

JA 582  
430

# A microcomputer-based book acquisition system in India using dBASE II

L.J. HARAVU, P.S. JADHAV and R. SREERAMANA

## ABSTRACT

This paper describes the general and functional criteria used in designing the book acquisition system at the International Crops Research Institute for the Semi-Arid Tropics in Hyderabad, India (ICRISAT). The schema, relations, and attributes of the systems' database are given. The capabilities of the system are described, and advantages of using dBase II as the applications language are outlined.

## 1. Introduction

Interest in the use of database management systems (DBMS) packages for library and information retrieval applications is recent. Wagner<sup>1</sup> provides an insight into the use of DBMS for librarians and information scientists. Macleod<sup>2</sup> describes the use of a DBMS for information retrieval on a mainframe computer. A research report of the British Library by Trevelyan and Rowat<sup>3</sup> has studied the feasibility of using systems software, including DBMS packages, for library applications on microcomputers. Crawford<sup>4</sup> provides the theoretical basis of the relational data model and the suitability of viewing information retrieval in terms of the relational model. The availability of dBase II, a DBMS based on the relational data model, on a wide range of microcomputers, and the increasing popularity of this package, have led to the development of library applications using dBase II as the applications language<sup>5,6</sup>. When a number of related files have to be managed, a DBMS provides several features not only for creating, updating, and searching the database files, but also for minimising redundant storage across files. Further, the DBMS insulates application programs from changes when alterations take place in the structure of the database records, or when new outputs or reports need to be prepared to meet a new requirement.

The book acquisitions function in a library is an example where several logically related files, e.g. the on-order file, vendor address file, and fund file have to be created, updated, maintained, and searched. This paper describes a book acquisition system developed using dBase II as the applications language. The acquisition system has been implemented on the ICRISAT library's Rainbow 100+ microcomputer system consisting of 384 Kbyte of RAM, dual 5-1/4" floppy disk drives, (400 Kbyte each), a 10 Mbyte Winchester hard disk drive, monitor, keyboard, and letter quality printer. Version 2.41 of dBase II has been used in this application, and was acquired for \$500.

## **2. Design criteria**

The following broad criteria were used in designing the acquisition system.

### **2.1 General criteria**

The system should be user-friendly, i.e. it should be capable of being used by library professionals and/or clerical personnel with little or no knowledge of computers. Specifically, the system should permit:

- data-entry and data updating through easily understood and well laid-out screen forms;
- menu-directed operations at all possible user interfaces – in other words, the system should provide menus and sub-menus, from which the user can select the desired functions/sub-functions;
- re-entry of data to be minimal in update operations through full-screen editing functions;
- easy recovery from errors;
- validation of input data, wherever possible, to ensure correctness of entered data.

In addition to user-friendly features, it was decided that the system should provide:

- data security through password features for access to database files, and for the conduct of certain operations on the database;
- ease of maintenance of the systems' programs through the use of structured programming concepts and principles;
- fast response in data-entry, update, and query operations.

### **2.2 Functional criteria**

It was considered important that the automated system should perform all the functions of a typical manual acquisitions system, for instance, checking for duplicates between new items and those already in the on-order file, a standard operation in manual systems. Other examples are informing requesters of the receipt of books ordered for them, and tasks in the payment of bills (verifying if the book ordered is what is actually received, correctness of currency conversion rates, and totals in invoices, etc.)

In addition to typical functions in a manual system it was decided that the automated system should enable:

- validation of input data using disc-held look-up files (e.g. a file of valid department or fund codes), and logical or computational criteria possible to be applied to some of the fields:

- querying of the database from several keys or fields, not normally easy or possible in manual systems;
- production of management information reports (e.g. vendor performance reports);
- alerting of users to exceptional conditions;
- efficient follow-up of items on-order beyond a given time;
- production of a variety of routine reports not normally produced in manual systems because of the labour-intensive nature of such jobs, and difficulties in rearranging data in manual systems;
- integration with budgeting and expenditure reporting;
- quick production of one-off or *ad hoc* reports that may be required, but cannot be pre-determined.

It is possible to develop systems to meet these criteria using languages such as BASIC, FORTRAN, COBOL, etc. However, it was decided to use dBase II in view of several advantages discussed in this paper.

