

Subqueries

Subqueries → Intuitive way to write SQL queries.

Agenda:



Subqueries

- IN
- ANY
- ALL
- Correlated subqueries
- EXISTS

→ Subqueries in FROM

→ Subqueries in WHERE

Q Given a Students table, find all the students who have Psp > the maximum fsp of students of batch 2

Students			
id	name	p,sp	batch-id
-	-	-	-

Code:

students = []

algo find max Psp of batch 2 \Rightarrow ans

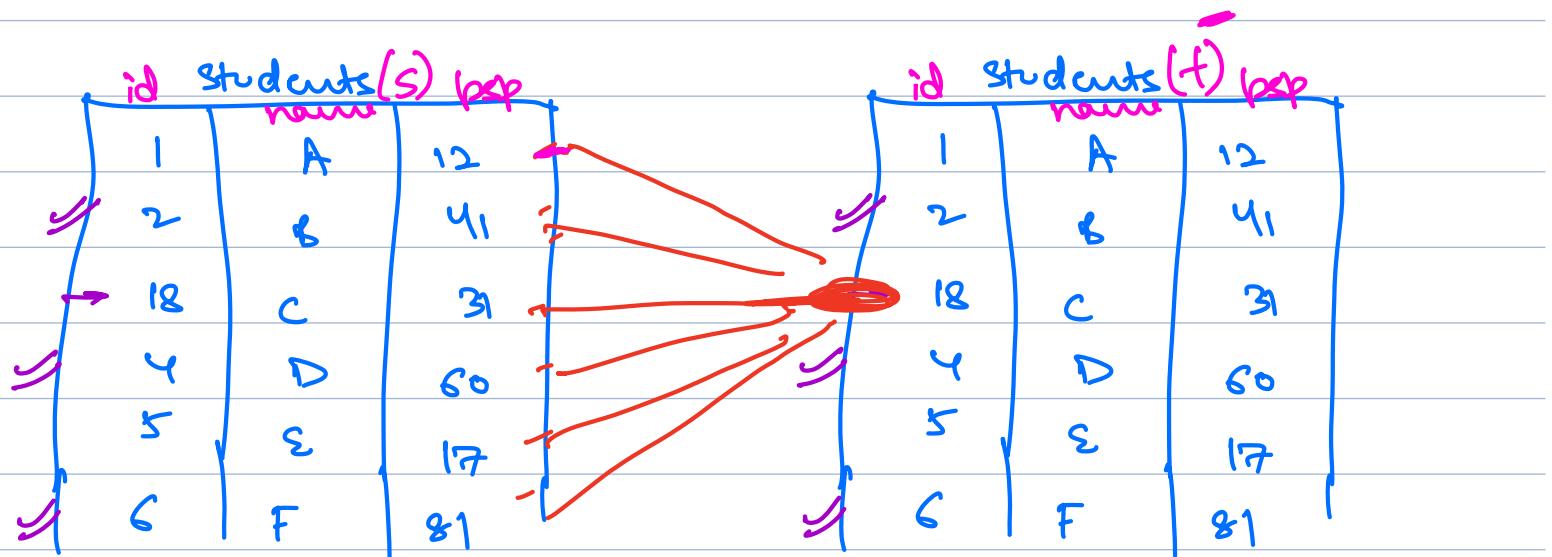
for all students (c) :

if c.psp \geq ans:

ans. add (c)

returns ans;

Q find all the students whose psp > psp of the student with id = 18;



Select

from Students s

join Students t

on t.id = 18

and s.psp > t.psp ;

In reality, we prefer

① Break problem into part

② Solve smaller problems and use these results to solve bigger problems.



Subqueries → Break a problem into smaller problems and combine their result to get complete answer

Most of the time problems that we solve via subqueries can also be solved via some other smart trick.



Subqueries make our queries easier to understand & create.

Q find all the students whose psp > psp of the student with id = 18;

- ② ∈ ① find the psp of the student with id = 18.
② find all the students with psp > n.

① Select psp
from student
where id = 18 ↗

②

② Select *
from students
where psp > n

Select *

from students

where psp > (select psp

from student

where id = 10) ;

→ Readable

→ Intuitive

* subqueries should be always enclosed in parenthesis.

