

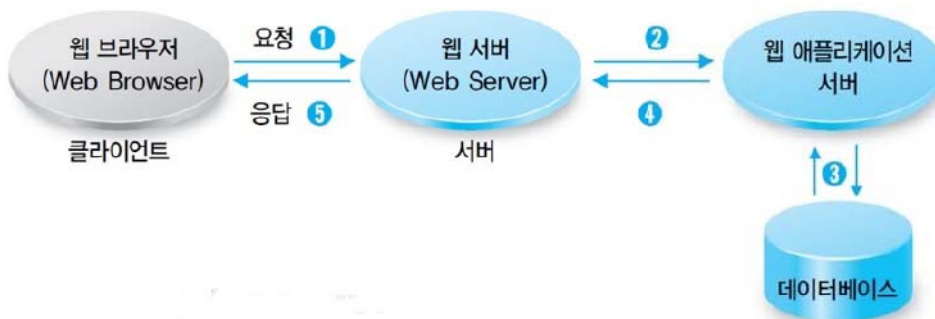
1장. JSP 개요

- Java Server Pages

1. JSP의 특징

- 객체 지향적
- 플랫폼 독립적
- 네트워크 지향적
- 뛰어난 보안성
- 멀티 쓰레드 기능
- 친근한 코드

2. 웹 애플리케이션의 구조 및 처리 순서



JSP와 서블릿을 사용한 웹 서버는 크게 url 주소의 해석을 맡아 주는 http 서버와 서블릿 클래스 또는 jsp 파일의 실행 요청을 처리해주는 웹 컨테이너(웹 애플리케이션 서버)로 구성된다.

http 서버는 단순히 어떤 주소요청이 들어왔을 경우 그 주소에 미리 매핑되어 있는 콘텐츠(html 파일이나 이미지 등)를 사용자의 브라우저에 응답 형태로 전송하는 역할을 한다.

이 때 만일 요청된 URL이 서블릿 클래스 또는 JSP 파일인 경우 HTTP 서버는 이를 웹 컨테이너에서 처리하도록 클라이언트의 요청을 넘겨준다.

웹 컨테이너에서는 요청된 URL에 맞는 서블릿 클래스 또는 JSP 파일을 실행하여 그 결과를 HTTP 서버에 넘겨주고 이는 응답 메시지의 형태로 사용자의 브라우저에 전송된다.

웹 컨테이너는 웹 서버의 내부에서 서블릿 클래스 또는 JSP 파일을 실행하기 위한 실행 환경을 제공하는 역할을 하며 특히 서블릿 클래스에 대한 웹 컨테이너를 서블릿 컨테이너, JSP파일에 대한 웹 컨테이너를 JSP 컨테이너라고 한다.

다만 실제적으로 이 둘을 혼용하여 웹 컨테이너로 통칭하는 경우가 많다.

위의 그림에서 웹 서버는 http 서버를, 웹 애플리케이션 서버는 웹 컨테이너를 나타낸다.

2장. JSP 개발 환경

1. 사용 프로그램

① JDK (Java Development kit, 자바 개발 도구)

- 미국 오라클 사가 제공하는 자바용 소프트웨어 개발 툴.
JDK는 각종 운영 체제 및 응용 프로그램과 연결시킬 수 있는 자바 API와 클래스 라이브러리, 자바 가상 머신 등으로 구성된다.
즉, Java Compiler 및 Java 개발을 위한 여러 라이브러리의 모음이다.
- Java SE로 설치

② JRE (Java Runtime Environment, 자바 런타임 환경)

- 자바는 가상머신에서 돌아가고, 그 가상머신을 돌아가게 하는 환경을 말한다.
- JRE가 설치된 폴더에 들어가면, java.exe 라는 실행파일이 있다.
이 실행파일을 실행하면, 자바가상머신이 메모리에 올라오고 그 가상머신이 자바프로그램을 실행시킨다.
엄밀히 따지면 JRE가 자바가상머신은 아니고, 자바가상머신이 도는 환경이다.
- JRE에 자바파일을 컴파일 하는 도구가 포함된 것이 JDK(자바개발환경)이다.
- 일반적으로 웹에서 자바프로그램이 돌때 사용자 컴퓨터에 설치되는 것이 JRE이다.

③ eclipse => 새로 프로그램을 만들 때

- 오픈소스 기반의 IDE(통합 개발 환경) 중 하나로 jsp 개발 툴로 가장 많이 사용된다.

④ Oracle

- 데이터베이스 프로그램

⑤ sqldeveloper

- 오라클 데이터베이스 보조 프로그램

⑥ apache-tomcat

- Apache : 서버 프로그램
- Tomcat : Servlet 이나 JSP의 컨테이너

2. JDK (Java Development kit), JRE 설치

(1) 작업폴더 생성

- ① D:\JSP\
 - ② D:\JSP\util\ => JSP 관련 프로그램 저장
 - ③ D:\JSP\workspace\ => 작업 파일 저장

(2) 프로그램 다운받기

- <http://www.oracle.com> 접속
- downloads -> Java SE -> Windows x64 (64bit os용) 다운로드 받기
또는 Windows x86 (32bit os용) 다운로드 받기

(3) JDK 설치

- jdk-8u91-windows-x64.exe 파일 실행 (JDK, JRE 설치 프로그램)
- C:\Program Files\Java\jdk1.8.0_91\ => 폴더이름 변경없이 설치
- 설치 완료후, JDK, JRE 설치 확인 (한꺼번에 같이 설치된다.)
C:\Program Files\Java\jdk1.8.0_91
C:\Program Files\Java\jre1.8.0_91

(4) 환경변수 설정

- Windows 8.1에서의 설정
윈도우 탐색기 -> [메뉴] 컴퓨터 -> 시스템 속성 -> 고급 시스템 설정
-> [시스템 속성 창] [메뉴] “고급” 클릭 -> “환경변수” 클릭

1) JAVA_HOME 설정

- [환경변수 창][시스템 변수] “새로 만들기” 클릭

변수이름 : JAVA_HOME

변수 값 : C:\Program Files\Java\jdk1.8.0_91

=> 확인

2) path 경로 설정

- [환경변수 창][시스템 변수] “Path” 선택 -> “편집” 클릭
- 편집창의 제일 뒤에 “;%JAVA_HOME%\bin”을 추가한다. (;%이 각 경로의 구분자이다.)

(5) JDK 설치 확인

- Winkey + R -> cmd 실행
=> javac -version 입력 후 실행
=> 아래의 내용이 나오면 설정 완료

C:\Users\ygs_000>javac -version

javac 1.8.0_91

C:\Users\ygs_000>

3. eclipse 설치

(1) 프로그램 다운받기

- <http://www.eclipse.org> 접속
- downloads
 - > Eclipse IDE for Java EE Developers
 - > Windows 64 Bit 또는 Windows 32 Bit 클릭
 - > eclipse-jee-mars-2-win32-x86_64.zip를 저장한다.

(2) eclipse 설치

- eclipse-jee-mars-2-win32-x86_64.zip의 압축을 푼다.
- D:\JSP\ 폴더에 eclipse폴더를 복사해서 붙여 놓는다.
- D:\JSP\eclipse 생성 및 복사 완료
- D:\JSP\eclipse\eclipse.exe를 실행시켜서 사용하면 된다.

(3) eclipse 작업 경로 설정

- 이클립스로 작성하는 내용이 저장될 경로를 지정한다.
- D:\JSP\workspace로 설정한다.

(4) eclipse 환경 설정

- [메뉴] window -> 맨마지막 Preferences

1) 글꼴변경

General -> Appearance -> Colors and Fonts -> Basic
Text Font => 오른쪽에 있는 Edit...버튼클릭 글꼴변경

2) 라인번호 보여주기

General -> Editors -> Text Editors => Show line numbers 체크

3) 인코딩 설정

- ① General -> Workspace -> "Text file encoding" -> Other : UTF-8
- ② Web -> CSS Files -> Encoding : ISO 10646/Unicode(UTF-8)
- ③ Web -> HTML Files -> Encoding : ISO 10646/Unicode(UTF-8)
- ④ Web -> JSP Files -> Encoding : ISO 10646/Unicode(UTF-8)

(5) eclipse 단축키

Ctrl + Shift + O : 자동으로 Import 시키기

Ctrl + Shift + Space : 메소드의 파라미터 목록 보기

(메소드의 괄호 안에 커서를 놓고 이 키를 누르면 파라미터 타입 힌트를 볼 수 있음)

Ctrl + Space : 문장 자동 완성

=> sysout -> ctrl + space : System.out.println(); 문장 자동 완성

Ctrl + Shift + F : 코드 자동 정리

Ctrl + Shift + / : 블록 주석(/ * */)

Ctrl + Shift + \ : 블록 주석 제거

Ctrl + / : 여러 줄이 한꺼번에 주석 처리됨 (주석 해제하려면 반대로 하면 됨)

Ctrl + 1 : 컴파일 에러가 발생한 곳에서 Ctrl + 1을 누를 경우 컴파일 에러에 대한 해결책을 제시

Shift + Alt + R : Source 화면 빨리가기

F2 : 컴파일 에러의 빨간 줄에 커서를 가져다가 이 키를 누르면 에러의 원인에 대한 힌트를 제공

3-1. eclipse 사용 방법 (JAVA)

(1) 이클립스 : 프로젝트 만들기

1) File -> new -> Java Project => 선택

2) Project name : Hello

=> finish

(2) 이클립스 : class 만들기

1) src 폴더 -> new -> class => 선택

2) Name : Exam1

☐ public static void main(String[] args) => 체크

=> finish

=> Exam1.java 생성

(3) 실행하기

- 단축키 : Ctrl + F11

- "Run" 버튼 누르기.