### 3. Schema

The Acquisition System's database has the following relations and attributes:

#### (i) ONORDER RELATION

Purchase Order No.	Request for Purchase no.	Title	Edn	Year	ISBN	Publisher	Requester Name	Fund Code	List Price
--------------------	--------------------------	-------	-----	------	------	-----------	----------------	-----------	------------

Cost Price	Vendor Code	Date Ordered	Date Received	Check no.	Check Date	Vendor Report Code	Group Code	Accn no.
------------	-------------	--------------	---------------	-----------	------------	--------------------	------------	----------

#### (ii) AUTHOR RELATION

Purchase Order no.	Author
--------------------	--------

#### (iii) FUND RELATION

Fund Code	Fund Name	Allocation	Commitment	Expenditure	Balance
-----------	-----------	------------	------------	-------------	---------

#### (iv) VENDOR RELATION

Vendor Code	Vendor Name	Address	City	PIN or ZIP Code
-------------	-------------	---------	------	-----------------

(v) ORDERNO RELATION

Group Code	Last Serial no.
------------	-----------------

The first attribute in each of the above relations is the primary key of the relation.

The conceptual schema or view linking the above relations is shown in Figure 1.

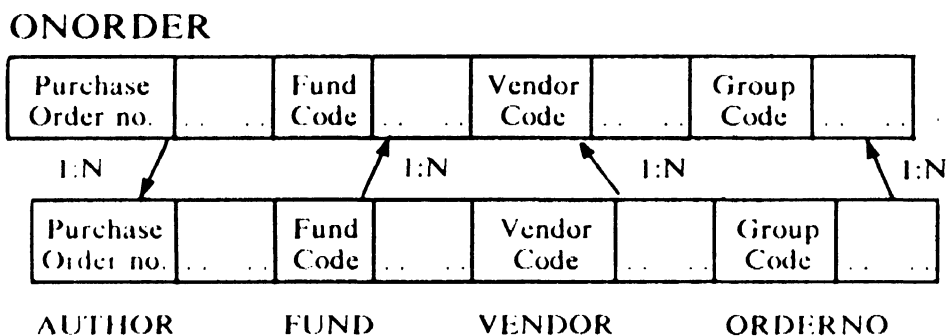


Figure 1. Conceptual schema of relations

The explicit recognition of 1:n relationships between two relations in the Relational Model reduces redundancy, e.g. the need to provide fixed storage space in every record for different values of a repeating field, irrespective of the number of values that the field may actually take in a record. Repeated storage of the same data across records is also avoided, eliminating wasted disc storage.

The ORDERNO relation shown above is used to generate a unique purchase order number for each input item. The purchase order number is made up of a group code and a running serial number. The group code is a three-digit broad subject code for subjects of interest to ICRISAT. Each time a book on a given subject is ordered, the last serial number is automatically increased by one. This ensures that each item ordered has a unique order number. Also, it is possible to know how many books in a given broad subject have been ordered.

The ORDERNO relation and the FUND relation are also used to validate input group codes and fund codes when a new record enters the system. These two files thus serve also as look-up files or as tables of valid group and fund codes.

Each of the five relations shown above is a Data Base File (DBF) in dBaseII parlance. Each of these DBFs is indexed by one or more attributes or keys as shown in Figure 2.

DBF	Indexed on
ONORDER	Purchase Order no, Fund Code,
AUTHOR	Purchase Order no, Author
FUND	Fund Code
VENDOR	Vendor Code
ORDERNO	Group Code

