

UNIT-5

Index data structures:

→ Indexing is a DS technique that helps to speed up data retrieval.

→ We can quickly locate and access the data in the database.

→ Index are combination of two columns.

→ Index can locate the desired record directly without First column:

Scanning each record in file.

- search key

- contains primary key or candidate key of the table.

- values can be either sorted or not

- If sorted accessed easily.

second column:

contains address of disk block where

we find key value.



↑
primary key

↑
set of pointers holding address of the disk block

Types of indexing:

1. Primary index

- dense index

- sparse index

2. Secondary indexing

file size ↑

3. Clustered indexing

search time ↑

4. Multilevel indexing

- B-tree indexing

- B+tree indexing

data sorted - ordered file

unsorted - unordered file

primary index	clustered index
secondary index	secondary index

key (unique)

non-key (not unique)

1 6
2 5
3 4
4 3
5 2
6 1

1
2
3
3
4

Types of Indexing

1. primary indexing

primary indexing only has two columns,

First column - primary key values which are the search keys.

second column - pointers that contain address of data block of the search key.

The table should have one-to-one relationship

b/w the records in index file & data block.

• Dense index

There is an index record that contains

a search key and pointer for every search key value in the data file.

- fast & requires more memory to index records

1	
2	
3	
4	

Index record

1	John	25
2	Jack	24
3	Amey	18
4	Ellena	29

data block

→ sparse index:-

There will be only few index records that point to the search key.

First, the index record starts searching sequentially, by pointing to a location of a value in the data file until it find actual location of the search key value.

- Time consuming

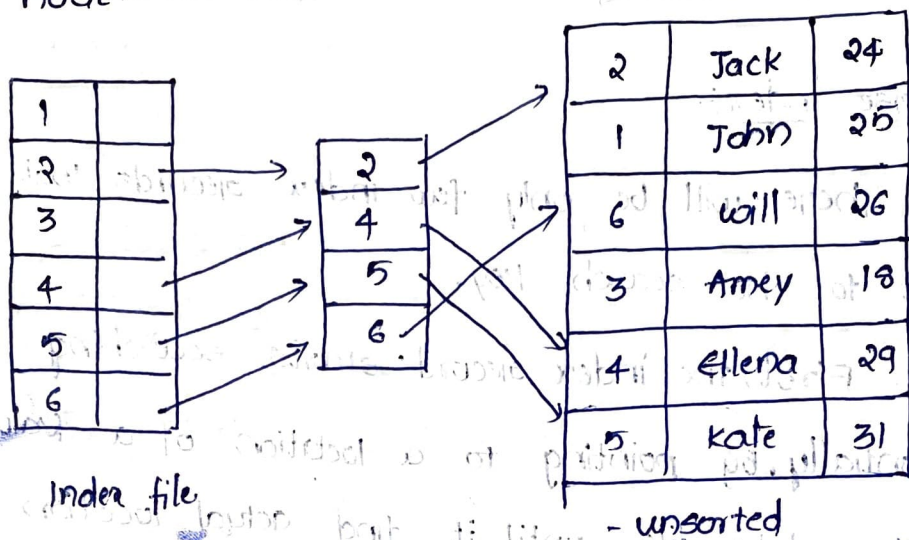
- less memory to store index records.

1		1	John	25
4		2	Jack	24
6		3	Amey	18
		4	ellena	29
		5	kate	31
		6	Will	26

2. Secondary indexing - {candidate} key

First column can be non-key & values along with respective pointer which has address to the location of value.

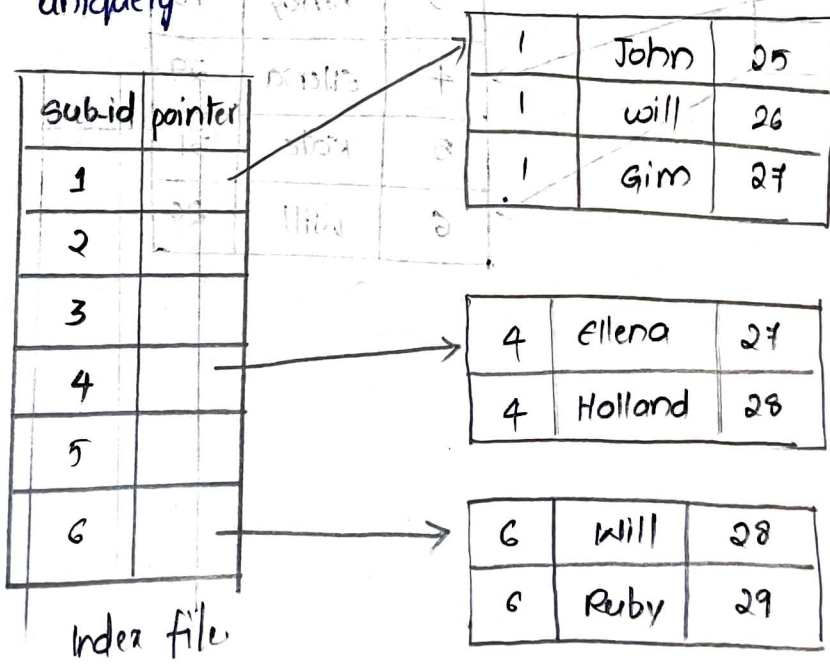
→ Index and data communicated through intermediate node.



3. Clustered indexing -

First column has non-key values along with respective pointer to the search key value.

→ We combine two or more columns together to get the unique values to identify data, uniquely.



	<u>search key</u>	<u>data file</u>
primary index	primary key	ordered
secondary index (non-clustered)	candidate key	unordered
clustered index	non-primary key	ordered

4. Multilevel Indexing

If the primary index does not fit in the memory, multilevel indexing is used. The main data block breaks down into smaller blocks that can be stored in main memory.

B+ tree indexing

B - tree indexing

Index data-structures:

- file data entries are organized in two ways.

i) Hash based indexing uses search key

ii) Tree based indexing

Comparison of file organisation:-

B+ Tree:-

→ B+ tree balanced BST

→ leaf nodes have actual data pointers and are at same height.

→ B+ tree can support random & sequential access.

→ leaf nodes are linked using linked list.

Structure:-

- B+ tree, every leaf node is at equal distance from the root node.

- The B+ tree is of the order n where n is fixed for every tree.

Internal node :-

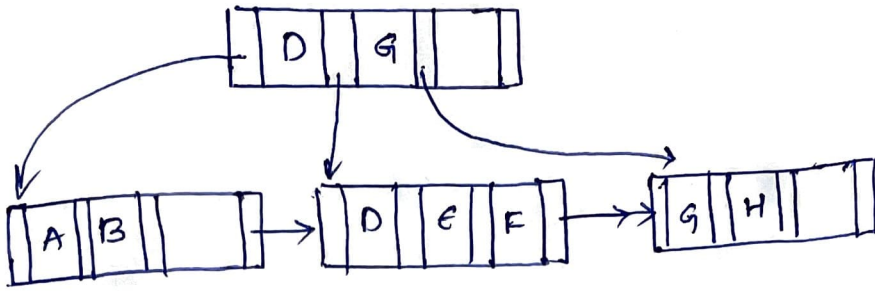
- contains atleast $n/2$ pointers except root node
atmost n pointers

leaf node :-

leaf node can have

- atleast (min of) $n/2$ record pointers & $n/2$ (key values)
- atmost (max of) n record pointers & n key values
- every leaf node of B+ tree contains one block pointer p to point next leaf node.

~~D | G |~~



B⁺ tree:

- B⁺ trees are filled from bottom and entry is done at leaf node.