



# sql class day4



## 목차

### MySQL에서 CRUD 구현

데이터 베이스 (Data Base) 새로 만들기

테이블 생성하기

테이블 내 열 추가

데이터 中 시간 자동 입력

값 자동 증가

PRIMARY KEY 지정

헤이디 활용해서 쿼리문 작성

### MySQL 활용 CRUD

C R U D 중 C

C R U D 중 R

C R U D 중 U

C R U D 중 D

JSP Project 생성

JSP에서 빈번한 에러 (404, 500)

Dynamic Web Project 에서 에러 발생 시

### JSP Database 활용 CRUD

JDBC - Basic

JSP로 CRUD (Basic.ver)

C R U D 중 CREATE

기본구문

응용구문 - 물음표 사용

기본구문 전체 (강사님 자료)

C R U D 중 READ

READ1 ( 목록조회)

READ에서 눈여겨봐야 할 코드 부분 ( Resultset, executeQuery )

READ 를 위해 필요한 <출력코드>

READ1 의 전체코드

READ2 ( 특정 데이터 상세조회)

C R U D 중 UPDATE

C R U D 중 DELETE

커넥트 부분 독립시키기

JDBC - File

디비연결을 집어넣는 용도로 사용할 부분

C R U D 중 CREATE ( 개별파일 활용 )

C R U D 중 READ ( 개별파일 활용 )

C R U D 중 UPDATE ( 개별파일 활용 )

C R U D 중 DELETE ( 개별파일 활용 )

JDBC - Class로 분리

커넥션 담긴 클래스 전체코드

값을 입력받을 수 있는 input 화면 만들기

input.jsp 코드

[create.jsp 코드](#)

[read.jsp코드](#)

[detail.jsp 코드](#)



#### JSP의 DB연결방법 4가지

##### 1. 각각의 페이지에서 직접 DB 사용

- 자원 소모가 심함
- 코드 중복 발생 ( 매 페이지에서 중복적으로 DB를 연결했다 끊었다 반복 )

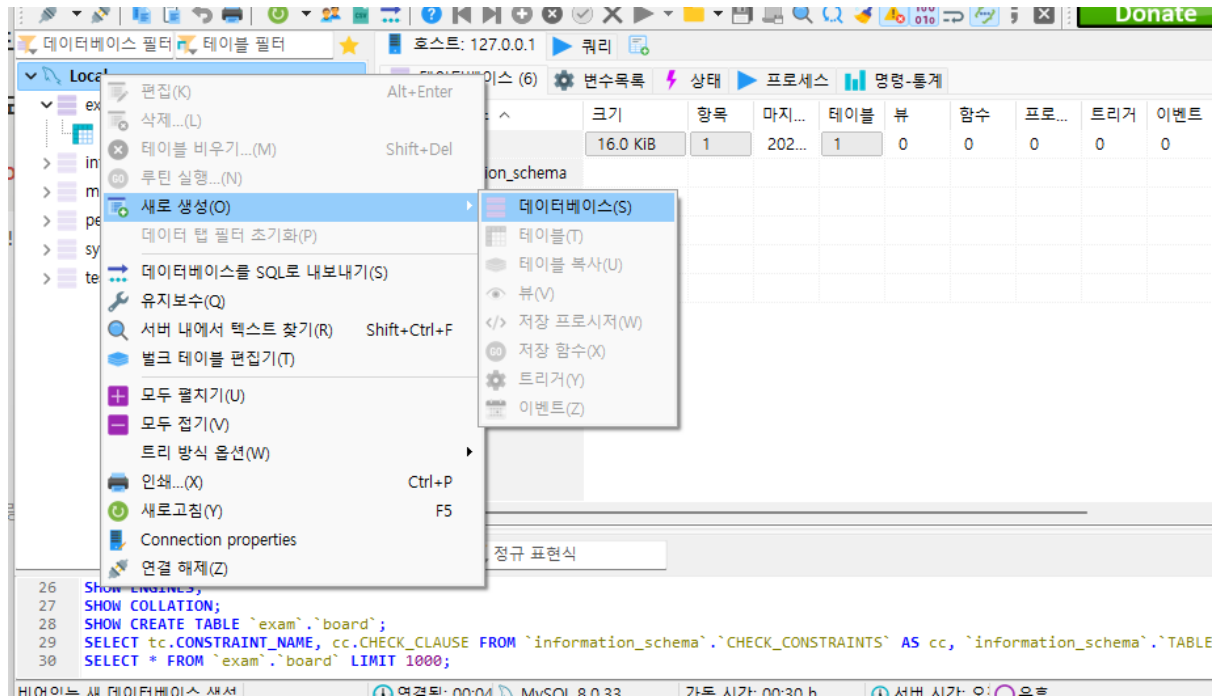
##### 2. DB연결만 별도로 공용으로 사용 , file로 구성

##### 3. DB 연결만 별도로 공용으로 사용, 자바 class로 구현

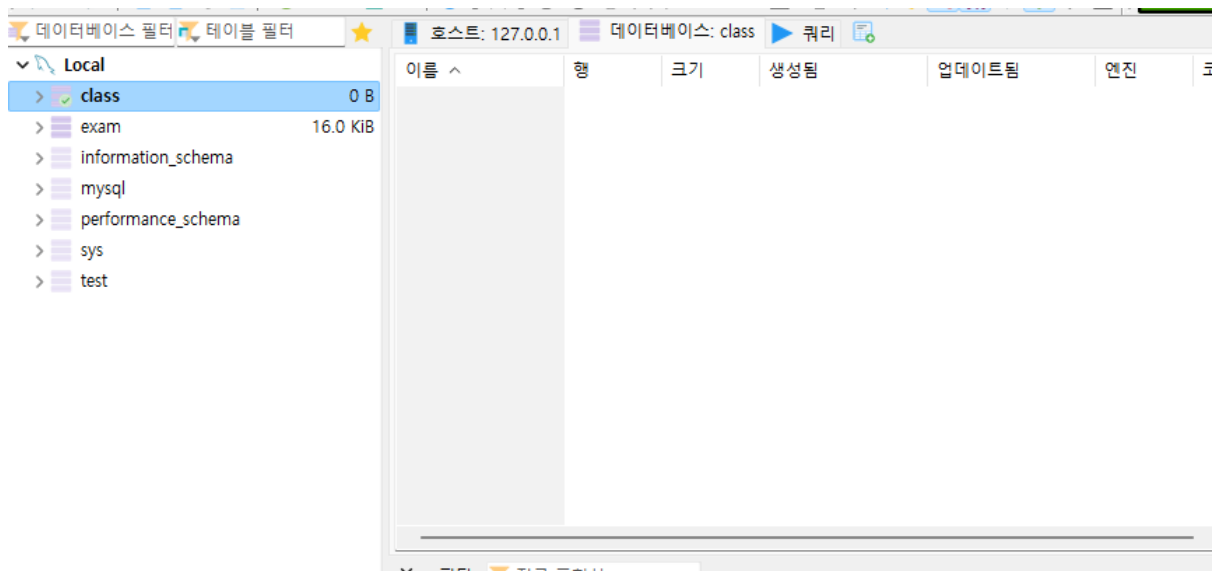
##### 4. **Connection Pool**로 DB 사용 [ 최종채택 ]

요건 day3에 봤던 내용이지!

## 데이터 베이스 (Data Base) 새로 만들기

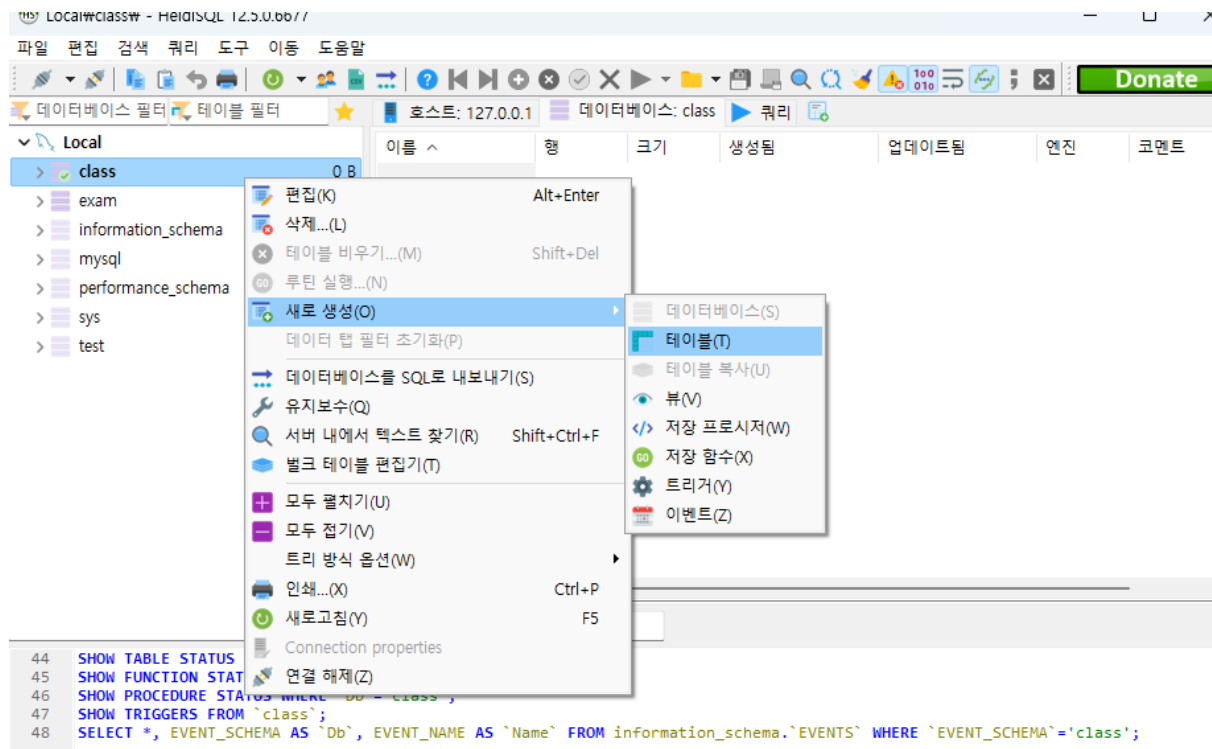


[ 세션 ] - [ 새로 생성 ] - [ 데이터베이스 ]

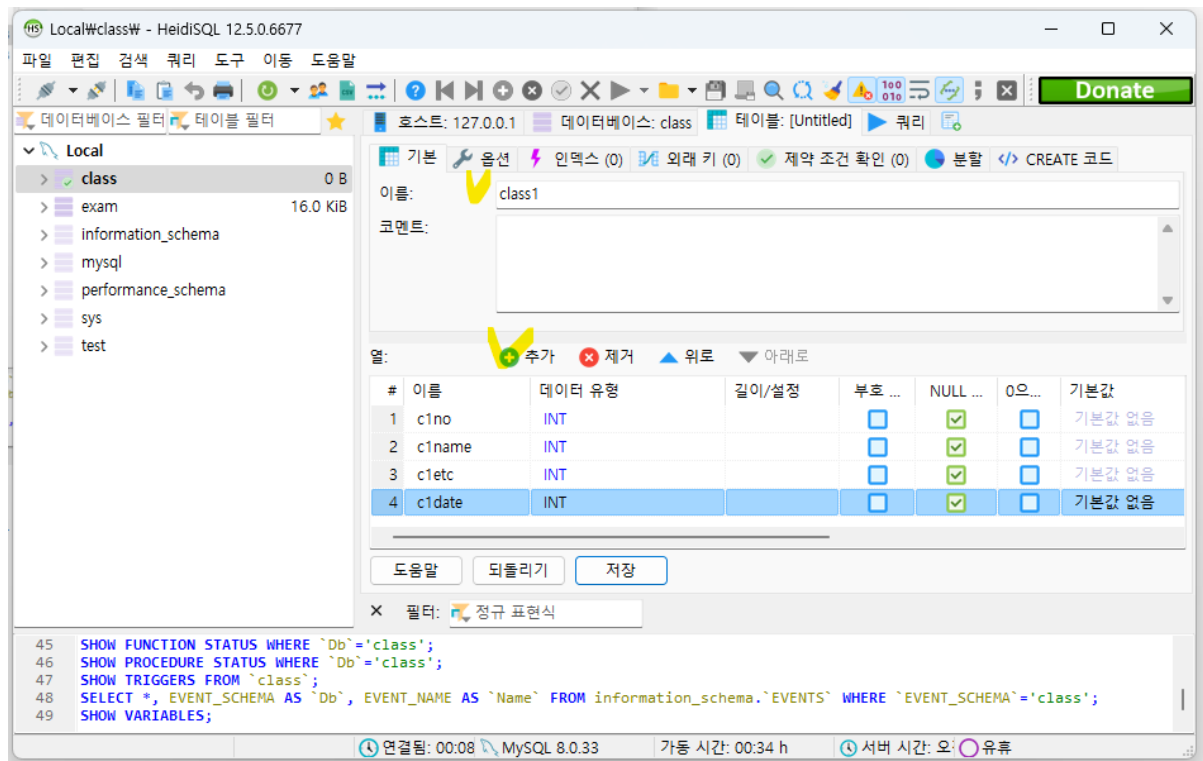


- DB는 일반적으로 업체에서 만들어 제공하거나, 지정해줌

# 테이블 생성하기

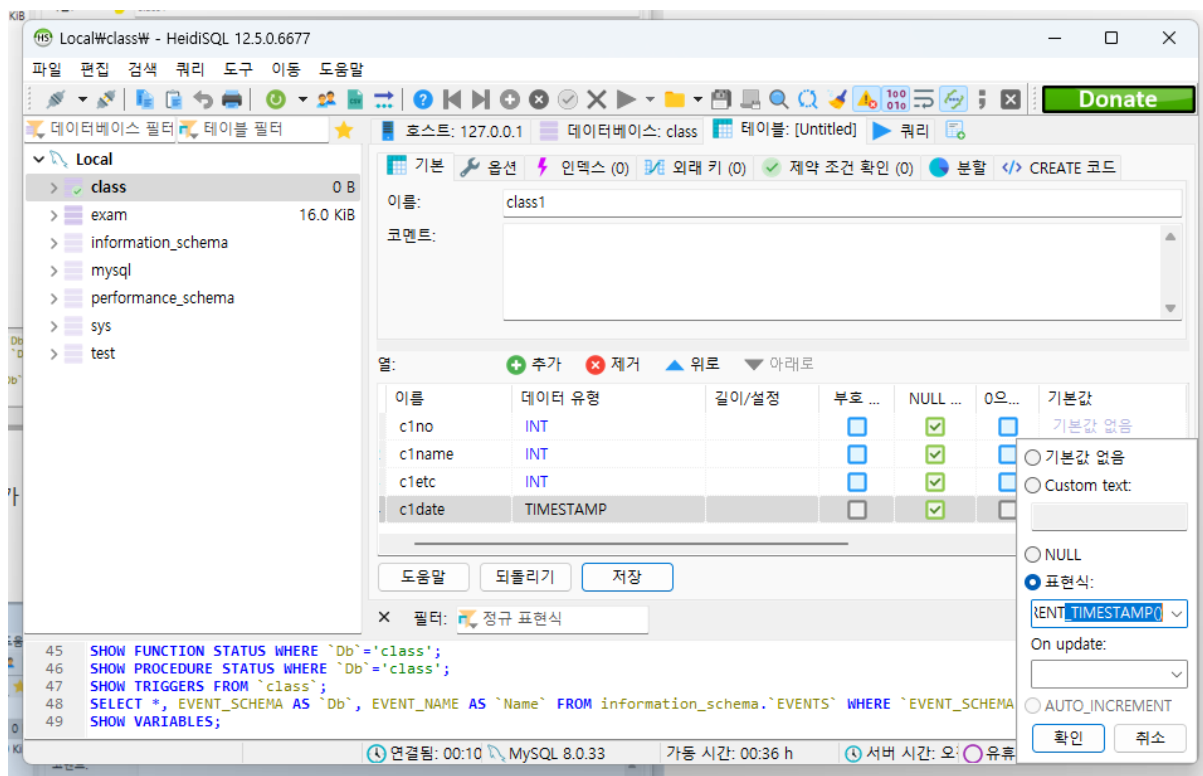
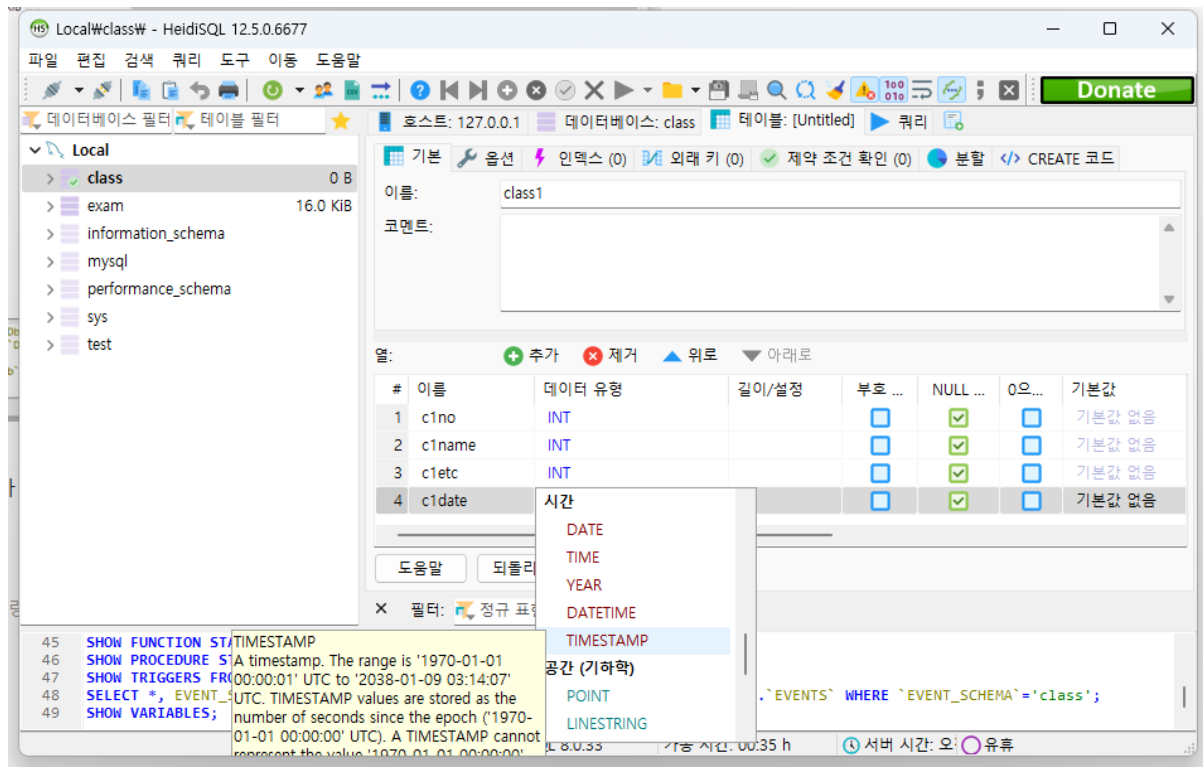


