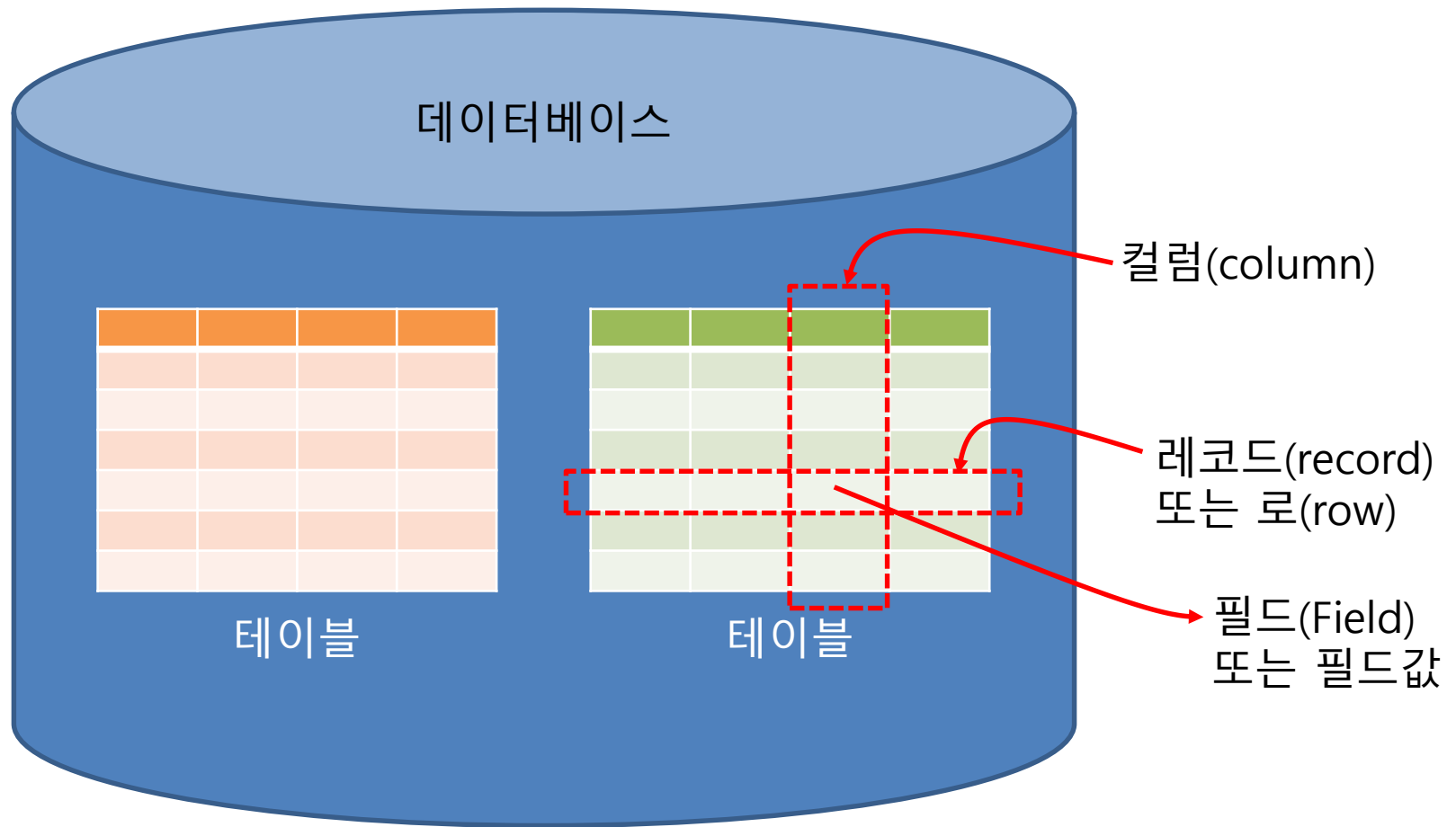


MySQL

A close-up, shallow depth-of-field photograph of a person's hands typing on a white Apple keyboard. The hands are positioned in the lower right foreground. The background is softly blurred, showing a white desk, a white cup, and the edge of a computer monitor. The overall lighting is warm and natural.

