

# 01 오라클과 데이터베이스

이번 장에서는 데이터 저장 장소인 데이터베이스(Database)와 관리하고자 하는 모든 데이터를 서로 연관관계를 가진 테이블(표) 형식으로 저장하는 관계형 데이터베이스(Relational Database)에 대해서 학습합니다.

## 학습 내용

- ❖ 데이터베이스 관리 시스템이란?
- ❖ 오라클 설치
- ❖ SQL와 데이터 조회하기
- ❖ DUAL 테이블

## 학습목표

- ❖ 오라클의 개념을 이해하고 설명할 수 있습니다.
- ❖ 오라클을 설치할 수 있습니다.
- ❖ 데이터베이스의 개념에 대해 이해하고 설명할 수 있습니다.
- ❖ 기본적인 SELECT 문을 구사할 수 있습니다.

# 데이터베이스 관리 시스템이란?

## ❖ 데이터베이스

- 조직화된 정보의 집합

## ❖ 데이터베이스 관리 시스템

- 연관성 있는 데이터들의 집합을 효율적으로 응용하기 위해 구성된 소프트웨어들의 집합체

## 관계형 데이터베이스

- ❖ 오라클(Oracle), 사이베이스(Sybase), 인포믹스(Infomix), MYSQL, Access, SQL Server
- ❖ 관계형 데이터베이스 정보를 테이블 형태로 저장합니다.
- ❖ 테이블은 2차원 형태의 표처럼 볼 수 있도록 로우(ROW:행)와 칼럼(COLUMN:열)으로 구성합니다.