# 테이블 내 열 추가



추가 버튼 활용해서 열 추가

## 데이터 中 시간 자동 입력



3	c1etc	INT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	기본값 없음
4	c1date	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP

데이터 유형 ⇒ TIMESTAMP

표현식 ⇒ CURRENT ( 대문자로 )

## 값 자동 증가

열: + 추가 x 제거 ▲ 위로 ▼ 아래로

이름	데이터 유형	길이/설정	부호 ...	NULL ...	0으...	기본값
c1no	INT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/> 기본값 없음
c1name	INT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/> Custom text:
c1etc	INT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/> NULL
c1date	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/> 표현식:

도움말 되돌리기 저장

X 필터: 정규 표현식

```

='class';
='class';

EVENT_NAME AS `Name` FROM information_schema.`EVENTS` WHERE `EVENT_SCHEMA`

```

On update:

☒ **AUTO\_INCREMENT**

확인 취소

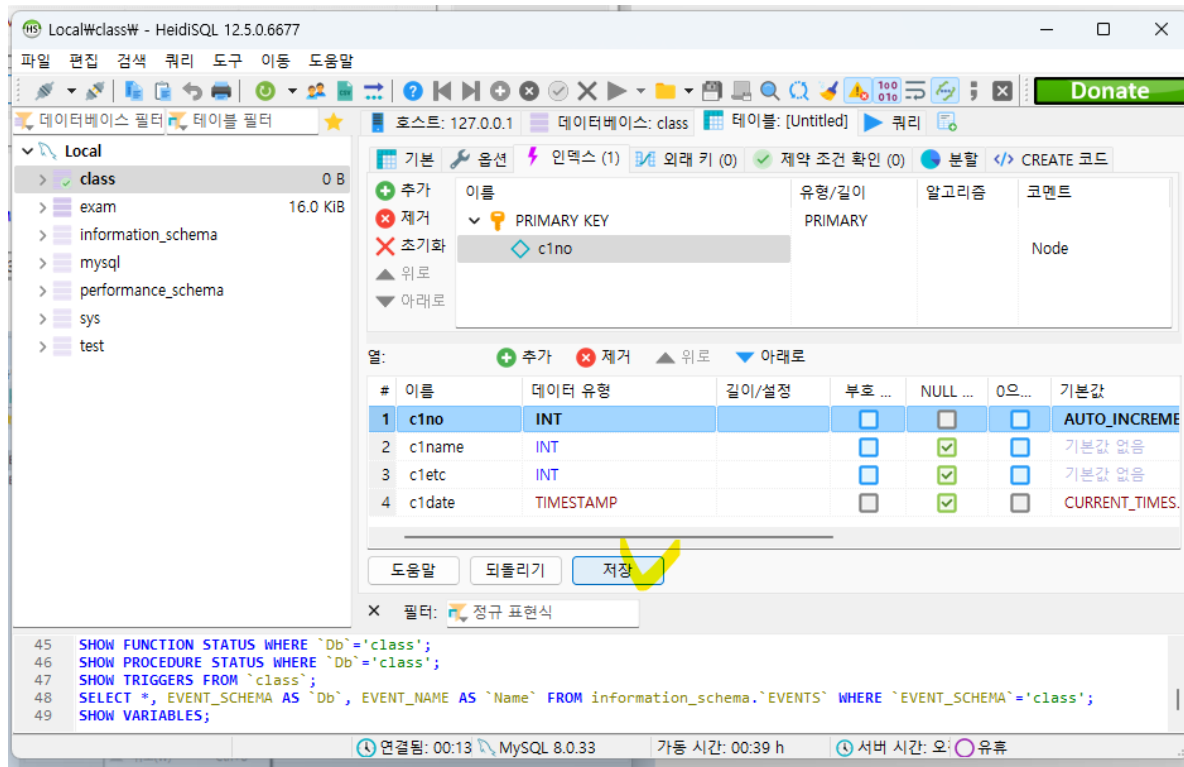
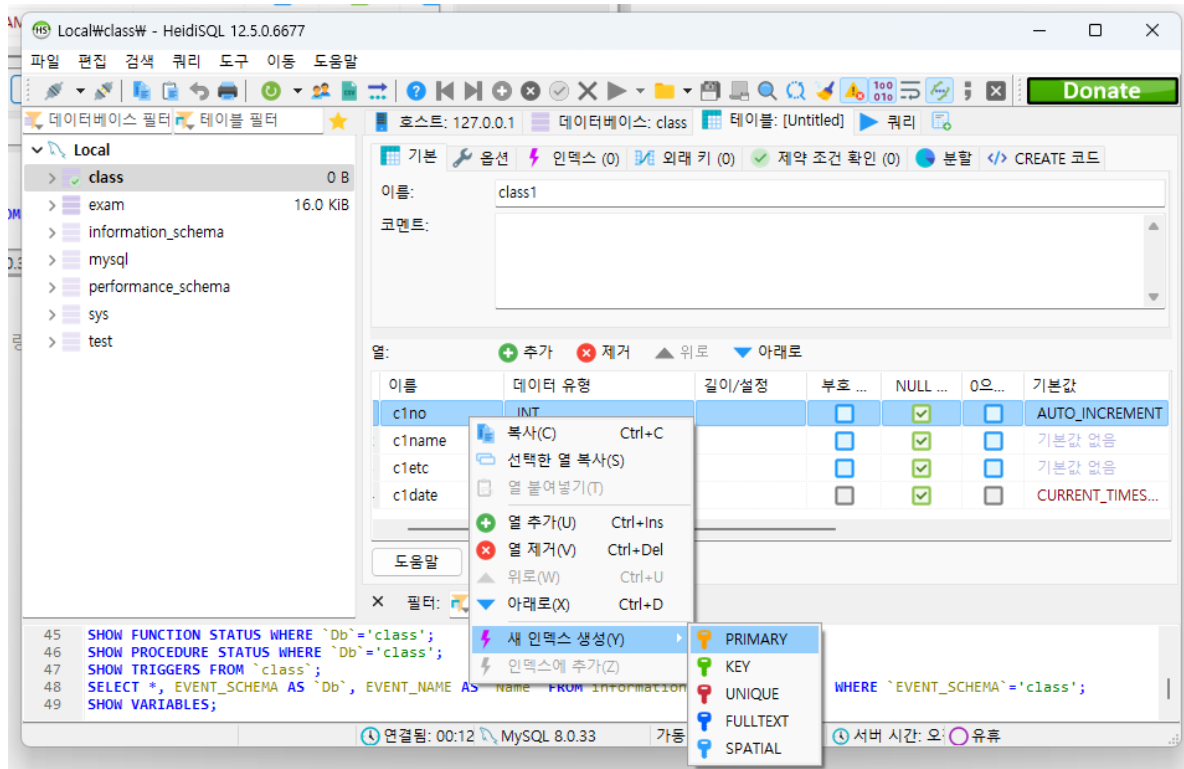
연결됨: 00:12 MySQL 8.0.33 가동 시간: 00:38 h 서버 시간: 오 휴

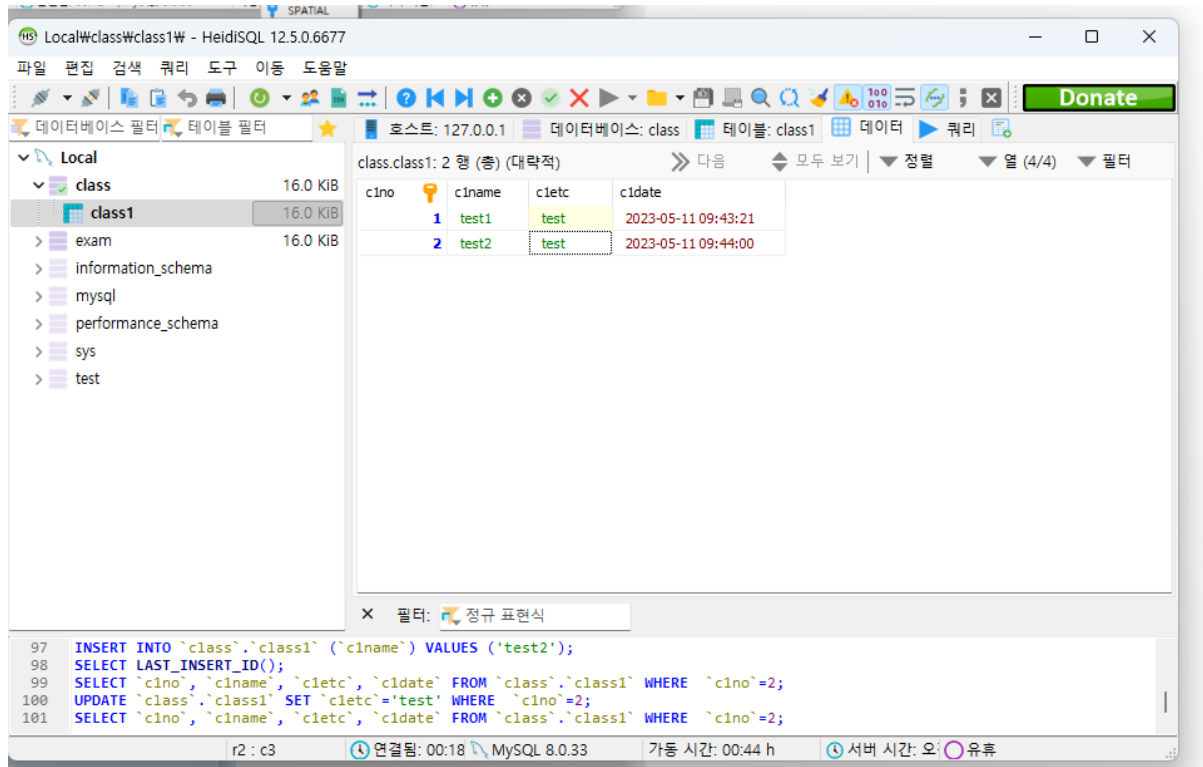
c1no 의 경우, 값이 자동으로 증가되게 하고 싶다면?

[ 기본값 ] - [ AUTO-INCREAMENT ]

