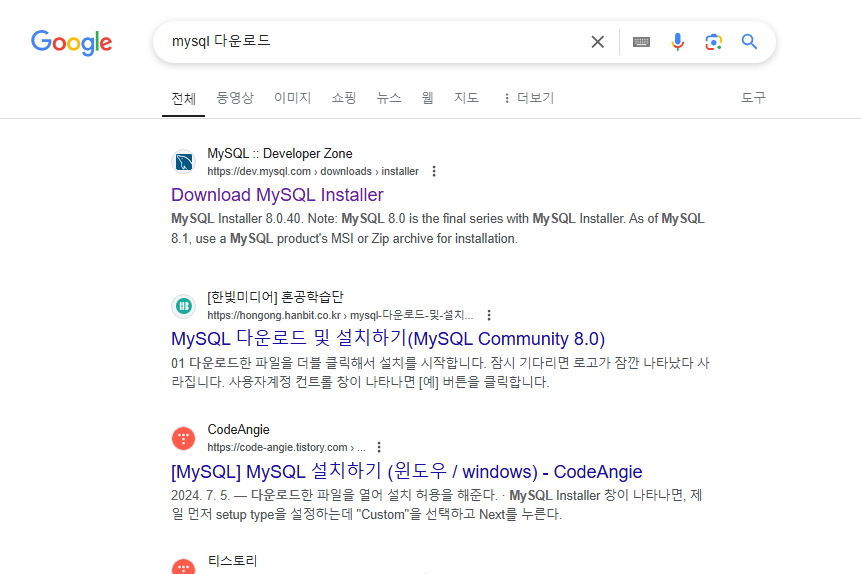
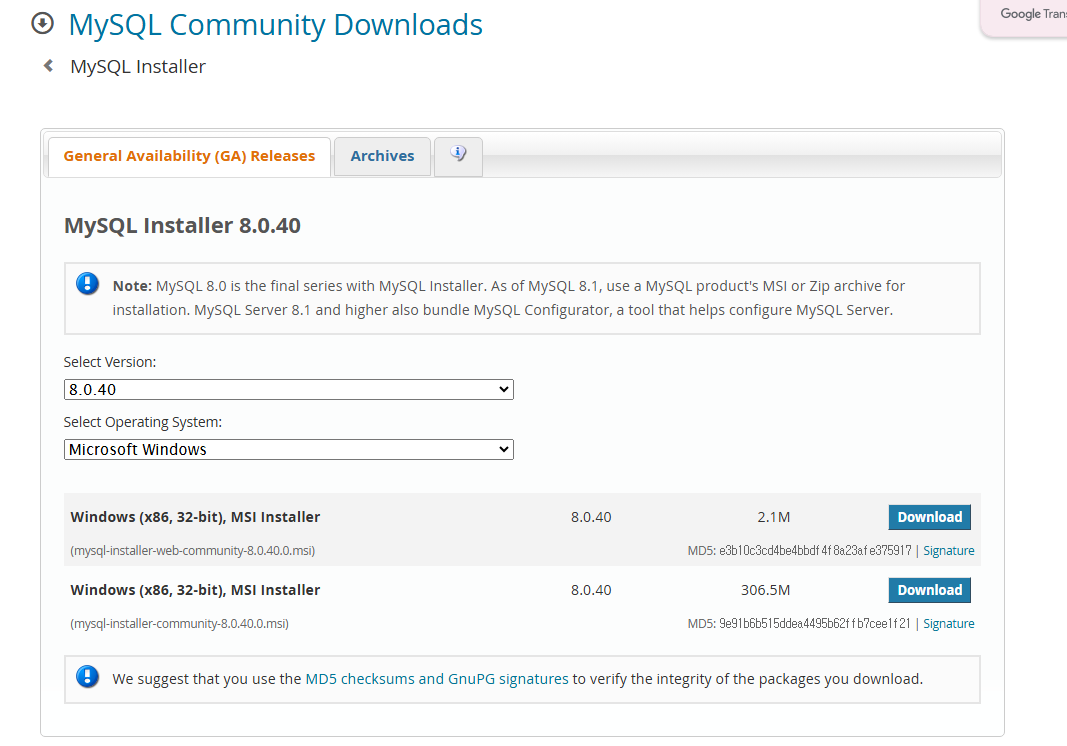
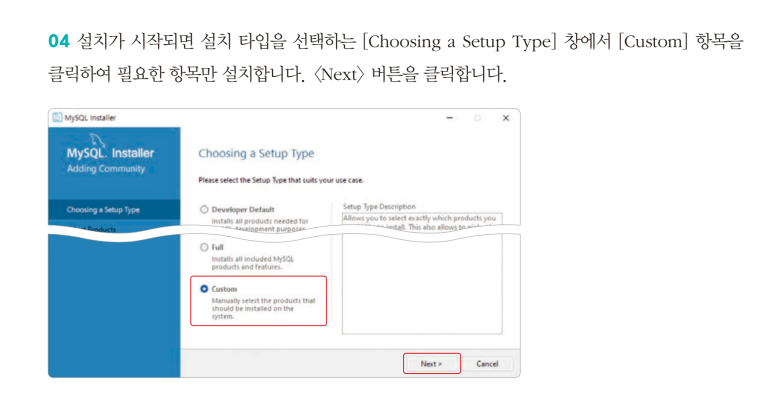
Mysql

