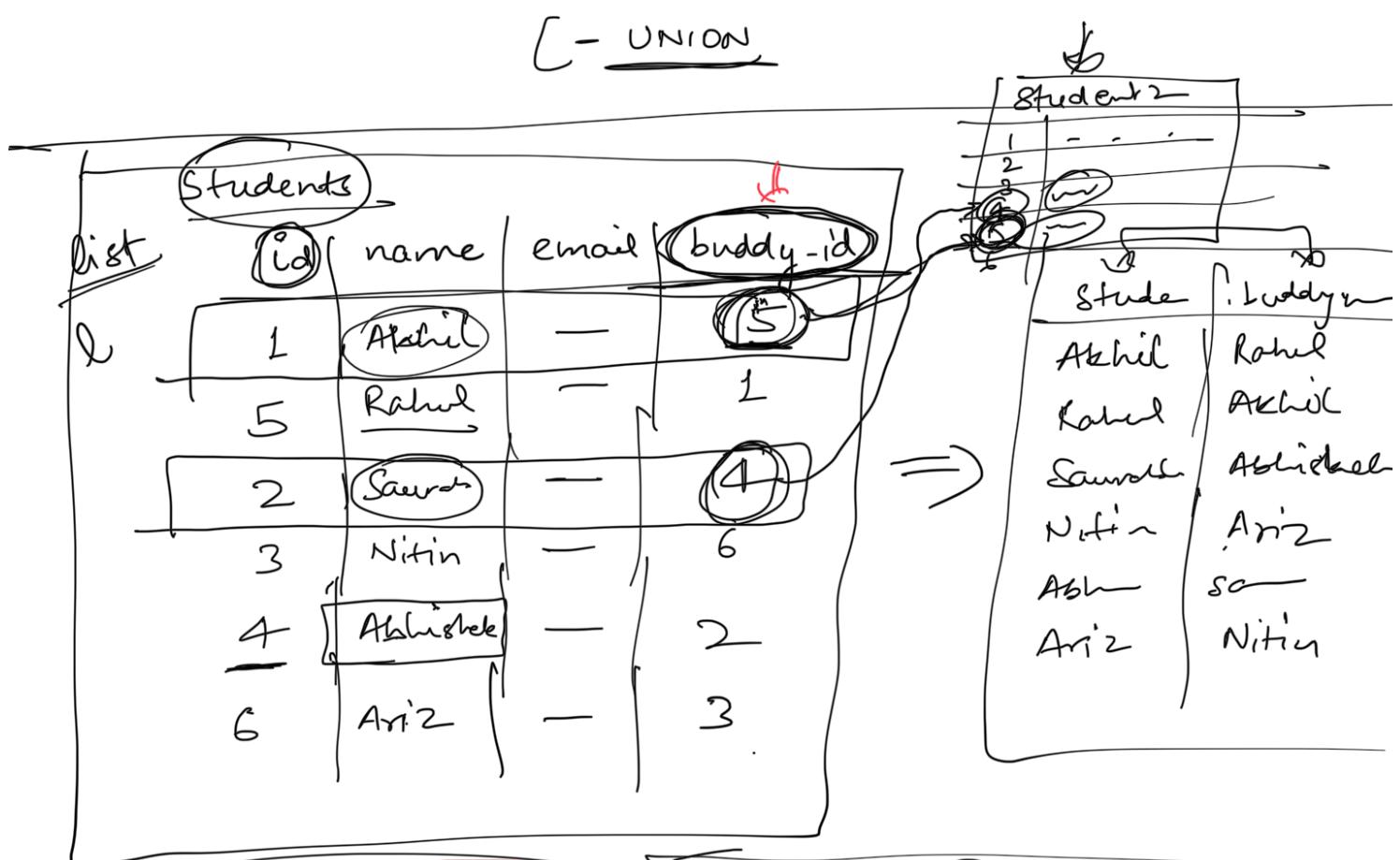


SQL 4 - JOINS + UNION

AGENDA :

