DESIGN AND DEVELOPMENT OF AN INFORMATION MANAGEMENT SYSTEM

A THESIS

Submited to the department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in partial fulfillment of the degree of

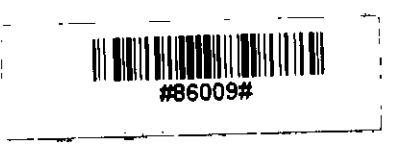
MASTER OF SCIENCE IN ENGINEERING

[COMPUTER SCIENCE AND ENGINEERING]

BY

MOHAMMAD ABDUL MANNAN

Roll No. 861823 F, Registration No.84256, Session: 1984-85



DESIGN AND DEVELOPMENT OF AN INFORMATION MANAGEMENT SYSTEM

By

Mohammad Abdul Mannan

Accepted as satisfactory for partial fulfillment of the requirements for the degree of Master of Science in Engineering [Computer Science and Engineering] on 22-04-93 .

BOARD OF EXAMINERS

1. ---------- Chairman
   DR. Mohammad Kaykobad and
   Assistant Professor Supervisor
   Dept. of Computer Science and Engg.
   B U E T , Dhaka.

2. ---------- Member
   Prof. DR. A.B.M. Siddique Hossain
   Head of the department of
   Computer Science and Engineering,
   B U E T , Dhaka.

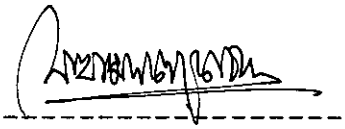
3. ---------- Member
   DR. Md. Shamsul Alam
   Associate Professor
   Dept. of Computer Science and Engg.
   B U E T Dhaka.

4. ---------- Member
   Prof. DR. Md. Sekendar Ali [External]
   Director
   Bangladesh Institute of Technology Dhaka.

DECLARATION

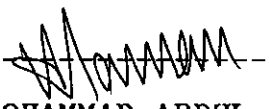
I do hereby declare that the work presented in this thesis is done by me under the supervision of DR. Mohammad Kaykobad, Assistant Professor, Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh. It is also declared that neither this thesis nor any part thereof has been submitted elsewhere for the award of any degree or diploma.

Counter-signed by the supervisor



[DR. MOHAMMAD KAYKOBAD]

Signature of the candidate



[MOHAMMAD ABDUL MANNAN]

ACKNOWLEDGMENTS

This research work was started under the supervision of DR. Syed Mahbubur Rahman, Associate Professor, department of Computer Science and Engg. (CSE), B U E T Dhaka and due to his absence was completed under the supervision of DR. Mohammad Kaykobad, Assistant Professor of the same department. Their continuous guidance, suggestions, inspiration and advice made it possible to complete this work. I express my heartiest gratitude and profound indebtedness to them. The same is also expressed to Prof. DR. Md. Sekendar Ali, Director, B I T Dhaka, Prof. DR. A.B.M. Siddique Hossain, Head, Dept. of CSE, B U E T, Dhaka and DR. Md. Shamsul Alam, Associate Professor, Dept. of CSE, B U E T, Dhaka for their kind co-operation, inspiration and valuable suggestions.

Sincere gratitude is also expressed to Mr. Md. Abdus Sattar, Assistant Professor, Dept. of CSE, B U E T, Dhaka for his cordial assistance and guidance in developing the Information Management System under this research work.

The support and services rendered by the staff of micro-computer laboratory of CSE department, B U E T, Dhaka and Computer Center, B I T Dhaka are also acknowledged with thanks.

The author must not forget to express his heartiest gratitude to the U N D P who has provided financial support for M.Sc.Engg. course.

ABSTRACT

This thesis presents a compete analysis, design and development of an information management system [IMS]. The existing manual information system is analyzed, different solutions with performance and costing are presented and a gantt chart is prepared for the recommended solution. A computerized information management system is designed. Design of the computerized I M S involves design of conceptual model, logical model, output, input, database, menu structure, process, program, package, system operation and maintenance, and system security and control. All these design works are presented in chapter 4 . Different queries are formulated in the form of menu structures so that any query can be resolved after entering appropriate data sought by the system. A database is designed to store data. Catalogs of various attributes, database files, program files etc are accommodated in database dictionary. As information is a vital factor in any organization, the system is designed with two levels of password-control. In Bangladesh, there a is random change of government policies regarding some attributes; so the system is made independent of government policies. That is, if the system manager enter changed data in related data file[s], the system will serve accurately after the change of government policies affecting some attributes.

Finally the designed system is developed using the programming language of dBASE III plus. Various development works are presented in chapter 5 . The performance and specifications of the developed system is then compared with that of the designed system.

CONTENTS

CONTENTS [contd.]

CONTENTS [contd.]

LIST OF FIGURES

LIST OF FIGURES [Contd.]

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Meaning
DBMS	Data Base Management System
IMS	Information Management System
DBA	Data Base Administrator
IPO	Input Process Output
ES	Expert System
KBS	Knowledge Based System
DBS	Decision Based System
LBS	Logic Based System
I/O	Input / Output
ER	Entity-Relation
S.C.	Security control
P M	Personnel Management
I M	Income Management
S M	Salary Management
N H M	Non-salary Heads Management
Sys.M	System Maintenance
D M L	Data Manipulation Language
Q L	Query Language
Q B E	Query By Example
S Q L	Structured Query Language
C Q L	Commercial Query Language
A_i	Attributes
r_i	Relations

CHAPTER ONE

INTRODUCTION

CHAPTER OUTLINE:

1.1 Introduction to Information Technology
1.2 Importance of Information Technology in Bangladesh
1.3 Problem formulation
1.4 Research objectives
1.5 Literature survey

This chapter presents an introduction to information technology and its importance in Bangladesh. It also includes a brief discussion on literature survey for the present thesis work, problem definition and objectives of the research work.

1.1 INTRODUCTION TO INFORMATION TECHNOLOGY

Information is the vital factor in every step of human life specially in developed countries. The united states has evolved from an industrial society to information society. Historically, the United States became a world power as an industrial society. As the need for information developed, Herman Hollerith's tabulating machine was one of the first attempts to streamline the collection and dissemination of information. Little did Hollerith realize that his machine would lead to the birth of another revolution- the information revolution. Increasingly, during the first half of 1900s, technology focused on developing machines. By 1950, the year that marks the beginning of the first generation of computers, information needs had grown so rapidly that workers employed in information-related jobs outnumbered the workers in industrial sector. In the United States, predictions for the future suggest that the number of information workers will soon surpass the total combined number of workers in industry, agriculture and service[15]. Information is increasing at an astounding rate. Of the total information available today, 75% became available in the last twenty years[15]. To meet the rapidly increased information, Computer information technology is also simultaneously developed in the last twenty years[15].

An information system can supply many types of information. Originally, information systems provided standard reports; such as accounting statements, sales summaries, payroll reports, personal reports etc. To handle the large volume of sophisticated and specific information. Information Management Systems (IMS) are developed. Recently information systems have been designed to provide information to support decision making. This application is called a Management Information System. Other applications are Expert Systems, Hyper Text Systems etc[14]. Expert Systems include knowledge Based System (KBS), Logic Based Systems (LBS), Decision Based Systems (DBS) etc. Disaster management systems are

in crime, corruption etc. This situation can be managed successfully by using information technology. Information technology can also be used to manage the disaster situation caused by cyclone, tornado, flood and other herocs. Such application of information technology provides us Disaster Management Systems. DMSs can forecast the volume of various destructions that will occur and various aids, transportation, service etc needed before the occurrence of cyclone, tornado or flood.

At present, different organizations have sufficient number of computers but most of the organizations are depending on imported package softwares to manage their information. These costly package softwares cannot serve specific application in an organization. Moreover to use these packages, some modifications must be made and more trained personnels are required. These machines should be utilized by locally developed customized information systems. It is hoped that within a few years Bangladesh will have sufficient number of software Engineers capable of designing and developing customized information systems at very low cost for specific application to the needs of various organizations in Bangladesh.

1.3 PROBLEM FORMULATION

Information Management Systems of all academic Institutions in Bangladesh are more or less same. There are numerous information processing jobs in the existing manual system and a large portion of manpower is involved. It is evident that the existing system is running on high cost, long-time and poor performance. Moreover, the volume of information is increasing day by day. A few problems in the existing system of management and administration are briefly focused below:

(a) GROUPING OR LISTING PROBLEM: Grouping or listing of manpower, items, machines, equipments etc are frequently needed for

also knowledge Based Expert Systems[15].

Modern business cannot be run without information; it is the life blood of an organization. Any organization can be viewed as a total system composed of three subsystems[14], namely,

(1) the operations subsystem

(2) the management subsystem and

(3) the information subsystem.

The information subsystem is the assemblage or collection of people, machines, ideas and activities that gather and process data in a manner that will meet the formal information requirements of an organization. Its purpose is to satisfy information requirements including and decision making needs of all levels of management; and the needs of concerned parties external to the organization. This relationship is depicted in Figure 1.1 below[14]:

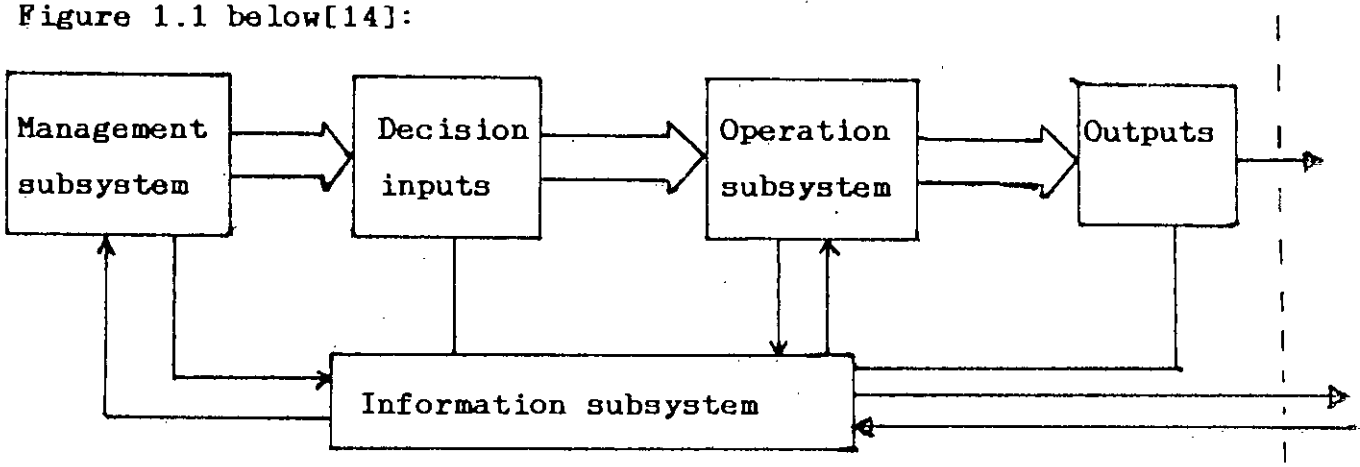


Figure 1.1 Modern organization

1.2 IMPORTANCE OF INFORMATION TECHNOLOGY IN BANGLADESH

Bangladesh as a developing country, is trying to get maximum benefit from Computer Information Technology to speedup its development. But still information technology is in primitive level. Most of the information are processed manually, and thus a large part of manpower is engaged. Moreover, most of the information processing manpower are overloaded with their jobs. On the other hand, volume of information is increasing day by day due to the development of technology, civilization and increase

management and administration. But these are very tedious, erroneous and time consuming job to office assistants. But in a computer based information management system, such grouping or listing reports can be generated very quickly and smoothly in a structured manner.

(b) REPORTING PROBLEMS: Various reports on different formats are generated frequently by office assistants under the direction of the executives at various levels. These reports are very essential for the top management. To create a single report, the office assistant collects necessary records from different files, manipulates records and creates a draft copy of the report. After the draft report is checked by the concerned executive, the assistant goes for preparation of final report. So to create a single report, enough time & energy is required but the possibility of error cannot be eliminated because of human factor. But in the computer based system, a large number of such reports can be generated within a few minutes by a single person.

(c) PROBLEMS IN ACCOUNTING SYSTEM: Accounting information systems of every organization are very complex. In B.U.E.T. Dhaka, the volume of accounting information is very large and the system is highly complex. Hundreds of reports are generated per month for salary system only. There are above six hundreds heads of income and expenditures. Thousands of various vouchers are manipulated per week. Thus the operating manpower is highly overloaded, the system is very time consuming, erroneous and of poor performance.

In this way it can be shown that the existing manual systems of all organizations, bodies, institutions, Govt. departments, directorates and corporations in Bangladesh have been facing such a major problem in management of information. Recently, corruption is the additional but serious problem in management and administration. But using computer technology, corruptionless high efficient systems can be developed.

This thesis is an approach to establish a new trend in

solution of such a national problem. Development of a high quality IMS depends on analysis & design of the system. Errors in analysis of existing system will be carried through design and coding i.e. even if the program satisfies the design specification, the system will not satisfy the users. System analysis is required because of a need to solve a problem, as a response to new information requirements, as a method of incorporating new technology into a system as a means of developing broad system improvements etc. In design, various techniques, concepts and thinking must be utilized efficiently to develop GRID-CHART, SYSTEM FLOW CHARTS, IPO [input, process, output] CHARTS, DECISION LOGIC TABLES, I/O [input/output] DESIGN etc. So that the whole system becomes more versatile, efficient & useful. This research is aimed at stepping up the conventional IMSs to a new generation level i.e. to search for a direction of new trend in analysis, design and development of IMSs.

1.4 RESEARCH OBJECTIVES

The scope being limited, the present research is confined into the following objectives:

(1) Study of the conventional concepts and techniques on analysis, design and development of IMSs.

(2) Study of the design concepts for the conventional database design.

(3) Preliminary analysis of the whole information processing systems of Bangladesh University of Engineering and Technology (BUET) Dhaka.

(4) Detail analysis of the administrative and accounting information processing systems of BUET Dhaka.

(5) Design of a new IMS for accounts & administration of BUET Dhaka.

(6) Design of a database for the proposed IMS.

(7) Development of the proposed IMS.

1.5 LITERATURE SURVEY

Literature survey is the most important part of this thesis work. On preliminary survey of various branches of computer science and engineering, interest grows on computer information technology. Literature survey is found as the way to gain vast theoritical knowledge in this field. It leads to realize that the existing problems in management and administration of different organizations, Government departments, directorates and corporations in Bangladesh can be solved by introducing computer based IMS. It has been proved that performance and efficiency of an IMS depend entirely on analysis and design of the system. The present work is directed towards exploring various concepts & techniques on analysis and design of IMS and searching for a direction of new trend in system analysis and design. Thus the literature survey is divided into the following constituent areas:

1. Study on various branches of Computer Information technology.
2. Study on analysis, design and development of Information Management Systems.
3. Thorough review of the various concepts techniques of analysis and design available in existing books, journals & similar research works.
4. Thorough study of various techniques & concepts on design of database.
5. Study of various approaches of structured system development.

There are many branches of Computer Information Technology viz. Information Management Systems, Management Information Systems, Expert Systems etc[15].

IMS is ideal for organizations that must provide information to the public. Universities, airlines, public utilities, banks, government offices etc. are organizations that are benefited from IMS support. Though its initials are similar to MIS, an IMS is designed for more specific objectives. While MIS has a decision

making orientation, IMS focuses its attention on the management of information generated by the computer. IMSs are concerned with the distribution of data through a system, data communication activities, access to a database and users' information need. A major function of an IMS is to control the flow of information to people when answering questions and solving problems. On the other hand MIS is most successful where a diverse collection of data must be managed to provide decision making information. These systems are supported by large computer systems. This vast processing capacity is wasted when employees provide faulty or incomplete input date. Todays MIS projects are much more complex, e.g. international companies use on-line real time processing systems and satellite communications in their international MISs. These sophisticated systems can give all levels of management the most up-to-date information[15].

Expert Systems, on therhand, include artificial intelligence. There are different types of expert Systems (ES), viz. knowledge based E.S., decision based E.S., logic based E.S. ect. The basic theory of developing Expert Systems is to collect knowledge / logic / rules from human experts and represent them in standard, systematic and organized fashion. The technique involved is called knowledge engineering and the concerned person is called knowledge engineer. The organized knowledge in standard fashion is then processed by application programs to supply output information. Application programs are written in special programming language / package like prolog, shell[15].

To develop an efficient IMS, the existing system should be analyzed thoroughly to find system requirements and satisfy user's request. analysis phase consists of preliminary and detail analysis. After each phase a report including cost, budget, gantt chart, alternate solutions and system recommendation should be prepared for management's action. After approval of management, the system analyst will proceed to design an IMS for the existing

CHAPTER TWO

TECHNIQUES AND CONCEPTS OF DEVELOPING I M S

CHAPTER OUTLINE:

This chapter is a review of the theories and concepts of developing Information Management System (IMS).

2.1 EVOLUTION OF DATA BASE TECHNOLOGY:

Traditionally, computerized systems are implemented in piecemeal fashion. A department within the enterprise decides or is cajoled into computerizing part of its function. An analyst goes in and resign a system. The system is implemented by Specifying a number of programs. The data is tailored to fulfill the needs of the given system within enterprise and implemented. In this approach data for a program or system cannot be shared by another program or system and thus data redundancy occurs. This concept of traditional file processing [14] is illustrated below:

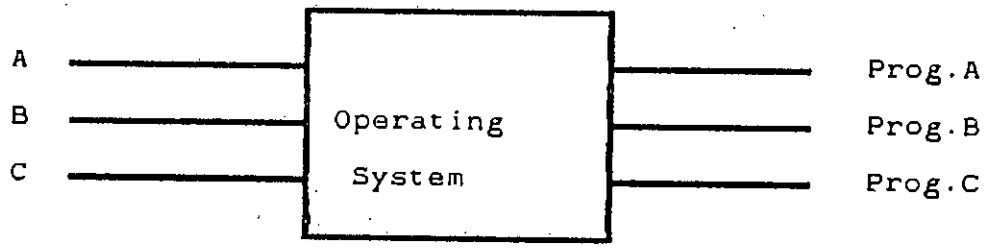


Fig.2.1 Traditional file processing.

In database technology, a single database is designed to store all data of the enterprise so that multiple application programs can use this database. The modern concept of information processing [14] with database technology is illustrated below:

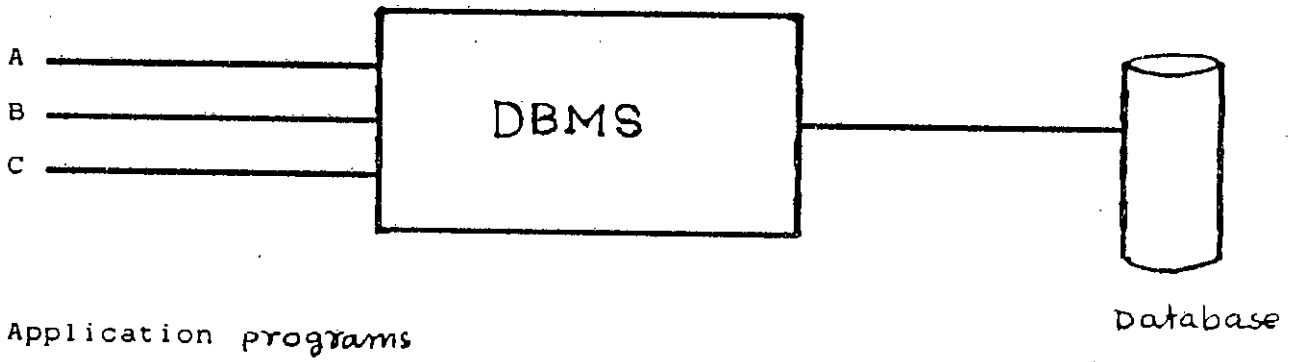


Fig.2.2 File processing in database technology

DISADVANTAGES OF TRADITIONAL FILE PROCESSING SYSTEM:

[a] Data redundancy & inconsistency: Since the files and application programs are created by different programmers over a long period of time, the same piece of information may be duplicated in several places (files). This redundancy leads to higher storage and access cost as well as potential data inconsistency. By data inconsistency, we mean that various copies of the same data on longer agree [12].

[b] Difficulty in accessing files: Suppose that one of the officers in a bank needs to find out the names of all customers who live in the area of the city zip code 788733. The officer calls data processing department and asks them to regenerate such a list . as this is an usual request that was not anticipated when the original system was designed there is no application program on hand to generate such a list . There is,however , an application to generate the list of all customers. The bank officer has now two choices. Either he can get the list of customers and one of his secretaries to extract manually the needed information , he can the data processing department to have one of the system programmers write such an application program neither of which is satisfactory. Since data is scattered in various files and files may be in different formats , it is difficult to write a new application programs to retrieve the appropriate data [12].

