MET's Institute of Engineering

Bhujbal Knowledge City, Adgaon, Nashik.

Department of Computer Engineering

"FILE ORGANIZATION"

Prepared By

Prof. Anand N. Gharu

(Assistant Professor)

Computer Dept.

CLASS : SE COMPUTER 2019

22 April 2024

SUBJECT : DSA (SEM-II)

UNIT : VI

SYLLABUS

Savitribai Phule Pune University

Second Year of Engineering (2019 Course)

210252: Data Structures and Algorithms

Teaching Scheme	Credit Scheme	Examination Scheme and Marks
Lecture: 03 Hours/Week	03	**Mid_Semester(TH):** 30 Marks **End_Semester(TH):** 70 Marks

Prerequisite Courses:	110005: Programming and Problem Solving 210242: Fundamentals of Data Structures
Companion Course:	210257: Data Structures and Algorithms Laboratory

Unit VI	**File Organization**	**(07 Hours)**
Files: concept, need, primitive operations. **Sequential file organization-** concept and primitive operations, **Direct Access File-** Concepts and Primitive operations, **Indexed sequential file organization-**concept, types of indices, structure of index sequential file, **Linked Organization-** multi list files, coral rings, inverted files and cellular partitions.		
#Exemplar/Case Studies	**External Sort-** Consequential processing and merging two lists, multiway merging- a k way merge algorithm	
***Mapping of Course Outcomes for Unit VI**	CO4, CO6	

SYLLABUS

- **Files:** concept, need, primitive operations.
- **Sequential file organization-** concept and primitive operations
- **Direct Access File-** Concepts and Primitive operations,
- **Indexed sequential file organization**-concept,
- **Types of indices**, structure of index sequential file,
- **Linked Organization-** multi list files, coral rings, inverted files and cellular partitions.

FILES

INTRODUCTION

- **The File** is a collection of records. Using the primary key, we can access the records. The type and frequency of access can be determined by the type of file organization which was used for a given set of 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.
- The first approach to map the database to the file is to use the several files and store only one fixed length record in any given file. An alternative approach is to structure our files so that we can contain multiple lengths for records.
- **Files of fixed length records are easier to implement than the files of variable length records.**

INTRODUCTION

Objectives of File Organization in Data Structure

- 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.

INTRODUCTION

- **A file** is a collection of records which holds information of similar data. Almost all information stored in a computer system is a file.

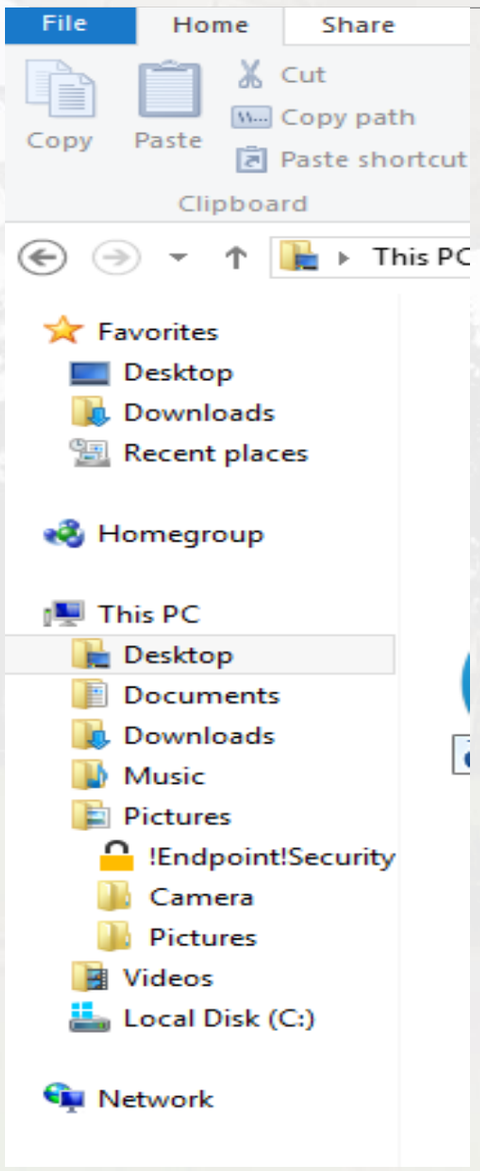
- **Need of file organization :**

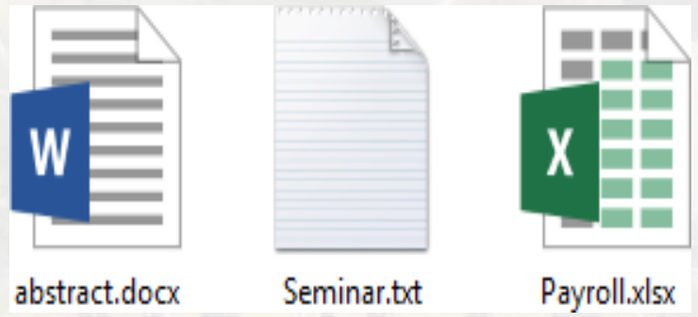
Way to organize & store data in a manner which determines standard method of data access, efficiency and flexibility.

INTRODUCTION

- **Different types o file available :**

e.g. text files, program files, directory files etc





Directory files

INTRODUCTION

Types of files supported by ‘C’

- A stream refers to the flow of data from one place (storage) to another (program) or vice-versa.
- C can handle files as Stream-oriented data and System oriented data.

- **There are two types of streams Text & Binary streams**
- **Text Stream**

1. It consists of sequence of characters and each line is terminated by a newline character.
2. The I/O operations like buffering, data conversions, etc. take place automatically.

INTRODUCTION

Types of files supported by ‘C’

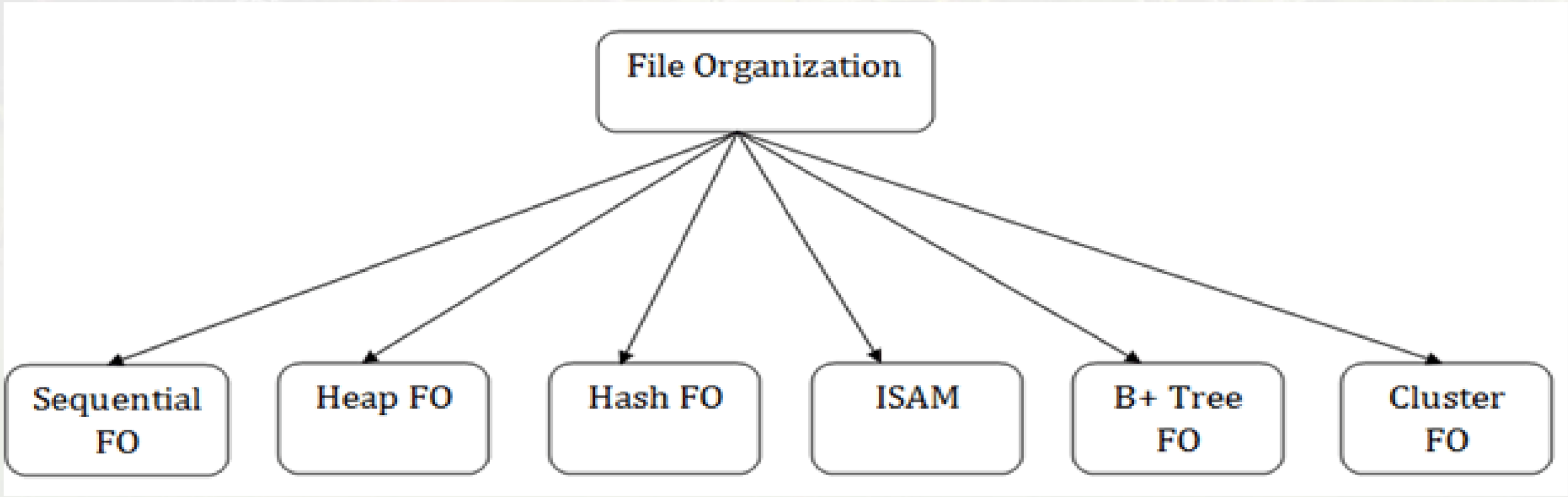
There are two types of streams Text & Binary streams

- **Binary Stream**

1. It is a series of bytes.
2. Binary streams are primarily used for non-textual data, which is required to keep exact contents of the file.
3. These are System-oriented data files that are more closely associated with the OS and data stored in memory without converting into text format.

