

데이터베이스 실습 1

실습 내용

- 오라클이 설치하여 제공한 가상머신을 다운로드 후 노트북 또는 PC 에 설치하는 과정을 수행
- 가상머신의 오라클을 구동하고, 가상머신 또는 터미널을 이용하여 오라클 데이터베이스에 접속
- 오라클 데이터베이스에 있는 SCOTT 계정의 테이블에 접속하여 주어진 쿼리문들을 실습 수행 후 결과 확인
- 실습 내용을 바탕으로 맨 마지막 페이지에 기재된 문제를 수행하고, 결과 값을 확인

오라클 가상 머신 PC 에 설치하기(윈도우 기준)

VirtualBox 다운로드 하여 설치하기

<http://www.virtualbox.org> 에 접속하여 최신 버전을 다운로드 받습니다.

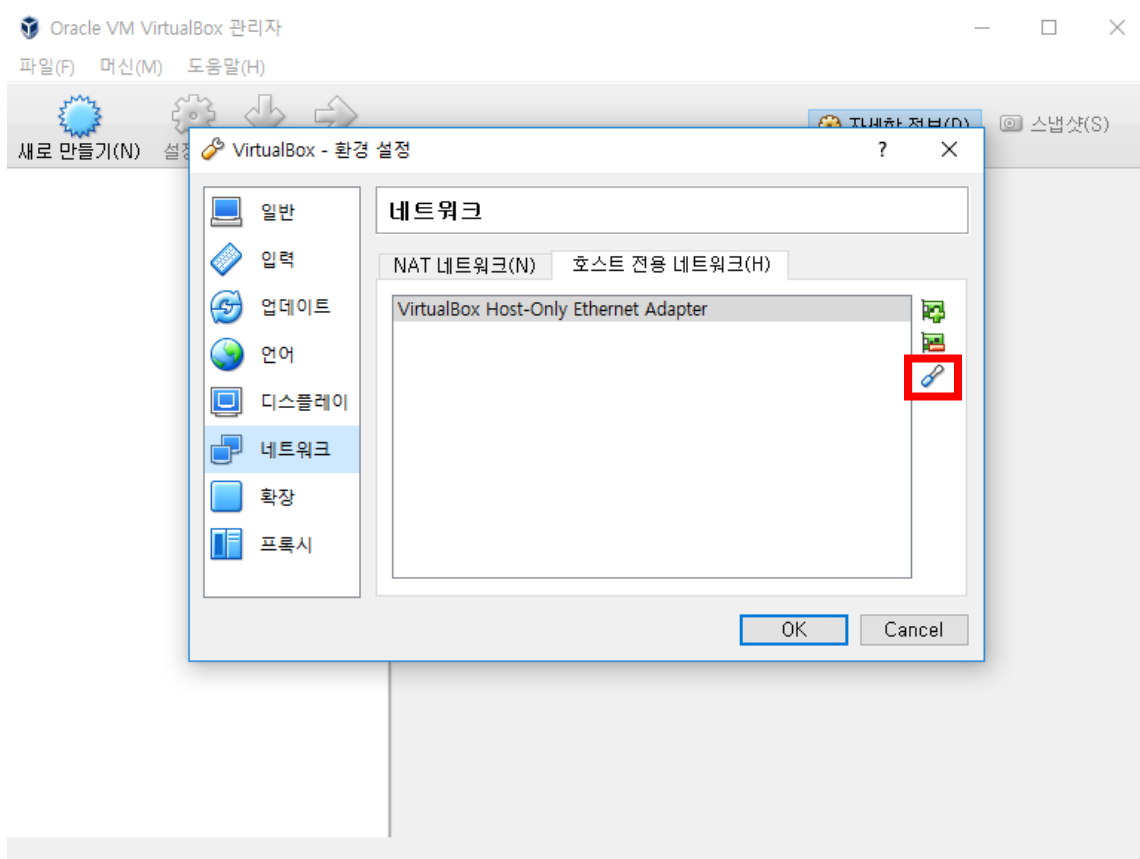


해당하는 OS 의 VirtualBox 를 받고, Extension 을 받아 설치합니다.



VirtualBox 네트워크 구성하기

VirtualBox 를 실행한 후, “파일 -> 환경설정 -> 네트워크”로 들어갑니다. 우측 가장 하단의 설정 메뉴를 선택합니다.



아래와 같이 설정합니다.

The image displays two screenshots of the '호스트 전용 네트워크 정보' (Hosts Dedicated Network Information) dialog box. The top screenshot shows the 'DHCP 서버(D)' (DHCP Server) tab with the following settings: IPv4 주소(I): 192.168.56.1, IPv4 서브넷 마스크(M): 255.255.255.0, IPv6 주소(P): (empty), and IPv6 네트워크 마스크 길이(L): (empty). The bottom screenshot shows the same dialog box with the '서버 사용(E)' (Server Use) checkbox checked, and the following settings: 서버 주소(R): 192.168.56.100, 서버 마스크(M): 255.255.255.0, 최저 주소 한계(L): 192.168.56.101, and 최고 주소 한계(U): 192.168.56.254. Both screenshots have 'OK' and 'Cancel' buttons at the bottom right.

가상머신 환경 다운로드 하기

가상머신 환경을 다운로드 합니다.

가상머신은 다음 둘 중 한 곳에서 다운로드 가능합니다.