4. Oracle Express 11g 설치

- www.oracle.com
- 회원가입해야만 다운 받을 수 있음
- Enterprise (서버용)
- Client
- Express → OracleXE112_Win64.zip
- 컴퓨터이름이나 계정이 모두 영문(한글X), 공백X, 반드시 관리자계정이어야 한다.

4-1. 설치

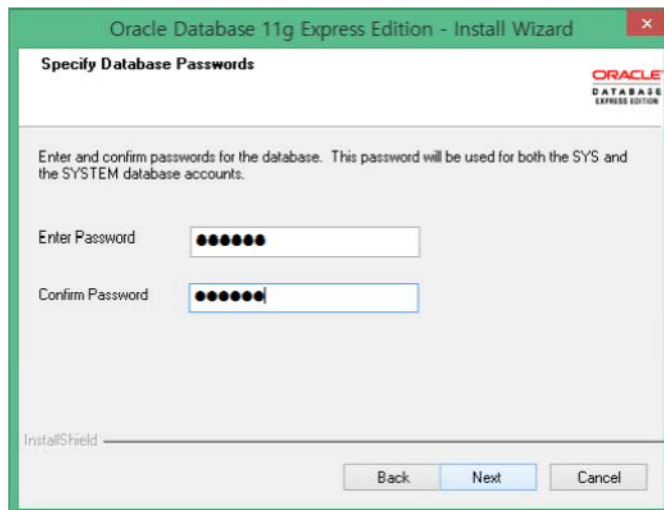
OracleXE112_Win64.zip 압축풀기



setup.exe 더블클릭



암호 : 『oracle』 입력 (SID : xe 기본설정)



Destination Folder: C:\oraclexe\

Oracle Home: C:\oraclexe\app\oracle\product\11.2.0\server\

Oracle Base:C:\oraclexe\

Port for 'Oracle Database Listener': 1521

Port for 'Oracle Services for Microsoft Transaction Server': 2030

Port for 'Oracle HTTP Listener': 8080

→ Tomcat 8080 : 충돌남.

따라서, HTTP : 9090으로 바꾸어 사용함.



설치



마침

4-2. 확인

① 시작 → 제어판 → 시스템 및 보안 → 관리도구 → 서비스

OracleServiceXE	시작됨	자동
OracleXETNSListener	시작됨	자동

4-3. Listener 확인

- OracleXETNSListener 데이터를 주고 받을 수 있게하는 서비스이다. (데이터송수신)

- 위치

C:\oracle\app\oracle\product\11.2.0\server\network\ADMIN\

listener.ora

tnsnames.ora

: (HOST = pc이름)(PORT = 1521) 인지 확인해봐야한다.

① listener.ora

클라이언트가 오라클 서버에 접속하기 위해서 서버 컴퓨터에 하는 설정

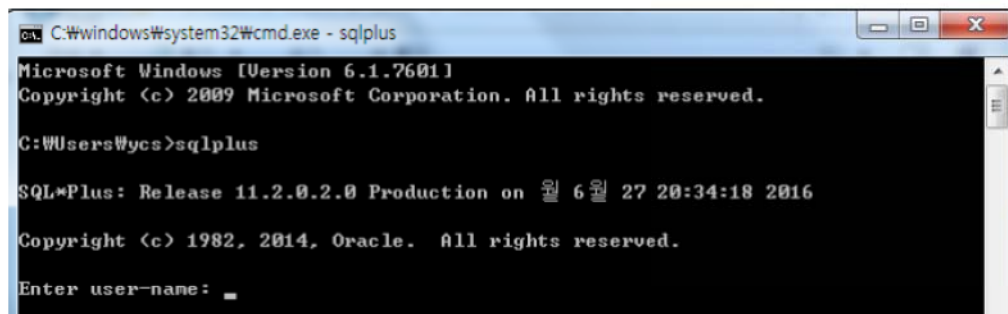
② tnsnames.ora

클라이언트에서 오라클 서버에 접속할 때 필요한 설정

4-4. 동작 확인

- 시작 -> cmd -> sqlplus 엔터

- 아래와 같이 표시되면 제대로 설치된 것이다.



```
C:\Windows\system32\cmd.exe - sqlplus
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Wycs>sqlplus

SQL*Plus: Release 11.2.0.2.0 Production on 월 6월 27 20:34:18 2016

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Enter user-name: _
```

5. sqldeveloper 설치 - 64bit

- www.oracle.com

- 회원가입해야만 다운 받을 수 있음

- sqldeveloper-4.1.3.20.78-x64.zip

- JSP 폴더에 sqldeveloper 폴더를 압축을 풀고 저장한다.

3장. Database

1. SQLPLUS에서 DB 사용하기

1-1. SQLPLUS 시작하기

(1) 시작 → cmd

C:\Users\ygs> sqlplus

사용자명 입력: system

암호 입력: oracle (입력되는 글자가 안보인다)

```
C:\Users\ygs>sqlplus
SQL*Plus: Release 11.2.0.2.0 Production on 토 2월 6 06:08:47 2016
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Enter user-name: system
Enter password:
Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
SQL>
```

(2) 시작 → 모든 프로그램 → Oracle Database 11g Express Edition → Run SQL Command Line

```
Run SQL Command Line
SQL*Plus: Release 11.2.0.2.0 Production on 토 2월 6 10:04:50 2016
Copyright (c) 1982, 2014, Oracle. All rights reserved.
SQL> conn system/oracle
Connected.
SQL>
```

(3) Oracle Database 11g Express Edition 설치후, 암호를 설정하기

SQL> alter user 계정 identified by 암호;

C:\Users\ygs> sqlplus "/ as sysdba"

SQL> alter user system identified by oracle;

```
C:\Users\ygs>sqlplus "/ as sysdba"
SQL*Plus: Release 11.2.0.2.0 Production on 토 2월 6 06:11:40 2016
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
SQL> alter user system identified by oracle;
User altered.
SQL>
```


1-2. OracleXE 11g port 변경

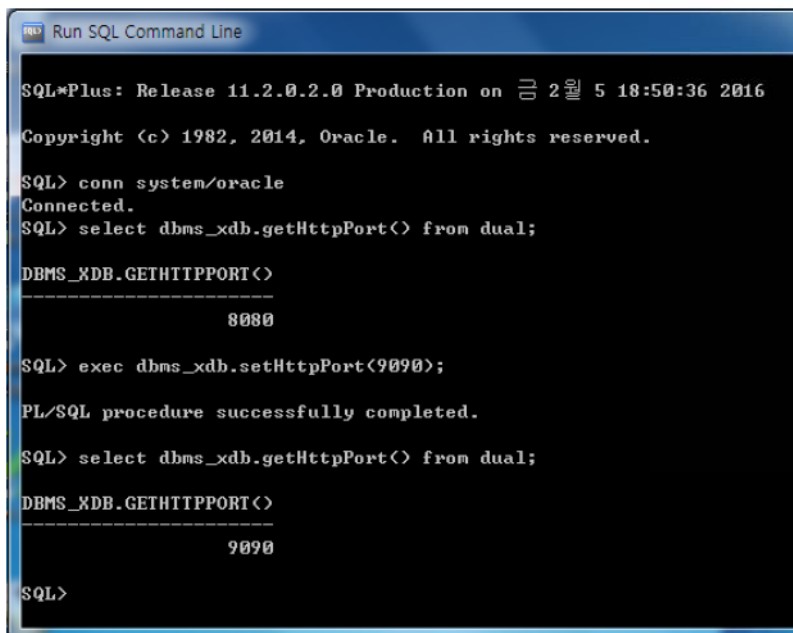
- 반드시 관리자 계정(sys, system)에서 해야한다.
- Apache Tomcat과 port 충돌을 피하기 위해서 변경한다.
=> Apache Tomcat port : 8080

SQL> conn system/oracle

Connected.

