

JSP 웹 프로그래밍

JDBC로 데이터베이스와 JSP 연동

상품 조회, 등록, 수정, 삭제하기

JDBC(Java DataBase Connectivity)

- ▶ 자바/JSP 프로그램 내에서 데이터베이스와 관련된 작업을 처리할 수 있도록 도와주는 자바 표준 인터페이스
- ▶ 관계형 데이터베이스 시스템에 접근하여 SQL 문을 실행하기 위한 자바 API 또는 자바 라이브러리
- ▶ JDBC API를 사용하면 DBMS의 종류에 상관없이 데이터베이스 작업 처리가능



JDBC의 개요

▶ JDBC를 사용한 JSP와 데이터베이스의 연동 단계

- ① `java.sql.*` 패키지 импорт.
- ② JDBC 드라이버 로딩.
- ③ 데이터베이스 접속을 위한 `Connection` 객체 생성.
- ④ 쿼리문을 실행하기 위한 `Statement` / `PreparedStatement` / `CallableStatement` 객체 생성.
- ⑤ 쿼리 실행.
- ⑥ 쿼리 실행의 결과 값(`int`, `ResultSet`) 사용.
- ⑦ 사용된 객체(`ResultSet`, `Statement`, `PreparedStatement`, `CallableStatement`, `Connection`) 종료



JDBC 드라이버 로딩 및 DBMS 접속

▶ JDBC 드라이버 로딩하기

- ▶ 드라이버 인터페이스를 구현하는 작업
- ▶ `Class.forName()` 메소드를 이용하여 JDBC 드라이버를 로딩

```
Class.forName(String className);
```

- ▶ JDBC 드라이버가 로딩되면 자동으로 객체가 생성되고 `DriverManager` 클래스에 등록
[MySQL 드라이버 로딩 예]

```
<%  
    try{  
        Class.forName("com.mysql.jdbc.Driver");  
    }catch(SQLException ex){  
        //예외 발생 처리  
    }  
%>
```

- ▶ JDBC 드라이버 로딩은 프로그램 수행 시 한 번만 필요



JDBC 드라이버 로딩 및 DBMS 접속

▶ Connection 객체 생성하기

- ▶ DriverManager 클래스의 getConnection() 메소드를 사용해 JDBC 드라이버에서 데이터베이스와 연결된 커넥션 가져오기
- ▶ DriverManager 클래스로 Connection 객체를 생성할 때 JDBC 드라이버를 검색해 Connection 객체를 생성한 후 반환

```
static Connection getConnection(String url)
static Connection getConnection(String url, String user, String password)
static Connection getConnection(String url, Properties info)
```

- ▶ [Connection 객체 생성 예: getConnection(String url, String user, String password) 메소드 사용]

```

Connection conn = null;
try {
    Class.forName("com.mysql.jdbc.Driver");
    conn = DriverManager.getConnection(
        "jdbc:mysql://localhost:3306/JSPBookDB", "root", "1234");
} catch (SQLException ex) {
    //예외 발생 처리
}

```

JDBC 드라이버 로딩 및 DBMS 접속

▶ 데이터베이스 연결 닫기

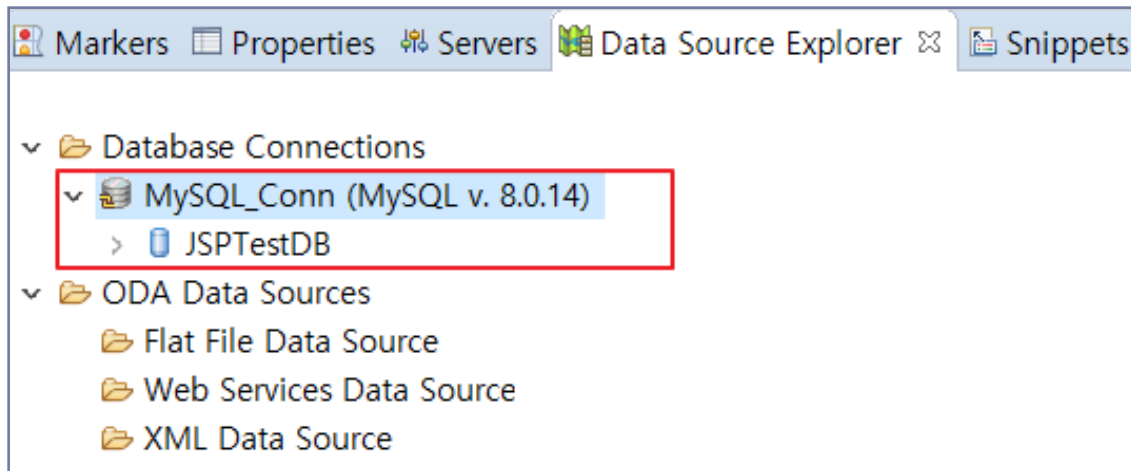
- ▶ 데이터베이스 연결이 더 이상 필요하지 않으면 `close()` 메소드로 `Connection` 객체를 해제
- ▶ 데이터베이스 리소스를 사용하지 않기 위해 사용을 끝내자마자 리소스를 해제

```
void close() throws SQLException
```

```
<%  
    Connection conn = null;  
    try{  
        //JDBC 드라이버 로딩  
        //Connection 객체 생성  
    } catch(SQLException e){  
        //예외 발생 처리  
    } finally{  
        if (conn != null) conn.close();  
    }  
%>
```