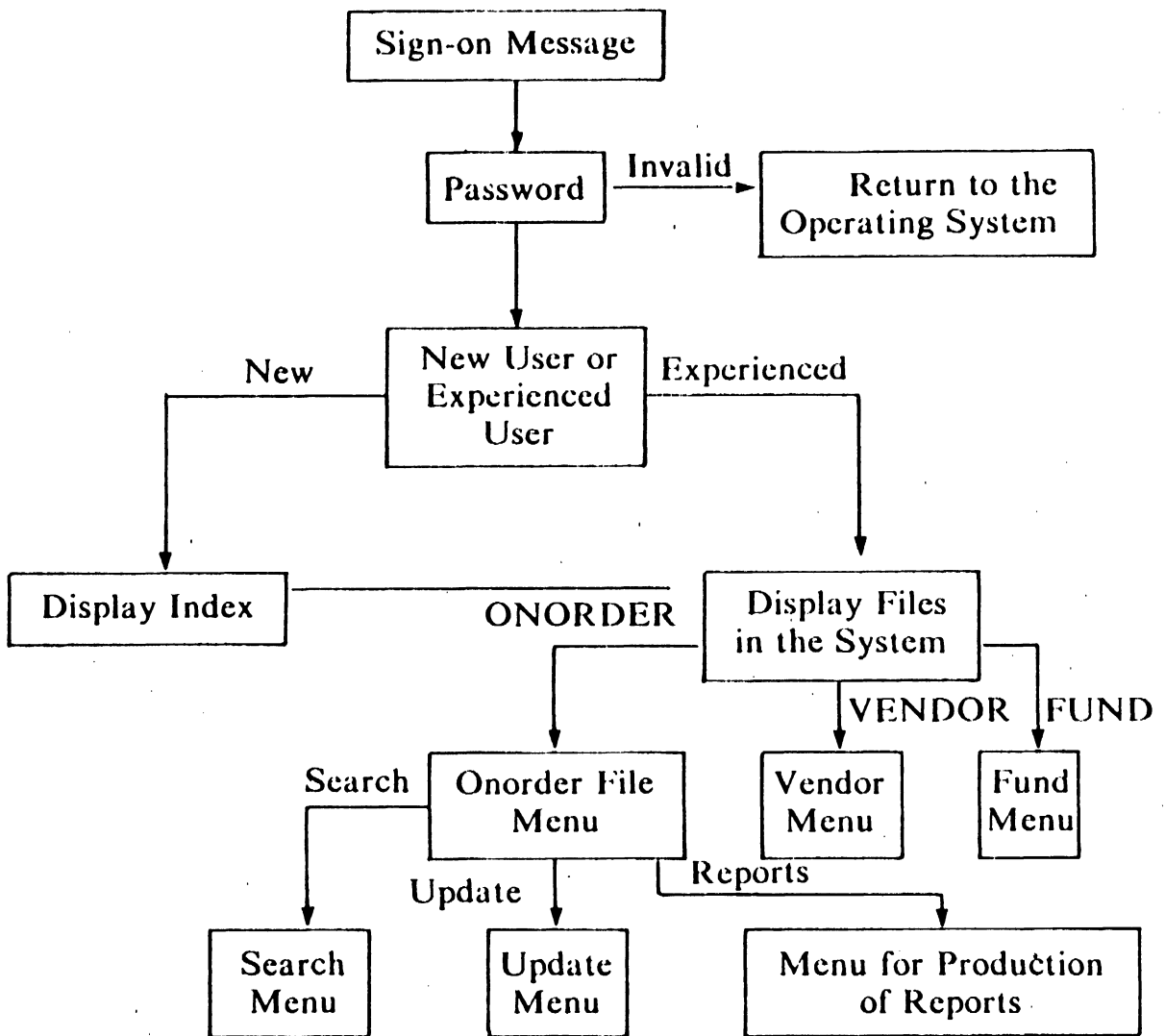
**Figure 2. Keys for the files**

The purpose of indexing a DBF by one or more keys is to enable random access to records of the DBF for any given value of the indexed key. An advantage of dBase II is that, when a DBF and its associated index files are opened, data entry, updating, and delete operations on that DBF, automatically update the index files. However, the use of many index files with a DBF slows down data entry, update, and delete operations on that DBF. It is useful to limit indexing only to those keys that need to be points of access to records of the DBF in the different processing programs. It is also possible in dBase II to search a DBF by non-indexed keys. Such searches are understandably slower than indexed key searches.

In the application developed, some index files are created only temporarily to facilitate production of a specific output product. For instance, in order to produce vendor performance reports, the ONORDER DBF is indexed by vendor code. The index file is deleted as soon as performance reports are produced. The ease and speed with which an index file can be created and updated, or deleted in dBase II without extensive programming effort is a useful feature in application development.

#### **4. The application**

Following a sign-on message and the validation of a password, the system asks whether the user is new or experienced. A new user is presented with a screen display of an index of functions possible in the system, and the route required to perform a given function (e.g. printing of purchase orders). An experienced user is presented with the main menu of the system which displays the names of the three files (ONORDER, VENDOR, and FUND) with which the user can work. Depending on the file chosen, the user is presented with a second menu containing a list of first-level functions, such as add new records, print reports, search, etc. Depending on the option chosen in the second menu, a specific function may be performed, or, where appropriate, a second-level menu may be displayed. For instance, if users choose to print reports in the first-level menu, they are directed to a second-level menu which lists the different reports possible to be produced. A schematic of the user interface is given in Figure 3.



**Figure 3. Schematic of user interface**

The following functions are possible in the system:

- Add new records to the ONORDER file
- Print Requests-for-Purchase
- Print purchase orders
- Receive books ordered
- Update the ONORDER file
- Create/Update the FUND file
- Print Fund Account Report
- Create/Update the VENDOR file
- Search the ONORDER file
- Follow-up on pending orders
- Clean-up the ONORDER file
- Print Vendor Performance Reports
- Print Fund Activity Reports

- Process advance payments
- Print 5" × 3" order cards
- Print the entire ONORDER file
- Print the Accession Register
- Cancel/Cancel and reorder books
- Leave the system

As a rule, all dialogue with the user is carried out on the last line of the screen, allowing the rest of the screen to be used for the screen form or menu. Error messages are also shown in fixed positions of the screen, and give instructions, wherever possible, to enable the user to recover from the error.

