

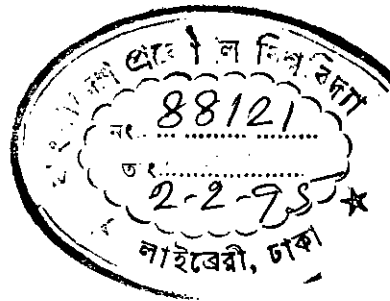
A DATA BASE MANAGEMENT SYSTEM FOR LIBRARY OPERATION

A project report

by

AHSANUL HABIB

Approved as to style and contents by



Syed Mahbubur Rahman
20/9/90

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

Chairman
[Supervisor]

Md. Shamsul Alam
25/7/90

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

Member



A.K.M. Ahsanullah
20/9/90

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 M. 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.M. 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. Towfigur Rahman, Zunaid Hamid Kazi, and M. Masroor Ali, for their comments and observations and the occasional troubleshooting tips -- to Mr. Nazmul 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.

ABSTRACT

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 existing 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

Chapter	Title	Page No.
1.	INTRODUCTION	1
1.1	OBJECTIVES OF THE PROJECT	1
1.2	THEORETICAL PERSPECTIVES ON DATA BASE DESIGN	2
1.2.1	DATA PROBLEMS IN TRADITIONAL FILE MANAGEMENT	3
1.2.2	THE LOGICAL .VS. PHYSICAL VIEW OF DATA	4
1.2.3	ATTRIBUTES OF A RELATIONAL DATA BASE	6
1.2.4	INFORMATION RETRIEVAL IN A RELATIONAL SYSTEM	6
1.2.5	NORMALIZING DATA STRUCTURES	7
2.	EVALUATION OF A MANUAL SYSTEM	10
2.1	EXISTING ACQUISITIONS PROCESS	10
2.2	EXISTING RENTAL MANAGEMENT SYSTEM	12
2.3	JUSTIFICATION FOR A NEW SYSTEM	13
3.	THE PROPOSED SYSTEM AND ITS OPERATION	16
3.1	ACQUISITIONS	16
3.2	RENTAL MANAGEMENT	17
3.3	ON-LINE INQUIRY	18
4.	RECORD DESIGN	19
4.1	THE ACQUISITIONS SUBSYSTEM ENTITIES	19
4.2	THE PERMANENT BOOK DATA BASE	21
4.3	THE RENTAL MANAGEMENT SUBSYSTEM	22
5.	OUTPUT DESIGN	25
5.1	REPORTS [acquisitions subsystem]	26
5.2	REPORTS [rental management subsystem]	28
5.3	VIDEO SCREEN DESIGN FOR ON-LINE INQUIRIES	32
6.	THE LOGICAL STRUCTURE OF THE SYSTEM'S OPERATION	35
6.1	INQUIRING THE DATABASE	39
6.2	GENERATING REPORTS	42
6.3	UPDATES AND MAINTENANCE	44
7.	IMPLEMENTATION AND CONCLUSION	47
7.1	TEST-RUN OF THE SYSTEM	47
7.2	LIMITATIONS AND POSSIBLE ENHANCEMENTS	47
7.3	IMPLEMENTATION CONSIDERATIONS	49
*	APPENDIX [source code listing]	52

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 100000 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 DATA BASE

A relational data base 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 data base file:

- 1) Each data base 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 data base is unique. That is, no two records may contain exactly the same values in all matching columns. Each record in a relational data base 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 data base files can be produced by combining and/or extracting subsets of data from other data base files.

1.2.4 INFORMATION RETRIEVAL IN A RELATIONAL SYSTEM

Three fundamental operations are performed on files constituting a relational data base 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 $n*m$ 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 MANUAL 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 borrowers 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

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 @ABCDEFGHIJKLMNO 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

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

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 <?cty>, 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.

TABLE 4.2 STRUCTURE OF SUPPLIER_INFORMATION FILE : SUPP.DBF

?	.NDX	fnam	width	type	description	recl = 126
S_S_ID	s_id		2	char	supplier identification code	
	x		2	char		
	snam		32	char	name of supplier	
	sadd		32	char	address of supplier	
	scty		32	char	city	
	bdue		8:2	N	bills outstanding (in Taka)	
	lord		8	D	date of last placement of order	
	remk		8	char	remarks, if any	
	flag		1	char	log of reports generated	
	stat		1	char	status_code	

TABLE 4.3 STRUCTURE OF PUBLISHER_INFORMATION FILE : PUBL.DBF

?	.NDX	fnam	width	type	description	recl = 126
P_P_ID	p_id		2	char	publisher identification code	
	x		2	char	< unused >	
	pnam		32	char	name of publisher	
	padd		32	char	address of publisher	
	pcty		32	char	city	
	ordr		8:2	N	total amount of order (in Taka)	
	dt_p		8	D	< unused >	
	remk		8	char	remarks, if any	
	flag		1	char	log of reports generated	
	stat		1	char	status_code	

TABLE 4.4 STRUCTURE OF DEPT. PARTICULARS FILE : DEPT.DBF

?	.NDX	fnam	width	type	description	recl = 126
D_D_ID	d_id		2	char	department identification code	
	x		2	char	< unused >	
	dnam		32	char	name of department	
	dadd		32	char	address of department (e.g. building)	
	dcty		32	char	city	
	budg		8:2	N	relevant budget (if any, in Taka)	
	dt_d		8	D	< unused >	
	remk		8	char	remarks, if any	
	flag		1	char	log of reports generated	
	stat		1	char	status_code	

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 <accn> 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

?	.NDX	fnam	width	type	description	recl = 126
TITL		titl	64	char	title of the book	
AUTH		auth	32	char	author	
		year	2	char	year of publication	
CALL		call	11	char	call number	
ACCN		accn	5	char	accession number	
SUBJ		subj	3	char	subject classification [primary]	
SUB2		sub2	3	char	subject classification [secondary]	
		p_id	2	char	publisher_id (code)	
		d_id	2	char	requisitioning department_id (code)	
		flag	1	char	log of reports generated	
		stat	1	char	status_code	

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 <ncur>, borrowing

limit <blim>, time limit [tlim], and lastly, date of expiry [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

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

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 <accn> which respectively are 'keys' to the user data base USER.DBF and the book data base BOOK.DBF. In addition to the usual <flag> and <stat> fields a 9-character remarks field <remk> 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

?	.NDX	fnam	width	type	description	recl = 30
L_U_ID		u_id	6	char	user identification number	
L_ACCN		accn	5	char	accession number of book borrowed	
		ddue	8	D	date book due for return	
		remk	9	char	remarks, if any	
		flag	1	char	log of reports generated	
		stat	1	char	status_code	

TABLE 4.8 STRUCTURE OF THE RESERVATION QUEUE FILE : RQUE.DBF

?	.NDX	fnam	width	type	description	recl = 30
R_U_ID		u_id	6	char	user identification number	
R_ACCN		accn	5	char	accession number of book reserved	
		dreq	8	D	date reservation recorded	
		remk	9	char	remarks, if any	
		flag	1	char	log of reports generated	
		stat	1	char	status_code	

CHAPTER 5

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 5.1.2 and 5.1.3 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.

