2. FILE ORGANIZATION

The database is stored as a collection of files.

- Each file is a sequence of records.
- A record is a sequence of fields.

Classifications of records

- Fixed length record
- Variable length record

(i) Fixed length record approach:

Assume record size is fixed each file has records of one particular type only different files are used for different relations

- Simple approach
- Record access is simple Example pseudo code type

account = record

account_number char(10);

branch_name char(22);

balance numeric(8);

end

Total bytes 40 for a record

record 0	A-102	Perryridge	400
record 1	A-305	Round Hill	350
record 2	A-215	Mianus	700
record 3	A-101	Downtown	500
record 4	A-222	Redwood	700
record 5	A-201	Perryridge	900
record 6	A-217	Brighton	750
record 7	A-110	Downtown	600
record 8	A-218	Perryridge	700

Two problems

- Difficult to delete record from this structure.

- Some record will cross block boundaries, that is part of the record will be stored in one block and part in another. It would require two block accesses to read or write Reuse the free space alternatives:

– Move records i + 1, . . ., n to n i, . . . , n – 1

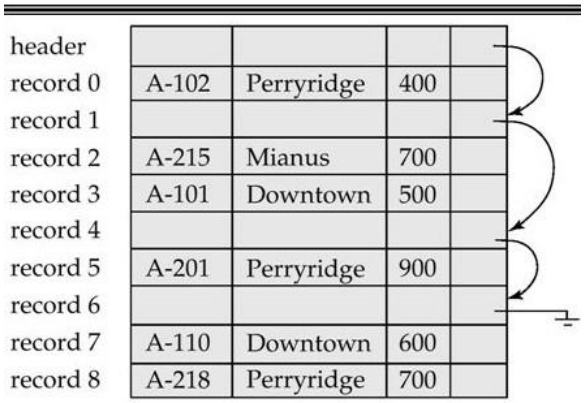
– do not move records, but link all free records on a free list

– Move the final record to deleted record place.

Free Lists

Store the address of the first deleted record in the file header.

Use this first record to store the address of the second deleted record, and so on



header				
record 0	A-102	Perryridge	400	
record 1				
record 2	A-215	Mianus	700	
record 3	A-101	Downtown	500	
record 4				
record 5	A-201	Perryridge	900	
record 6				
record 7	A-110	Downtown	600	
record 8	A-218	Perryridge	700	

Variable-Length Records

Byte string representation

Attach an end-of-record control character to the end of each record.

Difficulty with deletion

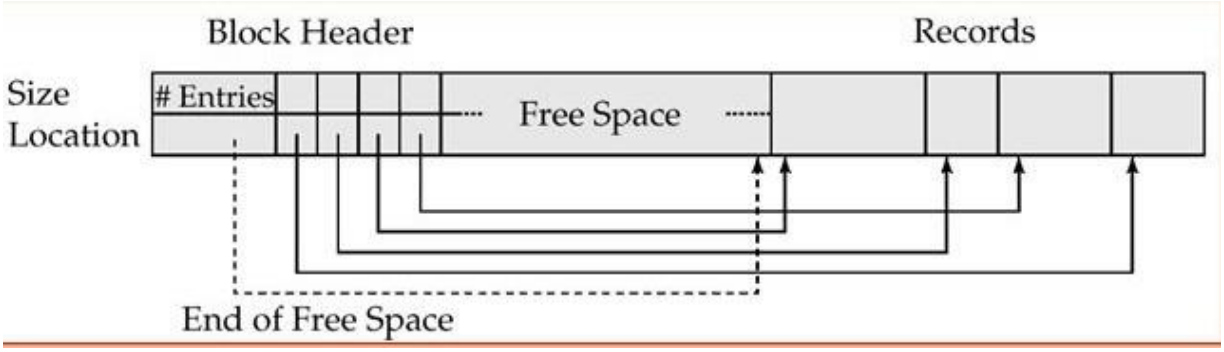
0	perryridge	A-102	400	A-201	900	⊥
1	roundhill	A-305	350	⊥		
2	mianus	A-215	700	⊥		

Disadvantage

It is not easy to reuse space occupied formerly by deleted record.