#### **4.1 Validation criteria**

The following validations are carried out during data-entry and data-updating operations on the ONORDER file:

- One of two prices, list price or cost price, is expected as input. The list price is the encumbrance price (or catalogue price) and is used to commit the amount to a fund. The cost price is the amount paid in advance or the actual cost paid to the vendor. This is added to the expenditure figures for a given fund. If both prices are input, the system rejects the entry and prompts the user to re-enter only one of the two prices.
- The name of the person requesting the item is expected as input. If this field is left blank at data-entry, or if it is blanked during data-updating, the entry or update is withheld until this data item is entered.
- The input or updated fund code is validated using the FUND file.
- The input or updated group code is validated using the ORDERNO file.
- The input ISBN is validated using Modulo 11 check digit computation.

Validation of new items input to the ONORDER file is done only after the new item is checked for duplication with those already in the ONORDER file. Duplicate checking is done by surnames of authors. The surname of the author of the input item is searched for in the AUTHOR file. If a match occurs the details of the book by the author already on the ONORDER file is displayed. The user is then prompted, either to continue with adding the new item or to suspend further input of data for the new item. If more than one record exists for a given author, all records are displayed one after the other until the user decides not to continue with the order, or until no more ONORDER records match the author being input.

#### **4.2 Search features**

The system allows search of the ONORDER file by the following keys:

- Author
- Author/Title
- Title
- Purchase Order no.
- Fund Code
- Vendor Code
- Requester Name

The results of the search are displayed using a screen form. Some search keys, e.g. Purchase Order no., retrieve just one record, since key values are unique. However, for some keys a search may retrieve more than one record, e.g. search by a given author. In such a case, the system displays a record and awaits response before continuing with the display of the next record, and so on until no more records match the search key.

The search by Fund Code, Vendor Code, and Requester Name is different from that by other keys. Here users are asked if they are interested in searching for books that are still on order, or for books that have already been received. The search is further qualified by choice of an earlier and a later order date. All books ordered between these dates and still on order, or already received, are then displayed. The output can also be directed to the printer, if necessary.

Searches such as the one by requester name or fund code enable the production of reports often demanded of the library, but difficult to produce in a manual system because of the need to rearrange the on-order file, and the time required to produce the report.

#### **4.3 Management information reports**

The system provides for three routine reports: a fund-activity report; vendor-performance reports; and a fund-account report.

The fund-activity report is a table showing the number of books ordered for different funds, the number of books received for the different funds, and the amount spent for each of the funds during a given period.

Vendor-performance reports can be produced for some or all vendors in the VENDOR file. The report for a given vendor shows the number of books supplied by the vendor in 1 to 15 days, 16 to 30 days, 31 to 45 days, and so on, and the total number of books ordered with the vendor and those actually supplied during a given period. A fund-account report shows the allocations, commitments, expenditures, and balance amounts for each fund.

The FUND file is automatically updated when a new item is input to the ONORDER file. The appropriate FUND record is accessed, and the commitment or expenditure figures are updated.



#### **4.4 Exception condition reporting**

When a book is received by the system, its cost as given in the vendor's invoice is compared with its list price. If there is a variation of 30 per cent or more between the invoice cost and list price, the user is alerted to this fact, and is given the option to suspend action on receiving the book, pending verification of the price.

When a book paid for in advance is received, the system reports this to the Fiscal Department so that the advance payment can be adjusted.

#### **4.5 Follow-up on pending orders**

One of the most difficult operations in a manual system is ensuring that the below mentioned follow-up actions are taken routinely:

- periodical reminders to vendors about books not yet supplied
- incorporation of vendor reports (e.g. out-of-print, publication mailed, etc.) into the file, and use of this information in follow-up with vendors and/or requesters
- cancellation and re-ordering of books based on vendor reports
- monitoring of receipts against advance payments made.

In the system developed, these follow-up actions are performed using one or more of the functions of which the system is capable.

#### **4.6 Printed outputs of the system**

The acquisition system provides the following printed outputs depending on the function invoked:

- Requests-for-Purchase
- Purchase Orders
- Intimations to requesters of books received
- Routing slips to the Fiscal Department for payment of invoices for books received
- Routing slips to the Fiscal Department on receipt of book for which advance payment was made
- Address lists of vendors in the VENDOR file
- Fund Account Report
- Routing Slips to the Fiscal Department for making advance payments
- Search outputs by vendors, funds, or requesters
- Routine reminders to vendors for books not supplied
- 5 × 3 inch order cards on continuous card stationery
- Fund Activity reports
- Accession Register
- Total listing of the ONORDER file.

