Database Management Systems Lab
Lab 8
CSE 4308

A Z Hasnain Kabir
200042102
Software Engineering

2 November 2022

Introduction

In this lab we are to work with Java database connectivity and complete some lab tasks.

Tasks

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.

Task 1

Analysis of the problem

We have to execute a query that calculates total number of transactions occured under account 49.

Code

```
1  public static void main (String args[])
2      {
3          Connection conn=null;
4          Statement stmt=null;
5          try
6          {
7              Class.forName(JDBC_DRIVER);
8              System.out.println("Connecting to database");
9              conn=DriverManager.getConnection(DB_URL, USER,
                   PASS);
10             System.out.println("Creating statement");
11             stmt=conn.createStatement();
12             String sql;
13             sql="SELECT count(A_ID) as accountID FROM
                   ACCOUNT WHERE A_ID = '49'";
14             System.out.println("Executing the query: " +
                   sql);
15             ResultSet rs=stmt.executeQuery(sql);
16             while(rs.next())
17             {
18                 int accounts = rs.getInt("accountID");
19                 System.out.println(accounts + "
                       transactions have occured");
20             }
21
22             rs.close();
23             stmt.close();
24             conn.close();
25             System.out.println("Thank you for banking with
                   us!");
26         }
27         catch(SQLException se)
28         {
29             se.printStackTrace();
30         }
31         catch(Exception e)
32         {
33             e.printStackTrace();
34         }
35     }
```

Explanation of the solution

After establishing the database connection, we write our SQL query which selects number of accounts under account id 49.

Findings

We find out that when establishing database connection, the name of the PC should be the same as defined in the Oracle files. Otherwise, the connection will not be established

Task 2

Analysis of the problem

We have to count the number of transactions that are credit.

Code

```
1  public static void main (String args[])
2      {
3          Connection conn=null;
4          Statement stmt=null;
5          try
6          {
7              Class.forName(JDBC_DRIVER);
8              System.out.println("Connecting to database");
9              conn=DriverManager.getConnection(DB_URL, USER,
                   PASS);
10             System.out.println("Creating statement");
11             stmt=conn.createStatement();
12             String sql;
13             sql="SELECT count(T_id) as numberOfCredit FROM
                   TRANSACTIONS WHERE TYPE = '0'";
14             System.out.println("Executing the query: " +
                   sql);
15             ResultSet rs=stmt.executeQuery(sql);
16             while(rs.next())
17             {
18                 int transactions = rs.getInt("numberOfCredit
                       ");
```

```
19                  System.out.println(transactions + " number
                       of credit");
20              }
21
22              rs.close();
23              stmt.close();
24              conn.close();
25              System.out.println("Thank you for banking with
                   us!");
26          }
27          catch(SQLException se)
28          {
29              se.printStackTrace();
30          }
31          catch(Exception e)
32          {
33              e.printStackTrace();
34          }
35      }
```

Explanation of the solution

Since the credit is defined as a transaction type of 0, we can easily select count of transaction id that correspond to the type 0.

Findings

We have to select the transaction type under single quotes even though it is of type integer

Task 3

Analysis of the problem

We have to select that transactions which occurred in the last six months of 2021 using the DTM attribute of the transactions table.

Code

```
1 public static void main (String args[])
2     {
```

```
3           Connection conn=null;
4           Statement stmt=null;
5           try
6           {
7               Class.forName(JDBC_DRIVER);
8               System.out.println("Connecting to database");
9               conn=DriverManager.getConnection(DB_URL, USER,
                    PASS);
10              System.out.println("Creating statement");
11              stmt=conn.createStatement();
12              String sql;
13              sql="SELECT count(T_id) AS TRANSACTION FROM ("
                    +
14                      "SELECT T_id, EXTRACT(YEAR FROM DTM) as
                            YEAR," +
15                      "EXTRACT(MONTH FROM DTM) AS MONTH FROM
                            TRANSACTIONS) WHERE YEAR= 2021" +
16                      "AND MONTH >6 ";
17              System.out.println("Executing the query: " +
                    sql);
18              ResultSet rs=stmt.executeQuery(sql);
19              while(rs.next())
20              {
21                  int transaction = rs.getInt("TRANSACTION");
22                  System.out.println(transaction + "
                        transactions in last 6 months of 2021");
23              }
24
25              rs.close();
26              stmt.close();
27              conn.close();
28              System.out.println("Thank you for banking with
                    us!");
29          }
30          catch(SQLException se)
31          {
32              se.printStackTrace();
33          }
34          catch(Exception e)
35          {
36              e.printStackTrace();
37          }
38      }
```

Explanation of the solution

We select number of transaction ids from a table we define through our nested query which selects transaction id and extracts year and month from transaction table. We then use our boolean expression to filter out the year and month.

Findings

Nested queries can be very useful in cases where we have many operations occurring at the same time.

Task 4

Analysis of the problem

The CIPs, VIPs and OPs are defined in our problem statement and we have to use these to count number of people categorically.

Code

```
1  public static void main (String args[])
2      {
3          Connection conn=null;
4          Statement stmt=null;
5          try
6          {
7              Class.forName(JDBC_DRIVER);
8              System.out.println("Connecting to database");
9              conn=DriverManager.getConnection(DB_URL, USER,
                   PASS);
10             System.out.println("Creating statement");
11             stmt=conn.createStatement();
12             int countCIP=0;
13             int countVIP=0;
14             int countOP=0;
15             String sql;
16             sql="SELECT A_ID FROM ACCOUNT ";
17             System.out.println("Executing the query: " +
                   sql);
18             ResultSet rs=stmt.executeQuery(sql);
```

```
19              ArrayList<Integer> newlist=new ArrayList<
                   Integer>();
20              while(rs.next())
21              {
22                  int result=rs.getInt(1);
23                  newlist.add(result);
24              }
25              for (int n:newlist) {
26                  String sql2;
27                  sql2="SELECT AMOUNT, TYPE FROM TRANSACTIONS
                        WHERE A_ID="+n;
28                  System.out.println("Executing the query: "
                       + sql2);
29                  ResultSet rs2=stmt.executeQuery(sql2);
30                  int amount=0, balance = 0;
31                  while(rs2.next()){
32                      int num=rs2.getInt(1);
33                      String type=rs2.getString(2);
34                      if(type.charAt(0)=='0'){
35                          balance=balance+num;
36                      }else{
37                          balance=balance-num;
38                      }
39                      amount=amount+num;
40                      //System.out.println("amount:- "+num);
41                  }if(balance>1000000&&amount>5000000){
42                      countCIP++;
43                  } else if (balance>5000000&&amount>2500000)
                        {
44                      countVIP++;
45                  }else{
46                      countOP++;
47                  }
48              }
49
50
51
52
53              rs.close();
54              stmt.close();
55              conn.close();
56              System.out.println("Thank you for banking with
                   us!");
```

```
57              System.out.println("Number of CIP " + countCIP)
                    ;
58              System.out.println("\nNumber of VIP " +
                    countCIP);
59              System.out.println("\nNumber of OP " + countOP)
                    ;
60          }
61          catch(SQLException se)
62          {
63              se.printStackTrace();
64          }
65          catch(Exception e)
66          {
67              e.printStackTrace();
68          }
69      }
```

Explanation of the solution

We store the balance of account holders in a variable and match our variable according to the values mentioned in the problem statement. We increment variables countCIP, countVIP, countOP to get our result.

Findings

Working with variables in our java code instead of writing SQL queries to arrive at our desired result is much easier and more managable.