

\$ functions used in nested queries:

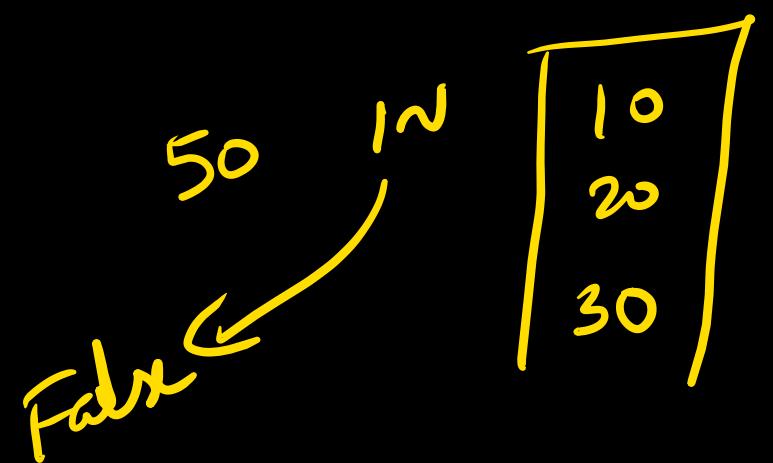
{ IN / NOT IN } There are best suitable for
 queries without correlation ✓

ANY
ALL

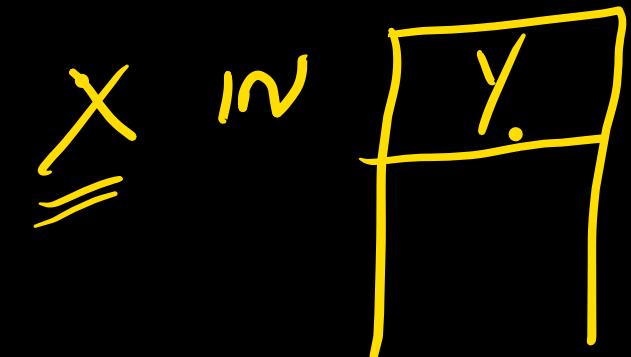
{ exists / not exists } Best suited for queries
 with correlation. ✓

Queries using functions:

In function: used for membership test



\in will return true, if x val is a member of y set



\in value must be equal to atleast one value of \in set.
 \in Then it will return true

\Rightarrow IN can be used in (equi join) queries



$$R(A \underset{=}{\cdot} B) \quad S(\underset{=}{\cdot} B C)$$

$$\pi_{AB} (R \bowtie S) \cong \text{Select } * \text{ from } R: \\ \text{natural join} \\ \text{when } \underset{=}{\cdot} B \text{ in } (\text{Select } B \text{ from } S)$$

$$R.B = S.B$$

$R(A, B) \quad S(B, C)$

$\pi_A(R \bowtie S)$
 $R.A = S.B$

\approx Select A
from R
where A in (Select B
from S) ✓
non correlated
executed only once ✓