JDBC 드라이버 로딩 및 DBMS 접속

▶ JDBC API로 데이터베이스 접속하기

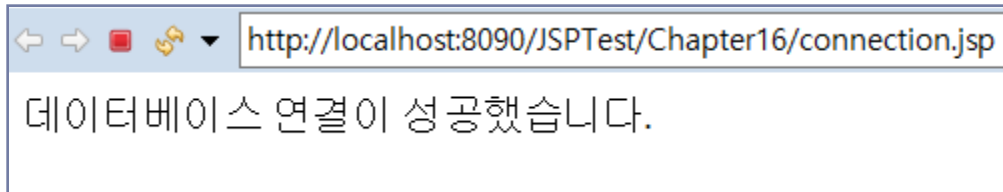


JDBC 드라이버 로딩 및 DBMS 접속

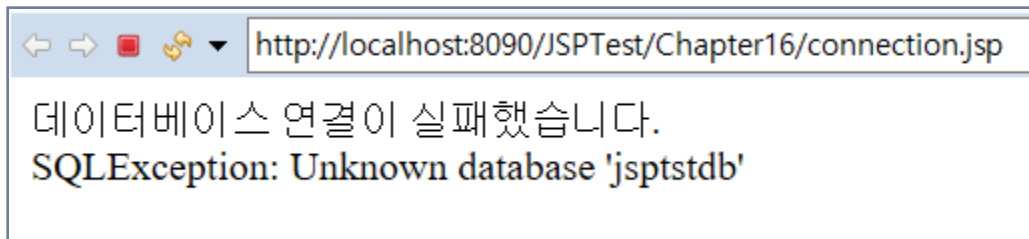
```
connection.jsp
1 <%@ page contentType="text/html; charset=UTF-8"%>
2 <%@ page import="java.sql.*" %>
3 <html>
4 <head>
5 <title>Database SQL</title>
6 </head>
7 <body>
8 <%
9     Connection conn = null;
10    try{
11        String url = "jdbc:mysql://localhost:3306/JSPTestDB";
12        String user = "root";
13        String password = "1234";
14
15        Class.forName("com.mysql.jdbc.Driver");
16        conn = DriverManager.getConnection(url, user, password);
17        out.println("데이터베이스 연결이 성공했습니다.");
18    }catch(SQLException ex){
19        out.println("데이터베이스 연결이 실패했습니다.<br>");
20        out.println("SQLException: " + ex.getMessage());
21    }finally{
22        if(conn != null)
23            conn.close();
24    }
25    %>
26 </body>
27 </html>
```


JDBC 드라이버 로딩 및 DBMS 접속

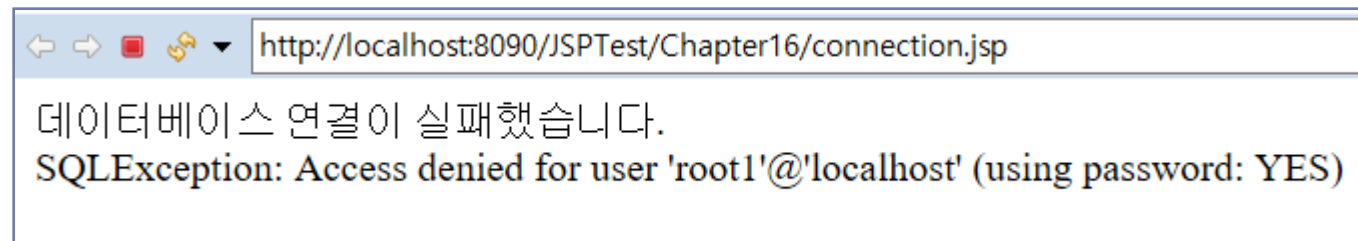
▶ [데이터베이스 연결 성공]



▶ [데이터베이스 이름이 잘못된 경우]



▶ [데이터베이스 사용자 이름이 잘못된 경우]



데이터베이스 쿼리 실행

▶ PreparedStatement객체로 INSERT쿼리문 실행하기

The screenshot shows an IDE window with a SQL editor and a results pane. The SQL editor contains the following code:

```
1 CREATE TABLE IF NOT EXISTS member(  
2     id VARCHAR(20) NOT NULL,  
3     passwd VARCHAR(20),  
4     name VARCHAR(30),  
5     PRIMARY KEY(id)  
6 );  
7  
8 select * from member;
```

The results pane shows the execution status and results for the SQL statements. The first statement, 'CREATE TABLE IF NOT EXISTS member', was executed successfully. The second statement, 'select * from member', was also executed successfully, and the results are displayed in a table with columns 'id', 'passwd', and 'name'.

Status	Operation	Result1
✓	Su CREATE TABLE IF NOT EXISTS	
✓	Su select * from member	

데이터베이스 쿼리 실행

▶ PreparedStatement 객체로 데이터 접근하기

▶ PreparedStatement 객체

- ▶ 동적인 쿼리에 사용
- ▶ 하나의 객체로 여러 번의 쿼리를 실행
- ▶ 동일한 쿼리문을 특정 값만 바꾸어서 여러 번 실행해야 할 때, 매개 변수가 많아서 쿼리문을 정리해야 할 때 유용

```
PreparedStatement prepareStatement(String sql) throws SQLException
```



데이터베이스 쿼리 실행

▶ setXxx() 메소드의 종류

메소드	반환 유형	설명
setString(int parameterIndex, String x)	void	필드 유형이 문자열인 경우
setInt(int parameterIndex, int x)	void	필드 유형이 정수형인 경우
setLong(int parameterIndex, long x)	void	필드 유형이 정수형인 경우
setDouble(int parameterIndex, double x)	void	필드 유형이 실수형인 경우
setFloat(int parameterIndex, float x)	void	필드 유형이 실수형인 경우
setObject(int parameterIndex, Object x)	void	필드 유형이 객체형인 경우
setDate(int parameterIndex, Date x)	void	필드 유형이 날짜형인 경우
setTimestamp(int parameterIndex, Timestamp x)	void	필드 유형이 시간형인 경우

▶ PreparedStatement 객체의 메소드 종류

메소드	반환 유형	설명
executeQuery()	ResultSet	SELECT 문을 실행할 때 사용합니다(ResultSet 객체 반환).
executeUpdate()	int	삽입, 수정, 삭제와 관련된 SQL 문 실행에 사용합니다.
close():	void	Statement 객체를 반환할 때 사용합니다.

