

# 02. Structured Query Language

## ▼ Index

[01. Data type](#)






[02. Structure](#)

[03. SQL](#)


## ▼ 01. Data type

### 정수형 소수점이 없는 숫자 데이터

※ 바이트: 컴퓨터의 저장공간 단위 중 하나

데이터 타입	바이트 수		표현 가능한 숫자 범위
TINYINT	1		-128 ~ 127
SMALLINT	2		-32,768 ~ 32,767
MEDIUMINT	3		약 -838백만 ~ 838백만
INT	4		약 -21억 ~ +21억
BIGINT	8		약 -900경 ~ +900경

### 실수형 소수점이 있는 숫자 데이터

데이터 타입	바이트 수		표현 가능한 숫자 범위
FLOAT	4		소수점 아래 7자리까지 표현
DOUBLE	8		소수점 아래 15자리까지 표현

### 문자형

※ 바이트: 컴퓨터의 저장공간 단위 중 하나

데이터 타입	최대 바이트 수	특징
CHAR(n)	255	n을 1부터 255까지 지정 가능, 지정 안 할 시 1 자동 입력 고정 길이로 문자열 저장.
VARCHAR(n)	65535	n을 1부터 65535까지 지정 가능, 지정 안 할 시 사용 불가 변동 길이로 문자열 저장.

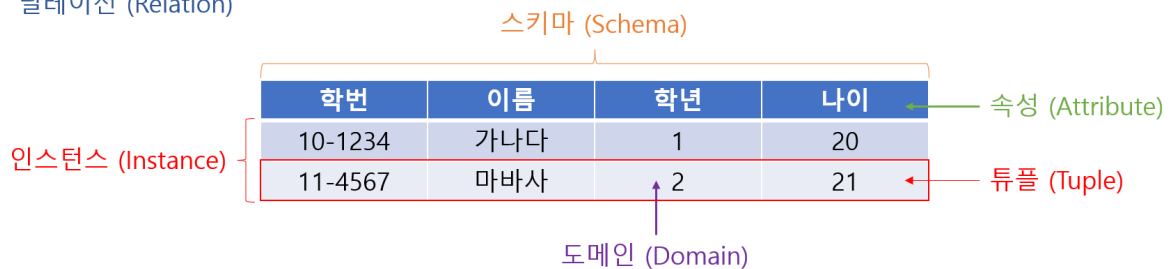
데이터 타입	고정 바이트 수	특징
TINYTEXT	255	255 바이트의 문자열까지 표현 가능
TEXT	65535	65535 바이트의 문자열까지 표현 가능
MEDIUMTEXT	약 천 6백만	약 천 6백만 바이트의 문자열까지 표현 가능
LONGTEXT	약 42억	약 42억 바이트의 문자열까지 표현 가능

## 날짜형

데이터 타입	바이트 수	표현 가능 범위
DATE	3	0000-00-00 ~ 9999-12-31
DATETIME	3	0000-00-00 00:00:00 ~ 9999-12-31 23:59:59
TIME	4	-838:59:59 ~ 838:59:59
YEAR	1	1901 ~ 2155

## ▼ 02. Structure

테이블 (Table)  
릴레이션 (Relation)



- 속성 (Attribute) == 컬럼 (Column)

## ▼ 03. SQL

### ▼ DDL : Data Definition Language - 데이터 정의를

#### ▼ CREATE : Database, Table 등 생성

##### ▼ Database

- Query

```
CREATE DATABASE [데이터베이스명];
```

- '데이터베이스명'으로 데이터베이스 생성

- Related features

```
SHOW DATABASES;
```

- 현재 존재하는 모든 데이터베이스 목록

```
USE [데이터베이스명];
```

- ‘데이터베이스명’ 사용 시작

## ▼ Table

- Query

```
CREATE TABLE [테이블명] (  
    [컬럼명] [데이터타입],  
    [컬럼명] [데이터타입],  
    ...  
);
```

- Related features

```
SHOW TABLES;
```

- 데이터베이스에 현재 존재하는 모든 테이블 목록

```
DESC [테이블명]
```

- 테이블 구조 보기

## ▼ ALTER : Table 수정

### ▼ Name

```
ALTER TABLE [OLD_테이블명] RENAME [NEW_테이블명];
```

### ▼ Column

```
# 속성(Attribute) 추가  
ALTER TABLE [테이블명] ADD COLUMN [속성명] [데이터타입];  
  
# 속성 타입 변경  
ALTER TABLE [테이블명] MODIFY COLUMN [속성명] [NEW_타입];  
  
# 속성명 & 타입 변경  
ALTER TABLE [테이블명] CHANGE COLUMN [OLD_속성명] [NEW_속성명] [타입];  
  
# 속성 삭제  
ALTER TABLE [테이블명] DROP COLUMN [속성명];
```

## ▼ DROP : Database, Table 등 삭제

### ▼ Database

```
DROP DATABASE [데이터베이스명];

# '존재한다면' 조건추가
DROP DATABASE IF EXISTS [데이터베이스명];
```

### ▼ Table

```
DROP TABLE [테이블명];

# '존재한다면' 조건추가
DROP TABLE IF EXISTS [테이블명];
```

## ▼ TRUNCATE : Table 초기화

```
TRUNCATE TABLE [테이블명];
```

### ※ DROP 과 TRUNCATE의 차이점



## ▼ DML : Data Manipulation Language - 데이터 조작어

### ▼ SELECT : 데이터 조회

```
SELECT * or (속성명, 속성명...) FROM ([데이터베이스명].)[테이블명];

# Case 01
SELECT * FROM 데이터베이스명.테이블명;

# Case 02
USING 데이터베이스명
SELECT * FROM 테이블명;
```

- Related features

```
# Alias : 출력 결과의 컬럼명을 새롭게 지정
SELECT 컬럼명 AS 별명 ...
FROM 데이터베이스명.테이블명;

# LIMIT : 가져올 데이터의 ROW 개수를 지정
SELECT 컬럼명 ...
FROM 데이터베이스명.테이블명
LIMIT 2; # 결과 인스턴스 출력 <= 2

# DISTINCT : 중복 데이터 제외
SELECT DISTINCT 컬럼명 ...
FROM 데이터베이스명.테이블명;
```

### ▼ INSERT : 데이터 삽입

```
INSERT INTO [테이블명] (속성1, 속성2, 속성3...)
VALUES (값1, 값2, 값3...),
       (값1, 값2, 값3...);
```

### ▼ UPDATE : 데이터 수정

```
UPDATE [테이블명]
SET [속성명] = [NEW_값]
WHERE [조건식];
```

### ▼ DELETE : 데이터 삭제

```
DELETE FROM [테이블명]
WHERE [조건식];
```