Query Processing (ORDER OF EXECUTION)

```
SELECT   col 1, col 2
FROM     tbl
WHERE    conditions col3
ORDER BY col 1, col 2 ....
LIMIT    5
```

S — 3
F → 1
W → 2
O → 4
L — 5

cost_to_customer quantity

[illegible] USING NEWLY GENERATED COLUMN IN WHERE

ISSUE WITH USING NEWLY GENERATED COLUMN IN WHERE

Q: Display orders where total cost < 50

```
Select
cost * quantity as total
from orders
WHERE    total < 50      → ✗

WHERE cost * quantity < 50   ✓
```

Write the correct order of execution of different clauses of a query

Q: Will the below query execute correctly ?

```
Select
cust-id,
product-id,
qty * cost   as total_cost ✓
from . - . .
ORDER BY   total_cost.
```

Often times we will be asked to provide summarized data.

aggregate functions

	NUMERIC	STRING	DATE
✓ count ( )	✓	✓	✓
✓ sum ( )	✓	✗	✗
min ( )	✓	✓	✓
max ( )	✓	✓	✓
avg ( )	✓	✗	✗

lexicograph ← [min ( ), max ( )]

aggregation vs **inline calculation**

↓ ↓

summarized view

creates a new column with some calculation.

no impact on rows

salary + commisions.

(inline calculation).

count (*)	count (col_name)	count(distinct col)
count all rows. null values are included	count all rows of a column except null values.	count only unique values in a column. doesn't include null values.

Imagine you work at the Amazon category team and asked to answer the following questions :

✓ 1) How many products are there in each category

✓ 2) What is the avg price of products in each category

✓ 3) Provide details of products purchased by more than 20 customers

GROUP-BY : Helps group data by various categories and provide aggregate metrics

EMP ID	NAME	DEP_ID
1	A	1
2	B	2
3	C	1
4	D	2
5	E	2

$$\xrightarrow[\text{dep-id}]{\text{GROUP BY}}$$

Find # of employees in each department

DEP_ID	COUNT
1	2
2	3

SYNTAX

no other column

SELECT [col] ✓ [aggregate function] ✓

FROM

→ WHERE

GROUP BY [col] . . . - - - .

ORDER OF EXECUTION

Where :

↓

group by

↓

Select

cust id	date	order id
1	1	1
1	2	2
2	1	3
2	2	4
2	2	5

cust date total orders

1	1	1
1	2	1
2	1	1
2	2	2

* Order of colums in group by does not affect the underlying aggregation.

IF YOU WANT TO APPLY A CONDITION ON

Q. WHAT IF YOU WANT TO APPLY A CONDITION ON AGGREGATE ?

↓

Find customers with more than 30 orders ✓

1   20 ✗

```
SELECT
→customer, count(*) ✓
→from orders
  WHERE count(*) > 30     ✗
→GROUP BY customer
```

```
SELECT
customer, count(*)
```

```
customer, count(*)
from orders
group by customer
HAVING count(*) > 30
```

having cannot be used without group by

ORDER OF EXECUTION

Quiz

WHERE

↓

GROUP BY

↓

HAVING

↓

$$\begin{tikzcd}
\text{Select.} \arrow[d] \\
\text{ORDERBY} \arrow[d] \\
\text{LIMIT.}
\end{tikzcd}$$

HAVING CLAUSE

Select

from

WHERE

group by product id

having $\underbrace{\text{count(*) > 200}}$

↓ ↓

Select name as <u>new_name</u> ✓

from

where $\underline{\hspace{4cm}}$

group by $\underbrace{\text{new\_name}}$ ↑

having

→ $\underbrace{\text{order of execution}}$

→ $\underline{\text{logical order of query}}$

$\boxed{\text{rating} > 3}$

TRUE → 1

FALSE → 0

FALSE 0
⋮ 0
⋮ 0
FALSE 1

avg
sum

nulls. but not the nulls values and
Imagine you work at ~~a~~ Amazon count
category team and asked to answer the nulls. th
following questions.

