

Class 2023.01.13

My SQL

데이터베이스에 관해 이것저것

마리아DB

- My SQL과 완전히 호환
- My SQL이 오라클에 인수된 이후에 하는 짓이 맘에 안들어서
- 비슷한 마리아 DB라는 애를 만들어서 무료로 풀음.
- 무료 DB를 찾을 때는 현업에서는 MariaDB를 많이 씀.

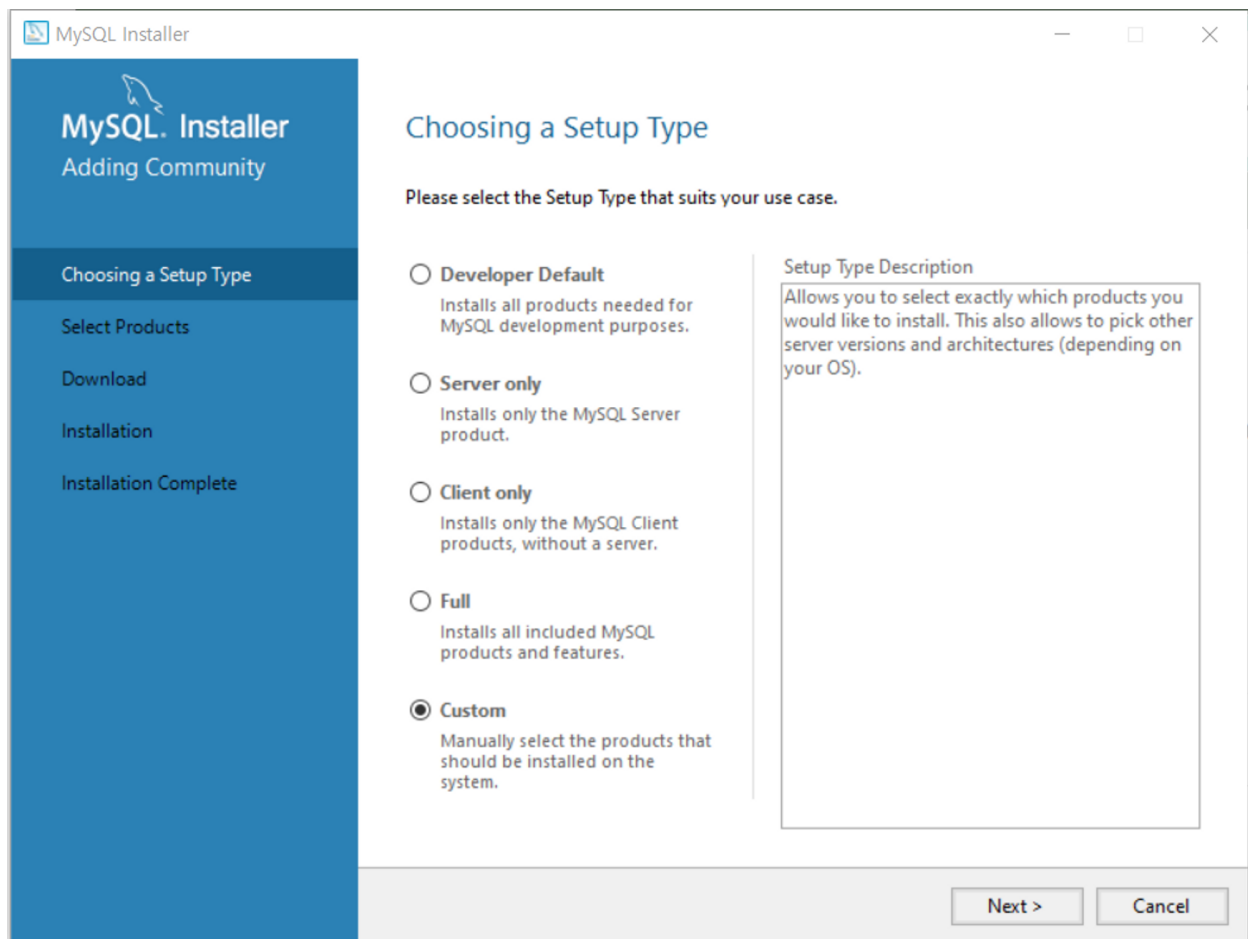
Heidi

- MySQL 워크벤치로써 현재는 헤이디가 최고

My SQL 삭제하기

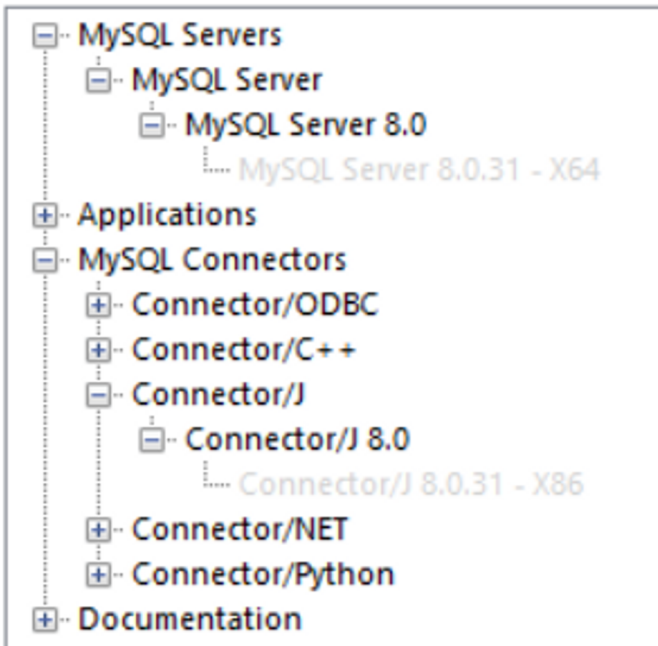
- 내 컴에 원래 MySQL이 깔려있었기 때문에..