Types of File Organization

- **Types of file organization:**
- File organization contains various methods. These particular methods have pros and cons on the basis of access or selection. In the file organization, the programmer decides the best-suited file organization method according to his requirement.
- **Types of file organization are as follows:**



Types of File Organization

- **Sequential file organization**
- **Indexed File**
- **Direct Access File**
- **Heap file organization**
- **Hash file organization**
- **B+ file organization**
- **Indexed sequential access method (ISAM)**
- **Cluster file organization**

Types of File Organization

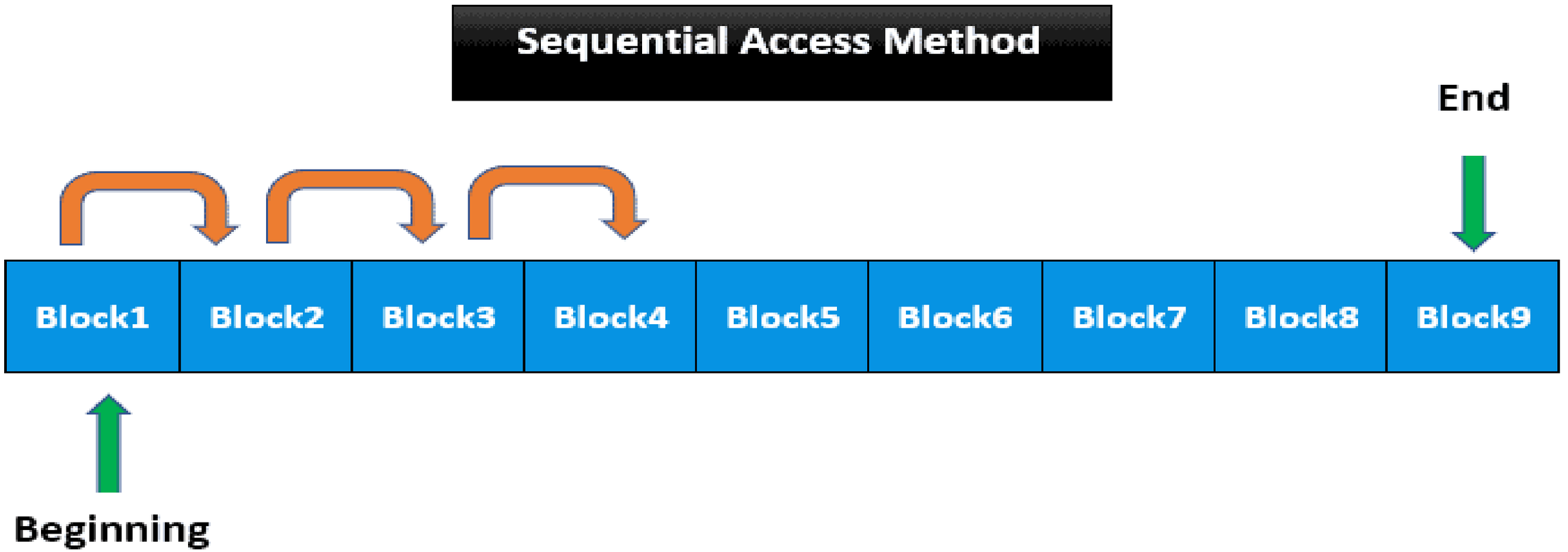
- **Sequential file organization :**

Sequential file access is the most straightforward method of accessing files. This method accesses data as one record at a time by starting from the beginning of the file to its end. Moreover, the records are read or written in the order they appear in the file.

Sequential file access is best suited for applications that linearly process data, such as reading or writing data to a log file or processing data in batch operations. For example, when analyzing a large dataset, it may be beneficial to write the data to a sequential file and then read it in one record at a time to perform the desired analysis.

Types of File Organization

- **Sequential file organization :**



The main advantage of sequential file access is its simplicity, which makes it easy to implement and use. In contrast, its main disadvantage is that it can be slow and inefficient for random access operations or when working with large files.

Types of File Organization

Sequential file organization :

- **Advantages of sequential file**

1. • It is simple to program and easy to design.
2. • Sequential file is best use if storage space.

- **Disadvantages of sequential file**

1. • Sequential file is time consuming process.
2. • It has high data redundancy.
3. • Random searching is not possible.

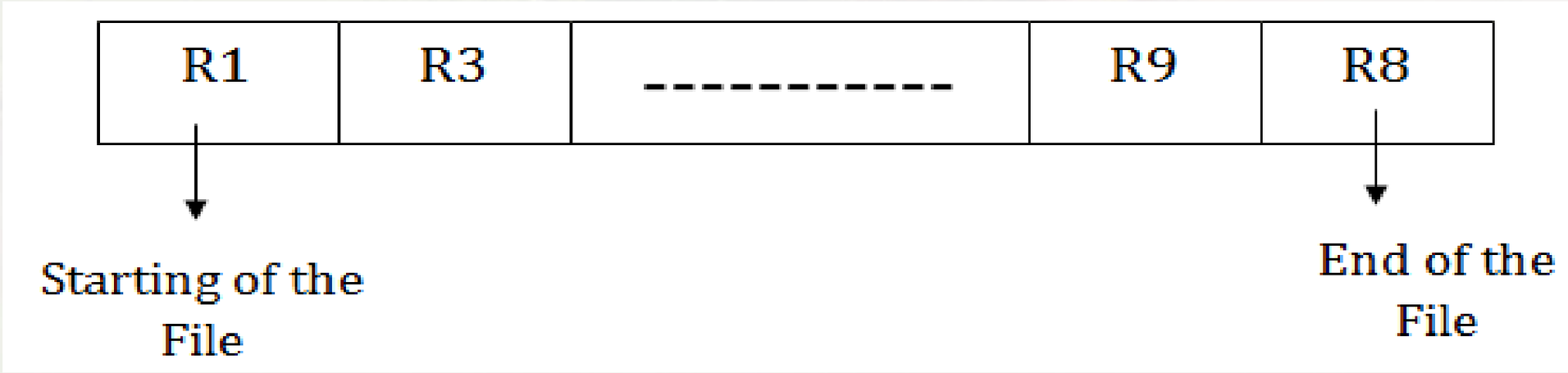
Types of File Organization

- **Sequential file organization :**

1. Pile File Method

It is a quite simple method. In this method, we store the record in a sequence, i.e., one after another. Here, the record will be inserted in the order in which they are inserted into tables.

In case of updating or deleting of any record, the record will be searched in the memory blocks. When it is found, then it will be marked for deleting, and the new record is inserted.



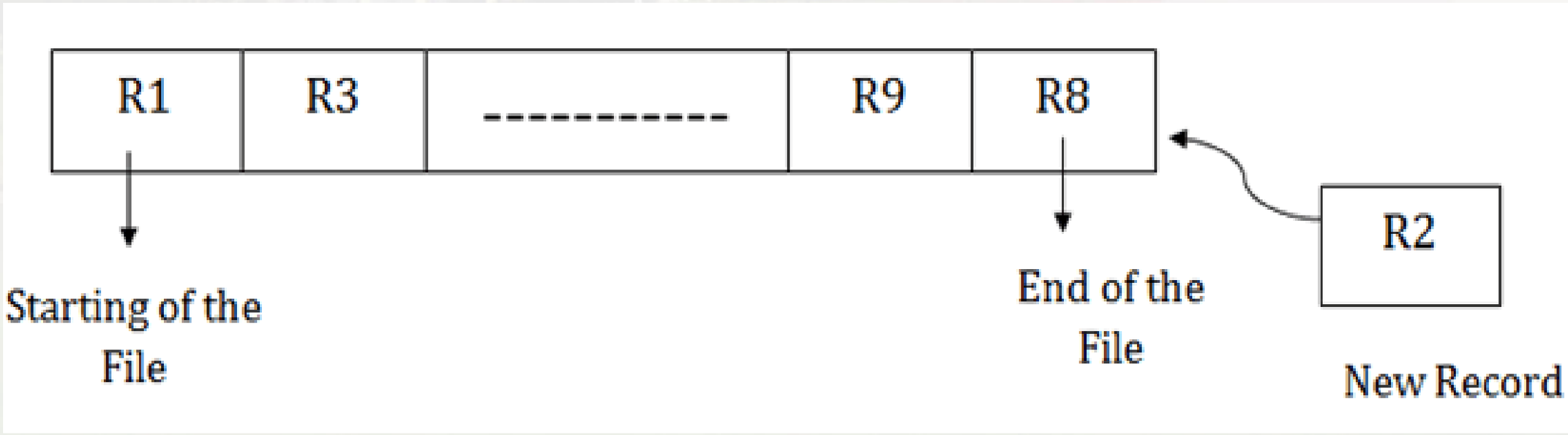
Types of File Organization

- **Sequential file organization :**

1. Pile File Method

Insertion of the new record:

Suppose we have four records R1, R3 and so on upto R9 and R8 in a sequence. Hence, records are nothing but a row in the table. Suppose we want to insert a new record R2 in the sequence, then it will be placed at the end of the file. Here, records are nothing but a row in any table.



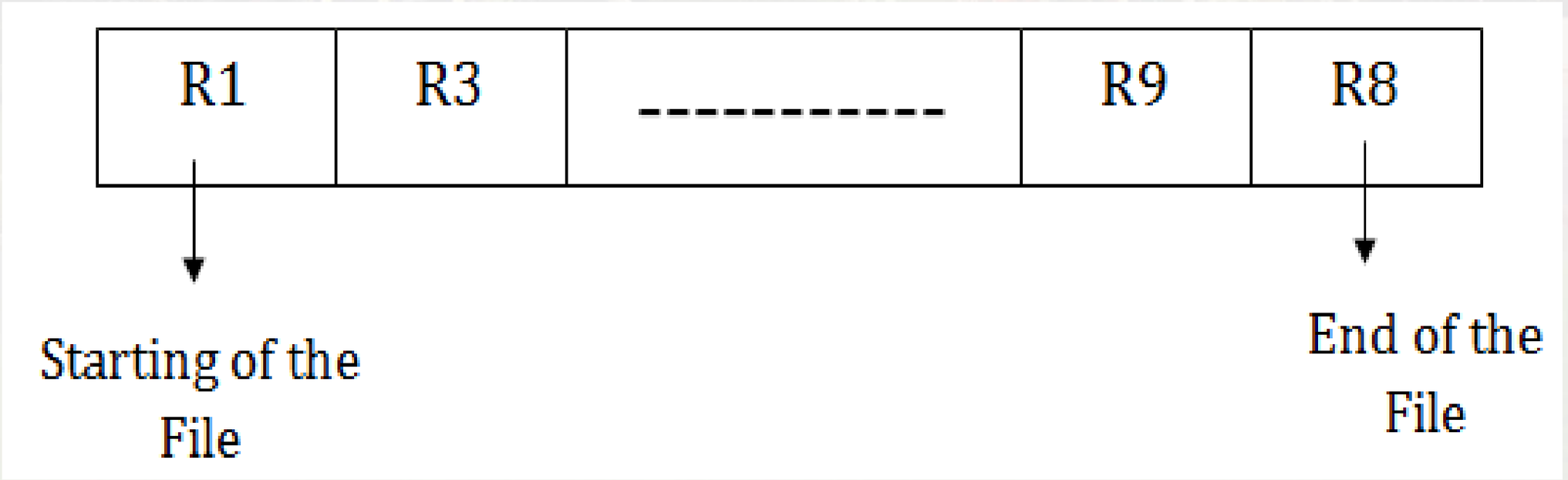
Types of File Organization

- **Sequential file organization :**

2. Sorted File Method:

In this method, the new record is always inserted at the file's end, and then it will sort the sequence in ascending or descending order. Sorting of records is based on any primary key or any other key.

