

INTERVIEW DEFINITION

Write a program to demonstrate crud operation using JDBC & Scanner class.

Student [RollNo, Name, Contact No, City, Email ID, Standard]

Ex.

- Add
- Display
- Update
- Delete
- Exit

E.g.: On execution of the code following Menu must be displayed.

Option 1:

When user select option 1 then following must be displayed:

```
Enter RollNo.: 1
Enter Name: Aegis
Enter Contact No.: 1234567890
Enter City:Rajkot
Enter Email ID: harsh.savani@aegisiscteam.com
Enter Standard: 12
-----
1, Aegis, 1234567890, Rajkot, harsh.savani@aegisiscteam.com,
-----
Do you want to add more record(Y/N): Y
```

If user enter Y then allow user to enter the new record.

```
-----
1, Aegis, 1234567890, Rajkot, harsh.savani@aegisiscteam.com,
-----
```

Option 2:

When user select option 2 then following must be displayed:

Option 3:

When user select option 3 then following must be displayed:

```
-----Menu-----  
1. Add  
2. Display  
3. Update  
4. Delete  
5. Exit  
-----  
Enter the choice:3  
-----
```

Option 4:

When user select option 4 then following must be displayed:

Before deleting the record take the confirmation from the user.

```
-----  
1, Aegis, 1234567890, Rajkot, harsh.savani@aegisiscteam.com,  
2, Aegis, 1234567890, Rajkot, harsh.savani@aegisiscteam.com,  
-----  
Enter RollNo.: 1  
Are you sure want to Delete(Y/N):Y  
-----  
2, Aegis, 1234567890, Rajkot, harsh.savani@aegisiscteam.com,  
-----
```

Option 5:

```
-----Menu-----
1. Add
2. Display
3. Update
4. Delete
5. Exit
-----
Enter the choice:5
-----
Exited Successfully
```

- Conditions

- Program should not be terminated till the user Exit it.
- After each selection Menu must be displayed asking for user input.
- After execution of each option confirmation message must be displayed.
- Clicking on New option take the inputs from the user with proper message and the entered record at the end.
- Clicking on Modify option ask user for inputting id for updating and after that record must be updated with new entered value and display the records.
- Remove option must ask to user for ID to delete and again ask for confirmation (Y/N) before deletion and display proper message after deletion.
- Finding student by roll no and name option, menu must be displayed to user for asking that "Want to search by Roll No then 1 and 2 for searching by Name".
- Search the particular student according to Roll No or Name entered for point 7 and display the record.
- Find multiple student by City and Standard option, menu must be displayed to user for asking that "Want to search by City then 1 and 2 for searching by Standard".
- Search the particular student according to City or Standard entered for point 9 and display the record.
- Display all the records when user enter Display All option.
- Program must exit only when user enter 7.

“Good luck for the day and may the best results come.”

CustomConnection:

```

package Jdbcpack;

import java.io.FileReader;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

public class CustomConnection {
    public static Connection getCustConnection()
    {

        Connection connection=null;
        try {
            FileReader freader = new FileReader("database.properties");
            Properties properties=new Properties();
            properties.load(freader);

            //load driver--mysql
            Class.forName(properties.getProperty("driver"));
            //connect to database
            connection=DriverManager.getConnection(properties.getProperty("url"),
            properties.getProperty("username"),properties.getProperty("password"));
            return connection;
        }catch(ClassNotFoundException e)
        {
            System.out.println(e.getMessage()+" "+e.getClass());
            return null;
        }
        catch(SQLException ex)
        {
            System.out.println(ex.getMessage());
            return null;
        }catch(Exception e)
        {
            System.out.println(e.getMessage());
            return null;
        }

    }
}

```

Student:

```

package Jdbcpack.model;

