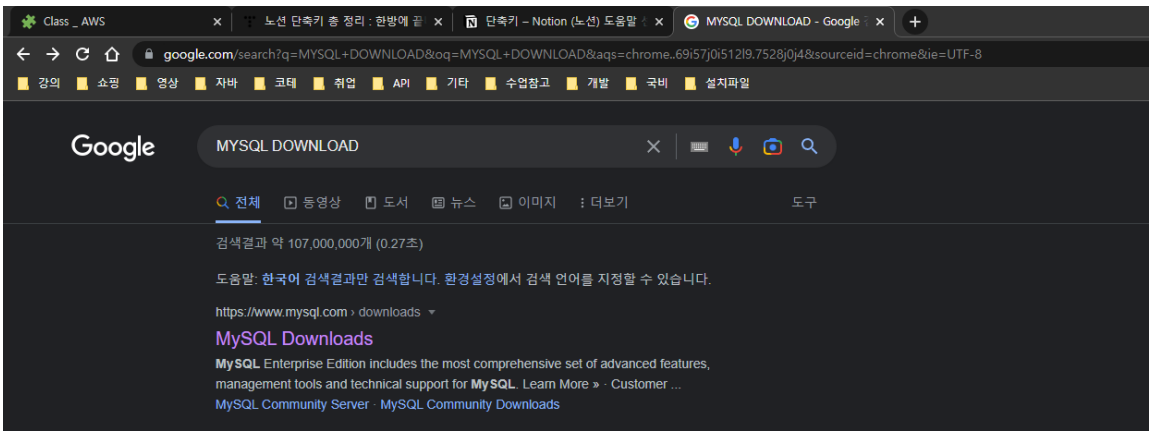
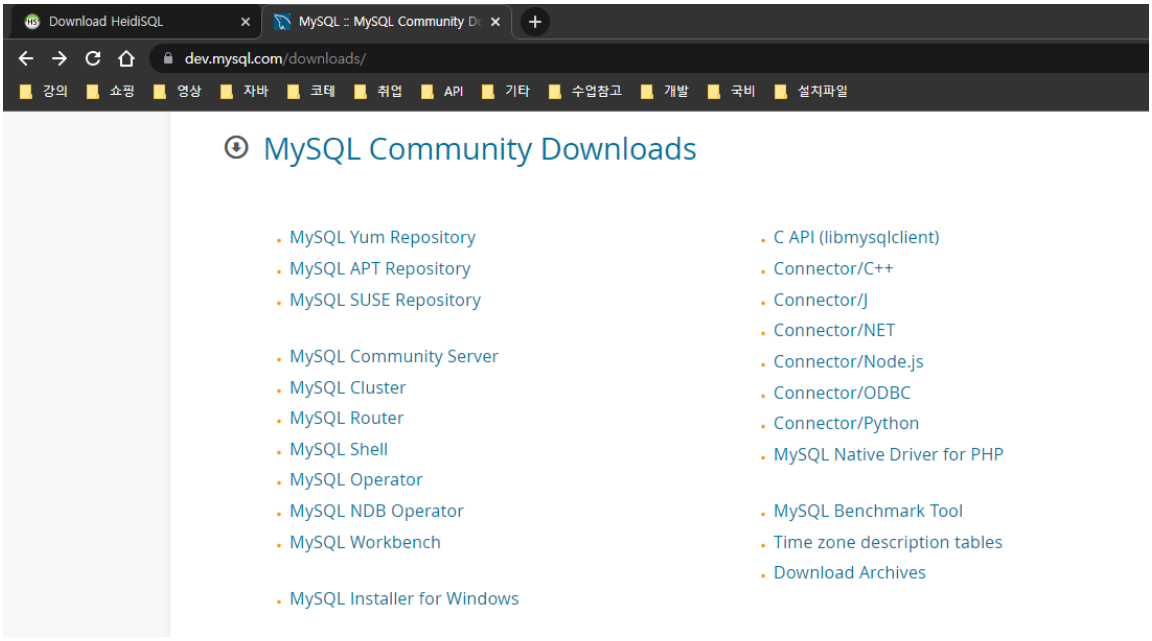
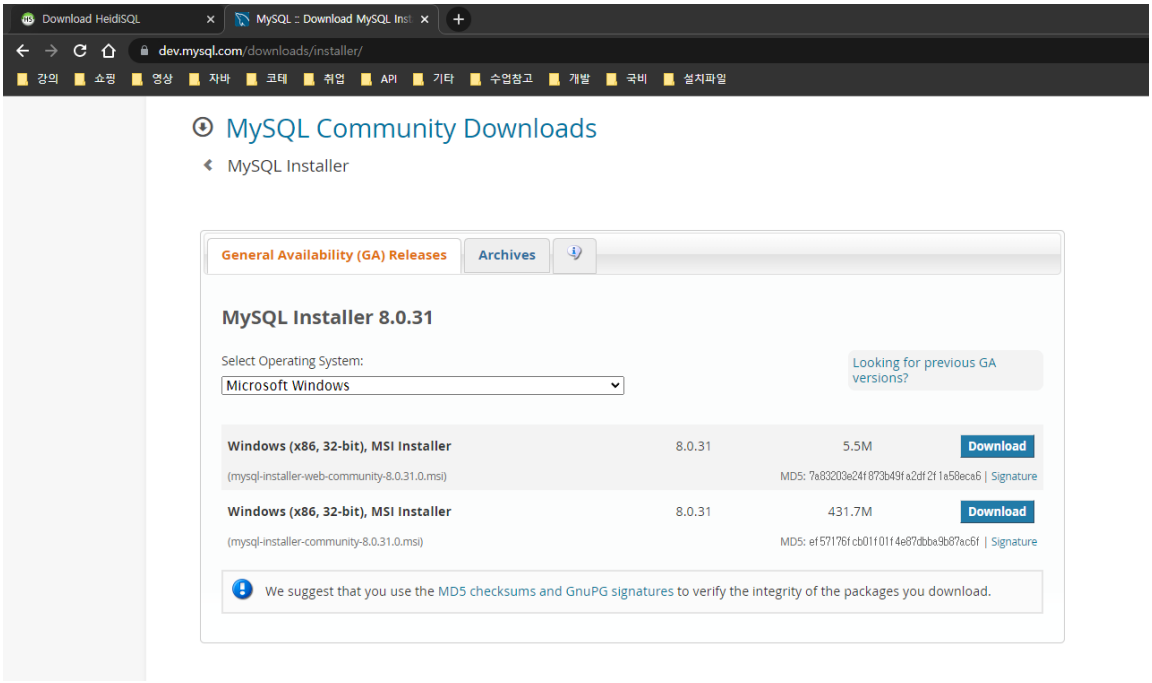
Class 20230113

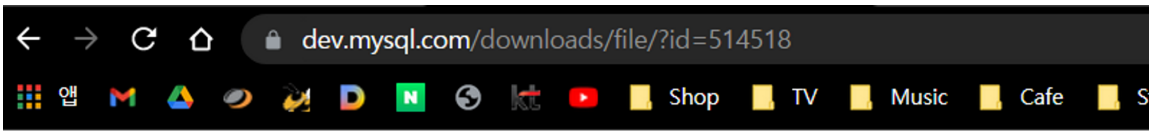
1교시 - SQL 설정 (9:00 ~ 9:50)

▼ MYSQL 설치









MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

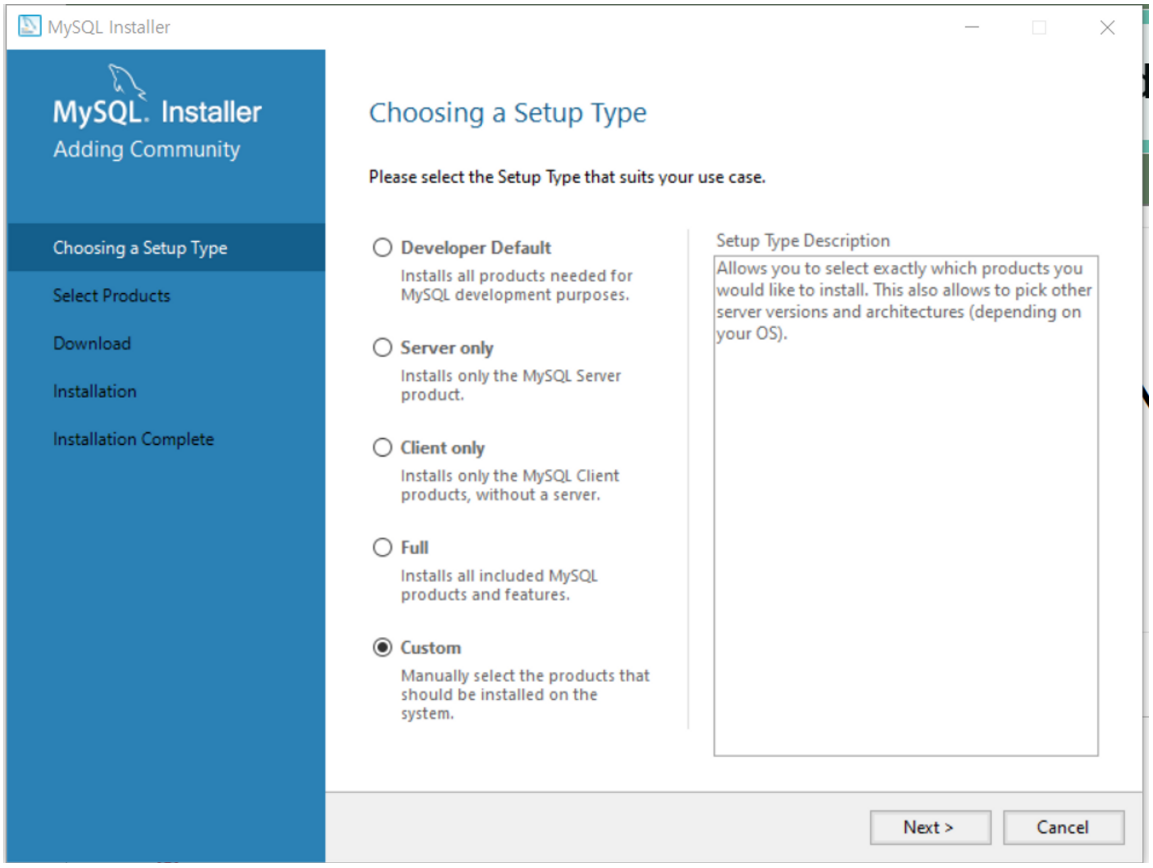
- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