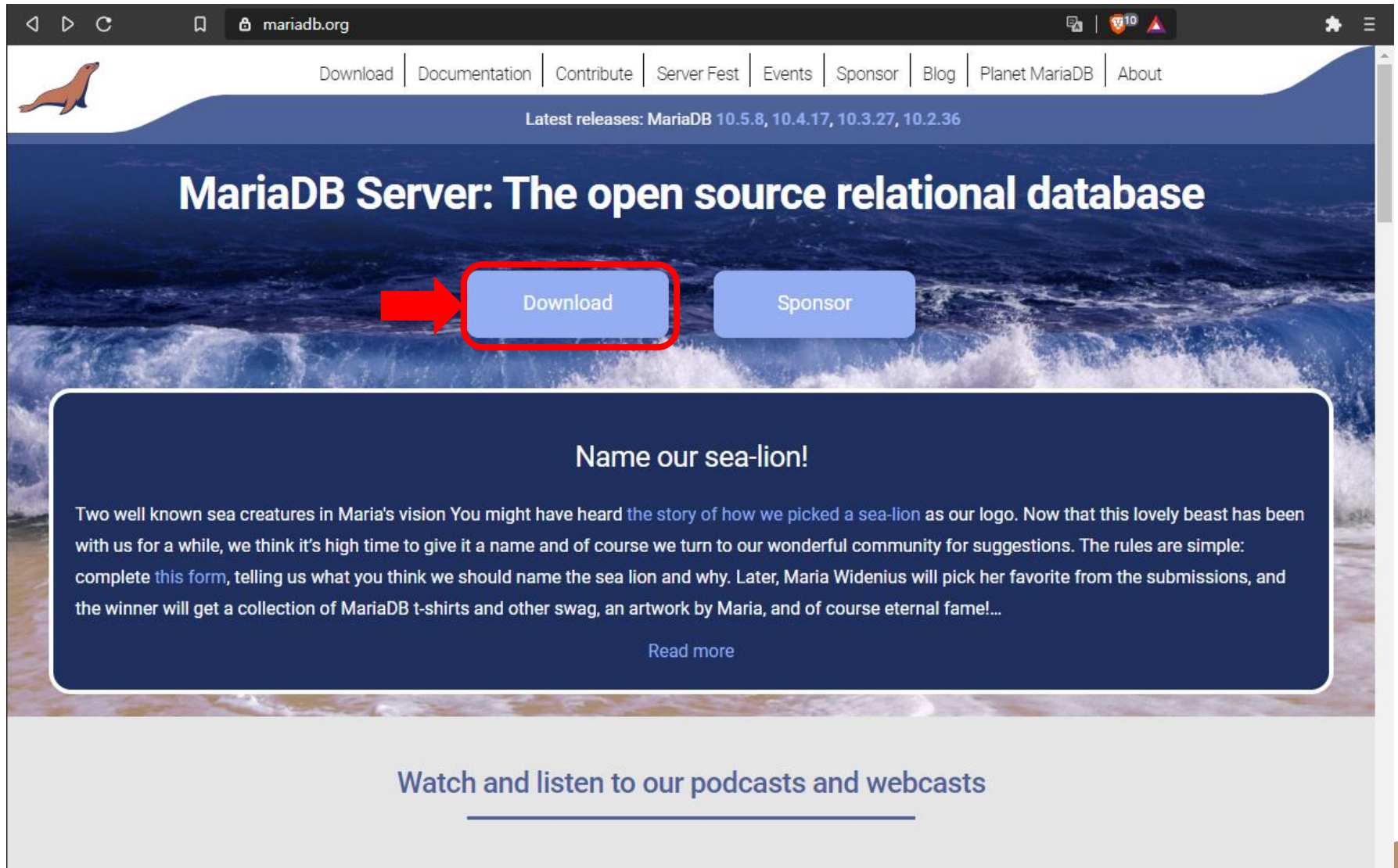
2023.06

데이터베이스 구조



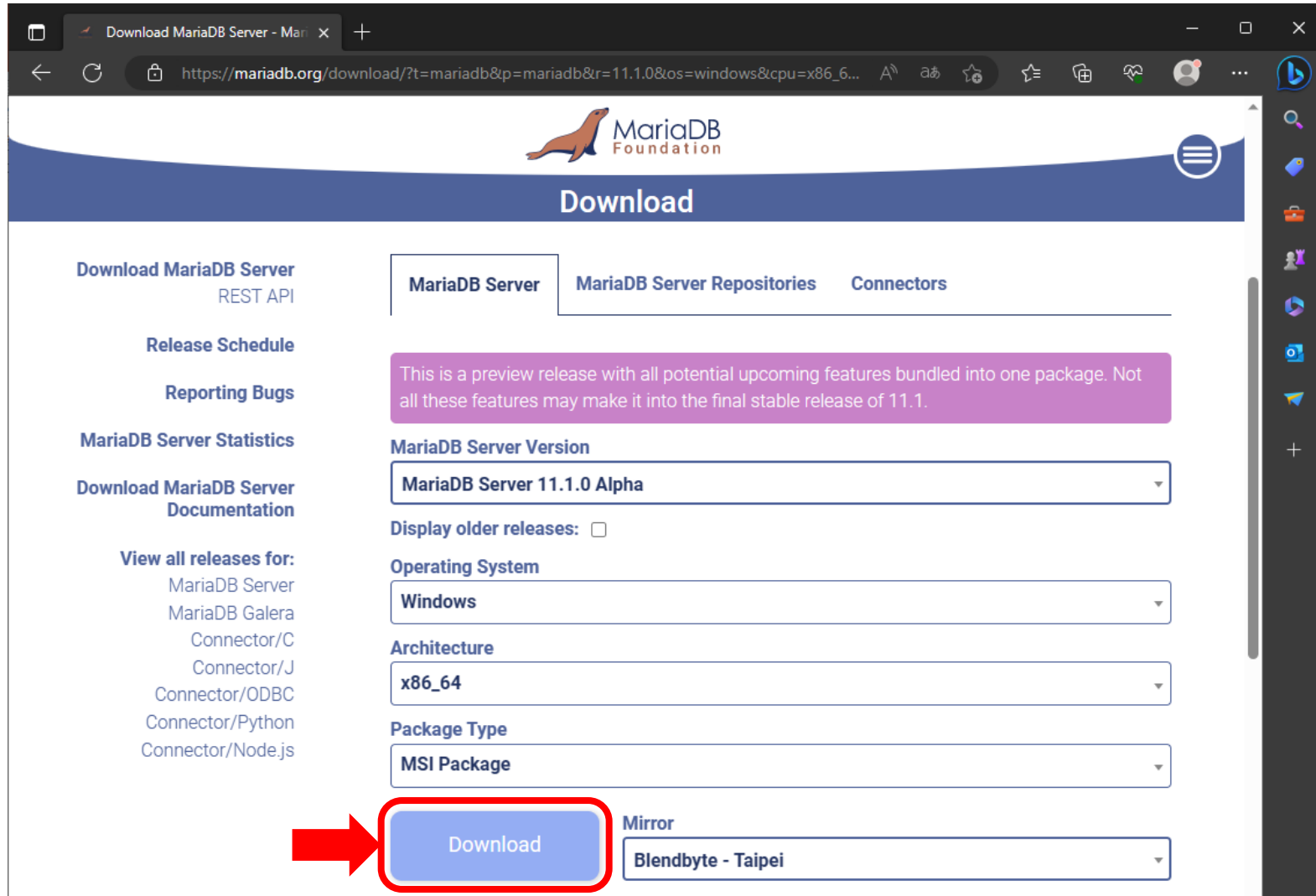
Maria DB Website

<http://mariadb.org> 에 방문합니다.



