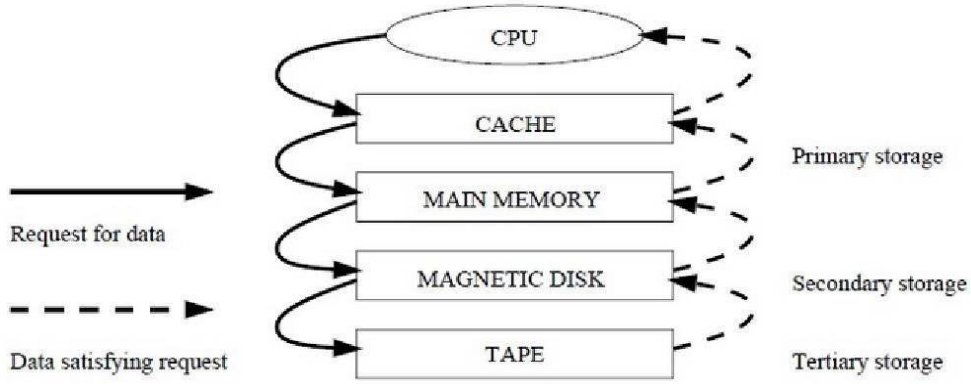
Data on External storage

- A DBMS stores the data on external storage because the amount of data is very huge, must persist across program executions and has to be fetched into main memory when DBMS processes the data.
- The unit of information for reading data from disk, or writing data to disk, is a page. The size of a page is 4KB or 8KB.Memory in a computer system is arranged in a hierarchy, as shown in Figure.
- Atthe top, we have primary storage, which consists of cache and main memory and provides very fast access to data. Then comes secondary storage, which consists of slower devices, such as magnetic disks (Floppy disk & Hard disk drive). Tertiary storage is the slowest class of storage devices; for example, optical disks (CD) and tapes.
- Each record in a file has a unique identifier called a record id, or **rid** for short. A rid has the property that we can identify the disk address of the page containing the record by using the rid.



Buffer Manager:

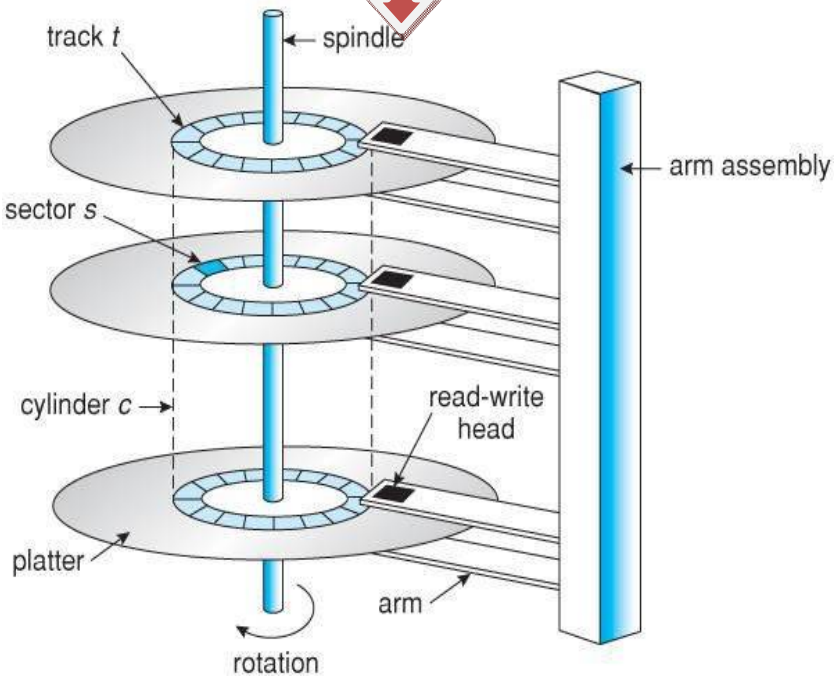
- Data is read into memory for processing, and written to disk for persistent storage, by a layer of software called the buffer manager. When the files and access methods layer (which we often refer to as just the file layer) needs to process a page, it asks the buffer manager to fetch the page, specifying the page's rid. The buffer manager fetches the page from disk if it is not already in memory.

