



SQL 응용

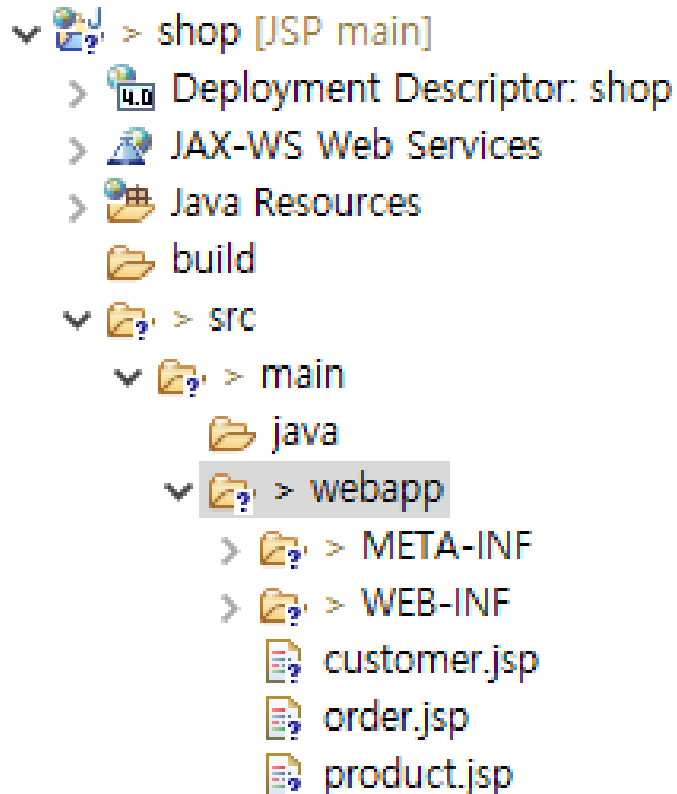
데이터베이스 구현

심규영

목차

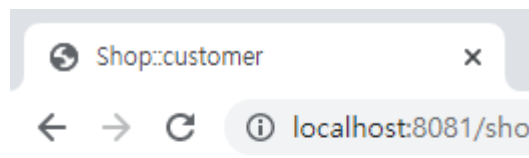
- 1) 프로젝트 생성 및 구성
- 2) 화면 구현
- 3) 기능 구현
- 4) 실행

문제 1) 프로젝트 생성 및 구현



- 프로젝트명 : shop
- WAS : Tomcat 9
- DB : Java2_bookstore
- 개발도구 : Eclipse, Workbench

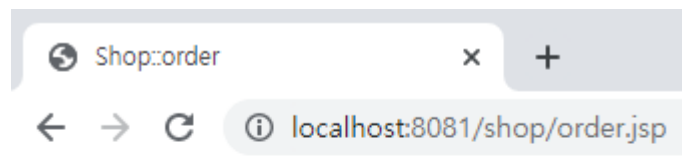
문제2) 화면 구현



고객목록

[고객목록](#) [주문목록](#) [상품목록](#)

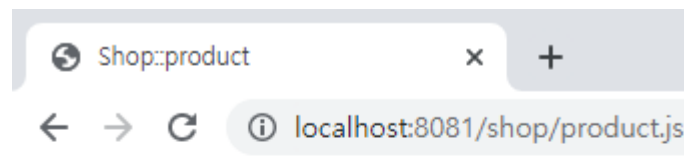
아이디	이름	휴대폰	주소	가입일



주문목록

[고객목록](#) [주문목록](#) [상품목록](#)

주문번호	주문자	주문상품	주문수량	주문일



상품목록

[고객목록](#) [주문목록](#) [상품목록](#)