1. 시작 → services → MySQL → properties → 사용 안함 체크
 2. 시작 → registry editor → MySQL로 써있는거 싹 삭제
 3. C:\Program Data 에서 MySQL폴더 제거(Program File 아님 주의)
 4. 제어판에서 MySQL로 시작하는거 싹 제거

My SQL 초기세팅

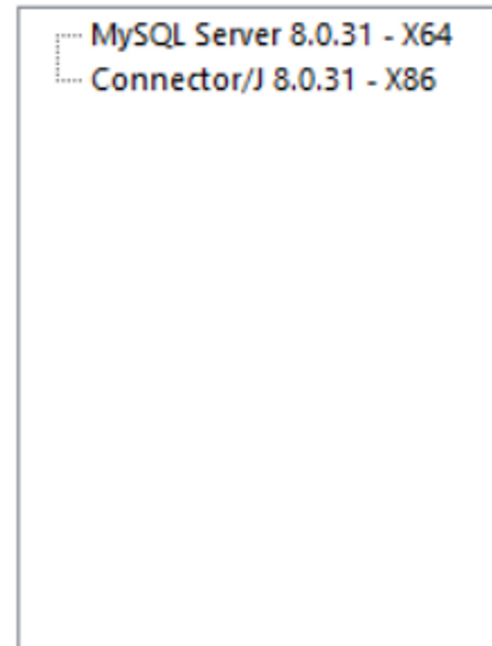


- 필요한 것만 설치하도록 custom

Available Products:



Products To Be Installed:



☐ Enable the Select Features pa
customize product features

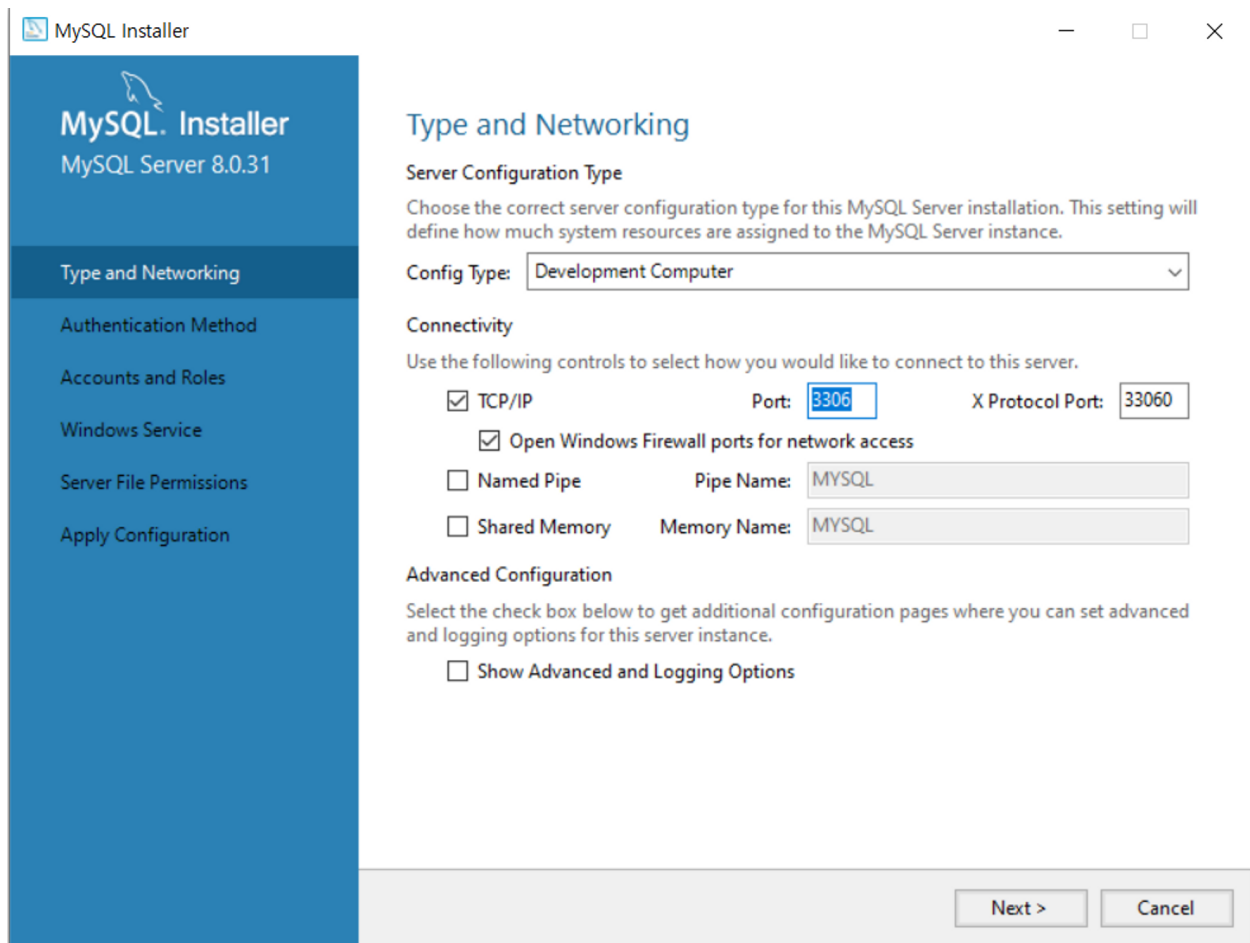
Published: 2022년 10월 11일 화요일

Release Notes: <https://dev.mysql.com/doc/relnotes/connector-j/8.0/en/news-8-0-31.html>

< Back

Next >

- 깔아야할 것 이렇게 두 개



- 포트번호 3306통일
- 맞춰놔야 나중에 어디서 얼마나 바꾸지? 이런 상황 안생김

MySQL Installer

MySQL Server 8.0.31

Type and Networking

Authentication Method

Accounts and Roles

Windows Service

Server File Permissions

Apply Configuration

Accounts and Roles

Root Account Password
Enter the password for the root account. Please remember to store this password in a secure place.

