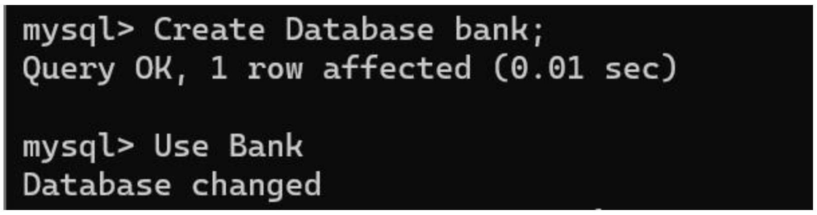
SQL ASSIGNMENT 3
NAME : VENKATA SAI SUKHESH

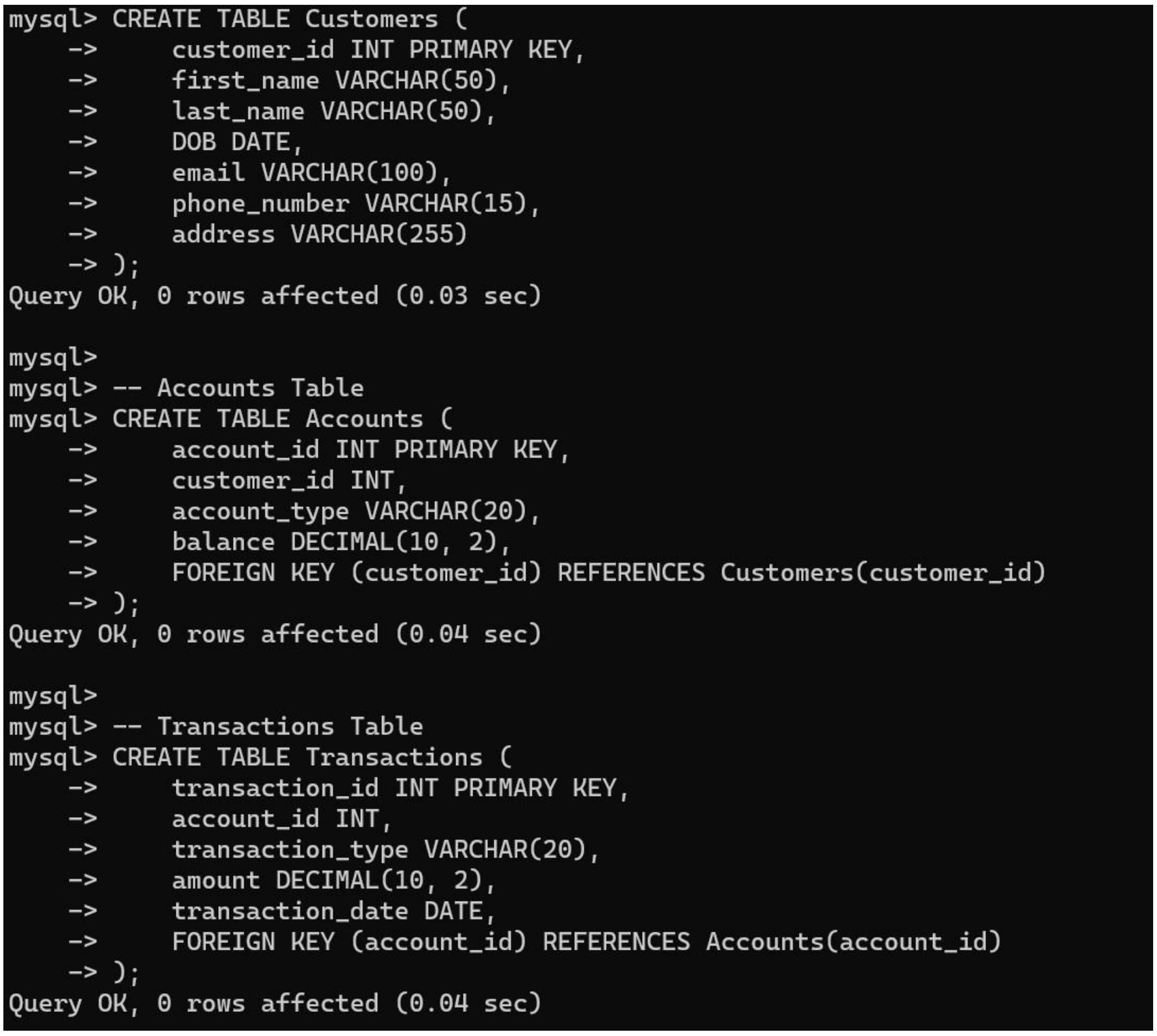
Creating DataBase:



```
mysql> Create Database bank;
Query OK, 1 row affected (0.01 sec)

mysql> Use Bank
Database changed
```

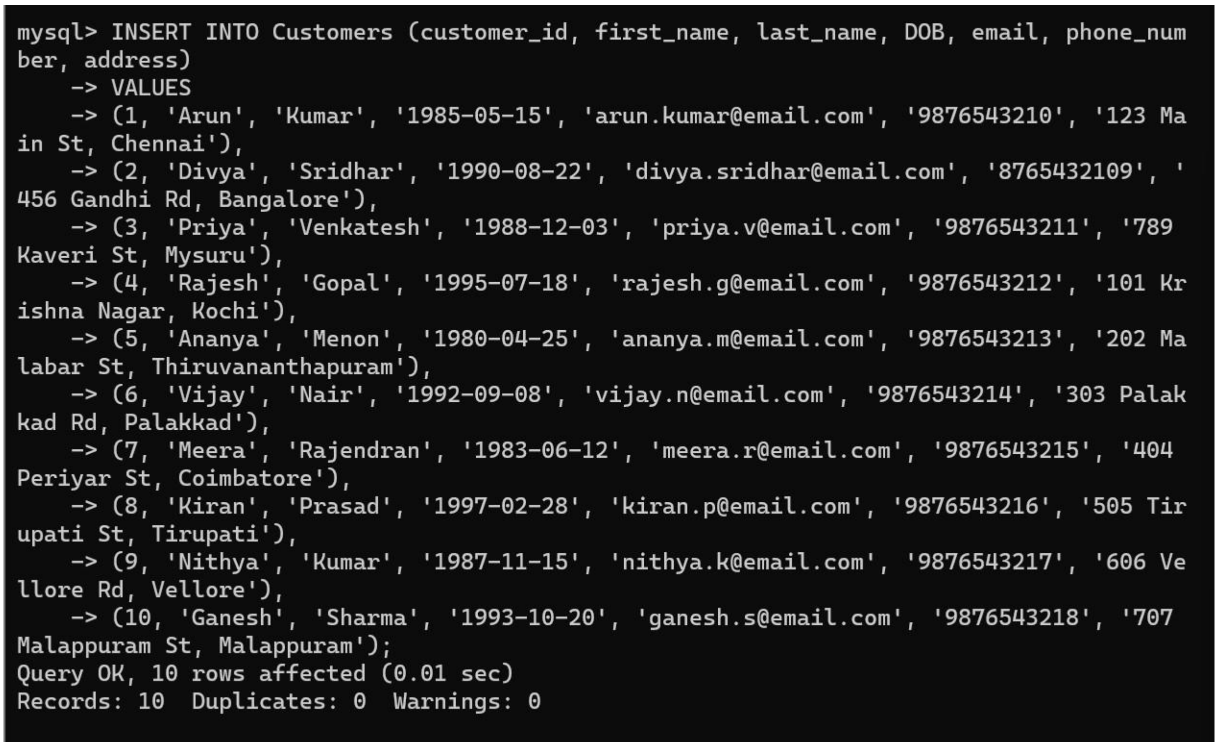
Task 1:

Creating tables:

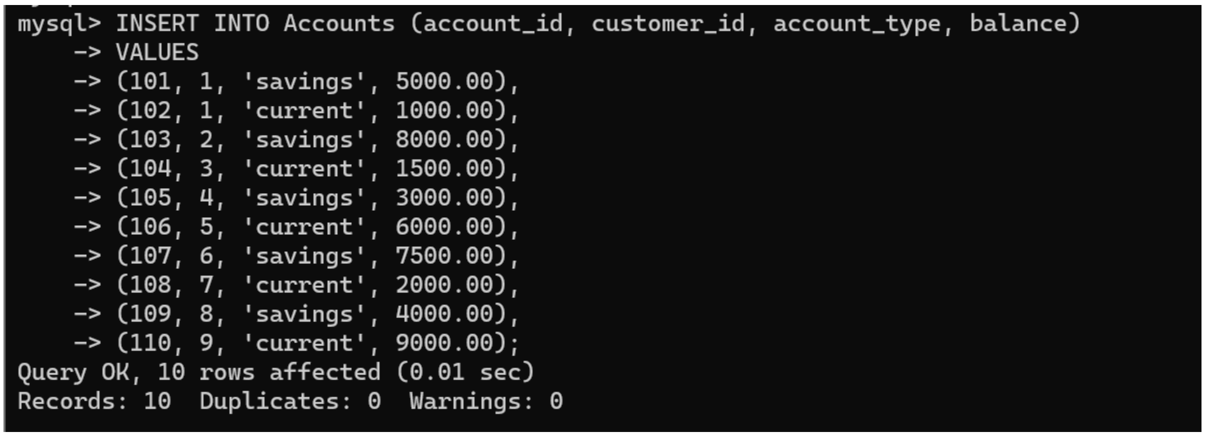


```
mysql> CREATE TABLE Customers (
    ->     customer_id INT PRIMARY KEY,
    ->     first_name VARCHAR(50),
    ->     last_name VARCHAR(50),
    ->     DOB DATE,
    ->     email VARCHAR(100),
    ->     phone_number VARCHAR(15),
    ->     address VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql>
mysql> -- Accounts Table
mysql> CREATE TABLE Accounts (
    ->     account_id INT PRIMARY KEY,
    ->     customer_id INT,
    ->     account_type VARCHAR(20),
    ->     balance DECIMAL(10, 2),
    ->     FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
    -> );
Query OK, 0 rows affected (0.04 sec)

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

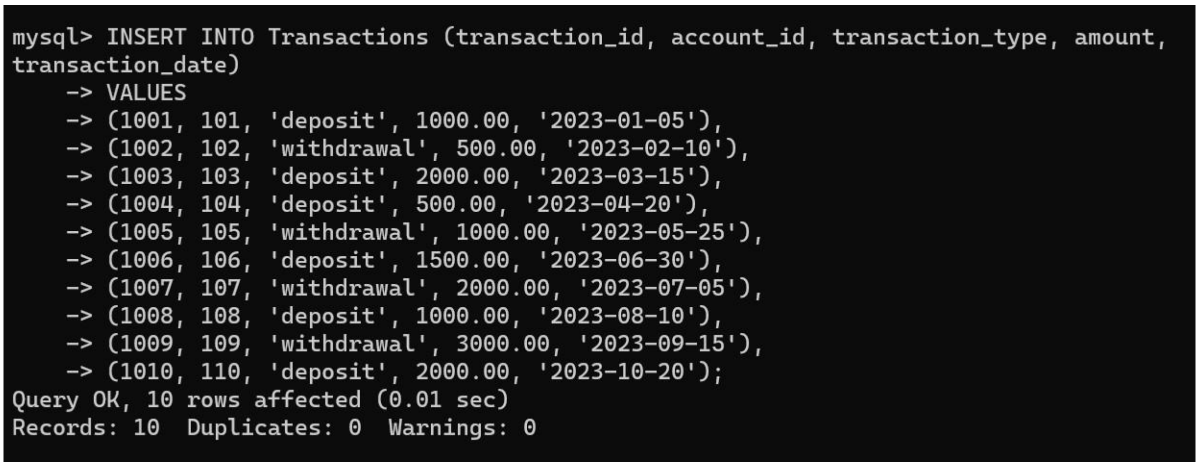
Inserting values:



```
mysql> INSERT INTO Customers (customer_id, first_name, last_name, DOB, email, phone_num
ber, address)
    -> VALUES
    -> (1, 'Arun', 'Kumar', '1985-05-15', 'arun.kumar@email.com', '9876543210', '123 Ma
in St, Chennai'),
    -> (2, 'Divya', 'Sridhar', '1990-08-22', 'divya.sridhar@email.com', '8765432109', '
456 Gandhi Rd, Bangalore'),
    -> (3, 'Priya', 'Venkatesh', '1988-12-03', 'priya.v@email.com', '9876543211', '789
Kaveri St, Mysuru'),
    -> (4, 'Rajesh', 'Gopal', '1995-07-18', 'rajesh.g@email.com', '9876543212', '101 Kr
ishna Nagar, Kochi'),
    -> (5, 'Ananya', 'Menon', '1980-04-25', 'ananya.m@email.com', '9876543213', '202 Ma
labar St, Thiruvananthapuram'),
    -> (6, 'Vijay', 'Nair', '1992-09-08', 'vijay.n@email.com', '9876543214', '303 Palak
kad Rd, Palakkad'),
    -> (7, 'Meera', 'Rajendran', '1983-06-12', 'meera.r@email.com', '9876543215', '404
Periyar St, Coimbatore'),
    -> (8, 'Kiran', 'Prasad', '1997-02-28', 'kiran.p@email.com', '9876543216', '505 Tir
upati St, Tirupati'),
    -> (9, 'Nithya', 'Kumar', '1987-11-15', 'nithya.k@email.com', '9876543217', '606 Ve
llore Rd, Vellore'),
    -> (10, 'Ganesh', 'Sharma', '1993-10-20', 'ganesh.s@email.com', '9876543218', '707
Malappuram St, Malappuram');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```



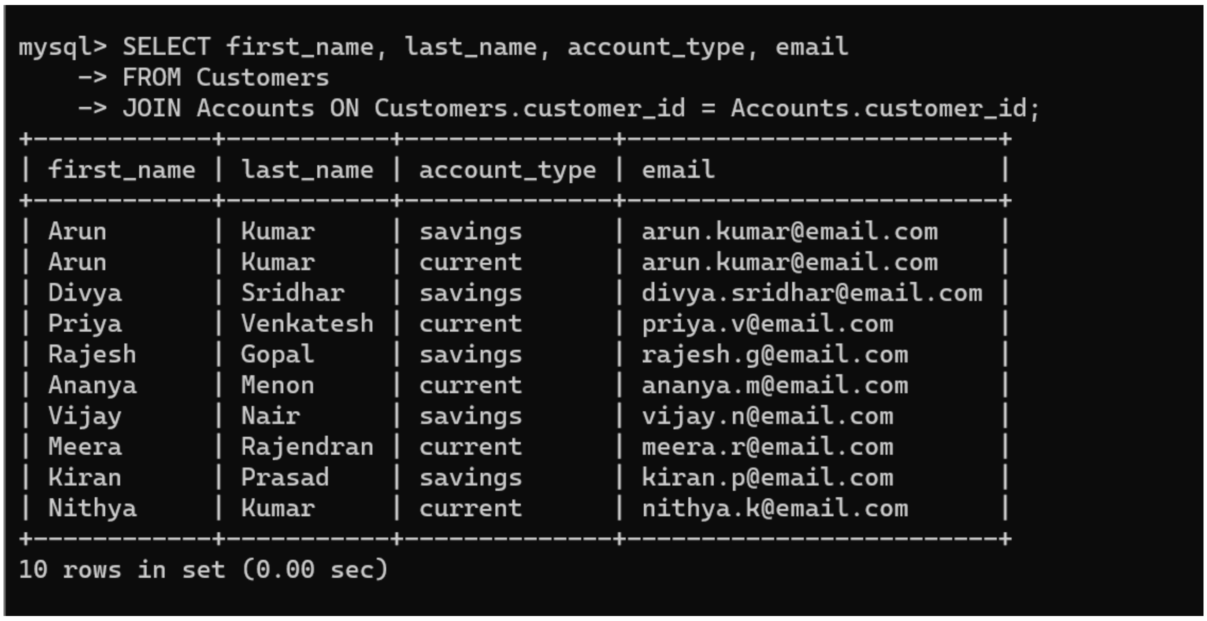
```
mysql> INSERT INTO Accounts (account_id, customer_id, account_type, balance)
    -> VALUES
    -> (101, 1, 'savings', 5000.00),
    -> (102, 1, 'current', 1000.00),
    -> (103, 2, 'savings', 8000.00),
    -> (104, 3, 'current', 1500.00),
    -> (105, 4, 'savings', 3000.00),
    -> (106, 5, 'current', 6000.00),
    -> (107, 6, 'savings', 7500.00),
    -> (108, 7, 'current', 2000.00),
    -> (109, 8, 'savings', 4000.00),
    -> (110, 9, 'current', 9000.00);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```



```
mysql> INSERT INTO Transactions (transaction_id, account_id, transaction_type, amount,
transaction_date)
    -> VALUES
    -> (1001, 101, 'deposit', 1000.00, '2023-01-05'),
    -> (1002, 102, 'withdrawal', 500.00, '2023-02-10'),
    -> (1003, 103, 'deposit', 2000.00, '2023-03-15'),
    -> (1004, 104, 'deposit', 500.00, '2023-04-20'),
    -> (1005, 105, 'withdrawal', 1000.00, '2023-05-25'),
    -> (1006, 106, 'deposit', 1500.00, '2023-06-30'),
    -> (1007, 107, 'withdrawal', 2000.00, '2023-07-05'),
    -> (1008, 108, 'deposit', 1000.00, '2023-08-10'),
    -> (1009, 109, 'withdrawal', 3000.00, '2023-09-15'),
    -> (1010, 110, 'deposit', 2000.00, '2023-10-20');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

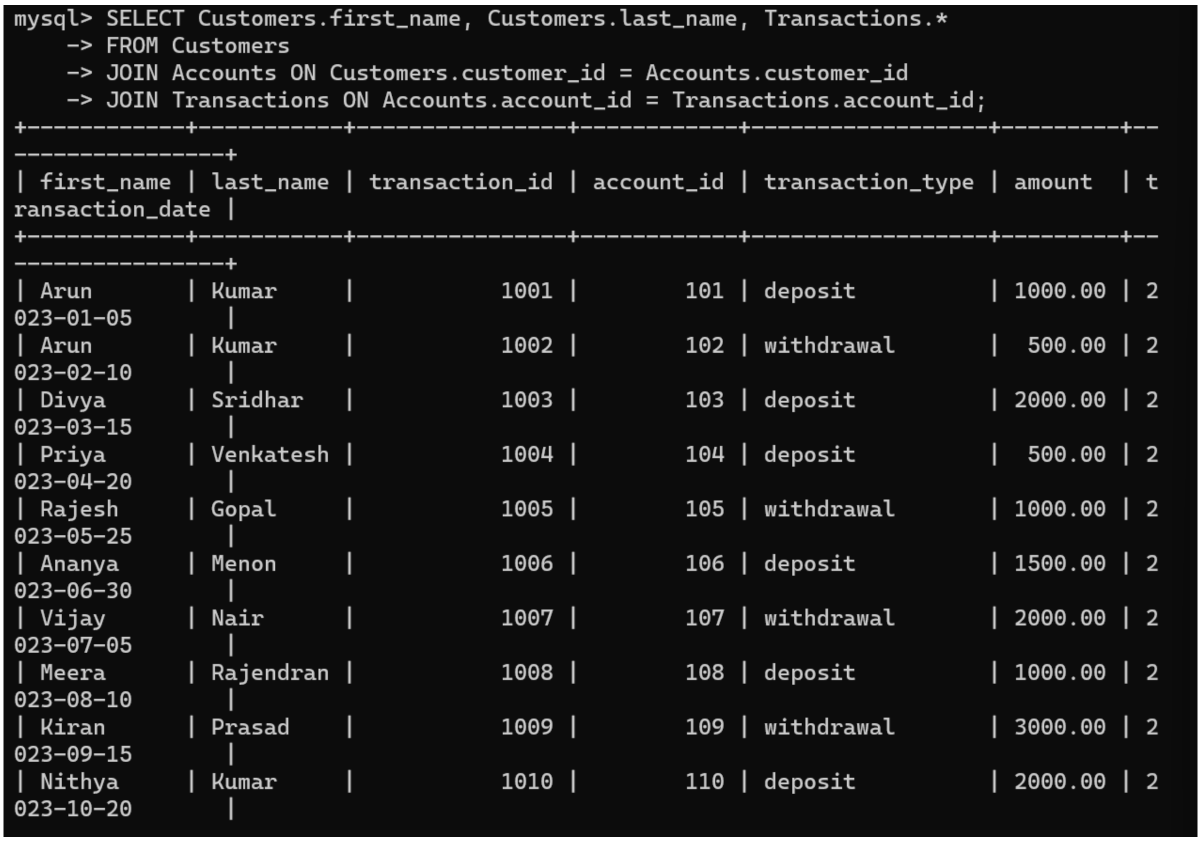
Task 2:

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



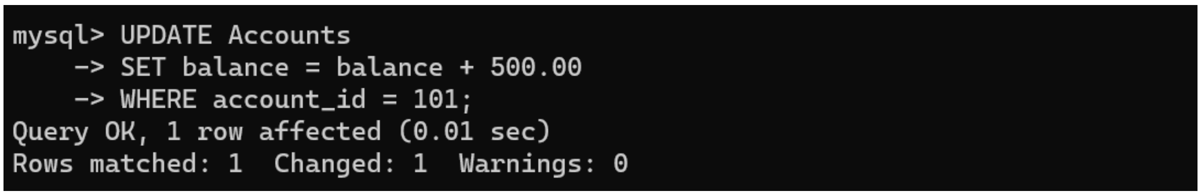
```
mysql> SELECT first_name, last_name, account_type, email
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id;
+------------+-----------+--------------+------------------------+
| first_name | last_name | account_type | email                  |
+------------+-----------+--------------+------------------------+
| Arun       | Kumar     | savings      | arun.kumar@email.com   |
| Arun       | Kumar     | current      | arun.kumar@email.com   |
| Divya      | Sridhar   | savings      | divya.sridhar@email.com |
| Priya      | Venkatesh | current      | priya.v@email.com      |
| Rajesh     | Gopal     | savings      | rajesh.g@email.com     |
| Ananya     | Menon     | current      | ananya.m@email.com     |
| Vijay      | Nair      | savings      | vijay.n@email.com      |
| Meera      | Rajendran | current      | meera.r@email.com      |
| Kiran      | Prasad    | savings      | kiran.p@email.com      |
| Nithya     | Kumar     | current      | nithya.k@email.com     |
+------------+-----------+--------------+------------------------+
10 rows in set (0.00 sec)
```

