Agenda

- Subqueries
- IN
- FROM
- ALL , ANY
- Correleated subqueries
- EXISTS

Students

id , name, psp, b-id

Q Find all students whose psp is greater than max psp of batch 2.

x - max psp

```
for all students (s):
    if s.batch = 2
        x = max(x, s.psp);
```

```
ans = []
for all students (s)
    if s.psp > x
        ans.append(s);
```

Q Find all the students whose psp is greater than student with id = 18.

```
select s.id, s.name
from students s
join students t
    on t.id = 18 AND s.psp > t.psp;
```

① get the psp of student with id = 18

x ← Select psp from students
where id = 18;

② compare every student psp with x

Select s.id from students
where psp > x;

Select s.id from students
where psp > (Select psp from students
where id = 18);

Q Find all students whose psp is greater than max psp of batch 2.

I

(I) Find max psp of batch 2

```
x ← Select MAX(psp) from students
    where b_id = 2;
```

(II) compare the psp of all students with x

```
Select s_id from students
    where psp > (x);
```

```
Select s_id from students
    where psp > (Select MAX(psp) from students
                 where b_id = 2);
```

Tradeoff

```
select s.id from students
where psp > (select psp from students
             where id = 18);
```

```
for s in students:
    psp
    for t in students:
        if t.id = 18
            psp = t.psp
    if s.psp > psp
        ans.insert(s);
```

N^2

join

Subquery

row	col	
1	1	single value
1	N	single row
N	1	single col
N	M	Table

users

id	name	is_stud	is_ta

Tell the name of the students which are also the name of TA's

id	name	is_stud	is_ta
1	Akshay	1	0
2	Aman	1	0
3	Akash	1	1
4	Aman	0	1
5	Akshay	0	1
6	Mohit	1	1
7	Mahesh	1	0

Akshay
Aman
Akash
Mohit

stud	
id	Name

TA	
id	name

```
select distinct s.name
from students s
join TA t
on s.name = t.name;
```

users

id	name	is_stud	is_ta

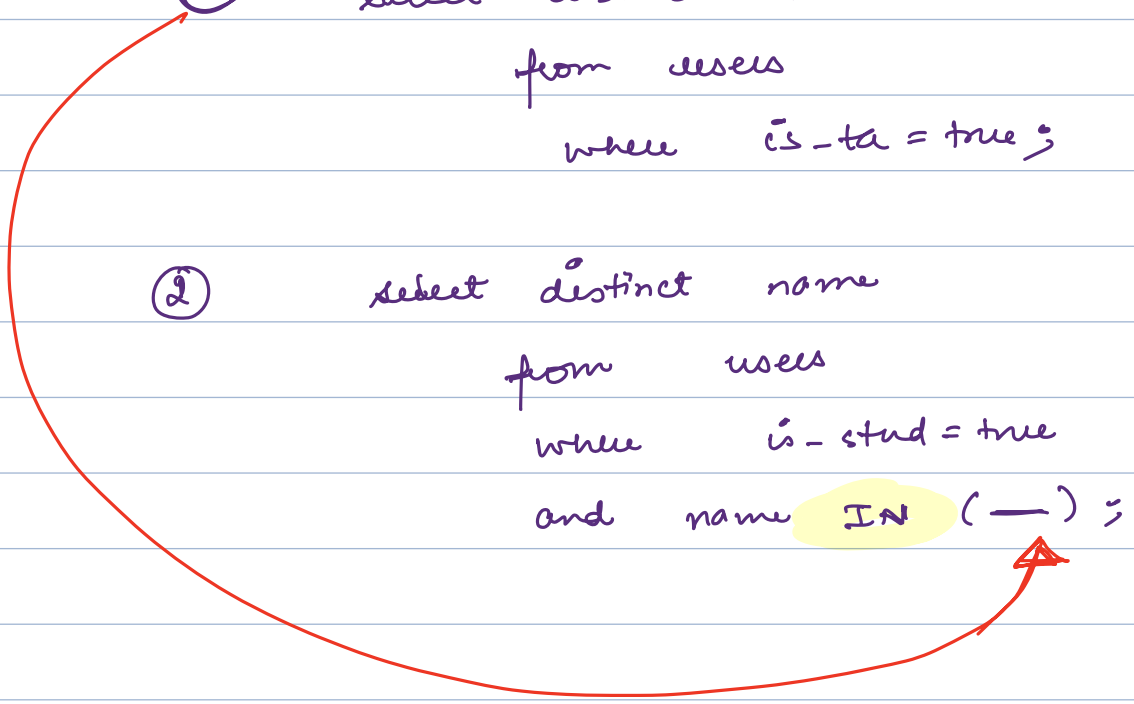
```
select distinct s.name
from users s
join users t
on s.is_stud = true
and t.is_ta = true
and s.name = t.name;
```