테이블명 : DEPARTMENT

칼럼

칼럼명

로우

DNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

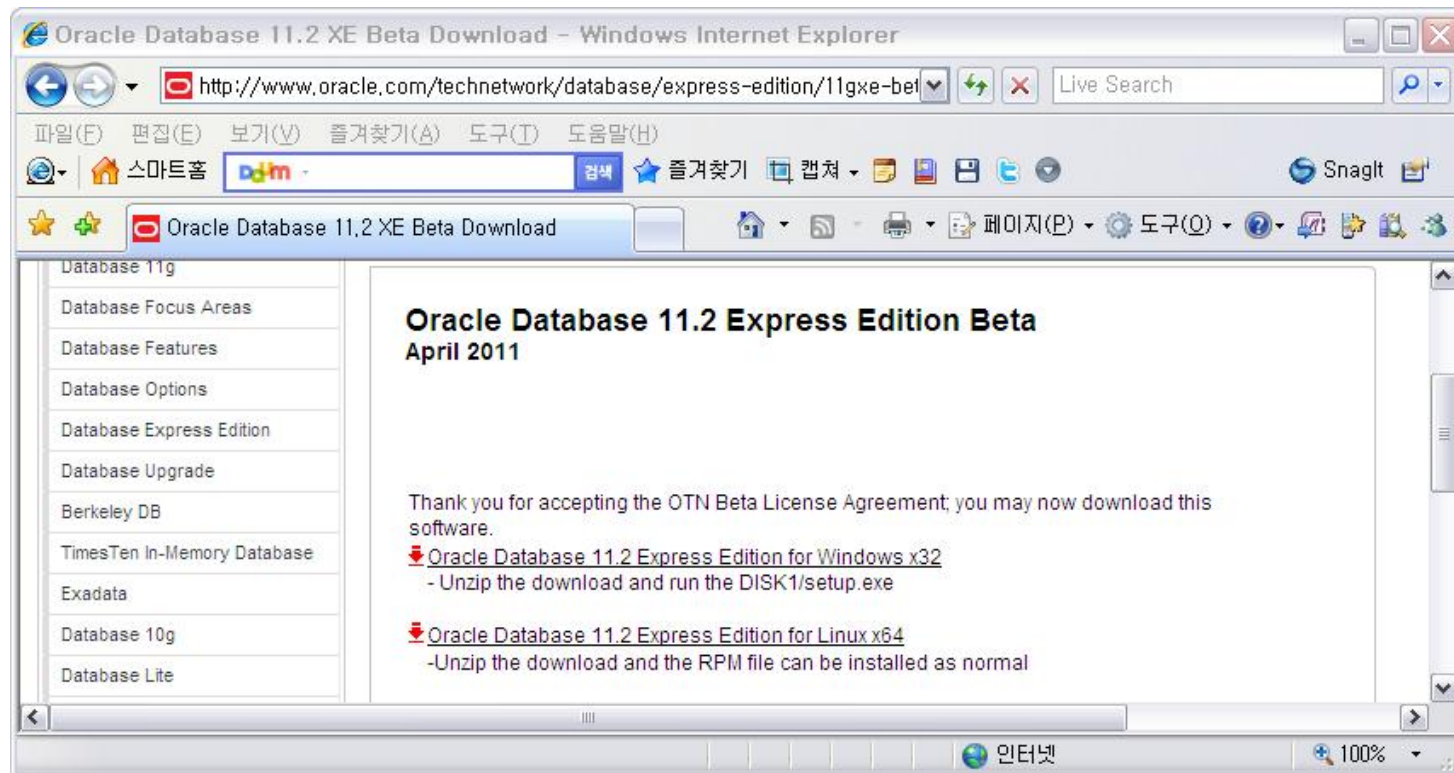
SQL>

- ❖ DEPARTMENT 테이블은 4개의 로우와 3개의 칼럼(부서번호:DNO, 부서이름:DNAME, 지역:LOC 으로 구성된 테이블입니다.

# 오라클 다운받기

- ❖ 웹 사이트에 접속하여 오라클을 다운받을 수 있습니다. 해당 OS에 맞는 것을 선택합니다.

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



# SQL와 데이터 조회하기

## ❖ SQL(Structured Query Language)

- 데이터베이스에 저장된 데이터를 조회, 입력, 수정 삭제하는 등의 조작이나, 테이블을 비롯한 다양한 객체(시퀀스, 인덱스 등)를 생성 및 제어하는 역할을 합니다.

## ❖ SQL의 종류

- 데이터 정의어(DDL)
  - 데이터베이스 관리자나 응용 프로그래머가 데이터베이스의 논리적 구조를 정의하기 위한 언어로서 데이터 사전(Data Dictionary)에 저장 됩니다.
- 데이터 조작어(DML)
  - 데이터베이스에 저장된 데이터를 조작하기 위해 사용하는 언어로서 데이터 검색(Retrieval), 추가(Insert), 삭제>Delete), 갱신(Update) 작업 수행 합니다.
- 데이터 제어어(DCL)
  - 데이터에 대한 접근 권한 부여 등의 데이터베이스 시스템의 트랜잭션을 관리하기 위한 목적으로 사용되는 언어입니다.

# SQL와 데이터 조회하기

유형	명령문
DQL:Data Query Language(질의어)	SELECT(데이터 검색시 사용)
DML:Data Manipulation Language (데이터 조작어)-데이터 변경시 사용	INSERT(데이터 입력) UPDATE(데이터 수정) DELETE(데이터 삭제)
DDL:Data Definition Language (데이터 정의어) 객체 생성및 변경시 사용	CREATE(데이터베이스 생성) ALTER(데이터베이스 변경) DROP(데이터베이스 삭제) RENAME(데이터베이스 객체이름 변경) TRUNCATE(데이터베이스 저장 공간 삭제)
TCL:Transaction Control Language (트랜잭션 처리어)	COMMIT(트랜잭션의 정상적인 종료처리) ROLLBACK(트랜잭션 취소) SAVEPOINT(트랜잭션내에 임시 저장점 설정)
DCL:Data Control Language (데이터 제어어)	GRANT(데이터베이스에 대한 일련의 권한 부여) REVOKE(데이터베이스에 대한 일련의 권한 취소)



## 오라클 접속하기

- ❖ 데이터베이스 사용자는 오라클 계정(Account)이라는 용어와 같은 의미로 사용됩니다. 오라클을 설치하면 한개 이상의 데이터베이스 권한을 갖는 디폴트(기본적인) 사용자가 존재합니다. 오라클에서 제공되는 사용자 계정은 다음과 같습니다.

