

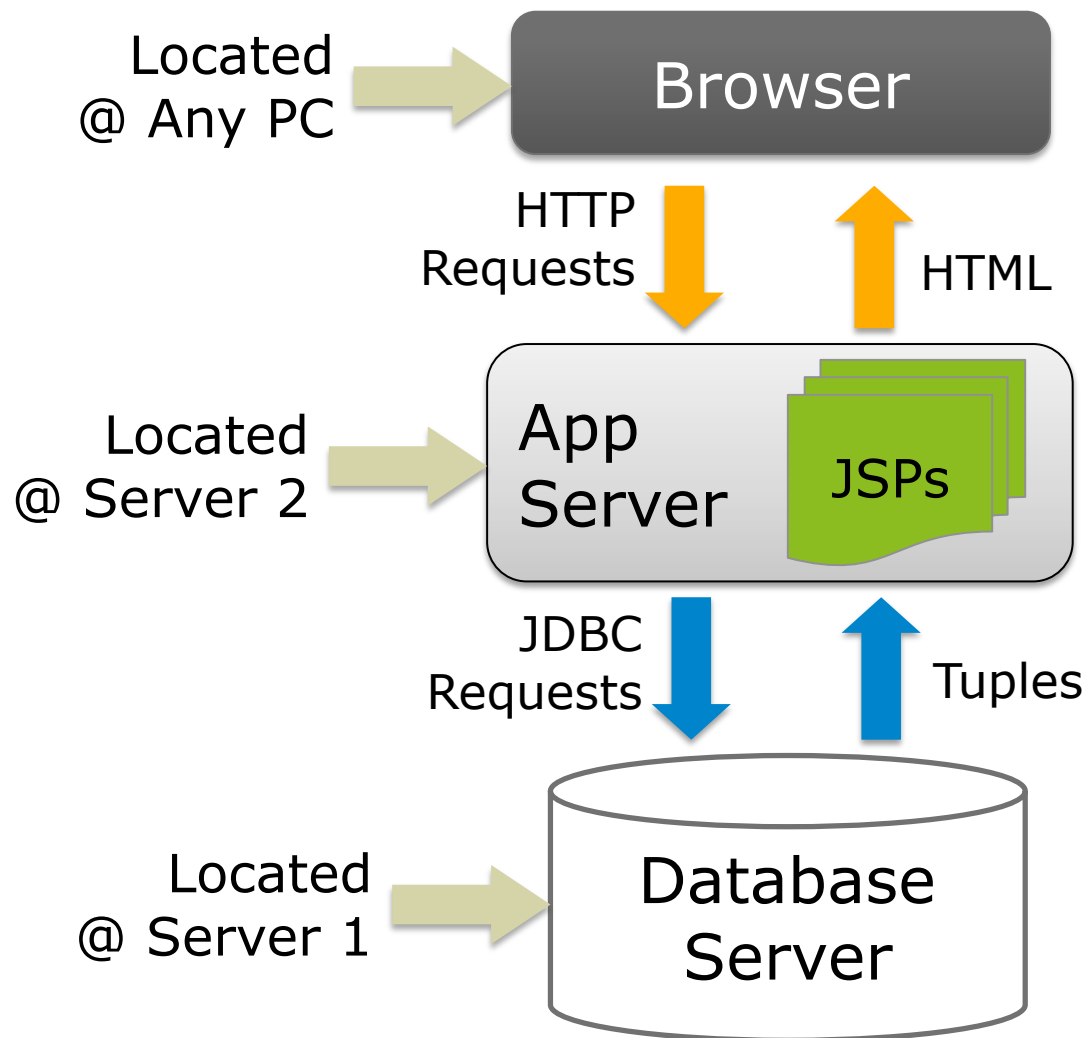
CSE 510

Web Data Engineering

**Application Server / Database Server
Communication**

cse@buffalo

Three-Tier Architecture



Data Entry Form

http://localhost:8080/sql-jsp/students.jsp

http://localhost:8080/sql-jsp/students.jsp Google

Saved Tabs Apple Yahoo! Google Maps YouTube Wikipedia News (743) Popular

Data Entry Menu	UBID	First Name	Middle Name	Last Name	Action
<ul style="list-style-type: none">StudentsClassesEnrollment	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Insert"/>
	77777777	Demian	<input type="text"/>	Lessa	<input type="button" value="Update"/> <input type="button" value="Delete"/>
	88888888	Michalis	A	Petropoulos	<input type="button" value="Update"/> <input type="button" value="Delete"/>

