

Tasks 4: Subquery and its type:

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

```
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
▶	106	Freddy	Salt	500000.00

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

```
SELECT customer_id, AVG(balance) AS average_balance FROM Accounts GROUP BY customer_id HAVING COUNT(account_id) > 1;
```

	customer_id	average_balance

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

```
SELECT account_id, transaction_type, amount FROM Transactions WHERE amount > (SELECT AVG(amount) FROM Transactions);
```

	account_id	transaction_type	amount
▶	2225	transfer	25000.00
	2226	deposit	50000.00
	2230	deposit	75000.00

4. Identify customers who have no recorded transactions.

```
SELECT a.account_id, first_name from customers c JOIN accounts a on a.customer_id=c.customer_id where not exists (select 1 from  
transactions t where t.account_id=a.account_id);
```

	account_id	first_name

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

```
SELECT SUM(balance) AS total_balance FROM Accounts a WHERE NOT EXISTS ( SELECT 1 FROM Transactions t WHERE a.account_id  
= t.account_id );
```

	total_balance
▶	NULL

6. Retrieve transactions for accounts with the lowest balance.

```
SELECT T.* FROM Transactions t JOIN Accounts a ON t.account_id = a.account_id WHERE a.balance = (SELECT MIN(balance) FROM  
Accounts);
```

	transaction_id	account_id	transaction_type	amount	transaction_date
▶	5003	2223	transfer	5500.00	2025-03-18

7. Identify customers who have accounts of multiple types.

```
SELECT customer_id FROM Accounts GROUP BY customer_id  
HAVING COUNT(DISTINCT account_type) > 1;
```

customer_id

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

```
SELECT account_type, COUNT(account_id) * 100.0 / (SELECT COUNT(*) FROM Accounts) AS percentage FROM Accounts  
GROUP BY account_type;
```

	account_type	percentage
▶	Savings	40.00000
	Current	30.00000
	Zero_balance	30.00000

9. Retrieve all transactions for a customer with a given customer_id.

```
SELECT t.*,c.customer_id FROM Customers c JOIN Accounts a ON a.customer_id= c.customer_id join transactions t on  
t.account_id=a.account_id where c.customer_id=101;
```

	transaction_id	account_id	transaction_type	amount	transaction_date	customer_id
▶	5001	2221	deposit	1000.00	2025-01-10	101

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

```
select account_type,sum(balance) as total_balance from Accounts group by account_type;
```

	account_type	total_balance
▶	Savings	660000.00
	Current	92000.00
	Zero_balance	71600.00