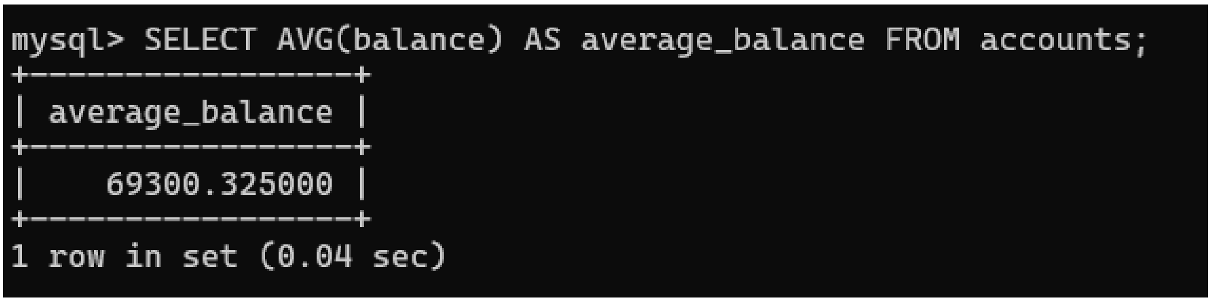
BANK SYSTEM

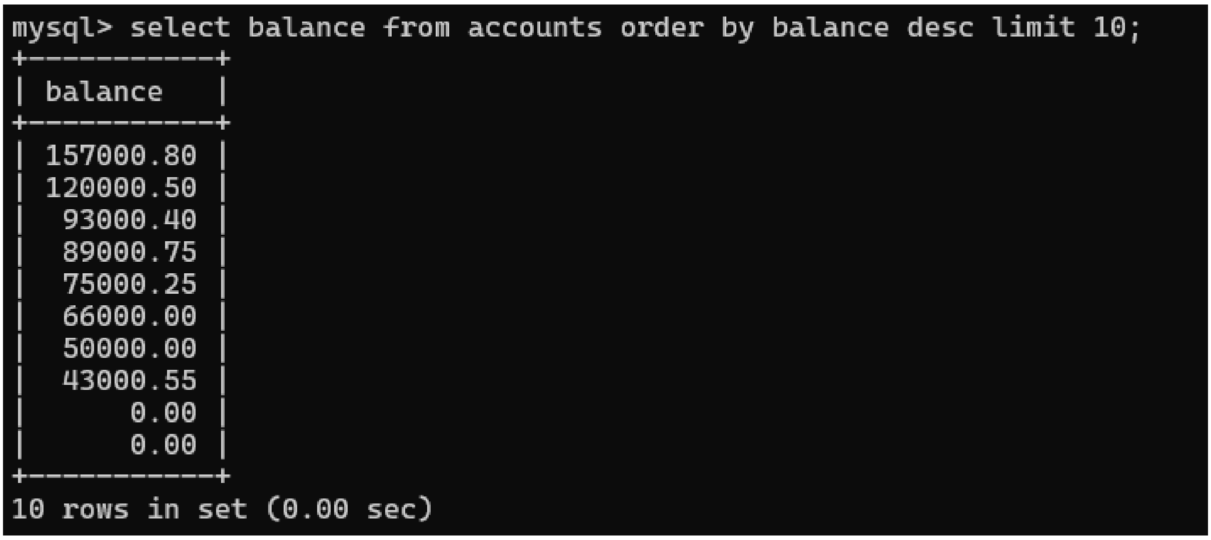
Tasks 3: Aggregate functions, Having, Order By, GroupBy and Joins:

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



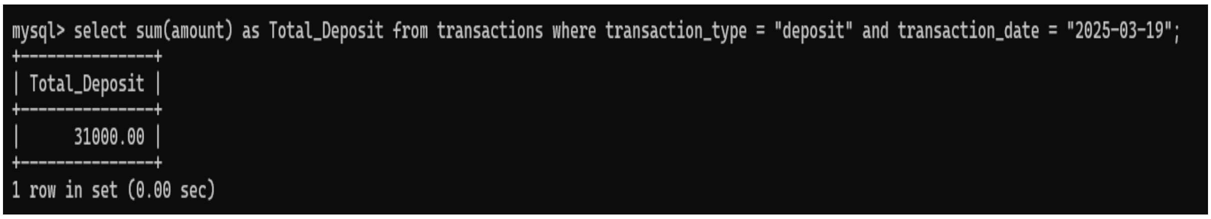
```
mysql> SELECT AVG(balance) AS average_balance FROM accounts;
+-----------------+
| average_balance |
+-----------------+
|    69300.325000 |
+-----------------+
1 row in set (0.04 sec)
```

