Agenda :-

- indexes ?
- How index works ?
- cons of Indexes
  - Indexes on multiple columns
  - Indexing on strings

Read ←

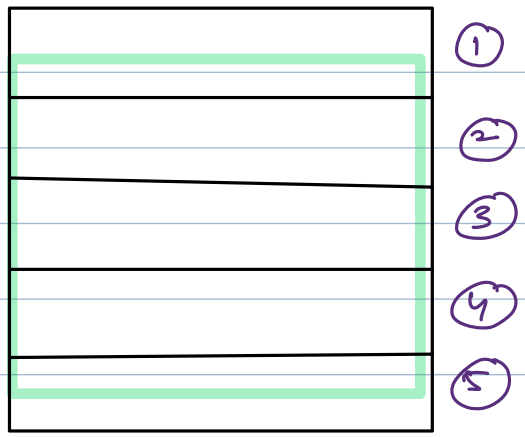
select * from students where id = 1000;

↓

O(n)

Book → table of content

indexes - at last




word , page no

↓

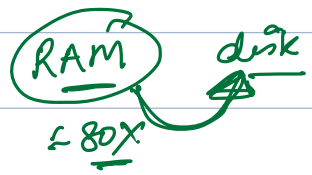
help to find keyword

very easily

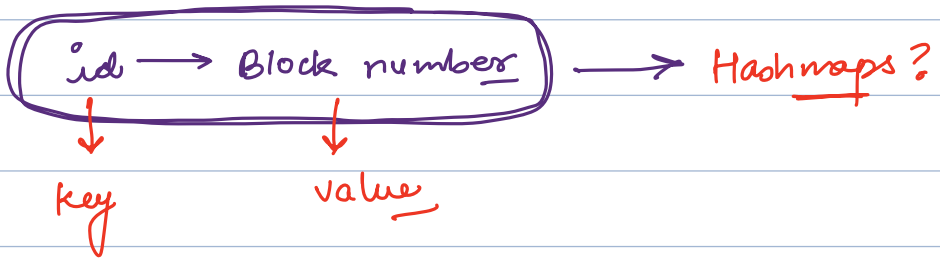
indexes — DB: Find the block very fast

↓

Reduce # disk accesses!



# How indexes work?




W/o HM: 5 blocks

W HM: 1 block.

select * from students where name = "MOHIT";

<name, block no>

some multiple Nomes

<name, list<block>>

| Key (Name) | BN |
|---|---|
| Mohit | 3, 7 |
| Keshav | 1, 2, 7, 8 |
| Rishab | 6 |

select * from students psp >= 50.1 and psp <= 70.9;

Are HM ordered?

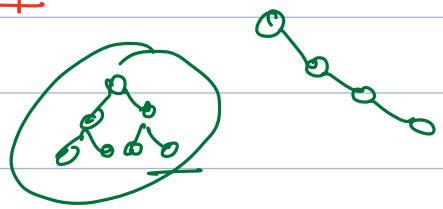
No

check a value → HM ✓

range → B/B+ Trees

Hashmap + ordered ?

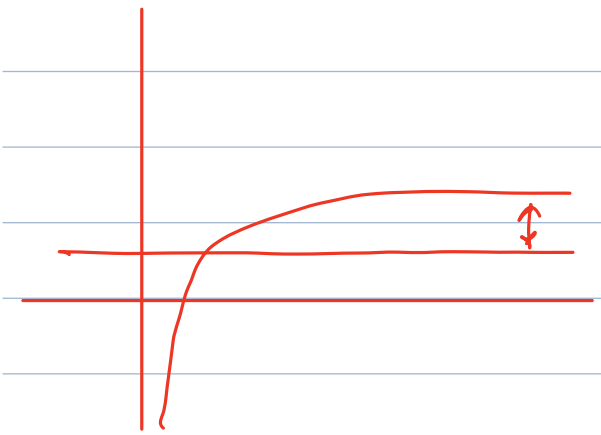
keys are stored in sorted manner → TreeMap / ordered_map



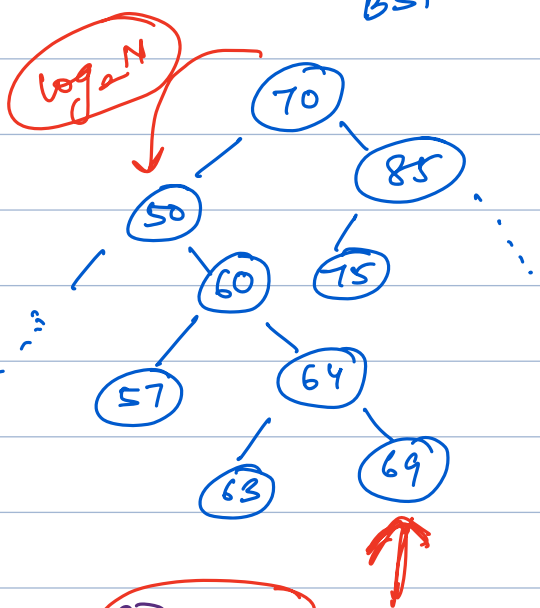
BBST

AVL, Red Black Trees ← Balanced Binary search Trees.

