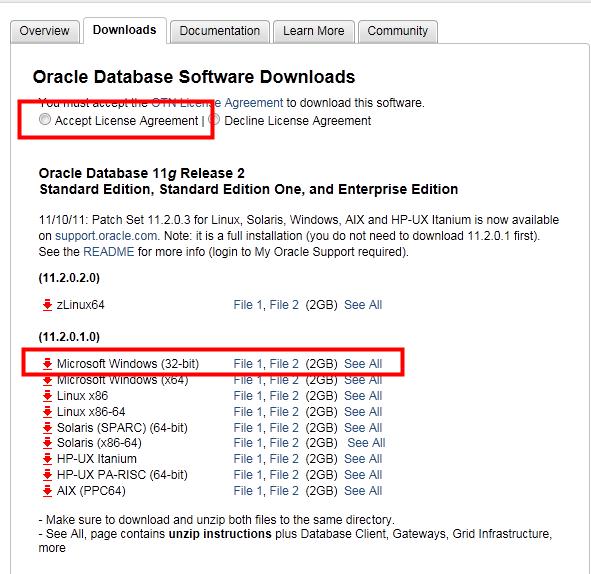
1. **오라클 설치**

사이트 주소는 : <http://www.oracle.com/kr/index.html>

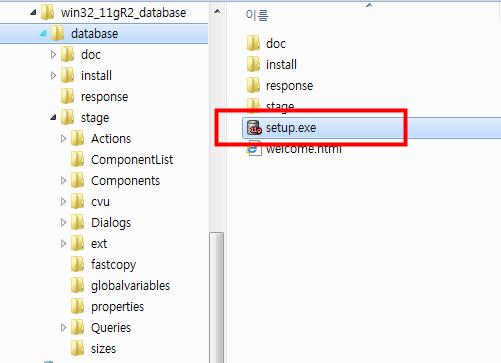
사이트에서 다운로드 메뉴에서 database 11g 를 클릭합니다.



우선 라이센스에 동의한후 자신에 맞는 OS 를 선택하여 다운로드 합니다.

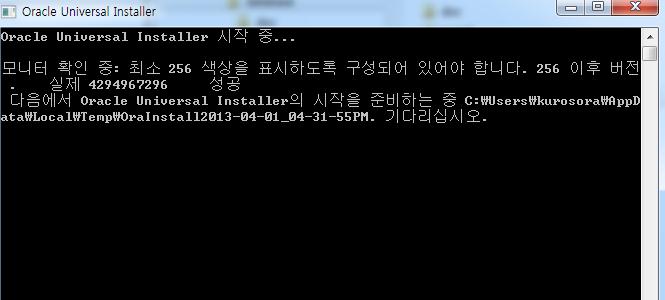
전 윈도우 32bit 이니까.. 위의 빨간 박스 파일을 다운로드 받습니다.

**참! 다운로드를 받기 위해서는 오라클 사이트에 우선 가입해야 합니다.**

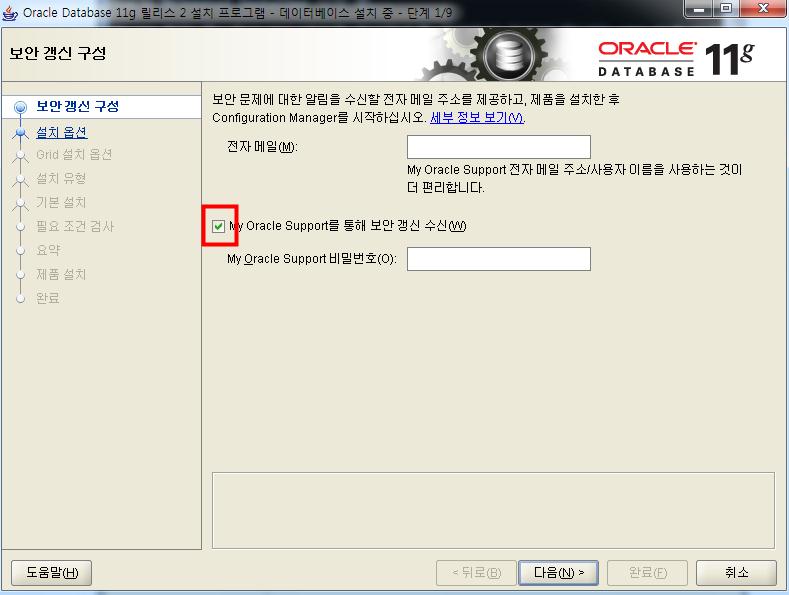


다운로드는 2개의 압축파일을 다운받는데.. 압축을 풀고 합치면 됩니다.

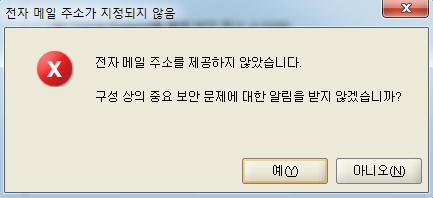
여기서 database 밑의 setup.exe 파일을 실행합니다.



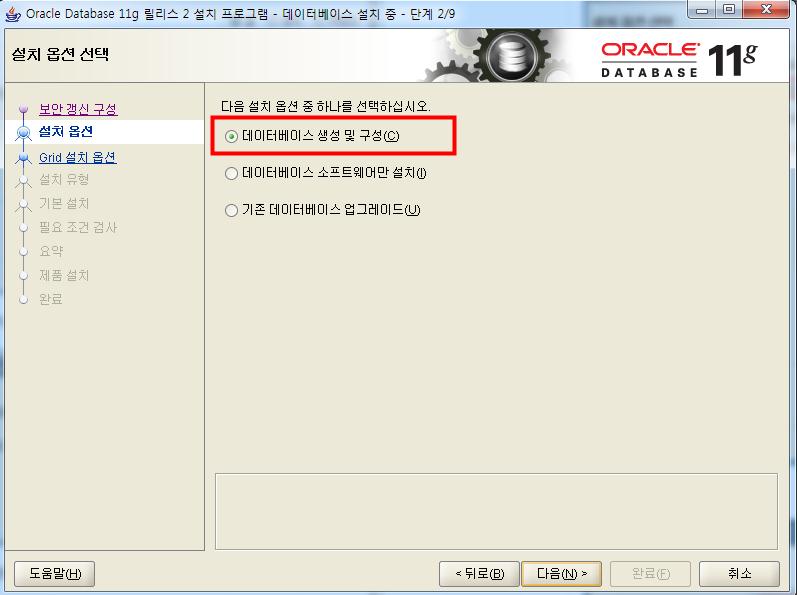
실행하면 잠깐 위의 화면이 나오고



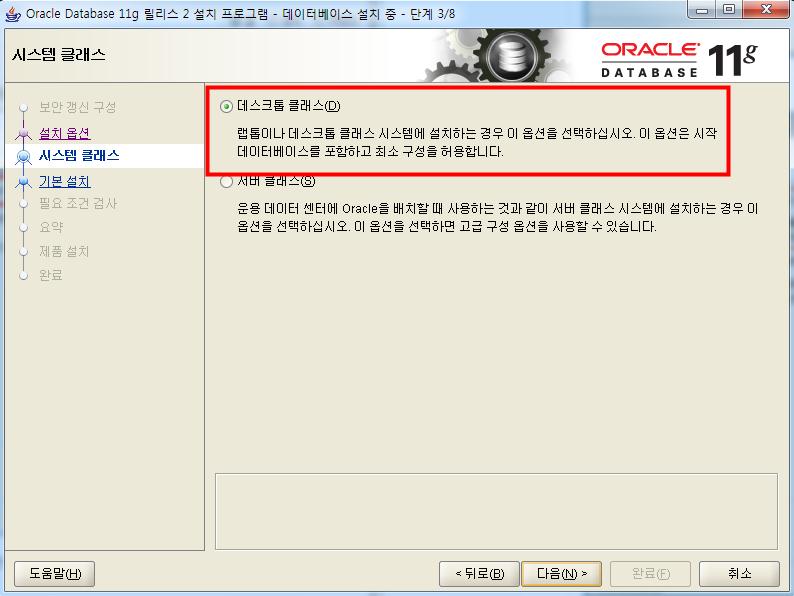
보안정보를 메일로 받으려면 메일주소를 입력하고 정보가 필요없다면 체크를 해제후 다음버튼을 클릭합니다.