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



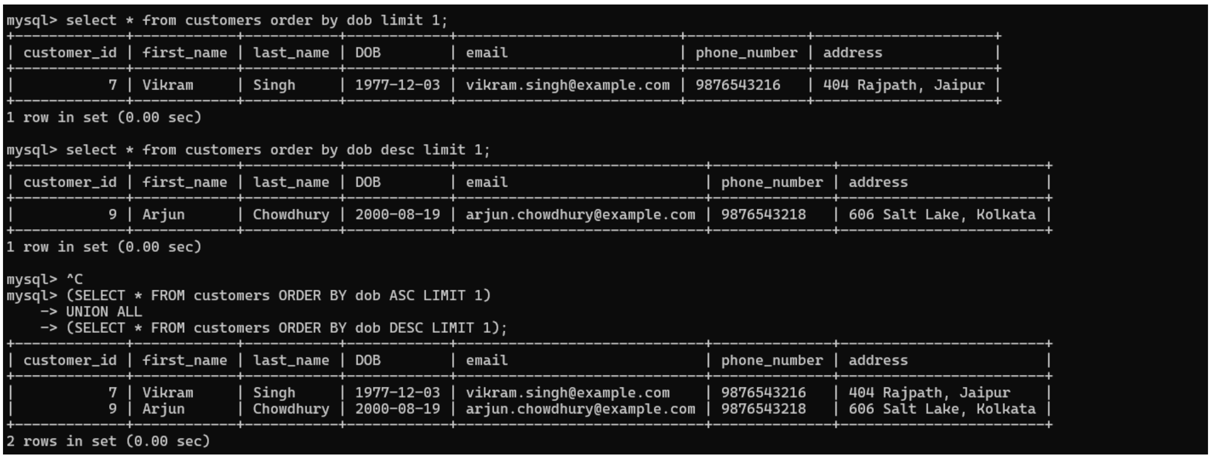
```
mysql> select balance from accounts order by balance desc limit 10;
+-----------+
| balance   |
+-----------+
| 157000.80 |
| 120000.50 |
|  93000.40 |
|  89000.75 |
|  75000.25 |
|  66000.00 |
|  50000.00 |
|  43000.55 |
|      0.00 |
|      0.00 |
+-----------+
10 rows in set (0.00 sec)
```

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



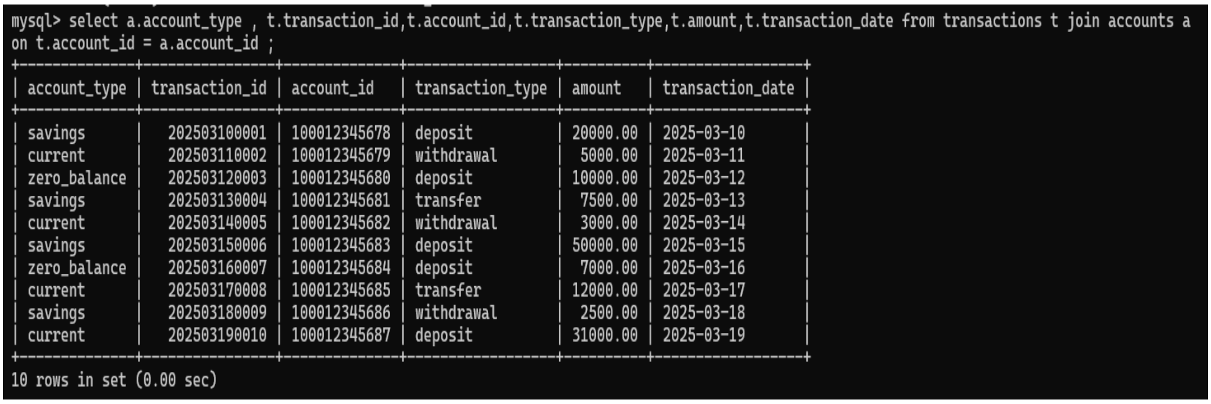
```
mysql> select sum(amount) as Total_Deposit from transactions where transaction_type = "deposit" and transaction_date = "2025-03-19";
+---------------+
| Total_Deposit |
+---------------+
|      31000.00 |
+---------------+
1 row in set (0.00 sec)
```

