

데이터베이스(Database)

서로 관련되는 데이터들을 일정한 체계와 순서로 모아 놓은 데이터의 집합을 의미 한다

DBMS(Data Base Management System)

데이터베이스 관리 시스템(DBMS)은 데이터베이스(Database)를 관리하는 소프트웨어를 의미 하며, 대표적인 기능은 데이터의 정의(Define), 생성, 조작, 공유, 보호, 유지 보수가 있다.

Oracle Express 21c 설치

www.oracle.com

컴퓨터이름이나 계정이 모두 영문(한글X), 공백X, 반드시 관리자 계정 이어야한다

1. 설치

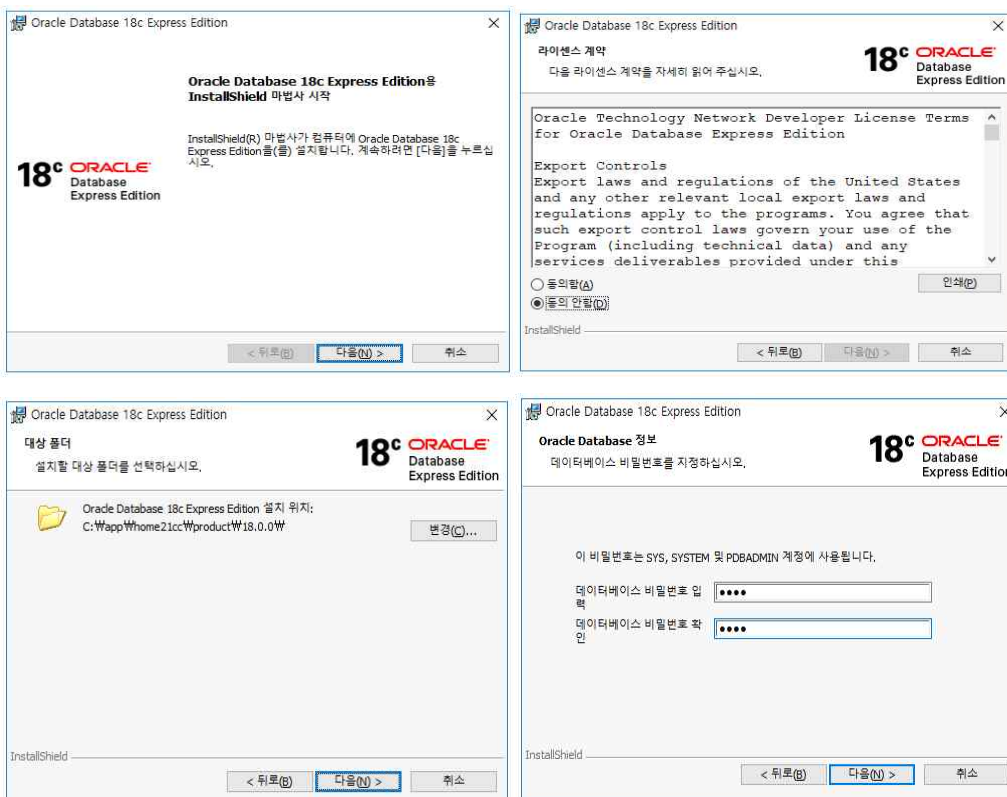
OracleXE213_Win64.zip 압축풀기

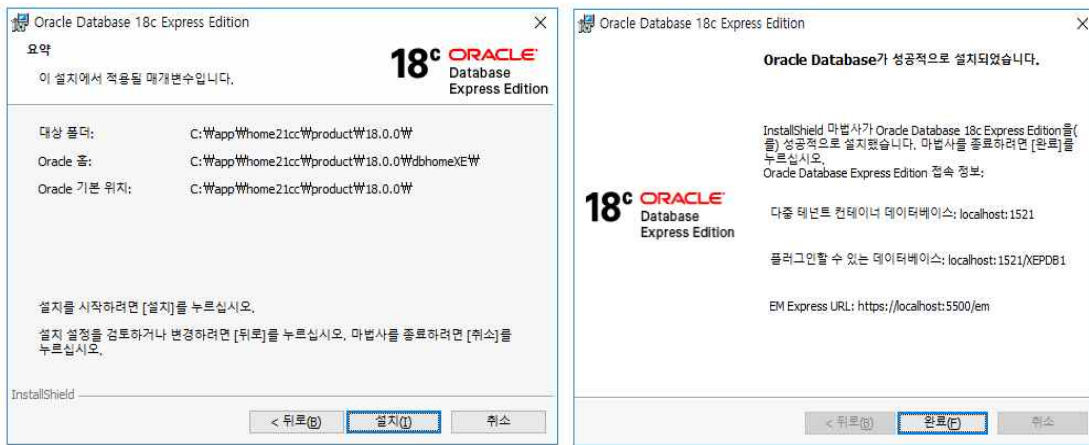


setup.exe 더블클릭



암호 : 『oracle』 입력 (SID : xe 기본설정)





2. 확인

① 시작 → 제어판 → 시스템 및 보안 → 관리도구 → 서비스

② 내PC → 우클릭 → 관리 → 서비스 및 응용 프로그램(왼쪽 창) → 서비스

| | | |
|-------------------------------|-----|----|
| OracleServiceXE | 시작됨 | 자동 |
| OracleOraDB21Home1TNSListener | 시작됨 | 자동 |

3. Listener 확인

: 2개의 파일에서 [HOST = pc이름][PORT = 1521] 확인한다

C:\app\903-21\product\21c\dbhomeXE\network\admin\sample

- listener.ora (서버)

: 클라이언트가 오라클 서버에 접속하기 위해서 서버 컴퓨터에 하는 설정

- tnsnames.ora (클라이언트)

: 클라이언트에서 오라클 서버에 접속할 때 필요한 설정

SQL

1. 방법

① 시작 → Oracle-OraDB21Home1 → SQL Plus

사용자명 입력: system

비밀번호 입력: oracle

SQL> show user;

USER은 "SYSTEM"입니다

SID (데이터베이스명) 확인

SQL> select name from v\$database;

NAME

XE

② 시작 → 실행 → cmd

C:\Users\user> sqlplus

사용자명 입력: system

비밀번호 입력: oracle

```
C:\Users\user> sqlplus system/oracle
```

③ 웹

<http://localhost:5560/isqlplus> ← Enterprise용

<http://localhost:8080/apex> ← Express용

<https://localhost:5500/em> ← Express 18c

Oracle Express 18c에서는 port가 5500 이다.

hr 계정

- 오라클 18c 버전 이후에는 연습용 스키마가 따로 분리되어 있어서 따로 설치해야 한다.
- 반드시 **관리자 계정(system)**에서 해야 한다
- cmd (관리자 권한으로 실행)

스키마 다운로드

<https://github.com/oracle-samples/db-sample-schemas/releases>

① 압축풀기

② C:\app\bitcamp\product\21c\dbhomeXE\demo\schema 폴더에 human_resources 복사한다.

③ > sqlplus system/oracle

SQL> select * from all_users;

hr계정이 안보인다

예전에는 lock으로 설정되어 있어서 unlock으로 풀어 써지만 지금은 hr 계정이 존재하지 않기 때문에 hr 계정을 만들어서 사용해야 한다

SQL> alter session set "_ORACLE_SCRIPT"=true;

세션이 변경되었습니다.

12c를 11g처럼 동작하게 만드는 것

컨테이너 데이터베이스를 만들지 않고 11g와 같은 사용자를 만들려면 12c 트릭은 문서화 되지 않은 매개 변수 _ORACLE_SCRIPT 를 true 로 설정하는 것

SQL> @?/demo/schema/human_resources/hr_main.sql

human_resources 폴더가 존재하는 위치

C:\app\bitcamp\product\21c\dbhomeXE\demo\schema\human_resources

specify password for HR as parameter 1:

1의 값을 입력하십시오: **hr**

specify default tablespace for HR as parameter 2: (기본 테이블 스페이스)

2의 값을 입력하십시오: **users**

specify temporary tablespace for HR as parameter 3: (임시 테이블 스페이스)

3의 값을 입력하십시오: **temp**

specify password for SYS as parameter 4: (system의 비밀번호 입력)

4의 값을 입력하십시오: **oracle**

specify log path as parameter 5: (로그 저장 디렉토리 지정)

5의 값을 입력하십시오: **\$ORACLE_HOME/demo/schema/log**

specify connect string as parameter 6: (접속 문자열 지정)

6의 값을 입력하십시오: **localhost:1521/x**e

다운로드가 시작된다.....

```
SP2-0310: unable to open file "__SUB__CWD__/human_resources/hr_cre.sql"
SP2-0310: unable to open file "__SUB__CWD__/human_resources/hr_popul.sql"
SP2-0310: unable to open file "__SUB__CWD__/human_resources/hr_idx.sql"
SP2-0310: unable to open file "__SUB__CWD__/human_resources/hr_code.sql"
SP2-0310: unable to open file "__SUB__CWD__/human_resources/hr_comnt.sql"
SP2-0310: unable to open file "__SUB__CWD__/human_resources/hr_analz.sql"
```

SQL>

* 마지막에 이런 error가 뜨는 경우

__SUB__CWD__는 현재 디렉토리를 저장하는 변수인데, 제대로 작동하지 못할 수 있다.

그러므로

C:\app\bitcamp\product\21c\dbhomeXE\demo\schema\human_resources\hr_main.sql 파일을 열어서

__SUB__CWD__ → **@?/demo/schema**로 수정하면 된다.

그리고 위의 명령문을 다시 수행한다.

SQL> @?/demo/schema/human_resources/hr_main.sql

SQL> conn hr/hr

연결되었습니다.

SQL> show user;

USER은 "HR"입니다

3번이나 또는 4번은 둘 중에서 1개만 선택해서 한다.

3. 사용자 계정 만들기

: 반드시 **관리자 계정(system)**에서 해야한다

계정명 : **c##java** (12c부터)

비밀번호 : bit

① SQL> create user **c##java** identified by 암호;

② SQL> grant create session, create table, create sequence, create view to **c##java**;

③ SQL> alter user **c##java** default tablespace users; -- users 가 테이블스페이스명이다

④ SQL> alter user **c##java** quota unlimited on users;

-- UNLIMITED는 사용자가 USERS 테이블스페이스가 완전히 채워질 때까지 데이터를 저장할 수 있음을 지정하는 키워드

테이블스페이스(TableSpace) 정의

테이블이 저장되는 공간입니다.

오라클 에서는 테이블스페이스라고 해서 테이블이 저장될 공간을 먼저 만들고 나서 테이블을 생성합니다. 각각의 테이블을 테이블스페이스별로 나누어서 관리와 퍼포먼스의 향상을 가지고 오는 것입니다.

테이블스페이스를 생성하면 정의된 용량만큼 미리 확보한 테이블스페이스가 생성되어지고 생성되어진 테이블스페이스에 테이블의 데이터가 저장됩니다.

예를 들면 많은 데이터가 쌓일 게시판 테이블은 기본용량 100MB, 자동 확장 10MB로 테이블스페이스를 만들어서 그곳에 게시판 테이블을 만들어 쓰면 게시판 데이터는 그곳에 100MB 까지 데이터가 저장되고 용량 초과 시 자동적으로 10MB 단위로 테이블 스페이스의 크기는 확장됩니다.

5. 계정 삭제

: 반드시 관리자 계정(system)에서 해야 한다

```
SQL> drop user 계정명
```

: 계정에 데이터(테이블)가 없을 경우

```
SQL> drop user 계정명 cascade;
```

: 계정에 데이터(테이블)가 있을 경우

6. 테이블 생성

```
create table 테이블명(필드명 필드타입, 필드명 필드타입,...);
```

: 테이블명, 필드명 한글 가능

[ex]

```
create table 연산(
```

```
x int, -- 고정형, int는 소수이하 자른다(반올림 된다)
```

```
y number, -- 가변형, number는 소수이하 뺀다
```

```
z number(10,3));
```

```
SQL> select * from tab; -- 테이블 보기
```

```
SQL> desc 연산;
```

```
create table dbtest(
```

```
name varchar2(15), -- char(고정형), varchar2(가변형)
```

```
age number,
```

```
height number(10,2),
```

```
logtime date);
```

Enterprise 한글 1자 = 2byte

Express 한글 1자 = 3byte

7. 레코드 추가

[형식]

```
insert into 테이블명(필드명,...) values(값,...);
```

insert into 테이블명 values(값,...); ← 테이블의 필드순서가 일치하면 필드명은 생략가능

[ex]

```
insert into 연산(x, y, z) values(25, 36, 12.34567);
insert into 연산(x, y, z) values(25.34567, 36.34567, 12.34567);
insert into 연산(x, y) values(25.666, 36.88888); -- 25.666 (반올림 되서 26 나온다)
insert into 연산(z,y,x) values(1, 2, 3); -- 순서가 바뀌어도 된다
```

```
insert into 연산 values(25, 36, 12.34567);
-- 필드명 생략 가능하게 되면 필드를 빠짐없이 순서대로 입력해야한다
insert into 연산 values(25, 36, 1234567.3456); -- 유효숫자는 최대7자리
insert into 연산 values(25, 36, 12345678.3456); -- error
```

```
-----
insert into dbtest(name,age,height,logtime) values('홍길동',25,185.567,sysdate);
insert into dbtest(name,age,height,logtime) values('Hong',30,175.56,sysdate);
insert into dbtest(name,age) values('희동이',3);
insert into dbtest(name, height) values('홍당무', 168.89);
insert into dbtest values('분홍신',5,123.5,sysdate);
insert into dbtest(name) values('진분홍');
```

8. 트랜잭션

: commit을 하기 전까지의 모든 명령어

insert, update, delete에 lock이 걸려있다

① commit - 갱신

② rollback

- 취소
- commit한 이후의 시점으로 돌아간다
- 현재 트랜잭션이 취소된다.

9. 레코드 추출

```
select 필드, 필드,... 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 * from 테이블명 where 조건;
```

```
select name, age from dbtest;
select * from dbtest;
select * from dbtest where name='홍길동';
select * from dbtest where name like '홍%';
select * from dbtest where name like '_홍%';
```

```
select * from dbtest where name like '__홍%';
select * from dbtest where name='hong'; -- 데이터는 대소문자 가린다
select * from dbtest where lower(name) = 'hong'; -- upper() 대문자
select * from dbtest where name like '%동%' and age<20;
select * from dbtest where age is null;
select * from dbtest where age is not null;
```

10. 레코드 수정

[형식]

update 테이블명 set 수정할 내용 where 조건

```
update dbtest set age=0 where name='홍당무';
update dbtest set age=0, height=0 where name='진분홍';
update dbtest set age=age+1;
update dbtest set age=age+1 where name='홍길동';
```

11. 레코드 삭제

[형식]

delete 테이블명; ← 모든 레코드 삭제
delete 테이블명 where 조건;

12. 테이블 삭제

```
drop table 테이블명;
drop table 테이블명 purge; ← 휴지통을 거치지 않고 바로 삭제
```

```
flashback table 테이블명 to before drop; ← 복원
purge recyclebin; ← 휴지통비우기
select * from recyclebin; → 휴지통에 테이블 정보 검색
show recyclebin;
```

시퀀스 객체

: 순차적으로 정수값을 자동으로 생성하는 객체

create sequence 시퀀스명

[increment by 증가값]

[start with 시작값]

[maxvalue 최대값 | minvalue 최소값]

[cycle | nocycle]

[cache | nocache]

- increment by 증가값 : 증가/감소 간격(기본값 : 1)
- start with : 시작번호(기본값 : 1)
- maxvalue/minvalue : 시퀀스의 최대/최소값 지정
- cycle/nocycle : 최대/최소값에 도달 시 반복여부 결정
- cache/nocache : 지정한 수량 만큼 메모리 생성여부 결정
(최소값 : 2, 기본값 : 20)

```
SQL> create sequence test increment by 2 start with 1 maxvalue 9 cycle nocache;
```

Sequence created.

```
SQL> select test.nextval from dual; -- dual 가상 테이블명
```

```
SQL> select test.currval from dual;
```

```
SQL> select * from user_sequences;
```

```
SQL> drop sequence test;
```

Sequence dropped.

1. Driver Loading - 1번 (*.jar이 필요)
2. Connection - URL, USER, PASSWORD
3. Statement - SQL(insert, delete, update, select...)
PreparedStatement

//-----

자바 1.6 이상부터는

Console 기반의 프로젝트에서 프로젝트 참조만으로 등록 사용 가능합니다

서비스로더(<http://docs.oracle.com/javase/6/docs/api/java/util/ServiceLoader.html>) 기반으로
JDBC Driver가 자동으로 등록됩니다.

그래서 Class.forName("com.mysql.jdbc.Driver") 류의 코드를 호출하지 않아도 됩니다.

//-----

ojdbc8.jar

필요한 Project의 lib에 넣기

[실습]

Java Project : dbtest

Package : dbtest.dao

Class : InsertTest.java

UpdateTest.java

SelectTest.java

[문제]

Project : student

Package : student.dao

Class : Student.java

테이블 작성

create table student(

name varchar2(15) not null, -- 이름

value varchar2(15), -- 학번 or 과목 or 부서

code number -- 1이면 학생, 2이면 교수, 3이면 관리자

);

menu()

관리

1. 입력
2. 검색
3. 삭제
4. 종료

번호선택 : 4

프로그램 종료합니다

1번인 경우

insertArticle()

1. 학생
2. 교수
3. 관리자
4. 이전메뉴

번호선택 :

1번 학생인 경우

이름입력 :

학번입력 :

2번 교수인 경우

이름입력 :

과목입력 :

3번 관리자인 경우

이름입력 :

부서입력 :

2번인 경우

selectArticle()

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

번호선택 : 1

1번 경우

검색할 이름 입력 : 이

이름=희동이 학번=2015

이름=이교수 과목=JAVA

3번인 경우

deleteArticle()

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

[문제]

Project : friend
Package : friend.action
Class : FriendManager.java
Package : friend.dao
Class : FriendDAO.java
Package : friend.bean
Class : FriendDTO.java

테이블 생성

```
create table friend(  
seq number primary key, -- 기본키(not null, unique)  
name varchar2(15) not null,  
tel1 varchar2(5),  
tel2 varchar2(5),  
tel3 varchar2(5),  
gender number, -- 남자는 0, 여자는 1  
read number default 0, -- 선택해제 0, 선택 1  
movie number default 0,  
music number default 0,  
game number default 0,  
shopping number default 0);
```

시퀀스

```
create sequence seq_friend nocycle nocache;
```