L-DBMS REPORTS	LR
 1. Acquisitions 2. Rental Management 3. Main Menu 	

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 | 5.1.1 |
| 2) books ordered (sorted on departments) | 5.1.2 |
| 3) books ordered (sorted on supplier) | 5.1.3 |
| 4) new arrivals (sorted on BOOK->record#) | 5.1.4 |
| 5) new arrivals (sorted on author) | 5.1.5 |
| 6) new arrivals (sorted on subject) | 5.1.6 |
| 7) new arrivals (sorted on title) | 5.1.7 |
| 8) books unavailable (sorted on departments) | 5.1.8 |

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

```
-----  
L-DBNS  REPORTS * ACQUISITIONS  LRA  
  
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.

- 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 a option here. It is automatically generated whenever a book is issued [by selecting the appropriate option in the UPDATES * RENTAL menu]

```

L-DBMS REPORTS * RENTAL MANAGEMENT LRR

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 2-PART ISSUE CARD
output type Hardcopy, 132 col. [vert. perforation at col.66]
purpose [1] Keeping track of issues [2] use as Gate Pass
generation Per issue
variants
copies [1], detachable in two parts [a] [b]
distribution [a] to be preserved at the issue counter
[b] to be shown at the library exit

```

<titl> : <titl>
<auth> : <auth>
<call> <accn> <cost> : <call> <accn> <cost>
date issued = : date issued =
date due = <ddue> : date due = <ddue>
borrower = <u_id> <unam> : borrower = <u_id> <unam>
: <uadd> : <uadd>
: <ucty> : <ucty>

```

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

```

date
<subj> <call> <titl> <auth> <p_id>                   <ddue>
<subj> <call> <titl> <auth> <p_id>                   <ddue>
<subj> <call> <titl> <auth> <p_id>                   <ddue>
..                   ..                   ..                   ..                   ..
<subj> <call> <titl> <auth> <p_id>                   <ddue>
..                   ..                   ..                   ..                   ..
..                   ..                   ..                   ..                   ..

```

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

```

date =
<accn> <titl> <auth> <call>                           <u_id> <ddue>
<accn> <titl> <auth> <call>                           <u_id> <ddue>
<accn> <titl> <auth> <call>                           <u_id> <ddue>
..                   ..                   ..                   ..                   ..
<accn> <titl> <auth> <call>                           <u_id> <ddue>
..                   ..                   ..                   ..                   ..
..                   ..                   ..                   ..                   ..

```

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

```

date =
<accn> <titl> <auth> <call>                           <u_id> <ddue> days
<accn> <titl> <auth> <call>                           <u_id> <ddue> ov'due
<accn> <titl> <auth> <call>                           <u_id> <ddue>
..                   ..                   ..                   ..                   ..
<accn> <titl> <auth> <call>                           <u_id> <ddue>
..                   ..                   ..                   ..                   ..
..                   ..                   ..                   ..                   ..

```


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:

```

L-DBMS  INQUIRIES'                                     LN
          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
purpose	Looking up orders information from the data base
generation	As required
access	Authorized library staff < inquiry screen >

```

L-DBMS  INQUIRIES * ORDERS                             LNO
          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

```

< output screen >

```
title XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
author XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
location XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
publisher XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
year of publication XXXX
price, in Taka nnnnn.nn
copies to order nnn
subject code [ primary ] XXX
subject code [ secondary ] XXX
date ordered mm-dd-yy
supplier XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
publisher_id XX
requisitioning department XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
remarks, if any XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
report_flag X
status_flag X
```

5.3.2 BOOK CATALOGUE INQUIRY
purpose Looking up book information from the data base
generation As required
access All users of the library
< inquiry screen >

```
L-DBMS INQUIRIES * BOOK_STATUS LNB

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
```

< output screen >

```
title XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
author XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
location XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
publisher XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
year of publication XXXX
call number XXXXXXXXXXXXXXXX
accession number XXXXX
subject code [ primary ] XXX
subject code [ secondary ] XXX
requisitioning department XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
report_flag X
status_flag X
```

5.3.3

USER STATUS INQUIRY

purpose

Looking up user information from the data base

generation

As required

access

Authorized library staff

< inquiry screen >

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

< output screen >