- SELF JOIN
- MORE PROBLEMS ON JOINS
- INNER VS OUTER JOINS
 - LEFT, RIGHT, FULL
- WHERE VS ON

C - UNION



SELECT

```

a.name AS student-name,
b.name AS buddy-name
FROM Students a
JOIN Students b
ON a.buddy-id = b.id
  
```

i in list:

j in list:

```

if i.buddy-id = j.id:
    print i, n, j
  
```

→ ① Figure out tasks that have my info (students)

(T.2) Joins, etc. on right join paren

② Filters (where)

③ what to print (cols)

④ Sort by

⑤ limit, offset.

Students: (a)

id	name	buddy-id
1	Jay.	2
2	Anshuman	1
3	Abhishek	4
4	Hrushikesh	3

① (Jay), 2

Jay.

b.name
= Jay

b.buddy-id = 1

b ↓ name, buddy-name

id = ↘

1	Jay.	2
2	Anshuman	1
3	Abhishek	4
4	Hrushikesh	3

SELECT

a.name AS student-name, b.name AS buddy-name

FROM

Students AS a

JOIN Students b AS b

~~ON~~ a.id = b.buddy-id

i in Students:

j in Students:

if i.buddy-id = j.id:

Sakila DB

film

film-id	title	language-id
1	Rock On	2
2	Xyz	3

language

id	name
2	Hindi

[title, name of language]

for i in film

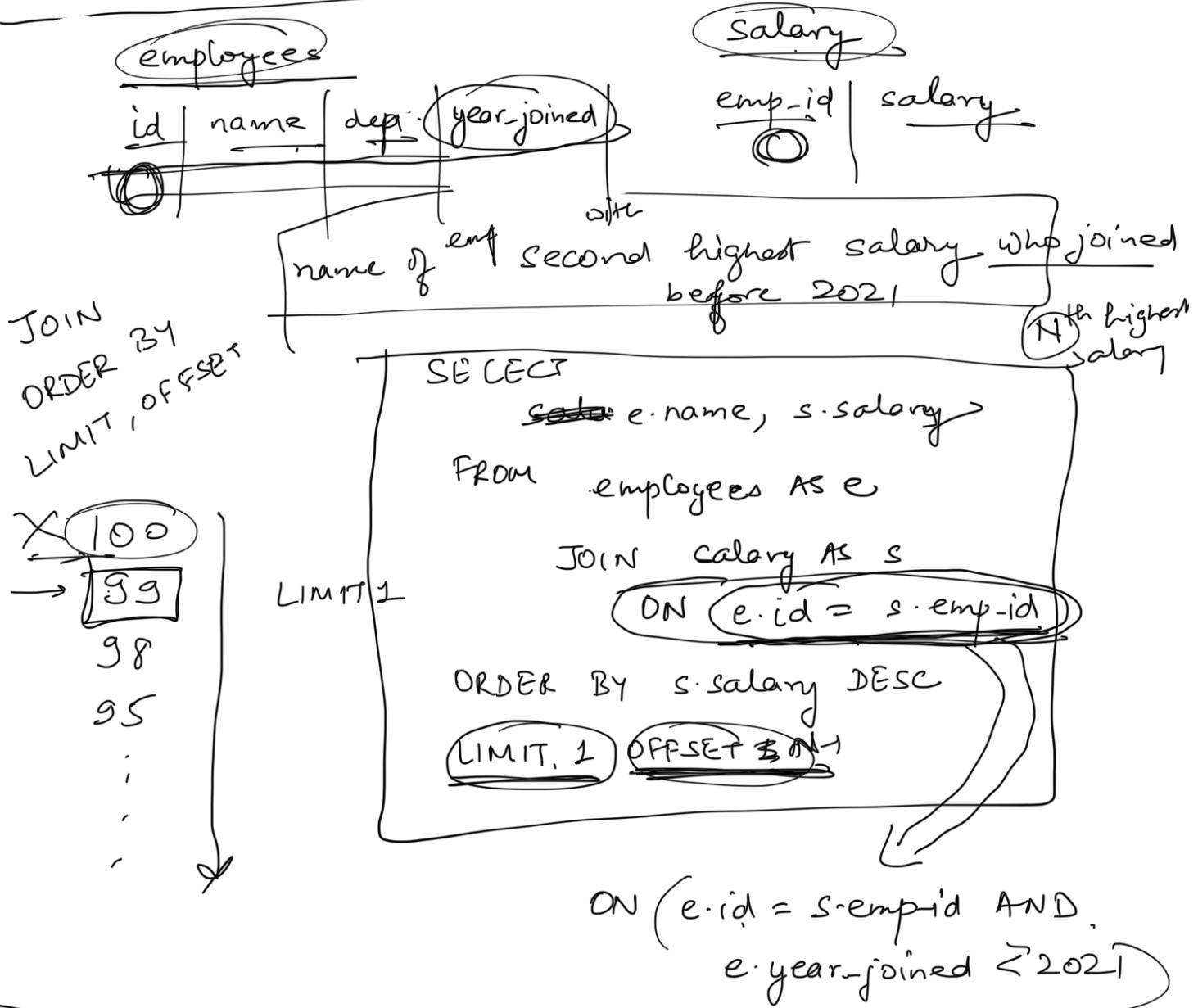
for j in language:

if i.id = j.id

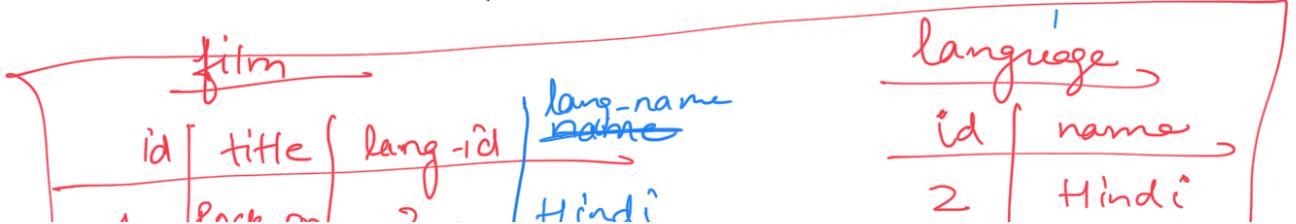
Γ return

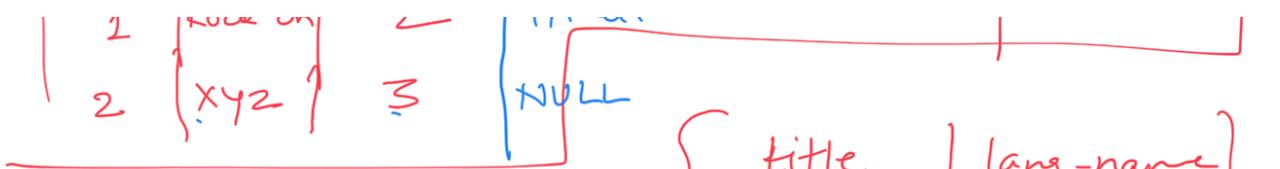
```

SELECT film.title, language.name
FROM film
JOIN language
ON film.language-id = language.id
  
