

## 10장. 데이터 무결성과 제약 조건

이번 장에서는 필수 입력 칼럼을 지정하는 방법 칼럼이 유일한 값만을 갖도록 하는 방법, 테이블의 모든 데이터가 항상 구분 가능하도록 기본 키를 설정하는 방법, 데이터의 참조가 항상 가능하도록 하는 외래키 설정 방법을 학습합니다.

## 학습 내용

- ❖ 제약 조건 이란?
- ❖ 제약 조건 변경하기

## 학습목표

- ❖ 제약 조건을 말하고 그 종류와 각각의 역할을 설명할 수 있습니다.
- ❖ 제약 조건을 생성하고 유지 및 관리 할 수 있습니다.

## 1. 제약 조건

- ❖ 제약 조건(Data Integrity Rule)이란 테이블에 유효하지 않은 (부적절한) 데이터가 입력되는 것을 방지하기 위해서 테이블 생성 시 각 칼럼에 대해 정의하는 규칙을 말합니다.
- ❖ oracle의 제약 조건 종류

| 구 분         | 설 명  |
|-------------|--|
| NOT NULL    | 칼럼에 NULL 값을 포함하지 못하도록 지정한다.  |
| UNIQUE      | 테이블의 모든 로우에 대해서 유일한 값을 갖도록 한다.   |
| PRIMARY KEY | 테이블의 각 행을 식별하기 위한 것으로 NULL과 중복된 값을 모두 허용하지 않는다. 즉, NOT NULL 조건과 UNIQUE 조건을 결합한 형태이다. |
| FOREIGN KEY | 참조되는 테이블에 칼럼 값이 항상 존재해야 한다.  |
| CHECK       | 저장 가능한 데이터 값의 범위나 조건을 지정하여 설정한 값만을 허용한다.   |

## 1. 제약 조건

- ❖ 제약 조건은 일반적으로 테이블이 생성 될 때 생성되며, 이 후 ALTER TABLE을 이용하여 추가 및 삭제가 가능합니다.

|     |  |
|-----|--|
| 형 식 | <pre>CREATE TABLE [schema.] table<br/>    (column datatype [DEFAULT expression]<br/>    [column_constraint], [table_constraint] [, ...]) ;</pre> |
|-----|--|

- ❖ 제약 조건은 칼럼 레벨과 테이블 레벨로 정의할 수 있습니다.
- ❖ 칼럼 레벨에서 정의하는 경우 하나의 칼럼에 대해서 제약 조건을 지정하며 모든 제약 조건을 정의할 수 있습니다.
- ❖ 테이블 레벨에서 제약 조건을 정의하는 경우 칼럼 정의와는 별도로 정의하여 NOT NULL을 제외한 모든 제약 조건을 정의할 수 있습니다.

## 1-1. NOT NULL 제약 조건

- ❖ NOT NULL 제약 조건을 사용하면 해당 칼럼이 NULL 값을 가질 수 없게 됩니다. 반드시 값이 있어야 하는 칼럼에 지정합니다.
- ❖ NOT NULL 제약 조건은 칼럼 레벨로만 정의할 수 있습니다. 테이블에 NOT NULL 제약 조건을 지정하려면 데이터 타입을 기술한 후 연이어 NOT NULL 기술하면 됩니다.

예 제

```
CREATE TABLE customer(  
    id          VARCHAR2(20) NOT NULL,  
    pwd         VARCHAR2(20) NOT NULL,  
    name        VARCHAR2(20) NOT NULL,  
    phone       VARCHAR2(30) ,  
    address     VARCHAR2(100)) ;
```

- ❖ 테이블에 NULL 데이터를 추가하여 봅시다.

예 제

```
INSERT INTO customer  
VALUES (null, null, null, '010-111-1111' , 'seoul' ) ;
```

## 1-2. 유일한 값만 허용하는 UNIQUE 제약 조건

- ❖ UNIQUE 제약 조건은 특정 칼럼에 모든 값이 고유하게 유지되도록 하는 고유키를 생성합니다. 즉 칼럼에 중복된 값을 가질 수 없도록 합니다.
- ❖ oracle은 고유키에 의해 인덱스를 암시적으로 생성합니다.

