

# ASSIGNMENT-BANKING SYSTEM

## TASK 4

### Subquery and its type

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

```
SELECT * FROM Accounts WHERE balance = (SELECT  
MAX(balance) FROM Accounts);
```

```
mysql> SELECT * FROM Accounts WHERE balance = (SELECT MAX(balance) FROM Accounts);  
+-----+-----+-----+-----+  
| account_id | customer_id | account_type | balance |  
+-----+-----+-----+-----+  
|          4 |           4 | current      | 777700.00 |  
+-----+-----+-----+-----+  
1 row in set (0.04 sec)
```

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

```
SELECT AVG(balance) FROM Accounts WHERE customer_id IN (  
    SELECT customer_id FROM Accounts GROUP BY customer_id  
HAVING COUNT(*) > 1  
);
```

```
mysql> SELECT AVG(balance) FROM Accounts WHERE customer_id IN (  
-> SELECT customer_id FROM Accounts GROUP BY customer_id HAVING COUNT(*) > 1  
-> );  
+-----+  
| AVG(balance) |  
+-----+  
| 6533.458333 |  
+-----+  
1 row in set (0.01 sec)
```

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

```
SELECT * FROM Transactions WHERE amount > (SELECT  
AVG(amount) FROM Transactions);
```

```
mysql> SELECT * FROM Transactions WHERE amount > (SELECT AVG(amount) FROM Transactions);
```

| transaction_id | account_id | transaction_type | amount  | transaction_date    |
|----------------|------------|------------------|---------|---------------------|
| 4              | 4          | transfer         | 1500.00 | 2024-03-23 11:10:00 |
| 6              | 6          | deposit          | 2500.00 | 2024-03-25 12:00:00 |
| 8              | 8          | transfer         | 1800.00 | 2024-03-27 15:05:00 |
| 9              | 9          | deposit          | 5000.00 | 2024-03-28 08:30:00 |
| 11             | 1          | deposit          | 2000.00 | 2024-03-30 10:30:00 |
| 13             | 12         | deposit          | 3000.00 | 2024-03-29 14:15:00 |
| 14             | 13         | transfer         | 1500.00 | 2024-03-28 09:00:00 |

```
7 rows in set (0.01 sec)
```

#### 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
);
```

```
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 |
|-------------|------------|-----------|
| 11          | Draco      | Malfoy    |

```
1 row in set (0.00 sec)
```

**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);
```

```
mysql> SELECT SUM(balance) AS total_balance  
-> FROM Accounts  
-> WHERE account_id NOT IN (SELECT DISTINCT account_id FROM Transactions);  
+-----+  
| total_balance |  
+-----+  
|      2500.00 |  
+-----+  
1 row in set (0.01 sec)
```

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

```
SELECT * FROM Transactions WHERE account_id = (  
    SELECT account_id FROM Accounts ORDER BY balance ASC  
LIMIT 1  
);
```

```
mysql> SELECT * FROM Transactions WHERE account_id = (  
->    SELECT account_id FROM Accounts ORDER BY balance ASC LIMIT 1  
-> );  
+-----+-----+-----+-----+-----+  
| transaction_id | account_id | transaction_type | amount | transaction_date |  
+-----+-----+-----+-----+-----+  
|             10 |          10 | withdrawal      | 300.00 | 2024-03-29 19:55:00 |  
+-----+-----+-----+-----+-----+  
1 row in set (0.00 sec)
```

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

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

```
mysql> SELECT customer_id, COUNT(DISTINCT account_type) AS account_types
-> FROM Accounts
-> GROUP BY customer_id
-> HAVING COUNT(DISTINCT account_type) > 1;
```

| customer_id | account_types |
|-------------|---------------|
| 1           | 2             |
| 5           | 2             |

2 rows in set (0.01 sec)

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

SELECT account\_type, COUNT(\*) \* 100 / (SELECT COUNT(\*) FROM Accounts) AS percentage FROM Accounts GROUP BY account\_type;

| account_type | percentage |
|--------------|------------|
| savings      | 50.0000    |
| current      | 42.8571    |
| zero_balance | 7.1429     |

3 rows in set (0.01 sec)

**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 = 1);

