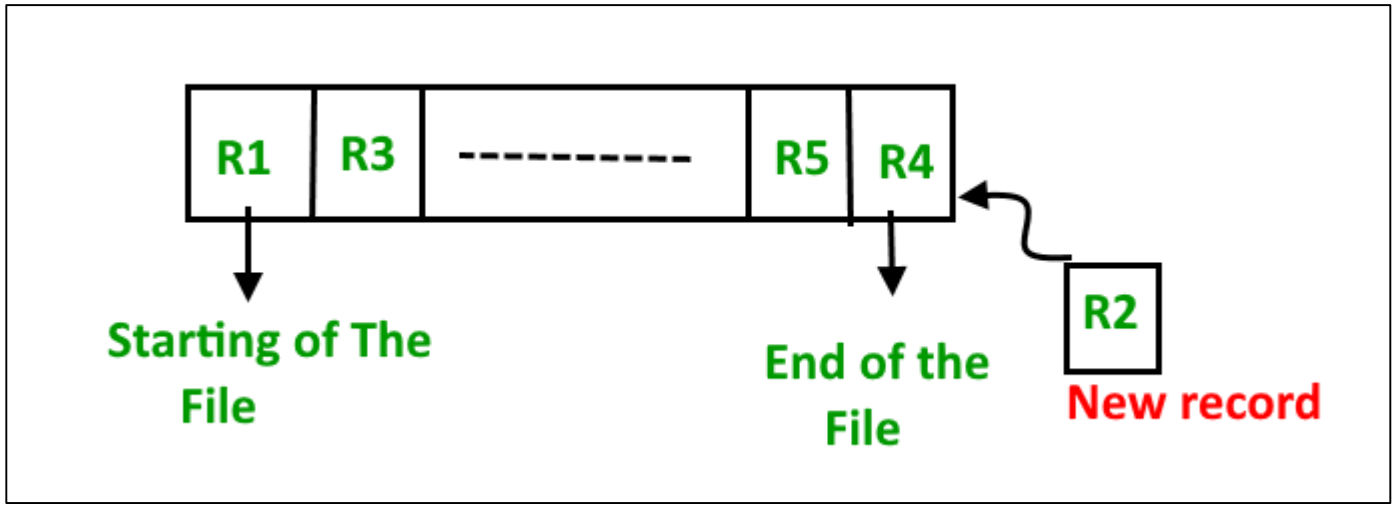
GLA UNIVERSITY MATHURA
Recognised by UGC Under Section 2(f)
Accredited with A Grade by NAAC
12-B Status from UGC

Database Management System (BCSC – 1003)

Topic: File Organization



Dr. Nikhil Govil

Assistant Professor, Dept. of CEA, GLA University, Mathura.

Topics to be covered

- File Organization
- Objectives of File Organization
- Indexing in DBMS
- Index Structure
- Indexing Methods

File Organization

- The File is a collection of records. Using the primary key, we can access the records.
- File organization is a logical relationship among various records. This method defines how file records are mapped onto disk blocks.
- File organization is used to describe the way in which the records are stored in terms of blocks, and the blocks are placed on the storage medium.
- Files of fixed length records are easier to implement than the files of variable length records.

Objective of File Organization

- It contains an optimal selection of records, i.e., records can be selected as fast as possible.
- To perform insert, delete or update transaction on the records should be quick and easy.
- The duplicate records cannot be induced as a result of insert, update or delete.
- For the minimal cost of storage, records should be stored efficiently.

Indexing in DBMS

- The index is a type of data structure.
- It is used to locate and access the data in a database table quickly.
- Indexing is used to optimize the performance of a database by minimizing the number of disk accesses required when a query is processed.

Index Structure

- Indexes can be created using some database columns.

