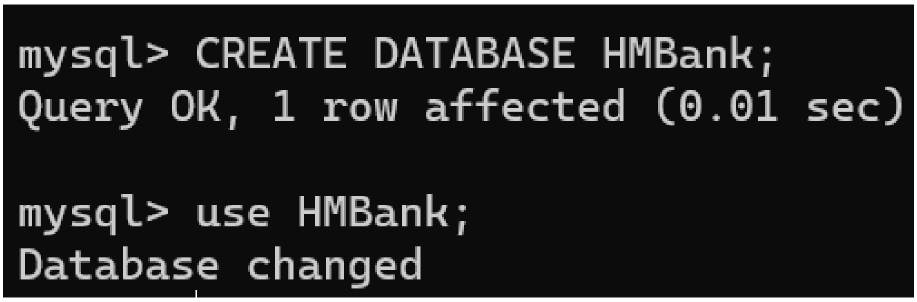
Assignment – 03

Task-1 Database Design:

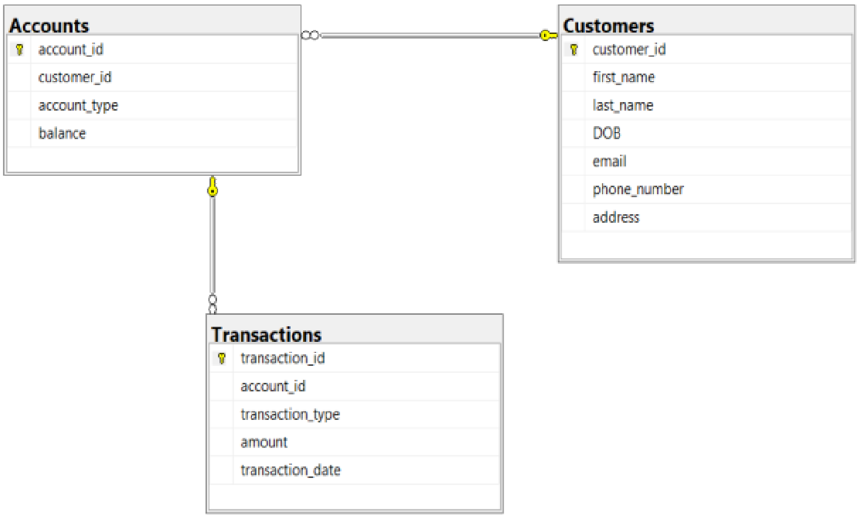
1. Create the database named "HMBank".



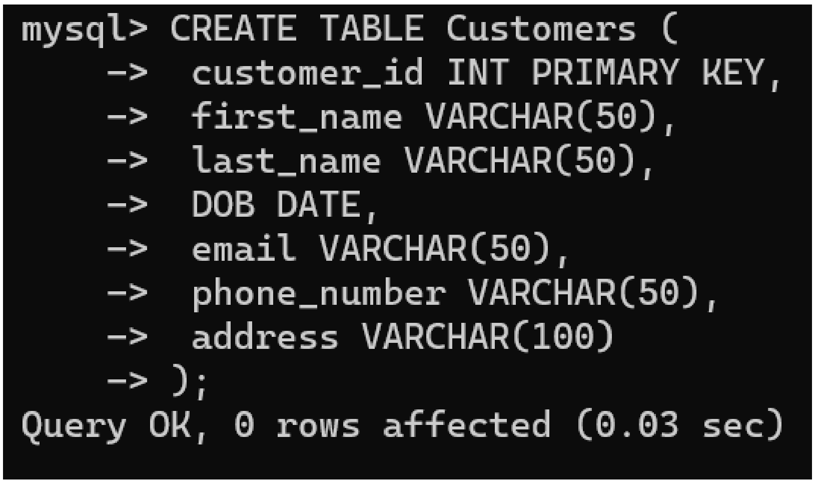
```
mysql> CREATE DATABASE HMBank;
Query OK, 1 row affected (0.01 sec)

mysql> use HMBank;
Database changed
```

2. ERD.

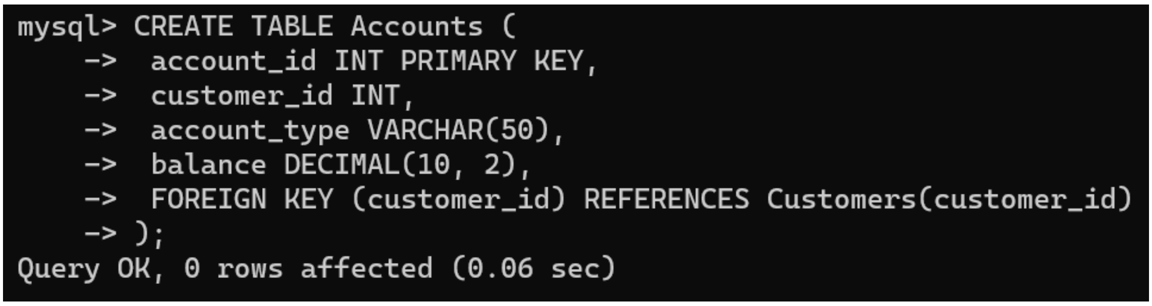


3. Define the schema for the Customers, Accounts, and Transactions tables based on the provided schema. Create appropriate Primary Key and Foreign Key constraints for referential integrity. Write SQL scripts to create the mentioned tables with appropriate data types, constraints, and relationships.
   - Customers:



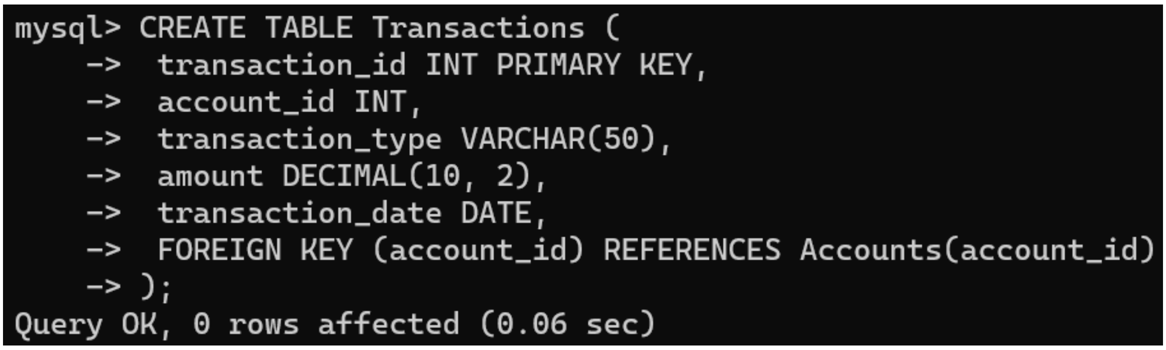
```
mysql> CREATE TABLE Customers (
    ->  customer_id INT PRIMARY KEY,
    ->  first_name VARCHAR(50),
    ->  last_name VARCHAR(50),
    ->  DOB DATE,
    ->  email VARCHAR(50),
    ->  phone_number VARCHAR(50),
    ->  address VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.03 sec)
```

- Accounts:



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

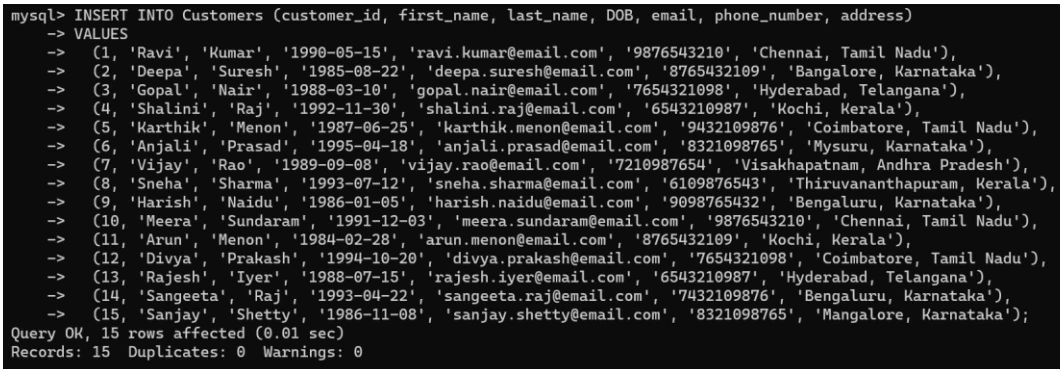
- Transactions:



```
mysql> CREATE TABLE Transactions (
    ->  transaction_id INT PRIMARY KEY,
    ->  account_id INT,
    ->  transaction_type VARCHAR(50),
    ->  amount DECIMAL(10, 2),
    ->  transaction_date DATE,
    ->  FOREIGN KEY (account_id) REFERENCES Accounts(account_id)
    -> );
Query OK, 0 rows affected (0.06 sec)
```

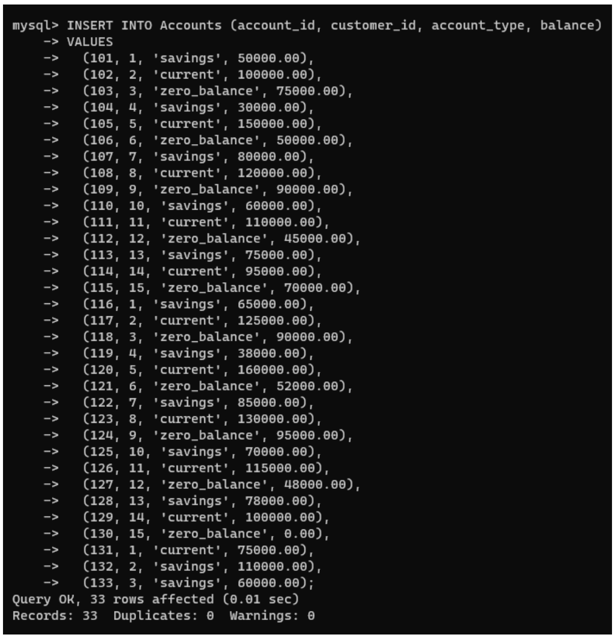
Task-2 Select, Where, Between, AND, Like:

