

SQL 응용

서정현

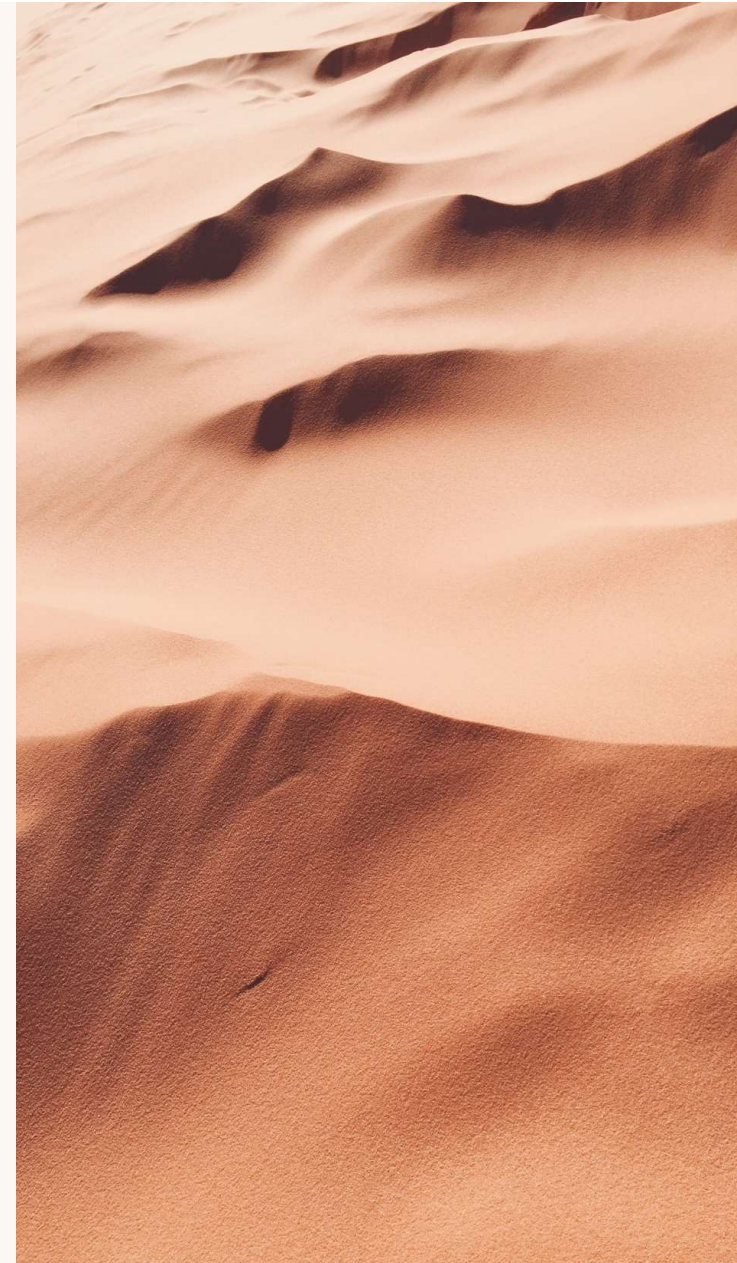
목차 A table of contents.

1.문제 1 – 프로젝트 생성 및 구성

2.문제 2 – 화면 구현

3.문제 3 – 기능 구현

4.문제 4 – 실행



1. 문제 1 – 프로젝트 생성 및 구성

1. 프로젝트명 : Shop

2. WAS : Tomcat 9

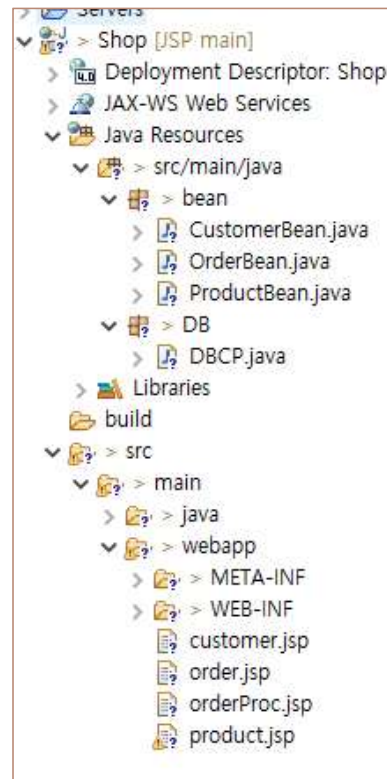
3. DB : java2_bookstore

4. 개발 도구 : Eclipse, workbench

2 > Tomcat v9.0 Server at localhost [Started, Restart]



1



3

custid	name	hp	addr	rdate
c101	김유신	010-1234-1001	김해시 통활동	2022-01-01
c102	김준주	010-1234-1002	경주시 보문동	2022-01-02
c103	장보고	010-1234-1003	완도군 정산면	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	정약물	010-1234-1010	경기도 광주시	2022-01-10

orderNo	orderId	orderProduct	orderCount	orderDate
1	c102	3	2	2022-07-01 13:15:10
2	c101	4	1	2022-07-01 14:16:11
3	c108	1	1	2022-07-01 17:23:18
4	c109	6	5	2022-07-02 10:46:36
5	c102	2	1	2022-07-03 09:15:37
6	c101	7	3	2022-07-03 12:35:12
7	c110	1	2	2022-07-03 16:55:36
8	c104	2	4	2022-07-04 14:23:23
9	c102		3	2022-07-04 21:54:34
10	c107	6	1	2022-07-05 14:21:03

prodNo	prodName	stock	price	company
1	새우장	4,960	1,500	농심
2	초코파이	2,500	2,500	오리온
3	포카칩	3,600	1,700	오리온
4	양파링	1,250	1,800	농심
5	조리품	2,200	(NULL)	크라온
6	마카렛트	3,500	3,500	롯데
7	뿌셔뿌셔	1,650	1,200	오뚜기

2. 문제 2 – 화면 구현

Customer(고객)

고객 목록				
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

order(주문)

주문 목록				
고객목록 주문목록 상품목록				
주문번호	주문자	주문상품	주문수량	주문일
1	김춘추	포카칩	2	2022-07-01 13:15:10
2	김유신	양파링	1	2022-07-01 14:16:11
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	뿌셔뿌셔	3	2022-07-03 12:35:12
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

product(상품)

상품 목록					
고객목록 주문목록 상품목록					
상품번호	상품명	재고량	가격	제조사	주문
1	새우깡	4960	1500	농심	주문
2	초코파이	2500	2500	오리온	주문
3	포카칩	3600	1700	오리온	주문
4	양파링	1250	1800	농심	주문
5	쥬리퐁	2200	0	크라운	주문
6	마카렛트	3500	3500	롯데	주문
7	뿌셔뿌셔	1650	1200	오뚜기	주문

3. 문제 3 – 기능 구현 – customer 목록 출력

```
<%
// 고객 정보를 가져오는 코드

List<CustomerBean> customers = null;

//데이터베이스 작업
try{
    // 1, 2단계 - DB 접속
    Connection con = DBCP.getConnection();

    // 3단계 - SQL실행 객체 생성
    Statement stmt = con.createStatement();

    // 4단계 - SQL실행
    ResultSet rs = stmt.executeQuery("SELECT * FROM `customer`");

    // 고객들을 저장하기위한 List 객체 생성
    customers = new ArrayList<>();

    // 5단계 - SQL결과 처리
    while(rs.next()){
        CustomerBean cb = new CustomerBean();
        cb.setCustId(rs.getString(1));
        cb.setName(rs.getString(2));
        cb.setHp(rs.getString(3));
        cb.setAddr(rs.getString(4));
        cb.setRdate(rs.getString(5));

        customers.add(cb);
    }

    // 6단계 - 연결 해제
    con.close();
    stmt.close();
    rs.close();

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

스크립트릿(프로그램 코드 영역)

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Shop::customer</title>
    </head>
    <body>
        <h3>고객 목록</h3>
        <nav>
            <a href="/customer.jsp">고객목록</a>
            <a href="/order.jsp">주문목록</a>
            <a href="/product.jsp">상품목록</a>
        </nav>
        <table border="1">
            <tr>
                <th>아이디</th>
                <th>이름</th>
                <th>휴대폰</th>
                <th>주소</th>
                <th>가입일</th>
            </tr>
            <%
                for(CustomerBean cb : customers){
            %>
            <tr>
                <td><%= cb.getCustId() %></td>
                <td><%= cb.getName() %></td>
                <td><%= cb.getHp() %></td>
                <td><%= cb.getAddr() %></td>
                <td><%= cb.getRdate() %></td>
            </tr>
            <%
                }
            %>
        </table>

    </body>
</html>
```

뷰 영역

3. 문제 3 – 기능 구현 – order 목록 출력

```
<%
// order 정보를 가져오는 코드
List<OrderBean> orders = null;

//데이터베이스 작업
try{
    // 1, 2단계 - DB 접속
    Connection con = DBCP.getConnection();

    // 3단계 - SQL실행 객체 생성
    Statement stmt = con.createStatement();
    String sql = "SELECT o.orderNo, c.name, p.prodName, o.orderCount, o.orderDate "
        + " FROM `order` as o"
        + " join `customer` as c on o.orderid = c.custid"
        + " join `product` as p on o.orderProduct = p.prodNo";

    // 4단계 - SQL실행
    ResultSet rs = stmt.executeQuery(sql);

    // 주문정보를 저장하기위한 List 객체 생성
    orders = new ArrayList<>();

    // 5단계 - SQL결과 처리
    while(rs.next()){
        OrderBean ob = new OrderBean();
        ob.setOrderNo(rs.getInt(1));
        ob.setOrderName(rs.getString(2));
        ob.setOrderProduct(rs.getString(3));
        ob.setOrderCount(rs.getInt(4));
        ob.setOrderDate(rs.getString(5));

        orders.add(ob);
    }

    // 6단계 - 연결 해제
    con.close();
    stmt.close();
    rs.close();

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

스크립트릿(프로그램 코드 영역)

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Shop::order</title>
    </head>
    <body>
        <h3>주문 목록</h3>
        <nav>
            <a href="./customer.jsp">고객목록</a>
            <a href="./order.jsp">주문목록</a>
            <a href="./product.jsp">상품목록</a>
        </nav>
        <table border="1">
            <tr>
                <th>주문번호</th>
                <th>주문자</th>
                <th>주문상품</th>
                <th>주문수량</th>
                <th>주문일</th>
            </tr>
            <%
                for(OrderBean ob : orders){
            %>
            <tr>
                <td><%= ob.getOrderNo() %></td>
                <td><%= ob.getOrderName() %></td>
                <td><%= ob.getOrderProduct() %></td>
                <td><%= ob.getOrderCount() %></td>
                <td><%= ob.getOrderDate() %></td>
            </tr>
            <%
                }
            %>
        </table>

    </body>
</html>
```

뷰 영역

3. 문제 3 – 기능 구현 – product 목록 출력

```
<%
// product 정보를 가져오는 코드
List<ProductBean> products = null;

//데이터베이스 작업
try{
    // 1, 2단계 - DB 접속
    Connection con = DBCP.getConnection();

    // 3단계 - SQL실행 객체 생성
    Statement stmt = con.createStatement();

    // 4단계 - SQL실행
    ResultSet rs = stmt.executeQuery("SELECT * FROM `product`");

    // 상품정보들을 저장하기위한 List 객체 생성
    products = new ArrayList<>();

    // 5단계 - SQL결과 처리
    while(rs.next()){
        ProductBean pb = new ProductBean();
        pb.setProdNo(rs.getInt(1));
        pb.setProdName(rs.getString(2));
        pb.setStock(rs.getInt(3));
        pb.setPrice(rs.getInt(4));
        pb.setCompany(rs.getString(5));

        products.add(pb);
    }

    // 6단계 - 연결 해제
    con.close();
    stmt.close();
    rs.close();

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

스크립트릿(프로그램 코드 영역)

```
<body>
<h3>상품 목록</h3>
<nav>
    <a href="./customer.jsp">고객목록</a>
    <a href="./order.jsp">주문목록</a>
    <a href="./product.jsp">상품목록</a>
</nav>
<table border="1">
    <tr>
        <th>상품번호</th>
        <th>상품명</th>
        <th>재고량</th>
        <th>가격</th>
        <th>제조사</th>
        <th>주문</th>
    </tr>
    <%
    for(ProductBean pb : products){
    %>
    <tr>
        <td><%= pb.getProdNo() %></td>
        <td><%= pb.getProdName() %></td>
        <td><%= pb.getStock() %></td>
        <td><%= pb.getPrice() %></td>
        <td><%= pb.getCompany() %></td>
        <td><button class="btnOrder">주문</button> </td>
    </tr>
    <%
    }
    %>
</table></br>
</body>
```

뷰 영역

3. 문제 3 – 기능 구현 – product.jsp

```
<script>
$(function () {
    $(''.btnOrder').click(function() {
        $('form').remove(); // 주문 버튼을 눌렀을때 중복해서 출력을 방지하기위해 삭제 후 한번하고 다시 출력한다.
        let prodNum = $(this).parent().parent().children("td:eq(0)").text();
        let tags = "<form>";
        tags += "<table border='1'>";
        tags += "<tr>";
        tags += "<td>상품번호</td>";
        tags += "<td><input type='text' name='prodNo' value='" + prodNum + "'></td>";
        tags += "</tr>";
        tags += "<tr>";
        tags += "<td>수량</td>";
        tags += "<td><input type='text' name='orderCount'></td>";
        tags += "</tr>";
        tags += "<tr>";
        tags += "<td>주문자</td>";
        tags += "<td><input type='text' name='custId'></td>";
        tags += "</tr>";
        tags += "<tr>";
        tags += "<td colspan='2' align='right'><input type='submit' value='주문하기'></td>";
        tags += "</tr>";
        tags += "</table>";
        tags += "</form>";

        $('body').append(tags); // tags안에 있는 태그들을 body에 append한다.
    });

    (document).on('click', 'input[type=submit]', function (e) {
        e.preventDefault(); // 원래 이벤트 기능을 제거

        // input에 입력된 값을 가져온다.
        let prodNo = $('input[name=prodNo]').val();
        let orderCount = $('input[name=orderCount]').val();
        let custId = $('input[name=custId]').val();

        // json 형태로 변환한다.
        let jsonData = {
            "prodNo": prodNo,
            "orderCount": orderCount,
            "custId": custId
        }
    });
});
</script>
```

```
// AJAX를 이용해 비동기 방식으로 orderProc으로 JSON데이터를 전송한다.
$.ajax({
    url: './orderProc.jsp',
    method: 'post',
    data: jsonData,
    dataType: 'json',
    success: function (data) {

        if(data.result == 1){
            alert('주문완료!');
            $('table:eq(1)').remove();
            location.reload();
        } else {
            alert('주문실패!');
        }
    }
});
});
</script>
```

※주문 버튼을 누르면 입력 창이 나타나게 되고 주문하기 버튼을 누르면 입력 된 값들을 JSON데이터로 변환하여 orderProc.jsp로 전송한다. orderProc.jsp로 부터 반환 받은 JSON데이터 값을 통해 1이면 주문 성공 0이면 주문 실패창을 띄운다.

3. 문제 3 – 기능 구현 – orderProc.jsp

```
<%
//product.jsp로 부터 전송받은 데이터 수신
request.setCharacterEncoding("utf-8");
String custId = request.getParameter("custId");
String orderProduct = request.getParameter("prodNo");
String orderCount = request.getParameter("orderCount");

// 전송 결과 성공/실패 여부를 알기위한 변수 생성
int result = 1;

// 데이터베이스 작업
try{
    // 1, 2단계 - DB 접속
    Connection con = DBCP.getConnection();

    String sql = "INSERT INTO `order` (`orderId`, `orderProduct`, `orderCount`, `orderDate`) "
        + " VALUES(?,?,?,NOW())";

    // 3단계 - SQL실행 객체 생성
    PreparedStatement psmt = con.prepareStatement(sql);
    psmt.setString(1, custId);
    psmt.setString(2, orderProduct);
    psmt.setString(3, orderCount);

    // 4단계 - SQL실행
    result = psmt.executeUpdate();

    // 6단계 - 연결 해제
    con.close();
    psmt.close();

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

// JSON데이터로 변환하여 product.jsp로 반환한다.
JsonObject json = new JsonObject();
json.addProperty("result", result);
String jsonData = json.toString();

out.print(jsonData);
%>
```

※product.jsp로 부터 받은 JSON 데이터를 수신해서 DB에 저장
을 한다. 성공 여부에 따라 결과 값을 result변수에 저장후
JSON데이터로 변환하고 product.jsp로 반환한다.

4. 문제 4 – 목록 실행 화면

Customer(고객)

고객 목록				
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

order(주문)

주문 목록				
고객목록 주문목록 상품목록				
주문번호	주문자	주문상품	주문수량	주문일
1	김춘추	포카칩	2	2022-07-01 13:15:10
2	김유신	양파링	1	2022-07-01 14:16:11
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	뿌셔뿌셔	3	2022-07-03 12:35:12
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

product(상품)

상품 목록					
고객목록 주문목록 상품목록					
상품번호	상품명	재고량	가격	제조사	주문
1	새우깡	4960	1500	농심	주문
2	초코파이	2500	2500	오리온	주문
3	포카칩	3600	1700	오리온	주문
4	양파링	1250	1800	농심	주문
5	쥬리퐁	2200	0	크라운	주문
6	마카렛트	3500	3500	롯데	주문
7	뿌셔뿌셔	1650	1200	오뚜기	주문

4. 문제 4 – 주문 실행 화면

상품 목록

[고객목록](#) [주문목록](#) [상품목록](#)

상품번호	상품명	재고량	가격	제조사	주문
1	새우깡	4960	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

상품 목록

[고객목록](#) [주문목록](#) [상품목록](#)

상품번호	상품명	재고량	가격	제조사	주문
1	새우깡	4960	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

상품번호

수량

주문자

localhost:8080/Shop/product.jsp

localhost:8080 내용:

주문원료:

상품 목록

[고객목록](#) [주문목록](#) [상품목록](#)

상품번호	상품명	재고량	가격	제조사	주문
1	새우깡	4960	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

상품번호

수량

주문자

주문 목록

[고객목록](#) [주문목록](#) [상품목록](#)

주문번호	주문자	주문상품	주문수량	주문일
1	김준추	포카칩	2	2022-07-01 13:15:10
2	김유신	양파링	1	2022-07-01 14:16:11
3	이순신	새우깡	1	2022-07-01 17:23:18
4	송상현	마카렛트	5	2022-07-02 10:46:36
5	김준추	초코파이	1	2022-07-03 09:15:37
6	김유신	뿌셔뿌셔	3	2022-07-03 12:35:12
7	정약용	새우깡	2	2022-07-03 16:55:36
8	강감찬	초코파이	4	2022-07-04 14:23:23
9	김준추	새우깡	3	2022-07-04 21:54:34
10	허준	마카렛트	1	2022-07-05 14:21:03
33	김유신	새우깡	10	2022-10-25 19:19:23

감사합니다