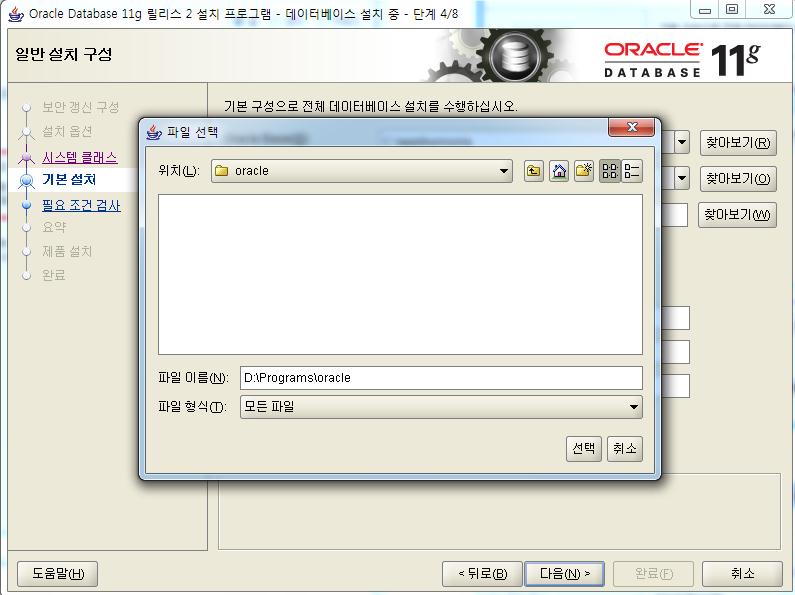
체크를 안할 경우 확인 메시지가 출력됩니다. 정말 정보를 받기 싫으시면 "예"를 클릭하세요.



다음 옵션중 하나를 선택합니다. 전 박스 표시한 것을 선택했습니다.

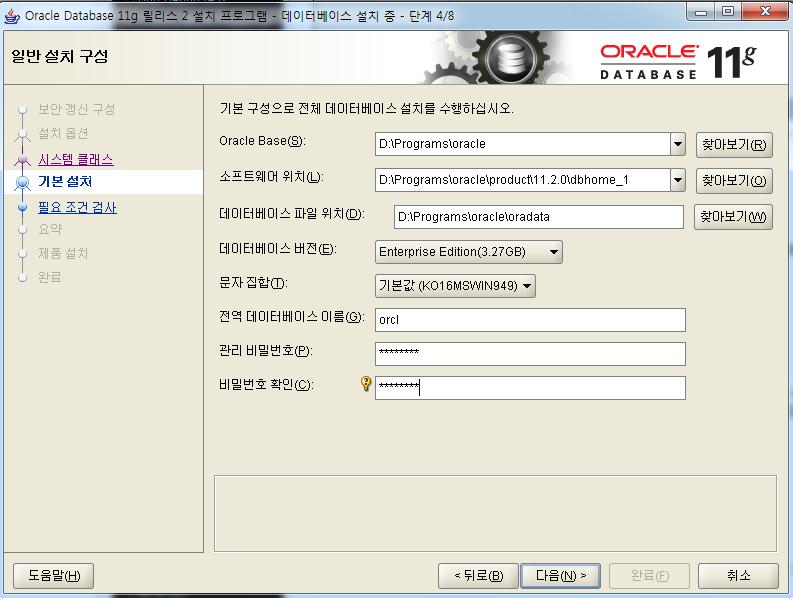


데스크톱 클래스를 선택하고 다음을 클릭합니다.



 오라클 베이직 경로를 지정합니다.

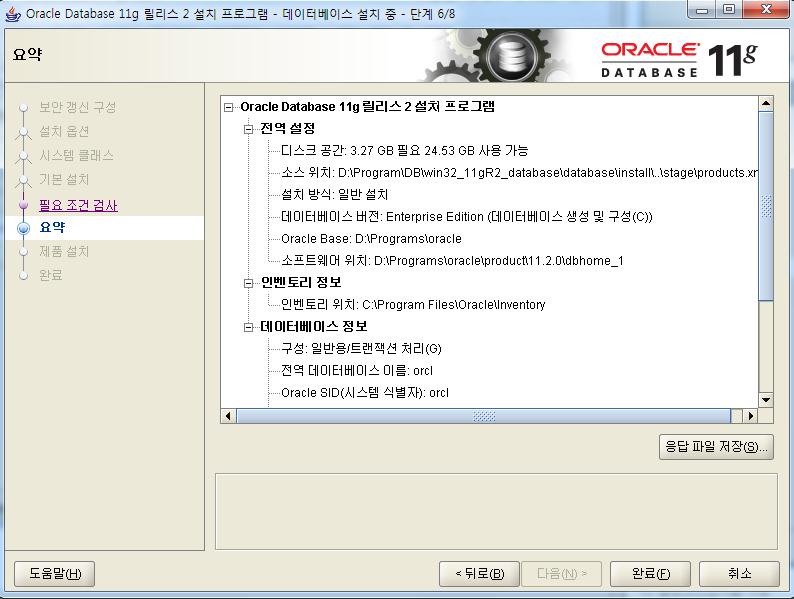
전 D:\Programs\oracle 로 선택하였습니다.



 관리 비밀번호(비밀번호 확인 포함)을 입력 합니다.

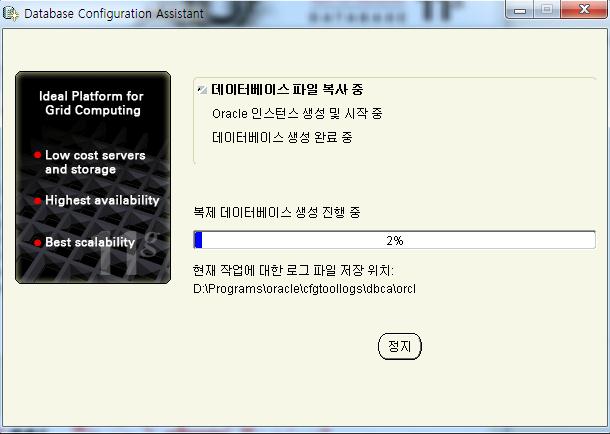
**\* 비밀번호는 대문자, 소문자, 숫자가 모두 포함 되어야 함 (9자 이상 ~ 128자 이하)**



