**Name : Prashant Maurya**
**empID: 12417**
**KIIT Roll : 1828259**

## 1.Write a program to insert 10 more items in the account_details table using JDBC.

Sol

```
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.Scanner;

class D6Ex1Pojo{

	int accId;
	String name;
	String branch;
	public int getAccId() {
		return accId;
	}
	public void setAccId(int accId) {
		this.accId = accId;
	}
	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}
	public String getBranch() {
		return branch;
	}
	public void setBranch(String branch) {
		this.branch = branch;
	}

}

//========================================================================
=
public class Day6JDBCex1 {

        static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
        static final String DB_URL = "jdbc:mysql://localhost/test";

        static final String USER = "root";
        static final String PASS = "root";


        public static ArrayList<D6Ex1Pojo> getVal() throws IOException{
                ArrayList<D6Ex1Pojo> ar = new ArrayList<D6Ex1Pojo>();
                Scanner scanner = new Scanner(System.in);
                System.out.println("enter number of entries:");
                int n = scanner.nextInt();

                for(int i=0;i<n;i++)
                {
                        System.out.println("enter branchID \n name\n branch city");
                        D6Ex1Pojo obj = new D6Ex1Pojo();
                        obj.setAccId(scanner.nextInt());
                        obj.setName(scanner.next());
                        obj.setBranch(scanner.next());

                        ar.add(obj);
                }
                return ar;
        }

//==========================================================================
=


        public static void addDetails(ArrayList<D6Ex1Pojo> ar) throws ClassNotFoundException
{

                Connection con = null;

                Statement st = null;

                try {

                        Class.forName("com.mysql.cj.jdbc.Driver");
```

```
			con = DriverManager.getConnection(DB_URL,USER,PASS);

			st = con.createStatement();

			for(D6Ex1Pojo o:ar)
			{
				String sql1 = "INSERT INTO account_detail VALUES
("+o.getAccId()+","+o.getName()+","+o.getBranch()+");";
				st.executeUpdate(sql1);
				System.out.println("inseted in table");
			}
		}
		catch (SQLException e) {
			// TODO: handle exception
			e.printStackTrace();
			System.out.println("something went wrong");
		}
	}


//================================================================
=

	public static void main(String[] args) throws IOException, ClassNotFoundException {
		// TODO Auto-generated method stub

		ArrayList<D6Ex1Pojo> list = new ArrayList<D6Ex1Pojo>();

		list = getVal();

		addDetails(list);

	}

}
```

