Subqueries

TABLE OF CONTENTS

Notes

17th Hard day challenge :

1. Assignments + Revision
2. Backlog (Assignments of prev. session)
3. Additional Questions

Subqueries

- This is very intuitive way of writing queries.
- Breakdown bigger problems into smaller ones.
- Will use result of smaller problems to get final answer.

Students

id	name	b_id	psp
1	John	Null	80
2	Jane	1	90
3	Jim	2	85
4	Jenny	3	95
5	Jack	2	78

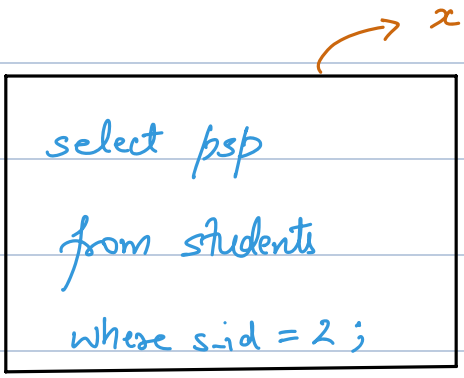
psp of s_id (2) = 90 → x

< *Question* > : Find all the students having psp > (psp of s_id_2).

Step-1 : Find psp of student with s_id = 2 ⟶ (x)

Step-2 : Find all students with psp > x

Query-1 : Code for x

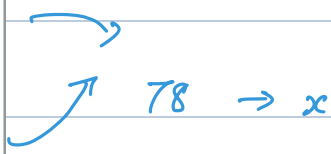


Query-2 : Main query

```
select *
from students
 where psp > (
        select psp
        from students
        where s_id = 2 ) ;
```

Students

id	name	b_id	psp
1	John	Null	80
2	Jane	1	90
3	Jim	2	85
4	Jenny	3	95
5	Jack	2	78



< *Question* > : Find data of all the students having psp > (min psp of b_id_2).

Step-1 : Find min psp of b_id = 2 ⟶ (x)

Step-2 : Find all students having psp > x

Query-1 : Query to find x

```
select min(psp)
from students
where b_id = 2;
```

x

Query-2 : Final query

```
select *
from students
where psp > (
    select min(psp)
    from students
    where b_id = 2);
```

TC = $O(N^2)$

Films

name	release_year	rental_rate
Hera Pheri	2008	250
Robot	2009	300
Welcome	2011	420
Bahubali	2016	250

Films

name	release_year	rental_rate
Hera Pheri	2008	250
Robot	2009	300
Welcome	2011	420
Bahubali	2016	250

< *Question* > : Find all years where average rental_rate > (The global average rental rate).

Subqueries and IN Clause

Rows	Column	Output
1	1	Single value
1	m	Single row
m	1	Single col
m	m	Table

Users →

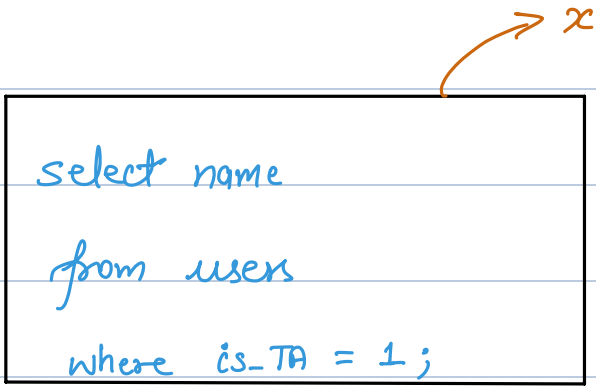
id	Name	is_student	is_TA
1	Gaurav	1	1
2	Rohit	1	0
3	Pravallika	1	0
4	Krish Na	1	1
5	Nandini	1	0
6	Rohit	0	1

< *Question* > : Get names of all the students who are (names of TA as well)

Step-1 : Get names of all TA. (Gaurav, Krishna, Rohit)

Step-2 : Check whether name of a student is a name of TA as well.

< / > Code -1



< / > Code -2

```
select name
from users
where is_student = 1 and
      name IN ( select name
                from users
                where is_TA = 1 );
```

Subqueries inside from clause

< *Question* > : Find data of all the students where psp > (min(psp) among avg(psp) of every batch).

Students

id	name	b_id	psp
1	John	Null	80
2	Jane	1	90
3	Jim	2	85
4	Jenny	3	95
5	Jack	2	78

1 → avg (psp) = 90
2 → avg (psp) = 81.5 → min
3 → avg (psp) = 95

Explanation :

1. Find avg(psp) of every batch. → x
2. Find min(psp) among x → y
3. Find all students with psp > y.

Note : Whenever we use subquery inside from clause, always use aliasing.

All and Any

Note: In operator is just to check membership.

< *Question* > : Find data of all the students where psp >= (min(psp) of student in every batch).

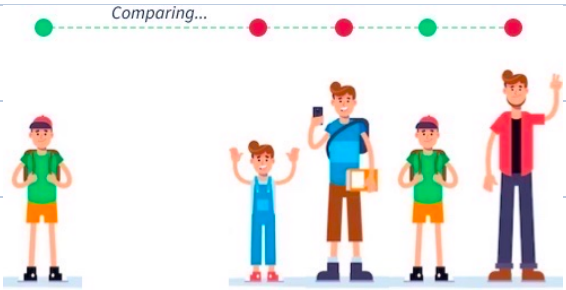
Students

id	name	b_id	psp
1	John	~~Null~~ 1	80
2	Jane	1	90
3	Jim	2	85
4	Jenny	3	95
5	Jack	2	78

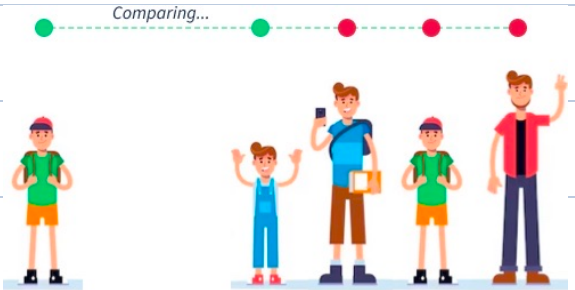
1 → 80

2 → 78

3 → 95



1. All



2. Any

Co-related Subqueries

< *Question* > : Find data of all the students where psp > (avg(psp) of their batch).

Students

id	name	b_id	psp
1	John	~~Null~~ 1	80
2	Jane	1	90
3	Jim	2	85
4	Jenny	3	95
5	Jack	2	78

' For Jane we need avg psp of batch 1. For Jim batch 2..... '

Conclusion : For every batch_id we need avg psp of that batch to compare the value.

Exists

Students

id	name	psp
1	Rahul	98
2	Rohit	95
3	Tarun	88

Tas

id	Name	student_id
1	Rahul	1
2	Mohit	Null
3	Sameer	Null

< Question > : Find all the students who are also TAs

Example

1. *' Does RED ball exists here? '*

2. *' And here ? '*

3. *' And what about in this one? '*

- **Exist is used when output depends on just having > 0 values.**
- **Exist gives us faster results than IN clause.**

Now, we will write query using exist.

```
select *
from students s
where exists (
        select s_id
        from Tas
        where Tas.s_id = s.id );
```