

cmd에서 sql 접속

mysql -u root -p

-> 나와도 이어쓰기 가능

\*\*대소문자 모두 동일처리\*\*

DB 목록 보기

show databases;

DB 생성

create database [DB\_name];

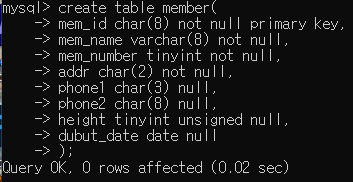
DB 삭제

drop database [DB\_name];

DB 사용(선택)

use [DB\_name];

table 생성



CREATE TABLE member(  
mem\_id CHAR(8) NOT NULL PRIMARY KEY,  
mem\_name VARCHAR(10) NOT NULL,   
mem\_number TINYINT NOT NULL,   
addr CHAR(2) NOT NULL,  
phone1 CHAR(3) NULL,  
phone2 CHAR(8) NULL,  
height TINYINT UNSIGNED NULL,   
debut\_date DATE NULL  
);

tinyint : 자바에서 byte와 유사, -128 ~ 127

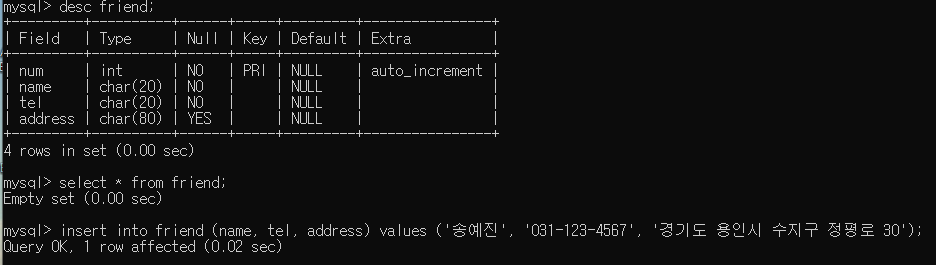
tinyint unsigned : 0~255, 음수x 음수만큼 양수 범위가 늘어남

table구조 보기

desc [table\_name];

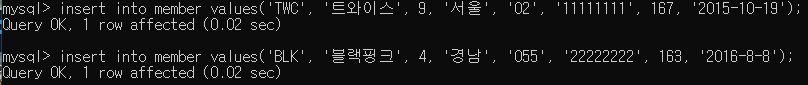
내용 추가

insert into [table\_name] (필드 나열) values (내용 작성);



나열한 필드만큼 내용 작성

insert into [table\_name] values (내용 작성);



모든 필드 수 만큼 내용을 작성

auto\_increment와 같이 직접 입력해주지 않아도 되는 값에 대해서는 null 넣기

table 지우기

drop table [table\_name];

table 내용 보기

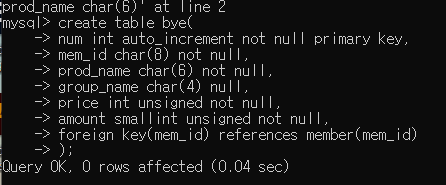
select \* from [table\_name];

field 지정해서 보기

select [field1], [field2] from [table\_name]

table 이름 변경

rename table [기존이름] to [새이름]



맨 뒤에 field 추가

alter table [table\_name] add [field\_name] [field\_type];

field 지우기

alter table [table\_name] drop [field\_name];

위치 지정해서 필드 추가

\*(field\_name0) 뒤에 추가

alter table [table\_name] add [field\_name] [field\_type] after [field\_name0]

field 수정

alter table [table\_name] change [old\_field\_name] [new\_field\_name] [new\_field\_type]

field type 수정

alter table [table\_name] modify [field\_name] [field\_new\_type]

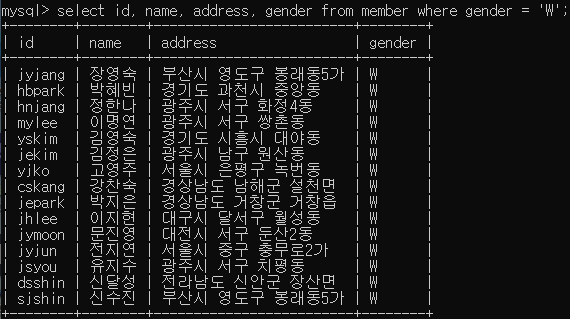
table 삭제

drop table [table\_name];

조건 검색

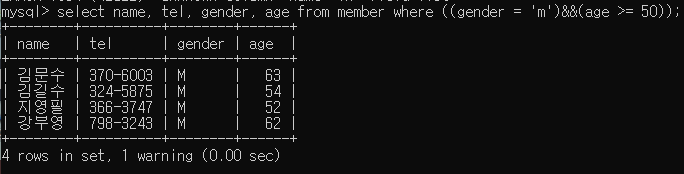
select [field1], [field2] from [table\_name] where [조건]