Maria DB 다운로드

MariaDB의 최신 버전 다운로드



The screenshot shows the MariaDB download page in a web browser. The browser's address bar displays the URL: `https://mariadb.org/download/?t=mariadb&p=mariadb&r=11.1.0&os=windows&cpu=x86_64...`. The page features the MariaDB Foundation logo and a 'Download' header. On the left, there is a sidebar with links: 'Download MariaDB Server REST API', 'Release Schedule', 'Reporting Bugs', 'MariaDB Server Statistics', 'Download MariaDB Server Documentation', and 'View all releases for:' followed by a list of connectors. The main content area has three tabs: 'MariaDB Server' (selected), 'MariaDB Server Repositories', and 'Connectors'. A purple warning box states: 'This is a preview release with all potential upcoming features bundled into one package. Not all these features may make it into the final stable release of 11.1.' Below this, there are four dropdown menus: 'MariaDB Server Version' (set to 'MariaDB Server 11.1.0 Alpha'), 'Operating System' (set to 'Windows'), 'Architecture' (set to 'x86_64'), and 'Package Type' (set to 'MSI Package'). There is also a checkbox for 'Display older releases:' which is unchecked. At the bottom, a blue 'Download' button is highlighted with a red rectangle and a red arrow pointing to it from the left. To the right of the button is a 'Mirror' dropdown menu set to 'Blendbyte - Taipei'.

Download MariaDB Server
REST API

Release Schedule

Reporting Bugs

MariaDB Server Statistics

Download MariaDB Server Documentation

View all releases for:

- MariaDB Server
- MariaDB Galera
- Connector/C
- Connector/J
- Connector/ODBC
- Connector/Python
- Connector/Node.js

MariaDB Server

MariaDB Server Repositories

Connectors

This is a preview release with all potential upcoming features bundled into one package. Not all these features may make it into the final stable release of 11.1.

MariaDB Server Version

MariaDB Server 11.1.0 Alpha

Display older releases: ☐

Operating System

Windows

Architecture

x86_64

Package Type

MSI Package

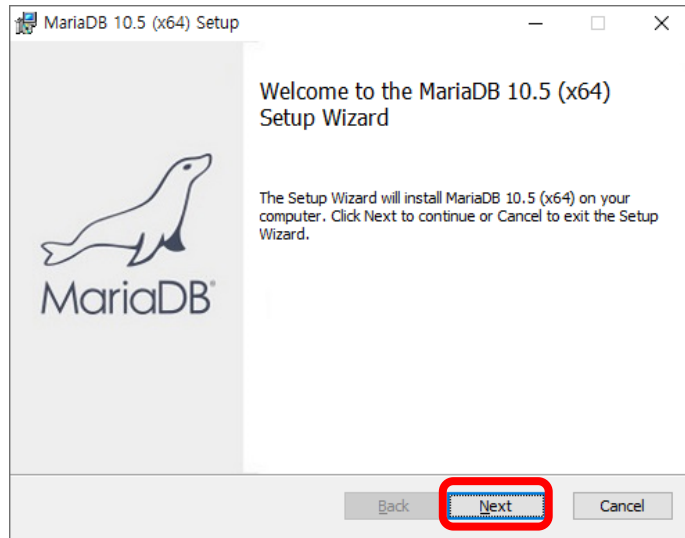
Download

Mirror

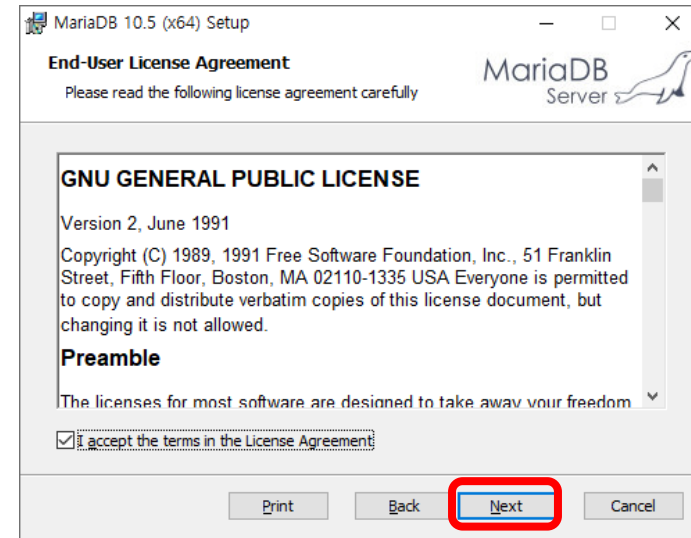
Blendbyte - Taipei

Maria DB 설치 순서 (1/2)