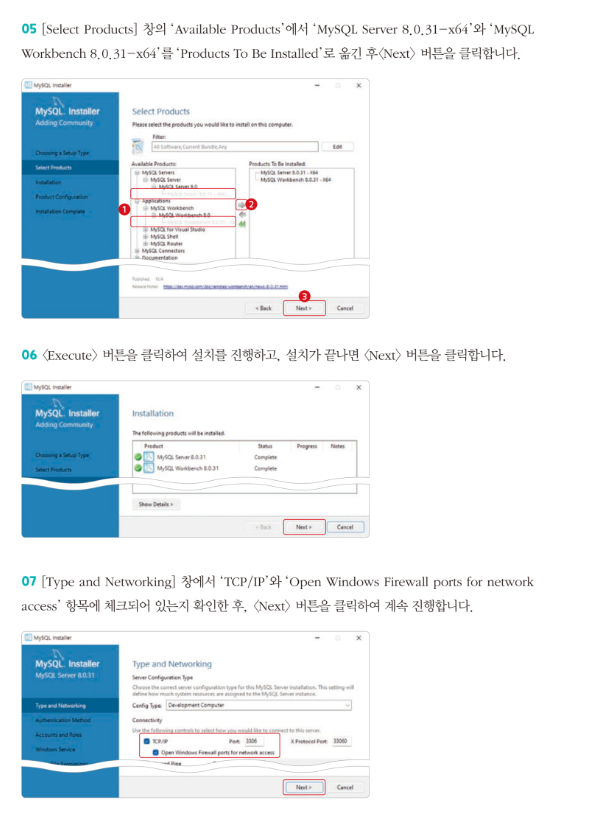
Mysql 다운로드

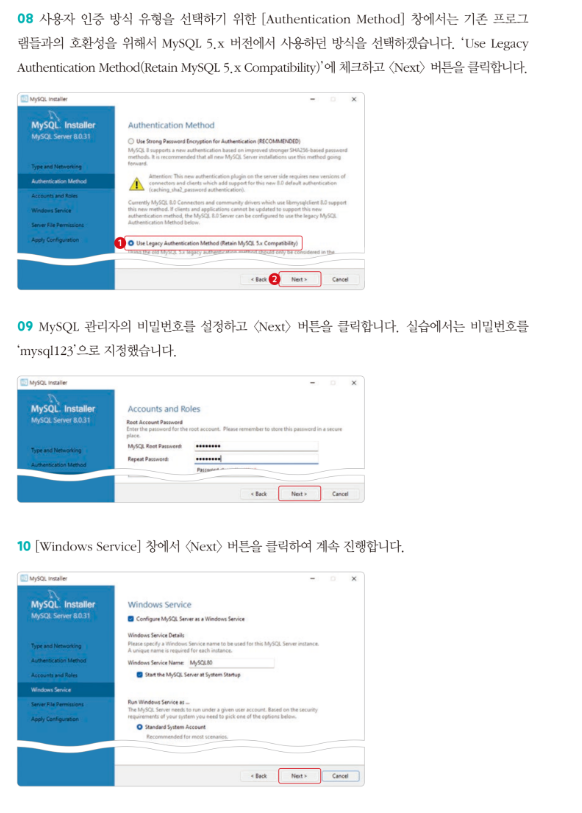


다운로드 진행







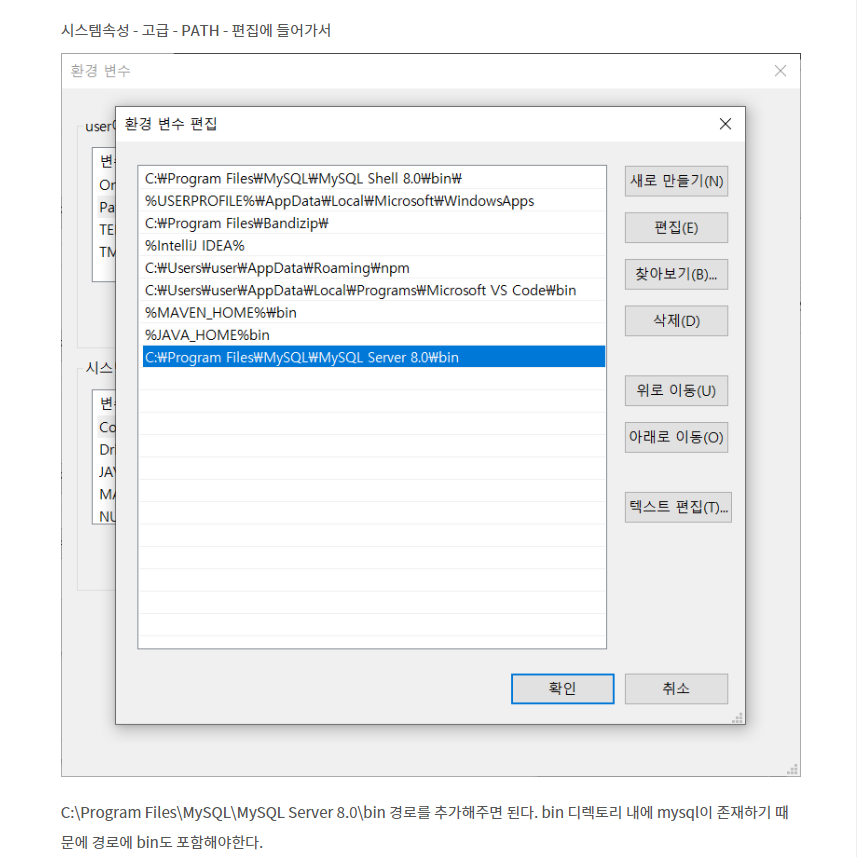




시작메뉴 - mysql – mysql command Line 실행

검색 메뉴 – 고급 시스템 설정 보기

다른방법은



Cmd를 관리자 권한으로 실행해서

Mysql 접속

mysql -u root -p 로그인 -> 비밀번호입력

Database

show databases; 데이터 목록표시

CREATE DATABASE [DB명]; 데이터베이스 생성

CREATE DATABASE sample;