```

```
public class Student {  
    private int rno;  
    private String name;  
    private String contactno;  
    private String city;  
    private String emailId;  
    private String standard;
```

```
    public Student() {  
  
    }
```

```
    public Student(int rno, String name, String contactno, String city, String emailId,  
String standard) {  
        super();  
        this.rno = rno;  
        this.name = name;  
        this.contactno = contactno;  
        this.city = city;  
        this.emailId = emailId;  
        this.standard = standard;  
    }
```

```
    public int getRno() {  
        return rno;  
    }
```

```
    public void setRno(int rno) {  
        this.rno = rno;  
    }
```

```
    public String getName() {  
        return name;  
    }
```

```
    public void setName(String name) {  
        this.name = name;  
    }
```

```
    public String getContactno() {
```

```

        return contactno;
    }

    public void setContactno(String contactno) {
        this.contactno = contactno;
    }

    public String getCity() {
        return city;
    }

    public void setCity(String city) {
        this.city = city;
    }

    public String getEmailId() {
        return emailId;
    }

    public void setEmailId(String emailId) {
        this.emailId = emailId;
    }

    public String getStandard() {
        return standard;
    }

    public void setStandard(String standard) {
        this.standard = standard;
    }

    @Override
    public String toString() {
        return "Student [rno=" + rno + ", name=" + name + ", contactno=" +
        contactno + ", city=" + city + ", emailId="
            + emailId + ", standard=" + standard + "]";
    }

```

```
}
```

StudentInterface:

```
package Jdbcpack.service;

import java.sql.SQLException;

import Jdbcpack.model.Student;

public interface StudentInterface {
    void addStudent(Student student) throws SQLException;
    int updateStudent(Student student , int rno,String property) throws
SQLException;
    int deleteStudent(int rno2) throws SQLException;
    Student findStudentByrno(int rno) throws SQLException;

    void displayStudentDetails() throws SQLException;
    public String validateEmailId(String emailId);
    Student findStudentBycity(String city) throws SQLException;
    Student findStudentByName(String name) throws SQLException;
    Student findStudentBystandard(String standard) throws SQLException;
}
```

StudentInterfaceimpl

```
package Jdbcpack.service;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;
import java.util.regex.Pattern;

import Jdbcpack.CustomConnection;
import Jdbcpack.model.Student;

public class StudentInterfaceimpl implements StudentInterface {

    Connection connection=null;
    PreparedStatement pstatement=null;
    private Scanner scanner;
    public StudentInterfaceimpl()
    {
```

```

        connection=CustomConnection.getCustConnection();
    }
    @Override
    public void addStudent(Student student) throws SQLException {
        // TODO Auto-generated method stub

        pstatement=connection.prepareStatement("insert into Student
values(?,?,?,?);");
        pstatement.setInt(1, student.getRno());
        pstatement.setString(2, student.getName());
        pstatement.setString(3, student.getContactno());
        pstatement.setString(4, student.getCity());
        pstatement.setString(5, student.getEmailId());
        pstatement.setNString(6, student.getStandard());
        int res=pstatement.executeUpdate();
        if(res==1)
        {
            System.out.println("inserted successfully");
        }
    }

    public String validateEmailId(String emailId) {
        if(emailId==null || emailId.isEmpty()) {
            System.out.println("invalid");
        }
        String regex = "^(.+)?@(.+)$";
        Pattern pattern=Pattern.compile(regex);
        if(pattern.matcher(emailId).matches()){
            return "Valid";
        }else {
            return "Invalid";
        }
    }
    @Override
    public int updateStudent(Student student, int rno,String property) throws
SQLException {
        // TODO Auto-generated method stub
        Student student1=findStudentByrno(rno);
        if(property.equals("Name"))
            student1.setName(student.getName());
        if(property.equals("contactno"))
            student1.setContactno(student.getContactno());
        if(property.equals("city"))
            student1.setCity(student.getCity());
        if(property.equals("emailid"))

```



```

        student1.setEmailId(student.getEmailId());
        if(property.equals("standard"))
            student1.setStandard(student.getStandard());
        pstatement=connection.prepareStatement("update student set
name=?,contactno=?,city=?,emailId=?,standard=? where rno=? ");
        pstatement.setString(1,student1.getName());
        pstatement.setString(2, student1.getContactno());
        pstatement.setString(3, student1.getCity());
        pstatement.setString(4, student1.getEmailId());
        pstatement.setString(5, student1.getStandard());
        pstatement.setInt(6, student1.getRno());
        int resultSet=pstatement.executeUpdate();
        return resultSet;
    }

    @Override
    public int deleteStudent(int rno2) throws SQLException {
        pstatement=connection.prepareStatement("delete from student
where rno=?");
        pstatement.setInt(1, rno2);
        boolean resultSet=pstatement.execute();
        return rno2;
    }

    @Override
    public Student findStudentByrno(int rno) throws SQLException {
        // TODO Auto-generated method stub
        pstatement=connection.prepareStatement("select *from student
where rno=?");
        pstatement.setInt(1,rno);
        ResultSet rs=pstatement.executeQuery();
        Student student=null;
        while(rs.next()) {
            student=new
Student(rs.getInt(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs
.getString(6));
            System.out.println(student);
        }
        return student;
    }

    @Override
    public Student findStudentByName(String name) throws SQLException {