상품번호	상품명	재고량	가격	제조사	주문
					주문

문제3) 기능구현

```
<Resource
auth="Container"
driverClassName="com.mysql.cj.jdbc.Driver"
maxIdle="16" maxTotal="16"
maxWaitMillis="3000" name="dbcp_java2_shop"
password="1234" type="javax.sql.DataSource"
url="jdbc:mysql://127.0.0.1:3306/java2_shop"
username="root"/>
</GlobalNamingResources>

<!-- java2_shop 커넥션 풀 자원 설정 -->
<ResourceLink
    global="dbcp_java2_shop"
    name="dbcp_java2_shop"
    type="javax.sql.DataSource"
/>
</Context>
```

```
package kr.co.shop.config;

import java.sql.Connection;
import java.sql.SQLException;

import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.sql.DataSource;

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_java2_shop");
        }
        return ds.getConnection();
    }
}
```

1. Java2_shop 커넥션 풀 자원 설정
2. DBCP.java 생성

문제3) 기능구현

1. javaBean 생성
2. orderBean은 조건에 따라 변형 생성

```
1 package kr.co.shop.bean;
2
3 public class customerBean {
4     private String custid;
5     private String name;
6     private String hp;
7     private String addr;
8     private String rdate;
9     public String getCustid() {
10         return custid;
11     }
12     public void setCustid(String custid) {
13         this.custid = custid;
14     }
15     public String getName() {
16         return name;
17     }
18     public void setName(String name) {
19         this.name = name;
20     }
21     public String getHp() {
22         return hp;
23     }
24     public void setHp(String hp) {
25         this.hp = hp;
26     }
27     public String getAddr() {
28         return addr;
29     }
30     public void setAddr(String addr) {
31         this.addr = addr;
32     }
33     public String getRdate() {
34         return rdate;
35     }
36     public void setRdate(String rdate) {
37         this.rdate = rdate;
38     }
39 }
```

```
1 package kr.co.shop.bean;
2
3 public class orderBean {
4     private int orderNo;
5     private String orderName;
6     private String productName;
7     private int orderCount;
8     private String orderDate;
9     public int getOrderNo() {
10         return orderNo;
11     }
12     public void setOrderNo(int orderNo) {
13         this.orderNo = orderNo;
14     }
15     public String getOrderName() {
16         return orderName;
17     }
18     public void setOrderName(String orderName) {
19         this.orderName = orderName;
20     }
21     public String getProductName() {
22         return productName;
23     }
24     public void setProductName(String productName) {
25         this.productName = productName;
26     }
27     public int getOrderCount() {
28         return orderCount;
29     }
30     public void setOrderCount(int orderCount) {
31         this.orderCount = orderCount;
32     }
33     public String getOrderDate() {
34         return orderDate;
35     }
36     public void setOrderDate(String orderDate) {
37         this.orderDate = orderDate;
38     }
39 }
```

```
1 package kr.co.shop.bean;
2
3 public class productBean {
4     private int prodNo;
5     private String prodName;
6     private int stock;
7     private int price;
8     private String company;
9     public int getProdNo() {
10         return prodNo;
11     }
12     public void setProdNo(int prodNo) {
13         this.prodNo = prodNo;
14     }
15     public String getProdName() {
16         return prodName;
17     }
18     public void setProdName(String prodName) {
19         this.prodName = prodName;
20     }
21     public int getStock() {
22         return stock;
23     }
24     public void setStock(int stock) {
25         this.stock = stock;
26     }
27     public int getPrice() {
28         return price;
29     }
30     public void setPrice(int price) {
31         this.price = price;
32     }
33     public String getCompany() {
34         return company;
35     }
36     public void setCompany(String company) {
37         this.company = company;
38     }
39 }
```

