

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.

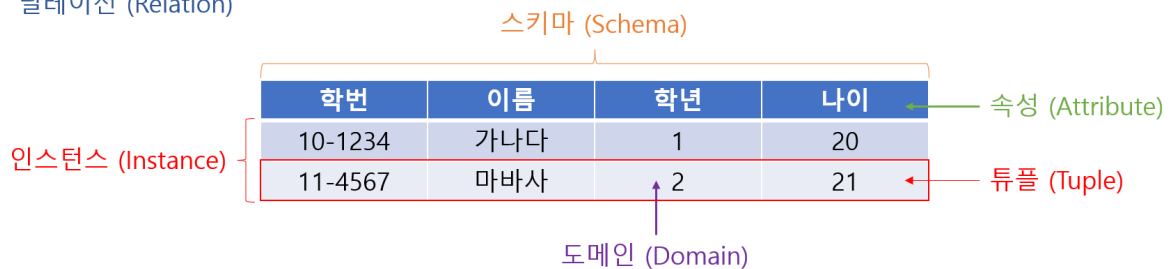
데이터 타입	고정 바이트 수	특징
TINYTEXT	255	255 바이트의 문자열까지 표현 가능
TEXT	65535	65535 바이트의 문자열까지 표현 가능
MEDIUMTEXT	약 천 6백만	약 천 6백만 바이트의 문자열까지 표현 가능
LONGTEXT	약 42억	약 42억 바이트의 문자열까지 표현 가능

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] [NEW_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];
```

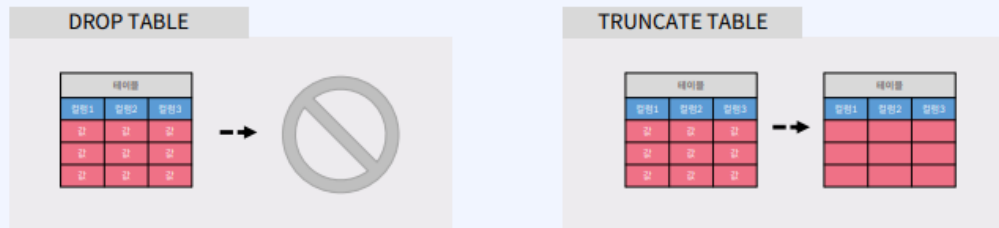
▼ Table

```
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

- ▼ 출력 결과 형식 지정

```

# 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;

```

- ▼ WHERE

```

# 문법
SELECT [컬럼명]
FROM [테이블명]
WHERE 조건식;

```

- ▼ 비교 연산자

연산자	활용	의미	예시
=	A = B	A와 B가 같다	1 = 1
!=	A != B	A와 B가 같지 않다	1 != 2
>	A > B	A가 B보다 크다	10 > 1
>=	A >= B	A가 B보다 크거나 같다	10 >= 10
<	A < B	A가 B보다 작다	10 < 100
<=	A <= B	A가 B보다 작거나 같다	10 <= 10

```

# =
SELECT number

```

```
FROM mypokemon
WHERE name='pikachu';
```

요청 : 피카츄의 number를 찾아주세요.

pokemon

mypokemon			
number	name	type	speed
10	caterpie	bug	45
25	pikachu	electric	90
26	raichu	electric	110
133	eevee	normal	55
152	chikorita	grass	45

쿼리

```
SELECT number
FROM mypokemon -- 테이블 사용(USE) 명시 했다고 가정
WHERE name = 'pikachu';
```

결과

number
▶ 25

```
# >
SELECT name
FROM mypokemon
WHERE speed > 50;
```

요청 : 속도가 50보다 큰
포켓몬의 이름을 찾아주세요.

pokemon

mypokemon			
number	name	type	speed
10	caterpie	bug	45
25	pikachu	electric	90
26	raichu	electric	110
133	eevee	normal	55
152	chikorita	grass	45

쿼리

```
SELECT name
FROM mypokemon
WHERE speed > 50;
```

결과

name
▶ pikachu
raichu
eevee

```
# !=
SELECT name
FROM mypokemon
WHERE type != 'electric';
```

요청 : 전기 타입이 아닌
포켓몬의 이름을 찾아주세요.

pokemon			
mypokemon			
number	name	type	speed
10	caterpie	bug	45
25	pikachu	electric	90
26	raichu	electric	110
133	eevee	normal	55
152	chikorita	grass	45

쿼리

```
SELECT name
FROM mypokemon
WHERE type != 'electric';
```

결과

name
caterpie
eevee
chikorita

▼ 논리 연산자

연산자	활용	의미
AND	A AND B	A와 B 모두 True이면 True
OR	A OR B	A와 B 둘 중 하나만 True이면 True
NOT	NOT A	A가 아니면 True

AND, &

```
SELECT name
FROM mypokemon
WHERE speed <= 100 AND type = 'electric';
```

요청 : **속도가 100 이하인**
전기 타입 포켓몬의 이름을 찾아주세요.

pokemon			
mypokemon			
number	name	type	speed
10	caterpie	bug	45
25	pikachu	electric	90
26	raichu	electric	110
133	eevee	normal	55
152	chikorita	grass	45

쿼리

```
SELECT name
FROM mypokemon
WHERE speed <= 100 AND type = 'electric';
```

결과

name
pikachu

OR, |

```
SELECT name
from mypokemon
WHERE type = 'bug' OR type = 'normal';
```

요청 : **벌레 타입**이거나 **노말 타입**인
포켓몬의 이름을 찾아주세요.

pokemon			
mypokemon			
number	name	type	speed
10	caterpie	bug	45
25	pikachu	electric	90
26	raichu	electric	110
133	eevee	normal	55
152	chikorita	grass	45