[c] Multiple users: In order to improve the over all performance of the system and a faster response time, many system allow multiple users to update the data simultaneously. In such a environment interaction of concurrent update may result in consistent data [12].

[d] Security problem: Not every user of the database system should be able to access all data. For example, in a banking system, a person who prepares the payroll checks can only see that part of the database that has information about customer accounts. Since application programs are added to the system in an adhoc manner, it is difficult to enforce such security constraints [12].

[e] Integrity problem: The data values stored in the database must satisfy certain types consistency constraints. For example, the balance of a bank account may never fall below a pre-specified amount. These constraints must be enforced in the system. This enforcement can be carried out by adding appropriate code in the various application programs. However, when new constraints are added, it is difficult to change the programs to enforce them [12].

These difficulties, among others, have prompted [12] the development of database technology.Today's requirements [12] in Database System are mentioned below:

1. Software should provide logical as well as physical data independence.
2. Facilities should be provided for a database administrator to act as a controller of the data.
3. Effective procedures should be provided for controlling privacy, security and integrity.
4. Inverted files should be used in some system to permit database searching.
5. Database should be designed to provide answers to unanticipated forms of information requirements.
6. Should have data migration facility.
7. There should be data definition language for database administrator, command language for programmer and data interrogation language for users.

2.2 OBJECTIVES OF DATABASE ORGANIZATION

Primary objectives [12]

1. The database is the foundation stone of the future application development.
2. Data can have multiple uses.
3. Ease of use.
4. Flexible use.
5. Unanticipated request for data can be handled quickly.
6. Clarity of data.
7. Change is easy.
8. Low cost.
9. Less data proliferation.
10. Performance.
11. Accuracy and consistency.
12. Privacy.
13. Protection for loss or damage.

Secondary objectives [12]:

1. Physical data independence.
2. Logical data
3. Controlled redundancy.
4. Suitably fast access.
5. Suitably fast searching.
6. Data standarization within corporation.
7. Data dictionary.
8. End user language.
9. Fast recovery from failure.
10. Tenability.
11. Design and monitoring aids.
12. Evolution of distributed database organization.

2.3 CONCEPTS & METHODOLOGIES:

A major problem in traditional system analysis is that the users do not understand the specification produced. The charge of the system analyst is to determine what the user wants. This is not an easy job. Various concepts and methodologies on system analysis & design have been developed based on database technology. Some of these are briefly discussed below:

American view point: American view [6] on analysis & design of information management systems leads to the development of structured analysis. Structured analysis was first conceptualized around 1973 with 1977 marking. The first publishing of a way to ensure that the user's needs are known and understood and that a new system will satisfy those needs. This methodology has the potential to reduce maintenance to error correction and modifications only when the business changes. Thus we see that it should:

1. Assist the analyst by directing him to collect the right information.
2. Provide a formal unambiguous way of writing down the findings of his discussion; these must be intelligible to end users and indeed it is preferable if the end users write down their requirements using the methodologies themself.
3. Include one or more diagrammatic techniques to enable the analyst visualize his system succinity.

UK-based methodologies [6]:

Once the area of data to be analyzed has been defined the first step is to determine the principal entity with which the enterprise is concerned. The next step is to construct a data model of that part of the enterprise being analyzed. In this data model are expressed the entity types and the kinds of relationships that exist between the individual entity occurrences. The recognition and definition of attributes is not

so critical that it need be complete missing the early stage of data analysis. On the contrary, data analysis is very much a re-iterative process and it is unlikely that a complete satisfactory data model will be obtained on the first iteration.

2.4 SYSTEM ANALYSIS

Analysis of the existing system is the first step of developing Computerized Information system. The following are the reasons for initiating system analysis[14]:

(1) Solving problems of the existing system.

(2) New requirements of the system.

(3) Implementation of new idea or technology.

(4) Broad system improvement

Sources of study facts for system analysis are [14]: (1) the existing system (2) other internal sources and (3) external sources. The primary advantages of analyzing the existing system are (a) effectiveness of the present system, (b) provide design ideas and (c) resource recognition (d) conversion knowledge and (e) common starting point. The most important internal source of study facts available to the analyst is people. A second source of study facts is the existing paperwork within the organization. A third source is the relationships between people, departments, functions etc. Exploring other information subsystems within the organization can be a useful external source of data collection,data processing or information reporting ideas and techniques.

Some major tools and techniques used by the system analyst in developing information management systems are briefly described below [14]:

THE INTERVIEW:

Within an organization, interviewing is the most significant and productive fact-finding technique available to the system analyst. It is a communication channel between the analyst and

the organization. Interviewing is used to gain information concerning what is required and how these requirements can be met. Interviewing is conducted at all levels within the organization , from the executive to the mail clerk or the maintenance engineer. In conducting the interview, the system analyst should behave in a manner and ask questions that will get the required study facts in as little time as possible.

THE QUESTIONNAIRE:

The questionnaire is another tool which can be used at various times by the systems analyst in the systems development process. The use of the questionnaire in systems analysis should be limited to only those situations where the analyst cannot conduct an interview. when the analyst decides to make use of questionnaire there are a few, but important guidelines to follow:

1. Explain the purpose, use, security and disposition of the responses.
2. Provide detailed instructions on how you want the questions completed.
3. Give a time limit or deadline for return of the questionnaire.
4. Ask pointed and concise questions.
5. Identify each questionnaire by respondent's name, job title, department etc.
6. Include a section where respondents can state their opinions and criticisms.

OBSERVATION:

Another technique available to the analyst during fact-finding is to observe people in the act of executing their job. Industrial engineers utilize this technique extensively for studying people in groups and organizational activities. observation can be used to verify what was revealed in an

interview or as a preliminary to the interview. Observation is also a valuable technique for gathering facts representing relationships.

SAMPLING AND DOCUMENT GATHERING:

Sampling is directed to collecting and accumulating data on problems that are either un-measurable or requires tremendous amount of detail work to obtain a given piece of data.

Collecting exhibits of source documents, work-sheets, reports etc. is another way for the analyst to gather information during system analysis. From these exhibits or documents, the analyst can gain an understanding of what is presently done,
how it is structured, what is not available and, perhaps, get a 'feel' for what is considered important.

CHARTING:

Charting is the technique that pictorially represents some dimension of an organization or an organizational activity. Of all the tools and techniques utilized by systems, charting is the one technique most closely identified with system efforts. It is also a valuable technique for performance analysis, synthesis, communication and documentation. There are three broad classifications of charting: Organization charts, Physical layout charts and Flow charts.

According to recent view, analysis phase is decomposed into preliminary and detail analysis. Fig 2.3 represents the decomposition of preliminary analysis into its components. Users, managements and computer expertise provide input at various stages of this phase. Detail analysis results in a feasibility study which becomes major input to design It includes detail study of the system, finding of facts, design of alternate solutions and presentation of th e recommended solution to management.

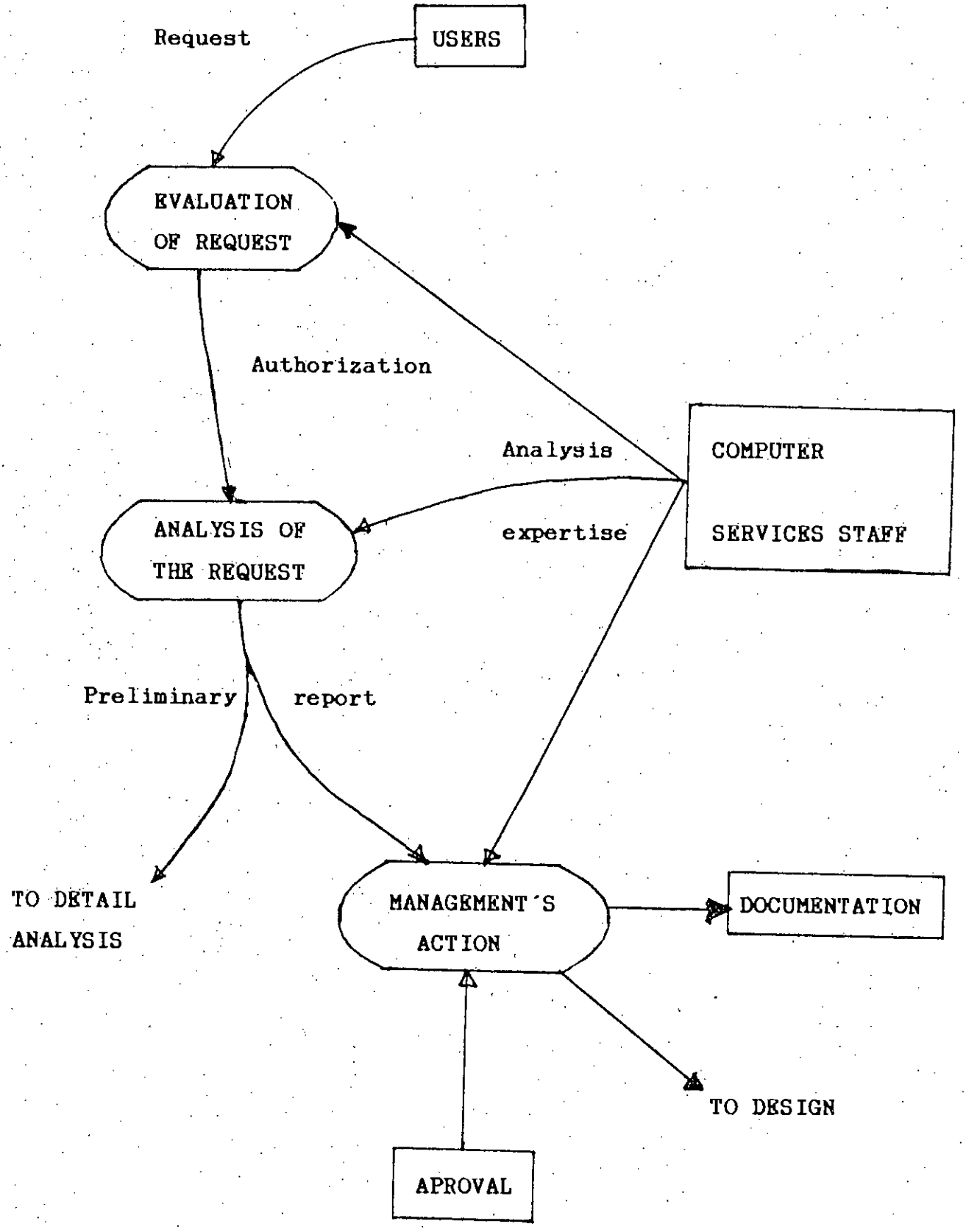


Figure 2.3 Data flow diagram decomposing the preliminary analysis

2.5 EVALUATION OF DATA MODELS:

There are three major data models [12] to represent data relationship: hierarchical, network and relational. The relational data model represents data relationships as tables. The traditional data manipulation language of the relational model was relational algebra and relational calculus. However, the current trend is to provide entries in a table using English like statements and to eliminate mathematical procedures for manipulation. SQL, the data manipulation language for IBM database management system, reflects th is trend. To evaluate data models, a framework comparing the three models is presented below [12]:

TABLE 1 Comparison of data models

	HIERARCHICAL	NETWORK	RELATIONAL
Representative System	DL/I(IBM)	IDMS(Cullinet)	SQL
Data building blocks	Field Segment Physical data -base	Data item Record Set	Attribute(column) Tuple(row) Relation(Table)
Logical data structures	Directly	Decomposition into sets	Decomposition intc tables
Simple network	Unidirectional relationship	Decomposition into sets	Decomposition into sets
Complex networks	Bi-directional relationship	Decomposition into sets	Decomposition into sets
Data indepedance:			
-Path	No	No	Yes
-Sequence	No	No	Yes
DML commands:			
-Retrieval	GU,GHU,GN,GHN, GNP, GHNP GNP	FIND, GET	SELECT

TABLE 1 Comparison of data models [contd.]

	HIERARCHICAL	NETWORK	RELATIONAL
-Data alteration	REPL	MODIFY	UPDATE
-Data addition	ISRT	STORE	INSERT
-Data deletion	DLET	ERASE	DELETE
-Miscellaneous		READY, DISCONNECT CONNECT, FINISH	
Navigator	Experienced and trained.	Experienced and trained.	End user
Means of database navigation.	Through hierarchical paths.	Through sets	Through value of the attributes.
Modification of data structure.	Redifine structure, reload new structure.	Redefine structure, reload new structure.	Restructure at any time including dur operation in onlin

2.6 ARCHITECTURE OF A DBMS

A DMBS (database management system) may be defined as a general purpose set of programs that aid and control access to and use of the database for adding, modifying etc & retrieving data.

As understood from the definition, a DBMS has an important role in database design because the schema processors, user interfaces, core database handling processors etc. are in a DBMS. It controls any manipulation over a database. Though the total database is independent of the DBMS programs, yet sometimes, acts as an important tool in database design.

If the conceptual components of a standard DBMS is illustrated, then the rule of a DBMS will be depicted. The conceptual architecture of a DBMS is shown in fig 2.4 below.

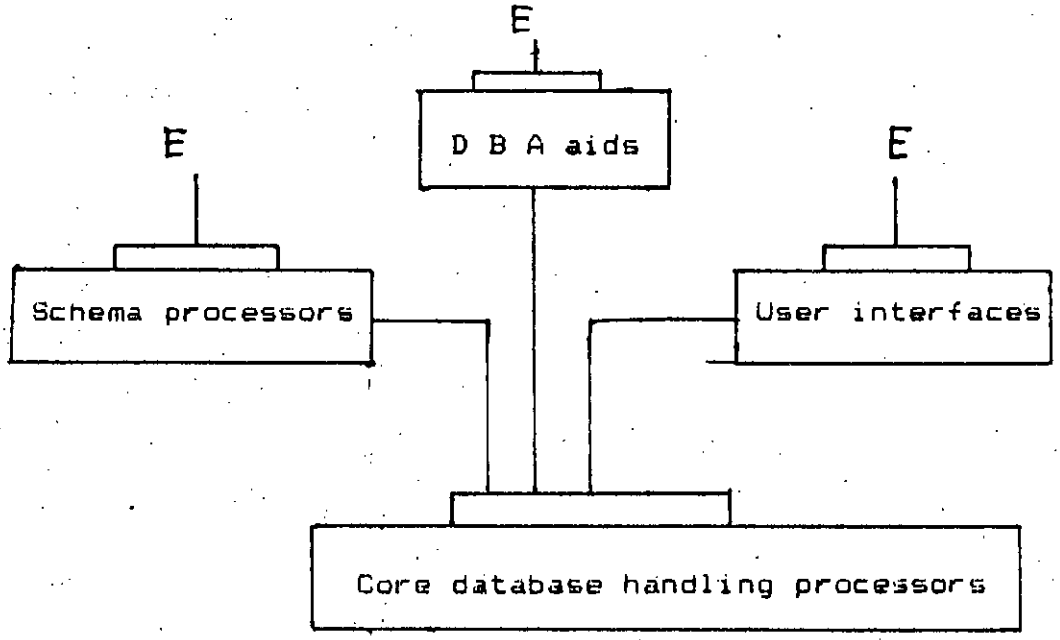


Fig. 2.4 Architechture of DBMS.

An overview of the four conceptual components is presented below:

(A) SCHEMA PROCESSOS: Schema processors are collection of components for handling all aspects of database definition, including structure definition, substructure definition, access control, logical performance measurement specifications. The schema processors provide language processing capabilities to translate user's request, validate them, additional user operations via the core database handler.

(B) USER INTERFACE: This portion of the architecture includes the components for query language processing, host language interfacing and other more advanced user interfaces, such as natural language interfaces and editors to assist in the construction of user's request.

(C) DATABASE ADMINISTRATION AIDS: This portion of the architecture includes the database dictionary and database design aids both of these functions require information that is defined by the schema processors and stored by the core database handler.

(D) THE CORE DATABASE HANDLERS: These components are the central elements of the architecture. They provide the storage and retrieval facilities for all data stored inthe system; that is, both the user data and system data that are required by other components. The interface to the core database handler is a logical interface that provides functions for both sets of records and individual records.

addition to providing and defining a target interface for query language processors the database handler also provide facilities to suport interfaces to host language application programs and to schema processors.

2.7 STEPS IN READING A RECORD BY A DBMS

The main events that occur when an application program reads a record from a database by means of a DBMS are shown in fig 2.5.

The events are described below:

1. Application program issue a call to the DBMS to read a record. The program states the programmer's name for the data type and gives the value of the key of the segment or record in the question.

2. The DBMS obtains the subschema (program data description) that is used by application program A and looks up the description of the data in question.

3. The DBMS obtains the schema (or giobal logical data description) and determines which logical data type or types are needed.

(4) The DBMS examines the physical database description and determines which physical record or records to be read.

(5) The DBMS issues a command to the computer operating system instructing it to read the requisite records.

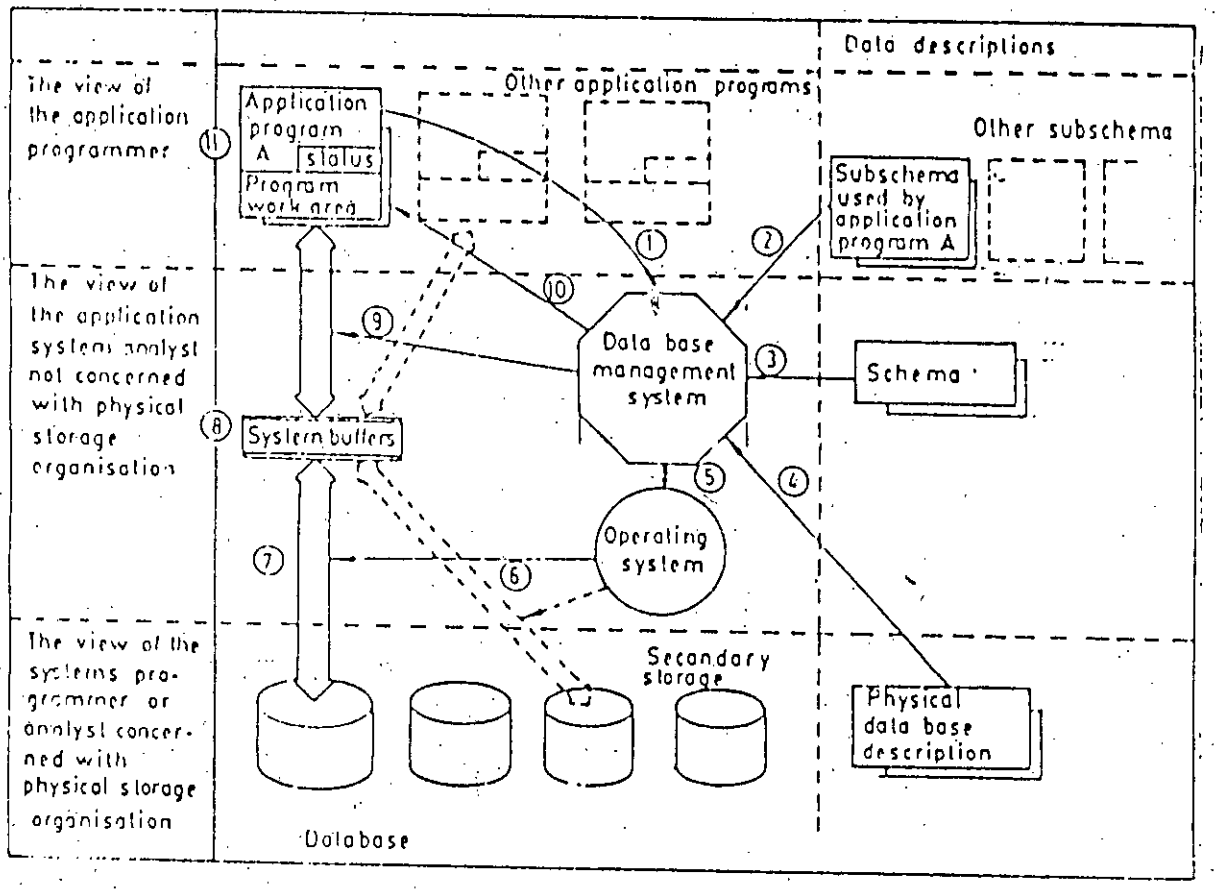


Fig:- 2.5 - Steps in reading a record by a DBMS.[19]

(6). The operating system interacts with the physical storage where the data are kept.

(7) The required data is transferred between the storage and the system buffers.

