

#밀키트 서점

박지선



김동현



조민석



노용천



#목 차

- 팀 구성원 소개
- 프로젝트 소개
- 코드 설명
- 수행 소감

#Best Team

박지선

- DB연동 및 구축
- 코드 작성
- 코드 점검 및 정리

김동현

- DB연동 및 구축
- 코드 작성
- 코드 점검 및 정리

조민석

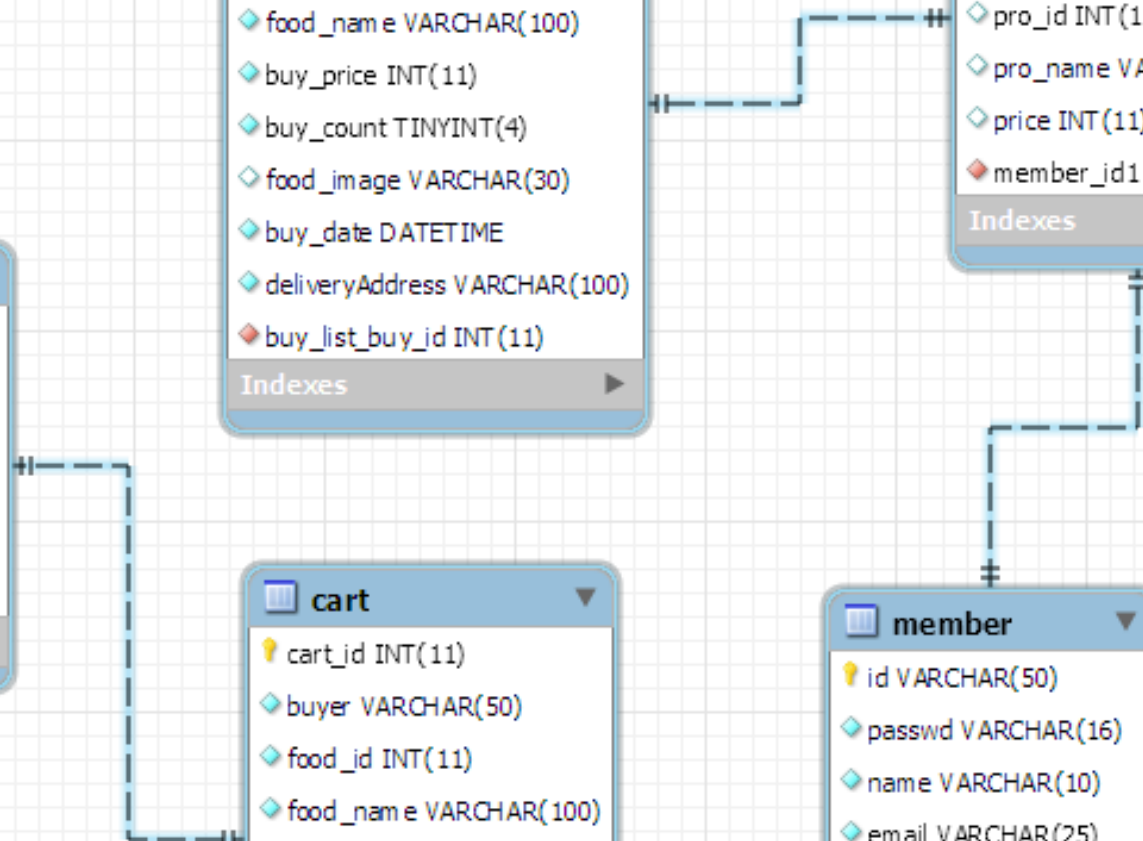
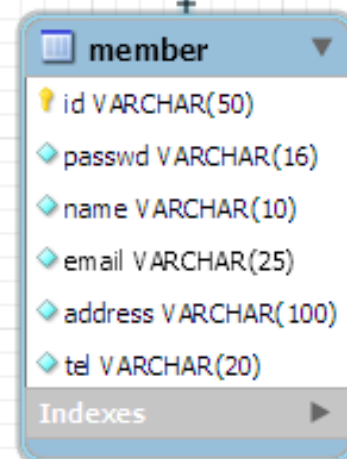
