

# Addresses

## 1. dBase Addresses

A dBase file contains in front of the primary data the metadata, by which the length, the type, ... of the fields is described. The primary data records have fixed length. They are stored one after the other. The tenth record has the address ten. This address is used in the indices. It is no problem to compute from these numbers the physical address of the record, simply by multiplication with the record length, plus the length of metadata.

Advantages	Disadvantages
simple structure	Records cannot grow (fixed length)
record can be larger than a block	Storage area of deleted records cannot be reused
quick access	Insertions only at the end
	No repeating groups

## 2. Page Addresses

It is assumed that all records are smaller than a page (several physical blocks). In each page are written as many records as possible. The end of each page is free. Each record is addressed simply by its page number.

Advantages	Disadvantages
short addresses	false drop
by one page access several records are transferred	a record is smaller than a page
records can grow and shrink within pages without changing the address	
free storage area can be reused	

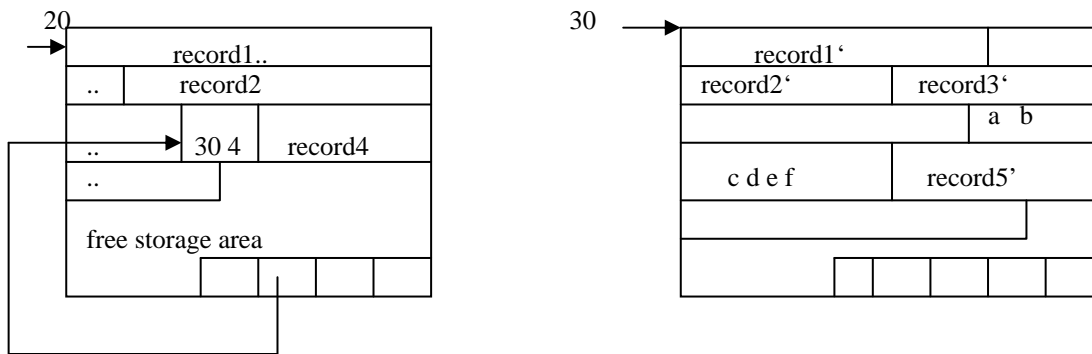
The false drop problem consists in the following:

If for a file indices for a field A and for a field B exist and we are looking for all records, which satisfy the condition  $A=a \wedge B=b$ , so it may be that as the first index as well the second index gives a page number  $pno$ , but this number results from different records. Then a page access to this page is necessary, without finding a record, which satisfies the conjunction.

### 3. TID's

The TID (tuple identifier)-concept refines the above page addressing concept. The false drop problem is solved by addition of a slot number to the page number. The slot number represents the number on which the record is within the page. Because of this, records can grow and shrink within a page without changing the address. If we delete a complete record, then a rudiment remains in the page foot such that the addresses of the remaining records do not change (no additional update of indices).

If a record grows and the free space in the page does not suffice, then the record is put in a page with enough free space. To avoid updating indices the new address of the record is written on its old position (following record are drawn back). In this case one additional access to the record is necessary, if the TID of the record is given. If the record has to be removed once more from the page, then it is “deleted” on its real position and the new address is written on its first position. Therefore also after more than 2 transformations of a record only 2 accesses to that record are necessary.



**Figure 1:** record4 with TID (20,4), record (a b c d e f) with TID (20,3)