

Database Management Systems

Lecture



Indexes





Contents

- Background
- Indexes
- Types of indexes
 - Primary index
 - Clustering index
 - Dense and sparse indexes
 - Secondary indexes



Background

- Data records are stored in files
- Files are stored on disks
- Disks are sequential access media
- How to locate and access a specific record on the disk efficiently?
- Using Indexes!



- How can we search a specific word in a book?
- Two methods
 - Start searching from the first page and go until we find the required word, or the end of the book
 - Use the index at the end of the book...
- Why is the first choice a bad one?
 - It is linear search – not efficient at all!
- What makes using an index a better option?



- The index contains:
 - A list of all important words used in the book in the alphabetical order
 - For each word, the page number(s) in the book where that word appears
- To find a particular word in the book, we
 1. Search the index for that word
 2. Get the page number in the book where that word appears
 3. Search that page for the required word



- Using index we are still doing two searches, but why is this a better option than searching the whole book?
- Two reasons:
 - Index is a sorted list of words – we can quickly find an item in a sorted list
 - The second search – locating a word on a page – requires considerably less time than searching in the whole book!



Indexes

- Databases use the concept of index for quickly locating a record on the disk
- A file contains several records in a certain structure
- A record contains several fields (attributes)
- For each file, the database stores an index structure
- The index is generally defined for a single field
 - Called the indexing field
- The index typically contains, in sorted order, all values of the indexing field and the addresses of disk blocks where the record containing those values are stored



Indexes

Index File

Search key	Block Ptr
063	●
098	●
...	
123	●
178	●
...	
542	●

Reg	Name	Address	Major
123			
285			
063			
197			
098			
542			
.			
.			
.			
268			
178			
204			

Data File

Disk blocks

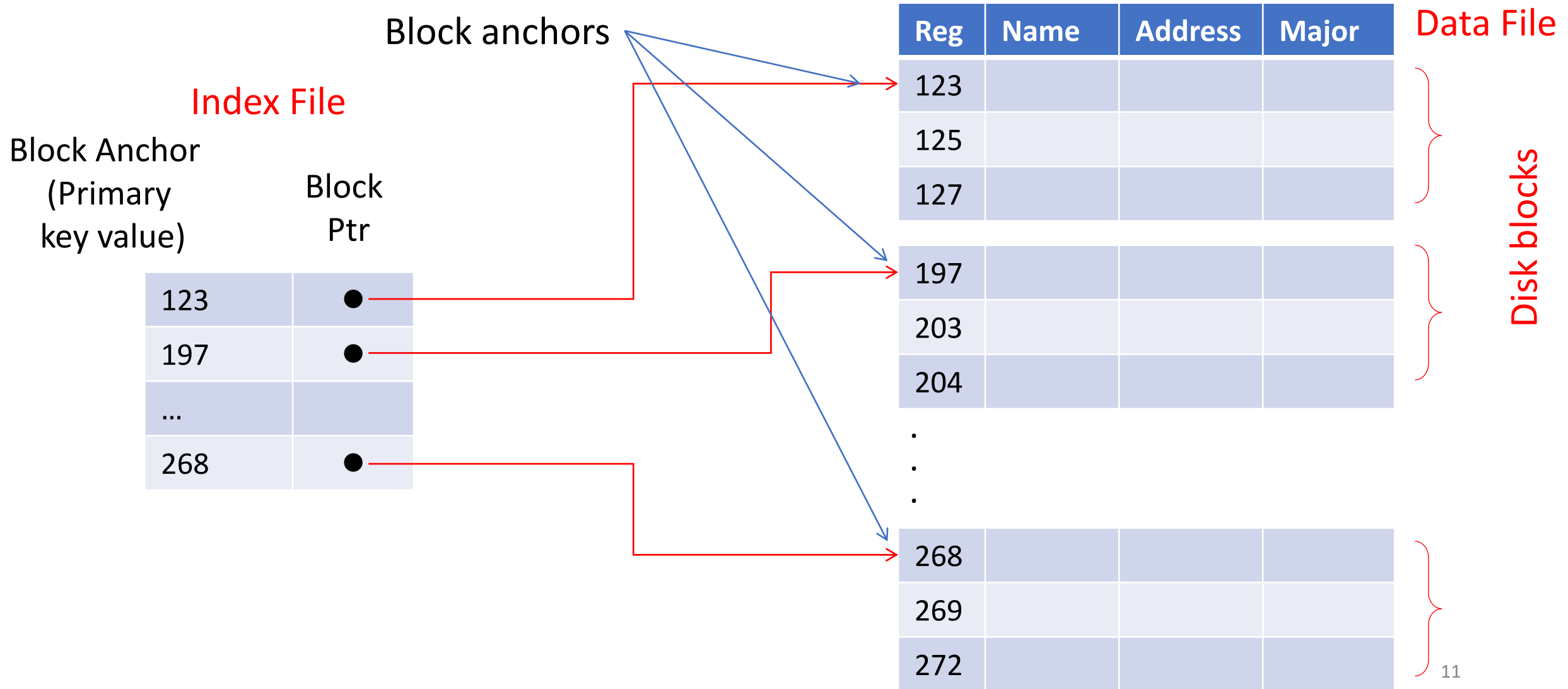


Types of Indexes

- Note that if the data file is also physically stored in the disk in sorted order according to the indexing field, then we don't need to store value of each indexing field in the index...
- Why? We can just put the first value in a block – called the block anchor – in the index
- When looking for a value, we search values in the index until we find a value which is greater than the required value
- We go to the previous block – that's where our required value is stored



