

테이블은 실제로 데이터들이 저장되는 곳 이라고 생각하면 쉽게 이해 할 수 있으며, CREATE TABLE 명령어를 이용해서 테이블을 생성 할 수 있다.

테이블이란?

- 테이블은 데이터베이스의 기본적인 데이터 저장 단위 이다.
- 데이터베이스 테이블은 사용자가 접근 가능한 모든 데이터를 보유하며 레코드와 컬럼으로 구성 된다.
- 관계형 데이터베이스가 아닌 예전의 데이터 베이스 용어에서는 파일과 테이블이, 필드와 컬럼이, 그리고 레코드와 행이 동일시 되었다.
- 테이블은 시스템내에서 독립적으로 사용되길 원하는 엔티티를 표현할수 있다. 예를 들면, 회사에서의 고용자나 제품에 대한 주문은 테이블로 표현 가능하다.
- 테이블은 두 엔티티간의 관계를 표현할 수 있다. 즉 테이블은 고용자와 그들의 작업 숙련도 혹은 제품과 주문과의 관계를 표현하는데 사용될 수 있다.
- 테이블내에 있는 외래 키 (Foreign Key)는 두 엔티티 사이의 관계를 표현하는데 사용 된다.
- 컬럼 : 테이블의 각 컬럼은 엔티티의 한 속성을 표현 한다
- 행(ROW, 레코드) : 테이블의 데이터는 행에 저장 된다

※ 테이블 생성시 제한사항과 고려할점

- 테이블 이름과 컬럼은 항상 알파벳 문자로 시작해야 하며 A~Z까지의 문자, 0~9까지의 숫자, 그리고 \$, #, _ (Under Bar)를 사용할 수 있다. (공백 사용 불가능)
- 테이블의 컬럼 이름은 30자를 초과할수 없고, 예약어를 사용할수 없다. (버전없이 되면서 컬럼수 제한 변경 됨)
- 오라클 테이블 한 계정안에서 테이블 이름은 다른 테이블 이름과 달리 유사해야 한다.
- 한 테이블 안에서 컬럼이름은 같을 수 없으며 다른 테이블에서의 컬럼이름과는 같을수 있다.

테이블 생성 문법

```
CREATE TABLE [schema.]table_name
( column datatype
  [, column datatype ...]
)
[TABLESPACE tablespace ]
[PCTFREE integer ]
[PCTUSED integer ]
[INITRANS integer ]
[MAXTRANS integer ]
[STORAGE storage-clause ]
[LOGGING | NOLOGGING ]
[CACHE | NOCACHE ];
```

- **schema** : 테이블의 소유자
- **table_name** : 테이블 이름
- **column** : 컬럼의 이름
- **datatype** : 컬럼의 데이터 유형
- **TABLESPACE** : 테이블이 데이터를 저장 할 테이블스페이스

- PCTFREE : 블록내에 이미 존재하고 있는 Row에 Update가 가능하도록 예약시켜 놓는 블록의 퍼센트 값을 지정 한다.
 - PCTUSED : 테이블 데이터가 저장될 블록의 행 데이터 부분의 크기를 퍼센트지로 지정 한다. PCTFREE에 의해 지정된 크기만큼 Block이 차면 PCTUSED 값보다 작아져야 새로운 행 삽입이 가능하다.
 - INITRANS : 하나의 데이터 블록에 지정될 초기 트랜잭션의 값을 지정 한다.
 - MAXTRANS : 하나의 데이터 블록에 지정될 수 있는 트랜잭션 최대 수를 지정 한다.
 - STORAGE : 익스텐트 스토리지에 대한 값을 지정 한다.
 - LOGGING : 테이블에 대해 이후의 모든 작업이 리두 로그 파일 내에 기록 되도록 지정 한다
 - NOLOGGING : 리두 로그 파일에 테이블의 생성과 특정 유형의 데이터 로드를 기록하지 않도록 지정 한다.
 - PCTFREE, PCTUSED에 대한 자세한 강좌는 오라클 어드민 강좌의 6. Storage Structure => PCTFREE와 PCTUSED를 참고해 주세요
- 오라클 버전없이 되면서 PCTFREE, PCTUSED 설정 및 관리가 많이 변경 되었습니다.

테이블 생성 예제

테이블 생성시 주의사항

- 테이블 이름을 지정하고 각 컬럼들은 괄호 "(" 로 묶어 지정한다.
- 컬럼뒤에 데이터 타입은 꼭 지정되어야 한다.
- 각 컬럼들은 콤마","로 구분되고, 항상 끝은 세미콜론";" 으로 끝난다.
- 한 테이블 안에서 컬럼이름은 같을수 없으며 다른 테이블에서의 컬럼이름과는 같을 수 있다.

```
-- emp2와 dept2테이블을 생성하는 예제이다.
SQL> CREATE TABLE EMP2 (
    EMPNO    NUMBER    CONSTRAINT emp_pk_empno PRIMARY KEY,
--   (컬럼) (데이터타입)   (제약조건)
    ENAME    VARCHAR2(20),
    JOB      VARCHAR2(40),
    MGR      NUMBER,
    HIREDATE  DATE,
    SAL      NUMBER,
    COMM     NUMBER,
    DEPTNO   NUMBER);
테이블이 생성되었습니다.

SQL> CREATE TABLE DEPT2 (
    DEPTNO   NUMBER    CONSTRAINT dept_pk_deptno PRIMARY KEY,
    DNAME    VARCHAR2(40),
    LOC      VARCHAR2(50));
테이블이 생성되었습니다.
```

USER가 소유한 모든 테이블 조회

```
-- USER_TABLES 데이터사전을 조회 하면 유저가 소유한 테이블을 확인 할 수 있다.  
SQL>SELECT table_name FROM USER_TABLES;
```

```
TABLE_NAME  
-----  
BONUS  
CRETABLE  
DEPT  
DUMMY  
EMP  
EMP2  
SALGRADE
```