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



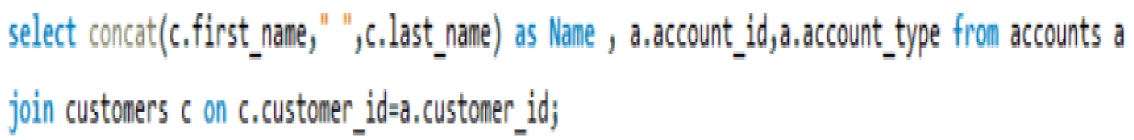
```
mysql> select * from customers order by dob limit 1;
+-------------+------------+-----------+------------+--------------------------+--------------+---------------------+
| customer_id | first_name | last_name | DOB        | email                    | phone_number | address             |
+-------------+------------+-----------+------------+--------------------------+--------------+---------------------+
|           7 | Vikram     | Singh     | 1977-12-03 | vikram.singh@example.com | 9876543216   | 404 Rajpath, Jaipur |
+-------------+------------+-----------+------------+--------------------------+--------------+---------------------+
1 row in set (0.00 sec)

mysql> select * from customers order by dob desc limit 1;
+-------------+------------+-----------+------------+-----------------------------+--------------+------------------------+
| customer_id | first_name | last_name | DOB        | email                       | phone_number | address                |
+-------------+------------+-----------+------------+-----------------------------+--------------+------------------------+
|           9 | Arjun      | Chowdhury | 2000-08-19 | arjun.chowdhury@example.com | 9876543218   | 606 Salt Lake, Kolkata |
+-------------+------------+-----------+------------+-----------------------------+--------------+------------------------+
1 row in set (0.00 sec)

mysql> ^C
mysql> (SELECT * FROM customers ORDER BY dob ASC LIMIT 1)
    -> UNION ALL
    -> (SELECT * FROM customers ORDER BY dob DESC LIMIT 1);
+-------------+------------+-----------+------------+-----------------------------+--------------+------------------------+
| customer_id | first_name | last_name | DOB        | email                       | phone_number | address                |
+-------------+------------+-----------+------------+-----------------------------+--------------+------------------------+
|           7 | Vikram     | Singh     | 1977-12-03 | vikram.singh@example.com    | 9876543216   | 404 Rajpath, Jaipur    |
|           9 | Arjun      | Chowdhury | 2000-08-19 | arjun.chowdhury@example.com | 9876543218   | 606 Salt Lake, Kolkata |
+-------------+------------+-----------+------------+-----------------------------+--------------+------------------------+
2 rows in set (0.00 sec)
```

