Islamic University of Technology (IUT)

CSE 4308: Database Management Systems Lab

Lab Report # 8

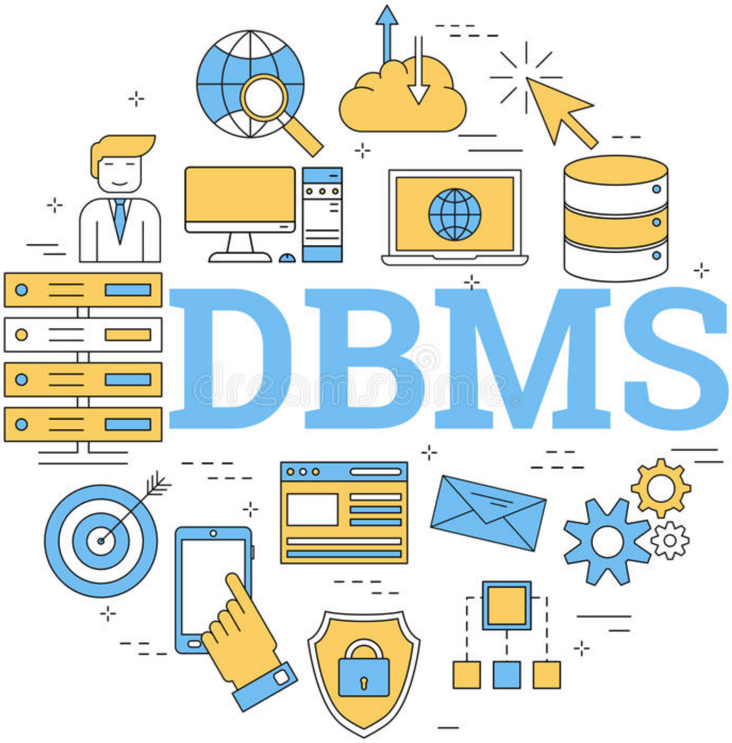
Submitted to:

Md. Bakhtiar Hasan,

Asst. Professor, CSE.

Zannatun Naim Srsity,

Lecturer, CSE.



Submitted by:

M M Nazmul Hossain
ID 200042118
CSE (SWE)

Submission Date:

30.10.2022

Introduction

The eighth Database Management Systems Lab was about running SQL queries through java by establishing a connection between java and oracle. This allows more flexible variable operation which would've required more complex SQL queries otherwise.

Method

First, the table.sql file had to be generated following the provided instructions. Then the file is loaded into the oracle database. The table.sql contained the DDL SQL code which will be required for this lab. After that, a connection had to be established between the Java and the Oracle database by adding an external ojdbc.jar file to the Java IDE following all the given instructions.

Analysis of the problem of Task 1

The task was to generate the total number of transactions that were made by an account holder whose a_id was 49.

The Code

```
import java.sql.*;
import java.util.ArrayList;

public class Lab_8
{
    static final String JDBC_DRIVER = "oracle.jdbc.driver.OracleDriver";
    static final String DB_URL= "jdbc:oracle:thin:@localhost:1521:xe";
    static final String USER="dbms_200042118";
    static final String PASS="cse4308";
    public static void main (String args[])
    {
        Connection conn=null;
        Statement stmt=null;
        try
        {
            Class.forName(JDBC_DRIVER);
            System.out.println("Connecting to database");
            conn=DriverManager.getConnection(DB_URL, USER, PASS);
            System.out.println("Creating statement");
            stmt=conn.createStatement();
            String sql;
            sql="SELECT A_ID AS ACCOUNT, COUNT(T_ID) AS NO_OF_TRANSACTIONS FROM
TRANSACTIONS WHERE a_id = 49 GROUP BY a_id";
            System.out.println("Executing the query of Task 1:\n" + sql);
            ResultSet rs=stmt.executeQuery(sql);
            while(rs.next()) {
                int account = rs.getInt("ACCOUNT");
                long no_of_transactions = rs.getLong("NO_OF_TRANSACTIONS");
                System.out.print("Account ID: "+account + " has a total of " +
no_of_transactions + " transactions.");
            }

            rs.close();
            stmt.close();
            conn.close();
            System.out.println("Thank you for banking with us!");
        }

        catch(SQLException se)
        {
            se.printStackTrace();
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

Explanation of Solution

- Following the provided solution file, a connection was established to the Oracle database providing the database URL, username, and password.
- Then just execute the SQL query by passing the query as a string to the executeQuery method. For this task, select a_id and count (t_id) as no_of_transactions from the transactions table, grouped by a_id. Set an additional where condition which imposes the a_id has to be equal to 49.

- From the generated result set, extract the required values to int variables and output them on screen.

Analysis of the problem of Task 2

The second task asked to count the number of credit transactions. There was a type attribute that imposed which type of transaction it was.

The Code

