1. Good Evening
2. We begin at 9:08 pm
3. Subqueries & Views.

Agenda

1. Subqueries
2. Views
3. IF, CASE, IFNULL, COALESCE.
4. Deadlock.
5. Full-text search.

Subqueries

1. Find all students who have psp greater than psp of student with id = 18.

Students

id	name	b_id	psp
1	A	1	90
2	B	1	95
3	C	2	75
4	D	2	85
⋮			
18	K	4	80

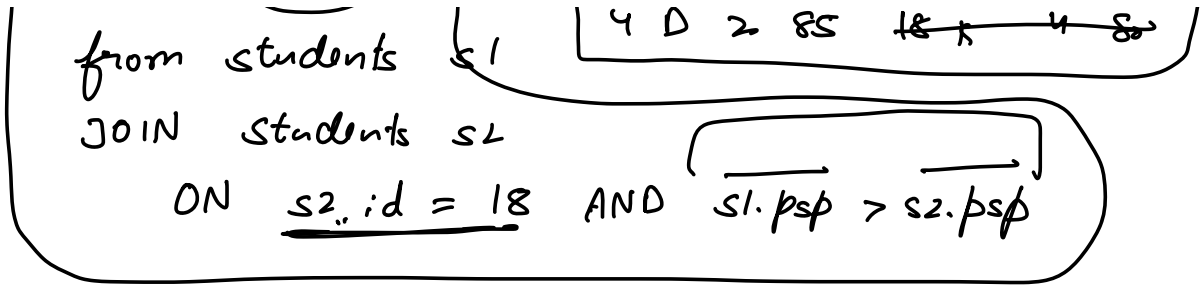
S1

1	A	1	90
2	B	1	95
3	C	2	75
4	D	2	85
18	K	4	80

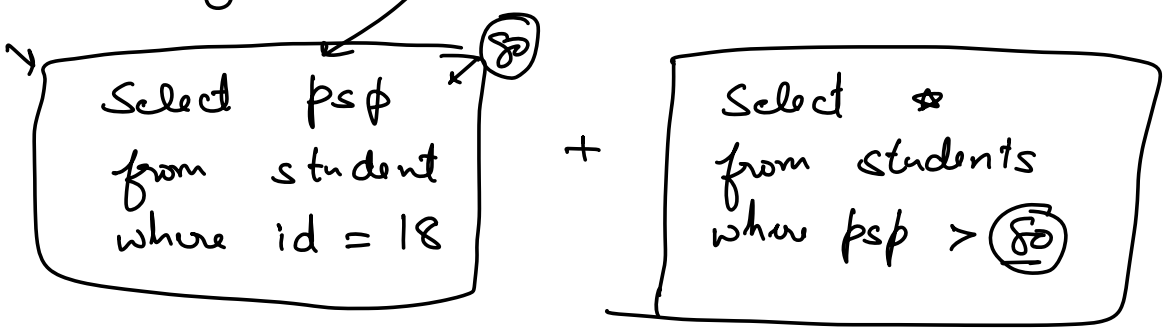
S2

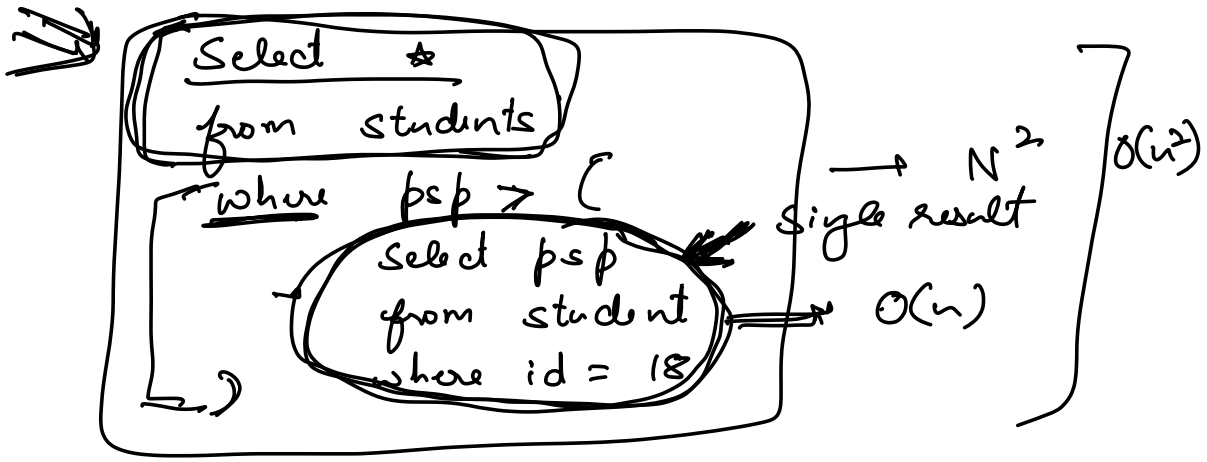
Joins

Select (s1.*)



Subquery = Step1 + Step2





Joins > SQ

1	A	1	95
2	B	1	85
3	C	2	75
4	D	2	87
18	E	2	80

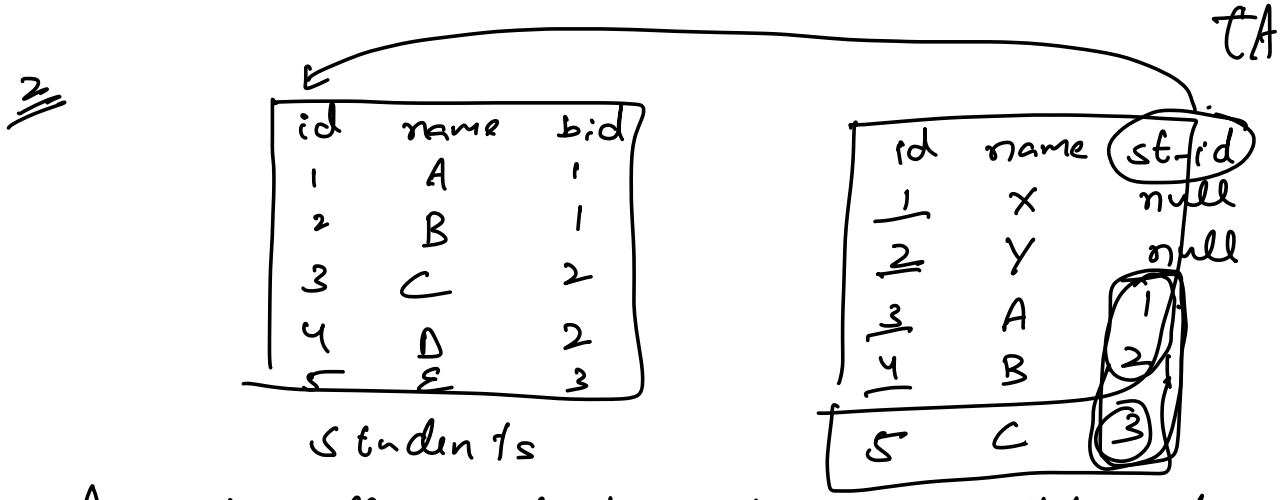
→ DECLARE @tpsp INT;

→ SELECT @tpsp = psp
from students

where id = 18;

→ Select *
from students
where psp > avg(psp)

2.



* Find all students who are TAs also

↳ JOIN

```
Select s.*
from students s
 JOIN  TAs  t
   ON  s.id = t.st_id
```

AND t.st_id IS NOT NULL

X != =

NULL = NULL → false
NULL IS NULL → true

```
Select st_id
from TAs
```

```
(
where st_id IS NOT NULL;
)
```

IN CLAUSE

```
Select *
from students
where id IN (1, 2, 3)
```

```
Select *
from students
where id IN (
    Select st.id
    from TAs
    where st_id IS NOT NULL;
)
```

1, 2, 3 — Many results

3. Select students who have psp greater than all students of bid 3

	id	name	bid	psp	
✓	1	A	1	90	✓
✗	2	B	2	70	✗
✗	3	C	3	85	✗
✗	4	D	3	80	✗
✗	5	E	3	75	✗
✗	6	F	2		