## PRIMARY KEY 지정







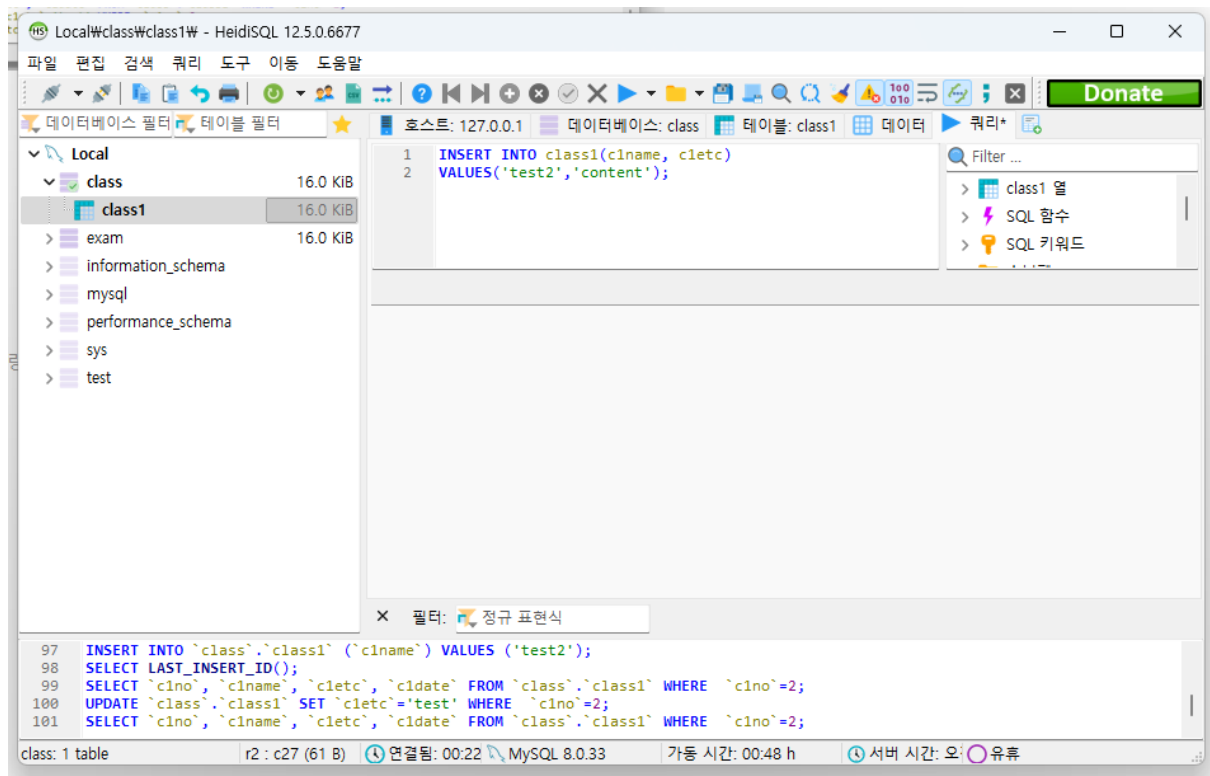
## 헤이디 활용해서 쿼리문 작성

⇒ 쿼리문 작성 후,  누르고 [ 데이터 ] 가서 실행 확인

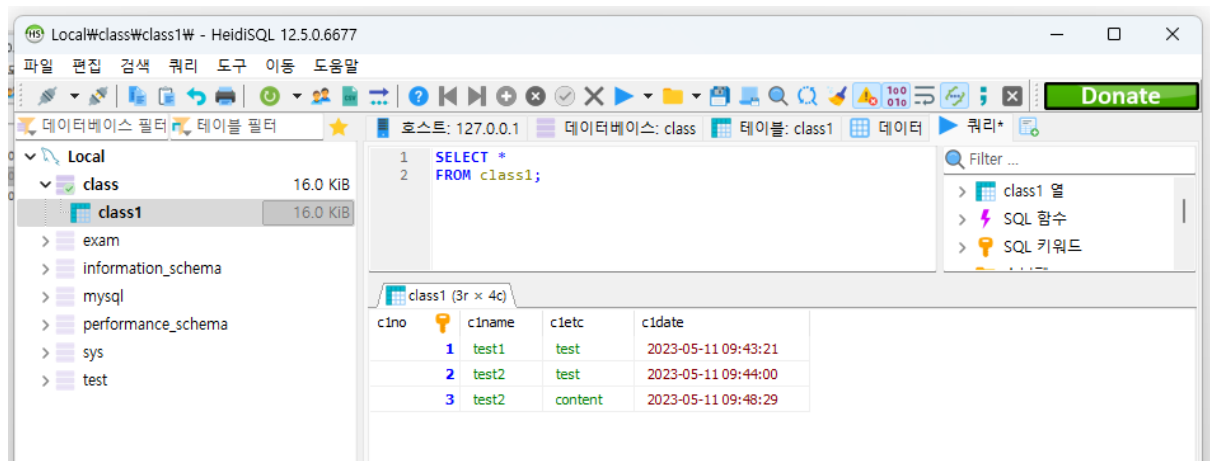
## MySQL 활용 CRUD

Basic

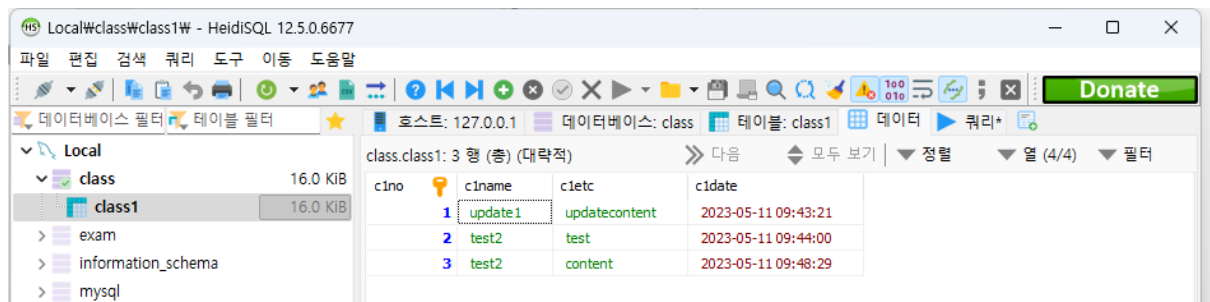
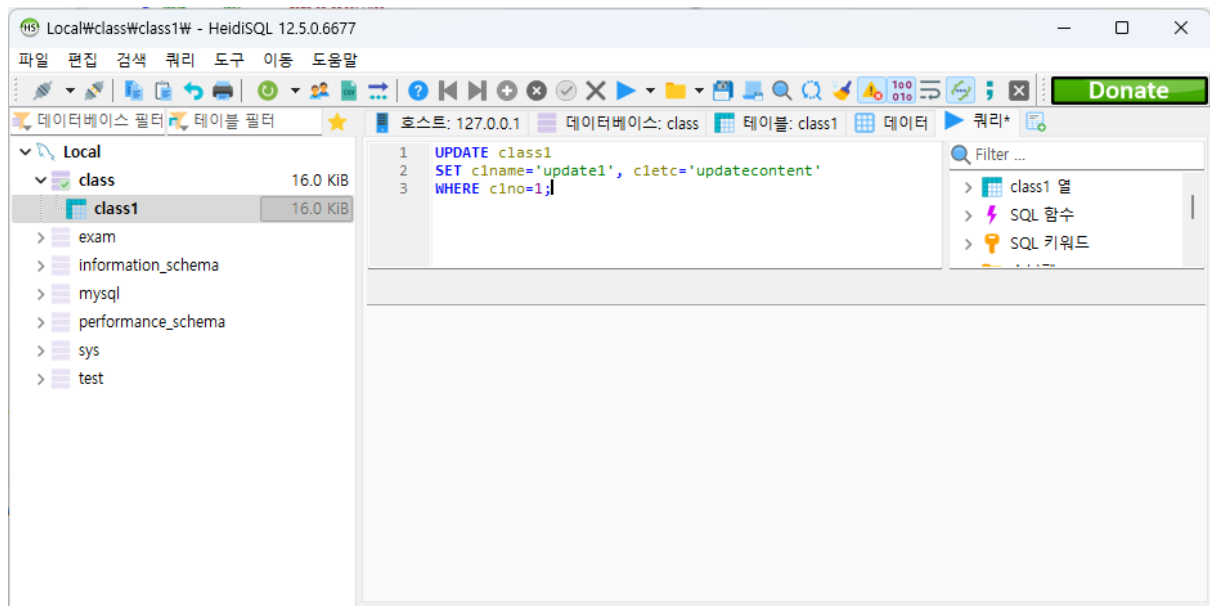
### CRUD 중 C



## CRUD중 R



## CRUD중 U



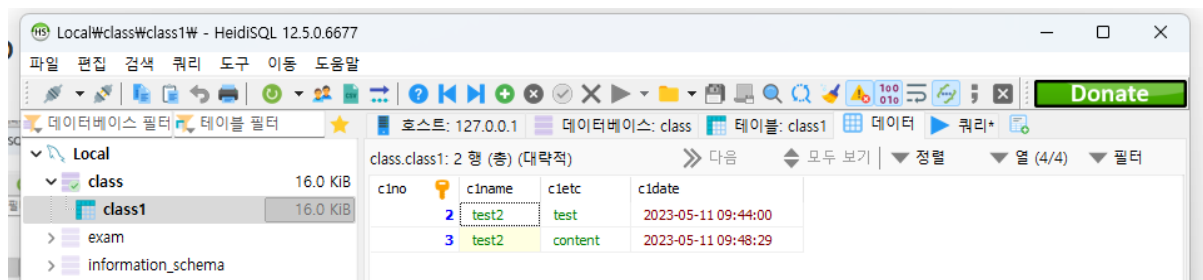
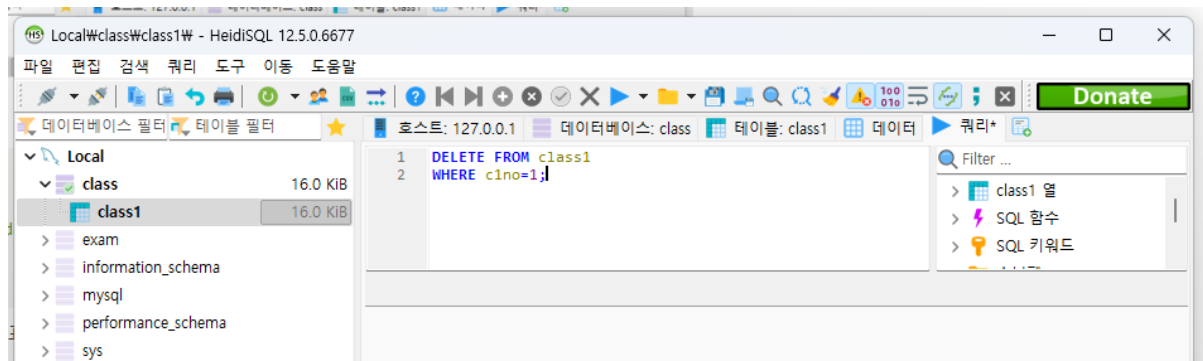
```

UPDATE class1
SET c1name='update1', c1etc='updatecontent'
WHERE c1no=1;

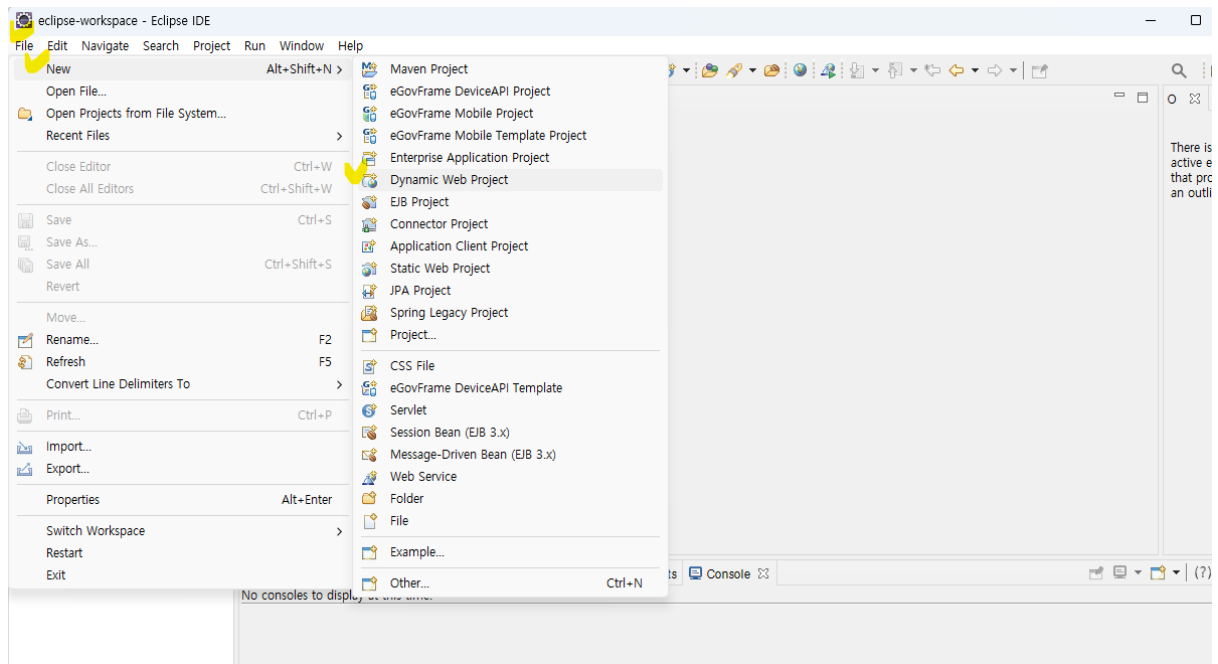
```

※ 자바에서는 물음표 사용해서 바꿀 수 있었음

## CRUD 중 D



## JSP Project 생성

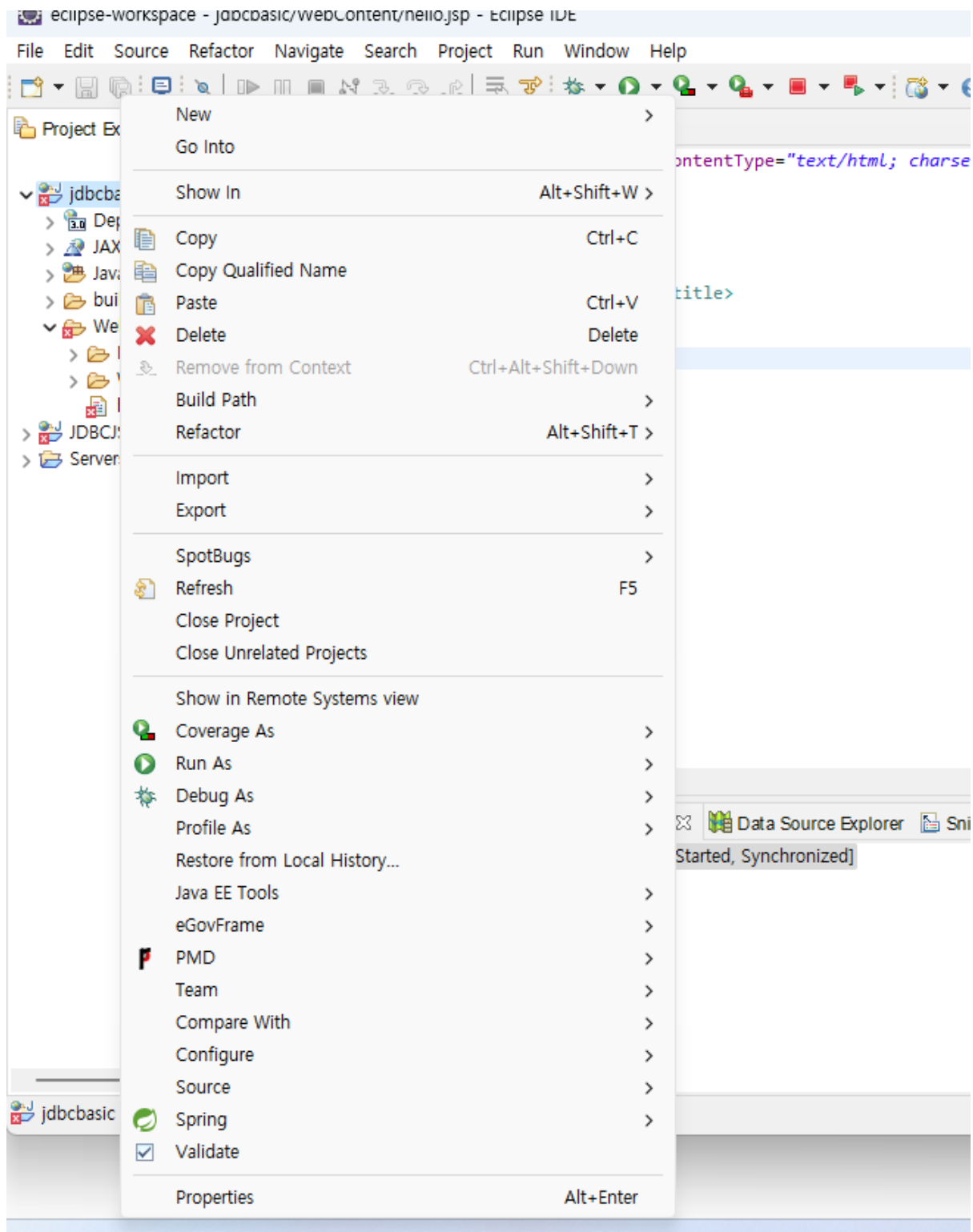


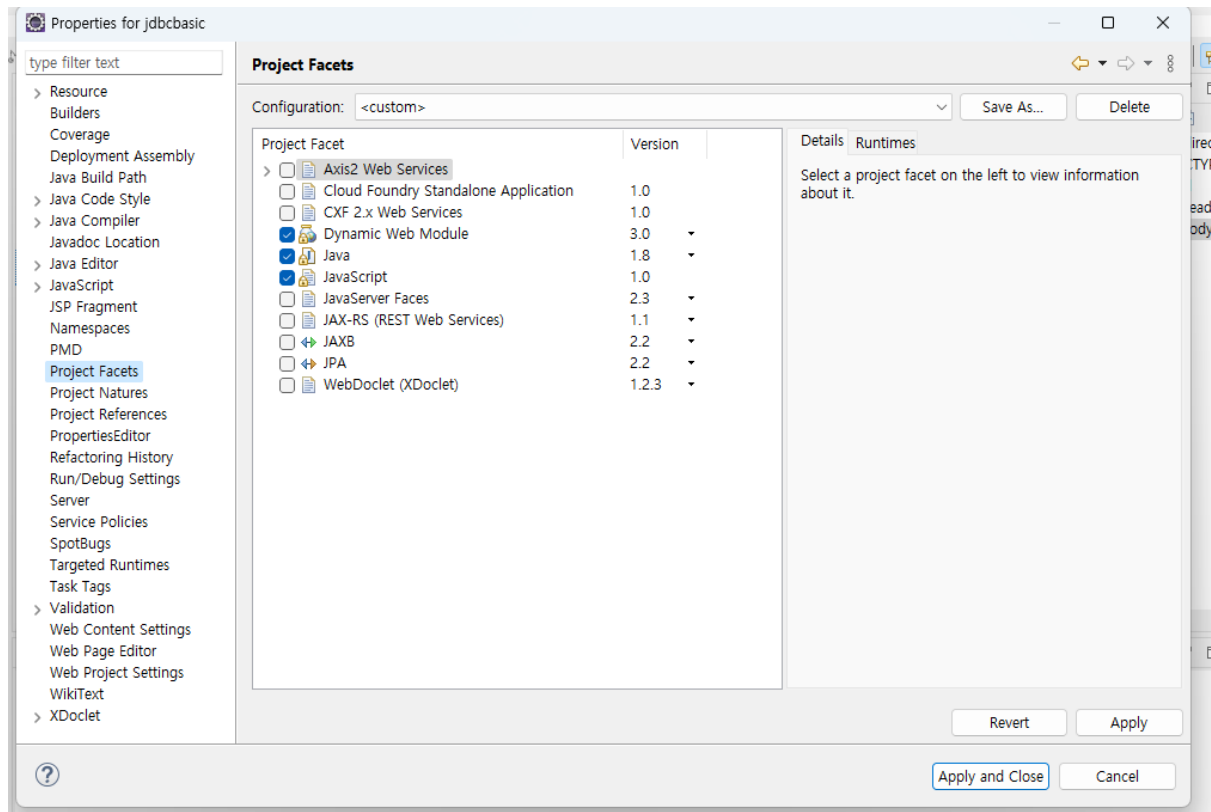
[ File ] - [ New ] - [ Dynamic Web Project ]

