

02. Database

▼ Index

01. Data type

02. Structure

03. SQL

[실습 1] 테이블을 만들고 데이터를 넣어보자.

Mission 01. '포켓몬(pokemon)' 데이터베이스와 '나의 포켓몬(mypokemon)' 테이블을 만들고, 캐터피, 피카츄, 이브이의 포켓몬 번호, 영문 이름, 타입 데이터를 넣어주세요.

Mission 02. Missoin 01에서 만든 '포켓몬' (pokemon) 데이터베이스에 '나의 새로운 포켓몬' (mynewpokemon) 테이블을 만들고, 포니타, 메타몽, 뮤의 포켓몬 번호, 이름, 타입 데이터를 넣어주세요.

[실습 2] 테이블을 변경하고 지워보자.






Mission 01. '포켓몬 (pokemon)' 데이터베이스 안에 있는 '나의 포켓몬 (mypokemon)' 테이블과 '나의 새로운 포켓몬 (mynewpokmon)'

Mission 02. 'pokemon' 데이터베이스 안에 있는 'myoldpokemon' 테이블 값만 지우고, 'mynewpokemon' 테이블은 전부 지워주세요.



▼ 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.

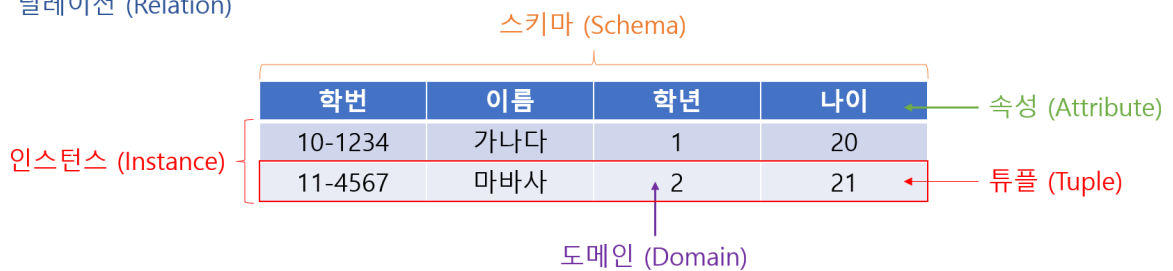
데이터 타입	고정 바이트 수	특징
TINYTEXT	255	255 바이트의 문자열까지 표현 가능
TEXT	65535	65535 바이트의 문자열까지 표현 가능
MEDIUMTEXT	약 천 6백만	약 천 6백만 바이트의 문자열까지 표현 가능
LONGTEXT	약 42억	약 42억 바이트의 문자열까지 표현 가능

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 정의어

▼ 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 [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];
```

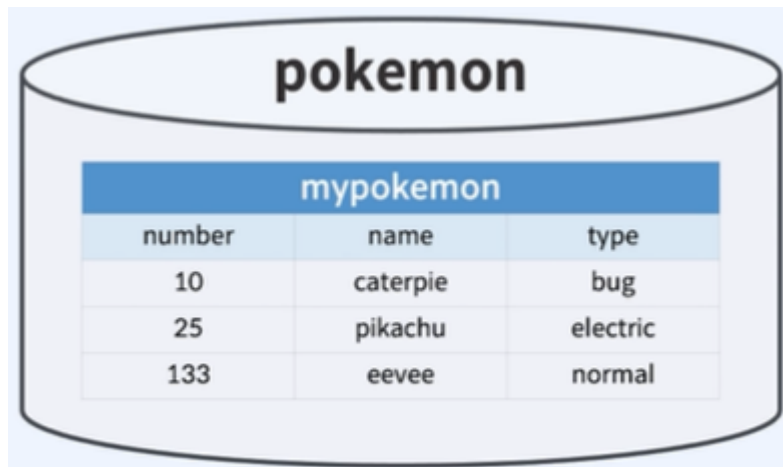
▼ DELETE : 데이터 삭제

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

[실습 1] 테이블을 만들고 데이터를 넣어보자.

Mission 01. ‘포켓몬(pokemon)’ 데이터베이스와 ‘나의 포켓몬 (mypokemon)’ 테이블을 만들고, 캐터피, 피카츄, 이브이의 포켓몬 번호, 영문 이름, 타입 데이터를 넣어주세요.

▼ Example



실습 1. Mission 01. Example

▼ Scenario

쿼리 순서 (1 STEP당 1개의 쿼리를 만들어 주세요.)

STEP1. 포켓몬 데이터베이스를 만들어 주세요. (데이터베이스 이름 : pokemon)

STEP2. 포켓몬 데이터베이스 안에 나의 포켓몬 테이블을 만들어 주세요. (테이블 이름 : mypokemon)

이 때, 컬럼은 (1) 포켓몬 번호, (2) 포켓몬 영문 이름, (3) 포켓몬 타입, 총 3가지로 하고, 이름과 데이터 타입을 지정해 주세요.

컬럼 이름 및 데이터 타입 : (1) number : INT, (2) name : VARCHAR(20), (3) type : VARCHAR(10)

STEP3. 포켓몬 테이블 안에 포켓몬 데이터를 각각의 로우로 넣어 주세요.

[포켓몬 정보 표]

포켓몬	번호	영문 이름	타입
캐터피	10	caterpie	bug
피카츄	25	pikachu	electric
이브이	133	eevee	normal

실습 1. Mission 01. Scenario

▼ Solve

▼ Step 01

