Nikit Gokhe
Class – Comp D1
Roll No. 224024
Gr. No. 21810522

Assignment No.3

AIM: Use SQL database for student data to perform the following operation through java using JDBC.

1. Insert
2. Delete
3. Update
4. Search
5. Display all

THEORY:

JDBC stands for Java Database Connectivity. JDBC is a Java API to connect and execute the query with the database. It is a part of JavaSE (Java Standard Edition). JDBC API uses JDBC drivers to connect with the database. There are four types of JDBC drivers:

- JDBC-ODBC Bridge Driver,
- Native Driver,
- Network Protocol Driver, and
- Thin Driver

Java Database Connectivity with MySQL

To connect a Java application with the MySQL database, we need to follow 5 following steps.

In this example we are using MySql as the database. So we need to know following informations for the mysql database:

1. **Driver class:** The driver class for the mysql database is **com.mysql.jdbc.Driver**.

2. **Connection URL:** The connection URL for the mysql database is **jdbc:mysql://localhost:3306/sonoo** where jdbc is the API, mysql is the database, localhost is the server name on which mysql is running, we may also use IP address, 3306 is the port number and sonoo is the database name. We may use any database, in such case, we need to replace the sonoo with our database name.
3. **Username:** The default username for the mysql database is **root**.
4. **Password:** It is the password given by the user at the time of installing the mysql database. In this example, we are going to use root as the password.

Why Should We Use JDBC

Before JDBC, ODBC API was the database API to connect and execute the query with the database. But, ODBC API uses ODBC driver which is written in C language (i.e. platform dependent and unsecured). That is why Java has defined its own API (JDBC API) that uses JDBC drivers (written in Java language).

We can use JDBC API to handle database using Java program and can perform the following activities:

1. Connect to the database
2. Execute queries and update statements to the database
3. Retrieve the result received from the database.

CODE:

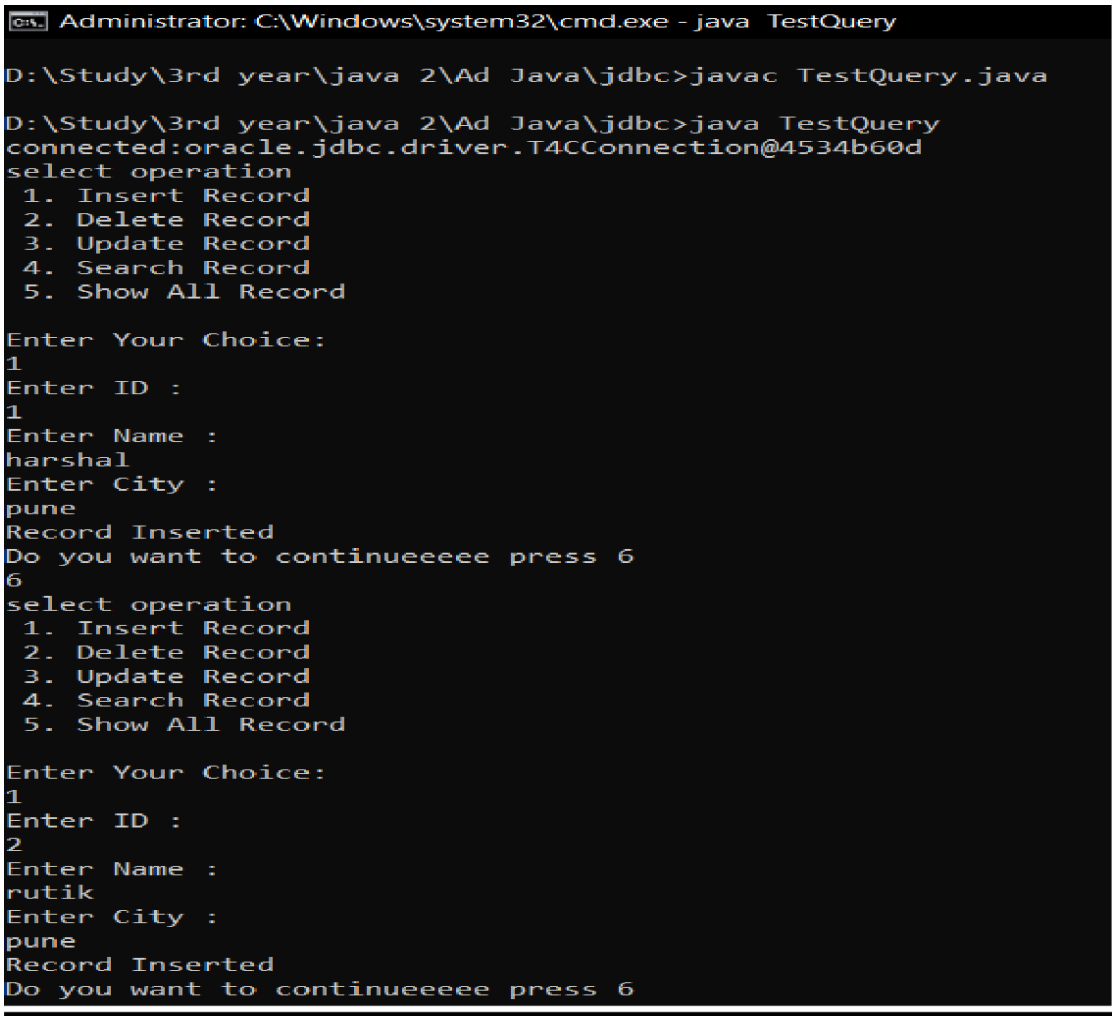
```
import java.sql.*;
import javax.sql.*;
import java.util.*;

public class TestQuery
{       public static void main(String args[])
```

```
{
        Scanner sc = new Scanner(System.in);
        try
        {
                Class.forName("oracle.jdbc.OracleDriver");

                Connection cn =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","r
oot");
            System.out.println("connected:"+cn);

                PreparedStatement ps = null;
                ResultSet rs=null;
            int id,i,c;
                String name, city;

                do
                {
                System.out.println("select operation\n 1. Insert Record\n 2.
Delete Record\n 3. Update Record\n 4. Search Record\n 5. Show All Record\n");
                        System.out.println("Enter Your Choice:");
                        int ch = sc.nextInt();
                        switch(ch)
                        {
                                case 1:
                                System.out.println("Enter ID :");
                                id=sc.nextInt();
                                System.out.println("Enter Name :");
                                name = sc.next();
                                System.out.println("Enter City :");
                                city = sc.next();
                        ps = cn.prepareStatement("insert into demo
values(?,?,?)");
                        ps.setInt(1,id);
                        ps.setString(2,name);
```

```
                    ps.setString(3,city);
                    i = ps.executeUpdate();
                    if(i>0)
                    {
                        System.out.println("Record Inserted");
                    }
                    break;
                    case 2:
                    System.out.println("Enter ID to delete Record");
                    id=sc.nextInt();
                    ps=cn.prepareStatement("delete from demo where
ID=?");
                    ps.setInt(1,id);
                    i = ps.executeUpdate();
                    if(i>0)
                    {
                        System.out.println("Record Deleted");
                    }
                    break;
                    case 3:
                    System.out.println("Enter Id to Update Record");
                    id=sc.nextInt();
                    System.out.println("Enter Name");
                    name=sc.next();
                    ps= cn.prepareStatement("update demo set NAME=?
where ID=?");
                    ps.setString(1,name);
                    ps.setInt(2,id);
                    i = ps.executeUpdate();
                    if(i>0)
                    {
                        System.out.println("Record Updated");
                    }
                    break;
                    case 4:
```

```
                        System.out.println("Enter ID to Search Record");
                        id= sc.nextInt();
                        ps=cn.prepareStatement("select * from demo where
ID=?");
                        ps.setInt(1,id);
                        rs = ps.executeQuery();
                        rs.next();
        System.out.println(rs.getInt(1)+"\t"+rs.getString(2)+"\t"+rs.getString(3));
                        break;
                        case 5:
                        String str="select * from demo";
                        Statement st = cn.createStatement();
                        rs = st.executeQuery(str);
                        ResultSetMetaData rmd = rs.getMetaData();
                        System.out.println("\n_____");
                        for(int j=1;j<=rmd.getColumnCount();j++)
                        {
                                System.out.print(rmd.getColumnName(j)+"\t");
                        }
                        System.out.println("\n_____");
                        while(rs.next())
                        {

        System.out.println(rs.getInt(1)+"\t"+rs.getString(2)+"\t"+rs.getString(3));
                        }
                }
                System.out.println("Do you want to continueeeee press 6");
                c = sc.nextInt();
                }while(c==6);
        }
        catch(Exception e)
        {
                System.out.println(e);
        }
}}
```

Output



CONCLUSION:

This *JDBC Java* assignment describes how to use *JDBC* API to create, insert into, update, and query tables. We also learned how to use simple and prepared statements in Java.