

# SQL CLASS 3

## ▼ 지난 과제 복습

11. 월급이 1000에서 3000 사이인 직원들의 이름과 월급을 출력하세요

| ENAME  | SAL  |
|--------|------|
| BLAKE  | 2850 |
| CLARK  | 2450 |
| ⋮      | ⋮    |
| ADAMS  | 1100 |
| MILLER | 1300 |

between **작은값** and **큰값**; 순서 중요

sal >= 1000 AND sal <= 3000

※ 변형) 1000에서 3000 사이를 제외

⇒ sal NOT BETWEEN 1000 AND 3000;

⇒ sal < 1000 or sal > 3000

12. 이름의 첫 글자가 S 로 시작하는 직원들의 이름과 월급을 출력하세요

| ENAME | SAL  |
|-------|------|
| SMITH | 800  |
| SCOTT | 3000 |

S% → s로 시작하는 단어 찾기 와일드 카드

※ 변형) \_M% → 첫글자는 상관없이 두번째 글자가 M인 모든 단어 찾기

13. 커미션이 NULL인 직원들의 이름과 커미션을 출력해보세요

| ENAME  | COMM |
|--------|------|
| KING   |      |
| ⋮      | ⋮    |
| MILLER |      |

NULL값 : 데이터가 할당되지 않은 상태

**!= NULL**, 널값이 아닌 것을 찾을 때 **사용할 수 없음**

17. 영어 단어 SMITH 에서 SMI만 잘라서 출력하세요

|     |
|-----|
| SMI |
|-----|

```
SELECT SUBSTR('SMITH', 1, 3) FROM DUAL;
```

19. SMITH에서 M이 몇 번째 자리에 있는지 출력하세요

|   |
|---|
| 2 |
|---|

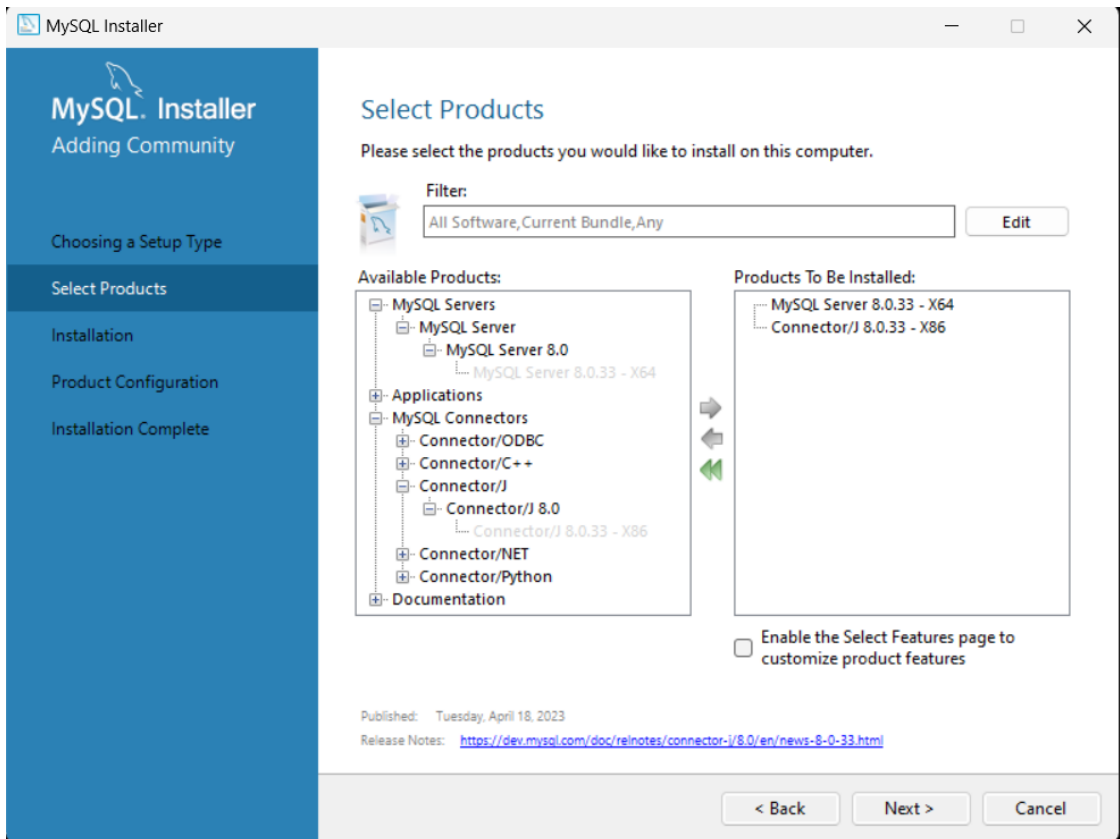
```
SELECT INSTR('SMITH', 'M') FROM DUAL;
```

19\_1) 함수 2개 이용하여 naver.com 추출!

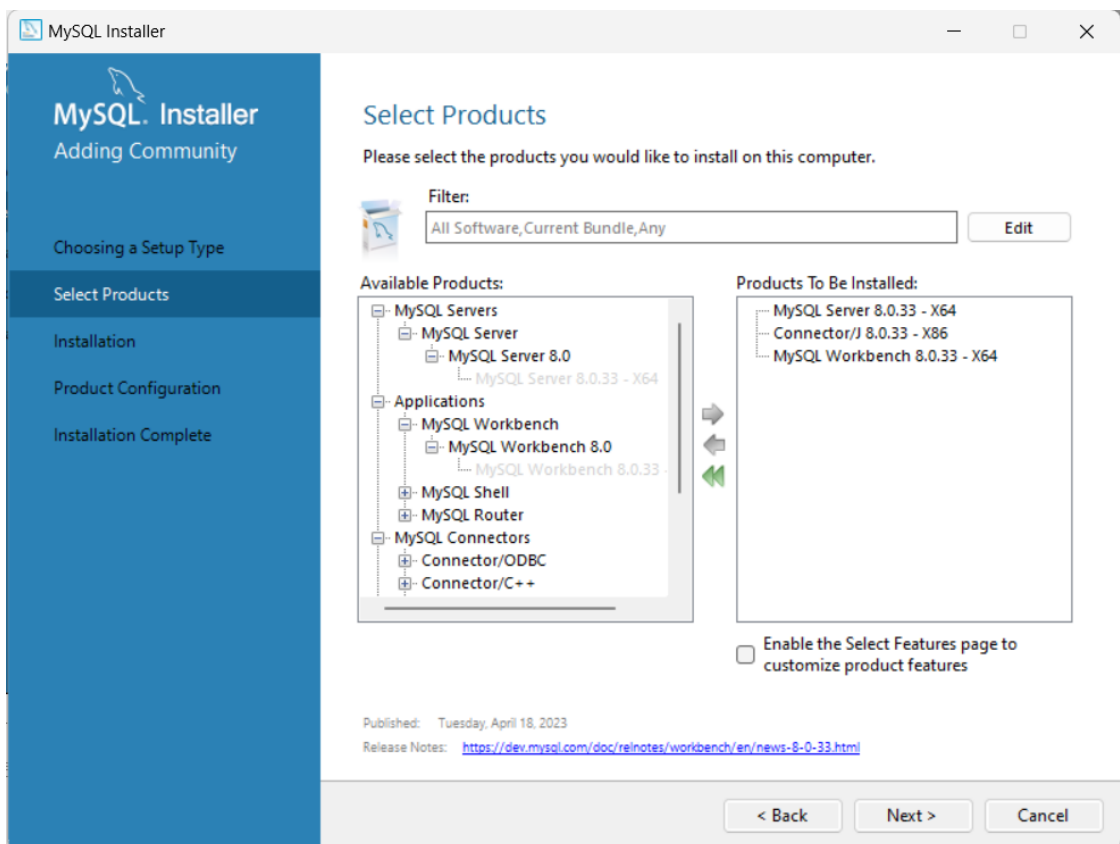
```
SELECT SUBSTR('austiny@naver.com', INSTR('austiny@naver.com','@')+1) FROM DUAL;
```

## ▼ 프로그램 다운로드

### ▼ MySQL Download



MySQL서버와 자바와 연결할 Connector/J 설치



Workbench (Oracle Developer역할) 나중에 사용할 수 있음

## Type and Networking

### Server Configuration Type

Choose the correct server configuration type for this MySQL Server installation. This setting will define how much system resources are assigned to the MySQL Server instance.

Config Type:

### Connectivity

Use the following controls to select how you would like to connect to this server.

