A DATA BASE MANAGEMENT SYSTEM FOR LIBRARY OPERATION

A project report

by

AHSANUL HABIB

Approved as to style and contents by

Dr. Syed Mahbubur Rahman
Associate Professor & Head,
Department of Computer Science & Engineering, BUET, Dhaka.

Chairman
[ Supervisor ]

Dr. Md. Shamsul Alam
Associate Professor
Department of Computer Science & Engineering, BUET, Dhaka.

Member

Dr. A.K.M. Ahsanullah
Director, BANSDOC
Science and Technology Division,
Ministry of Education,
Mirpur Road, Dhanmondi, Dhaka.

Member
[ V.C.'s nominee ]
A DATA BASE MANAGEMENT SYSTEM FOR LIBRARY OPERATION

by

AHSANUL HABIB

A project report submitted to the Department of Computer Science & Engineering in partial fulfillment of the requirements for the degree of Postgraduate Diploma in Computer Science & Engineering

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
BANGLADESH UNIVERSITY OF ENGINEERING & TECHNOLOGY
DHAKA 1000, BANGLADESH

July 1990
ACKNOWLEDGMENTS

I want to thank, first, my supervisor, Dr. Syed Mahbubur Rahman for his kind guidance of this project. His constructive skepticism contributed significantly to the refinement of the undertaking. I appreciate, more than anything else, his remarkable patience with a project that suffered deadline overruns at almost every stage. Were he a little less tolerant [of faults that were entirely mine!] this project could never be completed.

Of people whose encouragement and support made this undertaking possible, Professor A.K. Basak, former Administrator of Rajshahi University Computer Centre, deserves first mention. His action of recommending me [a person having no formal training at all in computing!] for a Postgraduate Diploma in Computer Science & Engineering was an act of bold unorthodoxy that could make him the target of criticisms. Appreciation, on similar counts, is due to Professor H. Shahjahan, National Coordinator of the AIT-BUET Programme, who too ran a risk of being criticized for allowing me to be admitted into the Programme. I am honoured by the confidence that they had on me, and consider myself lucky to have been able to show that their confidence was not misplaced.

Grateful acknowledgment is due to Mr. A.S.H. Humayun Kabir Khan, Software Engineer, CIPROCO, for his assistance with programming -- to BYTE, a computer training cum consulting firm in Rajshahi, for allowing me unlimited access to their computing facilities where a substantial part of the development and testing was done -- to fellow users of the CSE Microcomputer Laboratory, especially S. Towfiqur Rahman, Zunaid Hamid Kazi, and M. Masroor Ali, for their comments and observations and the occasional troubleshooting tips -- to Mr. Hazmul Haque of BUET Computer Centre for making available the list of book holdings of the Centre's library for use as a test data base -- and finally the many friends who have acted as unsung but invaluable advisors.

UNDP scholarship support towards this project, and my employer institution’s gesture in providing me with necessary leave is gratefully acknowledged. Without either this project would not have been possible. I want to thank, too, the examiners of this report for their comments and suggestions on an earlier draft.
The present report is the outcome of a project aimed at developing a PC based library data base management system capable of handling all information processing needs of a large rental library as are typically found in universities. Emphasis has been laid primarily on systems analysis and design -- identification of inputs, specification of storage and retrieval strategies, and design of outputs. Programs have been written to set up a working model which provides demonstration of the system using a 1024-record test data base.

Chapter 1 provides an introduction to the project and presents some theoretical perspectives on data base design issues. Chapter 2 attempts to evaluate the exiting manual system and establish the potential benefits of a computerized alternative. Chapter 3 provides a general orientation to the modus operandi of the proposed system. Chapters 4 and 5 concentrate, respectively, on the detailed specification of the data bases and the outputs to be generated. Chapter 6 deals with the organization of individual program modules supporting the system. Chapter 7 concludes the report with an objective look at the limitations of the proposed system in its present form and the feasibility of its adoption for a real library.
## CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>OBJECTIVES OF THE PROJECT</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>THEORETICAL PERSPECTIVES ON DATA BASE DESIGN</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>DATA PROBLEMS IN TRADITIONAL FILE MANAGEMENT</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>THE LOGICAL VS. PHYSICAL VIEW OF DATA</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>ATTRIBUTES OF A RELATIONAL DATA BASE</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>INFORMATION RETRIEVAL IN A RELATIONAL SYSTEM</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>NORMALIZING DATA STRUCTURES</td>
<td>7</td>
</tr>
<tr>
<td>2.</td>
<td>EVALUATION OF A MANUAL SYSTEM</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>EXISTING ACQUISITIONS PROCESS</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>EXISTING RENTAL MANAGEMENT SYSTEM</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>JUSTIFICATION FOR A NEW SYSTEM</td>
<td>13</td>
</tr>
<tr>
<td>3.</td>
<td>THE PROPOSED SYSTEM AND ITS OPERATION</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>ACQUISITIONS</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>RENTAL MANAGEMENT</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>ON-LINE INQUIRY</td>
<td>18</td>
</tr>
<tr>
<td>4.</td>
<td>RECORD DESIGN</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>THE ACQUISITIONS SUBSYSTEM ENTITIES</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>THE PERMANENT BOOK DATA BASE</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>THE RENTAL MANAGEMENT SUBSYSTEM</td>
<td>22</td>
</tr>
<tr>
<td>5.</td>
<td>OUTPUT DESIGN</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>REPORTS [ acquisitions subsystem ]</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>REPORTS [ rental management subsystem ]</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>VIDEO SCREEN DESIGN FOR ON-LINE INQUIRIES</td>
<td>32</td>
</tr>
<tr>
<td>6.</td>
<td>THE LOGICAL STRUCTURE OF THE SYSTEM'S OPERATION</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>INQUIRING THE DATABASE</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>GENERATING REPORTS</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>UPDATES AND MAINTENANCE</td>
<td>44</td>
</tr>
<tr>
<td>7.</td>
<td>IMPLEMENTATION AND CONCLUSION</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>TEST-RUN OF THE SYSTEM</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>LIMITATIONS AND POSSIBLE ENHANCEMENTS</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>IMPLEMENTATION CONSIDERATIONS</td>
<td>49</td>
</tr>
<tr>
<td>*</td>
<td>APPENDIX [ source code listing ]</td>
<td>52</td>
</tr>
</tbody>
</table>
1. **INTRODUCTION**

Data Base Management Systems (DBMS) have become so well accepted as an implementation approach for information systems that arguments today are more likely to centre on whether a particular type of data base organization is most effective for a given application, rather than whether or not to use a DBMS. This observation applies not only to businesses but to all installations where large volumes of data require to be managed and maintained. Large rental libraries, as is typically found in universities, is a prime candidate for useful application of DBMS. These libraries are large, often stock 100,000 or more books, handle new acquisitions at the rate of several thousand a year, and have a user base which too sometimes run into the thousands. Understandably, manual operation of such a colossal system becomes cumbersome, wasteful and time-consuming. Adoption of a computerized DBMS, by streamlining acquisitions, catalogue inquiries, and rental operations of such a library can greatly enhance the service provided to its users.

1.1 **THE OBJECTIVES OF THE STUDY**

This project undertakes the task of designing an integrated DBMS for such a library. 'Turnkey' application of library DBMS have been around for some time for mainframes and minicomputers [e.g. 'Alleycat' running on VAX]. However, these applications and the supporting hardware and software are expensive to purchase and maintain. To have reasonable prospect of being accepted by concerned authorities in our country, any such system has to be implementable on [relatively] inexpensive IBM PCs or compatibles using some easily available PC DBMS software.
The last consideration, easy availability of supporting PC DBMS software, effectively restricts the data model to be used to a relational one. PC DBMS software almost universally use a relational model, and the issue becomes one of deciding which of the available software to use. In the PC DBMS environment dBASE III Plus is the obvious choice. dBASE is the most widely used microcomputer data base management system. Its long standing popularity and consequent availability of trained manpower, wide availability of model applications in published sources, and third party software that extend dBASE (e.g. Clipper compiler for dBASE source code), should maintain this market domination.

Although it is not impossible to create, operate and maintain the library data base using dBASE III Plus command level functions, it requires quite a high level of learning which is unlikely to be attainable by most of the existing staff of the library. Maintaining and accessing a data base using dBASE command level functions is probably similar to learning how to use an on-line information retrieval system. The continued setting up of search strategies which are in regular use may also become somewhat boring. Thus, trained operators could find that it is not worth the bother, and the data base might not be used at all by untrained personnel and the general users of the library. A preferred solution, then, is to create a complete menu-driven software which can be used at considerably more ease by non-programmers.

1.2 THEORETICAL PERSPECTIVES ON DATA BASE DESIGN

DBMS software is a term that covers all programs that allow application construction without direct programming of input, output, and processing operations. A collection of data stored in at least two files is considered as a data base. Thus,
software systems that can work with fields from more than one file simultaneously is given the title DBMS.

1.2.1 DATA PROBLEMS IN TRADITIONAL FILE MANAGEMENT

DBMS as a distinct branch of programming application emerged from the increasing need of preventing data problems in automated information systems relying on traditional file management. It was often discovered, after elaborate investments in computer hardware and data collection, that data were sometimes inaccurate and could not be retrieved quickly or in a format useful for dealing with non-routine problems. Some of such data problems, which DBMSs seek to remove, are:

Data redundancy

Storage of the same data in multiple records or files is redundant and leads to a host of data management problems. For example, if the complete address of a publisher is to be entered for each entry of a book published by it, the same data would have to be repetitively entered as each book record is constructed. Further, multiple storage of the same data wastes space in the files -- and this costs money, both for the storage media and for the extra processing time required to deal with these larger files. If, under such a system, the address of a publisher is changed, the corrections will have to be entered into every record in the file that represents a book published by that publisher.

Poor data integrity

Integrity refers to the reliability of data, and the redundancy problem discussed above is a major cause of data integrity problems. As an example, the redundant storage of publisher addresses in the book file leads to inconsistency [and consequent integrity problem] if the publisher's address
changes and only a subset of the records are properly updated. **Difficult data access for application development**

Data needed to respond to a special information request may exist in a collection of files. For example, a library wants to prepare book overdue notices containing the user_id, name, and address of the borrower, list of books overdue and due date of each, etc. Preparation of such a report will require a connection between the 'book issue' file and the main 'book' file on the basis of accession number to obtain the title and other particulars of the books. A further link is required between the 'book issue' file and the 'user' file [on the basis of user_id] to obtain the name and address of the borrower. To explicitly program this complicated 'join' operation is difficult and expensive. If the files in which these information exist form part of an organized data base the time and expense involved in preparing such a report can be curtailed substantially.

### 1.2.2 THE LOGICAL VS. PHYSICAL VIEW OF DATA

To overcome the data problems introduced above, data base management software provides for 'transparent' integration of data from different files. There are a variety of ways of linking separately stored data. These alternative models for relating data are called logical structures. A specific data base management system deals with only one logical structure. There is no globally best model for dealing with data -- each has its advantages and disadvantages -- usually reflected as trade-offs among flexibility, ease of use, and the speed with which data may be accessed and processed.

Another element of data base management software is the high degree of separation of the user of data from the details of its
storage. This concept is called data independence. The idea is to let the user or programmer manipulate data from a logical viewpoint, concentrating on the relationships in the data. The DBMS software deals with the physical storage of data in records and files and provides mechanisms for efficiently retrieving and organizing data. Thus applications developed using a DBMS can be data independent. The programmer only concerns himself with the logical design for the overall data base which provides enough details of data base content and organization to perform all required retrieval and manipulation operations without actually knowing the physical storage details. Programs in a DBMS are not at all concerned about the format of the data, and programs which access variables by DBMS defined field names can run without modification even if the physical data format is changed in the meantime.

Most data base management software manipulate a data base using one of the three data structures: (1) Hierarchical or tree model, (2) Network or plex model, and (3) Relational model. Our choice, the relational model, is a relative newcomer to data organization concepts. Traditionally it was the hierarchical and network models which were used in large information systems, and they are still considerably faster in retrieval operations. The problem with them, however, is that these first two models must define relationships among records at the time the data base is designed and before any data entry is done. Flexible data retrieval requirements, and systems used to provide interactive data access, are better served by the relational model. Although data retrieval is often slower in this system than in those using hierarchical or network models, the advantage is that relationships among records can be defined
after data have been entered.

1.2.3 **ATTRIBUTES OF A RELATIONAL DATABASE**

A relational database is viewed logically as a collection of two dimensional tables. In dBASE III Plus, each of such tables represents a file, where each row in the table is one record in the corresponding file, and each column of the table is represented as a field [spanning a given number of characters] of each record. For a relational database the following rules must be satisfied for data in each database file:

1) Each database file may contain only one kind of record. In the system that we are presenting, the book, publisher, supplier, user, etc. all are stored in separate files.

2) Each record has the same fields, and each field has a unique name. Each record in a relational database is unique. That is, no two records may contain exactly the same values in all matching columns. Each record in a relational database represents some entity: a book, a publisher, a user, etc.

3) Each field occurs only once in a record. There can be no repeating fields or groups. Thus we cannot construct a book record which holds the accession number of three different books!

4) No predetermined sequence is required for the records in a file. The DBMS software takes care of all necessary indexing or sorting.

5) New database files can be produced by combining and/or extracting subsets of data from other database files.

1.2.4 **INFORMATION RETRIEVAL IN A RELATIONAL SYSTEM**

Three fundamental operations are performed on files constituting a relational database as part of a data retrieval process. The mechanisms for accomplishing these operations are provided by the DBMS software and do not require to be user-programmed. These are:

**Join**

The relational join provides the logic for temporarily combining records from different files. Records are combined by
matching field values between the two files that are joined. In theory, the join operation is performed thus -- beginning with the first record in the first file, all records in the second file are examined to see if the logical condition defining the join is satisfied. If the condition is satisfied a record of the 'joined' file is constructed by taking fields of the two parent files. The process ends when every record in the first file has been examined. For large files, the join operation could be very time consuming. If there are \( n \) records in the first file, and \( m \) records in the second, then at least \( n \times m \) comparisons are required. Many relational DBMSs, including dBASE III Plus, provide faster techniques for performing joins when records are to be linked on the basis of indexed fields.

**Selection**

The selection operation is relatively simple. It implies the construction of a new file by extracting a subset of records from an existing file.

**Projection**

Just as the selection operation performs extraction of a subset of the records, the projection operation performs the extraction of a subset of the columns of a data base file. There is an additional complicating factor with the projection operation. No two rows in a relational table can be identical (i.e. have the same value in all its fields), but extracting a subset of the columns in a table might lead to this condition.

1.2.5 **NORMALIZING DATA STRUCTURES**

The objective for grouping fields into records is to provide efficient access to data base contents, both from the standpoint of computer processing time and correctness of the information retrieved. Careless record design is perhaps the most important
factor generating data redundancy and data integrity problems. Unfortunately, it is possible to define records that work adequately for an initial application but lead to bottlenecks when new applications are built or ad hoc retrievals are attempted. For this reason, a formal application independent approach requires to be followed for grouping fields that form a record in a data base file.

The key to properly associating fields into records is a process called normalization. Records designed according to this process are said to be in normal form. Several "normal" forms have been identified, which represent successively higher degree of refinement in the organization of data from its initial 'unnormalized' state. These are:

First normal form (1NF)

Elimination of repeating columns represents the first step in the normalization process. Conversion of the unnormalized record design to first normal form is accomplished by replicating records so that all of the records can be present as a two dimensional table without repeating any fields of the table. Data in this form is also known as a 'flat file' -- data represented as a two dimensional table with no repeated fields or groups of fields.

Second normal form (2NF)

An understanding of the concept of 'key' of a record is a prerequisite for meaningful exposition of the second normal form. A key represents an identifier of a record. Three types of keys are normally defined: primary key [ unique identifier of a record ], secondary key [ an alternative identifier, may not be unique ], and foreign key [ fields other than the primary key used to link records together ]. Another frequently encountered
term is 'composite key' which is the concatenation of several fields in a record and can serve as primary key. The case for second normal form of data organization comes when a composite key is used. It is possible that some fields of the record are dependent on one constituent field of the composite key while others are dependent on one or more other fields constituting the key. Second normal form is obtained when fields are allocated to records so that every field in a record is dependent on the whole key for that record.

