  800.00 | 2023-05-05       | Charlie Brown  | charlie.brown@email.com  |
|            206 |        106 | transfer         |  300.00 | 2023-06-15       | Emma Davis     | emma.davis@email.com     |
|            207 |        107 | deposit          |  700.00 | 2023-07-20       | Michael Jones  | michael.jones@email.com  |
|            208 |        108 | withdrawal       | 1000.00 | 2023-08-12       | Sophia Taylor  | sophia.taylor@email.com  |
|            209 |        109 | transfer         |  400.00 | 2023-09-05       | David Moore    | david.moore@email.com    |
|            210 |        110 | deposit          | 1200.00 | 2023-10-18       | Olivia Clark   | olivia.clark@email.com   |
+----------------+------------+------------------+---------+------------------+----------------+--------------------------+
10 rows in set (0.00 sec)
```

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

```
mysql> UPDATE Accounts
    -> SET balance = balance + 500.00
    -> WHERE account_id =103;
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      |  5000.00 |
|        102 |           2 | current      | 10000.00 |
|        103 |           3 | savings      |  8000.00 |
|        104 |           4 | current      | 12000.00 |
|        105 |           5 | savings      |  3000.00 |
|        106 |           6 | current      |  8000.00 |
|        107 |           7 | savings      |  6000.00 |
|        108 |           8 | current      |  9500.00 |
|        109 |           9 | savings      |  4000.00 |
|        110 |          10 | current      | 11000.00 |
+------------+-------------+--------------+----------+
10 rows in set (0.00 sec)
```

Write SQL query to combine first and last names of customers as full name.

```
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 | Bob Johnson    |
|           4 | Alice Williams |
|           5 | Charlie Brown  |
|           6 | Emma Davis     |
|           7 | Michael Jones  |
|           8 | Sophia Taylor  |
|           9 | David Moore    |
|          10 | Olivia Clark   |
+-------------+----------------+
10 rows in set (0.00 sec)
```

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

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

mysql> select * from Accounts;
+------------+-------------+--------------+----------+
| account_id | customer_id | account_type | balance  |
+------------+-------------+--------------+----------+
|        101 |           1 | savings      |  5000.00 |
|        102 |           2 | current      | 10000.00 |
|        103 |           3 | savings      |  8000.00 |
|        104 |           4 | current      | 12000.00 |
|        105 |           5 | savings      |  3000.00 |
|        106 |           6 | current      |  8000.00 |
|        107 |           7 | savings      |  6000.00 |
|        108 |           8 | current      |  9500.00 |
|        109 |           9 | savings      |  4000.00 |
|        110 |          10 | current      | 11000.00 |
+------------+-------------+--------------+----------+
10 rows in set (0.00 sec)
```

Write SQL query to find customers living in specific city.

```
mysql> SELECT
    ->     customer_id,
    ->     first_name,
    ->     last_name,
    ->     address,
    ->     DOB,
    ->     email,
    ->     phone_number
    -> FROM
    ->     Customers
    -> WHERE
    ->     address='Mumbai';
+-------------+------------+-----------+---------+------------+-------------------+--------------+
| customer_id | first_name | last_name | address | DOB        | email             | phone_number |
+-------------+------------+-----------+---------+------------+-------------------+--------------+
|           1 | John       | Doe       | Mumbai  | 1990-05-15 | john.doe@email.com | 123-456-7890 |
+-------------+------------+-----------+---------+------------+-------------------+--------------+
1 row in set (0.00 sec)
```

Write SQL query to get account balance for specific account.

```
mysql> SELECT
    ->     account_id,
    ->     account_type,
    ->     balance
    -> FROM
    ->     Accounts
    -> WHERE
    ->     account_id =107;
+------------+--------------+---------+
| account_id | account_type | balance |
+------------+--------------+---------+
|        107 | savings      | 6000.00 |
+------------+--------------+---------+
1 row in set (0.00 sec)
```

