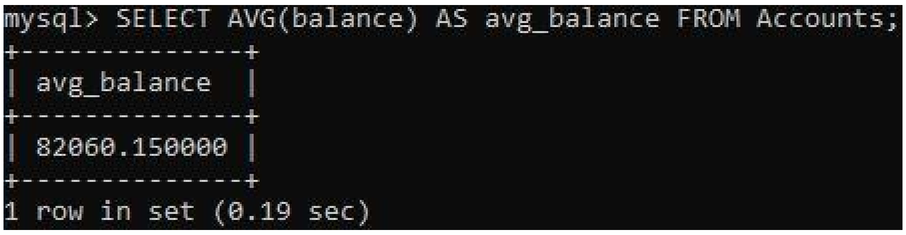
ASSIGNMENT-BANKING SYSTEM TASK- 3
Aggregate functions, Having, Order By, GroupBy and Joins

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

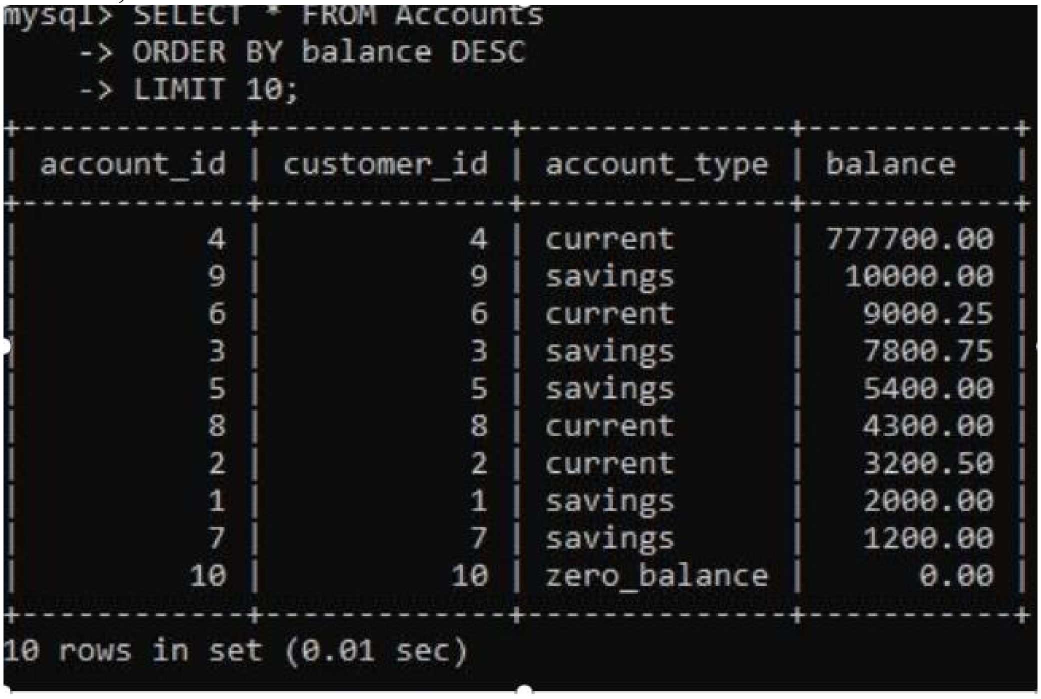
SELECT AVG(balance) AS avg_balance FROM Accounts;



```
mysql> SELECT AVG(balance) AS avg_balance FROM Accounts;
+--------------+
| avg_balance  |
+--------------+
| 82060.150000 |
+--------------+
1 row in set (0.19 sec)
```

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

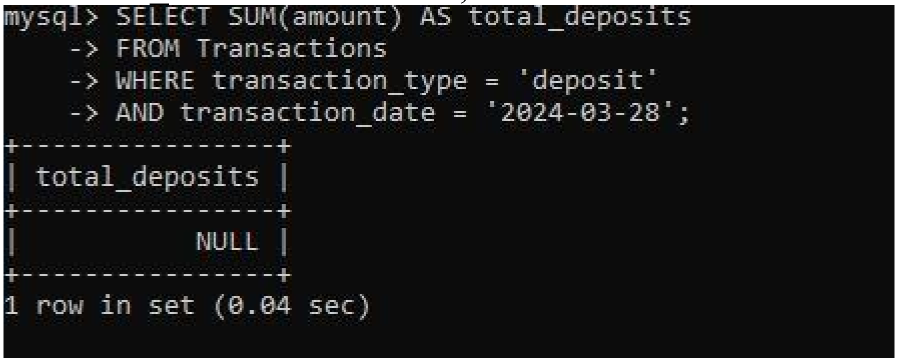
SELECT * FROM Accounts
ORDER BY balance DESC
LIMIT 10;



```
mysql> SELECT * FROM Accounts
    -> ORDER BY balance DESC
    -> LIMIT 10;
+------------+-------------+--------------+-----------+
| account_id | customer_id | account_type | balance   |
+------------+-------------+--------------+-----------+
|          4 |           4 | current      | 777700.00 |
|          9 |           9 | savings      |  10000.00 |
|          6 |           6 | current      |   9000.25 |
|          3 |           3 | savings      |   7800.75 |
|          5 |           5 | savings      |   5400.00 |
|          8 |           8 | current      |   4300.00 |
|          2 |           2 | current      |   3200.50 |
|          1 |           1 | savings      |   2000.00 |
|          7 |           7 | savings      |   1200.00 |
|         10 |          10 | zero_balance |      0.00 |
+------------+-------------+--------------+-----------+
10 rows in set (0.01 sec)
```