Search key	Data Reference

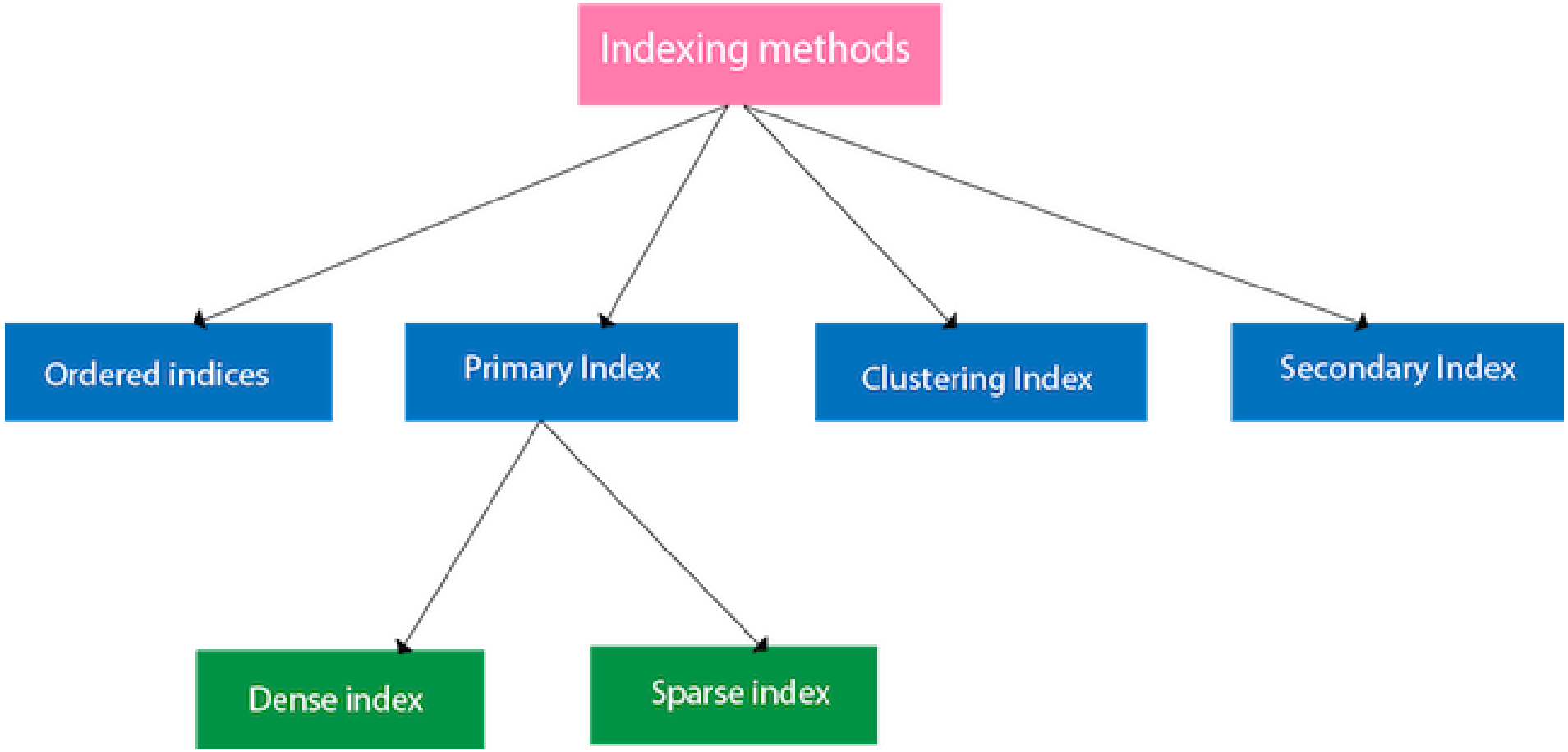
Fig: Structure of Index

- The first column of the database is the search key that contains a copy of the primary key or candidate key of the table. The values of the primary key are stored in sorted order so that the corresponding data can be accessed easily.

- The second column of the database is the data reference. It contains a set of pointers holding the address of the disk block where the value of the particular key can be found.

