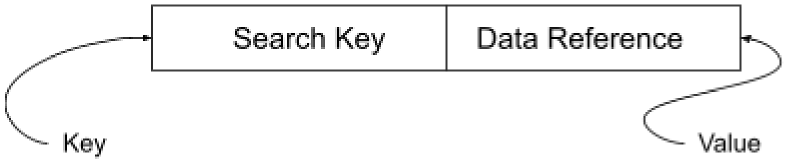
What is Indexing?

Indexing is a method to help us locate a record/data present in the memory faster and quicker. It gets this done by minimizing the number of disk accesses required when we process a query.
It is a Data Structure that we use to perform the above stated operation.
Indexes contain few database columns:

- The first column is the Search Key, which contains the copy of the Primary key of the table so that the data access time could be reduced which means data can be accessed quickly. Order of the key may or may not be sorted.
- Second column is the Data Reference. It contains pointers holding the address of the disk block where the value corresponding to the key is stored.



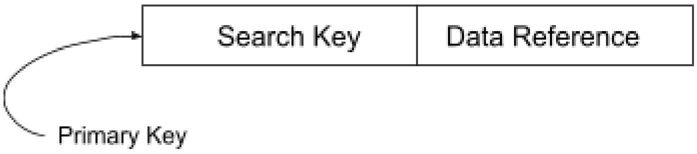
Why do we need Indexing?

Usually, many queries reference only a small proportion of the records in a file.
Suppose, you need to find all instructors in the “Anatomy department”.

- Now, It is quite inefficient for the system to read every tuple in the instructor relation to check if the dept_name value is “Anatomy”
- To quickly access, our DBMS stores additional structures associated with the files called indices.
- To retrieve Professors records given a department name, the database system would look up an index to find on which disk block the corresponding record resides and then will fetch the disk block, to get the appropriate Professor record.

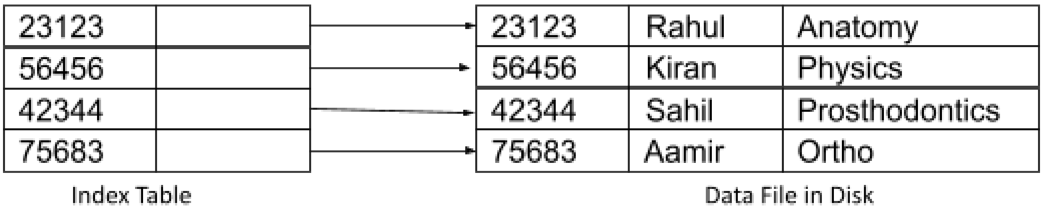
Types of Indexing:

➢ **Primary Index:-** It is the index that is created and ordered on the basis of the primary key of the table. It is further of two types Dense and Sparse Index.



Search Key	Data Reference

- **Dense Index:-** An index record appears for every search-key value in the file, i.e. number of records in the dense index table is the same as the main table. In this the structure contains the search-key value and a pointer to the data record with that search-key value.
  This type of indexing is not space friendly, as it takes extra space to store the index table.



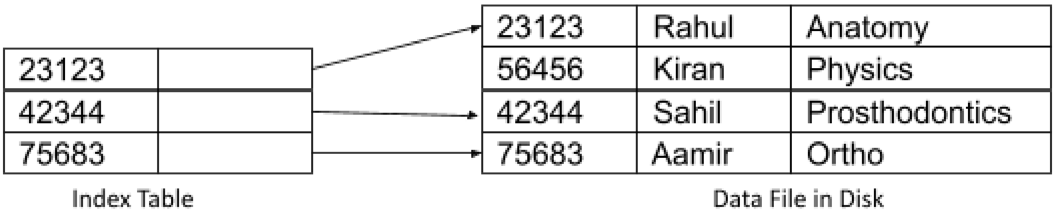
Index Table

23123	
56456	
42344	
75683	

Data File in Disk

23123	Rahul	Anatomy
56456	Kiran	Physics
42344	Sahil	Prosthodontics
75683	Aamir	Ortho

- **Sparse Index:-** An index record appears for some of the search-key values in the file**.** To locate a record when this type of indexing is performed on the datafile, we lookout for the largest value key in the table which is less than or equal to the our desired search key and once it is found, we then search that key sequentially starting from the row where the pointer of the index table pointed us till we find desired key and the record corresponding to it.   The space efficiency for this is better than that of Dense but it is slower than dense index.



Index Table

23123	
42344	
75683	

Data File in Disk

23123	Rahul	Anatomy
56456	Kiran	Physics
42344	Sahil	Prosthodontics
75683	Aamir	Ortho

.

Difference Between Dense and Sparse Indexing:

Dense Index	Sparse Index
Space taken for the index table is large.	Space taken for the index table is smaller.
Time taken to locate the record is less in comparison	Time taken to locate the record is more.
The records in the data file are in specific order and need not be in any kind of cluster or chunk.	The records in the data file are in specific order but the data records are in a cluster or chunk. (i.e. pointers from the index table point to certain data records, and all records between those pointers are considered in one cluster or chunk.)