SQL> select dbms_xdb.getHttpPort() from dual; ← 현재 포트 확인

SQL> exec dbms_xdb.setHttpPort(9090); ← 8080에서 9090으로 변경
PL/SQL 처리가 정상적으로 완료되었습니다.



```
Run SQL Command Line

SQL*Plus: Release 11.2.0.2.0 Production on 금 2월 5 18:50:36 2016

Copyright (c) 1982, 2014, Oracle. All rights reserved.

SQL> conn system/oracle
Connected.
SQL> select dbms_xdb.getHttpPort() from dual;

DBMS_XDB.GETHTTTPORT()
-----
                8080

SQL> exec dbms_xdb.setHttpPort(9090);

PL/SQL procedure successfully completed.

SQL> select dbms_xdb.getHttpPort() from dual;

DBMS_XDB.GETHTTTPORT()
-----
                9090

SQL>
```

1-3. 사용자 계정

- sys : 오라클 Super 사용자 계정이며, 데이터베이스에서 발생하는 모든 문제들을 처리할 수 있는 권한을 가지고 있다.
- system : 오라클 데이터베이스를 유지보수 및 관리할 때 사용하는 사용자 계정.
- hr : 오라클에 접근할 수 있도록 샘플로 만들어 놓은 사용자 계정

2. sqldeveloper에서 DB 사용하기

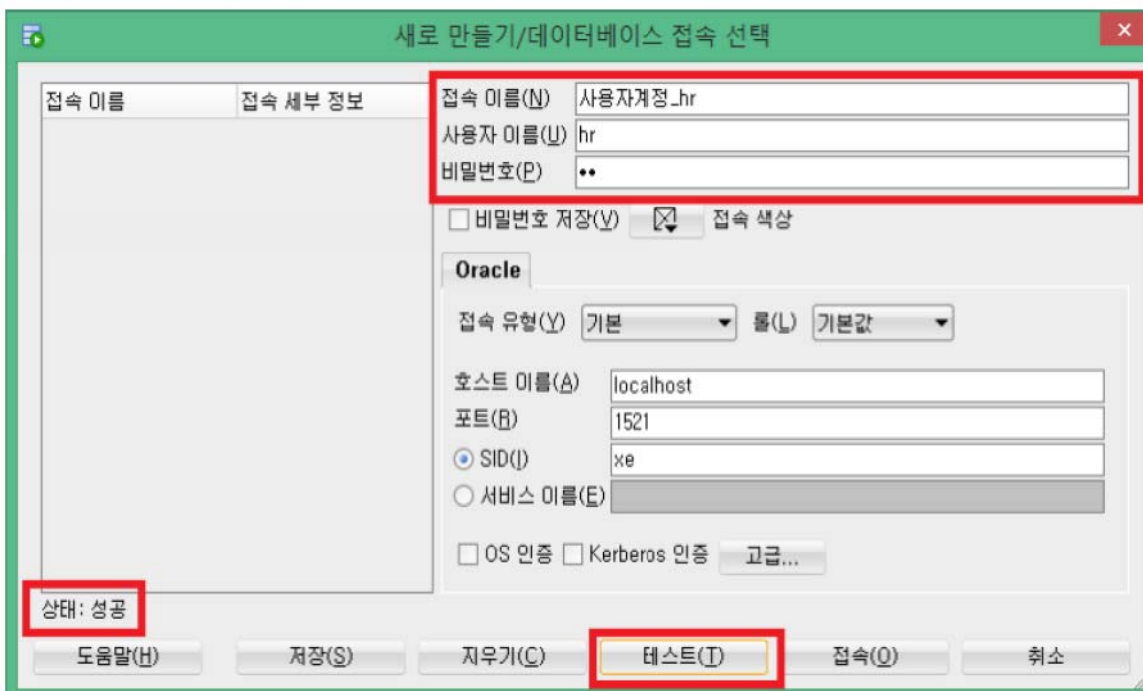
(1) D:\JSP\sqldeveloper\sqldeveloper.exe을 실행한다.

(2) + 버튼 클릭



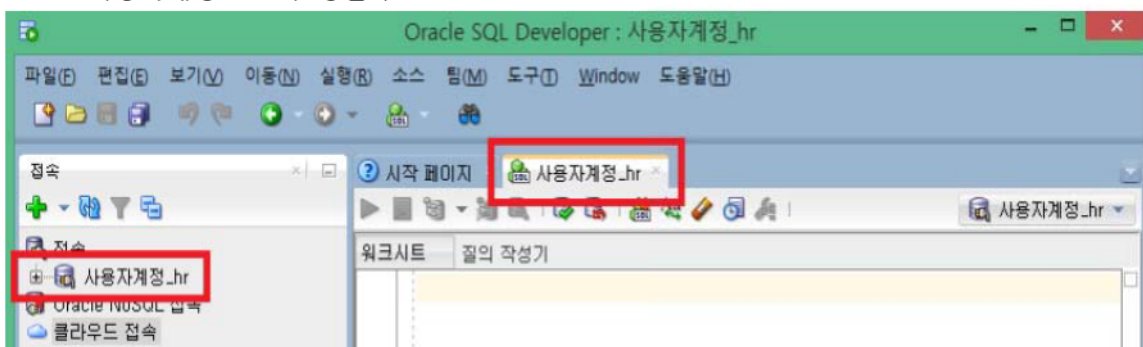
(3) 접속 이름, 사용자 이름, 비밀번호를 입력한다.

(4) 테스트 버튼을 클릭한 후, “상태 : 성공” 이 되는 지 확인한다.



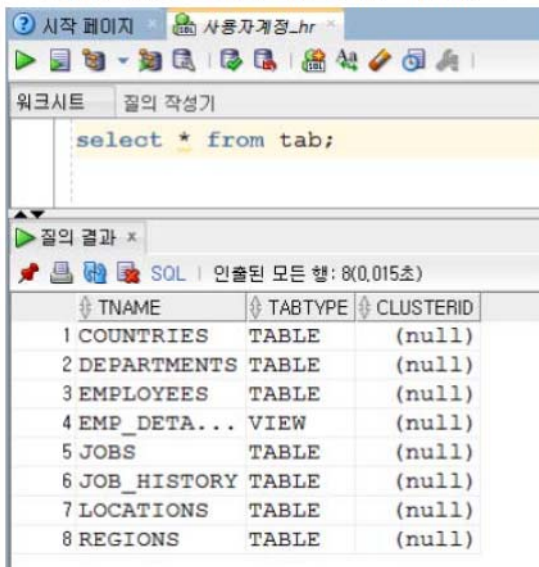
(5) 접속 버튼을 클릭한다.

=> “사용자계정_hr”이 생긴다.



(6) 명령어로 데이터베이스를 확인한다.

- 커서를 명령어에 위치 시키고, 실행버튼을 누른다.
- 또는, ctrl + enter 단축키를 사용한다.



3. SQL (Structured Query Language)

3-1. SQL (Structured Query Language)

- 구조화된 질의 언어
- 데이터베이스에 저장된 데이터를 조회, 입력, 수정, 삭제하는 등의 조작이나, 테이블을 비롯한 다양한 객체를 생성 및 제어한다.
- 오라클을 배운다는 것은 SQL을 배우는 것이라고 할 수 있다.

