→ OLTP

↳ Transcation (Large)

↳

(Select *
from table (100M)
where city = 'Pune')

↳

Table scan → Optimize code

→ GB + Join is better than Join + GB

→ filter + Join ✓

or

Join + filter

2)

✓ Select *,

Select Column ✓

3) Windows function Vs Self Join

✓

id	a	b	c
1	Ag	B	C
2	Au	Be	Co
3	Al	Br	Cr
4	Ar	Br	Cd

Words

4 ✓ - 11 - - 11 - - 1 -

Book → Chapter - 17

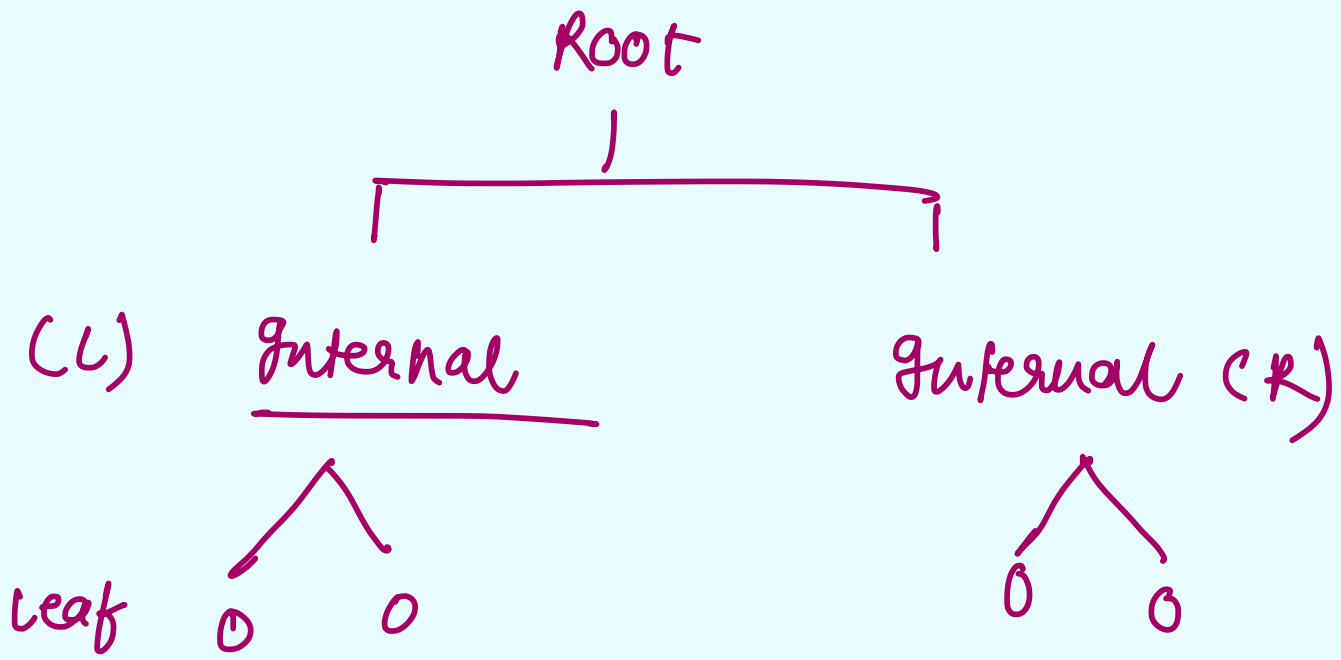
Index Page

NC

Pg no : Name

x y 2

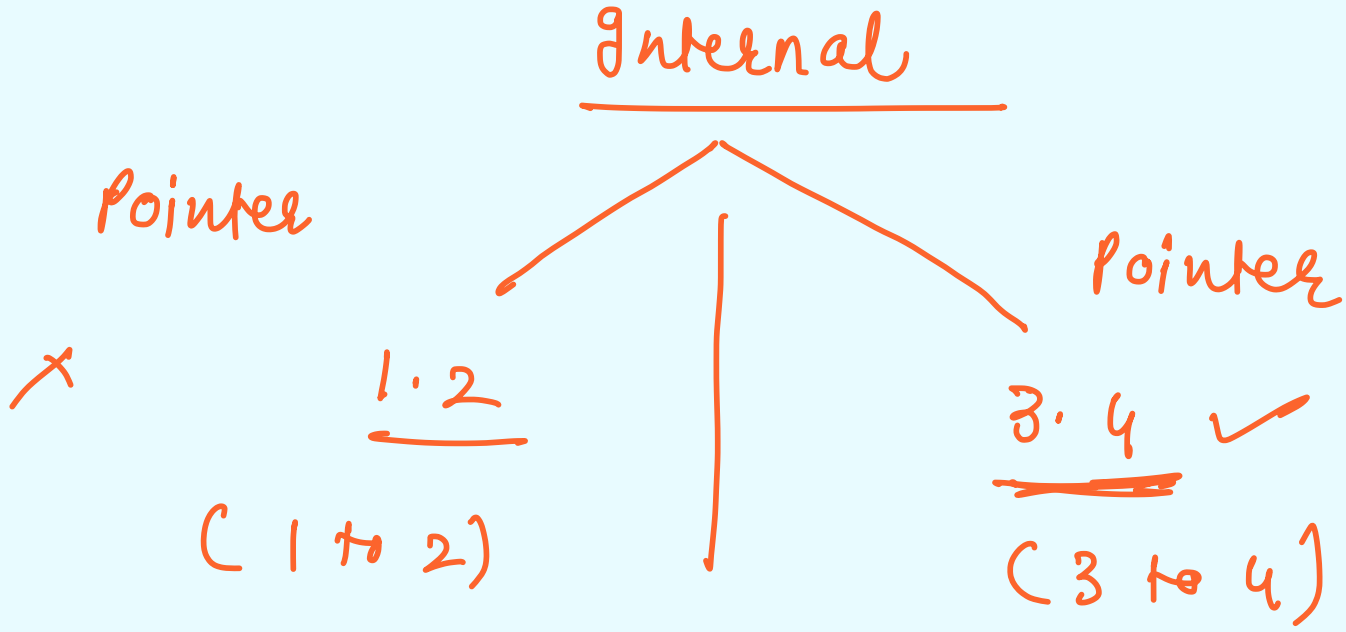
MySQL → Bt Tree Index



← Root → entry point

(No data)

Key → 1·4 → id → 1 to 4



leaf 1 : id=1 → (Ag, B, C)

2 : id=2 → (Au, Be, Co)