JDBC

```
import java.sql.*;

class JdbcTest {
    public static void main (String args [])
        throws SQLException {

        // Registering MySQL JDBC driver
        Class.forName ("com.mysql.jdbc.Driver") ;

        // Open a connection to the database
        Connection conn = DriverManager.getConnection (
            "jdbc:mysql://localhost/DemoClasses?" +
            "user=root&password=root") ;
    }
}
```

JDBC (cont'd)

```
// Query the student UB IDs
Statement stmt = conn.createStatement();
ResultSet rset =
    stmt.executeQuery("SELECT ubid FROM Students");
// Print out the UB ID (1st attribute)
while (rset.next ())
    System.out.println (rset.getInt(1));

//close the result set, statement, and connection
rset.close();
stmt.close();
conn.close();
}
}
```

PreparedStatement Object

- If you want to execute a `Statement` object many times, it will normally reduce execution time to use a `PreparedStatement` object instead:

```
PreparedStatement pstmt = conn.prepareStatement(  
    "UPDATE Students SET firstName = ?" +  
    "WHERE ubid = ?");
```

```
pstmt.setString(1, "Michalis");
```

```
pstmt.setInt(2, 88888888);
```

```
pstmt.executeUpdate();
```

PreparedStatement Object (cont'd)

- The following two code fragments accomplish the same thing:

```
String updateStr =  
    "UPDATE Students SET firstName = 'Michalis'" +  
    "WHERE ubid = 88888888";  
stmt.executeUpdate(updateStr);
```

```
PreparedStatement pstmt = conn.prepareStatement(  
    "UPDATE Students SET firstName = ?" +  
    "WHERE ubid = ?");  
pstmt.setString(1, "Michalis");  
pstmt.setInt(2, 88888888);  
updateSales.executeUpdate();
```

ResultSet Object

- `int getInt(int columnIndex)`
Retrieves the value of the designated column in the current row of a `ResultSet` object as an `int` Java type
- `int getInt(String columnName)`
- `String getString(int columnIndex)`
- `String getString(String columnName)`

Using Transactions

- When a connection is created, it is in `AutoCommit` mode, that is, each individual SQL statement is treated as a transaction and will be automatically committed right after it is executed

```
conn.setAutoCommit(false);
```

```
...
```

```
<transaction>
```

```
...
```

```
conn.commit();
```

```
conn.setAutoCommit(true);
```

Using Transactions – Example

```
conn.setAutoCommit(false);  
PreparedStatement pstmt = conn.prepareStatement(  
    "UPDATE Students SET firstName = ? WHERE ubid = ?");  
pstmt.setString(1, "Michalis");  
pstmt.setInt(2, 88888888);  
pstmt.executeUpdate();  
pstmt.setString(1, "Demian");  
pstmt.setInt(2, 77777777);  
pstmt.executeUpdate();  
conn.commit();  
conn.setAutoCommit(true);
```

Retrieving Exceptions

- JDBC lets you catch the exceptions generated by your DBMS

```
try {  
    // Code that could generate a SQLException  
} catch (SQLException sqle) {  
    System.out.println("SQLException: " + sqle.getMessage());  
    System.out.println("SQLState: " + sqle.getSQLState());  
    System.out.println("VendorError: " + sqle.getErrorCode());  
}
```

Retrieving Exceptions (cont'd)

- Release resources even if an exception is thrown

```
try{
    ...
} catch (SQLException sqle) {
    ...
} finally {
    // release resources in reverse-order of creation
    if (rs != null) {
        try { rs.close(); } catch (SQLException e) {} // ignore
        rs = null;
    }
    if (pstmt != null) {...}
    if (conn != null) {...}
}
```

