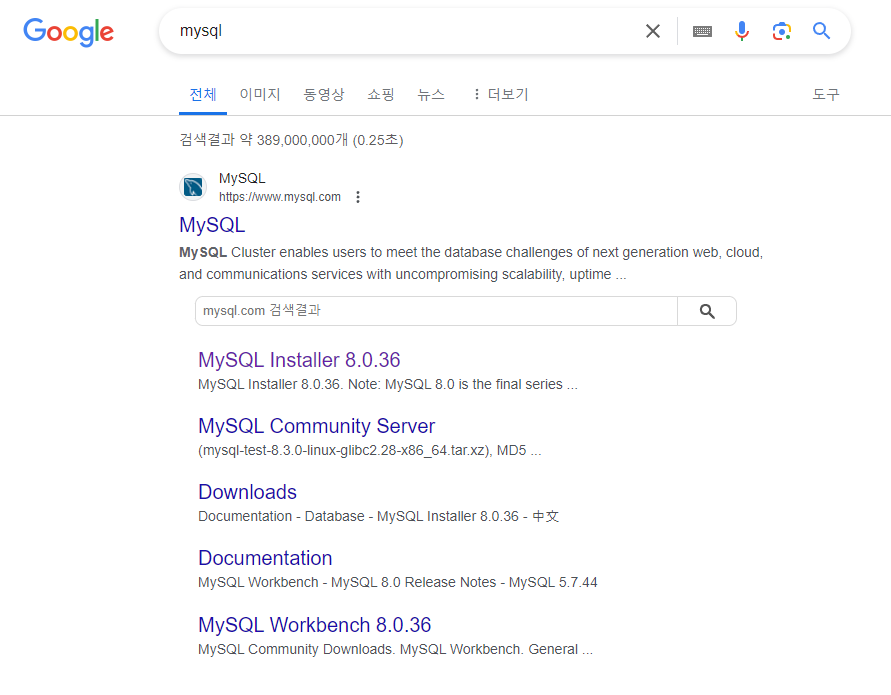
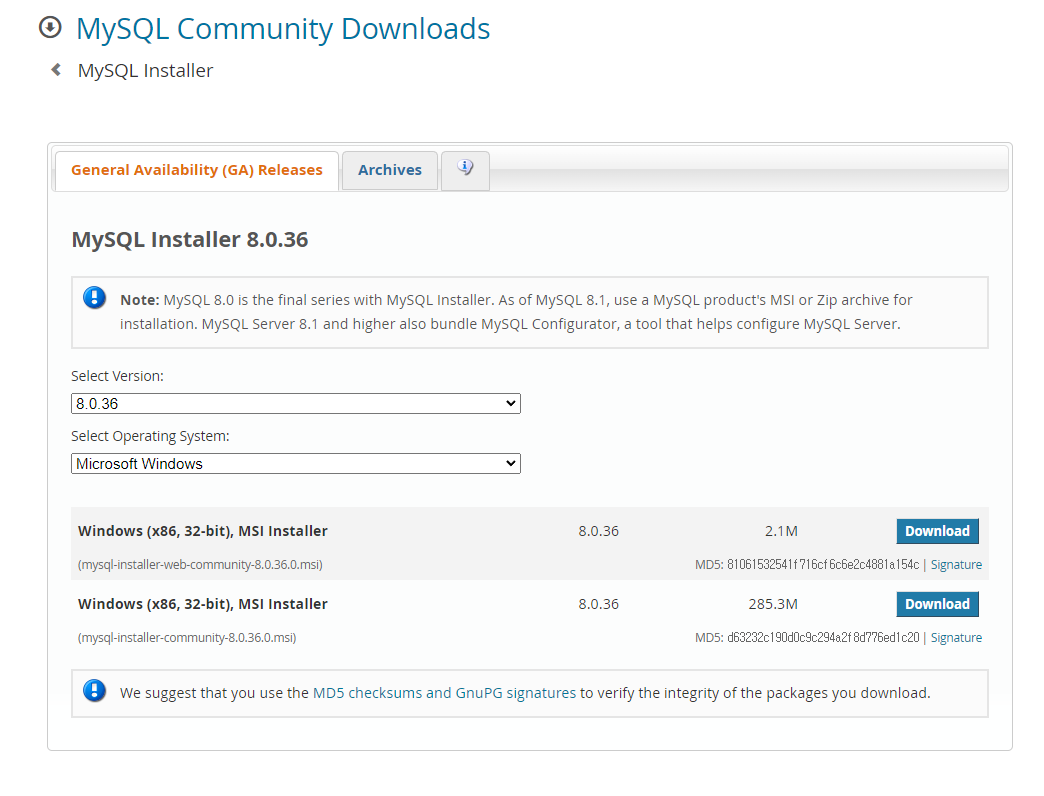
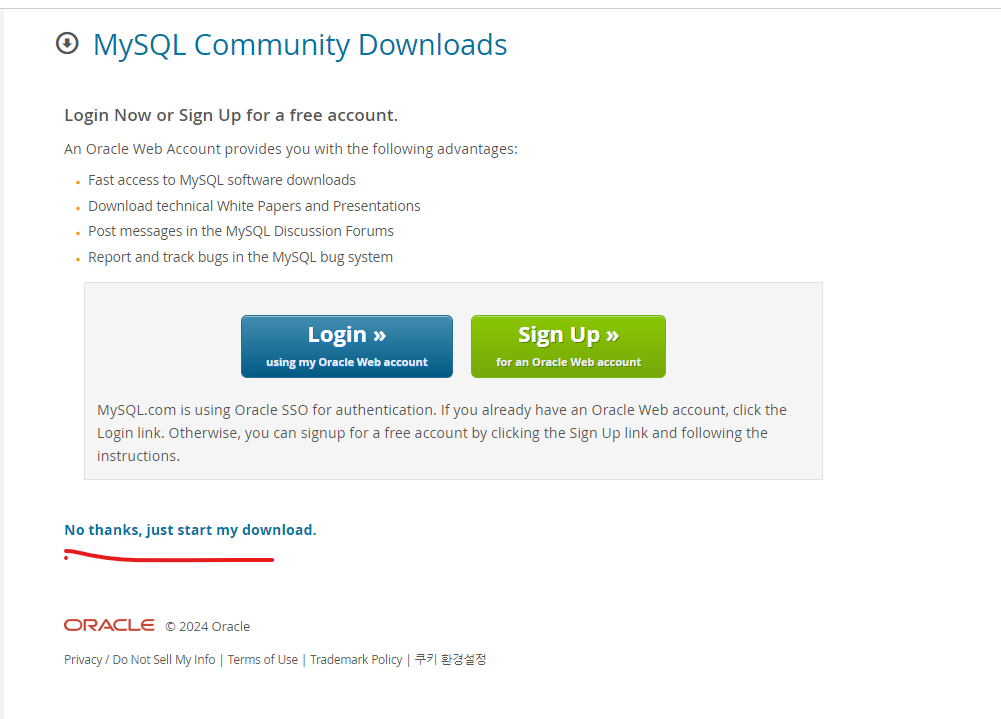
Mysql