데이터베이스 쿼리 실행

▶ PreparedStatement 객체로 데이터 접근하기

▶ executeQuery() 메소드로 데이터 조회하기

- ▶ 동적인 SELECT 쿼리문을 통해 데이터를 검색하는 데 사용

```
int executeQuery() throws SQLException
```

▶ executeUpdate() 메소드로 데이터 삽입, 수정, 삭제하기

- ▶ INSERT, UPDATE, SELECT 쿼리문을 통해 데이터를 삽입, 수정, 삭제하는 데 사용

```
int executeUpdate() throws SQLException
```



데이터베이스 쿼리 실행

dbconn.jsp ✕

```
1 <%@ page import = "java.sql.*"%>
2 <%
3     Connection conn = null;
4
5     String url = "jdbc:mysql://localhost:3306/JSPTestDB";
6     String user = "root";
7     String password = "1234";
8
9     Class.forName("com.mysql.jdbc.Driver");
10    conn = DriverManager.getConnection(url, user, password);
11 %>
12
```

데이터베이스 쿼리 실행

▶ PreparedStatement 객체로 INSERT 쿼리문 실행하기

```
insert02.jsp
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <html>
3 <head>
4 <title>Database SQL</title>
5 </head>
6 <body>
7     <form method="post" action="insert02_process.jsp">
8         <p>아이디 : <input type="text" name="id">
9         <p>비밀번호 : <input type="password" name="passwd">
10        <p>이름 : <input type="text" name="name">
11        <p><input type="submit" value="전송">
12    </form>
13 </body>
14 </html>
```

데이터베이스 쿼리 실행

insert02_process.jsp

```
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <%@ page import = "java.sql.*"%>
3 <html>
4 <head>
5 <title>Database SQL</title>
6 </head>
7 <body>
8   <%@ include file="dbconn.jsp" %>
9   <%
10       request.setCharacterEncoding("utf-8");
11
12       String id = request.getParameter("id");
13       String passwd = request.getParameter("passwd");
14       String name = request.getParameter("name");
15
```


데이터베이스 쿼리 실행

```
16 PreparedStatement pstmt = null;
17 try{
18     String sql = "INSERT INTO Member(id, passwd, name) VALUES(?,?,?)";
19     pstmt = conn.prepareStatement(sql);
20     pstmt.setString(1, id);
21     pstmt.setString(2, passwd);
22     pstmt.setString(3, name);
23     pstmt.executeUpdate();
24     out.println("Member 테이블 삽입이 성공했습니다.<br>");
25 }catch(SQLException ex){
26     out.println("Member 테이블 삽입이 실패했습니다.<br>");
27     out.println("SQLException: " + ex.getMessage());
28 }finally{
29     if(pstmt != null)
30         pstmt.close();
31     if(conn != null)
32         conn.close();
33 }
34 %>
```

Member 테이블 삽입이 성공했습니다.

Status	Result1
	idpasswdname
1	11234홍길순
2	21235홍길동
Total 2 records shown	

쿼리문 실행 결과 값 가져오기

▶ ResultSet 객체

- ▶ PreparedStatement 객체로 SELECT 문을 사용하여 얻어온 레코드 값을 테이블 형태로 가진 객체

▶ [PreparedStatement 객체를 사용하는 경우]

```
ResultSet executeQuery() throws SQLException
```



쿼리문 실행 결과 값 가져오기

▶ ResultSet객체의 메소드 종류

메소드	반환 유형	설명
getXxx(int ColumnIndex)	XXX	설정된 ColumnIndex(필드 순번)의 필드 값을 설정한 XXX 형으로 가져옵니다.
getXxx(String ColumnName)	XXX	설정된 ColumnName(필드 순번)의 필드 값을 설정한 XXX 형으로 가져옵니다.
absolute(int row)	boolean	설정된 row 행으로 커서를 이동합니다.
beforeFirst()	void	첫 번째 행의 이전으로 커서를 이동합니다.
afterLast()	void	마지막 행의 다음으로 커서를 이동합니다.
first()	void	첫 번째 행으로 커서를 이동합니다.
last()	void	마지막 행으로 커서를 이동합니다.
next()	boolean	다음 행으로 커서를 이동합니다.
previous()	boolean	현재 행의 이전 행으로 커서를 이동합니다.
close():	void	ResultSet 객체를 반환할 때 사용합니다.



쿼리문 실행 결과 값 가져오기

- ▶ PreparedStatement객체를 이용하여 SELECT쿼리문 실행 결과 값 가져오기

```
select02.jsp
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <%@ page import = "java.sql.*"%>
3 <html>
4 <head>
5 <title>Database SQL</title>
6 </head>
7 <body>
8     <%@ include file="dbconn.jsp" %>
9     <table width="300" border="1">
10         <tr>
11             <th>아이디</th>
12             <th>비밀번호</th>
13             <th>이름</th>
14         </tr>
15
16     <%
17         ResultSet rs = null;
18         PreparedStatement pstmt = null;
```

아이디	비밀번호	이름
1	1234	홍길순
2	1235	홍길동

쿼리문 실행 결과 값 가져오기

```
19     try{
20         String sql = "select * from member";
21         pstmt = conn.prepareStatement(sql);
22         rs = pstmt.executeQuery();
23         while(rs.next()){
24             String id = rs.getString("id");
25             String pw = rs.getString("passwd");
26             String name = rs.getString("name");
27         }
28         <tr>
29             <td><%=id %></td>
30             <td><%=pw %></td>
31             <td><%=name %></td>
32         </tr>
33     <%
34     }
35     }catch(SQLException ex){
36         out.println("Member 테이블 호출이 실패했습니다.<br>");
37         out.println("SQLException: " + ex.getMessage());
38     }finally{
39         if(rs != null)
40             rs.close();
41         if(pstmt != null)
42             pstmt.close();
43         if(conn != null)
44             conn.close();
45     }
46     <%
47     </table>
48 </body>
49 </html>
```

