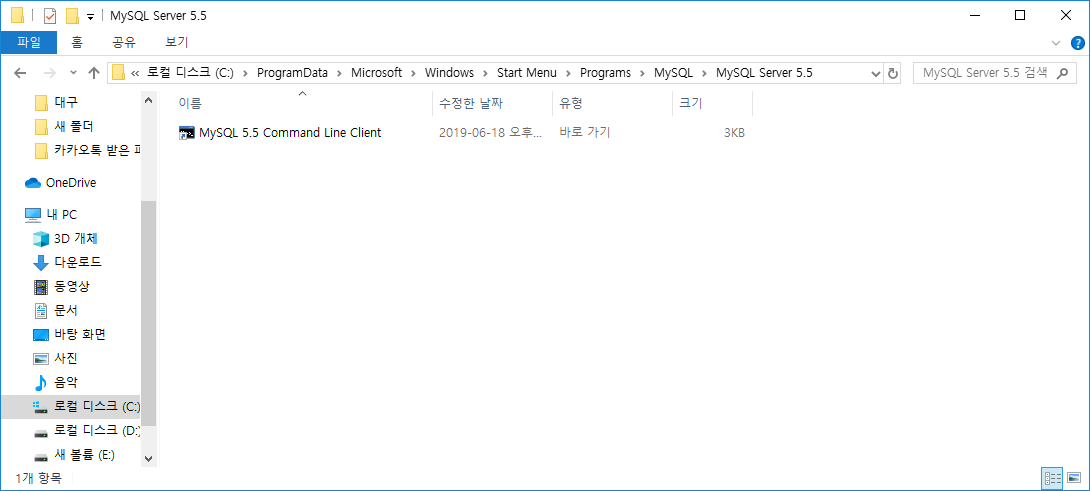
**4차시험 - 장채연**

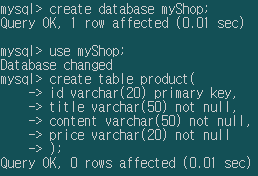
**깃허브 주소 :** [**https://github.com/oechomuchim**](https://github.com/oechomuchim)