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



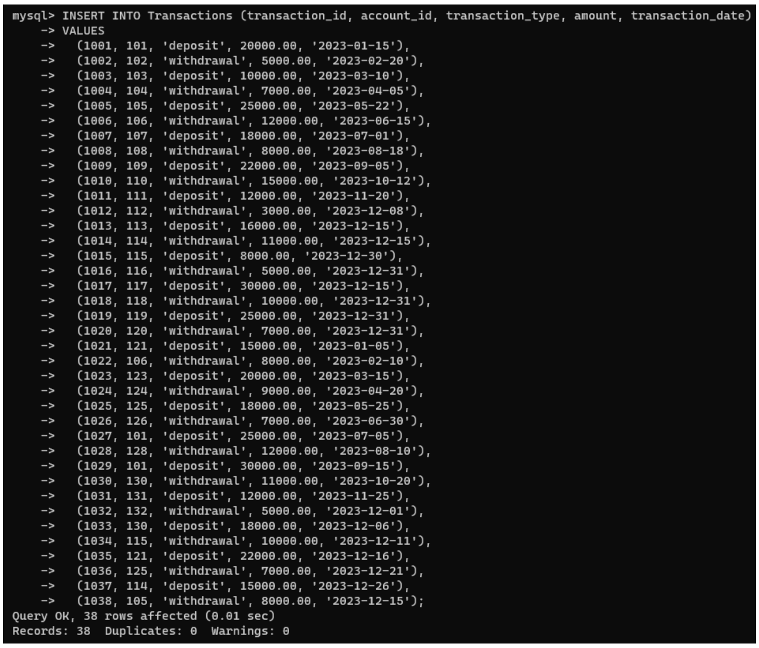
```
mysql> INSERT INTO Customers (customer_id, first_name, last_name, DOB, email, phone_number, address)
    -> VALUES
    ->   (1, 'Ravi', 'Kumar', '1990-05-15', 'ravi.kumar@email.com', '9876543210', 'Chennai, Tamil Nadu'),
    ->   (2, 'Deepa', 'Suresh', '1985-08-22', 'deepa.suresh@email.com', '8765432109', 'Bangalore, Karnataka'),
    ->   (3, 'Gopal', 'Nair', '1988-03-10', 'gopal.nair@email.com', '7654321098', 'Hyderabad, Telangana'),
    ->   (4, 'Shalini', 'Raj', '1992-11-30', 'shalini.raj@email.com', '6543210987', 'Kochi, Kerala'),
    ->   (5, 'Karthik', 'Menon', '1987-06-25', 'karthik.menon@email.com', '9432109876', 'Coimbatore, Tamil Nadu'),
    ->   (6, 'Anjali', 'Prasad', '1995-04-18', 'anjali.prasad@email.com', '8321098765', 'Mysuru, Karnataka'),
    ->   (7, 'Vijay', 'Rao', '1989-09-08', 'vijay.rao@email.com', '7210987654', 'Visakhapatnam, Andhra Pradesh'),
    ->   (8, 'Sneha', 'Sharma', '1993-07-12', 'sneha.sharma@email.com', '6109876543', 'Thiruvananthapuram, Kerala'),
    ->   (9, 'Harish', 'Naidu', '1986-01-05', 'harish.naidu@email.com', '9098765432', 'Bengaluru, Karnataka'),
    ->   (10, 'Meera', 'Sundaram', '1991-12-03', 'meera.sundaram@email.com', '9876543210', 'Chennai, Tamil Nadu'),
    ->   (11, 'Arun', 'Menon', '1984-02-28', 'arun.menon@email.com', '8765432109', 'Kochi, Kerala'),
    ->   (12, 'Divya', 'Prakash', '1994-10-20', 'divya.prakash@email.com', '7654321098', 'Coimbatore, Tamil Nadu'),
    ->   (13, 'Rajesh', 'Iyer', '1988-07-15', 'rajesh.iyer@email.com', '6543210987', 'Hyderabad, Telangana'),
    ->   (14, 'Sangeeta', 'Raj', '1993-04-22', 'sangeeta.raj@email.com', '7432109876', 'Bengaluru, Karnataka'),
    ->   (15, 'Sanjay', 'Shetty', '1986-11-08', 'sanjay.shetty@email.com', '8321098765', 'Mangalore, Karnataka');
Query OK, 15 rows affected (0.01 sec)
Records: 15  Duplicates: 0  Warnings: 0
```

    - Accounts:



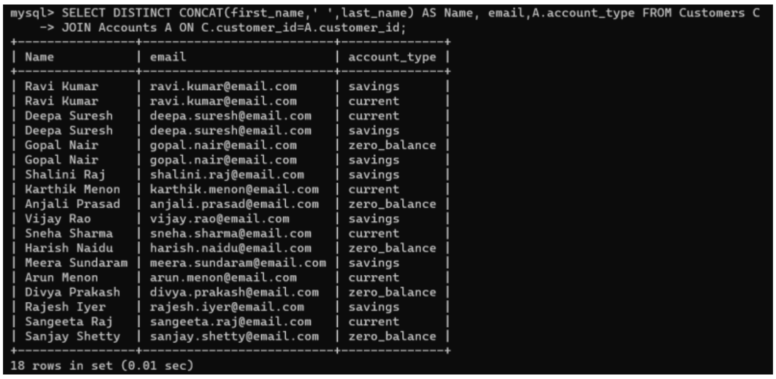
```
mysql> INSERT INTO Accounts (account_id, customer_id, account_type, balance)
    -> VALUES
    ->   (101, 1, 'savings', 50000.00),
    ->   (102, 2, 'current', 100000.00),
    ->   (103, 3, 'zero_balance', 75000.00),
    ->   (104, 4, 'savings', 30000.00),
    ->   (105, 5, 'current', 150000.00),
    ->   (106, 6, 'zero_balance', 50000.00),
    ->   (107, 7, 'savings', 80000.00),
    ->   (108, 8, 'current', 120000.00),
    ->   (109, 9, 'zero_balance', 90000.00),
    ->   (110, 10, 'savings', 60000.00),
    ->   (111, 11, 'current', 110000.00),
    ->   (112, 12, 'zero_balance', 45000.00),
    ->   (113, 13, 'savings', 75000.00),
    ->   (114, 14, 'current', 95000.00),
    ->   (115, 15, 'zero_balance', 70000.00),
    ->   (116, 1, 'savings', 65000.00),
    ->   (117, 2, 'current', 125000.00),
    ->   (118, 3, 'zero_balance', 90000.00),
    ->   (119, 4, 'savings', 38000.00),
    ->   (120, 5, 'current', 160000.00),
    ->   (121, 6, 'zero_balance', 52000.00),
    ->   (122, 7, 'savings', 85000.00),
    ->   (123, 8, 'current', 130000.00),
    ->   (124, 9, 'zero_balance', 95000.00),
    ->   (125, 10, 'savings', 70000.00),
    ->   (126, 11, 'current', 115000.00),
    ->   (127, 12, 'zero_balance', 48000.00),
    ->   (128, 13, 'savings', 78000.00),
    ->   (129, 14, 'current', 100000.00),
    ->   (130, 15, 'zero_balance', 0.00),
    ->   (131, 1, 'current', 75000.00),
    ->   (132, 2, 'savings', 110000.00),
    ->   (133, 3, 'savings', 60000.00);
Query OK, 33 rows affected (0.01 sec)
Records: 33  Duplicates: 0  Warnings: 0
```

    - Transactions:



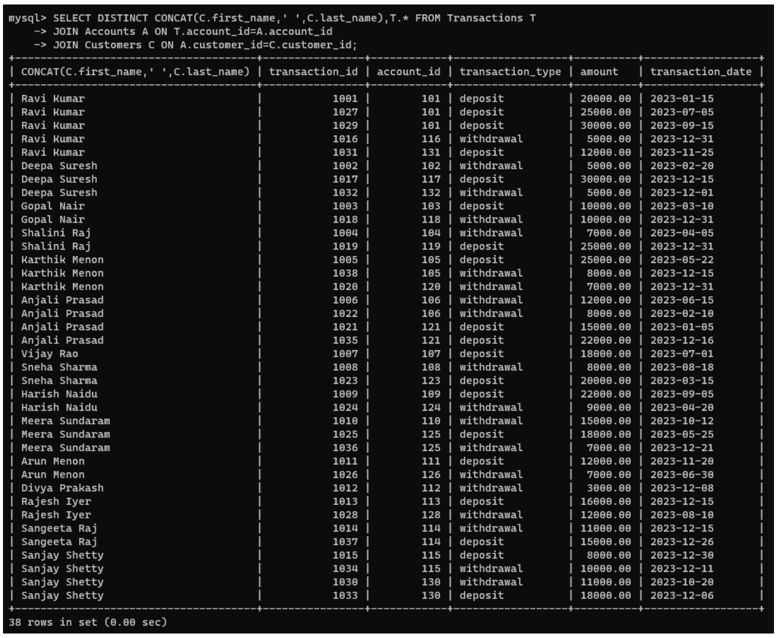
```
mysql> INSERT INTO Transactions (transaction_id, account_id, transaction_type, amount, transaction_date)
    -> VALUES
    ->   (1001, 101, 'deposit', 20000.00, '2023-01-15'),
    ->   (1002, 102, 'withdrawal', 5000.00, '2023-02-20'),
    ->   (1003, 103, 'deposit', 10000.00, '2023-03-10'),
    ->   (1004, 104, 'withdrawal', 7000.00, '2023-04-05'),
    ->   (1005, 105, 'deposit', 25000.00, '2023-05-22'),
    ->   (1006, 106, 'withdrawal', 12000.00, '2023-06-15'),
    ->   (1007, 107, 'deposit', 18000.00, '2023-07-01'),
    ->   (1008, 108, 'withdrawal', 8000.00, '2023-08-18'),
    ->   (1009, 109, 'deposit', 22000.00, '2023-09-05'),
    ->   (1010, 110, 'withdrawal', 15000.00, '2023-10-12'),
    ->   (1011, 111, 'deposit', 12000.00, '2023-11-20'),
    ->   (1012, 112, 'withdrawal', 3000.00, '2023-12-08'),
    ->   (1013, 113, 'deposit', 16000.00, '2023-12-15'),
    ->   (1014, 114, 'withdrawal', 11000.00, '2023-12-15'),
    ->   (1015, 115, 'deposit', 8000.00, '2023-12-30'),
    ->   (1016, 116, 'withdrawal', 5000.00, '2023-12-31'),
    ->   (1017, 117, 'deposit', 30000.00, '2023-12-15'),
    ->   (1018, 118, 'withdrawal', 10000.00, '2023-12-31'),
    ->   (1019, 119, 'deposit', 25000.00, '2023-12-31'),
    ->   (1020, 120, 'withdrawal', 7000.00, '2023-12-31'),
    ->   (1021, 121, 'deposit', 15000.00, '2023-01-05'),
    ->   (1022, 106, 'withdrawal', 8000.00, '2023-02-10'),
    ->   (1023, 123, 'deposit', 20000.00, '2023-03-15'),
    ->   (1024, 124, 'withdrawal', 9000.00, '2023-04-20'),
    ->   (1025, 125, 'deposit', 18000.00, '2023-05-25'),
    ->   (1026, 126, 'withdrawal', 7000.00, '2023-06-30'),
    ->   (1027, 101, 'deposit', 25000.00, '2023-07-05'),
    ->   (1028, 128, 'withdrawal', 12000.00, '2023-08-10'),
    ->   (1029, 101, 'deposit', 30000.00, '2023-09-15'),
    ->   (1030, 130, 'withdrawal', 11000.00, '2023-10-20'),
    ->   (1031, 131, 'deposit', 12000.00, '2023-11-25'),
    ->   (1032, 132, 'withdrawal', 5000.00, '2023-12-01'),
    ->   (1033, 130, 'deposit', 18000.00, '2023-12-06'),
    ->   (1034, 115, 'withdrawal', 10000.00, '2023-12-11'),
    ->   (1035, 121, 'deposit', 22000.00, '2023-12-16'),
    ->   (1036, 125, 'withdrawal', 7000.00, '2023-12-21'),
    ->   (1037, 114, 'deposit', 15000.00, '2023-12-26'),
    ->   (1038, 105, 'withdrawal', 8000.00, '2023-12-15');
Query OK, 38 rows affected (0.01 sec)
Records: 38  Duplicates: 0  Warnings: 0
```

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



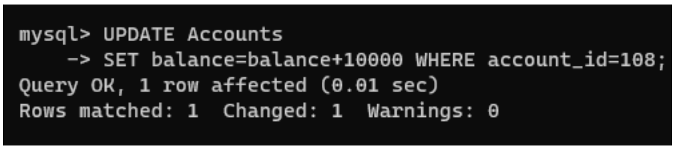
```
mysql> SELECT DISTINCT CONCAT(first_name,' ',last_name) AS Name, email,A.account_type FROM Customers C
    -> JOIN Accounts A ON C.customer_id=A.customer_id;
+----------------+--------------------------+--------------+
| Name           | email                    | account_type |
+----------------+--------------------------+--------------+
| Ravi Kumar     | ravi.kumar@email.com     | savings      |
| Ravi Kumar     | ravi.kumar@email.com     | current      |
| Deepa Suresh   | deepa.suresh@email.com   | current      |
| Deepa Suresh   | deepa.suresh@email.com   | savings      |
| Gopal Nair     | gopal.nair@email.com     | zero_balance |
| Gopal Nair     | gopal.nair@email.com     | savings      |
| Shalini Raj    | shalini.raj@email.com    | savings      |
| Karthik Menon  | karthik.menon@email.com  | current      |
| Anjali Prasad  | anjali.prasad@email.com  | zero_balance |
| Vijay Rao      | vijay.rao@email.com      | savings      |
| Sneha Sharma   | sneha.sharma@email.com   | current      |
| Harish Naidu   | harish.naidu@email.com   | zero_balance |
| Meera Sundaram | meera.sundaram@email.com | savings      |
| Arun Menon     | arun.menon@email.com     | current      |
| Divya Prakash  | divya.prakash@email.com  | zero_balance |
| Rajesh Iyer    | rajesh.iyer@email.com    | savings      |
| Sangeeta Raj   | sangeeta.raj@email.com   | current      |
| Sanjay Shetty  | sanjay.shetty@email.com  | zero_balance |
+----------------+--------------------------+--------------+
18 rows in set (0.01 sec)
```

   b) Write a SQL query to list all transaction corresponding customer.



