### Experiment\_5: CRUD Application with MySQL Database

#### Prerequisites:

1. Java Development Kit (JDK) 14 or later (required for switch expressions)
2. MySQL Server installed and running
3. MySQL Connector/J (version 9.1.0 or compatible)
4. MySQL database named “csd\_students\_info” with a table “students”

#### Database Setup:

1. Create the database:

* CREATE DATABASE csd\_students\_info;  
  USE csd\_students\_info;

1. Create the students table:

* CREATE TABLE students (  
   id VARCHAR(20) PRIMARY KEY,  
   name VARCHAR(100) NOT NULL,  
   age INT,  
   grade VARCHAR(2)  
  );

#### Setup Instructions:

1. Download MySQL Connector/J from https://dev.mysql.com/downloads/connector/j/
2. Extract the JAR file and note its location (e.g., C:-connector-j-9.1.0.jar)

#### Compilation and Execution:

1. Compile the application:

* javac -d . CRUDApplication.java

1. Run the application (replace the path with your actual MySQL Connector/J path):

* java -cp ".;C:\path\to\mysql-connector-j-9.1.0.jar" Skill\_Development\_Lab.Experiment\_5.CRUDApplication

#### Application Usage:

1. When the application starts, you’ll see a menu with these options:
   * 1. Add Student
     2. View Students
     3. Update Student Grade
     4. Delete Student
     5. Exit
2. Follow the on-screen prompts to perform CRUD operations.

#### Notes:

* The application connects to MySQL using:
  + URL: jdbc:mysql://localhost:3306/csd\_students\_info
  + Username: root
  + Password: 123
* You may need to modify these credentials in the CRUDApplication.java file if your MySQL setup is different
* The folder name which contain CRUDApplication.java should be named as Experiment\_5

#### Source Code:

package Skill\_Development\_Lab.Experiment\_5;

import java.sql.\*;

import java.util.Scanner;

public class CRUDApplication {

    // Database URL and credentials

    private static final String DB\_URL = "jdbc:mysql://localhost:3306/csd\_students\_info";

    private static final String DB\_USER = "root";

    private static final String DB\_PASSWORD = "mysql";

    // Method to establish a connection to the database

    public static Connection getConnection() throws SQLException {

        return DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD);

    }

    public static void main(String[] args) {

        try (Connection conn = getConnection(); Scanner scanner = new Scanner(System.in)) {

            System.out.println("Connected to the database.");

            boolean exit = false;

            while (!exit) {

                System.out.println("\n1. Add Student\n2. View Students\n3. Update Student Grade\n4. Delete Student\n5. Exit");

                System.out.print("Choose an option: ");

                int choice = scanner.nextInt();

                scanner.nextLine(); // Clear newline character

                switch (choice) {

                    case 1 -> addStudent(conn, scanner);

                    case 2 -> viewStudents(conn);

                    case 3 -> updateStudent(conn, scanner);

                    case 4 -> deleteStudent(conn, scanner);

                    case 5 -> {

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

                        exit = true;

                    }

                    default -> System.out.println("Invalid choice. Try again.");

                }

            }

        } catch (SQLException e) {

            System.err.println("Database connection error: " + e.getMessage());

            e.printStackTrace();

        }

    }

    private static void addStudent(Connection conn, Scanner scanner) {

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

        String id = scanner.nextLine();

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

        String name = scanner.nextLine();

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

        int age = scanner.nextInt();

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

        String grade = scanner.next();

        String sql = "INSERT INTO students (id, name, age, grade) VALUES (?, ?, ?, ?)";

        try (PreparedStatement stmt = conn.prepareStatement(sql)) {

            stmt.setString(1, id);

            stmt.setString(2, name);

            stmt.setInt(3, age);

            stmt.setString(4, grade);

            int rowsAffected = stmt.executeUpdate();

            System.out.println(rowsAffected + " student added.");

        } catch (SQLException e) {

            System.err.println("Error adding student: " + e.getMessage());

        }

    }

    private static void viewStudents(Connection conn) {

        String sql = "SELECT \* FROM students";

        try (Statement stmt = conn.createStatement(); ResultSet rs = stmt.executeQuery(sql)) {

            if (!rs.isBeforeFirst()) {

                System.out.println("No students found.");

            } else {

                while (rs.next()) {

                    System.out.println("ID: " + rs.getString("id") + ", Name: " + rs.getString("name") +

                            ", Age: " + rs.getInt("age") + ", Grade: " + rs.getString("grade"));

                }

            }

        } catch (SQLException e) {

            System.err.println("Error retrieving students: " + e.getMessage());

        }

    }

    private static void updateStudent(Connection conn, Scanner scanner) {

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

        String id = scanner.nextLine();

        System.out.print("Enter new grade: ");

        String grade = scanner.nextLine();

        String sql = "UPDATE students SET grade = ? WHERE id = ?";

        try (PreparedStatement stmt = conn.prepareStatement(sql)) {

            stmt.setString(1, grade);

            stmt.setString(2, id);

            int rowsAffected = stmt.executeUpdate();

            if (rowsAffected > 0) {

                System.out.println("Student updated.");