Tell the name of the students which are also the name of TA's

1. get the name of TA's = [ ]
2. get students whose name is present

1. select distinct name
   from users
   where is_ta = true;

2. select distinct name
   from users
   where is_stud = true
   and name IN (—) ;

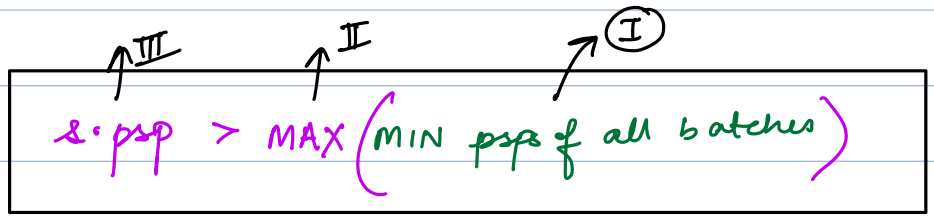


Q Find all students whose psp is not less than smallest psp of any batch.

↓

students whose psp is greater than minimum psps of every batch.

	min	stud	
1 →	62	75	✓
2 →	58	67	X
3 →	72		
4 →	66		



(I) select MIN(psp) from students group by b.id;

(II) select max(psp) from x;

(III) select s.id from student s where s.psp > y;

```
select s.id from
student s
where s.psp >
(select max(psp) from
(select MIN(psp) as psp from
students
group by b.id) minpsp
);
```

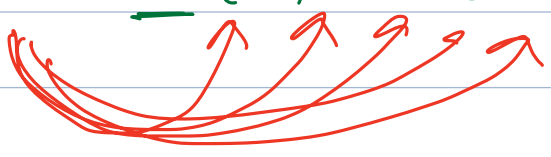
minpsp → alias for the subquery table

WHERE _ ①
IN _ col/row
FROM _ table .

x > —, —, —, —

x is greater than ALL of mins .

True ⇐ x > ALL(10, 20, 40, 60, 90)



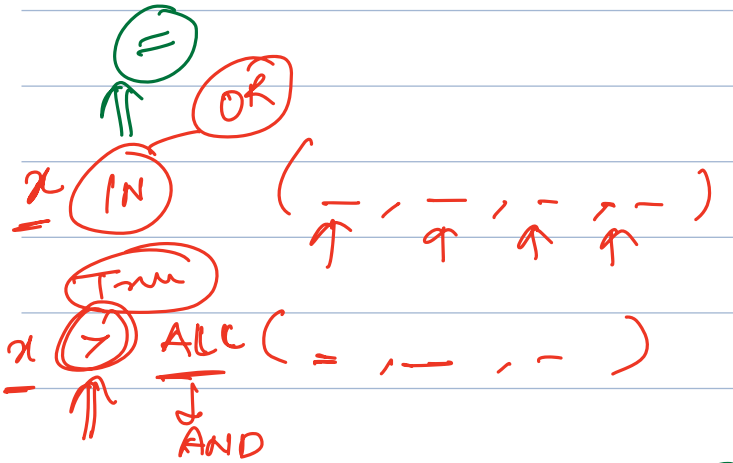
select s.sid

from students s

where s.psp > ALL (select MIN(psp)

from students

group by b_id) ;



OR

x > ANY (_ _ _ _)

- Get all students whose psp is greater than AVG PSP of their batch

(I) Get all students whose psp is > x;

(II) x = AVG psp of student's batch

(I) select * from students where psp > x;

(II) select AVG(psp) from student where b_id = y;

```
select * from
    students s
where psp > ( select AVG (psp)
              from students
              where b_id = s.b_id);
```

correlated subquery

id	name	psp	b_id
1	X	40	1 → 26. (80/3)
2	Y	60	2
3	Z	20	1
4	A	30	2
5	B	10	3
6	C	20	1
7	D	30	2
8	E	40	3

EXISTS

Get all the TA's who are also students

students

id	name	psp

TA

id	name	st_id
4		1

```
select * from TA
where st_id is NOT NULL;
```

Get all the students who are also the TA's

```
select * from students
where
id IN (select st_id from TA
where st_id is NOT NULL);
```

```
Select *
  from students s
  where
      EXISTS ( Select st_id from TA
               where st_id = s.id);
```

EXISTS → any number of rows ⇒ True
Even if row = 0 ⇒ False

- SUBQUERIES
- WHERE
- IN
- FROM
- ALL/ANY
- CORELATED
- EXISTS

Doubts

S C E

Id, name, co id nam c, seq Id, am, c_id, CON .