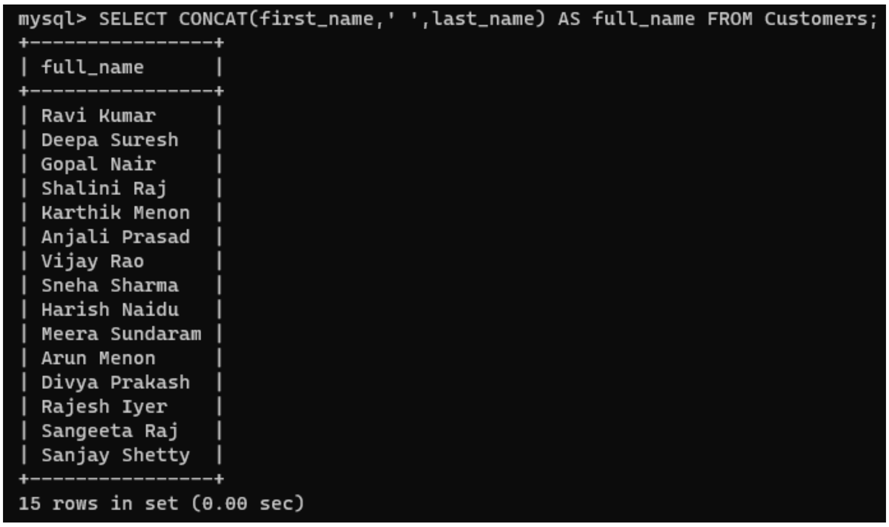
```
mysql> SELECT DISTINCT CONCAT(C.first_name,' ',C.last_name),T.* FROM Transactions T
    -> JOIN Accounts A ON T.account_id=A.account_id
    -> JOIN Customers C ON A.customer_id=C.customer_id;
+--------------------------------------+----------------+------------+------------------+----------+------------------+
| CONCAT(C.first_name,' ',C.last_name) | transaction_id | account_id | transaction_type | amount   | transaction_date |
+--------------------------------------+----------------+------------+------------------+----------+------------------+
| Ravi Kumar                           |           1001 |        101 | deposit          | 20000.00 | 2023-01-15       |
| Ravi Kumar                           |           1027 |        101 | deposit          | 25000.00 | 2023-07-05       |
| Ravi Kumar                           |           1029 |        101 | deposit          | 30000.00 | 2023-09-15       |
| Ravi Kumar                           |           1016 |        116 | withdrawal       |  5000.00 | 2023-12-31       |
| Ravi Kumar                           |           1031 |        131 | deposit          | 12000.00 | 2023-11-25       |
| Deepa Suresh                         |           1002 |        102 | withdrawal       |  5000.00 | 2023-02-20       |
| Deepa Suresh                         |           1017 |        117 | deposit          | 30000.00 | 2023-12-15       |
| Deepa Suresh                         |           1032 |        132 | withdrawal       |  5000.00 | 2023-12-01       |
| Gopal Nair                           |           1003 |        103 | deposit          | 10000.00 | 2023-03-10       |
| Gopal Nair                           |           1018 |        118 | withdrawal       | 10000.00 | 2023-12-31       |
| Shalini Raj                          |           1004 |        104 | withdrawal       |  7000.00 | 2023-04-05       |
| Shalini Raj                          |           1019 |        119 | deposit          | 25000.00 | 2023-12-31       |
| Karthik Menon                        |           1005 |        105 | deposit          | 25000.00 | 2023-05-22       |
| Karthik Menon                        |           1038 |        105 | withdrawal       |  8000.00 | 2023-12-15       |
| Karthik Menon                        |           1020 |        120 | withdrawal       |  7000.00 | 2023-12-31       |
| Anjali Prasad                        |           1006 |        106 | withdrawal       | 12000.00 | 2023-06-15       |
| Anjali Prasad                        |           1022 |        106 | withdrawal       |  8000.00 | 2023-02-10       |
| Anjali Prasad                        |           1021 |        121 | deposit          | 15000.00 | 2023-01-05       |
| Anjali Prasad                        |           1035 |        121 | deposit          | 22000.00 | 2023-12-16       |
| Vijay Rao                            |           1007 |        107 | deposit          | 18000.00 | 2023-07-01       |
| Sneha Sharma                         |           1008 |        108 | withdrawal       |  8000.00 | 2023-08-18       |
| Sneha Sharma                         |           1023 |        123 | deposit          | 20000.00 | 2023-03-15       |
| Harish Naidu                         |           1009 |        109 | deposit          | 22000.00 | 2023-09-05       |
| Harish Naidu                         |           1024 |        124 | withdrawal       |  9000.00 | 2023-04-20       |
| Meera Sundaram                       |           1010 |        110 | withdrawal       | 15000.00 | 2023-10-12       |
| Meera Sundaram                       |           1025 |        125 | deposit          | 18000.00 | 2023-05-25       |
| Meera Sundaram                       |           1036 |        125 | withdrawal       |  7000.00 | 2023-12-21       |
| Arun Menon                           |           1011 |        111 | deposit          | 12000.00 | 2023-11-20       |
| Arun Menon                           |           1026 |        126 | withdrawal       |  7000.00 | 2023-06-30       |
| Divya Prakash                        |           1012 |        112 | withdrawal       |  3000.00 | 2023-12-08       |
| Rajesh Iyer                          |           1013 |        113 | deposit          | 16000.00 | 2023-12-15       |
| Rajesh Iyer                          |           1028 |        128 | withdrawal       | 12000.00 | 2023-08-10       |
| Sangeeta Raj                         |           1014 |        114 | withdrawal       | 11000.00 | 2023-12-15       |
| Sangeeta Raj                         |           1037 |        114 | deposit          | 15000.00 | 2023-12-26       |
| Sanjay Shetty                        |           1015 |        115 | deposit          |  8000.00 | 2023-12-30       |
| Sanjay Shetty                        |           1034 |        115 | withdrawal       | 10000.00 | 2023-12-11       |
| Sanjay Shetty                        |           1030 |        130 | withdrawal       | 11000.00 | 2023-10-20       |
| Sanjay Shetty                        |           1033 |        130 | deposit          | 18000.00 | 2023-12-06       |
+--------------------------------------+----------------+------------+------------------+----------+------------------+
38 rows in set (0.00 sec)
```

