#ASLI ENGINEERING

How indexes make a database read faster?

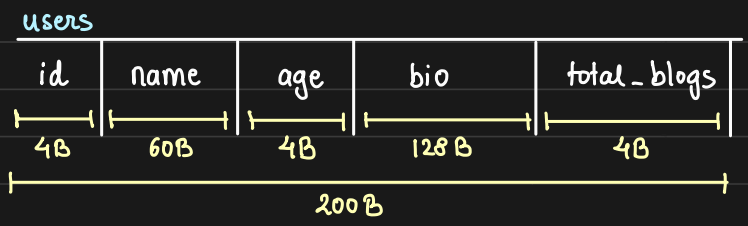
SWIPE

BY

ARPIT BHAYANI

How indexing makes your DB faster

Database is a collection of records.



Users table has 5 columns and each record of the table will be 200B long.

Say the users table has 100 rows
total size = 200 x 100 = 20000B

users

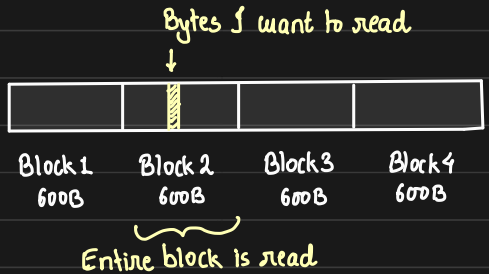
id	name	age	bio	total_blogs
1	A	23		7
2	B	21		10
3	C	22		8
4	D	23		2
5	E	22		1
6	F	24		15
⋮	⋮	⋮	⋮	⋮

How reads from disk happen?

↳ One disk read = one block read

600B

Even if we read one byte, it will read the entire block that contains the byte



Users table in blocks

Each record is 200B

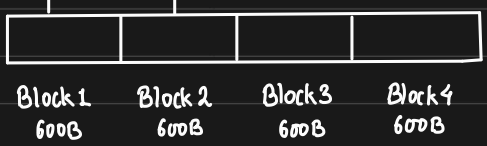
Block size is 600B

id	name	age	bio	total_blogs
1	A	23	———	7
2	B	21	———	10
3	C	22	———	8
4	D	23	———	2
5	E	22	———	1
6	F	24	———	15
⋮	⋮	⋮	⋮	⋮

Each block can hold 3 records

=

The entire 'users' having 100 rows can fit in $\frac{100}{3} = 33.3 \approx 34$ Blocks



So, reading the entire table will require the database engine to read 34 Blocks from the disk.

If hypothetically Read 1 Block takes 1 sec

time to read entire table = 34 sec.

Query: Find all users with age == 23

id	name	age	bio	total_blogs
1	A	23		7
2	B	21		10
3	C	22		8
4	D	23		2
5	E	22		1
6	F	24		15
⋮	⋮	⋮	⋮	⋮

Flow:

- ↳ iterate table row by row
  - ↳ block by block
- ↳ read the block in memory
- ↳ check age == 23 on each record
- ↳ if yes, add the record to an output buffer
- ↳ if no, discard
- ↳ return the output buffer

Time taken to answer this query is same as time taken to read the blocks

34 blocks = 34 sec

Let's see how indexes make this fast...

Indexes

↳ Indexes are smaller referential tables that holds row references against the indexed value

On a very high level

Eg: Say we create an index on 'age' it would look something like this

users_age_idx

age	id
21	2
22	3
22	5
23	1
23	4
24	6
⋮	⋮

Ordered by the indexed value

Each entry in index is 4B + 4B = 8B big

age id

There will be 100 entries in the index, so total size of index will be 8 × 100 = 800B

One disk block = 600 B

So, index will require only 2 blocks on the disk.

Query: Find all users with age == 23 with index

Flow:

- ↳ iterate index (worst case complete index)
  - ↳ block by block
- ↳ check age == 23 in entries
- ↳ if yes, add the 'id' in a buffer
- ↳ if no, discard
- ↳ for all the relevant id in the buffer
  - ↳ read the records from the dist
  - ↳ add to an output buffer
- ↳ return the output buffer

} Read the index and note the relevant ids matching criteria

} For the relevant ids fetch the actual records from the disk

let's walk through on our data & example

Ordered by the indexed value →

age	id	
21	2	
22	3	
22	5	
23	1	*
23	4	*
24	6	
⋮	⋮	

To find the relevant ids,

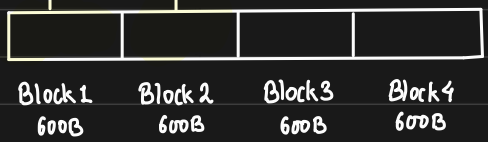
we read the index and filter out relevant ids

Blocks read = 2 ↑ worst case

Relevant IDs with age == 23 are [1, 4]

Users table in blocks

id	name	age	bio	total_blogs
1	A	23	———	7
2	B	21	———	10
3	C	22	———	8
4	D	23	———	2
5	E	22	———	1
6	F	24	———	15
⋮	⋮	⋮	⋮	⋮



We need row ids 1 and 4 which are present in Block 1 & Block 2. So, we read the two blocks from the disk, extract the records and return

Blocks read = 2

Total Blocks read = 2 (Index) + 2 (record) → 4 block ~ 4 sec

Time take without index = 32 sec
with index = 4 sec
} 8x