Tasks 4: Subquery and its type:

1. Retrieve the customer(s) with the highest account balance.

mysql> SELECT c.customer\_id, c.first\_name, c.last\_name, a.balance

-> FROM Customers c

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

-> WHERE a.balance = (SELECT MAX(balance) FROM Accounts);

+-------------+------------+-----------+----------+

| customer\_id | first\_name | last\_name | balance |

+-------------+------------+-----------+----------+

| 6 | Ananya | Singh | 12000.00 |

+-------------+------------+-----------+----------+

1 row in set (0.00 sec)

2. Calculate the average account balance for customers who have more than one account.

mysql> SELECT customer\_id, AVG(balance) AS average\_balance

-> FROM Accounts

-> WHERE customer\_id IN (

-> SELECT customer\_id

-> FROM Accounts

-> GROUP BY customer\_id

-> HAVING COUNT(account\_id) > 1

-> )

-> GROUP BY customer\_id;

+-------------+-----------------+

| customer\_id | average\_balance |

+-------------+-----------------+

| 1 | 3750.000000 |

| 6 | 8500.000000 |

+-------------+-----------------+

2 rows in set (0.00 sec)

3. Retrieve accounts with transactions whose amounts exceed the average transaction amount.

mysql> SELECT t.transaction\_id, t.account\_id, t.transaction\_type, t.amount

-> FROM Transactions t

-> WHERE t.amount > (SELECT AVG(amount) FROM Transactions);

+----------------+------------+------------------+---------+

| transaction\_id | account\_id | transaction\_type | amount |

+----------------+------------+------------------+---------+

| 206 | 106 | deposit | 5000.00 |

| 207 | 107 | withdrawal | 3000.00 |

| 208 | 108 | deposit | 2500.00 |

| 209 | 109 | deposit | 4000.00 |

+----------------+------------+------------------+---------+

4 rows in set (0.00 sec)

4. Identify customers who have no recorded transactions.

mysql> SELECT c.customer\_id, c.first\_name, c.last\_name

-> FROM Customers c

-> WHERE c.customer\_id NOT IN (SELECT DISTINCT a.customer\_id FROM Accounts a

-> JOIN Transactions t ON a.account\_id = t.account\_id);

+-------------+------------+-----------+

| customer\_id | first\_name | last\_name |

+-------------+------------+-----------+

| 8 | Meera | Nair |

| 9 | Karan | Kapoor |

| 10 | Pooja | Mehta |

+-------------+------------+-----------+

3 rows in set (0.00 sec)

5. Calculate the total balance of accounts with no recorded transactions.

mysql> SELECT SUM(a.balance) AS total\_balance

-> FROM Accounts a

-> WHERE a.account\_id NOT IN (SELECT DISTINCT account\_id FROM Transactions);

+---------------+

| total\_balance |

+---------------+

| NULL |

+---------------+

1 row in set (0.00 sec)

6. Retrieve transactions for accounts with the lowest balance.

mysql> SELECT t.transaction\_id, t.account\_id, t.transaction\_type, t.amount

-> FROM Transactions t

-> WHERE t.account\_id IN (SELECT account\_id FROM Accounts WHERE balance = (SELECT MIN(balance) FROM Accounts));

+----------------+------------+------------------+---------+

| transaction\_id | account\_id | transaction\_type | amount |

+----------------+------------+------------------+---------+

| 204 | 104 | deposit | 1500.00 |

+----------------+------------+------------------+---------+

1 row in set (0.00 sec)

7. Identify customers who have accounts of multiple types.

mysql> SELECT customer\_id

-> FROM Accounts

-> GROUP BY customer\_id

-> HAVING COUNT(DISTINCT account\_type) > 1;

+-------------+

| customer\_id |

+-------------+

| 1 |

| 6 |

+-------------+

2 rows in set (0.00 sec)

8. Calculate the percentage of each account type out of the total number of accounts.

mysql> SELECT account\_type, COUNT(\*) AS count\_accounts,(COUNT(\*) \* 100 / (SELECT COUNT(\*) FROM Accounts)) AS percentage

-> FROM Accounts

-> GROUP BY account\_type;

+--------------+----------------+------------+

| account\_type | count\_accounts | percentage |

+--------------+----------------+------------+

| savings | 4 | 40 |

| current | 4 | 40 |

| zero\_balance | 1 | 10 |

+--------------+----------------+------------+

3 rows in set (0.00 sec)

9. Retrieve all transactions for a customer with a given customer\_id.

mysql> SELECT t.\*

-> FROM Transactions t

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

-> WHERE a.customer\_id = 1;

+----------------+------------+------------------+---------+---------------------+

| transaction\_id | account\_id | transaction\_type | amount | transaction\_date |

+----------------+------------+------------------+---------+---------------------+

| 201 | 101 | deposit | 1000.00 | 2024-03-10 10:30:00 |

| 202 | 102 | withdrawal | 500.00 | 2024-03-11 11:00:00 |

+----------------+------------+------------------+---------+---------------------+

2 rows in set (0.00 sec)

10. Calculate the total balance for each account type, including a subquery within the SELECT clause.

mysql> SELECT account\_type, (SELECT SUM(balance) FROM Accounts a2 WHERE a2.account\_type = a1.account\_type) AS total\_balance

-> FROM Accounts a1

-> GROUP BY account\_type;

+--------------+---------------+

| account\_type | total\_balance |

+--------------+---------------+

| savings | 35500.00 |

| current | 11500.00 |

| zero\_balance | 0.00 |

+--------------+---------------+

3 rows in set (0.00 sec)