10. Write a java program that connects to a database using JDBC and demonstrate CRUD operations.

In terminal check for mysql by providing the command: mysql –version
if mysql is not installed, follow below procedure
sudo apt install mysql-server
sudo systemctl start mysql
sudo mysql_secure_installation
sudo mysql -u root –p or sudo mysql

Run this SQL in MySQL:

```
CREATE DATABASE demo;

USE demo;

CREATE TABLE students (
   id INT PRIMARY KEY,
   name VARCHAR(50),
   age INT
);
```

Download MySQL JDBC Connector

Download:
https://dev.mysql.com/downloads/connector/j/
Extract the `.jar`.
We will add this JAR to Eclipse.

Setup Eclipse Project

Step 1: Open Eclipse → File → New → Java Project

Name: JDBC_CRUD_Demo

Step 2: Add MySQL Connector JAR

Right-click project →
Build Path → Configure Build Path → Libraries → Add External JARs →
Select:
mysql-connector-j-8.x.x.jar

Click Apply and Close.

Create a CRUD Java Program

Create a new class:
src → new → Class → JDBCExample

Copy the complete CRUD code:

JDBC Program (CRUD Operations)

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

public class JDBCCRUDDemo {

   // Database credentials
   static final String URL = "jdbc:mysql://localhost:3306/demo";
```

```
static final String USER = "root";      // your MySQL username
static final String PASS = "password";  // your MySQL password

public static void main(String[] args) {
   Scanner sc = new Scanner(System.in);

   while (true) {
      System.out.println("\n-------- JDBC CRUD MENU --------");
      System.out.println("1. INSERT (Create)");
      System.out.println("2. SELECT (Read)");
      System.out.println("3. UPDATE");
      System.out.println("4. DELETE");
      System.out.println("5. EXIT");
      System.out.print("Choose an option: ");
      int choice = sc.nextInt();

      switch (choice) {
         case 1: insertRecord(); break;
         case 2: readRecords(); break;
         case 3: updateRecord(); break;
         case 4: deleteRecord(); break;
         case 5:
            System.out.println("Exiting...");
            System.exit(0);
         default:
            System.out.println("Invalid choice!");
      }
   }
}

// CREATE operation
public static void insertRecord() {
   try (Connection conn = DriverManager.getConnection(URL, USER, PASS)) {

      String sql = "INSERT INTO students (id, name, age) VALUES (?, ?, ?)";
      PreparedStatement pst = conn.prepareStatement(sql);

      Scanner sc = new Scanner(System.in);
      System.out.print("Enter ID: ");
      int id = sc.nextInt();
      sc.nextLine();

      System.out.print("Enter Name: ");
      String name = sc.nextLine();

      System.out.print("Enter Age: ");
      int age = sc.nextInt();

      pst.setInt(1, id);
      pst.setString(2, name);
      pst.setInt(3, age);

      int rows = pst.executeUpdate();
      System.out.println(rows + " record inserted successfully!");

   } catch (Exception e) {
      e.printStackTrace();
   }
}
```

```
// READ operation
public static void readRecords() {
    try (Connection conn = DriverManager.getConnection(URL, USER, PASS)) {

        String sql = "SELECT * FROM students";
        Statement st = conn.createStatement();
        ResultSet rs = st.executeQuery(sql);

        System.out.println("\n--- STUDENT RECORDS ---");
        while (rs.next()) {
            System.out.println(
                "ID: " + rs.getInt("id") +
                ", Name: " + rs.getString("name") +
                ", Age: " + rs.getInt("age")
            );
        }

    } catch (Exception e) {
        e.printStackTrace();
    }
}

// UPDATE operation
public static void updateRecord() {
    try (Connection conn = DriverManager.getConnection(URL, USER, PASS)) {

        String sql = "UPDATE students SET name=?, age=? WHERE id=?";
        PreparedStatement pst = conn.prepareStatement(sql);

        Scanner sc = new Scanner(System.in);
        System.out.print("Enter ID to update: ");
        int id = sc.nextInt();
        sc.nextLine();

        System.out.print("Enter New Name: ");
        String name = sc.nextLine();

        System.out.print("Enter New Age: ");
        int age = sc.nextInt();

        pst.setString(1, name);
        pst.setInt(2, age);
        pst.setInt(3, id);

        int rows = pst.executeUpdate();
        System.out.println(rows + " record updated successfully!");

    } catch (Exception e) {
        e.printStackTrace();
    }
}

// DELETE operation
public static void deleteRecord() {
    try (Connection conn = DriverManager.getConnection(URL, USER, PASS)) {

        String sql = "DELETE FROM students WHERE id=?";
        PreparedStatement pst = conn.prepareStatement(sql);

        Scanner sc = new Scanner(System.in);
```

```
            System.out.print("Enter ID to delete: ");
            int id = sc.nextInt();

            pst.setInt(1, id);

            int rows = pst.executeUpdate();
            System.out.println(rows + " record deleted successfully!");

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Run the Program in Eclipse

Right-click the file → Run As → Java Application

11. Write a java servlet program that takes name and age from an HTML page. If the age is less than 18, it should send a page with "Hello _____(name), you are not authorized to visit the site" message, where "_____" in the message should be replaced with the entered name. Otherwise it should send "Welcome to this site" message.