id=3 ✓

id=4 ✓ → ( )

```
select *
from orders
where 'id = 4
```

✓

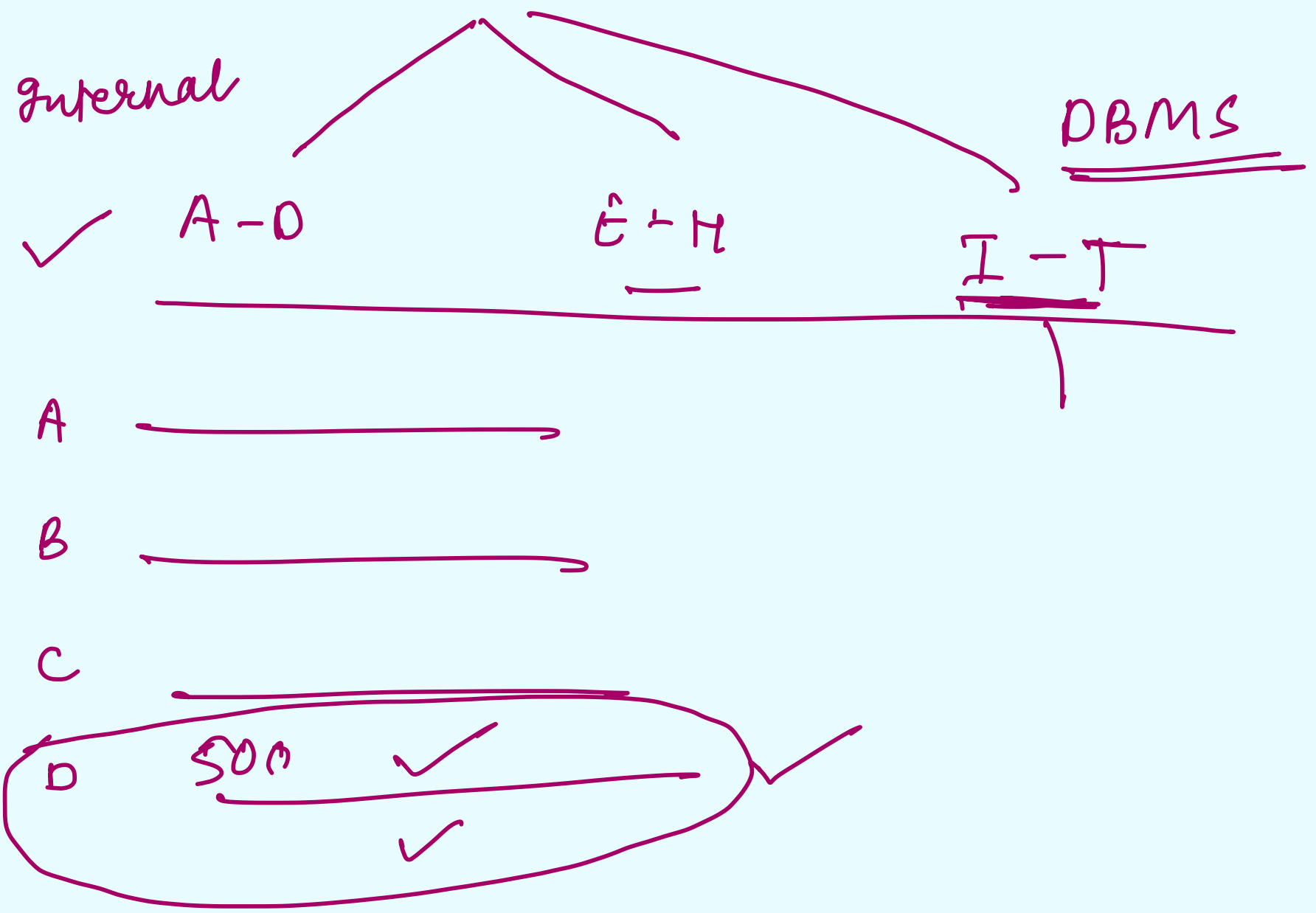
Range

where id between 2 and 4

left 2 → left 3 — left 4 ~~left 5~~

Library

→ Root (A to Z)



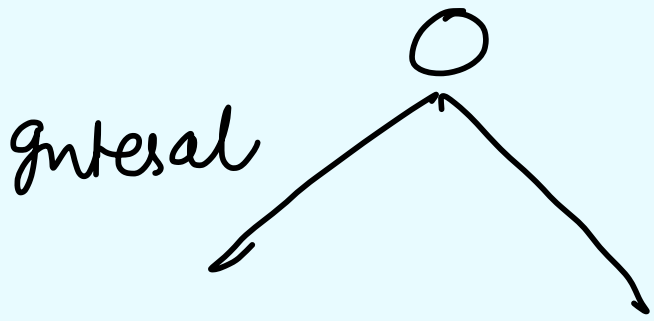
Cluster index

id = 1 (5)

Leaf - (data)

Row data

non index Cluster



leaf id = 10,

Point to row

	C	NC
1. Leaf contain	All rows	Key + Pointer
2. steps (look up)	1 ✓	2
3.	(Primary)	Region, Category

