

이 장에서 다룰 내용



- 1 **CREATE TABLE로 테이블 구조 정의하기**
.....●
- 2 **ALTER TABLE로 테이블 구조 변경하기**
.....●
- 3 **DROP TABLE로 테이블 구조 삭제하기**
.....●
- 4 **테이블의 모든 로우를 제거하는 TRUNCATE**
.....●
- 5 **테이블 명을 변경하는 RENAME**
.....●

01. 테이블 구조 정의하는 CREATE TABLE



- ❖ 지금까지는 오라클에서 학습용으로 제공해 주는 테이블을 사용하여 다양한 조회를 해 보았습니다.
- ❖ 이번 장에서는 DDL(Data Definition Language)을 사용하여 테이블 구조 자체를 새롭게 생성, 수정, 삭제해 보겠습니다.
- ❖ 우선 CREATE TABLE 명령어로 새로운 테이블을 생성해 보겠습니다.
- ❖ 다음은 CREATE TABLE 문의 기본 형식입니다.

```
CREATE TABLE table_name  
(column_name data_type, ...);
```

1.1 데이터 형



이름	비고
CHAR(size)	고정 길이 문자를 저장합니다. (최대 2000바이트)
VARCHAR2(size)	가변 길이 문자 저장합니다. (최대 4000바이트) 실제 입력된 문자열의 길이만큼 저장 영역을 차지합니다.
NUMBER(w)	W자리까지의 수치로 최대 38자리까지 가능합니다.
NUMBER(w, d)	W는 전체 길이, d는 소수점 이하 자릿수이다. 소수점은 자릿수에 포함되지 않는다.
DATE	BC 4712년 1월 1일~AD 9999년 12월 31일까지의 날짜
LONG	LOB타입이 나오기 전에 사용하던 문자 타입으로 가변 길이의 문자형 데이터 타입, 최대 크기는 2GB
LOB	4GB까지의 가변 길이 바이너리 데이터를 저장시킬 수 있습니다. 이 이미지 문서, 실행 파일을 저장할 수 있습니다.

식별자 명명 규칙

테이블 명과 컬럼 명과 같이 사용자가 이름을 부여하는 것을 식별자라고 한다.

식별자를 부여하기 위한 규칙은 다음과 같다.

1. 반드시 문자로 시작해야 한다.(1~30자까지 가능)
2. 대소문자와 숫자 특수 기호(underscore, dollar, hash)만 포함 가능, 공백 허용 안함
3. 오라클에서 사용되는 예약어나 다른 객체명과 중복 불가하다.

4. 사용예)

A01_ 32 (공백 사용-잘못된 표기)

AxAb106

94168A32_\$(숫자로 시작 -잘못된 표기)

A_6#_2829

〈실습하기〉 새롭게 테이블 생성하기

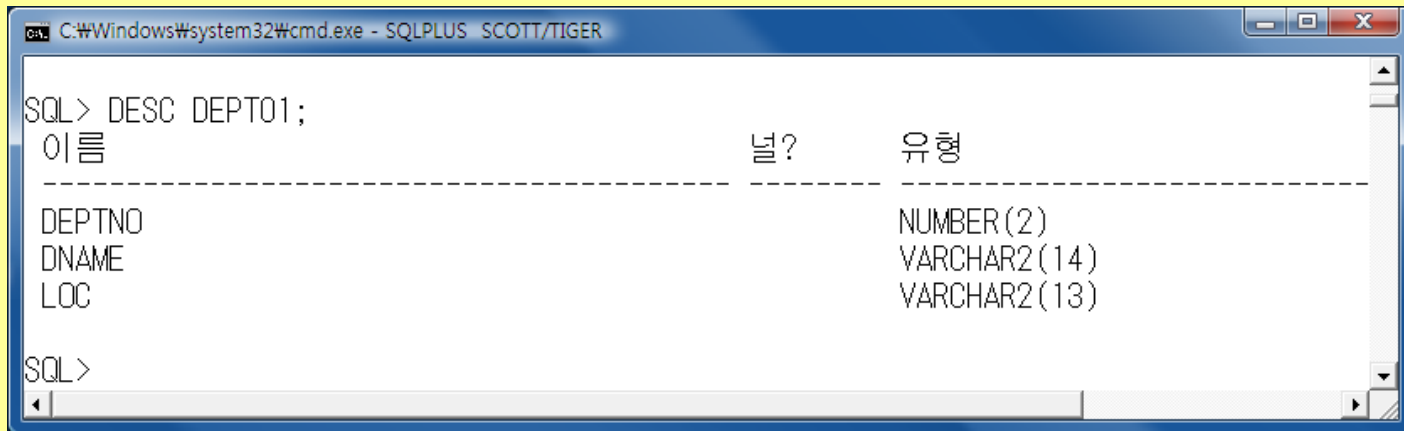
지금까지 실습에 사용했던 사원 테이블과 유사한 구조의 사원번호, 사원 이름, 급여 3개의 칼럼으로 구성된 EMP01 테이블을 생성해 봅시다.

