Islamic University of Technology (IUT)

Report on Lab 08

Submitted By

Shanta Maria (ID:200042172)

CSE 4308 Database Management Systems Lab

Submitted To

Md. Bakhtiar Hasan

Lecturer, Department of CSE

Zannatun Naim Srsity

Lecturer, Department of CSE

October 29, 2022

Introduction

In the lab class, we were given four tasks to solve using java programming language to run SQL queries instead of the command line.

Scenario

1 Scenario

Consider the (partial) schema of a Banking Management System shown in Figure 1:

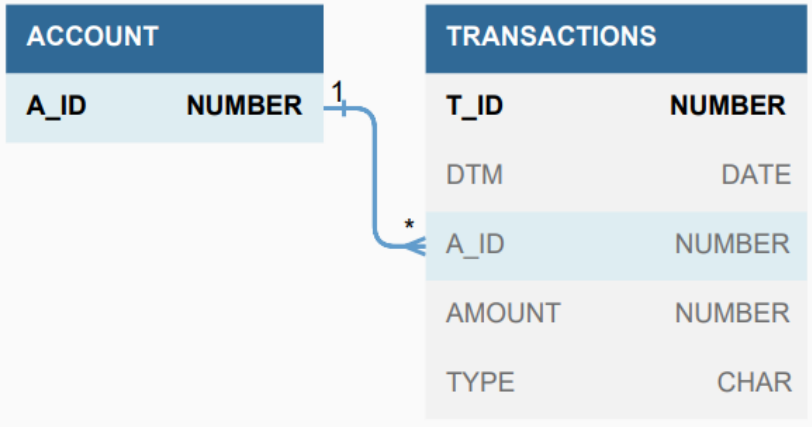

Figure 1: Schema for Lab Task

For simplicity, the ACCOUNT table contains only the account ID. The TRANSACTIONS table contains the information related to each transactions that occurs in the bank. The table stores transaction ID, date of the transaction, account involved in the transaction, the amount transacted, and the type of the transaction. The TYPE column is set to 0 if the money is credited to the account, i.e., the money is added to the account, and 1 if the money is debited from the account, i.e., the money is subtracted from the account. There are 3 types of accounts in the bank:

1. **Commercially Important Person (CIP):** If the person has a balance of more than 1,000,000 and all the transactions that they have made totals more than 5,000,000.
2. **Very Important Person (VIP):** If the person has a balance of more than 500,000 but less than 900,000 and all the transactions that they have made totals more than 2,500,000 but less than 4,500,000.
3. **Ordinary Person (OP):** If the user has a balance less than 100,000 and all the transactions they have made totals less than 1,000,000.

Figure 1: Scenario

Tasks

3 Lab Task

Write a JAVA code to:

1. Count the total number of transactions conducted under account 49.
2. Count the number of credit.
3. List the transactions that occurred in the last 6 months of 2021.
4. Count the number of CIP, VIP, and OPs. Also show the number of people that do not fall in any of the categories.

Figure 2: Tasks

Solution (Java Code)

0.1 Solution class

```
//Lab Task
task t=new task();

//Task 1
t.task1();
System.out.println("\nExecuting the query for task 1: ");
rs=stmt.executeQuery(t.sql);
while(rs.next())
{
    int count=rs.getInt("count(t_id)");
    System.out.println(count +
    " transactions have been conducted under account 49.");
}
```

```
//Task 2
t.task2();
System.out.println("\nExecuting the query for task 2: ");
rs=stmt.executeQuery(t.sql);
while(rs.next())
{
    int count=rs.getInt("count(t_id)");
    System.out.println(count + " is the number of total credits.");
}

//Task 3
t.task3();
System.out.println("\nExecuting the query for task 3:");
System.out.println("\nList of transactions that occured:");
rs=stmt.executeQuery(t.sql);
while(rs.next())
{
    int t_id=rs.getInt("t_id");
    String dtm=rs.getString("dtm");
    int a_id=rs.getInt("a_id");
    int amount=rs.getInt("amount");
    String type=rs.getString("type");

    System.out.println(t_id + " " + dtm + " " + a_id + " " +
    amount + " " + type);
}

//Task 4
```

```
t.task4();

System.out.println("\nExecuting the queries for task 4: ");

stmt.executeQuery(t.sql_for_view_balance);
stmt.executeQuery(t.sql_for_view_transaction_amount);
stmt.executeQuery(t.sql_for_view_cip);
stmt.executeQuery(t.sql_for_view_vip);
stmt.executeQuery(t.sql_for_view_op);

rs=stmt.executeQuery(t.sql_for_cip);
while(rs.next())
{
    int cip=rs.getInt("cip");
    System.out.println("Number of CIP accounts: " + cip);
}

rs=stmt.executeQuery(t.sql_for_vip);
while(rs.next())
{
    int vip=rs.getInt("vip");
    System.out.println("Number of VIP accounts: " + vip);
}

rs=stmt.executeQuery(t.sql_for_op);
while(rs.next())
{
    int op=rs.getInt("op");
    System.out.println("Number of OP accounts: " + op);
```

```
}

rs=stmt.executeQuery(t.sql_for_neither);
while(rs.next())
{
    int n=rs.getInt("neither");
    System.out.println("Number of accounts which are neither CIP, VIP nor OP: "
    + n);
}

rs.close();
stmt.close();
conn.close();
}
```

