

class 230113

1교시

mysql

- Community 버전 다운
 - 윈도우용 → 용량 많은거 → no thanks → heidisql(개발도구)(워크벤치가 있지만 이게 좋다)
 - 포트번호 무조건 3306을 사용해야만함

2교시

mysql, heidi

- CREATE DATABASE univ;
- SHOW DATABASES;

| 이거를 쉽게 해주는게 heidi

jdbc basic

- 6STEP
 - STEP 1
 - Import SQL Packages
 - STEP 2
 - Loading JDBC Driver
 - STEP 3
 - Create Connection Object
 - STEP 4
 - Create Statement Object
 - STEP 5
 - Excute SQL Query
 - STEP 6
 - Close Connection (java8 부터 생략 가능)

jdbc basic

- JSP 로 데이터 베이스를 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 ★** (다음주 월요일에,...)

3교시

step1.jsp (import SQL Package)

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

<!-- Step 1 import SQL Packages -->
<%@page import = "java.sql.*" %> // <----- Class를 사용하기 위해서
```

step2.jsp (Loading JDBC Driver)

```
<body>
<%
//STEP 2 Load JDBC Driver (lib아래 jar파일과 연결됨)
try{
    Class.forName("com.mysql.jdbc.Driver"); //forName 안에 이름이 바뀜 (maria, oracle 등)
}catch(ClassNotFoundException err){ // catch 안써도 되는데 예러 확인용
    out.print("JDBC Driver loading err<br>" + err.getMessage());
}
out.print("JDBC Driver loading success<br>"); //잘 연결됐는지 찍어봄

// com.mysql.jdbc.Driver   mySQL용
// org.maria.jdbc.Driver   maria용
%>
</body>
```

step3.jsp (Create Connection Object) - 진짜 연결

```
<body>
<%
//STEP 2 Load JDBC Driver (lib아래 jar파일과 인식)
try{
    Class.forName("com.mysql.jdbc.Driver"); //forName 안에 이름이 바뀜 (maria, oracle 등)
}catch(ClassNotFoundException err){
    out.print("JDBC Driver loading err<br>" + err.getMessage()); // catch 안써도 되는데 예러 확인용
}
out.print("JDBC Driver loading success<br>"); //잘 연결됐는지 찍어봄
// com.mysql.jdbc.Driver   mySQL용

//STEP 3 Create Connection Object (진짜 연결)
//Connection 자료형은 import java.sql에서 옴
try{
    Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306","root","0000");//root - 아이디, 0000 - 비번
}catch(SQLException err){
    out.print("Connection Object error<br>" + err.getMessage());
}

%>
</body>
```

step4.jsp (Create Statment Object) - 구문 작성

```
//step 4 - Create Statement Object (구문만 작성)

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

//1. conn 문제점 -> 지역변수여서
//2. conn 해결책 -> 29line
```

```

/*해결책*/
-> Connection conn = null;

try{
    -> conn = DriverManager.getConnection("jdbc:mysql://localhost:3306","root","0000");
    //root - 아이디, 0000 - 비번
}catch(SQLException err){
    out.print("Connection Object error<br>" + err.getMessage());
}
//step 4 - Create Statement Object (구문만 작성)
PreparedStatement pstmt = conn.prepareStatement("CREATE DATABASE test");

```

step5.jsp (Excute SQL Query) - 진짜 쿼리 실행

```

// step 5 - Excute SQL Query (진짜 던짐, 쿼리진짜실행)

pstmt.executeUpdate();

```

step6.jsp (Close Connection) , (java 8부터 안해줘도 됨)

```

// step 6 - Close Connection (안해줘도 되긴 함)

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

```

▼ 따로 공부할것

- try catch

4교시

▼ 파일

DBCCreate.jsp

DBDrop.jsp

- 자주 접하게 되는 에러(quiz)

- TBCCreate.jsp

Quiz 1 jsp로 테이블로 생성

테이블 명 student (univ DB 이용)

필드 : hakbun, name, dept, addr

```

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);

```

→ but 그대로는 안될꺼다,,, (해결법)

```

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/univ","root","0000");
//univ -> DB이름까지 지정!!

```

- TBIInsert (테이블에 데이터 넣기)

- 문제점

- 글자화 됨 (데이터가 null로 들어감)

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

String sql = "INSERT student VALUES(hakbun, name, dept, addr)";
```

hakbun	name	dept	addr
	(NULL)	(NULL)	(NULL)

