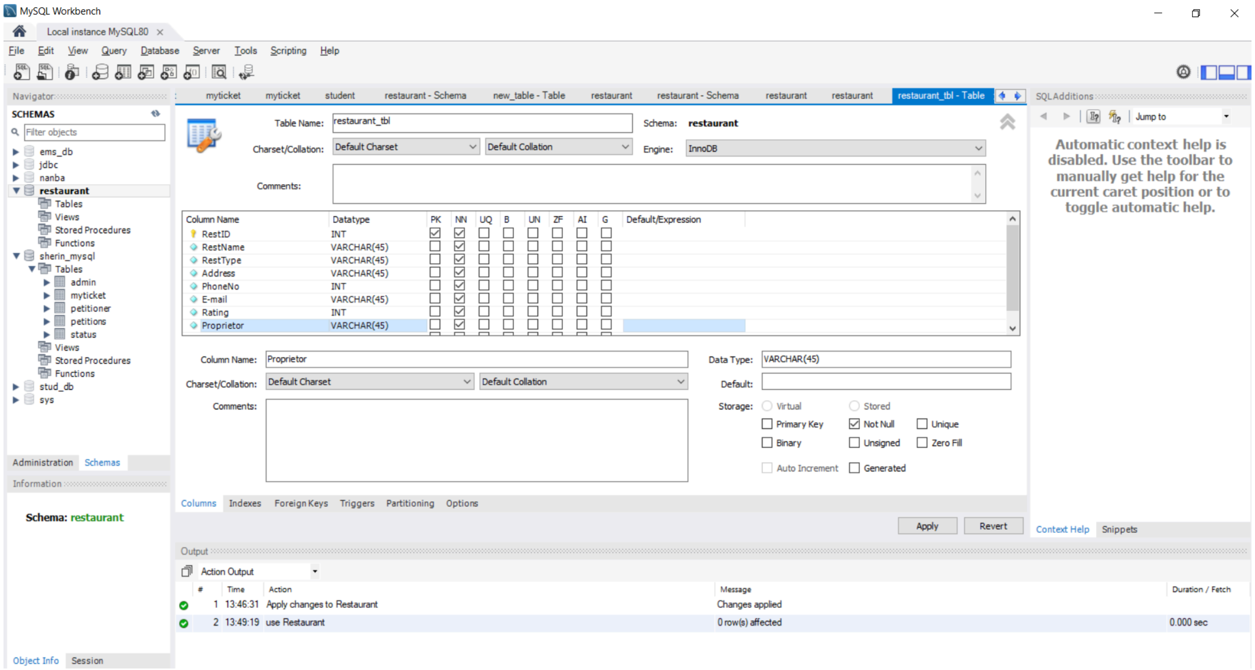
Q11) Restaurant Management System

Property	Type
RestID	Int
RestName	String
RestType	String
Address	String
PhoneNo	String
E-mail	String
Rating	Float
Proprietor	String

Create a Database, table and columns:



Schema: restaurant

Table Name: restaurant_tbl

Charset/Collation: Default Charset | Default Collation

Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
RestID	INT	☑	☑	□	□	□	□	□	□	
RestName	VARCHAR(45)	□	☑	□	□	□	□	□	□	
RestType	VARCHAR(45)	□	☑	□	□	□	□	□	□	
Address	VARCHAR(45)	□	☑	□	□	□	□	□	□	
PhoneNo	INT	□	☑	□	□	□	□	□	□	
E-mail	VARCHAR(45)	□	☑	□	□	□	□	□	□	
Rating	INT	□	☑	□	□	□	□	□	□	
Proprietor	VARCHAR(45)	□	☑	□	□	□	□	□	□	

Column Name: Proprietor

Data Type: VARCHAR(45)

Charset/Collation: Default Charset | Default Collation

Default:

Comments:

Storage: ○ Virtual ○ Stored

□ Primary Key ☑ Not Null □ Unique

□ Binary □ Unsigned □ Zero Fill

□ Auto Increment □ Generated

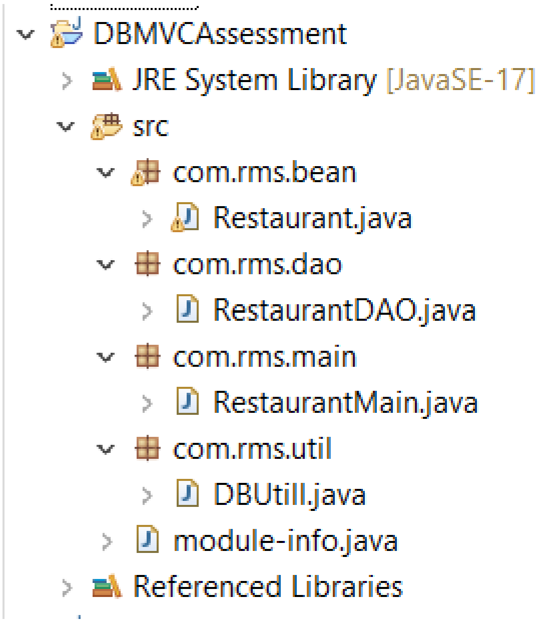
Output

Action Output

```
#  Time      Action                      Message             Duration / Fetch
1  13:46:31  Apply changes to Restaurant  Changes applied
2  13:49:19  use Restaurant               0 row(s) affected   0.000 sec
```

