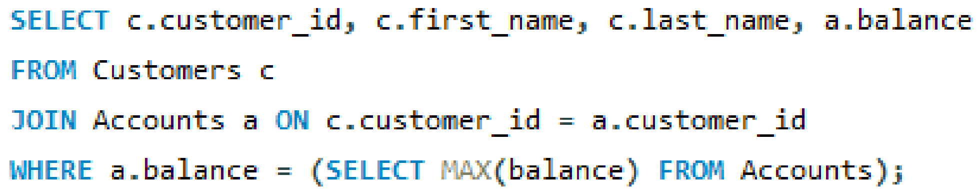
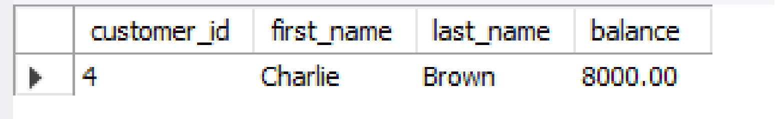
## Tasks 4: Subquery and its type

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

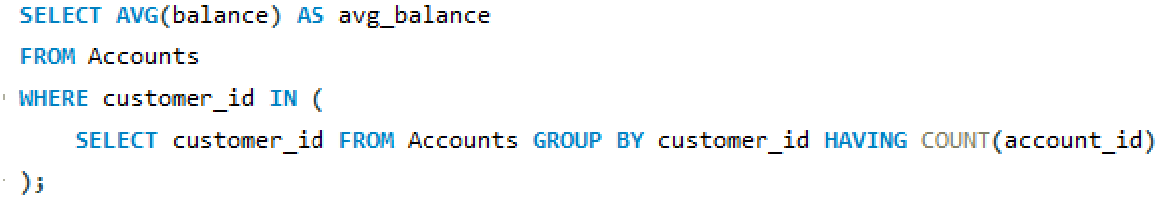


```
SELECT c.customer_id, c.first_name, c.last_name, a.balance
FROM Customers c
JOIN Accounts a ON c.customer_id = a.customer_id
WHERE a.balance = (SELECT MAX(balance) FROM Accounts);
```

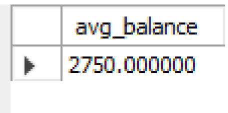


| | customer_id | first_name | last_name | balance |
|---|---|---|---|---|
| ▶ | 4 | Charlie | Brown | 8000.00 |

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

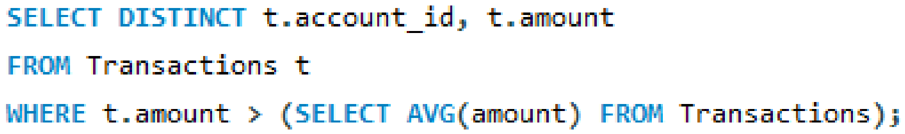


```
SELECT AVG(balance) AS avg_balance
FROM Accounts
WHERE customer_id IN (
    SELECT customer_id FROM Accounts GROUP BY customer_id HAVING COUNT(account_id)
);
```

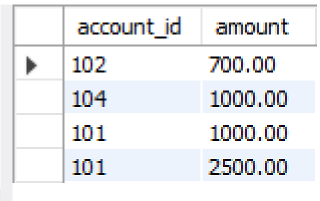


| | avg_balance |
|---|---|
| ▶ | 2750.000000 |

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

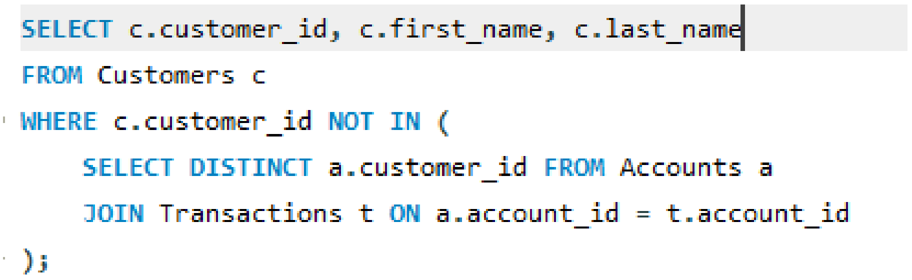


```
SELECT DISTINCT t.account_id, t.amount
FROM Transactions t
WHERE t.amount > (SELECT AVG(amount) FROM Transactions);
```