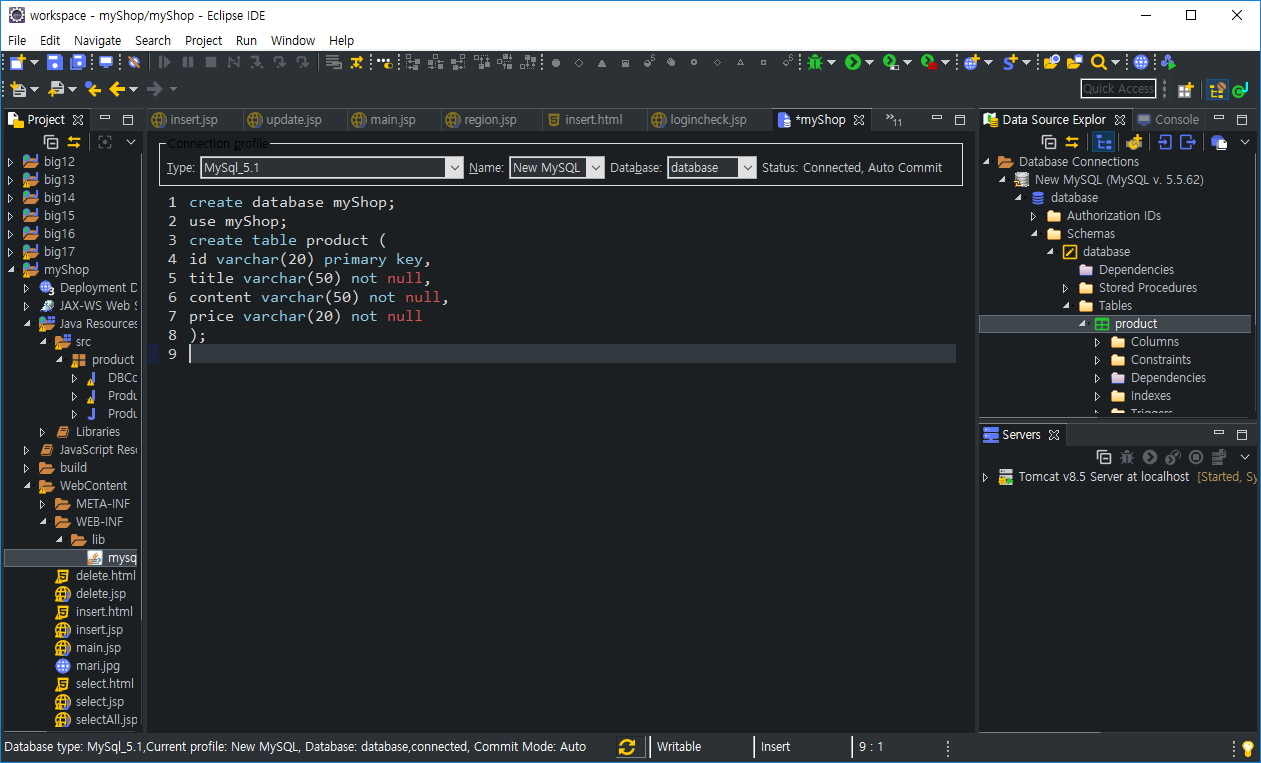
1. MySQL서버 설치 위치 화면 캡쳐



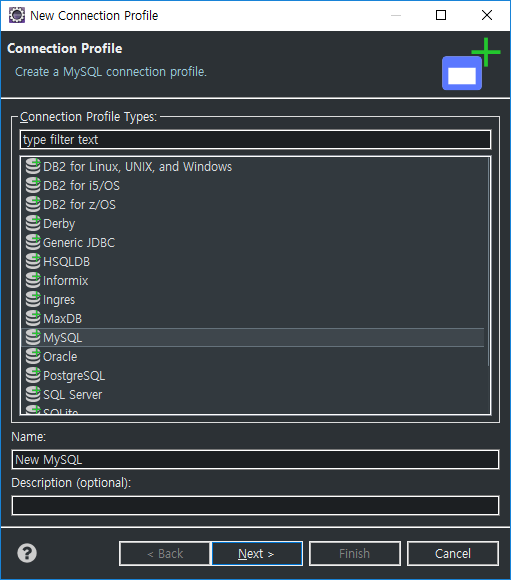
2) DB만드는 과정 화면 캡쳐

3) 테이블 만드는 과정 화면 캡쳐

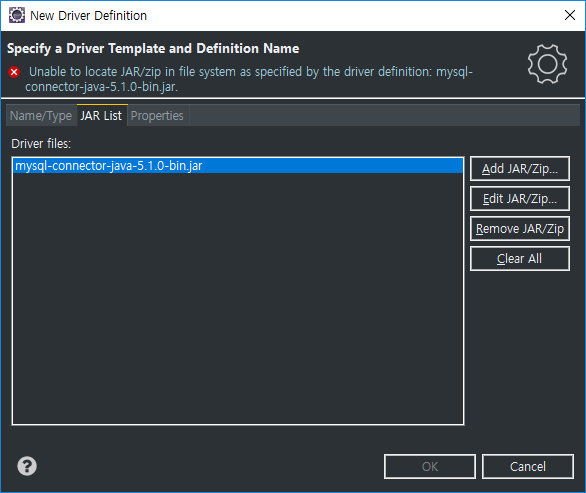




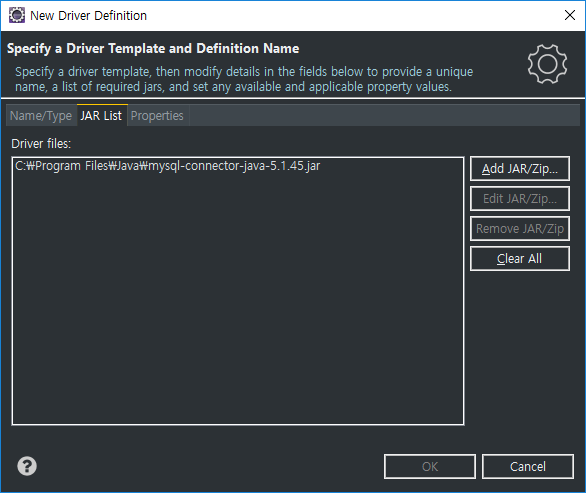
4) 이클립스에서 datasource Explorer설정 과정 화면 캡쳐