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



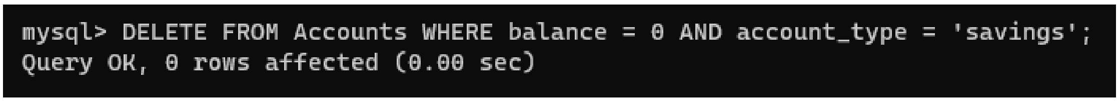
```
mysql> UPDATE Accounts
    -> SET balance=balance+10000 WHERE account_id=108;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

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



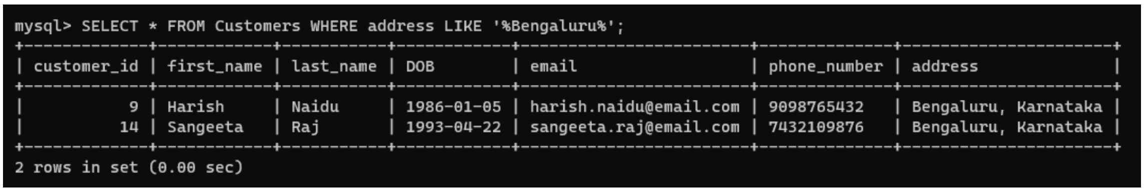
```
mysql> SELECT CONCAT(first_name,' ',last_name) AS full_name FROM Customers;
+----------------+
| full_name      |
+----------------+
| Ravi Kumar     |
| Deepa Suresh   |
| Gopal Nair     |
| Shalini Raj    |
| Karthik Menon  |
| Anjali Prasad  |
| Vijay Rao      |
| Sneha Sharma   |
| Harish Naidu   |
| Meera Sundaram |
| Arun Menon     |
| Divya Prakash  |
| Rajesh Iyer    |
| Sangeeta Raj   |
| Sanjay Shetty  |
+----------------+
15 rows in set (0.00 sec)
```

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



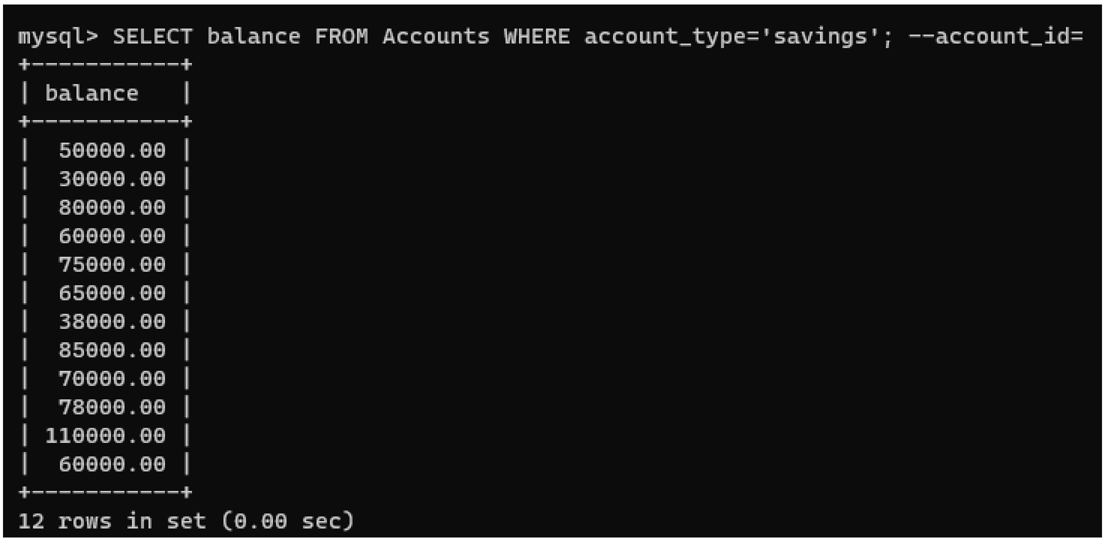
```
mysql> DELETE FROM Accounts WHERE balance = 0 AND account_type = 'savings';
Query OK, 0 rows affected (0.00 sec)
```

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



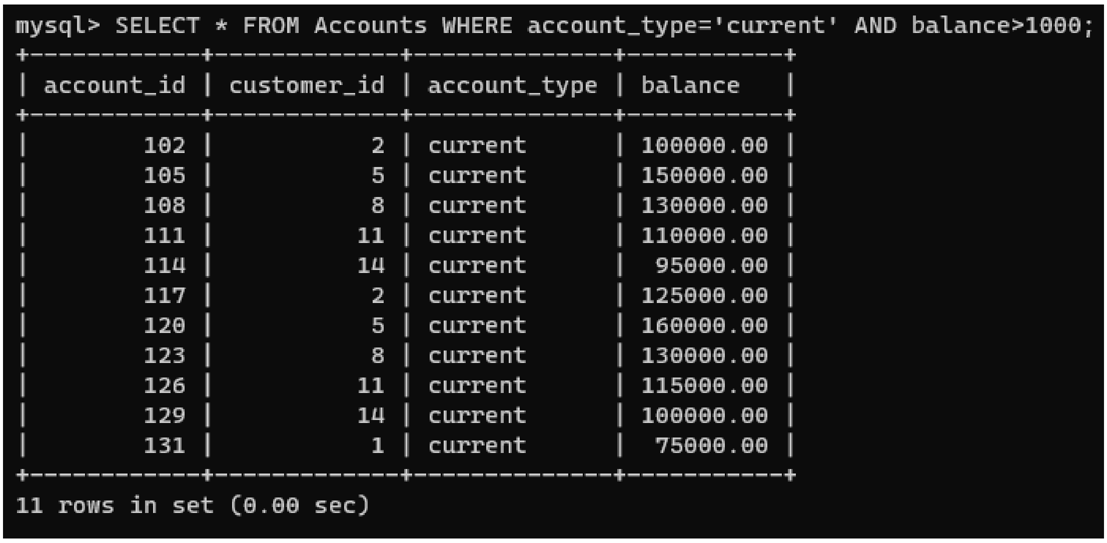
```
mysql> SELECT * FROM Customers WHERE address LIKE '%Bengaluru%';
+-------------+------------+-----------+------------+------------------------+--------------+----------------------+
| customer_id | first_name | last_name | DOB        | email                  | phone_number | address              |
+-------------+------------+-----------+------------+------------------------+--------------+----------------------+
|           9 | Harish     | Naidu     | 1986-01-05 | harish.naidu@email.com | 9098765432   | Bengaluru, Karnataka |
|          14 | Sangeeta   | Raj       | 1993-04-22 | sangeeta.raj@email.com | 7432109876   | Bengaluru, Karnataka |
+-------------+------------+-----------+------------+------------------------+--------------+----------------------+
2 rows in set (0.00 sec)
```

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



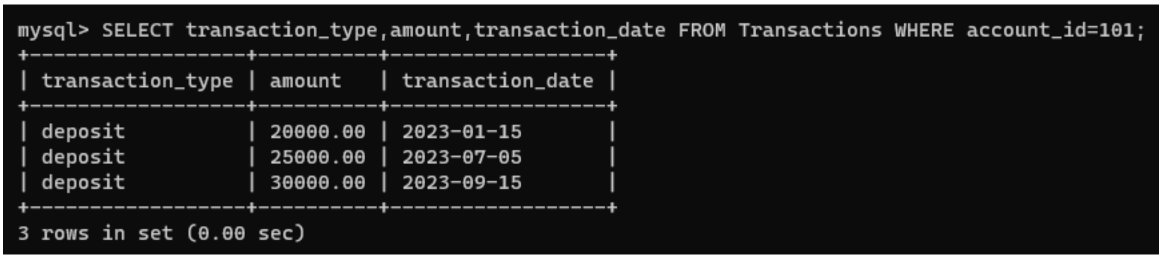
```
mysql> SELECT balance FROM Accounts WHERE account_type='savings'; --account_id=
+-----------+
| balance   |
+-----------+
|  50000.00 |
|  30000.00 |
|  80000.00 |
|  60000.00 |
|  75000.00 |
|  65000.00 |
|  38000.00 |
|  85000.00 |
|  70000.00 |
|  78000.00 |
| 110000.00 |
|  60000.00 |
+-----------+
12 rows in set (0.00 sec)
```

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



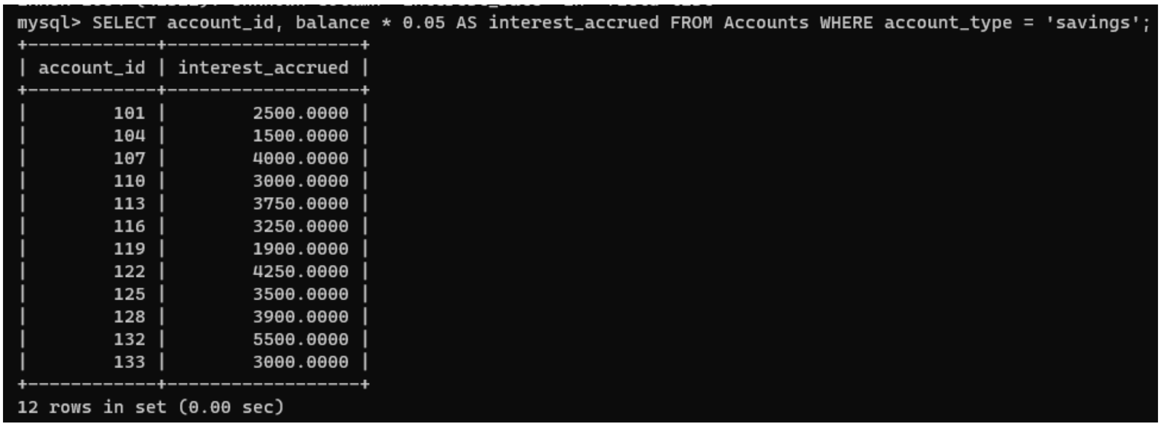
```
mysql> SELECT * FROM Accounts WHERE account_type='current' AND balance>1000;
+------------+-------------+--------------+-----------+
| account_id | customer_id | account_type | balance   |
+------------+-------------+--------------+-----------+
|        102 |           2 | current      | 100000.00 |
|        105 |           5 | current      | 150000.00 |
|        108 |           8 | current      | 130000.00 |
|        111 |          11 | current      | 110000.00 |
|        114 |          14 | current      |  95000.00 |
|        117 |           2 | current      | 125000.00 |
|        120 |           5 | current      | 160000.00 |
|        123 |           8 | current      | 130000.00 |
|        126 |          11 | current      | 115000.00 |
|        129 |          14 | current      | 100000.00 |
|        131 |           1 | current      |  75000.00 |
+------------+-------------+--------------+-----------+
11 rows in set (0.00 sec)
```

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



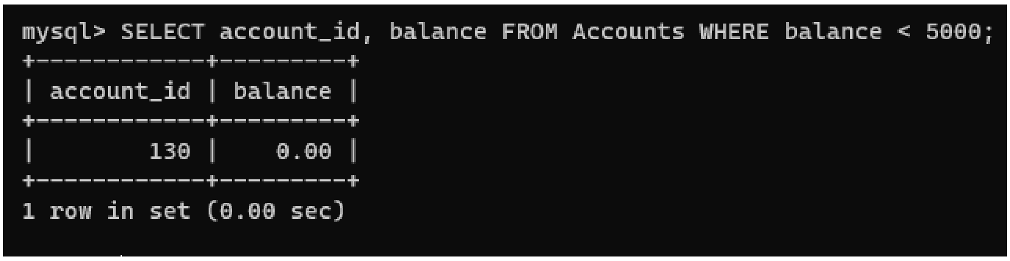
```
mysql> SELECT transaction_type,amount,transaction_date FROM Transactions WHERE account_id=101;
+------------------+----------+------------------+
| transaction_type | amount   | transaction_date |
+------------------+----------+------------------+
| deposit          | 20000.00 | 2023-01-15       |
| deposit          | 25000.00 | 2023-07-05       |
| deposit          | 30000.00 | 2023-09-15       |
+------------------+----------+------------------+
3 rows in set (0.00 sec)
```

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



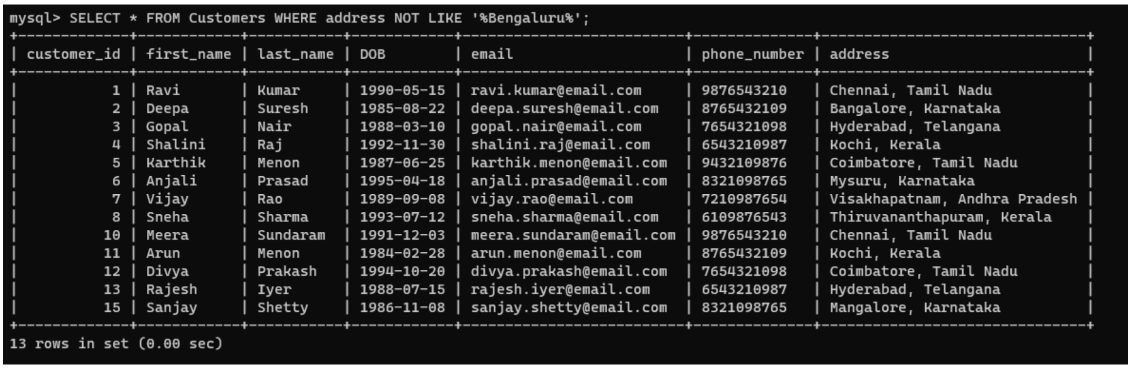
```
mysql> SELECT account_id, balance * 0.05 AS interest_accrued FROM Accounts WHERE account_type = 'savings';
+------------+------------------+
| account_id | interest_accrued |
+------------+------------------+
|        101 |        2500.0000 |
|        104 |        1500.0000 |
|        107 |        4000.0000 |
|        110 |        3000.0000 |
|        113 |        3750.0000 |
|        116 |        3250.0000 |
|        119 |        1900.0000 |
|        122 |        4250.0000 |
|        125 |        3500.0000 |
|        128 |        3900.0000 |
|        132 |        5500.0000 |
|        133 |        3000.0000 |
+------------+------------------+
12 rows in set (0.00 sec)
```

k) 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 < 5000;
+------------+---------+
| account_id | balance |
+------------+---------+
|        130 |    0.00 |
+------------+---------+
1 row in set (0.00 sec)
```

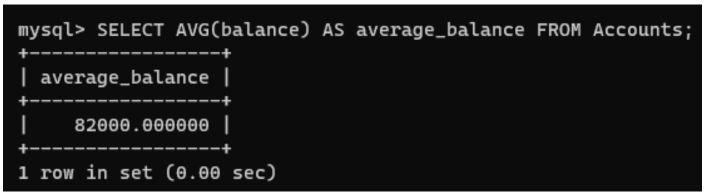
l) Write a SQL query to Find customers not living in a specific city.



```
mysql> SELECT * FROM Customers WHERE address NOT LIKE '%Bengaluru%';
+-------------+------------+-----------+------------+--------------------------+--------------+-------------------------------+
| customer_id | first_name | last_name | DOB        | email                    | phone_number | address                       |
+-------------+------------+-----------+------------+--------------------------+--------------+-------------------------------+
|           1 | Ravi       | Kumar     | 1990-05-15 | ravi.kumar@email.com     | 9876543210   | Chennai, Tamil Nadu           |
|           2 | Deepa      | Suresh    | 1985-08-22 | deepa.suresh@email.com   | 8765432109   | Bangalore, Karnataka          |
|           3 | Gopal      | Nair      | 1988-03-10 | gopal.nair@email.com     | 7654321098   | Hyderabad, Telangana          |
|           4 | Shalini    | Raj       | 1992-11-30 | shalini.raj@email.com    | 6543210987   | Kochi, Kerala                 |
|           5 | Karthik    | Menon     | 1987-06-25 | karthik.menon@email.com  | 9432109876   | Coimbatore, Tamil Nadu        |
|           6 | Anjali     | Prasad    | 1995-04-18 | anjali.prasad@email.com  | 8321098765   | Mysuru, Karnataka             |
|           7 | Vijay      | Rao       | 1989-09-08 | vijay.rao@email.com      | 7210987654   | Visakhapatnam, Andhra Pradesh |
|           8 | Sneha      | Sharma    | 1993-07-12 | sneha.sharma@email.com   | 6109876543   | Thiruvananthapuram, Kerala    |
|          10 | Meera      | Sundaram  | 1991-12-03 | meera.sundaram@email.com | 9876543210   | Chennai, Tamil Nadu           |
|          11 | Arun       | Menon     | 1984-02-28 | arun.menon@email.com     | 8765432109   | Kochi, Kerala                 |
|          12 | Divya      | Prakash   | 1994-10-20 | divya.prakash@email.com  | 7654321098   | Coimbatore, Tamil Nadu        |
|          13 | Rajesh     | Iyer      | 1988-07-15 | rajesh.iyer@email.com    | 6543210987   | Hyderabad, Telangana          |
|          15 | Sanjay     | Shetty    | 1986-11-08 | sanjay.shetty@email.com  | 8321098765   | Mangalore, Karnataka          |
+-------------+------------+-----------+------------+--------------------------+--------------+-------------------------------+
13 rows in set (0.00 sec)
```

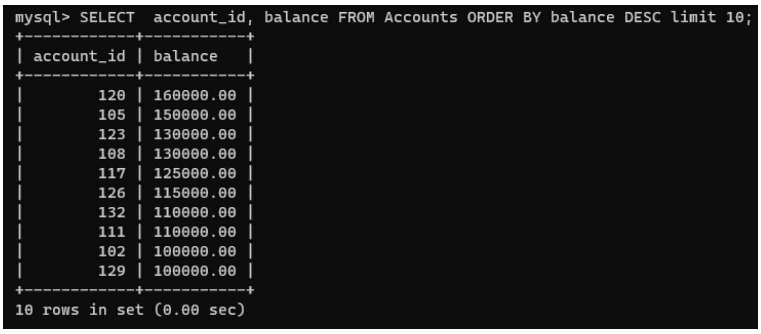
Task-3 Aggregate functions, Having, Order By, Group By and Joins:

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



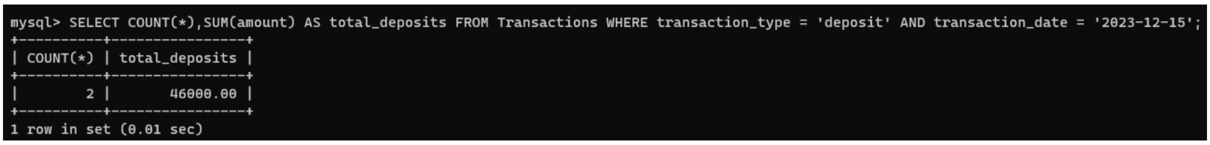
```
mysql> SELECT AVG(balance) AS average_balance FROM Accounts;
+-----------------+
| average_balance |
+-----------------+
|    82000.000000 |
+-----------------+
1 row in set (0.00 sec)
```

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



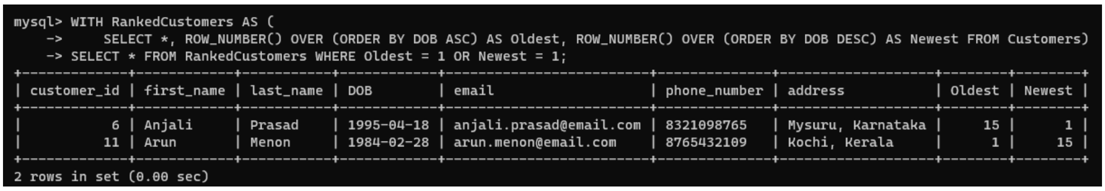
```
mysql> SELECT  account_id, balance FROM Accounts ORDER BY balance DESC limit 10;
+------------+-----------+
| account_id | balance   |
+------------+-----------+
|        120 | 160000.00 |
|        105 | 150000.00 |
|        123 | 130000.00 |
|        108 | 130000.00 |
|        117 | 125000.00 |
|        126 | 115000.00 |
|        132 | 110000.00 |
|        111 | 110000.00 |
|        102 | 100000.00 |
|        129 | 100000.00 |
+------------+-----------+
10 rows in set (0.00 sec)
```

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



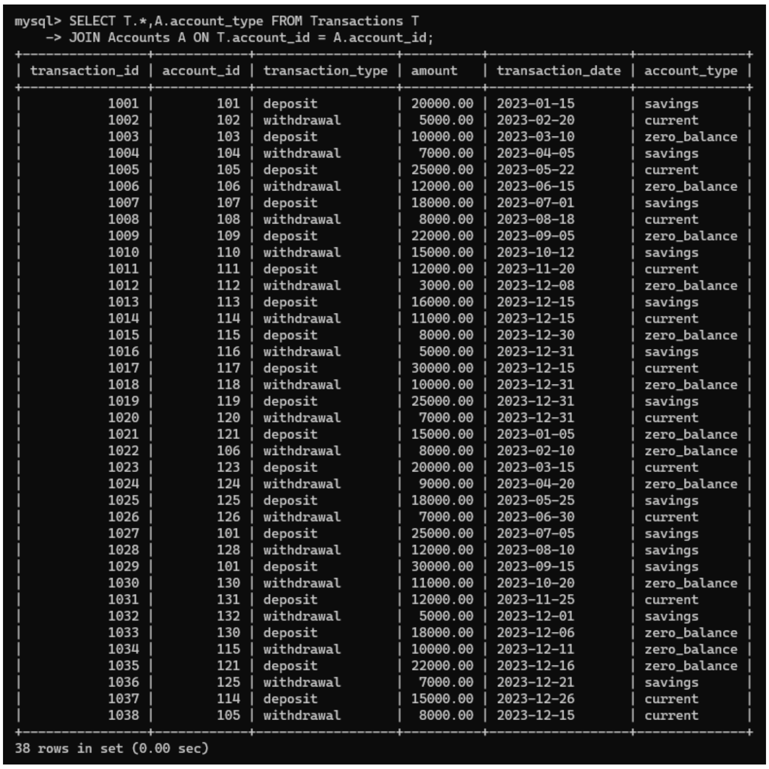
```
mysql> SELECT COUNT(*),SUM(amount) AS total_deposits FROM Transactions WHERE transaction_type = 'deposit' AND transaction_date = '2023-12-15';
+----------+----------------+
| COUNT(*) | total_deposits |
+----------+----------------+
|        2 |       46000.00 |
+----------+----------------+
1 row in set (0.01 sec)
```

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



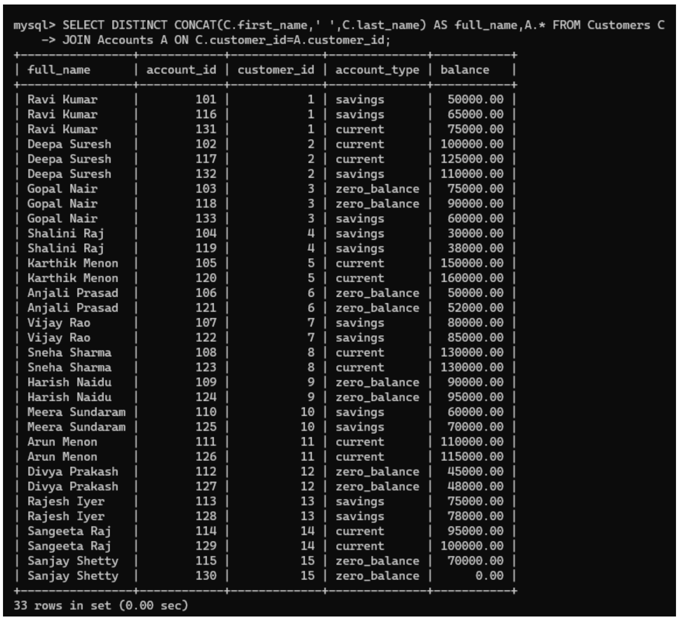
```
mysql> WITH RankedCustomers AS (
    ->     SELECT *, ROW_NUMBER() OVER (ORDER BY DOB ASC) AS Oldest, ROW_NUMBER() OVER (ORDER BY DOB DESC) AS Newest FROM Customers)
    -> SELECT * FROM RankedCustomers WHERE Oldest = 1 OR Newest = 1;
+-------------+------------+-----------+------------+-------------------------+--------------+-------------------+--------+--------+
| customer_id | first_name | last_name | DOB        | email                   | phone_number | address           | Oldest | Newest |
+-------------+------------+-----------+------------+-------------------------+--------------+-------------------+--------+--------+
|           6 | Anjali     | Prasad    | 1995-04-18 | anjali.prasad@email.com | 8321098765   | Mysuru, Karnataka |     15 |      1 |
|          11 | Arun       | Menon     | 1984-02-28 | arun.menon@email.com    | 8765432109   | Kochi, Kerala     |      1 |     15 |
+-------------+------------+-----------+------------+-------------------------+--------------+-------------------+--------+--------+
2 rows in set (0.00 sec)
```

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



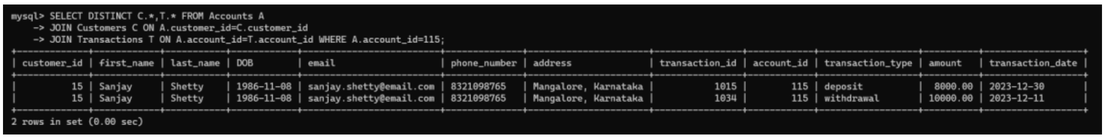
```
mysql> SELECT T.*,A.account_type FROM Transactions T
    -> JOIN Accounts A ON T.account_id = A.account_id;
+----------------+------------+------------------+----------+------------------+--------------+
| transaction_id | account_id | transaction_type | amount   | transaction_date | account_type |
+----------------+------------+------------------+----------+------------------+--------------+
|           1001 |        101 | deposit          | 20000.00 | 2023-01-15       | savings      |
|           1002 |        102 | withdrawal       |  5000.00 | 2023-02-20       | current      |
|           1003 |        103 | deposit          | 10000.00 | 2023-03-10       | zero_balance |
|           1004 |        104 | withdrawal       |  7000.00 | 2023-04-05       | savings      |
|           1005 |        105 | deposit          | 25000.00 | 2023-05-22       | current      |
|           1006 |        106 | withdrawal       | 12000.00 | 2023-06-15       | zero_balance |
|           1007 |        107 | deposit          | 18000.00 | 2023-07-01       | savings      |
|           1008 |        108 | withdrawal       |  8000.00 | 2023-08-18       | current      |
|           1009 |        109 | deposit          | 22000.00 | 2023-09-05       | zero_balance |
|           1010 |        110 | withdrawal       | 15000.00 | 2023-10-12       | savings      |
|           1011 |        111 | deposit          | 12000.00 | 2023-11-20       | current      |
|           1012 |        112 | withdrawal       |  3000.00 | 2023-12-08       | zero_balance |
|           1013 |        113 | deposit          | 16000.00 | 2023-12-15       | savings      |
|           1014 |        114 | withdrawal       | 11000.00 | 2023-12-15       | current      |
|           1015 |        115 | deposit          |  8000.00 | 2023-12-30       | zero_balance |
|           1016 |        116 | withdrawal       |  5000.00 | 2023-12-31       | savings      |
|           1017 |        117 | deposit          | 30000.00 | 2023-12-15       | current      |
|           1018 |        118 | withdrawal       | 10000.00 | 2023-12-31       | zero_balance |
|           1019 |        119 | deposit          | 25000.00 | 2023-12-31       | savings      |
|           1020 |        120 | withdrawal       |  7000.00 | 2023-12-31       | current      |
|           1021 |        121 | deposit          | 15000.00 | 2023-01-05       | zero_balance |
|           1022 |        106 | withdrawal       |  8000.00 | 2023-02-10       | zero_balance |
|           1023 |        123 | deposit          | 20000.00 | 2023-03-15       | current      |
|           1024 |        124 | withdrawal       |  9000.00 | 2023-04-20       | zero_balance |
|           1025 |        125 | deposit          | 18000.00 | 2023-05-25       | savings      |
|           1026 |        126 | withdrawal       |  7000.00 | 2023-06-30       | current      |
|           1027 |        101 | deposit          | 25000.00 | 2023-07-05       | savings      |
|           1028 |        128 | withdrawal       | 12000.00 | 2023-08-10       | savings      |
|           1029 |        101 | deposit          | 30000.00 | 2023-09-15       | savings      |
|           1030 |        130 | withdrawal       | 11000.00 | 2023-10-20       | zero_balance |
|           1031 |        131 | deposit          | 12000.00 | 2023-11-25       | current      |
|           1032 |        132 | withdrawal       |  5000.00 | 2023-12-01       | savings      |
|           1033 |        130 | deposit          | 18000.00 | 2023-12-06       | zero_balance |
|           1034 |        115 | withdrawal       | 10000.00 | 2023-12-11       | zero_balance |
|           1035 |        121 | deposit          | 22000.00 | 2023-12-16       | zero_balance |
|           1036 |        125 | withdrawal       |  7000.00 | 2023-12-21       | savings      |
|           1037 |        114 | deposit          | 15000.00 | 2023-12-26       | current      |
|           1038 |        105 | withdrawal       |  8000.00 | 2023-12-15       | current      |
+----------------+------------+------------------+----------+------------------+--------------+
38 rows in set (0.00 sec)
```

