

Unit 3

Lecture 50

File Organization

File Organization

- A file organization is a way of arranging the records in a file when the file is stored on secondary storage (disk, tape etc.).
- The different ways of arranging the records enable different operations to be carried out efficiently over the file.
- A database management system supports several file organization techniques. The most important task of DBA is to choose a best organization for each file, based on its use.

File Organization

- The organization of records in a file is influenced by number of factors that must be taken into consideration while choosing a particular technique. These factors are
 - a) fast retrieval, updation and transfer of records,
 - b) efficient use of disk space,
 - c) high throughput,
 - d) type of use,
 - e) efficient manipulation,
 - f) security from unauthorized access,
 - g) scalability,
 - h) reduction in cost,
 - i) protection from failure.

File Organization

- A file is a collection of related sequence of records.
- A collection of field names and their corresponding data types constitutes a *record*.
- A *data type*, associated with each field, specifies the types of values a field can take.
- All records in a file are of the same record type.

Records and Record Types

- Data is generally stored in the form of records.
- A record is a collection of fields or data items and data items is formed of one or more bytes.
- Each record has a unique identifier called record-id.
- The records in a file are one of the following two types:
 1. Fixed length records.
 2. Variable length records.

Fixed Length Records

- Every record in the file has exactly the same size (in bytes).
- The record slots are uniform and are arranged in a continuous manner in the file.
- A record is identified using both record-id and **slot number** of the record.
- Example : The Figure 1 shows a structure of fixed length STUDENT record and the Figure 2 shows a portion of a file of fixed length records.

```
type STUDENT = record
    NAME = char(20);
    Roll No = char(5);
    DOB = char(8);
end
```

Figure 1

Name	Roll No.	DOB
Naresh	3234	28-02-75
Suresh	5132	20-05-80
Ramesh	3535	24-10-77
Ashish	3987	15-09-72
Manish	4321	18-11-70
Harish	4983	09-06-73
Manoj	3590	05-01-81

Figure 2

Fixed Length Records

- ***Advantage of Fixed Length Records***

1. Insertion and deletion of records in the file are simple to implement since the space made available by a deleted record is same as needed to insert a new record.

- ***Disadvantage of Fixed Length Records***

1. In fixed length records, since the length of record is fixed, it causes wastage of memory space. For example, if the length is set up to 50 characters and most of the records are less than 25 characters, it causes wastage of precious memory space.
2. It is an inflexible approach. For example, if it is required to increase the length of a record, then major changes in program and database are needed.

Variable Length Records

- Every record in the file need not be of the same size (in bytes). Therefore, the records in the file have different sizes.
- The major problem with variable length record is that when a new record is to be inserted, an empty slot of the exact length is required.
- If the slot is smaller, it cannot be used and if it is too big, the extra space is just wasted.
- There are two methods to store variable length records with a fixed length representation.
- These are **Reserved space** and **pointers**.

Variable Length Records

- A file may have variable length records due to following reasons:
 1. One or more than one fields of a record are of varying size but the records in the file are of same record type.
 2. One or more than one fields may have multiple values for individual records and are called repeating fields but the records in the file are of same record type.
 3. One or more than one fields are optional *i.e.*, they may have values for some but not for all records. The file records in this case are also of the same record type.
 4. The file contains records of different record types and different sizes.

Variable Length Records

- ***Advantage of Variable Length Records***

1. It reduces manual mistakes as database automatically adjust the size of record.
2. It saves lot of memory space in case of records of variable lengths.
3. It is a flexible approach since future enhancements are very easy to implement.

- ***Disadvantage of Variable Length Records***

1. It increases the overhead of DBMS because database have to keep record of the sizes of all records.

Record Blocking and Spanned versus Unspanned Records

- The records of a file must be allocated to 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 numerous records, although some files may have unusually large records that cannot fit in one block.
- Suppose that the block size is B bytes. For a file of fixed-length records of size R bytes, with $B \geq R$, we can fit $bfr = \lfloor B/R \rfloor$ records per block, where the $\lfloor x \rfloor$ (*floor function*) rounds down the number x to an integer.
- The value bfr is called the **blocking factor** for the file. In general, R may not divide B exactly, so we have some unused space in each block equal to $B - (bfr * R)$ bytes.

