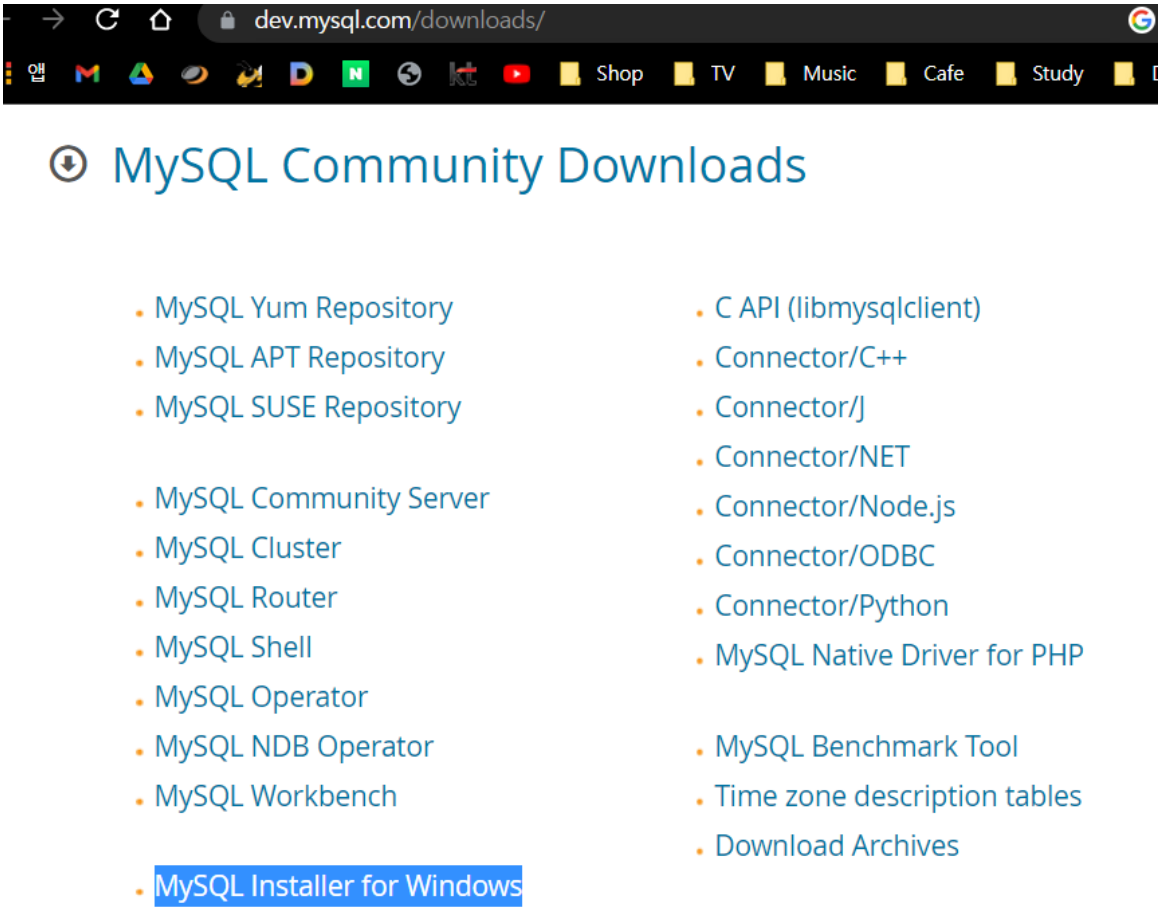
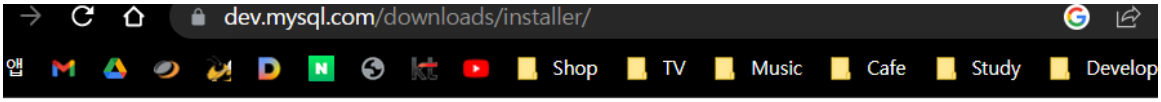
Class 230113

MySQL 설치

▼ 설치 과정





MySQL Community Downloads

MySQL Installer

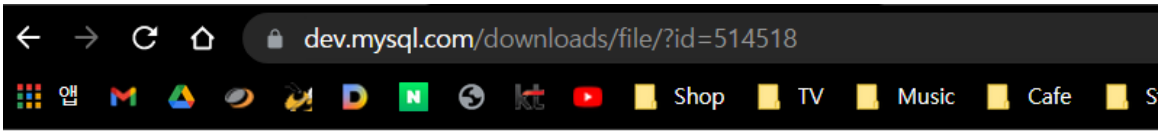
General Availability (GA) Releases | Archives

MySQL Installer 8.0.31

Select Operating System: Looking for previous GA versions?

Microsoft Windows

Windows (x86, 32-bit), MSI Installer	8.0.31	5.5M	**Download**
(mysql-installer-web-community-8.0.31.0.msi)	MD5: 7a83203e24f873b49fa2df2f1a58eca6 | Signature		
Windows (x86, 32-bit), MSI Installer	8.0.31	431.7M	**Download**
(mysql-installer-community-8.0.31.0.msi)	MD5: ef57176fcb01f01f4e87dbba9b87ac6f | Signature		



MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

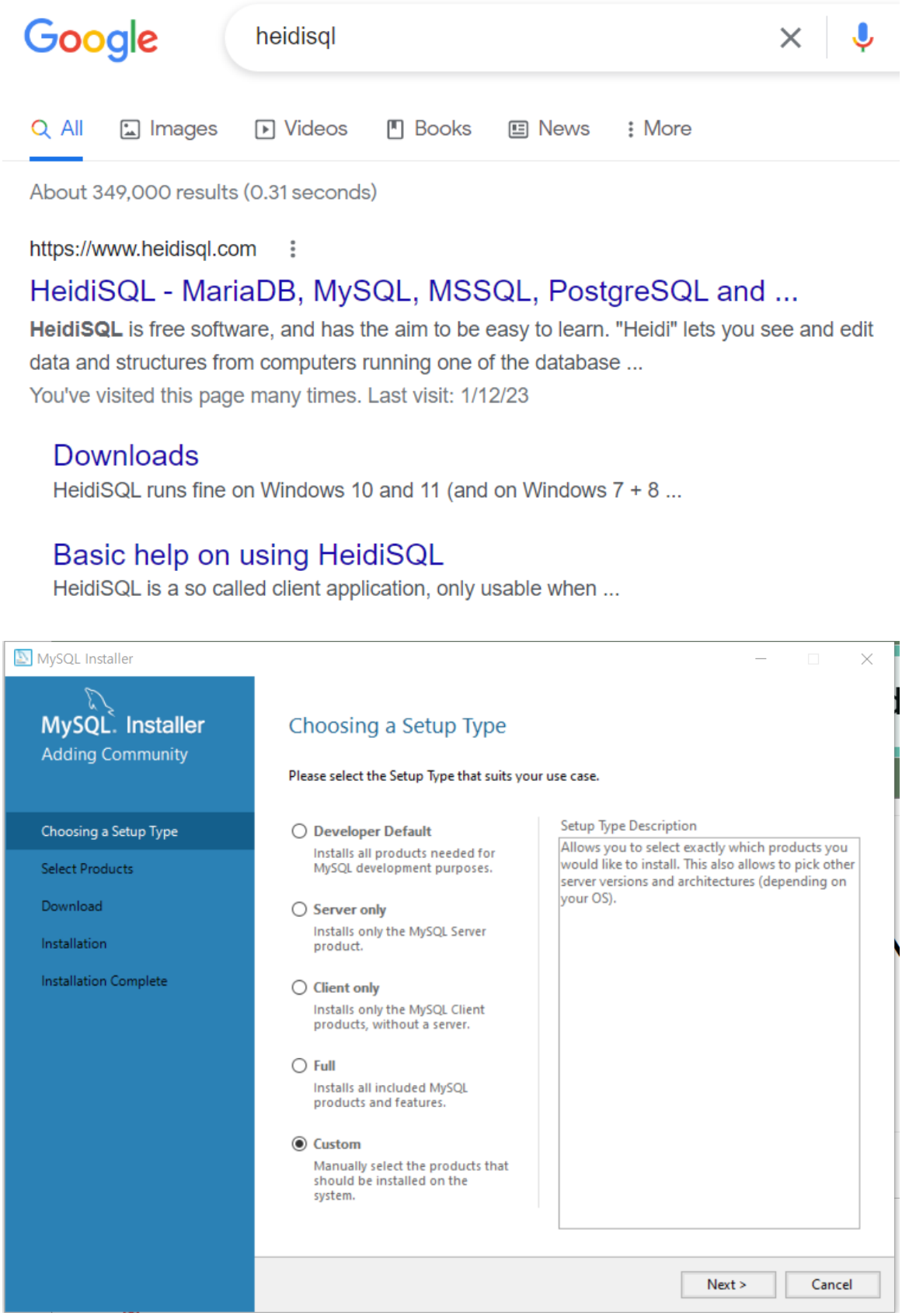
- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