Indexing Methods

- Indexing in DBMS can be performed in following methods:



Order Indices

- The indices are usually sorted to make searching faster. The indices which are sorted are known as ordered indices.

Example:

Suppose we have a student table with thousands of records, each of which is 10 bytes long. Imagine their IDs start from 1 2, 3… and goes on. And we have to search student with ID 678.

- In a normal database with no index, it searches the disk block from the beginning till it reaches 678.
- Now if we want to search record with ID 678, then it will search using indexes.

Primary Index

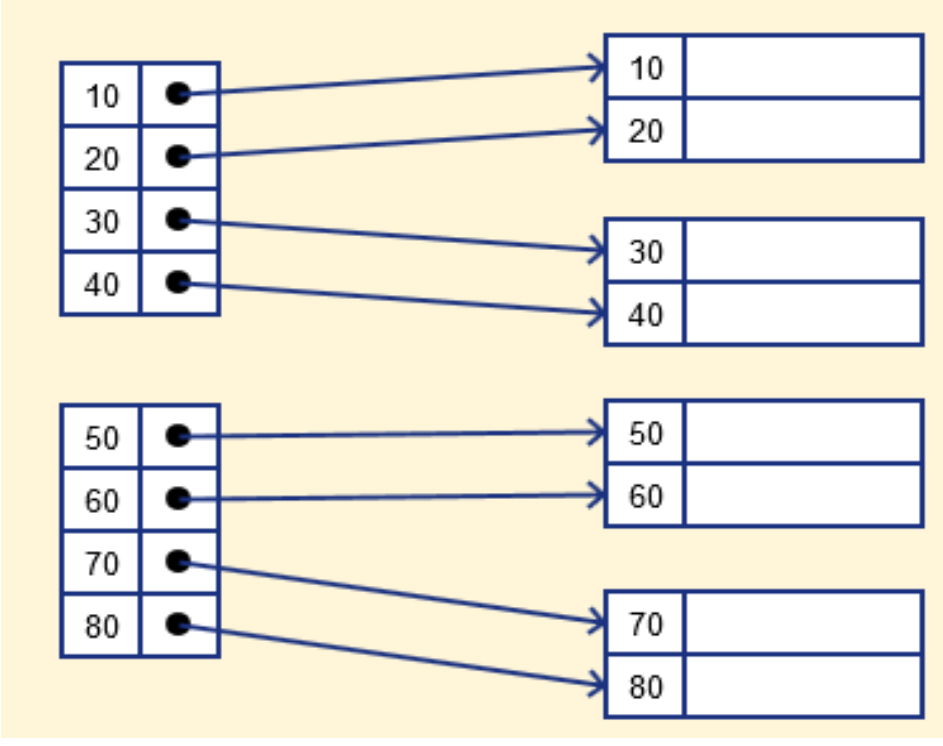
- If the index is created on the basis of the primary key of the table, then it is known as primary indexing. These primary keys are unique to each record and contain 1:1 relation between the records.

- As primary keys are stored in sorted order, the performance of the searching operation is quite efficient.

- The primary index can be classified into two types as:
  - ❖Dense index and
  - ❖Sparse index.