Record Blocking and Spanned versus Unspanned Records

- To utilize this unused space, 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 in case it is not the next consecutive block on disk. This organization is called **spanned** because records can span more than one block.
- Whenever a record is larger than a block, we *must* use a spanned organization.
- If records are not allowed to cross block boundaries, the organization is called **unspanned**. This is used with fixed-length records having $B > R$ because it makes each record start at a known location in the block, simplifying record processing.
- For variable-length records, either a spanned or an unspanned organization can be used.
- If the average record is large, it is advantageous to use spanning to reduce the lost space in each block.

Record Blocking and Spanned versus Unspanned Records

- For variable-length records using spanned organization, each block may store a different number of records.
- In this case, the blocking factor *bfr* represents the *average* number of records per block for the file.
- We can use *bfr* to calculate the number of blocks *b* needed for a file of *r* records :

$$b = \lceil (r/bfr) \rceil \text{ blocks}$$

where the $\lceil (x) \rceil$ (*ceiling function*) rounds the value *x* up to the next integer.

Record Blocking and Spanned versus Unspanned Records

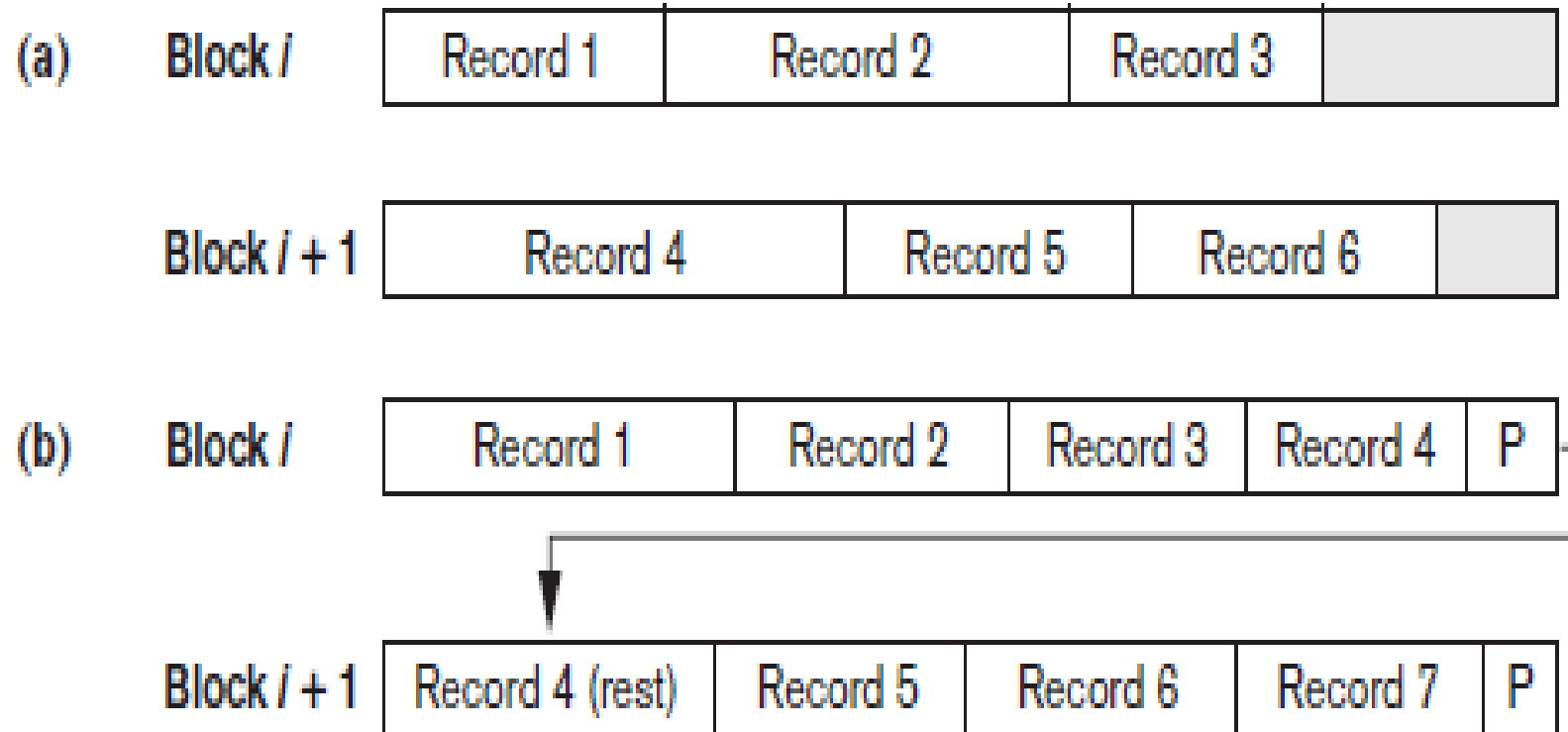


Fig (a) Unspanned Organization and Fig (b) Spanned Organization

File Organization Techniques

- The different types of file organizations are as follows :
 1. Heap file organization
 2. Sequential file organization
 3. Indexed—Sequential file organization
 4. Hashing or Direct file organization.

Heap File Organization

- In this file organization, the records are stored in the file, in the order in which they are inserted. All the new records are stored at the end of the file.
- This file organization is also called PILE FILE.
- This organization is generally used with additional access paths, like secondary indexes.
- Inserting a new record is very fast and efficient. But searching a record using any search condition involves a linear search through the file, which is comparatively more time consuming.
- Deleting a record from a heap file is not efficient as deletion of records resulted in wastage of storage space.
- It is generally used to store small files or in cases where data is difficult to organize.
- It is also used when data is collected at one place prior to processing.

Heap File Organization

- ***Advantages of Heap File Organization***

1. Insertion of new record is fast and efficient.
2. The filling factor of this file organization is 100%.
3. Space is fully utilized and conserved.

- ***Disadvantage of Heap File Organization***

- Searching and accessing of records is very slow.
- Deletion of many records result in wastage of space.
- Updation cost of data is comparatively high.
- It has limited applications.

Sequential File Organization

- In sequential file organization, records are stored in a sequential order according to the “search key”.
- A **Search key** is an attribute or a set of attributes which are used to serialize the records. It is not necessary that search key must be primary key.
- It is the simplest method of file organization.
- Sequential method is based on tape model.
- Devices who support sequential access are magnetic tapes, cassettes, card readers etc. Editors and compilers also use this approach to access files.

Sequential File Organization

- Structure of a sequential file is shown in Figure 3.
- The records are stored in sequential order one after another.
- To reach at the consecutive record from any record pointers are used.
- The Pointers are used for fast retrieval of records.
- To read any record from the file start searching from the very first record with the help of search key.
- Sequential file organization gives records in sorted form. This organization is used in small size files.