3-2. 사용 용도에 따른 SQL

유형		명령문
DQL	Data Query Language 데이터 질의어	SELECT : 데이터 검색 시 사용
DDL	Data Definition Language 데이터 정의어 : 객체 생성 및 변경	CREATE : 데이터베이스 생성 ALTER : 데이터베이스 변경 DROP : 데이터베이스 삭제 RENAME : 데이터베이스 객체 이름 변경 TRUNCATE : 데이터베이스 저장 공간 삭제
DML	Data Manipulation Language 데이터 조작어 : 데이터 변경	INSERT : 데이터 입력 UPDATE : 데이터 수정 DELETE : 데이터 삭제
DCL	Data Control Language 데이터 제어어	GRANT : 데이터베이스에 대한 일련의 권한 부여 REVOKE : 데이터베이스에 대한 일련의 권한 취소
TCL	Transaction Control Language 트랜잭션 처리어	COMMIT : 트랜잭션의 정상적인 종료 처리 ROLLBACK : 트랜잭션 취소 SAVEPOINT : 트랜잭션 내에 임시 저장점 설정

3-3. 계정 만들기

- 반드시 관리자 계정에서 해야한다. (system 계정)

- 사용자 계정 생성하기

SQL> create user 계정 identified by 암호;

- 사용자 계정에 권한 부여하기

SQL> grant create session, create table, create sequence, create view to 계정;

- 사용자 계정의 테이블스페이스를 users로 설정하기

SQL> alter user 계정 default tablespace users;

- 사용자 계정에 테이블 스페이스 쿼터 할당하기

SQL> alter user 계정 quota unlimited on users;

(1) jsp 계정만들기 (cmd창)

(2) jsp 계정만들기 (sqldeveloper)

3-4. 계정 삭제

- 반드시 관리자 계정에서 해야한다

SQL> drop user 계정명;

: 계정에 데이터(테이블)가 없을 경우

SQL> drop user 계정명 cascade;

: 계정에 데이터(테이블)가 있을 경우

(1) jsp 계정 없애기

4. CRUD (CREATE, READ, UPDATE, DELETE)

4-1. 테이블 생성

create table 테이블명(필드명 필드타입, 필드명 필드타입,...);
: 테이블명 한글 가능

문자 / 문자열	char(고정형)	varchar2(가변형)
숫자	int(고정형)	number(가변형)
정수		number, number(5)
실수		number(10, 2) => 10 : 소수점을 포함한 전체자릿수 2 : 소수점 자릿수
날짜/시간	date	
Express	한글 1자 = 3byte	

(1) 테이블 생성

- sqldeveloper에서 연습하기
- 사용자계정_jsp 만들기 : jspexam / m1234
- 테이블 만들기
- 테이블 확인

(2) 테이블 항목 확인

desc 테이블명;

4-2. 테이블 삭제 / 복원

drop table 테이블명;
drop table 테이블명 purge; //휴지통을 거치지않고 바로 삭제

flashback table 테이블명 to before drop; //복원
purge recyclebin //휴지통비우기

4-3. 레코드 추가

- 테이블에 데이터 추가하기
- insert into 테이블명(필드명,...) values(값,...);
insert into 테이블명 values(값,...); ← 테이블의 필드순서가 일치하면 필드명은 생략가능

4-4. 레코드 추출 (검색)

- 데이터 확인하기

① select 필드명, 필드명... from 테이블명;

select * from 테이블명;

select * from 테이블명 order by 필드명 asc; → 오름차순(asc는 생략가능)

select * from 테이블명 order by 필드명 desc; → 내림차순

select * from 테이블명 order by 필드명 asc, 필드명 desc;

→ 첫번째 필드에 똑같은 데이터가 존재하면 두번째 필드로 정렬한다.

select count(age) from 테이블명; → 필드명 age의 레코드 개수 구하기

select count(*) from 테이블명; → 모든 레코드 개수 구하기

② select * from 테이블명 where 조건;

- 조건에서 '문자'는 대소문자를 가린다

4-5. 트랜잭션

: commit을 하기 전까지의 모든 명령어

① commit - 갱신

② rollback

- 취소

- commit한 이후의 시점으로 돌아간다

- 현재 트랜잭션이 취소된다.

4-6. 레코드 수정

update 테이블명 set 수정할 내용 where 조건

예제1. 나이가 없는 곳에 나이를 100으로 수정하기

4-7. 레코드 삭제

delete 테이블명; // 모든 레코드 삭제

delete 테이블명 where 조건; // 조건에 맞는 레코드 삭제

예제2. 3번째 글자가 '홍'인 레코드를 찾아서 삭제하기

(연습문제)

1. department 테이블

```
CREATE TABLE department (  
    deptno number NOT NULL ,      -- 부서번호(학과번호)  
    dname varchar2(25) NOT NULL , -- 부서명(학과명)  
    loc varchar2(10) DEFAULT NULL -- 위치  
);
```

```
INSERT INTO department VALUES (101, '컴퓨터공학과', '1호관');  
INSERT INTO department VALUES (102, '멀티미디어학과', '2호관');  
INSERT INTO department VALUES (201, '전자공학과', '3호관');  
INSERT INTO department VALUES (202, '기계공학과', '4호관');
```

2. professor 테이블

```
CREATE TABLE professor (  
    profno number NOT NULL,      -- 교수번호  
    name varchar2(20) NOT NULL,  -- 이름  
    userid varchar2(10) NOT NULL, -- 아이디  
    position varchar2(20) NOT NULL, -- 직급  
    sal number NOT NULL,         -- 급여  
    hiredate date NOT NULL,      -- 입사일  
    comm number DEFAULT NULL,    -- 보직수당  
    deptno number NOT NULL,      -- 부서번호(학과번호)  
    PRIMARY KEY (profno));
```

```
INSERT INTO professor VALUES  
(9901, '김도훈', 'capool', '교수', 500,  
To_Date('2010.12.08 16:33', 'yyyy.mm.dd hh24:mi'), 20, 101);  
INSERT INTO professor VALUES  
(9902, '이재우', 'sweat413', '조교수', 320,  
To_Date('1995.04.12 00:00:00', 'yyyy.mm.dd hh24:mi:ss'), null, 201);  
INSERT INTO professor VALUES  
(9903, '성연희', 'pascal', '조교수', 360,  
To_Date('1993.03.17', 'YYYY.MM.DD'), 15, 101);  
INSERT INTO professor VALUES  
(9904, '염일웅', 'blue77', '전임강사', 240,  
To_Date('1998-10-11 00:00:00', 'yyyy-mm-dd hh24:mi:ss'), null, 102);  
INSERT INTO professor VALUES  
(9905, '권혁일', 'refresh', '교수', 450,  
To_Date('1986-02-11 00:00:00', 'yyyy-mm-dd hh24:mi:ss'), 25, 102);  
INSERT INTO professor VALUES  
(9906, '이만식', 'pocari', '부교수', 420,  
To_Date('1988-07-11 00:00:00', 'yyyy-mm-dd hh24:mi:ss'), null, 101);  
INSERT INTO professor VALUES  
(9907, '전은지', 'totoro', '전임강사', 210,  
To_Date('2001-05-11 00:00:00', 'yyyy-mm-dd hh24:mi:ss'), null, 101);  
INSERT INTO professor VALUES  
(9908, '남은혁', 'bird13', '부교수', 400,  
To_Date('1990-10-18 00:00:00', 'yyyy-mm-dd hh24:mi:ss'), 17, 202);
```

3. student 테이블

```

CREATE TABLE student (
    studno number NOT NULL,      -- 학생번호
    name varchar2(20) NOT NULL,  -- 이름
    userid varchar2(10) NOT NULL, -- 아이디
    grade number NOT NULL,       -- 학년
    idnum varchar2(13) NOT NULL,  -- 주민번호
    birthdate date NOT NULL,     -- 생년월일
    tel varchar2(15) NOT NULL,   -- 전화번호
    height number NOT NULL,      -- 키
    weight number NOT NULL,      -- 몸무게
    deptno number NOT NULL,      -- 학과번호
    profno number DEFAULT NULL,  -- 담당교수의 일련번호
    PRIMARY KEY (studno)) ;

INSERT INTO student VALUES
(10101, '전인하', 'jun123', 4, '7907021369824', To_Date('1979-07-02', 'yyyy-mm-dd'),
'051)781-2158', 176, 72, 101, 9903);
INSERT INTO student VALUES
(10102, '박미경', 'ansel414', 1, '8405162123648', To_Date('1984-05-16', 'yyyy-mm-dd'),
'055)261-8947', 168, 52, 101, null);
INSERT INTO student VALUES
(10103, '김영균', 'mandu', 3, '8103211063421', To_Date('1981-03-21', 'yyyy-mm-dd'),
'051)824-9637', 170, 88, 101, 9906);
INSERT INTO student VALUES
(10104, '지은경', 'gomo00', 2, '8004122298371', To_Date('1980-04-12', 'yyyy-mm-dd'),
'055)418-9627', 161, 42, 101, 9907);
INSERT INTO student VALUES
(10105, '임유진', 'youjin12', 2, '8301212196482', To_Date('1983-01-21', 'yyyy-mm-dd'),
'02)312-9838', 171, 54, 101, 9907);
INSERT INTO student VALUES
(10106, '서재진', 'seolly', 1, '8511291186273', To_Date('1985-11-29', 'yyyy-mm-dd'),
'051)239-4861', 186, 72, 101, null);
INSERT INTO student VALUES
(10107, '이광훈', 'huriky', 4, '8109131276431', To_Date('1981-09-13', 'yyyy-mm-dd'),
'055)736-4981', 175, 92, 101, 9903);
INSERT INTO student VALUES
(10108, '류민정', 'cleansky', 2, '8108192157498', To_Date('1981-08-19', 'yyyy-mm-dd'),
'055)248-3679', 162, 72, 101, 9907);
INSERT INTO student VALUES
(10201, '김진영', 'simply', 2, '8206062186327', To_Date('1982-06-06', 'yyyy-mm-dd'),
'055)419-6328', 164, 48, 102, 9905);
INSERT INTO student VALUES
(10202, '오유석', 'yousuk', 4, '7709121128379', To_Date('1977-09-12', 'yyyy-mm-dd'),
'051)724-9618', 177, 92, 102, 9905);
INSERT INTO student VALUES
(10203, '하나리', 'hanal', 1, '8501092378641', To_Date('1985-01-09', 'yyyy-mm-dd'),
'055)296-3784', 160, 68, 102, null);
INSERT INTO student VALUES
(10204, '윤진욱', 'samba/', 3, '79040213586/1', To_Date('1979-04-02', 'yyyy-mm-dd'),
'053)487-2698', 171, 70, 102, 9905);
INSERT INTO student VALUES
(20101, '이동훈', 'dals', 1, '8312101128467', To_Date('1983-12-10', 'yyyy-mm-dd'),
'055)426-1752', 172, 64, 201, null);
INSERT INTO student VALUES
(20102, '박동진', 'ping2', 1, '8511241639826', To_Date('1985-11-24', 'yyyy-mm-dd'),
'051)742-6384', 182, 70, 201, null);
INSERT INTO student VALUES
(20103, '김진경', 'lovely', 2, '8302282169387', To_Date('1983-02-28', 'yyyy-mm-dd'),
'052)175-3941', 166, 51, 201, 9902);
INSERT INTO student VALUES
(20104, '조명훈', 'rader214', 1, '8412141254963', To_Date('1984-12-14', 'yyyy-mm-dd'),
'02)785-6984', 184, 62, 201, null);

