



응용 SW 기초 활용 기술 part 2

# 테이블 조작하기



한국기술교육대학교  
온라인평생교육원



## 학습목표



- 테이블 생성
- 테이블 수정 및 삭제



- SQL을 설명하고, 테이블을 생성할 수 있다.
- 테이블의 열을 수정하고, 테이블을 삭제할 수 있다.



## 테이블 생성

### 1 SQL의 개요

SQL

- Structured Query Language
- 구조화된 질의를 이용하는 언어
- 관계형 데이터베이스 관리 시스템에서 사용하는 대화식 프로그래밍 언어



데이터베이스 스키마 생성과 수정, 자료의 검색과 관리



## 테이블 생성

### 2 데이터베이스 생성

#### 1 데이터 베이스 생성

■ **CREATE DATABASE** 데이터베이스 명

예 **CREATE DATABASE** school;

■ 일반적으로 데이터베이스 명은 저장하는 데이터의 집합을 대표하는 이름을 사용



데이터베이스는 실제 데이터를 저장하고 있는 테이블의 집합

#### 2 데이터베이스 목록 확인

■ **SHOW DATABASES;**

#### 3 데이터베이스 삭제

■ **DROP DATABASE** 데이터베이스 명

예 **DROP DATABASE** testDB;



데이터베이스 삭제 시, 해당 데이터베이스 내의 테이블과 테이블에 저장된 데이터도 함께 삭제되기 때문에 삭제 시 유의해야 함



SQL은 대소문자를 구분하지 않으므로, 구문을 소문자로 입력해도 무방함



## 테이블 생성

### 3 테이블 생성

#### 1) 기본 테이블 생성

##### 테이블 생성

- 열의 이름과 데이터타입 등을 지정하는 것
- '테이블을 정의한다'라고도 함

##### 테이블 생성하기

```
CREATE TABLE 테이블 명(
    열1 데이터타입,
    열2 데이터타입,
    .....
);
```



##### 테이블 생성하기의 예

```
CREATE TABLE student(
    std_code char(10),
    name char(10),
    address varchar(50),
    birthday date,
    dep_code char(10) );
```

#### 데이터베이스 관리 시스템마다 제공 데이터타입은 다름

- 서로 다른 데이터베이스 관리 시스템에서 동일한 데이터타입 명을 사용할지라도 의미가 다를 수 있음

1986년 미국 표준 연구소인 ANSI와 국제 표준화 기구인 ISO에서 표준 SQL의 데이터타입을 제안



## 테이블 생성

### 3 테이블 생성

#### 1) 기본 테이블 생성

#### 데이터타입

| 데이터타입                                 | 설명                                |
|---------------------------------------|-----------------------------------|
| CHARACTER(n) 또는 CHAR(n)               | 길이가 n 고정 문자열로써,<br>빈 공간은 공백으로 채워짐 |
| CHARACTER VARYING(n) 또는<br>VARCHAR(n) | 최대 길이가 n인 가변 문자열                  |
| BIT(n)                                | n 비트의 배열                          |
| INTEGER 또는 INT                        | 정수                                |
| FLOAT, REAL, DOUBLE PRECISION         | 실수                                |
| DATE                                  | 날짜(년,월,일)                         |
| TIME                                  | 시간(시,분,초)                         |



## 테이블 생성

3

### 테이블 생성

#### 2) 제약조건(Constraint)이 정의된 테이블 생성

테이블에 저장되는 데이터를 위한 규칙을 지정할 때 **제약조건**을 적용

##### 구문

```
CREATE TABLE 테이블 명(
    열1 데이터타입 제약조건,
    열2 데이터타입 제약조건,
    .....
);
```

##### 제약조건

- 테이블을 생성하거나 테이블 수정 시 적용
- 선택 사항이며, 중복해서 적용 가능
- NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK, DEFAULT

## 테이블 생성

### 3 테이블 생성

#### 2) 제약조건(Constraint)이 정의된 테이블 생성

##### 1 NOT NULL

- 데이터가 NULL 값을 저장하지 못하도록 함
- 새로운 행을 추가할 때, NOT NULL로 지정된 열이 항상 값을 가질 수 있도록 함

**예** `CREATE TABLE student( std_code char(10) NOT NULL);`

##### 2 UNIQUE

- 데이터베이스 설계 시 대체 키를 지정하는 것
- 저장되는 값들이 모두 다르도록 함
- 하나의 테이블에 여러 개의 열에 적용 가능

**예** `CREATE TABLE student( std_code char(10) NOT NULL UNIQUE,  
name char(10));`

```
CREATE TABLE student( std_code char(10) NOT NULL,  
name char(10),  
UNIQUE(std_code) );
```

##### 3 PRIMARY KEY

- 설계 시 기본 키를 지정하기 위한 것
- 저장되는 값들이 모두 다르도록 제한함
- 하나의 테이블에 하나의 열에만 적용 가능
- 자동적으로 UNIQUE 제약 조건을 가짐
- NULL 값을 가질 수 없음
- 새로운 행 추가 시, 자동으로 고유의 숫자 값을 부여하는 AUTO\_INCREMENT를 함께 이용하면 편리함
  - AUTO\_INCREMENT는 데이터베이스 관리 시스템마다 다름

```
CREATE TABLE student( std_code char(10) NOT NULL PRIMARY KEY,  
name char(10));
```

```
CREATE TABLE student( std_code char(10) NOT NULL,  
name char(10),  
PRIMARY KEY(std_code) );
```



