BANKING SYSTEM

Tasks 4: Subquery and its type:

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

```
select c.* from customers c join accounts a on a.customer_id=c.customer_id where balance = (select max(balance) from accounts);
```

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

```
select 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);
```

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

```
select account_id , amount from transactions where  amount > (select avg(amount) from transactions) ;
```

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

```
 SELECT customer_id, first_name, last_name
FROM customers
WHERE customer_id NOT IN (
    SELECT DISTINCT c.customer_id
    FROM customers c
    JOIN accounts a ON c.customer_id = a.customer_id
    JOIN transactions t ON a.account_id = t.account_id
);
```

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 );
```

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 balance = (select min(balance) from accounts a );
```

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

```
SELECT c.customer_id, CONCAT(c.first_name, ' ', c.last_name) AS customer_name
FROM customers c
JOIN accounts a ON a.customer_id = c.customer_id
WHERE c.customer_id IN (
    SELECT customer_id
    FROM accounts
    GROUP BY customer_id
    HAVING COUNT(account_id) > 1
);
```

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

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

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

```
SELECT *
FROM transactions
WHERE account_id IN (
    SELECT account_id
    FROM accounts
    WHERE customer_id = 6
);
```

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

```
SELECT DISTINCT account_type,
       (SELECT SUM(balance)
        FROM accounts a2
        WHERE a2.account_type = a1.account_type) AS total_balance
FROM accounts a1;
```