In the case of modification of any record, it will update the record and then sort the file, and lastly, the updated record is placed in the right place.



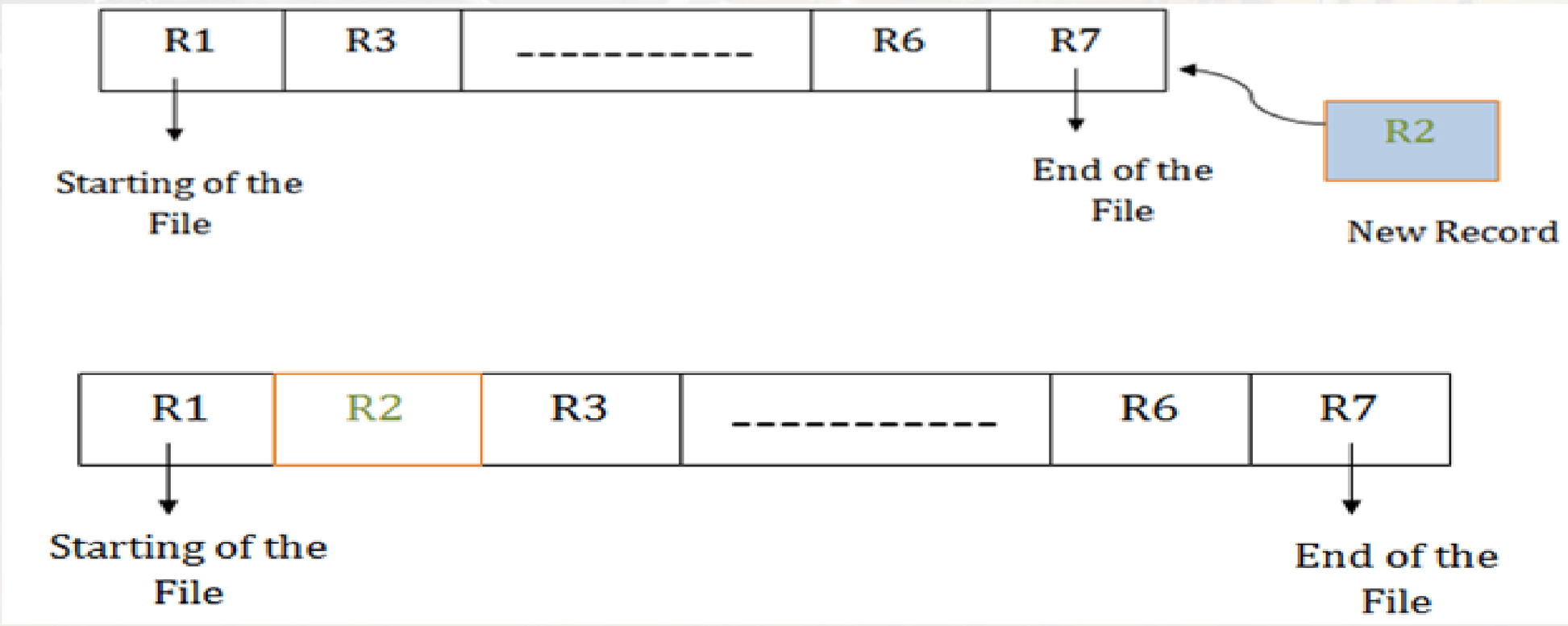
Types of File Organization

- **Sequential file organization :**

2. Sorted File Method:

Insertion of the new record:

Suppose there is a preexisting sorted sequence of four records R1, R3 and so on upto R6 and R7. Suppose a new record R2 has to be inserted in the sequence, then it will be inserted at the end of the file, and then it will sort the sequence..



Types of File Organization

- **Sequential file organization :**
- **Advantages of Sequential File Organization**

1. Fast and efficient method for huge amounts of data.
2. In this method, files can be easily stored in cheaper storage mechanism like magnetic tapes.
3. It is simple in design. It requires no much effort to store the data.
4. Files can be easily stored in magnetic tapes i.e. cheaper storage mechanism.

- **Disadvantages of Sequential File Organization**

1. Time wastage as we cannot jump on a particular record that is required, but we have to move in a sequential manner which takes our time.
2. The sorted file method is inefficient as it takes time and space for sorting records.

Types of File Organization

- **Indexed File Organization :**
- Indexed sequential access file combines both sequential file and direct access file organization.
- In indexed sequential access file, records are stored randomly on a direct access device such as magnetic disk

by a primary key.

- This file have multiple keys. These keys can be alphanumeric in which the records are ordered is called

primary key.

- The data can be access either sequentially or randomly using the index. The index is stored in a file and read

into memory when the file is opened

Types of File Organization

- **Indexed File Organization :**

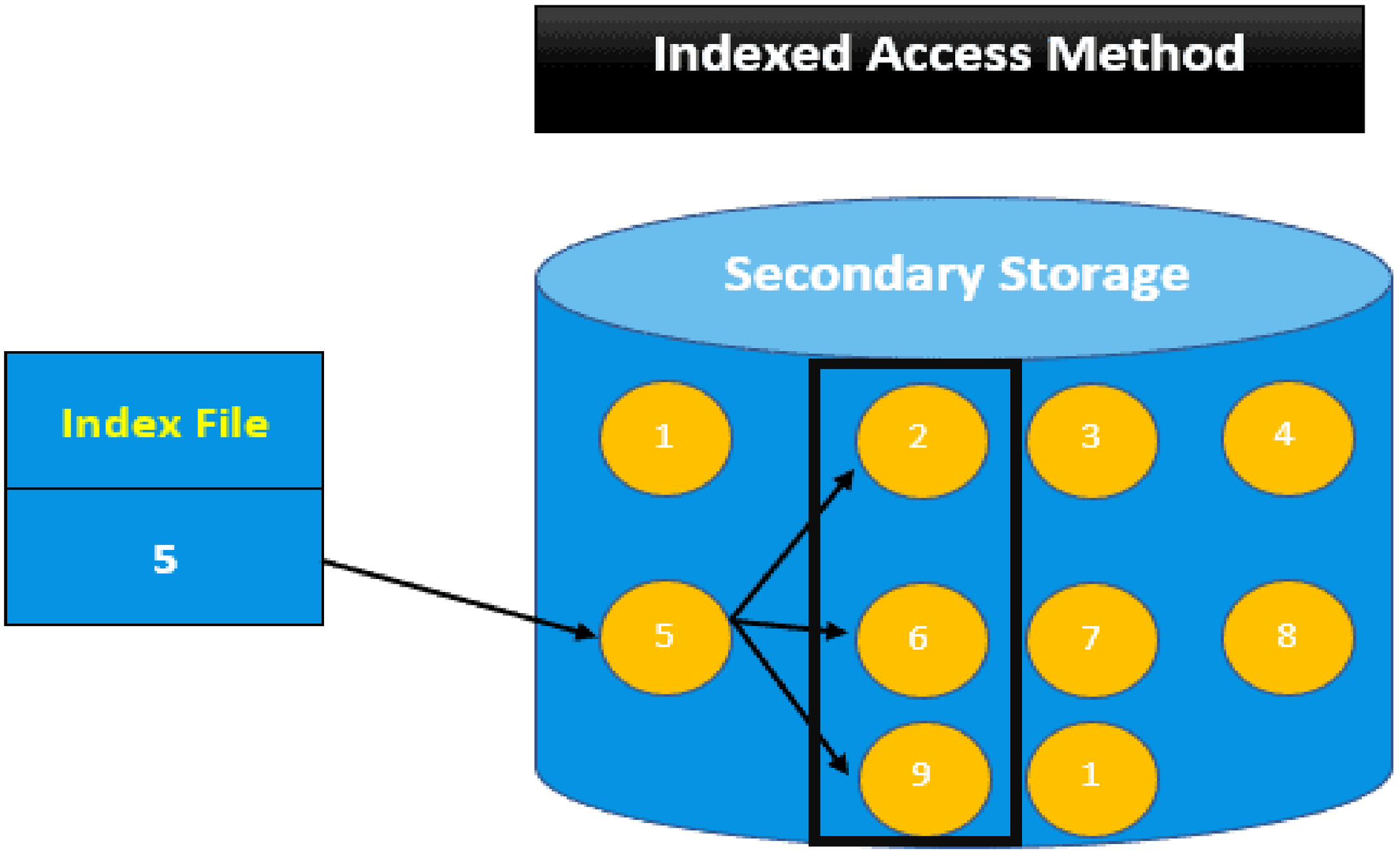
Indexed file access is a method that incorporates the benefits of both sequential and direct file access. This method involves creating an index file that maps logical keys or data elements to their corresponding physical addresses within the file. Moreover, the system stores the index separately from the data file, enabling quick access to locate the desired data.

Indexed file access is best suited for applications that require fast access to particular data elements within a large file. For example, in a file system, we may need to access specific files based on their name or location. The index created for the file system allows quick access to the file's physical location, enabling efficient access.

The **main advantage of indexed file** access is its speed and efficiency for random and sequential access operations. But**, its main disadvantage** is that it requires additional storage space for the index, which can increase the cost and complexity of the system.