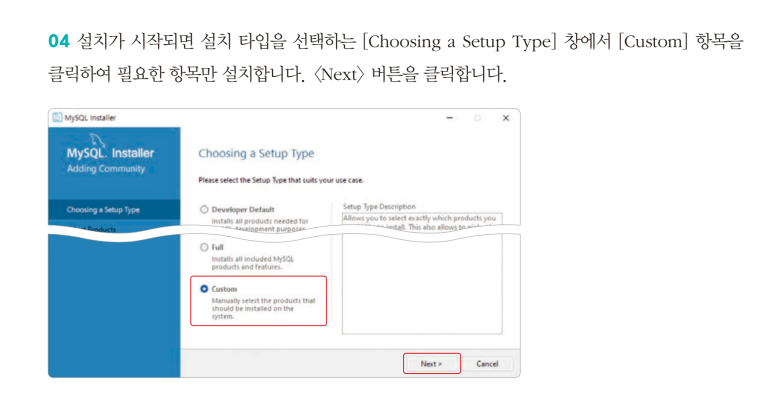
Mysql 다운로드

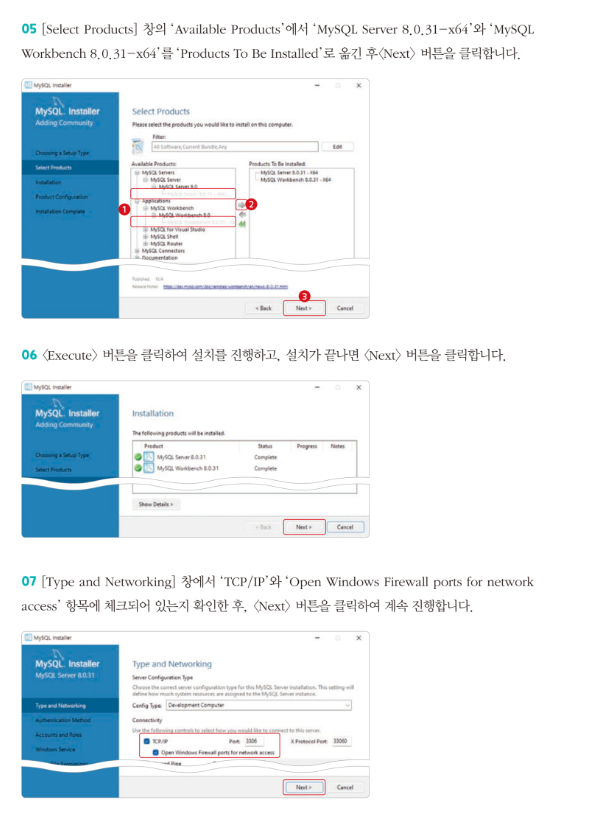


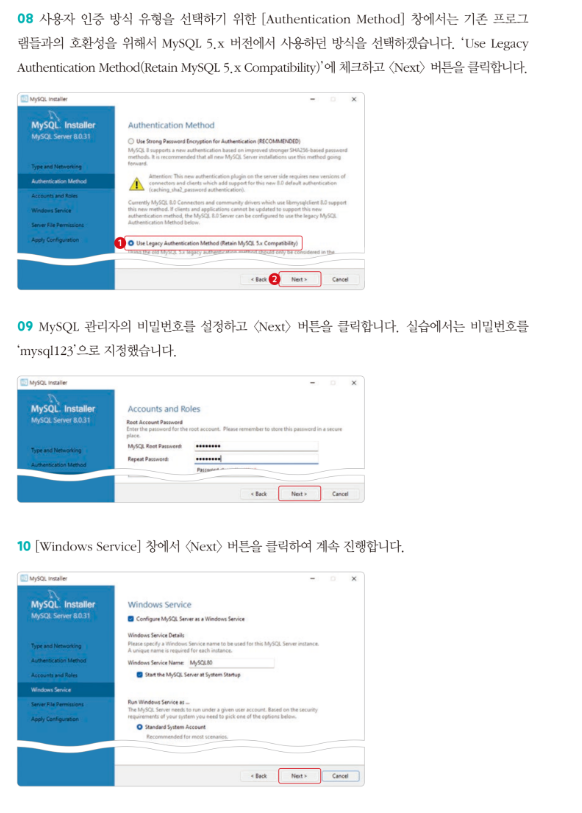


다운로드 진행





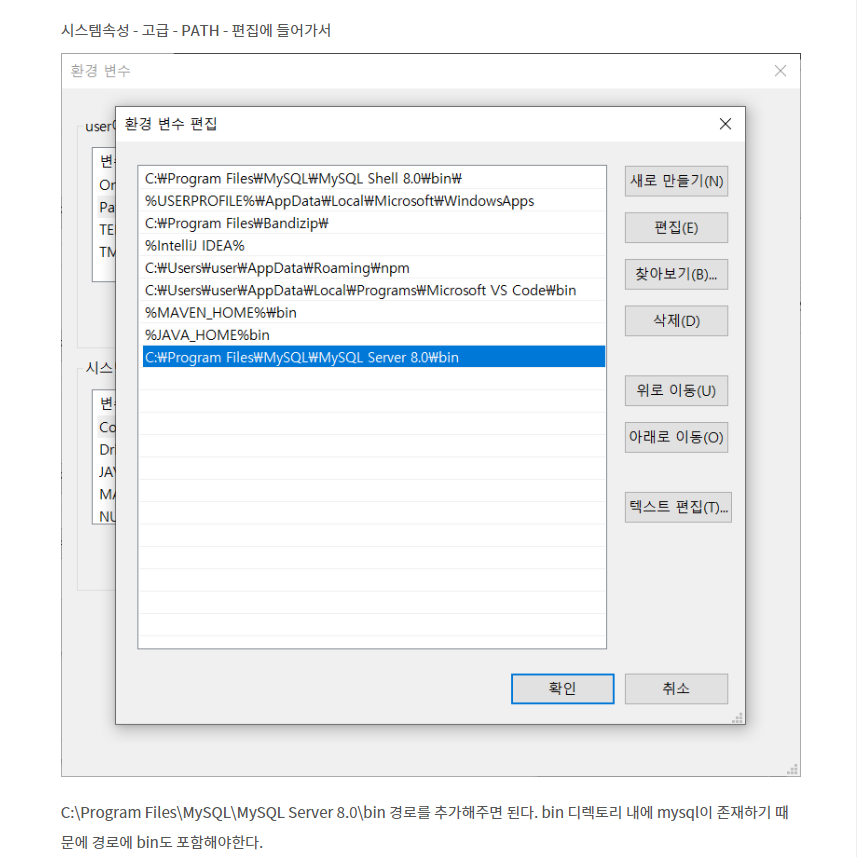






시작메뉴 - mysql – mysql command Line 실행

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

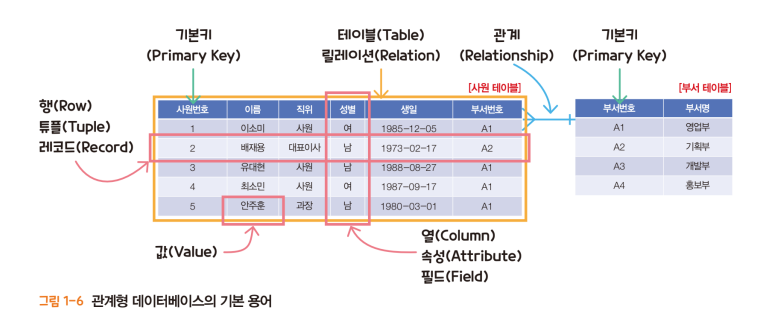
drop database sample; sample 데이터 베이스를 삭제

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])

);

CREATE TABLE member -- 회원 테이블

( mem\_id CHAR(8) NOT NULL PRIMARY KEY, # primary key로 설정

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

);

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

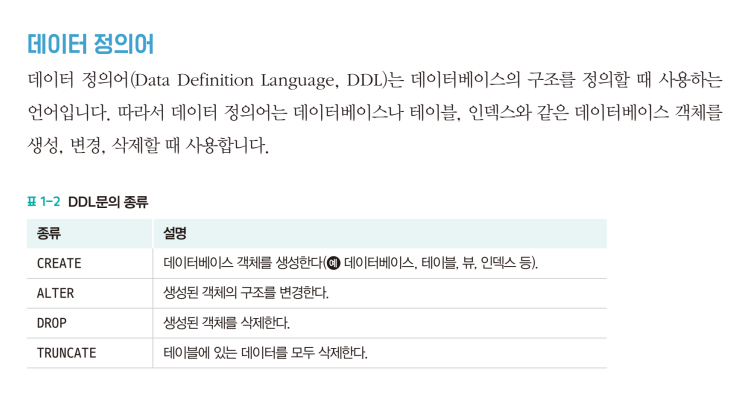
* 영어문장과 비슷하여 배우기 쉽고 사용하기 용이
* 데이터 연산에 대한 처리과정이 절차적이지 않으며 집합단위로 처리된다.
* 처리결과 중심 -> 사용자 편의 중심 언어
* 미국 표준협회 SQL 을 작성한다면 Oracle, MySQL, MS-SQL, PostgreSQL, MariaDB 등

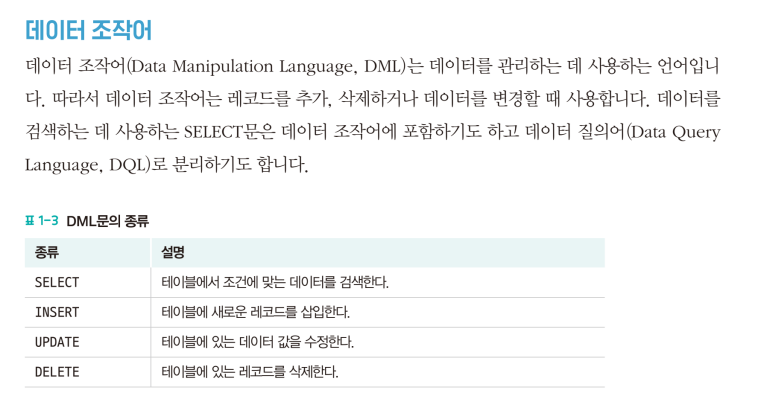
상용 RDBMS간에 호환과 전환이 용이

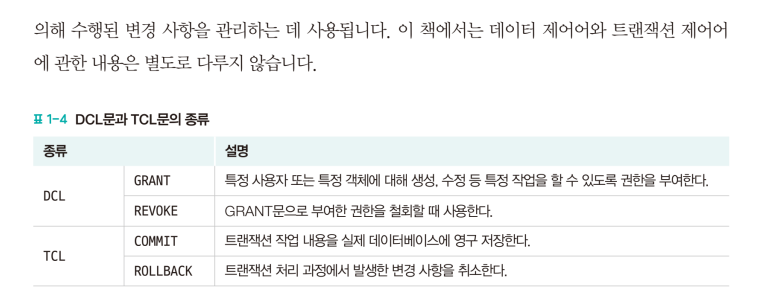
우리가 다뤄볼 SQL – MySQL, Oracle, PostgreSQL

SQL문의 종류

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



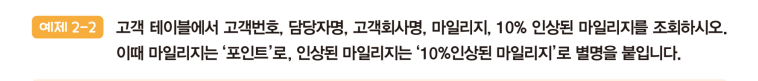


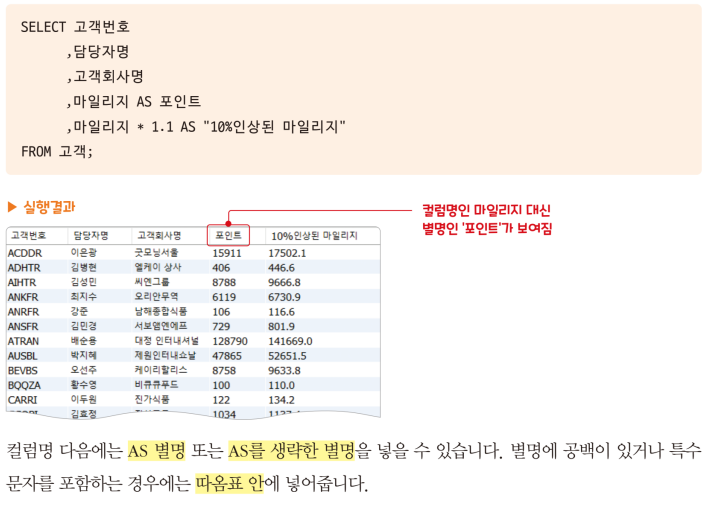


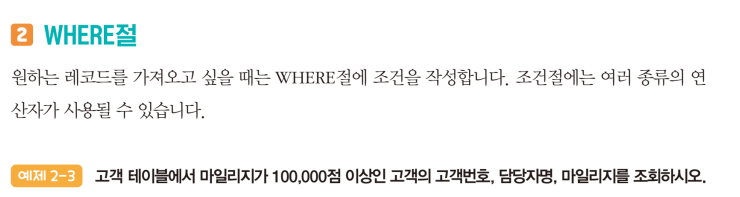
외부에서 작성된 SQL 문을 WORKBENCH에서 읽어들이려면?