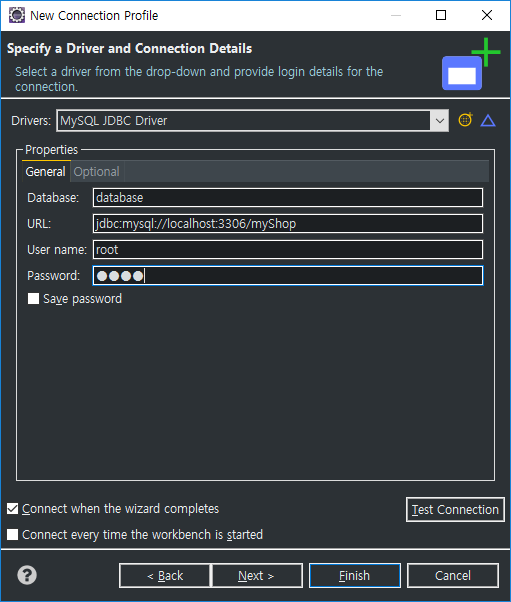


database connection 에서 우클릭 후, New 클릭



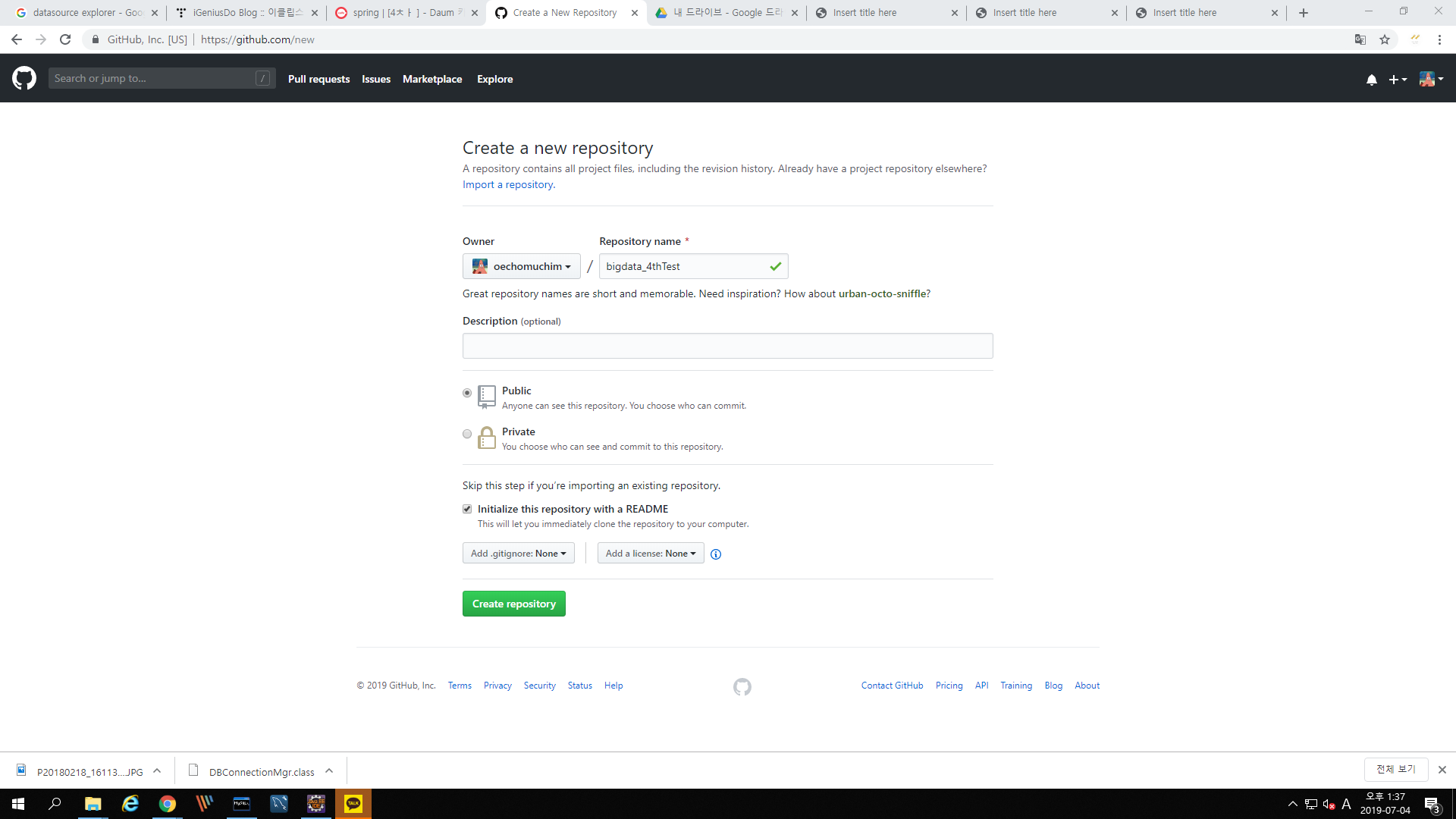
원래 있던 jar를 삭제하고, add jar클릭해서 다운받아 놓은 jar파일 추가