MySQL Root Password:

Repeat Password:

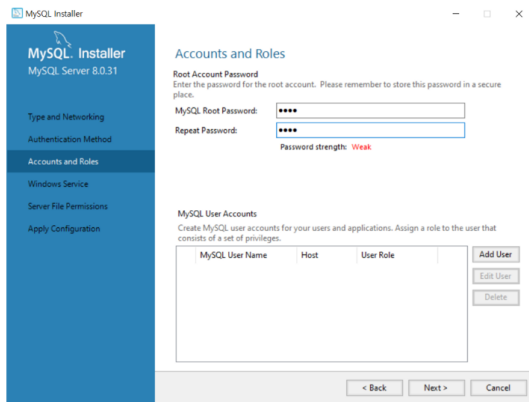
Password strength: **Weak**

MySQL User Accounts
Create MySQL user accounts for your users and applications. Assign a role to the user that consists of a set of privileges.

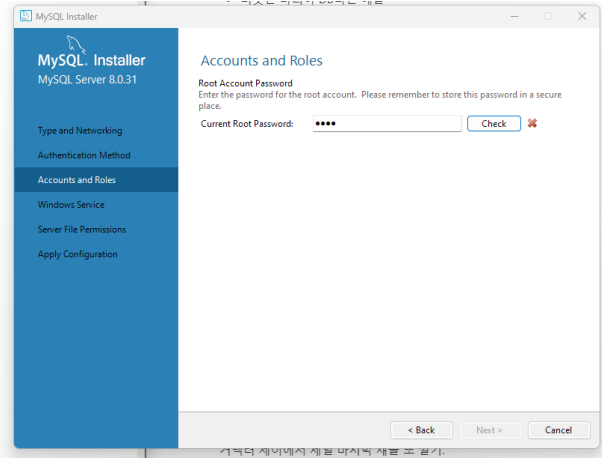
MySQL User Name	Host	User Role
-----------------	------	-----------

- 비밀번호 잘 기억하기

문제 생겼음..



암호는 0000으로 통일.



- 비밀번호 입력창이 제대로 안뜸

이전에 My SQL을 쓴 적이 있어서 데이터가 남아있어서 생겼던 문제로 추정.

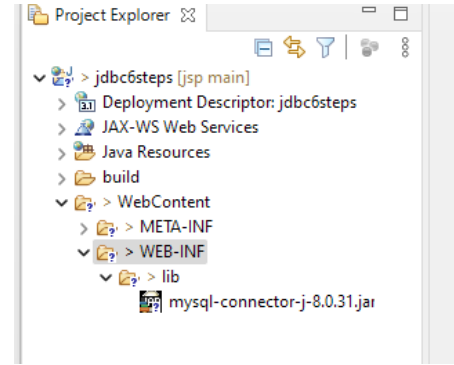
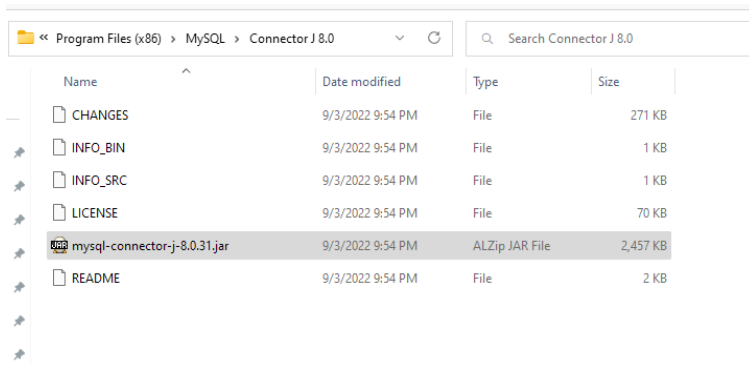
ProgramFile이 아니라 ProgramData에 데이터가 또 남아있었음. 데이터 다 날려서 해결함.

뭔가 노란색이 뜨면 잘 읽어보도록 하자.

