create table instructor(name type)

primary key(A)

foreign key(A) references R

insert into instructor values(a, b, c) / insert into instructor query

delete from instructor

update instructor set result where condition

update instructor set salary = case when ~ then ~ when ~ then ~ else ~ end

drop table R

alter table R add A D

alter table R drop A

select: projection, from: Cartesian, where: selection, as: rename

like ‘%, \_’, escape: \

order by name (default: asc) asc/desc

between and

union, intersect, except (all)

is null, is not null

avg, min, max, sum, count

group by A

having P (after group by)

subquery select: single value, from, any valid, where B operation subquery

in, not in, some, all

exists, not exists, unique

with max\_budget(value) as

(select max(budget)

from department)

join은 from clause에서 사용

natural join

R join S using (A)

R join S on P

join types: inner join, left outer join, right outer join, full outer join

join conditions: natural, on P, using (A)

create view V as query

constraints: not null, unique, check(P)

on delete cascade, on update cascade (set null, set default) -> foreign key의 constraint

create assertion assertion-name check(P)

blob, clob

create type Dollars as numeric(12,2) final

create domain person\_name char(20) not null -> constrain 가질 수 있음

create domain degree\_level varchar(10)

constraint degree\_level\_test

check(value in (~~~))

create index studentID\_index on student(ID)

grant select on department to Amit, Satoshi

revoke select on student from Amit

create role instructor

grant select on takes to instructor

grant instructor to Amit