- ☒ TCP/IP      Port:       X Protocol Port:
- ☒ Open Windows Firewall ports for network access
- ☐ Named Pipe      Pipe Name:
- ☐ Shared Memory      Memory Name:

### Advanced Configuration

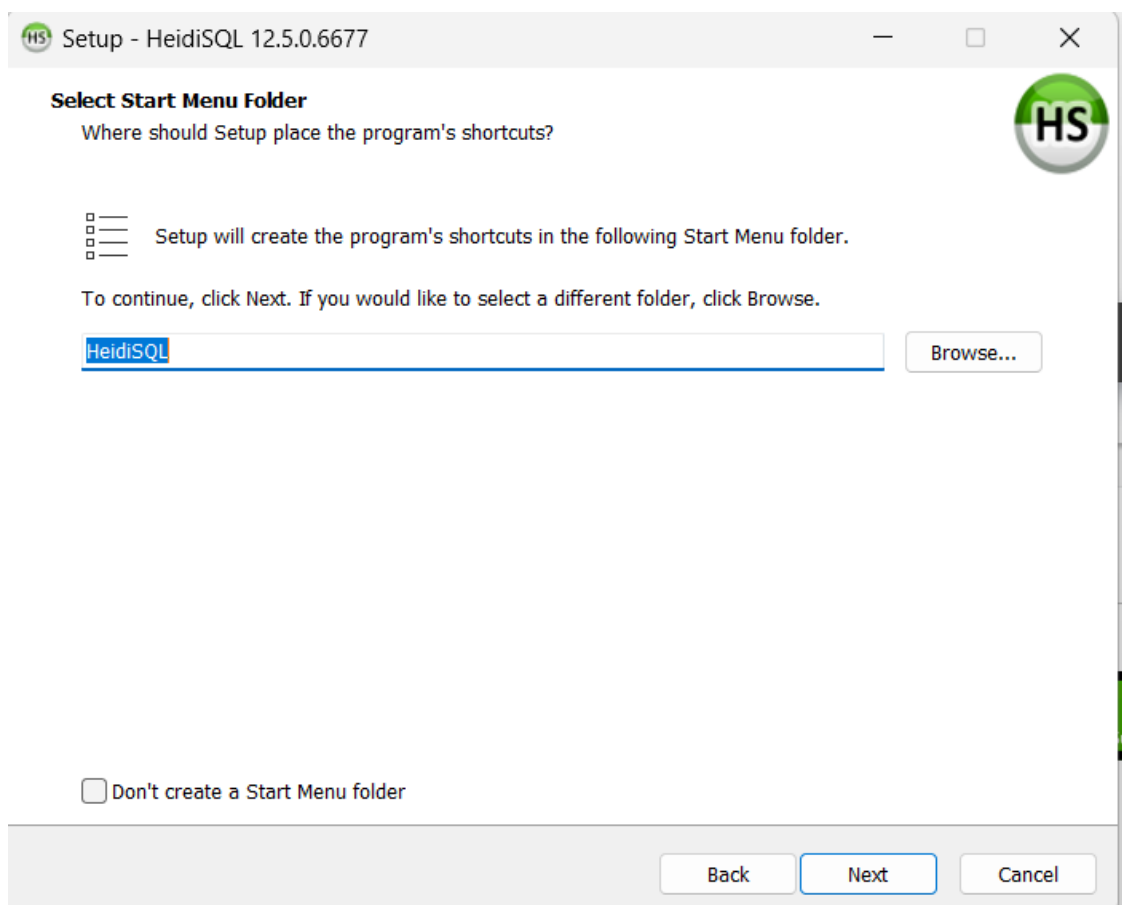
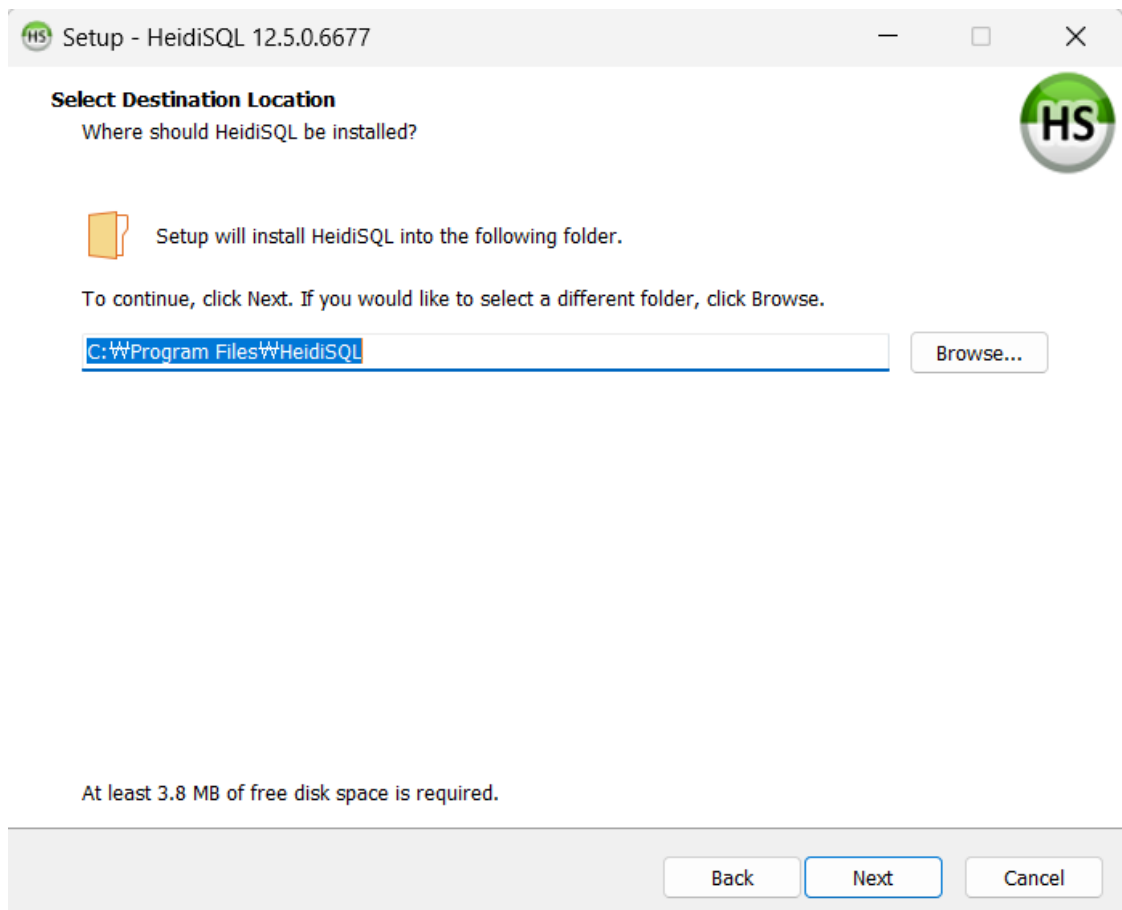
Select the check box below to get additional configuration pages where you can set advanced and logging options for this server instance.