```

user_id          XXXXXX
name             XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
address          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
department       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
date of expiry   mm-dd-yy
borrowing limit  nn
books borrowed   nn
books overdue [ accession number, title, date due ]
nnnnn XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX mm-dd-yy
..
publisher        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
remarks, if any  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
report_flag      X
status_flag      X

```

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.

TABLE 6.1 AN OUTLINE SHOWING THE LOGICAL STRUCTURE OF THE SYSTEM AND THE HIERARCHY OF PROGRAM MODULES

program module	function group	DATABASE/ subsystem description . . .	Program_id [6- . . .
L			< menu >
LN	INQUIRY		< menu >
LNO	INQUIRY	: ORDER	< menu >
LNOO	INQUIRY	: ORDER by date placement of order	1-1-1
LNOP	INQUIRY	: ORDER by publisher	1-1-2
LNOD	INQUIRY	: ORDER by requisitioning department	1-1-3
LNO2	INQUIRY	: ORDER by subject_code [composite]	1-1-4
ENOJ	INQUIRY	: ORDER by subject_code [primary]	1-1-5
LNOA	INQUIRY	: ORDER by supplier/agent	1-1-6
LNOF	INQUIRY	: ORDER by r_flag	1-1-7
LNOS	INQUIRY	: ORDER by s_flag	1-1-8
LN	INQUIRY	: BOOK	< menu >
LNBN	INQUIRY	: BOOK by accession number	1-2-1
LNBA	INQUIRY	: BOOK by author	1-2-2
LNBC	INQUIRY	: BOOK by call number	1-2-3
LNB2	INQUIRY	: BOOK by subject_code [composite]	1-2-4
LNB3	INQUIRY	: BOOK by subject_code [primary]	1-2-5
LNBT	INQUIRY	: BOOK by title	1-2-6
LNBF	INQUIRY	: BOOK by r_flag	1-2-7
LNBS	INQUIRY	: BOOK by s_flag	1-2-8
LN	INQUIRY	: USER	< menu >
LNUX	INQUIRY	: USER by date of expiry	1-3-1
LNUC	INQUIRY	: USER by user_city	1-3-2
LNUD	INQUIRY	: USER by user_department	1-3-3
LNUN	INQUIRY	: USER by user_id_number	1-3-4
LNUA	INQUIRY	: USER by user_name	1-3-5
LNUT	INQUIRY	: USER by user_type	1-3-6
LNUF	INQUIRY	: USER by r_flag	1-3-7
LNUS	INQUIRY	: USER by s_flag	1-3-8
LR	REPORTS		< menu >
LRA	REPORTS	: acqn	< menu >
LRAO	REPORTS	: acqn orders to suppliers	2-1-1
LRAD	REPORTS	: acqn books ordered, by dept.	2-1-2
LRAS	REPORTS	: acqn books ordered, by supp.	2-1-3
LRAR	REPORTS	: acqn new arrivals, by record#	2-1-4
LRAA	REPORTS	: acqn new arrivals, by author	2-1-5
LRAJ	REPORTS	: acqn new arrivals, by subject	2-1-6
LRAT	REPORTS	: acqn new arrivals, by title	2-1-7
LRAU	REPORTS	: acqn books unavailable	2-1-8

continued...

TABLE 6.1

...continued

program module	function group	DATABASE/ subsystem	description . . .	Program_id [6- ...
LRR	REPORTS	: rent	< menu >	
LRRL	REPORTS	: rent	books still on loan	2-2-1
LRRI	REPORTS	: rent	books issued	2-2-2
LRRR	REPORTS	: rent	books returned	2-2-3
LRRS	REPORTS	: rent	overdue reminder _students	2-2-4
LRRT	REPORTS	: rent	overdue reminder _teachers	2-2-5
LRRV	REPORTS	: rent	reserve_book coll. slip	2-2-6
LRRF	REPORTS	: rent	final fine report	2-2-7
LU	UPDATES		< menu >	
LUR	UPDATES	: rent	< menu >	
LURI	UPDATES	: rent	issue a book	3-1-1
LURR	UPDATES	: rent	record return of a book	3-1-2
LURV	UPDATES	: rent	reserve a book	3-1-3
LUT	UPDATES		transfer to BOOK.DBF	3-2-0
LUD	UPDATES	: DEPT	< menu >	
LUDA	UPDATES	: DEPT	add	3-3-1
LUDC	UPDATES	: DEPT	change	3-3-2
LUDD	UPDATES	: DEPT	delete	3-3-3
LUO	UPDATES	: ORDR	< menu >	
LUOA	UPDATES	: ORDR	add	3-4-1
LUOC	UPDATES	: ORDR	change	3-4-2
LUOD	UPDATES	: ORDR	delete	3-4-3
LUP	UPDATES	: PUBL	< menu >	
LUPA	UPDATES	: PUBL	add	3-5-1
LUPC	UPDATES	: PUBL	change	3-5-2
LUPD	UPDATES	: PUBL	delete	3-5-3
LUS	UPDATES	: SUPP	< menu >	
LUSA	UPDATES	: SUPP	add	3-6-1
LUSC	UPDATES	: SUPP	change	3-6-2
LUSD	UPDATES	: SUPP	delete	3-6-3
LUU	UPDATES	: USER	< menu >	
LUUA	UPDATES	: USER	add	3-7-1
LUUC	UPDATES	: USER	change	3-7-2
LUUD	UPDATES	: USER	delete	3-7-3

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

L-DBMS	INQUIRIES	LN
	1. Orders	
	2. Books	
	3. Users	
	4. Main Menu	

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:

6-1-1-1	LNOO.prg	
function	Inquire orders by date of placement of order	
input	date	
data base	ORDR.DBF [r]	index used: O_DORD.NDX [r]

6-1-1-2	LNOP.prg	
function	Inquire orders data base by publisher	
input	publisher_id	
data base	ORDR.DBF [r]	index used: O_P_ID.NDX [r]

6-1-1-3	LNOD.prg	
function	Inquire orders data base by requisitioning dept.	
input	department_id	
data base	ORDR.DBF [r]	index used: O_D_ID.NDX [r]

6-1-1-4	LNO2.prg	
function	Inquire orders data base by comp. subject code	
input	<subj> <sub2>	
data base	ORDR.DBF [r]	index used: O_SUB2.NDX [r]

6-1-1-5	LNOJ.prg	
function	Inquire orders data base by pry. subject code	
input	subject code	
data base	ORDR.DBF [r]	index used: O_SUBJ.NDX [r]

6-1-1-6	LNOA.prg	
function	Inquire orders data base by supplier/agent	
input	supplier_id	
data base	ORDR.DBF [r]	index used: O_S_ID.NDX [r]

6-1-1-7	LNOF.prg	
function	Display record(s) having <flag> = input	
input	Any character in the set @ABCDEFGHIJKLMNO	
data base	ORDR.DBF [r]	index used: O_FLAG.NDX [r]

6-1-1-8	LNOS.prg	
function	Display record(s) having <stat> = input	
input	Any character in the set @ABCDEFGHIJKLMNO	
data base	ORDR.DBF [r]	index used: O_STAT.NDX [r]

6-1-2-1 function input data base	LNBN.prg Inquire book by accession number accession number BOOK.DBF [r]	index used: ACCN.NDX [r]
6-1-2-2 function input data base	LNBA.prg Inquire book data base by author publisher_id BOOK.DBF [r]	index used: AUTH.NDX [r]
6-1-2-3 function input data base	LNBC.prg Inquire book data base by call number call number BOOK.DBF [r]	index used: CALL.NDX [r]
6-1-2-4 function input data base	LNB2.prg Inquire book data base by comp. subject code <subj> and <sub2> BOOK.DBF [r]	index used: SUB2.NDX [r]
6-1-2-5 function input data base	LNBJ.prg Inquire book data base by pry. subject code subject code BOOK.DBF [r]	index used: SUB2.NDX [r]
6-1-2-6 function input data base	LNBA.prg Inquire book data base by supplier/agent supplier_id BOOK.DBF [r]	index used: TITL.NDX [r]
6-1-2-7 function input data base	LNBF.prg Display record(s) having <flag> = input Any character in the set @ABCDEFGHIJKLMNO BOOK.DBF [r]	index used: FLAG.NDX [r]
6-1-2-8 function input data base	LNBS.prg Display record(s) having <stat> = input Any character in the set @ABCDEFGHIJKLMNO BOOK.DBF [r]	index used: STAT.NDX [r]
6-1-3-1 function input data base	LNUX.prg Inquire users by date of expiry date USER.DBF [r]	index used: U_DEXP.NDX [r]
6-1-3-2 function input data base	LNUC.prg Inquire users data base by user_city name of city USER.DBF [r]	index used: U_UCTY.NDX [r]

6-1-3-3 LNUD.prg
function Inquire users data base by requisitioning dept.
input department_id