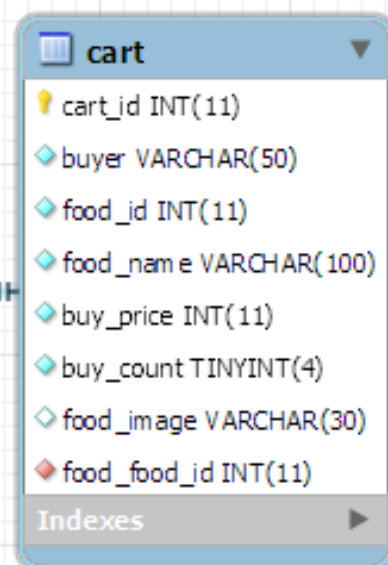
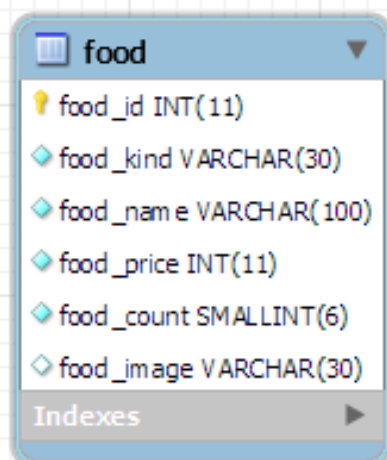
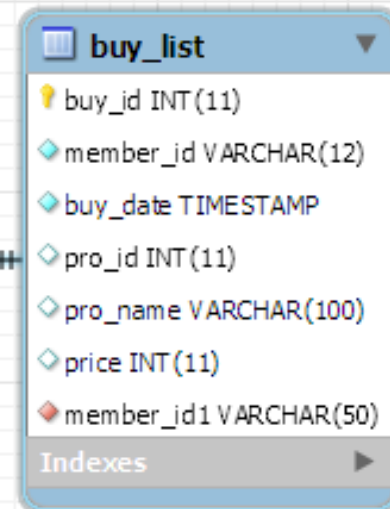
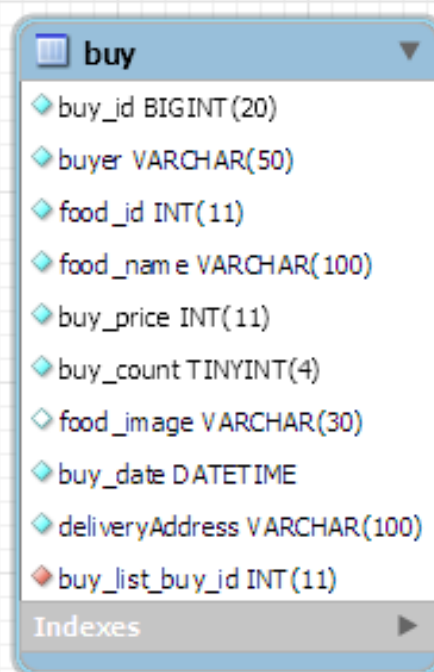
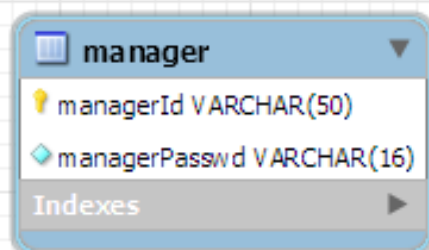
- DB연동 및 구축
- 코드 작성
- 코드 점검 및 정리

