Unit 1 File structure and organization

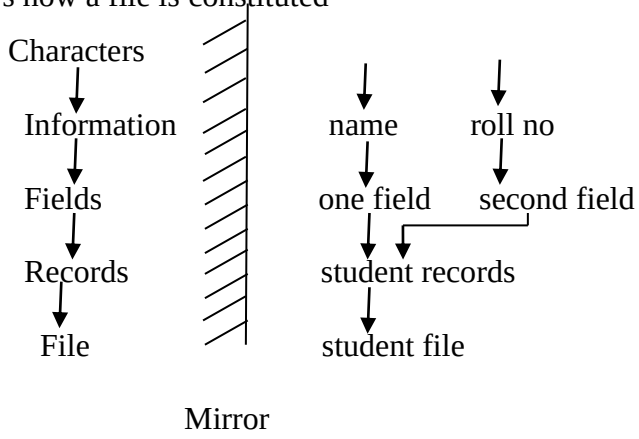
Introduction:-

- A file organization is a method of arranging the records in files.
- The file is stored on secondary storage device called as hard disk.
- A file can be accessed or modified in different ways.
- It is used to determine an efficient file organization for each base relation.
- For example, if we want to retrieve student records in alphabetical order of name, sorting the file by student name is a good file organization. However, if we want to retrieve all students whose marks is in a certain range, a file ordered by student name would not be a good file organization. Some file organizations are efficient for bulk loading data into the database but inefficient for retrieve and other activities.

Logical and Physical files:-

What is file?

- In any information systems we deal with the data.
- This data has to be arranged in a proper way to accept, process and communication operations and results.
- For arranging the data we need files.
- A manual file stores all the information relating to a particular activity.
- For example, inventory activities in a inventory file, payroll activities in a payroll file and so on.
- The basic unit of information for computer and manual files is a record.
- A collection of related data items form a record. For example:- each employee's record will contain data items such as Employee number, Employee name, Basic pay, Allowances, Deductions, Gross pay, Net pay.
- A set of logically related records form or constitute a file.
- Fig. shows how a file is constituted



- **File structure:-**

a) **Character or bit: -**

A smallest unit of data which cannot be further spilt-up is called a 'Bit'. It is a smallest addressable unit by the modern computer.

1 character = 1 bit.

b) Data item:-

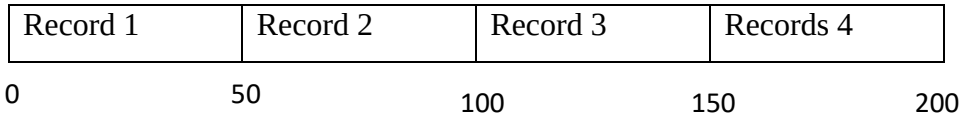
- One or more characters combined may form a data item.
- It is used to describe an attribute of an object or entity.
- For example: - student_no, student_name, age etc. Are data item.
- A data item is also referred to as a field. However, there is a slight difference between data item and field.
- A field is a physical space on a magnetic disc whereas a data item is the data stored in the field.

c) **Record**:-

- The data items related to an object or entity are grouped into a record.
- Record can also be defined as a set of logically related fields.
- There are two types of records:

i. Fixed Length Records:

- If every record in the file has exactly same size(in bytes), the file is said to be made p of fixed length records.
- All records are same in size. Each data is recorded in the same length block. That means suppose if the records of the size 50 bytes. Then records are stored like mansion in following fig. b;



- That means each records can store b the distance of 50 bytes.
- Advantages:-
  - Access become fast since computer knows the exact location where data is store on the fixed location.
- Disadvantages:-
  - Records are unnecessarily larger since all records are to be maintaining same size even they become small.

- They are take huge space from memory.
- Because of their large size they take long time of moving or saving or transferring from one location to another.

ii. Variable Length Records:

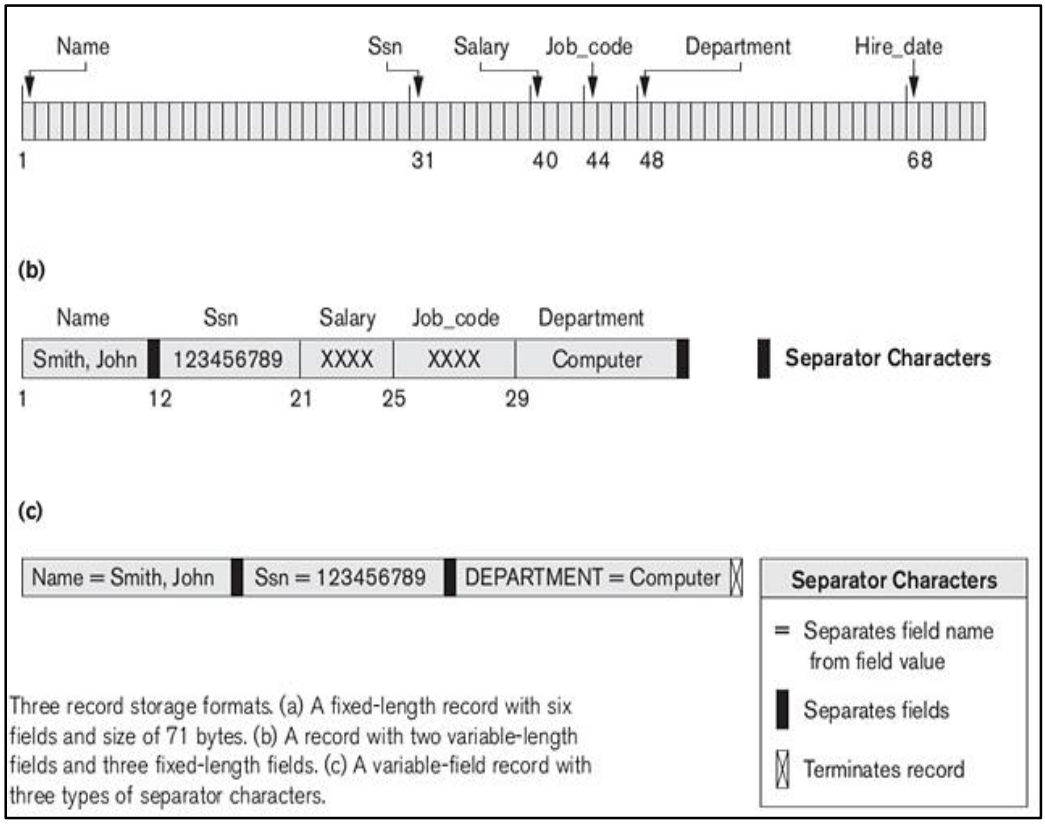
- If different record in the file has different sizes, the file is said to be made up of variable length records.
- Following are some reasons for file may have variable length records:
  - One or more field in the records is of varying size. For example Name and Subject form the student can be variable length field.
  - A record in the file may have repeating fields i.e. multiple values for individual record. For example, one student can take more than one subject.
  - Some fields in the records may be optional.
  - The file contains records of different records types and hence of varying size means a file may have records in records.
- For variable length field each records has a value for each field, but we do not know that exact length of some field values. To determine the bytes within a particular record that represented each field we use special separate charter (for example; ? or % or $ or ⑤) called delimiter.
- This delimiter cannot appear in the field but they are terminating the variable length filed.
- For example; Notation in C programming structure:

```
struct account
{
    char brach_name[20];
    struct account_info
    {
            char acno[10];
            float balabce;
    };
}
```