10
20
30
40

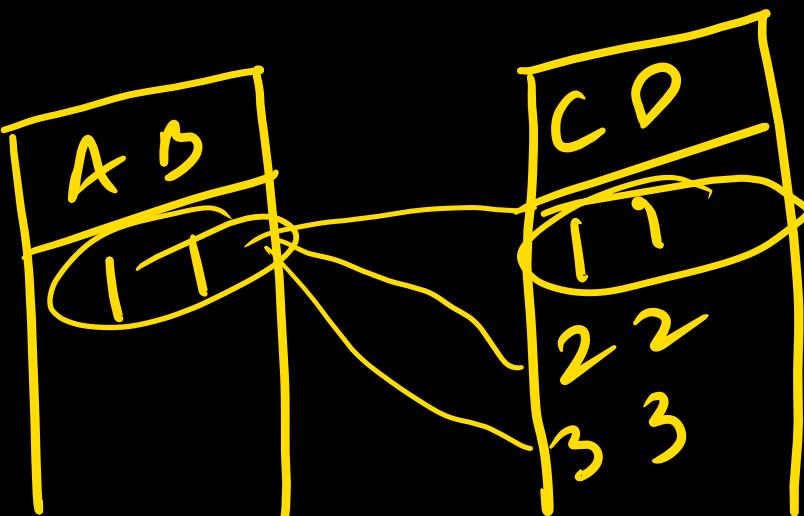


$R(A, B)$ $S(C, D)$

$\pi_{AB}(R \bowtie S)$

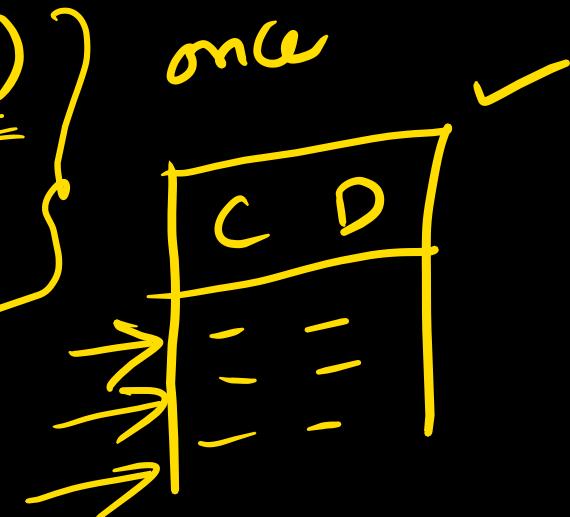
$R.A = S.C$
 $\wedge R.B = S.D$

↓
equijoin



$R.A = S.B \rightarrow$ equijoin
 $R.A < S.B \rightarrow$ not equijoin.

Select distinct A, B
from R
where $A, B \in \{$ select (C, D) from S $\}$

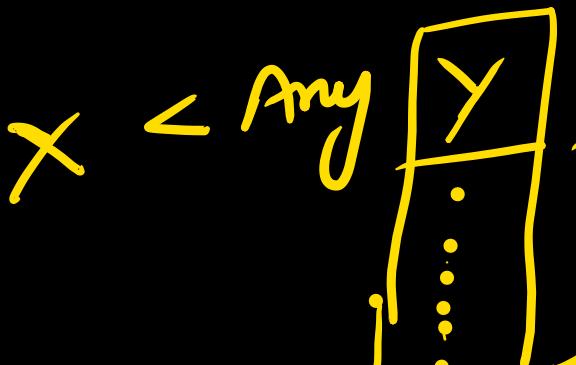


✓ ANY and ✓ All

Preceded by operator

<, <=, >, >=, =, <>

$x < \text{Any } Y$



$\begin{array}{ll} < \text{any} & > \text{All} \\ \Downarrow & \\ \text{some value} & \end{array}$

$\begin{array}{ll} < \text{any} & >= \text{any.} \\ > \text{any} & \end{array}$

$\begin{array}{ll} > \text{any} & < \text{any} \\ \Downarrow & \\ \text{some value} & \end{array}$

$\begin{array}{ll} > \text{any} & <> \text{any} \\ \Downarrow & \\ \text{some value} & \end{array}$