(8) Comparing the subschema and schema, the DBMS derives from the data the logical record needed by the application program. Any data transformation between the data as declared in the subschema and the data as declared in the schema are made by the DBMS.

(9) The DBMS transfers the data from the system buffers to the works area of application program.

(10) The DBMS provides status information to the application program on the outcome of its call, including any error indication.

(11) The application program can be then operates with the data in its work area.

2.8 DATA INDEPENDENCE AND DATA MODEL

There is a large number of data-item types, so we need a map showing how they are associated. This map is sometimes called a data model. The objectives of the modeling approach is to provide a basis for the design of effective and usable databases.

The term "data independence" is often quoted as being one of the main attributes of the database. It implies that the data and the application programs which use them are independent so that either may be changed without changing the other.

Two stages of data independence can be shown with the help of a data model:

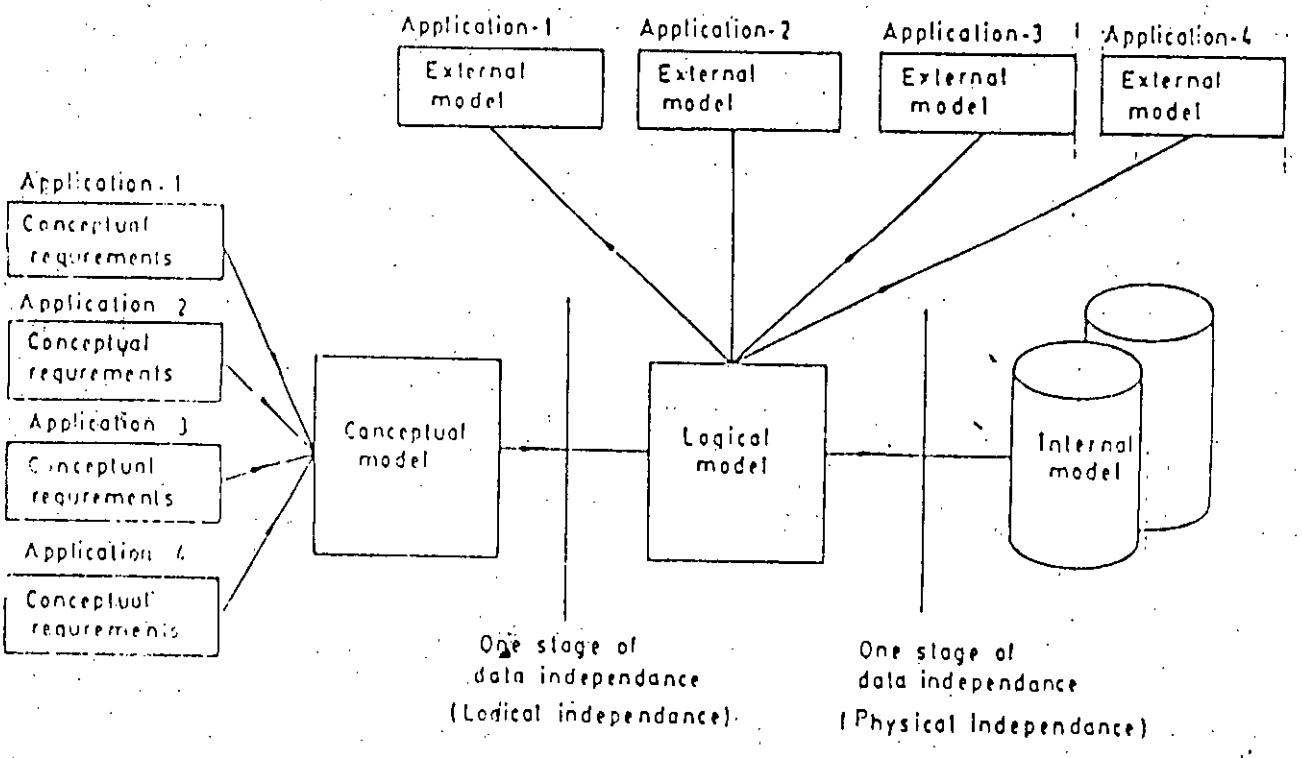


Fig:- 2.6 - Two stages of data independence [19]

(1) Logical independence

(2) Physical independence

These two stages in data independence can be shown as in figure 2.6 . Different sorts of data models are concerned to the database design process. With the help of these models, the data independence can be easily illustrated.

CONCEPTUAL MODEL: The database design starts with the conceptual requirements of a numbers of users. These requirements of individual users are integrated into a single "community" view, called conceptual model which represents the entities and their relationships. It gives us the ability to view all the data entities and their relationships to each other with no concern about their physical storage. The conceptual model is a communication tool between various users of data as such is developed without any concern for physical representation, it is independent of a DBMS.

LOGICAL MODEL: The modified version (compatible and implementable) of the conceptual model which can be presented to the DBMS is called a logical model. The DBMS is not a factor in designing the conceptual model, but designing the logical model is dependent on the DBMS to be used.

External model: The users are presented with the subsets of the logical models .This subsets are called logical models or sub-schema .These models are mapped from the views that users 'get' based on the logical model. The conceptual requirments are views that the users wanted initially and based on which conceptual model is to be developed.

Physical model or internal model schema: The logical model is mapped to physical storage such as disk, tape, drum etc. The physical model, which takes into consideration the distribution of data, access methods and indexing techniques is the internal model or schema. The external models should not be affected by physical storage changes or by access method changes to the database. This is the first stage of data independence.

2.9 SYSTEM DESIGN

System design can be defined as the drawing, planning, sketching or arranging of many seperate elements into a viable unified whole. The design phase is technically oriented to the extent that the analyst must answer the questions: " How do we do it ? " . On the other hand, design is an art, and creatively oriented to the extend that the analyst continuously asks: " What is it ? " and " Why not ? " questions.

To design a system the analyst must possess knowledge related to the following subjects:
(1) organizational resources, (2) user information requirments, (3) humanizing requirements, (4) other systems requirements, (5) methods of data processing, (6) data operation and (7) design

tools. Design can be decomposed into nine stages:

(a) Review and assignment of tasks: In this stage, various scheduled tasks or jobs are assigned to respective personnels involved.

(b) Output design: In this stage, the outputs of the system are reviewed and designed. This will be discussed seperately in the following article.

(c) Database design: In this stage all the attributes required to generate the system's output are organized into one or more data files (called the database). How to design a database will be discussed in article 2.11 .

(d) Input design: In this stage, various processes involved in implementing a basic system module are designed precisely.

(f) Program defination: In this stage, one or more programs are defined to implement a basic system module.

(g) Module design: Large programs are modulerized for efficient implementation.

(h) Package design: In this stages, various functionally related programs are linked together to act as a system module and then these package modules are interlinked to get a complete software system or package system.

(i) Design review: Here the design is checked and reviewed.

2.10 OUTPUT DESIGN

There are various methods of representing the output a system. One or more method may be applied in a specific system. To decide which method is more effective and useful for a specific system, the designer must have adequate knowledge on various methods; briefly discussed below:

(a) Filtering method
(b) Key variable method
(c) Monitoring method
(d) Modeling method
(e) Interrogative method

(a) Filtering method;

Filtering is a process of screening or extracting unwanted elements from which some entity as it passes or is communicated, from one point to another. The filtering method has widespread applicability in most organizations. The reporting of costs and sales dollars are two examples which can be used to illustrate the filtering process.

Advantages and disadvantages:

There are two major advantages to utilizing the filtering method; (i) the amount of useless data provided to each decision maker is reduced considerably since the level of detail received is based on individual requirements and (ii) organizational resources are conserved. Eliminating the need to produce massive reports con-

serves data processing resources.

There are two major disadvantages to utilizing the filtering method: (1) implementation is difficult when the threshold of detail among decision makers at the same level varies considerably, and (2) in large and more complex organizations, filtering alone does not provide adequate " action oriented " information to decision makers.

(b) Key Variable Reporting Method

An automobile's condition can be measured by checking compression, oil pressure, ampere and voltage output, and so forth. There are usually at least five key variables (also called "key success factors," "key result areas," and " pulse points ") for an organization as a whole.

Working with experienced personnel of the organization, the system analyst can isolate most of these key variables. In addition, the analyst can examine how decisions are made, where the major decision points are, and the factors that management is concerned about in making decisions. After the key variables that determine the success of the organization have been defined, the information system is designed to report their status, trends, and changes in trends.

With the reporting of key variables, management can see the direction of the current trends in all the key variables and determine whether they are moving the organization in the direction of its goals. More over, predictive key variables reveal developing opportunities that enable management to take early action to capitalize on them. This approach is better than waiting for results to be reported on the annual financial statements

before taking action, even assuming that such information would be included in the financial statements, which might not be the case.

(c) Monitoring Method

The monitoring method is another alternative for reducing the amount of data decision markers receive while still icreasing the amount of relevant information at their disposal. There are three basic ways to implement the monitering method (1) Variance reporting, (2) Programmed decision making, and (3) Automatic notification.

Advantages and Disadvantages

To summarize the discussion of the monitoring method, its major advantages are presented here.

1. Widespread applicability.
2. Provides a high level of action-oriented information.
3. Relieves decision markers from routine and tedious decision-making activities.
4. Adaptable to most approaches to management (e.g., mana--gement by objectives, management by costs, management by budget,etc.
5. Improves utilization of organizational resources.

The major disadvantages are:

Requires a high level of sytems analysis and design.

2. Requires a clear difination of how things are or should be.
3. Requires a large amount of data collection, storage, and processing activity.
4. Requires sophisticated hardware and software develop--ment.

(d) Modeling Method

The use of models to transform data into information is becoming increasingly important as a means of providing information needed by tactical-level decision markers. In many instances modeling is the only method that is capable of providing this information. While some logico-mathematical models require the model builder to possess a high degree of proficiency in mathematics, the vast majority of these models require a minimum of mathematical expertise.

In order to produce information, a model is usually a verbal or mathematical expression describing a set of relationships in a precise manner. A model can be useful simply in explaning or describing something, or it can be used to predict actions and events.

Advantages and Disadvantages

To summarize the discussion of the modeling method, the major advantages and disadvantages in using models are listed here. The major advantages of using models are that they:

1. Provide action-oriented information.
2. Provide future-oriented information.

3. Permit alternative courses of action to be evaluated before implementation.
4. Provide a formal, structured description of a complex problem situation.
5. Represent a scientific approach to replace intuition and speculation.

The major disadvantages are:

1. Ursers of the model tend to lose sight of the fact that the model represents an abstaction of reality and not reality itself.
2. Qulitative factors such as experience and judgment are minimized or eliminated.
3. The model building process is often very dificult and expensive.
4. Potential users of the model often have a fear or resistance to change which results in dificulties implementing the model
5. Many models assume linearity , a condition that is not applicable to most ''real world'' situations.

(E) Interrogative Method

In the interrgative method , the decision marker is required to request needed information from the system. This method of providing information is extremely valuable, since many decision markers are unable to indentify what information is necessary to perform their duties until the situation confronts them.

The essential elements of this method are:

(1) The information requestor needs only to format or structure his or her inquiry and submit it to some access mechanism or interface, and

(2) The information is presented to the requestor in a usable format and in a relevant time period. To implement the interrogative method it is necessary that an extensive data base exist, organized in a manner where a variety of users can access needed data elements.

Advantages and Disadvantages:

Major advantages:

1. Widespread applicability.

2. Permits each decision maker to obtain relvant, specific information when it is required.

3. Allows previously unanticipated quiries to be entered and processed.

4. Reduces paperwork (and paper pollution).

5. Reduces the time required to disseminate information.

6. Supports other methods of producing information such as filtering, monitoring and modeling.

7. Alleviates organigational controversy by allowing each decision maker independent access to a common data base.

The major disadvantages of the interrogative method are:

1. Requires an expensive investment in data processing resour--ces. This includes not only hardware, but also analysis, design, development and implementarion.
2. It has proved to be almost imposible to provide the necessary database required to respond to more than a small percentage of requests that one or more decision makers might structure.

2.11 DATABASE DESIGN

One of the major step in design of computerized IMS is to design a database. According to conventional concept, designing a database involves :

(i) Specifying the output information. These are obtained from the detail andysis of the existing system.

(ii) Determining the input information needed to obtain the specified output. In this stage different entities and corresponding attributes of each system models are determined to develop ER models and then converted into tables.

(iii) Organizing the input into a database. In this stage, data redundancy is minimized by optimizing the database(s) using the functional dependancy and normalization technique. During normalization ; the following properties of decomposition must be maintained.

(i) Lossless joint decomposition

(ii) Dependency preservation.

(iii) No repetition of information.

Recently, entity-relationship (E R) modeling has been found [18] most successful in design of relational databases. One of the reasons for itseffectiveness is that it is a top down approach using the concept of abstraction. The number of entities in system is much less than the number of data elements or attributes. Therefore, using entities as anabstraction for data elements and determining the inter-entity relationship greatly simplifies the system analysis. The basic steps in this methodology are mentioned bellow:

a. Development of the extended E R model for the information system under consideration.
b. Transformation of the extended E R model to relations.
c. Normalization of the relations.
d. Physical design and implementation.

a. EXTENDED E R MODELING:

The ER approach as proposed by Chen [18] still remains the premier model for conceptual design. This model represents information in terms of entities, their attributes and relationship among attributes. Other researchers have focused their extensions primarily on E R model, in particular the abstraction concepts such as generalization hierarchy, subset hierarchy and aggregation. Composite relationships and attributes were also studied by Ling [18] in 1985. A number of researchers (Martin [18] 1983, Hawryszkiewucz [18] 1984, Briend [18] 1985) devoted their works in transformation of the E R model to the relational model.

Chen's E R model has three class of objects; attributes, entities and relationships. Extended E R model introduces two

additional types of objects; subset hierarchy and generalization hierarchy. The concept of aggregation is also introduced to express relationship among relationships. Aggregation is an abstraction through which relationships are treated as higher level entities. Fundamental extendedE R constructs are illustrated in fig. 2.7. Subset hierarchy: An entity E1 is a subset of another entity E2 if every occurrence of E1 is also an occurrence of E2. For example, the entity 'employee' may include 'employee attending university', 'employee holding political office' or 'employee holding share'.

Generalization hierarchy: An entity E is generalization of the entities E1,E2,E3,.....En. if each occurrence of E is one and only one of the entities E1,E2,E3,...En. For example, the entity 'employee' is a generalization of 'Engineer', 'Director', 'Teacher' and 'Technician'.

Degree of relationship: The degree of relationship is the number of entities associated with the relationship. Unary, binary, tenary and n-ary relationships are of degree 1,2,3, and n respectively.

Connectivity of a relationship: The connectivity of a relationship specifies the mapping of the associated entity occurrences in the relationship. Basic constructs for connectivity are one to one, one to many and many to many. In EER diagram, a shaded corner denotes 'many' and an un-shaded corner denotes 'one'. An entity in a tenary relationship is considered to be 'one' if only one occurrence of it can be associated with one occurrence of each of the other two associated entities. It is 'many' if more than one occurrence of it can be associated with one occurrence of each of the other two associated entities.

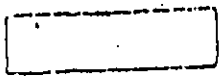
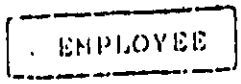
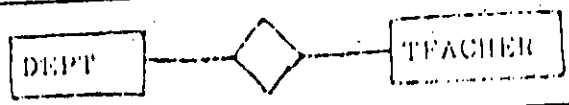
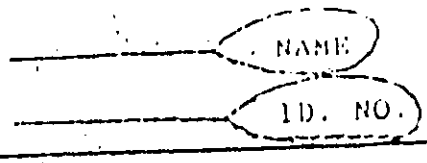
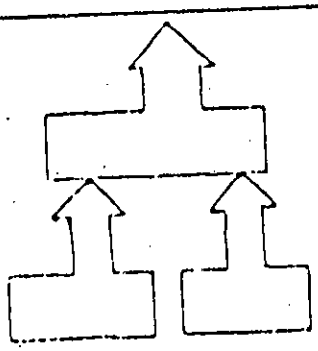
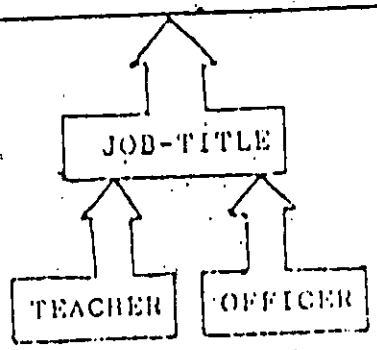
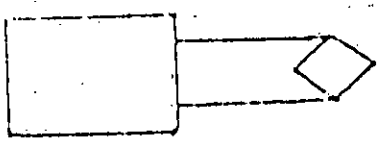
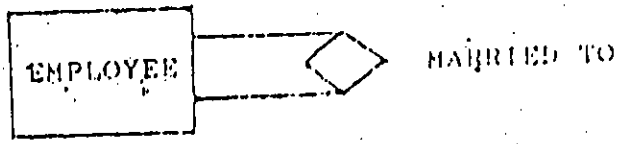
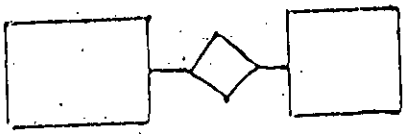
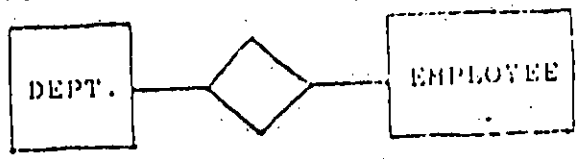
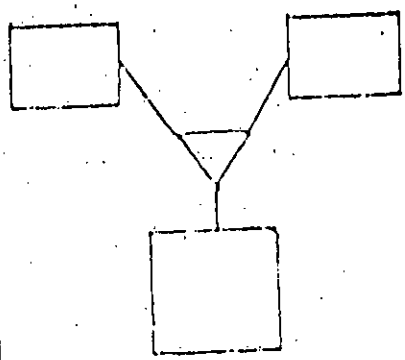
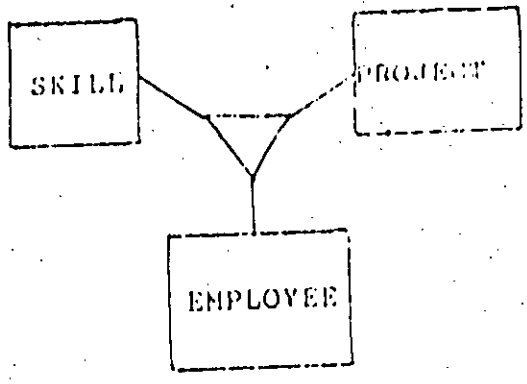
CONCEPT	REPRESENTATION	EXAMPLE
ENTITY		
RELATIONSHIP		
ATTRIBUTE: Descriptor Identifier		
SUBSET HIERARCHY		
GENERALIZATION HIERARCHY		
DEGREE OF RELATIONSHIP: Unary		
Binary		
Tenary		

Figure 2.7a Fundamental Extended E R constructs.