사용자계정	설명
SYS	오라클 Super 사용자 계정이며 데이터베이스에서 발생하는 모든 문제들을 처리할 수 있는 권한을 가지고 있다.
SYSTEM	오라클 데이터베이스를 유지보수 관리할 때 사용하는 사용자 계정이며, SYS 사용자와 차이점은 데이터베이스를 생성할 수 있는 권한이 없으면 불완전 복구를 할 수 없다.
SCOTT	처음 오라클을 사용하는 사용자의 실습을 위해 만들어 놓은 연습용 계정이다.
HR	이 역시 오라클에 접근할 수 있도록 샘플로 만들어 놓은 사용자 계정이다.

# 오라클 접속하기

## ❖ Command 환경에서 SQL\*Plus 로그인

- 데이터베이스 접속을 시도하면 오라클 데이터베이스를 사용할 수 있는 사용자인지를 검증하기 위해서 사용자 계정과 암호를 묻게 됩니다.

형식	SQLPLUS 사용자계정/암호
----	------------------

- 오라클을 설치하게 되면 기본적으로 생성되는 사용자 계정 중에서 hr을 사용합니다. hr의 패스워드는 1234입니다.

예	sqlplus hr/1234□
---	------------------

## 테이블 구조를 살펴보기 위한 DESC

- ❖ 테이블에서 데이터를 조회하기 위해서는 테이블의 구조를 알아야 합니다. 테이블의 구조를 확인하기 위한 명령어로는 DESCRIBE가 있습니다.

형식	DESC[RIBE] 테이블명
----	-----------------

- DESC 명령어는 테이블의 컬럼명, 데이터 형, 길이와 NULL 허용 유무 등과 같은 특정 테이블의 정보를 알려줍니다.

# 데이터를 조회하기 위한 SELECT 문

❖ SELECT 문은 데이터를 조회하기 위한 SQL 명령어입니다.

형식	<pre>SELECT [DISTINCT] {*, column[Alias], ...} FROM table_name [WHERE condition] [GROUP BY group_by_expression] [HAVING group_condition] [ORDER BY column];</pre>
----	---

절	기능
SELECT 절	조회하고자 하는 칼럼명의 리스트를 나열합니다.
DISTINCT 절	동일한 내용을 한 번씩만 출력하여 중복을 제거합니다.
FROM 절	조회하고자 하는 테이블명의 리스트를 나열합니다.
WHERE 절	조회하고자 하는 로우의 조건을 나열합니다.
GROUP BY 절	동일한 값을 갖는 로우들을 한 그룹으로 묶습니다.
HAVING 절	로우들의 그룹이 만족해야 하는 조건을 제시합니다.
ORDER BY 절	로우들의 정렬 순서를 제시합니다,

## 데이터를 조회하기 위한 SELECT 문

❖ 다음은 테이블의 내용을 살펴보기 위한 쿼리문입니다.

예

```
select *  
from employee; □ □
```

- SELECT는 데이터베이스 내에 저장되어있는 테이블을 조회하기 위한 명령어입니다.
- SELECT 다음에는 보고자 하는 대상의 칼럼명을 기술합니다. SELECT 다음에 \*을 기술하면 지정된 테이블의 모든 칼럼을 조회합니다.
- FROM 다음에는 보고자 하는 대상의 테이블 이름을 기술합니다.

## 특정 데이터만 보기

- ❖ **사원번호에 해당되는 칼럼명은 empno이고 사원명에 해당되는 칼럼명은 ename입니다. empno와 ename을 출력하기 위해서는 출력하고자 하는 순서대로 기술하되 칼럼과 칼럼 사이에 콤마를 기술합니다.**

예

```
select empno, ename  
from employee;
```

## 산술 연산자

종류	예
+	<code>select salary + 300 from employee</code>
-	<code>select salary - 300 from employee</code>
*	<code>select salary * 12 from employee</code>
/	<code>select salary / 4 from employee</code>

- ❖ 급여로 연봉 계산을 해보도록 합시다. 일반적으로 연봉은 급여를 12번 곱한 것이므로 연봉을 구하기 위해서 산술 연산자를 사용합시다.