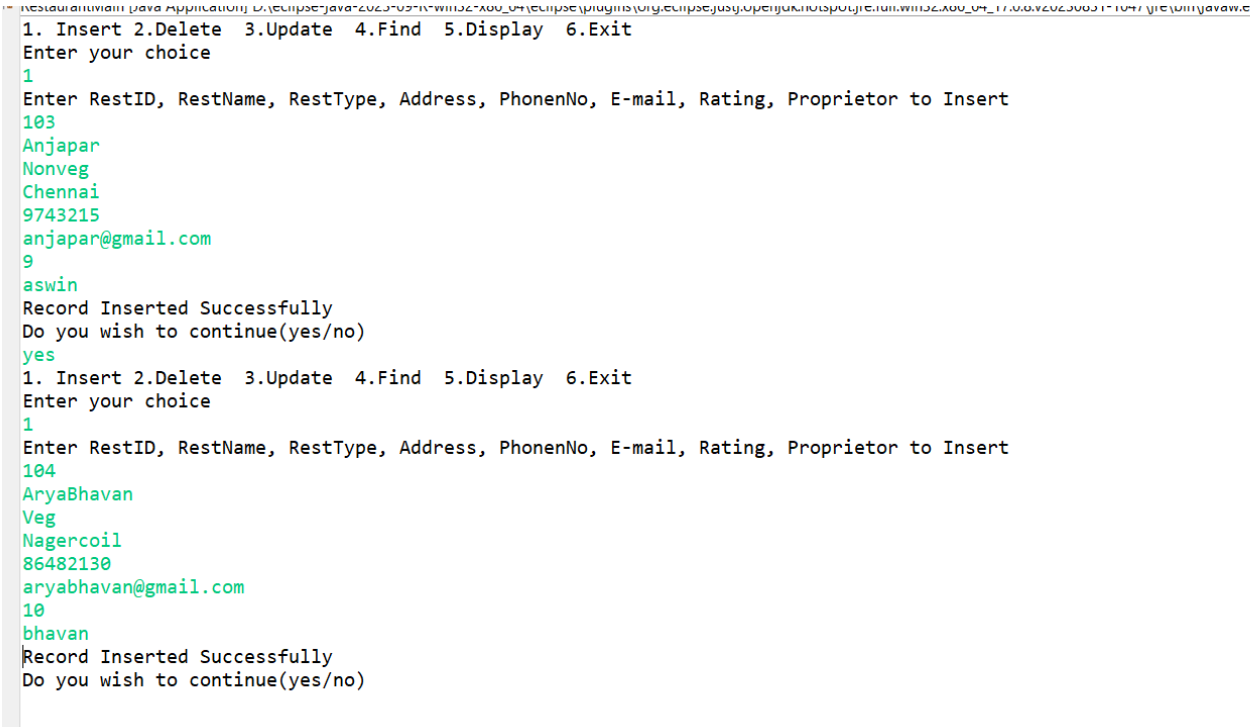
Schema: restaurant

Create a new project, package and class



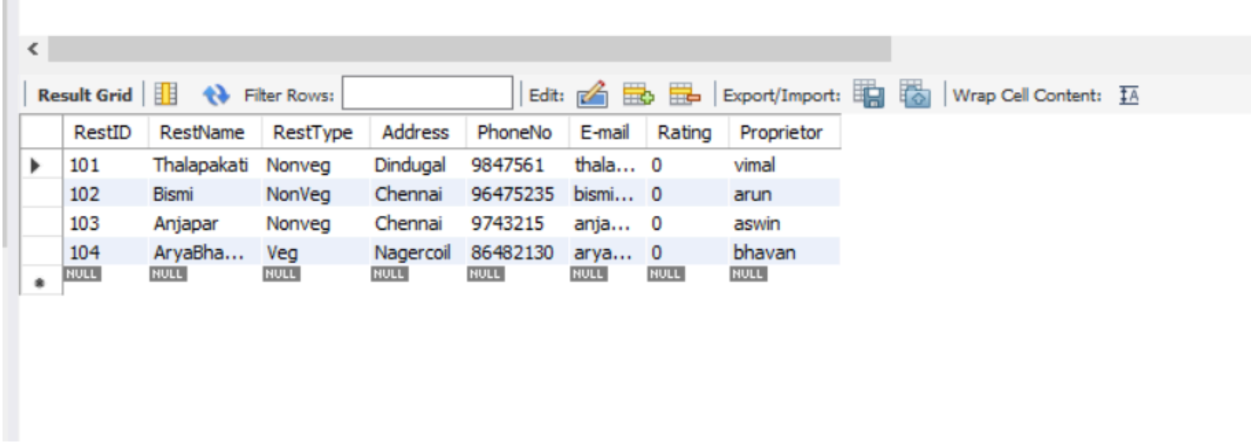
```
DBMVCAssessment
    JRE System Library [JavaSE-17]
    src
        com.rms.bean
            Restaurant.java
        com.rms.dao
            RestaurantDAO.java
        com.rms.main
            RestaurantMain.java
        com.rms.util
            DBUtill.java
        module-info.java
    Referenced Libraries
```

Insert the values in the columns



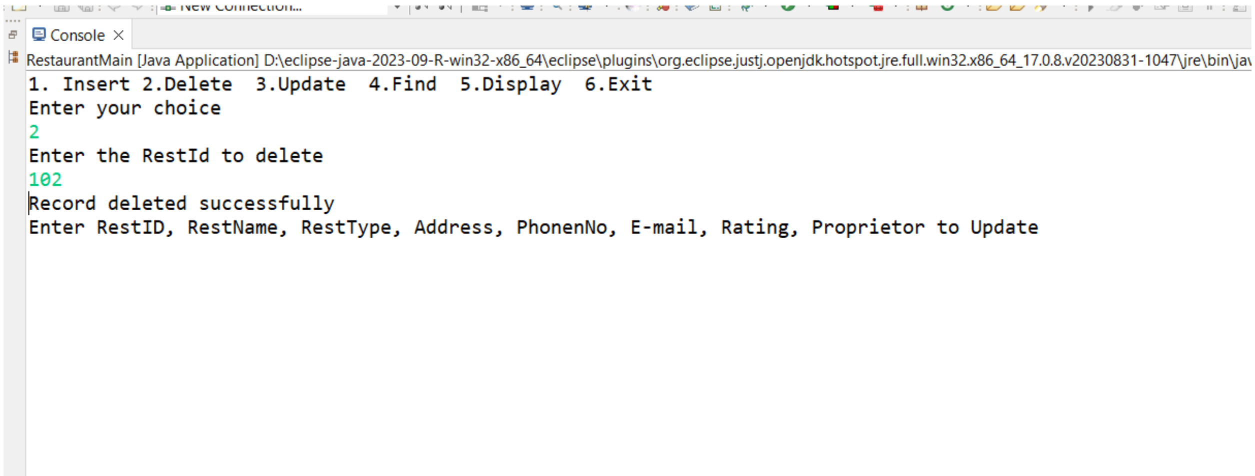
```
RestaurantMain [Java Application] D:\eclipse-java-2023-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.8.v20230831-1047\jre\bin\javaw.e
1. Insert 2.Delete  3.Update  4.Find  5.Display  6.Exit
Enter your choice
1
Enter RestID, RestName, RestType, Address, PhonenNo, E-mail, Rating, Proprietor to Insert
103
Anjapar
Nonveg
Chennai
9743215
anjapar@gmail.com
9
aswin
Record Inserted Successfully
Do you wish to continue(yes/no)
yes
1. Insert 2.Delete  3.Update  4.Find  5.Display  6.Exit
Enter your choice
1
Enter RestID, RestName, RestType, Address, PhonenNo, E-mail, Rating, Proprietor to Insert
104
AryaBhavan
Veg
Nagercoil
86482130
aryabhavan@gmail.com
10
bhavan
Record Inserted Successfully
Do you wish to continue(yes/no)
```