문제3) 기능구현

1. customerDAO 생성 및 고객 리스트 불러오기 구현
2. OrderDAO 생성 및 주문 리스트 불러오기 구현

```
29 public List<customerBean> CustomerListDAO() {  
30     List<customerBean> cbs = new ArrayList<>();  
31  
32     try {  
33         stmt = conn.createStatement();  
34         rs = stmt.executeQuery("select * from `customer`");  
35         while(rs.next()) {  
36             customerBean cb = new customerBean();  
37             cb.setCustid(rs.getString(1));  
38             cb.setName(rs.getString(2));  
39             cb.setHp(rs.getString(3));  
40             cb.setAddr(rs.getString(4));  
41             cb.setRdate(rs.getString(5));  
42             cbs.add(cb);  
43         }  
44     } catch (Exception e) {  
45         System.out.println("고객 리스트 불러오기 오류");  
46         e.printStackTrace();  
47     }  
48  
49     return cbs;  
50 }
```

```
29 public List<orderBean> ShopOrderListDAO(){  
30     List<orderBean> obs = new ArrayList<>();  
31  
32     try {  
33         stmt = conn.createStatement();  
34         rs = stmt.executeQuery(  
35             "select "  
36             + "a.`orderNo`,`"  
37             + "b.`name`,`"  
38             + "c.`prodName`,`"  
39             + "a.`orderCount`,`"  
40             + "a.`orderDate` "  
41             + "from `order` as a"  
42             + " join `customer` as b on a.orderId = b.custid"  
43             + " join `product` as c on a.orderProduct = c.prodNo");  
44         while(rs.next()) {  
45             orderBean ob = new orderBean();  
46             ob.setOrderNo(rs.getInt(1));  
47             ob.setOrderName(rs.getString(2));  
48             ob.setProductName(rs.getString(3));  
49             ob.setOrderCount(rs.getInt(4));  
50             ob.setOrderDate(rs.getString(5));  
51             obs.add(ob);  
52         }  
53     } catch (Exception e) {  
54         System.out.println("주문 리스트 불러오기 오류");  
55         e.printStackTrace();  
56     }  
57  
58     return obs;  
59 }
```

문제3) 기능구현

1. productDAO 생성 및 상품리스트 기능 구현
2. Order.js 생성, 주문 버튼 누를시 주문하기 테이블 생성 구현

```
30 public List<productBean> ShopProductListDAO(){
31     List<productBean> pbs = new ArrayList<>();
32
33     try {
34         stmt = conn.createStatement();
35         rs = stmt.executeQuery("select * from `product`");
36         while(rs.next()) {
37             productBean pb = new productBean();
38             pb.setProdNo(rs.getInt(1));
39             pb.setProdName(rs.getString(2));
40             pb.setStock(rs.getInt(3));
41             pb.setPrice(rs.getInt(4));
42             pb.setCompany(rs.getString(5));
43             pbs.add(pb);
44         }
45     } catch (Exception e) {
46         System.out.println("고객 리스트 불러오기 오류");
47         e.printStackTrace();
48     }
49
50     return pbs;
51 }
```

```
4 function order (productList) {
5     const prodNo = productList.eq(0).text();
6
7     $('section').empty();
8     $('nav').empty().append("<h4>주문하기</h4>");
9
10    let table = "<table border='1'>"
11                + "<tr>"
12                  + "<td>상품번호</td>"
13                  + "<td><input type='text' name='orderProduct' "
14                    + "readonly='readonly' value='"+prodNo+"'></td>"
15                  + "</tr>"
16                  + "<tr>"
17                    + "<td>수량</td>"
18                    + "<td><input type='text' name='orderCount'></td>"
19                  + "</tr>"
20                  + "<tr>"
21                    + "<td>주문자</td>"
22                    + "<td><input type='text' name='orderId'></td>"
23                  + "</tr>"
24                  + "<tr>"
25                    + "<td colspan='2' align='right'>"
26                    + "<input type='submit' id='btnProductOrder' value='주문하기'></td>"
27                  + "</tr>"
28                + "</table>";
29    $('section').append(table);
30 }
```

```
18 <script>
19     $(document).ready(()=>{
20         $(document).on('click', '.btnOrder', function(e){
21             e.preventDefault();
22             const productList = $(this).parent().parent().children('td');
23             order(productList);
24         });
25     });
```