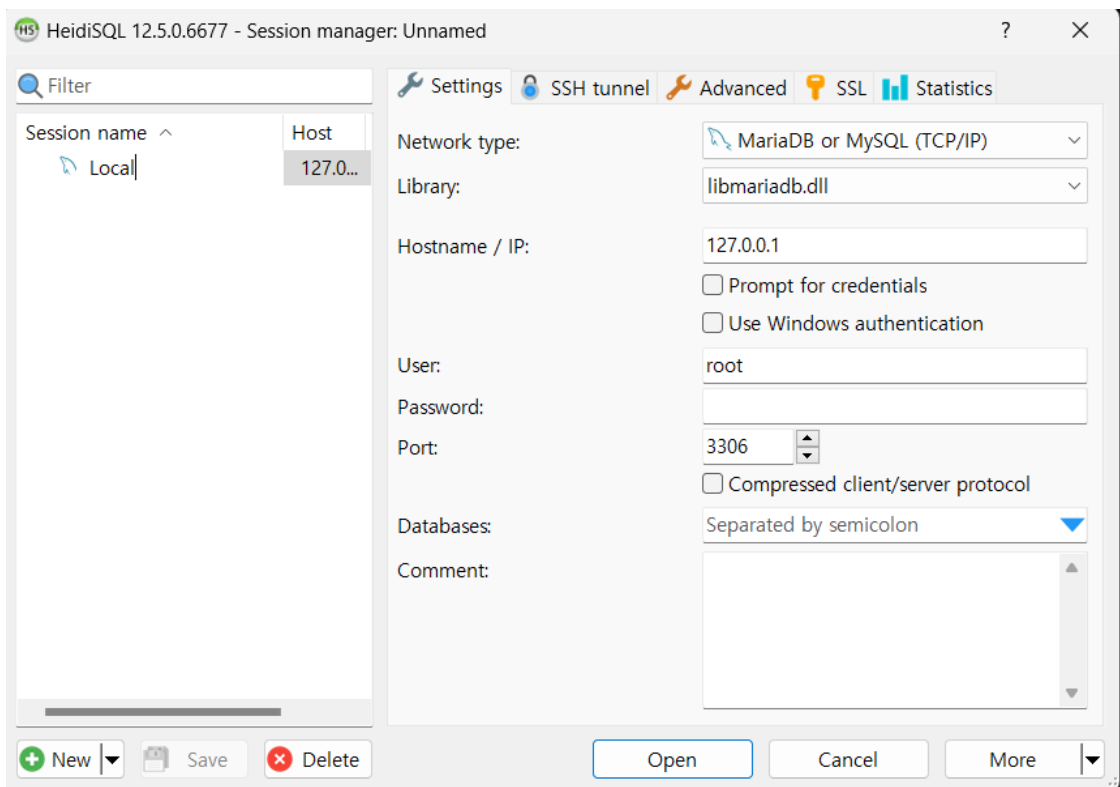
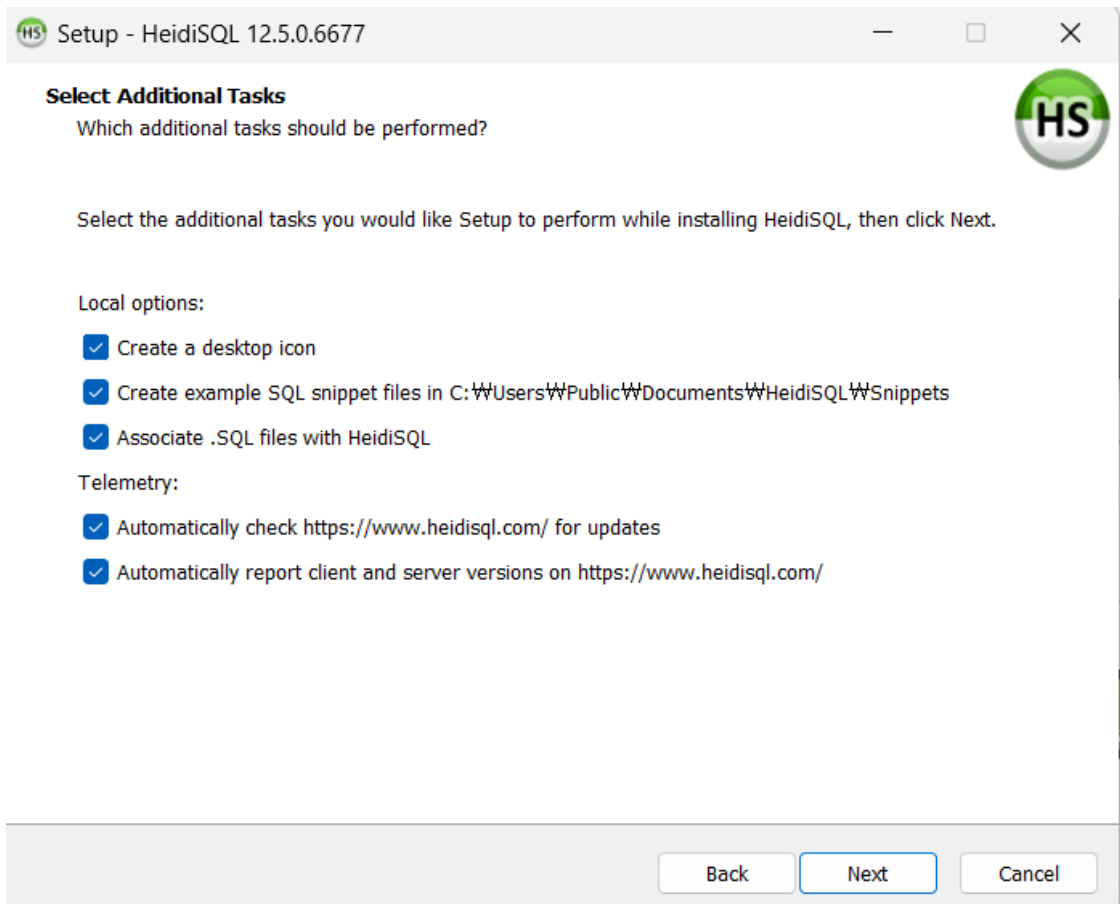
- ☐ Show Advanced and Logging Options

Next >

Cancel

## ▼ HeidiSQL Download





새로운 세션 생성 (패스워드 0000 통일)

## ▼ 정부프레임워크\_Eclipse Download

DOWNLOAD  
**다운로드**

실행환경   **개발환경**   운영환경   공통컴포넌트   모바일 실행환경   모바일 공통컴포넌트   모바일 디바이스 API

Release Notes   4.x 다운로드   **3.x 다운로드**   2.x 다운로드   1.0 다운로드

**개발환경 - 3.x 다운로드**

3.x 다운로드

제목+내용

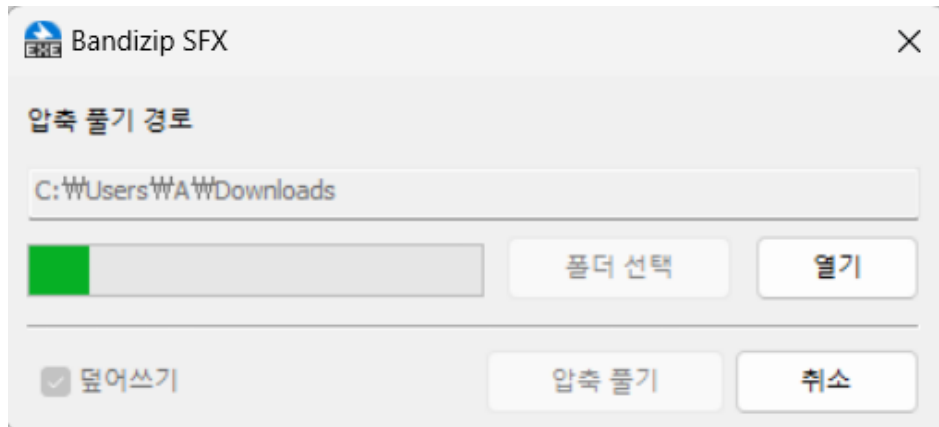
검색어를 입력하세요.

검색

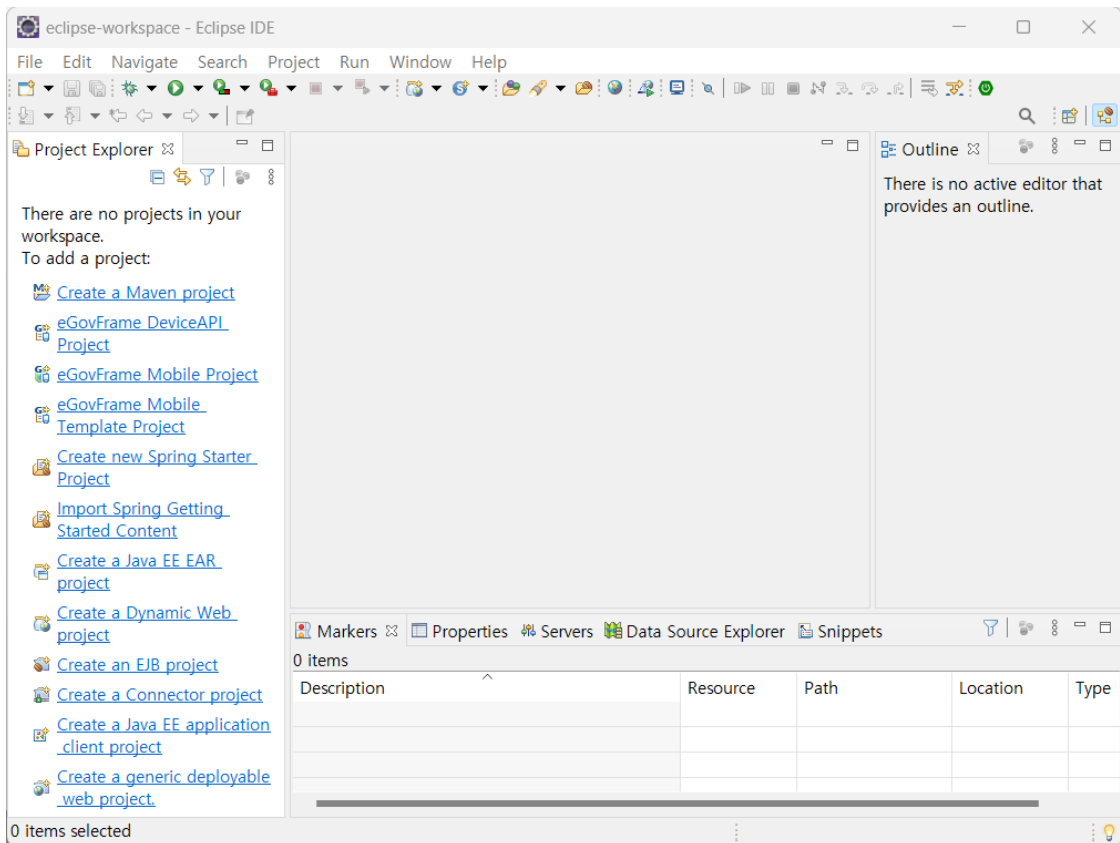
|        |   |                                |
|--------|---|--------------------------------|
| 3.10.0 | 개발자용 개발환경 64bit(Implementation Tool) Version 3.10.0 | 릴리즈 일자 - 2021-03-02            |
| 작성자    | 관리자   | 작성일   2021-03-02   조회수   50004 |