Display the values in the column



	RestID	RestName	RestType	Address	PhoneNo	E-mail	Rating	Proprietor
▶	101	Thalapakati	Nonveg	Dindugal	9847561	thala...	0	vimal
	102	Bismi	NonVeg	Chennai	96475235	bismi...	0	arun
	103	Anjapar	Nonveg	Chennai	9743215	anja...	0	aswin
	104	AryaBha...	Veg	Nagercoil	86482130	arya...	0	bhavan
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

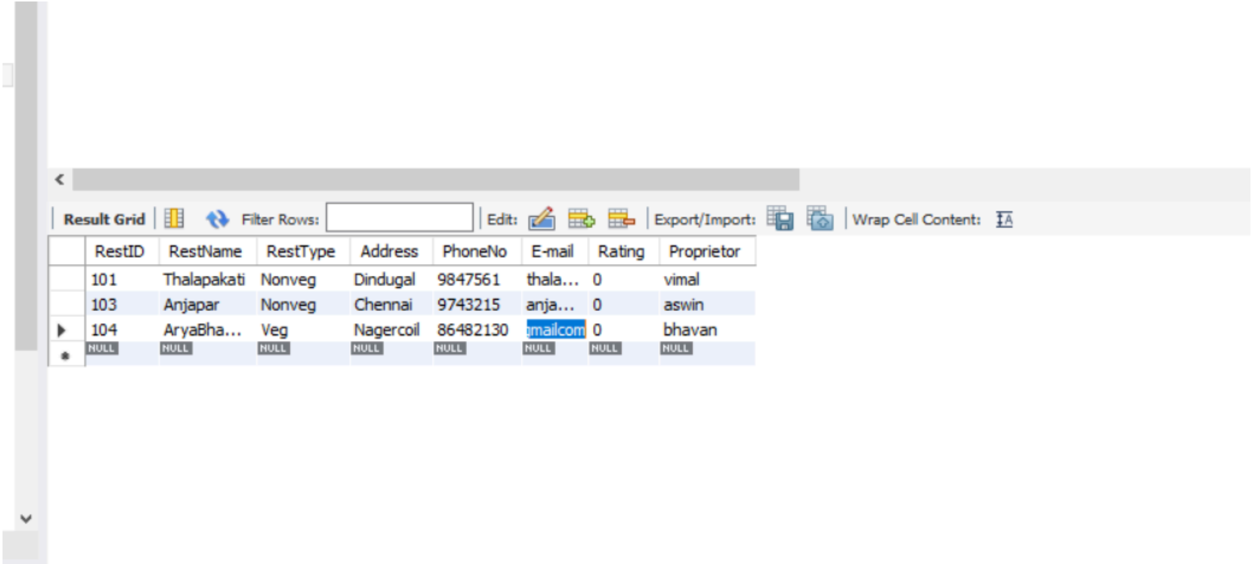
Delete the value in the column



RestaurantMain [Java Application] D:\eclipse-java-2023-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.8.v20230831-1047\jre\bin\jav

