ASSIGNMENT TITLE:

HMBank : MySQL ASSIGNMENT 3

SUBMITED BY:

BALAKUMARAN P

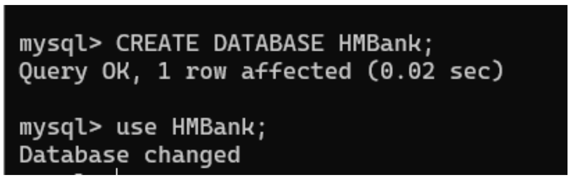
DATE OF SUBMISSION:

22/01/2024

HMBank : MySQL ASSIGNMENT 3

Tasks 1: Database Design:

1. Create the database named "HMBank"



```
mysql> CREATE DATABASE HMBank;
Query OK, 1 row affected (0.02 sec)

mysql> use HMBank;
Database changed
```

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

provided schema.

Customers Table:

CustomerID: Unique identifier for each customer.

FirstName: First name of the customer.

LastName: Last name of the customer.

Email: Email address of the customer.

City: City where customer lives.

Accounts Table:

AccountID: Unique identifier for each account.

CustomerID: Foreign key linking to the Customers table, indicating the customer associated with the account.

AccountType: Type of the account.

Balance: Current balance in the account.

Transactions Table:

TransactionID: Unique identifier for each transaction.

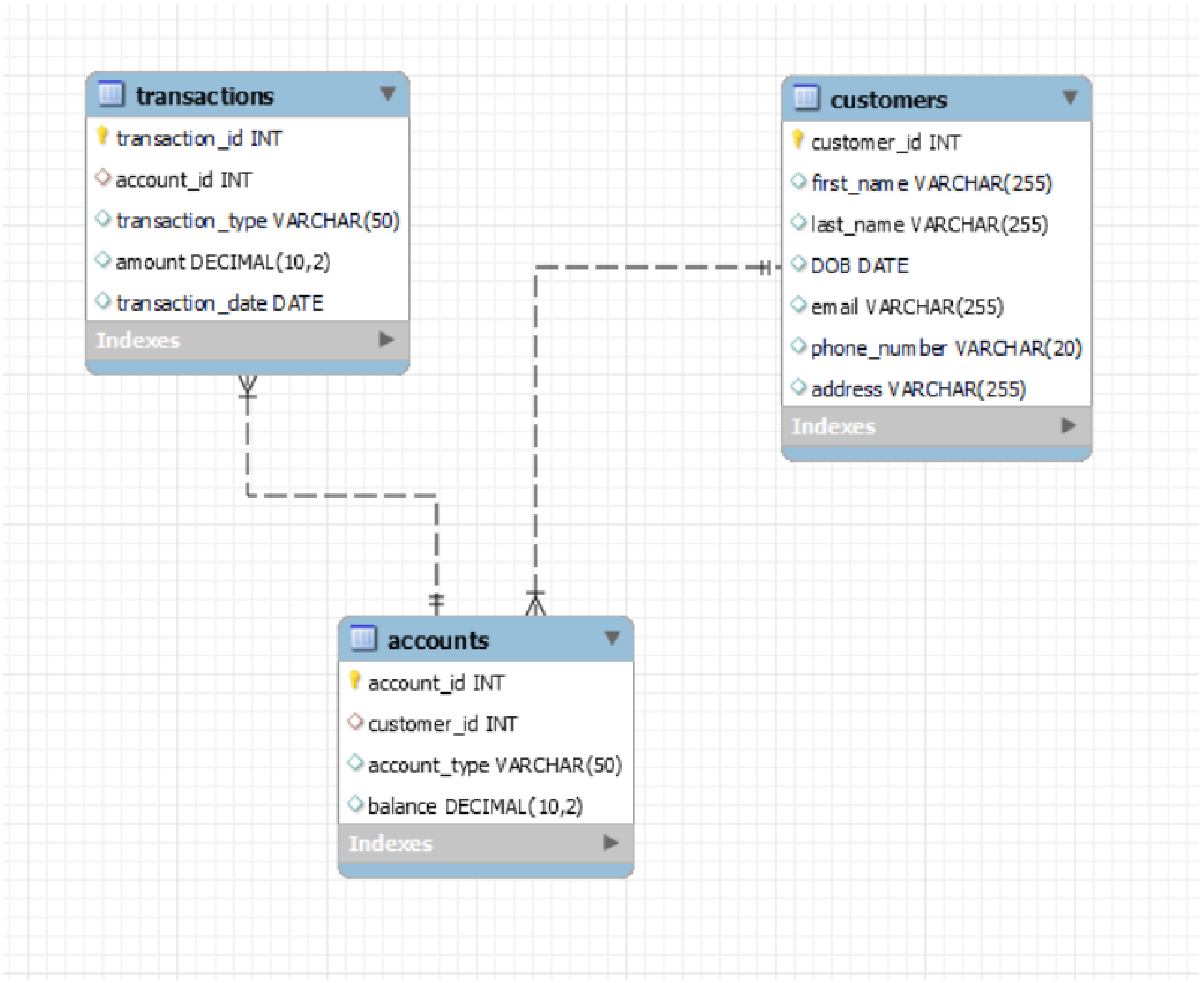
AccountID: Foreign key linking to the Accounts table, indicating the account associated with the transaction.

TransactionDate: Date when the transaction occurred.

Amount: Amount involved in the transaction.

Description: Description for the transaction.

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



5. Create appropriate Primary Key and Foreign Key constraints for referential integrity.

Customers Table:

CustomerID (Primary Key): Unique identifier for each customer.

Accounts Table:

AccountID (Pimary Key): Unique identifier for each account.

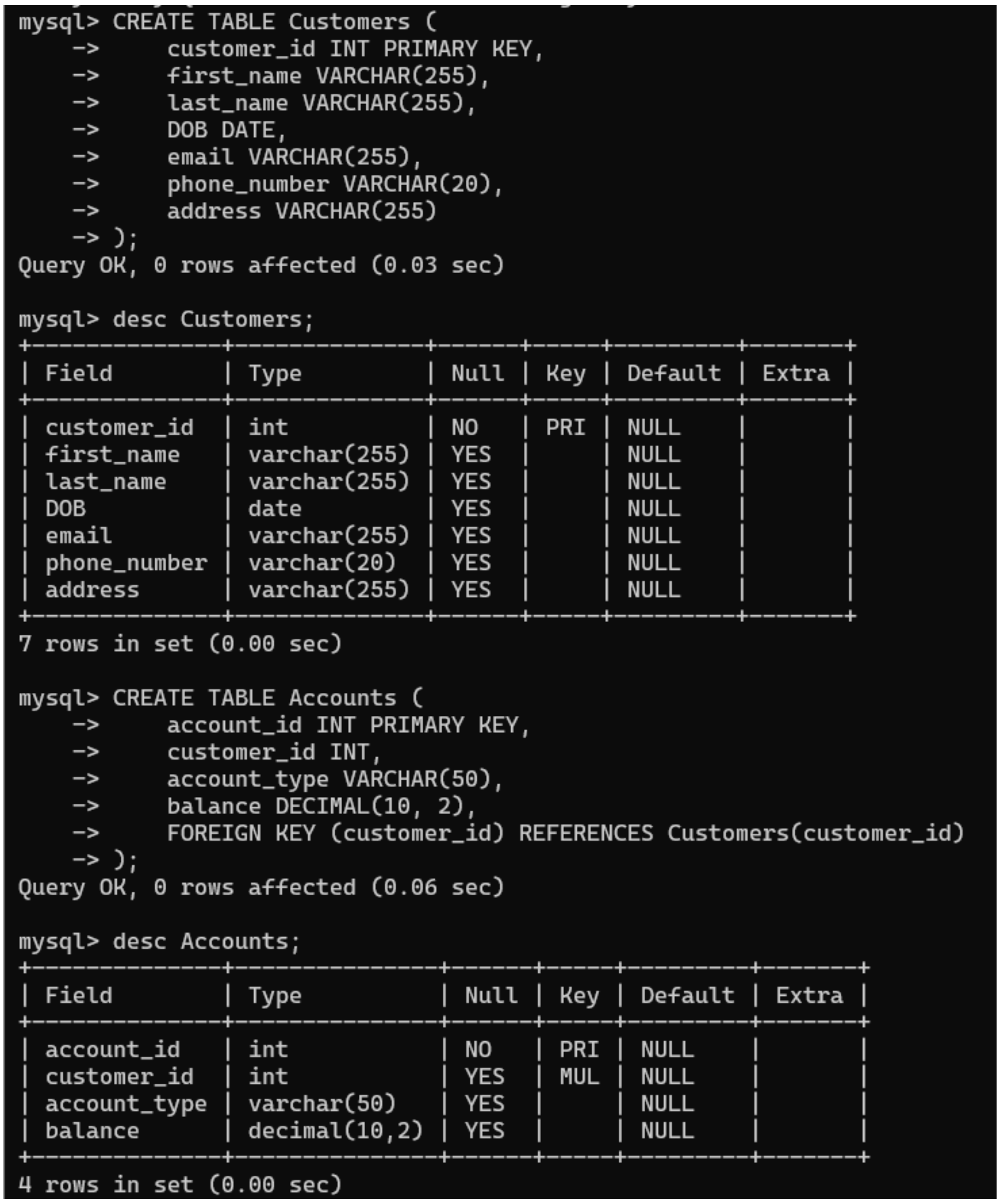
CustomerID (Foreign Key): Foreign key linking to the Customers table, indicating the customer associated with the account.

Transactions Table:

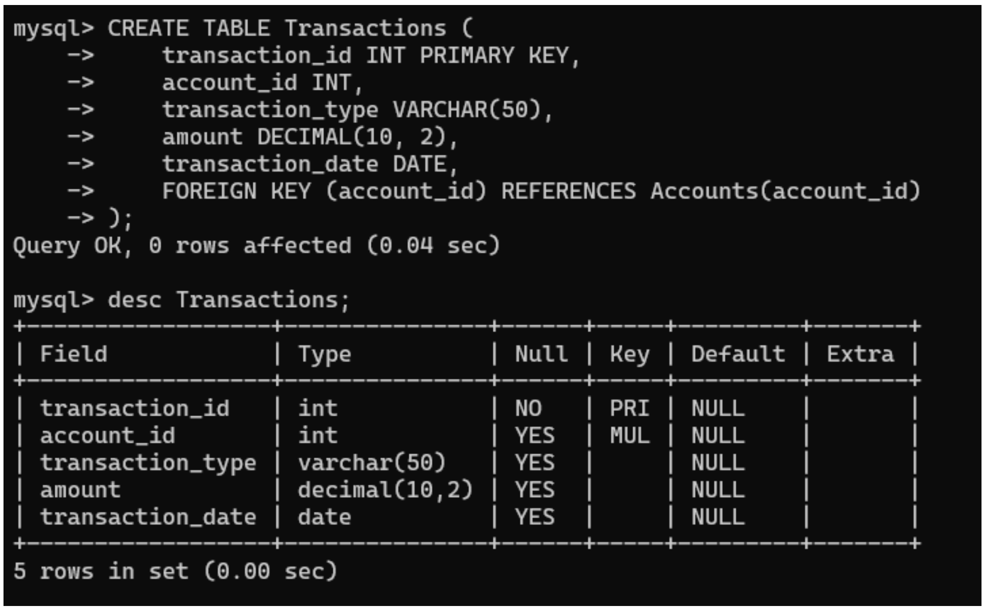
TransactionID (Primary Key): Unique identifier for each transaction.

AccountID (Foreign Key): Foreign key linking to the Accounts table, indicating the account associated with the transaction.