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

SELECT SUM(amount) AS total_deposits
FROM Transactions
WHERE transaction_type = 'deposit'
AND transaction_date = '2024-03-28';

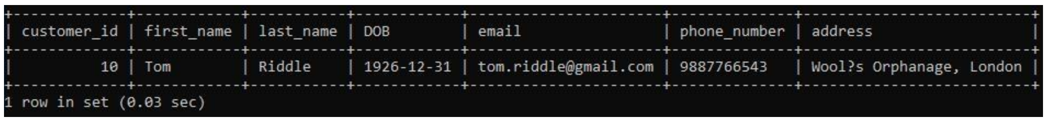


```
mysql> SELECT SUM(amount) AS total_deposits
    -> FROM Transactions
    -> WHERE transaction_type = 'deposit'
    -> AND transaction_date = '2024-03-28';
+----------------+
| total_deposits |
+----------------+
|           NULL |
+----------------+
1 row in set (0.04 sec)
```

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

For oldest Customers:

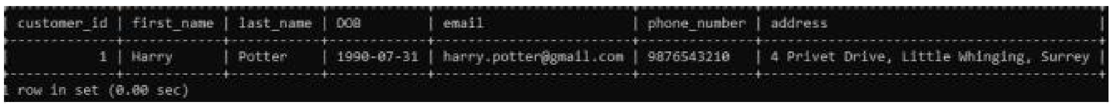
SELECT * FROM Customers
ORDER BY DOB ASC
LIMIT 1;



```
+-------------+------------+-----------+------------+----------------------+--------------+----------------------------+
| customer_id | first_name | last_name | DOB        | email                | phone_number | address                    |
+-------------+------------+-----------+------------+----------------------+--------------+----------------------------+
|          10 | Tom        | Riddle    | 1926-12-31 | tom.riddle@gmail.com | 9887766543   | Wool?s Orphanage, London |
+-------------+------------+-----------+------------+----------------------+--------------+----------------------------+
1 row in set (0.03 sec)
```

For Newest Customers:

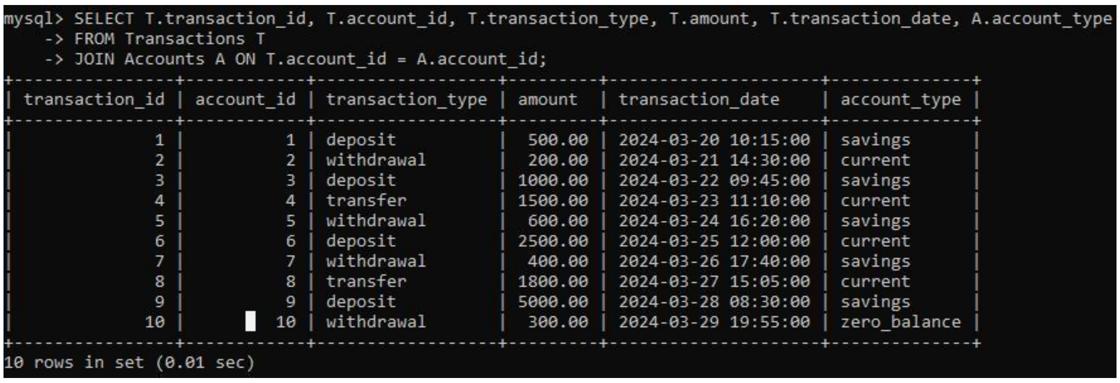
SELECT * FROM Customers
ORDER BY DOB DESC
LIMIT 1;



```
| customer_id | first_name | last_name | DOB        | email                  | phone_number | address                              |
+-------------+------------+-----------+------------+------------------------+--------------+--------------------------------------+
|           1 | Harry      | Potter    | 1990-07-31 | harry.potter@gmail.com | 9876543210   | 4 Privet Drive, Little Whinging, Surrey |
+-------------+------------+-----------+------------+------------------------+--------------+--------------------------------------+
1 row in set (0.00 sec)
```