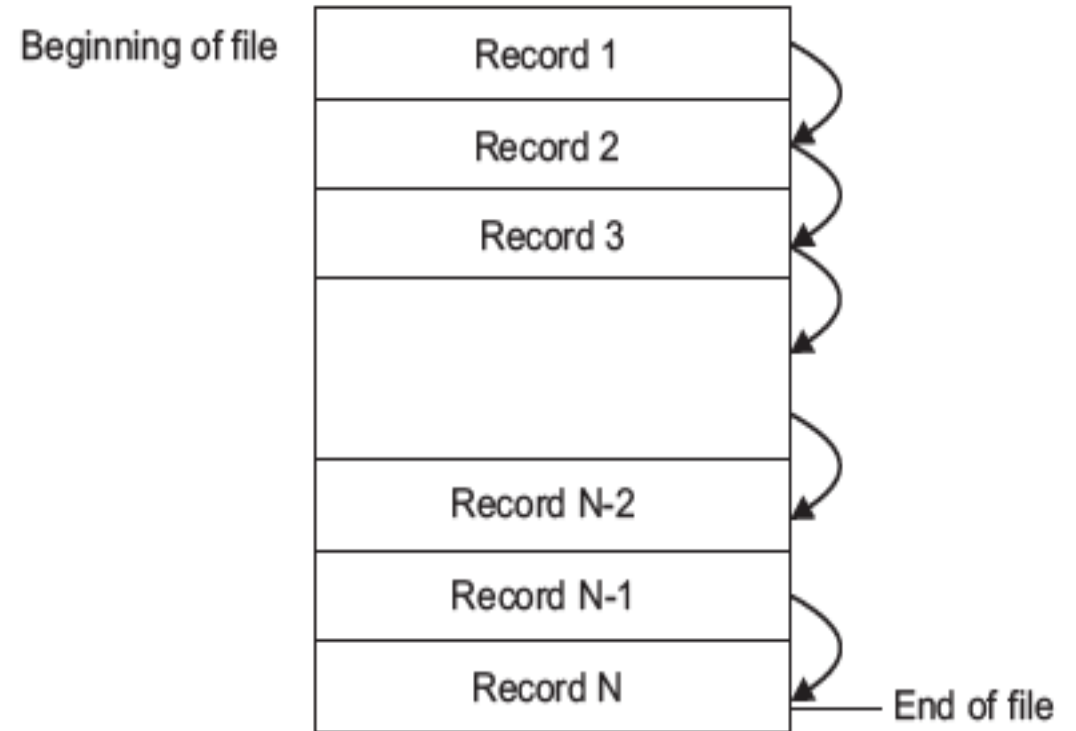


Figure 3

Sequential File Organization

- ***Advantages***

1. It is easy to understand.
2. Efficient file system for small size files.
3. Construction and reconstruction of files are much easier in comparison to other file systems.
4. Supports tape media, editors and compilers.
5. It contains sorted records.