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



```
mysql> CREATE TABLE Customers (
    ->     customer_id INT PRIMARY KEY,
    ->     first_name VARCHAR(255),
    ->     last_name VARCHAR(255),
    ->     DOB DATE,
    ->     email VARCHAR(255),
    ->     phone_number VARCHAR(20),
    ->     address VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> desc Customers;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| customer_id  | int          | NO   | PRI | NULL    |       |
| first_name   | varchar(255) | YES  |     | NULL    |       |
| last_name    | varchar(255) | YES  |     | NULL    |       |
| DOB          | date         | YES  |     | NULL    |       |
| email        | varchar(255) | YES  |     | NULL    |       |
| phone_number | varchar(20)  | YES  |     | NULL    |       |
| address      | varchar(255) | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
7 rows in set (0.00 sec)

mysql> CREATE TABLE Accounts (
    ->     account_id INT PRIMARY KEY,
    ->     customer_id INT,
    ->     account_type VARCHAR(50),
    ->     balance DECIMAL(10, 2),
    ->     FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
    -> );
Query OK, 0 rows affected (0.06 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(50)   | YES  |     | NULL    |       |
| balance      | decimal(10,2) | YES  |     | NULL    |       |
+--------------+---------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```



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

mysql> desc Transactions;
+------------------+---------------+------+-----+---------+-------+
| Field            | Type          | Null | Key | Default | Extra |
+------------------+---------------+------+-----+---------+-------+
| transaction_id   | int           | NO   | PRI | NULL    |       |
| account_id       | int           | YES  | MUL | NULL    |       |
| transaction_type | varchar(50)   | YES  |     | NULL    |       |
| amount           | decimal(10,2) | YES  |     | NULL    |       |
| transaction_date | date          | YES  |     | NULL    |       |
+------------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

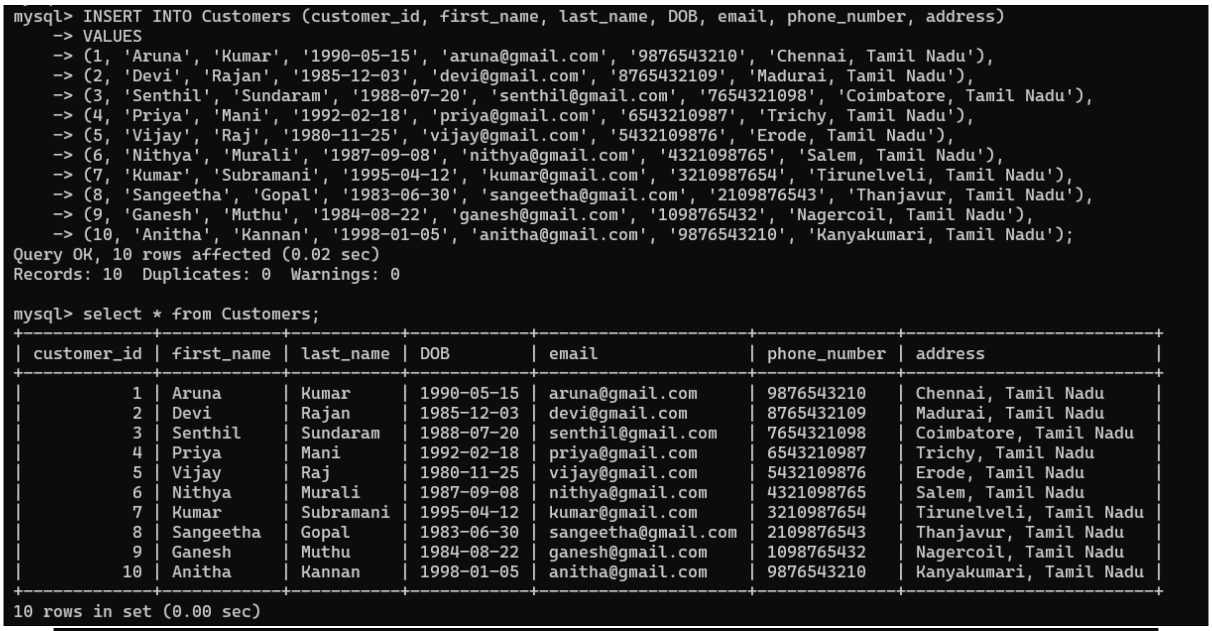
Tasks 2: Select, Where, Between, AND, LIKE:

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

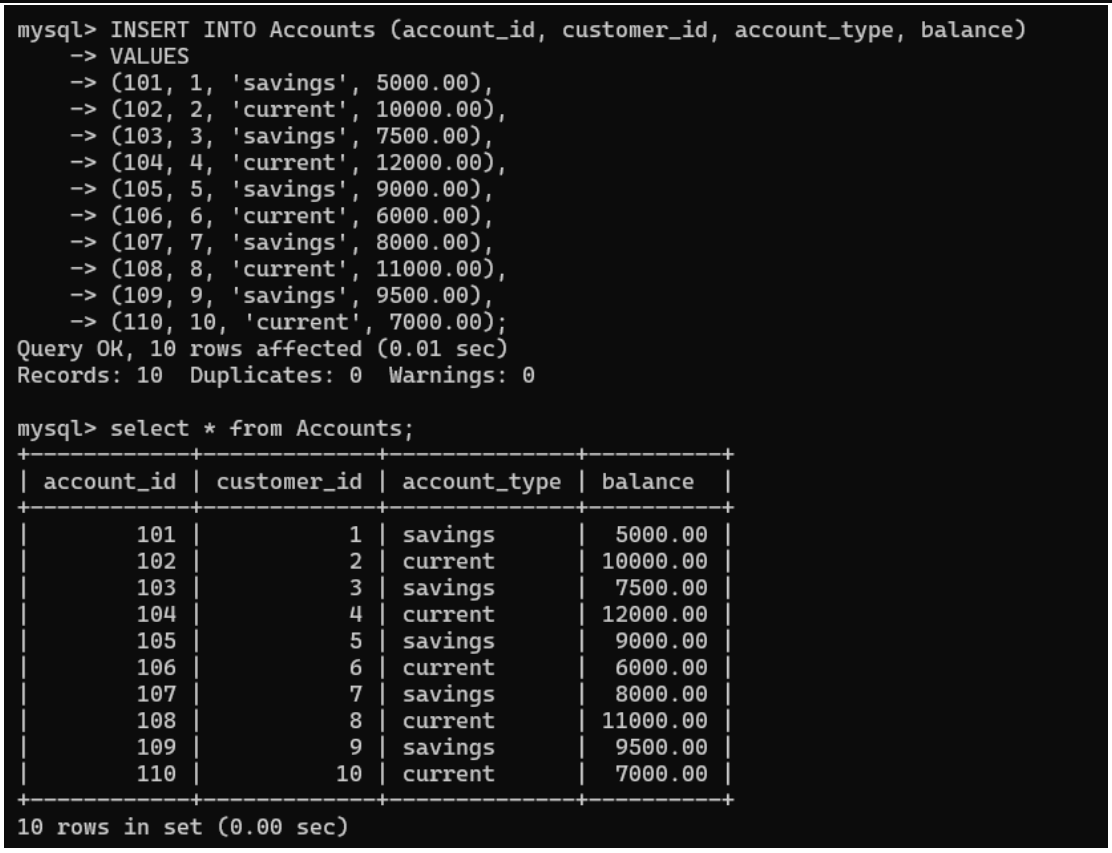
• Customers

• Accounts

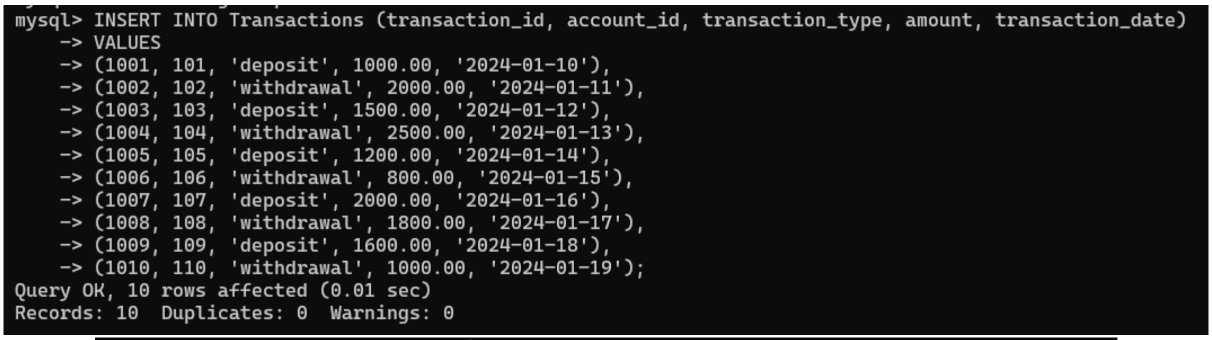
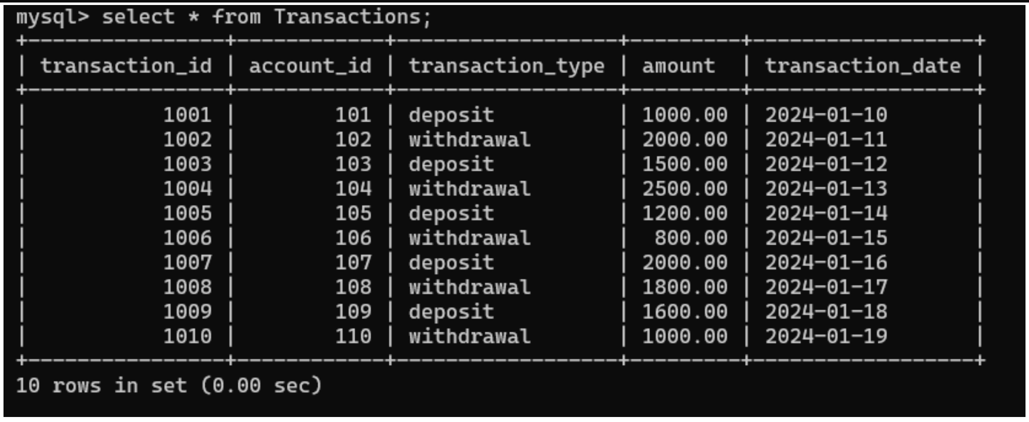
• Transactions



```
mysql> INSERT INTO Customers (customer_id, first_name, last_name, DOB, email, phone_number, address)
    -> VALUES
    -> (1, 'Aruna', 'Kumar', '1990-05-15', 'aruna@gmail.com', '9876543210', 'Chennai, Tamil Nadu'),
    -> (2, 'Devi', 'Rajan', '1985-12-03', 'devi@gmail.com', '8765432109', 'Madurai, Tamil Nadu'),
    -> (3, 'Senthil', 'Sundaram', '1988-07-20', 'senthil@gmail.com', '7654321098', 'Coimbatore, Tamil Nadu'),
    -> (4, 'Priya', 'Mani', '1992-02-18', 'priya@gmail.com', '6543210987', 'Trichy, Tamil Nadu'),
    -> (5, 'Vijay', 'Raj', '1980-11-25', 'vijay@gmail.com', '5432109876', 'Erode, Tamil Nadu'),
    -> (6, 'Nithya', 'Murali', '1987-09-08', 'nithya@gmail.com', '4321098765', 'Salem, Tamil Nadu'),
    -> (7, 'Kumar', 'Subramani', '1995-04-12', 'kumar@gmail.com', '3210987654', 'Tirunelveli, Tamil Nadu'),
    -> (8, 'Sangeetha', 'Gopal', '1983-06-30', 'sangeetha@gmail.com', '2109876543', 'Thanjavur, Tamil Nadu'),
    -> (9, 'Ganesh', 'Muthu', '1984-08-22', 'ganesh@gmail.com', '1098765432', 'Nagercoil, Tamil Nadu'),
    -> (10, 'Anitha', 'Kannan', '1998-01-05', 'anitha@gmail.com', '9876543210', 'Kanyakumari, Tamil Nadu');
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from Customers;
+-------------+------------+-----------+------------+---------------------+--------------+-------------------------+
| customer_id | first_name | last_name | DOB        | email               | phone_number | address                 |
+-------------+------------+-----------+------------+---------------------+--------------+-------------------------+
|           1 | Aruna      | Kumar     | 1990-05-15 | aruna@gmail.com     | 9876543210   | Chennai, Tamil Nadu     |
|           2 | Devi       | Rajan     | 1985-12-03 | devi@gmail.com      | 8765432109   | Madurai, Tamil Nadu     |
|           3 | Senthil    | Sundaram  | 1988-07-20 | senthil@gmail.com   | 7654321098   | Coimbatore, Tamil Nadu  |
|           4 | Priya      | Mani      | 1992-02-18 | priya@gmail.com     | 6543210987   | Trichy, Tamil Nadu      |
|           5 | Vijay      | Raj       | 1980-11-25 | vijay@gmail.com     | 5432109876   | Erode, Tamil Nadu       |
|           6 | Nithya     | Murali    | 1987-09-08 | nithya@gmail.com    | 4321098765   | Salem, Tamil Nadu       |
|           7 | Kumar      | Subramani | 1995-04-12 | kumar@gmail.com     | 3210987654   | Tirunelveli, Tamil Nadu |
|           8 | Sangeetha  | Gopal     | 1983-06-30 | sangeetha@gmail.com | 2109876543   | Thanjavur, Tamil Nadu   |
|           9 | Ganesh     | Muthu     | 1984-08-22 | ganesh@gmail.com    | 1098765432   | Nagercoil, Tamil Nadu   |
|          10 | Anitha     | Kannan    | 1998-01-05 | anitha@gmail.com    | 9876543210   | Kanyakumari, Tamil Nadu |
+-------------+------------+-----------+------------+---------------------+--------------+-------------------------+
10 rows in set (0.00 sec)
```

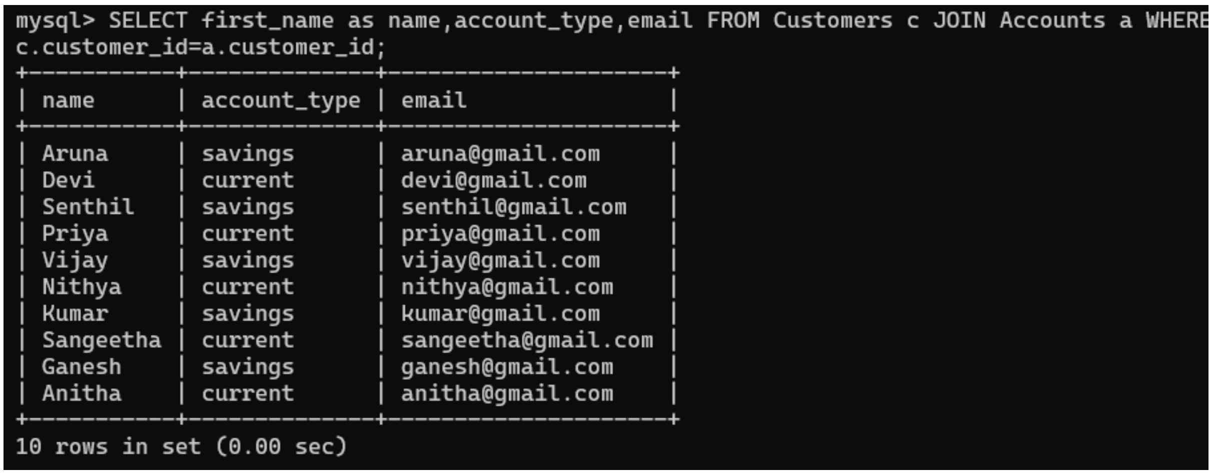