Third normal form (3NF)

2NF record design is not immune from the problem of 'transitive dependence' [when values stored in non-key fields depend directly on values stored in other non-key fields]. Such dependence result in data redundancies and should be removed by splitting the record and moving the transitively dependent fields to a different record [i.e. forming another data base file].

A record does not always require all three normalizing steps. If the 1NF record contains no fields that depend on only a part of the primary key, it is already in second normal form. If there are no transitive dependencies after conversion to second normal form, the record is also in third normal form. A fourth and fifth normal form have been defined, but for most real life applications designing records to meet the conditions of the third normal form is adequate to avoid most data management logical problems. The library DBMS presented in this report has maintained strict concordance with 3NF standard in designing records of all data base files used in the system.
2. EVALUATION OF A MANUFACTURED SYSTEM

An information system perspective of an integrated data base management system for a typical university rental library requires, first, the identification of its various functional areas. Close examination of the operations of two university libraries [BUET Rental Library, and Rajshahi University Central Library] identify two such distinct areas: (1) acquisitions, and (2) rental management. The first concerns all the steps from ordering of books to their final shelving. A book passes only once through this phase. The second area, rental management, on the other hand is a continuing process. The operations under each are described below in detail.

2.1 EXISTING ACQUISITIONS PROCESS

The acquisitions process starts with requisitions for books submitted to the library from individual departments. Periodically, when ready, the library requests various departments of the university to submit their requisitions. Sometimes, the departments send such requisitions of their own. In either case, to the library, the requisitions outstanding at any point in time constitutes, for a particular department, the list of books which the department concerned would ideally like to see acquired. Obviously, the sum of such requisitions usually becomes several times the size that the library's funds would allow. The steps in the acquisitions process are:

Selection of books

The lists received from individual departments get an initial screening for completeness of citation. It has been observed that quite a few citations are so inadequate as to make unique identification of a book impossible. Such cases are
referred back to the requesting departments. Where all books requested by a particular department cannot be purchased (most often due to budget constraints), assignment of priorities is sought and a decision is then taken about final selection of books to be ordered.

Selection of suppliers

At this stage the lists of books pertaining to various departments are merged and then grouped according to publishers of the books. The next step is to consult records to decide which supplier to order from. Sometimes such suppliers are the publishers themselves, but most often, it is one or two local firms who work as agents of various publishers.

Placement of order

A requisition is drawn up for each supplier where the books ordered are listed, grouped by publisher. In this list the name of individual publishers normally appear as a sub-heading. For each book the title, name of author, year of publication, and number of copies required are specified. Such order lists are sent to respective suppliers. Payments are made against invoices provided by the supplier after books are received.

Receiving of books

Once books arrive these are compared with the orders and discrepancies, if any, are noted. Individual slips containing the identification of each book and relevant accounting information are prepared at this stage. Such slips are preserved for audit purposes and are not used anywhere in subsequent processing.

Cataloging

This is a multi-step process and has an entire section devoted to it. For a book that has to be catalogued, the first
step is to find out if the book is a new one, or just another
copy of a book already held at the library. The shelf-cards
preserved at the cataloging section are consulted for this
purpose, and those matching are withdrawn from the card-file.
This is done manually, and often involves a considerable amount
of card sifting. The next step in the cataloging sequence is to
write the draft catalog card for the book. If the book is an
'existing' one, a clerk just makes a copy of the old card with
new accession number and date. In case of a new book it goes to
the cataloguers who assign class number and subject number to
the book and the usual draft catalog card is written out. The
draft catalog card is a piece of plain paper [as contrasted to
some printed form which could ensure a more consistent style]
which the typists copy on to cards. In all five different cards
are prepared for each book, e.g., book-card, shelf-card,
subject-card, and 2 copies of catalog cards. The book now goes
to the preparation section. The book-card accompanies the book,
other cards are placed into the appropriate shelf or card-file
by the cataloging department staff.

Preparation and shelving

Once preparation of the cards is complete the book, along
with its book-card, is sent to the preparation section for
labeling and pasting of card-jacket and issue slip. The book is
then shelved and this concludes the acquisition process.

2.2 EXISTING RENTAL MANAGEMENT SYSTEM

This functional area covers the book issue and book return
procedures. We will discuss them separately.

Book issue

Although the two libraries that we used as our model
differed in the specific details of the book-issue procedure,
the following steps appear to be shared by both. The book(s) to be issued, along with the borrower's library card, are placed before the issuing officer who makes appropriate entries on the book-card(s) and the user's library card and writes out a gate-slip so that the book may be legally carried out of the library. The gate pass is checked at the exit and retained. At the end of each day the book-cards accumulating throughout the day are sorted (on accession number) and merged with the collection of such cards from the past days.

Book return

On return of a book the first steps are to cancel the issue marks from the book, updating the borrower's library card, the fetching the book-card from the card-file collection. The book-card is then reinserted, and the book returned to its appropriate place in the shelves.

2.3 JUSTIFICATION FOR A COMPUTERIZED SYSTEM

Computerization can expedite almost all aspects of a library's operation, starting from the selection of books. With data on book holdings available in the computer it is possible to check easily which of the books are already held at the library. This may be important when all the books listed by an individual department cannot be ordered. It is also easy to check for any duplication in the orders, which is a quite frequent problem.

Computerization can benefit other steps of the acquisitions process as well. In a computerized system, actions like preparing order-list going to individual suppliers, catalogue cards, etc. can be performed faster and more accurately. For example, the five catalogue cards for each book have to be typed separately, and it is known to have occurred that
different copies of catalog cards for the same book have contained differing information as to call number, accession number, etc.

Poor report generation is another inadequacy of a manual system. To be useful, list of new book arrivals should be made available as soon as possible and, ideally, should be presented in at least four different forms -- sorted by title, by author, by subject classification, and by department. Because of the cumbersome manual system only one [if at all!] such list is prepared for public consumption. Worse still, the list is often of limited use because some of the books almost always fail to reach the shelves because of acquisitions process bottlenecks!

In general, report generation of all kinds can be made more versatile, introducing a general improvement in the quality of services provided by the library.

The card-based catalogue inquiry facilities now available is also grossly inadequate. Cards being physical entities can be sorted in only one order at any particular point in time. Thus, to provide searching facilities by author and subject requires two separate sets of cards. A computerized data base can support a much more versatile catalogue inquiry system at little extra cost. That is, if the system is computerized, the entire collection of the library can be searched in any desired order as long as the item of information constitutes an 'entity' of the data base. Thus a single inquiry service may provide options to search by author, by title, by call number, by accession number, by subject classification, etc. Moreover, using a computerized data base makes the system immune from many problems typically associated with card-based indexes [e.g. loss/misplacement of cards, typographic errors, etc.].
A computerized system for handling book issues and returns can greatly reduce the time required to serve a borrower. It has been observed that rush of students at the library counter at certain hours of the day is quite high. Computerization can curtail those long library-counter queues.

Computerization may also lead to better administration of the library's resources. Under a manual system it is difficult to implement a system of issuing reminders to defaulting borrowers. The result is that sometimes a default is noticed only when the user wants to leave the institution and asks for a clearance from the library.

Neither of the libraries under reference maintain a reservation system wherein users can reserve a lent-out book and form a queue for its subsequent issues. A reservation facility can greatly enhance the usefulness of a library to its users, at the same time that it is almost impossible to implement on a manual system. Introduction of a computerized DBMS for the library can easily support a book reservation system for the borrowers. It is quite simple to program the computer to record reservations against a book, and to print out appropriate notices, etc.
3. THE PROPOSED SYSTEM AND ITS OPERATION

The system will be built around a file [BOOK.DBF] containing data about all books in the library. Important data includes: title, author, year of publication, call number, accession number, subject classification code (primary), subject classification code (secondary), publisher-id, requisitioning department-id, current status (e.g. whether in the library or lent out), etc.

3.1 ACQUISITIONS

There will be a file [ORDR.DBF] containing data about books on order. The records of this file are to be created at the initiation of the purchasing process. A book will be first recorded as an entry in this file. Fields are provided for all relevant particulars [e.g. title, author, publisher, year of publication, price, number of copies to order, subject code(s), date of placement of order, etc.]. This information will be used to generate requisitions to individual suppliers.

Certain fields of this record [e.g. accession number, call number, etc.] cannot be completed before the book is received. It is also possible that some of the citations provided by the departments' requisition list may be incomplete. As long as there is adequate information to uniquely identify a book it can be entered in the orders data base, the remaining particulars to be completed when the book finally becomes available. The record will then be transferred to the main book data base [BOOK.DBF] and deleted from the books on order file [ORDR.DBF]. One or more reports on new book arrival may be printed at this stage. For books that are not available, the corresponding
A record may be flagged accordingly and later used to produce a printed list of books which could not be made available, for information of requisitioning departments.

Normally when books are ordered the library receives the ordered books within two or three months after ordering. In some cases it may receive notification from the suppliers that they are unable to supply such and such books. In still other cases, there may not be any reply at all from the suppliers. In our proposed system it is possible, easily, to check all pending orders older than a specified period and appropriate actions taken against each.

The acquisitions subsystem will require several supporting data bases to contain information pertaining to individual departments, publishers, and suppliers. These, respectively, will be named DEPT.DBF, PUBL.DBF and SUPP.DBF.

3.2 Rental Management

There will be a file [ USER.DBF ] of library users which indicate the user name, user address, fines due, user identification number, number of books currently on issue, and the borrowing limit (both in terms of number of books and duration allowed) that apply to a particular user, date the user's library privilege expires (if applicable), etc. A 'relation-file' [ LOAN.DBF ] comprising of due-date, accession number and user identification number will link the 'book' and 'user' data bases so that reports may be printed that show all available details on a book and the user borrowing it.

Under the system, the library lets users reserve any book. For example, if a user wants to borrow a particular book but it was checked out, the user can then enter his name in a reservation queue for that book. When the book is returned, it
is possible to check whether anyone has reserved this book. If yes, a notification slip will be prepared and the book kept for the first person who reserved that book. This is effected by using yet another 'relations file' [ RQUE.DBF ] which contains, as in LOAN.DBF discussed earlier, the date, accession number and user-id of person reserving the book.

3.3 ON-LINE INQUIRY

On-line inquiry is supported for the orders [ ORDR.DBF ], book [ BOOK.DBF ] and user [ USER.DBF ] data base files individually as well as in relationally joined form where necessary.

The orders data base may be queried on:

1) date placement of order
2) publisher
3) requisitioning department
4) subject-code [composite]
5) subject-code [primary]
6) supplier/agent

The book data base may be queried on:

1) accession number
2) author
3) call number
4) subject-code [composite]
5) subject-code [primary]
6) title

The user data base may be queried on

1) date of expiry
2) user-city
3) user-department
4) user-id-number
5) user-name
6) user-type
4. RECORD DESIGN

dBASE III Plus' CREATE command has been used to define the record structure for the various data base files. In all these data base files, there are two 1-byte fields, named <flag> and <stat> respectively, to be used as status indicator and/or maintaining log of reports generated, etc. The acceptable values for these fields are @ABCD	EFGH	IJ	KLMNO which are assigned different meanings in different contexts and some values remain unused. These two fields may sometimes be used or modified by programs in a manner transparent to the user and, except at initial entry time, will not be directly accessed by the user.

The various data base files comprising the system are introduced below. In the text we will use the convention of enclosing data base file field-names in paired angle brackets.

4.1 THE ACQUISITIONS SUBSYSTEM

The structure of ORDR.DBF [ the books on order data base ] and description of the contents of its various fields appear as Table 4.1. This file is not intended to be a permanent data base. Records will be kept in this file during the period that a book order is pending. Once received, the information that is to be permanently retained will be transferred to the book data base [ BOOK.DBF ] and the record will be deleted from ORDR.DBF. Because of its temporary nature and expected low number of record accumulation, we have not been stringent about the size of individual records. Records in this file are 254 character in length, which is twice that of most other data bases in the system.

All tables displaying the structure of various data bases used in the system follow the same pattern of presentation as
appears in Table 4.1. Explanation of the table headers are:

- NDX = name of relevant index file
- fnam = field name
- width = field width
  [ n indicates the number of decimals, where applicable ]
- type = field type [ char=character, N=numeric, D=date ]
- recl = record length [ in bytes ]

### TABLE 4.1 STRUCTURE OF THE BOOKS_ON_ORDER FILE: ORDR.DBF

<table>
<thead>
<tr>
<th>NDX</th>
<th>fnam</th>
<th>width</th>
<th>type</th>
<th>description</th>
<th>recl</th>
</tr>
</thead>
<tbody>
<tr>
<td>O_TITL</td>
<td>title</td>
<td>64</td>
<td>char</td>
<td>title of the book</td>
<td>254</td>
</tr>
<tr>
<td>O_AUTH</td>
<td>auth</td>
<td>32</td>
<td>char</td>
<td>author</td>
<td></td>
</tr>
<tr>
<td></td>
<td>city</td>
<td>32</td>
<td>char</td>
<td>location of publisher</td>
<td></td>
</tr>
<tr>
<td></td>
<td>fnam</td>
<td>32</td>
<td>char</td>
<td>publisher's name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>year</td>
<td>4</td>
<td>char</td>
<td>year of publication</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cost</td>
<td>8:2</td>
<td>N</td>
<td>price in Taka</td>
<td></td>
</tr>
<tr>
<td></td>
<td>copy</td>
<td>4:0</td>
<td>N</td>
<td>copies ordered</td>
<td></td>
</tr>
<tr>
<td></td>
<td>call</td>
<td>16</td>
<td>char</td>
<td>call number</td>
<td></td>
</tr>
<tr>
<td></td>
<td>acen</td>
<td>8</td>
<td>char</td>
<td>accession number</td>
<td></td>
</tr>
<tr>
<td>O_SUBJ</td>
<td>subj</td>
<td>4</td>
<td>char</td>
<td>subject classification [primary]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sub2</td>
<td>4</td>
<td>char</td>
<td>subject classification [secondary]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>dord</td>
<td>8</td>
<td>D</td>
<td>date order placed</td>
<td></td>
</tr>
<tr>
<td>O_S_ID</td>
<td>s_id</td>
<td>2</td>
<td>char</td>
<td>supplier_id (code)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>p_id</td>
<td>2</td>
<td>char</td>
<td>publisher_id (code)</td>
<td></td>
</tr>
<tr>
<td>O_D_ID</td>
<td>d_id</td>
<td>2</td>
<td>char</td>
<td>requisitioning department_id (code)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>remk</td>
<td>30</td>
<td>char</td>
<td>remarks, if any</td>
<td></td>
</tr>
<tr>
<td></td>
<td>flag</td>
<td>1</td>
<td>char</td>
<td>log of reports generated</td>
<td></td>
</tr>
<tr>
<td></td>
<td>stat</td>
<td>1</td>
<td>char</td>
<td>status_code</td>
<td></td>
</tr>
</tbody>
</table>

The data bases for suppliers [ SUPP.DBF ] publishers [ PUBL.DBF ] and departments [ DEPT.DBF ] have similar structures. The records start with a 2-character identification number <?_id>, the next field of 2 characters <x > is reserved for possible future use, followed by the name of the entity <?nam>, address <?add>, city <?city>, an 8-character numeric field, a date field, and a remarks field <remk> of 8 characters [ the meta-character ? representing d;p;s respectively for 'department', 'publisher' and 'book' data bases ]. The details of the structure of these three data bases appear, respectively, as Table 4.2, Table 4.3 and Table 4.4.
4.2 THE PERMANENT BOOK DATA BASE

The permanent book data base is named BOOK.DBF. In a way records in this file are a 'subset' of records in ORDR.DBF. It
'inherits' records from the latter once all the missing information have been filled in. Information in the fields <titl>, <auth>, <p_id> and <d_id> migrate literally. The year of publication field <year> is trimmed of the leading '19' and included as a 2-character field. Truncation of leading spaces trims accession number <acn> down to 5 digits. The width of subject classification mnemonics <subj>, <sub2> and call number <call> fields are reduced to 3, 3 and 11 characters respectively through truncation of trailing spaces. Some information contained in ORDR.DBF such as publisher-name <pnam>, location <city>, supplier identification <s_id>, date ordered <dord> and remarks <remk> are not carried over into BOOK.DBF. Details of the structure of this data base and description of various information items contained in it are presented in Table 4.5.

