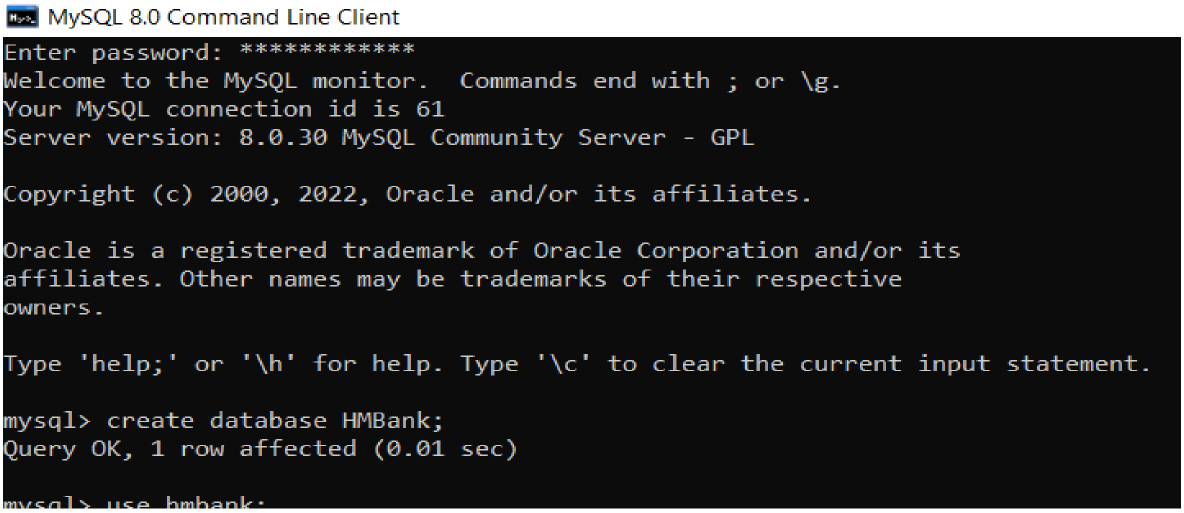
ASSIGNMENT 3

NAME : - SUSHANT KUMAR SINGH

EMAIL :- ssushant886@gmail.com

Tasks 1: Database Design:

1. **Create the database named "HMBank"**

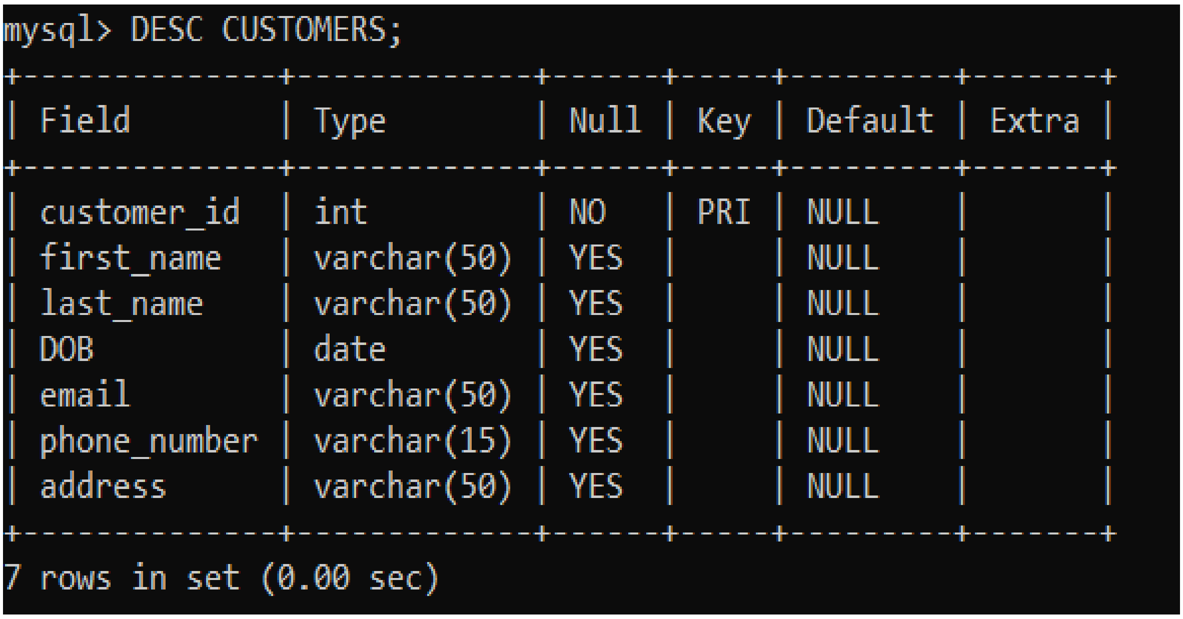


MySQL 8.0 Command Line Client

```
Enter password: *************
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 61
Server version: 8.0.30 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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

mysql> use hmbank;
```

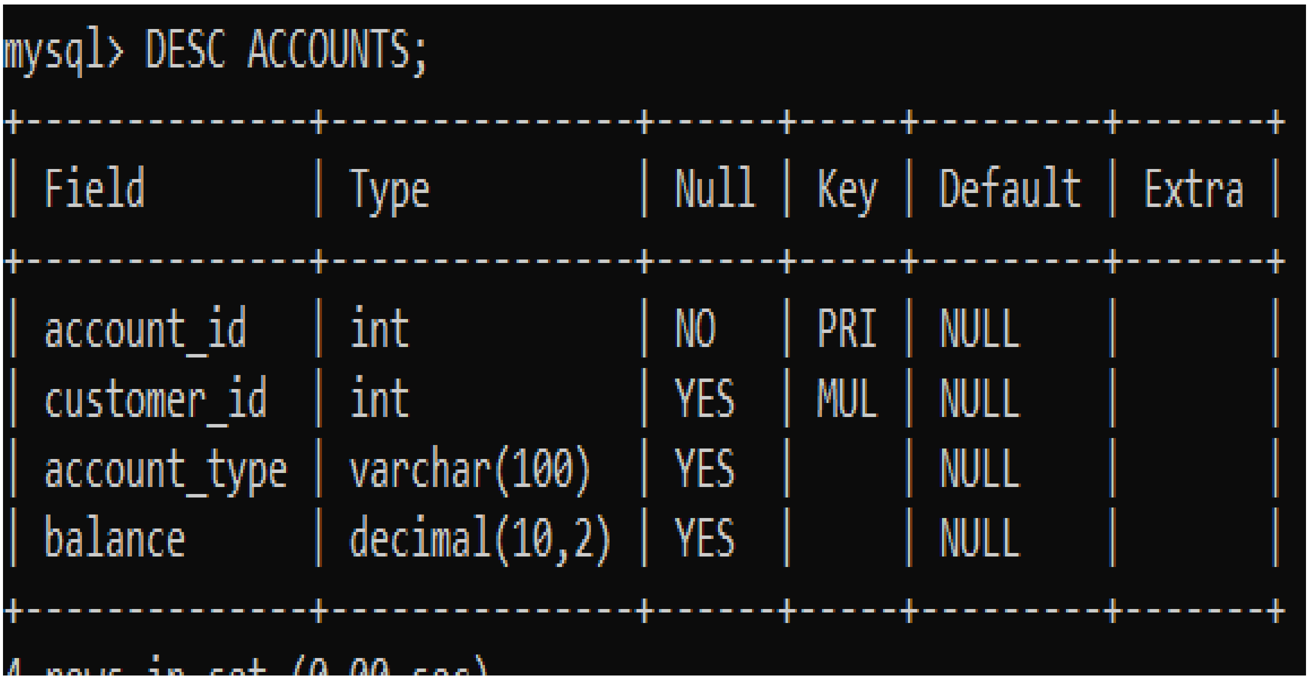
2. **Define the schema for the Customers, Accounts, and Transactions tables based on the provided schema.**

    1) **CUSTOMERS SCHEMA**



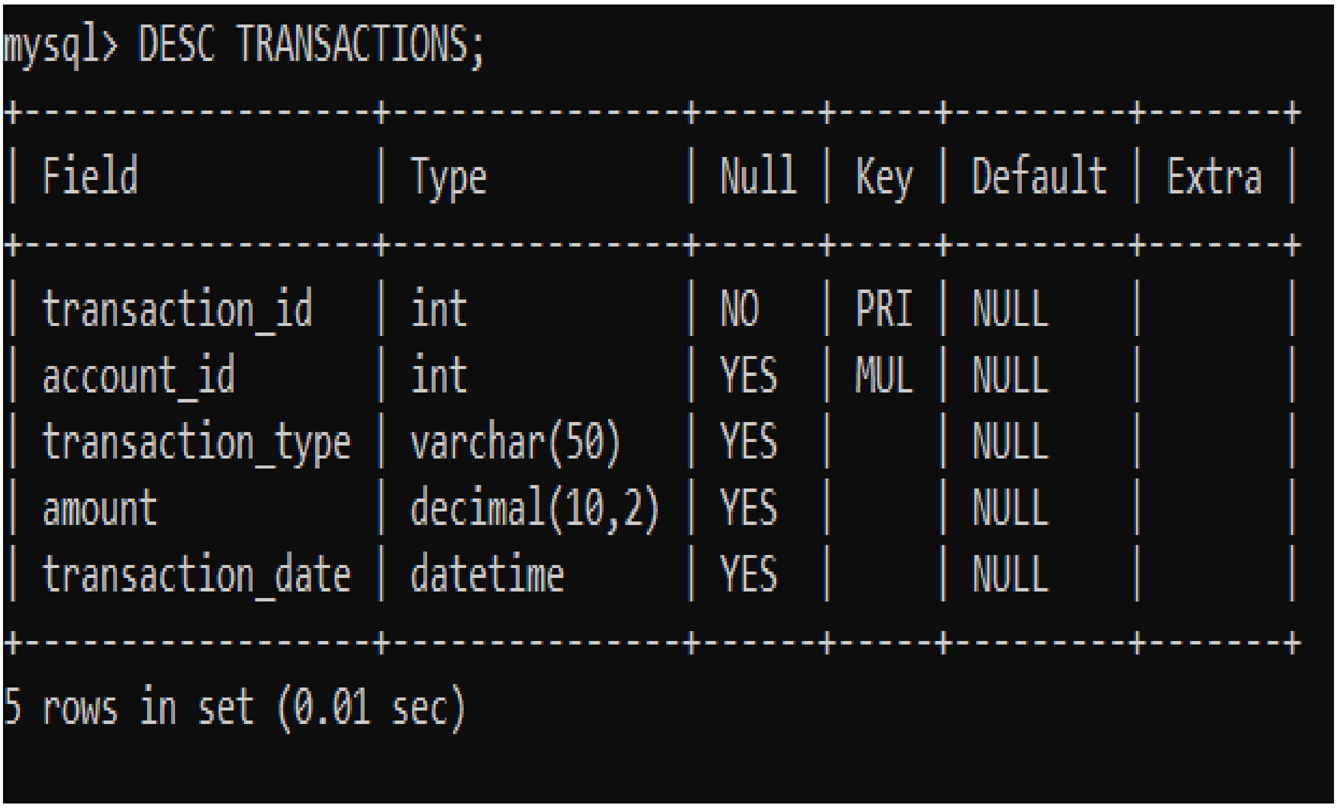
```
mysql> DESC CUSTOMERS;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| customer_id  | int         | NO   | PRI | NULL    |       |
| first_name   | varchar(50) | YES  |     | NULL    |       |
| last_name    | varchar(50) | YES  |     | NULL    |       |
| DOB          | date        | YES  |     | NULL    |       |
| email        | varchar(50) | YES  |     | NULL    |       |
| phone_number | varchar(15) | YES  |     | NULL    |       |
| address      | varchar(50) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

2) ACCOUNTS SCHEMA



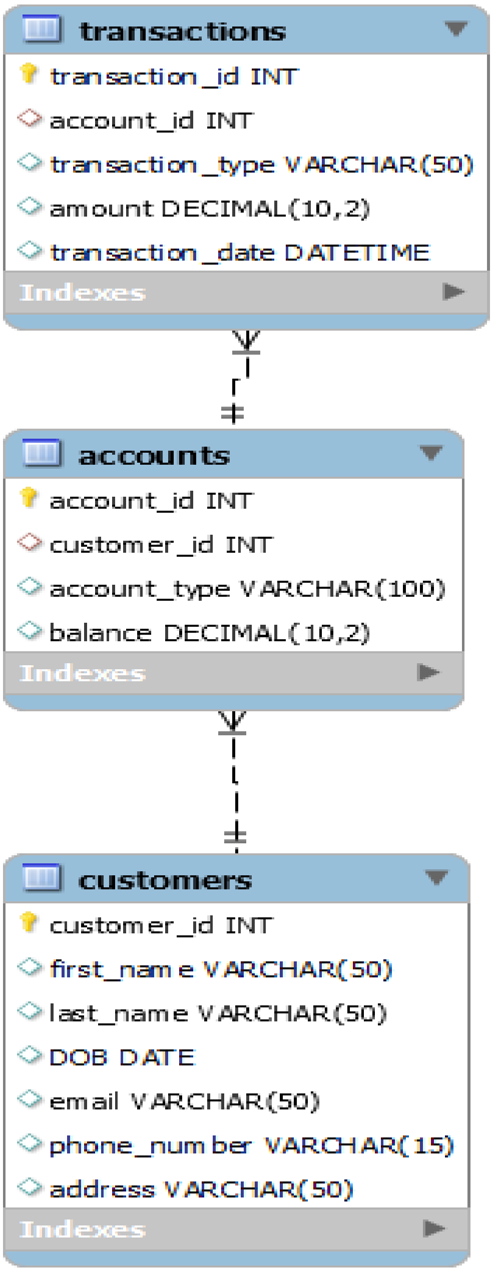
```
mysql> DESC ACCOUNTS;
+--------------+---------------+------+-----+---------+-------+
| Field        | Type          | Null | Key | Default | Extra |
+--------------+---------------+------+-----+---------+-------+
| account_id   | int           | NO   | PRI | NULL    |       |
| customer_id  | int           | YES  | MUL | NULL    |       |
| account_type | varchar(100)  | YES  |     | NULL    |       |
| balance      | decimal(10,2) | YES  |     | NULL    |       |
+--------------+---------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

3) TRANSACTIONS SCHEMA



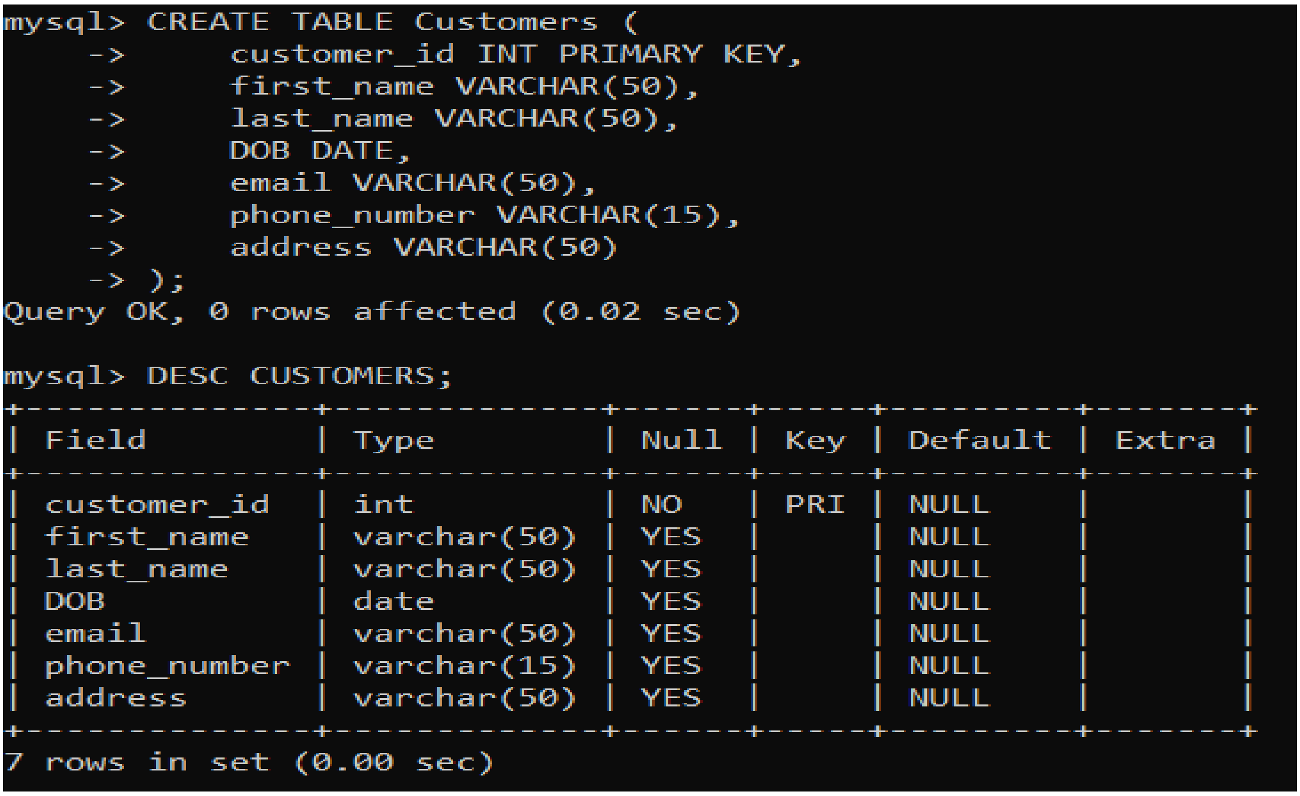
```
mysql> DESC TRANSACTIONS;
+------------------+---------------+------+-----+---------+-------+
| Field            | Type          | Null | Key | Default | Extra |
+------------------+---------------+------+-----+---------+-------+
| transaction_id   | int           | NO   | PRI | NULL    |       |
| account_id       | int           | YES  | MUL | NULL    |       |
| transaction_type | varchar(50)   | YES  |     | NULL    |       |
| amount           | decimal(10,2) | YES  |     | NULL    |       |
| transaction_date | datetime      | YES  |     | NULL    |       |
+------------------+---------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

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



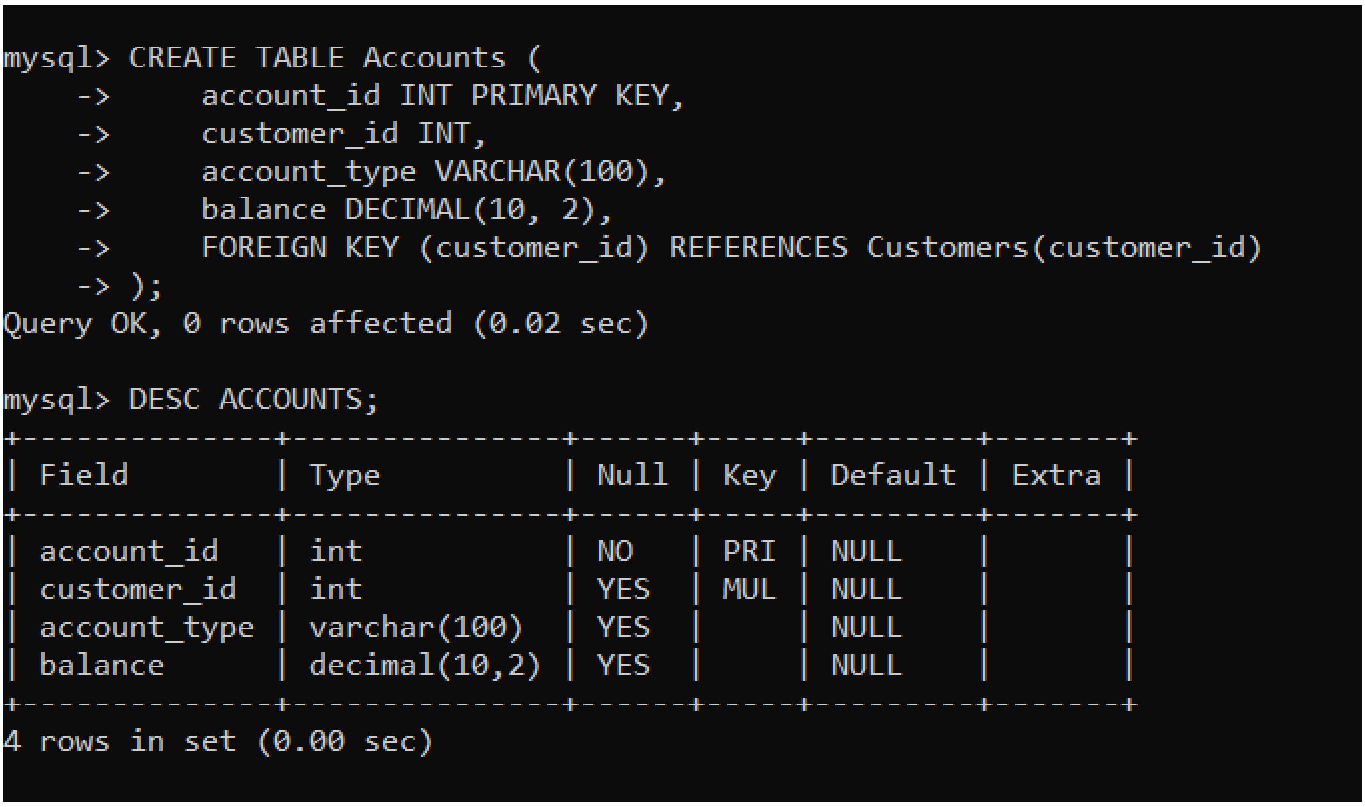
6. Write SQL scripts to create the mentioned tables with appropriate data types, constraints, and relationships. • Customers • Accounts • Transactions

1) CUSTOMES



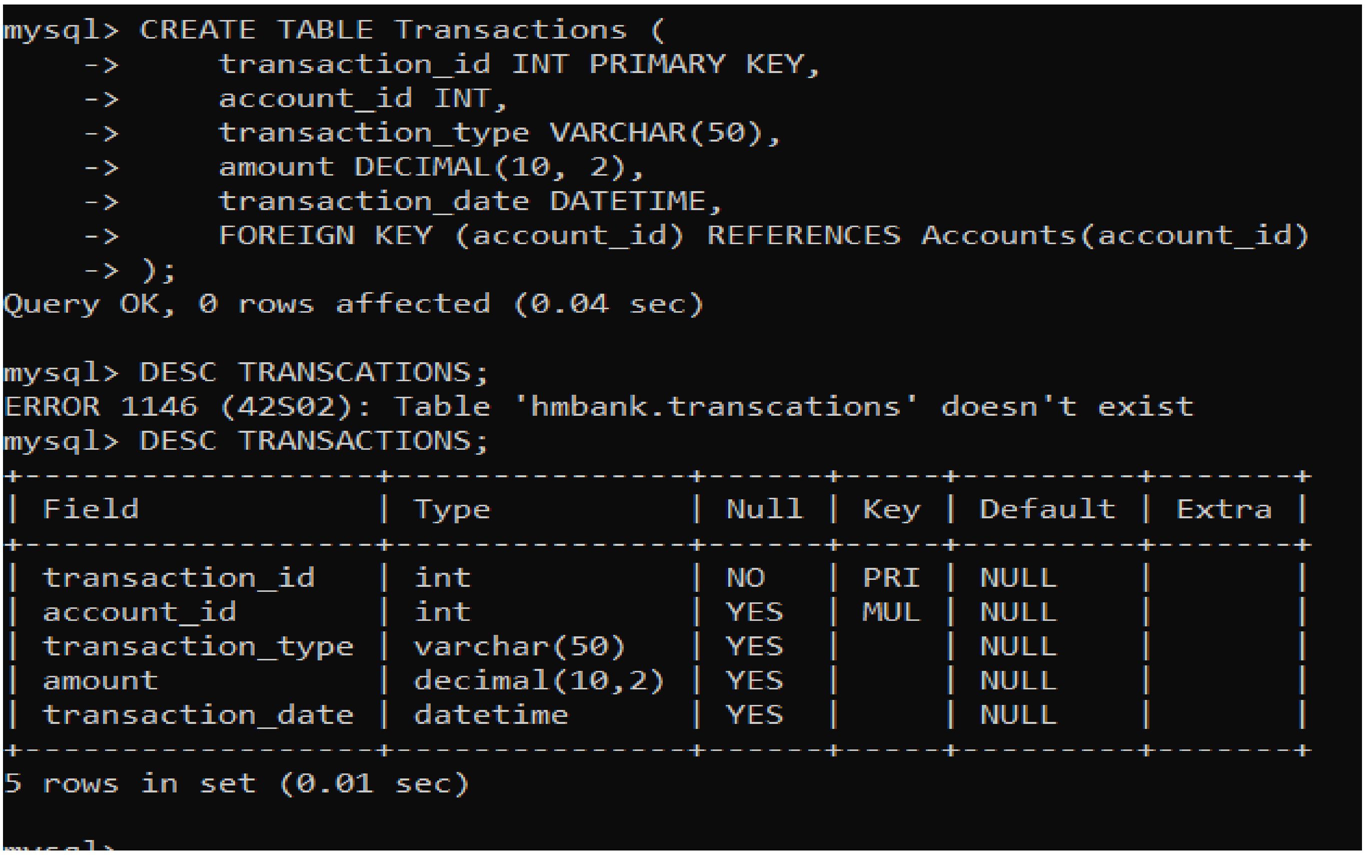
```
mysql> CREATE TABLE Customers (
    ->     customer_id INT PRIMARY KEY,
    ->     first_name VARCHAR(50),
    ->     last_name VARCHAR(50),
    ->     DOB DATE,
    ->     email VARCHAR(50),
    ->     phone_number VARCHAR(15),
    ->     address VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> DESC CUSTOMERS;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| customer_id  | int         | NO   | PRI | NULL    |       |
| first_name   | varchar(50) | YES  |     | NULL    |       |
| last_name    | varchar(50) | YES  |     | NULL    |       |
| DOB          | date        | YES  |     | NULL    |       |
| email        | varchar(50) | YES  |     | NULL    |       |
| phone_number | varchar(15) | YES  |     | NULL    |       |
| address      | varchar(50) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