```

문제1. 학생 테이블(student)의 학생이름(name)과 학생번호(studno)를 출력하여라.

문제2. 단일칼럼에서 DISTINCT(검색결과 중복 제거 옵션) 키워드 사용
-학생 테이블에서 중복되는 행을 제외한 학과번호 출력

문제3. 학과테이블에서 학과이름 칼럼의 별명은 dept_name, 학과번호 칼럼의 별명은 DN 으로
부여하여 출력

문제4. department 테이블을 사용하여 deptno 를 학과,
dname 를 학과명 , loc 를 위치로 별명을 설정하여 출력하세요.

문제5. 교수 테이블(professor)에서 교수이름(name), 급여(sal), 보너스를 포함한 연봉을
출력하여라.
단 보너스를 포함한 연봉은 급여*12를 한 결과에 보너스 100을 더한 값으로 계산한다.

문제6. 학생 테이블에서 1학년 학생만 검색하여 학번, 이름, 학과 번호를 출력

문제7. 학생테이블에서 학과번호가 101번인 학생들의 학번 , 이름, 학년을 출력

문제8. 학생 테이블에서 몸무게가 70kg 이하인 학생만 검색하여
학번, 이름, 학년, 학과번호, 몸무게를 출력

문제9. 학생테이블에서 키가 170 이상인 학생의 학번, 이름, 학년, 학과번호, 키를 출력하여라.

문제10. 학생 테이블에서 1학년이면서 몸무게가 70kg 이상인 학생만 검색하여
이름, 학년, 몸무게, 학과 번호를 출력

문제11. 학생 테이블에서 1학년 이거나 몸무게가 70kg 이상인 학생만 검색하여
이름, 학번, 학년, 몸무게, 학과 번호를 출력하여라.

문제12. 학생 테이블에서 성이 '김'씨인 학생의 이름, 학년, 학과번호를 출력

문제13. 학생 테이블에서 이름을 가나다 순으로 정렬하여 이름, 학년, 전화번호를 출력

문제14. 학생 테이블에서 101번 학과에 소속된 학생들의 이름, 학년, 학과번호를
생년월일이 빠른 순으로 출력하여라.

문제15. 학생 테이블에 아래의 결과처럼 '홍길동' 학생의 데이터 입력하기

STUDNO	NAME	USERID	GRADE	IDNUM	BIRTHDATE	TEL	HEIGHT	WEIGHT	DEPTNO	PROFNO
10110	홍길동	hgd85	1	8501011143098	13/10/01	031)-777-7777	170	70	101	9903

문제16. 교수 테이블에서 다음의 데이터를 입력

=====

교수번호: 9921 이름: 홍길동
아이디: hong 직급: 조교
급여: 230 입사일: 오늘날짜
학과: 102번 학과

=====

문제17. 교수 번호가 9903인 직원의 현재 직급을 '교수'로 수정하고 급여를 400으로 수정

문제18. 학생테이블에서 학번이 20103인 학생의 데이터를 삭제

4장. JDBC (Java DataBase Connectivity)

- java에서 사용하기

1. JDBC 준비

(1) 프로젝트에 라이브러리를 등록하는 방법

- 아래의 실습 참조

(2) JRE에 직접 라이브러리를 복사해 놓는 방법

- JRE에게 복사 (무식한 방법)

① 아래의 폴더위치에서 ojdbc6.jar를 복사한다.

- C:\oracle\app\oracle\product\11.2.0\server\jdbc\lib

② ojdbc6.jar 라이브러리를 복사해 JRE에 붙여넣는다.

C:\Program Files\Java\jdk1.8.0_45\jre\lib\ext

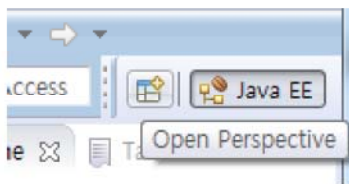
or

C:\Program Files\Java\jre1.8.0_45\lib\ext

(3) Eclipse에서 Java 준비하기

① Eclipse 실행한다.

② Perspective를 연다.



③ Java 선택 -> OK 클릭 -> Java 추가됨



2. JDBC 작업단계

① Driver Loading - 1번 (*.jar이 필요)

- 데이터베이스와 연결하는 드라이버 파일을 찾는다.

② Connection - URL, ID, PWD

- 연결을 관리하는 Connection 객체를 생성한다.

③ SQL 작업 처리 - insert, delete, update, select...

- 작업을 처리할 Statement or PreparedStatement or CallableStatement 객체를 생성한다.

④ 반환되는 결과는 int나 ResultSet 객체에 담는다.

⑤ 접속을 종료한다.

3. 사용 클래스

(1) Statement

: 2개의 Subinterfaces (CallableStatement, PreparedStatement)

: 실행시 sql문장을 줘야한다.

```
String sql = "완성된 문장-데이터값이 입력된 상태";  
stmt = conn.createStatement(); //생성  
stmt.executeUpdate(sql);          stmt.executeQuery(sql); //실행-sql문장을 입력해야 한다
```

(2) PreparedStatement

: 생성시 sql문장을 줘야한다.

```
String sql = "미완성된 문장-데이터값을 쓰는 대신 ?로 대치";  
pstmt = conn.prepareStatement(sql); //생성-sql문장을 입력해야 한다  
?에 데이터 값을 넣어주고(데이터 매핑)  
pstmt.executeUpdate();          pstmt.executeQuery(); //실행
```

(3) interface

① implements

모든 추상메소드 Override

② 대신 Override 클래스

```
Collection coll = new ArrayList();
```


예제1. JDBC 준비 (ojdbc6.jar 등록하기)