## JSP에서 빈번한 에러 (404, 500)

- 404 : 파일 경로를 찾을 수 없음 (Web Content 산하에 있는지 확인해야 함)
- 500 : 문법 에러

## Dynamic Web Project 에서 에러 발생 시

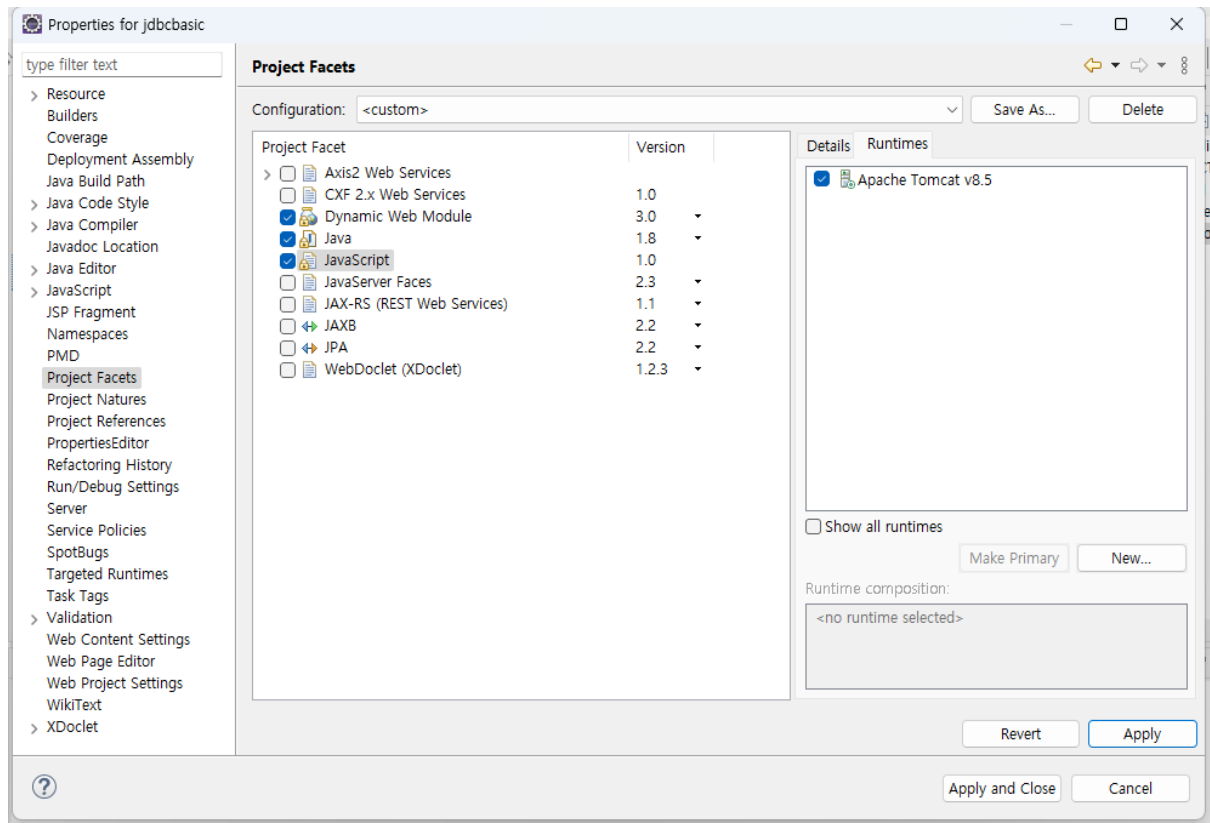




[ Project Facets ] 에서

자바버전이 1.8 맞는지 확인해야 함 → 우리가 설치해놓은 자바버전

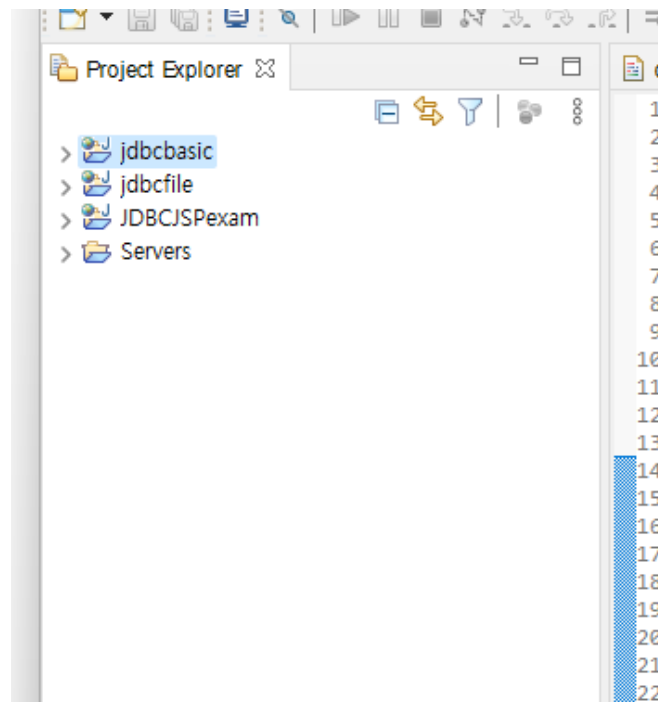
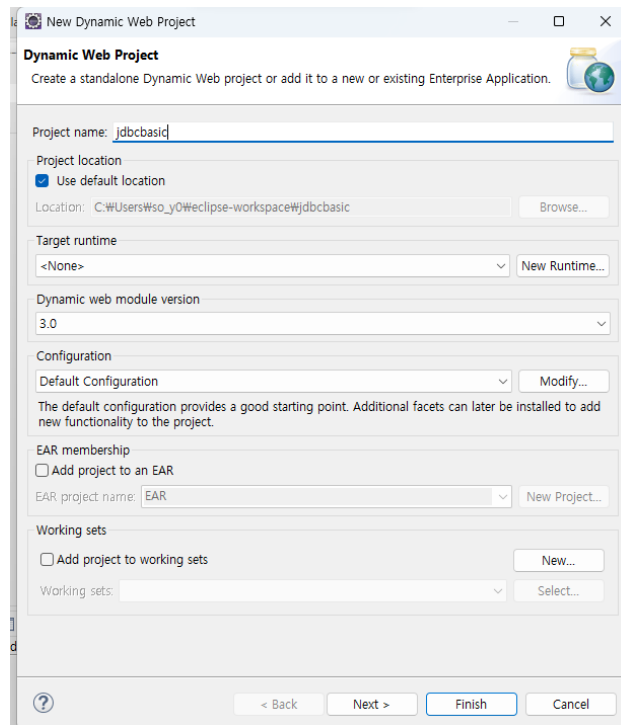




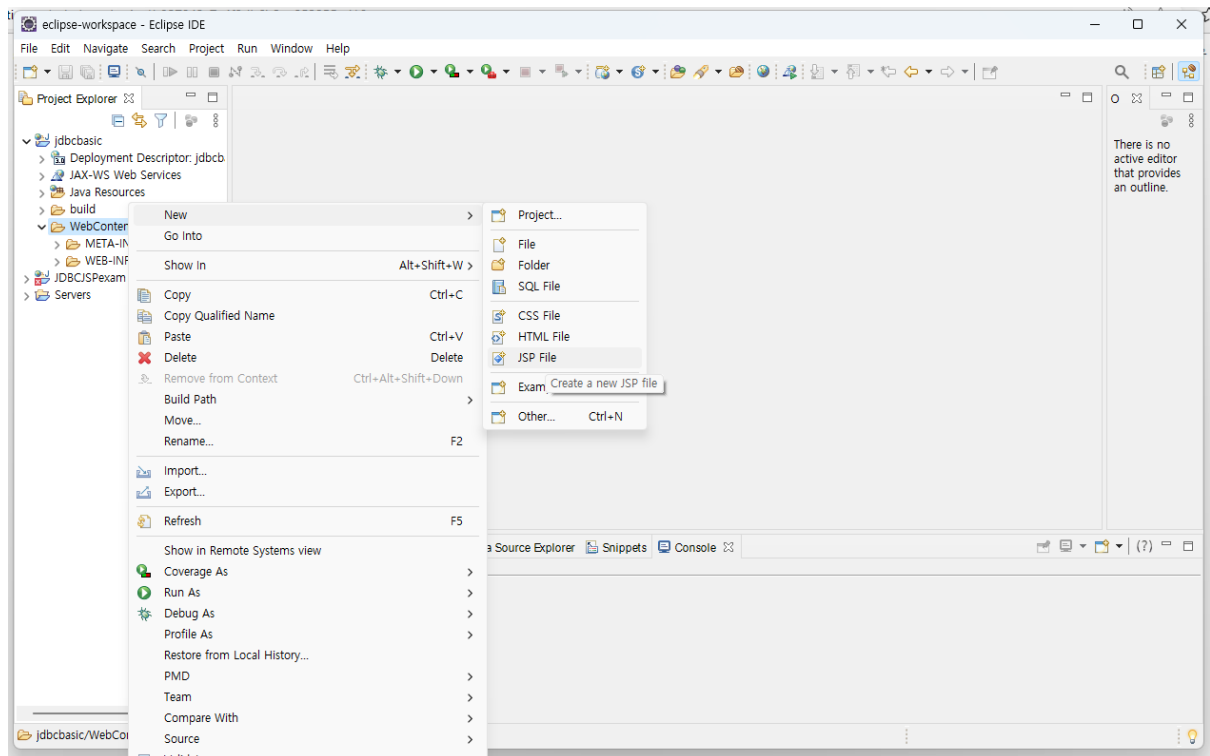
[ details ] 에서 톰캣 들어와있는지 확인 후, 없다면 체크 !

## JSP Database 활용 CRUD

### JDBC - Basic



프로젝트명 : jdbcbasic



[ Web Content ] 폴더 아래에 JSP project 생성

### [ 미니 과제 ]

class2 라는 테이블 만들고 !

class1과 동일하게 데이터값 넣기

## JSP로 CRUD (Basic.ver)



퀴리문 수정되지 않은 기본 틀 >> 외우기

```

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

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

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title> CREATE table class2 </title>
</head>
<body>

<%
    Class.forName("com.mysql.jdbc.Driver");

    Connection conn = null;

    String hostname = "jdbc:mysql://localhost:3306/class";
    String username = "root";
    String password = "0000";

    conn = DriverManager.getConnection(hostname, username, password);

%>

<%

    String sql = "";

    PreparedStatement pstmt = conn.prepareStatement(sql);

    pstmt.executeUpdate();

    pstmt.close();
    conn.close();

%>

</body>
</html>

```

## C R U D 중 CREATE

### 기본구문

```

String sql = "INSERT INTO class1 (c1name, c1etc)"
    + "VALUES ('temp', 'tempcontent')";

```

## 응용구문 - 물음표 사용

```
<%  
  
String sql = "INSERT INTO class1 (cname, c1etc) VALUES(?,?)";  
  
PreparedStatement pstmt = conn.prepareStatement(sql);  
    pstmt.setString(1, 'mmmm');  
    pstmt.setString(2, 'mcontent');  
  
pstmt.executeUpdate();  
  
%>
```

## 기본구문 전체 (강사님 자료)

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

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

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Create</title>
</head>
<body>

<%
    Class.forName("com.mysql.jdbc.Driver");

    Connection conn = null;

    String hostname = "jdbc:mysql://localhost:3306/class";

    String username = "root";

    String password = "0000";

    conn = DriverManager.getConnection(hostname, username, password);
%>

<%

    String sql = "INSERT INTO class1 (c1name, c1etc) VALUES('temp', 'tempcontent')";

    PreparedStatement pstmt = conn.prepareStatement(sql);

    pstmt.executeUpdate();

%>
```

```
</body>  
</html>
```

## C R U D 중 READ

READ1 과 READ2F로 나누어서 볼 건데,

### READ1 ( 목록조회)

#### READ에서 눈여겨봐야 할 코드 부분 ( ResultSet, executeQuery )

CRUD 중 R을 제외한 나머지는, 쿼리문을 던지면 끝이었음

( 실행이 성공했냐, 실패했냐 정도? )

그런데, READ에서는 정말 결과를 가져와야 함

```
26  
27  
28 <%  
29  
30     String sql = "SELECT * FROM class1";  
31  
32     PreparedStatement pstmt = conn.prepareStatement(sql);  
33  
34  
35     ResultSet rs = pstmt.executeQuery();  
36  
37  
38     pstmt.close();  
39     conn.close();  
40  
41 %>  
42
```

그 결과를 ResultSet 객체로 받음

⇒ ResultSet은 초반에 java.sql.\* 을 import해줬기 때문에 사용할 수 있는 것

pstmt.executeUpdate / pstmt.executeQuery 를 구별해야 함.

## READ 를 위해 필요한 <출력코드>

```
<%  
  
String sql = "SELECT * FROM class1";  
  
PreparedStatement pstmt = conn.prepareStatement(sql);  
  
ResultSet rset = pstmt.executeQuery();  
while(rset.next()){  
    1 String c1no = rset.getString("c1no");  
    String c1name = rset.getString("c1name");  
    String c1etc = rset.getString("c1etc");  
    String c1date = rset.getString("c1date");  
  