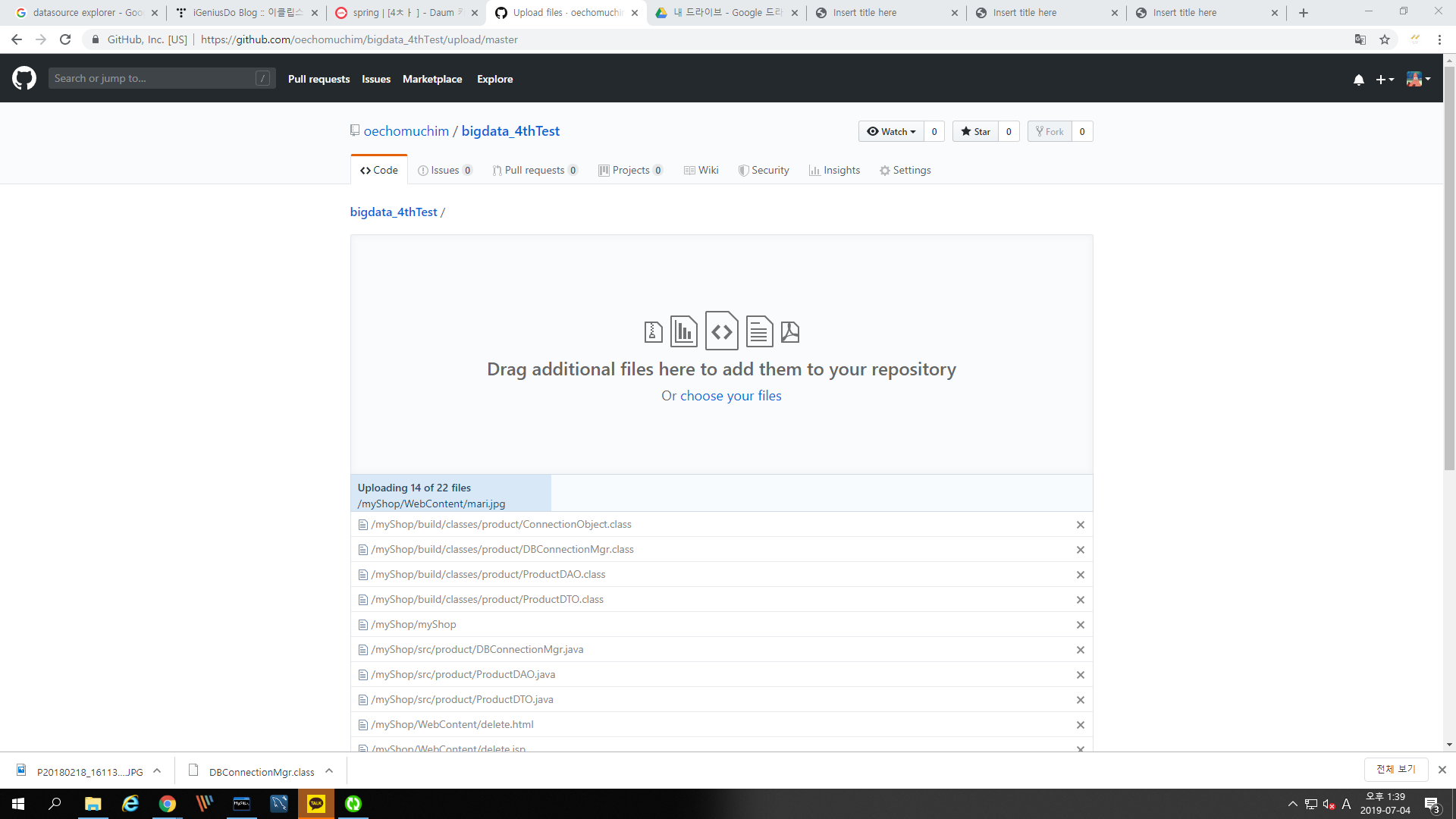




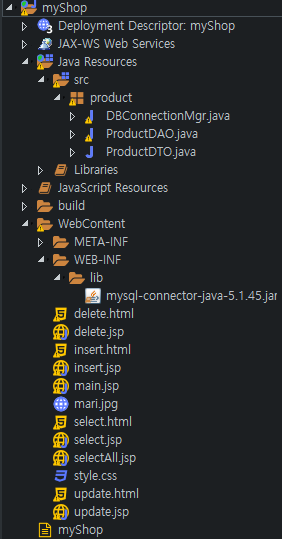
url database이름 으로 수정하고, user name과 password 입력

5) github주소와 화면캡쳐 프로젝트 생성 후, 파일 올리는 과정 캡쳐





6) 프로젝트 tree화면 캡쳐

****

7) 소스

DBConnectionMgr.java

/\*\*

\* Copyright(c) 2001 iSavvix Corporation (http://www.isavvix.com/)

\*

\* All rights reserved

\*

\* Permission to use, copy, modify and distribute this material for

\* any purpose and without fee is hereby granted, provided that the

\* above copyright notice and this permission notice appear in all

\* copies, and that the name of iSavvix Corporation not be used in

\* advertising or publicity pertaining to this material without the

\* specific, prior written permission of an authorized representative of

\* iSavvix Corporation.

\*

\* ISAVVIX CORPORATION MAKES NO REPRESENTATIONS AND EXTENDS NO WARRANTIES,

\* EXPRESS OR IMPLIED, WITH RESPECT TO THE SOFTWARE, INCLUDING, BUT

\* NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND

\* FITNESS FOR ANY PARTICULAR PURPOSE, AND THE WARRANTY AGAINST

\* INFRINGEMENT OF PATENTS OR OTHER INTELLECTUAL PROPERTY RIGHTS. THE

\* SOFTWARE IS PROVIDED "AS IS", AND IN NO EVENT SHALL ISAVVIX CORPORATION OR

\* ANY OF ITS AFFILIATES BE LIABLE FOR ANY DAMAGES, INCLUDING ANY

\* LOST PROFITS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES RELATING

\* TO THE SOFTWARE.

\*

\*/

package product;

import java.sql.\*;

import java.util.Properties;

import java.util.Vector;

public class DBConnectionMgr {

private Vector connections = new Vector(10); //connection list를 만들어줌 현재 10개

private String \_driver = "com.mysql.jdbc.Driver",

\_url = "jdbc:mysql://localhost:3306/myShop",

\_user = "root",

\_password = "1234";

private boolean \_traceOn = false;

private boolean initialized = false;

private int \_openConnections = 10;

private static DBConnectionMgr instance = null;

public DBConnectionMgr() {

}

/\*\* Use this method to set the maximum number of open connections before

unused connections are closed.