<http://naver.me/FFVaj3kP> (패스워드: ssu2018) - 노트북 메모리 8 기가 이상 권장

<http://naver.me/FMZTHgdb> (패스워드: ssu2018) - 노트북 메모리 4 기가 권장

내려받기

네이버 클라우드에 저장

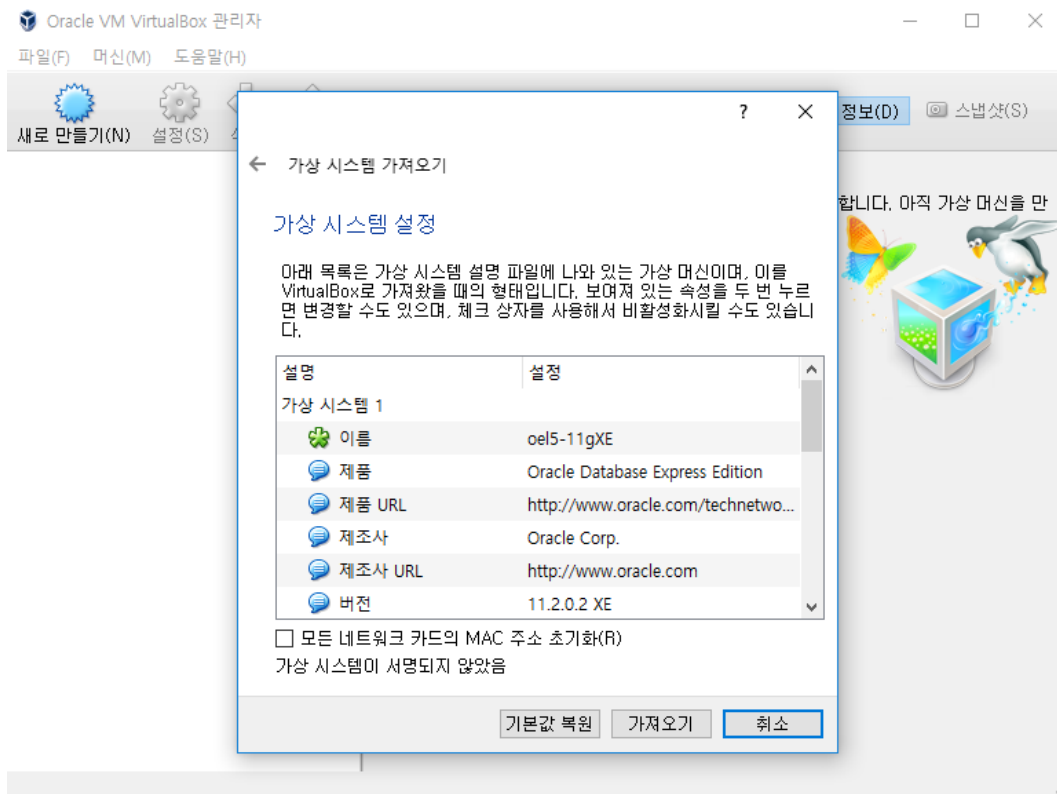
정렬기준: 수정한 날짜 - 최신 항목 우선 ↓

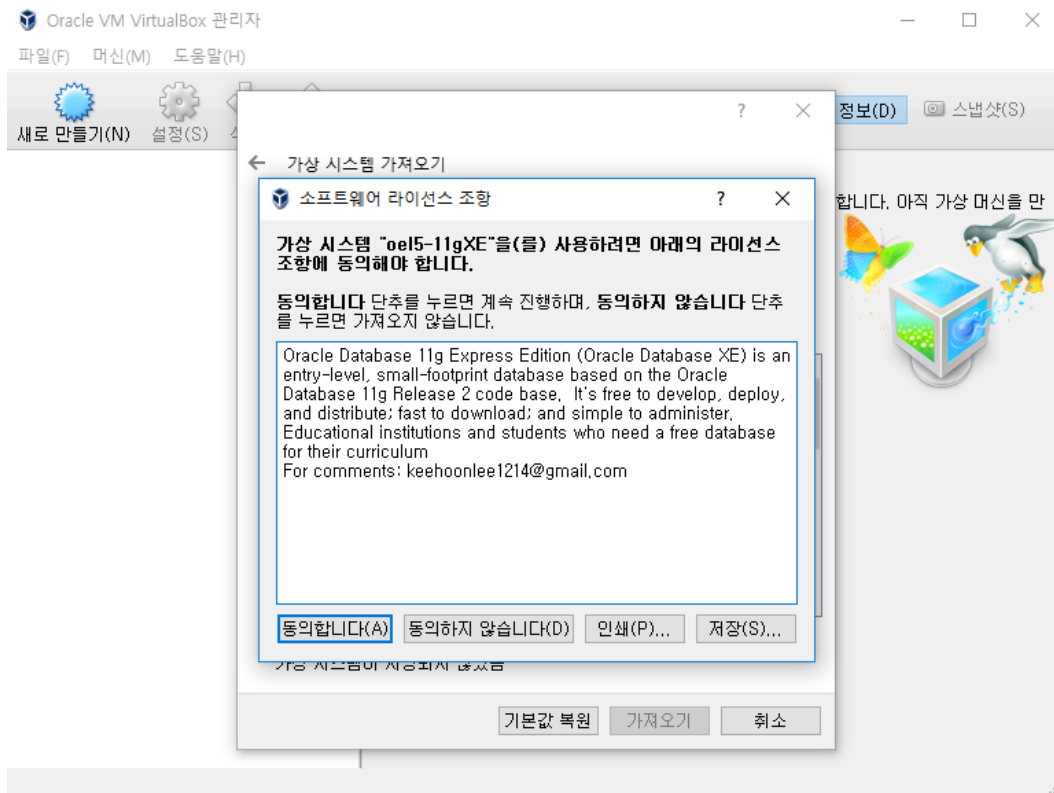


파일을 모두 다운로드 받은 후, DBMS_HUFS 압축을 해제합니다. 압축을 해제하면 DBMS_HUFS.ova 파일이 생성됩니다. (압축이 정상적으로 해제되면, 압축파일은 삭제해도 됩니다.)

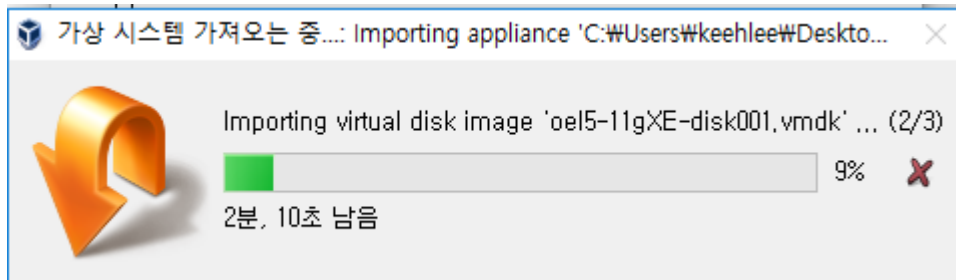
가상머신 가져오기(Import)