Trade off of subqueries → bad performance

Select *

from students

where psp > 70

Students = []

ans = []

for students s: Students:

if s.psp > 70

ans.add(s)



without
subquery

for student s: students :

If $s.bsp \rightarrow$ $\text{ans}.\text{add}(s)$

$\text{ans} = []$

for student o: Students

if $o.id = 18$

$\text{ans}.\text{add}(o);$

$O(N)$

if (_____)

return _____

↓

$O(N^2)$

Q Given a Students table, find all the students
who have $bsp >$ the maximum bsp of students
of batch 2

Select *

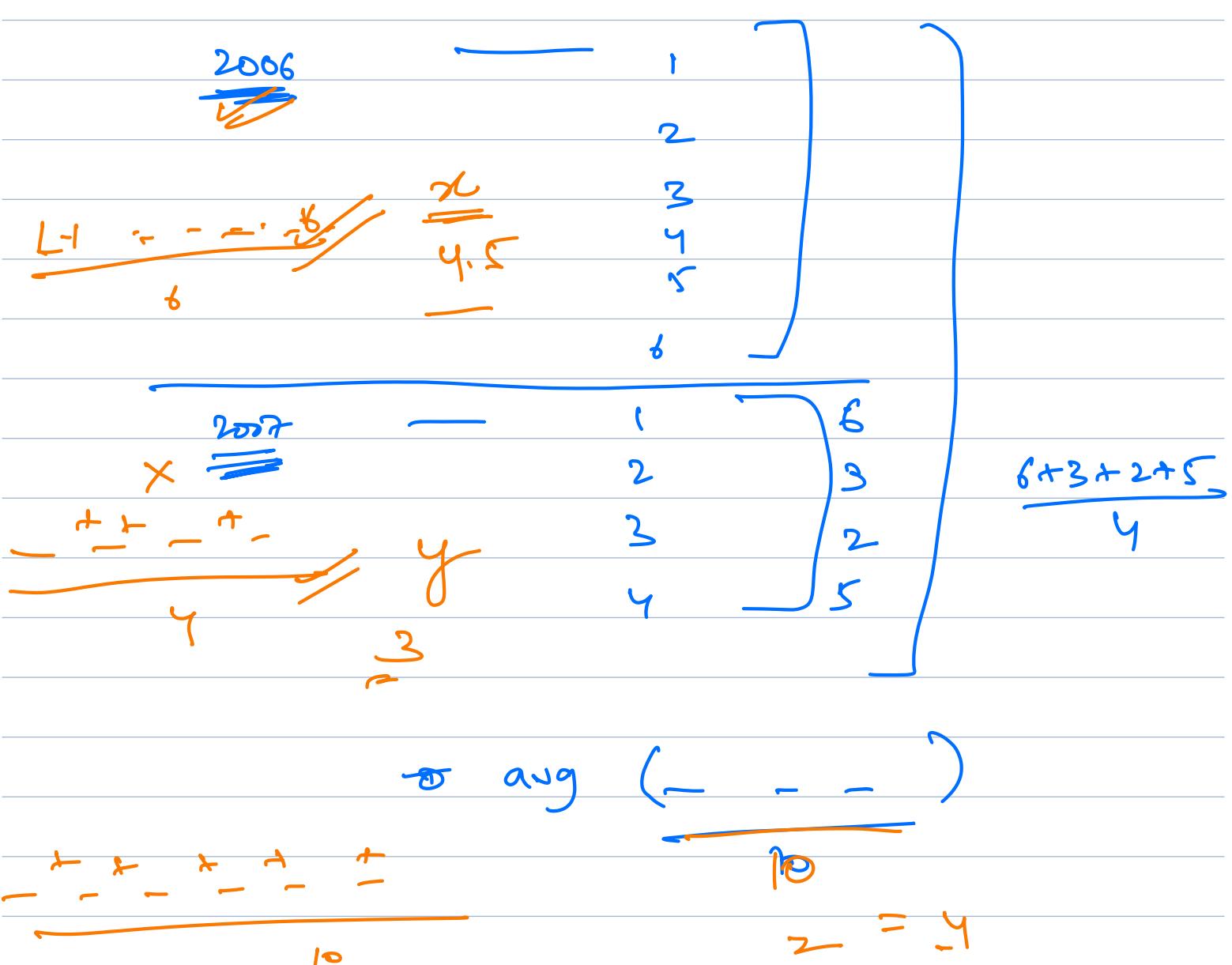
from Students

where $bsp > (\text{select max}(bsp)$

from Students

where $\text{batch_id} = 2$);

Q Tell all years where the average of the rental_rate of films of that year was greater than global average of rental rating.



① find global average



Select avg(rental_rate)
from film;

② find avg of every year.

