Class 220113

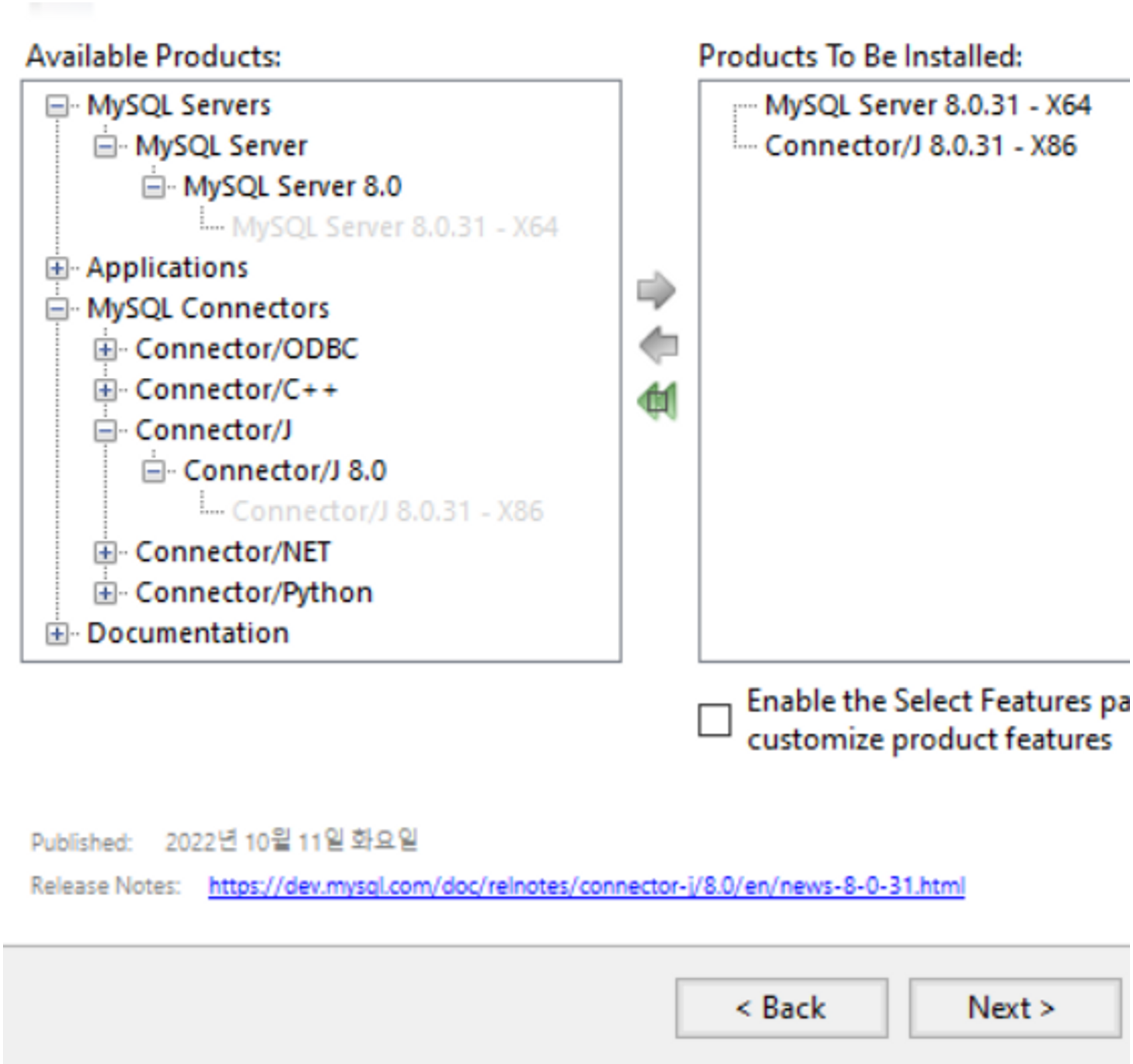
MYSQL DB 설치 및 설정

▼ MysqlDB 설치

1. mysql 홈페이지에서 다운로드

https://www.notion.so/Class-220113-edaeb5e55b674f4fb66cc0c7f0ac5864#43ddbbde3f2a4a12af56796a790d86a6

