

```

1 10. JSTL sql
2 1) SQL 관련 기능을 제공해 주는 JSTL 라이브러리
3 2) 데이터베이스 서버에 접근, 쿼리 전송, 트랜잭션 처리 가능
4 3) <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
5 4) <sql:setDataSource />
6   - 데이터베이스에 접근
7   - <sql:setDataSource var="variableName" scope="{page | request | session | application}"
8     {dataSource="dataSource" | driver="driver" url="url" user="user" password="password"} />
9   - 접근방법 2가지
10  a. dataSource
11     <sql:setDataSource dataSource="jdbc/myoracle" var="conn" />
12     dataSource 가 JNDI 로 설정되었으면 바로 <sql:query> 사용 가능
13  b. driver, url, driver, password
14     <sql:setDataSource driver="oracle.jdbc.driver.OracleDriver"
15       url="jdbc:oracle:thin:@localhost:1521:XE" user="scott" password="tiger" />
16   - Connection 정보는 var 속성에 입력된 변수에 저장
17
18 [server.xml]
19 <Resource auth="Container" driverClassName="oracle.jdbc.driver.OracleDriver"
20   maxActive="10" maxIdle="10" maxWait="1000" name="jdbc/myoracle"
21   password="tiger" type="javax.sql.DataSource"
22   url="jdbc:oracle:thin:@localhost:1521:XE" username="scott"/>
23
24 [web.xml]
25 <resource-ref>
26   <description>JNDI Test</description>
27   <res-ref-name>jdbc/myoracle</res-ref-name>
28   <res-type>javax.sql.DataSource</res-type>
29   <res-auth>Container</res-auth>
30 </resource-ref>
31
32 [sqldemo.jsp]
33 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
34 <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
35 <body>
36   <c:catch var="error">
37     <%-- 1st 방법 --%>
38     <%-- <sql:setDataSource driver="com.mysql.jdbc.Driver"
39       url="jdbc:mysql://localhost:3306/world"
40       user="root" password="javamysql" var="conn"/> --%>
41     <%-- 2nd 방법 --%>
42     <sql:setDataSource dataSource="jdbc/myoracle" var="conn" />
43     <c:out value="${conn}" />
44   </c:catch>
45   <c:if test="${!empty error}">
46     Error Message : <c:out value="${error}" />
47   </c:if>
48 </body>
49
50 5) <sql:query>
51   - 데이터베이스 서버에 쿼리를 전송할 때 사용
52   - <sql:query var="변수명" scope="범위" sql="sql" dataSource="dataSource" startRow="startRow"
53     maxRows="maxRows" />
54   - dataSource 속성은 JNDI 리소스나 <sql:setDataSource>로 얻어온 변수명으로 설정
55   - sql 속성은 실행할 SQL 문 지정 또는 바디에서 설정 가능
56   - var 속성은 sql 쿼리가 실행된 결과 저장
57   - startRow 는 얻어진 쿼리 결과의 시작 행 값을 의미, 1번째 레코드 값부터 시작하려면 0으로 설정
58   - maxRow 속성은 얻어온 쿼리 결과의 레코드 최대 수를 의미
59   - Result interface
60     See javax.servlet.jsp.jstl.sql.Result interface

```

```