- Advantages:-
  - Whatever exact size only that much size occupies in the memory by this kind of records. And hence they required less amount of memory.

- Because of its less memory they can fast move or save transfer from location to the another location.

- Disadvantages:
  - Computer takes time to record accessing since because of variable length it unable to find.



Difference Between Fixed length Records and Variable length records:

Sr. No.	Fixed Length Records	Variable Length Records
1	If every records in the file has exactly same size (in bytes), the file is said to be made up of fixed length records.	If different record in the file have different sizes, the file is said to be made up variable length records.
2	It take huge memory	It takes less memory.
3.	All records are of same length and hence they occupy same size of memory.	All records are differ in size.
4.	Access become fast due to computer knows exact location of records.	Access become slower since computer doesn't know the exact location of records.
5.	Slow in transferring due to its large size.	Fast in transferring due to its small in size.

Physical File and Logical File:

a. Physical File:
   - How the data or record of the file stored actually on the storage device like disc or tape is nothing but physical file.
   - How the operation done with file is concern with physical file.

b. Logical File:
   - What kind of the data or record is stored in storage, is view by logical file.
   - What kind of the operation done with the file is concerned with logical file.

Difference between Physical and logical file:

Sr. No.	Physical File	Logical File.
1	How the data or record of the file stored actually on the storage device like disc or tape is nothing but physical file. How the operation done with file is concern with physical file.	What kind of the data or record is stored in storage, is view by logical file. What kind of the operation done with the file is concerned with logical file?
2	Occupy memory space.	Cannot occupy memory space.
3	Contains data.	Doesn't contain any data.
4	CRTPF command is used to created physical file object.	CRTLF command is used to created logical file object.
5.	Physical file contains one record format.	A logical file contain up to 32 record formats.

Basic File Operation:-

- A file is an abstract data type. To define a file properly, we need to consider the operations that can be performed on files.
- When we want to use a file then we required to open it. For that we must know the following information.
  1. File pointer: - It denotes the location of the file which is currently accessible for read or writes operations. There are two types of file pointers. Get pointers and put pointer which points to locations of read and write positions respectively.
  2. File open count: -It counts total opened files.
  3. File mode: - There are many modes-Input mode, Output mode, and input and output mode etc.

The operating system can provide system calls to create, write, read, reposition, delete.

1. **Opening File**:
   - If we know that the physical file already exists, we can open that file. The program pointer is positioned at the beginning of the file and stored contents are not changed. (See fig)

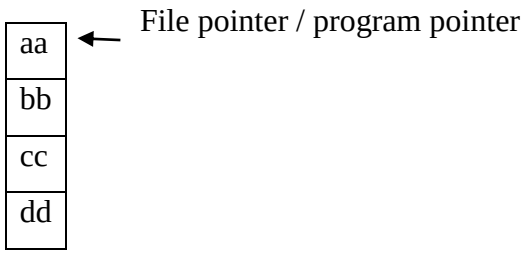


Fig: Opening an existing file

   - For example: picking up a phone to receiving a call.
   - If we want to create a new file, it will open a blank file with the program pointer. Then we can write something into the file.
   - But if the same name of a available physical file is given, then the contents are removed first and the file opened is blank again. (fig)
   - For example: picking up a phone to place a cell.