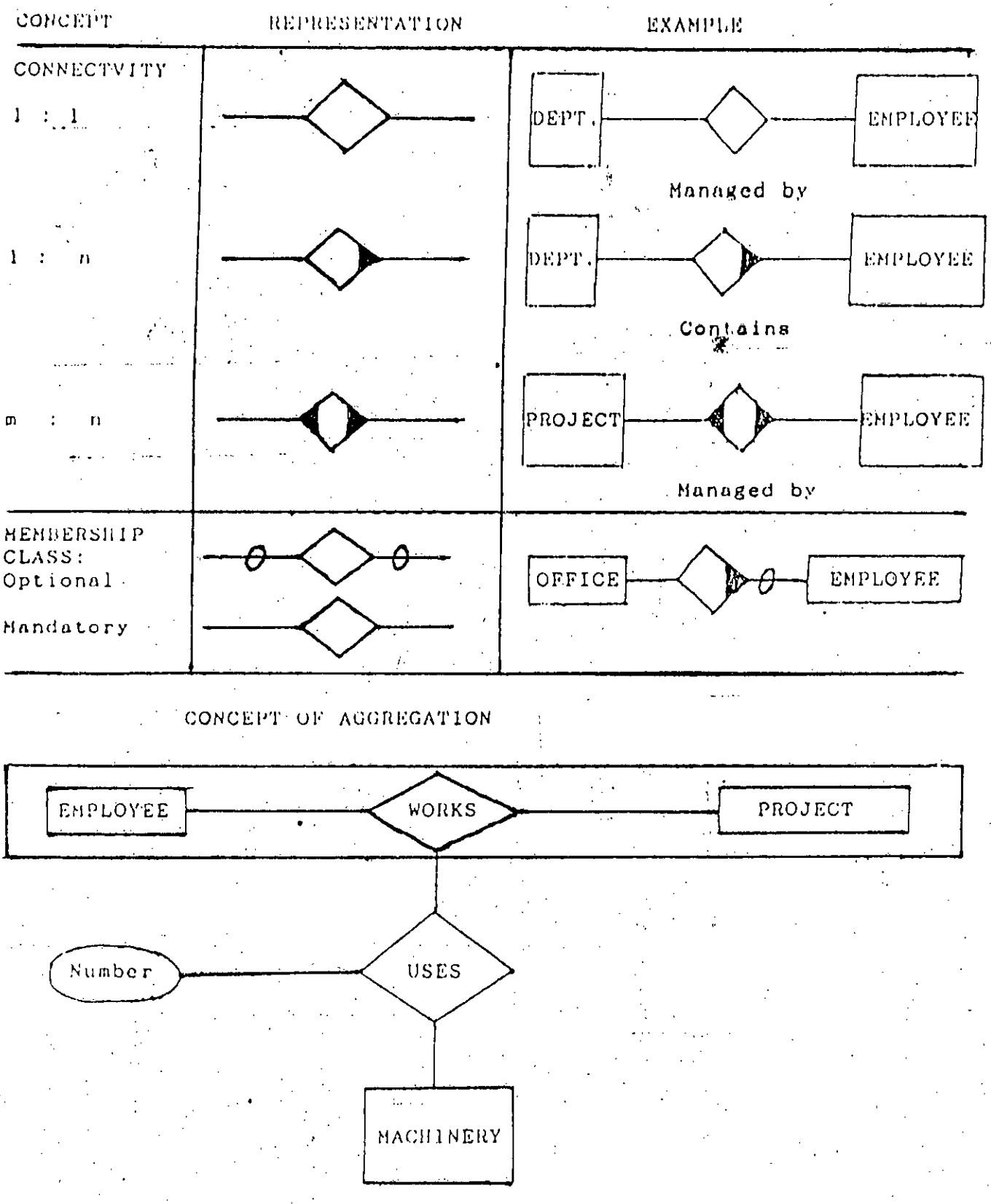


Figure 2.7 b Fundamental Extended E R constructs.

MEMBERSHIP CLASS IN A RELATIONSHIP: Membership class specifies whether either the 'one' or 'many' side in a relationship is mandatory or optional. If an occurrence of the 'one' side entity must always exist for the entity to be included in the system, then it is mandatory. When an occurrence of that entity need not exist, it is called optional. The following steps are involved in development of the extended E R model.

STEP-1 [Identification of entities and attributes]: Although it is easy to define entities, attributes and relationships , it is not so easy to identify them in modeling the database. The following guidelines are very helpful to identify entities and attributes.

(1) Entities have descriptive information; identifying attributes do not.

(2) Multivalued attributes should be classified as entities.

(3) Make an attribute that has a many-to-one relationship with an entity.

(4) Attach attributes to entities that they describe most directly.

(5) Avoid composite identifiers as much as possible.

STEP-2 [Identification of generalization and subset hierarchies]: The existence of generalization hierarchy and subset hierarchy is obvious by their definition. We have to reattach attributes to relevent entities putting identifier and generic descriptors in the generic entity, and identifier and specific descriptors in the subset entities.

STEP-3 [Defining relations]: Relations represent associations among entities. For every relationship, 'degree', 'connectivity', ,membership class' and attributes are specified.

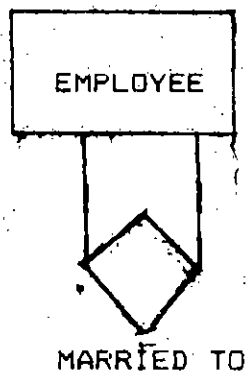
STEP-4 [Integration of multiple views] : When design is large and

more than one person is involved in system analysis, multiple views of data and relationship occur. These views are eventually consolidated into a single global view to eliminate redundancy and inconsistency from the model.

b. TRANSFORMATION OF THE EXTENDED E R MODEL TO RELATIONAL MODEL :

We first look at each E E R construct in more detail to see how each transformation rule is defined and applied. Rules are defined and applied for each class of various relations separately.

Rules for unary relationships: One entity with a one-to-one relationship forms such an entity occurrence and this must be either completely optional or completely mandatory. In both the cases, the pairing entity key appears as a foreign key in the resulting relation and two key attributes are taken from the same domain but are given different names to designate their unique use. The one-to-many relationship requires a foreign key in the entity relation for both the optional case, with nulls allowed, and the mandatory case with nulls not allowed. The following figures illustrate the application of these rules to develop a relational model from an unary relationship E R model.

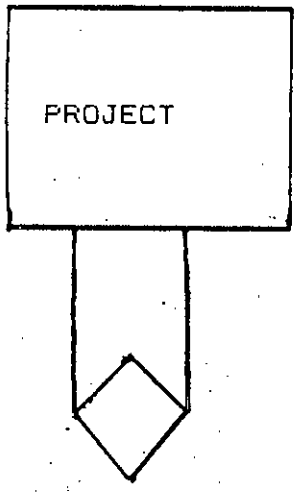


An employee could have one of the other employee as his or her spouse.

Relations:

EMPLOYEE (Emp-no, ...sp_Emp-no.)

Null Sp_Emp-no is allowed in EMPLOYEE



Each project may require special communication with many other projects.

Relations:

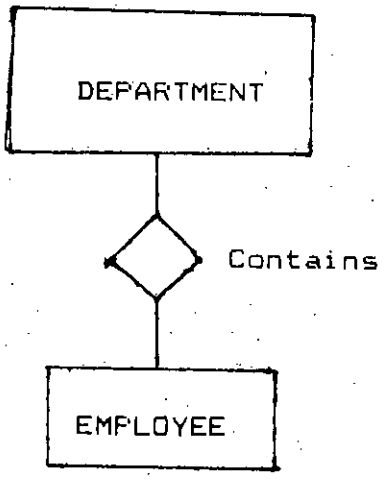
PROJECT (Proj-no, ...)

SP_COM-WITH (Proj-no, Rel_proj-no).

Rules for binary relationships: For one to one relationship, when both entities are mandatory, each entity becomes a relation, and the key of either entity can appear in other entity's relation as a foreign key. When one side is optional, the other side entity contais foreign key with nulls allowed. When both entities are optional, either entity could contain the embeded foreign key of the other entity, with nulls allowed.

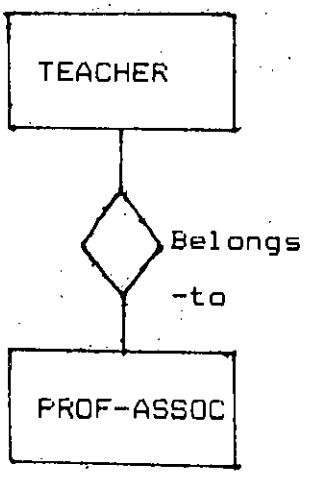
The one to many relationship may be either optional or mandatory in both sides. In all cases the foreign key must appear on the 'many' side, which represents the child entity, with nulls allowed for foreign keys only in the optional 'one' side.

The many to many requires a relationship relation with primary keys of both entities. The same transformation applies to either optional or mandatory case. The following figures provide illustration of these rules.



Each employee works in exactly one department (or office). Every deptt. could contain many employee.

Relations:
DEPARTMENT(dept-code, dept-name, ...)
EMPLOYEE(Emp-id, dept-code, ...)
Null dept-code not allowed in EMPLOYEE

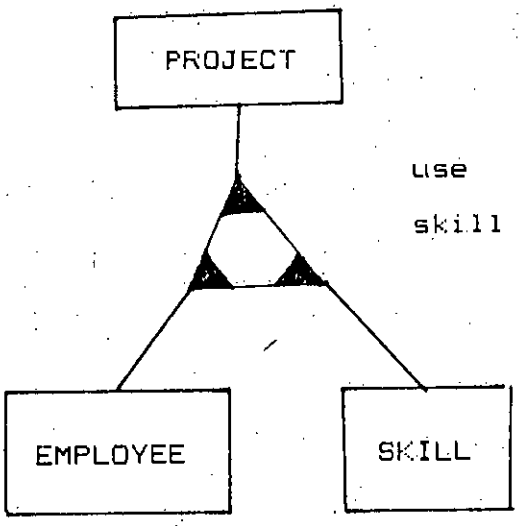


Every professional association could have many teachers or others. Each teacher could belong to many professional association, or none.

Relations:
PROF-ASSO (PA-name, ...)
TEACHER (Emp-id, ...)
BELONGS-TO(Emp-id, PA-name)

Rules for n-ary relationships: In an n-ary relationship, there are n+1 possible varities of connectivities: all n sides with connectivity 'one', n-1 sides with connectivity 'one' and one sides with 'many', n-2 sides with connectivity 'one' and two sides with 'many', and so on until all sides are 'many'. Thus in a tenary relationship, there are four possible varities. All varities are transformed by creating a relationship relation containing the primary keys of all n entities. When all relationships are 'one', the relationship rlation consists of three distinct candidate keys ie. there are three funtional dependencies (FDs) needed to describe its relationship. The

optional 'one' allows null foreign keys; the mandatory doesnot. When relationships are 'many' , the relationship relation is of all keys unless the relationship has its own attributes. The follwing example illustrates the application of these rules.



Project use a wide range of employee different skills on each employee assigned with.

Relations:

EMPLOYEE(Emp-id, ...)

SKILL (Skill-id, ...)

PROJECT (Proj-name, ...)

USE-SKILL(Emp-id,Skill-id,Proj-name)

Rules for generalization and subset hierarchy: Transformation of genaralization and subset produces a seperate relation for the whole set (the generic entity) and each subset. The generic entity contains the generic identifier and all common attributes. Each subset contains the generic identifier and the specific attributes. Transformation rules for disjoint and overlapping subsets are the same. The following figure illustares these rules.

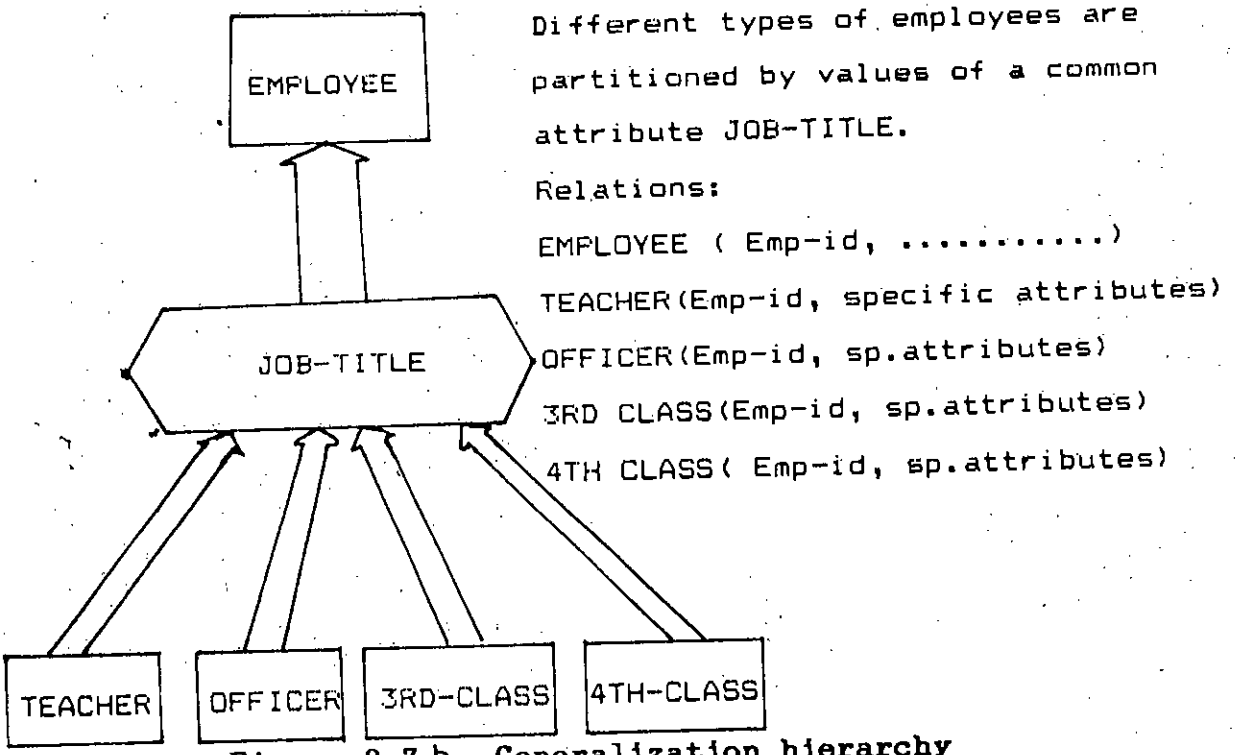


Figure 2.7 b Generalization hierarchy

Transformation of the E E R model to relational model results in a number of relations or tables called candidate relations.

c. NORMALIZATION OF CANDIDATE RELATIONS:

Normalization of candidate relations are accomplished by analyzing the functional dependencies (FDs) and multivalued dependencies (MVDs). Each candidate relation is examined to dtermine what dependencies exist among primary key, foreign key, and nonkey attributes. These dependencies detemine the current degree of normalization of the relation. Any well known techniques for increasing the degree of normalization can now be applied to each relation.

d. PHYSICAL DESIGN AND IMPLEMENTATION:

Physical design involves consideration of some physical

factors like field size, record size, data size, system life time, limitations of hardware & software resources etc. Various concepts and consideration of these physical factors are illustrated in chapter four of this thesis.

2.12 PROCESS DESIGN

Process design is the major step in developing different modules in an information system.It involves development of I P O charts and Decision Logic Tables. A decision table is a tabular representation of the decision-making process. It standarizes the logical process and allows the user to insert the values in both the conditions and actions related to the decision. The underlying premise for utilizing a decision table can be structured as an -if this occurs, then do this- proposition. The IPO chart shows the detail program logic in a module. There are three columns, namely, Input, Process, and Output in the I P O chart. Input specifies various input information required in a module. Output specifies detail logical process needed in a module to obtain the necessary outputs and the output column lists the various outputs of the module. In chapter-4, various decision tables and I P O charts are developed for the proposed I M S.

2.1 QUERY FORMULATION

A query language is a laguage in which a user requests information from database. These languages are typically higher level languages from standard programming languages. Query languages can be catagorized as either procedural or non

procedural language. In a procedural language, the user instructs the system to perform a sequence of operations on the database to compute the desired result. In a non-procedural language, the user describes the information desired without giving a specific procedure for obtaining that information. Relational algebra is the procedural query language, relational calculus is the non-procedural query language. Most commercial relational database systems offer a query language that includes elements of both procedural and non-procedural approaches. Commercial query languages are: Query language(Quel), Query By Example(QBE) and Structured Query Language(SQL).

#Relational Algebra: There are five foundamental operations in relational algebra. These are: 'select', 'project', 'cartesian-product', 'union' and 'set-difference'. All of these operations produce a new relation as their result. In addition to the five foundamental operations, there are several other operations, namely, 'set intersection' 'theta join' 'natural join' and 'division'. The 'select' and 'project' operations are called unary operations, since they operate on one relation. The other three foundamental operations operate on pairs of relations and are, therefore called binary operations.

The **select** operation selects tuples that satisfy a given predicate. The **project** operation copies its argument relation, with certain columns left out. The **cartesian product** is used in conjuntion with **select** and or **project** operation inorder to extract information from more than one relation. The **set-difference** operator allows us to find tuples that are in one relation, not in another.

#Relational calculus: There are two forms of relational calculus,

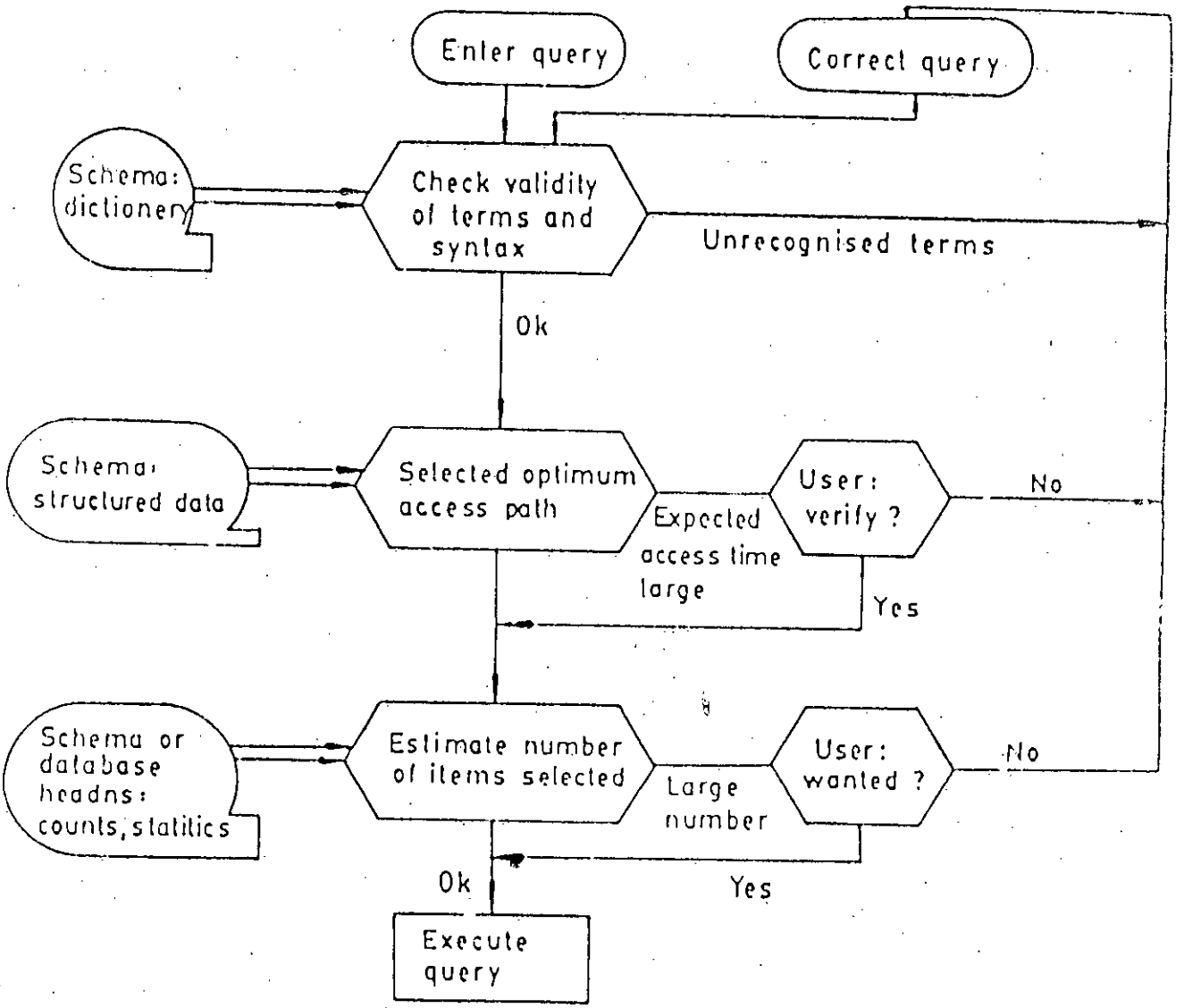


FIG 2.8 QUERY FORMULATION [19]

one in which the variables represent tuples, and one in which the variables represents values of domainsThese varients are called the **tuple relational calculus** and the **domain relational calculus** These two forms are very similar. A query in the tuple relational calculus is expressed as {t/P(t)}, that is the set of all tuple t such that predicate P is true for t.

#Commercial query languages(CQL): Among the three types of CQL, mentioned abobe, the SQL is accepted in recent trends of query formulation. The basic structure of an SQL expression consists of three clauses: **select**, **from** and **where**.

*The **select** clause corresponds to the poject operation of the relational algebra.It is used to list attributes desired in the result of a query.

*The **from** clause is a list of relations to be scanned in the execution of the expression.

*The **where** clause corresponds to the selection predicate of relational algebra. It consists of a predicate involving attributes of the relations that appear in the from clause.

Atypical SQL query has the form:

select $A_1, A_2, \ldots\ldots A_n$

from $r_1, r_2, \ldots\ldots r_m$

where P

The A_is represent attributes, the r_is represent relations and P represent a predicate.