Select release-year , avg (rental-date)
from film
group by release-year

③ Get filtered groups) years

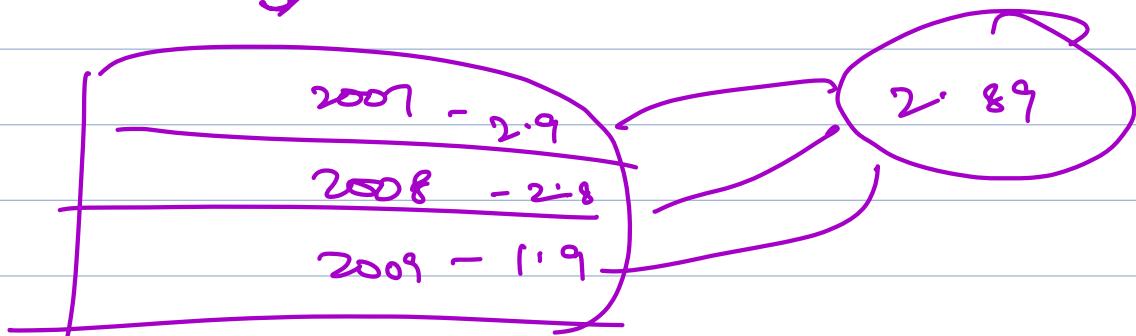
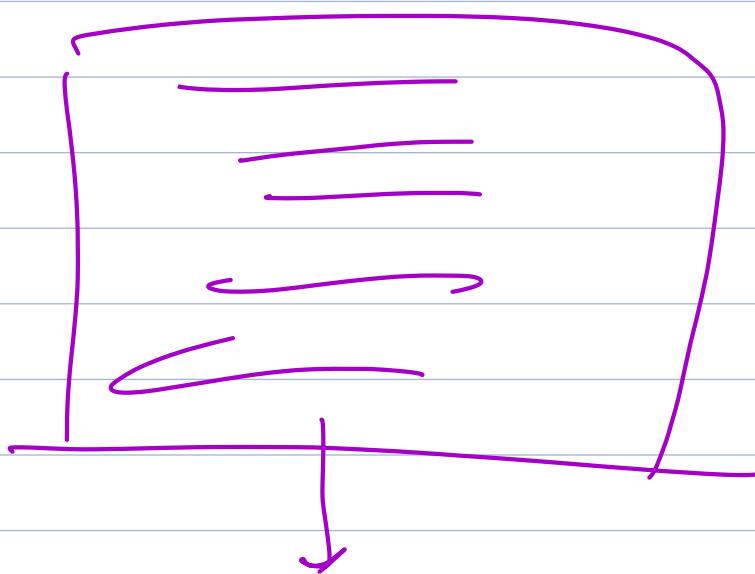
~~find~~

Select release-year , avg (rental-date)

from film

group by release-year

having avg(rental-date) > (select avg(rental-date)
from film



Break till 8:25 AM

→ Subquery we wrote till now just gave me a single value in o/p.

→ A subquery can give any value of

$\frac{\text{cols}}{\text{rows}}$
row →

$\begin{array}{c|c} \infty & C \\ 1 & 1 \rightarrow \text{single value} \\ 1 & m \rightarrow \text{single row} \\ m & 1 \rightarrow \text{single column} \\ m & m \rightarrow \text{table} \end{array}$

Q

Users				
id	name	is_stud	is_ta	
1	Kishan	T	F	
2	Vijayal	F	T	
3	Kishan	F	T	
	Lavanya	T	F	

Tell the names of students that are also names of TA.