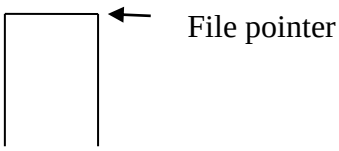


Fig. Create a new file

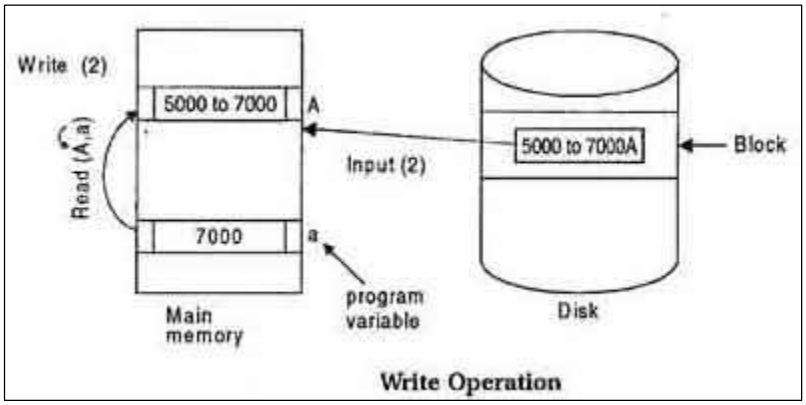
2. **Closing Files**:
   - In terms of telephone example, hanging up the phone is most important job.
   - Unless the phone hangs up, the phone line is not available for placing another call or receiving a call.
   - This is as good as closing a file. When we close a file,, like logical file is available for new equation.
   - Closing of file ensures that the buffer for that file has been flushed of data to a physical file.
   - Operating system itself closes a file when program terminates normally.
   - But for the protection of data against data loss closing a file is most important criteria.
   - When you open a file, it is duplicates and the contents are taken into primary memory Fig and when you say close

3. **Create a file:-**

- When file is created then first operating system do the following steps:
- Search whether file name of that file already exists in the directory or not? If it is exists then it must show error message.
- If it does not exist then make the new entry in directory with file attributes. Means file is opened into output mode for write operation. Here file pointer is pointed to first location of the file.
- If finds free space in the secondary storage and store content of file on that free space.
- Store end of file marker at least position of the file.

4. **Writing a file:-**

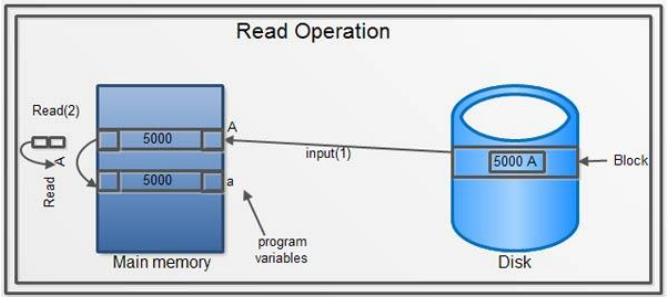
- For writing to a file system call is made specifying the name of the file and the information to be written to the file.
- The operating system searches for the file entry in the directory. The directory entry must store a pointer to the current end of the file. Means file is opened at appended mode. Here file pointer is pointed to last location of the file.



Write Operation

- Using this pointer, address of the next block can be computed and the information can be written, means added into a file.
- Store end of file marker at last position of the file.

5. **Reading a file:-**



- To read from a file, a system call specifies the name of the file.
- Operating system searches the directory for he named file and the directory also needs to have a read pointing to the next block to read.
- File is opened in input mode. Here file pointer is pointed to first location of the file.
- Now it reads content of files till it does not end.

6. **Deleting the file**:-

- To delete a file, operating system follows the steps:
- It searches the directory for the name of file.
- Find out the directory entry.
- It release all file space and adds it to free space.
- Invalidate the directory entry.

7. **Seeing the file:-**

- Generally, we read through the file sequentially i.e. reading one byte after another until we reach the end-of-file.
- Every time a byte is read, the read/write pointer (file pointer) moves ahead to read the next byte.
- This is done by the operating system.
- Suppose there are 20 thousand bytes and after reading first byte we want to read say 19 thousand byte without wasting time through sequential read.
- We want to jump there.
- For this we must be able to control the file pointer so that it is moved according to our need.
- The action of moving directly to a certain position in a file is called as seeking.

- Syntax of seeking is;

  Seek (source-file, offset)

  Here, source file is logical filename and the offset will give the number of where the file pointer should move from the start of file.

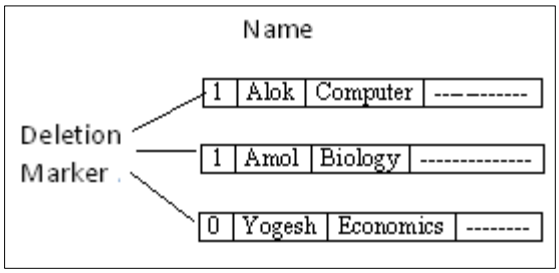
Types of File Organization:-

- File organization: A file organization refers to the organization of the data of file into records, blocks and access structure.
- The file organization is the way of arranging the records in the file when the file is store on disk.
- This means file organization places the records and blocks onto storage medium and assign there relation to each other.

Files of Unordered Records (Heap files)

- Definition: Records are placed in the file in the order in which they are inserted, so new records are inserted at the end of the file. Such an organization called as the heap file or pile file.
- **Some techniques in the Heap file:-**
  1. Insertion of the new record:-
     - Addition of the new record is very efficient.
     - Following steps are inserting new records in the file.
       - The last disk block of the file is copied into a buffer (Temporary storage).
       - Add new record.
       - Rewrite the block from buffer to disk.
       - The address of the last file block is kept in the file header.
  2. Deletion of the previous record:-
     - There are two methods to delete the record from file using Buffer and using deletion marker.
     - Using Buffer:-
     - First method include steps as follows:
       - First find the block which is to be deleted; this finding process is done with program.
       - Copy that block into buffer.
       - Delete the record block from the buffer.
       - Finally, rewrite the block back to the disk.
     - The disadvantage of this method is that it leaves the unused space in the disk storage. Means when we delete large number of records it shows wasted storage space.

