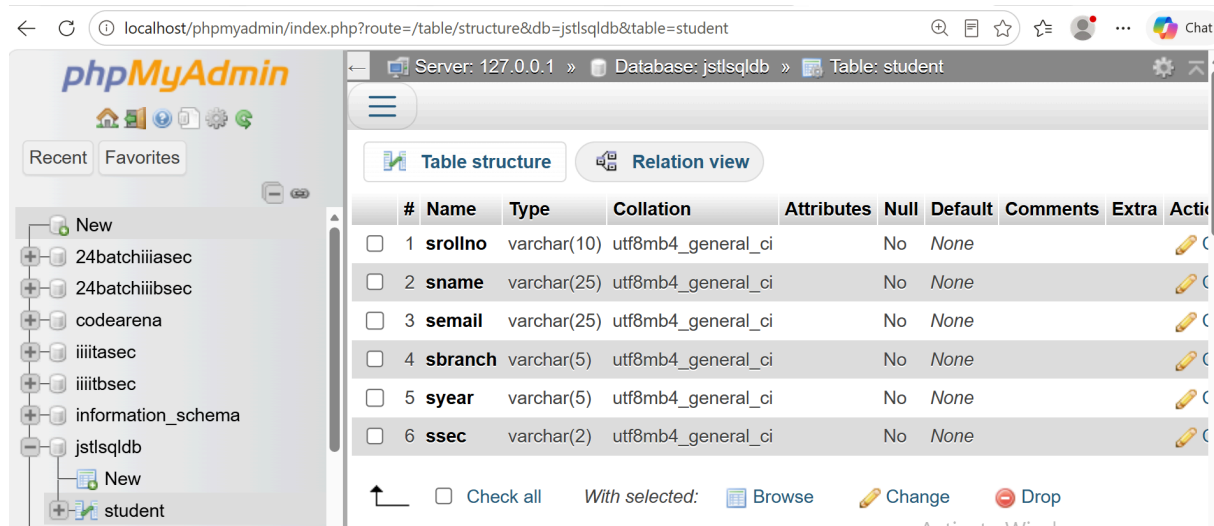


Week 15 Write a JSP JSTL Program perform CRUD operation on student table

Student Table Attributes



The screenshot shows the phpMyAdmin interface. The left sidebar displays the database structure with 'jstlsqldb' selected and 'student' table highlighted. The main area shows the 'Table structure' view for the 'student' table. The table has the following attributes:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	srollno	varchar(10)	utf8mb4_general_ci		No	None			
2	sname	varchar(25)	utf8mb4_general_ci		No	None			
3	semail	varchar(25)	utf8mb4_general_ci		No	None			
4	sbranch	varchar(5)	utf8mb4_general_ci		No	None			
5	seyear	varchar(5)	utf8mb4_general_ci		No	None			
6	ssec	varchar(2)	utf8mb4_general_ci		No	None			

Index.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>Student Portal</h1>
```

```
<button type="button" onclick="window.location.href='viewjstlsql.jsp'">View  
Students</button><br>
```

```
<button type="button" onclick="window.location.href='insertjstlsql.jsp'">ADD Student</button>
```

```
</body>
```

```
</html>
```

viewjstlsql.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
```

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql" %>
```

```
<html>
```

```
<body>
```

```
<sql:setDataSource var="con"
```

```
driver="com.mysql.jdbc.Driver"
```

```
url="jdbc:mysql://localhost:3306/jstlsqldb"
```

```
user="root"
```

```
password=""/>
```

```
<c:choose>
```

```

<c:when test="${not empty con}">

    <h3>Database connected successfully</h3>

    <!-- Run SQL query -->

    <sql:query dataSource="${con}" var="result">

        SELECT * FROM student;

    </sql:query>

    <!-- Display records -->

    <table border="1" cellpadding="5" cellspacing="0">

        <tr>

            <th>Rollno</th>

            <th>Name</th>

            <th>Email</th>

            <th>Branch</th>

            <th>Year</th>

            <th>Section</th>

        </tr>

        <c:forEach var="row" items="${result.rows}">

            <tr>

                <td>${row.srollno}</td>

                <td>${row.sname}</td>

                <td>${row.semail}</td>

                <td>${row.sbranch}</td>

                <td>${row.syear}</td>

                <td>${row.ssec}</td>

                <td><a href="editform.jsp?id=${row.srollno}">Edit</a></td>

                <td><a href="deletestudent.jsp?id=${row.srollno}">Delete</a></td></tr>

            </tr>

        </c:forEach>

    </table>

    <br>

    <button type="button" onclick="window.location.href='insertjstlsql.jsp'">ADD
Students</button><br>