Disk space manager:

- Space on disk is managed by the disk space manager. When the files and access methods layer needs additional space to hold new records in a file, it asks the disk space manager to allocate an additional disk page for the file;

Magnetic Disks:

- Magnetic disks support direct access (transfers the block of data between the memory and peripheral devices of the system, without the participation of the processor) to a desired location and are widely used for database applications. A DBMS provides seamless access to data on disk; applications need not worry about whether data is in main memory or disk.


track t
spindle
arm assembly
sector s
cylinder c
read-write head
platter
arm
rotation

Platter:

- A platter is a circular magnetic plate that is used for storing data in a hard disk. It is often made of aluminum, glass substrate or ceramic. A hard disk drive contains several platters. Each platter has two working surfaces. These surfaces of the platters hold the recorded data.

Spindle: A typical HDD design consists of a spindle, which is a motor that holds the platters.

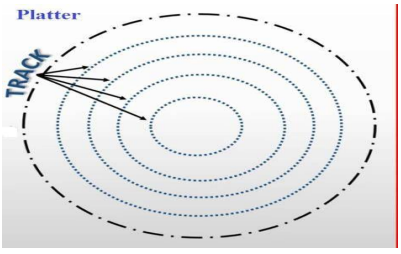
Tracks: Each working surface of the platter is divided into number of concentric rings, which are called tracks.

Cylinder: The collection of all the tracks that are of the same distance, from the edge of the platter, is called a cylinder.

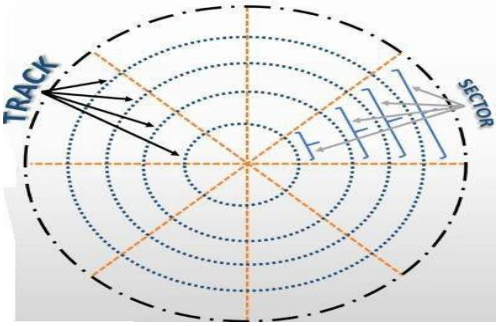
Read / Write Head: The data on a hard drive platter is read by read – write heads, of read – write arm. The read – write arm also known as actuator.

Arm assembly: Each on a separate read – write arm are controlled by a common arm assembly which moves all heads simultaneously from one cylinder to another.

Tracks: Each platter is broken into thousands of tightly packed concentric circles, known as tracks. These tracks resemble the structure of annual rings of a tree. All the information stored on the hard disk is recorded in tracks. Starting from zero at the outer side of the platter, the number of tracks goes on increasing to the inner side. Each track can hold a large amount of data counting to thousands of bytes.



Sectors: Each track is further broken down into smaller units called sectors. As sector is the basic unit of data storage on a hard disk, each track has the same number of sectors, which means that the sectors are packed much closer together on tracks near the center of the disk. A single track typically can have thousands of sectors. The data size of a sector is always a power of two, and is almost always either 512 or 4096 bytes.