Write SQL query to list all current accounts with balance greater than 1000.

```
mysql> SELECT
    ->     account_id,
    ->     account_type,
    ->     balance
    -> FROM
    ->     Accounts
    -> WHERE
    ->     account_type = 'current' AND balance > 1000;
+------------+--------------+----------+
| account_id | account_type | balance  |
+------------+--------------+----------+
|        102 | current      | 10000.00 |
|        104 | current      | 12000.00 |
|        106 | current      |  8000.00 |
|        108 | current      |  9500.00 |
|        110 | current      | 11000.00 |
+------------+--------------+----------+
5 rows in set (0.00 sec)
```

Write SQL query to retrieve all transactions for specific account.

```
mysql> SELECT
    ->     transaction_id,
    ->     account_id,
    ->     transaction_type,
    ->     amount,
    ->     transaction_date
    -> FROM
    ->     Transactions
    -> WHERE
    ->     account_id =109;
+----------------+------------+------------------+--------+------------------+
| transaction_id | account_id | transaction_type | amount | transaction_date |
+----------------+------------+------------------+--------+------------------+
|            209 |        109 | transfer         | 400.00 | 2023-09-05       |
+----------------+------------+------------------+--------+------------------+
1 row in set (0.00 sec)
```

Write SQL query to find customers not living in specific city.

```
mysql> SELECT
    ->     customer_id,
    ->     first_name,
    ->     last_name,
    ->     address,
    ->     DOB,
    ->     email,
    ->     phone_number
    -> FROM
    ->     Customers
    -> WHERE
    ->     address != 'Chennai';
+-------------+------------+-----------+-----------+------------+-------------------------+--------------+
| customer_id | first_name | last_name | address   | DOB        | email                   | phone_number |
+-------------+------------+-----------+-----------+------------+-------------------------+--------------+
|           1 | John       | Doe       | Mumbai    | 1990-05-15 | john.doe@email.com      | 123-456-7890 |
|           2 | Jane       | Smith     | Delhi     | 1985-08-22 | jane.smith@email.com    | 987-654-3210 |
|           4 | Alice      | Williams  | Pune      | 1988-11-30 | alice.williams@email.com | 321-654-9870 |
|           5 | Charlie    | Brown     | Nagpur    | 1992-07-18 | charlie.brown@email.com | 777-888-9999 |
|           6 | Emma       | Davis     | Kolkata   | 1982-04-05 | emma.davis@email.com    | 111-222-3333 |
|           7 | Michael    | Jones     | Hydrabad  | 1993-09-25 | michael.jones@email.com | 444-555-6666 |
|           8 | Sophia     | Taylor    | Chandigadh | 1987-06-12 | sophia.taylor@email.com | 999-888-7777 |
|           9 | David      | Moore     | Kanpur    | 1997-03-08 | david.moore@email.com   | 666-333-1111 |
|          10 | Olivia     | Clark     | Jaypur    | 1980-12-20 | olivia.clark@email.com  | 222-444-5555 |
+-------------+------------+-----------+-----------+------------+-------------------------+--------------+
9 rows in set (0.00 sec)
```

Task 3: Aggregate function, Having, OrderBy, GroupBy, Joins

Write SQL query to find average account balance for all customers.

```
mysql> SELECT
    ->     AVG(A.balance) AS average_balance
    -> FROM
    ->     Customers C
    -> JOIN
    ->     Accounts A ON C.customer_id = A.customer_id;
+-----------------+
| average_balance |
+-----------------+
|     7650.000000 |
+-----------------+
1 row in set (0.00 sec)
```

Write SQL query to retrieve top 7 highest account balances.

```
mysql> SELECT
    ->     account_id,
    ->     account_type,
    ->     balance
    -> FROM
    ->     Accounts
    -> ORDER BY
    ->     balance DESC
    -> LIMIT 7;
+------------+--------------+----------+
| account_id | account_type | balance  |
+------------+--------------+----------+
|        104 | current      | 12000.00 |
|        110 | current      | 11000.00 |
|        102 | current      | 10000.00 |
|        108 | current      |  9500.00 |
|        103 | savings      |  8000.00 |
|        106 | current      |  8000.00 |
|        107 | savings      |  6000.00 |
+------------+--------------+----------+
7 rows in set (0.00 sec)
```