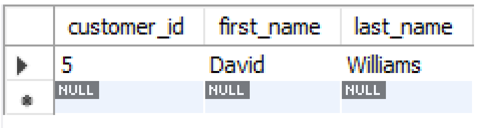


| | account_id | amount |
|---|---|---|
| ▶ | 102 | 700.00 |
| | 104 | 1000.00 |
| | 101 | 1000.00 |
| | 101 | 2500.00 |

4) **Identify customers who have no recorded transactions.**

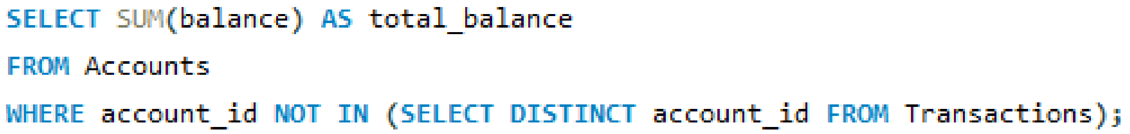


```
SELECT c.customer_id, c.first_name, c.last_name
FROM Customers c
WHERE c.customer_id NOT IN (
    SELECT DISTINCT a.customer_id FROM Accounts a
    JOIN Transactions t ON a.account_id = t.account_id
);
```

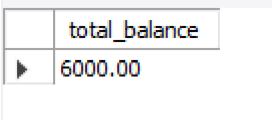


| | customer_id | first_name | last_name |
|---|---|---|---|
| ▶ | 5 | David | Williams |
| * | NULL | NULL | NULL |

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

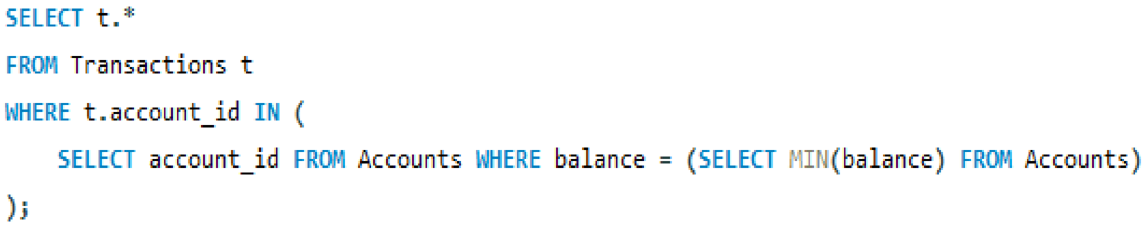


```
SELECT SUM(balance) AS total_balance
FROM Accounts
WHERE account_id NOT IN (SELECT DISTINCT account_id FROM Transactions);
```

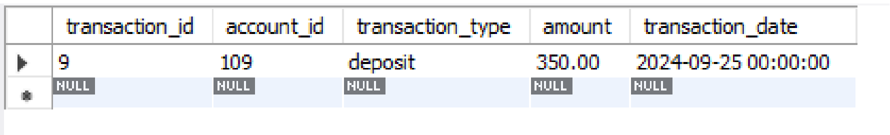


| | total_balance |
|---|---|
| ▶ | 6000.00 |

6) **Retrieve transactions for accounts with the lowest balance.**

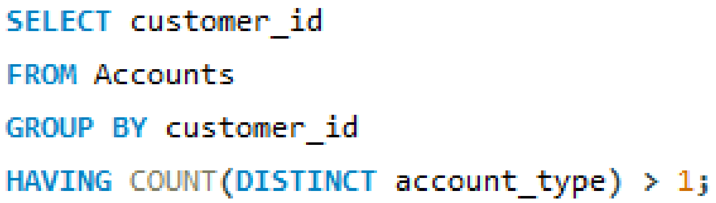


```
SELECT t.*
FROM Transactions t
WHERE t.account_id IN (
    SELECT account_id FROM Accounts WHERE balance = (SELECT MIN(balance) FROM Accounts)
);
```

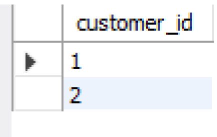


| | transaction_id | account_id | transaction_type | amount | transaction_date |
|---|---|---|---|---|---|
| ▶ | 9 | 109 | deposit | 350.00 | 2024-09-25 00:00:00 |
| * | NULL | NULL | NULL | NULL | NULL |