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



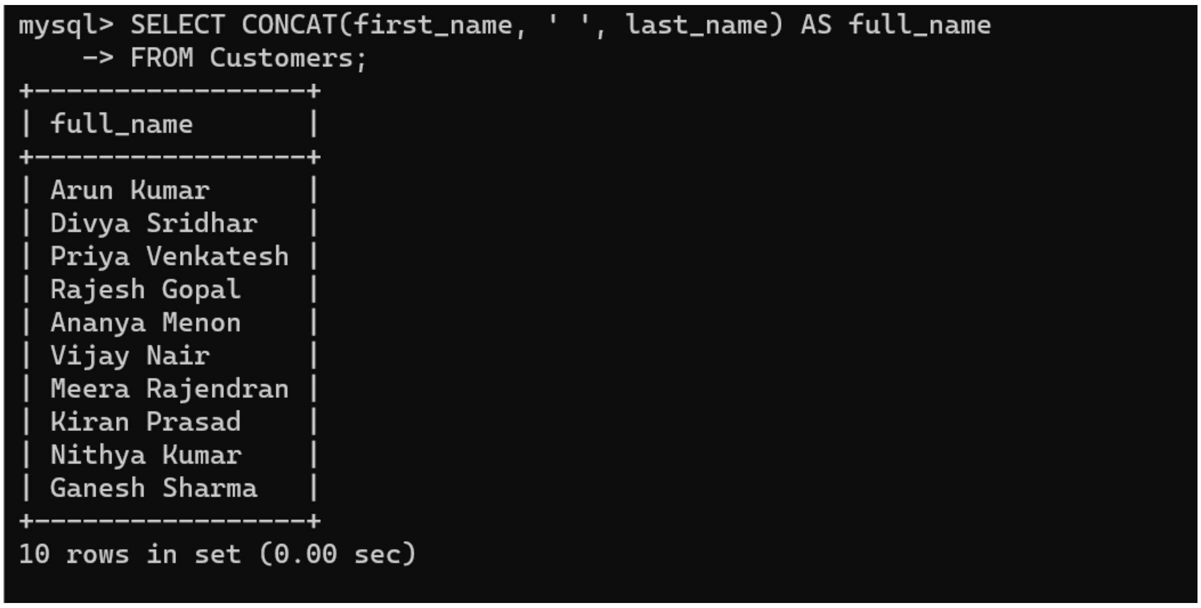
```
mysql> SELECT Customers.first_name, Customers.last_name, Transactions.*
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id
    -> JOIN Transactions ON Accounts.account_id = Transactions.account_id;
+------------+-----------+----------------+------------+------------------+---------+--
----------------+
| first_name | last_name | transaction_id | account_id | transaction_type | amount  | t
ransaction_date |
+------------+-----------+----------------+------------+------------------+---------+--
----------------+
| Arun       | Kumar     |           1001 |        101 | deposit          | 1000.00 | 2
023-01-05       |
| Arun       | Kumar     |           1002 |        102 | withdrawal       |  500.00 | 2
023-02-10       |
| Divya      | Sridhar   |           1003 |        103 | deposit          | 2000.00 | 2
023-03-15       |
| Priya      | Venkatesh |           1004 |        104 | deposit          |  500.00 | 2
023-04-20       |
| Rajesh     | Gopal     |           1005 |        105 | withdrawal       | 1000.00 | 2
023-05-25       |
| Ananya     | Menon     |           1006 |        106 | deposit          | 1500.00 | 2
023-06-30       |
| Vijay      | Nair      |           1007 |        107 | withdrawal       | 2000.00 | 2
023-07-05       |
| Meera      | Rajendran |           1008 |        108 | deposit          | 1000.00 | 2
023-08-10       |
| Kiran      | Prasad    |           1009 |        109 | withdrawal       | 3000.00 | 2
023-09-15       |
| Nithya     | Kumar     |           1010 |        110 | deposit          | 2000.00 | 2
023-10-20       |
```

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



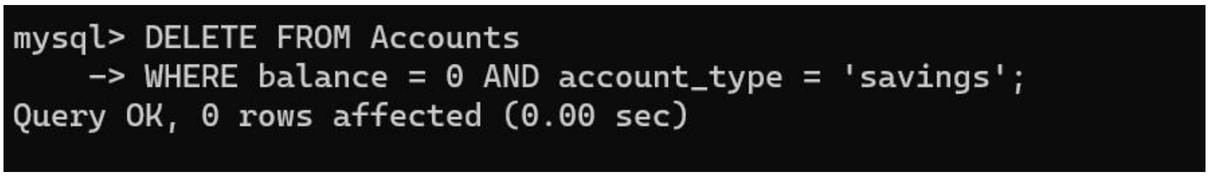
```
mysql> UPDATE Accounts
    -> SET balance = balance + 500.00
    -> WHERE account_id = 101;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

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



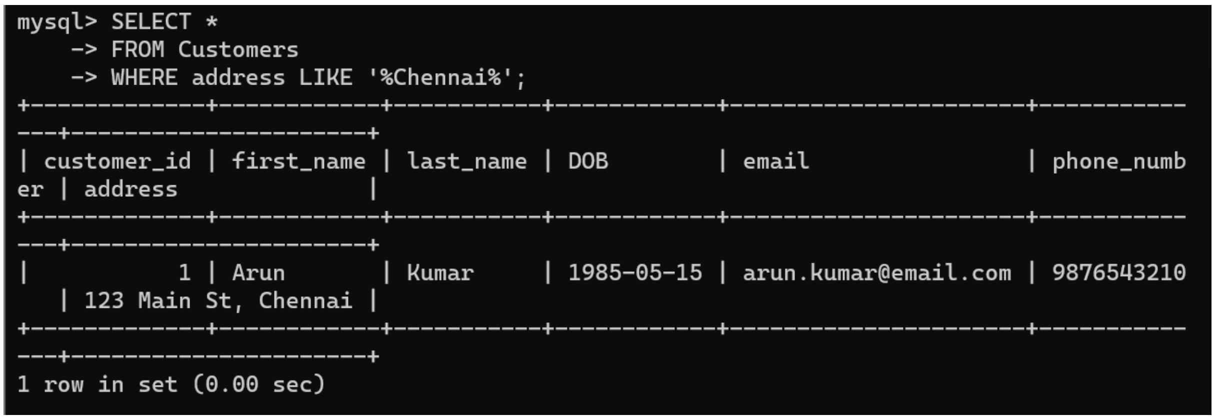
```
mysql> SELECT CONCAT(first_name, ' ', last_name) AS full_name
    -> FROM Customers;
+-----------------+
| full_name       |
+-----------------+
| Arun Kumar      |
| Divya Sridhar   |
| Priya Venkatesh |
| Rajesh Gopal    |
| Ananya Menon    |
| Vijay Nair      |
| Meera Rajendran |
| Kiran Prasad    |
| Nithya Kumar    |
| Ganesh Sharma   |
+-----------------+
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.



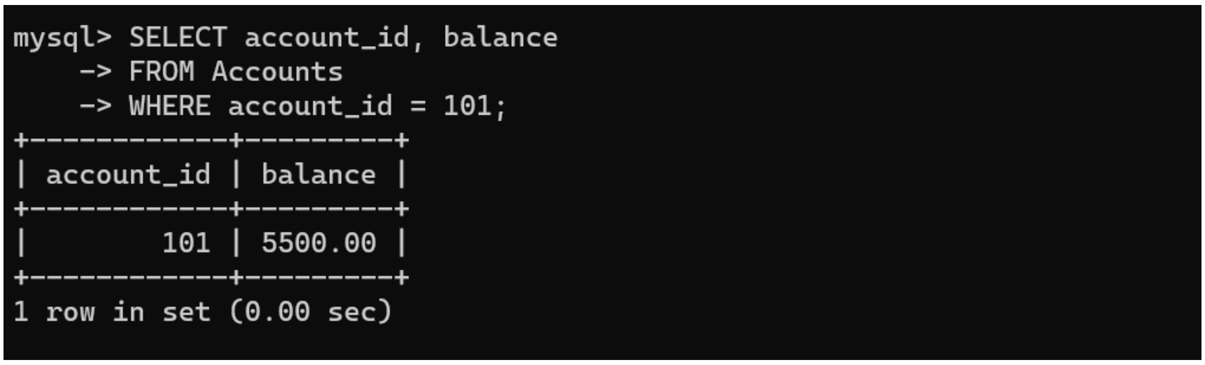
```
mysql> DELETE FROM Accounts
    -> WHERE balance = 0 AND account_type = 'savings';
Query OK, 0 rows affected (0.00 sec)
```

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



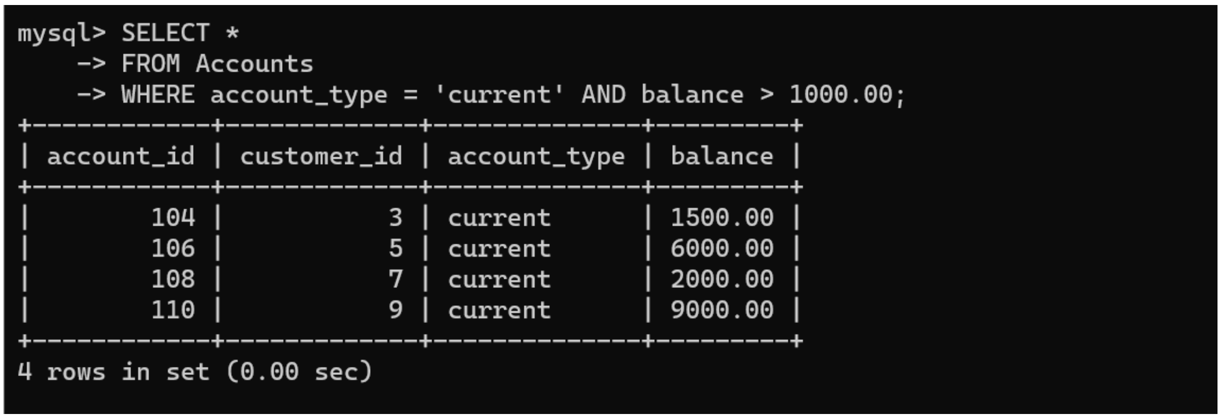
```
mysql> SELECT *
    -> FROM Customers
    -> WHERE address LIKE '%Chennai%';
+-------------+------------+-----------+------------+----------------------+-----------
---+----------------------+
| customer_id | first_name | last_name | DOB        | email                | phone_numb
er | address              |
+-------------+------------+-----------+------------+----------------------+-----------
---+----------------------+
|           1 | Arun       | Kumar     | 1985-05-15 | arun.kumar@email.com | 9876543210
   | 123 Main St, Chennai |
+-------------+------------+-----------+------------+----------------------+-----------
---+----------------------+
1 row in set (0.00 sec)
```

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



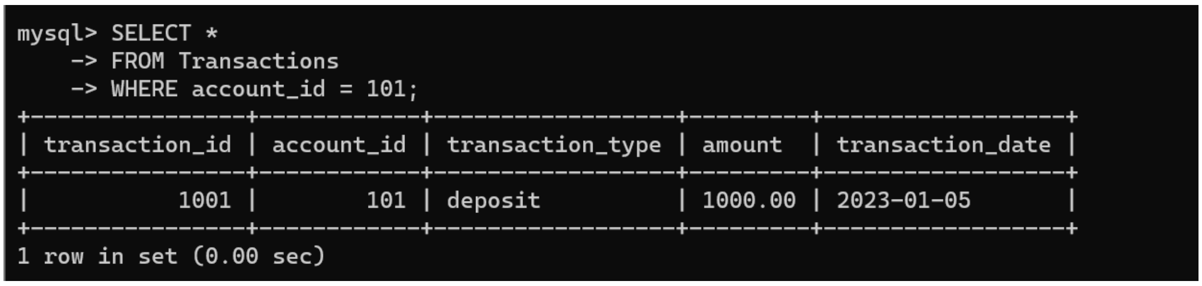
```
mysql> SELECT account_id, balance
    -> FROM Accounts
    -> WHERE account_id = 101;
+------------+---------+
| account_id | balance |
+------------+---------+
|        101 | 5500.00 |
+------------+---------+
1 row in set (0.00 sec)
```

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



```
mysql> SELECT *
    -> FROM Accounts
    -> WHERE account_type = 'current' AND balance > 1000.00;
+------------+-------------+--------------+---------+
| account_id | customer_id | account_type | balance |
+------------+-------------+--------------+---------+
|        104 |           3 | current      | 1500.00 |
|        106 |           5 | current      | 6000.00 |
|        108 |           7 | current      | 2000.00 |
|        110 |           9 | current      | 9000.00 |
+------------+-------------+--------------+---------+
4 rows in set (0.00 sec)
```