61 *****Result interface*****
62 This interface represents the result of a <sql:query> action.
63 It provides access to the following information in the query result:
64
65 - The result rows (getRows() and getRowsByIndex())
66 - The column names (getColumnNames())
67 - The number of rows in the result (getRowCount())
68 - An indication whether the rows returned represent the complete result or
69 just a subset that is limited by a maximum row setting (isLimitedByMaxRows())
70
71 - Methods
72 java.lang.String [] getColumnNames()
73 int getRowCount()
74 java.util.SortedMap[] getRows()
75 java.lang.Object [][] getRowsByIndex()
76 boolean isLimitedByMaxRows()
77 *****
78
79 [sqldemo1.jsp]
80 <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
81 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
82 <!DOCTYPE html>
83 <html lang="ko">
84 <head>
85 <style>
86     table {
87         width:600px;border-collapse:collapse;
88         margin:auto;
89     }
90     th, td{
91         border:1px solid black;
92     }
93     td{
94         text-align:center;
95     }
96     caption{
97         font-size:1.5em;color:blue;
98     }
99 </style>
100 </head>
101 <body>
102 <sql:setDataSource var="conn" driver="oracle.jdbc.driver.OracleDriver"
103     url="jdbc:oracle:thin:@localhost:1521:xe"
104     user="scott" password="tiger" />
105 <sql:query var="rs" dataSource="${conn}">
106     SELECT empno, ename, job, hiredate, sal, comm FROM emp
107 </sql:query>
108 <table>
109     <caption>사원 정보</caption>
110     <thead>
111         <tr height="40">
112             <th width="100">사원번호</th><th width="100">사원명</th><th width="100">업무</th>
113             <th width="100">입사일</th><th width="100">봉급</th><th width="100">보너스</th>
114         </tr>
115     </thead>
116     <tbody>
117         <c:forEach var="data" items="${rs.rows}">
118             <tr>
119                 <td><c:out value="${data.empno}"/></td>
120                 <td><c:out value="${data.ename}"/></td>
121                 <td><c:out value="${data.job}"/></td>

```

```

122         <td><c:out value="\${data.hiredate}"/></td>
123         <td><c:out value="\${data.sal}"/></td>
124         <td><c:out value="\${data.comm}"/></td>
125     </tr>
126 </c:forEach>
127 </tbody>
128 </table>
129 </body>
130 </html>
131
132 [sqldemo2.jsp]
133 <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
134 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
135 <!DOCTYPE html>
136 <html lang="ko">
137 <head>
138     <style>
139         table {
140             width:600px;border-collapse:collapse;
141             margin:auto;
142         }
143         th, td{
144             border:1px solid black;width:200px;
145         }
146         td{
147             text-align:center;
148         }
149         caption{
150             font-size:1.5em;color:blue;
151         }
152     </style>
153 </head>
154 <body>
155 <sql:query var="rs" dataSource="jdbc/myoracle">
156     SELECT deptno, dname, loc FROM dept ORDER BY deptno DESC
157 </sql:query>
158 <table>
159     <caption>부서 정보</caption>
160     <thead>
161         <tr height="40">
162             <c:forEach var="columnName" items="\${rs.columnNames}">
163                 <th><c:out value="\${columnName}" /></th>
164             </c:forEach>
165         </tr>
166     </thead>
167     <tbody>
168         <c:forEach var="row" items="\${rs.rowsByIndex}">
169             <tr height="50">
170                 <c:forEach var="column" items="\${row}" varStatus="i">
171                     <td>
172                         <c:if test="\${column != null}">
173                             <c:out value="\${column}" />
174                         </c:if>
175                         <c:if test="\${column == null}">
176                             &nbsp;
177                         </c:if>
178                     </td>
179                 </c:forEach>
180             </tr>
181         </c:forEach>
182     </tbody>

```

```
183 </table>
184 </body>
185 </html>
186
187 6) <sql:update>
188   - 레코드의 추가, 수정, 삭제 기능 담당
189   - <sql:update var="변수명" scope="범위" sql="sql" dataSource="dataSource" />
190   - var 속성에 입력한 변수에는 레코드를 업데이트한 결과가 저장
191   - sql 속성에는 레코드를 업데이트할 SQL 문
192   - dataSource 속성은 JNDI 리소스나 <sql:setDataSource>로 얻어온 변수명으로 설정