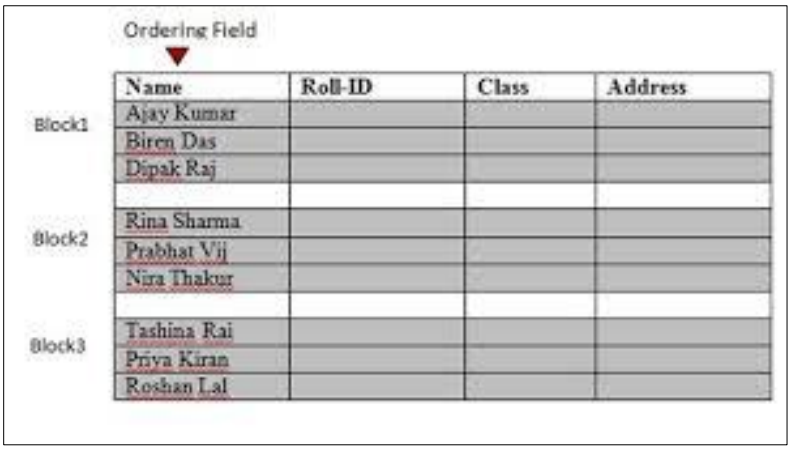
- Using deletion marker:-
  - An extra byte or bit called a deletion marker, stored with each record. A record is setting by deletion marker to a certain value. A different value of the marker indicates a valid (not deleted) record.
  - In fig., each record consists of one extra bit that is deletion marker. Each record consists value 1 if record remains as it is in the file and value 0 is deleted. Since the record of Alok and Amol is as it is in the file while record of Yogesh is deleted from file.



3. Searching of the records:
   a. This is very difficult and expensive process of searching the records.
   b. Linear search is used for searching the record from file, means record is searched block by block.

Files of ordered records (Sorted Files):-

- We can physically order the records of files on the disk based on the values of one of their fields called ordering fields. This leads to an ordered or sequential file.
- If the ordering field is also a key of the file that field is guaranteed to have unique value in each record, this field is called ordering key for the file.
- Following Fig shows the ordered file with Name as ordering key field (Assuming each name of the employee is different from each other).



- There are some advantages of ordered records over the unordered files which are as follows:
  1. Reading the records in order of the ordering key values becomes extremely efficient, because no sorting is required. This kind of file is best for the range selection is desired since records are arranged in ordered bock.

2. Finding the next record from the current one in order key usually requires no additional block access, because the next record is in the same block as the current one.
3. Using a search condition based on the values of the ordering key field results in faster access when the binary search technique is used.

- Insertion and deletion of the records are expensive operations and the records must remain physically ordered.

i. Insertion of new records:-
   - To insertion a record, we must find correct position in the file based on ordering field value.
   - After that make space in the file to insert the record in that position.
   - On the average half the records of the file must be moved to make space for new records.

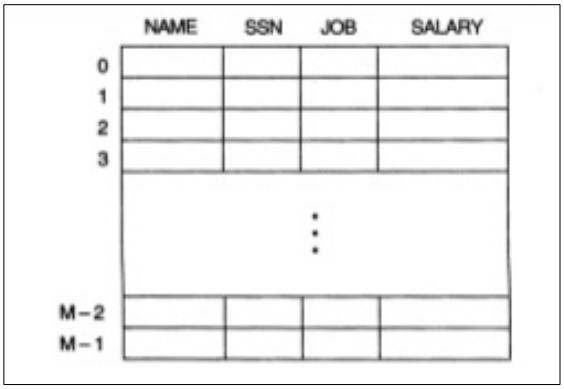
ii. Deletion of previous records:-
   - For deletion, the deleted records are marked and periodically reorganize the file.

iii. Modification of records:
   - Modification is based on two factors:
     1. Search condition to locate the record.
     2. The field to be modified.
   - If the search condition involves the ordering key field, we can located record using binary search. Otherwise we must do liner search.
   - Modifying ordering field means that the record can change its position in the field which requires deletion of the record followed by insertion of the modified record.

Hash File Organized Techniques:-

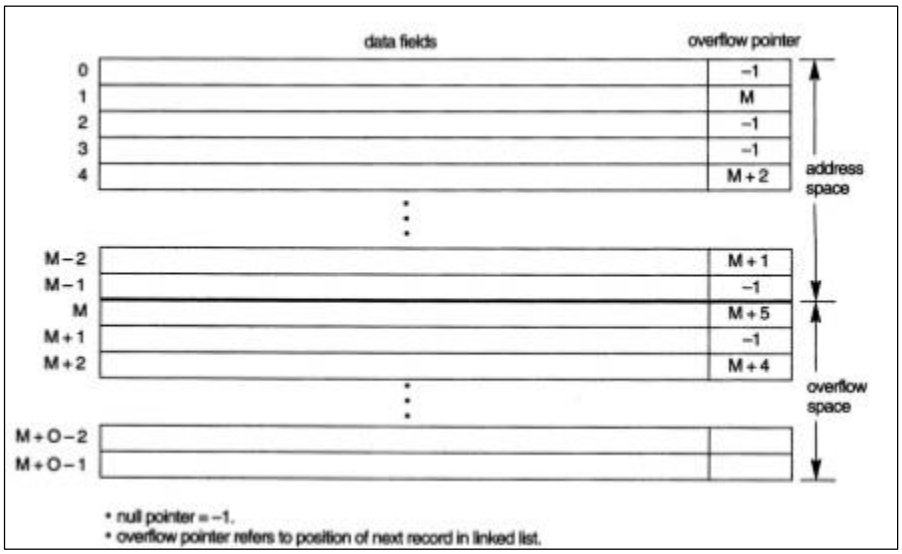
- Definition: - Hashing technique provides the fast access to record on certain search condition. This organization is usually called hash file.
- The search condition must be equality condition on a single field called hash field. In some cases hash field acts like key field of file, in that case it is called as hash key.
- The idea behind the hashing is to provide a function h called a hash function or randomizing function, that is applied to the hash field value of record and yields the address of the disk block in which the record is store.
- A search for the record within the block can be carried out in the main memory buffer.