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



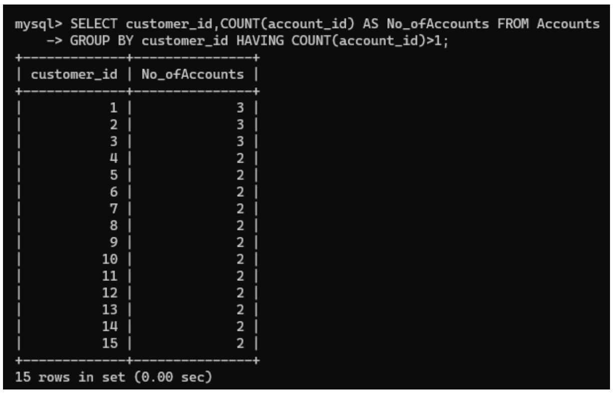
```
mysql> SELECT DISTINCT CONCAT(C.first_name,' ',C.last_name) AS full_name,A.* FROM Customers C
    -> JOIN Accounts A ON C.customer_id=A.customer_id;
+----------------+------------+-------------+--------------+-----------+
| full_name      | account_id | customer_id | account_type | balance   |
+----------------+------------+-------------+--------------+-----------+
| Ravi Kumar     |        101 |           1 | savings      |  50000.00 |
| Ravi Kumar     |        116 |           1 | savings      |  65000.00 |
| Ravi Kumar     |        131 |           1 | current      |  75000.00 |
| Deepa Suresh   |        102 |           2 | current      | 100000.00 |
| Deepa Suresh   |        117 |           2 | current      | 125000.00 |
| Deepa Suresh   |        132 |           2 | savings      | 110000.00 |
| Gopal Nair     |        103 |           3 | zero_balance |  75000.00 |
| Gopal Nair     |        118 |           3 | zero_balance |  90000.00 |
| Gopal Nair     |        133 |           3 | savings      |  60000.00 |
| Shalini Raj    |        104 |           4 | savings      |  30000.00 |
| Shalini Raj    |        119 |           4 | savings      |  38000.00 |
| Karthik Menon  |        105 |           5 | current      | 150000.00 |
| Karthik Menon  |        120 |           5 | current      | 160000.00 |
| Anjali Prasad  |        106 |           6 | zero_balance |  50000.00 |
| Anjali Prasad  |        121 |           6 | zero_balance |  52000.00 |
| Vijay Rao      |        107 |           7 | savings      |  80000.00 |
| Vijay Rao      |        122 |           7 | savings      |  85000.00 |
| Sneha Sharma   |        108 |           8 | current      | 130000.00 |
| Sneha Sharma   |        123 |           8 | current      | 130000.00 |
| Harish Naidu   |        109 |           9 | zero_balance |  90000.00 |
| Harish Naidu   |        124 |           9 | zero_balance |  95000.00 |
| Meera Sundaram |        110 |          10 | savings      |  60000.00 |
| Meera Sundaram |        125 |          10 | savings      |  70000.00 |
| Arun Menon     |        111 |          11 | current      | 110000.00 |
| Arun Menon     |        126 |          11 | current      | 115000.00 |
| Divya Prakash  |        112 |          12 | zero_balance |  45000.00 |
| Divya Prakash  |        127 |          12 | zero_balance |  48000.00 |
| Rajesh Iyer    |        113 |          13 | savings      |  75000.00 |
| Rajesh Iyer    |        128 |          13 | savings      |  78000.00 |
| Sangeeta Raj   |        114 |          14 | current      |  95000.00 |
| Sangeeta Raj   |        129 |          14 | current      | 100000.00 |
| Sanjay Shetty  |        115 |          15 | zero_balance |  70000.00 |
| Sanjay Shetty  |        130 |          15 | zero_balance |      0.00 |
+----------------+------------+-------------+--------------+-----------+
33 rows in set (0.00 sec)
```

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



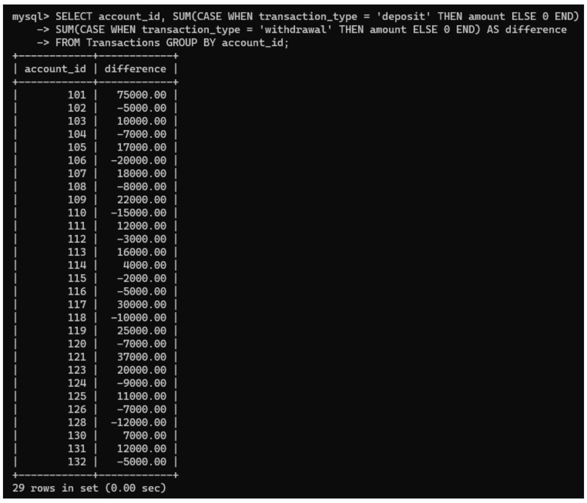
```
mysql> SELECT DISTINCT C.*,T.* FROM Accounts A
    -> JOIN Customers C ON A.customer_id=C.customer_id
    -> JOIN Transactions T ON A.account_id=T.account_id WHERE A.account_id=115;
+-------------+------------+-----------+------------+--------------------------+--------------+---------------------+----------------+------------+------------------+----------+------------------+
| customer_id | first_name | last_name | DOB        | email                    | phone_number | address             | transaction_id | account_id | transaction_type | amount   | transaction_date |
+-------------+------------+-----------+------------+--------------------------+--------------+---------------------+----------------+------------+------------------+----------+------------------+
|          15 | Sanjay     | Shetty    | 1986-11-08 | sanjay.shetty@email.com  | 8321098765   | Mangalore, Karnataka |           1015 |        115 | deposit          |  8000.00 | 2023-12-30       |
|          15 | Sanjay     | Shetty    | 1986-11-08 | sanjay.shetty@email.com  | 8321098765   | Mangalore, Karnataka |           1034 |        115 | withdrawal       | 10000.00 | 2023-12-11       |
+-------------+------------+-----------+------------+--------------------------+--------------+---------------------+----------------+------------+------------------+----------+------------------+
2 rows in set (0.00 sec)
```

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



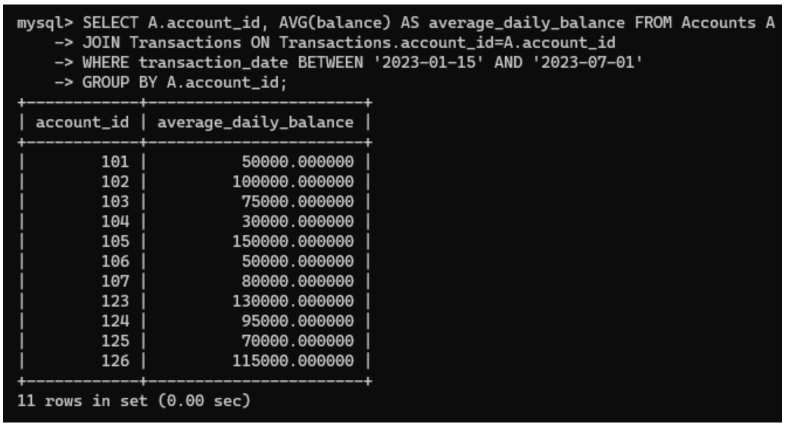
```
mysql> SELECT customer_id,COUNT(account_id) AS No_ofAccounts FROM Accounts
    -> GROUP BY customer_id HAVING COUNT(account_id)>1;
+-------------+---------------+
| customer_id | No_ofAccounts |
+-------------+---------------+
|           1 |             3 |
|           2 |             3 |
|           3 |             3 |
|           4 |             2 |
|           5 |             2 |
|           6 |             2 |
|           7 |             2 |
|           8 |             2 |
|           9 |             2 |
|          10 |             2 |
|          11 |             2 |
|          12 |             2 |
|          13 |             2 |
|          14 |             2 |
|          15 |             2 |
+-------------+---------------+
15 rows in set (0.00 sec)
```

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



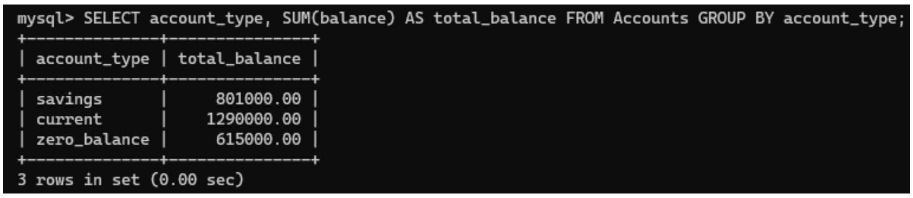
```
mysql> SELECT account_id, SUM(CASE WHEN transaction_type = 'deposit' THEN amount ELSE 0 END)
    -> SUM(CASE WHEN transaction_type = 'withdrawal' THEN amount ELSE 0 END) AS difference
    -> FROM Transactions GROUP BY account_id;
+------------+------------+
| account_id | difference |
+------------+------------+
|        101 |   75000.00 |
|        102 |   -5000.00 |
|        103 |   10000.00 |
|        104 |   -7000.00 |
|        105 |   17000.00 |
|        106 |  -20000.00 |
|        107 |   18000.00 |
|        108 |   -8000.00 |
|        109 |   22000.00 |
|        110 |  -15000.00 |
|        111 |   12000.00 |
|        112 |   -3000.00 |
|        113 |   16000.00 |
|        114 |    4000.00 |
|        115 |   -2000.00 |
|        116 |   -5000.00 |
|        117 |   30000.00 |
|        118 |  -10000.00 |
|        119 |   25000.00 |
|        120 |   -7000.00 |
|        121 |   37000.00 |
|        123 |   20000.00 |
|        124 |   -9000.00 |
|        125 |   11000.00 |
|        126 |   -7000.00 |
|        128 |  -12000.00 |
|        130 |    7000.00 |
|        131 |   12000.00 |
|        132 |   -5000.00 |
+------------+------------+
29 rows in set (0.00 sec)
```

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



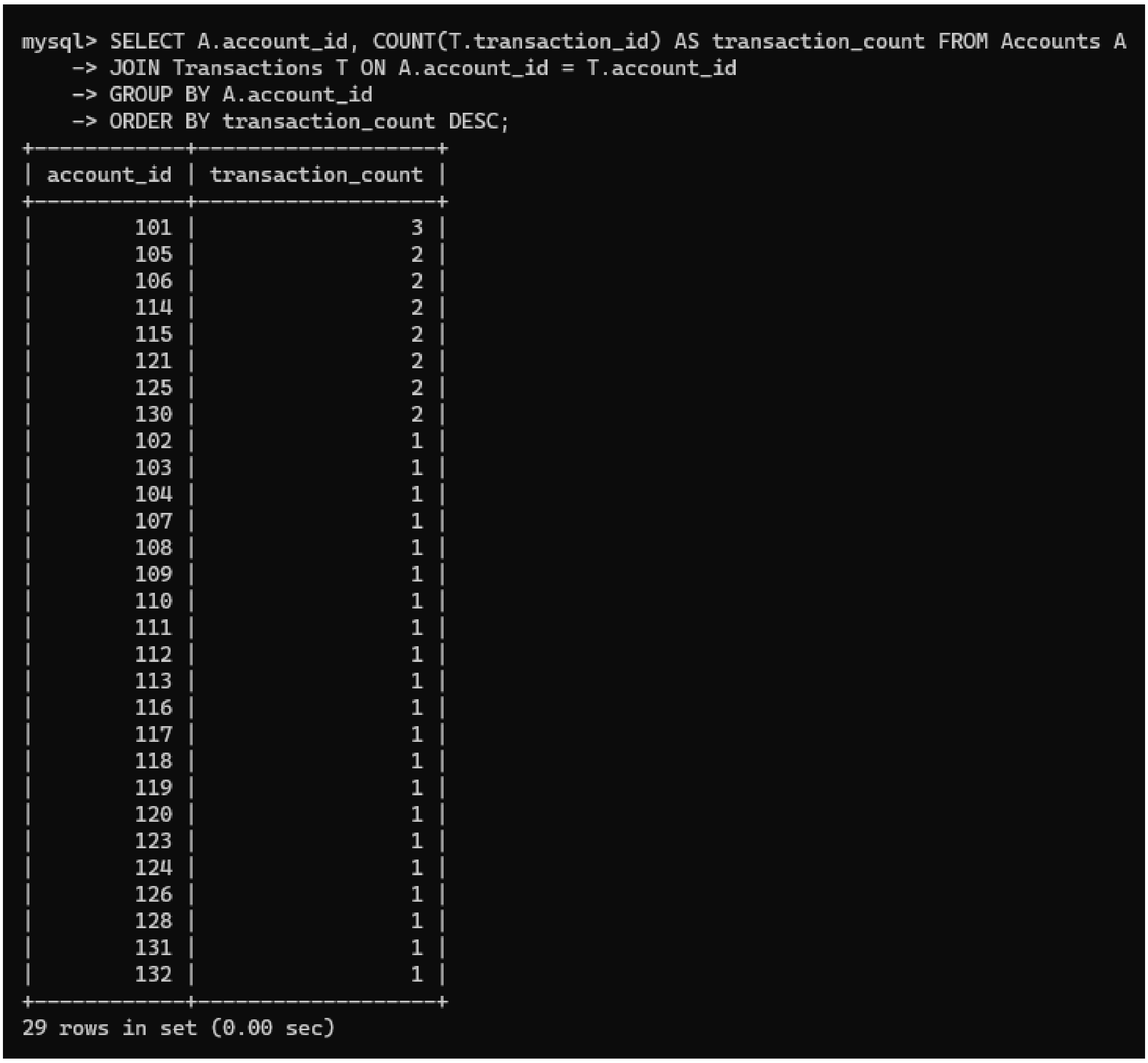
```
mysql> SELECT A.account_id, AVG(balance) AS average_daily_balance FROM Accounts A
    -> JOIN Transactions ON Transactions.account_id=A.account_id
    -> WHERE transaction_date BETWEEN '2023-01-15' AND '2023-07-01'
    -> GROUP BY A.account_id;
+------------+-----------------------+
| account_id | average_daily_balance |
+------------+-----------------------+
|        101 |          50000.000000 |
|        102 |         100000.000000 |
|        103 |          75000.000000 |
|        104 |          30000.000000 |
|        105 |         150000.000000 |
|        106 |          50000.000000 |
|        107 |          80000.000000 |
|        123 |         130000.000000 |
|        124 |          95000.000000 |
|        125 |          70000.000000 |
|        126 |         115000.000000 |
+------------+-----------------------+
11 rows in set (0.00 sec)
```