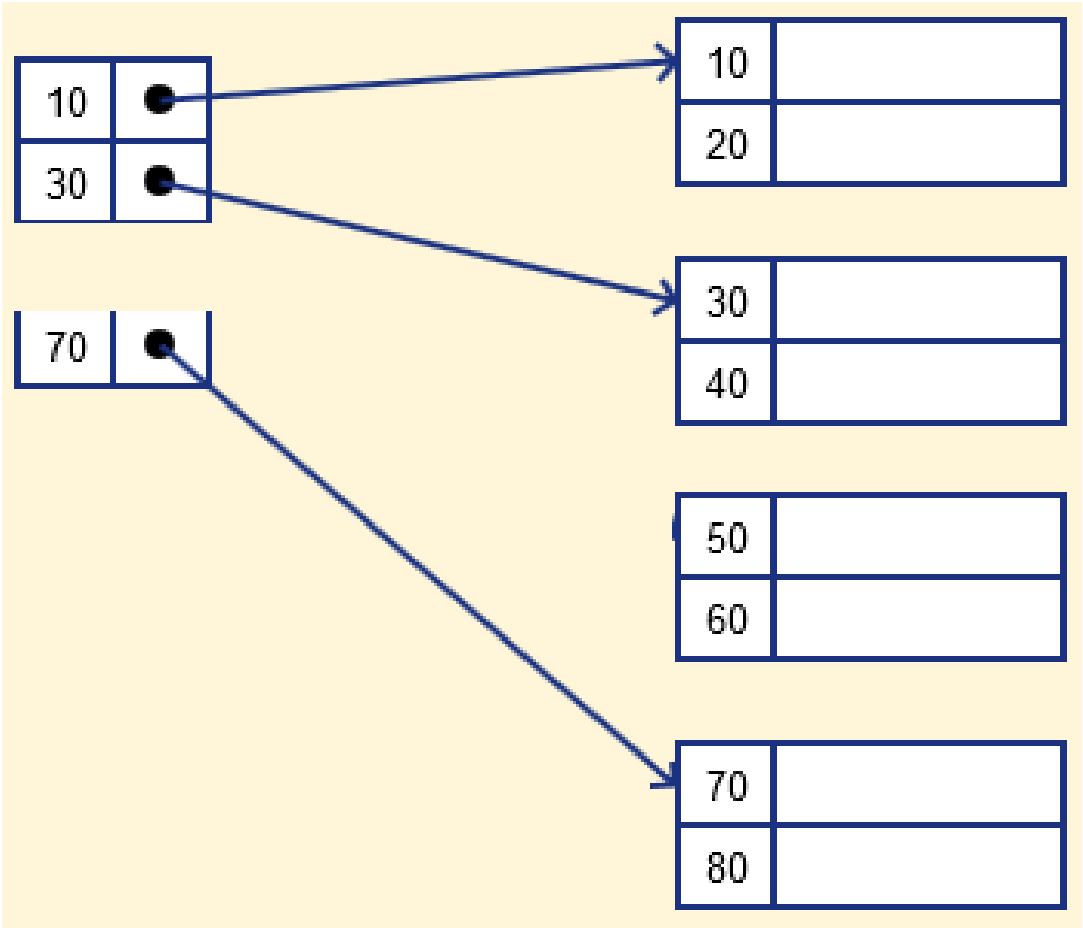
Dense Index

- In dense index, a record is created for every search key valued in the database.
- It helps us to search faster but needs more space to store index records.
- In this Indexing, method records contain search key value and points to the real record on the disk.



Sparse Index

- It is an index record that appears for only some of the values in the file.
- It needs less space, less maintenance overhead for insertion, and deletions.



