A DATABASE SYSTEM FOR PROMOTIONAL LITERATURE FOR PUBLISHERS

by

JAMAL MAKI ABDUL-JABBAR

A Master's Thesis
submitted in partial fulfilment of the requirements
for the award of Master of Philosophy
of the Loughborough University of Technology

July 1981

Supervisor and Director of Research:
Professor D J Evans, PhD, DSc
Head of Computer Studies Department

ABSTRACT

The aim of this thesis is to design a database system which could easily be used by a publishing company to store data concerning the products it publishes and to enable such data to be used in the regular processes of the production of lists of books and periodicals of certain promotional requirements. In our approach we have used a relational model which is based on the mathematical theory of relations. This has certain advantages over systems designed using tree or plex structures for as the database grows it will avoid causing upheaval with the logical representation of data and application programs and provides a basis for a high level retrieval language.

The query language is designed to answer quickly all enquiries to the database and is based on principles and techniques developed from menu construction.

The requirements of the promotional information produced by a typical publishing house are analysed and a model set up which tests the theories we have developed.

In addition, the security aspect of the database has been studied and checks incorporated into the systems to ensure the authority of the personnel using the system and to provide a permanent record of all legal and illegal entries for management information.

ACKNOWLEDGEMENTS

I wish to express my gratitude to the Iraqi Government who offered me the financial support to do this research project.

I would like to express my sincere thanks and gratitude to Professor D J Evans, the Director of Research, my supervisor, for his invaluable help, advice and guidance throughout the course of my research.

I wish to express my special thanks to my parents for their sacrifice and encouragement and to my wife for her patience and encouragement.

Finally, I would like to extend my thanks to Mrs J Smith for the typing of this thesis.

CONTENTS

CHAPTER ONE

INTRODUCTION

1.1 Overview

The amount of information stored in computer systems increases daily. Many organisations, realizing the value of their data, have switched from the collection of redundant (and possibly inconsistent) data storage to a non-redundant, integrated database or a database management system.

Martin (1976) refers to the rapid growth of Computer Technology which influences most of the installations to change their previous system to a new one, but at a cost of expense and efficiency. Martin (1977) says *"A database may be defined as a collection of inter-related data stored together without harmful or unnecessary redundancy to serve multiple applications; the data is stored so that it is independent of programs which use the data; a common and controlled approach is used in adding new data and in modifying and retrieving existing data within a database. The data is structured so as to provide a foundation for future application development"*.

The use of an integrated database management system (DBMS) provides the management and the enterprise with centralised control of its operational data which is its most valuable asset (Date, 1977).

The database era was initiated in the late 1960's and the evolution of the systems before that time was as follows:

1. The programs depend on the used data.

2. There was a high level of redundancy between data files.
3. The data security was insufficient.
4. The data structure was designed to serve one application only.

However database techniques overcame many of these problems. There is a growing recognition of the need to provide and maintain three levels of data description (Ansi, 1975):

a) The subschema, which represents the users or application programmers view.

b) The physical layout of data in storage, including indexes and linkages.

c) The schema, which represents the overall view or the overall logical structure of the database (Kent, 1976).

As the database grew, the overall logical structure of the data became complex in many cases and inevitably changed. It became important that a plan for database design should be developed in such a way that any changes can be made to it without having to modify the application programs.

Data security is one of the essential aspects of DBMS which refers to the protection of data against accidental or intentional disclosure to unauthorized persons. The authorization process in a protection system is the process which translates and stores specifications of all protection requirements (Hartson, 1976). The system must be surrounded by layers of external controls.

The most important security layer is the administrative control which ensures that the system is used correctly.

Also the protection against fire is so important and if it does occur for the damage to be minimized.

Protection against machine failure is also necessary, so dumping of the files periodically is recommended and its transaction should be recorded on tape as a back-up store.

The increased use of interactive database systems is perceived and the users interact with such systems through the computer terminals. The successful man-machine communications establishment is an essential factor in the future growth of the computer industry and the acceptance of computer methods.

Little attention was paid to effective man-machine dialogue during its first two decades (Martin, 1973).

The dialogue design becomes fundamental to the users which did not have the professional expertise to communicate with the computer and were often unable to develop this expertise (Eason, 1976; 1979). The design requirements are (a) ease of use, (b) ease of learning, and (c) ease of modifications (Hebditch, 1979).

The good design of such a dialogue is extremely important because it has a major effect on the success or failure of the total system.

The main dialogue styles in current use are as follows:-

1. Natural language based.
2. Question answering technique.
3. Menu selection.
4. Programming-like statements.
5. Simple instruction to the operator.
6. Query-by-example.
7. Displayed formats.
8. Panel modification techniques.
9. Form filling.

The requirement for a fast response constrains the design of both the interface sub-system and the database sub-system.

The database could be designed for on-line, batch-processing, real-time or serve many processing types.

The advantages of using a database can be listed as follows:

1. The logical and physical data structure are independent.
2. It allows the data structure to be changed easily.
3. It is independent of programs which use the data.
4. It can serve many applications.
5. There is more security on the data.
6. It removes the redundancy of data in the files.
7. It allows for constant growth to the systems.
8. It is more efficient.

The decision to use a DBMS for a company's information needs is as critical as the decision to introduce computers in the first

place (Tsichrizis, 1977).

The conversion to DBMS is costly because it represents a large commitment in terms of money and human resources.

1.2 Data Structure and Data Description

The association between the various items of data that are stored can cause complexities in data organization.

The essential elemental piece of data is the data item (field, data element). A data item cannot be subdivided into smaller data types and retain any meaning to the users of the data. A data item by itself is not much use. It becomes useful only when it is associated with other data items.

The data which is actually stored in the computer is called physical data and that which the applications programmers refer to, is called logical data. The users of the database (applications programmers or terminal users) see the relationships in different representations and according to their requirements.

There are three separate views of the data:

1. The physical data structure which represents the physical layout of the data in the storage and the organization of the files and indexes. It can be seen by systems programmers and systems designers.

2. The conceptual schema: which represents the overall view of the data as seen by the database administrator or by systems analysts.

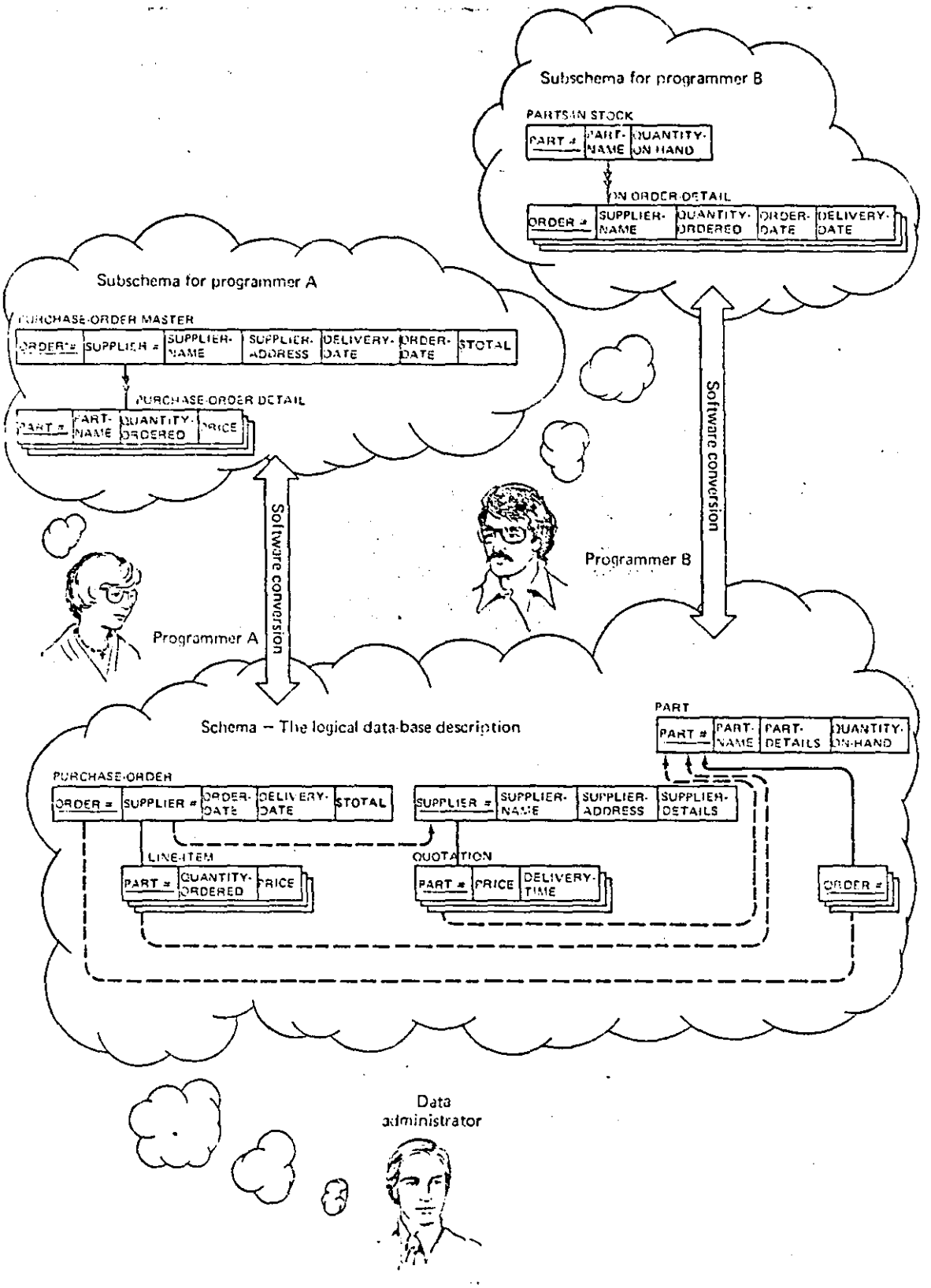


FIGURE 1.1 This figure shows the schema and two subschemas derived from it by different programmers (Martin, 1976).

The schema contains descriptors of the conceptual records and fields that constitute an information model of an enterprise.

3. The subschema: which is derived from the schema and contains descriptors of the data as seen and manipulated by users and application programs. This is shown in Figure 1.1.

The database management software must have the ability to separate the physical organization of the data from the logical organization or user's of the data.

The user's view of the data should be in whatever form is most convenient for him and his associates, and the data management software should carry out the translation between this logical organization and whatever physical organization gives efficient performance.

A mapping can be as simple as a one-to-one name association or may be quite complex for a one-to-many association.

The degree of complexity is limited, first by the flexibility of the database management system, and then by economic constraints of the use of the data. The greater the constraint on response time, the simpler the mapping has to be. In addition the less freedom different users have in viewing the same data different ways, and reduces the capability of the database administrator in returning the database without expensive conversions. There are several ways of drawing the association between two data items as follows:

1. 1:1 mapping: There is one-to-one mapping from data item A

to data item B if at every instance in time each value of A has one and only one value of B associated with it. This relationship is represented by a simple arrow on a line which connects the ellipse A to B as shown in Figure 1.2.

Ex: The relationship between an employee's personal number and social insurance number. Each employee has only one unique personal number and one unique social insurance number.

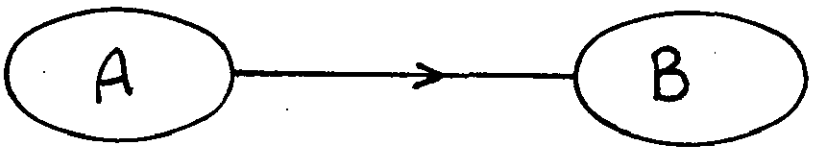


FIGURE 1.2

2. 1:N: There is one-to-many mapping from data item A to data item B, this means that one value of A has zero, one, or many values of B associated with it. This is shown in Figure 1.3.

   Ex: The relationship between an employee's personal number and salary history is, in general, one to many. An employee

has only one unique personal number, but may have had several different salaries.

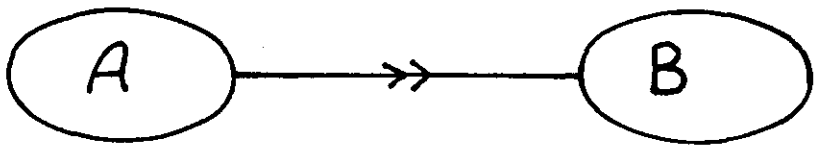


FIGURE 1.3

3. N:M: There are many-to-many mapping from A to B and from B to A. This is shown in Figure 1.4.
   Ex: Many-to-many relationhip is that between house colour and house price. That is, houses with a certain colour may sell at various prices and, similarly, houses at the same price may have various colours.

FIGURE 1.4.

A large number of associations exist for a large number of data items, so to minimize the number of associations, data items must be gathered into related groups called a record, segment, or tuple. Each record is identified by a primary key which is a node with one or more single arrows leaving it. The primary key is called the concatenated key, if it consists of more than one data item.

The schema representation comes from drawing its records and the relationship between them. It means the overall chart of the data item types and record types stored in a database.

1.3 Database Objectives

Many enterprises have been studied at great length, the database principles of which should guide us in selecting integrated organization techniques. Many reports have been issued about this subject (Martin, 1977).

There are many important objectives for database organization, which are listed as follows:

1. Future applications development should depend on database principles and should make application development easier, cheaper and more flexible.
2. The database systems should permit end users to employ data by using powerful languages whichhelp naive users to query, search and to manipulate the data.

3. It is very important to organize a database which can handle a spontaneous request.
4. The data model must be designed in such a way for it to accept new applications requirement from existing data.
5. The database growth must not lead to changes in the application programs which use it.
6. The detection of legal and illegal access must be included in the system design to protect the data from unauthorized users.
7. The data must be protected against damage and failure.
8. More than one user can use the system and perceive the data differently.
9. The data should be clear and easy to understand by the users.
10. Data redundancy should be eliminated as much as possible.

1.4 Three Approaches to Database Systems

These approaches are discussed below:

1.4.1 The Hierarchical Approach

The data is represented by a simple tree structure, which starts from the root then branches out with every nodes generating new nodes at higher levels. The tree is drawn upside down with the root at the top.

A tree can have up to 15 levels, the highest level is the root, it may have any number of dependents, each of these may have any

number of lower-level dependents, and so on, to any number of levels. The model represents the simplest relationship which every child has to its parents.

This approach is used in many existing database systems including IMS (Information Management System). DL/1 (Data Language/one) is used to specify the logical representation of database (schema).

The database consists of a collection of trees of segments. Pointers are used to avoid duplicating the same segment in different trees. This is shown in Figure 1.5.

The logical structure of data which is perceived by an application programmer is also a tree of segments.

The problem of asymmetry in the information retrieval of the hierarchy model leads to unnecessary complications for the terminal user. This causes upheaval in the structure and the hierarchy becomes more complex.

1.4.2 The Plex Approach (Network)

The plex approach or network, is organized by CODASYL between 1967 and 1971. Many commercial implementations have been done to these approaches, the best known of which are the UNIVAC's DMS 1100 and Siemen's UDS. The data description language (DDL) is used to define schema and sub-schema.

A set which is, a named two-level tree, is considered to be the basic construct of the language (Olle, 1978).

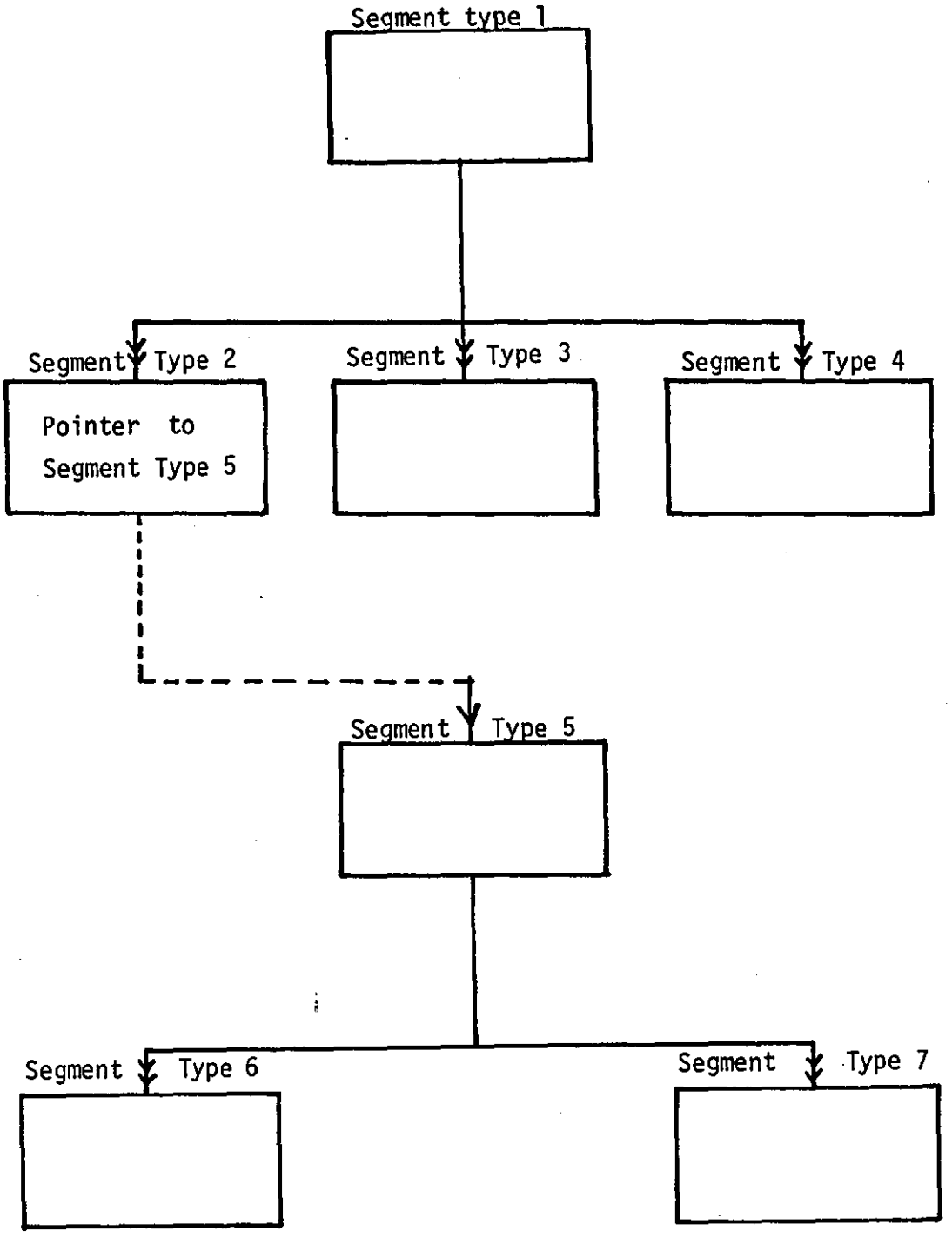


FIGURE 1.5 The hierarchical approach (Mohamad Salih, 1979)

The network data model is a formal model for representing the attribute relationship of an entity set and its associations between the entity sets. The model consists of record types and links. Links are used to represent the associations between the entity set. Each set can represent the associations between two or more record types. To represent a multilevel tree, more than one set is necessary. A record type which declares an owner at a lower level in the tree is also declared as a member of higher-level sets. This is shown in Figure 1.6.

The network model is more symmetric than the hierarchical model but in connection with queries one significant problem arises which refers to the retrieval procedures which are more complicated than in the hierarchical. Also, the internal structure of the file is more complex than in the hierarchical case.

1.4.3 The Relational Approach

The relational approach to database management systems differs from the previous approaches in that it provides a means of describing data using a two-dimensional table which is often the natural (overall) structure for the data.

The mathematical theory of relations is the basis for the relational approach. The results of relational mathematics can be applied directly to the relational database.

In mathematics, the set description is a collection of objects thought of as a whole as stated by Arthurs (Arthurs, 1965).

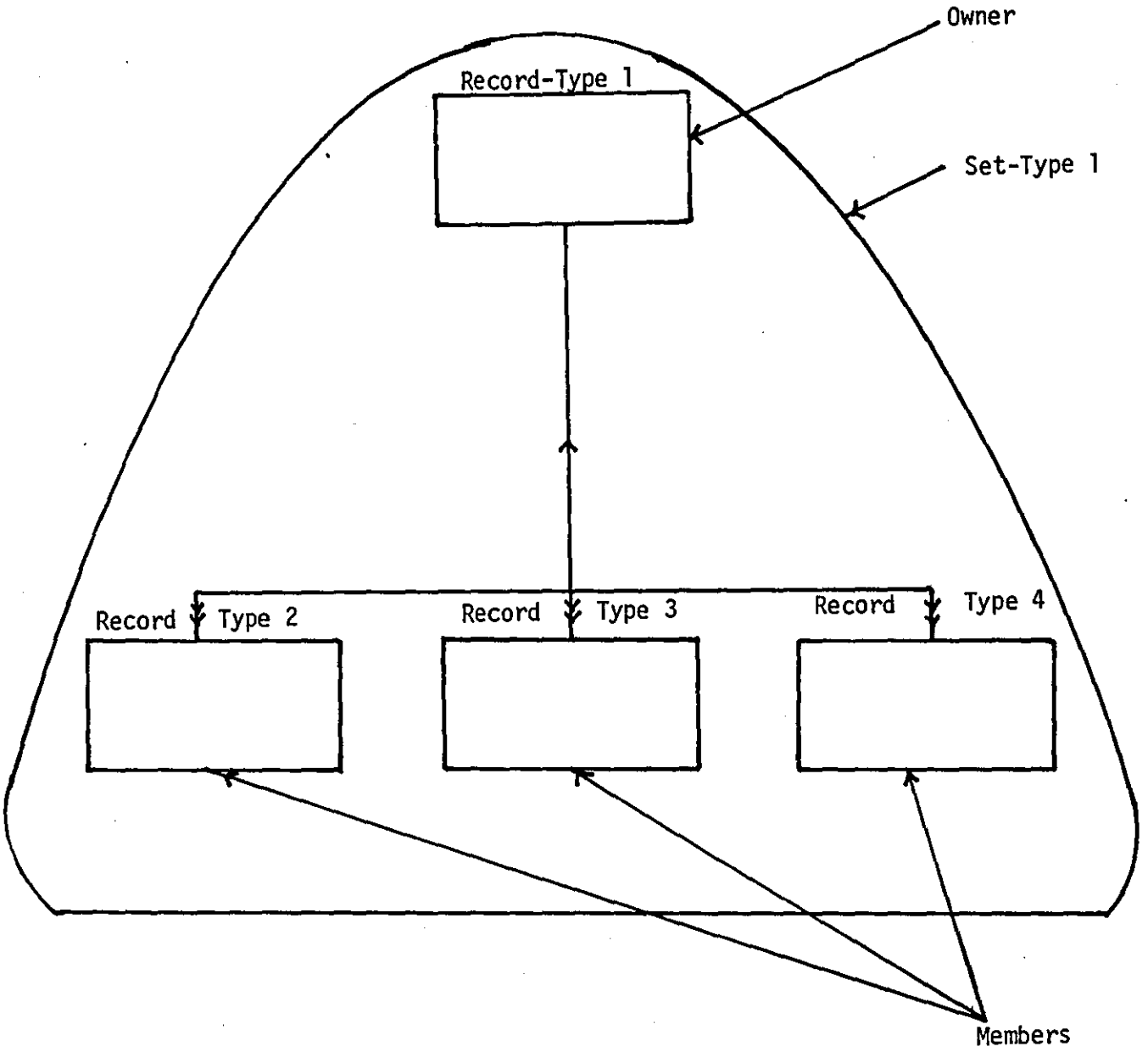


FIGURE 1.6 A set-type can have only one owner record-type but multiple member record-types (Martin, 1975).

The objects, of which the set is a collection, are called elements or members of the set.

If A is a set, and X is an element of A, we write

$$X \in A$$

If X does not belong to A we write

$$X \notin A$$

We may be able to specify a set by writing down names of all its elements. For instance, if $A_1, \ldots, A_n$ are objects, the set consisting of precisely these objects will be written as:

$$[A_1, A_2, \ldots, A_n]$$

The term relation was defined as follows:
Given sets $S_1, S_2, S_3, \ldots S_n$ (not necessarily distinct), R is a relation on these n sets if it is a set of n-tuple $S_1, S_2, S_3, \ldots S_n$ such that $S_1 \in S_1, S_2 \in S_2, \ldots, S_n \in S_n$. We shall refer to S_J as the J^{th} domain of R. R is said to have a degree n.

R $(S_1, S_2, S_3, \ldots, S_n)$ is used as a notation to represent the relation R.

The dimensional table referred to as relation. The rows of the table are referred to as tuples. A tuple is thus a set of data-item values relating to one entity. A tuple containing N

values is called an N-tuple. A column in the table is referred to as domain. If there are N-domains, it is of degree N. Relations of degree 2 are called binary, and degree N are called N-ary.

The table has the following other properties:

a) The ordering of the rows is immaterial.
b) No two rows are identical.
c) The data-item should be atomic (non-decomposible) in each tuple within the table.

It is possible to avoid the entanglements that build up in tree and plex structures, by a technique called normalization. Normalization is a step-by-step process for replacing relationships between data with relationships in two-dimensional tabular form. The table must be set up in such a way that no information about the relationships between the data is lost.

The objectives of normalization are:

a) To make it feasible to tabulate any relation in a database.
b) To free relations from undesirable insertion, update and deletion dependencies.
c) To obtain a powerful retrieval capability by means of simpler collections of relational operations to manipulate the relations.
d) To reduce the need for restructuring the relations as new types of data are introduced when further applications or user views require them.

e) To make the relational model more informative to the user.

f) To make the collection of relations neutral to the query statistics, where these statistics are liable to change by time.

Codd (1972) defined three levels of normalization, which he called the first, second, and third normal form respectively.

He proposed a set of normalization procedures which involved the following steps:

1. The separation of repeated groups into separate records.
2. The separation of attributes not dependent on all fields of the primary key.
3. The separation of attributes which depend on other attributes.

There are four normal forms as follows:

a) First normal form (1NF):

A relation is in 1NF if and only if all the underlying domains contain atomic (non-decomposible) values only.

b) Second normal form (2NF):

A normalized relation R, is said to be in 2 NF if and only if the non-key attribute is fully dependent on the primary key.

c) Third normal form (3NF):

A relation R is said to be in third normal form if it is in

2NF and every non-key attribute is non-transitively dependent on the primary key (Date, 1977).

The 2NF can be converted to 3NF by removing any transitive dependency.

For example, in the relation:

EMPLOYEE (EMPLOYEE#, EMPLOYEE NAME, SALARY, PROJECT#,
COMPLETION DATE)

The COMPLETION DATE is functionally dependent on PROJECT# and PROJECT# is functionally dependent on EMPLOYEE#.

Therefore, COMPLETION DATE is transitively dependent on EMPLOYEE#. So that the relation should be split into two relations in order to convert it to third normal form:

EMPLOYEE (EMPLOYEE#, EMPLOYEE NAME, SALARY, PROJECT#)
PROJECT (PROJECT#, COMPLETION DATE)

d) Fourth normal form (4NF):

Date (1977) defined the 4NF as *"A relation R is said to be in 4NF if and only if, for all time, each tuple of R consists of a primary key value that identifies some entity, together with a set of mutually dependent attribute values that describe that entity in some way"*.

Sometimes a relation R is said to be in 3NF, but in reality it consists of a multi-valued dependency, so that such a relation

needs a projection process to overcome this problem.

Normally, one column of a given relation has values which uniquely identifies each element (n-tuple) of that relation. This is called the primary key. A primary key is non-redundant if its components are in the form of either a simple domain or a combination of simple domains.

The relational model of a database may be defined as a collection of normalized relations. These relations are time varying in that they are subjected to tuple insertion, deletion and modification.

A relational database, based on relational calculus or relational algebra is easy to use for non-programmers, especially for mathematicians.

Also relational algebra and calculus form powerful data sublanguages or languages in which the operator is capable of manipulating entire sets as single objects (Date, 1977).

One such language is SEQUEL which can be used to express a wide range of retrieval operations (MacLeod, 1979).

CHAPTER TWO

PREVIOUS WORK

2.1 Definitions

2.1.1 Bibliography

A list of references (books, journals, etc) on a particular subject which contains the author(s) name, title, ISBN, publishing year, etc. for each reference.

2.1.2 Catalogue

A catalogue is a list of literature(s), in a prescribed order, containing bibliographic information (author, title, subject, etc) for each literature [Noerr, 1976].

2.1.3 MARC

MARC is a machine readable catalogue. Its format was introduced by the Library of Congress in March 1969 and designed for communicating the records of books, reports, periodicals etc. [Kimber, 1974].

The MARC record structure consists of three main sections as follows:

1. A fixed length descriptive record which contains a subfield of the overall length of the record, and other information.
2. The directory contains a series of fixed length fields, one for each data field.
3. The variable length section represents the data fields. A selected delimiter character is used to end each data field.

2.1.4 MERLIN

It is a large-scale machine-readable shared bibliographic database with facilities to amend, add and search for records and to aid in the production of catalogues, bibliographies, listings etc. [Ross, 1976].

2.2 Introduction

Publishing houses almost invariably produce catalogues and lists concerning their literature(s) which include bibliographical information for the users. The most beneficial lists are:

1. A list in alphabetical order of author/editor as shown in Figure (2.1).
2. A list of titles in alphabetical order of titles as shown in Figure (2.2).
3. A list of prices in a specific order as shown in Figure (2.3).

These lists are for the user interest, because they are of assistance to select the required literature easily.

Most of the libraries offer a range of such services for the selector, whether he is a branch librarian, a professor in the university, an engineer in an industrial laboratory, or a student [Heiliger and Henderson, 1971].

Catalogues are necessary in large libraries owing to the fallibility of most human memories and to the practical impossibility

BOOKS

ADLER Organic Solid State Chemistry

524pp 1969 0 677 13200 X

ALEXEYEV Quantitative Analysis (RM)

494pp 1969 0 677 20860 X

ASINGER/OVERBEEK/PAQUOT Chemistry, Physics and Application of Surface Active Substances

Volume 1: Chemistry of Surface Active Substances

610pp 1967 0 677 11800 7

Volume 2: Physics and Physical Chemistry of Surface Active Substances

1390pp 1967 0 677 11810 4

Volume 3: Application of Surface Active Substances

1004pp 1967 0 677 11820 1

3-Vol. set 0 677 10510 X

BAHN Reaction Rate Compilation for H-O-N System

254pp 1968 0 677 12750 2

○ BLOKLAND Elasticity and Structure of Polyurethane Networks

120pp 1969 0 677 61200 1 (cloth)
0 677 61205 2 (paper)

Ⓜ BROWN/DIENES/LABES Liquid Crystals 1

494pp 1967 0 677 11840 6

BROWN Liquid Crystals 2 In 2 Parts

Part 1
252pp 1969 0 677 13830 X
Part 2
910pp 1969 0 677 13840 7

BROWN/LABES Liquid Crystals 3 In 2 Parts

Part 1
596pp 1972 0 677 12620 4
Part 2
602pp 1972 0 677 12630 1

2-Part set 0 677 15010 5

BROWN/DIENES/LABES Proceedings of the Sixth International Liquid Crystal Conference In 4 Parts

Edited by Glen H. Brown, *Kent State University*, G.J. Dienes, *Brookhaven National Laboratory, and* M.M. Labes, *Temple University*

This important international conference took place at Kent State University in August 1976. The papers were published in the Gordon and Breach journal *Molecular Crystals and Liquid Crystals* but are available separately as the complete proceedings in four parts.

Approx. 1200pp 1977

BRUINS Basic Principles of Thermoforming

294pp 1973 0 677 14990 5

FIGURE 2.1 Books list in alphabetical order of author/editor

FIGURE 2.2 Title Index

BOOKS

ADLER			
0 677 13200 X	1969	$58.00	£30.70
ALEXEYEV			
0 677 20860 X	1969	$29.50	£19.70
ASINGER/OVERBEEK/PAQUOT			
Vol. 1			
0 677 11800 7	1967	$80.00	£53.30
Vol. 2			
0 677 11810 4	1967	$60.00	£40.00
Vol. 3			
0 677 11820 1	1967	$120.00	£80.00
3-Vol. set			
0 677 10510 X		$230.00	£153.30
BAHN			
0 677 12750 2	1968	$39.00	£26.00
BLOKLAND			
0 677 61200 1 cl	1969	$14.50	
0 677 61205 2 pa		$10.50	
BROWN/DIENES/LABES (1)			
0 677 11840 6	1967	$59.00	£39.30
BROWN (2)			
Part 1			
0 677 13830 X	1969	$29.50	£19.70
Part 2			
0 677 13840 7	1969	$109.00	£72.70
BROWN/LABES (3)			
Part 1			
0 677 12620 4	1972	$72.00	£48.00
Part 2			
0 677 12630 1	1972	$72.00	£48.00
2-Part set			
0 677 15010 5		$120.00	£80.00
BROWN/DIENES/LABES			
In prep.	1977		
BRUINS			
0 677 14990 5	1973	$29.50	£19.70
BRUINS			
0 677 14980 8	1971	$32.50	£21.70
BRUINS			
0 677 12200 4	1974	$22.50	£15.00
BRUINS			
0 677 11160 0	1976	$36.00	£24.00
BUDNIKOV/GINSTLING			
0 677 61250 8	1970	$59.00	£39.30
CLAWSON/LANDSBERG			
0 677 02710 9	1972	$24.50	£16.30
CNRS			
0 677 10060 4	1965	$59.00	£39.30
CNRS			
0 677 30730 6		$52.75	£35.20
COPELAND			
0 677 02830 X	1974	$14.50	£ 9.70
COUSSEMANT/HELLIN/TORCK (Fr.)			
0 677 50120 X cl	1969	$29.00	£19.30
0 677 50125 0 pa			£ 7.50
CSIRO			
0 677 65210 0	1975	$30.50	
DAVYDOV			
0 677 20610 0	1966	$42.00	£28.00
DeVRIES/KOCHVA			
Vol. 1			
0 677 12430 9	1971	$50.00	£33.30
Vol. 2			
0 677 12440 6	1972	$34.00	£27.70
Vol. 3			
0 677 12450 3	1973	$29.00	£19.30
3-Vol. set			
0 677 14710 4		$99.00	£66.00
DUBOIS			
0 677 50730 5 cl	1972	$45.00	£30.00
0 677 50735 6 pa		$22.50	£15.00
DUPLAN/CHAPIRO			
Part 1			
Vol. 1			
0 677 30640 7	1973	$36.00	£24.00
Vol. 2			
0 677 30650 4	1973	$31.00	£20.70
2-Vol. set (Part 1)			
0 677 15780 0		$59.00	£39.30
Part 2			
Vol. 1			
0 677 30880 9	1973	$49.00	£32.70
Vol. 2			
0 677 30890 6	1973	$54.00	£36.00
Vol. 3			
0 677 30900 7	1973	$54.00	£36.00

FIGURE 2.3 Price List

of acquainting the staff with the entire stock which they are called upon to serve. The British National Bibliography (BNB) is one of the range of services provided by the British Library, to the whole United Kingdom. The BNB is a weekly bibliography of all books published in the UK.

In 1961, the computer was first used in the processing of bibliographical information, when the chemical abstracts service (CAS) produced chemical titles (CT) [Bernard, 1977]. It was a machine-generated alphabetical subject index to the 600 most influential scientific journals.

Toni [1978] refers that since then there have been several significant developments which include a steady increase in the number of machine-readable bibliographic databases, especially in the library environment.

In spite of the increasing use of the machine-readable bibliographic databases, the printed products are considered a vital part for the publishing houses, because its distribution covers a large number of people. After the introduction of database management systems, the information retrieval community have avoided the type of DBMS applications which are used in business applications.

The design of most document retrieval systems have been around special-purpose file organizations and are often biased towards a particular type of database [MacLeod, 1979].

Finally, many existing systems concerning the production of catalogues and lists have been studied and are discussed in this chapter.

2.3 Previous Applications

2.3.1 The University of Sydney Library Catalogue System

The University of Sydney was founded in 1850 and its Library is considered the oldest and largest library in Australia.

The University has been engaged in automation activities since the 1960's [Jacob, 1975].

The first catalogue which was produced by the computer is the undergraduate book catalogue in 1960.

The library commenced automation in 1968 and the cataloguing activities concerning the design of cataloguing systems began in the early 1970's. The design of the cataloguing and the MARC systems were completed in late 1970.

The objectives of the system were to create and maintain a machine-readable record for their catalogue data, to produce catalogue cards for this material and to produce printed catalogues in book form. This was run on an IBM 360/20 machine and its programs written in Assembler code.

Data for each catalogue entry were recorded on a work sheet,

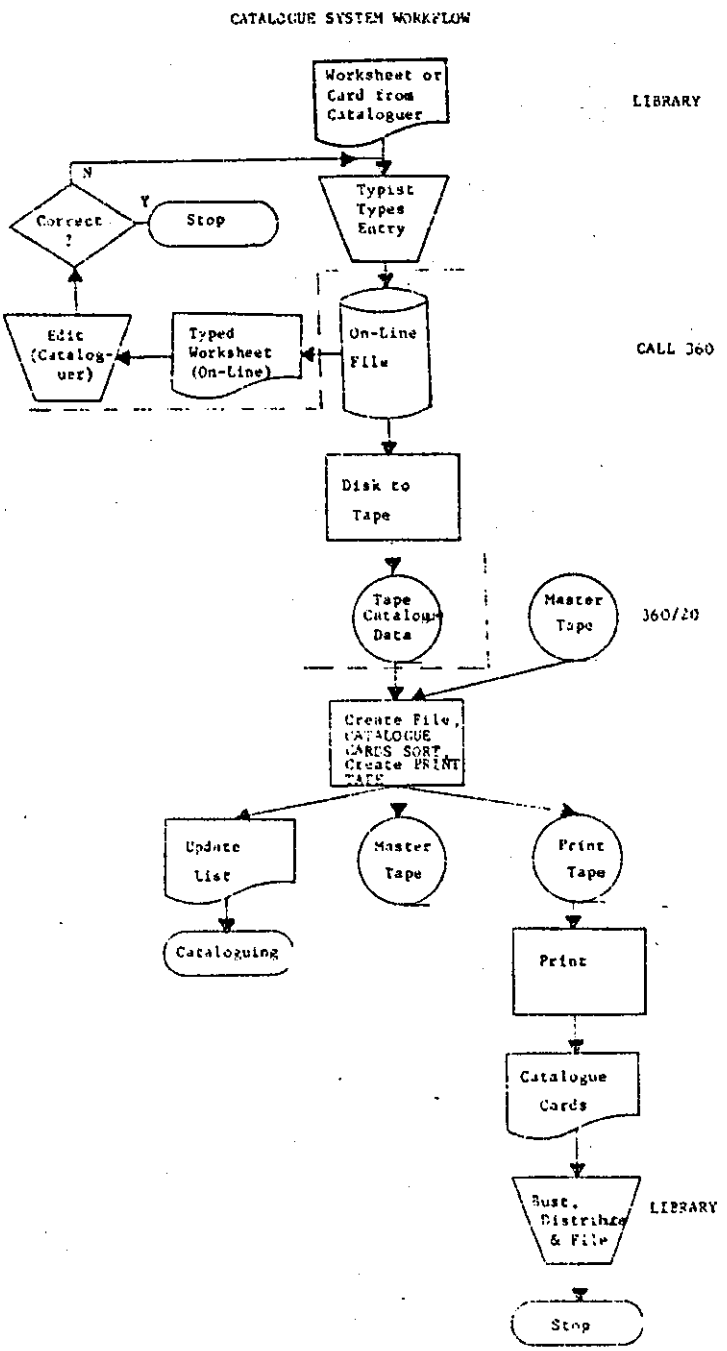


FIGURE 2.4 Catalogue system workflow (Jacob, 1975)

which was based on the MARC concepts of data bugging. Each entry was given a unique identification number, referred to as a 'Control Number'.

The pre-print form (worksheet) which was used for the entry aided the terminal operator to discover any incorrect entry. The data concerned the author(s), title, subject, series, notes, etc. Checking was done on the data immediately after entry. The system run flowchart is shown in Figure (2.4).

The disadvantages of the system are, the time taken by the production of cards catalogue and the easy corruption of the cards.

Also, the system does not allow for direct access to any specific data in the master file and it serves only one application.

2.3.2 Catalogues Production System at the Loughborough University of Technology (LUT) Library

In the mid-1970's, the Library Department at LUT started development of a computer system to produce catalogues concerning their library stock. In this system a MINICS format (minimal-input cataloguing system) is used to process and store the data for each item. The MINICS format prototype was developed in 1973, and the main aim of this format was to minimize the input cataloguing data [Wall, 1973].

The Library Department staff started forming files in 1976 and in 1978 the first microfiche catalogue was produced by the system.