11. Calculate the total balance for each account type.



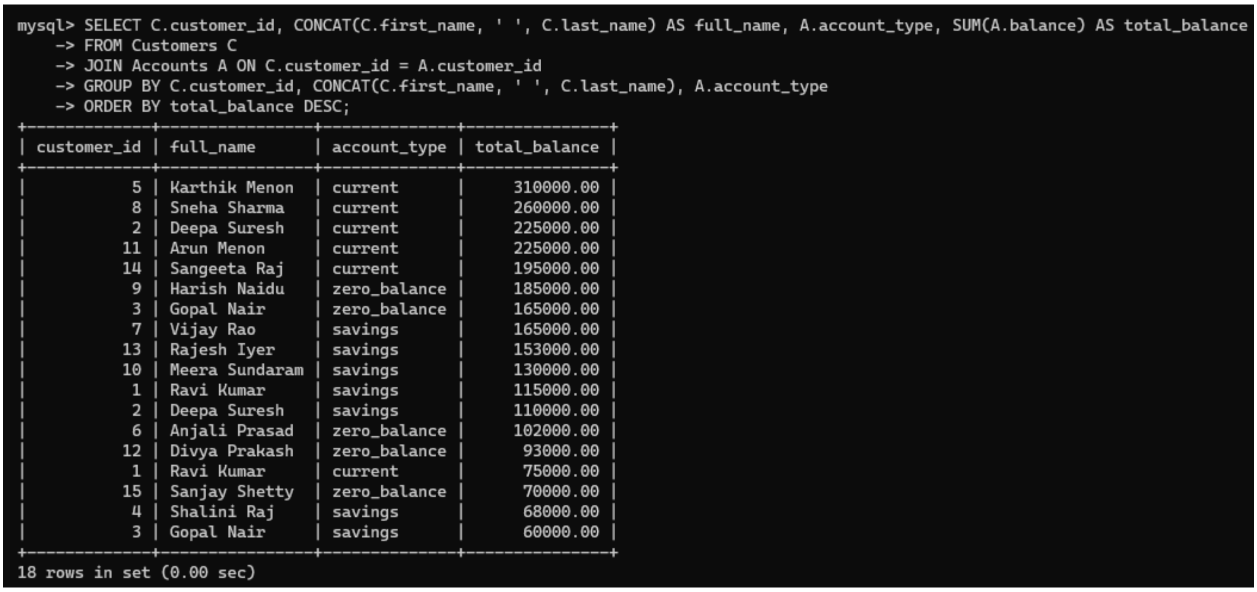
```
mysql> SELECT account_type, SUM(balance) AS total_balance FROM Accounts GROUP BY account_type;
+--------------+---------------+
| account_type | total_balance |
+--------------+---------------+
| savings      |     801000.00 |
| current      |    1290000.00 |
| zero_balance |     615000.00 |
+--------------+---------------+
3 rows in set (0.00 sec)
```

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



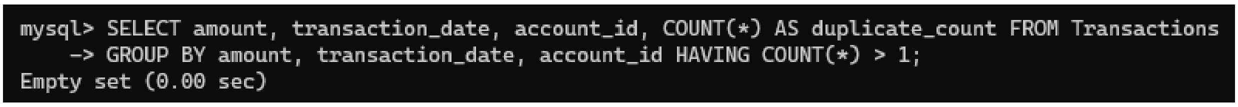
```
mysql> SELECT A.account_id, COUNT(T.transaction_id) AS transaction_count FROM Accounts A
    -> JOIN Transactions T ON A.account_id = T.account_id
    -> GROUP BY A.account_id
    -> ORDER BY transaction_count DESC;
+------------+-------------------+
| account_id | transaction_count |
+------------+-------------------+
|        101 |                 3 |
|        105 |                 2 |
|        106 |                 2 |
|        114 |                 2 |
|        115 |                 2 |
|        121 |                 2 |
|        125 |                 2 |
|        130 |                 2 |
|        102 |                 1 |
|        103 |                 1 |
|        104 |                 1 |
|        107 |                 1 |
|        108 |                 1 |
|        109 |                 1 |
|        110 |                 1 |
|        111 |                 1 |
|        112 |                 1 |
|        113 |                 1 |
|        116 |                 1 |
|        117 |                 1 |
|        118 |                 1 |
|        119 |                 1 |
|        120 |                 1 |
|        123 |                 1 |
|        124 |                 1 |
|        126 |                 1 |
|        128 |                 1 |
|        131 |                 1 |
|        132 |                 1 |
+------------+-------------------+
29 rows in set (0.00 sec)
```

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



```
mysql> SELECT C.customer_id, CONCAT(C.first_name, ' ', C.last_name) AS full_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, CONCAT(C.first_name, ' ', C.last_name), A.account_type
    -> ORDER BY total_balance DESC;
+-------------+----------------+--------------+---------------+
| customer_id | full_name      | account_type | total_balance |
+-------------+----------------+--------------+---------------+
|           5 | Karthik Menon  | current      |     310000.00 |
|           8 | Sneha Sharma   | current      |     260000.00 |
|           2 | Deepa Suresh   | current      |     225000.00 |
|          11 | Arun Menon     | current      |     225000.00 |
|          14 | Sangeeta Raj   | current      |     195000.00 |
|           9 | Harish Naidu   | zero_balance |     185000.00 |
|           3 | Gopal Nair     | zero_balance |     165000.00 |
|           7 | Vijay Rao      | savings      |     165000.00 |
|          13 | Rajesh Iyer    | savings      |     153000.00 |
|          10 | Meera Sundaram | savings      |     130000.00 |
|           1 | Ravi Kumar     | savings      |     115000.00 |
|           2 | Deepa Suresh   | savings      |     110000.00 |
|           6 | Anjali Prasad  | zero_balance |     102000.00 |
|          12 | Divya Prakash  | zero_balance |      93000.00 |
|           1 | Ravi Kumar     | current      |      75000.00 |
|          15 | Sanjay Shetty  | zero_balance |      70000.00 |
|           4 | Shalini Raj    | savings      |      68000.00 |
|           3 | Gopal Nair     | savings      |      60000.00 |
+-------------+----------------+--------------+---------------+
18 rows in set (0.00 sec)
```

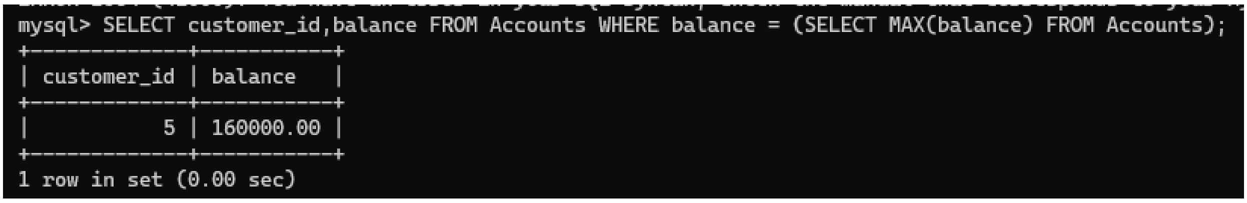
14. Identify and list duplicate transactions based on transaction amount, date, and account.



```
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)
```

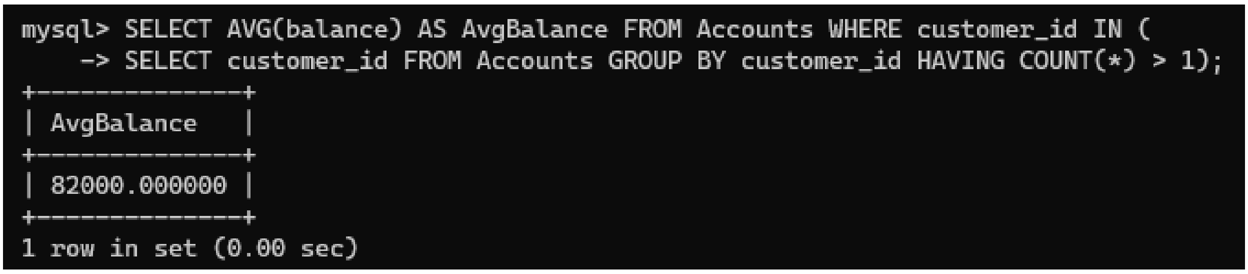
<u>Task-4 Subquery and its type:</u>

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



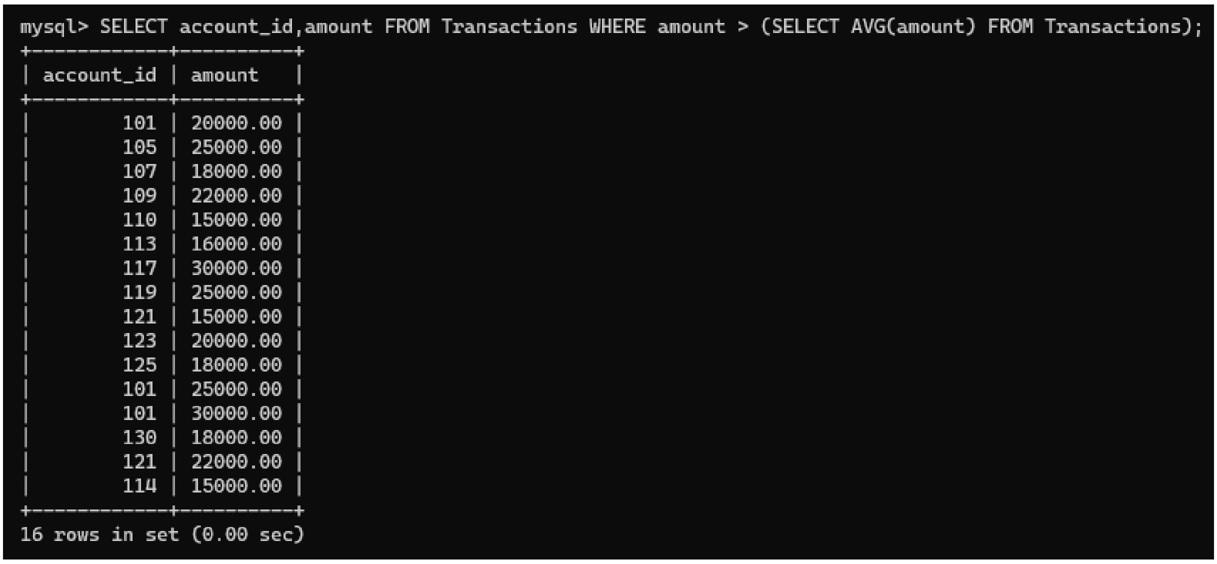
```
mysql> SELECT customer_id,balance FROM Accounts WHERE balance = (SELECT MAX(balance) FROM Accounts);
+-------------+-----------+
| customer_id | balance   |
+-------------+-----------+
|           5 | 160000.00 |
+-------------+-----------+
1 row in set (0.00 sec)
```