Types of File Organization

- **Indexed File Organization :**

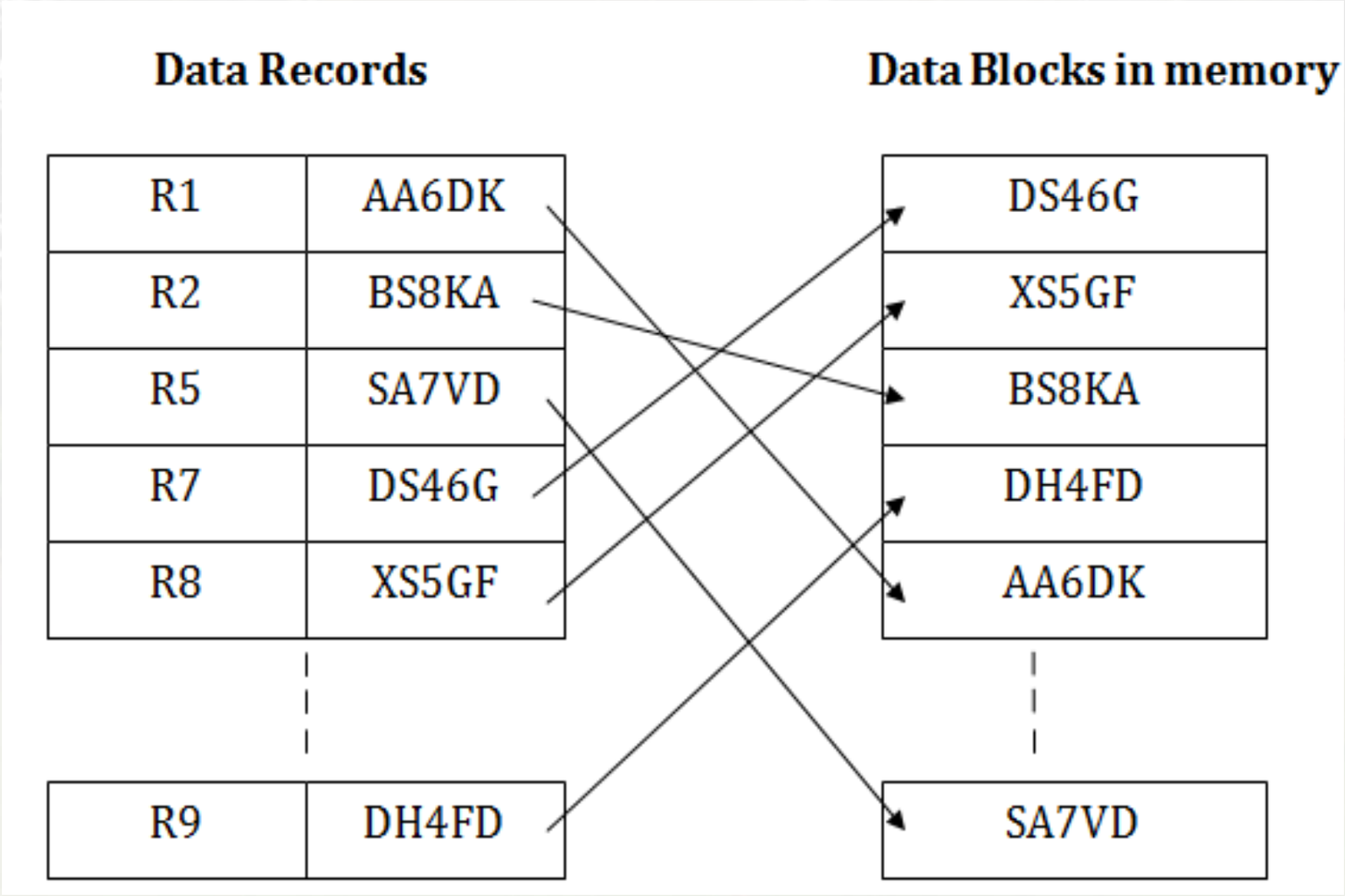


Types of File Organization

- **Indexed Sequential Access Method (ISAM) file organization :**

ISAM method is an advanced sequential file organization. In this method, records are stored in the file using the primary key. An index value is generated for each primary key and mapped with the record. This index contains the address of the record in the file.

If any record has to be retrieved based on its index value, then the address of the data block is fetched and the record is retrieved from the memory.



Types of File Organization

- **Indexed Sequential Access Method (ISAM) file organization :**
- **Advantages :**

• In indexed sequential access file, sequential file and random file access is possible.

• It accesses the records very fast if the index table is properly organized.

• The records can be inserted in the middle of the file.

• It provides quick access for sequential and direct processing.

• It reduces the degree of the sequential search.

Types of File Organization

- **Indexed Sequential Access Method (ISAM) file organization :**
- **Disadvantages :**
- Indexed sequential access file requires unique keys and periodic reorganization.
- Indexed sequential access file takes longer time to search the index for the data access or retrieval.
- It requires more storage space.
- It is expensive because it requires special software.
- It is less efficient in the use of storage space as compared to other file organizations.

Indexing in File Organization

What is Indexing?

Indexing is a data structure technique which allows you to quickly retrieve records from a database file. An Index is a small table having only two columns. The first column comprises a copy of the primary or candidate key of a table. Its second column contains a set of pointers for holding the address of the disk block where that specific key value stored.

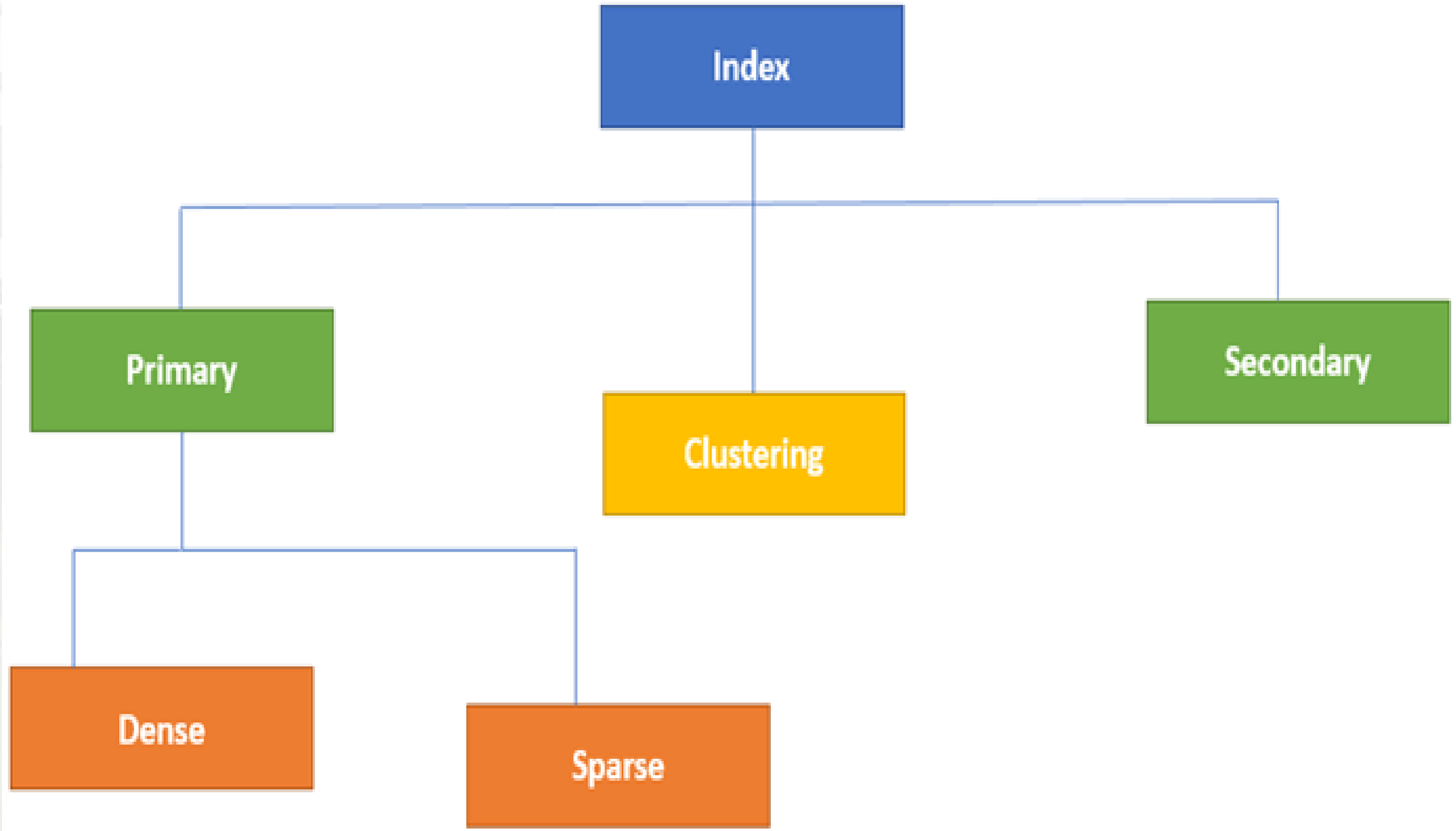
An index –

Takes a search key as input

Efficiently returns a collection of matching records

Indexing in File Organization

TYPES OF INDEXING



Indexing in File Organization

Indexing in Database is defined based on its indexing attributes. Two main types of indexing methods are:

1. Primary Indexing
2. Secondary Indexing

- **Primary Index in DBMS**

Primary Index is an ordered file which is fixed length size with two fields. The first field is the same a primary key and second, field is pointed to that specific data block. In the primary Index, there is always one to one relationship between the entries in the index table.

