will be lost. At first glance this looks somewhat improbable, but that lasts only until we encounter inverted files (see, Folk & Zoellick, 1992; and Horbron, 1988), which are the subject of the next section.

## 8.5 Inverted Files

The ideas underlying the creation of inverted files have been known since the 1960s. Even then this kind of file organization was successfully used in the Boolean IR system. Before analyzing inverted file organization, let us have another look at the illustration in Figure 8.3(a) of the record involved in a subject search. This record consists of the document's address (field 1) and a set of descriptors (field 2) obtained as a result of indexing this document. As was noted earlier, in a subject search such a record structure inevitably leads to a successive inspection of all records—something we would like to avoid. Thus, a need arose for some other kind of organization of data (record and file), and this other organization of data was supposed to secure not only a subject search but also to conduct it much faster; that is, it was to make direct access possible. But how can we do that in practice? What kind of file (record) must it be if it is to contain the same information? We will try to give a more detailed answer to these questions.

The record in Figure 8.3(a) contains an address and all the document descriptors. But what if we choose to do the opposite, placing a descriptor in the record together with all the addresses of documents containing this descriptor? In this case, the same information in represented in the file, but the new records are inversions of the old ones. This is actually the main idea of inversion in the IR system, and an inverted file consists exactly of such records. As we have not yet shown how a subject search can be organized in such a file, the positive effect of inversion is still unclear. For this reason we will consider in more detail the subject search process itself, looking first at a new "inverted" record. Figure 8.5 gives a general and simple description of such a record.

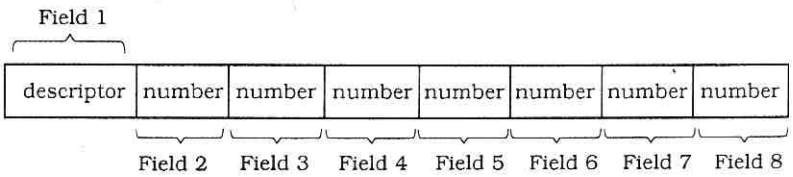


**Figure 8.5**
An example of a record in an inverted file.



It is clear from the figure that field 1 of this record contains a descriptor from the system's descriptor dictionary. All the other fields (from the second to the eighth) contain numbers of the documents where this descriptor appears. In other words, they are the documents containing at least one of the words included in the equivalence class of a given descriptor. The record, understandably, must have as many fields with numbers (addresses) as there are file documents indexed by the given descriptor. Our example contains seven such fields (from field 2 to field 8). Now it is clear that if you want to find documents indexed by the given descriptor it is enough to find only this record. The inverted file is composed of similar records. Earlier we gave an example of a certain IR system containing a search file with as many as 10,000 documents. Suppose that this same system has a descriptor dictionary containing as many as 800 descriptors (this size of a dictionary is realistic for many operating IR systems). Then the inverted file of this system might look as the one shown in Figure 8.6.

This example needs a detailed explanation. A1, A2, A3, . . . , I1, and I8 are descriptors from the system's descriptor dictionary. We see that every record corresponds to some descriptor written in the extreme left field. Only one record is organized for every descriptor in the file. Because we have assumed that the IR system contains a dictionary of 800 descriptors, it means that the file holds 800 records. The numbers written in all the other fields of the record are the numbers of the documents being introduced into the system upon their arrival (one after another). Because there are 10,000 documents in our example, the numbers, written in the other record fields, can only be within the range of 0 to 9999. It is clear from Figure 8.6 that the records could have different lengths

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| A 1 | 32 | 105 | 731 | 746 | 955 | 981 | 3426 | |
| A 2 | 503 | 1178 | 2177 | 4563 | | | | |
| A 3 | 2 | 17 | 5131 | 6978 | 7723 | | | |
| A 4 | 105 | 746 | 820 | 839 | 843 | 911 | 2562 | 3212 | 3600 | 4043 |
| A 5 | 208 | 319 | 503 | 724 | 981 | 3005 | 5131 | 4563 | 8119 | |
| ⋮ | | | | | | | | |
| I 7 | 342 | 361 | 694 | 1377 | 2274 | 2901 | 5624 | 7012 | | |
| I 8 | 78 | 746 | 3017 | 4261 | 5319 | 7717 | | | | |

**Figure 8.6**
An example of an inverted file.