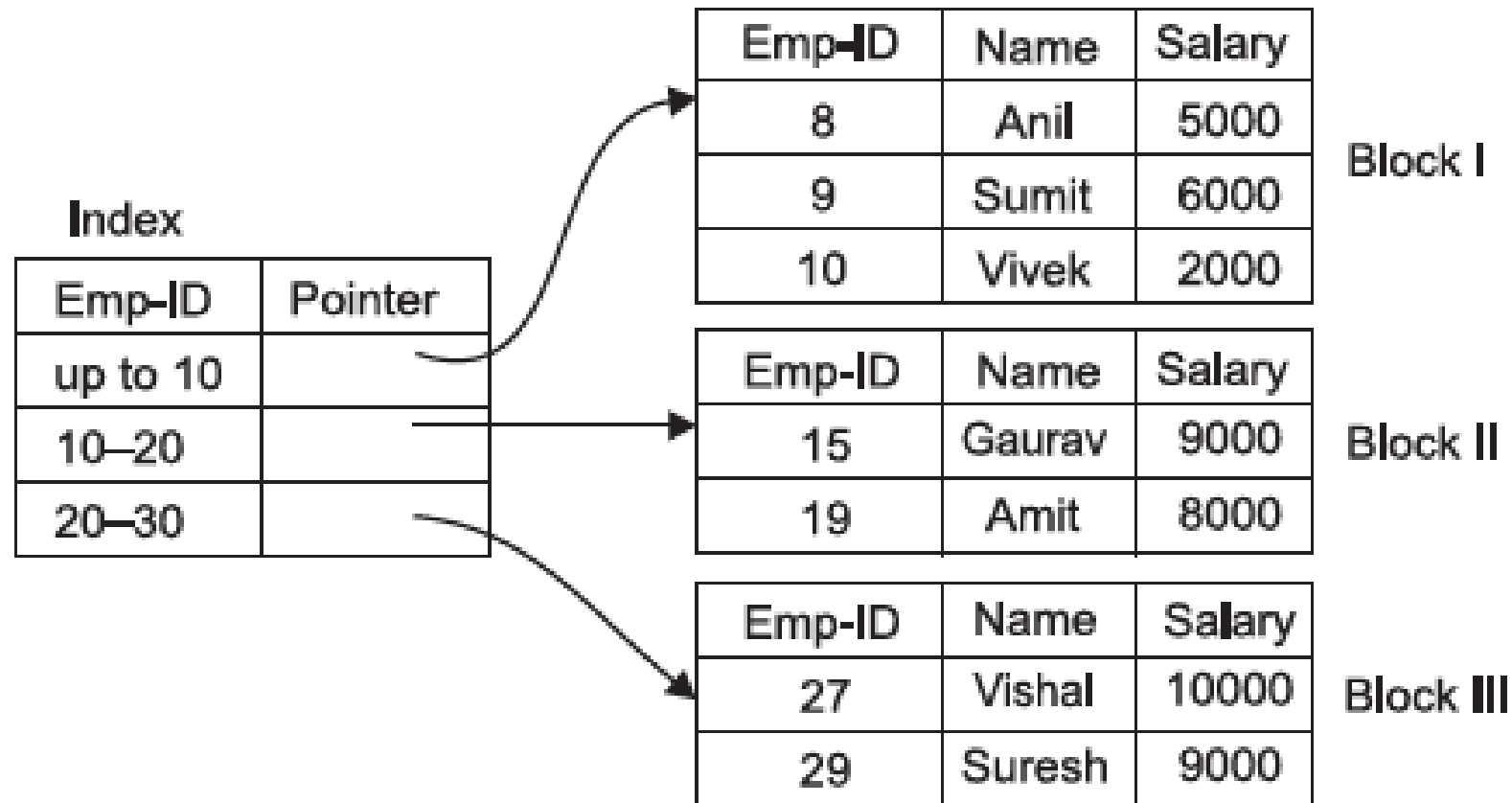
- ***Disadvantages***

1. Inefficient file system for medium and large size files.
2. Updations and maintenance are not easy.
3. Inefficient use of storage space because of fixed size blocks.
4. Before updations all transactions are stored sequentially.

Index Sequential File Organization

- Index sequential file organization is used to overcome the disadvantages of sequential file organization.
- It also preserves the advantages of sequential access.
- This organization enables fast searching of records with the use of index.
- Some basic terms associated with index sequential file organization are as follows :
 1. **Block** : Block is a unit of storage in which records are saved.
 2. **Index** : Index is a table with a search key by which block of a record can be find.
 3. **Pointer** : Pointer is a variable which points from index entry to starting address of block.
- To manipulate any record, search key of index is entered to find the starting address of block and then required record is searched sequentially within the block.

Index Sequential File Organization



Index Sequential File Organization

- ***Advantages***

1. Efficient file system for medium and large size files.
2. Easy to update.
3. Easy to maintain than direct files.
4. Efficient use of storage space.
5. Searching of records are fast.
6. Maintain advantages of sequential file system.

- ***Disadvantages***

1. Inefficient file system for small size files.
2. It is expensive method.
3. Typical structure than sequential files.
4. Indexes need additional storage space.
5. Performance degradation w.r.t. growth of files.

Direct File Organization

- To meet the requirement to access records randomly direct file organization is used.
- In direct file organization records can be stored anywhere in storage area but can be accessed directly, without any sequential searching.
- It overcomes the drawbacks of sequential, index sequential and B-trees file organization.
- For an efficient organization and direct access of individual record, some mapping or transformation procedure is needed that converts key field of a record into its physical storage location.
- Actually, direct file organization depends upon hashing that provides the base of mapping procedure.