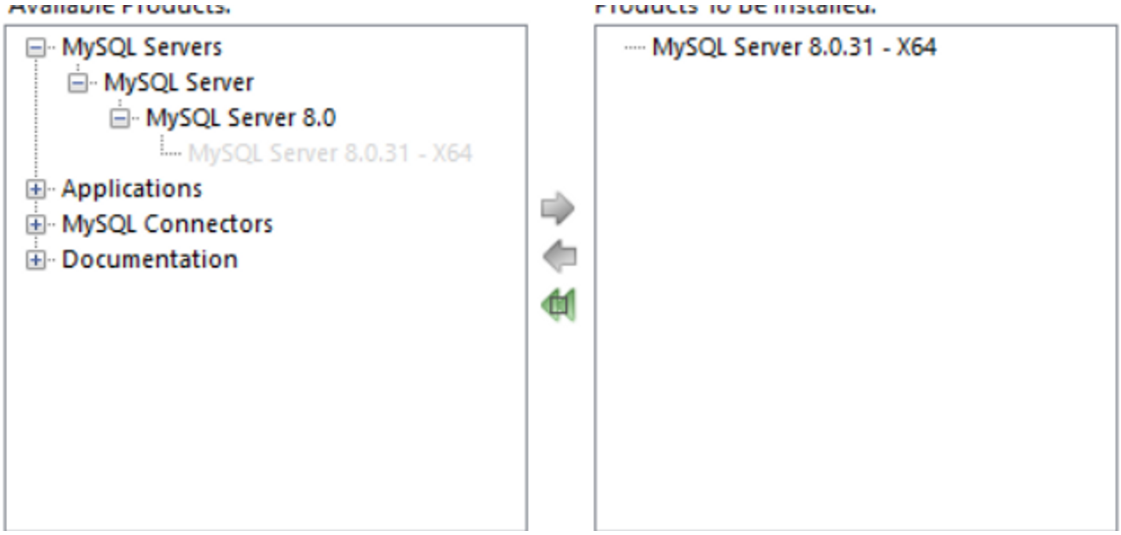
Login »
using my Oracle Web account

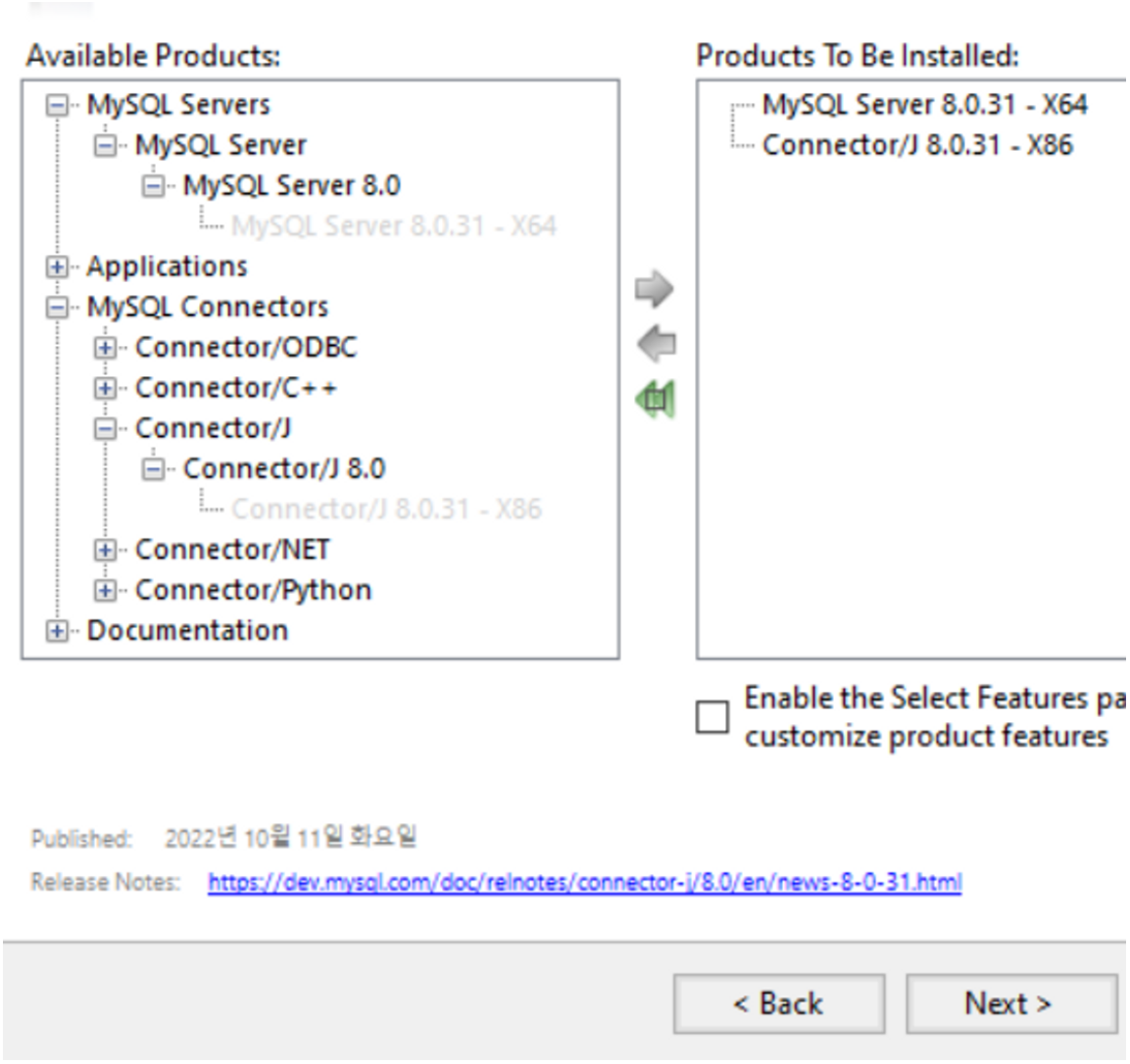
Sign Up »
for an Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.

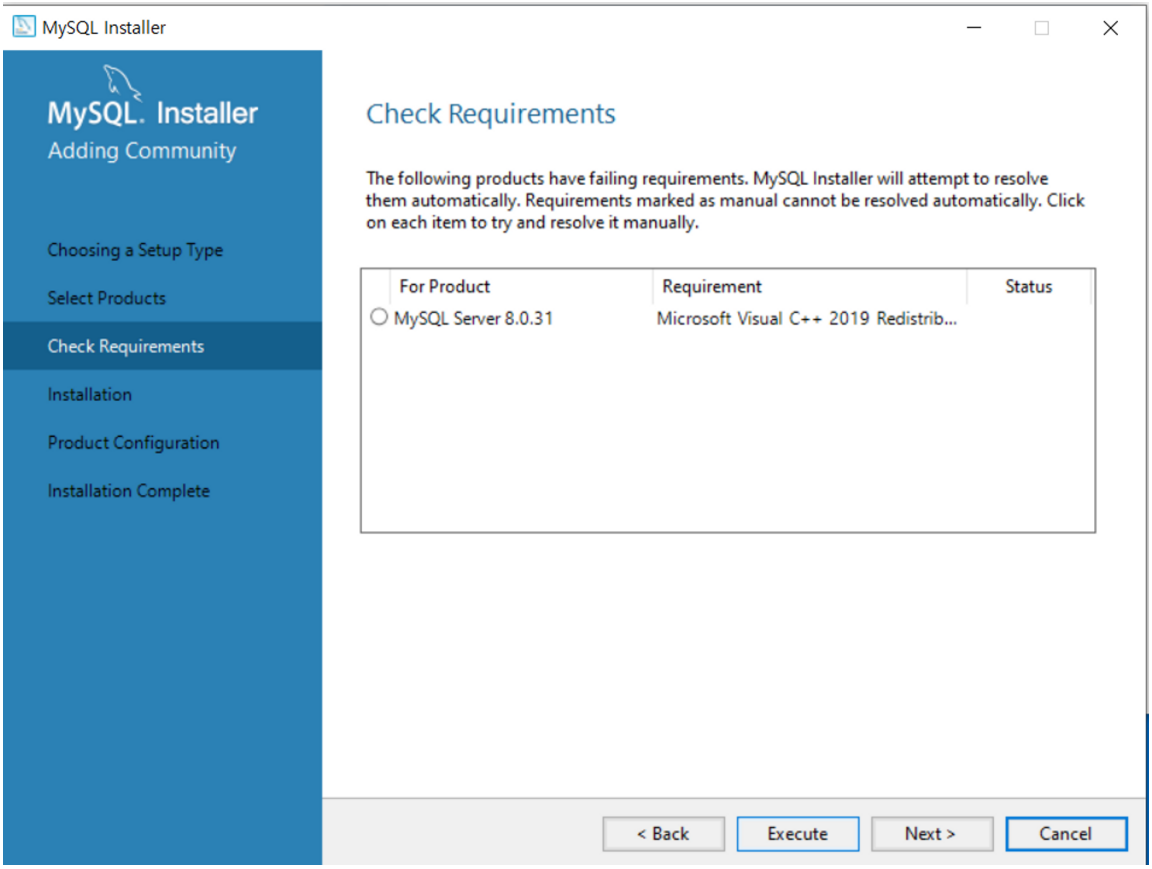
No thanks, just start my download.