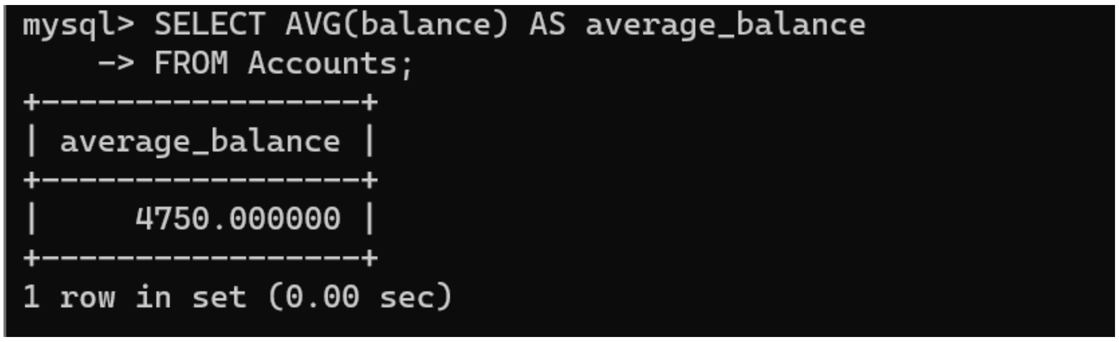
9. Write a SQL query to Retrieve all transactions for a specific account.



```
mysql> SELECT *
    -> FROM Transactions
    -> WHERE account_id = 101;
+----------------+------------+------------------+---------+------------------+
| transaction_id | account_id | transaction_type | amount  | transaction_date |
+----------------+------------+------------------+---------+------------------+
|           1001 |        101 | deposit          | 1000.00 | 2023-01-05       |
+----------------+------------+------------------+---------+------------------+
1 row in set (0.00 sec)
```

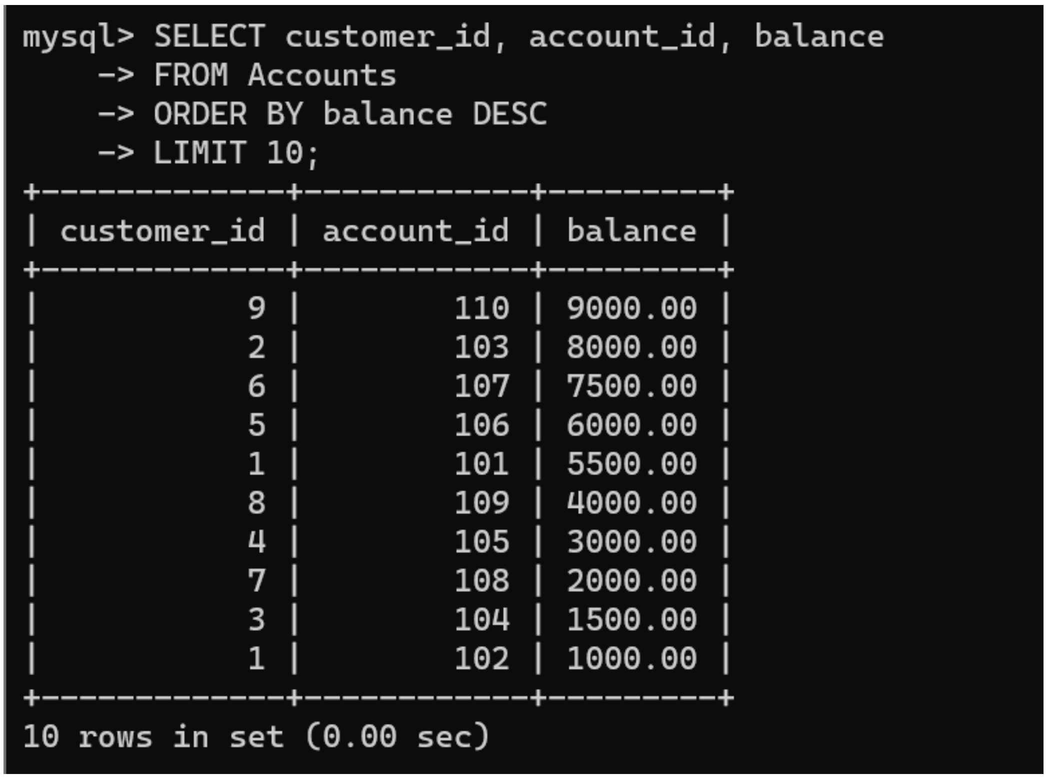
Task 3:

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



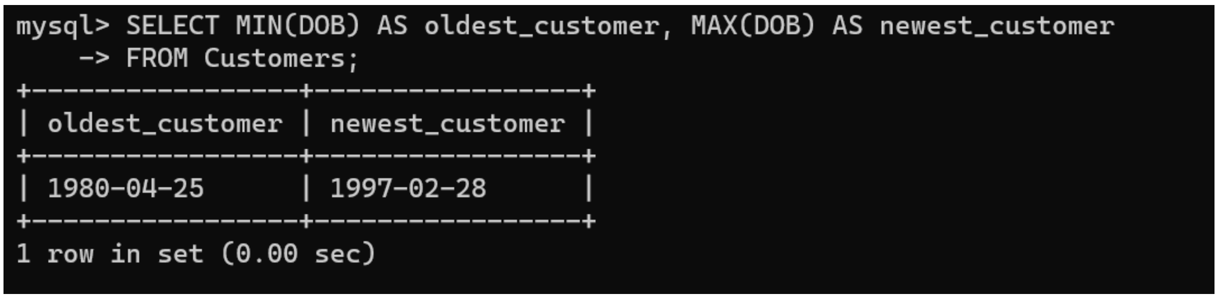
```
mysql> SELECT AVG(balance) AS average_balance
    -> FROM Accounts;
+-----------------+
| average_balance |
+-----------------+
|     4750.000000 |
+-----------------+
1 row in set (0.00 sec)
```

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



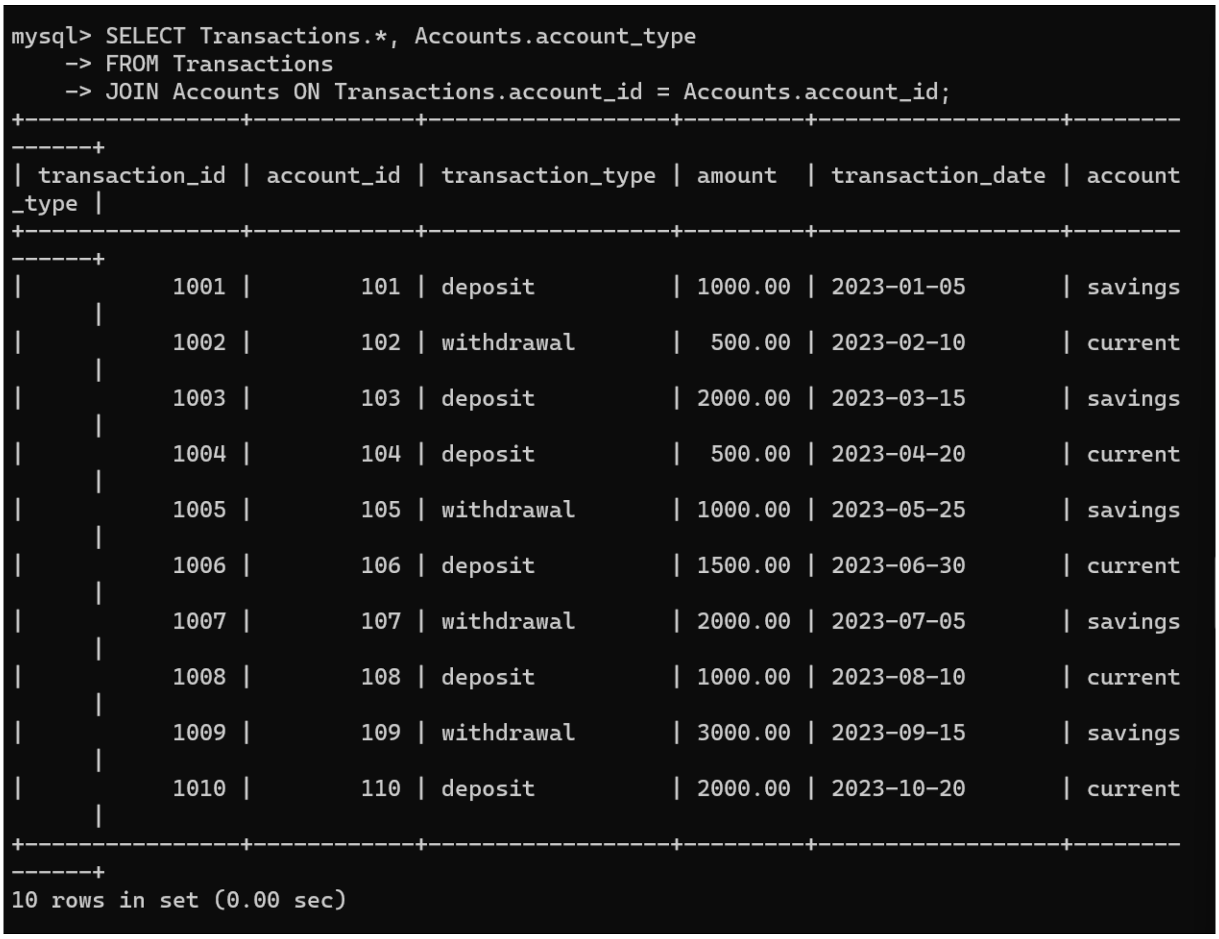
```
mysql> SELECT customer_id, account_id, balance
    -> FROM Accounts
    -> ORDER BY balance DESC
    -> LIMIT 10;
+-------------+------------+---------+
| customer_id | account_id | balance |
+-------------+------------+---------+
|           9 |        110 | 9000.00 |
|           2 |        103 | 8000.00 |
|           6 |        107 | 7500.00 |
|           5 |        106 | 6000.00 |
|           1 |        101 | 5500.00 |
|           8 |        109 | 4000.00 |
|           4 |        105 | 3000.00 |
|           7 |        108 | 2000.00 |
|           3 |        104 | 1500.00 |
|           1 |        102 | 1000.00 |
+-------------+------------+---------+
10 rows in set (0.00 sec)
```

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



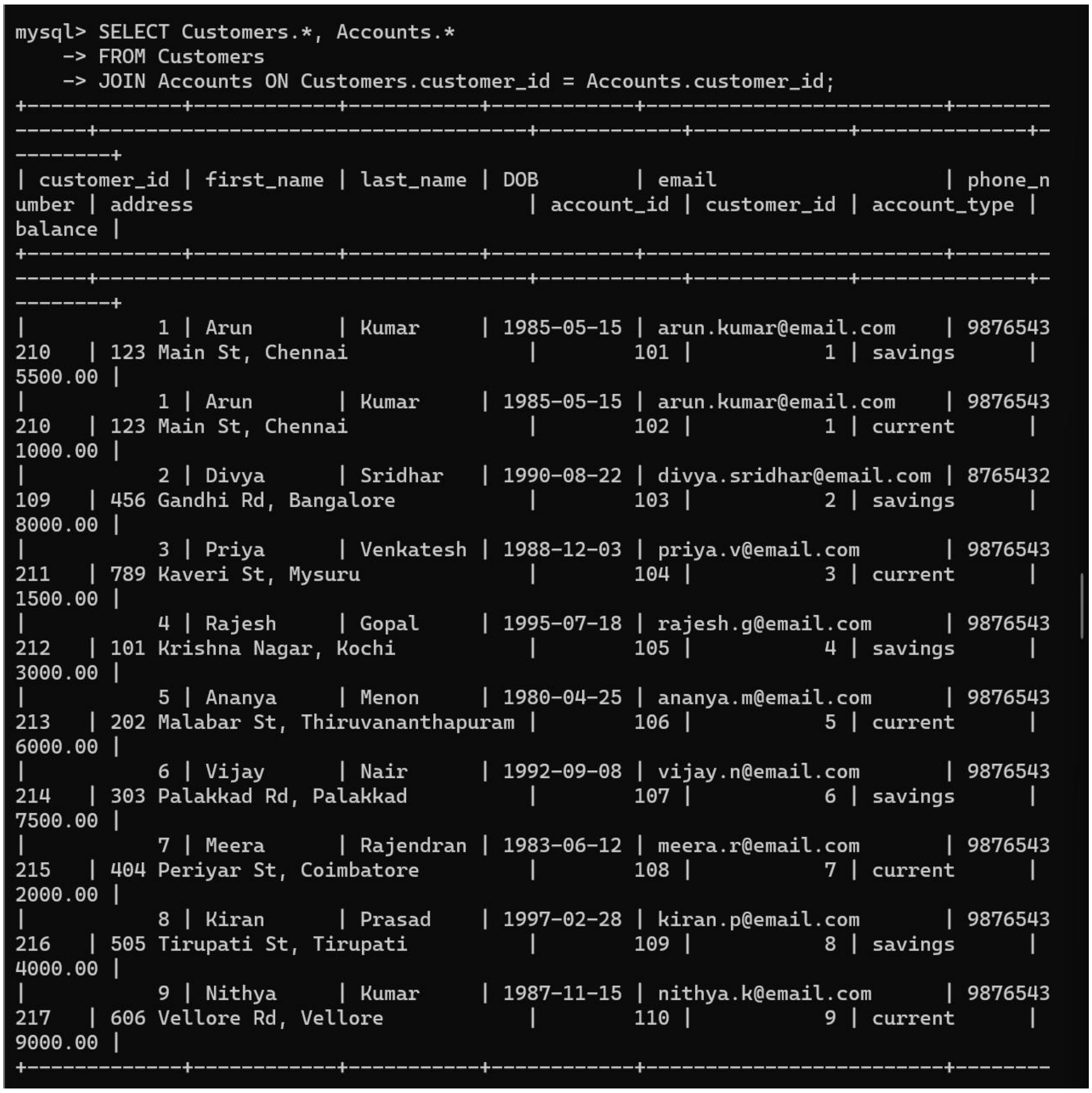
```
mysql> SELECT MIN(DOB) AS oldest_customer, MAX(DOB) AS newest_customer
    -> FROM Customers;
+-----------------+-----------------+
| oldest_customer | newest_customer |
+-----------------+-----------------+
| 1980-04-25      | 1997-02-28      |
+-----------------+-----------------+
1 row in set (0.00 sec)
```

