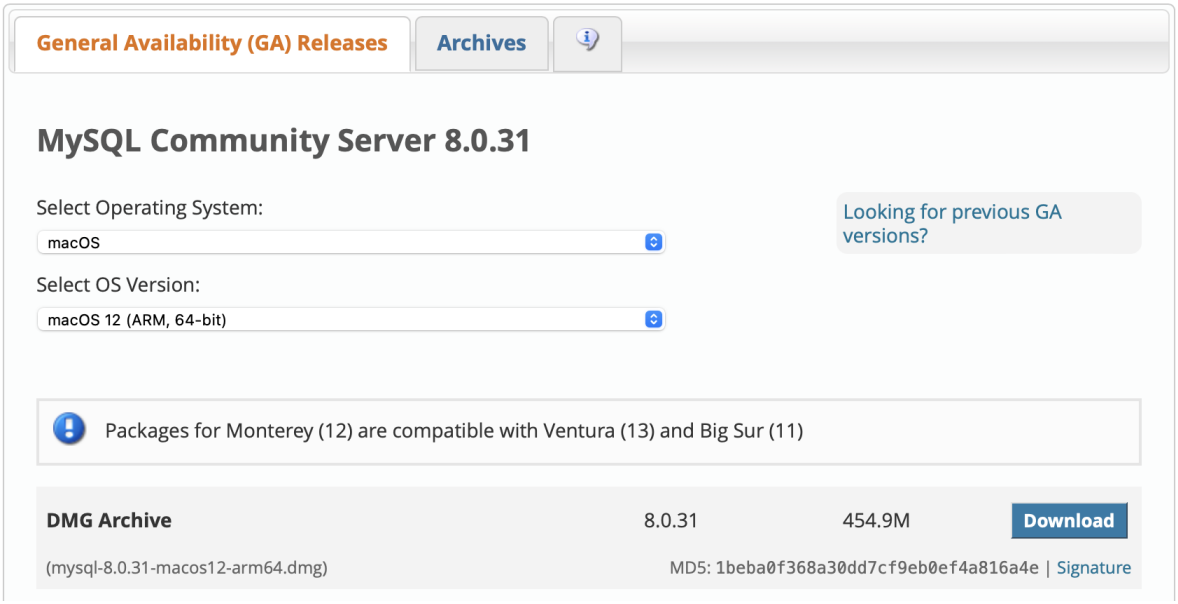
Class 230113

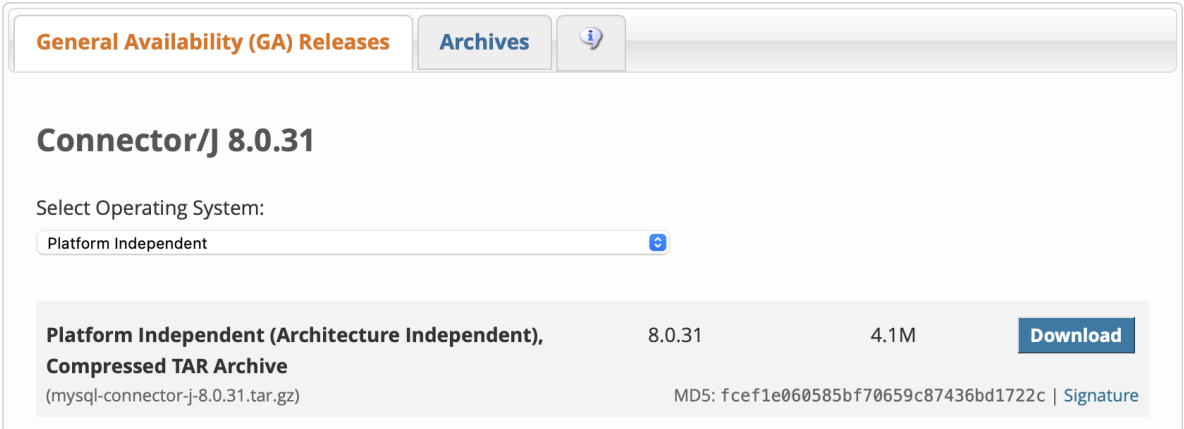
MySQL

MySQL

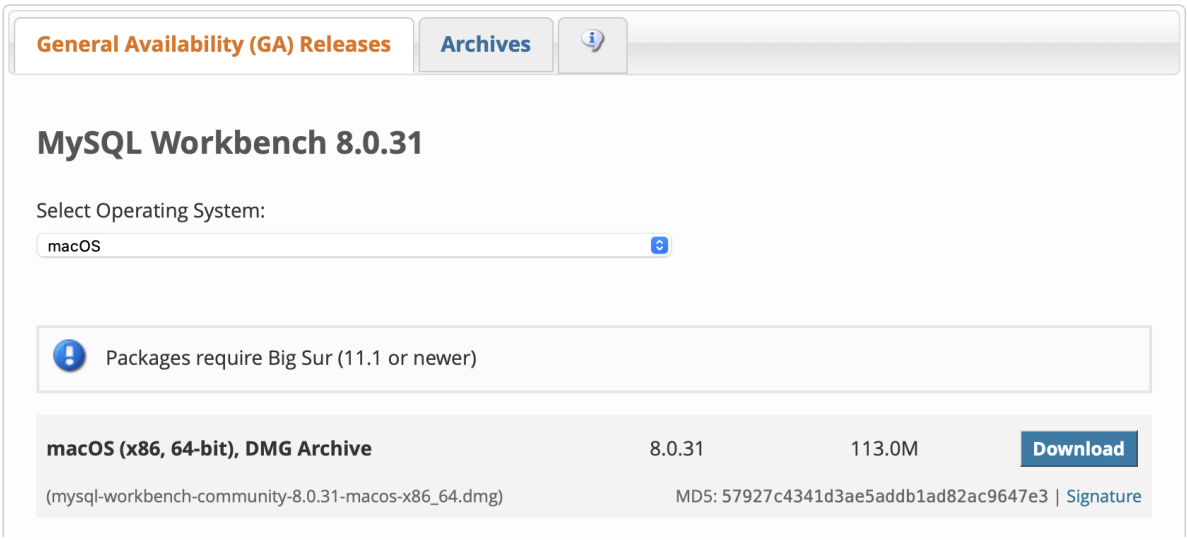


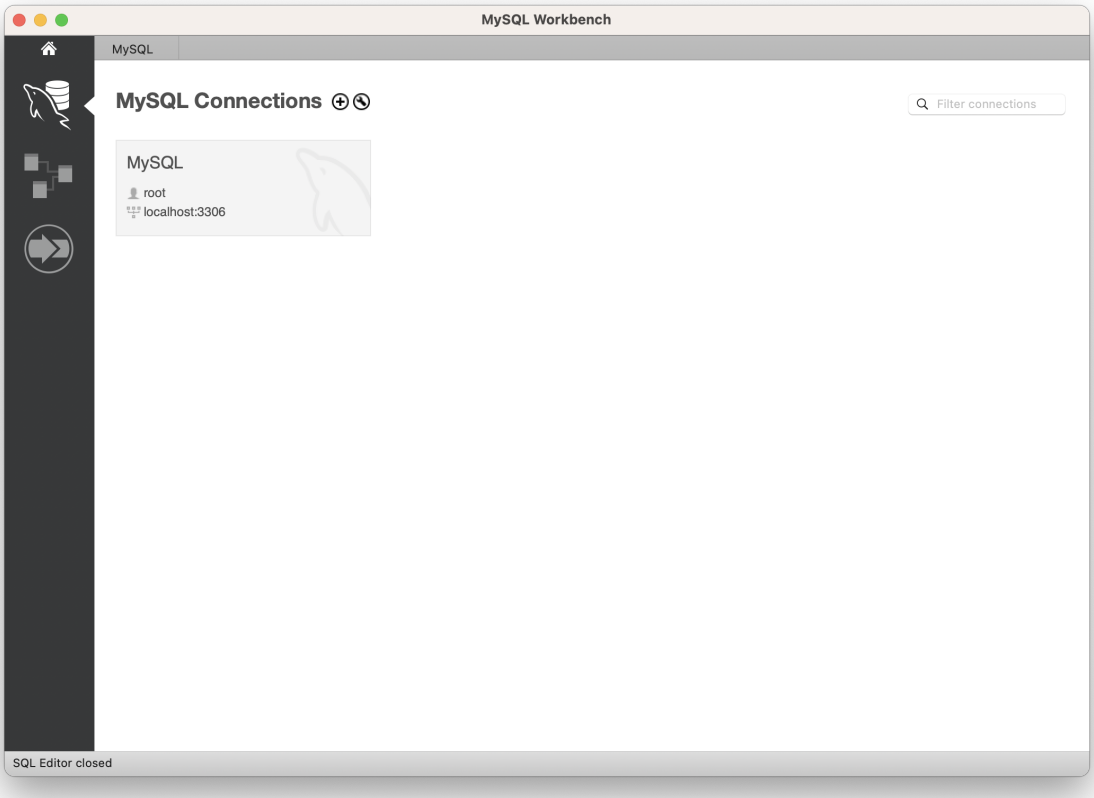
Connector

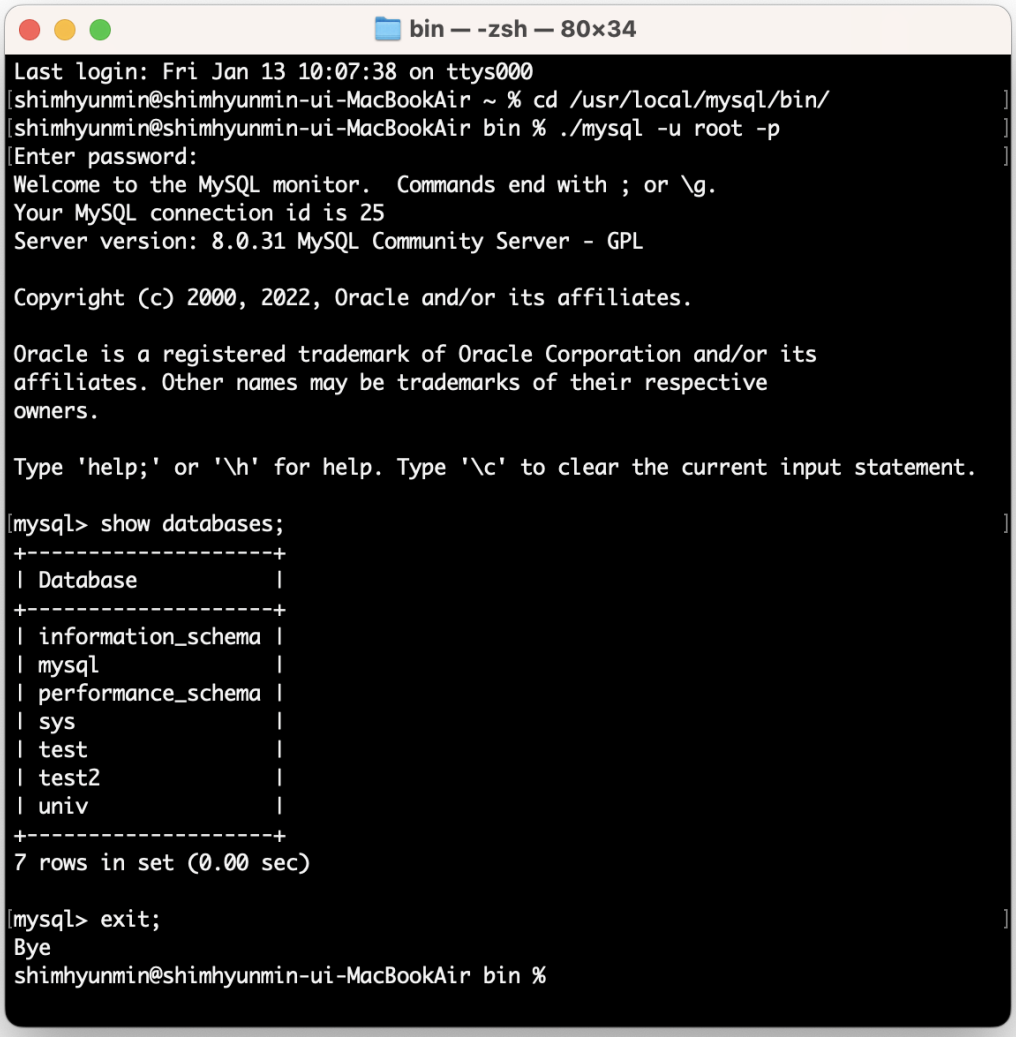
WEB-INF > lib > mysql-connector-j-8.0.31.jar



Workbench


General Availability (GA) Releases
Archives
MySQL Workbench 8.0.31
Select Operating System:
macOS
Packages require Big Sur (11.1 or newer)
macOS (x86, 64-bit), DMG Archive
8.0.31
113.0M
Download
(mysql-workbench-community-8.0.31-macos-x86_64.dmg)
MD5: 57927c4341d3ae5addb1ad82ac9647e3 | Signature


MySQL Workbench
MySQL
MySQL Connections
Filter connections
MySQL
root
localhost:3306
SQL Editor closed



JSP로 데이터 베이스를 사용하는 4가지 방법

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

JDBC basic

6steps

- STEP 1 import SQL Packages

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

- STEP 2 load JDBC Driver

```
//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