### TABLE 4.5 STRUCTURE OF PERMANENT BOOK DATABASE: BOOK.DBF

<table>
<thead>
<tr>
<th>Field</th>
<th>Width</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITL</td>
<td>64</td>
<td>char</td>
<td>title of the book</td>
</tr>
<tr>
<td>AUTH</td>
<td>32</td>
<td>char</td>
<td>author</td>
</tr>
<tr>
<td>CALL</td>
<td>11</td>
<td>char</td>
<td>call number</td>
</tr>
<tr>
<td>ACCTN</td>
<td>5</td>
<td>char</td>
<td>accession number</td>
</tr>
<tr>
<td>SUBJ</td>
<td>3</td>
<td>char</td>
<td>subject classification [primary]</td>
</tr>
<tr>
<td>SUB2</td>
<td>3</td>
<td>char</td>
<td>subject classification [secondary]</td>
</tr>
<tr>
<td>P_ID</td>
<td>2</td>
<td>char</td>
<td>publisher_id (code)</td>
</tr>
<tr>
<td>D_ID</td>
<td>2</td>
<td>char</td>
<td>requisitioning department_id (code)</td>
</tr>
<tr>
<td>FLAG</td>
<td>1</td>
<td>char</td>
<td>log of reports generated</td>
</tr>
<tr>
<td>STAT</td>
<td>1</td>
<td>char</td>
<td>status_code</td>
</tr>
</tbody>
</table>

4.3 **THE RENTAL MANAGEMENT SUBSYSTEM**

Records in the user data base [USER.DBF] start with a 2-digit identification number <u_id>, next 2-character field <d_id> contains the id of the user's department, followed by the name <unam>, address <uadd>, city <ucty>, fines accumulated <fdue>, number of books currently on loan <nour>, borrowing
limit \(\text{blim}\), time limit \(\text{tlim}\), and lastly, date of expiry \(\text{dexp}\). The description of USER.DBF record structure has been more thoroughly noted in Table 4.6.

**TABLE 4.6 STRUCTURE OF DATABASE OF LIBRARY USERS: USER.DBF**

<table>
<thead>
<tr>
<th>? .NDX</th>
<th>fnam width type</th>
<th>description</th>
<th>recl = 126</th>
</tr>
</thead>
<tbody>
<tr>
<td>U_U_ID</td>
<td>u_id  6 char</td>
<td>user identification number</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d_id  2 char</td>
<td>department identification code</td>
<td></td>
</tr>
<tr>
<td></td>
<td>unam 32 char</td>
<td>name of user</td>
<td></td>
</tr>
<tr>
<td></td>
<td>uadd 32 char</td>
<td>address of user</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ucty 30 char</td>
<td>city</td>
<td></td>
</tr>
<tr>
<td></td>
<td>fdue 8:2 N</td>
<td>fines outstanding (in Taka)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ncur 2:0 N</td>
<td>number of books currently on issue</td>
<td></td>
</tr>
<tr>
<td></td>
<td>blim 2:0 N</td>
<td>borrowing limit (number of books)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>tlim 2:0 N</td>
<td>time limit (days)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>dexp 8 D</td>
<td>date of expiry of privilege</td>
<td></td>
</tr>
<tr>
<td></td>
<td>flag 1 char</td>
<td>log of reports generated</td>
<td></td>
</tr>
<tr>
<td></td>
<td>stat 1 char</td>
<td>status_code [holds user_type code]</td>
<td></td>
</tr>
</tbody>
</table>

The rental management subsystem uses two ‘relations file’ LOAN.DBF and RQUE.DBF to keep track of book borrowings and book reservations, respectively. The ‘join’ is established by the fields \(<u\_id>\) and \(<\text{accn}\rangle\) which respectively are ‘keys’ to the user database USER.DBF and the book database BOOK.DBF. In addition to the usual \(<\text{flag}\rangle\) and \(<\text{stat}\rangle\) fields a 9-character remarks field \(<\text{remk}\rangle\) is included in both LOAN.DBF and RQUE.DBF. Table 4.7 and Table 4.8 provide a more formal exposition of the record structure of these two files respectively.

**TABLE 4.7 STRUCTURE OF THE BOOKS.ON.ISSUE FILE: LOAN.DBF**

<table>
<thead>
<tr>
<th>? .NDX</th>
<th>fnam width type</th>
<th>description</th>
<th>recl = 30</th>
</tr>
</thead>
<tbody>
<tr>
<td>L_U_ID</td>
<td>u_id  6 char</td>
<td>user identification number</td>
<td></td>
</tr>
<tr>
<td>L_ACNN</td>
<td>accn 5 char</td>
<td>accession number of book borrowed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ddue 8 D</td>
<td>date book due for return</td>
<td></td>
</tr>
<tr>
<td></td>
<td>remk 9 char</td>
<td>remarks, if any</td>
<td></td>
</tr>
<tr>
<td></td>
<td>flag 1 char</td>
<td>log of reports generated</td>
<td></td>
</tr>
<tr>
<td></td>
<td>stat 1 char</td>
<td>status_code</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Width</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>R_U_ID</td>
<td>u_id</td>
<td>6</td>
<td>user identification number</td>
</tr>
<tr>
<td>R_ACCN</td>
<td>accn</td>
<td>5</td>
<td>accession number of book reserved</td>
</tr>
<tr>
<td>dreq</td>
<td>D</td>
<td>8</td>
<td>date reservation recorded</td>
</tr>
<tr>
<td>remk</td>
<td>char</td>
<td>9</td>
<td>remarks, if any</td>
</tr>
<tr>
<td>flag</td>
<td>char</td>
<td>1</td>
<td>log of reports generated</td>
</tr>
<tr>
<td>stat</td>
<td>char</td>
<td>1</td>
<td>status_code</td>
</tr>
</tbody>
</table>
5. OUTPUT DESIGN

The 'output' from the system are primarily of two types: (a) reports, which appear as hardcopy printouts, and (b) response to inquiries which are presented as screen displays. Hardcopy printouts relate to either of the acquisitions or the rental management subsystems. Tabular forms for these reports appear, respectively for the acquisitions and rental management subsystems, in articles 5.1.? and 5.1.? later in this chapter. On the top of each report 'form' the parameters related to the output are given. These are:

a) Article number and Title [a caption for the output]
b) Type [paper size, special requirements, etc.]
c) Purpose [a short description of the use of the output]
d) Generation [when produced, e.g. daily, monthly, etc.]
e) Variants [e.g. orders sorted by (1) supplier, (2) dept.]
f) Copies [..to be printed]
g) Distribution [where each copy goes]

On selecting the REPORTS option from the main menu, the user is presented with the following sub-menu from which the user has the option to select either acquisitions or rental management.

<table>
<thead>
<tr>
<th>L-DBMS REPORTS</th>
<th>LR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Acquisitions</td>
<td></td>
</tr>
<tr>
<td>2. Rental Management</td>
<td></td>
</tr>
<tr>
<td>3. Main Menu</td>
<td></td>
</tr>
</tbody>
</table>
5.1 REPORTS [acquisitions subsystem]

In our design the acquisition subsystem will generate the following reports. Details of each appear in the article no. indicated in the right margin.

1) requisition to individual suppliers
2) books ordered (sorted on departments)
3) books ordered (sorted on supplier)
4) new arrivals (sorted on BOOK->record#)
5) new arrivals (sorted on author)
6) new arrivals (sorted on subject)
7) new arrivals (sorted on title)
8) books unavailable (sorted on departments)

The specific output to be generated is to be selected from the following menu:

```
L-DBMS REPORTS * ACQUISITIONS

1. Orders to suppliers
2. books ordered _Dept.
3. books ordered _Supp.
4. new arrivals, by Rec#
5. new arrivals, by Auth
6. new arrivals, by subj
7. new arrivals, by Titl
8. books Unavailable
9. Previous Menu
```

The report skeletons appear in the following pages.
5.1.1 REQUISITION TO INDIVIDUAL SUPPLIERS
output type Hardcopy, 132 col.
purpose List of books for placement of order
generation As required
variants
copies [1]
distribution [1] respective suppliers

---

5.1.2 BOOKS ORDERED
output type Hardcopy, 132 col.
purpose Informing departments of books actually ordered
generation As required
copies [1] of each
5.1.4 NEW ARRIVALS
output type Hardcopy. 132 col.
purpose Informing users of new books arrival
generation As required
copies [1] of each

<table>
<thead>
<tr>
<th>&lt;d_id&gt;</th>
<th>&lt;s_id&gt;</th>
<th>&lt;titl&gt;</th>
<th>&lt;auth&gt;</th>
<th>&lt;p_id&gt;</th>
<th>date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.1.8 BOOKS UNAVAILABLE
output type Hardcopy. 132 col.
purpose Informing departments of ordered books which the suppliers are unable to supply.
generation As required
variants
copies [1] [2]

<table>
<thead>
<tr>
<th>&lt;dept&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;titl&gt;</td>
</tr>
<tr>
<td>&lt;titl&gt;</td>
</tr>
<tr>
<td>&lt;titl&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;titl&gt;</th>
<th>&lt;auth&gt;</th>
<th>&lt;year&gt;</th>
<th>&lt;p_id&gt;</th>
<th>&lt;dord&gt;</th>
<th>&lt;copy&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.2 REPORTS [ rental management subsystem ]
The following reports will be generated by the rental management subsystem. Detailed format of the output is noted in the articles indicated in the margin.

1) two-part issue card 5.2.1
2) books currently on loan (in subject order) 5.2.2
3) books issued (sorted on accession no.) 5.2.3
4) books returned (sorted on accession no.) 5.2.4
5) overdue reminder slip (students) 5.2.5
6) overdue reminder slip (teachers/officers) 5.2.6
7) reserved_book collection request 5.2.7
8) final fine report 5.2.8

The specific output to be generated is to be selected from
the following menu. Output specified as article 5.2.1 above is,
however, not an option here. It is automatically generated
whenever a book is issued [by selecting the appropriate option
in the UPDATES * RENTAL menu].

---

1. books still on Loan
2. books Issued
3. books Returned
4. o_due rem (Students)
5. o_due rem (Teachers)
6. reserve collection
7. final Fine report
8. Previous Menu

---

5.2.1 output type 2-PART ISSUE CARD
purpose Hardcopy, 132 col. [vert. perforation at col.66]
Per issue
variants [1], detachable in two parts [a] [b]
copies [a] to be preserved at the issue counter
[b] to be shown at the library exit
distribution

---

<title> : <title>
<auth> : <auth>
<call> : <call> <acct> <cost>
date issued = <date> <date issued >
date due = <date due> <ddue>
borrower = <u_id> <unam> <uadd> <ucity>

5.2.2  BOOKS CURRENTLY ON LOAN
output type  Hardcopy, 132 col.
purpose  For information of library users
generation  Daily
variants
copies  [1]
distribution  [1] Library issue counter

<table>
<thead>
<tr>
<th>date</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;subj&gt; &lt;call&gt; &lt;titl&gt; &lt;auth&gt; &lt;p_id&gt; &lt;ddue&gt;</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

5.2.3  BOOKS ISSUED
output type  Hardcopy, 132 col.
purpose  Maintaining daily record of books issued
generation  Daily
variants
copies  [1]
distribution  [1] library management

<table>
<thead>
<tr>
<th>date =</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;acct&gt; &lt;titl&gt; &lt;auth&gt; &lt;call&gt; &lt;u_id&gt; &lt;ddue&gt;</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

5.2.4  BOOKS RETURNED
output type  Hardcopy, 132 col.
purpose  Maintaining daily record of books returned
generation  Daily
variants
copies  [1]
distribution  [1] library management

<table>
<thead>
<tr>
<th>date =</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;acct&gt; &lt;titl&gt; &lt;auth&gt; &lt;call&gt; &lt;u_id&gt; &lt;ddue&gt; days</td>
</tr>
<tr>
<td>&lt;acct&gt; &lt;titl&gt; &lt;auth&gt; &lt;call&gt; &lt;u_id&gt; &lt;ddue&gt; ov'due</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

---
30
5.2.5:6 OVERDUE REMINDER SLIP
output type Hardcopy, 80 col. [ horz.perforation cols.22,44]
purpose To remind users for returning overdue books
generation Daily
copies [1]
distribution [1] respective user

<u_id> date =
<unam> collect before =
<uadd> .... text ....
<ucty> <titl> <auth> <ddue>

5.2.7 RESERVED BOOK COLLECTION REQUEST
output type Hardcopy, 80 col. [ horz.perforation cols.22,44]
purpose Requesting users to collect books they reserved
generation Daily
variants
copies [1]
distribution [1] respective users

<u_id> date =
<unam> collect before =
<uadd> .... TEXT ....
<ucty> <titl> <auth> <publ> <year> <call>

5.2.8 FINAL FINE REPORT
output type Hardcopy, 80 col. [ horz.perforation cols.22,44]
purpose Final fine report for clearance purposes
generation Annually
variants
copies [1] [2] [3]

<u_id> date of issue =
<unam> payment deadline =
<uadd> .... TEXT ....
<ucty> <fdue> =
5.3 VIDEO SCREEN DESIGN FOR ON-LINE INQUIRIES

On-line inquiries of various items of the data base are invoked by selecting INQUIRIES option from the main menu which causes the following menu to be displayed:

```
1. Orders
2. Books
3. Users
4. Main Menu
```

The user then chooses either orders [ form in article 5.3.1 ], books [ 5.3.2 ] or users [ 5.3.3 ]. For each of these options we specify:

a) Title [ a caption for the output ]

b) Purpose [ a short description of intended use ]

c) Generation [ when produced, e.g. daily, monthly, etc. ]

d) Access [ who are authorized to use the inquiry option ]

e) Two forms [ showing [1] layout of the input screen and [2] the output screen. ]

5.3.1 ORDERS INQUIRY

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Looking up orders information from the data base</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generation</td>
<td>As required</td>
</tr>
<tr>
<td>Access</td>
<td>Authorized library staff</td>
</tr>
<tr>
<td></td>
<td>&lt; inquiry screen &gt;</td>
</tr>
</tbody>
</table>

```
1. by supplier/Agent
2. by Department
3. by date_of_Order
4. by subj_code [pry.]
5. by subj_code [compo]
6. by Publisher
7. by report_gen_Flag
8. by Status_flag
9. Previous Menu
```
5.3.2

BOOK CATALOGUE INQUIRY

Purpose: Looking up book information from the data base

Generation: As required

Access: All users of the library

L-DBMS INQUIRIES * BOOK_STATUS

1. by Title
2. by Author
3. by Call number
4. by accession Number
5. by subject
6. by subj [composite]
7. by report_gen_Flag
8. by Status_flag
9. Previous Menu
### 5.3.3 USER STATUS INQUIRY

**Purpose:** Looking up user information from the database as required.

**Access:** Authorized library staff.

#### L-DBMS INQUIRIES * USER_STATUS LNU

1. by user_id_Number
2. by user_Type
3. by user_Name
4. by date_of_expiry
5. by user_City
6. by user_Department
7. by report_gen_flag
8. by status_flag
9. Previous Menu

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_id</td>
<td>XXXXXX</td>
</tr>
<tr>
<td>name</td>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
<tr>
<td>address</td>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
<tr>
<td>department</td>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
<tr>
<td>date_of_expiry</td>
<td>mm-dd-yy</td>
</tr>
<tr>
<td>borrowing_limit</td>
<td>nn</td>
</tr>
<tr>
<td>books_borrowed</td>
<td>nn</td>
</tr>
<tr>
<td>books_overdue</td>
<td>mm-dd-yy</td>
</tr>
<tr>
<td>publisher</td>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
<tr>
<td>remarks, if any</td>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
<tr>
<td>report_flag</td>
<td>X</td>
</tr>
<tr>
<td>status_flag</td>
<td>X</td>
</tr>
</tbody>
</table>
6. THE LOGICAL STRUCTURE OF THE SYSTEM'S OPERATION

The system is implemented as a collection of about sixty program modules written in dBASE programming language. These programs are accessed through a hierarchical menu system. A further twenty or so programs are used to support the nested menu structure. A tree diagram showing the organization of the program modules comprising our library data base management system appears in Figure 6.1. A brief description of the program modules is presented in Table 6.1.

