데이터 베이스

데이터 베이스 생성

문자 인코딩을 기본값으로 지정하여 데이터베이스 생성 가능

create database testdb

character set = ‘utf8’

collate = ‘utf8\_general\_ci’;

데이터베이스 확인

Show datebases 를 통하여 작성된 데이터베이스 확인

Show datebases;

데이터베이스 선택

Use database 명령어는 데이터베이스를 선택하는데 사용

Mariadb 데이터 서버에 여러 데이터베이스가 있다고 가정하면 특정 데이터베이스 선택

Use testdb;

테이블 생성

Create table mem(

Num int not null auto\_increment,

Id char(20) not null,

Name char(20) not null,

Age int,

Primary key(num)

);

테이블 구조 보기

Desc mem(=테이블명);

데이터베이스 테이블 필드 관리

* 새로운 필드 추가하기

Alter table “테이블명” add “새로운 필드명” “필드 데이터형” [first 또는 after “필드명”];

Mem 테이블에 email 필드를 char(30) 데이터형으로 추가하려면 다음과 같이 입력

Alter table mem add email char(30);

문제) password 필드를 pass 다음에 추가하려면?

Alter table mem add password char(20) after pass;

* 필드 삭제

테이블에서 특정 필드를 삭제하는 명령

Alter table “테이블명” drop “삭제할 필드명”, “삭제할 필드명2”;

Mem 테이블에서 email 필드를 삭제한 뒤 확인하는 명령

Alter table mem drop email;

Desc mem;

* 필드 데이터형 수정하기

테이블에서 특정 필드의 데이터형을 수정하는 명령

Alter table “테이블명” modify “필드명” “수정할 데이터형”;

Mem 테이블에서 phone 필드의 데이터형을 char(20) 에서 int 로 변경한 뒤 확인하는 명령

Alter table mem modify phone int;

Desc mem;

* 데이터베이스 테이블 이름관리

이미 존재하는 DB 테이블의 이름을 변경하는 명령

Alter table “수정할 테이블명” rename “새로운 테이블명”;

Mem 테이블의 이름을 mem2로 변경

Alter table mem rename mem2;

* 테이블 삭제

Drop table 테이블명

Mem2 테이블을 삭제하시오

Drop table mem2;

특정 컬럼에 auto\_increment 추가

Alter table “테이블명” modify “컬럼명” int not null auto\_increment;

Alter table “테이블명” modify “컬럼명” int not null auto\_increment primary key(num);

Num 필드에 auto\_increment

1. create table employee(

-> num int not null,

-> id char(20) not null,

-> name char(10) not null,

-> gender char(1) not null,

-> age int not null,

-> department char(20) not null,

-> hp char(20) not null,

-> address char(100) not null\

-> );

2. desc employee;

3. alter table employee add email char(30);

4. alter table employee rename member;

5. alter table member drop age;

6. drop table member;

데이터베이스2

Friend 테이블 생성

create table friend (

num int not null auto\_increment,

name char(20) not null,

tel char(20) not null,

address char(80),

primary key(num)

);

-테이블에 데이터 삽입

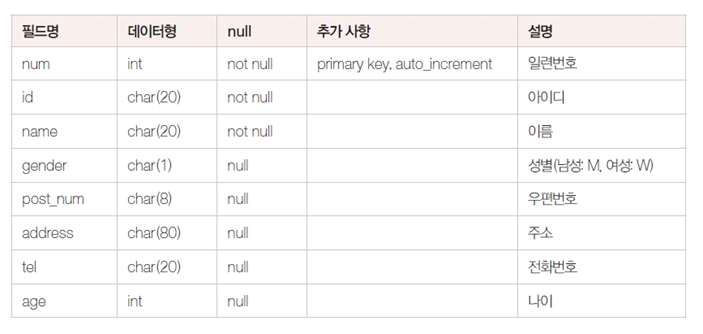
Insert into 테이블명(필드명1, 필드명2…) value (필드값1, 필드값2…);

이름이 테이블명인 DB테이블의 (필드명1, 필드명2…) 필드에 각각 (필드값1, 필드값2…) 를 삽입하는 명령

Insert into friend (name, tel, address) values (‘송예진’,’031-129-4567’,’경기도 용인시 수지구 정평로 30’);

테이블 설계

테이블 명 : member



Member 테이블의 전체 필드를 검색

Select \* from member;

Member 테이블의 특정 필드를 검색

Select 필드명1, 필드명2… from 테이블명;

Select name, tel, address from member;

특정 조건에 맞는 레코드 검색

Select 필드명1, 필드명2… from 테이블명 where 조건;

select name, tel, gender, age from member where gender = “M”&&age>=50;

select name, gender, address, age from member where 19<age && 30>age;

select name, tel, age, gender from member where (29<age&&40>age) || (49<age&&60>age) && gender = “M”;

select num, name, tel, address, post\_num, age, gender from member where name = “이상훈”;

select name, tel from member where name like “박%”;

-박으로 시작하는 모든 문자열을 의미

Select num, name, address, gender from member where address like “서울%”;

Select \* from member where address like “%은평%”;

-은평이 포함된 모든 문자열을 의미

레코드의 정렬

‘테이블명’ 이라는 db테이블에서 ‘필드명’을 기준으로 오름차순 정렬된 필드명1, 필드명2… 을 검색하는 명령

Select 필드명1, 필드명2… from 테이블명 order by 필드명;

Select num, name, address, gender from member order by name;

Select name, age, address, tel, gender from member where address like “서울%” order by age;

Select name, age, address, tel, gender from member where address like “서울%” order by age desc;

레코드의 수정

‘테이블명’ 이라는 db테이블에서 조건식을 만족하는 레코드의 ‘필드명’ 값을 ‘필드값’으로 변경하는 명령

Update 테이블명 set 필드명 = 필드값 where 조건식;

Update member set tel = ‘123-4567’ where name = ‘박철호’;

Update member set age=’37’ where name=’박혜빈’;

레코드의 삭제

‘테이블명’ 이라는 db 테이블에서 조건식을 만족하는 레코드를 삭제하는 명령

Delete from 테이블명 where 조건식;

Delete from member where name=‘김문수’;

Insert into member (id, name, gender, post\_num, address, tel, age) values (‘jangsu’, '김장수','M','123-454','서울시 강남구 도곡동','459-4578',55);