쿼리

```
SELECT name
FROM mypokemon
WHERE type = 'bug' OR type = 'normal';
```

결과

name
caterpie
eevee

```
# NOT, !
SELECT name
FROM mypokemon;
WHERE speed <= 100
      AND NOT(type='bug');
      or
      AND type != 'bug';
```

요청 : **속도가 100 이하**이고
벌레 타입이 아닌 포켓몬의 이름을 찾아주세요.

pokemon			
mypokemon			
number	name	type	speed
10	caterpie	bug	45
25	pikachu	electric	90
26	raichu	electric	110
133	eevee	normal	55
152	chikorita	grass	45

쿼리

```
SELECT name
FROM mypokemon
WHERE speed <= 100 AND NOT(type = 'bug');
```

결과

name
pikachu
eevee
chikorita

▼ 범위

```
# BETWEEN : 범위
SELECT name
FROM mypokemon
WHERE speed BETWEEN 50 AND 100;
( WHERE 50 <= speed and speed <= 100; )
```


요청 : 속도가 50과 100 사이인
포켓몬의 이름을 찾아주세요.

pokemon

mypokemon			
number	name	type	speed
10	caterpie	bug	45
25	pikachu	electric	90
26	raichu	electric	110
133	eevee	normal	55
152	chikorita	grass	45

쿼리

```
SELECT name
FROM mypokemon
WHERE speed BETWEEN 50 AND 100;
```

결과

name
pikachu
eevee

IN : 목록 내 포함

```
SELECT name
FROM mypokemon
WHERE type IN ('bug', 'normal');
( type = 'bug' or type = 'normal'; )
```

요청 : 벌레 타입이거나 노말 타입인
포켓몬의 이름을 찾아주세요.

pokemon

mypokemon			
number	name	type	speed
10	caterpie	bug	45
25	pikachu	electric	90
26	raichu	electric	110
133	eevee	normal	55
152	chikorita	grass	45

쿼리

```
SELECT name
FROM mypokemon
WHERE type IN ('bug', 'normal');
```

결과

name
caterpie
eevee

▼ 와일드카드

와일드카드	의미
%	0개 이상의 문자
_	1개의 문자

- '%e' 으로 끝나는 문자열
e, ee, eevee, apple, pineapple
- 'e%' 으로 시작하는 문자열
e, ee, eevee, eric
- '%e%' 이가 포함된 문자열
e, ee, eevee, apple, pineapple, aespа

- '_e' 으로 끝나고 e 앞에 1개 의 문자가 있는 문자열
ae, ee, ce
- '%_e' 으로 끝나고 e 앞에 1개 이상의 문자가 있는 문자열
ee, eevee, apple, pineapple
- '%_e_%' 을 포함하고 e 앞 뒤로 각각 1개 이상의 문자가 있는 문자열
eevee, aespа

LIKE

SELECT [컬럼명]

FROM [테이블명]

WHERE [컬럼명] LIKE [검색할 문자열];

요청 : 이름이 'chu'로 끝나는
포켓몬의 이름을 찾아주세요.

pokemon			
mypokemon			
number	name	type	speed
10	caterpie	bug	45
25	pikachu	electric	90
26	raichu	electric	110
133	eevee	normal	55
152	chikorita	grass	45

쿼리

```
SELECT name
FROM mypokemon
WHERE name LIKE '%chu';
```

결과

name
pikachu
raichu

요청 : 이름에 'a'가 포함되는
포켓몬의 이름을 찾아주세요.

pokemon

mypokemon			
number	name	type	speed
10	caterpie	bug	45
25	pikachu	electric	90
26	raichu	electric	110
133	eevee	normal	55
152	chikorita	grass	45

쿼리

```
SELECT name
FROM mypokemon
WHERE name LIKE '%a%';
```

결과

name
caterpie
pikachu
raichu
chikoirita

▼ NULL

쿼리

```
INSERT INTO mypokemon (name, type)
VALUES ('kkobugi', '');
```

```
SELECT *
FROM mypokemon;
```

결과

number	name	type	height	weight	attack	defense	speed
▶ 10	caterpie	bug	0.3	2.9	30	35	45
25	pikachu	electric	0.4	6	55	40	90
26	raichu	electric	0.8	30	90	55	110
133	eevee	normal	0.3	6.5	55	50	55
152	chikoirita	grass	0.9	6.4	49	65	45
	NULL	kkobugi		NULL	NULL	NULL	NULL

공백

IS NULL

```
SELECT [컬럼명]
FROM [테이블명]
WHERE [컬럼명] IS NULL;
```

요청 : number가 null인
포켓몬의 이름을 찾아주세요.

pokemon			
mypokemon			
number	name	type	speed
10	caterpie	bug	45
25	pikachu	electric	90
26	raichu	electric	110
133	eevee	normal	55
152	chikorita	grass	45
NULL	kkobugi		NULL

쿼리

```
SELECT name
FROM mypokemon
WHERE number IS NULL;
```

결과

name
▶ kkobugi

```
# IS NOT NULL
SELECT [컬럼명]
FROM [테이블명]
WHERE [컬럼명] IS NOT NULL;
```

요청 : type이 null이 아닌
포켓몬을 찾아주세요.

pokemon			
mypokemon			
number	name	type	speed
10	caterpie	bug	45
25	pikachu	electric	90
26	raichu	electric	110
133	eevee	normal	55
152	chikorita	grass	45
NULL	kkobugi		NULL

쿼리

```
SELECT name
FROM mypokemon
WHERE type IS NOT NULL;
```

결과

name
▶ caterpie
▶ pikachu
▶ raichu
▶ eevee
▶ chikorita
▶ kkobugi

▼ 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];
```