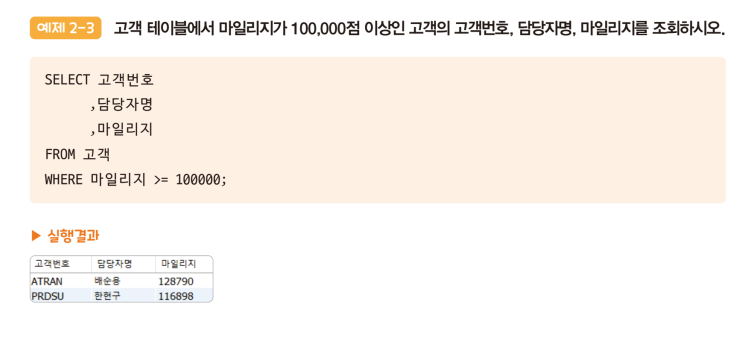
FILE – OPEN SQL SCRIPT

한빛무역 데이터베이스에서

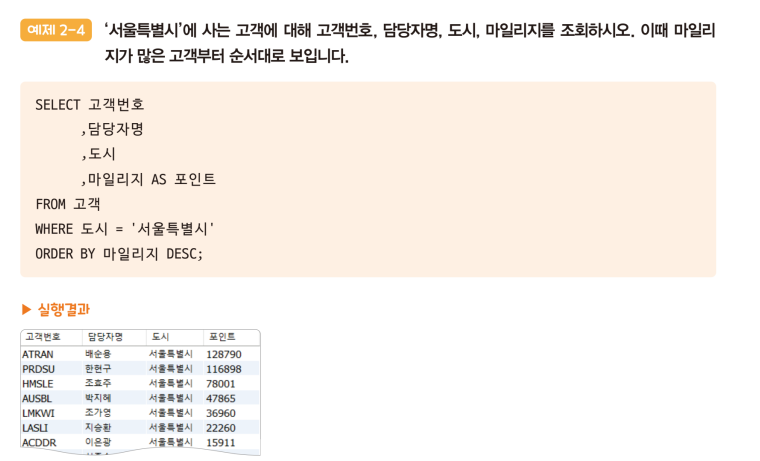




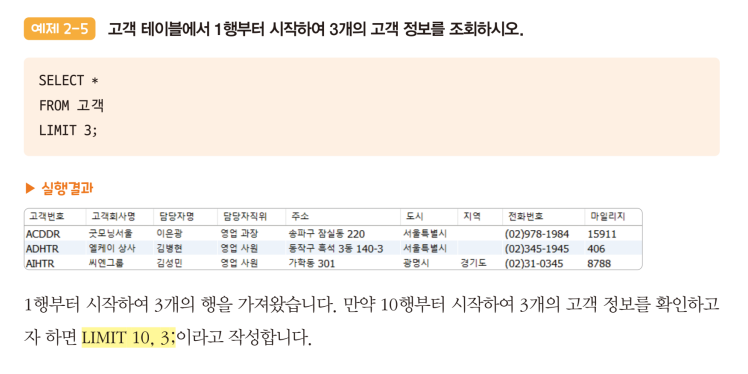








Limit 갯수 제한



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

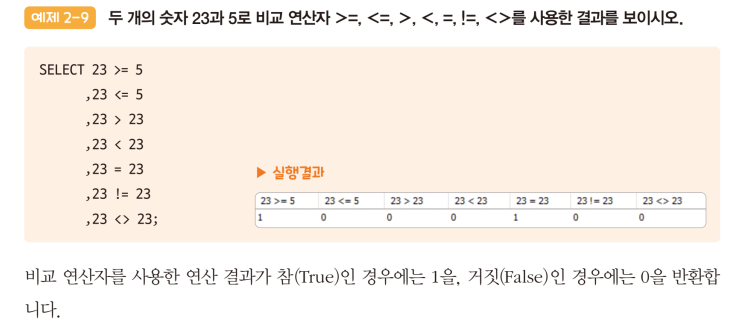
,23 / 5 AS 실수나누기

,23 DIV 5 AS 정수나누기

,23 % 5 AS 나머지1

,23 MOD 5 AS 나머지2;

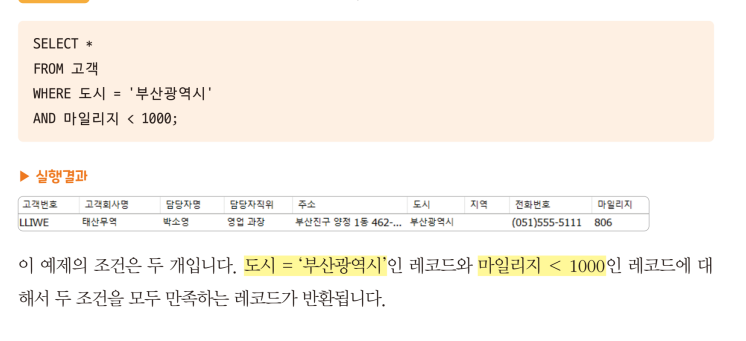
비교연산자

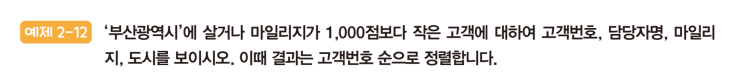


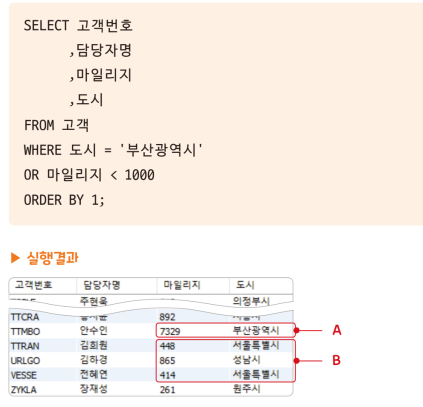
담당자가 ‘대표 이사’가 아닌 고객의 모든 정보를 보이시오

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









지역에 값이 들어있지 않는 고개의 정보를 보이시오

select \* from 고객 where 지역 IS NULL;

CSV 파일에서 테이블로 가져오기 했을 때 값이 들어있지 않은 셀은 NULL 이 아닌 빈문자열

select \* from 고객 where 지역 = '';

된다

그래서 빈문자열을 NULL로 변경하려면

UPDATE 고객 SET 지역 = NULL WHERE 지역 = '';

안됨 – safe update mode 에서는 실수로 전체 테이블을 업데이트 방지

SET SQL\_SAFE\_UPDATES = 0; //붙여줘야함

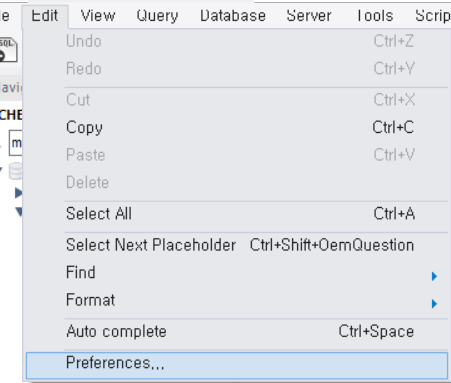
UPDATE 고객 SET 지역 = NULL WHERE 지역 = '';

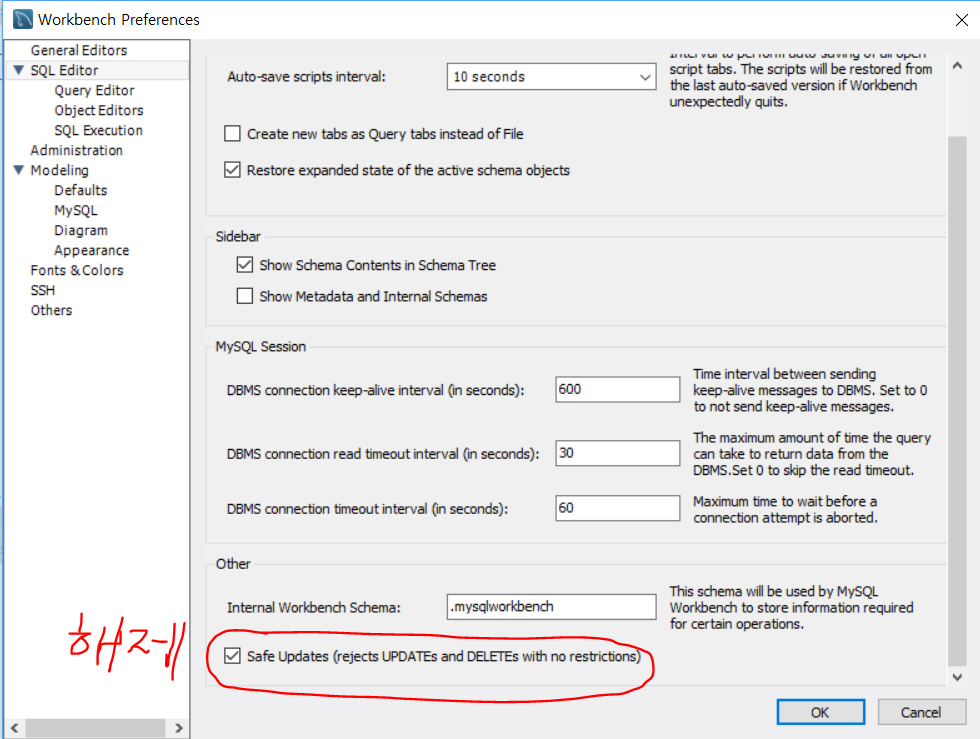
SET SQL\_SAFE\_UPDATES = 1; // safe update mode 원래데로