✓	7	5.	1	87 ✓

```
Select *
from students
where psp > (
    select MAX(psp)
    from students
    where bid = 3
)
```

85 — n^2

```
Select *
from students
where psp > ALL (
    Select psp
    from students
    where bid = 3
)
```

$n^2 \cdot x$ → 85, 80, 75

→ ANY ?

```
Select *
from students
where psp > ANY (
    SELECT PSP
    FROM students
    where bid = 3
)
```

→ 85, 80, 75

Where (80, 85, 75) vs where (80, 85, 75)

psp > ALL (–)

psp > (MAX(PSP))

psp > ANY (–)

psp > (MIN(psp))

psp > 80. && psp > 85. && psp > 75.] psp > 85

psp > 80 || psp > 85 || psp > 75] psp > 75

= [Single Result SQ]

SQ < IN, ALL, ANY.] → Multi-result SQ

5. Co-related Subqueries

Select all students who have psp greater than average psp of their respective batch.

	id	name	bid	psp	#psp
✓	1 ✓	A	1	90 ✓	85
✓	2 ✓	B	1	85	85
✓	3 ✓	C	1	80	85
✓	4	D	2	100	90
	5	E	2	90	90

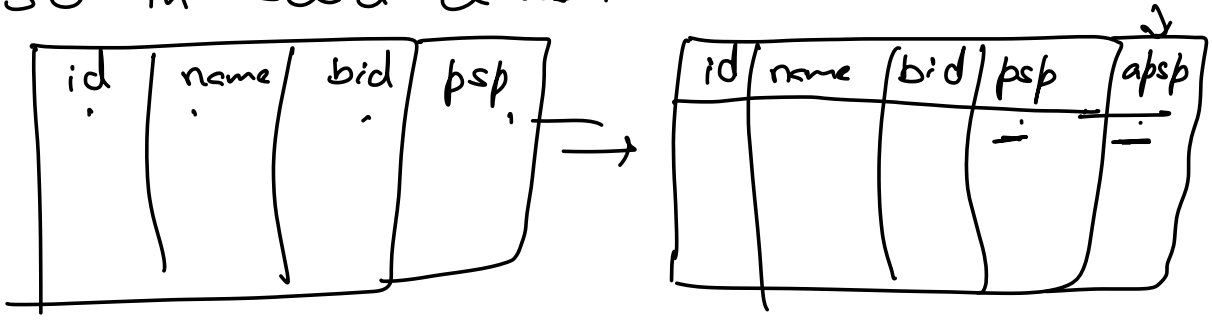
6	F	3	60	75	
7	G	3	70	75	75
8	H	3	80	75	
9	I	3	90	75	

```
Select *
from students sout
where psp > (
        SELECT AVG(psp)
        from students sin
        where sin.bid = sout.bid.
)
```

→ SQ in FROM clause

Break 10:33 – 10:40

7. SQ in Select clause.




```
Select s1.*, (SELECT AVG(psp) from students s2
                where s2.bid = s1.bid)
```