data base USER.DBF [r] index used: U_D_ID.NDX [r]

6-1-3-4 LNUN.prg
function Inquire users data base by user_id number
input <subj> <sub2>
data base USER.DBF [r] index used: U_U_ID.NDX [r]

6-1-3-5 LNUA.prg
function Inquire users data base by user name
input user name
data base USER.DBF [r] index used: U_UNAM.NDX [r]

6-1-3-6 LNUT.prg
function Inquire users data base by user type
input supplier_id
data base USER.DBF [r] index used: U_TYPE.NDX [r]

6-1-3-7 LNUF.prg
function Display record(s) having <flag> = input
input Any character in the set @ABCDEFGHIJKLMNO
data base USER.DBF [r] index used: U_FLAG.NDX [r]

6-1-3-8 LNUS.prg
function Display record(s) having <stat> = input
input Any character in the set @ABCDEFGHIJKLMNO
data base USER.DBF [r] index used: U_STAT.NDX [r]

6.2 GENERATING REPORTS

L-DBMS	REPORTS	LR
	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:

6-2-1-1 LRAO.prg
function Prints requisition to individual suppliers
input none [covers all new entries in ORDR.DBF]
data base ORDR.DBF [r/w] index used: O_S_ID.NDX [r]
 SUPP.DBF [r] index used: S_S_ID.NDX [r]

6-2-1-2 LRAD.prg
function Prints list of books ordered [sorted by dept.]
input none [covers all new entries in ORDR.DBF]
data base ORDR.DBF [r/w] index used: O_D_ID.NDX [r]

6-2-1-3 LRAS.prg
function Prints list of books ordered [sorted by supp.]
input none [covers all new entries in ORDR.DBF]
data base ORDR.DBF [r/w] index used: O_S_ID.NDX [r]

6-2-1-4 LRAR.prg
function Prints list of new arrivals [sorted by recnum]
input none [covers all new receipts]
data base ORDR.DBF [r/w] index used:

6-2-1-5 LRAA.prg
function Prints list of new arrivals [sorted by author]
input none [covers all new receipts]
data base ORDR.DBF [r/w] index used: O_AUTH.NDX [r]

6-2-1-6 LRAJ.prg
function Prints list of new arrivals [sorted by subject]
input none [covers all new receipts]
data base ORDR.DBF [r/w] index used: O_SUBJ.NDX [r]

6-2-1-7 LRAT.prg
function Prints list of new arrivals [sorted by title]
input none [covers all new receipts]
data base ORDR.DBF [r/w] index used: O_TITL.NDX [r]

6-2-1-8 LRAU.prg
function Prints list of books unavailable
input none [covers current orders and receipts]
data base ORDR.DBF [r/w] index used: O_D_ID.NDX [r]

6-2-2-1 LRRL.prg
function Prints list of books currently on loan
input none [all books out on loan covered]
data base BOOK.DBF [r] index used: SUBJ.NDX [r]

6-2-2-2	LRR1.prg		
function	Prints list of books issued on given date		
input	date		
data base	BOOK.DBF [r]	index used:	ACCN.NDX [r]
	LOAN.DBF [r]	index used:	ACCN.NDX [r]

6-2-2-3	LRRR.prg		
function	Prints list of books returned on given date		
input	date		
data base	BOOK.DBF [r]	index used:	ACCN.NDX [r]
	LOAN.DBF [r]	index used:	ACCN.NDX [r]

6-2-2-4	LRRS.prg		
function	Prints overdue reminder slip [for students]		
input	date		
data base	BOOK.DBF [r]	index used:	ACCN.NDX [r]
	LOAN.DBF [r]	index used:	L_ACCN.NDX [r]
			L_U_ID.NDX [r]
	USER.DBF [r]	index used:	U_U_ID.NDX [r]

6-2-2-5	LRRT.prg		
function	Prints overdue reminder slip [for teachers]		
input	date		
data base	BOOK.DBF [r]	index used:	ACCN.NDX [r]
	LOAN.DBF [r]	index used:	L_ACCN.NDX [r]
			L_U_ID.NDX [r]
	USER.DBF [r]	index used:	U_U_ID.NDX [r]

6-2-2-6	LRRV.prg		
function	Prints reserved book collection request		
input	date		
data base	BOOK.DBF [r]	index used:	ACCN.NDX [r]
	RQUE.DBF [r]	index used:	R_ACCN.NDX [r]
			R_U_ID.NDX [r]
	USER.DBF [r]	index used:	U_U_ID.NDX [r]

6-2-2-7	LRRF.prg		
function	Prints final fine report		
input	user_id		
data base	USER.DBF [r]	index used:	U_U_ID.NDX [r]

6.3 UPDATES AND MAINTENANCE

L-DBMS	UPDATE DATABASES	LU
<ol style="list-style-type: none"> 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:

6-3-1-1	LURI.prg		
function	Record issue of book, print 2-part issue card		
input	user_id of borrower, accession number of book		
data base	BOOK.DBF [r/w]	index used:	ACCN.NDX [r]
	LOAN.DBF [r/w]	index used:	L_ACCN.NDX [r/w]
			L_U_ID.NDX [r/w]
	USER.DBF [r/w]	index used:	U_U_ID.NDX [r]

6-3-1-2	LURR.prg		
function	Record return of book, print 2-part issue card		
input	user_id of borrower, accession number of book		
data base	BOOK.DBF [r/w]	index used:	ACCN.NDX [r]
	LOAN.DBF [r/w]	index used:	L_ACCN.NDX [r/w]
			L_U_ID.NDX [r/w]
	USER.DBF [r/w]	index used:	U_U_ID.NDX [r]

6-3-1-3	LURV.prg		
function	Reserve a book		
input	user_id of borrower, accession number of book		
data base	BOOK.DBF [r/w]	index used:	ACCN.NDX [r]
	RQUE.DBF [r/w]	index used:	R_ACCN.NDX [r/w]
			R_U_ID.NDX [r/w]
	USER.DBF [r/w]	index used:	U_U_ID.NDX [r]

6-3-2-0	LUT.prg		
function	Transfer records having <stat>=input to BOOK.DBF		
input	appropriate status code		
data base	ORDR.DBF [r/w]	index used:	O_????.NDX [r/w]
	BOOK.DBF [r/w]	index used:	????.NDX [r/w]

6-3-3-1	LUDA.prg	> add	[input : none
6-3-3-2	LUDC.prg	> change	[input : department_id
6-3-3-3	LUDD.prg	> delete	[input : department_id
function	Modify data base		
data base	DEPT.DBF [r/w]	index used:	D_????.NDX [r/w]

6-3-4-1 LUOA.prg > add [input : none
6-3-4-2 LUOC.prg > change [input : ORDR->record #
6-3-4-3 LUOD.prg > delete [input : ORDR->record #
function Modify data base
data base ORDR.DBF [r/w] index used: O_????.NDX [r/w]

6-3-5-1 LUPA.prg > add [input : none
6-3-5-2 LUPC.prg > change [input : publisher_id
6-3-5-3 LUPD.prg > delete [input : publisher_id
function Modify data base
data base PUBL.DBF [r/w] index used: P_????.NDX [r/w]

6-3-6-1 LUSA.prg > add [input : none
6-3-6-2 LUSC.prg > change [input : supplier_id
6-3-6-3 LUSD.prg > delete [input : supplier_id
function Modify data base
data base SUPP.DBF [r/w] index used: S_????.NDX [r/w]

6-3-7-1 LUUA.prg > add [input : none
6-3-7-2 LUUC.prg > change [input : user_id
6-3-7-3 LUUD.prg > delete [input : user_id
function Modify data base
data base USER.DBF [r/w] index used: U_????.NDX [r/w]

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

Cardenas, A.F., Data Base Management Systems, Boston, MA: Allyn and Bacon, 1979.

Date, C.J., An Introduction to Database Systems, Volume 1, Reading, MA: Addison-Wesley, 1981.

Date, C.J., Database: A Primer, Reading, MA: Addison-Wesley, 1983.

McLeod, R. Jr., Management Information Systems, Chicago, IL: Science Research Associates, 1983.

McNichols, C.W. and S.F. Rushinek, Data Base Management: A Microcomputer Approach, Englewood Cliffs, NJ: Prentice-Hall, 1988.

Sandberg, G., "A Primer on Relational Data Base Concepts", IBM Systems Journal, 20(1), 1981, 23-40.

Senn, J.A., Information Systems in Management, Belmont, CA: Wadsworth, 1982.

Simpson, A., dBASE III Plus Programmers Reference Guide, Alameda, CA: Sybex, 1987.

APPENDIX

1 program source code listing, identified by filename 1