\*\*여성회원의 아이디, 이름, 주소, 성별 검색

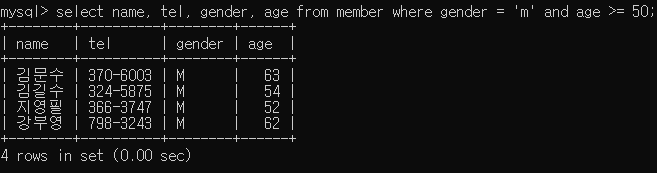


((조건1) && (조건2))

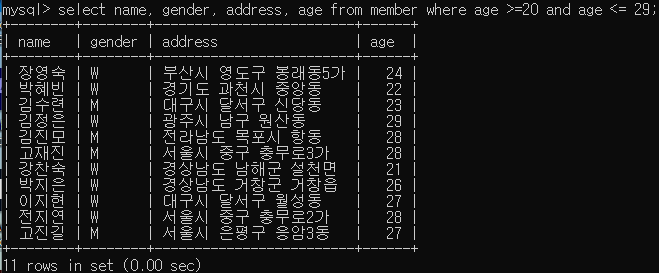
\*\*50세 이상인 남성 회원의 이름, 전화번호, 성별, 나이 검색



조건1 and 조건2

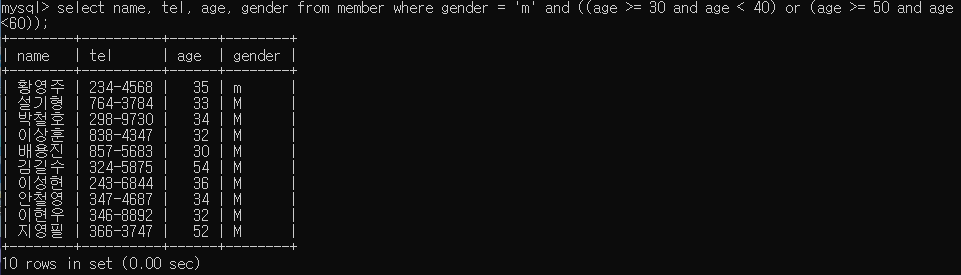


\*\*20대 회원의 이름 성별, 주소, 나이 검색

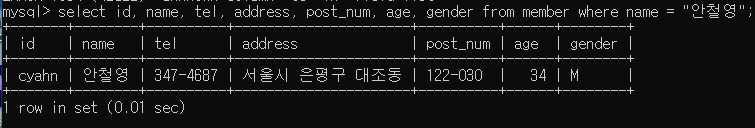


\*\*30대 또는 50대 남자 회원이 이름, 전화번호, 나이, 성별 검색

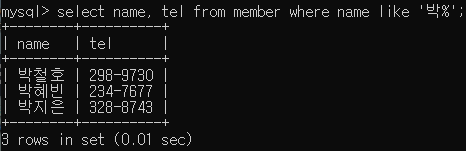
select name, tel, age, gender from member where gender = 'm' and ((age >= 30 and age < 40) or (age >= 50 and age <60));



\*\* “안철영”의 일련번호, 이름, 전화번호, 주소, 우편번호, 나이, 성별 검색



\*\* 박씨 이름과 전화번호 검색

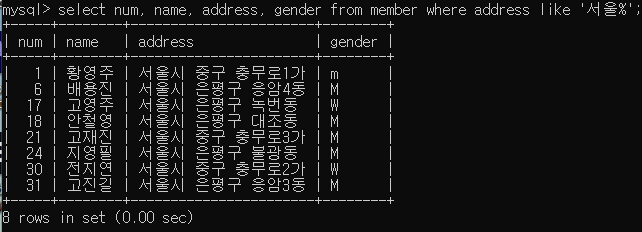


like ‘%A’ : A로 끝나는

like ‘A%’ : A로 시작하는

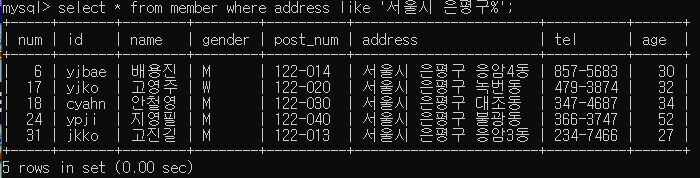
like ‘%A%’ : 중간에 A가 포함된

\*\*서울에 거주하는 회원들의 일련번호, 이름, 주소, 성별 검색

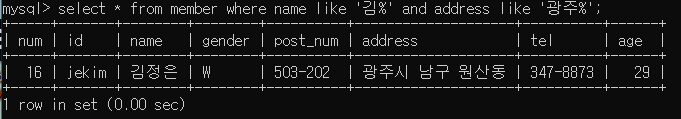


\*\* 은평구에 거주하는 회원의 모든 정보 검색

like ‘%은평구%’



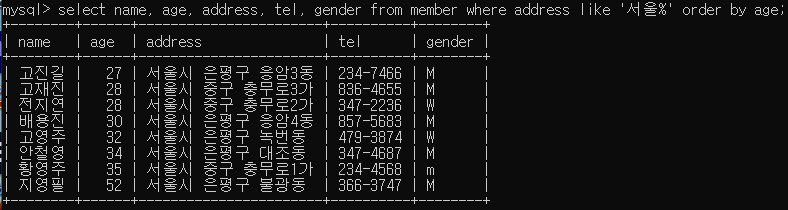
\*\* 광주에 거주하는 김씨 회원들의 모든 정보 검색



레코드 정렬

select [field\_name1], [field\_name2] from [table\_name] order by [기준이 되는 field\_name]