문제3) 기능구현

1. 주문 기능 구현
2. ShopOrderDAO에 주문 등록 기능 구현

```
--
26 // 주문 하기
27 $(document).on('click', '#btnProductOrder', function(){
28     const orderProduct = $('input[name=orderProduct]').val();
29     const orderCount = $('input[name=orderCount]').val();
30     const orderId = $('input[name=orderId]').val();
31
32     const jsonData = {
33         "orderProduct":orderProduct,
34         "orderCount":orderCount,
35         "orderId":orderId
36     };
37
38     $.ajax({
39         url: './data/order.jsp',
40         method: 'post',
41         data: jsonData,
42         dataType: 'json',
43         success: (data)=>{
44             if(data.result == 1) {
45                 alert('주문성공!');
46             } else {
47                 alert("주문실패!");
48             }
49         }
50     });
51
52     //주문 끝나고 테이블 닫기
53     $('section').empty();
54     $('nav').empty();
55 });
56
57 </script>
```

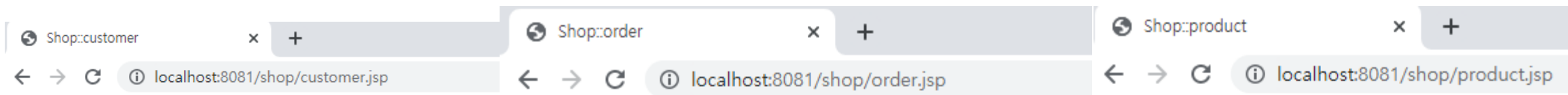
```
62 public int ShopOrderAdd(String orderId, String orderProduct, String orderCount) {
63     int result = 0;
64
65     try {
66         pstmt = conn.prepareStatement(
67             "insert into `order` (`orderId`, `orderProduct`, `orderCount`"
68             + ", `orderDate`) values (?, ?, ?, NOW())");
69         pstmt.setString(1, orderId);
70         pstmt.setString(2, orderProduct);
71         pstmt.setString(3, orderCount);
72
73         result = pstmt.executeUpdate();
74     } catch (Exception e) {
75         System.out.println("주문 등록 오류!");
76         e.printStackTrace();
77     }
78
79     return result;
80 }
```

문제3) 기능구현

```
1 <%@page import="com.google.gson.JsonObject"%>
2 <%@page import="kr.co.shop.DAO.ShopOrderDAO"%>
3 <%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
4 <%
5     request.setCharacterEncoding("utf-8");
6     String orderProduct = request.getParameter("orderProduct");
7     String orderCount   = request.getParameter("orderCount");
8     String orderId      = request.getParameter("orderId");
9
10    int result = 0;
11
12    ShopOrderDAO sodao = new ShopOrderDAO();
13    result = sodao.ShopOrderAdd(orderId, orderProduct, orderCount);
14    sodao.close();
15
16    JsonObject json = new JsonObject();
17    json.addProperty("result", result);
18
19    String jsonData = json.toString();
20
21    out.print(jsonData);
22 %>
```

문제4) 실행

1. 고객 목록
2. 주문 목록
3. 상품 목록



고객목록

[고객목록](#) [주문목록](#) [상품목록](#)

아이디	이름	휴대폰	주소	가입일
c101	김유신	010-1234-1001	김해시 봉황동	2022-01-01
c102	김춘추	010-1234-1002	경주시 보문동	2022-01-02
c103	장보고	010-1234-1003	완도군 청산면	2022-01-03
c104	강감찬	010-1234-1004	서울시 마포구	2022-01-04
c105	이성계	null	null	2022-01-05
c106	정철	010-1234-1006	경기도 용인시	2022-01-06
c107	허준	null	null	2022-01-07
c108	이순신	010-1234-1008	서울시 영등포구	2022-01-08
c109	송상현	010-1234-1009	부산시 동래구	2022-01-09
c110	정약용	010-1234-1010	경기도 광주시	2022-01-10

주문목록

[고객목록](#) [주문목록](#) [상품목록](#)

주문번호	주문자	주문상품	주문수량	주문일
1	김춘추	포카칩	2	2022-07-01 13:15:10
2	김유신	새우깡	10	2022-10-25 17:33:54
3	이순신	새우깡	1	2022-07-01 17:23:18
4	송상현	마카렛트	5	2022-07-02 10:46:36
5	김춘추	초코파이	1	2022-07-03 09:15:37
6	김유신	새우깡	10	2022-10-25 17:33:54
7	정약용	새우깡	2	2022-07-03 16:55:36
8	강감찬	초코파이	4	2022-07-04 14:23:23
9	김춘추	새우깡	3	2022-07-04 21:54:34
10	허준	마카렛트	1	2022-07-05 14:21:03
11	김유신	새우깡	10	2022-10-25 17:38:58

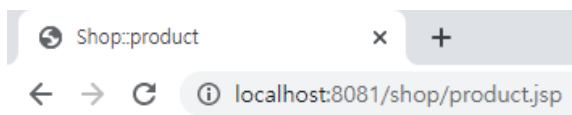
상품목록

[고객목록](#) [주문목록](#) [상품목록](#)

상품번호	상품명	재고량	가격	제조사	주문
1	새우깡	5000	1500	농심	주문
2	초코파이	2500	2500	오리온	주문
3	포카칩	3600	1700	오리온	주문
4	양파링	1250	1800	농심	주문
5	조리퐁	2200	0	크라운	주문
6	마카렛트	3500	3500	롯데	주문
7	뿌셔뿌셔	1650	1200	오뚜기	주문

1. 주문 등록 성공 화면

문제4) 실행



상품목록

[고객목록](#) [주문목록](#) [상품목록](#)

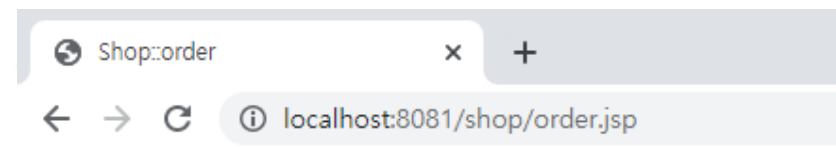
상품번호	상품명	재고량	가격	제조사	주문
1	새우깡	5000	1500	농심	주문
2	초코파이	2500	2500	오리온	주문
3	포카칩	3600	1700	오리온	주문
4	양파링	1250	1800	농심	주문
5	쪄리퐁	2200	0	크라운	주문
6	마카렛트	3500	3500	롯데	주문
7	뿌서뿌서	1650	1200	오뚜기	주문

주문하기

상품번호	2
수량	20
주문자	c102
<input type="button" value="주문하기"/>	

localhost:8081 내용:
주문성공!

확인



주문목록

[고객목록](#) [주문목록](#) [상품목록](#)

주문번호	주문자	주문상품	주문수량	주문일
1	김춘추	포카칩	2	2022-07-01 13:15:10
2	김유신	새우깡	10	2022-10-25 17:33:54
3	이순신	새우깡	1	2022-07-01 17:23:18
4	송상현	마카렛트	5	2022-07-02 10:46:36
5	김춘추	초코파이	1	2022-07-03 09:15:37
6	김유신	새우깡	10	2022-10-25 17:33:54
7	정약용	새우깡	2	2022-07-03 16:55:36
8	강감찬	초코파이	4	2022-07-04 14:23:23
9	김춘추	새우깡	3	2022-07-04 21:54:34
10	허준	마카렛트	1	2022-07-05 14:21:03
11	김유신	새우깡	10	2022-10-25 17:38:58
12	김춘추	초코파이	20	2022-10-25 17:41:17