




SQL 활용

2022/10/21(금)

김지홍

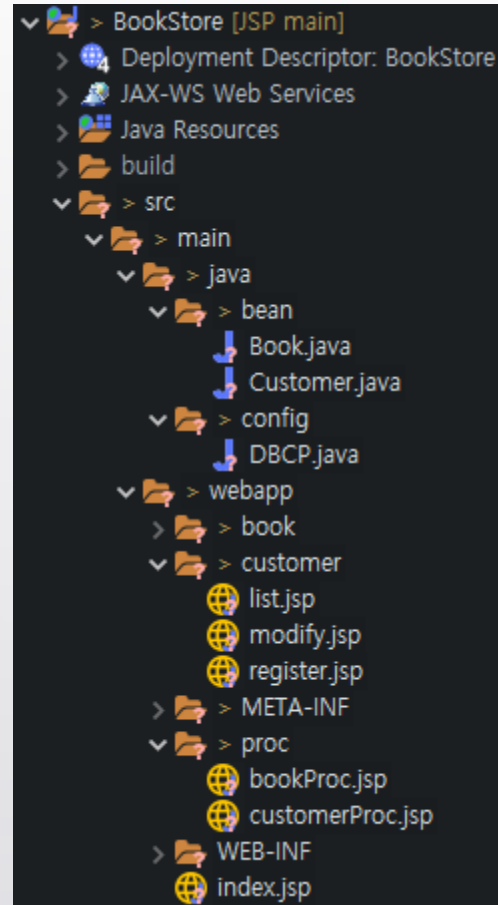


목차

1. 프로젝트 생성
2. 화면 구현
3. 기능 구현
4. 실행

문제 1. 프로젝트 생성

3



문제 2. 화면 구현

4

BookStore

[도서목록](#) [고객목록](#)

도서목록

[처음으로 도서등록](#)

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	수정 삭제
2	축구아는 여자	나무수	13000	수정 삭제
3	축구의 이해	대한미디어	22000	수정 삭제
4	골프 바이블	대한미디어	35000	수정 삭제
5	피겨 교본	굿스포츠	8000	수정 삭제
6	역도 단계별기술	굿스포츠	6000	수정 삭제
7	야구의 추억	이상미디어	20000	수정 삭제
8	야구를 부탁해	이상미디어	13000	수정 삭제
9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Champions	Pearson	13000	수정 삭제

도서등록

[처음으로 도서목록](#)

도서명	<input type="text" value="도서명 입력"/>
출판사	<input type="text" value="출판사 입력"/>
가격	<input type="text" value="가격 입력"/>
<input type="button" value="등록"/>	

도서수정

[처음으로 도서목록](#)

도서번호	<input type="text" value="1"/>
도서명	<input type="text" value="축구의 역사"/>
출판사	<input type="text" value="굿스포츠"/>
가격	<input type="text" value="7000"/>
<input type="button" value="수정"/>	

문제 2. 화면 구현

5

고객목록

[처음으로 고객등록](#)

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스터	000-5000-0001	수정 삭제
2	김연아	대한민국 서울	000-6000-0001	수정 삭제
3	장미란	대한민국 강원도	000-7000-0001	수정 삭제
4	추신수	미국 클리블랜드	000-8000-0001	수정 삭제
5	박세리	대한민국 부산	null	수정 삭제
6	이단계사	광주	010-1234	수정 삭제
7	큐일삼	큐일도	001	수정 삭제
9	김유신	제주	010-1234	수정 삭제

고객등록

[처음으로 고객목록](#)

고객명	<input type="text" value="고객명 입력"/>
주소	<input type="text" value="주소 입력"/>
휴대폰	<input type="text" value="휴대폰 입력"/>
<input type="button" value="등록"/>	

고객수정

[처음으로 고객목록](#)

고객번호	<input type="text" value="1"/>
고객명	<input type="text" value="박지성"/>
주소	<input type="text" value="영국 맨체스터"/>
휴대폰	<input type="text" value="000-5000-0001"/>
<input type="button" value="수정"/>	

문제 3. 기능 구현(Book)