2. 설치 목록 중 MYSQL Server, Connector/J 설치 (Java에 연결)



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:

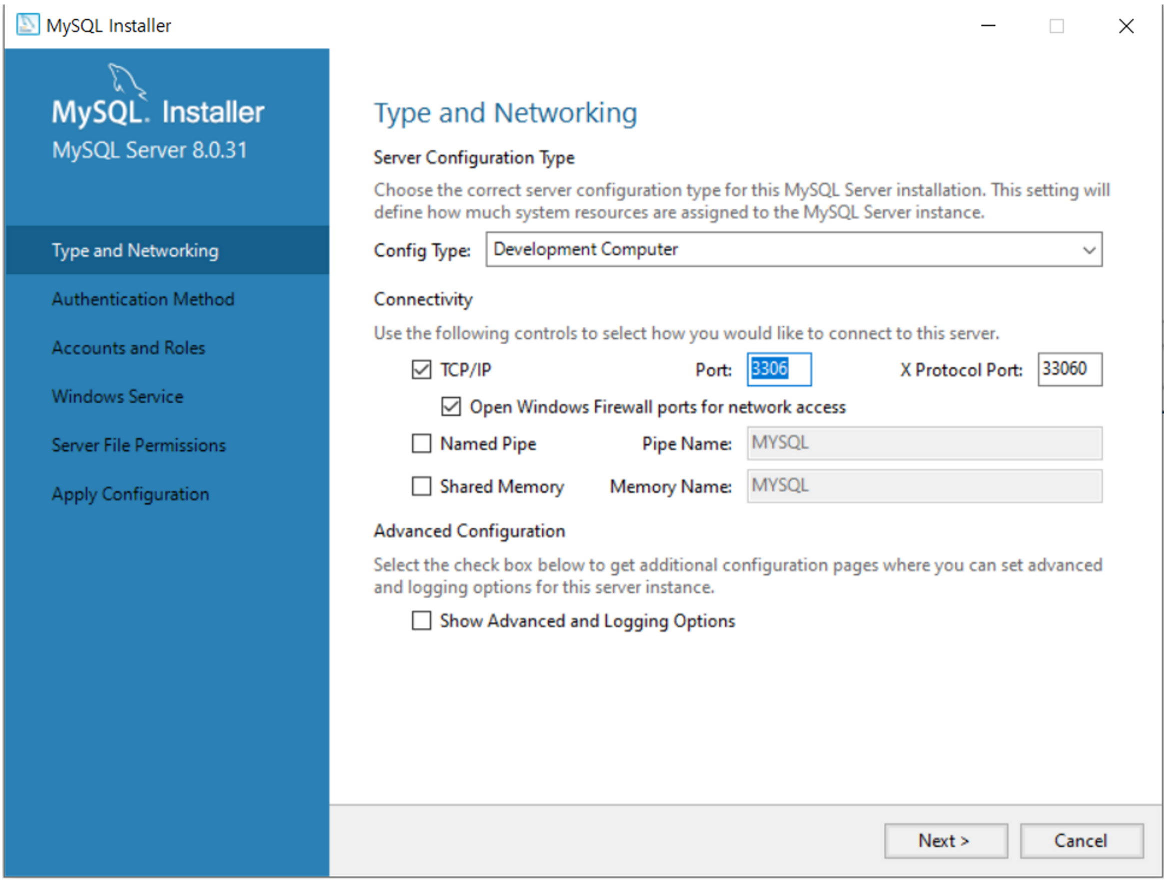
- MySQL Server 8.0.31 - X64
- Connector/J 8.0.31 - X86

