

Lab Sheet 3

[1] WAP in java to insert, select, modify and delete data of a database.

Source code:

```
package Lab3;

import java.sql.*;

public class Question1 {
    private static final String URL = "jdbc:mysql://localhost:3306/Student";
    private static final String USER = "root";
    private static final String PASSWORD = "rootpassword";
    public static void main(String[] args) {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            insertData();
            readData();
            modifyData();
            deleteData();
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }

    private static void deleteData() {
        try (Connection con = DriverManager.getConnection(URL, USER, PASSWORD)) {
            String sql = "DELETE FROM studenttable WHERE roll = ?";
            try (PreparedStatement statement = con.prepareStatement(sql)) {
                statement.setInt(1, 2); // Specify the roll value to delete
                int rowsAffected = statement.executeUpdate();
                System.out.println(rowsAffected + " record(s) deleted.");
            }
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    private static void modifyData() {
        try (Connection con = DriverManager.getConnection(URL, USER, PASSWORD)) {
            String sql = "UPDATE studenttable SET name = ?, roll = ?, height = ? WHERE roll = ?";
            try (PreparedStatement statement = con.prepareStatement(sql)) {
                statement.setString(1, "Karan");
                statement.setInt(2, 5);
                statement.setFloat(3, (float) 1.5);
                statement.setInt(4, 1); // Specify the roll value to update
                int rowsAffected = statement.executeUpdate();
                System.out.println(rowsAffected + " record(s) updated.");
            }
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

```

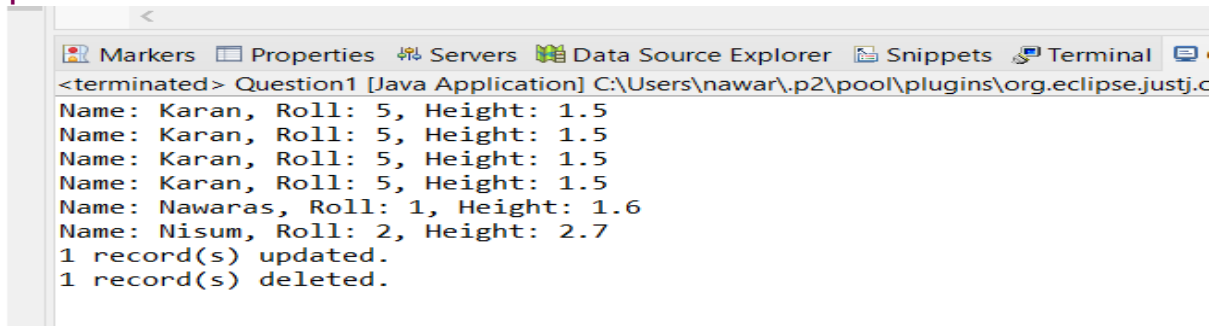
}
private static void readData() {
    try (Connection con = DriverManager.getConnection(URL, USER, PASSWORD);
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM studenttable")) {

        while (rs.next()) {
            System.out.println("Name: " + rs.getString("name")
                               + ", Roll: " + rs.getInt("roll")
                               + ", Height: " + rs.getFloat("height"));
        }
    } catch (SQLException e) {
        System.out.println(e.getMessage());
    }
}

private static void insertData() {
    String insertQuery = "INSERT INTO studenttable (name, roll, height) VALUES
(?, ?, ?)";
    try (Connection con = DriverManager.getConnection(URL, USER, PASSWORD)) {
        // Inserting the first record
        try (PreparedStatement preparedStatement1 =
con.prepareStatement(insertQuery)) {
            preparedStatement1.setString(1, "Nawaras");
            preparedStatement1.setInt(2, 1);
            preparedStatement1.setFloat(3, (float) 1.6);
            int rowsAffected1 = preparedStatement1.executeUpdate();
            System.out.println(rowsAffected1 + " record(s) inserted");
        }
        // Inserting the second record
        try (PreparedStatement preparedStatement2 =
con.prepareStatement(insertQuery)) {
            preparedStatement2.setString(1, "Nisum");
            preparedStatement2.setInt(2, 2);
            preparedStatement2.setFloat(3, (float) 2.7);
            int rowsAffected2 = preparedStatement2.executeUpdate();
            System.out.println(rowsAffected2 + " record(s) inserted");
        }
    } catch (SQLException e) {
        System.out.println(e.getMessage());
    }
}
}

```

Output:

A screenshot of the Eclipse IDE's terminal window. The title bar shows 'Question1 [Java Application] C:\Users\nawar\.p2\pool\plugins\org.eclipse.justj.c'. The terminal output displays a list of records: 'Name: Karan, Roll: 5, Height: 1.5' (repeated four times), 'Name: Nawaras, Roll: 1, Height: 1.6', and 'Name: Nisum, Roll: 2, Height: 2.7'. Below the records, it shows '1 record(s) updated.' and '1 record(s) deleted.'.