예	<code>SELECT ename, salary, salary*12 FROM employee;</code>
---	---

## nvl 함수 사용하기

- ❖ 넓은 ? 혹은  $\infty$ 의 의미이기 때문에 연산, 할당, 비교가 불가능합니다.

$$100 + ? = ?$$
$$100 + \infty = \infty$$

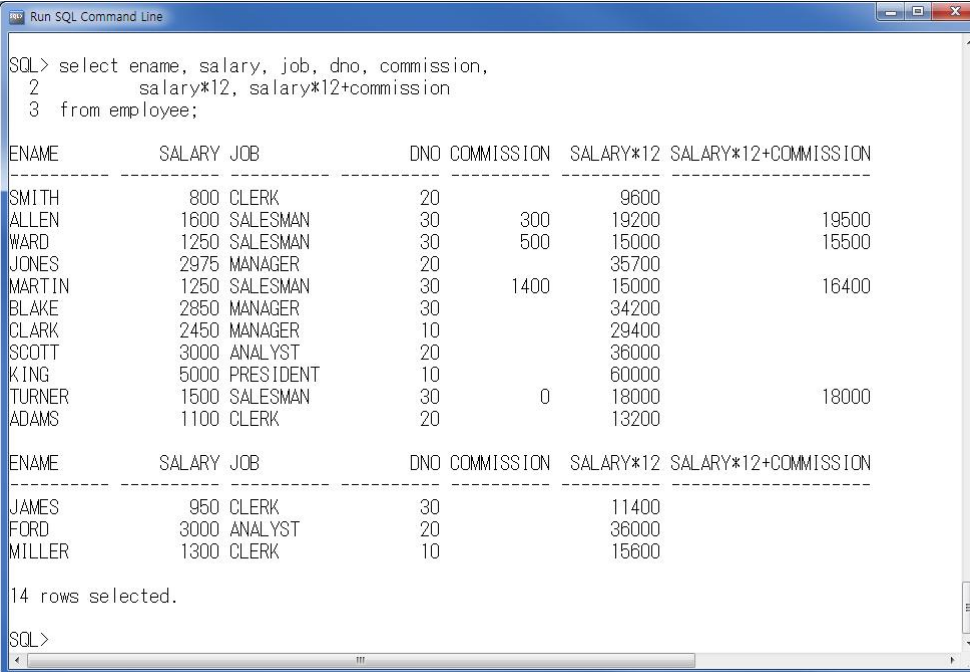
- ❖ 다음은 산술 연산에 NULL을 사용하는 경우에는 특별한 주의가 필요합니다. 이를 설명하기 위해서 연봉 계산에 커미션을 더하는 예를 살펴봅시다.



# nvl 함수 사용하기

예

```
select ename, salary, job, dno, commission,  
       salary*12, salary*12+commission  
from employee;
```



```
SQL> select ename, salary, job, dno, commission,  
2          salary*12, salary*12+commission  
3          from employee;
```

ENAME	SALARY	JOB	DNO	COMMISSION	SALARY*12	SALARY*12+COMMISSION
SMITH	800	CLERK	20		9600	
ALLEN	1600	SALESMAN	30	300	19200	19500
WARD	1250	SALESMAN	30	500	15000	15500
JONES	2975	MANAGER	20		35700	
MARTIN	1250	SALESMAN	30	1400	15000	16400
BLAKE	2850	MANAGER	30		34200	
CLARK	2450	MANAGER	10		29400	
SCOTT	3000	ANALYST	20		36000	
KING	5000	PRESIDENT	10		60000	
TURNER	1500	SALESMAN	30	0	18000	18000
ADAMS	1100	CLERK	20		13200	
JAMES	950	CLERK	30		11400	
FORD	3000	ANALYST	20		36000	
MILLER	1300	CLERK	10		15600	

14 rows selected.

```
SQL>
```

영업직인 경우 커미션 칼럼에 값이 저장되어 있으므로 제대로 연봉 계산을 하게 된다.

