# SELECTING A DATA BASE MANAGEMENT SYSTEM FOR INFORMATION RETRIEVAL ON A MICROCOMPUTER

Michael Nelson, Assistant Professor
School of Library and Information Science
University of Western Ontario
London, Ontario N6G 1H1

## ABSTRACT

Retrieval of bibliographic records has many characteristics which are not addressed by conventional data base management systems (DBMS). The capabilities of most DBMS packages on the market for microcomputers is contrasted with the needs for information retrieval such as variable length records and keyword searching. The limitations of microcomputers for information retrieval and some of the solutions which have been used is discussed. The influence of the relational model of data on DBMS is evaluated in terms of the applicability to bibliographic records and information retrieval. Examples are given using the dBASE II system on an Osborne I microcomputer.

# LE CHOIX D'UN SYSTEME DE GESTION DE BASES DE DONNEES POUR LE REPERAGE DE L'INFORMATION SUR UN MICROORDINATEUR.

## RESUME

Le repérage de dossiers bibliographiques possède plusieurs caractéristiques qui ne sont pas prises en considération par les systèmes conventionnels de gestion de bases de données (SGBD). L'applicabilité aux microordinateurs de la plupart des logiciels commerciaux de SGBD est étudiée en fonction des besoins particuliers du repérage de l'information, tels la longueur variable des dossiers et la recherche par mots-clés. On traite des limites des microordinateurs pour le repérage de l'information et des solutions proposées jusqu'à présent. L'influence du modèle relationnel des données sur les SGBD est évaluée en termes de son applicabilité aux dossiers bibliographiques et au repérage de l'information. On présente quelques exemples tirés du système dBASE II sur un microordinateur Osborne I.

# DBMS FOR INFORMATION RETRIEVAL

## INTRODUCTION

Microcomputers are invading every organization with more applications being invented every day. What is the potential of the microcomputer for bibliographic information retrieval? The two major problems which become evident are the storage capacity of small systems, and the appropriate software for information retrieval. For example, Clark (1982) reports the Microsearch system can only store 200 to 300 ERIC records on a 5.25" floppy disk. With the availability of hard disks of several megabytes of storage for a few thousand dollars has brought large storage within the reach of small systems.

The second problem is more serious; software to perform all the data manipulations, storage, and retrieval that is needed. The options are to write your own software, contract for special software, or buy a software package. Developing new sofware on a micro is usually a large undertaking which is often under-estimated, the small size of the computer doesn't necessarily make the task any easier. Two examples of this approach are described in Nightingale (1982) and Waldron and Cooke (1982). The advantages of buying a package are that it is immediately available and usually very much less expensive. One of the more popular developments has been data base management systems for microcomputers, designed as general problem solvers. Since DBMS software is widely available, inexpensive, and many people already have access to a DBMS, their potential for bibliographic retrieval needs to be evaluated. Finally, there is yet one other alternative, usually more expensive, which is to acquire specialized software designed for bibliographic records, such as Golden Retriever and others reviewed by Rorvig (1981).

## CHARACTERISTICS OF BIBLIOGRAPHIC DATA

When most suppliers and users of software think of data, they think of inventory, personnel records, mailing lists, accounting and scientific data which can be represented as numbers and short names or descriptions. Virtually all DBMS's are designed for this market and so have some disadvantages when it comes to bibliographic data.

Bibliographic data is characterized by a large number of fields, variable length fields, some fields which only occur rarely, and multiply occurring fields such as author or keyword. Access is not often by a simple unique key, or even by a single key on one field, but by many combinations of keys derived from the data such as author's last name, words from the title, truncated keywords, etc.

# DBMS FOR INFORMATION RETRIEVAL

## CHARACTERISTICS OF DBMS

To find a DBMS to process this type of data, a survey of DBMS packages by Barley and Driscoll (1981) provided the first summary of characteristics. Twenty packages were surveyed with with many features recorded in tables and commented on. Some of the features were: computer configuration needed, price (range $25 to $900), language written in, error recovery, levels of security, sort capabilities, printer interface, and facilities for defining data fields and entering data. In the area of field definition, only one system accomodated explicitly defined variable length fields, the FMS-80 system, also reviewed by Abbott (1982). The number of fields allowed varied from ten to 255. A common maximum number of records allowed is typically 64K, but this is usually limited more by disk space available for data than the limitations imposed by the software. Unfortunately neither the maximum record lengths nor the maximum field lengths allowed by each system were given in the survey. Some systems do not have maximum field lengths, but put a maximum on total record length, still using fixed fields.