(A) Internal hashing:-



- For internal files, hashing is typically implemented through the use of an array of records.
- Suppose that the array of index range is from 0 to M-1, (Ref fig.) then there are M slots whose addresses correspond to the array indexes.
- We choose a hash function the transforms the field value into an integer between 0 to M-1.
- One famous hash function is the h(K)=K MOD M function.
    - The above function returns the remainder of an integer hash field value of K after division by M; this value is then used for the record address.
    - Non-integer hash field's values can be transformed into integer before the MOD function is applied. If the field value is charter then numeric (ASCII) codes associated with charters are used are used in the transformation.
    - A collision occurs when the hash field value of a new record that is being inserted hashes to an address that already contains different records.
    - In such situation we must the new record in some other position is called collision resolution. There are three methods of collision resolution
        1. Open addressing:-

           Proceeding from the occupied position specified by the hash address, the program checks the subsequent position in order until an unused empty position is found.
        2. Chaining:-
            - For this method various overflow location are kept, usually by extending the array with a number of overflow position.
            - In addition, a pointer field is added to each record location.
            - A collision is resolve by placing the new record in an unused overflow location and setting the pointer of the occupied hash address location to the address of that overflow location.
            - A linked list of overflow records for each hash address is maintained (Refer fig.)



3. Multiple hashing:-
    - The program applies a second hash function if the first results in collision. Is another collision results, the program uses open addressing or applies a third hash function and then uses open addressing if necessary.
    - Each collision resolution method requires its own algorithms for insertion, deletion as well as retrieval of the record.
    - The goal of good hashing function is to distribute the records uniformly over the address space so as to minimize collision while not leaving may unused location.

(B) External Hashing:-

- Hashing for disk file is called external hashing.
- To suit the characteristics of disk storage the target address space is made of buckets, each of which holds multiple records.
- A bucket is either one disk block or a cluster of continuous blocks.
- The hashing function maps a key into relative bucket number, rather than assign an absolute block address to the bucket.
- A table maintained in the file header coverts the bucket number into the corresponding disk block address which is shown in the following Fig.

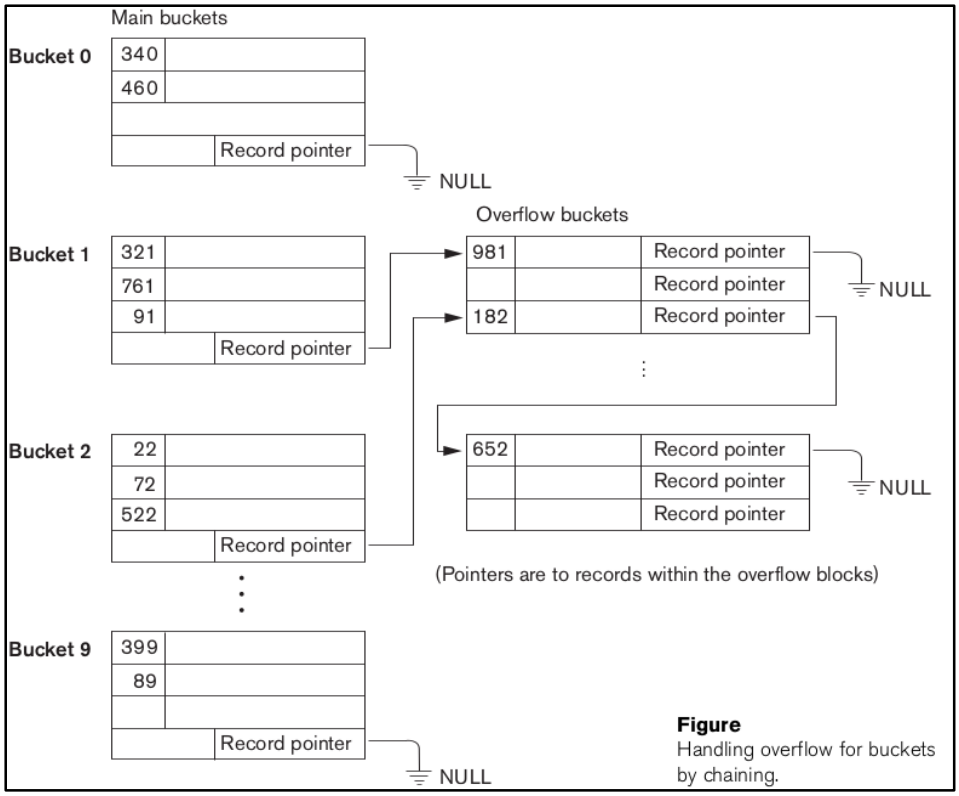


Figure
Handling overflow for buckets by chaining.

- The collision problem is less serve with buckets, because as many records as will fit in a bucket can hash to the same bucket without causing problems.
- If bucket is null, we can use a variation of changing in which the maintain a pointer in each bucket to a link list of overflow records for the bucket to a link list of overflow records for the bucket as shown in fig.
- The pointers in the linked list should be records pointers, which includes both address and a relative record position within the block.