DROP DATABASE [삭제할 DB명]; 데이터베이스 삭제

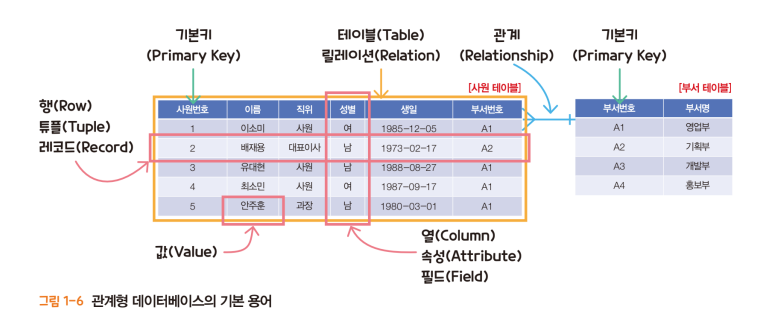
drop database sample; sample 데이터 베이스를 삭제

CREATE DATABASE test;

USE [선택할 DB명]; 데이터베이스에 접근(선택)

use test;

관계형 데이터 베이스 – 열과 행이 있는 테이블 집합으로 구성



1. 테이블 생성

CREATE TABLE [table명] (

[column1] [datatype] [option],

[column2] [datatype] [option],

[column3] [datatype] [option],

...

PRIMARY KEY ([PK로 지정할 column명]),

FOREIGN KEY ([FK로 지정할 column명]) REFERENCES [참조할 table명] ([참조할 PK명])

);

2. 데이터 베이스 테이블의 구조

데이터 베이스 테이블의 구조 보기

desc 테이블명

desc mem;

3.1 필드 추가 하기

Alter table 테이블명 add 새로운\_필드명 필드\_데이터형 [first 또는 after 필드명]

alter table mem add post\_num char(20);

3.2 필드 삭제 하기

Alter table 테이블명 drop 삭제할\_필드명1, 삭제할\_필드명2;

Alter table mem drop post\_num;

3.3 필드 수정

Alter table 테이블명 change 수정할\_필드명 새로운\_필드명 필드\_데이터형

Alter table mem change age phone char(20);

3.4. 필드 데이터형 수정하기

Alter table 테이블명 modify 필드명 수정할\_데이터형

Alter table mem modify phone int

4. 데이터베이스 테이블 이름 관리

이미 존재하는 DB 테이블 이름을 변경하는 명령

alter table 수정할\_데이블명 rename 새로운\_테이블명;

mem 테이블의 이름을 mem2로 변경.

alter table mem rename mem2;

데이터베이스 테이블 삭제

drop table 테이블명

mem2 테이블을 삭제하는 명령.

drop table mem2;

다음과 같이 friend 라는 테이블을 만들어 보시오

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)

);

use my\_database;

show tables; 테이블 전체목록

my\_database 에 있는 모든 테이블의 목록을 반환한다.(보여준다.)

member 라는 table 작성

create table member (

num int not null auto\_increment,

id char(20) not null,

name char(20) not null,

gender char(1),

post\_num char(8),

address char(80),

tel char(20),

age int,

primary key(num)

);

전체 필드와 데이터 검색

다음은 DB 테이블의 전체 필드와 데이터를 보여주는 명령 형식.

select \* from 테이블명;

select \* from member;

- 특정 필드의 전체 레코드 검색

select 필드명1, 필드명2,필드명3....from 테이블명

member 테이블의 전체 레코드에서 name, tel, address 필드 데이터만 검색하는 명령.

select name, tel, address from member;

select 필드명1, 필드명2... from 테이블명 where 조건

member 테이블에서

여성 회원의 아이디, 이름, 주소, 성별 검색

Select id, name, address, gender from member where gender = “w”;

문제1) 50세 이상인 남성 회원의 이름, 전화번호 , 성별, 나이검색

select name, tel, gender, age from member where age >= 50 and gender= "M";

문2) 20대 회원의 이름, 성별, 주소, 나이검색

select name, gender, address, age from member where age >= 20 and age<= 29

문3) 30대 또는 50대 남성회원의 이름, 전화번호, 나이, 성별 검색

select name, tel, age, gender from member where ( (age>=30 and age<=39) or (age>=50 and age<=59) ) and gender='M';

[예제 10-5] 이름이 ‘안철영’인 회원의 일련번호, 이름, 전화번호, 주소, 우편번호, 나이, 성별 검색

select num, name, tel, address, post\_num, age, gender from member where name='안철영';

특정문자열이 포함된 레코드 검색

테이블명’이라는 DB 테이블에서 ‘검색필드’에 포함된 ‘문자열\_수식’ 조건을 만족하는 레코드의 필드명1, 필드명2, …를 검색하는 명령 형식.