Login »
using my Oracle Web account

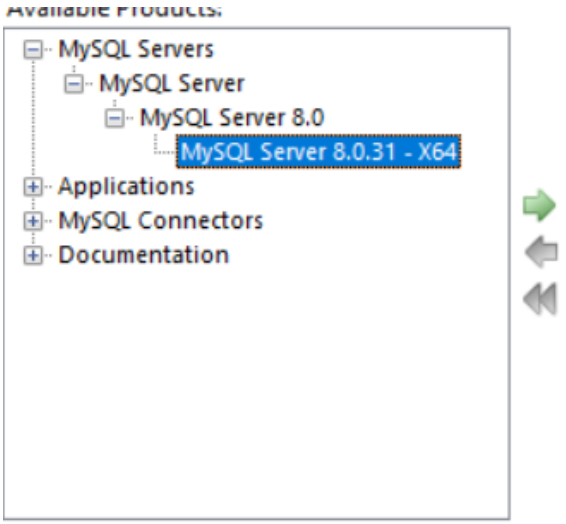
Sign Up »
for an Oracle Web account

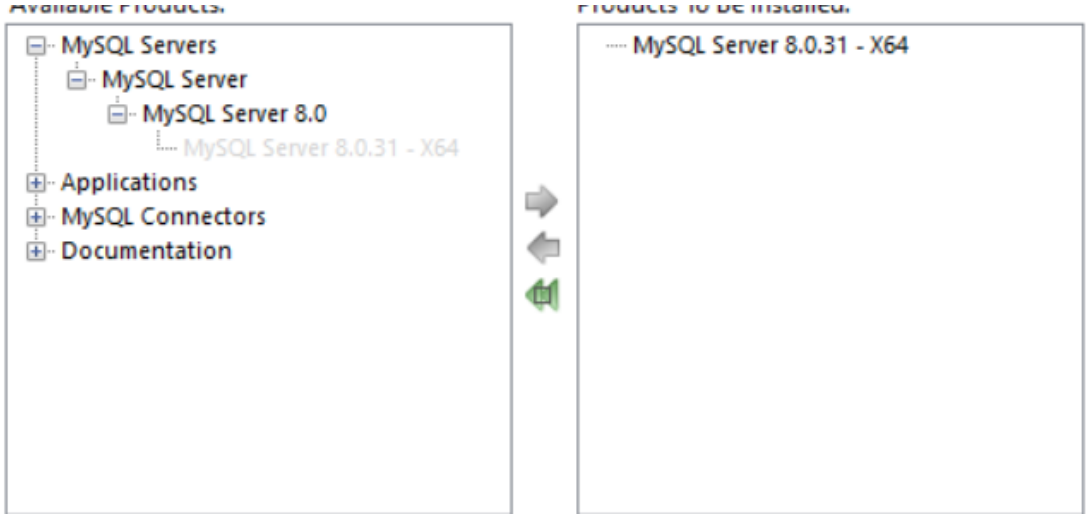
MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.

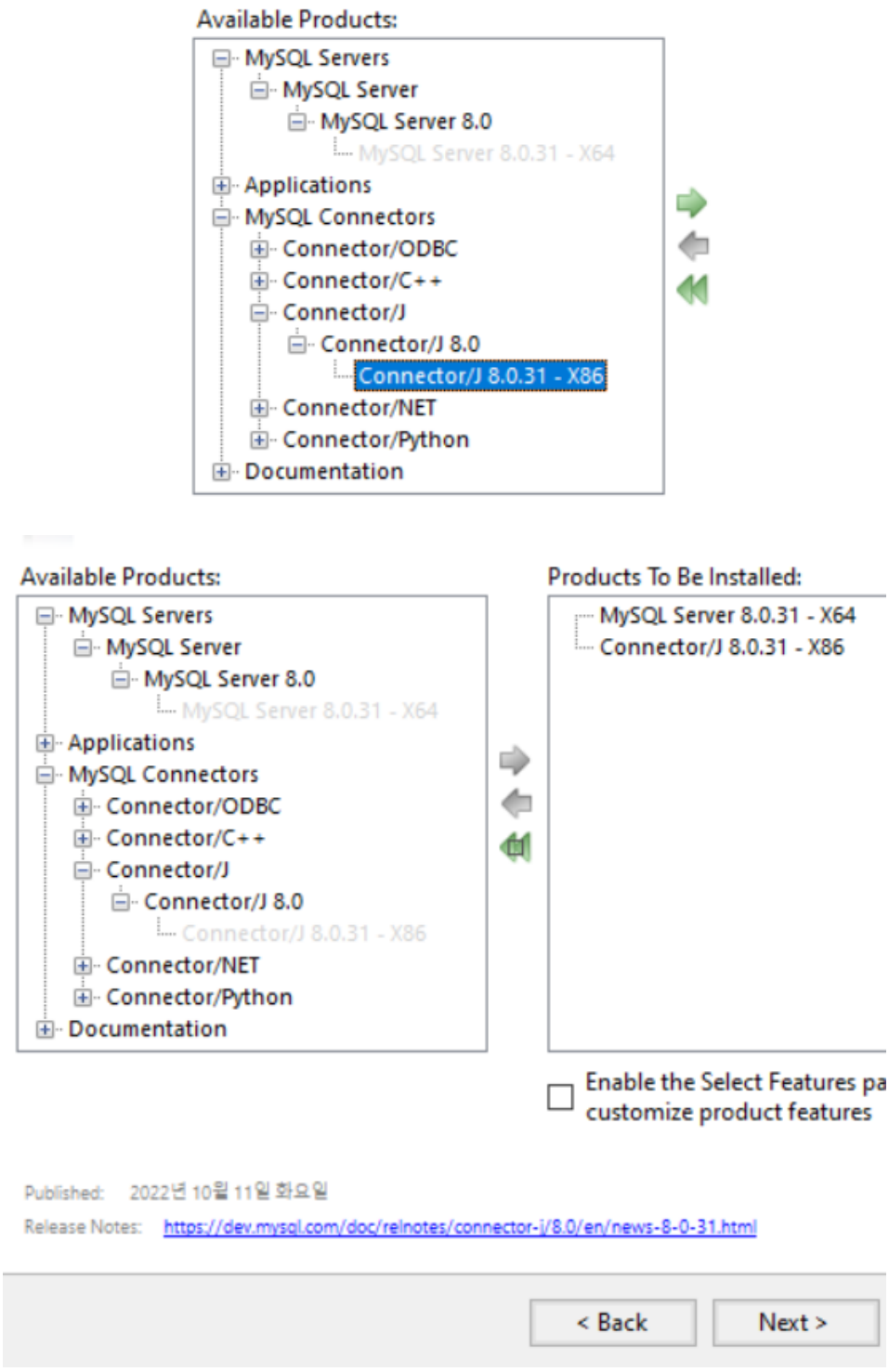
No thanks, just start my download.