```
import java.sql.*;
import java.util.ArrayList;

public class Lab_8
{
    static final String JDBC_DRIVER = "oracle.jdbc.driver.OracleDriver";
    static final String DB_URL= "jdbc:oracle:thin:@localhost:1521:xe";
    static final String USER="dbms_200042118";
    static final String PASS="cse4308";
    public static void main (String args[])
    {
        Connection conn=null;
        Statement stmt=null;
        try
        {
            Class.forName(JDBC_DRIVER);
            System.out.println("Connecting to database");
            conn=DriverManager.getConnection(DB_URL, USER, PASS);
            System.out.println("Creating statement");
            stmt=conn.createStatement();
            String sql;
            sql="SELECT COUNT (TYPE) AS NO_OF_CREDIT FROM TRANSACTIONS WHERE TYPE
= 0 GROUP BY TYPE";
            System.out.println("Executing the query of Task 2: " + sql);
            ResultSet rs =stmt.executeQuery(sql);
            while(rs.next())
            {
                long no_of_credit=rs.getLong("NO_OF_CREDIT");
                System.out.print("There are a total of "+no_of_credit+" credit
transactions.");
            }
            rs.close();
            stmt.close();
            conn.close();
            System.out.println("Thank you for banking with us!");
        }

        catch(SQLException se)
        {
            se.printStackTrace();
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

Explanation of Solution

- Establishing a connection similar to task 1.
- The SQL query selects the count (t_id) as no_of_credit_transactions from the transactions table with the where condition that the type had to ‘’0’’ (credit) grouped by type.
- Then extract the no_of_credit_transactions to a variable from the generated result set and print that out on the screen.

Analysis of the problem of Task 3

For the third task, the transactions that occurred between the last six months of 2021 had to be listed.

