

## 1. 테이블 생성(DDL문)

Data Definition Language으로 테이블(table), 뷰(view), 인덱스(index) 등과 같은 데이터베이스주요를 생성하거나 수정하거나 삭제하는 SQL문이다. CREATE TABLE문으로 새로운 테이블을 생성한다.

```
CREATE TABLE 테이블명 (  
    컬럼명1    데이터타입    [NULL | NOT NULL],  
    .....  
    컬럼명N    데이터타입    [NULL | NOT NULL],  
    [CONSTRAINT 제약조건명1 PRIMARY KEY (컬럼명1, 컬럼명2, .....)],  
    [CONSTRAINT 제약조건명2 FOREIGN KEY (컬럼명1, 컬럼명2, .....)  
        REFERENCES 참조테이블명(컬럼명1, 컬럼명2, .....)] );
```

### ※ 기술 방법

- 테이블명 : 생성할 테이블 이름을 기술한다.
- 컬럼명1, 컬럼명2, ..... : 테이블의 컬럼 이름을 기술한다.
- 데이터타입 : 컬럼의 데이터타입과 크기를 지정한다.
- NULL : 선택 컬럼을 지정하는 제약조건으로 생략 가능하다.
- NOT NULL : 필수 컬럼을 지정하는 제약조건이다.

### ☞ 테이블명 정의

- 사용자 이름에 속한 각 테이블명은 유일해야 한다.
- 테이블명은 영문자로 시작해야 한다.
- 테이블명은 영문자, 1부터 9까지 숫자, 특수문자 \$, #, \_(under bar)를 사용할 수 있다.
- 테이블명은 30자를 초과할 수 없다.
- SQL 예약어는 사용할 수 없다.
- 테이블명은 서술적이어야 한다.
- 테이블명에 대소문자로 구분할 수 있다. 단, 소문자인 경우 큰따옴표("")를 사용한다.
- 테이블명으로 한글을 사용할 수도 있다.

### ☞ 컬럼명 정의 : 테이블에 생성되는 컬럼들은 괄호 안에 기술

- 하나의 테이블에서 컬럼들은 괄호 안에 기술한다.
- 컬럼명은 영문자로 시작해야 한다.
- 컬럼명은 영문자, 1부터 9까지 숫자, 특수문자 \$, #, \_(under bar)를 사용할 수 있다.
- 컬럼명은 30자를 초과할 수 없다.
- SQL 예약어는 사용할 수 없다.
- 컬럼명도 서술적이어야 한다.

- 컬럼명으로 한글을 사용할 수도 있다.

```

Oracle SQL*Plus
File Edit Search Options Help
SQL> CREATE TABLE GRN_MASTER
2  (
3  GRN_NUM          NUMBER,
4  GRN_DATE         DATE,
5  SUPPLIER_ID      NUMBER,
6  GATE_PASS_ID     NUMBER,
7  STORE_ID         NUMBER,
8  APPROVED_ID      NUMBER,
9  CONSTRAINTS PK_GRN_NUM PRIMARY KEY (GRN_NUM)
10 );

Table created.

```

데이터 무결성을 위하여 테이블의 각 컬럼에 기본 키, 외부 키, 고유 키, Null/Not Null, 체크 제약 조건을 지정하면 무효한 데이터의 입력을 방지할 수 있다.

### 1.1. 컬럼에 Null 혹은 Not Null 제약조건 지정

테이블의 각각의 컬럼에 대하여 Null이나 Not Null을 지정한다.

- Null 제약조건이란 데이터 입력시 널(null)값을 허용하는 선택 컬럼이며, 생략할 수 있다.
- Not Null 제약조건은 널(null)값을 허용하지 않는 필수 컬럼이다.

널(null)값이란 아직 입력되지 않은 값 또는 아직 알려지지 않은 값을 나타내며, 테이블의 컬럼에 값이 비어 있는 것을 말한다.

### 예제) Department 테이블 만들어보기

#### ☞ 테이블 구조 출력

사용자가 생성한 테이블의 구조를 출력할 수 있다. Describe 명령은 테이블의 구조에 대하여 컬럼 명(이름), Null/Not Null(Null?), 데이터타입과 크기를 출력한다.

**SQL> Describe 테이블명;**

### 1.2. 사용자 소유의 객체명 조회

사용자 소유에 대한 테이블명, 뷰명, 인덱스명 등의 객체명을 중복 사용할 수 없다. 객체를 생성하기 전에 객체명으로 사용이 가능한지 확인이 가능하다.