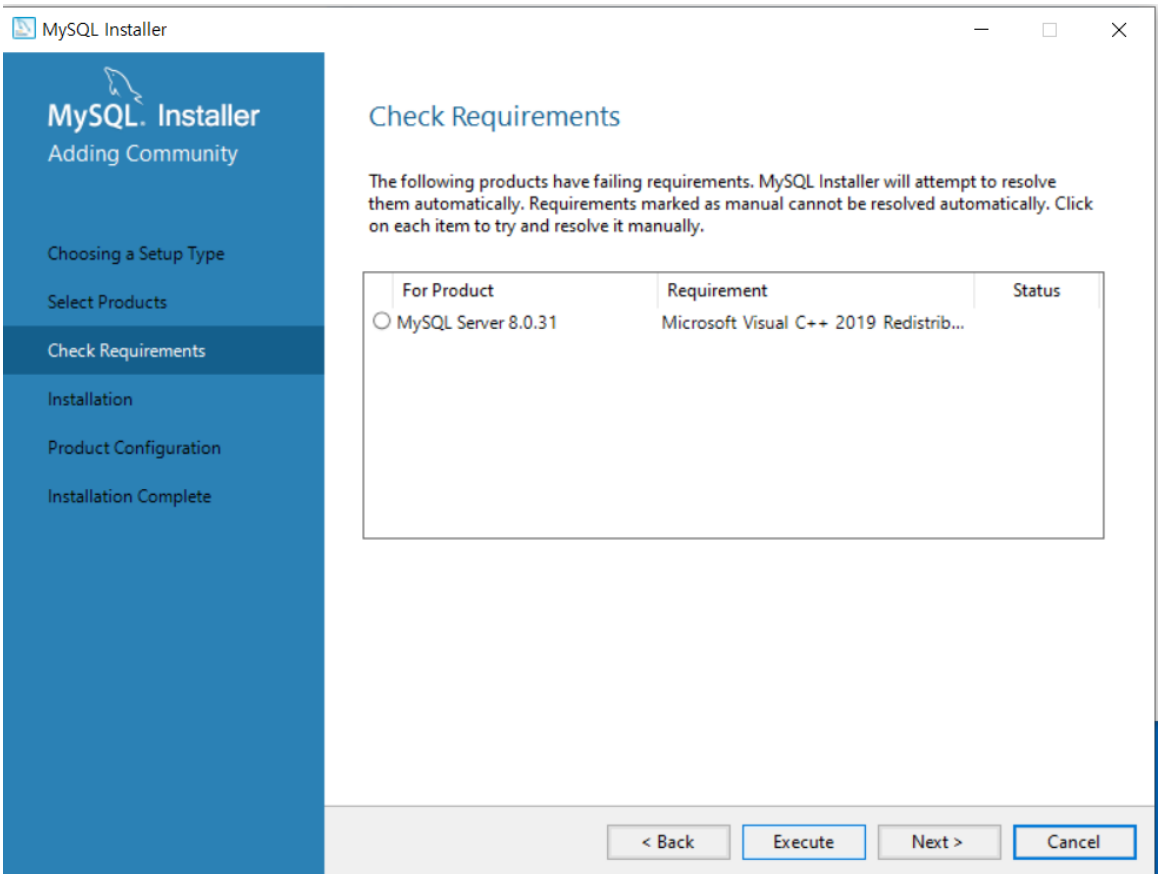

Google
heidisql
All
Images
Videos
Books
News
More
About 349,000 results (0.31 seconds)
https://www.heidisql.com
HeidiSQL - MariaDB, MySQL, MSSQL, PostgreSQL and ...
HeidiSQL is free software, and has the aim to be easy to learn. "Heidi" lets you see and edit data and structures from computers running one of the database ...
You've visited this page many times. Last visit: 1/12/23
Downloads
HeidiSQL runs fine on Windows 10 and 11 (and on Windows 7 + 8 ...
Basic help on using HeidiSQL
HeidiSQL is a so called client application, only usable when ...
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


Custom
Manually select the products that should be installed on the system.
Next >

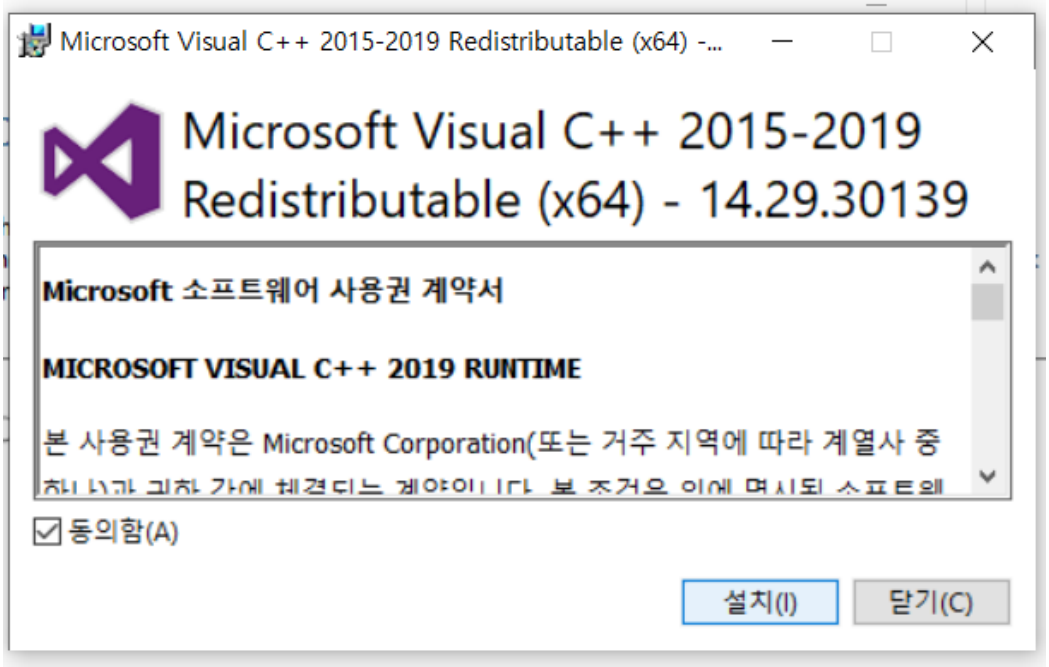

MySQL Servers
MySQL Server
MySQL Server 8.0
MySQL Server 8.0.31 - X64
Applications
MySQL Connectors
Documentation


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


Available Products:
MySQL Servers
MySQL Server
MySQL Server 8.0
MySQL Server 8.0.31 - X64
Applications
MySQL Connectors
Connector/ODBC
Connector/C++
Connector/J
Connector/J 8.0
Connector/J 8.0.31 - X86
Connector/NET
Connector/Python
Documentation
Available Products:
MySQL Servers
MySQL Server
MySQL Server 8.0
MySQL Server 8.0.31 - X64
Applications
MySQL Connectors
Connector/ODBC
Connector/C++
Connector/J
Connector/J 8.0
Connector/J 8.0.31 - X86
Connector/NET
Connector/Python
Documentation
Products To Be Installed:
MySQL Server 8.0.31 - X64
Connector/J 8.0.31 - X86
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 >



Execute Click !!!



설치 클릭 !!!