</c:when>

<c:otherwise>

```

```

        <h3>Database connection not successful</h3>

    </c:otherwise>

</c:choose>

</body>

</html>

insertjstlsql.jsp

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql" %>

<body>

    <form method="POST" action="insertjstlsql.jsp">

        <h3>Student Registration Form</h3><br>

        Enter RollNo:<input type="text" name="srollno" required><br>

        Enter Name:<input type="text" name="sname" required><br>

        Enter Email Id:<input type="email" name="semail" required><br>

        Choose Branch:

        <select name="sbranch" required>

            <option></option>

            <option value="IT">IT</option>

            <option value="CSE">CSE</option>

            <option value="ECE">ECE</option>

            <option value="EEE">EEE</option>

        </select> <br>

        Choose Year:

        <select name="syear" required>

            <option></option>

            <option value="I">I</option>

            <option value="II">II</option>

            <option value="III">III</option>

            <option value="IV">IV</option>

        </select> <br>

        Choose Section

        <select name="ssec" required>

```

```

        <option></option>

        <option value="A">A</option>

        <option value="B">B</option>

    </select> <br>

    <input type="submit" value="Add Student"><br>

    <button type="button" onclick="window.location.href='viewjstlsql.jsp'">View
Students</button><br>

</form>

<sql:setDataSource var="con" driver="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/jstlsqlldb"
    user="root" password=""/>

<c:choose>

    <c:when test="${not empty con and not param.srollno}">

        <h3>Database connected successfully</h3>

        <c:set var="rollno" value="${param.srollno}"/>
        <c:set var="name" value="${param.sname}"/>
        <c:set var="email" value="${param.semail}"/>
        <c:set var="branch" value="${param.sbranch}"/>
        <c:set var="year" value="${param.syear}"/>
        <c:set var="section" value="${param.ssec}"/>
        <c:catch var="error">

            <sql:update dataSource="${con}" var="result"
sql="INSERT INTO student (srollno, sname, semail, sbranch, syear, ssec)
VALUES (?, ?, ?, ?, ?, ?)">

                <sql:param value="${rollno}"/>
                <sql:param value="${name}"/>
                <sql:param value="${email}"/>
                <sql:param value="${branch}"/>
                <sql:param value="${year}"/>
                <sql:param value="${section}"/>

            </sql:update>

        </c:catch>

    </c:choose>

    <c:when test="${empty error}">

```

```

        <h3>Record inserted successfully!</h3>

    </c:when>

    <c:otherwise>

        <h3>Error inserting record: ${error.message}</h3>

    </c:otherwise>

</c:choose>

</c:when>

<c:otherwise>

    <h3>Database connection not successfully</h3>

</c:otherwise>

</c:choose>
</body>
editform.jsp
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql" %>
<html>
<body>
<sql:setDataSource var="con"
    driver="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/jstlsqldb"
    user="root"
    password="" />
<!-- Retrieve student by rollno -->
<sql:query dataSource="${con}" var="student">
    SELECT * FROM student WHERE srollno = ?;
    <sql:param value="${param.id}" />
</sql:query>
<c:forEach var="row" items="${student.rows}">
    <h2>Edit Student Record</h2>
    <form action="updatestudent.jsp" method="post">
        <input type="hidden" name="srollno" value="${row.srollno}" />
    <table>
        <tr>

```

```

<td>Name:</td>

<td><input type="text" name="sname" value="{row.sname}" required /></td>

</tr>

<tr>

<td>Email:</td>

<td><input type="email" name="semail" value="{row.semail}" required /></td>

</tr>

<tr>

<td>Branch:</td>

<td>

<select name="sbranch" required>

<option value="">--Select Branch--</option>

<option value="CSE" <c:if test="{row.sbranch == 'CSE'}">selected</c:if>>CSE</option>

<option value="ECE" <c:if test="{row.sbranch == 'ECE'}">selected</c:if>>ECE</option>

<option value="EEE" <c:if test="{row.sbranch == 'EEE'}">selected</c:if>>EEE</option>