    2 out.print(c1no);  
    out.print(c1name);  
    out.print(c1etc);  
    out.print(c1date);  
};  
%>
```

ResultSet에 있는 여러 개의 데이터들을 출력할 건데,

while(rset.next) ⇒ **rset에 데이터가 있으면~** 출력해라

ResultSet 에는 회원의 데이터가 50만명짜리가 올 수도 있고, 100만 명짜리가 올 수도 있음

몇 개의 데이터가 들어있을지 모르기 때문에 next() 를 활용해서

데이터가 들어있는 횟수만큼 while문으로 반복



1

rset에 담겨있는 "c1no"라는 이름을 가진 것의 값을 String c1no에 담는 것



2

답은 것들을 출력

>> 근데 출력할 때 out.print( ); 사용하는 게 번거롭잖아?

이 때, JSP의 장점을 활용해서 !

자바코드와 html코드를 혼합해서 사용할 수 있음

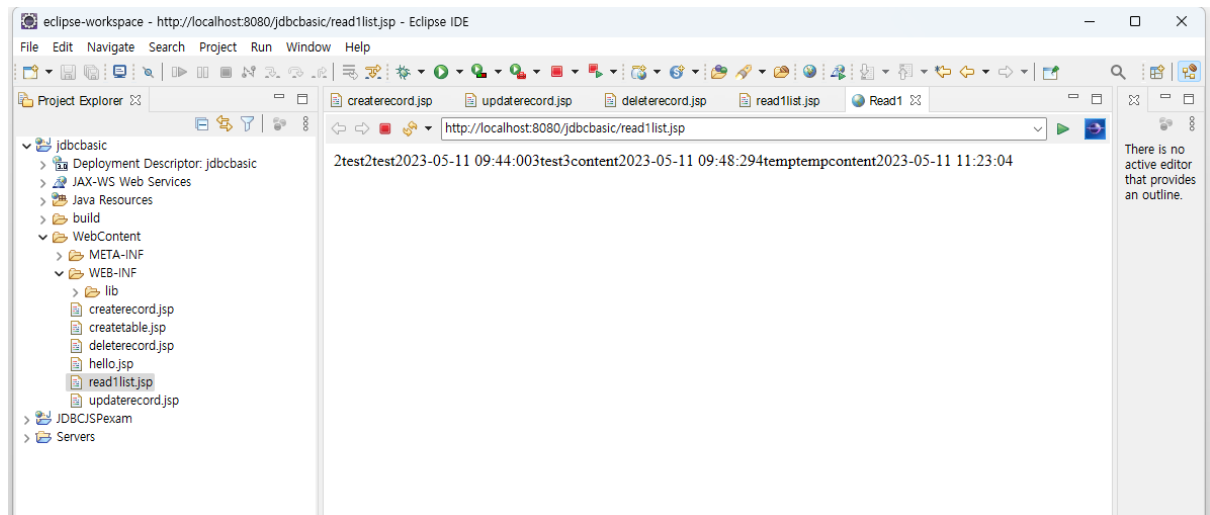
```
39      String c1no = rset.getString("c1no");
40      String c1name = rset.getString("c1name");
41      String c1etc = rset.getString("c1etc");
42      String c1date = rset.getString("c1date");
43
44
45      <%
46      <%= c1no %> <%= c1name %> <%= c1etc %> <%= c1date %> <br>
47
48
49
50
51
52      <%
53      }
54      <%
55
56  </body>
57  </html>
```

## - JSP에서의 스크립틀릿

<% %>	자바영역
일반 공간	HTML영역
<%= %>	자바 변수 출력 영역

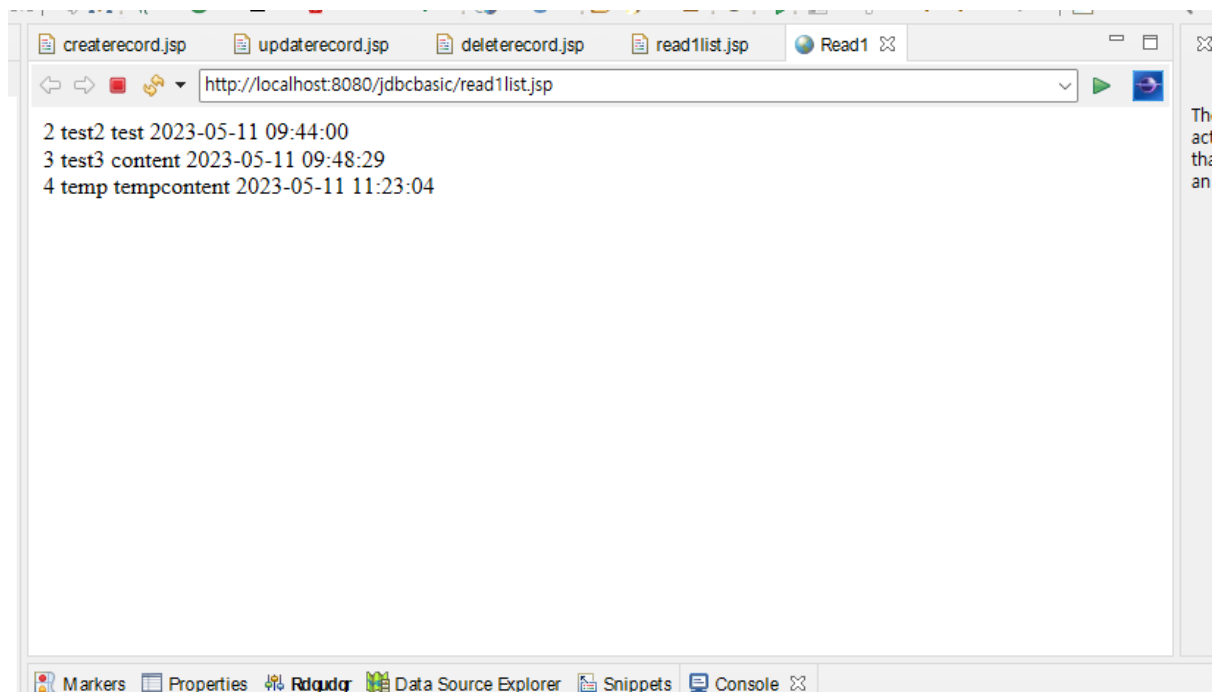
[ 결과 ] - out.print 활용 시





⇒ 모양 정돈은 안되었지만, 게시판에 있는 데이터들을 적절하게 가져온 것을 확인할 수 있음

[ 결과 ] - 자바코드와 html코드 혼용 시



⇒ HTML을 활용했기 때문에 어느정도 모양도 정돈됨

## READ1 의 전체코드

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

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

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title> Read1 </title>
</head>
<body>

<%
    Class.forName("com.mysql.jdbc.Driver");

    Connection conn = null;

    String hostname = "jdbc:mysql://localhost:3306/class";
    String username = "root";
    String password = "0000";

    conn = DriverManager.getConnection(hostname, username, password);

%>

<%

    String sql = "SELECT * FROM class1";

    PreparedStatement pstmt = conn.prepareStatement(sql);

    ResultSet rset = pstmt.executeQuery();

    while(rset.next()){

        String c1no = rset.getString("c1no");
        String c1name = rset.getString("c1name");
        String c1etc = rset.getString("c1etc");
        String c1date = rset.getString("c1date");

        out.print(c1no);
        out.print(c1name);
        out.print(c1etc);
        out.print(c1date);
    };

%>

</body>
</html>
```

## READ2 ( 특정 데이터 상세조회)

```
28 //
29
30 String sql = "SELECT * FROM class1 WHERE c1no = ?";
31
32 PreparedStatement pstmt = conn.prepareStatement(sql);
33 pstmt.setInt(1, 2);
34
35
36 ResultSet rset = pstmt.executeQuery();
37
38 rset.next();
39
40 String c1no = rset.getString("c1no");
41 String c1name = rset.getString("c1name");
42 String c1etc = rset.getString("c1etc");
43 String c1date = rset.getString("c1date");
44
45
46 %>
```

- 한 가지를 상세조회하는 것이기 때문에, while문이 불필요함
- rset에 담긴 했지만, 해당 데이터가 존재하는지 아닌지 확인하는 과정이 필요함

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

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

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title> Read1 </title>
</head>
<body>

<%
    Class.forName("com.mysql.jdbc.Driver");

    Connection conn = null;

    String hostname = "jdbc:mysql://localhost:3306/class";
    String username = "root";
    String password = "0000";
```

```

        conn = DriverManager.getConnection(hostname, username, password);

    %>

    <%

        String sql = "SELECT * FROM class1 WHERE c1no = ?";

        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, 2);

        ResultSet rset = pstmt.executeQuery();

        rset.next();

        String c1no = rset.getString("c1no");
        String c1name = rset.getString("c1name");
        String c1etc = rset.getString("c1etc");
        String c1date = rset.getString("c1date");

    %>

    <%= c1no %> <%= c1name %> <%= c1etc %> <%= c1date %> <br>

</body>
</html>

```

## C R **U** D 중 UPDATE

```

    <%

        String sql = "UPDATE class1 SET cname = ?, cetc = ? WHERE cno = ?";

        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, "김민수");
        pstmt.setString(2, "tset5");
        pstmt.setInt(3, 5);

        pstmt.executeUpdate();

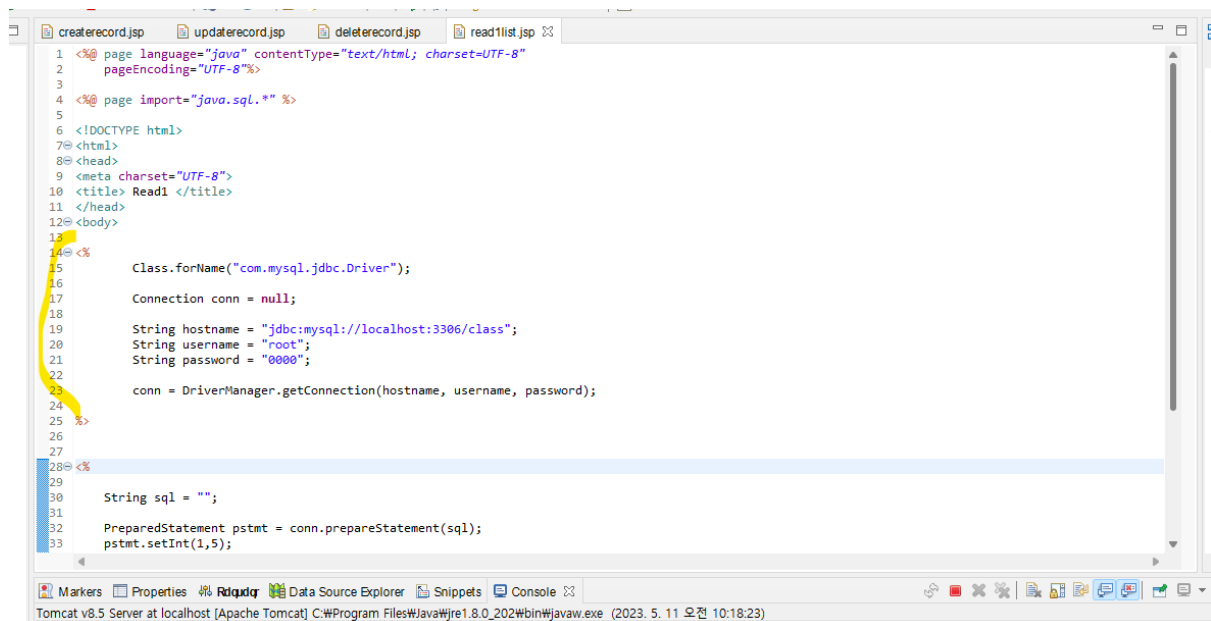
    %>

```

## C R U D 중 DELETE

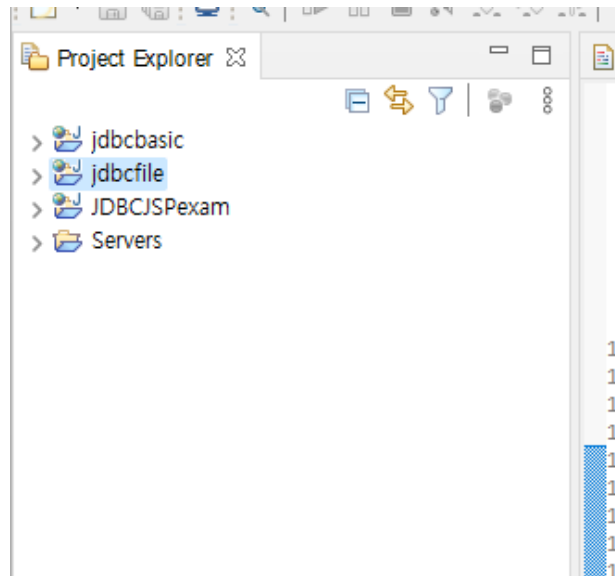
```
<%  
  
String sql = "DELETE FROM class1 WHERE c1no=?";  
  
PreparedStatement pstmt = conn.prepareStatement(sql);  
pstmt.setInt(1,5);  
  
pstmt.executeUpdate();  
  
pstmt.close();  
conn.close();  
  