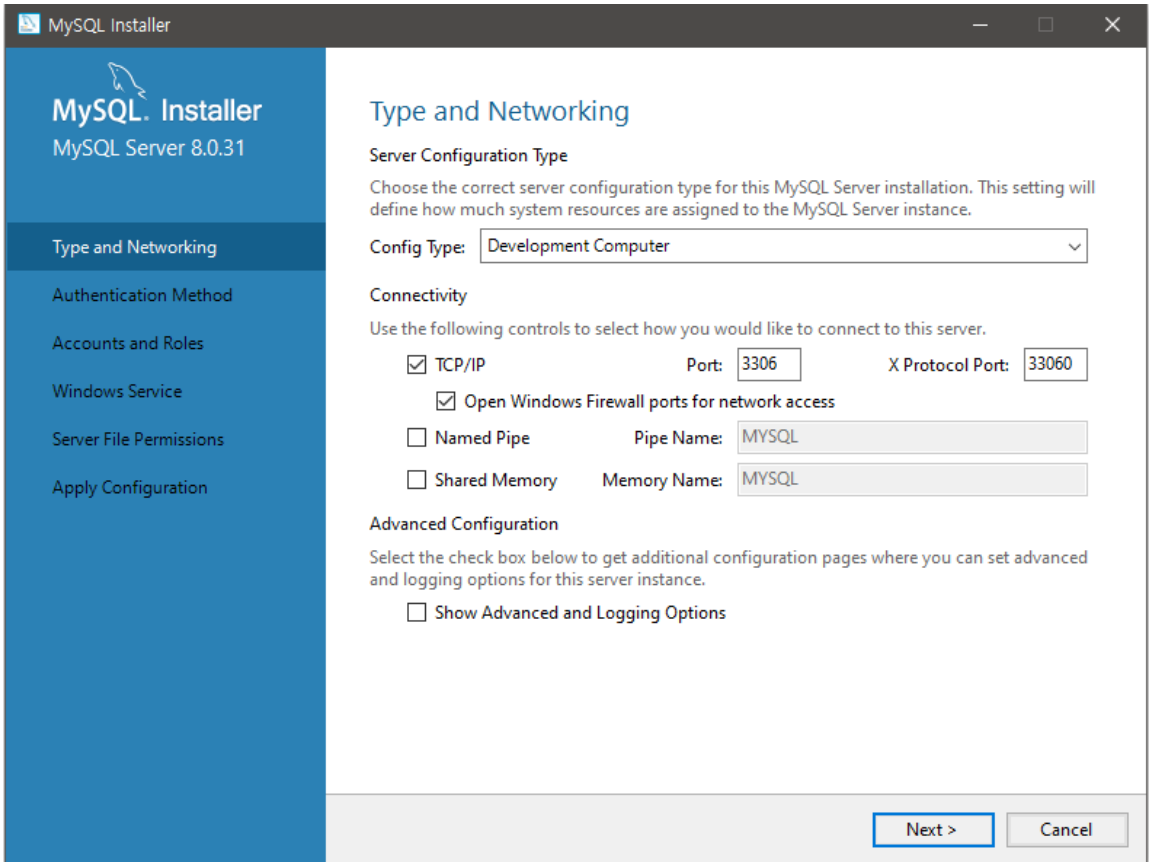

MySQL Installer
MySQL. Installer
Adding Community
Choosing a Setup Type
Select Products
Download
Installation
Installation Complete
Choosing a Setup Type
Please select the Setup Type that suits your use case.
Developer Default
Installs all products needed for MySQL development purposes.
Server only
Installs only the MySQL Server product.
Client only
Installs only the MySQL Client products, without a server.
Full
Installs all included MySQL products and features.
Custom
Manually select the products that should be installed on the system.
Setup Type Description
Allows you to select exactly which products you would like to install. This also allows to pick other server versions and architectures (depending on your OS).
Next >
Cancel


MySQL Servers
MySQL Server
MySQL Server 8.0
MySQL Server 8.0.31 - X64
Applications
MySQL Connectors
Documentation
MySQL Server 8.0.31 - X64

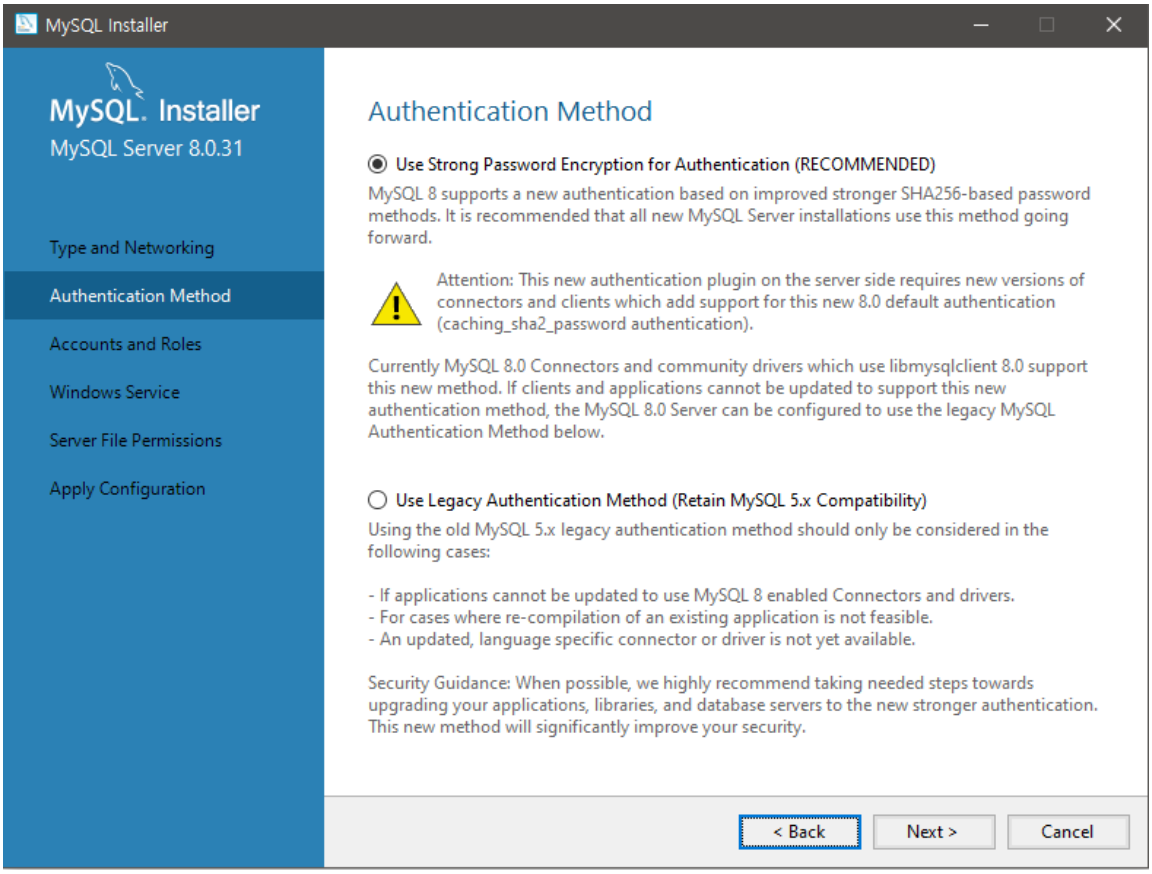


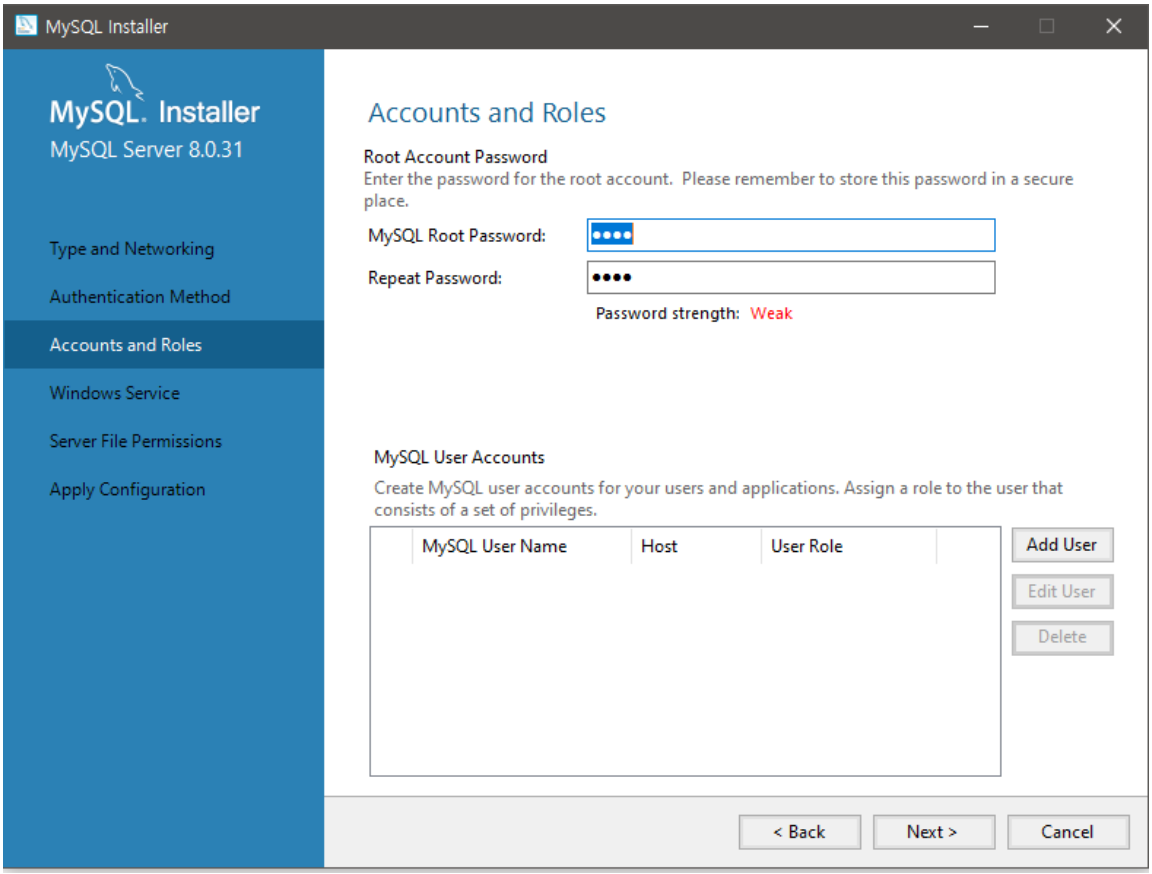
> 자바랑 MYSQL을 설정하는 커넥터 설치.