2.15 SYSTEM DEVELOPMENT

From the discussions presented in this chapter it is evident that the development of an I M S includes analysis of the existing system, design of a new system and development of the

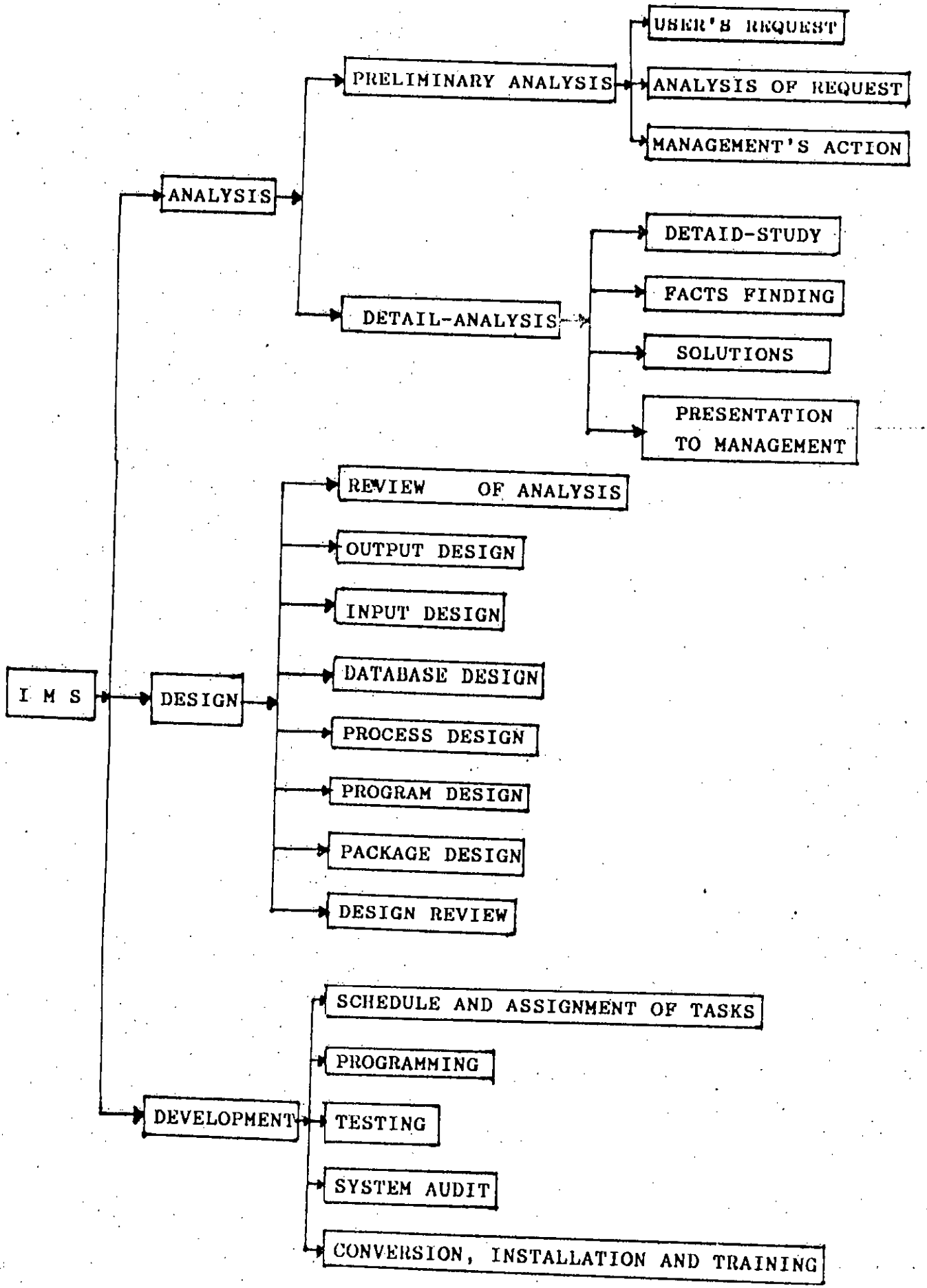


Figure 2.9 Stages of developing I M S

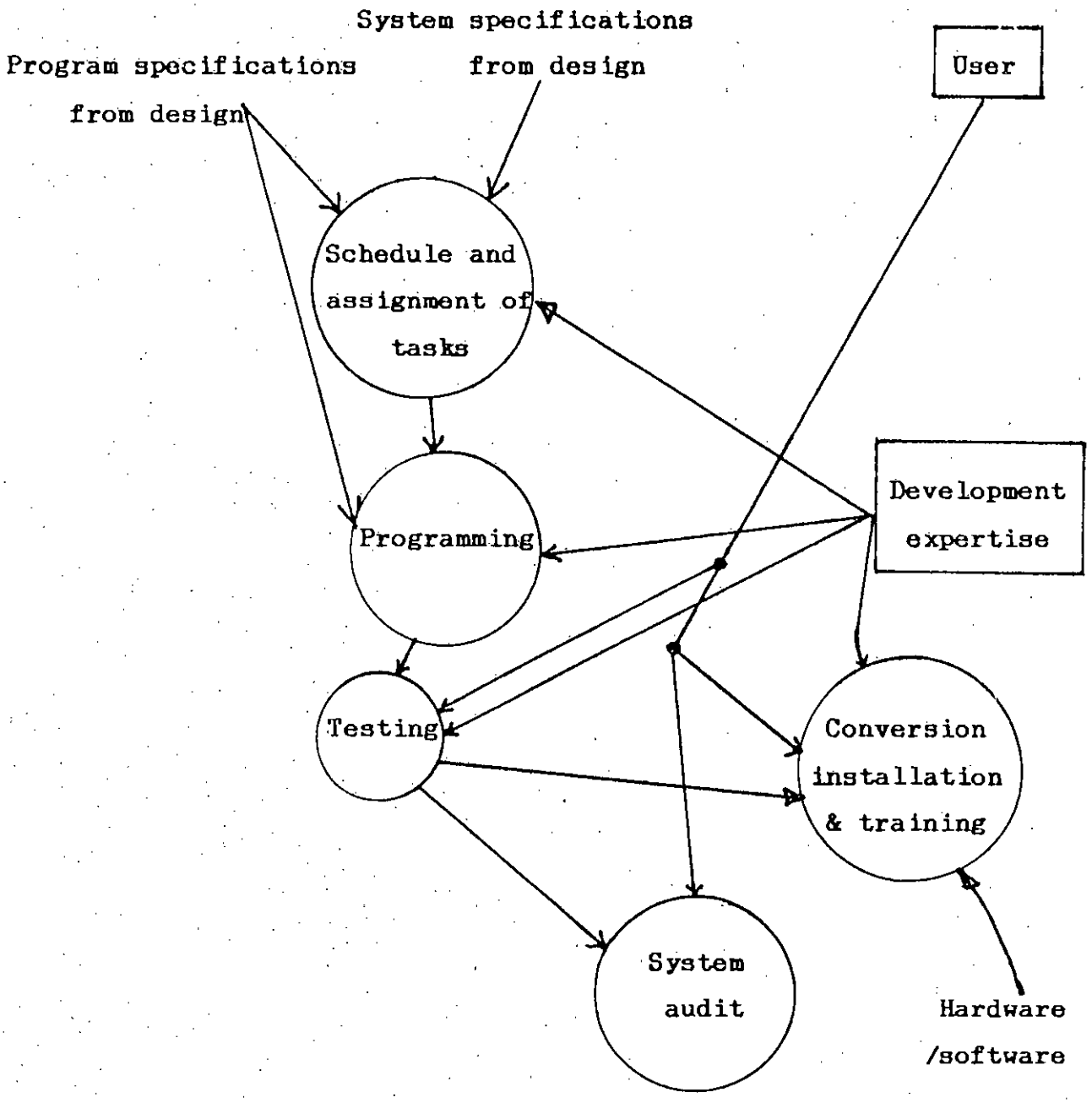


Figure 2.10 Data flow diagram for development phase[17]

proposed system. Analysis phase includes primary and detail analysis. Design comprises of output design, input design, database design, program design, module design, package design etc. and development phase includes schedule and assignment of tasks, programming, testing, conversion. installation, training, system audit etc. This overall development process iis illustrated in figure 2.9 and figure 2.10 represents the data-flow diagram for the development phase according to recent trends.

CHAPTER THREE

ANALYSIS OF THE EXISTING SYSTEM

CHAPTER OUTLINE:

This chapter presents preliminary and detail analysis of the existing systems, alternative solutions and recommendation of a computerized system with justification and a gantt chart showing the time schedule for design and development of the recommended system.

3.1 PRELIMINARY ANALYSIS

Analysis is the most important phase of designing information management system, because errors in analysis will be carried on to design and coding.Thus even the programs satisfy the design specifications, the system will not satisfy the users. The preliminary analysis phase is decomposed into the following three steps:

a.Evaluation of user's request

b.Analysis of the request

c.Management's action

A data-flow diagram shown in figure 2.3, decomposes the preliminary analysis phase into its components. Users, management and the computer services staff (System Analysts) provide input at various stages. Preliminary report and documents are outputs of this phase.

a.Evaluation of user's request:

Preliminary study of the information systems of B U E T Dhaka shows that there are 46 offices / departments each of which independently process information. These are mentioned in appendix A 1.

Study of the Organizational structure and records depicts that there are 1565 operating manpower, about 2500 students, a number of contractors and suppliers, some donating bodies (like UNDP), govt. Offices (like UGC, Secretariats etc.), some international universities / organizations and a large number of govt. / non -govt. organizations involved in the information systems of B.U.E.T. Thus there is a highly complex information system that must be analyzed and modularized.

b.Analysis of the request:

The existing system is analyzed through interviews and personal contact with the operating manpower. Before approaching them, technical preparation is taken. They have been given the understanding that this analysis job is to develop a new computer-based system that will just help them but never replace them. A closed co-operational environment is created in order to prepare them for responding readily and heartily to questions needed for analysis. The facts and information found are organized and summarized in the preliminary report.

3.2 PRELIMINARY REPORT

On the basis of the preliminary analysis made on information systems of B U E T Dhaka, the report is presented below in structured manner:

PROBLEM REVIEW: The main problem is the poor performance. Most of the operating manpower is overloaded with their jobs. Preliminary investigating depicts that a new system based on computer can be designed and developed that will solve all problems and improve performance to a high level. The quantitive analysis of the obtainable performance and the existing performance is shown below (assuming a standard manual system of doing 40 basic jobs per day by a manpower strength of 20).

System	Manpower	Jobs/day	Time reqd.	Jobs/man-hr.
Existing system	20	40	8 hrs	.25
New system	2	200	4 hrs	25=.25*100

FINDINGS: Facts and information found during system study, interviews and investigation are summarize below:

(1) The system is too large and too complex to be efficiently

managed manually by current strength of manpower.

(2) There are above 20 independent systems to be analyzed separately.

(3) The work load of every system is heavy; specially on accounting and administrative systems.

(4) The most important and central systems which are used by maximum users are administrative and accounting systems.

(5) The existing system is running very slowly, roughly and inefficiently.

(6) A large number of users are involved information systems.

(7) Since that current strength of manpower is not sufficient to manage the system efficiently there are provisions of adopting unfairmeans by the related personnel.

RECOMMENDATIONS:Following recommendations are made on the basis of the preliminary analysis of the existing system.

(1) The information systems should be decomposed into around 20 independent systems.

(2) Each independent system should be analyzed separately.

(3) Administrative and accounting systems should be given preference for computerization.

(4) A detail analysis should begin immediately on administrative and accounting systems to determine the feasibility study of computerizing the existing systems.

(5) System analyst should be authorized to begin detail analysis on these systems.

COST AND SCHEDULE: Cost and time schedule for detail analysis on administrative and accounting systems of B U E T Dhaka is estimated below:

1. Six week for system analysis(including TA and DA) Tk=7200.00
2. Secretarial aid (30 hours) Tk= 600.00
3. Two weeks for other staff (interview & discussion)Tk= 200.00

Total Eight weeks: Tk=8000.00

The Gantt chart shown below is the graphical representation for the time schedule of detail analysis.

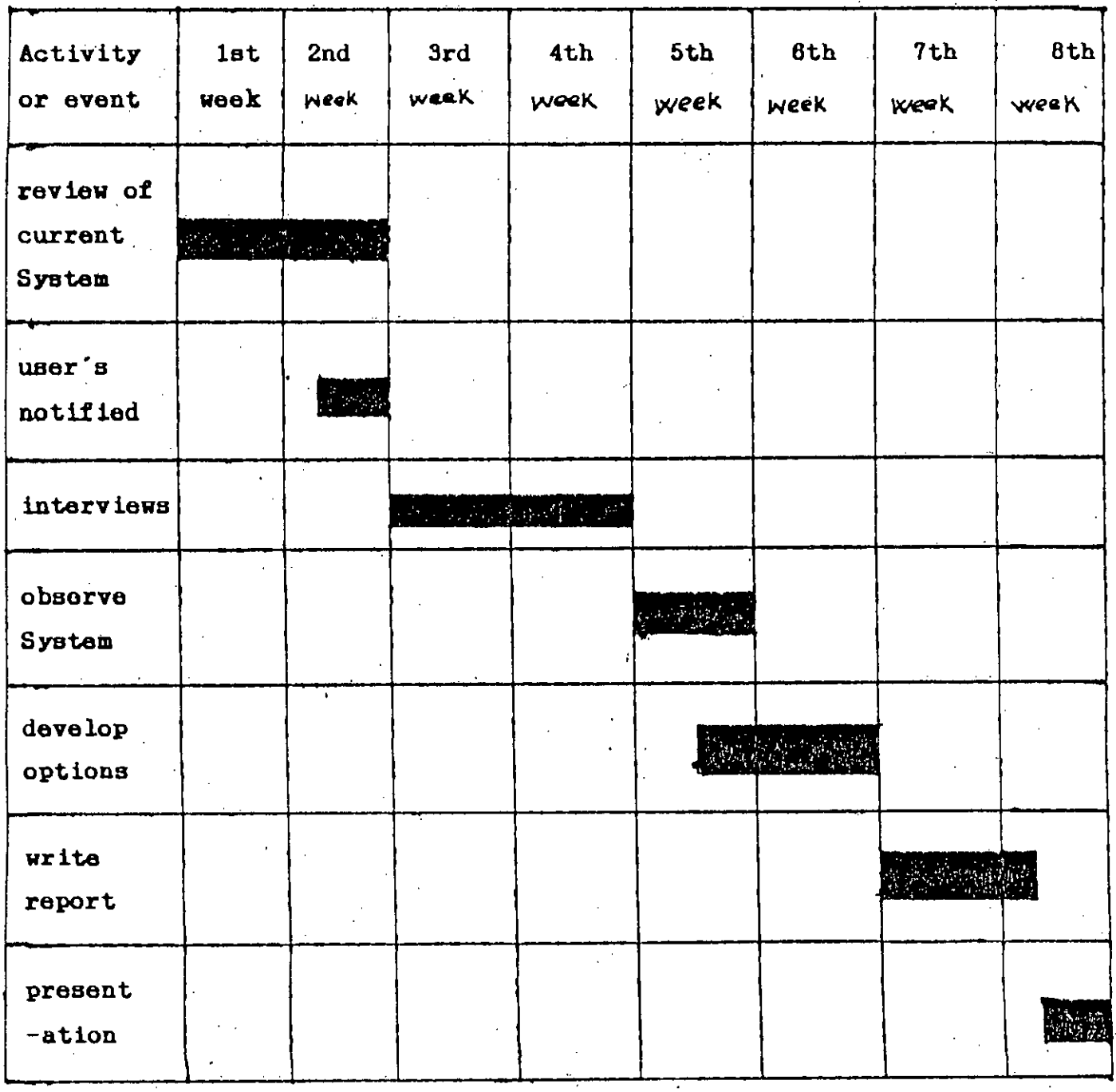


Figure 3.1 Gantt chart for detail analysis

3.3 DETAIL STUDY AND SYSTEM CHARTING

According to the recommendations in preliminary report, accounting and administrative systems of B U E T Dhaka are taken under detail analysis. Functional flow diagram for the detail analysis phase is shown below:

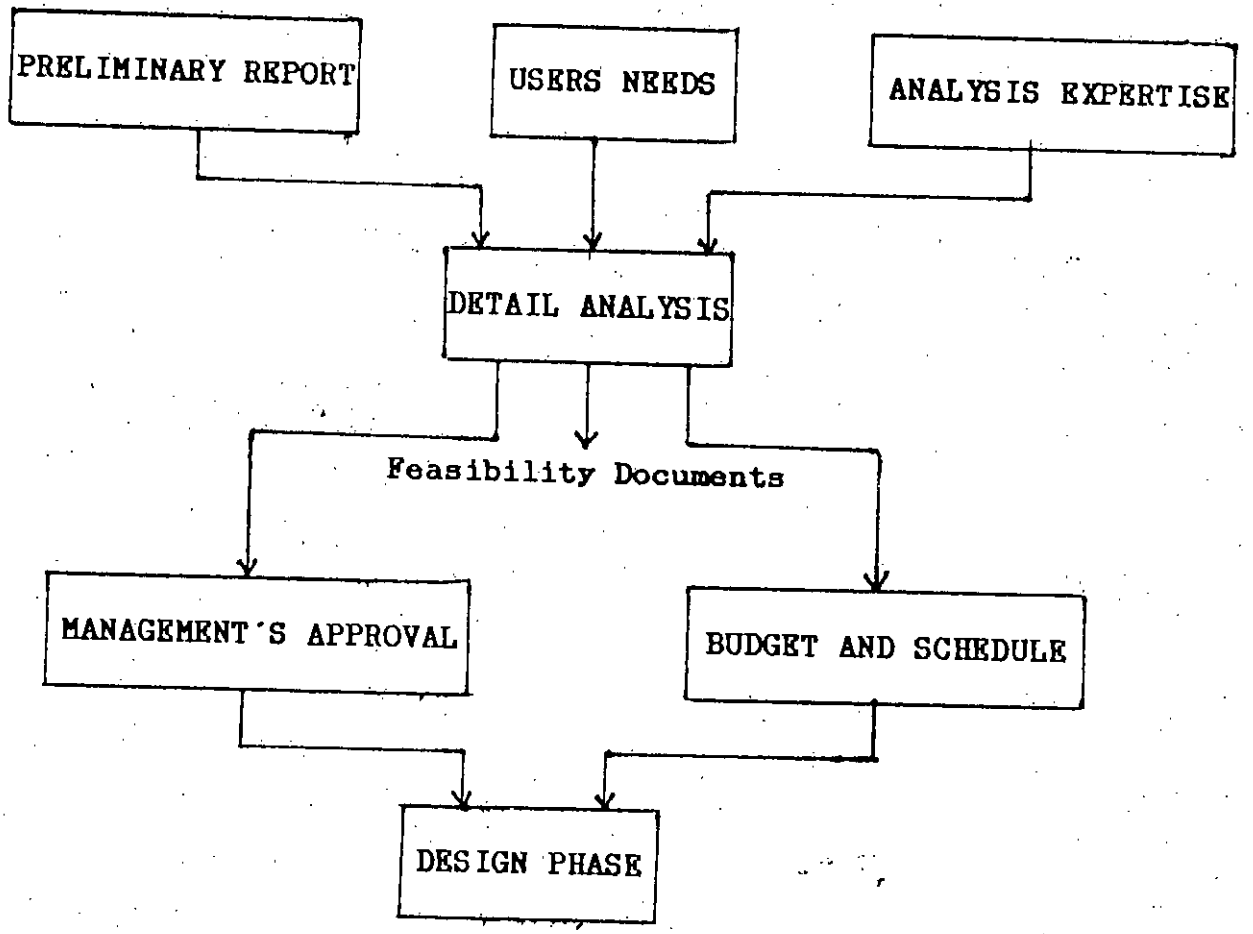


Fig. 3.2 Functional flow diagram for detail analysis

In detail analysis, the user's needs, approved preliminary report and the analysis expertise (ie Supervisors) are inputs whereas the detail analysis report, documents and the gantt chart are outputs.