The system is running on two computer machines which are the dual Prime 750 and ICL 1904 machines.

The system uses a prime terminal for data entry, as shown in Figure (2.5).

According to the MINICS format, every item entered starts with the item number field, then the rest of the fields. Each field starts with a special two character code then the field code and the data field. Each item terminatedby a special terminator (<<>>).

The Prime Editor is used for checking and correction of the data through the terminal directly. Then the data entered is transferred to the ICL 1900 machine on magnetic tape and this is followd by a specific program which runs to convert each item data to three records. The first record represents the leader of the item which consists of the whole record length in words (ICL word = 4 characters), the item number and other information.

The second record represents the directory which consists of, the field code, field length in characters and the position of the starting character of each item's field.

The third record represents the data field which consists of, a two character field code, the data field and terminator for each field. The last two being variable length records.

These three records represent the MINICS format in the storage.

M 19865201N

<<81 s

<<82 Publisher

<<88 800805

<<10 0115111816

<<23 Z

<<24 *0 623 865

<<28 *0 Great Britain Department ^of^ Trade

<<31 Survey of fire appliances: instructions for the guidance of surveyors

<<32 3rd ed

<<37 H.M.S.O.

<<38 1980

<<>>

M 19865801N

<<81 s

<<82 Publisher

<<88 800805

<<10 0117010227

<<23 Z

<<24 *0 614 5996

<<26 *0 Todd J E *1 Walker A M

<<28 *1 Great Britain Office ^of^ Population Censuses ^and^ Surveys Social Survey Division *1 University ^of^ Birmingham Dental School

<<30 Adult dental health

<<31 Vol 1 England and Wales, 1968-1978

<<33 by J E Todd and A M Walker

<<37 H.M.S.O.

<<38 1980

<<>>

FIGURE 2.5 Data entered through a Prime terminal

The update process is applied to the master file which already exists in the 1900 machine after a validation process to the entered data.

Finally the master file can be used to produce the catalogues. These processes are shown in Figures (2.6) and (2.7). The disadvantages of this system are it does not allow for direct access to any specific data and the time taken to process the data.

2.3.3 MERLIN

The MERLIN system started development in 1975 by the British Library whilst the British Library was running various computer based systems including the production of BNB, the distribution of MARC tapes and a catalogue service [Noerr, 1976: White, 1976: Robinson, 1980]. Many computers from different manufacturers were used to run these systems in many departments.

The aim of the MERLIN system was to cover all these services in a central database shown in Figure (2.8).

MERLIN was designed to provide commands for adding new data to the database and manipulating the existing data. Also, it offers a very flexible service to a wide range of users by the execution of a given process in several different ways through one program. This means that such a program can be used by a set of parameters.

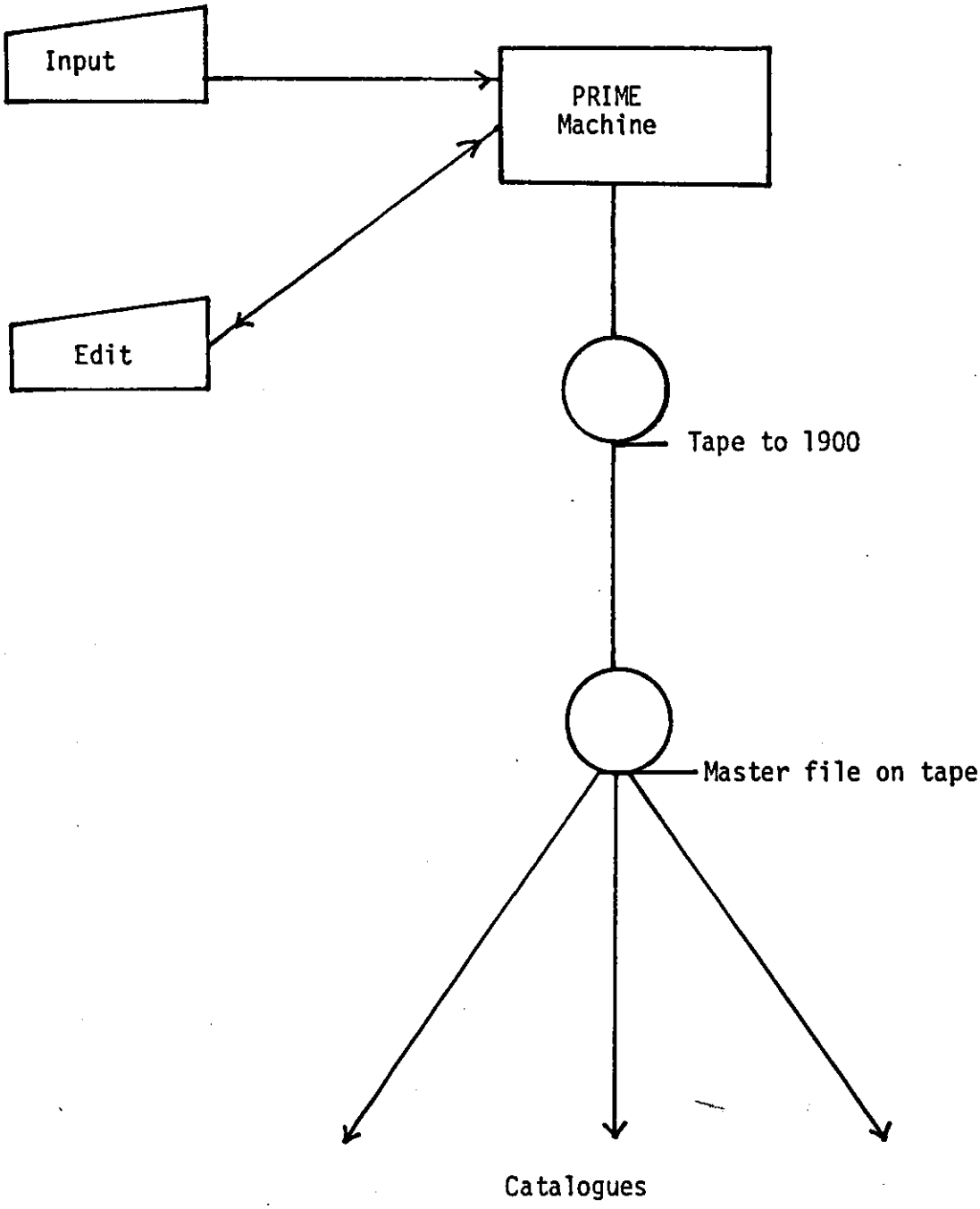


FIGURE 2.6 Outline system run chart

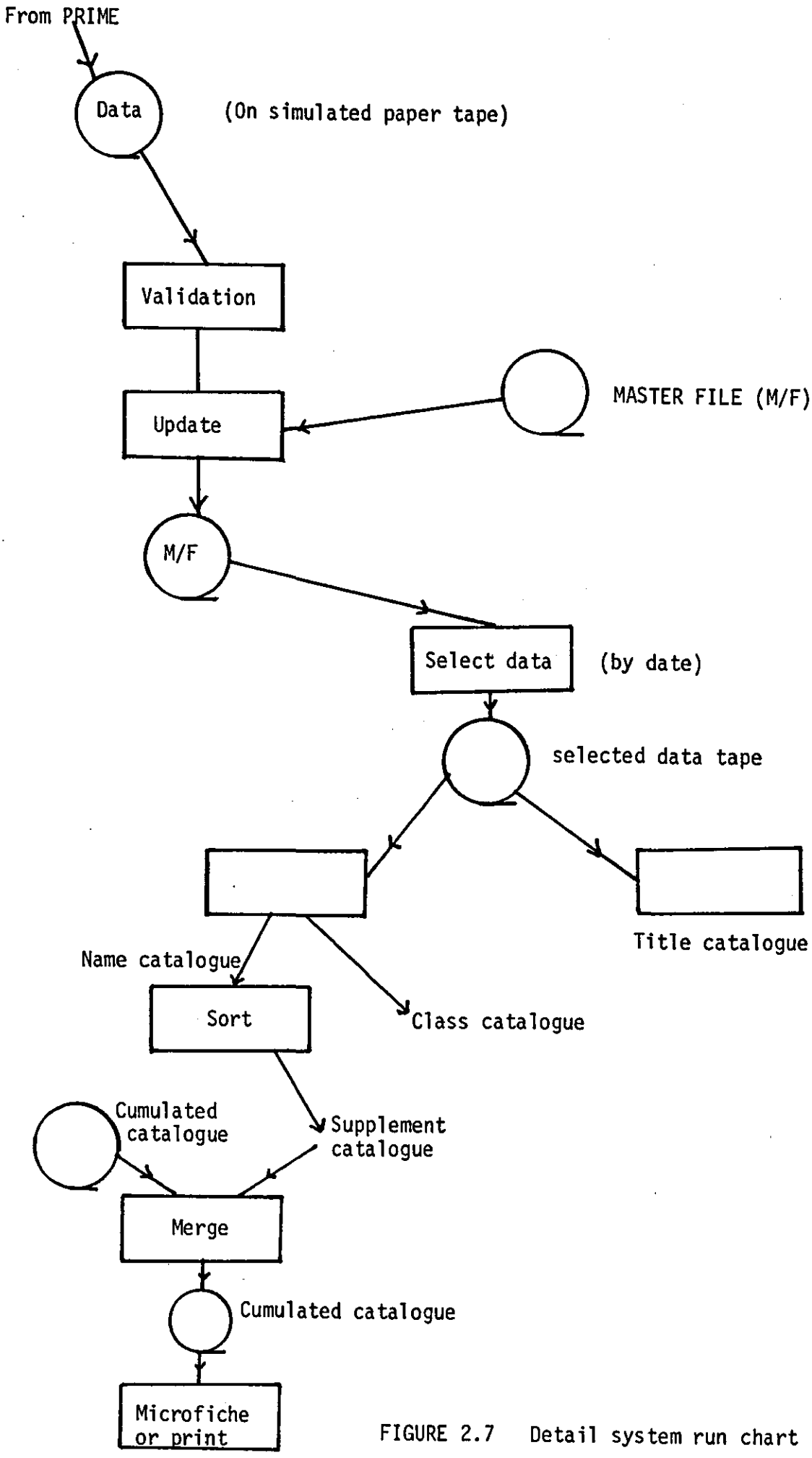


FIGURE 2.7 Detail system run chart

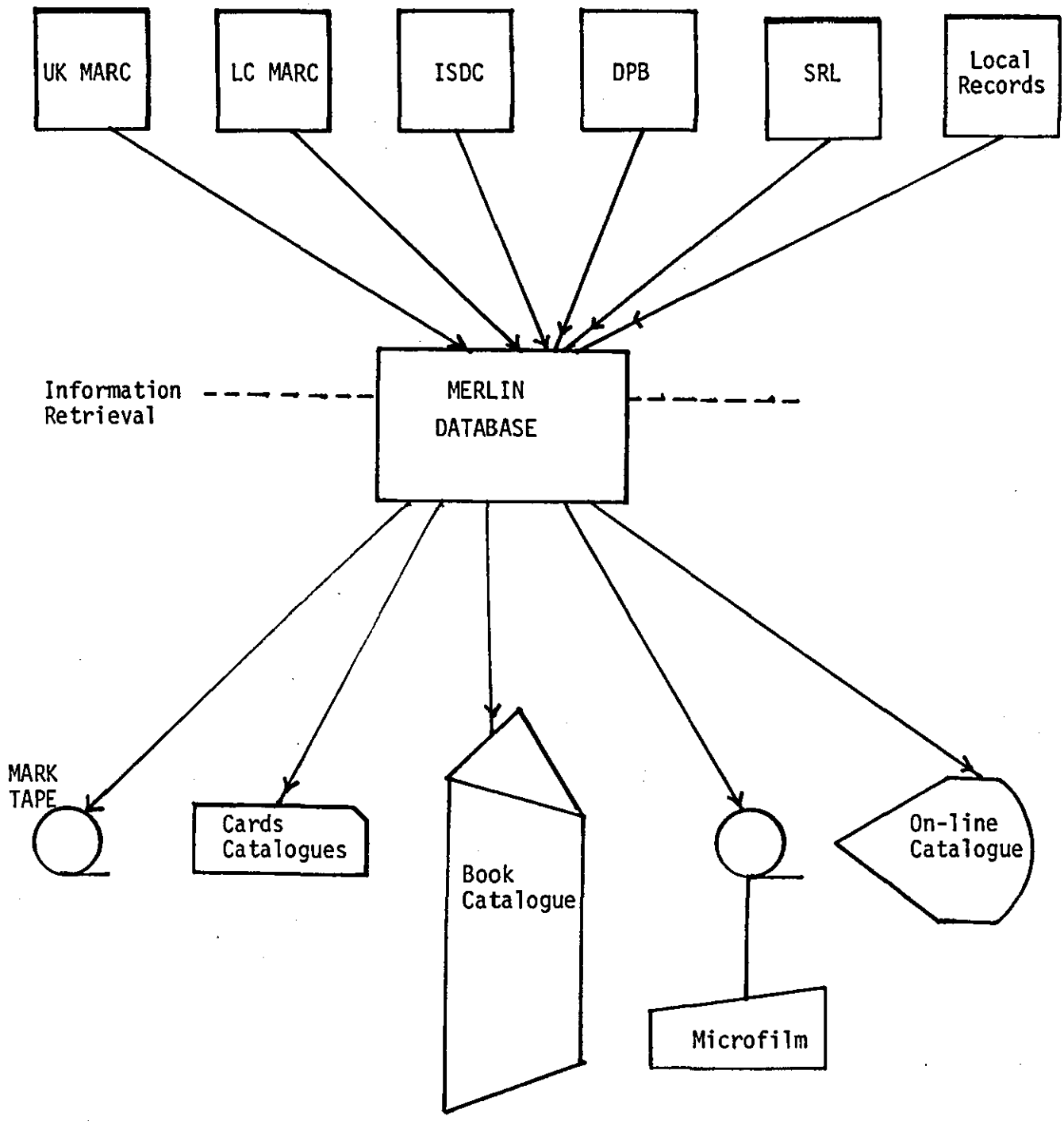


FIGURE 2.8 MERLIN database showing all services which can be offered

[John Ross, 1976]

MERLIN was divided into more than one phase. The first phase aimed to comprise the new system for producing BNB, MARC, tapes and other services [Ross, 1976].

The design of MERLIN is a modular hierarchical database system [Robinson, 1980].

The storage of each catalogue record in MERLIN format is not a string of characters as in MARC format. The data for each catalogue record is stored in many record types. Each record type is indexed by the computer and referred to by its address or location on the computer disk [Hopkinson, 1977].

Pointers are used in the records to indicate the association between them for each of the catalogue data.

The advantages of this storage format are, it saves space in the storage media and makes the retrieval process of any record type faster. The disadvantages of the MERLIN system are, the variability of the fields and the N:M relationships between the data in the data model, which could cause an inconvenient and awkward representation [Tsichrizis, 1976: Robinson, 1980].

2.3.4 A Typical Publishing House Database

A science publishers' company uses a card-based database to store data about the books it publishes and uses it in a regular process to produce "promotional literature, leaflets, etc" typically lists of books belonging to certain specialized categories as shown in

Figure (2.9). They prepare book 'blurbs' for each book which includes, author(s) name, book title, main category of the book, sub-category, description etc, before the arrival of the manuscript. However, finalized information on some of the aspects (for example, number of pages, publication date, price, etc) is established after an advance copy of the book is received from the binder. Blurbs for each book are kept on the cards. After the main 'blurb' is written, there are small additions and changes which come to light from time to time, and require the correction of the original blurb.

The cards are updated by hand on a regular basis, by removing the old card and writing and inserting new ones with the updated data. These cards are ready for inputting at any time, to the computer.

Regular promotional pieces are produced of the following types:

Book list 19XX

This is published annually and lists in alphabetical order the author/editor all titles in print, together with the minimum details such as pages, publication date, ISBN, and price.

New books 19XX

The list for new books is produced annually which includes those published books within a certain period of time (usually one year) and those forthcoming books where a publication date is reasonably certainly known.

Catalogue sections

These catalogues are generally produced, on a cyclic basis. Catalogue sections are centred around a particular subject, which includes books list, price list and title-author list.

Sectional title lists

It is a list in alphabetical order of the title around a specific subject, i.e. chemistry, gardening.

Database size

There are usually 1200 books in print at any one time. The amount of data held about each book 'blurb' varies from book to book because some books are in series or some of the 'blurb' includes the contents or description or both as shown in Figure (2.10).

The disadvantages of the system are as follows:

a) Unless great care is taken with the input cards, the database could be easily corrupted, for example, by dropping some, or all, of the cards and replacing them in the wrong order.

b) The processing time for producing lists will always be high, because of the need to sort the cards according to the required criteria, although there may have been very few additions to the database.

c) There is no security in the system to protect the database against illegal access.

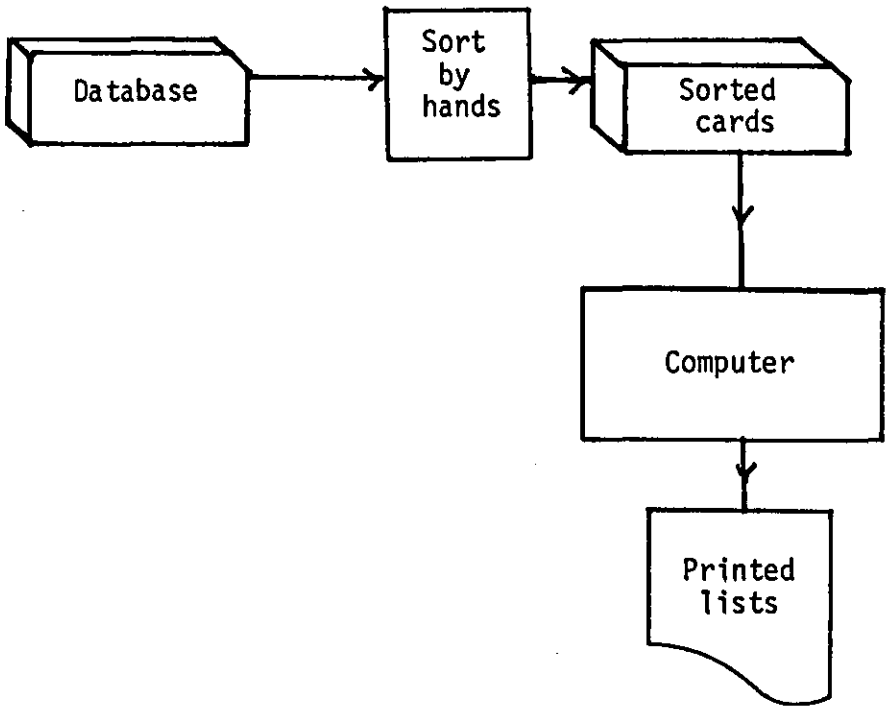


FIGURE 2.9 System run chart

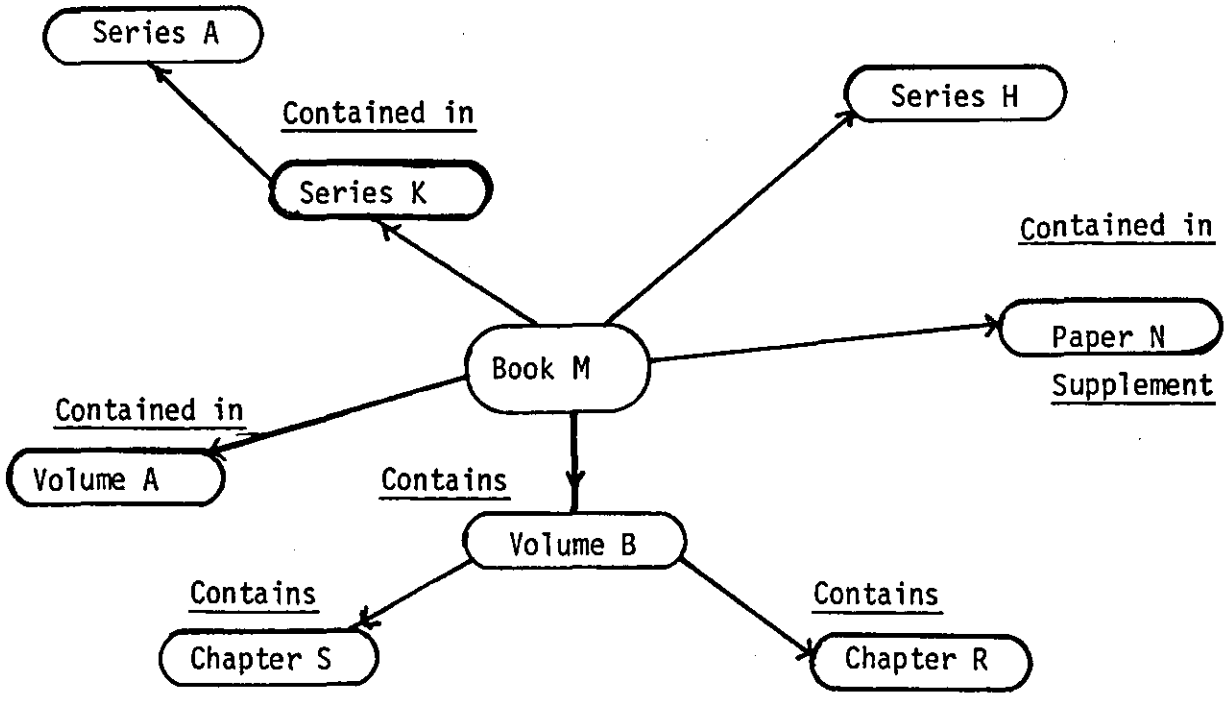


FIGURE 2.10 Literature relationships

d) This system has been designed around a special purpose application, which does not serve other applications readily.

2.4 The Publishing House Requirements

After analysing the database system which is used by the publishing house, it has been seen that there are many problems facing the company. These include the storage, security of the data and the retrieval of the data. To solve these problems the publishing house defined the following requirements:

1. The data should be stored in a compact and safe manner.
2. The data should be retrieved and accessed in an easy way.
3. The system should be structured in a flexible manner so that any changes in the data held or any change in the requirements of the system, can be dealt with easily and without having to re-design or rewrite very large parts of the system.
4. The management should be provided with reports concerning the legal and illegal entries to the system.
5. The data should be accessed through terminals.

In this thesis I have concentrated the work in such a way as to design a DBMS which could fulfill the requirements of this publishing house. A proposed relational model is designed and tested. Full details are given in the remaining chapters.

CHAPTER THREE

THE DESIGN OF THE PROPOSED SYSTEM

3.1 Introduction

A relatively recent development in database management systems has been the evolution of relational data models and much current researches are based on this model and its terminology.

A relational model is preferred to a hierarchical or network, because of its simplicity and it is natural in the context of the data structures associated with a bibliographic database.

The complexity of the hierarchical and network models has caused some problems to both models.

The insertion and deletion operations become quite complex in the hierarchical database system because of the strict hierarchical ordering. Also, users have to be careful when performing a delete operation because this operation could lead to the loss of information, if null records are not permitted. On the other hand, the network database does not separate the structural aspects of the data model from the physical placement of tuples, thus enhancing data independence.

This means the applications programmer has to be aware about the data representation, and programming can become extremely complex.

The relational approach has been chosen over the other because it offers various advantages and it will satisfy the user requirements. The many advantages of the relational model with respect

to the document retrieval systems (to regard data structures such as indexes and dictionaries as two dimentional) are, the relational view of data is basically a tabular view and the retrieval languages which are based on relational calculus or algebra, forms powerful languages for the manipulation of the database.

3.2 Objectives of the Proposed Relational Model

The objectives of the proposed database has been determined from the publishing house requirements. It was decided that the proposed database should have the following characteristics:

1. The database organisation is to achieve fast and flexible search capabilities.
2. The database should be planned in such a way that changes can be made to it without having to modify the application programs.
3. The user's view of data must be separated from the physical representation of data in storage.
4. The data must be stored and used with minimum cost.
5. Data in the database must be kept secure and private.
6. To provide the following main types of book listings required:
   a) Annual book listing
   b) Listing of books by category
   c) List of all books by author.

7. The management should be able to deal with all aspects of the security and to control the database.

8. The model must have the ability of answering all users queries and spontaneous requests as well.

9. New book items should be added easily to the database.

10. The database should be organised to accept a steady growth.

11. The model should be organised for the naive user requirements.

12. The system must be protected against hardware failures and various types of accidents which occur occasionally.

The storage of the data and the updating and insertion procedures must be such that the system can recover from these occurrences without harm to the data.

3.3 User's View of Data

The two-dimensional flat-file is regarded the simplest structure of the data representation for the users.

According to the publishing house system as discussed in the previous chapter, a card-based database was used. The cards were selected and sorted by hand for a particular user requirement.

The structure of the data which was used and as seen by the user is shown in Figures (3.1), (3.2) and (3.3).

From these figures, one can see the complex representation of some data structures and the absence of real associations between the data items. Figure (3.1) reflects the data structure which is used for producing book lists. This figure shows the repeating group problem which exists in this structure.

Therefore, the normalization procedures should be applied to restructure the data in a normalized form taking into account the user requirements. As a result of the normalization rules which were discussed in Chapter 1, the new user's view of data represents the natural associations between the data and the user requirements.

These user's views (subschemas) are the basis for the overall logical database description (schema) which will be discussed in the next section. The subschema represents a portion of the data which is oriented to the needs of one user. The database management software assembles the data described in the subschema from the data described in the schema automatically, and gives it to the user.

The new user's perception of the normalized data are shown in Figures (3.4), (3.5), and (3.6) respectively.

3.4 The Global Logical Database (Schema)

The logical database design is one of the database administrator's responsibilities. He has to design a model of data which can serve the needs of its users in the best manner.

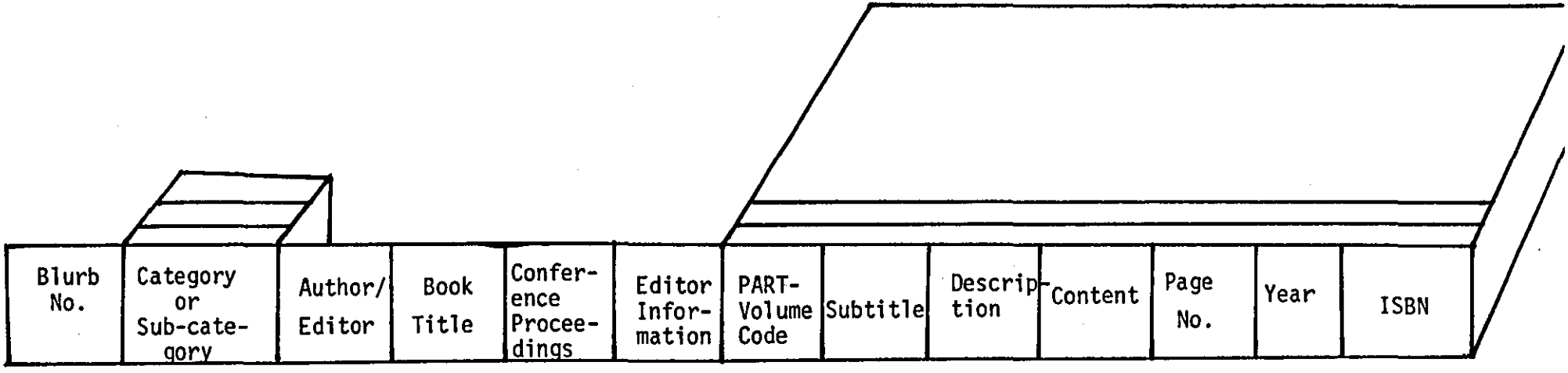


FIGURE (3.1) User's View of Data to Produce a Books List

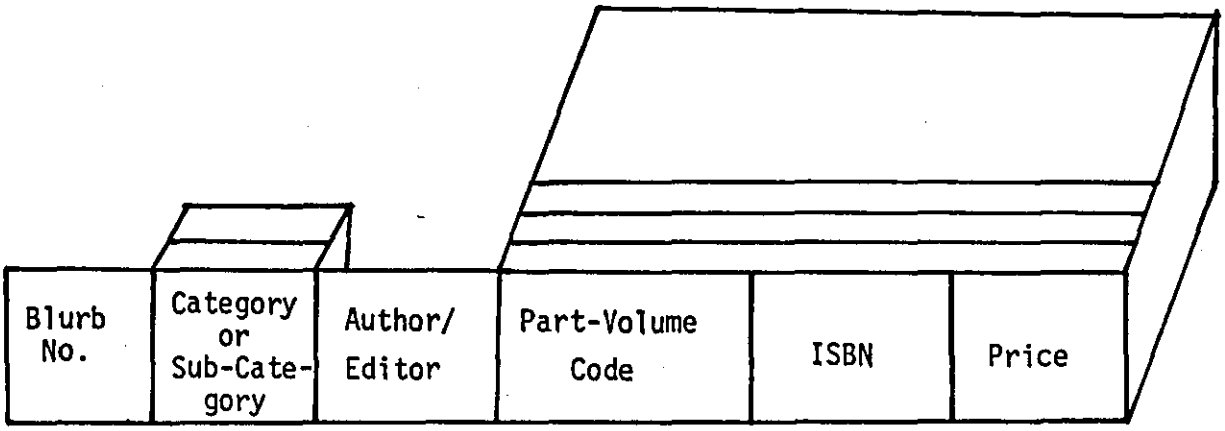


FIGURE (3.2) User's View of Data to Produce Price List

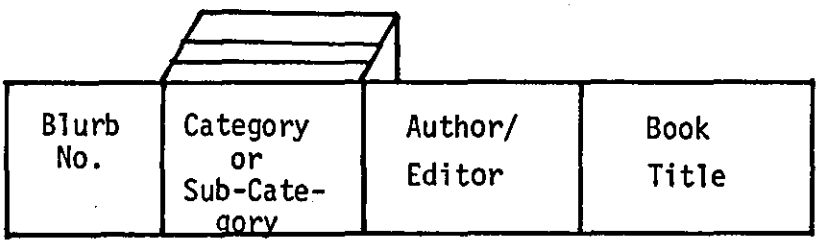


FIGURE (3.3) User's View of Data to Produce a Title List

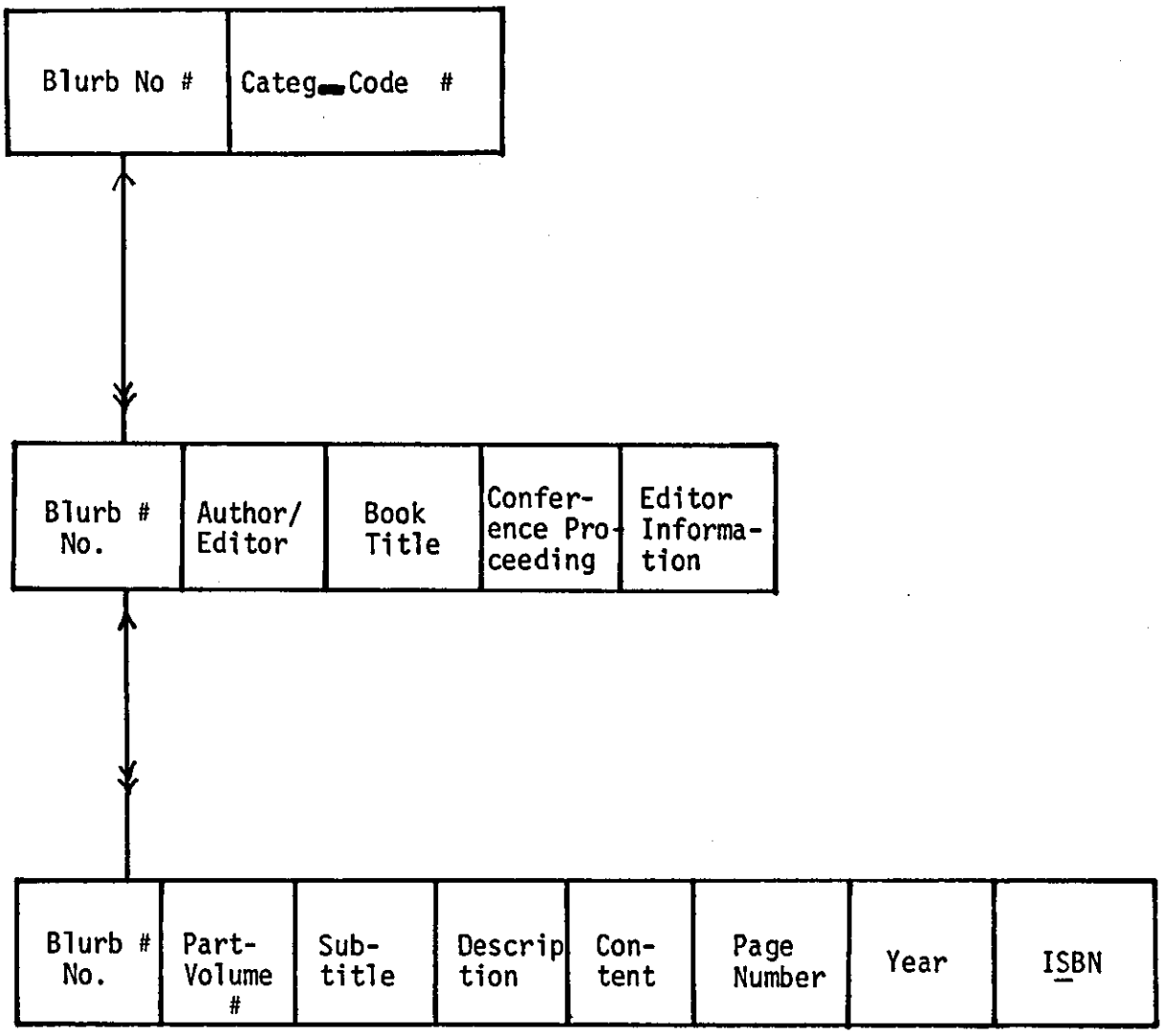


FIGURE (3.4) Normalized User's View of Data to Produce a Books List

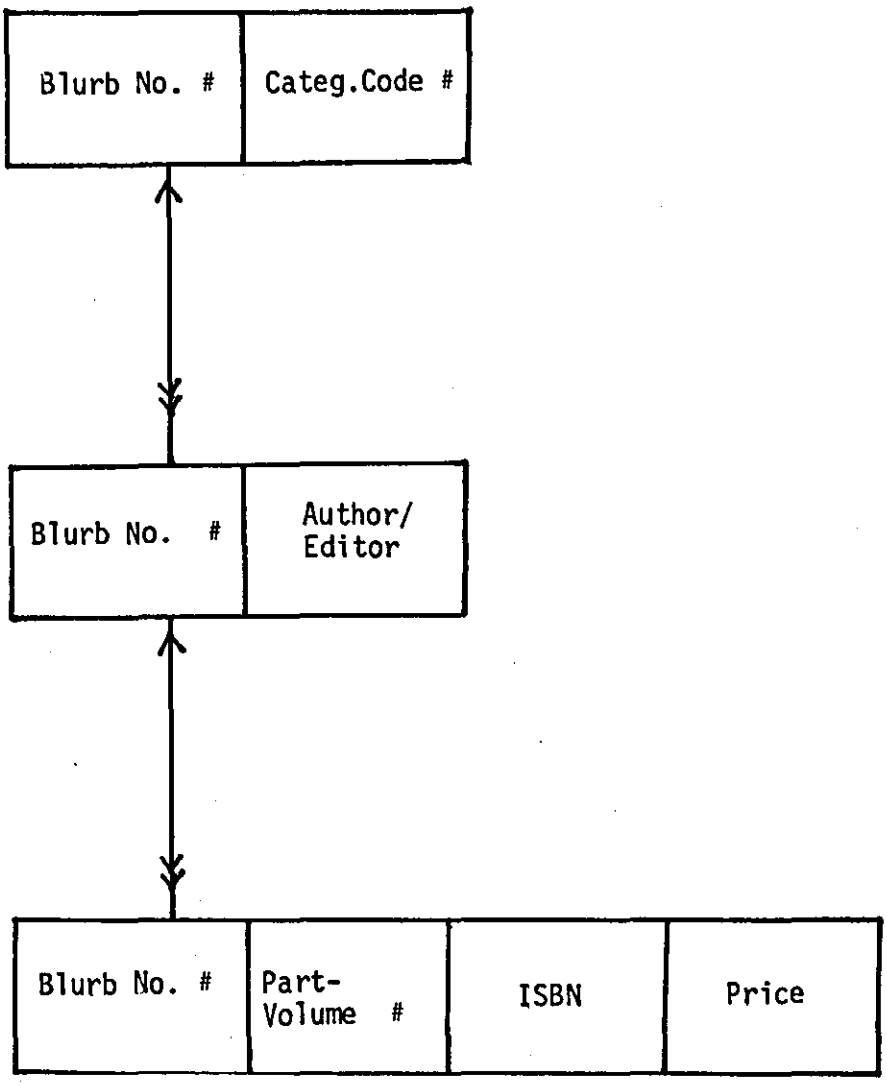


FIGURE (3.5) Normalized User's View of Data to Produce a Price List

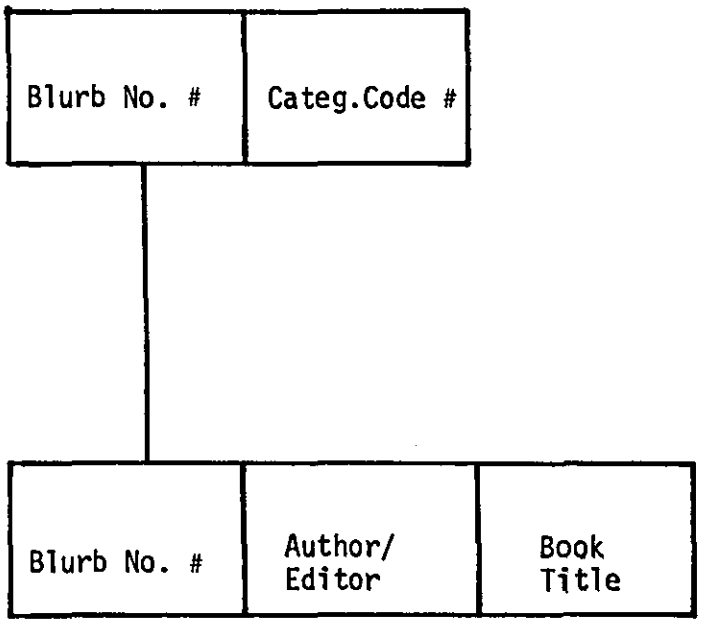


FIGURE (3.6) Normalized User's View of Data to Produce a Title List

Martin [1977] referred to canonical structuring as the best technique of the many different ways available in which a collection of data items can be associated to form a logical database. The application of its procedure to form a canonical schema will result in a minimal structure. The canonical schema as defined by Martin [1977] as "a model of data which represents the inherent structure of that data and hence is independent of individual applications of the data and also of the software or hardware mechanisms which are employed in representing and using the data".

The constructed data model by the canonical procedure offers many advantages which are as follows:

1. It increases the inherent stability of the database.
2. It gives the best chance of surviving future changes.
3. Its independence of the data representation (hierarchical, CODASYL, relational, or other structures).
4. It minimizes the risk of having to rewrite application programs as the database changes.

The canonical procedure has been applied to form the canonical schema as follows:

1. The first user's view of data has been taken and drawn in the form of a bubble chart (a graph with point-to-point directed links between single data items) representing associations of the two types; one-to-one and one-to-many. The concatenated key is drawn as one bubble, and its component data items drawn

as separate bubbles.

The hidden transitive dependencies were avoided and the representation of the user's view was ensured in the third normal form.

2. Further user's view is taken in account and drawn up in the same way as the first user's view.
   The two views were merged together and checked for any synonyms or homonyms in order to remove them if they appeared.

3. The primary key in the resulting graph was distinguished from the attribute nodes. (A primary key node has one or more single-arrow links leaving it).

4. The inverse association for each association between keys was added to ensure that many-to-many links between keys exist or not exist. If they exist and could be used at any time in the future, the association should be replaced by the introduction of an extra concatenated key incorporating the key data items that are linked.

5. The associations were examined to identify any which appear redundant in order to remove associations which from their meaning are genuinely redundant.

6. The previous four steps were repeated until all user views were merged into the graph.

7. The root keys were identified. (A root key is a primary key with no single arrow leaving it to another key).

8. The data items arranged into groups were redrawn (records, segments, tuples) with each having one primary key and its associated attributes.

9. Finally, the canonical schema was ready to convert into relational, CODASYL, or DL/1 schema.
   The resulting schema is shown in Figure (3.7) and the converted relational schema is shown in Figure (3.8).