노용천

- DB연동 및 구축
- 코드 작성
- PPT 자료수집 및 작성

#프로젝트 소개

- Book Base
- Food Shop
- MemberShip
- Login
- Cart
- Buy List

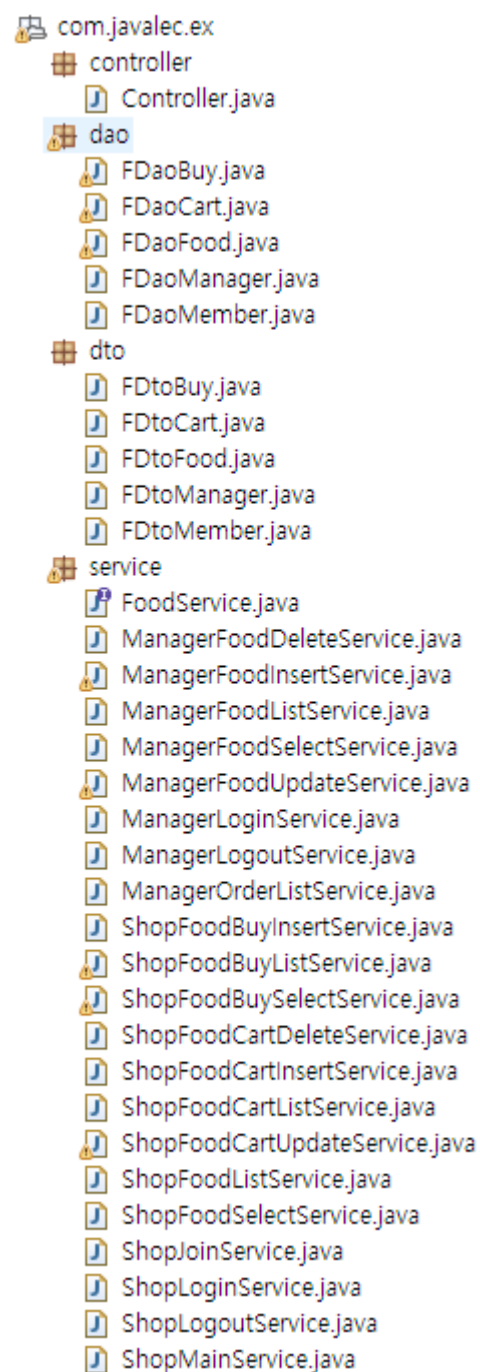


Project Explorer

변경 전

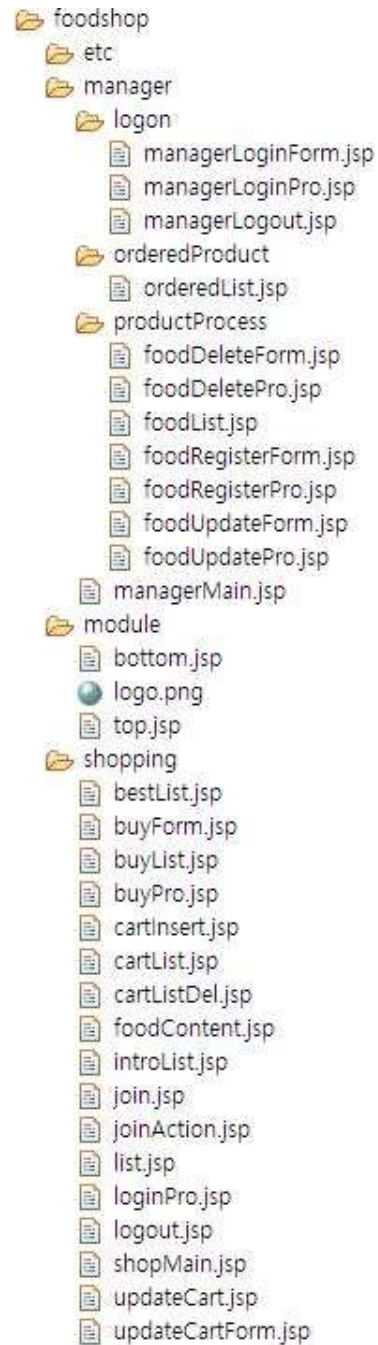


변경 후

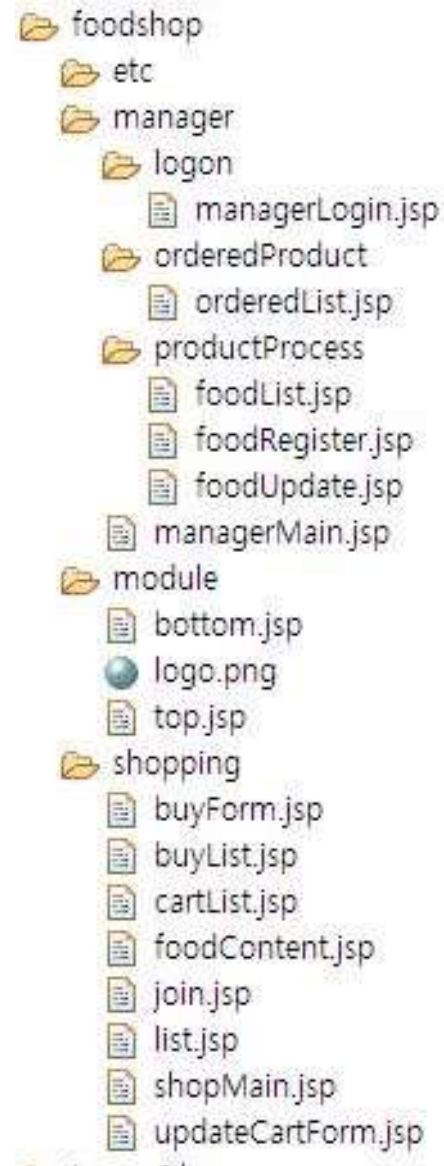


Project Explorer

변경 전



변경 후



Manager Controller

```
@WebServlet("*.do")
```

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    doAction(request, response);
}

protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    doGet(request, response);
}

protected void doAction(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    request.setCharacterEncoding("utf-8");
    response.setCharacterEncoding("utf-8");
    String viewPage = null;
    FoodService service = null;
    String uri = request.getRequestURI();
    String conPath = request.getContextPath();
    String com = uri.substring(conPath.length());
}
```


Manager Controller

```
if (com.equals("/foodshop/manager/managerLogin.do")) {
    viewPage = "/foodshop/manager/managerLogin.jsp"; // 로그인 페이지로 강제 이동
} else if (com.equals("/foodshop/manager/managerMain.do")) { // 로그인된 상황
    viewPage = "/foodshop/manager/managerMain.jsp";
    service = new ManagerLoginService();
    service.execute(request, response); // 관리자로그인
    try {
        if (session.getAttribute("managerId") == null)
            return;
    } catch (Exception e) {
        e.printStackTrace();
    }
} else if (com.equals("/foodshop/manager/managerLogout.do")) {
    System.out.println("로그아웃!");
    service = new ManagerLogoutService();
    service.execute(request, response); // 관리자로그아웃
    return;
} else if (com.equals("/foodshop/manager/productProcess/foodRegisterView.do")) {
    System.out.println("작성 페이지로 이동!");