✓ 1) How many products are there in each category

✓ 2) What is the avg price of products in each category

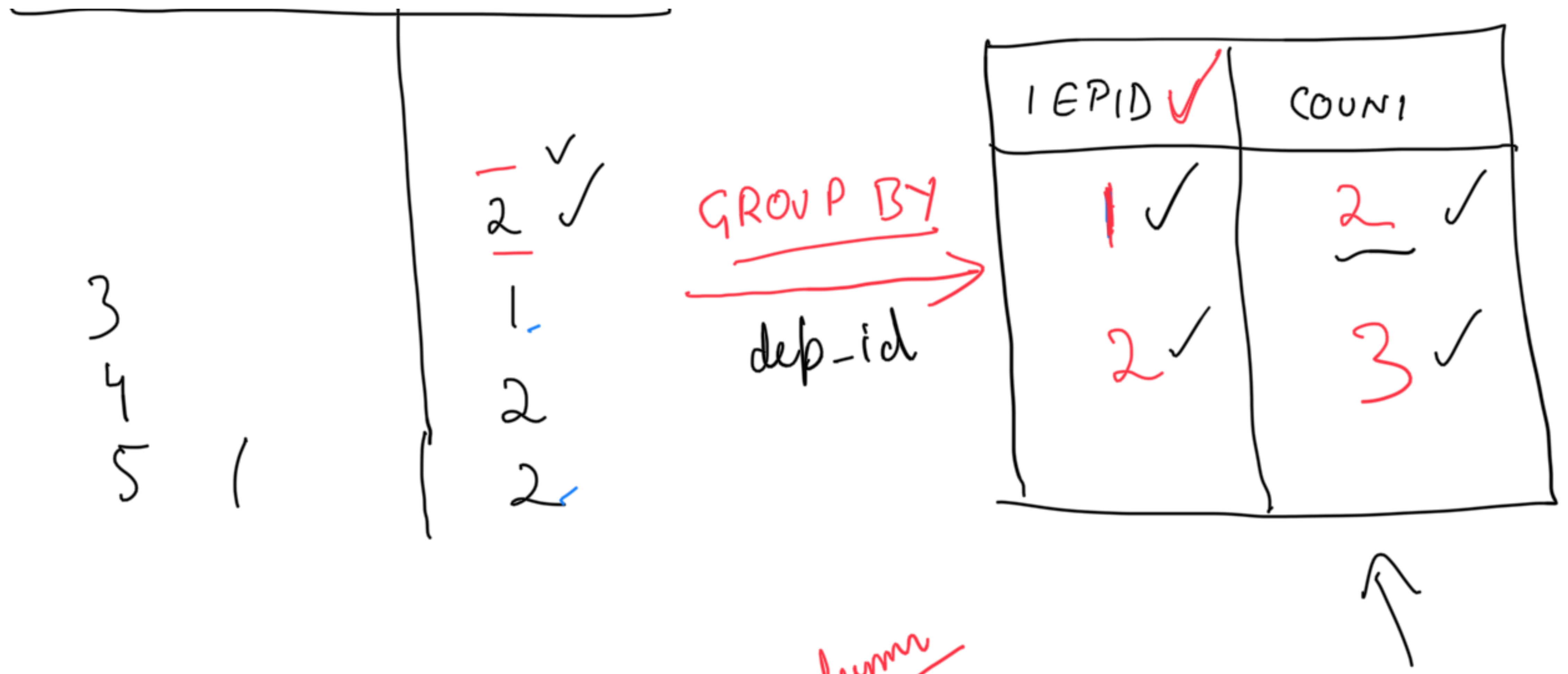
✓ 3] Provide details of products purchased by more than 20 customers

<u>GROUP-BY</u> : Helps group data by various categories and provide aggregate metrics

EMPID	NAME	DEP_ID
→ 1	A	1
→ 2	B	
→	C	
→	D	
→	E	

Find # of employees in each department

✓ name



SYNTAX

no other column

SELECT [col] ✓ [aggregate function] ✓

FROM —

→ WHERE —

GROUP BY [col]

ORDER OF EXECUTION

$$\begin{tikzcd} \text{where :} \arrow[d] \\ \text{group by} \ ✓ \arrow[d] \\ \text{Select} \ ✓ \end{tikzcd}$$

cust id	date	order id
1	1	1
1	2	2
2	1	3

2		
2	2	4
2	2	5

cust	date	total orders
1	1	1
1	2	1
2	1	1
2	2	2

* Order of colums in group by does not affect the underlying aggregation.

Q: Display departments with more than 2 employees

Emp_id	Dept_id
1	1
2	2
3	2
4	2
5	3
6	3

① GROUP BY →

Dep_id	count
✓ 1	1
✓ 2	3
✓ 3	2

② filter the rows.

Now filter the rows from grouped table

```
Select depid,    count(emp_id)
FROM   Employees
~~WHERE  count(empid) ≥ 2~~        → this is incorrect
GROUP BY   dep-id
HAVING   count (empid) ≥ 2  ✓
```

HAVING

special clause used to filter results after group by.

ORDER OF PROCESSING

WHERE ⟶ GROUP BY ⟶ HAVING

ORDER OF EXECUTION

Quiz

WHERE

↓

GROUP BY

↓

HAVING

↓

$$\begin{tikzcd}
\text{Select .} \arrow[d] \\
\text{ORDER Y} \arrow[d] \\
\underline{\text{LIMIT .}}
\end{tikzcd}$$