Clustering Index

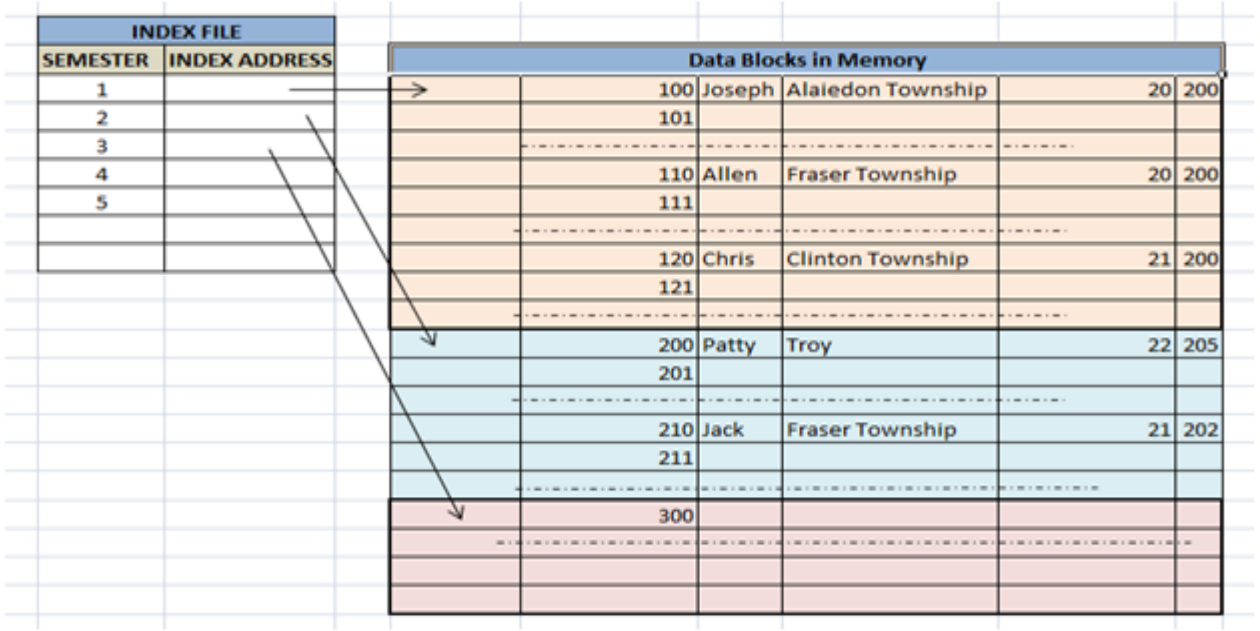
- In some cases, the index is created on non-primary key columns which may not be unique for each record.

- In such cases, in order to identify the records faster, we will group two or more columns together to get the unique values and create index out of them.

- In clustering index, records with similar characteristics are grouped together and indexes are created for these groups.

Clustering Index

- For example, students studying in each semester are grouped together, i.e.; 1st semester students, 2nd semester students, 3rd semester students etc. are grouped.



INDEX FILE	
SEMESTER	INDEX ADDRESS
1	
2	
3	
4	
5	

Data Blocks in Memory					
100	Joseph	Alaiedon Township	20	200
101				
110	Allen	Fraser Township	20	200
111				
120	Chris	Clinton Township	21	200
121				
200	Patty	Troy	22	205
201				
210	Jack	Fraser Township	21	202
211				
300				

- Here, indexes are created for each semester in the index file. In the data block, the students of each semester are grouped together to form the cluster. The address in the index file points to the beginning of each cluster. In the data blocks, requested student ID is then search in sequentially.