select 필드명1, 필드명2, ..from 테이블명 where 검색필드 like 문자열\_수식

-성씩 박씨인 회원의 이름, 전화번호 검색

-select name, tel from member where name like ‘박%’;

서울에 거주하는 회원의 일련번호, 이름, 주소, 성별

select num, name, address, gender from member where address like ‘서울%’;

- 은평구에 거주하는 회원의 모든 정보 검색

**select \* from member where address like '%은평구%';**

**성이 김씨이고 광주에 거주하는 회원의 모든 정보 검색**

**select \* from member where address like '%광주%' and name like '김%';**

레코드 정렬

‘테이블명’이라는 DB 테이블에서 ‘필드명’을 기준으로 오름차순 정렬된 필드명1, 필드 명2, …를 검색하는 명령 형식.

Select 필드명1,필드명2, … from 테이블명 order by 필드명;

**서울에 사는 회원을 나이순으로 오름차순 정렬한 뒤 이름, 나이, 주소, 전화번호, 성별 출력**

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

**부산에 사는 회원을 나이순으로 내림차순 정렬한 뒤 이름, 나이, 주소, 전화번호 출력**

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

desc – descending(내림차순)

레코드 수정

Update 테이블명 set 필드명 = 필드값 where 조건식;

이름이 ‘고재진’ 인 레코드의 전화 번호를 123-4567로 변경한 뒤 데이터 확인

update member set tel='123-4567' where name='고재진';

**select name, tel from member where name="고재진";**

이름이 ‘정한나’인 회원의 나이를 확인하고 값을 37로 변경

> select name, age from member where name='정한나';   
> update member set age=37 where name='정한나';   
> select name, age from member where name='정한나';

레코드 삭제

‘테이블명’이라는 DB테이블에서 조건식을 만족하는 레코드를 삭제하는 명령 형식

Delete from 테이블명 where 조건식;

> delete from member where name='김수련';   
> select \* from member where name like '김%';

**40대 회원의 레코드를 삭제한 뒤 35~53세 회원의 이름, 나이, 성별을 나이순으로 출력하여 확인**

> delete from member where age>=40 and age<=49;   
> select name, age, gender from member where age>=35 and age<=53 order by age;

member 테이블의 전체 레코드를 삭제하는 명령 형식.

Delete from member;

조건식을 생략한 채 delete from 명령을 실행하면 member 테이블의 전체 레코드를 삭제할 수 있음.

실수로 테이블의 레코드를 날려버릴 수도 있으니 이 명령은 조심히 사용해야 함.

SQL

관계형 데이터베이스 관리시스템(RDBMS)의 데이터를 관리하고 다양한 데이터 동작을 수행하는 데 사용하는 표준화된 프로그래밍 언어

SQL 특징

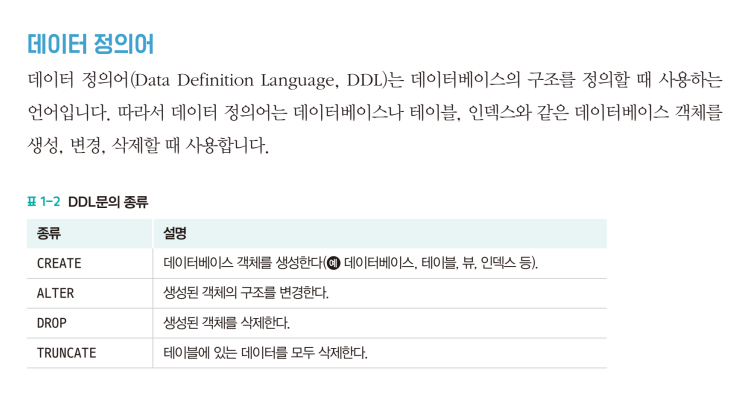
* 영어문장과 비슷하여 배우기 쉽고 사용하기 용이
* 데이터 연산에 대한 처리과정이 절차적이지 않으며 집합단위로 처리된다.
* 처리결과 중심 -> 사용자 편의 중심 언어
* 미국 표준협회 SQL 을 작성한다면 Oracle, MySQL, MS-SQL, PostgreSQL, MariaDB 등