MySQL Installer
MySQL. Installer
Adding Community
Choosing a Setup Type
Select Products
Check Requirements
Installation
Product Configuration
Installation Complete
Check Requirements
The following products have failing requirements. MySQL Installer will attempt to resolve them automatically. Requirements marked as manual cannot be resolved automatically. Click on each item to try and resolve it manually.
For Product
Requirement
Status
MySQL Server 8.0.31
Microsoft Visual C++ 2019 Redistrib...
< Back
Execute
Next >
Cancel

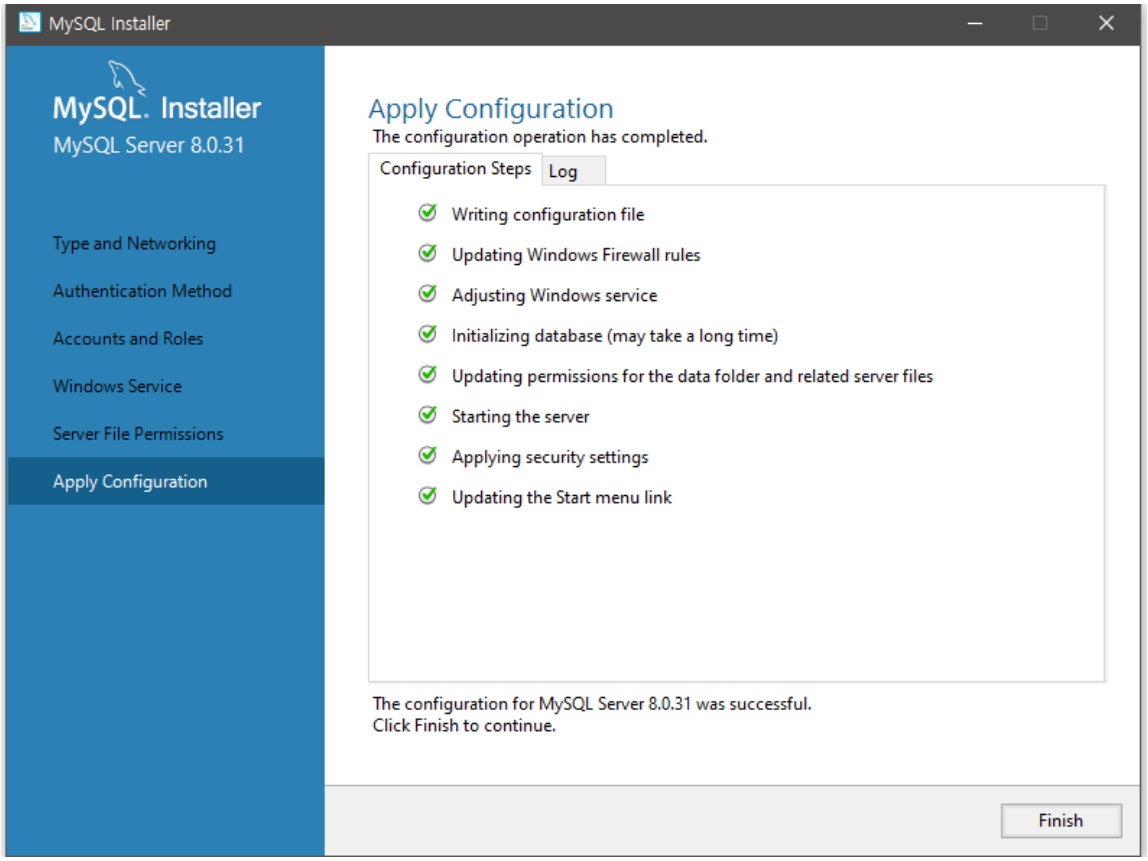

MySQL Installer
MySQL. Installer
MySQL Server 8.0.31
Type and Networking
Authentication Method
Accounts and Roles
Windows Service
Server File Permissions
Apply Configuration
Type and Networking
Server Configuration Type
Choose the correct server configuration type for this MySQL Server installation. This setting will define how much system resources are assigned to the MySQL Server instance.
Config Type: Development Computer
Connectivity
Use the following controls to select how you would like to connect to this server.
TCP/IP
Port: 3306
X Protocol Port: 33060
Open Windows Firewall ports for network access
Named Pipe
Pipe Name: MYSQL
Shared Memory
Memory Name: MYSQL
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

포트 번호는 반드시 3306 사용

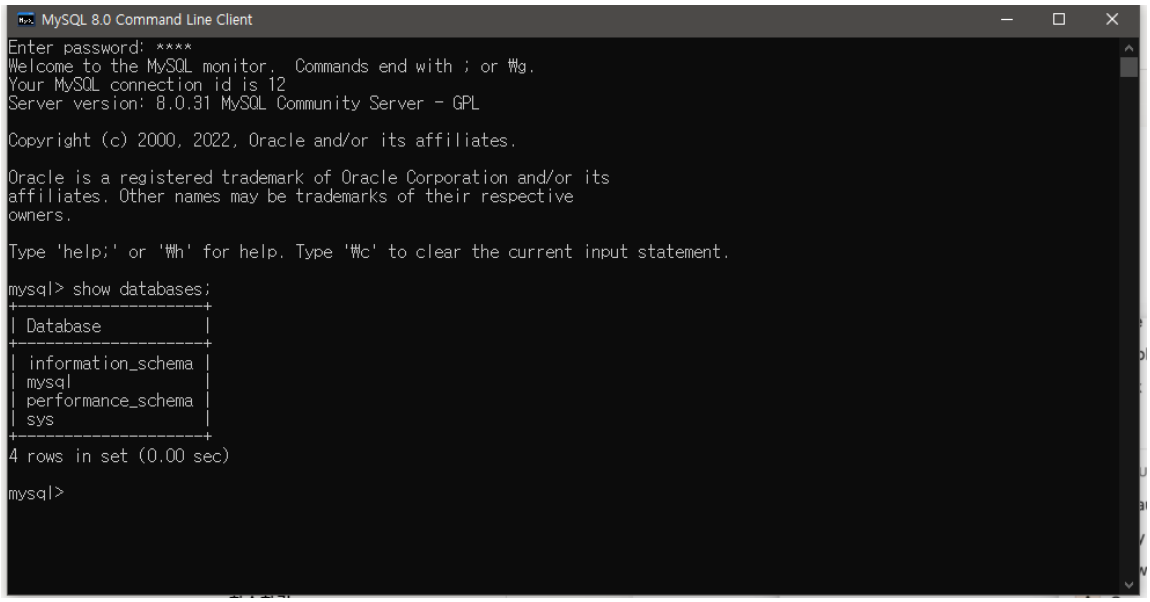




> 암호는 0000으로 통일

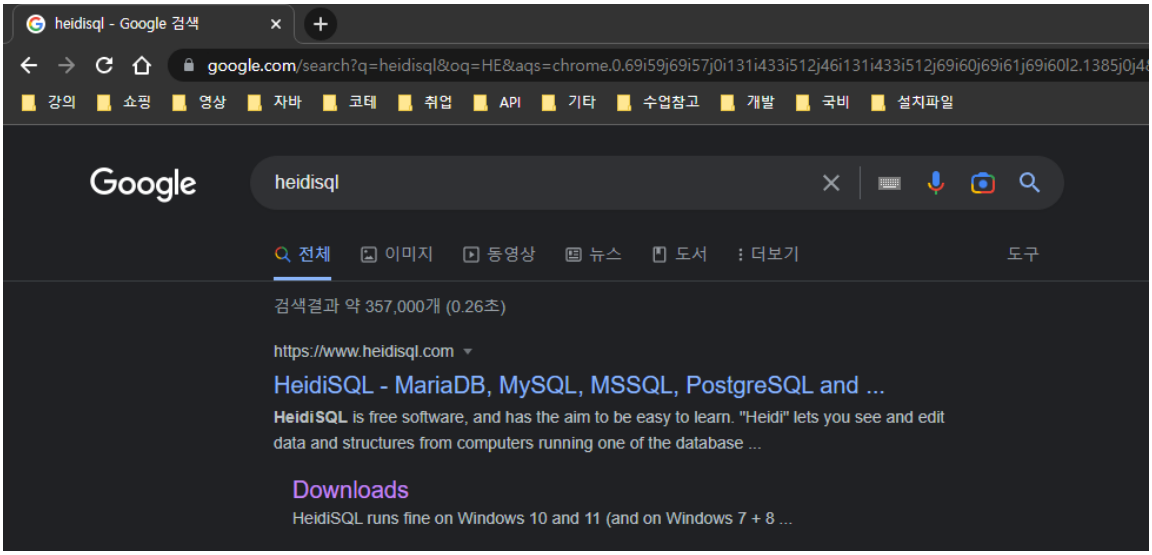

MySQL Installer
MySQL Installer
MySQL Server 8.0.31
Type and Networking
Authentication Method
Accounts and Roles
Windows Service
Server File Permissions
Apply Configuration
Apply Configuration
The configuration operation has completed.
Configuration Steps
Log
Writing configuration file
Updating Windows Firewall rules
Adjusting Windows service
Initializing database (may take a long time)
Updating permissions for the data folder and related server files
Starting the server
Applying security settings
Updating the Start menu link
The configuration for MySQL Server 8.0.31 was successful.
Click Finish to continue.
Finish