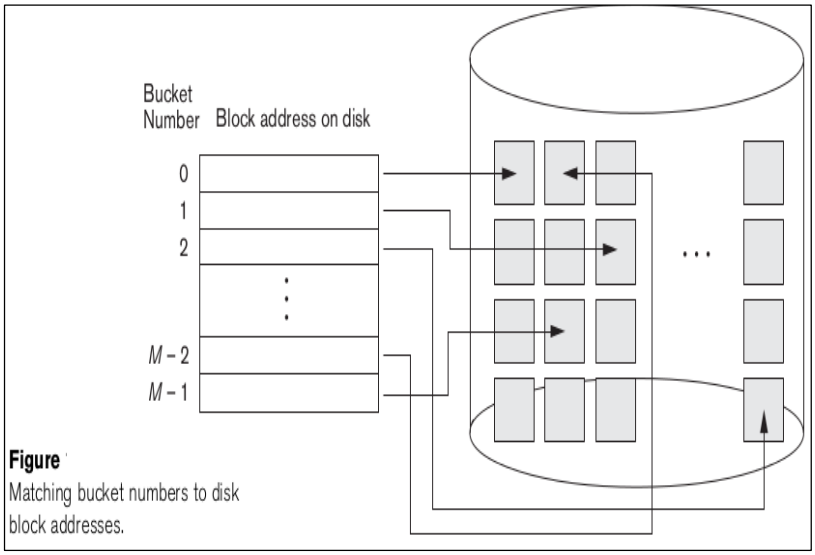


Figure
Matching bucket numbers to disk block addresses.

Choosing a File Organization:

- According to view on the more advantages and few disadvantages we have to choose the exact file organization. Why we choose the above three criteria on their principle, the points are given below;
- Advantageous points are with ✔ symbol and disadvantageous points are with ✕ symbol.

  Heap File:-
  - ✓ Good storage efficiency.
  - ✓ Fast scan, insertion and deletion.
  - × Slow search.

  Sorted file:-
  - ✓ Good storage efficiency.
  - ✓ Fast search.
  - × Slow insertion and deletion.

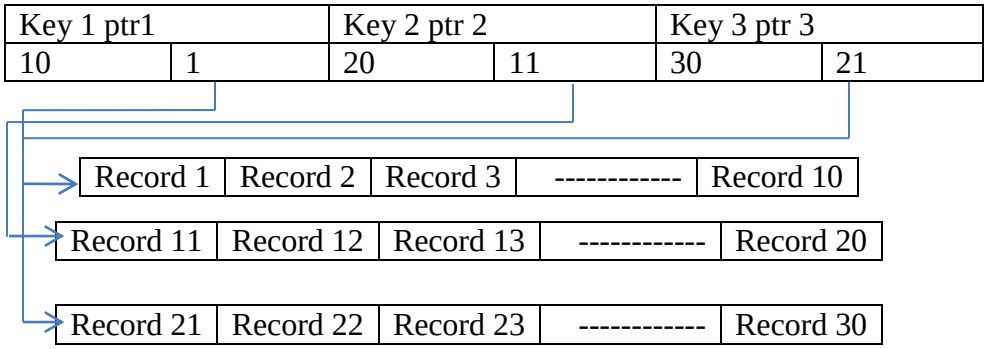
  Hash File:-
  - ✓ Insertion and deletion is fast.
  - ✓ Fast equality search.
  - × When number of pages increases then full scan is slower.
  - × Scanning cost increase.

  Conclusion:-
  - No one file is best for each and every situation. Every file show different advantages means heap file is useful for full file scan, sorted file useful for best range selection and hash file is useful for equality selection.
  - The best file organization is selected according to w purpose of the file is the file.

Indexed Files:

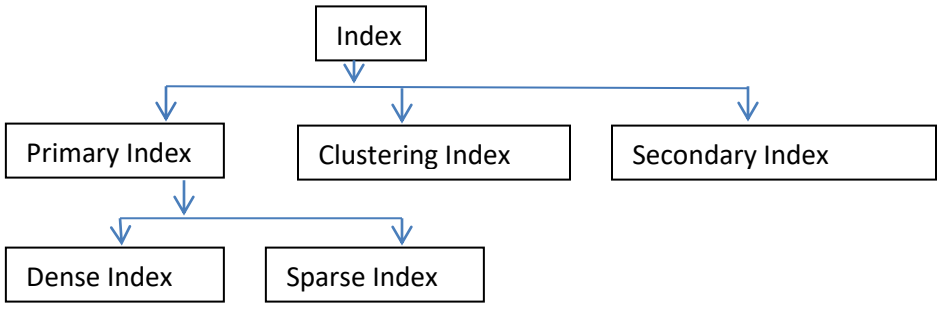
- It is also called as direct file.
- An index is a data structure the organizes data records on disk to optimize certain file operation.
- An index allows us to efficiently search or retrieve all records. Using an index, we can achieve a fast search of data records.
- In the following ways, the files are organized:
  - The data file is stored in the order of the primary key value.
  - The index file contains two fields: the key value and the pointer to the data records.

- One record in the index file thus, consists of a key value and a pointer to the corresponding data record. The pointer points to the first entry within the range of data records.



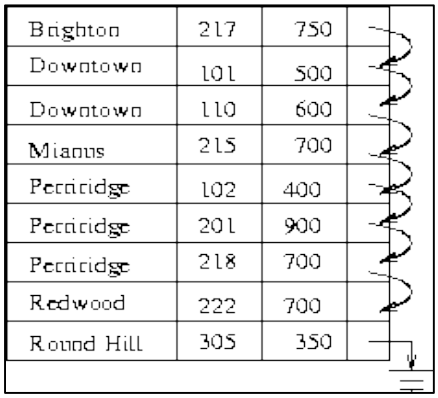
- Generally the key value is the largest primary key value in a given range of records.
- Types of Indexes:

  There are several types of ordered indexes which are shown in Fig.