    viewPage = "/foodshop/manager/productProcess/foodRegister.jsp";
} else if (com.equals("/foodshop/manager/productProcess/foodRegister.do")) {
    System.out.println("작성 작성 기능 시작!");
    service = new ManagerFoodInsertService();
    service.execute(request, response);
    viewPage = "/foodshop/manager/productProcess/foodList.do";
} else if (com.equals("/foodshop/manager/productProcess/foodList.do")) {
    System.out.println("음식 목록 출력!");
    viewPage = "/foodshop/manager/productProcess/foodList.jsp";
    System.out.println(viewPage);
    service = new ManagerFoodListService();
    service.execute(request, response);
} else if (com.equals("/foodshop/manager/productProcess/foodDelete.do")) {
    System.out.println("음식 삭제!");
    service = new ManagerFoodDeleteService();
    service.execute(request, response);
    viewPage = "/foodshop/manager/productProcess/foodList.do";
} else if (com.equals("/foodshop/manager/productProcess/foodUpdateView.do")) {
    System.out.println("음식 수정 페이지로 이동!");
    viewPage = "/foodshop/manager/productProcess/foodUpdate.jsp";
    service = new ManagerFoodSelectService();
    service.execute(request, response);
} else if (com.equals("/foodshop/manager/productProcess/foodUpdate.do")) {
    System.out.println("음식 수정!");
    service = new ManagerFoodUpdateService();
    service.execute(request, response);
    viewPage = "/foodshop/manager/productProcess/foodList.do";
} else if (com.equals("/foodshop/manager/orderedProduct/orderedList.do")) {
    System.out.println("구매된 상품관련 작업 전체구매목록 확인");
    service = new ManagerOrderListService();
    service.execute(request, response);
    viewPage = "/foodshop/manager/orderedProduct/orderedList.jsp";
}
}
```

Manager Main.jsp 변경 전