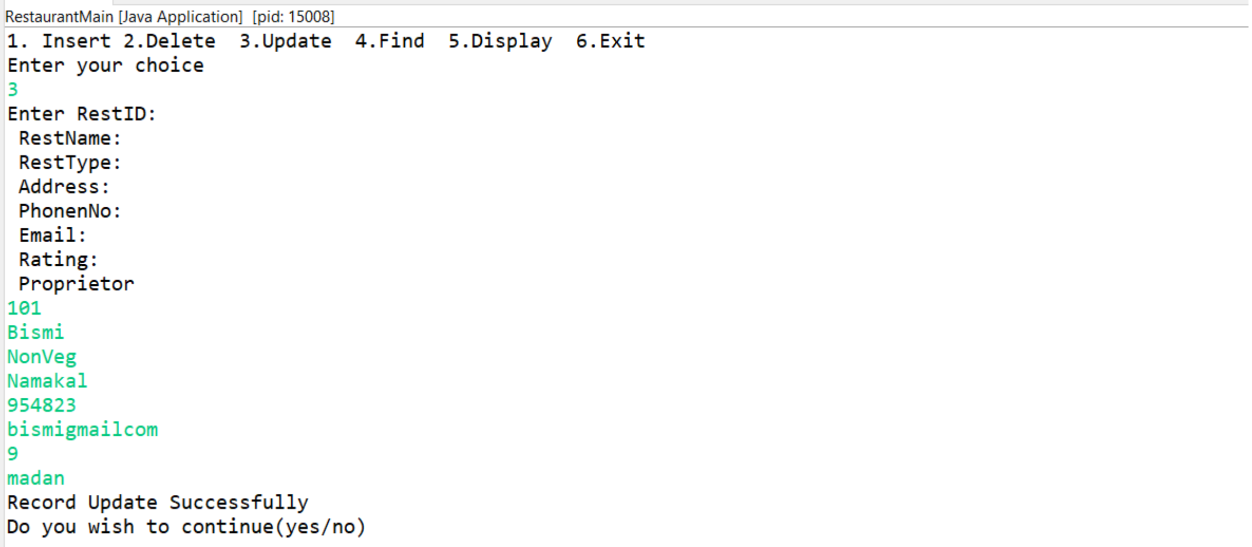
```
1. Insert 2.Delete  3.Update  4.Find  5.Display  6.Exit
Enter your choice
2
Enter the RestId to delete
102
Record deleted successfully
Enter RestID, RestName, RestType, Address, PhonenNo, E-mail, Rating, Proprietor to Update
```

Display the value in the column



	RestID	RestName	RestType	Address	PhoneNo	E-mail	Rating	Proprietor
	101	Thalapakati	Nonveg	Dindugal	9847561	thala...	0	vimal
	103	Anjapar	Nonveg	Chennai	9743215	anja...	0	aswin
▶	104	AryaBha...	Veg	Nagercoil	86482130	gmailcom	0	bhavan
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