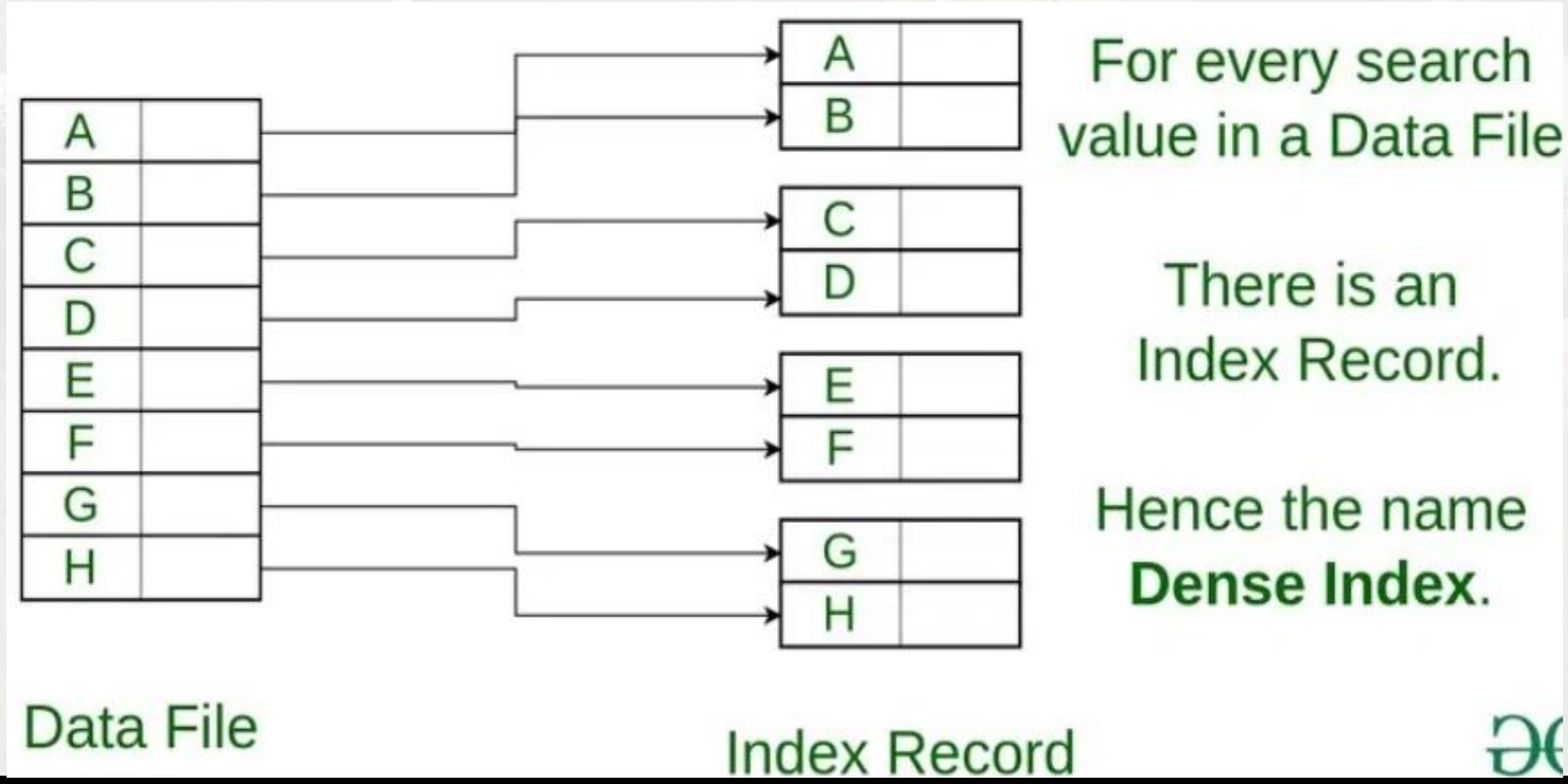
The primary Indexing in DBMS is also further divided into two types.

1. Dense Index
2. Sparse Index

Indexing in File Organization

Dense Index

In a dense index, a record is created for every search key valued in the database. This helps you 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.



Indexing in File Organization

Sparse Index :

It is an index record that appears for only some of the values in the file. Sparse Index helps you to resolve the issues of dense Indexing in DBMS.

In this method of indexing technique, a range of index columns stores the same data block address, and when data needs to be retrieved, the block address will be fetched.

However, sparse Index stores index records for only some search-key values.

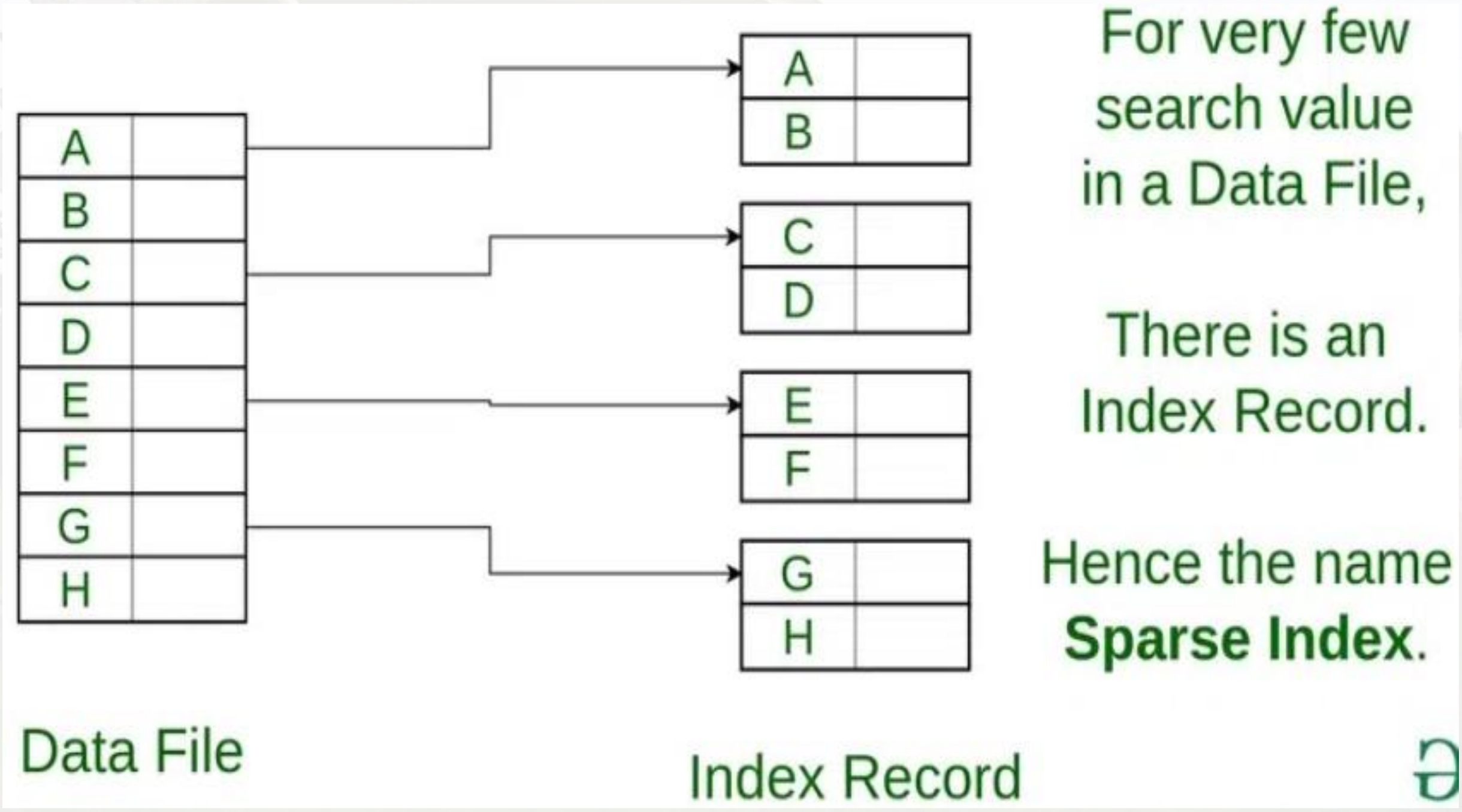
It needs less space, less maintenance overhead for insertion, and deletions but

It is slower compared to the dense Index for locating records.

Indexing in File Organization

Sparse Index :

Below is an database index Example of Sparse Index



Indexing in File Organization

Clustered Indexing: When more than two records are stored in the same file this type of storing is known as cluster indexing. By using cluster indexing we can reduce the cost of searching reason being multiple records related to the same thing are stored in one place and it also gives the frequent joining of more than two tables (records).

The clustering index is defined on an ordered data file. The data file is ordered on a non-key field. 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 an index out of them. This method is known as the clustering index. Essentially, records with similar properties are grouped together, and indexes for these groupings are formed.

Students studying each semester, for example, are grouped together. First-semester students, second-semester students, third-semester students, and so on are categorized

Indexing in File Organization

Cluster indexing :

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				

Indexing in File Organization

Non Clustered or Secondary Indexing :

The secondary Index in DBMS can be generated by a field which has a unique value for each record, and it should be a candidate key. It is also known as a non-clustering index.

This two-level database indexing technique is used to reduce the mapping size of the first level. For the first level, a large range of numbers is selected because of this; the mapping size always remains small.