```
# Step 01. Database 생성
create database pokemon;
show databases;
```

Result Grid	Filter Rows
Database	
information_schema	
mysql	
nbe_db	
performance_schema	
▶ pokemon	
sakila	
sys	
world	

▼ Step 02

```
# Step 02. Table 생성
use pokemon;
create table mypokemon(
    number int,
    name varchar(20),
    type varchar(10)
);
show tables;
desc mypokemon;
```

Result Grid	Filter Rows
Tables_in_pokemon	
▶ mypokemon	

Result Grid

Filter Rows:

Export:

	Field	Type	Null	Key	Default	Extra
▶	number	int	YES		NULL	
	name	varchar(20)	YES		NULL	
	type	varchar(10)	YES		NULL	

▼ Step 03

```
# Step 03. Data 추가
insert into mypokemon(number, name, type)
value (10, 'caterpie', 'bug'),
      (25, 'pikachu', 'electirc'),
      (133, 'eevee', 'normal');
select * from mypokemon;
```

Result Grid			
	number	name	type
▶	10	caterpie	bug
	25	pikachu	electirc
	133	eevee	normal

Mission 02. Missoin 01에서 만든 ‘포켓몬’ (pokemon) 데이터베이스에 ‘나의 새로운 포켓몬’ (mynewpokemon) 테이블을 만들고, 포니타, 메타몽, 뮤의 포켓몬 번호, 이름, 타입 데이터를 넣어주세요.

▼ Example

pokemon			
mynewpokemon			
number	name	type	
77	포니타	불꽃	
132	메타몽	노말	
151	뮤	에스퍼	

실습 1. Mission 02. Example

▼ Scenario

쿼리 순서 (1 STEP당 1개의 쿼리를 만들어 주세요.)

STEP1. 포켓몬 데이터베이스 안에 나의 새로운 포켓몬 테이블을 만들어 주세요. (테이블 이름 : mynewpokemon)

이 때, 컬럼은 (1) 포켓몬 번호, (2) 포켓몬 영문 이름, (3) 포켓몬 타입, 총 3가지로 하고, 이름과 데이터 타입을 지정해 주세요.
컬럼 이름 및 데이터 타입 : (1) number : INT, (2) name : VARCHAR(20), (3) type : VARCHAR(10)

STEP2. ‘나의 새로운 포켓몬’ 테이블 안에 포켓몬 데이터를 각각의 로우로 넣어 주세요.

[포켓몬 정보 표]			
포켓몬	번호	이름	타입
포니타	77	포니타	불꽃
메타몽	132	메타몽	노말
뮤	151	뮤	에스퍼

실습 1. Mission 02. Scenario

▼ Solve

▼ Step 01


```
# table 생성
create table mynewpokemon (
    number int,
    name varchar(20),
    type varchar(10)
);
show tables;
desc mypokemon;
```

Result Grid | Filter Rows

	Tables_in_pokemon
▶	mynewpokemon
	mypokemon

Result Grid | Filter Rows: | Export:

	Field	Type	Null	Key	Default	Extra
▶	number	int	YES		NULL	
	name	varchar(20)	YES		NULL	
	type	varchar(10)	YES		NULL	

▼ Step 02