#### ☞ 테이블명 조회

사용자가 생성한 테이블명들을 조회할 수 있다. SELECT \* FROM TAB; 또는 SELECT \* FROM USER\_CATALOG; 명령으로 로그인한 오라클이름[U], 소유의 생성된 모든 객체에 대하여 객체명

(TNAME), 객체타입(TABTYPE), 클러스터번호(CLUSTERID)를 출력한다.

```
SQL> SELECT * FROM TAB;
```

```
SELECT * FROM USER_CATALOG;
```

#### ☞ 중복된 객체명 사용시 오류 메시지

테이블 등의 객체 생성 시 중복된 객체명을 사용하면 "ORA-00955: 기존의 객체가 이름을 사용하고 있습니다"라는 메시지가 출력되고 객체는 생성되지 않는다. 동일한 이름으로 재생성할 경우에는 객체를 삭제한 후에 생성해야 한다.

### 1.3. 테이블에 기본 키 제약조건 지정

기본키란 테이블에 각 행을 유일하게 식별하기 위하여 사용되는 칼럼 또는 칼럼의 조합을 말한다.

- 기본 키 칼럼에 중복된 데이터를 허용하지 않는다.
- 기본 키 칼럼에 널(Null) 값을 허용하지 않는다.
- 기본 키 칼럼은 널이 될 수 없다.
- 기본 키 칼럼은 자동으로 Not Null이 된다.
- 기본 키 칼럼의 데이터타입은 LONG이나 LONG RAW가 될 수 없다.
- 기본 키를 구성하는 칼럼의 최대 수는 16개이다.

CONSTRAINT 제약조건명 PRIMARY KEY (칼럼명1, 칼럼명2, ....)
---

#### ※ 기술 방법

- CONSTRAINT : 무결성 규칙의 제약조건을 정의하는 예이다.
- 제약조건명 : 기본 키에 대한 제약조건명을 기술한다.
- PRIMARY KEY : 기본 키를 정의하는 예이다.
- (칼럼명1, 칼럼명2, ...) : 기본 키의 칼럼명을 기술한다.

#### 예제) Department 삭제 후 다시 생성해보기

#### 1.4. 테이블에 외부 키 제약조건 지정

테이블과 테이블 간의 관계(relationship)를 정의할 때 사용되며, 관계를 정의하는 생성할 테이블의 하나 이상의 칼럼이 외부 키가 되고, 이 외부 키는 참조할 테이블의 기본 키와 관계가 정의된다. 이것은 외부 키 칼럼의 입력 값이 참조할 테이블의 기본 키 중의 하나임을 확인하는 제약조건이다.

##### ☞ 외부 키와 기본 키와의 관계

- 생성할 테이블의 관계를 정의하는 칼럼이 외부 키가 된다.
- 참조하는 테이블의 기본 키와 관계가 정의된다.
- 외부 키 칼럼의 수와 참조하는 테이블의 기본 키 칼럼의 수가 동일해야 한다.
- 외부 키 칼럼의 데이터타입과 참조하는 테이블의 기본 키 칼럼의 데이터 타입이 동일해야 한다.
- 외부 키 칼럼의 데이터타입을 생략하면, 참조하는 테이블의 기본 키 칼럼의 데이터타입을 참조한다.
- 테이블 생성 시 참조할 테이블의 기본 키 칼럼은 생략할 수 있다.

CONSTRAINT 제약조건명 FOREIGN KEY (칼럼명1, 칼럼명2, .....)

REFERENCE 참조테이블명 (칼럼명P11, 칼럼명P12, ...).

##### ※ 기술 방법

- CONSTRAINT : 무결성 규칙의 제약조건을 정의하는 예약어
- 제약조건명 : 외부 키에 대한 제약조건명을 기술한다.
- FOREIGN KEY : 외부 키를 정의하는 예약어
- (칼럼명1, ...) : 외부 키의 칼럼명을 기술한다.
- REFERENCES : 참조할 테이블명을 지정하기 위한 예약어
- 참조테이블명 : 참조할 자식 테이블명을 기술한다.
- (칼럼명P11, ...) : 참조되는 테이블의 기본 키 칼럼명을 기술한다.(생략가능)

#### 예제) Student 테이블, Professor 테이블 예제 만들어 보기

##### ☞ 제약조건명을 사용해야 하는 이유?

제약조건을 선언할 때, 제약조건명을 명시적으로 기술하는 것이 좋은 방법이다. 이유는 외부 키, 고유 키, 체크 제약조건은 제약조건명으로 삭제한다. 만약, 제약조건명을 생략하면, 오라클 데이터

베이스가 자동으로 "STUE155.SYS\_C0026241"과 같이 제약조건명을 부여한다. 자동으로 부여하는 제약조건명은 사용자 사용하기에 불편하다.

### 1.5. 테이블의 컬럼에 고유 키 제약조건 지정

고유 키(Unique key) 제약조건이란 컬럼에 저장되는 데이터가 유일한 값을 가져야 한다는 것이다. 고유 키 제약조건이 선언된 컬럼은 중복되는 동일한 데이터가 입력되는 것을 방지한다. 고유 키 제약조건은 기본 키를 대신하여 사용하는 것은 아니다. 고유 키 제약조건의 지정은 컬럼명에 'unique'를 명시적으로 기술한다.