Write SQL query to calculate total deposits for all customers in specific date.

```
mysql> SELECT
    ->     C.customer_id,
    ->     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 = '2023-04-10'
    -> GROUP BY
    ->     C.customer_id, C.first_name, C.last_name;
+-------------+------------+-----------+----------------+
| customer_id | first_name | last_name | total_deposits |
+-------------+------------+-----------+----------------+
|           4 | Alice      | Williams  |        1500.00 |
+-------------+------------+-----------+----------------+
1 row in set (0.00 sec)
```

Write SQL query to find oldest and newest customers.

```
mysql> SELECT
    ->     customer_id,
    ->     first_name,
    ->     last_name,
    ->     DOB
    -> FROM
    ->     Customers
    -> ORDER BY
    ->     DOB ASC
    -> LIMIT 1;
+-------------+------------+-----------+------------+
| customer_id | first_name | last_name | DOB        |
+-------------+------------+-----------+------------+
|          10 | Olivia     | Clark     | 1980-12-20 |
+-------------+------------+-----------+------------+
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        |
+-------------+------------+-----------+------------+
|           9 | David      | Moore     | 1997-03-08 |
+-------------+------------+-----------+------------+
1 row in set (0.00 sec)
```

Write SQL query to retrieve transaction details along with 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 |
+----------------+------------+------------------+---------+------------------+--------------+
|            201 |        101 | deposit          | 1000.00 | 2023-01-15       | savings      |
|            202 |        102 | withdrawal       |  500.00 | 2023-02-20       | current      |
|            203 |        103 | transfer         |  200.00 | 2023-03-25       | savings      |
|            204 |        104 | deposit          | 1500.00 | 2023-04-10       | current      |
|            205 |        105 | withdrawal       |  800.00 | 2023-05-05       | savings      |
|            206 |        106 | transfer         |  300.00 | 2023-06-15       | current      |
|            207 |        107 | deposit          |  700.00 | 2023-07-20       | savings      |
|            208 |        108 | withdrawal       | 1000.00 | 2023-08-12       | current      |
|            209 |        109 | transfer         |  400.00 | 2023-09-05       | savings      |
|            210 |        110 | deposit          | 1200.00 | 2023-10-18       | current      |
+----------------+------------+------------------+---------+------------------+--------------+
10 rows in set (0.00 sec)
```

Write SQL query to get list of customers along with their account details.

```
mysql> SELECT
    ->     C.customer_id,
    ->     C.first_name,
    ->     C.last_name,
    ->     C.DOB,
    ->     C.email,
    ->     C.phone_number,
    ->     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 | DOB        | email
      | phone_number | account_id | account_type | balance  |
+-------------+------------+-----------+------------+-------------------------+--------------+------------+--------------+----------+
|           1 | John       | Doe       | 1990-05-15 | john.doe@email.com      | 123-456-7890 |        101 | savings      |  5000.00 |
|           2 | Jane       | Smith     | 1985-08-22 | jane.smith@email.com    | 987-654-3210 |        102 | current      | 10000.00 |
|           3 | Bob        | Johnson   | 1995-02-10 | bob.johnson@email.com   | 555-123-4567 |        103 | savings      |  8000.00 |
|           4 | Alice      | Williams  | 1988-11-30 | alice.williams@email.com | 321-654-9870 |        104 | current      | 12000.00 |
|           5 | Charlie    | Brown     | 1992-07-18 | charlie.brown@email.com | 777-888-9999 |        105 | savings      |  3000.00 |
|           6 | Emma       | Davis     | 1982-04-05 | emma.davis@email.com    | 111-222-3333 |        106 | current      |  8000.00 |
|           7 | Michael    | Jones     | 1993-09-25 | michael.jones@email.com | 444-555-6666 |        107 | savings      |  6000.00 |
|           8 | Sophia     | Taylor    | 1987-06-12 | sophia.taylor@email.com | 999-888-7777 |        108 | current      |  9500.00 |
|           9 | David      | Moore     | 1997-03-08 | david.moore@email.com   | 666-333-1111 |        109 | savings      |  4000.00 |
|          10 | Olivia     | Clark     | 1980-12-20 | olivia.clark@email.com  | 222-444-5555 |        110 | current      | 11000.00 |
+-------------+------------+-----------+------------+-------------------------+--------------+------------+--------------+----------+
10 rows in set (0.00 sec)
```

Write SQL query to retrieve transactions details along with customer information for a specific account.

