CS320 Web and Internet Programming
Database Access with JSTL SQL

Chengyu Sun
California State University, Los Angeles

JSTL SQL

- sql:transaction
- sql:query
- sql:update
- sql:param
- sql:dateParam
- sql:setDataSource

http://download.oracle.com/docs/cd/E17802_01/products/products/jsp/jstl/1.1/docs/tlddocs/index.html

Example: HelloSQL.jsp

- Import the taglib
- Data source
- Query
- Results display

sql:setDataSource

- `var` – data source name. Only needed when you have multiple db sources.
- `scope` – scope of the data source
- `driver` – "com.mysql.jdbc.Driver"
- `url`
- `user`
- `password`
- `dataSource`

sql:query

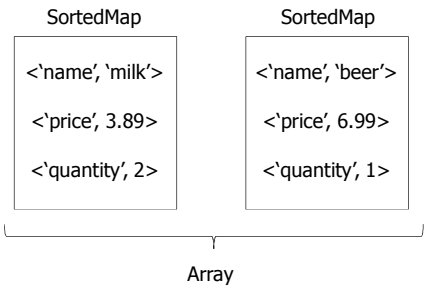
- `var` – name of the result set
- `scope` – scope of the result set
- `sql` – query statement
- `dateSource` – name of the data source
- `startRow`
- `maxRows` – max number of rows in the result set

sql:query Result Set

- javax.servlet.jsp.jstl.sql.Result
  - SortedMap[] getRows()
  - Object[][] getRowsByIndex()
  - String[] getColumnNames()
  - int getRowCount()
  - boolean isLimitedByMaxRows()

http://docs.oracle.com/cd/E17802_01/products/products/jsp/jstl/1.1/docs/api/javax/servlet/jsp/jstl/sql/Result.html

An Array of SortedMap



sql:query example 1

```
<sql:query var="results" sql="select * from items"/>

<table>
 <c:forEach items="${results.rows}" var="row">
  <c:forEach items="${row}" var="col">
    <tr>
      <td>${col.key}</td><td>${col.value}</td>
    </tr>
  </c:forEach>
 </c:forEach>
</table>
```

sql:query example 2

```
<sql:query var="results">
 select * from items where price > 2.00
</sql:query>

<table>
 <c:forEach items="${results.rowsByIndex}" var="row">
 <tr>
   <c:forEach items="${row}" var="col">
      <td>${col}</td>
   </c:forEach>
 </tr>
 </c:forEach>
</table>
```

Use of <sql:param>

- Similar to `PreparedStatement` in JDBC

```
<sql:query var="results">

   select * from items where
      price < ? and quantity > ?

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

</sql:query>
```

Example: GuestBook (JSTL SQL)

- GuestBook.jsp
- AddComment.jsp

sql:update

- `var` – name of the result variable. int
  - number of rows affected by the update
  - 0 if the update statement doesn't return anything
- `scope`
- `sql`
- `dateSource` – name of the data source

sql:update example

```
<c:if test="${! empty param.setPrice}">

   <sql:update var="r">
      update items set price = ? where name = ?
      <sql:param value="${param.price}"/>
      <sql:param value="${param.name}"/>
   </sql:update>

</c:if>
```

JSTL SQL vs. JDBC

- JSTL SQL
  - Simple applications
  - Small relations
  - Straight-forward operations

⇩

Model 1

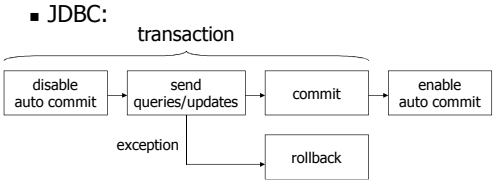
- JDBC
  - Everything else

⇩

MVC

Beyond Basics ...

- ACID
- Transaction
  - <sql:transaction>
  - JDBC:



... Beyond Basics ...

- It's rather expensive to open a db connection
  - So how about once we open a connection, we leave it open forever??
- Connection pooling
  - Max number of connections
  - Max number of idle connections
  - Abandoned connection timeout
  - http://tomcat.apache.org/tomcat-7.0-doc/jndi-datasource-examples-howto.html

... Beyond Basics

- Mismatch between an OO design and a relational design
- Object-relational mapping
  - hibernate - http://www.hibernate.org/