6

목록 / 삭제

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>도서목록</title>
  </head>
  <body>
    <div class="container">
      <h3>도서목록</h3>
      <a href="../index.jsp">처음으로</a>
      <a href="../register.jsp">도서등록</a>
      <table border="1">
        <tr>
          <th>도서번호</th>
          <th>도서명</th>
          <th>출판사</th>
          <th>가격</th>
          <th>관리</th>
        </tr>
        <% For(Book book : books){ %>
        <tr>
          <td><%= book.getBookId() %></td>
          <td><%= book.getBookName() %></td>
          <td><%= book.getPublisher() %></td>
          <td><%= book.getPrice() %></td>
          <td>
            <a href="../modify.jsp?bookId=<%= book.getBookId() %>">수정</a>
            <a href="../list.jsp?bookId=<%= book.getBookId() %>">삭제</a>
          </td>
        </tr>
        <% %>
      </table>
    </div>
  </body>
</html>
```

```
<%
request.setCharacterEncoding("UTF-8");
String bookId = request.getParameter("bookId");
List<Book> books = new ArrayList<>();
try{ // 삭제
  if(bookId != null){
    Connection conn = DBCP.getConnection();
    String sql = "DELETE FROM `book` WHERE `bookId`=?";
    PreparedStatement psmt = conn.prepareStatement(sql);
    psmt.setString(1, bookId);
    psmt.executeUpdate();
    psmt.close();
    conn.close();

    response.sendRedirect("../list.jsp");
  }else{ // 목록
    Connection conn = DBCP.getConnection();
    String sql = "SELECT * FROM `book`";
    PreparedStatement psmt = conn.prepareStatement(sql);

    ResultSet rs = psmt.executeQuery();

    while(rs.next()){
      Book book = new Book();
      book.setBookId(rs.getInt(1));
      book.setBookName(rs.getString(2));
      book.setPublisher(rs.getString(3));
      book.setPrice(rs.getInt(4));

      books.add(book);
    }
    rs.close();
    psmt.close();
    conn.close();
  }
}catch(Exception e){
  e.printStackTrace();
}
%>
```


문제 3. 기능 구현(Book)

7

등록

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>도서등록</title>
  </head>
  <body>
    <h3>도서등록</h3>
    <a href="../index.jsp">처음으로</a>
    <a href="../list.jsp">도서목록</a>
    <form action="../proc/bookProc.jsp" method="post">
      <input type="hidden" name="opt" value="opt_register">
      <input type="hidden" name="bookId" value="<%= book.getBookId()+1 %>">
      <table border="1">
        <tr>
          <td>도서명</td>
          <td><input type="text" name="bookName" placeholder="도서명 입력"></td>
        </tr>
        <tr>
          <td>출판사</td>
          <td><input type="text" name="publisher" placeholder="출판사 입력"></td>
        </tr>
        <tr>
          <td>가격</td>
          <td><input type="text" name="price" placeholder="가격 입력"></td>
        </tr>
        <tr>
          <td colspan="2" align="right">
            <input type="submit" value="등록">
          </td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

```
<%
// bookId 자동증가
Book book = null;
try{
    Connection conn = DBCP.getConnection();
    String sql = "SELECT `bookId` from `book` ORDER BY `bookId` DESC LIMIT 1";
    PreparedStatement psmt = conn.prepareStatement(sql);
    ResultSet rs = psmt.executeQuery();
    if(rs.next()){
        book = new Book();
        book.setBookId(rs.getInt(1));
    }
    rs.close();
    psmt.close();
    conn.close();
}catch(Exception e){
    e.printStackTrace();
}
%>
```

문제 3. 기능 구현(Book)

8

수정

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>도서수정</title>
  </head>
  <body>
    <h3>도서수정</h3>
    <a href="../index.jsp">처음으로</a>
    <a href="../list.jsp">도서목록</a>
    <form action="../proc/bookProc.jsp" method="post">
      <input type="hidden" name="opt" value="opt_modify">
      <table border="1">
        <tr>
          <td>도서번호</td>
          <td><input type="text" name="bookId" value="%= book.getBookId() %%" readonly="readonly"></td>
        </tr>
        <tr>
          <td>도서명</td>
          <td><input type="text" name="bookName" value="%= book.getBookName() %%"></td>
        </tr>
        <tr>
          <td>출판사</td>
          <td><input type="text" name="publisher" value="%= book.getPublisher() %%"></td>
        </tr>
        <tr>
          <td>가격</td>
          <td><input type="text" name="price" value="%= book.getPrice() %%"></td>
        </tr>
        <tr>
          <td colspan="2" align="right">
            <input type="submit" value="수정">
          </td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

```
<% // bookId 로 데이터 가져오기
    request.setCharacterEncoding("UTF-8");
    String bookId = request.getParameter("bookId");

    Book book = new Book();
    try{
        Connection conn = DBCP.getConnection();
        String sql = "SELECT * FROM `book` WHERE `bookId` = ?";
        PreparedStatement psmt = conn.prepareStatement(sql);
        psmt.setString(1, bookId);

