Week 11 : Programming Assignment 2

Due on 2025-04-10, 23:59 IST

Update a Player's Age Using JDBC (Improved Guided Version)

📝 Problem Statement

You are given a table `players` containing a player's UID, first name, last name, and age.
Your task is to update the age of the player whose UID is 2, using a `PreparedStatement`.

You only need to write **one line** of code to perform this update operation.

Private Test cases used for evaluation	Input	Expected Output	Actual Output	Status
Test Case 1		2 John Mayer 23	2 John Mayer 23\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 2025-04-10, 01:10 IST
Your last recorded submission was :

```java
import java.sql.*; // Required for database access

public class W11_P2 {
    public static void main(String args[]) {
        try {
            // Set SQLite temp directory (necessary in NPTEL environment)
            System.setProperty("org.sqlite.tmpdir", "/tempfs");

            // Connect to SQLite database file
            Connection conn = DriverManager.getConnection("jdbc:sqlite:/tempfs/db");

            // Create a Statement object to run SQL queries
            Statement stmt = conn.createStatement();

            // Create the table if it does not already exist
            String CREATE_TABLE_SQL = "CREATE TABLE IF NOT EXISTS players (UID INT, First_Name VARCHAR(45), Last_Name VAR
            stmt.executeUpdate(CREATE_TABLE_SQL);

            // Clean up old data and insert one row
            stmt.executeUpdate("DELETE FROM players");
            stmt.executeUpdate("INSERT INTO players VALUES(2, 'John', 'Mayer', 22)");

            // Prepare SQL update query with placeholders ( ? )
            String sql = "UPDATE players SET Age = ? WHERE UID = ?";
            PreparedStatement pstmt = conn.prepareStatement(sql);
pstmt.setInt(1, 23);
            pstmt.setInt(2, 2);
            pstmt.executeUpdate();

// Check the updated row by selecting and printing it
            ResultSet rs = stmt.executeQuery("SELECT * FROM players WHERE UID = 2");

            while (rs.next()) {
                System.out.println(rs.getInt(1) + "  " +
                                   rs.getString(2) + "  " +
                                   rs.getString(3) + "  " +
                                   rs.getInt(4));
            }

            // Close the connection
            conn.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

Sample solutions (Provided by instructor)

```java
import java.sql.*; // Required for database access

public class W11_P2 {
    public static void main(String args[]) {
        try {
            // Set SQLite temp directory (necessary in NPTEL environment)
            System.setProperty("org.sqlite.tmpdir", "/tempfs");

            // Connect to SQLite database file
            Connection conn = DriverManager.getConnection("jdbc:sqlite:/tempfs/db");

            // Create a Statement object to run SQL queries
            Statement stmt = conn.createStatement();

            // Create the table if it does not already exist
            String CREATE_TABLE_SQL = "CREATE TABLE IF NOT EXISTS players (UID INT, First_Name VARCHAR(45), Last_Name VAR
            stmt.executeUpdate(CREATE_TABLE_SQL);

            // Clean up old data and insert one row
            stmt.executeUpdate("DELETE FROM players");
            stmt.executeUpdate("INSERT INTO players VALUES(2, 'John', 'Mayer', 22)");

            // Prepare SQL update query with placeholders ( ? )
            String sql = "UPDATE players SET Age = ? WHERE UID = ?";
            PreparedStatement pstmt = conn.prepareStatement(sql);
pstmt.setInt(1, 23); // Bind 23 to the first placeholder (new age)
pstmt.setInt(2, 2);  // Bind 2 to the second placeholder (UID to match)
pstmt.executeUpdate(); // Perform the update

/*
Explanation for beginners:
- The SQL statement has two placeholders: one for the age and one for the UID.
- The '1' and '2' in setInt refer to the positions of ? in the query.
- This approach is safer and cleaner than using plain string concatenation in SQL.
*/
// Check the updated row by selecting and printing it
            ResultSet rs = stmt.executeQuery("SELECT * FROM players WHERE UID = 2");

            while (rs.next()) {
                System.out.println(rs.getInt(1) + "  " +
                                   rs.getString(2) + "  " +
                                   rs.getString(3) + "  " +
                                   rs.getInt(4));
            }

            // Close the connection
            conn.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```