3.5 Database Physical Structure

Every relation in Figure (3.8) represents a simple flat file structure. Each relation was given a name and all tuples in it were given key(s) which uniquely identify them. This structure will minimize the maintenance costs. The physical structure of the database is represented by storing the information concerned with relation (name of the relation, name of the attribute, and the form of the attribute representation) separately from the body of the relation which contains the data that makes up the information content of the relation.

The important advantage of this separation storage is to provide more complete data independence.

Some of the relations are stored in binary form, such storage will be valuable to the interactive query systems because it can easily handle the spontaneous queries.

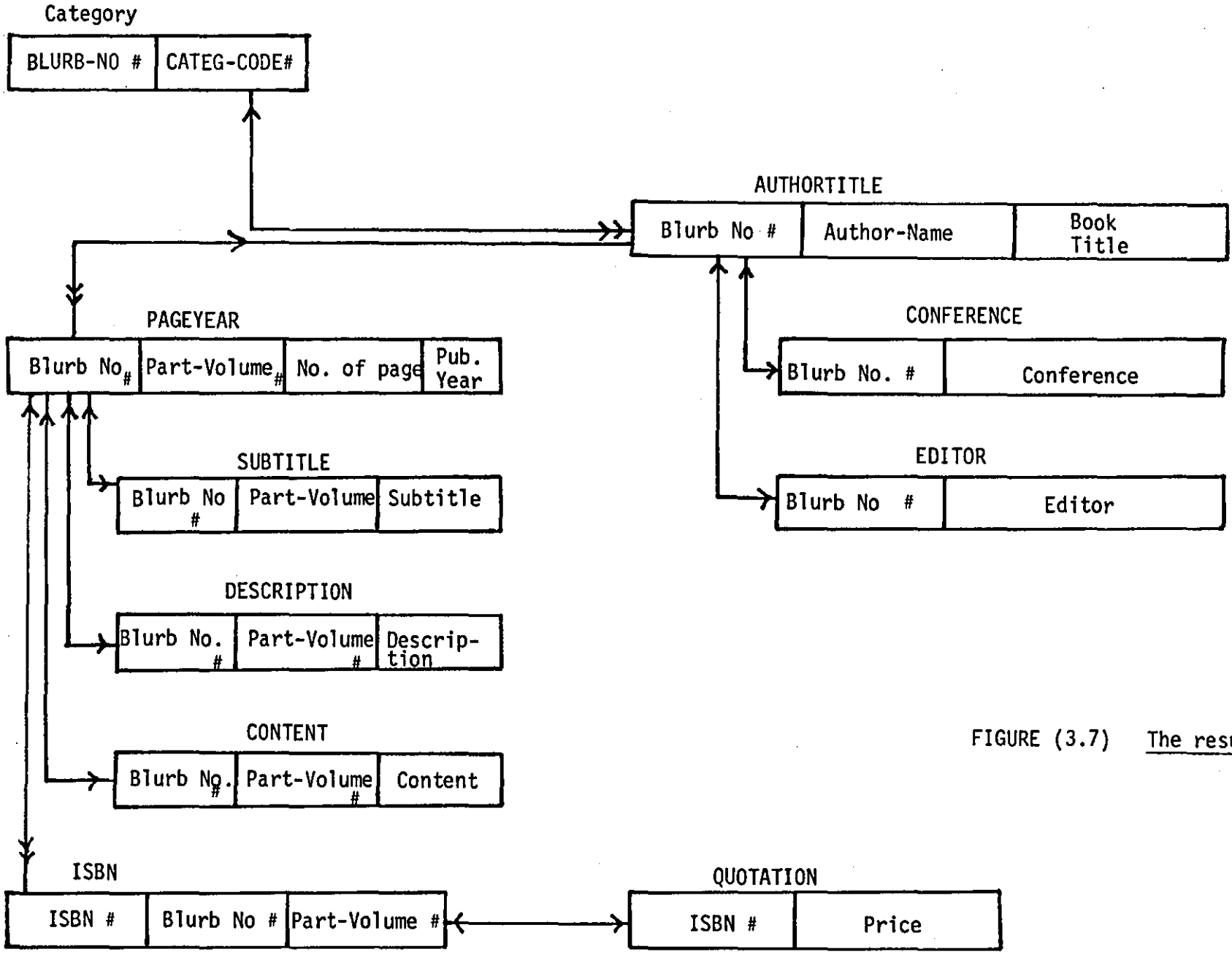


FIGURE (3.7) The resulting schema

CATEGORY (BLURB-NO#, CATEG-CODE#)

AUTHORTITLE (BLURB-NO#, AUTHOR-NAME, BOOK-TITLE)

CONFERENCE (BLURB-NO#, CONFERENCE)

EDITOR (BLURB-NO#, EDITOR)

PAGEYEAR (BLURB-NO#, PART-VOLUME#, NO-OF-PAGE, PUB-YEAR)

SUBTITLE (BLURB-NO#, PART-VOLUME#, SUBTITLE)

DESCRIPTION (BLURB-NO#, PART-VOLUME#, DESCRIPTION)

CONTENT (BLURB-NO#, PART-VOLUME#, CONTENT)

ISBN (ISBN#, BLURB-NO#, PART-VOLUME#)

QUOTATION (ISBN#, PRICE)

FIGURE (3.8) The converted relational schema for the resulting canonical schema

Also, it will save spaces for reasons that some books do not include such fields (conference, description, content).

3.6 Limitation of Some Fields Size

One of the problems encountered was that there were no guidelines on the size of items that were to be included in the 'blurb'. For instance, what were the average and maximum number of characters in any book title, or author(s) name etc.

To overcome this several book lists by different publishers were examined and statistics collected on all these items that were of uncertain length.

Eventually it was decided after sufficient information had been obtained to take a decision on the lengths of various fields concerned.

The fields were then fixed in order to make the operations on relations easy and fast.

3.7 The Relational Operations

A variety of operators are defined by the relational mathematics with which these relations may be manipulated in order to achieve any desired requirements in a tabular form of the data.

Relational algebra is considered as a collection of high-level operations on relations. Its operation is one which takes one or more relations as its operand(s) and manipulates them to form a new relation.

Date [1977] referred to a complete set of operators which are involved in the retrieval and storage operations.

A set of algebraic operations (union, intersection, difference, and complement) are used in the database to manipulate the relations and special relational operations (selection, projection, and join) as well. The usage of these operations can be divided into two functions:

1. Retrieval operations which involve the following operators:

   a) SELECT is an operator which manipulates a specific relation to extract those tuples within the relation for which a specified predicate is satisfied. The extracted tuples will be written on a new relation which is of the same type as the original one.

   b) PROJECT is an operator which manipulates one relation to extract a vertical subset of a relation. A subset can be obtained by selecting specified attributes and eliminating others. Also, the project operator can provide us with a way to re-order the attributes of the new relation.

   c) JOIN is an operator which operates on two specified relations to construct a new relation. The join operator is

mostly based on equality of values in the common domain. It might be used for each of the other comparison operators ($\neq$, $<$, $>$, $\leq$), if required.

d) COMPLEMENT is an operator which acts on two specific relations to construct a third relation which is regarded as the complement of one of the old relations only.
For example:

Let $A \subset S$. The complement of A in S is the set of elements that belong to S but not to A. The complement of A in S denoted by A^-. Thus we have $A^- = (X;\ X \notin A)$.

2) Storage operations which involve the following operators:

a) UNION is an operator which acts on two relations of the same type to produce a third relation of the same type.
For example:

A U B (A UNION B), of two relations A and B means that the set of all tuples which belong to at least one of the two relations A, B. Thus:

$A \cup B \{X;\ X \in A \text{ OR } X \in B\}$.

b) MINUS is an operator which operates on two specified relations (A,B) to construct a new one of the same type.
For example:

A-B (A MINUS B) = $\{X;\ X \in A \text{ and } X \in B\}$

3.8 Privacy and Security

Data in the proposed relational model is planned to be secure and private in order to protect it from any unauthorized persons and to give the rights for the individuals and organizations to determine for themselves when, how and to what extent information about them is to be transmitted to others.

Privacy can be achieved through a security mechanism.

The designed database model included many different procedures of authentication for the users and these are as follows:

a) The checking of the user number (code).

b) The checking of the user password by typing the password on the terminal to compare it with the stored password of the same person. The typed password will not appear on the terminal for security purposes.

c) The checking of a black mark which means that such users are not allowed to use the system after a specified number of illegal attempts of entry.

d) The checking of a time period each day when specified users are allowed to use the database.

e) Every user is allowed to run only a specific type of job, so that the system will prevent any user from using unauthorized jobs.

f) All the legal and illegal entries to the database are logged on a specific file for management information purposes.

3.9 System Integrity

In reality, there is no absolute system integrity but the system must be protected against hardware failures and various types of accidents which will occur occasionally. The data in the database must be ensured that it is accurate at all times.

The proposed database system is designed to have the ability to recover from accidental corruption to the database either through human or system error.

The database has the following integrity constraints:

a) Each transaction passes through a validation procedure to ensure consistency and completeness.

b) By copying the database periodically and keeping the copy in a safe place.

c) Each relation is defined by a primary key and no two tuples in the same relation having the same values.

d) The values which occur in a particular attribute must lie in a specified domain i.e., they must lie within certain ranges or they may conform to certain specified rules.

e) The update of a specific field might be followed by certain rules, like the update of a book price which should not exceed a determined value.

3.10 The User Interface

There are a variety of ways to organise the information flow between the computer and the user of the information. On real-time systems the simplest form of information flow is that in which the person requiring information or entering data communicates directly with the computer in a two-way dialogue.

The user interface is designed as a directed dialogue interface in order to achieve rapid familiarity and appeal through recognisable elements in the system messages.

This interface includes all the security procedures which were discussed in the previous section.

Such interfaces enable the database system to be centred upon their various users who interact with the system through computer terminals. It is the way that users can communicate with the database and controls all the access to the system and logs both the successful and the unsuccessful attempts to use the database resources.

The results of this design is a consistent user interface which is automatically derived from a set of files (user account file, and menu file), and containing flexible security features supported by system management reports drawn from the usage log.

Each user is presented with a menu-selection which is determined by the records held on the user accounts file and the menu file.

This is achieved in the following way:

1. The user is first identified by account number and password check.
2. If the user is allowed to use the database the menu will be structured and displayed from the menu file with the contents and the structure of the interface belonging to that user.
3. The user can select the required function from the displayed menu. The user interface will prevent the user from doing other functions which do not appear in the user menu.

The user interface components are shown in Figure (3.9).
The user interface mechanism is shown in Figure (3.10).

3.11 Database Administrator's Dialogues (Security Officer Dialogues)

In the enhanced database system dialogues are provided to the database administrator employing menu, question and answer, and the command techniques. This dialogue allows the administrator to establish all the security components before users may commence operations. This includes the creation of the initial records in the user account file and the menu file. The dialogue is designed to assist the administrator or the manager to create and maintain these files as required.

Also, such dialogue provides an on-line interrogation on any of these files and a log file summary feature to assist in the file maintenance. The security officer dialogue enables the management to control the overall database by centralizing and monitoring the database activities.

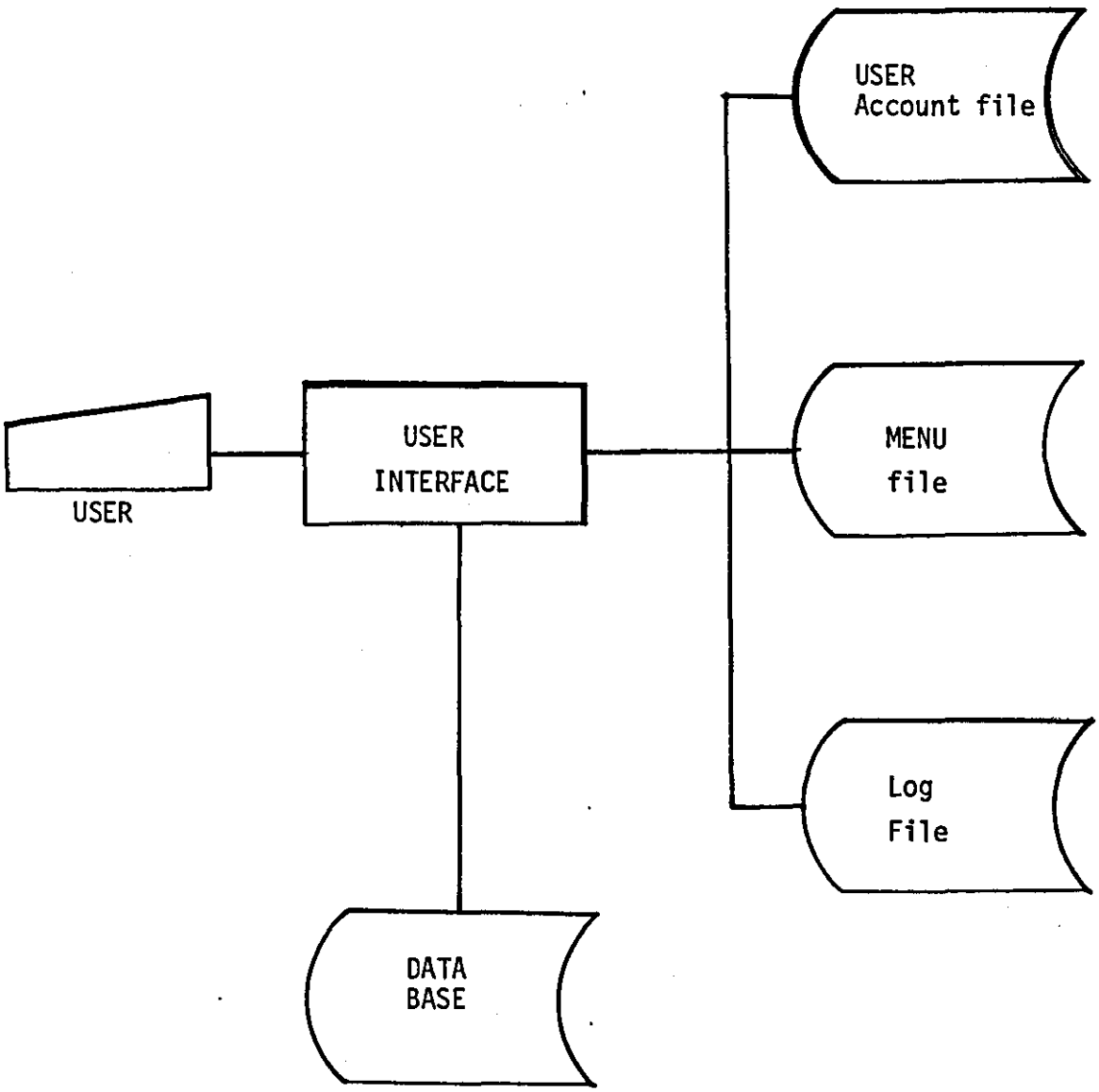
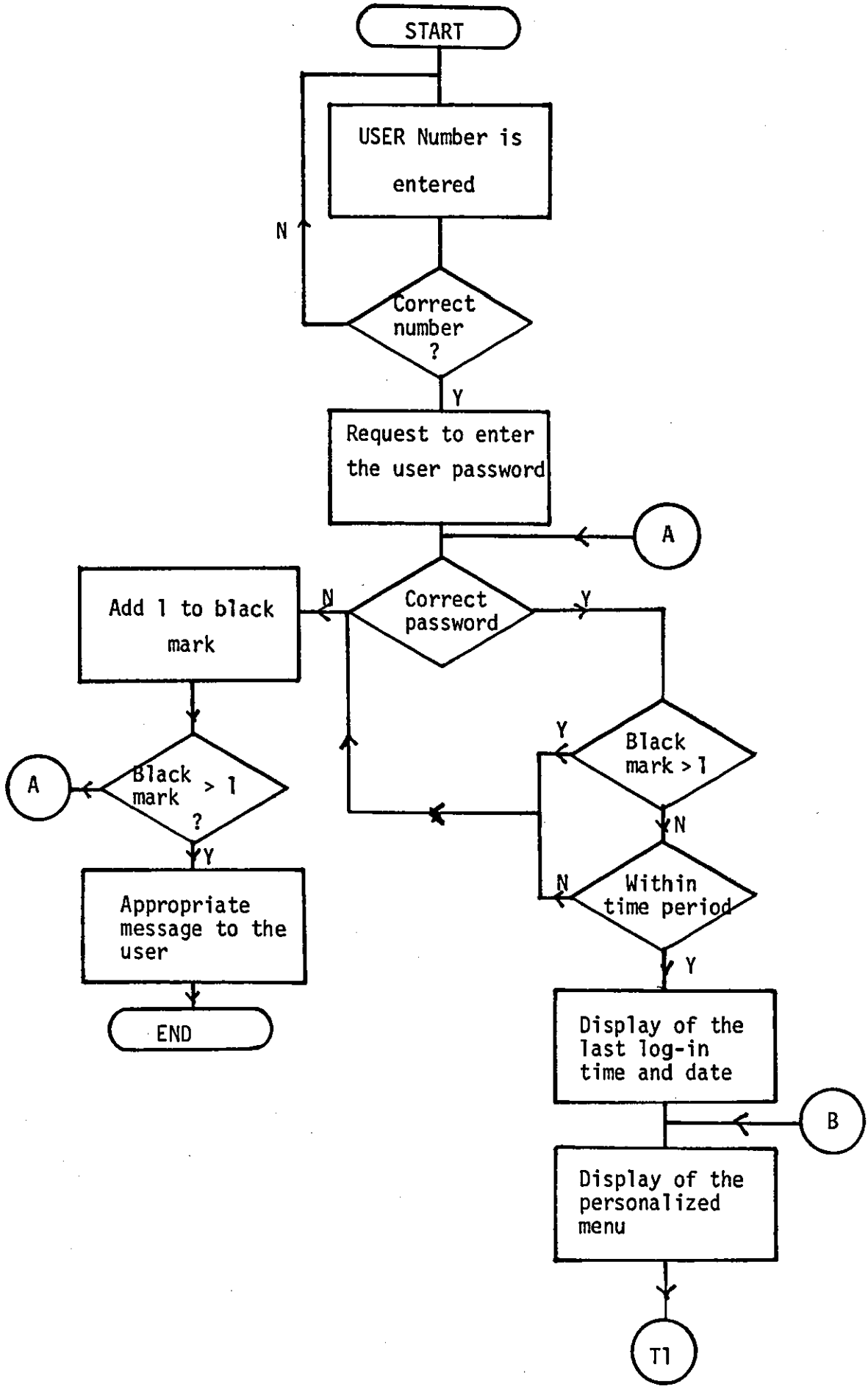


FIGURE (3.9) User Interface Components


START
USER Number is entered
N
Correct number ?
Y
Request to enter the user password
A
Add 1 to black mark
N
Correct password
Y
A
Black mark > 1 ?
Y
Appropriate message to the user
END
Y
Black mark > 1
N
N
Within time period
Y
Display of the last log-in time and date
B
Display of the personalized menu
T1

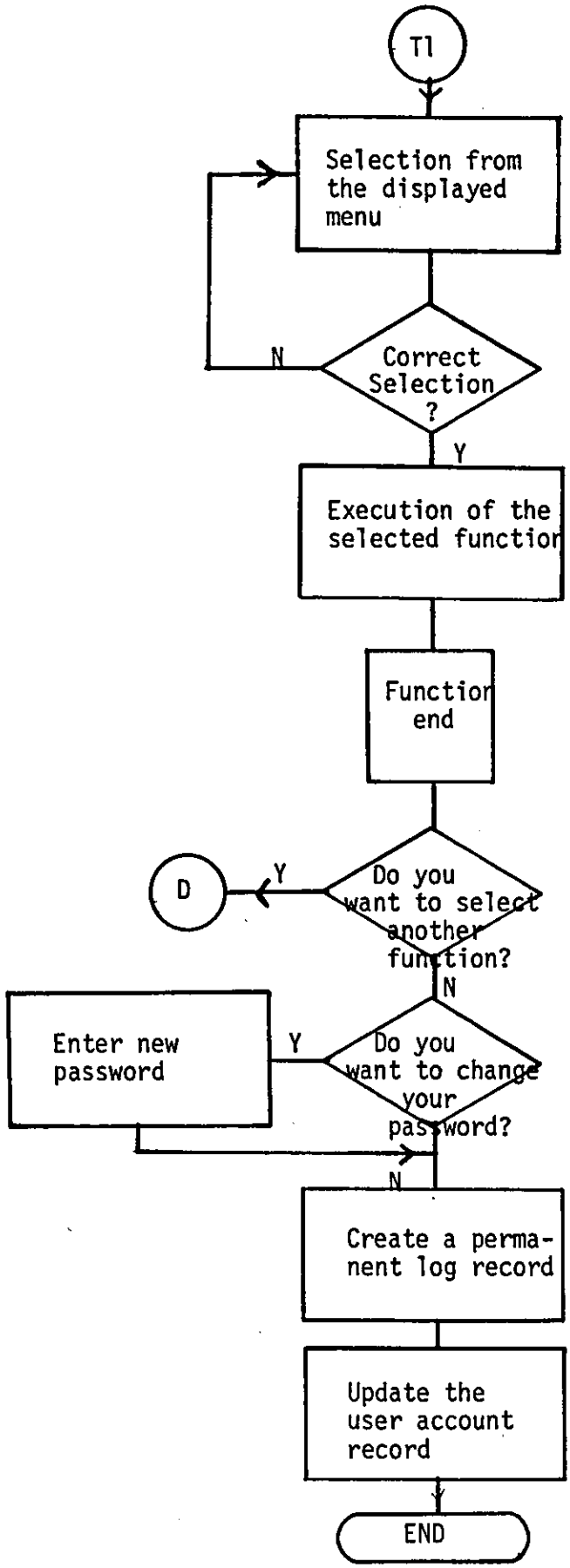


FIGURE (3.10)
A flow chart presenting features of the user interface mechanism

The manager of the administrator can use the dialogue through the user interface because there is no specific terminal available for them.

The administrator's dialogues structures are shown in Figure (3.11).

The dialogue is designed to be used easily by both the naive user and the computer specialists.

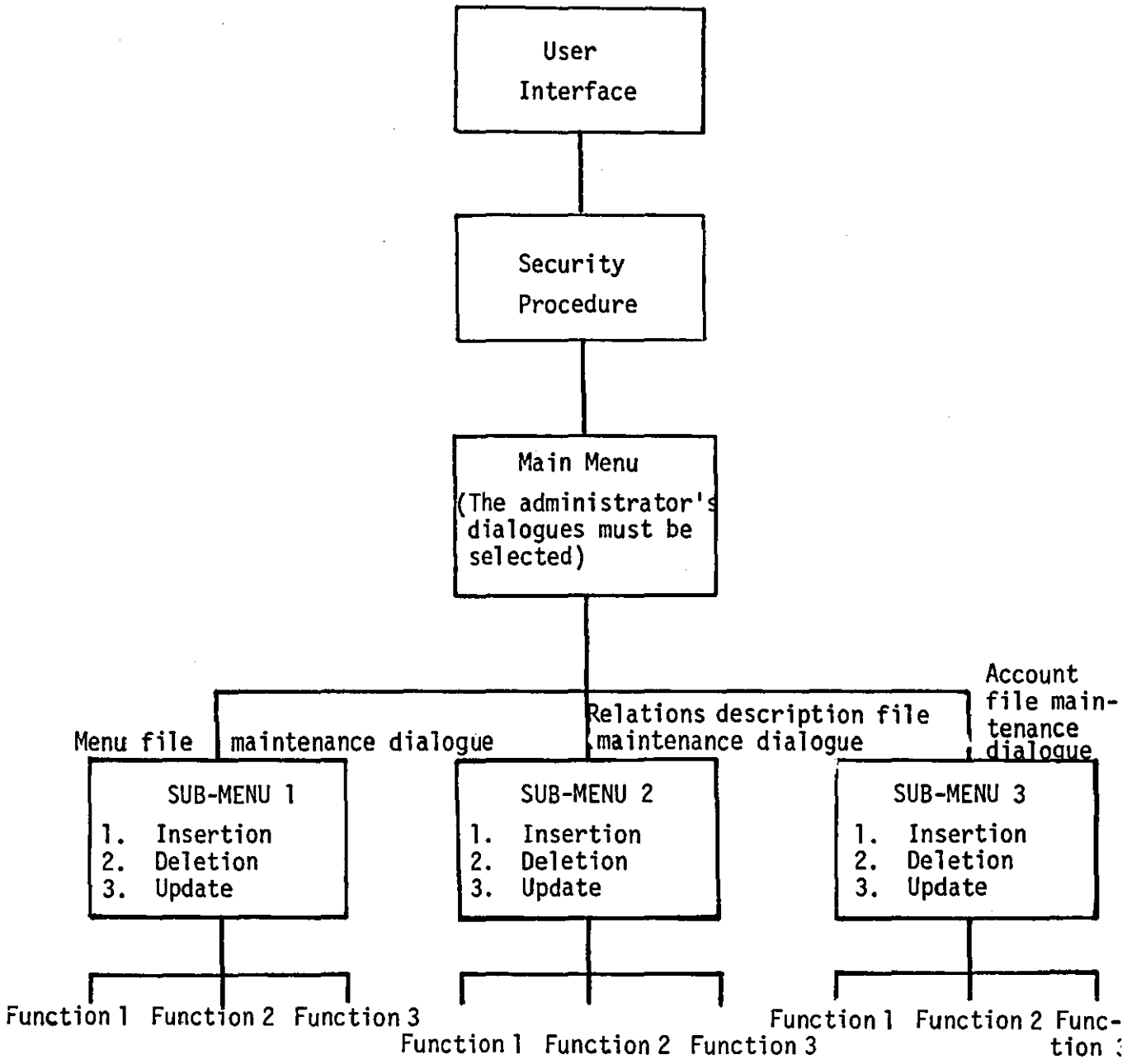


FIGURE (3.11) Administrator's dialogues structures

CHAPTER FOUR

IMPLEMENTATION

4.1 Introduction

Before considering the actual implementation, a brief introduction is given to discuss the choice of the computer system which is preferable for the project requirements. However, the database required an interactive computer service for its implementation. Also, it was agreed that COBOL 74 is the best language for the system implementation since it is always used to implement commercial applications.

There are many interactive computer systems available in the Computer Department and the Computer Centre at Loughborough University. These are:

The Department of Computer Studies PDP 11/40 is the Computer system which is run under the UNIX operating system. It is universally regarded as a powerful computer system,however the COBOL language is not available on the UNIX.

The ICL 1904 is the computer system which is run under GEORG2L operating system. Modular 1 Terminal system is one of the computer system services which is responsible for all the interactive side of the work and all terminals are controlled by it. [Prentice, 1979A]. The system provides the ability to manipulate files on the 1900 and submit these as batch jobs from terminals at various sites on the campus. The COBOL 74 language is available on the computer operating system.

Another computer system in service is the PRIME 400. The COBOL language is also available on this.

The PRIME 400 computer system was chosen to implement the proposed database because not only is it easy to learn and use, but it will satisfy the designed database requirements.

The next section of this chapter will be an introduction to the PRIME system.

4.2 The PRIME 400 Series

4.2.1 System Structure

The PRIME 400 consists of two 768 byte Prime processors, 320 megabytes of disc storage, a card reader, line printer, paper tape reader/punch and a magnetic tape deck compatible with the ICL 1904S which is used to transfer files to and from the ICL 1904S filing system under operator control [Prentice, 1979B]. As it is an interactive system, it is accessed by terminals spread about the campus and connected to it by telephone/telegraph lines via a Gandalf exchange.

The tape equipment allows input and output for interchange with other computers.

Ultimately the card reader will allow the running of simple non-interactive jobs (e.g. compilations of interactive file transfer) from the card decks. The system has a considerable variety of

terminals attached to it. These are divided into three main classes as follows:

1. The 10 character per seconds terminals (e.g. teletypes) which operate in half duplex mode (i.e. the terminal prints what is typed and sends it to the computer).

2. The 30 character per second terminals (e.g. VDUs) which operate in full duplex mode (i.e. the terminal sends the typed character to the computer which then generally returns it back to the terminal).

3. The 30 character per second terminals (e.g. Trend 800) which operate in a similar way as in (2).

The terminals are not connected directly to the computer but via a switching system called PACX (Private Automatic Computer Exchange).

4.2.2 The Editor

The PRIME system has several useful and powerful utilities. The 'text editor' is one of them which enables ease of change when developing files and is very simple to use. It is a line-oriented system.

EDITOR is a system designed to assist the user to create and edit text files on the computer [Gerrard, 1978]. There are many commands available in the EDITOR. Each command is followed by one space and a specified parameter.

Normally, the data and the programs are entered, checked and corrected by using the EDITOR (ED).

Editor has a special file area called the work file which is reserved for its own use. EDITOR puts all input into this work file and does all its own editing (searching, examining, changing) on the contents of this work file.

The EDITOR has two modes: Input mode and Edit mode. Any user can switch from one mode to the other quite easily. In Input mode, the EDITOR treats whatever the user types as text which is put directly into the user work area, line by line. In Edit mode, the Editor treats the user input as commands, and executes each using the contents of the work file.

Finally, it was concluded that these EDITOR's facilities provided a beneficial assistance to the database implementation process.

4.2.3 The Multiple Index Data Access System (MIDAS)

MIDAS, provides the users with a series of program and subroutines for the creation and maintenance of keyed/index and/or keyed-index direct access files.

Sometimes, the Index Sequential Access (ISAM) files are referred to as Keyed-Index files.

The usages of MIDAS are as follows:

1. To create/modify the template (file description) which means that the user defines the data file, indices, etc. by loading the CREATK utility.

2. To build the data file which exists in a text or binary file which can be converted to a MIDAS file by loading KBUILD utility.

3. To maintain a file which means that data can be added, deleted, changed or viewed.

4. To reconstruct the files which means that MIDAS can perform the house-keeping responsibility. The files can be reconstructed after significant maintenance (REMAKE), deleted in part or full (KIDDEL) or rebuilt crashes (REPAIR).

4.2.4 The PRIME COBOL

The Prime COBOL is based upon the American standard (ANSI) X3.23-1974.COBOL is a conventional compilation system. The compiler processes source programs from disc file and produces files of loadable binary code. This is loaded together with subroutine libraries by the loader to produce a binary program file which is finally loaded and run. The compilation and loading process is not interactive and so is run by a simple batch system. Prime's COBOL compiler operates on COBOL source code to generate object code. It is also possible to generate a program listing only. Since syntax checking can be achieved in a shorter period of time, this feature

can produce a quick and useful reference to the source program. When the program is compiled and loaded it may be run, inputting data and obtaining results interactively at the terminal.

4.3 Program Development and Implementation

The program development and implementation includes the following:

1. The user interface.
2. The database administrator's dialogues.
   a) User account file maintenance dialogue.
   b) Menu file maintenance dialogue.
3. Security reports.
   a) User account report.
   b) Interface report.
4. The relational operators.
   a) Retrieval operators.
   b) Storage operators.

The programs are coded in COBOL and entered (source code) via the terminals by using the PRIME EDITOR in the input mode. Also, the EDITOR is used in the edit mode to change or correct any incorrect code. After each end of a compilation process, the source program is corrected if necessary using the EDITOR in the edit mode and the cycle repeated.

The top-down test approach is used to validify the database performance. This approach is regarded as a strategy of testing the high-level modules of a system before low-level modules have been coded and possibly before they have been designed [Yourdon, 1979]. The implementation of this approach includes an important advantage which resolves any problems encountered by the designer in the early stages.

4.3.1 Implementation of the User Interface

The user interface consists of several modules or sub-systems. It has two main functions. One function is to supervise the accesses to the database. The second function is to enable users, after they successfully identify themselves, to access the database system. The first test made on the interface was to check the authentication process.

When the user wishes to access the database, the security control asks the user to type in the user code, and after the user code has been entered, the security controls will check the user code against the user code which is recorded on the user account file. The security control will ask the user once again to type the correct user code if it was incorrect or it will ask for the password for correct user code. The system will allow one further attempt for the incorrect password. The system checks further verification of user authority. Each valid or invalid attempt is recorded in a log file.

After the user has completed the authentication sequence and is permitted to use the system, he is shown a personalized menu from the menu file according to the job codes that are recorded on the user account record. The user can choose the required function. When this function is completed he is given the choice of making further functions from the same menu.

The second test was made by an inexperienced user to see the extent of acceptance of the user to the interface performance.

The interface was tested for different paths to ensure that it behaved correctly and avoided the breakdown of the system.

4.3.2 Implementation of the Database Administrator's (DBA) Dialogues

Three modules are used by the DBA to maintain the security files. Each module represents a dialogue which performs a certain routine. This routine is also divided into sub-routines.

These dialogues can be called and used from the displayed menu on the screen to the security officer or to the DBA only.

The first dialogue is used for the menu file maintenance, the second for the user's account file maintenance and the third dialogue is used for the relations description file maintenance. Each dialogue has been tested which proved the ease of usage of these dialogues to maintain the files even by naive users.

The use of the question-answer technique which is employed in these dialogues assists the users to manipulate the files with a

simple directed manner. The dialogues were tested by using meaningful artificial data obtained from a model problem..

4.3.3 Implementation of the Security Reports

The security reports can be obtained from two programs. The first is the interface report program which produces the interface report. The second is the user account report program. These programs can be selected and used from the displayed menu by the security officer.

These programs were implemented and the reports were obtained. It was concluded that both reports were very important to the security officer because it assists him to control the database access and to inform him about any legal or illegal entry to the system. Also, it could help him to set up new authentication procedures and conditions for the users. On the other hand, these reports include vital information which has been used for the maintenance of security files.

4.3.4 Implementation of the Relational Operators

Two groups of relational operators were implemented in this project which demonstrated the relational foundation theory and the powerful retrieval capability of the relational model.

The first implemented group were the retrieval operators. The second group were the storage operators.

The implementation of these operators provided the ability to manipulate and to extract any type of data that is required by the users.

The Retrieval Operators

SELECT is an operator which is implemented to construct tuples satisfying a specific condition. This operator needs five parameters. The first parameter is the input relation name and the fifth and last parameters is the output relation name. The remaining parameters represent a specified condition to extract the satisfied tuples from the input relation. This condition includes one type of comparison between a subset of a relation and a constant. The second parameter represents a subset name and the third is one of the comparison operators =, $\neq$, >, <, $\leq$. The fourth parameter may represent a constant number or name.

The output relation will contain all tuples which satisfy the specified condition. The process logic of the SELECT operator is shown in Figure 4.1.

PROJECT is an operator which is implemented to enable a user to select which columns he requires from a relation, and to specify in what order he wants them. This operator needs three parameters. The first parameter is the input relation name and the third is the output relation name. The second parameter includes a list of one or more attribute names. The process logic of the PROJECT operator is shown in Figure 4.2.

JOIN is also an operator which is implemented to enable a user to join two relations which share a common data-item type. This operator is actually an equi-join, that is, a join based on the equality of values in the common domain. This operator needs four parameters. The first and second parameters are the two input relation names. The third one is the shared attribute name and the fourth parameter is the output relation name. The process logic of the JOIN operator is shown in Figure 4.3.

COMPLEMENT is an operator which is implemented to complement specific relations in order to obtain the required data. This operator requires four parameters. The first parameter is the name of the relation to be completed and the second one is the name of the complement relation. The third parameter is the name(s) of the shared data-item type and the remaining parameter is the output relation name. Figure 4.4 represents the process logic of the COMP operator.

The Storage Operators

UNION is an operator which is implemented in order to add new tuples to a specific relation. This operator needs two parameters, which represent the input relation names. Figure 4.5 shows the process logic of the UNION operator.

MINUS is an operator which is implemented to enable users to delete a specific tuple(s) from a relation. This operator needs

the same parameters as used with the UNION operator. Figure 4.6 shows the process logic of the MINUS operator.

4.3.5 Direct Use of the Operators

The relational operators used were tested directly in a specific sequence according to the user requirements. The lists required by the publishing house were obtained as a result of these retrieval operations.

The update operation is achieved by a suitable sequence of MINUS and UNION operations.

The direct use of the operators proved clearly the ability of the relational module to retrieve and answer any type of enquiry but the user of these operations must know exactly how the data is structured in the database.

Thus, the personalized menu method allows the user to select the facility he requires without needing to specify or to know how the data is organized.

4.3.6 Programming Technique

The programming technique used for a large part of the database programs was the top-down design and each program was coded to some extent as a structured program. Structure is an approach to programming in which is concerned with clarity as well as effectiveness [Gane, 1979; Ashley, 1980; Lyons, 1980]. A structured approach

makes a complex subject easier to learn and helps the user to develop good coding habits automatically. This technique involves the coding of levels of blocks of instructions to do basic functions and these blocks are performed once a certain function is called. This means the elimination of the GO TO statement as much as possible and on using the PERFORM statement. The GO TO effect is to transfer control to the named paragraph and to leave it there. Unlike the PERFORM, GO TO does not return. Also, it does not even keep track of where in the body of the program it came from. Thus, when the GO TO instruction is used extensively in programs, it becomes very difficult for the reader to understand what the program is doing, much less how it carries out its task.

Thus, by eliminating the GO TO statements that jump backwards and forwards in the program listing, it will result with codes that can be read and understood by a person who is not a programmer at all.

In fact, if we begin reading code at the top of the page, and continue in a straight-line fashion then by the time we reach the bottom of the page we have completed a specific process.

Structured coding makes the testing and debugging part of programming much more manageable.

In addition, when a program is developed from the top down and is coded as a structured program, there is a greater confidence in the reliability of the program. This confidence is based on experience. Installations developing programs using this technique

have found that these programs have fewer errors at the end of a year's use than in comparable programs in which these methods were not used.

Finally, the process logic of each relational operator has been written out in English sentences, using capitalization and indentation conventions. Programming in this fashion is known as structured English (see Figures 4.1, 4.2, ... 4.6).

4.3.7 User Subroutines

There are several subroutines which have been written and tested as a part of the database model to constrain the model performance.