```
mysql> INSERT INTO Accounts (account_id, customer_id, account_type, balance)
    -> VALUES
    -> (101, 1, 'savings', 5000.00),
    -> (102, 2, 'current', 10000.00),
    -> (103, 3, 'savings', 7500.00),
    -> (104, 4, 'current', 12000.00),
    -> (105, 5, 'savings', 9000.00),
    -> (106, 6, 'current', 6000.00),
    -> (107, 7, 'savings', 8000.00),
    -> (108, 8, 'current', 11000.00),
    -> (109, 9, 'savings', 9500.00),
    -> (110, 10, 'current', 7000.00);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from Accounts;
+------------+-------------+--------------+----------+
| account_id | customer_id | account_type | balance  |
+------------+-------------+--------------+----------+
|        101 |           1 | savings      |  5000.00 |
|        102 |           2 | current      | 10000.00 |
|        103 |           3 | savings      |  7500.00 |
|        104 |           4 | current      | 12000.00 |
|        105 |           5 | savings      |  9000.00 |
|        106 |           6 | current      |  6000.00 |
|        107 |           7 | savings      |  8000.00 |
|        108 |           8 | current      | 11000.00 |
|        109 |           9 | savings      |  9500.00 |
|        110 |          10 | current      |  7000.00 |
+------------+-------------+--------------+----------+
10 rows in set (0.00 sec)
```

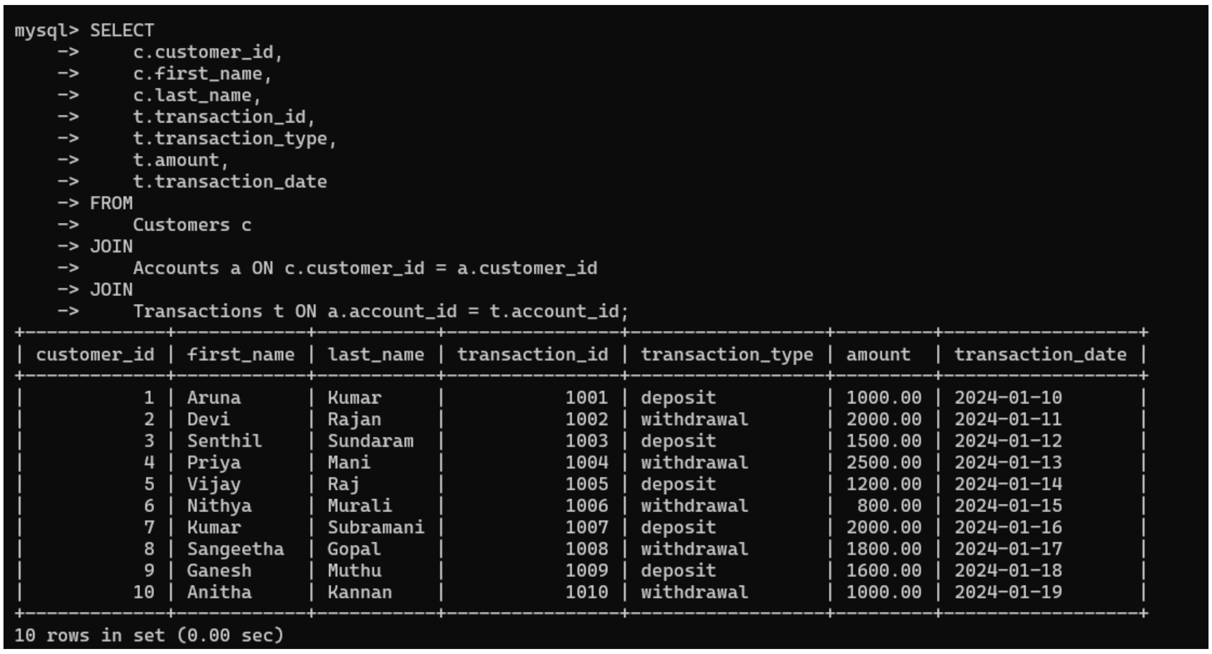

```
mysql> INSERT INTO Transactions (transaction_id, account_id, transaction_type, amount, transaction_date)
    -> VALUES
    -> (1001, 101, 'deposit', 1000.00, '2024-01-10'),
    -> (1002, 102, 'withdrawal', 2000.00, '2024-01-11'),
    -> (1003, 103, 'deposit', 1500.00, '2024-01-12'),
    -> (1004, 104, 'withdrawal', 2500.00, '2024-01-13'),
    -> (1005, 105, 'deposit', 1200.00, '2024-01-14'),
    -> (1006, 106, 'withdrawal', 800.00, '2024-01-15'),
    -> (1007, 107, 'deposit', 2000.00, '2024-01-16'),
    -> (1008, 108, 'withdrawal', 1800.00, '2024-01-17'),
    -> (1009, 109, 'deposit', 1600.00, '2024-01-18'),
    -> (1010, 110, 'withdrawal', 1000.00, '2024-01-19');
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 |
+----------------+------------+------------------+---------+------------------+
|           1001 |        101 | deposit          | 1000.00 | 2024-01-10       |
|           1002 |        102 | withdrawal       | 2000.00 | 2024-01-11       |
|           1003 |        103 | deposit          | 1500.00 | 2024-01-12       |
|           1004 |        104 | withdrawal       | 2500.00 | 2024-01-13       |
|           1005 |        105 | deposit          | 1200.00 | 2024-01-14       |
|           1006 |        106 | withdrawal       |  800.00 | 2024-01-15       |
|           1007 |        107 | deposit          | 2000.00 | 2024-01-16       |
|           1008 |        108 | withdrawal       | 1800.00 | 2024-01-17       |
|           1009 |        109 | deposit          | 1600.00 | 2024-01-18       |
|           1010 |        110 | withdrawal       | 1000.00 | 2024-01-19       |
+----------------+------------+------------------+---------+------------------+
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.

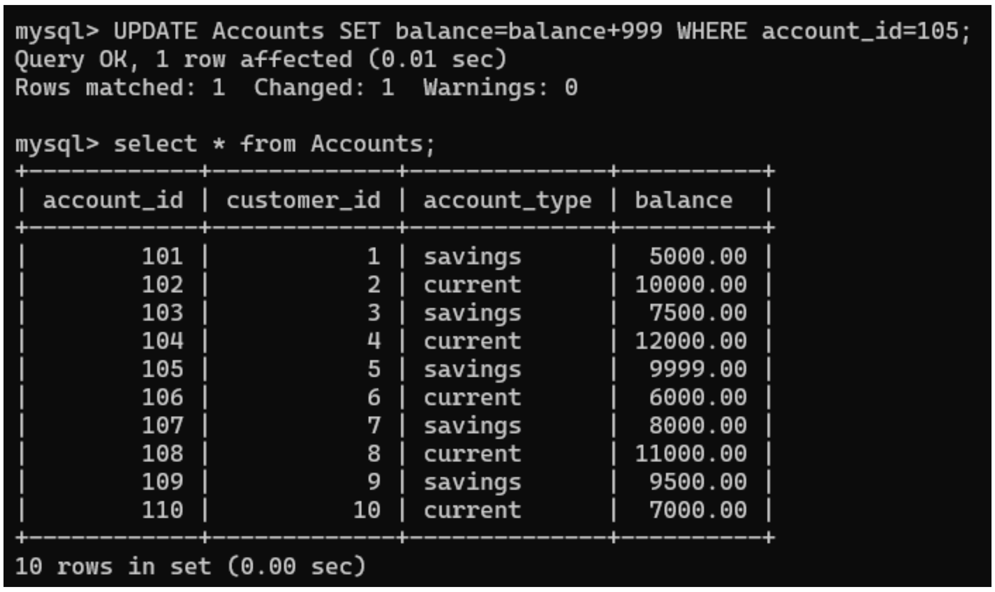

```
mysql> SELECT first_name as name,account_type,email FROM Customers c JOIN Accounts a WHERE
c.customer_id=a.customer_id;
+-----------+--------------+---------------------+
| name      | account_type | email               |
+-----------+--------------+---------------------+
| Aruna     | savings      | aruna@gmail.com     |
| Devi      | current      | devi@gmail.com      |
| Senthil   | savings      | senthil@gmail.com   |
| Priya     | current      | priya@gmail.com     |
| Vijay     | savings      | vijay@gmail.com     |
| Nithya    | current      | nithya@gmail.com    |
| Kumar     | savings      | kumar@gmail.com     |
| Sangeetha | current      | sangeetha@gmail.com |
| Ganesh    | savings      | ganesh@gmail.com    |
| Anitha    | current      | anitha@gmail.com    |
+-----------+--------------+---------------------+
10 rows in set (0.00 sec)
```

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



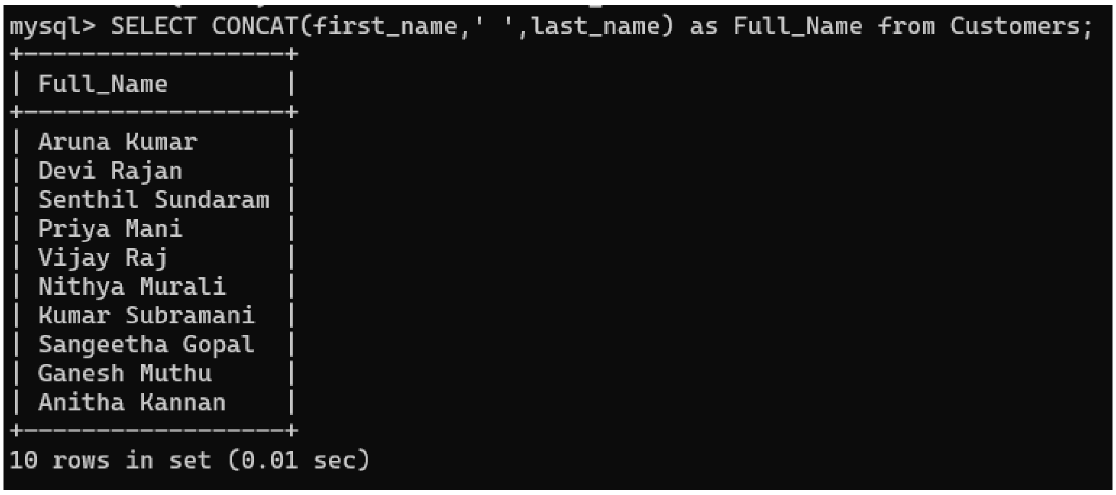
```
mysql> SELECT
    ->     c.customer_id,
    ->     c.first_name,
    ->     c.last_name,
    ->     t.transaction_id,
    ->     t.transaction_type,
    ->     t.amount,
    ->     t.transaction_date
    -> FROM
    ->     Customers c
    -> JOIN
    ->     Accounts a ON c.customer_id = a.customer_id
    -> JOIN
    ->     Transactions t ON a.account_id = t.account_id;
+-------------+------------+-----------+----------------+------------------+---------+------------------+
| customer_id | first_name | last_name | transaction_id | transaction_type | amount  | transaction_date |
+-------------+------------+-----------+----------------+------------------+---------+------------------+
|           1 | Aruna      | Kumar     |           1001 | deposit          | 1000.00 | 2024-01-10       |
|           2 | Devi       | Rajan     |           1002 | withdrawal       | 2000.00 | 2024-01-11       |
|           3 | Senthil    | Sundaram  |           1003 | deposit          | 1500.00 | 2024-01-12       |
|           4 | Priya      | Mani      |           1004 | withdrawal       | 2500.00 | 2024-01-13       |
|           5 | Vijay      | Raj       |           1005 | deposit          | 1200.00 | 2024-01-14       |
|           6 | Nithya     | Murali    |           1006 | withdrawal       |  800.00 | 2024-01-15       |
|           7 | Kumar      | Subramani |           1007 | deposit          | 2000.00 | 2024-01-16       |
|           8 | Sangeetha  | Gopal     |           1008 | withdrawal       | 1800.00 | 2024-01-17       |
|           9 | Ganesh     | Muthu     |           1009 | deposit          | 1600.00 | 2024-01-18       |
|          10 | Anitha     | Kannan    |           1010 | withdrawal       | 1000.00 | 2024-01-19       |
+-------------+------------+-----------+----------------+------------------+---------+------------------+
10 rows in set (0.00 sec)
```

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



```
mysql> UPDATE Accounts SET balance=balance+999 WHERE account_id=105;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Accounts;
+------------+-------------+--------------+----------+
| account_id | customer_id | account_type | balance  |
+------------+-------------+--------------+----------+
|        101 |           1 | savings      |  5000.00 |
|        102 |           2 | current      | 10000.00 |
|        103 |           3 | savings      |  7500.00 |
|        104 |           4 | current      | 12000.00 |
|        105 |           5 | savings      |  9999.00 |
|        106 |           6 | current      |  6000.00 |
|        107 |           7 | savings      |  8000.00 |
|        108 |           8 | current      | 11000.00 |
|        109 |           9 | savings      |  9500.00 |
|        110 |          10 | current      |  7000.00 |
+------------+-------------+--------------+----------+
10 rows in set (0.00 sec)
```