from students s1 → APSP

id	n	bid	psp
1	A	1	80
2	B	1	90
3	C	2	60
4	D	2	70

1	A	1	80	85
2	B	1	90	85
3	C	2	60	65
4	D	2	70	65

8. SQ in FROM CLAUSE

id	name	bid	psp

Student

id	name	iid

Batch

Multi-level Join

id	name

Instructor

s.name , i.name

Students

id	name	bid	psp
1	A	1	60
2	B	1	70
3	C	2	80
4	D	2	90
5	E	3	100

Batches

id	name	iid
1	B1	1
2	B2	2
3	B3	1
4	B4	2

id	name
1	abc
2	def

6	F	4	75

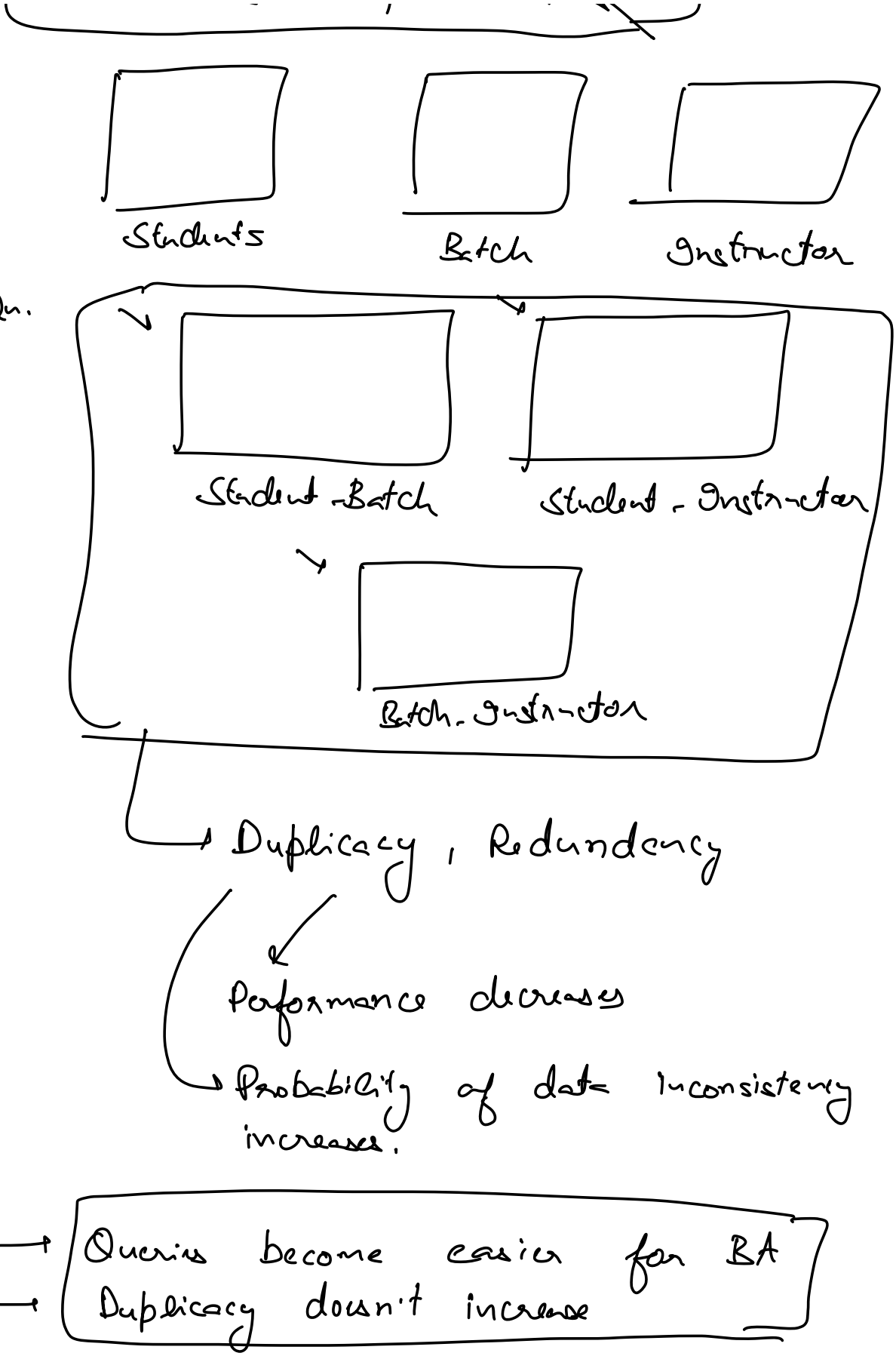
A	abc
B	abc ✓
C	def ✓
D	def ✓
E	abc ✓
F	def ✓

```
Select s.name, i.name,
from students s
JOIN Batches b
    ON s.bid = b.id
JOIN instructors i
    ON b.iid = i.id
```