[JSP] DataBase Connection

JSP로 데이터베이스를 사용하는 4가지 방법

1. 각각의 JSP파일에서 매번 직접 DB사용
2. DB 연결 부분만 별도의 파일로 구성
3. DB 연결 부분만 별도의 Class로 구성
4. Connection Pool 사용



- Program Files → MySQL → connector에서 jar파일 복사
- WEB-INF → lib 안에 넣기

JDBC basic (1번 방법)

6steps

1. Import SQL Packages

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

- JSP 파일 위에 임포트하기

2. Load JDBC Driver

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

- 이거 한 줄이면 lib에 있는 jar랑 연결
- mariaDB 는 `org.maria.jdbc.Driver`
- 아무튼 다른 DB쓰면서 필요하면 검색

```
<%
try {
```

```

    Class.forName("com.mysql.jdbc.Driver");
} catch(ClassNotFoundException err) {
    out.print("JDBC Driver loading error<br>" + err.getMessage());
}
out.print("JDBC Driver loading success<br>");
%>

```

3. Create Connection

```

Connection conn = null;

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306", "root", "0000");

```

매개변수 : (위치, 아이디, 비밀번호)

```

try {
    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/", "root", "0000");
} catch (SQLException err) {
    out.print("Connection Object error<br>" + err.getMessage());
}

```

4. Create Statement Object (구문객체 만들기)

```

PreparedStatement pstmt = conn.prepareStatement("CREATE DATABASE test");

```

5. Excute SQL Query(쿼리 실행)

```

pstmt.executeUpdate();

```

6. Close connection(연결 끊기)

- 자바9부터 생략가능

```

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

```


전체 모음

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

<!-- step 1 import SQL Packages -->

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

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <%
        //Step 2
        // load JDBC Driver

        try {
            Class.forName("com.mysql.jdbc.Driver");
        } catch (ClassNotFoundException err) {
            out.print("JDBC Driver loading error<br>" + err.getMessage());
        }

        // mariaDB는 org.maria.jdbc.Driver

        //3. create Connection Object

        Connection conn = null;

        try {
            conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/", "root", "0000");
        } catch (SQLException err) {
            out.print("Connection Object error<br>" + err.getMessage());
        }

        //4. create Statement Object

        PreparedStatement pstmt = conn.prepareStatement("CREATE DATABASE test");

        //5. Excute SQL Query(쿼리 실행)
        pstmt.executeUpdate();

        //6. Close connection(연결 끊기)
        pstmt.close();
        conn.close();

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

테이블 만들기

```
conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/univ", "root", "0000");
```

- 테이블을 만들 때는 어떤 데이터베이스를 이용하는지 명시해주어야함!!

INSERT

- (UPDATE, DELETE도 비슷한 구조임)

```
String hakbun = "1111";
String name = "홍길동";
String dept = "컴공";
String addr = "서울";

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

PreparedStatement pstmt = conn.prepareStatement(sql);
pstmt.setString(1, hakbun);
pstmt.setString(2, name);
pstmt.setString(3, dept);
pstmt.setString(4, addr);
```

`setString()`, `setInt()` 등등의 함수 쓴다는 것 기억해두기

SELECT

- 기본적으로 step3까지는 동일(커넥션 관련이니깐)

```
String sql = "SELECT * FROM student";

PreparedStatement pstmt = conn.prepareStatement(sql);

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

`ResultSet` 객체로 받아주어야 함

```

while (rs.next()) {
    %>

    <tbody>
    <tr>
        <td><%=rs.getString("hakbun")%></td>
        <td><%=rs.getString("name")%></td>
        <td><%=rs.getString("dept")%></td>
        <td><%=rs.getString("addr")%></td>
    </tr>
</tbody>
<%
}

```

| `rs.next()` 값이 있다면~

| `rs.getString()` 함수로 받아온 결과를 뽑아냄

| 이 때 매개변수는 컬럼명과 동일

전송 받는 데이터 한글 인코딩

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

| 한글이 깨진다면 넘겨받는 파일 위쪽에 넣어주도록 하자

SELECT 풀버전

```

<%@ 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>Insert title here</title>
</head>
<body>
    <table>
        <thead>
            <tr>

```

```

        <td>학번</td>
        <td>이름</td>
        <td>학부</td>
        <td>사는 곳</td>
    </tr>
</thead>

