

Class starts at 7:05 PM

LA → pls give two make classes to Take

Option 1: go with dbms (7-9) ~~pm~~.

Option 2:

stop dbms for few days and do algo

✓ Option 3 DBMS and algo in parallel.

{ DBMS  $\rightarrow$  31%

algo - 36%

DB and alg parallel - 32.500 ✓

DBMS

From tomorrow

6-8 PM - DBMS

8-10 PM - C program

over in 15 days

After 15 days:

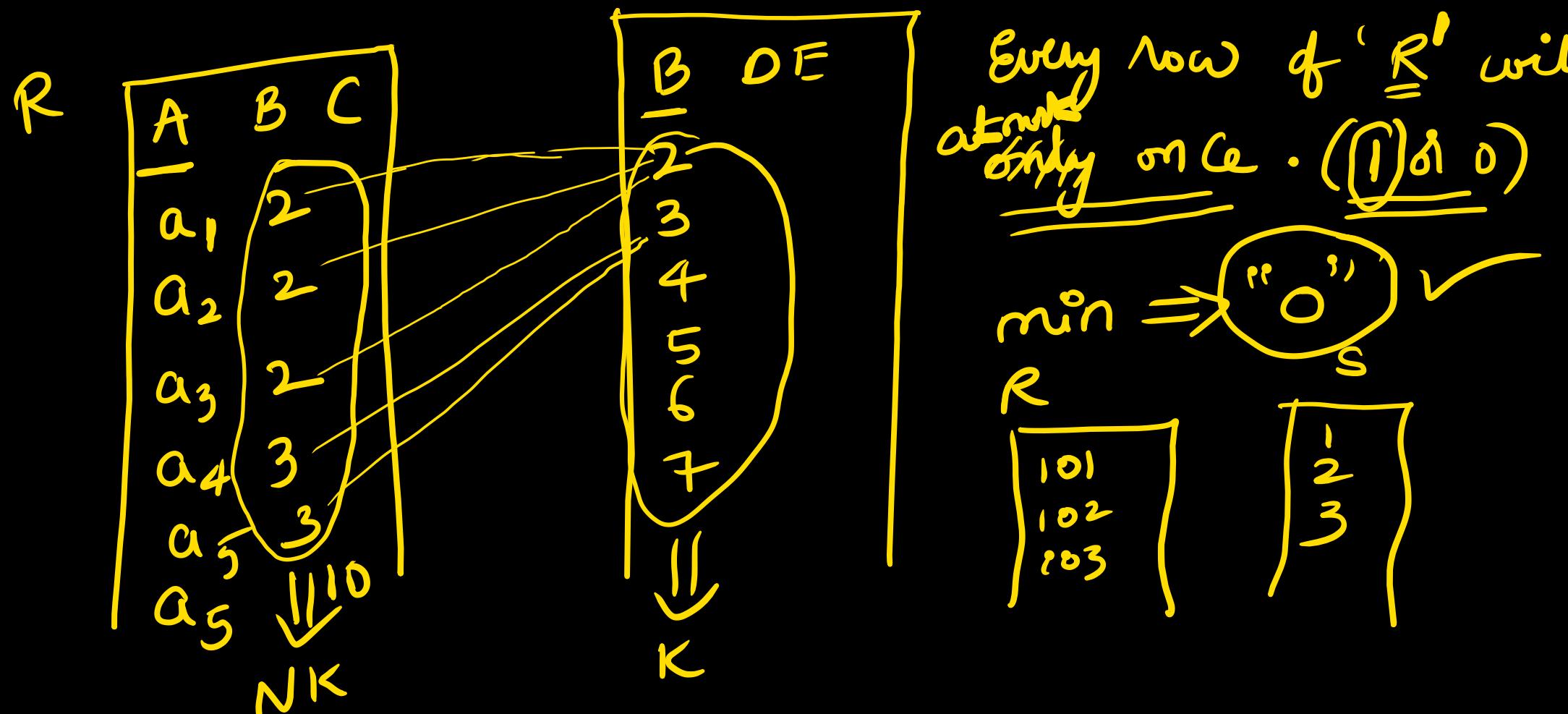
{ 6-8 PM - DBMS  
8-10 PM - Algo. } Parallel.

majority < 6-10 .. . . .