```

                                                                    L.PRG
set      talk      off          && this is the main menu
set      proc to LBAR          && procedure to enable light_bar
optn = 4          && no. of options at this level
os      = 0
do while os # optn
    opt1 = "      INQUIRY      "
    opt2 = "      REPORTS     "
    opt3 = "      UPDATES     "
    opt4 = "..... Quit to dBASE "
    optn = 4
titl = "L-DBMS LIBRARY DATA BASE MANAGEMENT SYSTEM          L"
clear          && clear screen
do LBAR with optn,titl && invoke light_bar menu
    do case          && identify choice..
        case os = 1
            do LN          && invoke inquiries menu
        case os = 2
            do LR          && invoke reports menu
        case os = 3
            do LU          && invoke updates menu
    endcase
enddo
clear          && clear screen

```

```

                                                                    LN.PRG
set      procedure to LBAR      && this is the inquiries menu
optn = 4
os      = 0
do while os # optn
    opt1 = "Orders      "
    opt2 = "Books      "
    opt3 = "Users      "
    opt4 = "Main Menu  "
    optn = 4
titl = "L-DBMS INQUIRIES          LN"
clear          && clear screen
do LBAR with optn,titl && invoke light_bar menu
    do case          && identify choice..
        case os = 1
            do LND          && orders inquiry
        case os = 2
            do LNB          && book inquiry
        case os = 3
            do LNU          && users inquiry
    endcase
enddo
os = 0
return          && previous menu

```

```

                                                                    LNB.PRG
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
titl = "L-DBMS INQUIRIES * BOOK_STATUS"
clear                                       && clear screen
do LBAR with optn,titl                    && invoke light_bar menu
do case                                     && identify choice..
    case os = 1
        do LNBT                    && by title
    case os = 2
        do LNBA                    && by author_name
    case os = 3
        do LNBC                    && by call number
    case os = 4
        do LNBN                    && by accession number
    case os = 5
        do LNBJ                    && by subject code
    case os = 6
        do LNE2                    && by composite subject
    case os = 7
        do LNBF                    && by report_flag
    case os = 8
        do LNBS                    && by status_flag
endcase
enddo
os = 0
return                                     && previous menu

```

```

                                                                    LNB2.PRG
set      talk        off                && inquiry on composite subject
set      stat        off
clear                                         && clear screen
@ 01,00 to 03,78                               && draw box for header
@ 02,15 say "LNBJ - 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 clear                               && clear screen display area

```

```

@ 15,33      say "<cr> to exit..."
@ 12,01      say "? subject code [ primary ] " get m_subp
              read          && get primary code
              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
                  do book_d      && display book info.
                  skip
                enddo
@ 23,02      wait "[ press X to interrupt ]          -- More --" to m_next
              if upper(m_next) = 'X'
@ 23,00      clea
              exit
              endif
              enddo
              else
@ 15,33      say "^ No such subject code combination."
              endif
@ 23,02      wait          && wait until key pressed
              enddo
              use          && close data bases
set          stat      on
              return      && previous menu

```

```

LNBA.PRG
set          stat      off          && inquiry on author_name
set          talk      off
              clea          && clear screen
@ 01,00 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 PUBL
              m_auth = "?"          && author_name inquired
              m_next = "?"          && continuation code ( X exits
do while m_auth # " "
              m_auth = space(32) && initialize memory variable
@ 04,00      clea          && clear screen display area
@ 14,13      say "<cr> to exit..."
@ 12,01      say "? author " get m_auth
              read          && get author_name inquired
              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)
                  do book_d      && display book info.
                  skip
                enddo
@ 23,02

```

```

wait "[ press X to interrupt ]" -- More --" to m_next
if upper(m_next) = 'X'
@ 23,00   clea
          exit
endif
          enddo
else
@ 14,13   say "" No such author."
endif
@ 23,02
          wait          && wait until key pressed
          enddo
          use          && close data bases
set       stat      on
          return      && previous menu

```

```

LNBC.PRG
set       stat      off          && inquiry on call number
set       talk      off
          clea          && clear screen
@ 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
@ 14,19   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          && wait until key pressed
          enddo
          use          && close data bases
set       stat      on
          return      && previous menu

```

```

LNBF.PRG
set       stat      off
set       talk      off
          clea
@ 01,00 to 03,78
@ 02,25 say "LNBF -- book inquiry by report_flag"
          select A
          use BOOK inde flag
          select B

```

```

use      PUBL inde p_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))
                do book_d      && display book info.
                    skip
            enddo
@ 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 report_flag."
    endif
@ 23,02  wait
        && wait until key pressed
    enddo
use      && close data bases
set      stat on
return   && previous menu

```

LNBJ.PRG

```

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_p_id
select A
set      rela to p_id into PUBL
m_subj = "?"
m_next = "?"
do while m_subj # " "
    m_subj = space(3)
@ 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)

```

