Name: Miheer Abhyankar

Assignment - Banking System

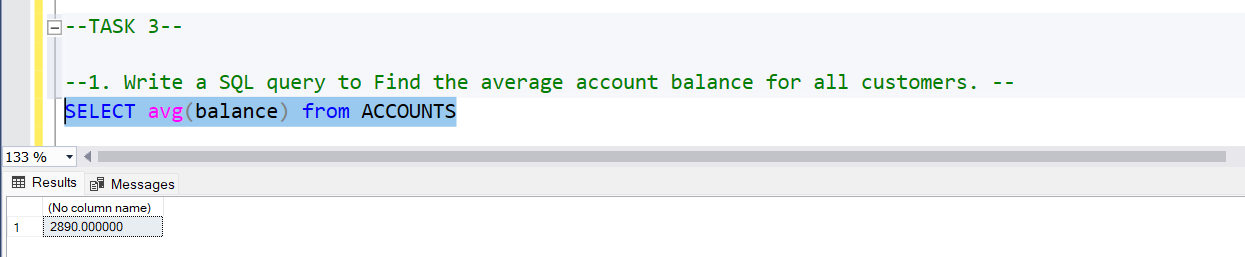
SQL

Q1 –

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

Query - SELECT avg(balance) from ACCOUNTS

Output:



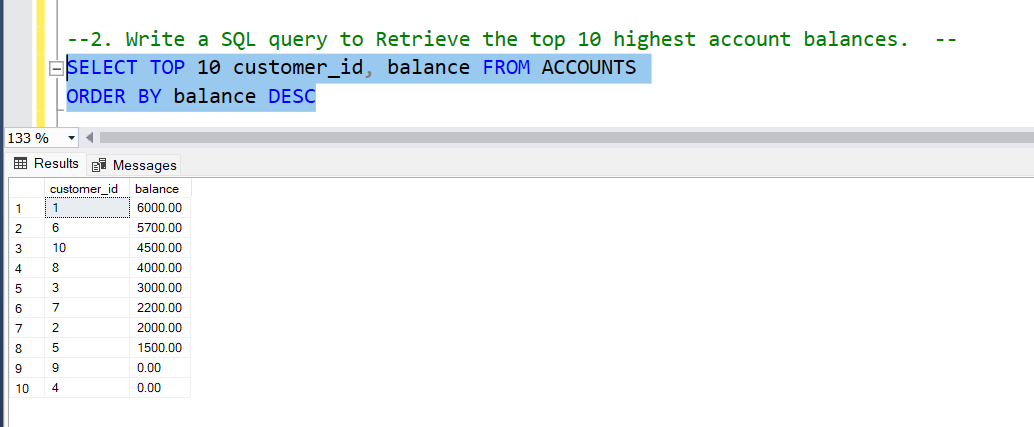
Q2 –

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

Query - SELECT TOP 10 customer\_id, balance from ACCOUNTS

ORDER BY balance DESC

Output:



Q3 - Write an SQL query to Calculate Total Deposits for All Customers in specific date.

Query –

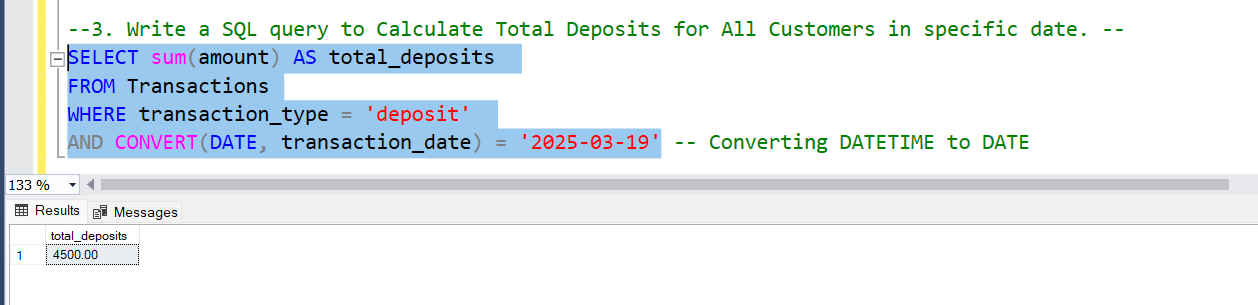
SELECT sum(amount) AS total\_deposits

FROM Transactions

WHERE transaction\_type = 'deposit'