\*/

//싱글톤 코드( DBConnectionMgr은 객체가 생성될 때 connection을 10개를 만들어줌 )

public static DBConnectionMgr getInstance() { //DBconnectionMgr은 객체를 생성하고 분기문을 이용하여 한개만 사용하게끔 만듦( 있으면 있는 것을 끌어와 쓰고, 없으면 새로 만들어주는 것 )

if (instance == null) {

synchronized (DBConnectionMgr.class) {

if (instance == null) {

instance = new DBConnectionMgr();

}

}

}

return instance;

}

public void setOpenConnectionCount(int count) {

\_openConnections = count;

}

public void setEnableTrace(boolean enable) {

\_traceOn = enable;

}

/\*\* Returns a Vector of java.sql.Connection objects \*/

public Vector getConnectionList() {

return connections;

}

/\*\* Opens specified "count" of connections and adds them to the existing pool \*/

public synchronized void setInitOpenConnections(int count)

throws SQLException {

Connection c = null;

ConnectionObject co = null;

for (int i = 0; i < count; i++) {

c = createConnection();

co = new ConnectionObject(c, false);

connections.addElement(co); //vector에 하나씩 클래스를 만들어서 넣어줌

trace("ConnectionPoolManager: Adding new DB connection to pool (" + connections.size() + ")");

}

}

/\*\* Returns a count of open connections \*/

public int getConnectionCount() {

return connections.size();

}

/\*\* Returns an unused existing or new connection. \*/

public synchronized Connection getConnection() //getConnection을 이용하여 주소를 받아옴(반환값이 Connection인 것을 볼 수 있음)

throws Exception {

if (!initialized) {

Class c = Class.forName(\_driver);

DriverManager.registerDriver((Driver) c.newInstance());

initialized = true;

}

Connection c = null;

ConnectionObject co = null;

boolean badConnection = false;

for (int i = 0; i < connections.size(); i++) {

co = (ConnectionObject) connections.get(i);

// If connection is not in use, test to ensure it's still valid!

if (!co.inUse) {

try {

badConnection = co.connection.isClosed();

if (!badConnection)

badConnection = (co.connection.getWarnings() != null);

} catch (Exception e) {

badConnection = true;

e.printStackTrace();

}

// Connection is bad, remove from pool

if (badConnection) {

connections.removeElementAt(i);

trace("ConnectionPoolManager: Remove disconnected DB connection #" + i);

continue;

}

c = co.connection;

co.inUse = true;

trace("ConnectionPoolManager: Using existing DB connection #" + (i + 1));

break;

}

}

if (c == null) {

c = createConnection();

co = new ConnectionObject(c, true);

connections.addElement(co);

trace("ConnectionPoolManager: Creating new DB connection #" + connections.size());

}

return c;

}

/\*\* Marks a flag in the ConnectionObject to indicate this connection is no longer in use \*/

public synchronized void freeConnection(Connection c) {

if (c == null)

return;

ConnectionObject co = null;

for (int i = 0; i < connections.size(); i++) {

co = (ConnectionObject) connections.get(i);

if (c == co.connection) {

co.inUse = false;

break;

}

}

for (int i = 0; i < connections.size(); i++) {

co = (ConnectionObject) connections.get(i);

if ((i + 1) > \_openConnections && !co.inUse)

removeConnection(co.connection);

}

}

public void freeConnection(Connection c, PreparedStatement p, ResultSet r) {

try {

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

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

freeConnection(c);

} catch (SQLException e) {

e.printStackTrace();

}

}

public void freeConnection(Connection c, Statement s, ResultSet r) {

try {

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

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

freeConnection(c);

} catch (SQLException e) {

e.printStackTrace();

}

}

public void freeConnection(Connection c, PreparedStatement p) {

try {

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

freeConnection(c);

} catch (SQLException e) {

e.printStackTrace();

}

}

public void freeConnection(Connection c, Statement s) {

try {

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

freeConnection(c);

} catch (SQLException e) {

e.printStackTrace();

}

}

/\*\* Marks a flag in the ConnectionObject to indicate this connection is no longer in use \*/

public synchronized void removeConnection(Connection c) {

if (c == null)

return;

ConnectionObject co = null;

for (int i = 0; i < connections.size(); i++) {

co = (ConnectionObject) connections.get(i);

if (c == co.connection) {

try {

c.close();

connections.removeElementAt(i);

trace("Removed " + c.toString());

} catch (Exception e) {

e.printStackTrace();

}

break;

}

}

}

private Connection createConnection()

throws SQLException {

Connection con = null;

try {

if (\_user == null)

\_user = "";

if (\_password == null)

\_password = "";

Properties props = new Properties();

props.put("user", \_user);

props.put("password", \_password);

con = DriverManager.getConnection(\_url, props);

} catch (Throwable t) {

throw new SQLException(t.getMessage());

}

return con;

}

/\*\* Closes all connections and clears out the connection pool \*/

public void releaseFreeConnections() {

trace("ConnectionPoolManager.releaseFreeConnections()");

Connection c = null;