2) ACCOUNTS



```
mysql> CREATE TABLE Accounts (
    ->     account_id INT PRIMARY KEY,
    ->     customer_id INT,
    ->     account_type VARCHAR(100),
    ->     balance DECIMAL(10, 2),
    ->     FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
    -> );
Query OK, 0 rows affected (0.02 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(100)  | YES  |     | NULL    |       |
| balance      | decimal(10,2) | YES  |     | NULL    |       |
+--------------+---------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

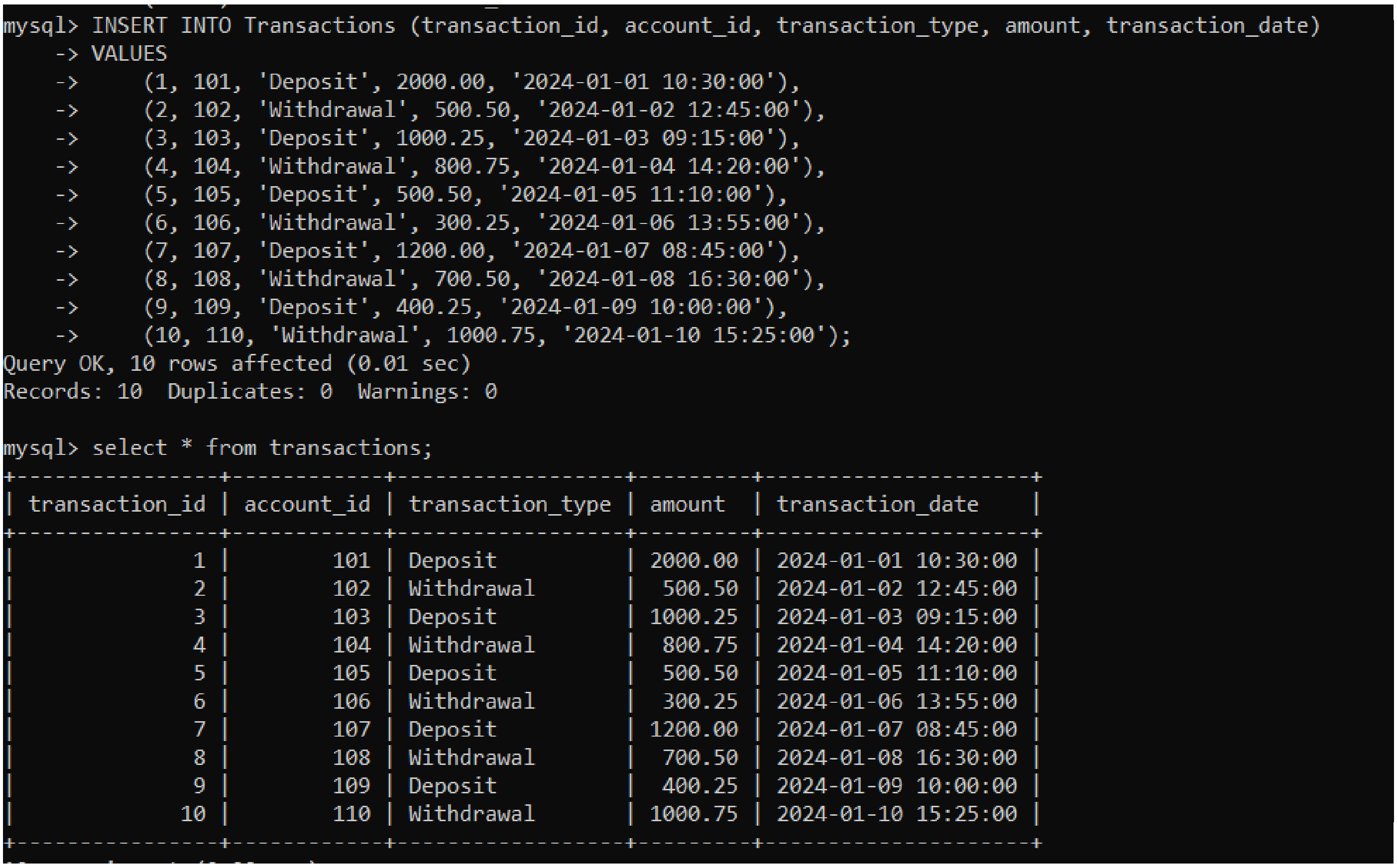
3) TRANSACTIONS



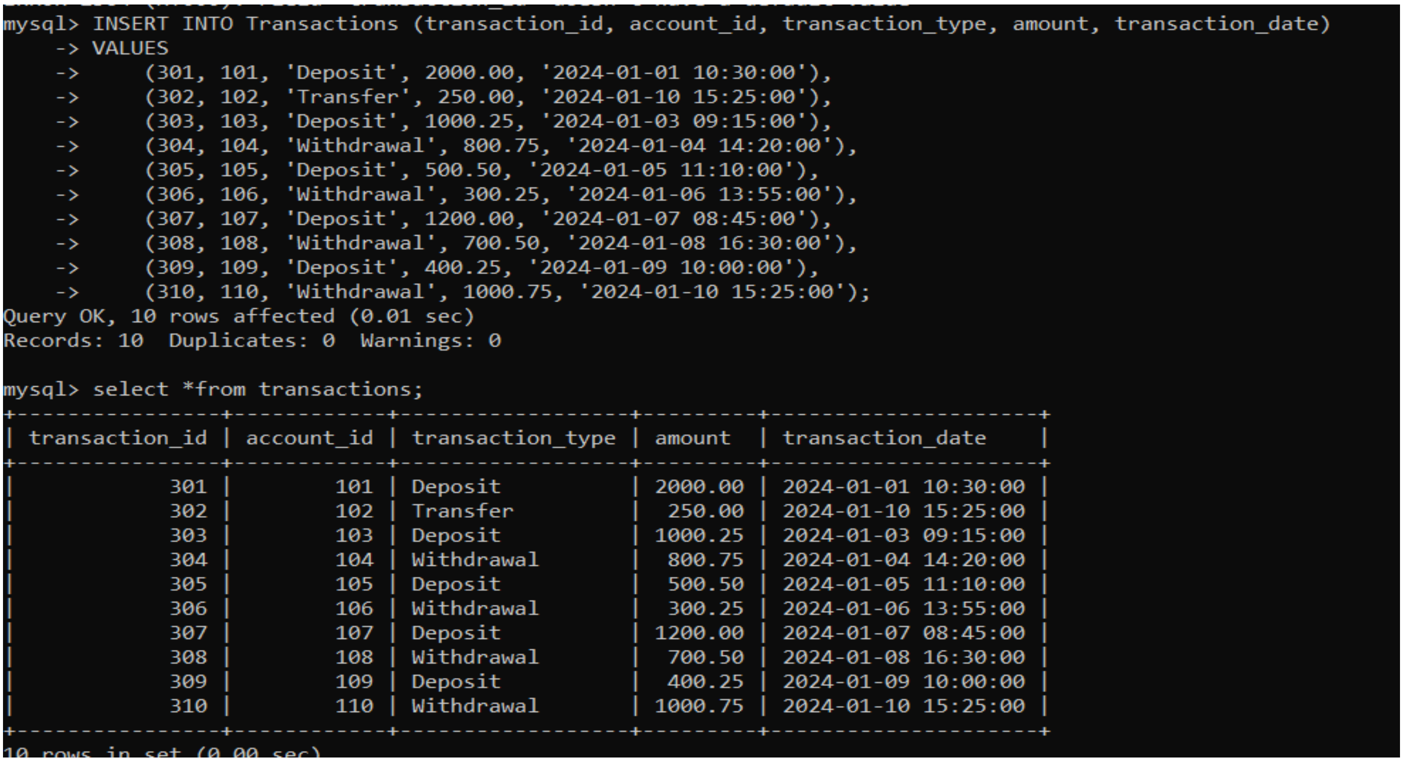
```
mysql> CREATE TABLE Transactions (
    ->     transaction_id INT PRIMARY KEY,
    ->     account_id INT,
    ->     transaction_type VARCHAR(50),
    ->     amount DECIMAL(10, 2),
    ->     transaction_date DATETIME,
    ->     FOREIGN KEY (account_id) REFERENCES Accounts(account_id)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> DESC TRANSCATIONS;
ERROR 1146 (42S02): Table 'hmbank.transcations' doesn't exist
mysql> DESC TRANSACTIONS;
+------------------+---------------+------+-----+---------+-------+
| Field            | Type          | Null | Key | Default | Extra |
+------------------+---------------+------+-----+---------+-------+
| transaction_id   | int           | NO   | PRI | NULL    |       |
| account_id       | int           | YES  | MUL | NULL    |       |
| transaction_type | varchar(50)   | YES  |     | NULL    |       |
| amount           | decimal(10,2) | YES  |     | NULL    |       |
| transaction_date | datetime      | YES  |     | NULL    |       |
+------------------+---------------+------+-----+---------+-------+
5 rows in set (0.01 sec)

mysql>
```

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

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



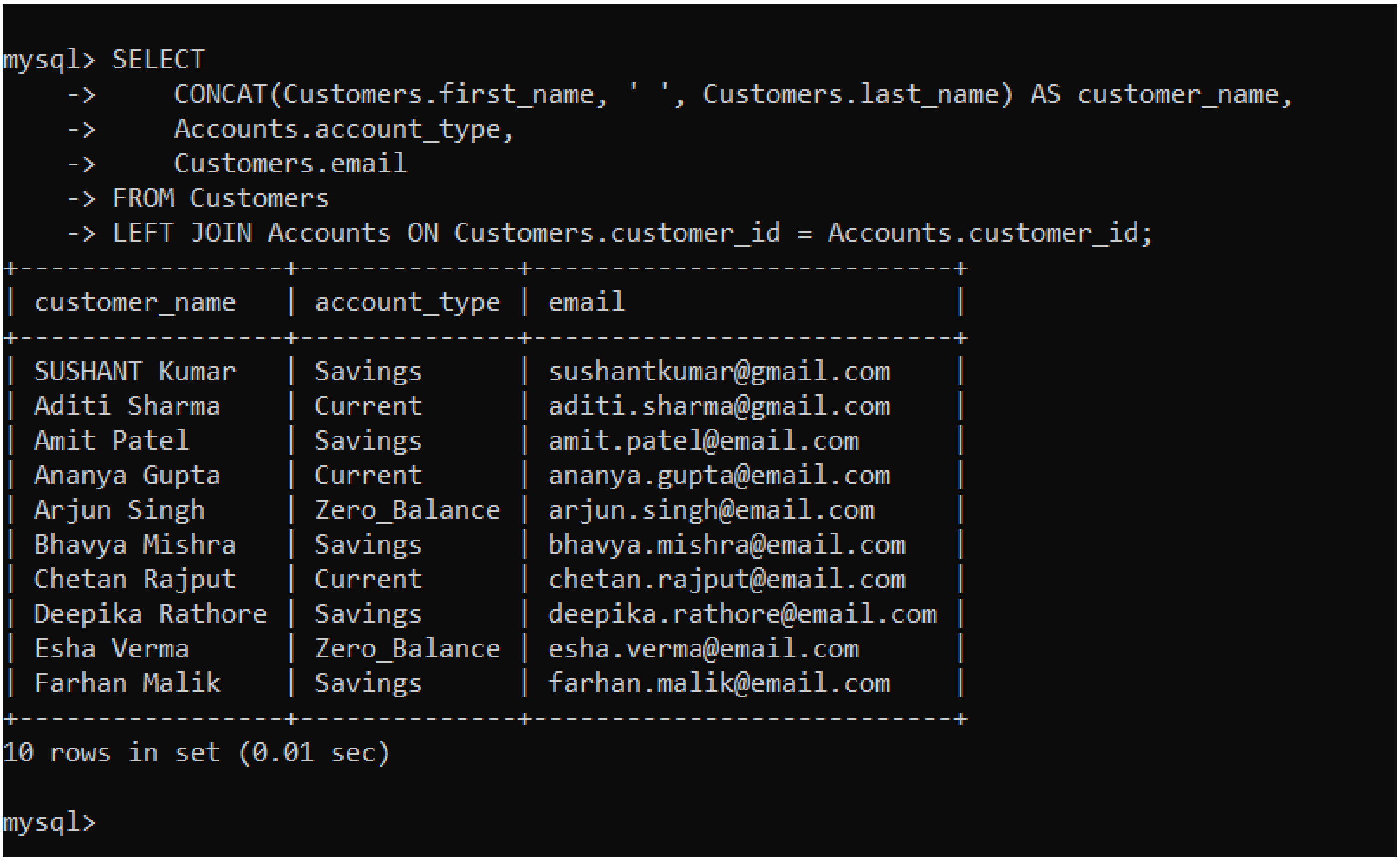
```
mysql> INSERT INTO Transactions (transaction_id, account_id, transaction_type, amount, transaction_date)
    -> VALUES
    ->     (1, 101, 'Deposit', 2000.00, '2024-01-01 10:30:00'),
    ->     (2, 102, 'Withdrawal', 500.50, '2024-01-02 12:45:00'),
    ->     (3, 103, 'Deposit', 1000.25, '2024-01-03 09:15:00'),
    ->     (4, 104, 'Withdrawal', 800.75, '2024-01-04 14:20:00'),
    ->     (5, 105, 'Deposit', 500.50, '2024-01-05 11:10:00'),
    ->     (6, 106, 'Withdrawal', 300.25, '2024-01-06 13:55:00'),
    ->     (7, 107, 'Deposit', 1200.00, '2024-01-07 08:45:00'),
    ->     (8, 108, 'Withdrawal', 700.50, '2024-01-08 16:30:00'),
    ->     (9, 109, 'Deposit', 400.25, '2024-01-09 10:00:00'),
    ->     (10, 110, 'Withdrawal', 1000.75, '2024-01-10 15:25:00');
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    |
+----------------+------------+------------------+---------+---------------------+
|              1 |        101 | Deposit          | 2000.00 | 2024-01-01 10:30:00 |
|              2 |        102 | Withdrawal       |  500.50 | 2024-01-02 12:45:00 |
|              3 |        103 | Deposit          | 1000.25 | 2024-01-03 09:15:00 |
|              4 |        104 | Withdrawal       |  800.75 | 2024-01-04 14:20:00 |
|              5 |        105 | Deposit          |  500.50 | 2024-01-05 11:10:00 |
|              6 |        106 | Withdrawal       |  300.25 | 2024-01-06 13:55:00 |
|              7 |        107 | Deposit          | 1200.00 | 2024-01-07 08:45:00 |
|              8 |        108 | Withdrawal       |  700.50 | 2024-01-08 16:30:00 |
|              9 |        109 | Deposit          |  400.25 | 2024-01-09 10:00:00 |
|             10 |        110 | Withdrawal       | 1000.75 | 2024-01-10 15:25:00 |
+----------------+------------+------------------+---------+---------------------+
```