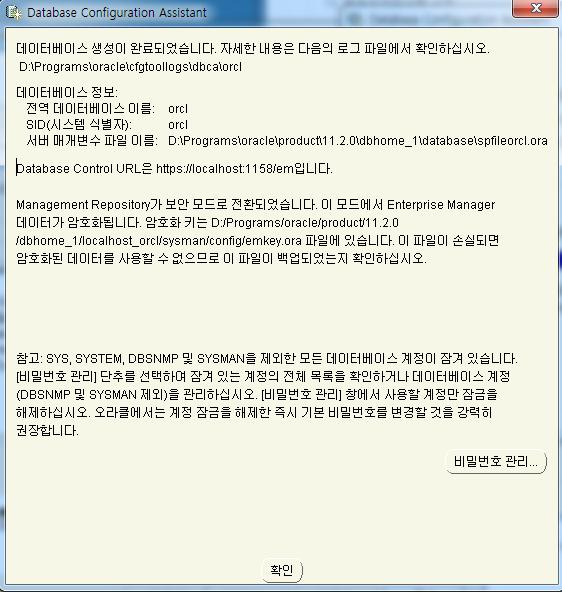


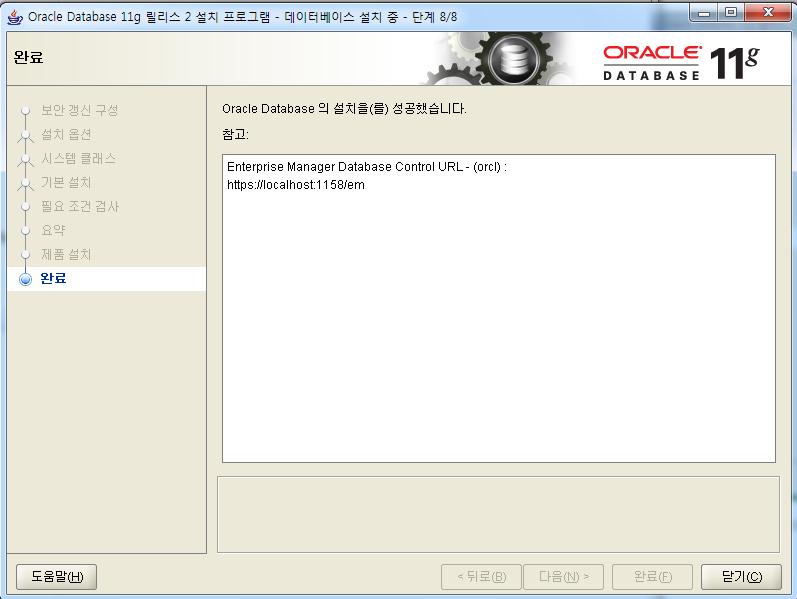
 필요조건검사를 수행합니다. 완료를 클릭하면





오라클 설치를 진행합니다.





이상이 없다면 제품설치과정을 거쳐 "Oraccle Database 의 설치를 성공했습니다." 라는 문구를 확인 할 수 있습니다

1. **오라클 설치시 비밀번호 권장 표준(오라클)**

 비밀번호는 30자를 초과할 수 없습니다.

 비밀번호는 비워 둘 수 없습니다.

 사용자 이름과 비밀번호는 달라야 합니다.

 SYS 계정 비밀번호는 change\_on\_install일 수 없습니다(대소문자를 구분하지 않음).

권장 사항은 아래와 같습니다.

 **소문자를 하나 이상 포함합니다.**

 **대문자를 하나 이상 포함합니다.**

 **숫자를 하나 이상 포함합니다.**

 **8자 이상으로 구성합니다.**

 **밑줄(\_), 달러($) 및 파운드 기호(#)를 포함할 수 있는 데이터베이스 문자 집합을 사용합니다.**

 **비밀번호가 숫자 또는 기호로 시작하는 경우를 비롯하여 특수 문자를 포함하는 경우 비밀번호를 큰 따옴표로 묶습니다.**

 **사전상의 단어를 사용하지 않아야 합니다.**

1. **오라클 삭제 방법**

1. 관리도구 -> 서비스 에서 **Oracle** 로 시작하는 모든 서비스를 먼저 중지시킵니다.

2. 콘솔창은 반드시 **관리자 권한으로 실행**시킵니다.

3. 자신이 설치한 폴더 밑의 **product\11.2.0\dbhome\_1\deinstall\deinstaller.bat  파일 실행**시킴

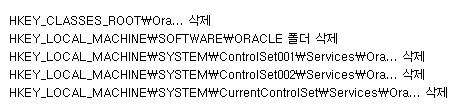
    문구가 여러개 뜨면서 삭제과정이 진행됩니다.

    중간중간 입력을 요구하지만 그냥 엔터치고 가도 됩니다.

    ORCL  입력할 시점에는 조금 시간이 걸리기도 하니.. 참을성이 있어야 할 것 같네요.

4. 모든 과정이 끝나면 재부팅하고 오라클 설치된 폴더 및 파일을 전부 삭제합니다.

5. 깔끔하게 지워지길 원하신다면 레지스트리도 찾아서 지워주세요.



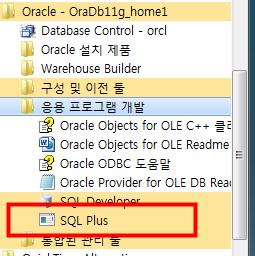
그럼 한번 지워보시길...

저도 지우고 다시 설치하는 중입니다.

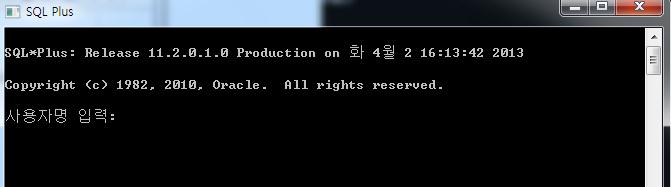
1. **오라클 잠긴 계정 푸는 방법( 계정 : scott / 비번 : tiger )**

오라클 11g 를 설치하다 보면 깜빡 잊고 테스트할 사용자 계정을 풀지 않고 설치하는 경우가 있습니다.

1. 먼저 sql plus 실행 또는 cmd 창을 열어서 sql에 접근해도 됩니다.



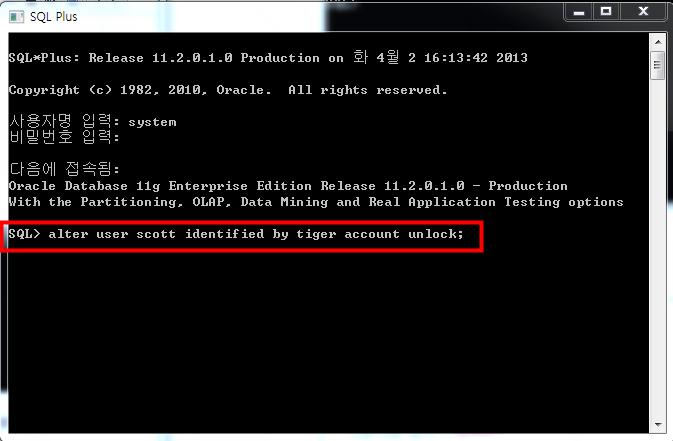
2. sql plus 창에서 사용자 입력 / 비밀번호를 입력합니다.



여기서 사용자명 입력은 : 관리 계정인 **system**, 비밀번호 입력 : **설치시 입력한 비밀번호**를 넣어주세요.

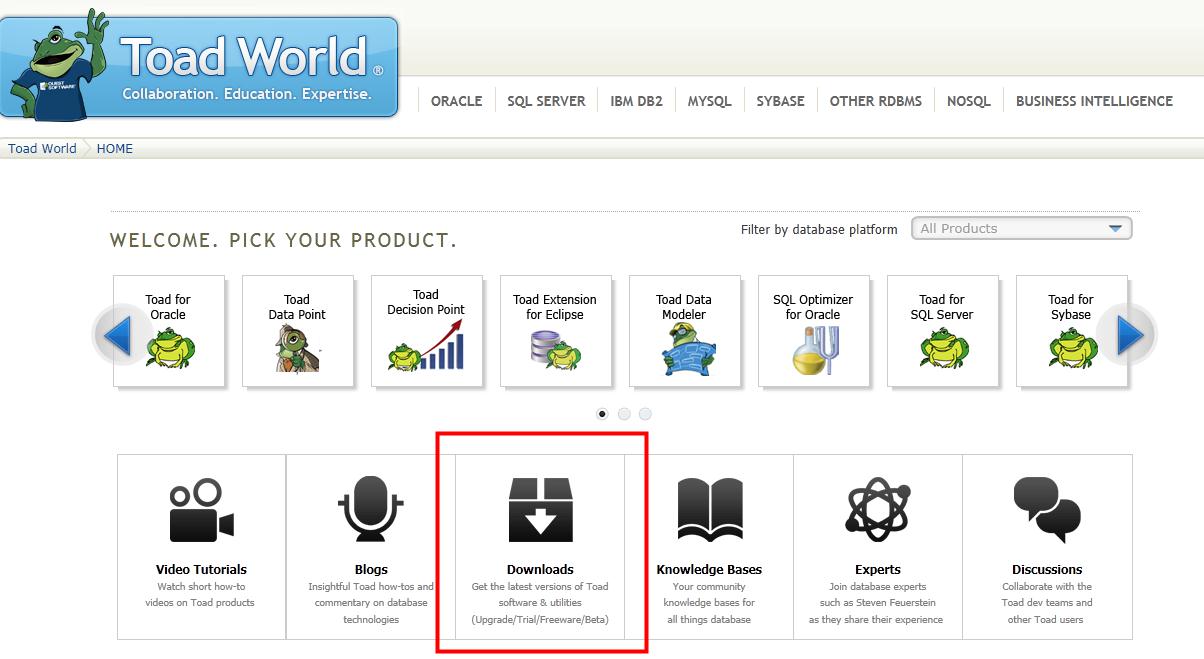
**관리자계정접속 sqlplus / as sysdba**

3. 실습용 scott 계정을 풀어 보겠습니다. (alter user scott identified by tiger account unlock; )