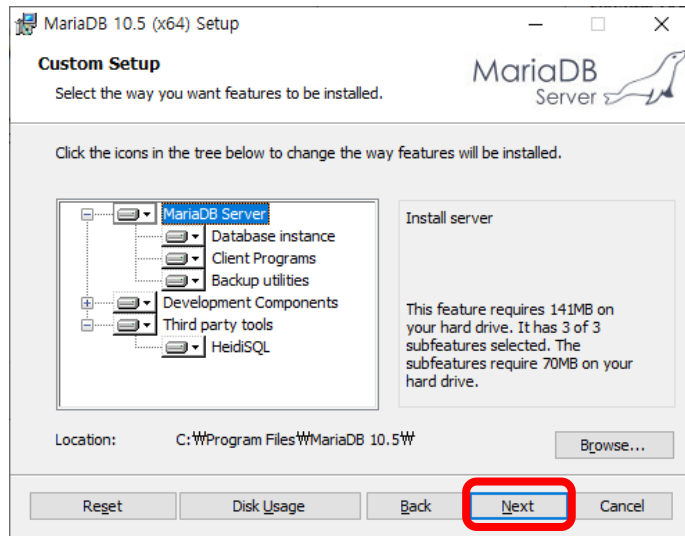
1



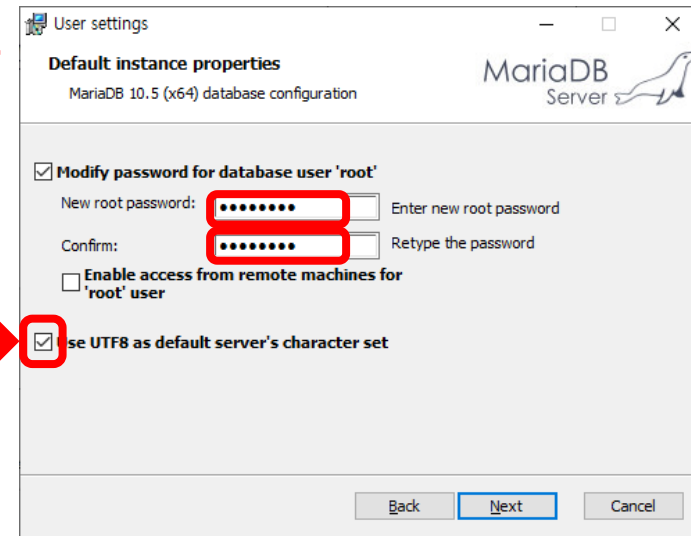
2



3

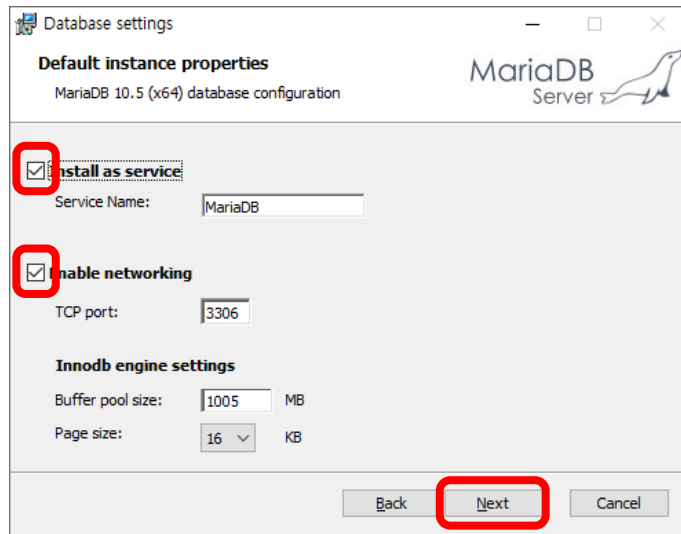


4

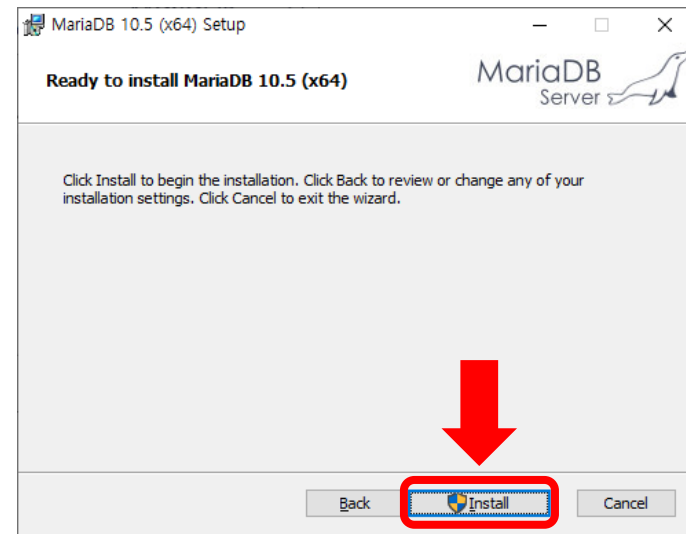


Maria DB 설치 순서 (2/2)

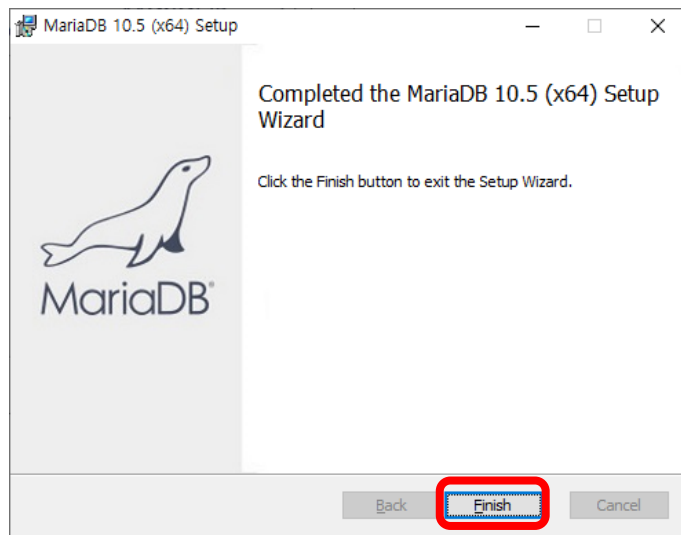
1



2



3



Maria DB 접속하기

실행창에 cmd를 입력하고 커맨드 창이 뜨면 아래와 같이 입력

cd ₩

cd program files

cd mariaDB<tab>

cd bin

mysql -u root -p

```
명령 프롬프트 - mysql -u root -p
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\₩크리스 크로스>cd ₩
C:\₩>cd program files
C:\₩Program Files>cd "MariaDB 10.5"
C:\₩Program Files\₩MariaDB 10.5>cd bin
C:\₩Program Files\₩MariaDB 10.5\₩bin>mysql -u root -p
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or ₩g.
Your MariaDB connection id is 4
Server version: 10.5.8-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '₩h' for help. Type '₩c' to clear the current input statement.
MariaDB [(none)]> _
```

데이터베이스 생성과 조회

* 데이터베이스 생성하기
create database [DB이름];

* 데이터베이스 조회하기
show databases;

```
MariaDB [(none)]> create database mydb;  
Query OK, 1 row affected (0.000 sec)  
  
MariaDB [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| Kset     |  
| classicmodels |  
| information_schema |  
| mydb     |  
| mysql    |  
| performance_schema |  
| petshop  |  
+-----+  
7 rows in set (0.022 sec)
```

명령어 끝에는
반드시 ;(세미콜론)

;을 사용하지 않고 엔터를 친 경우,
;을 추가로 붙이면 됨
먹통이 되어 프롬프트가 안보이면
\c를 입력하고 엔터
만일, "가 열려있는 경우라면 "를
입력하고 \c를 사용하면 됨("\c)

이 문장을 실행하면 MySQL data 디렉토리에 db_name이란 서브디렉토리가 생성된다.
만일 같은 이름의 data base 디렉토리가 있으면 에러메시지를 내보낸다.