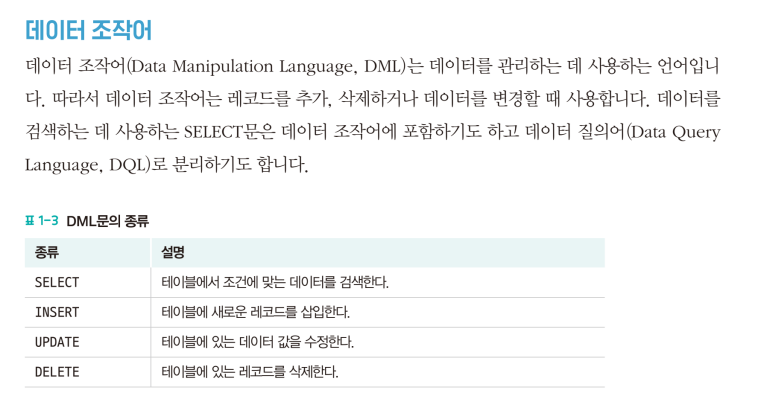
상용 RDBMS간에 호환과 전환이 용이

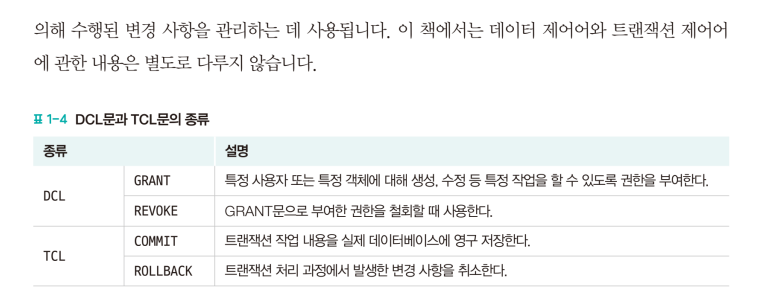
우리가 다뤄볼 SQL – MySQL, Oracle, PostgreSQL

SQL문의 종류

SQL문에서 데이터 정의어 , 데이터 조작어 , 데이터 제어어, 트랜잭션 제어어가 있습니다.







외부에서 작성된 SQL 문을 WORKBENCH에서 읽어들이려면?

FILE – OPEN SQL SCRIPT

select \* from 한빛무역.고객 limit 3;

3개만 가져오기

마일리지가 많은 고객부터 상위3명의 고객에 대한 모든 정보를 출력하시오

select \* from 한빛무역.고객 order by 마일리지 desc limit 3;

고객테이블의 도시컬럼에 들어있는 값 중 중복되는 도시 데이터를 한번만 보여주세요

select distinct 도시 from 한빛무역.고객

산술연산

SELECT 23 + 5 AS 더하기

,23 - 5 AS 빼기

,23 \* 5 AS 곱하기

,23 / 5 AS 실수나누기

,23 DIV 5 AS 정수나누기

,23 % 5 AS 나머지1

,23 MOD 5 AS 나머지2;

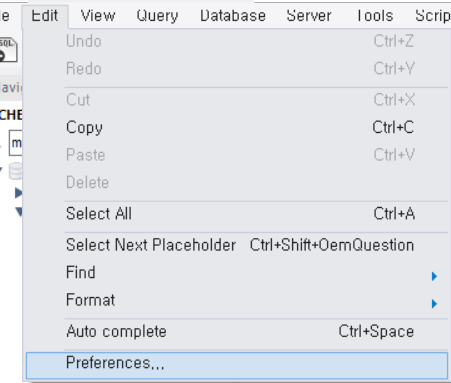
담당자가 ‘대표 이사’가 아닌 고객의 모든 정보를 보이시오

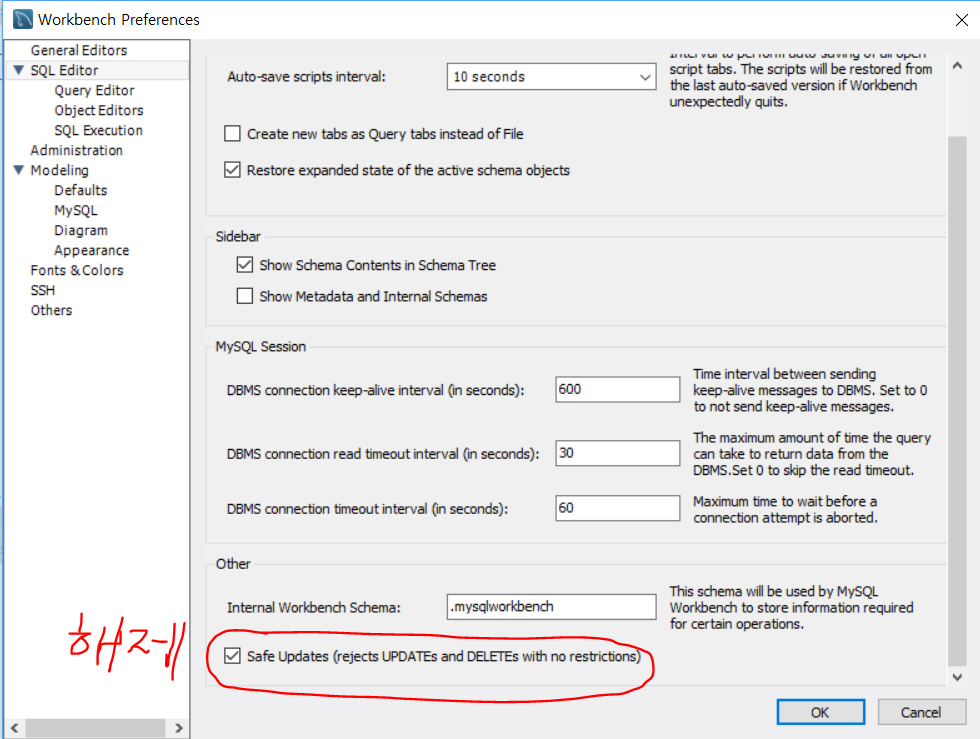
SELECT \* from 한빛무역.고객 where 담당자직위 <> '대표 이사';

UPDATE 명령이 제대로 수해되지 않으면

**1. UI 사용**

**Edit> Preferences > SQL Editor > Safe Updates 옵션 설정 후 재부팅**





SELECT 고객번호

,담당자명

,담당자직위

FROM 고객

WHERE 담당자직위 = '영업 과장'

OR 담당자직위 = '마케팅 과장';