첨부파일   eGovFrameDev-3.10.0-64bit.exe [993,611,485 Byte]

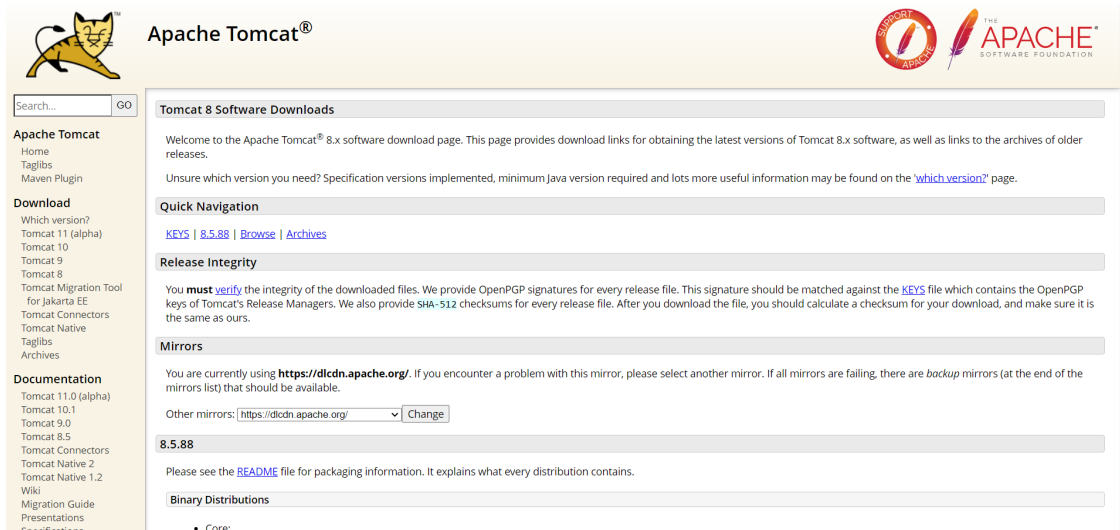
egov 다운로드 홈페이지



별도의 Installer 없이 압축 풀어서 사용 가능



정부에서 제공하는 Open Source Eclipse (기존 Eclipse와 동일)



Apache Tomcat 다운로드 화면



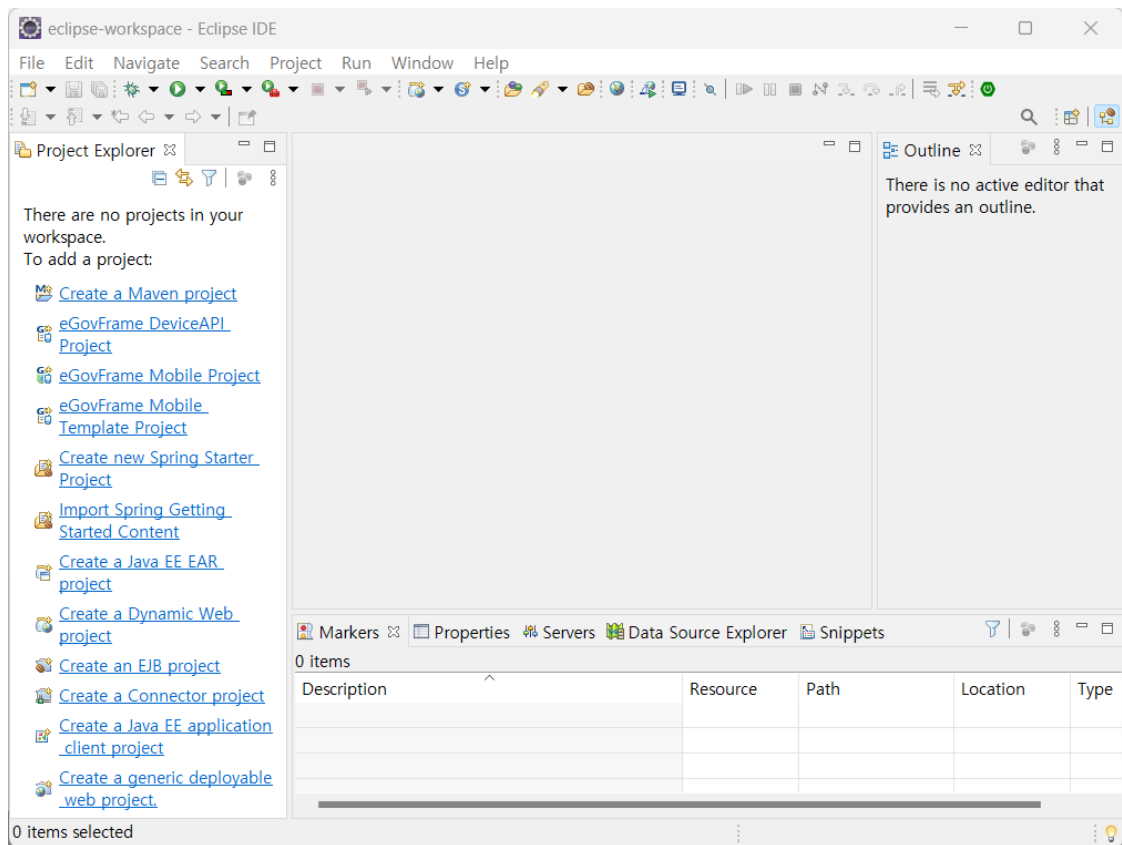
## 8.5.88

Please see the [README](#) file for packaging information. It explains what every distribution contains.

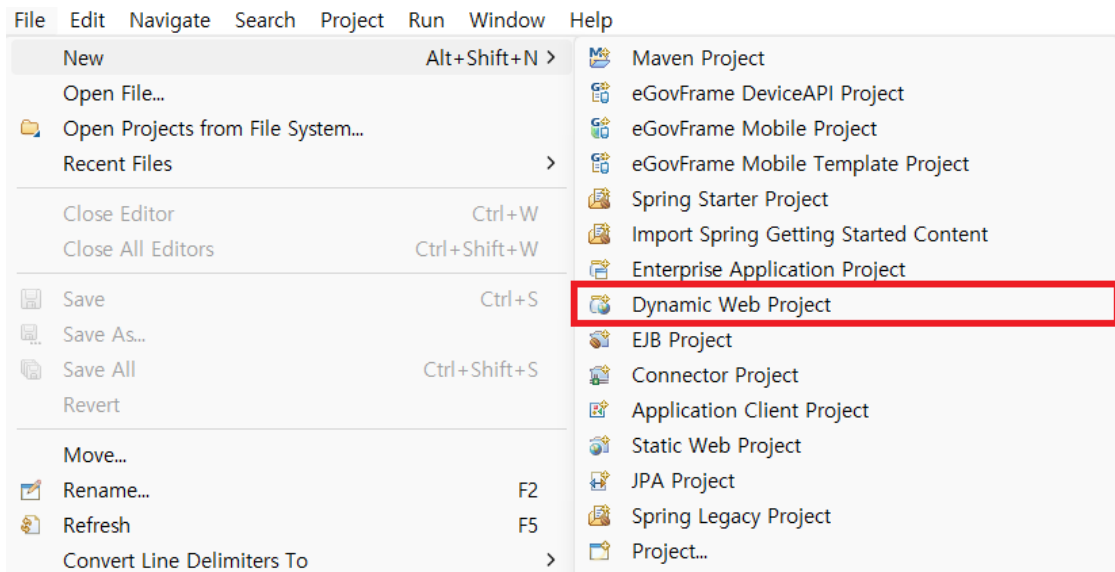
### Binary Distributions

- Core:
  - [zip](#) (pgp, sha512)
  - [tar.gz](#) (pgp, sha512)
  - [32-bit Windows zip](#) (pgp, sha512)
  - [64-bit Windows zip](#) (pgp, sha512)
  - [32-bit/64-bit Windows Service Installer](#) (pgp, sha512)
- Full documentation:
  - [tar.gz](#) (pgp, sha512)
- Deployer:
  - [zip](#) (pgp, sha512)
  - [tar.gz](#) (pgp, sha512)
- Extras:
  - [Web services jar](#) (pgp, sha512)
- Embedded:
  - [tar.gz](#) (pgp, sha512)
  - [zip](#) (pgp, sha512)