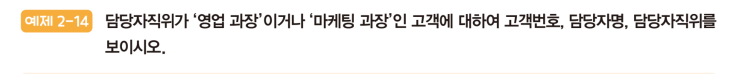
또는

**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';

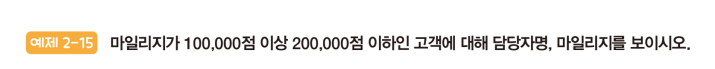
SELECT 고객번호

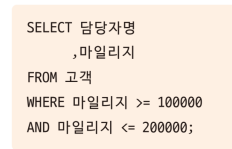
,담당자명

,담당자직위

FROM 고객

WHERE 담당자직위 IN ('영업 과장' , '마케팅 과장');



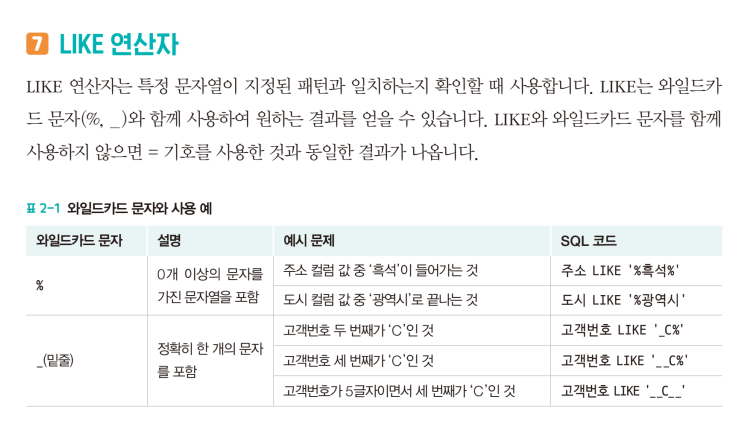


SELECT 담당자명

,마일리지

FROM 고객

where 마일리지 between 100000 AND 200000;

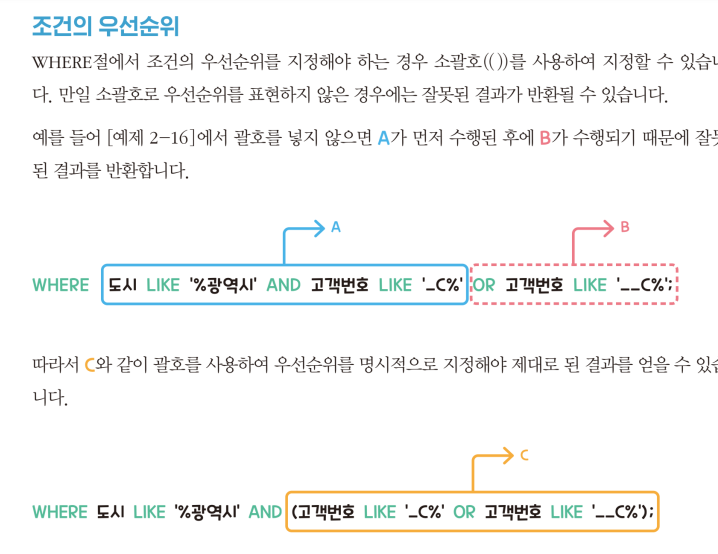


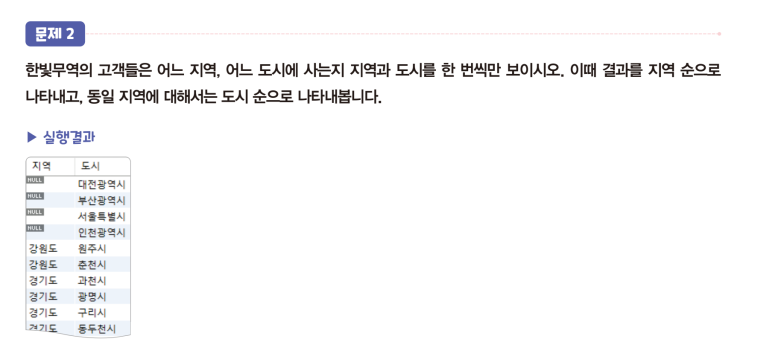
도시가 ‘광역시’ 이면서 고객번호 두 번째 글자 또는 세 번째 글자가 ‘C’인 고객의 모든정보

SELECT \* FROM 고객

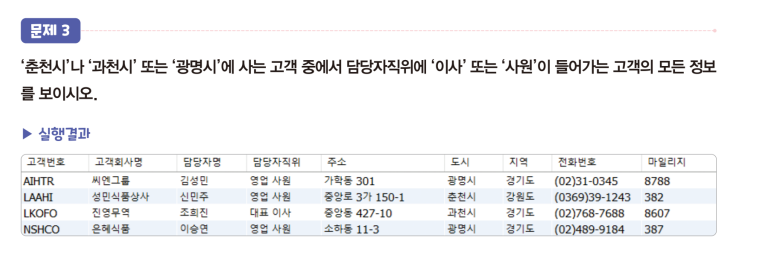
where 도시 LIKE '%광역시'

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





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

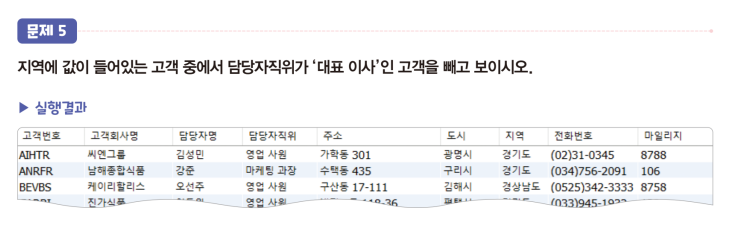


SELECT \*

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') //영문자 개수반환 -5

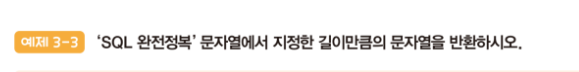
,LENGTH('HELLO') //영문자 바이트수 5

,CHAR\_LENGTH('안녕') //문자 개수 2

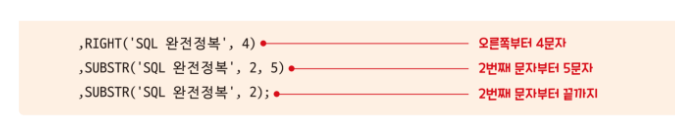
,LENGTH('안녕'); //utf-8 바이트수 6

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

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



SELECT LEFT(‘SQL 완전정복’,3) -> 왼쪽부터 3글자 추출



Left(문자열. 길이);

Right(문자열. 길이);

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

SELECT LPAD('SQL',10,'#') //#######SQL

,RPAD('SQL',5,'\*');//SQL\*\*

LPAD(문자열, 길이, 채울문자열)

SELECT LENGTH(LTRIM(' SQL '))

,LENGTH(RTRIM(' SQL '))

,LENGTH(TRIM(' SQL ')); //양쪽공백을 제거후 글자수

SELECT TRIM(BOTH 'abc' FROM 'abcSQLabcabc') //양쪽의 모든 ABC제거

,TRIM(LEADING 'abc' FROM 'abcSQLabcabc') // 왼쪽의 ABC제거

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

반복

Select repeat(“\*”,5);

Select replace('010.1234.5678','.','-'); // ‘.’ 으로 구분되어있는 전화번호를 ‘-‘로 대체

Select ceiling(123.56)

,floor(123.56)

,ROUND(123.56)

,ROUND(123.56,1) //소수 둘째자리 반올림 첫째짜리 표현

,TRUNCATE(123.56,1); // 소수 둘째자리 버림

Select mod(203, 4) //나눈 나머지

,203 % 4

,203 MOD 4

Select POWER(2,3) //거듭제곱

,SQRT(16) // 제곱근

,rand() //랜덤값

,RAND(100)

,ROUND(RAND()\*100)

Select NOW(), sysdate(), curdate(),curtime();

현재의 시간이 다양한 형태로 나옴

SELECT NOW()

,YEAR(NOW())

,QUARTER(NOW())

,MONTH(NOW())

,DAY(NOW())

,HOUR(NOW())

,MINUTE(NOW())

,SECOND(NOW());

SELECT NOW()

,DATEDIFF('2026-12-20',NOW())

,DATEDIFF(NOW(),'2026-12-20')

,timestampdiff(YEAR, NOW(), '2026-12-20')

,timestampdiff(MONTH, NOW(), '2026-12-20')

,timestampdiff(DAY, NOW(), '2026-12-20') 날짜 형식 변환



SELECT NOW()

,adddate(NOW(),50)

,adddate(NOW(),INTERVAL 50 DAY)

,adddate(NOW(),INTERVAL 50 MONTH) 50개얼후

,subdate(NOW(),INTERVAL 50 HOUR); 50시간전

SELECT NOW()

,LAST\_DAY(NOW()) //이번달 마지막날짜

,DAYOFYEAR(NOW()) 일년중 몇일째

,MONTHNAME(NOW())

,WEEKDAY(NOW()); //월요일0 오늘이 목요일 3

SELECT cast('1' AS UNSIGNED)

,cast(2 AS CHAR(1))

,CONVERT('1', UNSIGNED)