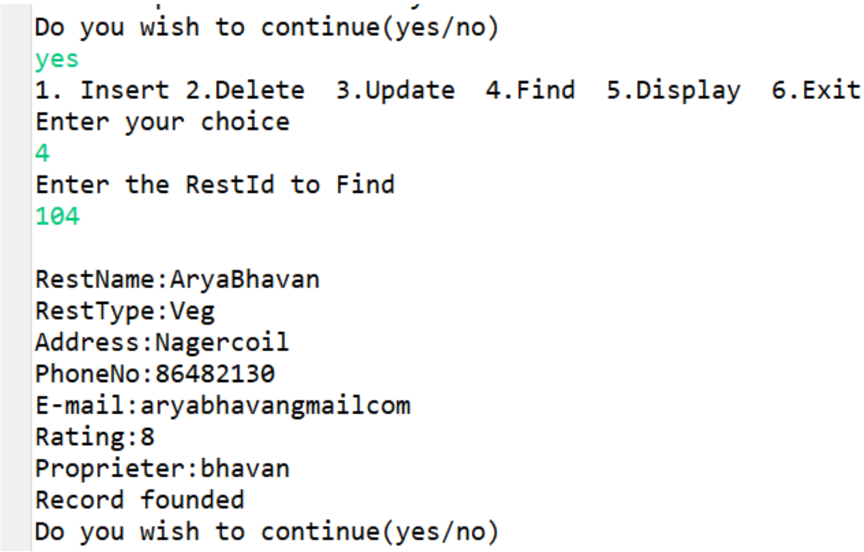
Update the value in column



RestaurantMain [Java Application] [pid: 15008]

```
1. Insert 2.Delete  3.Update  4.Find  5.Display  6.Exit
Enter your choice
3
Enter RestID:
 RestName:
 RestType:
 Address:
 PhonenNo:
 Email:
 Rating:
 Proprietor
101
Bismi
NonVeg
Namakal
954823
bismigmailcom
9
madan
Record Update Successfully
Do you wish to continue(yes/no)
```

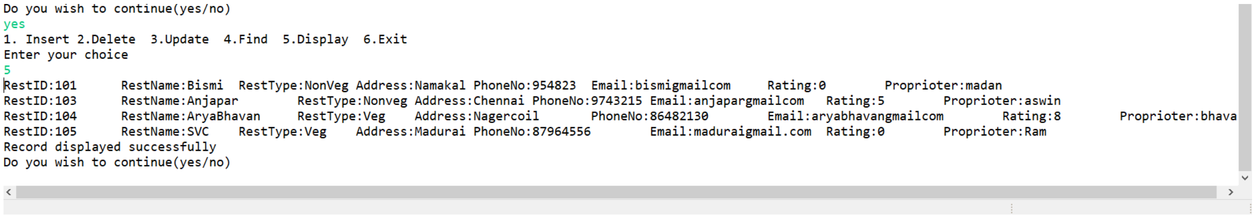
Find the value in the column



```
Do you wish to continue(yes/no)
yes
1. Insert 2.Delete  3.Update  4.Find  5.Display  6.Exit
Enter your choice
4
Enter the RestId to Find
104

RestName:AryaBhavan
RestType:Veg
Address:Nagercoil
PhoneNo:86482130
E-mail:aryabhavangmailcom
Rating:8
Proprieter:bhavan
Record founded
Do you wish to continue(yes/no)
```

Display all the values



```
Do you wish to continue(yes/no)
yes
1. Insert 2.Delete  3.Update  4.Find  5.Display  6.Exit
Enter your choice
5
RestID:101      RestName:Bismi  RestType:NonVeg Address:Namakal PhoneNo:954823  Email:bismigmailcom     Rating:0        Proprioter:madan
RestID:103      RestName:Anjapar        RestType:Nonveg Address:Chennai PhoneNo:9743215 Email:anjapargmailcom   Rating:5        Proprioter:aswin
RestID:104      RestName:AryaBhavan     RestType:Veg    Address:Nagercoil       PhoneNo:86482130        Email:aryabhavangmailcom        Rating:8        Proprioter:bhava
RestID:105      RestName:SVC    RestType:Veg    Address:Madurai PhoneNo:87964556        Email:maduraigmail.com  Rating:0        Proprioter:Ram
Record displayed successfully
Do you wish to continue(yes/no)
```