```



INNER vs OUTER JOINS





OUTER JOINS

LEFT

RIGHT

FULL

SELECT

f.title, l.name
FROM film AS f

LEFT OUTER JOIN language AS l
ON f.lang-id = l.id

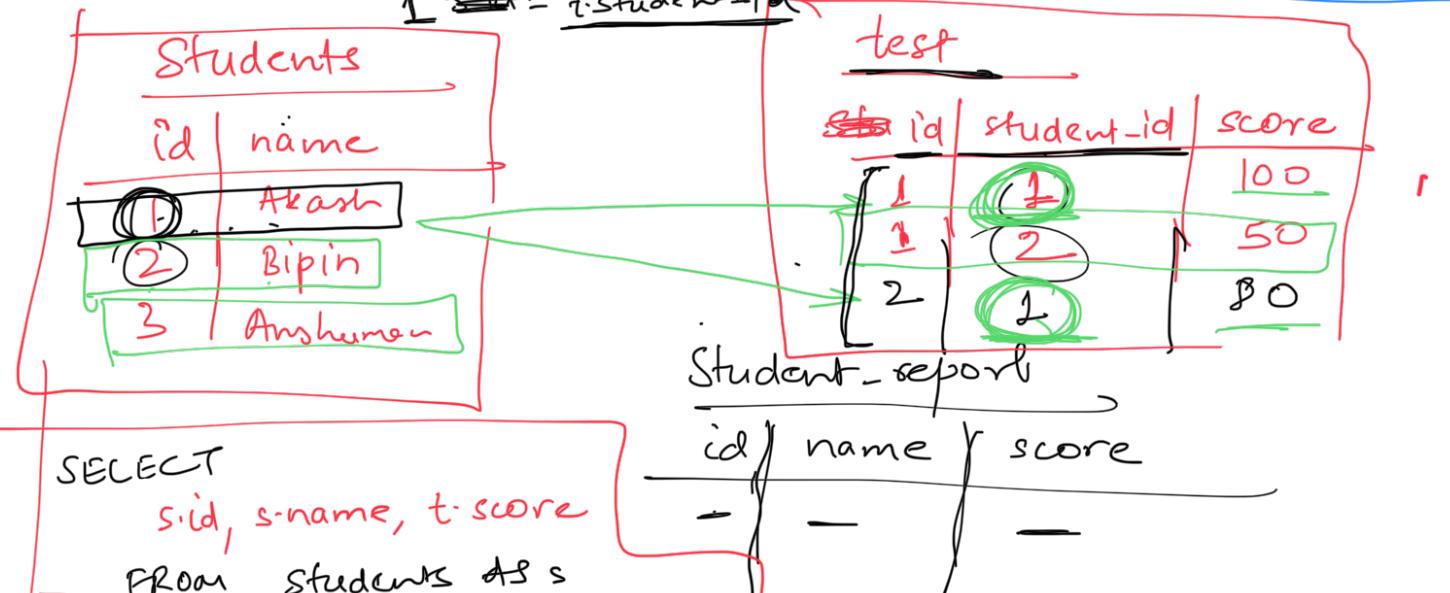
SELECT

f.title, l.name
FROM film AS f
JOIN language AS l
ON f.lang-id = l.id



for i in film
for j in language:
if i.lang-id = j.id:
if no-match:
point i..., NULL

~~1 = t.student-id~~



SELECT
s.id, s.name, t.score

FROM Students AS s

LEFT OUTER JOIN test AS t
ON s.id = t.student-id

RIGHT

Output ?