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

SELECT T.transaction_id, T.account_id, T.transaction_type, T.amount, T.transaction_date, A.account_type
FROM Transactions T
JOIN Accounts A ON T.account_id = A.account_id;



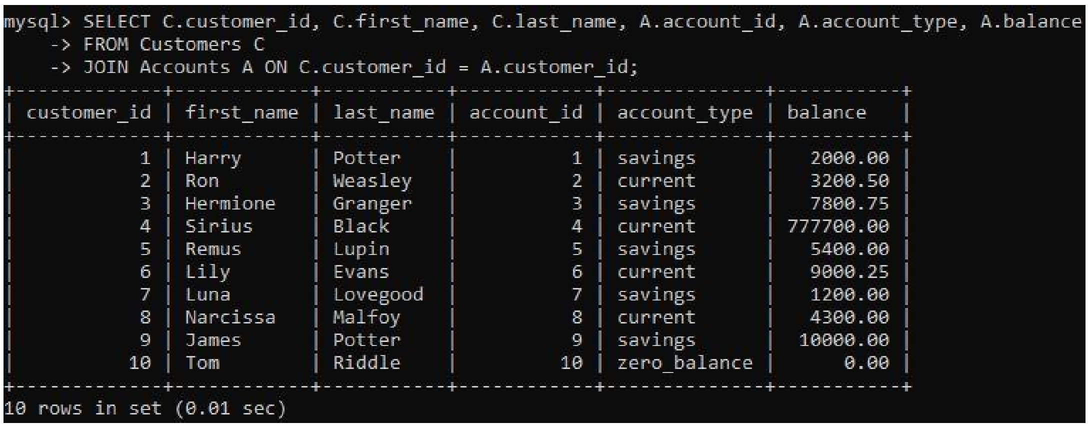
```
mysql> SELECT T.transaction_id, T.account_id, T.transaction_type, T.amount, T.transaction_date, A.account_type
    -> FROM Transactions T
    -> JOIN Accounts A ON T.account_id = A.account_id;
+----------------+------------+------------------+---------+---------------------+--------------+
| transaction_id | account_id | transaction_type | amount  | transaction_date    | account_type |
+----------------+------------+------------------+---------+---------------------+--------------+
|              1 |          1 | deposit          |  500.00 | 2024-03-20 10:15:00 | savings      |
|              2 |          2 | withdrawal       |  200.00 | 2024-03-21 14:30:00 | current      |
|              3 |          3 | deposit          | 1000.00 | 2024-03-22 09:45:00 | savings      |
|              4 |          4 | transfer         | 1500.00 | 2024-03-23 11:10:00 | current      |
|              5 |          5 | withdrawal       |  600.00 | 2024-03-24 16:20:00 | savings      |
|              6 |          6 | deposit          | 2500.00 | 2024-03-25 12:00:00 | current      |
|              7 |          7 | withdrawal       |  400.00 | 2024-03-26 17:40:00 | savings      |
|              8 |          8 | transfer         | 1800.00 | 2024-03-27 15:05:00 | current      |
|              9 |          9 | deposit          | 5000.00 | 2024-03-28 08:30:00 | savings      |
|             10 |         10 | withdrawal       |  300.00 | 2024-03-29 19:55:00 | zero_balance |
+----------------+------------+------------------+---------+---------------------+--------------+
10 rows in set (0.01 sec)
```

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

SELECT C.customer_id, C.first_name, C.last_name, A.account_id, A.account_type, A.balance

FROM Customers C

JOIN Accounts A ON C.customer_id = A.customer_id;



```
mysql> SELECT C.customer_id, C.first_name, C.last_name, A.account_id, A.account_type, A.balance
    -> FROM Customers C
    -> JOIN Accounts A ON C.customer_id = A.customer_id;
+-------------+------------+-----------+------------+--------------+-----------+
| customer_id | first_name | last_name | account_id | account_type | balance   |
+-------------+------------+-----------+------------+--------------+-----------+
|           1 | Harry      | Potter    |          1 | savings      |   2000.00 |
|           2 | Ron        | Weasley   |          2 | current      |   3200.50 |
|           3 | Hermione   | Granger   |          3 | savings      |   7800.75 |
|           4 | Sirius     | Black     |          4 | current      | 777700.00 |
|           5 | Remus      | Lupin     |          5 | savings      |   5400.00 |
|           6 | Lily       | Evans     |          6 | current      |   9000.25 |
|           7 | Luna       | Lovegood  |          7 | savings      |   1200.00 |
|           8 | Narcissa   | Malfoy    |          8 | current      |   4300.00 |
|           9 | James      | Potter    |          9 | savings      |  10000.00 |
|          10 | Tom        | Riddle    |         10 | zero_balance |      0.00 |
+-------------+------------+-----------+------------+--------------+-----------+
10 rows in set (0.01 sec)
```