The main menu [ generated by L.PRG ] is invoked by the dBASE command DO L and appears with three options -- inquiries [generated by LN.PRG], reports [ LR.PRG ] and updates [ LU.PRG ]. Menus at all levels follow a consistent pattern, with available options [ selectable by moving the light-bar or pressing the serial number of the desired item ] followed, as the last item, by an option to return to the previous level. The last item of the main menu returns the user to dBASE command prompt.

Programs appearing downmost in a 'path' [ please refer to Figure 6.1 ] are functional -- all others represent menus at various levels. Thus the main menu [ generated by program module L ] provides the user with the option of selecting any one of the next level menus [ LN for inquiries, LR for reports, LU for updates ]. These second level menus themselves provide path to yet other menus at a lower level. Third level options, too, are mostly menus. Programs at the fourth level [ those having 4-character names ] all perform one or another specific function.
FIGURE 6.1
TREES DIAGRAM SHOWING THE ORGANIZATION OF PROGRAM MODULES

[Intermediate modules represent menus, please refer to Table 6.1]
<table>
<thead>
<tr>
<th>Program function</th>
<th>DATABASE/</th>
<th>Program_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>module group</td>
<td>subsystem</td>
<td>description</td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>LN</td>
<td>INQUIRY</td>
<td></td>
</tr>
<tr>
<td>LNO</td>
<td>INQUIRY : ORDR</td>
<td></td>
</tr>
<tr>
<td>LNOO</td>
<td>INQUIRY : ORDR</td>
<td>by date placement of order</td>
</tr>
<tr>
<td>LNOP</td>
<td>INQUIRY : ORDR</td>
<td>by publisher</td>
</tr>
<tr>
<td>LNOQ</td>
<td>INQUIRY : ORDR</td>
<td>by requisitioning department</td>
</tr>
<tr>
<td>LNO2</td>
<td>INQUIRY : ORDR</td>
<td>by subject_code [composite]</td>
</tr>
<tr>
<td>LNOJ</td>
<td>INQUIRY : ORDR</td>
<td>by subject_code [primary]</td>
</tr>
<tr>
<td>LNOA</td>
<td>INQUIRY : ORDR</td>
<td>by supplier/agent</td>
</tr>
<tr>
<td>LNOF</td>
<td>INQUIRY : ORDR</td>
<td>by r_flag</td>
</tr>
<tr>
<td>LNOQ</td>
<td>INQUIRY : ORDR</td>
<td>by s_flag</td>
</tr>
<tr>
<td>LNB</td>
<td>INQUIRY : BOOK</td>
<td></td>
</tr>
<tr>
<td>LNBH</td>
<td>INQUIRY : BOOK</td>
<td>by accession number</td>
</tr>
<tr>
<td>LNEA</td>
<td>INQUIRY : BOOK</td>
<td>by author</td>
</tr>
<tr>
<td>LNBC</td>
<td>INQUIRY : BOOK</td>
<td>by call number</td>
</tr>
<tr>
<td>LNB2</td>
<td>INQUIRY : BOOK</td>
<td>by subject_code [composite]</td>
</tr>
<tr>
<td>LNBJ</td>
<td>INQUIRY : BOOK</td>
<td>by subject_code [primary]</td>
</tr>
<tr>
<td>LNBK</td>
<td>INQUIRY : BOOK</td>
<td>by title</td>
</tr>
<tr>
<td>LNBF</td>
<td>INQUIRY : BOOK</td>
<td>by r_flag</td>
</tr>
<tr>
<td>LNBG</td>
<td>INQUIRY : BOOK</td>
<td>by s_flag</td>
</tr>
<tr>
<td>LNU</td>
<td>INQUIRY : USER</td>
<td></td>
</tr>
<tr>
<td>LNUX</td>
<td>INQUIRY : USER</td>
<td>by date of expiry</td>
</tr>
<tr>
<td>LNUC</td>
<td>INQUIRY : USER</td>
<td>by user_city</td>
</tr>
<tr>
<td>LNUO</td>
<td>INQUIRY : USER</td>
<td>by user_department</td>
</tr>
<tr>
<td>LNUC</td>
<td>INQUIRY : USER</td>
<td>by user_id_number</td>
</tr>
<tr>
<td>LNUA</td>
<td>INQUIRY : USER</td>
<td>by user_name</td>
</tr>
<tr>
<td>LNUT</td>
<td>INQUIRY : USER</td>
<td>by user_type</td>
</tr>
<tr>
<td>LNUF</td>
<td>INQUIRY : USER</td>
<td>by r_flag</td>
</tr>
<tr>
<td>LNUQ</td>
<td>INQUIRY : USER</td>
<td>by s_flag</td>
</tr>
<tr>
<td>LR</td>
<td>REPORTS</td>
<td></td>
</tr>
<tr>
<td>LRA</td>
<td>REPORTS : acqn</td>
<td></td>
</tr>
<tr>
<td>LRAO</td>
<td>REPORTS : acqn</td>
<td>orders to suppliers</td>
</tr>
<tr>
<td>LRAD</td>
<td>REPORTS : acqn</td>
<td>books ordered, by dept.</td>
</tr>
<tr>
<td>LRAS</td>
<td>REPORTS : acqn</td>
<td>books ordered, by supp.</td>
</tr>
<tr>
<td>LRAR</td>
<td>REPORTS : acqn</td>
<td>new arrivals, by record#</td>
</tr>
<tr>
<td>LRAA</td>
<td>REPORTS : acqn</td>
<td>new arrivals, by author</td>
</tr>
<tr>
<td>LRAJ</td>
<td>REPORTS : acqn</td>
<td>new arrivals, by subject</td>
</tr>
<tr>
<td>LRAT</td>
<td>REPORTS : acqn</td>
<td>new arrivals, by title</td>
</tr>
<tr>
<td>LRAQ</td>
<td>REPORTS : acqn</td>
<td>books unavailable</td>
</tr>
</tbody>
</table>

continued...
<table>
<thead>
<tr>
<th>TABLE 6.1</th>
<th>DATABASE/ subsystem description</th>
<th>Program_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>program function</td>
<td>Module group description</td>
<td></td>
</tr>
<tr>
<td>LRR</td>
<td>REPORTS : rent</td>
<td></td>
</tr>
<tr>
<td>LRRRL</td>
<td>reports : rent</td>
<td></td>
</tr>
<tr>
<td>LRRR</td>
<td>books still on loan</td>
<td>2-2-1</td>
</tr>
<tr>
<td>LRRR</td>
<td>books issued</td>
<td>2-2-2</td>
</tr>
<tr>
<td>LRRR</td>
<td>books returned</td>
<td>2-2-3</td>
</tr>
<tr>
<td>LRRS</td>
<td>overdue reminder _students</td>
<td>2-2-4</td>
</tr>
<tr>
<td>LRRT</td>
<td>overdue reminder _teachers</td>
<td>2-2-5</td>
</tr>
<tr>
<td>LRRV</td>
<td>reserve_book coll. slip</td>
<td>2-2-6</td>
</tr>
<tr>
<td>LRRF</td>
<td>final fine report</td>
<td>2-2-7</td>
</tr>
<tr>
<td>LU</td>
<td>UPDATES</td>
<td></td>
</tr>
<tr>
<td>LUR</td>
<td>reports : rent</td>
<td></td>
</tr>
<tr>
<td>LURR</td>
<td>record return of a book</td>
<td>3-1-2</td>
</tr>
<tr>
<td>LURV</td>
<td>reserve a book</td>
<td>3-1-3</td>
</tr>
<tr>
<td>LUR</td>
<td>transfer to BOOK.DBF</td>
<td>3-2-0</td>
</tr>
<tr>
<td>LUD</td>
<td>UPDATES : DEPT</td>
<td></td>
</tr>
<tr>
<td>LUDA</td>
<td>add</td>
<td>3-3-1</td>
</tr>
<tr>
<td>LUDC</td>
<td>change</td>
<td>3-3-2</td>
</tr>
<tr>
<td>LUDD</td>
<td>delete</td>
<td>3-3-3</td>
</tr>
<tr>
<td>LUO</td>
<td>UPDATES : ORDR</td>
<td></td>
</tr>
<tr>
<td>LUOA</td>
<td>add</td>
<td>3-4-1</td>
</tr>
<tr>
<td>LUOC</td>
<td>change</td>
<td>3-4-2</td>
</tr>
<tr>
<td>LUOD</td>
<td>delete</td>
<td>3-4-3</td>
</tr>
<tr>
<td>LUP</td>
<td>UPDATES : PUBL</td>
<td></td>
</tr>
<tr>
<td>LUPA</td>
<td>add</td>
<td>3-5-1</td>
</tr>
<tr>
<td>LUPC</td>
<td>change</td>
<td>3-5-2</td>
</tr>
<tr>
<td>LUPD</td>
<td>delete</td>
<td>3-5-3</td>
</tr>
<tr>
<td>LUS</td>
<td>UPDATES : SUPP</td>
<td></td>
</tr>
<tr>
<td>LUSA</td>
<td>add</td>
<td>3-6-1</td>
</tr>
<tr>
<td>LUSC</td>
<td>change</td>
<td>3-6-2</td>
</tr>
<tr>
<td>LUSD</td>
<td>delete</td>
<td>3-6-3</td>
</tr>
<tr>
<td>LUU</td>
<td>UPDATES : USER</td>
<td></td>
</tr>
<tr>
<td>LUUA</td>
<td>add</td>
<td>3-7-1</td>
</tr>
<tr>
<td>LUUC</td>
<td>change</td>
<td>3-7-2</td>
</tr>
<tr>
<td>LUUD</td>
<td>delete</td>
<td>3-7-3</td>
</tr>
</tbody>
</table>
The menus offer choices at the same level under the same 'parent'. That is a user cannot branch to another menu, even if it is at the same level, if the latter is a 'child' to a different parent. For example, the inquiries menu [ LN ] has three 'child' menus [ LNO, LNB, LNU ] and the reports menu [ LR ] has two 'child' menus [ LRA, LRR ]. A user currently under LN can choose either of LNO, LNB or LNU but cannot choose LRA or LRR, even though the latter too belong to the same third level. The user will have to return to L, and then select LR to have access to the LRA or LRR options.

A brief description of the 'functional' program modules of each of the function groups [ inquiries, reports and updates ] comprises the rest of this chapter. We use a semi-tabular form which specifies, for each program module listed, the following:

a) program_id and the name of the file containing the program
b) Function [ specific task that the program module performs ]
c) input [ required user input ]
d) data base used [ .DBF files accessed. Following each filename is indicated (1) whether the program only reads the file [r ] or alters it [r/w], and (2) name of the index file if one requires to be opened along with the data base.]

6.1 INQUIRING THE DATA BASE

<table>
<thead>
<tr>
<th>L-DBMS INQUIRIES</th>
<th>LN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Orders</td>
<td></td>
</tr>
<tr>
<td>2. Books</td>
<td></td>
</tr>
<tr>
<td>3. Users</td>
<td></td>
</tr>
<tr>
<td>4. Main Menu</td>
<td></td>
</tr>
</tbody>
</table>

This function group comprise of programs performing on-line inquiry into the orders, books and user data bases. Programs in this group do not generate hardcopy printouts -- nor do they
modify any of the files accessed. Inquiries are supported on
(1) orders, (2) books, and (3) users data bases, which appear as
options in the INQUIRIES menu. Functional programs in this
group are:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Input</th>
<th>Data Base</th>
<th>Index Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-1-1-1</td>
<td>Inquire orders by date of placement of order</td>
<td>date</td>
<td>ORD.R.DBF</td>
<td>O_DORD.NDX</td>
</tr>
<tr>
<td>6-1-1-2</td>
<td>Inquire orders data base by publisher</td>
<td>publisher_id</td>
<td>ORD.R.DBF</td>
<td>O_P_ID.NDX</td>
</tr>
<tr>
<td>6-1-1-3</td>
<td>Inquire orders data base by requisitioning dept.</td>
<td>department_id</td>
<td>ORD.R.DBF</td>
<td>O_D_ID.NDX</td>
</tr>
<tr>
<td>6-1-1-4</td>
<td>Inquire orders data base by comp. subject code</td>
<td>subj</td>
<td>ORD.R.DBF</td>
<td>O_SUBJ.NDX</td>
</tr>
<tr>
<td>6-1-1-5</td>
<td>Inquire orders data base by pry. subject code</td>
<td>subject_code</td>
<td>ORD.R.DBF</td>
<td>O_SUB2.NDX</td>
</tr>
<tr>
<td>6-1-1-6</td>
<td>Inquire orders data base by supplier/agent</td>
<td>supplier_id</td>
<td>ORD.R.DBF</td>
<td>O_S_ID.NDX</td>
</tr>
<tr>
<td>6-1-1-7</td>
<td>Display record(s) having &lt;flag&gt; = input</td>
<td>flag</td>
<td>ORD.R.DBF</td>
<td>O_FLAG.NDX</td>
</tr>
<tr>
<td>6-1-1-8</td>
<td>Display record(s) having &lt;stat&gt; = input</td>
<td>stat</td>
<td>ORD.R.DBF</td>
<td>O_STAT.NDX</td>
</tr>
<tr>
<td>Function Number</td>
<td>Procedure Name</td>
<td>Function Description</td>
<td>Input Parameters</td>
<td>Database Used</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------</td>
<td>-------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>6-1-2-1</td>
<td>LNBN.prg</td>
<td>Inquire book by accession number</td>
<td>accession number</td>
<td>BOOK.DBF [r ]</td>
</tr>
<tr>
<td>6-1-2-2</td>
<td>LNBA.prg</td>
<td>Inquire book database by author</td>
<td>publisher_id</td>
<td>BOOK.DBF [r ]</td>
</tr>
<tr>
<td>6-1-2-3</td>
<td>LNBC.prg</td>
<td>Inquire book database by call number</td>
<td>call number</td>
<td>BOOK.DBF [r ]</td>
</tr>
<tr>
<td>6-1-2-4</td>
<td>LNB2.prg</td>
<td>Inquire book database by comp. subject code</td>
<td>&lt;subj&gt; and &lt;sub2&gt;</td>
<td>BOOK.DBF [r ]</td>
</tr>
<tr>
<td>6-1-2-5</td>
<td>LNBJ.prg</td>
<td>Inquire book database by subj. subject code</td>
<td>subject code</td>
<td>BOOK.DBF [r ]</td>
</tr>
<tr>
<td>6-1-2-6</td>
<td>LNBA.prg</td>
<td>Inquire book database by supplier/agent</td>
<td>supplier_id</td>
<td>BOOK.DBF [r ]</td>
</tr>
<tr>
<td>6-1-2-7</td>
<td>LNBF.prg</td>
<td>Display record(s) having &lt;flag&gt; = input</td>
<td>Any character in the set</td>
<td>BOOK.DBF [r ]</td>
</tr>
<tr>
<td>6-1-2-8</td>
<td>LNBS.prg</td>
<td>Display record(s) having &lt;stat&gt; = input</td>
<td>Any character in the set</td>
<td>BOOK.DBF [r ]</td>
</tr>
<tr>
<td>6-1-3-1</td>
<td>LNUX.prg</td>
<td>Inquire users by date of expiry</td>
<td>date</td>
<td>USER.DBF [r ]</td>
</tr>
<tr>
<td>6-1-3-2</td>
<td>LNUC.prg</td>
<td>Inquire users database by user_city</td>
<td>name of city</td>
<td>USER.DBF [r ]</td>
</tr>
<tr>
<td>6-1-3-3</td>
<td>LNUD.prg</td>
<td>function Inquire users data base by requisitioning dept.</td>
<td>department_id</td>
<td>USER.DBF [r ]</td>
</tr>
<tr>
<td>6-1-3-4</td>
<td>LNUN.prg</td>
<td>function Inquire users data base by user_id number</td>
<td>&lt;subj&gt; &lt;sub2&gt;</td>
<td>USER.DBF [r ]</td>
</tr>
<tr>
<td>6-1-3-5</td>
<td>LNUA.prg</td>
<td>function Inquire users data base by user name</td>
<td>user name</td>
<td>USER.DBF [r ]</td>
</tr>
<tr>
<td>6-1-3-6</td>
<td>LNUT.prg</td>
<td>function Inquire users data base by user type</td>
<td>supplier_id</td>
<td>USER.DBF [r ]</td>
</tr>
<tr>
<td>6-1-3-7</td>
<td>LNUF.prg</td>
<td>function Display record(s) having &lt;flag&gt; = input</td>
<td>Any character in the set @ABCDEFGHIJKLMNO</td>
<td>USER.DBF [r ]</td>
</tr>
<tr>
<td>6-1-3-8</td>
<td>LNUS.prg</td>
<td>function Display record(s) having &lt;stat&gt; = input</td>
<td>Any character in the set @ABCDEFGHIJKLMNO</td>
<td>USER.DBF [r ]</td>
</tr>
</tbody>
</table>