1. Primary Index:-

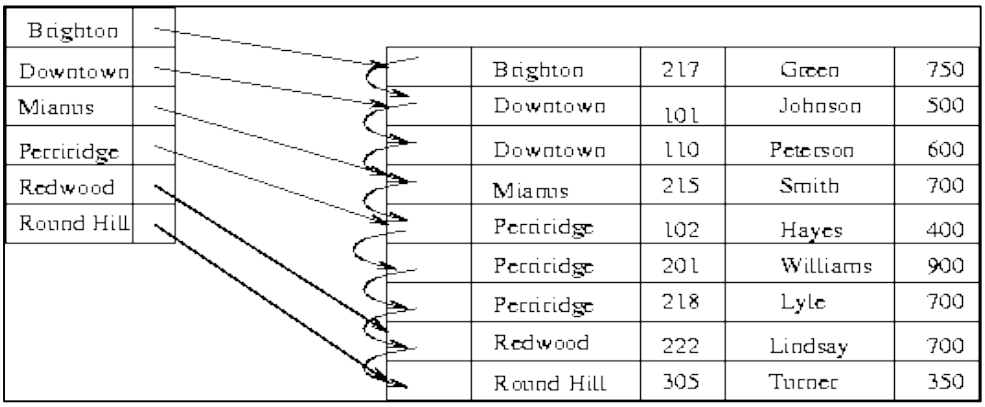
- A primary index is an ordered file, whose records is of fixed length with two fields- primary key and address pointer.
- Key value: (K (i)) is the first field which having some data as the ordering key field.
- Address pointer: (P (i)) refers to that location of disk where data file is actually stored.
- There is one index entry (or index record) which is present in index file for each block in the data file.



- There are two types of primary index field, as follows:

A) Dense Index:

- An index record appears for every search-key value (for each record) in the file, such index is called as dense index.
- The record contains the search-key value and a pointer to the record.



Brighton	217	Green	750
Downtown	101	Johnson	500
Downtown	110	Peterson	600
Miamus	215	Smith	700
Perridge	102	Hayes	400
Perridge	201	Williams	900
Perridge	218	Lyle	700
Redwood	222	Lindsay	700
Round Hill	305	Turner	350

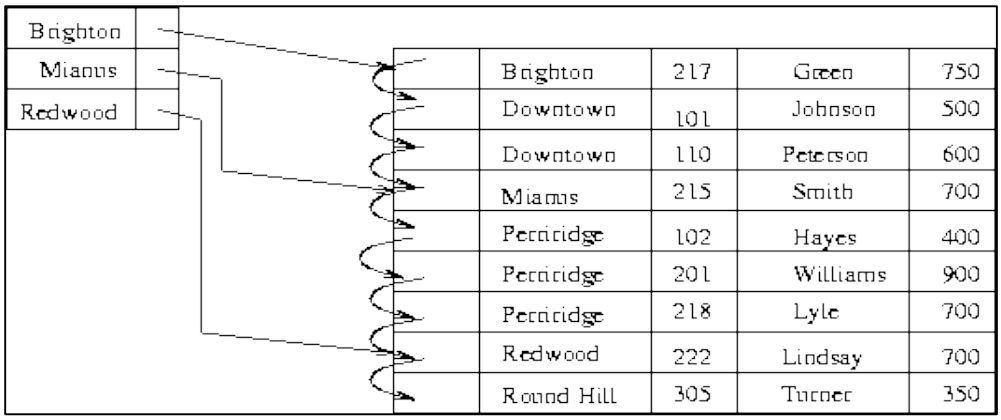
- The above fig. shows the dense index, the index file contains two fields one is search key for location and another is address pointer.
- The data file contains the records of area-wise hotels.
- The address pointer links with each and every different area.
- Deletion:-

  For dense index, we delete a search-key value for the record deletion.
- Insertion:-

  First search key value appearing in the record to be inserted. If the index is dense, and the search-key value does not appear in the index, insert it.
- Advantages:
  - Every record is random access.
  - It is very use full for the sequential access files that means variable record length files.
  - Index table is also sorted so binary search is used for searching.
  - Access speed is very high as compare to direct ordinary file.
  - We can avoid the sequential sort.
- Disadvantages:
  - Extra space is required to store index table.
  - Key of every record is present in index table so index table is very large.

B) Sparse Index:-

- Index records are created for only some of the records; such index is known as sparse index.
- A prime index is hence sparse index, since it includes as entry for each disk block of the file rather than for every search value.
- Because of above reason, the search key in index value has less number than the total available block.

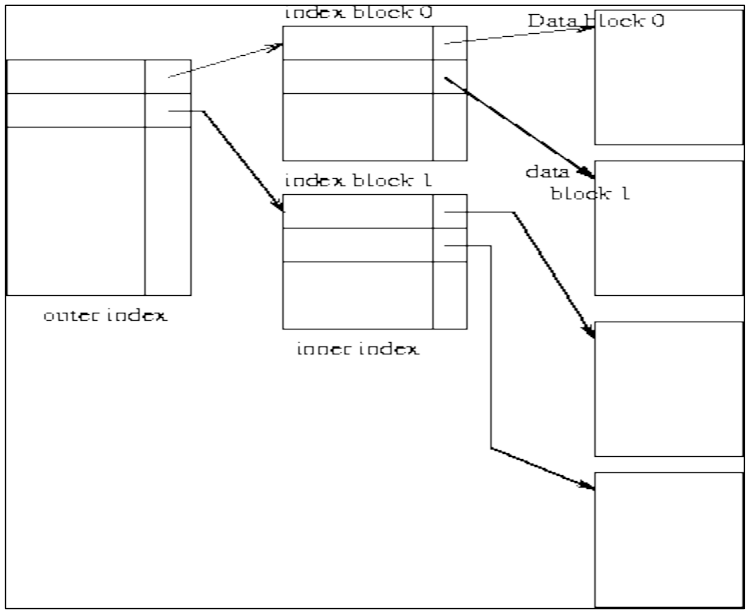


- Deletion:-

  In order to delete the record, it is necessary to see the record to be deleted. If the deleted record was the last record with its particular search-key value, then we delete the search-key value for the record deletion. We delete the key value for replacing its entry in the index (if one exists) with the next search-key value. If the next search-key value is already has an index entry, we delete the entry.