SELECT 고객번호

,담당자명

,담당자직위

FROM 한빛무역.고객

WHERE 담당자직위 IN ('영업 과장' , '마케팅 과장');

SELECT 담당자명

,마일리지

FROM 한빛무역.고객

where 마일리지 between 100000 AND 200000;

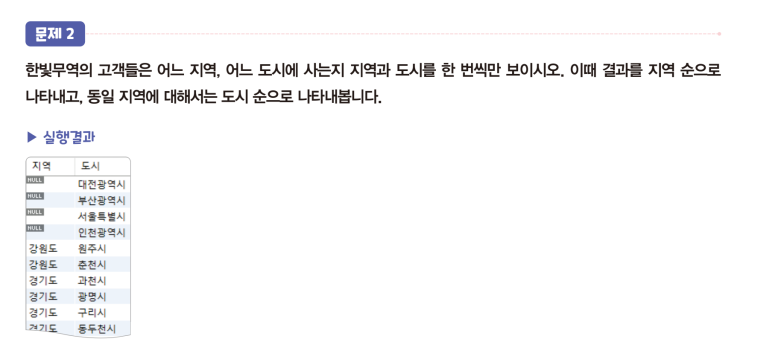
도시가 ‘광역시’ 이면서 고객번호 두 번째 글자 또는 세 번째 글자가 ‘C’인 고객의 모든정보

SELECT \*

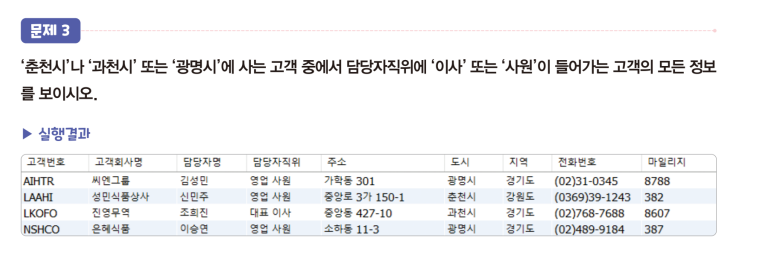
FROM 한빛무역.고객

where 도시 LIKE '%광역시'

AND (고객번호 like '\_\_C%' OR 고객번호 LIKE '\_C%');



SELECT distinct 지역, 도시 FROM 한빛무역.고객 ORDER BY 1,2;

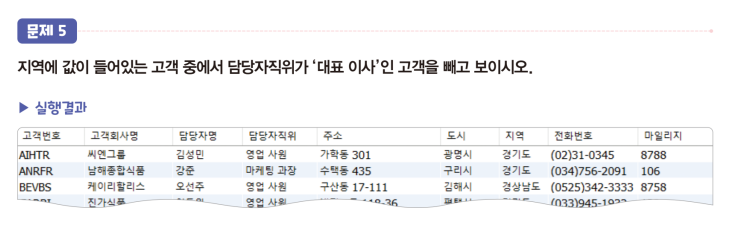


SELECT \*

FROM 한빛무역.고객

where 도시 IN ('춘천시', '과천시', '광명시')

AND (담당자직위 liKE '%이사%' OR 담당자직위 LIKE '%사원%');



SELECT \*

FROM 고객

WHERE 지역 IS NOT NULL

AND 담당자직위 <> '대표 이사';

글자의 길이

SELECT CHAR\_LENGTH('HELLO')

,LENGTH('HELLO')

,CHAR\_LENGTH('안녕')

,LENGTH('안녕');

SELECT CONCAT('DREAMS', 'COME', 'TRUE')

,CONCAT\_WS('-', '2023', '01', '29');

Left(문자열. 길이);

Right(문자열. 길이);

Substr(문자열, 시작\_위치,길이 ) 또는 substring(문자열, 시작\_위치, 길이);

반복

Select repeat(“\*”,5);

SELECT NOW()

,YEAR(NOW())

,QUARTER(NOW())

,MONTH(NOW())

,DAY(NOW())

,HOUR(NOW())

,MINUTE(NOW())

,SECOND(NOW());

SELECT NOW()

,DATEDIFF('2025-12-20', NOW())

,DATEDIFF(NOW(), '2025-12-20')

,TIMESTAMPDIFF(YEAR, NOW(), '2025-12-20')

,TIMESTAMPDIFF(MONTH, NOW(), '2025-12-20')

,TIMESTAMPDIFF(DAY, NOW(), '2025-12-20');

날짜 형식 변환

**DATE\_FORMAT(now(), '%Y-%m-%d %H:%i:%s')**

SELECT NOW()

,LAST\_DAY(NOW())

,DAYOFYEAR(NOW()) //이달의 마지막 날

,MONTHNAME(NOW())

,WEEKDAY(NOW());

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

SELECT LENGTH(LTRIM(' SQL ')) ,LENGTH(RTRIM(' SQL ')) ,LENGTH(TRIM(' SQL '));

주어진 문자열에서 왼쪽 공백을 제거한 후의 길이, 오른쪽 공백을 제거한 후의 길이, 그리고 양쪽 공백을 제거한 후의 길이를 반환합니다.