<option value="IT" <c:if test="{row.sbranch == 'IT'}">selected</c:if>>IT</option>

<option value="MECH" <c:if test="{row.sbranch ==
'MECH'}">selected</c:if>>MECH</option>

</select>

</td>

</tr>

<tr>

<td>Year:</td>

<td>

<select name="syear" required>

<option value="">--Select Year--</option>

<option value="I" <c:if test="{row.syear == 'I'}">selected</c:if>>I</option>

<option value="II" <c:if test="{row.syear == 'II'}">selected</c:if>>II</option>

<option value="III" <c:if test="{row.syear == 'III'}">selected</c:if>>III</option>

<option value="IV" <c:if test="{row.syear == 'IV'}">selected</c:if>>IV</option>

</select>

</td>

</tr>

<tr>

```

```

        <td>Section:</td>

        <td>

            <select name="ssec" required>

                <option value="">--Select Section--</option>

                <option value="A" <c:if test="\${row.ssec == 'A'}">selected</c:if>>A</option>

                <option value="B" <c:if test="\${row.ssec == 'B'}">selected</c:if>>B</option>

            </select>

        </td>

    </tr>

    <tr>

        <td colspan="2" style="text-align:center;">

            <input type="submit" value="Update Student" />

        </td>

    </tr>

</table>

</form>

</c:forEach>

</body>

</html>

deletestudent.jsp

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql" %>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<html>

<head>

    <title>Delete Student</title>

</head>

<body>

<sql:setDataSource var="con"

    driver="com.mysql.jdbc.Driver"

    url="jdbc:mysql://localhost:3306/jstlsqldb"

    user="root"

    password="" />

```

```

<!-- Delete the record -->

<sql:update dataSource="{con}">

    DELETE FROM student WHERE srollno = ?;

    <sql:param value="{param.id}" />
</sql:update>

<!-- Show alert and redirect -->

<script>

    alert("Student deleted successfully!");

    window.location.href = "viewjstlsql.jsp";

</script>
</body>
</html>
updatestudent.jsp

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<html>
<head>

    <title>Update Student</title>

</head>
<body>

<sql:setDataSource var="con"

    driver="com.mysql.jdbc.Driver"

    url="jdbc:mysql://localhost:3306/jstlsqldb"

    user="root"

    password="" />
<sql:update dataSource="{con}">

    UPDATE student

    SET sname = ?, semail = ?, sbranch = ?, syear = ?, ssec = ?

    WHERE srollno = ?;

    <sql:param value="{param.sname}" />

    <sql:param value="{param.semail}" />

    <sql:param value="{param.sbranch}" />

    <sql:param value="{param.syear}" />

```

```
<sql:param value="${param.ssec}" />

<sql:param value="${param.srollno}" />

</sql:update>

<script>

    alert("Student updated successfully!");