AND CONVERT(DATE, transaction\_date) = '2025-03-19'

Output:



Q4 - Write an SQL query to Find the Oldest and Newest Customers.

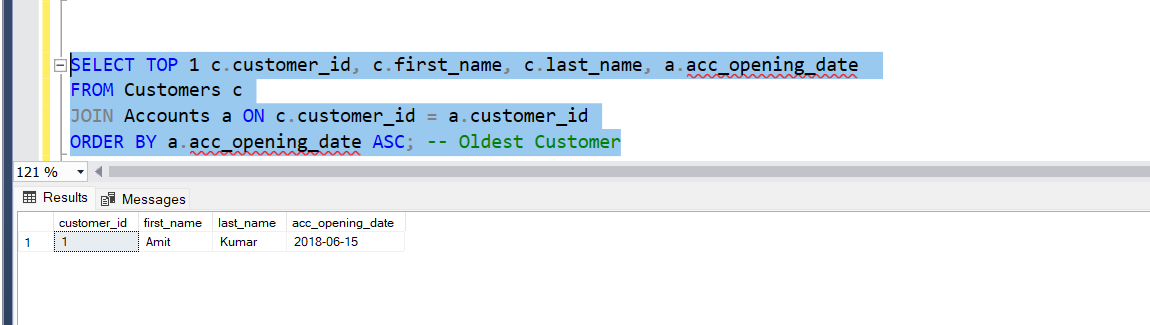
Query –

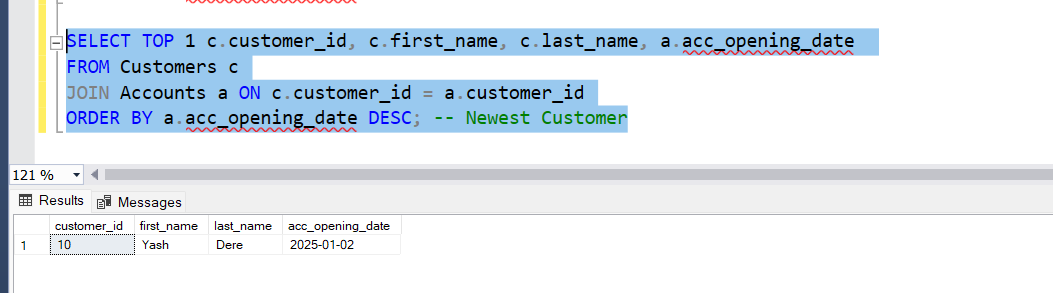
SELECT TOP 1 c.customer\_id, c.first\_name, c.last\_name, a.acc\_opening\_date

FROM Customers c

JOIN Accounts a ON c.customer\_id = a.customer\_id

Output:





Q5 - Write an SQL query to Retrieve transaction details along with the account type.