DAO code

```java
package com.rms.dao;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import com.rms.bean.restaurant_tbl;
import com.rms.util.DBUtill;
public class RestaurantDAO {
    public int insertRestaurant(restaurant_tbl rm) {
        int n = 0;
        try {
            Connection con = DBUtill.getDBConnection();
            String sql = "insert into restaurant_tbl values(?,?,?,?,?,?,?,?)";
            PreparedStatement ps = con.prepareStatement(sql);
            ps.setInt(1, rm.getRestId());
            ps.setString(2, rm.getRestName());
            ps.setString(3, rm.getRestType());
            ps.setString(4, rm.getAddress());
            ps.setInt(5, rm.getPhoneNo());
            ps.setString(6, rm.getEmail());
            ps.setInt(7, rm.getRating());
            ps.setString(8, rm.getProprietor());
            n = ps.executeUpdate();
        } catch (SQLException e) {
            System.out.println(e);
        }
        return n;
    }
    // delete the value
    public int deleteRestaurant(int RestID) {
        int n = 0;
        try {
            Connection con = DBUtill.getDBConnection();
            String sql = "delete from restaurant_tbl where RestID=?";
            PreparedStatement ps = con.prepareStatement(sql);
            ps.setInt(1, RestID);
            n = ps.executeUpdate();
        } catch (SQLException e) {
            System.err.println(e);
        }
        return n;
    }
//update the value
    public int updateRestaurant(restaurant_tbl rm) {
        int n = 0;
        try {
            Connection con = DBUtill.getDBConnection();
            String sql = "update restaurant_tbl set
RestName=?,RestType=?, Address=?, PhoneNo=?, Email=?, Rating=?,
Proprietor=? where RestID=?";
            PreparedStatement ps = con.prepareStatement(sql);
            ps.setString(1, rm.getRestName());
            ps.setString(2, rm.getRestType());
            ps.setString(3, rm.getAddress());
            ps.setInt(4, rm.getPhoneNo());
            ps.setString(5, rm.getEmail());
            ps.setInt(6, rm.getRating());
            ps.setString(7, rm.getProprietor());
            ps.setInt(8, rm.getRestId());
            n = ps.executeUpdate();
        } catch (SQLException e) {
            System.out.println(e);
        }
        return n;
    }
//find the value
    public int findRestaurant(int id) {
        int n = 0;
        try {
            Connection con = DBUtill.getDBConnection();// RETURN the
connection object
            String sql = "select*from restaurant_tbl where RestID=?";
            PreparedStatement ps = con.prepareStatement(sql);
            ps.setInt(1, id);
            ResultSet rs = ps.executeQuery();
            while (rs.next()) {
                System.out.println("\nRestName:" + rs.getString(2) +
"\nRestType:" + rs.getString(3) + "\nAddress:"
```

```
                        + rs.getString(4) + "\nPhoneNo:" +
rs.getInt(5) + "\nE-mail:" + rs.getString(6) + "\nRating:"
                        + rs.getInt(7) + "\nProprieter:" +
rs.getString(8));
                n = 1;
            }
        } catch (Exception e) {
            System.out.println(e);
        }
        return n;
    }
    // display
    public int Display() {
        int n = 0;
        try {
            Connection con = DBUtill.getDBConnection();
            String q = "select * from restaurant_tbl";
            PreparedStatement ps = con.prepareStatement(q);
            ResultSet rs = ps.executeQuery();
            while (rs.next())
                System.out.println("RestID:" + rs.getInt(1) +
"\tRestName:" + rs.getString(2) + "\tRestType:"
                        + rs.getString(3) + "\tAddress:" +
rs.getString(4) + "\tPhoneNo:" + rs.getInt(5) + "\tEmail:"
                        + rs.getString(6) + "\tRating:" + rs.getInt(7)
+ "\tProprioter:" + rs.getString(8));
            n = 1;
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return n;
    }
}
```