Create Dynamic Web Project

File → New → Dynamic Web Project

Project Name: CheckAge

Target runtime: Apache Tomcat v9.0

Dynamic Web Module Version: 3.1

Finish.

This creates the standard structure:

```
WebContent/
   index.html
   WEB-INF/
      web.xml
src/
   ...
```

Create the HTML Form

Right-click WebContent → New → HTML File
File name: index.html

HTML Form (name: index.html)

index.html:

```
<html>
<head>
<title>VoterApp</title>
</head>
<body>
<form action= "http://localhost:8080/CheckAge/check" method="get">
<fieldset style="width:20%; background-color:#80ffcc">
<table>
<tr><td>Name</td><td><input type="text" name="name"></td></tr>
```

```
<tr><td>Age</td><td><input type="text" name="age"></td></tr>
<tr><td></td>
<td><input type = "submit" value="Check Eligibility"></td></tr>
</table>
</fieldset>
</form>
</body>
</html>
```

Create the Servlet
Right-click src → New → Servlet
Servlet Name: VoterSrv
Replace auto-generated code with:

Java Servlet Program

```
//VoterSrv.java:
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class VoterSrv extends HttpServlet
{
public void service(HttpServletRequest req, HttpServletResponse res) throws
IOException,ServletException
{
//set response content type
res.setContentType("text/html");
//get printWrite obj
PrintWriter pw = res.getWriter();
//read form data from page as request parameter
String name = req.getParameter("name");
int age = Integer.parseInt(req.getParameter("age"));
if (age>=18)
{
pw.println("<font color='green' size='4'>Welcome "+name+" to this site</font>");
}
else
pw.println("<font color='red' size='4'>Hello "+name+", you are not authorized to
visit the site</font>");
//add hyperlink to dynamic page
pw.println("<br><br><a href= 'index.html'>back</a>");
//close the stream
pw.close();
}
}
```

Map the Servlet in web.xml

(If Eclipse already created annotation mapping, skip this step.)

Open:
WebContent → WEB-INF → web.xml

Add inside <web-app>:

web.xml:

```
<web-app>
<servlet>
<servlet-name>abc</servlet-name>
<servlet-class>VoterSrv</servlet-class>
```

</servlet>
<servlet-mapping>
<servlet-name>abc</servlet-name>
<url-pattern>/check</url-pattern>
</servlet-mapping>
</web-app>

Configure Tomcat in Eclipse

Bottom panel → Servers tab
(If not visible → Window → Show View → Servers)

Click: No servers → Create a new server

Select:
Apache → Tomcat v9.0 Server

Run the Project on Tomcat

Right-click project → Run As → Run on Server

Choose Tomcat v9.0 → Finish.

Your project runs at:
http://localhost:8080/CheckAge/

12. Write a java program using JSP that takes a name as input and has a submit button, on clicking submit button it shows a hello <name> page where name is taken from the request. It shows the start time at the right top corner of the page and provides a logout button. On clicking this button, it should show a logout page with Thank You <name > message with the duration of usage (hint: Use session to store name and time).

Configure Tomcat in Eclipse

1. Window → Show View → **Servers**
2. "Create new server"
3. Select **Apache Tomcat v9.0**
4. Set directory: `/usr/share/tomcat9`
5. Finish

Create Dynamic Web Project

1. File → New → **Dynamic Web Project**
2. Project Name: **JSPSessionDemo**
3. Target Runtime: **Apache Tomcat v9.0**
4. Module version: **3.1**
5. Finish

Create JSP Files

You will create **3 JSP pages**:

Right-click **WebContent → New → JSP File**

Session1.jsp

```
<%@ page language="java" %>
<html>
<body>
<h2>Enter Your Name</h2>

<form action="Session2.jsp" method="get">
   Name: <input type="text" name="uname" required>
   <br><br>
   <input type="submit" value="Submit">
</form>

</body>
</html>
```

Session2.jsp

```
<%@ page language="java" import="java.util.*" %>

<%
   String name = request.getParameter("uname");

   // Save name and start time into the session
   if (name != null) {
      session.setAttribute("user", name);
      session.setAttribute("startTime", new Date().getTime());
   }

   name = (String) session.getAttribute("user");
   long start = (Long) session.getAttribute("startTime");
```

```
%>

<html>
<body>

<p align="right">
   Start Time: <%= new Date(start).toString() %>
</p>

<h2>Hello <%= name %>!</h2>

<form action="Logout.jsp" method="get">
   <input type="submit" value="Logout">
</form>

</body>
</html>
```

Logout.jsp

```
<%@ page language="java" import="java.util.*" %>

<%
   String name = (String) session.getAttribute("user");
   long start = (Long) session.getAttribute("startTime");

   long end = new Date().getTime();      // current time
   long duration = end - start;          // duration in milliseconds

   long seconds = duration / 1000;
   long minutes = duration / (1000 * 60);
   long hours = duration / (1000 * 60 * 60);
```

```
   session.invalidate();                // logout
%>


<html>
<body>


<h2>Thank You <%= name %>!</h2>


Session Duration:
<br>
<%= hours %> hours,
<%= minutes %> minutes,
<%= seconds %> seconds


</body>
</html>
```

Run the Application

1. Right-click project → **Run As → Run on Server**
2. Select **Tomcat v9.0**
3. Open in browser:
4. http://localhost:8080/JSPSessionDemo/index.jsp