□ 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

3. Port 번호 3306으로 지정(해당 포트번호로 사용하는 경우가 많기 때문)



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

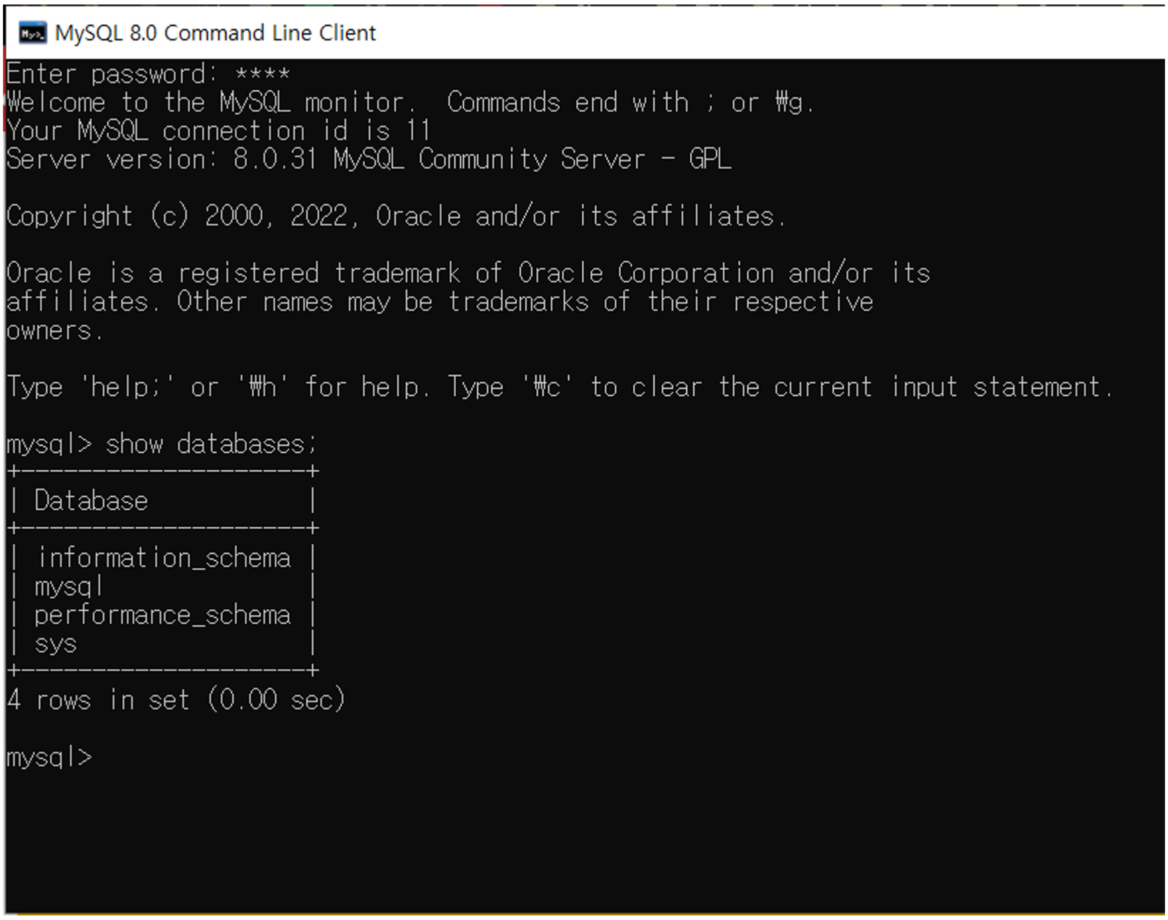
□ Shared Memory Memory Name: MYSQL

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

4. MySQL Command Line Client를 통해 정상 설치 확인



MySQL 8.0 Command Line Client

```
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or ₩g.
Your MySQL connection id is 11
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>
```

5. Developer 설치

- HeidiSQL (추천)

Download HeidiSQL

HeidiSQL runs fine on Windows 10 and 11 (and on Windows 7 + 8 with some minor issues) Running HeidiSQL on Wine can have various issues, depending on the version of Wine and HeidiSQL. Latest automatically compiled main executable and installers for

https://www.heidisql.com/download.php

- WorkBench

MySQL :: Download MySQL Workbench

MySQL Workbench provides DBAs and developers an integrated tools environment for: Database Design & Modeling SQL Development Database Administration Database Migration The Community (OSS) Edition is available from this page under the GPL.

https://dev.mysql.com/downloads/workbench/

QL Installer
for Windows
L Products. For All Windows Platforms.
In One Package.
(x86, 32 & 64-bit), MySQL Installer MSI
Go to Download P

DB와 JAVA 연결방식

1. 각각의 JSP 페이지를 통해 연결

▼ 코드

```
//MYSQL DB에 연결하여 test Database 생성
```

```
<%@page import="javax.servlet.jsp.tagext.TryCatchFinally"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>

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

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<%
 // 2. Step 2 load JDBC Driver
  try {
    Class.forName("com.mysql.jdbc.Driver"); //mysql DB 연결
    out.print("JDBC Driver loading success2<br>");
  } catch(ClassNotFoundException err){
    out.print("JDBC Driver loading error<br>" + err.getMessage());
  }

// 3. Step 3 create Connection Object

  Connection conn = null; // 전역변수화
  try {
    //try문 안에서 사용 시 지역변수
    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/", "root", "0000");
    out.print("JDBC Driver loading success3<br>");
  } catch(SQLException err){
    out.print("JDBC Driver loading error<br>" + err.getMessage());
  }

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

//5. Step 5 excute SQL Query;
  pstmt.executeUpdate();

//6. Step 6 Connection close (java9부터 생략 가능)
  pstmt.close(); // 나중에 생성 된 pstmt 객체 close
  conn.close(); // conn 객체 close

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

▼ 실습

1. Form으로 값을 입력 받아 Insert 처리

    - TBForm

    ```
    <%@ 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="get">
    학번 <input type="text" name="hakbun"> <br><br>
    이름 <input type="text" name="name"> <br><br>
    전공 <input type="text" name="dept"> <br><br>
    주소 <input type="text" name="addr"> <br><br>
    <button type="submit">insert</button>
    </form>

    </body>
    </html>
    ```

    - Insert 처리

```
// Step 4 create Statement Object

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

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

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


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

2. DB의 데이터를 가져와서 화면에 표시

```
<%@ 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());
  }

// Step 3 create Connection Object

  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());
  }



// Step 4 Statement Object

  String sql = "SELECT * FROM student";
  PreparedStatement pstmt = conn.prepareStatement(sql);


// Step 5 excute SQL Query

/*  pstmt.executeUpdate(); */            <---- 리턴되는 결과값이 없는 구문
  ResultSet rset = pstmt.executeQuery(); <---- 리턴되는 결과값을 가져오는 구문


while(rset.next()){
%>

<%=rset.getString("hakbun") %>|
<%=rset.getString("name") %>|
<%=rset.getString("dept") %>|
<%=rset.getString("addr") %><br>

<%
  }
  //Step 6 Close Connection
  rset.close();
  pstmt.close();
  conn.close();
%>



</body>
</html>
```

3. Login / Logout with Session

- LoginForm

```
<%@ 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="loginCheck.jsp">
    학번 : <input type="text" name="hakbun"><br>
    <button type="submit">로그인</button>
  </form>
</body>
</html>
```

- LoginCheck

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<% // 전송 받는 데이터 한글 처리
  request.setCharacterEncoding("UTF-8");
%>
<!-- import package -->
<%@ page import="java.sql.*" %>
<%

 // 2 load JDBC Driver
  try {
    Class.forName("com.mysql.jdbc.Driver"); //mysql DB 연결
  } catch(ClassNotFoundException err){
    out.print("JDBC Driver loading error<br>" + err.getMessage());
  }

// Step 3 create Connection Object

  Connection conn = null; // 전역변수화
  try {
    //try문 안에서 사용 시 지역변수
    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/univ", "root", "0000");
  } catch(SQLException err){
    out.print("JDBC Driver loading error<br>" + err.getMessage());
  }

//4. Step 4 create Statement Object
    String hakbun = request.getParameter("hakbun");

  String sql = "SELECT * FROM student WHERE hakbun = ? ";
  PreparedStatement pstmt = conn.prepareStatement(sql);
  pstmt.setString(1, hakbun);