%>
```

## 커넥트 부분 독립시키기



CRUD 할 때마다 중복해서 작성해야 했던 부분들을 개별파일로 독립시키자!

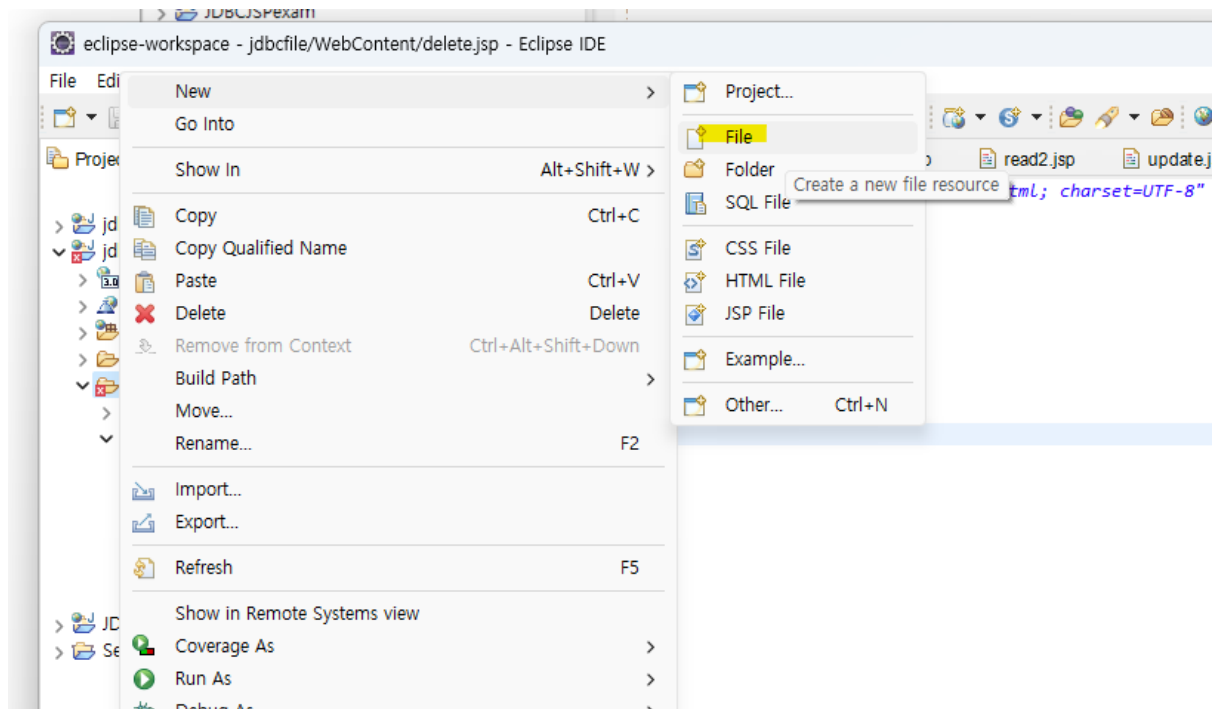
## JDBC - File



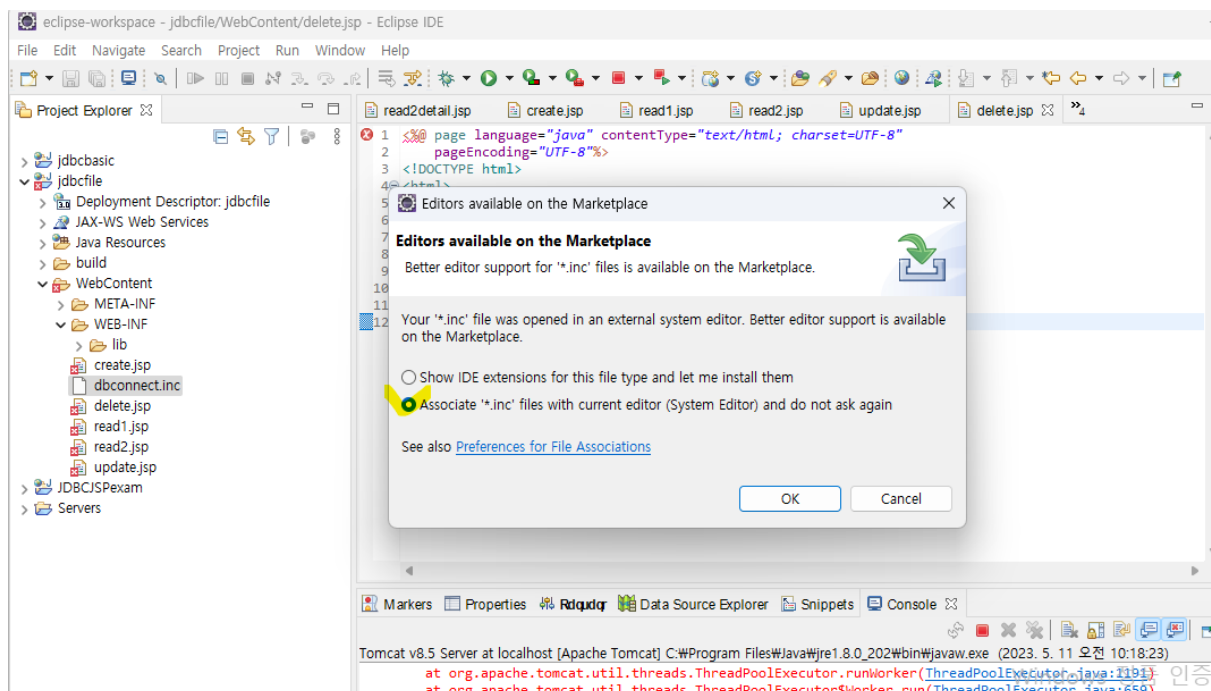
## 디비연결을 집어넣는 용도로 사용할 부분



독립적인 file로 만들어서 갖다 쓸 것

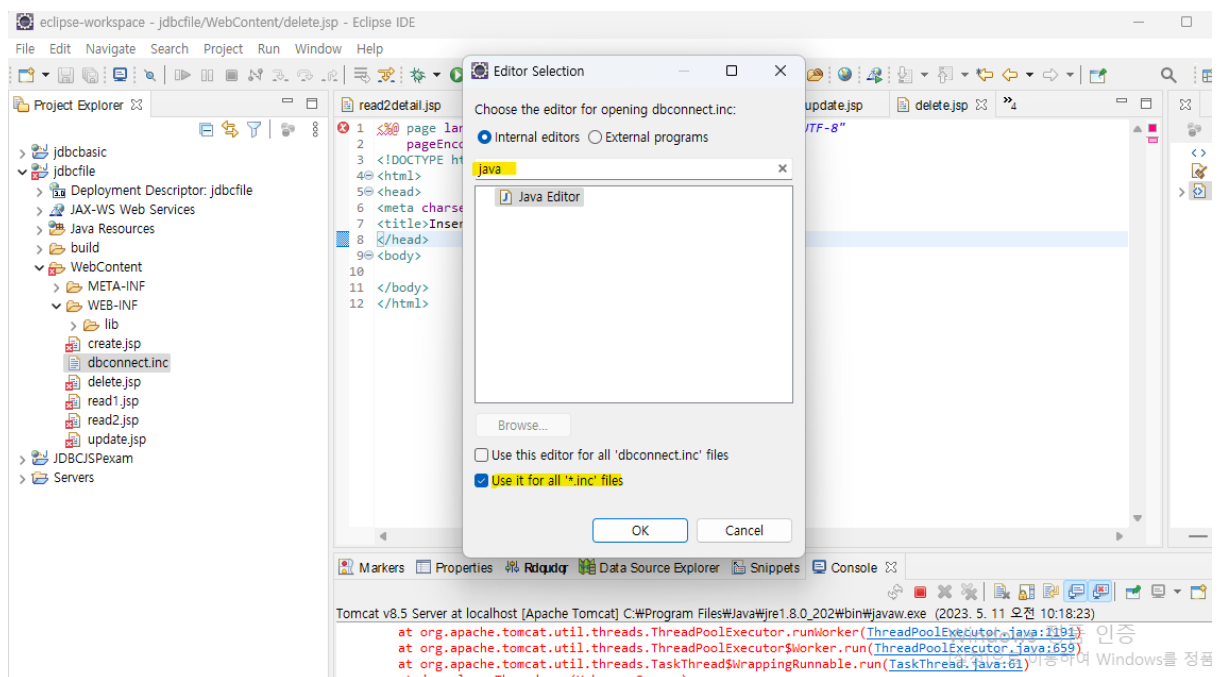
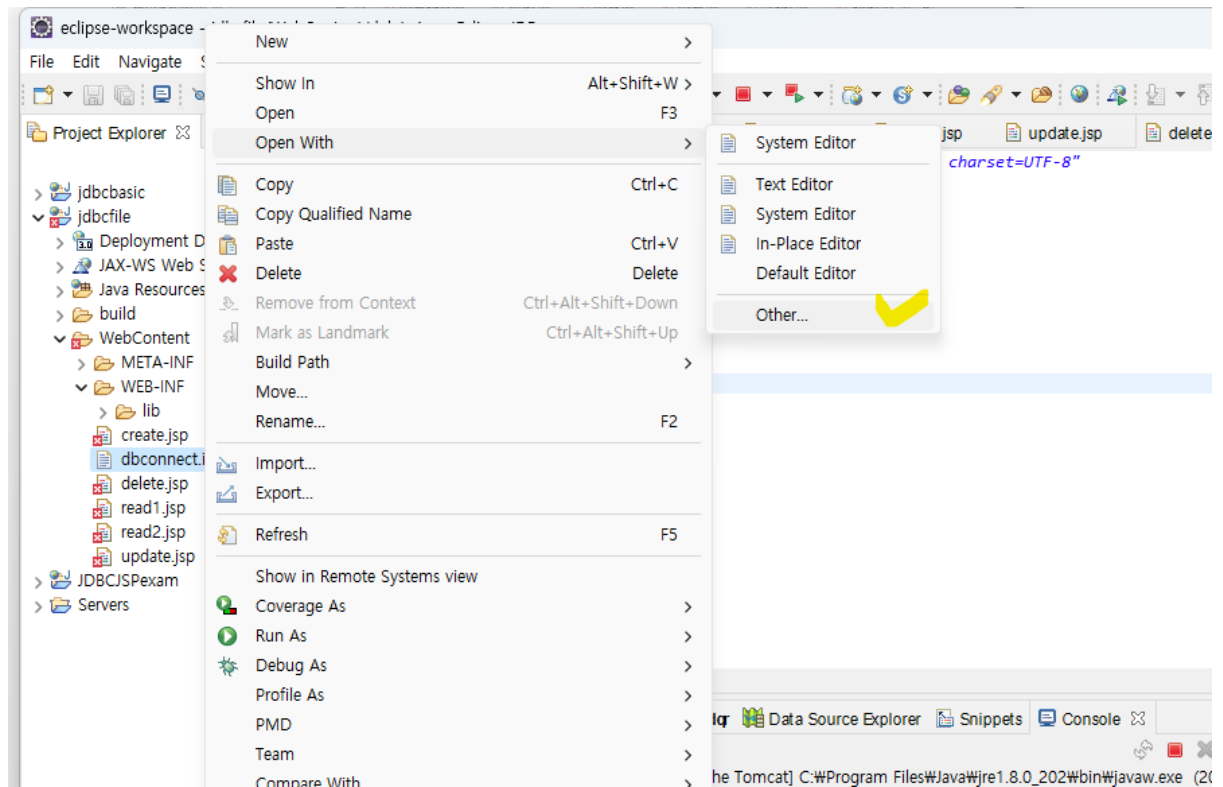


[ File ] - [ New ] - [ File ] 로 생성 !!!!



냅다 File 선택하고, inc라는 확장자명 사용하니까

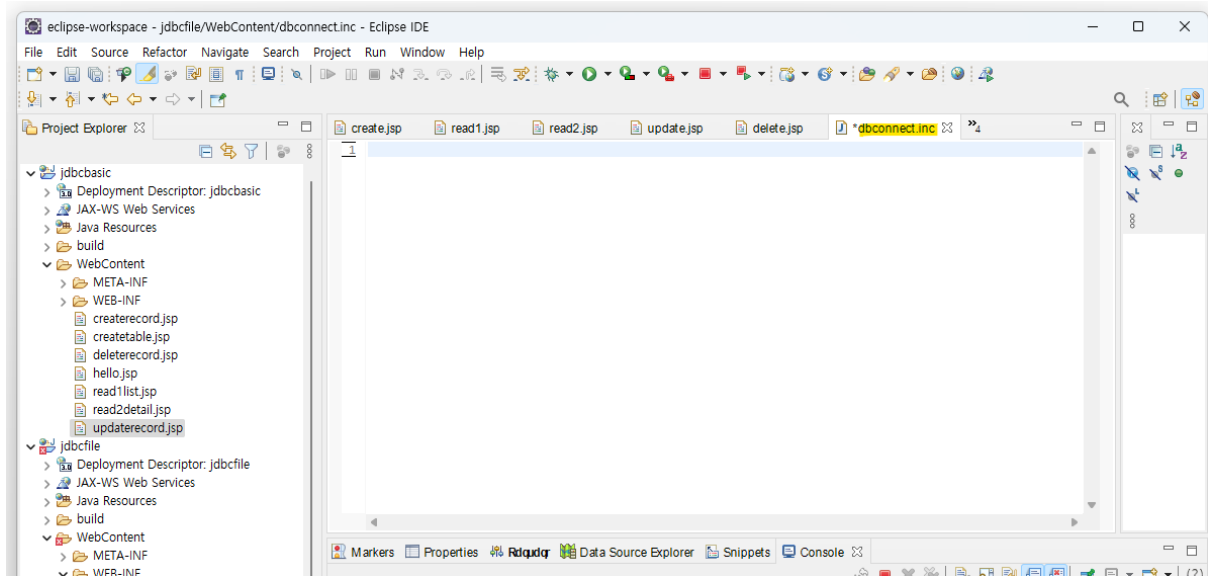
어떤 프로그램으로 어떻게 연결해야 할지 모르겠대





우리가 자바 쓰고 있으니까 [ Java ] - [ Use it for all inc files ]

이렇게 선택하고 나면,



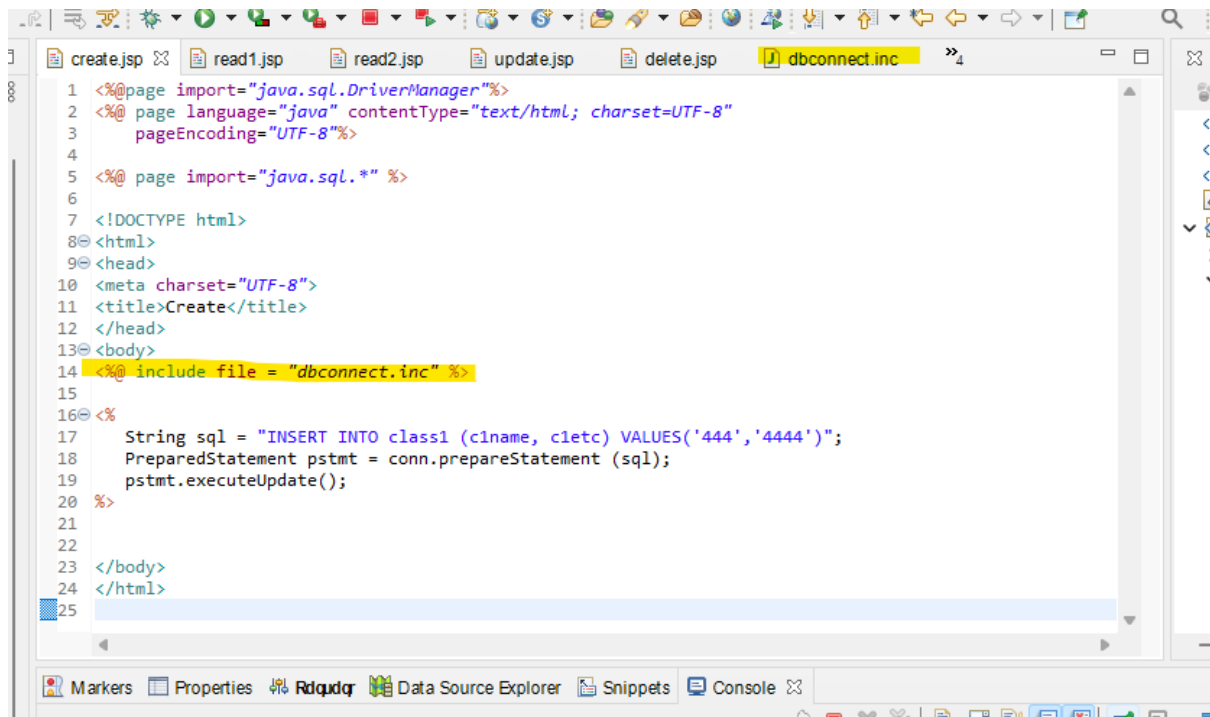
자바 내에서 사용할 수 있게 됨

### 이 코드를 dbconnect 파일에 넣고 쓰면 되는 것

```
<%  
    Class.forName("com.mysql.jdbc.Driver");  
  
    Connection conn = null;  
  
    String hostname = "jdbc:mysql://localhost:3306/class";  
  
    String username = "root";  
  
    String password = "0000";  
  
    conn = DriverManager.getConnection(hostname, username, password);  
  
%>
```

- include 파일을 고대로 첨부할 것이기 때문에 스크립틀릿도 함께 작성

## C R U D 중 CREATE ( 개별파일 활용 )



```
1 <%@page import="java.sql.DriverManager"%>
2 <%@ page language="java" contentType="text/html; charset=UTF-8"
3   pageEncoding="UTF-8"%>
4
5 <%@ page import="java.sql.*" %>
6
7 <!DOCTYPE html>
8 <html>
9 <head>
10 <meta charset="UTF-8">
11 <title>Create</title>
12 </head>
13 <body>
14 <%@ include file = "dbconnect.inc" %>
15
16 <%
17   String sql = "INSERT INTO class1 (cname, c1etc) VALUES('444','4444')";
18   PreparedStatement pstmt = conn.prepareStatement (sql);
19   pstmt.executeUpdate();
20 %>
21
22
23 </body>
24 </html>
25
```

커넥트 부분을 담고 있는 개별파일 import

```
<%@page import="java.sql.DriverManager"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
  pageEncoding="UTF-8"%>

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

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Create</title>
</head>
<body>
<%@ include file = "dbconnect.inc" %>
```

```
<%  
    String sql = "INSERT INTO class1 (c1name, c1etc) VALUES('444','4444')";  
    PreparedStatement pstmt = conn.prepareStatement (sql);  
    pstmt.executeUpdate();  
%>  
  