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



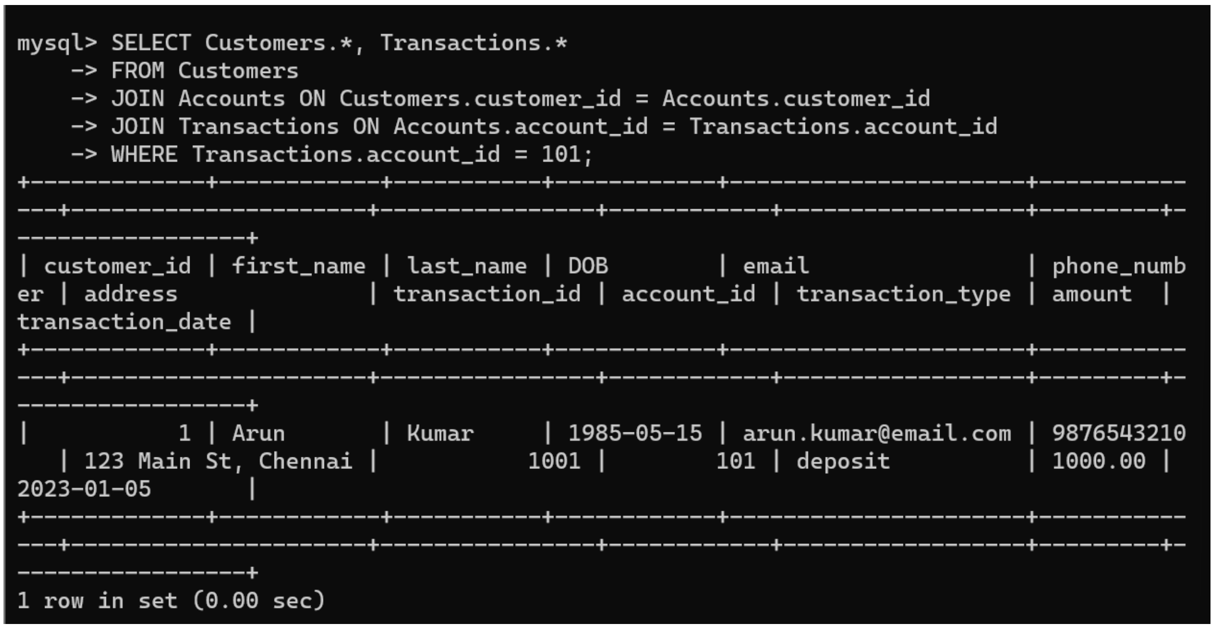
```
mysql> SELECT Transactions.*, Accounts.account_type
    -> FROM Transactions
    -> JOIN Accounts ON Transactions.account_id = Accounts.account_id;
+----------------+------------+------------------+---------+------------------+--------
------+
| transaction_id | account_id | transaction_type | amount  | transaction_date | account
_type |
+----------------+------------+------------------+---------+------------------+--------
------+
|           1001 |        101 | deposit          | 1000.00 | 2023-01-05       | savings
      |
|           1002 |        102 | withdrawal       |  500.00 | 2023-02-10       | current
      |
|           1003 |        103 | deposit          | 2000.00 | 2023-03-15       | savings
      |
|           1004 |        104 | deposit          |  500.00 | 2023-04-20       | current
      |
|           1005 |        105 | withdrawal       | 1000.00 | 2023-05-25       | savings
      |
|           1006 |        106 | deposit          | 1500.00 | 2023-06-30       | current
      |
|           1007 |        107 | withdrawal       | 2000.00 | 2023-07-05       | savings
      |
|           1008 |        108 | deposit          | 1000.00 | 2023-08-10       | current
      |
|           1009 |        109 | withdrawal       | 3000.00 | 2023-09-15       | savings
      |
|           1010 |        110 | deposit          | 2000.00 | 2023-10-20       | current
      |
+----------------+------------+------------------+---------+------------------+--------
------+
10 rows in set (0.00 sec)
```

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



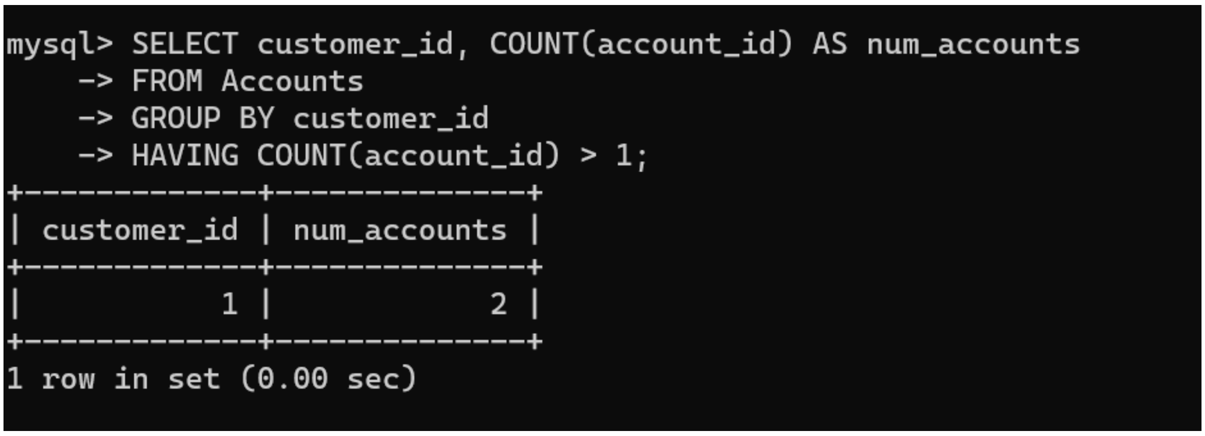
```
mysql> SELECT Customers.*, Accounts.*
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id;
+-------------+------------+-----------+------------+-------------------------+--------
------+--------------------------------+------------+-------------+--------------+-
--------+
| customer_id | first_name | last_name | DOB        | email                   | phone_n
umber | address                        | account_id | customer_id | account_type |
balance |
+-------------+------------+-----------+------------+-------------------------+--------
------+--------------------------------+------------+-------------+--------------+-
--------+
|           1 | Arun       | Kumar     | 1985-05-15 | arun.kumar@email.com    | 9876543
210   | 123 Main St, Chennai           |        101 |           1 | savings      |
5500.00 |
|           1 | Arun       | Kumar     | 1985-05-15 | arun.kumar@email.com    | 9876543
210   | 123 Main St, Chennai           |        102 |           1 | current      |
1000.00 |
|           2 | Divya      | Sridhar   | 1990-08-22 | divya.sridhar@email.com | 8765432
109   | 456 Gandhi Rd, Bangalore       |        103 |           2 | savings      |
8000.00 |
|           3 | Priya      | Venkatesh | 1988-12-03 | priya.v@email.com       | 9876543
211   | 789 Kaveri St, Mysuru          |        104 |           3 | current      |
1500.00 |
|           4 | Rajesh     | Gopal     | 1995-07-18 | rajesh.g@email.com      | 9876543
212   | 101 Krishna Nagar, Kochi       |        105 |           4 | savings      |
3000.00 |
|           5 | Ananya     | Menon     | 1980-04-25 | ananya.m@email.com      | 9876543
213   | 202 Malabar St, Thiruvananthapuram |    106 |           5 | current      |
6000.00 |
|           6 | Vijay      | Nair      | 1992-09-08 | vijay.n@email.com       | 9876543
214   | 303 Palakkad Rd, Palakkad      |        107 |           6 | savings      |
7500.00 |
|           7 | Meera      | Rajendran | 1983-06-12 | meera.r@email.com       | 9876543
215   | 404 Periyar St, Coimbatore     |        108 |           7 | current      |
2000.00 |
|           8 | Kiran      | Prasad    | 1997-02-28 | kiran.p@email.com       | 9876543
216   | 505 Tirupati St, Tirupati      |        109 |           8 | savings      |
4000.00 |
|           9 | Nithya     | Kumar     | 1987-11-15 | nithya.k@email.com      | 9876543
217   | 606 Vellore Rd, Vellore        |        110 |           9 | current      |
9000.00 |
+-------------+------------+-----------+------------+-------------------------+--------
```

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



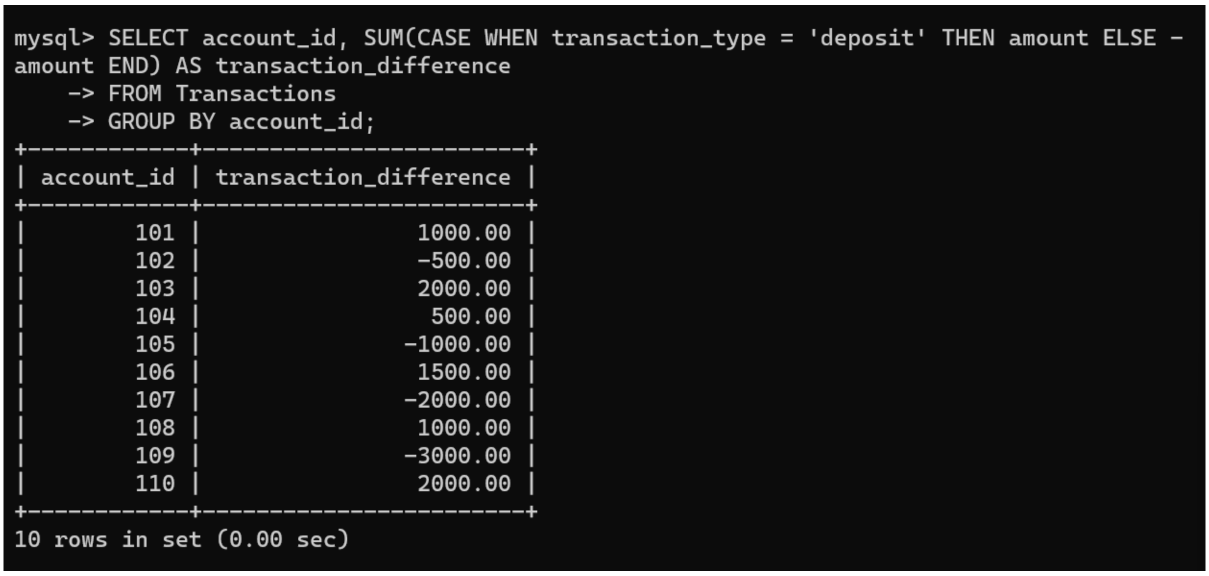
```
mysql> SELECT Customers.*, Transactions.*
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id
    -> JOIN Transactions ON Accounts.account_id = Transactions.account_id
    -> WHERE Transactions.account_id = 101;
+-------------+------------+-----------+------------+----------------------+-----------
---+----------------------+----------------+------------+------------------+---------+-
-----------------+
| customer_id | first_name | last_name | DOB        | email                | phone_numb
er | address              | transaction_id | account_id | transaction_type | amount  |
transaction_date |
+-------------+------------+-----------+------------+----------------------+-----------
---+----------------------+----------------+------------+------------------+---------+-
-----------------+
|           1 | Arun       | Kumar     | 1985-05-15 | arun.kumar@email.com | 9876543210
   | 123 Main St, Chennai |           1001 |        101 | deposit          | 1000.00 |
2023-01-05       |
+-------------+------------+-----------+------------+----------------------+-----------
---+----------------------+----------------+------------+------------------+---------+-
-----------------+
1 row in set (0.00 sec)
```