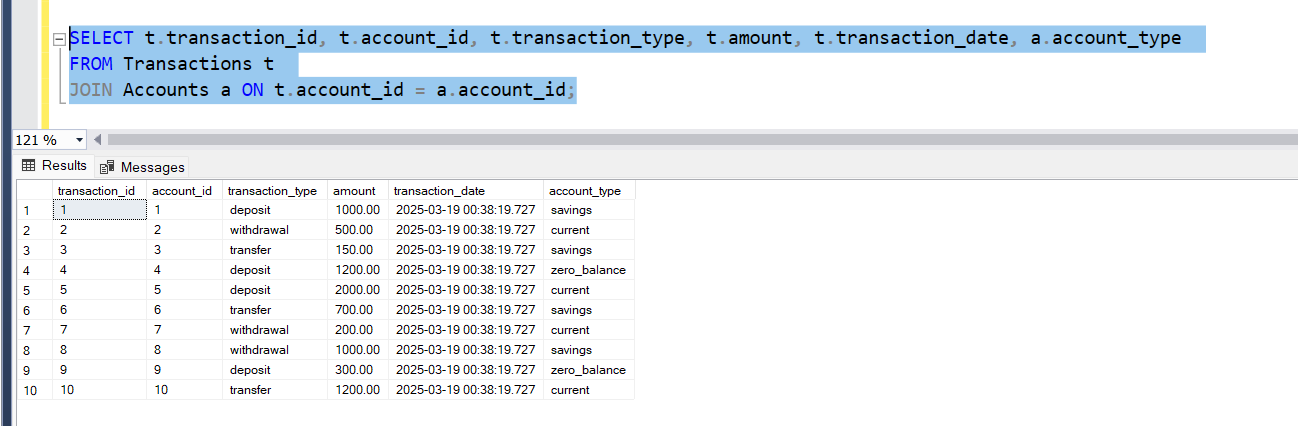
Query –

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;

Output:



Q6 - Write an SQL query to Get a list of customers along with their account details.

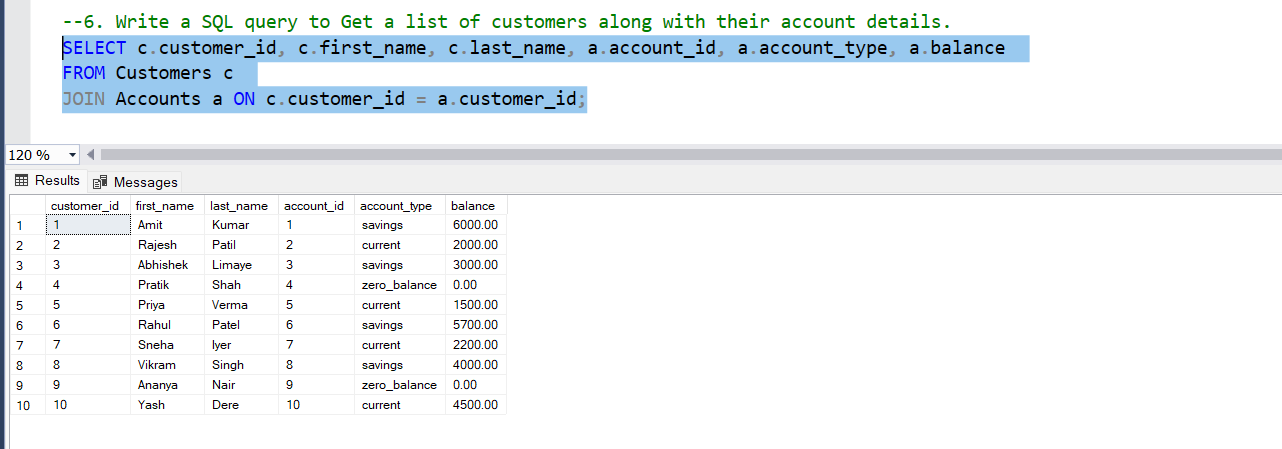
Query –

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;

Output:



Q7 - Write a SQL query to Retrieve transaction details along with customer information for a specific account.