```
<%
String managerId = "";
try{
    managerId = (String)session.getAttribute("managerId");

    if(managerId==null || managerId.equals("")){
        response.sendRedirect("logon/managerLoginForm.jsp");
    }else{

%>
<html>
<head>
<title>상품관리</title>
</head>
<body>
<form method="post" action="Logon/managerLogout.jsp">
    <b>관리작업중.. </b><input type="submit" value="로그아웃">
</form>

<table>
<tr><td align="center" bgcolor="<%=bar%>">상품관련 작업</td></tr>
<tr><td>
<a href='productProcess/foodRegisterForm.jsp'>상품등록</a></td></tr>
<tr><td>
<a href='productProcess/foodList.jsp?food_kind=all'>상품수정/삭제</a>
</td></tr>
</table><br><br>

<table>
<tr><td align="center" bgcolor="<%=bar%>">구매된 상품관련 작업</td></tr>
<tr><td>
<a href='orderedProduct/orderedList.jsp'>전체구매목록 확인</a></td></tr>
</table>
```

Manager Controller.java 변경 후

```
if (com.contains("/manager/")) {
    if (com.equals("/foodshop/manager/managerLogin.do") == false
        && isLogin(request, response, session) == false) {

        response.sendRedirect(request.getContextPath() + "/foodshop/manager/managerLogin.do");
        return;
    }

    if (com.equals("/foodshop/manager/managerLogin.do") {
        viewPage = "/foodshop/manager/logon/managerLogin.jsp"; // 로그인 페이지로 강제 이동
    } else if (com.equals("/foodshop/manager/managerMain.do") { // 로그인인 된 상황
        viewPage = "/foodshop/manager/managerMain.jsp";
        service = new ManagerLoginService();
        service.execute(request, response); // 관리자로그인
    }
}
```

Manager Main.jsp 변경 후

```
<html>
<head>
<title>상품관리</title>
</head>
<body>
<form method="post" action="managerLogout.do">
    <b>관리작업중.. </b><input type="submit" value="로그아웃">
</form>

<table>
<tr><td align="center" bgcolor="<%=bar%>">상품관련 작업</td></tr>
<tr><td>
<a href='productProcess/foodRegisterView.do'>상품등록</a></td></tr>
<tr><td>
<a href='productProcess/foodList.do'>상품수정/삭제</a>
</td></tr>
</table><br><br>

<table>
<tr><td align="center" bgcolor="<%=bar%>">구매된 상품관련 작업</td></tr>
<tr><td>
<a href='orderedProduct/orderedList.do'>전체구매목록 확인</a></td></tr>
</table>
</body>
```


ShopFoodDBBean.java 변경 전

```
public int managerCheck(String id, String passwd)
throws Exception {
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs= null;
    String dbpasswd="";
    int x=-1;

    try {
        conn = getConnection();

        pstmt = conn.prepareStatement(
            "select managerPasswd from manager where managerId = ? ");
        pstmt.setString(1, id);

        rs= pstmt.executeQuery();

        if(rs.next()){
            dbpasswd= rs.getString("managerPasswd");
            if(dbpasswd.equals(passwd))
                x= 1; //인증 성공
            else
                x= 0; //비밀번호 틀림
        }else
            x= -1; //해당 아이디 없음
    }
```

FDaoManager.java 변경 후 ManagerDAO