,CONVERT(2, CHAR(1));

//숫자를 문자로 또는 문자를 숫자로 변경



SELECT 고객회사명

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

,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;

**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;

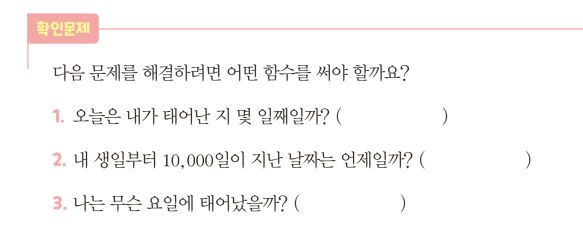
SELECT POWER(2, 3) //2의 3승

,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

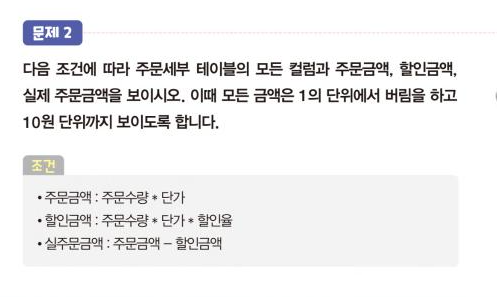
SUBSTR 세번째문자부터 끝까지 선택 한후 , **CONCAT** 함수를 사용하여 "\*\*"와 결합

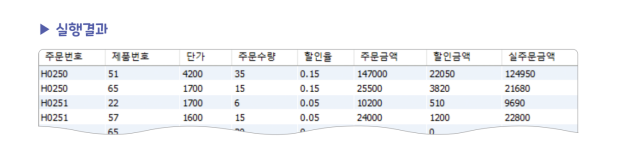
결과적으로 이 부분을 **고객회사명2**라는 새로운 열로 생성합니다.

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

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

결과적으로 전화번호2 라는 새로운열 생성



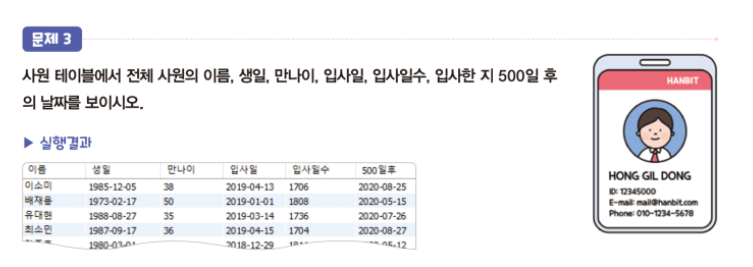


Select \* , 단가 \* 주문수량 as 주문금액

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

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

From 주문세부



SELECT 이름, 생일

,timestampdiff(YEAR, 생일, CURDATE()) AS 만나이

,입사일

,datediff(CURDATE(), 입사일) AS 입사일수

,adddate(입사일, 500) AS `500일후`

FROM 사원;



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'

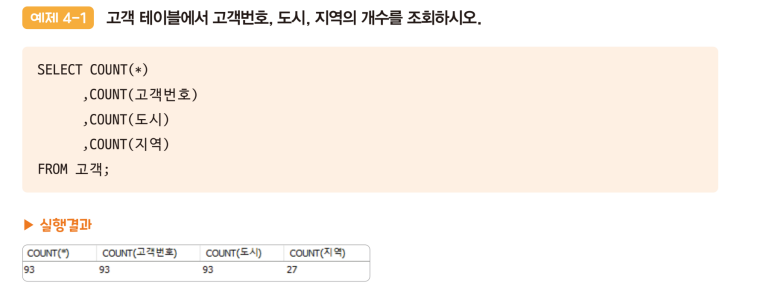
ELSE '일요일'

END AS 한글요일

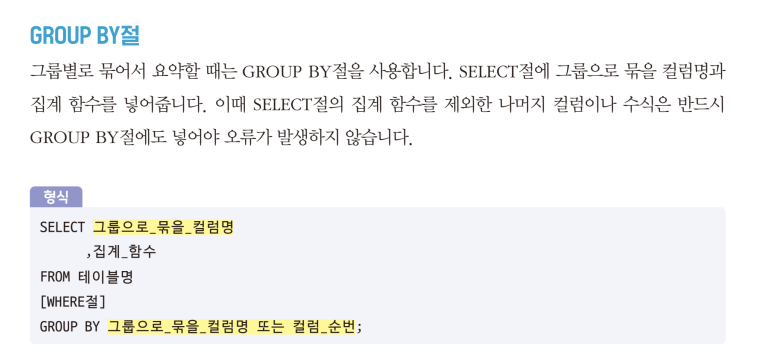
FROM 주문;