### 6.2 GENERATING REPORTS

**L-DBMS REPORTS**

1. Acquisitions  
2. Rental Management  
3. Main Menu

Program modules in this group generate various hardcopy reports. With the exception of the <flag> field of certain data bases, all other information in the files accessed remain unaltered. Report generation operations can be viewed as belonging to two areas: (1) those related to book acquisitions, and (2) those related to rental management. These appear as options in the
### REPORTS menu

Functional programs in this group are:

<table>
<thead>
<tr>
<th>Function ID</th>
<th>Program</th>
<th>Description</th>
<th>Input</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-2-1-1</td>
<td>LRAO.prg</td>
<td>Prints requisition to individual suppliers</td>
<td>none [ covers all new entries in ORDR.DBF ]</td>
<td>ORD.R.DBF [r/w] index used: O_S.ID.NDX [r ] SUPP.DBF [r ] index used: S_S.ID.NDX [r ]</td>
</tr>
<tr>
<td>6-2-1-2</td>
<td>LRAD.prg</td>
<td>Prints list of books ordered [ sorted by dept. ]</td>
<td>none [ covers all new entries in ORDR.DBF ]</td>
<td>ORDR.DBF [r/w] index used: O_D.ID.NDX [r ]</td>
</tr>
<tr>
<td>6-2-1-3</td>
<td>LRAS.prg</td>
<td>Prints list of books ordered [ sorted by supp. ]</td>
<td>none [ covers all new entries in ORDR.DBF ]</td>
<td>ORDR.DBF [r/w] index used: O_S.ID.NDX [r ]</td>
</tr>
<tr>
<td>6-2-1-4</td>
<td>LRAR.prg</td>
<td>Prints list of new arrivals [ sorted by recnum ]</td>
<td>none [ covers all new receipts ]</td>
<td>ORDR.DBF [r/w] index used:</td>
</tr>
<tr>
<td>6-2-1-5</td>
<td>LRAA.prg</td>
<td>Prints list of new arrivals [ sorted by author ]</td>
<td>none [ covers all new receipts ]</td>
<td>ORDR.DBF [r/w] index used: O_AUTH.NDX [r ]</td>
</tr>
<tr>
<td>6-2-1-6</td>
<td>LRAJ.prg</td>
<td>Prints list of new arrivals [ sorted by subject ]</td>
<td>none [ covers all new receipts ]</td>
<td>ORDR.DBF [r/w] index used: O_SUBJ.NDX [r ]</td>
</tr>
<tr>
<td>6-2-1-7</td>
<td>LRAT.prg</td>
<td>Prints list of new arrivals [ sorted by title ]</td>
<td>none [ covers all new receipts ]</td>
<td>ORDR.DBF [r/w] index used: O_TITL.NDX [r ]</td>
</tr>
<tr>
<td>6-2-1-8</td>
<td>LRAU.prg</td>
<td>Prints list of books unavailable</td>
<td>none [ covers current orders and receipts ]</td>
<td>ORDR.DBF [r/w] index used: O_D.ID.NDX [r ]</td>
</tr>
<tr>
<td>6-2-2-1</td>
<td>LRRL.prg</td>
<td>Prints list of books currently on loan</td>
<td>none [ all books out on loan covered ]</td>
<td>BOOK.DBF [r ] index used: SUBJ.NDX [r ]</td>
</tr>
</tbody>
</table>
### 6-2-2-2
**Function**: LRRR.prg  
**Input data base**: BOOK.DBF [r ]  
**Index used**: ACCN.NDX [r ]  
**Description**: Prints list of books issued on given date

### 6-2-2-3
**Function**: LRRR.prg  
**Input data base**: BOOK.DBF [r ]  
**Index used**: ACCN.NDX [r ]  
**Description**: Prints list of books returned on given date

### 6-2-2-4
**Function**: LRRS.prg  
**Input data base**: BOOK.DBF [r ]  
**Index used**: L_ACN.NDX [r ]  
**Description**: Prints overdue reminder slip [for students]

### 6-2-2-5
**Function**: LRRT.prg  
**Input data base**: BOOK.DBF [r ]  
**Index used**: L_U_ID.NDX [r ]  
**Description**: Prints overdue reminder slip [for teachers]

### 6-2-2-6
**Function**: LRRV.prg  
**Input data base**: BOOK.DBF [r ]  
**Index used**: R_UCN.NDX [r ]  
**Description**: Prints reserved book collection request

### 6-2-2-7
**Function**: LRFF.prg  
**Input data base**: USER.DBF [r ]  
**Index used**: U_U_ID.NDX [r ]  
**Description**: Prints final fine report

### 6.3 UPDATES AND MAINTENANCE

#### L-DBMS UPDATE DATABASES

1. Rental  
2. Transfer  
3. Departments  
4. Orders  
5. Publishers  
6. Suppliers  
7. Users  
8. Main menu
All operations that alter the current contents of one or more data base file have been classified in this function group.

Update operations can be viewed as belonging to three categories: (1) addition / alteration / deletion of records in the orders, departments, publishers, suppliers, and users data bases, (2) periodic transfer of records from the orders to the permanent book data base, and (3) routine rental operations involving issue, return, and reservation of books. All these appear as options in the UPDATES sub-menu. Functional programs in this group are:

<table>
<thead>
<tr>
<th>Function</th>
<th>Input</th>
<th>Data Base</th>
</tr>
</thead>
<tbody>
<tr>
<td>LURl.prg</td>
<td>Record issue of book, print 2-part issue card</td>
<td>BOOK(DBF [r/w] index used: ACCN.NDX [r ]</td>
</tr>
<tr>
<td></td>
<td>user_id of borrower, accession number of book</td>
<td>LOAN(DBF [r/w] index used: Lendon.NDX [r/w]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USER(DBF [r/w] index used: U_U_ID.NDX [r ]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Input</th>
<th>Data Base</th>
</tr>
</thead>
<tbody>
<tr>
<td>LURR.prg</td>
<td>Record return of book, print 2-part issue card</td>
<td>BOOK(DBF [r/w] index used: ACCN.NDX [r ]</td>
</tr>
<tr>
<td></td>
<td>user_id of borrower, accession number of book</td>
<td>LOAN(DBF [r/w] index used: Lendon.NDX [r/w]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USER(DBF [r/w] index used: U_U_ID.NDX [r ]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Input</th>
<th>Data Base</th>
</tr>
</thead>
<tbody>
<tr>
<td>LURV.prg</td>
<td>Reserve a book</td>
<td>BOOK(DBF [r/w] index used: ACCN.NDX [r ]</td>
</tr>
<tr>
<td></td>
<td>user_id of borrower, accession number of book</td>
<td>RQVE(DBF [r/w] index used: Rendon.NDX [r/w]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USER(DBF [r/w] index used: U_U_ID.NDX [r ]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Input</th>
<th>Data Base</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUT.prg</td>
<td>Transfer records having &lt;stat&gt;=input to BOOK.DBF</td>
<td>ORDR(DBF [r/w] index used: O_????.NDX [r/w]</td>
</tr>
<tr>
<td></td>
<td>appropriate status code</td>
<td>BOOK(DBF [r/w] index used: ????.NDX [r/w]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Input</th>
<th>Data Base</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUDA.prg</td>
<td>add</td>
<td>[ input : none \</td>
</tr>
<tr>
<td>LUDC.prg</td>
<td>change</td>
<td>[ input : department_id</td>
</tr>
<tr>
<td>LUDD.prg</td>
<td>delete</td>
<td>[ input : department_id</td>
</tr>
<tr>
<td>DEPT(DBF [r/w] index used: D_????.NDX [r/w]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>6-3-4-1</strong></td>
<td><strong>LUOA.prg</strong></td>
<td><strong>add</strong></td>
</tr>
<tr>
<td><strong>6-3-4-2</strong></td>
<td><strong>LUOC.prg</strong></td>
<td><strong>change</strong></td>
</tr>
<tr>
<td><strong>6-3-4-3</strong></td>
<td><strong>LUOD.prg</strong></td>
<td><strong>delete</strong></td>
</tr>
<tr>
<td><strong>function</strong></td>
<td>Modify data base</td>
<td></td>
</tr>
<tr>
<td><strong>data base</strong></td>
<td>ORDR.DBF [r/w]</td>
<td>index used: O_????NDX [r/w]</td>
</tr>
</tbody>
</table>

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>6-3-5-1</strong></td>
<td><strong>LUPA.prg</strong></td>
<td><strong>add</strong></td>
<td>[ input : none ]</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>6-3-5-2</strong></td>
<td><strong>LUFC.prg</strong></td>
<td><strong>change</strong></td>
<td>[ input : publisher_id ]</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>6-3-5-3</strong></td>
<td><strong>LUPD.prg</strong></td>
<td><strong>delete</strong></td>
<td>[ input : publisher_id ]</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>function</strong></td>
<td>Modify data base</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>data base</strong></td>
<td>PUBL.DBF [r/w]</td>
<td>index used: P_????NDX [r/w]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>6-3-6-1</strong></td>
<td><strong>LUSA.prg</strong></td>
<td><strong>add</strong></td>
<td>[ input : none ]</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>6-3-6-2</strong></td>
<td><strong>LUSC.prg</strong></td>
<td><strong>change</strong></td>
<td>[ input : supplier_id ]</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>6-3-6-3</strong></td>
<td><strong>LUSD.prg</strong></td>
<td><strong>delete</strong></td>
<td>[ input : supplier_id ]</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>function</strong></td>
<td>Modify data base</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>data base</strong></td>
<td>SUPP.DBF [r/w]</td>
<td>index used: S_????NDX [r/w]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>6-3-7-1</strong></td>
<td><strong>LUAU.prg</strong></td>
<td><strong>add</strong></td>
<td>[ input : none ]</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>6-3-7-2</strong></td>
<td><strong>LUUC.prg</strong></td>
<td><strong>change</strong></td>
<td>[ input : user_id ]</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>6-3-7-3</strong></td>
<td><strong>LUUD.prg</strong></td>
<td><strong>delete</strong></td>
<td>[ input : user_id ]</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>function</strong></td>
<td>Modify data base</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>data base</strong></td>
<td>USER.DBF [r/w]</td>
<td>index used: U_????NDX [r/w]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7. IMPLEMENTATION

7.1 TEST-RUN OF THE SYSTEM

The system has been tested using a book data base containing 1024 records. This test data base contains particulars of 'real' books held at the BUET Computer Centre Library and BUET Rental Library. However, to provide the required degree of range and variation in the values, hypothetical figures were used for the call number <call> and accession number <accn> fields and for other fields where relevant data did not exist. Smaller supporting data bases [e.g. ORDR, DEPT, PUBL, SUPP, USER] were created with hypothetical data. During the limited amount of testing that could be undertaken, given obvious time and data constraints, we found the system working reasonably satisfactorily. Some of the outputs generated by the system appear in the appendices.

7.2 LIMITATIONS AND POSSIBLE ENHANCEMENTS

The primary limitation of the system presented here, and one severely restricting its fruitful use in a real library environment is that it has been developed as a single user system. This means that at any one time the system can serve only one user -- which clearly does not satisfy the demands of a heavy-use university library data base whose predominant function tends to be servicing user queries regarding books available. The preferred solution could be some form of a shared data base which could be simultaneously accessed by a number of terminals [or network supported PCs]. This feature however first calls for a data base management software capable of shared file handling which dBASE III Plus is, at its present form, unable to do.
There are two possible alternatives to overcome this limitation. First, and most obvious, is to switch to a data base management software with built-in simultaneous file access capability (e.g. R:BASE, Informix, Oracle, Paradox, etc.). However, given the popularity of dBASE III Plus, and the resultant availability of trained manpower and third-party books, it is probably advisable to stick to it. An improvised, and admittedly less than ideal, solution could be to use as many copies of the BOOK data base as there are PCs available. Updates, of necessity, have to be restricted to one computer which will contain the 'master version' of the data base and will serve all update operations including issue and return of books. The other computers will just support user queries. Whenever the BOOK data base is altered, the updated version will be copied to the 'slave' computers. An undesirable facet of such a scheme is that persons performing inquiries on a book cannot get on-line information about the book's current status, nor can they reserve it from the inquiring PCs.

This deficiency notwithstanding, just a versatile computerized substitute of the traditional card-based catalogue inquiry system can be viewed as enough benefit to warrant the adoption of such a system. A further issue that can crop up should the data base become very large is the speed of processing queries. dBASE's processing, especially where 'joins' of two or more files are involved, can be terribly slow, and thus nullify much of the advantages of a computerized catalogue inquiry system. It is possible that speed considerations may require the catalogue inquiry routines to be written in a high-level language, bypassing dBASE altogether. Such data bases may use an ASCII text copy of the data base,
preparing and using its own set of indices. It may even be necessary to allow some redundancy in the data records in order to eliminate expensive 'join' operations.

The issue-return operations can be made more effective by using some form of OCR [optical character recognition] or OMR [optical mark recognition, e.g. bar-code reader] device for inputting the book and borrower numbers. Manual entry of numbers can be slow and error-prone. If the system is to rely on manual entry, it would be advisable to have check-digit protection for both the accession numbers of books and ids of users.

7.3 IMPLEMENTATION CONSIDERATIONS

About implementation, the easiest one to commission is the acquisitions subsystem, requiring little other than a PC and the software already developed here. Commissioning the library catalogue inquiry system requires, first, entry of data on the entire existing collection of the library, plus additional software to circumvent the concurrent access problems discussed earlier. Computerizing book issue and returns is perhaps the most difficult to implement for two reasons. First, it represents the replacement of a system and thus requires enough caution in testing/debugging and training of personnel. The second requirement is the preparation (e.g. labeling books with accession numbers in OCR compatible labels or bar-codes, etc.) of ALL the books before the system may be commissioned. Given these considerations, the immediate application of a computerized system for any university library in this country is likely to be confined to the acquisition operations. On-line inquiry and computerized management of book rental operations require more facilities than are presently available. So these
uses have to wait.

In the design and development of a system of this level of complexity some imperfections are necessarily almost inevitable. It is therefore recommended, too, that any ideas of using the library data base management system developed here for real world application should provide for exhaustive testing of the robustness of the software before actually commissioning it for library use. There is uncertainty too regarding the possible behaviour of the system when handling the massive volume of data of a typical library which could be 50 or more times that of the test. It would probably be wise not to exclude the possibility of extensive revision of the software in case such problems reach unacceptable proportions.