쿼리문 실행 결과 값 가져오기

- ▶ PreparedStatement객체를 이용하여 UPDATE 쿼리문 실행 결과 값 가져오기

```
update02.jsp
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <html>
3 <head>
4 <title>Database SQL</title>
5 </head>
6 <body>
7     <form method="post" action="update02_process.jsp">
8         <p>아이디 : <input type="text" name="id">
9         <p>비밀번호 : <input type="password" name="passwd">
10        <p>이름 : <input type="text" name="name">
11        <p><input type="submit" value="전송">
12    </form>
13 </body>
14 </html>
```

Database SQL

/JSPBook/ch16/update02.jsp

아이디 : 2

비밀번호 :

이름 : 수여인

전송

쿼리문 실행 결과 값 가져오기

update02_process.jsp

```
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <%@ page import = "java.sql.*"%>
3 <html>
4 <head>
5 <title>Database SQL</title>
6 </head>
7 <body>
8     <%@ include file="dbconn.jsp" %>
9     <%
10         request.setCharacterEncoding("utf-8");
11         String id = request.getParameter("id");
12         String passwd = request.getParameter("passwd");
13         String name = request.getParameter("name");
14
15         ResultSet rs = null;
16         PreparedStatement pstmt = null, pstmt2 = null;
17         try{
18             String sql = "select id, passwd from member where id = ?";
19             pstmt = conn.prepareStatement(sql);
20             pstmt.setString(1, id);
21             rs = pstmt.executeQuery();
```

http://localhost:8090/JSPTest

Member 테이블을 수정했습니다.

Status	Result1		
	id	passwd	name
1	1	1234	관리자
2	2	1235	수여인

쿼리문 실행 결과 값 가져오기

```
23         if(rs.next()){
24             String rId = rs.getString("id");
25             String rPasswd = rs.getString("passwd");
26             if(id.equals(rId) && passwd.equals(rPasswd)){
27
28                 sql = "update member set name = ? where id = ?";
29                 pstmt2 = conn.prepareStatement(sql);
30                 pstmt2.setString(1, name);
31                 pstmt2.setString(2, id);
32                 pstmt2.executeUpdate();
33                 out.println("Member 테이블을 수정했습니다.");
34             }else
35                 out.println("일치하는 비밀번호가 아닙니다.");
36         }else
37             out.println("Member 테이블에 일치하는 아이디가 없습니다.");
38     }catch(SQLException ex){
39         out.println("SQLException: " + ex.getMessage());
40     }finally{
41         if(rs != null)
42             rs.close();
43         if(pstmt != null)
44             pstmt.close();
45         if(pstmt2 != null)
46             pstmt2.close();
47         if(conn != null)
48             conn.close();
49     }
50     %>
51 </body>
52 </html>
```


쿼리문 실행 결과 값 가져오기

- ▶ PreparedStatement객체를 이용하여 DELETE쿼리문 실행 결과 값 가져오기

delete02.jsp

```
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <html>
3 <head>
4 <title>Database SQL</title>
5 </head>
6 <body>
7     <form method="post" action="delete02_process.jsp">
8         <p>아이디 : <input type="text" name="id">
9         <p>비밀번호 : <input type="password" name="passwd">
10        <p><input type="submit" value="전송">
11    </form>
12 </body>
13 </html>
```

← → ⏏ 🔄

http://localhost:8090/JSPTest_/Chapter16/delete02.jsp

아이디 :

비밀번호 :

전송

쿼리문 실행 결과 값 가져오기

delete02_process.jsp

```
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <%@ page import = "java.sql.*"%>
3 <html>
4 <head>
5 <title>Database SQL</title>
6 </head>
7 <body>
8     <%@ include file="dbconn.jsp" %>
9     <%
10         request.setCharacterEncoding("utf-8");
11         String id = request.getParameter("id");
12         String passwd = request.getParameter("passwd");
13
14         ResultSet rs = null;
15         PreparedStatement pstmt = null, pstmt2 = null;
16         try{
17             String sql = "select id, passwd from member where id = ?";
18             pstmt = conn.prepareStatement(sql);
19             pstmt.setString(1, id);
20             rs = pstmt.executeQuery();
21         } catch (Exception e) {
22             e.printStackTrace();
23         }
24     %>
```

Database SQL

/ch16/delete01_process.jsp

Member 테이블을 삭제했습니다.