Apply Indexing on Multiple col

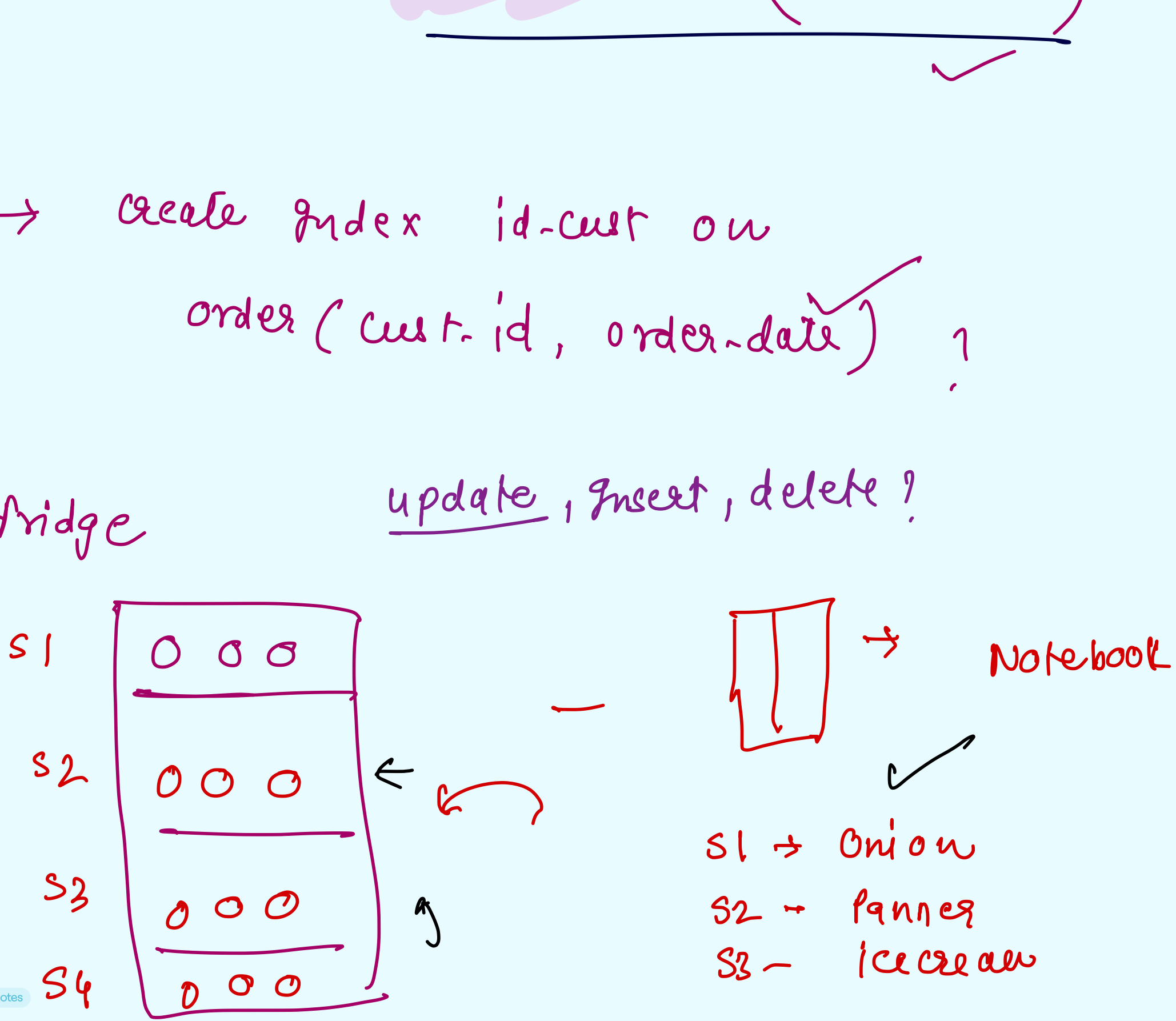
yes ?

```
select *
from order
where cust_id = 101
and order_date >= ('2025-10-05')
```

→ create index id_cust on order (cust_id, order_date) ?

Fridge

update, insert, delete ?



New stock

eggs

↓

update ?

→ S,16 in table (10 gb) ✓

↳ index (30 gb) cost ?