Another area of interest is the indexing of the database which can be done by the system. In many cases these indexes are automatically maintained during record creation, deletion and update. Twelve of the packages allowed one field to be designated as a key, and only nine of these allowed multiple fields as keys. It was not clear from the tables given but the impression given is that 'multi key fields' means that several fields could be combined to produce a single key for the record. It must be emphasized once more that this means only one key per record, whereas in information retrieval we need several keys for each record.

In selecting a system for testing some of these capabilities, the relational approach (to be discussed later) was of particular interest, which narrowed the search considerably. After finding a good review by Abbott(1982a) the dBASE II system by Ashton-Tate Inc. was chosen. It was advertised as a relational system, it had a powerful built in structured programming language for building command files, it was fast being written in assembler, it ran under the CP/M operating system, and it had B-tree indexes which could be automatically created and maintained. The main disadvantage was the fixed length records with a maximum of one thousand bytes and a maximum of 254 bytes per field.

## THE RECORD STORAGE PROBLEM

There are two basic methods for overcoming the fixed length record problem, free field input with special characters for separators and a record with a directory giving the position of each field within the record. The maximum record length must still be set, and in fact in most sytems each record will use all this space. It is still a savings over giving maximums for each field in a fixed length field system.

The directory system of organizing records is the basis for the MARC record format. Extra space must be taken up by the directory, which gives a field code, field position and field length, and more complicated routines for packing and unpacking records are needed.

A more common technique used for simpler systems is to pack several variable length fields into one fixed length record, each separated by a special character such as a '$' in the following example:

```
Abbott, Jack L.$Database Management with Ashton Tate's
dBASE II$July 1982$BYTE$7$7$412-416$
```

This takes up less total space than the directory style, and is simpler to pack and unpack, especially when string manipulation functions are present.

Another solution to the record storage problem is to use the relational model of data for bibliographic records as explained by Crawford (1981). One advantage of this solution is that there is a large body of theory to draw on which promises consistency of access to information and data independence from the details of file organization. As will be seen later, most of the disadvantages come in the implementation of the basic concepts in an operational system.

## THE RELATIONAL MODEL

Consider a simple bibliographic record for a periodical article which consists of authors, title, journal, volume, number, date of publication, pages and keywords. This will be our beginning relation. One of the major problems in designing a database for such data is that both the author and keyword field may be repeated a different number of times for each record. The relational model separates such repeating groups into separate relations. In order to keep all the information from the original relation, there must be a unique key, the simplest being to assign a unique document number to each entry. We then have the following three relations:

```
DOCUMENT  <DOC#,TITLE,DATE,JOURNAL,VOLUME,NUMBER,PAGES>
AUTHOR    <DOC#,AUTHOR>
KEYWORD   <DOC#,KEYWORD>
```

Remember that each author for a document will have an entry in the authors relation, and each keyword an entry in the keyword relation. Thus our original relation is now in the first normal form, sometimes referred to as 'flat files' (see Crawford 1981). Incidentally, for such a simple example, the data base is also in fourth normal form.

How can this be implemented in dBASE II? What can be done is to create three separate database files with the fields corresponding to the three relations above. Unfortunately, the linkage between relations is not automatic, so even for data entry, a special data entry program must be written to process the inputs and enter the data into the appropriate relation with the correct key unless this is done manually before data entry. This also applies to displaying all the information for one bibliographic record. There must be a program written which searches each relation to assemble all the information for one document for display. In a relational DBMS this is generally not easy as the display corresponds to an unnormalized relation, so special processing is needed (see Crawford, 1981, p63).

## Searching in dBASE II

Using our relational model described in the previous section, a single author search can easily be done be scanning the AUTHOR relation and a single keyword search by scanning the KEYWORD relation. To make these searches more efficient each of these relations could be indexed, which can be done with one command in dBASE II. Of course the index uses at least as much disk space as the original relation. Note that once the AUTHOR and KEYWORD relations have been indexed, they are essentially inverted files. They occupy more disk space than the traditional inverted files because each time an author or keyword occurs in the data base, the document number - author pair is entered again into the relation. Normally, in a simple inverted file system the author or keyword is entered once into the file followed by a list of document numbers:

| AUTHOR relation | | Inverted File | |
|---|---|---|---|
| Abbott | 1 | Abbott | 1,2 |
| Abbott | 2 | Barley | 3 |
| Barley | 3 | Blair | 4 |
| Blair | 4 | ... | |
| ... | | | |

If Boolean combinations of authors or keywords is need using the relational approach in dBASE II there are even more difficulties. Since there are none of the standard relational operators of restrict, project or join, all query processing must programmed using the built in programming facility. The basic DISPLAY command in dBASE II does provide for Boolean combinations of conditions on the fields of one data base (i.e. relation). This actually leads us to the next solution to the information retrieval problem.