193
194 [sqldemo3.jsp]
195 <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
196 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
197 <!DOCTYPE html>
198 <html lang="ko">
199 <body>
200 <sql:update dataSource="jdbc/myoracle" var="result">
201   INSERT INTO dept VALUES(50, 'MARKETING', 'SEOUL')
202 </sql:update>
203 <h2>Insert Success</h2>
204 <c:out value="${result}" />
205 </body>
206 </html>
207
208 7) <sql:param>
209   - <sql:query> 태그나 <sql:update> 태그로 SQL 문장을 전송할 때 파라미터를 전달해 주는 태그
210   - <sql:param value="value" />
211   - 전달해야 할 파라미터가 두 가지 이상일 경우는 순차적으로 사용
212
213 [sqldemo4.jsp]
214 <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
215 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
216 <!DOCTYPE html>
217 <html lang="ko">
218 <body>
219 <sql:update dataSource="jdbc/myoracle" var="result">
220   INSERT INTO dept VALUES(50, 'MARKETING', 'SEOUL')
221   <sql:param value="60" />
222   <sql:param value="IT" />
223   <sql:param value="PUSAN" />
224 </sql:update>
225 <c:if test="${result == 1}">
226   <h2>Insert Success</h2>
227 </c:if>
228 </body>
229 </html>
230 [sqldemo5.jsp]
231 <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>
232 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
233 <!DOCTYPE html>
234 <html lang="ko">
235 <head>
236   <style>
237     table {
238       width:600px;border-collapse:collapse;
239       margin:auto;
240     }
241     th, td{
242       border:1px solid black;width:200px;
243     }
```

```

244     td{
245         text-align:center;
246     }
247     caption{
248         font-size:1.5em;color:blue;
249     }
250 </style>
251 </head>
252 <body>
253 <sql:query var="rs" dataSource="jdbc/mymysql">
254     SELECT zipcode, sido, gugun, dong, bunji FROM zipcode
255     WHERE dong LIKE CONCAT('%', ?, '%')
256     <sql:param value="{param.keyword}" />
257 </sql:query>
258 <table>
259     <caption>우편번호검색</caption>
260     <thead>
261         <tr height="40">
262             <c:forEach var="columnName" items="{rs.columnNames}">
263                 <th><c:out value="{columnName}" /></th>
264             </c:forEach>
265         </tr>
266     </thead>
267     <tbody>
268         <c:forEach var="row" items="{rs.rowsByIndex}">
269             <tr height="50">
270                 <c:forEach var="column" items="{rs.rowsByIndex}" varStatus="i">
271                     <td>
272                         <c:if test="{column != null}">
273                             <c:out value="{column}" />
274                         </c:if>
275                         <c:if test="{column == null}">
276                             &nbsp;
277                         </c:if>
278                     </td>
279                 </c:forEach>
280             </tr>
281         </c:forEach>
282     </tbody>
283 </table>
284 </body>
285 </html>
286
287 8) <sql:dateParam>
288     - <sql:param> 과 같이 파라미터를 전달할 때 사용
289     - <sql:dataParam value="value" type="type" />
290     - 차이점은 이 태그는 날짜 정보의 파라미터를 전달할 때 사용
291     - type 속성으로 data, time, timestamp 중 하나의 값 가능
292     - Statement 의 setDate(), setTime(), setTimestamp() 메소드와 같은 기능
293
294 9) <sql:transaction >
295     - 데이터베이스 작업 시 트랜잭션을 사용하도록 하는 태그
296     - <sql:transaction dataSource="dataSource" isolation="isolation" />
297     - dataSource 속성은 JNDI 리소스나 <sql:setDataSource>로 얻어온 변수명으로 설정
298     - isolation 속성을 이용하여 격리 수준을 지정가능
299     - 가능한 값은 read_committed, read_uncommitted, repeatable_read, serializable
300     - 이 태그의 바디 부분에 쿼리 태그를 사용하며, 실패할 경우는 트랜잭션 태그 내의
301       작업을 모두 반영하지 않기 때문에 트랜잭션 관리가 가능
302
303 [sqldemo6.jsp]
304 <%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %>

```

```
305 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
306 <!DOCTYPE html>
307 <html lang="ko">
308 <head>
309   <style>
310     table {
311       width:600px;border-collapse:collapse;
312       margin:auto;
313     }
314     th, td{
315       border:1px solid black;width:200px;
316     }
317     td{
318       text-align:center;
319     }
320     caption{
321       font-size:1.5em;color:blue;
322     }
323   </style>
324 </head>
325 <body>
326 <c:catch var="ex">
327   <sql:setDataSource var="conn" driver="oracle.jdbc.driver.OracleDriver"
328     url="jdbc:oracle:thin:@localhost:1521:xe"