```

```

        // TODO Auto-generated method stub
        pstatement=connection.prepareStatement("select *from student
where name=?");
        pstatement.setString(1,name);
        ResultSet rs=pstatement.executeQuery();
        Student student=null;
        while(rs.next()) {
            student=new
Student(rs.getInt(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs
.getString(6));
            System.out.println(student);
        }
        return student;
    }
    @Override
    public Student findStudentBycity(String city) throws SQLException {
        // TODO Auto-generated method stub
        pstatement=connection.prepareStatement("select *from student
where city=?");
        pstatement.setString(1,city);
        ResultSet rs=pstatement.executeQuery();
        Student student=null;
        while(rs.next()) {
            student=new
Student(rs.getInt(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs
.getString(6));
            System.out.println(student);
        }
        return student;
    }
    @Override
    public Student findStudentBystandard(String standard) throws
SQLException {
        // TODO Auto-generated method stub
        pstatement=connection.prepareStatement("select *from student
where standard=?");
        pstatement.setString(1,standard);
        ResultSet rs=pstatement.executeQuery();
        Student student=null;
        while(rs.next()) {
            student=new
Student(rs.getInt(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs
.getString(6));
            System.out.println(student);
        }
        return student;
    }

```

```

    }

    @Override
    public void displayStudentDetails() throws SQLException {
        // TODO Auto-generated method stub
        pstatement=connection.prepareStatement("select *from student");
        ResultSet resultSet = pstatement.executeQuery();
        while(resultSet.next())//next() --brings the cursor to the first record
            //returns true
        {
            int rno=resultSet.getInt("rno");
            String name=resultSet.getString("name");
            String contactno=resultSet.getString("contactno");
            String city=resultSet.getString("city");
            String emailId=resultSet.getString("emailId");
            String standard=resultSet.getString("standard");
            Student student = new Student(rno,name,contactno,city,emailId,standard);
            System.out.println(student);
        }
    }
}

```

PreparedStatement:

```

package preparedpack;
import java.sql.*;
import java.util.Scanner;

public class PreparedStatementEx {

    public static void main(String[] args) throws ClassNotFoundException,
    SQLException {
        // TODO Auto-generated method stub
        Scanner scanner =new Scanner(System.in);
        Class.forName("com.mysql.jdbc.Driver");
        Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/class","root","A
poorva@123");
        PreparedStatement stmt =con.prepareStatement("insert into learners
values(?,?,?,?)");
        stmt.setInt(1,scanner.nextInt());
        stmt.setString(2,scanner.next());
        stmt.setString(3,scanner.next());
        stmt.setString(4,scanner.next());
    }
}

```

```

        stmt.setString(5,scanner.next());
        stmt.setString(6,scanner.next());
        int res=stmt.executeUpdate();
        if(res==1)
        {
            System.out.println("Updated.....");
        }
        PreparedStatement stmt1 =con.prepareStatement("update learners set
name=?,contactNo=?,city=?,emailId=?,standard=? where eno=?");
        stmt1.setString(1,"harish kumar");
        stmt1.setString(2,"9108677232");
        stmt1.setString(3,"gul");
        stmt1.setString(4,"Apoorva@gmail.com");
        stmt1.setString(5, "puc");

        int res1=stmt1.executeUpdate();//2
        if(res1>=1)
        {
            System.out.println("Updated.....");
        }
    }
}

```

Database.properties:

```

driver=com.mysql.jdbc.Driver
url=jdbc:mysql://localhost:3306/class
username=root
password=Apoorva@123

```

TestJDBC:

```

package Jdbcpack;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;
import Jdbcpack.model.Student;
import Jdbcpack.service.StudentInterface;
import Jdbcpack.service.StudentInterfaceimpl;

public class TestJDBC {

    public static void main(String[] args) throws SQLException {

