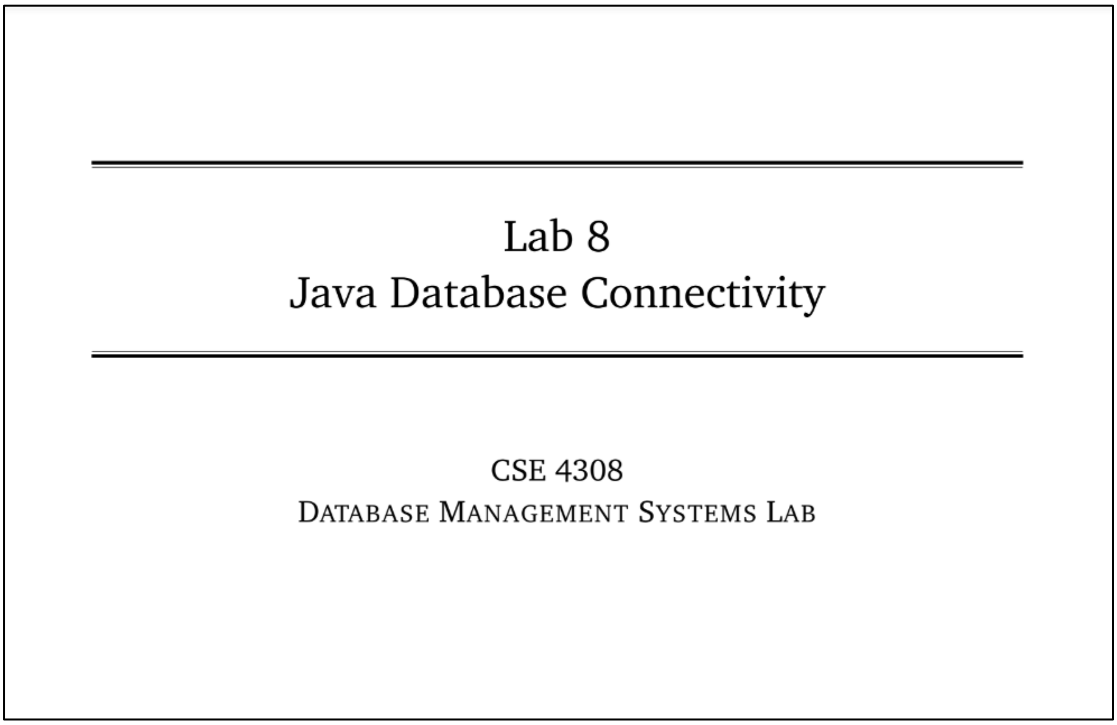
# LAB REPORT

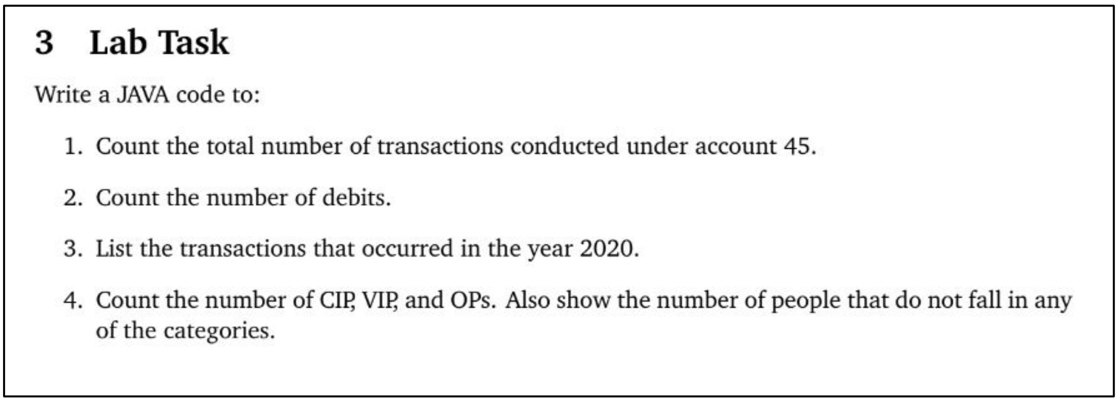


Lab 8
Java Database Connectivity

CSE 4308
DATABASE MANAGEMENT SYSTEMS LAB

**NAME: CHOWDHURY ASHFAQ**
**STUDENT ID: 200042123**
**PROGRAM: SWE**
**GROUP: 1A**
**DATE: 30/10/22**

# Tasks:



## 3 Lab Task

Write a JAVA code to:

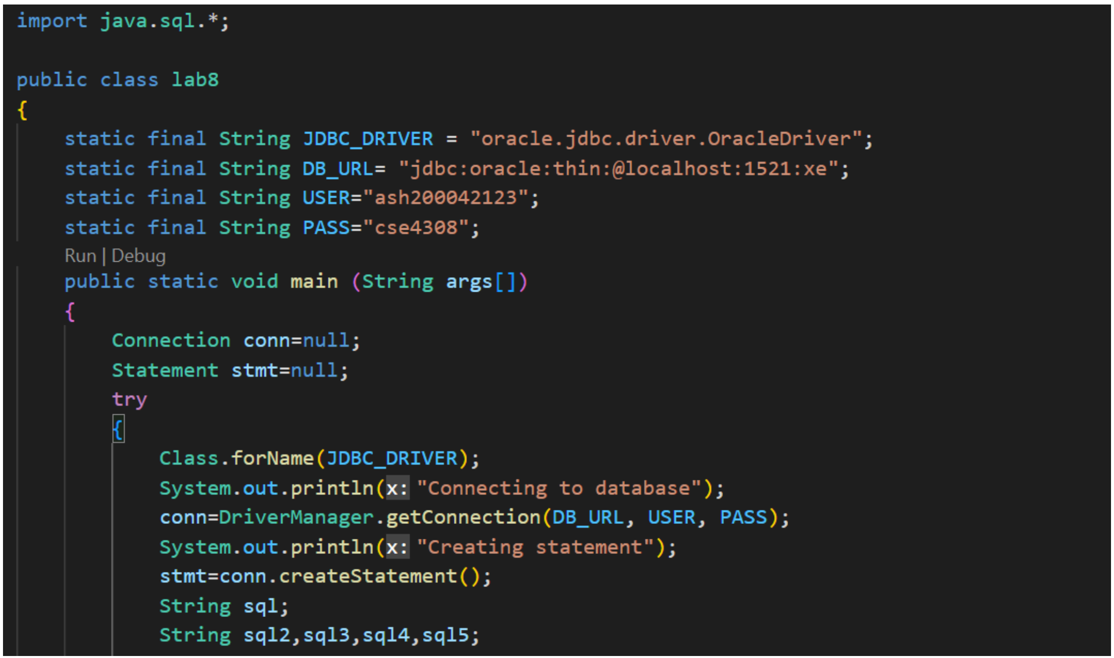
1. Count the total number of transactions conducted under account 45.
2. Count the number of debits.
3. List the transactions that occurred in the year 2020.
4. Count the number of CIP, VIP, and OPs. Also show the number of people that do not fall in any of the categories.

# Analysis of the problem:

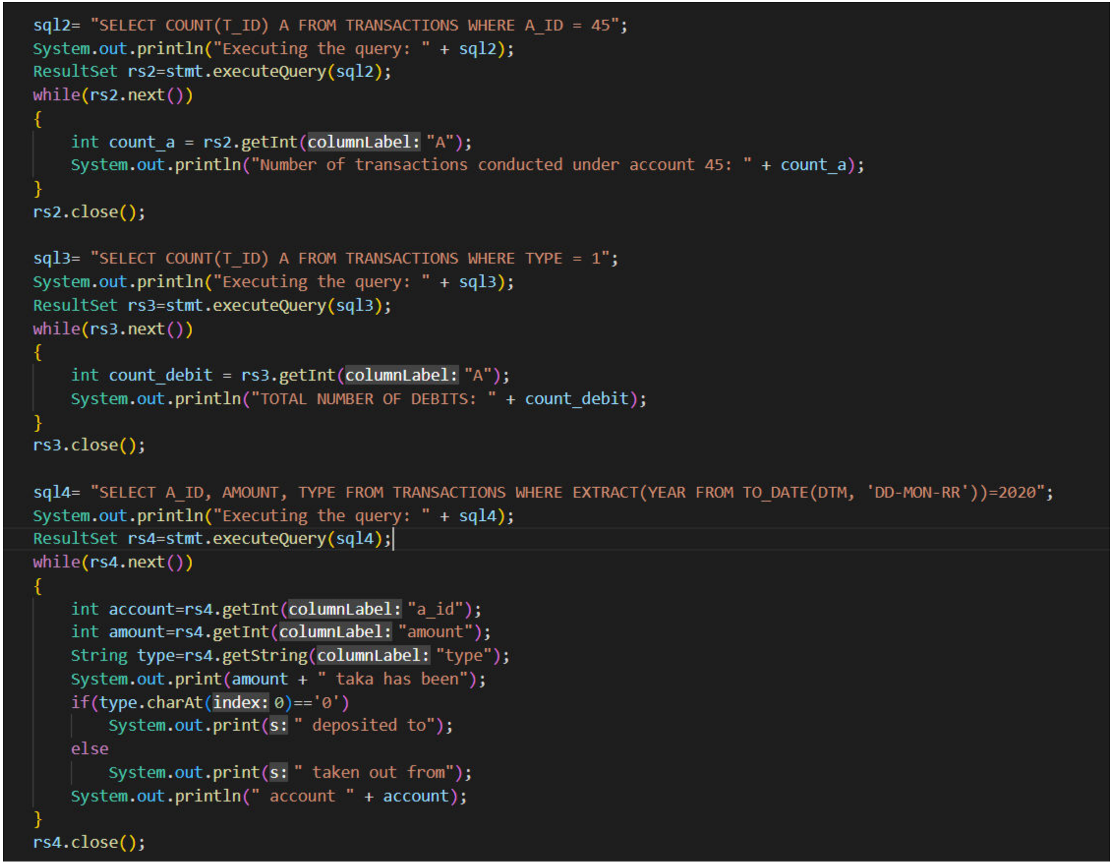
A database instance was given and we had to extract data with various conditions. The job was told to be done through Java.

To solve the tasks we need to setup the environment of IDE at first. The procedure was mentioned in the Lab manual. Following the procedure we set up the environment and also execute the table.sql file.

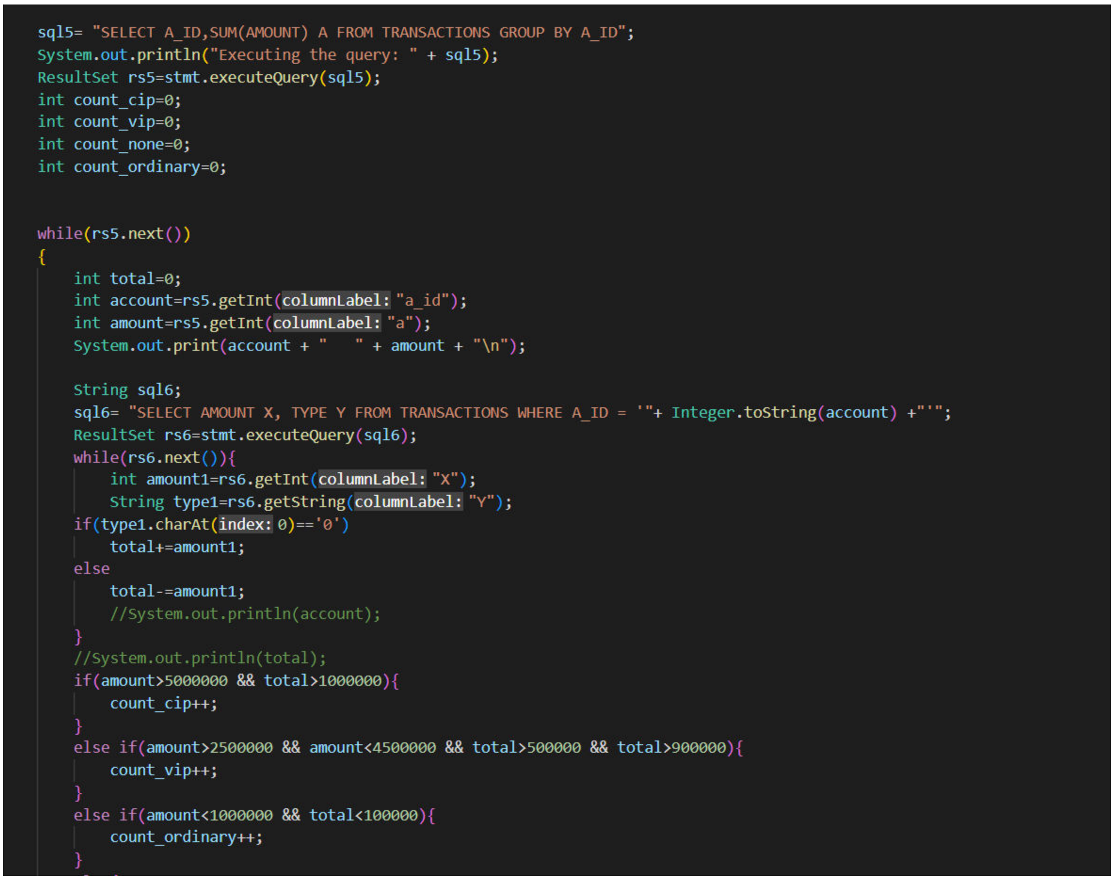
# Solution:



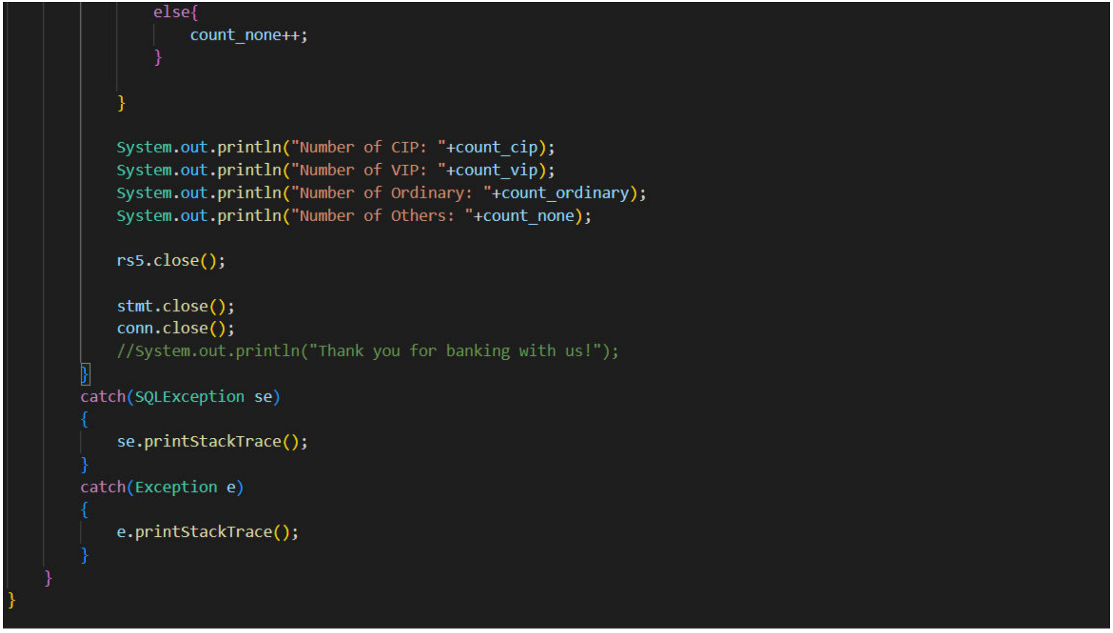
```
import java.sql.*;

public class lab8
{
    static final String JDBC_DRIVER = "oracle.jdbc.driver.OracleDriver";
    static final String DB_URL= "jdbc:oracle:thin:@localhost:1521:xe";
    static final String USER="ash200042123";
    static final String PASS="cse4308";
    Run | Debug
    public static void main (String args[])
    {
        Connection conn=null;
        Statement stmt=null;
        try
        {
            Class.forName(JDBC_DRIVER);
            System.out.println(x: "Connecting to database");
            conn=DriverManager.getConnection(DB_URL, USER, PASS);
            System.out.println(x: "Creating statement");
            stmt=conn.createStatement();
            String sql;
            String sql2,sql3,sql4,sql5;
```



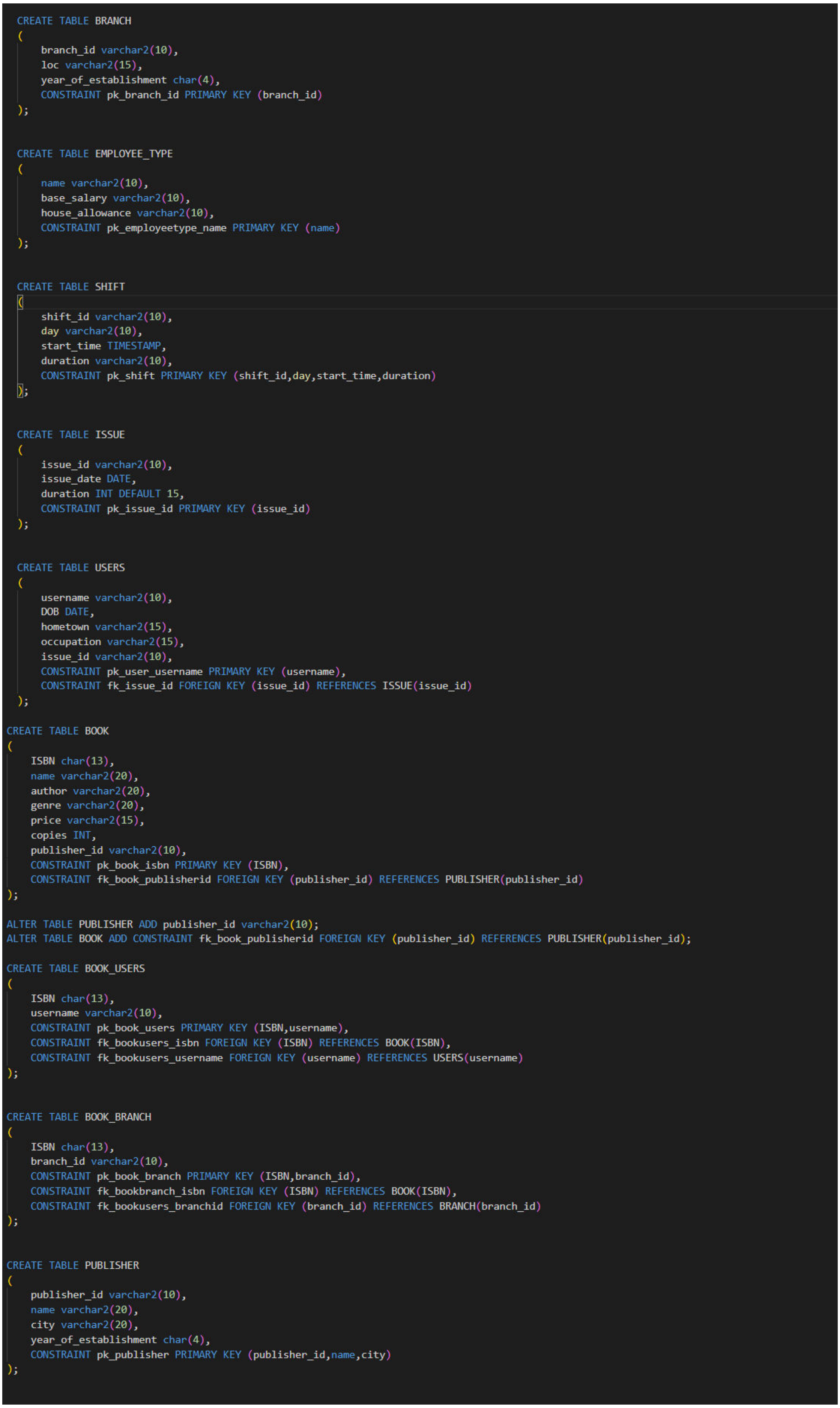
```
sql2= "SELECT COUNT(T_ID) A FROM TRANSACTIONS WHERE A_ID = 45";
System.out.println("Executing the query: " + sql2);
ResultSet rs2=stmt.executeQuery(sql2);
while(rs2.next())
{
    int count_a = rs2.getInt(columnLabel: "A");
    System.out.println("Number of transactions conducted under account 45: " + count_a);
}
rs2.close();

sql3= "SELECT COUNT(T_ID) A FROM TRANSACTIONS WHERE TYPE = 1";
System.out.println("Executing the query: " + sql3);
ResultSet rs3=stmt.executeQuery(sql3);
while(rs3.next())
{
    int count_debit = rs3.getInt(columnLabel: "A");
    System.out.println("TOTAL NUMBER OF DEBITS: " + count_debit);
}
rs3.close();

sql4= "SELECT A_ID, AMOUNT, TYPE FROM TRANSACTIONS WHERE EXTRACT(YEAR FROM TO_DATE(DTM, 'DD-MON-RR'))=2020";
System.out.println("Executing the query: " + sql4);
ResultSet rs4=stmt.executeQuery(sql4);
while(rs4.next())
{
    int account=rs4.getInt(columnLabel: "a_id");
    int amount=rs4.getInt(columnLabel: "amount");
    String type=rs4.getString(columnLabel: "type");
    System.out.print(amount + " taka has been");
    if(type.charAt(index: 0)=='0')
        System.out.print(s: " deposited to");
    else
        System.out.print(s: " taken out from");
    System.out.println(" account " + account);
}
rs4.close();
```



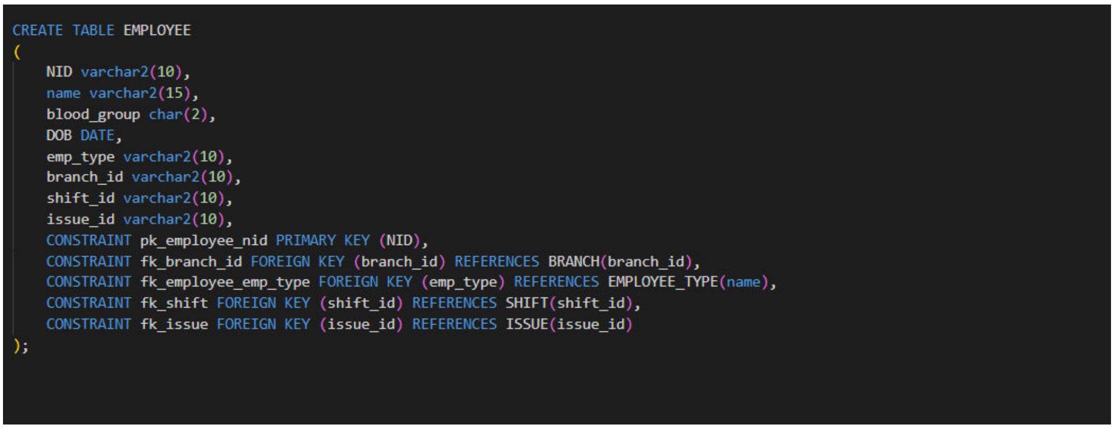
```
sql5= "SELECT A_ID,SUM(AMOUNT) A FROM TRANSACTIONS GROUP BY A_ID";
System.out.println("Executing the query: " + sql5);
ResultSet rs5=stmt.executeQuery(sql5);
int count_cip=0;
int count_vip=0;
int count_none=0;
int count_ordinary=0;


while(rs5.next())
{
    int total=0;
    int account=rs5.getInt(columnLabel: "a_id");
    int amount=rs5.getInt(columnLabel: "a");
    System.out.print(account + "    " + amount + "\n");

    String sql6;
    sql6= "SELECT AMOUNT X, TYPE Y FROM TRANSACTIONS WHERE A_ID = '"+ Integer.toString(account) +"'";
    ResultSet rs6=stmt.executeQuery(sql6);
    while(rs6.next()){
        int amount1=rs6.getInt(columnLabel: "X");
        String type1=rs6.getString(columnLabel: "Y");
    if(type1.charAt(index: 0)=='0')
        total+=amount1;
    else
        total-=amount1;
        //System.out.println(account);
    }
    //System.out.println(total);
    if(amount>5000000 && total>1000000){
        count_cip++;
    }
    else if(amount>2500000 && amount<4500000 && total>500000 && total>900000){
        count_vip++;
    }
    else if(amount<1000000 && total<100000){
        count_ordinary++;
    }
```



```
                else{
                    count_none++;
                }

            }

            System.out.println("Number of CIP: "+count_cip);
            System.out.println("Number of VIP: "+count_vip);
            System.out.println("Number of Ordinary: "+count_ordinary);
            System.out.println("Number of Others: "+count_none);

            rs5.close();

            stmt.close();
            conn.close();
            //System.out.println("Thank you for banking with us!");
        }
        catch(SQLException se)
        {
            se.printStackTrace();
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```



```sql
CREATE TABLE BRANCH
(
    branch_id varchar2(10),
    loc varchar2(15),
    year_of_establishment char(4),
    CONSTRAINT pk_branch_id PRIMARY KEY (branch_id)
);


CREATE TABLE EMPLOYEE_TYPE
(
    name varchar2(10),
    base_salary varchar2(10),
    house_allowance varchar2(10),
    CONSTRAINT pk_employeetype_name PRIMARY KEY (name)
);


CREATE TABLE SHIFT
(
    shift_id varchar2(10),
    day varchar2(10),
    start_time TIMESTAMP,
    duration varchar2(10),
    CONSTRAINT pk_shift PRIMARY KEY (shift_id,day,start_time,duration)
);


CREATE TABLE ISSUE
(
    issue_id varchar2(10),
    issue_date DATE,
    duration INT DEFAULT 15,
    CONSTRAINT pk_issue_id PRIMARY KEY (issue_id)
);


CREATE TABLE USERS
(
    username varchar2(10),
    DOB DATE,
    hometown varchar2(15),
    occupation varchar2(15),
    issue_id varchar2(10),
    CONSTRAINT pk_user_username PRIMARY KEY (username),
    CONSTRAINT fk_issue_id FOREIGN KEY (issue_id) REFERENCES ISSUE(issue_id)
);

CREATE TABLE BOOK
(
    ISBN char(13),
    name varchar2(20),
    author varchar2(20),
    genre varchar2(20),
    price varchar2(15),
    copies INT,
    publisher_id varchar2(10),
    CONSTRAINT pk_book_isbn PRIMARY KEY (ISBN),
    CONSTRAINT fk_book_publisherid FOREIGN KEY (publisher_id) REFERENCES PUBLISHER(publisher_id)
);

ALTER TABLE PUBLISHER ADD publisher_id varchar2(10);
ALTER TABLE BOOK ADD CONSTRAINT fk_book_publisherid FOREIGN KEY (publisher_id) REFERENCES PUBLISHER(publisher_id);

CREATE TABLE BOOK_USERS
(
    ISBN char(13),
    username varchar2(10),
    CONSTRAINT pk_book_users PRIMARY KEY (ISBN,username),
    CONSTRAINT fk_bookusers_isbn FOREIGN KEY (ISBN) REFERENCES BOOK(ISBN),
    CONSTRAINT fk_bookusers_username FOREIGN KEY (username) REFERENCES USERS(username)
);


CREATE TABLE BOOK_BRANCH
(
    ISBN char(13),
    branch_id varchar2(10),
    CONSTRAINT pk_book_branch PRIMARY KEY (ISBN,branch_id),
    CONSTRAINT fk_bookbranch_isbn FOREIGN KEY (ISBN) REFERENCES BOOK(ISBN),
    CONSTRAINT fk_bookusers_branchid FOREIGN KEY (branch_id) REFERENCES BRANCH(branch_id)
);


CREATE TABLE PUBLISHER
(
    publisher_id varchar2(10),
    name varchar2(20),
    city varchar2(20),
    year_of_establishment char(4),
    CONSTRAINT pk_publisher PRIMARY KEY (publisher_id,name,city)
);
```



```
CREATE TABLE EMPLOYEE
(
    NID varchar2(10),
    name varchar2(15),
    blood_group char(2),
    DOB DATE,
    emp_type varchar2(10),
    branch_id varchar2(10),
    shift_id varchar2(10),
    issue_id varchar2(10),
    CONSTRAINT pk_employee_nid PRIMARY KEY (NID),
    CONSTRAINT fk_branch_id FOREIGN KEY (branch_id) REFERENCES BRANCH(branch_id),
    CONSTRAINT fk_employee_emp_type FOREIGN KEY (emp_type) REFERENCES EMPLOYEE_TYPE(name),
    CONSTRAINT fk_shift FOREIGN KEY (shift_id) REFERENCES SHIFT(shift_id),
    CONSTRAINT fk_issue FOREIGN KEY (issue_id) REFERENCES ISSUE(issue_id)
);
```

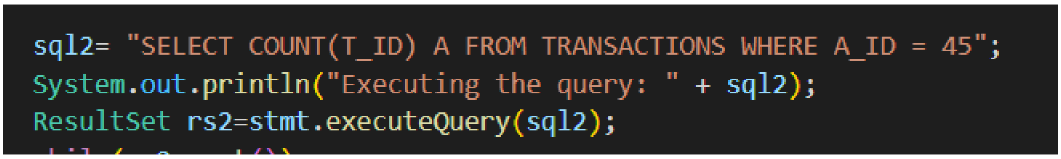
# Explanation:

At first we need to make the connection with the database. The DB_URL, User and password is used for this reason.

The ‘conn=DriverManager.getConnection(DB_URL, USER, PASS);

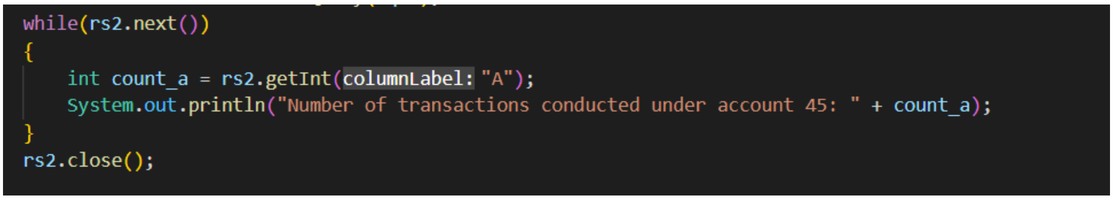
Part makes the connection with database.

Next we need to write the required sql codes in a string which are named as sql1,sql2, sql3, sql4 and sql5. Next we execute the sql code and the results are stored in rs2, rs3, rs4, rs5 respectively.



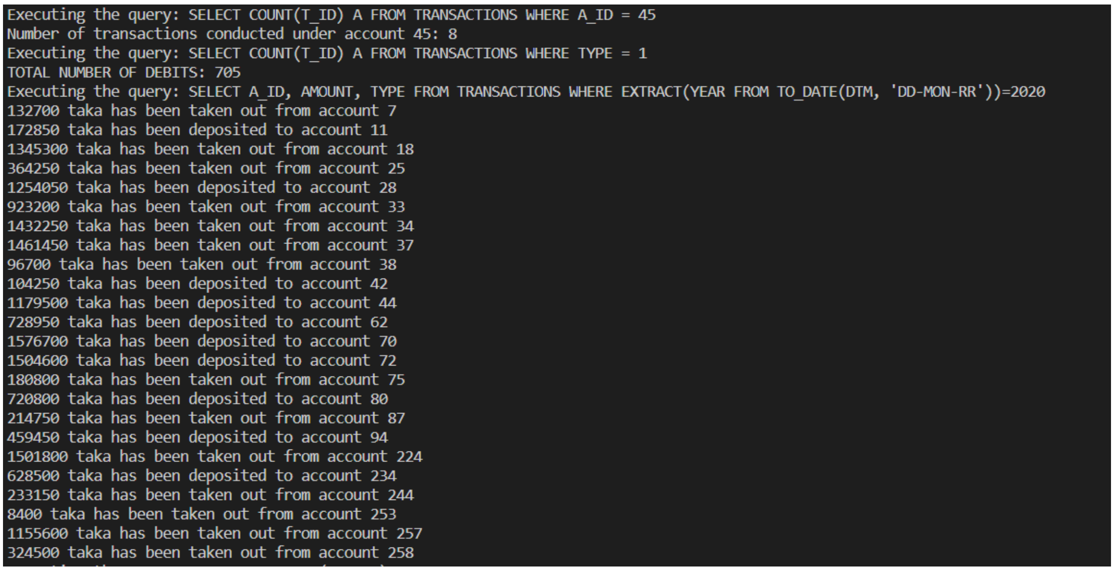
```
sql2= "SELECT COUNT(T_ID) A FROM TRANSACTIONS WHERE A_ID = 45";
System.out.println("Executing the query: " + sql2);
ResultSet rs2=stmt.executeQuery(sql2);
```

Next we use the data stored in ResultSet. A loop is executed which runs until there’s some data in the Resultset. After our work with a particular ResultSet is done we close that ResultSet.



```
while(rs2.next())
{
    int count_a = rs2.getInt(columnLabel: "A");
    System.out.println("Number of transactions conducted under account 45: " + count_a);
}
rs2.close();
```

## OUTPUT:



```
Executing the query: SELECT COUNT(T_ID) A FROM TRANSACTIONS WHERE A_ID = 45
Number of transactions conducted under account 45: 8
Executing the query: SELECT COUNT(T_ID) A FROM TRANSACTIONS WHERE TYPE = 1
TOTAL NUMBER OF DEBITS: 705
Executing the query: SELECT A_ID, AMOUNT, TYPE FROM TRANSACTIONS WHERE EXTRACT(YEAR FROM TO_DATE(DTM, 'DD-MON-RR'))=2020
132700 taka has been taken out from account 7
172850 taka has been deposited to account 11
1345300 taka has been taken out from account 18
364250 taka has been taken out from account 25
1254050 taka has been deposited to account 28
923200 taka has been taken out from account 33
1432250 taka has been taken out from account 34
1461450 taka has been taken out from account 37
96700 taka has been taken out from account 38
104250 taka has been deposited to account 42
1179500 taka has been deposited to account 44
728950 taka has been deposited to account 62
1576700 taka has been deposited to account 70
1504600 taka has been deposited to account 72
180800 taka has been taken out from account 75
720800 taka has been deposited to account 80
214750 taka has been taken out from account 87
459450 taka has been deposited to account 94
1501800 taka has been taken out from account 224
628500 taka has been deposited to account 234
233150 taka has been taken out from account 244
8400 taka has been taken out from account 253
1155600 taka has been taken out from account 257
324500 taka has been taken out from account 258
```

## Interesting Findings:

Java and Sql together is very powerful.