\*\* 서울에 사는 회원을 나이 오름차순으로 정렬한 뒤, 이름, 나이, 주소, 전화번호, 성별 검색



\*\* 부산에 사는 회원을 나이 내림차순으로 정렬한 뒤, 이름, 나이, 주소, 전화번호 검색

select name, age, address, tel from member where address like '부산%' order by age desc;

desc : 내림차순(descending)

레코드 수정

update 테이블명 set 필드명=필드값 where 조건식

\*\*고재진의 전화번호를 123-4567로 변경

update member set tel = ‘123-4567’ where name = “고재진”;

\*\* 정한나의 나이를 37로 변경

update member set age = 37 where name = “정한나”;

레코드 삭제

delete from 테이블명 where 조건식

\*\*김수련의 레코드 삭제

delete from member where name = ‘김수련’;

\*\* 40대 회원의 레코드를 삭제하고 35~53세 회원의 이름, 나이, 성별을 나이순으로 출력

delete from member where age >= 40 and age < 50;

select name, age, gender from member age >= 35 and age <= 53 order by age;

delete from 테이블명;

테이블 전체 레코드 삭제

조건(where)을 생략하면 테이블의 전체 레코드를 삭제할 수 있으니 주의

table 출력

show tables;

열 추가

alter table [table\_name] add column [name] [type];

열 이름 변경

alter table [table\_name] change column [name] [new\_name] [type];

열 타입 변경

alter table [table\_name] modify column [name] [new\_type];

열 삭제

alter table [table\_name] drop column [name];

.sql 파일로 저장

\*\*\*\*CMD에서

mysqldump -uroot -p1234 [table\_name] >[filename.sql]



A as B, A(value)의 제목을 B로 표현하겠다.

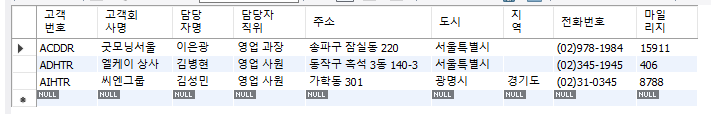




limit : 개수제한

고객 테이블에서 1행부터 시작하여 3개의 고객정보를 검색



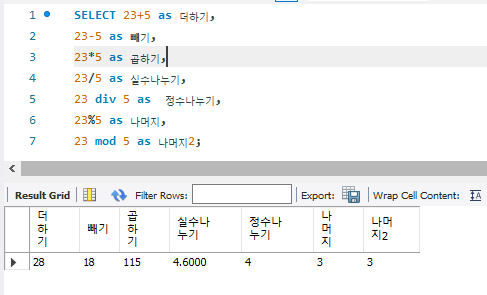


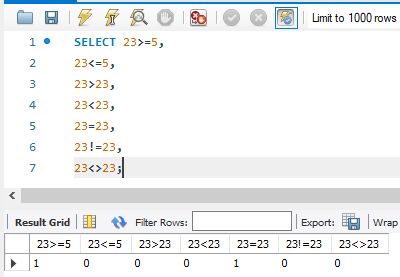


limit n, k; n행부터 k개



distinct 중복데이터 제거





!=, <> 둘 다 “다르다”는 의미를 가짐

지역 값에 들어있지 않는 고객의 정보 검색

select \* from 고객 where 지역 is null;

CSV 파일에서 테이블을 읽어올 때, 값이 들어있지 않는 셀은 “”(빈문자열)

select \* from 고객 where 지역 = “”;

그래서 빈 문자열을 null로 변경

update 고객 set 지역 = null where 지역 = “”; (x : safemode는 전체 테이블에서 업데이트 방지)

* SET SQL\_SAFE\_UPDATES = 0; 을 앞에 선언
* 이후 SET SQL\_SAFE\_UPDATES = 1;로 safemode 복구

Edit> Preferences > SQL Editor > Safe Updates 옵션 설정 후 재부팅 해도 됨.

null에 대한 접근은 = 나 !=는 사용하지 않고, is나 is not을 통해 접근한다

select 고객번호, 담당자명, 담당자직위 from 고객 where 담당자직위 = '영업 과장' or 담당자직위 = '마케팅 과장';

select 고객번호, 담당자명, 담당자직위 from 고객 where 담당자직위 **in** ('영업 과장', '마케팅 과장');

select 담당자명, 마일리지 from 고객 where 100000 <= 마일리지 && 마일리지 <= 200000;

select 담당자명, 마일리지 from 고객 where 마일리지 **between** 100000 and 200000;

SELECT \* FROM 고객 WHERE 도시 LIKE "%광역시" AND (고객번호 LIKE '\_C%' OR 고객번호 LIKE '\_\_C%');

\_C : 두번째 글자가 C

\_\_C : 세번째 글자가 C