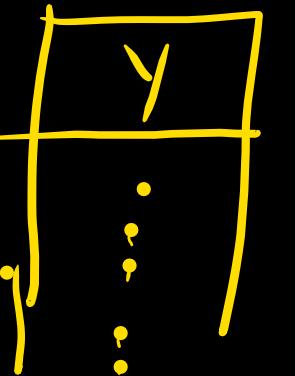
R.B	
B	
10	
20	
40	
50	✓

$50 < \text{any}$

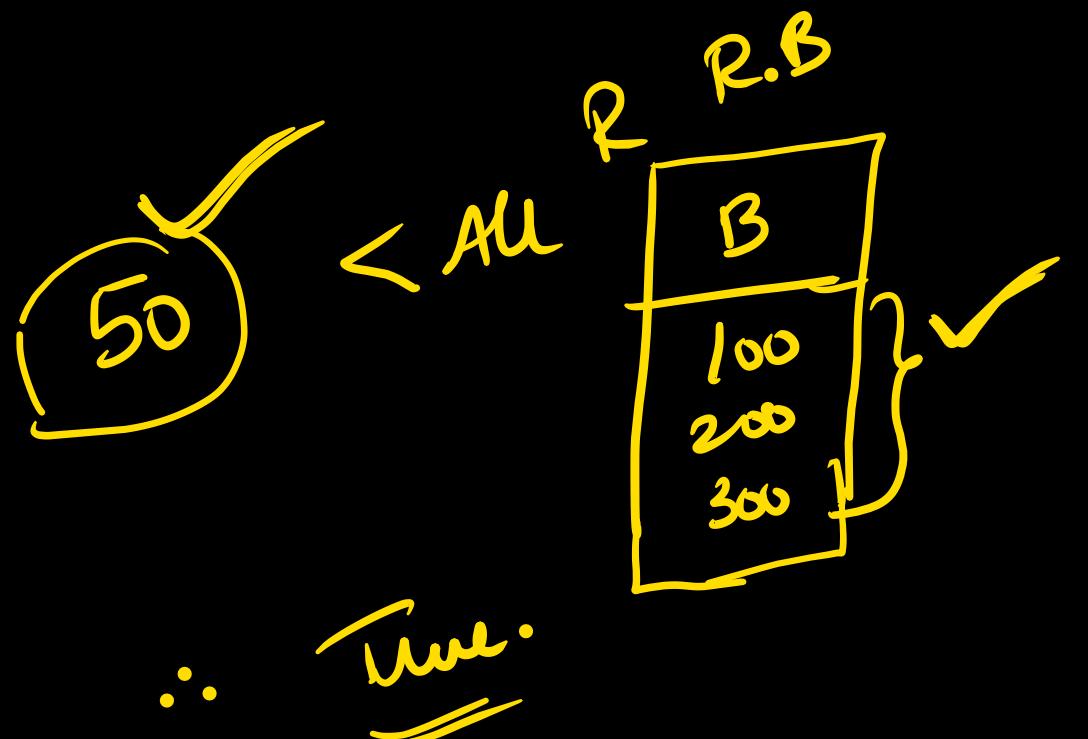
True if $x \text{ vale} < \text{some value of } Y.$

Tree

$x < \text{All } y$



True if x value $<$ every value of y set.



"ANY" function can be used for Conditional join Δ_C queries.

\bowtie
 $R.A < S.B.$

$R(A \dots)$ $S(B \dots)$

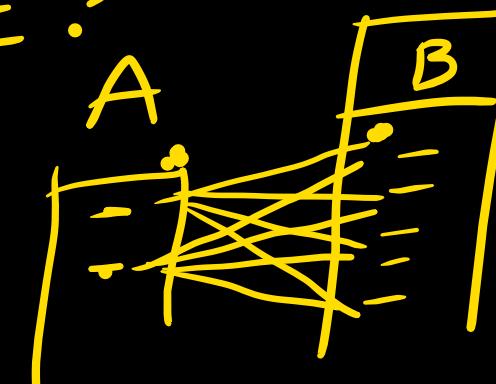
Retrieve A values of R there are less than $\frac{\text{Some } B \text{ values of } S}{\times c}$.

$$\pi_A(R \bowtie_S) \\ R.A < S.B$$

Select distinct A
From R
where $R.A < \text{Any}$

Select distinct $R.A$
From R, S
where $R.A < S.B$

(Select B from S) \rightarrow $\theta 1$ time



You don't have to write queries in exam.

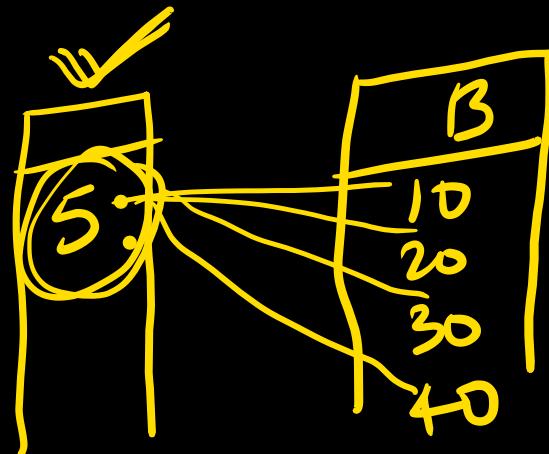
- Q: Given a question, what is the query.
- a) Given a query, what is the meaning.

$R(A \dots) \quad S(B \dots)$ Retrieve 'A' values of 'R' those are less than
 $\underline{\underline{every}} \quad \underline{\underline{B}}$ value of 'S'.