```
mysql> SELECT
    ->     T.transaction_id,
    ->     T.account_id,
    ->     T.transaction_type,
    ->     T.amount,
    ->     T.transaction_date,
    ->     C.customer_id,
    ->     C.first_name,
    ->     C.last_name,
    ->     C.DOB,
    ->     C.email,
    ->     C.phone_number
    -> FROM
    ->     Transactions T
    -> JOIN
    ->     Accounts A ON T.account_id = A.account_id
    -> JOIN
    ->     Customers C ON A.customer_id = C.customer_id
    -> WHERE
    ->     T.account_id = 108;
+----------------+------------+------------------+---------+------------------+-------------+------------+-----------+------------+-------------------------+--------------+
| transaction_id | account_id | transaction_type | amount  | transaction_date | customer_id | first_name | last_name | DOB        | email                   | phone_number |
+----------------+------------+------------------+---------+------------------+-------------+------------+-----------+------------+-------------------------+--------------+
|            208 |        108 | withdrawal       | 1000.00 | 2023-08-12       |           8 | Sophia     | Taylor    | 1987-06-12 | sophia.taylor@email.com | 999-888-7777 |
+----------------+------------+------------------+---------+------------------+-------------+------------+-----------+------------+-------------------------+--------------+
1 row in set (0.00 sec)
```

Write SQL query to identify customers who have more than one account.

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

Write SQL query to calculate difference in transaction amounts between deposits and withdrawals.

```
mysql> SELECT
    ->     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;
+----------------+-------------------+------------+
| total_deposits | total_withdrawals | difference |
+----------------+-------------------+------------+
|        4400.00 |           2300.00 |    1200.00 |
+----------------+-------------------+------------+
1 row in set (0.00 sec)
```

Write SQL query to calculate average daily balance for each account over specified period.

```
mysql> SELECT
    ->     A.account_id,
    ->     AVG(A.balance) AS average_daily_balance
    -> FROM
    ->     Transactions T
    -> JOIN
    ->     Accounts A ON T.account_id = A.account_id
    -> WHERE
    ->     T.transaction_date BETWEEN '2023-03-15' AND '2023-08-18'
    -> GROUP BY
    ->     A.account_id;
+------------+----------------------+
| account_id | average_daily_balance |
+------------+----------------------+
|        103 |           8000.000000 |
|        104 |          12000.000000 |
|        105 |           3000.000000 |
|        106 |           8000.000000 |
|        107 |           6000.000000 |
|        108 |           9500.000000 |
+------------+----------------------+
6 rows in set (0.00 sec)
```

Calculate 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      |      26000.00 |
| current      |      50500.00 |
+--------------+---------------+
2 rows in set (0.00 sec)
```

Identify accounts with highest number of transactions. Order by descending order.

```
mysql> SELECT
    ->     A.account_id,
    ->     COUNT(T.transaction_id) AS num_transactions
    -> FROM
    ->     Accounts A
    -> JOIN
    ->     Transactions T ON A.account_id = T.account_id
    -> GROUP BY
    ->     A.account_id
    -> ORDER BY
    ->     num_transactions DESC;
+------------+------------------+
| account_id | num_transactions |
+------------+------------------+
|        101 |                1 |
|        102 |                1 |
|        103 |                1 |
|        104 |                1 |
|        105 |                1 |
|        106 |                1 |
|        107 |                1 |
|        108 |                1 |
|        109 |                1 |
|        110 |                1 |
+------------+------------------+
10 rows in set (0.00 sec)
```

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, C.first_name, C.last_name, A.account_type
    -> ORDER BY
    ->     total_balance DESC;
+-------------+------------+-----------+--------------+---------------+
| customer_id | first_name | last_name | account_type | total_balance |
+-------------+------------+-----------+--------------+---------------+
|           4 | Alice      | Williams  | current      |      12000.00 |
|          10 | Olivia     | Clark     | current      |      11000.00 |
|           2 | Jane       | Smith     | current      |      10000.00 |
|           8 | Sophia     | Taylor    | current      |       9500.00 |
|           3 | Bob        | Johnson   | savings      |       8000.00 |
|           6 | Emma       | Davis     | current      |       8000.00 |
|           7 | Michael    | Jones     | savings      |       6000.00 |
|           1 | John       | Doe       | savings      |       5000.00 |
|           9 | David      | Moore     | savings      |       4000.00 |
|           5 | Charlie    | Brown     | savings      |       3000.00 |
+-------------+------------+-----------+--------------+---------------+
10 rows in set (0.00 sec)
```

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

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

Task 4. Subquery and its type:

Retrieve the customers with the highest account balance.