<%

try {
    Class.forName("com.mysql.jdbc.Driver");
} catch (ClassNotFoundException err) {
    out.print("JDBC Driver loading error<br>" + err.getMessage());
}

Connection conn = null;

try {
    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/univ", "root", "0000");
} catch (SQLException err) {
    out.print("Connection Object error<br>" + err.getMessage());
}

String sql = "SELECT * FROM student";

PreparedStatement pstmt = conn.prepareStatement(sql);

ResultSet rs = pstmt.executeQuery();

while (rs.next()) {
    %>

<tbody>
    <tr>
        <td><%=rs.getString("hakbun")%></td>
        <td><%=rs.getString("name")%></td>
        <td><%=rs.getString("dept")%></td>
        <td><%=rs.getString("addr")%></td>
    </tr>
</tbody>
<%
    }

//6. Close connection(연결 끊기)
pstmt.close();
conn.close();
%>
</table>
</body>
</html>

```

Log in / Log out with Session

로그인의 세 가지 경우

1. 아이디와 비밀번호를 모두 정상적으로 입력한 경우
 - 로그인 성공
2. 비밀번호가 틀린 경우
 - 다시 시도하세요
3. 아이디가 DB에 존재하지 않는 경우
 - 회원가입 페이지로 리다이렉트

로그인 체크

- 연습이니까 학번을 비교해서 **이름**과 **학번**을 가져와보자
- 얘기했듯이 1~3번은 똑같음

```
String sql = "SELECT * FROM student WHERE hakbun=?";

PreparedStatement pstmt = conn.prepareStatement(sql);
pstmt.setString(1, hakbun);

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

- 역시 쿼리 적용해서 쓰기

```
if (!rs.isBeforeFirst()) {
    out.print("<script>alert('해당 학번은 존재하지 않습니다.');"
        + "history.back();"
        + "</script>");
    return;
}
rs.next(); //있는 그 데이터로 바로 가서
String dbhakbun = rs.getString("hakbun");
String dbname = rs.getString("name"); //데이터를 가져옴
```

rs.isBeforeFirst()

- 결과의 커서가 첫 번째 행 앞에 있으면 **true**
- 결과의 커서가 그 외의 위치에 있거나 결과 집합에 행이 들어 있지 않으면 **false**.

- 즉 값이 있으면 `true` 없으면 `false` 가 나온다

예외 발생

`java.sql.SQLException: Illegal operation on empty result set.`

- SELECT한 데이터가 없는 경우 스크립트 실행이 안되고 오류가 뜬.
- 데이터가 없을 때 `rs.next()` 를 해서 생긴 문제로 추정
- if문 걸렸을 때 `return` 넣어주는 것으로 해결

```
if (dbhakbun.equals(hakbun)) {
    session.setAttribute("hakbun", dbhakbun);
    session.setAttribute("name", dbname);
}
```

`session.setAttribute("hakbun", dbhakbun)` 으로 세션 생성

문자열 비교시에는 `.equals()` 쓴다는 것 유의

LoginCheck 전체 코드

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

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

<%
    try {
        Class.forName("com.mysql.jdbc.Driver");
    } catch (ClassNotFoundException err) {
        out.print("JDBC Driver loading error<br>" + err.getMessage());
    }

    Connection conn = null;

    try {
        conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/univ", "root", "0000");
```

```

} catch (SQLException err) {
    out.print("Connection Object error<br>" + err.getMessage());
}

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

String sql = "SELECT * FROM student WHERE hakbun=?";

PreparedStatement pstmt = conn.prepareStatement(sql);
pstmt.setString(1, hakbun);

ResultSet rs = pstmt.executeQuery();

if (!rs.isBeforeFirst()) {
    out.print("<script>alert('해당 학번은 존재하지 않습니다.');" + "history.back();" + "</script>");
    return;
}
rs.next(); //있는 그 데이터로 바로 가서
String dbhakbun = rs.getString("hakbun");
String dbname = rs.getString("name"); //데이터를 가져옴