with joins/ no subqueries

Select

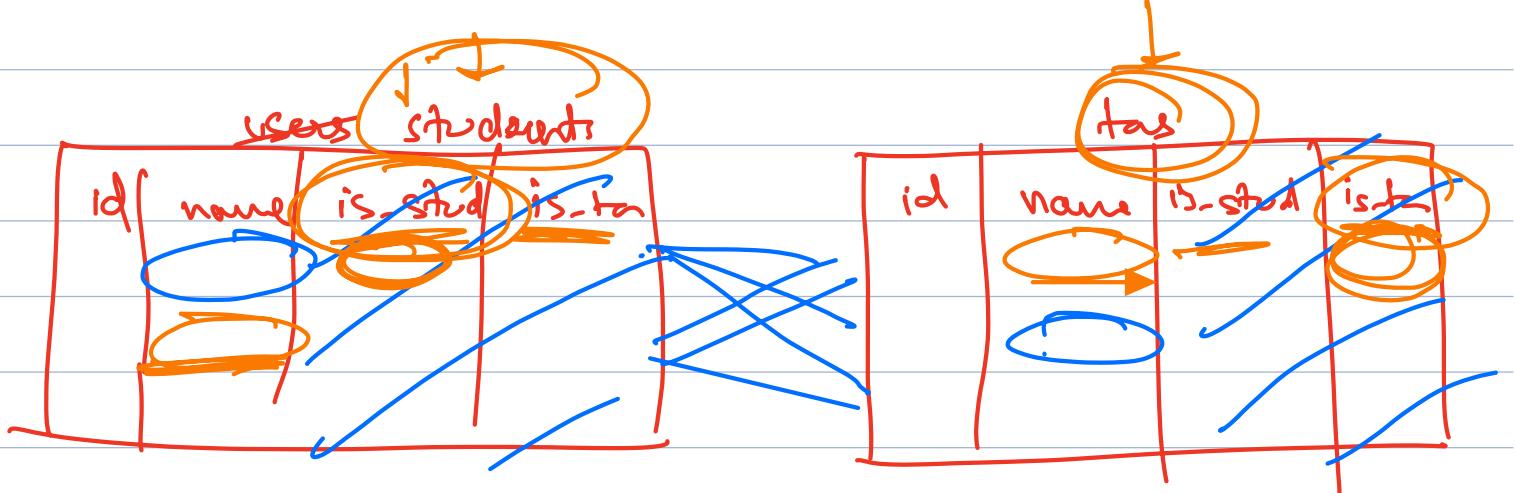
from users s

join users t

on s.name = t.name and

s.is-stud = true

and t.is-ta = true



select

from students s

join tas t

where s.name = t.name

with subqueries.

- ① Get names of all the TAs = []
- ② Get students whose name is ↗

select name

from users

where is-ta = true

```
select *  
from users  
where is_stud = true  
and name IN ( _____ );
```

final

```
select * name  
from users  
where is_stud = true  
and name IN ( select name  
from users  
where is_ta = true )
```

Q Get all the students where fsp is not less than smallest fsp of any batch?

batch	<u>min</u>
1	20
2	30
3	39

<u>student</u>	
= 18	X
= 32	X
= 42	-

all students whose fsp
is greater than max (fsp)
of name of batches

① select min(psip)
from students
group by batch-id;

}

list
x

② select max(bsp)
from x j

y

→

③ select *
from students
where bsp > y ;

find.

select *

from students



where bsp > (select max(bsp))

from (select min(psip)

from students

group by batch-id ;

minpsip

)

);

Subquery in FROM

- We can have subquery in from clause
- this subquery's output is considered a table in itself.
- upon which you can write any other query.
- You should name a subquery in from clause MANDATORY.

All

$x > (- - - -)$

Select

from students

where $\underline{psb} > \text{ALL} (\text{select min(psb)}$
from student
group by batch-id))

Compare LHS of query with every value of the RHS.

If all of them return true, only then ALL will return true.

ANY → Compare LHS with all the values of RHS
if any of them returns true , ANY gives true .

Correlated Subqueries

Q Get all the students whose bsp is greater than average bsp of their batch.

① Get students with $\text{bsp} > \textcircled{x}$

② $\textcircled{x} = \text{Avg bsp of student's batch}$

Select *
from student
where bsp > x

select avg(bsp)
from student
where batch_id = y

Select *

from student s

where bsp > (select avg(bsp)

from student

where batch-id = s.batch-id);

=

for s : students

for v : in students.

v.batch = s.batch-id

EXISTS

Q Get all students who are also TA

Students

id	name	bsp
1	Vineeth	70

tas

id	name	st-id
1	Vineeth	1
2	Ujjwal	null

Select st-id

from tas

where st-id is NOT NULL

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

EXIST

Select *
 from students s
 where EXISTS (
 select st_id
 from ta
 where ta.st_id = s.id)

for each row of students:

- ① it will run subquery
- ② If the subquery returns any no. of rows > 0
it will return true.

Q

Students	
id	name
1	ujjwal
2	sarali
<u>3</u>	bhaarat

mentor_session		
ser_id	stu_id	mentor_id
1	<u>1</u>	—
2	2	—
3	3	—
4	1	—

Get all students that have take a mentor session.

select *

from students

where id IN (



select stud_id

from mentor_session);

IM

select *

from students s

where EXISTS (

<

select *

from mentor_session

where st_id = s.id).

vijual
braester