NAME: NAZIA KARIM KHAN OISHEE

STUDENT ID: 200042137

COURSE CODE: CSE 4308

SUBJECT: DATABASE MANAGEMENT SYSTEMS LAB

Introduction

Our task in this lab was to use JDBC to conduct sql queries through Java.

Java Database Connectivity (JDBC) is an application programming interface for Java through which we can access a database. It is a Java-based data access technology used for Java database connectivity.

Our task was divided into two parts. Firstly, we were to set up the environment.Secondly, write java code to complete queries.

Task1

We were given a TableGenerator.cpp file. After executing the file it asked to enter a student ID. After providing student id a table.sql sql file was created.The SQL file contains the required SQL queries to create and populate the database.

```
#include <cstdio>
#include <cstdlib>
#include <ctime>

bool leap(int year)
{
    if((year%400==0 || year%100) &&(year%4==0))
        return true;
    else
        return false;
}

int main()
{
    printf("Enter your student ID: ");
    int ID;
    scanf("%d", &ID);
    srand(ID);
    freopen("table.sql", "w", stdout);
    int a_id;
    int t_id=1;
    printf("DROP TABLE TRANSACTIONS CASCADE CONSTRAINTS;\n");
    printf("DROP TABLE ACCOUNT CASCADE CONSTRAINTS;\n");
    printf("CREATE TABLE ACCOUNT\n(\n\tA_ID NUMBER,\n\tCONSTRAINT PK_ACCOUNT
PRIMARY KEY (A_ID)\n);\n");
    printf("\n");
    printf("CREATE TABLE TRANSACTIONS\n(\n\tT_ID NUMBER,\n\tDTM DATE,\n\tA_ID
NUMBER,\n\tAMOUNT NUMBER,\n\tTYPE CHAR(1),\n\tCONSTRAINT PK_TRANSACTIONS
PRIMARY KEY (T_ID),\n\tCONSTRAINT FK_TRANSACTIONS_ACCOUNT FOREIGN KEY (A_ID)
REFERENCES ACCOUNT\n);\n");
    printf("\n\n");
    int no_of_accounts=100+rand()%200;
    for(int i=0; i<no_of_accounts; i++)
        printf("INSERT INTO ACCOUNT VALUES(%d);\n", i+1);
    printf("\n");
    for(int i=0; i<no_of_accounts; i++)
    {
        int times=rand()%10+1;
        for(int k=0; k<times; k++)
        {
            int amount=rand();
            int year=rand()%23+2000;
            int month=rand()%12+1;
            int day=rand()%28+1;
            if(rand()%2)
            {
                if(month==2)
                {
                    if(leap(year))
                        day++;
                }
                else
                {
                    if(month==4 || month==6 || month==9 || month==11)
                        day+=2;
                    else
                        day+=3;
                }
            }
            int hour=rand()%24;
            int minute=rand()%60;
            int second=rand()%60;
            int type=rand()%2;
            printf("INSERT INTO TRANSACTIONS VALUES(%d, TO_DATE('%d/%02d/%02d
%02d:%02d:%02d', 'yyyy/mm/dd hh24:mi:ss'), %d, %d, %d);\n", t_id++, year,
month, day, hour, minute, second, i+1, amount*50, type);
        }
    }
    printf("COMMIT;\n");

    return 0;
}
```

TableGenerator.cpp

Then our task was to login to our oracle database and create or connect to a user.I created a new user and then entered the directory of table.sql file.

After that I installed the Java Development Kit (JDK) using jdk-19_windows-x64_bin.msi.It took me a bit of time to do so due to interrupted internet connection.

After that I used Visual Studio Code IDE to complete the task.We were given an ojdbc14.jar file as an external JAR file.After creating a new Java project I added the jar file to the project.

We were given a solution.java file which contained fragments of code to understand how to actually implement JDBC.

Task2

1. Count the total number of transactions conducted under account 45.

Solution:

```
import java.sql.*;

public class Task1
{
    static final String JDBC_DRIVER = "oracle.jdbc.driver.OracleDriver";
    static final String DB_URL= "jdbc:oracle:thin:@localhost:1521:xe";
    static final String USER="SID_137";
    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,COUNT(T_ID) FROM TRANSACTIONS WHERE A_ID = 45
GROUP BY A_ID";
            System.out.println("Executing the query: " + sql);
            ResultSet rs=stmt.executeQuery(sql);
            while(rs.next())
            {
                int account=rs.getInt("a_id");
                int count=rs.getInt("COUNT(T_ID)");
                System.out.print(account + " has made ");
              System.out.println(count+"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:

We were provided with a specific scenario.According to the scenario I performed the task.

I followed the given solution.java file to understand the syntax. I changed the user and password and put the username under which I created the table. Here my username was SID_137.

In JDBC we write queries as string and then connect the query with the database using conn.createStatement();.

In this task I counted the occurrences of account id 45 . The count depicts the transactions conducted under account 45.

2. Count the number of debits.

Solution:

```
import java.sql.*;

public class Task2
{
    static final String JDBC_DRIVER = "oracle.jdbc.driver.OracleDriver";
    static final String DB_URL= "jdbc:oracle:thin:@localhost:1521:xe";
    static final String USER="SID_137";
    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(T_ID) FROM TRANSACTIONS WHERE TYPE=1";
            System.out.println("Executing the query: " + sql);
            ResultSet rs=stmt.executeQuery(sql);
            while(rs.next())
            {
              int count=rs.getInt("COUNT(T_ID)");
              System.out.println(count+" 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:

According to our given scenario the type would be 1 if the money is debited from the account, i.e. the money is subtracted from the account.
So I simply counted the number of transactions where type was “1”.