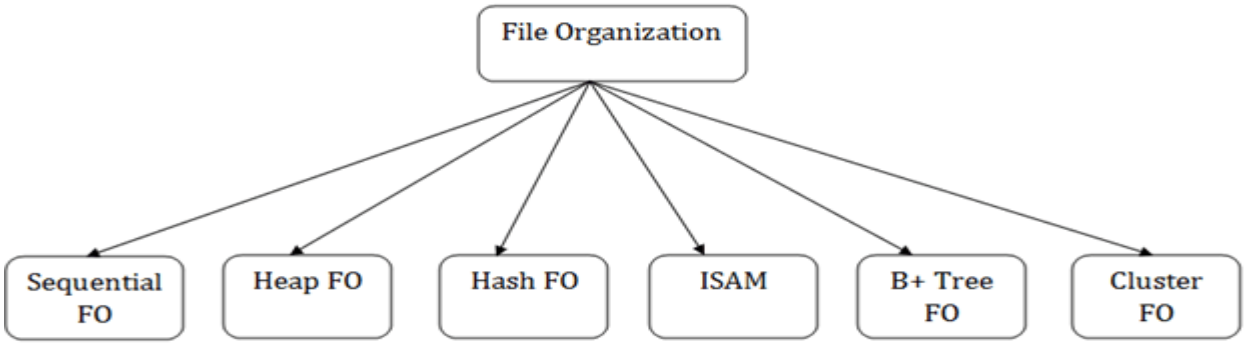


Clusters: Sectors are often grouped together to form clusters. A cluster is the smallest possible unit of storage on a hard disk. If contiguous clusters (clusters thatare next to each other on the disk) are not available, the data is written elsewhere on the disk, and the file is considered to be fragmented.

FILE ORGANIZATION

- The database is stored as a collection of files. Each file contains a set of records. Each record is a collection of fields. For example, a student table (or file) contains many records and each record belongs to one student with fields (attributes) such as Name, Date of birth, class, department, address, etc.
- File organization defines how file records are mapped onto disk blocks.
- The records of a file are stored in the disk blocks because a block is the unit of data transfer between disk and memory.
- When the block size is larger than the record size, each block will contain more than one record. Sometimes, some of the files may have large records that cannot fit in one block.
- In this case, we can store part of a record on one block and the rest on another. A pointer at the end of the first block points to the block containing the remainder of the record.

The different types of file organization are given below:

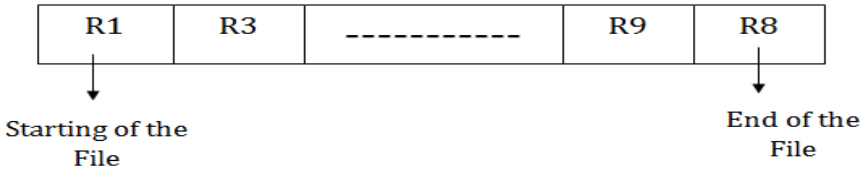


Sequential File Organization

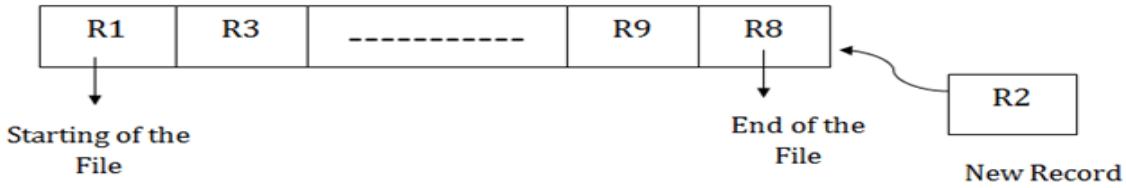
This method is the easiest method for file organization. In this method, files are stored sequentially. This method can be implemented in two ways:

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.

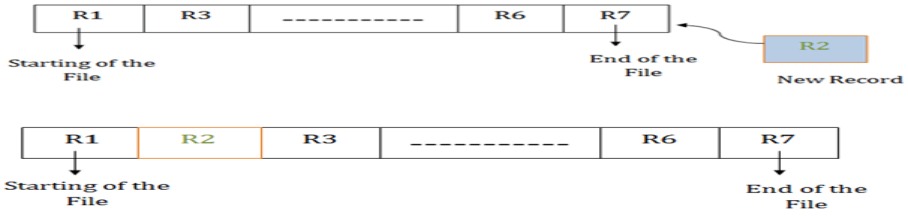


- Suppose we want to insert a new record R2 in the sequence, then it will be placed at the end of the file. Here, records are nothing but a row in any table.