Query –

SELECT t.transaction\_id, t.transaction\_type, t.amount, t.transaction\_date,

c.customer\_id, c.first\_name, c.last\_name, c.email

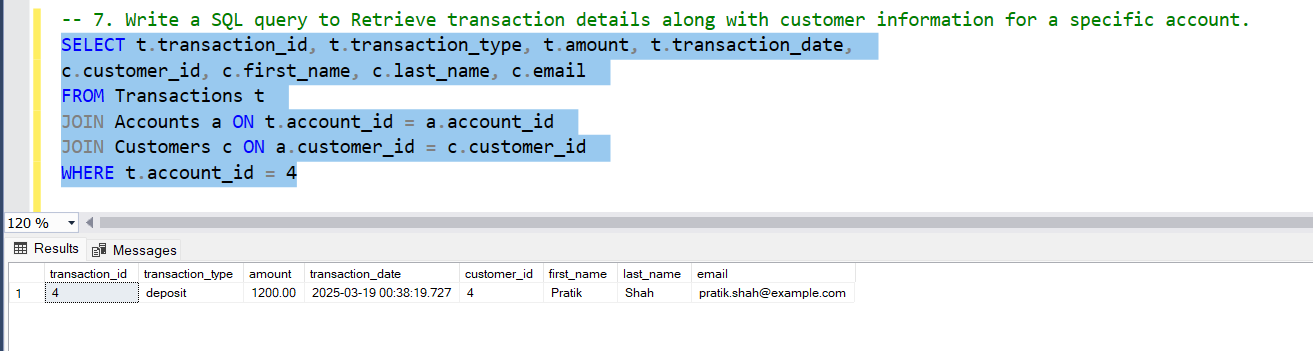
FROM Transactions t

JOIN Accounts a ON t.account\_id = a.account\_id

JOIN Customers c ON a.customer\_id = c.customer\_id

WHERE t.account\_id = 4

Output:



Q8 - Write a SQL query to Identify customers who have more than one account.

Query –

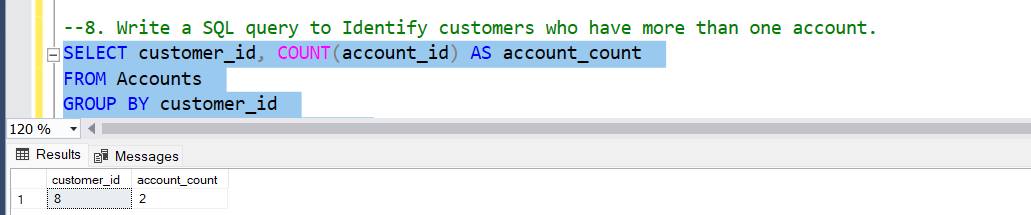
SELECT customer\_id, COUNT(account\_id) AS account\_count

FROM Accounts

GROUP BY customer\_id

HAVING COUNT(account\_id) > 1;

Output:



Q9 - Write a SQL query to Calculate the difference in transaction amounts between deposits and withdrawals.

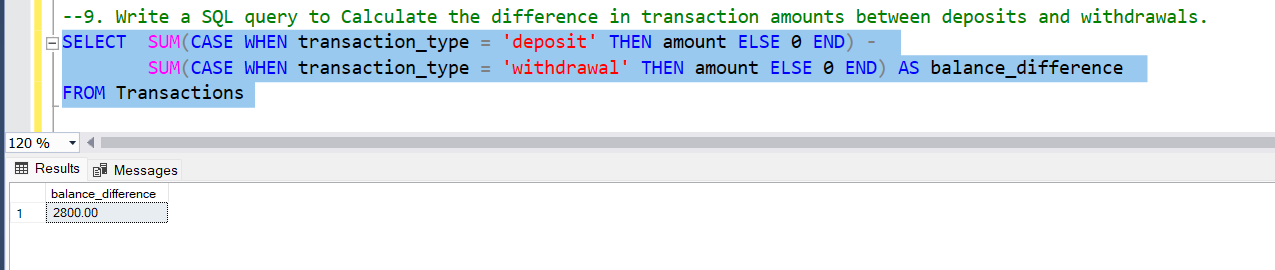
Query –

SELECT 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

Output:



Q10 - Write a SQL query to Calculate the average daily balance for each account over a specified period.