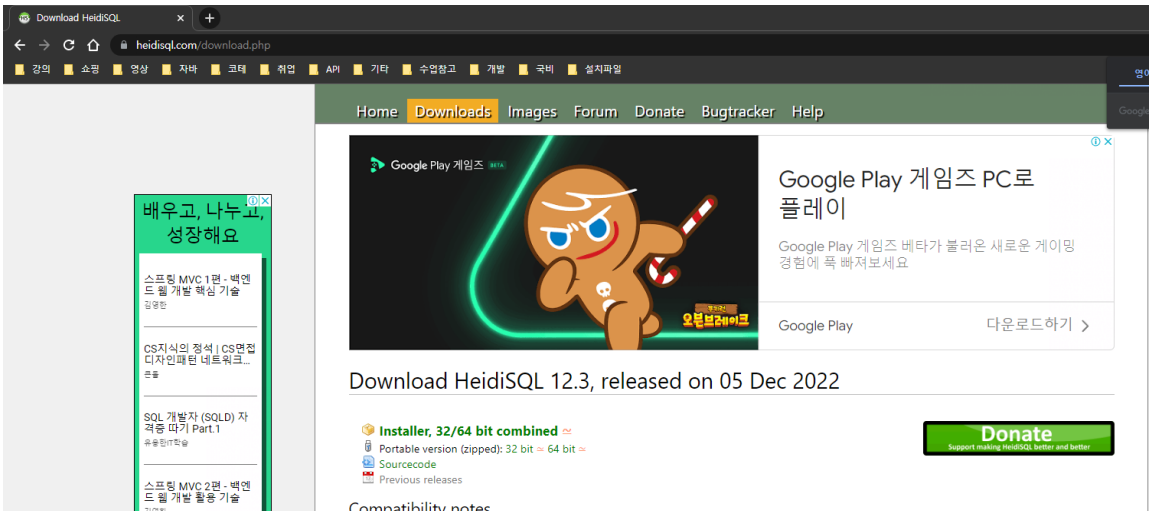
MYSQL 설정 완료 !


MySQL 8.0 Command Line Client
Enter password: ****
Welcome to the MySQL monitor. Commands end with ; or ₩g.
Your MySQL connection id is 12
Server version: 8.0.31 MySQL Community Server - GPL
Copyright (c) 2000, 2022, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '₩h' for help. Type '₩c' to clear the current input statement.
mysql> show databases;
+--------------------+
| Database |
+--------------------+
| information_schema |
| mysql |
| performance_schema |
| sys |
+--------------------+
4 rows in set (0.00 sec)
mysql>