- 해결법

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

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);
```

univ.student: 2 행 (총) (대략적)

hakbun	name	dept	addr
	(NULL)	(NULL)	(NULL)
1111	홍길동	컴공	서울

5교시

- 화면에서 입력받아서 db에 던져넣기

- TBFom.jsp → TBIInsert.jsp

TBFom.jsp

```
<body>
<form action = TBIInsert.jsp method = post>
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"><br>
<input type = "submit" value = "보내기"><p>
</form>
</body>

```

TBInsert.jsp

```

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

```

- student에 있는 리스트 몽창 빼기
 - TBList.jsp
 - (JDBC 스텝 3까지는 똑같다) → 모듈화 가능!
 - 일단 던지고
 - 받아와야함 : ResultSet rset
 - (ResultSet → impots. java.sql 안에 있음)

```

// pstmt.executeUpdate(); //그냥 던질때 (테이블 생성)

ResultSet rset = pstmt.executeQuery();
//던지고 다시 가지고 나올때 (조회)

```

- 화면에 출력
 - rset을 뿌려주면 됨

```

// step 5 - Excute SQL Query (진짜 던짐, 쿼리진짜실행)

// pstmt.executeUpdate(); //그냥 던질때 (테이블 만들어라)
ResultSet rset = pstmt.executeQuery(); //던지고 다시 가지고 나올때 (조회)

while(rset.next()){
  %>
  <%= rset.getString("hakbun")%>
  <%= rset.getString("name")%>
  <%= rset.getString("dept")%>
  <%= rset.getString("addr")%><br>
  <% }
// step 6 - Close Connection (안해줘도 되긴 함)

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

```

6교시

login in / log out with session

1. 아이디와 비번을 모두 정상적으로 입력한 경우 → 로그인 성공
2. 비번이 틀릴 경우 → 로그인 실패
3. 아이디가 디비에 존재하지 않는 경우 → 로그인 실패(회원 가입 페이지로 리다이렉트)
→ 오늘 이것만 살짝

- DB에 한글 깨질 때

```
<% // 전송 받는 데이터 한글 처리    request.setCharacterEncoding("UTF-8");%>
```

- 로그인 (db에 있는 아이디와 비교, 지금은 hakbun)

LoginForm.jsp

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

LoginCheck.jsp

```
//DB에서 학번에 해당하는 자료가 있어서 반환되었는지 확인
if(!rset.isBeforeFirst()){ //<- 없으면
    out.print("<script>alert('해당 학번은 존재하지 않습니다.');
```

▼ 공부해야할것

- 리다이렉트
- 자바스크립트(js)
- session
 - setAttribute, getAttribute

▼ *

- jsp에서는 println안먹
- 컨트롤 알트 화살표 아래 → 똑같은거 밑줄에 복사
- out.print → jsp문법

7교시

dbconnect.inc + dbclose.inc

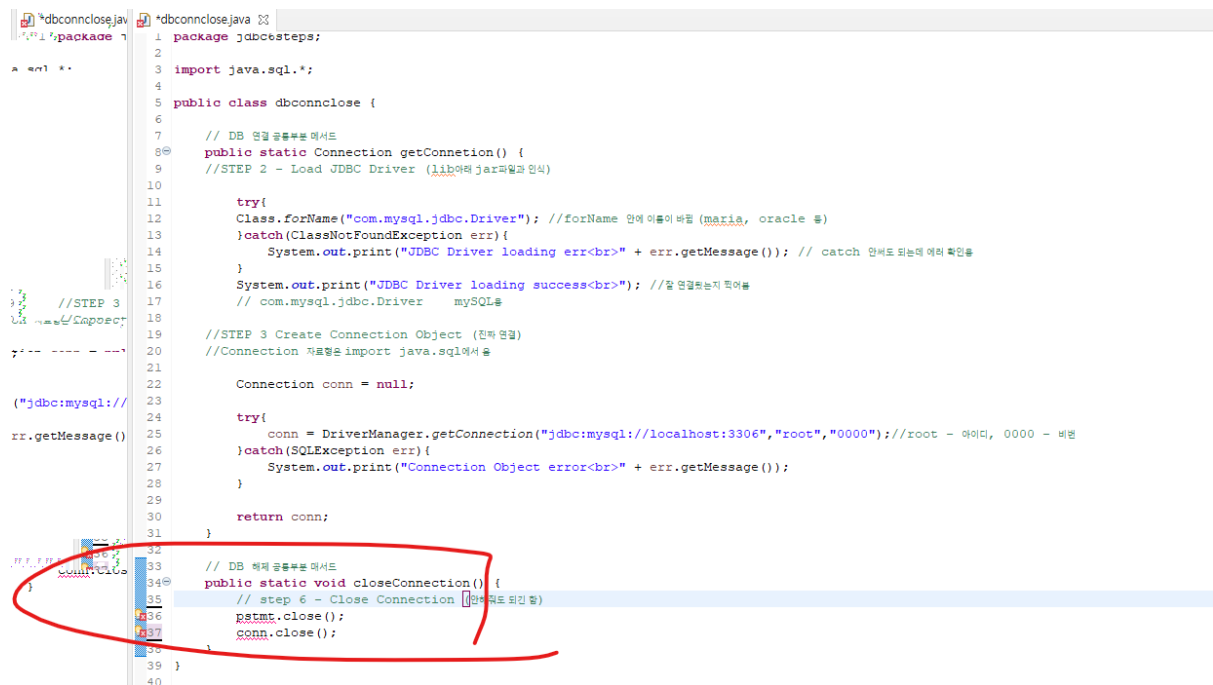
- 공통된 부분 결과파일 (모듈화)

→ DBCreate2.jsp

모듈화 연습

→ DBDrop2

- 오류가남