condensed → $H = \log_2 N$



BST




a inorder → sorted order

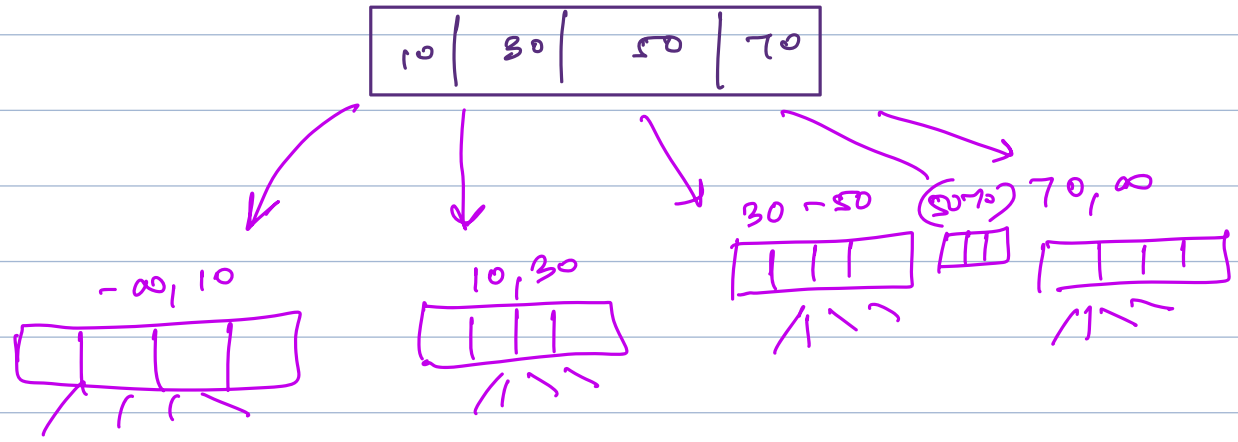
50-69

Indexes — B+ Trees

Balanced <= x children

BBST → Binary

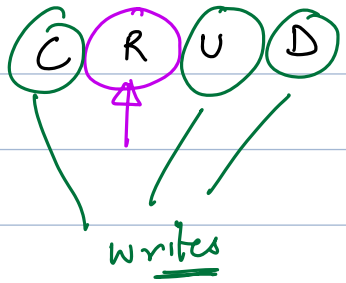
$x = 4$




B+ $h \downarrow$ BBST

TC Improves

## Cons of Indexing




change the data on disks → update the indexes.

writes are slower

Why don't we keep indexes in RAM?

↓

Need to persist

disks → ↑ storage space

Don't create indexes prematurely

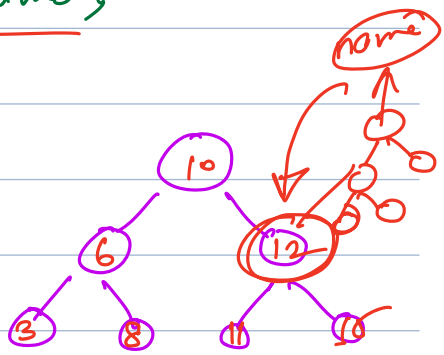
↓

only when you see the need

# Multiple columns

PK

_ _ _ . _ _ _ order by id, name ;




name ✗

id ✓

(name, id) → order is important

(id, name) ✓

| query | index | |
|---|---|---|
| name | psp | ✗ |
| name | (name) (psp) | ✓ |
| name | (psp, name) | ✗ |
| name | (name, psp) | ✓ |
| name = _ & psp = _ | (psp, name) | ✓ |
| | (name, psp) | ✓ |

# Indexes on strings

select * from user
where email = "___";

- size of string ↑
- string comparison is slow

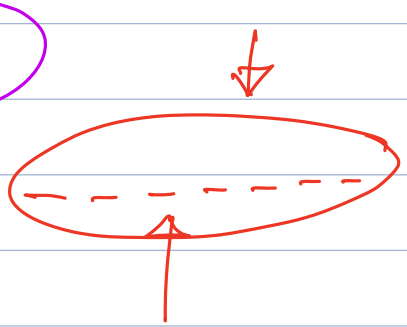
80-90%

@gmail.com

mohit.share@gmail.com

a [_,_,_,_]
b [_,_,_]

2B record



ch
26 * 26 * 26
676 * 26 = 17576

$26^4 = 4 * 10^5$

$26^5 = 1 * 10^7$

$26^6 = 3 * 10^8$

$26^7 = 8 * 10^9$