        ResultSet rs = psmt.executeQuery();

        if(rs.next()){
            book = new Book();
            book.setBookId(rs.getInt(1));
            book.setBookName(rs.getString(2));
            book.setPublisher(rs.getString(3));
            book.setPrice(rs.getInt(4));

        }
        rs.close();
        psmt.close();
        conn.close();

    }catch(Exception e){
        e.printStackTrace();
    }
}>
```


문제 3. 기능 구현(Book)

9

등록/수정

```
<%
    request.setCharacterEncoding("UTF-8");
    String bookId    = request.getParameter("bookId");
    String bookName  = request.getParameter("bookName");
    String publisher  = request.getParameter("publisher");
    String price     = request.getParameter("price");
    String opt       = request.getParameter("opt");

    try{
        Connection conn = DBCP.getConnection();
        String sql = "";
        if(opt.equals("opt_modify")){ // 수정
            sql += "UPDATE `book` SET ";
            sql += "`bookName` = ?, ";
            sql += "`publisher` = ?, ";
            sql += "`price` = ? ";
            sql += "WHERE `bookId` =? ";
        }else if(opt.equals("opt_register")){ // 등록
            sql += "INSERT INTO `book` SET ";
            sql += "`bookName` = ?, ";
            sql += "`publisher` = ?, ";
            sql += "`price` = ?, ";
            sql += "`bookId` = ? ";
        }

        PreparedStatement psmt = conn.prepareStatement(sql);
        psmt.setString(1, bookName);
        psmt.setString(2, publisher);
        psmt.setString(3, price);
        psmt.setString(4, bookId);

        psmt.executeUpdate();

        psmt.close();
        conn.close();

    }catch(Exception e){
        e.printStackTrace();
    }
    response.sendRedirect("../book/list.jsp");
%>
```

문제 3. 기능 구현(CUSTOMER)

10

목록 / 삭제

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>고객목록</title>
  </head>
  <body>
    <h3>고객목록</h3>
    <a href=" ../index.jsp">처음으로</a>
    <a href=" ../register.jsp">고객등록</a>
    <table border="1">
      <tr>
        <th>고객번호</th>
        <th>고객명</th>
        <th>주소</th>
        <th>휴대폰</th>
        <th>관리</th>
      </tr>
      <% for(Customer customer : customers) { %>
      <tr>
        <td><%= customer.getCustId() %></td>
        <td><%= customer.getName() %></td>
        <td><%= customer.getAddress() %></td>
        <td><%= customer.getPhone() %></td>
        <td>
          <a href=" ../modify.jsp?custId=<%= customer.getCustId() %>">수정</a>
          <a href=" ../list.jsp?custId=<%= customer.getCustId() %>">삭제</a>
        </td>
      </tr>
      <% } %>
    </table>
  </body>
</html>
```

```
<%
request.setCharacterEncoding("UTF-8");
String custId = request.getParameter("custId");

List<Customer> customers = new ArrayList<>();
try{
  if(custId != null){
    Connection conn = DBCP.getConnection();
    String sql = "DELETE FROM `customer` WHERE `custId`=?";
    PreparedStatement psmt = conn.prepareStatement(sql);
    psmt.setString(1, custId);
    psmt.executeUpdate();
    psmt.close();
    conn.close();

    response.sendRedirect("../list.jsp");
  }else{
    Connection conn = DBCP.getConnection();
    String sql = "SELECT * FROM `customer`";
    PreparedStatement psmt = conn.prepareStatement(sql);

    ResultSet rs = psmt.executeQuery();

    while(rs.next()){
      Customer customer = new Customer();
      customer.setCustId(rs.getInt(1));
      customer.setName(rs.getString(2));
      customer.setAddress(rs.getString(3));
      customer.setPhone(rs.getString(4));

      customers.add(customer);
    }
    rs.close();
    psmt.close();
    conn.close();
  }

}catch(Exception e){
  e.printStackTrace();
}
%>
```

문제 3. 기능 구현 (CUSTOMER)