Data Entry Form

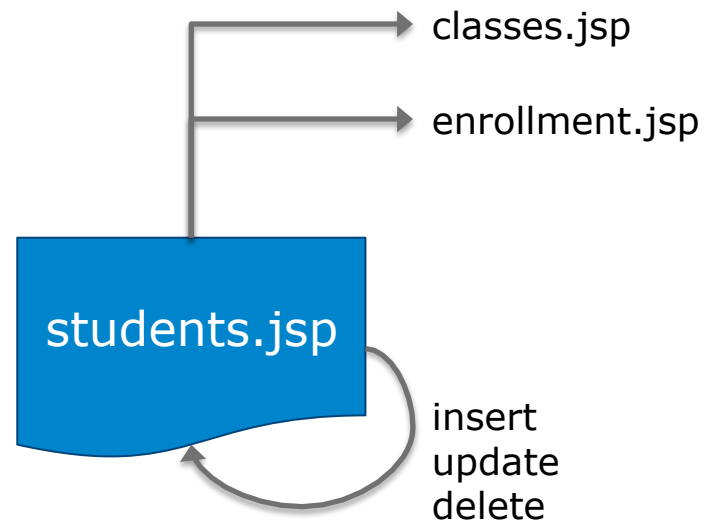
http://localhost:8080/sql-jsp/students.jsp

http://localhost:8080/sql-jsp/students.jsp Google

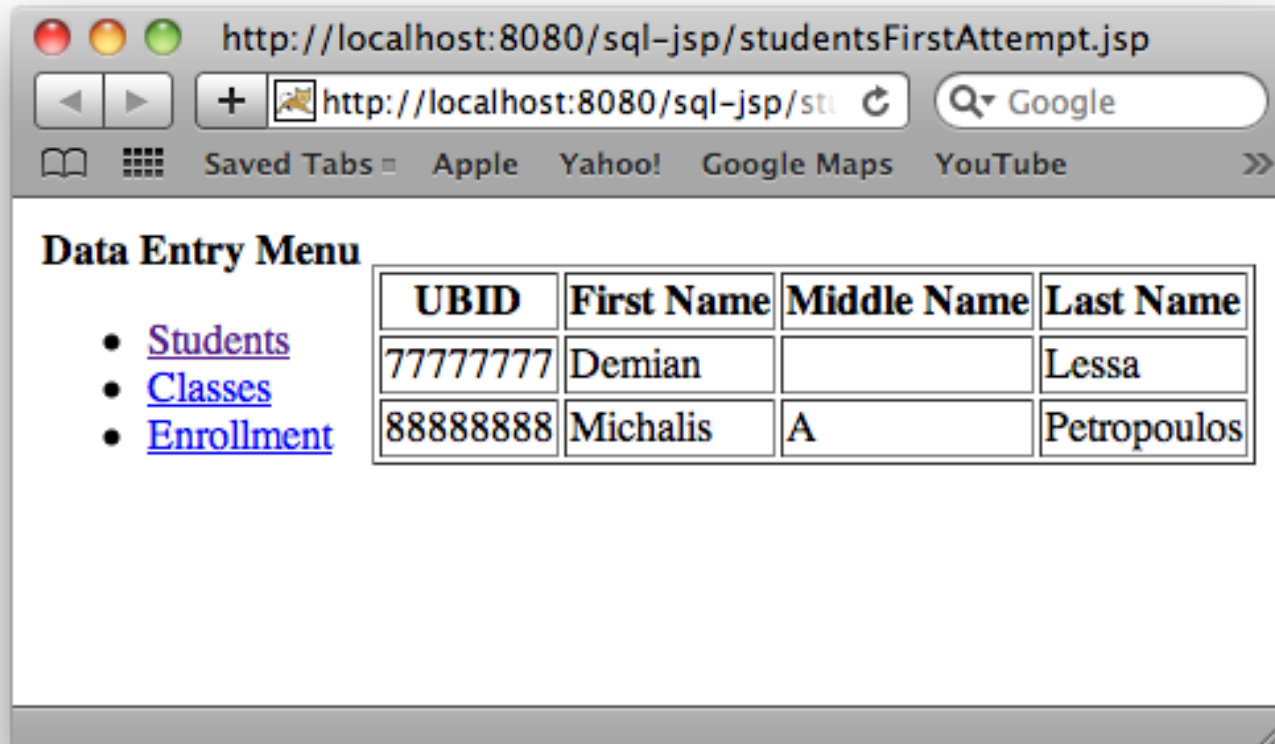
Saved Tabs Apple Yahoo! Google Maps YouTube Wikipedia News (743) Popular

Data Entry Menu	UBID	First Name	Middle Name	Last Name	Action
<ul style="list-style-type: none">StudentsClassesEnrollment	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Insert"/>
	77777777	Demian	<input type="text"/>	Lessa	<input type="button" value="Update"/> <input type="button" value="Delete"/>
	88888888	Michalis	A	Petropoulos	<input type="button" value="Update"/> <input type="button" value="Delete"/>