컬럼명	데이터타입	[NULL   NOT NULL]	UNIQUE,
-----	-------	-------------------	---------

예제) Professor 테이블로 테스트...

#### ☞ 고유 키 제약조건과 기본 키 제약조건의 차이점

- 기본 키는 테이블에 대하여 단지 한번만 지정할 수 있다.
- 고유 키 제약조건은 각 컬럼에 대하여 여러 번 지정할 수 있다.
- 기본 키는 자동으로 Not Null 컬럼으로 지정되나, 고유 키 제약조건은 Not Null로 지정해야 Not Null이 된다.

### 1.6. 체크 제약조건으로 테이블의 컬럼에 값을 제한

각 컬럼에 대하여 체크 제약조건으로 값을 제한할 수 있다. 체크 제약조건으로 값을 제한할 수 있는 컬럼은

- 일정한 범위의 값이 입력되는 컬럼
- 특정 값들 중의 하나인 경우
- 범위가 일정하지 않으나, 컬럼 값이 입력되거나 수정될 때 계산되는 경우이다.

컬럼명	데이터타입	[NULL   NOT NULL]
[ CONSTRAINT 제약조건명 ] [ CHECK (조건) ],		

예제) Professor 테이블로 테스트 해보기

## 1.7. 칼럼에 기본 값 지정

테이블의 각 칼럼에 대한 기본(default)값을 지정할 수 있다. 기본 값을 지정한 칼럼에 데이터가 입력되지 않으면, 자동으로 기본 값이 대체되어 입력된다. 기본값을 지정하는 칼럼은 Not Null 제약조건을 지정하지 않으며, 칼럼명의 데이터 타입과 기본 값의 데이터타입이 반드시 동일하게 지정해야 한다.

칼럼명	데이터타입	[ DEFAULT 값]
-----	-------	--------------

예제) Courser 테이블로 테스트 해보기

## 2. 테이블 수정

ALTER TABLE문을 이용해 테이블의 구조만 변경하고, 데이터의 내용은 변경되지 않도록 테이블을 수정할 수 있다.

### 🔑 테이블 변경 사항

- 기존 테이블에 새로운 칼럼 추가
- 기존 테이블에 칼럼의 크기를 늘리거나 줄이기
- 기존 칼럼에 Null을 Not Null, Not Null을 Null로 제약조건 변경
- 기존 칼럼 삭제하기
- 기존 칼럼에 기본 값 지정
- 기존 테이블에 기본 키와 외부 키 추가 또는 삭제
- 기존 칼럼에 다른 제약조건 지정 또는 삭제

### 2.1. ALTER TABLE의 문법

#### --- 문법1) 기본 키와 외부 키의 제약조건 추가 작업 시

ALTER TABLE 테이블명

ADD (Column-specification | constraint-specification);

#### --- 문법2) 기존 컬럼을 변경할 때 사용. 칼럼 크기의 변경, 제약조건의 NULL, Not Null, 고유 키, 체크와 기본 값을 변경할 때 작업 시

ALTER TABLE 테이블명 Modify (column-specification | constraint-specification);

--- 문법3) 테이블에 선언된 기본 키 삭제 작업 시

```
ALTER TABLE 테이블명 DROP PRIMARY KEY;
```

--- 문법4) 테이블이나 칼럼에 선언된 제약조건을 제약조건명으로 삭제 작업 시

```
ALTER TABLE 테이블명 DROP CONSTRAINT 제약조건명;
```

--- 문법5) 테이블의 컬럼 삭제 작업 시

```
ALTER TABLE 테이블명 DROP COLUMN 컬럼명;
```

--- 문법6) 테이블명을 다른 테이블명으로 변경 작업 시

```
ALTER TABLE 테이블명 RENAME TO 변경테이블명;
```

※ 기술 방법

• column-specification 구문

칼럼명 데이터타입(크기) [ NULL | NOT NULL | UNIQUE | CHECK ]

• constraint-specification 구문

① CONSTRAINT 제약조건명 PRIMARY KEY (칼럼명1, 칼럼명2,...)

② CONSTRAINT 제약조건명 FOREIGN KEY (칼럼명1, 칼럼명2,...)

REFERENCE 참조테이블명 (칼럼명1, 칼럼명2,...)

## 2.2. 새로운 칼럼 추가

위의 정리된 문법 중 1번 문법을 이용하여 테이블에 칼럼을 추가할 수 있다.

예) Professor 테이블로 컬럼 추가 작업 TEST

## 2.3. 칼럼 크기의 줄이기와 늘이기

### ☞ 칼럼 크기 줄이기

칼럼의 크기를 줄일 경우에는 데이터가 저장된 칼럼은 불가능하고 다음과 같은 조건일 때에만 크기를 줄일 수 있다.