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...	INSTL DONE

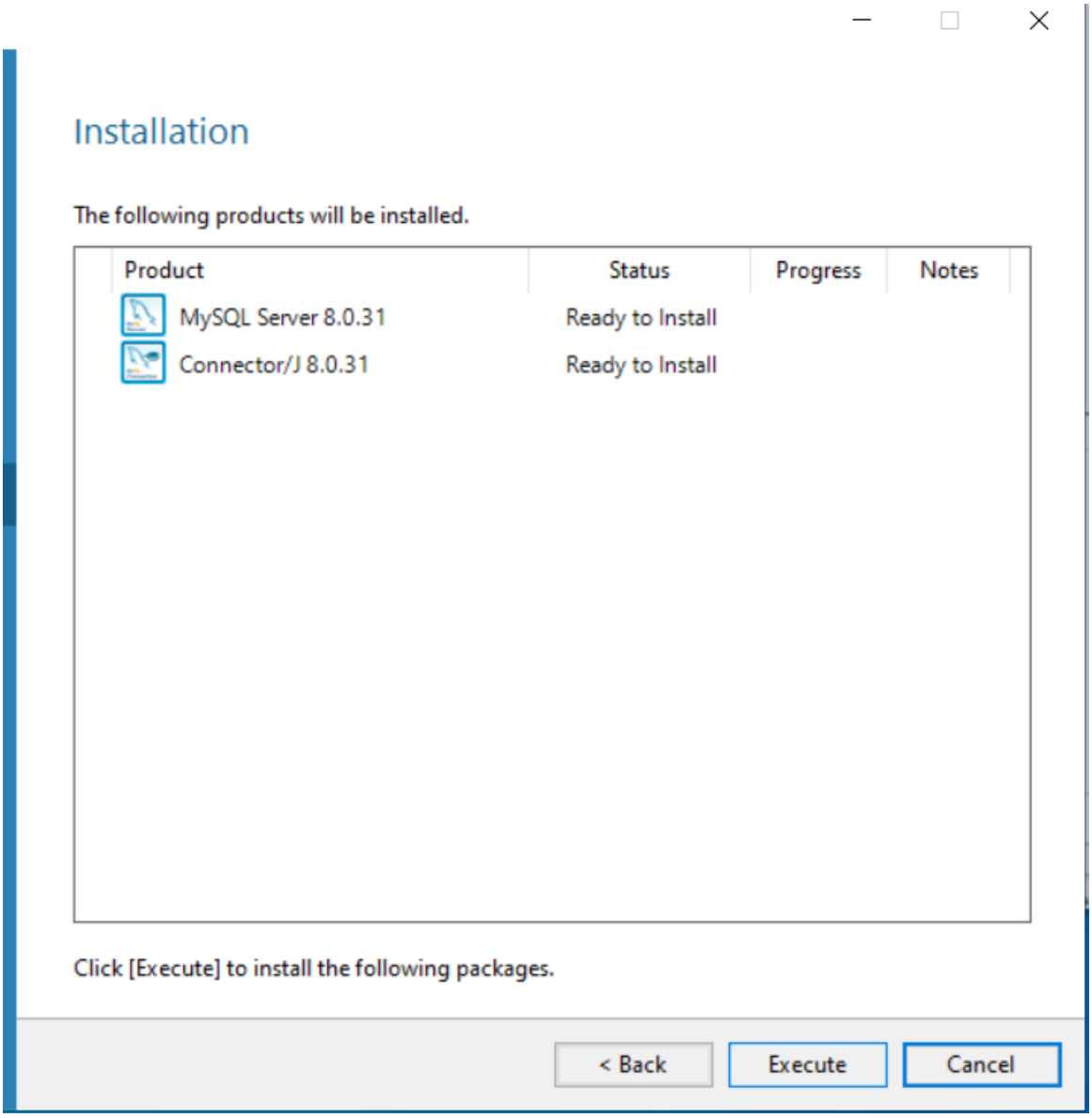
Requirement Details

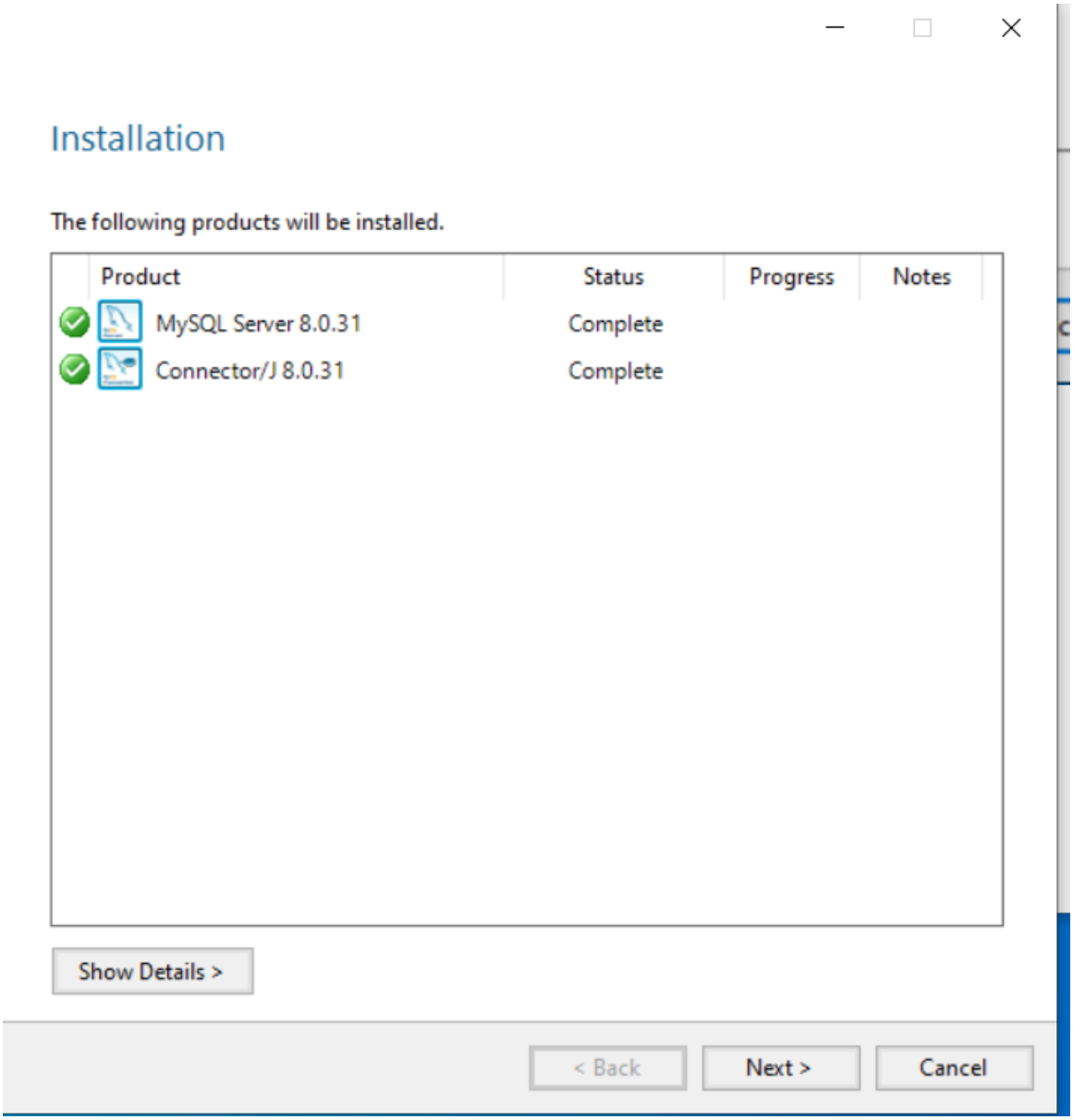
MySQL Installer is trying to resolve this requirement automatically.
There is nothing you need to do.

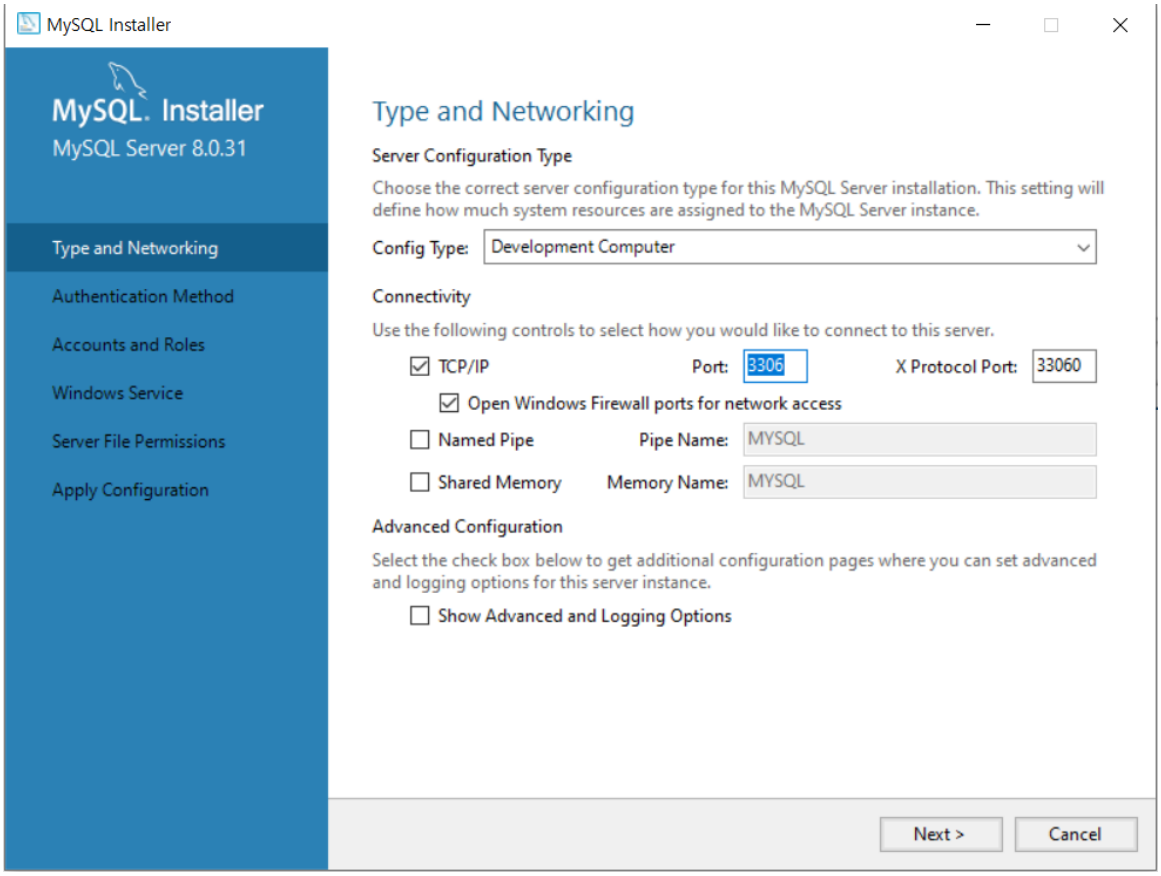
Requirement: Microsoft Visual C++ 2019 Redistributable Package (x64) is not installed. Latest binary compatible version will be installed if agreed