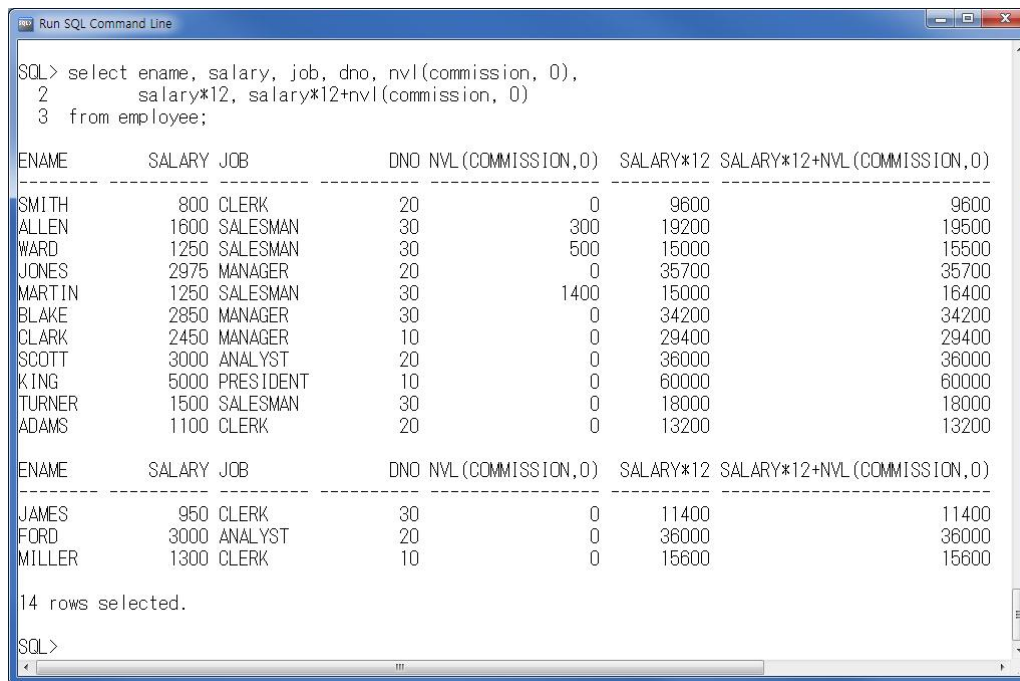
영업직인 경우 무능력하여 받을 커미션이 없더라도 0으로 저장되어 있으므로 연봉 계산이 제대로 된다.

영업직인 아닌 경우에는 커미션에 널 값이 저장되어 있어서 연봉 계산 결과도 널값으로 구해지는 모순이 발생한다.

# nvl 함수 사용하기

예

```
select ename, salary, job, dno, nvl(commission, 0),  
salary*12, salary*12+nvl(commission, 0)  
from employee;
```



```
SQL> select ename, salary, job, dno, nvl(commission, 0),  
2 salary*12, salary*12+nvl(commission, 0)  
3 from employee;
```

ENAME	SALARY	JOB	DNO	NVL (COMMISSION,0)	SALARY*12	SALARY*12+NVL (COMMISSION,0)
SMITH	800	CLERK	20	0	9600	9600
ALLEN	1600	SALESMAN	30	300	19200	19500
WARD	1250	SALESMAN	30	500	15000	15500
JONES	2975	MANAGER	20	0	35700	35700
MARTIN	1250	SALESMAN	30	1400	15000	16400
BLAKE	2850	MANAGER	30	0	34200	34200
CLARK	2450	MANAGER	10	0	29400	29400
SCOTT	3000	ANALYST	20	0	36000	36000
KING	5000	PRESIDENT	10	0	60000	60000
TURNER	1500	SALESMAN	30	0	18000	18000
ADAMS	1100	CLERK	20	0	13200	13200
JAMES	950	CLERK	30	0	11400	11400
FORD	3000	ANALYST	20	0	36000	36000
MILLER	1300	CLERK	10	0	15600	15600

14 rows selected.

```
SQL>
```

연봉 계산을 위해 사원 테이블에서 급여와 커미션 칼럼을 살펴본 결과 영업사원이 아닌 사원들의 커미션은 NULL로 지정되어 있으므로 연봉을 올바르게 계산하기 위해서는 커미션이 NULL인 경우 0으로 변경하여 계산에 참여하도록 해야 합니다.