MYSQL COMMEND LINE에서 암호 넣고 DB 연결 확인

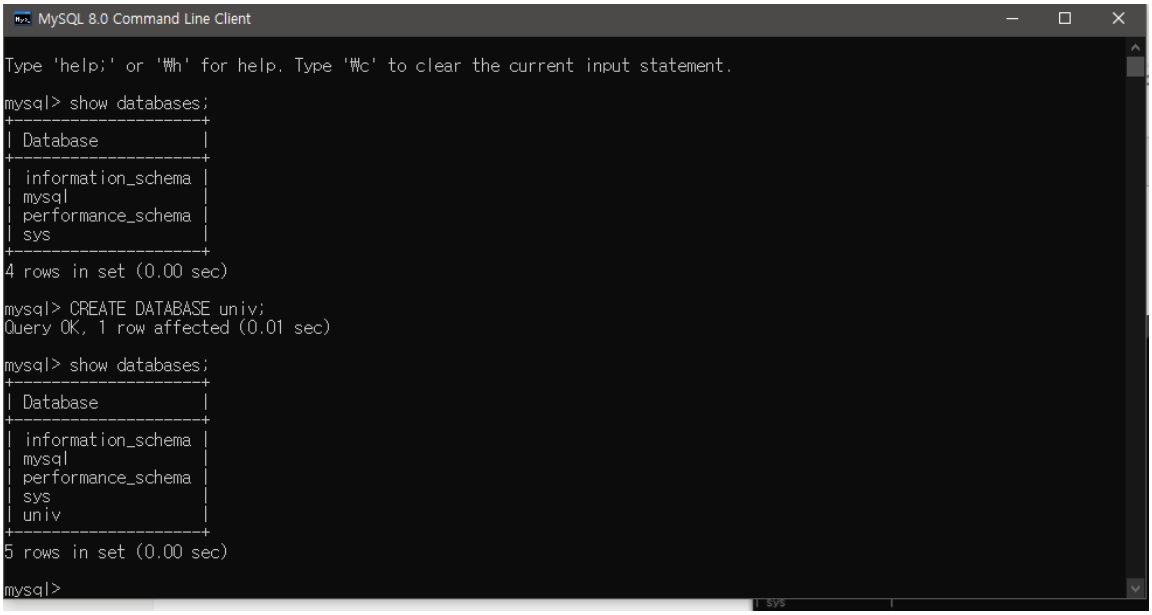
▼ heidiSQL 설치





- 오라클 삭제

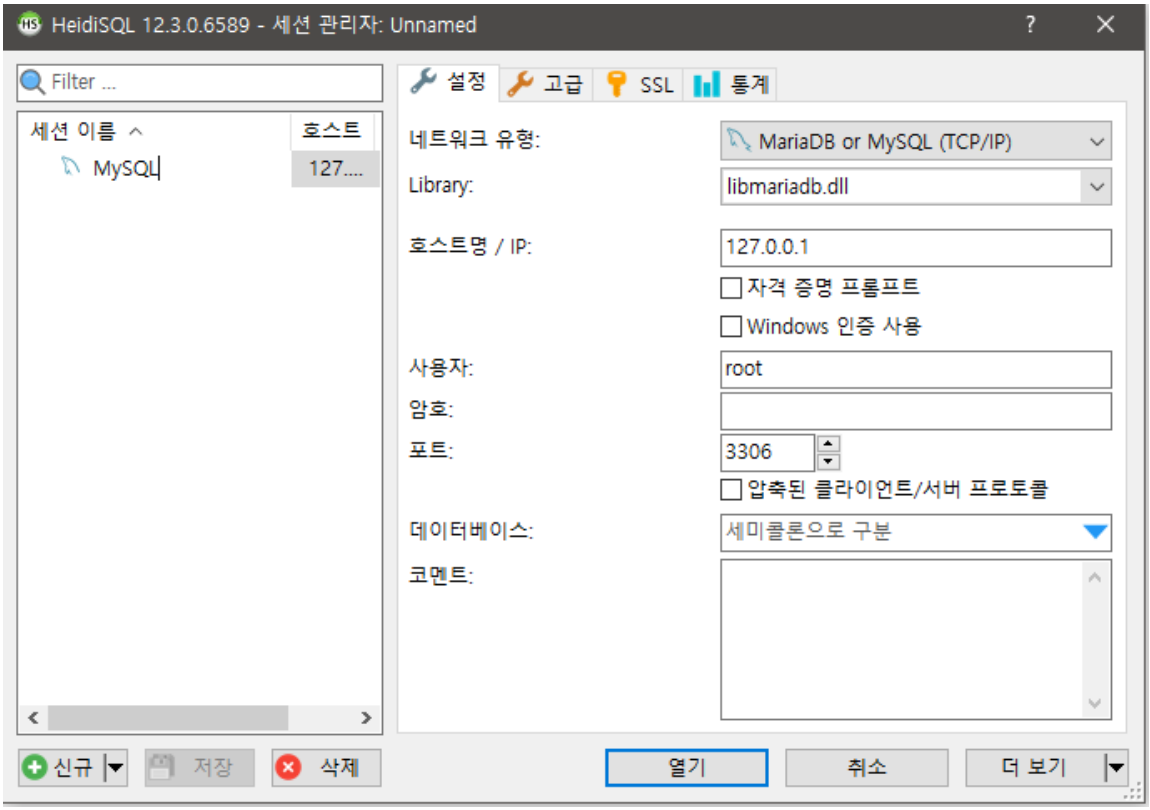
▼ MySQL Command Line으로 DB 생성



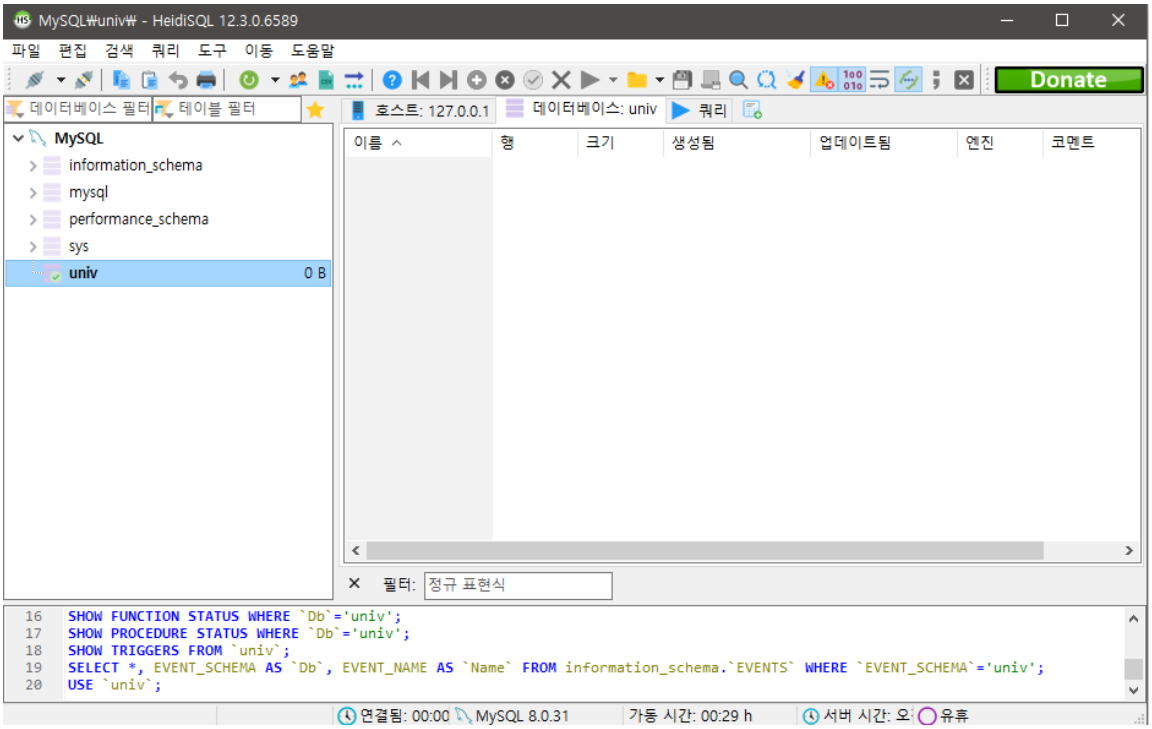
2교시 - Heidi 세팅 (10:00 ~ 10:50)

> **Heidi** : MySQL 와 MariaDB 를 사용하기 쉽게 도와준다.

▼ 세팅


HeidiSQL 12.3.0.6589 - 세션 관리자: Unnamed
Filter ...
세션 이름
호스트
MySQL
127....
설정
고급
SSL
통계
네트워크 유형:
MariaDB or MySQL (TCP/IP)
Library:
libmariadb.dll
호스트명 / IP:
127.0.0.1
자격 증명 프롬프트
Windows 인증 사용
사용자:
root
암호:
포트:
3306
압축된 클라이언트/서버 프로토콜
데이터베이스:
세미콜론으로 구분
코멘트:
신규
저장
삭제
열기
취소
더 보기

포트번호 3306 확인


MySQL@univ@ - HeidiSQL 12.3.0.6589
파일 편집 검색 쿼리 도구 이동 도움말
Donate
데이터베이스 필터
테이블 필터
호스트: 127.0.0.1
데이터베이스: univ
쿼리
MySQL
information_schema
mysql
performance_schema
sys
univ
0 B
이름
행
크기
생성됨
업데이트됨
엔진
코멘트
필터:
정규 표현식
SHOW FUNCTION STATUS WHERE `Db`='univ';
SHOW PROCEDURE STATUS WHERE `Db`='univ';
SHOW TRIGGERS FROM `univ`;
SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name` FROM information_schema.`EVENTS` WHERE `EVENT_SCHEMA`='univ';
USE `univ`;
연결됨: 00:00
MySQL 8.0.31
가동 시간: 00:29 h
서버 시간: 오
유휴

JSP로 데이터베이스를 사용하는 4가지 방법

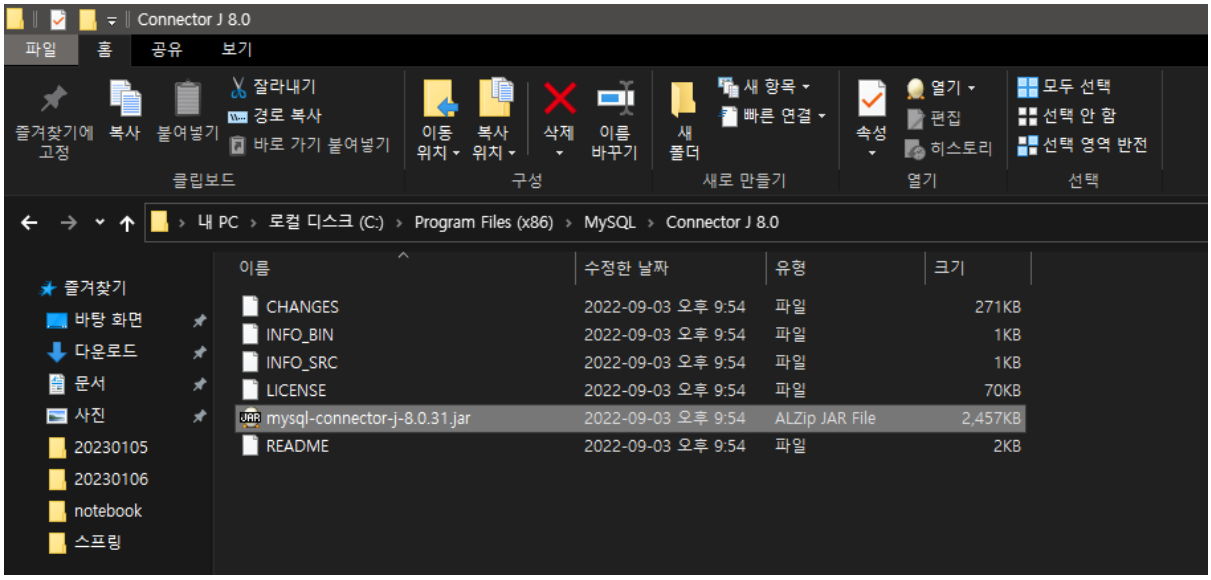
1. 각각의 JSP 파일에서 매번 직접 DB 사용
2. DB 연결 부분만 별도의 File로 구성
3. DB 연결 부분만 별도의 Class로 구성
4. Connection Pool

JDBC BASIC

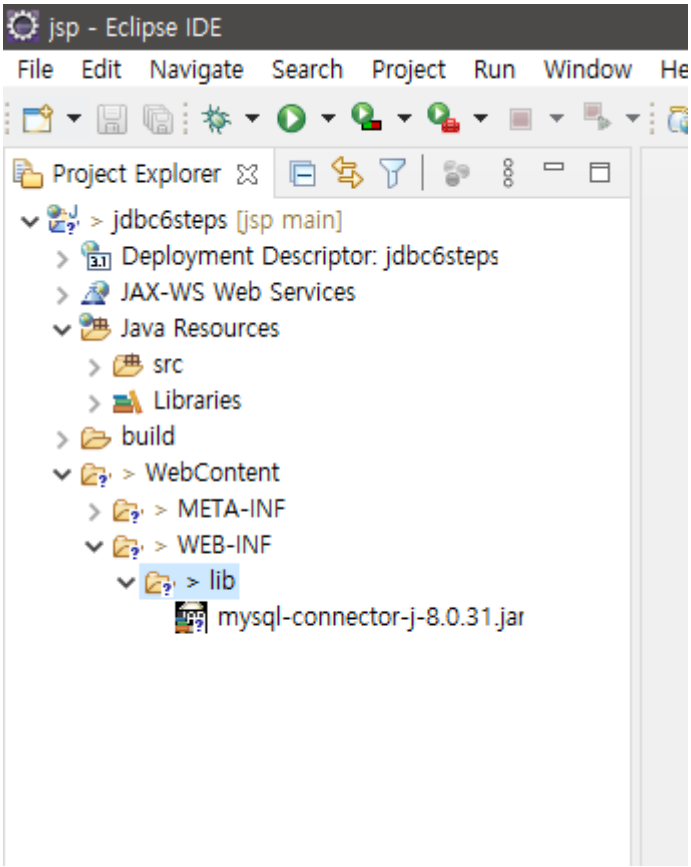
6 steps

- Step 1 : import SQL Packages
- Step 2 : load JDBC Driver
- Step 3 : create Connection Object
- Step 4 : create Statement Object
- Step 5 : excute SQL Query
- Step 6 : close Connection (java 9부터 생략 가능)

- JSP 깃허브 추가 및 워크스페이스 JSP로 변경



> MYSQL 커넥터 경로로 들어가 복사하기.



> WEB-INF 밑에 lib 밑에 커넥터를 넣어줘야 연결이 가능.

