# File Organization of Variable Length Records.

Lets take an example for Variable Length Records

```
type account-list = record
                    branch-name : char(22);
                    account-info : array[1....a]
                                   of record;
                    account-number : char(10);
                    balance : real;
                    end
end
```

In this we define account_info as an array with an arbitrary number of elements.

→ the type definition does not limit the number of elements in the array,

→ There is no limit on how large a record can be (up to of course the size of the disk storage!).

## → Byte-String Representation :- Approaches

1st approach End-of-record (⊥) Symbol :- Our first Approach to implement Variable length records is to attach a special end-of-record (⊥) symbol to the end of each record. So by this we can then store each record as a string of consecutive bytes.

2nd approach: Storing the record length at the beginning of each Record, instead of use of end-of-record Symbol.

## Disadvantages of 2 approaches:-

1) Not easy to reuse space occupied by deleted record. Small space of disk is wasted inorder to insert / delete operations.

| | | |
|---|---|---|
| A-892 | Rohl | 18001 |
| A-675 | PJ | 1900 |
| A-601 | Rohl | 18000 |
| A-985 | PJ | 15000 |
| A-663 | RaviB | 16000 |
| A-613 | Shk | 19000 |
| A-904 | Rohl | 17000 |
| A-501 | P.O.P | 16500 |
| A-898 | SKL | 15500 |

Fixed Length Based Record - file

| | | | | | | |
|---|---|---|---|---|---|---|
| Rohl | A-892 | 18001 | A-601 | 18000 | A-904 | 17000 |
| PJ | A-675 | 1900 | A-985 | 15000 | ⊥ | ⊥ |
| RaviB | A-663 | 16000 | ⊥ | | | |
| Shk | A-613 | 19000 | ⊥ | | | |
| POP | A-501 | 16500 | ⊥ | | | |
| SKL | A-898 | 15500 | ⊥ | | | |

→ Variable Length Representation of Record

2) → No space for records to grow longer:- if a variable-length record becomes longer, it must be moved.
→ Movement is costly if pointers to the record are stored elsewhere in the database.

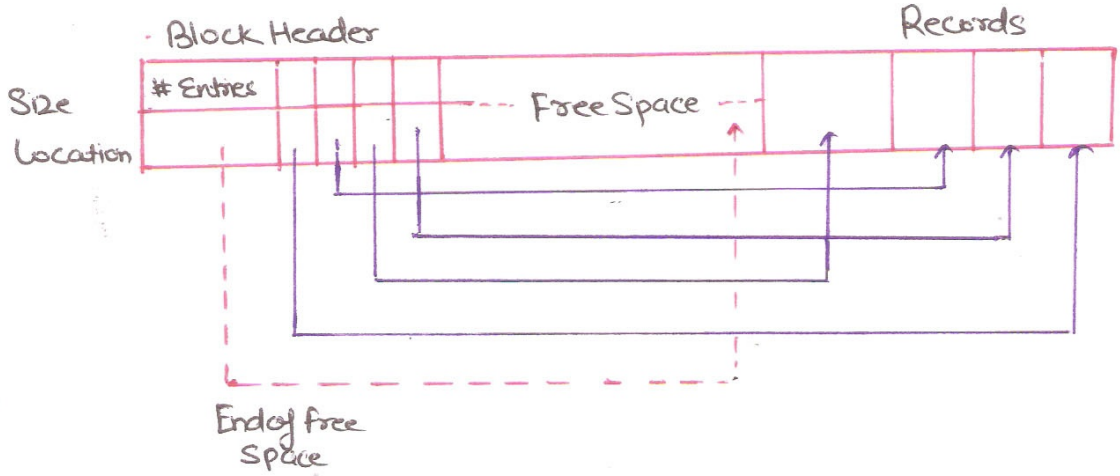
## Variable Length Records : Slotted page structure:-

A modified form of byte string representation, called the 'Slotted Page Structure' used for organizing records within a single block.

Slotted-Page Structure has a header at the beginning of each block, containing the following information

1. No. of Records entries
2. end of free space in the block
3. location and size of each Record.

## Slotted Page Structure




→ Records can be Moved Around within a page to keep them contiguous with no empty space between them; Entery in the header must be updated

→ Pointers should not point directly to records - instead they should point to the entry for the record in Header.