- 테이블의 값이 비어 있을 때(즉 0행일 경우)
- 테이블의 값이 저장되어 있을 때는 널 값이 저장된 칼럼일 때

예) Department로 테스트

#### ☞ 칼럼 크기 늘이기

칼럼의 크기를 늘릴 때는 조건 없이 늘일 수 있다.

예) Department로 테스트

## 2.4. NULL을 NOT NULL로, NOT NULL을 NULL로 변경

선택(NULL) 칼럼을 필수(NOT NULL) 칼럼을 변경하거나, 필수(NOT NULL) 칼럼을 선택(NULL) 칼럼으로 변경할 수 있다.

#### ☞ NULL을 NOT NULL로 변경하기

NULL 제약조건이 지정된 칼럼을 NOT NULL 제약조건의 칼럼으로 변경할 때는 테이블이 비어 있거나, 칼럼의 모든 행이 저장된 칼럼에 대해서만 가능하며, 다음과 같은 조건을 만족해야 한다.

- 테이블의 값이 비어 있을 때 (즉 0행일 경우)
- 테이블의 값이 저장되어 있을 때는, 필수 칼럼으로 변경하고자 하는 칼럼이 널 값을 포함하고 있지 않은 경우

#### ☞ NOT NULL을 NULL로 변경하기

필수(Not Null) 칼럼에서 선택(Null) 칼럼으로 변경할 때는, 언제나 변경할 수 있다. 그러나 반드시 NULL을 명시적으로 기술해야 한다.

예) Department로 테스트

## 2.5. 기본 키의 추가와 삭제

테이블의 행을 유일하게 식별하는 기본 키에 대하여 삭제하거나, 추가할 수 있다.

#### ☞ 기본 키 추가하기

기본 키를 추가하기 위해서는 테이블이 비어 있거나, 기본 키로 추가할 칼럼에 널값이나 중복된 데이터가 존재해서는 불가능하다.

- 테이블의 값이 비어 있을 때(즉 0행일 경우)



- 테이블의 칼럼에 널 값을 포함하지 않고, 중복된 데이터를 포함하고 있지 않을 때

#### ☞ 기본 키 삭제하기

테이블에 정의된 기본 키를 삭제할 수 있다. 그러나 외부 키로 관계가 정의되어 있지 않은 테이블일 경우에만 가능하다.

- 외부 키로 관계가 정의되어 있지 않은 테이블의 기본 키

#### ☞ CASCADE 절

기본 키를 삭제하는 ALTER TABLE문에 CASCADE절을 사용하면, 외부 키로 관계가 정의된 테이블의 기본 키를 강제로 삭제할 수 있다.

예) SG\_Scores Table 생성 후 테스트, Student 테이블로 테스트, Department 테이블로 테스트

## 2.6 외부 키의 추가와 삭제

#### ☞ 외부 키 추가하기

외부 키의 추가는 참조하는 테이블의 칼럼 값이 만족해야 한다.

- 테이블의 값이 비어 있을 때 (즉 0행일 경우)
- 테이블의 값이 저장되어 있을 때는 널 값이 저장된 칼럼일 때

#### ☞ 외부 키 삭제하기

외부 키의 삭제는 조건 없이 삭제할 수 있다.

예) Course 테이블의 Professor\_ID 칼럼을 Professor 테이블을 참조할 수 있게 테스트,  
Professor 테이블로 테스트

## 2.7 기존 칼럼 삭제하기

테이블의 칼럼을 삭제할 수 있다.

예) Professor 테이블로 칼럼 삭제 테스트

## 3. 테이블의 삭제

테이블을 삭제하고자 할 때는 DROP TABLE문을 사용한다. 데이터가 저장된 테이블이 삭제될 수도

있기 때문에 삭제 전에 반드시 확인하는 습관이 필요하다. 삭제된 테이블은 복수할 수 없다.

### 3.1. DROP TABLE문

DROP TABLE문으로 테이블을 삭제할 수 있다. 그러나 외부 키로 관계가 정의된 참조 테이블은 삭제되지 않는다. 이 경우에는 외부 키를 삭제하거나, 외부 키가 정의된 테이블을 먼저 삭제하고, 참조하는 테이블을 삭제해야 한다.

DROP TABLE 테이블명;

예) Student 테이블로 테스트

### 3.2. 테이블 삭제가 불가능한 경우

DROP TABLE문으로 테이블을 삭제할 수 없는 경우가 있다. 삭제할 테이블이 다른 테이블의 외부 키에 의해서 참조되고 있을 때 테이블은 삭제되지 않는다.

- 다른 테이블의 외부 키에 의해서 참조되고 있을 때 삭제 불가



[그림 II-1-5] 테이블의 정규화

예) Courese 테이블로 테스트