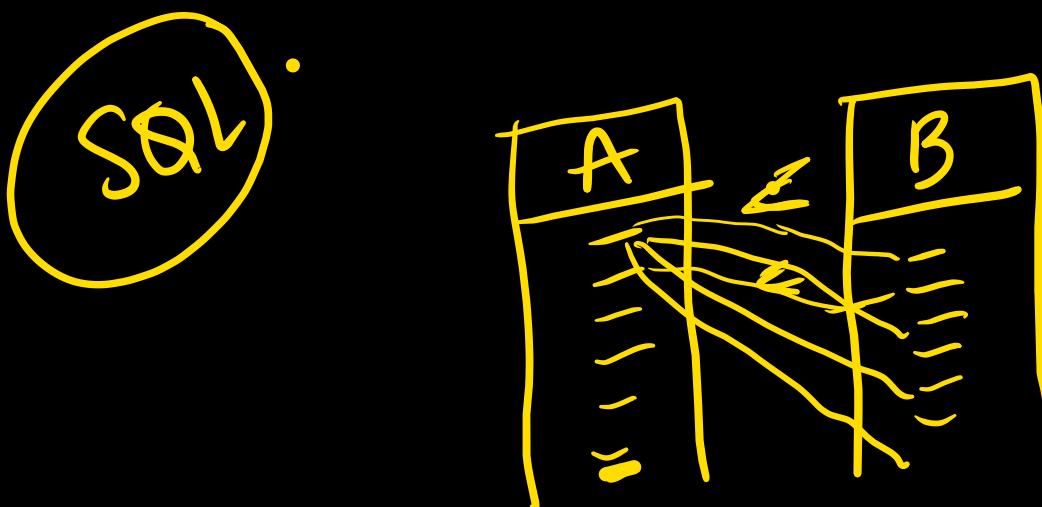
$$A < \text{every}_x B \cdot \quad \underline{\underline{U - \bar{A}}}$$

$$A < \text{every} \approx \{ \text{all values of } A \} - \{ A \geq \text{some } B \}$$

$$\pi_A(R) - \pi_A(R \bowtie S) \quad R.A \geq \cancel{S.B}$$



\approx Select Distinct A From R where
 $R.A < \text{All} (\text{Select } B \text{ From } S)$



\checkmark I time.

IN:

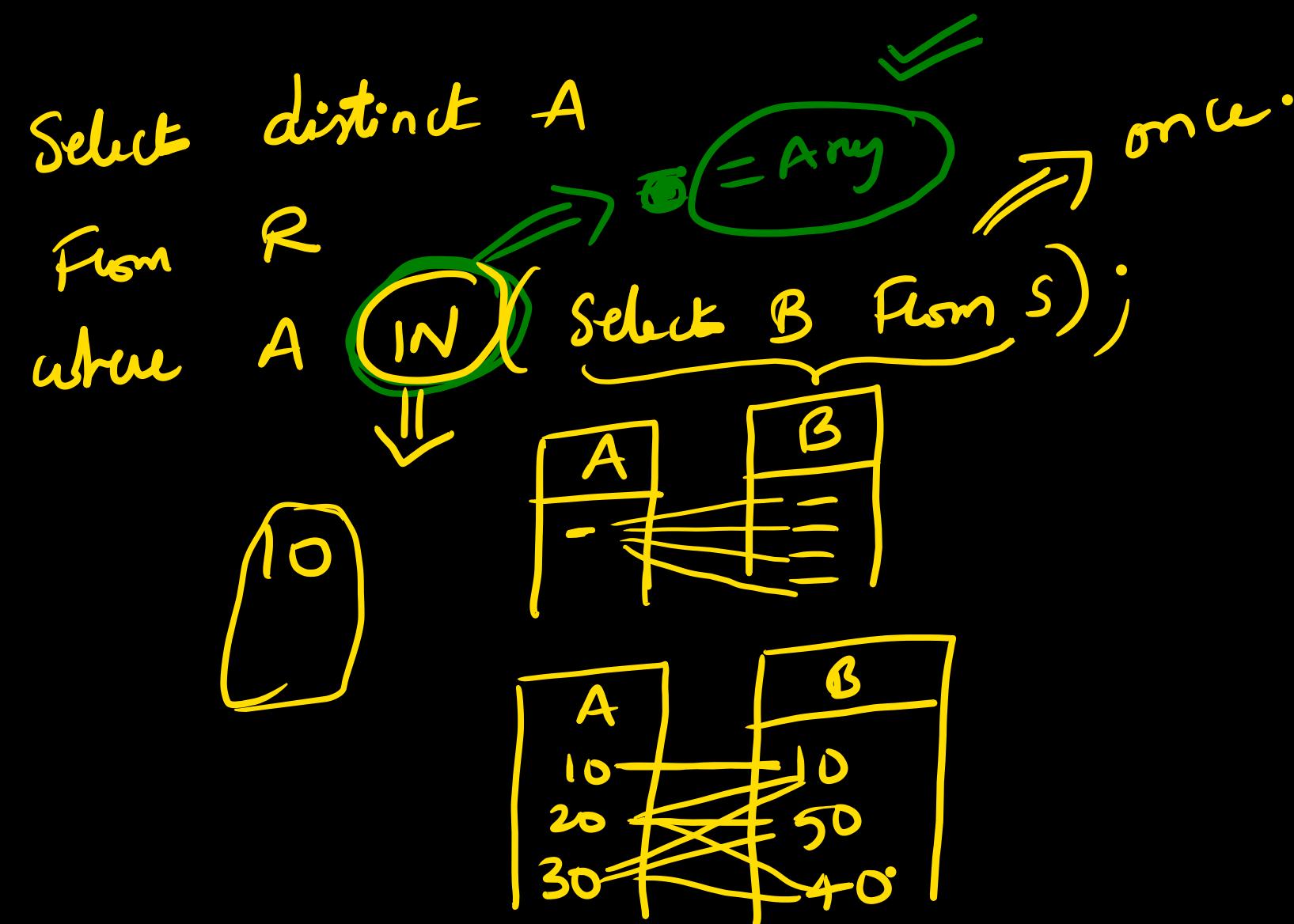
$R(A) \text{ } S(B)$

$\pi_A(R \bowtie S) \Rightarrow$

$R.A = S.B$

Compare

given



$$\begin{array}{l}
 R(A, B) \quad S(C, D) \\
 \pi_{AB} (R \bowtie S) \equiv \\
 R.A = S.C \\
 \wedge R.B = S.D
 \end{array}
 \qquad
 \begin{array}{l}
 \text{Select distinct } (A, B) \\
 \text{From } R \\
 \text{where } \underline{(A, B)} \text{ in } (\text{Select } \underline{(C, D)} \\
 \text{from } S)
 \end{array}$$



 Because there are
two columns.

(NOT IN)

10 not in

↓
True

B
11
12
13

~~6~~

~~<>All~~ \cong NOT IN

11 not in

↓

False

B
11
12
13

Exists:

used to test result of inner query : Empty & not empty .

exists (inner query)
= \emptyset

True

False

at least 1 ✓
↓
✓ Some & any ✓

Exists (inner query)

Return true if result of inner query

has at least one record .

Some / any
 \equiv

Emp(eid, dno, Sal)

Retrieve eids whose sal less than some emp sal of dept 5. ✓

$\pi_{\overline{eid}}(\overline{\text{Emp}})$

$\left(\begin{array}{c} \text{Some} \\ \cancel{\text{---}} \\ \text{Sal} < \cancel{\text{---}} \end{array} \right) \circ \overline{\text{Emp}}$

dno=5

Select

Select Eid
 From $Emp T_1$
 where exists
 (Select * From $Emp T_2$
 where $T_2.dno = 5$
 and $T_1.Sal < T_2.Sal$)

Emp

T_1

	Eid	dno	Sal
✓	e ₁	4	10
✓	e ₂	5	20
✓	e ₃	4	30
✗	e ₄	5	40
✗	e ₅	4	50

Emp

T_2

	Eid	dno	Sal
	e ₁	4	10
	e ₂	5	20
	e ₃	4	30
	e ₄	5	40
	e ₅	4	50

e_1
 e_2
 e_3

correlated subquery will execute ~~for~~ ~~any~~ multiple times, one time
for each row in the outer ~~table~~ variable.

non correlated subquery will execute only once.