**============================================================**

**2.Write a program which searches a record in a table using JDBC.**

**Or**

**3.Write a program to print the records in Descending Order using JDBC.**

**Or**

**4.Write a program that delete a particular record from the table using JDBC**

Sol

```
package com.maurya.prashant;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.Scanner;



class PojoDay6Ex{

        int accId;
        String name;
        String Branch;
        public int getAccId() {
                return accId;
        }
        public void setAccId(int accId) {
                this.accId = accId;
        }
        public String getName() {
                return name;
        }
        public void setName(String name) {
                this.name = name;
```

```
	}
	public String getBranch() {
		return Branch;
	}
	public void setBranch(String branch) {
		Branch = branch;
	}
	@Override
	public String toString() {
		return "PojoDay6Ex [accId=" + accId + ", name=" + name + ",
Branch=" + Branch + "]";
	}

}

//===================================================
===========================

public class Day6ExJDBC {

	private static Connection con = null;
	static Statement st = null;

	private static Scanner sc = new Scanner(System.in);

	static final String JDBC_DRIVER = "com.mysql.cj.jdbc.Driver";
	static final String DB_URL = "jdbc:mysql://localhost/test";

	static final String USER = "root";
	static final String PASS = "root";


//
====================================================
==========================
    private static void insertRecord() throws IOException, SQLException {


        ArrayList<PojoDay6Ex> ar = new ArrayList<PojoDay6Ex>();

        BufferedReader br = new BufferedReader(new
FileReader("C:\\Users\\prashant.maurya\\Downloads\\hrc\\account
detail.csv"));
        String line = "";
        int count=0;
        while((line = br.readLine()) !=null)
        {
            PojoDay6Ex obj = new PojoDay6Ex();
            String[] eachLineDetail = line.split(",");
            obj.setAccId(Integer.parseInt(eachLineDetail[0]));
            obj.setName(eachLineDetail[1]);
            obj.setBranch(eachLineDetail[2]);

            ar.add(obj);
            count++;

        }
        System.out.println(count +" rows ready to be added");

        br.close();

        try {
            st = con.createStatement();

            for(PojoDay6Ex o:ar)
            {
                String sql1 = "INSERT INTO account_detail
VALUES ("+o.getAccId()+","+o.getName()+","+o.getBranch()+");";
```

```
                st.executeUpdate(sql1);
                System.out.println("inseted in table");
            }
        }
        catch (SQLException e) {
            // TODO: handle exception
            e.printStackTrace();
            System.out.println("something went wrong");
        }

        st.close();
        con.close();

    }

    //
=============================================================
==============================

    private static void printFetchedRecord(String sql) throws
SQLException {

        ArrayList<PojoDay6Ex> ar2 = new ArrayList<PojoDay6Ex>();

        try {
            st = con.createStatement();
            ResultSet rs = st.executeQuery(sql);

            while(rs.next())
            {
                PojoDay6Ex obj = new PojoDay6Ex();
                obj.setAccId(rs.getInt("Acc_ID"));
                obj.setName(rs.getString("Acc_holder_Name"));
                obj.setBranch(rs.getString("Branch"));
                ar2.add(obj);

            }

            if(ar2.isEmpty())
                System.out.println("no such entity in table!! ");

            System.out.println("data fetched \n");
            for(PojoDay6Ex o : ar2)
            {
                System.out.println(o.toString());
            }
            rs.close();
            st.close();
            con.close();


        }

        catch (Exception e) {
            // TODO: handle exception
            e.printStackTrace();
        }


    }

    private static void fetchRecord() throws SQLException{


        System.out.println("give on what basis you want too serach the data, enter a chice:");
        System.out.println("1. id\n2.name\n3.branch\n4. give all record in table");
        int key = sc.nextInt();
        switch (key) {
        case 1:
            System.out.println("on the bsis of id:\nenter id:");
            int idToSearch = sc.nextInt();
            String sql21 = "SELECT * FROM account_detail WHERE
Acc_ID ="+ idToSearch +";";

            printFetchedRecord(sql21);

            break;

        case 2:
            System.out.println("on basis of name:\nenter the name
You want to search");
            String nameToSearch = sc.next();

            String sql22 = "SELECT * FROM account_detail WHERE
Acc_holder_Name=\""+nameToSearch+"\";";

            printFetchedRecord(sql22);
            break;

        case 3:
            System.out.println("enter the branch");
            String branchToSearch = sc.next();

            String sql23 = "SELECT * FROM account_detail WHERE
Branch=\""+branchToSearch+"\";";

            printFetchedRecord(sql23);
            break;

        case 4:
            System.out.println("all record to be fetched ");
            String sql24 = "SELECT * FROM account_detail;";

            printFetchedRecord(sql24);
            break;

        default:
            System.out.println("entered wrong choice");
            break;
        }

    }


    //
=====================================================================
==================================

    private static void deleteARecord() {
        // TODO Auto-generated method stub
        System.out.println("enter the name of holder to be deleted");
        String nameDelete = sc.next();

        String sql3 = "DELETE FROM account_detail WHERE
Acc_holder_Name=\""+nameDelete+"\";";

        try {
            st = con.createStatement();
            st.executeUpdate(sql3);

            System.out.println("row deleted with name : "
+nameDelete);
            st.close();
            con.close();
        }
        catch (Exception e) {
            // TODO: handle exception
            e.printStackTrace();
        }

    }


    //
====================================================
=========================

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        try {

            Class.forName(JDBC_DRIVER);

            con =
DriverManager.getConnection(DB_URL,USER,PASS);

            System.out.println("enter ur chice to be performed in
jdbc");
            System.out.println("1. choice for insertion of data in
table");
            System.out.println("2. fetching of data in table");
            System.out.println("3. choice for deletion of row in table");

            int key = Integer.parseInt(sc.nextLine());

            switch (key) {
            case 1:
                insertRecord();
                break;
            case 2:

                fetchRecord();

                break;

            case 3:
                deleteARecord();
                break;

            default:

                System.out.println("entered choice is incorrect");

                break;
            }



        }
        catch (Exception e) {
            // TODO: handle exception
            e.printStackTrace();
            throw new RuntimeException("Something went wrong");

        }

    }

}
```

