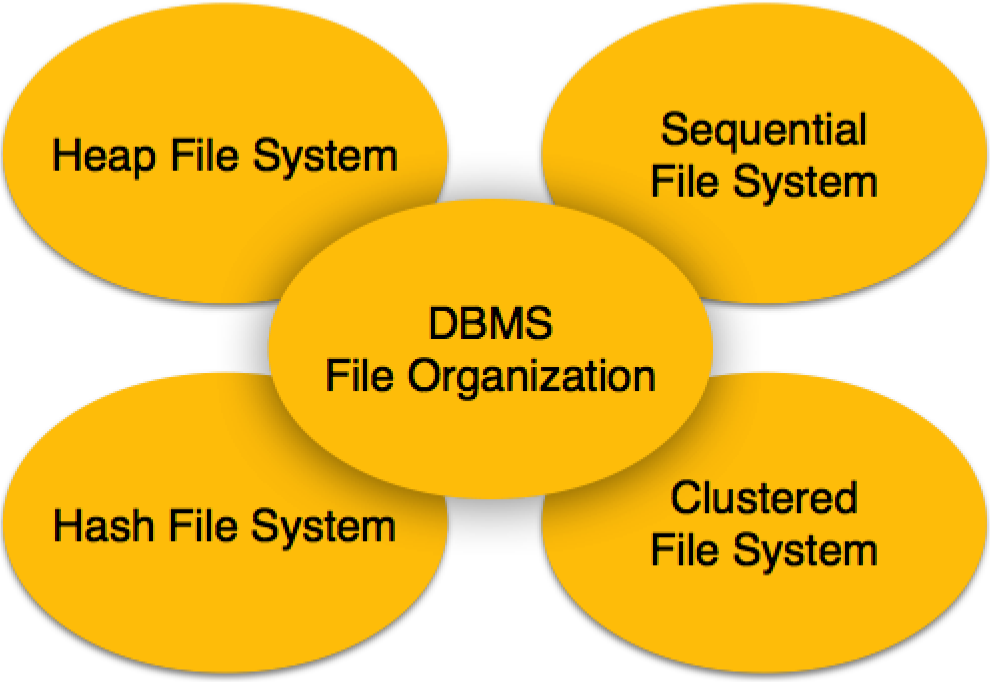
Relative data and information is stored collectively in file formats. A file is a sequence of records stored in binary format. A disk drive is formatted into several blocks that can store records. File records are mapped onto those disk blocks.

File Organization

File Organization defines how file records are mapped onto disk blocks. We have four types of File Organization to organize file records −



Heap File Organization

When a file is created using Heap File Organization, the Operating System allocates memory area to that file without any further accounting details. File records can be placed anywhere in that memory area. It is the responsibility of the software to manage the records. Heap File does not support any ordering, sequencing, or indexing on its own.

Sequential File Organization

Every file record contains a data field (attribute) to uniquely identify that record. In sequential file organization, records are placed in the file in some sequential order based on the unique key field or search key. Practically, it is not possible to store all the records sequentially in physical form.

Hash File Organization

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

Clustered File Organization

Clustered file organization is not considered good for large databases. In this mechanism, related records from one or more relations are kept in the same disk block, that is, the ordering of records is not based on primary key or search key.

File Operations

Operations on database files can be broadly classified into two categories −

- **Update Operations**
- **Retrieval Operations**

Update operations change the data values by insertion, deletion, or update. Retrieval operations, on the other hand, do not alter the data but retrieve them after optional conditional filtering. In both types of operations, selection plays a significant role. Other than creation and deletion of a file, there could be several operations, which can be done on files.

- **Open** − A file can be opened in one of the two modes, **read mode** or **write mode**. In read mode, the operating system does not allow anyone to alter data. In other words, data is read only. Files opened in read mode can be shared among several entities. Write mode allows data modification. Files opened in write mode can be read but cannot be shared.
- **Locate** − Every file has a file pointer, which tells the current position where the data is to be read or written. This pointer can be adjusted accordingly. Using find (seek) operation, it can be moved forward or backward.
- **Read** − By default, when files are opened in read mode, the file pointer points to the beginning of the file. There are options where the user can tell the operating system where to locate the file pointer at the time of opening a file. The very next data to the file pointer is read.
- **Write** − User can select to open a file in write mode, which enables them to edit its contents. It can be deletion, insertion, or modification. The file pointer can be located at the time of opening or can be dynamically changed if the operating system allows to do so.
- **Close** − This is the most important operation from the operating system's point of view. When a request to close a file is generated, the operating system
    - removes all the locks (if in shared mode),
    - saves the data (if altered) to the secondary storage media, and
    - releases all the buffers and file handlers associated with the file.

The organization of data inside a file plays a major role here. The process to locate the file pointer to a desired record inside a file various based on whether the records are arranged sequentially or clustered.

File Organization Storage

There are different ways of storing data in the database. Storing data in files is one of them. A user can store the data in files in an organized manner. These files are organized logically as a sequence of records and reside permanently on disks. Each file is divided into fixed-length storage units known as **Blocks**. These blocks are the units of storage allocation as well as data transfer. Although the default block size in the database is 4 to 8 kilobytes, many databases allow specifying the size at the time of creating the database instance.

Usually, the record size is smaller than the block size. But, for large data items such as images, the size can vary. For accessing the data quickly, it is required that one complete record should reside in one block only. It should not be partially divided between one or two blocks. In RDBMS, the size of tuples varies in different relations. Thus, we need to structure our files in multiple lengths for implementing the records. In file organization, there are two possible ways of representing the records:

- Fixed-length records
- Variable-length records

Let's discuss this in detail.

Fixed-Length Records

Fixed-length records means setting a length and storing the records into the file. If the record size exceeds the fixed size, it gets divided into more than one block. Due to the fixed size there occurs following two problems:

1. Partially storing subparts of the record in more than one block requires access to all the blocks containing the subparts to read or write in it.
2. It is difficult to delete a record in such a file organization. It is because if the size of the existing record is smaller than the block size, then another record or a part fills up the block.

However, including a certain number of bytes is the solution to the above problems. It is known as **File Header**. The allocated file header carries a variety of information about the file, such as the address of the first record. The address of the second record gets stored in the first record and so on. This process is similar to pointers. The method of insertion and deletion is easy in fixed-length records because the space left or freed by the deleted record is exactly similar to the space required to insert the new records. But this process fails for storing the records of variable lengths.

Variable-Length Records

Variable-length records are the records that vary in size. It requires the creation of multiple blocks of multiple sizes to store them. These variable-length records are kept in the following ways in the database system:

1. Storage of multiple record types in a file.
2. It is kept as Record types that enable repeating fields like multisets or arrays.
3. It is kept as Record types that enable variable lengths either for one field or more.