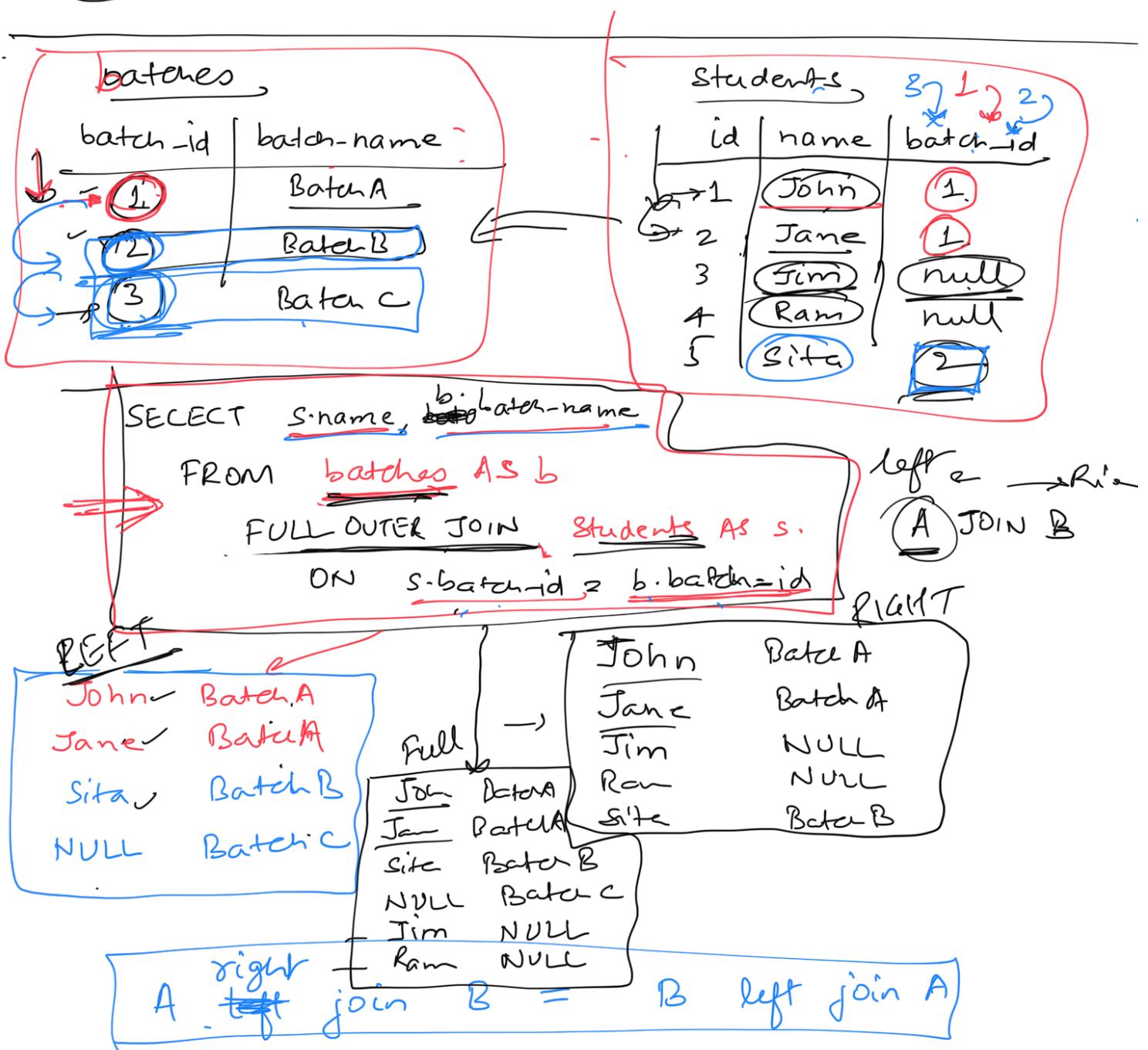
1, Akash, 100 7

```

for (i) in students
  for j in test :
    if i.id = j.st :
      {
        = 
      }
    no-match:
      print i, NULL
  X