The Code

```
public class Lab_8
{
    static final String JDBC_DRIVER = "oracle.jdbc.driver.OracleDriver";
    static final String DB_URL= "jdbc:oracle:thin:@localhost:1521:xe";
    static final String USER="dbms_200042118";
    static final String PASS="cse4308";
    public static void main (String args[])
    {
        Connection conn=null;
        Statement stmt=null;
        try
        {
            Class.forName(JDBC_DRIVER);
            System.out.println("Connecting to database");
            conn=DriverManager.getConnection(DB_URL, USER, PASS);
            System.out.println("Creating statement");
            stmt=conn.createStatement();
            String sql;
            sql="SELECT T_ID FROM TRANSACTIONS WHERE MONTHS_BETWEEN(DATE'2022-1-
1', DTM)<=6 AND DTM < DATE '2022-1-1'";
            System.out.println("Executing the query of Task 3: " + sql);
            ResultSet rs =stmt.executeQuery(sql);
            System.out.println("The list of transactions which occurred in the
last 6 months of 2021 are: ");
            while(rs.next())
            {
                int t_id=rs.getInt("T_ID");
                System.out.print(t_id+"\n");
            }
            rs.close();
            stmt.close();
            conn.close();
            System.out.println("Thank you for banking with us!");
        }

        catch(SQLException se)
        {
            se.printStackTrace();
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

Explanation of Solution

- A connection is first established with the database.
- To print the transactions which occurred between the given date, the SQL query selects t_id from the transactions table where the months between

DTM 2022-01-01 is less than equal to 6, and to select only the dates before 2022-01-01 DTM has to be less than 2022-01-01.

- Then extract the transaction ids from the result set and output it on the screen.

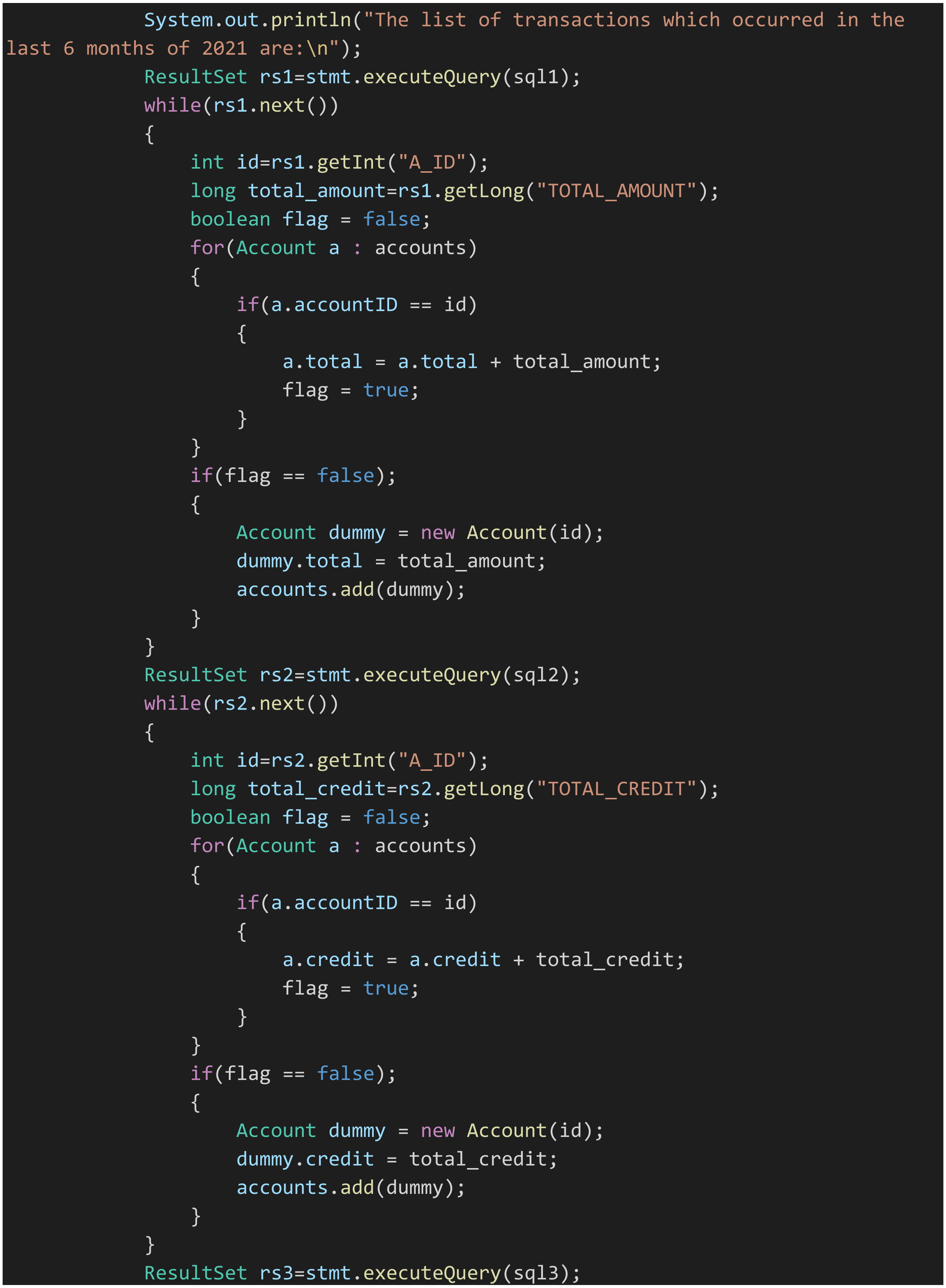
Analysis of the problem of Task 4

The fourth task was to count the number of CIPs, VIPs, and OPs and the number of people who fall under none of the provided categories.

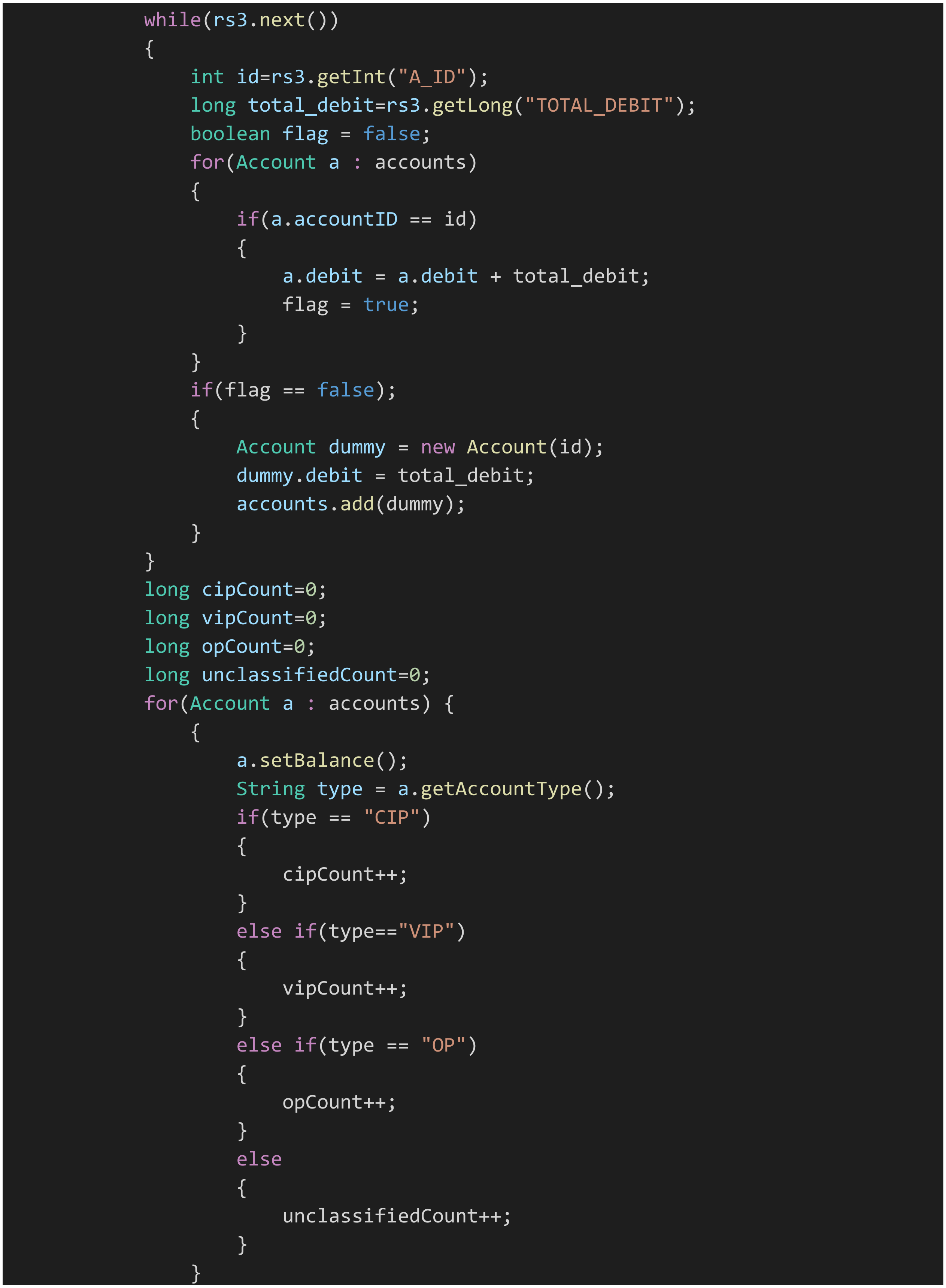
The Code

```
import java.sql.*;
import java.util.ArrayList;

public class Lab_8
{
    static final String JDBC_DRIVER = "oracle.jdbc.driver.OracleDriver";
    static final String DB_URL= "jdbc:oracle:thin:@localhost:1521:xe";
    static final String USER="dbms_200042118";
    static final String PASS="cse4308";
    public static void main (String args[])
    {
        Connection conn=null;
        Statement stmt=null;
        try
        {
            Class.forName(JDBC_DRIVER);
            System.out.println("Connecting to database");
            conn=DriverManager.getConnection(DB_URL, USER, PASS);
            System.out.println("Creating statement");
            stmt=conn.createStatement();;
            ArrayList<Account> accounts = new ArrayList<Account>();
            String sql1="SELECT A_ID, SUM(AMOUNT) AS TOTAL_AMOUNT FROM
TRANSACTIONS GROUP BY A_ID ORDER BY A_ID";
            String sql2="SELECT A_ID, SUM(AMOUNT) AS TOTAL_CREDIT FROM
TRANSACTIONS WHERE TYPE = 0 GROUP BY A_ID ORDER BY A_ID";
            String sql3="SELECT A_ID, SUM(AMOUNT) AS TOTAL_DEBIT FROM
TRANSACTIONS WHERE TYPE = 1 GROUP BY A_ID ORDER BY A_ID";
            System.out.println("Executing the query of Task 4: \n" + sql1 +
"\n"+sql2+"\n"+sql3);
```



```
            System.out.println("The list of transactions which occurred in the
last 6 months of 2021 are:\n");
            ResultSet rs1=stmt.executeQuery(sql1);
            while(rs1.next())
            {
                int id=rs1.getInt("A_ID");
                long total_amount=rs1.getLong("TOTAL_AMOUNT");
                boolean flag = false;
                for(Account a : accounts)
                {
                    if(a.accountID == id)
                    {
                        a.total = a.total + total_amount;
                        flag = true;
                    }
                }
                if(flag == false);
                {
                    Account dummy = new Account(id);
                    dummy.total = total_amount;
                    accounts.add(dummy);
                }
            }
            ResultSet rs2=stmt.executeQuery(sql2);
            while(rs2.next())
            {
                int id=rs2.getInt("A_ID");
                long total_credit=rs2.getLong("TOTAL_CREDIT");
                boolean flag = false;
                for(Account a : accounts)
                {
                    if(a.accountID == id)
                    {
                        a.credit = a.credit + total_credit;
                        flag = true;
                    }
                }
                if(flag == false);
                {
                    Account dummy = new Account(id);
                    dummy.credit = total_credit;
                    accounts.add(dummy);
                }
            }
            ResultSet rs3=stmt.executeQuery(sql3);
```



```java
            while(rs3.next())
            {
                int id=rs3.getInt("A_ID");
                long total_debit=rs3.getLong("TOTAL_DEBIT");
                boolean flag = false;
                for(Account a : accounts)
                {
                    if(a.accountID == id)
                    {
                        a.debit = a.debit + total_debit;
                        flag = true;
                    }
                }
                if(flag == false);
                {
                    Account dummy = new Account(id);
                    dummy.debit = total_debit;
                    accounts.add(dummy);
                }
            }
            long cipCount=0;
            long vipCount=0;
            long opCount=0;
            long unclassifiedCount=0;
            for(Account a : accounts) {
                {
                    a.setBalance();
                    String type = a.getAccountType();
                    if(type == "CIP")
                    {
                        cipCount++;
                    }
                    else if(type=="VIP")
                    {
                        vipCount++;
                    }
                    else if(type == "OP")
                    {
                        opCount++;
                    }
                    else
                    {
                        unclassifiedCount++;
                    }
                }
            }
            System.out.println("The number of CIP accounts are "+cipCount+".\n");
            System.out.println("The number of VIP accounts are "+vipCount+".\n");
            System.out.println("The number of OP accounts are "+opCount+".\n");
            System.out.println("The number of accounts which don't fall under any 
of the given categories are "+unclassifiedCount+".\n");
            rs1.close();
            rs2.close();
            rs3.close();
            stmt.close();
            conn.close();
            System.out.println("Thank you for banking with us!");
        }
        catch(SQLException se)
        {
            se.printStackTrace();
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

Accounts Class

```
public class Account {
    public long balance=0;
    public long credit = 0;
    public long debit =0;
    public long total =0;
    public int accountID;
    public String getAccountType(){
        String str;
        if(total >=5000000 && balance>=1000000)
        {
            str = "CIP";
        }
        else if(balance>=500000 && balance<=900000 && total >=2500000 && total 
<=4500000)
        {
            str = "VIP";
        }
        else if(balance<=100000 && total <=1000000){
            str = "OP";
        }
        else{str = "Unclassified";}
        return str;
    }
    public void setBalance(){
        balance = credit - debit;
    }

    Account(int accountID){
        this.accountID = accountID;
    }
}
```

Explanation of Solution

- First, a connection is established with the database.
- Then three separate SQL queries are run, the first of which calculates the total sum of all transactions made by someone. The other two queries calculate the sum of total credit and total debit respectively.
- From these queries, data is extracted and stored in an Account Class. There the account balance and type can be determined.
- All instances of accounts present in the database are stored in a List. Traversing the list, the number of different types of accounts is counted.
- Then the final output is printed on Screen.

Problems

The most difficult hurdle to cross this lab was establishing a connection with the Oracle database. There was a problem with the installation of Oracle in the Lab pc which made it so that Oracle had to be reinstalled. The provided tasks were easy and intuitive to understand. Task 4 was a bit more difficult, but it truly highlighted the benefits of Java in running SQL queries, as it would've been far too complex to do using only SQL queries.

Findings

Using Java to run SQL queries is immensely beneficial. It allows us to organize data and allows a lot more flexibility and simplicity in the data operations. Needlessly complex SQL queries are no longer required. Data can be manipulated in a much more intuitive and simple way in Java, after extracting it from the database.

Conclusion

Overall, the lab was a positive learning experience. Crossing the hurdle of establishing a connection with the database was difficult, but now that it has been overcome, my skills in many fields have increased by a lot.