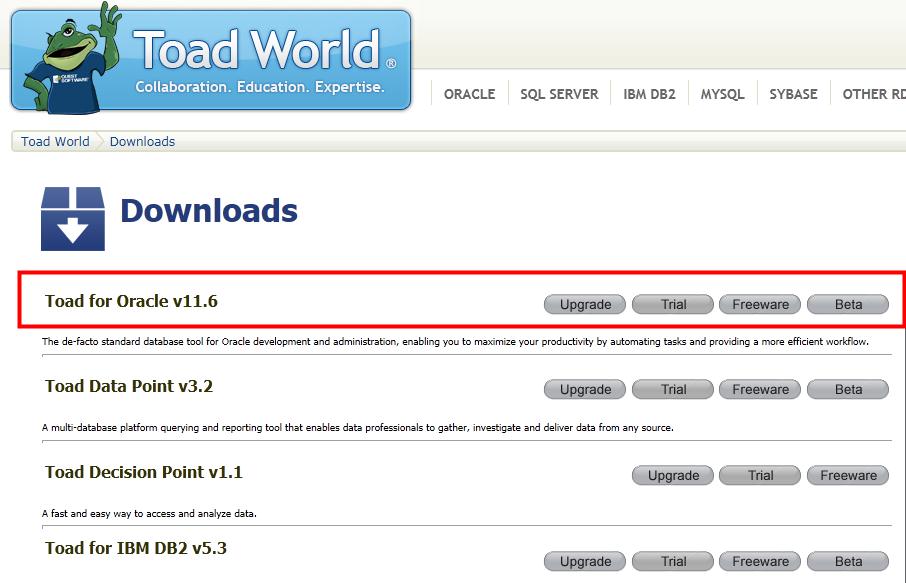
1. **오라클 Toad 다운**

1. 먼저 토드 사이트에 접속해 봅니다. <http://toadworld.com/Default.aspx>



2. 상기 빨간 박스의 다운로드를 클릭합니다.

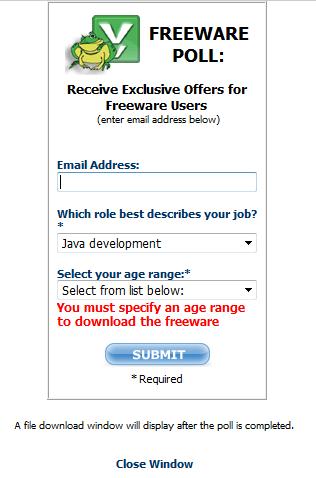
3. 무료 버전이니 freeware 를 선택합니다.



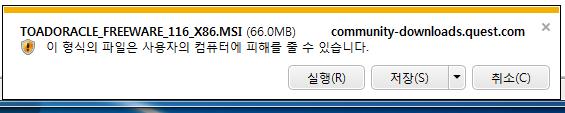
3. 자신의 OS 에 맞게 다운로드 받으세요. 전 32bit 입니다.



 4. 메일 정보와 기타 정보를 입력해야 되는군요.

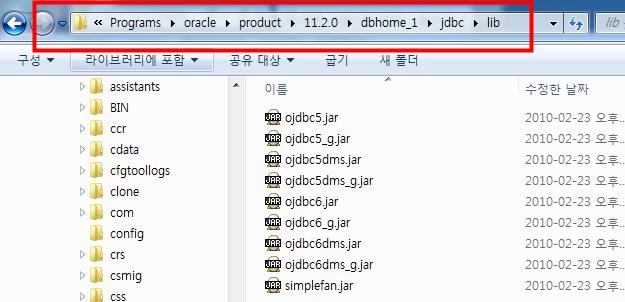


 5. 해당 정보를 입력하고 submit 버튼을 클릭하면 파일을 다운받을 수 있습니다.



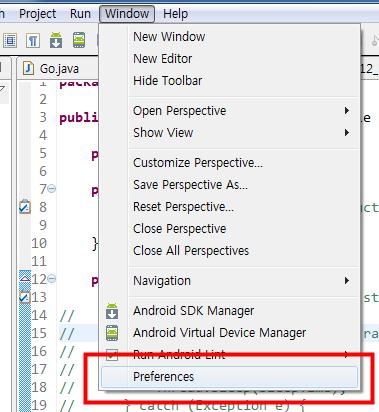
1. **오라클 드라이버를 이클립스에서 클래스패스에 설정하기**

1. 오라클이 설치되어 있다면 오라클 드라이버는 **설치위치\product\11.2.0\dbhome\_1\jdbc\lib 밑에** 있습니다.

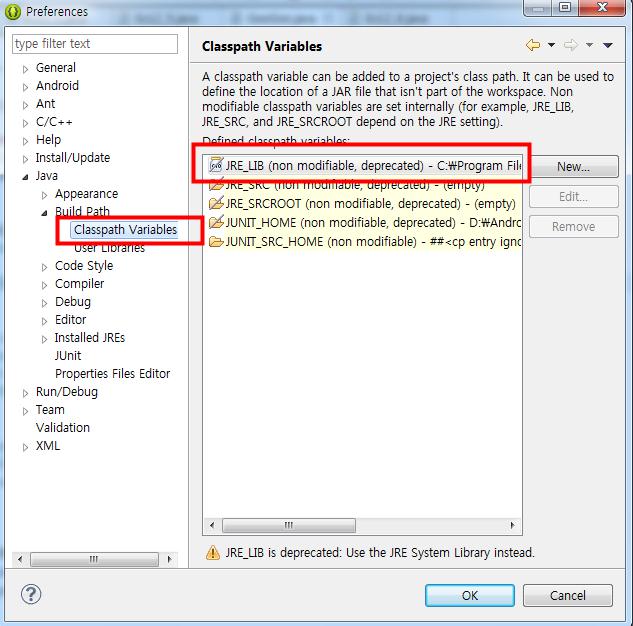


2. 이제 이클립스에 클래스패스를 설정해 볼까요

   먼저 windows -> perfernces 를 선택합니다.



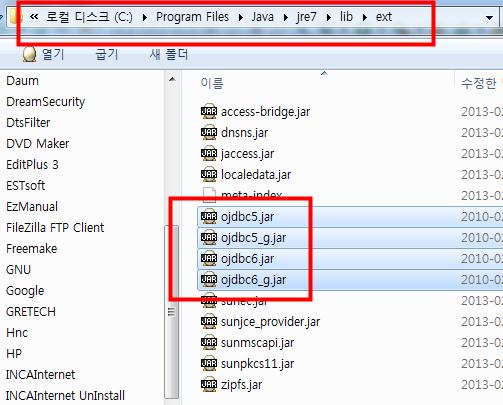
3. java -> build path -> classpath variables 에서 JRE\_LIB 경로를 파악해 봅니다.



4. 저의 경우는 C:\Program Files\Java\jre7\lib 경로가 이클립스 툴에서 인식하고 있는 클래스 패스 경로군요.

 5. lib 폴더 밑의 ext 폴더에 ojdbc.jar 관련 파일을 복사하여 넣어줍니다.

그럼 자동으로 압축파일이 클래스패스에 추가됩니다.