There is no space in general for records grows longer

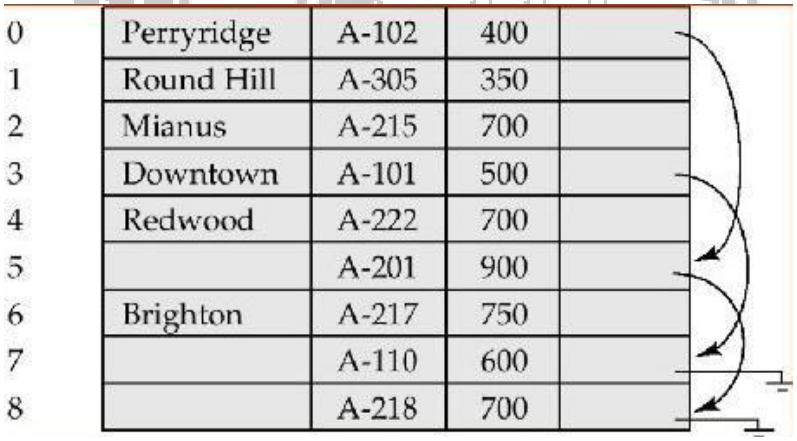
Slotted Page Structure



Slotted page header contains:

- number of record entries
- end of free space in the block
- location and size of each record

Pointer Method



0	Perryridge	A-102	400	
1	Round Hill	A-305	350	
2	Mianus	A-215	700	
3	Downtown	A-101	500	
4	Redwood	A-222	700	
5		A-201	900	
6	Brighton	A-217	750	
7		A-110	600	
8		A-218	700	

- A variable-length record is represented by a list of fixed-length records, chained together via pointers.
- Can be used even if the maximum record length is not known.

Disadvantage to pointer structure; space is wasted in all records except the first in a a chain.

Solution is to allow two kinds of block in file:

- Anchor block – contains the first records of chain
- Overflow block – contains records other than those that are the first records of chains.

Organization of Records in Files

• **Sequential** – store records in sequential order, based on the value of the search key of each record

• **Heap** – a record can be placed anywhere in the file where there is space

• **Hashing** – a hash function computed on some attribute of each record; the result specifies in which block of the file the record should be placed

1. **SEQUENTIAL FILE ORGANIZATION**

• Suitable for applications that require sequential processing of the entire file

• The records in the file are ordered by a search-key.

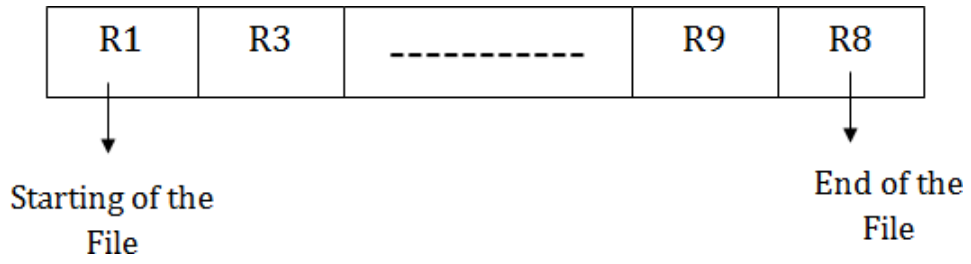
Deletion – use pointer chains

Insertion – locate the position where the record is to be inserted – if there is free space insert there – if no free space, insert the record in an overflow block – In either case, pointer chain must be updated

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

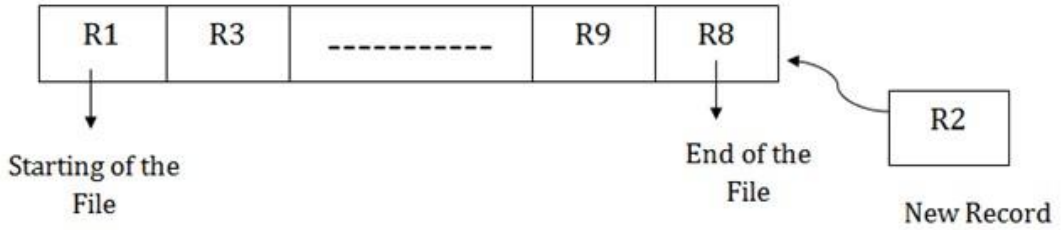
1. **Pile File Method:**

- 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.