```
# Step 02. 데이터 추가
insert into mynewpokemon (number, name, type)
value (77, '포니타', '불꽃'),
      (132, '메타몽', '노말'),
      (151, '뮤', '에스퍼');
select * from mynewpokemon;
```

Result Grid | Filter Rows:

	number	name	type
▶	77	포니타	불꽃
	132	메타몽	노말
	151	뮤	에스퍼

[실습 2] 테이블을 변경하고 지워보자.

Mission 01. ‘포켓몬 (pokemon)’ 데이터베이스 안에 있는 ‘나의 포켓몬 (mypokemon)’ 테이블과 ‘나의 새로운 포켓몬 (mynewpokmon)’

▼ Example

pokemon					
myoldpokemon			mynewpokemon		
number	eng_name	type	number	kor_name	type
10	caterpie	bug	77	포니타	불꽃
25	pikachu	electric	132	메타몽	노말
133	eevee	normal	151	뮤	에스퍼

실습 2. Mission 01. Example

▼ Scenario

쿼리 순서 (1 STEP당 1개의 쿼리를 만들어 주세요.)

STEP1. 'mypokemon' 테이블의 이름을 'myoldpokemon'으로 변경해 주세요.

STEP2. 'myoldpokemon' 테이블의 'name' 컬럼의 이름을 'eng_nm'으로 변경해 주세요.
(컬럼 이름 : eng_nm, 데이터 타입 : VARCHAR(20))

STEP3. 'mynewpokemon' 테이블의 'name' 컬럼의 이름을 'kor_nm'으로 변경해 주세요.
(컬럼 이름 : kor_nm, 데이터 타입 : VARCHAR(20))

실습 2. Mission 01. Scenario

▼ Solve

▼ Step 01

```
# Step 01. 테이블 이름 변경
alter table mypokemon rename myoldpokemon;
show tables;
```

Result Grid	Filter Row
Tables_in_pokemon	
mynewpokemon	
myoldpokemon	

▼ Step 02

```
# Step 02. Column Name 변경
alter table myoldpokemon
change column name eng_nm varchar(20);
desc myoldpokemon;
```

Result Grid						
	Field	Type	Null	Key	Default	Extra
	number	int	YES		NULL	
▶	eng_nm	varchar(20)	YES		NULL	
	type	varchar(10)	YES		NULL	

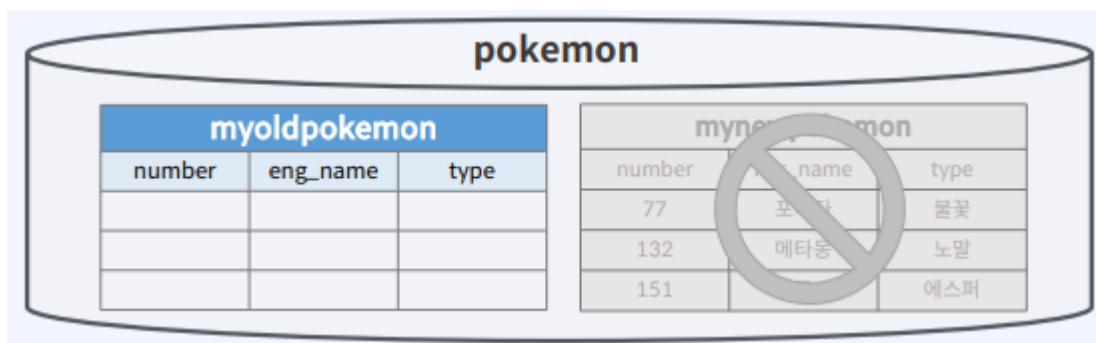
▼ Step 03

```
# Step 03. Column Name 변경
alter table mynewpokemon
change column name kor_nm varchar(20);
desc mynewpokemon;
```

Result Grid						
	Field	Type	Null	Key	Default	Extra
▶	number	int	YES		NULL	
	kor_nm	varchar(20)	YES		NULL	
	type	varchar(10)	YES		NULL	

Mission 02. 'pokemon' 데이터베이스 안에 있는 'myoldpokemon' 테이블 값만 지우고, 'mynewpokemon' 테이블은 전부 지워주세요.

▼ Example



실습 2. Mission 02. Example

▼ Solve

▼ Step 01

```
# Step 01. Table Value 지우기
truncate table myoldpokemon;
select * from myoldpokemon;
```

Result Grid		Filter Rows:
	number	eng_nm type

▼ Step 02

```
# Step 02. Table 삭제
Drop table if exists mynewpokemon;
show tables;
```

Result Grid		Filter Rows
	Tables_in_pokemon	
▶	myoldpokemon	