

Information Retrieval
ITIS401
Chapter-6
2021-2022
Dr Mohamed Abdeldaiem

The Physical Structure of Data

- Introduction to Physical Structures
- Record Structures and Their Effects
 - 1-Basic Structures
 - 2-Space-Time and Transaction Rate
- Basic Concepts of File Structure
- Organizational Methods
 - 1-Sequential Files
 - 2-Index-File Structures
 - 3-Lists
 - 4-Trees

Introduction to Physical Structures

- The physical organization of data in memory is a complex and highly technical subject (Batory and Gotlieb, 1982; Cardenas, 1985; Date, 1985; Frakes and Baeza-Yates, 1992; Korth and Silberschatz, 1986; Standish, 1980; Tremblay and Sorenson, 1985; Wiederhold, 1987).

Record Structures and Their Effects

- The main structural elements of a physical record are essentially the same as those of a virtual record: the individual data elements or attributes and their relationships.
- But at the physical level we have to know where the elements are in memory and what is the actual basis for their positioning.
- It is not enough merely to know that they exist or to know what assumptions we are permitted to make about them.

Basic Structures

- The simplest record structure consists of a set of attribute values, stored one after the other, each of a predetermined, fixed size in number of bits or bytes.
- A record, if stored on disk, will normally be read into RAM as a unit, or as part of a larger unit.

Record Structures and Their Effects

<i>Attribute</i>	<i>Number of Bytes</i>	<i>Content</i>
name	11	SMITH, JOHN
address	32	1287 MAPLE ST
telephone	10	8085551234
date_of_last_change	6	031228
name	14	STEINBERG, MAE
address	27	327 MAIN ST
telephone	10	8085554321
date_of_last_change	6	000116

<i>Attribute</i>	<i>Content</i>	<i>Length (bytes)</i>
name	SMITH, JOHN	10
account number	1234 567 890	12
no_transactions	003	3
Transactions		
date	990624	6
type	2	1
amount	000005.45	8
date	000113	6
type	2	1
amount	000237.00	8
date	000115	6
type	3	1
amount	000100.00	8

- **Variable-length record:**

In this form, each attribute value carries with it an explicit tag showing its length. The name and address attributes vary widely in length; date and telephone do not.

- **Variable-length record:**

Each field or attribute is of fixed length, but there can be a variable number of occurrences of the structure Transactions. The number of bytes for each field is explicitly given in the table and the number of transactions is given (no_transactions).

Space-Time and Transaction Rate

- It takes time to find the length attributes, interpret them, and access the location of the next element. The pointer or list approach is one of many examples of a data structure in which there is a trade-off between space and time.
- One facet of transaction rate is volatility, the rate at which a file or database changes. A file of stock market transactions or of positions of aircraft in an air traffic control system is going to change frequently and in both cases it is essential that the computer keep up with what can be very high volatility.

Basic Concepts of File Structure

- A primary consideration is the assumption that records will be stored on disks, or some auxiliary memory that is larger in capacity and slower in read/write speed than RAM.
- Access time is critical. We do not want the auxiliary memory to be slower, but will tolerate it because of the high cost of speed, and because with good organization and program design, we can do without some of the speed.

Organizational Methods

- Each of the methods described below has variations on how it is implemented by any given computer operating system.
- It may be difficult to learn exactly how a favorite retrieval or database system organizes records and it may not matter until a file grows very large and has a high level of activity

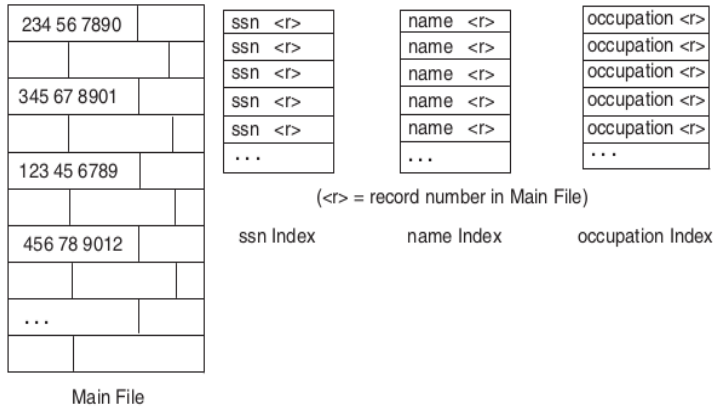
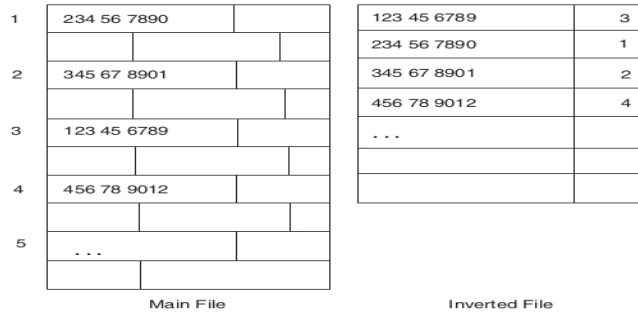
Sequential Files

- In a sequential file, records are stored contiguously and are normally in order based on a sort key. New records are added to a sequential file only by appending them to the end of the file.

Index-File Structures

- Within RAM, a record can be accessed directly if its location is known. With a disk memory, we can go directly to only the track or sector (portion of a track) that contains the record.
- One method of finding the location in terms of track or sector is to create an index , a separate file that tells where records are in the first file.
- For example, as shown in Fig-3, if we have a file, called the Main File, of rather large records, using ssn as a sort key, we might create a second file that consists only of ssn and the location of the corresponding record in the main file.

