CSE 135

Improvements on how to access the database:
Connection Pools

Better Alternative: Connection Pool

- Application server creates a resource that is a pool of connections to a DBMS
- So that each web app process does not have to open and close a connection

Developer specifies:
- Pool size
- Minimum number of open connections
  - Even if nobody asked them yet
- Minimum number of connections that will not close
- Timeouts

Handling Database Connections

Within a JSP:
- Opening a connection for every HTTP request penalizes the DB server (in terms of resources) and the client (in terms of waiting time)

The Connection Pool as a Proxy to the database

Data Entry Form - 4th 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>
```
Data Entry Form - 4th Attempt

<%@ page import="java.sql.*, javax.sql.*, javax.naming.*"%>

<%-- Open Connection Code --%>
<% Connection conn = null; try {
    // Obtain the environment naming context
    Context initCtx = new InitialContext();
    // Look up the data source
    DataSource ds = (DataSource) initCtx.lookup("java:comp/env/jdbc/ClassesDBPool");
    // Allocate and use a connection from the pool
    conn = ds.getConnection();
}%>

DB Specific Parameters

- Databases close connections after many hours of inactivity – fix this

<?xml version="1.0" encoding="UTF-8"?>
<Context path="" debug="5" override="true" reloadable="true">
    <Resource
        url="jdbc:mysql://localhost/DemoClasses?autoReconnectForPools=true" username="root" password="root"...
    />
</Context>