0.2 Task class

```
import java.sql.ResultSet;
import java.sql.Statement;

public class task {

    String sql=null;
    String sql_for_view_balance=null;
    String sql_for_view_transaction_amount=null;
    String sql_for_view_cip=null;
    String sql_for_view_vip=null;
    String sql_for_view_op=null;
    String sql_for_cip=null;
```

```
String sql_for_vip=null;
String sql_for_op=null;
String sql_for_neither=null;

public void task1()
{
    sql="SELECT COUNT(T_ID) FROM TRANSACTIONS WHERE A_ID=49";
}

public void task2()
{
    sql="SELECT COUNT(T_ID) FROM TRANSACTIONS WHERE TYPE='0'";
}

public void task3()
{
    sql="SELECT * FROM TRANSACTIONS WHERE DTM BETWEEN
    TO_DATE('2021/06/01 00:00:00', 'yyyy/mm/dd hh24:mi:ss') AND
    TO_DATE('2022/01/01 00:00:00', 'yyyy/mm/dd hh24:mi:ss')";
}

public void task4()
{
    sql_for_view_balance="create or replace view balance as\n" +
            "select sum(case\n" +
            "            when t.type='0' then amount\n" +
            "            when t.type='1' then -amount\n" +
            "            else 0\n" +
            "            end) as total, a_id\n" +
```

```
        "from transactions t\n" +
        "group by a_id";

sql_for_view_transaction_amount=
"create or replace view total_amount as\n" +
        "select sum(amount) as sum_amount, a_id\n" +
        "from transactions t\n" +
        "group by a_id";

sql_for_view_cip="create or replace view total_cip as\n" +
        "select distinct b.a_id\n" +
        "from balance b, total_amount t\n" +
        "where b.total>1000000 and t.sum_amount>5000000";

sql_for_view_vip="create or replace view total_vip as\n" +
        "select distinct b.a_id\n" +
        "from balance b, total_amount t\n" +
        "where b.total>500000 and b.total<900000 and \n" +
        "t.sum_amount>2500000 and t.sum_amount<4500000";

sql_for_view_op="create or replace view total_op as\n" +
        "select distinct b.a_id\n" +
        "from balance b, total_amount t\n" +
        "where b.total<100000 and t.sum_amount<1000000";

sql_for_cip="select count(*) as cip\n" +
        "from total_cip";

sql_for_vip="select count(*) as vip\n" +
```

```
                "from total_vip";

        sql_for_op="select count(*) as op\n" +
                "from total_op";

        sql_for_neither="select count(a_id) as neither\n" +
                "from account\n" +
                "where a_id not in (select a_id from total_vip) and\n" +
                "a_id not in (select a_id from total_cip) and \n" +
                "a_id not in (select a_id from total_op)";
    }
}
```

Analysis and Explanation

Task 1

This task was easy to solve. I counted the number of transaction IDs from the transaction table for the account with ID 49.

Task 2

This task was also easy to solve. I counted the number of transaction IDs where the type of transaction was `'0'` which means the amount was credited.

Task 3

This task was a little challenging. I showed all the transactions from the transaction table for the transactions whose dates were between the first of June 2021 from 12am to the first of January 2022 12am.

Task 4

This task was difficult to solve. I made the use of many views for my solution. First, I created a view which calculated the total balance for a particular account. I used the sum() function with a case statement inside it which subtracted the amount if it was of type='1' that is debit otherwise it would add the amount since it would be credit. I created another view to calculate the total transaction amount for an account which was basically the sum of all the amounts that the user used in transactions. Then I created three separate views for CIP, VIP and OP accounts. For each, I used the previous two views to find the accounts where their account balance and total transaction amount was above a certain threshold. I used the three views to calculate the number of each type of account using the count() function and displayed that amount. For the last part of the task, I counted the number of accounts which were neither in the CIP view, VIP view or the OP view and displayed that amount.

Difficulties

I did face some difficulties when doing the fourth task. I tried solving the task with nested queries at first but it showed a lot of error and it was confusing to keep track of all the results. Thus, I created separate views for each of the nested queries which made it a lot easier to write the final query statement.

Conclusion

As shown in the report, I have solved and tested the solutions for all four tasks given in the lab. All the commands used were written in IntelliJ IDE.