1. CREATE TABLE 명령어로 EMP01 테이블을 새롭게 생성합니다.

```
CREATE TABLE EMP01(  
    EMPNO NUMBER(4),  
    ENAME VARCHAR2(20),  
    SAL NUMBER(7, 2)  
);
```

<탄탄히 다지기>

1. 다음과 같은 구조의 테이블을 CREATE TABLE 명령어로 생성하되 테이블의 이름은 DEPT01 로 하시오.)



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - SQLPLUS SCOTT/TIGER". The user has entered the command "SQL> DESC DEPT01;". The output displays the table structure for DEPT01, including column names, widths, and data types.

이름	널?	유형
DEPTNO		NUMBER(2)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)

SQL>

〈실습하기〉 서브 쿼리로 테이블 생성하기

CREATE TABLE 문에서 서브 쿼리를 사용하여 이미 존재하는 테이블과 동일한 구조와 내용을 갖는 새로운 테이블을 생성할 수 있습니다.

1. CREATE TABLE 명령어 다음에 컬럼을 일일이 정의하는 대신 AS 절을 추가하여 EMP 테이블과 동일한 내용과 구조를 갖는 EMP02 테이블을 생성해 봅시다.

```
CREATE TABLE EMP02  
AS  
SELECT * FROM EMP;
```

〈실습하기〉 원하는 컬럼으로 구성된 복제 테이블 생성하기

기존 테이블에서 원하는 컬럼만 선택적으로 복사해서 생성할 수도 있습니다.

1. 서브 쿼리문의 SELECT 절에 * 대신 원하는 컬럼명을 명시하면 기존 테이블에서 일부의 컬럼만 복사할 수 있습니다.

```
CREATE TABLE EMP03  
AS  
SELECT EMPNO, ENAME FROM EMP;
```


〈탄탄히 다지기〉

2. EMP 테이블을 복사하되 사원번호, 사원이름, 급여 컬럼으로 구성된 테이블을 생성하시오.(테이블의 이름은 EMP04 로 하시오.)



The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe - sqlplus scott/tiger". The prompt is "SQL> SELECT * FROM EMP04;". The output displays two tables of employee data. The first table has 10 rows and the second table has 3 rows. At the bottom, a message states "14 개의 행이 선택되었습니다." (14 rows selected).

```
SQL> SELECT * FROM EMP04;
```

EMPNO	ENAME	SAL
7369	SMITH	800
7499	ALLEN	1600
7521	WARD	1250
7566	JONES	2975
7654	MARTIN	1250
7698	BLAKE	2850
7782	CLARK	2450
7788	SCOTT	3000
7839	KING	5000
7844	TURNER	1500
7876	ADAMS	1100

EMPNO	ENAME	SAL
7900	JAMES	950
7902	FORD	3000
7934	MILLER	1300

14 개의 행이 선택되었습니다.

〈실습하기〉 원하는 행으로 구성된 복제 테이블 생성하기

기존 테이블에서 원하는 행만 선택적으로 복사해서 생성할 수도 있습니다.

1. 서브 쿼리문의 SELECT 문을 구성할 때 WHERE 절을 추가하여 원하는 조건을 제시하면 기존 테이블에서 일부의 행만 복사합니다.

```
CREATE TABLE EMP05  
AS  
SELECT * FROM EMP  
WHERE DEPTNO=10;
```