Status:

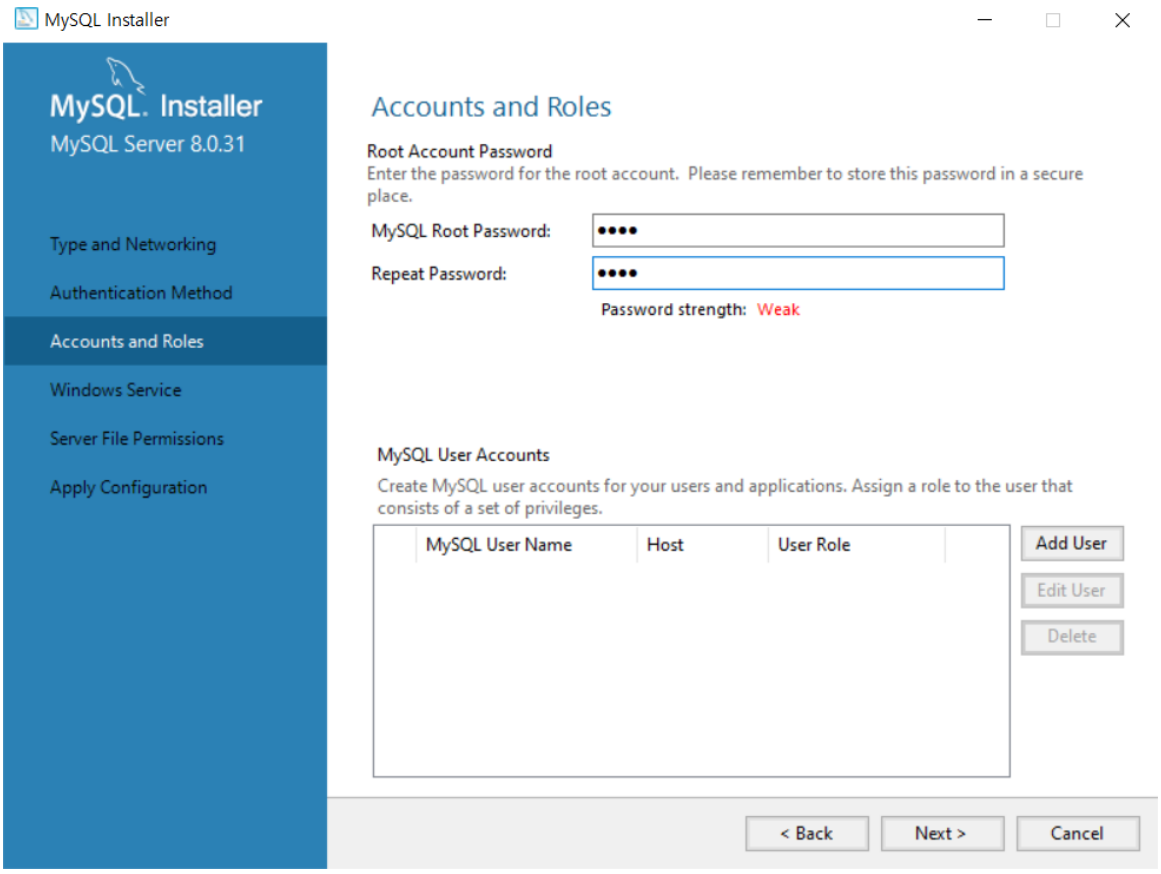
< Back | Next > | Cancel

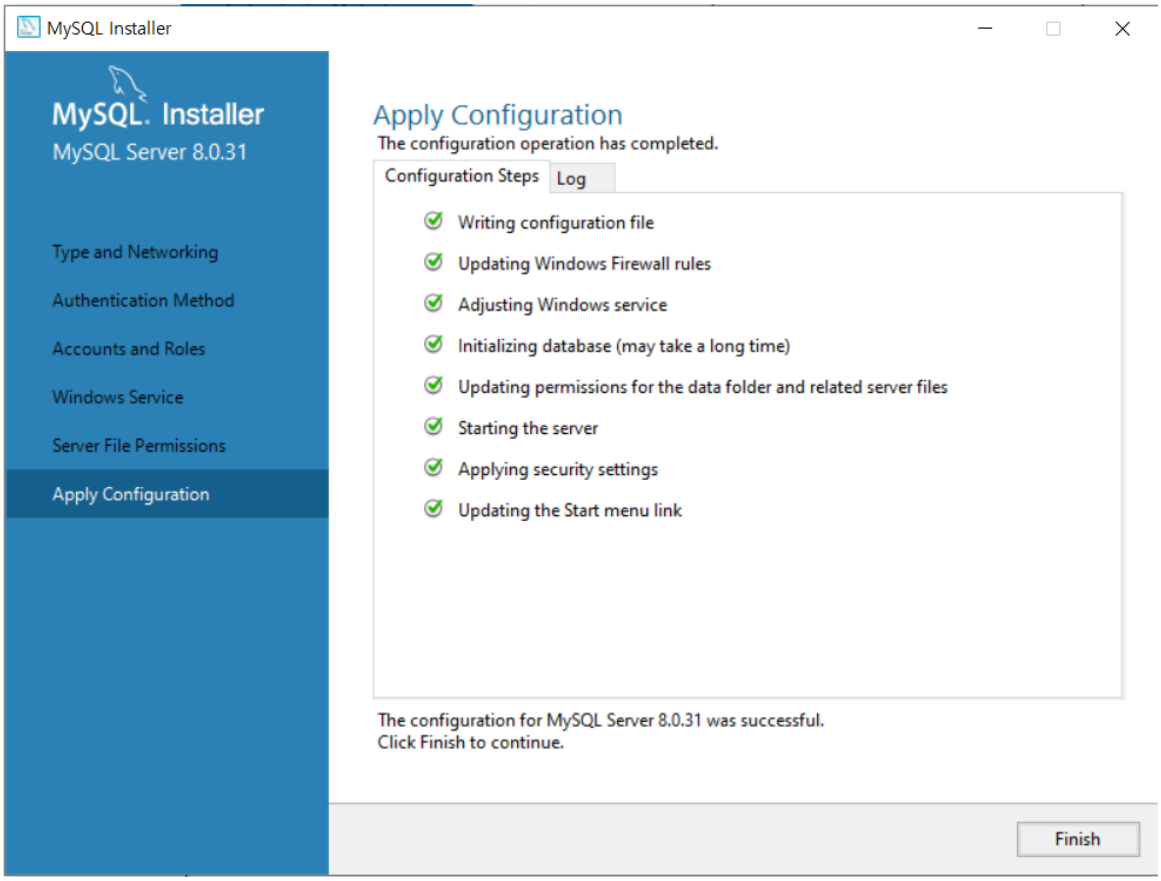

Installation
The following products will be installed.
Product
Status
Progress
Notes
MySQL Server 8.0.31
Ready to Install
Connector/J 8.0.31
Ready to Install
Click [Execute] to install the following packages.
< Back
Execute
Cancel

