**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 Average FROM Accounts;

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

Select\* from Accounts order by balance desc limit 10 ;

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

Select sum(amount)

From transactions

where transaction\_type="deposit" and transaction\_date="2025-03-09" ;

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

SELECT \* FROM Customers ORDER BY DOB DESC LIMIT 1;

SELECT \* FROM Customers ORDER BY DOB ASC LIMIT 1;

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

select T.\*,A.account\_type

from transactions T

left join Accounts A on T.account\_id=A.account\_id;

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

Select C.first\_name,C.last\_name,A.account\_id,A.account\_type,A.balance

from Customers C

LEFT JOIN Accounts A on C.customer\_id=A.customer\_id;

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

select T.\*, C.first\_name,C.last\_name

from Transactions T

INNER JOIN Accounts A on A.account\_id =T.account\_id

INNER JOIN Customers C on A.customer\_id=C.customer\_id

where T.Account\_id=201;

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

SELECT customer\_id, COUNT(\*) FROM Accounts

GROUP BY customer\_id

HAVING COUNT(\*) > 1;

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

SELECT account\_id,

SUM(IF(transaction\_type = 'deposit', amount, 0)) -

SUM(IF(transaction\_type = 'withdrawal', amount, 0)) AS balance\_difference

FROM Transactions

GROUP BY account\_id;

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

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

FROM Accounts

GROUP BY account\_id;

1. **Calculate the total balance for each account type.**

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

FROM Accounts

GROUP BY account\_type;

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

SELECT account\_id, COUNT(\*) as Transaction\_count

FROM Transactions

GROUP BY account\_id

ORDER BY COUNT(\*) DESC;

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

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

FROM Customers

JOIN Accounts ON Customers.customer\_id = Accounts.customer\_id

GROUP BY first\_name, last\_name, account\_type

HAVING total\_balance > 50000;

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

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

FROM Transactions

GROUP BY account\_id, amount, transaction\_date

HAVING COUNT(\*) > 1;