Indexing in File Organization

Secondary Index Example

Let’s understand secondary indexing with a database index

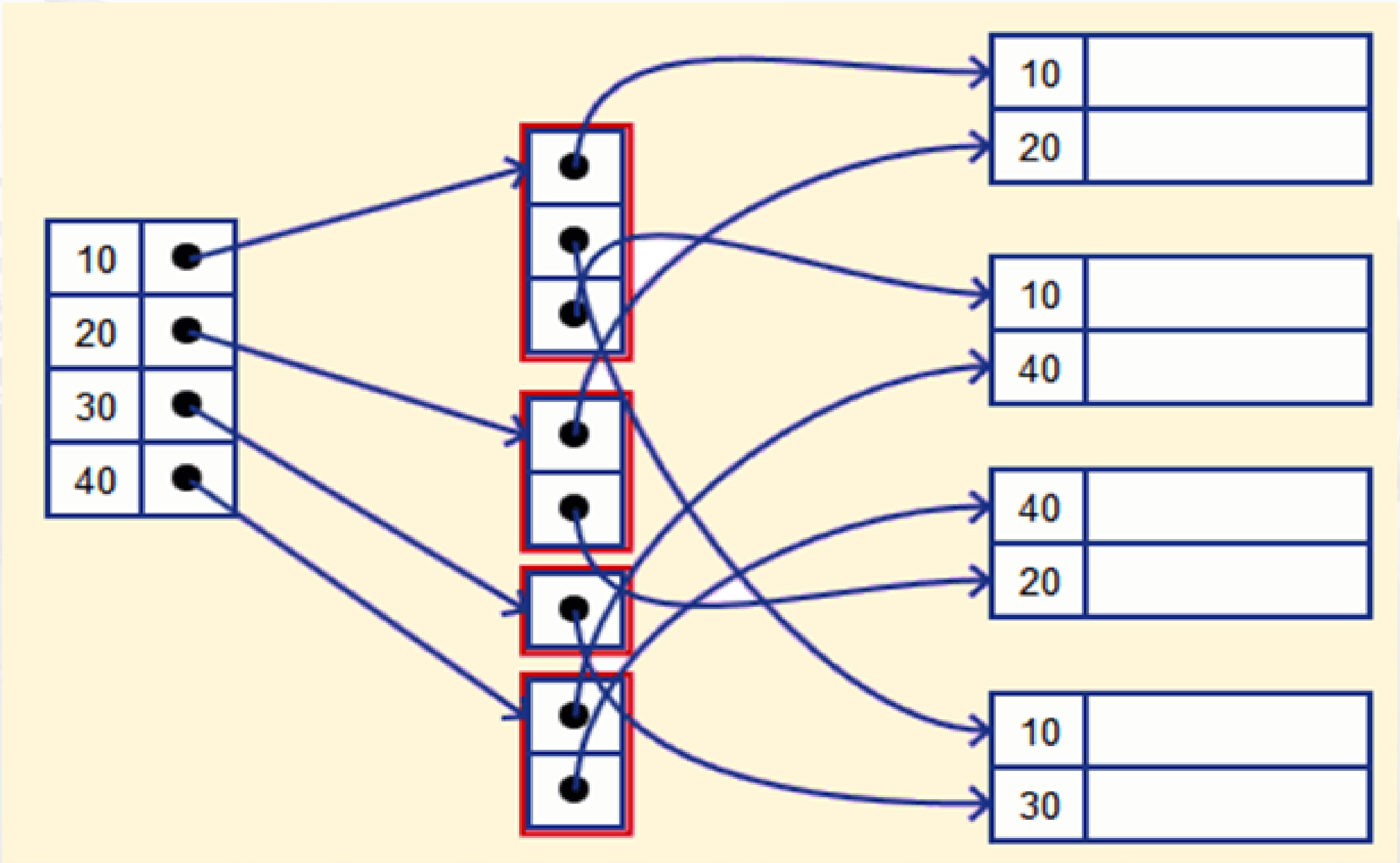
example:

In a bank account database, data is stored sequentially by acc_no; you may want to find all accounts in of a specific branch of ABC bank.

Here, you can have a secondary index in DBMS for every search-key. Index record is a record point to a bucket that contains pointers to all the records with their specific search-key value.