```
mysql> INSERT INTO Transactions (transaction_id, account_id, transaction_type, amount, transaction_date)
    -> VALUES
    ->     (301, 101, 'Deposit', 2000.00, '2024-01-01 10:30:00'),
    ->     (302, 102, 'Transfer', 250.00, '2024-01-10 15:25:00'),
    ->     (303, 103, 'Deposit', 1000.25, '2024-01-03 09:15:00'),
    ->     (304, 104, 'Withdrawal', 800.75, '2024-01-04 14:20:00'),
    ->     (305, 105, 'Deposit', 500.50, '2024-01-05 11:10:00'),
    ->     (306, 106, 'Withdrawal', 300.25, '2024-01-06 13:55:00'),
    ->     (307, 107, 'Deposit', 1200.00, '2024-01-07 08:45:00'),
    ->     (308, 108, 'Withdrawal', 700.50, '2024-01-08 16:30:00'),
    ->     (309, 109, 'Deposit', 400.25, '2024-01-09 10:00:00'),
    ->     (310, 110, 'Withdrawal', 1000.75, '2024-01-10 15:25:00');
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    |
+----------------+------------+------------------+---------+---------------------+
|            301 |        101 | Deposit          | 2000.00 | 2024-01-01 10:30:00 |
|            302 |        102 | Transfer         |  250.00 | 2024-01-10 15:25:00 |
|            303 |        103 | Deposit          | 1000.25 | 2024-01-03 09:15:00 |
|            304 |        104 | Withdrawal       |  800.75 | 2024-01-04 14:20:00 |
|            305 |        105 | Deposit          |  500.50 | 2024-01-05 11:10:00 |
|            306 |        106 | Withdrawal       |  300.25 | 2024-01-06 13:55:00 |
|            307 |        107 | Deposit          | 1200.00 | 2024-01-07 08:45:00 |
|            308 |        108 | Withdrawal       |  700.50 | 2024-01-08 16:30:00 |
|            309 |        109 | Deposit          |  400.25 | 2024-01-09 10:00:00 |
|            310 |        110 | Withdrawal       | 1000.75 | 2024-01-10 15:25:00 |
+----------------+------------+------------------+---------+---------------------+
10 rows in set (0.00 sec)
```

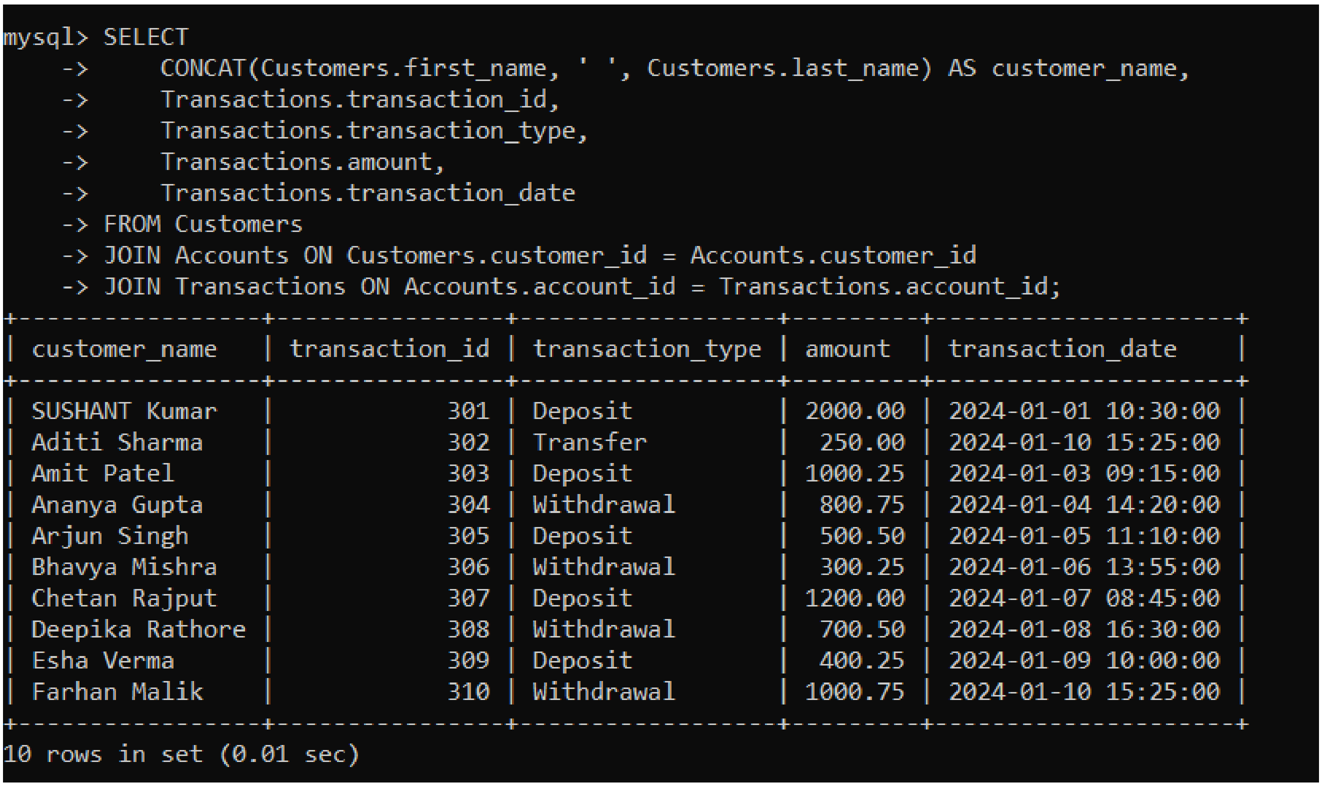
2. **Write SQL queries for the following tasks:**

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



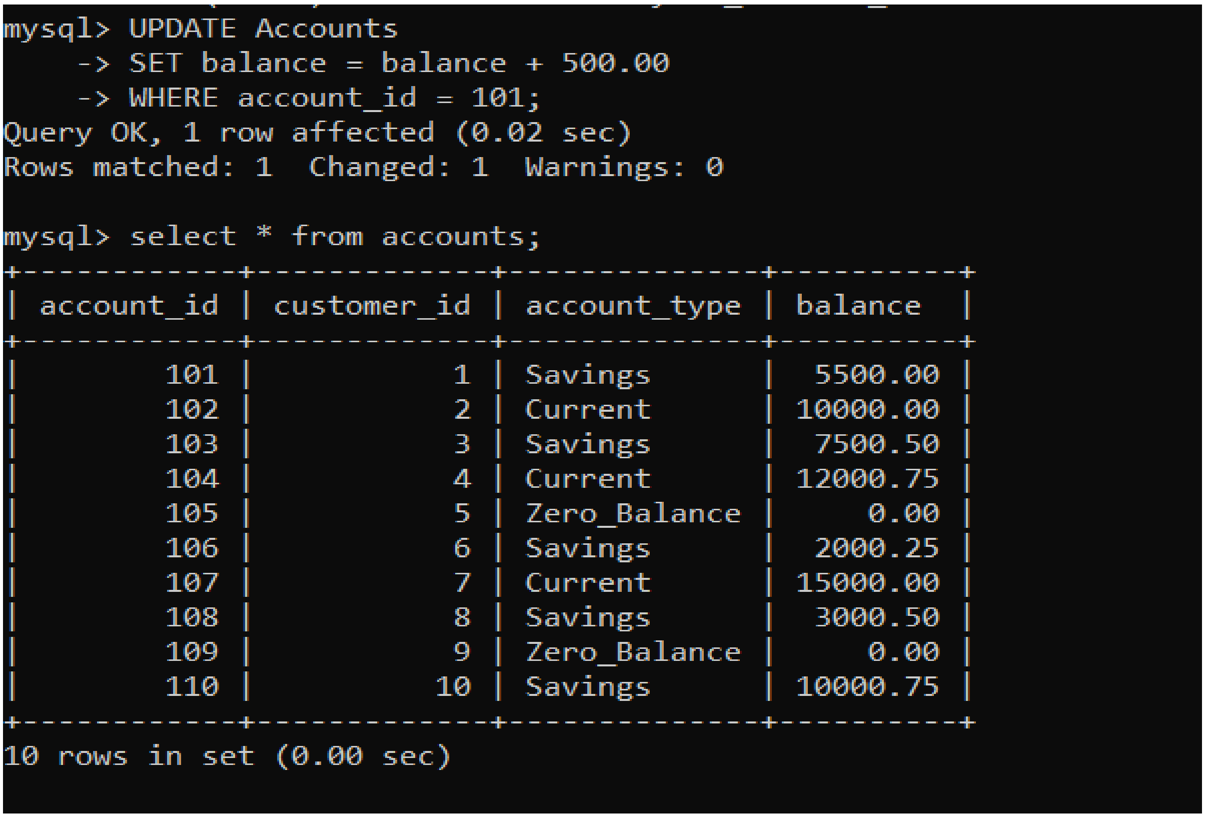
```
mysql> SELECT
    ->     CONCAT(Customers.first_name, ' ', Customers.last_name) AS customer_name,
    ->     Accounts.account_type,
    ->     Customers.email
    -> FROM Customers
    -> LEFT JOIN Accounts ON Customers.customer_id = Accounts.customer_id;
+-----------------+--------------+--------------------------+
| customer_name   | account_type | email                    |
+-----------------+--------------+--------------------------+
| SUSHANT Kumar   | Savings      | sushantkumar@gmail.com   |
| Aditi Sharma    | Current      | aditi.sharma@gmail.com   |
| Amit Patel      | Savings      | amit.patel@email.com     |
| Ananya Gupta    | Current      | ananya.gupta@email.com   |
| Arjun Singh     | Zero_Balance | arjun.singh@email.com    |
| Bhavya Mishra   | Savings      | bhavya.mishra@email.com  |
| Chetan Rajput   | Current      | chetan.rajput@email.com  |
| Deepika Rathore | Savings      | deepika.rathore@email.com |
| Esha Verma      | Zero_Balance | esha.verma@email.com     |
| Farhan Malik    | Savings      | farhan.malik@email.com   |
+-----------------+--------------+--------------------------+
10 rows in set (0.01 sec)

mysql>
```

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



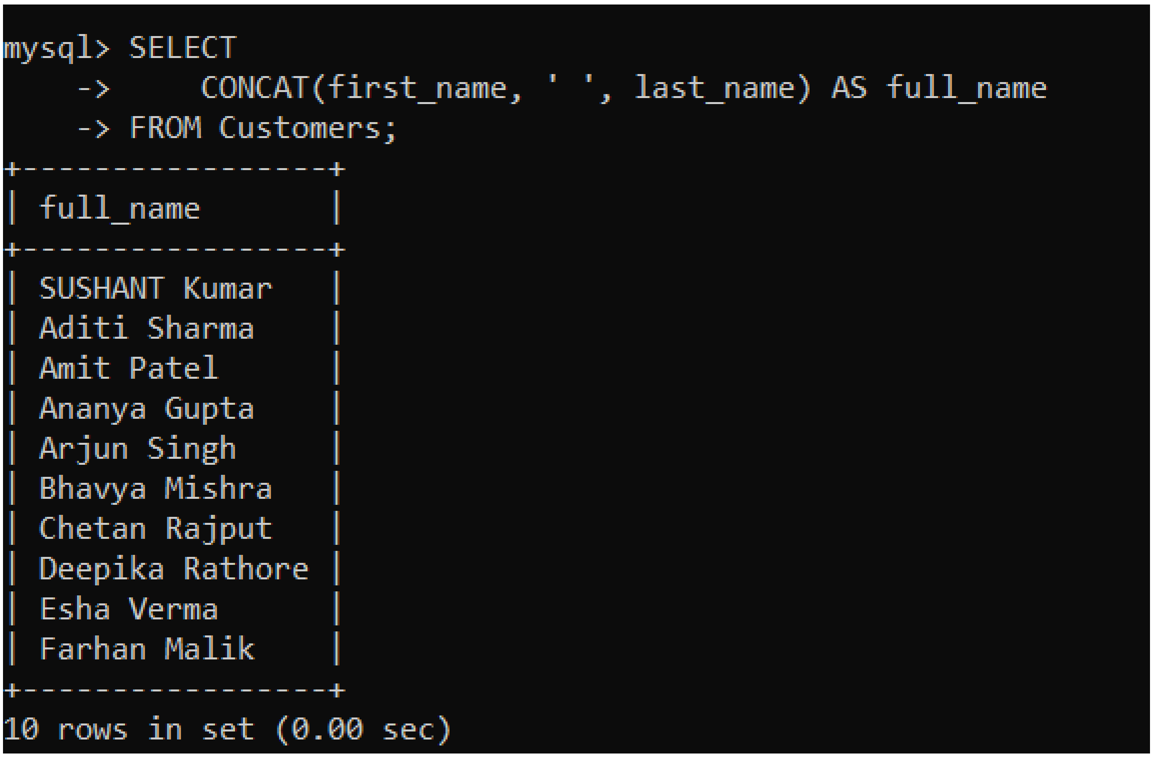
```
mysql> SELECT
    ->     CONCAT(Customers.first_name, ' ', Customers.last_name) AS customer_name,
    ->     Transactions.transaction_id,
    ->     Transactions.transaction_type,
    ->     Transactions.amount,
    ->     Transactions.transaction_date
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id
    -> JOIN Transactions ON Accounts.account_id = Transactions.account_id;
+-----------------+----------------+------------------+---------+---------------------+
| customer_name   | transaction_id | transaction_type | amount  | transaction_date    |
+-----------------+----------------+------------------+---------+---------------------+
| SUSHANT Kumar   |            301 | Deposit          | 2000.00 | 2024-01-01 10:30:00 |
| Aditi Sharma    |            302 | Transfer         |  250.00 | 2024-01-10 15:25:00 |
| Amit Patel      |            303 | Deposit          | 1000.25 | 2024-01-03 09:15:00 |
| Ananya Gupta    |            304 | Withdrawal       |  800.75 | 2024-01-04 14:20:00 |
| Arjun Singh     |            305 | Deposit          |  500.50 | 2024-01-05 11:10:00 |
| Bhavya Mishra   |            306 | Withdrawal       |  300.25 | 2024-01-06 13:55:00 |
| Chetan Rajput   |            307 | Deposit          | 1200.00 | 2024-01-07 08:45:00 |
| Deepika Rathore |            308 | Withdrawal       |  700.50 | 2024-01-08 16:30:00 |
| Esha Verma      |            309 | Deposit          |  400.25 | 2024-01-09 10:00:00 |
| Farhan Malik    |            310 | Withdrawal       | 1000.75 | 2024-01-10 15:25:00 |
+-----------------+----------------+------------------+---------+---------------------+
10 rows in set (0.01 sec)
```

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



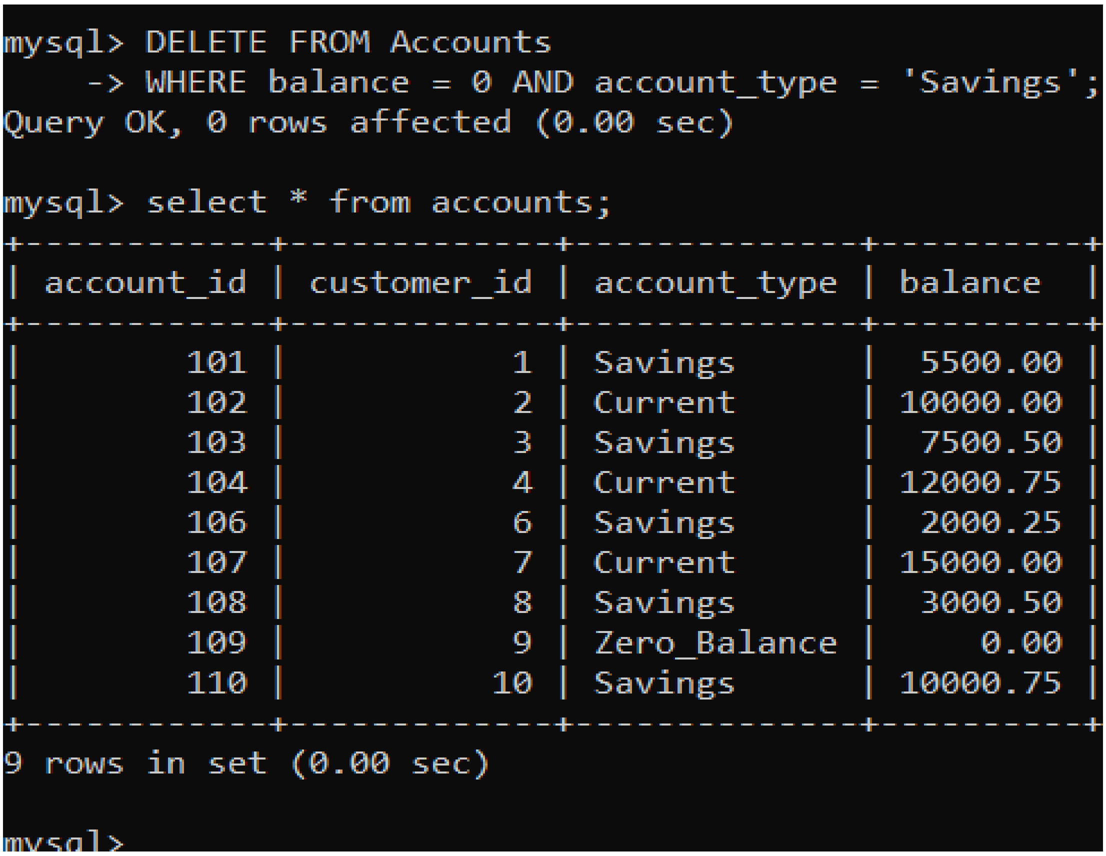
```
mysql> UPDATE Accounts
    -> SET balance = balance + 500.00
    -> WHERE account_id = 101;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from accounts;
+------------+-------------+--------------+----------+
| account_id | customer_id | account_type | balance  |
+------------+-------------+--------------+----------+
|        101 |           1 | Savings      |  5500.00 |
|        102 |           2 | Current      | 10000.00 |
|        103 |           3 | Savings      |  7500.50 |
|        104 |           4 | Current      | 12000.75 |
|        105 |           5 | Zero_Balance |     0.00 |
|        106 |           6 | Savings      |  2000.25 |
|        107 |           7 | Current      | 15000.00 |
|        108 |           8 | Savings      |  3000.50 |
|        109 |           9 | Zero_Balance |     0.00 |
|        110 |          10 | Savings      | 10000.75 |
+------------+-------------+--------------+----------+
10 rows in set (0.00 sec)
```

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



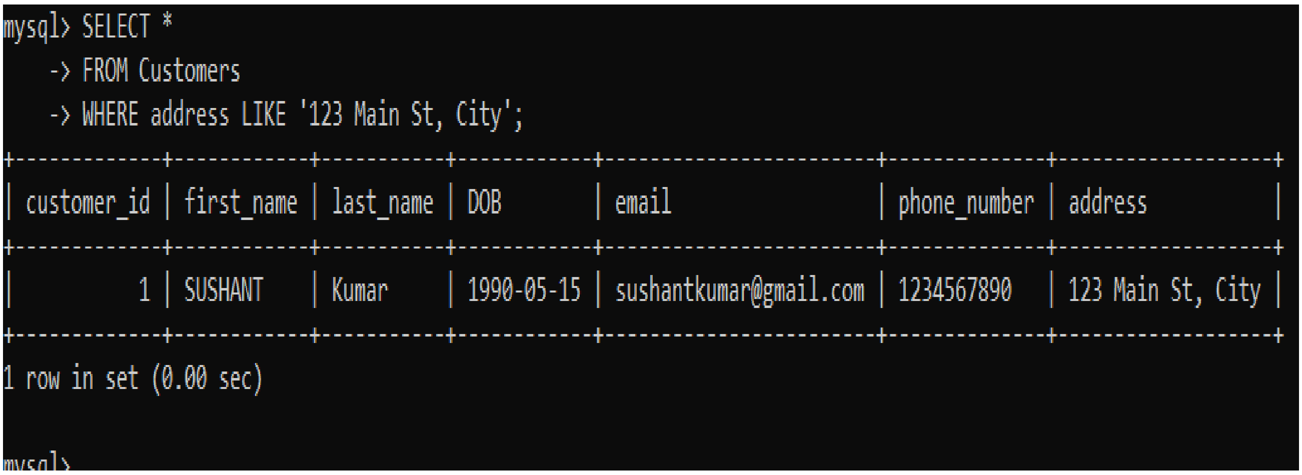
```
mysql> SELECT
    ->     CONCAT(first_name, ' ', last_name) AS full_name
    -> FROM Customers;
+-----------------+
| full_name       |
+-----------------+
| SUSHANT Kumar   |
| Aditi Sharma    |
| Amit Patel      |
| Ananya Gupta    |
| Arjun Singh     |
| Bhavya Mishra   |
| Chetan Rajput   |
| Deepika Rathore |
| Esha Verma      |
| Farhan Malik    |
+-----------------+
10 rows in set (0.00 sec)
```

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



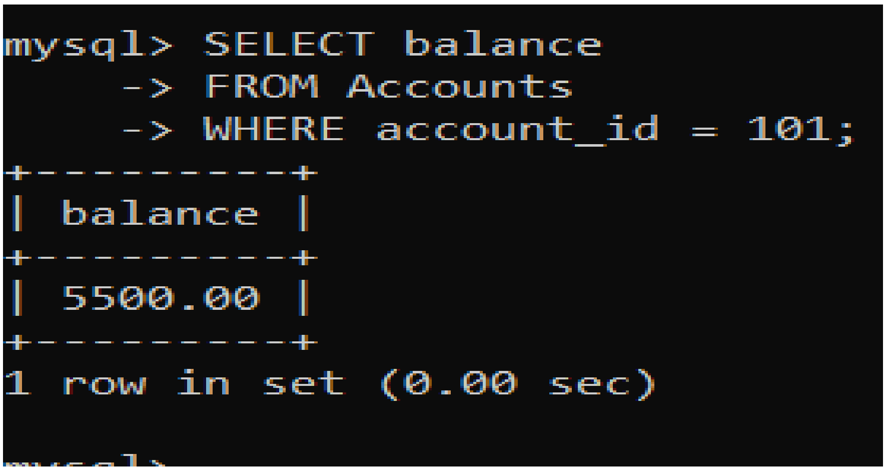
```
mysql> DELETE FROM Accounts
    -> WHERE balance = 0 AND account_type = 'Savings';
Query OK, 0 rows affected (0.00 sec)

mysql> select * from accounts;
+------------+-------------+--------------+----------+
| account_id | customer_id | account_type | balance  |
+------------+-------------+--------------+----------+
|        101 |           1 | Savings      |  5500.00 |
|        102 |           2 | Current      | 10000.00 |
|        103 |           3 | Savings      |  7500.50 |
|        104 |           4 | Current      | 12000.75 |
|        106 |           6 | Savings      |  2000.25 |
|        107 |           7 | Current      | 15000.00 |
|        108 |           8 | Savings      |  3000.50 |
|        109 |           9 | Zero_Balance |     0.00 |
|        110 |          10 | Savings      | 10000.75 |
+------------+-------------+--------------+----------+
9 rows in set (0.00 sec)

mysql>
```

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



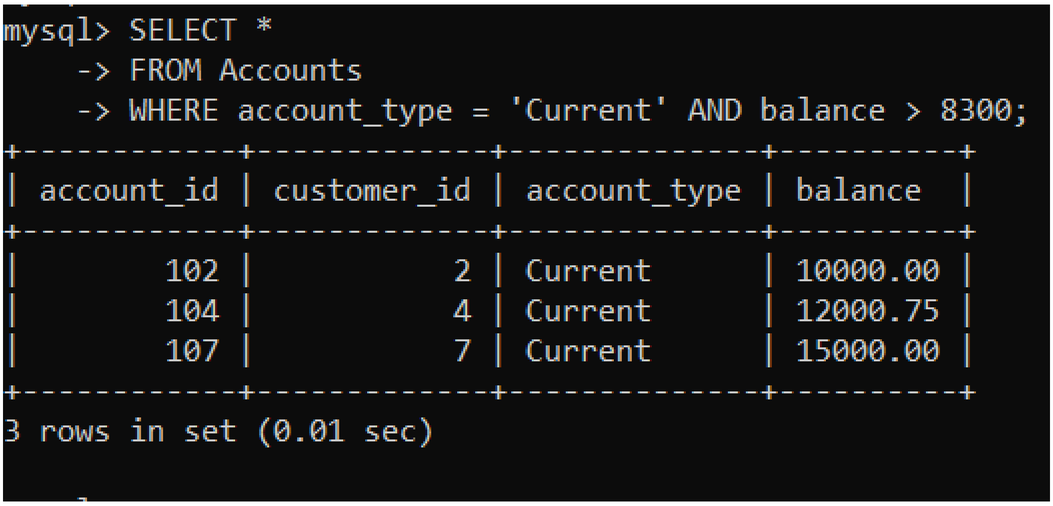
```
mysql> SELECT *
    -> FROM Customers
    -> WHERE address LIKE '123 Main St, City';
+-------------+------------+-----------+------------+------------------------+--------------+-------------------+
| customer_id | first_name | last_name | DOB        | email                  | phone_number | address           |
+-------------+------------+-----------+------------+------------------------+--------------+-------------------+
|           1 | SUSHANT    | Kumar     | 1990-05-15 | sushantkumar@gmail.com | 1234567890   | 123 Main St, City |
+-------------+------------+-----------+------------+------------------------+--------------+-------------------+
1 row in set (0.00 sec)

mysql>
```

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



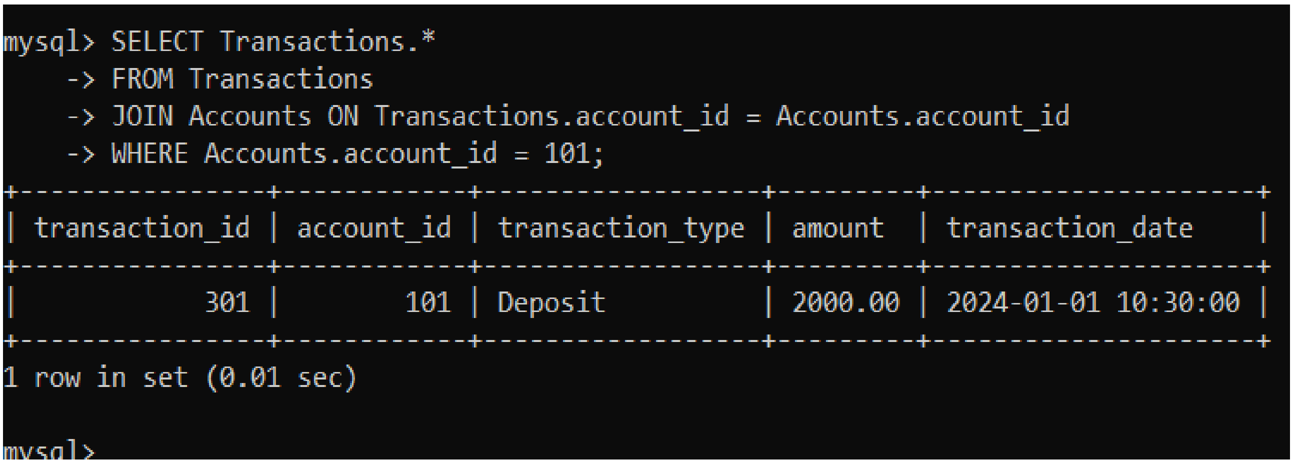
```
mysql> SELECT balance
    -> FROM Accounts
    -> WHERE account_id = 101;
+----------+
| balance  |
+----------+
| 5500.00  |
+----------+
1 row in set (0.00 sec)