## 테이블 생성

### 3 테이블 생성

#### 2) 제약조건(Constraint)이 정의된 테이블 생성

##### 4 FOREIGN KEY

- 2개의 테이블을 연결하기 위해 사용
- REFERENCES를 이용하여 참조하는 테이블의 속성을 제시함

예

```
CREATE TABLE student( std_code char(10) NOT NULL PRIMARY KEY,  
                        name char(10) FOREIGN KEY REFERENCES department(dep_code));
```

```
CREATE TABLE student( std_code char(10) NOT NULL,  
                        name char(10),  
                        PRIMARY KEY(std_code),  
                        FOREIGN KEY (name) REFERENCES department(dep_code));
```

##### 5 CHECK

- 값의 범위를 제한하기 위해 사용

예

```
CREATE TABLE student( std_code char(10) NOT NULL PRIMARY KEY,  
                        age int CHECK (age >=18) );
```

```
CREATE TABLE student( std_code char(10) NOT NULL PRIMARY KEY,  
                        age int,  
                        CHECK (age >=18) );
```

##### 6 DEFAULT

- 기본 값을 지정하기 위해 사용
- 새로운 행을 추가할 때, 값을 따로 입력하지 않으면 DEFAULT로 지정한 값이 입력됨

예

```
CREATE TABLE student( std_code char(10) NOT NULL PRIMARY KEY,  
                        age int DEFAULT 18);
```



## 테이블 수정 및 삭제

### 1 테이블 수정

#### ALTER TABLE 구문

- 기존 테이블의 열(Column)을 추가 · 삭제 · 수정
- 기존 테이블의 다양한 제약 조건(Constraint)들을 추가 · 삭제

#### 1 열의 추가

- ALTER TABLE 테이블 명 ADD 열 이름 데이터타입 [제약 조건];

**예** ALTER TABLE department ADD position varchar(10) NOT NULL UNIQUE;

#### 2 열의 삭제

- ALTER TABLE 테이블 명 DROP COLUMN 열 이름;

**예** ALTER TABLE department DROP COLUMN position;



삭제를 원하는 열이 키로 지정되어 있을 경우, 먼저 키를 삭제해야 해당 열을 삭제할 수 있음



## 테이블 수정 및 삭제

### 1 테이블 수정

3

#### 제약 조건의 추가

- ALTER TABLE 테이블 명 ADD CONSTRAINT 제약조건이름 제약조건내용;
  - UNIQUE : UC\_이름, PRIMARY KEY : PK\_이름, FOREIGN KEY : FK\_이름



예 ALTER TABLE student ADD CONSTRAINT FK\_Student  
FOREIGN KEY(dep\_code) REFERENCES department(dep\_code);



기본 키를 추가할 경우, 기존 테이블에 기본 키로 설정된 열이 없어야 한다는 것과  
외래 키를 추가할 때, REFERENCES를 이용하여 참조하는 테이블과 열을 지정해야 함

4

#### 제약 조건의 삭제

- ALTER TABLE 테이블 명 DROP CONSTRAINT 제약조건이름;



예 ALTER TABLE student DROP CONSTRAINT FK\_Student;



데이터베이스 관리 시스템에 따라 테이블을 수정하는 SQL 구문이 다를 수 있음

## 테이블 수정 및 삭제

### 2 테이블 삭제

DROP TABLE 테이블 명;

예 DROP TABLE department;



테이블 삭제 시, 테이블 내의 데이터도 함께 삭제되므로 주의해야 함

## 테이블 생성

### + SQL의 개요

- SQL은 관계형 데이터베이스 시스템에서 데이터베이스를 생성 및 관리를 위해 사용하는 대화식 프로그래밍 언어임

### + 데이터베이스 생성

- CREATE DATABASE 데이터베이스 명; 구문을 이용하여 데이터베이스를 생성할 수 있음

### + 테이블 생성

- CREATE TABLE 테이블 명 다음에 열의 속성과 데이터타입, 제약조건 등을 입력하여 테이블을 생성할 수 있음
- 제약조건에는 NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK, DEFAULT 등이 있음

## 테이블 수정 및 삭제

### + 테이블 수정

- ALTER TABLE 테이블 명 ADD 열이름 데이터타입 [제약 조건]; 구문을 이용하여 새로운 열을 추가할 수 있음
- ALTER TABLE 테이블 명 DROP COLUMN 열이름; 구문을 이용하여 기존의 열을 삭제할 수 있음
- ALTER TABLE 테이블 명 ADD CONSTRAINT 제약조건이름 제약조건내용; 구문을 이용하여 새로운 제약조건을 추가할 수 있음
- ALTER TABLE 테이블 명 DROP CONSTRAINT 제약조건이름; 구문을 이용하여 기존의 제약조건을 삭제할 수 있음

### + 테이블 삭제

- DROP TABLE 테이블 명; 구문을 이용하여 테이블을 삭제할 수 있음



### + 데이터베이스 생성 관련 구문

| 기능           | 구문                        |
|--------------|---------------------------|
| 데이터베이스 생성    | CREATE DATABASE 데이터베이스 명; |
| 데이터베이스 목록 확인 | SHOW DATABASES;           |
| 데이터베이스 삭제    | DROP DATABASE 데이터베이스 명;   |