SELECT REPLACE('010.1234.5678', '.', '-');

SELECT REVERSE('OLLEH');

SELECT CEILING(123.56)

,FLOOR(123.56)

,ROUND(123.56)

,ROUND(123.56, 1)

,TRUNCATE(123.56, 1); //버림할\_자릿수

SELECT ABS(-120)

,ABS(120)

,SIGN(-120)

,SIGN(120);

Sign 양수인경우 1, 음수 는 -1을 반환

SELECT MOD(203, 4)

,203 % 4

,203 MOD 4;

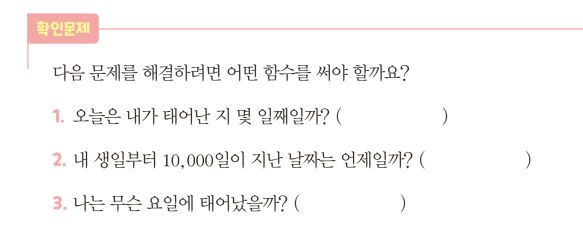
SELECT POWER(2, 3) //2의 3승

,SQRT(16) //루트

,RAND() // 난수

,RAND(100)

,ROUND(RAND() \* 100);





SELECT 고객회사명

,CONCAT('\*\*', SUBSTR(고객회사명, 3)) AS 고객회사명2

,전화번호,REPLACE(SUBSTR(전화번호,2), ')', '-') AS 전화번호2

FROM 고객;

CONCAT('\*\*', SUBSTR(고객회사명, 3)) AS 고객회사명2

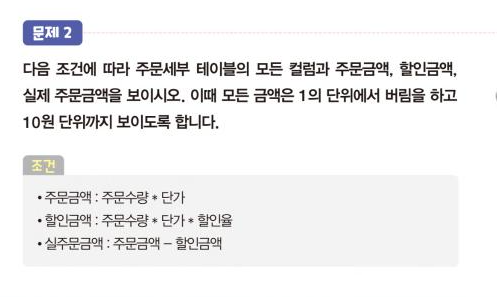
SUBSTR 세번째문자부터 끝까지 선택 한후 , **CONCAT** 함수를 사용하여 "\*\*"와 결합

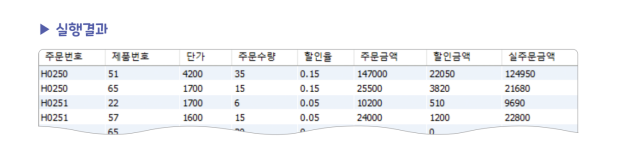
결과적으로 이 부분을 **고객회사명2**라는 새로운 열로 생성합니다.

REPLACE(SUBSTR(전화번호,2), ')', '-') AS 전화번호2

전화번호의 두번째 끝까지 선택한후, replace함수를 사용해서 “)” 를 “-“로 대체

결과적으로 전화번호2 라는 새로운열 생성





Select \* , 단가 \* 주문수량 as 주문금액

, Truncate(단가 \* 주문수량 \* 할인율, -1) as 할인금액

,단가 \* 주문수량 – truncate(단가 \* 주문수량 \* 할인율, -1) as 실주문금액

From 주문세부;

숫자를 카운트

고객테이블에서 고객번호, 도시 , 지역의 개수 조회

SELECT COUNT(\*)

,COUNT(고객번호)

,COUNT(도시)

,COUNT(지역)

FROM 고객;

고객테이블의 마일리지 컬럼에 대하여 마일리지 합과 평균 마일리지, 최소, 최대 마일리지 조회

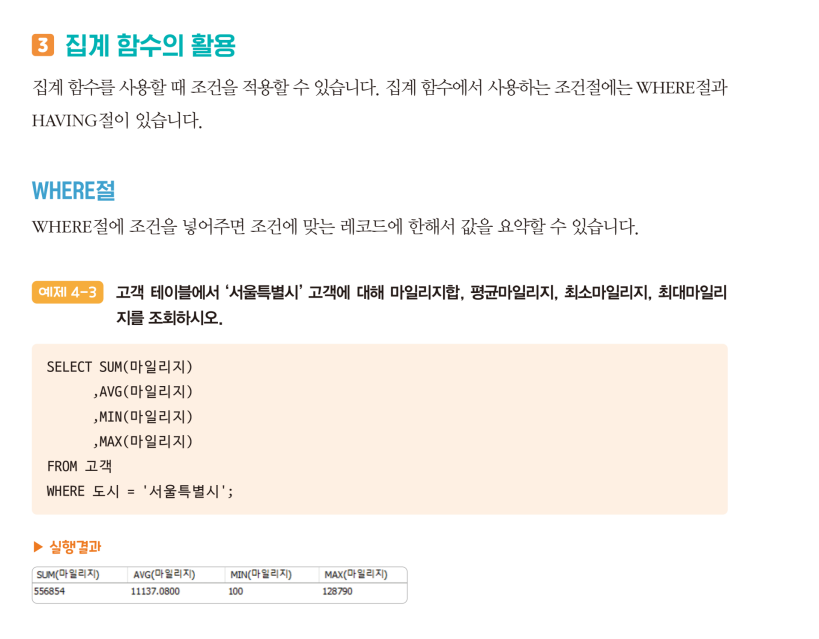
SELECT SUM(마일리지)

,AVG(마일리지)

,MIN(마일리지)

,MAX(마일리지)

FROM 고객;



고객 테이블에서 도시별 고객의 수와 해당 도시 고객들의 평균마일리지를 조회하시오

SELECT 도시 ,COUNT(\*) AS 고객수 ,AVG(마일리지) AS 평균마일리지 FROM 고객 GROUP BY 도시;

도시별로 그룹을 묶어서 요약하려면 GROUP BY절에 도시 컬럼을 넣어야 한다.

