



Practical 2. JDBC Connectivity & CRUD Operations.

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

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

    public static void main (String [] args) {
        try (Connection connection = DriverManager.getConnection
            (URL, USER, PASSWORD)) {
            Scanner scanner = new Scanner (System.in);
            int choice;
            do {
                System.out.println ("In --- Student Information
                    System --- ");
                System.out.println ("1. Insert Student");
                System.out.println ("2. Update Student");
                System.out.println ("3. Delete Student");
                System.out.println ("4. Display All Students");
                System.out.println ("5. Exit");
                System.out.print ("Enter your choice: ");
                choice = scanner.nextInt();
                switch (choice) {
                    case 1:
                        insertStudent (connection, scanner);
                        break;
                }
            } while (choice != 5);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    static void insertStudent (Connection connection, Scanner scanner) {
        String name, address, phone;
        int id;
        id = scanner.nextInt();
        name = scanner.nextLine();
        address = scanner.nextLine();
        phone = scanner.nextLine();
        String query = "insert into student values (" + id + ", "
            + "' " + name + " ', "
            + "' " + address + " ', "
            + "' " + phone + " ' )";
        Statement statement = connection.createStatement();
        statement.executeUpdate (query);
    }
}
```



case 2 :

updateStudent (connection, scanner);
break;

case 3 :

deleteStudent (connection, scanner);
break;

case 4 :

displayStudents (connection);
break;

case 5 :

System.out.println ("Exiting...");
break;

default:

System.out.println ("Invalid choice !
Please try again.");

{

} while (choice != 5);

scanner.close();

} catch (SQLException e) {

e.printStackTrace();

{

{

private static void insertStudent (Connection
connection, Scanner scanner) {

try {

System.out.print ("Enter Student ID: ");

int id = scanner.nextInt();

scanner.nextLine();

System.out.print ("Enter Name: ");

String name = scanner.nextLine();



```
System.out.print("Enter Age: ");
int age = scanner.nextInt();
scanner.nextLine();
System.out.print("Enter Course: ");
String course = scanner.nextLine();
String sql = "INSERT INTO student
(Cstudent_id, name, age, course) VALUES
(?, ?, ?, ?);"
try {
PreparedStatement statement =
connection.prepareStatement(sql);
statement.setInt(1, id);
statement.setString(2, name);
statement.setInt(3, age);
statement.setString(4, course);
int rows = statement.executeUpdate();
if (rows > 0) {
System.out.println("Student inserted
successfully!");
}
} catch (SQLException e) {
System.out.println("Error inserting
student:" + e.getMessage());
}
private static void updateStudent(Connection
connection, Scanner scanner) {
try {
System.out.print("Enter Student ID to
Update: ");
int id = scanner.nextInt();
}
```



```
scanner.nextLine();
System.out.print("Enter New Name: ");
String name = scanner.nextLine();
System.out.print("Enter New Age: ");
int age = scanner.nextInt();
scanner.nextLine();
System.out.print("Enter New Course: ");
String course = scanner.nextLine();
String sql = "UPDATE student SET name=?
, age = ?, course = ? WHERE student_id=?";
try {
PreparedStatement statement =
connection.prepareStatement(sql);
statement.setString(1, name);
statement.setInt(2, age);
statement.setString(3, course);
statement.setInt(4, id);
int rows = statement.executeUpdate();
if (rows > 0) {
System.out.println("Student updated
successfully!");
} else {
System.out.println("Student ID
not found.");
}
} catch (SQLException e) {
System.out.println("Error updating
Student: " + e.getMessage());
}
```



```
private static void deleteStudent(Connection connection, Scanner scanner) {
    try {
        System.out.print("Enter Student ID to Delete : ");
        int id = scanner.nextInt();
        String sql = "DELETE FROM student WHERE student_id = ? ";
        try (PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setInt(1, id);
            int rows = statement.executeUpdate();
            if (rows > 0) {
                System.out.println("Student deleted successfully!");
            } else {
                System.out.println("Student ID not found.");
            }
        }
    } catch (SQLException e) {
        System.out.println("Error deleting student: " + e.getMessage());
    }
}

private static void displayStudents(Connection connection) {
    try {
        String sql = "SELECT * FROM student";
        try (Statement statement = connection.createStatement()) {
```



ResultSet resultSet = statement.executeQuery(sql);
System.out.println("In -- Student Records ---");
while (resultSet.next()) {

int id = resultSet.getInt("student_id");

String name = resultSet.getString("name");

int age = resultSet.getInt("age");

String course = resultSet.getString("course");

System.out.println("ID: " + id + ", Name: "

+ name + ", Age: " +

age + ", Course: " +

course);

}

}

} catch (SQLException e) {

System.out.println("Error displaying
students: " + e.getMessage());

}

}

spat