Data Entry Form



Data Entry Form - 1st Attempt



The screenshot shows a web browser window with the address bar displaying `http://localhost:8080/sql-jsp/studentsFirstAttempt.jsp`. The browser's address bar also shows a search engine (Google) and a list of saved tabs including Apple, Yahoo!, Google Maps, and YouTube. The main content area of the browser displays a "Data Entry Menu" with a list of links: [Students](#), [Classes](#), and [Enrollment](#). To the right of the links is a table with the following data:

UBID	First Name	Middle Name	Last Name
77777777	Demian		Lessa
88888888	Michalis	A	Petropoulos

Data Entry Form - 1st Attempt

Menu HTML Code

```
<b>Data Entry Menu</b>
<ul>
  <li><a href="students.jsp">Students<a></li>
  <li><a href="classes.jsp">Classes<a></li>
  <li><a href="enrollment.jsp">Enrollment<a></li>
</ul>
```


Data Entry Form - 1st Attempt

JSP Code

```
<html>
<body><table><tr>
  <td><jsp:include page="menu.html" /></td>
  <td>
    <Open Connection Code>
    <Statement Code>
    <Presentation Code>
    <Close Connection Code>
  </td>
</tr></table></body>
</html>
```

Data Entry Form - 1st Attempt

Open Connection Code

```
<%-- Import the java.sql package --%>  
<%@ page import="java.sql.*" %>  
<%  
    <variable declarations and initializations>  
    try {  
        // Registering MySQL JDBC driver  
        Class.forName("com.mysql.jdbc.Driver");  
        // Open a connection to the database  
        conn = DriverManager.getConnection(  
            "jdbc:mysql://localhost/DemoClasses?" +  
            "user=root&password=root");  
  
    %>
```

Data Entry Form - 1st Attempt

Statement Code

```
<%  
    // Create the statement  
    stmt = conn.createStatement();  
  
    // Use the statement to SELECT the students  
    // FROM the Students table.  
    rs = stmt.executeQuery("SELECT * FROM Students");  
%>
```

Data Entry Form - 1st Attempt

Presentation Code

```
<table>
  <tr>
    <th>UBID</th>
    <th>First Name</th>
    <th>Middle Name</th>
    <th>Last Name</th>
  </tr>
  <!-- Iterate over the ResultSet --%>
  <% while ( rs.next() ) { %>
    <Iteration Code>
  <% } %>
</table>
```

Data Entry Form - 1st Attempt

Iteration Code

```
<tr>
  <%-- Get the ubid, which is a number --%>
  <td><%=rs.getInt("ubid") %></td>

  <%-- Get the first name --%>
  <td><%=rs.getString("firstName") %></td>

  <%-- Get the middle name --%>
  <td><%=rs.getString("middleName") %></td>

  <%-- Get the last name --%>
  <td><%=rs.getString("lastName") %></td>
</tr>
```

Data Entry Form - 1st Attempt

Close Connection Code

```
<%  
    // Close the ResultSet  
    rs.close();  
  
    // Close the Statement  
    statement.close();  
  
    // Close the Connection  
    conn.close();  
  
} catch (SQLException sqle) { <Exception handling> }  
%>
```

Data Entry Form - 2nd Attempt

http://localhost:8080/sql-jsp/studentsSecondAttempt.jsp

http://localhost:8080/sql-jsp/studentsSecondAttempt.jsp Google

Saved Tabs Apple Yahoo! Google Maps YouTube Wikipedia News (759) Popular

Data Entry Menu

- [Students](#)
- [Classes](#)
- [Enrollment](#)

	UBID	First Name	Middle Name	Last Name	Action
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Insert"/>
	77777777	Demian		Lessa	
	88888888	Michalis	A	Petropoulos	

Data Entry Form - 2nd Attempt

JSP Code

```
<html>
<body><table><tr>
  <td><jsp:include page="menu.html"/></td>
  <td>
    <Open Connection Code>
    <Insertion Code>
    <Statement Code>
    <Presentation Code>
    <Close Connection Code>
  </td>
</tr></table></body>
</html>
```