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



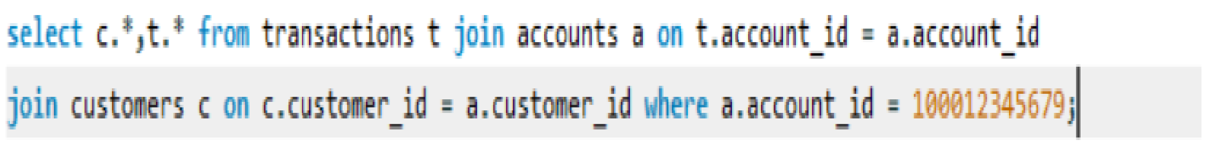
```
mysql> select a.account_type , t.transaction_id,t.account_id,t.transaction_type,t.amount,t.transaction_date from transactions t join accounts a
on t.account_id = a.account_id ;
+--------------+----------------+--------------+------------------+----------+------------------+
| account_type | transaction_id | account_id   | transaction_type | amount   | transaction_date |
+--------------+----------------+--------------+------------------+----------+------------------+
| savings      |   202503100001 | 100012345678 | deposit          | 20000.00 | 2025-03-10       |
| current      |   202503110002 | 100012345679 | withdrawal       |  5000.00 | 2025-03-11       |
| zero_balance |   202503120003 | 100012345680 | deposit          | 10000.00 | 2025-03-12       |
| savings      |   202503130004 | 100012345681 | transfer         |  7500.00 | 2025-03-13       |
| current      |   202503140005 | 100012345682 | withdrawal       |  3000.00 | 2025-03-14       |
| savings      |   202503150006 | 100012345683 | deposit          | 50000.00 | 2025-03-15       |
| zero_balance |   202503160007 | 100012345684 | deposit          |  7000.00 | 2025-03-16       |
| current      |   202503170008 | 100012345685 | transfer         | 12000.00 | 2025-03-17       |
| savings      |   202503180009 | 100012345686 | withdrawal       |  2500.00 | 2025-03-18       |
| current      |   202503190010 | 100012345687 | deposit          | 31000.00 | 2025-03-19       |
+--------------+----------------+--------------+------------------+----------+------------------+
10 rows in set (0.00 sec)
```

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



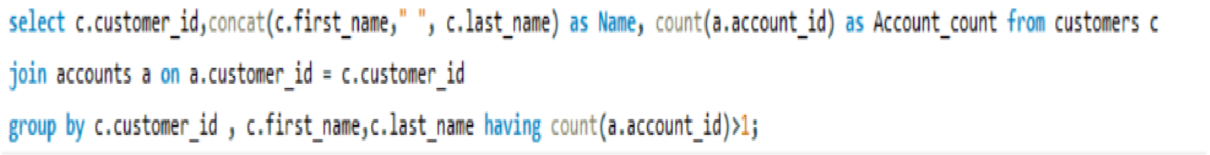
```
select concat(c.first_name," ",c.last_name) as Name , a.account_id,a.account_type from accounts a
join customers c on c.customer_id=a.customer_id;
```

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



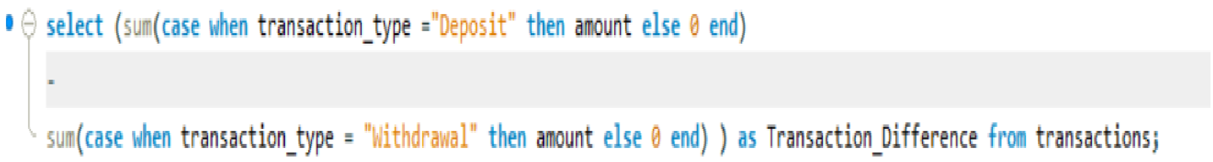
```
select c.*,t.* from transactions t join accounts a on t.account_id = a.account_id
join customers c on c.customer_id = a.customer_id where a.account_id = 100012345679;
```

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



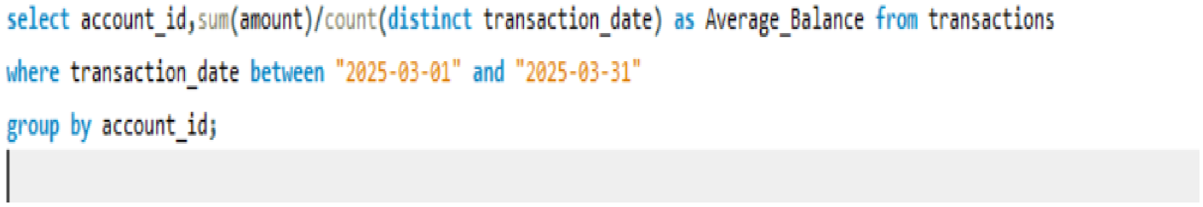
```
select c.customer_id,concat(c.first_name," ", c.last_name) as Name, count(a.account_id) as Account_count from customers c
join accounts a on a.customer_id = c.customer_id
group by c.customer_id , c.first_name,c.last_name having count(a.account_id)>1;
```