            } else {

                System.out.println("Student with ID " + id + " not found.");

            }

        } catch (SQLException e) {

            System.err.println("Error updating student: " + e.getMessage());

        }

    }

    private static void deleteStudent(Connection conn, Scanner scanner) {

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

        String id = scanner.nextLine();

        String sql = "DELETE FROM students WHERE id = ?";

        try (PreparedStatement stmt = conn.prepareStatement(sql)) {

            stmt.setString(1, id);

            int rowsAffected = stmt.executeUpdate();

            if (rowsAffected > 0) {

                System.out.println("Student deleted.");

            } else {

                System.out.println("Student with ID " + id + " not found.");

            }

        } catch (SQLException e) {

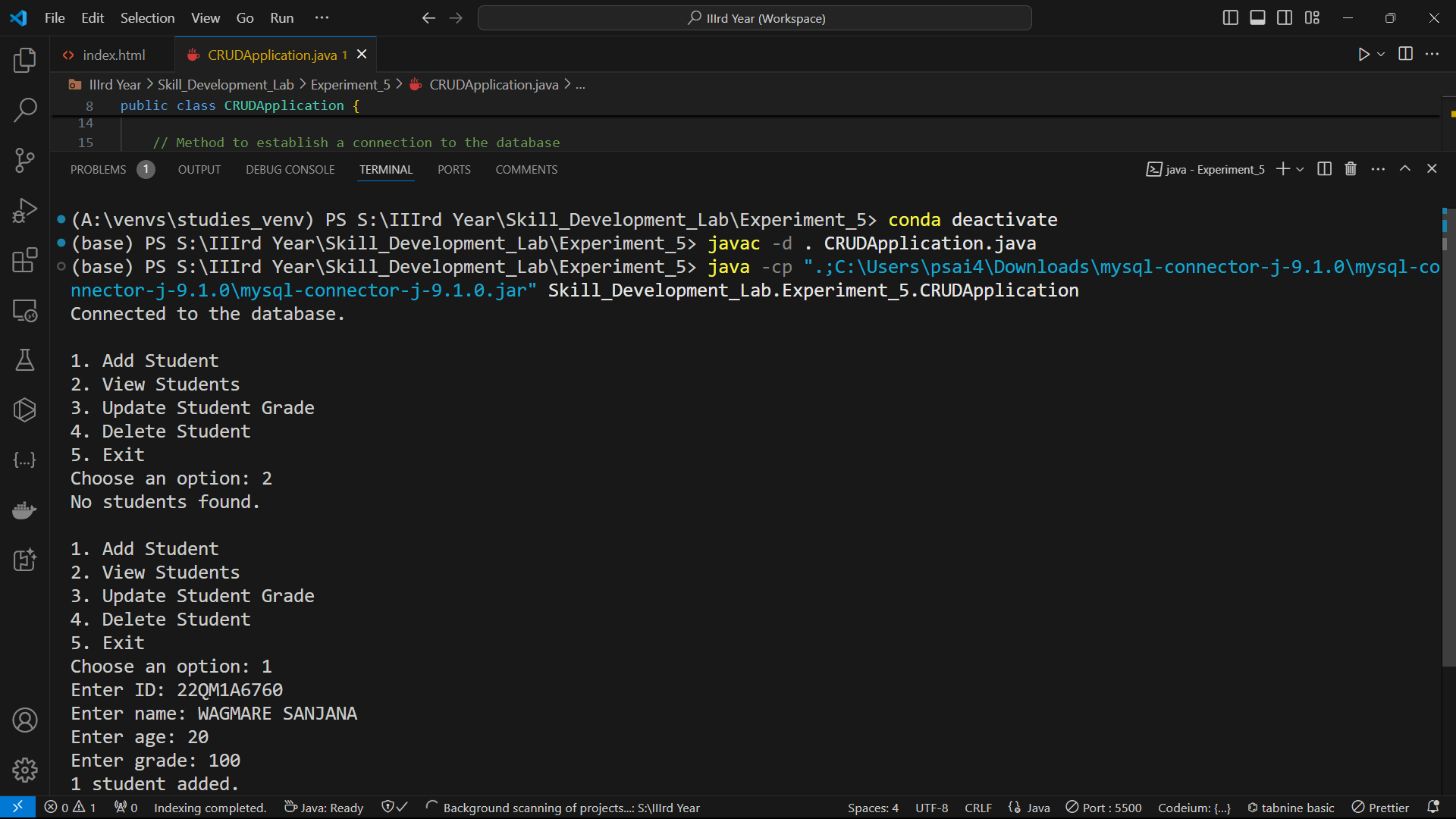
            System.err.println("Error deleting student: " + e.getMessage());

        }

    }

}

**READ Operation:**



**READ UPDATE Operation:**

A screenshot of a computer

AI-generated content may be incorrect.

**UDATE DELETE Operation**