SQL Results

Status Result1

id	passwd	name

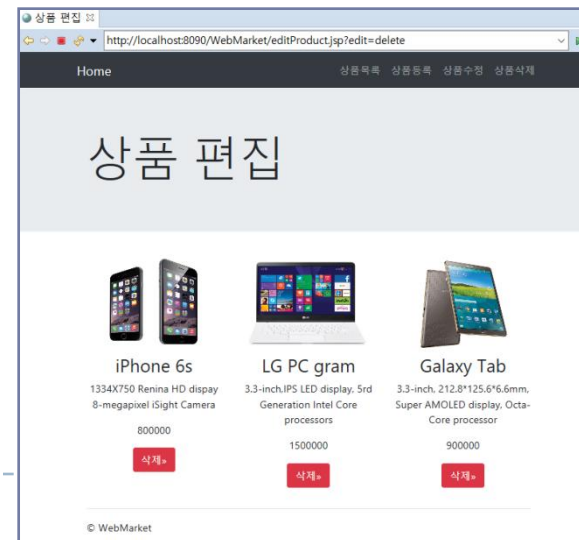
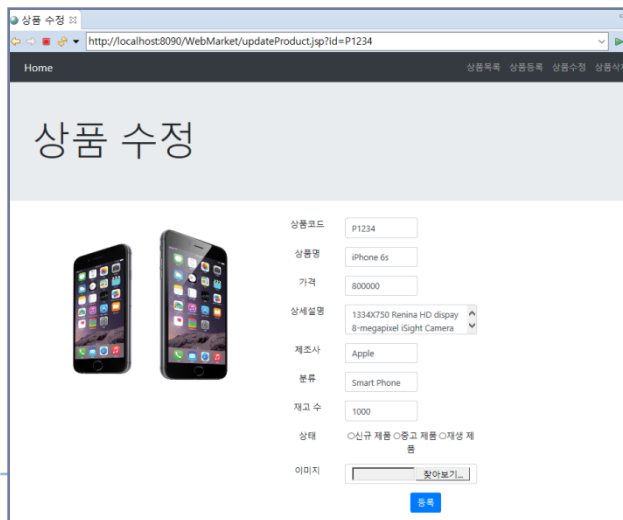
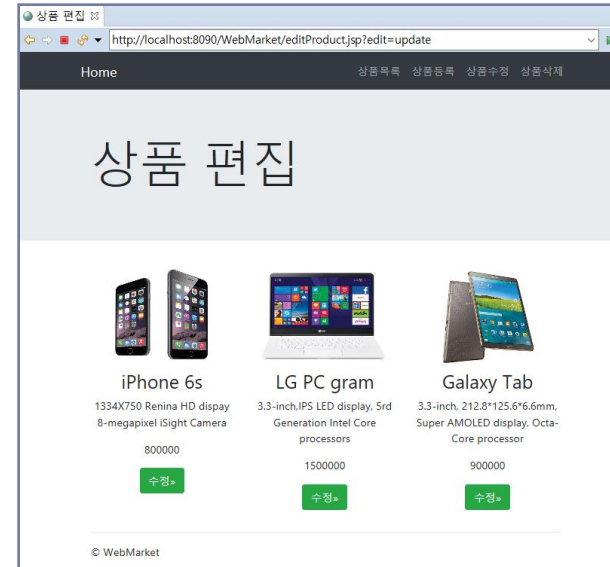
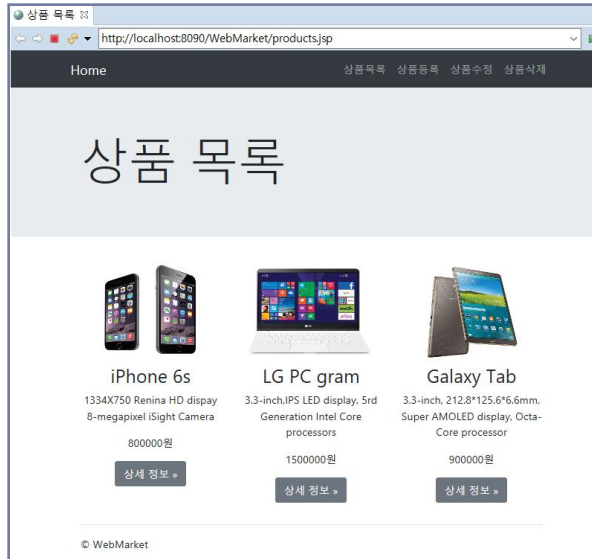
Total 0 records shown

쿼리문 실행 결과 값 가져오기

```
22     if(rs.next()){
23         String rId = rs.getString("id");
24         String rPasswd = rs.getString("passwd");
25         if(id.equals(rId) && passwd.equals(rPasswd)){
26             sql = "delete from member where id = ? and passwd = ?";
27             pstmt2 = conn.prepareStatement(sql);
28             pstmt2.setString(1, id);
29             pstmt2.setString(2, passwd);
30             pstmt2.executeUpdate();
31             out.println("Member 테이블을 삭제했습니다.");
32         }else
33             out.println("일치하는 비밀번호가 아닙니다.");
34     }else
35         out.println("Member 테이블에 일치하는 아이디가 없습니다.");
36 }catch(SQLException ex){
37     out.println("SQLException: " + ex.getMessage());
38 }finally{
39     if(rs != null)
40         rs.close();
41     if(pstmt != null)
42         pstmt.close();
43     if(pstmt2 != null)
44         pstmt2.close();
45     if(conn != null)
46         conn.close();
47 }
48 %>
49 </body>
50 </html>
```

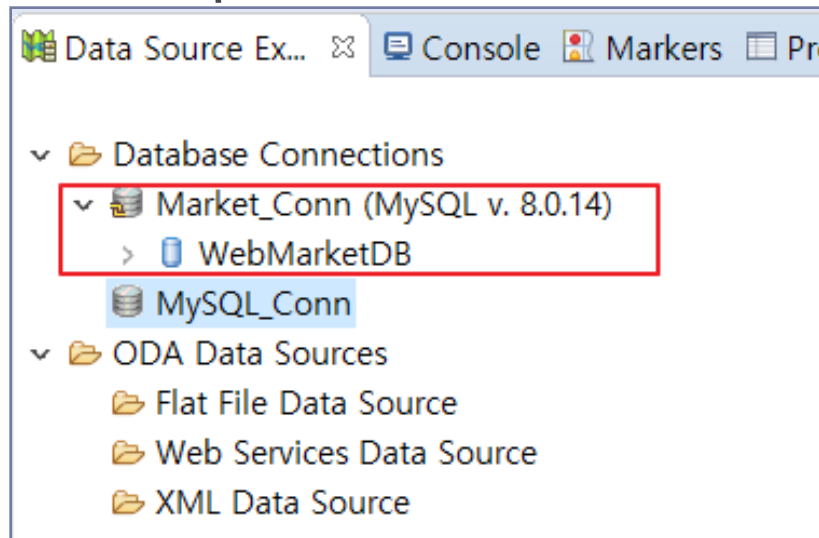
http://localhost:8090/JSPTest_			
Member 테이블을 삭제했습니다.			
Status	Result1		
	id	passwd	name
Total 0 records shown			

상품 조회, 등록, 수정, 삭제하기



상품 조회, 등록, 수정, 삭제하기

- ▶ 데이터베이스에 저장된 상품 목록을 가져와 출력하기
 - ▶ 데이터베이스 커넥션생성: 커넥션 이름은 Market_Conn이고 데이터베이스 이름은 WebMarketDB를 사용하여 Data Source Explorer에서 실행.
 - ▶ 생성한 product 테이블을 사용