Query –

SELECT account\_id, AVG(balance) AS avg\_daily\_balance

FROM Accounts

WHERE account\_id IN (

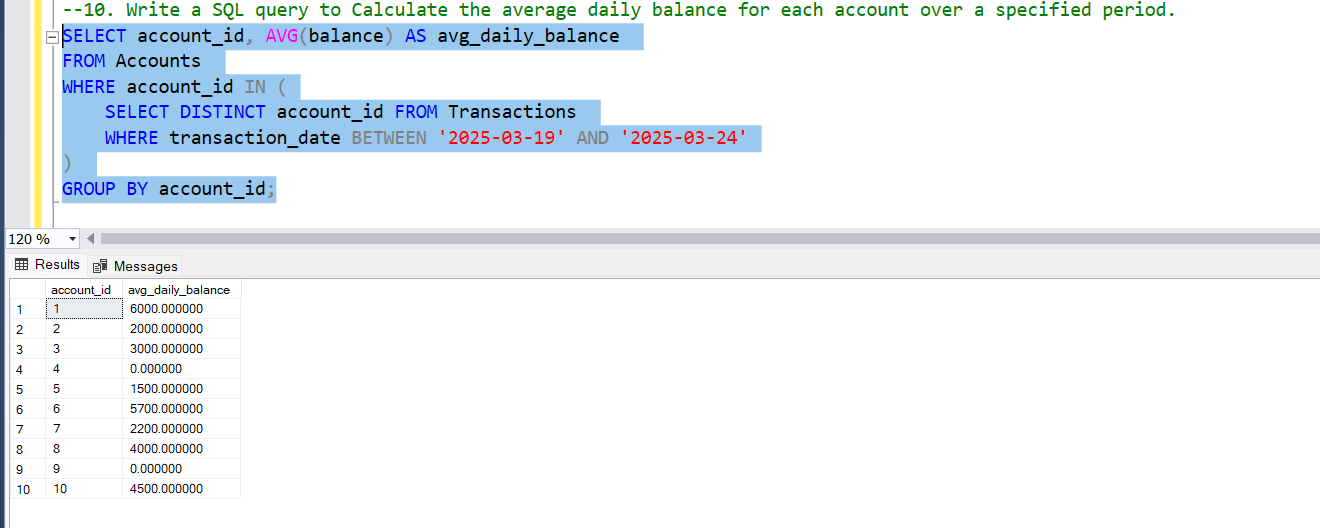
SELECT DISTINCT account\_id FROM Transactions

WHERE transaction\_date BETWEEN '2025-03-19' AND '2025-03-24'

)

GROUP BY account\_id;

Output:



Q11 - Calculate the total balance for each account type.

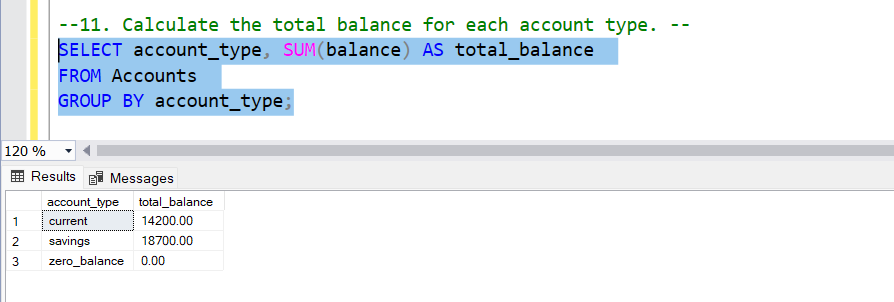
Query –

SELECT account\_type, SUM(balance) AS total\_balance

FROM Accounts

GROUP BY account\_type;

Output:



Q12 - Identify accounts with the highest number of transactions order by descending order.