7-11

$\rightarrow R(\underline{A} BC)$  no nulls with  $n$  tuples  
 $S(\underline{B} DE)$  with  $m$  tuples  
 How many max tuples in the result of  $R \bowtie S$   
 " " min "

" " " "  
 $\bowtie \rightarrow$  common attr.  $\rightarrow B.$



$R(\underline{A} BC)$

with  $n$  tuples.

$S(\underline{A} DE)$

with  $m$  tuples.

max tuples

min

in the resulting of  $R \bowtie S$

" "  $R \bowtie S$

maximum possible

$R \bowtie S \rightarrow mxn$

Values are  
not given  
any constraint

$R$

A	
1	
2	
3	
4	
5	
6	
7	

$S$

A	
1	
2	
3	
4	
5	

$\min(n, m)$

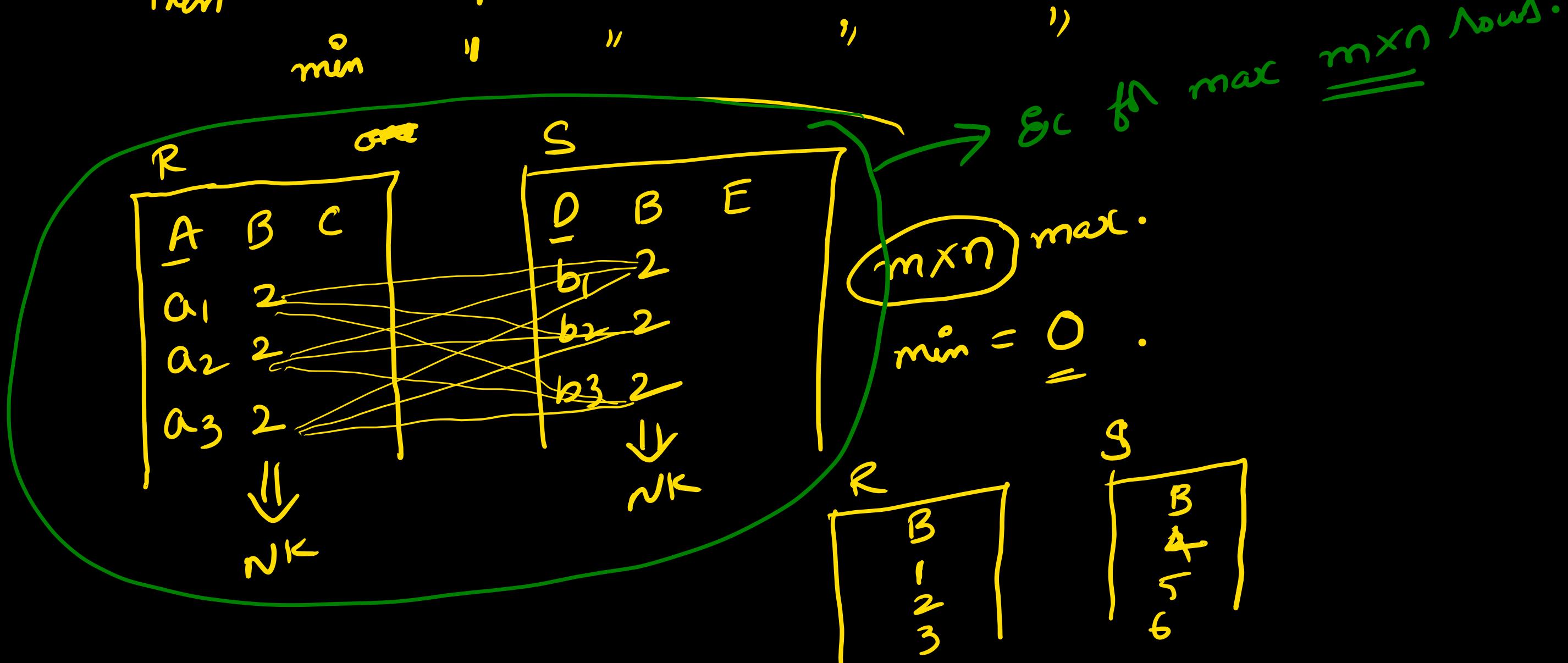
$\min = 0$

R	
1	
2	
3	
4	

S	
10	
20	
30	
40	

$R(A \underline{B} C)$  with  $n$  tuples .  $S(D \underline{B} E)$  with  $m$  tuples