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

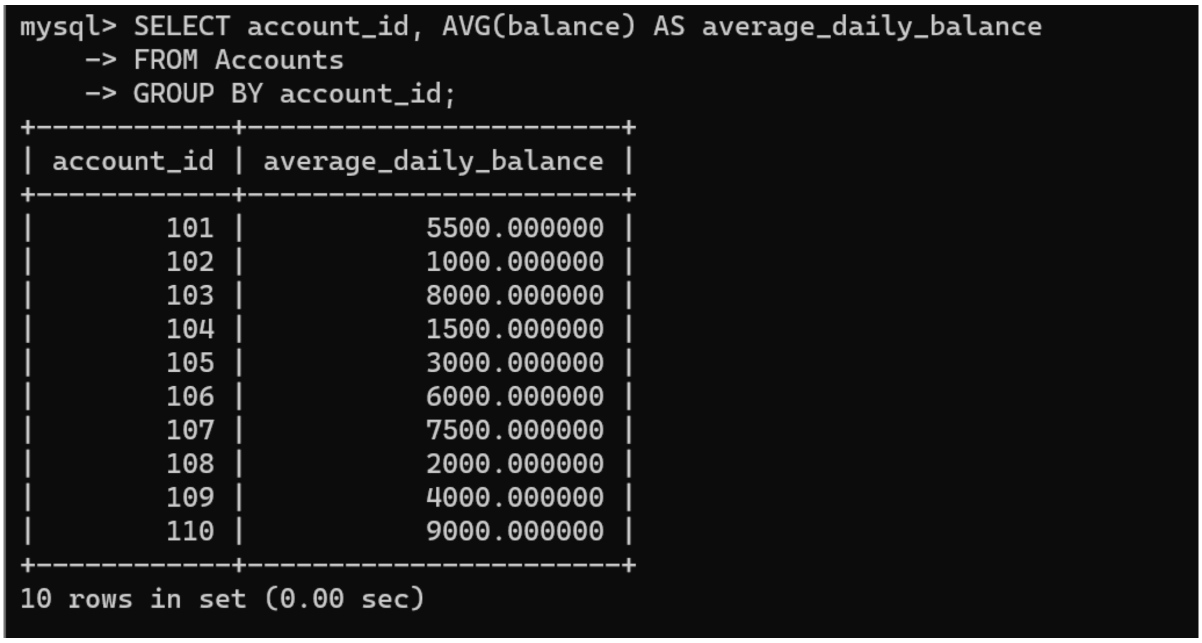


```
mysql> SELECT customer_id, COUNT(account_id) AS num_accounts
    -> FROM Accounts
    -> GROUP BY customer_id
    -> HAVING COUNT(account_id) > 1;
+-------------+--------------+
| customer_id | num_accounts |
+-------------+--------------+
|           1 |            2 |
+-------------+--------------+
1 row in set (0.00 sec)
```

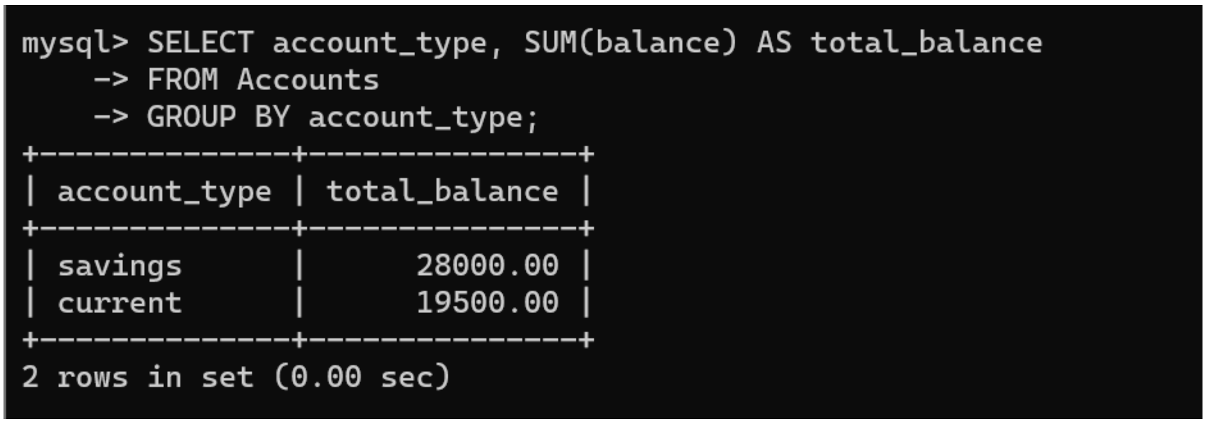
8. Write a SQL query to identify customers who have more than one account.


```
mysql> SELECT account_id, SUM(CASE WHEN transaction_type = 'deposit' THEN amount ELSE -
amount END) AS transaction_difference
    -> FROM Transactions
    -> GROUP BY account_id;
+------------+------------------------+
| account_id | transaction_difference |
+------------+------------------------+
|        101 |                1000.00 |
|        102 |                -500.00 |
|        103 |                2000.00 |
|        104 |                 500.00 |
|        105 |               -1000.00 |
|        106 |                1500.00 |
|        107 |               -2000.00 |
|        108 |                1000.00 |
|        109 |               -3000.00 |
|        110 |                2000.00 |
+------------+------------------------+
10 rows in set (0.00 sec)
```

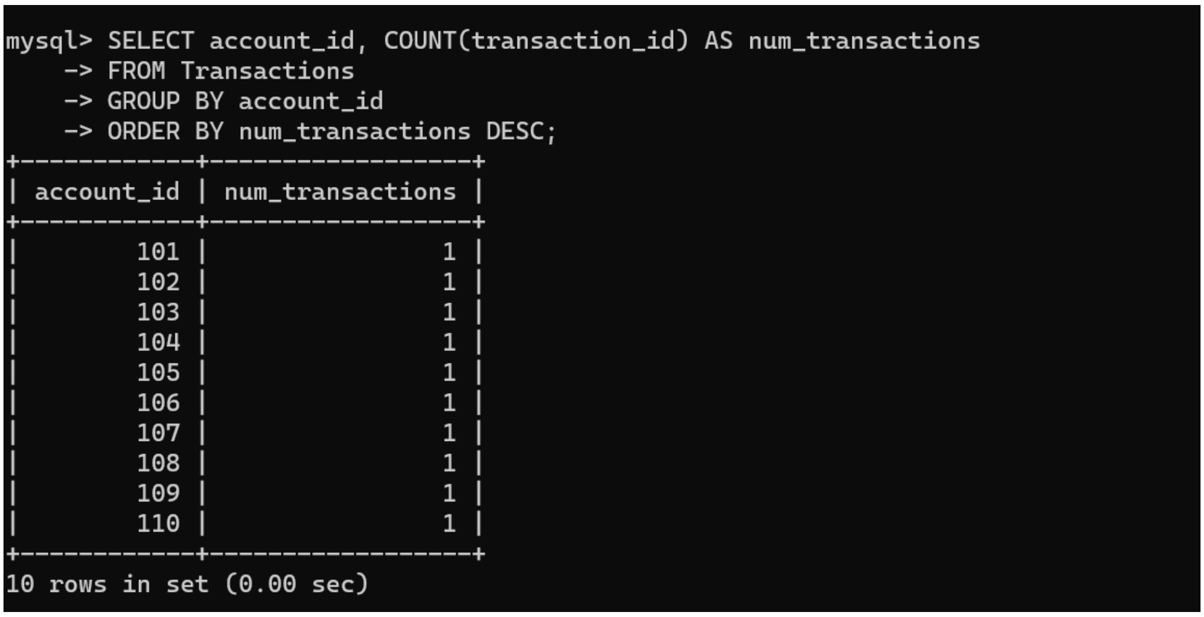
9. Write a SQL query to Calculate the average daily balance for each account over a specified period


```
mysql> SELECT account_id, AVG(balance) AS average_daily_balance
    -> FROM Accounts
    -> GROUP BY account_id;
+------------+-----------------------+
| account_id | average_daily_balance |
+------------+-----------------------+
|        101 |           5500.000000 |
|        102 |           1000.000000 |
|        103 |           8000.000000 |
|        104 |           1500.000000 |
|        105 |           3000.000000 |
|        106 |           6000.000000 |
|        107 |           7500.000000 |
|        108 |           2000.000000 |
|        109 |           4000.000000 |
|        110 |           9000.000000 |
+------------+-----------------------+
10 rows in set (0.00 sec)
```

10. Calculate the total balance for each account type.

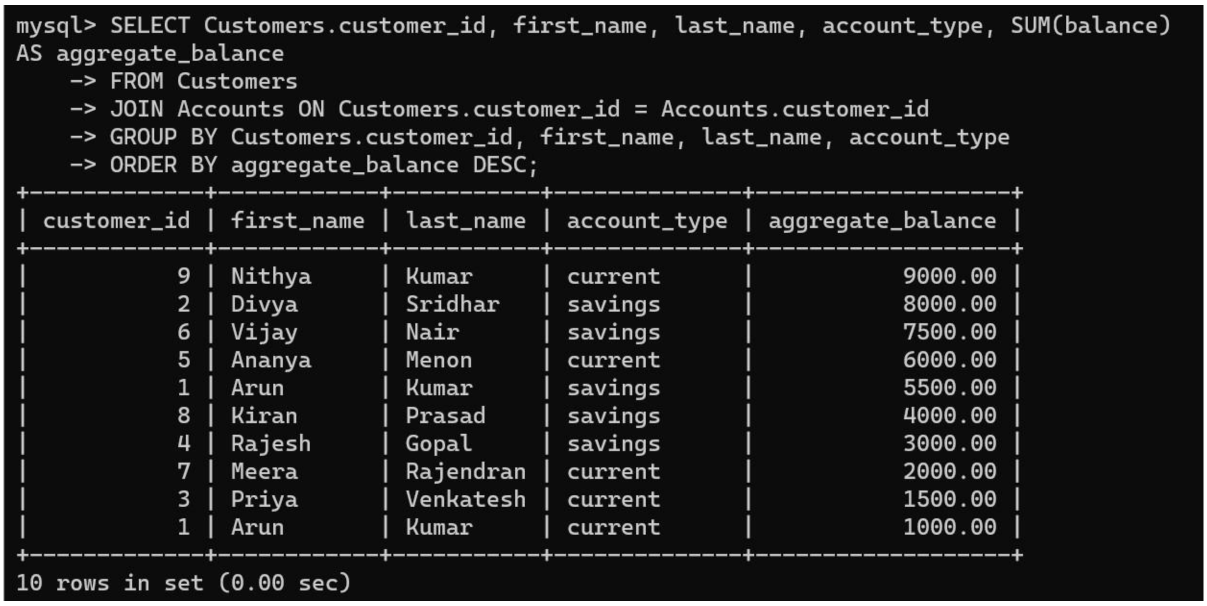

```
mysql> SELECT account_type, SUM(balance) AS total_balance
    -> FROM Accounts
    -> GROUP BY account_type;
+--------------+---------------+
| account_type | total_balance |
+--------------+---------------+
| savings      |      28000.00 |
| current      |      19500.00 |
+--------------+---------------+
2 rows in set (0.00 sec)
```