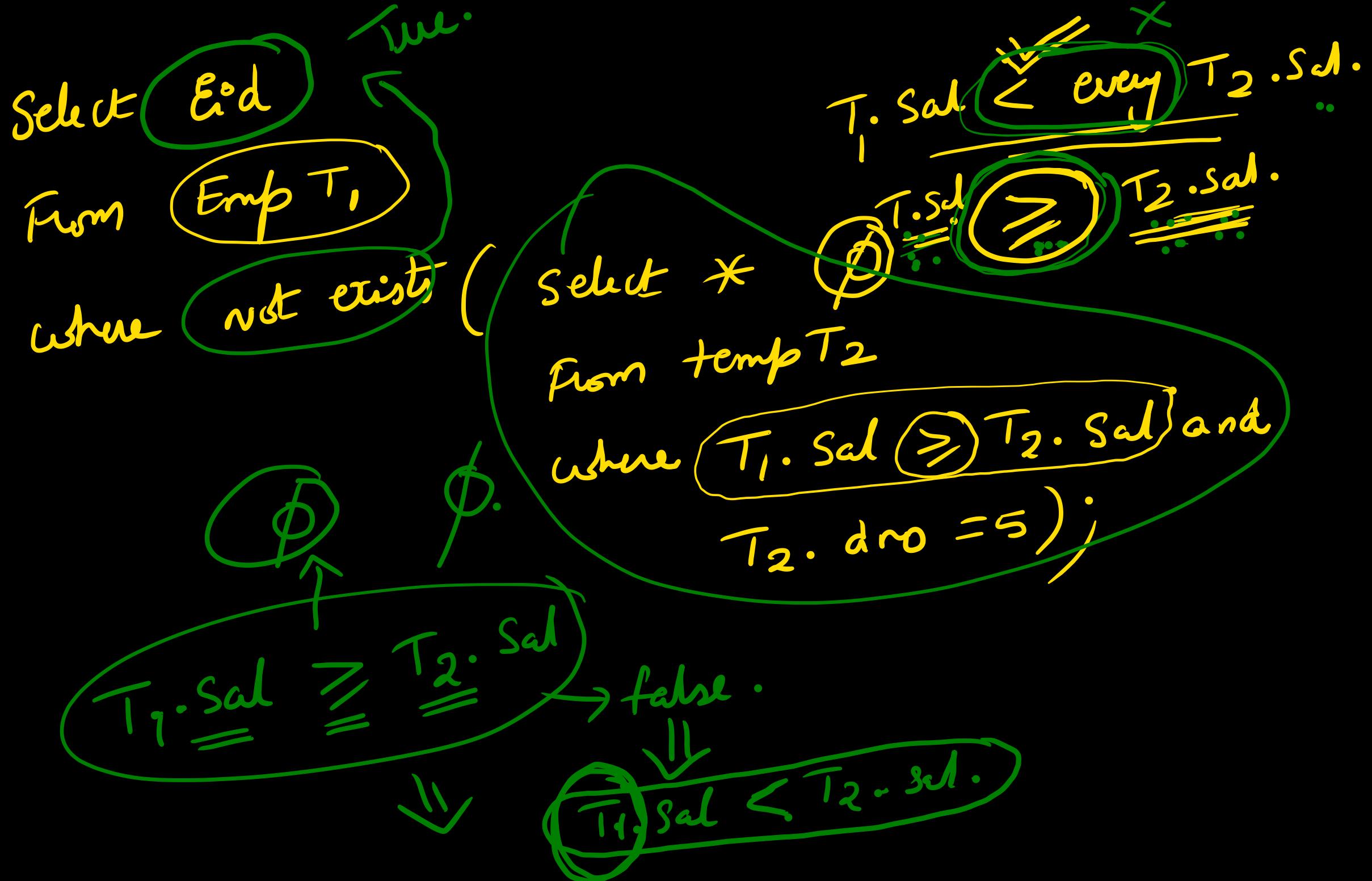
Exists clause behaves like \bigcup_c queries.

$\text{Emp}(e\text{id}, \text{dno}, \text{Sal})$

Retrieve $e\text{id}$'s whose salary less than every emp of dept 5.

$\pi_{\underline{e\text{id}}}^{(\text{emp})} - \pi_{\underline{e\text{id}}} \left(\underset{\text{sal} \geq s}{\cancel{\text{Emp}}} \setminus \left\{ \begin{array}{l} \text{Emp} \\ \text{dno=5} \end{array} \right\} \right)$