```
mysql> SELECT * FROM Transactions WHERE account_id IN (SELECT account_id FROM Accounts WHERE customer_id = 1);
```

| transaction_id | account_id | transaction_type | amount  | transaction_date    |
|----------------|------------|------------------|---------|---------------------|
| 1              | 1          | deposit          | 500.00  | 2024-03-20 10:15:00 |
| 11             | 1          | deposit          | 2000.00 | 2024-03-30 10:30:00 |
| 12             | 1          | withdrawal       | 500.00  | 2024-03-31 12:45:00 |
| 15             | 11         | withdrawal       | 1000.00 | 2024-03-27 16:30:00 |

4 rows in set (0.01 sec)

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

SELECT account\_type, SUM(balance) FROM Accounts GROUP BY account\_type;

```
mysql> SELECT account_type, SUM(balance) FROM Accounts GROUP BY account_type;
```

| account_type | SUM(balance) |
|--------------|--------------|
| savings      | 35900.75     |
| current      | 811200.75    |
| zero_balance | 0.00         |

3 rows in set (0.00 sec)

## NOTES:

Additionally a record is inserted in both the Customers and Accounts table.

## Customers table:

```
mysql> select * from Customers;
```

| customer_id | first_name | last_name | DOB        | email                      | phone_number | address                                 |
|-------------|------------|-----------|------------|----------------------------|--------------|---|
| 1           | Harry      | Potter    | 1990-07-31 | harry.potter@gmail.com     | 9876543210   | 4 Privet Drive, Little Whinging, Surrey |
| 2           | Ron        | Weasley   | 1990-03-01 | ron.weasley@gmail.com      | 9845678901   | The Burrow, Ottery St. Catchpole, Devon |
| 3           | Hermione   | Granger   | 1989-09-19 | hermione.granger@gmail.com | 9786543210   | Hampstead Garden Suburb, London         |
| 4           | Sirius     | Black     | 1959-11-03 | sirius.black@gmail.com     | 9765432109   | 12 Grimmauld Place, London              |
| 5           | Remus      | Lupin     | 1960-03-10 | remus.lupin@gmail.com      | 9876504321   | 13th Avenue Road, London                |
| 6           | Lily       | Evans     | 1960-01-30 | lily.evans@gmail.com       | 9876543098   | Spinner's End, Cokeworth                |
| 7           | Luna       | Lovegood  | 1981-02-13 | luna.lovegood@gmail.com    | 9785463210   | Lovegood House, Ottery St. Catchpole    |
| 8           | Narcissa   | Malfoy    | 1955-06-01 | narcissa.malfoy@gmail.com  | 9898765432   | Malfoy Manor, Wiltshire                 |
| 9           | James      | Potter    | 1960-03-27 | james.potter@gmail.com     | 9876234510   | Godric's Hollow, West Country, England  |
| 10          | Tom        | Riddle    | 1926-12-31 | tom.riddle@gmail.com       | 9887766543   | Wool's Orphanage, London                |
| 11          | Draco      | Malfoy    | 1980-06-05 | draco.malfoy@gmail.com     | 9876543111   | Malfoy Manor, Wiltshire                 |

11 rows in set (0.00 sec)

## Accounts table:

| account_id | customer_id | account_type | balance   |
|------------|-------------|--------------|-----------|
| 1          | 1           | savings      | 2000.00   |
| 2          | 2           | current      | 3200.50   |
| 3          | 3           | savings      | 7800.75   |
| 4          | 4           | current      | 777700.00 |
| 5          | 5           | savings      | 5400.00   |
| 6          | 6           | current      | 9000.25   |
| 7          | 7           | savings      | 1200.00   |
| 8          | 8           | current      | 4300.00   |
| 9          | 9           | savings      | 10000.00  |
| 10         | 10          | zero_balance | 0.00      |
| 11         | 1           | current      | 5000.00   |
| 12         | 3           | savings      | 7000.00   |
| 13         | 5           | current      | 12000.00  |