```

1, Akash, 80
2, Bipin, 50
3, Anshu, NULL

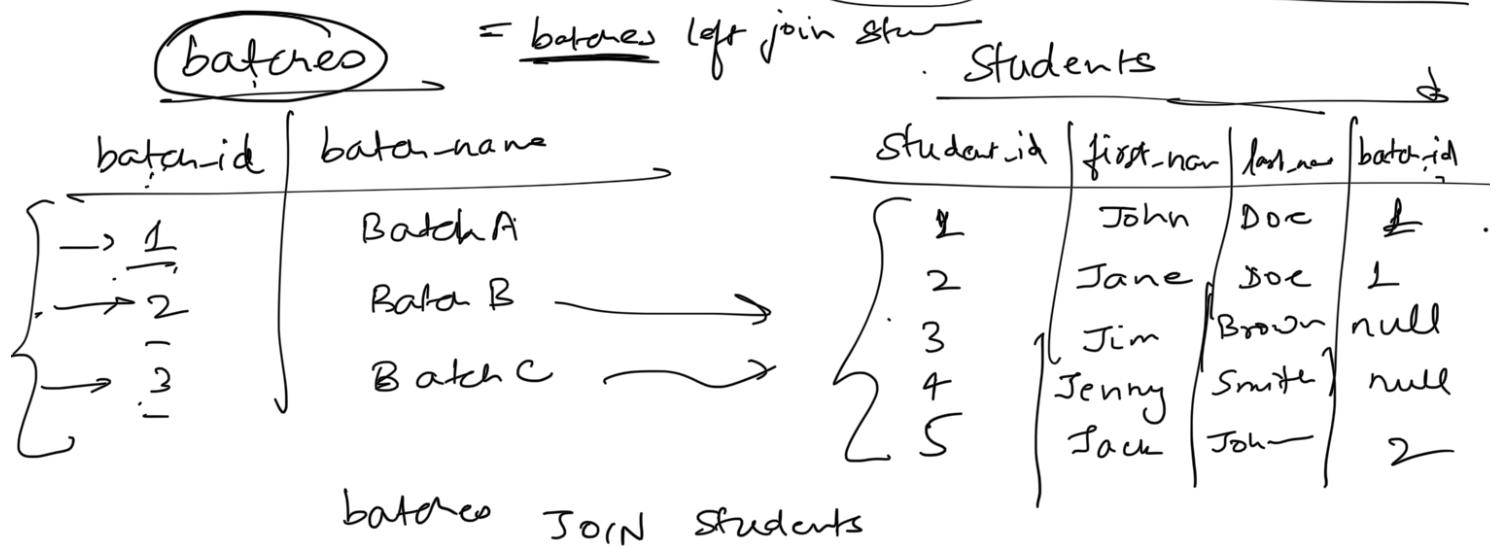


batches left join students

= Students right join batches

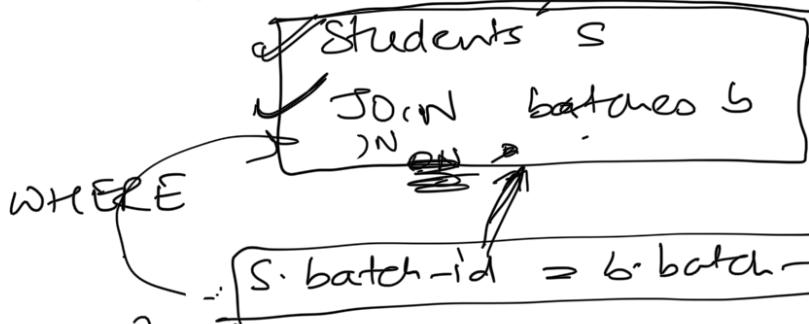
Student RIGHT JOIN batches

= batches left join student



SELECT *

FROM



WHERE

for i in t1:

for j in t2:

Give me email of every person associated with scalar

for i in student

for j in batches:

possible outputs add (i,j)

Students

id	name	email
100		
101		
102		

instructor

id	name
1000	
1001	
1002	

mentor

id	name

TA

id

SELECT * FROM ... PRO.