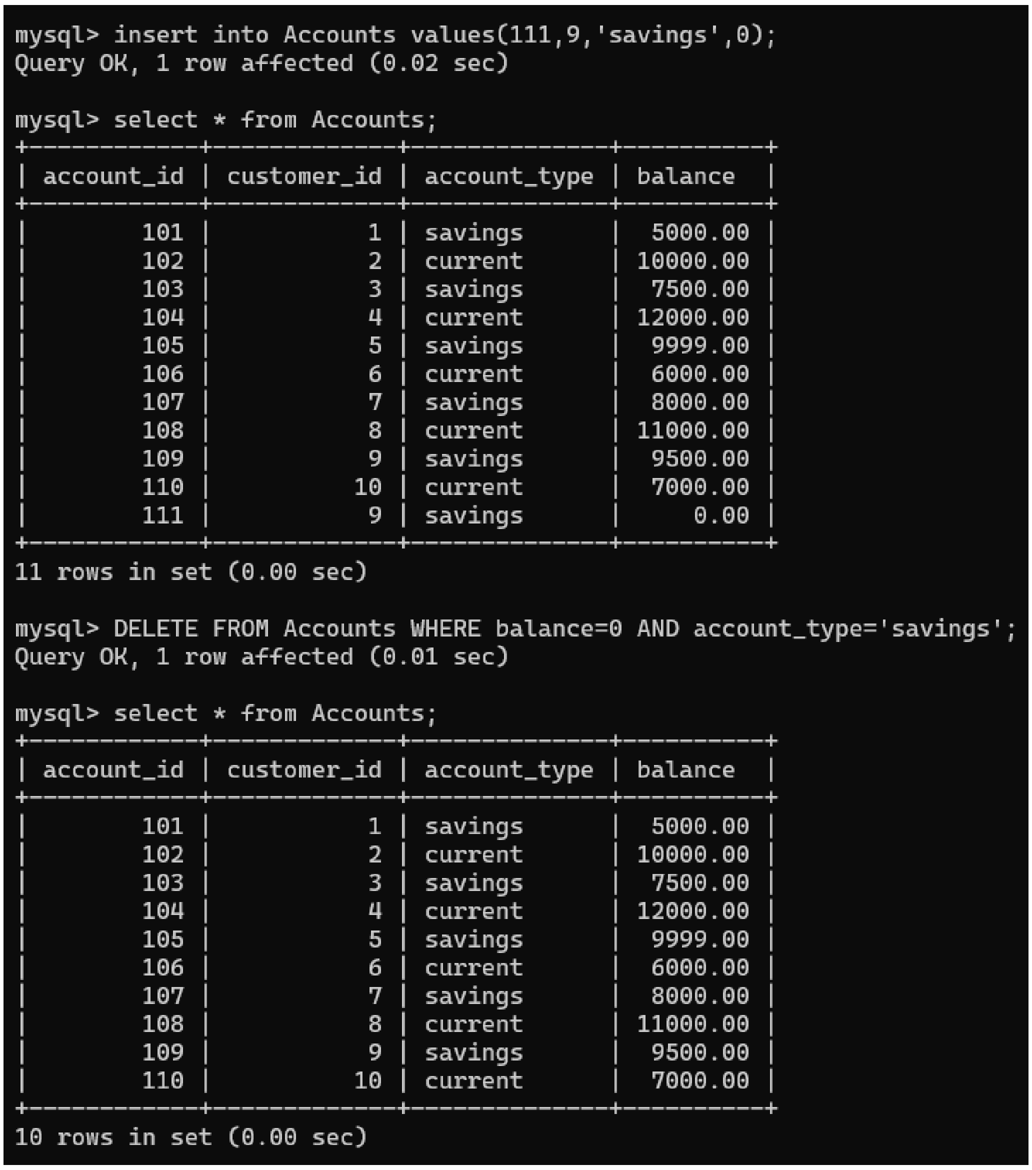
4. Write a SQL query to Combine first and last names of customers as a full_name.



```
mysql> SELECT CONCAT(first_name,' ',last_name) as Full_Name from Customers;
+-------------------+
| Full_Name         |
+-------------------+
| Aruna Kumar       |
| Devi Rajan        |
| Senthil Sundaram  |
| Priya Mani        |
| Vijay Raj         |
| Nithya Murali     |
| Kumar Subramani   |
| Sangeetha Gopal   |
| Ganesh Muthu      |
| Anitha Kannan     |
+-------------------+
10 rows in set (0.01 sec)
```

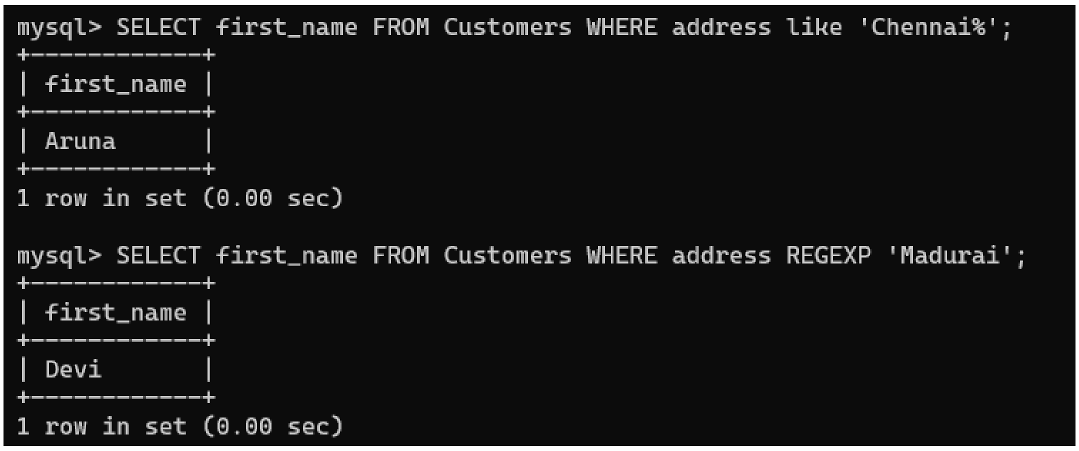
5. Write a SQL query to remove accounts with a balance of zero where the account

type is savings.



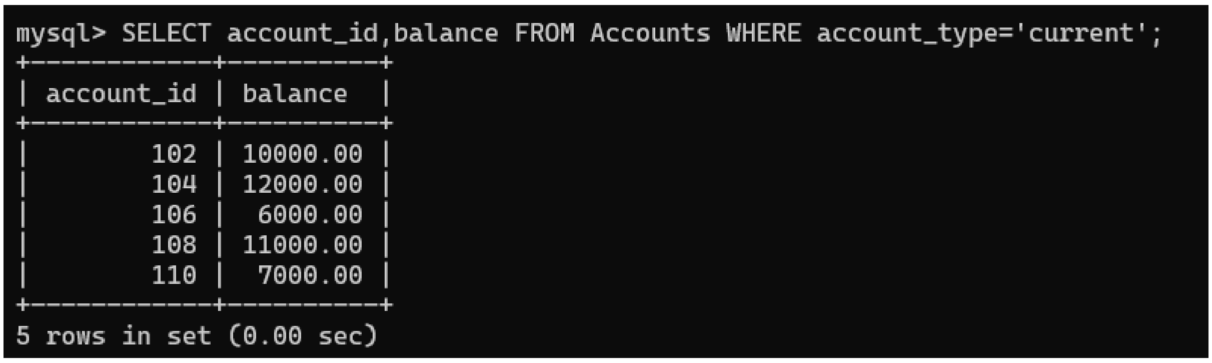
```
mysql> insert into Accounts values(111,9,'savings',0);
Query OK, 1 row affected (0.02 sec)

mysql> select * from Accounts;
+------------+-------------+--------------+----------+
| account_id | customer_id | account_type | balance  |
+------------+-------------+--------------+----------+
|        101 |           1 | savings      |  5000.00 |
|        102 |           2 | current      | 10000.00 |
|        103 |           3 | savings      |  7500.00 |
|        104 |           4 | current      | 12000.00 |
|        105 |           5 | savings      |  9999.00 |
|        106 |           6 | current      |  6000.00 |
|        107 |           7 | savings      |  8000.00 |
|        108 |           8 | current      | 11000.00 |
|        109 |           9 | savings      |  9500.00 |
|        110 |          10 | current      |  7000.00 |
|        111 |           9 | savings      |     0.00 |
+------------+-------------+--------------+----------+
11 rows in set (0.00 sec)

mysql> DELETE FROM Accounts WHERE balance=0 AND account_type='savings';
Query OK, 1 row affected (0.01 sec)

mysql> select * from Accounts;
+------------+-------------+--------------+----------+
| account_id | customer_id | account_type | balance  |
+------------+-------------+--------------+----------+
|        101 |           1 | savings      |  5000.00 |
|        102 |           2 | current      | 10000.00 |
|        103 |           3 | savings      |  7500.00 |
|        104 |           4 | current      | 12000.00 |
|        105 |           5 | savings      |  9999.00 |
|        106 |           6 | current      |  6000.00 |
|        107 |           7 | savings      |  8000.00 |
|        108 |           8 | current      | 11000.00 |
|        109 |           9 | savings      |  9500.00 |
|        110 |          10 | current      |  7000.00 |
+------------+-------------+--------------+----------+
10 rows in set (0.00 sec)
```

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



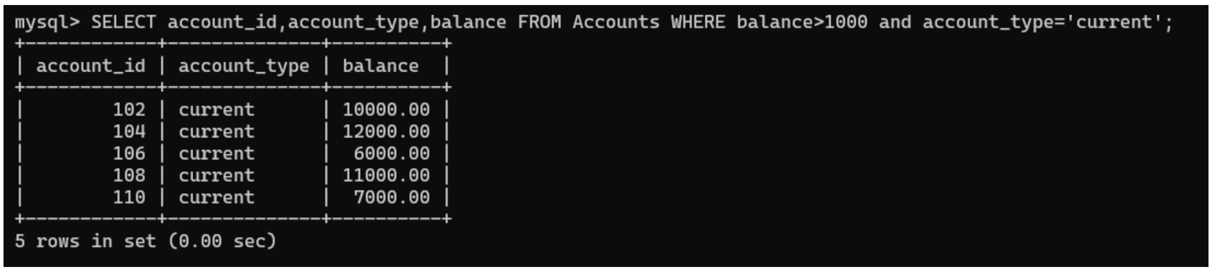
```
mysql> SELECT first_name FROM Customers WHERE address like 'Chennai%';
+------------+
| first_name |
+------------+
| Aruna      |
+------------+
1 row in set (0.00 sec)

mysql> SELECT first_name FROM Customers WHERE address REGEXP 'Madurai';
+------------+
| first_name |
+------------+
| Devi       |
+------------+
1 row in set (0.00 sec)
```

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



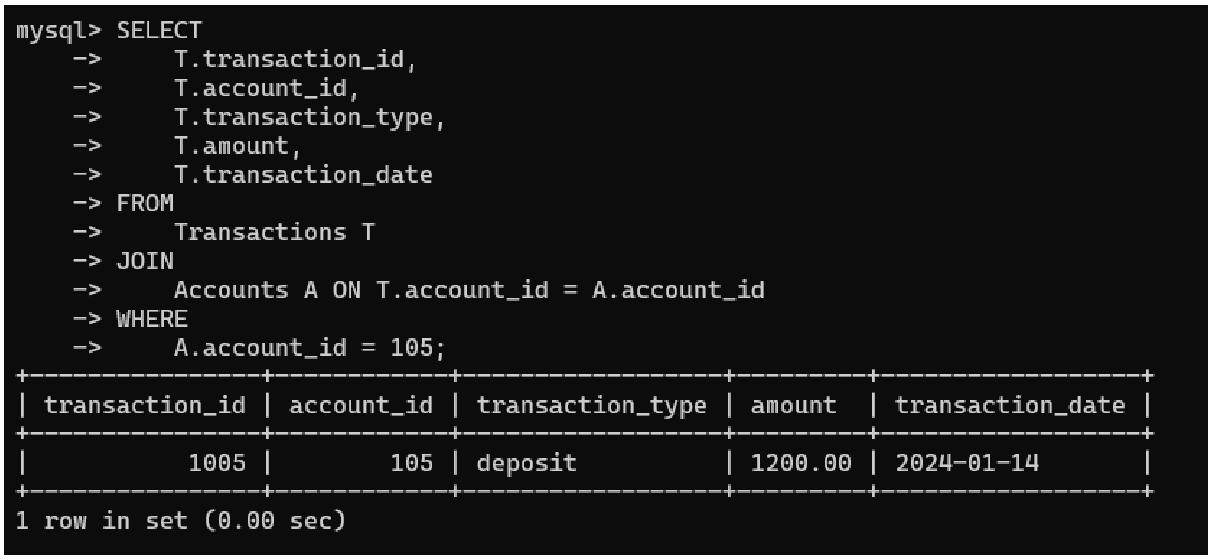
```
mysql> SELECT account_id,balance FROM Accounts WHERE account_type='current';
+------------+----------+
| account_id | balance  |
+------------+----------+
|        102 | 10000.00 |
|        104 | 12000.00 |
|        106 |  6000.00 |
|        108 | 11000.00 |
|        110 |  7000.00 |
+------------+----------+
5 rows in set (0.00 sec)
```

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



```
mysql> SELECT account_id,account_type,balance FROM Accounts WHERE balance>1000 and account_type='current';
+------------+--------------+----------+
| account_id | account_type | balance  |
+------------+--------------+----------+
|        102 | current      | 10000.00 |
|        104 | current      | 12000.00 |
|        106 | current      |  6000.00 |
|        108 | current      | 11000.00 |
|        110 | current      |  7000.00 |
+------------+--------------+----------+
5 rows in set (0.00 sec)
```

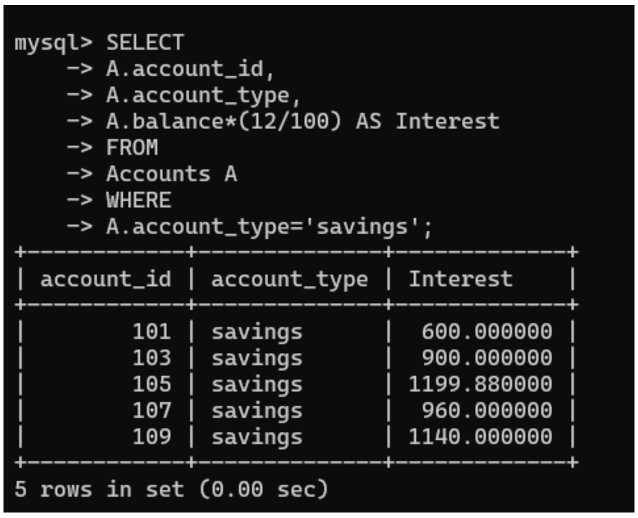
9. Write a SQL query to Retrieve all transactions for a specific account.



```
mysql> SELECT
    ->     T.transaction_id,
    ->     T.account_id,
    ->     T.transaction_type,
    ->     T.amount,
    ->     T.transaction_date
    -> FROM
    ->     Transactions T
    -> JOIN
    ->     Accounts A ON T.account_id = A.account_id
    -> WHERE
    ->     A.account_id = 105;
+----------------+------------+------------------+---------+------------------+
| transaction_id | account_id | transaction_type | amount  | transaction_date |
+----------------+------------+------------------+---------+------------------+
|           1005 |        105 | deposit          | 1200.00 | 2024-01-14       |
+----------------+------------+------------------+---------+------------------+
1 row in set (0.00 sec)
```

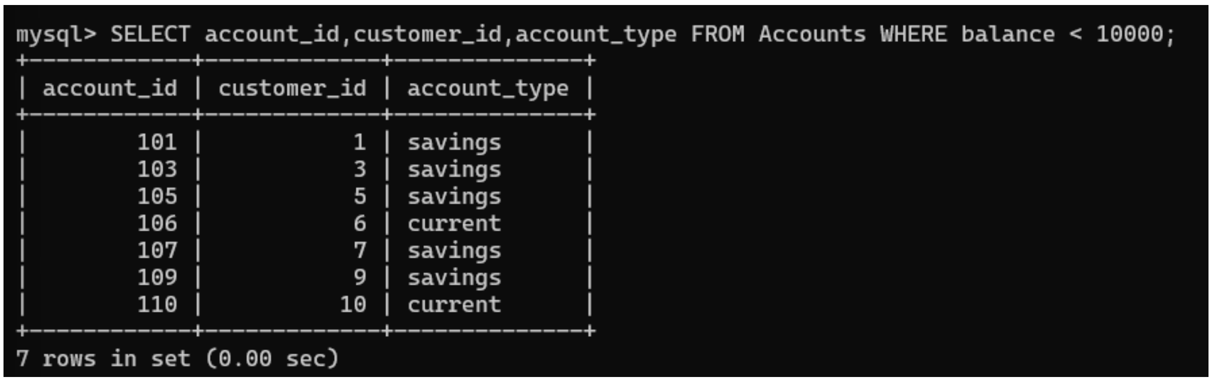
10. Write a SQL query to Calculate the interest accrued on savings accounts based on a

given interest rate.