ConnectionObject co = null;

for (int i = 0; i < connections.size(); i++) {

co = (ConnectionObject) connections.get(i);

if (!co.inUse)

removeConnection(co.connection);

}

}

/\*\* Closes all connections and clears out the connection pool \*/

public void finalize() {

trace("ConnectionPoolManager.finalize()");

Connection c = null;

ConnectionObject co = null;

for (int i = 0; i < connections.size(); i++) {

co = (ConnectionObject) connections.get(i);

try {

co.connection.close();

} catch (Exception e) {

e.printStackTrace();

}

co = null;

}

connections.removeAllElements();

}

private void trace(String s) {

if (\_traceOn)

System.err.println(s);

}

}

class ConnectionObject {

public java.sql.Connection connection = null;

public boolean inUse = false;

public ConnectionObject(Connection c, boolean useFlag) {

connection = c;

inUse = useFlag;

}

}

ProductDAO.java

package product;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import product.ProductDTO;

public class ProductDAO {

DBConnectionMgr mgr;

public ProductDAO(){

mgr = DBConnectionMgr.getInstance();

}

// 제너릭 프로그래밍(객체 생성시 타입을 결정 가능)

// 형변환 필요 없어서 내부적인 처리 속도 더 빠름

public ProductDTO select(ProductDTO dto) throws Exception {

//1,2단계를 해주는 DBConnectinMgr객체 필요

Connection con = mgr.getConnection();

//3단계 sql문 결정

String sql = "select \* from product where id = ?";

PreparedStatement ps = con.prepareStatement(sql);

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

//4단계 sql문 전달 요청

ResultSet rs = ps.executeQuery();

ProductDTO dto2 = null;

while(rs.next()) {

dto2 = new ProductDTO();

String id = rs.getString(1);

String title = rs.getString(2);

String content = rs.getString(3);

String price = rs.getString(4);

dto2.setId(id);

dto2.setTitle(title);

dto2.setContent(content);

dto2.setPrice(price);

}

return dto2;

}

public void update(ProductDTO dto) throws Exception {

//1,2단계를 해주는 DBConnectinMgr객체 필요

Connection con = mgr.getConnection();

//3단계 sql문 결정

String sql = "update product set id = ?, title =?, content = ?, price = ? where id = ?";

PreparedStatement ps = con.prepareStatement(sql);

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

ps.setString(2, dto.getTitle());

ps.setString(3, dto.getContent());

ps.setString(4, dto.getPrice());

ps.setString(5, dto.getId());

//4단계 sql문 전달 요청

ps.executeUpdate();

}

public void delete(ProductDTO dto) throws Exception {

//1,2단계를 해주는 DBConnectinMgr객체 필요

Connection con = mgr.getConnection();

//3단계 sql문 결정

String sql = "delete from product where id=?";

PreparedStatement ps = con.prepareStatement(sql);

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

//4단계 sql문 전달 요청

ps.executeUpdate();

mgr.freeConnection(con, ps);

}

public void insert(ProductDTO dto) throws Exception {

//1,2단계를 해주는 DBConnectinMgr객체 필요

Connection con = mgr.getConnection();

//3단계 sql문 결정

String sql = "insert into product values (?,?,?,?)";

PreparedStatement ps = con.prepareStatement(sql);

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

ps.setString(2, dto.getTitle());

ps.setString(3, dto.getContent());

ps.setString(4, dto.getPrice());

//4단계 sql문 전달 요청

ps.executeUpdate();

mgr.freeConnection(con, ps);

}

public ArrayList<ProductDAO> selectAll() throws Exception {

ArrayList list = new ArrayList();

Connection con = mgr.getConnection();

ProductDTO dto = null; //변수의 생존범위 = 선언의 위치

//3. SQL문 결정(객체화)

String sql = "select \* from product";

PreparedStatement ps = con.prepareStatement(sql);

System.out.println("SQL문 객체화 OK");

//4. SQL문 전송

ResultSet rs = ps.executeQuery();

while(rs.next()) {

dto = new ProductDTO();

String id = rs.getString(1);

String title = rs.getString(2);

String content = rs.getString(3);

String price = rs.getString(4);

dto.setId(id);

dto.setTitle(title);

dto.setContent(content);

dto.setPrice(price);

list.add(dto);

}

System.out.println("SQL문 전송 OK");

try {

rs.close();

ps.close();

con.close();

} catch (SQLException e) {

//e.printStackTrace();

System.out.println("자원 해제 중 에러 발생!!");

}

return list;

}

}

ProductDTO.java

package product;

public class ProductDTO {

String id;

String title;

String content;

String price;

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getContent() {

return content;

}

public void setContent(String content) {

this.content = content;

}

public String getPrice() {

return price;

}

public void setPrice(String price) {

this.price = price;