```
mysql> SELECT
    ->     C.customer_id,
    ->     C.first_name,
    ->     C.last_name,
    ->     A.account_id,
    ->     A.account_type,
    ->     A.balance
    -> FROM
    ->     Customers C
    -> JOIN
    ->     Accounts A ON C.customer_id = A.customer_id
    -> ORDER BY
    ->     A.balance DESC
    -> LIMIT 1;
+-------------+------------+-----------+------------+--------------+----------+
| customer_id | first_name | last_name | account_id | account_type | balance  |
+-------------+------------+-----------+------------+--------------+----------+
|           4 | Alice      | Williams  |        104 | current      | 12000.00 |
+-------------+------------+-----------+------------+--------------+----------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT
    ->     C.customer_id,
    ->     C.first_name,
    ->     C.last_name,
    ->     AVG(A.balance) AS average_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
    -> HAVING
    ->     COUNT(A.account_id) > 1;
Empty set (0.00 sec)
```

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

```
mysql> SELECT
    ->     A.account_id,
    ->     A.account_type,
    ->     T.transaction_id,
    ->     T.transaction_type,
    ->     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 | account_type | transaction_id | transaction_type | amount  | transaction_date |
+------------+--------------+----------------+------------------+---------+------------------+
|        101 | savings      |            201 | deposit          | 1000.00 | 2023-01-15       |
|        104 | current      |            204 | deposit          | 1500.00 | 2023-04-10       |
|        105 | savings      |            205 | withdrawal       |  800.00 | 2023-05-05       |
|        108 | current      |            208 | withdrawal       | 1000.00 | 2023-08-12       |
|        110 | current      |            210 | deposit          | 1200.00 | 2023-10-18       |
+------------+--------------+----------------+------------------+---------+------------------+
5 rows in set (0.00 sec)
```

Calculate 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 A.account_id = T.account_id
    ->     );
Empty set (0.00 sec)
```

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, C.first_name, C.last_name
    -> HAVING
    ->     COUNT(DISTINCT A.account_type) > 1;
Empty set (0.00 sec)
```

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