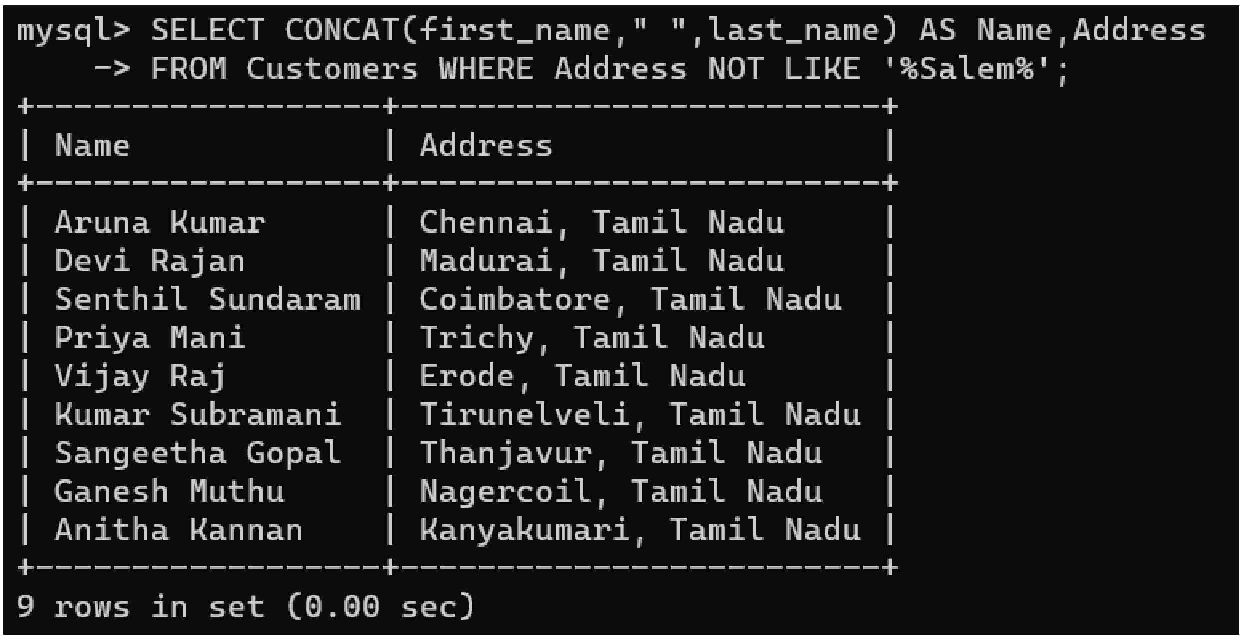
```
mysql> SELECT
    -> A.account_id,
    -> A.account_type,
    -> A.balance*(12/100) AS Interest
    -> FROM
    -> Accounts A
    -> WHERE
    -> A.account_type='savings';
+------------+--------------+-------------+
| account_id | account_type | Interest    |
+------------+--------------+-------------+
|        101 | savings      |  600.000000 |
|        103 | savings      |  900.000000 |
|        105 | savings      | 1199.880000 |
|        107 | savings      |  960.000000 |
|        109 | savings      | 1140.000000 |
+------------+--------------+-------------+
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,customer_id,account_type FROM Accounts WHERE balance < 10000;
+------------+-------------+--------------+
| account_id | customer_id | account_type |
+------------+-------------+--------------+
|        101 |           1 | savings      |
|        103 |           3 | savings      |
|        105 |           5 | savings      |
|        106 |           6 | current      |
|        107 |           7 | savings      |
|        109 |           9 | savings      |
|        110 |          10 | current      |
+------------+-------------+--------------+
7 rows in set (0.00 sec)
```

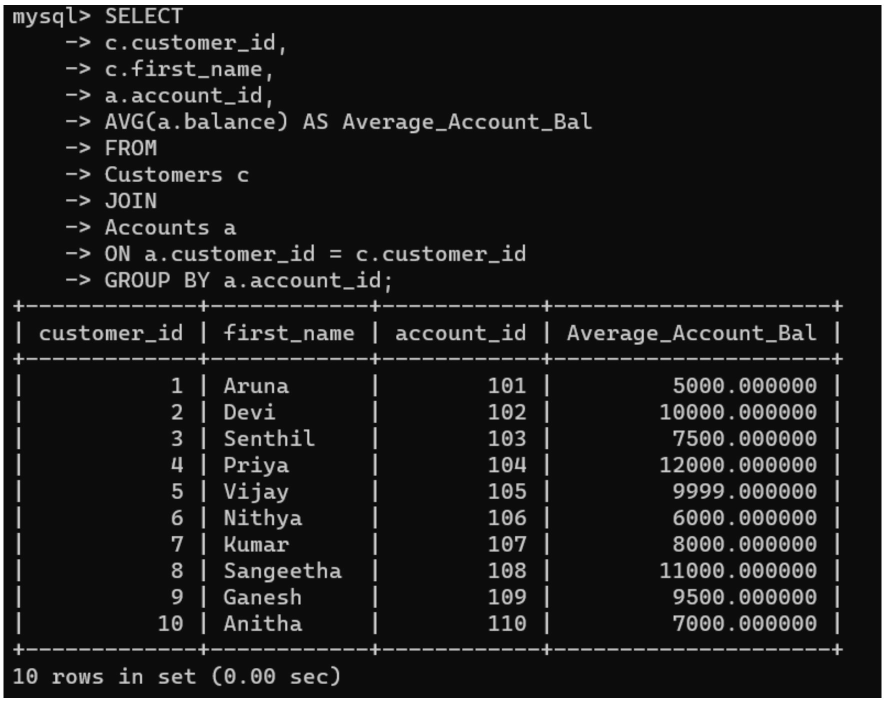
12. Write a SQL query to Find customers not living in a specific city



```
mysql> SELECT CONCAT(first_name," ",last_name) AS Name,Address
    -> FROM Customers WHERE Address NOT LIKE '%Salem%';
+------------------+------------------------+
| Name             | Address                |
+------------------+------------------------+
| Aruna Kumar      | Chennai, Tamil Nadu    |
| Devi Rajan       | Madurai, Tamil Nadu    |
| Senthil Sundaram | Coimbatore, Tamil Nadu |
| Priya Mani       | Trichy, Tamil Nadu     |
| Vijay Raj        | Erode, Tamil Nadu      |
| Kumar Subramani  | Tirunelveli, Tamil Nadu |
| Sangeetha Gopal  | Thanjavur, Tamil Nadu  |
| Ganesh Muthu     | Nagercoil, Tamil Nadu  |
| Anitha Kannan    | Kanyakumari, Tamil Nadu |
+------------------+------------------------+
9 rows in set (0.00 sec)
```

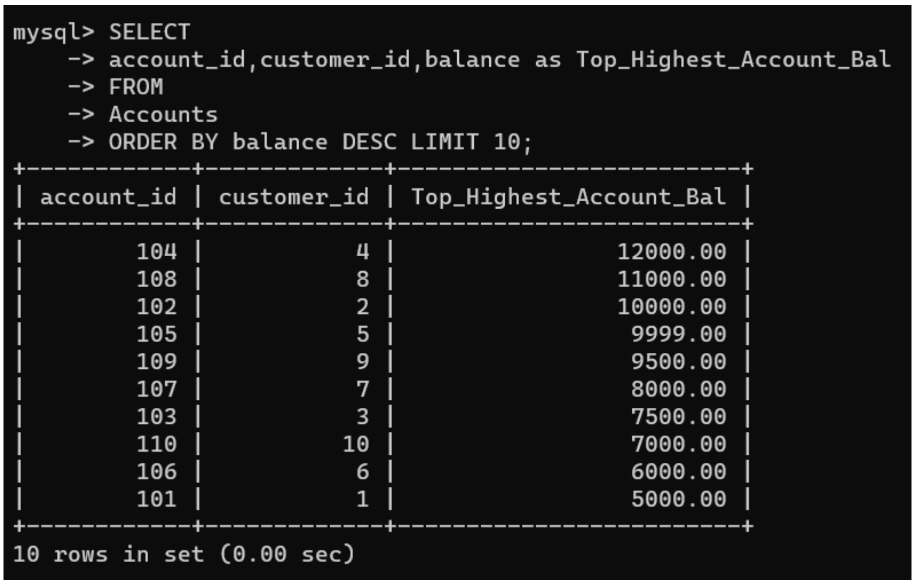
Tasks 3: Aggregate functions, Having, Order By, GroupBy and Joins:

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



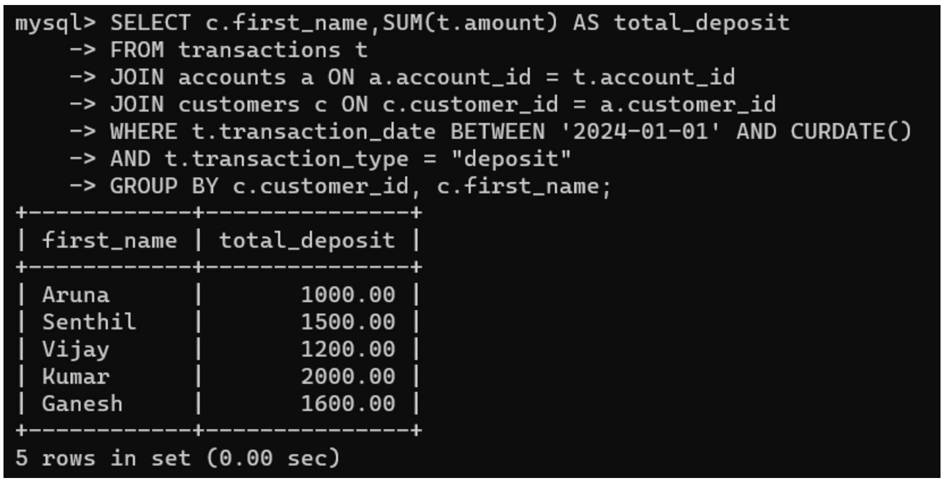
```
mysql> SELECT
    -> c.customer_id,
    -> c.first_name,
    -> a.account_id,
    -> AVG(a.balance) AS Average_Account_Bal
    -> FROM
    -> Customers c
    -> JOIN
    -> Accounts a
    -> ON a.customer_id = c.customer_id
    -> GROUP BY a.account_id;
+-------------+------------+------------+---------------------+
| customer_id | first_name | account_id | Average_Account_Bal |
+-------------+------------+------------+---------------------+
|           1 | Aruna      |        101 |         5000.000000 |
|           2 | Devi       |        102 |        10000.000000 |
|           3 | Senthil    |        103 |         7500.000000 |
|           4 | Priya      |        104 |        12000.000000 |
|           5 | Vijay      |        105 |         9999.000000 |
|           6 | Nithya     |        106 |         6000.000000 |
|           7 | Kumar      |        107 |         8000.000000 |
|           8 | Sangeetha  |        108 |        11000.000000 |
|           9 | Ganesh     |        109 |         9500.000000 |
|          10 | Anitha     |        110 |         7000.000000 |
+-------------+------------+------------+---------------------+
10 rows in set (0.00 sec)
```

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



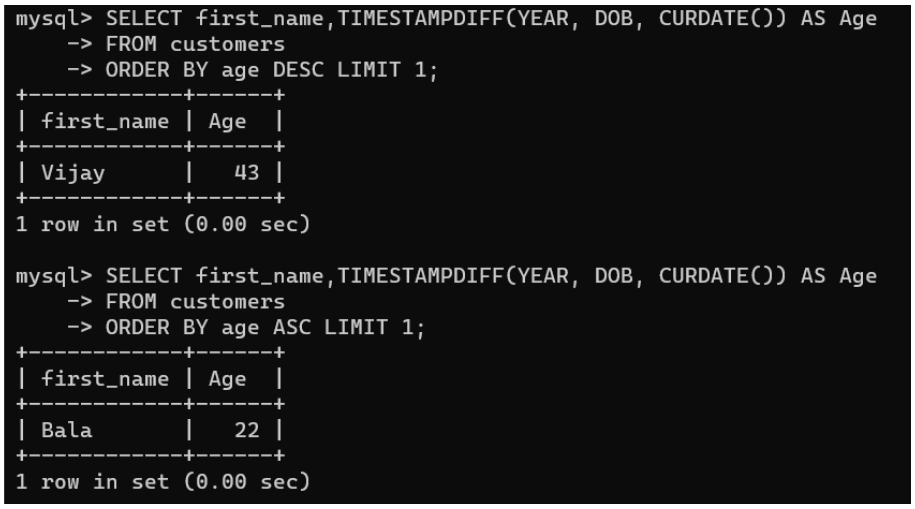
```
mysql> SELECT
    -> account_id,customer_id,balance as Top_Highest_Account_Bal
    -> FROM
    -> Accounts
    -> ORDER BY balance DESC LIMIT 10;
+------------+-------------+-------------------------+
| account_id | customer_id | Top_Highest_Account_Bal |
+------------+-------------+-------------------------+
|        104 |           4 |                12000.00 |
|        108 |           8 |                11000.00 |
|        102 |           2 |                10000.00 |
|        105 |           5 |                 9999.00 |
|        109 |           9 |                 9500.00 |
|        107 |           7 |                 8000.00 |
|        103 |           3 |                 7500.00 |
|        110 |          10 |                 7000.00 |
|        106 |           6 |                 6000.00 |
|        101 |           1 |                 5000.00 |
+------------+-------------+-------------------------+
10 rows in set (0.00 sec)
```