For detail study, the system is reviewed, users and the operating manpower are notified and then various facts and information are obtained to develop system modules efficiently. The assigned part of the information processing system comprises the following modules:

1. Salary Billing
2. Salary Management
3. Non-salary Heads Management
4. Income Management
5. Summary Report
6. Service Reports

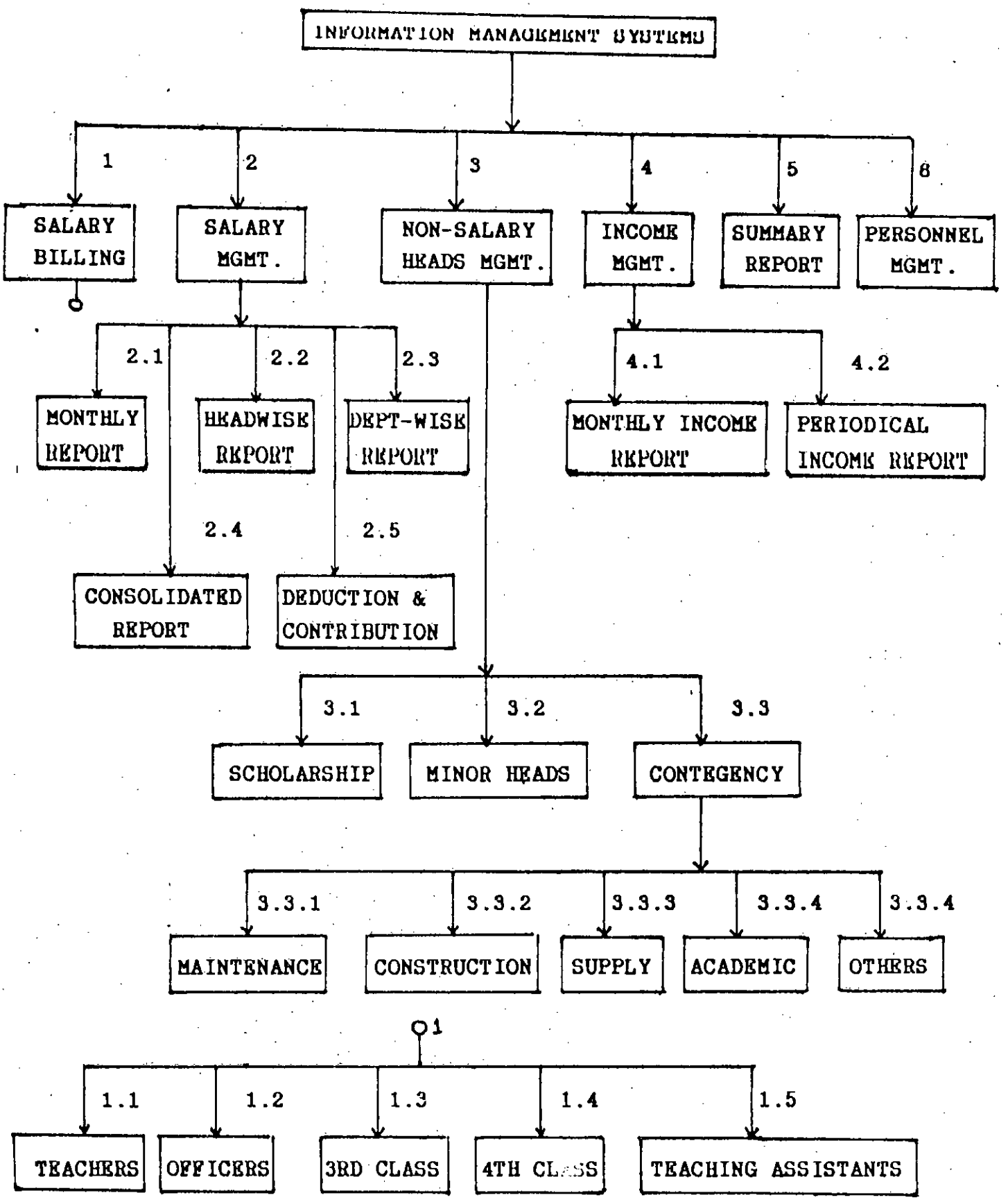


Figure 3.3 System charting

7. Personnel Management

Each module is devided into various sub-modules in figure 3.3 . This process is called system charting.

3.4 QUESTIONNAIRE, INTERVIEWS AND FACTS

These facts-finding techniques as described in article 2.4 are used in analyzing the existing system. Interviewing is used to gain information concerning what is required and how these requirements can be met. Interviewing is conducted at all levels within the organization. It is found as a communication channel between the analyst and the organization.

Questionnaire is used at various times in the system development process. But this technique is limited to only those situations where an interview cannot be conducted. During questionnaire (i) the purpose, use, security and disposition of the responses are clearly explained, (ii) a time limit is given for return of the questionnaire, (iii) detail instructions on how the questions should be answered are provided, (iv) pointed and concise questions are asked, (v) each questionnaire is identified byrespondent's name, job title, department etc. and (vi) a section is included where respondents can state their opinions and criticisms.

Various information obtained by these facts-finding techniques are used in development of the data flow diagram and alternative solutions.

3.5 DATA FLOW DIAGRAM

Data (or information) flow diagram in existing administrative and accounting system of B U E T Dhaka based on the detail analysis is presented in figure 3.4.

3.6 ALTERNATIVE SOLUTIONS

Following are alternative solutions to the problems in existing manual information processing systems in accounts and administration of B U E T Dhaka based on detail analysis.

1. DO NOTHING : Leave the system alone. No improvement of performance, no cost and no benefit. As the time goes, the system will get worse, and morale will drop.

2. HIRE MORE STAFF : Continue with a manual system.

Cost:

Tk= 480000.00 for first year

Tk= 600000.00 for second year and so on.

Currently this will stabilize the system but will face the same problem after a few years. Performance will not be in expected level owing to drop of morality and human factors.

3. PURCHASING SOFTWARE:

Administrative and accounting software is very costly; price ranges from 1.5 lacs to 3.5 lacs. Performance will be improved but cost will be high.

Expected cost of such a solution is estimated below:

1. Software package	Tk= 250000.00
2. Modification for B U E T	Tk= 30000.00
3. Yearly software update	Tk= 20000.00
4. Training of staff	Tk= 20000.00
5. Equipments	Tk= 130000.00
Total	Tk= 450000.00

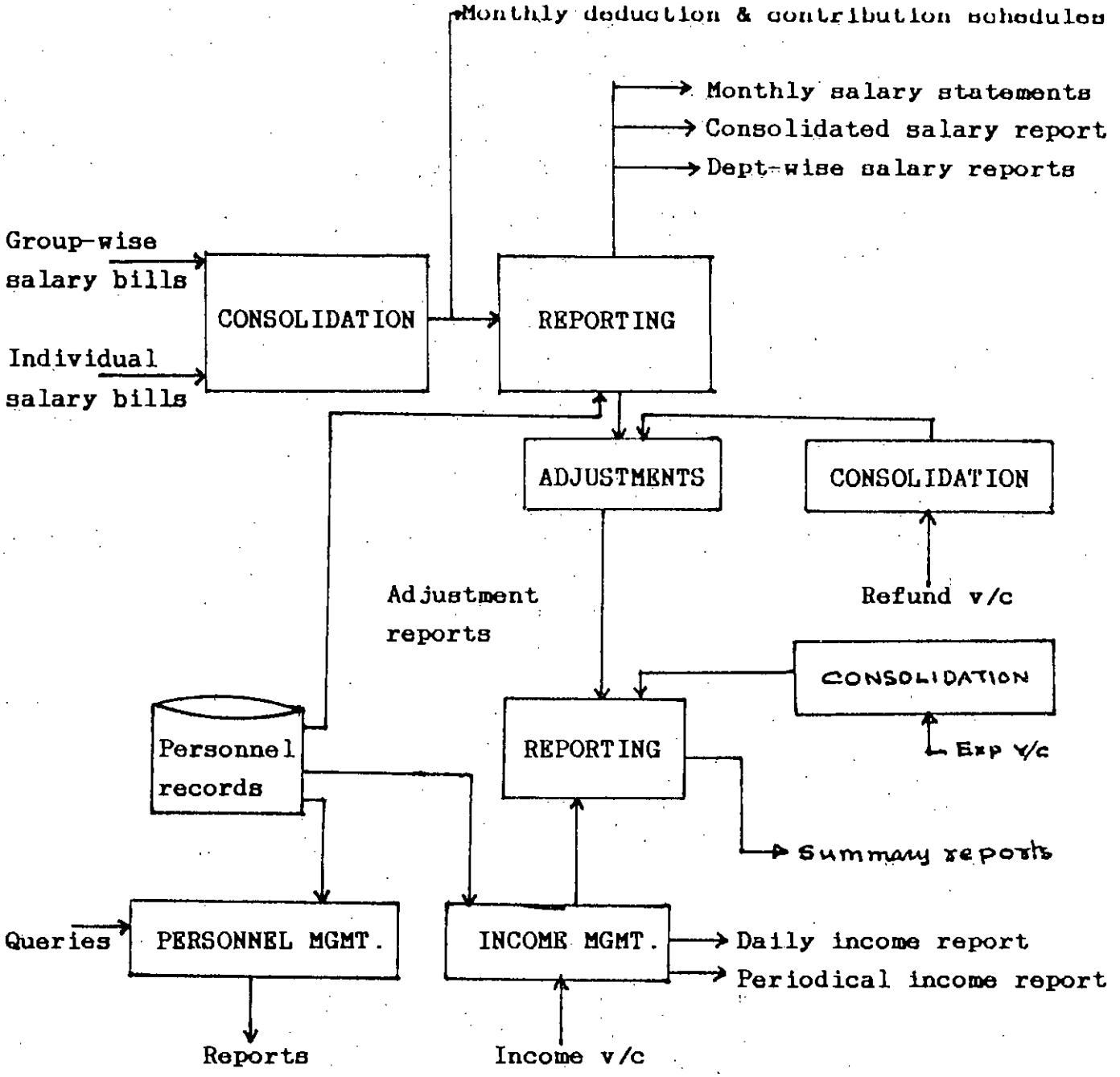


Figure 3.4 Data flow diagram in existing system

4. DEVELOPING CUSTOMIZED SOFTWARE:

Performance will be improved to a very high level and cost will be as estimated below:

1. System design	Tk= 50000.00
2. Programming	Tk= 20000.00
3. Yearly system maintenance	Tk= 8000.00
4. Training of staff	Tk= 12000.00
5. Equipments	Tk=130000.00
Total	Tk=220000.00

3.7 SYSTEM RECOMMENDATION

The solution (1) is rejected, as it shows no gain or no loss. The other solutions are analyzed to get relative performances in the next five years. In the following table, F is the future value of performance, P is the present value of performance, C is the cost of improving performance. P, F and C are all approximate values. To estimate the values of P and F it is assumed that performance of the existing system is 1% and the highest attainable performance is 100%. Units of P and F are in %, and that of C is Taka in thousands.

TABLE 2 Analysis of performances

Year	Solution (2)			Solution (3)			Solution (4)		
	P	F	C	P	F	C	P	F	C
1st	1	1	48	1	50	450	1	100	220
2nd	1	1	60	50	50	20	100	100	8
3rd	1	1	72	50	50	20	100	100	8
4th	1	1	84	50	50	20	100	100	8
Total	1%		264	50%		510	100%		244

On the basis of the above analysis, the solution (4) is recommended because of its high performance and overall minimum cost. Thus the budget of the recommended system is Tk= 220,000.00 only. Time schedule for the proposed I M S for administrative and accounting system of B U E T Dhaka is presented in the following section.

3.8 GANTT CHART FOR DESIGN AND DEVELOPMENT

Time schedule for the proposed I M S is presented in the following gantt chart:

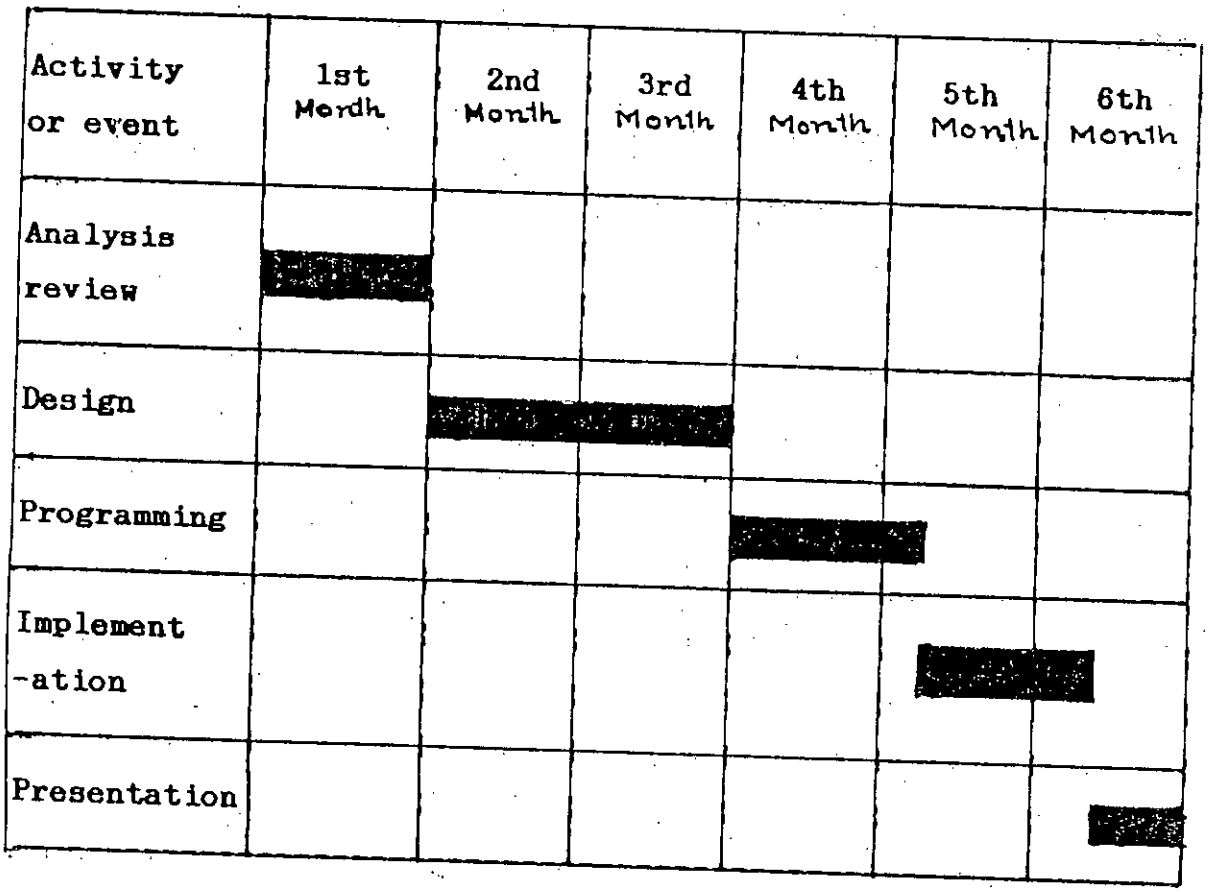


Figure 3.5 Gantt chart for design and development

CHAPTER FOUR

DESIGN OF A COMPUTERIZED I M S

CHAPTER OUTLINE :

4.1 Design of system modules
4.2 Design of conceptual model
4.3 Design of logical model
4.4 Database dictionary
4.5 Different files in the physical system
4.6 Relation between files
4.7 Output design
4.8 Input design
4.9 Database design
4.10 Design of menu structure
4.11 System operation and maintenance
4.12 Process design
4.13 Program design
4.14 Package design
4.15 Design of security and control

In this chapter, a complete design of the proposed I M S for administrative and accounting systems of B U E T, Dhaka is presented.

4.1 DESIGN OF SYSTEM MODULES

Structured approach of system design is followed to offer a set of strategies for developing a design solution from the well defined statements of the analysis phase. The design actually starts during analysis phase with the completion of funtional specification.

The first step of the design phase is the creation of a structured chart known as **System Grid Chart** which is a graphic tool that shows the partitioning of the system into modules and illustrates the hierarchy and organization of those modules.

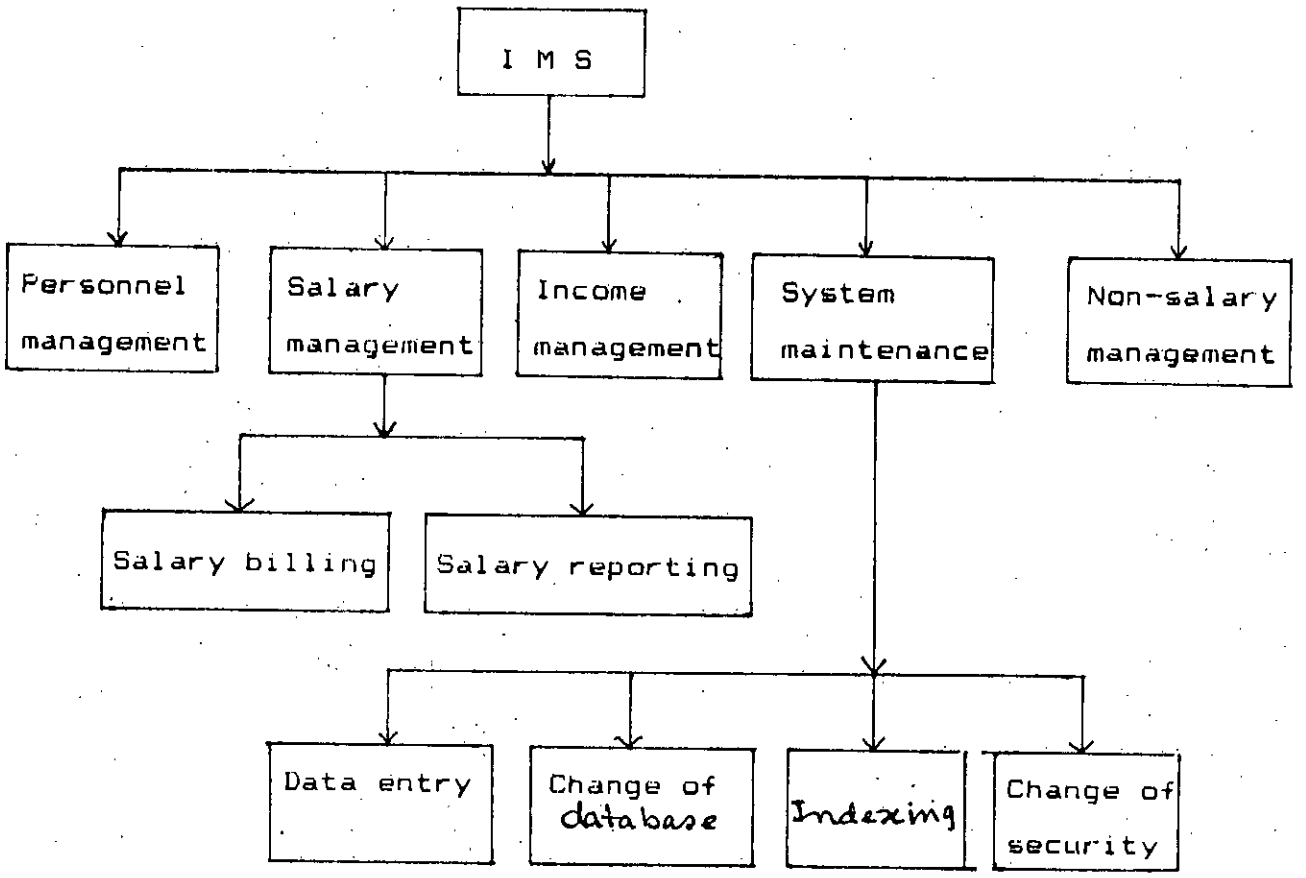


Fig. 4.1 System Grid Chart for the proposed I M S .