Types of Indexes: Primary Indexes





Types of Indexes: Primary Indexes

- Such an index is called a primary index, where:
 - We have one specific field (attribute) which is the ordering key field for the ordered data file, and
 - There is a unique value for that attribute in each record
 - Question: How many primary indexes can we have for a file?
 - At most one! Because the file can be physically stored according to sorting order for just one field!
- Since index has fewer entries than the data file, performing a binary search on the index is more efficient



Classification of Indexes: Dense and Sparse

- **Dense indexes** are those in which we have an entry for every value in the indexed field
 - The number of index entries is equal to the number of unique values in the indexing field
- **Sparse indexes**, on the other hand, have index entries for a few search keys
 - The primary index that we just discussed is an example of sparse index

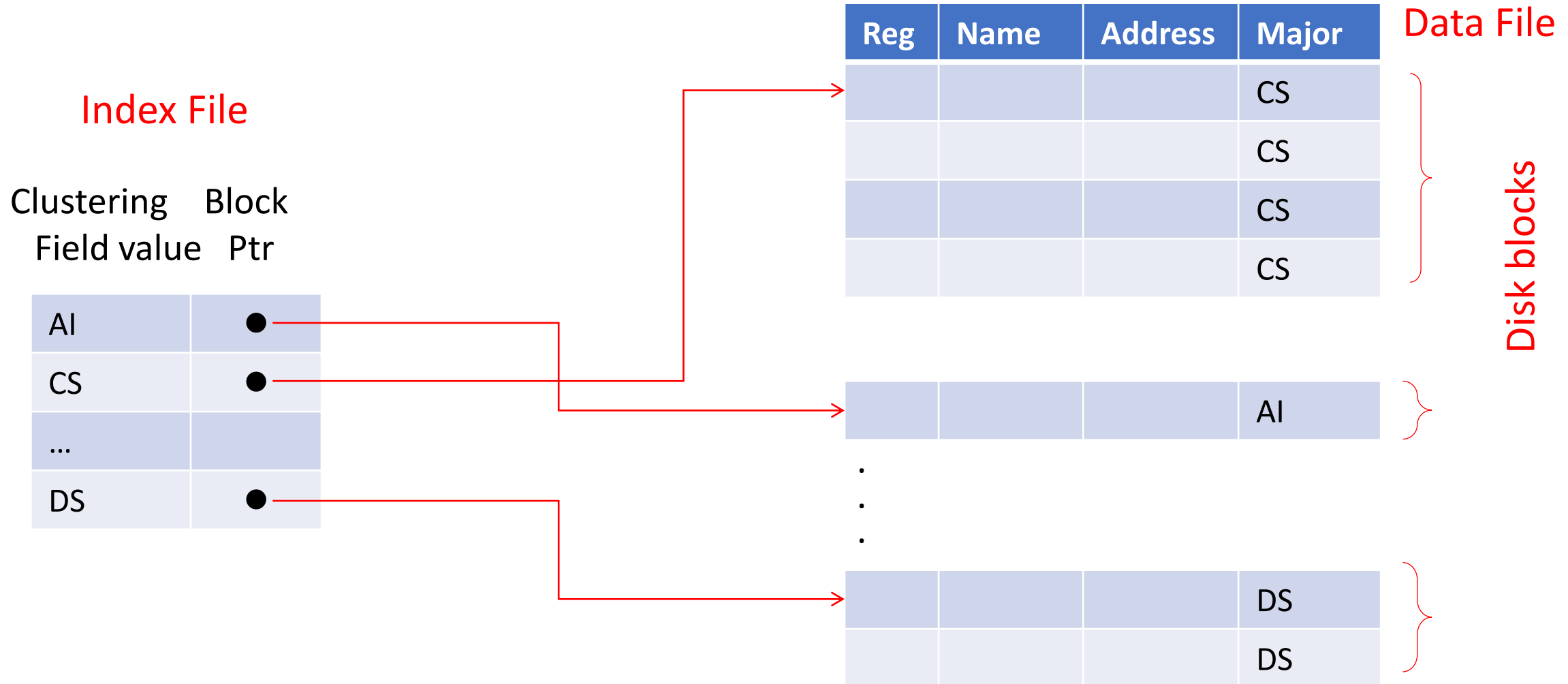


Types of Indexes: Clustering Indexes

- If file records are physically ordered on a non-key field, then multiple records can have the same value in the indexing field
- Such an index is called a **clustering index**
- For example, if we index on 'Major' attribute for the 'Student' table, then it'll be a clustering index because we have a lot of duplicate values in the 'Major' column
- The index file in such a case will have ordered, unique search values
- The second column in the index will have the address of the block that has all the records containing the value being searched
- This is another example of a sparse index



Types of Indexes: Clustering Indexes





Types of Indexes: Clustering Indexes

- Do you see a problem in this kind of data file?
- The block size is fixed, but there can be more records with a given value in the indexing field than can be stored in the block...
- The solution is to store as many records as can fit in a block, and then if another record is to be stored, new block is created and its pointer is stored in the end of the previous block
- Sounds familiar?
 - Yes! The idea is similar to what we did in Data Structures– linked lists!