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



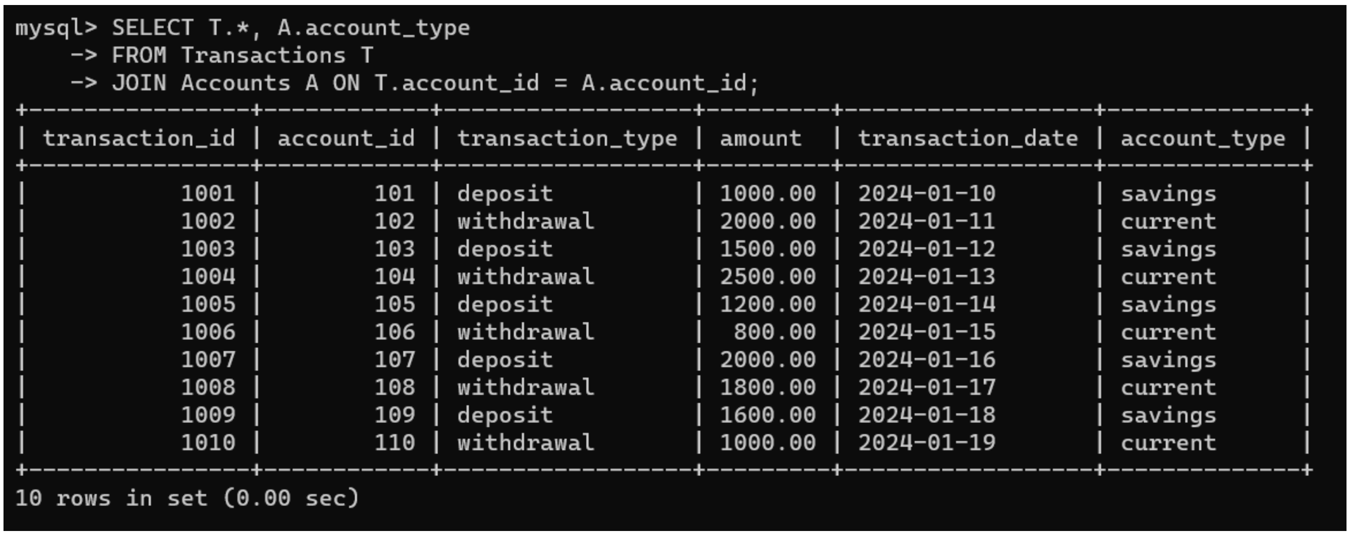
```
mysql> SELECT c.first_name,SUM(t.amount) AS total_deposit
    -> FROM transactions t
    -> JOIN accounts a ON a.account_id = t.account_id
    -> JOIN customers c ON c.customer_id = a.customer_id
    -> WHERE t.transaction_date BETWEEN '2024-01-01' AND CURDATE()
    -> AND t.transaction_type = "deposit"
    -> GROUP BY c.customer_id, c.first_name;
+------------+---------------+
| first_name | total_deposit |
+------------+---------------+
| Aruna      |       1000.00 |
| Senthil    |       1500.00 |
| Vijay      |       1200.00 |
| Kumar      |       2000.00 |
| Ganesh     |       1600.00 |
+------------+---------------+
5 rows in set (0.00 sec)
```

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



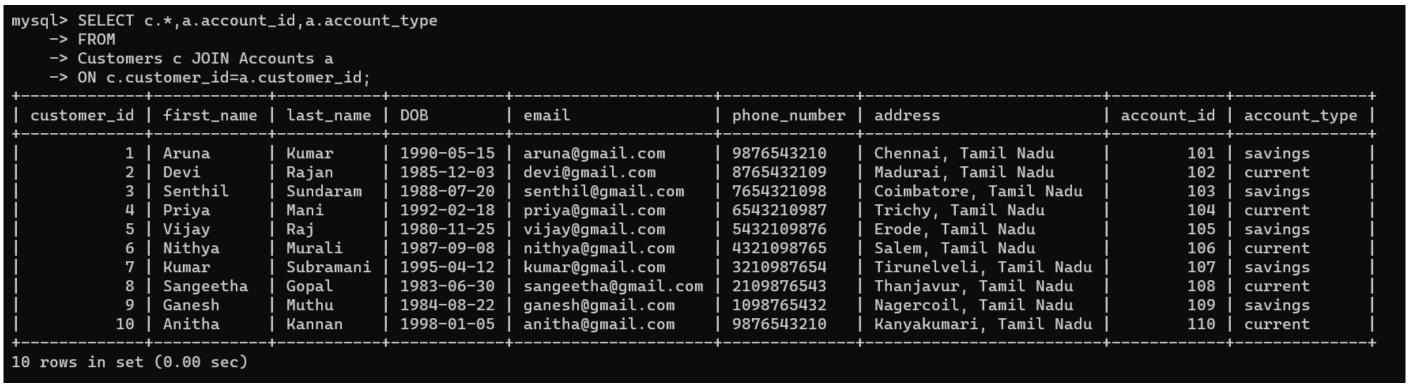
```
mysql> SELECT first_name,TIMESTAMPDIFF(YEAR, DOB, CURDATE()) AS Age
    -> FROM customers
    -> ORDER BY age DESC LIMIT 1;
+------------+------+
| first_name | Age  |
+------------+------+
| Vijay      |   43 |
+------------+------+
1 row in set (0.00 sec)

mysql> SELECT first_name,TIMESTAMPDIFF(YEAR, DOB, CURDATE()) AS Age
    -> FROM customers
    -> ORDER BY age ASC LIMIT 1;
+------------+------+
| first_name | Age  |
+------------+------+
| Bala       |   22 |
+------------+------+
1 row 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          | 1000.00 | 2024-01-10       | savings      |
|           1002 |        102 | withdrawal       | 2000.00 | 2024-01-11       | current      |
|           1003 |        103 | deposit          | 1500.00 | 2024-01-12       | savings      |
|           1004 |        104 | withdrawal       | 2500.00 | 2024-01-13       | current      |
|           1005 |        105 | deposit          | 1200.00 | 2024-01-14       | savings      |
|           1006 |        106 | withdrawal       |  800.00 | 2024-01-15       | current      |
|           1007 |        107 | deposit          | 2000.00 | 2024-01-16       | savings      |
|           1008 |        108 | withdrawal       | 1800.00 | 2024-01-17       | current      |
|           1009 |        109 | deposit          | 1600.00 | 2024-01-18       | savings      |
|           1010 |        110 | withdrawal       | 1000.00 | 2024-01-19       | current      |
+----------------+------------+------------------+---------+------------------+--------------+
10 rows in set (0.00 sec)
```

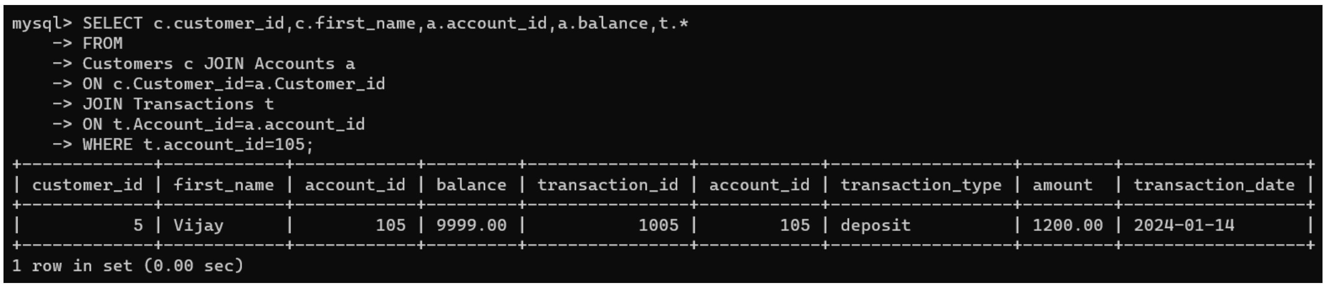
6. Write a SQL query to Get a list of customers along with their account details.



```
mysql> SELECT c.*,a.account_id,a.account_type
    -> FROM
    -> Customers c JOIN Accounts a
    -> ON c.customer_id=a.customer_id;
+-------------+------------+-----------+------------+---------------------+--------------+--------------------------+------------+--------------+
| customer_id | first_name | last_name | DOB        | email               | phone_number | address                  | account_id | account_type |
+-------------+------------+-----------+------------+---------------------+--------------+--------------------------+------------+--------------+
|           1 | Aruna      | Kumar     | 1990-05-15 | aruna@gmail.com     | 9876543210   | Chennai, Tamil Nadu      |        101 | savings      |
|           2 | Devi       | Rajan     | 1985-12-03 | devi@gmail.com      | 8765432109   | Madurai, Tamil Nadu      |        102 | current      |
|           3 | Senthil    | Sundaram  | 1988-07-20 | senthil@gmail.com   | 7654321098   | Coimbatore, Tamil Nadu   |        103 | savings      |
|           4 | Priya      | Mani      | 1992-02-18 | priya@gmail.com     | 6543210987   | Trichy, Tamil Nadu       |        104 | current      |
|           5 | Vijay      | Raj       | 1980-11-25 | vijay@gmail.com     | 5432109876   | Erode, Tamil Nadu        |        105 | savings      |
|           6 | Nithya     | Murali    | 1987-09-08 | nithya@gmail.com    | 4321098765   | Salem, Tamil Nadu        |        106 | current      |
|           7 | Kumar      | Subramani | 1995-04-12 | kumar@gmail.com     | 3210987654   | Tirunelveli, Tamil Nadu  |        107 | savings      |
|           8 | Sangeetha  | Gopal     | 1983-06-30 | sangeetha@gmail.com | 2109876543   | Thanjavur, Tamil Nadu    |        108 | current      |
|           9 | Ganesh     | Muthu     | 1984-08-22 | ganesh@gmail.com    | 1098765432   | Nagercoil, Tamil Nadu    |        109 | savings      |
|          10 | Anitha     | Kannan    | 1998-01-05 | anitha@gmail.com    | 9876543210   | Kanyakumari, Tamil Nadu  |        110 | current      |
+-------------+------------+-----------+------------+---------------------+--------------+--------------------------+------------+--------------+
10 rows in set (0.00 sec)
```

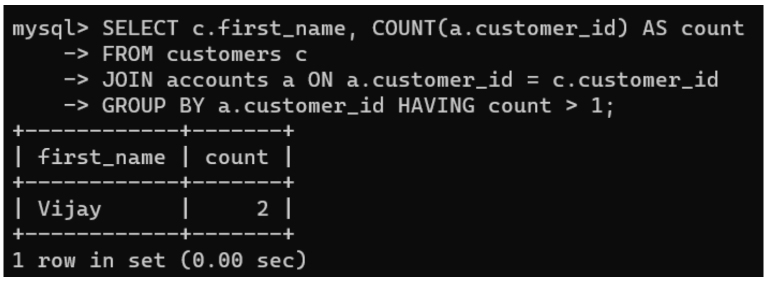
7. Write a SQL query to Retrieve transaction details along with customer information for a

specific account.



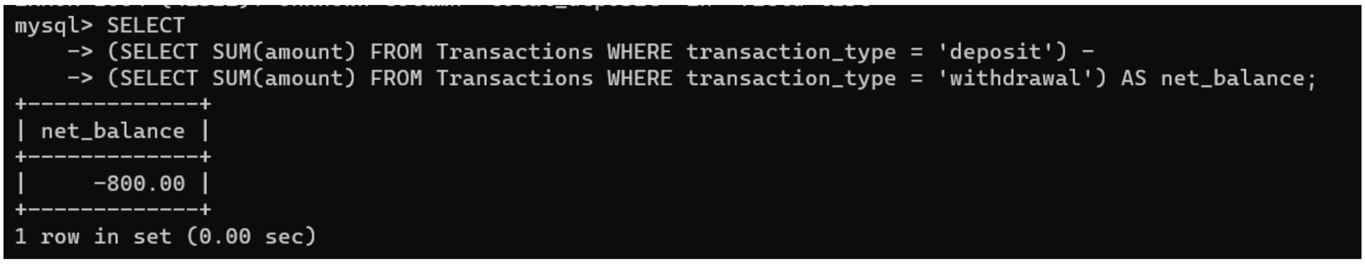
```
mysql> SELECT c.customer_id,c.first_name,a.account_id,a.balance,t.*
    -> FROM
    -> Customers c JOIN Accounts a
    -> ON c.Customer_id=a.Customer_id
    -> JOIN Transactions t
    -> ON t.Account_id=a.account_id
    -> WHERE t.account_id=105;
+-------------+------------+------------+---------+----------------+------------+------------------+---------+------------------+
| customer_id | first_name | account_id | balance | transaction_id | account_id | transaction_type | amount  | transaction_date |
+-------------+------------+------------+---------+----------------+------------+------------------+---------+------------------+
|           5 | Vijay      |        105 | 9999.00 |           1005 |        105 | deposit          | 1200.00 | 2024-01-14       |
+-------------+------------+------------+---------+----------------+------------+------------------+---------+------------------+
1 row in set (0.00 sec)
```