=======================================================

5.Read the details of 5 students(student_roll, student_name, admission_date) from command line or scanner and store them in a new table. Create a new field student_ID which is a combination of normalised(removing any special characters) admission_date and student_roll and add that column in the database as well. [ For example if a student’s roll number is 101 and admission_date is 2020-06-06, then student_ID will be 20200606101. ]

Sol.

PojoClass:-

```
package com.jdbc;

public class StudentPojoClass {
	
	public String getSname() {
		return sname;
	}

	public void setSname(String sname) {
		this.sname = sname;
	}

	public String getDate() {
		return date;
	}

	public void setDate(String date) {
		this.date = date;
	}

	public String getSid() {
		return sid;
	}

	public void setSid(String sid) {
        this.sid = sid;
    }

    public String getSrollno() {
        return srollno;
    }

    public void setSrollno(String srollno) {
        this.srollno = srollno;
    }

    String sname,date,sid,srollno;
}
```

StudentJdbc:-

```
package com.jdbc;

import java.sql.*;
import java.util.*;
public class StudentJdbc {
    public static void main(String[] args) {
        try {
            Scanner sc  = new Scanner(System.in);
            Class.forName("com.mysql.jdbc.Driver");

            String url="jdbc:mysql://localhost:3306/assignment";
            String username="root";
            String password="root";

            Connection con = DriverManager.getConnection(url,
username, password);
            String query ="ALTER TABLE student_details ADD
COLUMN student_id VARCHAR(25);";
            Statement stmt = null;
            stmt = con.createStatement();
            stmt.execute(query);

            String query1 ="insert into
student_details(student_rollno,student_name,admission_date,student_id)
values(?,?,?,?)";

            StudentPojoClass s1 = new StudentPojoClass();
            PreparedStatement pstmt
=con.prepareStatement(query1);

            for(int i=0;i<5;i++) {
                String srollno =sc.next();
                s1.setSrollno(srollno);
                String studname =sc.next();
                s1.setSname(studname);
                String date =sc.next();
                s1.setDate(date);
                String sid = date + srollno;
                s1.setSid(sid);


                pstmt.setString(1,s1.getSrollno());
                pstmt.setString(2,s1.getSname());
                pstmt.setString(3,s1.getDate());
                pstmt.setString(4,s1.getSid());

                pstmt.execute();
            }
            con.close();
            sc.close();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }
}
```

=================================================

Question :Write a regex for Capturing the following fields in groups:

**ClaveProdServ**

**TipoCambioP**

**ImpSaldoAnt**

**ImpPagado**

Sol

(single regex) =>

```
ClaveProdServ="(\d+)"[^>]+><[^>]+><[^>]+><[^>]+><[^>]+?TipoCambioP="
([\d.]+)"[^>]+><[^>]+?ImpSaldoAnt="([\d.]+)"[^>]ImpPagado="([\d].+)"[^>]+>
```

=================================================

Write a regex to capture the following fields in groups:

Invoice Number
Invoice Date
Gross amount

Text from HTML is given below the screenshot.

```
<tr
class=”griddata”[^>]*\s+<td[^>]+>[^<]+<\td><td[^>]+><a[^>]+>(\d+)</a></td><td[^>]+>[^>]+>[^>]
+>([^<]+)</td>[^>]+>([\d..]+)</td>
```