Index-File Structures



- Inverted file or index:

To find a record with a given key, look first in the inverted file and retrieve its record number. Then look in the Main File for that record. If the Inverted File can be stored in RAM, then finding a Main File record requires only one disk access.

- Use of multiple indexes: a separate inverted file or index can be created for as many attributes as desired. To search the main file on an attribute for which there is no index means a lengthy sequential search. If there is an index, using it to retrieve record number (r) means there need to be only one accession of a record in the main file.

Lists

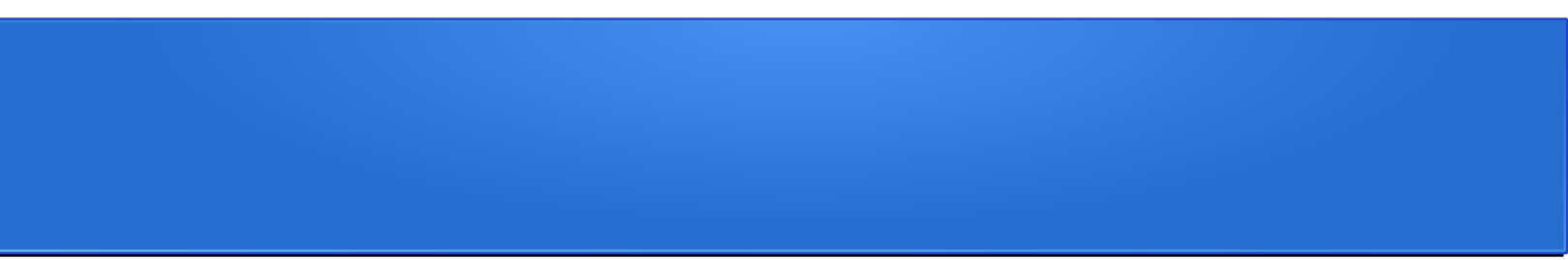
Record No.

1	234 56 7890	2
2	345 67 8901	4
3	123 45 6789	1
4	456 78 9012	18
5	...	

- One method of linking is to use pointers between records.
- Fig-5 shows a sequential file, to which, has been added a pointer to the next record as an attribute of each record. Next, in this case, means the record with the next higher value of the sort key.

Trees

- A tree structure is a set of records linked by two or more pointers. Each points to succeeding records whose keys are immediately “above” or “below” the current one.
- A directory points the search program to the location of the initial record, which is the one having the median key value, i.e., half the other keys



Information Retrieval
ITIS401
Chapter-6
Part-2



- Parsing of Data Elements

- 1-Phrase Parsing

- 2-Word Parsing

- 3-Word and Phrase Parsing

- Summary

Parsing of Data Elements

- One possible selection rule is to select every word in the text. This has the advantage of mechanical simplicity, but results in selection of many words that common sense suggests cannot make a meaningful contribution to a good index, hence fills memory with useless entries.
- There are two basic mechanical ways of parsing a syntactic expression to create an index, by word or by phrase. There can also be a combination of the two or, of course, no index at all.

Phrase Parsing

- Phrase parsing means to treat the entirety of an attribute value as a single phrase, or single entity for indexing purposes. The ssn is of this type, because the entire value of the attribute is used as the index element.
- An attribute like author is conventionally recognized as a phrase, consisting typically of last name, a comma, a space, first name, comma, space, and middle name.

Word Parsing

- This means to break up the content of an attribute value into its individual words, possibly deleting “stop” words. In this case, the original syntax may be lost within the index.
- This method is used with relatively large bodies of text, in which inclusion of the entirety in an index would be meaningless.

Word and Phrase Parsing

- Both word and phrase parsing might be used with such attributes as subject headings, job titles, or names of inventoried items. This allows those who know the correct phrase to find it quickly.

Figure-7: shows some examples of the methods.

- The form of parsing used to create indexes is of critical importance to the success of an IRS. Economically, an error in judgment can result in memory being allocated to the storage of useless data.

Terms extracted for the Inverted File

au=meadow, charles t.
 a.au=cochrane, pauline a.
 information/ti
 retrieval/ti
 [and] **
 [the] **
 human/ti
 retrieval/de
 condition/ti |
 information retrieval/de
 information/de
 retrieval/de
 impact of information retrieval/de
 impact/de
 [of] **
 information/de
 retrieval/de
 jn=information retrieval

Notes: ** These words are normally dropped by use of a stop list.

Terms as they will appear in the Inverted File

au=cochrane, pauline
 au=meadow, charles t.
 condition/ti
 effects/de
 human/ti
 impact/de
 impact of information
 information/de,ti
 information retrieval/de
 jn=information retrieval
 retrieval/de,ti

<u>Field</u>	<u>Content</u>	<u>Parsing</u>
AU	MEADOW, CHARLES T.; COCHRANE, PAULINE A.	PHRASE*
TI	INFORMATION RETRIEVAL AND THE HUMAN CONDITION	WORD
DE	INFORMATION RETRIEVAL; IMPACT OF INFORMATION RETRIEVAL	BOTH*
JN	INFORMATION RETRIEVAL	PHRASE

* Parsing recognizes subfield boundaries denoted by a semi-colon

Summary

For any file, we have a choice between placing records in random order (or arrival order) and putting them in order according to a key.

Random order means that the location of the record is not determined by the value of an attribute in the record. Such a method is fast at placing the record in memory, but may render searching impractically slow.

The file structures in use with most database software make use of combinations of the various methods we have surveyed. Largely, this is because different files and usage patterns place different demands on file structures. To avoid a different structure for each file, we tend to compromise on some aspects in order to achieve good performance on most of them.



End of Chapter-6

Any Question...?