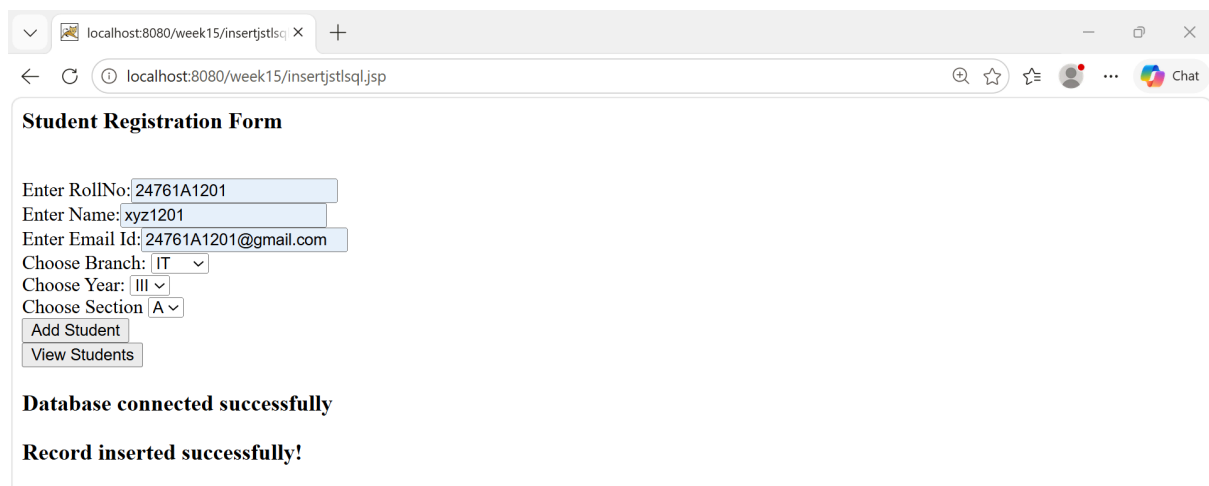
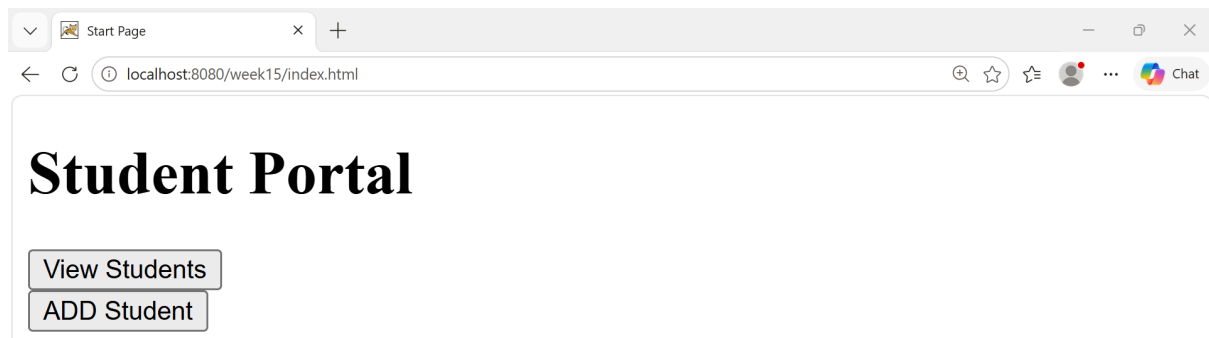
    window.location.href = "viewjstlsql.jsp";

</script>

</body>

</html>
```

Output:



localhost:8080/week15/insertjstlsq

localhost:8080/week15/insertjstlsql.jsp

Student Registration Form

Enter RollNo: 24761A1267

Enter Name: xyz12067

Enter Email Id: 24761A1267@gmail.com

Choose Branch: IT

Choose Year: III

Choose Section: B

Add Student

View Students

Database connected successfully

Record inserted successfully!

Retrieve Records using JSTL SQL

localhost:8080/week15/viewjstlsql.jsp

Database connected successfully

Rollno	Name	Email	Branch	Year	Section	
24761A1201	xyz1201	24761A1201@gmail.com	IT	III	A	Edit Delete
24761A1267	xyz12067	24761A1267@gmail.com	IT	III	B	Edit Delete

ADD Students

Edit Student

localhost:8080/week15/editform.jsp?id=24761A1267

Edit Student Record

Name: xyz12067 K

Email: 24761A1267@gmail.com

Branch: IT

Year: III

Section: B

Update Student

Update Student

localhost:8080/week15/updatestudent.jsp

localhost:8080 says

Student updated successfully!

OK

Retrieve Records using JSTL SQL

localhost:8080/week15/viewjstlsql.jsp

Database connected successfully

Rollno	Name	Email	Branch	Year	Section		
24761A1201	xyz1201	24761A1201@gmail.com	IT	III	A	Edit	Delete
24761A1267	xyz12067 K	24761A1267@gmail.com	IT	III	B	Edit	Delete

ADD Students

Delete Student

localhost:8080/week15/deletestudent.jsp?id=24761A1201

localhost:8080 says
Student deleted successfully!

OK

Retrieve Records using JSTL SQL

localhost:8080/week15/viewjstlsql.jsp

Database connected successfully

Rollno	Name	Email	Branch	Year	Section		
24761A1267	xyz12067 K	24761A1267@gmail.com	IT	III	B	Edit	Delete

ADD Students

Note add below dependencies if pom.xml under dependency

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.33</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.33</version>

</dependency>

<dependency>

<groupId>jakarta.servlet.jsp.jstl</groupId>