```

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

```

LNBN.FRG

```

set      stat    off
set      talk    off
        clea
@ 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
        clea
@ 14,24
        say "<cr> to exit. . ."
@ 12,01
        say "? accession number" " get m_accn
        read
            if m_accn = " "
                loop
            endif
        seek upper(m_accn)
        if found()
            do book_d && display book information
        else
@ 14,24
            say "^ No such accession number."
        endif
@ 23,02
        wait      && wait until key pressed
        enddo
        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      "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
@ 14,18  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 book_d      && display book info.
                skip
@ 23,02  wait "l press X to interrupt j"      -- 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      && close data bases
set      stat      on
        return    && previous menu

```

```

set      stat      off
set      talk      off
        clea
@ 01,00  to 03,78
@ 02,25  say      "LNBT - 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_titl = space(64)
@ 04,00 clea
@ 14,13 say "<cr> to exit..."
@ 12,01 say "? title " get m_titl
read
    if m_titl = " "
        loop
    endif
seek upper(m_titl)
    if found()
        do book_d  && display book information
    else
@ 14,13     say "^ No such title."
    endif
@ 23,02
    wait          && wait until key pressed
enddo
use             && close data bases
set stat on
return         && previous menu

```

LNO.PRG

```

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 [pry.] "
        opt5 = "by subj_code [comp]"
        opt6 = "by Publisher "
        opt7 = "by report_gen_Flag "
        opt8 = "by Status_flag "
        opt9 = "Previous Menu "
        optn = 9
titl = "L-DBMS INQUIRIES & ORDERS"
clear
do LBAR with optn,titl
    do case  && identify choice..
        case os = 1
            do LNOA  && by supplier/agent
        case os = 2
            do LNOD  && by department
        case os = 3
            do LNOD  && by date of order
        case os = 4
            do LNOJ  && by subject code
        case os = 5
            do LNOD  && by composite subject
        case os = 6
            do LNOD  && by publisher
        case os = 7
            do LNOD  && by report_flag
        case os = 8
            do LNOS  && by status_flag
    endcase
    enddo
    os = 0
return && previous menu

```



```

set      talk      off
set      stat      off
clea

@ 01,00 to 03,78
@ 02,15 say "LN02 - orders inquiry by subject code [ composite ]"
select A
use      ORDR inde o_sub2
select B
use      DEPT inde 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(8)
@ 04,00 clea
@ 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(subj)+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'
@ 23,00 clea
        exit
    endif
        enddo
    else
@ 15,33 say "^ No such subject code combination."
        endif
@ 23,02 wait && wait until key pressed
        enddo
        use && close data bases
set      stat      on
return && previous menu

```

```

set      stat      off
set      talk      off
clea
@ 01,00 to 03,78
@ 02,15 say "LN0A - orders inquiry by supplier code"
select A
use      ORDR inde o_s_id
select B

```

```

use DEPT inde d_d_id
select 0
set rela to s_id into DEPT
m_s_id = "?"
m_next = "?"
do while m_s_id # " "
    m_s_id = space(2)
@ 04,00    clea
@ 14,20    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 ordr_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
    enddo
else
@ 14,20    say "% No such supplier code."
endif
@ 23,02
wait : && wait until key pressed
enddo
use : && close data bases
set stat on
return : && previous menu

```

LNOD.PRG

```

set stat off
set talk off
clea
@ 01,00 to 03,78
@ 02,15 say "LNOD - orders inquiry by department code"
select 0
use ORDR inde o_d_id
select 0
use DEPT inde d_d_id
select 0
set rela to d_id into DEPT
m_d_id = "?"
m_next = "?"
do while m_d_id # " "
    m_d_id = space(2)
@ 04,00    clea
@ 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

```

```

        if found()
            do while d_id=m_d_id
                do ordr_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
        enddo
    else
@ 14,22        say "^ No such department code."
    endif
@ 23,02
        wait      && wait until key pressed
    enddo
    use      && close data bases
set  stat  on
    return      && previous menu

```

LNDF.PRG

```

set  stat  off
set  talk  off
clea
@ 01,00 to 03,78
@ 02,25 say  "LNDF - orders inquiry by report_flag"
        select A
        use  ORDR inde o_flag
        select B
        use  DEPT inde d_d_id
        select A
        set  rela to d_id into DEPT
        m_flag = "?"
        m_next = "?"
        do while m_flag # " "
            m_flag = spacc(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))
                do ordr_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
        enddo
    else
@ 14,18        say "^ No such report_flag."
    endif
@ 23,02

```

```

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

```

LNQJ.PRG

```

set     stat      off
set     talk      off
    clea
@ 01,00 to 03,78
@ 02,25 say "LNQJ - orders inquiry by subject code"
    select A
    use     ORDR inde o_subj
    select B
    use     DEFT inde d_d_id
    select A
    set     rela to d_id into DEPT
    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)
        if found()
            do while subjupper(m_subj)
                do ordr_d      && display orders info.
                    skip
@ 23,02
wait "[ press X to interrupt ]" -- More --" to m_next
            if upper(m_next) = 'X'
                clea
                exit
            endif
        enddo
    else
@ 14,19    say "" No such subject code."
    endif
@ 23,02
        wait                && wait until key pressed
    enddo
    use                    && close data bases
set     stat      on
    return                && previous menu

```

LNQD.PRG

```

set     stat      off
set     talk      off
    clea
@ 01,00 to 03,78
@ 02,15 say "LNQD - orders inquiry by date [ ..on or after ]"
    select A
    use     ORDR inde o_dord
    select B

```

```

use      DEPT inde d_d_id
select  A
set     rela 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 ordr_id      && display orders info.
                skip
@ 23,02  wait "L press X to interrupt ]          -- More --" to m_next
        if upper(m_next) = 'X'
@ 23,00  clea
        exit
        endif
        enddo
    else
@ 14,15  say "^ No entry against this date."
    endif
@ 23,02  wait          && wait until key pressed
        enddo
use          && close data bases
set     stat  on
return   && previous menu

```

LNOP.PRG

```

set     stat  off
set     talk  off
clea
@ 01,00 to 03,78
@ 02,15 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     rela 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
                                enddo
        else
@ 14,21            say "^ No such publisher code."
        endif
@ 23,02
                wait      && wait until key pressed
            enddo
        use      && close data bases
set  stat  on
return      && previous menu

```

LNOS.PRG

```

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 # " "
@ 04,00            m_stat = space(1)
        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 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
                                enddo
        else
@ 14,18            say "^ No such status_flag."
        endif
@ 23,02

```