4.2 DESIGN OF CONCEPTUAL MODEL

Conceptual model is the combination of several ways used to process the information for different applications. It is independent of DBMS, independent of hardware used for storing data and independent of the physical model of storage media. The objective of the modeling approach is to provide a basis for the design of an effective and useful I M S. In conceptual model, various information processing units, their inputs and outputs, and data flow among them etc. are depicted. A conceptual model of the proposed I M S for administrative and accounting systems of B U E T Dhaka is presented below:

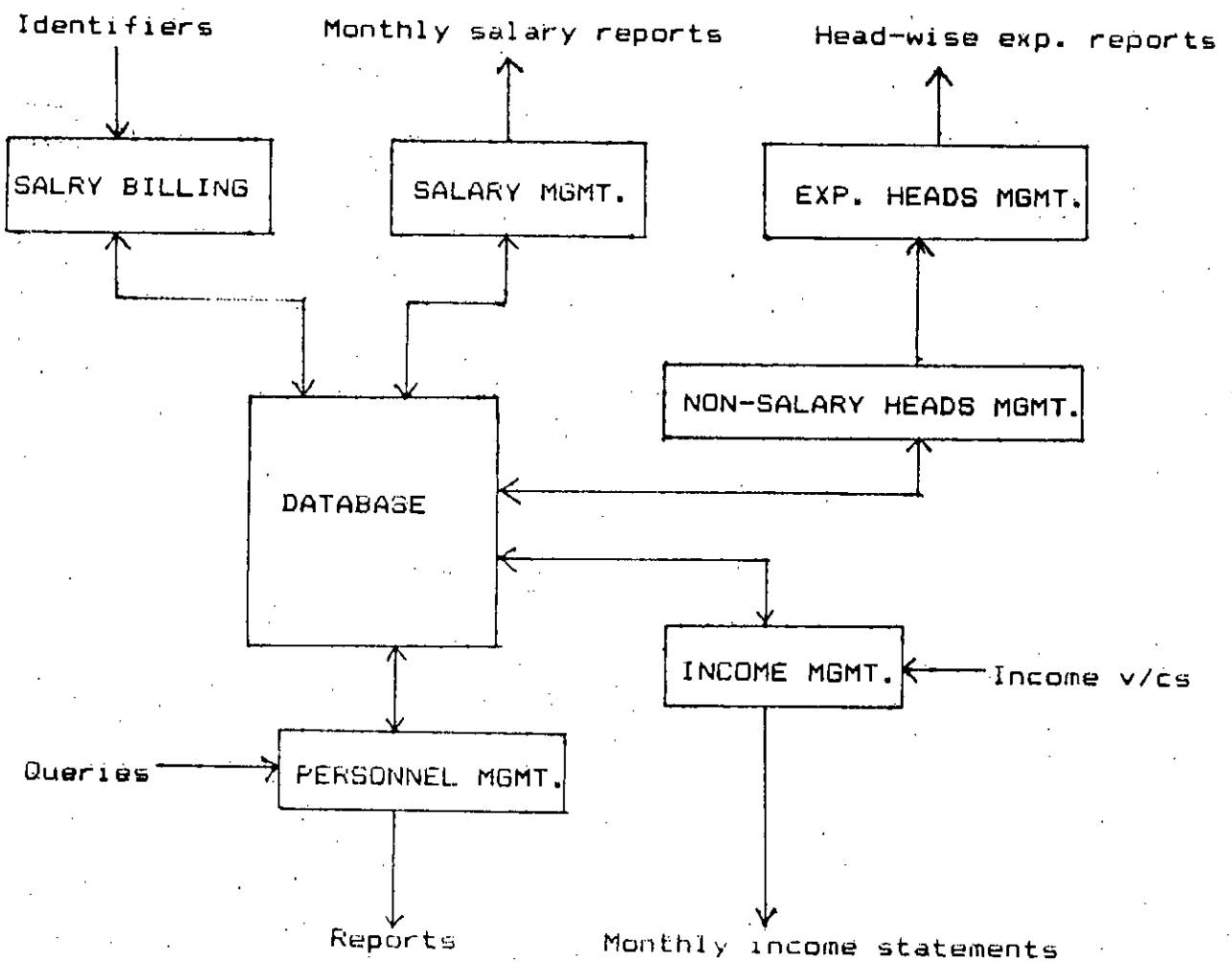


Fig. 4.2 A conceptual model

4.3 DESIGN OF A LOGICAL MODEL

Logical model shows the logical layout of the flow of information among various system modules and their inputs and outputs. It is developed on the basis of the conceptual model. A logical model for the proposed I M S is presented in figure 4.3 .

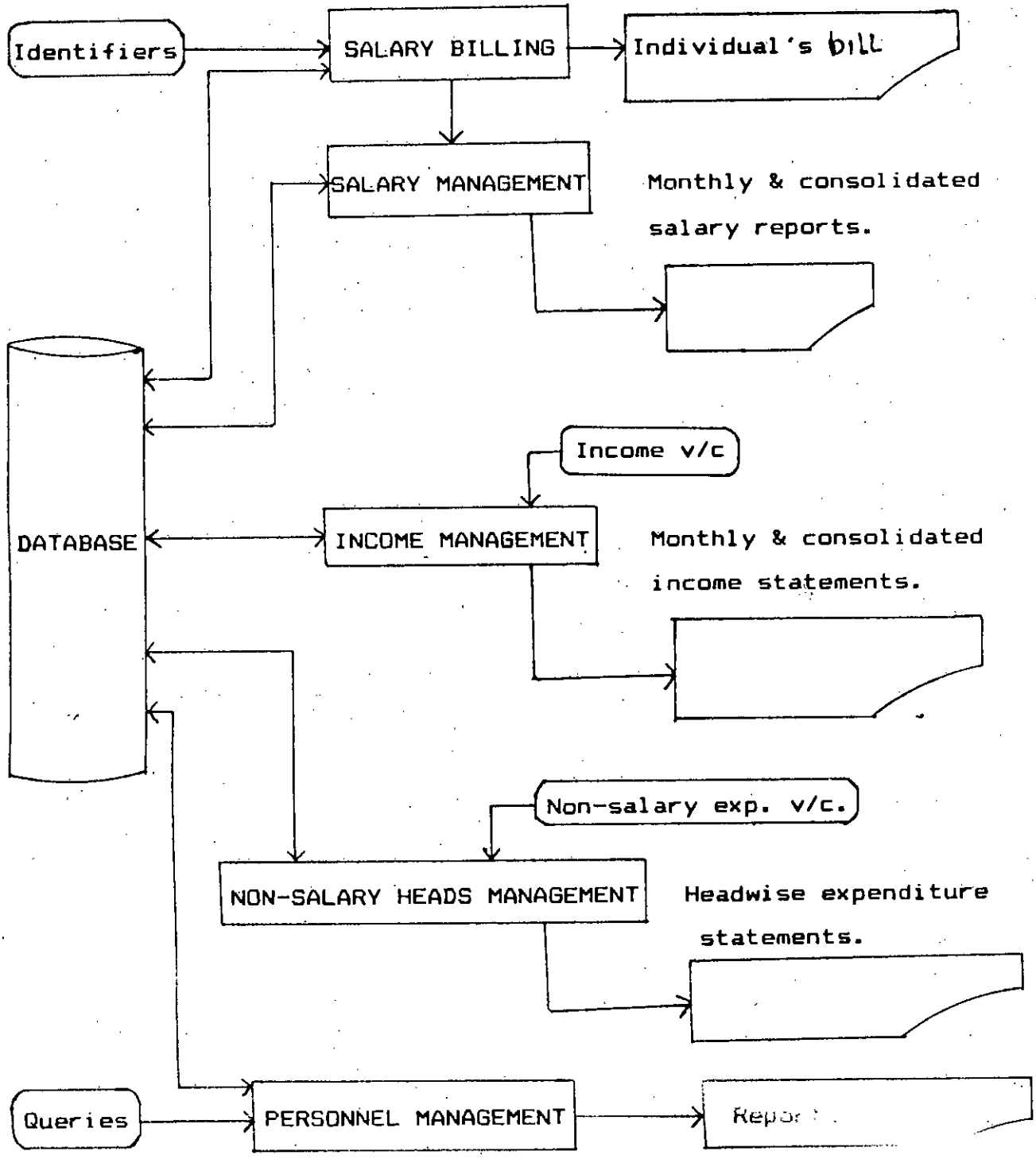


Fig. 4.3 A Logical Model for the proposed I M S .

4.4 DATA DICTIONARY

Formal documentation must be prepared as an educational tool whenever a new individual is added to the effort. One such documentation is the previously presented entity-relation model. A second tool is the data dictionary. Definations of entities identified during the development of conceptual schema are placed in the data dictionary. The data is updated regularly during the life of the database. A data dictionary is some what like an English language dictionary. Within a database envirnment, it is used as an automated means to define entities, attributes and relationships during conceptual design. As the design progresses, the data dictionary is used to define records, the location of database(s) in which record occurs, the means by which the value of the attribute is obtained and the programs which access the record.

The data dictionary performs several additional functions. First, it provides an automated means to store, update and retrieve information about the data used by the University. It also provides an automated means of examining the databases affected when new applications are installed which cause new attributes to be added to the existing databases. It also identifies which programs need to be modified, such as the case of the change from five digit numerical attribute to a nine digit attribute.

Second, if standards for its use are properly controlled by the database administrator, it provides an automated means of documentation. However, if it is not properly managed, it does not provide this function. Worse it can lead to data redundancy and inefficient database design.

Third, some data dictionaries generate the schema and subschema definations for the DBMS, reducing the work of the database administrator in installing a new system.

Fourth, database dictionary permits modifications which aid in the operation of DBMS.

Finally, datbase dictionary can be accessed directly by the user to determine the names used to identify data. Use of the data definition in this manner allows the user to write some of his own programs.

Database dictionary includes (i) catalogs of attributes (ii) catalogs of data (iii) catalogs of database files (iv) catalogs of programs etc. Database dictionary (model) is presented in appendix A 2.

4.5 DIFFERENT FILES IN THE PHYSICAL SYSTEM

In the physical system, all the data are inthe form of different entities and are manipulated in different files. There are several types of data files. Each type of data file has its own extension name in dBASE environment. These are tabulated in table 3.

TABLE 3 Types of data files

Sl.No.	File type	Extension name
1.	Database file	.dbf
2.	Database text file	.dbt
3.	Database index file	.ndx
4.	Format file	.fmt
5.	Screen file	.scr
6.	Query file	.qry
7.	Report file	.frm
8.	Program file	.prg

Functional relationship between these files is presented in next article.

4.6 RELATION BETWEEN FILES

Actually the raw data stored are in database files. The other files are associated files. When any data are entered in the general data entry form or specific data entry form using the format of screen files, the data goes to database files. When any report file is printed, data comes from database files ; the form files simply stores the information regarding printer and manipulate data to be printed or displayed. Similarly the indexed files actually brings records from database files through its pointer table and shows the data in sorted form. The following figure 4.4 shows the relationship between different files in the proposed physical Information Management System.

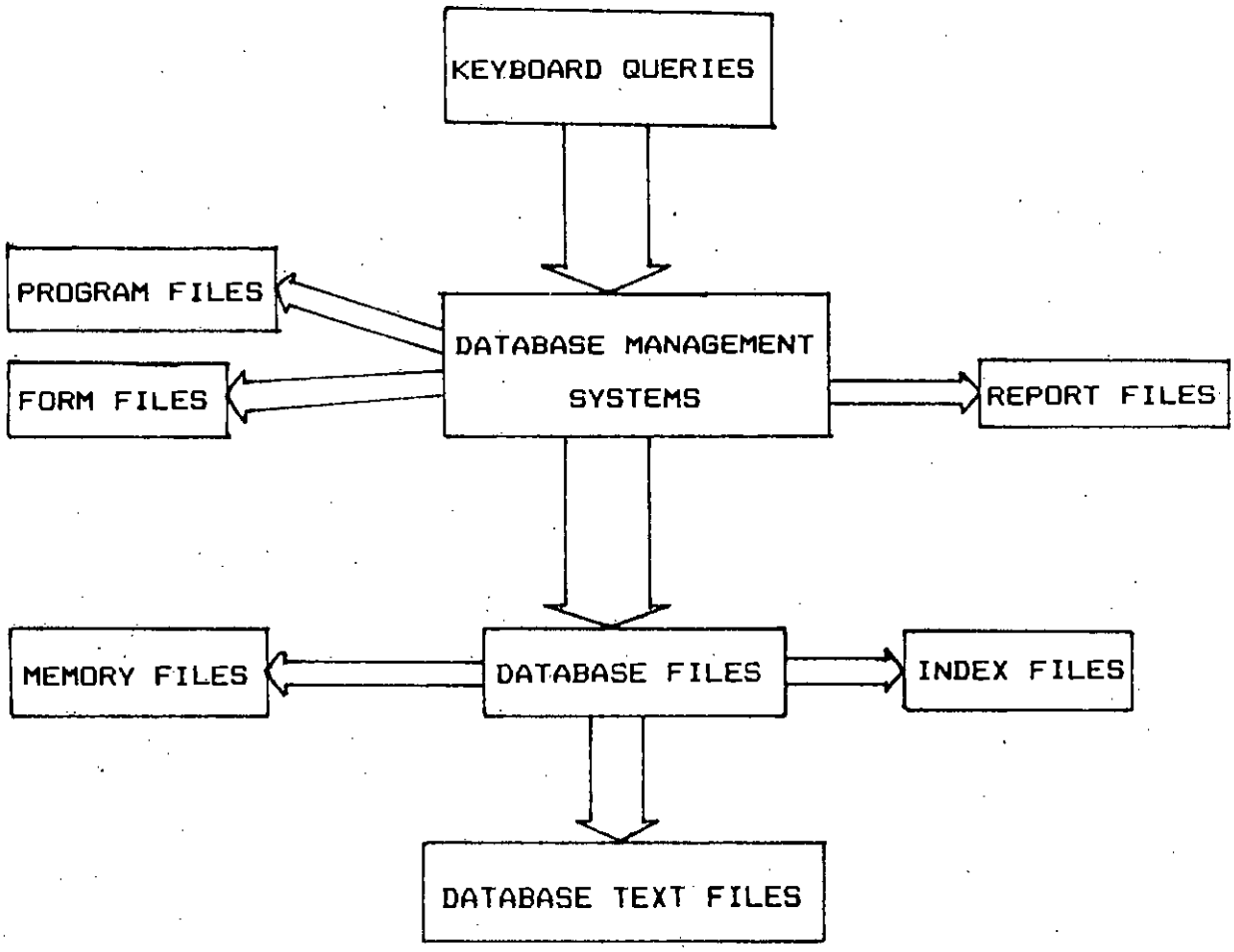


Fig. 4.4 Relationship between Files.

4.7 OUTPUT DESIGN

Outputs of the information management system are classified into:

a. Outputs of the personnel management system.

b. Outputs of the salary management system

c. Outputs of the non-salary heads management

d. Outputs of the income management

Structure of the various outputs are presented below;

A. OUTPUTS OF PERSONNEL MANAGEMENT SYSTEM

A1. List of Professors / Associate Professors / Assistant Professors / Lecturers / all teachers in ... department who have joined after [a given date] / [starting date]:

BANGLADESH UNIVERSITY OF ENGG. AND TECHNOLOGY, DHAKA

List of teachers in the department of .....

ID-NO.	NAME	POST
430001	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	PPPPPPPPPPPPP
430002	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	PPPPPPPPPPPPP
430003	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	PPPPPPPPPPPPP
430004	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	PPPPPPPPPPPPP
430005	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	PPPPPPPPPPPPP
430006	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	PPPPPPPPPPPPP
.....		
.....		

Total number of ..... in the department who have joined after the date : ..... = ...

A2. List of 3rd class / 3rd[tech.] class / 4th class / all employees in department/ office of ..... who have joined after the date ..... = ...

BANGLADESH UNIVERSITY OF ENGG. AND TECHNOLOGY, DHAKA

List of employee in the department/office of .....

ID-NO.	NAME	DATE OF JOINING	DISTRICT
430001	XXXXXXXXXXXXXXXXXXXXXXXXX	DDDDDDDD	KKKKKK
430002	XXXXXXXXXXXXXXXXXXXXXXXXX	DDDDDDDD	KKKKKK
430003	XXXXXXXXXXXXXXXXXXXXXXXXX	DDDDDDDD	KKKKKK
430004	XXXXXXXXXXXXXXXXXXXXXXXXX	DDDDDDDD	KKKKKK
.....			

A8. Academic records of .....[id-no.]

Academic records of .....[Name]

ID-NO. .....

Examination	Div./Class/GPA	Year of passing
S S C	1	1970
H S C	1	1972
Graduation	1	1976
Master's	3.5 Out of 4	1978
Ph.D.		1983

A3. List of teachers holding Ph.D. in department of .....

BANGLADESH UNIVERSITY OF ENGG. AND TECHNOLOGY, DHAKA

List of teachers holding Ph.D. in the department of .....

ID-NO.	NAME	POST
430001	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	PPPPPPPPPPPP
430002	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	PPPPPPPPPPPP
.....		
.....		

A4. List of employee on leave

BANGLADESH UNIVERSITY OF ENGG. AND TECHNOLOGY, DHAKA

List of employee on leave

ID-NO.	NAME	STARTING DATE	EXPIRED DATE
430001	XXXXXXXXXXXXXXXXXXXXXXXX	DDDDDD	DDDDDDD
430002	XXXXXXXXXXXXXXXXXXXXXXXX	DDDDDD	DDDDDDD
.....			
.....			

A5. List of adhoc employee

BANGLADESH UNIVERSITY OF ENGG. AND TECHNOLOGY, DHAKA

List of adhoc employee

ID-NO.	NAME	DATE OF JOINING
430001	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	PPPPPPPPPPPP
430002	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	PPPPPPPPPPPP
.....		

A6. List of employee in scale ....

BANGLADESH UNIVERSITY OF ENGG. AND TECHNOLOGY, DHAKA

List of employee in scale of .....

ID-NO.	NAME	POST
430001	XXXXXXXXXXXXXXXXXXXXXXXXXXX	PPPPPPPPPPPP
430002	XXXXXXXXXXXXXXXXXXXXXXXXXXX	PPPPPPPPPPPP
.....		
.....		

A8. Address of .....

Name: xxxxxxxxx

Father's name: yyyyyyyyy

Present address: kkkkkkk

Permanent address: zzzzz

A9. List of female or non-muslim employee

BANGLADESH UNIVERSITY OF ENGG. AND TECHNOLOGY, DHAKA

List of female/non-muslim employee

ID-NO.	NAME	DISTRICT
430001	XXXXXXXXXXXXXXXXXXXXXXXXXXX	PPPPPPPPPPPP
430002	XXXXXXXXXXXXXXXXXXXXXXXXXXX	PPPPPPPPPPPP
.....		
.....		

B. OUTPUTS OF SALARY MANAGEMENT SYSTEM

B1. Output of salary billing:

Monthly salary bill of ..... ID-NO. .....

For the month: .../93

Basic	=	D A	=
Session allowance	=	P P	=
P A	=	House rent all.	=
Medical allowance	=	.....	=
Electric bill	=	Gas bill	=
Medicine bill	=	Telephone bill	=
GPF	=	GPA	=
BFD	=	GID	=
House rent	=	.....	=

Gross amount = Net deduction =

Net amount =

B2. Monthly deduction and contribution schedule

MONTHLY DEDUCTION AND CONTRIBUTION SCHEDULE

Dept./ Office code:..... For the month: .....

ID-NO	BASIC	GPF	GPA	TOTAL	PENSION	HBL	BFD	CLUB	ASSO
.....	.....	....	....	....	.....	....	....	....	....
.....	.....	....	....	....	.....	....	....	....	....
.....	.....	....	....	....	.....	....	....	....	....

B3. Consolidated salary report

CONSOLIDATED SALARY REPORT FOR THE MONTH : .....YEAR .....

HEAD	HEAD-NAME	BUDGET	PAID IN THE MONTH	PAID UPTO THE MONTH
...	.....	.....	.....	.....
...	.....	.....	.....	.....
...	.....	.....	.....	.....

B4. Monthly adjustment statement

MONTHLY ADJUSTMENT STATEMENT FOR THE MONTH: ..... YEAR: .....

HEAD	HEAD-NAME	REFUND AMOUNT
.....	.....	.....
.....	.....	.....
.....	.....	.....

C. OUTPUTS OF NON-SALARY HEADS MANAGEMENT

C1. Head-wise Expenditure Statement

HEAD-WISE EXPENDITURE STATEMENT FOR THE MONTH: ..... YEAR: .....

HEAD	HEAD-NAME	AMOUNT
.....	.....	.....
.....	.....	.....
.....	.....	.....

C2. Departmental Expenditure Statement

DEPARTMENTAL EXPENDITURE STATEMENT FOR THE MONTH: .... YEAR: ...

HEAD	HEAD-NAME	AMOUNT
.....	.....	.....
.....	.....	.....
.....	.....	.....

D. OUTPUTS OF INCOME MANAGEMENT SYSTEM

D1. Daily income report

DAILY INCOME REPORT FOR THE DATE: .....

HEAD	HEAD-NAME	AMOUNT
.....	.....	.....
.....	.....	.....
.....	.....	.....

D2. Periodical income report

PERIODICAL INCOME REPORT FOR THE PERIOD ..... TO .....

HEAD	HEAD-NAME	AMOUNT
.....	.....	.....
.....	.....	.....
.....	.....	.....

4.8 INPUT DESIGN

Inputs of the proposed I M S can be catagorized into:

1. **Permanent inputs to database files:** These are the permanent values of some attributes in database files e.g. name, date of birth, father's name, permanent address etc. These inputs are entered during the installation of the system.

2. **Updateable inputs to database files:** These are the values of some attributes in database files that should be periodically updated on running system e.g. post, basic pay, scale etc.