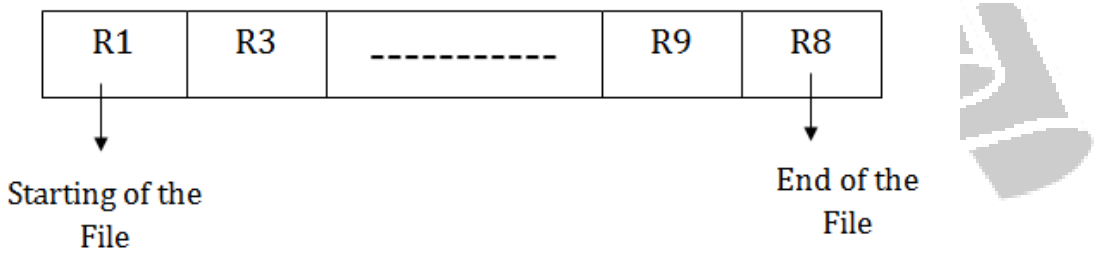
Insertion of the new record:

Suppose we have four records R1, R3 and so on upto R9 and R8 in a sequence. Hence, records are nothing but a row in the table. 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.



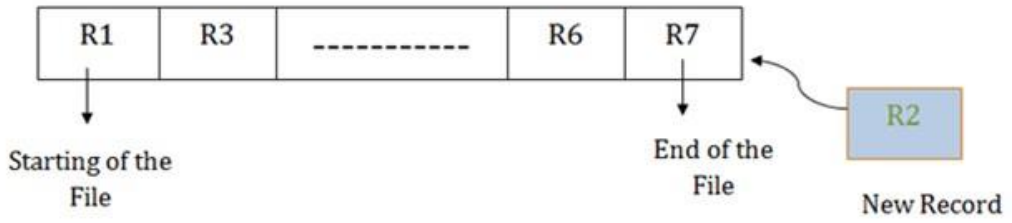
2. **Sorted File Method:**

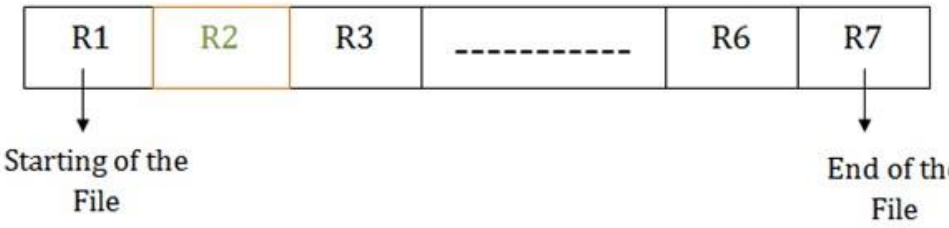
- In this method, the new record is always inserted at the file's end, and then it will sort the sequence in ascending or descending order. Sorting of records is based on any primary key or any other key.
- In the case of modification of any record, it will update the record and then sort the file, and lastly, the updated record is placed in the right place.



Insertion of the new record:

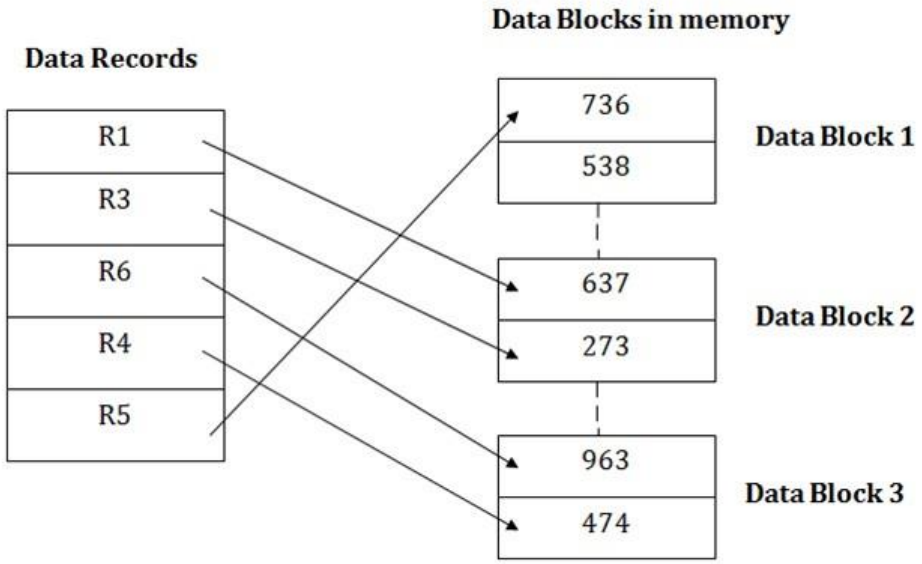
Suppose there is a preexisting sorted sequence of four records R1, R3 and so on upto R6 and R7. Suppose a new record R2 has to be inserted in the sequence, then it will be inserted at the end of the file, and then it will sort the sequence.





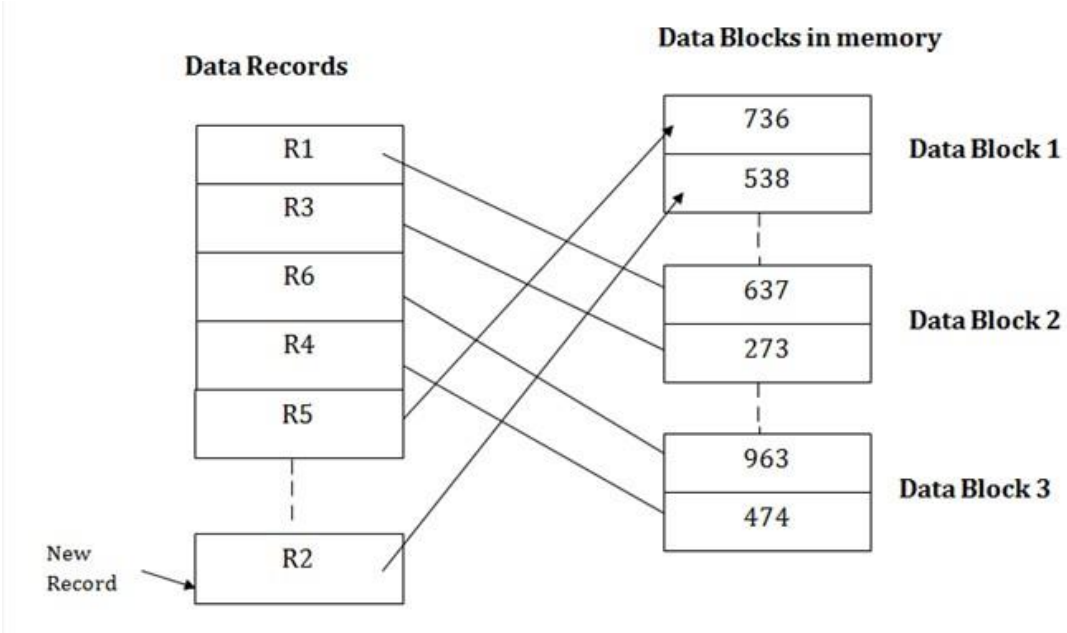
2. HEAP FILE ORGANIZATION

- 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.
- In the file, every record has a unique id, and every page in a file is of the same size. It is the DBMS responsibility to store and manage the new records.



Insertion of a new record

Suppose we have five records R1, R3, R6, R4 and R5 in a heap and suppose we want to insert a new record R2 in a heap. If the data block 3 is full then it will be inserted in any of the database selected by the DBMS, let's say data block 1.



If we want to search, update or delete the data in heap file organization, then we need to traverse the data from staring of the file till we get the requested record.

If the database is very large then searching, updating or deleting of record will be time-consuming because there is no sorting or ordering of records. In the heap file organization, we need to check all the data until we get the requested record.