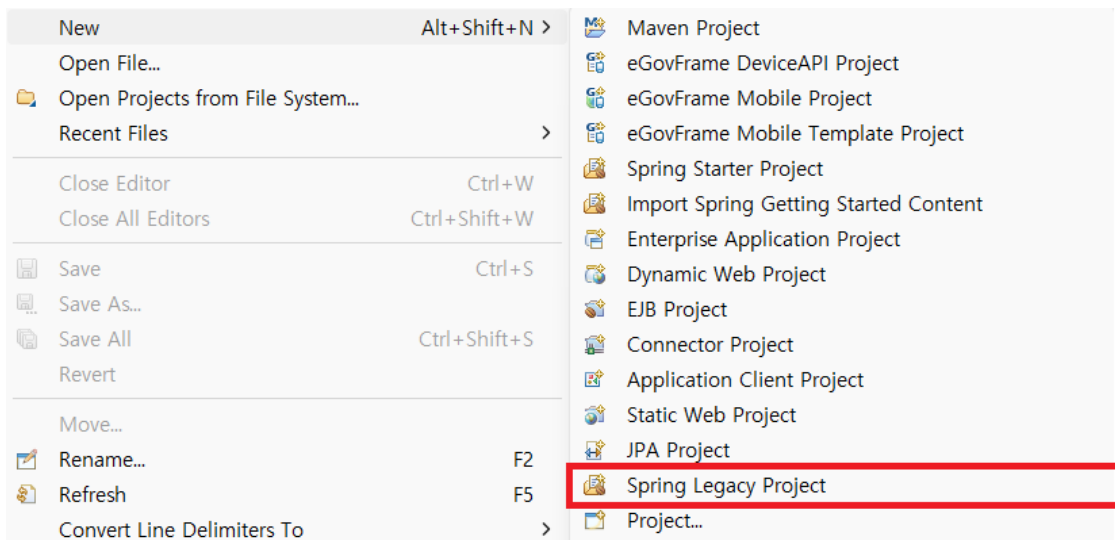
## ▼ Eclipse 초기 실행(JSP File)



Eclipse 기본 창



JSP 프로젝트 ⇒ Dynamic Web Project



Spring 프로젝트 ⇒ Spring Legacy Project

New Dynamic Web Project

### Dynamic Web Project

Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name: JDBCJSPexam

Project location

☒ Use default location

Location: C:\Users\WAW\workspace\WJDBCJSPexam Browse...

Target runtime

<None> New Runtime...

Dynamic web module version

3.0

Configuration

Default Configuration Modify...

The default configuration provides a good starting point. Additional facets can later be installed to add new functionality to the project.

EAR membership

☐ Add project to an EAR

EAR project name: EAR New Project...

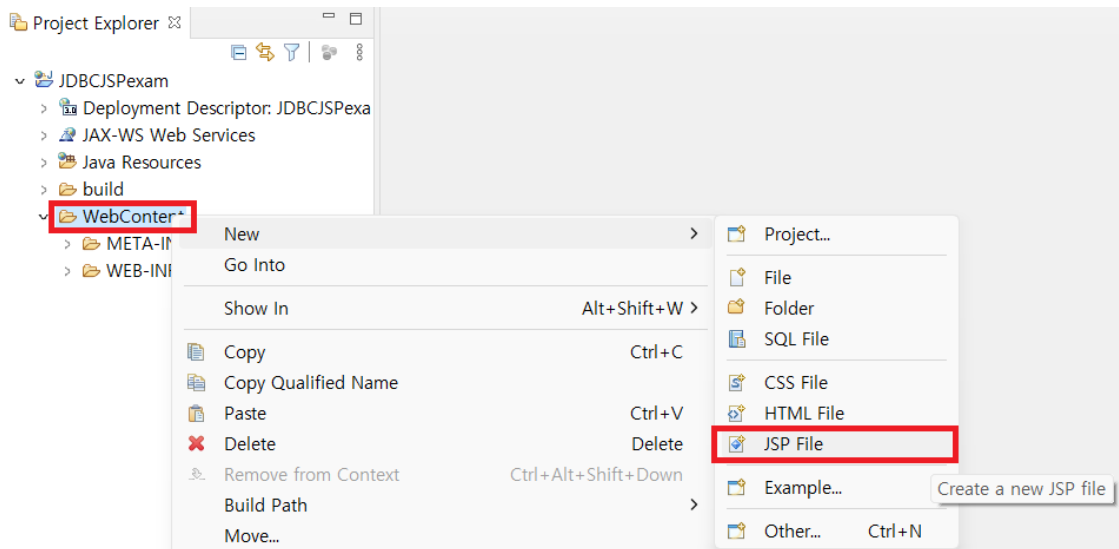
Working sets

☐ Add project to working sets New...

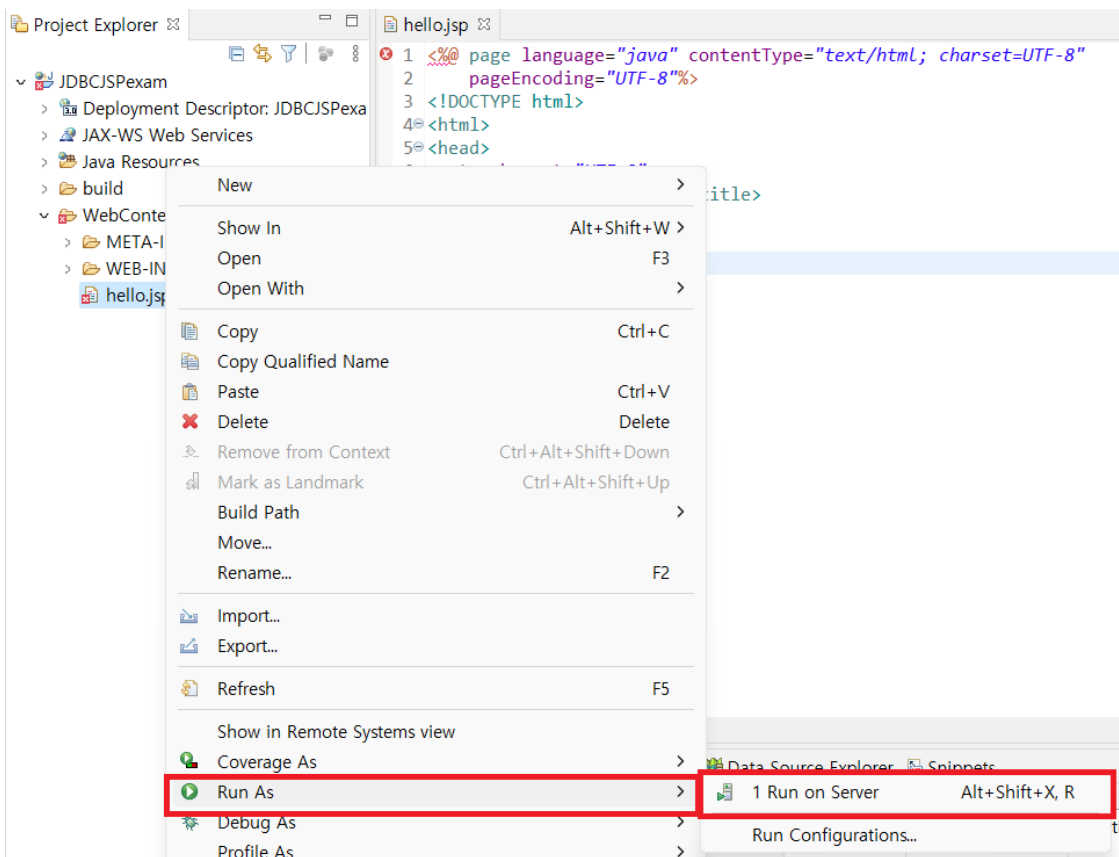
Working sets: Select...

? < Back Next > Finish Cancel


Dynamic Web Project 생성



JSP File 생성



서버에서 실행시키기

 Run On Server

**Run On Server**

Select which server to use

How do you want to select the server?

☐ Choose an existing server

☒ Manually define a new server

Select the server type:

type filter text


- Tomcat v8.0 Server
- Tomcat v8.5 Server**
- Tomcat v9.0 Server

Publishes and runs J2EE and Java EE Web projects and server configurations to a local Tomcat server.