Secondary Index

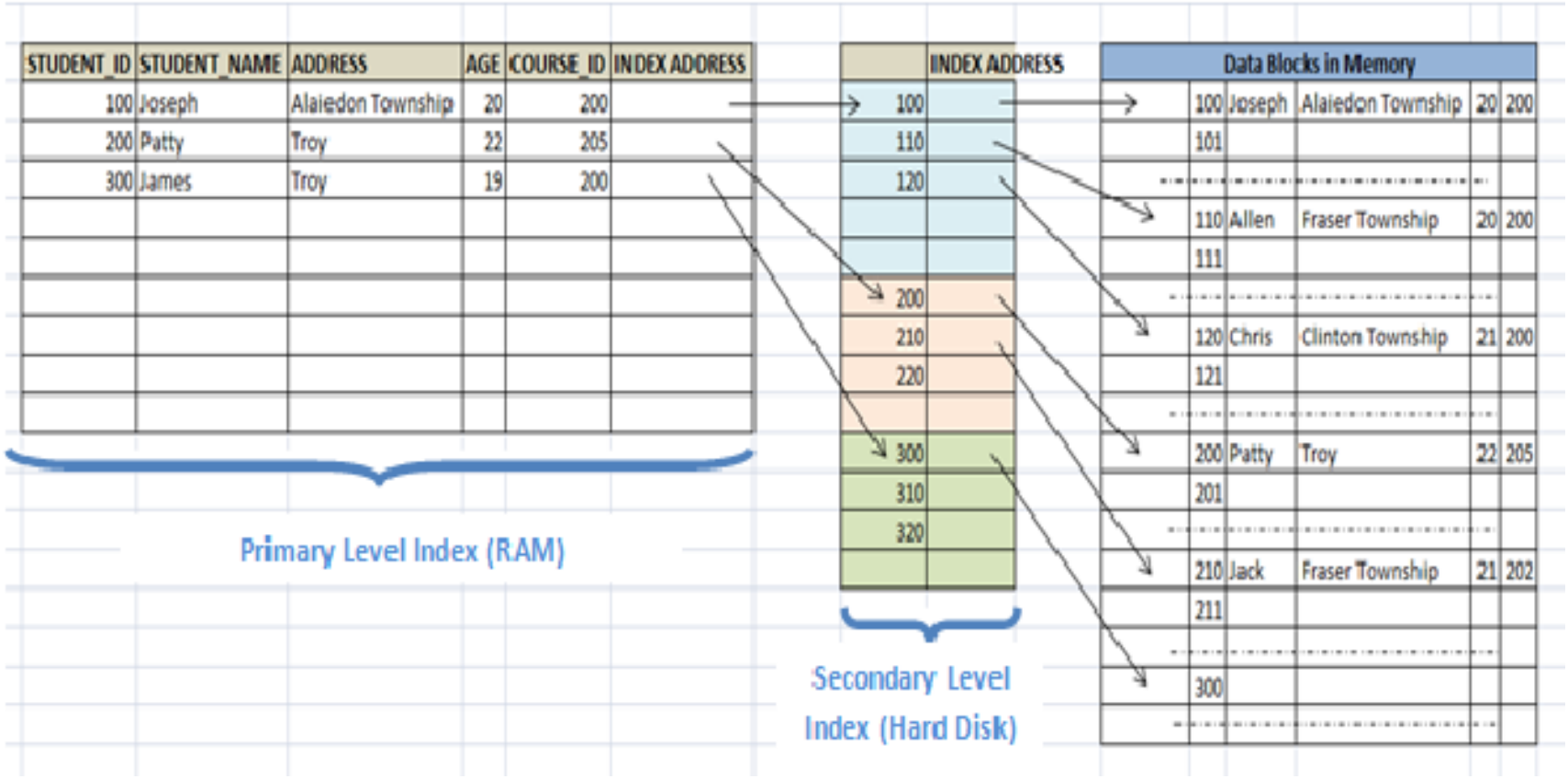
- In the sparse indexing, as the table size grows, the (index, address) mapping file size also grows.

- In the memory, usually these mappings are kept in the primary memory so that address fetch should be faster. And latter the actual data is searched from the secondary memory based on the address got from mapping.

- If the size of this mapping grows, fetching the address itself becomes slower. Hence sparse index will not be efficient.

- In order to overcome this problem next version of sparse indexing is introduced i.e.; Secondary Indexing.

Secondary Index



- In the above diagram, we can see that columns are divided into groups of 100s first. In the secondary memory, these groups are further divided into sub-groups. Actual data records are then stored in the physical memory.

References

- Korth, Silbertz and Sudarshan (1998), “Database Concepts”, 4th Edition, TMH.
- Elmasri and Navathe (2010), “Fundamentals of Database Systems”, 5th Edition, Addision Wesley.
- Date C J,” An Introduction to Database Systems”, 8th Edition, Addision Wesley.
- M. Tamer Oezsu, Patrick Valduriez (2011). “Principles of Distributed Database Systems”, 2nd Edition, Prentice Hall.
- https://www.javatpoint.com/dbms-file-organization/ last accessed on 25 September' 2022.


Thank
you