데이터베이스 삭제와 조회

* 데이터베이스 삭제

drop database [DB이름];

```
MariaDB [(none)]> drop database mydb;
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| Kset      |
| classicmodels |
| information_schema |
| mysql     |
| performance_schema |
| petshop   |
+-----+
6 rows in set (0.000 sec)
```

IF EXISTS: 삭제하려는 데이터베이스가 없는 경우에 에러메시지를 막아줌
drop database if exists mydb;

조회할 때에는 database가 여러 개(복수)일 것이므로 databases라고 해야 함.

데이터베이스 생성 및 사용

새로운 database world를 만들고, 이를 handling하기 위한 명령문은 다음과 같다.

```
MariaDB [(none)]> create database world;  
Query OK, 1 row affected (0.022 sec)  
  
MariaDB [(none)]> use world;  
Database changed  
MariaDB [world]>  
MariaDB [world]>  
MariaDB [world]>  
MariaDB [world]>
```

* 데이터베이스를 사용하려면 use라는 명령을 사용한다.

use [DB이름];

문자셋 입력하기

- 문자셋(Character Set): 문자의 인코딩 규칙이 정의된 집합 (ASCII, EUC-KR, UTF-8 등)
- 콜레이션(Collation): 문자셋으로 인코딩된 문자들간 비교시 사용하는 규칙집합
- 데이터베이스 생성시 character set과 collation을 지정하는 방법:

```
MariaDB [world]> create database korean  
-> default charset utf8  
-> default collate utf8_general_ci;  
Query OK, 1 row affected (0.000 sec)
```

collation

→ MySQL에서는 Collation이라는 속성을 가지고 있어서 문자 데이터 타입에 대해 어떻게 다룰것인가를 명시함

utf8은 여러 나라의 문자를 표시할 수 있는 unicode이므로 일반적인 general_ci를 선택한다. (ci를 뒤에 반드시 붙여야 함. ci=code information)

ex) default collation persian_ci //페르시아 언어를 Default로 선택

테이블 생성

CREATE TABLE [IF NOT EXISTS] 테이블이름(
컬럼이름 자료속성 기타속성 정의
);

```
MariaDB [korean]> create table city(  
-> city_name varchar(10),  
-> phone_no int(3) primary key,  
-> people bigint,  
-> color char(3)  
-> );  
Query OK, 0 rows affected (0.018 sec)
```

테이블 이름 : city

city_name	phone_no	people	color

만들어진 테이블 확인 방법

```
MariaDB [korean]> desc city;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| city_name  | varchar(10)   | YES  |     | NULL     |       |  
| phone_no   | int(3)        | NO   | PRI | NULL     |       |  
| people     | bigint(20)    | YES  |     | NULL     |       |  
| color      | char(3)       | YES  |     | NULL     |       |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.014 sec)
```

컬럼(Column) 속성 지정

컬럼속성:

INTEGER[n], INT[n]: n자리의 정수

REAL: 실수값 지정

CHAR(n): 고정크기(n) 문자열

VARCHAR(n): 가변크기 문자열

DATE

TIME

컬럼 기타 속성:

PRIMARY KEY : 기본키로 설정

NOT NULL : 반드시 값을 가져야 함을 의미

AUTO_INCREMENT : 레코드가 삽입될 때 마다 자동으로 값을 증가

DEFAULT default_value: 디폴트값을 지정

컬럼(Column) 속성 지정

유 형	자 료 형	크 기	설 명
정수	TINYINT	1 byte	-128 ~ 127 (부호 없으면 0 ~ 255)
	SMALLINT	2 byte	-32,768 ~ 32,767
	MEDIUMINT	3 byte	-2 ²⁴ (-8,388,608) ~ 2 ²⁴ - 1 (8,388,607)
	INT	4 byte	-2 ³² ~ 2 ³² - 1 (2,147,483,647)
	BIGINT	8 byte	-2 ⁶⁴ ~ 2 ⁶⁴ -1 (9,223,372,036,854,775,807)
부정확한 숫자	FLOAT	4 byte	-3.40E + 38 ~ 3.40E + 38
	DOUBLE	8 byte	-1.79E + 308 ~ 1.79E + 308
정확한 숫자	DECIMAL		숫자데이터지만 내부적으로 String형태로 저장됨
날짜	DATE	3 byte	1000-01-01 ~ 9999-12-31
	DATETIME	8 byte	1000-01-01 00:00:00 ~ 9999-12-31...
	TIMESTAMP	4 byte	1970-01-01 00:00:00 ~ 2037 년 까지
	TIME	3 byte	-839:59:59 ~ 839:59:59
	YEAR	1 byte	1901 ~ 2155 년
문자	CHAR		1~255 개의 고정길이 문자열
	VARCHAR		1~255 개의 가변길이 문자열
	TINYBLOB, TINYTEXT		최대길이 255 (2 ⁸ - 1)
	BLOB, TEXT		최대길이 65,535 (2 ¹⁶ - 1)
	MEDIUMBLOB, MEDIUMTEXT		최대길이 16,777,215
	LONGBLOB, LONGTEXT		최대길이 4,294,967,295 (2 ³² - 1)
특수	ENUM		최대 65535 개 목록 중에서 하나를 택함
	SET		최대 64 개 목록 중에서 하나를 택함
	BOOL		TRUE, FALSE

PK(Primary Key)

테이블을 생성할 때 PK를 정의한다. (PK : Primary Key, 주키 또는 기본키)
PK는 각 행을 고유하게 식별하는 역할을 담당한다.

테이블당 하나만 정의 가능하다.

지정된 컬럼에는 중복된 값이나 NULL값이 입력될 수 없다.

NOT NULL + UNIQUE(UK)를 한 것과 같은 기능을 한다.