mysql>
```

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



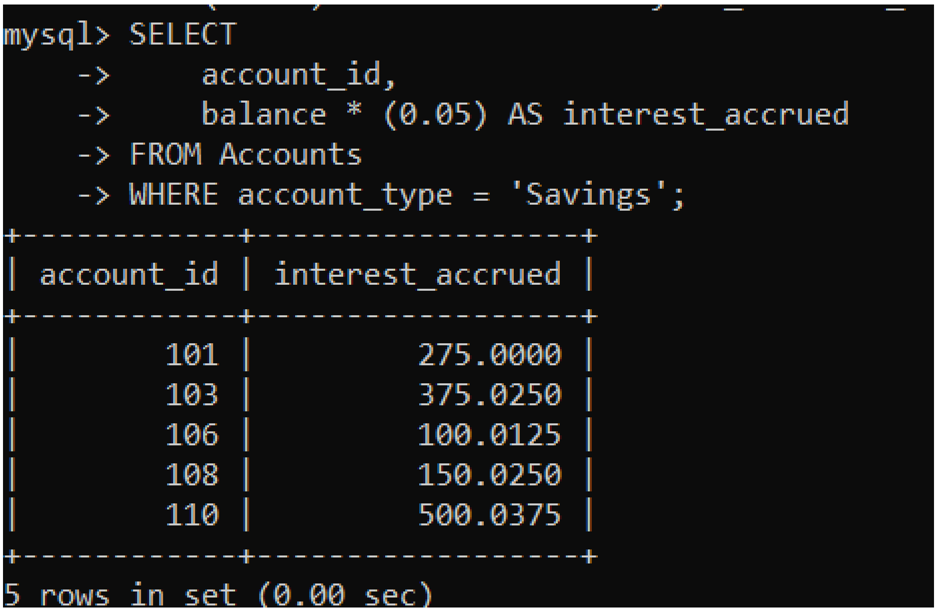
```
mysql> SELECT *
    -> FROM Accounts
    -> WHERE account_type = 'Current' AND balance > 8300;
+------------+-------------+--------------+----------+
| account_id | customer_id | account_type | balance  |
+------------+-------------+--------------+----------+
|        102 |           2 | Current      | 10000.00 |
|        104 |           4 | Current      | 12000.75 |
|        107 |           7 | Current      | 15000.00 |
+------------+-------------+--------------+----------+
3 rows in set (0.01 sec)
```

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



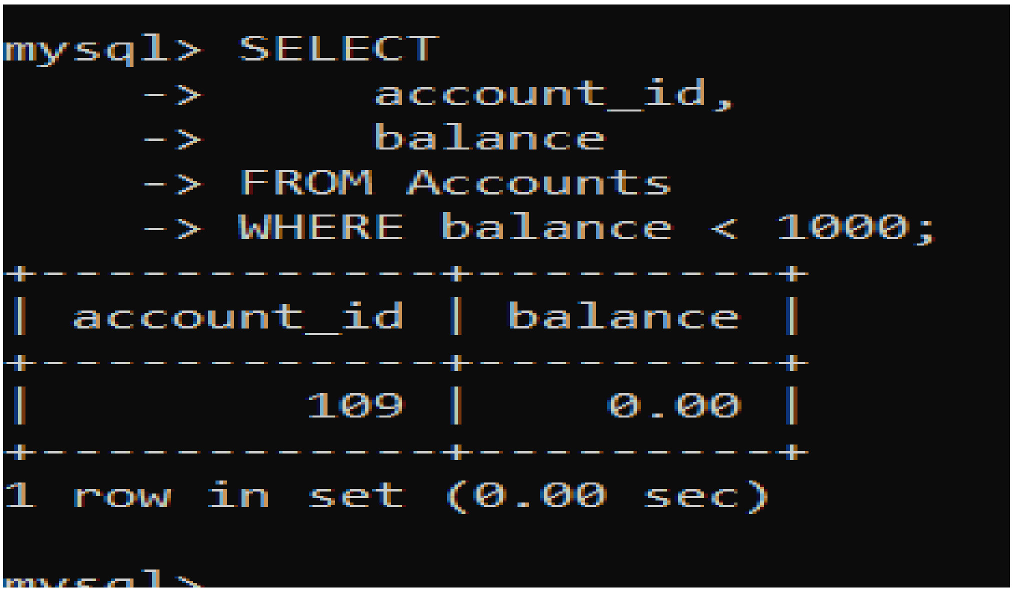
```
mysql> SELECT Transactions.*
    -> FROM Transactions
    -> JOIN Accounts ON Transactions.account_id = Accounts.account_id
    -> WHERE Accounts.account_id = 101;
+----------------+------------+------------------+---------+---------------------+
| transaction_id | account_id | transaction_type | amount  | transaction_date    |
+----------------+------------+------------------+---------+---------------------+
|            301 |        101 | Deposit          | 2000.00 | 2024-01-01 10:30:00 |
+----------------+------------+------------------+---------+---------------------+
1 row in set (0.01 sec)

mysql>
```

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



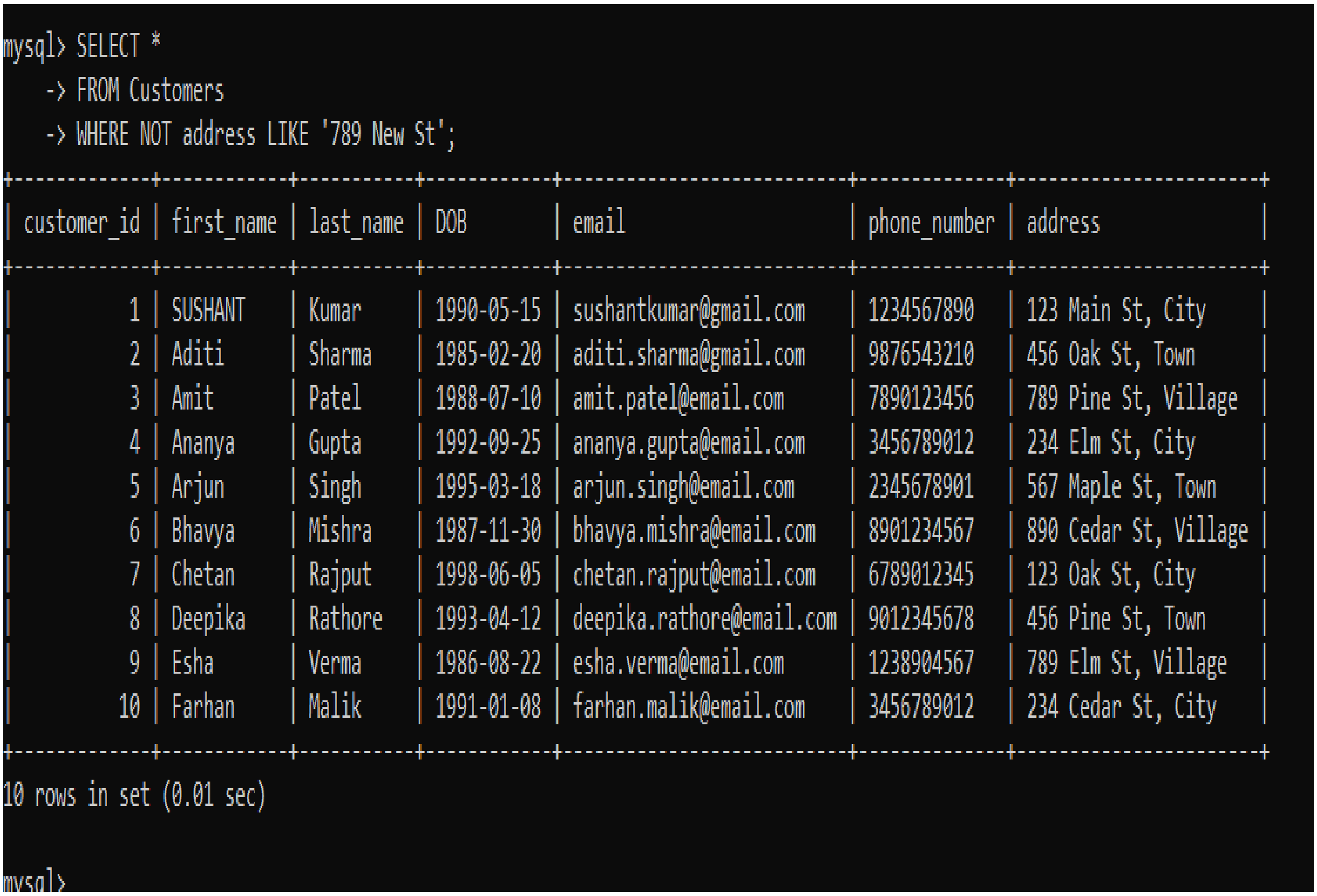
```
mysql> SELECT
    ->     account_id,
    ->     balance * (0.05) AS interest_accrued
    -> FROM Accounts
    -> WHERE account_type = 'Savings';
+------------+------------------+
| account_id | interest_accrued |
+------------+------------------+
|        101 |         275.0000 |
|        103 |         375.0250 |
|        106 |         100.0125 |
|        108 |         150.0250 |
|        110 |         500.0375 |
+------------+------------------+
5 rows in set (0.00 sec)
```

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



```
mysql> SELECT
    ->     account_id,
    ->     balance
    -> FROM Accounts
    -> WHERE balance < 1000;
+------------+---------+
| account_id | balance |
+------------+---------+
|        109 |    0.00 |
+------------+---------+
1 row in set (0.00 sec)

mysql>
```

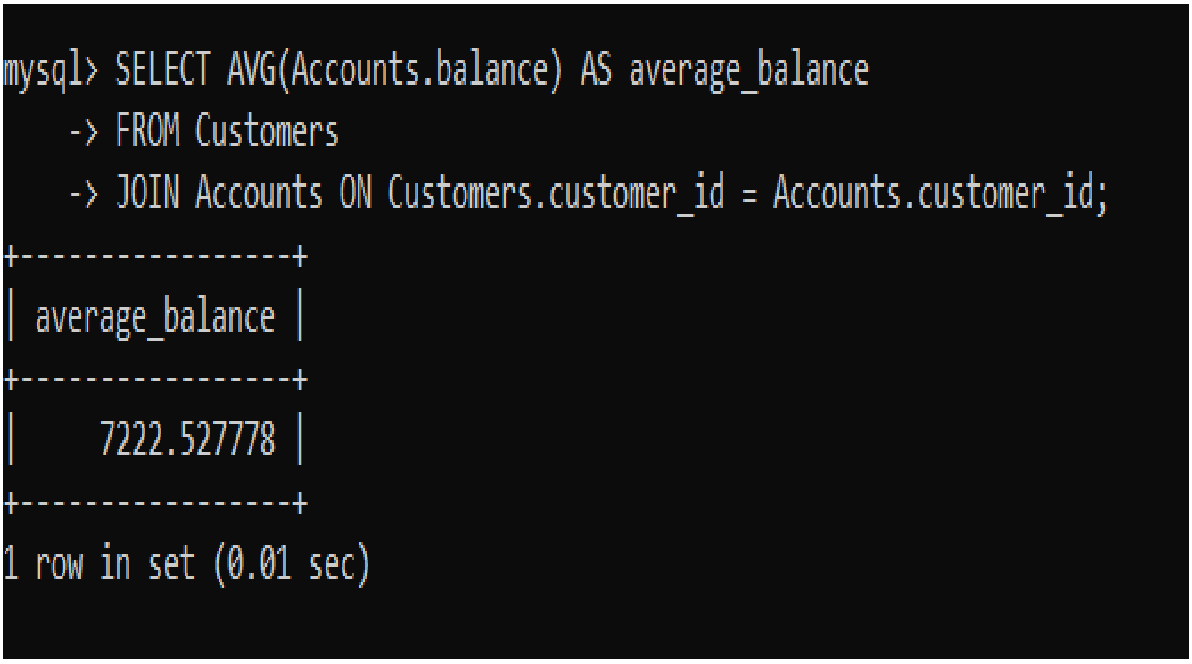
12. Write a SQL query to Find customers not living in a specific city



```
mysql> SELECT *
    -> FROM Customers
    -> WHERE NOT address LIKE '789 New St';
+-------------+------------+-----------+------------+---------------------------+--------------+----------------------+
| customer_id | first_name | last_name | DOB        | email                     | phone_number | address              |
+-------------+------------+-----------+------------+---------------------------+--------------+----------------------+
|           1 | SUSHANT    | Kumar     | 1990-05-15 | sushantkumar@gmail.com    | 1234567890   | 123 Main St, City    |
|           2 | Aditi      | Sharma    | 1985-02-20 | aditi.sharma@gmail.com    | 9876543210   | 456 Oak St, Town     |
|           3 | Amit       | Patel     | 1988-07-10 | amit.patel@email.com      | 7890123456   | 789 Pine St, Village |
|           4 | Ananya     | Gupta     | 1992-09-25 | ananya.gupta@email.com    | 3456789012   | 234 Elm St, City     |
|           5 | Arjun      | Singh     | 1995-03-18 | arjun.singh@email.com     | 2345678901   | 567 Maple St, Town   |
|           6 | Bhavya     | Mishra    | 1987-11-30 | bhavya.mishra@email.com   | 8901234567   | 890 Cedar St, Village |
|           7 | Chetan     | Rajput    | 1998-06-05 | chetan.rajput@email.com   | 6789012345   | 123 Oak St, City     |
|           8 | Deepika    | Rathore   | 1993-04-12 | deepika.rathore@email.com | 9012345678   | 456 Pine St, Town    |
|           9 | Esha       | Verma     | 1986-08-22 | esha.verma@email.com      | 1238904567   | 789 Elm St, Village  |
|          10 | Farhan     | Malik     | 1991-01-08 | farhan.malik@email.com    | 3456789012   | 234 Cedar St, City   |
+-------------+------------+-----------+------------+---------------------------+--------------+----------------------+
10 rows in set (0.01 sec)

mysql>
```

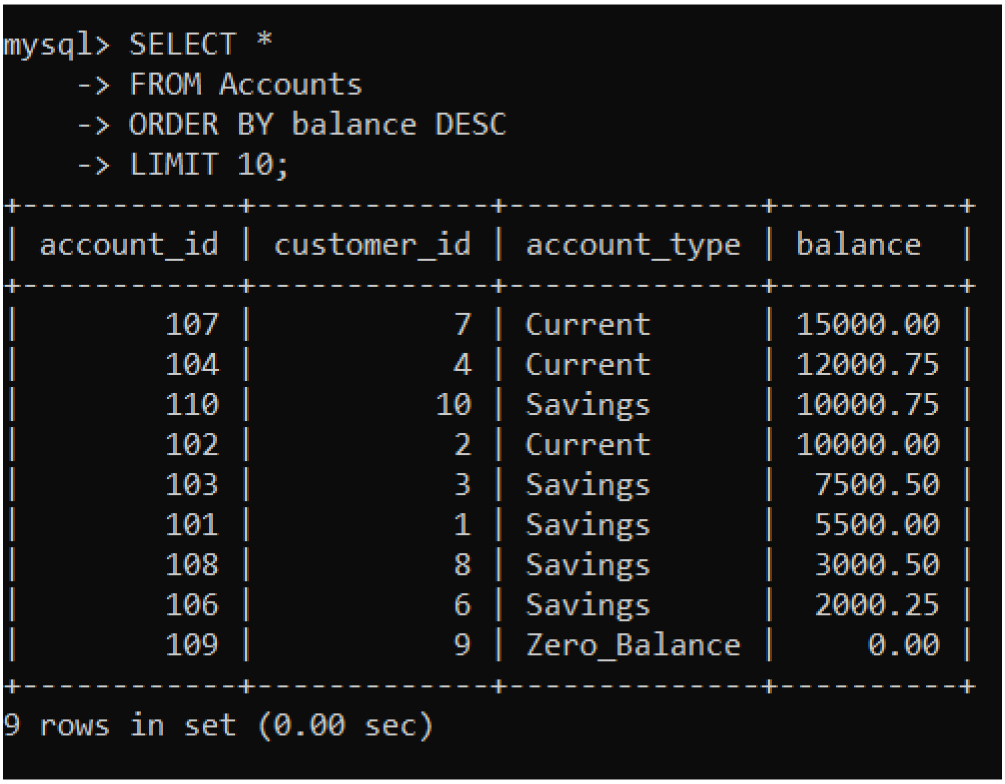
Tasks 3: Aggregate functions, Having, Order By, GroupBy and Joins:

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



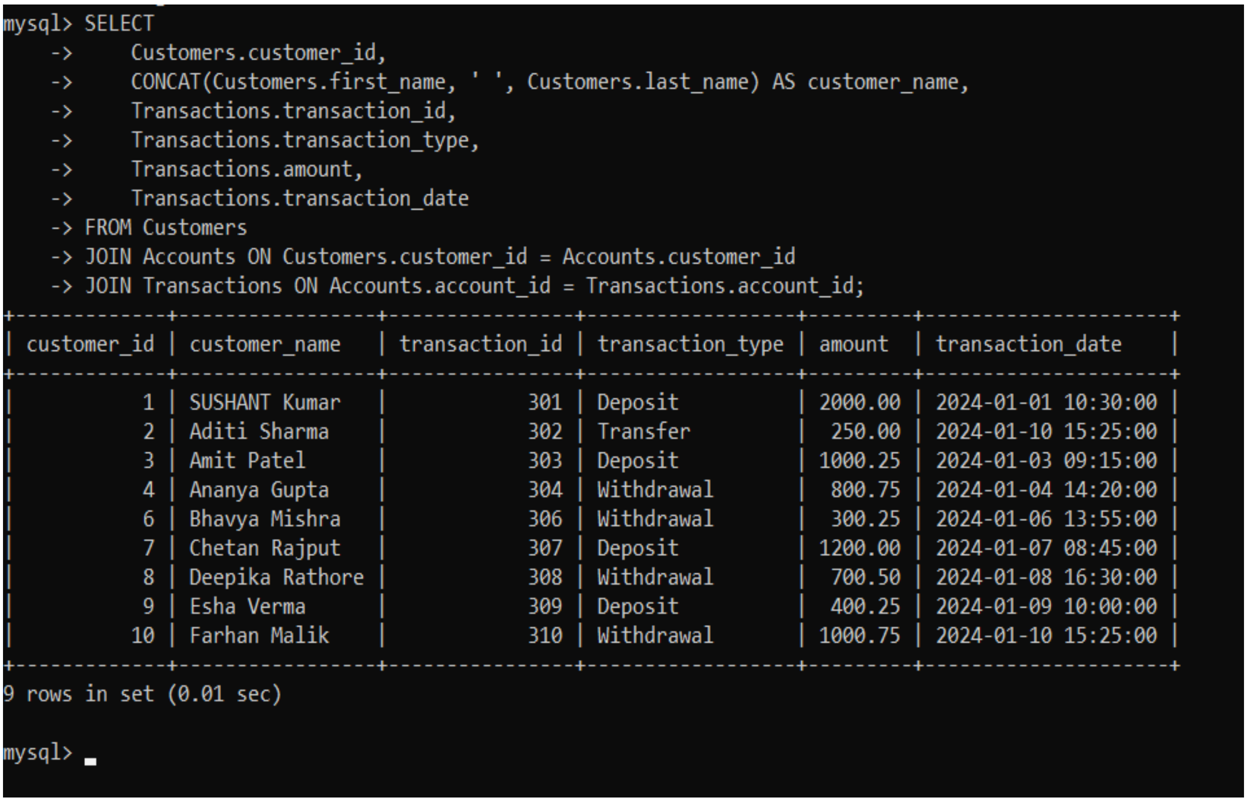
```
mysql> SELECT AVG(Accounts.balance) AS average_balance
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id;
+-----------------+
| average_balance |
+-----------------+
|     7222.527778 |
+-----------------+
1 row in set (0.01 sec)
```

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



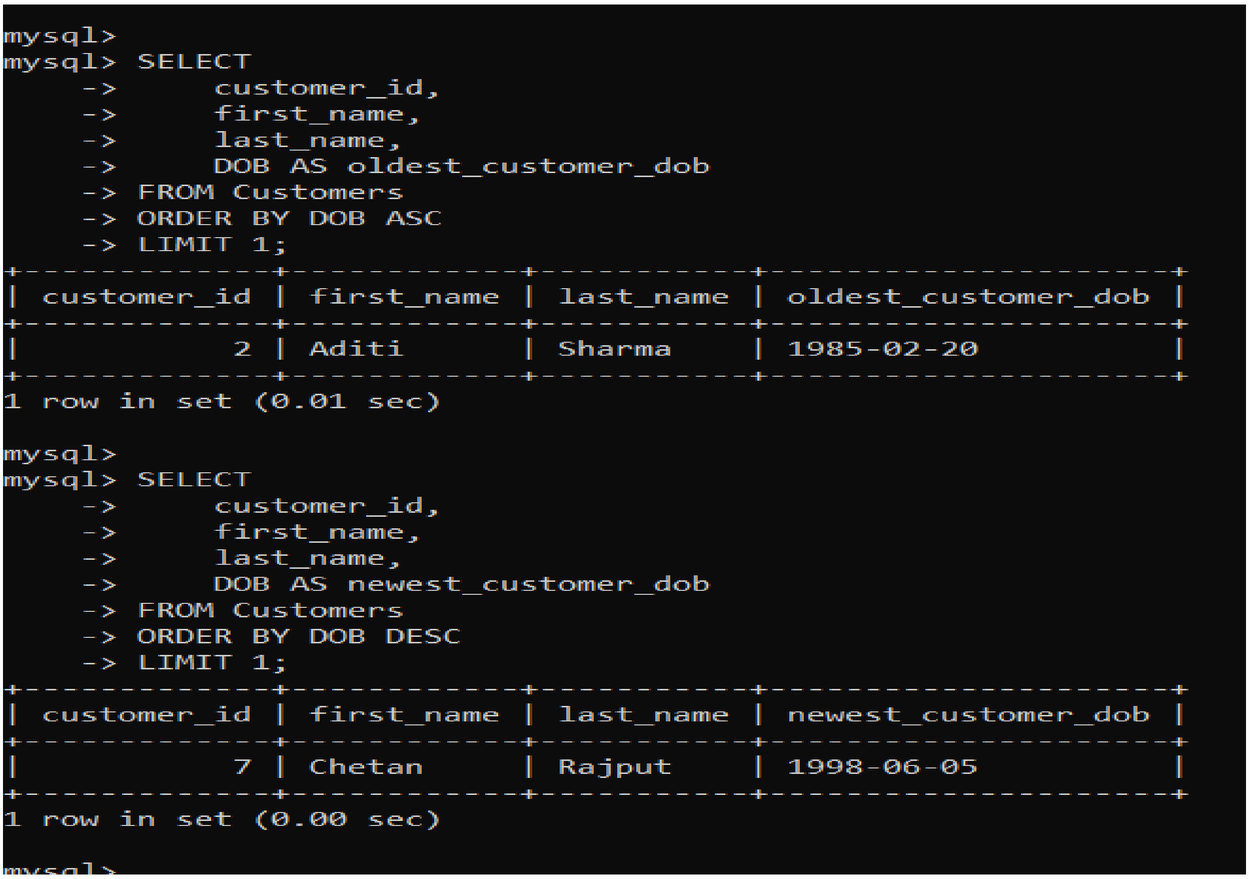
```
mysql> SELECT *
    -> FROM Accounts
    -> ORDER BY balance DESC
    -> LIMIT 10;
+------------+-------------+--------------+----------+
| account_id | customer_id | account_type | balance  |
+------------+-------------+--------------+----------+
|        107 |           7 | Current      | 15000.00 |
|        104 |           4 | Current      | 12000.75 |
|        110 |          10 | Savings      | 10000.75 |
|        102 |           2 | Current      | 10000.00 |
|        103 |           3 | Savings      |  7500.50 |
|        101 |           1 | Savings      |  5500.00 |
|        108 |           8 | Savings      |  3000.50 |
|        106 |           6 | Savings      |  2000.25 |
|        109 |           9 | Zero_Balance |     0.00 |
+------------+-------------+--------------+----------+
9 rows in set (0.00 sec)
```

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



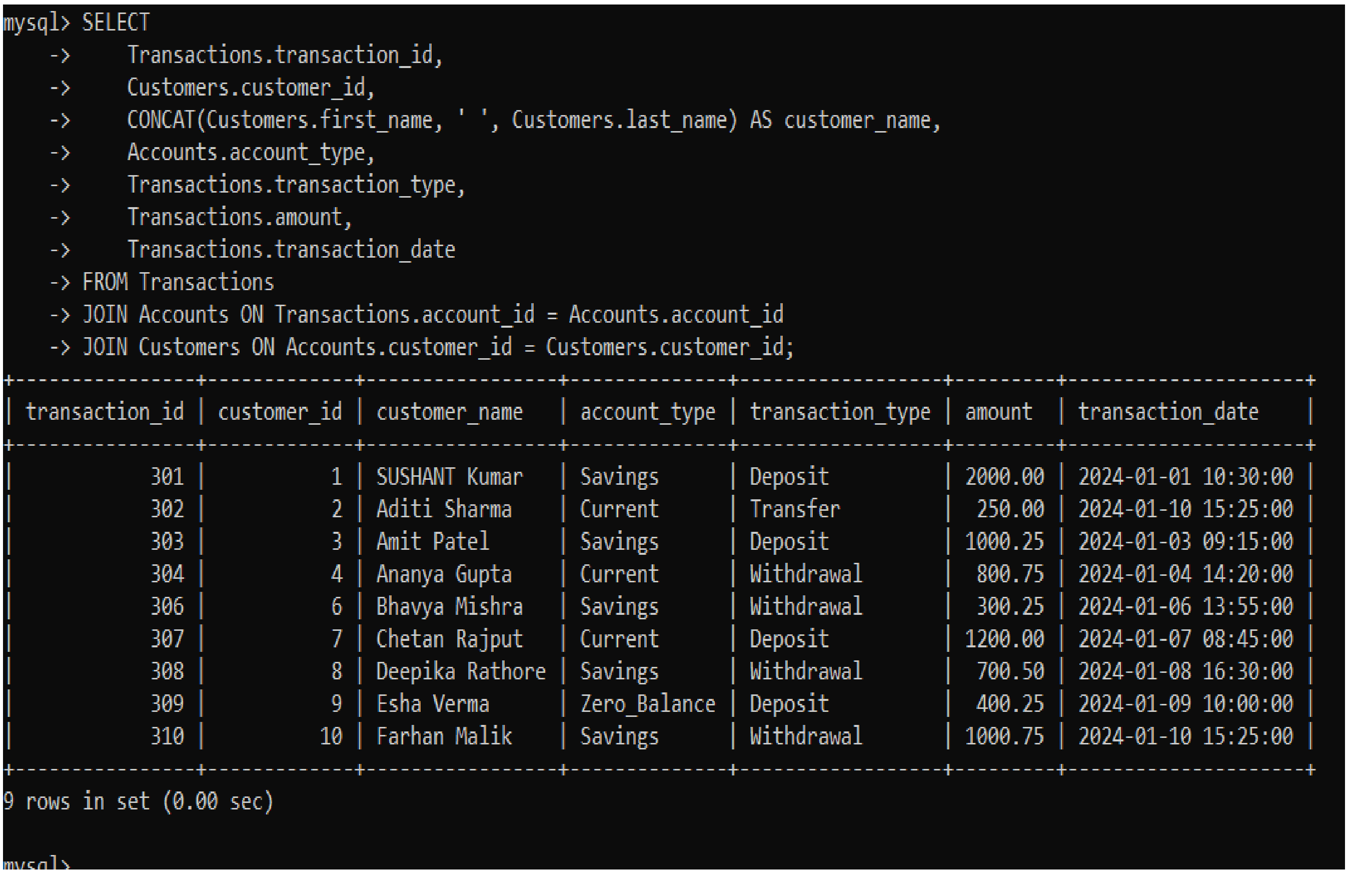
```
mysql> SELECT
    ->     Customers.customer_id,
    ->     CONCAT(Customers.first_name, ' ', Customers.last_name) AS customer_name,
    ->     Transactions.transaction_id,
    ->     Transactions.transaction_type,
    ->     Transactions.amount,
    ->     Transactions.transaction_date
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id
    -> JOIN Transactions ON Accounts.account_id = Transactions.account_id;
+-------------+-----------------+----------------+------------------+---------+---------------------+
| customer_id | customer_name   | transaction_id | transaction_type | amount  | transaction_date    |
+-------------+-----------------+----------------+------------------+---------+---------------------+
|           1 | SUSHANT Kumar   |            301 | Deposit          | 2000.00 | 2024-01-01 10:30:00 |
|           2 | Aditi Sharma    |            302 | Transfer         |  250.00 | 2024-01-10 15:25:00 |
|           3 | Amit Patel      |            303 | Deposit          | 1000.25 | 2024-01-03 09:15:00 |
|           4 | Ananya Gupta    |            304 | Withdrawal       |  800.75 | 2024-01-04 14:20:00 |
|           6 | Bhavya Mishra   |            306 | Withdrawal       |  300.25 | 2024-01-06 13:55:00 |
|           7 | Chetan Rajput   |            307 | Deposit          | 1200.00 | 2024-01-07 08:45:00 |
|           8 | Deepika Rathore |            308 | Withdrawal       |  700.50 | 2024-01-08 16:30:00 |
|           9 | Esha Verma      |            309 | Deposit          |  400.25 | 2024-01-09 10:00:00 |
|          10 | Farhan Malik    |            310 | Withdrawal       | 1000.75 | 2024-01-10 15:25:00 |
+-------------+-----------------+----------------+------------------+---------+---------------------+
9 rows in set (0.01 sec)