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

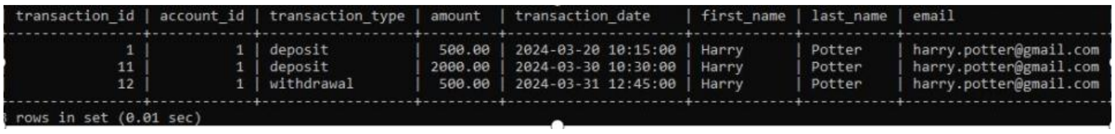
SELECT T.*, C.first_name, C.last_name, C.email

FROM Transactions T

JOIN Accounts A ON T.account_id = A.account_id

JOIN Customers C ON A.customer_id = C.customer_id

WHERE A.account_id = 1;



```
 transaction_id | account_id | transaction_type | amount  | transaction_date    | first_name | last_name | email
----------------+------------+------------------+---------+---------------------+------------+-----------+------------------------
              1 |          1 | deposit          |  500.00 | 2024-03-20 10:15:00 | Harry      | Potter    | harry.potter@gmail.com
             11 |          1 | deposit          | 2000.00 | 2024-03-30 10:30:00 | Harry      | Potter    | harry.potter@gmail.com
             12 |          1 | withdrawal       |  500.00 | 2024-03-31 12:45:00 | Harry      | Potter    | harry.potter@gmail.com
----------------+------------+------------------+---------+---------------------+------------+-----------+------------------------
 rows in set (0.01 sec)
```

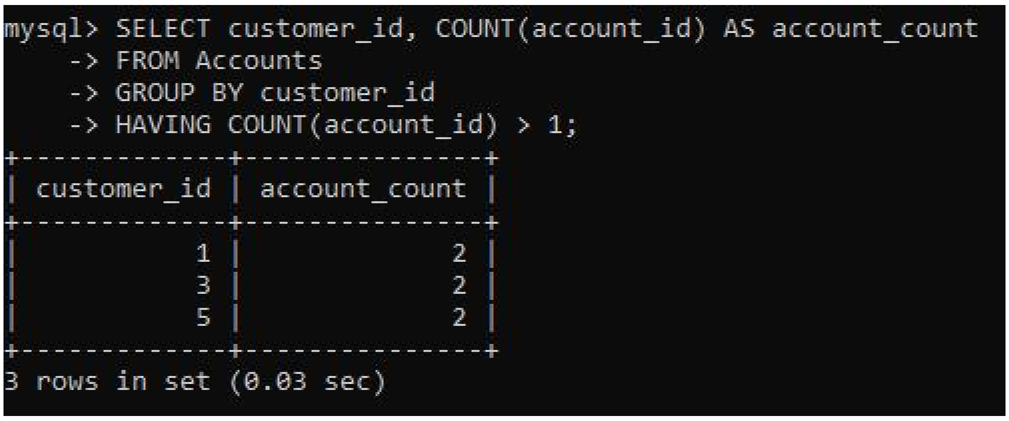
8. Write a SQL query to Identify customers who have more than one account.

SELECT customer_id, COUNT(account_id) AS account_count

FROM Accounts

GROUP BY customer_id

HAVING COUNT(account_id) > 1;;



```
mysql> SELECT customer_id, COUNT(account_id) AS account_count
    -> FROM Accounts
    -> GROUP BY customer_id
    -> HAVING COUNT(account_id) > 1;
+-------------+---------------+
| customer_id | account_count |
+-------------+---------------+
|           1 |             2 |
|           3 |             2 |
|           5 |             2 |
+-------------+---------------+
3 rows in set (0.03 sec)
```