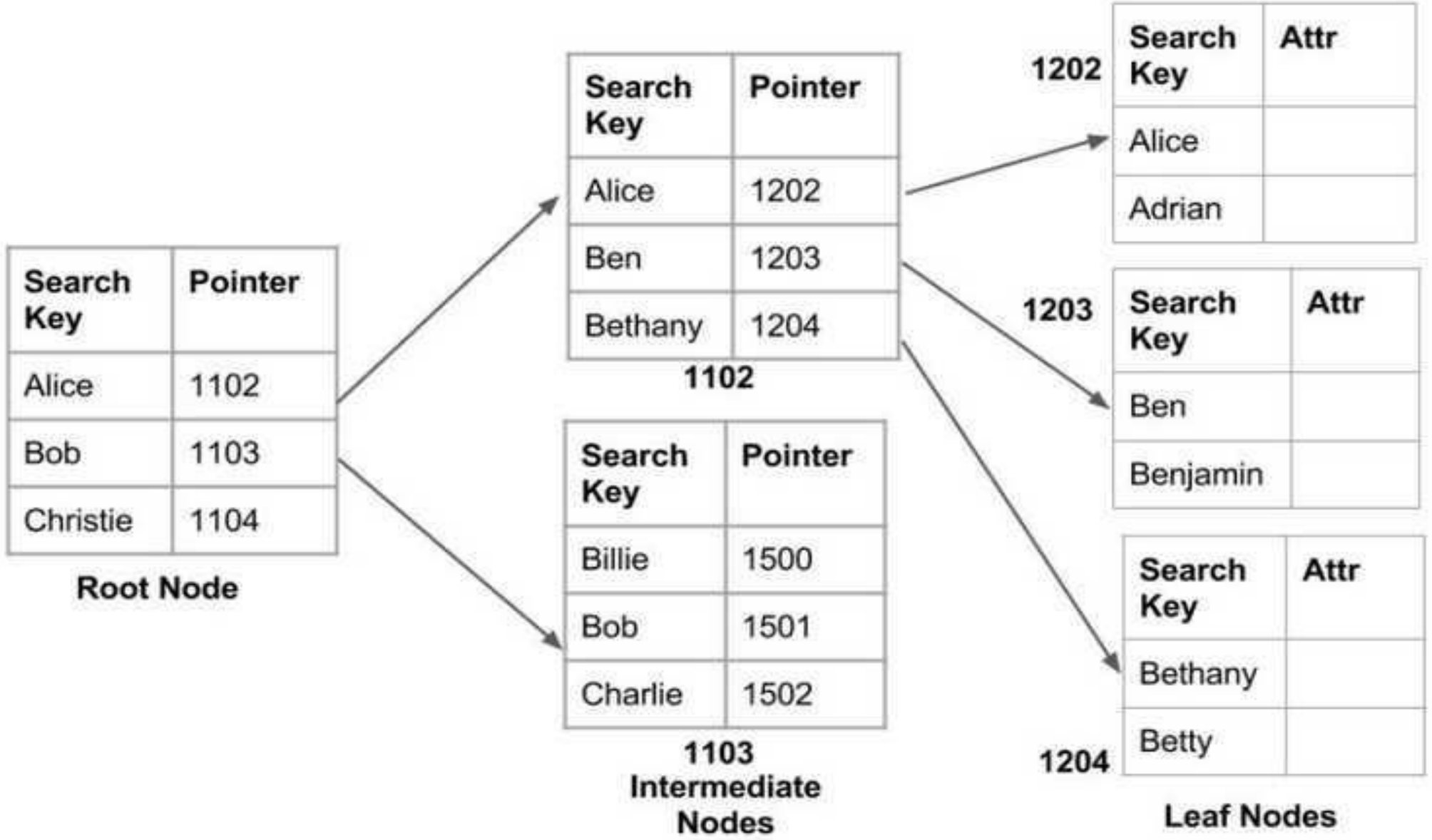
Indexing in File Organization

Secondary Index Example



Indexing in File Organization

Secondary Index Example



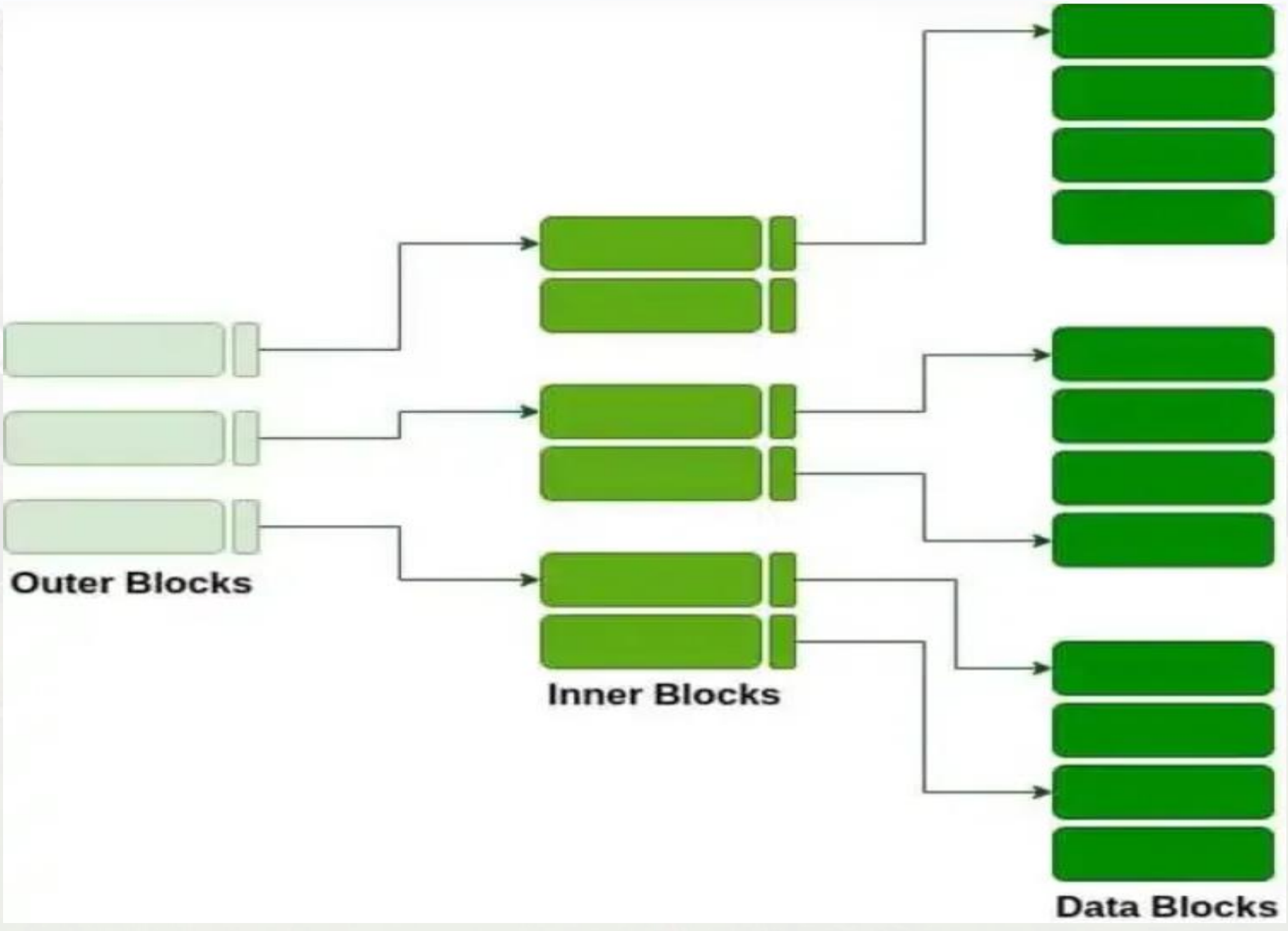
Non clustered index

Indexing in File Organization

Multilevel Indexing: With the growth of the size of the database, indices also grow. As the index is stored in the main memory, a single-level index might become too large a size to store with multiple disk accesses. The multilevel indexing segregates the main block into various smaller blocks so that the same can be stored in a single block. The outer blocks are divided into inner blocks which in turn are pointed to the data blocks. This can be easily stored in the main memory with fewer overheads.

Indexing in File Organization

Multilevel Indexing:



Indexing in File Organization

Advantages of Indexing

Important pros/ advantage of Indexing are:

1. It helps you to reduce the total number of I/O operations needed to retrieve that data, so you don't need to access a row in the database from an index structure.
2. **Offers Faster search and retrieval of data to users.**
3. Indexing also helps you to **reduce tablespace** as you don't need to link to a row in a table, as there is no need to store the ROWID in the Index. Thus you will able to reduce the tablespace.
4. You can't sort data in the lead nodes as the value of the primary key classifies it.

Indexing in File Organization

Disadvantages of Indexing :

Important drawbacks/cons of Indexing are:

1. To perform the indexing database management system, you need **a primary key** on the table with a unique value.
2. You can't perform any other indexes in Database on the Indexed data.
3. You are not allowed to partition an index-organized table.
4. SQL Indexing Decrease performance in INSERT, DELETE, and UPDATE query.

Types of File Organization

- **Direct Access File organization :**

Direct access file is also known as random access or relative file organization.

• In direct access file, all records are stored in direct access storage device (DASD), such as hard disk. The

records are randomly placed throughout the file.

• The records does not need to be in sequence because they are updated directly and rewritten back in the

same location.

• This file organization is useful for immediate access to large amount of information. It is used in accessing

large databases.

• It is also called as hashing.

Types of File Organization

- **Direct Access File organization :**
- **Direct file access, also known as random access.**
- It allows us to access data directly from any location within the file, without the need to read or write all the records that come before it. Furthermore, this method accesses records within the file by using their physical addresses or positions.
- Direct file access is best suited for applications that require quick and efficient access to specific records or data elements within a file.
- For example, in a database application, we may need to quickly retrieve customer data based on a specific customer ID. Direct file access can quickly access the record containing the customer data without having to read through all the records that come before it.

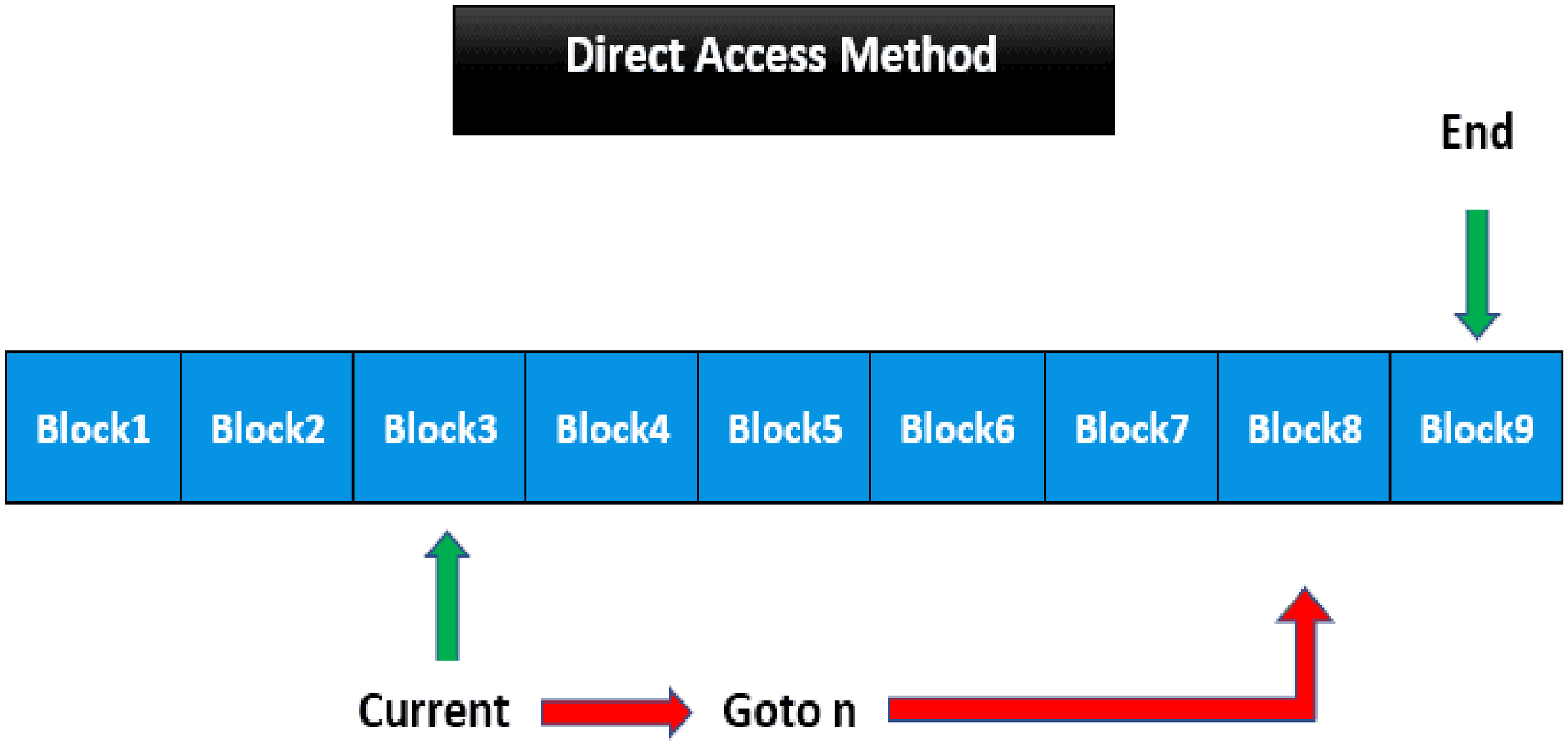
Types of File Organization

- **Direct Access File organization :**

The **main advantage** of direct file access is its speed and efficiency for random access operations. On the other hand, its **main disadvantage** is that it can be more complex and difficult to implement and use than sequential file access.

Types of File Organization

- **Direct Access File organization :**



Types of File Organization

Direct Access File Organization :

Advantages of direct access file organization

1. • Direct access file helps in online transaction processing system (OLTP) like online railway reservation system.
2. • In direct access file, sorting of the records are not required.
3. • It accesses the desired records immediately.
4. • It updates several files quickly.
5. • It has better control over record allocation.

Disadvantages of direct access file organization

1. • Direct access file does not provide back up facility.
2. • It is expensive.
3. • It has less storage space as compared to sequential file.

Comparison of file method

Method	Sequential File Access	Direct File Access	Indexed File Access
Access method	Accesses data one record at a time in sequential order from beginning to end	Accesses data directly from any location within the file, using physical addresses or positions	Uses an index that maps logical keys or data elements to their corresponding physical addresses within the file
Suited for	Applications that process data in a linear fashion, such as reading or writing data to a log file or processing data in batch operations	Applications that require quick and efficient access to specific records or data elements within a file	Applications that require fast and efficient access to specific data elements within a large file
Advantages	Simplicity and ease of implementation and use	Speed and efficiency for random access operations	Speed and efficiency for both random and sequential access operations
Disadvantages	Slow and inefficient for random access operations or when working with large files	More complex and difficult to implement and use than sequential file access	Requires additional storage space for the index, which can increase the cost and complexity of the system

File Openin/Closing Modes

- **File opening/Closing modes in C++:**

Parameter	Meaning
ios::app	Append to end-of-file
ios::ate	Go to end-of-file on opening
ios::binary	Binary file
ios::in	Open file for reading only
ios::nocreate	Open fails if file the file does not exist
ios::noreplace	Open fails if the file already exists
ios::out	Open file for writing only
ios::trunk	Delete the contents of the file if it exists

File Operation

- **File modes in C++:**

In C++, files can be opened in various modes depending on the operations we want to perform on them. Here are the commonly used file modes in C++:

std::ios::in:

This mode is considered when we want to open a file for reading. When this mode opens a file, we can only read from it but not write to it.

```
std::ifstream infile;

infile.open("myfile.txt", std::ios::in);
```

File Operation

- **File modes in C++:**

std::ios::out:

This mode is considered when we want to open a file for writing. When this mode opens a file, we can only write to it but not read from it. If the file doesn't exist, it will be created.

```
std::ofstream outfile;
outfile.open("myfile.txt", std::ios::out);
```

std::ios::app:

This mode is considered when we want to open a file and append data to the end of it. If the file doesn't exist, it will be created.

```
std::ofstream outfile;
outfile.open("myfile.txt", std::ios::app);
```

File Operation

- **File modes in C++:**

std::ios::ate:

This mode is considered when we want to open a file and seek to the end of it immediately. This is useful when we want to append data to the end of a file or read the entire file at once.

```
std::ifstream infile;

infile.open("myfile.txt", std::ios::ate);
```

std::ios::binary:

This mode is considered when we want to open a file in binary mode. In binary mode, no newline translation is performed, and the file is treated as a sequence of bytes. This mode is typically used when working with non-text files such as images or executables.

```
std::ifstream infile;

infile.open("myfile.bin", std::ios::binary);
```

File Operation

- **Opening a file in C++:**

To open a file in C++, we can use the ofstream and ifstream classes. The ofstream class is used to write to a file, while the ifstream class is used to read from a file. Both classes are derived from the fstream class, which can be used for reading and writing files.