11

등록/수정

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>고객등록</title>
  </head>
  <body>
    <h3>고객등록</h3>
    <a href="../index.jsp">처음으로</a>
    <a href="../list.jsp">고객목록</a>
    <form action="../proc/customerProc.jsp" method="post">
      <input type="hidden" name="opt" value="opt_register">
      <table border="1">
        <tr>
          <td>고객명</td>
          <td><input type="text" name="name" placeholder="고객명 입력"></td>
        </tr>
        <tr>
          <td>주소</td>
          <td><input type="text" name="address" placeholder="주소 입력"></td>
        </tr>
        <tr>
          <td>휴대폰</td>
          <td><input type="text" name="phone" placeholder="휴대폰 입력"></td>
        </tr>
        <tr>
          <td colspan="2" align="right">
            <input type="submit" value="등록">
          </td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>고객수정</title>
  </head>
  <body>
    <h3>고객수정</h3>
    <a href="../index.jsp">처음으로</a>
    <a href="../list.jsp">고객목록</a>
    <form action="../proc/customerProc.jsp">
      <input type="hidden" name="opt" value="opt_modify">
      <table border="1">
        <tr>
          <td>고객번호</td>
          <td><input type="text" name="custId" value="%= customer.getCustId() %" readonly="readonly"></td>
        </tr>
        <tr>
          <td>고객명</td>
          <td><input type="text" name="name" value="%= customer.getName() %"></td>
        </tr>
        <tr>
          <td>주소</td>
          <td><input type="text" name="address" value="%= customer.getAddress() %"></td>
        </tr>
        <tr>
          <td>휴대폰</td>
          <td><input type="text" name="phone" value="%= customer.getPhone() %"></td>
        </tr>
        <tr>
          <td colspan="2" align="right">
            <input type="submit" value="수정">
          </td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

문제 3. 기능 구현(CUSTOMER)

12

등록/수정

```
<% // custId로 data 불러오기
request.setCharacterEncoding("UTF-8");
String custId = request.getParameter("custId");

Customer customer = new Customer();
try{
    Connection conn = DBCP.getConnection();
    String sql = "SELECT * FROM `customer` WHERE `custId` = ?";
    PreparedStatement psmt = conn.prepareStatement(sql);
    psmt.setString(1, custId);

    ResultSet rs = psmt.executeQuery();

    if(rs.next()){
        customer = new Customer();
        customer.setCustId(rs.getInt(1));
        customer.setName(rs.getString(2));
        customer.setAddress(rs.getString(3));
        customer.setPhone(rs.getString(4));
    }
    rs.close();
    psmt.close();
    conn.close();
}catch(Exception e){
    e.printStackTrace();
}
%>
```

```
<%
request.setCharacterEncoding("UTF-8");
String custId = request.getParameter("custId");
String name = request.getParameter("name");
String address = request.getParameter("address");
String phone = request.getParameter("phone");
String opt = request.getParameter("opt");

try{
    Connection conn = DBCP.getConnection();

    String sql = "";
    if(opt.equals("opt_modify")){ // 수정
        sql += "UPDATE `customer` SET ";
        sql += "`name` = ?, ";
        sql += "`address` = ?, ";
        sql += "`phone` = ? ";
        sql += "WHERE `custId` =? ";
    }else if(opt.equals("opt_register")){ // 등록
        sql += "INSERT INTO `customer` SET ";
        sql += "`name` = ?, ";
        sql += "`address` = ?, ";
        sql += "`phone` = ?, ";
        sql += "`custId` = ? ";
    }

    PreparedStatement psmt = conn.prepareStatement(sql);
    psmt.setString(1, name);
    psmt.setString(2, address);
    psmt.setString(3, phone);
    psmt.setString(4, custId);

    psmt.executeUpdate();

    psmt.close();
    conn.close();

}catch(Exception e){
    e.printStackTrace();
}
response.sendRedirect("../customer/list.jsp");
%>
```

문제 3. 기능 구현(bean/DBCP)

13

```
public class Book {
    private int bookId;
    private String bookName;
    private String publisher;
    private int price;

    public int getBookId() {
        return bookId;
    }
    public void setBookId(int bookId) {
        this.bookId = bookId;
    }
    public String getBookName() {
        return bookName;
    }
    public void setBookName(String bookName) {
        this.bookName = bookName;
    }
    public String getPublisher() {
        return publisher;
    }
    public void setPublisher(String publisher) {
        this.publisher = publisher;
    }
    public int getPrice() {
        return price;
    }
    public void setPrice(int price) {
        this.price = price;
    }
}
```

```
public class Customer {
    private int custId;
    private String name;
    private String address;
    private String phone;

    public int getCustId() {
        return custId;
    }
    public void setCustId(int custId) {
        this.custId = custId;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public String getPhone() {
        return phone;
    }
    public void setPhone(String phone) {
        this.phone = phone;
    }
}
```

```
<!-- java1_bookstore -->
<Resource
    name="dbcp_java1_bookstore"
    auth="Container"
    type="javax.sql.DataSource"
    driverClassName="com.mysql.cj.jdbc.Driver"
    url="jdbc:mysql://127.0.0.1:3306/java1_bookstore"
    username="root"
    password="1234"
    maxTotal="8"
    maxIdle="8"
    maxWaitMillis="3000"
/>
</Context>
```

```
public class DBCP {

    private static DataSource ds = null;

    public static Connection getConnection() throws NamingException, SQLException {

        if(ds == null) {
            ds = (DataSource)new InitialContext().lookup("java:comp/env/dbcp_java1_bookstore");
        }
        return ds.getConnection(); // 커넥션 가져오기
    }

}
```