}

}

delete.html

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

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

</head>

<body>

<center>

<h1>상품 삭제</h1>

<table id="main\_table">

<tr>

<a href="main.jsp">홈 | </a>

</tr>

<tr>

<a href="insert.html">상품 등록 | </a>

</tr>

<tr>

<a href="select.html">상품 검색 | </a>

</tr>

<tr>

<a href="selectAll.jsp">상품 전체 검색 | </a>

</tr>

<tr>

<a href="update.html">상품 수정 | </a>

</tr>

<tr>

<a href="delete.html">상품 삭제</a>

</tr>

</table>

<form action="delete.jsp">

삭제할 상품의 아이디를 입력하세요<br><br>

<div>

<input type="text" name="id"><br><br>

<div>

<button type="submit">삭제</button>

</div>

</div>

</form>

</center>

</body>

</html>

delete.jsp

<%@page import="product.ProductDTO"%>

<%@page import="product.ProductDAO"%>

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

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

</head>

<body>

<jsp:useBean id="dto" class="product.ProductDTO"></jsp:useBean>

<!-- import + new -->

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

<%

ProductDAO dao = new ProductDAO();

dao.delete(dto);

%>

<center>

<h1>상품 삭제</h1>

<table id="main\_table">

<tr>

<a href="main.jsp">홈 | </a>

</tr>

<tr>

<a href="insert.html">상품 등록 | </a>

</tr>

<tr>

<a href="select.html">상품 검색 | </a>

</tr>

<tr>

<a href="selectAll.jsp">상품 전체 검색 | </a>

</tr>

<tr>

<a href="update.html">상품 수정 | </a>

</tr>

<tr>

<a href="delete.html">상품 삭제</a>

</tr>

</table>

<h2><%= dto.getId() %></h2> 의 상품이 삭제 되었습니다 :)

<button type="button"><a href="main.jsp">홈으로 가기</a></button>

</center>

</body>

</html>

insert.html

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

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

</head>

<body>

<center>

<div>

<h1>상품 등록</h1>

<table id="main\_table">

<tr>

<a href="main.jsp">홈 | </a>

</tr>

<tr>

<a href="insert.html">상품 등록 | </a>

</tr>

<tr>

<a href="select.html">상품 검색 | </a>

</tr>

<tr>

<a href="selectAll.jsp">상품 전체 검색 | </a>

</tr>

<tr>

<a href="update.html">상품 수정 | </a>

</tr>

<tr>

<a href="delete.html">상품 삭제</a>

</tr>

</table>

<form action="insert.jsp" >

<table id="insert\_table">

<tr>

<td>상품 아이디</td>

<td><input type="text" name="id"></td>

</tr>

<tr>

<td>상품 설명 제목</td>

<td><input type="text" name="title"></td>

</tr>

<tr>

<td>상품 설명 내용</td>

<td><input type="text" name="content"></td>

</tr>

<tr>

<td>상품 가격</td>

<td><input type="text" name="price"></td>

</tr>

</table>

<br>

<button type="submit">등록</button>

</form>

</div>

</center>

</body>

</html>

insert.jsp

<%@page import="product.ProductDTO"%>

<%@page import="product.ProductDAO"%>

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

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

</head>

<body>

<jsp:useBean id="dto" class="product.ProductDTO"></jsp:useBean>

<!-- import + new -->

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

<%

ProductDAO dao = new ProductDAO();

dao.insert(dto);

%>

<center>

<h1>상품 삽입</h1>

<table id="main\_table">

<tr>

<a href="main.jsp">홈 | </a>

</tr>

<tr>

<a href="insert.html">상품 등록 | </a>

</tr>

<tr>

<a href="select.html">상품 검색 | </a>

</tr>

<tr>

<a href="selectAll.jsp">상품 전체 검색 | </a>

</tr>

<tr>

<a href="update.html">상품 수정 | </a>

</tr>

<tr>

<a href="delete.html">상품 삭제</a>

</tr>

</table>

상품 등록 완료!

<br>

<button type="button"><a href="main.jsp">홈으로 가기</a></button>

</center>

</body>

</html>

main.jsp

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

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

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

</head>

<body>

<center>

<h1>마리 쇼핑몰</h1>

<table id="main\_table">

<tr>

<a href="main.jsp">홈 | </a>

</tr>

<tr>

<a href="insert.html">상품 등록 | </a>

</tr>

<tr>

<a href="select.html">상품 검색 | </a>

</tr>

<tr>

<a href="selectAll.jsp">상품 전체 검색 | </a>

</tr>

<tr>

<a href="update.html">상품 수정 | </a>

</tr>

<tr>

<a href="delete.html">상품 삭제</a>

</tr>

</table>

<img src="mari.jpg" width="500" heigth="400">

</center>

</body>

</html>

select.html

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

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

</head>

<body>

<center>

<h1>상품 검색</h1>

<table id="main\_table">

<tr>

<a href="main.jsp">홈 | </a>

</tr>

<tr>

<a href="insert.html">상품 등록 | </a>

</tr>

<tr>

<a href="select.html">상품 검색 | </a>

</tr>

<tr>

<a href="selectAll.jsp">상품 전체 검색 | </a>

</tr>

<tr>

<a href="update.html">상품 수정 | </a>

</tr>

<tr>

<a href="delete.html">상품 삭제</a>

</tr>

</table>

<form action="select.jsp">

검색할 상품의 아이디를 입력하세요<br><br>

<div>

<input type="text" name="id"><br><br>

<div>

<button type="submit">검색</button>

</div>

</div>

</form>

</center>

</body>

</html>

select.jsp

<%@page import="product.ProductDTO"%>

<%@page import="product.ProductDAO"%>

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

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

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

</head>

<body>

<jsp:useBean id="dto" class="product.ProductDTO"></jsp:useBean>

<!-- import + new -->

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

<%

ProductDAO dao = new ProductDAO();