</body>  
</html>
```

## C R U D 중 READ ( 개별파일 활용 )

## C R U D 중 UPDATE ( 개별파일 활용 )

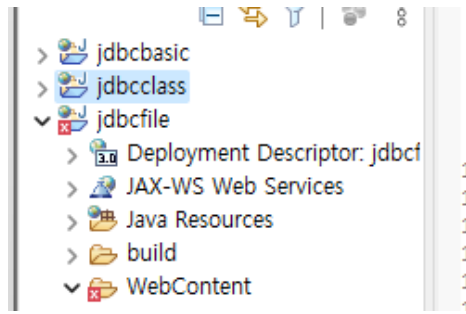
## C R U D 중 DELETE ( 개별파일 활용 )

→ 이 부분 강사님 자료 보고 채우기!

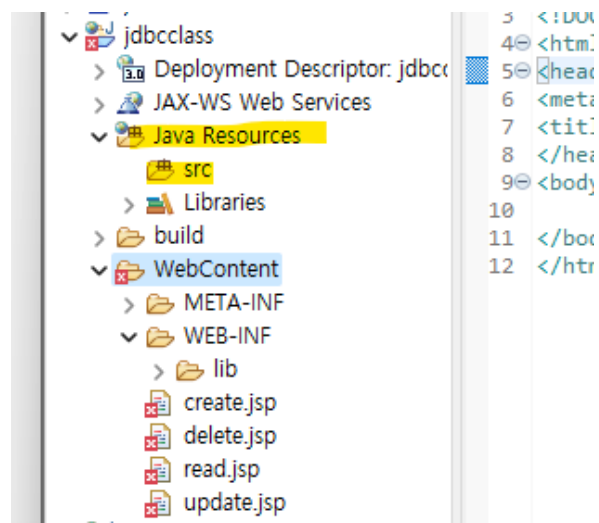
내 거 왜 안돌아가지? update부터 에러 남

## JDBC - Class로 분리

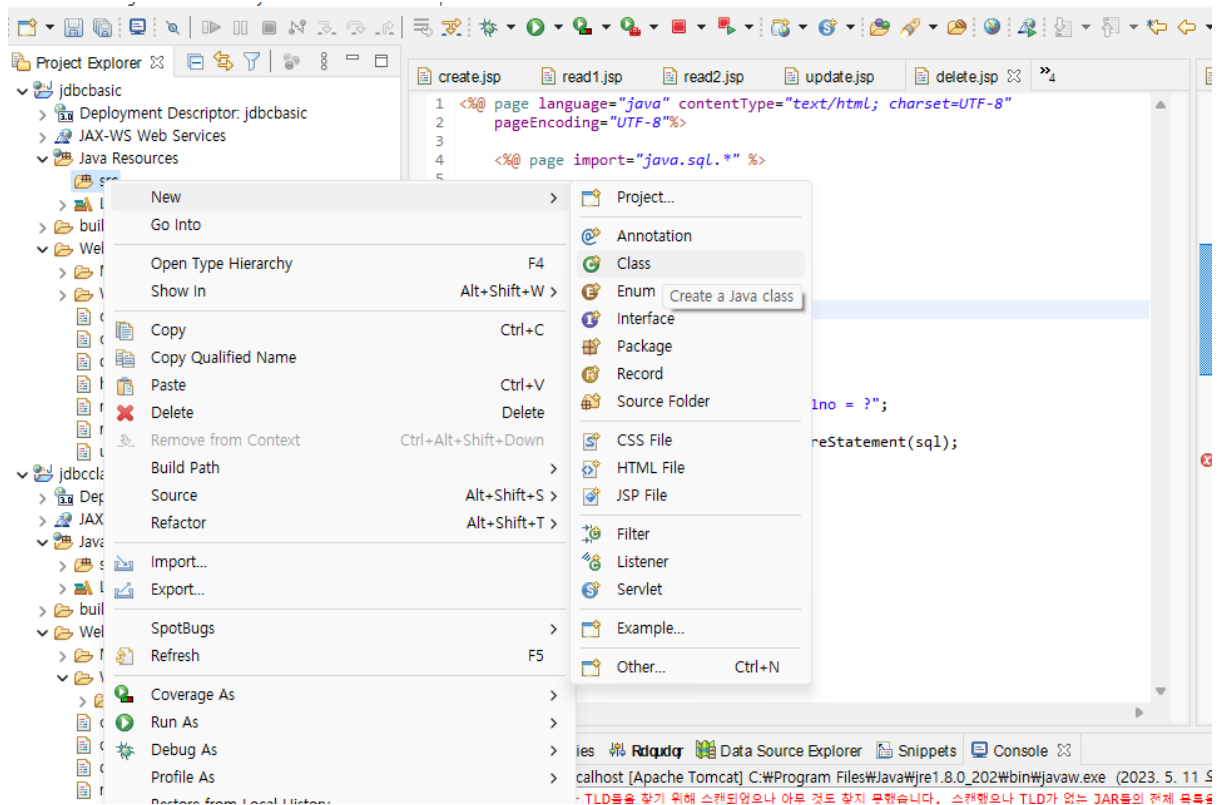
커넥션을 클래스로 분리시키기 ⇒ 자바스럽게 분리시키는 것

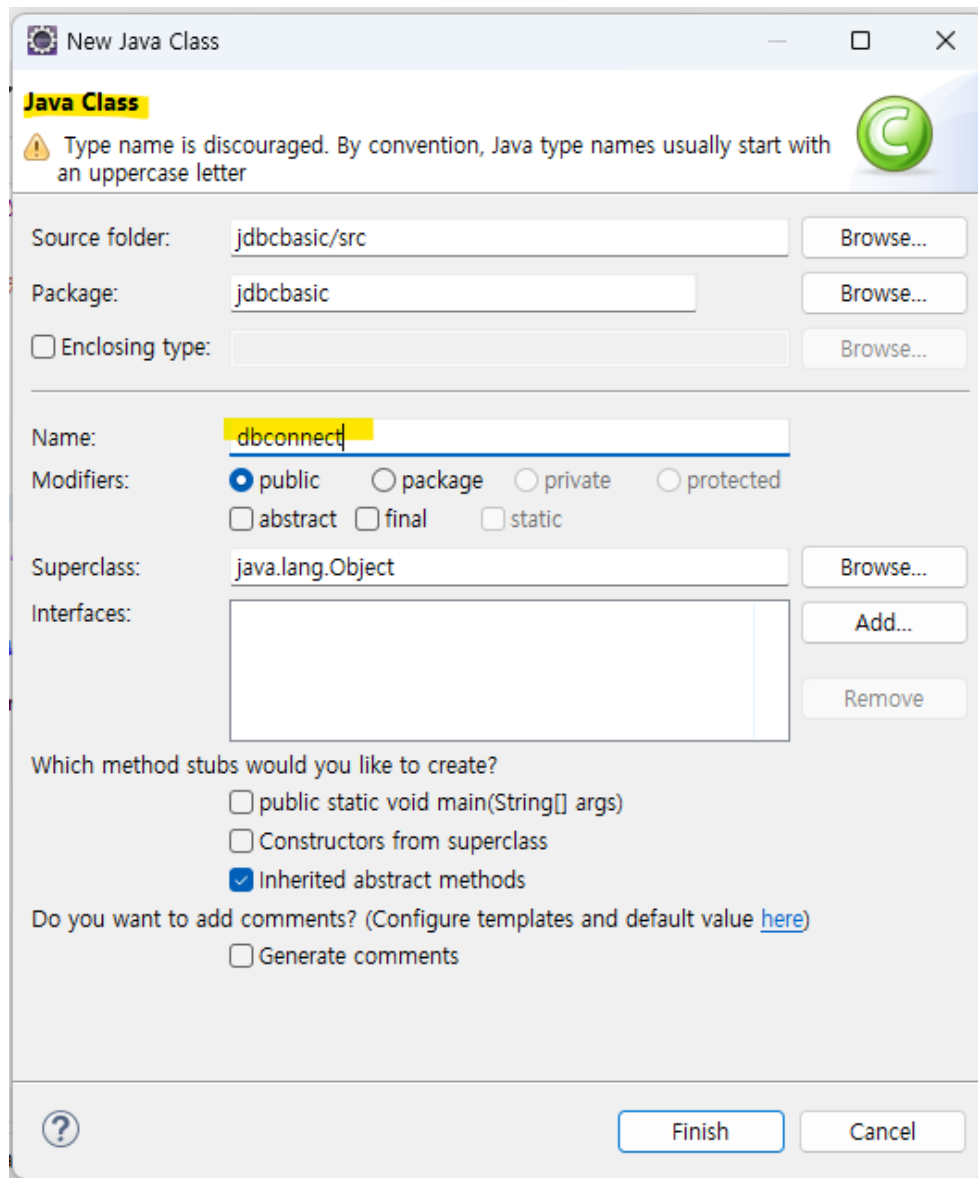


커넥션 부분을 include 파일로 표현했던 것을, class에 담아보자 !

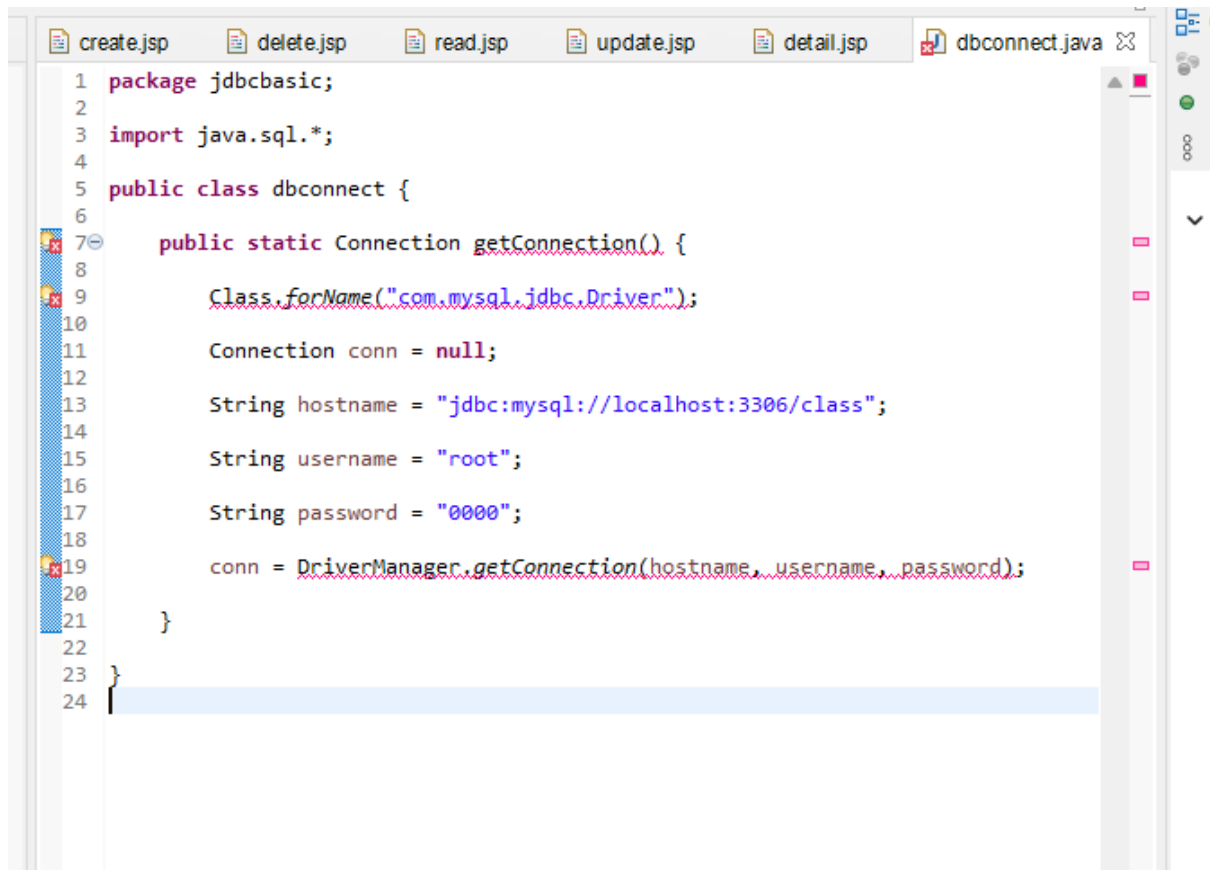


자바 파일의 위치는 [ Java Resources ] - [ src ] !





[ dbconnect ] 클래스 생성

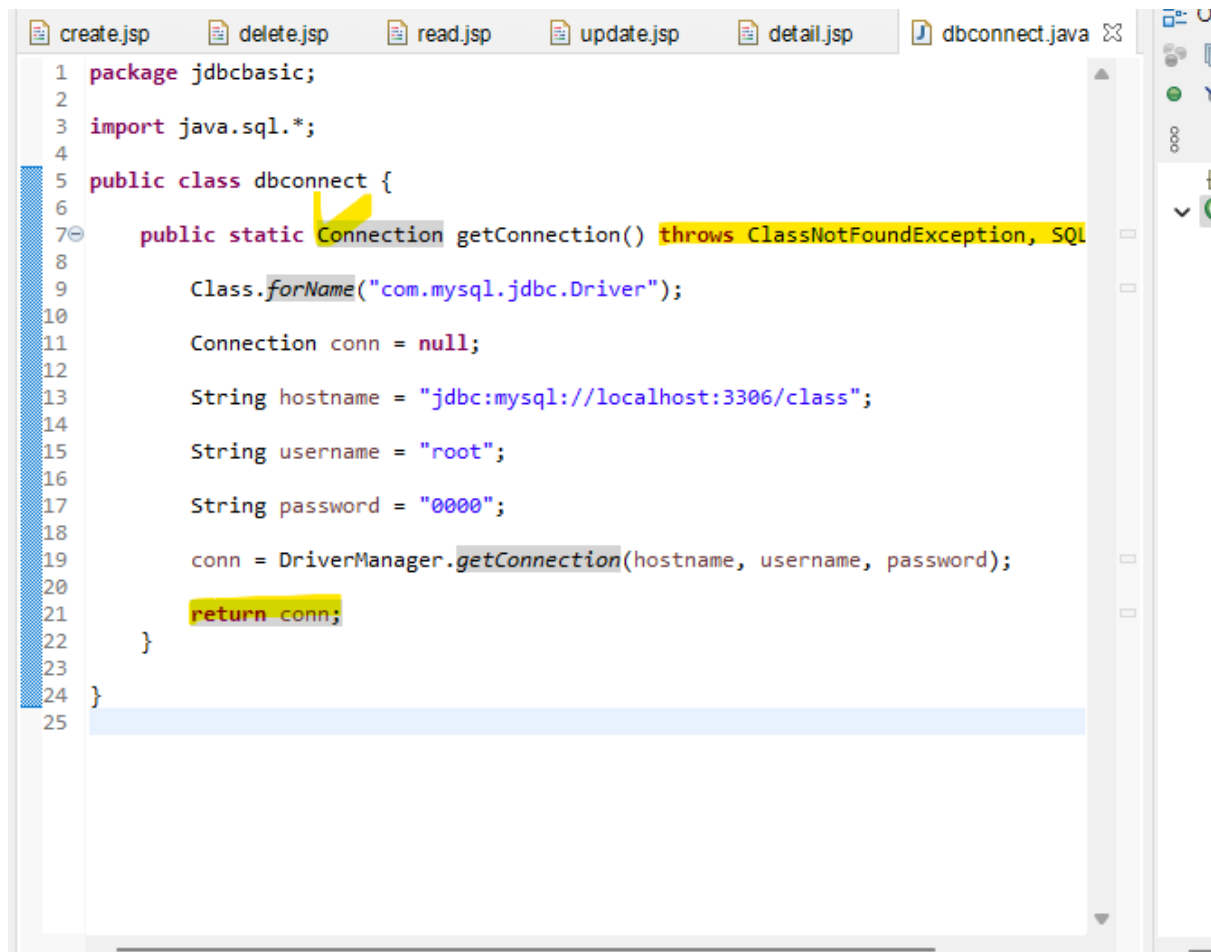


```
1 package jdbcbasic;
2
3 import java.sql.*;
4
5 public class dbconnect {
6
7     public static Connection getConnection() {
8
9         Class.forName("com.mysql.jdbc.Driver");
10
11         Connection conn = null;
12
13         String hostname = "jdbc:mysql://localhost:3306/class";
14
15         String username = "root";
16
17         String password = "0000";
18
19         conn = DriverManager.getConnection(hostname, username, password);
20
21     }
22
23 }
24 }
```

**static** 으로 메서드 생성한 이유 ⇒ 호출할 때 객체 생성 없이 사용하기 위해서

에러 발생

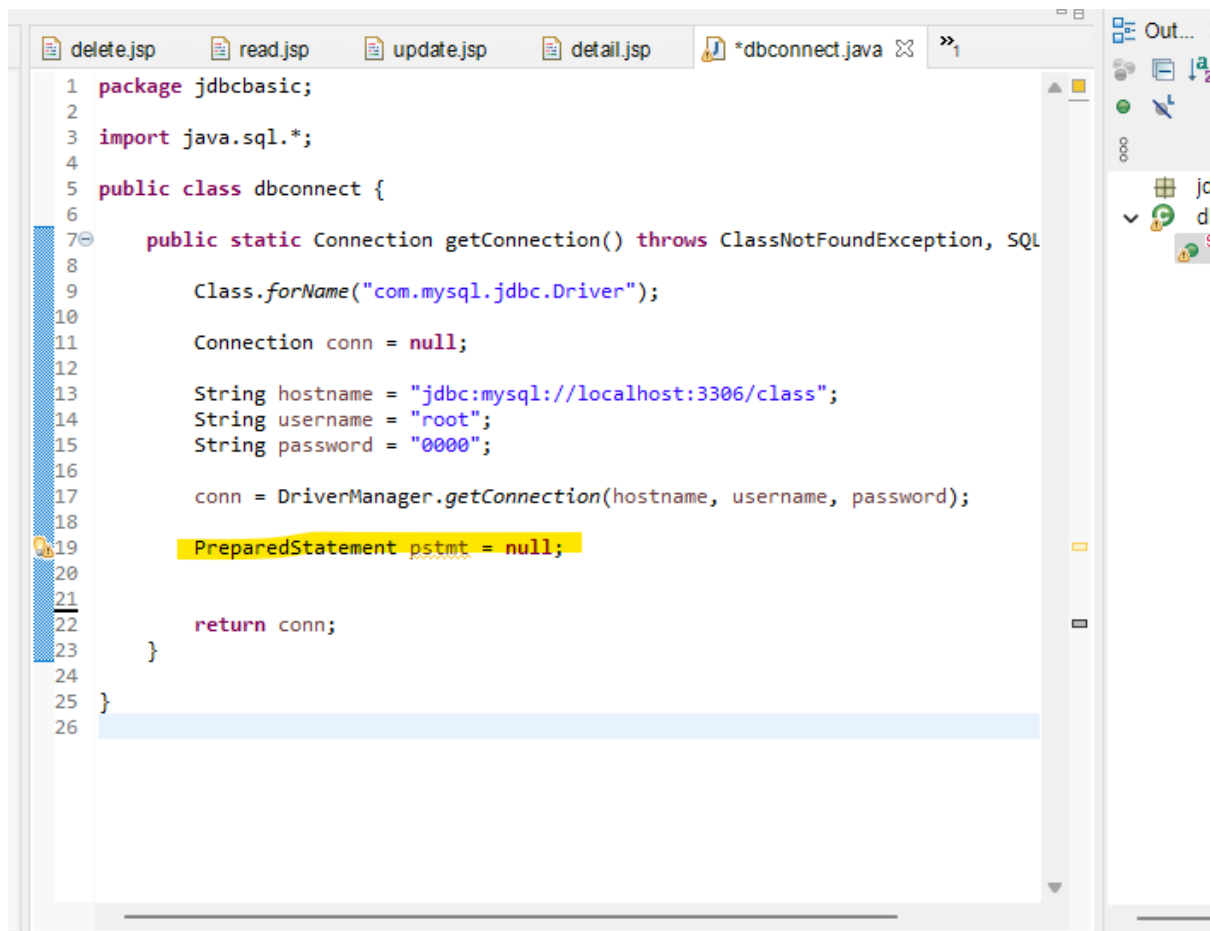
⇒ 마우스 올리고 add to throws 눌러서 시그너처 부분에 throws Exception 추가