예 제

```
CREATE TABLE customer(  
    id          VARCHAR2(20) unique,  
    pwd         VARCHAR2(20) NOT NULL,  
    name        VARCHAR2(20) NOT NULL,  
    phone       VARCHAR2(30) ,  
    address     VARCHAR2(100)) ;
```

## 1-2. 유일한 값만 허용하는 UNIQUE 제약 조건

[예제] 제약 조건의 이름을 지정하여 봅시다.

예 제

```
CREATE TABLE customer(  
  id    VARCHAR2(20) constraint customer_id_uk unique,  
  pwd   VARCHAR2(20) constraint customer_pwd_nn NOT NULL,  
  name  VARCHAR2(30) constraint customer_name_nn NOT NULL,  
  phone VARCHAR2(30) ,  
  address VARCHAR2(100)) ;
```



## 1-2. 유일한 값만 허용하는 UNIQUE 제약 조건

- ❖ 데이터 제약 조건을 보기 위해 USER\_CONSTRAINTS 데이터 사전을 사용합니다.
- ❖ USER\_CONS\_COLUMNS 데이터 사전은 USER\_CONSTRAINTS 데이터 사전으로 알 수 없는 제약 조건이 설정된 칼럼 이름을 알려줍니다.

[예제] 제약 조건을 확인해봅시다 .

예 제

```
SELECT table_name, constraint_name  
FROM   user_constraints  
WHERE  table_name IN ( 'CUSTOMER' ) ;
```

## 1-3. 데이터 구분을 위한 PRIMARY KEY 제약 조건

- ❖ PRIMARY KEY 제약 조건은 테이블에 기본 키를 생성합니다.
- ❖ 제약 조건이 지정된 칼럼은 중복된 데이터를 저장하지 못하는 고유성이 부여됩니다.
- ❖ 기본 키에 속하는 칼럼은 NULL 값을 가질 수 없습니다.
- ❖ 고유 키 제약 조건과 NOT NULL 제약 조건을 결합한 개념으로 테이블의 모든 로우를 구별하기 위한 식별자를 정의하게 됩니다.

예 제

```
CREATE TABLE customer(  
  id      VARCHAR(20),  
  pwd     VARCHAR(20) constraint customer_pwd_nn not null,  
  name    VARCHAR(20) constraint customer_name_nn not null,  
  phone   VARCHAR(30),  
  Address VARCHAR(100),  
  CONSTRAINT customer_id_pk primary key(id) ;
```

## 1-4. 참조 무결성을 위한 FOREIGN KEY 제약 조건

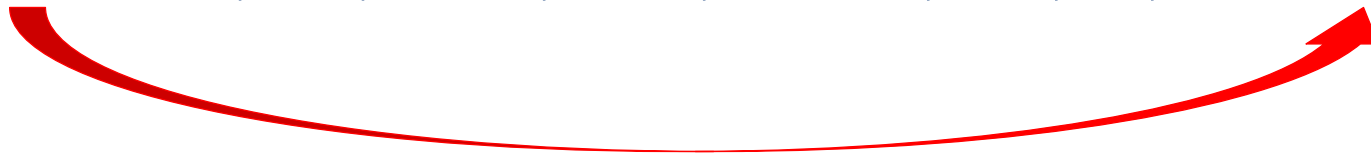
- ❖ FOREIGN KEY 테이블에 외래 키 제약 조건을 생성합니다.
- ❖ 참조 무결성은 테이블과 테이블 사이의 주종 관계 설정을 위한 제약 조건입니다.