문제 4. 실행(Book)

14

BookStore

[도서목록](#) [고객목록](#)

도서목록

[처음으로](#) [도서등록](#)

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	수정 삭제
2	축구하는 여자	나무수	13000	수정 삭제
3	축구의 이해	대한미디어	22000	수정 삭제
4	골프 바이블	대한미디어	35000	수정 삭제
5	피겨 교본	굿스포츠	8000	수정 삭제
6	역도 단계별기술	굿스포츠	6000	수정 삭제
7	야구의 추억	이상미디어	20000	수정 삭제
8	야구를 부탁해	이상미디어	13000	수정 삭제
9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Champions	Pearson	13000	수정 삭제

도서등록

[처음으로](#) [도서목록](#)

도서명	<input type="text" value="심판"/>
출판사	<input type="text" value="카프카"/>
가격	<input type="text" value="23000"/>
<input type="button" value="등록"/>	

8	야구를 부탁해	이상미디어	13000	수정 삭제
9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Champions	Pearson	13000	수정 삭제
11	심판	카프카	23000	수정 삭제

도서수정

[처음으로](#) [도서목록](#)

도서번호	<input type="text" value="11"/>
도서명	<input type="text" value="Judgement"/>
출판사	<input type="text" value="Kafka"/>
가격	<input type="text" value="30000"/>
<input type="button" value="수정"/>	

9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Champions	Pearson	13000	수정 삭제
11	Judgement	Kafka	30000	수정 삭제

도서목록

[처음으로](#) [도서등록](#)

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	수정 삭제
2	축구하는 여자	나무수	13000	수정 삭제
3	축구의 이해	대한미디어	22000	수정 삭제
4	골프 바이블	대한미디어	35000	수정 삭제
5	피겨 교본	굿스포츠	8000	수정 삭제
6	역도 단계별기술	굿스포츠	6000	수정 삭제
7	야구의 추억	이상미디어	20000	수정 삭제
8	야구를 부탁해	이상미디어	13000	수정 삭제
9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Champions	Pearson	13000	수정 삭제

문제 4. 실행(Customer)

15

BookStore

[도서목록](#) [고객목록](#)

고객목록

[처음으로](#) [고객등록](#)

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스터	000-5000-0001	수정 삭제
2	김연아	대한민국 서울	000-6000-0001	수정 삭제
3	장미란	대한민국 강원도	000-7000-0001	수정 삭제
4	추신수	미국 클리블랜드	000-8000-0001	수정 삭제
5	박세리	대한민국 부산	null	수정 삭제
6	이단계사	광주	010-1234	수정 삭제
7	큐일삼	큐일도	001	수정 삭제
9	김유신	제주	010-1234	수정 삭제

고객등록

[처음으로](#) [고객목록](#)

고객명	홍길동
주소	대한민국 경주
휴대폰	010-0000-1234
<input type="button" value="등록"/>	

/	큐일삼	큐일도	001	수정 삭제
9	김유신	제주	010-1234	수정 삭제
10	홍길동	대한민국 경주	010-0000-1234	수정 삭제

고객수정

[처음으로](#) [고객목록](#)

고객번호	10
고객명	전우치
주소	미국 뉴욕
휴대폰	010-4578-0000
<input type="button" value="수정"/>	

9	김유신	제주	010-1234	수정 삭제
10	전우치	미국 뉴욕	010-4578-0000	수정 삭제

고객목록

[처음으로](#) [고객등록](#)

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스터	000-5000-0001	수정 삭제
2	김연아	대한민국 서울	000-6000-0001	수정 삭제
3	장미란	대한민국 강원도	000-7000-0001	수정 삭제
4	추신수	미국 클리블랜드	000-8000-0001	수정 삭제
5	박세리	대한민국 부산	null	수정 삭제
6	이단계사	광주	010-1234	수정 삭제
7	큐일삼	큐일도	001	수정 삭제
9	김유신	제주	010-1234	수정 삭제