1.2 테이블의 구조만 복사하기



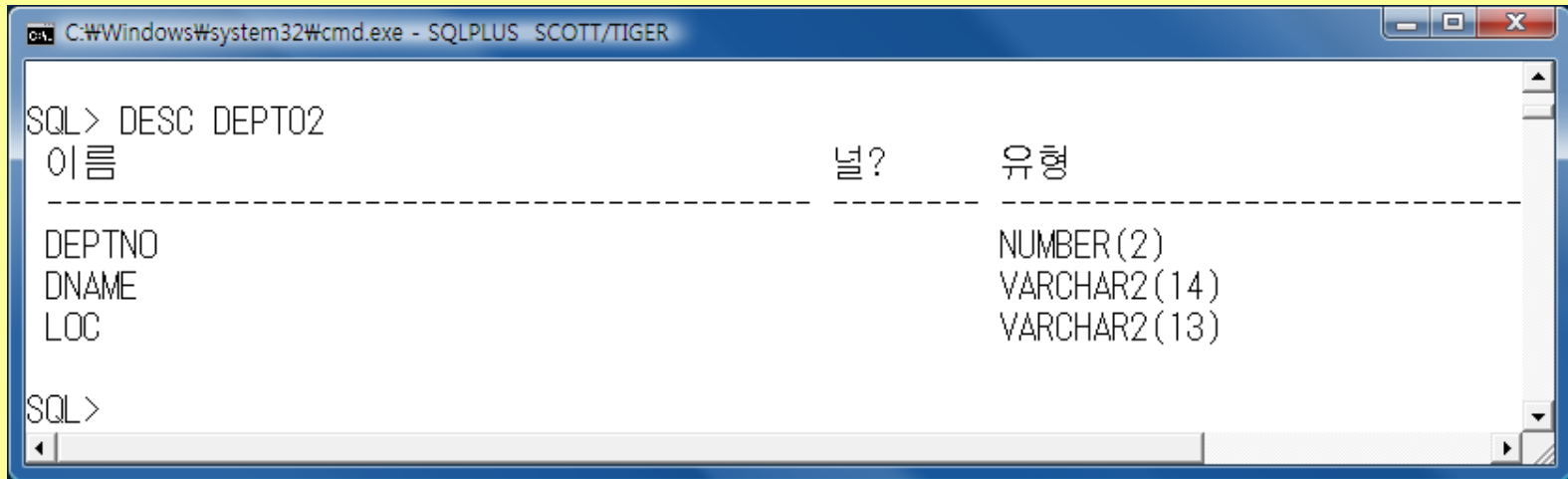
- ❖ 서브 쿼리를 이용하여 테이블을 복사하되 데이터는 복사하지 않고 기존 테이블의 구조만 복사하는 것을 살펴봅시다.
- ❖ 테이블의 구조만 복사하는 것은 별도의 명령이 있는 것이 아닙니다. 이 역시 서브 쿼리를 이용해야 하는데 WHERE 조건 절에 항상 거짓이 되는 조건을 지정하게 되면 테이블에서 얻어질 수 있는 로우가 없게 되므로 빈 테이블이 생성되게 됩니다.

```
CREATE TABLE EMP06  
AS  
SELECT * FROM EMP WHERE 1=0;
```

- ❖ WHERE 1=0; 조건은 항상 거짓입니다. 이를 이용하여 테이블의 데이터는 가져오지 않고 구조만 복사하게 됩니다.

<탄탄히 다지기>

3. DEPT 테이블과 동일한 구조의 빈 테이블 을 생성하시오.(테이블의 이름은 DEPT02 로 하시오.)



```
C:\Windows\system32\cmd.exe - SQLPLUS SCOTT/TIGER

SQL> DESC DEPT02
이름                널?       유형
-----
DEPTNO              NUMBER(2)
DNAME               VARCHAR2(14)
LOC                 VARCHAR2(13)

SQL>
```

02. 테이블 구조 변경하는 ALTER TABLE



- ❖ ALTER TABLE 명령문은 기존 테이블의 구조를 변경하기 위한 DDL 명령문입니다. 테이블에 대한 구조 변경은 컬럼의 추가, 삭제, 컬럼의 타입이나 길이를 변경할 때 사용합니다. 테이블의 구조를 변경하게 되면 기존에 저장되어 있던 데이터에 영향을 주게 됩니다.
- ❖ ALTER TABLE로 칼럼 추가, 수정, 삭제하기 위해서는 다음과 같은 명령어를 사용합니다.
 - ADD COLUMN 절을 사용하여 새로운 칼럼을 추가한다.
 - MODIFY COLUMN 절을 사용하여 기존 칼럼을 수정한다.
 - DROP COLUMN 절을 사용하여 기존 칼럼을 삭제한다.