Data Entry Form - 2nd Attempt

Insertion Code

```
// Check if an insertion is requested
String action = request.getParameter("action");
if (action != null && action.equals("insert")) {
    ...
    // Create the prepared statement to INSERT student values
    pstmt = conn.prepareStatement(
        "INSERT INTO Students VALUES (?, ?, ?, ?)");

    pstmt.setInt(1, Integer.parseInt(request.getParameter("id")));
    ...
    int rowCount = pstmt.executeUpdate();
    ...
}
```

Data Entry Form - 2nd Attempt

Presentation Code

```
<table>
  <tr>
    <th>UBID</th>
    <th>First Name</th>
    <th>Middle Name</th>
    <th>Last Name</th>
  </tr>
  <Insert Form Code>
  <%-- Iterate over the ResultSet --%>
  <% while ( rs.next() ) { %>
    <Iteration Code>
  <% } %>
</table>
```

Data Entry Form - 2nd Attempt

Insert Form Code

```
<tr>
  <form action="students.jsp" method="GET">
    <input type="hidden" name="action" value="insert"/>
    <th><input value="" name="id" size="10"/></th>
    <th><input value="" name="first" size="15"/></th>
    <th><input value="" name="middle" size="15"/></th>
    <th><input value="" name="last" size="15"/></th>
    <th><input type="submit" value="Insert"/></th>
  </form>
</tr>
```

Data Entry Form - 3rd Attempt

http://localhost:8080/sql-jsp/students.jsp

http://localhost:8080/sql-jsp/students.jsp Google

Saved Tabs Apple Yahoo! Google Maps YouTube Wikipedia News (743) Popular

Data Entry Menu	UBID	First Name	Middle Name	Last Name	Action
<ul style="list-style-type: none">StudentsClassesEnrollment	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Insert"/>
	77777777	Demian	<input type="text"/>	Lessa	<input type="button" value="Update"/> <input type="button" value="Delete"/>
	88888888	Michalis	A	Petropoulos	<input type="button" value="Update"/> <input type="button" value="Delete"/>

Data Entry Form - 3rd Attempt

JSP Code

```
<html><body><table><tr>
  <td><jsp:include page="menu.html"/></td>
  <td>
    <Open Connection Code>
    <Insertion Code>
    <Update Code>
    <Delete Code>
    <Statement Code>
    <Presentation Code>
    <Close Connection Code>
  </td>
</tr></table></body></html>
```

Data Entry Form - 3rd Attempt

Update Code

```
// Check if an update is requested
if (action != null && action.equals("update")) {
    ...
    // Create the prepared statement to UPDATE student values
    pstmt = conn.prepareStatement(
        "UPDATE Students SET firstName = ?, " +
        "middleName = ?, lastName = ? WHERE ubid = ?");

    pstmt.setString(1, request.getParameter("first"));
    ...
    int rowCount = pstmt.executeUpdate();
    ...
}
```

Data Entry Form - 3rd Attempt

Delete Code

```
// Check if a delete is requested
if (action != null && action.equals("delete")) {
    ...
    // Create the prepared statement to DELETE students
    pstmt = conn.prepareStatement(
        "DELETE FROM Students WHERE ubid = ?");

    pstmt.setInt(1, Integer.parseInt(request.getParameter("id")));
    int rowCount = pstmt.executeUpdate();
    ...
}
```

Data Entry Form - 3rd Attempt

Presentation Code

```
<table>
  <tr>
    <th>UBID</th>
    <th>First Name</th>
    <th>Middle Name</th>
    <th>Last Name</th>
  </tr>
  <Insert Form Code>
  <%-- Iterate over the ResultSet --%>
  <% while ( rs.next() ) { %>
    <Iteration Code>
  <% } %>
</table>
```


Data Entry Form - 3rd Attempt

Iteration Code

```
<tr>
<form action="students.jsp" method="GET">
  <input type="hidden" name="action" value="update"/>
  <input type="hidden" name="id" value="<%=rs.getInt("ubid")%>"/>
  ...
  <!-- Get the first name -->
  <td><input value="<%=rs.getString("firstName")%>" name="first"/></td>
  ...
  <td><input type="submit" value="Update"></td>
</form>
<form action="students.jsp" method="GET">
  <input type="hidden" name="action" value="delete"/>
  <input type="hidden" value="<%=rs.getInt("ubid")%>" name="id"/>
  <td><input type="submit" value="Delete"/></td>
</form>
</tr>
```