mysql>
```

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



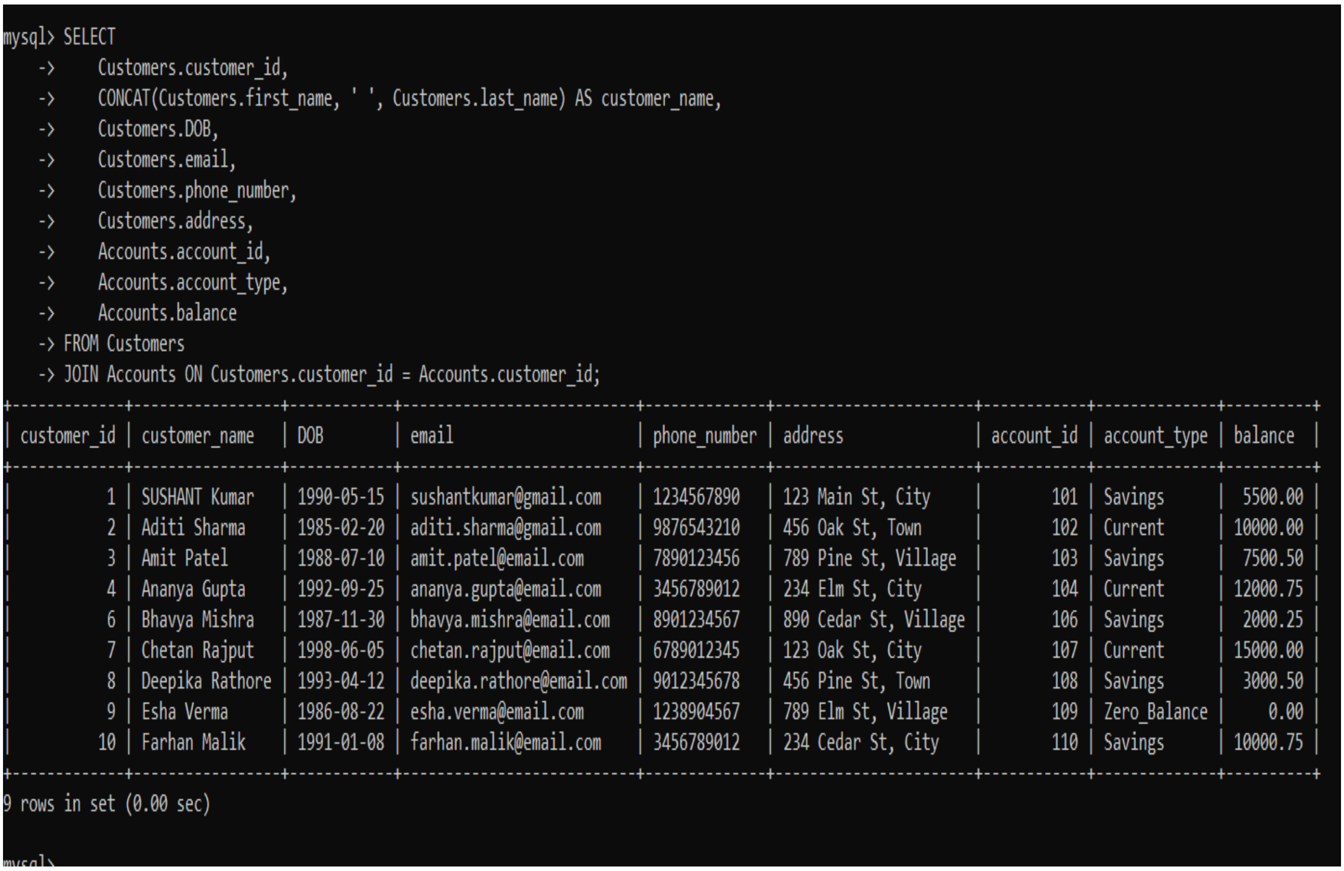
```
mysql>
mysql> SELECT
    ->     customer_id,
    ->     first_name,
    ->     last_name,
    ->     DOB AS oldest_customer_dob
    -> FROM Customers
    -> ORDER BY DOB ASC
    -> LIMIT 1;
+-------------+------------+-----------+---------------------+
| customer_id | first_name | last_name | oldest_customer_dob |
+-------------+------------+-----------+---------------------+
|           2 | Aditi      | Sharma    | 1985-02-20          |
+-------------+------------+-----------+---------------------+
1 row in set (0.01 sec)

mysql>
mysql> SELECT
    ->     customer_id,
    ->     first_name,
    ->     last_name,
    ->     DOB AS newest_customer_dob
    -> FROM Customers
    -> ORDER BY DOB DESC
    -> LIMIT 1;
+-------------+------------+-----------+---------------------+
| customer_id | first_name | last_name | newest_customer_dob |
+-------------+------------+-----------+---------------------+
|           7 | Chetan     | Rajput    | 1998-06-05          |
+-------------+------------+-----------+---------------------+
1 row in set (0.00 sec)

mysql>
```

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



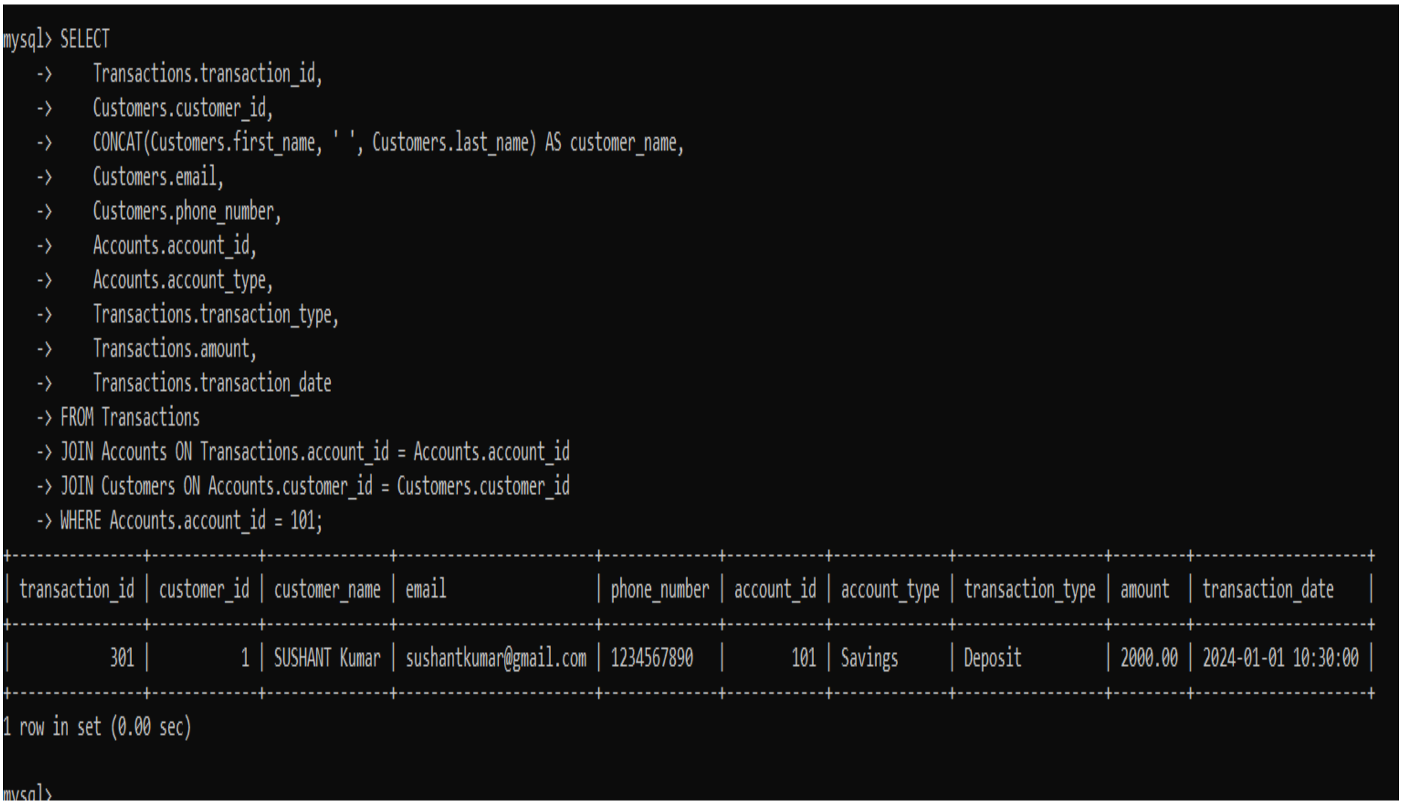
```
mysql> SELECT
    ->     Transactions.transaction_id,
    ->     Customers.customer_id,
    ->     CONCAT(Customers.first_name, ' ', Customers.last_name) AS customer_name,
    ->     Accounts.account_type,
    ->     Transactions.transaction_type,
    ->     Transactions.amount,
    ->     Transactions.transaction_date
    -> 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   | account_type | transaction_type | amount  | transaction_date    |
+----------------+-------------+-----------------+--------------+------------------+---------+---------------------+
|            301 |           1 | SUSHANT Kumar   | Savings      | Deposit          | 2000.00 | 2024-01-01 10:30:00 |
|            302 |           2 | Aditi Sharma    | Current      | Transfer         |  250.00 | 2024-01-10 15:25:00 |
|            303 |           3 | Amit Patel      | Savings      | Deposit          | 1000.25 | 2024-01-03 09:15:00 |
|            304 |           4 | Ananya Gupta    | Current      | Withdrawal       |  800.75 | 2024-01-04 14:20:00 |
|            306 |           6 | Bhavya Mishra   | Savings      | Withdrawal       |  300.25 | 2024-01-06 13:55:00 |
|            307 |           7 | Chetan Rajput   | Current      | Deposit          | 1200.00 | 2024-01-07 08:45:00 |
|            308 |           8 | Deepika Rathore | Savings      | Withdrawal       |  700.50 | 2024-01-08 16:30:00 |
|            309 |           9 | Esha Verma      | Zero_Balance | Deposit          |  400.25 | 2024-01-09 10:00:00 |
|            310 |          10 | Farhan Malik    | Savings      | Withdrawal       | 1000.75 | 2024-01-10 15:25:00 |
+----------------+-------------+-----------------+--------------+------------------+---------+---------------------+
9 rows in set (0.00 sec)

mysql>
```

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



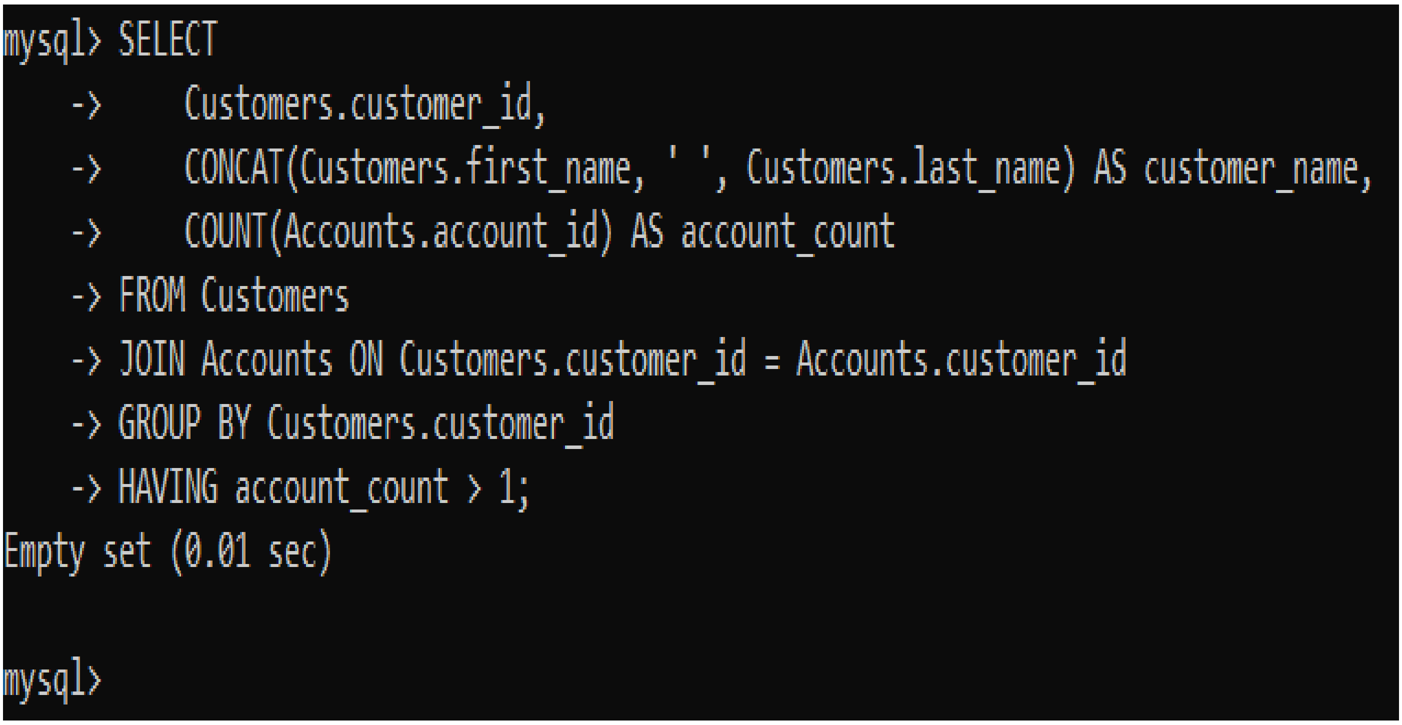
```
mysql> SELECT
    ->     Customers.customer_id,
    ->     CONCAT(Customers.first_name, ' ', Customers.last_name) AS customer_name,
    ->     Customers.DOB,
    ->     Customers.email,
    ->     Customers.phone_number,
    ->     Customers.address,
    ->     Accounts.account_id,
    ->     Accounts.account_type,
    ->     Accounts.balance
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id;
+-------------+-----------------+------------+---------------------------+--------------+-----------------------+------------+--------------+----------+
| customer_id | customer_name   | DOB        | email                     | phone_number | address               | account_id | account_type | balance  |
+-------------+-----------------+------------+---------------------------+--------------+-----------------------+------------+--------------+----------+
|           1 | SUSHANT Kumar   | 1990-05-15 | sushantkumar@gmail.com    | 1234567890   | 123 Main St, City     |        101 | Savings      |  5500.00 |
|           2 | Aditi Sharma    | 1985-02-20 | aditi.sharma@gmail.com    | 9876543210   | 456 Oak St, Town      |        102 | Current      | 10000.00 |
|           3 | Amit Patel      | 1988-07-10 | amit.patel@email.com      | 7890123456   | 789 Pine St, Village  |        103 | Savings      |  7500.50 |
|           4 | Ananya Gupta    | 1992-09-25 | ananya.gupta@email.com    | 3456789012   | 234 Elm St, City      |        104 | Current      | 12000.75 |
|           6 | Bhavya Mishra   | 1987-11-30 | bhavya.mishra@email.com   | 8901234567   | 890 Cedar St, Village |        106 | Savings      |  2000.25 |
|           7 | Chetan Rajput   | 1998-06-05 | chetan.rajput@email.com   | 6789012345   | 123 Oak St, City      |        107 | Current      | 15000.00 |
|           8 | Deepika Rathore | 1993-04-12 | deepika.rathore@email.com | 9012345678   | 456 Pine St, Town     |        108 | Savings      |  3000.50 |
|           9 | Esha Verma      | 1986-08-22 | esha.verma@email.com      | 1238904567   | 789 Elm St, Village   |        109 | Zero_Balance |     0.00 |
|          10 | Farhan Malik    | 1991-01-08 | farhan.malik@email.com    | 3456789012   | 234 Cedar St, City    |        110 | Savings      | 10000.75 |
+-------------+-----------------+------------+---------------------------+--------------+-----------------------+------------+--------------+----------+
9 rows in set (0.00 sec)