329     user="scott" password="tiger" />
330   <sql:update dataSource="${conn}">
331     INSERT INTO dept VALUES(?,?,?)
332     <sql:param value="${param.deptno}" />
333     <sql:param value="${param.dname}" />
334     <sql:param value="${param.loc}" />
335   </sql:update>
336   <sql:query var="rs" dataSource="${conn}">
337     SELECT deptno, dname, loc FROM dept
338   </sql:query>
339   <table>
340     <caption>부서정보</caption>
341     <thead>
342       <tr height="40">
343         <th>부서번호</th><th>부서이름</th><th>위치</th>
344       </tr>
345     </thead>
346     <tbody>
347       <c:forEach var="row" items="${rs.rowsByIndex}">
348         <tr height="50">
349           <c:forEach var="column" items="${row}">
350             <td>
351               <c:if test="${column != null}">
352                 <c:out value="${column}" />
353               </c:if>
354               <c:if test="${column == null}">
355                 &nbsp;
356               </c:if>
357             </td>
358           </c:forEach>
359         </tr>
360       </c:forEach>
361     </tbody>
362   </table>
363 </c:catch>
364 <c:if test="${!empty(ex)}">
365   ERROR : <c:out value="${ex}" />
```

```

366 </c:if>
367 </body>
368 </html>
369
370 11. Function
371 1) 주로 문자열 처리
372 2) java.lang.String 클래스에 속하는 메소드와 비슷한 기능
373 3) <%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
374 4) substring(str, index1, index2)
375     str의 index1부터 index2 -1 까지의 부분자열을 리턴
376 5) substringAfter(str1, str2)
377     str1에서 str2를 찾아서 그 후의 부분자열을 리턴
378     ${fn:substringAfter("Stella Cadente", "e")}
379     --> lla Cadente
380 6) substringBefore(str1, str2)
381     str1에서 str2를 찾아서 그 전의 부분자열을 리턴
382     ${fn:substringBefore("Stella Cadente", "e")}
383     --> St
384 7) toUpperCase(str)
385 8) toLowerCase(str)
386 9) trim(str)
387 10) replace(str, src, dest)
388     str문자열에 포함된 src를 모두 dest로 치환한 결과를 리턴
389     ${fn:replace("Stella Cadente", "e", "E")}
390 11) indexOf(str1, str2)
391     str1에 포함된 str2의 시작 인덱스를 리턴
392     ${fn:indexOf("Stella Cadente", "Cadente")} --> 7
393 12) startsWith(str1, str2)
394     str1이 str2로 시작하면 true, 아니면 false
395 13) endsWith(str1, str2)
396     str1이 str2로 끝나면 true, 아니면 false
397     ${fn:endsWith("Hello World", "World")}
398 14) contains(str1, str2)
399     <c:set var="myString" value="Hello World" />
400     ${fn:contains(myString, "Hello")}
401     ${fn:contains("Stella Cadente", "Cadente")}
402 15) containsIgnoreCase(str1, str2)
403     ${fn:contains("Stella Cadente", "CADENTE")}
404 16) split(str1, str2)
405     str1을 str2를 기준으로 분리해서 만든 부분자열들의 배열을 리턴
406     <c:set var="split" value="${fn:split('my,world', ',')}"/>
407     <table>
408     <c:forEach var="substring" items="${split}">
409     <tr><td>${substring}</td></tr>
410     </table>
411
412     <table>
413     <tr><td>my</td></tr>
414     <tr><td>world</td></tr>
415     </table>
416 17) join(arr, str2)
417     arr배열의 모든 항목을 합쳐서 리턴. 항목사이에는 str2가 들어감
418 18) escapeXml(str)
419     HTML 문법에 의해 특수 문자로 취급되는 모든 문자표를 이스케이프 시퀀스로 치환한 결과를 리턴
420     ${fn:escapeXml("Use <br /> to change lines")}
421     --> Use &lt;br /&gt; to change lines
422 19) length(obj)
423     obj 가 문자열이면 문자열의 길이, List 나 Collection이면 항목의 수를 리턴
424     ${fn:length("Stella Cadente")} --> 14
425
426 [fndemo.jsp]

```

```
427 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
428 <%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
429 <html>
430 <head><title>함수 사용</title></head>
431 <body>
432 <c:set var="str1" value="Functions <태그>를 사용합니다. " />
433 <c:set var="str2" value="사용" />
434 <c:set var="tokens" value="1,2,3,4,5,6,7,8,9,10" />
435
436 length(str1) = "${fn:length(str1)}" <br>
437 toUpperCase(str1) = "${fn:toUpperCase(str1)}" <br>
438 toLowerCase(str1) = "${fn:toLowerCase(str1)}" <br>
439 substring(str1, 3, 6) = "${fn:substring(str1, 3, 6)}" <br>
440 substringAfter(str1, str2) = "${fn:substringAfter(str1, str2)}" <br>
441 substringBefore(str1, str2) = "${fn:substringBefore(str1, str2)}" <br>
442 trim(str1) = "${fn:trim(str1)}" <br>
443 replace(str1, src, dest) = "${fn:replace(str1, " ", "-")}" <br>
444 indexOf(str1, str2) = "${fn:indexOf(str1, str2)}" <br>
445 startsWith(str1, str2) = "${fn:startsWith(str1, 'Fun')}" <br>
446 endsWith(str1, str2) = "${fn:endsWith(str1, "합니다.")}" <br>
447 contains(str1, str2) = "${fn:contains(str1, str2)}" <br>
448 containsIgnoreCase(str1, str2) = "${fn:containsIgnoreCase(str1, str2)}" <br>
449
450 <c:set var="array" value="${fn:split(tokens, ',')}" />
451
452 join(array, "-") = "${fn:join(array, '-')}" <br>
453 escapeXml(str1) = "${fn:escapeXml(str1)}" <br>
454
455 </body>
```