1. PRINT:

This subroutine has two functions, the first is to provide a listing of a given relation, including the relation description components and the data of each component. The second function is responsible for producing the required book lists in the correct format.

e.g. PRINT <relation name>

or PRINT <relation name> (<the required list>)

2. SORT:

Many SORT subroutines have been written and tested to help the user to order the data as required. These subroutines are included in the database operations and the sort can be achieved automatically.

```
SELECT OPERATOR
     DO PARAMETERS-VALIDATION
     DO TUPLES-SELECTION
     DO END-ROUTINE
PARAMETERS-VALIDATION
     IF parameters are correct
     store the constant (included in parameter 4) in AREA-A
     ELSE (not correct)
     SO display 'ERROR' message
     DO END-ROUTINE
TUPLES-SELECTION
     REPEAT TAKE-A-TUPLE until all tuples have been tested
TAKE-A-TUPLE
     take a tuple from input relation
     Store the data-item value (indicated by parameter 2) in AREA-B
     DO DATA-COMPARISON
     DO TUPLE-STORAGE
DATA-COMPARISON
     IF parameter-3 = "E"
     THEN DO EQUAL-ROUTINE
     ELSE IF parameters-3 = "L"
     THEN DO LESS-ROUTINE
     ELSE IF parameter-3 = "G"
     THEN DO GREATER-ROUTINE
     ELSE IF parameter-3 = "LE"
     THEN DO LESS-EQUAL-ROUTINE
```

```
     ELSE (GE)
     SO DO GREATER-EQUAL-ROUTINE
EQUAL-ROUTINE
     IF AREA-B E AREA-A
     Set flag to one
     ELSE
     Set flag to zero
LESS-ROUTINE
     IF AREA-B L AREA-A
     Set flag to one
     ELSE
     Set flag to ZERO
GREATER-ROUTINE
     IF AREA-B G AREA-A
     Set flag to one
     ELSE
     Set flag to ZERO
LESS-EQUAL-ROUTINE
     IF AREA-B LE AREA-A
     Set flag to one
     ELSE
     Set flag to zero
GREATER-EQUAL-ROUTINE
     IF AREA-B GE AREA-A
     Set flag to one
     ELSE
     Set flag to zero
```

```
TUPLE-STORAGE
     IF flag = 1
     Store the new tuple in the output relation
     Set flag to zero
END-ROUTINE
     Terminate Program
```

FIGURE (4.1) Structured English of the Process Logic of the SELECT Operator

```
PROJECT OPERATOR
     DO PARAMETERS-VALIDATION
     DO PROJECTION-PROCESS
     DO END-PROCESS
PARAMETERS-VALIDATION
     IF parameters are correct
         set count to zero
     ELSE (not correct)
         SO display "ERROR" message
     DO END-PROCESS
PROJECTION-PROCESS
     REPEAT GET-A-TUPLE UNTIL all tuples have been processed
GET-A-TUPLE
     Get a tuple from input relation
     REPEAT DATA-ITEM-PROJECTION UNTIL all items-list have been
         stored from the input tuple to the new tuple
     Set count to one
     Store the new tuple in the output relation
DATA-ITEM-PROJECTION
     Transfer one item's tuple [items-list (count)]  to the new
         tuple
     Add 1 to count
END-PROCESS
     Terminate Program
```

FIGURE (4.2) Structured English of the Process Logic of the PROJECT Operator

```
JOIN OPERATOR
     DO PARAMETERS-VALIDATION
     DO JOIN-PROCESS
     DO END-PROCESS
PARAMETERS-VALIDATION
     IF parameters are correct
         Set new tuple area to space
     ELSE (not correct)
         SO display "ERROR" message
         DO END-PROCESS
JOIN-PROCESS
     REPEAT GET-A-TUPLE-FROM-FIRST-RELATION UNTIL all tuples
         have been tested
GET-A-TUPLE-FROM-FIRST-RELATION
     Get a tuple from the first input relation
     REPEAT-GET-A-TUPLE-FROM-SECOND-RELATION UNTIL all tuples
         have been compared with the first relation tuple
     Initialize the second relation
GET-A-TUPLE-FROM-SECOND-RELATION
     Get a tuple from the second input relation
     IF both tuples have equal values in the common shared data items
         pair the two tuples in the new tuple
     Store the new tuple in the output relation
     ELSE (not equal)
         So set new tuple to space
END-PROCESS
     Terminate program
```

FIGURE (4.3) Structured English of the Process Logic of the JOIN Operator

```
COMPLEMENT OPERATOR
     DO PARAMETERS-VALIDATION
     DO COMPLEMENT-PROCESS
     DO END-PROCESS
PARAMETERS-VALIDATION
     IF parameters are correct
         DO GET-TWO-TUPLES-FROM-INPUT-RELATIONS
     ELSE (not correct)
     SO display "ERROR" message
     DO END-PROCESS
GET-TWO-TUPLES-FROM-INPUT-RELATIONS
     Get a tuple from the first input relation (R1)
     Get a tuple from the second input relation (R2)
COMPLEMENT-PROCESS
     REPEAT KEYS-COMPARISON UNTIL both inputs keys equal to high-
         value
KEYS-COMPARISON
     IF key 1 E key 2
         DO STORE-R1-TUPLE
     ELSE IF key 1 G key 2
         DO CREATE-A-TUPLE-FROM-R2
     ELSE (key 1 L key 2)
         DO STORE-R1-TUPLE
STORE-R1-TUPLE
     Store R1 tuple in the output relation
     Get next R1 tuple
     DO R1-END-CHECK
```

```
R1-END-CHECK
     IF end of R1
         Set key 1 to high-value
CREATE-A-TUPLE-FROM-R2
     Store key(s) 2 in WORK-AREA (identical to R1 tuple)
     Store WORK-AREA in the output relation
     Get next R2 tuple
     DO R2-END-CHECK
R2-END-CHECK
     IF end of R2
     Set keys to high-value
END-PROCESS
     Terminate program
```

FIGURE (4.4) Structured English of the Process Logic of the COMP Operator

```
UNION OPERATOR
     DO PARAMETERS-VALIDATION
     DO UNION-PROCESS
     DO END-PROCESS
PARAMETERS-VALIDATION
     IF parameters are correct
          DO-GET-A-TUPLE-FROM-THE-FIRST-RELATION
     ELSE (not correct)
          SO display "ERROR" message
          DO END-PROCESS
GET-A-TUPLE-FROM-THE-FIRST-RELATION
     Get a tuple from the first input relation (R1)
UNION-PROCESS
     REPEAT KEYS-COMPARISON UNTIL first relation key equal to high-
     value
KEYS-COMPARISON
     IF key 1 E key 2
          DO ERROR-ROUTINE
     ELSE
          DO-STORE-R1-TUPLE
ERROR-ROUTINE
     Display "EXIST" message
     DO GET-NEXT-R1-TUPLE
```

```
GET-NEXT-R1-TUPLE
     Get next R1 tuple
     DO R1-END-CHECK
R1-END-CHECK
     IF end of R1
          Set key 1 to high-value
STORE-R1-TUPLE
     Store R1 tuple in the main relation
     DO GET-NEXT-R1-TUPLE
END-PROCESS
     Terminate Program
```

FIGURE (4.5) Structured English of the Process Logic of the UNION Operator

```
MINUS OPERATOR
     DO PARAMETERS-VALIDATION
     DO MINUS-PROCESS
     DO END-PROCESS
PARAMETERS-VALIDATION
     IF parameters are correct
     DO-GET-A-TUPLE-FROM-THE-FIRST-RELATION
     ELSE (not correct)
          SO display "ERROR" message
          DO END-PROCESS
GET-A-TUPLE-FROM-THE-FIRST-RELATION
     Get a tuple from the first input relation (R1)
MINUS-PROCESS
     REPEAT KEYS-COMPARISON UNTIL first relation key equal to high-
          value
KEYS-COMPARISON
     IF key 1 E key 2
          DO GET-NEXT-R1-TUPLE
     ELSE
          DO ERROR-ROUTINE
GET-NEXT-R1-TUPLE
     DO GET-NEXT-R1-TUPLE
```

```
GET-NEXT-R1-TUPLE
     Get next R1 tuple
     DO R1-END-CHECK
R1-END-CHECK
     IF R1 end
          set key 1 to high value
ERROR-ROUTINE
     Display "NOT EXIST" message
     DO GET-NEXT-R1-TUPLE
END-PROCESS
     Terminate Program
```

FIGURE (4.6) Structured English of the Process Logic of the MINUS Operator

CHAPTER FIVE

RESULTS AND CONCLUSIONS

Introduction

This chapter will present sample results which have been obtained by running the model database.

The amount of data contained in the model system was kept as small as possible within the constraints that it should be possible to demonstrate all the essential features of the database performance.

The model system consists of 10 relations as shown in Figure 3.8. Three work files (WF1, WF2, WF3) were used to handle the required data during the operations.

Also, the system's database contains the menu file, the user's account file, the system log file, the relations description file and the listing operations detail file.

The model data which has been chosen for the database system is shown in Appendix (A).

The listing of the programs are given in Appendix (B). As an aid to readability most of the programs are commented upon.

5.1 Examples Showing the Use of the User Interface

At the log-in process time, many system activities occur. The full listing of the system activities at the log-in time are given as follows:

1. A request of the user account number (user code) is made and the following operations are done automatically:

a) checking the user code

b) checking the lock field (block mark field).

2. A request of the password is made and the following operations are done automatically.

   a) checking the password

   b) checking of the time limit.

3. A display of the last log-in date and the time is given.

At any stage of the checking, the system will display an appropriate message (for every erroneous action), stating the wrong action that has occurred. Figures 5.1 to 5.4 show examples of each of these erroneous attempts.

After the completion of the log-in process, the system will present the user with a personalized menu. This menu differs from one user to another and this depends entirely on the jobs allowed to each user which are recorded on the user's account file. Each displayed menu has different menu items for security purposes since the user is not allowed to see the menu items which are of no concern. Figures 5.5 and 5.7 show two different menus.

The user must select and input one of the choices in the displayed menu. Then the system will reject the wrong choice with a message displayed to the user. This is shown in Figure 5.8.

After the completion of each choice, the system will ask for another choice from the user.

The system will request a new password when the user wishes to leave the system and to change the previous password. Finally, the

system will display a message indicating the end of the functions.

Figure 5.9 illustrates these final activities.

Each legal and illegal action will be recorded on the log-file indicating the event type, time, date and the user code. This is shown in the log-file list in Figure 5.6.

The updating of the user's account record occurs at the same time and is detailed as follows:

a) Updating the last log-in date and time.
b) Updating the lock field if an erroneous action occurs.
c) Updating the number of the log-in field.
d) Updating the password if changed.

Figure 5.4 shows the user's account report which reflects the user's account file status.

5.2 Examples Showing the Use of the Menu

5.2.1 Output from the database administrator's dialogues

These dialogues can create and maintain the user's account file, the menu file and the relations description file. List 5.1 shows the menu displayed to the database administrator (security officer) once he has completed the authentication procedure. It illustrates the choices which are available to him which include the dialogues. The menu file maintenance dialogue assists the database administrator (DBA) to set up the menu elements and update the

records created during the creation phase. The List 5.1 illustrates the maintenance process which includes insertion, amendment, and deletion of a record from the menu file.

The relations file maintenance dialogue assists the DBA to create and maintain the relations description file records. List 5.2 illustrates the maintenance process of this file.

Finally List 5.3 illustrates the maintenance process of the user's account file by the user's account file maintenance dialogue.

This dialogue can assist the DBA to lock the system against any authorized user by setting the lock field to 2 or to re-open the system by re-setting the lock field to zero.

Also, the DBA can set up the elements of the jobs allowed for each user. Lists 5.4 and 5.5 show the user's account report which reflects the user's account file status before and after maintenance.

5.2.2 Listings of the security reports

The model database produces two formats of reports in order to assist the DBA or the security officer in the protection of the database against accidental or intentional disclosure to unauthorized persons, or unauthorized modifications or destruction. The List 5.6 illustrates the method of the selection and the production of these reports from the displayed menu to the DBA.

Since there is no direct use of the printer in the PRIME system, a file will be created with the same information. This file can then be spooled by the DBA on a printer. List 5.5 illustrates the USER ACCOUNT REPORT which reflects the status of each user record in the

user's account file. List 5.7 illustrates the INTERFACE REPORT which includes all the legal and illegal entries to the system. Each entry is commented on to indicate the type of each action.

5.2.3 Operational aspects of the database maintenance

The maintenance of the database includes updating tuples, deleting tuples and adding new tuples to the relation which can be completed by using the available operators. The operator UNION can be used to add new tuples to any relation and the operator MINUS can be used to delete any tuples from a relation.

The PRIME Editor utility was used to create a work file which included the new tuples which are required for operations such as add, union with the main relation. An error message was displayed for the existing tuples in the main relation. List 5.8 shows the complete UNION operation. The main relation was printed out before and after the UNION operation by using the RELATION LIST from the displayed menu.

The update operation can be achieved by a sequence of MINUS and UNION operations. Also, the PRIME EDITOR was used to change and correct the data. List 5.9 illustrates the update process in which the MINUS operation is included. The SELECT operator was used to extract the required tuple.

5.2.4 The production of the main lists

The user can enquire and produce the required list through the displayed menu. The three main types of book listing enquiries

require extra information which should be supplied by the user to select the particular books category. In the price list enquiry the user should supply the system with an exchange rate because it varies from time to time.

Unfortunately, there is no '£' sign in the PRIME system but the 'P' letter is used to indicate the '£' sign in the price list. In List 5.10, there is an illustration of the books list production according to the author(s) and category. List 5.11 shows the books list produced by the system. In List 5.12 the Price list production is shown according to the author(s) and the category. List 5.13 shows the price list produced for the same category chosen by the user. List 5.14 illustrates the title list production for the same category chosen for the previous lists and finally List 5.15 presents the title list produced by the system. The production of these lists was carried out by the relational operators and by the special subroutines which are stored in a special file called the listing operations detail file.

5.2.5 Examples showing the direct use of the relational operators

The direct use of the relational operator gives the user the ability to manipulate the database and to extract the required data. However, the user of these kind of sophisticated functions must know exactly how the database is organized.

The user should type in the required operators in detail. Example:

JOIN AUTHORTITLE AND WF1 OVER # BLURB-NO WF2

The final operation to be inputted must be the "END OF OPERATIONS" which informs the system to commence processing.

Any error detected by the system on these operations will be rejected with an appropriate message, stating the wrong part of the operation. Then the system will stop operations on the remaining operators. But the user can resume the operation after correcting the error and entering the corrected operator with the remaining operators. These activities are shown in Figure 5.16. If the operators require extra information (exchange rate, category, etc) then this will be requested from the user during the operation of each operator.

List 5.17 shows an example of these operations required to produce the price list of the new published books after 1979.

List 5.18 presents the price list produced by these operations.

If there is no "ERROR" message displayed to the user during the "Direct Use of the Relational Operators" and the message "no output record" displayed, this means that there is no information satisfying the user requirements. This is shown in List 5.19.

Finally, List 5.20 illustrates the selection of the quotation's tuples which satisfied the condition: "Book price greater than $5". The selected data were written on WF1 (temporarily work file one) and shown in the same list. Each selected tuple consists of two parts. The first part represents the ISBN (10 digit). The second part represents the book price (5 digit; 3 digit for the integer number and the other two digits for the fraction).

```
SEG #INTERFACE4384

 ************************************************************
 *     THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL     *
 *     THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM.  *
 ************************************************************
ENTER USER CODE:HYU678
ENTER USER CODE:KI864
*SORRY YOU ARE NOT AUTHORIZED TO USE THIS SYSTEM*


END OF THE FUNCTIONS

OK,
```

FIGURE 5.1 Entry of an unauthorized user

```
SEG #INTERFACE4384

 ************************************************************
 *     THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL     *
 *     THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM.  *
 ************************************************************
ENTER USER CODE:FMA111

*SORRY THE SYSTEM IS LOCKED*


END OF THE FUNCTIONS

OK,
```

FIGURE 5.2 The system is locked for this user

```
SEG #INTERFACE4384

 ************************************************************
 *     THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL     *
 *     THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM.  *
 ************************************************************
ENTER USER CODE:AHS259
ENTER PASSWORD:

*SORRY YOU CAN NOT USE THE SYSTEM AT THIS TIME*
YOUR TIME OF WORK BETWEEN 09 AND 10


END OF THE FUNCTIONS

OK,
```

FIGURE 5.3 The user is not allowed to work at this time

```
SEG #INTERFACE4384

*****************************************************************
*     THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL          *
*     THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM.      *
*****************************************************************
ENTER USER CODE:DJE201
ENTER PASSWORD:

INCORRECT PASSWORD
ENTER PASSWORD:

INCORRECT PASSWORD

*SORRY THE SYSTEM IS LOCKED*


END OF THE FUNCTIONS

OK,
```

FIGURE 5.4 Entry of incorrect password

```
SEG #INTERFACE4384

*****************************************************************
*     THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL          *
*     THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM.      *
*****************************************************************
ENTER USER CODE:JMD243
ENTER USER CODE:JMA243
ENTER PASSWORD:

17/06/1981    YOUR LAST ACCESS DATE

11:12:17      YOUR LAST ACCESS TIME
*****************************************************************
*                                                               *
*            COMPUTER  STUDIES  DEPARTMENT                      *
*       PROMOTIONAL LITERATURE PUBLISHING SYSTEM                *
*                                                               *
*                        MENU                                   *
*                        ----                                   *
*                                                               *
*                                                               *
* 01-MENU FILE MAINTENANCE DIALOGUE.                            *
* 02-RELATIONS FILE MAINTENANCE DIALOGUE.                       *
* 03-USERS ACCOUNT FILE MAINTENANCE DIALOGUE.                   *
* 04-USERS FILE STATUS REPORT.                                  *
* 05-INTERFACE REPORT(LEGAL AND ILLEGAL ENTRIES).               *
* 06-DATABASE MAINTENANCE.                                      *
* 07-BOOKS LIST ACCORDING TO AUTHOR AND CATEGORY.               *
* 08-PRICE LIST ACCORDING TO AUTHOR AND CATEGORY.               *
* 09-TITLE LIST ACCORDING TO AUTHOR AND CATEGORY.               *
* 10-DIRECT USE OF THE RELATIONAL OPERATORS.                    *
* 11-RELATION LIST(FIELDS :NAME, LENGTH, AND DATA).             *
*****************************************************************

ENTER CHOICE:      04
```

FIGURE 5.5 Entry of authorized user

```
    SLIST LOGG4384
ZZZZZZ8106171430111430311LLEGAL ATTEMPT
FMA1118106171431271431481HE SYSTEM IS LOCKED
AHS2598106171432591435241NOT ALLOWED IN THIS TIME
DJE2018106171436371436551NCORRECT PASSWORD
DJE2018106171436551436591NCORRECT PASSWORD
DJE2018106171436591436591HE SYSTEM IS LOCKED
JMA2438106171438231440111SERS FILE STATUS REPORT.

OK,
```

FIGURE 5.6 List of the log-file data after the entries

```
    SEG #INTERFACE4384

     ************************************************************
     *    THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL      *
     *    THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM.  *
     ************************************************************
ENTER USER CODE:LNB235
ENTER PASSWORD:

01/06/1981   YOUR LAST ACCESS DATE

12:58:48     YOUR LAST ACCESS TIME
     ************************************************************
     *                                                          *
     *            COMPUTER  STUDIES  DEPARTMENT                 *
     *       PROMOTIONAL LITERATURE PUBLISHING SYSTEM           *
     *                                                          *
     *                      MENU                                *
     *                      ----                                *
     *                                                          *
     *                                                          *
     * 01-MENU FILE MAINTENANCE DIALOGUE.                       *
     * 02-RELATIONS FILE MAINTENANCE DIALOGUE.                  *
     * 03-DATABASE MAINTENANCE.                                 *
     * 04-RELATION LIST(FIELDS :NAME,LENGTH,AND DATA).          *
     ************************************************************

ENTER CHOICE:     01
MENU FILE MAINTENANCE STARTED
     ********************************
     *   MENU  FILE  UPDATE         *
     *   ------------------         *
     *                              *
     * THE FUNCTIONS AVAILABLE ARE :*
     *                              *
     *   1-RECORD CREATION.         *
     *   2-RECORD DELETION.         *
     *   3-RECORD AMENDMENT.        *
     ********************************
ENTER CHOICE : 3
RECORD AMENDMENT ROUTINE:
-------------------------
```

FIGURE 5.7 Another menu for another user

```
SEG #INTERFACE4384

 ***************************************************************
 *     THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL        *
 *     THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM.    *
 ***************************************************************
ENTER USER CODE:JMA243
ENTER PASSWORD:

17/06/1981    YOUR LAST ACCESS DATE

14:38:23      YOUR LAST ACCESS TIME
      ***************************************************************
      *                                                             *
      *            COMPUTER  STUDIES  DEPARTMENT                    *
      *       PROMOTIONAL LITERATURE PUBLISHING SYSTEM              *
      *                                                             *
      *                        MENU                                 *
      *                        ----                                 *
      *                                                             *
      *                                                             *
      * 01-MENU FILE MAINTENANCE DIALOGUE.                          *
      * 02-RELATIONS FILE MAINTENANCE DIALOGUE.                     *
      * 03-USERS ACCOUNT FILE MAINTENANCE DIALOGUE.                 *
      * 04-USERS FILE STATUS REPORT.                                *
      * 05-INTERFACE REPORT(LEGAL AND ILLEGAL ENTRIES).             *
      * 06-DATABASE MAINTENANCE.                                    *
      * 07-BOOKS LIST ACCORDING TO AUTHOR AND CATEGORY.             *
      * 08-PRICE LIST ACCORDING TO AUTHOR AND CATEGORY.             *
      * 09-TITLE LIST ACCORDING TO AUTHOR AND CATEGORY.             *
      * 10-DIRECT USE OF THE RELATIONAL OPERATORS.                  *
      * 11-RELATION LIST(FIELDS :NAME, LENGTH, AND DATA).           *
      ***************************************************************

ENTER CHOICE:     A
INCORRECT CHOICE

ENTER CHOICE:     16
INCORRECT CHOICE

ENTER CHOICE:     03
USERS ACCOUNT  FILE UPDATE STARTED
      ********************************
      *    USERS  FILE  UPDATE       *
      *   ---------------------      *
      *                              *
      * THE FUNCTION AVAILABLE ARE : *
      *                              *
      *   1-RECORD CREATION.         *
      *   2-RECORD DELETION.         *
      *   3-RECORD AMENDMENT.        *
      ********************************
ENTER CHOICE:B
INCORRECT CHOICE
ENTER CHOICE:7
INCORRECT CHOICE
ENTER CHOICE:1
RECORD  CREATION  ROUTINE
-------------------------
```

FIGURE 5.8 The rejection of the incorrect choice

```
      ----------------------------------------------------------------------
      *     THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL               *
      *     THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM.           *
      **********************************************************************
ENTER USER CODE:LNB235
ENTER PASSWORD:

17/06/1981    YOUR LAST ACCESS DATE

14:42:08      YOUR LAST ACCESS TIME
      **********************************************************************
      *                                                                    *
      *              COMPUTER  STUDIES  DEPARTMENT                         *
      *       PROMOTIONAL LITERATURE PUBLISHING SYSTEM                     *
      *                                                                    *
      *                          MENU                                      *
      *                          ----                                      *
      *                                                                    *
      *                                                                    *
      * 01-MENU FILE MAINTENANCE DIALOGUE.                                 *
      * 02-RELATIONS FILE MAINTENANCE DIALOGUE.                            *
      * 03-DATABASE MAINTENANCE.                                           *
      * 04-RELATION LIST(FIELDS :NAME, LENGTH, AND DATA).                  *
      **********************************************************************

ENTER CHOICE:      04
ENTER MAX. 72 CHARACTERS OPERATION DETAIL
PRINT PAGEYEAR
ENTER MAX. 8 CH. FILE NAME
PAGE4384
0677107502000024068

0677118007001061067

0677118007002139067

0677118007003100467

067711800703V000000

0677124309001050281

0677124309002033881

0677124309003029281

067712430903V000000

0677138308100025269

0677138308200091069

0677158904000020475

0677507305000030480

END OF RELATION LIST
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  NO
DO YOU WANT TO CHANGE YOUR PASSWORD?
ENTER YES OR NO!  YES
ENTER NEW PASSWORD:


END OF THE FUNCTIONS
```

FIGURE 5.9 The request for the new password

LIST 5.1

```
     ************************************************************
     *     THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL     *
     *     THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM. *
     ************************************************************
ENTER USER CODE:JMA243
ENTER PASSWORD:

17/06/1981    YOUR LAST ACCESS DATE

14:45:48      YOUR LAST ACCESS TIME
     ************************************************************
     *                                                          *
     *            COMPUTER  STUDIES  DEPARTMENT                 *
     *       PROMOTIONAL LITERATURE PUBLISHING SYSTEM           *
     *                                                          *
     *                        MENU                              *
     *                        ----                              *
     *                                                          *
     *                                                          *
     * 01-MENU FILE MAINTENANCE DIALOGUE.                       *
     * 02-RELATIONS FILE MAINTENANCE DIALOGUE.                  *
     * 03-USERS ACCOUNT FILE MAINTENANCE DIALOGUE.              *
     * 04-USERS FILE STATUS REPORT.                             *
     * 05-INTERFACE REPORT(LEGAL AND ILLEGAL ENTRIES).          *
     * 06-DATABASE MAINTENANCE.                                 *
     * 07-BOOKS LIST ACCORDING TO AUTHOR AND CATEGORY.          *
     * 08-PRICE LIST ACCORDING TO AUTHOR AND CATEGORY.          *
     * 09-TITLE LIST ACCORDING TO AUTHOR AND CATEGORY.          *
     * 10-DIRECT USE OF THE RELATIONAL OPERATORS.               *
     * 11-RELATION LIST(FIELDS :NAME, LENGTH, AND DATA).        *
     ************************************************************

ENTER CHOICE:     01
MENU FILE MAINTENANCE STARTED
     ************************************
     *   MENU  FILE  UPDATE             *
     *   ------------------             *
     *                                  *
     * THE FUNCTIONS AVAILABLE ARE :    *
     *                                  *
     *   1-RECORD CREATION.             *
     *   2-RECORD DELETION.             *
     *   3-RECORD AMENDMENT.            *
     ************************************
ENTER CHOICE : 2
RECORD DELETION ROUTINE:
------------------------
ENTER 2 NUMERIC CHARACTERS  APPLICATION CODE : 16
APPLICATION CODE IS NOT EXIST, TRY TO CORRECT IT
DO YOU WANT TO DELETE ANOTHER RECORD?
ENTER YES OR NO:YES
ENTER 2 NUMERIC CHARACTERS  APPLICATION CODE : 04
RECORD DELETED IS:
04USERS FILE STATUS REPORT.
DO YOU WANT TO DELETE ANOTHER RECORD?
ENTER YES OR NO:NO
DO YOU WANT MORE UPDATE?
ENTER YES OR NO:YES
ENTER CHOICE : 3
RECORD AMENDMENT ROUTINE:
-------------------------
```

```
ENTER 2 NUMERIC CHARACTERS  APPLICATION CODE : 30
APPLICATION CODE IS NOT EXIST, TRY TO CORRECT IT
DO YOU WANT TO AMEND ANOTHER RECORD?
ENTER YES OR NO:YES
ENTER 2 NUMERIC CHARACTERS  APPLICATION CODE : 11
ENTER MAX. 46 CHARACTERS APPLICATION NAME :
RELATION DATA LIST.
RECORD AMENDED IS:
11RELATION DATA LIST.
DO YOU WANT TO AMEND ANOTHER RECORD?
ENTER YES OR NO:NO
DO YOU WANT MORE UPDATE?
ENTER YES OR NO:YES
ENTER CHOICE : 1
RECORD CREATION ROUTINE:
------------------------
ENTER 2 NUMERIC CHARACTERS  APPLICATION CODE : 01
APPLICATION CODE EXIST, TRY TO CORRECT IT
DO YOU WANT TO CREATE ANOTHER RECORD?
ENTER YES OR NO:YES
ENTER 2 NUMERIC CHARACTERS  APPLICATION CODE : 04
ENTER MAX. 46 CHARACTERS APPLICATION NAME :
USERS ACCOUNT REPORT.
RECORD WRITTEN IS:
04USERS ACCOUNT REPORT.
DO YOU WANT TO CREATE ANOTHER RECORD?
ENTER YES OR NO:NO
DO YOU WANT MORE UPDATE?
ENTER YES OR NO:NO
MENU FILE UPDATE FINISHED
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  YES

ENTER CHOICE:     10
BEWARE PLEASE:
 THE SORTN(AUTHOR'NAME SORT ALPHABETICALLY)
 SUBROUTINE, USE WF1 AS INPUT AND WF2 AS OUTPUT

 THE SORTT(TITLE SORT ALPHABETICALLY)
 SUBROUTINE,USE WF3 AS INPUT AND WF1 AS OUTPUT
**********************************************
ENTER MAX. 72 CHARACTERS OPERATION DETAIL:
PO K
ENTER MAX. 72 CHARACTERS OPERATION DETAIL:
END OF
OPERATOR NAME ERROR, YOU CAN NOT CONTINUE
PO K
END OF DIRECT USE OF THE OPERATIONS
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  NO
DO YOU WANT TO CHANGE YOUR PASSWORD?
ENTER YES OR NO!  NO


END OF THE FUNCTIONS

OK,
```

LIST 5.2

```
     SEG #INTERFACE4384

      ************************************************************
      *     THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL     *
      *     THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM. *
      ************************************************************
ENTER USER CODE:JMA243
ENTER PASSWORD:

17/06/1981    YOUR LAST ACCESS DATE

14:57:57      YOUR LAST ACCESS TIME
      ************************************************************
      *                                                          *
      *            COMPUTER  STUDIES  DEPARTMENT                 *
      *       PROMOTIONAL LITERATURE PUBLISHING SYSTEM           *
      *                                                          *
      *                        MENU                              *
      *                        ----                              *
      *                                                          *
      *                                                          *
      * 01-MENU FILE MAINTENANCE DIALOGUE.                       *
      * 02-RELATIONS FILE MAINTENANCE DIALOGUE.                  *
      * 03-USERS ACCOUNT FILE MAINTENANCE DIALOGUE.              *
      * 04-USERS ACCOUNT REPORT.                                 *
      * 05-INTERFACE REPORT(LEGAL AND ILLEGAL ENTRIES).          *
      * 06-DATABASE MAINTENANCE.                                 *
      * 07-BOOKS LIST ACCORDING TO AUTHOR AND CATEGORY.          *
      * 08-PRICE LIST ACCORDING TO AUTHOR AND CATEGORY.          *
      * 09-TITLE LIST ACCORDING TO AUTHOR AND CATEGORY.          *
      * 10-DIRECT USE OF THE RELATIONAL OPERATORS.               *
      * 11-RELATION DATA LIST.                                   *
      ************************************************************

ENTER CHOICE:     02
MAINTENANCE OF THE FILE IS STARTED
      ***********************************
      *      FILE  MAINTENANCE          *
      *      -----------------          *
      *                                 *
      * THE FUNCTIONS AVAILABLE ARE:    *
      *                                 *
      *   1-RECORD CREATION.            *
      *   2-RECORD DELETION.            *
      *   3-RECORD AMENDMENT.           *
      ***********************************
ENTER CHOICE : 1
RECORD CREATION ROUTINE:
------------------------
ENTER MAX.  12  CHARACTERS RELATION  NAME :
WF1
RELATION EXIST IN THE FILE
DO YOU WANT TO CREATE ANOTHER RECORD?
ENTER YES OR NO:YES
RECORD CREATION ROUTINE:
------------------------
ENTER MAX.  12  CHARACTERS RELATION  NAME :
WF5
ENTER  2  NUMERIC CHARACTERS  RELATION  CODE :
15
ENTER MAX. 15 FIELDS FOR EACH RELATION
ENTER MAX. 12 CHARACTERS FIELD NAME :
#ISBN
ENTER MAX  999 FIELD LENGTH :
```

Continue List 5.2

```
010
ANY MORE ENTRY!
ENTER YES OR NO:YES
ENTER MAX. 12 CHARACTERS FIELD NAME :
#BLURB-NO
ENTER MAX. 999 FIELD LENGTH :
010
ANY MORE ENTRY!
ENTER YES OR NO:NO
RECORD WRITTEN IS:
WF5          15#ISBN        001010#BLURB-NO    011010

DO YOU WANT TO CREATE ANOTHER RECORD?
ENTER YES OR NO:NO
DO YOU WANT MORE UPDATE?
ENTER YES OR NO:YES
ENTER CHOICE : 3
RECORD AMENDMENT ROUTINE:
-------------------------
ENTER MAX.  12  CHARACTERS RELATION  NAME :
WF1
ENTER MAX. 15 FIELDS FOR EACH RELATION
ENTER MAX. 12 CHARACTERS FIELD NAME :
#BLURB-NO
ENTER MAX. 999 FIELD LENGTH :
010
ANY MORE ENTRY!
ENTER YES OR NO:YES
ENTER MAX. 12 CHARACTERS FIELD NAME :
#PART-VOLUME
ENTER MAX. 999 FIELD LENGTH :
003
ANY MORE ENTRY!
ENTER YES OR NO:YES
ENTER MAX. 12 CHARACTERS FIELD NAME :
DESCRIPTION
ENTER MAX. 999 FIELD LENGTH :
200
ANY MORE ENTRY!
ENTER YES OR NO:YES
ENTER MAX. 12 CHARACTERS FIELD NAME :
PUB-YEAR
ENTER MAX. 999 FIELD LENGTH :
002
ANY MORE ENTRY!
ENTER YES OR NO:NO
RELATION REWRITTEN IS:
WF1          11#BLURB-NO    001010#PART-VOLUME011003DESCRIPTION 014
214002
DO YOU WANT TO AMEND ANOTHER RECORD?
ENTER YES OR NO:YES
```

Continue List 5.2

```
RECORD AMENDMENT ROUTINE:
-------------------------
ENTER MAX.  12  CHARACTERS RELATION  NAME :
WF8
RELATION DOES NOT EXIST IN THE FILE
DO YOU WANT TO AMEND ANOTHER RECORD?
ENTER YES OR NO:NO
DO YOU WANT MORE UPDATE?
ENTER YES OR NO:YES
ENTER CHOICE : 2
RECORD DELETION ROUTINE:
------------------------
ENTER MAX.  12  CHARACTERS RELATION  NAME :
WF4
RECORD IS NOT EXIST IN THE FILE
DO YOU WANT TO DELETE ANOTHER RECORD?
ENTER YES OR NO:YES
RECORD DELETION ROUTINE:
------------------------
ENTER MAX.  12  CHARACTERS RELATION  NAME :
WF5
RECORD DELETED IS:
WF5          15#ISBN        001010#BLURB-NO    011010

DO YOU WANT TO DELETE ANOTHER RECORD?
ENTER YES OR NO:NO
DO YOU WANT MORE UPDATE?
ENTER YES OR NO:NO
RELATIONS DESCRIPTION FILE UPDATE FINISHED
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  NO
DO YOU WANT TO CHANGE YOUR PASSWORD?
ENTER YES OR NO!  NO


END OF THE FUNCTIONS

OK.
```

LIST 5.3

```
     SEG #INTERFACE4384

     ************************************************************
     *     THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL     *
     *     THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM. *
     ************************************************************
ENTER USER CODE:JMA243
ENTER PASSWORD:

17/06/1981    YOUR LAST ACCESS DATE

15:00:23      YOUR LAST ACCESS TIME
     ************************************************************
     *                                                          *
     *            COMPUTER  STUDIES  DEPARTMENT                 *
     *       PROMOTIONAL LITERATURE PUBLISHING SYSTEM           *
     *                                                          *
     *                        MENU                              *
     *                        ----                              *
     *                                                          *
     *                                                          *
     * 01-MENU FILE MAINTENANCE DIALOGUE.                       *
     * 02-RELATIONS FILE MAINTENANCE DIALOGUE.                  *
     * 03-USERS ACCOUNT FILE MAINTENANCE DIALOGUE.              *
     * 04-USERS ACCOUNT REPORT.                                 *
     * 05-INTERFACE REPORT(LEGAL AND ILLEGAL ENTRIES).          *
     * 06-DATABASE MAINTENANCE.                                 *
     * 07-BOOKS LIST ACCORDING TO AUTHOR AND CATEGORY.          *
     * 08-PRICE LIST ACCORDING TO AUTHOR AND CATEGORY.          *
     * 09-TITLE LIST ACCORDING TO AUTHOR AND CATEGORY.          *
     * 10-DIRECT USE OF THE RELATIONAL OPERATORS.               *
     * 11-RELATION DATA LIST.                                   *
     ************************************************************

ENTER CHOICE:     03
USERS ACCOUNT  FILE UPDATE STARTED
     ************************************
     *    USERS  FILE  UPDATE           *
     *   ---------------------          *
     *                                   *
     * THE FUNCTION AVAILABLE ARE :      *
     *                                   *
     *   1-RECORD CREATION.              *
     *   2-RECORD DELETION.              *
     *   3-RECORD AMENDMENT.             *
     ************************************
ENTER CHOICE:2
RECORD DELETION ROUTINE
-----------------------
ENTER SIX CHARACTERS USER CODE:FGD453
USER CODE DOES NOT EXIST, TRY TO CORRECT USER CODE
DO YOU WANT TO DELETE ANOTHER RECORD?
ENTER YES OR NO : YES
RECORD DELETION ROUTINE
-----------------------
ENTER SIX CHARACTERS USER CODE:HMY301
RECORD DELETED IS:
HMY301    03012141540408106O2
MR.    H.  M.  YOUSIF        07080911
DO YOU WANT TO DELETE ANOTHER RECORD?
ENTER YES OR NO : NO
DO YOU WANT MORE UPDATE?
ENTER YES OR NO : YES
```

Continue List 5.3

```
ENTER CHOICE:1
RECORD  CREATION  ROUTINE
------------------------
ENTER SIX CHARACTERS USER CODE:MJA555
ENTER 2 CHARATERS START TIME : 10
ENTER 2 CHARATERS  END  TIME : 20
ENTER MAX. 25 CHAR. USER NAME: MR.    M. J. ABDUL-JABBAR
ENTER MAX. 28 CHARS APPLICATION CODE 2 CHAR. EACH:040511
RECORD CREATED IS:
MJA555     0001020000000000000
MR.    M. J. ABDUL-JABBAR 040511
DO YOU WANT TO CREATE ANOTHER RECORD?
ENTER YES OR NO : YES
RECORD  CREATION  ROUTINE
------------------------
ENTER SIX CHARACTERS USER CODE:FMA111
USER RECORD EXIST, TRY TO CORRECT THE USER CODE
OR ENTER ANOTHER USER CODE
DO YOU WANT TO CREATE ANOTHER RECORD?
ENTER YES OR NO : NO
DO YOU WANT MORE UPDATE?
ENTER YES OR NO : YES
ENTER CHOICE:3
RECORD AMENDMENT ROUTINE
------------------------
ENTER SIX CHARACTERS USER CODE:DJE201
DO YOU WANT TO AMEND THE PASSWORD?
ENTER YES OR NO : YES
DO YOU WANT TO AMEND ANOTHER ITEM?
ENTER YES OR NO : YES
DO YOU WANT TO AMEND THE BLACK MARK?
ENTER YES OR NO : YES
ENTER 1 CHARACTER BLACKMARK     0
DO YOU WANT TO AMEND ANOTHER ITEM?
ENTER YES OR NO : NO
RECORD AMENDED IS:
DJE201     0310924143659810617
PROF. D. J. EVANS          0102030405061 1
DO YOU WANT TO AMEND ANOTHER RECORD?
ENTER YES OR NO : YES
RECORD AMENDMENT ROUTINE
------------------------
ENTER SIX CHARACTERS USER CODE:FMA111
DO YOU WANT TO AMEND THE PASSWORD?
ENTER YES OR NO : YES
DO YOU WANT TO AMEND ANOTHER ITEM?
ENTER YES OR NO : YES
DO YOU WANT TO AMEND THE BLACK MARK?
ENTER YES OR NO : YES
ENTER 1 CHARACTER BLACKMARK     0
DO YOU WANT TO AMEND ANOTHER ITEM?
ENTER YES OR NO : YES
DO YOU WANT TO AMEND THE NO OF ACCESS?
ENTER YES OR NO : YES
ENTER  2  CHARACTERS  NO OF LOGOS
DO YOU WANT TO AMEND ANOTHER ITEM?
ENTER YES OR NO : YES
DO YOU WANT TO AMEND THE TIME ALLOWED?
ENTER YES OR NO : NO
DO YOU WANT TO AMEND THE USER NAME?
ENTER YES OR NO : NO
DO YOU WANT TO AMEND THE JOBS CODES?
```

Continue List 5.3

```
ENTER YES OR NO : YES
ENTER MAX. 28 CHARS APPLICATION CODE 2 CHAR. EACH:040507080911
RECORD AMENDED IS:
FMA111     0051020143127810617
MR.    F. M. ABDUL-JABBAR 040507080911
DO YOU WANT TO AMEND ANOTHER RECORD?
ENTER YES OR NO : NO
DO YOU WANT MORE UPDATE?
ENTER YES OR NO : NO
USERS ACCOUNT UPDATE FINISHED
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  YES

ENTER CHOICE:      04
USERS ACCOUNT REPORT STARTED
USERS ACCOUNT REPORT FINISHED
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  NO
DO YOU WANT TO CHANGE YOUR PASSWORD?
ENTER YES OR NO!  NO

PLEASE ENTER: SPOOL UREP4384 -X1 (AFTER OK)
END OF THE FUNCTIONS

OK, SPOOL UREP4384 -X1

OK,
```