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



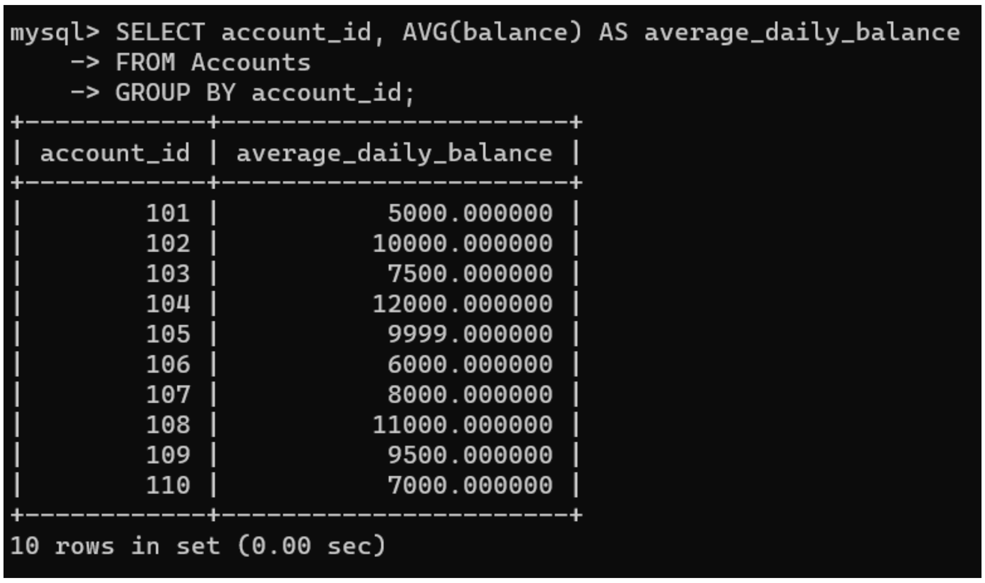
```
mysql> SELECT c.first_name, COUNT(a.customer_id) AS count
    -> FROM customers c
    -> JOIN accounts a ON a.customer_id = c.customer_id
    -> GROUP BY a.customer_id HAVING count > 1;
+------------+-------+
| first_name | count |
+------------+-------+
| Vijay      |     2 |
+------------+-------+
1 row in set (0.00 sec)
```

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



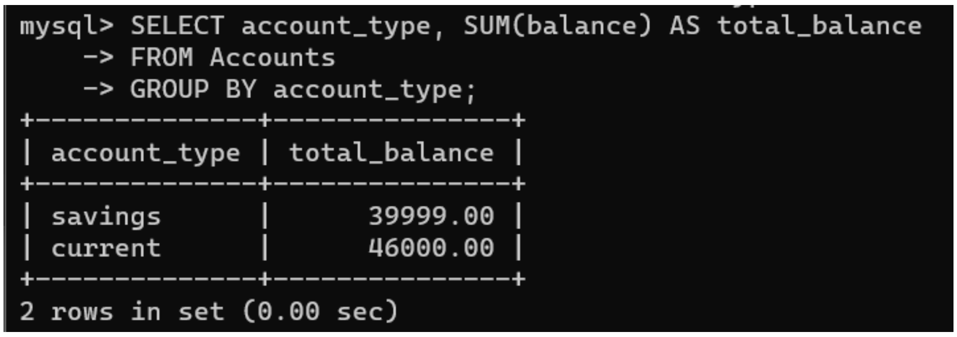
```
mysql> SELECT
    -> (SELECT SUM(amount) FROM Transactions WHERE transaction_type = 'deposit') -
    -> (SELECT SUM(amount) FROM Transactions WHERE transaction_type = 'withdrawal') AS net_balance;
+-------------+
| net_balance |
+-------------+
|     -800.00 |
+-------------+
1 row in set (0.00 sec)
```

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



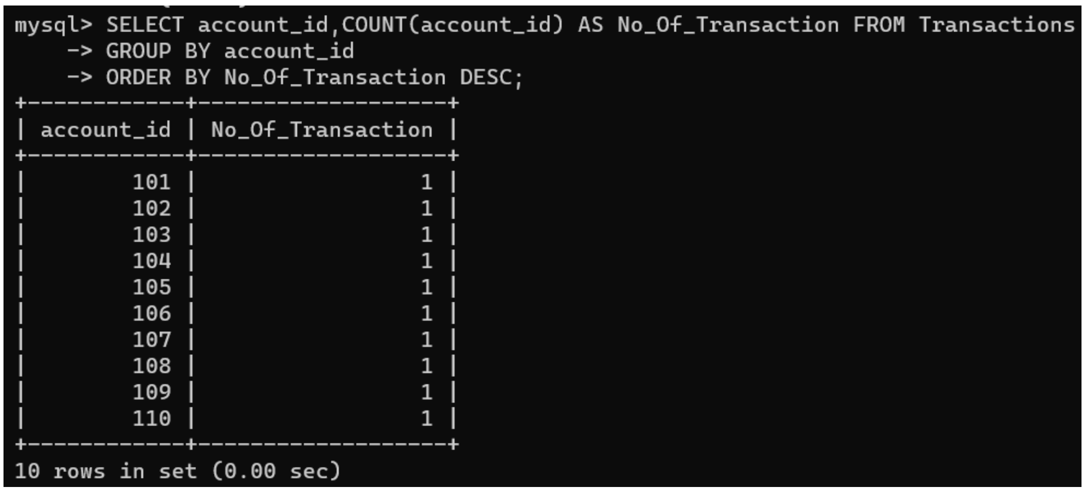
```
mysql> SELECT account_id, AVG(balance) AS average_daily_balance
    -> FROM Accounts
    -> GROUP BY account_id;
+------------+-----------------------+
| account_id | average_daily_balance |
+------------+-----------------------+
|        101 |           5000.000000 |
|        102 |          10000.000000 |
|        103 |           7500.000000 |
|        104 |          12000.000000 |
|        105 |           9999.000000 |
|        106 |           6000.000000 |
|        107 |           8000.000000 |
|        108 |          11000.000000 |
|        109 |           9500.000000 |
|        110 |           7000.000000 |
+------------+-----------------------+
10 rows in set (0.00 sec)
```

11. Calculate the total balance for each account type.



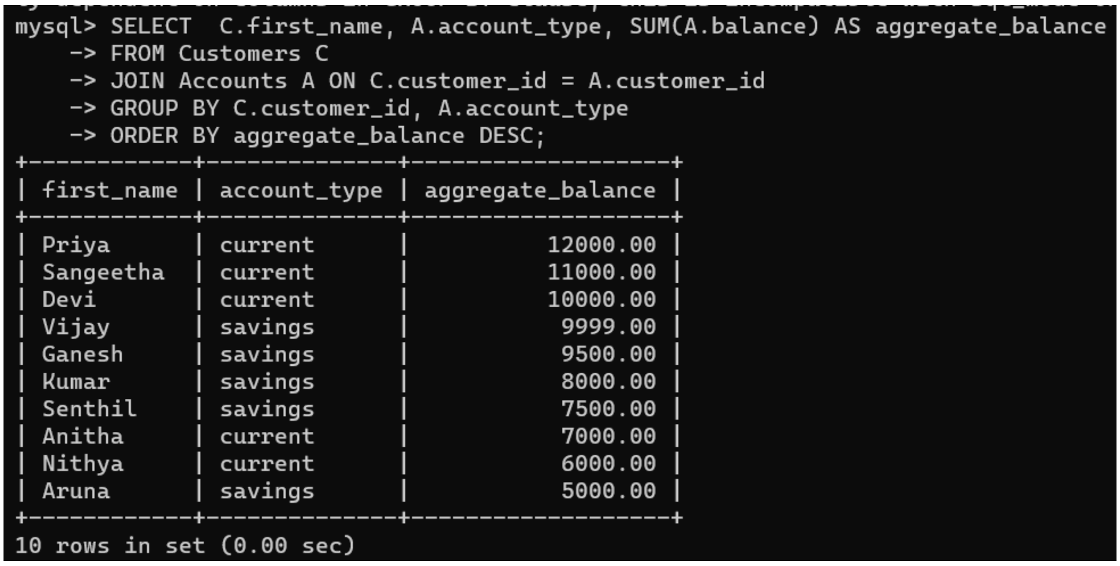
```
mysql> SELECT account_type, SUM(balance) AS total_balance
    -> FROM Accounts
    -> GROUP BY account_type;
+--------------+---------------+
| account_type | total_balance |
+--------------+---------------+
| savings      |      39999.00 |
| current      |      46000.00 |
+--------------+---------------+
2 rows in set (0.00 sec)
```

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



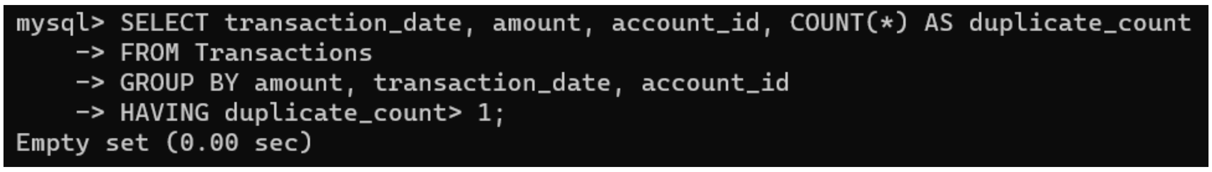
```
mysql> SELECT account_id,COUNT(account_id) AS No_Of_Transaction FROM Transactions
    -> GROUP BY account_id
    -> ORDER BY No_Of_Transaction DESC;
+------------+-------------------+
| account_id | No_Of_Transaction |
+------------+-------------------+
|        101 |                 1 |
|        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)
```

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



```
mysql> SELECT  C.first_name, A.account_type, SUM(A.balance) AS aggregate_balance
    -> FROM Customers C
    -> JOIN Accounts A ON C.customer_id = A.customer_id
    -> GROUP BY C.customer_id, A.account_type
    -> ORDER BY aggregate_balance DESC;
+------------+--------------+-------------------+
| first_name | account_type | aggregate_balance |
+------------+--------------+-------------------+
| Priya      | current      |          12000.00 |
| Sangeetha  | current      |          11000.00 |
| Devi       | current      |          10000.00 |
| Vijay      | savings      |           9999.00 |
| Ganesh     | savings      |           9500.00 |
| Kumar      | savings      |           8000.00 |
| Senthil    | savings      |           7500.00 |
| Anitha     | current      |           7000.00 |
| Nithya     | current      |           6000.00 |
| Aruna      | savings      |           5000.00 |
+------------+--------------+-------------------+
10 rows in set (0.00 sec)
```

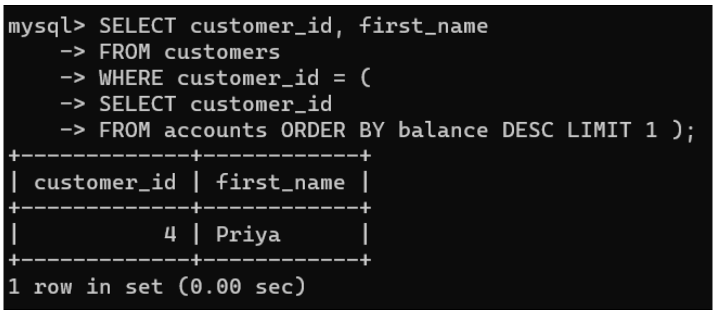
14. Identify and list duplicate transactions based on transaction amount, date, and account.



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

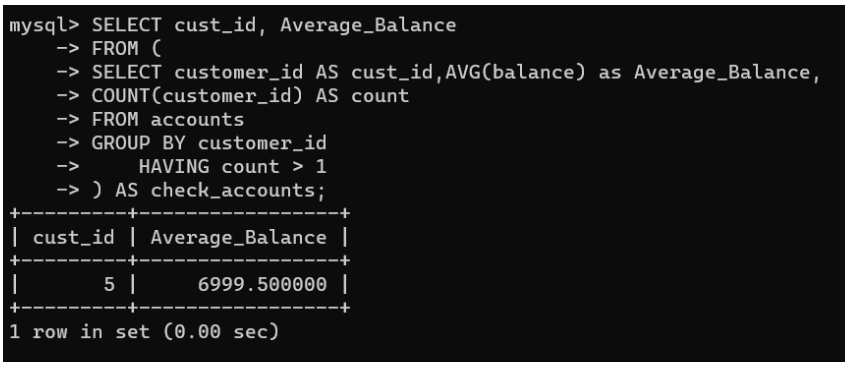
Tasks 4: Subquery and its type:

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



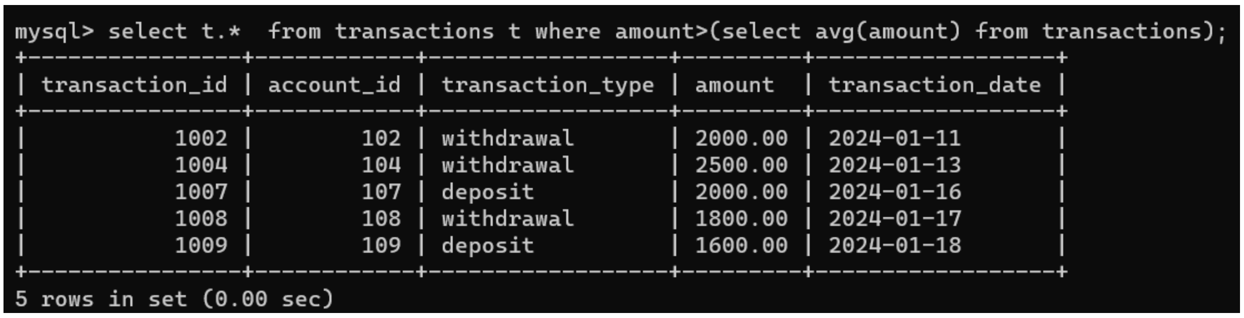
```
mysql> SELECT customer_id, first_name
    -> FROM customers
    -> WHERE customer_id = (
    -> SELECT customer_id
    -> FROM accounts ORDER BY balance DESC LIMIT 1 );
+-------------+------------+
| customer_id | first_name |
+-------------+------------+
|           4 | Priya      |
+-------------+------------+
1 row in set (0.00 sec)
```