These reservations notwithstanding we believe that our system, even in its present form, can be a good starting point for launching a gradual computerization program in any library that plans to move in this direction.
REFERENCES


APPENDIX

| program source code listing, identified by filename |

```
set talk off
set proc to LBAR
optn = 4
os = 0

while os # optn
    opt1 = "INQUIRY"
    opt2 = "REPORTS"
    opt3 = "UPDATES"
    opt4 = ",..., Quit to dBASE"
    os = 4
enddo

set procedure to LBAR
optn = 4
os = 0

while os # optn
    opt1 = "Orders" optn = 4
    opt2 = "Books"
    opt3 = "Users"
    opt4 = "Main Menu"
    os = 4
enddo

return
```

```
set talk off
set proc to LBAR
optn = 4
os = 0

do while os # optn
    opt1 = "INQUIRY"
    opt2 = "REPORTS"
    opt3 = "UPDATES"
    opt4 = ",..., Quit to dBASE"
    os = 4
enddo

titl = "L-DBMS LIBRARY DATA BASE MANAGEMENT SYSTEM"
clear

-do LBAR with optn,titl && invoke light_bar menu
    do case
        case os = 1
            do LM && invoke inquiries menu
        case os = 2
            do LR && invoke reports menu
        case os = 3
            do LU && invoke updates menu
        endcase
    enddo

clear && clear screen
```

```
set procedure to LBAR
optn = 4
os = 0

while os # optn
    opt1 = "Orders"
    opt2 = "Books"
    opt3 = "Users"
    opt4 = "Main Menu"
    os = 4
enddo

titl = "L-DBMS INQUIRIES"
clear

-do LBAR with optn,titl && invoke light_bar menu
    do case
        case os = 1
            do LNO && orders inquiry
        case os = 2
            do LMB && book inquiry
        case os = 3
            do LNU && users inquiry
        endcase
    enddo

clear && clear screen
```

```
set procedure to LBAR
optn = 4
os = 0

while os # optn
    opt1 = "Orders"
    opt2 = "Books"
    opt3 = "Users"
    opt4 = "Main Menu"
    os = 4
enddo

titl = "L-DBMS INQUIRIES"
clear

-do LBAR with optn,titl && invoke light_bar menu
    do case
        case os = 1
            do LNO && orders inquiry
        case os = 2
            do LMB && book inquiry
        case os = 3
            do LNU && users inquiry
        endcase
    enddo

clear && clear screen
```

```
set talk off
set proc to LBAR
optn = 4
os = 0

while os # optn
    opt1 = "INQUIRY"
    opt2 = "REPORTS"
    opt3 = "UPDATES"
    opt4 = ",..., Quit to dBASE"
    os = 4
enddo

titl = "L-DBMS LIBRARY DATA BASE MANAGEMENT SYSTEM"
clear

-do LBAR with optn,titl && invoke light_bar menu
    do case
        case os = 1
            do LM && invoke inquiries menu
        case os = 2
            do LR && invoke reports menu
        case os = 3
            do LU && invoke updates menu
        endcase
    enddo

clear && clear screen
```

```
set procedure to LBAR
optn = 4
os = 0

while os # optn
    opt1 = "Orders"
    opt2 = "Books"
    opt3 = "Users"
    opt4 = "Main Menu"
    os = 4
enddo

titl = "L-DBMS INQUIRIES"
clear

-do LBAR with optn,titl && invoke light_bar menu
    do case
        case os = 1
            do LNO && orders inquiry
        case os = 2
            do LMB && book inquiry
        case os = 3
            do LNU && users inquiry
        endcase
    enddo

clear && clear screen
```

```
set talk off
set proc to LBAR
optn = 4
os = 0

while os # optn
    opt1 = "INQUIRY"
    opt2 = "REPORTS"
    opt3 = "UPDATES"
    opt4 = ",..., Quit to dBASE"
    os = 4
enddo

titl = "L-DBMS LIBRARY DATA BASE MANAGEMENT SYSTEM"
clear

-do LBAR with optn,titl && invoke light_bar menu
    do case
        case os = 1
            do LM && invoke inquiries menu
        case os = 2
            do LR && invoke reports menu
        case os = 3
            do LU && invoke updates menu
        endcase
    enddo

clear && clear screen
```

```
set procedure to LBAR
optn = 4
os = 0

while os # optn
    opt1 = "Orders"
    opt2 = "Books"
    opt3 = "Users"
    opt4 = "Main Menu"
    os = 4
enddo

titl = "L-DBMS INQUIRIES"
clear

-do LBAR with optn,titl && invoke light_bar menu
    do case
        case os = 1
            do LNO && orders inquiry
        case os = 2
            do LMB && book inquiry
        case os = 3
            do LNU && users inquiry
        endcase
    enddo

clear && clear screen
```

```
set talk off
set proc to LBAR
optn = 4
os = 0

while os # optn
    opt1 = "INQUIRY"
    opt2 = "REPORTS"
    opt3 = "UPDATES"
    opt4 = ",..., Quit to dBASE"
    os = 4
enddo

titl = "L-DBMS LIBRARY DATA BASE MANAGEMENT SYSTEM"
clear

-do LBAR with optn,titl && invoke light_bar menu
    do case
        case os = 1
            do LM && invoke inquiries menu
        case os = 2
            do LR && invoke reports menu
        case os = 3
            do LU && invoke updates menu
        endcase
    enddo

clear && clear screen
```

```
set procedure to LBAR
optn = 4
os = 0

while os # optn
    opt1 = "Orders"
    opt2 = "Books"
    opt3 = "Users"
    opt4 = "Main Menu"
    os = 4
enddo

titl = "L-DBMS INQUIRIES"
clear

-do LBAR with optn,titl && invoke light_bar menu
    do case
        case os = 1
            do LNO && orders inquiry
        case os = 2
            do LMB && book inquiry
        case os = 3
            do LNU && users inquiry
        endcase
    enddo

clear && clear screen
```

```
set talk off
set proc to LBAR
optn = 4
os = 0

while os # optn
    opt1 = "INQUIRY"
    opt2 = "REPORTS"
    opt3 = "UPDATES"
    opt4 = ",..., Quit to dBASE"
    os = 4
enddo

titl = "L-DBMS LIBRARY DATA BASE MANAGEMENT SYSTEM"
clear

-do LBAR with optn,titl && invoke light_bar menu
    do case
        case os = 1
            do LM && invoke inquiries menu
        case os = 2
            do LR && invoke reports menu
        case os = 3
            do LU && invoke updates menu
        endcase
    enddo

clear && clear screen
```
```plaintext
set procedure to LBAR & book inquiry menu
optn = 9
os = 0
    do while os # optn
        opt1 = "by Title"
        opt2 = "by Author"
        opt3 = "by Call number"
        opt4 = "by accession Number"
        opt5 = "by subject"
        opt6 = "by subj [composite]"
        opt7 = "by report_gen_Flag"
        opt8 = "by Status_flag"
        opt9 = "Previous Menu"
        optn = 9
    enddo
    os = 0
return & & previous menu
```

```plaintext
LNB2.PRG
set talk off & & inquiry on composite subject
set stat off & & clear screen
clea @ 01,09 to 03,78 & & draw box for header
@ 02,15 say "LNB3 - book inquiry by subject code [ composite ]"
select A
    use BOOK inde sub2 & & use index on composite subject
select B
    use PUBL inde p_p_id & & use index on publisher_id
select A
    set rela to p_id into PUBL & & perform logical join
    m_subp = "?" & & primary code inquired
    m_next = "?" & & continuation code ( X exits
    do while m_subp # ""
        m_subp = space(3): & & initialize memory variables
        m_subs = space(3)
        m_sub2 = space(6)
    @ 04,00 clea & & clear screen display area
```
say "<cr> to exit..."
@ 12,01 say "? subject code [ primary ] " get m_subp
read
if m_subp = " " && exit if no choice entered
loop
endif
@ 13,01 say "? subject code [ secondary ] " get m_subs
read
& & get secondary code
m_sub2 = upper(m_subp)+upper(m_subs)
seek m_sub2
if found()
do while upper(subj)+upper(sub2)=m_sub2
skip
@end 23,02
wait "[ press X to interrupt ] " -- More -- " to m_next
if upper(m_next) = 'X'
clea
exit
endif
else
say "& No such subject code combination.".
endif
@ 23,02
wait && wait until key pressed
endo
use && close data bases
set stat on
return && previous menu

set stat off && inquiry on author_name
set talk off
 clea && clear screen
@ 01,09 to 03,78 && draw box for header
@ 02,25 say "LNBA - book inquiry by author"
select A
use BOOK inde auth && use index on author_name
select B
use PUBL inde p_p_id && use index on publisher_id
select A
set rela to p_id into PURL
m_auth = "?" && author_name inquired
m_next = "?" && continuation code ( X exits
do while m_auth # " ".
m_auth = space(32) && initialize memory variable
clea && clear screen display area
@ 04,00
@ 14,13 say "<cr> to exit..."
@ 12,01 say "? author " get m_auth
read
if m_auth = " " && exit if no entry
loop
endif
seek upper(m_auth) && search and display if found
if found()
do while auth=upper(m_auth)
skip
@end 23,02
wait "[ press X to interrupt ]". More "" to m_next
if upper(m_next) = 'X'
clea
exit
endif
endif
e14,13
say "No such author."
endif
@ 23,02
wait
endo
dlo
de
use
set stat on
return
& & wait until key pressed
& & close data bases
& & previous menu
set
stat off
& & inquiry on call number
set
talk off
& & clear screen
clea
@ 01,00 to 03,78
& & draw box for header
@ 02,25
say "LNBC - book inquiry by call number"
select A
use BOOK inde call & & use index on call number
select B
use PUBL inde p_p_id & & use index on publisher_id
select A
set rela to p_id into PUBL & & perform logical join
m_call = "?"
do while m_call # ""
  m_call = space(11)
@ 04,00
clea
@ 14,17
say "<cr> to exit"
@ 12,01
say "? call number " get m_call
read
  if m_call = ""
  loop
  endif
  seek upper(m_call)
  if found()
  do book_d & & display book information
  else
  @ 14,19
  say "No such call number."
  endif
@ 23,02
wait
endo
dlo
use
set stat on
return
& & wait until key pressed
& & close data bases
& & previous menu
set
stat off
set	talk off
clea
@ 01,00 to 03,78
@ 02,25
say "LNBC - book inquiry by report_flag"
select A
use BOOK inde flag
select B

use PUBL inde p_id
select A
set rela to p_id into PUBL
m_flag = "?"
m_next = "?"
do while m_flag # ""
m_flag = space(1)
@ 04,00
clea
@ 14,18
say "<cr> to exit..."
@ 12,01
say "? report_flag " get m_flag
read
if m_flag = ""
loop
endif
seek rtrim(upper(m_flag))
if found()
do while flag=rtrim(upper(m_flag))
skip
@ 23,02
wait "[ press X to interrupt ] "  -- More ---" to m_next
if upper(m_next) = 'X'
clea
exit
endif
endo
else
say "? No such report_flag."
endif
@ 23,02
wait   && wait until key pressed
endo
use   && close data bases
set stat on
return   && previous menu

set stat off
set talk off
clea
@ 01,00 to 03,78
@ 02,25
say "LNBJ - book inquiry by subject code"
select A
use BOOK inde subj
select B
use PUBL inde p_id
select A
set rela to p_id into PUBL
m_subj = "?"
m_next = "?"
do while m_subj # ""
m_subj = space(1)
@ 04,00
clea
@ 14,19
say "<cr> to exit..."
@ 12,01
say "? subject code " get m_subj
read
if m_subj = ""
loop
endif
seek upper(m_subj)

LNBJ.FRG6

56
if found() 
  do while upper(subj)=upper(m_subj) 
      skip
  @ 23,02
  wait "[ press X to interrupt ]" -- More --" to m_next 
    if upper(m_next) = 'X'
      clear
      exit
    endif
  @ 23,00
  enddo
  else
    say "^ No such subject code." 
  endif
  @ 14,19
  endif
@ 23,02
  wait && wait until key pressed 
  use 
  set stat on return && close data bases
  & previous menu

set stat off
set talk off
clear
@ 01,00 to 03,78
@ 02,25 say "LNBN - book inquiry by accession number"
select A
use BOOK inde accn
select B
use PUBL inde p_p_id
select A
set rela to p_id into PUBL
m_accn = 
  do while m_accn 
    m_accn = space(5)
  @ 04,00
    clear
  @ 14,24
    say 
  @ 12,01
    say "? accession number " get m_accn
    read
      if m_accn = 
        loop
      endif
      seek upper(m_accn)
      if found()
        do book_id && display book information
      else
        say "^ No such accession number." 
      endif
      @ 14,24
      endif
      @ 23,02
      wait && wait until key pressed 
      use 
      set stat on return && close data bases
      & previous menu

LNBIN.PRG
set stat off
set talk off
clea
@ 01,00 to 03,7B
@ 02,25 say "LNBF - book inquiry by status_flag"
  select A
  use BOOK inde stat
  select B
  use PUBL inde p_p_id
  select A
  set rela to p_id into PUBL.
m_stat = "?"
m_next = "?"
do while m_stat # ""
m_stat = space(1)
  @ 04,00
clea
  @ 14,19 say "<cr> to exit..."
  @ 12,01 say "? status_flag " get m_stat
  read
    if m_stat = ""
      loop
    endif
  seek rtrim(upper(m_stat))
  if found()
    do while stat=rtrim(upper(m_stat))
      skip
  @ 23,02
  wait "[press X to interrupt ]" -- More --" to m_next
  if upper(m_next) = 'X'
    @ 23,00
    clea
    exit
    endif
  enddo
  else
    @ 14,18 say "^ No such status_flag."
    endif
  @ 23,02
  wait && wait until key pressed
  enddo
  use
  set stat on && close data bases
  return && previous menu
  @ 01,00 to 03,7B
@ 02,25 say "LNBF - book inquiry by title"
  select A
  use BOOK inde titl
  select B
  use PUBL inde p_p_id
  select A
  set rela to p_id into PUBL.
m_titl = "?"
do while m_titl # ""

m_title = space(64)
@ 04,00
clear
@ 14,13
say "<cr> to exit..."
@ 12,01
say "? title " get m_title
read
    if m_title = ""
        loop
    endif
seek upper(m_title)
    if found()
        do book_d & display book information
    else
        say "^ No such title."
    endif
@ 14,13
endo
wait & wait until key pressed
endo
give
use
stat on
return & previous menu.

set procedure to LBAR
optn = 9
os = 0
    do while os # optn
        opt1 = "by supplier/Agent "
        opt2 = "by Department "
        opt3 = "by date_of_Order "
        opt4 = "by subj_code [pny.] "
        opt5 = "by subj_code [comp]"
        opt6 = "by Publisher"
        opt7 = "by report_gen_Flag"
        opt8 = "by Status_flag"
        opt9 = "Previous Menu"
        optn = 9
        tit1 = "L-DEMS INQUIRIES & ORDERS"
        clear
        do LBAR with optn,tit1
            do case & identify choice...
                case os = 1
                    do LNOA & by supplier/agent
                case os = 2
                    do LNOD & by department
                case os = 3
                    do LNOO & by date of order
                case os = 4
                    do LNOJ & by subject code
                case os = 5
                    do LNO2 & by composite subject
                case os = 6
                    do LNOP & by publisher
                case os = 7
                    do LNOF & by report_flag
                case os = 8
                    do LNOs & by status_flag
            endcase
        os = 0
        return & previous menu
set talk off
clear
@ 01:00 to 03:78
@ 02:15 say "LN02 - orders inquiry by subject code [ composite ]"
select A
use ORDR inda o_sub2
select B
use DEPT inda d_d_id
select A
set rela to d_id into DEPT
m_subp = "$"
m_next = "$"
do while m_subp $"
  m_subp = space(4)
  m_subs = space(4)
  m_sub2 = space(4)
@ 04:00
clear
@ 15:33 say "<cr> to exit..."
@ 12:01 say "? subject code [ primary ] " get m_subp
  read
  if m_subp $"
    loop
  endif
@ 13:01 say "? subject code [ secondary ] " get m_subs
  read m_sub2 = upper(m_subp)+upper(m_subs)
  seek m_sub2
  if found()
    do while upper(sub)+upper(sub2)=m_sub2
      do ordr_d: && display orders info.
        skip
    @ 23:02
    wait "] press X to interrupt ": -- More --" to m_next
      if upper(m_next) = 'X'
        clear
        exit
      endif
      endif
      else
      @ 15:33
      say "^ No such subject code combination."
    endif
    @ 23:02
    wait && wait until key pressed
    endif
    use
    set stat on
    && close data bases
    return
    && previous menu
use DEPT inde d_d_id
select A
set rela to s_id into DEPT
m_s_id = "?"
m_next = "?"
do while m_s_id # ""
m_s_id = space(2)
@ 04,00 clear
@ 04,20 say "<cr> to exit..."
@ 12,01 say "? supplier code " get m_s_id
read
if m_s_id = ""
  loop
endif
seek m_s_id
if found()
do while s_id = m_s_id
  do ord_r_d && display orders info.
skip:
@enddo
@ 23,02 wait "[ press X to interrupt ]" " More --> " to m_next
if upper(m_next) = 'X'
@ 23,00 clear
  exit
endif
else say "^ No such supplier code.
@endif
@ 14,20 endif
@ 23,02 wait && wait until key pressed
@ 01,00 to 03,7B
clear
@ 02,15 say "LNOD - orders inquiry by department code"
select A
use URDR inde o_d_id
select B
use DEPT inde d_d_id
select A
set rela to d_id into DEPT
m_d_id = "?"
m_next = "?"
do while m_d_id # ""
m_d_id = space(2)
@ 04,00 clear
@ 14,22 say "<cr> to exit..."
@ 12,01 say "? department code " get m_d_id
read
if m_d_id = ""
  loop
endif
seek m_d_id
return
set stat off
set talk off
clear
if found()
do while d_id=m_d_id
    do ordr_d       && display orders info.
skip
@end 23,02
wait "[ press X to interrupt ]"    -- More "=" to m_next
    if upper(m_next) = 'X'
        clea
        exit
        endif
    enddo
else
    say "^ No such department code."
endif
@end 23,02
wait
    && wait until key pressed
enddo
use
    && close data bases
set stat on
    && previous menu
return