```
1 package jdbcsteps;
2
3 import java.sql.*;
4
5 public class dbconnclose {
6
7     // DB 연결 공통부분 메서드
8     public static Connection getConnection() {
9         //STEP 2 - Load JDBC Driver (lib아래 jar파일과 일치)
10
11         try{
12             Class.forName("com.mysql.jdbc.Driver"); //forName 안에 이름이 바뀔 (maria, oracle 등)
13         }catch(ClassNotFoundException err){
14             System.out.print("JDBC Driver loading err<br>" + err.getMessage()); // catch 안쳐도 되는데 예외 확인을
15         }
16         System.out.print("JDBC Driver loading success<br>"); //잘 연결되는지 확인을
17         // com.mysql.jdbc.Driver   mySQL용
18
19         //STEP 3 Create Connection Object (전체 연결)
20         //Connection 자료형은 import java.sql에서 올
21
22         Connection conn = null;
23
24         try{
25             conn = DriverManager.getConnection("jdbc:mysql://localhost:3306","root","0000");//root - 아이디, 0000 - 비번
26         }catch(SQLException err){
27             System.out.print("Connection Object error<br>" + err.getMessage());
28         }
29
30         return conn;
31     }
32
33     // DB 해제 공통부분 메서드
34     public static void closeConnection() {
35         // step 6 - Close Connection (여기까지도 되긴 함)
36         pstmt.close();
37         conn.close();
38     }
39 }
40
```

→ 객체를 가져오자

```

32
33 // DB 해제 공통부분 메서드
34 public static void closeConnection(PreparedStatement pstmt, Connection conn) {
35     // step 6 - Close Connection (안해줘도 되긴 함)
36
37     // unhandled exception type SQLException 오류들
38
39     pstmt.close();
40     conn.close();
41 }
42 }
43

```

→ 예외처리 강제됨

```

33 // DB 해제 공통부분 메서드
34 public static void closeConnection(PreparedStatement pstmt, Connection conn) {
35     // step 6 - Close Connection (안해줘도 되긴 함)
36
37     // unhandled exception type SQLException 오류들
38
39     try {
40         pstmt.close();
41         conn.close();
42     } catch (SQLException err) {
43         System.out.println("Error + " + err.getMessage());
44     }
45 }
46 }

```

→ DBCreate3에 dbconnclose 클래스를 끌어와서 구현

(혼자 생각 : step이 나뉘어 있는데 어떻게 적용 시키는 건가, 순서 상관이 없나)

→ 상관이 없이 class에서 메서드를 가져와서 사용하는건데

연습

자바를 임포트 해서 DBDrop3.jsp → 클래스로 test5 데이터베이스 드랍

▼ 공부할것

- jsp:useBean : 자바

8교시

- 다른사람 깃허브 가져가기
 - fork 해서 가져가기
- 이클립스에보기
 - file → clone a repository → clone
- 숙제
 - 영화보고 감상문쓰기(이미테이션게임, 스티븐잡스)