집계 함수의 개념

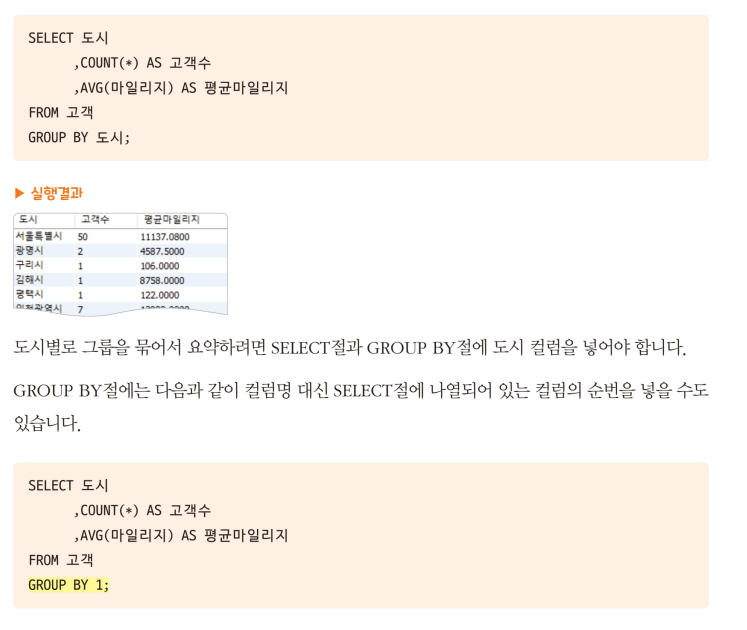
SELECT 집계\_함수 FROM 테이블명 WHERE절

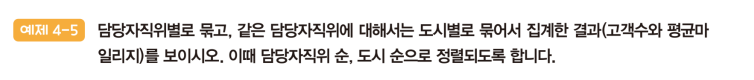


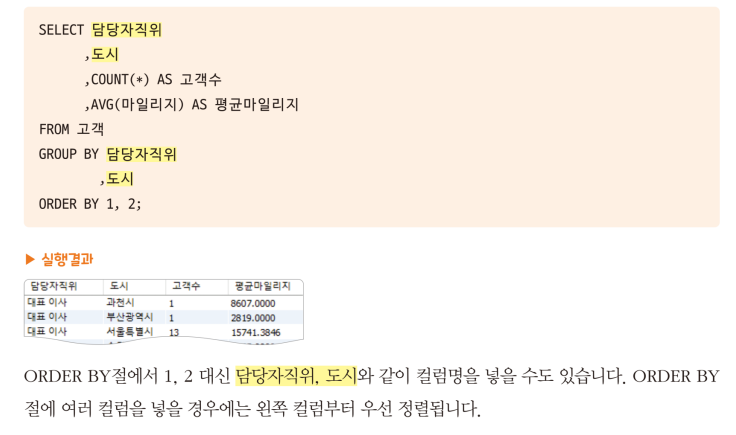




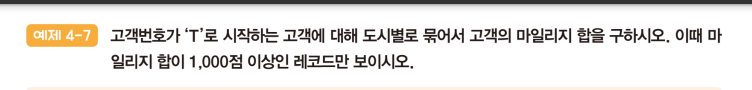


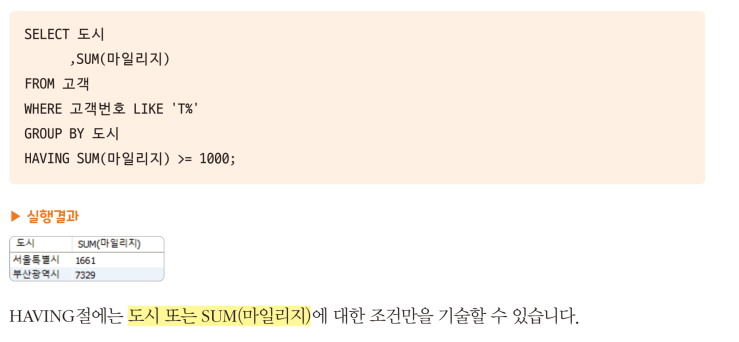


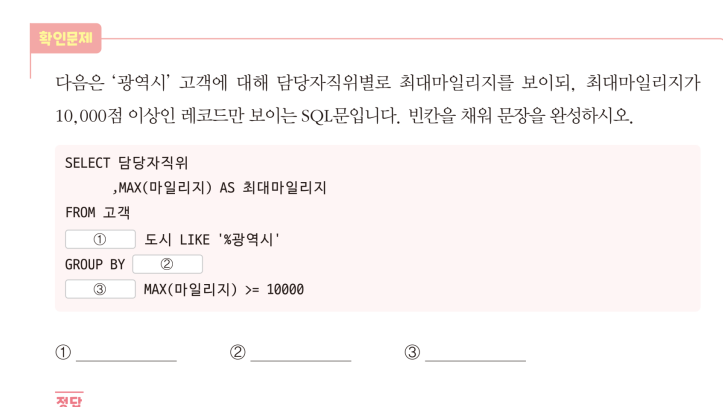










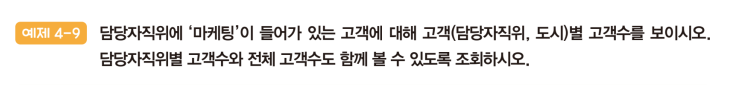


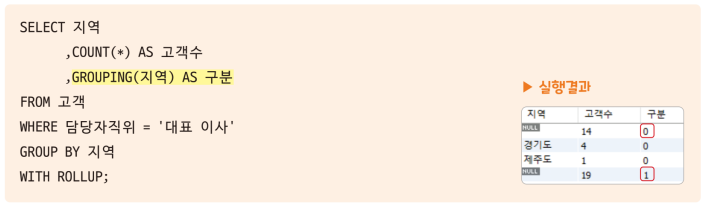
1. WHERE 2. 담당자 직위 3. HAVIING

WITH ROLLUP

그룹별 소계와 전체 총계를 한번 확인하고 싶을 때 사용







GROUPING()은 WITH ROLLUP의 결과로 나온 NULL 에 대해서 1을 반환

그렇지 않으면 0



SQL문의 종류

데이터 정의어

데이터 베이스나 테이블, 인덱스와 같은 데이터베이스 객체를 생성, 변경 , 삭제할 째 쓰임

DDL

CREATE 데이터베이스 객체생성

ALTER 생성된 객체의 구조 변경

DROP 생성된 객체를 삭제

TRUNCATE 테이블에 있는 데이터 모두 삭제

데이터 조작어

DML – 데이터 관리

레코드 추가, 삭제하거나 변경, 검색

SELECT 테이블에서 조건에 맞는 데이터 검색

INSERT 테이블에 새로운 레코드 삽입

UPDATE 테이블에 있는 테이터값 을 수정

DELETE 테이블에 있는 레코드 삭제

데이터 제어어 와 트랜잭션 제어어

DCL

DCL - GRANT 특정 사용자 또는 특정 객체에 대해 생성, 수정등 특정 작업에 권한 부여

* REVOKE GRANT문으로 부여한 권한을 철회할 때 사용한다.

TCL – COMMIT 트랜잭션 작업 내용을 실제 데이터에이스에서 영구 저장

-ROLLBACK 트랜잭션 처리과정에서 발생한 변경 사항을 취소한다.

Root 로 진입

Mysql -u root -p

Mysql 사용

Use my sql;

현재 로그인한 사용자

Select current\_user();

결과 : root@localhost

1. 새로운 User 생성

CREATE USER junsuk@localhost IDENTIFIED BY ‘12345’;

CREATE USER junsuk@localhost IDENTIFIED BY ‘ password’;

특정 db(test)에 대한 전체 권한 추가

GRANT ALL ON test.\* TO junsuk@localhost;

메모리에 적용(저장)

Flush PRIVILEGES;

전체 db에 대한 전체 권한 추가

GRANT ALL ON \*.\* TO junsuk@localhost;

GRANT ALL ON \*.\* TO junsuk@’%’;

전체 db에 대한 SELECT, INSERT 권한 추가

GRANT SELECT, INSERT ON \*.\* TO junsuk@localhost;

특정db(TEST)에 대한 SELECT, INSERT 권한 추가

GRANT SELECT, INSERT ON test.\* TO junsuk@localhost;

특정 db(test)에 포함된 특정 Table(friend)대한 권한 주기

GRANT ALL ON test.friend TO junsuk@localhost;

쓰기 , 수정 권한을 삭제

REVOKE INSERT, UPDATE ON test.\* FROM junsuk;

사용자 전체권한 삭제

REVOKE ALL ON \*.\* FROM junsuk;

사용자 전체 쓰기 권한 삭제

REVOKE INSERT ON \*.\* TO junsuk@localhost

유저 삭제

DROP USER junsuk@localhost;

localhost; 와 % 차이

로컬컴퓨터(자기 자신)에서만 접속 허용

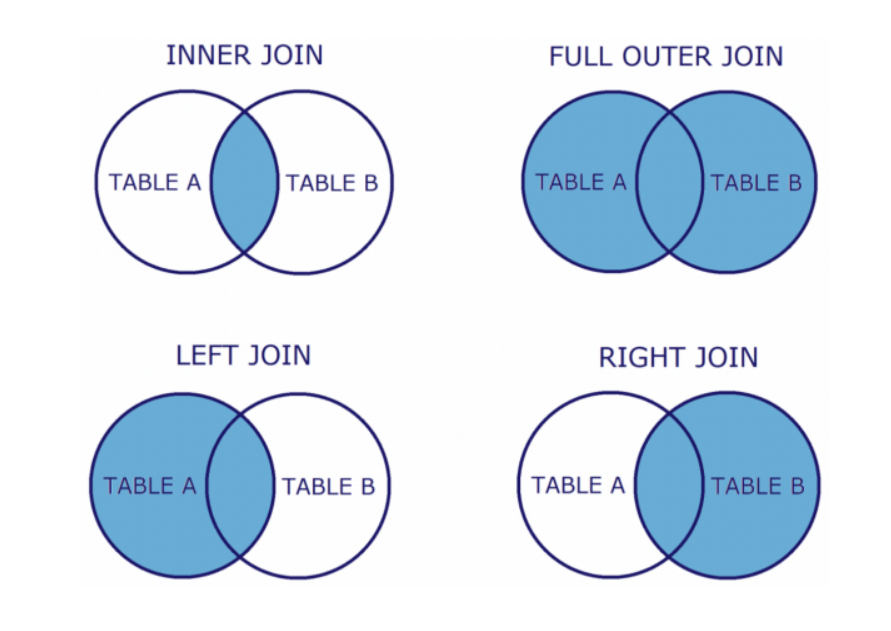
% 어디서든 접속 가능 (보안위험, 방화벽 필수)

Create user [junsuk@192.168.1.10](mailto:junsuk@192.168.1.10) IDENTIFIED BY ‘1234’;

조인

조인(Join)은 두 개 이상의 테이블을 연결하여 데이터를 검색하는 방법으로 서로 다른 테이블에서 저장된 관련된 데이터를 함께 가져와 하나의 결과로 표시합니다. 검색하고 싶은 컬럼이 서로 다른 테이블에 있을 때 사용할 수 있으며, 조인을 사용하면 여러 개의 테이블을 마치 하나의 테이블 인 것처럼 쓸 수 있다.





Inner join – 두테이블에서 공통된 요소들을 통해 결합하는 조인방식

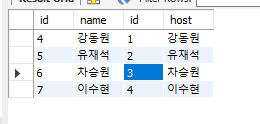
ex) snl\_show에 호스트로 출연한 celeb을 기준으로, celeb table과 snl\_show table을 inner join하여 각 teble의 id와 이름 조회

select celeb.id, celeb.name, snl\_show.id, snl\_show.host

from celeb

inner join snl\_show

on celeb.name = snl\_show.host;



Left Join

두개의 테이블에서 공통영역을 포함해 왼쪽 테이블의 다른 데이터를 포함하는 조인방식

Ex) ex) snl\_show에 호스트로 출연한 celeb을 기준으로, celeb table과 snl\_show table을 left join하여 각 teble의 id와 이름 조회

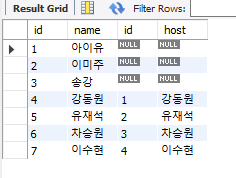
select celeb.id, celeb.name, snl\_show.id, snl\_show.host

from celeb

left join snl\_show

on celeb.name = snl\_show.host;

아이유,이미주,송강은 snl\_show에 출연하지 않았기에 NaN 값으로 조회가 되었다.



Right join

두개의 테이블에서 공통영역을 포함해 왼쪽 테이블의 다른 데이터를 포함하는 조인방식

Ex) ex) snl\_show에 호스트로 출연한 celeb을 기준으로, celeb table과 snl\_show table을 right join하여 각 teble의 id와 이름 조회

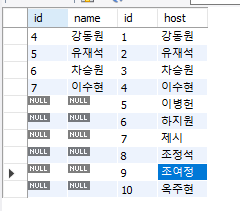
select celeb.id, celeb.name, snl\_show.id, snl\_show.host

from celeb

right join snl\_show

on celeb.name = snl\_show.host;

이병헌,하지원,제시,조정석,조여정,옥주현은 celeb table에 없는 데이터이기에 NaN값으로 조회가 되었다.



FULL OUTER JOIN

두개의 테이블에서 공통영역을 포함하여 양쪽 테이블의 다른영역을 모두 포함하는 조인방식

sql에서 지원하지 않는다.

select celeb.id, celeb.name, snl\_show.id, snl\_show.host

from celeb

full join snl\_show

on celeb.name = snl\_show.host;

mysql은 full outer join 을 지원하지 않기에 left join과 right join을 union하여 만들 수 있다.

select celeb.id, celeb.name, snl\_show.id, snl\_show.host

from celeb

left join snl\_show

on celeb.name = snl\_show.host

union

select celeb.id, celeb.name, snl\_show.id, snl\_show.host

from celeb

right join snl\_show

on celeb.name = snl\_show.host;

self join

inner join과 유사하고 table 간의 공통된 데이터를 조회하며 가장 자주 쓰는 join 방법 이다.

select celeb.id, celeb.name, snl\_show.id, snl\_show.host

from celeb, snl\_show

where celeb.name = snl\_show.host;

문제1 celeb table의 셀럽중, snl\_show로 출연했고 소속사가 안테나인 셀럽의 이름과 직업을 조회

left join(ANSI)으로 먼저 해보면

ex)

select celeb.name, celeb.job\_title

from celeb

left join snl\_show

on celeb.name = snl\_show.host

where celeb.agency = ‘안테나’;

self join(NON - ANSi)

select celeb.name, celeb.job\_title

from celeb, snl\_show

where celeb.name = snl\_show.host and celeb.agency = '안테나';