##

⇒ There is an another approach to implement Variable length records efficiently in a file System

→ By ~~doing~~ using one or more fixed Length records

We have two approaches of doing this

1) Reserved space : We can fixed a length of block to the know maximum length (It should not be exceeded) of the Variable-length record.

→ Unused space of shorter record filled with a null or end-of-record Symbol.

2) List Representation :-

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | Rohi | A-892 | 18001 | A-601 | 18000 | A-904 | 17000 |
| 1 | PJ | A-675 | 1900 | A-985 | 15000 | ⊥ | ⊥ |
| 2 | Rani.B | A-663 | 16000 | ⊥ | ⊥ | ⊥ | ⊥ |
| 3 | Shk | A-613 | 19000 | ⊥ | ⊥ | ⊥ | ⊥ |
| 4 | PoP | A-501 | 16500 | ⊥ | ⊥ | ⊥ | ⊥ |
| 5 | SKL | A898 | 15500 | ⊥ | ⊥ | ⊥ | ⊥ |

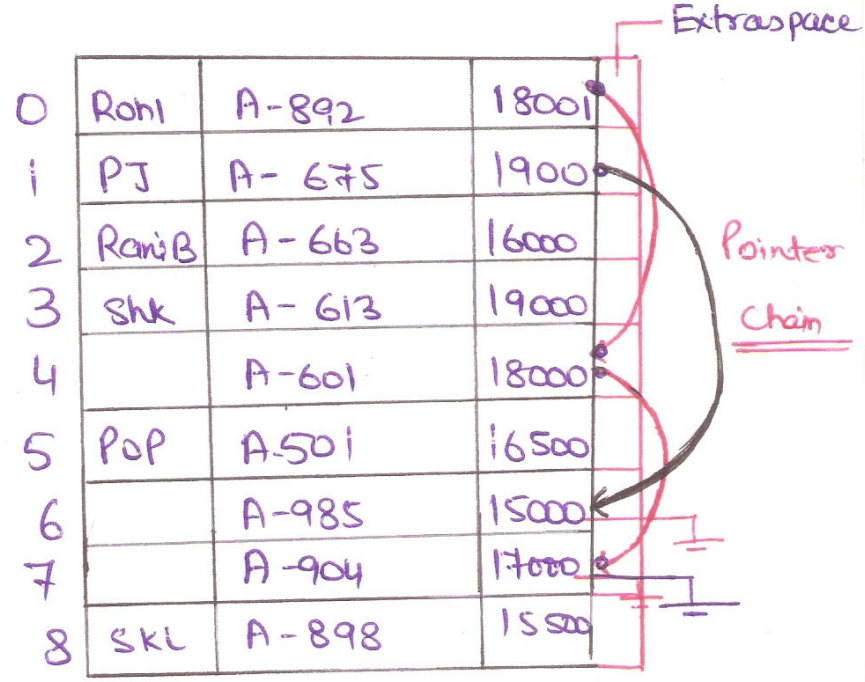
Reserved Space Approach

2) List Representation :-

→ 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 of this approach is wastage of space




| | | | |
|---|---|---|---|
| 0 | Rohi | A-892 | 18001 |
| 1 | PJ | A-675 | 1900 |
| 2 | Rani B | A-663 | 16000 |
| 3 | Shk | A-613 | 19000 |
| 4 | | A-601 | 18000 |
| 5 | PoP | A-501 | 16500 |
| 6 | | A-985 | 15000 |
| 7 | | A-904 | 17000 |
| 8 | SKL | A-898 | 15500 |

So overcome to this disadvantage, allowing 2 kinds of block in file.

## 2 blocks are Anchor block & Overflow block

**Anchor block** :- Which contains the first set of record of a chain

**Overflow block** :- Which Contains records other than those that are the first record of a chain

Anchor Block

| | | | |
|---|---|---|---|
| Rohl | A-892 | 18001 | |
| PJ | A-675 | 1900 | |
| RaviB | A-663 | 16000 | |
| Shk | A-613 | 19000 | |
| Pop | A-501 | 16500 | |
| SKL | A-898 | 15500 | |

Overflow Block

| | | |
|---|---|---|
| A-601 | 18000 | |
| A-985 | 15000 | |
| A-904 | 17000 | |