[ 11 ] 효율적인 데이터베이스 관리 – 회원인증 프로그램 예제

DTO, DAO, 커넥션풀

1. DAO, DTO
   * DAO : Data Access Object
     + 데이터 베이스에 접속해서 데이터 추가, 삭제, 수정 등의 작업을 하는 클래스이다
     + 일반적인 JSP 혹은 Servlet 페이지 내에 위의 로직을 함께 기술할 수도 있지만, 유지보수 및 코드의 모듈화를 위해 별도의 DAO클래스를 만들어 사용한다
   * DTO : Data Transfer Object
     + DAO 클래스를 이용하여 데이터 베이스에서 데이터를 관리할 때 데이터를 일반적인 변수에 할당하여 작업 할 수도 있지만, 해당 데이터의 클래스 객체를 만들어 사용한다. 데이터만 하나로 모아 객체에 담는다.

WAS

웹브라우저

데이터베이스

Servlet/JSP

Servlet/JSP

WAS

DTO

웹브라우저

데이터베이스

Servlet/JSP

DAO

DTO

1. 회원인증 프로그램 전체 흐름도

로그인

login.jsp

회원정보 update

modifyOk.jsp

회원가입

join.jsp

로그아웃

logout.jsp

로그인

loginOk.jsp

회원가입인증

JoinOk.jsp

회원정보 수정

modify.jsp

메인

main.jsp

DAO

MemberDao



1. Members 테이블 생성

**DROP** **TABLE** MEMBER **CASCADE** CONSTRAINTS;

**CREATE** **TABLE** MEMBER(

ID VARCHAR2(30) **PRIMARY** **KEY**,

PW VARCHAR2(30) **NOT** **NULL**,

NAME VARCHAR2(30) **NOT** **NULL**,

phone1 varchar2(10),

phone2 varchar2(10),

phone3 varchar2(10),

gender varchar2(5),

EMAIL VARCHAR2(30),

BIRTH **DATE** **NOT** **NULL**,

RDATE **DATE** **NOT** **NULL**,

ADDRESS VARCHAR2(200));

-- 1. 회원가입시 id 중복체크를 위한 SQL

**SELECT** \* **FROM** MEMBER **WHERE** ID='aaa';

-- 2. 회원가입 SQL

**INSERT** **INTO** MEMBER (ID, PW, NAME, PHONE1, PHONE2, PHONE3, GENDER, EMAIL, BIRTH, RDATE, ADDRESS)

**VALUES** ('aaa','111','홍길동',**null**, **null**, **null**, 'm', 'hong@h.com', '1990-01-01',SYSDATE, '서울시 종로구');

-- 3. 로그인할 때 (ID/PW)

**SELECT** \* **FROM** MEMBER **WHERE** ID='aaa';

-- 4. ID로 DTO 가져오기

**SELECT** \* **FROM** MEMBER **WHERE** ID='aaa';

-- 5. 회원정보 수정

**UPDATE** MEMBER **SET** PW='111',

NAME='김길동',

PHONE1 = '02',

PHONE2='555',

PHONE3='6666',

GENDER='M',

EMAIL='a@a.com',

BIRTH='1990-01-01',

ADDRESS='서울시 용산구'

**WHERE** ID='aaa';

**commit**;

**SELECT** \* **FROM** member;

1. DTO

**package** com.lec.member;

**import** java.sql.Date;

**import** java.sql.Timestamp;

**public** **class** MemberDto {

**private** String id;

**private** String pw;

**private** String name;

**private** String phone1;

**private** String phone2;

**private** String phone3;

**private** String gender;

**private** String email;

**private** Timestamp birth;

**private** Date rdate;

**private** String address;

**public** MemberDto() { }

**public** MemberDto(String id, String pw, String name, String phone1, String phone2, String phone3, String gender,

String email, Timestamp birth, Date rdate, String address) {

**super**();

**this**.id = id;

**this**.pw = pw;

**this**.name = name;

**this**.phone1 = phone1;

**this**.phone2 = phone2;

**this**.phone3 = phone3;

**this**.gender = gender;

**this**.email = email;

**this**.birth = birth;

**this**.rdate = rdate;

**this**.address = address;

}

**public** String getName() {**return** name;}

**public** **void** setName(String name) {**this**.name = name;}

**public** String getId() {**return** id;}

**public** **void** setId(String id) {**this**.id = id;}

**public** String getPw() {**return** pw;}

**public** **void** setPw(String pw) {**this**.pw = pw;}

**public** String getPhone1() {**return** phone1;}

**public** **void** setPhone1(String phone1) {**this**.phone1 = phone1;}

**public** String getPhone2() {**return** phone2;}

**public** **void** setPhone2(String phone2) {**this**.phone2 = phone2;}

**public** String getPhone3() {**return** phone3;}

**public** **void** setPhone3(String phone3) {**this**.phone3 = phone3;}

**public** String getGender() {**return** gender;}

**public** **void** setGender(String gender) {**this**.gender = gender;}

**public** String getEmail() {**return** email;}

**public** **void** setEmail(String email) {**this**.email = email;}

**public** Timestamp getBirth() {**return** birth;}

**public** **void** setBirth(Timestamp birth) {**this**.birth = birth;}

**public** Date getRdate() { **return** rdate; }

**public** **void** setRdate(Date rdate) { **this**.rdate = rdate; }

**public** String getAddress() { **return** address; }

**public** **void** setAddress(String address) { **this**.address = address; }

@Override

**public** String toString() {

**return** "MemberDto [name=" + name + ", id=" + id + ", pw=" + pw + ", phone1=" + phone1 + ", phone2=" + phone2

+ ", phone3=" + phone3 + ", gender=" + gender + ", email=" + email + ", birth=" + birth + ", rdate="