```
<artifactId>jakarta.servlet.jsp.jstl-api</artifactId>  
  <version>3.0.1</version>  
</dependency>  
<dependency>  
  <groupId>org.glassfish.web</groupId>  
  <artifactId>jakarta.servlet.jsp.jstl</artifactId>  
  <version>3.0.1</version>  
</dependency>
```

JSTL SQL Tags – Notes for Students

1. What is JSTL SQL?

JSTL SQL tags are a set of JSTL tags used to perform **database-related operations directly from a JSP page**.

JSTL provides the SQL tag library with the prefix:

Taglib declaration:

```
<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
```

Main purpose

JSTL SQL tags can be used to:

- Establish a database connection
- Execute SQL `SELECT` queries
- Execute `INSERT`, `UPDATE`, and `DELETE`
- Execute parameterized SQL queries
- Display query results
- Perform database transactions

Note: JSTL SQL tags are mainly useful for learning and simple JSP examples. In real-world applications, database operations are normally placed in **DAO/service classes**, rather than directly inside JSP.

2. JSTL SQL Tags

The important SQL tags are:

Tag	Purpose
<code><sql:setDataSource></code>	Creates/configures a database connection
<code><sql:query></code>	Executes a <code>SELECT</code> query
<code><sql:update></code>	Executes <code>INSERT</code> , <code>UPDATE</code> , or <code>DELETE</code>
<code><sql:param></code>	Supplies parameters to a SQL statement
<code><sql:dateParam></code>	Supplies a <code>java.util.Date</code> parameter
<code><sql:transaction></code>	Groups multiple SQL operations into a transaction

3. <sql:setDataSource>

This tag is used to define the database connection.

Syntax

```
<sql:setDataSource
  var="db"
  driver="com.mysql.cj.jdbc.Driver"
  url="jdbc:mysql://localhost:3306/college"
  user="root"
  password="root"/>
```

Explanation

- `var` → Name used to refer to the database connection
- `driver` → JDBC driver class
- `url` → Database URL
- `user` → Database username
- `password` → Database password

Example:

```
<sql:setDataSource
  var="db"
  driver="com.mysql.cj.jdbc.Driver"
  url="jdbc:mysql://localhost:3306/college"
  user="root"
  password="root"/>
```

Now `db` can be used by other SQL tags.

4. <sql:query>

The `<sql:query>` tag is used to execute a **SELECT** query.

Syntax

```
<sql:query var="result" dataSource="${db}">
  SELECT * FROM student
</sql:query>
```

Here:

- `result` stores the query result.
- `${db}` refers to the database connection.

5. Displaying Query Results

Suppose the `student` table contains:

id	name	branch
101	Ravi	IT
102	Sita	CSE
103	Kumar	ECE

JSTL program:

```
<%@ page contentType="text/html; charset=UTF-8" %>
```

```

<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<html>
<body>

<sql:setDataSource
    var="db"
    driver="com.mysql.cj.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/college"
    user="root"
    password="root"/>

<sql:query var="result" dataSource="${db}">
    SELECT * FROM student
</sql:query>

<h2>Student Details</h2>

<c:forEach var="row" items="${result.rows}">
    <c:out value="${row.id}"/>
    <c:out value="${row.name}"/>
    <c:out value="${row.branch}"/>
</c:forEach>

</body>
</html>

```

Output

Student Details

```

101 - Ravi - IT
102 - Sita - CSE
103 - Kumar - ECE

```

6. Understanding `${result.rows}`

This is important for students.

When we execute:

```

<sql:query var="result" dataSource="${db}">
    SELECT * FROM student
</sql:query>

```

the result is stored in:

```
result
```

The actual rows can be accessed using:

```
${result.rows}
```

Each row behaves like a map containing column names and values.

For example:

```

${row.id}
${row.name}

```

```
${row.branch}
```

7. <sql:update>

The <sql:update> tag is used for:

- INSERT
- UPDATE
- DELETE

INSERT Example

```
<sql:update dataSource="${db}">
    INSERT INTO student VALUES (104, 'Ramesh', 'IT')
</sql:update>
```

This inserts a new student.

8. UPDATE Example

```
<sql:update dataSource="${db}">
    UPDATE student
    SET branch='CSE'
    WHERE id=104
</sql:update>
```

This changes the branch of student 104.

9. DELETE Example

```
<sql:update dataSource="${db}">
    DELETE FROM student
    WHERE id=104
</sql:update>
```

This deletes student 104.

10. Why Do We Need <sql:param>?

Suppose the user enters an ID through a form.

We should **not** construct SQL like this:

```
SELECT * FROM student WHERE id=${param.id}
```

Instead, use a parameterized query.