Query –

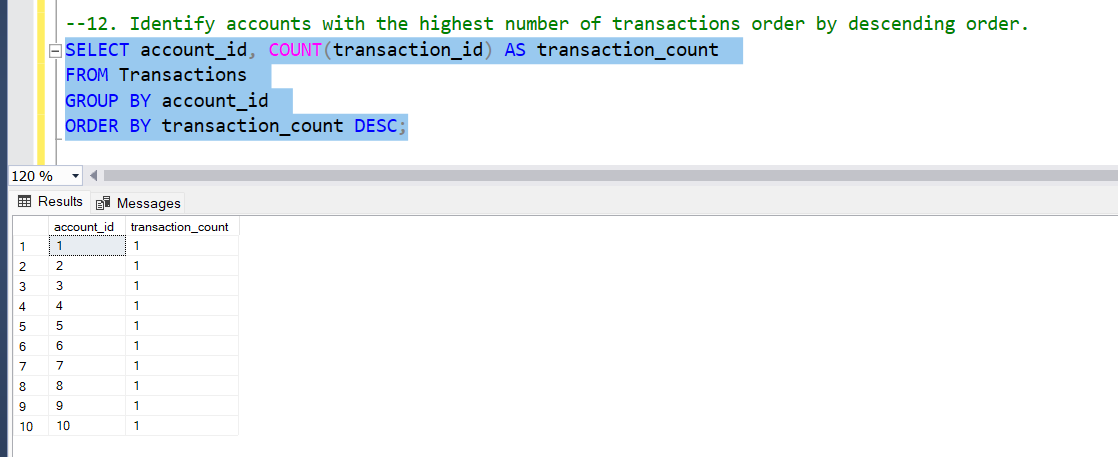
SELECT account\_id, COUNT(transaction\_id) AS transaction\_count

FROM Transactions

GROUP BY account\_id

ORDER BY transaction\_count DESC;

Output:



Q13 - List customers with high aggregate account balances, along with their account types.

Query –

SELECT c.customer\_id, c.first\_name, c.last\_name, a.account\_type, SUM(a.balance) AS total\_balance

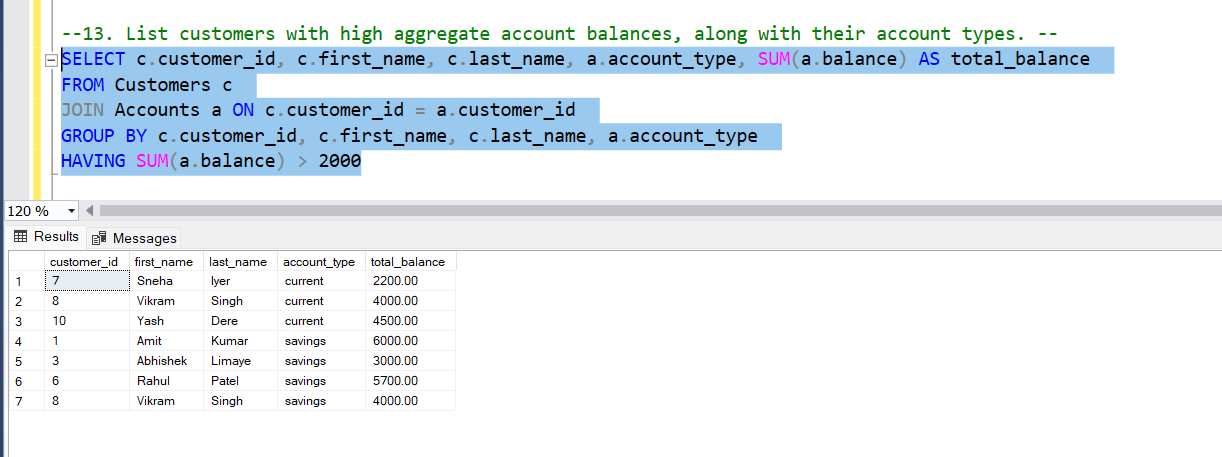
FROM Customers c

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) > 2000

Output:



Q14 - Identify and list duplicate transactions based on transaction amount, date, and account.

Query –

SELECT account\_id, transaction\_date, amount, COUNT(\*) AS duplicate\_count

FROM Transactions

GROUP BY account\_id, transaction\_date, amount