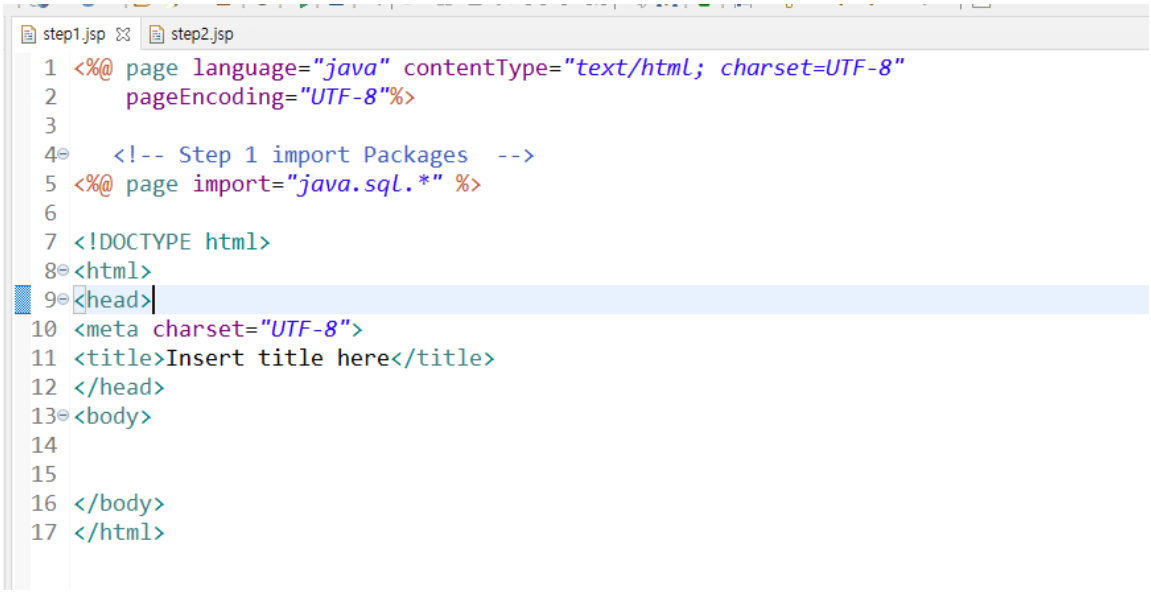
3교시 - 데이터베이스 연결 및 생성 (11:00 ~ 11:50)

- Dynamic Web Project는 JSP, Servlet을 만들때 사용
- Spring Legacy Project는 Spring을 사용할 때

JSP에서 데이터베이스 생성

▼ JDBC Basic

Step 1 - import SQL Packages



Step 2 - load JDBC Driver

```
<body>
<%
        // Step 2 load JDBC Driver
    try{
    Class.forName("com.mysql.jdbc.Driver"); // JDBC 드라이버 가져오기
    } catch(ClassNotFoundException e){
        out.print("JDBC Driver loading error<br>" + e.getMessage());
    }
    // com.mysql.jdbc.Driver    mySQL용
    // org.maria.jdbc.Driver    maria용

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

Step 3 - create Connection Object

```
    //  Step 3 create Connection Object

    Connection conn = null;

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

```

Step 4 - create Statement Object

```
    // Step 4 create Statement Object

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

```

Step 5 - excute SQL Query

```
    // Step 5 excute SQL Query

    pstmt.executeUpdate();

```

Step 6 - close Connection (java 9부터 생략 가능)

```
43      // Step 6 close Connection
44
45      pstmt.close();
46      conn.close();
47
```

4교시 - (12:00 ~ 12:50)

- Step6.jsp 복사 → DBCreate.jsp 및 DBDrop.jsp생성

실행하면 에러 발생

```
31
32      // Step 4 create Statement Object
33
34      //테이블을 JSP로 생성
35      //테이블명 student (univ db 이용)
36      //hakbun, name, dept, addr
37
38      String sql = "CREATE TABLE student("
39                  + "hakbun   varchar(10), "
40                  + "name varchar(10), "
41                  + "dept varchar(10), "
42                  + "addr varchar(10), "
43                  + "primary key(hakbun)";
44      PreparedStatement pstmt = conn.prepareStatement(sql);    //구문 생성
45
```

에러페이지

HTTP 상태 500 – 내부 서버 오류

타입 예외 보고

메시지 행 [49]에서 [/TBCreate.jsp]을(를) 처리하는 중 예외 발생

설명 서버가, 해당 요청을 충족시키지 못하게 하는 예기치 않은 조건을 맞닥뜨렸습니다.

예외

```
org.apache.jasper.JasperException: 행 [49]에서 [/TBCreate.jsp]을(를) 처리하는 중 예외 발생

46:
47:     // Step 5 excute SQL Query
48:
49:     pstmt.executeUpdate();
50:
51:     // Step 6 close Connection
52:


Stacktrace:
        org.apache.jasper.servlet.JspServletWrapper.handleJspException(JspServletWrapper.java:605)
        org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:488)
        org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:383)
        org.apache.jasper.servlet.JspServlet.service(JspServlet.java:331)
        javax.servlet.http.HttpServlet.service(HttpServlet.java:765)
        org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52)
```

근본 원인 (root cause)

```
javax.servlet.ServletException: java.sql.SQLException: No database selected
        org.apache.jasper.runtime.PageContextImpl.doHandlePageException(PageContextImpl.java:907)
        org.apache.jasper.runtime.PageContextImpl.handlePageException(PageContextImpl.java:833)
        org.apache.jsp.TBCreate_jsp._jspService(TBCreate_jsp.java:183)
        org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:70)
        javax.servlet.http.HttpServlet.service(HttpServlet.java:765)
        org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:465)
        org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:383)
        org.apache.jasper.servlet.JspServlet.service(JspServlet.java:331)
        javax.servlet.http.HttpServlet.service(HttpServlet.java:765)
        org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52)
```

근본 원인 (root cause)

```
java.sql.SQLException: No database selected
        com.mysql.cj.jdbc.exceptions.SQLError.createSQLException(SQLError.java:129)
        com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:122)
        com.mysql.cj.jdbc.ClientPreparedStatement.executeInternal(ClientPreparedStatement.java:916)
        com.mysql.cj.jdbc.ClientPreparedStatement.executeUpdateInternal(ClientPreparedStatement.java:1061)
        com.mysql.cj.jdbc.ClientPreparedStatement.executeUpdateInternal(ClientPreparedStatement.java:1009)
        com.mysql.cj.jdbc.ClientPreparedStatement.executeLargeUpdate(ClientPreparedStatement.java:1320)
        com.mysql.cj.jdbc.ClientPreparedStatement.executeUpdate(ClientPreparedStatement.java:994)
        org.apache.jsp.TBCreate_jsp._jspService(TBCreate_jsp.java:158)
        org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:70)
        javax.servlet.http.HttpServlet.service(HttpServlet.java:765)
        org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:465)
```

해결법

```
24      try{
25          conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/univ",
26                                                  "root", "0000");     // JDBC 드라이버 연결
27      } catch(SQLException e){
28          out.print("Connection Object error<br>" + e.getMessage());
29      }
30
31
32      // Step 4 create Statement Object
33
34      //테이블을 JSP로 생성
35      //테이블명 student (univ db 이용)
36      //hakbun, name, dept, addr
37
38      String sql = "CREATE TABLE student("
39                  + "hakbun   varchar(10), "
40                  + "name varchar(10), "
41                  + "dept varchar(10), "
42                  + "addr varchar(10), "
43                  + "primary key(hakbun))";
44      PreparedStatement pstmt = conn.prepareStatement(sql);   //구문 생성
45
```

> **원인** : 25번째줄 conn 에서 db접근을 할때 univ 경로 설정을 안해줘서.

데이터 넣기

TBInsert.jsp | Insert title here

```
19
20      //  Step 3 create Connection Object
21
22      Connection conn = null;
23
24      try{
25          conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/univ",
26                                                  "root", "0000");     // JDBC 드라이브
27      } catch(SQLException e){
28          out.print("Connection Object error<br>" + e.getMessage());
29      }
30
31
32      // Step 4 create Statement Object
33
34      //테이블을 JSP로 생성
35      //테이블명 student (univ db 이용)
36      //hakbun, name, dept, addr
37
38      String sql = "INSERT INTO student VALUES('1111', 'Son', '축구부', '서울')";
39
40      PreparedStatement pstmt = conn.prepareStatement(sql);   //구문 생성
41
42
```

```
    // Step 4 create Statement Object

    //테이블을 JSP로 생성
    //테이블명 student (univ db 이용)
    //hakbun, name, dept, addr

    String hakbun = "2222";
    String name = "Kim";
    String dept = "컴공";
    String addr = "인천";

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


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

    // Step 5 excute SQL Query

    pstmt.executeUpdate();

```

5교시 - (14:00 ~ 14:50)

TBForm.jsp 생성

```
<%@ 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="TBInsert.jsp" method="post">
    학번 <input type="text" name="hakbun"><br>
    이름 <input type="text" name="name"><br>
    부서 <input type="text" name="dept"><br>
    주소 <input type="text" name="addr"><br>
    <button>제출</button>
  </form>
</body>
</html>
```

TBInsert.jsp 수정

```
// Step 4 create Statement Object

  //테이블을 JSP로 생성
  //테이블명 student (univ db 이용)
  //hakbun, name, dept, addr

  /* String hakbun = "2222";
  String name = "Kim";
  String dept = "컴공";
  String addr = "인천"; */



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


  PreparedStatement pstmt = conn.prepareStatement(sql); //구문 생성
  pstmt.setString(1, request.getParameter("hakbun"));
  pstmt.setString(2, request.getParameter("name"));
  pstmt.setString(3, request.getParameter("dept"));
  pstmt.setString(4, request.getParameter("addr"));
```

TBList.jsp 생성

```
// Step 4 Statement Object


  String sql = "SELECT * FROM student";


  PreparedStatement pstmt = conn.prepareStatement(sql); //구문 생성




  // Step 5 excute SQL Query

  ResultSet rs = pstmt.executeQuery();            <---- 리턴되는 결과값 가져오기
```

```
// Step 5 excute SQL Query

  ResultSet rs = pstmt.executeQuery();

  while(rs.next()){


%>
  <!-- 화면에 뿌려 주기  -->
  <%= rs.getString("hakbun") %>|
  <%= rs.getString("name") %>|
  <%= rs.getString("dept") %>|
  <%= rs.getString("addr") %><br>