PK로 지정 가능한 컬럼이 여러 개 있을 때는 검색에 많이 사용되며,
간단하고 짧은 컬럼을 지정한다.

주 식별자, 주키 등으로 불린다.

고유 인덱스(Unique index)가 자동으로 생성된다.



FK(Foreign Key)

테이블을 생성할 때 FK를 정의한다. (FK : Foreign Key, 외래키)

FK가 정의된 테이블이 자식 테이블이다.

참조되는 테이블을 부모 테이블이라고 한다.

부모 테이블은 미리 생성되어 있어야 한다.

부모 테이블의 참조되는 컬럼에 존재하는 값을 입력 할 수 있다.

부모 테이블은 FK로 인해 삭제가 불가능하다.

REFERENCES : 참조할 부모 테이블과 부모 테이블에 있는 컬럼을 정의한다.

ON DELETE CASCADE : 참조되는 부모 테이블의 행에 대한 DELETE를 허용한다.

부모 테이블의 행이 지워지면 자식 테이블의 행도 같이 지워진다.

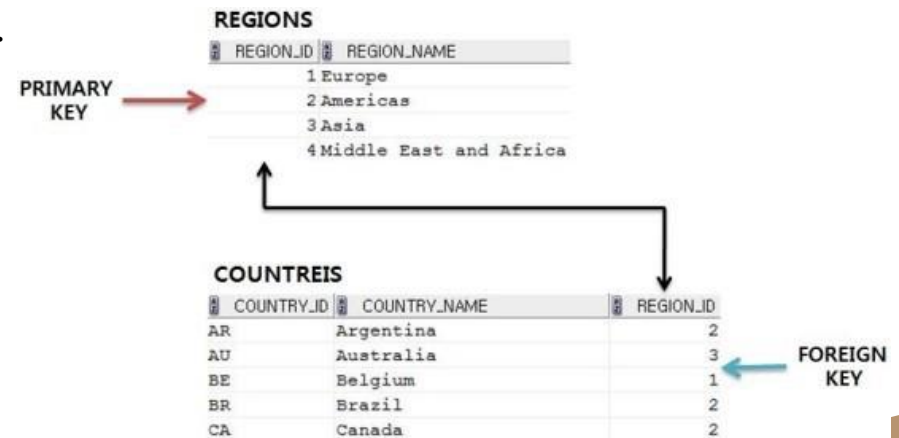
ON DELETE SET NULL : 참조되는 부모 테이블의 행에 대한 DELETE를 허용한다.

부모 테이블의 행이 지워지면 자식 테이블의 행은 NULL 값으로 설정된다.

데이터 타입이 반드시 일치해야 한다.

참조되는 컬럼은 PK이거나 UK(Unique key)만 가능하다.

외부키, 참조키, 외부 식별자 등으로 불린다.



drop : 테이블 삭제

* 테이블 삭제

drop table [테이블명];

```
MariaDB [korean]> show tables;
+-----+
| Tables_in_korean |
+-----+
| city              |
| city_1            |
+-----+
2 rows in set (0.000 sec)

MariaDB [korean]>
MariaDB [korean]>
MariaDB [korean]> drop table city_1;
Query OK, 0 rows affected (0.071 sec)

MariaDB [korean]> show tables;
+-----+
| Tables_in_korean |
+-----+
| city              |
+-----+
1 row in set (0.000 sec)
```

주의!

테이블 삭제 명령은 한번 명령을
실행하면 되돌릴 수 없기 때문에
매우 신중해야 한다.

insert : 데이터 추가

insert into 테이블명 (필드명1, 필드명2, ...) values (필드값1, 필드값2, ...);

```
MariaDB [korean]> insert into city value('Seoul',002,10000000,'BLU');
Query OK, 1 row affected (0.184 sec)

MariaDB [korean]>
MariaDB [korean]> insert into city value('Incheon',031,3000000,'WHT');
Query OK, 1 row affected (0.014 sec)

MariaDB [korean]> insert into city value('Daegu',052,4000000,'RED');
Query OK, 1 row affected (0.001 sec)

MariaDB [korean]> insert into city value('Gwangju',063,2000000,'YLW');
Query OK, 1 row affected (0.001 sec)

MariaDB [korean]> insert into city value('Busan',051,5000000,'BLK');
Query OK, 1 row affected (0.001 sec)
```

확인

```
MariaDB [korean]> select * from city;
+-----+-----+-----+-----+
| city_name | phone_no | people  | color |
+-----+-----+-----+-----+
| Seoul     | 2        | 10000000 | BLU   |
| Incheon   | 31       | 3000000  | WHT   |
| Busan     | 51       | 5000000  | BLK   |
| Daegu     | 52       | 4000000  | RED   |
| Gwangju   | 63       | 2000000  | YLW   |
+-----+-----+-----+-----+
5 rows in set (0.103 sec)
```

update : 데이터 수정

update 테이블명 set 필드명=필드값 [where 조건식];

```
MariaDB [korean]> update city set phone_no=053 where city_name='Daegu';
Query OK, 1 row affected (1.812 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [korean]> update city set phone_no=062 where city_name='Gwangju';
Query OK, 1 row affected (0.001 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [korean]> select * from city;
+-----+-----+-----+-----+
| city_name | phone_no | people  | color |
+-----+-----+-----+-----+
| Seoul    | 2        | 10000000 | BLU   |
| Incheon  | 31       | 3000000  | WHT   |
| Busan    | 51       | 5000000  | BLK   |
| Daegu    | 53       | 4000000  | RED   |
| Gwangju  | 62       | 2000000  | YLW   |
+-----+-----+-----+-----+
5 rows in set (0.000 sec)
```

delete : 레코드 삭제

delete from 테이블명 where 조건식;

```
MariaDB [korean]> delete from city where city_name='Incheon';  
Query OK, 1 row affected (0.034 sec)
```

```
MariaDB [korean]> select * from city;
```

city_name	phone_no	people	color
Seoul	2	10000000	BLU
Busan	51	5000000	BLK
Daegu	53	4000000	RED
Gwangju	62	2000000	YLW