Below is an example of how to open a file for writing using the ofstream class:

```
#include <fstream>
using namespace std;

int main() {
  ofstream outfile;
  outfile.open("example.txt", ios::out);
  // Write to the file here
  outfile.close();
  return 0;
}
```

LINKED ORGANIZATION

- **In a linked organization,** the elements or components of the system are connected through relationships, but the order in which they occur may not be fixed. In Linked Organization, elements might or might not be stored in consecutive memory locations and the order is determined by the links between elements. This makes it easy to insert and delete elements without requiring any movement of other elements and it can be extended or reduced according to requirements.

LINKED ORGANIZATION

- **In a linked organization,** the elements or components of the system are connected through relationships, but the order in which they occur may not be fixed. In Linked Organization, elements might or might not be stored in consecutive memory locations and the order is determined by the links between elements. This makes it easy to insert and delete elements without requiring any movement of other elements and it can be extended or reduced according to requirements.

LINKED ORGANIZATION

Multikey File Organization :

When a file records are made accessed based on more than one key are called as Multikey file organization. This file organization is needed many times. E.g. In banking system we keep records of accounts in file. Now account holder needs account information which can be access through account no, while loan officer needs account records with a given value of overdue limit. So we need to provide to access path to the record based on different need. Generally these files are index sequential file in which file is stored sequentially based on primary key and more than one index table are provided based on different keys. Basically there are 2 approaches for implementing multi key organization.

LINKED ORGANIZATION

Inverted File Organization:

In this file organization a key's inversion index contain all of the values that the key presently has in the records of the data file. Each key-value entry in the inversion index points to all of the data records that have the corresponding value. The data file is said to be inverted on that key. Inverted files are sorted on inversion index so that binary search can be applied to find out index of record. Whenever record is added in data file its corresponding entry has to be made in inverted file.

LINKED ORGANIZATION

Multi List File Organization:

In multi list file organization the index contain all values that the secondary key has in data file same as inverted file but the difference is that the entry in the multi index for a secondary key value is pointer to the first data record with that key value. That data record contains pointer to second record having same key. Thus there is a linked list of data records for each value of secondary key. Multi list chains usually are bidirectional and occasionally are circular to improve update operation.

LINKED ORGANIZATION

Example :

Sequential Data File Sorted on primary key A/C No

Physical Address	A/C No	Name	Amount	A/C Type
1	1111	ABC	500	01
2	2574	XYZ	2000	02
3	2389	STU	3000	03
4	3000	KBC	4000	01
5	2494	YQR	800	01
6	3678	SPZ	500	02

Inverted Index File for secondary key A/C type

A/C Type	Physical Address
01	1,4,5
02	2,6
03	3

LINKED ORGANIZATION

Example :

Physical Address	A/C No	Name	Amount	A/C Type
1	1111	ABC	500	01
2	2574	XYZ	2000	02
3	2389	STU	3000	03
4	3000	KBC	4000	01
5	2494	YQR	800	01
6	3678	SPZ	500	02

A/C Type	Pointer
01	
02	
03	

You can see the advantage of Multi key file organization that you can search all the records of particular A/C type directly from inverted file as well as all records based on A/C no also.

Coral Ring

The coral ring structure is same as the doubly linked multi list structure. Each list is structured as a circular list with a head node. The headnode for the list for the key value Ki = x will have an information field with value x. The field for the key Ki is replaced by a link field. Thus for each record the coral ring contains 2 fields : y↑.alink[i], y↑.blink[i]

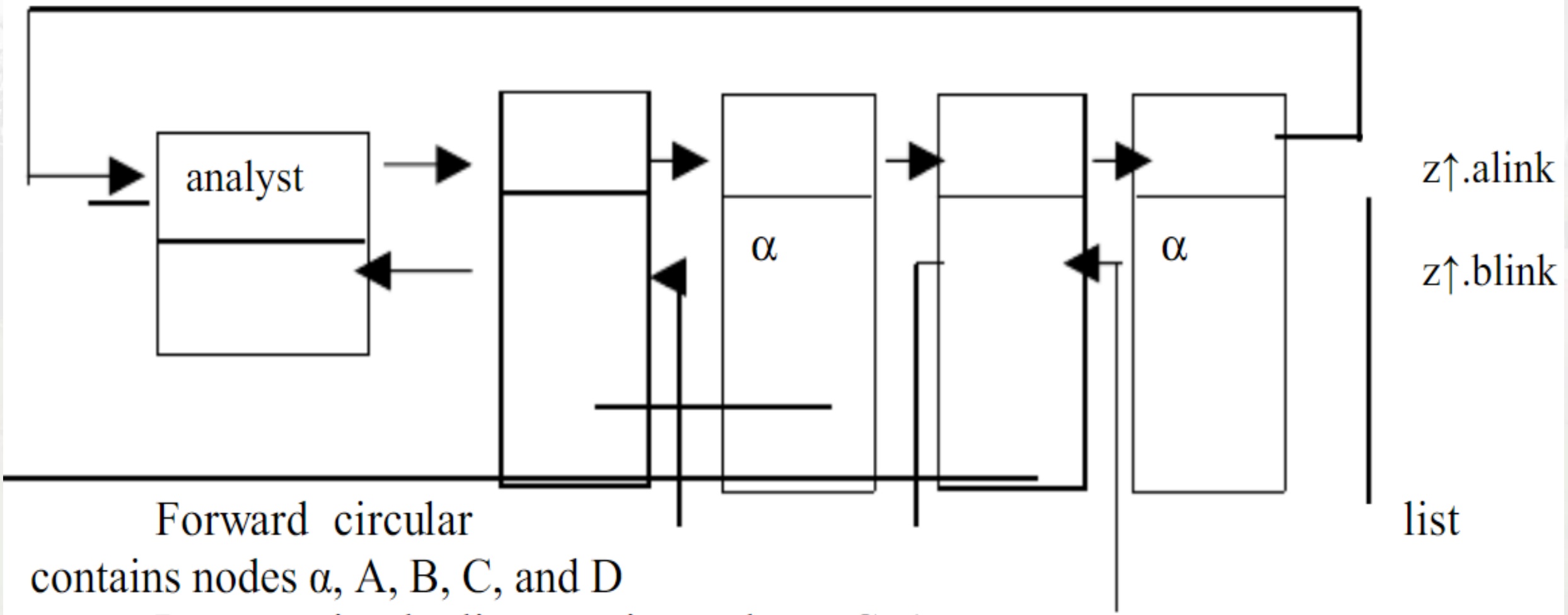
The alink is used to link together all records with the same key Ki. The alinks form a circular list with a head node whose information field retains the value of Ki for the records in the ring.

The blink is used for some records as a back pointer and for some records it is a pointer to the head node. To distinguish between these two y↑.field[i] is used. If y↑.field[i] = 1 then it is a back pointer and y↑.field[i] = 0 it is a pointer to the nearest record z preceding it its circular list for Ki having z↑.blink[i] also a back pointer.

Coral Ring

In any given circular list, all records with back pointers form another circular list in the reverse direction.

The presence of these back pointers makes it possible to carry out the deletion without having to start from the beginning of the list.



Forward circular list contains nodes α, A, B, C, and D

Reverse circular list contains nodes α, C, A

Cellular Partition

In order to reduce file search times, the storage media may be divided into cells. A cell may be an entire disk pack or it may simply be a cylinder. Lists are localized to lie with in a cylinder. If we have a multi list organization in which the list for key1 = prog included records on several different cylinders then we can break the list in to several smaller lists where each prog list included only those records in the same cylinder. By doing this all the records in the same cell (i.e. the same cylinder)may be accessed with out moving the read/write heads. In case a cell is a disk pack then using cellular partitions it is possible to search different cells in parallel.

References

1. https://www.javatpoint.com/dbms-sequential-file-organization
2. https://www.javatpoint.com/opening-and-closing-a-file-in-c-in-cpp-pdf
3. https://www.javatpoint.com/dbms-cluster-file-organization
4. https://www.baeldung.com/cs/file-access
5. https://www.guru99.com/indexing-in-database.html#:~:text=Two%20main%20types%20of%20indexing%20methods%20are%201)Primary%20Indexing,Dense%20Index%202)Sparse%20Index.
6. https://www.geeksforgeeks.org/indexing-in-databases-set-1/

References

1. https://coursecontent.indusuni.ac.in/wp-content/uploads/sites/8/2020/03/CE0417_FDS_UNIT4-ZT-file_structure-converted.pdf

THANK YOU!!!

My Blog : https://anandgharu.wordpress.com/

Email : gharu.anand@gmail.com