+ rdate + ", address=" + address + "]";

}

}

1. DAO

**package** com.lec.member;

**import** java.sql.Connection;

**import** java.sql.Date;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.Timestamp;

**public** **class** MemberDao {

**public** **static** **final** **int** ***SUCCESS*** = 1; // 회원가입, 정보수정 할 때

**public** **static** **final** **int** ***FAIL*** = 0; // 회원가입, 정보수정 할 때

**public** **static** **final** **int** ***LOGIN\_SUCCESS*** = 1; // 로그인 성공

**public** **static** **final** **int** ***LOGIN\_FAIL\_ID*** = -1; // 로그인 오류(ID오류)

**public** **static** **final** **int** ***LOGIN\_FAIL\_PW*** = 0; // 로그인 오류(PW오류)

**public** **static** **final** **int** ***MEMBER\_EXISTENT*** = 0; // 있는(중복된) ID

**public** **static** **final** **int** ***MEMBER\_NONEXISTENT*** = 1; // 없는(사용가능한) ID

**public** **static** MemberDao *instance*; //자기가 자기 클래스를 참조하고 있는 변수

**public** **static** MemberDao getInstance() {

**if**(*instance*==**null**) {

*instance* = **new** MemberDao();

}

**return** *instance*;

}

**private** MemberDao() {}

// conn 객체 가져오는 함수

**public** **static** Connection getConnection() **throws** Exception{

Class.*forName*("oracle.jdbc.driver.OracleDriver");

Connection conn = DriverManager.*getConnection*("jdbc:oracle:thin:@localhost:1521:xe", "scott","tiger");

**return** conn;

}

// 1. 회원가입시 id 중복체크 (매개변수 :id)

**public** **int** confirmId(String id) {

**int** result = ***MEMBER\_EXISTENT***; // 초기화

Connection conn = **null**;

PreparedStatement pstmt = **null**;

ResultSet rs = **null**;

String sql = "SELECT \* FROM MEMBER WHERE ID=?";

**try** {

conn = *getConnection*();

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, id);

rs = pstmt.executeQuery();

**if**(rs.next()) { // 중복된 ID(있는 ID)

System.***out***.println("있다");

result = ***MEMBER\_EXISTENT***;

}**else** { // 사용가능한 ID(없는 ID)

System.***out***.println("없다");

result = ***MEMBER\_NONEXISTENT***;

}

} **catch** (Exception e) {

System.***out***.println(e.getMessage());

} **finally** {

**try** {

**if**(rs != **null**) rs.close();

**if**(pstmt != **null**) pstmt.close();

**if**(conn != **null**) conn.close();

} **catch** (Exception e) { }

}

**return** result;

}

// 2. 회원가입 (매개변수 :dto)

**public** **int** insertMember(MemberDto dto) {

**int** result = ***FAIL***;

Connection conn = **null**;

PreparedStatement pstmt = **null**;

String sql = "INSERT INTO MEMBER (ID, PW, NAME, PHONE1, PHONE2, PHONE3, GENDER, EMAIL, BIRTH, RDATE, ADDRESS) " +

" VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, SYSDATE, ?)";

**try** {

conn = *getConnection*();

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, dto.getId());

pstmt.setString(2, dto.getPw());

pstmt.setString(3, dto.getName());

pstmt.setString(4, dto.getPhone1());

pstmt.setString(5, dto.getPhone2());

pstmt.setString(6, dto.getPhone3());

pstmt.setString(7, dto.getGender());

pstmt.setString(8, dto.getEmail());

pstmt.setTimestamp(9, dto.getBirth());

pstmt.setString(10, dto.getAddress());

result = pstmt.executeUpdate();

System.***out***.println(result==***SUCCESS*** ? "회원가입 성공":"회원가입 실패"+dto.toString());

}**catch** (Exception e) {

System.***out***.println(e.getMessage());

System.***out***.println("회원가입 실패 "+dto.toString());

} **finally** {

**try** {

**if**(pstmt!=**null**) pstmt.close();

**if**(conn !=**null**) conn.close();

} **catch** (Exception e) { }

}

**return** result;

}

// 3. 로그인 처리 (매개변수 : id, pw)

**public** **int** loginCheck(String id, String pw) {

**int** result = ***LOGIN\_FAIL\_ID***;

Connection conn = **null**;

PreparedStatement pstmt = **null**;

ResultSet rs = **null**;

String sql = "SELECT \* FROM MEMBER WHERE ID=?";

**try** {

conn = *getConnection*();

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, id);

rs = pstmt.executeQuery();

**if**(rs.next()) {//유효한 id를 입력한 경우 - pw가 맞는지 체크

String dbPw = rs.getString("pw");

**if**(dbPw.equals(pw)) { // 로그인 성공

result = ***LOGIN\_SUCCESS***;

}**else** { // pw오류

result = ***LOGIN\_FAIL\_PW***;

}

}**else** { // 유효하지 않는 id를 입력한 경우

result = ***LOGIN\_FAIL\_ID***;

}

} **catch** (Exception e) {

System.***out***.println(e.getMessage());

}**finally** {

**try** {

**if**(rs !=**null**) rs.close();

**if**(pstmt!=**null**) pstmt.close();

**if**(conn !=**null**) conn.close();

} **catch** (Exception e) { }

}

**return** result;

}

// 4. id로 DTO가져오기 (매개변수 : id)

**public** MemberDto getMember(String id) {

MemberDto dto = **null**;

Connection conn = **null**;

PreparedStatement pstmt = **null**;

ResultSet rs = **null**;

String sql = "SELECT \* FROM MEMBER WHERE ID=?";

**try** {

conn = *getConnection*();

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, id);

rs = pstmt.executeQuery();

**if**(rs.next()) {

// dto = new MemberDto();

// dto.setId(id);

// dto.setPw(rs.getString("pw"));

// dto.setName(rs.getString("name"));

// dto.setEmail(rs.getString("email"));

// dto.setBirth(rs.getTimestamp("birth"));

// dto.setRdate(rs.getDate("rdate"));

// dto.setAddress(rs.getString("address"));

String pw = rs.getString("pw");

String name = rs.getString("name");

String phone1 = rs.getNString("phone1");

String phone2 = rs.getNString("phone2");

String phone3 = rs.getNString("phone3");

String gender = rs.getNString("gender");

String email = rs.getString("email");

Timestamp birth = rs.getTimestamp("birth");

Date rdate = rs.getDate("rdate");

String address = rs.getString("address");

dto = **new** MemberDto(id, pw, name, phone1, phone2, phone3, gender, email, birth, rdate, address);

}

} **catch** (Exception e) {

System.***out***.println(e.getMessage());

}**finally** {

**try** {

**if**(rs !=**null**) rs.close();

**if**(pstmt!=**null**) pstmt.close();

**if**(conn !=**null**) conn.close();

} **catch** (Exception e) { }

}

**return** dto;

}

// 5. 회원정보 수정 (매개변수 : dto)

**public** **int** updateMember(MemberDto dto) {

**int** result = ***FAIL***;

Connection conn = **null**;

PreparedStatement pstmt = **null**;

String sql = "UPDATE MEMBER SET PW=?, " +

" NAME=?, " +

" PHONE1 = ?, " +

" PHONE2 = ?, " +

" PHONE3 = ?, " +

" GENDER = ?, " +

" EMAIL=?, " +

" BIRTH=?, " +

" ADDRESS=? " +

" WHERE ID=?";

**try** {

conn = *getConnection*();

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, dto.getPw());

pstmt.setString(2, dto.getName());

pstmt.setString(3, dto.getPhone1());

pstmt.setString(4, dto.getPhone2());

pstmt.setString(5, dto.getPhone3());

pstmt.setString(6, dto.getGender());

pstmt.setString(7, dto.getEmail());

pstmt.setTimestamp(8, dto.getBirth());

pstmt.setString(9, dto.getAddress());

pstmt.setString(10, dto.getId());

result = pstmt.executeUpdate();

System.***out***.println(result==***SUCCESS*** ? "정보수정 성공":"정보수정 실패"+dto.toString());

}**catch** (Exception e) {

System.***out***.println(e.getMessage());

System.***out***.println("정보수정 실패 "+dto.toString());

} **finally** {

**try** {

**if**(pstmt!=**null**) pstmt.close();

**if**(conn !=**null**) conn.close();

} **catch** (Exception e) { }

}

**return** result;

}

}

1. test.jsp

<%@page import=*"com.lec.member.\*"*%>

<%@page import=*"java.sql.Date"*%>

<%@page import=*"java.sql.Timestamp"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<% String conPath = request.getContextPath(); %>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*<%=conPath %>*/css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<h1>테스트 페이지</h1>

<%

MemberDao mDao = MemberDao.getInstance();

**int** result = mDao.confirmId("bbb");

**if**(result==MemberDao.MEMBER\_EXISTENT){

out.println("<h3>1.confirmId테스트 : bbb는 있는(중복된) id</h3>");

}**else**{

out.println("<h3>1.confirmId테스트 :bbb는 없는(사용가능한) id</h3>");

}

out.println("<hr><h3>2. insertMember테스트</h3>");

// id, pw, name, email, Timestamp birth, Date rdate, address

MemberDto dto = **new** MemberDto("bbb","111","홍홍홍","02","555","5959","여","hong@g.com", Timestamp.valueOf("1990-02-13 12:21:00"), **null**, "서울시 강남구");

result = mDao.insertMember(dto);

**if**(result==MemberDao.SUCCESS){

out.println("가입 성공한 멤버 : "+dto.toString());

}**else**{

out.print("가입 실패");

}

out.println("<hr>3.로그인 loginCheck");

result = mDao.loginCheck("aaa", "111");

**if**(result == MemberDao.LOGIN\_SUCCESS){

out.println("aaa/111 로그인 성공");

}**else** **if**(result == MemberDao.LOGIN\_FAIL\_PW){

out.println("aaa/111 비밀번호 오류");

}**else** **if**(result == MemberDao.LOGIN\_FAIL\_ID){

out.println("aaa/111 아이디 오류");

}

out.println("<br>");

result = mDao.loginCheck("sdsd", "222");

**if**(result == MemberDao.LOGIN\_SUCCESS){

out.println("sdsd/222 로그인 성공");

}**else** **if**(result == MemberDao.LOGIN\_FAIL\_PW){

out.println("sdsd/222 비밀번호 오류");

}**else** **if**(result == MemberDao.LOGIN\_FAIL\_ID){

out.println("sdsd/222 아이디 오류");

}

out.println("<hr>4. id로 dto가져오기");

dto = mDao.getMember("aaa");

out.println("aaa아이디인 사람의 정보 : "+dto);

out.println("<hr>5. 회원정보 수정하기 aaa회원 정보 수정");

dto = **new** MemberDto("aaa","111","신길동","031","555","5959","남","sin@g.com", Timestamp.valueOf("1990-02-13 12:21:00"), **null**, "서울시 강남구");

result = mDao.updateMember(dto);

**if**(result==MemberDao.SUCCESS){

out.println("수정 후 데이터 : "+mDao.getMember("aaa"));

}**else**{

out.println("수정 실패"+dto);

}

%>

</body>

</html>

1. main.jsp

 

<%@page import=*"com.lec.member.MemberDto"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<% String conPath = request.getContextPath(); %>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*<%=conPath %>*/css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<%

MemberDto member = (MemberDto)session.getAttribute("member");

**if**(member!=**null**){

out.println("<h2>"+member.getName()+"님 어서오세요</h2>");

out.println("<button onclick=\"location.href='modify.jsp'\">정보수정</button>");

out.println("<button onclick=\"location.href='logout.jsp'\">로그아웃</button>");

}**else**{

out.println("<h2>손님 어서오세요</h2>");

out.println("<button onclick=\"location.href='join.jsp'\">회원가입</button>");

out.println("<button onclick=\"location.href='login.jsp'\">로그인</button>");

}

%>

</body>

</html>

1. join.jsp



<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<% String conPath = request.getContextPath(); %>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"../css/style.css"* rel=*"stylesheet"*>

<script src=*"../js/join.js"*></script>

</head>

<body>

<form action=*"joinOk.jsp"* method=*"post"* name=*"join\_frm"*>

<table>

<caption>회원가입</caption>

<tr><th>아이디</th><td><input type=*"text"* name=*"id"*></td></tr>

<tr><th>비밀번호</th><td><input type=*"password"* name=*"pw"*></td></tr>

<tr><th>비밀번호</th><td><input type=*"password"* name=*"pwChk"*></td></tr>

<tr><th>이름</th><td><input type=*"text"* name=*"name"*></td></tr>

<tr><th>전화번호</th>

<td><select name=*"phone1"*>

<option></option><option>02</option><option>010</option>

</select> -

<input type=*"text"* name=*"phone2"* size=*"2"*> -

<input type=*"text"* name=*"phone3"* size=*"2"*>

</td>

</tr>

<tr><th>성별</th>

<td><input type=*"radio"* name=*"gender"* value=*"m"*>남

<input type=*"radio"* name=*"gender"* value=*"f"*>여

</td>

</tr>

<tr><th>메일</th><td><input type=*"email"* name=*"email"*></td></tr>

<tr><th>생년월일</th><td><input type=*"date"* name=*"tempbirth"*></td></tr>

<tr><th>주소</th><th><input type=*"text"* name=*"addresss"*></th></tr>

<tr><td colspan=*"2"*>

<input type=*"button"* value=*"회원가입"* onclick="infoConfirm()">

<input type=*"reset"* value=*"취소"*>

<input type=*"button"* value=*"로그인"* onclick="location.href='login.jsp'">

</table>

</form>

</body>

</html>

1. joinOk.jsp

<%@page import=*"com.lec.member.MemberDao"*%>

<%@page import=*"java.sql.Timestamp"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%request.setCharacterEncoding("utf-8");

String conPath = request.getContextPath();%>

<jsp:useBean id=*"dto"* class=*"com.lec.member.MemberDto"*/>

<jsp:setProperty name=*"dto"* property=*"\*"* />

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*<%=conPath %>*/css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<%

String tempbirth = request.getParameter("tempbirth"); // 1991-12-12

dto.setBirth(Timestamp.valueOf(tempbirth+" 00:00:00"));

// dto.setRdate(new Date(System.currentTimeMillis()));

MemberDao mDao = MemberDao.getInstance();

**int** result = mDao.confirmId(dto.getId()); // ID 중복체크

**if**(result == MemberDao.MEMBER\_NONEXISTENT){ // 사용가능한 ID

result = mDao.insertMember(dto); // 회원가입

**if**(result == MemberDao.SUCCESS){ //회원가입 성공

session.setAttribute("id", dto.getId());

%> <script>

alert('회원가입 감사합니다. 로그인 이후에 서비스를 이용하세요');

location.href='login.jsp';

</script>

<% }**else**{ //회원가입 실패

%>

<script>

alert('회원가입이 실패되었습니다. 다시 가입해 주세요');

location.href='join.jsp';

</script>

<%

}

}**else**{ // 중복된 ID

%> <script>

alert('중복된 ID입니다. 다른 아이디를 사용하세요');

history.back();

</script>

<%}%>

</body>

</html>

1. login.jsp



<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<% String conPath = request.getContextPath(); %>

<%**if**(session.getAttribute("member")!=**null**){

response.sendRedirect("main.jsp");

}

%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*<%=conPath %>*/css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<form action=*"loginOk.jsp"* method=*"post"*>

<table>

<caption>로그인</caption>

<tr><th>아이디</th><td><input type=*"text"* name=*"id"* required=*"required"*

value=*"*<%

String id = (String)session.getAttribute("id");

out.print(id!=**null**? id:"");

%>*"*></td>

</tr>

<tr><th>비밀번호</th><td><input type=*"password"* name=*"pw"* required=*"required"*></td></tr>

<tr><td colspan=*"2"*>

<input type=*"submit"* value=*"로그인"*>

<input type=*"button"* value=*"회원가입"* onclick="location.href='join.jsp'">

</td>

</tr>

</table>

</form>

</body>

</html>

1. loginOk.jsp

<%@page import=*"com.lec.member.\*"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<% String conPath = request.getContextPath(); %>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*<%=conPath %>*/css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<%request.setCharacterEncoding("utf-8");

String id = request.getParameter("id");

String pw = request.getParameter("pw");

MemberDao mDao = MemberDao.getInstance();

**int** result = mDao.loginCheck(id, pw);

**if**(result==MemberDao.LOGIN\_SUCCESS){ // 로그인 성공

MemberDto member = mDao.getMember(id);

session.setAttribute("member", member);

response.sendRedirect("main.jsp");

}**else** **if**(result==MemberDao.LOGIN\_FAIL\_PW){//pw오류

%>

<script>

alert('비밀번호를 확인하세요');

history.go(-1);

</script>

<%}**else** **if**(result==MemberDao.LOGIN\_FAIL\_ID){%>

<script>

alert('존재하지 않는 아이디입니다');

history.go(-1);

</script>

<%}%>

</body>

</html>

1. modify.jsp

<%@page import=*"java.sql.Timestamp"*%>

<%@page import=*"com.lec.member.MemberDto"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<% String conPath = request.getContextPath(); %>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*<%=conPath %>*/css/style.css"* rel=*"stylesheet"*>

<script src=*"*<%=conPath %>*/js/modify.js"*></script>

</head>

<body>

<%!String id, name, phone1, phone2, phone3, gender, email, address; Timestamp birth;%>

<%MemberDto member = (MemberDto)session.getAttribute("member");

**if**(member==**null**){

response.sendRedirect("login.jsp");

}**else**{

id = member.getId();

name = member.getName();

phone1 = member.getPhone1();

phone2 = member.getPhone2();

phone3 = member.getPhone3();

gender = member.getGender();

email = member.getEmail();

birth = member.getBirth();

address = member.getAddress();

}

%>

<form action=*"modifyOk.jsp"* method=*"post"* name=*"modify\_frm"* onsubmit="return infoConfirm()">

<table>

<caption>정보수정</caption>

<tr><th>아이디</th><td><input type=*"text"* name=*"id"* value=*"*<%=id%>*"* readonly=*"readonly"*></td></tr>

<tr><th>현비밀번호</th><td><input type=*"password"* name=*"oldPw"* required=*"required"*></td></tr>

<tr><th>새비밀번호</th><td><input type=*"password"* name=*"pw"*></td></tr>

<tr><th>새비밀번호</th><td><input type=*"password"* name=*"pwChk"*></td></tr>

<tr><th>이름</th>

<td><input type=*"text"* name=*"name"* value=*"*<%=name%>*"* required=*"required"*></td>

</tr>

<tr><th>전화번호</th>

<td><select name=*"phone1"*>

<option></option>

<option <%**if**("02".equals(phone1)){out.print("selected='selected'");} %>>02</option>

<option <%**if**("010".equals(phone1)){out.print("selected='selected'");} %>>010</option>

</select> -

<input type=*"text"* name=*"phone2"* value=*"*<%=phone2!=**null**? phone2:"" %>*"* size=*"3"*> -

<input type=*"text"* name=*"phone3"* value=*"*<%=phone3==**null**? "":phone2 %>*"* size=*"3"*>

</td>

</tr>

<tr><th>성별</th>

<td>

<%**if**("m".equals(gender)){%>

<input type=*"radio"* name=*"gender"* value=*"m"* checked=*"checked"*>남

<input type=*"radio"* name=*"gender"* value=*"f"*>여

<%}**else** **if**("f".equals(gender)){%>

<input type=*"radio"* name=*"gender"* value=*"m"*>남

<input type=*"radio"* name=*"gender"* value=*"f"* checked=*"checked"*>여

<%}**else**{%>

<input type=*"radio"* name=*"gender"* value=*"m"*>남

<input type=*"radio"* name=*"gender"* value=*"f"*>여

<%}%>

</td>

</tr>

<tr><th>메일</th>

<td><input type=*"email"* name=*"email"* value=*"*<%=email==**null**? "":email%>*"*></td></tr>

<tr><th>생년월일</th>

<td><input type=*"date"* name=*"tempbirth"*

value=*"*<%

**if**(birth!=**null**) out.print(birth.toString().substring(0, 10));

%>*"*

required=*"required"*></td></tr>

<tr><th>주소</th>

<td><input type=*"text"* name=*"address"* value=*"*<%=address!=**null**? address:"" %>*"*></td></tr>

<tr><td colspan=*"2"*>

<input type=*"submit"* value=*"정보수정"*>

<input type=*"reset"* value=*"취소"*>

<input type=*"button"* value=*"로그아웃"* onclick="location.href='logout.jsp'">

</table>

</form>

</body>

</html>

1. modifyOk.jsp

<%@page import=*"java.sql.Timestamp"*%>

<%@page import=*"com.lec.member.\*"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<% String conPath = request.getContextPath(); %>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*<%=conPath %>*/css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<%request.setCharacterEncoding("utf-8");%>

<jsp:useBean id=*"dto"* class=*"com.lec.member.MemberDto"*/>

<jsp:setProperty name=*"dto"* property=*"\*"* />

<%

String tempbirth = request.getParameter("tempbirth");

dto.setBirth(Timestamp.valueOf(tempbirth+" 00:00:00"));

String oldPw = request.getParameter("oldPw");

MemberDto member = (MemberDto)session.getAttribute("member");

String sessionPw = **null**;

**if**(member!=**null**){

sessionPw = member.getPw();

}

// 새비밀번호를 수정하지 않을 때(pw가 null이 들어옴)

**if**(dto.getPw()==**null**){

dto.setPw(sessionPw);

}

**if**(oldPw.equals(sessionPw)){ // 현비밀번호를 맞게 입력한 경우 정보 수정 진행

MemberDao mDao = MemberDao.getInstance();

**int** result = mDao.updateMember(dto);

**if**(result == MemberDao.SUCCESS){

session.setAttribute("member", dto);

%> <script>

alert('회원정보 수정이 완료되었습니다');

location.href = 'main.jsp';

</script>

<% }**else**{ %>

<script>

alert('회원정보 수정이 실패되었습니다. 정보가 너무 깁니다');

location.href = 'modify.jsp';

</script>

<% }

}**else**{ // 현비밀번호를 틀리게 입력한 경우 뭐가 카고 돌려보냄

%>

<script>

alert('현비밀번호가 바르지 않습니다. 확인하세요');

history.back();

</script>

<% }%>

</body>

</html>

1. logout.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<% String conPath = request.getContextPath(); %>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*<%=conPath %>*/css/style.css"* rel=*"stylesheet"*>

<script>

setTimeout(**function**(){

location.href = 'main.jsp';

},2000);

</script>

</head>

<body>

<%session.invalidate(); %>

<h2>로그아웃 되었습니다. 잠시 후 메인 페이지로 이동합니다</h2>

</body>

</html>

/\*\* join.js

\* join\_frm.id는 3글자 이상, join\_frm.pw==join\_frm, 이름, 메일

\*/

**function** infoConfirm(){

**if**(join\_frm.id.value.length<3){

alert('아이디는 반드시 3글자 이상');

join\_frm.id.focus();

**return**;

}

**if**(!join\_frm.pw.value){

alert('비밀번호는 필수 입력 사항입니다');

join\_frm.pw.focus();

**return**;

}

**if**(join\_frm.pw.value!=join\_frm.pwChk.value){

alert('비밀번호를 확인하세요');

join\_frm.pw.value = '';

join\_frm.pwChk.value = '';

join\_frm.pw.focus();

**return**;

}

**if**(!join\_frm.name.value){

alert('이름은 필수 입력 사항입니다');

join\_frm.name.focus();

**return**;

}

**if**(join\_frm.tempbirth.value.length==0){

alert('생년월일은 필수 입력 사항입니다');

join\_frm.tempbirth.focus();

**return**;

}

join\_frm.submit(); // 강제로 submit 이벤트 발생

}

/\*\* modify.js

\* 정보수정 시 modify\_frm.pw == modify\_frm.pwChk

\*/

**function** infoConfirm(){

**if**(modify\_frm.pw.value != modify\_frm.pwChk.value){

alert('새비밀번호를 확인하세요');

modify\_frm.pw.value = '';

modify\_frm.pwChk.value = '';

modify\_frm.pw.focus();

**return** **false**;

}

}

1. 커넥션 풀
   * 클라이언트에서 다수의 요청이 발생할 경우 데이터베이스에 부하가 발생하게 된다

웹브라우저

WAS

DTO

데이터베이스

Servlet JSP

DAO

웹브라우저

DTO

웹브라우저

…

ⓒⓒ.. ⓒ

웹브라우저

* + 이러한 문제를 해결 하기 위해서 커넥션 풀(DataBase Connection Pool)기법을 이용

tomcat컨테이너가 데이터베이스 인증을 하도록 context.xml 파일에 아래의 코드를 추가

<Resource

auth=*"Container"*

driverClassName = *"oracle.jdbc.driver.OracleDriver"*

url = *"jdbc:oracle:thin:@localhost:1521:xe"*

username = *"scott"*

password = *"tiger"*

name = *"jdbc/Oracle11g"*

type = *"javax.sql.DataSource"*

maxActive = *"10"*

maxWait = *"1000"*

/>

</Context>

**커넥션 풀을 이용한 DAO**

**package** com.lec.member;

**import** java.sql.Connection;

**import** java.sql.Date;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.Timestamp;

**import** javax.naming.Context;

**import** javax.naming.InitialContext;

**import** javax.naming.NamingException;

**import** javax.sql.DataSource;

**public** **class** MemberDao2 {

**public** **static** **final** **int** ***SUCCESS*** = 1; // 회원가입, 정보수정 할 때

**public** **static** **final** **int** ***FAIL*** = 0; // 회원가입, 정보수정 할 때

**public** **static** **final** **int** ***LOGIN\_SUCCESS*** = 1; // 로그인 성공

**public** **static** **final** **int** ***LOGIN\_FAIL\_ID*** = -1; // 로그인 오류(ID오류)

**public** **static** **final** **int** ***LOGIN\_FAIL\_PW*** = 0; // 로그인 오류(PW오류)

**public** **static** **final** **int** ***MEMBER\_EXISTENT*** = 0; // 있는(중복된) ID

**public** **static** **final** **int** ***MEMBER\_NONEXISTENT*** = 1; // 없는(사용가능한) ID

**public** **static** MemberDao2 *instance*; //자기가 자기 클래스를 참조하고 있는 변수

**public** **static** MemberDao2 getInstance() {

**if**(*instance*==**null**) {

*instance* = **new** MemberDao2();

}

**return** *instance*;

}

// conn 객체 가져오는 함수

**public** **static** Connection getConnection() **throws** Exception{

Connection conn = **null**;

**try** {

Context ctx = **new** InitialContext();

DataSource ds = (DataSource) ctx.lookup("java:comp/env/jdbc/Oracle11g");

conn = ds.getConnection();

} **catch** (NamingException e) {

System.***out***.println(e.getMessage());

}

**return** conn;

}

// 1. 회원가입시 id 중복체크 (매개변수 :id)

**public** **int** confirmId(String id) {

**int** result = ***MEMBER\_EXISTENT***; // 초기화

Connection conn = **null**;

PreparedStatement pstmt = **null**;

ResultSet rs = **null**;

String sql = "SELECT \* FROM MEMBER WHERE ID=?";

**try** {

conn = *getConnection*();

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, id);

rs = pstmt.executeQuery();

**if**(rs.next()) { // 중복된 ID(있는 ID)

System.***out***.println("있다");

result = ***MEMBER\_EXISTENT***;

}**else** { // 사용가능한 ID(없는 ID)

System.***out***.println("없다");

result = ***MEMBER\_NONEXISTENT***;

}

} **catch** (Exception e) {

System.***out***.println(e.getMessage());

} **finally** {

**try** {

**if**(rs != **null**) rs.close();

**if**(pstmt != **null**) pstmt.close();

**if**(conn != **null**) conn.close();

} **catch** (Exception e) { }

}

**return** result;

}

// 2. 회원가입 (매개변수 :dto)

**public** **int** insertMember(MemberDto dto) {

**int** result = ***FAIL***;

Connection conn = **null**;

PreparedStatement pstmt = **null**;

String sql = "INSERT INTO MEMBER (ID, PW, NAME, PHONE1, PHONE2, PHONE3, GENDER, EMAIL, BIRTH, RDATE, ADDRESS) " +

" VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, SYSDATE, ?)";

**try** {

conn = *getConnection*();

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, dto.getId());

pstmt.setString(2, dto.getPw());

pstmt.setString(3, dto.getName());

pstmt.setString(4, dto.getPhone1());

pstmt.setString(5, dto.getPhone2());

pstmt.setString(6, dto.getPhone3());

pstmt.setString(7, dto.getGender());

pstmt.setString(8, dto.getEmail());

pstmt.setTimestamp(9, dto.getBirth());

pstmt.setString(10, dto.getAddress());

result = pstmt.executeUpdate();

System.***out***.println(result==***SUCCESS*** ? "회원가입 성공":"회원가입 실패"+dto.toString());

}**catch** (Exception e) {

System.***out***.println(e.getMessage());

System.***out***.println("회원가입 실패 "+dto.toString());

} **finally** {

**try** {

**if**(pstmt!=**null**) pstmt.close();

**if**(conn !=**null**) conn.close();

} **catch** (Exception e) { }

}

**return** result;

}

// 3. 로그인 처리 (매개변수 : id, pw)

**public** **int** loginCheck(String id, String pw) {

**int** result = ***LOGIN\_FAIL\_ID***;

Connection conn = **null**;

PreparedStatement pstmt = **null**;

ResultSet rs = **null**;

String sql = "SELECT \* FROM MEMBER WHERE ID=?";

**try** {

conn = *getConnection*();

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, id);

rs = pstmt.executeQuery();

**if**(rs.next()) {//유효한 id를 입력한 경우 - pw가 맞는지 체크

String dbPw = rs.getString("pw");

**if**(dbPw.equals(pw)) { // 로그인 성공

result = ***LOGIN\_SUCCESS***;

}**else** { // pw오류

result = ***LOGIN\_FAIL\_PW***;

}

}**else** { // 유효하지 않는 id를 입력한 경우

result = ***LOGIN\_FAIL\_ID***;

}

} **catch** (Exception e) {

System.***out***.println(e.getMessage());

}**finally** {

**try** {

**if**(rs !=**null**) rs.close();

**if**(pstmt!=**null**) pstmt.close();

**if**(conn !=**null**) conn.close();

} **catch** (Exception e) { }

}

**return** result;

}

// 4. id로 DTO가져오기 (매개변수 : id)

**public** MemberDto getMember(String id) {

MemberDto dto = **null**;

Connection conn = **null**;

PreparedStatement pstmt = **null**;

ResultSet rs = **null**;

String sql = "SELECT \* FROM MEMBER WHERE ID=?";

**try** {

conn = *getConnection*();

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, id);

rs = pstmt.executeQuery();

**if**(rs.next()) {

// dto = new MemberDto();

// dto.setId(id);

// dto.setPw(rs.getString("pw"));

// dto.setName(rs.getString("name"));

// dto.setEmail(rs.getString("email"));

// dto.setBirth(rs.getTimestamp("birth"));

// dto.setRdate(rs.getDate("rdate"));

// dto.setAddress(rs.getString("address"));

String pw = rs.getString("pw");

String name = rs.getString("name");

String phone1 = rs.getNString("phone1");

String phone2 = rs.getNString("phone2");

String phone3 = rs.getNString("phone3");

String gender = rs.getNString("gender");

String email = rs.getString("email");

Timestamp birth = rs.getTimestamp("birth");

Date rdate = rs.getDate("rdate");

String address = rs.getString("address");

dto = **new** MemberDto(id, pw, name, phone1, phone2, phone3, gender, email, birth, rdate, address);

}

} **catch** (Exception e) {

System.***out***.println(e.getMessage());

}**finally** {

**try** {

**if**(rs !=**null**) rs.close();

**if**(pstmt!=**null**) pstmt.close();

**if**(conn !=**null**) conn.close();

} **catch** (Exception e) { }

}

**return** dto;

}

// 5. 회원정보 수정 (매개변수 : dto)

**public** **int** updateMember(MemberDto dto) {

**int** result = ***FAIL***;

Connection conn = **null**;

PreparedStatement pstmt = **null**;

String sql = "UPDATE MEMBER SET PW=?, " +

" NAME=?, " +

" PHONE1 = ?, " +

" PHONE2 = ?, " +

" PHONE3 = ?, " +

" GENDER = ?, " +

" EMAIL=?, " +

" BIRTH=?, " +

" ADDRESS=? " +

" WHERE ID=?";

**try** {

conn = *getConnection*();

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, dto.getPw());

pstmt.setString(2, dto.getName());

pstmt.setString(3, dto.getPhone1());

pstmt.setString(4, dto.getPhone2());

pstmt.setString(5, dto.getPhone3());

pstmt.setString(6, dto.getGender());

pstmt.setString(7, dto.getEmail());

pstmt.setTimestamp(8, dto.getBirth());

pstmt.setString(9, dto.getAddress());

pstmt.setString(10, dto.getId());

result = pstmt.executeUpdate();

System.***out***.println(result==***SUCCESS*** ? "정보수정 성공":"정보수정 실패"+dto.toString());

}**catch** (Exception e) {

System.***out***.println(e.getMessage());

System.***out***.println("정보수정 실패 "+dto.toString());

} **finally** {

**try** {

**if**(pstmt!=**null**) pstmt.close();

**if**(conn !=**null**) conn.close();

} **catch** (Exception e) { }

}

**return** result;

}

}

응용예제. 회원인증(customer).pdf 참조

통합구현 평가 : 친구 추가가 가능한 아래의 화면을 구현하시오(예시화면 참조) 120분

**<세부 요구사항>**

▪ 테이블 FRIEND(NO, NAME, TEL) 생성

▪ 시퀀스 FRIEND\_NO\_SEQ(1부터 하나씩 증가하는 시퀀스) 생성

▪ 친구 추가 및 친구 전체 목록 출력하는 QUERY 작성 => friend.sql

▪ DTO 및 DAO(커넥션 풀 이용) 구현

▪ 뷰단 작성(friendInputList.jsp) : 아래 예시 화면과 같이 FRIEND 테이블에 있는 전체 친구 출력

▪ 프로세스단 작성(friendInputListPro.jsp) : 파라미터로 넘어온 친구이름과 전화를 FRIEND테이블에 INSERT한 후 친구 추가 성공 여부를 경고창으로 뿌려주고 뷰단으로 이동한다.





**<파일명>**

|  |
| --- |
| friend.sql  FriendDto.java  FriendDato.java  friendInputList.jsp(뷰단)  friendInputListPro.jsp(프로세스단)  첫실행화면.png  추가실행화면.png |

**<결과물 제출>**

위의 파일이 압축된 파일 : 훈련생성명\_통합구현평가.zip

<참고사항>

- 테이블 FRIEND : no, name, tel

- 시퀀스 FRIEND\_NO\_SEQ 1부터 하나씩 증가하는 시퀀스

1. 친구 추가 QUERY

2. 친구 목록 출력 QUERY

- DTO, DAO(컨넥션 풀 이용)

- friendInputList.jsp (첫화면 = 뷰단)

- friendInputListPro.jsp(친구 추가 프로세스단)

**<평가항목>**

▪ 기능에 필요한 테이블 및 시퀀스를 올바로 생성

▪ 기능에 맞는 쿼리의 사용

▪ 효율적인 DB 접근에 필요한 DTO 구현

▪ 효율적인 DB 연결을 위한 커넥션 풀을 이용한 구현

▪ DAO 클래스의 싱글톤 구현

▪ DAO에서 쿼리에 맞는 값 반환 함수 사용

▪ 뷰단에서의 form구현과 FRIEND 테이블 데이터 출력

▪ 프로세스단에서 받은 파라미터를 DAO를 이용하여 insert할 함수를 이용한 구현

▪ 프로세스단에서 상황에 따른 경고창을 출력하고 뷰단으로 location 변환 구현

위의 구현을 제출후 다음의 추가 요구사항을 추가 할 것을 사색해 봅시다