2.1 새로운 컬럼 추가하기



- ❖ ALTER TABLE ADD 문은 기존 테이블에 새로운 컬럼을 추가합니다.
- ❖ 새로운 컬럼은 테이블 맨 마지막에 추가되므로 자신이 원하는 위치에 만들어 넣을 수 없습니다.
- ❖ 또한 이미 이전에 추가해 놓은 로우가 존재한다면 그 로우에도 컬럼이 추가되지만, 컬럼 값은 NULL 값으로 입력됩니다.

--한 개의 컬럼 추가

```
ALTER TABLE table_name  
ADD column_name data_type;
```

--여러 개의 컬럼 추가

```
ALTER TABLE table_name  
ADD (column_name data_type, ...);
```

〈실습하기〉 EMP01 테이블에 JOB 컬럼 추가하기

EMP01 테이블에 문자 타입의 직급(JOB) 컬럼을 추가해 봅시다.

```
ALTER TABLE EMP01  
ADD(JOB VARCHAR2(9));
```

```
DESC EMP01
```

<탄탄히 다지기>

4. DEPT02 테이블에 문자 타입의 부서장(DMGR) 칼럼을 추가해 봅시다.

명령 프롬프트 - sqlplus scott/tiger

SQL> desc dept02;

Name	Null?	Type
DEPTNO		NUMBER(2)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)
DMGR		VARCHAR2(10)

SQL>

2.2 기존 컬럼 속성 변경하기



- ❖ ALTER TABLE MODIFY 문을 다음과 같은 형식으로 사용하면 테이블에 이미 존재하는 컬럼을 변경할 수 있게 됩니다.

```
-- 한 개의 컬럼 변경
ALTER TABLE table_name
MODIFY column_name data_type expr;
-- 여러 개의 컬럼 변경
ALTER TABLE table_name
MODIFY (column_name data_type expr, ...);
```

- ❖ 컬럼을 변경한다는 것은 컬럼에 대해서 데이터 타입이나 크기, 기본값들을 변경한다는 의미입니다.
- 해당 컬럼에 자료가 없는 경우
컬럼의 데이터 타입과 크기를 변경할 수 있습니다.
- 해당 컬럼에 자료가 있는 경우
컬럼의 데이터 타입을 변경할 수 없습니다. 크기는 늘릴 수 있지만 데이터의 크기보다 작은 크기로 변경할 수 없습니다.

〈실습하기〉 컬럼의 크기 변경하기

1. 직급(JOB) 컬럼을 최대 30글자까지 저장할 수 있게 변경해 보도록 하자.

```
ALTER TABLE EMP01  
MODIFY(JOB VARCHAR2(30));
```

<탄탄히 다지기>

5. DEPT02 테이블의 부서장(DMGR) 칼럼을 숫자 타입으로 변경해 봅시다.

```
SCOTT>DESC DEPT02
```

Name	Null?	Type
DEPTNO		NUMBER(2)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)
DMGR		NUMBER(4)

2.3 기존 컬럼 삭제



- ❖ 테이블에 이미 존재하는 컬럼을 삭제해 봅시다.
- ❖ ALTER TABLE ~ DROP COLUMN 명령어로 컬럼을 삭제할 수 있습니다.