Then max tuples in the result of  $R \bowtie S$  :



SQL    Structured query language

Sub languages of SQL:

{ DDL  
DML  
DCL

## Data definition language: (DDL)

used to define structure / definition of DB table

Ex: { CREATE TABLE  
DROP TABLE  
ALTER TABLE (add & remove attributes)  
PRIMARY KEY  
UNIQUE (alternate key)  
FOREIGN KEY (FK constraints)  
NON NULL  
CHECK (used to set range for attr values)}

most important & DDL  
there are more

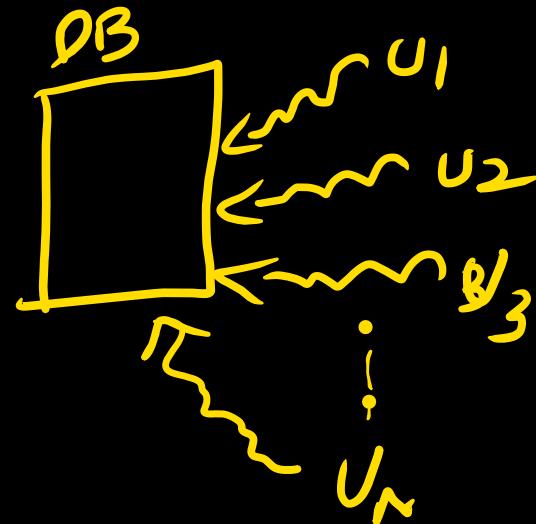
## Data manipulation language (DML)

Used to modify data records and access required data from DB tables.

{  
    INSERT INTO  
    DELETE FROM  
    UPDATE..... SET  
  
    SELECT \*  
    FROM Tables  
    where condition

## Data control language: (DCL)

- ① Data control for transaction mgmt [to avoid inconsistency because of concurrent users]



- ROLLBACK
- COMMIT
- CHECKPOINT

② Data control for user authentication: (Security)

GRANT ACCESS

REVOKE ACCESS

## SQL vs RA query:

(3) SELECT DISTINCT  $\underbrace{A_1, A_2, A_3, \dots, A_n}_{\text{Projection } (\pi) \text{ of RA}}$   $\Rightarrow$  Projection( $\pi$ ) of RA.

(1) FROM  $\underbrace{R_1, R_2, \dots, R_m}_{\text{crossproduct of RA}}$   $\Rightarrow$  crossproduct of RA

(2) WHERE  $\rho$ ;  $\Rightarrow$  Selection ( $\sigma$ ) of RA

---

$\pi_{\underbrace{A_1, A_2, \dots, A_n}_P} (R_1 \times R_2 \times \dots \times R_m)$