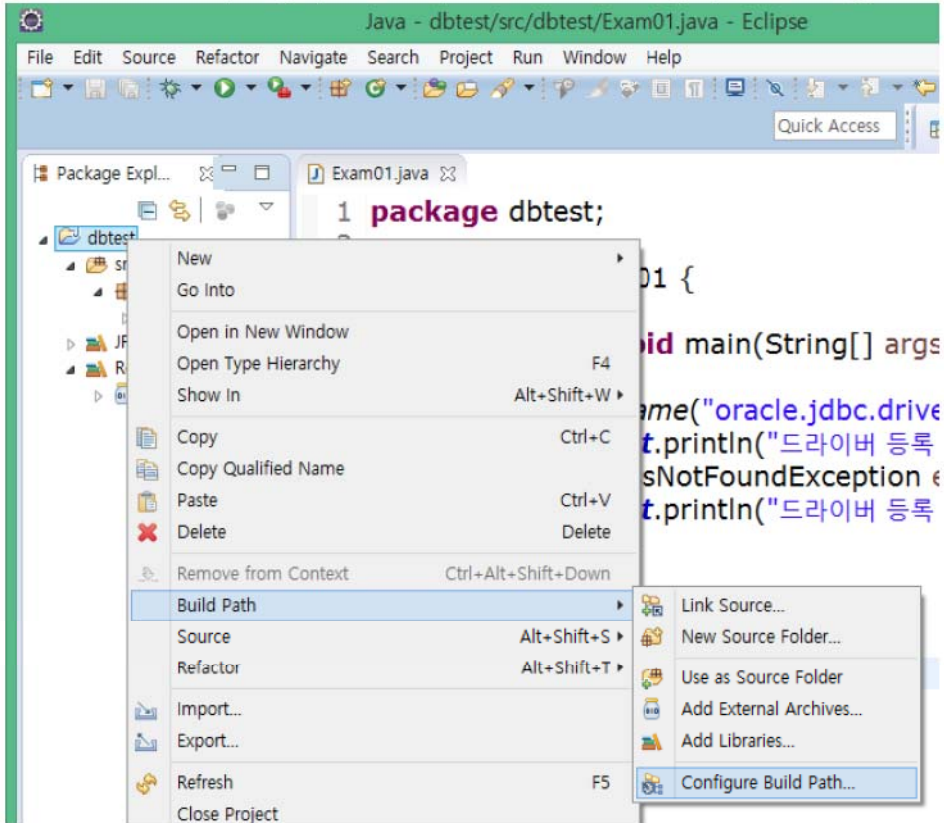
Java Project : 4-dbtest

Package : com.dao

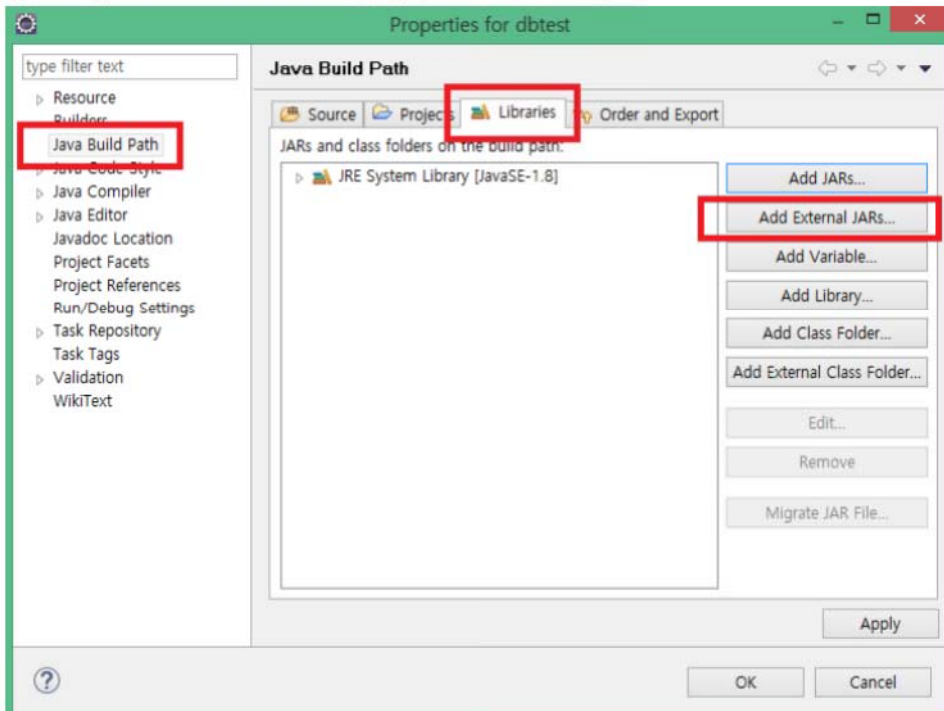
Class File : Exam1.java

- ojdbc6.jar 등록하기

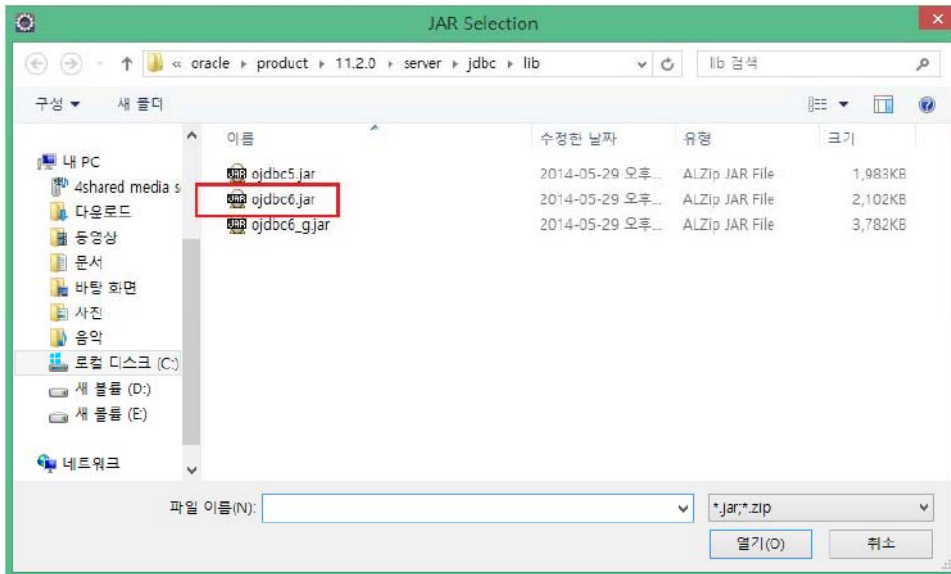
① "dbtest" 프로젝트 선택 -> 오버 -> "Build Path" 클릭 -> "Configure Build Path" 클릭



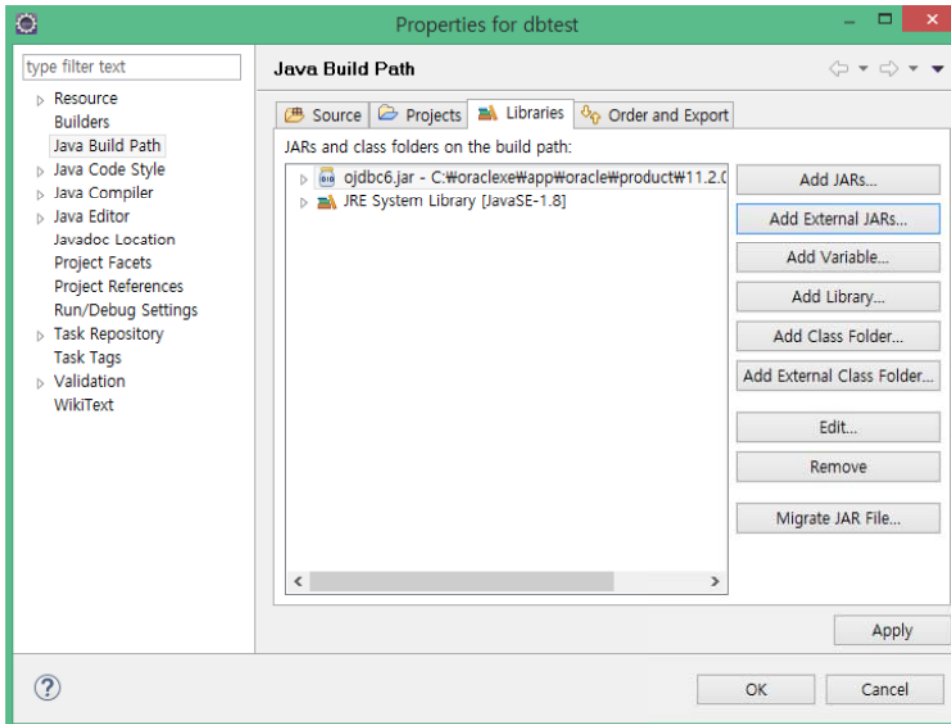
② "Properties for dbtest" 창에서 "Libraries" 탭 선택 -> "Add External JARs..." 버튼 클릭



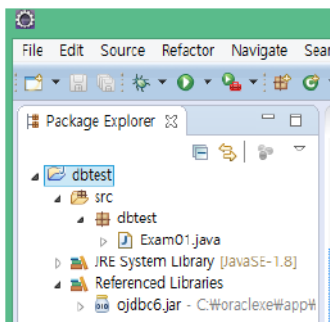
- ③ "JAR Selection" 창에서 "C:\oracle\app\oracle\product\11.2.0\server\jdbc\lib" 폴더로 이동한다. -> "ojdbc6.jar"를 선택한다.