//값이 같으면 세션 생성
if (dbhakbun.equals(hakbun)) {
    session.setAttribute("hakbun", dbhakbun);
    session.setAttribute("name", dbname);

    out.print(session.getAttribute("hakbun") + "(" + session.getAttribute("name") + ")님 방문을 환영합니다 <br>");
}

pstmt.close();
conn.close();
%>

```

Connection 파일 분리 (2번 방법)

- 1~3, 6번이 매번 똑같다며?
 - 그럼 하나의 파일로 만들어버리자
 - 굳이 JSP로 만들 필요도 없음
-
- `dbconnect.inc` 파일 만들. 1~3번 복사 및 붙여넣기(스크립트릿 포함)
 - `dbclose.inc` 파일 만들. 6번 복사 및 붙여넣기(스크립트릿 포함)
 - 처음보는 확장자라 어떻게 써야하나 물어보는데 무시
 - 내부에서 그냥 text-editor로 열어줌

```

<%@ include file="dbconnect.inc" %>

<%
//4. create Statement Object
PreparedStatement pstmt = conn.prepareStatement("CREATE DATABASE test");

//5. Excute SQL Query(쿼리 실행)
pstmt.executeUpdate();
%>

<%@ include file="dbclose.inc" %>

```

include 로 원래 들어갈 위치에 넣어주기만 하면, 그 텍스트를 그대로 가져옴

Connection Class로 분리 (3번 방법)

```

try {
    Class.forName("com.mysql.jdbc.Driver");
} catch (ClassNotFoundException err) {
    System.out.println("JDBC Driver loading error<br>" + err.getMessage());
}

Connection conn = null;

try {
    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/", "root", "0000");
} catch (SQLException err) {
    System.out.println("Connection Object error<br>" + err.getMessage());
}

return conn;

```

- 싹 다 가져와서
- out.print라는 jsp문법 빼고
- 임포트 좀 할거하면 끝!

```

public static void closeConnection(PreparedStatement pstmt,
                                   Connection conn) {
    try {
        pstmt.close();
    }
}

```



```

        conn.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

- `PreparedStatement` 와 `Connection` 객체를 받아와서 처리
- 오류 뜨는 건 `Try-Catch` 구문으로 잡아주면 끝!

별 차이는 없지만 그냥 올려보는 전체 클래스

```

import java.sql.*;

public class DBConClose {

    //DB 연결 공통부분 메서드
    public static Connection getConnection() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
        } catch (ClassNotFoundException err) {
            System.out.println("JDBC Driver loading error<br>" + err.getMessage());
        }

        Connection conn = null;

        try {
            conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/", "root", "0000");
        } catch (SQLException err) {
            System.out.println("Connection Object error<br>" + err.getMessage());
        }

        return conn;
    }

    //DB 해제 공통부분 메서드
    public static void closeConnection(PreparedStatement pstmt,
                                       Connection conn) {
        try {
            pstmt.close();
            conn.close();
        } catch (SQLException e) {
            System.out.println("Error " + e.getMessage());
        }
    }
}

```

만든 클래스 이용하여 JSP에서 실행

```
<%@page import="jdbc6steps.DBConClose"%>

<%
    Connection conn = DBConClose.getConnection();

    PreparedStatement pstmt = conn.prepareStatement("CREATE DATABASE test1");
    pstmt.executeUpdate();

    DBConClose.closeConnection(pstmt, conn);
%>
```

- `DBConClose` 의 메소드는 `Static` 선언 해놓았으므로 클래스명으로 접근가능

```
<%
    Connection conn = DBConClose.getConnection();

    //4. create Statement Object
    PreparedStatement pstmt = conn.prepareStatement("DROP DATABASE test1");

    //5. Excute SQL Query(쿼리 실행)
    pstmt.executeUpdate();

    DBConClose.closeConnection(pstmt, conn);
%>
```

- Drop도 똑같이 하면 됨~

다음주 예고

- 월요일은 `summernote` 라는 걸 할거다
- 내 컴퓨터에 게시판 템플릿을 붙일 수 있는 기능
- 좀 기대됨