```
<sql:query var="result" dataSource="${db}">
    SELECT * FROM student WHERE id = ?
    <sql:param value="${param.id}"/>
</sql:query>
```

Here:

?

is a placeholder.

The value is supplied by:

```
<sql:param value="${param.id}"/>
```

11. Example – Search Student

```
<%@ page contentType="text/html; charset=UTF-8" %>

<%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<sql:setDataSource
    var="db"
    driver="com.mysql.cj.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/college"
    user="root"
    password="root"/>

<form method="post">

    Enter Student ID:
    <input type="text" name="id">

    <input type="submit" value="Search">

</form>

<sql:query var="result" dataSource="${db}">
    SELECT * FROM student WHERE id = ?
    <sql:param value="${param.id}"/>
</sql:query>

<c:forEach var="row" items="${result.rows}">

    ID: ${row.id}<br>
    Name: ${row.name}<br>
    Branch: ${row.branch}<br>

</c:forEach>
```

12. <sql:dateParam>

<sql:dateParam> is used when a SQL statement requires a date value.

Example:

```
<sql:update dataSource="${db}">
    INSERT INTO student VALUES (?, ?, ?)

    <sql:param value="105"/>
    <sql:param value="Anil"/>
    <sql:dateParam value="${someDate}"/>
</sql:update>
```

It is useful when working with database date/time columns.

13. <sql:transaction>

The <sql:transaction> tag is used when multiple database operations should be treated as **one transaction**.

Example:

```
<sql:transaction dataSource="${db}">

    <sql:update>
        UPDATE account
        SET balance = balance - 1000
        WHERE id = 101
    </sql:update>

    <sql:update>
        UPDATE account
        SET balance = balance + 1000
        WHERE id = 102
    </sql:update>

</sql:transaction>
```

Both operations belong to the same transaction.

Simple idea

```
Transaction
|
+-- Operation 1
|
+-- Operation 2
|
+-- Success → COMMIT
|
+-- Failure → ROLLBACK
```

This is useful for operations such as **money transfer**, where both updates should succeed together.

14. Complete JSTL SQL Flow

Students can remember the basic flow as:

```
JSP Page
↓
<sql:setDataSource>
↓
Database Connection
↓
<sql:query> / <sql:update>
↓
<sql:param>
↓
Database
↓
Result
```

↓
<c:forEach>
↓
Display

15. JSTL SQL vs JDBC

JDBC	JSTL SQL
Java API	JSP/JSTL tags
Java code required	Less Java code
Connection handled using Java	<sql:setDataSource>
PreparedStatement	<sql:param>
ResultSet	result.rows
executeQuery()	<sql:query>
executeUpdate()	<sql:update>
Better for application development	Mainly useful for JSP learning/simple applications
Important point	

JSTL SQL **does not** replace JDBC.

Actually, JSTL SQL tags internally depend on JDBC/database connectivity.

16. Important Tags to Remember for Examination

SQL Tag Library

```
<%@ taglib prefix="sql"
      uri="http://java.sun.com/jsp/jstl/sql" %>
```

Most important tags

```
<sql:setDataSource>
<sql:query>
<sql:update>
<sql:param>
<sql:dateParam>
<sql:transaction>
```

Easy memory trick

D-Q-U-P-T

- D → DataSource
- Q → Query
- U → Update
- P → Param
- T → Transaction

17. Advantages

- Simple database operations

- Reduces Java code in JSP
- Easy to learn
- Supports parameterized queries
- Supports transactions
- Integrates with other JSTL tags such as `<c:forEach>`

18. Limitations

- Mixes database logic with presentation
- Not recommended for large applications
- Difficult to maintain complex applications
- Less flexible than DAO/JDBC-based architecture
- JSP should generally be used for presentation rather than database processing

Best practice

JSP
↓
Servlet / Controller
↓
Service
↓
DAO
↓
JDBC
↓
Database

rather than:

JSP
↓
JSTL SQL
↓
Database

For your Advanced Java students, I would teach JSTL SQL after JDBC and before/alongside the MVC/DAO concept: it makes the difference between “database access in JSP” and “proper application architecture” very clear.