

cmd에서 sql 접속

mysql -u root -p

-> 나와도 이어쓰기 가능

\*\*대소문자 모두 동일처리\*\*

DB 목록 보기

show databases;

DB 생성

create database [DB\_name];

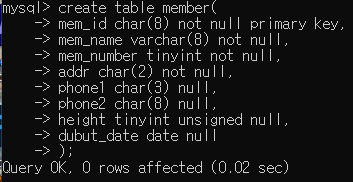
DB 삭제

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

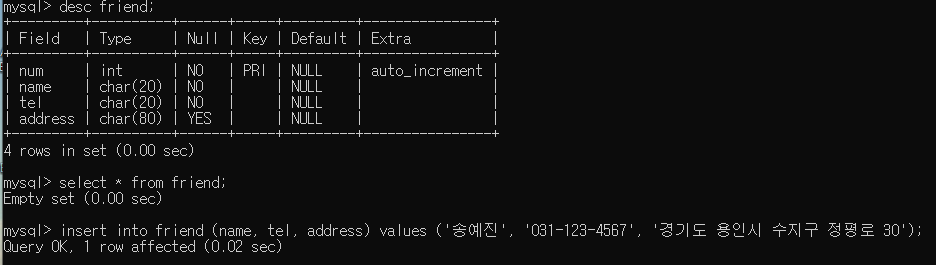
tinyint unsigned : 0~255, 음수x 음수만큼 양수 범위가 늘어남

table구조 보기

desc [table\_name];

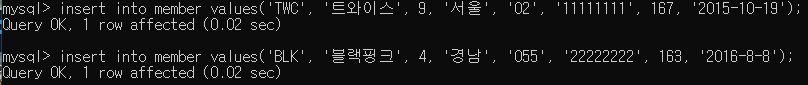
내용 추가

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

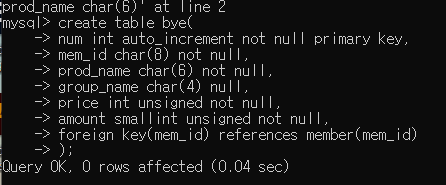
select \* from [table\_name];

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]

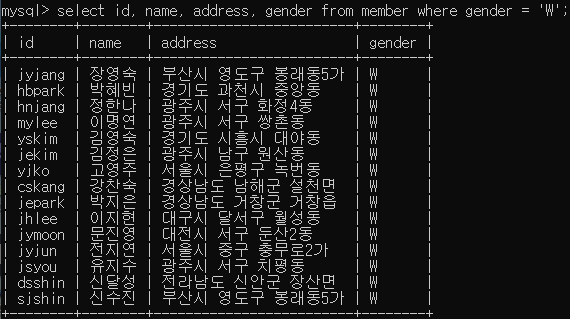
table 삭제

drop table [table\_name];

조건 검색

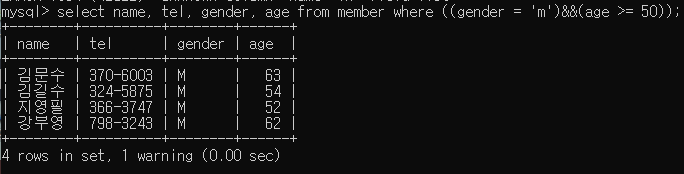
select [field1], [field2] from [table\_name] where [조건]

\*\*여성회원의 아이디, 이름, 주소, 성별 검색

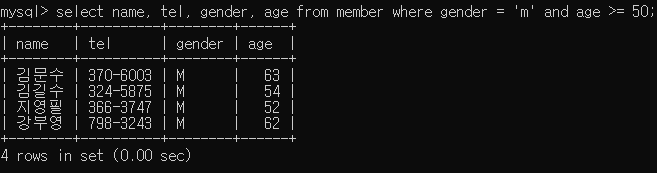


((조건1) && (조건2))

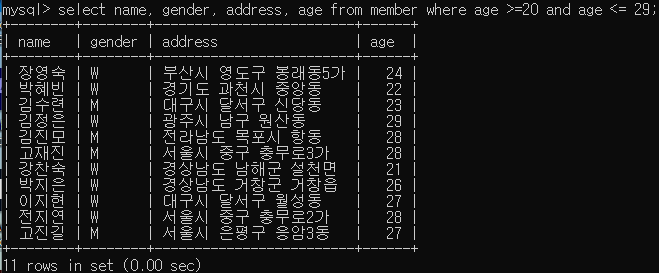
\*\*50세 이상인 남성 회원의 이름, 전화번호, 성별, 나이 검색



조건1 and 조건2

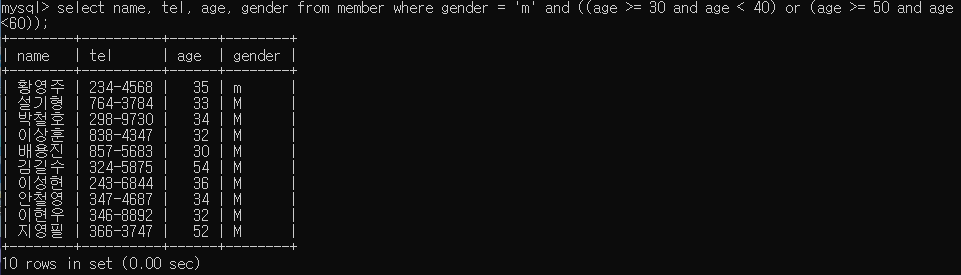


\*\*20대 회원의 이름 성별, 주소, 나이 검색

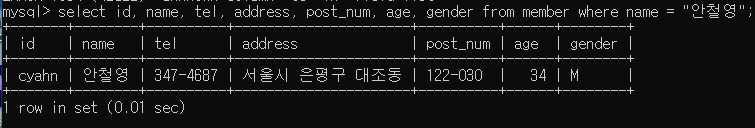


\*\*30대 또는 50대 남자 회원이 이름, 전화번호, 나이, 성별 검색

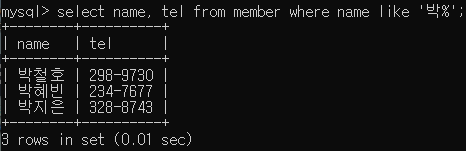
select name, tel, age, gender from member where gender = 'm' and ((age >= 30 and age < 40) or (age >= 50 and age <60));



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



\*\* 박씨 이름과 전화번호 검색

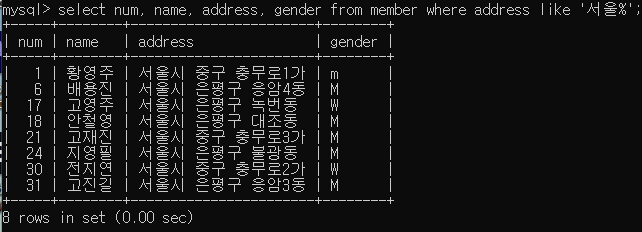


like ‘%A’ : A로 끝나는

like ‘A%’ : A로 시작하는

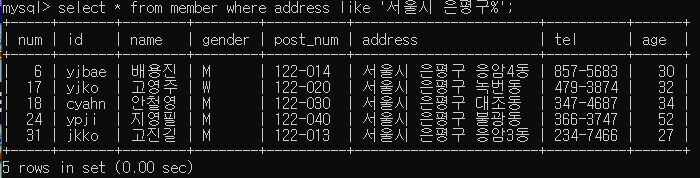
like ‘%A%’ : 중간에 A가 포함된

\*\*서울에 거주하는 회원들의 일련번호, 이름, 주소, 성별 검색

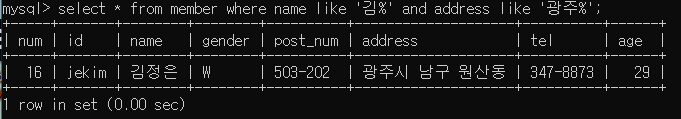


\*\* 은평구에 거주하는 회원의 모든 정보 검색

like ‘%은평구%’



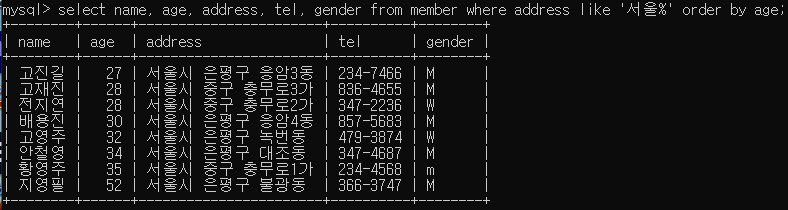
\*\* 광주에 거주하는 김씨 회원들의 모든 정보 검색



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.

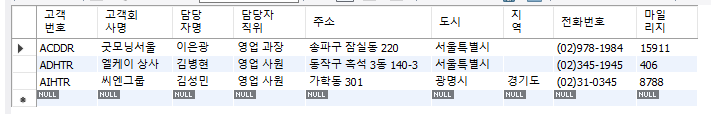




limit : 개수제한

고객 테이블에서 1행부터 시작하여 3개의 고객정보를 검색



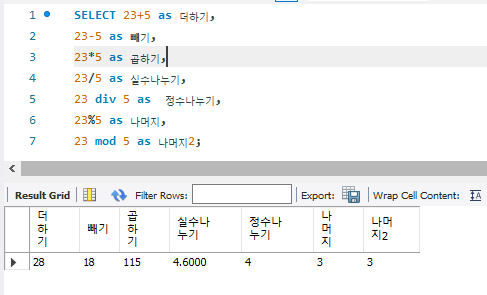


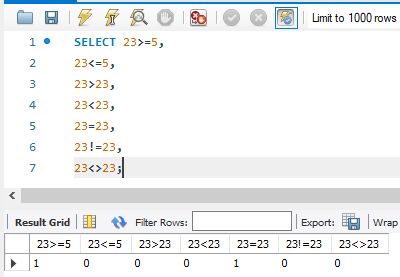


limit n, k; n행부터 k개



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');

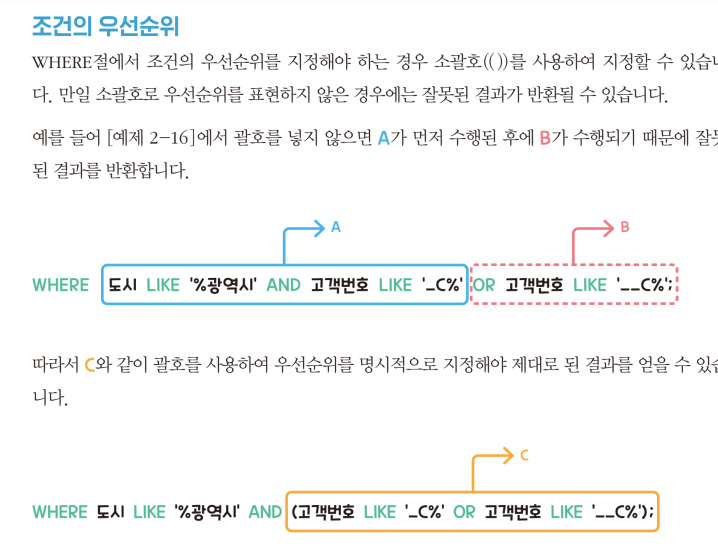
select 담당자명, 마일리지 from 고객 where 100000 <= 마일리지 && 마일리지 <= 200000;

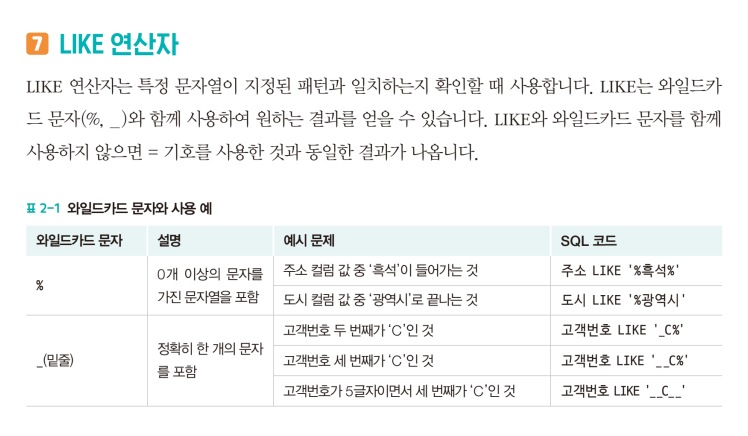
select 담당자명, 마일리지 from 고객 where 마일리지 **between** 100000 and 200000;

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

\_C : 두번째 글자가 C

\_\_C : 세번째 글자가 C





글자의 길이

select char\_length('hello'), // 영문자 개수 반환 5

length('hello'), // 영문자 바이트 수 5

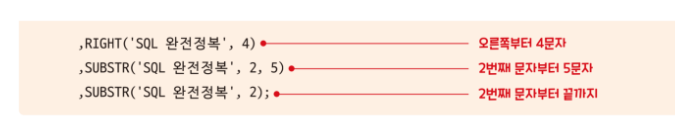
char\_length('안녕'), // 문자 개수2

length('안녕'); // utf-8 바이트 수 6

select concat('dreams', 'come', 'true'), // dreamscometrue

concat\_ws('-','2025','04','10'); // 2025-04-10

SELECT LEFT("SQL 완전정복", 3) -> 왼쪽부터 지정한 길이만큼의 문자열 반환



(문자열, 구분할 문자, 개수)

SELECT SUBSTRING\_INDEX('서울시 동작구 흑석로', ' ', 2)

,SUBSTRING\_INDEX('서울시 동작구 흑석로', ' ', -2);

(문자열, 개수, 채울 문자)

SELECT lpad('SQL', 10, '#'), // 문자열 앞에 전체길이가 될 때까지 문자를 채운다.

rpad('SQL', 5, '\*'); // 문자열 뒤에 전체길이가 딜 때까지 문자를 채운다

글자 지우기(TRIM) -> 글자수(LENGTH)

SELECT LENGTH(LTRIM(' SQL '))

,LENGTH(RTRIM(' SQL '))

,LENGTH(TRIM(' SQL '));

SELECT TRIM(BOTH 'abc' FROM 'abcSQLabcabc') // 양쪽 모든 abc 제거

,TRIM(LEADING 'abc' FROM 'abcSQLabcabc') // 왼쪽 모든 abc 제거

,TRIM(TRAILING 'abc' FROM 'abcSQLabcabc'); // 오른쪽 모든 abc 제거

SELECT repeat('\*', 5); // \*\*\*\*\*

SELECT replace('010.1234.5678', '.', '-'); .을 -로 바꾸기

SELECT CEILING(123.56), // 올림

FLOOR(123.56), // 버림

ROUND(123.56), // 반올림

ROUND(123.56, 1), // 소수 둘째자리 반올림, 첫째자리 표기

TRUNCATE(123.56, 1); // 소수 한자리까지 남김, 둘째자리부터 버림

나머지 구하기

SELECT MOD(203, 4), 203 % 4, 203 MOD 4;

SELECT POWER(2,3), // 2^3

SQRT(16), // 루트 16

RAND(), // 0~1 무작위수

RAND(100), // SEED 설정

round(RAND()\*100); // 0~100

SELECT NOW(), sysdate(), // 현재시간 년-월-일 시:분:초

curdate(), // 현재 날짜만 년-월-일

curtime(); // 현재 시간만 시:분:초

SELECT NOW()

,YEAR(NOW())

,QUARTER(NOW()) // 분기

,MONTH(NOW())

,DAY(NOW())

,HOUR(NOW())

,MINUTE(NOW())

,SECOND(NOW());

SELECT NOW()

,DATEDIFF('2026-12-20', NOW()) // 앞 – 뒤

,DATEDIFF(NOW(), '2026-12-20') // 뒤 – 앞

,timestampdiff(YEAR ,NOW(), '2026-12-20') // 년 차이

,timestampdiff(MONTH ,NOW(), '2026-12-20') // 월 차이

,timestampdiff(DAY ,NOW(), '2026-12-20'); // 일 차이

SELECT NOW()

,adddate(NOW(), 50)

,adddate(now(), interval 50 DAY) // 50일 후

,adddate(now(), interval 50 MONTH) // 50 개월 후

,subdate(now(), interval 50 YEAR); // 50년 전

SELECT NOW()

,LAST\_DAY(NOW()) // 마지막 날자

,dayofyear(NOW()) // 1년중 몇번째 일인지

,DAYOFMONTH(NOW())

,WEEKDAY(now());

형변환

SELECT CAST('1' AS UNSIGNED INTEGER),

CAST('2' AS CHAR(1)),

CONVERT('1', UNSIGNED INTEGER),

CONVERT('2', CHAR(1));

SELECT IF(12500 \* 450 > 5000000, '초과달성', '미달성');

(조건, TRUE, FALSE);

IF

SELECT CASE

WHEN 12500 \* 450 > 5000000 THEN '초과달성'

WHEN 2500 \* 450 > 4000000 THEN '달성'

ELSE '미달성'

END;

select 고객회사명,

concat("\*\*", SUBSTR(고객회사명, 3)) AS 고객회사2,

전화번호,

replace(substr(전화번호, 2), ')', '-') AS 전화번호2

FROM 고객;



SELECT \*,

단가 \* 주문수량 AS 주문금액,

TRUNCATE(단가 \* 주문수량 \* 할인율, -1) AS 할인금액,

단가 \* 주문수량 - TRUNCATE(단가 \* 주문수량 \* 할인율, -1) AS 실주문금액

FROM 주문세부;



SELECT 담당자명

,고객회사명

,도시

,IF(도시 LIKE '%특별시' OR 도시 LIKE '%광역시', '대도시', '도시') AS 도시구분

,마일리지

,CASE WHEN 마일리지 >= 100000 THEN 'VVIP고객'

WHEN 마일리지 >= 10000 THEN 'VIP고객'

ELSE '일반고객'

END AS 마일리지구분

FROM 고객



SELECT 주문번호, 고객번호, 주문일

, YEAR(주문일) AS '주문년도'

, quarter(주문일) AS '주문분기'

, month(주문일) AS '주문월'

, day(주문일) AS '주문일'

, weekday(주문일) AS '주문요일'

, CASE weekday(주문일) WHEN 0 THEN '월요일'

WHEN 1 THEN '화요일'

WHEN 2 THEN '수요일'

WHEN 3 THEN '목요일'

WHEN 4 THEN '금요일'

WHEN 5 THEN '토요일'

ELSE '일요일'

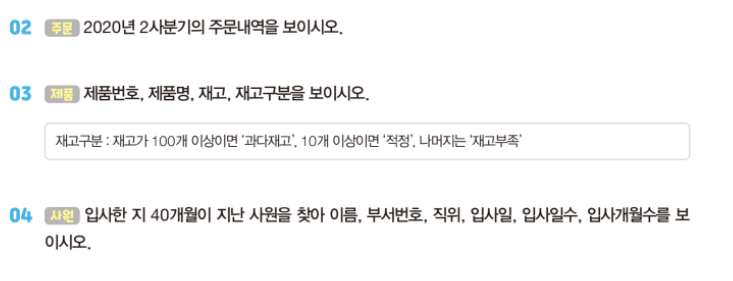
END AS '한글요일'

from 주문;



SELECT \*, datediff(발송일, 요청일) AS 지연일수

from 주문 WHERE datediff(발송일, 요청일) >= 7;



SELECT \* from 주문 WHERE (YEAR(주문일) = 2020) AND (quarter(주문일) = 2);

SELECT 제품번호, 제품명, 재고,

CASE 재고 WHEN 재고 > 100 THEN '과다재고'

WHEN 재고 >= 10 THEN '적정'

ELSE '재고부족'

END AS 재고구분 FROM 제품;

SELECT 이름, 부서번호, 직위, 입사일,

datediff(NOW(), 입사일) AS 입사일수,

TIMESTAMPDIFF(MONTH, 입사일, NOW()) AS 입사개월수

FROM 사원 WHERE TIMESTAMPDIFF(MONTH, 입사일, NOW()) >= 40;

root 진입

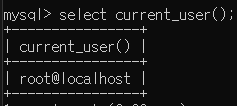
mysql -u root -p

mysql 사용

use mysql

현재 로그인한 사용자

select current\_user();



새로운 user 생성

CREATE USER [name]@localhost IDENTIFIED BY ‘[PASSWORD]’;

특정 DB에 권한주기

GRANT ALL ON [DB\_NAME].\* TO [name]@localhost;

flush privileges; 설정(변경사항) 저장

모든 DB에 대한 권한

GRANT ALL ON \*.\* TO [name]@localhost;

GRANT ALL ON \*.\* TO [name]@%localhost;

SELECT, INSERT 권한만 주기

GRANT SELECT, INSERT ON [DB\_NAME].\* TO [name]@localhost;

특정 DB에 포함된 TABLE에 권한 주기

GRANT ALL ON [DB.TABLE] TO [name]@localhost

ex) GRANT ALL ON test.frient TO [name]@localhost

쓰기, 수정 권한 삭제

REVOKE INSERT, UPDATE ON [DB].\* FROM [name];

전체 권한 삭제

REVOKE ALL ON [DB].\* FROM [name];

사용자 전체 쓰기 권한 삭제

REVOKE INSERT ON \*.\* TO [name]

사용자 삭제

DROP USER [name]@localhost

localhost 와 % 차이

localhost 로컬 컴퓨터(자기 자신)에서만 접속 허용

% 어디서든 접속 허용(외부 ip에서도 => 보안 위험, 방화벽 필수)

CREATE USER [user@192.168.1.10](mailto:user@192.168.1.10) IDENTIFIED BY ‘1234’;

특정 IP에서만 접근 가능하도록

집계 함수의 개념

SELECT [집계함수] FORM [TABLE] WHERE [조건];



EX) 대전에 있는 모든 고객의 마일리지 합

SELECT SUM(마일리지) AS 마일리지합 FROM 고객 WHERE 도시 LIKE "대전%";



SELECT count(\*) // NULL과 상관없이 모든 CELL의 개수를 셈.

, COUNT(고객번호) , COUNT(도시) , COUNT(지역)

FROM 고객;

합, 평균, 최소값, 최대값

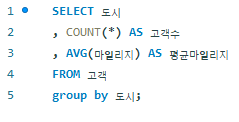
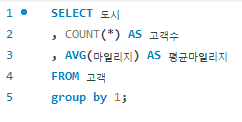
SELECT SUM(마일리지), AVG(마일리지), MIN(마일리지), MAX(마일리지)

FROM 고객;

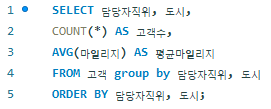
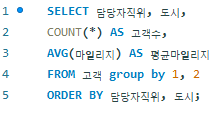
SELECT SUM(마일리지), AVG(마일리지), MIN(마일리지), MAX(마일리지)

FROM 고객 WHERE 도시 = "서울특별시"

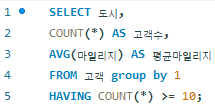
도시 별 고객의 수와 해당 도시 고객들의 평균 마일리지

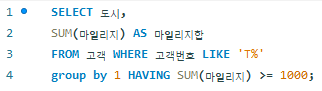
SELECT와 GROUP BY에 동일한 COLUMN 이름 혹은 순번을 넣어줘야함

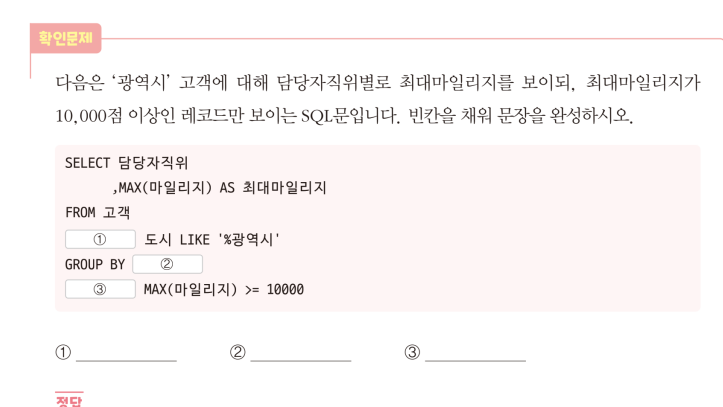
 

예제 4-6  
고객 테이블에서 도시 별 그룹으로 묶어 고객수와 평균 마일리지  
이 중 고객수가 10 이상인 레코드 보기  
GROUP BY에 대한 조건을 걸어줄 때 : HAVING



고객번호가 T로 시작하는 고객에 대해 도시별로 묶어 고객의 마일리지 합 구하고, 마일리지 합이 1000 이상인 레코드 보기





SELECT 담당자직위

,MAX(마일리지) AS 최대마일리지

FROM 고객

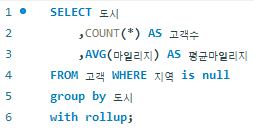
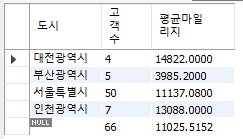
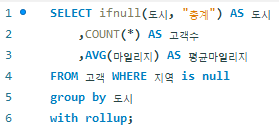
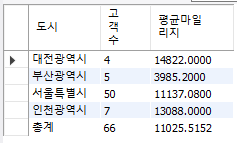
WHERE 도시 LIKE '%광역시'

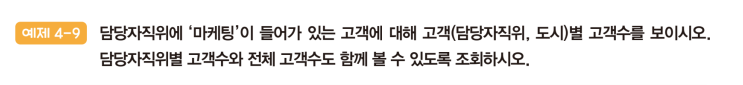
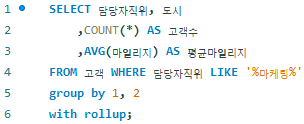
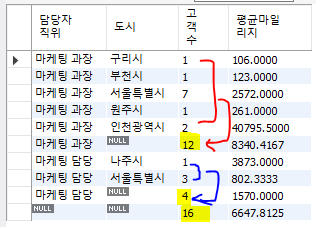
GROUP BY 1

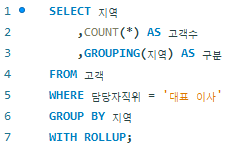
HAVING MAX(마일리지) >= 10000;

WITH ROLLUP

그룹별 소개와 전체 총계 구하기

GROUPING : ROLL UP 결과로 나온 NULL에 대해 1 나머지는 0

GROUP\_CONCAT() 모든 COLUMN 값을 출력(중복 상관 없이 모두)