```
//Step 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());
}
```

- STEP 4 create Statement Object

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

- STEP 5 excute SQL Query

```
//Step 5 Execute SQL Query
pstmt.executeUpdate();
```

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

```
//Step 6 Close Connection
pstmt.close();
conn.close();
```

DBCreate

```
<%@ 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/","root","0000");
  } catch(SQLException err) {
    out.print("Connection Object error<br>" + err.getMessage());
  }

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

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

  //Step 6 Close Connection
  pstmt.close();
  conn.close();
%>
</body>
</html>
```

DBDrop

```
jdbc:mysql://localhost:3306/

DROP DATABASE test
```

TBCreate

💡 jdbc:mysql://localhost:3306/univ

```
  //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 Create Statement Object
  String sql = "CREATE TABLE student("
      + "hakbun   varchar(10),"
      + "name   varchar(10),"
      + "dept   varchar(20),"
      + "addr   varchar(30),"
      + "primary  key(hakbun))";

  PreparedStatement pstmt = conn.prepareStatement(sql);
```

```
  //Step 5 Execute SQL Query
  pstmt.executeUpdate();
```

TBForm

```
<form action="TBInsert.jsp">
    학번<input type="text" name="hakbun"><br>
    이름<input type="text" name="name"><br>
    전공<input type="text" name="dept"><br>
    주소<input type="text" name="addr">
    <input type="submit" value="전송">
</form>
```

TBInsert

```
  //Step 4 Create Statement Object
  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);

  //Step 5 Execute SQL Query
  pstmt.executeUpdate();
```

TBList

```
  //Step 4 Create Statement Object
  String sql = "SELECT * FROM student";
  PreparedStatement pstmt = conn.prepareStatement(sql);

  //Step 5 Execute(Query VS Update) SQL Query
  ResultSet rset = pstmt.executeQuery();
  //while 반복횟수모를때 vs for 반복횟수알때
  while(rset.next()){//다음 데이터가 있을 때까지 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>
```

---

login / logout with session

1. 아이디와 비밀번호를 모두 정상적으로 입력한 경우 > 로그인 성공
2. 비밀번호가 틀린 경우 > 로그인 실패 (다시시도하세요)
3. 아이디가 DB에 존재하지 않는 경우 > 로그인 실패 (회원가입 페이지로 리다이렉트)

LoginCheck

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

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

<%
  //한글 처리
  request.setCharacterEncoding("utf-8");

  //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 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 Execute(Query VS Update) SQL Query
  ResultSet rset = pstmt.executeQuery();

  //디비에서 학번에 해당하는 자료가 있어서 반환되었는지 확인
  if(!rset.isBeforeFirst()) {//없으면
    out.print("<script>alert('해당 학번은 존재하지 않습니다.');"
        +"history.back();"
        +"</script>");
    return;
  }
  rset.next();//있으면 다음 자료
  String dbhakbun = rset.getString("hakbun");
  String dbname = rset.getString("name");

  //dbhakbun hakbun 같으면 session 생성
  if(dbhakbun.equals(hakbun)) {//문자열 비교
      session.setAttribute("hakbun", dbhakbun);
      session.setAttribute("name", dbname);
```

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

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

공통부분 module

dbconnect.inc

```
<%
  //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/","root","0000");
  } catch(SQLException err) {
    out.print("Connection Object error<br>" + err.getMessage());
  }
%>
```

dbclose.inc

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

inc 적용

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

dbconnclose.java

```
package jdbc6steps;

import java.sql.*;
```

```
public class dbconnclose {
  //DB 연결 공통부분 메소드
  public static Connection getConnection() {
    //Step 2 Load JDBC Driver
    try {
      Class.forName("com.mysql.jdbc.Driver");
    } catch(ClassNotFoundException err) {
      System.out.println("JDBC Driver loading error<br>" + 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("Connection Object error<br>" + err.getMessage());
    }

    return conn;
  }

  //DB 해제 공통부분 메소드
  public static void closeConnection(PreparedStatement pstmt, Connection conn) {
    //Step 6 Close Connection
    try {
      pstmt.close();
      conn.close();
    } catch (SQLException e) {
      e.printStackTrace();
    }
  }
}
```

dbconnclose 적용 (DBCreate)

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

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

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

<%
  //Step 2,3
  Connection conn = dbconnclose.getConnection();

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

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

  //Step 6
  dbconnclose.closeConnection(pstmt, conn);
%>

</body>
</html>
```