Types of Indexes: Clustering Indexes

Index File

Clustering
Field value Block
 Ptr

AI	●
CS	●
...	
DS	●

Reg	Name	Address	Major
			CS
			CS
			CS
Block Ptr ●			
			CS
			AI
			...
			DS
			DS

Data File

Disk blocks



Types of Indexes: Secondary Indexes

- Sometimes it is required to speed up the access by maintaining an additional index for a non-key, non-ordering field
- The indexing field may have duplicate values in a lot of records, e.g., Student's last name, Employee's City, etc.
- The data file is assumed to be physically un-ordered with respect to the indexing field
 - A primary index already exists for the file according to the ordering field of the file...

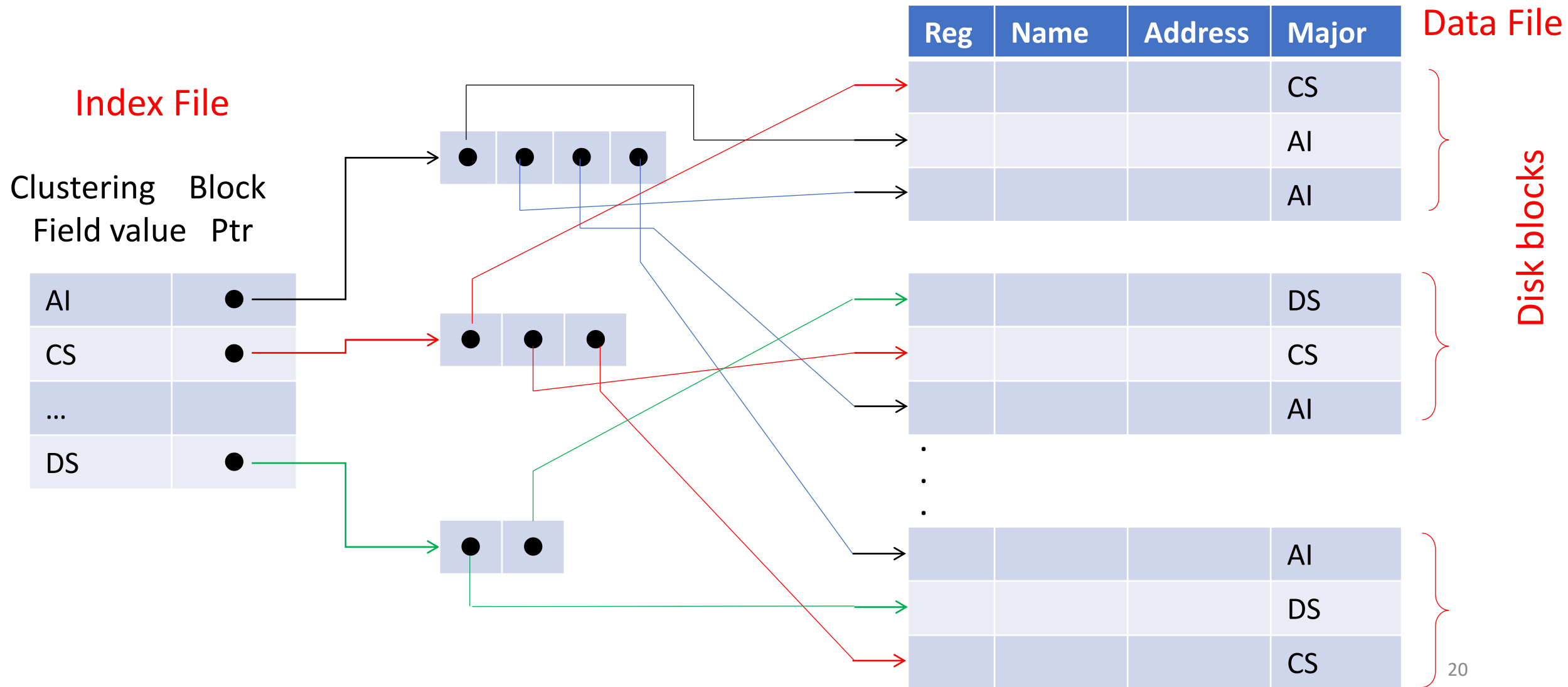


Types of Indexes: Secondary Indexes

- The first column of each entry in the index file contains the ordered indexing field, as usual, but the second entry holds a pointer to a record of pointers
- Note that each value in the indexing field appears in multiple records, so each pointer in the record of pointers points to a specific record where the search value is present



Types of Indexes: Secondary Indexes





Thanks a lot