- To overcome the drawbacks of hashing algorithm, collision resolution technique is needed.
- Devices that support direct access are CD's, Floppy etc.

Direct File Organization

- ***Advantages***

- Records are not needed to be sorted in order during addition.
- It gives fastest retrieval of records.
- It gives efficient use of memory.
- Operations on direct file are fast so there is no need to collect same type of operations in a file, as in sequential file system.
- Searching time depends upon mapping procedure not logarithm of the number of search keys as in B-trees.
- Supports fast storage devices.

- ***Disadvantages***

- Wastage of storage space (Clustering) if hashing algorithm is not chosen properly.
- It does not support sequential storage devices.
- Direct file system is complex and hence expensive.
- Extra overhead due to collision resolution techniques.

Difference between Sequential, Indexed and Direct File Organization

SNO	Sequential	Indexed	Direct
1	Random retrieval on primary key is impractical.	Random retrieval of primary key is moderately fast.	Random retrieval of primary key is very fast.
2	There is no wasted space for data.	No wasted space for data but there is extra space for index.	Extra space for addition and deletion of records.
3	Sequential retrieval on primary key is very fast.	Sequential retrieval on primary key is moderately fast.	Sequential retrieval of primary key is impractical.
4	Multiple key retrieval in sequential file organization is possible.	Multiple key retrieval is very fast with multiple indexes.	Multiple key retrieval is not possible.
5	Updating of records generally requires rewriting the file.	Updating of records requires maintenance of indexes.	Updating of records is the easiest one.
6	Addition of new records requires rewriting the file.	Addition of new records is easy and requires maintenance of indexes.	Addition of new records is very easy.
7	Deletion of records can create wasted space.	Deletion of records is easy if space can be allocated dynamically.	Deletion of records is very easy.

For Video lecture on this topic please subscribe to my youtube channel.

The link for my youtube channel is

https://www.youtube.com/channel/UCRWGtE76JITp1iim6aOTRuW?sub_confirmation=1