--한 개의 컬럼 삭제

```
ALTER TABLE table_name  
DROP COLUMN column_name;
```

--여러 개의 컬럼 삭제

```
ALTER TABLE table_name  
DROP(column_name, ..) ;
```

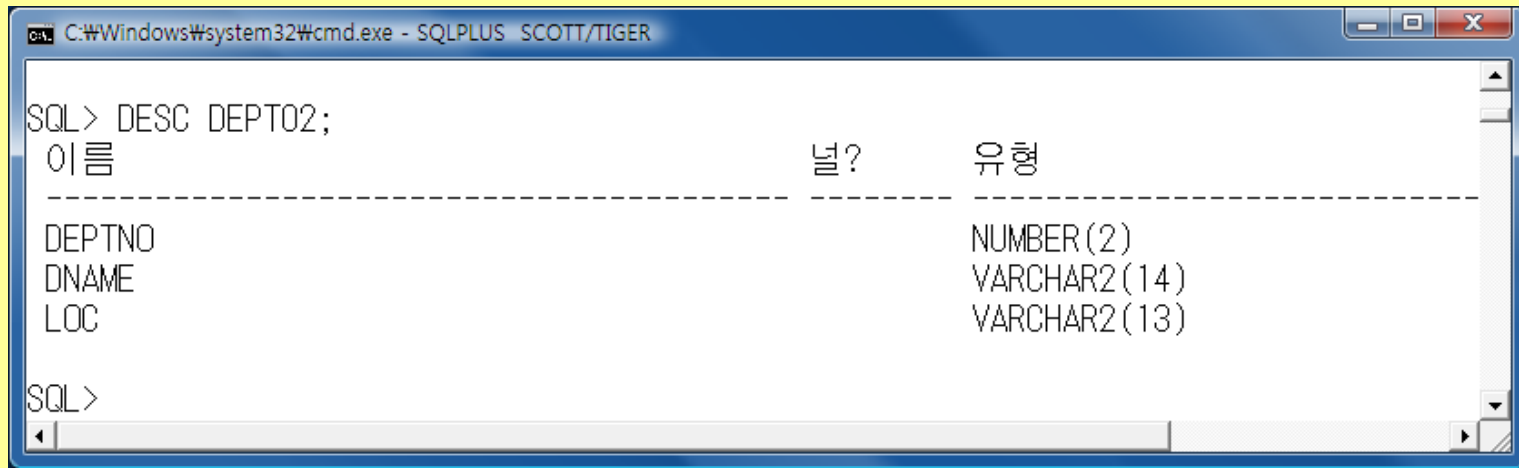
〈실습하기〉 직급 컬럼 삭제하기

1. EMP01 테이블의 직급 컬럼을 삭제해 보도록 합시다.

```
ALTER TABLE EMP01  
DROP COLUMN JOB;
```

<탄탄히 다지기>

6. DEPT02 테이블의 부서장(DMGR) 칼럼을 삭제해 봅시다.



```
C:\Windows\system32\cmd.exe - SQLPLUS SCOTT/TIGER

SQL> DESC DEPT02;
이름                                날?      유형
-----
DEPTNO                             NUMBER(2)
DNAME                              VARCHAR2(14)
LOC                                VARCHAR2(13)

SQL>
```

03. 테이블 구조 삭제하는 DROP TABLE



❖ DROP TABLE문은 기존 테이블을 제거합니다.

```
DROP TABLE table_name [cascade constraints];
```

Drop으로 테이블을 삭제하면 참조 제약조건을 제외한 인덱스, 트리거, 해당 테이블에 연관된 제약조건 자동 삭제됩니다.
cascade constraints 옵션을 사용하지 않고 부모 테이블을 제거하면 오류 발생합니다.

〈실습하기〉 테이블 삭제하기

1. CREATE TABLE을 학습할 때 만들어 놓았던 EMP01 테이블을 삭제해 봅시다.

```
SELECT * FROM EMP01;
```

```
DROP TABLE EMP01;
```

```
SELECT * FROM EMP01;
```


04. 테이블의 모든 로우를 제거하는 TRUNCATE

- ❖ 기존에 사용하던 테이블의 모든 로우를 제거하기 위한 명령어로 TRUNCATE가 제공됩니다.



```
TRUNCATE TABLE table_name
```

〈실습하기〉 테이블의 내용 전체 제거하기

테이블 EMP02 에 저장된 데이터를 확인하였으면 이번에는 테이블의 모든 로우를 제거해 보도록 하겠습니다.

```
TRUNCATE TABLE EMP02;
```

```
SELECT * FROM EMP02;
```

05. 테이블 명을 변경하는 RENAME



- ❖ 기존에 사용하던 테이블의 이름을 변경하기 위한 명령어로 RENAME이 제공됩니다.

```
RENAME old_name TO new_name
```

- ❖ EMP02 테이블의 이름을 TEST 란 이름으로 변경합니다.

```
RENAME EMP02 TO TEST;
```

05. 컬럼 명을 변경하는 RENAME



- ❖ 기존에 사용하던 컬럼 이름을 변경하기 위한 명령어로 RENAME이 제공됩니다.

```
ALTER TABLE table_name  
RENAME column old_name TO new_name
```

- ❖ EMP02 테이블의 컬럼 EMPNO를 EMPNO2란 이름으로 변경합시다.

```
ALTER TABLE EMP02  
RENAME COLUMN EMPNO TO EMPNO2;
```

<탄탄히 다지기>



7. DEPT02 테이블의 컬럼 LOC를 LOC2로 변경해 봅시다.

```
C:\Windows\system32\cmd.exe - SQLPLUS SCOTT/TIGER

SQL> DESC DEPT02;
이름                널?       유형
-----
DEPTNO              NUMBER(2)
DNAME               VARCHAR2(14)
LOC                 VARCHAR2(13)

SQL>
```