mysql>
```

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



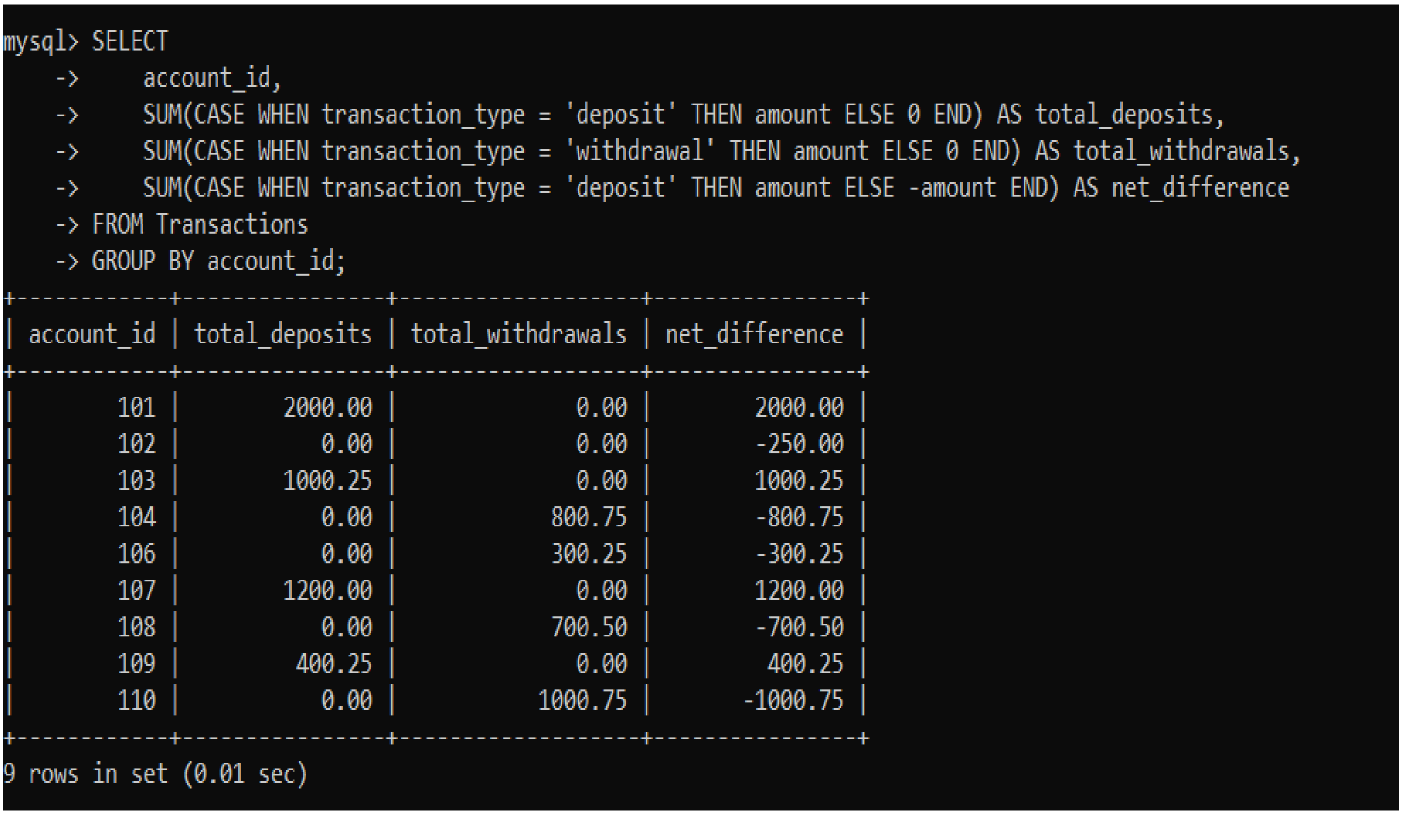
```
mysql> SELECT
    ->     Transactions.transaction_id,
    ->     Customers.customer_id,
    ->     CONCAT(Customers.first_name, ' ', Customers.last_name) AS customer_name,
    ->     Customers.email,
    ->     Customers.phone_number,
    ->     Accounts.account_id,
    ->     Accounts.account_type,
    ->     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
    -> WHERE Accounts.account_id = 101;
+----------------+-------------+---------------+------------------------+--------------+------------+--------------+------------------+---------+---------------------+
| transaction_id | customer_id | customer_name | email                  | phone_number | account_id | account_type | transaction_type | amount  | transaction_date    |
+----------------+-------------+---------------+------------------------+--------------+------------+--------------+------------------+---------+---------------------+
|            301 |           1 | SUSHANT Kumar | sushantkumar@gmail.com | 1234567890   |        101 | Savings      | Deposit          | 2000.00 | 2024-01-01 10:30:00 |
+----------------+-------------+---------------+------------------------+--------------+------------+--------------+------------------+---------+---------------------+
1 row in set (0.00 sec)

mysql>
```

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



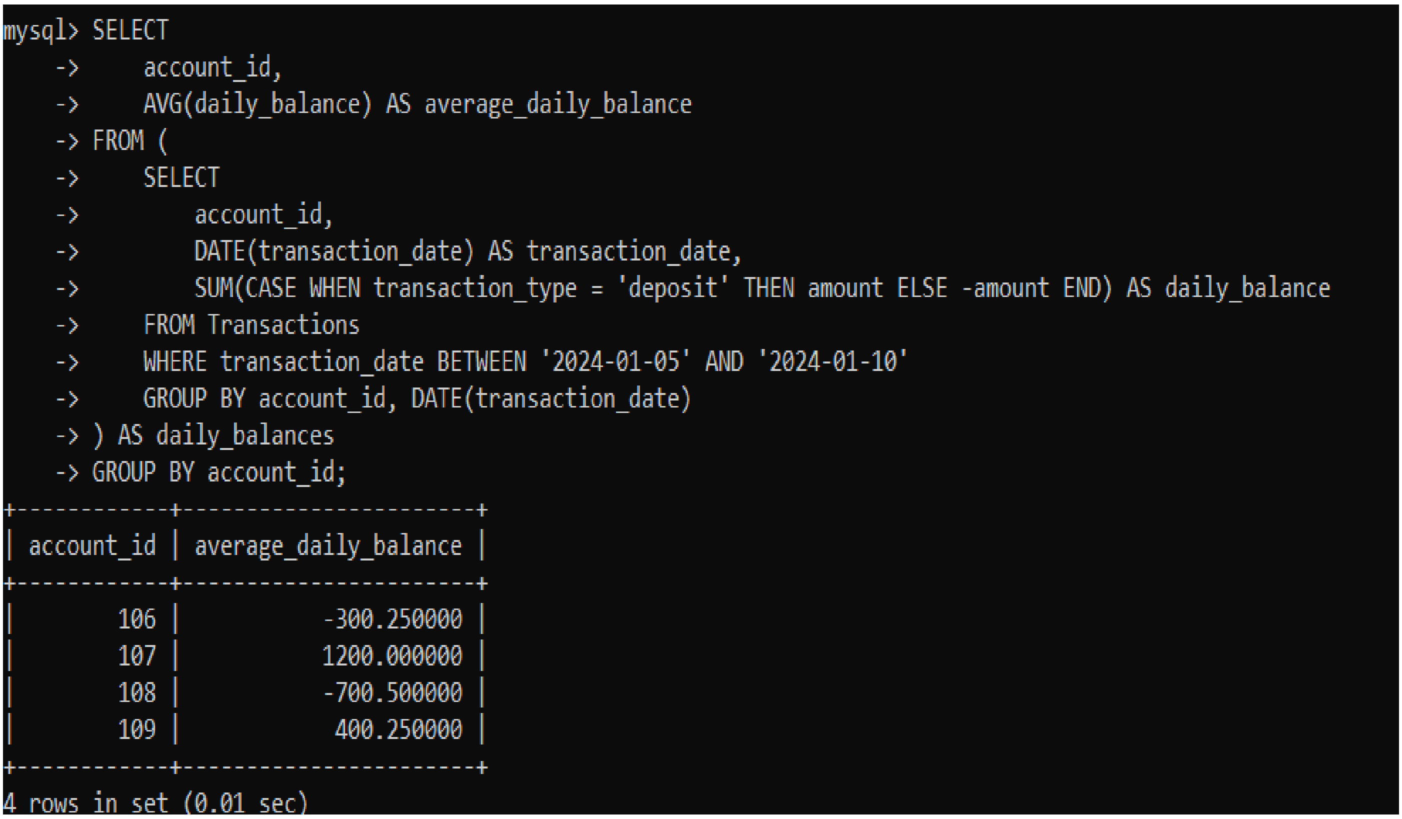
```
mysql> SELECT
    ->     Customers.customer_id,
    ->     CONCAT(Customers.first_name, ' ', Customers.last_name) AS customer_name,
    ->     COUNT(Accounts.account_id) AS account_count
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id
    -> GROUP BY Customers.customer_id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


mysql>
```

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



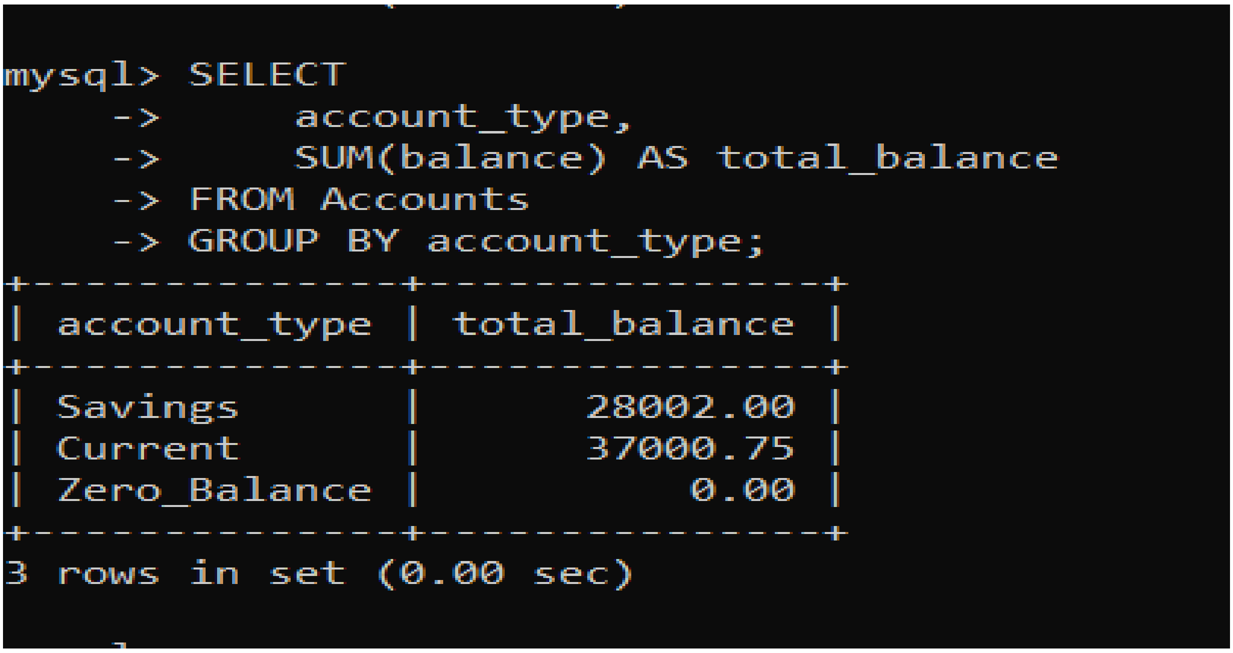
```
mysql> SELECT
    ->     account_id,
    ->     SUM(CASE WHEN transaction_type = 'deposit' THEN amount ELSE 0 END) AS total_deposits,
    ->     SUM(CASE WHEN transaction_type = 'withdrawal' THEN amount ELSE 0 END) AS total_withdrawals,
    ->     SUM(CASE WHEN transaction_type = 'deposit' THEN amount ELSE -amount END) AS net_difference
    -> FROM Transactions
    -> GROUP BY account_id;
+------------+----------------+-------------------+----------------+
| account_id | total_deposits | total_withdrawals | net_difference |
+------------+----------------+-------------------+----------------+
|        101 |        2000.00 |              0.00 |        2000.00 |
|        102 |           0.00 |              0.00 |        -250.00 |
|        103 |        1000.25 |              0.00 |        1000.25 |
|        104 |           0.00 |            800.75 |        -800.75 |
|        106 |           0.00 |            300.25 |        -300.25 |
|        107 |        1200.00 |              0.00 |        1200.00 |
|        108 |           0.00 |            700.50 |        -700.50 |
|        109 |         400.25 |              0.00 |         400.25 |
|        110 |           0.00 |           1000.75 |       -1000.75 |
+------------+----------------+-------------------+----------------+
9 rows in set (0.01 sec)
```

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



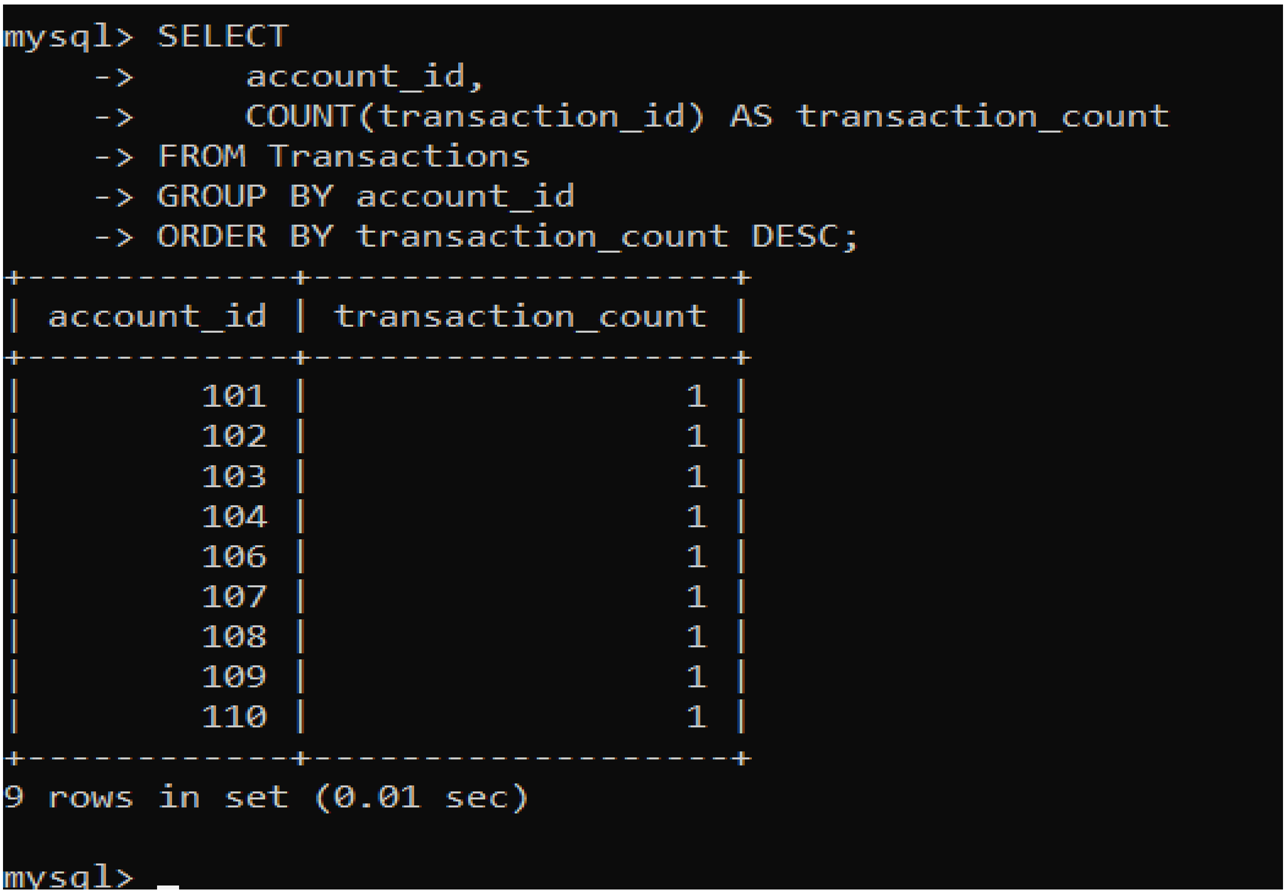
```
mysql> SELECT
    ->     account_id,
    ->     AVG(daily_balance) AS average_daily_balance
    -> FROM (
    ->     SELECT
    ->         account_id,
    ->         DATE(transaction_date) AS transaction_date,
    ->         SUM(CASE WHEN transaction_type = 'deposit' THEN amount ELSE -amount END) AS daily_balance
    ->     FROM Transactions
    ->     WHERE transaction_date BETWEEN '2024-01-05' AND '2024-01-10'
    ->     GROUP BY account_id, DATE(transaction_date)
    -> ) AS daily_balances
    -> GROUP BY account_id;
+------------+-----------------------+
| account_id | average_daily_balance |
+------------+-----------------------+
|        106 |           -300.250000 |
|        107 |           1200.000000 |
|        108 |           -700.500000 |
|        109 |            400.250000 |
+------------+-----------------------+
4 rows in set (0.01 sec)
```

11. Calculate the total balance for each account type.



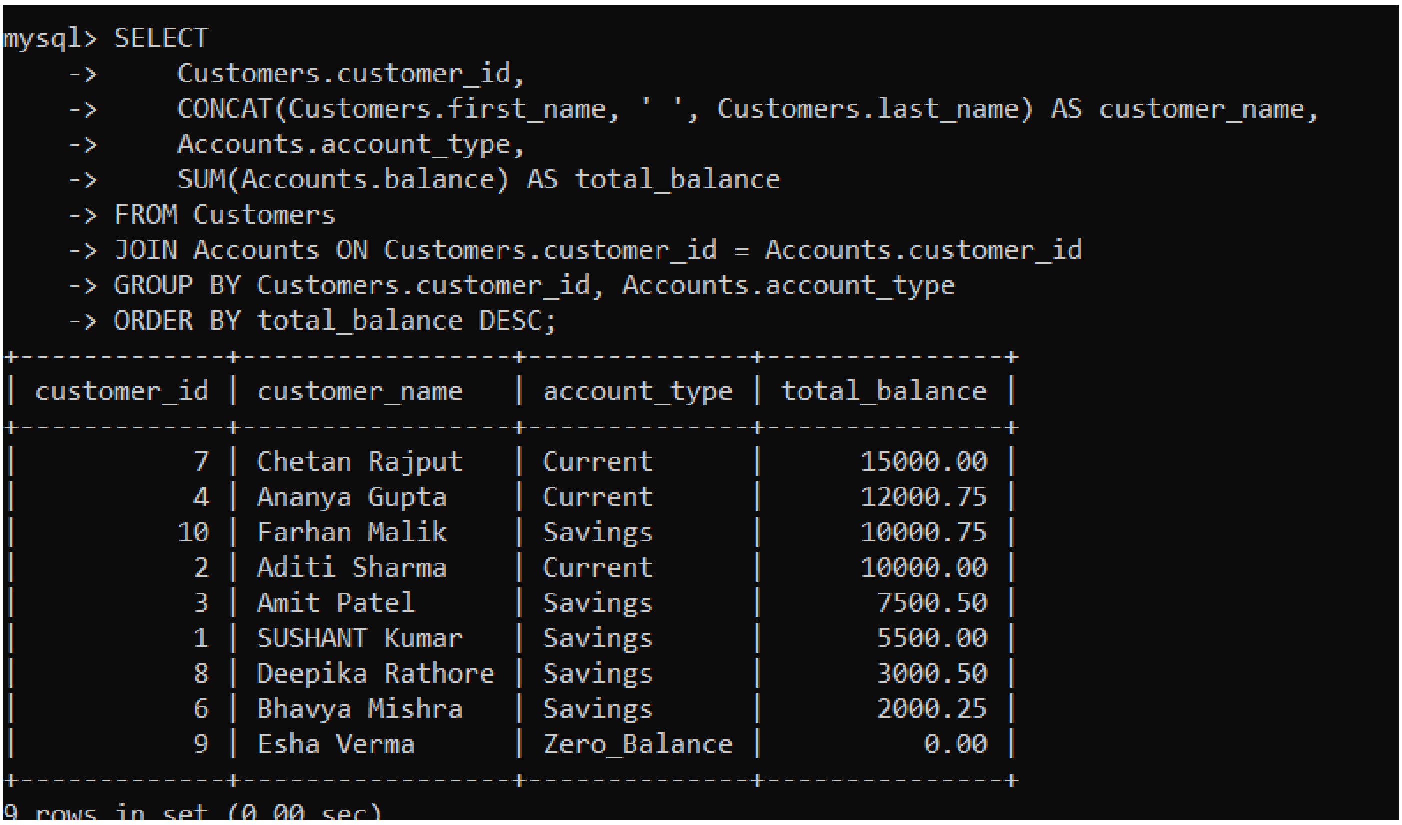
```
mysql> SELECT
    ->     account_type,
    ->     SUM(balance) AS total_balance
    -> FROM Accounts
    -> GROUP BY account_type;
+--------------+---------------+
| account_type | total_balance |
+--------------+---------------+
| Savings      |      28002.00 |
| Current      |      37000.75 |
| Zero_Balance |          0.00 |
+--------------+---------------+
3 rows in set (0.00 sec)
```

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



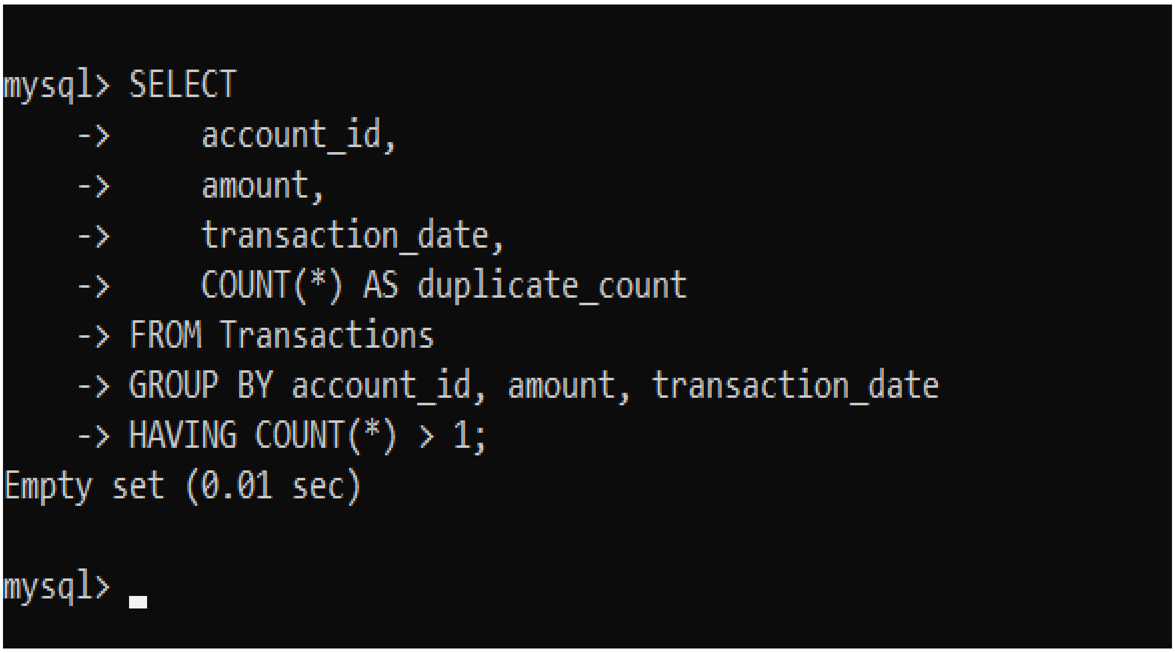
```
mysql> SELECT
    ->     account_id,
    ->     COUNT(transaction_id) AS transaction_count
    -> FROM Transactions
    -> GROUP BY account_id
    -> ORDER BY transaction_count DESC;
+------------+-------------------+
| account_id | transaction_count |
+------------+-------------------+
|       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