4 rows in set (0.000 sec)

select : 데이터를 추출하기

select * from city;

// *는 모든 것

```
MariaDB [korean]> insert into city value('Incheon',031,3000000,'WHT');  
Query OK, 1 row affected (0.013 sec)
```

```
MariaDB [korean]>  
MariaDB [korean]> select * from city;  
+-----+-----+-----+-----+  
| city_name | phone_no | people  | color |  
+-----+-----+-----+-----+  
| Seoul     |         2 | 10000000 | BLU   |  
| Incheon   |        31 | 3000000  | WHT   |  
| Busan     |        51 | 5000000  | BLK   |  
| Daegu     |        53 | 4000000  | RED   |  
| Gwangju   |        62 | 2000000  | YLW   |  
+-----+-----+-----+-----+  
5 rows in set (0.000 sec)
```

select city_name, people from city;

```
MariaDB [korean]> select city_name, people from city;  
+-----+-----+  
| city_name | people |  
+-----+-----+  
| Seoul     | 10000000 |  
| Incheon   | 3000000  |  
| Busan     | 5000000  |  
| Daegu     | 4000000  |  
| Gwangju   | 2000000  |  
+-----+-----+  
5 rows in set (0.000 sec)
```

where 조건식 사용하기

인구가 300만 보다 큰 도시를 추출하기 (숫자는 ' '를 사용하지 않음)
select * from city where people > 3000000;

```
MariaDB [korean]> select * from city where people > 3000000;
+-----+-----+-----+-----+
| city_name | phone_no | people  | color |
+-----+-----+-----+-----+
| Seoul     |         2 | 10000000 | BLU   |
| Busan     |        51 | 5000000  | BLK   |
| Daegu     |        53 | 4000000  | RED   |
+-----+-----+-----+-----+
3 rows in set (0.000 sec)
```

컬러가 'red'인 도시를 추출하기 (red는 문자열이므로 ' '를 사용함)
select * from city where color = 'red';

```
MariaDB [korean]> select * from city where color='red';
+-----+-----+-----+-----+
| city_name | phone_no | people  | color |
+-----+-----+-----+-----+
| Daegu     |        53 | 4000000 | RED   |
+-----+-----+-----+-----+
1 row in set (0.000 sec)
```


오름차순 정렬과 내림차순 정렬

select * from city order by people asc;

```
MariaDB [korean]> select * from city order by people asc;
+-----+-----+-----+-----+
| city_name | phone_no | people  | color |
+-----+-----+-----+-----+
| Gwangju   |      62 | 2000000 | YLW   |
| Incheon   |      31 | 3000000 | WHT   |
| Daegu      |      53 | 4000000 | RED   |
| Busan      |      51 | 5000000 | BLK   |
| Seoul      |       2 | 10000000 | BLU   |
+-----+-----+-----+-----+
5 rows in set (0.001 sec)
```

order by는 정렬

asc는 오름차순

desc는 내림차순

select * from city order by people desc;

```
MariaDB [korean]> select * from city order by people desc;
+-----+-----+-----+-----+
| city_name | phone_no | people  | color |
+-----+-----+-----+-----+
| Seoul      |       2 | 10000000 | BLU   |
| Busan      |      51 | 5000000 | BLK   |
| Daegu      |      53 | 4000000 | RED   |
| Incheon     |      31 | 3000000 | WHT   |
| Gwangju     |      62 | 2000000 | YLW   |
+-----+-----+-----+-----+
5 rows in set (0.000 sec)
```

alter : 테이블을 변경할 때

컬럼을 추가

```
ALTER TABLE [table_name] ADD [col_name] [column_type];
```

```
alter table city add 한글명 varchar(20) not null;
```

```
MariaDB [korean]> alter table city add 한글명 varchar(20) not null;
Query OK, 0 rows affected (0.031 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [korean]> desc city;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| city_name  | varchar(10)   | YES  |     | NULL    |       |
| phone_no   | int(3)        | NO   | PRI | NULL    |       |
| people     | bigint(20)    | YES  |     | NULL    |       |
| color      | char(3)       | YES  |     | NULL    |       |
| 한글명     | varchar(20)   | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.014 sec)
```

alter : 테이블을 변경할 때

컬럼을 제거

ALTER TABLE [table_name] DROP [col_name];

alter table city drop color;

```
MariaDB [korean]> alter table city drop color;  
Query OK, 0 rows affected (0.022 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

```
MariaDB [korean]> desc city;
```

Field	Type	Null	Key	Default	Extra
city_name	varchar(10)	YES		NULL	
phone_no	int(3)	NO	PRI	NULL	
people	bigint(20)	YES		NULL	
한글명	varchar(20)	NO		NULL	

4 rows in set (0.001 sec)

alter로 컬럼 이름 바꾸기

alter table [table_name] change [old_name] [new_name] [column_type];

alter table city change 한글명 city_ko varchar(10);

```
MariaDB [korean]> alter table city change 한글명 city_ko varchar(10);
Query OK, 5 rows affected (0.018 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [korean]>
MariaDB [korean]> desc city;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| city_name  | varchar(10)   | YES  |     | NULL    |       |
| phone_no   | int(3)        | NO   | PRI | NULL    |       |
| people     | bigint(20)    | YES  |     | NULL    |       |
| city_ko    | varchar(10)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)
```

alter로 속성 변경

원래 phone_no의 속성이 int(3) 이었는데, varchar(3)으로 변경

```
alter table city change phone_no phone_no varchar(3);
```

```
MariaDB [korean]> desc city;
```

Field	Type	Null	Key	Default	Extra
city_name	varchar(10)	YES		NULL	
phone_no	int(3)	NO	PRI	NULL	
people	bigint(20)	YES		NULL	
city_ko	varchar(10)	YES		NULL	

```
4 rows in set (0.001 sec)
```

```
MariaDB [korean]>
MariaDB [korean]>
MariaDB [korean]> alter table city change phone_no phone_no varchar(3);
Query OK, 5 rows affected (0.013 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
MariaDB [korean]> select * from city;
```

city_name	phone_no	people	city_ko
Seoul	2	10000000	
Incheon	31	3000000	
Busan	51	5000000	
Daegu	53	4000000	
Gwangju	62	2000000	

update문으로 입력값 수정

update city set phone_no='02' where city_name='Seoul';