```

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

```

LNU.PRG

```

set     procedure    to LBAR    && user inquiry menu
    optn = 9
    os = 0
    do while os # optn
        opt1 = "by user_id_Number  "
        opt2 = "by user_Type        "
        opt3 = "by user_Name        "
        opt4 = "by date_of_eXpiry   "
        opt5 = "by fines due        "
        opt6 = "by user_Department  "
        opt7 = "by report_gen_flag  "
        opt8 = "by Status_flag      "
        opt9 = "Previous Menu       "
        optn = 9
        titl = "L-DBMS  INQUIRIES * USER_STATUS"

        clear
        do LBAR with optn,titl
            do case                && identify choice..
                case os = 1        && by user_id_number
                    do LNUN
                case os = 2        && by user type code
                    do LNUT
                case os = 3        && by user name
                    do LNUA
                case os = 4        && by date of expiry
                    do LNUX
                case os = 5        && those with fines due
                    do LNUC
                case os = 6        && by user department
                    do LNUD
                case os = 7        && by report_flag
                    do LNUF
                case os = 8        && by status_flag
                    do LNUS
            endcase
        enddo
        os = 0
        return                && previous menu

```

LNU"

LNUA.PRG

```

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 inde u_unam
        select B
        use      DEPT inde d_d_id
        select A
        set     rela to d_id into DEPT
        m_unam = "?"
        do while m_unam # " "

```

```

        m_unam = space(32)
@ 04,00      clea
@ 14,17      say "<cr> to exit..."
@ 12,01      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
@ 14,17      say "^ No such user."
              endif
@ 23,02
              wait          && wait until key pressed
            enddo
            use             && close data bases
set          stat         on
            return        && previous menu.

```

LNUC.PRG

```

set          stat         off
set          talk         off
              clea
@ 01,00 to 03,78
@ 02,15      say "LNUX - 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
@ 23,02
wait "[ press X to interrupt ]      -- More --" to m_next
              if upper(m_next) = 'X'
@ 23,00      clea
              exit
              endif
            enddo
@ 23,02
              wait          && wait until key pressed
            use             && close data bases
set          stat         on
            return        && previous menu

```

LNUD.PRG

```

set          stat         off
set          talk         off
              clea
@ 01,00 to 03,78
@ 02,25      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      rela to d_id into DEPT
m_d_id = "?"
m_next = "?"
do while m_d_id # " "
    m_d_id = space(2)
@ 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 use 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
enddo
else
@ 14,22   say "No such user department."
endif
@ 23,02   wait                && wait until key pressed
enddo
use                && close data bases
set stat on
return            && previous menu

```

LNUF.PRG

```

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      rela to d_id into DEPT
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))
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
enddo
else
@ 14,18      say "^ No such report_flag."
endif
@ 23,02
wait      && wait until key pressed
enddo
use      && close data bases
set stat on
return   && previous menu

```

LNUN.PRG

```

set stat off
set talk off
clea
@ 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)
@ 04,00 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
@ 14,17 say "^ No such user_id."
endif
@ 23,02
wait      && wait until key pressed
enddo
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 "LNUS - user inquiry by status_flag"
select A
use      USER inde u_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)
@ 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
@ 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          && close data bases
set      stat      on
return          && previous menu

```

```

set      stat      off
set      talk      off
clea
@ 01,00 to 03,78
@ 02,25 say "LNUT - user inquiry by user type"
select A
use      USER inde u_u_ty
select B
use      DEPT inde d_d_id
select A
set      rela to d_id into DEPT
m_u_ty = "?"
m_next = "?"

```

```

do while m_u_ty # " "
    m_u_ty = space(1)
@ 04,00    clea
@ 14,16    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
                    enddo
                else
@ 14,16                    say "^ No such user type."
                endif
@ 23,02
                wait      && wait until key pressed
            enddo
            use      && 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_id into DEPT
m_dexp = "?"
m_next = "?"
do while m_dexp # " "
    m_dexp = space(8)
@ 04,00    clea
@ 14,16    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'
@ 23,00                                clea
                                        exit
                                endif
                                enddo
else
@ 14,15                                say " No entry in the range specified."
                                endif
@ 23,02
                                wait                                && wait until key pressed
                                enddo
                                use                                && close data bases
set    stat    on
                                return                            && previous menu

```

```

                                                                    LR.PRG
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      LR"
clear
do LBAR with optn,titl
    do case                                && identify choice..
        case os = 1
            do LRA                                && acquisitions
        case os = 2
            do LRR                                && rental management
    endcase
enddo
os = 0
return                                && previous menu

```

```

                                                                    LRA.PRG
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  LRA"
clear
do LBAR with optn,titl
    do case                                && identify choice..
        case os = 1
            do LRAD                                && orders to suppliers
        case os = 2
            do LRAD                                && books ordered by dept.

```

```

                                case os = 3
                                  do LRAS      && books ordered by supp.
                                case os = 4
                                  do LRAU      && books unavailable
                                case os = 5
                                  do LRAR      && orders, by record #
                                case os = 6
                                  do LRAJ      && orders, by subject
                                case os = 7
                                  do LRAT      && orders, by title
                                case os = 8
                                  do LRAA      && orders, by author
                                endcase
                                enddo
                                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 LRAD
                                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

```

LRAD.PRG

```
clear
set status off
close all
erase suppodr.dbf
select a
use ordr index o_s_id
select b
use supp index s_s_id
select a
join with supp to suppodr for s_id=b->s_id fields titl,auth,
year,copy,pnam,b->snam,b->sadd,b->scly
close all
use suppodr
report form LRAD
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
```

LRAS.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 inde o_titl
report form LRAN for flag='N'
close all
set stat on
return && previous menu
```

LRAU.PRG

```
clear
set stat off
close all
use ORDR inde o_d_id
report form LRAU for flag='M'
close all
set stat on
return && previous menu
```

```

                                                                    LRG.PRG
set      procedure      to LBAR      && general reports
optn = 3      && * SPECIFICATION INCOMPLETE *
os = 0
      do while os # optn
          opt1 = "          "
          opt2 = "          "
          opt3 = "Previous Menu"
          optn = 3
titl = "LDBMS * REPORTS * GENERAL"      LRG"
      clear
      do LBAR with optn,titl
          do case      && identify choice..
              case os = 1
                  do NYET      && not yet installed!
              case os = 2
                  do NYET      && not yet installed!
          endcase.
      enddo
os = 0
return      && previous menu

```

```

                                                                    LRR.PRG
set      procedure      to LBAR      && rental related reports menu
optn = 8
os = 0
      do while os # optn
          opt1 = "books Issued"
          opt2 = "books Returned"
          opt3 = "books still on Loan"
          opt4 = "o_due rem (Students)"
          opt5 = "o_due rem (Teachers)"
          opt6 = "reserve collection"
          opt7 = "final Fine report"
          opt8 = "Previous Menu"
          optn = 8
titl = "L-DBMS REPORTS * RENTAL MANAGEMENT"      LRR"
      clear
      do LBAR with optn,titl
          do case      && identify choice..
              case os = 1
                  do LRR1      && books issued
              case os = 2
                  do LRR2      && books returned
              case os = 3
                  do LRR3      && books still on loan
              case os = 4
                  do LRR4      && ov'due rem - students
              case os = 5
                  do LRR5      && ov'due rem - teachers
              case os = 6
                  do LRR6      && reserved book coll.req.
              case os = 7
                  do LRR7      && final fine report
          endcase
      enddo
os = 0
return      && previous menu

