Week 10 : Programming Assignment 4

Due on 2025-04-03, 23:59 IST

Create a Table in a SQLite Database Using JDBC

Once a connection to a database is established, SQL commands can be executed using JDBC.

In this task, your job is to create a table named `students` with the following columns:

- `roll` – an integer that represents the student's roll number
- `name` – a string (up to 30 characters) representing the student's name

You will complete one line of code that executes a SQL `CREATE TABLE` statement using a `Statement` object.

You are not required to insert or retrieve any data — only to create the table using proper JDBC methods.

Private Test cases used for evaluation	Input	Expected Output	Actual Output	Status
Test Case 1		success	success\n	Passed

The due date for submitting this assignment has passed.
1 out of 1 tests passed.
You scored 100.0/100.

Assignment submitted on 2025-03-26, 22:35 IST

Your last recorded submission was :

```java
import java.sql.*; // Required for JDBC classes

public class W10_P4 {
    public static void main(String[] args) {
        try {
            // Set SQLite temp directory to avoid native driver errors in restricted environments
            System.setProperty("org.sqlite.tmpdir", "/tempfs");

            // Create a connection to the SQLite database
            Connection conn = DriverManager.getConnection("jdbc:sqlite:/tempfs/studentdb");

            // Create a Statement object to send SQL statements to the database
            Statement stmt = conn.createStatement();

            // SQL query string to create the 'students' table
            String sql = "CREATE TABLE students(roll INTEGER, name VARCHAR(30))";
stmt.executeUpdate(sql);
// If the table is created without exceptions, print success
            System.out.println("success");

            // Close statement and connection to release resources
            stmt.close();
            conn.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

Sample solutions (Provided by instructor)

```java
import java.sql.*; // Required for JDBC classes

public class W10_P4 {
    public static void main(String[] args) {
        try {
            // Set SQLite temp directory to avoid native driver errors in restricted environments
            System.setProperty("org.sqlite.tmpdir", "/tempfs");

            // Create a connection to the SQLite database
            Connection conn = DriverManager.getConnection("jdbc:sqlite:/tempfs/studentdb");

            // Create a Statement object to send SQL statements to the database
            Statement stmt = conn.createStatement();

            // SQL query string to create the 'students' table
            String sql = "CREATE TABLE students(roll INTEGER, name VARCHAR(30))";
// This line sends the SQL CREATE TABLE statement to the database for execution.
// The executeUpdate(...) method is used for SQL statements that modify the database
// such as CREATE, INSERT, UPDATE, and DELETE.
// If the table is created successfully, no exception is thrown.
stmt.executeUpdate(sql);
// If the table is created without exceptions, print success
            System.out.println("success");

            // Close statement and connection to release resources
            stmt.close();
            conn.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```