Query 1 → same no. of columns

```

    (SELECT email from
     students
    UNION
    SELECT email from
     instructors
    UNION
    SELECT email from mentors)
  
```

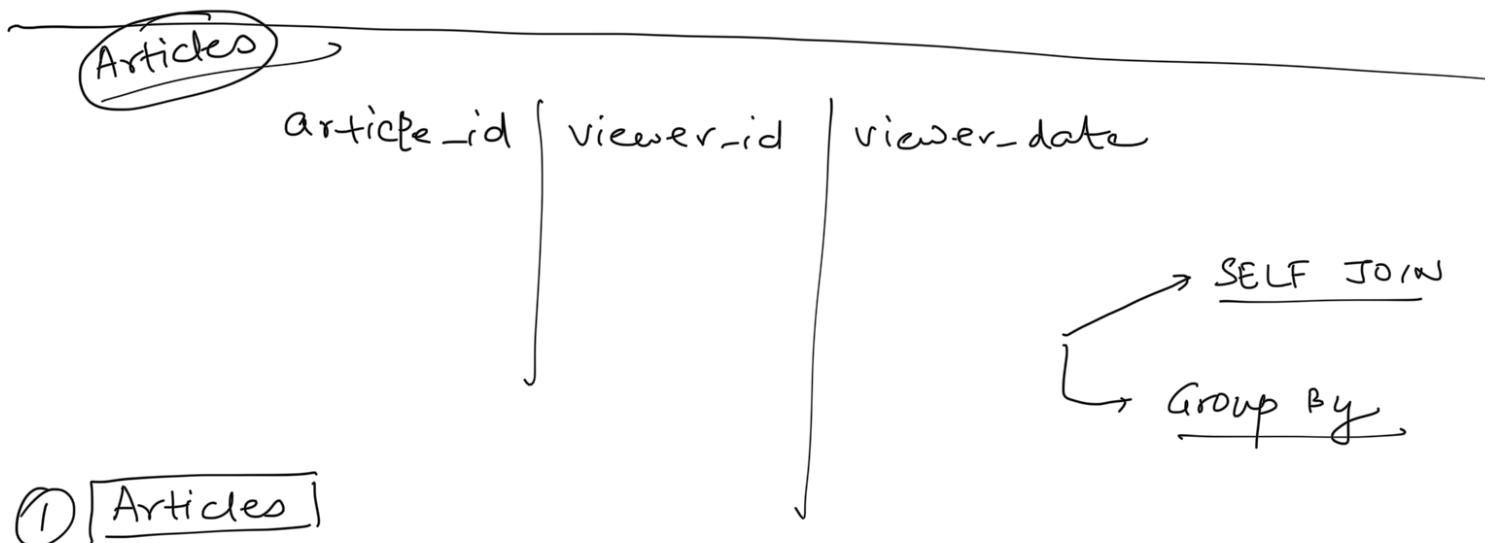
) = o -

UNION

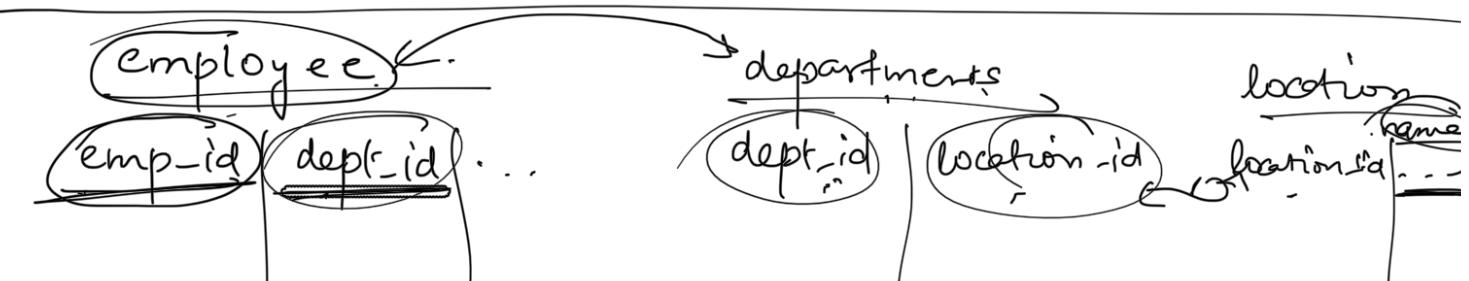