Server's host name:

Server name:

☒ Always use this server when running this project



초기에는 Tomcat 경로 지정, 체크박스 클릭 시 항상 Tomcat 자동으로 인식하고 실행

Run On Server

### Tomcat Server

Specify the installation directory

Name:  
Apache Tomcat v8.5

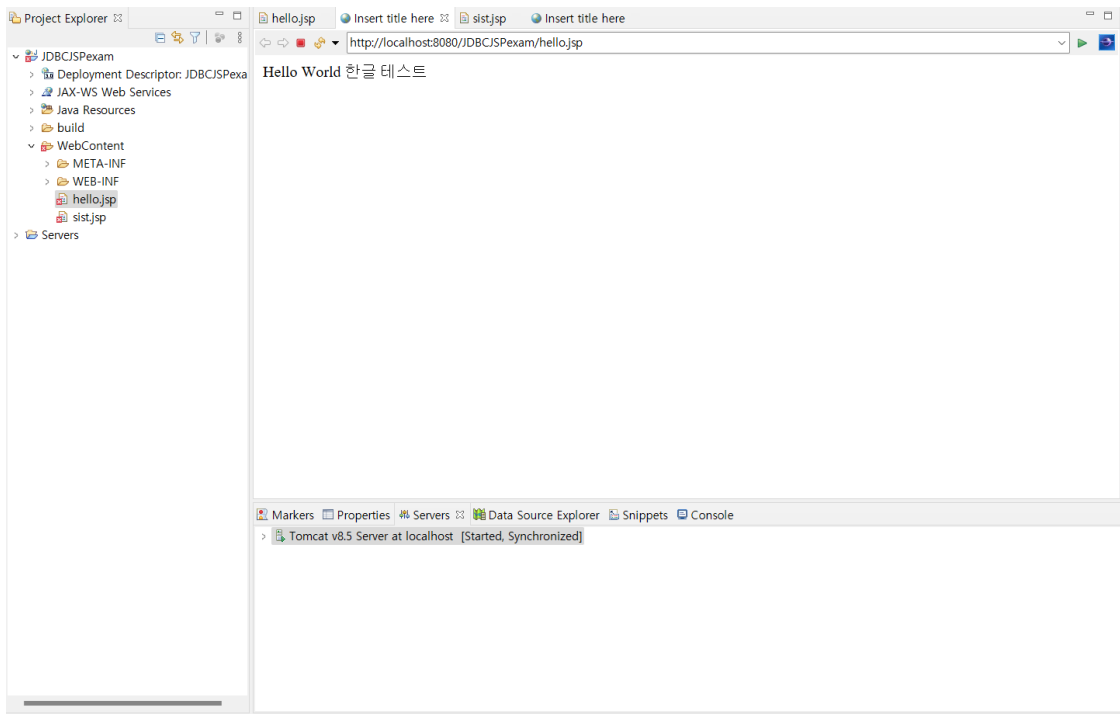
Tomcat installation directory:  
C:\develop\w\tomcat85 Browse...

Download and Install...

JRE:  
Workbench default JRE Installed JREs...

? < Back Next > Finish Cancel

Tomcat 경로 지정



서버에서 실행 성공

## ▼ DB

### ▼ DB 사용방법

JSP - JDBC(Connection Pool)

Spring - JDBC(MyBatis)

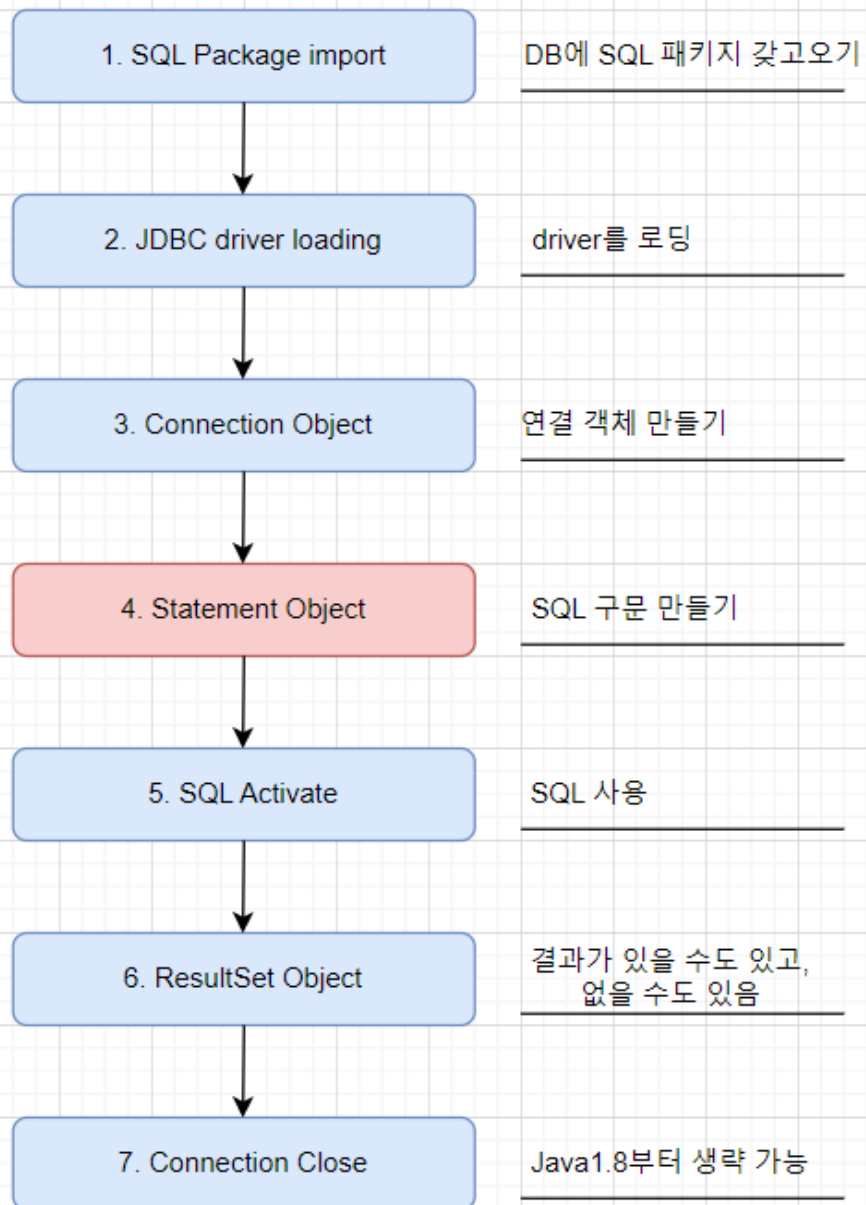
Boot - JDBC(JPA)

### ▼ JSP DB 연결방법 4가지

1. 각각의 페이지에서 직접 DB 사용
2. DB 연결만 별도로 공용으로 사용 → file로 구성
3. DB 연결만 별도로 공용으로 사용 → 자바 class로 구성
4. Connection Pool로 DB 사용 ⇒ 실제로 주로 사용

### ▼ JSP DB 연결 과정

## JSP DB 연결 7단계



- 1단계



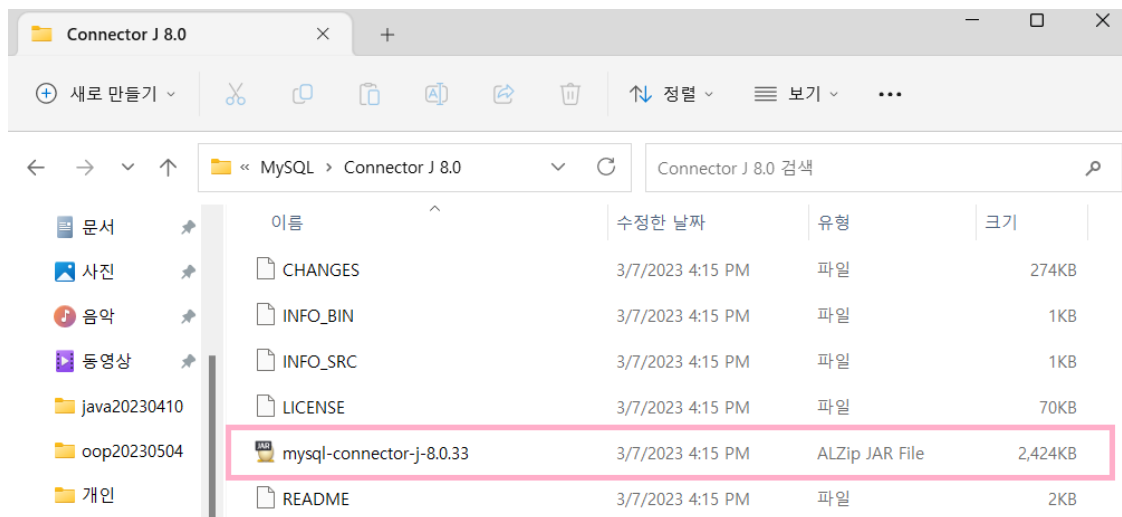
```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ page import="java.sql.*" %>
4 <!-- Step 1 import package -->
5 <!DOCTYPE html>
6 <html>
7 <head>
8 <meta charset="UTF-8">
9 <title>Step 1</title>
10 </head>
11 <body>
12
13 </body>
14 </html>

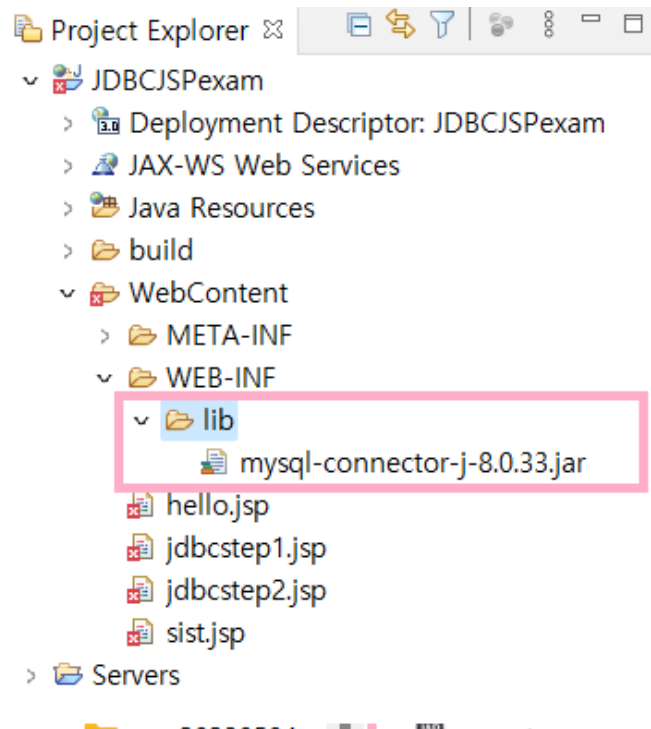
```