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



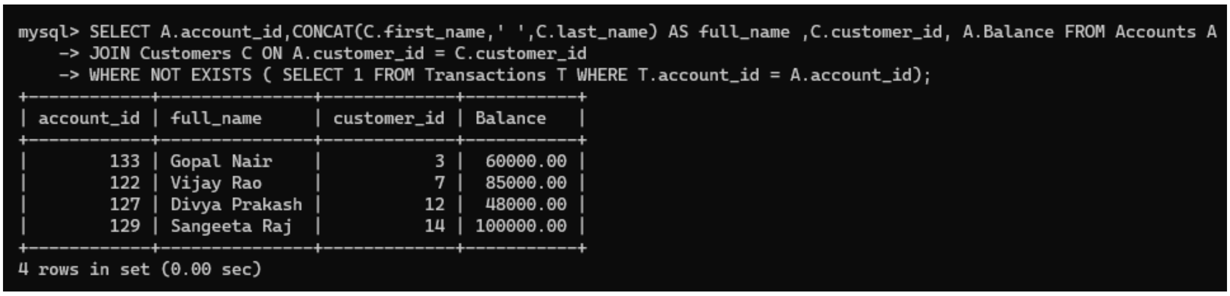
```
mysql> SELECT AVG(balance) AS AvgBalance FROM Accounts WHERE customer_id IN (
    -> SELECT customer_id FROM Accounts GROUP BY customer_id HAVING COUNT(*) > 1);
+--------------+
| AvgBalance   |
+--------------+
| 82000.000000 |
+--------------+
1 row in set (0.00 sec)
```

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



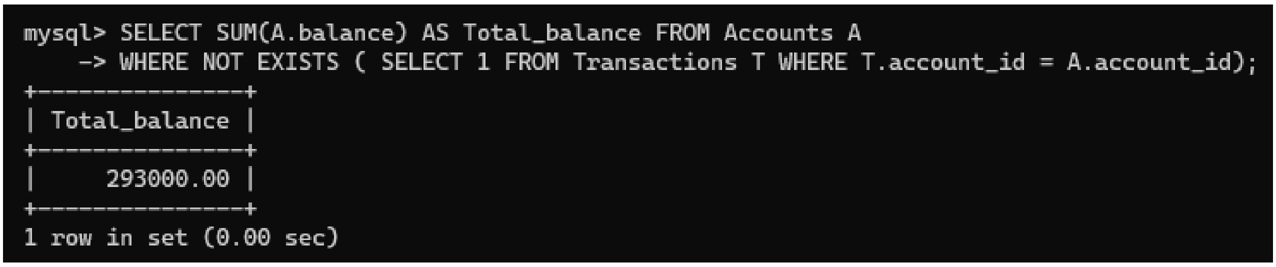
```
mysql> SELECT account_id,amount FROM Transactions WHERE amount > (SELECT AVG(amount) FROM Transactions);
+------------+----------+
| account_id | amount   |
+------------+----------+
|        101 | 20000.00 |
|        105 | 25000.00 |
|        107 | 18000.00 |
|        109 | 22000.00 |
|        110 | 15000.00 |
|        113 | 16000.00 |
|        117 | 30000.00 |
|        119 | 25000.00 |
|        121 | 15000.00 |
|        123 | 20000.00 |
|        125 | 18000.00 |
|        101 | 25000.00 |
|        101 | 30000.00 |
|        130 | 18000.00 |
|        121 | 22000.00 |
|        114 | 15000.00 |
+------------+----------+
16 rows in set (0.00 sec)
```

4. Identify customers who have no recorded transactions.



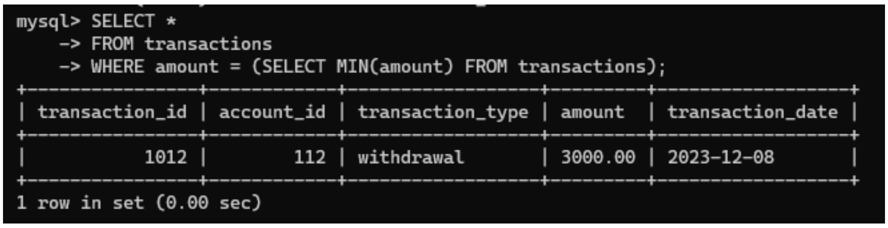
```
mysql> SELECT A.account_id,CONCAT(C.first_name,' ',C.last_name) AS full_name ,C.customer_id, A.Balance FROM Accounts A
    -> JOIN Customers C ON A.customer_id = C.customer_id
    -> WHERE NOT EXISTS ( SELECT 1 FROM Transactions T WHERE T.account_id = A.account_id);
+------------+---------------+-------------+-----------+
| account_id | full_name     | customer_id | Balance   |
+------------+---------------+-------------+-----------+
|        133 | Gopal Nair    |           3 |  60000.00 |
|        122 | Vijay Rao     |           7 |  85000.00 |
|        127 | Divya Prakash |          12 |  48000.00 |
|        129 | Sangeeta Raj  |          14 | 100000.00 |
+------------+---------------+-------------+-----------+
4 rows in set (0.00 sec)
```

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



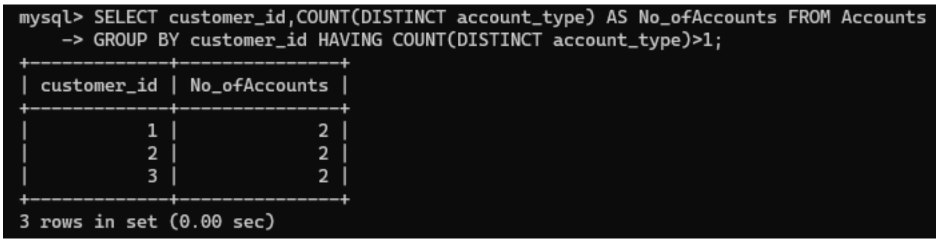
```
mysql> SELECT SUM(A.balance) AS Total_balance FROM Accounts A
    -> WHERE NOT EXISTS ( SELECT 1 FROM Transactions T WHERE T.account_id = A.account_id);
+---------------+
| Total_balance |
+---------------+
|     293000.00 |
+---------------+
1 row in set (0.00 sec)
```

6. Retrieve transactions for accounts with the lowest balance.



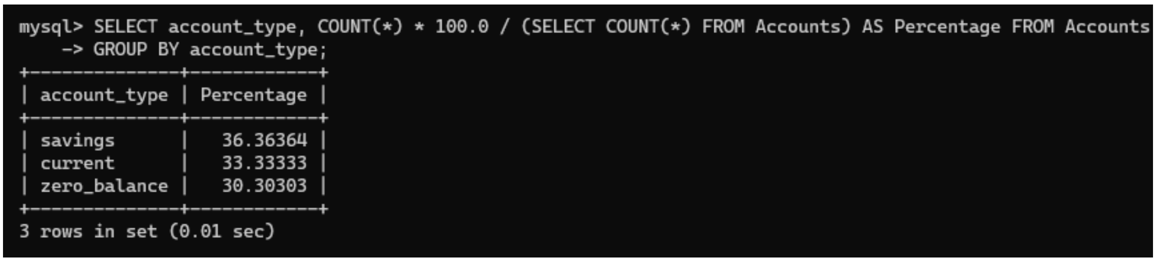
```
mysql> SELECT *
    -> FROM transactions
    -> WHERE amount = (SELECT MIN(amount) FROM transactions);
+----------------+------------+------------------+---------+------------------+
| transaction_id | account_id | transaction_type | amount  | transaction_date |
+----------------+------------+------------------+---------+------------------+
|           1012 |        112 | withdrawal       | 3000.00 | 2023-12-08       |
+----------------+------------+------------------+---------+------------------+
1 row in set (0.00 sec)
```

7. Identify customers who have accounts of multiple types.



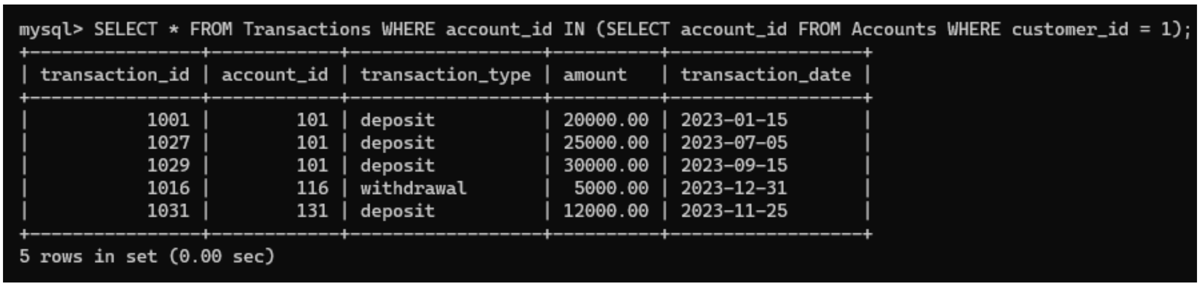
```
mysql> SELECT customer_id,COUNT(DISTINCT account_type) AS No_ofAccounts FROM Accounts
    -> GROUP BY customer_id HAVING COUNT(DISTINCT account_type)>1;
+-------------+---------------+
| customer_id | No_ofAccounts |
+-------------+---------------+
|           1 |             2 |
|           2 |             2 |
|           3 |             2 |
+-------------+---------------+
3 rows in set (0.00 sec)
```

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



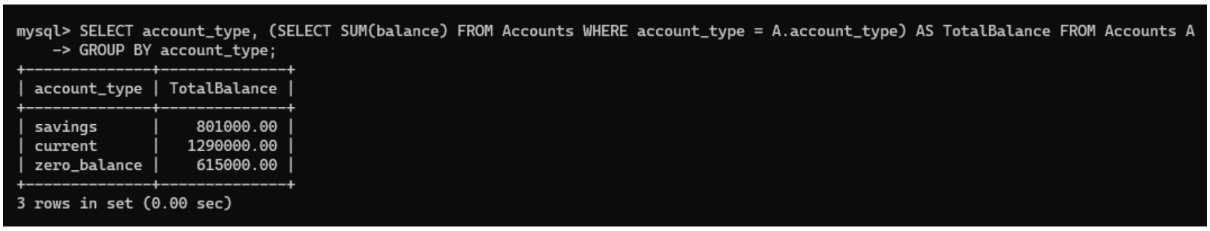
```
mysql> SELECT account_type, COUNT(*) * 100.0 / (SELECT COUNT(*) FROM Accounts) AS Percentage FROM Accounts
    -> GROUP BY account_type;
+--------------+------------+
| account_type | Percentage |
+--------------+------------+
| savings      |   36.36364 |
| current      |   33.33333 |
| zero_balance |   30.30303 |
+--------------+------------+
3 rows in set (0.01 sec)
```

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



```
mysql> SELECT * FROM Transactions WHERE account_id IN (SELECT account_id FROM Accounts WHERE customer_id = 1);
+----------------+------------+------------------+----------+------------------+
| transaction_id | account_id | transaction_type | amount   | transaction_date |
+----------------+------------+------------------+----------+------------------+
|           1001 |        101 | deposit          | 20000.00 | 2023-01-15       |
|           1027 |        101 | deposit          | 25000.00 | 2023-07-05       |
|           1029 |        101 | deposit          | 30000.00 | 2023-09-15       |
|           1016 |        116 | withdrawal       |  5000.00 | 2023-12-31       |
|           1031 |        131 | deposit          | 12000.00 | 2023-11-25       |
+----------------+------------+------------------+----------+------------------+
5 rows in set (0.00 sec)
```

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



```
mysql> SELECT account_type, (SELECT SUM(balance) FROM Accounts WHERE account_type = A.account_type) AS TotalBalance FROM Accounts A
    -> GROUP BY account_type;
+--------------+--------------+
| account_type | TotalBalance |
+--------------+--------------+
| savings      |    801000.00 |
| current      |   1290000.00 |
| zero_balance |    615000.00 |
+--------------+--------------+
3 rows in set (0.00 sec)
```