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



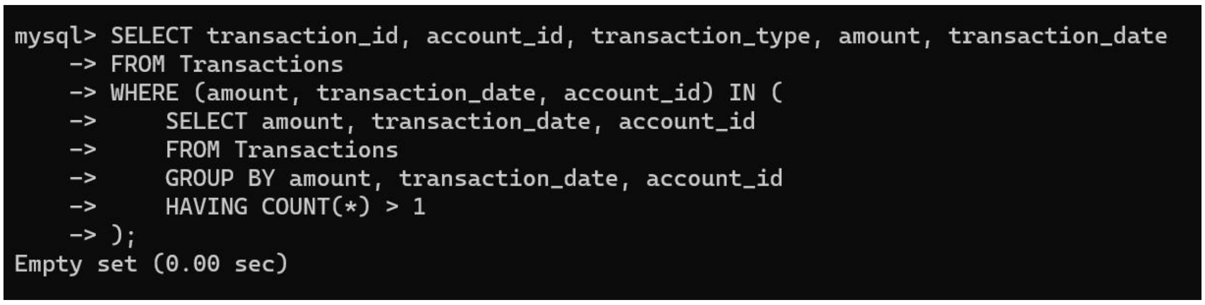
```
mysql> SELECT account_id, COUNT(transaction_id) AS num_transactions
    -> FROM Transactions
    -> GROUP BY account_id
    -> ORDER BY num_transactions DESC;
+------------+------------------+
| account_id | num_transactions |
+------------+------------------+
|        101 |                1 |
|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

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