mysql> _
```

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



```
mysql> SELECT
    ->     Customers.customer_id,
    ->     CONCAT(Customers.first_name, ' ', Customers.last_name) AS customer_name,
    ->     Accounts.account_type,
    ->     SUM(Accounts.balance) AS total_balance
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id
    -> GROUP BY Customers.customer_id, Accounts.account_type
    -> ORDER BY total_balance DESC;
+-------------+-----------------+--------------+---------------+
| customer_id | customer_name   | account_type | total_balance |
+-------------+-----------------+--------------+---------------+
|           7 | Chetan Rajput   | Current      |      15000.00 |
|           4 | Ananya Gupta    | Current      |      12000.75 |
|          10 | Farhan Malik    | Savings      |      10000.75 |
|           2 | Aditi Sharma    | Current      |      10000.00 |
|           3 | Amit Patel      | Savings      |       7500.50 |
|           1 | SUSHANT Kumar   | Savings      |       5500.00 |
|           8 | Deepika Rathore | Savings      |       3000.50 |
|           6 | Bhavya Mishra   | Savings      |       2000.25 |
|           9 | Esha Verma      | Zero_Balance |          0.00 |
+-------------+-----------------+--------------+---------------+
9 rows in set (0.00 sec)
```

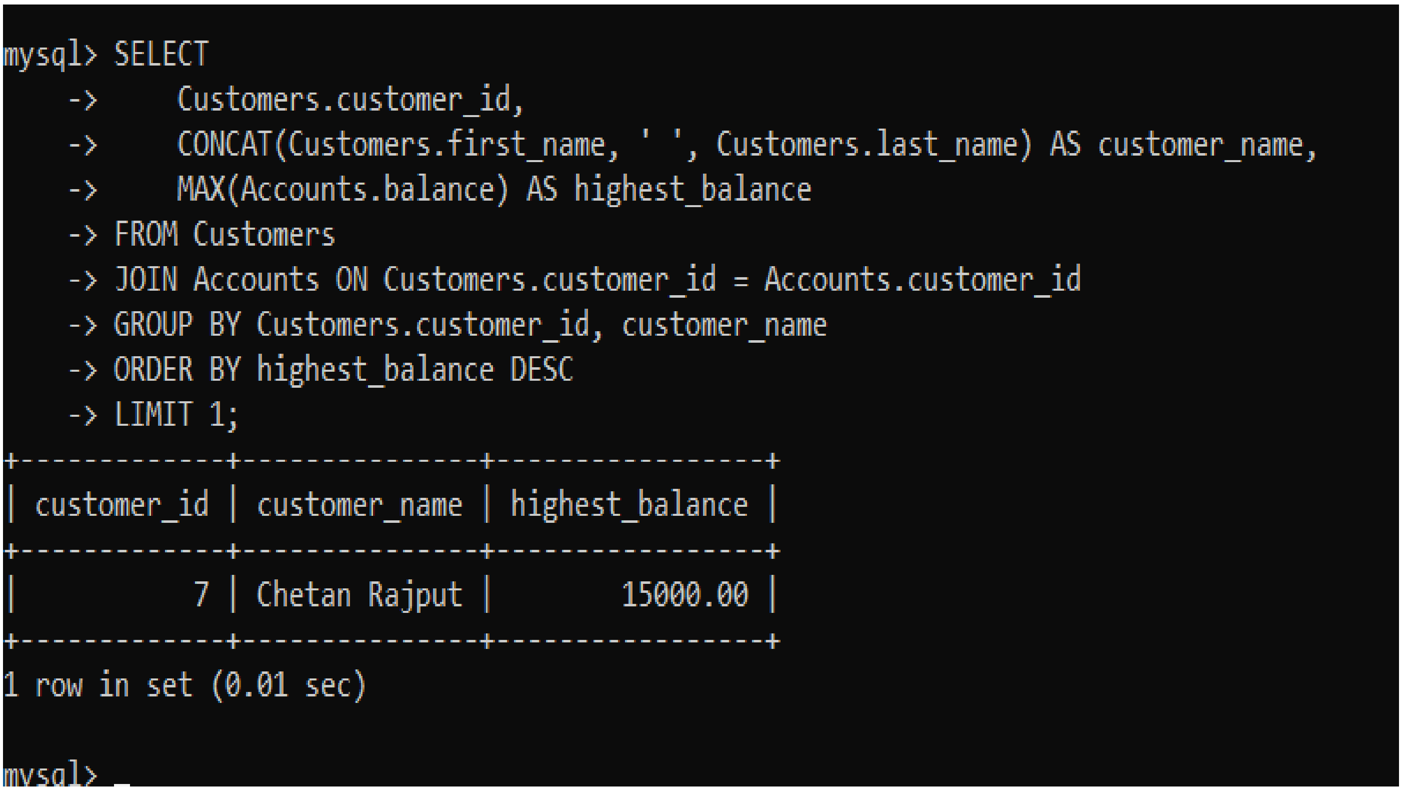
14. Identify and list duplicate transactions based on transaction amount, date, and account.



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

mysql>
```

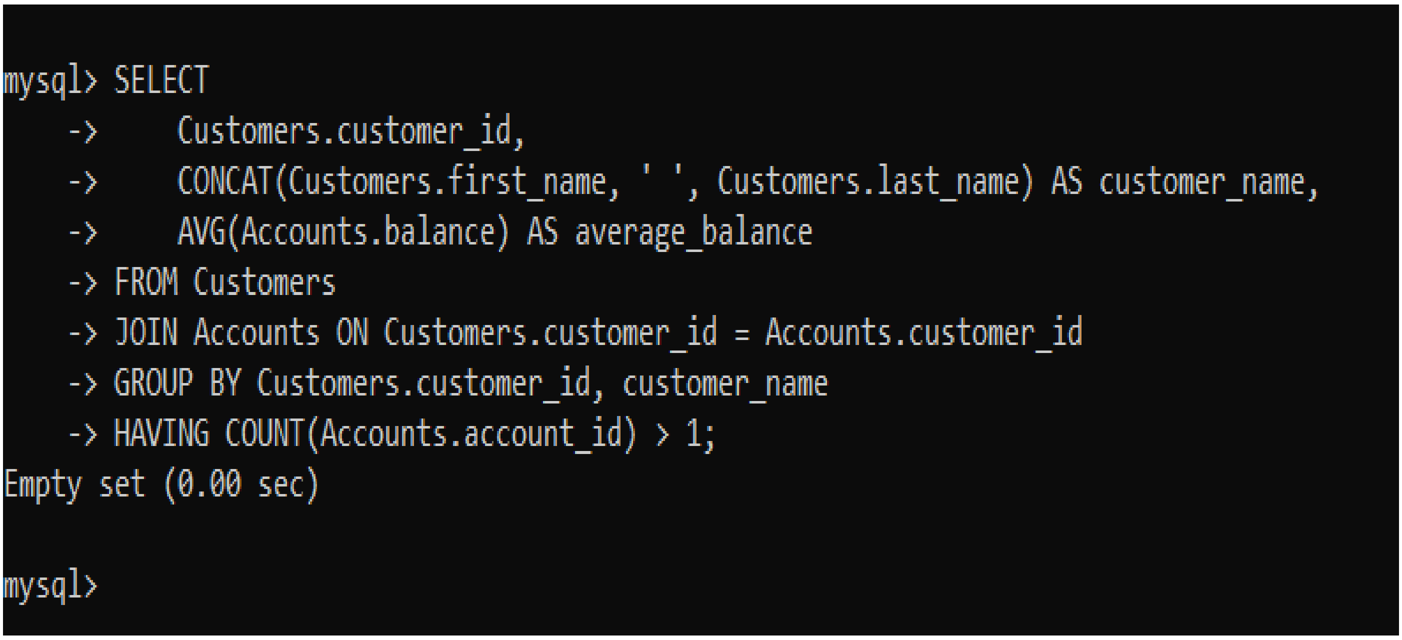
Tasks 4: Subquery and its type:

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



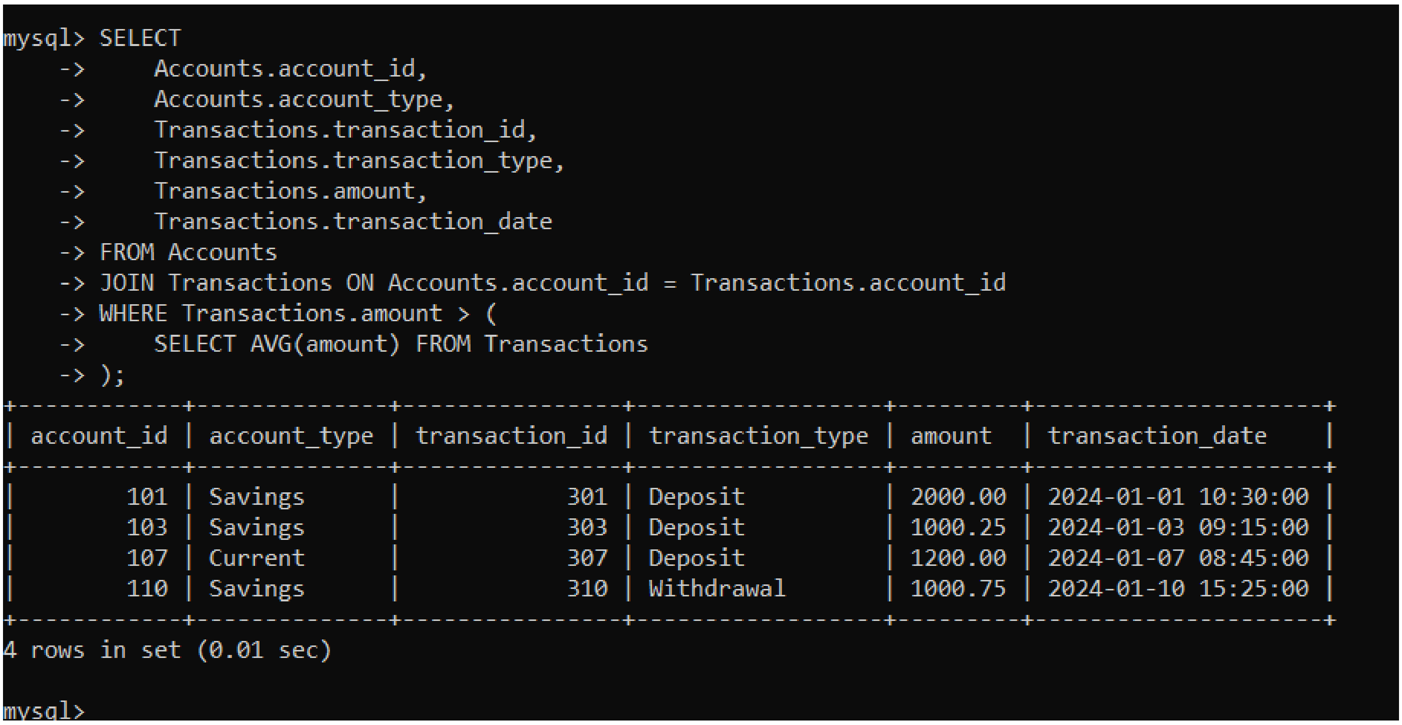
```
mysql> SELECT
    ->     Customers.customer_id,
    ->     CONCAT(Customers.first_name, ' ', Customers.last_name) AS customer_name,
    ->     MAX(Accounts.balance) AS highest_balance
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id
    -> GROUP BY Customers.customer_id, customer_name
    -> ORDER BY highest_balance DESC
    -> LIMIT 1;
+-------------+---------------+-----------------+
| customer_id | customer_name | highest_balance |
+-------------+---------------+-----------------+
|           7 | Chetan Rajput |        15000.00 |
+-------------+---------------+-----------------+
1 row in set (0.01 sec)

mysql> _
```

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



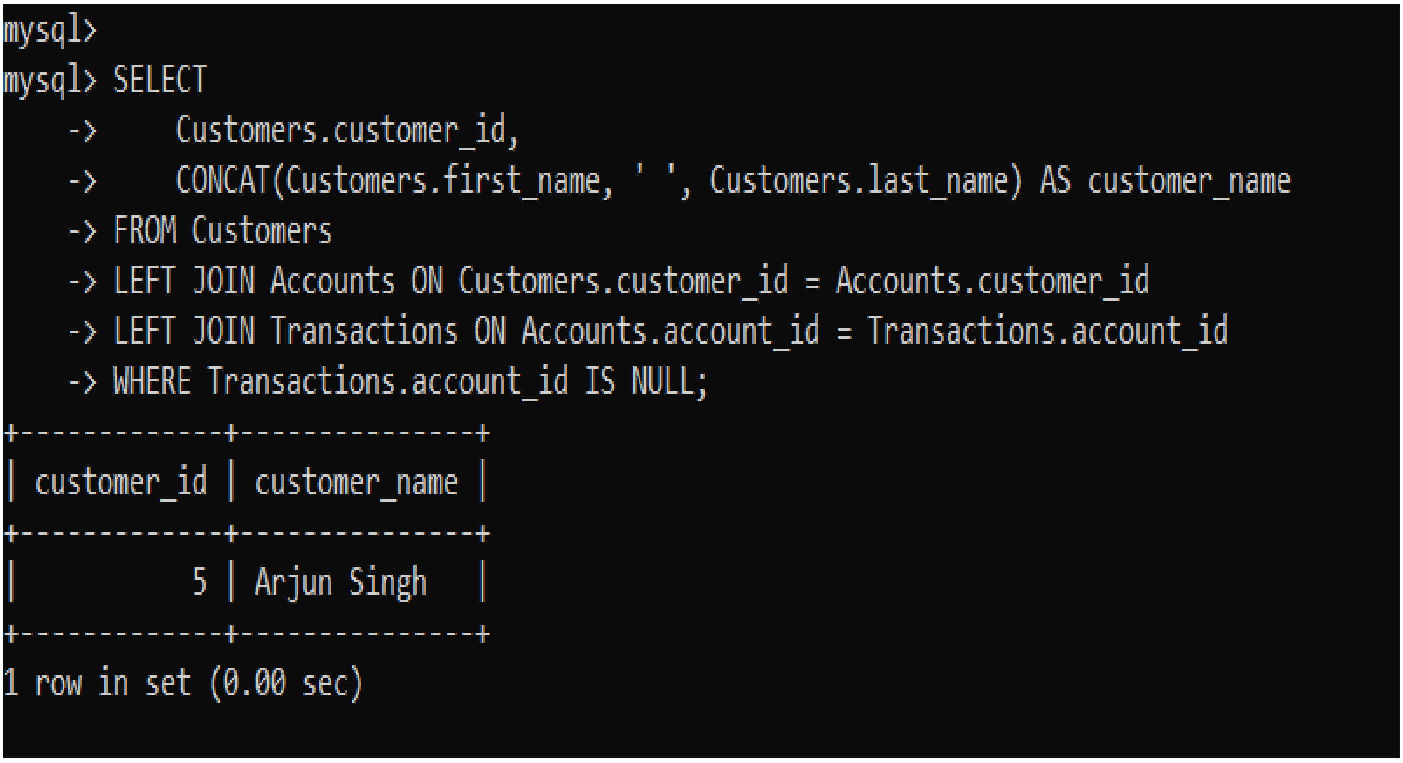
```
mysql> SELECT
    ->     Customers.customer_id,
    ->     CONCAT(Customers.first_name, ' ', Customers.last_name) AS customer_name,
    ->     AVG(Accounts.balance) AS average_balance
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id
    -> GROUP BY Customers.customer_id, customer_name
    -> HAVING COUNT(Accounts.account_id) > 1;
Empty set (0.00 sec)

mysql>
```

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



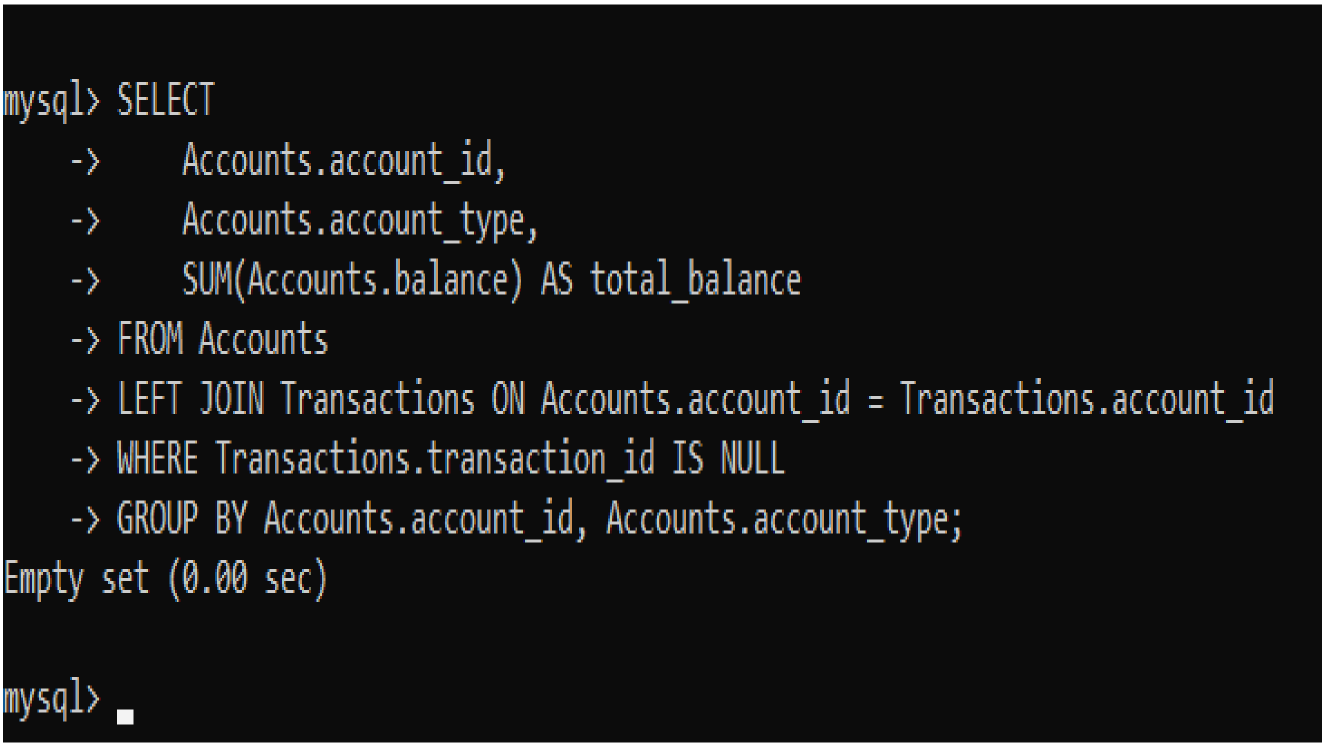
```
mysql> SELECT
    ->     Accounts.account_id,
    ->     Accounts.account_type,
    ->     Transactions.transaction_id,
    ->     Transactions.transaction_type,
    ->     Transactions.amount,
    ->     Transactions.transaction_date
    -> FROM Accounts
    -> JOIN Transactions ON Accounts.account_id = Transactions.account_id
    -> WHERE Transactions.amount > (
    ->     SELECT AVG(amount) FROM Transactions
    -> );
+------------+--------------+----------------+------------------+---------+---------------------+
| account_id | account_type | transaction_id | transaction_type | amount  | transaction_date    |
+------------+--------------+----------------+------------------+---------+---------------------+
|        101 | Savings      |            301 | Deposit          | 2000.00 | 2024-01-01 10:30:00 |
|        103 | Savings      |            303 | Deposit          | 1000.25 | 2024-01-03 09:15:00 |
|        107 | Current      |            307 | Deposit          | 1200.00 | 2024-01-07 08:45:00 |
|        110 | Savings      |            310 | Withdrawal       | 1000.75 | 2024-01-10 15:25:00 |
+------------+--------------+----------------+------------------+---------+---------------------+
4 rows in set (0.01 sec)

mysql>
```