오라클에서는 NULL을 0 또는 다른 값으로 변환하기 위해서 사용하는 함수로 NVL을 제공합니다. 커미션에 널이 저장되어 있더라도 널을 다른 값으로 변환하는 NVL 함수를 사용하면 제대로 된 계산 결과를 얻을 수 있습니다.

## 칼럼에 별칭 지정하기

- ❖ SQL에서 쿼리문의 결과가 출력될 때, 칼럼명이 칼럼에 대한 헤딩(heading)으로 출력됩니다.

예

```
select ename, salary*12+nvl(commission, 0) 연봉  
from employee;
```

## AS 로 칼럼에 별칭 부여하기

- ❖ 칼럼명 대신 별칭을 출력하고자 하면 칼럼을 기술한 바로 뒤에 AS 라는 키워드를 쓴 후 별칭을 기술합니다.

예

```
select ename, salary*12+nvl(commission, 0) as 연봉  
from employee;
```

## “ ” 로 별칭 부여하기

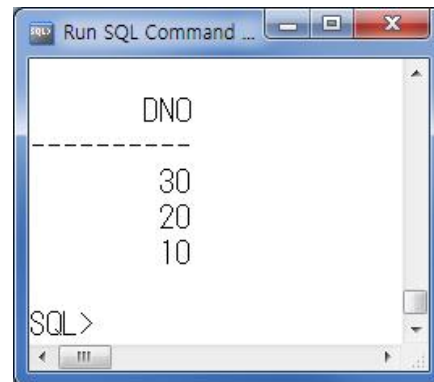
- ❖ 칼럼명과 별칭 사이에 공백을 추가하거나 특수문자( \$, \_, #, 등)를 추가하거나 대소문자를 구분해야 하는 경우에는 ""을 사용하여 별칭을 부여해야 합니다.

```
select ename, salary*12+nvl(commission, 0) "연 봉"  
from employee;
```

## 중복된 데이터를 한번씩 만 출력하게 하는 DISTINCT

- ❖ 중복되는 부서번호를 한 번씩만 출력하기 위해서는 키워드 **DISTINCT**를 사용합니다.

```
select distinct dno  
from employee;
```