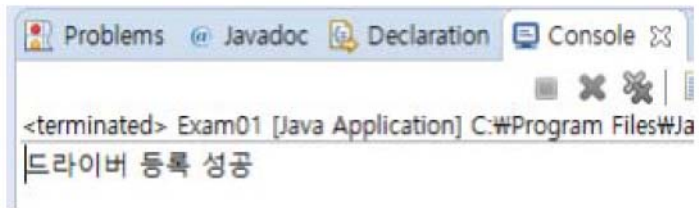
- ④ "Properties for dbtest" 창에서 "ojdbc6.jar"가 등록된 것을 확인하고, "OK"를 클릭한다.



- ⑤ "dbtest" 프로젝트에 추가된 것을 확인할 수 있다.



<실행 결과>



예제2. dbtest 테이블에 레코드 추가

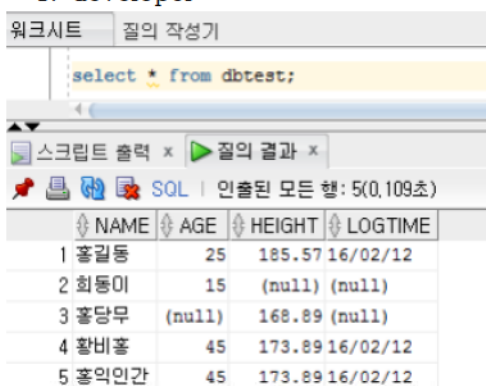
Java Project : 4-dbtest
Package : com.dao
Class File : Exam2.java

- dbtest 테이블을 먼저 만들어야 한다.

```
> create table dbtest(  
name varchar2(15),  
age number,  
height number(10, 2),  
logtime date);
```

<실행 결과>

- 1. developer



워크시트 | 질의 작성기

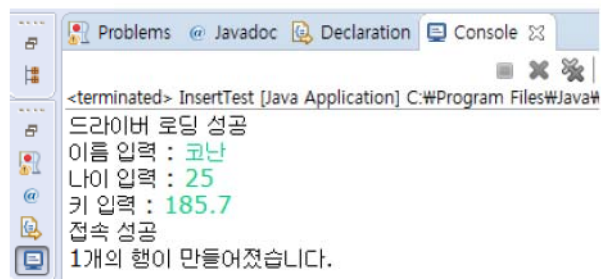
```
select * from dbtest;
```

스크립트 출력 x | 질의 결과 x

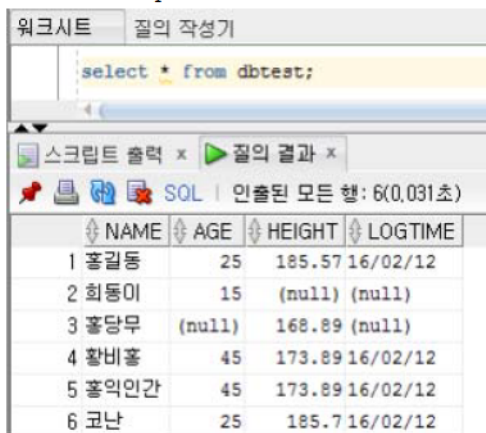
SQL | 인출된 모든 행: 5(0.109초)

	NAME	AGE	HEIGHT	LOGTIME
1	홍길동	25	185.57	16/02/12
2	최동미	15	(null)	(null)
3	홍당무	(null)	168.89	(null)
4	황비홍	45	173.89	16/02/12
5	홍익인간	45	173.89	16/02/12

- 2. eclipse



- 3. developer



워크시트 | 질의 작성기

```
select * from dbtest;
```

스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 6(0.031초)

	NAME	AGE	HEIGHT	LOGTIME
1	홍길동	25	185.57	16/02/12
2	최동미	15	(null)	(null)
3	홍당무	(null)	168.89	(null)
4	황비홍	45	173.89	16/02/12
5	홍익인간	45	173.89	16/02/12
6	코난	25	185.7	16/02/12

<작업 순서>

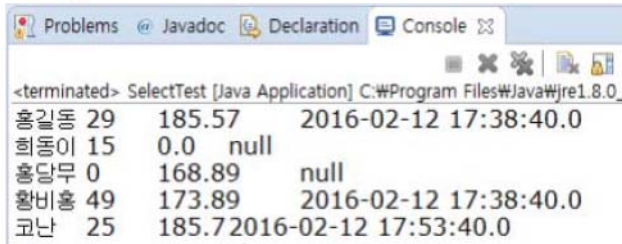
1. Driver Loading
=> 데이터베이스와 연결하는 드라이버 파일을 찾는다.
=> 생성자에서 구현한다.
2. Connection
=> 연결을 관리하는 Connection 객체를 생성한다.
3. SQL 작업처리
=> 작업을 처리할 PreparedStatement 객체를 생성한다.
=> insert into dbtest values('홍길동', 25, 175.7, sysdate);
=> 데이터를 추가한다.
4. 반환되는 결과는 int 변수에 담는다.
5. 접속을 종료한다.

예제3. 레코드 추출 (검색)

Java Project : 4-dbtest
Package : com.dao
Class File : Exam3.java

<실행 결과>

- eclipse



```
<terminated> SelectTest [Java Application] C:\Program Files\Java\jre1.8.0
홍길동 29 185.57 2016-02-12 17:38:40.0
희동이 15 0.0 null
홍달무 0 168.89 null
황비홍 49 173.89 2016-02-12 17:38:40.0
코난 25 185.7 2016-02-12 17:53:40.0
```

<작업 순서>

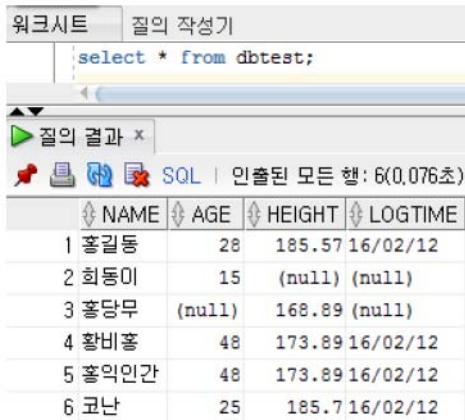
1. Driver Loading
 - => 데이터베이스와 연결하는 드라이버 파일을 찾는다.
 - => 생성자에서 구현한다.
2. Connection
 - => 연결을 관리하는 Connection 객체를 생성한다.
3. SQL 작업처리
 - => 작업을 처리할 PreparedStatement 객체를 생성한다.
 - => select * from 테이블명;
 - => 테이블에 저장된 모든 데이터를 검색한다.
4. 반환되는 결과는 ResultSet 객체에 담는다.
5. 접속을 종료한다.

예제4. 레코드 수정

Java Project : 4-dbtest
Package : com.dao
Class File : Exam4.java

<실행 결과>

- developer



워크시트 | 질의 작성기

```
select * from dbtest;
```

질의 결과 x

SQL | 인출된 모든 행: 6(0.076초)

	NAME	AGE	HEIGHT	LOGTIME
1	홍길동	28	185.57	16/02/12
2	최동미	15	(null)	(null)
3	홍당무	(null)	168.89	(null)
4	황비홍	48	173.89	16/02/12
5	홍익인간	48	173.89	16/02/12
6	코난	25	185.7	16/02/12

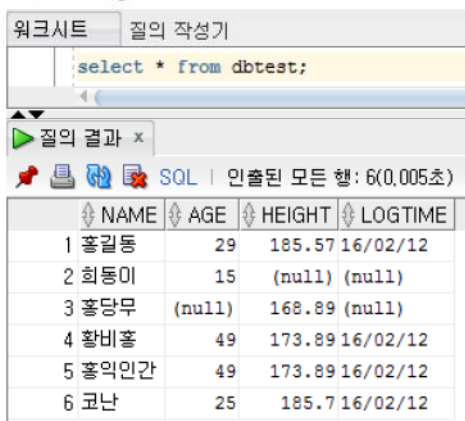
- eclipse



Problems | Javadoc | Declaration | Console

<terminated> UpdateTest [Java Application] C:\Program File:
드라이버 로딩 성공
수정할 이름 입력 : 홍
접속 성공
4개의 행이 수정되었습니다.