set stat off
set talk off
clea
@ 01,00 to 03,78
@ 02,25 say "LNOF - orders inquiry by report_flag"
select A
    use ORDR inde d_flag
select B
    use DEPT inde d_d_id
select A
    set rela to d_id into DEPT
eflag = "?"
m_next = "?"
do while m_flag != " "
    m_flag = space(1)
@end 01,00
clea
@ 14,18 say "(cr) to exit..."
@ 12,01 say "? report_flag " get m_flag
read
    if m_flag = " "
        loop
    endif
seek rtrim(upper(m_flag))
if found()
do while flag=rtrim(upper(m_flag))
    do ordr_d       && display orders info.
skip
@end 23,02
wait "[ press X to interrupt ]"    -- More "=" to m_next
    if upper(m_next) = 'X'
        clea
        exit
        endif
    enddo
else
    say "^ No such report_flag."
endif
@end 14,18
wait
enddo
use
set stat on
return

set stat off
set talk off
clear
@ 01.00 to 03.78
@ 02.23 say "LNOJ - orders inquiry by subject code"
select A
use ORDR inde c_subj
select B
use DEPT inde d_id
select A
set rela to d_id into DEPT
m_subj = "?"
m_next = "?"
do while m_subj = ""
  m_subj = space(1)
enddo
@ 04.09 clear
@ 14.17 say "<cr> to exit...
@ 12.01 say "? subject code? " get m_subj
read
  if m_subj = ""
    loop
  endif
seek upper(m_subj)
if found
  do while subj-upper(m_subj)
    do ordr_d  & & display orders info.
      skip
  enddo
@ 23.02 wait "[ press X to interrupt ]
      -- More -- " to m_next
@ 23.00 if upper(m_next) = 'X'
    clear
    exit
  endif
enddo
else
  say " No such subject code."
endif
@ 14.17
dowhile
@ 23.02 wait
endo
use
set stat on
return

set stat off
set talk off
clear
@ 01.00 to 03.78
@ 02.19 say "LNOO - orders inquiry by date [ ...on or after ]"
select A
use ORDR inde a_dord
select B

LNOJ.PRG
LNOO.PRG
use DEPT inde d_d_id
select A
set rel to d_id into DEPT
m_dord = "?"
m_next = "?"
do while m_dord # " "
    m_dord = space(8)
@ 04,00
clea
@ 14,15
say "<cr> to exit..."
@ 12,01
say "? mm/dd/yy " get m_dord
read
    if m_dord = " "
        loop
    endif
    seek m_dord
    if found()
        do while .not. eof()
            do ord
                & display orders info.
                skip
        @ 21,02
    wait "[ press X to interact ] " -- More --" to m_next
        if upper(m_next) = 'X'
            clea
            exit
            endif
        enddo
    else
        say "No entry against this date."
        endif
    @ 23,02
    wait
    @ 14,15
    else
        say "No entry against this date."
        endif
    @ 20,02
    use
set stat on
return
@ 01,00 to 05,78
@ 02,13
say "LNOD - orders inquiry by publisher code"
select A
use ORDR inde o_p_id
select B
use DEPT inde d_d_id
select A
set rel to d_id into DEPT
m_p_id = "?"
m_next = "?"
do while m_p_id # " "
    m_p_id = space(2)
@ 04,00
clea
@ 14,21
say "<cr> to exit..."
@ 12,01
say "? publisher code " get m_p_id
read
    if m_p_id = " "
        loop
    endif
    seek m_p_id
if found()
    do while p_id=m_p_id
        do ord_r_d     && display orders info.
            skip
@ 23,02
wait "[ press X to interrupt ]"      -- More ---" to m_next
    if upper(m_next) = 'X'
@ 23,00
        clea
        exit
endif
endo
endo
else
    say "^ No such publisher code."
endif
@ 23,02
wait
endo
use
set stat on
return
& & previous menu

set stat off
set talk off
clea
@ 01,00 to 03,78
@ 02,25 say "LNOS - orders inquiry by status_flag"
select A
use ORDR inde o_stat
select B
use DEPT inde d_d_id
select A
set rela to d_id into DEPT
m_stat = "?"
m_next = "?"
do while m_stat # " "
    m_stat = space(1)
    m_next = "?"
do while m_stat = " "
        loop
endif
seek rtrim(upper(m_stat))
if found()
    do while stat=rtrim(upper(m_stat))
        do ord_r_d     && display orders info.
            skip
@ 23,02
wait "[ press X to interrupt ]"      -- More ---" to m_next
    if upper(m_next) = 'X'
@ 23,00
        clea
        exit
endif
endo
endo
else
    say "^ No such status_flag."
@end
@ 14,18
say "<cr> to exit...
@ 12,01 say "? status_flag " get m_stat
read
    if m_stat = " "
        loop
endif
seek rtrim(upper(m_stat))
if found()
wait
enddo

%% wait until key pressed

set
use
return

%% close data bases

%% previous menu

set
procedure
to
LBAR
%% user inquiry menu

opth = 9
as = 0

do while as if opth
  opt1 = "by user_id_Number"
  opt2 = "by user_Type"
  opt3 = "by user_Name"
  opt4 = "by date_of_Exprie"
  opt5 = "by fines due"
  opt6 = "by user_Department"
  opt7 = "by report_gen_flag"
  opt8 = "by Status_flag"
  opt9 = "Previous Menu"
  optn = 9
  title = "L-DBMS INQUIRIES * USER_STATUS"

  clear
  do LBAR with optn,title
    do case
      case as = 1
        do LNUN
          %% by user_id_number
      case as = 2
        do LNU
          %% by user_type_code
      case as = 3
        do LNUA
          %% by user_name
      case as = 4
        do LINUX
          %% by date of expiry
      case as = 5
        do LNUC
          %% those with fines due
      case as = 6
        do LNU
          %% by user_department
      case as = 7
        do LNUF
          %% by report_flag
      case as = 8
        do LNUS
          %% by status_flag
      endcase
    os = 0
    return
  enddo
  return
  %% previous menu

set
stat
off
set
talk
off

clea

@ 01:00 to 03:78
@ 02,25 say "LNUA - user inquiry by user_name"

select A
use USER in de u_unam
select B
use DEPT in de d_d_id
select A
set rela to d_id into DEPT
m_unam = "?"
do while m_unam = " "

m_unam = space(32)
clear
say "<cr> to exit..."
say "'? user_name " get m_unam
read
if m_unam = ""
loop
endif
seek upper(m_unam)
if found{}
do user_d && display user info.
else
say "^ No such user."
endif
wait && wait until key pressed
enddo
use && close data bases
set stat on
return && previous menu.

set stat off
set talk off
clear
say "LNUN - inquiring users with outstanding fines"
select A
use USER inde u_fdue
select B
use DEPT inde d_d_id
select A
set rela to d_id into DEPT
m_next = "?"
do while fdue=0.0
skip
enddo
do while .not. eof()
do user_d && display user info.
skip
wait "[ press X to interrupt ]" -- More --" to m_next
if upper(m_next) = 'X'
clear
exit
endif
enddo

wait && wait until key pressed
enddo
use && close data bases
set stat on
return && previous menu.

set stat off
set talk off
clear
say "LNUD - user inquiry by user department"
select A
use USER inde u_d_id
select B
use DEPT inde d_d_id
select A
set rel to d_id into DEPT
m_d_id = ""
m_next = ""
do while m_d_id # ""
    m_d_id = space(2)
enddo
@ 04,00
    clea
@ 14,22
    say "<cr> to exit...
@ 12,01
say "? user department " get m_d_id
read
    if m_d_id = ""
        loop
    endif
seek m_d_id
    if found()
        do while d_id = m_d_id
            do user_d && display user info.
                skip
        enddo
    endif
@ 23,02
    wait "[ press X to interrupt ]: More " to m_next
    if upper(m_next) = 'X'
        clea
        exit
    endif
@enddo
@ 14,22
    else
        say "No such user department."
    endif
@ 23,02
    wait && wait until key pressed
@enddo
    use && close data bases
set stat on && previous menu
return

set stat off
set talk off
clea
@ 01,00 to 03,78
@ 02,25
say "LNUF - user inquiry by report_flag"
select A
use USER inde u_flag
select B
use DEPT inde d_d_id
select A
set rel to d_id into DEPT
m_flag = ""
m_next = ""
do while m_flag # ""
    m_flag = space(1)
enddo
@ 04,00
    clea
@ 14,18
    say "<cr> to exit...
@ 12,01
say "? report_flag " get m_flag
read
    if m_flag = ""
        loop
endif
}
seek rtrim(upper(m_flag))
if found()
do while flag=rtrim(upper(m_flag))
do user_id && display user info.
skip
@end 23,02
wait "[ press X to interrupt ]" -- More --" to m_next
if upper(m_next) = 'X'

clea
exit
endif
enddo
else
say "~ No such report_flag."
endif
@end 23,02
wait && wait until key pressed
enddo
use && close data bases
set stat on && previous menu
return

set stat off
set talk off
clea
@end 01,00 to 03,78
@ 02,25 say "LNUN - user inquiry by user_id"
select A
use USER inde u_u_id
select B
use DEPT inde d_d_id
select A
set rela to d_id into DEPT
m_u_id = "?"
do while m_u_id # " "
    m_u_id = space(6)
@end 04,09
clea
@ 14,17 say "<cr> to exit..."
@ 12,01 say "? user_id " get m_u_id
read
    if m_u_id = " "
        loop
    endif
seek upper(m_u_id)
if found()
do user_d && display user info.
else
    say "~ No such user_id."
endif
@end 14,17
@ 23,02
wait && wait until key pressed
enddo
use && close data bases
set stat on && previous menu
return

LNUN.PRG

69
set stat off
do while m_stat # ""
   m_stat = space(1)
endo
@ 04,00 clea
@ 14,18 say "<cr> to exit...
@ 12,01 say "? status_flag" get m_stat
read
   if m_stat = ""
      loop
   endif
   seek rtrim(upper(m_stat))
   if found()
      do while stat=rtrim(upper(m_stat))
         do user_d & display user info.
            skip
         enddo
      endif
      wait "[ press X to interrupt ]" quit
      if upper(m_next) = 'x'
         @ 23,00 clea
      endif
      @ 14,18 say "^ No such status_flag."
      endif
   else
      say "^ More ...
endo
@ 23,02 wait & wait until key pressed
endo
set stat on
return & previous menu

set stat off
do while m_stat # ""
   m_stat = space(1)
endo
@ 04,00 clea
@ 14,18 say "<cr> to exit...
@ 12,01 say "? user type" get m_u_ty
read
   if m_u_ty = ""
      loop
   endif
   seek d_d_id
   if found()
      do while m_u_ty = rtrim(d_d_id)
         do user_d & display user info.
            skip
         enddo
      endif
      wait "[ press X to interrupt ]" quit
      if upper(m_next) = 'x'
         @ 23,00 clea
      endif
      @ 14,18 say "^ No such user type."
      endif
   else
      say "^ More ...
endo
@ 23,02 wait & wait until key pressed
endo
set stat on
return & previous menu
do while m_u_ty # ""
m_u_ty = space(1)
enddo
@ 04,00
say "<cr> to exit..."
@ 12,01
say "? user type "
get m_u_ty
read
if m_u_ty = ""
loop
endif
seek rtrim(upper(m_u_ty))
if found()
do while u_ty=rtrim(upper(m_u_ty))
do user_d && display user info.
skip
@ 23,02
wait "[ press X to interrupt ]" " More " to m_next
if upper(m_next) = 'X'
@ 23,00
clea
exit
endif
else
enddo
@ 14,16
say "No such user type."
endif
@ 23,02
wait && wait until key pressed
enddo
&& close data bases
set stat on
return && previous menu

LNUX.PRG

set stat off
set talk off
clea
@ 01,00 to 03,78
@ 02,15
say "LNUX - user inquiry by date of expiry [ on or before ]"
select A
use USER inde u_dexp
select B
use DEPT inde d_d_id
select A
set rela to d_d_id into DEPT
m_dexp = "?"
m_next = "?"
do while m_dexp # ""
m_dexp = space(8)
enddo
@ 04,00
clea
@ 14,15
say "<cr> to exit...
@ 12,01
say "? mm/dd/yy "
get m_dexp
read
if m_dexp = ""
loop
endif
if dtoc(dexp) <= m_dexp
do while dtoc(dexp) <= m_dexp
do user_d && display user info.
skip
@ 23,02
wait "[ press X to interrupt ]" " More " to m_next
if upper(m_next) = 'X'
  | clear
  | exit
  | endif
  | enddo
else
  | say "No entry in the range specified."
  | endif
@ 14,15
doit clear
else wait && wait until key pressed
  | endif
  | use
  | && close data bases
  | set
  | stat on
  | return && previous menu

set procedure to LBAR && reports menu
optn = 3
os = 0
  | do while os ! optn
    | opt1 = "Acquisitions"
    | opt2 = "Rental Management"
    | opt3 = "Main Menu"
    | optn = 3
    | titl = "L-DBMS REPORTS"
    | clear
    | do LBAR with optn,titl && identify choice..
    | do case
      | case os = 1 && acquisitions
        | do LRA
      | case os = 2 && rental management
        | do LRR
    | endcase
  | endif
  | os = 0
  | return && previous menu

set procedure to LBAR && acquisitions reports menu
optn = 9
os = 0
  | do while os ! optn
    | opt1 = "Orders to suppliers"
    | opt2 = "books ordered _Dept."
    | opt3 = "books ordered _Supp."
    | opt4 = "books Unavailable"
    | opt5 = "new arrivals,by Rec#"
    | opt6 = "new arrivals,by subj"
    | opt7 = "new arrivals,by Titl"
    | opt8 = "new arrivals,by Auth"
    | opt9 = "Previous Menu"
    | optn = 9
    | titl = "L-DBMS REPORTS & ACQUISITIONS"
    | clear
    | do LBAR with optn,titl && identify choice..
    | do case
      | case os = 1 && orders to suppliers
        | do LRAD
      | case os = 2 && books ordered by dept.
        | do LRAD
    | endcase

72
case as = 5
  do LRAS
    & books ordered by supp.
  enddo
endcase

case as = 4
  do LRAU
    & books unavailable
endcase

case as = 5
  do LRAR
    & orders, by record #
endcase

case as = 6
  do LRAJ
    & orders, by subject
endcase

case as = 7
  do LRAT
    & orders, by title
endcase

case as = 8
  do LRRA
    & orders, by author
endcase

os = 0
return

&& previous menu

------------------------------------------------------------
LRAA.PRG
------------------------------------------------------------

clear
set
stat off
close all
use ORDR inde o_auth
report form LRAN for flag='N'
close all
set
stat on
return

&& previous menu

------------------------------------------------------------
LRAD.PRG
------------------------------------------------------------

clear
set
status off
close all
erase deptordr.dbf
select a
use ordr index o_d_id
select b
use dept index d_d_id
select a
join with dept to deptordr for d_id=b->d_id fields titl,auth,
  year,cost,copy,pnam,b->dnam
  close all
use deptordr
report form LRAQ
close all
return

&& previous menu

------------------------------------------------------------
LRAJ.PRG
------------------------------------------------------------

clear
set
stat off
close all
use ORDR inde o_sub2
report form LRAN for flag='N'
close all
set
stat on
return

&& previous menu
clear
set status off
close all
erase suppordr.dbf
select a
use ordr index o_s_id
select b
use supp index s_s_id
select a
join with supp to suppordr for s_id=b->s_id fields titl,auth,
year,copy,pnam,b->snam,b->sadd,b->scty
close all
use suppordr
report form LRAO
close all
return  %& previous menu