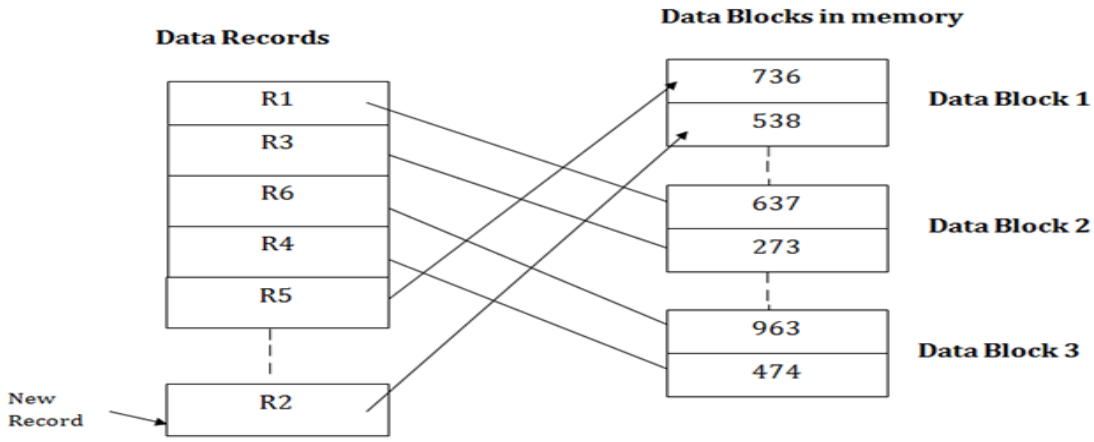
2 Sorted File Method:

- 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.



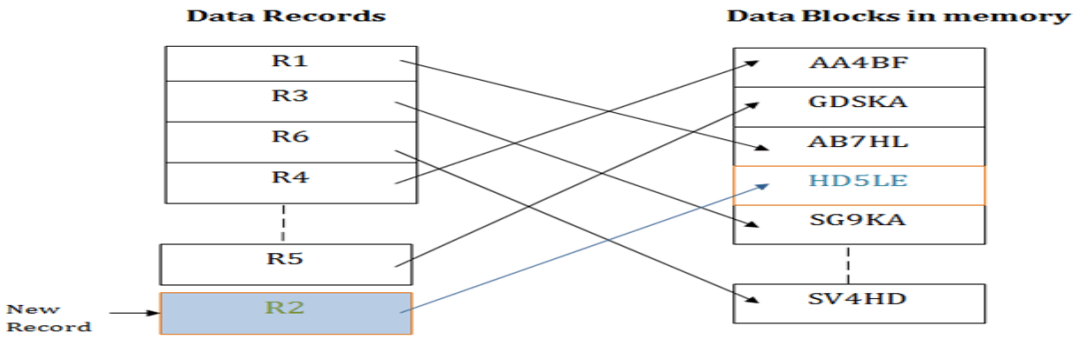
Heap File Organization:

- It is the simplest and most basic type of organization. It works with data blocks. In heap file organization, the records are inserted at the file's end. When the records are inserted, it doesn't require the sorting and ordering of records.
- When the data block is full, the new record is stored in some other block. This new data block need not to be the very next data block, but it can select any data block in the memory to store new records. The heap file is also known as an unordered file.



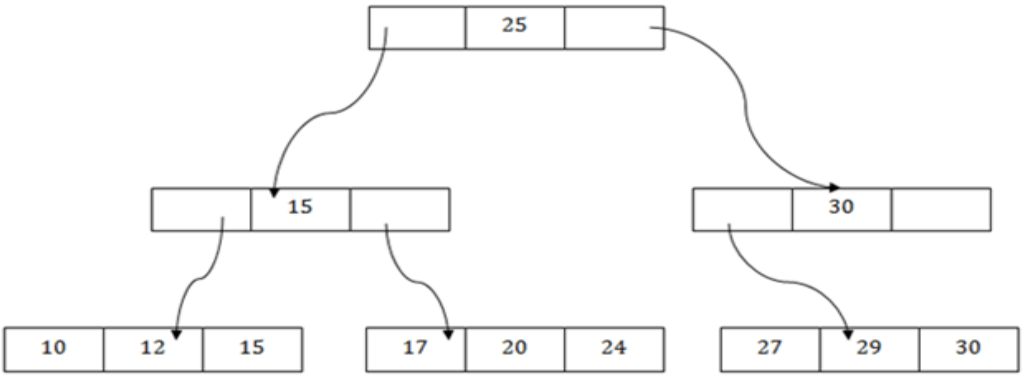
Hash File Organization

Hash File Organization uses the computation of hash function on some fields of the records. The hash function's output determines the location of disk block where the records are to be placed.