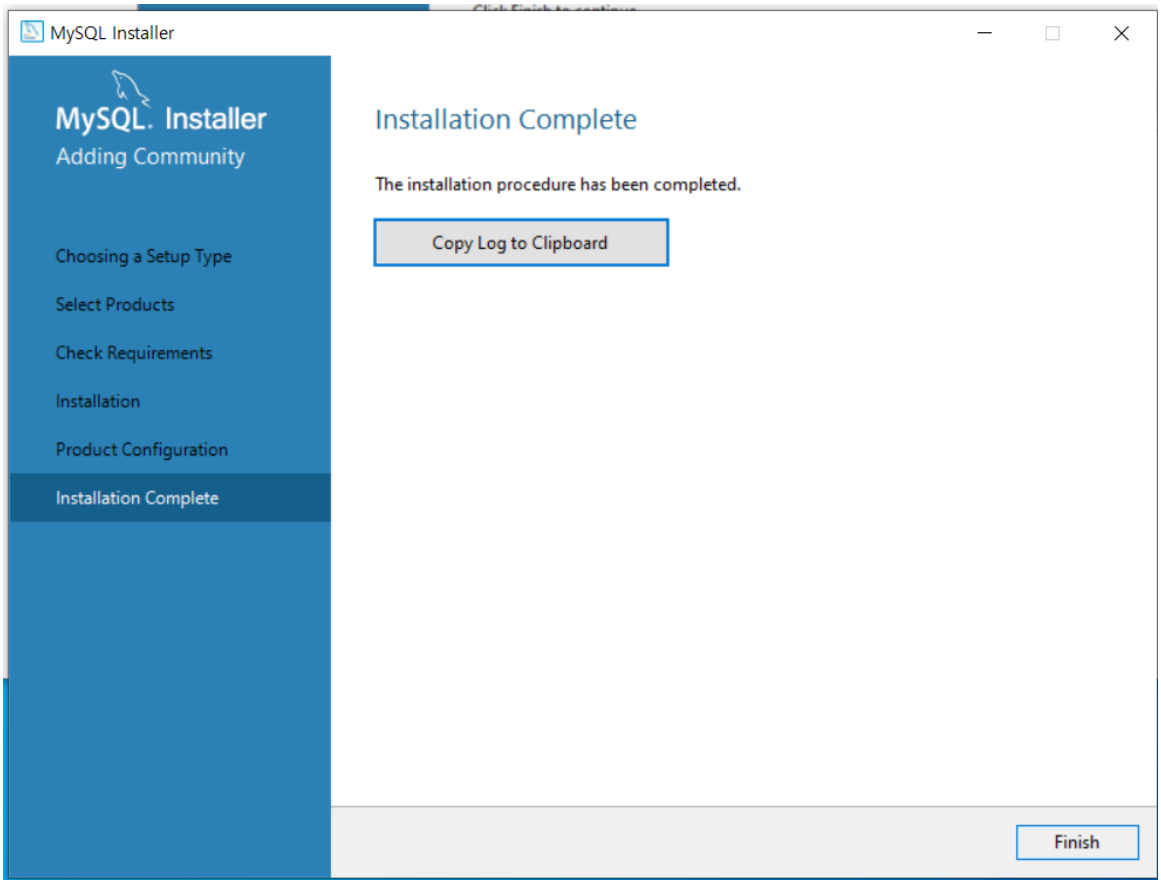

Installation
The following products will be installed.
Product
Status
Progress
Notes
MySQL Server 8.0.31
Complete
Connector/J 8.0.31
Complete
Show Details >
< Back
Next >
Cancel

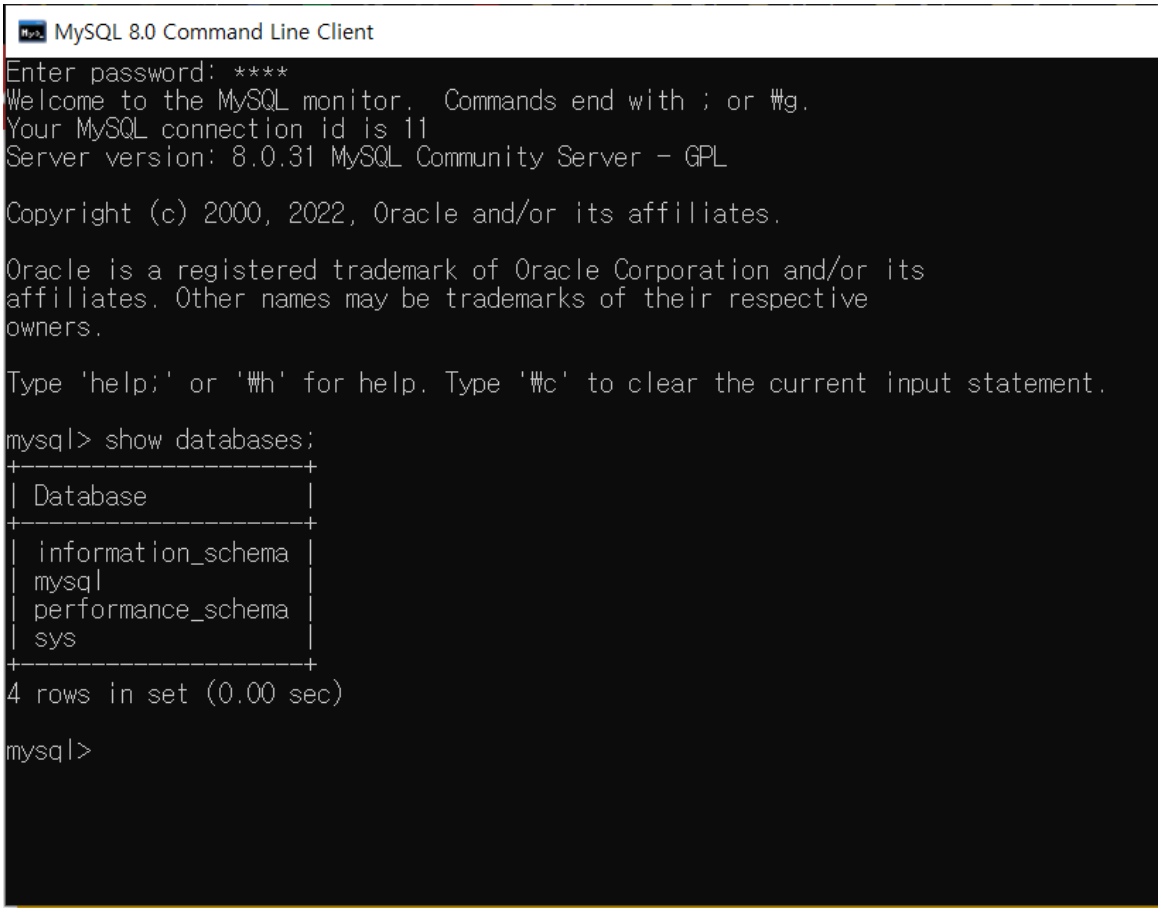


포트 번호는 3306을 사용하는 것이 좋다.