```

```

// TODO Auto-generated method stub
StudentInterface sinterface=new StudentInterfaceimpl();
    char ch=' ';
do {
    System.out.println("-----
Menu-----");
    System.out.println("1---add \n 2---display \n 3---update \n 4---
delete \n 5---find student by rno or name \n 6---find student city or name \n 7---
exit");

    System.out.println("-----");
    System.out.print("enter the option:");
    Scanner scanner = new Scanner(System.in);
    int option =scanner.nextInt();

    System.out.println("-----");
    switch(option)
    {
    case 1: System.out.println("enter the student details
rno,name,contactno,city,emailId,standard");
        int rno=scanner.nextInt();
        String name=scanner.next();
        String contactno=scanner.next();
        String city=scanner.next();
        String emailId=scanner.next();
        String pattern = null;
        do {
            System.out.println("emailId" + validateEmailId(emailId));
        }while(emailId==pattern);
        String standard=scanner.next();
        Student student = new
Student(rno,name,contactno,city,emailId,standard);
        try {
            sinterface.addStudent(student);
        } catch (SQLException e) {
            System.out.println("adding
student -->" +e.getMessage());
        }
        break;

    case 2:try {
        sinterface.displayStudentDetails();
    } catch (SQLException e) {
        System.out.println("adding student -->" +e.getMessage());
    }
        break;

```

```

case 3:
modify");
    System.out.println("Enter the student RollNo you want to
    int rno1=scanner.nextInt();
    System.out.println("Enter the property you want to change");
    String property=scanner.next();
    Student mStudent=new Student();
    if(property.equals("Name")) {
        System.out.println("Enter the name");
        mStudent.setName(scanner.next());
    }
    if(property.equals("contactno")) {
        System.out.println("Enter the Contact number");
        mStudent.setContactno(scanner.next());
    }
    if(property.equals("city")) {
        System.out.println("Enter the City");
        mStudent.setCity(scanner.next());
    }
    if(property.equals("emailId")) {
        System.out.println("Enter the EmailId");
        mStudent.setEmailId(scanner.next());
    }

    if(property.equals("standard")) {
        System.out.println("Enter the Standard");
        mStudent.setStandard(scanner.next());
    }
    try {
        int res=sinterface.updateStudent(mStudent, rno1, property);
        if(res==1) {
            System.out.println("Updated Sucessfully");
            sinterface.displayStudentDetails();
        }
    }
    catch(SQLException e){
        System.out.println("updating student--->" +e.getMessage());
    }
    break;

```

```

case 4:
System.out.println("Enter the RollNo you want to delete");

```

```

int rno2=scanner.nextInt();
Student dStudent=new Student();
System.out.println("Are you sure...you want to delete(y/n)");
char ch1=scanner.next().charAt(0);
if(ch1=='y'||ch1=='Y') {
    try {
        sinterface.deleteStudent(rno2);
        System.out.println("Delete Sucessfully");
    } catch(SQLException e){
        System.out.println("Deleting student--->" +e.getMessage());
    }
}
break;
case 5:
System.out.println("Enter your choice for finding student data");
System.out.println("1.find by rno \n2.find by name");
System.out.println("enter your choice:");
int value=scanner.nextInt();
switch(value)
{
    case 1:
        System.out.println("enter the rno");
        rno=scanner.nextInt();
        sinterface.findStudentByrno(rno);
        break;
    case 2:
        System.out.println("enter the name of the student");
        name=scanner.next();
        sinterface.findStudentByName(name);
        break;
}
break;
case 6:
System.out.println("enter your choice for finding student :");
System.out.println("1.find by city \n 2.find by standard");
System.out.println("enter your choice:");
int value1=scanner.nextInt();
switch(value1) {
    case 1:
        System.out.println("enter the city :");
        city=scanner.next();
        sinterface.findStudentBycity(city);
        break;
    case 2:
        System.out.println("enter the standard.");
        standard=scanner.next();

```

```

        sinterface.findStudentBystandard(standard);
        break;

    }
    break;
case 7:
    System.out.println("Exited succesfully");
    System.exit(0);
    break;
    }
    System.out.println("do you wish to continue say yes");
    ch=scanner.next().charAt(0);
    }while(ch=='y' || ch=='Y');
}

private static String validateEmailId(String emailId) {
    // TODO Auto-generated method stub
    return null;
}

}

```