VirtualBox 에서 “파일 -> 가상시스템 가져오기” 를 선택한 후 ova 이미지를 선택

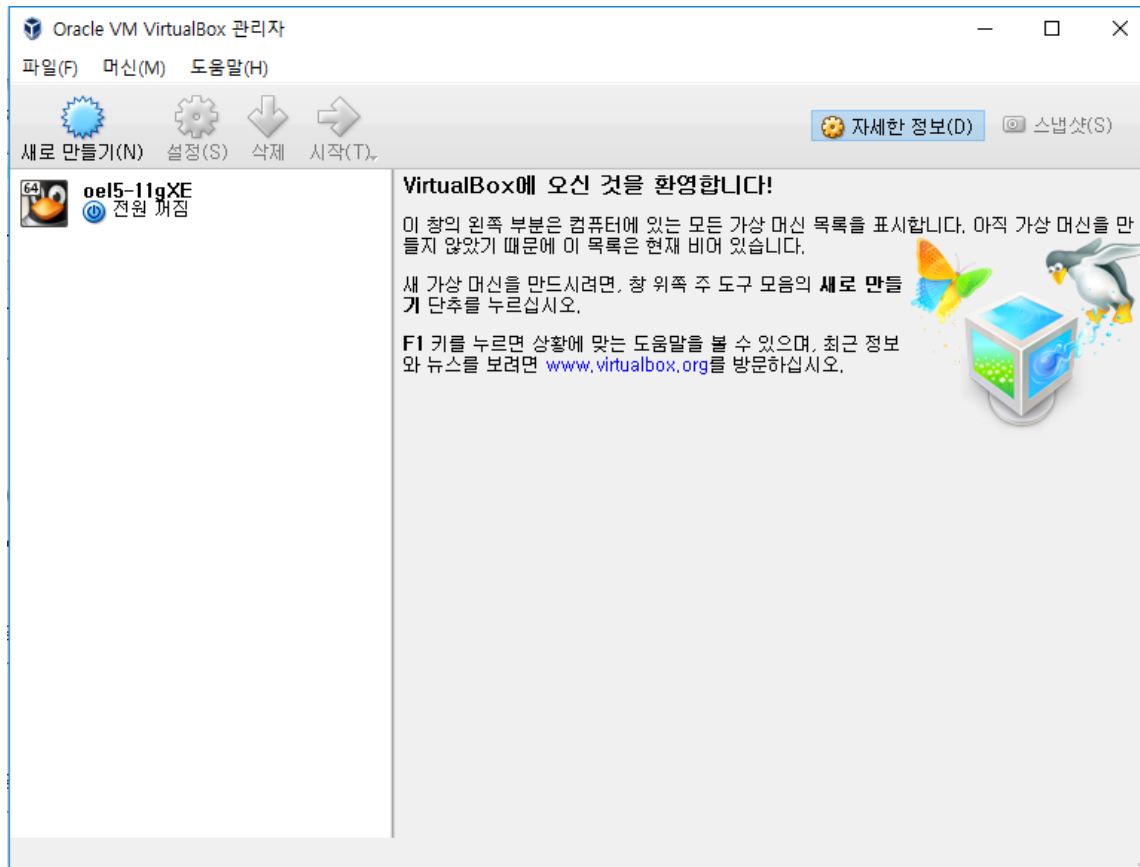




동의하기를 선택하면 가상머신을 가져오는 단계가 시작됩니다.



이제 가상머신 가져오기가 완료되었습니다.

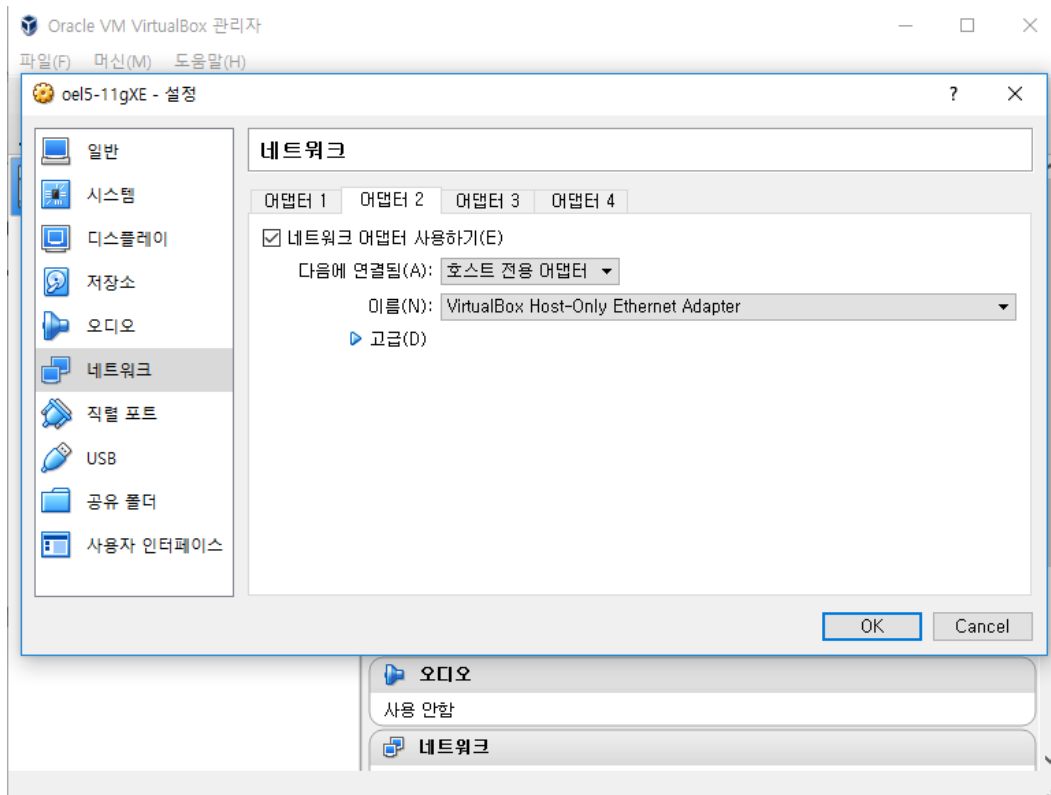


가상머신 구동하여 접속하기

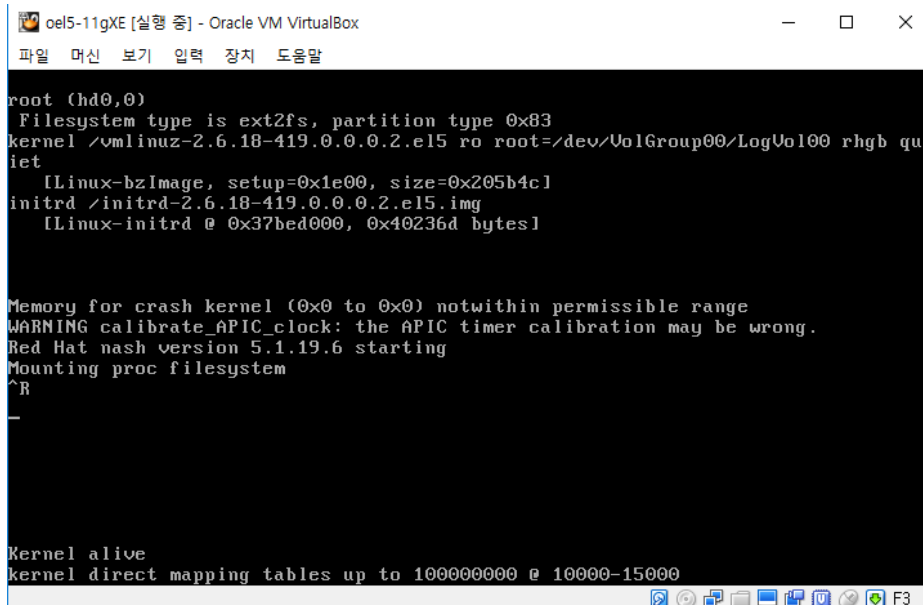
가상머신 구동하기

가상머신은 간단히 생성된 가상머신을 더블클릭하면 실행이 가능합니다. (다만 머신이 Mac 환경에서 생성되어 네트워크 변경이 필요합니다.)

oel5-11gXE 를 우클릭 하여 "설정"->"네트워크"->"어댑터 2"로 들어가 아래와 같이 설정을 확인한 후 OK 를 선택합니다.



머신을 더블클릭 하면, 가상머신이 시작됩니다.



(간혹 BIOS 에서 Hypervisor 설정이 안되서 바로 꺼지는 경우가 있습니다. 이는 인터넷에서 BIOS Hypervisor 설정하기를 검색하시길 바랍니다.)

이제 가상머신이 구동됩니다. 계정은 **root**, 패스워드는 **oracle** 로 설정되어 있습니다.

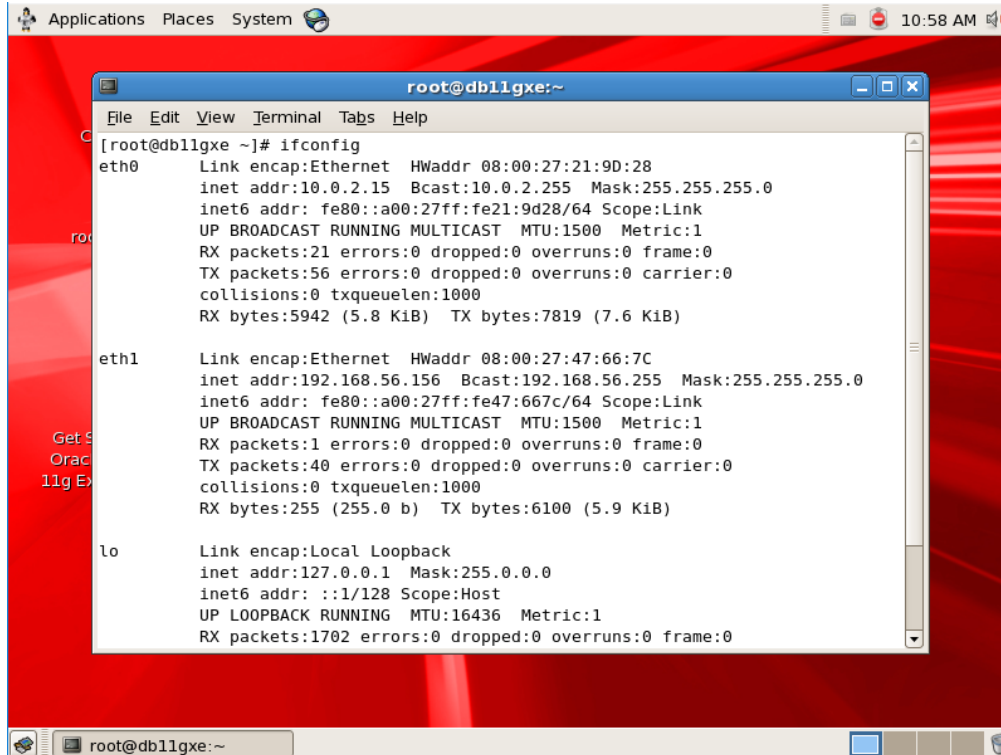


이제 구동이 완료되었습니다.



Putty 접속하기

원활한 실습을 위해 터미널을 사용할 수 있도록 설정되어 있습니다.

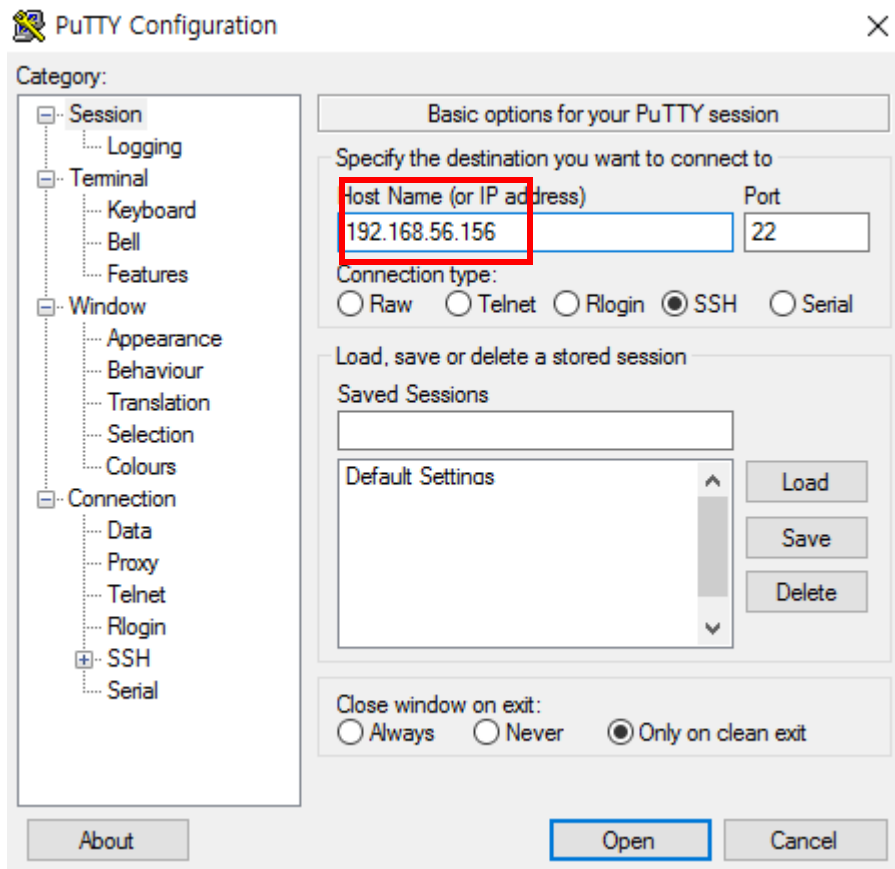


```
root@db1lgxe:~  
File Edit View Terminal Tabs Help  
[root@db1lgxe ~]# ifconfig  
eth0      Link encap:Ethernet  HWaddr 08:00:27:21:9D:28  
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0  
          inet6 addr: fe80::a00:27ff:fe21:9d28/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:21 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:56 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:5942 (5.8 KiB)  TX bytes:7819 (7.6 KiB)  
  
eth1      Link encap:Ethernet  HWaddr 08:00:27:47:66:7C  
          inet addr:192.168.56.156  Bcast:192.168.56.255  Mask:255.255.255.0  
          inet6 addr: fe80::a00:27ff:fe47:667c/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:40 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:255 (255.0 b)  TX bytes:6100 (5.9 KiB)  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          inet6 addr: ::1/128 Scope:Host  
          UP LOOPBACK RUNNING  MTU:16436  Metric:1  
          RX packets:1702 errors:0 dropped:0 overruns:0 frame:0
```

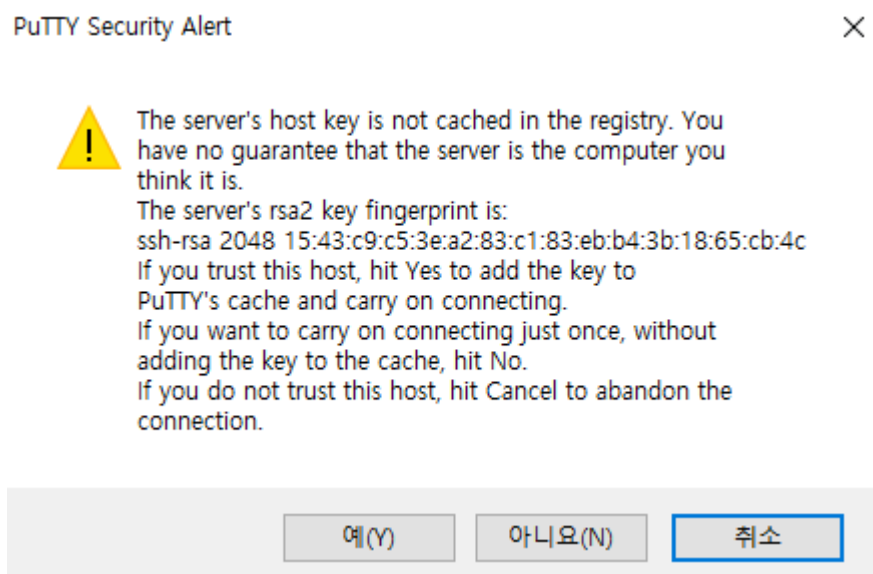
처음에 함께 다운로드 한 Putty.exe 를 실행합니다.



아래와 같이 접속 정보를 입력한 후, OPEN 을 선택합니다.



처음에 키 등록 화면이 나옵니다. “예”를 선택합니다.



계정 정보는 마찬가지로 **root, oracle** 입니다.

```
root@db11gxe:~  
login as: root  
root@192.168.56.156's password:  
Last login: Wed Aug 23 10:56:23 2017  
[root@db11gxe ~]#
```

데이터베이스 접속하기

데이터베이스 접속하기

데이터베이스는 OS 가 시작될 때 함께 구동되기로 설정되어 있습니다. 구동되어 있는 데이터베이스에 접속하면 됩니다. (음영체 부분을 입력하면 됩니다.)

현재 오라클 **sys**(시스템 관리자)의 패스워드는 **oracle_4U** 로 되어있습니다.

```
root [~] $ sqlplus sys/oracle_4U as sysdba  
  
SQL*Plus: Release 11.2.0.2.0 Production on Mon Jun 10 11:49:47 2019  
  
Copyright (c) 1982, 2011, Oracle. All rights reserved.  
  
Connected to:  
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production  
  
SQL>
```

아래와 같이 데이터베이스 정보를 조회할 수 있습니다.

```
SQL> select name, version_time from v$database;  
  
NAME    VERSION_TIME  
-----  
XE      22-AUG-17  
  
SQL> exit  
Disconnected from Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit  
Production
```

SCOTT 스키마(계정) 접속하여 실습하기

SCOTT 스키마 접속하기

SCOTT 은 모든 데이터베이스에 기본적인 실습을 위해 존재하는 샘플 스키마 입니다. 이에 접속하여 간단한 쿼리를 수행해보도록 하겠습니다.

스키마 접속은 scott/tiger 로 수행하면 됩니다.

```
root [~] $ sqlplus scott/tiger
```

```
SQL*Plus: Release 11.2.0.2.0 Production on Mon Jun 10 11:50:41 2019
```

```
Copyright (c) 1982, 2011, Oracle. All rights reserved.
```

```
Connected to:
```

```
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
```

```
SQL> show user
```

```
USER is "SCOTT"
```

쿼리문 수행해보기 - 테이블 조회

SCOTT 스키마 내에 있는 테이블을 조회하는 쿼리는 다음과 같습니다.

```
SQL> select * from tab;
```

TNAME	TABTYPE	CLUSTERID

BONUS	TABLE	
DEPT	TABLE	
EMP	TABLE	
SALGRADE	TABLE	

만약 몇 개의 테이블이 있는지 확인하고 싶으면, count() 함수를 활용하면 됩니다.

```
SQL> select count(*) from tabs;
```

COUNT(*)

4

테이블의 내용을 조회하고 싶으면 다음과 같이 입력합니다.

```
SQL> select * from emp;
```

```
SQL> select * from dept;
```

```
SQL> select * from bonus;
SQL> select * from salgrade;
```

각 테이블의 레코드 수를 확인하려면, 마찬가지로 count() 함수를 씁니다.

```
SQL> select count(*) from emp;
SQL> select count(*) from dept;
SQL> select count(*) from bonus;
SQL> select count(*) from salgrade;
```

쿼리문 수행해보기 - 테이블 구조 조회하기

테이블이 어떤 구조로 되어 있는지 확인하기 위해서는 desc(ribe) 명령어를 활용합니다.

여기서는 emp 테이블의 구조를 확인해보겠습니다.

```
SQL> desc emp;
Name                Null?  Type
-----
EMPNO               NOT NULL NUMBER(4)
ENAME               VARCHAR2(10)
JOB                 VARCHAR2(9)
MGR                 NUMBER(4)
HIREDATE            DATE
SAL                 NUMBER(7,2)
COMM                NUMBER(7,2)
DEPTNO              NUMBER(2)
```

```
SQL> describe emp;
Name                Null?  Type
-----
EMPNO               NOT NULL NUMBER(4)
ENAME               VARCHAR2(10)
JOB                 VARCHAR2(9)
MGR                 NUMBER(4)
HIREDATE            DATE
SAL                 NUMBER(7,2)
COMM                NUMBER(7,2)
DEPTNO              NUMBER(2)
```

쿼리문 수행해보기 - 간단한 조건문 조회

조건문을 조회하기 위해서는 WHERE 절을 이용합니다.

아래는 empno(사번), ename(이름), job(직업), 고용일(hiredate)를 emp 테이블에서 읽습니다.

조건) 직업이 SALESMAN 인 경우를 찾는 쿼리입니다.

```
SQL> select empno, ename, job, hiredate
      from emp
      where job='SALESMAN'
      ;
```

EMPNO	ENAME	JOB	HIREDATE
7499	ALLEN	SALESMAN	20-FEB-81
7521	WARD	SALESMAN	22-FEB-81
7654	MARTIN	SALESMAN	28-SEP-81
7844	TURNER	SALESMAN	08-SEP-81

조건) 사번이 7800 번 이하에 해당하는 경우를 찾는 쿼리입니다.

```
SQL> select empno, ename, job, hiredate
      from emp
      where empno <= 7800;
```

EMPNO	ENAME	JOB	HIREDATE
7369	SMITH	CLERK	17-DEC-80
7499	ALLEN	SALESMAN	20-FEB-81
7521	WARD	SALESMAN	22-FEB-81
7566	JONES	MANAGER	02-APR-81
7654	MARTIN	SALESMAN	28-SEP-81
7698	BLAKE	MANAGER	01-MAY-81
7782	CLARK	MANAGER	09-JUN-81
7788	SCOTT	ANALYST	19-APR-87

8 rows selected.

조건) 직업이 SALESMAN 이고, 사번이 7800 번 이하를 동시에 만족하는 경우를 찾는 쿼리입니다. 이 경우는 AND 를 이용합니다.

```
SQL> select empno, ename, job, hiredate
      from emp
      where job='SALESMAN'
      and empno <= 7800;
```

EMPNO	ENAME	JOB	HIREDATE
7499	ALLEN	SALESMAN	20-FEB-81
7521	WARD	SALESMAN	22-FEB-81
7654	MARTIN	SALESMAN	28-SEP-81

조건) 고용일이 1981 년 3 월 1 일부터 1981 년 10 월 31 일 사이에 있는 경우를 찾는
경우입니다.

```
SQL> select empno, ename, job, hiredate
      from emp
      where hiredate between to_date ('1981/03/01', 'yyyy/mm/dd') and to_date
('1981/10/31', 'yyyy/mm/dd');
```

EMPNO	ENAME	JOB	HIREDATE
7566	JONES	MANAGER	02-APR-81
7654	MARTIN	SALESMAN	28-SEP-81
7698	BLAKE	MANAGER	01-MAY-81
7782	CLARK	MANAGER	09-JUN-81
7844	TURNER	SALESMAN	08-SEP-81

LIKE 조건과 SUM 함수 이용하기

LIKE 는 확실하지 않은 데이터 (일부의 값만 아는 경우)에 사용하는 조건입니다.

조건) 이름이 M 으로 시작하는 사람의 사번, 이름, 직업, 연봉을 출력하는 쿼리

```
SQL> select empno, ename, job, sal
      from emp
      where ename like 'M%';
```

EMPNO	ENAME	JOB	SAL
7654	MARTIN	SALESMAN	1250
7934	MILLER	CLERK	1300

조건) 모든 직원의 연봉의 합을 출력하는 쿼리

```
SQL> select sum(sal) from emp;
```

SUM(SAL)
29025

조건) 직업이 SALESMAN 인 직원의 연봉의 합을 출력하는 쿼리

```
SQL> select sum(sal) from emp
      where job='SALESMAN';
```

SUM(SAL)
5600

간단한 조인쿼리

조인은 여러 테이블을 같이 보는 경우에 활용됩니다.

SCOTT의 스키마에는 EMP와 DEPT가 있는데, 이를 조회해 보면 구조가 아래와 같습니다.

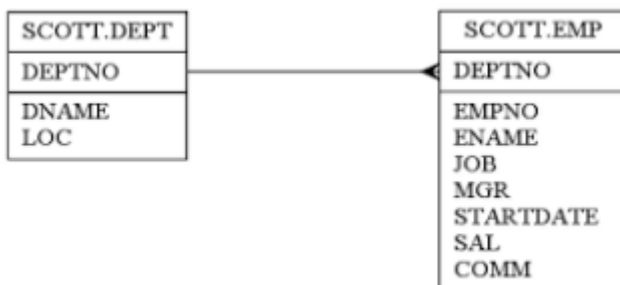
```
SQL> desc emp;
```

Name	Null?	Type
EMPNO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)
MGR		NUMBER(4)
HIREDATE		DATE
SAL		NUMBER(7,2)
COMM		NUMBER(7,2)
DEPTNO		NUMBER(2)

```
SQL> desc dept;
```

Name	Null?	Type
DEPTNO	NOT NULL	NUMBER(2)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)

조회해서 보면, DEPTNO는 두 테이블에 모두 존재하고 있습니다. 이는 두 테이블이 연관이 있음을 보여주며, 그림으로 나타내면 다음과 같습니다. (상세 내용은 추후 수업에서 다룸)



한번 DEPT 테이블을 조회해보겠습니다.

```
SQL> select * from dept;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS

30 SALES	CHICAGO
40 OPERATIONS	BOSTON

이번엔, EMP 테이블에 사번, 이름, 직업, 연봉, DEPTNO(부서번호)를 조회해보겠습니다.

SQL> **select empno, ename, job, deptno from emp;**

EMPNO	ENAME	JOB	DEPTNO
7369	SMITH	CLERK	20
7499	ALLEN	SALESMAN	30
7521	WARD	SALESMAN	30
7566	JONES	MANAGER	20
7654	MARTIN	SALESMAN	30
7698	BLAKE	MANAGER	30
7782	CLARK	MANAGER	10
7788	SCOTT	ANALYST	20
7839	KING	PRESIDENT	10
7844	TURNER	SALESMAN	30
7876	ADAMS	CLERK	20

EMPNO	ENAME	JOB	DEPTNO
7900	JAMES	CLERK	30
7902	FORD	ANALYST	20
7934	MILLER	CLERK	10

14 rows selected.

EMP 와 DEPT 를 동시에 조회해서, DEPTNO 가 일치시켜 DEPTNO 대신 DNAME(부서명)과 LOC(위치)를 넣어보겠습니다. 이를 조인쿼리라고 합니다.

출력하는 값은 사번, 이름, 직업, 부서명, 위치 가 될 것입니다.

SQL> **select e.empno, e.ename, e.job, d.dname, d.loc
from emp e, dept d
where e.deptno = d.deptno;**

EMPNO	ENAME	JOB	DNAME	LOC
7369	SMITH	CLERK	RESEARCH	DALLAS
7499	ALLEN	SALESMAN	SALES	CHICAGO
7521	WARD	SALESMAN	SALES	CHICAGO
7566	JONES	MANAGER	RESEARCH	DALLAS
7654	MARTIN	SALESMAN	SALES	CHICAGO
7698	BLAKE	MANAGER	SALES	CHICAGO
7782	CLARK	MANAGER	ACCOUNTING	NEW YORK

```

7788 SCOTT ANALYST RESEARCH DALLAS
7839 KING PRESIDENT ACCOUNTING NEW YORK
7844 TURNER SALESMAN SALES CHICAGO
7876 ADAMS CLERK RESEARCH DALLAS

```

```

EMPNO ENAME JOB DNAME LOC
-----

```

```

7900 JAMES CLERK SALES CHICAGO
7902 FORD ANALYST RESEARCH DALLAS
7934 MILLER CLERK ACCOUNTING NEW YORK

```

14 rows selected.

다음은 직업이 SALESMAN 이면서, CHICAGO 에 위치한 직원들의 사번, 이름, 직업, 부서명, 위치를 출력하는 쿼리입니다. (위의 쿼리에 조건절을 추가)

```

SQL> select e.empno, e.ename, e.job, d.dname, d.loc
      from emp e, dept d
      where e.deptno = d.deptno
      and e.job='SALESMAN' and d.loc='CHICAGO'
      ;

```

```

EMPNO ENAME JOB DNAME LOC
-----

```

```

7499 ALLEN SALESMAN SALES CHICAGO
7521 WARD SALESMAN SALES CHICAGO
7654 MARTIN SALESMAN SALES CHICAGO
7844 TURNER SALESMAN SALES CHICAGO

```

SCOTT 스키마 구조 파악하기 (EMP, DEPT)

SCOTT 테이블 구조 파악하기

SCOTT 의 테이블을 조회합니다. 4 개의 테이블을 확인할 수 있지만, 본 실습에서는 EMP 와 DEPT 에 대한 실습을 수행할 것입니다.

```

SQL> select * from tab;

```

```

TNAME          TABTYPE CLUSTERID
-----
BONUS          TABLE
DEPT           TABLE
EMP            TABLE
SALGRADE       TABLE

```

EMP 테이블의 구조를 확인합니다.

```
SQL> desc emp;
```

Name	Null?	Type
EMPNO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)
MGR		NUMBER(4)
HIREDATE		DATE
SAL		NUMBER(7,2)
COMM		NUMBER(7,2)
DEPTNO		NUMBER(2)

DEPT 테이블의 구조를 확인합니다.

```
SQL> desc dept
```

Name	Null?	Type
DEPTNO	NOT NULL	NUMBER(2)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)

SCOTT 스키마 제약 조건 파악하기

제약 조건은 ALL_CONSTRAINTS 에서 확인이 가능합니다. 다만, 전체 조회를 하게 되면 현재 계정인 SCOTT 이 접근 가능한 모든 제약 조건이 출력됩니다. 아래를 조회해보면 현재 SCOTT 은 189 개의 제약 조건에 조회 권한이 있는 것을 확인할 수 있습니다.

```
SQL> select count(*) from all_constraints;
```

COUNT(*)

189

ALL_CONSTRAINTS 의 구조를 확인합니다. 상대적으로 필요한 정보라고 여겨지는 부분에 형광 표시를 하였습니다.

```
SQL> desc all_constraints;
```

Name	Null?	Type
OWNER		VARCHAR2(120)
CONSTRAINT_NAME	NOT NULL	VARCHAR2(30)
CONSTRAINT_TYPE		VARCHAR2(1)
TABLE_NAME	NOT NULL	VARCHAR2(30)

SEARCH_CONDITION	LONG
R_OWNER	VARCHAR2(120)
R_CONSTRAINT_NAME	VARCHAR2(30)
DELETE_RULE	VARCHAR2(9)
STATUS	VARCHAR2(8)
DEFERRABLE	VARCHAR2(14)
DEFERRED	VARCHAR2(9)
VALIDATED	VARCHAR2(13)
GENERATED	VARCHAR2(14)
BAD	VARCHAR2(3)
RELY	VARCHAR2(4)
LAST_CHANGE	DATE
INDEX_OWNER	VARCHAR2(30)
INDEX_NAME	VARCHAR2(30)
INVALID	VARCHAR2(7)
VIEW_RELATED	VARCHAR2(14)

OWNER 를 SCOTT 으로 형광 표기한 컬럼만 조회를 수행해봅니다.

```
SQL> select OWNER, CONSTRAINT_NAME, CONSTRAINT_TYPE, TABLE_NAME,
R_OWNER, R_CONSTRAINT_NAME
from all_constraints
where owner='SCOTT';
```

OWNER

CONSTRAINT_NAME C TABLE_NAME

R_OWNER

R_CONSTRAINT_NAME

SCOTT

FK_DEPTNO R EMP

SCOTT

PK_DEPT

OWNER

CONSTRAINT_NAME C TABLE_NAME

R_OWNER

R_CONSTRAINT_NAME

SCOTT

PK_DEPT P DEPT

OWNER	

CONSTRAINT_NAME	C TABLE_NAME

R_OWNER	

R_CONSTRAINT_NAME	

SCOTT	
PK_EMP	P EMP

출력이 깔끔하지 않으므로, 화면을 조정하겠습니다. 이때는 SET LINESIZE 와 COL
<컬럼명> FOR <포맷형> 명령어를 사용합니다. 그리고 최근 쿼리를 재수행 할 때는 “/”만
입력하면 됩니다.

```
SQL> set linesize 150
SQL> col owner for a10
SQL> col constraint_name for a15
SQL> col table_name for a10
SQL> col r_owner for a10
SQL> col r_constraint_name for a15
SQL> /
```

OWNER	CONSTRAINT_NAME	C	TABLE_NAME	R_OWNER	R_CONSTRAINT_NAME

SCOTT	FK_DEPTNO	R	EMP	SCOTT	PK_DEPT
SCOTT	PK_DEPT	P	DEPT		
SCOTT	PK_EMP	P	EMP		

세 개의 제약 조건이 걸려 있으며, PK 2 개와 FK 1 개를 확인할 수 있습니다.
어느 컬럼에 제약 조건이 있는지는 ALL_CONS_COLUMNS 를 확인하면 됩니다.
ALL_CONS_COLUMNS 의 구조를 확인해보겠습니다.

```
SQL> desc ALL_CONS_COLUMNS;
Name          Null?  Type
-----
OWNER          NOT NULL VARCHAR2(30)
CONSTRAINT_NAME NOT NULL VARCHAR2(30)
TABLE_NAME     NOT NULL VARCHAR2(30)
COLUMN_NAME    VARCHAR2(4000)
POSITION      NUMBER
```

마찬가지로 SCOTT 스키마에 대해 조회해 보겠습니다.

```
SQL> set linesize 100
SQL> col column_name for a15
SQL> select OWNER, CONSTRAINT_NAME, TABLE_NAME, COLUMN_NAME
       from ALL_CONS_COLUMNS
       where owner='SCOTT';
```

OWNER	CONSTRAINT_NAME	TABLE_NAME	COLUMN_NAME
SCOTT	PK_DEPT	DEPT	DEPTNO
SCOTT	PK_EMP	EMP	EMPNO
SCOTT	FK_DEPTNO	EMP	DEPTNO

여기에서는, PK 가 DEPT 에는 DEPT_NO 라는 컬럼으로, EMP 에는 EMP_NO 라는 컬럼으로, 그리고 FK(외래키)가 EMP 테이블의 DEPT_NO 로 걸려 있음을 확인할 수 있습니다.

이를 수월하게 보기 위해서 ALL_CONS_COLUMNS 와 ALL_CONSTRAINTS 를 조인합니다.

ORDER BY (컬럼 수)는 몇 번째 컬럼의 오름차순에 따라 정리할지를 정의합니다. 아래의 경우는 혼선을 방지하기 위해 TABLE_NAME 이 두번째에 있으므로 ORDER BY 2 를 추가로 입력합니다.

```
SQL> select a.owner, a.table_name, b.column_name, a.constraint_name ,
       a.constraint_type, a.r_owner, a.r_constraint_name
       from all_constraints a, all_cons_columns b
       where a.constraint_name=b.constraint_name and a.owner='SCOTT'
       order by 2;
```

OWNER	TABLE_NAME	COLUMN_NAME	CONSTRAINT_NAME	C	R_OWNER	R_CONSTRAINT_NAME
SCOTT	DEPT	DEPTNO	PK_DEPT	P		
SCOTT	EMP	DEPTNO	FK_DEPTNO	R	SCOTT	PK_DEPT
SCOTT	EMP	EMPNO	PK_EMP	P		

```
SQL> exit
Disconnected from Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit
Production
```

연습 문제

과제는 MUSIC 스키마의 내용을 아래의 조건들에 맞춰서 과제를 수행한 결과 값을 캡처하는 것입니다. MUSIC 스키마의 패스워드는 MUSIC 으로 설정되어 있습니다.

<문제 1> MUSIC 스키마에는 ALBUMS 와 GENRE 두 개의 테이블이 있습니다. 각 테이블의 COLUM 의 수와 레코드의 개수를 구하시오.

<문제 2> ALBUMS 테이블에서 SOLD(판매량)이 '1000 장 미만'인 앨범의 개수를 구하시오.
(Hint: count() 이용)

<문제 3> ALBUMS 테이블에서 다음 조건을 모두 만족하는 ANAME(앨범 이름)을 찾고, 그 쿼리와 결과를 캡처하여 첨부하십시오. (Hint: WHERE 절과 AND 의 사용)

- (i) AID(식별자)가 50000 번 미만
- (ii) PRICE(가격)이 280 초과
- (iii) ARTIST_FIRST(아티스트 이름)이 "Zn" 으로 시작
- (iv) REVIEW(평점)이 3 점 미만
- (v) SOLD(판매량)이 20000 초과

<문제 4> ALBUMS 와 GENRE 테이블에서 다음 조건을 모두 만족하는 앨범들의 PRICE 가격의 합을 구하고, 그 쿼리와 결과를 첨부하십시오. (Hint: 조인, sum() 함수 이용)

- (i) GENRE(장르)는 Musical
- (ii) LENGTH(앨범 길이)는 90 초과
- (iii) SOLD(판매)는 80000 초과