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

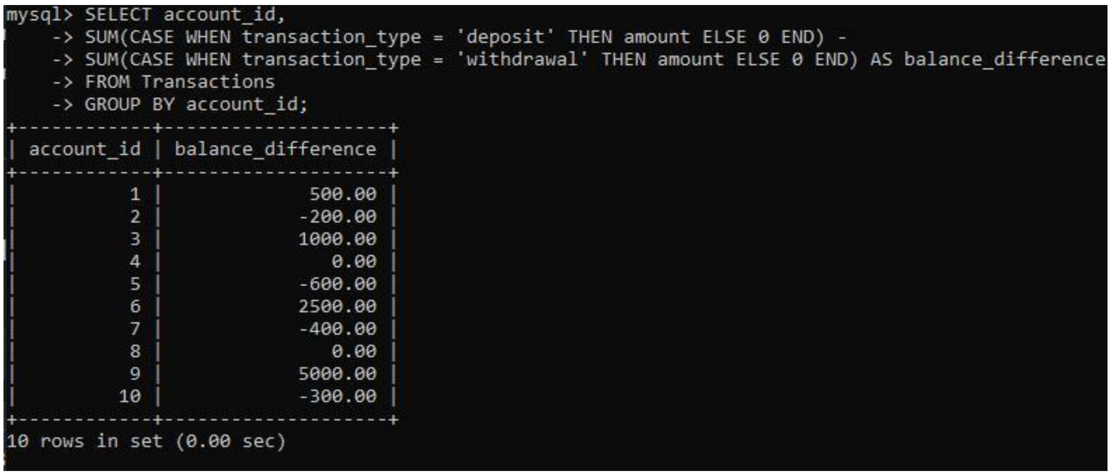
SELECT account_id,

SUM(CASE WHEN transaction_type = 'deposit' THEN amount ELSE 0 END) -

SUM(CASE WHEN transaction_type = 'withdrawal' THEN amount ELSE 0 END) AS balance_difference

FROM Transactions

GROUP BY account_id;



```
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 balance_difference
    -> FROM Transactions
    -> GROUP BY account_id;
+------------+--------------------+
| account_id | balance_difference |
+------------+--------------------+
|          1 |             500.00 |
|          2 |            -200.00 |
|          3 |            1000.00 |
|          4 |               0.00 |
|          5 |            -600.00 |
|          6 |            2500.00 |
|          7 |            -400.00 |
|          8 |               0.00 |
|          9 |            5000.00 |
|         10 |            -300.00 |
+------------+--------------------+
10 rows in set (0.00 sec)
```

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

SELECT account_id, AVG(balance) AS avg_daily_balance

FROM (

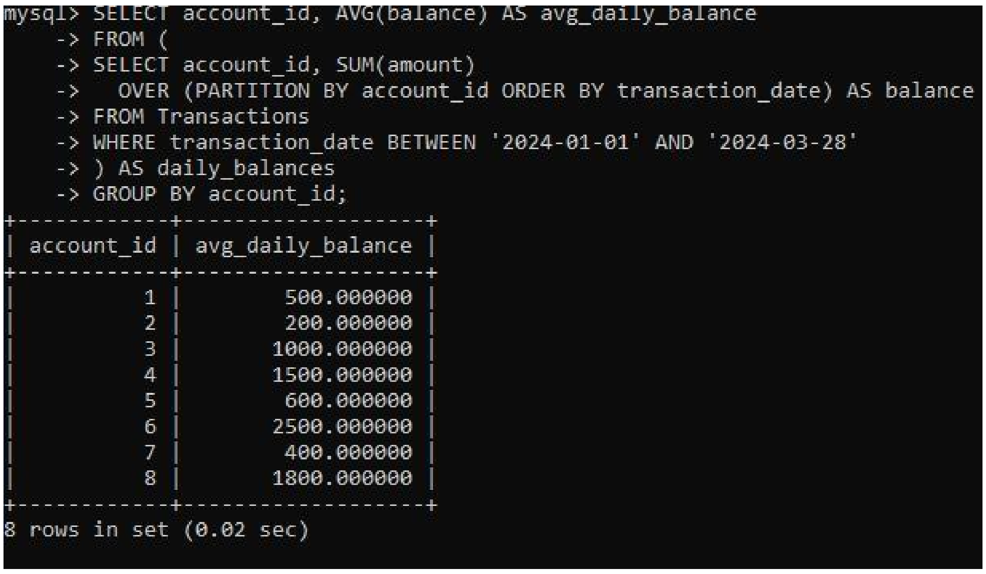
SELECT account_id, SUM(amount) OVER (PARTITION BY account_id ORDER BY transaction_date) AS balance

FROM Transactions

WHERE transaction_date BETWEEN '2024-01-01' AND '2024-03-28'

) AS daily_balances

GROUP BY account_id;