Main code

```
package com.rms.main;
import java.util.Scanner;
import com.rms.bean.restaurant_tbl;
import com.rms.dao.RestaurantDAO;
public class RestaurantMain {
        static Scanner sc = new Scanner(System.in);
        public static int displaymenu() {
                System.out.println("1. Insert 2.Delete  3.Update  4.Find  5.Display
6.Exit");
                System.out.println("Enter your choice");
                int choice = sc.nextInt();
                return choice;
        }
        public static restaurant_tbl insertRestaurant() {
                System.out.println("Enter RestID, RestName, RestType, Address,
PhonenNo, E-mail, Rating, Proprietor to Insert");
                return new restaurant_tbl(sc.nextInt(), sc.next(), sc.next(), sc.next(),
sc.nextInt(), sc.next(), sc.nextInt(),
                                sc.next());
        }
        public static int deleteRestaurant() {
                System.out.println("Enter the RestId to delete");
                return (sc.nextInt());
        }
        public static restaurant_tbl updateRestaurant() {
                System.out.println(
                                "Enter RestID:\n RestName:\n RestType:\n Address:\n
PhonenNo:\n Email:\n Rating:\n Proprietor");
                return new restaurant_tbl(sc.nextInt(), sc.next(), sc.next(), sc.next(),
sc.nextInt(), sc.next(), sc.nextInt(),
                                sc.next());
        }

        public static int findRestaurant() {
                System.out.println("Enter the RestId to Find");
                return (sc.nextInt());
        }
        public static void main(String[] args) {
                String msg = " ";
        int n;
        RestaurantDAO dao = new RestaurantDAO();
        do {
            switch (displaymenu()) {
            case 1:
                restaurant_tbl n1 = insertRestaurant();
                n = dao.insertRestaurant(n1);
                if (n == 1)
                    System.out.println("Record Inserted
Successfully");
                else
                    System.out.println("Record Insertion Failure");
                break;
            case 2:
                int RestID = deleteRestaurant();
                n = dao.deleteRestaurant(RestID);
                if (n == 1)
                    System.out.println("Record deleted
successfully");
                else
                    System.out.println("Record not found");

                break;
            case 3:{

                restaurant_tbl n2 = updateRestaurant();
                n = dao.updateRestaurant(n2);
                if (n == 1)
                    System.out.println("Record Update
Successfully");
                else
                    System.out.println("Record Updation Failure");
            }
                break;
            case 4:
                int id = findRestaurant();
                int n4 = dao.findRestaurant(id);
                if (n4 == 1)
                    System.out.println("Record founded");
                else
```

```
                    System.out.println("Record not found");
                break;
            case 5:
                n4 = dao.Display();
                if (n4 == 1)
                    System.out.println("Record displayed
successfully");
                else
                    System.out.println("Record not found");
                break;
            case 6:
                System.exit(0);

                break;
            default:
                System.out.println("There is no process for your
Input....!!!!!!");
            }
            System.out.println("Do you wish to continue(yes/no)");
            msg = sc.next();
        }
        while (msg.equals("yes"));
    }
}
```

Bean code

```
package com.rms.bean;
public class restaurant_tbl {
        private int RestId;
        private String RestName;
        private String RestType;
        private String Address;
        private int PhoneNo;
        private String Email;
        private int Rating;
            public restaurant_tbl(int restId, String restName, String restType,
String address, int phoneNo, String email,
                        int Rating, String proprietor) {
                  super();
                  RestId = restId;
                  RestName = restName;
                  RestType = restType;
                  Address = address;
                  PhoneNo = phoneNo;
                  Email = email;
                  Rating = Rating;
                  Proprietor = proprietor;
            }
            private String Proprietor;


        public restaurant_tbl() {
          super();
        }
            public int getRestId() {
                  return RestId;
            }
            public void setRestId(int restId) {
                  RestId = restId;
            }
            public String getRestName() {
                  return RestName;
            }
            public void setRestName(String restName) {
                  RestName = restName;
```

```
    }
    public String getRestType() {
        return RestType;
    }
    public void setRestType(String restType) {
        RestType = restType;
    }
    public String getAddress() {
        return Address;
    }
    public void setAddress(String address) {
        Address = address;
    }
    public int getPhoneNo() {
        return PhoneNo;
    }
    public void setPhoneNo(int phoneNo) {
        PhoneNo = phoneNo;
    }
    public String getEmail() {
        return Email;
    }
    public void setEmail(String email) {
        Email = email;
    }
    public int getRating() {
        return Rating;
    }
    public void setRating(int rating) {
        Rating = rating;
    }
    public String getProprietor() {
        return Proprietor;
    }
    public void setProprietor(String proprietor) {
        Proprietor= proprietor;
    }
}
```

Util code

```
package com.rms.util;

import java.sql.Connection;
import java.sql.DriverManager;

public class DBUtill {

    public static Connection getDBConnection()
    {
        Connection con =null;
        final String
URL="jdbc:mysql://localhost:3306/restaurant";
        final String User="root";
        final String pass="Password@12345";
        try {
            con =
DriverManager.getConnection(URL,User,pass);
    }
        catch(Exception e)
        {
            System.out.println(e);
        }
        return con;
}

}
```