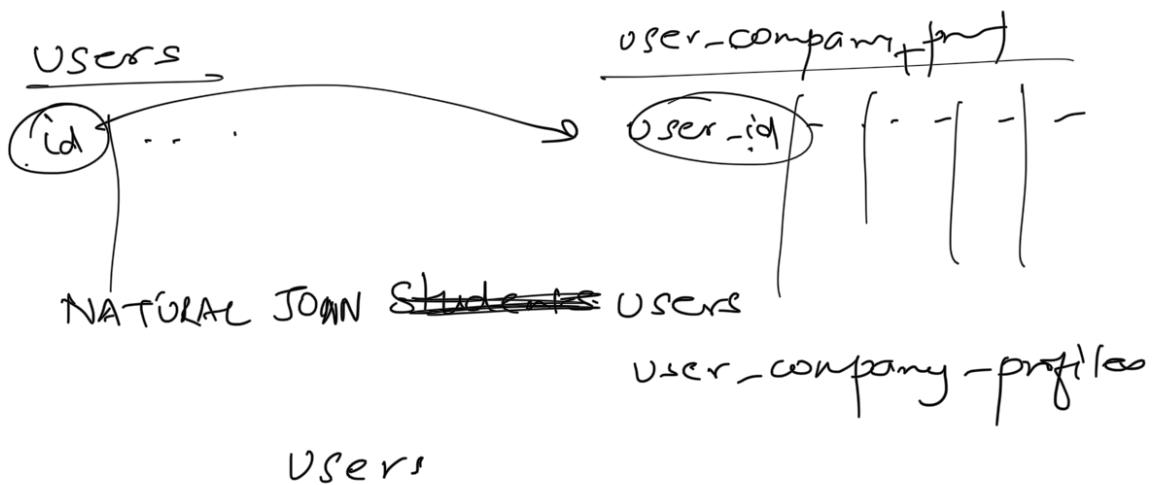
← Query 2

UNION

Query 3



① Articles

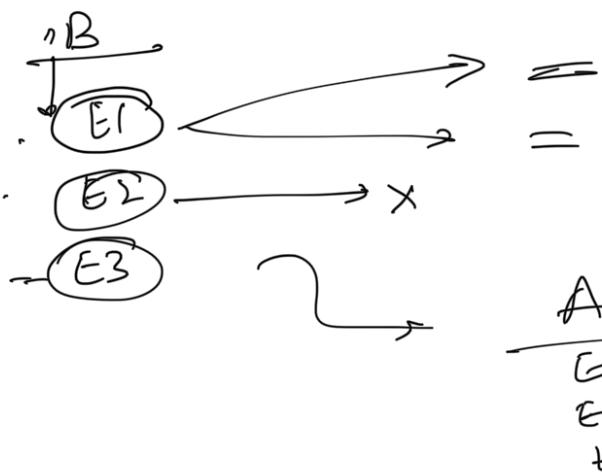
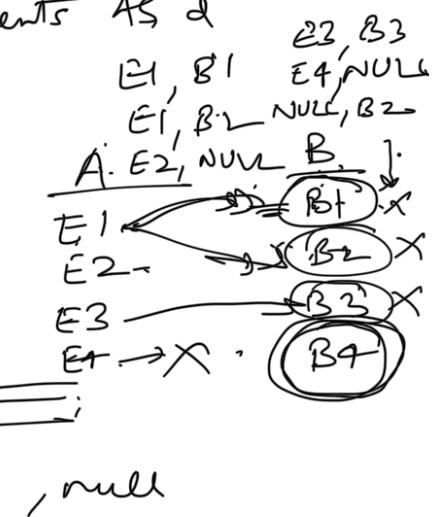


employee AS e

JOIN departments AS d

A FULL OUTER JOIN B

\Rightarrow



E2, null

LEFT JOIN

+

print entries in
right table
that had no match