MySQL Installer
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
Add User
Edit User
Delete
< Back
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
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 Installer
MySQL. Installer
Adding Community
Choosing a Setup Type
Select Products
Check Requirements
Installation
Product Configuration
Installation Complete
Installation Complete
The installation procedure has been completed.
Copy Log to Clipboard
Finish

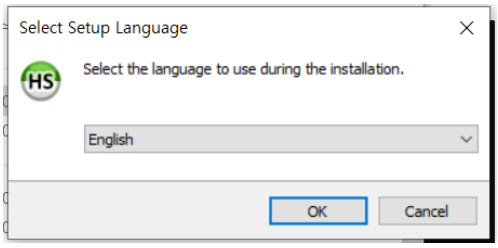


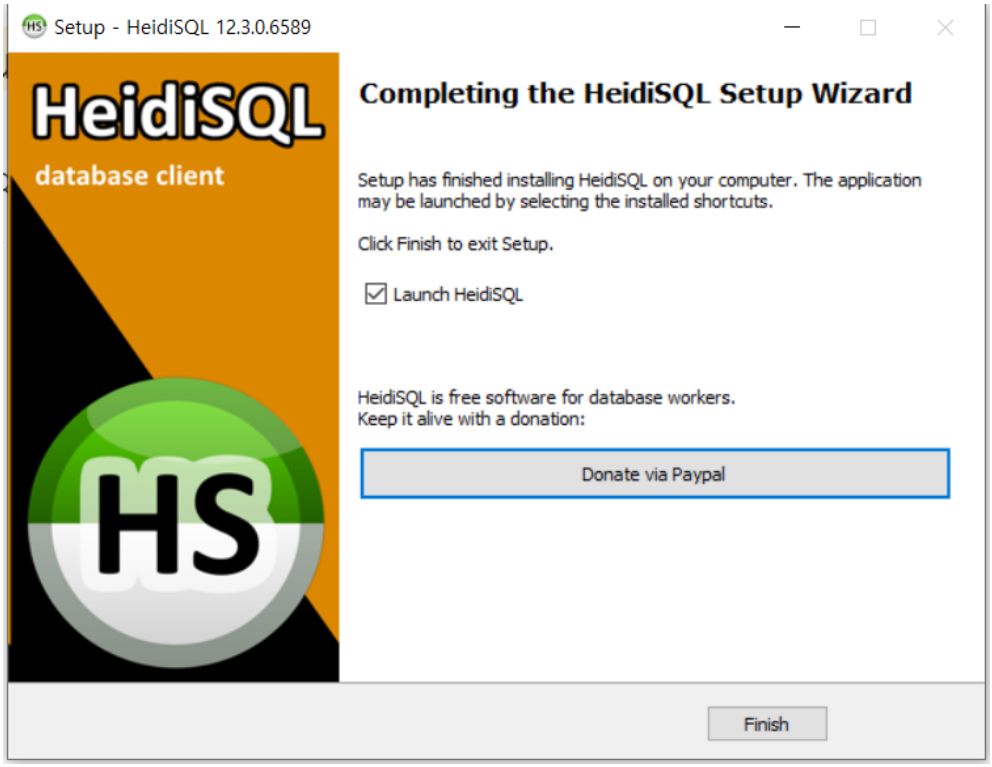
Heidi 설치

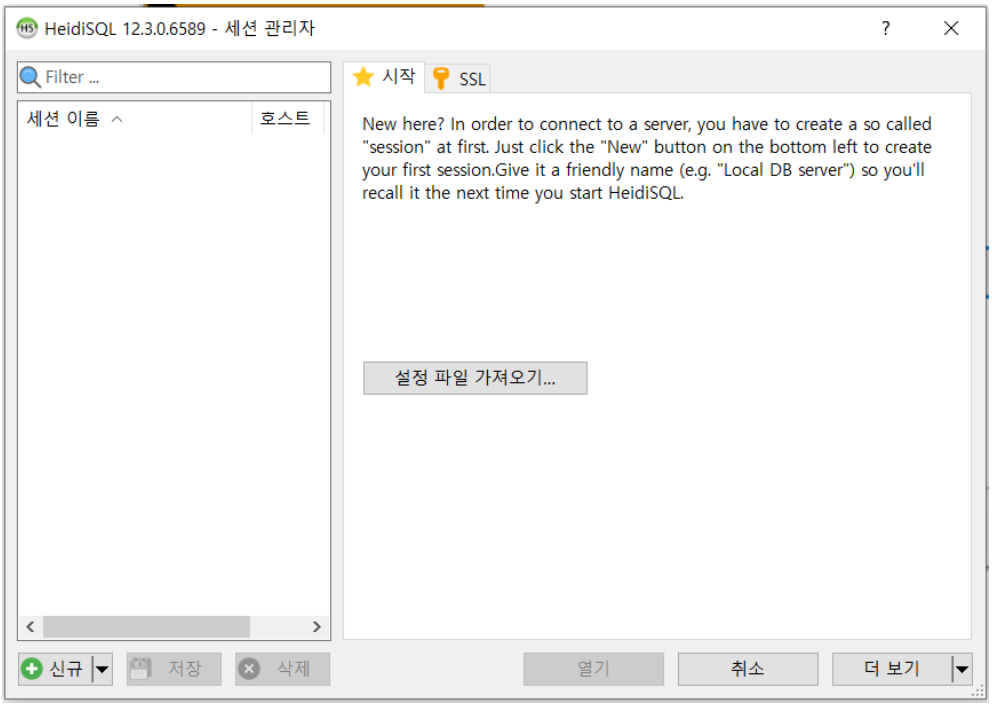
heidiSQL

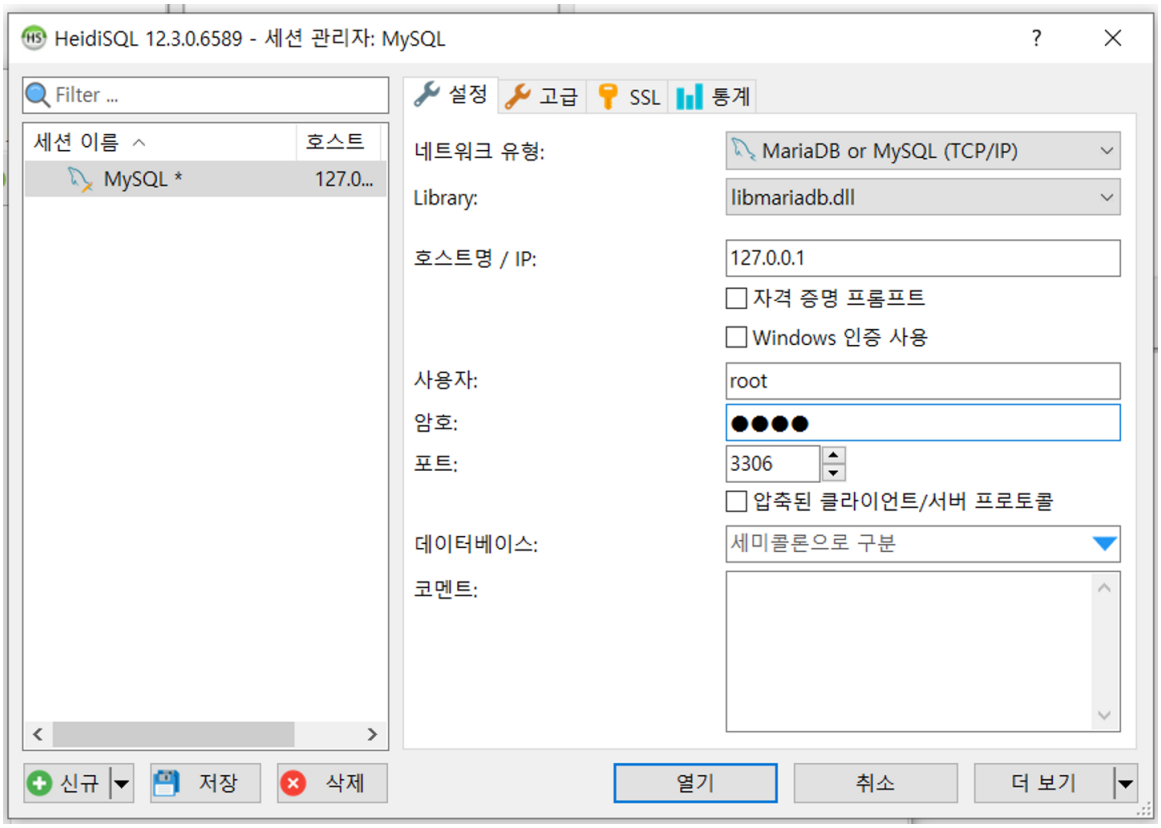
- MySQL 와 MariaDB 를 사용하기 쉽게 도와주는 도구

▼ 설치 및 사용




Setup - HeidiSQL 12.3.0.6589
HeidiSQL
database client
Completing the HeidiSQL Setup Wizard
Setup has finished installing HeidiSQL on your computer. The application may be launched by selecting the installed shortcuts.
Click Finish to exit Setup.
Launch HeidiSQL
HeidiSQL is free software for database workers.
Keep it alive with a donation:
Donate via Paypal
Finish


HeidiSQL 12.3.0.6589 - 세션 관리자
Filter ...
세션 이름
호스트
시작
SSL
New here? In order to connect to a server, you have to create a so called "session" at first. Just click the "New" button on the bottom left to create your first session.Give it a friendly name (e.g. "Local DB server") so you'll recall it the next time you start HeidiSQL.
설정 파일 가져오기...
신규
저장
삭제
열기
취소
더 보기



JDBC

JDBC 사용 6단계

1. import SQL packages
2. load JDBC Driver
3. create Connection Object
4. create Statement Object
5. execute SQL Query
6. close Connection Object (java 9부터 생략 가능)

▼ 코드 예시

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!-- step1 : import SQL package -->
<%@ page import="java.sql.*" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title></title>
</head>
```

```
<body>
<%
  /* Step2 : load JDBC Driver */
  try {
    Class.forName("com.mysql.jdbc.Driver");
  } catch (ClassNotFoundException e) {
    out.print("JDBC Driver load failed <br>" + e.getMessage());
  }

  /* step3 : create Connection Object */
  Connection conn = null;
  try {
    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/", "root", "0000");
  } catch (Exception e) {
    out.print("JDBC Driver load failed <br>" + e.getMessage());
  }

  /* step4 : create Statement Object */
  PreparedStatement pstmt = conn.prepareStatement("CREATE DATABASE test");

  /* step5 : execute SQL Query */
  pstmt.execute();

  /* step6 : close Connection Object (java 9부터 생략 가능) */
  pstmt.close();
  conn.close();
%>
</body>
</html>
```

Connection 객체의 PreparedStatement 사용

- Create Table

  ▼ 코드 예시

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!-- step1 : import SQL package -->
<%@ page import="java.sql.*" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title></title>
</head>
<body>
<%
  /* Step2 : load JDBC Driver */
  try {
    Class.forName("com.mysql.jdbc.Driver");
  } catch (ClassNotFoundException e) {
    out.print("JDBC Driver load failed <br>" + e.getMessage());
  }

  /* step3 : create Connection Object */
  Connection conn = null;
  try {
    // *테이블 지정*
    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/univ", "root", "0000");
  } catch (Exception e) {
    out.print("JDBC Driver load failed <br>" + e.getMessage());
  }
```

```
  String aql = "CREATE TABLE student ("
      + "hakbun   VARCHAR(10), "
      + "name   VARCHAR(20), "
      + "dept   VARCHAR(30), "
      + "addr   VARCHAR(50), "
      + "primary key(hakbun)"
      + ")";

  /* step4 : create Statement Object */
  PreparedStatement pstmt = conn.prepareStatement(aql);

  /* step5 : execute SQL Query */
  pstmt.execute();

  /* step6 : close Connection Object (java 9부터 생략 가능) */
  pstmt.close();
  conn.close();
%>
</body>
</html>
```

- Insert Table

▼ 코드 예시

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!-- step1 : import SQL package -->
<%@ page import="java.sql.*" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title></title>
</head>
<body>
<%
  /* Step2 : load JDBC Driver */
  try {
    Class.forName("com.mysql.jdbc.Driver");
  } catch (ClassNotFoundException e) {
    out.print("JDBC Driver load failed <br>" + e.getMessage());
  }

  /* step3 : create Connection Object */
  Connection conn = null;
  try {
    // *테이블 지정*
    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/univ", "root", "0000");
  } catch (Exception e) {
    out.print("JDBC Driver load failed <br>" + e.getMessage());
  }

  String hakbun = request.getParameter("hakbun");
  String name = request.getParameter("name");
  String dept = request.getParameter("dept");
  String addr = request.getParameter("addr");

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

  /* step4 : create Statement Object */
  PreparedStatement pstmt = conn.prepareStatement(sql);
  pstmt.setString(1, hakbun);
  pstmt.setString(2, name);
```

```
  pstmt.setString(3, dept);
  pstmt.setString(4, addr);

  /* step5 : execute SQL Query */
  pstmt.executeUpdate();

  /* step6 : close Connection Object (java 9부터 생략 가능) */
  pstmt.close();
  conn.close();
%>
</body>
</html>
```

- Select Table

▼ 코드 예시

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!-- step1 : import SQL package -->
<%@ page import="java.sql.*" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title></title>
</head>
<body>
<%
  /* Step2 : load JDBC Driver */
  try {
    Class.forName("com.mysql.jdbc.Driver");
  } catch (ClassNotFoundException e) {
    out.print("JDBC Driver load failed <br>" + e.getMessage());
  }

  /* step3 : create Connection Object */
  Connection conn = null;
  try {
    // *테이블 지정*
    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/univ", "root", "0000");
  } catch (Exception e) {
    out.print("JDBC Driver load failed <br>" + e.getMessage());
  }

  String sql = "SELECT * FROM student";

  /* step4 : create Statement Object */
  PreparedStatement pstmt = conn.prepareStatement(sql);
  /* step5 : execute SQL Query */
  pstmt.execute();
  ResultSet rs = pstmt.getResultSet();

%>
  <h3>List</h3>
  <table>
    <tr>
      <th>학번</th>
      <th>이름</th>
      <th>전공</th>
      <th>주소</th>
    </tr>
<%
  while(rs.next()) {
%>
```

```
    <tr>
      <td><%= rs.getString("hakbun") %></td>
      <td><%= rs.getString("name") %></td>
      <td><%= rs.getString("dept") %></td>
      <td><%= rs.getString("addr") %></td>
    </tr>
<%
  }
  /* step6 : close Connection Object (java 9부터 생략 가능) */
  rs.close();
  pstmt.close();
  conn.close();
%>

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

JSP로 데이터베이스 사용 4가지 방법

1. 각각의 JSP 파일에서 매번 직접 DB사용
2. DB 연결 부분만 별도의 File로 구성

```
<%@ include file="파일명" %>
```

▼ 코드 예시

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!-- step1 : import SQL package -->
<%@ page import="java.sql.*" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title></title>
</head>
<body>
<%@ include file="dbconnect.inc" %>
<%

  /* step4 : create Statement Object */
  PreparedStatement pstmt = conn.prepareStatement("CREATE DATABASE test");

  /* step5 : execute SQL Query */
  pstmt.execute();
%>
<%@ include file="dbclose.inc" %>
</body>
</html>
```

3. DB 연결 부분만 별도의 Class로 구성
    a. Connection 객체를 만들고 닫는 클래스

▼ 코드 예시

```
package jdbc6steps;

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

public class DBconnClose {
  private DBconnClose() {

  }

  // DB 연결 공통 부분 메서드
  public static Connection getConnextion() {
    /* Step2 : load JDBC Driver */
    try {
      Class.forName("com.mysql.jdbc.Driver");
    } catch (ClassNotFoundException e) {
      System.out.print("JDBC Driver load failed <br>" + e.getMessage());
    }

    /* step3 : create Connection Object */
    Connection conn = null;
    try {
      conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/", "root", "0000");
    } catch (Exception e) {
      System.out.print("JDBC Driver load failed <br>" + e.getMessage());
    }

    return conn;
  }

  // DB 해제 공통 부분 메서드
  public static void close(Connection conn, PreparedStatement pstmt) {
    try {
      if (pstmt != null){
        pstmt.close();
      }
      if (conn != null){
        conn.close();
      }
    } catch (SQLException e) {
      e.printStackTrace();
    }
  }

}
```

b. 사용

▼ 코드 예시

```
<%@page import="jdbc6steps.DBconnClose"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!-- step1 : import SQL package -->
<%@ page import="java.sql.*" %>
<!DOCTYPE html>
<html>
<head>
```

```
<meta charset="UTF-8">
<title></title>
</head>
<body>
<%
  // step 2, 3
  Connection conn = DBconnClose.getConnextion();

  /* step4 : create Statement Object */
  PreparedStatement pstmt = conn.prepareStatement("CREATE DATABASE test2");

  /* step5 : execute SQL Query */
  pstmt.execute();

  // step 6
  DBconnClose.close(conn, pstmt);
%>
</body>
</html>
```

4. Connection Pool 이용 (권장) → 내일 계속