// Step 5 excute SQL Query
```

```
  /*  pstmt.executeUpdate(); */
  ResultSet rs = pstmt.executeQuery();

  if(!rs.isBeforeFirst()){ // DB에서 학번에 해당하는 자료가 있어서 반환되었는지 확인
    //isBeforFirst() : 커서가 첫 번째 행 앞에 있으면 true, 커서가 그 외의 위치에 있거나 결과 집합에 행이 들어 있지 않으면 false
    out.print("<script>alert('해당 학번은 존재하지 않습니다.');"
                  + "history.back();"
                  + "</script>");
    return;
  }

  rs.next();
  String dbhakbun = rs.getString("hakbun");
  String dbname = rs.getString("name");

  // DB 학번과 입력한 학번을 비교하여 값이 같으면 세션을 생성
  // "hakbun" dbhakbun, "name" dbname
  if(hakbun.equals(dbhakbun)){
    session.setAttribute("hakbun", dbhakbun);
    session.setAttribute("name", dbname);

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

  //6. Step 6 Connection close (java9부터 생략 가능)
      rs.close();
      pstmt.close(); // 나중에 생성 된 pstmt 객체 close
      conn.close(); // conn 객체 close
%>
```

2. 연결 객체를 inc파일로 생성하여 연결

▼ 코드

- DBConnection.inc

```
<%
 // Step 2 load JDBC Driver
  try {
    Class.forName("com.mysql.jdbc.Driver"); //mysql DB 연결
    out.print("JDBC Driver loading success2<br>");
  } catch(ClassNotFoundException err){
    out.print("JDBC Driver loading error<br>" + err.getMessage());
  }

// Step 3 create Connection Object

  Connection conn = null; // 전역변수화
  try {
    //try문 안에서 사용 시 지역변수
    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/", "root", "0000");
    out.print("JDBC Driver loading success3<br>");
  } catch(SQLException err){
    out.print("JDBC Driver loading error<br>" + err.getMessage());
  }
 %>
```

- DBClose.inc

```
<%
  //6. Step 6 Connection close (java9부터 생략 가능)
      pstmt.close(); // 나중에 생성 된 pstmt 객체 close
      conn.close(); // conn 객체 close
%>
```

- jsp 파일에 include 처리

```
<%@page import="javax.servlet.jsp.tagext.TryCatchFinally"%>
<%@ 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>

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

<%


//4. Step 4 create Statement Object
  PreparedStatement pstmt = conn.prepareStatement("CREATE DATABASE test3");

//5. Step 5 excute SQL Query;
  pstmt.executeUpdate();
%>
<%@ include file="DBClose.inc" %>
</body>
</htm
```

3. 연결 객체를 class로 생성하여 연결

▼ 코드

- DB연결 Class(DBconnclose.java)

```
package jdbc6steps;

import java.sql.*;

public class DBconnclose {
  //DB연결 공통 부분 메서드
  public static Connection getConnection() {
    try {
      Class.forName("com.mysql.jdbc.Driver"); //mysql DB 연결
    } catch(ClassNotFoundException err){
      System.out.println(err.getMessage());
    }

  // Step 3 create Connection Object

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

    return conn;
  }


  //DB해제 공통부분 메서드
  public static void closeConnection(PreparedStatement pstmt, Connection conn) {
    try {
      pstmt.close();
      conn.close();
    } catch (SQLException e) {
      System.out.println(e.getMessage());
    }
  }
}
```

- jsp파일 처리

```
<%@page import="javax.servlet.jsp.tagext.TryCatchFinally"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>

<!-- Step 1 import SQL Packages -->
<%@ page import="java.sql.*, jdbc6steps.DBconnclose" %>

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

<%
// class로 DB연결
  Connection conn = DBconnclose.getConnection();
%>

<%
//4. Step 4 create Statement Object
  PreparedStatement pstmt = conn.prepareStatement("CREATE DATABASE test4");

//5. Step 5 excute SQL Query;
  pstmt.executeUpdate();
%>
<% DBconnclose.closeConnection(pstmt, conn); %>
</body>
</html>
```

4. DB Connection Pool 사용

1/16 진행