ProductDTO dto2 = dao.select(dto);

%>

<center>

<h1>상품 검색</h1>

<table id="main\_table">

<tr>

<a href="main.jsp">홈 | </a>

</tr>

<tr>

<a href="insert.html">상품 등록 | </a>

</tr>

<tr>

<a href="select.html">상품 검색 | </a>

</tr>

<tr>

<a href="selectAll.jsp">상품 전체 검색 | </a>

</tr>

<tr>

<a href="update.html">상품 수정 | </a>

</tr>

<tr>

<a href="delete.html">상품 삭제</a>

</tr>

</table>

<h2><%= dto.getId() %></h2>로 검색된 결과 입니다 :)<br>

<table id="selectJsp\_table" border="1" align="center">

<tr>

<td>상품 아이디</td>

<td><%= dto2.getId() %></td>

</tr>

<tr>

<td>상품 설명 제목</td>

<td><%= dto2.getTitle() %></td>

</tr>

<tr>

<td>상품 설명 내용</td>

<td><%= dto2.getContent() %></td>

</tr>

<tr>

<td>상품 가격</td>

<td><%= dto2.getPrice() %></td>

</tr>

</table>

<br>

<button type="button"><a href="main.jsp">홈으로 가기</a></button>

</center>

</body>

</html>

selectAll.jsp

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

<%@page import="product.ProductDTO"%>

<%@page import="product.ProductDAO"%>

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

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

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

</head>

<body>

<center>

<h1>상품 전체 검색</h1>

<table>

<tr>

<a href="main.jsp">홈 | </a>

</tr>

<tr>

<a href="insert.html">상품 등록 | </a>

</tr>

<tr>

<a href="select.html">상품 검색 | </a>

</tr>

<tr>

<a href="selectAll.jsp">상품 전체 검색 | </a>

</tr>

<tr>

<a href="update.html">상품 수정 | </a>

</tr>

<tr>

<a href="delete.html">상품 삭제</a>

</tr>

</table>

<table id="selectAll\_table" border="1" align="center">

<tr>

<td>상품 아이디</td>

<td>상품 설명 제목</td>

<td>상품 설명 내용</td>

<td>상품 가격</td>

</tr>

<%

ProductDAO dao = new ProductDAO();

ProductDTO dto = new ProductDTO();

ArrayList list = dao.selectAll();

for(int i=0; i<list.size(); i++) {

dto = (ProductDTO)list.get(i);

%>

<tr>

<td><%= dto.getId() %></td>

<td><%= dto.getTitle() %></td>

<td><%= dto.getContent() %></td>

<td><%= dto.getPrice() %></td>

</tr>

<%

}

%>

</table>

</center>

</body>

</html>

style.css

@charset "UTF-8";

a {

text-decoration: none;

}

tr {

width: 600px;

}

#insert\_table {

border: 1px solid #000000;

}

#update\_table {

border: 1px solid #000000;

}

#selectAll\_table {

border-collapse: collapse;

border: 1px solid #000000;

}

#selectJsp\_table {

border-collapse: collapse;

border: 1px solid #000000;

width: 200px;

}

update.html

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

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

</head>

<body>

<center>

<h1>상품 수정</h1>

<table id="main\_table">

<tr>

<a href="main.jsp">홈 | </a>

</tr>

<tr>

<a href="insert.html">상품 등록 | </a>

</tr>

<tr>

<a href="select.html">상품 검색 | </a>

</tr>

<tr>

<a href="selectAll.jsp">상품 전체 검색 | </a>

</tr>

<tr>

<a href="update.html">상품 수정 | </a>

</tr>

<tr>

<a href="delete.html">상품 삭제</a>

</tr>

</table>

<form action="update.jsp">

<table id="update\_table">

<tr>

<td>수정할 상품 아이디</td>

<td><input type="text" name="id"></td>

</tr>

<tr>

<td>수정할 상품 설명 제목</td>

<td><input type="text" name="title"></td>

</tr>

<tr>

<td>수정할 상품 설명 내용</td>

<td><input type="text" name="content"></td>

</tr>

<tr>

<td>수정할 상품 가격</td>

<td><input type="text" name="price"></td>

</tr>

</table>

<br>

<button type="submit">수정</button>

</form>

</center>

</body>

</html>

update.jsp

<%@page import="product.ProductDAO"%>

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

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

</head>

<body>

<jsp:useBean id="dto" class="product.ProductDTO"></jsp:useBean>

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

<%

ProductDAO dao = new ProductDAO();

dao.update(dto);

%>

상품 정보 수정이 완료되었습니다.

<br>

<button type="button"><a href="main.jsp">홈으로 가기</a></button>

</body>

</html>