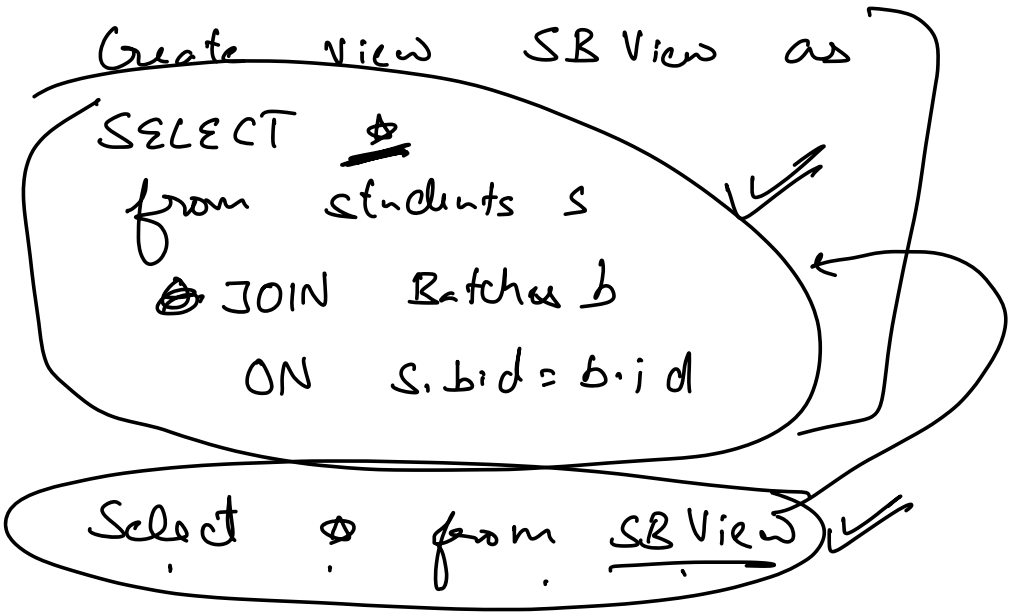
```
Select T.iid, AVG(T.psp)
from
(
  Select i.id As iid, i.name As iname, s.id As sid, s.name As sname, s.psp
  from students s
  JOIN batches b
      ON s.bid = b.id
  JOIN instructors i
      ON b.iid = i.id
) as T
where T.iname Like 'a%'
GROUP BY T.iid
```

4. Views

1. Instructor & Student
2. Student, Batch
3. Instructor, Batch.


Students
Batch
Instructor
Soln.
Student-Batch
Student-Instructor
Batch-Instructor
Duplicacy, Redundency
Performance decreases
Probability of data inconsistency increases.
Queries become easier for BA
Duplicacy doesn't increase

→ Views : Stored Queries



Typical Reasons
- Control what data to show & what to hide
- Lot of features to Business Analytics but still hide complexity

10. Demo on view

11. ~~IF, CASE~~, IFNULL, COALESCE

IFNULL (p<p, 0)

if this value is not null we will get psp otherwise 0

TA

id	name	st.id
1	X	null
2	Y	null
3	A	1
4	B	2

Select id, name, IFNULL(st.id, -1)
FROM TA

1	X	-1
2	Y	-1
3	A	1
4	B	2

COALESCE is many ifnulls put together

COALESCE (psp, assn, hw, 0)

12. IF CLAUSE

x == 7 ? "Seven" : "Not Seven"

IF (condition , t , f)

id	name	psp
1	A	82
2	B	78
3	C	64
4	D	92

psp ≥ 80
↳ eligible for placements

Select id, name, IF(psp ≥ 80, "Yes", "No")
from students

1	A	Yes
2	B	No
3	C	No
4	D	Yes

CASE CLAUSE

- psp ≥ 80 — P0
- psp ≥ 70 — P1
- psp ≥ 60 — P2
- ELSE — NA

```
SELECT id, name,
       CASE
           WHEN psp ≥ 80 Then "P0"
           WHEN psp ≥ 70 Then "P1"
```

```
        WHEN psp ≥ 60 then "P2"
        ELSE "NA"
    END AS placement_priority
FROM
Students.
```