```
mysql> SELECT Customers.customer_id, first_name, last_name, account_type, SUM(balance)
AS aggregate_balance
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id
    -> GROUP BY Customers.customer_id, first_name, last_name, account_type
    -> ORDER BY aggregate_balance DESC;
+-------------+------------+-----------+--------------+-------------------+
| customer_id | first_name | last_name | account_type | aggregate_balance |
+-------------+------------+-----------+--------------+-------------------+
|           9 | Nithya     | Kumar     | current      |           9000.00 |
|           2 | Divya      | Sridhar   | savings      |           8000.00 |
|           6 | Vijay      | Nair      | savings      |           7500.00 |
|           5 | Ananya     | Menon     | current      |           6000.00 |
|           1 | Arun       | Kumar     | savings      |           5500.00 |
|           8 | Kiran      | Prasad    | savings      |           4000.00 |
|           4 | Rajesh     | Gopal     | savings      |           3000.00 |
|           7 | Meera      | Rajendran | current      |           2000.00 |
|           3 | Priya      | Venkatesh | current      |           1500.00 |
|           1 | Arun       | Kumar     | current      |           1000.00 |
+-------------+------------+-----------+--------------+-------------------+
10 rows in set (0.00 sec)
```

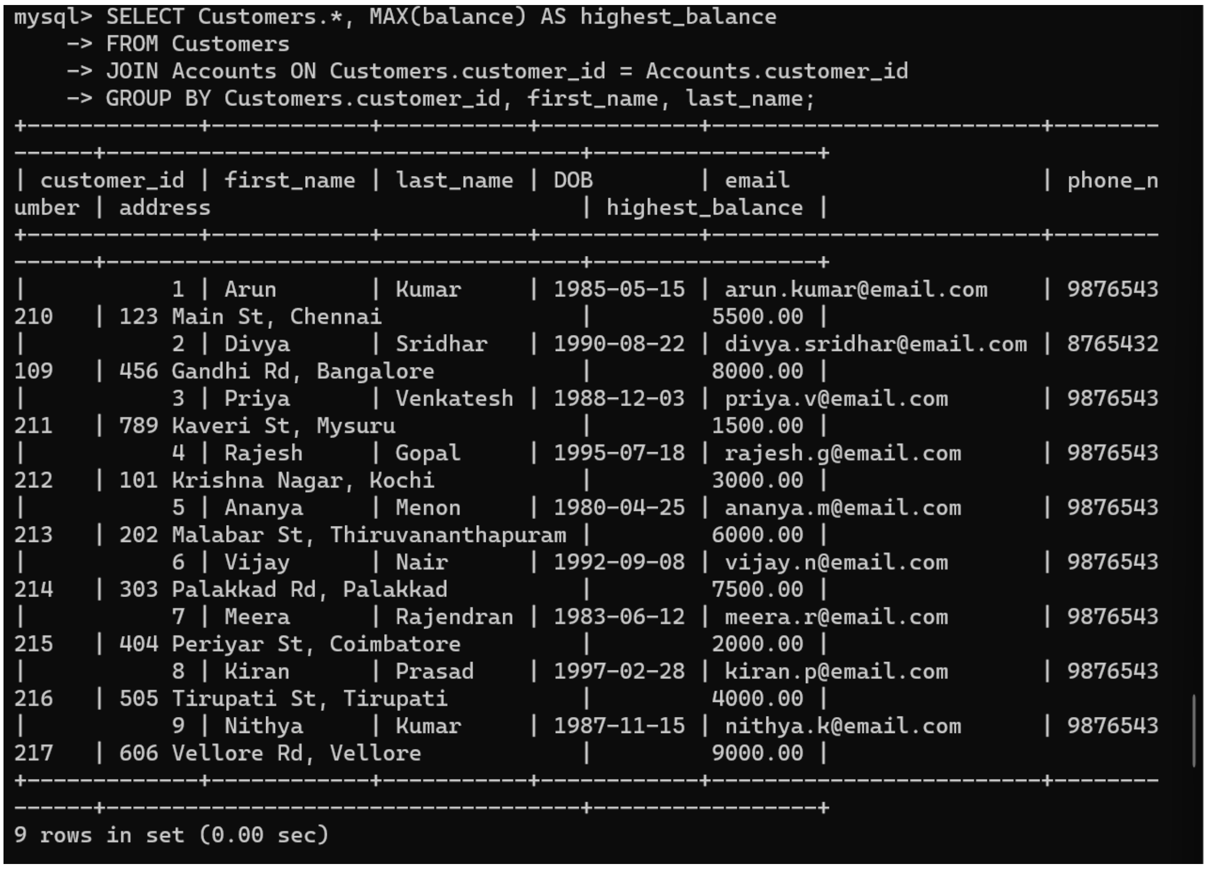
13. Identify and list duplicate transactions based on transaction amount, date, and account.



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

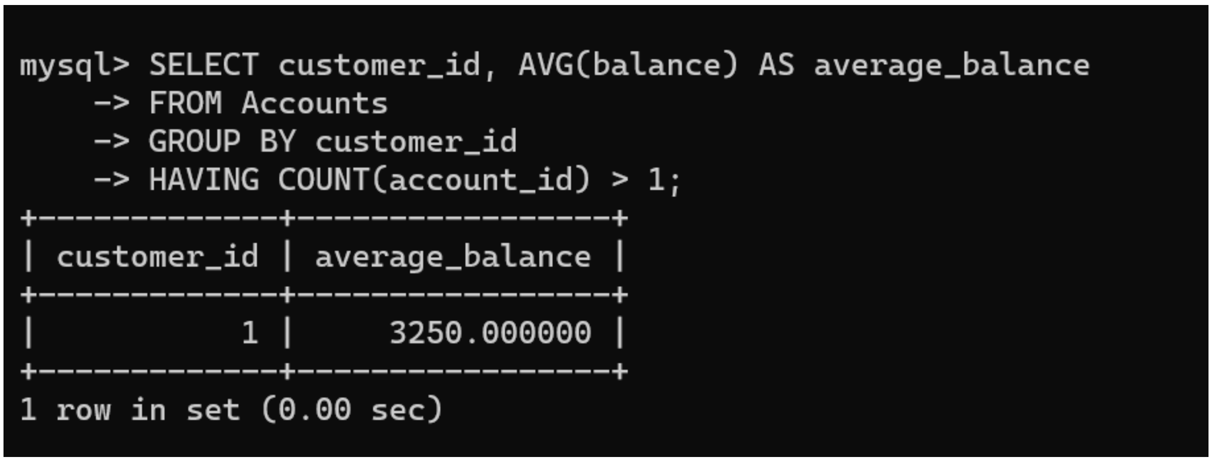
Task 4:

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



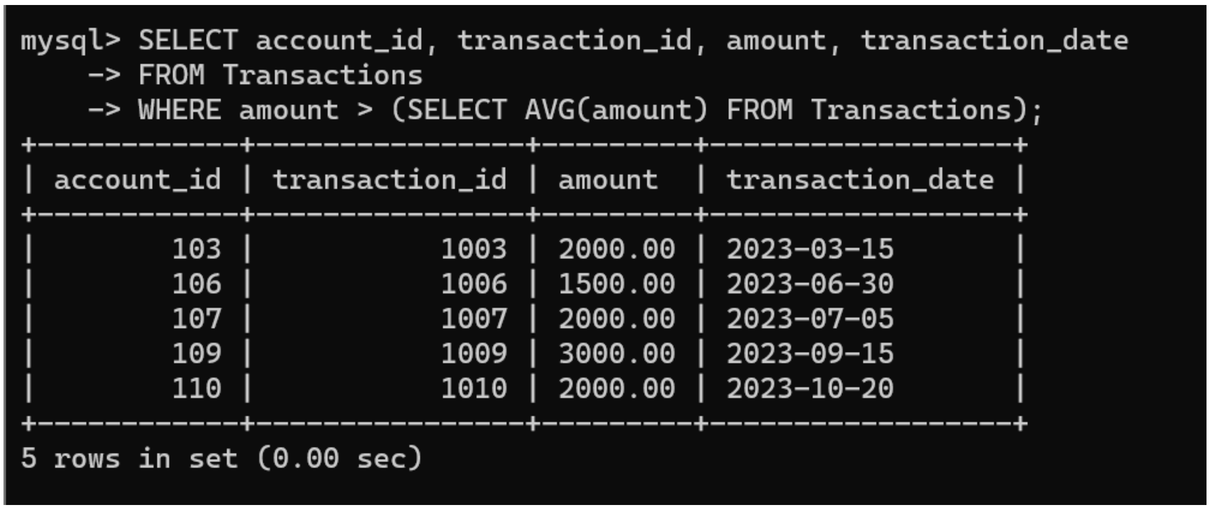
```
mysql> SELECT Customers.*, MAX(balance) AS highest_balance
    -> FROM Customers
    -> JOIN Accounts ON Customers.customer_id = Accounts.customer_id
    -> GROUP BY Customers.customer_id, first_name, last_name;
+-------------+------------+-----------+------------+-------------------------+--------
------+-------------------------------------+-----------------+
| customer_id | first_name | last_name | DOB        | email                   | phone_n
umber | address                             | highest_balance |
+-------------+------------+-----------+------------+-------------------------+--------
------+-------------------------------------+-----------------+
|           1 | Arun       | Kumar     | 1985-05-15 | arun.kumar@email.com    | 9876543
210   | 123 Main St, Chennai                |         5500.00 |
|           2 | Divya      | Sridhar   | 1990-08-22 | divya.sridhar@email.com | 8765432
109   | 456 Gandhi Rd, Bangalore            |         8000.00 |
|           3 | Priya      | Venkatesh | 1988-12-03 | priya.v@email.com       | 9876543
211   | 789 Kaveri St, Mysuru               |         1500.00 |
|           4 | Rajesh     | Gopal     | 1995-07-18 | rajesh.g@email.com      | 9876543
212   | 101 Krishna Nagar, Kochi            |         3000.00 |
|           5 | Ananya     | Menon     | 1980-04-25 | ananya.m@email.com      | 9876543
213   | 202 Malabar St, Thiruvananthapuram |         6000.00 |
|           6 | Vijay      | Nair      | 1992-09-08 | vijay.n@email.com       | 9876543
214   | 303 Palakkad Rd, Palakkad           |         7500.00 |
|           7 | Meera      | Rajendran | 1983-06-12 | meera.r@email.com       | 9876543
215   | 404 Periyar St, Coimbatore          |         2000.00 |
|           8 | Kiran      | Prasad    | 1997-02-28 | kiran.p@email.com       | 9876543
216   | 505 Tirupati St, Tirupati           |         4000.00 |
|           9 | Nithya     | Kumar     | 1987-11-15 | nithya.k@email.com      | 9876543
217   | 606 Vellore Rd, Vellore             |         9000.00 |
+-------------+------------+-----------+------------+-------------------------+--------
------+-------------------------------------+-----------------+
9 rows in set (0.00 sec)
```

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



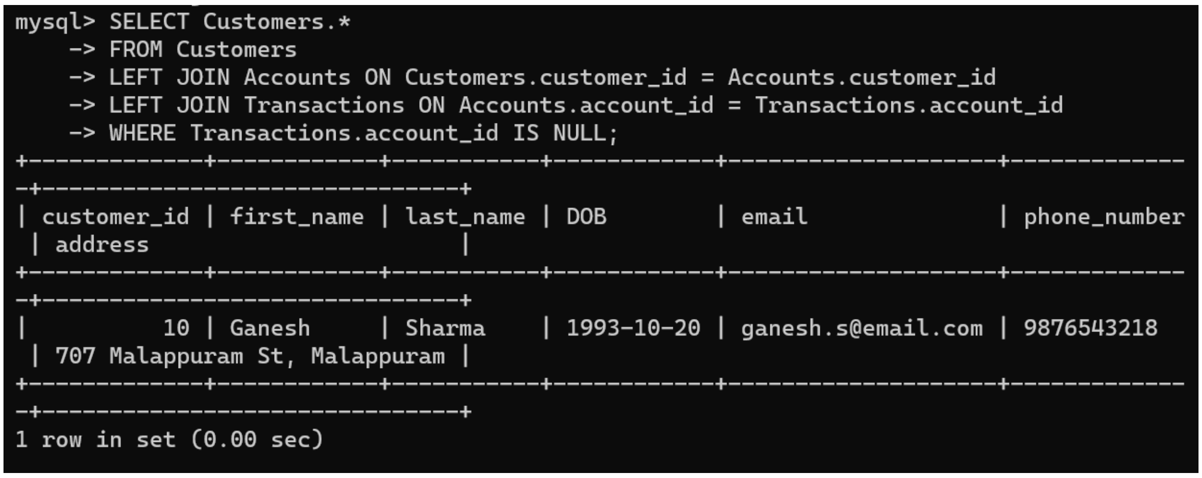
```
mysql> SELECT customer_id, AVG(balance) AS average_balance
    -> FROM Accounts
    -> GROUP BY customer_id
    -> HAVING COUNT(account_id) > 1;
+-------------+-----------------+
| customer_id | average_balance |
+-------------+-----------------+
|           1 |     3250.000000 |
+-------------+-----------------+
1 row in set (0.00 sec)
```

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



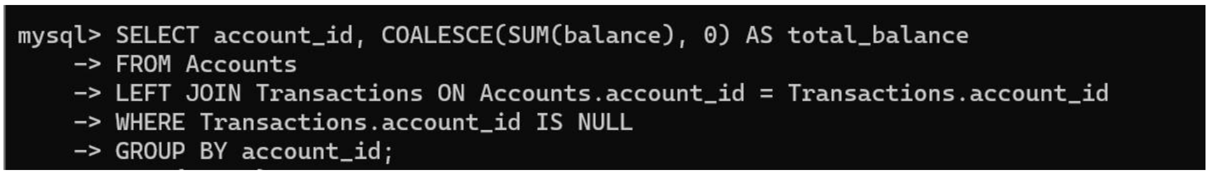
```
mysql> SELECT account_id, transaction_id, amount, transaction_date
    -> FROM Transactions
    -> WHERE amount > (SELECT AVG(amount) FROM Transactions);
+------------+----------------+---------+------------------+
| account_id | transaction_id | amount  | transaction_date |
+------------+----------------+---------+------------------+
|        103 |           1003 | 2000.00 | 2023-03-15       |
|        106 |           1006 | 1500.00 | 2023-06-30       |
|        107 |           1007 | 2000.00 | 2023-07-05       |
|        109 |           1009 | 3000.00 | 2023-09-15       |
|        110 |           1010 | 2000.00 | 2023-10-20       |
+------------+----------------+---------+------------------+
5 rows in set (0.00 sec)
```

5. Identify customers who have no recorded transactions.



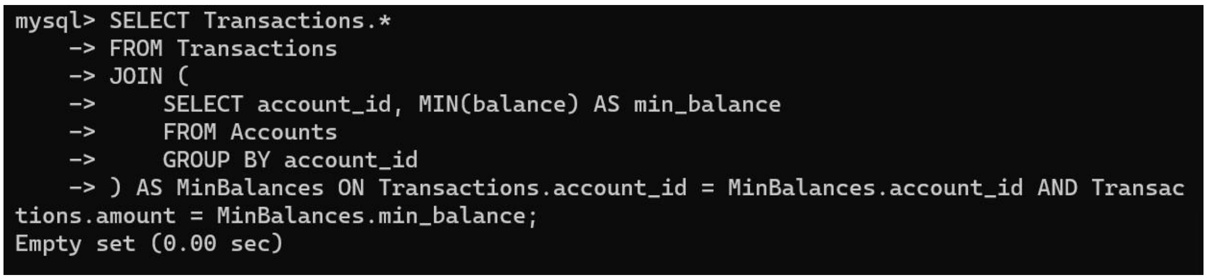
```
mysql> SELECT Customers.*
    -> FROM Customers
    -> LEFT JOIN Accounts ON Customers.customer_id = Accounts.customer_id
    -> LEFT JOIN Transactions ON Accounts.account_id = Transactions.account_id
    -> WHERE Transactions.account_id IS NULL;
+-------------+------------+-----------+------------+--------------------+-------------
-+------------------------------+
| customer_id | first_name | last_name | DOB        | email              | phone_number
 | address                      |
+-------------+------------+-----------+------------+--------------------+-------------
-+------------------------------+
|          10 | Ganesh     | Sharma    | 1993-10-20 | ganesh.s@email.com | 9876543218
 | 707 Malappuram St, Malappuram |
+-------------+------------+-----------+------------+--------------------+-------------
-+------------------------------+
1 row in set (0.00 sec)
```

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



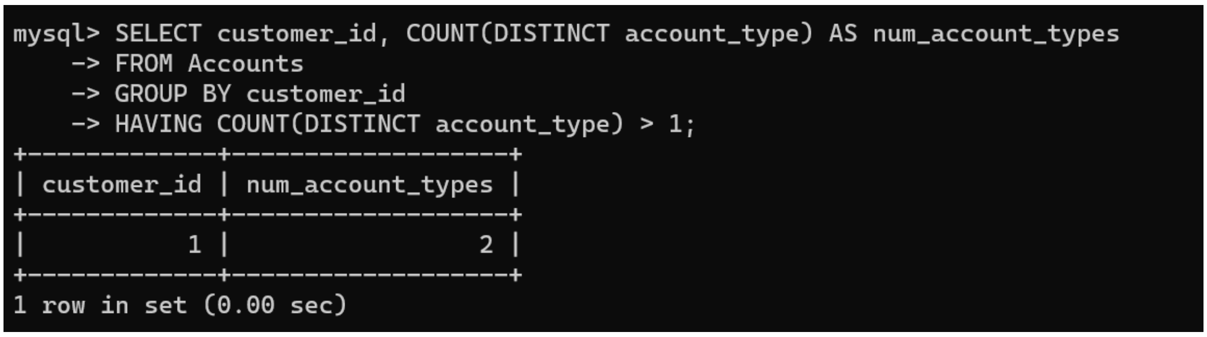
```
mysql> SELECT account_id, COALESCE(SUM(balance), 0) AS total_balance
    -> FROM Accounts
    -> LEFT JOIN Transactions ON Accounts.account_id = Transactions.account_id
    -> WHERE Transactions.account_id IS NULL
    -> GROUP BY account_id;
```

7. Retrieve transactions for accounts with the lowest balance.



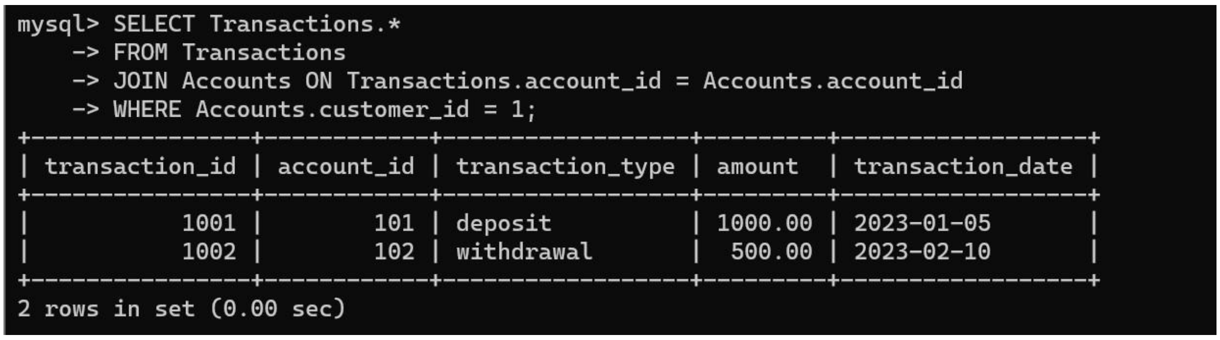
```
mysql> SELECT Transactions.*
    -> FROM Transactions
    -> JOIN (
    ->     SELECT account_id, MIN(balance) AS min_balance
    ->     FROM Accounts
    ->     GROUP BY account_id
    -> ) AS MinBalances ON Transactions.account_id = MinBalances.account_id AND Transac
tions.amount = MinBalances.min_balance;
Empty set (0.00 sec)
```

8. Identify customers who have accounts of multiple types.



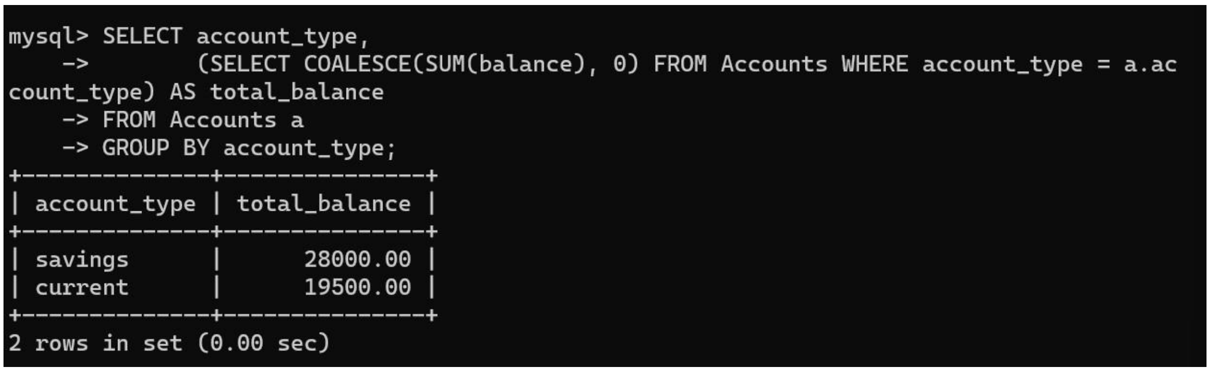
```
mysql> SELECT customer_id, COUNT(DISTINCT account_type) AS num_account_types
    -> FROM Accounts
    -> GROUP BY customer_id
    -> HAVING COUNT(DISTINCT account_type) > 1;
+-------------+-------------------+
| customer_id | num_account_types |
+-------------+-------------------+
|           1 |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

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



```
mysql> SELECT Transactions.*
    -> 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 |
+----------------+------------+------------------+---------+------------------+
|           1001 |        101 | deposit          | 1000.00 | 2023-01-05       |
|           1002 |        102 | withdrawal       |  500.00 | 2023-02-10       |
+----------------+------------+------------------+---------+------------------+
2 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 COALESCE(SUM(balance), 0) FROM Accounts WHERE account_type = a.ac
count_type) AS total_balance
    -> FROM Accounts a
    -> GROUP BY account_type;
+--------------+---------------+
| account_type | total_balance |
+--------------+---------------+
| savings      |      28000.00 |
| current      |      19500.00 |
+--------------+---------------+
2 rows in set (0.00 sec)
```