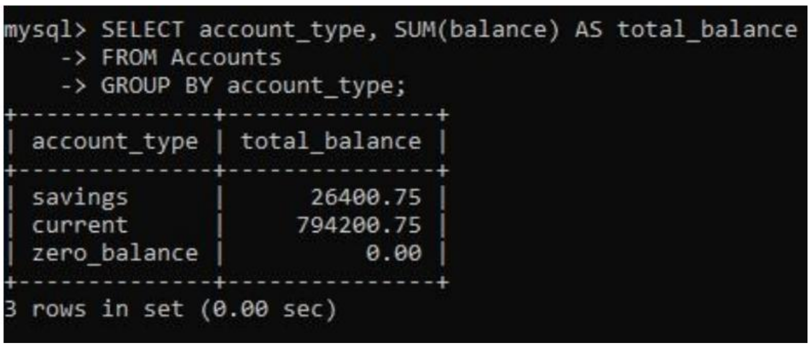
```
mysql> SELECT account_id, AVG(balance) AS avg_daily_balance
    -> FROM (
    -> SELECT account_id, SUM(amount)
    ->   OVER (PARTITION BY account_id ORDER BY transaction_date) AS balance
    -> FROM Transactions
    -> WHERE transaction_date BETWEEN '2024-01-01' AND '2024-03-28'
    -> ) AS daily_balances
    -> GROUP BY account_id;
+------------+-------------------+
| account_id | avg_daily_balance |
+------------+-------------------+
|          1 |        500.000000 |
|          2 |        200.000000 |
|          3 |       1000.000000 |
|          4 |       1500.000000 |
|          5 |        600.000000 |
|          6 |       2500.000000 |
|          7 |        400.000000 |
|          8 |       1800.000000 |
+------------+-------------------+
8 rows in set (0.02 sec)
```

11.Calculate the total balance for each account type.

SELECT account_type, SUM(balance) AS total_balance

FROM Accounts

GROUP BY account_type;



```
mysql> SELECT account_type, SUM(balance) AS total_balance
    -> FROM Accounts
    -> GROUP BY account_type;
+--------------+---------------+
| account_type | total_balance |
+--------------+---------------+
| savings      |      26400.75 |
| current      |     794200.75 |
| zero_balance |          0.00 |
+--------------+---------------+
3 rows in set (0.00 sec)
```

12.Identify accounts with the highest number of transactions order

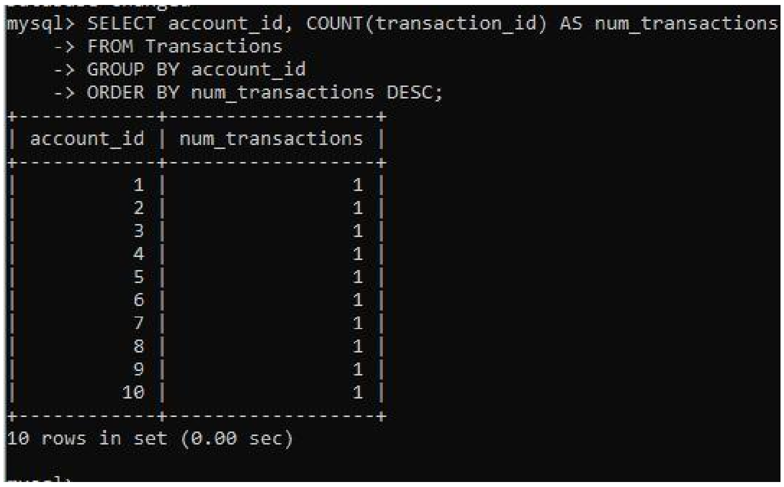
by descending order.

SELECT account_id, COUNT(transaction_id) AS num_transactions

FROM Transactions

GROUP BY account_id

ORDER BY num_transactions DESC;



```
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 |
+------------+------------------+
|          1 |                1 |
|          2 |                1 |
|          3 |                1 |
|          4 |                1 |
|          5 |                1 |
|          6 |                1 |
|          7 |                1 |
|          8 |                1 |
|          9 |                1 |
|         10 |                1 |
+------------+------------------+
10 rows in set (0.00 sec)
```