## **5. Advantages of dBase II**

There are several advantages in using dBase II as a general-purpose applications language for microcomputers. Some of these have already been discussed in this paper. Some of the other advantages are discussed below. It must however, be said that one or more of these advantages would probably be equally true of other DBMS packages.

The programming effort required to develop the acquisition system was considerably less than would have been required had a programming language such as BASIC been used. This is primarily because of the ease with which it is possible in dBase II to provide indexed access to the database from one or more fields or attributes of one or more relations, and the fact that index files are automatically updated on data-entry and data updating operations. In a programming language such as BASIC, considerable programming effort would have been required to create and update index files whenever a new item enters the database or when an item is updated or deleted.

The programming was done by one of the authors (Mr Haravu), a librarian with formal training in computer programming and systems analysis, and with several years of experience in computer-based systems for library and information work. The system comprises about 70 different programs and was developed by one person in six months.

It is extremely easy in dBase II to define the structure of a database file or relation. Having once created a DBF, it is equally simple to modify the structure (e.g. add a new attribute, delete an attribute, redefine parameters of an attribute). The records of the unmodified DBF may be loaded into the DBF with the modified structure using a simple instruction. Changes required in processing programs to use a DBF whose structure has been modified are minimal, and may not at all be required in some situations.

Control of screen characteristics (e.g. cursor positioning, inverse video, etc.) are done using simple instructions. The use of the Screen Editor package permits the quick definition and/or modification of screen forms.

The full-screen editing features of dBase II make it possible to correct entered data without the need for extensive re-entry. It is possible to toggle between the exchange and insert modes of data entry. The full screen-editing features of dBase II alone would justify the use of dBase II as a general purpose applications language as compared to other languages such as BASIC or COBOL in applications involving considerable entry of textual data.

A significant advantage of dBase II is that it is possible to query a database interactively using simple instructions. Such a facility is useful for quickly producing reports which cannot be predetermined and therefore cannot be provided as part of the processing programs of an application. With a programming language such as BASIC such a requirement can

be met only by writing special programs every time a special report is required. The DO WHILE and the IF-ELSE-ENDIF structures in dBase II, and the DO statement that calls and executes a procedure or subroutine with passing of parameters, facilitate the writing of modular programs, that are easy to understand, correct and maintain. The ONORDER file, which is the biggest file in the system, has 1,350 records at the time of writing (September 1986). The ONORDER file is purged every three months to remove records for books received. Each ONORDER file record is 338 characters in length and it is estimated that the file could expand to 5,000 or so records and still leave enough space for the other data and command files. The time taken to print the entire ONORDER file depends on the speed of the printer – about 12 seconds/record using a letter quality printer (such as a Spinwriter) and about 4 seconds/record using a 120 character per second Dotmatrix printer. The response time in the search function for searches on indexed keys is between 4 and 5 seconds. On keys which are not indexed the search is sequential, and can vary from a few seconds to a few minutes depending on the record's position in the file.

©Aslib, L.J. Haravu, P.S. Jadhav and R. Sreeramana

## References

1. J. Wagner. Database management system design for library automation. *Journal of Library Automation*, vol. 13 no. 4, 1980, pp. 56-61.
2. I.A. Macleod. A database management system for library automation. *Information Systems*, vol. 6 no. 2, 1981, pp. 131-138.
3. A. Trevelyan and M. Rowat. *An investigation of the use of systems programs in library applications of microcomputers*. Library and Information Research Report 12. London: British Library, 1983. ISBN 0-7123-3017-8.
4. R.G. Crawford. The relational model in information retrieval. *Journal of the American Society for Information Science*, vol. 32 no. 1, 1981, pp. 51-64.
5. S. Bordwell. dBase II—Library use of a microcomputer database management system. *Program*, vol. 18 no. 2, 1984, pp. 157-165.
6. P. Millar and J. Cochrane. Administration of a reserve collection at Paisley College using dBase II. *Program*, vol. 19 no. 3, 1985, pp. 262-270.

## Authors

L.J. Haravu, Manager, Library and Documentation Services, International Crops Research Institute for the Semi-Arid Tropics (ICRISAT), Patancheru P.O. 502 324, A.P., India.

P.S. Jadhav, Library Officer, ICRISAT, Patancheru P.O. 502 324, A.P.,  
India. ,  
R. Sreeramana, Library Associate, ICRISAT, Patancheru P.O. 502 324,  
A.P., India