4. Identify customers who have no recorded transactions.



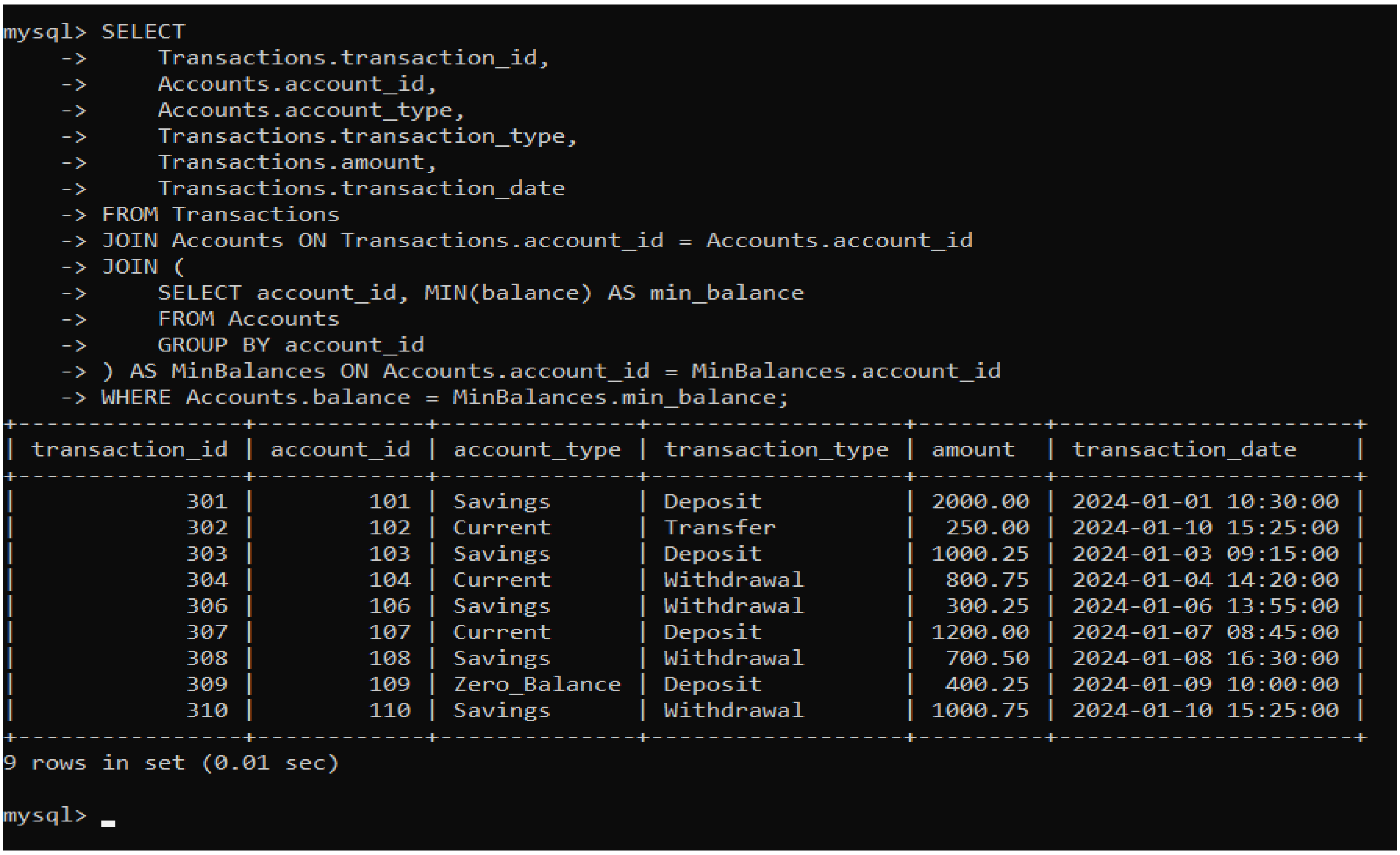
```
mysql>
mysql> SELECT
    ->     Customers.customer_id,
    ->     CONCAT(Customers.first_name, ' ', Customers.last_name) AS customer_name
    -> FROM Customers
    -> LEFT JOIN Accounts ON Customers.customer_id = Accounts.customer_id
    -> LEFT JOIN Transactions ON Accounts.account_id = Transactions.account_id
    -> WHERE Transactions.account_id IS NULL;
+-------------+---------------+
| customer_id | customer_name |
+-------------+---------------+
|           5 | Arjun Singh   |
+-------------+---------------+
1 row in set (0.00 sec)
```

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



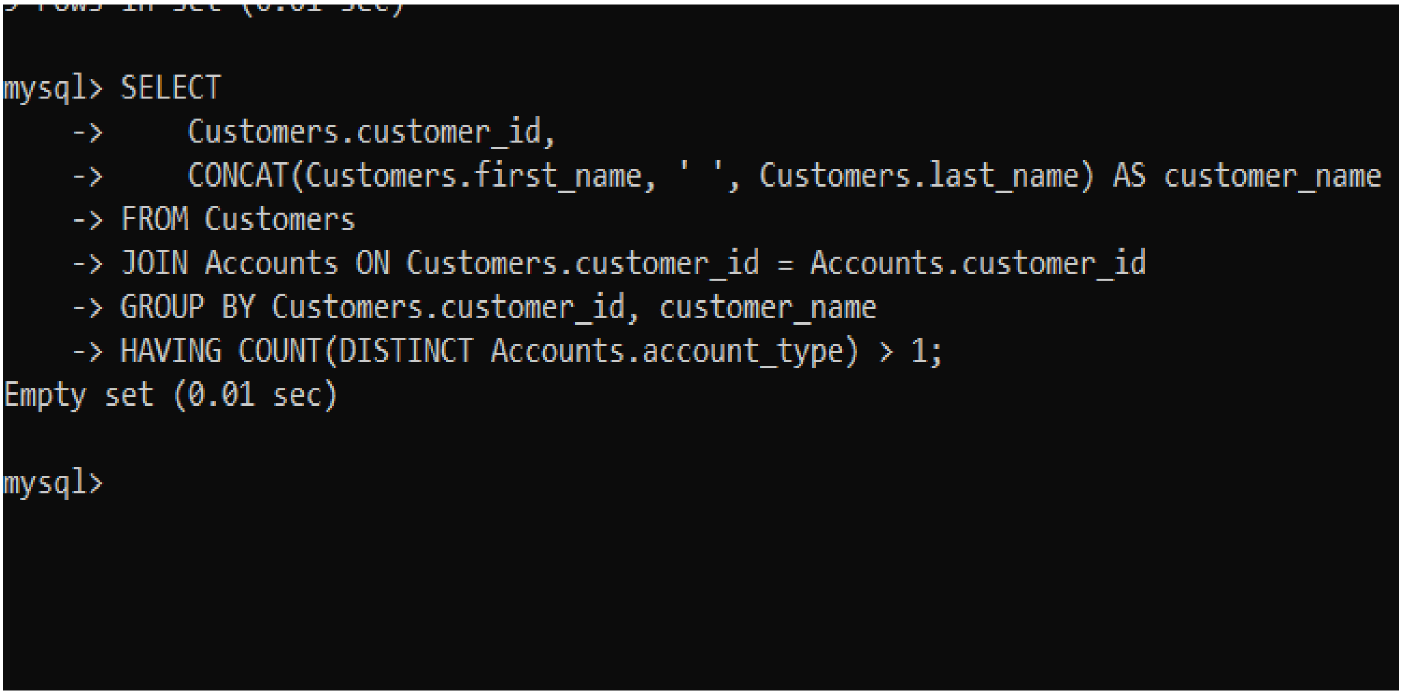
```
mysql> SELECT
    ->     Accounts.account_id,
    ->     Accounts.account_type,
    ->     SUM(Accounts.balance) AS total_balance
    -> FROM Accounts
    -> LEFT JOIN Transactions ON Accounts.account_id = Transactions.account_id
    -> WHERE Transactions.transaction_id IS NULL
    -> GROUP BY Accounts.account_id, Accounts.account_type;
Empty set (0.00 sec)

mysql>
```

6. Retrieve transactions for accounts with the lowest balance.



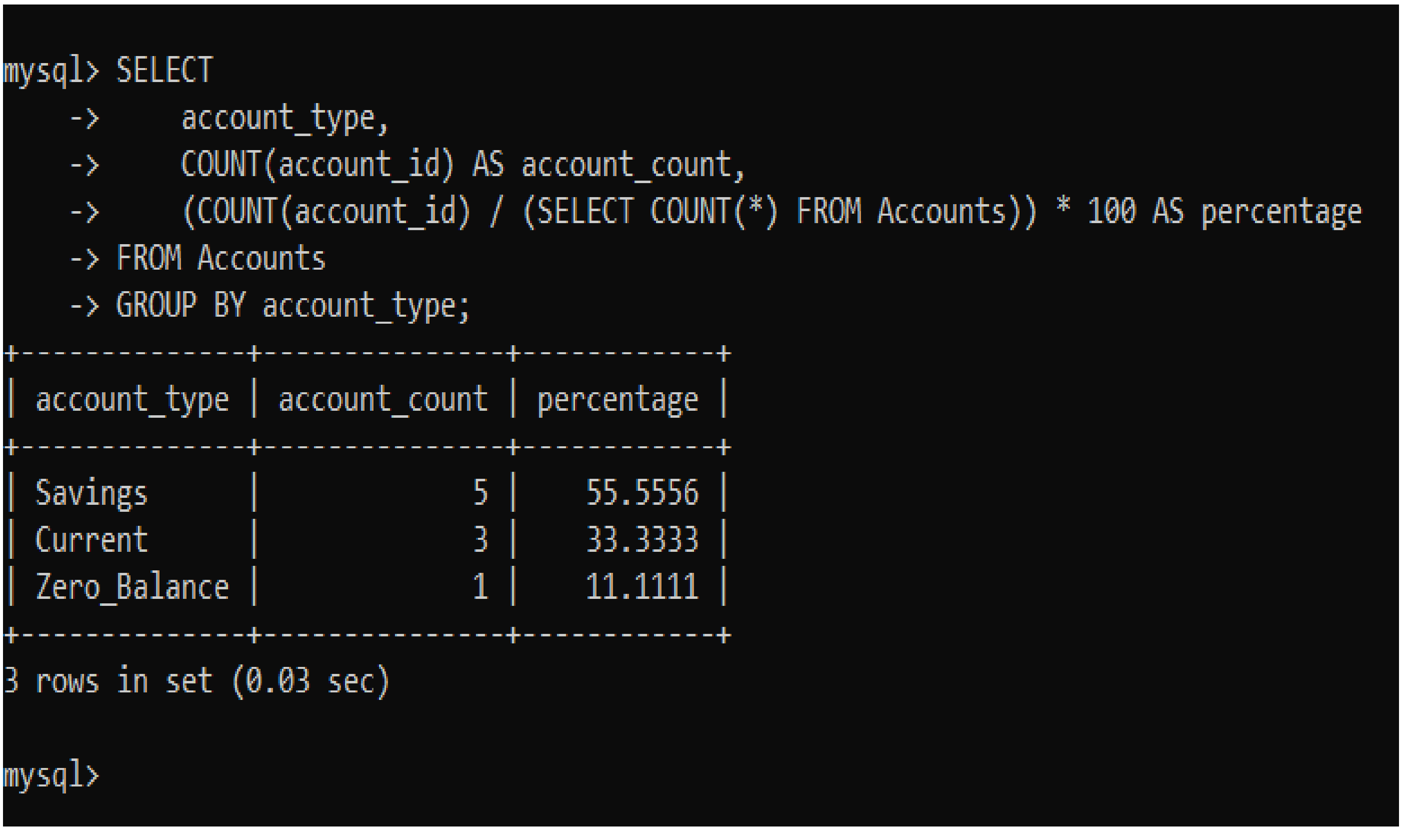
```
mysql> SELECT
    ->     Transactions.transaction_id,
    ->     Accounts.account_id,
    ->     Accounts.account_type,
    ->     Transactions.transaction_type,
    ->     Transactions.amount,
    ->     Transactions.transaction_date
    -> FROM Transactions
    -> JOIN Accounts ON Transactions.account_id = Accounts.account_id
    -> JOIN (
    ->     SELECT account_id, MIN(balance) AS min_balance
    ->     FROM Accounts
    ->     GROUP BY account_id
    -> ) AS MinBalances ON Accounts.account_id = MinBalances.account_id
    -> WHERE Accounts.balance = MinBalances.min_balance;
+----------------+------------+--------------+------------------+---------+---------------------+
| transaction_id | account_id | account_type | transaction_type | amount  | transaction_date    |
+----------------+------------+--------------+------------------+---------+---------------------+
|            301 |        101 | Savings      | Deposit          | 2000.00 | 2024-01-01 10:30:00 |
|            302 |        102 | Current      | Transfer         |  250.00 | 2024-01-10 15:25:00 |
|            303 |        103 | Savings      | Deposit          | 1000.25 | 2024-01-03 09:15:00 |
|            304 |        104 | Current      | Withdrawal       |  800.75 | 2024-01-04 14:20:00 |
|            306 |        106 | Savings      | Withdrawal       |  300.25 | 2024-01-06 13:55:00 |
|            307 |        107 | Current      | Deposit          | 1200.00 | 2024-01-07 08:45:00 |
|            308 |        108 | Savings      | Withdrawal       |  700.50 | 2024-01-08 16:30:00 |
|            309 |        109 | Zero_Balance | Deposit          |  400.25 | 2024-01-09 10:00:00 |
|            310 |        110 | Savings      | Withdrawal       | 1000.75 | 2024-01-10 15:25:00 |
+----------------+------------+--------------+------------------+---------+---------------------+
9 rows in set (0.01 sec)

mysql>
```

7. Identify customers who have accounts of multiple types.



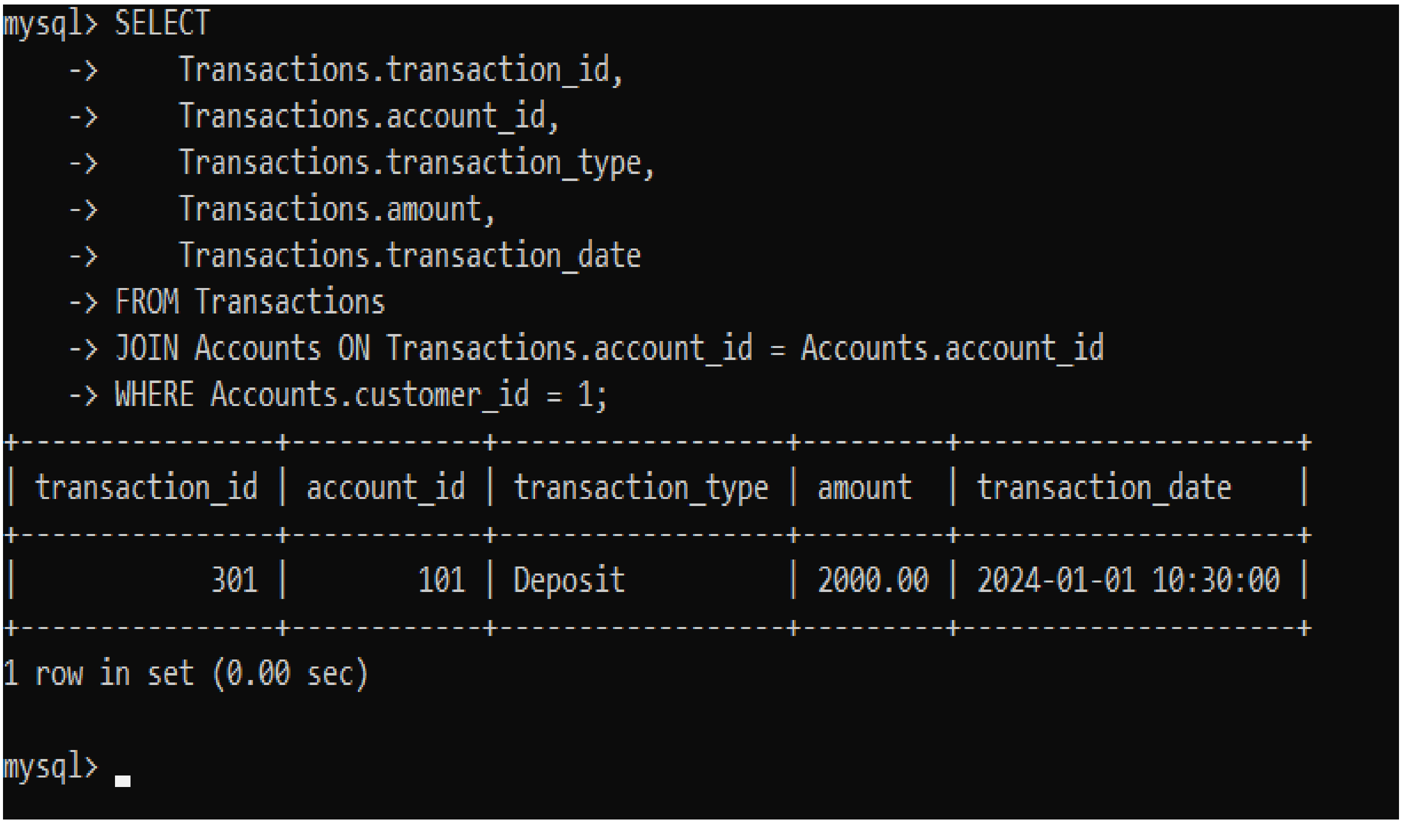
```
mysql> SELECT
    ->     Customers.customer_id,
    ->     CONCAT(Customers.first_name, ' ', Customers.last_name) AS customer_name
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id
    -> GROUP BY Customers.customer_id, customer_name
    -> HAVING COUNT(DISTINCT Accounts.account_type) > 1;
Empty set (0.01 sec)

mysql>
```

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



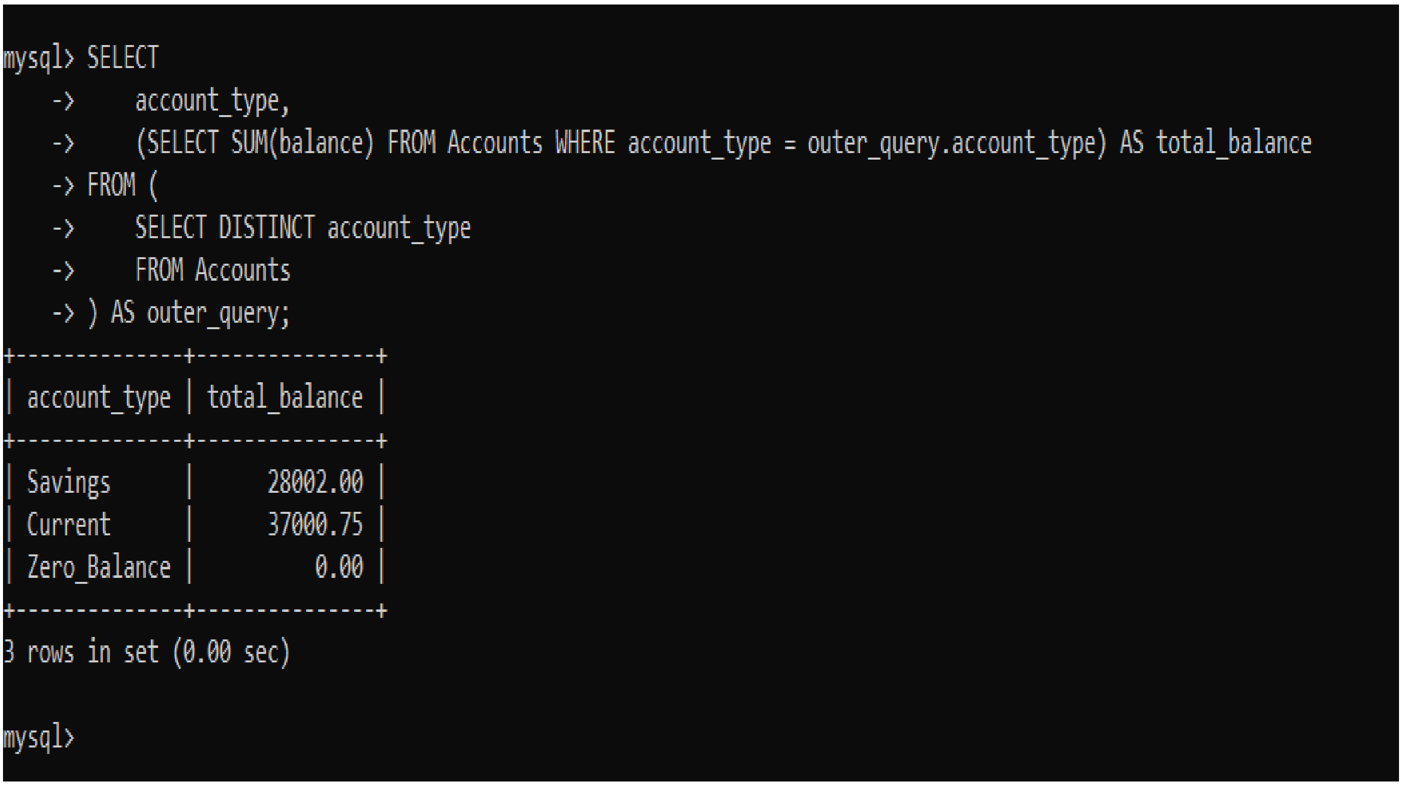
```
mysql> SELECT
    ->     account_type,
    ->     COUNT(account_id) AS account_count,
    ->     (COUNT(account_id) / (SELECT COUNT(*) FROM Accounts)) * 100 AS percentage
    -> FROM Accounts
    -> GROUP BY account_type;
+--------------+---------------+------------+
| account_type | account_count | percentage |
+--------------+---------------+------------+
| Savings      |             5 |    55.5556 |
| Current      |             3 |    33.3333 |
| Zero_Balance |             1 |    11.1111 |
+--------------+---------------+------------+
3 rows in set (0.03 sec)

mysql>
```

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



```
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 = 1;
+----------------+------------+------------------+---------+---------------------+
| transaction_id | account_id | transaction_type | amount  | transaction_date    |
+----------------+------------+------------------+---------+---------------------+
|            301 |        101 | Deposit          | 2000.00 | 2024-01-01 10:30:00 |
+----------------+------------+------------------+---------+---------------------+
1 row in set (0.00 sec)

mysql> 
```

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



```
mysql> SELECT
    ->     account_type,
    ->     (SELECT SUM(balance) FROM Accounts WHERE account_type = outer_query.account_type) AS total_balance
    -> FROM (
    ->     SELECT DISTINCT account_type
    ->     FROM Accounts
    -> ) AS outer_query;
+--------------+---------------+
| account_type | total_balance |
+--------------+---------------+
| Savings      |      28002.00 |
| Current      |      37000.75 |
| Zero_Balance |          0.00 |
+--------------+---------------+
3 rows in set (0.00 sec)

mysql>
```