```
1 package jdbcbasic;
2
3 import java.sql.*;
4
5 public class dbconnect {
6
7     public static Connection getConnection() throws ClassNotFoundException, SQLException {
8
9         Class.forName("com.mysql.jdbc.Driver");
10
11         Connection conn = null;
12
13         String hostname = "jdbc:mysql://localhost:3306/class";
14
15         String username = "root";
16
17         String password = "0000";
18
19         conn = DriverManager.getConnection(hostname, username, password);
20
21         return conn;
22     }
23 }
24
25
```

반환형이 Connection이기 때문에  
return conn; 작성필요





19번 코드 추가 !

pstmt를 사용하는 곳은 따로 있지만, 여기서 미리 초기화해두면 사용 시 편리함

## 커넥션 담긴 클래스 전체코드

```
package jdbcbasic;

import java.sql.*;

public class dbconnect {

    public static Connection getConnection() throws ClassNotFoundException, SQLException {

        Class.forName("com.mysql.jdbc.Driver");

        Connection conn = null;

        String hostname = "jdbc:mysql://localhost:3306/class";
        String username = "root";
        String password = "0000";

        conn = DriverManager.getConnection(hostname, username, password);
```

```

        PreparedStatement pstmt = null;

        return conn;
    }
}

```

## C R U D 중 CREATE ( 클래스 활용 )

```

<%@page import="java.sql.DriverManager"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>

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

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Create</title>
</head>
<body>

<%
    Connection conn = null;
    conn = dbconnect.getConnection();

    String sql = "INSERT INTO class1 (c1name, c1etc) VALUES ('cup', 'drink')";
    PreparedStatement pstmt = conn.prepareStatement(sql);
    pstmt.executeUpdate();

%>

</body>
</html>

```

## C R U D 중 READ ( 클래스 활용 )

```

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

```

```

<%@ page import="java.sql.*" %>
<%@ page import="jdbc.class.*" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title> Read </title>
</head>
<body>

<%

    Connection conn = null;
    conn = dbconnect.getConnection();

    String sql = "SELECT * FROM class1";
    PreparedStatement pstmt = conn.prepareStatement(sql);
    ResultSet rset = pstmt.executeQuery();

    while(rset.next()){
        String c1no = rset.getString("c1no");
        String c1name = rset.getString("c1name");
        String c1etc = rset.getString("c1etc");
        String c1date = rset.getString("c1date");

%>

<%=c1no %> <%=c1name %> <%=c1etc %> <%=c1date %> <br>

<%
    }
%>

</body>
</html>

```

## C R U D 중 DETAIL ( 클래스 활용 )

디테일 부분에서 !

```
21 String hostname = "jdbc:mysql://localhost:3306/class";
22 String username = "root";
23 String password = "0000";
24
25 conn = DriverManager.getConnection(hostname, username, password);
26
27 %>
28
29
30 <%
31 int no = 6;
32 String sql = "SELECT * FROM class1 WHERE c1no = ?";
33
34 PreparedStatement pstmt = conn.prepareStatement(sql);
35 pstmt.setInt(1, no);
36
37
38 ResultSet rset = pstmt.executeQuery();
39
40 rset.next();
41
42 String c1no = rset.getString("c1no");
43 String c1name = rset.getString("c1name");
44 String c1etc = rset.getString("c1etc");
45 String c1date = rset.getString("c1date");
46 %>
47
48 <%=c1no %> <%=c1name %> <%=c1etc %> <%=c1date %> <br>
49
50
51
52 </body>
```

상세보기 할 게시물번호를 변수처리해줌

근데! 저렇게 변수 처리를 하면 안되고,

[ 클릭 ] 을 하면 그 번호를 넘겨받아야 함 . → request 사용

```

1  <%
2  String no = request.getParameter("c1no");
3  String sql = "SELECT * FROM class1 WHERE c1no = ?";
4
5  PreparedStatement pstmt = conn.prepareStatement(sql);
6  pstmt.setString(1, no);
7
8  ResultSet rset = pstmt.executeQuery();
9
10 rset.next();
11
12     String c1no = rset.getString("c1no");
13     String c1name = rset.getString("c1name");
14     String c1etc = rset.getString("c1etc");
15     String c1date = rset.getString("c1date");
16 %>
17
18 <%=c1no %> <%=c1name %> <%=c1etc %> <%=c1date %> <br>
19

```

<< 포인트가 되는 부분들 >>

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3
4 <%@ page import="java.sql.*" %>
5 <%@ page import="jdbc.class.*" %>
6
7 <!DOCTYPE html>
8 <html>
9 <head>
10 <meta charset="UTF-8">
11 <title>Update title here</title>
12 </head>
13 <body>
14
15 <%
16
17     Connection conn = null;
18     conn = dbconnect.getConnection();
19
20
21     int no = 6;
22     String sql = "SELECT * FROM class1 WHERE c1no = ?";
23
24     PreparedStatement pstmt = conn.prepareStatement(sql);
25     pstmt.setInt(1, no);
26
27
28     ResultSet rset = pstmt.executeQuery();
29
30     rset.next();
31
32     String c1no = rset.getString("c1no");
33     String c1name = rset.getString("c1name");
34     String c1etc = rset.getString("c1etc");
35     String c1date = rset.getString("c1date");
36 %>
37
38     <%=c1no %> <%=c1name %> <%=c1etc %> <%=c1date %> <br>
39
40
41 </body>
42 </html>
```

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

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

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Update title here</title>
</head>
<body>
```

```

<%

    Connection conn = null;
    conn = dbconnect.getConnection();

    int no = Integer.parseInt(request.getParameter("no"));
    String sql = "SELECT * FROM class1 WHERE c1no = ?";

    PreparedStatement pstmt = conn.prepareStatement(sql);
    pstmt.setInt(1, no);

    ResultSet rset = pstmt.executeQuery();

    rset.next();

    String c1no = rset.getString("c1no");
    String c1name = rset.getString("c1name");
    String c1etc = rset.getString("c1etc");
    String c1date = rset.getString("c1date");
%>

    <%=c1no %> <%=c1name %> <%=c1etc %> <%=c1date %> <br>

</body>
</html>

```

## C R U D 중 UPDATE ( 클래스 활용 )

```

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

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

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Update title here</title>
</head>
<body>

<%
    Connection conn = null;
    conn = dbconnect.getConnection();

    Class.forName("com.mysql.jdbc.Driver");

```

```

String hostname = "jdbc:mysql://localhost:3306/class";
String username = "root";
String password = "0000";

conn = DriverManager.getConnection(hostname, username, password);

%>

<%
String sql = "UPDATE class1 SET c1name=? WHERE c1no = ?";

PreparedStatement pstmt = conn.prepareStatement(sql);
pstmt.setString(1, "cup");
pstmt.setInt(2, 6);

pstmt.executeUpdate();
%>

</body>
</html>

```

## CRUD 중 DELETE ( 클래스 활용 )

```

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

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

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>delete title here</title>
</head>
<body>

<%

Connection conn = null;
conn = dbconnect.getConnection();

String sql = "DELETE FROM class1 WHERE c1no = ?";

PreparedStatement pstmt = conn.prepareStatement(sql);
pstmt.setInt(1, 8);

pstmt.executeUpdate();

```



```
%>

</body>
</html>
```

## 값을 입력받을 수 있는 input 화면 만들기

일단 기본 틀이 되는 화면을 만들어볼게

### input.jsp 코드

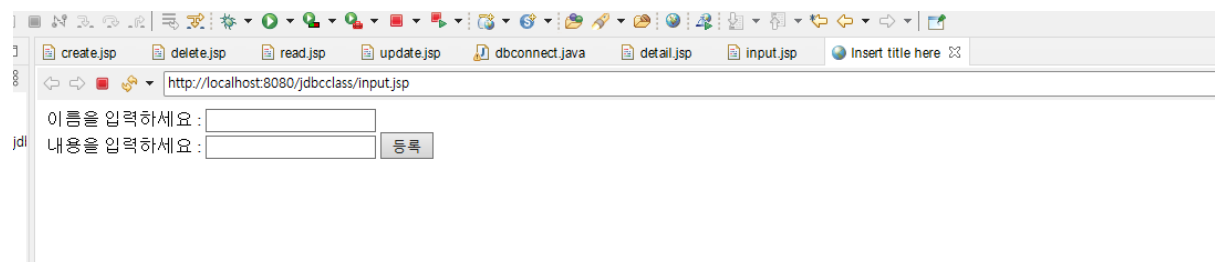
```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

<form action="create.jsp">

이름을 입력하세요 : <input type="text" name="name">
<br>
내용을 입력하세요 : <input type="text" name="etc">
<input type="submit" value="등록">

</form>

</body>
</html>
```



input.jsp에서 입력받은 값을

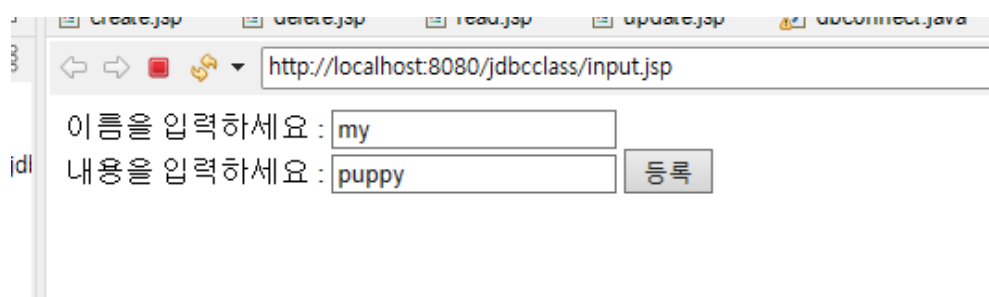
이제 create.jsp 로 보내야 하기 때문에

```
10
11 <form action="create.jsp">
12 이름을 입력하세요 : <input type="text" name="name">
13 <br>
14 내용을 입력하세요 : <input type="text" name="etc">
15
16 <input type="submit" value="등록">
17 </form>
18
19 </body>
20 </html>
```

<form> </form> 으로 묶은 후,

action = " " 을 이용해서 입력받은 값을 create.jsp로 보내겠다고 설정

<before>

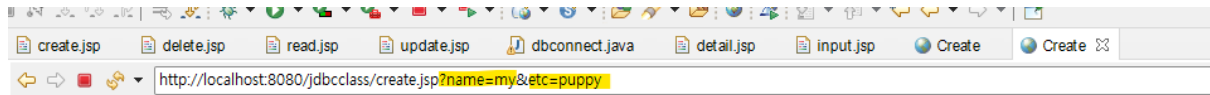


이름을 입력하세요 : my

내용을 입력하세요 : puppy 등록

input.jsp 에서 값을 입력하고 [ 등록 ] 버튼을 눌렀더니

<after>



url 창에, **create.jsp**로 이동하고 **입력한 값들이** 들어가있음



**input.jsp에서 보낸 값을 create.jsp 에서 받아서 DB까지 전달되도록 하자**  
**그 다음, read.jsp 가 실행되게 하기**

```
*create.jsp delete.jsp read.jsp update.jsp dbconnect.java detail.jsp input.jsp Create
1 <%@page import="java.sql.DriverManager"%>
2 <%@page language="java" contentType="text/html; charset=UTF-8"
3   pageEncoding="UTF-8"%>
4
5 <%@page import="java.sql.*"%>
6 <%@page import="jdbc.class.*"%>
7
8 <!DOCTYPE html>
9 <html>
10 <head>
11 <meta charset="UTF-8">
12 <title>Create</title>
13 </head>
14 <body>
15 <%
16     Class.forName("com.mysql.jdbc.Driver");
17
18     Connection conn = null;
19
20     String hostname = "jdbc:mysql://localhost:3306/class";
21     String username = "root";
22     String password = "0000";
23
24     conn = DriverManager.getConnection(hostname, username, password);
25
26     String name = request.getParameter("name");
27     String etc = request.getParameter("etc");
28 %>
29
30 <%
31
32     String sql = "INSERT INTO class1 (cname, c1etc) VALUES (?, ?)";
33     PreparedStatement pstmt = conn.prepareStatement(sql);
34
35     pstmt.setString(1, name);
36     pstmt.setString(2, etc);
37
38     pstmt.executeUpdate();
39
40 %>
41 </body>
42 </html>
```

```
pstmt.executeUpdate();

response.sendRedirect("read.jsp");

%>
```

26번, 27번과 같이

name, ect 변수를 만들어서, input.jsp에서 request객체에 넣었던 name과 etc를 담기

그 다음에 **response.sendRedirect("read.jsp");** 한 줄 추가하면

read.jsp 페이지가 실행됨

## create.jsp 코드

```
<%@page import="java.sql.DriverManager"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>

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

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Create</title>
</head>
<body>
<%
    Class.forName("com.mysql.jdbc.Driver");

    Connection conn = null;

    String hostname = "jdbc:mysql://localhost:3306/class";
    String username = "root";
    String password = "0000";

    conn = DriverManager.getConnection(hostname, username, password);

    String name = request.getParameter("name");
    String etc = request.getParameter("etc");
%>

<%

    String sql = "INSERT INTO class1 (c1name, c1etc) VALUES (? , ?)";
    PreparedStatement pstmt = conn.prepareStatement(sql);

    pstmt.setString(1, name);
    pstmt.setString(2, etc);

    pstmt.executeUpdate();

    response.sendRedirect("read.jsp");

%>
</body>
</html>
```



**read.jsp에서 특정 번호를 누르면**

**detail.jsp로 넘어가서 특정 번호의 상세조회가 가능하도록 하자**

```

8  %>
9
0  <%=c1no %> <a href="detail.jsp?no=<%=c1no %>"><%=c1name %></a> <%=c1etc %> <%=c1date %> <br>
1
2  <%
3      }

```

## read.jsp코드

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import="java.sql.*" %>
<%@ page import="jdbc.class.*" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title> Read </title>
</head>
<body>

<%

    Connection conn = null;
    conn = dbconnect.getConnection();

    String sql = "SELECT * FROM class1";
    PreparedStatement pstmt = conn.prepareStatement(sql);
    ResultSet rset = pstmt.executeQuery();

    while(rset.next()){
        String c1no = rset.getString("c1no");
        String c1name = rset.getString("c1name");
        String c1etc = rset.getString("c1etc");
        String c1date = rset.getString("c1date");

    %>

    <%=c1no %> <a href="detail.jsp?no=<%=c1no %>"><%=c1name %></a> <%=c1etc %> <%=c1date %> <br>

    <%
    }
    %>

</body>
</html>

```

그럼 이제 detail.jsp에서 받아줘야 함

```

19
20
21 int no = Integer.parseInt(request.getParameter("no"));
22 String sql = "SELECT * FROM class1 WHERE c1no = ?";
23
24 PreparedStatement pstmt = conn.prepareStatement(sql);
25 pstmt.setInt(1, no);
26
27

```

detail.jsp에서는 몇 번 게시물의 상세조회를 할 것인지가 중요함

그래서, 그 번호를 request 객체에서 뽑아와야 함

그런데

getParameter("no")로 뽑은값은 String 형태이기 때문에, int형으로 형변환해주고

그렇게 받은 no를 상세조회하는 것

## detail.jsp 코드

```

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

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

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Update title here</title>
</head>
<body>

<%

    Connection conn = null;
    conn = dbconnect.getConnection();

    int no = Integer.parseInt(request.getParameter("no"));
    String sql = "SELECT * FROM class1 WHERE c1no = ?";

    PreparedStatement pstmt = conn.prepareStatement(sql);
    pstmt.setInt(1, no);

```

```
ResultSet rset = pstmt.executeQuery();

rset.next();

String c1no = rset.getString("c1no");
String c1name = rset.getString("c1name");
String c1etc = rset.getString("c1etc");
String c1date = rset.getString("c1date");
%>

<%=c1no %> <%=c1name %> <%=c1etc %> <%=c1date %> <br>

</body>
</html>
```