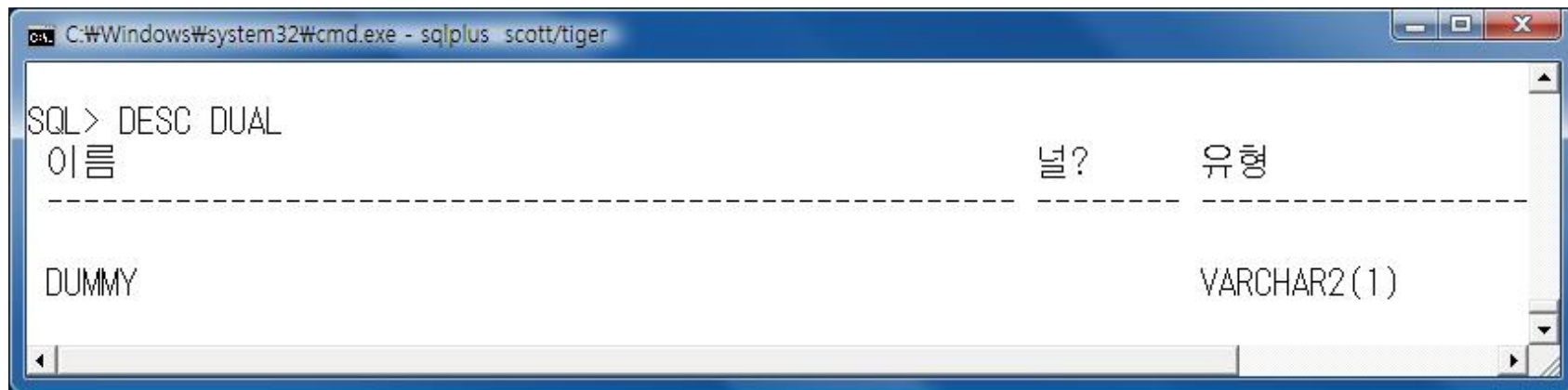
## DUAL 테이블

- ❖ DUAL 테이블의 구조를 살펴보기 위해서 DESC 명령어를 사용합니다.

예

**DESC DUAL**

- ❖ DUAL 테이블은 DUMMY라는 단 하나의 칼럼으로 구성되어 있습니다.
- ❖ 이 칼럼에는 최대 길이는 1입니다.



```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger

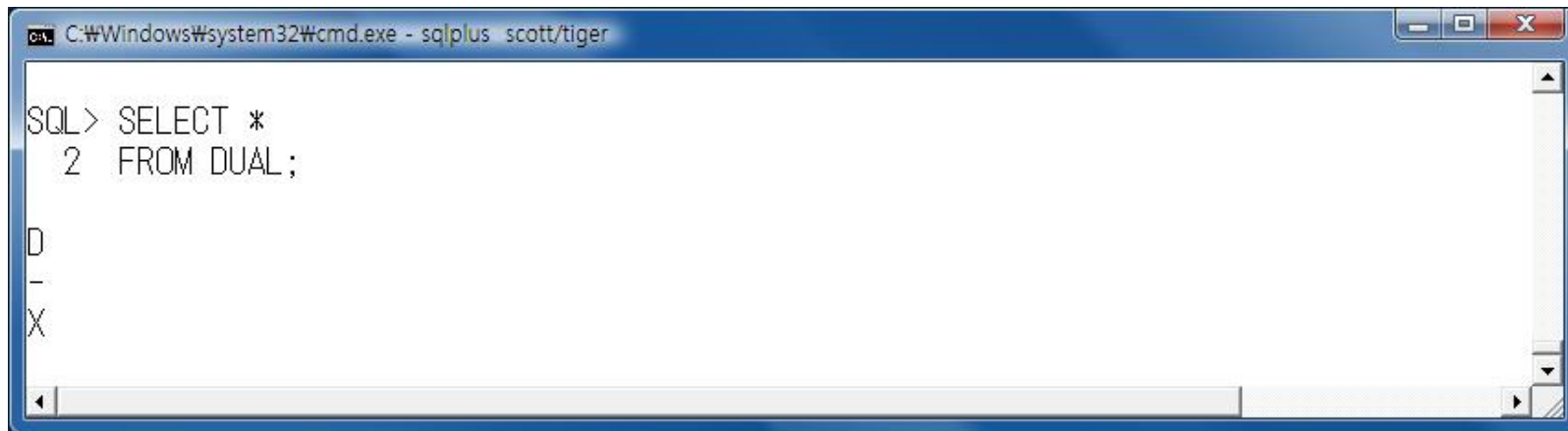
SQL> DESC DUAL
이름                                널?       유형
-----
DUMMY                                YES        VARCHAR2(1)
```

## DUAL 테이블

❖ DUMMY 칼럼엔 과연 어떤 값이 저장되어 있는 것일까요?

예

```
SELECT *  
FROM DUAL;
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The prompt is "SQL>". The user has entered the query "SELECT \*  
2 FROM DUAL;". The output displayed is a single row with a single column containing the value "X".

```
SQL> SELECT *  
2  FROM DUAL;  
  
D  
-  
X
```

- ❖ DUAL 테이블은 DUMMY라는 단 하나의 칼럼에 X라는 단 하나의 로우만을 저장하고 있으나 이 값은 아무런 의미가 없습니다.
- ❖ 쿼리문의 수행 결과가 하나의 로우로 출력되도록 하기 위해서 단 하나의 로우를 구성하고 있을 뿐입니다.