

## 09강 테이블 제어(생성, 수정, 삭제)

## DDL

- DDL(데이터 정의어) – 테이블의 구조 자체를 생성하거나 변경, 제거를 하기 위한 명령어의 집합체
  - 테이블 생성 : CREATE
  - 테이블 변경 : ALTER
  - 테이블 제거 : DROP

## 테이블 생성 - CREATE

- CREATE는 테이블의 구조를 생성할 때 사용하는 쿼리문이다.
- 테이블을 만들기 위해서 여러가지 조건을 생성해야 한다.
  - 테이블명
  - 테이블의 구성하는 컬럼의 이름과 타입
  - 테이블에 적용한 무결성 제약조건
- 테이블명과 컬럼명을 짓는 조건
  - 문자(A~Z,a~z)로 시작해야 하며 30자 이내로 작성
  - 문자(A~Z,a~z),숫자(0~9),특수문자(underscore, dollar sign, hash)만 사용 가능
  - 대소문자 구별은 없다, 소문자로 저장하려면 별도로 "로 묶어준다
  - 동일 사용자가 소유한 다른 객체의 이름과 중복되지 않아야 한다.

## 테이블 생성 - CREATE

- CREATE는 테이블의 구조를 생성할 때 사용하는 쿼리문이다.
- 기본 사용법
  - CREATE TABLE 테이블명(  
    컬럼명1 타입,  
    컬럼명2 타입,...  
);
- 제약조건은 제약조건을 배운 다음 적용하는 방법을 배우도록 하자.

## 테이블 생성 - CREATE

- 예제 : 다음과 표와 같은 테이블 구조를 생성해보자
  - 테이블명 dept2

컬럼명	타입	크기
dno	number	2
dname	varchar2	14
loc	varchar2	13

```
CREATE TABLE dept2(  
    dno NUMBER(2),  
    dname VARCHAR2(14),  
    loc VARCHAR2(13)  
);
```

## 테이블 생성 - CREATE

- 서브 쿼리를 통해서 새로운 테이블을 생성할 수도 있다.
  - 이 경우 기존의 테이블의 구조를 복사해 올수 있다
  - 다만 NOT NULL을 제외한 제약조건을 복사 해올수 없다.

- CREATE TABLE 테이블명[컬럼 순서...]

AS 서브쿼리

- 예제: department 테이블의 데이터를 가지고 새로운 dept\_second 테이블을 생성해 본다.

```
CREATE TABLE dept_second  
AS SELECT * FROM department;
```

```
SELECT * FROM dept_second;
```

## 테이블 생성 - CREATE

- 서브 쿼리로 기존 테이블의 모든 내용을 가져올 수도 있지만 레코드는 제외하고 테이블 구조만 복사해 올수 있다.

```
CREATE TABLE dept_third
AS
SELECT * FROM department
WHERE 1=0;

SELECT * FROM dept_third;
```

- 또한 서브 쿼리를 생성되는 모든 형태를 테이블로 작성가능하지만 컬럼에 수식이 들어간 형태는 반드시 별칭을 지정해야한다.

```
CREATE TABLE dept20
AS
SELECT eno,ename,salary*12 AS "ani_sal"
FROM employee
WHERE dno=20;
```

## 테이블 변경 - ALTER

- ALTER는 컬럼을 추가하거나, 컬럼의 정보를 변경하거나, 컬럼을 삭제함으로써 테이블의 구조를 변경하기 위해 사용하는 쿼리문이다.
- 기본 사용법
  - 컬럼 생성 : ALTER TABLE 테이블명 ADD(컬럼명 타입(크기))
  - 컬럼 변경 : ALTER TABLE 테이블명 MODIFY 컬럼명 타입(크기)
  - 컬럼 삭제 : ALTER TABLE 테이블명 DROP COLUMN 컬럼명



## 테이블 변경 - 컬럼 추가

- 컬럼 생성 : ALTER TABLE 테이블명 ADD(컬럼명 타입(크기))

```
ALTER TABLE dept20  
ADD(birth date);
```

## 테이블 변경 - 컬럼 변경

- 컬럼 변경 : ALTER TABLE 테이블명 MODIFY 컬럼명 타입(크기)

```
ALTER TABLE dept20  
MODIFY ename varchar2(30);
```