SQL

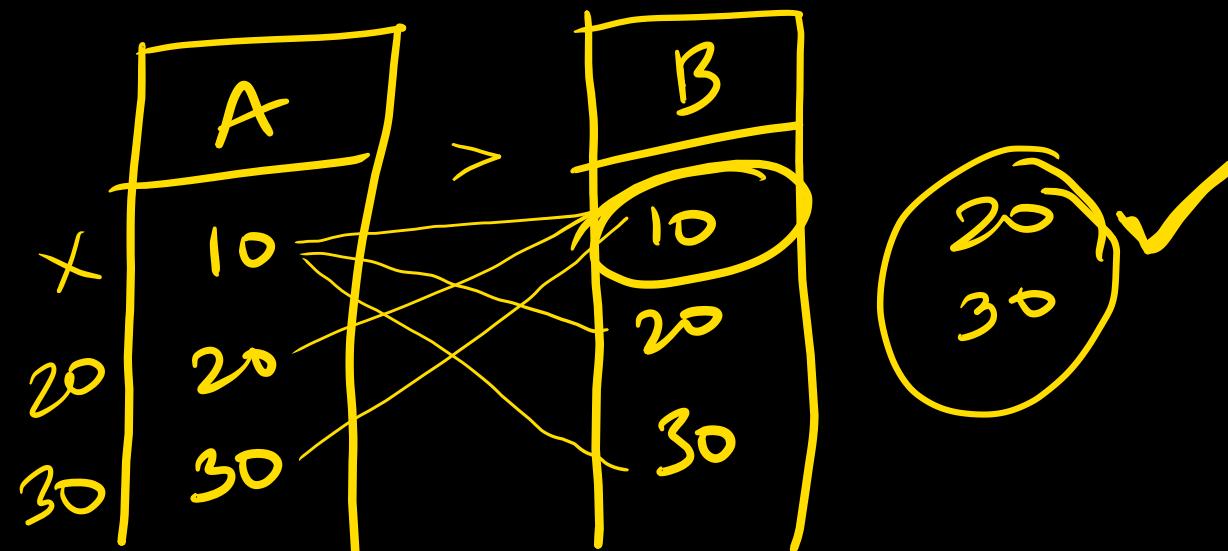


$R(A....)$ $S(B....)$

Retrieve 'A' values of R which are more than some B values of S.

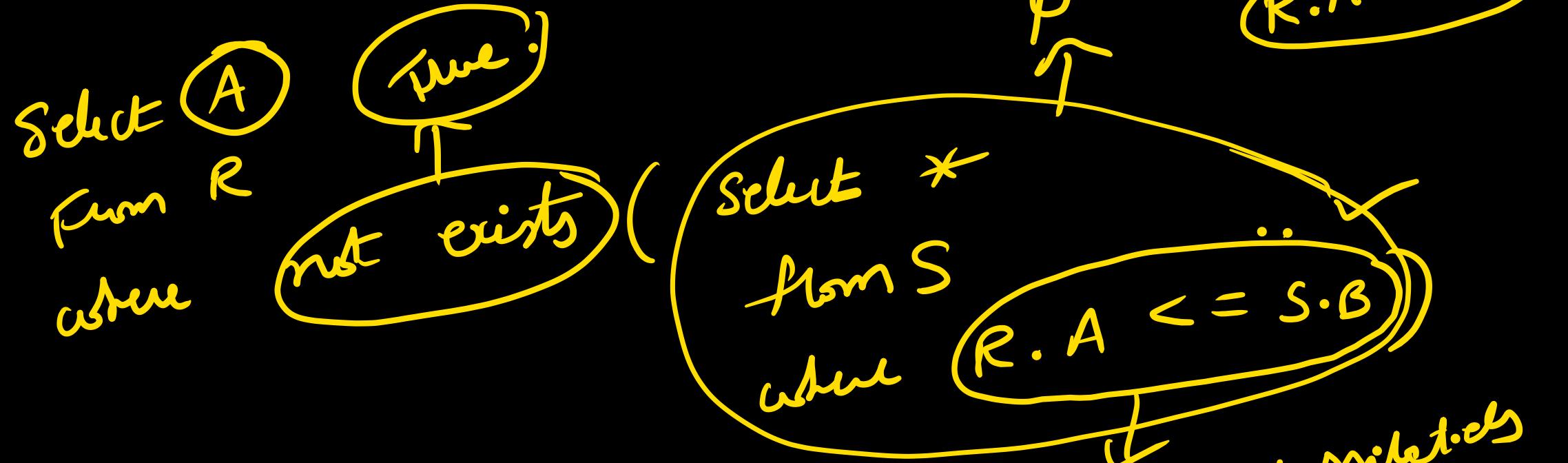
Select R.A
from R
where exists (select *
from S
where R.A > S.B)

Correlated
nested
query.



$R(A\ldots) \quad S(B\ldots)$

Retrieve 'A' values of 'R' which are \leq than every 'B' values of S.



For all possibilities
if it fails, the
means $R.A > S.B$.