1단계 package import

- 2단계



2단계 connector 위치 확인



2단계 connector를 라이브러리에 복사

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6   <meta charset="UTF-8">
7   <title>Step 2</title>
8 </head>
9 <body>
10 <!-- Step 2 Load JDBC Driver -->
11 <%
12 try {
13     Class.forName("com.mysql.jdbc.Driver");
14     out.print("드라이버 로드 성공! <br>");
15 } catch(ClassNotFoundException err) {
16     out.print("드라이버 로드 실패! <br>" + err.getMessage());
17 }
18 %>
19 </body>
20 </html>

```

2단계 Driver loading ⇒ 드라이버 로드 성공! 출력

### • 3단계

```
<!-- Step 3 Connection Object-->
```

```
<%
```

```
Connection conn = null;

try {
    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/", "root", "0000");
    out.print("연결 객체 생성 성공 !!!<br>");
} catch (SQLException err) {
    out.print("연결 객체 생성 실패 !!!<br>" + err.getMessage());
}
%>
```

conn이라는 객체 생성

- 4단계

```
%>
```

```
<!-- Step 4 Statement Object-->
```

```
<%
```

```
String sql = "CREATE DATABASE exam";
PreparedStatement pstmt = conn.prepareStatement(sql);
out.print("구문 생성 성공");
```

```
%>
```

SQL구문 생성

- 5단계

```
<!-- Step 5 SQL Activate -->
```

```
<%
```

```
pstmt.executeUpdate();
out.print("DB 생성 성공!<br>");
```

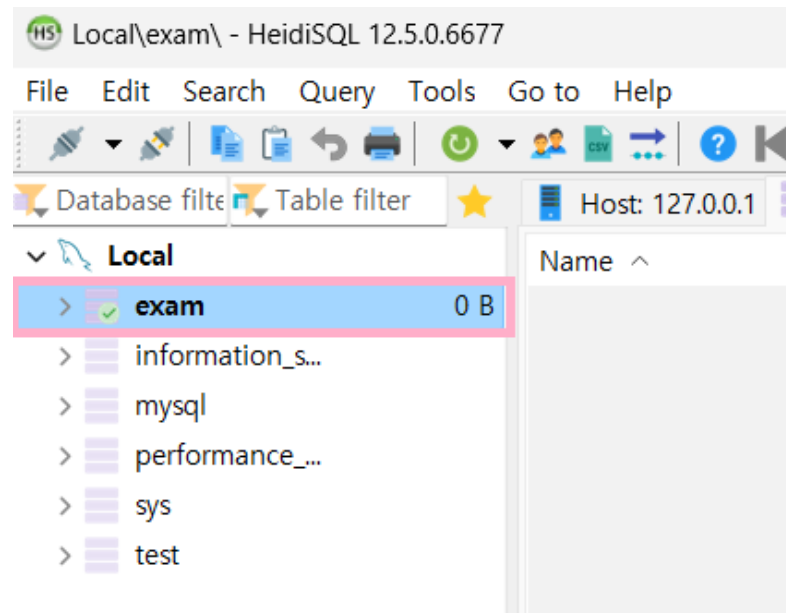
```
%>
```

exam DB생성

- 6단계

```
<!-- Step 6 ResultSet Object -->
```

```
<!-- DB 생성은 가지고 오는 값이 없으므로 생략 -->
```



Heidi에 exam 생성된 것 확인

- 7단계

```

<!-- Step 7 Connection Close -->
<!-- 자바 1.8부터는 생략 가능 -->
<%
    pstmt.close();
    conn.close();
%>

```

연결 끊기

```

<%@ 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>
<%
    try {
        Class.forName("com.mysql.jdbc.Driver");
        out.print("드라이버 로드 성공 !!!<br>");

    }catch(ClassNotFoundException err) {

        out.print("드라이버 로드 실패 !!!<br>" + err.getMessage());
    }

    Connection conn = null;

    try {

```

```

    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/", "root", "0000");
    out.print("연결 객체 생성 성공 !!!<br>");

} catch (SQLException err) {

    out.print("연결 객체 생성 실패 !!!<br>" + err.getMessage());
}

String sql = "CREATE DATABASE exam";
PreparedStatement pstmt = conn.prepareStatement(sql);
out.print("구문 생성 성공!<br>");

pstmt.executeUpdate();
out.print("DB 생성 성공!<br>");

pstmt.close();
conn.close();
%>
</body>
</html>

```

## ▼ JSP DB 연결 실습★

### ★CRUD실습★

- 테이블 생성

```

Connection conn = null;

try {
    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/exam", "root", "0000");
    out.print("연결 객체 생성 성공 !!!<br>");

} catch (SQLException err) {

    out.print("연결 객체 생성 실패 !!!<br>" + err.getMessage());
}

String sql = "CREATE TABLE board("
    + "bno int,"
    + "btitle varchar(50),"
    + "bcontent varchar(50))";

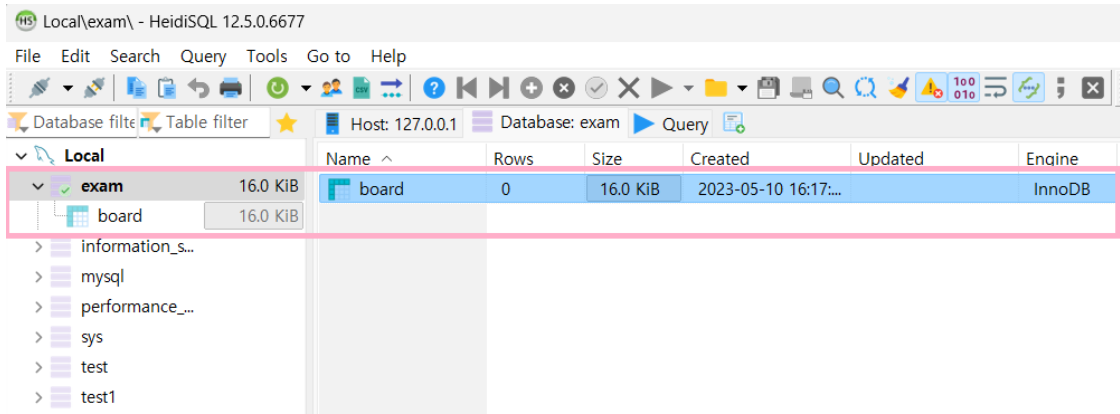
PreparedStatement pstmt = conn.prepareStatement(sql);
out.print("구문 생성 성공!<br>");

pstmt.executeUpdate();
out.print("DB 생성 성공!<br>");

```

★경로 설정★

자바에서 테이블 생성 시 경로 설정 중요!



Heidi에서 확인 가능

```
<%@ 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>
<%
    try {

        Class.forName("com.mysql.jdbc.Driver");
        out.print("드라이버 로드 성공 !!!<br>");

    }catch(ClassNotFoundException err) {

        out.print("드라이버 로드 실패 !!!<br>" + err.getMessage());
    }

    Connection conn = null;

    try {

        conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/exam","root","0000");
        out.print("연결 객체 생성 성공 !!!<br>");

    }catch(SQLException err) {

        out.print("연결 객체 생성 실패 !!!<br>" + err.getMessage());
    }

    String sql = "CREATE TABLE board("
        + "bno int,"
        + "btitle varchar(50),"
        + "bcontent varchar(50))";

    PreparedStatement pstmt = conn.prepareStatement(sql);
    out.print("구문 생성 성공!<br>");

    pstmt.executeUpdate();
    out.print("DB 생성 성공!<br>");
```

```

    pstmt.close();
    conn.close();
%>
</body>
</html>

```

## • 데이터 대입

```

Connection conn = null;

try {

    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/exam","root","0000");
    out.print("연결 객체 생성 성공 !!!<br>");

} catch(SQLException err) {

    out.print("연결 객체 생성 실패 !!!<br>" + err.getMessage());
}

String sql = "INSERT INTO board values(1,'제목', '내용')";

PreparedStatement pstmt = conn.prepareStatement(sql);
out.print("삽입 성공!<br>");

pstmt.executeUpdate();
out.print("DB 생성 성공!<br>");

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

```

테이블 안에 값 대입

```

Connection conn = null;

try {

    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/exam","root","0000");
    out.print("연결 객체 생성 성공 !!!<br>");