- developer



워크시트 | 질의 작성기

```
select * from dbtest;
```

질의 결과 x

SQL | 인출된 모든 행: 6(0.005초)

	NAME	AGE	HEIGHT	LOGTIME
1	홍길동	29	185.57	16/02/12
2	최동미	15	(null)	(null)
3	홍당무	(null)	168.89	(null)
4	황비홍	49	173.89	16/02/12
5	홍익인간	49	173.89	16/02/12
6	코난	25	185.7	16/02/12

<작업 순서>

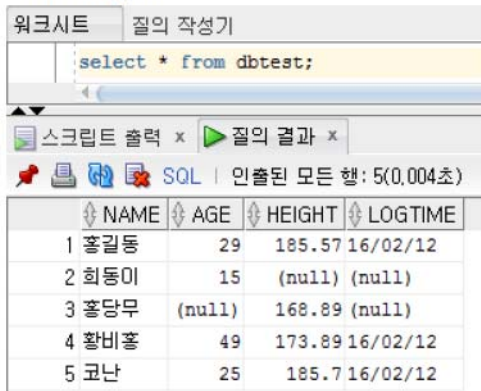
1. Driver Loading
=> 데이터베이스와 연결하는 드라이버 파일을 찾는다.
=> 생성자에서 구현한다.
2. Connection
=> 연결을 관리하는 Connection 객체를 생성한다.
3. SQL 작업처리
=> 작업을 처리할 PreparedStatement 객체를 생성한다.
=> update dbtest set age=age+1 where name like '%홍%';
=> updateArticle() - 수정할 이름 입력 : 홍
=> 홍이 들어가는 모든 레코드의 나이를 1살 증가
4. 반환되는 결과는 int 변수에 담는다.
5. 접속을 종료한다.

예제5. 레코드 삭제

Java Project : 4-dbtest
Package : com.dao
Class File : Exam5.java

<실행 결과>

- developer



워크시트 | 질의 작성기

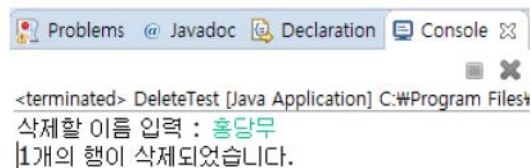
```
select * from dbtest;
```

스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 5(0.004초)

	NAME	AGE	HEIGHT	LOGTIME
1	홍길동	29	185.57	16/02/12
2	최동미	15	(null)	(null)
3	홍당무	(null)	168.89	(null)
4	황비홍	49	173.89	16/02/12
5	코난	25	185.7	16/02/12

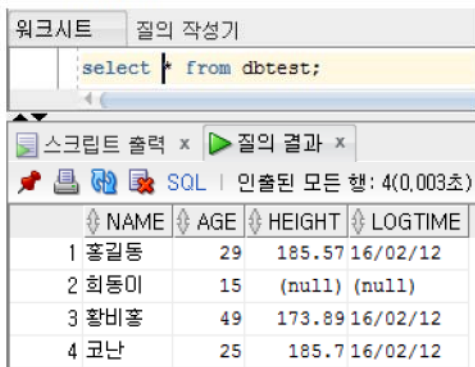
- eclipse



Problems @ Javadoc Declaration Console

<terminated> DeleteTest [Java Application] C:#Program Files
삭제할 이름 입력 : 홍당무
1개의 행이 삭제되었습니다.

- developer



워크시트 | 질의 작성기

```
select * from dbtest;
```

스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 4(0.003초)

	NAME	AGE	HEIGHT	LOGTIME
1	홍길동	29	185.57	16/02/12
2	최동미	15	(null)	(null)
3	황비홍	49	173.89	16/02/12
4	코난	25	185.7	16/02/12

<작업 순서>

1. Driver Loading
=> 데이터베이스와 연결하는 드라이버 파일을 찾는다.
=> 생성자에서 구현한다.
2. Connection
=> 연결을 관리하는 Connection 객체를 생성한다.
3. SQL 작업처리
-> 작업을 처리할 PreparedStatement 객체를 생성한다.
=> delete dbtest where name='홍당무';
=> 이름이 '홍당무'인 데이터를 삭제한다.
4. 반환되는 결과는 int 변수에 담는다.
5. 접속을 종료한다.

문제1. 간단 학교 인적 관리 DB

Java Project : 4-school

Package : com.dao

Class File	: com.dao/Practice1.java	=> main
	com.dao/SchoolDTO.java	=> Java Beans
	com.dao/SchoolDAO.java	=> JDBC 처리
	com.dao/SchoolController.java	=> 메뉴 기능 처리

(1) 테이블 작성

```
create table school (  
    name varchar2(15) not null,  
    value varchar2(15),  
    code number -- 1이면 학생, 2이면 교수, 3이면 관리자  
);
```

(2) menu

관리

1. 입력
2. 검색
3. 삭제
4. 종료

번호선택 :

(3) 1번인 경우

1. 학생
2. 교수
3. 관리자
4. 이전메뉴

번호선택 :

1번 학생인 경우

이름입력 :

학번입력 :

2번 교수인 경우

이름입력 :

과목입력 :

3번 관리자인 경우

이름입력 :

부서입력 :

(4) 2번인 경우

1. 이름 검색 (1개 글자가 포함된 이름은 모두 검색)
2. 전체 검색
3. 이전메뉴

번호선택 : 1

1번 경우

검색할 이름 입력 : 교

이름=홍길교 학번=2015

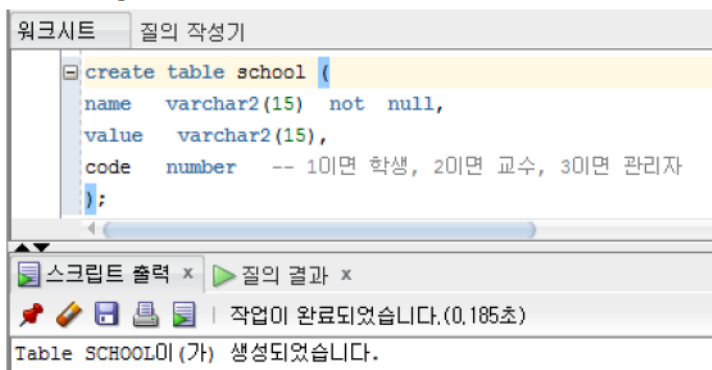
이름=이교수 과목=JAVA

(5) 3번인 경우

삭제를 원하는 이름 입력 : (정확하게 3개의 글자가 모두 입력된 상태)

<실행 결과>

- developer



- eclipse

관리

1. 입력
2. 검색
3. 삭제
4. 종료

번호 선택 : 1

1. 학생
2. 교수
3. 관리자
4. 이전메뉴

번호 선택 : 1

이름 입력 : 홍길동

학번 입력 : 2016001

1개의 행이 만들어졌습니다.

1. 학생
2. 교수
3. 관리자
4. 이전메뉴

번호 선택 : 2

이름 입력 : 고길동

과목 입력 : 컴퓨터

1개의 행이 만들어졌습니다.

1. 학생
2. 교수
3. 관리자
4. 이전메뉴

번호 선택 : 3
이름 입력 : 김길동
부서 입력 : 경리부
1개의 행이 만들어졌습니다.

1. 학생
2. 교수
3. 관리자
4. 이전메뉴

번호 선택 : 4

관리

1. 입력
2. 검색
3. 삭제
4. 종료

번호 선택 : 2

1. 이름 검색
2. 전체 검색
3. 이전메뉴

번호 선택 : 1
이름 입력 : 홍길동

이름=홍길동 학번=2016001

|

1. 이름 검색
2. 전체 검색
3. 이전메뉴

번호 선택 : 2

이름=홍길동 학번=2016001
이름=고길동 과목=컴퓨터
이름=김길동 부서=경리부

1. 이름 검색
2. 전체 검색
3. 이전메뉴

번호 선택 : 3

관리

1. 입력
2. 검색
3. 삭제

4. 종료

 번호 선택 : 3
 삭제를 원하는 이름 입력 : 홍길동
 삭제되었습니다.

 관리

 1. 입력
 2. 검색
 3. 삭제
 4. 종료

 번호 선택 : 2

 1. 이름 검색
 2. 전체 검색
 3. 이전메뉴

 번호 선택 : 2

이름=고길동 과목=컴퓨터
 이름=김길동 부서=경리부

 1. 이름 검색
 2. 전체 검색
 3. 이전메뉴

 번호 선택 : 3

 관리

 1. 입력
 2. 검색
 3. 삭제
 4. 종료

 번호 선택 : 4
 프로그램을 종료합니다.

<작업 순서>

1. Driver Loading
 => 데이터베이스와 연결하는 드라이버 파일을 찾는다.
 => 생성자에서 구현한다.
2. Connection
 => 연결을 관리하는 Connection 객체를 생성한다.
3. SQL 작업처리
 => 작업을 처리할 PreparedStatement 객체를 생성한다.
4. 반환되는 결과는 int나 ResultSet 객체에 담는다.
5. 접속을 종료한다.