**ASSIGNMENT -1 (BANKING 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**.

1.select avg(balance) as avg\_bal from accounts;

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

2. select balance from accounts order by balance desc Limit 10;

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

3. SELECT SUM(amount) AS total\_deposits FROM transactions

WHERE transaction\_type = 'deposit' AND transaction\_date = '2025-03-08';

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

4. SELECT \* FROM customers

WHERE DOB = (SELECT MAX(DOB) FROM customers) OR DOB = (SELECT MIN(DOB) FROM customers);

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

5. 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;

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

6. 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;

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

7. 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 WHERE a.account\_id = 1;

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

8. SELECT customer\_id, COUNT(account\_id) AS tot\_acc

FROM accounts GROUP BY customer\_id HAVING COUNT(account\_id) > 1;

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

9. 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 net\_diff

FROM transactions GROUP BY account\_id;

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

10. SELECT t.account\_id, AVG(a.balance) AS avg\_daily

FROM transactions t JOIN accounts a ON t.account\_id = a.account\_id

WHERE t.transaction\_date BETWEEN '2025-03-01' AND '2025-03-26'

GROUP BY t.account\_id;

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

11. SELECT account\_type, SUM(balance) AS tot\_bal

FROM accounts GROUP BY account\_type;

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

12. SELECT account\_id, COUNT(transaction\_id) AS tot\_transactions

FROM transactions GROUP BY account\_id ORDER BY tot\_transactions DESC;

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

13. 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, a.account\_type HAVING SUM(a.balance) > 1000;

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

14. SELECT account\_id, amount, transaction\_date, COUNT(\*) AS dup

FROM transactions

GROUP BY account\_id, amount, transaction\_date

HAVING COUNT(\*) > 1;