- Insertion:-

  If the index is sparse index, no change needs to be made to the index unless a new block is created. In this case, the first search-key value appearing in the new block is inserted into database.

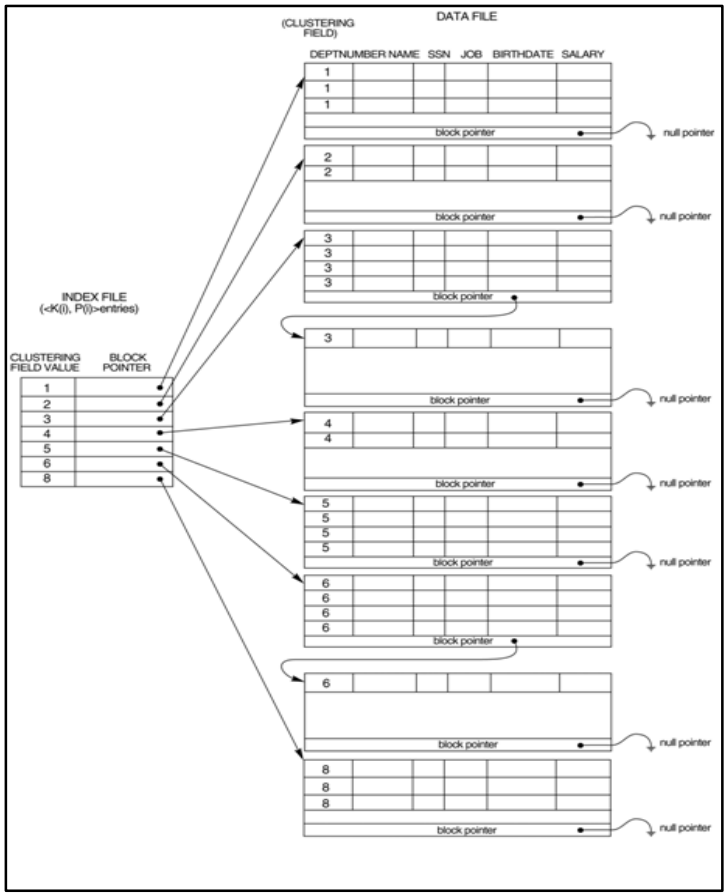


- Advantages:-
  - Index table of sparse index is very small as compared to dense index
  - It is very useful for the sequential access files that mean variable record length files.
  - Index table is also sorted so binary search is used for searching.
  - Access speed is very high as compare to ordinary file.
- Disadvantages:-
  - Extra space is required to store index table.
  - Access speed is very slow as compare to dense index.

2) Clustering Index:-
   - If the record of a file are physically ordered on a non-key field that does not have a district value for each record, that field is called clustering field. On the basic of clustering field we make different type of index called as the clustering index.
   - Basically, clustering index is used to speed up retrieval of the records that have the same value for the clustering field.
   - Clustering index is different from a primary index, which requires that the ordering field of the data file have a distinct value for each record.
   - A clustering index is also an ordered file with two fields;
     - Same type as the clustering field of the data file and
     - Address pointer to the block.
   - There is the one entry in the clustering index for each different value of the clustering field, containing the value and a pointer to the first block in the data file that has a record with that

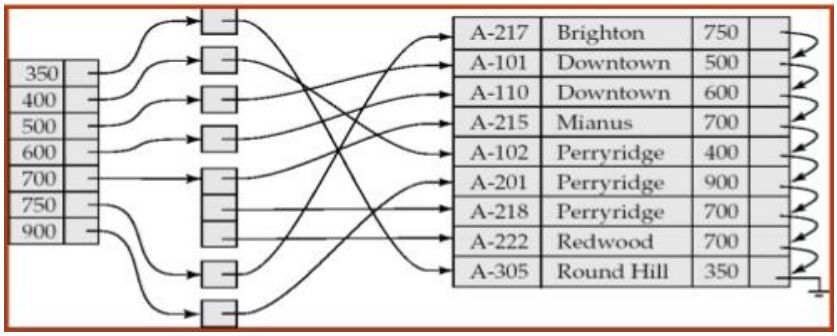
value for its clustering filed, containing the value and a pointer to the first block in the data file that has the record with that value for its clustering field.

- There is a problem with the clustering index for insertion and deletion of records, reason of this problem is that all records are physically ordered.
- To avoid this problem of insertion, it is common to reserve a whole block for each value of the clustering field; all records with that value are placed in the block.
- The fig. shows this schemes;



Secondary index:-

- A secondary index is an order file with the two attributes:-
  - Indexing field: - is the first field which having some data as some non-ordering key of data file.
  - Blocking: - refers to that location of disk where data file is actually stored.
- A secondary access structure on a key filed that has a distinct value for every record. Such field sometimes called as secondary key.



- Fig shows the secondary index;
- The pointers used in the indices do not point directly to the file, they point to a bucket which contains pointer to the file.
- A bucket is collection of one or more blocks chained together.
- This feature allows this entire pointer for one secondary search-key value to be stored together. It is useful in queries for which we do considerable processing using only the pointers. For primary keys, we can obtain all the pointers for one primary search-key value using a sequential scan.

Question: