[ 12 ] 효율적인 데이터베이스 관리

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

**package** com.ch.daotoex;

**public** **class** MemberDTO {

**private** String name;

**private** String id;

**private** String pw;

**private** String phone1;

**private** String phone2;

**private** String phone3;

**private** String gender;

**public** MemberDTO(){}

**public** MemberDTO(String name, String id, String pw, String phone1, String phone2, String phone3, String gender) {

**this**.name = name;

**this**.id = id;

**this**.pw = pw;

**this**.phone1 = phone1;

**this**.phone2 = phone2;

**this**.phone3 = phone3;

**this**.gender = gender;

}

**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; }

@Override

**public** String toString() {

**return** name+"("+id+")"+pw+" : "+phone1+phone2+phone3+gender;

}

}

**public** **class** MemberDAO {

**private** String driver = "oracle.jdbc.driver.OracleDriver";

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

**private** String uid = "scott";

**private** String upw = "tiger";

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

**public** **static** MemberDAO getInstance() {

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

*instance* = **new** MemberDAO();

}

**return** *instance*;

}

**private** MemberDAO() {

**try** {

Class.*forName*(driver);

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

}

}

**public** ArrayList<MemberDTO> memberSelectAll(){

ArrayList<MemberDTO> dtos = **new** ArrayList<MemberDTO>();

Connection conn = **null**;

Statement stmt = **null**;

ResultSet rs = **null**;

String sql = "SELECT \* FROM MEMBER";

**try** {

conn = DriverManager.*getConnection*(url, uid, upw);

stmt = conn.createStatement();

rs = stmt.executeQuery(sql);

**while**(rs.next()) {

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

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

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

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

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

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

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

dtos.add(**new** MemberDTO(name, id, pw, phone1, phone2, phone3, gender));

}

} **catch** (SQLException e) {

e.printStackTrace();

} **finally** {

**try** {

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

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

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

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

}

**return** dtos;

}

}

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

</head>

<body>

<table>

<caption>가입회원 리스트</caption>

<tr><th>이름</th><th>ID</th><th>비번</th><th>전화</th><th>성별</th></tr>

<%

MemberDAO memberDAO = MemberDAO.getInstance();

ArrayList<MemberDTO> dtos = memberDAO.memberSelectAll();

**for**(MemberDTO d:dtos){%>

<tr><td><%=d.getName() %></td><td><%=d.getId() %></td><td><%=d.getPw() %></td>

<td><%=d.getPhone1()==**null**? "": d.getPhone1()%>-

<%=d.getPhone2()==**null**? "": d.getPhone2() %>-

<%=d.getPhone3()==**null**? "": d.getPhone3() %></td>

<td><%=d.getGender() %></td>

</tr>

<% }%>

</table>

</body>

1. PreparedStatement
   * SQL문 실행을 위해 Statement객체를 이용할 단점을 보완한 PreparedStatement객체를 이용하는 경우가 많다

join.jsp (전에 만든 것 그대로 copy)

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

</head>

<body>

<!-- <form action="joinOk.jsp" method="post"> -->

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

<table>

<caption>회원가입</caption>

<tr><th>이름</th><td><input type=*"text"* name=*"name"* size=*"20"* required=*"required"*></td></tr>

<tr><th>아이디</th><td><input type=*"text"* name=*"id"* size=*"20"* required=*"required"*></td></tr>

<tr><th>패스워드</th><td><input type=*"password"* name=*"pw"* size=*"20"* required=*"required"*></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=*"5"*> -

<input type=*"text"* name=*"phone3"* size=*"5"*>

</td>

</tr>

<tr><th>성별</th>

<td><input type=*"radio"* name=*"gender"* value=*"남"* checked=*"checked"*>남 &nbsp;

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

</td>

</tr>

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

<input type=*"submit"* value=*"가입"*>

<input type=*"reset"* value=*"취소"*>

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

</td>

</tr>

</table>

</form>

</body>

</html>

**DAO추가 사항**

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

**int** result = 0;

Connection conn = **null**;

PreparedStatement pstmt = **null**;

String sql = "INSERT INTO MEMBER VALUES (?,?,?,?,?,?,?)";

//('홍','aaa','111','02','9999','9999','남')

**try** {

conn = DriverManager.*getConnection*(url, uid, upw);

pstmt = conn.prepareStatement(sql);

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

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

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

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

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

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

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

result = pstmt.executeUpdate();

System.***out***.println(result>0? "가입성공":"가입실패");

} **catch** (SQLException e) {

e.printStackTrace();

} **finally** {

**try** {

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

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

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

}

**return** result;

}

joinOk.jsp

<body>

<%

MemberDao memberDao = MemberDao.getInstance();

**int** result = memberDao.memberInsert(dto);

**if**(result>0){

out.println("<h2>회원가입이 완료 되었습니다.</h2>");

}**else**{

out.println("<h2>회원가입이 거부되었습니다.</h2>");

}

%> <hr>

<jsp:include page=*"memberView.jsp"*/>

</body>

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 = *"20"*

maxWait = *"1000"*

/>

</Context>

public class MemberConnDao {

// private String driver = "oracle.jdbc.driver.OracleDriver";

// private String url = "jdbc:oracle:thin:@localhost:1521:xe";

// private String uid = "scott";

// private String upw = "tiger";

private DataSource dataSource;

public static MemberConnDao *instance*;

public static MemberConnDao getInstance() {

if(*instance*==null) {

*instance* = new MemberConnDao();

}

return *instance*;

}

private MemberConnDao() {

/\*try {

Class.forName(driver);

} catch (ClassNotFoundException e) {

e.printStackTrace();

}\*/

try {

Context context = new InitialContext();

dataSource = (DataSource) context.lookup("java:comp/env/jdbc/Oracle11g");

} catch (NamingException e) {

e.printStackTrace();

}

}

public ArrayList<MemberDto> memberSelectAll(){

ArrayList<MemberDto> dtos = new ArrayList<MemberDto>();

Connection conn = null;

Statement stmt = null;

ResultSet rs = null;

String sql = "SELECT \* FROM MEMBER";

try {

//conn = DriverManager.getConnection(url, uid, upw);

conn = dataSource.getConnection();

stmt = conn.createStatement();

rs = stmt.executeQuery(sql);

while(rs.next()) {

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

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

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

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

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

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

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

dtos.add(new MemberDto(name, id, pw, phone1, phone2, phone3, gender));

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

try {

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

if(stmt!=null) stmt.close();

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

}catch (Exception e) { }

}

return dtos;

}

public int memberInsert(MemberDto dto) {

int result = 0;

Connection conn = null;

PreparedStatement pstmt = null;

String sql = "INSERT INTO MEMBER VALUES (?,?,?,?,?,?,?)";

//('홍','aaa','111','02','9999','9999','남')

try {

//conn = DriverManager.getConnection(url, uid, upw);

conn = dataSource.getConnection();

pstmt = conn.prepareStatement(sql);

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

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

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

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

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

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

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

result = pstmt.executeUpdate();

System.*out*.println(result>0? "가입성공":"가입실패");

} catch (SQLException e) {

e.printStackTrace();

} finally {

try {

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

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

}catch (Exception e) { }

}

return result;

}

}

과제

**DROP** SEQUENCE FRIEND\_SEQ;

**CREATE** SEQUENCE FRIEND\_SEQ;

**DROP** **TABLE** FRIEND;

**CREATE** **TABLE** FRIEND(

**NO** NUMBER(4,0) **PRIMARY** **KEY**,

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

TEL VARCHAR2(30));

**SELECT** \* **FROM** FRIEND;

**INSERT** **INTO** FRIEND **VALUES** (FRIEND\_SEQ.NEXTVAL, '홍길동', '010-9999-9999');

**SELECT** \* **FROM** FRIEND **WHERE** NAME **LIKE** '%길%' **AND** TEL **LIKE** '%9%';

**COMMIT**;

**delete** **from** friend;

**package** com.tj.dto;

**public** **class** FriendDto {

**private** **int** no;

**private** String name;

**private** String tel;

**public** FriendDto() { }

**public** FriendDto(**int** no, String name, String tel) {

**this**.no = no;

**this**.name = name;

**this**.tel = tel;

}

**public** **int** getNo() {**return** no;}

**public** **void** setNo(**int** no) {**this**.no = no;}

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

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

**public** String getTel() {**return** tel;}

**public** **void** setTel(String tel) {**this**.tel = tel;}

@Override

**public** String toString() {

**return** "FriendDto [no=" + no + ", name=" + name + ", tel=" + tel + "]";

}

}

**package** com.tj.dao;

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.util.ArrayList;

**import** javax.naming.Context;

**import** javax.naming.InitialContext;

**import** javax.naming.NamingException;

**import** javax.sql.DataSource;

**import** com.tj.dto.FriendDto;

**public** **class** FriendDao {

**private** DataSource dataSource;

**private** **static** FriendDao *instance* = **new** FriendDao();

**public** **static** FriendDao getInstance() {

**return** *instance*;

}

**private** FriendDao() {

**try** {

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

dataSource = (DataSource) context.lookup("java:comp/env/jdbc/Oracle11g");

} **catch** (NamingException e) {

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

}

}

**public** ArrayList<FriendDto> friendsSelect(){

ArrayList<FriendDto> dtos = **new** ArrayList<FriendDto>();

Connection conn = **null**;

PreparedStatement pstmt = **null**;

ResultSet rs = **null**;

String sql = "SELECT \* FROM FRIEND";

**try** {

conn = dataSource.getConnection();

pstmt = conn.prepareStatement(sql);

rs = pstmt.executeQuery();

**while**(rs.next()) {

FriendDto dto = **new** FriendDto();

dto.setNo(rs.getInt("no"));

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

dto.setTel(rs.getString("tel"));

dtos.add(dto);

}

} **catch** (SQLException e) {

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

}**finally**{

**try** {

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

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

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

} **catch** (SQLException e) { }

}

**return** dtos;

}

**public** ArrayList<FriendDto> friendsSelect(String name, String tel) {

ArrayList<FriendDto> dtos = **new** ArrayList<>();

Connection conn = **null**;

PreparedStatement pstmt = **null**;

ResultSet rs = **null**;

String sql = "select \* from friend where name like '%'||?||'%' and tel like ?";

**try** {

conn = dataSource.getConnection();

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, name);

pstmt.setString(2, "%"+tel+"%");

rs = pstmt.executeQuery();

**while** (rs.next()) {

FriendDto dto = **new** FriendDto();

dto.setNo(rs.getInt("no"));

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

dto.setTel(rs.getString("tel"));

dtos.add(dto);

}

} **catch** (SQLException e) {

e.printStackTrace();

} **finally** {

**try** {

**if** (rs != **null**)

rs.close();

**if** (pstmt != **null**)

pstmt.close();

**if** (conn != **null**)

conn.close();

} **catch** (SQLException e) {

}

}

**return** dtos;

}

**public** **int** friendInsert(FriendDto dto) {

**int** result = 0;

Connection conn = **null**;

PreparedStatement pstmt = **null**;

String sql = "INSERT INTO FRIEND VALUES (FRIEND\_SEQ.NEXTVAL, ?, ?)";

**try** {

conn = dataSource.getConnection();

pstmt = conn.prepareStatement(sql);

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

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

result = pstmt.executeUpdate();

System.***out***.println(result > 0 ? "친구추가성공" : "친구추가실패");

} **catch** (SQLException e) {

e.printStackTrace();

} **finally** {

**try** {

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

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

} **catch** (SQLException e) {

}

}

**return** result;

}

}

<%@page import=*"com.tj.dto.FriendDto"*%>

<%@page import=*"java.util.ArrayList"*%>

<%@page import=*"com.tj.dao.FriendDao"*%>

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

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<title>Insert title here</title>

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

<style>

**input** {border: *1px solid darkblue*;}

**p** {text-align: *center*;}

</style>

<%String result=request.getParameter("result");

**if**("1".equals(result)){%>

<script>alert('친구추가 성공');</script>

<%}**else** **if**("0".equals(result)){%>

<script>alert('친구추가 실패. 이름,전화는 30자 이내로');</script>

<%}%>

<script>

**function** searchFriend(){

**var** name = document.getElementsByName("name")[0].value;

**var** tel = document.getElementsByName("tel")[0].value;

location.href='friendView2.jsp?name=' + name + '&tel=' + tel;

}

</script>

</head>

<body>

<form action=*"friendPro.jsp"*>

<p>

친구이름 <input type=*"text"* name=*"name"* size=*"10"*

required=*"required"*>

전화 <input type=*"text"* name=*"tel"* size=*"10"*

required=*"required"*>

<input type=*"submit"* value=*"추가"*>

<input type=*"button"* value=*"검색"*

onclick="searchFriend()">

</p>

</form>

<br><br>

<%

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

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

**if** (name == **null**)

name = "";

**if** (tel == **null**)

tel = "";

System.out.println("name="+name+"\ttel="+tel);

FriendDao dao = FriendDao.getInstance();

ArrayList<FriendDto> dtos = dao.friendsSelect(name, tel);

%>

<table>

<tr><th>번호</th><th>이름</th><th>전화</th></tr>

<%**for** (FriendDto d : dtos) {%>

<tr>

<td><%=d.getNo()%></td>

<td><%=d.getName()%></td>

<td><%=d.getTel()%></td>

</tr>

<%}%>

</table>

</body>

</html>

<%@page import=*"com.tj.dao.FriendDao"*%>

<%@page import=*"com.tj.dto.FriendDto"*%>

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

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<title>Insert title here</title>

</head>

<body>

<jsp:useBean id=*"dto"* class=*"com.tj.dto.FriendDto"*/>

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

<%

FriendDao dao = FriendDao.getInstance();

**int** result = dao.friendInsert(dto);

System.out.println(result>0? "입력성공":"입력실패");

response.sendRedirect("friendView.jsp?result="+result);

%>

</body>

</html>