데이터베이스 연결 페이지 만들기

```
dbconn.jsp
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <%@ page import = "java.sql.*"%>
3 <%
4     Connection conn = null;
5
6     try{
7         String url = "jdbc:mysql://localhost:3306/WebMarketDB";
8         String user = "root";
9         String password ="1234";
10
11         Class.forName("com.mysql.jdbc.Driver");
12         conn = DriverManager.getConnection(url, user, password);
13     }catch(SQLException ex) {
14         out.println("데이터베이스 연결이 실패했습니다.<br>");
15         out.println("SQLException: " + ex.getMessage());
16     }
17 %>
```

메뉴 페이지 수정하기

menu.jsp

```
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <nav class="navbar navbar-expand navbar-dark bg-dark">
3   <div class="container">
4     <div class="navbar-header">
5       <a class="navbar-brand" href="./welcome.jsp">Home</a>
6     </div>
7     <div>
8       <ul class="navbar-nav mr-auto">
9         <li class="nav-item">
10          <a class="nav-link" href="./products.jsp">상품목록</a>
11        </li>
12      </ul>
13    </div>
14  </div>
15 </nav>
```

상품 목록 페이지 수정하기

products.jsp

```
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <%@ page import="java.sql.*"%>
3 <html>
4 <head>
5 <link rel="stylesheet" href="../resources/css/bootstrap.min.css" />
6 <title>상품 목록</title>
7 </head>
8 <body>
9 <jsp:include page="menu.jsp" />
10 <div class="jumbotron">
11 <div class="container">
12 <h1 class="display-3">상품 목록</h1>
13 </div>
14 </div>
15 <div class="container">
16 <div class="row" align="center">
17 <%@ include file="dbconn.jsp" %>
```



```
16<div class="row" align="center">
17  <%@ include file="dbconn.jsp" %>
18  <%
19    PreparedStatement pstmt = null;
20    ResultSet rs = null;
21    String sql = "select * from product";
22    pstmt = conn.prepareStatement(sql);
23    rs = pstmt.executeQuery();
24    while(rs.next()){
25  %>
26    <div class="col-md-4">
27      "
28      style="width: 100%">
29      <h3><%=rs.getString("p_name")%></h3>
30      <p><%=rs.getString("p_description")%>
31      <p><%=rs.getString("p_UnitPrice")%>원
32      <p><a href="./product.jsp?id=<%=rs.getString("p_id")%>"
33        class="btn btn-secondary" role="button"> 상세 정보
34        &raquo;</a>
35    </div>
36    <%=}%>
37  </div>
38  <hr>
39 </div>
40 <jsp:include page="footer.jsp" />
41 </body>
42 </html>
```

상품 조회, 등록, 수정, 삭제 하기

- ▶ 데이터베이스에 새로운 상품 등록하기
 - ▶ 메뉴 페이지 수정하기

```
menu.jsp
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <nav class="navbar navbar-expand navbar-dark bg-dark">
3   <div class="container">
4     <div class="navbar-header">
5       <a class="navbar-brand" href="./welcome.jsp">Home</a>
6     </div>
7     <div>
8       <ul class="navbar-nav mr-auto">
9         <li class="nav-item">
10           <a class="nav-link" href="./products.jsp">상품목록</a>
11         </li>
12         <li class="nav-item">
13           <a class="nav-link" href="./addProduct.jsp">상품등록</a>
14         </li>
15       </ul>
16     </div>
17   </div>
18 </nav>
```

상품 등록 처리 페이지 수정하기

processAddProduct.jsp

```
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <%@ page import="com.oreilly.servlet.*"%>
3 <%@ page import="com.oreilly.servlet.multipart.*"%>
4 <%@ page import="java.util.*"%>
5 <%@ page import="java.sql.*"%>
6 <%@ include file="dbconn.jsp" %>
```

상품 등록 처리 페이지 수정하기

```
44 PreparedStatement pstmt = null;
45 String sql = "insert into product values(?,?,?,?,?,?,?,?,?,?)";
46 pstmt = conn.prepareStatement(sql);
47 pstmt.setString(1, productId);
48 pstmt.setString(2, name);
49 pstmt.setInt(3, price);
50 pstmt.setString(4, description);
51 pstmt.setString(5, category);
52 pstmt.setString(6, manufacturer);
53 pstmt.setLong(7, stock);
54 pstmt.setString(8, condition);
55 pstmt.setString(9, fileName);
56 pstmt.executeUpdate();
57
58 if(pstmt != null)
59     pstmt.close();
60 if(conn != null)
61     conn.close();
62
63 response.sendRedirect("products.jsp");
64 %>
```

데이터베이스에 저장된 상품 수정하기

▶ 메뉴 페이지 수정하기

```
menu.jsp ✖
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <nav class="navbar navbar-expand navbar-dark bg-dark">
3   <div class="container">
4     <div class="navbar-header">
5       <a class="navbar-brand" href="./welcome.jsp">Home</a>
6     </div>
7     <div>
8       <ul class="navbar-nav mr-auto">
9         <li class="nav-item">
10           <a class="nav-link" href="./products.jsp">상품목록</a>
11         </li>
12         <li class="nav-item">
13           <a class="nav-link" href="./addProduct.jsp">상품등록</a>
14         </li>
15         <li class="nav-item">
16           <a class="nav-link" href="./editProduct.jsp?
17             edit=update">상품수정</a>
18         </li>
```

상품 편집 페이지 작성하기

editProduct.jsp

```
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <%@ page import = "java.sql.*"%>
3 <html>
4 <head>
5 <link rel="stylesheet" href="./resources/css/bootstrap.min.css" />
6 <title>상품 편집</title>
7 </head>
8 <%
9     String edit = request.getParameter("edit");
10 %>
11 <body>
12     <jsp:include page="menu.jsp" />
13     <div class="jumbotron">
14         <div class="container">
15             <h1 class="display-3">상품 편집</h1>
16         </div>
17     </div>
18     <div class="container">
19         <div class="row" align="center">
20             <%@ include file="dbconn.jsp" %>
```