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



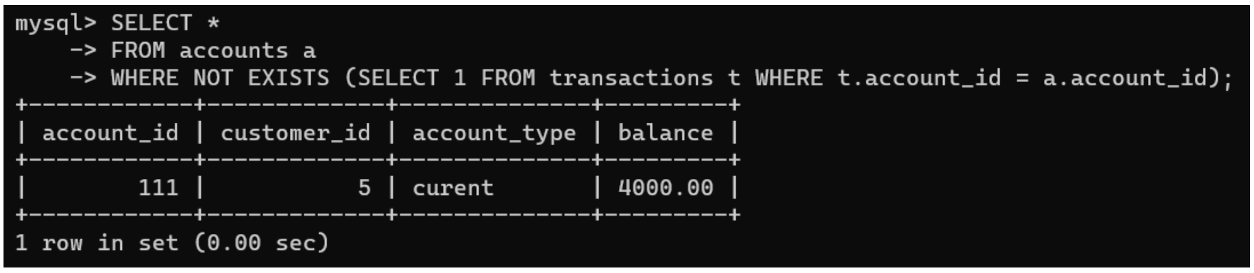
```
mysql> SELECT cust_id, Average_Balance
    -> FROM (
    -> SELECT customer_id AS cust_id,AVG(balance) as Average_Balance,
    -> COUNT(customer_id) AS count
    -> FROM accounts
    -> GROUP BY customer_id
    ->     HAVING count > 1
    -> ) AS check_accounts;
+---------+-----------------+
| cust_id | Average_Balance |
+---------+-----------------+
|       5 |     6999.500000 |
+---------+-----------------+
1 row in set (0.00 sec)
```

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



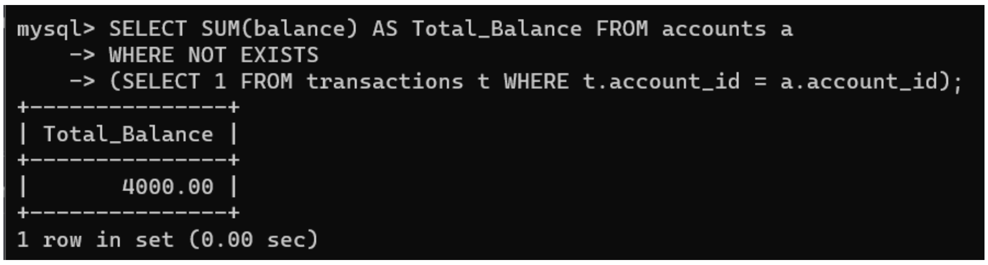
```
mysql> select t.*  from transactions t where amount>(select avg(amount) from transactions);
+----------------+------------+------------------+---------+------------------+
| transaction_id | account_id | transaction_type | amount  | transaction_date |
+----------------+------------+------------------+---------+------------------+
|           1002 |        102 | withdrawal       | 2000.00 | 2024-01-11       |
|           1004 |        104 | withdrawal       | 2500.00 | 2024-01-13       |
|           1007 |        107 | deposit          | 2000.00 | 2024-01-16       |
|           1008 |        108 | withdrawal       | 1800.00 | 2024-01-17       |
|           1009 |        109 | deposit          | 1600.00 | 2024-01-18       |
+----------------+------------+------------------+---------+------------------+
5 rows in set (0.00 sec)
```

4. Identify customers who have no recorded transactions.



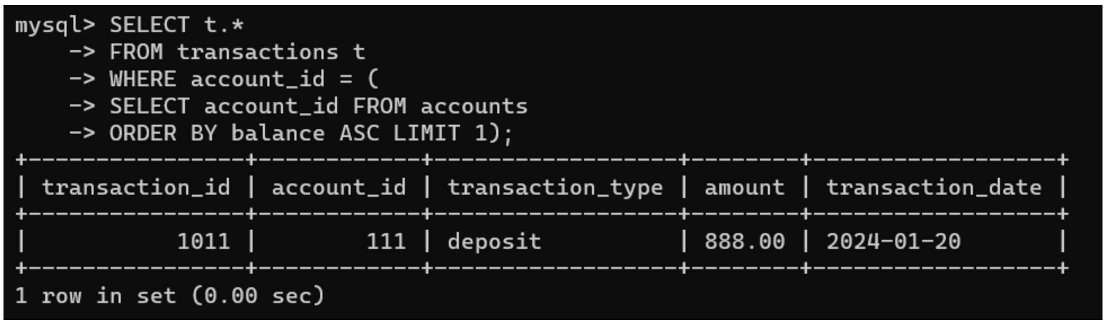
```
mysql> SELECT *
    -> FROM accounts a
    -> WHERE NOT EXISTS (SELECT 1 FROM transactions t WHERE t.account_id = a.account_id);
+------------+-------------+--------------+---------+
| account_id | customer_id | account_type | balance |
+------------+-------------+--------------+---------+
|        111 |           5 | curent       | 4000.00 |
+------------+-------------+--------------+---------+
1 row in set (0.00 sec)
```

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



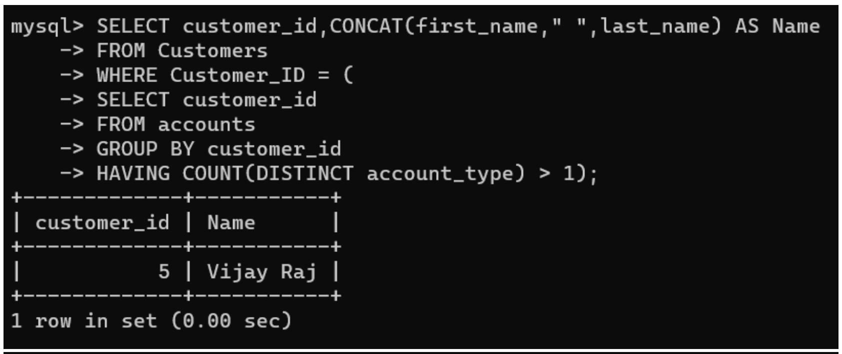
```
mysql> SELECT SUM(balance) AS Total_Balance FROM accounts a
    -> WHERE NOT EXISTS
    -> (SELECT 1 FROM transactions t WHERE t.account_id = a.account_id);
+---------------+
| Total_Balance |
+---------------+
|       4000.00 |
+---------------+
1 row in set (0.00 sec)
```

6. Retrieve transactions for accounts with the lowest balance.

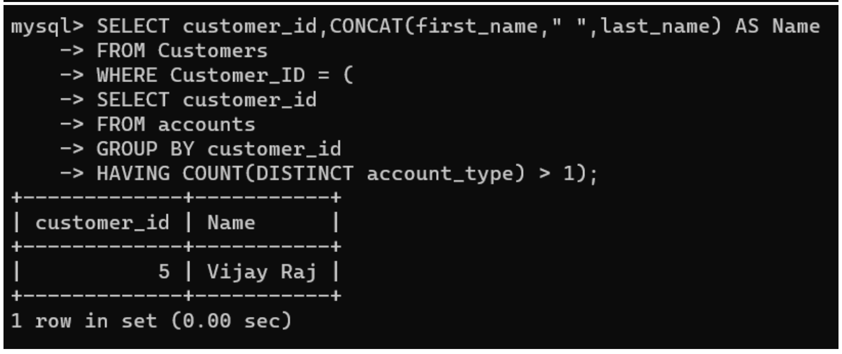


```
mysql> SELECT t.*
    -> FROM transactions t
    -> WHERE account_id = (
    -> SELECT account_id FROM accounts
    -> ORDER BY balance ASC LIMIT 1);
+----------------+------------+------------------+--------+------------------+
| transaction_id | account_id | transaction_type | amount | transaction_date |
+----------------+------------+------------------+--------+------------------+
|           1011 |        111 | deposit          | 888.00 | 2024-01-20       |
+----------------+------------+------------------+--------+------------------+
1 row in set (0.00 sec)
```

7. Identify customers who have accounts of multiple types.

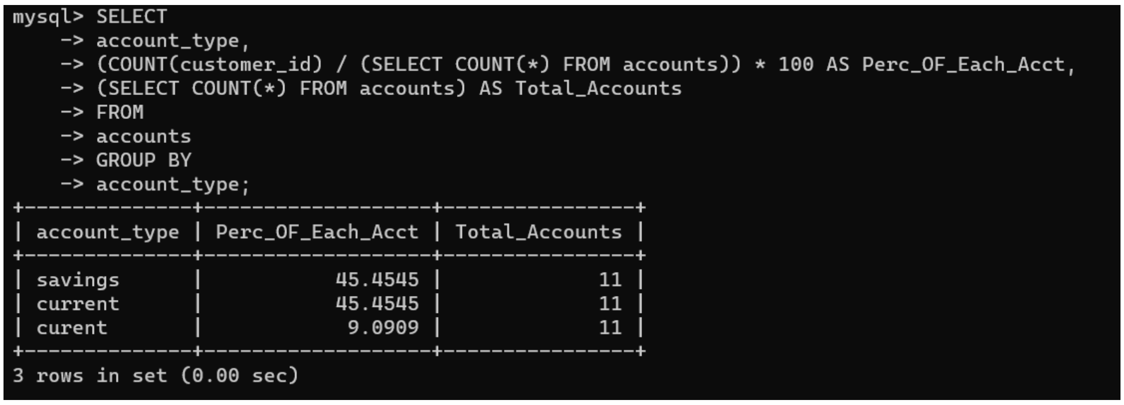


```
mysql> SELECT customer_id,CONCAT(first_name," ",last_name) AS Name
    -> FROM Customers
    -> WHERE Customer_ID = (
    -> SELECT customer_id
    -> FROM accounts
    -> GROUP BY customer_id
    -> HAVING COUNT(DISTINCT account_type) > 1);
+-------------+-----------+
| customer_id | Name      |
+-------------+-----------+
|           5 | Vijay Raj |
+-------------+-----------+
1 row in set (0.00 sec)
```



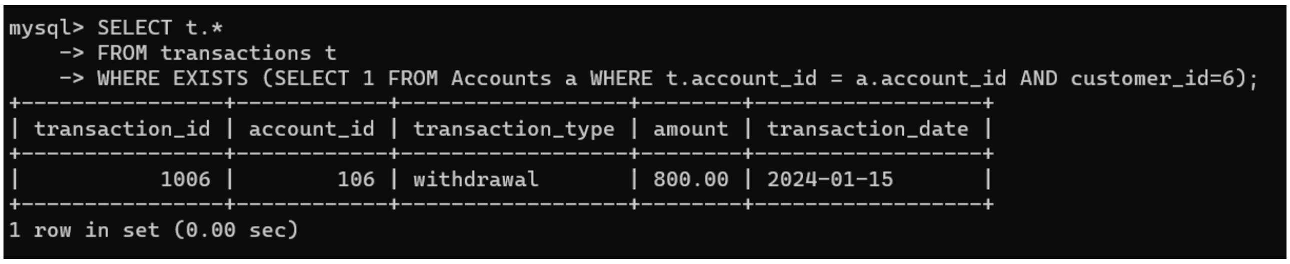
```
mysql> SELECT customer_id,CONCAT(first_name," ",last_name) AS Name
    -> FROM Customers
    -> WHERE Customer_ID = (
    -> SELECT customer_id
    -> FROM accounts
    -> GROUP BY customer_id
    -> HAVING COUNT(DISTINCT account_type) > 1);
+-------------+-----------+
| customer_id | Name      |
+-------------+-----------+
|           5 | Vijay Raj |
+-------------+-----------+
1 row 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(customer_id) / (SELECT COUNT(*) FROM accounts)) * 100 AS Perc_OF_Each_Acct,
    -> (SELECT COUNT(*) FROM accounts) AS Total_Accounts
    -> FROM
    -> accounts
    -> GROUP BY
    -> account_type;
+--------------+-------------------+----------------+
| account_type | Perc_OF_Each_Acct | Total_Accounts |
+--------------+-------------------+----------------+
| savings      |           45.4545 |             11 |
| current      |           45.4545 |             11 |
| curent       |            9.0909 |             11 |
+--------------+-------------------+----------------+
3 rows in set (0.00 sec)
```

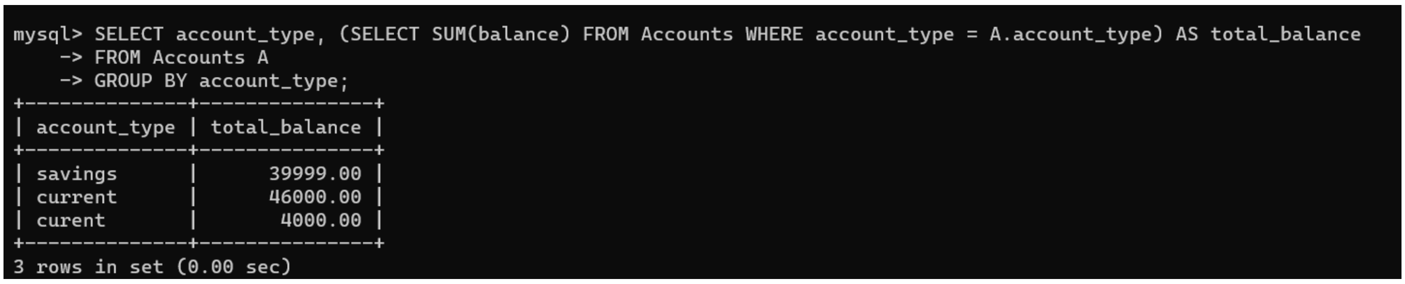
9. Retrieve all transactions for a customer with a given customer_id.



```
mysql> SELECT t.*
    -> FROM transactions t
    -> WHERE EXISTS (SELECT 1 FROM Accounts a WHERE t.account_id = a.account_id AND customer_id=6);
+----------------+------------+------------------+--------+------------------+
| transaction_id | account_id | transaction_type | amount | transaction_date |
+----------------+------------+------------------+--------+------------------+
|           1006 |        106 | withdrawal       | 800.00 | 2024-01-15       |
+----------------+------------+------------------+--------+------------------+
1 row in set (0.00 sec)
```

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

clause.



```
mysql> SELECT account_type, (SELECT SUM(balance) FROM Accounts WHERE account_type = A.account_type) AS total_balance
    -> FROM Accounts A
    -> GROUP BY account_type;
+--------------+---------------+
| account_type | total_balance |
+--------------+---------------+
| savings      |      39999.00 |
| current      |      46000.00 |
| curent       |       4000.00 |
+--------------+---------------+
3 rows in set (0.00 sec)
```