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



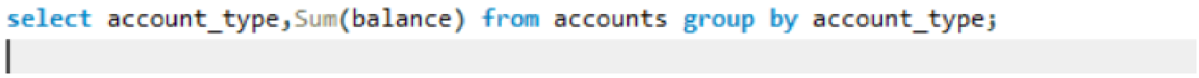
```
select (sum(case when transaction_type ="Deposit" then amount else 0 end)
-
sum(case when transaction_type = "Withdrawal" then amount else 0 end) ) as Transaction_Difference from transactions;
```

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



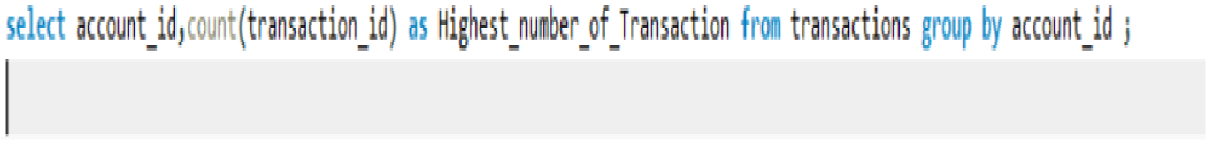
```
select account_id,sum(amount)/count(distinct transaction_date) as Average_Balance from transactions
where transaction_date between "2025-03-01" and "2025-03-31"
group by account_id;
```

11. Calculate the total balance for each account type.



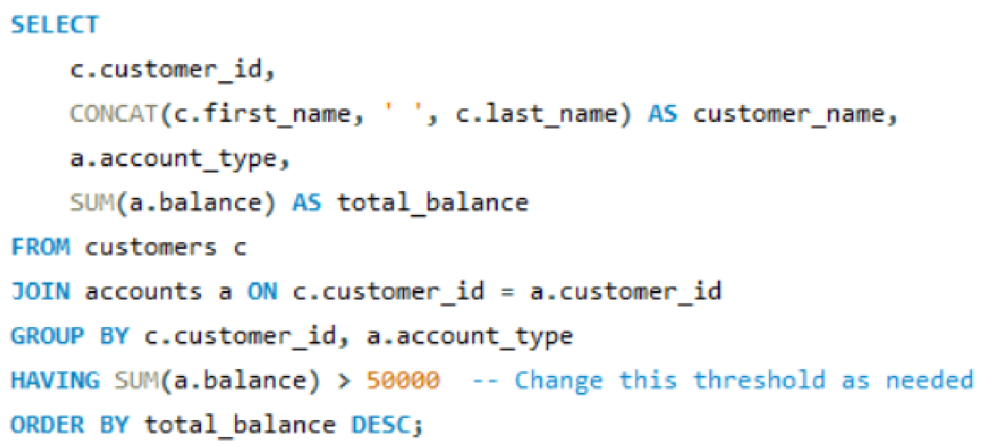
```
select account_type,Sum(balance) from accounts group by account_type;
```

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



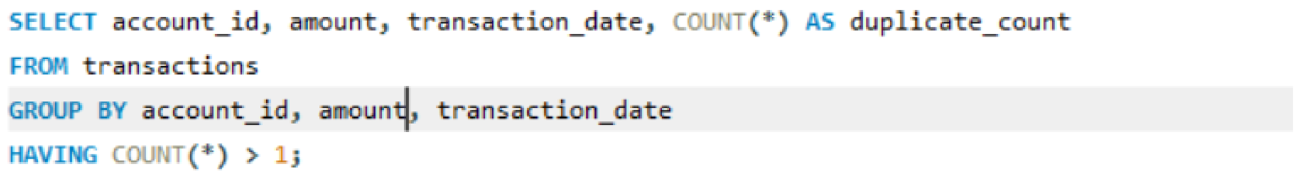
```
select account_id,count(transaction_id) as Highest_number_of_Transaction from transactions group by account_id ;
```

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



```sql
SELECT
    c.customer_id,
    CONCAT(c.first_name, ' ', c.last_name) AS customer_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, a.account_type
HAVING SUM(a.balance) > 50000  -- Change this threshold as needed
ORDER BY total_balance DESC;
```

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



```sql
SELECT account_id, amount, transaction_date, COUNT(*) AS duplicate_count
FROM transactions
GROUP BY account_id, amount, transaction_date
HAVING COUNT(*) > 1;
```