UNION

여러 개의 Sql문을 합쳐서 하나의 SQL 문으로 만들어주는 방법(주의- 컬럼의 개수가 같아야함)

실습테이터 작성

실습 데이터 작성  
test1과 test2 table를 생성하고 데이터를 입력하여 진행해보자.

test1, test2 table 생성

mysql> create table test1

-> (

-> no int

-> );

Query OK, 0 rows affected (0.02 sec)

mysql> create table test2

-> (

-> no int

-> );

Query OK, 0 rows affected (0.02 sec)

test1, test2 table에 데이터 입력

# test 1

mysql> insert into test1 values (1);

Query OK, 1 row affected (0.00 sec)

mysql> insert into test1 values (2);

Query OK, 1 row affected (0.01 sec)

mysql> insert into test1 values (3);

Query OK, 1 row affected (0.00 sec)

# test2

mysql> insert into test2 values (5);

Query OK, 1 row affected (0.00 sec)

mysql> insert into test2 values (6);

Query OK, 1 row affected (0.00 sec)

mysql> insert into test2 values (3);

Query OK, 1 row affected (0.01 sec)

**UNION**

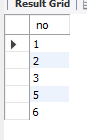
중복된 값을 제외하고 조회

ex) test1, 2 table로 확인  
중복이 되는 데이터 '3'이 중복되어 하나만 조회가 되었다.

mysql> select \* from test1

-> union

-> select \* from test2;



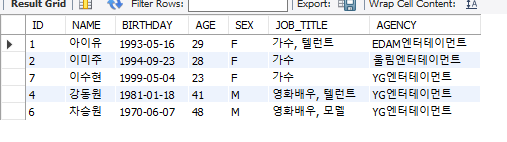
성별이 여자인 데이터를 검색하는 쿼리와 소속사가 YG엔터테이먼트인 데이터를 검색하는 쿼리를

union으로 실행 조회

SELECT \* FROM celeb where sex ='F'

union

SELECT \* FROM celeb where AGENCY = 'YG엔터테이먼트';



중복을 포함(union all 은 중복 포함)

SELECT \* FROM celeb

WHERE sex = 'F' OR agency = 'YG엔터테이먼트';

이수현은 조건을 2번 만족하더라도 ,

데이터 베이슨 실제 테이블을 스캔할 때 중복된 행을 하나만 반환할 가능성이 높음

SELECT distinct \* FROM celeb

WHERE sex = 'F' OR agency = 'YG엔터테이먼트';

정확한 방법임

문제5-2

‘이소미’ 사원의 사원번호, 직위, 사원.부서번호, 부서명을 보이시오

사원테이블에서 부서테이블을 inner join (ansi join) 하고,

None ansi join(self join)을 이용해 보시오

Inner join

SELECT 사원번호, 직위, 사원.부서번호, 부서명

from 사원 inner join 부서

on 사원.부서번호 = 부서.부서번호

where 이름 = '이소미';

none ansi

SELECT 사원번호, 직위, 사원.부서번호, 부서명

from 사원 , 부서

where 사원.부서번호 = 부서.부서번호 and 이름 = '이소미';

고객과 주문을 조인하시오

고객회사들이 주문한 주문건수를 주문건수가 많은 순서대로 보이시오. 이때 고객회사의 정보로는 고객번호, 담당자명, 고객회사명을 보이시오

select 고객.고객번호, 담당자명, 고객회사명, count(\*) as 고객건수

from 고객 inner join 주문

on 고객.고객번호 = 주문.고객번호

group by 고객.고객번호, 담당자명, 고객회사명

order by count(\*) desc;

select 고객.고객번호, 담당자명, 고객회사명, count(\*) as 고객건수

from 고객 , 주문

where 고객.고객번호 = 주문.고객번호

group by 고객.고객번호, 담당자명, 고객회사명

order by count(\*) desc;

고객테이블과 마일리지 등급 조인

고객 테이블에서 담당자가 ‘이은광’인 경우의 고객번호 , 고객회사명, 담당자명, 마일리지와 마일리지 등급을 보이시오

Select 고객번호, 고객회사명, 담당자명, 마일리지, 등급명

From 고객 inner join 마일리지등급

On 마일리지 >= 하한마일리지 and 마일리지 <= 상한마일리지

Where 담당자명 = ‘이은광’;

Select 고객번호, 고객회사명, 담당자명, 마일리지, 등급명

From 고객 inner join 마일리지등급

On 마일리지 between 하한마일리지 and 상한마일리지

Where 담당자명 = ‘이은광’;

Select 고객번호, 고객회사명, 담당자명, 마일리지, 등급명

From 고객 inner join 마일리지등급

On 마일리지 between 하한마일리지 and 상한마일리지

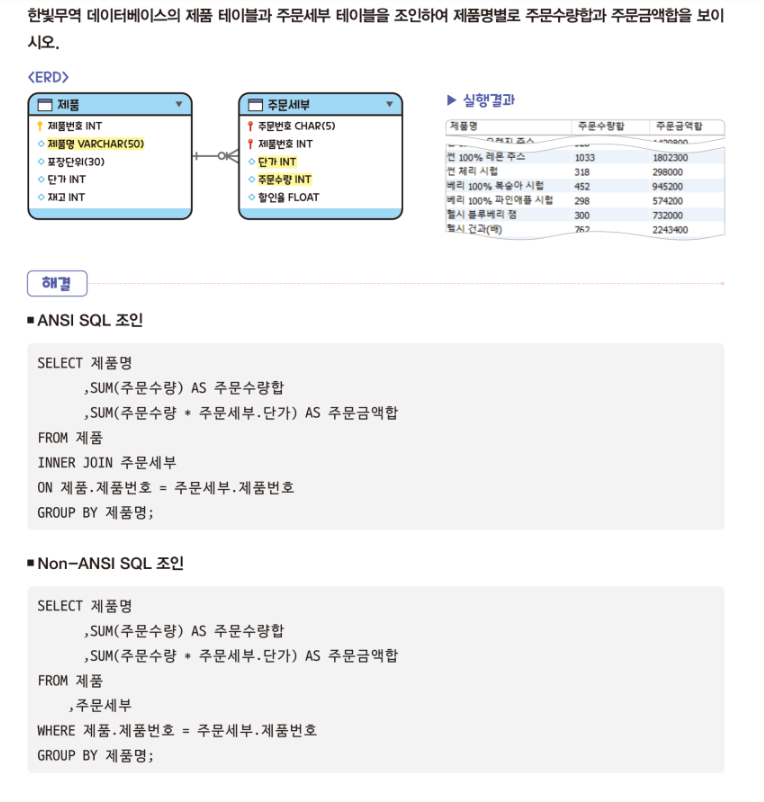
Where 담당자명 = ‘이은광’;

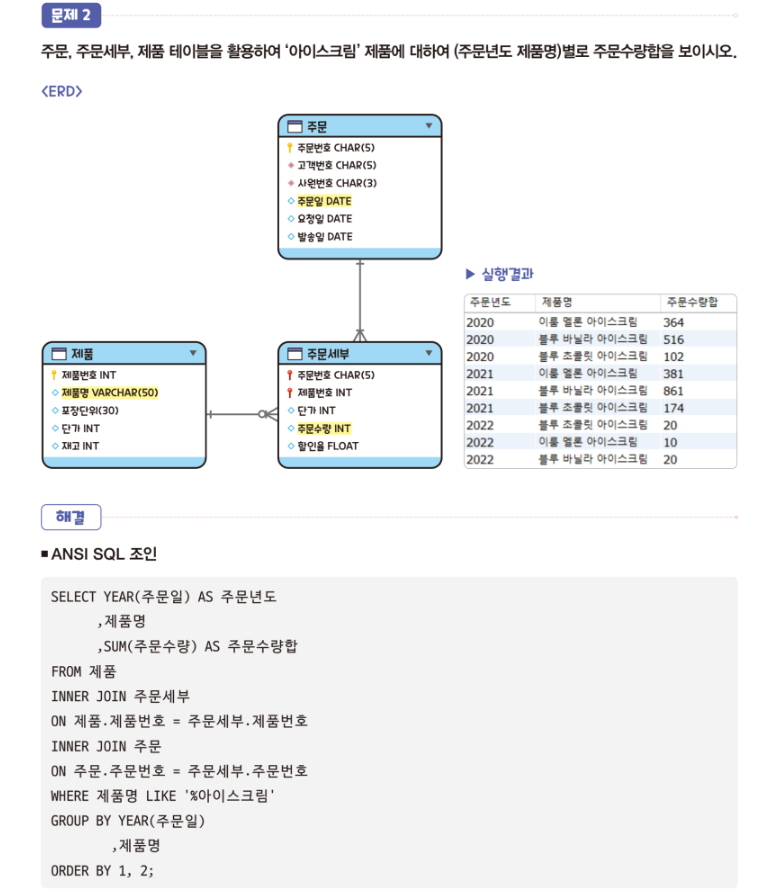
Select 고객번호, 고객회사명, 담당자명, 마일리지, 등급명

From 고객,마일리지등급

Where 마일리지 between 하한마일리지 and 상한마일리지 and

담당자명 = '이은광';





Select YEAR(주문일) as 주문년도

,제품명

,sum(주문수량) as 주문수량합

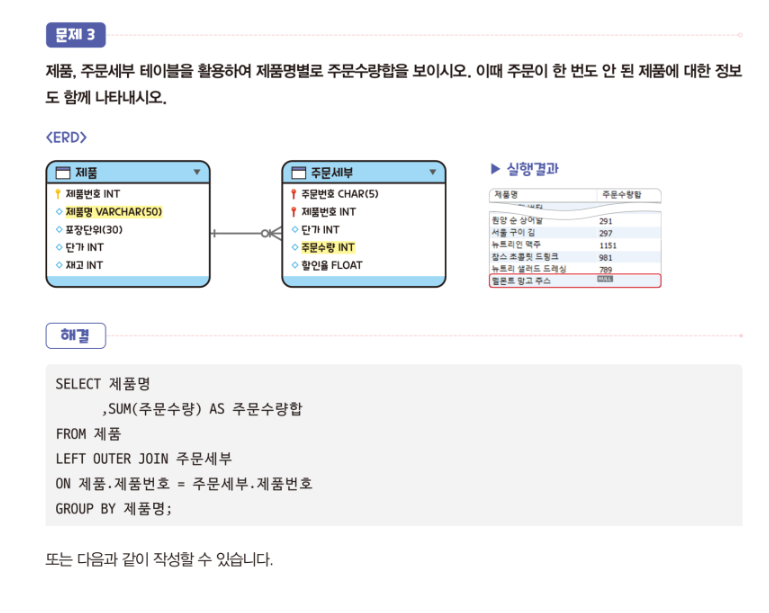
from 제품,주문세부,주문

where 제품.제품번호 = 주문세부.제품번호 and 주문.주문번호 = 주문세부.주문번호

and 제품명 LIKE '%아이스크림'

group by YEAR(주문일) , 제품명

order by 1,2;



서브쿼리

Sql 서브쿼리는 메인쿼리 안에 또 다른 쿼리(서브퀄리) 사용하여 데이터를 추출하거나 처리

데이터 필터링

집계 및 계산

비교 및 검증

중첩된 데이터 추출

EXIsts절 : 특정조건을 만족하는 레코드 존재여부

6-1 최고 마일리지를 보유한 고객 정보를 보이시오

Sum() 과 같은 집계함수는 where절에 직접 사용할 수 없고 반드시 서브쿼리를 통해 결과를 획득

select 고객번호

,고객회사명

,담당자명

,마일리지

from 고객

where 마일리지 = (select max(마일리지) from 고객);

비슷한 방법

select 고객번호

,고객회사명

,담당자명

,마일리지

from 고객

order by 마일리지 desc limit 1;

주문번호 ‘H0250’을 주문한 고객에 대해 고객회사명과 담당자명을 inner join 과 서브쿼리로 구해보자

select 고객회사명

,담당자명

from 고객

inner join 주문

on 고객.고객번호 = 주문.고객번호

where 주문번호 = 'H0250';

서브쿼리

select 고객회사명

,담당자명

from 고객

where 고객번호 = (select 고객번호 from 주문 where 주문번호 = 'H0250' );

6-3 ‘부산광역시’ 고객의 최소 마일리지보다 더 큰 마일리지를 가진 고객 정보를 보이시오

select 담당자명

,고객회사명

,마일리지

from 고객

where 마일리지 > (select min(마일리지) from 고객 where 도시='부산광역시');

복수행 서브 쿼리

복수 행 서브쿼리는 서브쿼리의 결과가 여러 행이 나오는 쿼리

(IN, ALL , ANY, SOME , EXISTS) 와 같은 복수행 비교연산자를 사용하여 메인쿼리와 서브쿼리연결

IN – 서브쿼리 결과에 해당 값이 포함되어 있는지

ALL – 서브쿼리의 결과 모든 값과 비교해서 조건을 만족해야함

ANY/SOME 서브쿼리 결과 어는 하나라도 조건을 만족하면 참

EXISTS 서브쿼리의 결과가 존재하기만 하면 TRUE 보통 WHERE 절 내부 존재 여부 확인에 사용

‘부산광역시’ 고객이 주문한 주문건수를 보이시오

SELECT COUNT(\*) AS 주문건수

FROM 주문

where 고객번호 IN (SELECT 고객번호 from 고객 WHERE 도시='부산광역시');

6-5) ‘부산광역시’ 전체 고객의 마일리지보다 마일리지가 큰 고객의 정보를 보이시오

SELECT 담당자명,고객회사명,마일리지

FROM 고객

where 마일리지 > ANY(SELECT 마일리지 FROM 고객 WHERE 도시='부산광역시');

ANY와 ALL 차이점을 직접 넣어서 확인해보기

6-6) 각지역의 어느 평균 마일리지보다도 마일리지가 큰 고객의 정보를 보이시오

SELECT 담당자명,고객회사명,마일리지

FROM 고객

where 마일리지 > ALL(SELECT AVG(마일리지) FROM 고객 group by 지역);

모든 지역 평균 마일리지보다 높은 고객을 찾으려고 GROUP BY

6-7) 한 번이라도 주문한 적이 있는 고객의 정보를 보이시오

SELECT 고객번호, 고객회사명

FROM 고객

where exists(select \* from 주문 where 고객번호 = 고객.고객번호);

SELECT distinct 고객.고객번호, 고객.고객회사명

FROM 고객

inner JOIN 주문

ON 고객.고객번호 = 주문.고객번호;

EXISTS 쿼리는 INNER JOIN + DISTINCT로 동일한 결과를 얻을 수 있다.

조건절에서 사용하는 서브쿼리

HAVING : 그룹핑 후 조건을 거는 것. 집계값 필터링

WHERE : 그룹핑 전에 조건을 거는 것. 일반행 필터링

6-8 고객 전체의 평균마일리지 보다 평균마일리지가 큰 토시에 대해 도시명과 도시의 평균말일리지를 보이시오

SELECT 도시, AVG(마일리지) AS 평균마일리지

FROM 고객

group by 도시

having AVG(마일리지) > (select AVG(마일리지) FROM 고객);

GROUP BY 도시 – 고객테이블을 도시별로 그룹으로 나눕니다.

AVG(마일리지) - 각 도시 그룹안에서 마일리지 평균을 계산합니다.

having AVG(마일리지) – 계산된 도시별 평균 마일리지를 기준으로 조건 을 거는 것

FROM 절에서 사용하는 서브쿼리

FROM 절에서도 서브쿼리 사용할 수 있다. 이를 인라인 뷰라고 합니다. 인라인 뷰에는 반드시 별명을 지정해주어야 하며, 인라인 뷰의 별명은 테이블명처럼 사용할 수 있습니다.

6-9) 담당자명, 고객회사명, 마일리지, 도시, 해당도시의 평균 마일리지를 보이시오. 그리고 고객이 위치하는 도시의 평균 마일리지와 각 고객의 마일리지 간의 차이도 함께 보이시오

SELECT 담당자명,고객회사명,마일리지,고객.도시,도시\_평균마일리지, 도시\_평균마일리지-마일리지 AS 차이

FROM 고객

,(select 도시, AVG(마일리지) AS 도시\_평균마일리지 FROM 고객 group by 도시) AS 도시별요약

WHERE 고객.도시 = 도시별요약.도시;

SELECT 담당자명,고객회사명,마일리지,고객.도시,도시\_평균마일리지, 도시\_평균마일리지-마일리지 AS 차이

FROM 고객

INNER JOIN

(select 도시, AVG(마일리지) AS 도시\_평균마일리지 FROM 고객 group by 도시) AS 도시별요약

ON 고객.도시 = 도시별요약.도시;

SELECT 절에서 사용하는 서브쿼리

스칼라 서브쿼리 정확히 한 개의 행에 대해 한 개의 값을 반환

6-10) 고객번호, 담당자명과 고객의 최종 주문일을 보이시오.

SELECT 고객번호, 담당자명

,(select MAX(주문일) FROM 주문 WHERE 주문.고객번호 = 고객.고객번호) AS 최종주문일

from 고객;

SELECT 고객.고객번호, 고객.담당자명,MAX(주문.주문일) AS 최중주문일

from 고객 LEFT JOIN 주문

ON 고객.고객번호 = 주문.고객번호

group by 고객.고객번호, 고객.담당자명;

SELECT 담당자명,고객회사명,마일리지,고객.도시,도시\_평균마일리지, 도시\_평균마일리지-마일리지 AS 차이

FROM 고객

,(select 도시, AVG(마일리지) AS 도시\_평균마일리지 FROM 고객 group by 도시) AS 도시별요약

WHERE 고객.도시 = 도시별요약.도시;

위에 처럼 구하는 것이 쉽지 않음

CTE 는 뷰와 마찬가지로 파생 테이블 처럼 사용가능

형식

WITH CTE명 AS

(

SELECT

)

6-9)를 CTE 사용하여 작성

WITH 도시별요약 AS

(

select 도시

, AVG(마일리지) AS 도시\_평균마일리지

FROM 고객

group by 도시

)

SELECT 담당자명,고객회사명,마일리지,고객.도시,도시\_평균마일리지, 도시\_평균마일리지-마일리지 AS 차이

FROM 고객, 도시별요약

where 고객.도시 = 도시별요약.도시;

상관 서브쿼리

메인쿼리와 서브쿼리 간의 상관관계를 포함하는 형태의 쿼리

사원테이블에서 사원번호, 사원의 이름, 상사의 사원번호, 상사의 이름을 보이시오

SELECT 사원.사원번호, 사원.이름,사원.상사번호, 상사.이름 AS 상사이름

from 사원 LEFT JOIN 사원 AS 상사 ON 사원.상사번호 = 상사.사원번호;

사원 테이블을 자기자신과 조인할 때, 별칭을 상사 로 주었다.

LEFT JOIN 을 사용해 사상번호가 없는 사원도 함께 표시

사원.상사번호 = 상사.사원번호; 조건을 통해 상사의 이름을 가져옵니다.

다중 컬럼 서브 쿼리

여러 개의 컬럼을 사용하여 다중비교하는 쿼리

각도시마다 최고 마일리지를 보유한 고개의 정보를 보이시오

SELECT 도시, 담당자명, 고객회사명 , 마일리지

FROM 고객

where (도시, 마일리지) IN (SELECT 도시, MAX(마일리지) FROM 고객 group by 도시);