상품 편집 페이지 작성하기

```
editProduct.jsp
18  <div class="container">
19  <div class="row" align="center">
20  <%@ include file="dbconn.jsp" %>
21  <%
22      PreparedStatement pstmt = null;
23      ResultSet rs = null;
24      String sql = "select * from product";
25      pstmt = conn.prepareStatement(sql);
26      rs = pstmt.executeQuery();
27      while(rs.next()){
28  %>
29  <div class="col-md-4">
30      "
31      style="width:100%">
32      <h3><%=rs.getString("p_name")%></h3>
33      <p><%=rs.getString("p_description")%>
34      <p><%=rs.getString("p_UnitPrice")%>
35      <p><%
36          if(edit.equals("update")){
37              %>
38              <a href="./updateProduct.jsp?id=<%=rs.getString("p_id")%>"
39              class="btn btn-success" role="button">수정</a>
40              <%}
41              %>
42  </div>
```

상품 편집 페이지 작성하기

```
43  <%  
44      }  
45      if(rs != null)  
46          rs.close();  
47      if(pstmt != null)  
48          pstmt.close();  
49      if(conn != null)  
50          conn.close();  
51  %>  
52  </div>  
53  <hr>  
54  </div>  
55  <jsp:include page="footer.jsp" />  
56 </body>  
57 </html>
```



상품 수정 페이지 작성하기

updateProduct.jsp

```
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <%@ page import="java.sql.*"%>
3 <html>
4 <head>
5 <link rel="stylesheet" href="../resources/css/bootstrap.min.css" />
6 <title>상품 수정</title>
7 </head>
8 <body>
9 <jsp:include page="menu.jsp" />
10 <div class="jumbotron">
11 <div class="container">
12 <h1 class="display-3">상품 수정</h1>
13 </div>
14 </div>
15 <%@ include file="dbconn.jsp" %>
```

```
16<%
17    String productId = request.getParameter("id");
18    PreparedStatement pstmt = null;
19    ResultSet rs = null;
20    String sql = "select * from product where p_id=?";
21    pstmt = conn.prepareStatement(sql);
22    pstmt.setString(1, productId);
23    rs = pstmt.executeQuery();
24    while(rs.next()){
25%>
26<div class="container">
27    <div class="row" align="center">
28        <div class="col-md-5">
29            "
30                style="width: 100%">
31        </div>
32        <div class="col-md-7">
33            <form name="newProduct" action="./processUpdateProduct.jsp"
34                class="form-horizontal" method="post"
35                enctype="multipart/form-data">
36                <div class="form-group row">
37                    <label class="col-sm-2">상품코드</label>
38                    <div class="col-sm-3">
39                        <input type="text" id="productId" name="productId"
40                            class="form-control" value='<%=rs.getString("p_id")%>'>
41                    </div>
42                </div>
```

```
43<div class="form-group row">
44  <label class="col-sm-2">상품명</label>
45  <div class="col-sm-3">
46    <input type="text" id="name" name="name"
47      class="form-control" value='<%=rs.getString("p_name")%>'>
48  </div>
49</div>
50<div class="form-group row">
51  <label class="col-sm-2">가격</label>
52  <div class="col-sm-3">
53    <input type="text" id="unitPrice" name="unitPrice"
54      class="form-control" value='<%=rs.getInt("p_unitPrice")%>'>
55  </div>
56</div>
57<div class="form-group row">
58  <label class="col-sm-2">상세설명</label>
59  <div class="col-sm-5">
60    <textarea name="description" cols="50" rows="2"
61      class="form-control"><%=rs.getString("p_description")%>
62    </textarea>
63  </div>
64</div>
65<div class="form-group row">
66  <label class="col-sm-2">제조사</label>
67  <div class="col-sm-3">
68    <input type="text" id="manufacturer" name="manufacturer"
69      class="form-control" value='<%=rs.getString("p_manufacturer")%>'>
70  </div>
71</div>
```

```
72<div class="form-group row">
73    <label class="col-sm-2">분류</label>
74    <div class="col-sm-3">
75        <input type="text" id="category" name="category"
76            class="form-control" value='<%=rs.getString("p_category")%>'>
77    </div>
78</div>
79<div class="form-group row">
80    <label class="col-sm-2">재고 수</label>
81    <div class="col-sm-3">
82        <input type="text" id="unitsInStock" name="unitsInStock"
83            class="form-control" value='<%=rs.getLong("p_unitsInStock")%>'>
84    </div>
85</div>
86<div class="form-group row">
87    <label class="col-sm-2">상태</label>
88    <div class="col-sm-5">
89        <input type="radio" name="condition" value = "New">신규 제품
90        <input type="radio" name="condition" value = "Old">중고 제품
91        <input type="radio" name="condition" value = "Refurbished">재생 제품
92    </div>
93</div>
94<div class="form-group row">
95    <label class="col-sm-2">이미지</label>
96    <div class="col-sm-5">
97        <input type="file" name="productImage" class="form-control">
98    </div>
99</div>
```

*updateProduct.jsp

```
95         <label class="col-sm-2">이미지</label>
96         <div class="col-sm-5">
97             <input type="file" name="productImage" class="form-control">
98         </div>
99     </div>
100     <div class="form-group row">
101         <div class="col-sm-offset-2 col-sm-10">
102             <input type="submit" name="productImage"
103                 class="btn btn-primary" value="등록">
104         </div>
105     </div>
106 </form>
107 </div>
108 </div>
109 </div>
110 <%
111     }
112     if(rs != null)
113         rs.close();
114     if(pstmt != null)
115         pstmt.close();
116     if(conn != null)
117         conn.close();
118 %>
119 </body>
120 </html>
```