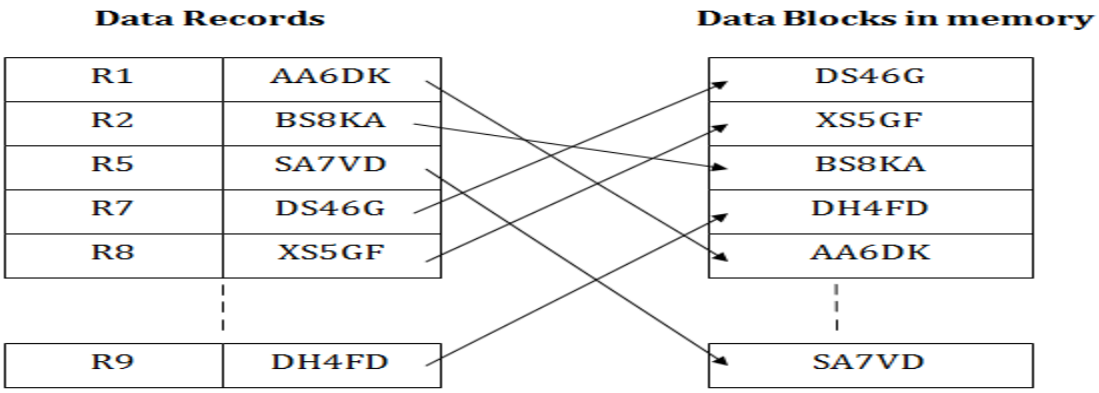
- When a record has to be received using the hash key columns, then the address is generated, and the whole record is retrieved using that address. In the same way, when a new record has to be inserted, then the address is generated using the hash key and record is directly inserted. The same process is applied in the case of delete and update.

B+ File Organization

- B+ tree file organization is the advanced method of an indexed sequential access method. It uses a tree-like structure to store records in File.
- It uses the same concept of key-index where the primary key is used to sort the records. For each primary key, the value of the index is generated and mapped with the record.
- The B+ tree is similar to a binary search tree (BST), but it can have more than two children. In this method, all the records are stored only at the leaf node. Intermediate nodes act as a pointer to the leaf nodes. They do not contain any records.
- 

Indexed sequential access method (ISAM)

- 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.



Cluster file organization

- When the two or more records are stored in the same file, it is known as clusters. These files will have two or more tables in the same data block, and key attributes which are used to map these tables together are stored only once.

- This method reduces the cost of searching for various records in different files.
- The cluster file organization is used when there is a frequent need for joining the tables with the same condition. These joins will give only a few records from both tables. In the given example, we are retrieving the record for only particular departments. This method can't be used to retrieve the record for the entire department.

EMPLOYEE

EMP_ID	EMP_NAME	ADDRESS	DEP_ID
1	John	Delhi	14
2	Robert	Gujarat	12
3	David	Mumbai	15
4	Amelia	Meerut	11
5	Kristen	Noida	14
6	Jackson	Delhi	13
7	Amy	Bihar	10
8	Sonoo	UP	12

DEPARTMENT

DEP_ID	DEP_NAME
10	Math
11	English
12	Java
13	Physics
14	Civil
15	Chemistry

Cluster Key

DEP_ID	DEP_NAME	EMP_ID	EMP_NAME	ADDRESS
10	Math	7	Amy	Bihar
11	English	4	Amelia	Meerut
12	Java	2	Robert	Gujarat
12		8	Sonoo	UP
13	Physics	6	Jackson	Delhi
14	Civil	1	John	Delhi
14		5	Kristen	Noida
15	Chemistry	3	David	Mumbai