-----------------------------
LRAR.PRG
-----------------------------
clear
set stat off
close all
use ORDR
report form LRAR for flag='N'
close all
set stat on
return  %& previous menu

-----------------------------
LRAJ.PRG
-----------------------------
clear
set stat off
close all
use ORDR index o_dord
report form LRAS
close all
set stat on
return  %& previous menu

-----------------------------
LRAT.PRG
-----------------------------
clear
set stat off
close all
use ORDR index o_litl
report form LRAN for flag='N'
close all
set stat on
return  %& previous menu

-----------------------------
LRAU.PRG
-----------------------------
clear
set stat off
close all
use ORDR index o_d_id
report form LRAU for flag='M'
close all
set stat on
return  %& previous menu
**LRG.PRG**

```plaintext
set procedure to LBAR && general reports
optn = 3 && & SPECIFICATION INCOMPLETE *

* as = 0
    do while as & optn
        opt1 = "",
        opt2 = "",
        opt3 = "Previous Menu",
        optn = 3
    titl = "LDBMS * REPORTS * GENERAL"
clear
    do LBAR with optn,titl
        do case : && identify choice..
            case as = 1
                do NYET && not yet installed!
            case as = 2
                do NYET && not yet installed!
        endcase.
    enddo
    as = 0
    return && previous menu
```

**LRR.PRG**

```plaintext
set procedure to LBAR && rental related reports menu
optn = 8
as = 0
    do while as & optn
        opt1 = "books Issued"
        opt2 = "books Returned"
        opt3 = "books still on Loan"
        opt4 = "o.due rem (Students)"
        opt5 = "o.due rem (Teachers)"
        opt6 = "reserved collection"
        opt7 = "final Fine report"
        opt8 = "Previous Menu"
    enddo
    as = 0
    return && previous menu
```
set procedure to LBAR & updates menu
optn = 8
os = 0

  do while os # optn
    opt1 = "Rental"
    opt2 = "Transfer to BOOK.DBF"
    opt3 = "Departments"
    opt4 = "Orders"
    opt5 = "Publishers"
    opt6 = "Suppliers"
    opt7 = "Users"
    opt8 = "Main Menu"
    optn = 8

  titl = "L-DBMS UPDATE DATABASES"
  clear
  do LBAR with optn, titl
    & identify choice...
    do case
      case os = 1
        & rental related < menu >
        do LUR
      case os = 2
        & transfer to BOOK.DBF
        do LUT
      case os = 3
        & update department info.
        do LUD
      case os = 4
        & update orders info.
        do LUD
      case os = 5
        & update publisher info.
        do LUP
      case os = 6
        & update supplier info.
        do LUS
      case os = 7
        & update user info.
        do LUU
    endcase
    os = 0
  return & previous menu

set procedure to LBAR & department file maintenance
optn = 4
os = 0

  do while os # optn
    opt1 = "Add department"
    opt2 = "Change dept. info."
    opt3 = "Delete department"
    opt4 = "Previous Menu"
    optn = 4

  titl = "L-DBMS UPDATE -> DEPT.DBF"
  clear
  do LBAR with optn, titl
    & identify choice...
    do case
      case os = 1
        & add department
        do LUDA
      case os = 2
        & change department
        do LUDC
      case os = 3
        & delete department
        do LUDD
    endcase
    enddo : os = 0 : return & previous menu
LUDA.PRG

clear
set stat off
use DEPT
set format to LUDA
append
close all
return && previous menu

LUDC.PRG

clear
set stat off
use DEPT
DC="
@10,10 SAY "ENTER THE DEPARTMENT CODE:" GET DC PICT "XX"
READ
set format to LUDA
EDIT FOR D_ID=DC
close all
return && previous menu

LUDD.PRG

clear
set stat off
use DEPT index d_d_id
DC="
@10,10 SAY "ENTER THE DEPARTMENT CODE TO DELETE:" GET DC PICT "XX"
READ
DELETE FOR D_ID=DC
PACK
close all
return && previous menu

LU1.PRG

clear && record issue of a book
set status off
@ 01,00 to 03,78
@ 02,25 say "LU1 ___ issue a book"
@ 20,00 to 22,29
@ 21,02 say "Option not yet implemented"
@ 23,00
wait "Press a key to continue..."
set status on
return && previous menu

LUO.PRG

set procedure to LBAR && orders data base maintenance
optn = 4
os = 0
  do while os # optn
    opt1 = "Add orders."
    opt2 = "Change orders info."
    opt3 = "Delete orders."
    opt4 = "Previous Menu."
    optn = 4
    titl = "L-DBMS UPDATE -> ORDR.DBF"
clear
do LBAR with optn,titl
  do case && identify choice..
case os = 1
    do LUDA  && add orders
endcase

case os = 2
    do LUOC  && change orders info.
endcase

case os = 3
    do LUDD  && delete orders
endcase

endif

os = 0
return  && previous menu

set  bell  off
set  stat  off
set  talk  off
    clear
@ 00,00 to 21,79
@ 01,02 say "ORDR <= adding new records"
@ 01,35 say "HIGHLIGHT indicates currently ACTIVE field"
@ 03,02 say "title = "
@ 04,02 say "author = "
@ 05,02 say "location = "
@ 06,02 say "publisher = "
@ 07,02 say "year of publication = "
@ 08,02 say "price, in Taka = "
@ 09,02 say "copies to order = "
@ 10,02 say "call number, if known = "
@ 11,02 say "acct number, if known = "
@ 12,02 say "subject code [primary ] = "
@ 13,02 say "subject code [secondary ] = "
@ 14,02 say "date ordered = "
@ 15,02 say "supplier_id = "
@ 16,02 say "publisher_id = "
@ 17,02 say "department_id = "
@ 18,02 say "remarks, if any = "
@ 19,02 say "report_flag (normally @) = "
@ 20,02 say "status_flag (normally @) = "
MORE = Y.
    DO  while  MORE
        m_title= space (64)
        m_author= space (32)
        m_city= space (32)
        m_pname= space (32)
        m_year= space (4)
        m_cost= 0.00
        m_copy= 0
        m_call= space (16)
        m_acct= space (8)
        m_subj= space (4)
        m_sub2= space (4)
        m_dord= ctod ('19//00/00')
        m_s_id= space (2)
        m_p_id= space (2)
        m_d_id= space (2)
        m_remark= space (30)
        m_flag= space (1)
        m_stat= space (1)
        set  conf on
    @ 03,15 clear to 03,77
    @ 04,35 clear to 20,77

LUOA.PRG
@ 03,15 get m_title
read
@ 03,15 say m_title
@ 04,35 get m_auth
read
@ 04,35 say m_auth
@ 05,35 get m_city
read
@ 05,35 say m_city
@ 06,35 get m_pnam
read
@ 06,35 say m_pnam
@ 07,35 get m_year
read
@ 07,35 say m_year
@ 08,35 get m_cost range 0.00,1999.99
read
@ 08,35 say m_cost
@ 09,00 to 21,79
@ 09,35 get m_copy range 1.99
read
@ 09,35 say m_copy
@ 09,00 to 21,79
@ 10,35 get m_call
read
@ 10,35 say m_call
@ 11,35 get m_accn
read
@ 11,35 say m_accn
@ 12,35 get m_subj
read
@ 12,35 say m_subj
@ 13,35 get m_sub2
read
@ 13,35 say m_sub2
@ 14,35 get m_dord range ctod('07/01/89'),ctod('06/30/90')
read
@ 14,35 say m_dord
@ 09,00 to 21,79
@ 15,35 get m_s_id
read
@ 15,35 say m_s_id
@ 16,35 get m_p_id
read
@ 16,35 say m_p_id
@ 17,35 get m_d_id
read
@ 17,35 say m_d_id
@ 18,35 get m_rank
read
@ 18,35 say m_rank
DO while .not. upper(m_flag) "$ @ABCDEFGHIJKLMNOPQRSTUVWXYZKLMNO" 
@ 19,35 get m_flag
read
ENDDO
@ 19,35 say m_flag
DO while .not. upper(m_stat) "$ @ABCDEFGHIJKLMNOPQRSTUVWXYZKLMNO" 
@ 20,35 get m_stat
read
ENDDO
@ 20,35 say m_stat
set conf off
OK: = .Y.
@ 23,02 say "File the above data? Y/N"
@ 23,35 get OK
read
@ 23,35 say OK
if OK
use UDR
append blank
replace titl with m_titl
replace auth with m_auth
replace city with m_city
replace pnam with m_pnam
replace year with m_year
replace cost with m_cost
replace copy with m_copy
replace call with m_call
replace acrn with m_acrn
replace subj with m_subj
replace sub2 with m_sub2
replace dord with m_dord
replace s_id with m_s_id
replace p_id with m_p_id
replace d_id with m_d_id
replace remk with m_remk
replace flag with m_flag
replace stat with m_stat
use && close data bases
endif
@ 24,02 say "Enter more records? Y/N"
@ 24,35 get MORE
read
ENDDO while MORE
set status on
cancel

set procedure to LBAR && publisher file maintenance
optn = 4
os = 0
do while os # optn
opt1 = "Add publisher"
opt2 = "Change publisher info"
opt3 = "Delete publisher"
opt4 = "Previous Menu"
optn = 4
titl = "L-DEMS UPDATE -> PUBL:DBE"
set LUP.FRG LUP
endif
do LBAR with optn,titl && identify choice...
do case
  case os = 1
    do LUPA && add publisher
  case os = 2
    do LUPC && change publisher info.
  case os = 3
    do LUDP && delete publisher
  endcase
  os = 0 : return && previous menu
set procedure to LBAR && rental service menu
optn = 4
os = 0
   do while os <> optr
      opt1 = "Issue book"
      opt2 = "Record returned book"
      opt3 = "Reserve a book"
      opt4 = "Previous Menu"
      optr = 4
      title = "L-DBMS UPDATES -> RENTAL MANAGEMENT LUR"
clear
do LBAR with optr,title && identify choice.
do case
   case os = 1
      do LUR1 && record issue of book
   case os = 2
      do LUR2 && record return of book
   case os = 3
      do LURV && reserve a book
   endcase
os = 0
enddo
return && previous menu

set procedure to LBAR && supplier data base maintenance
optr = 4
os = 0
   do while os <> optr
      opt1 = "Add supplier"
      opt2 = "Change supplier info"
      opt3 = "Delete supplier"
      opt4 = "Previous Menu"
      optr = 4
      title = "L-DBMS UPDATES -> SUPP, DBF LUS"
clear
do LBAR with optr,title && identify choice.
do case
   case os = 1
      do LUSA && add supplier
   case os = 2
      do LUSC && change supplier info.
   case os = 3
      do LUSD && delete supplier
   endcase
os = 0
enddo
return && previous menu
LUU.FRG

set procedure to LBAR
& user data base maintenance
optn = 4
os = 0

do while os # optn
    opt1 = "Add user"
    opt2 = "Change user info."
    opt3 = "Delete user"
    opt4 = "Previous Menu"
optn = 4
    titl = "L-DBMS UPDATE -> USER.DBF"
clear
do LBAR with optn,titl
    & identify choice.
do case
    case os = 1
        do LUUA & add user
    case os = 2
        do LUUC & change user info.
    case os = 3
        do LUUD & delete user
endcase
os = 0

return & previous menu

BOOK_D.FRG

@ 04.00 clear & displays book information
@ 04.00 to 15.79
@ 05.02 say "title "+chr(295)+chr(16)+""
@ 06.02 say "author "= ""
@ 07.02 say "location "= ""
@ 08.02 say "publisher "= ""
@ 09.02 say "year of publication "= ""
@ 10.02 say "call number "= ""
@ 11.02 say "accession number "= ""
@ 12.02 say "subject code [primary ] "= ""
@ 13.02 say "subject code [secondary ] "= ""
@ 14.02 say "requisitioning dept. [ code ] "= ""
@ 05.14 say titl
@ 06.35 say auth
@ 07.35 say b->pcty
@ 08.35 say b->pnam
@ 09.33 say '1' +year
@ 10.33 say call
@ 11.33 say accn
@ 12.35 say subj
@ 13.35 say sub2
@ 14.33 say d_id
DEPT_D.FRG
@ 04,00 clear
@ 04,00 to 10,78
@ 05,02 say "department_name ="
@ 06,02 say "address ="
@ 07,02 say ""
@ 08,02 say "department_id ="
@ 09,02 say "remarks ="
@ 05,35 say dnam
@ 06,35 say dadd
@ 07,35 say dcty
@ 08,35 say d_id
@ 09,35 say remk

ORDR_D.FRG
@ 04,00 clear
@ 04,00 to 17,78
@ 05,02 say "title = "chr(205)+chr(16)+"
@ 06,02 say "author ="
@ 07,02 say "location ="
@ 08,02 say "publisher ="
@ 09,02 say "year of publication ="
@ 10,02 say "price, in laka ="
@ 11,02 say "copies to order ="
@ 12,02 say "subject code [ primary ] ="
@ 13,02 say "subject code [ secondary ] ="
@ 14,02 say "date ordered ="
@ 15,02 say "supplier_id ="
@ 16,02 say "requisitioning department ="
@ 05,14 say titl
@ 06,35 say auth
@ 07,35 say city
@ 08,35 say pnam
@ 09,35 say year
@ 10,35 say str(cost,6,2)
@ 11,35 say str(copy,1)
@ 12,35 say subj
@ 13,35 say sub2
@ 14,35 say dord
@ 15,35 say s_id
@ 16,35 say b-dnam

PUBL_D.FRG
@ 04,00 clear
@ 04,00 to 10,78
@ 05,02 say "publisher_name ="
@ 06,02 say "address ="
@ 07,02 say ""
@ 08,02 say "publisher_id ="
@ 09,02 say "remarks ="
@ 05,35 say pnam
@ 06,35 say padd
@ 07,35 say pcty
@ 08,35 say p_id
@ 09,35 say remk
SUPP_D.PRG

@ 04,00 clear
@ 04,00 to 10,78
@ 05,02 say "supplier_name ="
@ 06,02 say "address ="
@ 07,02 say ""
@ 08,02 say "supplier_id ="
@ 09,02 say "remarks ="
@ 05,35 say snam
@ 06,35 say sadd
@ 07,35 say scty
@ 08,35 say s_id
@ 09,35 say remk

USER_D.PRG

@ 04,00 clear
@ 04,00 to 15,78
@ 05,02 say "user_name ="
@ 06,02 say "department ="
@ 07,02 say "address ="
@ 08,02 say ""
@ 09,02 say "user_id number ="
@ 10,02 say "user category ="
@ 11,02 say "date of expiry ="
@ 12,02 say "borrowing limit ="
@ 13,02 say "books currently on loan ="
@ 14,02 say "fines due ="
@ 05,35 say unam
@ 06,35 say b->dnam
@ 07,35 say uadd
@ 08,35 say ucty
@ 09,35 say u_id
@ 10,35 say u_ty
@ 11,35 say dtoc(desp)
@ 12,35 say str(blim,1,0)
@ 13,35 say str(ncur,1,0)
@ 14,35 say str(fdue,6,2)
LIBRARY
para opno, tit1
set status off
clear
row = 5
@ 00,00 to 02,79
@ 01,17 say tit1
@ 21,18 to 24,79
@ 22,18 say "Highlight option with"
@ 22,40 say chr(24)+" or "+chr(25)+" and press " +chr(17)+chr(217)
@ 23,22 say "or press appropriate menu number"
@ 04,23
row = 5
do while row-4 <= opno
sub = str(row-4,1)
opt&sub = if (val(opt&sub)=0, str(row-4,1)+". "+opt&sub, opt&sub)
enddo
say opt&sub
row = row+1
endo
opt = 1
sub = "1"
se1 = 0
@ 05,23 get opt!
clear gets
cs = 0
do while cs = 0
se1 = 0
do while se1 = 0
se1 = inkey ()
endo
if (se1=24) .or. (se1=5)
say opt&sub
opt = iiif (se1=24, opt+1, opt-1)
opt = iiif (opt*opno, 1, opt)
sub = str (opt,1)
endo
get opt&sub
clear gets
loop
if (se1=49) .and. (se1 < 49+opno) then
cs = se1-48
endo
if (se1=13) then
cs = opt
endo
% activate option
set status on
% and
return
% return to calling program