- 단순히 컬럼 이름만 변경할 시

```
- ALTER TABLE dept20  
  RENAME COLUMN SCHOOL TO Education
```

## 테이블 변경 - 컬럼 삭제

- 컬럼 삭제 : ALTER TABLE 테이블명 DROP COLUMN 컬럼명

```
ALTER TABLE dept20  
DROP COLUMN AGE  
.
```

## 테이블 변경 - UNUSED

- 컬럼 삭제 : 실제 테이블에서 컬럼을 삭제하는 것이 아닌 사용하지 않는다고 명령을 내릴 수 있다
- 실제 삭제하는 것보다 빠르게 삭제한 것과 같은 효과를 낼 수 있다
  - 삭제의 효율성을 위한 것으로 복원이 불가능하다.

```
ALTER TABLE dept20  
SET UNUSED (BIRTH);
```

- 숨겨진 UNUSED를 테이블로부터 완전히 삭제하는 명령은 다음과 같다

```
ALTER TABLE dept20  
DROP UNUSED COLUMNS
```

## 테이블이름 변경 - RENAME

- RENAME는 테이블의 이름을 변경할 때 사용하는 쿼리문 이다.
- 기본 사용법
  - RENAME 구 테이블명 TO 신규 테이블명
- 예제 : dept20테이블을 emp20테이블로 변경해 보자

```
RENAME dept20 TO emp20;
```

## 테이블 삭제 - DROP

- DROP는 테이블의 구조를 제거할 때 사용하는 쿼리문이다.

- 기본 사용법

- DROP TABLE 테이블 명

- 예제 : 새로 생성한 emp20 테이블을 삭제해 본다

```
DROP TABLE emp20;
```

- **[경고]** 테이블 삭제하면 내부에 포함된 모든 레코드가 삭제되어서 복원이 불가능하므로 테이블 삭제는 신중해야 하며 가급적 하지 않는 것을 권장한다.

## 테이블속 데이터 삭제 - TRUNCATE

- TRUNCATE는 테이블의 구조는 남겨두고 모든 레코드를 삭제할 때 사용하는 쿼리문이다.
- 기본 사용법
  - TRUNCATE TABLE 테이블 명;
  - 예제 : 새로 생성한 dept\_second 테이블을 삭제해 본다

```
TRUNCATE TABLE dept_second;
```

## 데이터 사전

- 데이터 사전이란 데이터베이스를 효율적으로 관리하기 위해 다양한 정보를 저장하는 시스템 테이블의 집합체

접두어	의미
USER_	자신의 계정이 소유한 객체 등에 관한 정보 조회
ALL_	자신 계정 소유 또는 권한을 부여 받는 객체 등에 관한 정보 조회
DBA_	데이터 베이스 관리자만 접근 가능한 객체 등의 정보 조회



## 데이터 사전 - USER\_

- 접두어로 USER가 붙은 데이터 사전은 사용자와 가장 관계가 깊은 뷰이다.
- 자신이 생성한 테이블, 인덱스, 뷰, 동의어 등 객체나 해당 사용자에게 부여된 권한 정보를 제공

```
SELECT TABLE_NAME FROM USER_TABLES;
```

- USER\_SEQUENCES : 사용자가 소유한 시퀀스 정보
- USER\_INDEXES : 사용자가 소유한 인덱스정보
- USER\_VIEWS : 사용자가 소유한 뷰 정보

## 데이터 사전 - ALL\_

- 접두어로 ALL가 붙은 데이터 사전은 전체 사용자와 관련된 뷰로서 사용자가 접근 가능한 모든 객체에 대한 정보 조회 가능하다.
- 조회중인 객체의 소유권을 확인할 수 있다.

```
SELECT OWNER, TABLE_NAME FROM ALL_TABLES;
```

## 데이터 사전 - DBA\_

- 접두어로 DBA가 붙은 데이터 사전은 시스템 관리와 관련된 뷰이다.
- 시스템 관리자만이 접근 가능하다.
  - - 시스템 계정으로 접속해서 테스트 해보자

```
SELECT OWNER, TABLE_NAME FROM DBA_TABLES;
```

- **[경고]** 중요한 정보가 있으므로 함부로 삭제하지 않도록 한다.