LISTING OF UREP4384 14:40 17 JUN 81

LIST 5.4

```
                                * USERS  ACCOUNT  REPORT *
**********************************************************************
**                                                                  **
**USER    BLACK  NO. OF      TIME ALLOWED              LAST  ACCESS  **
**NUMBER  MARK   LOGIN   START TIME  END TIME      TIME        DATE    **
**------  -----  ------  ----------  --------    ----------  ----------**
**AHS259   1      24        09          10       14:32:59    17/06/1981**
**                                                                  **
**USER NAME                 APPLICATION CODE AND NAME               **
**------------------------  ------------------------------------------**
**MR.   A. H. SALIH        04  USERS FILE STATUS REPORT.              **
**                          07  BOOKS LIST ACCORDING TO AUTHOR AND CATEGORY.**
**                          08  PRICE LIST ACCORDING TO AUTHOR AND CATEGORY.**
**                          09  TITLE LIST ACCORDING TO AUTHOR AND CATEGORY.**
**                          11  RELATION LIST(FIELDS :NAME,LENGTH,AND DATA).**
**********************************************************************
**                                                                  **
**USER    BLACK  NO. OF      TIME ALLOWED              LAST  ACCESS  **
**NUMBER  MARK   LOGIN   START TIME  END TIME      TIME        DATE    **
**------  -----  ------  ----------  --------    ----------  ----------**
**DJE201    2      31        09          24       14:36:59    17/06/1981**
**                                                                  **
**USER NAME                 APPLICATION CODE AND NAME               **
**------------------------  ------------------------------------------**
**PROF. D. J. EVANS         01  MENU FILE MAINTENANCE DIALOGUE.       **
**                          02  RELATIONS FILE MAINTENANCE DIALOGUE.  **
**                          03  USERS ACCOUNT FILE MAINTENANCE DIALOGUE.**
**                          04  USERS FILE STATUS REPORT.            **
**                          05  INTERFACE REPORT(LEGAL AND ILLEGAL ENTRIES).**
**                          06  DATABASE MAINTENANCE.                **
**                          11  RELATION LIST(FIELDS :NAME,LENGTH,AND DATA).**
**********************************************************************
**                                                                  **
**USER    BLACK  NO. OF      TIME ALLOWED              LAST  ACCESS  **
**NUMBER  MARK   LOGIN   START TIME  END TIME      TIME        DATE    **
**------  -----  ------  ----------  --------    ----------  ----------**
**FMA111    2      15        10          20       14:31:27    17/06/1981**
**                                                                  **
**USER NAME                 APPLICATION CODE AND NAME               **
**------------------------  ------------------------------------------**
**MR.   F. M. ABDUL-JABBAR  04  USERS FILE STATUS REPORT.            **
**                          07  BOOKS LIST ACCORDING TO AUTHOR AND CATEGORY.**
**                          08  PRICE LIST ACCORDING TO AUTHOR AND CATEGORY.**
**                          09  TITLE LIST ACCORDING TO AUTHOR AND CATEGORY.**
**********************************************************************
**                                                                  **
```

```
**USER    BLACK  NO. OF      TIME ALLOWED              LAST  ACCESS  **
**NUMBER  MARK   LOGIN   START TIME  END TIME      TIME        DATE    **
**------  -----  ------  ----------  --------    ----------  ----------**
**HMY301    0      30        12          14       15:40:40    02/06/1981**
**                                                                  **
**USER NAME                 APPLICATION CODE AND NAME               **
**------------------------  ------------------------------------------**
**MR.   H. M. YOUSIF       07  BOOKS LIST ACCORDING TO AUTHOR AND CATEGORY.**
**                          08  PRICE LIST ACCORDING TO AUTHOR AND CATEGORY.**
**                          09  TITLE LIST ACCORDING TO AUTHOR AND CATEGORY.**
**                          11  RELATION LIST(FIELDS :NAME,LENGTH,AND DATA).**
**********************************************************************
**                                                                  **
**USER    BLACK  NO. OF      TIME ALLOWED              LAST  ACCESS  **
**NUMBER  MARK   LOGIN   START TIME  END TIME      TIME        DATE    **
**------  -----  ------  ----------  --------    ----------  ----------**
**JMA243    0      27        09          24       11:12:17    17/06/1981**
**                                                                  **
**USER NAME                 APPLICATION CODE AND NAME               **
**------------------------  ------------------------------------------**
**MR.   J. M. ABDUL-JABBAR  01  MENU FILE MAINTENANCE DIALOGUE.       **
**                          02  RELATIONS FILE MAINTENANCE DIALOGUE.  **
**                          03  USERS ACCOUNT FILE MAINTENANCE DIALOGUE.**
**                          04  USERS FILE STATUS REPORT.            **
**                          05  INTERFACE REPORT(LEGAL AND ILLEGAL ENTRIES).**
**                          06  DATABASE MAINTENANCE.                **
**                          07  BOOKS LIST ACCORDING TO AUTHOR AND CATEGORY.**
**                          08  PRICE LIST ACCORDING TO AUTHOR AND CATEGORY.**
**                          09  TITLE LIST ACCORDING TO AUTHOR AND CATEGORY.**
**                          10  DIRECT USE OF THE RELATIONAL OPERATORS.**
**                          11  RELATION LIST(FIELDS :NAME,LENGTH,AND DATA).**
**********************************************************************
**                                                                  **
**USER    BLACK  NO. OF      TIME ALLOWED              LAST  ACCESS  **
**NUMBER  MARK   LOGIN   START TIME  END TIME      TIME        DATE    **
**------  -----  ------  ----------  --------    ----------  ----------**
**LNB235    0      01        10          21       12:58:48    01/06/1981**
**                                                                  **
**USER NAME                 APPLICATION CODE AND NAME               **
**------------------------  ------------------------------------------**
**MR.   L. N. BLACKBURN     01  MENU FILE MAINTENANCE DIALOGUE.       **
**                          02  RELATIONS FILE MAINTENANCE DIALOGUE.  **
**                          06  DATABASE MAINTENANCE.                **
```

LIST 5.5

```
                                * USERS  ACCOUNT  REPORT *
**********************************************************************************
**                                                                              **
**USER     BLACK  NO. OF      TIME ALLOWED                  LAST  ACCESS       **
**NUMBER   MARK   LOGIN   START TIME  END TIME       TIME         DATE          **
**-------  -----  ------  ----------  --------      ----------  ----------      **
**AHS259     1      24        09          10        14:32:59    17/06/1981      **
**                                                                              **
**USER NAME                  APPLICATION CODE AND NAME                         **
**--------------------------  ------------------------------------------------  **
**MR.   A. H. SALIH          04  USERS ACCOUNT REPORT.                           **
**                           07  BOOKS LIST ACCORDING TO AUTHOR AND CATEGORY.    **
**                           08  PRICE LIST ACCORDING TO AUTHOR AND CATEGORY.    **
**                           09  TITLE LIST ACCORDING TO AUTHOR AND CATEGORY.    **
**                           11  RELATION DATA LIST.                             **
**********************************************************************************
**                                                                              **
**USER     BLACK  NO. OF      TIME ALLOWED                  LAST  ACCESS       **
**NUMBER   MARK   LOGIN   START TIME  END TIME       TIME         DATE          **
**-------  -----  ------  ----------  --------      ----------  ----------      **
**DJE201     0      31        09          24        14:36:59    17/06/1981      **
**                                                                              **
**USER NAME                  APPLICATION CODE AND NAME                         **
**--------------------------  ------------------------------------------------  **
**PROF. D. J. EVANS          01  MENU FILE MAINTENANCE DIALOGUE.                 **
**                           02  RELATIONS FILE MAINTENANCE DIALOGUE.            **
**                           03  USERS ACCOUNT FILE MAINTENANCE DIALOGUE.        **
**                           04  USERS ACCOUNT REPORT.                           **
**                           05  INTERFACE REPORT(LEGAL AND ILLEGAL ENTRIES).    **
**                           06  DATABASE MAINTENANCE.                           **
**                           11  RELATION DATA LIST.                             **
**********************************************************************************
**                                                                              **
**USER     BLACK  NO. OF      TIME ALLOWED                  LAST  ACCESS       **
**NUMBER   MARK   LOGIN   START TIME  END TIME       TIME         DATE          **
**-------  -----  ------  ----------  --------      ----------  ----------      **
**FMA111     0      05        10          20        14:31:27    17/06/1981      **
**                                                                              **
**USER NAME                  APPLICATION CODE AND NAME                         **
**--------------------------  ------------------------------------------------  **
**MR.   F. M. ABDUL-JABBAR   04  USERS ACCOUNT REPORT.                           **
**                           05  INTERFACE REPORT(LEGAL AND ILLEGAL ENTRIES).    **
**                           07  BOOKS LIST ACCORDING TO AUTHOR AND CATEGORY.    **
**                           08  PRICE LIST ACCORDING TO AUTHOR AND CATEGORY.    **
**                           09  TITLE LIST ACCORDING TO AUTHOR AND CATEGORY.    **
**                           11  RELATION DATA LIST.                             **
**********************************************************************************
**                                                                              **
**USER     BLACK  NO. OF      TIME ALLOWED                  LAST  ACCESS       **
**NUMBER   MARK   LOGIN   START TIME  END TIME       TIME         DATE          **
**-------  -----  ------  ----------  --------      ----------  ----------      **
**JMA243     0      31        09          24        15:09:06    17/06/1981      **
**                                                                              **
**USER NAME                  APPLICATION CODE AND NAME                         **
**--------------------------  ------------------------------------------------  **
**MR.   J. M. ABDUL-JABBAR   01  MENU FILE MAINTENANCE DIALOGUE.                 **
**                           02  RELATIONS FILE MAINTENANCE DIALOGUE.            **
**                           03  USERS ACCOUNT FILE MAINTENANCE DIALOGUE.        **
**                           04  USERS ACCOUNT REPORT.                           **
**                           05  INTERFACE REPORT(LEGAL AND ILLEGAL ENTRIES).    **
**                           06  DATABASE MAINTENANCE.                           **
**                           07  BOOKS LIST ACCORDING TO AUTHOR AND CATEGORY.    **
**                           08  PRICE LIST ACCORDING TO AUTHOR AND CATEGORY.    **
**                           09  TITLE LIST ACCORDING TO AUTHOR AND CATEGORY.    **
**                           10  DIRECT USE OF THE RELATIONAL OPERATORS.         **
**                           11  RELATION DATA LIST.                             **
**********************************************************************************
**                                                                              **
**USER     BLACK  NO. OF      TIME ALLOWED                  LAST  ACCESS       **
**NUMBER   MARK   LOGIN   START TIME  END TIME       TIME         DATE          **
**-------  -----  ------  ----------  --------      ----------  ----------      **
**LNB235     0      03        10          21        14:49:22    17/06/1981      **
**                                                                              **
**USER NAME                  APPLICATION CODE AND NAME                         **
**--------------------------  ------------------------------------------------  **
**MR.   L. N. BLACKBURN      01  MENU FILE MAINTENANCE DIALOGUE.                 **
**                           02  RELATIONS FILE MAINTENANCE DIALOGUE.            **
**                           06  DATABASE MAINTENANCE.                           **
**                           11  RELATION DATA LIST.                             **
**********************************************************************************
**                                                                              **
**USER     BLACK  NO. OF      TIME ALLOWED                  LAST  ACCESS       **
**NUMBER   MARK   LOGIN   START TIME  END TIME       TIME         DATE          **
**-------  -----  ------  ----------  --------      ----------  ----------      **
**MJA555     0      00        10          20        00:00:00    00/00/1900      **
**                                                                              **
**USER NAME                  APPLICATION CODE AND NAME                         **
**--------------------------  ------------------------------------------------  **
**MR.   M. J. ABDUL-JABBAR   04  USERS ACCOUNT REPORT.                           **
```

LIST 5.6

```
      SEG #INTERFACE4384

       ************************************************************
       *     THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL     *
       *     THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM. *
       ************************************************************
ENTER USER CODE:FMA111
ENTER PASSWORD:

17/06/1981    YOUR LAST ACCESS DATE

14:31:27      YOUR LAST ACCESS TIME
       ************************************************************
       *                                                          *
       *              COMPUTER  STUDIES  DEPARTMENT               *
       *       PROMOTIONAL LITERATURE PUBLISHING SYSTEM           *
       *                                                          *
       *                        MENU                              *
       *                        ----                              *
       *                                                          *
       *                                                          *
       * 01-USERS ACCOUNT REPORT.                                 *
       * 02-INTERFACE REPORT(LEGAL AND ILLEGAL ENTRIES).          *
       * 03-BOOKS LIST ACCORDING TO AUTHOR AND CATEGORY.          *
       * 04-PRICE LIST ACCORDING TO AUTHOR AND CATEGORY.          *
       * 05-TITLE LIST ACCORDING TO AUTHOR AND CATEGORY.          *
       * 06-RELATION DATA LIST.                                   *
       ************************************************************

ENTER CHOICE:      02
SORT STARTED ON LOGGING FILE
LOGGING FILE SORT FINISHED
INTERFACE REPORT STARTED
ZZZZZZ USER NUMBER NOT KNOWN
INTERFACE REPORT FINISHED
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  NO
DO YOU WANT TO CHANGE YOUR PASSWORD?
ENTER YES OR NO!  NO

PLEASE ENTER: SPOOL IREP4384 -X1 (AFTER OK)
END OF THE FUNCTIONS

OK, SPOOL IREP4384 -X1

OK,
```

LIST 5.7

```
LISTING OF IREP4784  15:09  17 JUN 81

                                ** INTERFACE REPORT **
**********************************************************************
**                                                                  **
**USER                                                              **
**NUMBER  USER NAME                                                 **
**------  --------------------------                                **
**AHS259  MR.   A. H. SALIH                                         **
**                                                                  **
**ACCESS DATE  START TIME END TIME  COMMENT                         **
**------ ----  ---------- --------  ------------------------------  **
**17/06/1981   14:32:59   14:35:24  NOT ALLOWED IN THIS TIME         **
**********************************************************************
**                                                                  **
**USER                                                              **
**NUMBER  USER NAME                                                 **
**------  ---------------------------                               **
**DJE201  PROF. D. J. EVANS                                         **
**                                                                  **
**ACCESS DATE  START TIME END TIME  COMMENT                         **
**------ ----  ---------- --------  ------------------------------  **
**17/06/1981   14:36:37   14:36:55  INCORRECT PASSWORD               **
**17/06/1981   14:36:55   14:36:59  INCORRECT PASSWORD               **
**17/06/1981   14:36:59   14:36:59  THE SYSTEM IS LOCKED             **
**********************************************************************
**                                                                  **
**USER                                                              **
**NUMBER  USER NAME                                                 **
**------  -------------------------                                 **
**FMA111  MR.   F. M. ABDUL-JABBAR                                  **
**                                                                  **
**ACCESS DATE  START TIME END TIME  COMMENT                         **
**------ ----  ---------- --------  ------------------------------  **
**17/06/1981   14:31:27   14:31:48  THE SYSTEM IS LOCKED             **
**********************************************************************
**                                                                  **
**USER                                                              **
**NUMBER  USER NAME                                                 **
**------  -------------------------                                 **
**JMA243  MR.   J. M. ABDUL-JABBAR                                  **
**                                                                  **
**ACCESS DATE  START TIME END TIME  COMMENT                         **
**------ ----  ---------- --------  ------------------------------  **
**17/06/1981   14:38:23   14:40:11  USERS FILE STATUS REPORT.        **
**17/06/1981   14:43:48   14:48:10  USERS ACCOUNT FILE MAINTENANCE DIALOGUE.  **
**17/06/1981   14:53:05   14:57:51  MENU FILE MAINTENANCE DIALOGUE.  **
**17/06/1981   14:57:57   14:59:15  DIRECT USE OF THE RELATIONAL OPERATORS.  **



**17/06/1981   15:00:23   15:07:13  RELATIONS FILE MAINTENANCE DIALOGUE.  **
**17/06/1981   15:09:06   15:16:04  USERS ACCOUNT FILE MAINTENANCE DIALOGUE.  **
**17/06/1981   15:16:17   15:16:43  USERS ACCOUNT REPORT.            **
**********************************************************************
**                                                                  **
**USER                                                              **
**NUMBER  USER NAME                                                 **
**------  -------------------------                                 **
**LNB235  MR.   L. N. BLACKBURN                                     **
**                                                                  **
**ACCESS DATE  START TIME END TIME  COMMENT                         **
**------ ----  ---------- --------  ------------------------------  **
**17/06/1981   14:42:08   14:44:34  MENU FILE MAINTENANCE DIALOGUE.  **
**17/06/1981   14:49:22   14:51:40  RELATION LIST(FIELDS :NAME,LENGTH,AND DATA).  **
**********************************************************************
**                                                                  **
**USER                                                              **
**NUMBER  USER NAME                                                 **
**------  -------------------------                                 **
**        NOT KNOWN                                                 **
**                                                                  **
**ACCESS DATE  START TIME END TIME  COMMENT                         **
**------ ----  ---------- --------  ------------------------------  **
**17/06/1981   14:30:11   14:30:31  ILLEGAL ATTEMPT                  **
**********************************************************************
```

LIST 5.8

```
    ED
INPUT
067711800716
067715890411
067750730507
067760880402
068822334009

EDIT
FILE W1FL4384

OK, SLIST W1FL4384
067711800716
067715890411
067750730507
067760880402
068822334009

OK,



    SEG #INTERFACE4384

     ************************************************************************
     *    THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL                  *
     *    THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM.              *
     ************************************************************************
ENTER USER CODE:DJE201
ENTER PASSWORD:

17/06/1981    YOUR LAST ACCESS DATE

15:25:57      YOUR LAST ACCESS TIME
     ************************************************************************
     *                                                                      *
     *              COMPUTER  STUDIES  DEPARTMENT                           *
     *       PROMOTIONAL LITERATURE PUBLISHING SYSTEM                       *
     *                                                                      *
     *                         MENU                                         *
     *                         ----                                         *
     *                                                                      *
     *                                                                      *
     * 01-MENU FILE MAINTENANCE DIALOGUE.                                   *
     * 02-RELATIONS FILE MAINTENANCE DIALOGUE.                              *
     * 03-USERS ACCOUNT FILE MAINTENANCE DIALOGUE.                          *
     * 04-USERS ACCOUNT REPORT.                                             *
     * 05-INTERFACE REPORT(LEGAL AND ILLEGAL ENTRIES).                      *
     * 06-DATABASE MAINTENANCE.                                             *
     * 07-RELATION DATA LIST.                                               *
     ************************************************************************

ENTER CHOICE:      08
INCORRECT CHOICE
```

```
ENTER CHOICE:      07
ENTER MAX. 72 CHARACTERS OPERATION DETAIL
PRINT CATEGORY
ENTER MAX. 8 CH. FILE NAME
CATE4384
067710750202

067710750205

067710750211

067711800703

067711800704

067711800705

067711800707

067712430905

067713830X04

067713840715

067715890401

067715890405

067715890411

067715890412

067750730504

067750730507

067750730508

067750730512

END OF RELATION LIST
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  YES

ENTER CHOICE:      06
ENTER MAX. 72 CHARACTERS OPERATION DETAIL
UNION WF1 AND CATEGORY
RECORD TO BE INSERTED IS:
067711800716

RECORD TO BE INSERTED IS:
067715890411

067715890411ERROR, RECORD EXISTS IN THE FILE
RECORD TO BE INSERTED IS:
067750730507

067750730507ERROR, RECORD EXISTS IN THE FILE
RECORD TO BE INSERTED IS:
067760880402

RECORD TO BE INSERTED IS:
068822334009
```

```
DO YOU WANT MORE UPDATE OPERATIONS
ENTER YES OR NO!  NO
END OF DATABASE UPDATE
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  YES

ENTER CHOICE:      07
ENTER MAX. 72 CHARACTERS OPERATION DETAIL
PRINT CATEGORY
ENTER MAX. 8 CH. FILE NAME
CATE4384
067710750202

067710750205

067710750211

067711800703

067711800704

067711800705

067711800707

067711800716

067712430905

067713830X04

067713840715

067715890401

067715890405

067715890411

067715890412

067750730504

067750730507

067750730508

067750730512

067760880402

068822334009

END OF RELATION LIST
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  NO
DO YOU WANT TO CHANGE YOUR PASSWORD?
ENTER YES OR NO!  NO


END OF THE FUNCTIONS

OK.
```

LIST 5.9

```
    SEG #INTERFACE4384

     ************************************************************
     *     THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL     *
     *     THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM.  *
     ************************************************************
ENTER USER CODE:DJE201
ENTER PASSWORD:

17/06/1981    YOUR LAST ACCESS DATE

16:03:22      YOUR LAST ACCESS TIME
     ************************************************************
     *                                                          *
     *            COMPUTER  STUDIES  DEPARTMENT                 *
     *       PROMOTIONAL LITERATURE PUBLISHING SYSTEM           *
     *                                                          *
     *                        MENU                              *
     *                        ----                              *
     *                                                          *
     *                                                          *
     * 01-MENU FILE MAINTENANCE DIALOGUE.                       *
     * 02-RELATIONS FILE MAINTENANCE DIALOGUE.                  *
     * 03-USERS ACCOUNT FILE MAINTENANCE DIALOGUE.              *
     * 04-USERS ACCOUNT REPORT.                                 *
     * 05-INTERFACE REPORT(LEGAL AND ILLEGAL ENTRIES).          *
     * 06-DATABASE MAINTENANCE.                                 *
     * 07-DIRECT USE OF THE RELATIONAL OPERATORS.               *
     * 08-RELATION DATA LIST.                                   *
     ************************************************************

ENTER CHOICE:      08
ENTER MAX. 72 CHARACTERS OPERATION DETAIL
PRINT QUOTATION
ENTER MAX. 8 CH. FILE NAME
QUOT4384
067710510X00300

067710750202260

067711800701000

067711810400500

067711820102000

067712430900900

067712440600800

067712450300660

067713830X00790

067713840701450

067714710402550

067715890400800
```

Continue List 5.9

```
067715895500600

067750730500900

067750735600400

END OF RELATION LIST
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  YES

ENTER CHOICE:      07
BEWARE PLEASE:
 THE SORTN(AUTHOR NAME SORT ALPHABETICALLY)
 SUBROUTINE, USE WF1 AS INPUT AND WF2 AS OUTPUT

 THE SORTT(TITLE SORT ALPHABETICALLY)
 SUBROUTINE,USE WF3 AS INPUT AND WF1 AS OUTPUT
**************************************************
ENTER MAX. 72 CHARACTERS OPERATION DETAIL:
SELECT QUOTATION WHERE #ISBN = ,06771075""510X, WF1
ENTER MAX. 72 CHARACTERS OPERATION DETAIL:
END OF OPERATIONS
ENTER #ISBN         PLEASE:
067710510X
END OF DIRECT USE OF THE OPERATIONS
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  YES

ENTER CHOICE:      06
ENTER MAX. 72 CHARACTERS OPERATION DETAIL
MINUS WF1 FROM QUOTATION
RECORD TO BE DELETED IS:
067710510X00300

DO YOU WANT MORE UPDATE OPERATIONS
ENTER YES OR NO!  NO
END OF DATABASE UPDATE
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  YES
```

Continue List 5.9

```
ENTER CHOICE:      08
ENTER MAX.  72 CHARACTERS OPERATION DETAIL
PRINT QUOTATION
ENTER MAX.  8 CH.  FILE NAME
QUOT4384
067710750202260

067711800701000

067711810400500

067711820102000

067712430900900

067712440600800

067712450300660

067713830X00790

067713840701450

067714710402550

067715890400800

067715895500600

067750730500900

067750735600400

END OF RELATION LIST
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  NO
DO YOU WANT TO CHANGE YOUR PASSWORD?
ENTER YES OR NO!  NO


END OF THE FUNCTIONS

OK,
```

Continue List 5.9

```
    ED W1FL4384
EDIT
N
067710510X00300
C/03/32/
067710510X03200
FIL

OK, SEG #INTERFACE4384

      ************************************************************
      *     THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL     *
      *     THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM.  *
      ************************************************************
ENTER USER CODE:DJE201
ENTER PASSWORD:

17/06/1981    YOUR LAST ACCESS DATE

16:20:48      YOUR LAST ACCESS TIME
      ************************************************************
      *                                                          *
      *            COMPUTER  STUDIES  DEPARTMENT                 *
      *       PROMOTIONAL LITERATURE PUBLISHING SYSTEM           *
      *                                                          *
      *                        MENU                              *
      *                        ----                              *
      *                                                          *
      *                                                          *
      * 01-MENU FILE MAINTENANCE DIALOGUE.                       *
      * 02-RELATIONS FILE MAINTENANCE DIALOGUE.                  *
      * 03-USERS ACCOUNT FILE MAINTENANCE DIALOGUE.              *
      * 04-USERS ACCOUNT REPORT.                                 *
      * 05-INTERFACE REPORT(LEGAL AND ILLEGAL ENTRIES).          *
      * 06-DATABASE MAINTENANCE.                                 *
      * 07-DIRECT USE OF THE RELATIONAL OPERATORS.               *
      * 08-RELATION DATA LIST.                                   *
      ************************************************************

ENTER CHOICE:     06
ENTER MAX. 72 CHARACTERS OPERATION DETAIL
UNION WF1 AND QUOTATION
RECORD TO BE INSERTED IS:
067710510X03200

DO YOU WANT MORE UPDATE OPERATIONS
ENTER YES OR NO!  NO
END OF DATABASE UPDATE
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  YES

ENTER CHOICE:     08
ENTER MAX. 72 CHARACTERS OPERATION DETAIL
```

Continue List 5.9

```
PRINT QUOTATION
ENTER MAX.  8 CH.  FILE NAME
QUOT4384
067710510X03200

067710750202260

067711800701000

067711810400500

067711820102000

067712430900900

067712440600800

067712450300660

067713830X00790

067713840701450

067714710402550

067715890400800

067715895500600

067750730500900

067750735600400

END OF RELATION LIST
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  NO
DO YOU WANT TO CHANGE YOUR PASSWORD?
ENTER YES OR NO!  NO


END OF THE FUNCTIONS

OK.
```

LIST 5.10

```
    SEG #INTERFACE4384

      ************************************************************
      *     THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL     *
      *     THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM. *
      ************************************************************
ENTER USER CODE:FMA111
ENTER PASSWORD:

09/06/1981    YOUR LAST ACCESS DATE

19:40:40      YOUR LAST ACCESS TIME
      ************************************************************
      *                                                          *
      *             COMPUTER  STUDIES  DEPARTMENT                *
      *       PROMOTIONAL LITERATURE PUBLISHING SYSTEM           *
      *                                                          *
      *                         MENU                             *
      *                         ----                             *
      *                                                          *
      *                                                          *
      * 01-USERS ACCOUNT REPORT.                                 *
      * 02-BOOKS LIST ACCORDING TO AUTHOR AND CATEGORY.          *
      * 03-PRICE LIST ACCORDING TO AUTHOR AND CATEGORY.          *
      * 04-TITLE LIST ACCORDING TO AUTHOR AND CATEGORY.          *
      ************************************************************

ENTER CHOICE:      02
ENTER #CATEG-CODE  PLEASE:
05
SORT STARTED ON AUTHOR NAME
SORT FINISHED
END OF LISTING
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  NO
DO YOU WANT TO CHANGE YOUR PASSWORD?
ENTER YES OR NO!  NO

PLEASE ENTER: SPOOL PRINT384 -X1 (AFTER OK)
END OF THE FUNCTIONS

OK, SPOOL PRINT384 -X1

OK,
```

LIST 5.11

LISTING OF PRINT384 17:13 17 JUN 81

ASINGER/OVERBEEK/PAQOUT

CHEMISTRY, PHYSICS AND APPLICATION OF SU RFACE ACTIVE SUBSTANCES

VOLUME 1:CHEMISTRY OF SURFACE ACTIVE SUB STANCES

0610PP 1967 0 677 11800 7

VOLUME 2:PHYSICS AND PHYSICAL CHEMISTRY OF SURFACE ACTIVE SUBSTANCES

1390PP 1967 0 677 11810 4

VOLUME 3:APPLICATION OF SURFACE ACTIVE S UBSTANCES

1004PP 1967 0 677 11820 1

3 -VOL. SET

0 677 10510 X

BAHN

REACTION RATE COMPILATION FOR H-O-H SYST EM

0240PP 1968 0 677 10750 2

DEVRIES/KOCHVA

TOXINS OF ANIMAL AND PLANT ORIGIN

PROCEEDING OF THE 31ST ANNUAL MEETING OF THE ANIMAL AND PLANT ASSOCIATION ORIGIN , FRANCE 1971

EDITED BY A. J. DEVRIES AND N. M. KOCHVA , DEPARTMENT OF BIOLOGY, UNIVERSITY OF T ECHNOLOGY, LOUGHBOROUGH.

VOL. 1

0502PP 1981 0 677 12430 9

VOLUME 2:BRIEF HISTORY OF ANIMAL AND PLA NT ORIGIN

IT IS VERY IMPORTANT TO STUDY HISTORY AN D ORIGIN OF THE ANIMALS AND PLANT IN WOR LD.

CONTENTS: THIS BOOK CONTAINS THE HISTORY OF ANIMAL IN THE EARLY STAGE OF THE LIF E

0338PP 1981 0 677 12440 6

VOL. 3

0292PP 1981 0 677 12450 3

3 -VOL. SET

0 677 14710 4

FREI/HUZINGUR

ANALYTICAL ASPECTS OF MERCURY AND OTHER HEAVY METALS IN THE ENVIRONMENT

PROCEEDING OF THE 1ST HEAVY METAL, U. S. A. 1975

EDITED BY ROLAND W. FREI, ANALYTICAL REA SEARCH AND DEVELOPMENT,PHARMACEUTICAL DE PARTMENT, SANDOZ LTD., SWITZERLAND.

INCREASED AWARENESS OF THE HARMFUL EFFEC TS OF MERCURY AND OTHER HEAVY METALS SUC H AS LEAD, CADMIUM, ANTIMONY, ETC., IN T HE ENVIRONMENT MEANS THAT LIMITS OF CONT AMINATION HAVE BEEN SET IN THE WORLD

CONTENTS IN BRIEF: USE OF MERCURY IN AGR ICULTURE AND ITS RELATIONSHIP TO ENVIRON MENTAL POLLUTION. THE MICRODETECTION OF MERCURY AND ORGANOMERCURY COMPOUNDS IN E NVIRONMENTAL MATERIALS.

0204PP 1975 0 677 15890 4(CLOTH)

0 677 15895 5(PAPER)

LIST 5.12

```
OK, SEG #INTERFACE4384

     ************************************************************************
     *     THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL                 *
     *     THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM.             *
     ************************************************************************
ENTER USER CODE:FMA111
ENTER PASSWORD:

17/06/1981    YOUR LAST ACCESS DATE

16:31:37      YOUR LAST ACCESS TIME
     ************************************************************************
     *                                                                      *
     *              COMPUTER  STUDIES  DEPARTMENT                           *
     *       PROMOTIONAL LITERATURE PUBLISHING SYSTEM                       *
     *                                                                      *
     *                         MENU                                         *
     *                         ----                                         *
     *                                                                      *
     *                                                                      *
     * 01-USERS ACCOUNT REPORT.                                             *
     * 02-BOOKS LIST ACCORDING TO AUTHOR AND CATEGORY.                      *
     * 03-PRICE LIST ACCORDING TO AUTHOR AND CATEGORY.                      *
     * 04-TITLE LIST ACCORDING TO AUTHOR AND CATEGORY.                      *
     ************************************************************************

ENTER CHOICE:      03
ENTER #CATEG-CODE  PLEASE:
05
SORT STARTED ON AUTHOR NAME
SORT FINISHED
ENTER EXCHANGE RATE PLEASE :
(SUCH AS 9.999)
1.940
END OF LISTING
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  NO
DO YOU WANT TO CHANGE YOUR PASSWORD?
ENTER YES OR NO!  NO

PLEASE ENTER: SPOOL PRINT384 -X1 (AFTER OK)
END OF THE FUNCTIONS

OK, SPOOL PRINT384 -X1

OK,
```

LIST 5.13

LISTING OF PRINT38J 17:41 17 JUN 81

ASINGER/OVERBEEK/PAGOUT

VOL. 1
0 677 11800 7 1967 $10.00 P5.15

VOL. 2
0 677 11810 4 1967 $5.00 P2.57

VOL. 3
0 677 11820 1 1967 $20.00 P10.30

3 -VOL. SET
0 677 10510 X $32.00 P16.49

BAHN
0 677 10750 2 1968 $22.60 P11.64

DEVRIES/KOCHVA

VOL. 1
0 677 12430 9 1981 $9.00 P4.63

VOL. 2
0 677 12440 6 1981 $8.00 P4.12

VOL. 3
0 677 12450 3 1981 $6.60 P3.40

3 -VOL. SET
0 677 14710 4 $25.50 P13.14

FREI/HUZINGUR
0 677 15890 4 CL 1975 $8.00 P4.12
0 677 15895 5 PA $6.00 P3.09

LIST 5.14

```
     ************************************************************
     *     THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL     *
     *     THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM. *
     ************************************************************
ENTER USER CODE:FMA111
ENTER PASSWORD:

17/06/1981    YOUR LAST ACCESS DATE

17:19:05      YOUR LAST ACCESS TIME
     ************************************************************
     *                                                          *
     *            COMPUTER  STUDIES  DEPARTMENT                 *
     *       PROMOTIONAL LITERATURE PUBLISHING SYSTEM           *
     *                                                          *
     *                        MENU                              *
     *                        ----                              *
     *                                                          *
     *                                                          *
     * 01-USERS ACCOUNT REPORT.                                 *
     * 02-BOOKS LIST ACCORDING TO AUTHOR AND CATEGORY.          *
     * 03-PRICE LIST ACCORDING TO AUTHOR AND CATEGORY.          *
     * 04-TITLE LIST ACCORDING TO AUTHOR AND CATEGORY.          *
     ************************************************************

ENTER CHOICE:      04
ENTER #CATEG-CODE  PLEASE:
05
SORT STARTED ON TITLE DATA
SORT FINISHED
END OF LISTING
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  NO
DO YOU WANT TO CHANGE YOUR PASSWORD?
ENTER YES OR NO!  NO

PLEASE ENTER: SPOOL PRINT384 -X1 (AFTER OK)
END OF THE FUNCTIONS

OK, SPOOL PRINT384 -X1

OK,
```

LIST 5.15

LISTING OF PRINT384 17:46 17 JUN 81

001

ANALYTICAL ASPECTS OF MERCURY AND OTHER HEAVY METALS IN THE ENVIRONMENT	/FREI/HUZINGUR
CHEMISTRY, PHYSICS AND APPLICATION OF SURFACE ACTIVE SUBSTANCES	/ASINGER/OVERBEEK/PAUOUT
REACTION RATE COMPILATION FOR H-O-H SYSTEM	/BAHN
TOXINS OF ANIMAL AND PLANT ORIGIN	/DEVRIES/KOCHVA

LIST 5.16

```
      ************************************************************
      *     THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL     *
      *     THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM.  *
      ************************************************************
ENTER USER CODE:JMA243
ENTER PASSWORD:

09/06/1981    YOUR LAST ACCESS DATE

18:50:50      YOUR LAST ACCESS TIME
      ************************************************************
      *                                                          *
      *            COMPUTER  STUDIES  DEPARTMENT                 *
      *       PROMOTIONAL LITERATURE PUBLISHING SYSTEM           *
      *                                                          *
      *                        MENU                              *
      *                        ----                              *
      *                                                          *
      *                                                          *
      * 01-MENU FILE MAINTENANCE DIALOGUE.                       *
      * 02-RELATIONS FILE MAINTENANCE DIALOGUE.                  *
      * 03-USERS ACCOUNT FILE MAINTENANCE DIALOGUE.              *
      * 04-USERS ACCOUNT REPORT.                                 *
      * 05-INTERFACE REPORT(LEGAL AND ILLEGAL ENTRIES).          *
      * 06-DATABASE MAINTENANCE.                                 *
      * 07-BOOKS LIST ACCORDING TO AUTHOR AND CATEGORY.          *
      * 08-PRICE LIST ACCORDING TO AUTHOR AND CATEGORY.          *
      * 09-TITLE LIST ACCORDING TO AUTHOR AND CATEGORY.          *
      * 10-DIRECT USE OF THE RELATIONAL OPERATORS.               *
      * 11-RELATION DATA LIST.                                   *
      ************************************************************

ENTER CHOICE:     10
BEWARE PLEASE:
 THE SORTN(AUTHOR NAME SORT ALPHABETICALLY)
 SUBROUTINE. USE WF1 AS INPUT AND WF2 AS OUTPUT

 THE SORTT(TITLE SORT ALPHABETICALLY)
 SUBROUTINE,USE WF3 AS INPUT AND WF1 AS OUTPUT
************************************************
ENTER MAX. 72 CHARACTERS OPERATION DETAIL:
SELECT CATEGORY WHERE CATEG-CODE = ,02. WF7
ENTER MAX. 72 CHARACTERS OPERATION DETAIL:
END OF OPERATIONS
ENTER CATEG-CODE   PLEASE:
02
CATEGORY     INCORRECT RELATION NAME
WF7          INCORRECT RELATION NAME
CATEG-CODE  (INCORRECT NAME)
THERE IS NO OUTPUT RECORD
THE RUN WAS NOT SATISFACTORY
YOU CAN NOT CONTINUE WITH THIS RUN
END OF DIRECT USE OF THE OPERATIONS
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  NO
DO YOU WANT TO CHANGE YOUR PASSWORD?
ENTER YES OR NO!  NO


END OF THE FUNCTIONS

OK.
```

```
      ************************************************************
      *     THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL     *
      *     THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM. *
      ************************************************************
ENTER USER CODE:JMA243
ENTER PASSWORD:

17/06/1981    YOUR LAST ACCESS DATE

17:51:40      YOUR LAST ACCESS TIME
      ************************************************************
      *                                                          *
      *            COMPUTER  STUDIES  DEPARTMENT                 *
      *       PROMOTIONAL LITERATURE PUBLISHING SYSTEM           *
      *                                                          *
      *                        MENU                              *
      *                        ----                              *
      *                                                          *
      *                                                          *
      * 01-MENU FILE MAINTENANCE DIALOGUE.                       *
      * 02-RELATIONS FILE MAINTENANCE DIALOGUE.                  *
      * 03-USERS ACCOUNT FILE MAINTENANCE DIALOGUE.              *
      * 04-USERS ACCOUNT REPORT.                                 *
      * 05-INTERFACE REPORT(LEGAL AND ILLEGAL ENTRIES).          *
      * 06-DATABASE MAINTENANCE.                                 *
      * 07-BOOKS LIST ACCORDING TO AUTHOR AND CATEGORY.          *
      * 08-PRICE LIST ACCORDING TO AUTHOR AND CATEGORY.          *
      * 09-TITLE LIST ACCORDING TO AUTHOR AND CATEGORY.          *
      * 10-DIRECT USE OF THE RELATIONAL OPERATORS.               *
      * 11-RELATION DATA LIST.                                   *
      ************************************************************

ENTER CHOICE:     10
BEWARE PLEASE:
 THE SORTN(AUTHOR NAME SORT ALPHABETICALLY)
 SUBROUTINE, USE WF1 AS INPUT AND WF2 AS OUTPUT

 THE SORTT(TITLE SORT ALPHABETICALLY)
 SUBROUTINE,USE WF3 AS INPUT AND WF1 AS OUTPUT
*************************************************
ENTER MAX. 72 CHARACTERS OPERATION DETAIL:
SELECT PAGEYEAR WHERE PUB-YEAR > ,79, WF3
ENTER MAX. 72 CHARACTERS OPERATION DETAIL:
JOIN WF3 AND AUTHORTITLE OVER #BLURB-NO WF1
ENTER MAX. 72 CHARACTERS OPERATION DETAIL:
PROJECT WF1 F01F02F06F04,E WF2
ENTER MAX. 72 CHARACTERS OPERATION DETAIL:
JOIN WF2 AND ISBN OVER #BLURB-NO #PART-VOLUME WF3
ENTER MAX. 72 CHARACTERS OPERATION DETAIL:
PROJECT WF3 F01F02F03F05F04,E WF1
ENTER MAX. 72 CHARACTERS OPERATION DETAIL:
JOIN WF1 AND QUOTATION OVER #ISBN WF2
ENTER MAX. 72 CHARACTERS OPERATION DETAIL:
PROJECT WF2 F01F02F03F04F05F07,E WF1
ENTER MAX. 72 CHARACTERS OPERATION DETAIL:
SORTN
ENTER MAX. 72 CHARACTERS OPERATION DETAIL:
PRINT WF2 (PRICE LIST)
ENTER MAX. 72 CHARACTERS OPERATION DETAIL:
END OF OPERATIONS
ENTER PUB-YEAR     PLEASE:
79
SORT STARTED ON AUTHOR NAME
SORT FINISHED
ENTER EXCHANGE RATE PLEASE :
```

Continue List 17

```
(SUCH AS 9.999)
1.940
END OF DIRECT USE OF THE OPERATIONS
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  NO
DO YOU WANT TO CHANGE YOUR PASSWORD?
ENTER YES OR NO!  NO

PLEASE ENTER: SPOOL PRINT384 -X1 (AFTER OK)
END OF THE FUNCTIONS

OK, SPOOL PRINT384 -X1

OK,
```

LIST 5.18

```
LISTING OF PRINT384   18:07   17 JUN 81

DEVRIES/KOCHVA

VOL. 1
0 677 12430 9      1981    $9.00     P4.63

VOL. 2
0 677 12440 6      1981    $8.00     P4.12

VOL. 3
0 677 12450 3      1981    $6.60     P3.40

INSTITUTE OF REACTION TRANSITION
0 677 50730 5 CL 1980    $9.00     P4.63
0 677 50735 6 PA         $4.00     P2.06
```

LIST 5.19

```
      ************************************************************
      *     THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL     *
      *     THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM.  *
      ************************************************************
ENTER USER CODE:DJE201
ENTER PASSWORD:

17/06/1981    YOUR LAST ACCESS DATE

16:25:56      YOUR LAST ACCESS TIME
      ************************************************************
      *                                                          *
      *            COMPUTER  STUDIES  DEPARTMENT                 *
      *       PROMOTIONAL LITERATURE PUBLISHING SYSTEM           *
      *                                                          *
      *                        MENU                              *
      *                        ----                              *
      *                                                          *
      *                                                          *
      * 01-MENU FILE MAINTENANCE DIALOGUE.                       *
      * 02-RELATIONS FILE MAINTENANCE DIALOGUE.                  *
      * 03-USERS ACCOUNT FILE MAINTENANCE DIALOGUE.              *
      * 04-USERS ACCOUNT REPORT.                                 *
      * 05-INTERFACE REPORT(LEGAL AND ILLEGAL ENTRIES).          *
      * 06-DATABASE MAINTENANCE.                                 *
      * 07-DIRECT USE OF THE RELATIONAL OPERATORS.               *
      * 08-RELATION DATA LIST.                                   *
      ************************************************************

ENTER CHOICE:      07
BEWARE PLEASE:
 THE SORTN(AUTHOR NAME SORT ALPHABETICALLY)
 SUBROUTINE, USE WF1 AS INPUT AND WF2 AS OUTPUT

 THE SORTT(TITLE SORT ALPHABETICALLY)
 SUBROUTINE,USE WF3 AS INPUT AND WF1 AS OUTPUT
*************************************************
ENTER MAX. 72 CHARACTERS OPERATION DETAIL:
SELECT CATEGORY WHERE #CATEG-CODE = ,22, WF1
ENTER MAX. 72 CHARACTERS OPERATION DETAIL:
END OF OPERATIONS
ENTER #CATEG-CODE  PLEASE:
22
THERE IS NO OUTPUT RECORD
THE RUN WAS NOT SATISFACTORY
YOU CAN NOT CONTINUE WITH THIS RUN
END OF DIRECT USE OF THE OPERATIONS
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  NO
DO YOU WANT TO CHANGE YOUR PASSWORD?
ENTER YES OR NO!    NO


END OF THE FUNCTIONS

OK,
```

List 20

```
      ************************************************************
      *     THE FUNCTION OF THIS INTERFACE IS TO CONTROL ALL     *
      *     THE PROCESSING OF THE PROMOTIONAL PUBLISHING SYSTEM.  *
      ************************************************************
ENTER USER CODE:DJE201
ENTER PASSWORD:

17/06/1981    YOUR LAST ACCESS DATE

18:07:50      YOUR LAST ACCESS TIME
      ************************************************************
      *                                                          *
      *            COMPUTER  STUDIES  DEPARTMENT                 *
      *       PROMOTIONAL LITERATURE PUBLISHING SYSTEM           *
      *                                                          *
      *                        MENU                              *
      *                        ----                              *
      *                                                          *
      *                                                          *
      * 01-MENU FILE MAINTENANCE DIALOGUE.                       *
      * 02-RELATIONS FILE MAINTENANCE DIALOGUE.                  *
      * 03-USERS ACCOUNT FILE MAINTENANCE DIALOGUE.              *
      * 04-USERS ACCOUNT REPORT.                                 *
      * 05-INTERFACE REPORT(LEGAL AND ILLEGAL ENTRIES).          *
      * 06-DATABASE MAINTENANCE.                                 *
      * 07-DIRECT USE OF THE RELATIONAL OPERATORS.               *
      * 08-RELATION DATA LIST.                                   *
      ************************************************************

ENTER CHOICE:      07
BEWARE PLEASE:
 THE SORTN(AUTHOR NAME SORT ALPHABETICALLY)
 SUBROUTINE, USE WF1 AS INPUT AND WF2 AS OUTPUT

 THE SORTT(TITLE SORT ALPHABETICALLY)
 SUBROUTINE,USE WF3 AS INPUT AND WF1 AS OUTPUT
*************************************************
ENTER MAX. 72 CHARACTERS OPERATION DETAIL:
SELECT QUOTATION WHERE BOOK-PRICE => ,00500, WF2
ENTER MAX. 72 CHARACTERS OPERATION DETAIL:
END OF OPERATIONS
ENTER BOOK-PRICE   PLEASE:
00500
END OF DIRECT USE OF THE OPERATIONS
DO YOU WANT TO RUN ANOTHER JOB?
ENTER YES OR NO!  NO
DO YOU WANT TO CHANGE YOUR PASSWORD?
ENTER YES OR NO! · NO


END OF THE FUNCTIONS

OK, SLIST W2FL4384
067710510X03200
067710750202260
067711800701000
067711810400500
067711820102000
067712430900900
067712440600800
067712450300660
067713830X00790
067713840701450
067714710402550
067715890400800
067715895500600
067750772050000
```

5.3 Conclusions

The object of the work described in this thesis was to develop a model database for a typical promotional publishing house. Initially, the problems of their cataloguing system which was required to produce the main types list, and the security and integrity of the batch system was discussed.