} catch(SQLException err) {

    out.print("연결 객체 생성 실패 !!!<br>" + err.getMessage());
}

String sql = "INSERT INTO board values(?, ?, ?)";

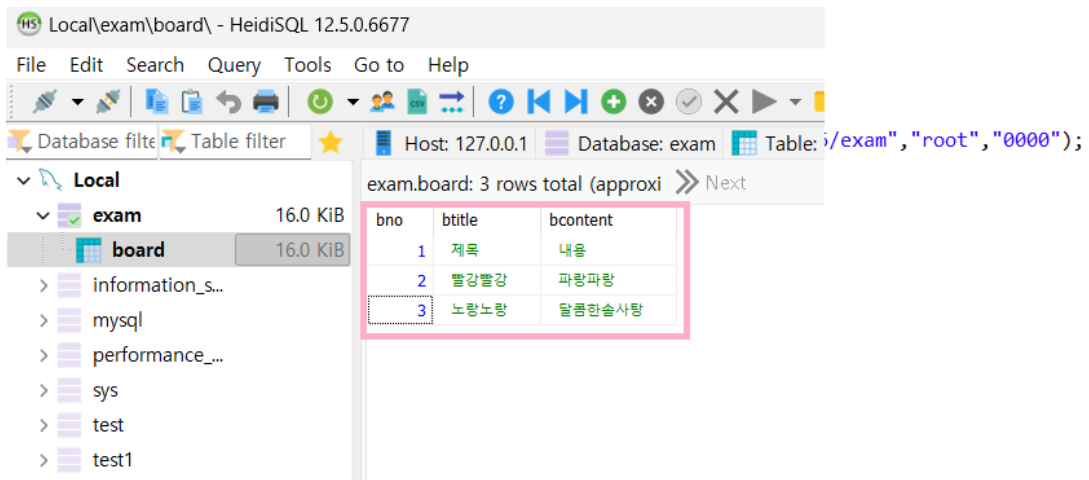
PreparedStatement pstmt = conn.prepareStatement(sql);
pstmt.setInt(1, 2);
pstmt.setString(2, "test2");
pstmt.setString(2, "content2");
out.print("삽입 성공!<br>");

pstmt.executeUpdate();
out.print("DB 생성 성공!<br>");

```

자리 표시자

더 쉬운 방법(자리 표시자 사용)



Heidi에서 테이블에 값 들어온 것 확인

```
<%@ 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>
<%
    try {

        Class.forName("com.mysql.jdbc.Driver");
        out.print("드라이버 로드 성공 !!!<br>");

    }catch(ClassNotFoundException err) {

        out.print("드라이버 로드 실패 !!!<br>" + err.getMessage());
    }

    Connection conn = null;

    try {

        conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/exam","root","0000");
        out.print("연결 객체 생성 성공 !!!<br>");

    }catch(SQLException err) {

        out.print("연결 객체 생성 실패 !!!<br>" + err.getMessage());
    }

    String sql = "INSERT INTO board values(?,?,?)";

    PreparedStatement pstmt = conn.prepareStatement(sql);
    pstmt.setInt(1, 2);
    pstmt.setString(2, "빨강빨강");
    pstmt.setString(3, "파랑파랑");
    out.print("삽입 성공 !!!<br>");

    pstmt.executeUpdate();
```



```

        out.print("디비 생성 성공 !!!<br>");

%>
</body>
</html>

```

- 데이터 삭제

```

Connection conn = null;

try {

    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/exam","root","0000");
    out.print("연결 객체 생성 성공 !!!<br>");

}catch(SQLException err) {

    out.print("연결 객체 생성 실패 !!!<br>" + err.getMessage());

}

String sql = "DELETE FROM board where bno = ?";

PreparedStatement pstmt = conn.prepareStatement(sql);
pstmt.setInt(1, 3);

pstmt.executeUpdate();
out.print("삭제 성공 !!!<br>");
out.print("DB 생성 성공!<br>");

```

delete 사용하여 데이터 삭제

Local\exam\board\ - HeidiSQL 12.5.0.6677

File Edit Search Query Tools Go to Help

Database filter Table filter Host: 127.0.0.1 Database: exam

exam.board: 2 rows total (approx) Next

| bno | btitle | bcontent |
|-----|--------|----------|
| 1   | 제목     | 내용       |
| 2   | 빨강빨강   | 파랑파랑     |

```

pstmt.executeUpdate();
out.print("삭제 성공 !!!<br>");

```

Heidi에서 확인 가능

```

<%@ 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>
<%
    try {

        Class.forName("com.mysql.jdbc.Driver");
        out.print("드라이버 로드 성공 !!!<br>");

    }catch(ClassNotFoundException err) {

        out.print("드라이버 로드 실패 !!!<br>" + err.getMessage());
    }

    Connection conn = null;

    try {

        conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/exam","root","0000");
        out.print("연결 객체 생성 성공 !!!<br>");

    }catch(SQLException err) {

        out.print("연결 객체 생성 실패 !!!<br>" + err.getMessage());
    }

    String sql = "DELETE FROM board where bno = ?";

    PreparedStatement pstmt = conn.prepareStatement(sql);
    pstmt.setInt(1, 3);

    pstmt.executeUpdate();
    out.print("삭제 성공 !!!<br>");

%>
</body>
</html>

```

- 데이터 변경

```

Connection conn = null;

try {
    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/exam","root","0000");
    out.print("연결 객체 생성 성공 !!!<br>");

} catch (SQLException err) {

    out.print("연결 객체 생성 실패 !!!<br>" + err.getMessage());
}

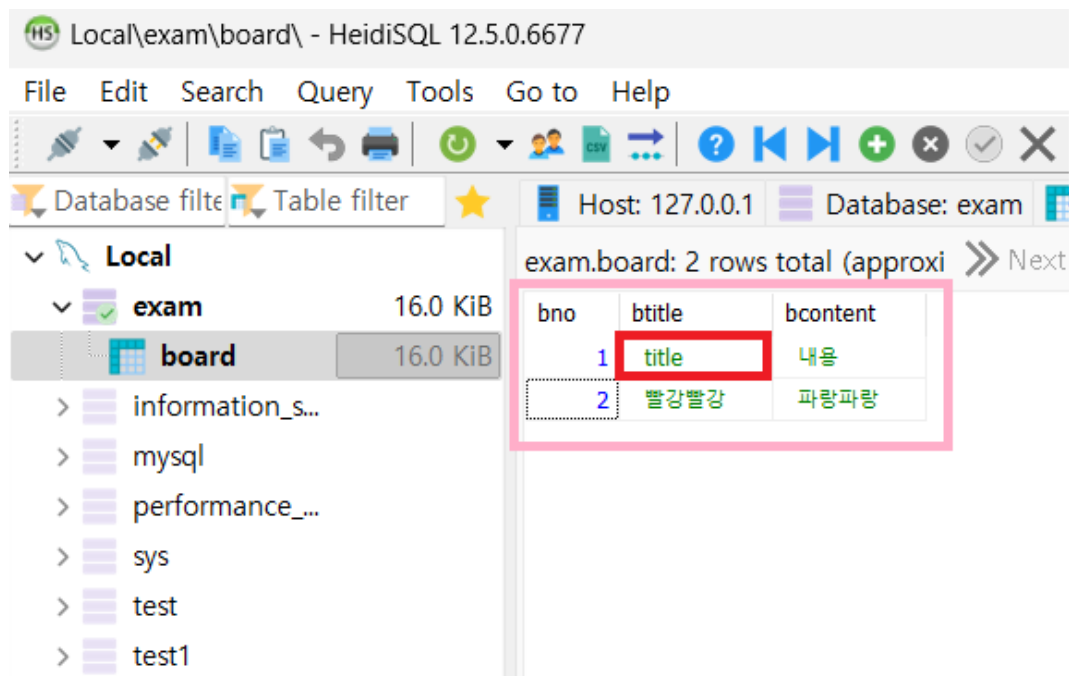
String sql = "UPDATE board SET btitle = ? WHERE bno = ?";

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

pstmt.executeUpdate();
out.print("변경 성공 !!!<br>");

```

1번째 행 중 btitle에서 “제목” → “title”로 변경



Heidi에서 “제목” → “title”로 변경 확인

```

<%@ 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>
<%
try {
    Class.forName("com.mysql.jdbc.Driver");

```

```

        out.print("드라이버 로드 성공 !!!<br>");

    }catch(ClassNotFoundException err) {

        out.print("드라이버 로드 실패 !!!<br>" + err.getMessage());
    }

    Connection conn = null;

    try {

        conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/exam","root","0000");
        out.print("연결 객체 생성 성공 !!!<br>");

    }catch(SQLException err) {

        out.print("연결 객체 생성 실패 !!!<br>" + err.getMessage());
    }

    String sql = "UPDATE board SET btitle = ? WHERE bno = ?";

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

    pstmt.executeUpdate();
    out.print("변경 성공 !!!<br>");
    %>
</body>
</html>

```