```
MariaDB [korean]> select * from city;
```

city_name	phone_no	people	city_ko
Seoul	2	10000000	
Incheon	31	30000000	
Busan	51	50000000	
Daegu	53	40000000	
Gwangju	62	20000000	

5 rows in set (0.001 sec)

int였을 때의 값

```
MariaDB [korean]> update city set phone_no='02' where city_name='Seoul';
```

```
Query OK, 1 row affected (0.017 sec)
```

```
Rows matched: 1 Changed: 1 Warnings: 0
```

```
MariaDB [korean]> select * from city;
```

city_name	phone_no	people	city_ko
Seoul	02	10000000	
Incheon	31	30000000	
Busan	51	50000000	
Daegu	53	40000000	
Gwangju	62	20000000	

5 rows in set (0.000 sec)

varchar였을 때의 값

update문으로 한글 필드 추가 입력

```
update city set city_ko='서울' where city_name='Seoul';  
update city set city_ko='인천' where city_name='Incheon';  
update city set city_ko='부산' where city_name='Busan';  
update city set city_ko='대구' where city_name='Daegu';  
update city set city_ko='광주' where city_name='Gwangju';
```

```
select * from city;
```

```
MariaDB [korean]> select * from city;  
+-----+-----+-----+-----+  
| city_name | phone_no | people  | city_ko |  
+-----+-----+-----+-----+  
| Seoul     | 02       | 10000000 | 서울    |  
| Incheon   | 31       | 3000000  | 인천    |  
| Busan     | 51       | 5000000  | 부산    |  
| Daegu     | 53       | 4000000  | 대구    |  
| Gwangju   | 62       | 2000000  | 광주    |  
+-----+-----+-----+-----+  
5 rows in set (0.000 sec)
```


or 조건과 and 조건

select * from city where city_ko='서울' or city_ko='부산';

```
MariaDB [korean]> select * from city where city_ko='서울' or city_ko='부산';
+-----+-----+-----+-----+
| city_name | phone_no | people  | city_ko |
+-----+-----+-----+-----+
| Seoul     | 02       | 10000000 | 서울    |
| Busan     | 51       | 5000000  | 부산    |
+-----+-----+-----+-----+
2 rows in set (0.000 sec)
```

select * from city where people<7000000 and people>3000000;

```
MariaDB [korean]> select * from city where people<7000000 and people>3000000;
+-----+-----+-----+-----+
| city_name | phone_no | people  | city_ko |
+-----+-----+-----+-----+
| Busan     | 51       | 5000000 | 부산    |
| Daegu     | 53       | 4000000 | 대구    |
+-----+-----+-----+-----+
2 rows in set (0.000 sec)
```

패턴 검색하기

5글자인 도시를 검색하기

```
SELECT * FROM city WHERE city_name LIKE "_____";
```

```
MariaDB [korean]> SELECT * FROM city WHERE city_name LIKE "_____";
+-----+-----+-----+-----+
| city_name | phone_no | people | city_ko |
+-----+-----+-----+-----+
| Seoul    | 02       | 10000000 | 서울    |
| Busan    | 51       | 50000000 | 부산    |
| Daegu    | 53       | 40000000 | 대구    |
+-----+-----+-----+-----+
3 rows in set (0.000 sec)
```

%를 사용해서 검색해보기

```
select * from city where city_name like "B%";  
select * from city where city_name like "%u";  
select * from city where city_name like "%o%";
```

```
MariaDB [korean]> select * from city where city_name like "B%";
```

```
+-----+-----+-----+-----+  
| city_name | phone_no | people | city_ko |  
+-----+-----+-----+-----+  
| Busan     | 51       | 5000000 | 부산     |  
+-----+-----+-----+-----+  
1 row in set (0.000 sec)
```

```
MariaDB [korean]> select * from city where city_name like "%u";
```

```
+-----+-----+-----+-----+  
| city_name | phone_no | people | city_ko |  
+-----+-----+-----+-----+  
| Daegu     | 53       | 4000000 | 대구     |  
| Gwangju   | 62       | 2000000 | 광주     |  
+-----+-----+-----+-----+  
2 rows in set (0.000 sec)
```

```
MariaDB [korean]> select * from city where city_name like "%o%";
```

```
+-----+-----+-----+-----+  
| city_name | phone_no | people | city_ko |  
+-----+-----+-----+-----+  
| Seoul     | 02       | 10000000 | 서울     |  
| Incheon   | 31       | 3000000 | 인천     |  
+-----+-----+-----+-----+  
2 rows in set (0.000 sec)
```

count(*)로 개수 세기

```
select count(*) from city;
```

```
select count(*) from city where people>3000000;
```

```
MariaDB [korean]> select count(*) from city;
+-----+
| count(*) |
+-----+
|         5 |
+-----+
1 row in set (0.000 sec)

MariaDB [korean]> select count(*) from city where people>3000000;
+-----+
| count(*) |
+-----+
|         3 |
+-----+
1 row in set (0.000 sec)
```

union으로 두개의 select문 연결하기

인구 300만 초과인 도시의 개수를 세고, 그 도시의 이름을 출력하시오.

```
select count(*) from city where people>3000000 union select city_ko  
from city where people>3000000;
```

```
MariaDB [korean]> select count(*) from city where people>3000000 union select  
city_ko from city where people>3000000;  
+-----+  
| count(*) |  
+-----+  
| 3        |  
| 서울     |  
| 부산     |  
| 대구     |  
+-----+  
4 rows in set (0.000 sec)
```

주의) union으로 select문을 연결할 때에는 앞에서 사용했던 select문의 요청 개수와 뒤의 select문의 요청 개수가 일치해야 한다.

QnA

본 강의 자료는 K-DT의 강의를 위해 제작되었습니다.
저자의 서면 허락없이 배포를 금합니다. 저자 : 소영재 jdrsecure@gmail.com
Copyright© 2023 소영재. All rights reserved