

MySQL 02

2023.06

CentOS 9에서 Mariadb-server 설치 및 실행

```
$ sudo dnf update
```

```
$ sudo dnf install mariadb-server
```

```
$ sudo systemctl start mariadb // 서비스 시작
```

```
$ sudo systemctl enable mariadb // 재부팅후에도 서비스 실행
```

```
$ sudo mysql -u root -p
```

(초기에는 비밀번호 없음)

참고) 원격에서 접속하려면, 3306포트로 접속할 수 있도록 방화벽 허용 필수

```
$ sudo firewall-cmd --permanent --add-service=mysql
```

```
$ sudo firewall-cmd --reload
```

원격 접속을 위한 사용자 생성 및 권한 부여

```
[j@localhost ~]$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 5
Server version: 10.5.16-MariaDB MariaDB Server

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)]> create user ive@'%' identified by 'cyber12#$';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> grant all privileges on *.* to ive@'%';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> 
```

MariaDB [(none)]> create user ive@'%' identified by 'cyber12#\$';
// %는 와일드 문자로서, 전세계 어디서나 접속가능하도록 설정 (위험, 주의)

MariaDB [(none)]> grant all privileges on *.* to ive@'%';
// *.*는 모든 DB의 모든 Table에 대해 모든 권한을 부여한다는 의미

아래와 같이 만들어 봅시다.

데이터베이스 이름 : petshop

테이블명 : pet

name	owner	species	sex	birth	death

데이터베이스 만들기

데이터베이스 만들기

```
create database petshop;
```

데이터베이스 사용 선언

```
use petshop;
```

```
// 어떤 database에서 작업을 할 것인지 지정해주어야 함
```

테이블 만들기

pet이라는 테이블을 만들어봅시다.

```
CREATE TABLE pet(  
  name VARCHAR(20) NOT NULL PRIMARY KEY,  
  owner VARCHAR(20),  
  species VARCHAR(20),  
  sex VARCHAR(1),  
  birth DATE,  
  death DATE  
);
```

혹시 잘못 만들었다면

```
DROP TABLE pet;
```

위와 같이하면 테이블이 지워집니다.
지우고 다시 만들면 됩니다.

테이블에 내용을 넣기

insert문으로 데이터를 넣어 봅시다.

```
insert into 테이블명 values(입력할 데이터 전체);
```

```
INSERT INTO pet VALUES('Puffball','Diane','hamster','f','2019-03-30',NULL);  
INSERT INTO pet VALUES('Fluffy','Harold','cat','f','2013-02-04',NULL);  
INSERT INTO pet VALUES('Claws','Gwen','cat','m','2014-03-17',NULL);  
INSERT INTO pet VALUES('Buffy','Harold','dog','f','2019-05-13',NULL);  
INSERT INTO pet VALUES('Fang','Benny','dog','m','2010-08-27',NULL);  
INSERT INTO pet VALUES('Bowser','Joey','dog','m','2009-08-31','2015-07-29');  
INSERT INTO pet VALUES('Chirpy','Gwen','bird','f','2018-09-11',NULL);  
INSERT INTO pet VALUES('Whistler','Gwen','bird',Null,'2017-12-09',NULL);  
INSERT INTO pet VALUES('Slim','Benny','snake','m','2016-04-29',NULL);
```

잘못 입력한 데이터 지우기

데이터를 잘못 입력했다면 delete문으로 한줄(row) 삭제하기

```
delete from 테이블명 where name='이름';
```

혹시, Fang이라는 동물에 대한 정보를 잘못 입력해서 지우려고 한다면,

```
delete from pet where name='Fang';
```

잘 지워졌는지 확인하려면,

```
SELECT * FROM pet;
```


데이터 수정하기

행(row) 전체를 지우는 것이 아니라,
내용을 수정하려면 update문을 사용합니다.

```
update 테이블명 set 컬럼명='바꿀값' where name='이름';
```

Slim의 death를 기록하려면,

```
update pet set death='2022-01-26' where name='Slim';
```

데이터를 조회하기

```
select * from pet;  
// 보고싶은 내용은 select 뒤에 컬럼명을 씁니다.  
// from 뒤에는 테이블명을 씁니다.
```

```
select name,birth from pet;  
// 이름과 생년월일만 조회
```

```
select name,birth from pet order by birth asc;  
// 생년월일에 따라 오름차순(asc)으로 정렬
```

```
select name,birth from pet order by birth desc;  
// 생년월일에 따라 내림차순(desc)으로 정렬
```

데이터를 수정하고 중복을 제거하기

Bowser의 주인이 Joey가 아니라 Diane이었으므로 수정합니다.
`update pet set owner='Diane' where name='Bowser';`

전체 테이블 내용을 확인합니다.
`select * from pet;`

주인의 이름을 중복제거 하고 보여줌
`select distinct owner from pet;`

AND와 OR 사용하기

2018년 이후에 태어난 동물 검색

```
select * from pet where birth > '2018-01-01';
```

AND: (species='dog')이면서 (sex='f')인 동물 검색

// 로그인할 때도 ID와 PW를 확인

```
select * from pet where species='dog' and sex='f';
```

ex) 로그인 과정

```
select * from member where id='      ' and pw='      ';
```

// 둘중에 하나라도 틀리면 로그인 안됨

OR : species='snake')이거나 (species='bird')인 동물 검색

정규표현식(PCRE)을 활용한 검색

PCRE (Perl Compatible Regular Expressions, 정규표현식) : 패턴 검색

```
SELECT * FROM pet WHERE name regexp "ffy";  
// 정규표현식은 ffy가 들어가는 모든 문자열을 검색
```

```
SELECT * FROM pet WHERE name like "%ffy";  
// like로 검색할 때에는 %를 앞 또는 뒤, 앞뒤에 붙여서 사용
```

내림차순, 오름차순 정렬하기

```
SELECT owner, COUNT(*) FROM pet GROUP BY owner order by owner desc;
```

```
SELECT species, COUNT(*) FROM pet GROUP BY species order by species asc;
```

```
SELECT distinct owner AS customer FROM pet;  
// owner 컬럼명을 customer(별칭)로 바꾸어서 보여주기
```

묶기와 조건 적용하기

group by와 having 조건문

성별로 묶어서 개수를 세어보세요.

```
select sex,count(*) from pet group by sex having count(*) ;
```

```
select name,sex from pet group by sex having 1=1;
```

// having 1=1은 모두 참이 되는 조건, name은 첫줄만 출력

```
select sex,count(*) from pet group by sex having 1=1;
```

// 성별 개수로 표시

```
select sex,count(*) from pet group by sex;
```

// having 1=1은 사족(蛇足)이므로 제거 가능

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