```



```

                                                                    LU.PRG
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
    do case
        && identify choice..
        case os = 1
            do LUR
                && rental related < menu >
        case os = 2
            do LUT
                && transfer to BOOK.DBF
        case os = 3
            do LUD
                && update department info.
        case os = 4
            do LUO
                && update orders info.
        case os = 5
            do LUP
                && update publisher info.
        case os = 6
            do LUS
                && update supplier info.
        case os = 7
            do LUU
                && update user info.
    endcase
enddo
os = 0
return
&& previous menu

```

```

                                                                    LUD.PRG
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 UPDATES -> DEPT.DBF"
clear
do LBAR with optn,titl
    do case
        && identify choice..
        case os = 1
            do LUDA
                && add department
        case os = 2
            do LUDC
                && change department
        case os = 3
            do LUDD
                && delete department
    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
```

LUID.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
```

LUI.PRG

```
clear && record issue of a book
set status off
@ 01,00 to 03,78
@ 02,25 say "LUI ___ 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" LUO"
clear
do LBAR with optn,titl
do case && identify choice..
```

```

        case os = 1
            do LUOA      && add orders
        case os = 2
            do LUOC      && change orders info.
        case os = 3
            do LUOD      && delete orders
        endcase
    enddo
os = 0
return      && previous menu

```

LUOA.PRG

```

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      "accn 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_titl= space (64)
    m_auth= space (32)
    m_city= space (32)
    m_pnam= space (32)
    m_year= space ( 4)
    m_cost= 0.00
    m_copy= 0
    m_call= space (16)
    m_accn= space ( 8)
    m_subj= space ( 4)
    m_sub2= space ( 4)
    m_dord= ctod ('00/00/00')
    m_s_id= space ( 2)
    m_p_id= space ( 2)
    m_d_id= space ( 2)
    m_remk= 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

```

```

@ 03,15 get      m_titl
      read
@ 03,15 say      m_titl
@ 04,35 get      m_auth
      read
@ 04,35 say      m_auth
@ 05,35 get      m_rity
      read
@ 05,35 say      m_rity
@ 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_remk
      read
@ 18,35 say      m_remk
      DO while .not. upper(m_flag) # '@ABCDEFGHIJKLMNO'
@ 19,35 get      m_flag
      read
      ENDDO
@ 19,35 say      m_flag
      DO while .not. upper(m_stat) # '@ABCDEFGHIJKLMNO'
@ 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      BRDR
            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  accn with     m_accn
            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

```

```

                                                                    LUP.PRG
set      procedure to LPAR      && publisher file maintenance
        optn = 4
        os = 0
        do while os # optn
            opt1 = "Add      publisher      "
            opt2 = "Change publisher inf"
            opt3 = "Delete publisher      "
            opt4 = "Previous Menu      "
            optn = 4
titl = "L-DBMS UPDATE -> PUBL.DBF      LUP"
            clear
            do LPAR with optn,titl
                do case
                    && identify choice..
                    case os = 1
                        do LUFA      && add publisher
                    case os = 2
                        do LUFC      && change publisher info.
                    case os = 3
                        do LUPD      && delte publisher
                endcase
            enddo
        os = 0 : return      && previous menu

```

```

set procedure to LBAR && rental service menu
optn = 4
os = 0
do while os # optn
  opt1 = "Issue book      "
  opt2 = "Record returned book"
  opt3 = "Reserve a book   "
  opt4 = "Previous Menu   "
  optn = 4
titl = "L-DBMS  UPDATES -> RENTAL MANAGEMENT      LUR"
clear
do LBAR with optn,titl
  do case && identify choice..
    case os = 1
      do LURI && record issue of book
    case os = 2
      do LURR && record return of book
    case os = 3
      do LURV && reserve a book
  endcase
enddo
os = 0
return && previous menu

```

```

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

```

```

                                                                    LUU.PRG
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-DEMS UPDATE -> USER.DBF      LUU"
clear
do LBAR with optn,titl
    do case                && identify choice..
        case os = 1
            do LUUA        && add user
        case os = 2
            do LUUC        && change user info.
        case os = 3
            do LUUD        && delete user
    endcase
enddo
os = 0
return                && previous menu

```

```

                                                                    BOOK_D.PRG
@ 04,00 clea                && displays book information
@ 04,00 to 15,78
@ 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 "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
@ 09,35 say '19'+year
@ 10,35 say call
@ 11,35 say accn
@ 12,35 say subj
@ 13,35 say sub2
@ 14,35 say d_id

```

DEFT_D.PRG

```

@ 04,00 clear                && displays department info.
@ 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.PRG

```

@ 04,00 clea                && displays orders information
@ 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 Taka" = "
@ 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.PRG

```

@ 04,00 clear                && displays publisher information
@ 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                                && displays supplier information
@ 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 clea
@ 04,00 to 15,78                                && displays user information
@ 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(dexp)
@ 12,35 say    str(blim,1,0)
@ 13,35 say    str(ncur,1,0)
@ 14,35 say    str(fdue,4,2)

```

```

proc      LBAR                                && implements light_bar menu
para     opno,titl                            && get number of options & title
set      status off                          && from the calling program
        clea                                  && clear screen
        row = 5                               && start from line# 5
@ 00,00  to 02,79                             && draw box for header
@ 01,17  say      titl                        && write header
@ 21,00  to 24,79                             && draw box for instruction
@ 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"
@ 04,25
        row = 5
        do while row-4 <= opno
            sub = str(row-4,1)
            opt&sub = iif (val(opt&sub)=0,str(row-4,1)+". "+opt&sub,opt&
@ row,25      say opt&sub
            row = row+1
        enddo
        opt = 1
        sub = "1"
        sel = 0
@ 05,25  get      opt1
        clear     gets
        os       = 0
        do while os = 0
            sel = 0
            do while sel = 0
                sel = inkey ()          && interpret key pressed
            enddo                          && if up/down arrow..
            if ( sel=24 ) .or. ( sel=5 )
@opt+4,25      say      opt&sub
                opt = iif ( sel=24, opt+1,opt-1 )
                opt = iif ( opt>opno, 1,opt )
                opt = iif ( opt<1, opno,opt )
                sub = str ( opt,1 )
@opt+4,25      get      opt&sub
                clear   gets
                loop
            endif
            if ( sel >= 49 ) .and. ( sel < 49+opno ) then
                os = sel-48          && key is numeric
            endif
            if ( sel=17 ) then
                os = opt          && ENTER pressed
            endif
        enddo
        && activate option
set      stat   on          && and
        return                                && return to calling program

```