3. List the transactions that occurred in the year 2020.

Solution:

```
import java.sql.*;

public class Task3
{
    static final String JDBC_DRIVER = "oracle.jdbc.driver.OracleDriver";
    static final String DB_URL= "jdbc:oracle:thin:@localhost:1521:xe";
    static final String USER="SID_137";
    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,A_ID,AMOUNT,TYPE FROM TRANSACTIONS WHERE
EXTRACT(YEAR FROM DTM)='2020'";
            System.out.println("Executing the query: " + sql);
            ResultSet rs=stmt.executeQuery(sql);
            while(rs.next())
            {
                int account=rs.getInt("a_id");
                int transaction =rs.getInt("t_id");
                int amount=rs.getInt("amount");
                String type=rs.getString("type");
                System.out.print(amount + " taka has been");
                if(type.charAt(0)=='0')
                    System.out.print(" deposited to");
                else
                    System.out.print(" taken out from");
                System.out.println(" account" + account + ", transaction id is"
+ transaction);
            }

            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:

In this task I used the Extract function to extract the year from the date and selected the transactions' information where the date fell in 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.

Solution:

```
import java.sql.*;

public class Task4
{
    static final String JDBC_DRIVER = "oracle.jdbc.driver.OracleDriver";
    static final String DB_URL= "jdbc:oracle:thin:@localhost:1521:xe";
    static final String USER="SID_137";
    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 CountCIP,CountVIP,CountOP,NewTable,totalAccountsSql;
            NewTable = "CREATE OR REPLACE VIEW NewTable AS SELECT A_ID ,
(CREDIT - DEBIT) AS BALANCE  , (CREDIT+DEBIT) AS TOTAL FROM (( SELECT A_ID ,
SUM(AMOUNT) AS DEBIT FROM TRANSACTIONS WHERE TYPE='1' GROUP BY A_ID) NATURAL
JOIN ( SELECT A_ID , SUM(AMOUNT) AS CREDIT FROM TRANSACTIONS WHERE TYPE='0'
GROUP BY A_ID))";
            stmt.executeUpdate(NewTable);
            CountCIP = "SELECT COUNT(*) AS CIP FROM NewTable WHERE
BALANCE>=1000000 AND TOTAL>=5000000";
            CountVIP = "SELECT COUNT(*) AS VIP FROM NewTable    WHERE
BALANCE>=500000 AND BALANCE<=900000 AND TOTAL>=2500000 AND TOTAL<=4500000";
            CountOP = "SELECT COUNT(*) AS OP    FROM NewTable WHERE
BALANCE<100000 AND TOTAL<1000000";
            ResultSet cip=stmt.executeQuery(CountCIP);
            cip.next();
            int numberOfCIP = cip.getInt("CIP");
            ResultSet vip =stmt.executeQuery(CountVIP);
            vip.next();
            int numberOfVIP = vip.getInt("VIP");
            ResultSet op = stmt.executeQuery(CountOP);
            op.next();
            int numberOfOP = op.getInt("OP");
            totalAccountsSql = "SELECT COUNT(UNIQUE A_ID) AS TOTAL_ACCOUNTS
FROM ACCOUNT";
            ResultSet total = stmt.executeQuery(totalAccountsSql);
            total.next();
            int totalNumber = total.getInt("TOTAL_ACCOUNTS");
            System.out.println("NUMBER OF CIP: " + numberOfCIP + "\n");
            System.out.println("NUMBER OF VIP: " + numberOfVIP + "\n");
            System.out.println("NUMBER OF OP: " + numberOfOP + "\n");
            System.out.println("OTHER: " + (totalNumber -
(numberOfCIP+numberOfOP+numberOfVIP)));
            cip.close();
            vip.close();
            op.close();
            total.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:

For this task I created a view that would store information regarding account id and their total balance and number of transactions. From this view I counted the number of Commercially Important Person(CIP),Very Important Person (VIP),Ordinary Person (OP) based on the given constraints.

Then I subtracted the total number of CIP,VIP and OP from the total number of accounts to find the number of people that do not fall in any of the categories.

Problems I faced during the task:

At first I faced some difficulties downloading the JDK and then while including the jar file. As JDBC is a new concept to me it felt like too many complex instructions to follow. But gradually I could follow all the steps and include the jar file.

After that I faced problems understanding the syntax. It took me quite a while to get the syntax of Java as I am not familiar with Java much.

The first three queries were easy to understand and complete. I could complete it in the lab hour itself. But due to time constraints I could not complete the fourth task in lab hours. Later I did that on my own.

I faced some difficulties while doing the fourth task. The queries would run perfectly in my SQL command prompt . But for some reason in VS Code it was showing me the error “Invalid Column Name”. I tried to resolve it on my own.Then I got to know about the next() function and resolved the problem.