상품 수정 처리 페이지 작성하기

processUpdateProduct.jsp ☒

```
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <%@ page import="com.oreilly.servlet.*"%>
3 <%@ page import="com.oreilly.servlet.multipart.*"%>
4 <%@ page import="java.util.*"%>
5 <%@ page import="java.sql.*"%>
6 <%@ include file="dbconn.jsp" %>
7 <%
8     request.setCharacterEncoding("UTF-8");
9     String filename = "";
10    String realFolder = "d:\\upload"; //웹 어플리케이션상의 절대 경로
11    String encType = "utf-8"; //인코딩 타입
12    int maxSize = 5 * 1024 * 1024; //최대 업로드될 파일의 크기5Mb
13
14    MultipartRequest multi = new MultipartRequest(request, realFolder,
15        maxSize, encType, new DefaultFileRenamePolicy());
16
17    String productId = multi.getParameter("productId");
18    String name = multi.getParameter("name");
19    String unitPrice = multi.getParameter("unitPrice");
20    String description = multi.getParameter("description");
21    String manufacturer = multi.getParameter("manufacturer");
22    String category = multi.getParameter("category");
23    String unitsInStock = multi.getParameter("unitsInStock");
24    String condition = multi.getParameter("condition");
```

상품 수정 처리 페이지 작성하기

processUpdateProduct.jsp

```
25
26 Integer price;
27 if (unitPrice.isEmpty())
28     price = 0;
29 else
30     price = Integer.valueOf(unitPrice);
31
32 long stock;
33
34 if (unitsInStock.isEmpty())
35     stock = 0;
36 else
37     stock = Long.valueOf(unitsInStock);
38
39 Enumeration files = multi.getFileNames();
40 String fname = (String) files.nextElement();
41 String fileName = multi.getFilesystemName(fname);
```

```
43 PreparedStatement pstmt = null;
44 ResultSet rs = null;
45 String sql = "select * from product where p_id=?";
46 pstmt = conn.prepareStatement(sql);
47 pstmt.setString(1, productId);
48 rs = pstmt.executeQuery();
49 if(rs.next()){
50     sql="UPDATE product SET p_name=?, p_unitPrice=," +
51         "p_description=?, p_manufacturer=?, p_category=," +
52         "p_unitsInStock=?, p_condition=?";
53     if(fileName != null){
54         sql+=" , p_fileName=? WHERE p_id=?";
55     } else{
56         sql+=" WHERE p_id=?";
57     }
58     pstmt=conn.prepareStatement(sql);
59     pstmt.setString(1, name);
60     pstmt.setInt(2, price);
61     pstmt.setString(3, description);
62     pstmt.setString(4, manufacturer);
63     pstmt.setString(5, category);
64     pstmt.setLong(6, stock);
65     pstmt.setString(7, condition);
```



```
66     if(fileName != null){
67         pstmt.setString(8, fileName);
68         pstmt.setString(9, productId);
69     }else{
70         pstmt.setString(8, productId);
71     }
72     pstmt.executeUpdate();
73 }// if(rs.next())
74 if(rs != null)
75     rs.close();
76 if(pstmt != null)
77     pstmt.close();
78 if(conn != null)
79     conn.close();
80 response.sendRedirect("editProduct.jsp?edit=update");
81 %>
```



데이터베이스에 저장된 상품 삭제하기

▶ 메뉴 페이지 수정하기

```
menu.jsp
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <nav class="navbar navbar-expand navbar-dark bg-dark">
3   <div class="container">
4     <div class="navbar-header">
5       <a class="navbar-brand" href="./welcome.jsp">Home</a>
6     </div>
7     <div>
8       <ul class="navbar-nav mr-auto">
9         <li class="nav-item">
10          <a class="nav-link" href="./products.jsp">상품목록</a>
11        </li>
12        <li class="nav-item">
13          <a class="nav-link" href="./addProduct.jsp">상품등록</a>
14        </li>
15        <li class="nav-item">
16          <a class="nav-link" href="./editProduct.jsp?
17            edit=update">상품수정</a>
18        </li>
19        <li class="nav-item">
20          <a class="nav-link" href="./editProduct.jsp?
21            edit=delete">상품삭제</a>
22        </li>
23      </ul>
24    </div>
25  </div>
26 </nav>
```

상품 편집 페이지 작성하기

▶ editProduct.jsp파일 수정하기

```
editProduct.jsp
7 <title>상품 편집</title>
8 <script type="text/javascript">
9     function deleteConfirm(id){
10         if(confirm("해당 상품을 삭제합니다!!") == true)
11             location.href="./deleteProduct.jsp?id="+ id;
12         else
13             return;
14     }
15 </script>
16 </head>

44 <p><%
45     if(edit.equals("update")){
46         %>
47         <a href="./updateProduct.jsp?id=<%=rs.getString("p_id")%>"
48             class="btn btn-success" role="button">수정&raquo;</a>
49         <%
50     } else if(edit.equals("delete")){
51         %>
52         <a href="#" onclick="deleteConfirm('<%=rs.getString("p_id")%>')"
53             class="btn btn-danger" role="button">삭제&raquo;</a>
54         <%
55     }
56     %>
57 </div>
```

상품 삭제 페이지 작성하기

```
deleteProduct.jsp
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <%@ page import = "java.sql.*"%>
3 <%@ include file="dbconn.jsp" %>
4 <%
5     String productId = request.getParameter("id");
6     PreparedStatement pstmt = null;
7     ResultSet rs = null;
8     String sql = "select * from product";
9     pstmt = conn.prepareStatement(sql);
10    rs = pstmt.executeQuery();
11    if(rs.next()){
12        sql = "delete from product where p_id=?";
13        pstmt = conn.prepareStatement(sql);
14        pstmt.setString(1, productId);
15        pstmt.executeUpdate();
16    }else{
17        out.println("일치하는 상품이 없습니다.");
18    }
19    if(pstmt != null)
20        pstmt.close();
21    if(conn != null)
22        conn.close();
23    response.sendRedirect("editProduct.jsp?edit=delete");
24 %>
```