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

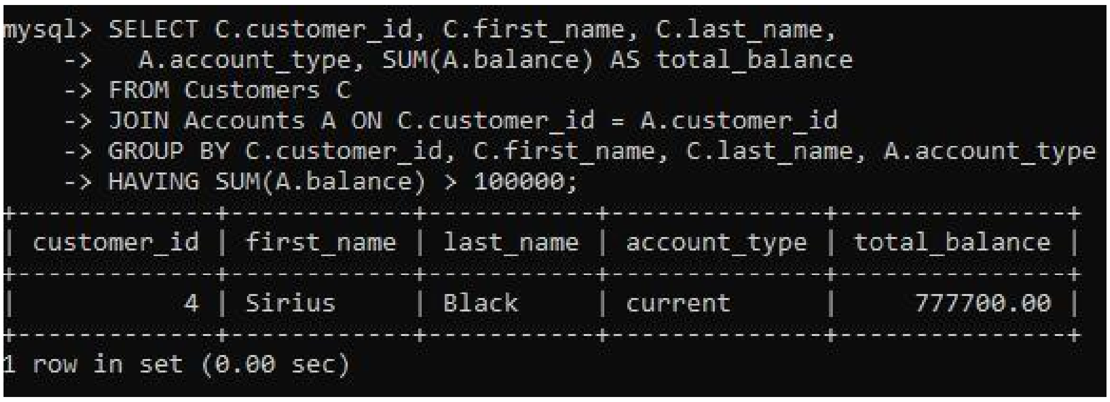
SELECT C.customer_id, C.first_name, C.last_name, A.account_type, SUM(A.balance) AS total_balance

FROM Customers C

JOIN Accounts A ON C.customer_id = A.customer_id

GROUP BY C.customer_id, C.first_name, C.last_name, A.account_type

HAVING SUM(A.balance) > 100000;



```
mysql> SELECT C.customer_id, C.first_name, C.last_name,
    ->   A.account_type, SUM(A.balance) AS total_balance
    -> FROM Customers C
    -> JOIN Accounts A ON C.customer_id = A.customer_id
    -> GROUP BY C.customer_id, C.first_name, C.last_name, A.account_type
    -> HAVING SUM(A.balance) > 100000;
+-------------+------------+-----------+--------------+---------------+
| customer_id | first_name | last_name | account_type | total_balance |
+-------------+------------+-----------+--------------+---------------+
|           4 | Sirius     | Black     | current      |     777700.00 |
+-------------+------------+-----------+--------------+---------------+
1 row in set (0.00 sec)
```

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

SELECT account_id, amount, transaction_date, COUNT(*) AS duplicate_count

FROM Transactions

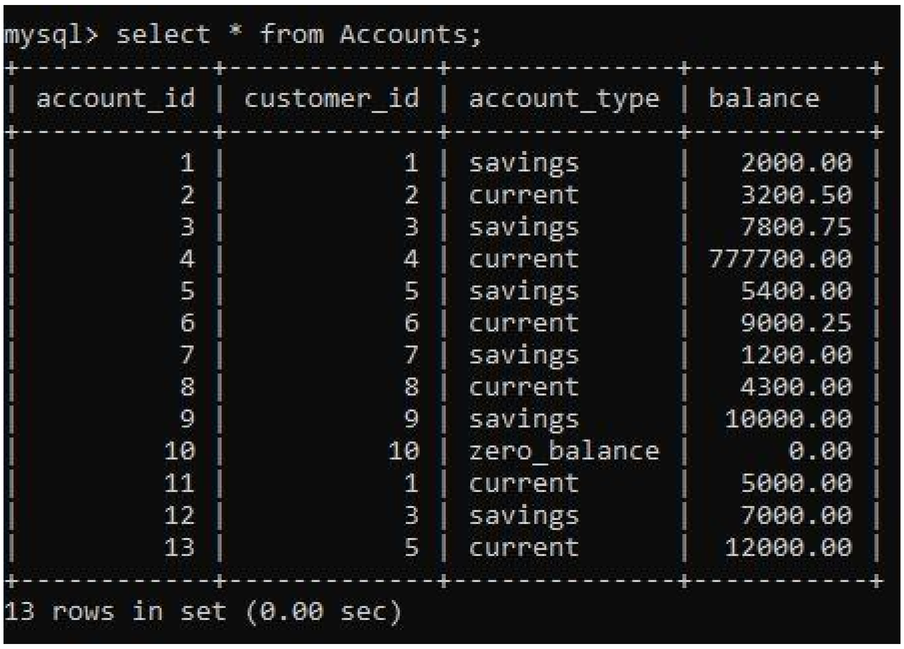
GROUP BY account_id, amount, transaction_date

HAVING COUNT(*) > 1;

Notes:

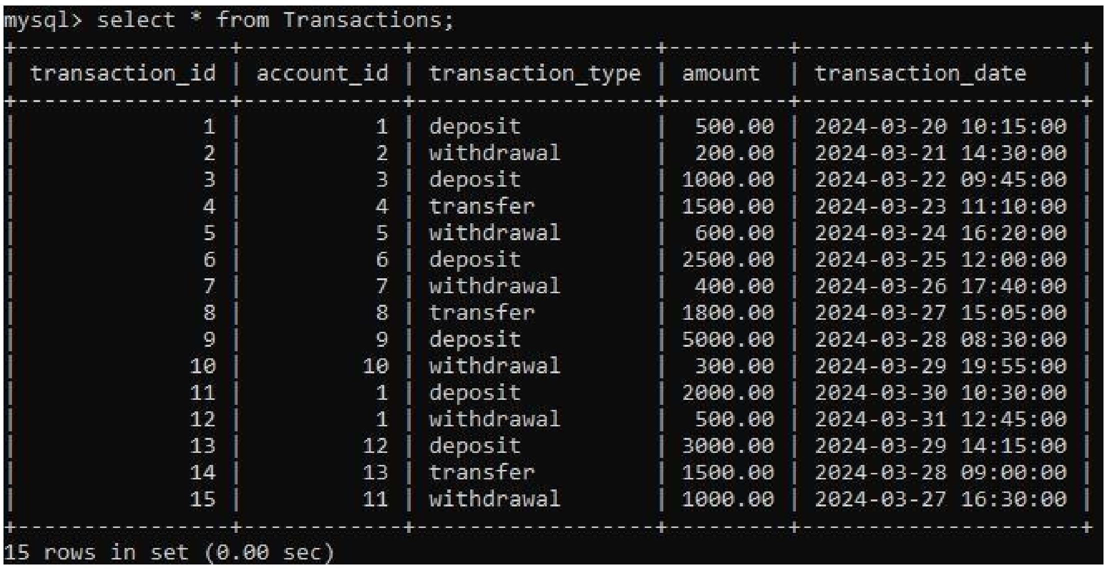
Few record have been included in both the Accounts and the Transactions tables because the join operations performed for the 7th and 8th query gave empty sets . So as to provide some proper records for the queries given,some values are inserted.

Accounts table:



```
mysql> select * from Accounts;
+------------+-------------+--------------+-----------+
| account_id | customer_id | account_type | balance   |
+------------+-------------+--------------+-----------+
|          1 |           1 | savings      |   2000.00 |
|          2 |           2 | current      |   3200.50 |
|          3 |           3 | savings      |   7800.75 |
|          4 |           4 | current      | 777700.00 |
|          5 |           5 | savings      |   5400.00 |
|          6 |           6 | current      |   9000.25 |
|          7 |           7 | savings      |   1200.00 |
|          8 |           8 | current      |   4300.00 |
|          9 |           9 | savings      |  10000.00 |
|         10 |          10 | zero_balance |      0.00 |
|         11 |           1 | current      |   5000.00 |
|         12 |           3 | savings      |   7000.00 |
|         13 |           5 | current      |  12000.00 |
+------------+-------------+--------------+-----------+
13 rows in set (0.00 sec)
```

Transactions table:



```
mysql> select * from Transactions;
+----------------+------------+------------------+---------+---------------------+
| transaction_id | account_id | transaction_type | amount  | transaction_date    |
+----------------+------------+------------------+---------+---------------------+
|              1 |          1 | deposit          |  500.00 | 2024-03-20 10:15:00 |
|              2 |          2 | withdrawal       |  200.00 | 2024-03-21 14:30:00 |
|              3 |          3 | deposit          | 1000.00 | 2024-03-22 09:45:00 |
|              4 |          4 | transfer         | 1500.00 | 2024-03-23 11:10:00 |
|              5 |          5 | withdrawal       |  600.00 | 2024-03-24 16:20:00 |
|              6 |          6 | deposit          | 2500.00 | 2024-03-25 12:00:00 |
|              7 |          7 | withdrawal       |  400.00 | 2024-03-26 17:40:00 |
|              8 |          8 | transfer         | 1800.00 | 2024-03-27 15:05:00 |
|              9 |          9 | deposit          | 5000.00 | 2024-03-28 08:30:00 |
|             10 |         10 | withdrawal       |  300.00 | 2024-03-29 19:55:00 |
|             11 |          1 | deposit          | 2000.00 | 2024-03-30 10:30:00 |
|             12 |          1 | withdrawal       |  500.00 | 2024-03-31 12:45:00 |
|             13 |         12 | deposit          | 3000.00 | 2024-03-29 14:15:00 |
|             14 |         13 | transfer         | 1500.00 | 2024-03-28 09:00:00 |
|             15 |         11 | withdrawal       | 1000.00 | 2024-03-27 16:30:00 |
+----------------+------------+------------------+---------+---------------------+
15 rows in set (0.00 sec)
```