<%
  }
  // Step 6 close Connection
  rs.close();
  pstmt.close();
  conn.close();
%>
```

login / logout session

1. 아이디와 비밀번호를 모두 정상적으로 입력한 경우 → 로그인 성공
2. 비밀번호가 틀린 경우 → 로그인 실패 (다시 시도하세요)
3. 아이디가 디비에 존재하지 않는 경우 → 로그인 실패 (회원 가입 페이지로 리다이렉트)

6교시 - (15:00 ~ 15:50)

TBInsert.jsp | TBForm.jsp | TBList.jsp

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3        <!-- Step 1 import Packages  -->
4 <%@ page import="java.sql.*" %>
5
6 <% // 전송 받는 데이터 한글 처리
7     request.setCharacterEncoding("UTF-8");
8 %>
9
```

💡 데이터 받는 JSP에서 위쪽에 소스를 추가하여 한글 인코딩 처리

▼ loginForm.jsp 생성

```
<form action="loginCheck.jsp">
  학번 : <input type="text" name="hakbun">
  <button>로그인</button>
  </form>
```

▼ loginCheck.jsp 생성

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
       <!-- Step 1 import 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"); // JDBC 드라이버 가져오기
  } catch(ClassNotFoundException e){
    out.print("JDBC Driver loading error<br>" + e.getMessage());
  }

  //  Step 3 create Connection Object

  Connection conn = null;

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


  // Step 4 Statement Object

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

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


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


  // Step 5 excute SQL Query

  ResultSet rs = pstmt.executeQuery();

  // DB에서 학번에 해당하는 자료가 있어서 반환되었는지 확인
  if(!rs.isBeforeFirst()){
    out.print("<script>alert('해당 학번은 존재하지 않습니다.');"
          + "history.back();" //뒤로 돌아가기
          + "</script>");
    return;   ---->>> return을 안쓰면 밑에도 실행이 되어 오류 발생
  }

  rs.next();
  String dbHakbun = rs.getString("hakbun");
  String dbName = rs.getString("name");

  // dbHakbun과 hakbun 값이 같으면 세션 생성
  // "hakbun" dbHakbun  "name"  dbName
  if(dbHakbun.equals(hakbun)){  //hakbun과 dbHakbun이 String이기 때문에 문자열 비교은 .e
quals를 써야함.
    session.setAttribute("hakbun", dbHakbun);
    session.setAttribute("name", dbName);

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

  // Step 6 close Connection
  rs.close();
  pstmt.close();
  conn.close();
%>

</body>
</html>
```

7교시 - dbconnect, close 모듈화 (16:00 ~ 16:50)

중복된 dbconnect, close File로 분리

▼ DBCreate2.jsp 생성

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

   <!-- Step 1 import Packages  -->
<%@ page import="java.sql.*" %>

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

  // Step 4 create Statement Object

  PreparedStatement pstmt = conn.prepareStatement("CREATE DATABASE test2"); //구문
 생성

  // Step 5 excute SQL Query

  pstmt.executeUpdate();


%>
<%@ include file="dbclose.inc" %>
</body>
</html>
```

▼ dbconnect.inc 생성

```
<%
    // Step 2 load JDBC Driver
  try{
  Class.forName("com.mysql.jdbc.Driver"); // JDBC 드라이버 가져오기
  } catch(ClassNotFoundException e){
    out.print("JDBC Driver loading error<br>" + e.getMessage());
  }

  //  Step 3 create Connection Object

  Connection conn = null;

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

▼ dbclose.inc 생성

```
<%
// Step 6 close Connection

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

▼ DBDrop2.jsp 생성

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

   <!-- Step 1 import Packages  -->
<%@ page import="java.sql.*" %>

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

  // Step 4 create Statement Object

  PreparedStatement pstmt = conn.prepareStatement("DROP DATABASE test3"); //구문 생
성

  // Step 5 excute SQL Query

  pstmt.executeUpdate();

%>
<%@ include file="dbclose.inc" %>
</body>
</html>
```

중복된 dbconnect, close Class로 분리

▼ DBConnClose.java 생성

```
package jdbc6steps;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class DBConnClose {
  // DB 연결 공통부분 메서드
  public static Connection getConnection() {
    // Step 2 load JDBC Driver
    try{
    Class.forName("com.mysql.jdbc.Driver"); // JDBC 드라이버 가져오기
    } catch(ClassNotFoundException e){
      System.out.print("JDBC Driver loading error<br>" + e.getMessage());
    }

    //  Step 3 create Connection Object

    Connection conn = null;

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

  // DB 해제 공통부분 메서드
  public static void closeConnection(PreparedStatement pstmt, Connection conn) {

    // Step 6 close Connection
    try {
      pstmt.close();
      conn.close();
    } catch(SQLException e) {
      System.out.println("Error : " + e.getMessage());
    }


  }


}
```

▼ DBCreate3.jsp 생성

```
<%@page import="jdbc6steps.DBConnClose"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>

   <!-- Step 1 import Packages  -->
```

```
<%@ page import="java.sql.*" %>
<%@ page import="jdbc6steps.*" %>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<%
  Connection conn = DBConnClose.getConnection();

  // Step 4 create Statement Object

  PreparedStatement pstmt = conn.prepareStatement("CREATE DATABASE test3"); //구문
 생성

  // Step 5 excute SQL Query

  pstmt.executeUpdate();

  DBConnClose.closeConnection(pstmt, conn); // static으로 메소드를 선언해서 바로 클래스 이
름을 써서 사용.


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

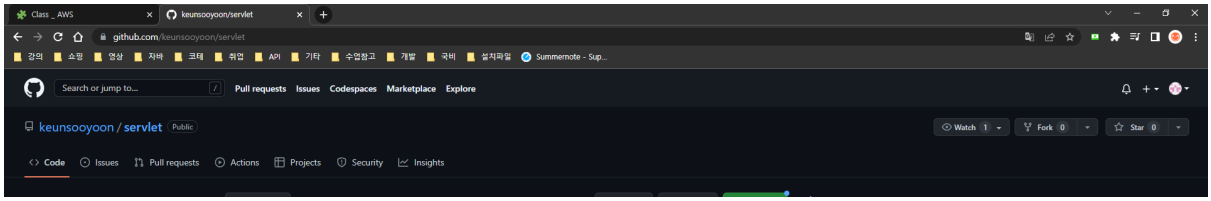
▼ DBDrop3.jsp 생성

```
<%@page import="jdbc6steps.DBConnClose"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>

   <!-- Step 1 import Packages  -->
<%@ page import="java.sql.*" %>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<%
  Connection conn = DBConnClose.getConnection();

  // Step 4 create Statement Object

  PreparedStatement pstmt = conn.prepareStatement("DROP DATABASE test3"); //구문 생
성

  // Step 5 excute SQL Query
```

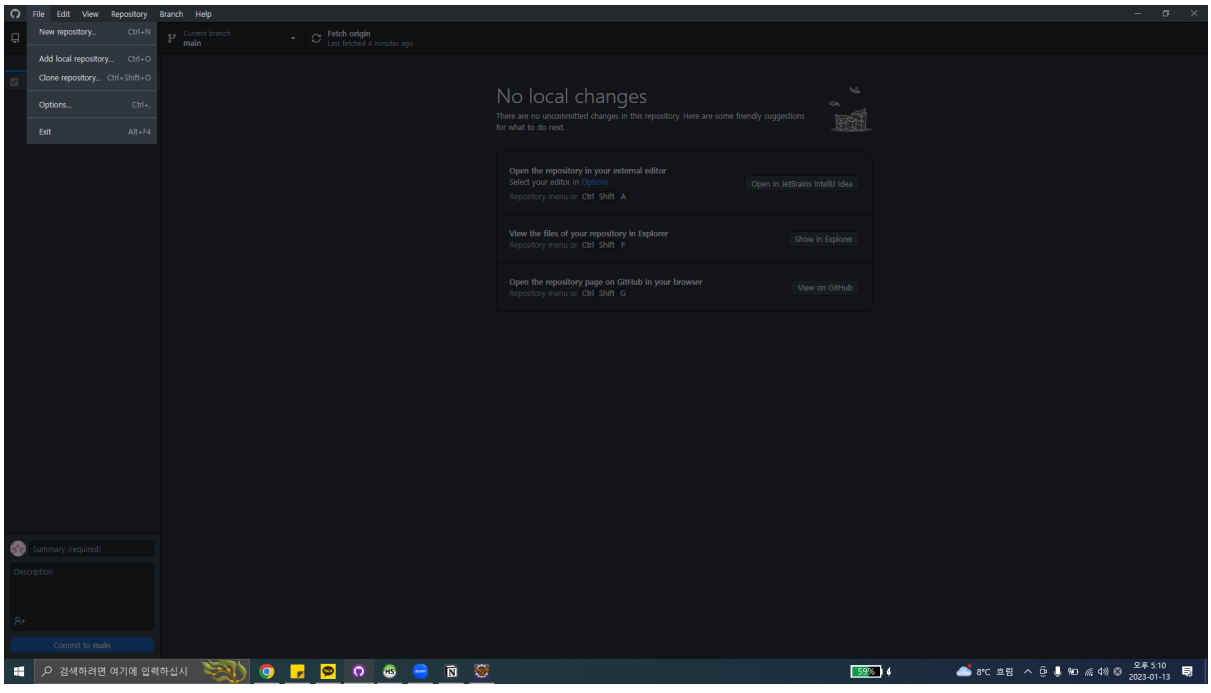
```
  pstmt.executeUpdate();

  DBConnClose.closeConnection(pstmt, conn);

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

8교시 - 정리 (17:00 ~ 17:50)

남의 깃허브 가져오기



- 오른쪽 상단에 포크 눌러서 가져오기



- File → Clone repository 클릭