SELECT 도시 ,COUNT(\*) AS 고객수 ,AVG(마일리지) AS 평균마일리지 FROM 고객 GROUP BY 1;

**SELECT 담당자직위**

**,도시**

**,COUNT(\*) AS 고객수**

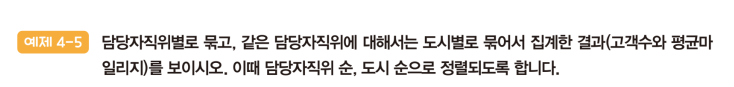
**,AVG(마일리지) AS 평균마일리지**

**FROM 고객**

**GROUP BY 담당자직위**

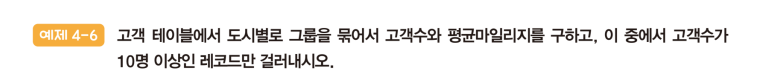
**,도시**

**ORDER BY 1, 2;**

****

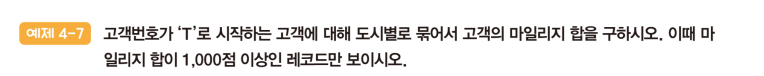
**SELECT 담당자직위 ,도시 ,COUNT(\*) AS 고객수 ,AVG(마일리지) AS 평균마일리지 FROM 고객**

**GROUP BY 담당자직위 ,도시 ORDER BY 1, 2;**

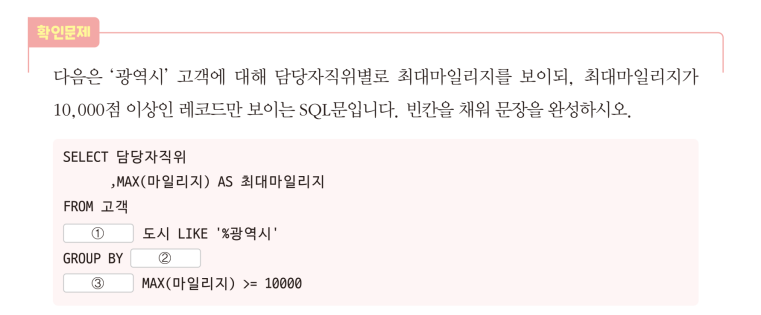
****

**SELECT 도시 ,COUNT(\*) AS 고객수 ,AVG(마일리지) AS 평균마일리지 FROM 고객**

**GROUP BY 도시 HAVING COUNT(\*) >= 10;**

****

**SELECT 도시, SUM(마일리지) FROM 고객 WHERE 고객번호 LIKE ‘T%’ GROUP BY 도시 HAVING SUM(마일리지) >=1000;**



SELECT 담당자직위, MAX(마일리지) AS 최대마일리지 FROM 고객 WHERE 도시 LIKE ‘%광역시’

GROUP BY 담당자직위 HAVING MAX(마일리지) >=10000;

WITH ROLLUP 그룹별 소계와 전체 총계를 한번에 확인하고 싶을 때 사용

SELECT 도시

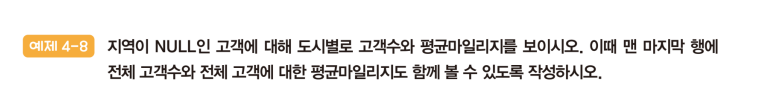
,COUNT(\*) AS 고객수

,AVG(마일리지) AS 평균마일리지

FROM 고객

WHERE 지역 IS NULL

GROUP BY 도시 WITH ROLLUP;



SELECT IFNULL(도시, ’총계’) AS 도시

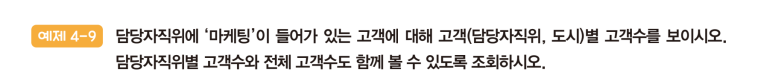
,COUNT(\*) AS 고객수

,AVG(마일리지) AS 평균마일리지

FROM 고객

WHERE 지역 IS NULL

GROUP BY 도시 WITH ROLLUP;



SELECT 담당자직위

,도시

,COUNT(\*) AS 고객수

FROM 고객

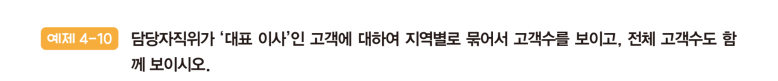
WHERE 담당자직위 LIKE '%마케팅%'

GROUP BY 담당자직위

,도시

WITH ROLLUP;

(담당자직위, 도시)별 소계 (담당자직위별) 소계 , 총계를 한번 확인 가능



SELECT 지역 ,COUNT(\*) AS 고객수, GROUPING(지역) AS 구분 FROM 고객 WHERE 담당자직위 = '대표 이사' GROUP BY 지역 WITH ROLLUP;

* 만약 특정 지역의 부분 합계 행이라면, **GROUPING(지역)**은 1을 반환합니다.
* 만약 특정 지역의 실제 그룹 행이라면, **GROUPING(지역)**은 0을 반환합니다.

SELECT GROUP\_CONCAT(이름) FROM 사원;

SELECT GROUP\_CONCAT(DISTINCT 지역) FROM 고객;

SELECT 도시 ,GROUP\_CONCAT(고객회사명) AS 고객회사명목록

FROM 고객

GROUP BY 도시;