7) **Identify customers who have accounts of multiple types.**

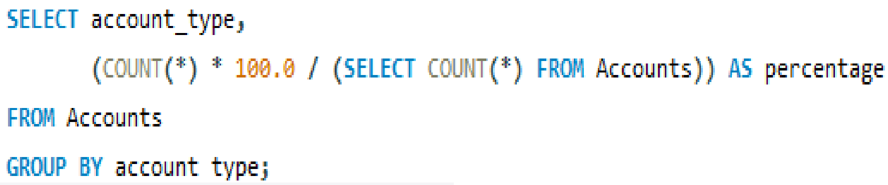


```
SELECT customer_id
FROM Accounts
GROUP BY customer_id
HAVING COUNT(DISTINCT account_type) > 1;
```

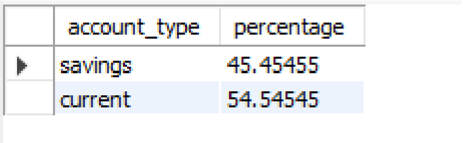


| | customer_id |
|---|---|
| ▶ | 1 |
| | 2 |

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

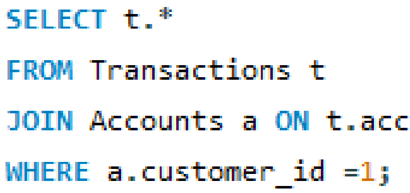


```
SELECT account_type,
       (COUNT(*) * 100.0 / (SELECT COUNT(*) FROM Accounts)) AS percentage
FROM Accounts
GROUP BY account_type;
```

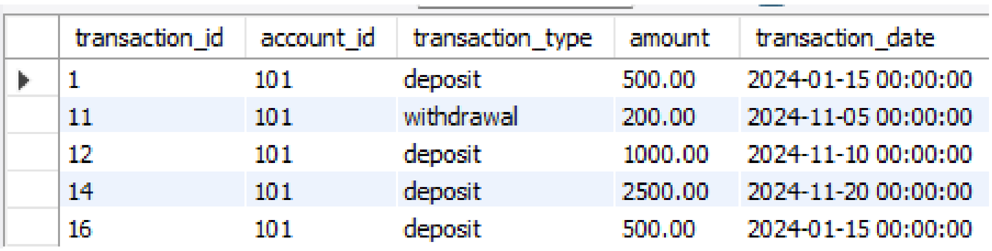


| | account_type | percentage |
|---|---|---|
| ▶ | savings | 45.45455 |
| | current | 54.54545 |

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

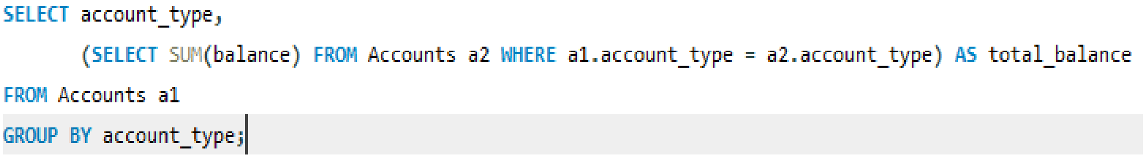


```
SELECT t.*
FROM Transactions t
JOIN Accounts a ON t.account_id = a.account_id
WHERE a.customer_id =1;
```

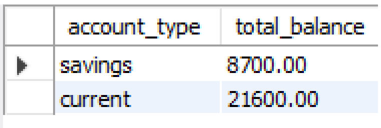


| transaction_id | account_id | transaction_type | amount | transaction_date |
|---|---|---|---|---|
| 1 | 101 | deposit | 500.00 | 2024-01-15 00:00:00 |
| 11 | 101 | withdrawal | 200.00 | 2024-11-05 00:00:00 |
| 12 | 101 | deposit | 1000.00 | 2024-11-10 00:00:00 |
| 14 | 101 | deposit | 2500.00 | 2024-11-20 00:00:00 |
| 16 | 101 | deposit | 500.00 | 2024-01-15 00:00:00 |

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



```
SELECT account_type,
       (SELECT SUM(balance) FROM Accounts a2 WHERE a1.account_type = a2.account_type) AS total_balance
FROM Accounts a1
GROUP BY account_type;
```



| account_type | total_balance |
|---|---|
| savings | 8700.00 |
| current | 21600.00 |