```
public int managerCheck(String id, String passwd)
{
    String dbpasswd="";
    int x=-1;
    try {
        conn=ds.getConnection();
        pstmt = conn.prepareStatement(
            "select managerPasswd from manager where managerId = ? ");
        pstmt.setString(1, id);

        rs= pstmt.executeQuery();

        if(rs.next()){
            dbpasswd= rs.getString("managerPasswd");
            if(dbpasswd.equals(passwd))
                x= 1; //인증 성공
            else
                x= 0; //비밀번호 틀림
        }else
            x= -1; //해당 아이디 없음
    } catch (Exception ex) {
        ex.printStackTrace();
    } finally {
        if (rs != null)
            try { rs.close(); } catch (SQLException ex) {}
        if (pstmt != null)
            try { pstmt.close(); } catch (SQLException ex) {}
        if (conn != null)
            try { conn.close(); } catch (SQLException ex) {}
    }
    return x;
}
```

Manager Controller.j

ava 변경 전

```
if (com.contains("/manager/")) {
    if (com.equals("/foodshop/manager/managerLogin.do") == false
        && isLogin(request, response, session) == false) {

        response.sendRedirect(request.getContextPath() + "/foodshop/manager/managerLogin.do");
        return;
    }

    if (com.equals("/foodshop/manager/managerLogin.do")) {
        viewPage = "/foodshop/manager/logon/managerLogin.jsp"; // 로그인 페이지로 강제 이동
    } else if (com.equals("/foodshop/manager/managerMain.do")) { // 로그인이 된 상황
        viewPage = "/foodshop/manager/managerMain.jsp";
        service = new ManagerLoginService();
        service.execute(request, response); // 관리자로그인
    }
}
```

Manager Controller.j

```
ava
private boolean isLogin(HttpServletRequest request, HttpServletResponse response, HttpSession session) {
    try {
        if (request.getParameter("managerId") == null && session.getAttribute("managerId") == null) {
            return false;
        }
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }

    return true;
}
```



```

if (count == 0) {%>
<table>
<tr><td align="center">
    등록된 음식이 없습니다.</td>
</table>
<%> } else {%>
<table>
    <tr height="30" bgcolor="<%=value_c%>">
        <td align="center" width="30">번호</td>
        <td align="center" width="30">음식 종류</td>
        <td align="center" width="100">음식 이름</td>
        <td align="center" width="50">가격</td>
        <td align="center" width="50">수량</td>
        <td align="center" width="50">이미지</td>
        <td align="center" width="50">수정</td>
        <td align="center" width="50">삭제</td>
    </tr><%
    for (int i = 0 ; i < foodList.size() ; i++) {
        ShopFoodDataBean food =
            (ShopFoodDataBean) foodList.get(i);%>
    <tr height="30">
        <td width="30"><%=++number%></td>
        <td width="30"><%=food.getFood_kind()%></td>
        <td width="100" align="left">
            <%=food.getFood_name()%></td>
        <td width="50" align="right"><%=food.getFood_price()%></td>
        <td width="50" align="right">
            <% if(food.getFood_count()==0) {%>
                <font color="red"><%= "일시품절"%></font>
            <% }else{ %>
                <%=food.getFood_count()%>
            <% } %>
        </td>
        <td width="50"><%=food.getFood_image()%></td>
        <td width="50">
            <a href="foodUpdateForm.jsp?food_id=<%=food.getFood_id()%>">수정</a>
        </td>
        <td width="50">
            <a href="foodDeleteForm.jsp?food_id=<%=food.getFood_id()%>">삭제</a>
        </td>
    </tr>
<%}%>
</table>

```

foodList.jsp 변경 전