JDBC 연결을 위한 환경설정이 모두 끝났으니 이제 자바 코드로 데이터베이스 연동할 수 있습니다.

1. **Oracle system 또는 scott 계정 비밀번호 변경 방법**

SYS, SYSTEM 비번을 잃어버렸을때.. 다음과 같이 비번을 변경 할 수 있습니다.

1. 오라클 설치 폴더의 bin 폴더 아래로 이동한 후 다음과 같이 입력하세요.

C:\> sqlplus "/as sysdba"

2. 비번 변경을 위해 다음과 같이 입력하세요.

SQL> alter user system identified by yourPassword;

SQL> alter user sys identified by yourPassword;

**scott 비번 분실시**

admin 계정으로 로그인하셔서 (sysdba 또는 system)

SQL> alter user scott identified by tigertiger;

로 다시 변경하셔서 접속해보세요

1. **Oracle SQL 쿼리문 정리**

**■ 오라클 접속**

**1) 관리자로 접근하는 방법**

sqlplus sys(system)/[비밀번호] as sysdba

sqlplus / as sysdba (계정명과 비번 생략할 수 있음) => DBA로 접속

**2) 일반 계정으로 접근하는 방법**

sqlplus 계정명/계정비밀번호

**■ 락(lock) 계정을 풀려면 sys 계정으로 접속하여**

alter user scott identified tiger account unlock; 실행함

alter user [추가할 계정] identified [비밀번호] account unlock;

**■ 데이타베이스 사용자 변경**

conn 사용자명/사용자비밀번호

**■ 데이타베이스 정보 확인**

show parameter db\_name;

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

**■ 테이블 구조 확인 방법**

desc 테이블명;

**■ 테이블 조회**

select \* from tab;

문자데이터와 날짜데이터는 반드시 ''(단일 따옴표)를 사용하여 표현함

리터널은 대소문자를 구분함

오라클에서 NULL값

1) 데이타 검색시 컬럼의 값이 비어있는 형태임

2) 기본적으로 NULL값을 허용하며 제약조건에서 NULL값을 허용하지 않을 수 있음

3) NULL값의 연산 결과는 NULL값으로 나옴

=> NULL값 가진 컬럼 연산시 NVL 함수를 사용함

NVL(컬럼명, 기본값)

##########################################################################

**SELECT (검색)**

SELECT [DISTINCT] {컬럼명} [as alias]

FROM 테이블명

[WHERE 조건식]

[ORDER BY 컬럼명 ASC|DESC];

=> DISTINCT : 중복되어 출력되는 값을 제거하는 키워드

**=> WHERE 절 조건식**

**[비교연산자]**

= (같다), != (같지않다), > (크다), >= (크거나 같다), < (작다), <= (작거나 같다)

**[논리연산자]**

AND, OR, NOT

**[BETWEEN a AND b]** <=> 반대는 NOT BETWEEN a AND b 임

특정 범위의 값을 검색할 때 사용함

**[IN]** <=> 반대는 NOT IN 임

여러 개의 값을 동시에 비교하여 일치하는 데이터를 얻을 때 사용

**[LIKE]** <=> 반대는 NOT LIKE 임

임의의 문자 또는 문자열을 포함한 데이타를 검색할 때 사용

% (퍼센트) : 없거나 여러개의 문자를 대체함

\_(언더바) : 하나의 문자를 대체함

**[IS NULL]** <=> 반대는 IS NOT NULL 임

컬럼의 데이터 중에서 NULL값을 얻을 때 사용

**=> ORDER BY 절**

특정 컬럼에 대해 정렬해서 보기 원할 때 사용

ASC : 오름차순 (기본값으로 생략가능함)

DESC : 내림차순

**<오라클 유용한 함수들>**

테이블명에 dual 을 넣으면 임시적으로 사용되는 더미(dummy) 테이블임

SYSDATE : 시스템의 현재 날짜를 구하는 함수

MAX(컬럼명) : 지정한 컬럼의 값중에서 최대값을 구하는 함수

MIN(컬럼명) : 지정한 컬럼의 값중에서 최소값을 구하는 함수

SUM(컬럼명) : 컬럼들의 총합을 구하는 함수

AVG(컬럼명) : 평균을 구하는 함수

COUNT(컬럼명) : 컬럼의 행의 개수를 구하는 함수

COUNT(\*) : 전체 레코드의 개수를 구함

TO\_CHAR(컬럼명, [포멧]) : 날짜 및 수치 데이터를 문자로 변환하기 위한 함수, 특정 포멧 지정할 수 있음

**[날짜 출력 포멧 형식]**

YYYY : 년도 4자리

YY : 년도 2자리

MM : 월을 숫자로 표현

MON : 월을 알파벳으로 표현

DAY : 요일 표현

DY : 요일을 약어로 표현

**[시간 출력 포멧 형식]**

AM, PM : 오전, 오후 시각표시

A.M, P.M : 오전, 오후 시각표시

HH, HH12 : 시간 (1~12)

HH24 : 24시간으로 표현 (0~24)

MI : 분 표현

SS : 초 표현

**[숫자 출력 포멧 형식]**

9 : 한자리의 숫자 표현

0 : 앞부분을 0으로 표현

$ : 달러 기호를 앞에 표현

. : 소수점을 표시

, : 특정 위치에 , 표시

B : 공백을 0으로 표현

L : 지역 통화

##########################################################################

**INSERT (입력)**

INSERT INTO 테이블명 [ (컬럼명, 컬럼명)]

VALUES (값, 값);

**UPDATE (수정)**

UPDATE 테이블명

SET 컬럼명=새로운값 [, 컬럼명=새로운값]

[WHERE 조건식];

**DELETE (삭제)**

DELETE FROM 테이블명

[WHERE 조건식];

1) ROLLBACK 으로 트랜잭션 취소가 가능함

2) 테이블의 데이타만 삭제됨

3) ROLLBACK 구현을 위해서는 UNDO 기능이 추가됨

##########################################################################

**CREATE (생성)**

<컬럼 레벨 지정방식>

CREATE TABLE 테이블명 (

컬럼명 데이타형 DEFAULT 값,

컬럼명 데이타형[CONSTRAINT 제약조건이름] 제약조건);

<테이블 레벨 지정방식>

CREATE TABLE 테이블명 (

컬럼명 데이타형 DEFAULT 값,

컬럼명 데이타형,

[CONSTRAINT 제약조건이름] 제약조건);

=> DEFAULT 값

1) DEFAULT 에서 지정한 값으로 저장되고 지정하지 않은 곳은 NULL값으로 저장됨

2) 현재 날짜를 입력하는 경우이거나 항상 동일한 값만 가지는 컬럼에 사용됨

=> 제약조건(Constraints)

1) 컬럼에 데이타를 저장할 때 유효한 데이타 저장을 위한 제약조건 사용할 수 있음

2) 제약조건이름 지정 방식 : 테이블명\_컬럼명\_제약조건약어 로 표현

**[제약조건]**

**PRIMARY KEY** : 레코드의 중복을 방지할 목적으로 사용, 유일한 값을 가지고 NULL값 허용안됨 (약어 PK)

**FOREIGN KEY** : 여러 테이블간의 관계를 맺기 위해 사용, 다른 테이블 PK 컬럼값 또는 UNIQUE 컬럼값 및 NULL값만을 가져야 함, (약어 FK)

**UNIQUE** : 유일한 값을 가지거나 NULL값을 가질 수 있음 (약어 UK)

**NOT NULL** : 반드시 값을 가져야 함 (약어 NN), 컬럼 레벨 방식으로만 지정 가능함

**CHECK** : 지정한 조건에 일치하는 데이타 값만 저장 가능함 (약어 CK)

**=> 테이블 복사해서 만들기**

CREATE TABLE 새테이블명 (컬럼명, 컬럼명)

AS

SELECT 컬럼명, 컬럼명 FROM 기존테이블명

[WHERE 만족하지 않는 조건식];

1) CTAS(씨탁스)라 부르며 제약조건은 복사안됨

2) 데이타까지 복사됨

3) WHERE절에 만족하지 않는 조건식을 넣으면 데이타는 복사가 안됨

##########################################################################

**DROP (삭제)**

1) 테이블을 삭제하면 저장된 데이타도 삭제됨

2) FK 제약조건이 있는 경우 테이블 삭제가 불가능함

3) 참조하는 테이블부터 먼저 삭제하거나 제약조건 비활성 시킨후 삭제요망

4) 10g 부터 휴지통같은 recyclebin 을 이용하여 테이블 관리함

5) recyclebin 저정된 테이블은 복구도 가능하고 완전 삭제도 가능함

6) 테이블을 삭제하면 BIN 으로 시작되는 테이블 생성됨 (recyclebin 저장된 이름)

DROP TABLE 테이블명 [purge];

=> recyclebin 테이블 확인 방법

SHOW recyclebin;

=> recyclebin 테이블 복구 방법

FLASHBACK TABLE 삭제된테이블명 TO BEFORE DROP;

=> recyclebin 테이블 휴지통 지우는 방법

PURGE recyclebin;

=> recyclebin 넣지 않고 완전 삭제하는 방법

PURGE 키워드를 사용하면 FLASHBACK 으로 복구 불가능함

##########################################################################

**TRUNCATE (저장공간 삭제)**

1) 테이블에 저장된 데이타를 삭제하는 구문

2) ROLLBACK 으로 트랜잭션 취소할려고 해도 불가능함

3) 테이블의 데이타 및 저장구조가 삭제됨 따라서 메모리를 효율적으로 관리 가능함

4) ROLLBACK 이 안되니까 UNDO 기능이 필요없음

TRUNCATE TABLE 테이블명;

##########################################################################

**ALTER (수정)**

1) 기존 테이블의 구조를 변경하는 구문

2) 컬럼 추가, 삭제, 데이타형 변환 및 길이 변경이 가능함

3) 제약조건도 추가/삭제 가능함

=> 컬럼 추가

ALTER TABLE 테이블명

ADD (컬럼명 데이터형, [컬럼명 데이터형]);

=> 제약조건 추가

ALTER TABLE 테이블명

ADD [CONSTRAINT 제약조건명] 제약조건(컬럼명);

=> 컬럼 변경

ALTER TABLE 테이블명

MODIFY (컬럼명 데이터형);

=> 제약조건 변경

ALTER TABLE 테이블명

MODIFY (컬럼명 데이타형 [CONSTRAINT 제약조건명] 제약조건);

=> 컬럼 삭제

ALTER TABLE 테이블명

DROP (컬럼명);

##########################################################################

**시퀸스 (SEQUENCE)**

1) 일정 규칙에 의해 연속적으로 자동 증가, 감소해야 하는 경우 적용

2) 테이블내 정수값을 가지는 PK 에 사용됨

CREATE SEQUENCE 시퀀스명

[ START WITH n ] 시퀀스 번호의 시작값 지정 (기본 1부터 시작)

[ INCREMENT BY n ] 시퀀스의 증가, 감소치 지정 (기본 1씩 증가, 음수도 가능)

[ MAXVALUE n | NOMAXVALUE ] 최대값 지정

[ MINVALUE n | NOMINVALUE ] 최소값 지정

[ CYCLE | NOCYCLE ] 최대값까지 증가가 완료하면 다시 초기값에서 시작하도록 지정

[ CACHE n | NOCACHE ]; 기본 20개의 cache 를 관리

**시퀀스명.NEXTVAL : 시퀀스의 다음값을 얻음**

**시퀀스명.CURRVAL : 시퀀스의 현재값을 얻음**

##########################################################################

RENAME (이름변경)

##########################################################################

**COMMIT (트랜잭션 저장)**

**ROLLBACK (트랜잭션 취소)**

SAVEPOINT (트랜잭션 책갈피 기능)

##########################################################################

**GRANT (권한 부여)**

GRANT 권한 및 롤

TO 사용자계정;

=> 권한 및 롤

1) **CONNECT** : 데이타베이스 접속할 수 있는 롤

2) **RESOURCE** : 테이블 생성, 데이터 조회, 삭제, 수정, 저장 할 수 있는 롤

REVOKE (권한 취소)

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

**<사용자 계정 생성>**

CREATE USER 사용자명

IDENTIFIED BY 비밀번호;

**<사용자 계정 비번 변경>**

ALTER USER 사용자명

IDENTIFIED BY 새비밀번호;

**<사용자 삭제 - DBA만 가능함>**

DROP USER 사용자명 CASCADE;

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

**조인 (JOIN)**

**Equi Join** : 두 테이블의 공통적으로 존재하는 컬럼값이 일치하는 행을 연결하여 조인

SELECT 테이블1.컬럼명, 테이블2.컬럼명

FROM 테이블1, 테이블2

WHERE 테이블1.공통컬럼명 = 테이블2.공통컬럼명

별칭을 사용할 경우

SELECT 별칭1.컬럼명, 별칭2.컬럼명

FROM 테이블1 별칭1, 테이블2 별칭2

WHERE 별칭1.공통컬럼명 = 별칭2.공통컬럼명

**Non Equi Join** : 동등연산자(=) 이외의 연산자를 사용하여 조인하는 방법

**Self Join** : 다른 테이블이 아닌 자기 자신과의 조인을 의미함

**Outer Join** : 조인 조건에 만족하지 않아도 해당 행을 출력하고 싶을 때 사용할 수 있음

조인할 데이터가 없는 쪽에 (+) 연산자를 사용함

1. **오라클 이클립스에서 JDBC 연결 작업 테스트**

이제 콘솔창에서 오라클 SQL 쿼리문을 연습하였다면 이제 이클립스에서 오라클 JDBC 연결 테스트를 해보겠습니다.

|  |
| --- |
| package com\_kurosora;  import java.sql.DriverManager; // 클래스  import java.sql.Connection; // 인터페이스  import java.sql.Statement; // 인터페이스  import java.sql.ResultSet; // 인터페이스  import java.sql.SQLException; // 예외  /\*\*  \* 라인3 JDBC 드라이버를 관리하기 위한 기본적인 서비스  \* 라인4 특정의 DB와의 접속(세션)을 표현, 접속의 문맥 내에서 SQL문이 실행되고 결과가 리턴 됨  \* 라인5 정적 SQL 문을 실행하여 작성된 결과를 돌려주기 위해 사용되는 객체  \* 라인6 DB의 결과 세트를 나타내는 테이터의 테이블로 DB에 조회하는 문장을 실행하는 것에 의해 생성  \* 라인7 DB 액세스 에러 또는 그 외의 에러에 관한 정보를 제공하는 예외  \*/  public class Ex14\_1\_Jdbc\_Select {  /\*\*  \* @param args  \*/  public static void main(String[] args) {  // TODO Auto-generated method stub  /\*\*  \* JDBC 실행 순서  \* 1. 오라클 데이터베이스 연동을 위한 4가지 정보를 문자열에 저장함  \* oracle.jdbc.driver 패키지내에 있는 OracleDriver 클래스를 driver 변수에 저장  \* 오라클 위치(현재 로컬에 설치되어 있으니 localhost), 포트번호(기본 포트번호 : 1521), 데이터베이스명(orcl) 정보를 url 변수에 저장  \* 계정명와 계정비번  \*/  String driver = "oracle.jdbc.driver.OracleDriver";  String jdbcUrl = "jdbc:oracle:thin:@localhost:1521:orcl";  String userid = "scott";  String passwd = "tiger";    Connection conn = null;  Statement stmt = null;  ResultSet rs = null;    try {    // 2. 드라이버 로딩 (OracleDriver 클래스를 메모리에 로딩함)  Class.forName(driver);    // 3. Connection 맺기 (java.sql 패키지의 Connection 사용)  // DriverManager 클래스 이용 Connection 을 구함  // getConnection 메소드 인자로 url, userid, passwd 값을 지정하여 데이타베이스 연결함  conn = DriverManager.getConnection(jdbcUrl, userid, passwd);    // 4. Statement 생성  // 요청할 SQL 문을 전송할 때 사용되는 API  // Connection 의 createStatement() 매소드를 이용 Statement 를 생성함  stmt = conn.createStatement();    // 5. SQL 문 작성 (; 세미콜론은 입력하지 않음)  String querySql = "SELECT deptno, dname, loc FROM dept";    // 6. SQL 문 전송은 Statement 객체를 이용해 데이터베이스에 전송함  // executeQuery() 매소드를 사용함  // executeUpdate() 매소드 (INSERT, UPDATE, DELETE) 예) int n = stmt.executeUpdate(query);  rs = stmt.executeQuery(querySql);    // ResultSet 객체로 포인터를 이용해 원하는 데이터를 얻음  while (rs.next()) {    int deptno = rs.getInt("deptno");  String dname = rs.getString("dname");  String loc = rs.getString("loc");  System.out.println(deptno + " / " + dname + " / " + loc);    }  } catch (Exception e) {  // TODO: handle exception  e.printStackTrace();    } finally {    try {  // 사용한 자원 해제 시킴(사용한 역순으로 해제)  rs.close();  stmt.close();  conn.close();    } catch (SQLException e) {  // TODO: handle exception  e.printStackTrace();    }  }    }  } |

1. **오라클 이클립스에서 JDBC 연결 작업 테스트(PrepareStatement() 이용)**

Statement() 객체를 이용하는 경우

일반적인 SQL 문을 전송할 때 사용하는 API 입니다. 이것의 단점은 새로운 레코드를 여러번 저장하는 경우에 동일한 INSERT 문을 여러 번 작성해야 되며 문자나 날짜의 경우 '' 를 지정해야 하는데.. '' 를 지정하지 않을 경우 예외가 발생되어 불편한 것을..

PrepareStatementt() 객체를 사용하면 SQL 문을 한번만 작성하고 데이터만 나중에 추가로 설정할 수 있습니다.

받는 값은 ? (물음표) 로 표시합니다.

package com\_kurosora;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class Ex14\_5\_PrepareStatement\_Select {

/\*\*

\* PrepareStatement 사용

\* 1. Statement 객체

\* 일반적인 SQL 문을 전송할 때 사용하는 API

\* 새로운 레코드를 여러번 저장하는 경우 동일한 INSERT문을 여러번 작성해야 문제발생

\*

\* 2. PrepareStatement 객체

\* SQL 문을 한번만 작성하고 데이터만 나중에 추가로 설정하면서 작업할 수 있는 API

\*

\* 3. CallableStatement 객체

\* 자바언어와 PL/SQL 과의 연동시 사용되는 API

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

// 오라클 데이터베이스 연동 정보

String driver = "oracle.jdbc.driver.OracleDriver";

String jdbcUrl = "jdbc:oracle:thin:@localhost:1521:orcl";

String userId = "scott";

String passWd = "tiger";

Connection conn = null;

// Statement stmt = null;

// Statement 대신에 PreparedStatement 사용

PreparedStatement pstmt= null;

ResultSet rs = null;

try {

// 드라이버 로딩

Class.forName(driver);

// Connection 맺기

conn = DriverManager.getConnection(jdbcUrl, userId, passWd);

// stmt = conn.createStatement();

// PrepareStatement 사용시 실제 변수값은 ? 를 사용함

String querySql = "SELECT deptno, dname, loc FROM dept WHERE deptno = ?";

// PrepareStatement 생성전에 SQL 문을 먼저 작성해야 함

pstmt = conn.prepareStatement(querySql);

// set 메소드를 사용하며 인텍스를 이용해서 값을 지정함

// 예) setInt(인덱스, 값) setString(인덱스, "값")

pstmt.setInt(1, 40);

// executeUpdate() 매소드 (INSERT, UPDATE, DELETE) 예) int n = pstmt.executeUpdate();

rs = pstmt.executeQuery();

while (rs.next()) {

int deptno = rs.getInt("deptno");

String dname = rs.getString("dname");

String loc = rs.getString("loc");

System.out.println(deptno + " / " + dname + " / " + loc);

}

} catch (Exception e) {

// TODO: handle exception

e.printStackTrace();

} finally {

try {

rs.close();

pstmt.close();

conn.close();

} catch (SQLException e2) {

// TODO: handle exception

e2.printStackTrace();

}

}

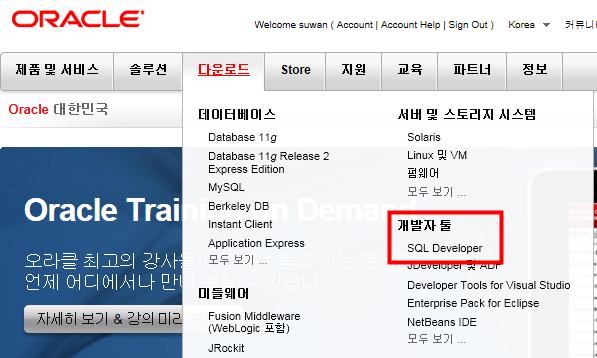
}

}

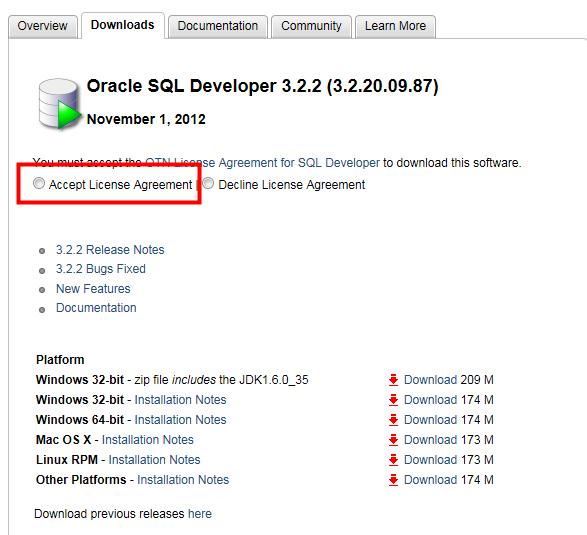
1. **오라클 SQLDeveloper 툴 다운로드 및 사용**

1. 오라클 사이트에 접속합니다. : <http://www.oracle.com/kr/index.html>

2. 아래와 같이 **다운로드 -> 개발자 툴 -> SQL Developer** 를 선택합니다.

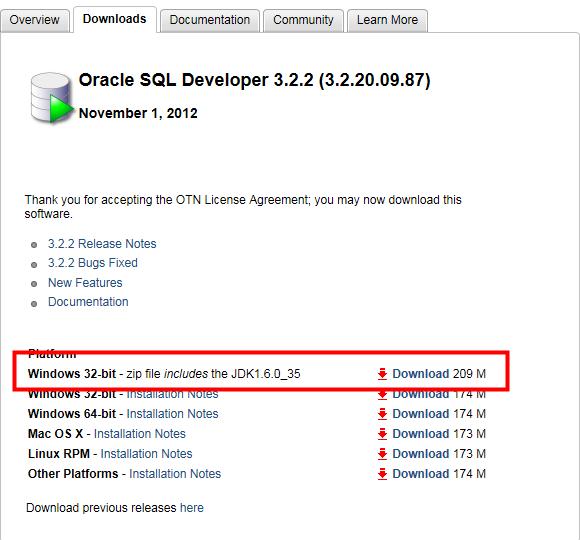


3. 라이센스에 동의를 해야 합니다.

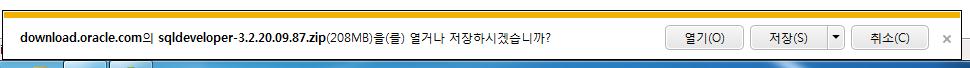


4. 다운로드 받을려면 회원으로 가입해야 합니다. 자신의 OS 에 맞는 버전을 선택하세요.

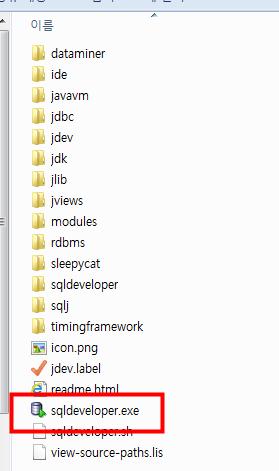
    전 윈도우 32비트를 다운로드 받습니다.



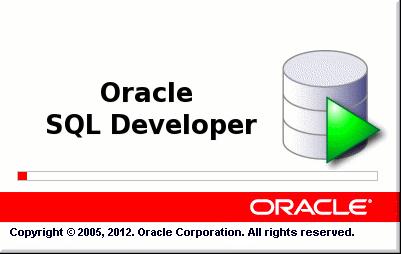
5. 클릭하면 파일을 다운로드 받을 수 있습니다. 200M 정도 되네요!



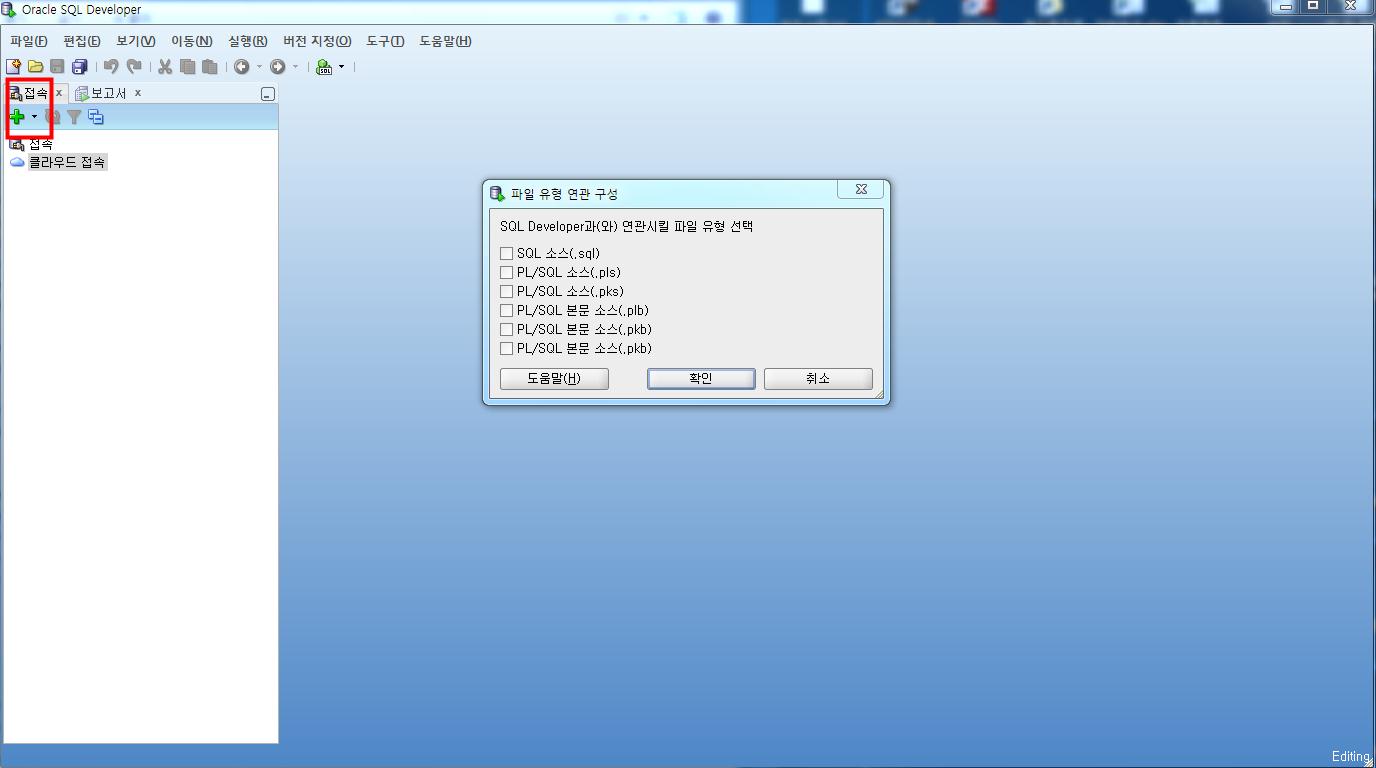
6. 다운로드 받은 파일을 압축 풀면 **sqldeveloper.exe 실행 파일**이 있습니다. 클릭하세요.



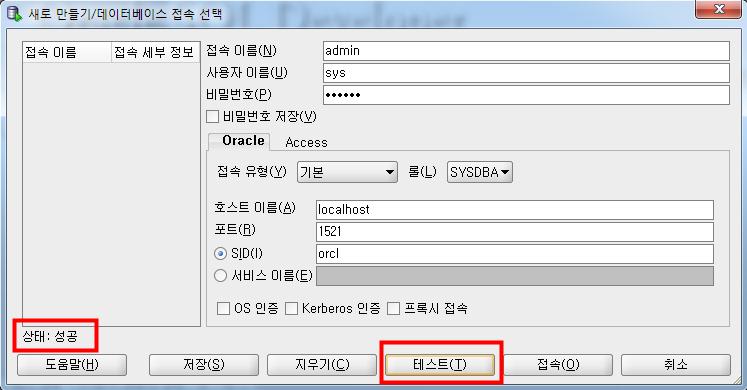
7. 실행시키면 화면이 나오고



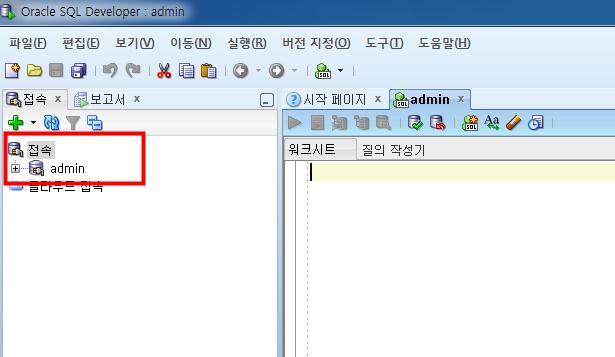
8. **왼쪽 상단의 + 버튼을 클릭**하면 새 접속을 생성하는 창이 나옵니다.



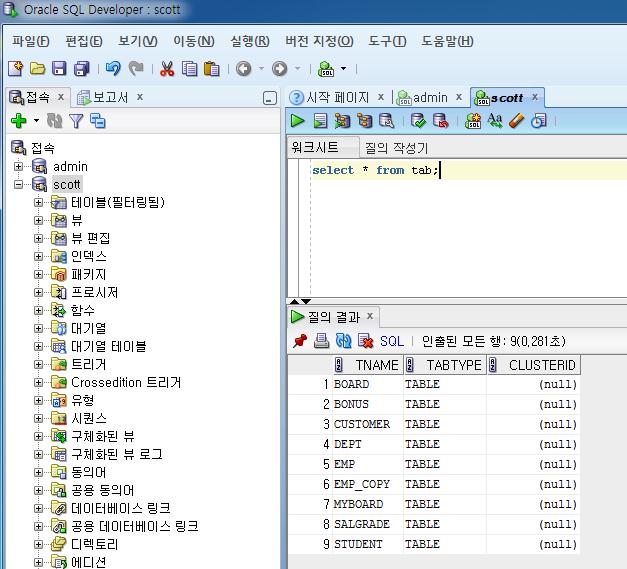
9. 데이터베이스 접속 정보를 넣어주세요. 테스트시 성공하면 아래와 같이 성공 메시지가 출력됩니다.



10. 접속을 하게 되면 아래와 같이 나옵니다.



11. 접속에 성공하면 쿼리등 이제 작업을 수행할 수 있습니다.



http://diyall.tistory.com/entry/오라클-Toad-for-Oracle-무료버전Freeware-다운로드-받기