## A SIMPLE SOLUTION

To allow maximum flexibility of access to records using Boolean combinations of authors, keywords, words from title, or strings of characters from any field, the easiest solution to implement in dBASE II is to enter the records in free format and to use the excellent string functions and sequential search facilities. In this method the record is treated as one long string of characters for searching. In fact the field separators can be eliminated and the record is then treated as free text for searching purposes. The ability to specify searches within fields and to format the display of records depending on the fields is lost of course.

Both the relational system with the three relations DOCUMENT, AUTHOR and KEYWORD, and the simple free format records were implemented in dBASE II using an Osborne I microcomputer which has two 5.25" diskettes with a capacity of 92K each. Ninety-eight journal articles were used as the data base. For the relational implementation using fixed length fields in the relations, the DOCUMENT relation used 34K bytes, the AUTHOR relation 13K bytes and the KEYWORD relation 8K bytes for a total of 50K bytes, or more than half a diskette. If the relations are to be indexed even more space is needed.

The free format approach used 36K with 354 byte records. These statistics are influenced by the design of each relation or file, of course.

Using these data bases a test was done of the searching capabilities, which were programmed for the relational version and which used the built in single DISPLAY command for the free format version. Only a simple author search and a simple keyword search were implemented for the relational version. The query in this case was to display all documents with a particular author or a particular keyword. This was very fast only taking three or four seconds, and would not increase substantialy for a data base of a thousand records. Using the DISPLAY command with string search functions on the free format data base produced search times of twenty to twenty-four seconds. Since this is a sequential search this time will increase linearly with the number of records. This is four or five records per second and is mainly limited by the disk speed in this case. It also works out to about fifty to sixty seconds search time for the 250 records that fit onto one disk. The search time only increases slightly for more complicated queries, which could not be searched on the relational system without more programming.

## SUMMARY

For a small bibliographic data base of one or two thousand records maximum, typical of a personal research collection, the simple solution using a DBMS can work with a minimum of implementation problems. Some of the potential problems such as slow retrieval times for a large data base, vocabulary control and consistent updating are a trade-off for the efficient use of storage space and built in capabilities of the DBMS. Although the relational approach has many theoretical advantages, the actual implementation of these concepts has been minimal, particularly for microcomputers.

## REFERENCES

ABBOTT, Jack L. "Database Management with Ashton Tate's dBASE II", in BYTE. Vol. 7, No. 7 (July 1982), pp.412-416.

_______. "Systems Plus: FMS-80", in BYTE. Vol. 7, No. 10 (October 1982), pp. 447-450.

BARLEY, Kathryn S. and DRISCOLL, James R. "A Survey of Data-Base Management Systems for Microcomputers", in BYTE. Vol. 6, No. 11 (November 1981), pp.208-234.

BLAIR, John C. Jr. "Creating Your Own Database", in Database. Vol. 5, No. 3 (August 1982), pp.11-17.

CRAWFORD, Robert G. "The Relational Model in Information Retrieval", in Journal of the American Society for Information Science. Vol. 32, No. 1 (January 1981), pp. 51-64

CLARK, W. Bruce. "Microsearch-- A Project to extend the ERIC Database to Microcomputer", in Proceedings of the 45th ASIS Annual Meeting Vol. 19 (1982), pp.60-62.

NIGHTINGALE, R. A. "The Use of a Microcomputer for Information Retrieval and other Purposes in the Engineering Departments of BP International Ltd.", in Journal of Information Science Vol. 4, No. 4 (July 1982), pp.149-154.

RATLIFF, Wayne. dBASE II User Manual. Culver City, CA, Ashton-Tate.

RORVIG, Mark E. Microcomputers and Libraries: A Guide to Technology, Products and Applications. White Plains, New York, Knowledge Industry Publications, 1981. 135 p.

WALDRON, C. B. and COOKE, Deborah M. "A Personal On-line Reference Retrieval Program for Microcomputers", in Journal of Information Science Vol. 4, No. 4 (July 1982), pp. 155-160.

WILLIAMS, Philip W. and GOLDSMITH, Gerry. "Information Retrieval on Mini- and Microcomputers", in Annual Review of Information Science and Technology. Vol. 16 (1981), edited by Martha E. Williams, pp. 85-111.