```

<c:if test="<${!empty foodList}>">
<table>
    <tr height="30" bgcolor="<%=value_c%>">
        <td align="center" width="30">번호</td>
        <td align="center" width="30">음식 종류</td>
        <td align="center" width="100">음식 이름</td>
        <td align="center" width="50">가격</td>
        <td align="center" width="50">수량</td>
        <td align="center" width="50">이미지</td>
        <td align="center" width="50">수정</td>
        <td align="center" width="50">삭제</td>
    </tr>

    <c:forEach items="<${foodList}>" var="foodDto">
        <c:set var="i" value="<${i+1}>" />
        <tr height="30">
            <td width="30"><${i}></td>
            <td width="30"><${foodDto.foodKind}></td>
            <td width="100" align="left"><${foodDto.foodName}></td>
            <td width="50" align="right"><${foodDto.foodPrice}></td>
            <td width="50" align="right">
                <c:choose>
                    <c:when test="<${foodDto.foodCount==0}>"><font color="red"><%= "일시품절"%></font></c:when>
                    <c:otherwise><${foodDto.foodCount}></c:otherwise>
                </c:choose>
            </td>
            <td width="50" align="right"><${foodDto.foodImage}></td>
            <td width="50"><a href="foodUpdateView.do?food_id=<${foodDto.foodId}>">수정</a></td>
            <td width="50"><a href="foodDelete.do?food_id=<${foodDto.foodId}>">삭제</a></td>
        </tr>
    </c:forEach>

    <c:if test="<${empty foodList}>">
        등록된 음식이 없습니다.
    </c:if>

```

foodList.jsp 변경 후

ShopFoodDBBean.jsp

변경 전

```
//음식 등록 메소드
public void insertFood(ShopFoodDataBean food)
throws Exception {
    Connection conn = null;
    PreparedStatement pstmt = null;

    try {
        conn = getConnection();

        pstmt = conn.prepareStatement(
            "insert into food values (?, ?, ?, ?, ?, ?, ?)");
        pstmt.setInt(1, food.getFood_id());
        pstmt.setString(2, food.getFood_kind());
        pstmt.setString(3, food.getFood_name());
        pstmt.setInt(4, food.getFood_price());
        pstmt.setShort(5, food.getFood_count());
        pstmt.setString(6, food.getFood_image());

        pstmt.executeUpdate();
    }
```

FDaoFood.java

변경 후 FoodDAO

```
// 음식 입력
public void insertfood(String foodkind, String foodname, int foodprice, int foodcount, String foodimage) {
    try {
        conn = ds.getConnection();
        String sql =
            "insert into food(food_kind, food_name, food_price, food_count, food_image) values (?, ?, ?, ?, ?)";
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, foodkind);
        pstmt.setString(2, foodname);
        pstmt.setInt(3, foodprice);
        pstmt.setInt(4, foodcount);
        pstmt.setString(5, foodimage);
        pstmt.executeUpdate();
    } catch (Exception e) {
    } finally {
        try {
            pstmt.close();
            conn.close();
        } catch (Exception e2) {
        }
    }
}
```

ShopFoodDBBean.jsp

변경 전

```
// 등록된 음식의 정보를 수정시 사용하는 메소드
public void updateFood(ShopFoodDataBean food, int foodId)
throws Exception {
    Connection conn = null;
    PreparedStatement pstmt = null;
    String sql;

    try {
        conn = getConnection();

        sql = "update food set food_kind=?,food_name=?,food_price=?";
        sql += ",food_count=?";
        sql += ",food_image=?";
        sql += " where food_id=?";

        pstmt = conn.prepareStatement(sql);

        pstmt.setString(1, food.getFood_kind());
        pstmt.setString(2, food.getFood_name());
        pstmt.setInt(3, food.getFood_price());
        pstmt.setShort(4, food.getFood_count());
        //pstmt.setString(5, food.getAuthor());
        //pstmt.setString(6, food.getPublishing_com());
        //pstmt.setString(7, food.getPublishing_date());
        pstmt.setString(5, food.getFood_image());
        //pstmt.setString(9, food.getFood_content());
        //pstmt.setByte(10, food.getDiscount_rate());
        pstmt.setInt(6, foodId);

        pstmt.executeUpdate();
    }
```

FDaoFood.java

변경 후 FoodDAO

```
// 음식 수정
public void updatefood(int foodid, String foodkind,
    String foodname, int foodprice, int foodcount, String foodimage)
try {
    conn = ds.getConnection();
    String sql =
        "update food set food_kind=?, food_name=?, food_price=?, "
        + "food_count=?, food_image=? where food_id=?";
    pstmt = conn.prepareStatement(sql);
    pstmt.setString(1, foodkind);
    pstmt.setString(2, foodname);
    pstmt.setInt(3, foodprice);
    pstmt.setInt(4, foodcount);
    pstmt.setString(5, foodimage);
    pstmt.setInt(6, foodid);
    pstmt.executeUpdate();
} catch (Exception e) {
    e.printStackTrace();
} finally {
    try {
        pstmt.close();
        conn.close();
    } catch (Exception e2) {
        e2.printStackTrace();
    }
}
```

foodUpdatePro.jsp 변경 전

```
<%  
int food_id= Integer.parseInt( imageUp.getParameter("food_id"));  
String food_kind = imageUp.getParameter("food_kind");  
String food_name = imageUp.getParameter("food_name");  
String food_price = imageUp.getParameter("food_price");  
String food_count = imageUp.getParameter("food_count");  
  
food.setFood_kind(food_kind);  
food.setFood_name(food_name);  
food.setFood_price(Integer.parseInt(food_price));  
food.setFood_count(Short.parseShort(food_count));  
food.setFood_image(filename);  
  
ShopFoodDBBean foodProcess = ShopFoodDBBean.getInstance();  
foodProcess.updateFood(food, food_id);  
  
response.sendRedirect("foodList.jsp?food_kind="+food_kind);  
%>
```

ManagerFoodUpdateService.java 변경 후 Service

```
String foodid = imageUp.getParameter("food_id");  
String food_kind = imageUp.getParameter("food_kind");  
String food_name = imageUp.getParameter("food_name");  
String food_price = imageUp.getParameter("food_price");  
String food_count = imageUp.getParameter("food_count");  
  
// FDtoFood fdto = dao.getFood(foodid);  
dao.updatefood(Integer.parseInt(foodid), food_kind, food_name, Integer.parseInt(food_price),  
Integer.parseInt(food_count), filename);
```


ShopFoodDBBean.jsp

변경 전

```
// foodId에 해당하는 음식의 정보를 삭제시 사용하는 메소드
public void deleteFood(int foodId)
throws Exception {
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs= null;

    try {
        conn = getConnection();

        pstmt = conn.prepareStatement(
            "delete from food where food_id=?");
        pstmt.setInt(1, foodId);

        pstmt.executeUpdate();
    }
```

FDaoFood.java

변경 후 FoodDAO

```
// 음식 삭제
public void deletefood(int foodid) {
    try {
        conn = ds.getConnection();
        String sql = "delete from food where food_id=?";
        pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, foodid);
        pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            pstmt.close();
            conn.close();
        } catch (Exception e2) {
            // TODO: handle exception
        }
    }
}
```

foodDeletePro.jsp 변경 전

```
<%  
    int food_id = Integer.parseInt(request.getParameter("food_id"));  
    String food_kind = request.getParameter("food_kind");  
  
    ShopFoodDBBean foodProcess = ShopFoodDBBean.getInstance();  
    foodProcess.deleteFood(food_id);  
  
%>
```

ManagerFoodDeleteService.java

변경 후 Service

```
public class ManagerFoodDeleteService implements FoodService {  
  
    @Override  
    public void execute(HttpServletRequest request, HttpServletResponse response) {  
        // TODO Auto-generated method stub  
        int foodid = Integer.parseInt(request.getParameter("food_id"));  
        FDaoFood dao = new FDaoFood();  
        dao.deletefood(foodid);  
    }  
}
```


#수행 소감

박지선

조민석

김동현

노용천



Q&A

• 감사합니다. •