부모 테이블

|                    |  |  |
|--------------------|--|--|
| <b>Primary Key</b> |  |  |
|                    |  |  |
|                    |  |  |

자식테이블

|  |  |  |                    |
|--|--|--|--------------------|
|  |  |  | <b>Foreign Key</b> |
|  |  |  |                    |
|  |  |  |                    |



## 1-4. 참조 무결성을 위한 FOREIGN KEY 제약 조건

[예제] FOREIGN KEY 제약 조건을 지정하여 봅시다

예 제

```
CREATE TABLE emp_second(  
  eno      NUMBER(4) constraint emp_second_eno_pk primary key,  
  ename    VARCHAR2(10),  
  Job      VARCHAR2(9),  
  dno      VARCHAR2(9),  
  CONSTRAINT emp_second_dno_라 references department );
```

## 1-5. 값의 범위를 설정하는 CHECK 제약 조건

- ❖ CHECK 제약 조건은 컬럼에서 허용 가능한 데이터의 범위나 조건을 정의합니다.
- ❖ 입력되는 값을 체크하여 설정된 값 이외의 값이 들어오면 오류 메시지와 함께 명령이 수행되지 못하게 된다.
- ❖ CHECK 제약 조건의 수에는 제한이 없지만, CURRVAL, NEXTVAL, ROWNUM 같은 의사 컬럼이나 SYSDATE, USER와 같은 함수에는 사용할 수 없습니다.

## 1-5. 값의 범위를 설정하는 CHECK 제약 조건

[예제] CHECK 제약 조건을 지정하여 보시다

예 제

```
CREATE TABLE emp_second(  
  eno      NUMBER(4) constraint emp_second_eno_pk primary key,  
  ename    VARCHAR2(10),  
  salary   constraint emp_second_salary_min CHECK (salary > 0 );
```

## 1-6. DEFAULT 제약 조건

- ❖ 어떤 값도 입력하지 않았을 때 DEFAULT 제약의 값이 입력됩니다.

**[예제] DEFAULT 제약 조건을 지정하여 봅시다.**

**예 제**

```
CREATE TABLE emp_second(  
  eno      NUMBER(4) constraint emp_second_eno_pk primary key,  
  ename    VARCHAR2(10),  
  salary   NUMBER(7,2) default 1000 );
```

## 2. 제약 조건 추가

- ❖ ALTER ... ADD CONSTRAINT 문을 사용하면 기존 테이블에 대해서도 제약 조건을 추가할 수 있습니다.

|    |  |
|----|--|
| 형식 | <pre>ALTER TABLE table_name ADD [CONSTRAINT constraint_name] constraint_type (column_name)</pre> |
|----|--|

- ❖ NULL 무결성 제약 조건은 ALTER ... ADD 명령문으로 추가하지 못합니다.



## 2-2. 제약 조건 제거

- ❖ **ALTER TABLE ... DROP CONSTRAINT** 명령문을 사용하면 기존 테이블에 대해서도 제약 조건을 제거 할 수 있습니다.

|    |  |
|----|--|
| 형식 | <b>ALTER TABLE table_name<br/>DROP PRIMARY KEY   UNIQUE (column)   CONSTRAINT<br/>constraint_name[CASCADE]</b> |
|----|--|

**[예제] 제약 조건을 제거하여 봅시다.**

|    |  |
|----|--|
| 예제 | <b>ALTER TABLE emp_copy<br/>DROP PRIMARY KEY ;</b> |
|----|--|

### 3. 제약 조건 활성화 및 비활성화

- ❖ ALTER TABLE 문에 DISABLE 절을 사용하면 제약 조건을 삭제하지 않고 일시적으로 비활성화 할 수 있습니다.

형 식

```
ALTER TABLE table_name  
DISABLE CONSTRAINT constraint_name [CASCADE]
```

#### [예제] 제약 조건 비활성화

예 제

```
ALTER TABLE emp_copy  
DISABLE CONSTRAINT emp_copy_dno_fk ;
```

### 3. 제약 조건 활성화 및 비활성화

- ❖ ALTER TABLE 문에 ENABLE 절을 사용하면 제약 조건을 일시적으로 활성화 할 수 있습니다.

형 식

```
ALTER TABLE table_name  
ENABLE CONSTRAINT constraint_name [CASCADE]
```

#### [예제] 제약 조건 활성화

예 제

```
ALTER TABLE emp_copy  
ENABLE CONSTRAINT emp_copy_eno_fk ;
```