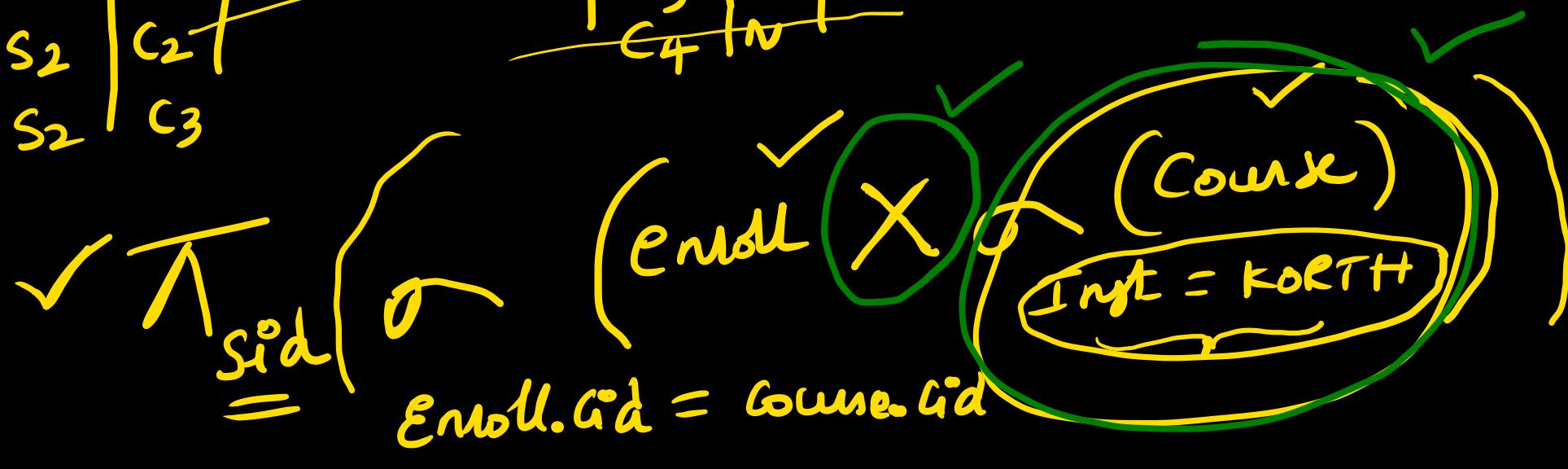
Enroll

sid	cid
S <sub>1</sub>	C <sub>1</sub>
S <sub>1</sub>	C <sub>2</sub>
S <sub>2</sub>	C <sub>2</sub>
S <sub>2</sub>	C <sub>3</sub>

course

cid	Inst
C <sub>1</sub>	K
C <sub>2</sub>	K
C <sub>3</sub>	N
C <sub>4</sub>	N

Sids enrolled in some course taught by KORTH.



S <sub>1</sub>
S <sub>1</sub>
S <sub>2</sub>

S <sub>1</sub>
S <sub>2</sub>

③ Select distinct SID,  
ENROLL, COURSE

① FROM

② where Inst = KORTH and  
enroll.cid = course.cid

Computer's will take.

oracle  
mysql;  
PostgreSQL;

## Basic SQL clauses:

[ ] → optional

✓ Select [Distinct] ✓ A<sub>1</sub>, A<sub>2</sub>, A<sub>3</sub>... A<sub>n</sub>

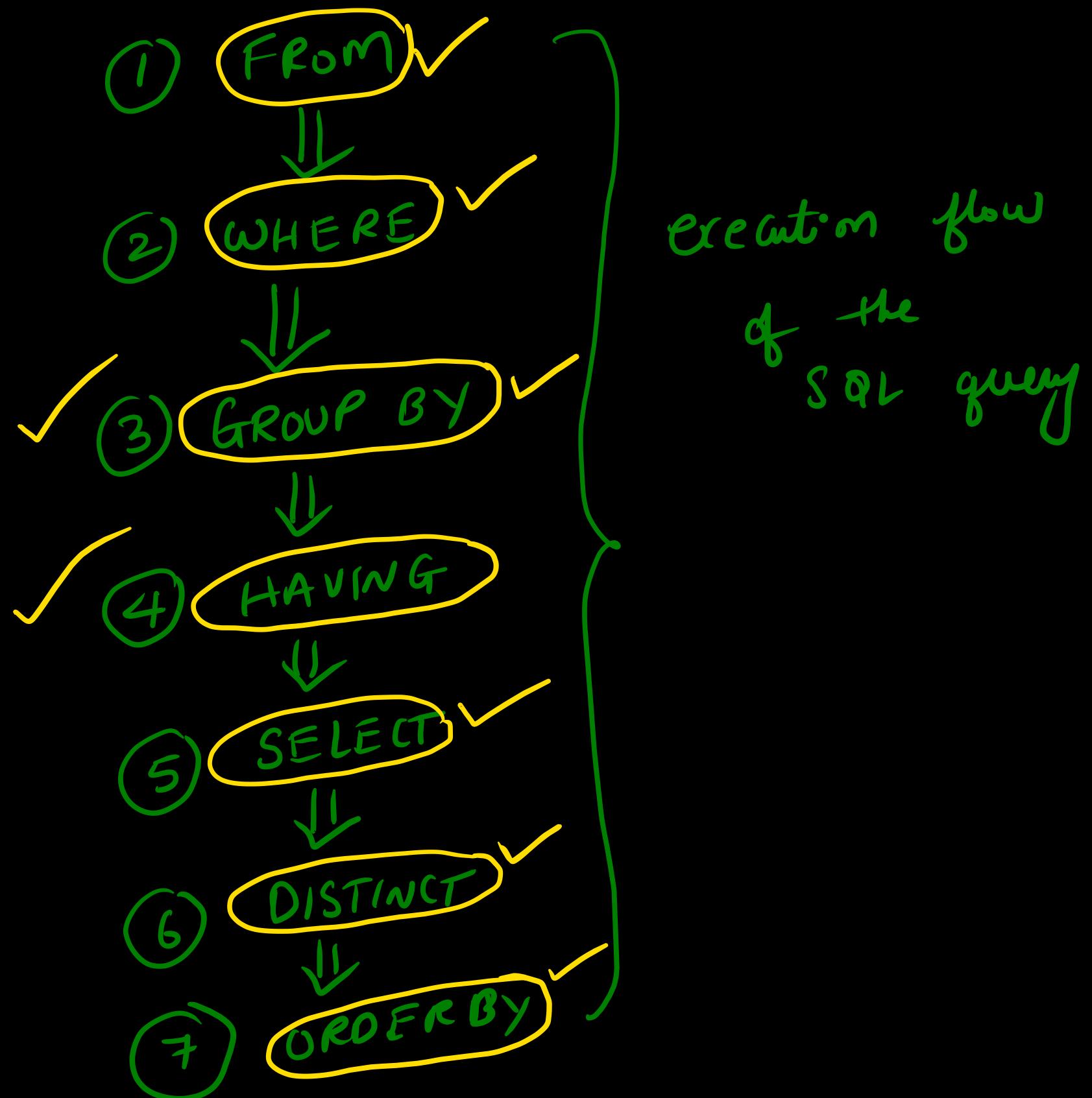
✓ FROM R<sub>1</sub>, R<sub>2</sub>..., R<sub>n</sub>