```
mysql> SELECT
    ->     account_type,
    ->     COUNT(*) AS account_count,
    ->     (COUNT(*) * 100.0 / SUM(COUNT(*)) OVER ()) AS percentage
    -> FROM
    ->     Accounts
    -> GROUP BY
    ->     account_type;
+--------------+---------------+------------+
| account_type | account_count | percentage |
+--------------+---------------+------------+
| savings      |             5 |   50.00000 |
| current      |             5 |   50.00000 |
+--------------+---------------+------------+
2 rows in set (0.01 sec)
```

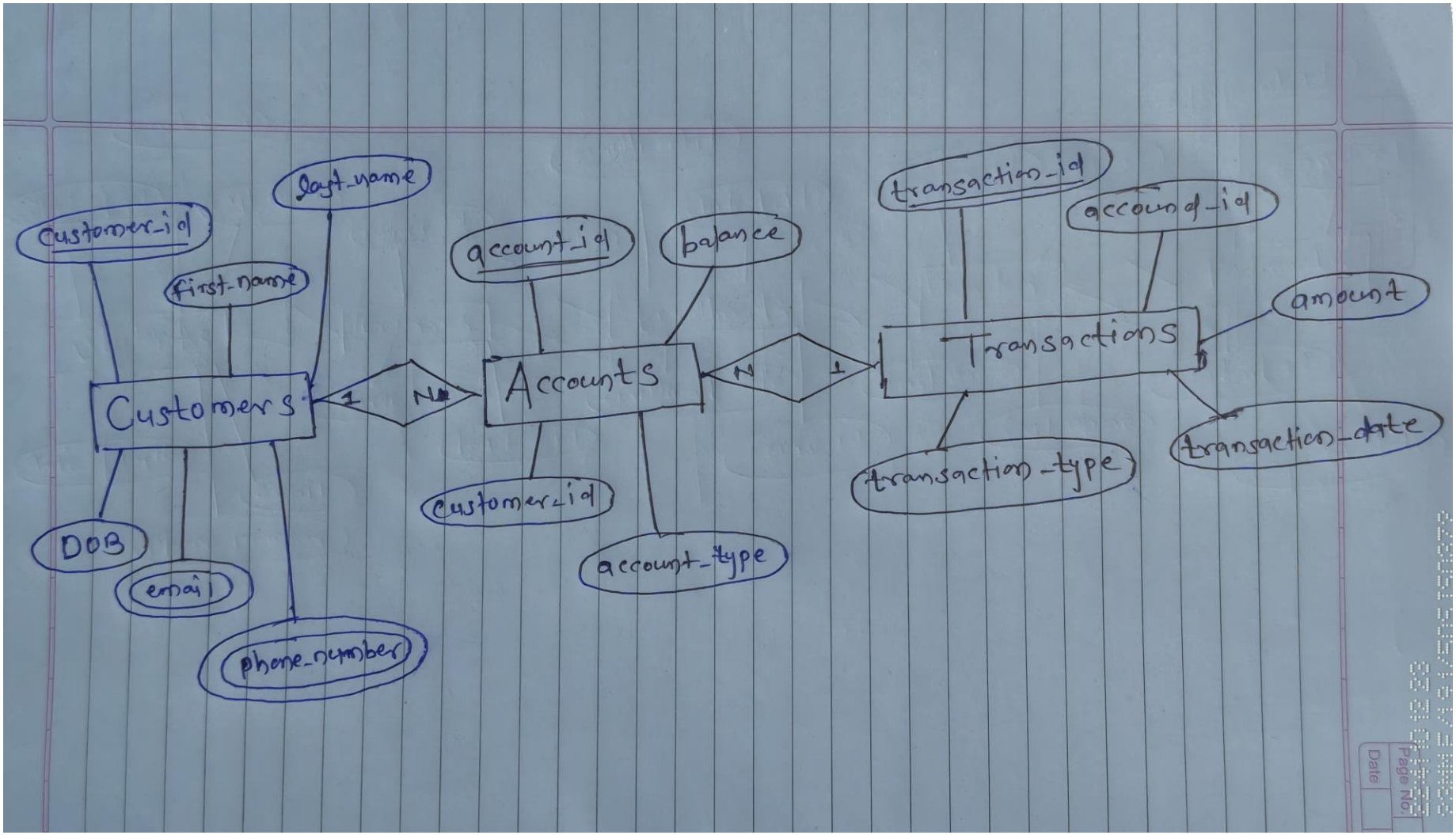
Retrieve all transactions for customer with given customer_id.

```
mysql> SELECT
    ->     T.transaction_id,
    ->     T.account_id,
    ->     T.transaction_type,
    ->     T.amount,
    ->     T.transaction_date
    -> FROM
    ->     Transactions T
    -> JOIN
    ->     Accounts A ON T.account_id = A.account_id
    -> WHERE
    ->     A.customer_id =7;
+----------------+------------+------------------+--------+------------------+
| transaction_id | account_id | transaction_type | amount | transaction_date |
+----------------+------------+------------------+--------+------------------+
|            207 |        107 | deposit          | 700.00 | 2023-07-20       |
+----------------+------------+------------------+--------+------------------+
1 row in set (0.00 sec)
```

Calculate 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 = A1.account_type
    ->     ) AS total_balance
    -> FROM
    ->     Accounts A1
    -> GROUP BY
    ->     account_type;
+--------------+---------------+
| account_type | total_balance |
+--------------+---------------+
| savings      |      26000.00 |
| current      |      50500.00 |
+--------------+---------------+
2 rows in set (0.00 sec)
```

Entity Relationship Diagram


Customers
customer_id
first_name
last_name
DOB
email
phone_number
1
N
Accounts
account_id
balance
customer_id
account_type
N
1
Transactions
transaction_id
account_id
amount
transaction_type
transaction_date