**PRACTICAL SUBMISSION RECORD- A.Y. 2024-25**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class: FYMCA Div: A**  **Semester: II** | **Course Code: MCA01554**  **Course Name: Java Programming Laboratory** | | **Batch: F2** |
| **Name: Pranav Raju Malwatkar** | | **Roll No: 51037** | |
| **CO No: CO515.4** | | **Assignment No: 10** | |

# Title:

**Write JDBC backend class, which establishes a connection with the database having a table “STUDENT” with structure – roll no, name, class, DOB. Using the above backend class WAP to accept roll\_no from command line and display the details with proper error messages. Display all the records from table with proper formatting.**

# Code:

**SQL:**

Create Database student

CREATE TABLE IF NOT EXISTS students (

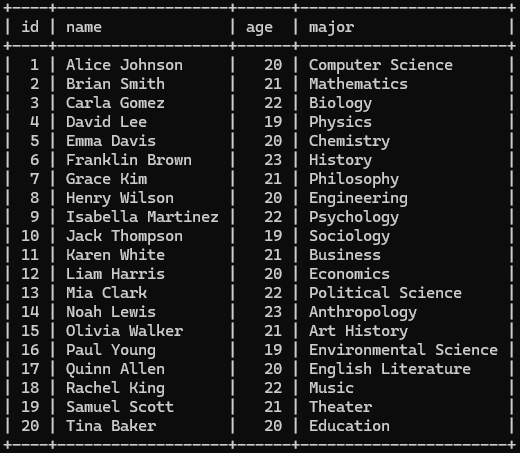
id INT AUTO\_INCREMENT PRIMARY KEY, name VARCHAR(100),

age INT,

major VARCHAR(100)

);

Sample Records:



Java:

package practice\_java; import javax.swing.\*; import java.awt.\*; import java.awt.event.\*; import java.sql.\*;

public class Assignment\_No\_10 extends JFrame { JTextField idField, nameField, ageField, majorField; JButton firstBtn, nextBtn, prevBtn, lastBtn;

Connection conn;

Statement stmt;

ResultSet rs;

public Assignment\_No\_10() { super("Student Record Navigator");

setLayout(new GridLayout(6, 2)); add(new JLabel("ID:"));

idField = new JTextField(20);

idField.setEditable(false); add(idField);

add(new JLabel("Name:")); nameField = new JTextField(20); nameField.setEditable(false); add(nameField);

add(new JLabel("Age:")); ageField = new JTextField(20); ageField.setEditable(false); add(ageField);

add(new JLabel("Major:")); majorField = new JTextField(20); majorField.setEditable(false); add(majorField);

firstBtn = new JButton("First"); nextBtn = new JButton("Next"); prevBtn = new JButton("Previous"); lastBtn = new JButton("Last");

add(firstBtn); add(nextBtn);

add(prevBtn); add(lastBtn);

connectDatabase();

firstBtn.addActionListener(e -> navigate("FIRST")); nextBtn.addActionListener(e -> navigate("NEXT")); prevBtn.addActionListener(e -> navigate("PREVIOUS")); lastBtn.addActionListener(e -> navigate("LAST"));

navigate("FIRST");

setSize(400, 250); setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***); setVisible(true);

}

void connectDatabase() { try {

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

conn = DriverManager.*getConnection*("jdbc:mysql://localhost:3308/student", "root", "tiger");

stmt = conn.createStatement(ResultSet.***TYPE\_SCROLL\_INSENSITIVE***, ResultSet.***CONCUR\_READ\_ONLY***);

createTableIfNotExists();

rs = stmt.executeQuery("SELECT \* FROM students");

} catch (Exception e) {

JOptionPane.*showMessageDialog*(this, "Database Connection Failed: " + e);

}

}

void createTableIfNotExists() { try {

String sql = """

CREATE TABLE IF NOT EXISTS students (

id INT AUTO\_INCREMENT PRIMARY KEY, name VARCHAR(100),

age INT,

major VARCHAR(100)

); """;

stmt.executeUpdate(sql);

} catch (SQLException e) {

JOptionPane.*showMessageDialog*(this, "Table Creation Failed: " + e);

}

}

void navigate(String direction) {

try {

switch (direction) { case "FIRST": rs.first();

break;

case "NEXT":

if (!rs.isLast()) rs.next(); break;

case "PREVIOUS":

if (!rs.isFirst()) rs.previous(); break;

case "LAST": rs.last(); break;

}

displayRecord();

} catch (SQLException e) {

JOptionPane.*showMessageDialog*(this, "Navigation Error: " + e);

}

}

void displayRecord() { try {

idField.setText(String.*valueOf*(rs.getInt("id"))); nameField.setText(rs.getString("name")); ageField.setText(String.*valueOf*(rs.getInt("age"))); majorField.setText(rs.getString("major"));

} catch (SQLException e) {

JOptionPane.*showMessageDialog*(this, "Error Displaying Record: " + e);

}

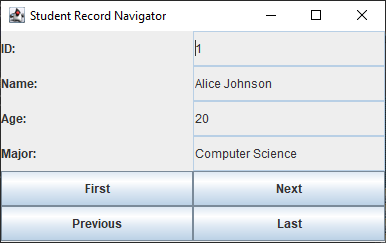
}

public static void main(String[] args) { new Assignment\_No\_10();

}

}

**Output:**

****