```
<terminated> Question1 [Java Application] C:\Users\nawar\.p2\pool\plugins\org.eclipse.justj.c
Name: Karan, Roll: 5, Height: 1.5
Name: Karan, Roll: 5, Height: 1.5
Name: Karan, Roll: 5, Height: 1.5
Name: Karan, Roll: 5, Height: 1.5
Name: Nawaras, Roll: 1, Height: 1.6
Name: Nisum, Roll: 2, Height: 2.7
1 record(s) updated.
1 record(s) deleted.
```

[2] WAP in java to insert two numbers from UI and then find the sum of these numbers and display the sum. [Hint: Use GUI controls and database connectivity]

Source code:

```
package Lab3;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;
import javax.swing.*;

public class Question2 {
    JFrame frame;
    JTextField textField;
    JTextField textField_1;
    JButton btn1;
    JLabel lblNewLabel_2 ;
    public Question2(){
        frame = new JFrame();
        frame.setBounds(100, 100, 500, 500);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.getContentPane().setLayout(null);
        JLabel lblNewLabel = new JLabel("Enter number:");
        lblNewLabel.setBounds(36, 98, 110, 20);
        frame.getContentPane().add(lblNewLabel);
        JLabel lblNewLabel_1 = new JLabel("Enter number:");
        lblNewLabel_1.setBounds(36, 146, 110, 20);
        frame.getContentPane().add(lblNewLabel_1);
        textField = new JTextField();
        textField.setBounds(167, 99, 207, 19);
        frame.getContentPane().add(textField); textField.setColumns(10);

        textField_1 = new JTextField();
        textField_1.setBounds(167, 147, 207, 19);
        frame.getContentPane().add(textField_1);
        textField_1.setColumns(10);
        btn1 = new JButton("Add");
        btn1.setBounds(152, 210, 115, 21);
        frame.getContentPane().add(btn1);
        btn1.addActionListener(new ActionListener() {
```

```

@Override
public void actionPerformed(ActionEvent e) {

    int a = Integer.parseInt(textField.getText());
    int b = Integer.parseInt(textField_1.getText());
    int c = a+b;
    System.out.println(c);
    lblNewLabel_2.setText("Result: "+c+" And Added to DB");

    String URL = "jdbc:mysql://localhost:3306/Addddb";
    String USER = "root";
    String PASSWORD = "rootpassword";

    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        String insertQuery = "INSERT INTO Addtable (sum) VALUES
(?)";

        Connection con =
DriverManager.getConnection(URL,USER,PASSWORD);

        PreparedStatement preparedStatement=
con.prepareStatement(insertQuery);

        preparedStatement.setInt(1, c);//
        int rowsAffected1 =
preparedStatement.executeUpdate();
        System.out.println(rowsAffected1 + " record(s)
inserted");

    }catch(Exception e1) {
        System.out.println(e1.getMessage());
    }
});
lblNewLabel_2= new JLabel("Result:");
lblNewLabel_2.setBounds(36, 176, 200, 20);

frame.getContentPane().add(lblNewLabel_2);
frame.setVisible(true);
}
public static void main(String[] args) {

    new Question2();
}
}

```

Output:

Enter number:

Enter number:

Result: 22 And Added to DB

[3] WAP in java to implement DDL statement.

Source code:

```
package Lab3;
```

```
import java.sql.*;
```

```
public class Question3 {
```

```
    private static final String URL = "jdbc:mysql://localhost:3306/Question3";
```

```
    private static final String USER = "root";
```

```
    private static final String PASSWORD = "rootpassword";
```

```
    public static void main(String[] args) {
```

```
        try {
```

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

```
            createTable();
```

```
        } catch (ClassNotFoundException | SQLException e) {
```

```
            e.printStackTrace();
```

```
        }
```

```
    }
```

```
    private static void createTable() throws SQLException {
```

```
        try (Connection connection = DriverManager.getConnection(URL, USER, PASSWORD);
```