The advantages of the relational database which has been presented include simplicity, data independence, symmetry, with a theoretical foundation.

A relational database model has been implemented which satisfies the requirements for data storage i.e. safe, compact and is easily accessible and changeable.

The relationship operators (UNION, MINUS, SELECT, PROJECT, JOIN, COMPLEMENT) have been implemented to manipulate the database.

Also, this thesis has discussed the terminology of the relational data model and traced its development in terms of normalization theory and implementation techniques. A high level interface has been implemented which assists the unsophisticated user for making the system easy to use. Menus are automatically generated with each menu displaying a complete list of processing alternatives to the user. This is a simple but effective technique which saves time and reduces errors.

A number of authentication techniques were implemented to demonstrate possible methods of stopping the unauthorized user from accessing the information stored in the database. Two programs were also implemented to produce reports to inform the DAB about any legal or

illegal attempts to the database helping him by indicating the weak points of the security techniques for future developments. The direct use of the relational operators and the subroutines has been implemented which permits special users (users who know exactly how the database is organized) to manipulate the database and obtain the required data.

The model provides the foundation for a complete listing system.

The coding is based on structured programming techniques, as far as possible, and can be altered fairly simply. The COPY statement is used in conjunction with a 'library' of copied material for areas which occur more than once. So that once the master copy of an item to be altered is updated, then all that has to be done is to re-compile any program using that particular 'copy file'. The model has been implemented on the PRIME 400 computer system. The PRIME had several useful and powerful utilities which assisted the developed model.

It is clear, however, that further attention could be directed to data validation, and to securing the information against system malfunction.

5.4 Future Enhancements

The major enhancement possible for the future is to extend the system so that stock records can be added to the database concerning the books, so that orders can be dealt with directly. This would require a considerable amount of work.

It should also be possible to develop a systematic method for evaluating the database system and for measuring its cost and effectiveness.

Finally, it would be beneficial if the same database system could be applied to bibliographic retrieval for use in libraries. This will need a considerable amount of work on the user communication aspects to convert it to a complete on-line system for use by more than one user.

APPENDIX A

LISTING OF FILES

```
LISTING OF GENERALP  11:47  28 MAY 81

SECU4384
DJE201     0151024113940810528PROF. D. J. EVANS         0102030405060708091O
JMA243     0240924182617810527MR.   J. M. ABDUL-JABBAR 0102030405060708091O
LNB235     0001122000000000000MR.   L. N. BLACKBURN     01020910
```

List of the User's Account File.

```
LISTING OF GENERALP  16:14  16 JUN 81

MENU4384
01MENU FILE MAINTENANCE DIALOGUE.
02RELATIONS FILE MAINTENANCE DIALOGUE.
03USERS ACCOUNT FILE MAINTENANCE DIALOGUE.
04USERS FILE STATUS REPORT.
05INTERFACE REPORT(LEGAL AND ILLEGAL ENTRIES).
06DATABASE MAINTENANCE.
07BOOKS LIST ACCORDING TO AUTHOR AND CATEGORY.
08PRICE LIST ACCORDING TO AUTHOR AND CATEGORY.
09TITLE LIST ACCORDING TO AUTHOR AND CATEGORY.
10DIRECT USE OF THE RELATIONAL OPERATORS.
11RELATION LIST(FIELDS :NAME,LENGTH,AND DATA).
```

List of the Menu File.

```
LISTING OF GENERALP  16:32  16 JUN 81

RELA4384
AUTHORTITLE 02#BLURB-NO    001010AUTH-NAME    011042BOOK-TITLE  053100
CATEGORY    01#BLURB-NO    001010#CATEG-CODE  011002
CONFERENCE  03#BLURB-NO    001010CONFERENCE   011120
CONTENT     08#BLURB-NO    001010#PART-VOLUME011003CONTENT      014200
DESCRIPTION 07#BLURB-NO    001010#PART-VOLUME011003DESCRIPTION  014200
EDITOR      04#BLURB-NO    001010EDITOR       011160
ISBN        09#ISBN        001010#BLURB-NO    011010#PART-VOLUME021003
PAGEYEAR    05#BLURB-NO    001010#PART-VOLUME011003NO-OF-PAGE   014004PUB-YEAR
QUOTATION   10#ISBN        001010BOOK-PRICE   011005
SUBTITLE    06#BLURB-NO    001010#PART-VOLUME011003SUBTITLE     014100
WF1         11#BLURB-NO    001010#CATEG-CODE  011002
WF2         12#BLURB-NO    001010#PART-VOLUME011003AUTH-NAME    014042#ISBN
10BOOK-PRICE  078005
WF3         13#BLURB-NO    001010#PART-VOLUME011003AUTH-NAME    014042PUB-YEAR
10#PART-VOLUME078003
```

List of the Relations Description File

```
LISTING OF GENERALP  16:21  16 JUN 81

FUNC4384
0701SELECT CATEGORY WHERE #CATEG-CODE = '05' WF1
0702JOIN AUTHORTITLE AND WF1 OVER #BLURB-NO WF2
0703PROJECT WF2 F01F02F03,E WF3
0704COMP CONFERENCE BY WF3 OVER #BLURB-NO WF1
0705JOIN WF3 AND WF1 OVER #BLURB-NO WF2
0706PROJECT WF2 F01F02F03F05,E WF3
0707COMP EDITOR BY WF3 OVER #BLURB-NO WF1
0708JOIN WF3 AND WF1 OVER #BLURB-NO WF2
0709PROJECT WF2 F01F02F03F04F06,E WF3
0710JOIN WF3 AND PAGEYEAR OVER #BLURB-NO WF1
0711PROJECT WF1 F01F07F02F03F04F05F08F09,E WF2
0712COMP SUBTITLE BY WF2 OVER #BLURB-NO #PART-VOLUME WF3
0713JOIN WF2 AND WF3 OVER #BLURB-NO #PART-VOLUME WF1
0714PROJECT WF1 F01F02F03F04F05F06F11F07F08,E WF2
0715COMP DESCRIPTION BY WF2 OVER #BLURB-NO #PART-VOLUME WF3
0716JOIN WF2 AND WF3 OVER #BLURB-NO #PART-VOLUME WF1
0717PROJECT WF1 F01F02F03F04F05F06F07F12F08F09,E WF2
0718COMP CONTENT BY WF2 OVER #BLURB-NO #PART-VOLUME WF3
0719JOIN WF2 AND WF3 OVER #BLURB-NO #PART-VOLUME WF1
0720PROJECT WF1 F01F02F03F04F05F06F07F08F13F09F10,E WF2
0721JOIN WF2 AND ISBN OVER #BLURB-NO #PART-VOLUME WF3
0722PROJECT WF3 F01F02F03F04F05F06F07F08F09F10F11F12,E WF1
0723SORTN
0724PRINT WF2 (BOOKS LIST)
0725END OF OPERATIONS
0801SELECT CATEGORY WHERE #CATEG-CODE = '05' WF1
0802JOIN AUTHORTITLE AND WF1 OVER #BLURB-NO WF2
0803PROJECT WF2 F01F02F03,E WF3
0804JOIN PAGEYEAR AND WF3 OVER #BLURB-NO WF1
0805PROJECT WF1 F01F02F06F04,E WF2
0806JOIN WF2 AND ISBN OVER #BLURB-NO #PART-VOLUME WF3
0807PROJECT WF3 F01F02F03F05F04,E WF1
0808JOIN WF1 AND QUOTATION OVER #ISBN WF2
0809PROJECT WF2 F01F02F03F04F05F07,E WF1
0810SORTN
0811PRINT WF2 (PRICE LIST)
0812END OF OPERATIONSS
0901SELECT CATEGORY WHERE #CATEG-CODE = '05' WF1
0902JOIN AUTHORTITLE AND WF1 OVER #BLURB-NO WF2
0903PROJECT WF2 F02F03,E WF3
0904SORTT
0905PRINT WF1 (TITLE AUTHORS LIST)
0906END OF OPERATIONS
```

List of the Listing Operations Detail File.

```
LISTING OF GENERALP  18:26  24 JUN 81

CATE4384
CATEGORY     01#BLURB-NO   001010#CATEG-CODE 011002

067710750202
067710750205
067710750211
067711800703
067711800704
067711800705
067711800707
067711800716
067712430905
067713830X04
067713840715
067715890401
067715890405
067715890411
067715890412
067750730504
067750730507
067750730508
067750730512
067760880402
068822334009
```

List of the Category Relation.

```
LISTING OF GENERALP  12:37  16 JUN 81

AUTH4384
AUTHORTITLE 02#BLURB-NO   001010AUTH-NAME   011042BOOK-TITLE  053100

0677107502BAHN                                    AJ    REACTION RATE COMPILATION FOR
0677118007ASINGER     OVERBEEK     PAGOUT         LMSTTNCHEMISTRY, PHYSICS AND APPLICA
0677124309DEVRIES     KOCHVA                      AJNM  TOXINS OF ANIMAL AND PLANT ORI
06771383OXBROWN                                   LB    LIQUID CRYSTALS IN 2 PARTS
0677158904FREI        HUZINGUR                    TLOP  ANALYTICAL ASPECTS OF MERCURY
ENT
067750730SINSTITUTE OF REACTION TRANSITION        **REACTION TRANSITION STATES
```

List of the AUTHORTITLE Relation.

```
LISTING OF GENERALP  12:41  16 JUN 81

CONF4384
CONFERENCE  03#BLURB-NO   001010CONFERENCE  011120

0677124309PROCEEDING OF THE 21ST ANNUAL MEETING OF THE ANIMAL AND PLANT ASSOCIATIO
0677158904PROCEEDING OF THE 1ST HEAVY METAL, U. S. A. 1975
```

List of the CONFERENCE Relation.

```
LISTING OF GENERALP  12:45  16 JUN 81

EDIT4384
EDITOR      04#BLURB-NO   001010EDITOR      011160

0677124309EDITED BY A. J. DEVRIES AND N. M. KOCHVA, DEPARTMENT OF BIOLOGY, UNIVERS
0677158904EDITED BY ROLAND W. FREI, ANALYTICAL REASEARCH AND DEVELOPMENT,PHARMACEU
LAND.
```

List of the EDITOR Relation.

```
LISTING OF GENERALP  12:48  16 JUN 81

PAGE4384
PAGEYEAR    05#BLURB-NO   001010#PART-VOLUME011003NO-OF-PAGE  014004PUB-YEAR    018002

0677107502000024068
0677118007001061067
0677118007002139067
0677118007003100467
0677118007003V000000
0677124309001050281
0677124309002033881
0677124309003029281
067712430903V000000
0677133OX100025269
0677133OX200091069
0677158904000020475
0677507305000030480
```

List of the PAGEYEAR Relation.

```
LISTING OF GENERALP  12:52  16 JUN 81

SUBT4384
SUBTITLE     06#BLURB-NO    001010#PART-VOLUME011003SUBTITLE     014100

0677118007001VOLUME 1:CHEMISTRY OF SURFACE ACTIVE SUBSTANCES
0677118007002VOLUME 2:PHYSICS AND PHYSICAL CHEMISTRY OF SURFACE ACTIVE SUBSTANCES
0677118007003VOLUME 3:APPLICATION OF SURFACE ACTIVE SUBSTANCES
0677124309002VOLUME 2:BRIEF HISTORY OF ANIMAL AND PLANT ORIGIN
```

List of the SUB-TITLE Relation.

```
LISTING OF GENERALP  12:55  16 JUN 81

DESC4384
DESCRIPTION 07#BLURB-NO    001010#PART-VOLUME011003DESCRIPTION 014200

0677124309002IT IS VERY IMPORTANT TO STUDY HISTORY AND ORIGIN OF THE ANIMALS AND
0677158904000INCREASED AWARENESS OF THE HARMFUL EFFECTS OF MERCURY AND OTHER HEAV
Y, ETC., IN THE ENVIRONMENT MEANS THAT LIMITS OF CONTAMINATION HAVE BEEN SET IN TI
```

List of the DESCRIPTION Relation

```
LISTING OF GENERALP  12:57  16 JUN 81

CONT4384
CONTENT      08#BLURB-NO    001010#PART-VOLUME011003CONTENT      014200

0677124309002CONTENTS: THIS BOOK CONTAINS THE HISTORY OF ANIMAL IN THE EARLY STAG
0677158904000CONTENTS IN BRIEF: USE OF MERCURY IN AGRICULTURE AND ITS RELATIONSHI
DETECTION OF MERCURY AND ORGANOMERCURY COMPOUNDS IN ENVIRONMENTAL MATERIALS.
```

List of the CONTENT Relation.

```
LISTING OF GENERALP  13:01  16 JUN 81

ISBN4384
ISBN          09#ISBN         001010#BLURB-NO    011010#PART-VOLUME021003

0677105l0X0677118007037
0677107502067710750200O
0677118007067711800700l
0677118104067711810040l
0677118201067711820070l
0677124309067712430900l
0677124406067712440060l
0677124503067712450030l
0677138l0X0677138l0X100
0677138407067713840l450
0677147104067714710402l
0677158904067715890400O
0677158955067715895500O
0677507305067750730500O
0677507356067750735600O
```

List of the I S B N Relation.

```
LISTING OF GENERALP  18:30  24 JUN 81

QUOT4384
QUOTATION    10#ISBN         001010BOOK-PRICE   011005

067710510X03200
067710750202260
067711800701000
067711810400500
067711820102000
067712430900900
067712440600800
067712450300660
067713830X00790
067713840701450
067714710402550
067715890400800
067715895500600
067750730500900
067750735600400
```

List of the QUOTATION Relation.

APPENDIX B

LISTING OF PROGRAMS

```
       IDENTIFICATION DIVISION.
      ************************************************************************
      *THE USER INTERFACE CONSISTS OF SEVERAL MODUELS OR SUB-SYSTEMS. IT HAS *
      *TWO MAIN FUNCTIONS. ONE FUNCTION IS TO SUPERVISE THE ACCESSES TO THE  *
      *DATABASE. THE SECOND FUNCTION , IS TO ENABLE USERS, AFTER THEY        *
      *SUCCESSFULLY IDENTIFY THEMSELVES, TO ACCESS THE DATABASE SYSTEM       *
      ************************************************************************
       PROGRAM-ID. INTR01.
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
       SOURCE-COMPUTER. PRIME.
       OBJECT-COMPUTER. PRIME.
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
           SELECT USER-FILE         ASSIGN PFMS
           ORGANIZATION INDEXED
           ACCESS MODE RANDOM
           RECORD KEY USER-NO
           FILE STATUS FILE-STATUS1.
           SELECT MENU-FILE         ASSIGN PFMS
           ORGANIZATION INDEXED
           ACCESS MODE RANDOM
           RECORD KEY MENU-KEY
           FILE STATUS FILE-STATUS2.
           SELECT FUNCTIONS-FILE    ASSIGN PFMS
           ORGANIZATION INDEXED
           ACCESS MODE RANDOM
           RECORD KEY FUNCTION-NO
           FILE STATUS FILE-STATUS3.
           SELECT LOGG-FILE         ASSIGN PFMS.
       DATA DIVISION.
       FILE SECTION.
       FD  USER-FILE
           LABEL RECORDS STANDARD VALUE OF FILE-ID    'SECU4384'.
       01  USER-RECORD.
        02 USER-NO                 PIC X(6).
        02 U-PASSWORD              PIC XXXX.
        02 U-BLACKMARK             PIC 9.
        02 U-NO-LOGIN              PIC 99.
        02 TIME-ALLOWED.
         03 START-TIME             PIC 99.
         03 END-TIME               PIC 99.
        02 LAST-T-ACCESS.
         03 L-HR                   PIC 99.
         03 L-MIN                  PIC 99.
         03 L-SEC                  PIC 99.

        02 LAST-D-ACCESS.
         03 LAST-YER               PIC 99.
         03 LAST-MON               PIC 99.
         03 LAST-DAY               PIC 99.
        02 USER-NAME               PIC X(25).
        02 JOB-ALLOWED             PIC XX    OCCURS 14.
       FD  MENU-FILE
           LABEL RECORDS STANDARD VALUE OF FILE-ID     'MENU4384'.
       01  MENU-RECORD.
        02 MENU-KEY                PIC XX.
        02 APPLICATION-NAME        PIC X(46).
       FD  FUNCTIONS-FILE
           LABEL RECORDS STANDARD VALUE OF FILE-ID 'FUNC4384'.
       01  FUNCTION-RECORD.
        02 FUNCTION-NO.
         03 FUNCTION-NO1           PIC XX.
         03 FUNCTION-NO2           PIC 99.
        02 FILE-COMMAND-DETAIL.
         03 COMMAND-PART1          PIC X(17).
         03 COMMAND-PART2          PIC X(55).
       FD  LOGG-FILE
           LABEL RECORDS STANDARD VALUE OF FILE-ID     'LOGG4384'.
       01  LOGG-RECORD.
        02 L-USER-NO               PIC X(6).
        02 LOG-DATE.
         03 LOG-YER                PIC 99.
         03 LOG-MON                PIC 99.
         03 LOG-DAY                PIC 99.
        02 LOG-START-TIME.
         03 ST-HR                  PIC 99.
         03 ST-MIN                 PIC 99.
         03 ST-SEC                 PIC 99.
        02 LOG-END-TIME.
         03 END-HR                 PIC 99.
         03 END-MIN                PIC 99.
         03 END-SEC                PIC 99.
        02 COMMENT-TYPE            PIC X(46).
       WORKING-STORAGE SECTION.
       77  FILE-STATUS1             PIC XX.
       77  FILE-STATUS2             PIC XX.
       77  FILE-STATUS3             PIC XX.
```

```
*
 77  HALF-DUPLX     COMPUTATIONAL VALUE -32768.
 77  FULL-DUPLX     COMPUTATIONAL VALUE 0.
*
 77  CHECK-AREA           PIC 99.
 77  OPERATION-INDICATOR PIC 9 VALUE 0.
 77  UPDATE-INDICATOR     PIC 9.
 77  END-AREA             PIC XXX VALUE SPACES.
 77  FUNCTION-COUNT       PIC 99 VALUE ZEROS.
 77  LOGIN-COUNT          PIC 9 VALUE ZERO.
 77  COUNT-MOVE           PIC 99.
 77  COUNT-SPOOL          PIC 99.
 77  REWRITE-SW           PIC 9 VALUE ZERO.
*
 77  LINKAGE-MESSAG       PIC X(46).
*
 77  DISPLAY1             PIC X(16) VALUE
     'ENTER USER CODE:'.
 77  DISPLAY2             PIC X(19) VALUE
     'INCORRECT USER CODE'.
 77  DISPLAY3             PIC X(16) VALUE
     'ENTER PASSWORD: '.
 77  DISPLAY4             PIC X(16) VALUE
     'SYSTEM LOCKED  '.
 77  DISPLAY5             PIC X(18) VALUE
     'ENTER CHOICE:     '.
 77  DISPLAY6             PIC X(18) VALUE
     'ENTER YES OR NO!  '.
 77  DISPLAY7             PIC X(20) VALUE
     'ENTER NEW PASSWORD: '.
*
 77  W-USER-CODE          PIC X(6).
 77  W-PASSWORD           PIC X(4).
 77  COUNT-JOB            PIC 99 VALUE ZEROS.
 77  CALL-AREA          PIC 99.
 77  COUNT-AREA           PIC 99 VALUE 0.
 77  YES-NO-AREA          PIC XXX.
*
 77  COUNT1       COMPUTATIONAL VALUE 16.
 77  COUNT2         COMPUTATIONAL VALUE 18.
 77  COUNT3         COMPUTATIONAL VALUE 20.
*
 01  CHOICE-AREA          PIC 99.
 01  CHOICE-AREA1 REDEFINES CHOICE-AREA PIC XX.
 01  MENU-TABEL.
  02 MENU-AREA            OCCURS 24.
   03 MENU-LINE           PIC X(60).
 01  MENU-LINE-A.
  02 MENU-P1            PIC X.
  02 MENU-P2              PIC X.
   02 MENU-P3             PIC 99.
  02 MENU-P4              PIC X.
   02  MENU-P5.
    03 MENU-P51  PIC X(46).
    03 MENU-P52  PIC X(8).
  02 MENU-P6              PIC X.
 01  DATE-TIME-LINE.
  02 DT-1                PIC 99.
  02 DT-2                PIC X.
  02 DT-3                 PIC 99.
  02 DT-4                PIC X.
  02 DT-5                PIC 99.
  02 DT-6                PIC 99.
  02 DT-AREA             PIC X(30).
 01  TABEL-A.
  02  JOB-SEQ            PIC 99 OCCURS 14.
 01  EXEC-DATE.
  02  EX-YER  PIC 99.
  02  EX-MON  PIC 99.
  02  EX-DAY PIC 99.
 01  EXEC-TIME.
  02  EX-HR  PIC 99.
  02  EX-MIN PIC 99.
  02  EX-SEC PIC 99.
 01  SPOOL-AREA.
  02 SPOOL-FILE           OCCURS 14.
   03 SPOOL-ACTION        PIC X(50).
*
 01  FUNCTION-TABEL.
  02 TABEL-COMMAND        OCCURS 50.
   03 TABEL-COMMAND1      PIC X(5).
   03 TABEL-COMMAND2      PIC X(67).
 01  LINKAGE-FUNCTION.
  02 OPERATION-MESSAGE  PIC X(72).
  02 PROGRAM-COMMENT     PIC X(28)
```

```
PROCEDURE DIVISION.
CONTROL-ROUTINE.
    PERFORM INITIALIZE.
    PERFORM MAIN-PARAGRAGH UNTIL END-AREA = 'END'.
    PERFORM CLEANUP.
EXIT-PARA.
    EXIT PROGRAM.
STOP-RUN.
    DISPLAY 'END OF THE FUNCTIONS'.
    STOP RUN.
INITIALIZE.
    OPEN I-O       USER-FILE.
    OPEN INPUT     MENU-FILE.
    OPEN EXTEND    LOGG-FILE.
    ACCEPT EXEC-TIME FROM TIME.
    ACCEPT EXEC-DATE FROM DATE.
    MOVE SPACES          TO MENU-TABEL SPOOL-AREA FUNCTION-TABEL
    END-AREA LINKAGE-MESSAG.
    MOVE ZEROS           TO COUNT-SPOOL CHECK-AREA
    FUNCTION-COUNT LOGIN-COUNT CALL-AREA
    COUNT-MOVE COUNT-SPOOL REWRITE-SW.
    PERFORM DISPLAY-INTERFACE-DESCRIPTION.
    MOVE ZEROS           TO LOGIN-COUNT.
    MOVE 1               TO CHECK-AREA.
    PERFORM DISPLAY-LOGIN UNTIL CHECK-AREA ) 2.
    MOVE ZEROS           TO LOGIN-COUNT CHECK-AREA.
    IF END-AREA = SPACES PERFORM BLACK-MARK-CHECK.
    IF END-AREA = SPACES
    MOVE 1               TO CHECK-AREA
    PERFORM PASSWORD-REQUEST UNTIL CHECK-AREA = 2.
    MOVE ZEROS           TO CHECK-AREA.
    IF END-AREA = SPACES PERFORM TIME-CHECK.
*****************************************************************
*    THE FOLLOWING ROUTINES ARE TO SET UP AND DISPLAY THE MENU TO*
*    THE USER.                                                   *
*****************************************************************
    IF END-AREA = SPACES
    MOVE 9               TO COUNT-JOB
    MOVE SPACES          TO MENU-TABEL
    MOVE 1               TO COUNT-AREA CHECK-AREA
    PERFORM SET-UP-MENU UNTIL CHECK-AREA ) 14
    MOVE ZEROS           TO CHECK-AREA
    PERFORM COMPLETE-THE-MENU
    PERFORM LAST-ACCESS-DISPLAY
    MOVE 1               TO COUNT-JOB
    PERFORM MENU-DISPLAY UNTIL COUNT-JOB ) 24.
MAIN-PARAGRAGH.
    MOVE ZEROS           TO COUNT-JOB.
    MOVE 1               TO CHECK-AREA.
    PERFORM JOB-CHOICE UNTIL CHECK-AREA = 3.
    MOVE ZEROS           TO CHECK-AREA.
    PERFORM GET-THE-MENU-RECORD.
    PERFORM THE-CALL-ROUTINES.
CLEANUP.
    IF REWRITE-SW = ZERO
    ADD 1                TO U-NO-LOGIN
    REWRITE USER-RECORD.
    CLOSE USER-FILE MENU-FILE LOGG-FILE.
    DISPLAY '    '.
    MOVE 1               TO COUNT-SPOOL.



    PERFORM SPOOL-PARAGRAGH UNTIL COUNT-SPOOL ) 14.
SPOOL-PARAGRAGH.
    DISPLAY SPOOL-FILE(COUNT-SPOOL).
    ADD 1                TO COUNT-SPOOL.
    IF SPOOL-FILE(COUNT-SPOOL) = SPACES MOVE 15 TO COUNT-SPOOL.
DISPLAY-INTERFACE-DESCRIPTION.
    DISPLAY '      '.
    MOVE SPACES TO MENU-LINE-A.
    MOVE ALL '*'         TO MENU-LINE-A.
    DISPLAY '      '  MENU-LINE-A.
    MOVE SPACES          TO MENU-LINE-A.
    MOVE '*'             TO MENU-P1 MENU-P6.
    MOVE 'THE FUNCTION OF THIS INTERFACE IS TO CONTROL A'
    TO MENU-P51.
    MOVE 'LL'            TO MENU-P52.
    DISPLAY '      ' MENU-LINE-A.
    MOVE SPACES          TO MENU-P52.
    MOVE 'THE PROCESSING OF THE PROMOTIONAL PUBLISHING S' TO
    MENU-P51.
    MOVE 'YSTEM.'        TO MENU-P52.
    DISPLAY '      ' MENU-LINE-A.
    MOVE ALL '*'         TO MENU-LINE-A.
    DISPLAY '      ' MENU-LINE-A.
    MOVE SPACES          TO MENU-LINE-A.
DISPLAY-LOGIN.
    CALL 'TNOUA'  USING DISPLAY1 COUNT1.
    ACCEPT W-USER-CODE.
    MOVE W-USER-CODE    TO USER-NO.
```

```
     READ USER-FILE INVALID KEY MOVE 1 TO LOGIN-COUNT.
     ADD 1                  TO CHECK-AREA.
     IF LOGIN-COUNT = 1 AND CHECK-AREA ) 2
     DISPLAY
     '*SORRY YOU ARE NOT AUTHORIZED TO USE THIS SYSTEM*'
     MOVE 1                 TO REWRITE-SW
     MOVE 'ZZZZZZ'          TO USER-NO
     MOVE 'ILLEGAL ATTEMPT' TO COMMENT-TYPE
     PERFORM LOGGING-ROUTINE
     MOVE 'END'             TO END-AREA.
     IF LOGIN-COUNT = ZEROS MOVE 3 TO CHECK-AREA.
     MOVE ZEROS             TO LOGIN-COUNT.
 BLACK-MARK-CHECK.
     IF U-BLACKMARK ) 1
     MOVE 'THE SYSTEM IS LOCKED' TO COMMENT-TYPE
     DISPLAY '    '
     DISPLAY
     '*SORRY THE SYSTEM IS LOCKED*'
     PERFORM LOGGING-ROUTINE
     MOVE 'END'             TO END-AREA.
*****************************************************************
*    THIS ROUTINE IS TO LOG THE EVENT OCCURRED INTO THE LOG FILE.*
*****************************************************************
 LOGGING-ROUTINE.
     MOVE USER-NO           TO L-USER-NO.
     MOVE EXEC-DATE         TO LOG-DATE LAST-D-ACCESS.
     MOVE EXEC-TIME         TO LOG-START-TIME LAST-T-ACCESS.
     ACCEPT EXEC-TIME       FROM TIME.
     MOVE EXEC-TIME         TO LOG-END-TIME.
     WRITE LOGG-RECORD.
 PASSWORD-REQUEST.
     CALL 'TNOUA'  USING DISPLAY3 COUNT1.
     CALL 'DUPLX$' USING HALF-DUPLX.
     ACCEPT W-PASSWORD.



     CALL 'DUPLX$' USING FULL-DUPLX.
     IF W-PASSWORD      = U-PASSWORD
     MOVE 2             TO CHECK-AREA.
     IF W-PASSWORD      NOT = U-PASSWORD
     MOVE 1             TO CHECK-AREA
     DISPLAY '    '
     DISPLAY 'INCORRECT PASSWORD'
     PERFORM INCORRECT-PASSWORD.
     IF END-AREA = 'END' MOVE 2 TO CHECK-AREA.
 INCORRECT-PASSWORD.
     ADD 1              TO U-BLACKMARK.
     MOVE 'INCORRECT PASSWORD' TO COMMENT-TYPE.
     PERFORM LOGGING-ROUTINE.
     PERFORM BLACK-MARK-CHECK.
 TIME-CHECK.
     IF EX-HR = START-TIME MOVE 1 TO CHECK-AREA.
     IF EX-HR ) START-TIME AND EX-HR ( END-TIME
     MOVE 1             TO CHECK-AREA.
     IF CHECK-AREA = ZERO PERFORM ILLEGAL-TIME-CHECK.
     MOVE ZERO          TO CHECK-AREA.
 ILLEGAL-TIME-CHECK.
     MOVE 'END'         TO END-AREA.
     ADD 1              TO U-BLACKMARK.
     MOVE 'NOT ALLOWED IN THIS TIME' TO COMMENT-TYPE.
     PERFORM LOGGING-ROUTINE.
     DISPLAY '    '.
     DISPLAY
     '*SORRY YOU CAN NOT USE THE SYSTEM AT THIS TIME*'.
     DISPLAY
     'YOUR TIME OF WORK BETWEEN ' START-TIME ' AND ' END-TIME.
 SET-UP-MENU.
     ADD 1 TO COUNT-JOB.
     MOVE JOB-ALLOWED(COUNT-AREA) TO MENU-KEY JOB-SEQ(COUNT-AREA).
**   READ A RECORD FROM THE MENU FILE
     READ MENU-FILE INVALID KEY  PERFORM MENU-FILE-ERROR.
     MOVE SPACES TO MENU-LINE-A.
     MOVE '*'     TO MENU-P1.
     MOVE COUNT-AREA  TO MENU-P3.
     MOVE '-'   TO MENU-P4.
     MOVE APPLICATION-NAME  TO MENU-P5.
     MOVE '*'           TO MENU-P6.
     MOVE MENU-LINE-A  TO MENU-LINE(COUNT-JOB).
     MOVE SPACES TO MENU-LINE-A.
     ADD 1              TO COUNT-AREA CHECK-AREA.
     IF JOB-ALLOWED(COUNT-AREA) = SPACES
     MOVE 15            TO CHECK-AREA.
 MENU-FILE-ERROR.
     DISPLAY 'INCORRECT DATA ' MENU-KEY 'MENU-CODE'.
     MOVE 'DATA FILES ERROR' TO COMMENT-TYPE.
     PERFORM LOGGING-ROUTINE.
```

```
    PERFORM CLEANUP.
    PERFORM EXIT-PARA.
    PERFORM STOP-RUN.
COMPLETE-THE-MENU.
      SUBTRACT 1 FROM COUNT-AREA.
    MOVE ALL '*'
    TO  MENU-LINE(1).
    MOVE '*' TO MENU-P1  MENU-P6.
    MOVE MENU-LINE-A  TO MENU-LINE(2) MENU-LINE(5)
     MENU-LINE(8) MENU-LINE(9).
    MOVE
    '         COMPUTER  STUDIES  DEPARTMENT



    TO MENU-P5.
    MOVE  MENU-LINE-A TO MENU-LINE(3).
    MOVE
    '  PROMOTIONAL LITERATURE PUBLISHING SYSTEM'
    TO MENU-P5.
    MOVE MENU-LINE-A  TO MENU-LINE(4).
     MOVE '                      MENU'
    TO MENU-P5.
    MOVE  MENU-LINE-A TO MENU-LINE(6).
    MOVE
    '                      ----'
    TO  MENU-P5.
    MOVE MENU-LINE-A  TO MENU-LINE(7).
    ADD 1 TO COUNT-JOB.
    MOVE ALL '*'  TO MENU-LINE(COUNT-JOB).
LAST-ACCESS-DISPLAY.
    MOVE SPACES         TO DATE-TIME-LINE.
    MOVE LAST-DAY       TO DT-1.
    MOVE LAST-MON       TO DT-3.
    MOVE LAST-YER       TO DT-6.
    MOVE '/'            TO DT-2 DT-4.
    MOVE 19             TO DT-5.
    MOVE '   YOUR LAST ACCESS DATE'  TO DT-AREA.
    DISPLAY '    '.
    DISPLAY  DATE-TIME-LINE.
    MOVE SPACES         TO DATE-TIME-LINE.
    MOVE L-HR           TO DT-1.
    MOVE L-MIN          TO DT-3.
    MOVE L-SEC          TO DT-5.
    MOVE ':'            TO DT-2 DT-4.
    MOVE '   YOUR LAST ACCESS TIME' TO DT-AREA.
    DISPLAY '    '.
    DISPLAY  DATE-TIME-LINE.
MENU-DISPLAY.
    DISPLAY '      ' MENU-LINE(COUNT-JOB).
    ADD 1             TO COUNT-JOB.
    IF MENU-LINE(COUNT-JOB) = SPACES
    AND COUNT-JOB ) 10 MOVE 25 TO COUNT-JOB.
JOB-CHOICE.
    DISPLAY '   '.
    CALL 'TNOUA'  USING DISPLAY5 COUNT2.
    ACCEPT   CHOICE-AREA.
    IF CHECK-AREA = 2 MOVE 1 TO CHECK-AREA.
    IF CHOICE-AREA ) COUNT-AREA OR CHOICE-AREA ( 01
    PERFORM INCORRECT-CHOICE.
    IF CHOICE-AREA1 ALPHABETIC
    PERFORM INCORRECT-CHOICE.
    IF CHECK-AREA = 1 MOVE 3 TO CHECK-AREA.
INCORRECT-CHOICE.
    DISPLAY 'INCORRECT CHOICE'.
    MOVE 2                TO CHECK-AREA.
GET-THE-MENU-RECORD.
    MOVE JOB-SEQ(CHOICE-AREA) TO CALL-AREA MENU-KEY.
    READ MENU-FILE INVALID KEY PERFORM MENU-FILE-ERROR.
    MOVE APPLICATION-NAME TO COMMENT-TYPE.
*****************************************************************
*    THE FOLLOWING ROUTINES ARE TO EXECUTE THE SELECTED ENQUIRY  *
*    FROM THE DISPLAYED MENU. EACH ROUTINE REPRESENTS A SPECIFIC *
*    FUNCTION WHICH IS IDENTIFIED IN THE MENU.                   *
*****************************************************************
THE-CALL-ROUTINES.
    IF CALL-AREA = 01 PERFORM MENU-FILE-MAINTENANCE.



    IF CALL-AREA = 02 PERFORM RELATIONS-DESC-MAINTENANCE.
    IF CALL-AREA = 03 PERFORM USER-FILE-MAINTENANCE.
    IF CALL-AREA = 04 PERFORM USERS-ACCOUNT-REPORT.
    IF CALL-AREA = 05 PERFORM INTERFACE-REPORT.
    IF CALL-AREA = 06 PERFORM DATABASE-UPDATE.
    IF CALL-AREA = 07 OR CALL-AREA = 08 OR CALL-AREA = 09
    PERFORM LISTING-PROCESSING.
    IF CALL-AREA = 10 PERFORM DIRECT-USE-OF-OPERATORS.
    IF CALL-AREA = 11 PERFORM RELATION-LIST.
    IF CALL-AREA ) 11 OR CALL-AREA ( 01
```

```
     DISPLAY 'CALL AREA' CALL-AREA 'ERROR'
     MOVE 'END' TO END-AREA.
 MENU-FILE-MAINTENANCE.
     CLOSE MENU-FILE.
     MOVE 'MENU FILE MAINTENANCE STARTED' TO LINKAGE-MESSAG.
     CALL 'MU4384' USING LINKAGE-MESSAG.
     DISPLAY LINKAGE-MESSAG.
     OPEN INPUT MENU-FILE.
     PERFORM LOGGING-ROUTINE.
     PERFORM ANOTHER-JOB-CHOICE.
*****************************************************************
*     THIS ROUTINE IS TO ASK THE USER FOR FURTHER CHOICE.       *
*****************************************************************
 ANOTHER-JOB-CHOICE.
     DISPLAY 'DO YOU WANT TO RUN ANOTHER JOB?'.
     CALL 'TNOUA' USING DISPLAY6 COUNT2.
     ACCEPT YES-NO-AREA.
     ACCEPT EXEC-TIME FROM TIME.
     IF YES-NO-AREA = 'YES' OR YES-NO-AREA = 'Y  '
     PERFORM TIME-CHECK.
     IF YES-NO-AREA = 'NO ' OR YES-NO-AREA = 'N  '
     OR END-AREA = 'END' PERFORM NEW-PASSWORD-ROUTINE.
 NEW-PASSWORD-ROUTINE.
     DISPLAY 'DO YOU WANT TO CHANGE YOUR PASSWORD?'.
     CALL 'TNOUA' USING DISPLAY6 COUNT2.
     ACCEPT YES-NO-AREA.
     IF YES-NO-AREA = 'YES' OR YES-NO-AREA = 'Y  '
     PERFORM NEW-PASSWORD-ENTRY.
     MOVE 'END' TO END-AREA.
 NEW-PASSWORD-ENTRY.
     CALL 'TNOUA' USING DISPLAY7 COUNT3.
     CALL 'DUPLX$' USING HALF-DUPLX.
     ACCEPT W-PASSWORD.
     CALL 'DUPLX$' USING FULL-DUPLX.
     MOVE W-PASSWORD TO U-PASSWORD.
 RELATIONS-DESC-MAINTENANCE.
     MOVE
     'MAINTENANCE OF THE FILE IS STARTED' TO
     LINKAGE-MESSAG.
     CALL 'RU4384' USING LINKAGE-MESSAG.
     DISPLAY LINKAGE-MESSAG.
     PERFORM LOGGING-ROUTINE.
     PERFORM ANOTHER-JOB-CHOICE.
 USER-FILE-MAINTENANCE.
     REWRITE  USER-RECORD.
     CLOSE    USER-FILE.
     MOVE 'USERS ACCOUNT  FILE UPDATE STARTED' TO
     LINKAGE-MESSAG.
     CALL 'UU4384'  USING LINKAGE-MESSAG.
     DISPLAY LINKAGE-MESSAG.
     OPEN I-O USER-FILE.
     MOVE W-USER-CODE TO USER-NO.



     READ USER-FILE INVALID KEY DISPLAY '(ERROR 2)' W-USER-CODE.
     PERFORM LOGGING-ROUTINE.
     PERFORM ANOTHER-JOB-CHOICE.
 USERS-ACCOUNT-REPORT.
     REWRITE USER-RECORD.
     CLOSE USER-FILE MENU-FILE.
     MOVE 'USERS ACCOUNT REPORT STARTED' TO
     LINKAGE-MESSAG.
     CALL 'UR4384' USING LINKAGE-MESSAG.
     DISPLAY LINKAGE-MESSAG.
     ADD 1                TO COUNT-SPOOL.
     MOVE
     'PLEASE ENTER: SPOOL UREP4384 -X1 (AFTER OK)' TO
     SPOOL-FILE(COUNT-SPOOL).
     OPEN I-O USER-FILE.
     OPEN INPUT MENU-FILE.
     MOVE W-USER-CODE TO USER-NO.
     READ USER-FILE INVALID KEY DISPLAY
     '(ERROR 3)' W-USER-CODE.
     PERFORM LOGGING-ROUTINE.
     PERFORM ANOTHER-JOB-CHOICE.
 INTERFACE-REPORT.
     REWRITE USER-RECORD.
     CLOSE USER-FILE LOGG-FILE.
     MOVE 'SORT STARTED ON LOGGING FILE' TO LINKAGE-MESSAG.
     CALL 'SI4384' USING LINKAGE-MESSAG.
     DISPLAY LINKAGE-MESSAG.
      MOVE 'INTERFACE REPORT STARTED' TO LINKAGE-MESSAG.
     CALL 'IR4384' USING LINKAGE-MESSAG.
     DISPLAY LINKAGE-MESSAG.
     ADD 1                TO COUNT-SPOOL.
     MOVE
     'PLEASE ENTER: SPOOL IREP4384 -X1 (AFTER OK)' TO
     SPOOL-FILE(COUNT-SPOOL).
     OPEN I-O USER-FILE.
     MOVE W-USER-CODE TO USER-NO.
```

```
     READ USER-FILE INVALID KEY DISPLAY '(ERROR 4)' W-USER-CODE.
     OPEN OUTPUT LOGG-FILE.
     PERFORM LOGGING-ROUTINE.
     CLOSE LOGG-FILE.
     OPEN EXTEND LOGG-FILE.
     PERFORM ANOTHER-JOB-CHOICE.
 DATABASE-UPDATE.
     MOVE ZERO           TO UPDATE-INDICATOR.
     PERFORM UNION-MINUS-ROUTINE UNTIL UPDATE-INDICATOR = 1.
     DISPLAY 'END OF DATABASE UPDATE'.
     PERFORM LOGGING-ROUTINE.
     PERFORM ANOTHER-JOB-CHOICE.
 UNION-MINUS-ROUTINE.
     MOVE ZERO           TO UPDATE-INDICATOR.
     DISPLAY 'ENTER MAX. 72 CHARACTERS OPERATION DETAIL'.
     ACCEPT OPERATION-MESSAGE.
     MOVE OPERATION-MESSAGE TO TABEL-COMMAND(1).
     IF TABEL-COMMAND1(1) = 'UNION'
     MOVE 2              TO UPDATE-INDICATOR
     CALL 'UNION1' USING LINKAGE-FUNCTION.
     IF TABEL-COMMAND1(1) = 'MINUS'
     MOVE 2              TO UPDATE-INDICATOR
     CALL 'MINUS1' USING LINKAGE-FUNCTION.
     IF UPDATE-INDICATOR = 0
     DISPLAY 'OPERATOR NAME ERROR, YOU CAN NOT CONTINUE'
     DISPLAY OPERATION-MESSAGE.



     IF PROGRAM-COMMENT = 'THE RUN WAS NOT SATISFACTORY'
     DISPLAY 'YOU CAN NOT CONTINUE WITH THIS RUN'
     DISPLAY PROGRAM-COMMENT.
     MOVE SPACES         TO LINKAGE-FUNCTION.
     DISPLAY 'DO YOU WANT MORE UPDATE OPERATIONS'.
     CALL 'TNOUA' USING DISPLAY6 COUNT2.
     ACCEPT YES-NO-AREA.
     IF YES-NO-AREA = 'NO ' OR YES-NO-AREA = 'N  '
     MOVE 1              TO UPDATE-INDICATOR.
     IF YES-NO-AREA = 'YES' OR YES-NO-AREA = 'Y  '
     MOVE ZERO           TO UPDATE-INDICATOR.
 LISTING-PROCESSING.
     OPEN INPUT FUNCTIONS-FILE.
*************************************************************************
* GET ALL RECORDS CONCERNING THE REQUIRED RUN FROM THE FUNCTIONS FILE   *
* AND STORE THEM IN THE FUNCTION TABEL                                  *
*************************************************************************
     MOVE SPACES         TO FUNCTION-TABEL.
     MOVE CALL-AREA      TO FUNCTION-NO1.
     MOVE 1              TO FUNCTION-NO2 COUNT-JOB.
     PERFORM READ-AND-STORE UNTIL COUNT-JOB ) 51.
     MOVE 1              TO COUNT-JOB.
     IF END-AREA = SPACES
     PERFORM RETRIEVAL-OPERATIONS UNTIL COUNT-JOB ) 51.
     MOVE ZEROS          TO COUNT-JOB.
     MOVE SPACES         TO FUNCTION-TABEL.
     CLOSE FUNCTIONS-FILE.
     DISPLAY 'END OF LISTING'.
     PERFORM LOGGING-ROUTINE.
     IF END-AREA = SPACES
     PERFORM ANOTHER-JOB-CHOICE.
 READ-AND-STORE.
     READ FUNCTIONS-FILE INVALID KEY
     MOVE 'END'          TO END-AREA
     MOVE 99             TO COUNT-JOB
     DISPLAY 'COMMAND ERROR IN THE FUNCTIONS FILE' FUNCTION-NO.
     IF COMMAND-PART1 = 'END OF OPERATIONS'
     MOVE 99            .TO COUNT-JOB.
     IF COUNT-JOB ( 51
     MOVE FILE-COMMAND-DETAIL TO TABEL-COMMAND(COUNT-JOB)
     ADD 1               TO COUNT-JOB FUNCTION-NO2.
 RETRIEVAL-OPERATIONS.
     IF TABEL-COMMAND(COUNT-JOB) = SPACES MOVE 99 TO COUNT-JOB.
     IF COUNT-JOB ( 51
     MOVE TABEL-COMMAND(COUNT-JOB) TO OPERATION-MESSAGE
     PERFORM LISTS-PRODUCTION
     ADD 1               TO COUNT-JOB.
 LISTS-PRODUCTION.
     IF TABEL-COMMAND1(COUNT-JOB) = 'SELEC'
     PERFORM SPECIAL-CHECK
     CALL 'SE4384' USING LINKAGE-FUNCTION.
*
     IF TABEL-COMMAND1(COUNT-JOB) = 'PROJE'
     PERFORM SPECIAL-CHECK
     CALL 'PR4384' USING LINKAGE-FUNCTION.
*
     IF TABEL-COMMAND1(COUNT-JOB) = 'JOIN '
     PERFORM SPECIAL-CHECK
     CALL 'JO4384' USING LINKAGE-FUNCTION.
*
     IF TABEL-COMMAND1(COUNT-JOB) = 'COMP '
     PERFORM SPECIAL-CHECK
```

```
     CALL 'COMP84' USING LINKAGE-FUNCTION.
*
     IF   TABEL-COMMAND1(COUNT-JOB) = 'PRINT'
     PERFORM SPECIAL-CHECK
     CALL 'PRINT4' USING LINKAGE-FUNCTION
     PERFORM SPOOL-MESSAGE.
*
     IF TABEL-COMMAND1(COUNT-JOB) = 'SORTN'
     PERFORM SPECIAL-CHECK
     MOVE 'SORT STARTED ON AUTHOR NAME' TO LINKAGE-MESSAG
     CALL 'SORT1N' USING LINKAGE-MESSAG
     DISPLAY LINKAGE-MESSAG.
*
     IF TABEL-COMMAND1(COUNT-JOB) = 'SORTT'
     PERFORM SPECIAL-CHECK
     MOVE 'SORT STARTED ON TITLE DATA' TO LINKAGE-MESSAG
     CALL 'SORT2T' USING LINKAGE-MESSAG
     DISPLAY LINKAGE-MESSAG.
*
     IF OPERATION-INDICATOR = 0
     DISPLAY 'OPERATOR NAME ERROR, YOU CAN NOT CONTINUE'
     DISPLAY OPERATION-MESSAGE
     MOVE 88              TO COUNT-JOB.
*
     IF PROGRAM-COMMENT = 'THE RUN WAS NOT SATISFACTORY'
     DISPLAY 'YOU CAN NOT CONTINUE WITH THIS RUN'
     MOVE 88              TO COUNT-JOB.
     MOVE ZERO            TO OPERATION-INDICATOR.
     MOVE SPACES          TO LINKAGE-FUNCTION.
 SPOOL-MESSAGE.
     ADD 1                TO COUNT-SPOOL.
     MOVE
     'PLEASE ENTER: SPOOL PRINT384 -X1 (AFTER OK)'
     TO SPOOL-FILE(COUNT-SPOOL).
 SPECIAL-CHECK.
     MOVE 1               TO OPERATION-INDICATOR.
*****************************************************************
*    THE DIRECT USE OF THE OPERATORS AND SPECIFIC SUBROUTINES    *
*    WILL ASSIST THE USER TO MANIPULATE THE DATABASE AND OBTAIN  *
*    THE REQUIRED DATA.                                         *
*****************************************************************
 DIRECT-USE-OF-OPERATORS.
     MOVE SPACES          TO FUNCTION-TABEL.
     MOVE 1               TO COUNT-JOB.
     PERFORM ACCEPT-OPERATION-DETAIL UNTIL COUNT-JOB > 51.
     MOVE 1               TO COUNT-JOB.
     PERFORM RETRIEVAL-OPERATIONS UNTIL COUNT-JOB > 51.
     DISPLAY 'END OF DIRECT USE OF THE OPERATIONS'.
     PERFORM LOGGING-ROUTINE.
     IF END-AREA = SPACES
     PERFORM ANOTHER-JOB-CHOICE.
*****************************************************************
*    THIS ROUTINE WILL DISPLAY A NOTE TO THE USER CONCERNING THE *
*    SORT SUBROUTINES AVAILABLE TO THE USER WHO WISHES TO PRODUCE*
*    A SPECIFIC LIST WHICH ARE NOT AVAILABLE IN THE MENU.        *
*****************************************************************
 ACCEPT-OPERATION-DETAIL.
     IF COUNT-JOB = 01
     DISPLAY 'BEWARE PLEASE:'
     DISPLAY
     ' THE SORTN(AUTHOR NAME SORT ALPHABETICALLY)'
     DISPLAY



     ' SUBROUTINE, USE WF1 AS INPUT AND WF2 AS OUTPUT'
     DISPLAY '       '
     DISPLAY
     ' THE SORTT(TITLE SORT ALPHABETICALLY)'
     DISPLAY
     ' SUBROUTINE,USE WF3 AS INPUT AND WF1 AS OUTPUT'
     DISPLAY
     '*********************************************'.
     DISPLAY 'ENTER MAX. 72 CHARACTERS OPERATION DETAIL'.
     ACCEPT OPERATION-MESSAGE.
     MOVE OPERATION-MESSAGE TO TABEL-COMMAND(COUNT-JOB).
     IF TABEL-COMMAND1(COUNT-JOB) = 'END O'
     MOVE SPACES          TO TABEL-COMMAND(COUNT-JOB)
     MOVE 88              TO COUNT-JOB.
     ADD 1                TO COUNT-JOB.
 RELATION-LIST.
     MOVE ZERO            TO UPDATE-INDICATOR.
     DISPLAY 'ENTER MAX. 72 CHARACTERS OPERATION DETAIL'.
     ACCEPT OPERATION-MESSAGE.
     MOVE OPERATION-MESSAGE TO TABEL-COMMAND(1).
     IF TABEL-COMMAND1(1) = 'PRINT'
     CALL 'LIST01' USING LINKAGE-FUNCTION
     MOVE 1               TO UPDATE-INDICATOR.
     IF UPDATE-INDICATOR = 0
     DISPLAY 'OPERATION NAME ERROR'
     DISPLAY OPERATION-MESSAGE.
```

```
           MOVE SPACES          TO LINKAGE-FUNCTION.
           DISPLAY 'END OF RELATION LIST'.
           IF UPDATE-INDICATOR NOT = 0
           ADD 1                TO COUNT-SPOOL
           MOVE
           'PLEASE ENTER: SPOOL GENERALP -X1 (AFTER OK)' TO
           SPOOL-FILE(COUNT-SPOOL).
           MOVE ZERO            TO UPDATE-INDICATOR.
           PERFORM LOGGING-ROUTINE.
           PERFORM ANOTHER-JOB-CHOICE.
```

LISTING OF MU4384 11:43 16 JUN 81

```
       ID DIVISION.
      ******************************************************************
      *    THE FUNCTION OF THIS DIALOGUE IS TO MAINTAIN THE MENU FILE.*
      *    THE DIALOGUE CAN ASSIST THE USER TO CREATE , DELETE AND     *
      *    AMEND THE MENU FILE RECORDS EASILY. THERE WILL BE A         *
      *    SUB-MENU DISPLAYED TO THE USER INCLUDING THESE FUNCTIONS.   *
      *    THE QUESTION-ANSWERING TECHNIQUES ARE ALSO INCLUDED.        *
      ******************************************************************
       PROGRAM-ID. MU4384.
      *THE FUNCTION OF THIS PROGRAM IS TO UPDATE THE MENU FILE.
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
       SOURCE-COMPUTER. PRIME.
       OBJECT-COMPUTER. PRIME.
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
           SELECT MENU-FILE ASSIGN PFMS
           ORGANIZATION INDEXED
           ACCESS MODE RANDOM
           RECORD KEY MENU-KEY
           FILE STATUS FILE-STATUS1.
       DATA DIVISION.
       FILE SECTION.
       FD  MENU-FILE
           LABEL RECORDS STANDARD VALUE OF FILE-ID 'MENU4384'.
       01  MENU-RECORD.
        02 MENU-KEY             PIC XX.
        02 APPLICATION-NAME     PIC X(46).
       WORKING-STORAGE SECTION.
       77  FILE-STATUS1         PIC XX.
       77  COUNT-A              PIC 99.
       77  COUNT1               COMPUTATIONAL VALUE 15.
       77  DISPLAY1             PIC X(15) VALUE 'ENTER CHOICE : '.
       77  COUNT2               COMPUTATIONAL VALUE 47.
       77  DISPLAY2             PIC X(47) VALUE
           'ENTER 2 NUMERIC CHARACTERS  APPLICATION CODE : '.
       77  APP-CODE-AREA        PIC 99.
       77  APP-NAME-AREA        PIC X(46).
       77  DISPLAY3             PIC X(43) VALUE
           'ENTER MAX. 46 CHARACTERS APPLICATION NAME :'.
       77  DISPLAY4             PIC X(15) VALUE
           'DO YOU WANT TO '.
       77  DISPLAY5             PIC X(16) VALUE
           'ENTER YES OR NO:'.
       77  COUNT3               COMPUTATIONAL VALUE 16.
       77  YES-NO-AREA          PIC XXX.
```

```
01  CHOICE-AREA          PIC 9.
01  CHOICE-AREA1 REDEFINES CHOICE-AREA PIC X.
01  FUNCTION-TABEL.
 02 F-AREA                OCCURS 10.
 03 PART1                 PIC X.
 03 PART2                 PIC X(30).
 03 PART3                 PIC X.
LINKAGE SECTION.
77  LINKAGE-MESSAG       PIC X(46).
PROCEDURE DIVISION USING LINKAGE-MESSAG.
START-PARA.
    OPEN I-O MENU-FILE.
    DISPLAY LINKAGE-MESSAG.
DISPLAY-FUNCTION.
    MOVE SPACES           TO FUNCTION-TABEL.
    MOVE ALL '*'          TO F-AREA(1) F-AREA(10).
    MOVE '    MENU  FILE  UPDATE'       TO PART2(2).
    MOVE '    ------------------'       TO PART2(3).
    MOVE ' THE FUNCTIONS AVAILABLE ARE : ' TO PART2(5).
    MOVE '    1-RECORD CREATION. '      TO PART2(7).
    MOVE '    2-RECORD DELETION. '      TO PART2(8).
    MOVE '    3-RECORD AMENDMENT.'      TO PART2(9).
    MOVE ZEROS            TO COUNT-A.
START-DISPLAY.
    ADD 1                 TO COUNT-A.
    IF COUNT-A ) 10 GO TO ENTER-CHOICE.
    MOVE '*'              TO PART1(COUNT-A) PART3(COUNT-A).
    DISPLAY '      ' F-AREA(COUNT-A).
    GO TO START-DISPLAY.
ENTER-CHOICE.
    CALL 'TNOUA' USING DISPLAY1 COUNT1.
    ACCEPT CHOICE-AREA.
    IF CHOICE-AREA1    ALPHABETIC
    DISPLAY 'INCORRECT CHOICE' GO TO ENTER-CHOICE.
    GO TO RECORD-CREATION  RECORD-DELETION
    RECORD-AMENDMENT DEPENDING ON CHOICE-AREA.
    DISPLAY 'INCORRECT CHOICE'.
    GO TO ENTER-CHOICE.
RECORD-CREATION.
    DISPLAY 'RECORD CREATION ROUTINE:'.
    DISPLAY '------------------------'.
ACCEPT-MENU-CODE.
    CALL 'TNOUA' USING DISPLAY2 COUNT2.
    ACCEPT APP-CODE-AREA.
    IF APP-CODE-AREA ) 0 AND APP-CODE-AREA ( 15
    GO TO READ-MENU-FILE.
    DISPLAY 'INCORRECT APPLICATION CODE'.
    GO TO ACCEPT-MENU-CODE.
READ-MENU-FILE.
    MOVE APP-CODE-AREA TO MENU-KEY.
    READ MENU-FILE INVALID KEY GO TO ACCEPT-APP-NAME.
    DISPLAY 'APPLICATION CODE EXIST, TRY TO CORRECT IT'.
    GO TO ASK-FOR-CREAT.
ACCEPT-APP-NAME.
    DISPLAY DISPLAY3.
    ACCEPT APP-NAME-AREA.
    MOVE APP-CODE-AREA TO MENU-KEY.
    MOVE APP-NAME-AREA TO APPLICATION-NAME.
WRITE-R.
    WRITE MENU-RECORD.
    DISPLAY 'RECORD WRITTEN IS:'.
    DISPLAY MENU-RECORD.


ASK-FOR-CREAT.
    DISPLAY DISPLAY4 'CREATE ANOTHER RECORD?'.
    CALL 'TNOUA' USING DISPLAY5 COUNT3.
    ACCEPT YES-NO-AREA.
    IF YES-NO-AREA = 'YES' GO TO ACCEPT-MENU-CODE.
    GO TO ASK-FOR-UPDATE.
RECORD-DELETION.
    DISPLAY 'RECORD DELETION ROUTINE:'.
    DISPLAY '------------------------'.
ACCEPT-DELETE-CODE.
    CALL 'TNOUA' USING DISPLAY2 COUNT2.
    ACCEPT APP-CODE-AREA.
READ-MENU-F.
    MOVE APP-CODE-AREA TO MENU-KEY.
    READ MENU-FILE INVALID KEY GO TO DELETE-ERROR.
    DELETE MENU-FILE RECORD.
    DISPLAY 'RECORD DELETED IS:'.
    DISPLAY MENU-RECORD.
    GO TO ASK-FOR-DELETE.
DELETE-ERROR.
    DISPLAY 'APPLICATION CODE IS NOT EXIST, TRY TO CORRECT IT'.
ASK-FOR-DELETE.
```

```
           DISPLAY DISPLAY4 'DELETE ANOTHER RECORD?'.
           CALL 'TNOUA' USING DISPLAY5 COUNT3.
           ACCEPT YES-NO-AREA.
           IF YES-NO-AREA = 'YES' GO TO ACCEPT-DELETE-CODE.
           GO TO ASK-FOR-UPDATE.
       RECORD-AMENDMENT.
           DISPLAY 'RECORD AMENDMENT ROUTINE:'.
           DISPLAY '-------------------------'.
       ACCEPT-CODE.
           CALL 'TNOUA' USING DISPLAY2 COUNT2.
           ACCEPT APP-CODE-AREA.
       READ-MENU-FL.
           MOVE APP-CODE-AREA TO MENU-KEY.
           READ MENU-FILE INVALID KEY GO TO AMEND-ERROR.
           PERFORM ACCEPT-APP-NAME.
           REWRITE MENU-RECORD.
           DISPLAY 'RECORD AMENDED IS:'.
           DISPLAY MENU-RECORD.
           GO TO ASK-FOR-AMEND.
       AMEND-ERROR.
           PERFORM DELETE-ERROR.
       ASK-FOR-AMEND.
           DISPLAY DISPLAY4 'AMEND ANOTHER RECORD?'.
           CALL 'TNOUA' USING DISPLAY5 COUNT3.
           ACCEPT YES-NO-AREA.
           IF YES-NO-AREA = 'YES' GO TO ACCEPT-CODE.
       ASK-FOR-UPDATE.
           DISPLAY 'DO YOU WANT MORE UPDATE?'.
           CALL 'TNOUA' USING DISPLAY5 COUNT3.
           ACCEPT YES-NO-AREA.
           IF YES-NO-AREA = 'YES' GO TO START-DISPLAY.
       FINAL-ROTINE.
           MOVE 'MENU FILE UPDATE FINISHED' TO LINKAGE-MESSAG.
       CLOSE-FILE.
           CLOSE MENU-FILE.
       EXIT-PARA.
           EXIT PROGRAM.
```

LISTING OF SE4384 11:56 16 JUN 81

```
       ID DIVISION.
       PROGRAM-ID. SE4384.
      *****************************************************************
      *THE FUNCTION OF THIS PROGRAM IS TO SELECT A SPECIFIC RECORDS*
      *FROM AN INPUT FILE AND WRITE IT ON ANOTHER FILE.              *
      *****************************************************************
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
       SOURCE-COMPUTER. PRIME.
       OBJECT-COMPUTER. PRIME.
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
           COPY FLCONT.
       DATA DIVISION.
       FILE SECTION.
       COPY JFDDES.
       WORKING-STORAGE SECTION.
       COPY JWORKA.
       01  DETAIL-AREA.
        02 INPUT-RELATION-NAME.
         03 IN-R                PIC X OCCURS 12.
        02 QUESION-FIELD.
         03 QUESION-AREA        PIC X OCCURS 6.
        02 ACCEPT-FIELD-NAME.
         03 ACCEPT-NAME-CH      PIC X OCCURS 12.
        02 PARAMETER2.
         03 PARA2               PIC X OCCURS 2.
        02 CONSTANT-AREA.
         03 CONSTANT-CH         PIC X OCCURS 14.
        02 OUTPUT-RELATION-NAME.
         03 OUT-R               PIC X OCCURS 12.
       COPY LINKAG.
       PROCEDURE DIVISION USING LINKAGE-FUNCTION.
       CONTROL-ROUTINE.
           PERFORM INITIALIZE.
           PERFORM MAIN-PARAGRAGH UNTIL RECORD1-CH(1) = HIGH-VALUE.
           PERFORM CLEANUP.
       EXIT-PARAG.
           EXIT PROGRAM.
       STOP-RUN.
           STOP RUN.
       INITIALIZE.
           MOVE ZEROS TO INPUT-AREAS OUTPUT-AREAS READ-AREAS
           WRITE-AREAS CLOSE-AREAS COUNT-B RECORDS-COUNT.
           MOVE SPACES TO  WORK-RECORD WORK-RECORD1 WORK-AREA
           WORK-AREA1 ACCEPT-COMMAND DETAIL-AREA INPUT-RELATION.
```

```
           MOVE OPERATION-MESSAGE TO ACCEPT-COMMAND.
           MOVE 8 TO COUNT-A.
           PERFORM PART1-ROT UNTIL COMMAND-CH(COUNT-A) = SPACE.
           MOVE ZEROS TO COUNT-B.
           ADD 1 TO COUNT-A.
           PERFORM PART2-ROT UNTIL COMMAND-CH(COUNT-A) = SPACE.
           MOVE ZEROS TO COUNT-B.
           ADD 1 TO COUNT-A.
           PERFORM PART3-ROT UNTIL COMMAND-CH(COUNT-A) = SPACE.
           MOVE ZEROS TO COUNT-B.
           ADD 1 TO COUNT-A.
           PERFORM PART4-ROT UNTIL COMMAND-CH(COUNT-A) = SPACE.
           MOVE ZEROS TO COUNT-B.
           ADD 1 TO COUNT-A.
           PERFORM PART44-ROT UNTIL COMMAND-CH(COUNT-A) = SPACES.
           MOVE ZEROS TO COUNT-B.
           ADD 1 TO COUNT-A.
           PERFORM PART5-ROT UNTIL COMMAND-CH(COUNT-A) = SPACE.
           DISPLAY 'ENTER ' ACCEPT-FIELD-NAME ' PLEASE:'.
           ACCEPT CONSTANT-FIELD-VALUE.
           MOVE CONSTANT-FIELD-VALUE TO CONSTANT-AREA.
           PERFORM READ-RELATIONS-F1.
           PERFORM READ-RELATIONS-F2.
           MOVE CONSTANT-AREA TO WORK-AREA.
           MOVE 1 TO COUNT-A.
           PERFORM FIELD-NAME-CHECK UNTIL
           IN-RFIELDS(COUNT-A) = SPACES.
           IF PARAMETER1 = ZEROS
           DISPLAY ACCEPT-FIELD-NAME '(INCORRECT NAME)'
           MOVE HIGH-VALUE TO INPUT-RKEY WORK-RECORD1.
           IF INPUT-RKEY NOT = HIGH-VALUE
           PERFORM MOVE-AND-OPEN
           PERFORM READ-FILES
           PERFORM MOVE-TO-WORKRECORD.
       MAIN-PARAGRAGH.
           PERFORM DATA-MOVE.
           PERFORM FINAL-CHECKING.
           PERFORM READ-FILES.
           PERFORM MOVE-TO-WORKRECORD.
       CLEANUP.
           PERFORM RECORDS-COUNT-CHECK.
           IF INPUT-RKEY NOT = HIGH-VALUE REWRITE RELATION-RECORD.
           IF CLOSE-COUNT = 1 PERFORM CLOSE-FILES.
           CLOSE RELATIONS-FILE.
       COPY LASTCH.
      **
       PART1-ROT.
           ADD 1 TO COUNT-B.
           MOVE COMMAND-CH(COUNT-A) TO IN-R(COUNT-B).
           ADD 1 TO COUNT-A.
       PART2-ROT.
           ADD 1 TO COUNT-B.
           MOVE COMMAND-CH(COUNT-A) TO QUESION-AREA(COUNT-B).
           ADD 1 TO COUNT-A.
       PART3-ROT.
           ADD 1 TO COUNT-B.
           MOVE COMMAND-CH(COUNT-A) TO ACCEPT-NAME-CH(COUNT-B).
           ADD 1 TO COUNT-A.
       PART4-ROT.
           ADD 1 TO COUNT-B.
           MOVE COMMAND-CH(COUNT-A) TO PARA2(COUNT-B).
           ADD 1 TO COUNT-A.



       PART44-ROT.
           IF COMMAND-CH(COUNT-A) NOT = ','
           ADD 1 TO COUNT-B
           MOVE COMMAND-CH(COUNT-A) TO CONSTANT-CH(COUNT-B).
           ADD 1 TO COUNT-A.
       PART5-ROT.
           ADD 1 TO COUNT-B.
           MOVE COMMAND-CH(COUNT-A) TO OUT-R(COUNT-B).
           ADD 1 TO COUNT-A.
       READ-RELATIONS-F1.
           OPEN I-O RELATIONS-FILE.
           MOVE INPUT-RELATION-NAME TO RELATION-KEY.
           READ RELATIONS-FILE INVALID KEY DISPLAY RELATION-KEY
           'INCORRECT RELATION NAME'
           MOVE HIGH-VALUE TO WORK-RECORD1 INPUT-RKEY.
           IF INPUT-RKEY NOT = HIGH-VALUE
           MOVE RELATION-RECORD TO INPUT-RELATION.
       READ-RELATIONS-F2.
           MOVE OUTPUT-RELATION-NAME TO RELATION-KEY.
           READ RELATIONS-FILE INVALID KEY DISPLAY RELATION-KEY
           'INCORRECT RELATION NAME'
           MOVE HIGH-VALUE TO WORK-RECORD1 INPUT-RKEY.
```

```
FIELD-NAME-CHECK.
    IF FIELD-WNAME(COUNT-A) = ACCEPT-FIELD-NAME
    MOVE COUNT-A TO PARAMETER1.
    ADD 1 TO COUNT-A.
MOVE-AND-OPEN.
    MOVE INPUT-STRUCTURE TO RELATION-STRUCTURE.
    MOVE 1 TO INPUT-AREA(IN-RCODE) READ-AREA(IN-RCODE)
    CLOSE-AREA(IN-RCODE).
    MOVE RELATION-CODE TO OUT-COUNT.
    MOVE 1 TO CLOSE-AREA(OUT-COUNT).
    SUBTRACT 10  FROM OUT-COUNT.
    MOVE 1 TO OUTPUT-AREA(OUT-COUNT) WRITE-AREA(OUT-COUNT).
    MOVE 1 TO COUNT-D.
    IF WORK-AREA NOT = SPACES
    PERFORM PARAG-1 UNTIL WORK-CH(COUNT-D) = SPACES
    MOVE 1 TO SW-IND
    SUBTRACT 1 FROM COUNT-D.
    IF SW-IND = 0 MOVE ZEROS TO COUNT-D.
    MOVE 0 TO SW-IND.
    PERFORM OPEN-INPUT-FILES.
    PERFORM OPEN-OUTPUT-FILES.
PARAG-1.
    ADD 1 TO COUNT-D.
INPUT-PARAG.
COPY JINPFL.
COPY JOUTFL.
COPY JSREAD.
COPY JSMOVE.
**
DATA-MOVE.
                    MOVE IN-RF1(PARAMETER1) TO COUNT-A
                    MOVE 1 TO COUNT-C
               PERFORM MOVE-TO-WORKAREA1 UNTIL COUNT-C ) COUNT-D.
MOVE-TO-WORKAREA1.
    MOVE RECORD1-CH(COUNT-A) TO WORK-CH1(COUNT-C).
    ADD 1 TO COUNT-A COUNT-C.
FINAL-CHECKING.
    IF PARAMETER2 = '= '  MOVE 2 TO SW-IND PERFORM EQUAL-CHECK.
    IF PARAMETER2 = '( ' MOVE 2 TO SW-IND PERFORM LESS-CHECK.
    IF PARAMETER2 = ') ' MOVE 2 TO SW-IND PERFORM GREATER-CHECK.



    IF PARAMETER2 = '(=' OR PARAMETER2 = '=('
    MOVE 2 TO SW-IND PERFORM L-E-CHECK.
    IF PARAMETER2 = ')=' OR PARAMETER2 = '=)'
    MOVE 2 TO SW-IND PERFORM G-E-CHECK.
    IF SW-IND = 1 MOVE WORK-RECORD1 TO WORK-RECORD
    ADD 1                 TO RECORDS-COUNT
    PERFORM WRITE-OUTPUT-RECORD.
    IF SW-IND = 0 DISPLAY 'LOGICAL OPERATORS ERROR'
                  MOVE HIGH-VALUE TO WORK-RECORD1.
    MOVE 0 TO SW-IND.
EQUAL-CHECK.
    IF WORK-AREA1 = WORK-AREA MOVE 1 TO SW-IND.
LESS-CHECK.
    IF WORK-AREA1 ( WORK-AREA MOVE 1 TO SW-IND.
GREATER-CHECK.
    IF WORK-AREA1 ) WORK-AREA MOVE 1 TO SW-IND.
L-E-CHECK.
    IF WORK-AREA1 ( WORK-AREA OR WORK-AREA1 = WORK-AREA
    MOVE 1 TO SW-IND.
G-E-CHECK.
    IF WORK-AREA1 ) WORK-AREA OR WORK-AREA1 = WORK-AREA
    MOVE 1 TO SW-IND.
COPY-WRITE.
COPY JWRITE.
COPY JCLOSE.
```

```
       ID DIVISION.
       PROGRAM-ID. PR4384.
      ***************************************************************
      *THE FUNCTION OF THIS PROGRAM IS TO PROJECT A SPECIFIC FIELDS    *
      *FROM AN INPUT RELATION AND WRITES IT ON ANOTHER RELATION.       *
      ***************************************************************
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
       SOURCE-COMPUTER. PRIME.
       OBJECT-COMPUTER. PRIME.
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
           COPY FLCONT.
       DATA DIVISION.
       FILE SECTION.
       COPY JFDDES.
       WORKING-STORAGE SECTION.
       COPY JWORKA.
       01  DETAIL-AREA.
        02 INPUT-RELATION-NAME.
        03 IN-NAME              PIC X OCCURS 12.
        02 PARAMETERS-GROUP.
        03 PARAMETR1          PIC 99 OCCURS 15.
        02 PARAMETERS-GROUP1 REDEFINES PARAMETERS-GROUP.
        03 PARAMETER-DATA      PIC X OCCURS 30.
        02 OUTPUT-RELATION-NAME.
        03 OUT-NAME             PIC X OCCURS 12.
       COPY LINKAG.
       PROCEDURE DIVISION USING LINKAGE-FUNCTION.
       CONTROL-ROUTINE.
           PERFORM INITIAL-PARAGRAGH.
           PERFORM MAIN-PARAGRAGH UNTIL RECORD1-CH(1) = HIGH-VALUE.
           PERFORM CLEANUP-PARAGRAGH.
       EXIT-PARAG.
           EXIT PROGRAM.
       STOP-RUN.
           STOP RUN.
       INITIAL-PARAGRAGH.
           MOVE ZEROS TO INPUT-AREAS OUTPUT-AREAS READ-AREAS
                WRITE-AREAS CLOSE-AREAS PARAMETERS-GROUP RECORDS-COUNT.
           MOVE SPACES TO WORK-RECORD WORK-RECORD1 ACCEPT-COMMAND
           INPUT-RELATION INPUT-RELATION-NAME OUTPUT-RELATION-NAME.
           MOVE OPERATION-MESSAGE TO ACCEPT-COMMAND.
           MOVE 9 TO COUNT-A.
           MOVE ZEROS TO COUNT-B.
           PERFORM PART1-ROT UNTIL COMMAND-CH(COUNT-A) = SPACE.



           MOVE ZEROS TO COUNT-B.
           ADD 1 TO COUNT-A.
           PERFORM PART2-ROT UNTIL COMMAND-CH(COUNT-A) = SPACE.
           MOVE ZEROS TO COUNT-B.
           ADD 1 TO COUNT-A.
           PERFORM PART3-ROT UNTIL COMMAND-CH(COUNT-A) = SPACE.
           PERFORM READ-RELATIONS-F1.
           PERFORM READ-RELATIONS-F2.
           IF INPUT-RKEY NOT = HIGH-VALUE
           PERFORM MOVE-AND-OPEN
           PERFORM READ-FILES
           PERFORM MOVE-TO-WORKRECORD
           PERFORM CLEAN-AREAS.
       MAIN-PARAGRAGH.
           MOVE SPACES TO WORK-RECORD.
           PERFORM CLEAN-AREAS.
           PERFORM DATA-MOVE UNTIL PARAMETR1(COUNT-A) = ZEROS.
           ADD 1                 TO RECORDS-COUNT.
           PERFORM WRITE-OUTPUT-RECORD.
           PERFORM READ-FILES.
           PERFORM MOVE-TO-WORKRECORD.
       CLEANUP-PARAGRAGH.
           PERFORM RECORDS-COUNT-CHECK.
           IF INPUT-RKEY NOT = HIGH-VALUE REWRITE RELATION-RECORD.
           IF CLOSE-COUNT = 1 PERFORM CLOSE-FILES.
           CLOSE RELATIONS-FILE.
       COPY LASTCH.
      **
       PART1-ROT.
           ADD 1 TO COUNT-B.
           MOVE COMMAND-CH(COUNT-A) TO IN-NAME(COUNT-B).
           ADD 1 TO COUNT-A.
       PART2-ROT.
           ADD 1 TO COUNT-B.
           IF COMMAND-CH(COUNT-A) = 'F' OR COMMAND-CH(COUNT-A) = ','
           ADD 1 TO COUNT-A.
           IF COMMAND-CH(COUNT-A) = ',' OR COMMAND-CH(COUNT-A) = 'F'
```

```
    ADD 1 TO COUNT-A.
    MOVE COMMAND-CH(COUNT-A) TO PARAMETER-DATA(COUNT-B).
    IF COMMAND-CH(COUNT-A) = 'E'
    MOVE 0 TO PARAMETER-DATA(COUNT-B).
    ADD 1 TO COUNT-A.
PART3-ROT.
    ADD 1 TO COUNT-B.
    MOVE COMMAND-CH(COUNT-A) TO OUT-NAME(COUNT-B).
    ADD 1 TO COUNT-A.
READ-RELATIONS-F1.
    OPEN I-O RELATIONS-FILE.
    MOVE INPUT-RELATION-NAME TO RELATION-KEY.
    READ RELATIONS-FILE INVALID KEY DISPLAY RELATION-KEY
    'INCORRECT RELATION-NAME'
    MOVE HIGH-VALUE TO WORK-RECORD1 INPUT-RKEY.
    IF INPUT-RKEY NOT = HIGH-VALUE
    MOVE RELATION-RECORD TO INPUT-RELATION.
READ-RELATIONS-F2.
    MOVE OUTPUT-RELATION-NAME TO RELATION-KEY.
    READ RELATIONS-FILE INVALID KEY DISPLAY RELATION-KEY
    'INCORRECT RELATION NAME'
    MOVE HIGH-VALUE TO WORK-RECORD1 INPUT-RKEY.
    MOVE SPACES TO RELATION-STRUCTURE.
MOVE-AND-OPEN.
    MOVE 1 TO INPUT-AREA(IN-RCODE) READ-AREA(IN-RCODE)



    CLOSE-AREA(IN-RCODE).
    MOVE RELATION-CODE TO OUT-COUNT.
    MOVE 1 TO CLOSE-AREA(OUT-COUNT).
    SUBTRACT 10 FROM OUT-COUNT.
    MOVE 1 TO OUTPUT-AREA(OUT-COUNT) WRITE-AREA(OUT-COUNT).
    MOVE 1 TO COUNT-A COUNT-E.
    MOVE ZEROS TO COUNT-B.
    PERFORM STRUCTURE-MOVE UNTIL
    PARAMETR1(COUNT-A) = ZEROS.
    PERFORM OPEN-INPUT-FILES.
    PERFORM OPEN-OUTPUT-FILES.
STRUCTURE-MOVE.
    ADD 1 TO COUNT-B.
    MOVE PARAMETR1(COUNT-A) TO COUNT-D.
    MOVE FIELD-WNAME(COUNT-D) TO NAME-OF-FIELD(COUNT-B).
    MOVE COUNT-E TO FIELD1(COUNT-B).
    MOVE IN-RF2(COUNT-D) TO FIELD2(COUNT-B).
    ADD IN-RF2(COUNT-D) TO COUNT-E.
    ADD 1 TO COUNT-A.
INPUT-OUTPUT-PARAG.
COPY JINPFL.
COPY JOUTFL.
COPY JSREAD.
COPY JSMOVE.
**
CLEAN-AREAS.
    MOVE ZEROS TO COUNT-A COUNT-B COUNT-C COUNT-D COUNT-F
                  COUNT-G COUNT-H.
    MOVE 1 TO COUNT-A.
DATA-MOVE.
    MOVE PARAMETR1(COUNT-A) TO COUNT-D.
    MOVE IN-RF2(COUNT-D) TO COUNT-C.
    MOVE IN-RF1(COUNT-D) TO COUNT-F.
    MOVE 1 TO COUNT-H.
    PERFORM MOVE-CH UNTIL COUNT-H ) COUNT-C.
    MOVE ZERO          TO END-FIELD-COUNT.
    ADD 1 TO COUNT-A.
MOVE-CH.
    ADD 1 TO COUNT-G.
    MOVE RECORD1-CH(COUNT-F) TO RECORD-CH(COUNT-G).
    IF FIELD-WNAME(COUNT-D) NOT = 'AUTH-NAME    '
    PERFORM END-OF-FIELD-CHECK.
    ADD 1 TO COUNT-H COUNT-F.
END-OF-FIELD-CHECK.
    IF RECORD1-CH(COUNT-F) = SPACE ADD 1 TO END-FIELD-COUNT.
    IF RECORD1-CH(COUNT-F) NOT = SPACE
    MOVE ZERO          TO END-FIELD-COUNT.
    IF END-FIELD-COUNT ) 1 MOVE 0 TO END-FIELD-COUNT
    SUBTRACT COUNT-H FROM COUNT-C GIVING FINAL-RESULT-COUNT
    ADD FINAL-RESULT-COUNT TO COUNT-G
    MOVE COUNT-C       TO COUNT-H.
    COPY JWRITE.
    COPY JCLOSE.
```

```
       ID DIVISION.
       PROGRAM-ID. JO4384.
      ******************************************************************
      *THE FUNCTION OF THIS PROGRAM IS TO JOIN RECORDS OF TWO INPUT    *
      *RELATIONS AND WRITES IT ON ANOTHER RELATION.                    *
      ******************************************************************
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
       SOURCE-COMPUTER. PRIME.
       OBJECT-COMPUTER. PRIME.
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
           COPY FLCONT.
       DATA DIVISION.
       FILE SECTION.
       COPY JFDDES.
       WORKING-STORAGE SECTION.
       COPY JWORKA.
       01  DETAIL-AREA.
        02 INPUT-RELATION-NAME.
         03 INPUT-NAME              PIC X OCCURS 12.
        02 AND-AREA.
         03 AND-CH                  PIC X OCCURS 3.
        02 INPUT2-RELATION-NAME.
         03 INPUT2-NAME             PIC X OCCURS 12.
        02 OVER-AREA.
         03 OVER-CH                 PIC X OCCURS 4.
        02 KEY-AREA.
         03 KEY-CH                  PIC X OCCURS 12.
        02 KEY2-AREA.
        03 KEY2-CH                  PIC X OCCURS 12.
        02 OUTPUT-RELATION-NAME.
         03 OUTPUT-NAME             PIC X OCCURS 12.
       COPY LINKAG.
       PROCEDURE DIVISION USING LINKAGE-FUNCTION.
       CONTROL-ROUTINE.
           PERFORM INITIAL-PARAGRAGH THRU INITIAL2-PARAG.
           PERFORM MAIN-PARAGRAGH UNTIL END-AREA = HIGH-VALUE.
           PERFORM CLEANUP-PARAGRAGH.
       EXIT-PARAG.
           EXIT PROGRAM.
       STOP-RUN.
           STOP RUN.
       COPY JBEGIN.
       INITIAL2-PARAG.
           IF END-AREA = HIGH-VALUE NEXT SENTENCE



                       ELSE
               PERFORM STRUCTURE-MOVE
               PERFORM FIRST-IND-MOVE
               PERFORM OPEN-INPUT-FILES
               PERFORM OPEN-OUTPUT-FILES
               PERFORM READ-FILES
               PERFORM MOVE-TO-WORKRECORD
               MOVE WORK-RECORD1 TO WORK-RECORD
               PERFORM CLEAN-COUNTS
               PERFORM WORK-AREA-MOVE UNTIL
               KEY1-VALUE(COUNT-A) = ZEROS.
       MAIN-PARAGRAGH.
           PERFORM SECOND-IND-MOVE.
           PERFORM READ-FILES.
           PERFORM MOVE-TO-WORKRECORD.
           PERFORM CLEAN-COUNTS.
           PERFORM WORK-AREA1-MOVE UNTIL
           KEY2-VALUE(COUNT-A) = ZEROS.
           PERFORM MOVE-AND-WRITE UNTIL RECORD1-CH(1) = HIGH-VALUE.
           PERFORM CLOSE-FILES.
           PERFORM OPEN-INPUT-FILES.
           PERFORM THIRD-IND-MOVE.
           PERFORM READ-FILES.
           PERFORM MOVE-TO-WORKRECORD.
           MOVE WORK-RECORD1 TO WORK-RECORD.
           PERFORM CLEAN-COUNTS.
           PERFORM WORK-AREA-MOVE UNTIL
           KEY1-VALUE(COUNT-A) = ZEROS.
           IF RECORD1-CH(1) = HIGH-VALUE
           PERFORM LAST-IND-MOVE
           MOVE HIGH-VALUE TO END-AREA.
       CLEANUP-PARAGRAGH.
           PERFORM RECORDS-COUNT-CHECK.
           IF RECORD1-CH(1) = HIGH-VALUE REWRITE RELATION-RECORD.
           IF CLOSE-COUNT = 1 PERFORM CLOSE-FILES.
           CLOSE RELATIONS-FILE.
       COPY LASTCH.
       COPY JCNAME.
```

```
STRUCTURE-MOVE.
     PERFORM CLEAN-COUNTS.
     MOVE INPUT-STRUCTURE TO RELATION-STRUCTURE.
     PERFORM FIND-SPACES UNTIL RELATION-DATA(COUNT-A) = SPACES.
     MOVE COUNT-A TO COUNT-B.
     SUBTRACT 1 FROM COUNT-B.
     MOVE 1 TO COUNT-C.
     PERFORM MOVE-RELATION-FIELDS UNTIL
             INPUT2-RFIELDS(COUNT-C) = SPACES.
CLEAN-COUNTS.
     MOVE ZEROS TO COUNT-B COUNT-C COUNT-D COUNT-E COUNT-F
                   COUNT-G COUNT-H.
     MOVE 1 TO COUNT-A.
FIND-SPACES.
     ADD FIELD1(COUNT-A) FIELD2(COUNT-A) GIVING WRITE-COUNT.
     ADD 1 TO COUNT-A.
MOVE-RELATION-FIELDS.
     MOVE INPUT2-FNAME(COUNT-C) TO NAME-OF-FIELD(COUNT-A).
     ADD FIELD1(COUNT-B) FIELD2(COUNT-B) GIVING COUNT-E.
     MOVE COUNT-E TO FIELD1(COUNT-A).
     MOVE INPUT2-FIELD2(COUNT-C) TO FIELD2(COUNT-A).
     ADD INPUT2-FIELD1(COUNT-C) INPUT2-FIELD2(COUNT-C)
     GIVING READ-COUNT.
     ADD 1 TO COUNT-A COUNT-B COUNT-C.



FIRST-IND-MOVE.
     MOVE 1 TO INPUT-AREA(IN-RCODE) INPUT-AREA(INPUT2-RCODE).
     MOVE 1 TO READ-AREA(IN-RCODE).
     MOVE 1 TO CLOSE-AREA(INPUT2-RCODE).
     MOVE RELATION-CODE TO OUT-COUNT.
     SUBTRACT 10 FROM OUT-COUNT.
     MOVE 1 TO WRITE-AREA(OUT-COUNT) OUTPUT-AREA(OUT-COUNT).
OPEN-PARAGRAGH.
COPY JINPFL.
COPY JOUTFL.
COPY JSREAD.
COPY JSMOVE.
*
WORK-AREA-MOVE.
     MOVE KEY1-VALUE(COUNT-A) TO COUNT-F.
     MOVE IN-PF1(COUNT-F) TO COUNT-G.
     MOVE 1 TO COUNT-B.
     PERFORM KEY-MOVE UNTIL COUNT-B ) IN-RF2(COUNT-F).
     ADD 1 TO COUNT-A.
KEY-MOVE.
     ADD 1 TO COUNT-C.
     MOVE RECORD-CH(COUNT-G) TO WORK-CH(COUNT-C).
     ADD 1 TO COUNT-G COUNT-B.
WORK-AREA1-MOVE.
     MOVE KEY2-VALUE(COUNT-A) TO COUNT-F.
     MOVE INPUT2-FIELD1(COUNT-F) TO COUNT-G.
     MOVE 1 TO COUNT-B.
     PERFORM KEY2-MOVE UNTIL COUNT-B ) INPUT2-FIELD2(COUNT-F).
     ADD 1 TO COUNT-A.
KEY2-MOVE.
     ADD 1 TO COUNT-C.
     MOVE RECORD1-CH(COUNT-G) TO WORK-CH1(COUNT-C).
     ADD 1 TO COUNT-G COUNT-B.
SECOND-IND-MOVE.
     MOVE 0 TO READ-AREA(IN-RCODE) INPUT-AREA(IN-RCODE).
     MOVE 1 TO READ-AREA(INPUT2-RCODE).
*
MOVE-AND-WRITE.
     PERFORM CLEAN-COUNTS.
     IF WORK-AREA = WORK-AREA1
     MOVE WRITE-COUNT TO COUNT-H
     PERFORM DATA-MOVE UNTIL COUNT-A = READ-COUNT
     ADD 1                  TO RECORDS-COUNT
     PERFORM WRITE-OUTPUT-RECORD.
     PERFORM READ-FILES.
     PERFORM MOVE-TO-WORKRECORD.
     PERFORM CLEAN-COUNTS.
     PERFORM WORK-AREA1-MOVE UNTIL
     KEY2-VALUE(COUNT-A) =  ZEROS.
DATA-MOVE.
     MOVE RECORD1-CH(COUNT-A) TO RECORD-CH(COUNT-H).
     ADD 1 TO  COUNT-A COUNT-H.
*
COPY JWRITE.
COPY JCLOSE.
THIRD-IND-MOVE.
     MOVE 1 TO READ-AREA(IN-RCODE).
     MOVE 0 TO READ-AREA(INPUT2-RCODE).
LAST-IND-MOVE.
     MOVE 1 TO CLOSE-AREA(IN-RCODE) CLOSE-AREA(RELATION-CODE).
```

LISTING OF COMP84 18:50 27 MAY 81

```
       ID DIVISION.
       PROGRAM-ID. COMP84.
      *****************************************************************
      *THE FUNCTION OF THIS PROGRAM IS TO CREAT A DUMMY RECORDS WHICH  *
      * IS NOT EXIST IN THE FIRST RELATION.                            *
      *****************************************************************
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
       SOURCE-COMPUTER. PRIME.
       OBJECT-COMPUTER. PRIME.
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
           COPY FLCONT.
       DATA DIVISION.
       FILE SECTION.
       COPY JFDDES.
       WORKING-STORAGE SECTION.
       COPY JWORKA.
       01  DETAIL-AREA.
        02 INPUT-RELATION-NAME.
         03 INPUT-NAME              PIC X OCCURS 12.
        02 AND-AREA.
         03 AND-CH                  PIC X OCCURS 3.
        02 INPUT2-RELATION-NAME.
         03 INPUT2-NAME             PIC X OCCURS 12.
        02 OVER-AREA.
         03 OVER-CH                 PIC X OCCURS 4.
        02 KEY-AREA.
         03 KEY-CH                  PIC X OCCURS 12.
        02 KEY2-AREA.
         03 KEY2-CH                 PIC X OCCURS 12.
        02 OUTPUT-RELATION-NAME.
         03 OUTPUT-NAME             PIC X OCCURS 12.
       COPY LINKAG.
       PROCEDURE DIVISION USING LINKAGE-FUNCTION.
       CONTROL-ROUTINE.
           PERFORM INITIAL-PARAGRAGH THRU INITIAL2-PARAG.
           PERFORM MAIN-PARAGRAGH UNTIL END-AREA = HIGH-VALUE.
           PERFORM CLEANUP-PARAGRAGH.
       EXIT-PARAG.
           EXIT PROGRAM.
       STOP-RUN.
           STOP RUN.
       COPY JBEGIN.
       INITIAL2-PARAG.
           IF END-AREA = HIGH-VALUE NEXT SENTENCE



                       ELSE
               PERFORM STRUCTURE-MOVE
               PERFORM FIRST-IND-MOVE
               PERFORM OPEN-INPUT-FILES
               PERFORM OPEN-OUTPUT-FILES
               PERFORM FIRST-FILE-READ
               PERFORM SECOND-FILE-READ.
       MAIN-PARAGRAGH.
           ADD 1                  TO RECORDS-COUNT.
           IF WORK-AREA1 = WORK-AREA
               MOVE WORK-RECORD2 TO WORK-RECORD
               PERFORM WRITE-OUTPUT-RECORD
               PERFORM FIRST-FILE-READ
               PERFORM SECOND-FILE-READ.
           IF WORK-AREA ) WORK-AREA1
               MOVE SPACES TO WORK-RECORD
               PERFORM CREAT-DUMMY-RECORD
               PERFORM WRITE-OUTPUT-RECORD
               PERFORM SECOND-FILE-READ.
           IF WORK-AREA ( WORK-AREA1
               PERFORM  FIRST-FILE-READ.
           IF RECORD2-CH(1) = HIGH-VALUE AND RECORD3-CH(1) = HIGH-VALUE
               MOVE HIGH-VALUE TO END-AREA.
       CLEANUP-PARAGRAGH.
           PERFORM RECORDS-COUNT-CHECK.
           IF RECORD1-CH(1) = HIGH-VALUE REWRITE RELATION-RECORD.
           IF CLOSE-COUNT = 1 PERFORM CLOSE-FILES.
           CLOSE RELATIONS-FILE.
       COPY LASTCH.
       OTHER-PARAGRAGH.
       COPY JCNAME.
      **
```

```
 STRUCTURE-MOVE.
     MOVE INPUT-STRUCTURE TO RELATION-STRUCTURE.
 FIRST-IND-MOVE.
     MOVE 1 TO INPUT-AREA(IN-RCODE) INPUT-AREA(INPUT2-RCODE).
     MOVE 1 TO CLOSE-AREA(IN-RCODE) CLOSE-AREA(INPUT2-RCODE)
               CLOSE-AREA(RELATION-CODE).
     MOVE RELATION-CODE TO OUT-COUNT.
     SUBTRACT 10 FROM OUT-COUNT.
     MOVE 1 TO WRITE-AREA(OUT-COUNT) OUTPUT-AREA(OUT-COUNT).
 OPEN-PARAGRAGH.
*
 COPY JINPFL.
 COPY JOUTFL.
**
 FIRST-FILE-READ.
     MOVE 1 TO READ-AREA(IN-RCODE).
     MOVE 0 TO READ-AREA(INPUT2-RCODE).
     PERFORM READ-FILES.
     PERFORM MOVE-TO-WORKRECORD.
     MOVE WORK-RECORD1 TO WORK-RECORD2.
     PERFORM CLEAN-COUNTS.
     PERFORM WORK-AREA-MOVE UNTIL KEY1-VALUE(COUNT-A) = ZEROS.
 SECOND-FILE-READ.
     MOVE 0 TO READ-AREA(IN-RCODE).
     MOVE 1 TO READ-AREA(INPUT2-RCODE).
     PERFORM READ-FILES.
     PERFORM MOVE-TO-WORKRECORD.
     MOVE WORK-RECORD1 TO WORK-RECORD3.
     PERFORM CLEAN-COUNTS.
     PERFORM WORK-AREA1-MOVE UNTIL KEY2-VALUE(COUNT-A) = ZEROS.



*
 READ-MOVE-PARAGRAGH.
 COPY JSREAD.
 COPY JSMOVE.
**
 CLEAN-COUNTS.
     MOVE ZEROS TO COUNT-B COUNT-C COUNT-D COUNT-E COUNT-F
                   COUNT-G COUNT-H.
     MOVE 1 TO COUNT-A.
 WORK-AREA-MOVE.
     MOVE KEY1-VALUE(COUNT-A) TO COUNT-F.
     MOVE IN-RF1(COUNT-F) TO COUNT-G.
     MOVE 1 TO COUNT-B.
     PERFORM KEY-MOVE UNTIL COUNT-B ) IN-RF2(COUNT-F).
     ADD 1 TO COUNT-A.
 KEY-MOVE.
     ADD 1 TO COUNT-C.
     MOVE RECORD2-CH(COUNT-G) TO WORK-CH(COUNT-C).
     ADD 1 TO COUNT-G COUNT-B.
*
 WORK-AREA1-MOVE.
     MOVE KEY2-VALUE(COUNT-A) TO COUNT-F.
     MOVE INPUT2-FIELD1(COUNT-F) TO COUNT-G.
     MOVE 1 TO COUNT-B.
     PERFORM KEY2-MOVE UNTIL COUNT-B ) INPUT2-FIELD2(COUNT-F).
     ADD 1 TO COUNT-A.
 KEY2-MOVE.
     ADD 1 TO COUNT-C.
     MOVE RECORD3-CH(COUNT-G) TO WORK-CH1(COUNT-C).
     ADD 1 TO COUNT-G COUNT-B.
*
 CREAT-DUMMY-RECORD.
     PERFORM CLEAN-COUNTS.
     PERFORM DUMMY-MOVE UNTIL KEY2-VALUE(COUNT-A) = ZEROS.
 DUMMY-MOVE.
     MOVE KEY2-VALUE(COUNT-A) TO COUNT-F.
     MOVE INPUT2-FIELD1(COUNT-F) TO COUNT-G.
     MOVE 1 TO COUNT-B.
     MOVE KEY1-VALUE(COUNT-A) TO COUNT-D.
     MOVE IN-RF1(COUNT-D) TO COUNT-E.
     PERFORM MOVE-ALL UNTIL COUNT-B ) INPUT2-FIELD2(COUNT-F).
     ADD 1 TO COUNT-A.
 MOVE-ALL.
     MOVE RECORD3-CH(COUNT-G) TO RECORD-CH(COUNT-E).
     ADD 1 TO COUNT-B COUNT-G COUNT-E.
*
 COPY JWRITE.
 COPY JCLOSE.
```

```
LISTING OF UNION1  18:49  27 MAY 81

       ID DIVISION.
       PROGRAM-ID. UNION1.
      ***************************************************************
      *THE FUNCTION OF THIS PROGRAM IS TO INSERT RECORD(S) FROM THE 1ST*
      * RELATION INTO THE 2ND RELATION PROVIDS THAT THESE RECORD(S) ARE*
      *NOT EXIST IN THE SECOND RELATION.                             *
      ***************************************************************
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
       SOURCE-COMPUTER. PRIME.
       OBJECT-COMPUTER. PRIME.
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
           COPY FCONT1.
       DATA DIVISION.
       FILE SECTION.
       COPY JFDDES.
       WORKING-STORAGE SECTION.
       COPY JWORK1.
       01  DETAIL-AREA.
        02 INPUT-RELATION-NAME.
         03 INPUT-NAME              PIC X OCCURS 12.
        02 AND-AREA.
         03 AND-CH                  PIC X OCCURS 4.
        02 INPUT2-RELATION-NAME.
         03 INPUT2-NAME             PIC X OCCURS 12.
       COPY LINKAG.
       PROCEDURE DIVISION USING LINKAGE-FUNCTION.
       CONTROL-ROUTINE.
           PERFORM INITIAL-PARAGRAGH THRU INITIAL2-PARAG.
           PERFORM MAIN-PARAGRAGH UNTIL END-AREA = HIGH-VALUE.
           PERFORM CLEANUP-PARAGRAGH.
       EXIT-PARAG.
           EXIT PROGRAM.
       STOP-RUN.
           STOP RUN.
       COPY BEGIN1.
       INITIAL2-PARAG.
           IF END-AREA = HIGH-VALUE NEXT SENTENCE
                       ELSE
           PERFORM MOVE-TO-INDICATORS
           PERFORM OPEN-INPUT-FILES
           PERFORM READ-FILES.
       MAIN-PARAGRAGH.
           ADD 1                TO RECORDS-COUNT.
           PERFORM MOVE-TO-MAIN-RELATION.



           DISPLAY 'RECORD TO BE INSERTED IS:'.
           DISPLAY WORK-RECORD.
           PERFORM WRITE-RELATION-RECORD.
           PERFORM READ-FILES.
           IF RECORD-CH(1) = HIGH-VALUE MOVE HIGH-VALUE TO END-AREA.
       CLEANUP-PARAGRAGH.
           PERFORM RECORDS-COUNT-CHECK.
           IF CLOSE-COUNT = 1 PERFORM CLOSE-FILES.
           CLOSE RELATIONS-FILE.
       COPY LASTCH.
       OTHER-PARAGRAGH.
       COPY JNAME1.
      **
       MOVE-TO-INDICATORS.
           MOVE 1 TO INPUT-AREA(IN-RCODE) INPUT-AREA(INPUT2-RCODE).
           MOVE 1 TO CLOSE-AREA(IN-RCODE) CLOSE-AREA(INPUT2-RCODE).
           MOVE IN-RCODE        TO OUT-COUNT.
           SUBTRACT 10 FROM OUT-COUNT.
           MOVE 1               TO READ-AREA(OUT-COUNT).
       OPEN-PARAGRAGH.
      *
       COPY INPFL1.
      **
```

```
READ-MOVE-PARAGRAGH.
READ-FILES.
    IF READ-AREA(1) = 01 PERFORM READ-WORK-FILE1.
    IF READ-AREA(2) = 02 PERFORM READ-WORK-FILE2.
    IF READ-AREA(3) = 03 PERFORM READ-WORK-FILE3.
READ-WORK-FILE1.
    READ WORKBOOK1-F AT END MOVE HIGH-VALUE TO WORKBOOK1-RECORD.
    MOVE WORKBOOK1-RECORD TO WORK-RECORD.
READ-WORK-FILE2.
    READ WORKBOOK2-F AT END MOVE HIGH-VALUE TO WORKBOOK2-RECORD.
    MOVE WORKBOOK2-RECORD TO WORK-RECORD.
READ-WORK-FILE3.
    READ WORKBOOK3-F AT END MOVE HIGH-VALUE TO WORKBOOK3-RECORD.
    MOVE WORKBOOK3-RECORD TO WORK-RECORD.
MOVE-TO-MAIN-RELATION.
    IF INPUT2-RCODE = 01 MOVE WORK-RECORD TO CATEGORY-RECORD.
    IF INPUT2-RCODE = 02 MOVE WORK-RECORD TO AUTHORTITLE-RECORD.
    IF INPUT2-RCODE = 03 MOVE WORK-RECORD TO CONFERENCE-RECORD.
    IF INPUT2-RCODE = 04 MOVE WORK-RECORD TO EDITOR-RECORD.
    IF INPUT2-RCODE = 05 MOVE WORK-RECORD TO PAGEYEAR-RECORD.
    IF INPUT2-RCODE = 06 MOVE WORK-RECORD TO SUBTITLE-RECORD.
    IF INPUT2-RCODE = 07 MOVE WORK-RECORD TO DESCRIPTION-RECORD.
    IF INPUT2-RCODE = 08 MOVE WORK-RECORD TO CONTENT-RECORD.
    IF INPUT2-RCODE = 09 MOVE WORK-RECORD TO ISBNSEQ-RECORD.
    IF INPUT2-RCODE = 10 MOVE WORK-RECORD TO QUOTATION-RECORD.
**
WRITE-RELATION-RECORD.
    IF INPUT2-RCODE = 01 PERFORM WRITE-CATEGORY-RECORD.
    IF INPUT2-RCODE = 02 PERFORM WRITE-AUTHORTITLE-RECORD.
    IF INPUT2-RCODE = 03 PERFORM WRITE-CONFERENCE-RECORD.
    IF INPUT2-RCODE = 04 PERFORM WRITE-EDITOR-RECORD.
    IF INPUT2-RCODE = 05 PERFORM WRITE-PAGEYEAR-RECORD.
    IF INPUT2-RCODE = 06 PERFORM WRITE-SUBTITLE-RECORD.
    IF INPUT2-RCODE = 07 PERFORM WRITE-DESCRIPTION-RECORD.
    IF INPUT2-RCODE = 08 PERFORM WRITE-CONTENT-RECORD.
    IF INPUT2-RCODE = 09 PERFORM WRITE-ISBNSEQ-RECORD.
    IF INPUT2-RCODE = 10 PERFORM WRITE-QUOTATION-RECORD.
WRITE-CATEGORY-RECORD.
    WRITE CATEGORY-RECORD INVALID KEY



    DISPLAY BLURB-CATEG-KEY ERROR1.
WRITE-AUTHORTITLE-RECORD.
    WRITE AUTHORTITLE-RECORD INVALID KEY
    DISPLAY  BLURB-NO2 ERROR1.
WRITE-CONFERENCE-RECORD.
    WRITE CONFERENCE-RECORD INVALID KEY
    DISPLAY BLURB-NO3 ERROR1.
WRITE-EDITOR-RECORD.
    WRITE EDITOR-RECORD INVALID KEY
    DISPLAY BLURB-NO4 ERROR1.
WRITE-PAGEYEAR-RECORD.
    WRITE PAGEYEAR-RECORD INVALID KEY
    DISPLAY BLURB-PART-VOLUME1 ERROR1.
WRITE-SUBTITLE-RECORD.
    WRITE SUBTITLE-RECORD INVALID KEY
    DISPLAY BLURB-PART-VOLUME2 ERROR1.
WRITE-DESCRIPTION-RECORD.
    WRITE DESCRIPTION-RECORD INVALID KEY
    DISPLAY BLURB-PART-VOLUME3 ERROR1.
WRITE-CONTENT-RECORD.
    WRITE CONTENT-RECORD INVALID KEY
    DISPLAY BLURB-PART-VOLUME4 ERROR1.
WRITE-ISBNSEQ-RECORD.
    WRITE ISBNSEQ-RECORD INVALID KEY
    DISPLAY ISBN1-KEY ERROR1.
WRITE-QUOTATION-RECORD.
    WRITE QUOTATION-RECORD INVALID KEY
    DISPLAY ISBN2-KEY ERROR1.
COPY JCLOSE.
```

LISTING OF MINUS1 19:14 27 MAY 81

```
       ID DIVISION.
       PROGRAM-ID. MINUS1.
      ************************************************************************
      *THE FUNCTION OF THIS PROGRAM IS TO DELETE RECORD(S) FROM THE 2ND*
      * RELATION  BY  THE 1ST RELATION PROVIDS THAT THESE RECORD(S) ARE*
      * EXIST IN THE SECOND RELATION.                                  *
      ************************************************************************
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
       SOURCE-COMPUTER. PRIME.
       OBJECT-COMPUTER. PRIME.
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
           COPY FCONT1.
       DATA DIVISION.
       FILE SECTION.
       COPY JFDDES.
       WORKING-STORAGE SECTION.
       COPY JWORK1.
       01  DETAIL-AREA.
        02 INPUT-RELATION-NAME.
         03 INPUT-NAME              PIC X OCCURS 12.
        02 AND-AREA.
         03 AND-CH                  PIC X OCCURS 4.
        02 INPUT2-RELATION-NAME.
         03 INPUT2-NAME             PIC X OCCURS 12.
       COPY LINKAG.
       PROCEDURE DIVISION USING LINKAGE-FUNCTION.
       CONTROL-ROUTINE.
           PERFORM INITIAL-PARAGRAGH THRU INITIAL2-PARAG.
           PERFORM MAIN-PARAGRAGH UNTIL END-AREA = HIGH-VALUE.
           PERFORM CLEANUP-PARAGRAGH.
       EXIT-PARAG.
           EXIT PROGRAM.
       STOP-RUN.
           STOP RUN.
       COPY BEGIN1.
       INITIAL2-PARAG.
           IF END-AREA = HIGH-VALUE NEXT SENTENCE
                       ELSE
           PERFORM MOVE-TO-INDICATORS
           PERFORM OPEN-INPUT-FILES
           PERFORM READ-FILES.
       MAIN-PARAGRAGH.
           ADD 1                 TO RECORDS-COUNT.
           PERFORM MOVE-TO-MAIN-RELATION.



           DISPLAY 'RECORD TO BE DELETED IS:'.
           DISPLAY WORK-RECORD.
           PERFORM DELETE-RELATION-RECORD.
           PERFORM READ-FILES.
           IF RECORD-CH(1) = HIGH-VALUE MOVE HIGH-VALUE TO END-AREA.
       CLEANUP-PARAGRAGH.
           PERFORM RECORDS-COUNT-CHECK.
           IF CLOSE-COUNT = 1 PERFORM CLOSE-FILES.
           CLOSE RELATIONS-FILE.
       COPY LASTCH.
       OTHER-PARAGRAGH.
       COPY JNAME1.
      **
       MOVE-TO-INDICATORS.
           MOVE 1 TO INPUT-AREA(IN-RCODE) INPUT-AREA(INPUT2-RCODE).
           MOVE 1 TO CLOSE-AREA(IN-RCODE) CLOSE-AREA(INPUT2-RCODE).
           MOVE IN-RCODE       TO OUT-COUNT.
           SUBTRACT 10 FROM OUT-COUNT.
           MOVE 1              TO READ-AREA(OUT-COUNT).
       OPEN-PARAGRAGH.
      *
       COPY INPFL1.
      **
```

```
READ-MOVE-PARAGRAGH.
READ-FILES.
     IF READ-AREA(1) = 01 PERFORM READ-WORK-FILE1.
     IF READ-AREA(2) = 02 PERFORM READ-WORK-FILE2.
     IF READ-AREA(3) = 03 PERFORM READ-WORK-FILE3.
READ-WORK-FILE1.
     READ WORKBOOK1-F AT END MOVE HIGH-VALUE TO WORKBOOK1-RECORD.
     MOVE WORKBOOK1-RECORD TO WORK-RECORD.
READ-WORK-FILE2.
     READ WORKBOOK2-F AT END MOVE HIGH-VALUE TO WORKBOOK2-RECORD.
     MOVE WORKBOOK2-RECORD TO WORK-RECORD.
READ-WORK-FILE3.
     READ WORKBOOK3-F AT END MOVE HIGH-VALUE TO WORKBOOK3-RECORD.
     MOVE WORKBOOK3-RECORD TO WORK-RECORD.
MOVE-TO-MAIN-RELATION.
     IF INPUT2-RCODE = 01 MOVE WORK-RECORD TO CATEGORY-RECORD.
     IF INPUT2-RCODE = 02 MOVE WORK-RECORD TO AUTHORTITLE-RECORD.
     IF INPUT2-RCODE = 03 MOVE WORK-RECORD TO CONFERENCE-RECORD.
     IF INPUT2-RCODE = 04 MOVE WORK-RECORD TO EDITOR-RECORD.
     IF INPUT2-RCODE = 05 MOVE WORK-RECORD TO PAGEYEAR-RECORD.
     IF INPUT2-RCODE = 06 MOVE WORK-RECORD TO SUBTITLE-RECORD.
     IF INPUT2-RCODE = 07 MOVE WORK-RECORD TO DESCRIPTION-RECORD.
     IF INPUT2-RCODE = 08 MOVE WORK-RECORD TO CONTENT-RECORD.
     IF INPUT2-RCODE = 09 MOVE WORK-RECORD TO ISBNSEQ-RECORD.
     IF INPUT2-RCODE = 10 MOVE WORK-RECORD TO QUOTATION-RECORD.
**
DELETE-RELATION-RECORD.
     IF INPUT2-RCODE = 01 PERFORM DELETE-CATEGORY-RECORD.
     IF INPUT2-RCODE = 02 PERFORM DELETE-AUTHORTITLE-RECORD.
     IF INPUT2-RCODE = 03 PERFORM DELETE-CONFERENCE-RECORD.
     IF INPUT2-RCODE = 04 PERFORM DELETE-EDITOR-RECORD.
     IF INPUT2-RCODE = 05 PERFORM DELETE-PAGEYEAR-RECORD.
     IF INPUT2-RCODE = 06 PERFORM DELETE-SUBTITLE-RECORD.
     IF INPUT2-RCODE = 07 PERFORM DELETE-DESCRIPTION-RECORD.
     IF INPUT2-RCODE = 08 PERFORM DELETE-CONTENT-RECORD.
     IF INPUT2-RCODE = 09 PERFORM DELETE-ISBNSEQ-RECORD.
     IF INPUT2-RCODE = 10 PERFORM DELETE-QUOTATION-RECORD.
DELETE-CATEGORY-RECORD.
     DELETE CATEGORY-F RECORD INVALID KEY



     DISPLAY BLURB-CATEG-KEY ERROR2.
DELETE-AUTHORTITLE-RECORD.
     DELETE AUTHORTITLE-F RECORD INVALID KEY
     DISPLAY  BLURB-NO2 ERROR2.
DELETE-CONFERENCE-RECORD.
     DELETE CONFERENCE-F RECORD INVALID KEY
     DISPLAY BLURB-NO3 ERROR2.
DELETE-EDITOR-RECORD.
     DELETE EDITOR-F RECORD INVALID KEY
     DISPLAY BLURB-NO4 ERROR2.
DELETE-PAGEYEAR-RECORD.
     DELETE PAGEYEAR-F RECORD INVALID KEY
     DISPLAY BLURB-PART-VOLUME1 ERROR2.
DELETE-SUBTITLE-RECORD.
     DELETE SUBTITLE-F RECORD INVALID KEY
     DISPLAY BLURB-PART-VOLUME2 ERROR2.
DELETE-DESCRIPTION-RECORD.
     DELETE DESCRIPTION-F RECORD INVALID KEY
     DISPLAY BLURB-PART-VOLUME3 ERROR2.
DELETE-CONTENT-RECORD.
     DELETE CONTENT-F RECORD INVALID KEY
     DISPLAY BLURB-PART-VOLUME4 ERROR2.
DELETE-ISBNSEQ-RECORD.
     DELETE ISBNSEQ-F RECORD INVALID KEY
     DISPLAY ISBN1-KEY ERROR2.
DELETE-QUOTATION-RECORD.
     DELETE QUOTATION-F RECORD INVALID KEY
     DISPLAY ISBN2-KEY ERROR2.
COPY JCLOSE.
```

REFERENCES

ANSI/X3/SPARC,
"Study group on database management systems",
FDT (Bulletin of ACM SIGMOD), 7(2).
1975.

ARTHURS, A M
"Probability theory",
The Gresham Press Ltd, Old Woking, Surrey.
1965.

ASHLEY, R
"Structured COBOL a self teaching guide",
John Wiley & Sons Inc, USA.
1980.

CODD, E F,
"Further normalization of the database relational model"
Current Computer Science Symposia, Vol 6,
1972.

CODD, E F,
"Relational completeness of database sub-language",
Current Computer Science Symposia, Vol 6,
1972.

DATE, C J,
"An introduction to database systems",
1977.

EASON, K D,
"Understanding the naive computer user",
In: Computer Journal, Vol 19, No 1, pp 3-7,
1976.

EASON, K D,
"Man-computer communication in public and private computing",
LUT, Loughborough,
1979.

GANE, C and SARSON, T,
"Structured systems analysis: tools and technique",
By Improved System Technologies Inc,
1979.

GERRARD, G,
"Introduction to the 'Editor'" ,
Doc. No. G1/IN/406,
LUT (Computer Centre),
1978.

HARTSON, H R,
"A semantic model for database protection language",
2nd International IFIP Conference on Very Large Databases,
Brussels, Belgium,
Sept. 1976.

HEBDITCH, D,
"Design of dialogues for interactive commercial application",
David Hebditch Ltd, Otley, West Yorkshire, UK,
1979.

HEILIGER and HENDERSON,
"Library automation, experience, methodology and technology of the library as an information system",
By McGraw-Hill Inc.
1979.

HOPKINSON, A,
"British Library Bibliographic Services Division",
In: Aslib Proceedings 29(8), pp 284-294,
August 1977.

HOUGHTON, Bernard,
"On line information retrieval systems",
Published by Clive Bingley Ltd,
1977.

JACOB, M E L,
"The University of Sydney Library catalogue data entry system",
In: The Lark Association Computerized Cataloguing Systems Series,
Vol 1, Issue 4,
1975.

KENT, W,
"New criteria for the conceptual model",
2nd International IFIP Conference on Very Large Databases,
Brussels, Belgium,
Sept. 1976.

KIMBER, R T,
"Automation in libraries",
2nd Edition,
By Pergamon Press, Oxford, UK,
1974.

LYONS, N,
"Structured COBOL for data processing",
By Glencoe Publishing Co Inc,
1980.

MACLEOD, I A,
"SEQUEL as a language for document retrieval",
In: Journal of the American Society for Information Science,
Vol 30, page 243,
1979.

MARTIN, J,
"Computer database organization",
2nd Edition,
1977.

MARTIN, J,
"Design of man-computer dialogues",
By Prentice-Hall Inc, Series in Automatic Computation,
1973.

MARTIN, J,
"Principle of database management",
1976.

MARTIN, J,
"Security, accuracy and privacy in computer systems",
By Prentice-Hall, Inc, Englewood Cliffs, New Jersey,
1973.

MIDAS,
"Multiple index database access system reference guide",
By Prime Computer Inc, Doc PDR 3061.

MOHAMMAD SALIH, A H,
"Integrity, security aspects of data management",
MPhil Thesis, Department of Computer Studies, LUT,
1979.

NOERR, P L and WHITE, M C,
"MERLIN: Design of national bibliographic database",
In: Systems for Large Database, Proceedings of a 2nd International Conference, Brussels, 8-10 September 1976.

OLLE, W T,
"The CODASYL approach to database management",
By John Wiley and Sons Ltd,
1978.

PRENTICE, J A,
"Terminal system, Doc No. G2/RE/315",
By LUT, Computer Centre,
1979A.

PRENTICE, J A,
"Preliminary introduction to use of 'Prime' interactive system",
By LUT, Computer Centre,
1979B.

ROBINSON, S,
"Sleeping Beauty: MERLIN, a state of the art report",
In: Program, Vol 14, No 1,
1980.

ROSS, J and ROYAN, B,
"MERLIN: a new computer system for British Library",
In: Program, Vol 10, No 3, pp 95-102, June,
1976.

THE COBOL PROGRAMMER'S GUIDE,
By Prime Computer Inc, Doc PDR 3056.

TONI, C B,
"Secondary information systems and services",
By The National Federation of Abstracting and Indexing Services (NFAIS),
In: Annual Review of Information Science and Technology, Vol 13, Part 2,
1978.

TSICHRIZIS, D C and LOCHOVISKY, F H,
"Database management system",
By The Academic Press Inc,
1977.

TSICHRIZIS, D C and LOCHOVISKY, F H,
"Hierarchical database management: a survey by the Department of Computer Science, University of Toronto, Toronto, Canada",
In: Computing Surveys (ACM), Vol 8, No 1, page 105,
1976.

WALL, R A,
"MINICS (Minimal-Input Cataloguing System",
Development report,
By Loughborough University of Technology (LUT),
1973.

WHITE, M C and NOERR, P L,
"A brief introduction to data storage and retrieval in MERLIN (Machine Readable Library Information)",
By MERLIN Project Team, The British Library Bibliographic Service Division,
In: Program, Vol 10, No 4, pp 123-134, October,
1976.

YOURDON, E,
"Managing the structured techniques",
Second Edition formally titled: "How to manage structured programming",
1979.