[ WHERE Condition ] ✓ grouping the rows together

[ GROUP BY attributes  
[ HAVING Condition ] ]

[ ORDER BY (attributes) [ Desc ] ]

↓ sorting the o/p



GROUP BY: It is used to group records based on specific attribute values.

SELECT ~~X~~, ~~X~~ AVG(Y) <sub>attributes</sub>  
FROM R

GROUP BY ~~X~~ → a set of attributes.

These rules vary from implementation to implementation

Ex: Oracle, MySQL, DB2.

what we are discussing in SQL Standard

- If GROUP BY clause is used, then according to standard SQL all the attributes of group by clause must be in select clause.
- We can use **aggregate functions** in SELECT clause
- Not allowed to select any other attributes in select clause

## Aggregate functions:

Count()  
Sum()  
Avg()  
Min()  
Max()

agg functions  
computes aggregation  
of non null values.  
null values are not  
considered.

$R$

Row ID

	A	B
1	20✓ 5.	
2	40✓ 10	
3	null	15
4	20✓ 20	
5	60✓ 25	
6	80✓ 30	
7	null	null

Select     $\text{sum}(A) A_1$ ,  $\text{sum}(\text{Distinct } A) A_2$ ,  $\text{Avg}(A) A_3$ ,  
 $\text{avg}(\text{Distinct } A) A_4$ ,  $\text{min}(A) A_5$ ,  $\text{max}(A) A_6$ .  
from  $R$  ;

$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$
220	200	$\frac{220}{5}$	$\frac{200}{4}$	20	80

Count (\*) : give the number of records.

Count (A) : Counts no of non null values of attribute A.

A	B
1	1
2	2
2	3
N	4
N	5
3	S

$$\text{Count}(\text{*}) = 5.$$

$$\text{Count}(A) = 3$$

A	B	C
N	N	N

A	B	C
a <sub>1</sub>	b <sub>1</sub>	80
a <sub>2</sub>	b <sub>4</sub>	90
null	b <sub>3</sub>	60
a <sub>2</sub>	b <sub>4</sub>	60
a <sub>1</sub>	b <sub>1</sub>	90
null	b <sub>3</sub>	40
a <sub>1</sub>	a <sub>2</sub>	null

GROUP BY (A)



A	B	C
a <sub>1</sub>	b <sub>1</sub>	80
a <sub>1</sub>	b <sub>1</sub>	90
a <sub>1</sub>	b <sub>2</sub>	null
null	b <sub>3</sub>	60
null	b <sub>3</sub>	40
a <sub>2</sub>	b <sub>4</sub>	90
a <sub>2</sub>	b <sub>4</sub>	60

Select A  
From R  
Group by A ✓

a <sub>1</sub>	✓
null	✓
a <sub>2</sub>	✓

Select A, Avg(c)  
From R  
Group by A;

A	Avg(c)
a <sub>1</sub>	85
null	50
else	75

(6 - 8) DBWR  
(8 - 10) Javg.