```
            Statement statement = connection.createStatement()) {
```

```
            String createTableSQL = "CREATE TABLE hamroStudent (" +
```

```
                "id INT PRIMARY KEY," +
```

```
                "name VARCHAR(255)," +
```

```
                "position VARCHAR(255)," +
```

```
                "salary DOUBLE)";
```

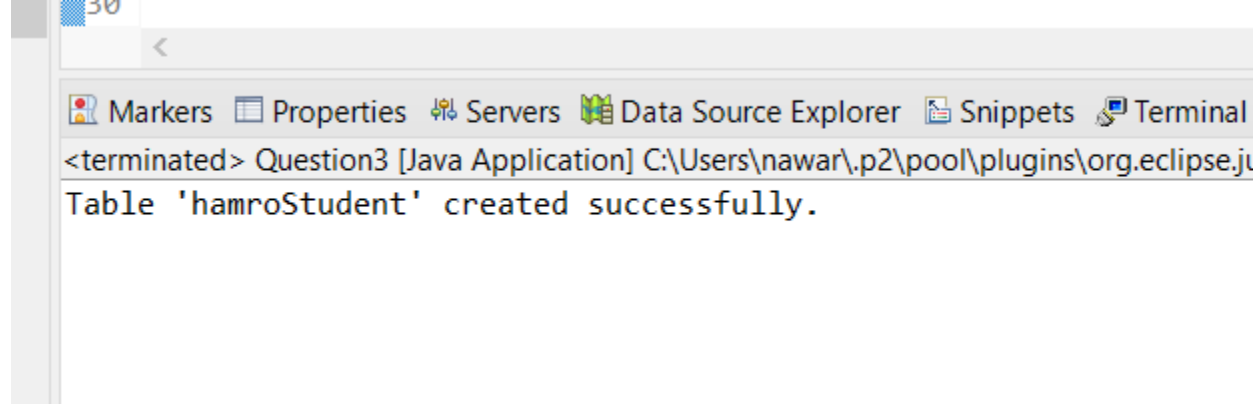
```

        statement.execute(createTableSQL);

        System.out.println("Table 'hamroStudent' created successfully.");
    }
}

```

Output:



[4] WAP in java to implement DML statements.

Source code:

```

package Lab3;

import java.sql.*;

public class Question4 {
    private static final String URL = "jdbc:mysql://localhost:3306/Question4";
    private static final String USER = "root";
    private static final String PASSWORD = "rootpassword";

    public static void main(String[] args) {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            insertData();
            updateData();
            deleteData();
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
    }

    private static void insertData() throws SQLException {
        try (Connection connection = DriverManager.getConnection(URL, USER,
PASSWORD)) {

            Statement statement = connection.createStatement();
            String createTableSQL = "CREATE TABLE employees (" +
                "id INT PRIMARY KEY," +
                "name VARCHAR(255)," +

```

```

        "position VARCHAR(255)," +
        "salary DOUBLE)";

statement.execute(createTableSQL);

String insertSQL = "INSERT INTO employees (id, name, position, salary)
VALUES (?, ?, ?, ?)";
try (PreparedStatement preparedStatement =
connection.prepareStatement(insertSQL)) {
    preparedStatement.setInt(1, 1);
    preparedStatement.setString(2, "Nawaras");
    preparedStatement.setString(3, "Software Engineer");
    preparedStatement.setDouble(4, 75000.0);

    int rowsInserted = preparedStatement.executeUpdate();
    System.out.println(rowsInserted + " row(s) inserted.");
}
}

private static void updateData() throws SQLException {
    try (Connection connection = DriverManager.getConnection(URL, USER,
PASSWORD)) {
        String updateSQL = "UPDATE employees SET salary = ? WHERE name = ?";
        try (PreparedStatement preparedStatement =
connection.prepareStatement(updateSQL)) {
            preparedStatement.setDouble(1, 80000.0);
            preparedStatement.setString(2, "John Doe");

            int rowsUpdated = preparedStatement.executeUpdate();
            System.out.println(rowsUpdated + " row(s) updated.");
        }
    }
}

private static void deleteData() throws SQLException {
    try (Connection connection = DriverManager.getConnection(URL, USER,
PASSWORD)) {
        String deleteSQL = "DELETE FROM employees WHERE name = ?";
        try (PreparedStatement preparedStatement =
connection.prepareStatement(deleteSQL)) {
            preparedStatement.setString(1, "John Doe");

            int rowsDeleted = preparedStatement.executeUpdate();
            System.out.println(rowsDeleted + " row(s) deleted.");
        }
    }
}
}

```

Output:

```
<terminated> Question4 [Java Application] C:\Users\nawar  
1 row(s) inserted.  
0 row(s) updated.  
0 row(s) deleted.
```