3. **Inputs to hidden tables:** As the system is designed to be highly flexible and hence intelligent, there are some hidden tables to keep records of all possible government policies so that the system can response with the change of government policies regarding some attributes. As for example, whenever government announces an additional percentage of D.A. or any other allowance under certain condition, the present system will run accurately after a minor adjustment of the related hidden file(s). These adjustment should be made by the 'System Manager' of the running system.

4. **Regular inputs to various system modules:** There will be some regular inputs to various system modules of the running system. As for example, income vouchers, non-salary expenditure vouchers, keyboard queries.

All the attributes of the present system are catagorized on the above view and placed in the data dictionary presented in appendix .... So the data dictionary is also the input design of the physical system.

4.9 DATABASE DESIGN

To implement the basic system modules, an efficient and optimised database is designed to store data elements or raw data so that multiple applications can be done without any data redundancy. According to the evaluation of data model discussed in chapter two, it is intended to design a relational database following the E R modeling technique. As a first step, an extended E R model for the proposed information system is developed and presented in figure 4.5 .

The second step is to transform the extended E R model into relational model. According to the rules described in article 2.11 of chapter two, the following candidate relations are obtained.

TABLE 4 List of candidate relations

1. UNIVERSITY (University-Name, Location, VC-Name)
2. DEPARTMENT (Dept-Code, Dept-Name, Head-Emp.Id, ...)
3. ORGANIZATION (Org-Code, Org-Name, Org-Add, ...)
4. PROJECT (Proj-Id, Proj-Name, P-Location, ...)
5. SKILL (Emp-Id, Skill-Code)
6. EMPLOYEE (Emp-Id, Common attributes)
7. TEACHER (Emp-Id, Specific attributes)
8. OFFICER (Emp-Id, Specific attributes)
9. 3RD CLASS (Emp-Id, Specific attributes)
10. 3RD[TECH] (Emp-Id, Specific attributes)
11. 4TH CLASS (Emp-Id, Specific attributes)
12. ASSOCIATION (Asso-Code, Asso-Name, A-Location)
13. BELONGS-TO (Asso-Code, Emp-Id)
14. USE-SKILL (Proj-Id, Emp-Id, Skill-Code)
15. INCOME (In-Head, In-amt, In-dt, ...)
16. EXPENDITURE (Ex-Head, Ex-amt, Ex-dt, ...)
17. HEADED-BY (Head_Emp-Id, Dept-Code,..)

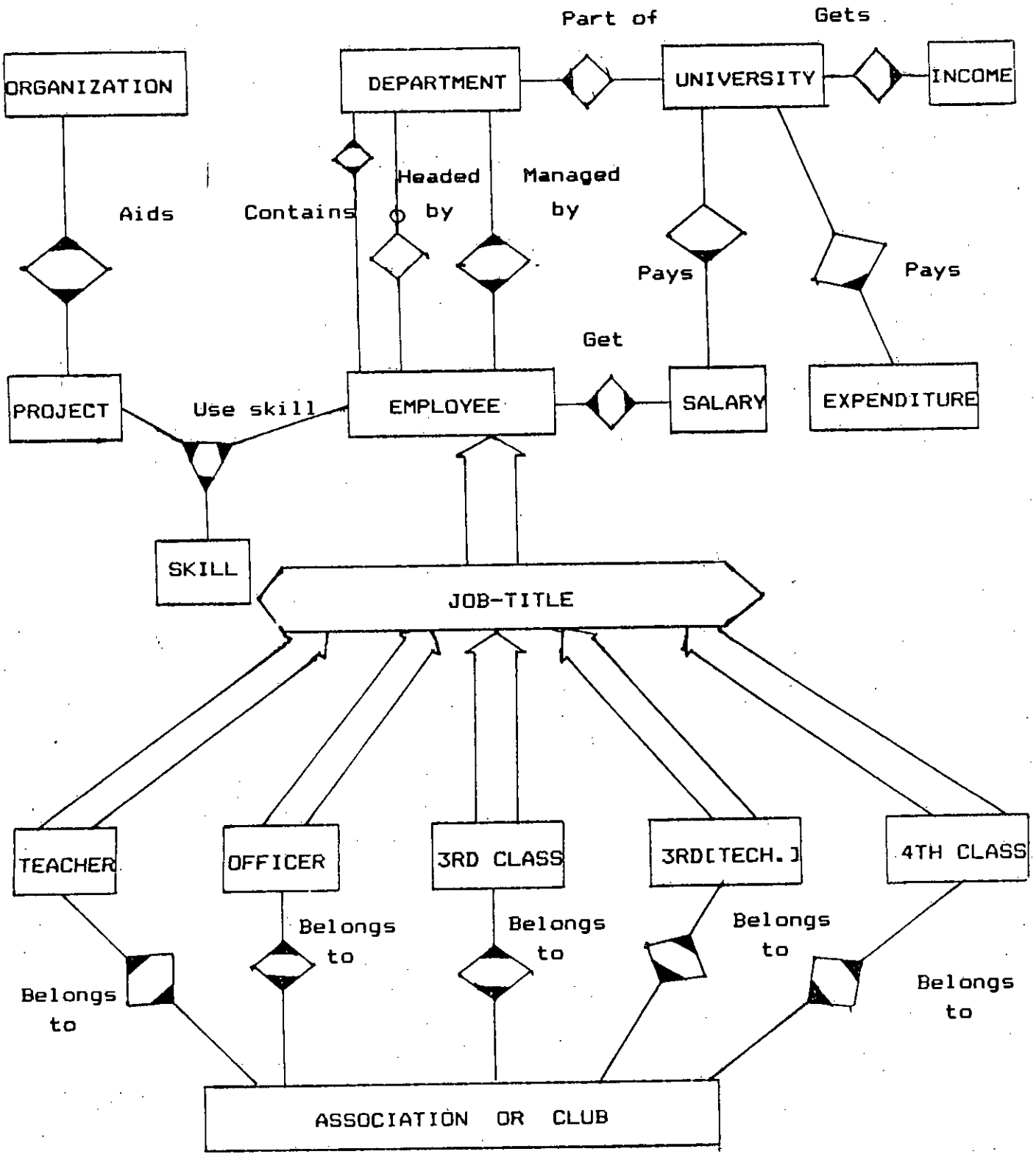


Figure 4.5 E E R model for the proposed I M S [Dept. includes all teaching depts, offices, research institutes etc. ; BUET has 46 such departments].

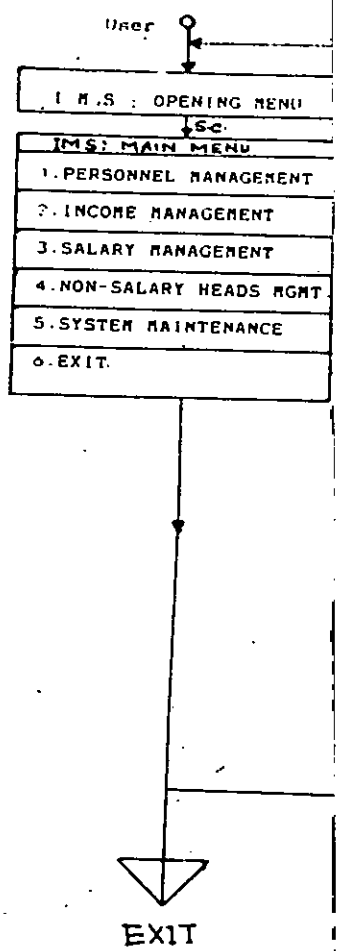
As a third step, normalization of candidate relations are accomplished by analyzing the functional dependencies [F Ds] and multi-valued dependencies [MVDs] associated with those relations. Each candidate relation is examined to determine what dependencies exists among primary key, foreign key and non-key attributes. These dependencies determine the current degree of normalization. Well-known techniques are applied to each relation to increase the degree of normalization. Detail structures of each normalized relation is presented in database dictionary.

4.10 DESIGN OF MENU STRUCTURE

Menus in the physical system are organized as in figure 4.6 . An opening menu with security control is designed at the starting of the system. In this menu, the user is requested to enter his valid id-no. and his name-code so that unauthorized users cannot enter into the system. For an authorized user, the opening menu led him to main menu where six options are available; any one can be selected. Menus are very simple but user-friendly. Technical words are not used in menus, so that non-technical users can easily understand the operation and usefulness of the system. Detail operation of the system is discussed in the following article.

4.11 SYSTEM OPERATION AND MAINTENANCE

To make the system user-friendly, menu structure is designed simply but efficiently so that the user has to pass minimum levels to do his job. At main menu level, six options are available, any one can be selected. The user can also exit from the system. Detail flow diagram [F.D.] for operations on each system module are presented below:


User
I M.S : OPENING MENU
Sc.
IMS: MAIN MENU
1.PERSONNEL MANAGEMENT
2.INCOME MANAGEMENT
3.SALARY MANAGEMENT
4.NON-SALARY HEADS MGMT
5.SYSTEM MAINTENANCE
6.EXIT.
EXIT

(a) P.D. for operation on personnel management

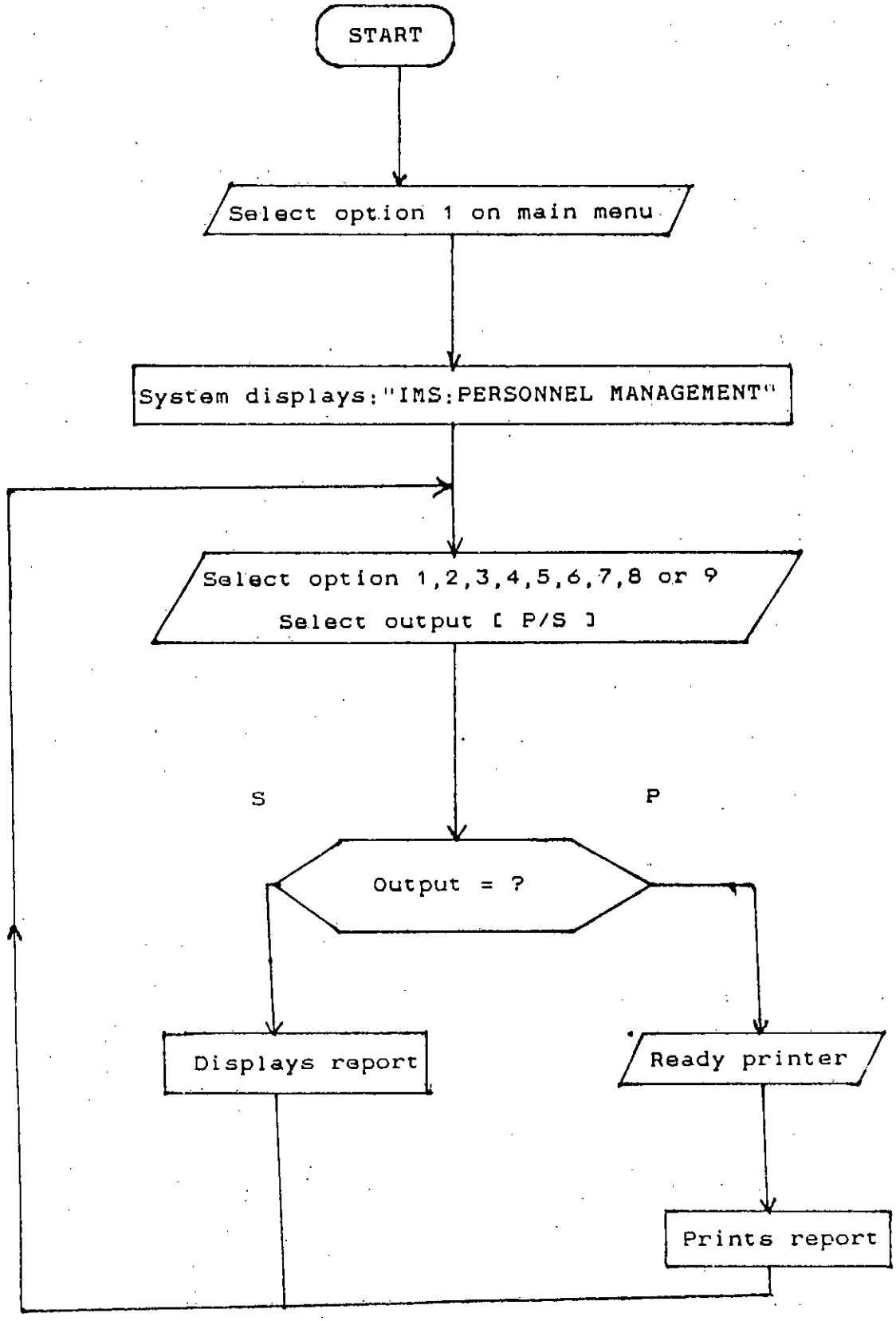


Fig. 4.7 Flow diagram for operation on Personnel Management

(b) F.D. for operation on income management

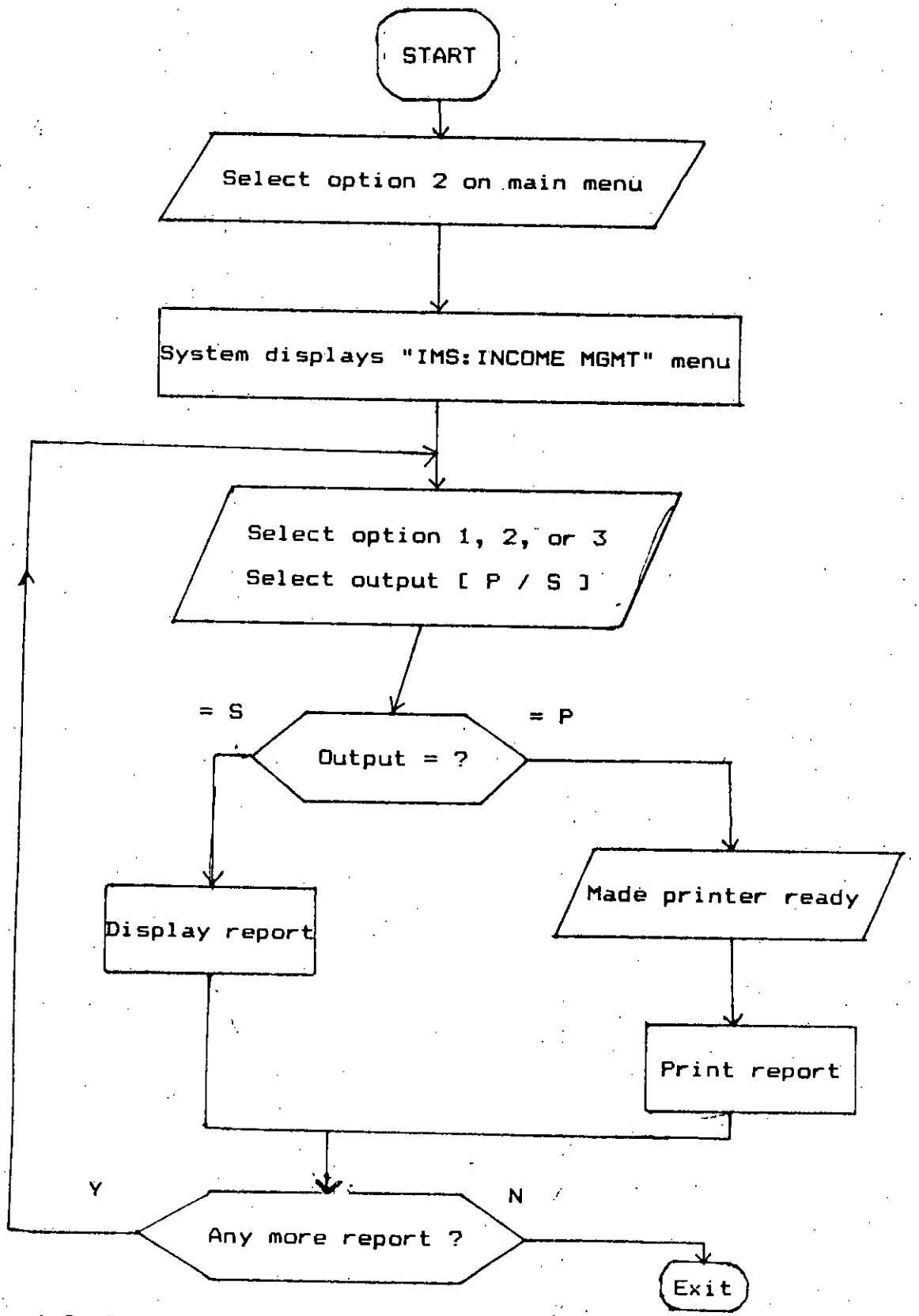


Fig. 4.8 Flow diagram for operation on Income Management

(c) Flow diagram for operation on System Maintenance

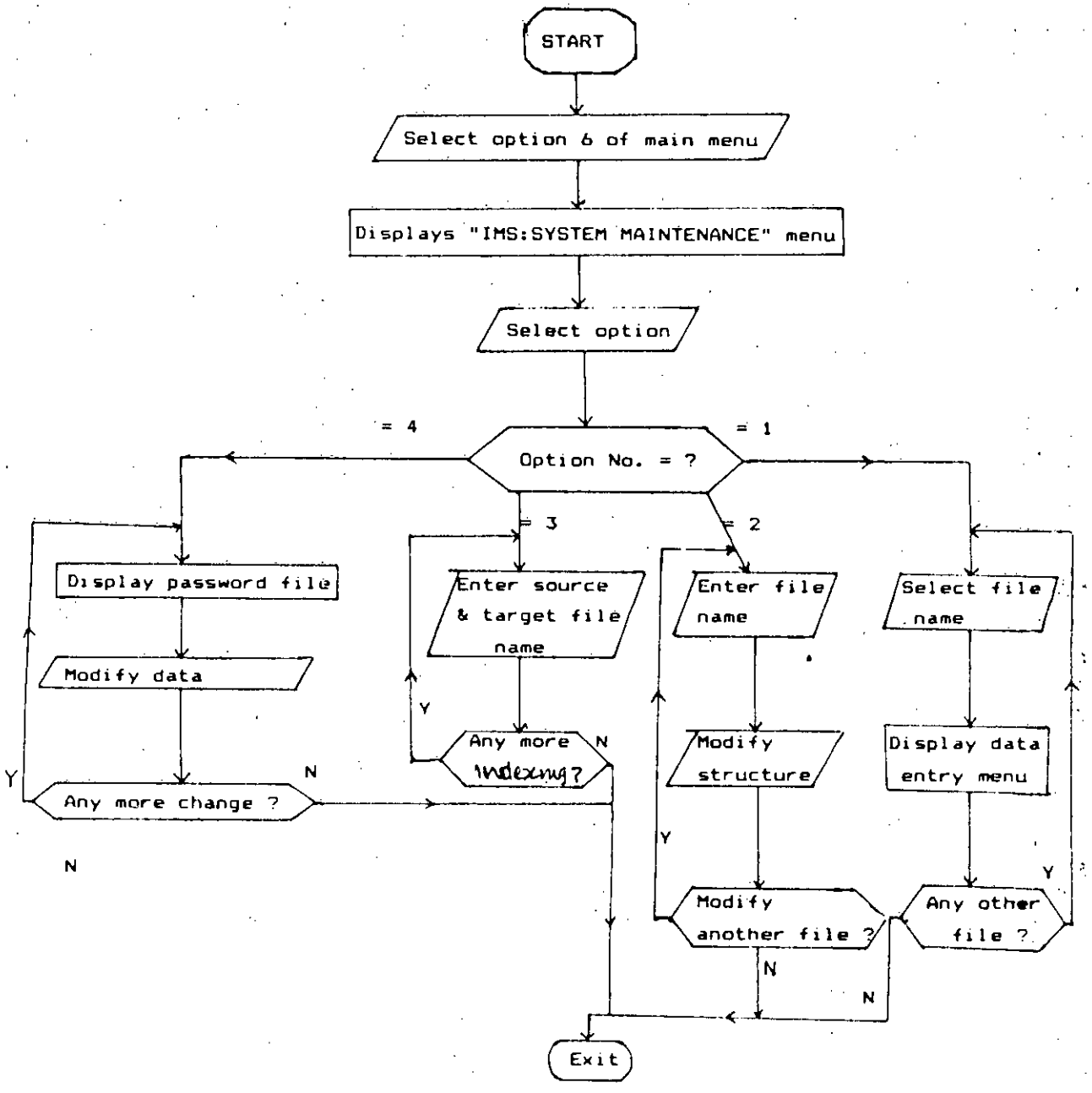


Fig. 4.9 Flow diagram for operation on System Maintenance

(d) F.D. for operation on salary management

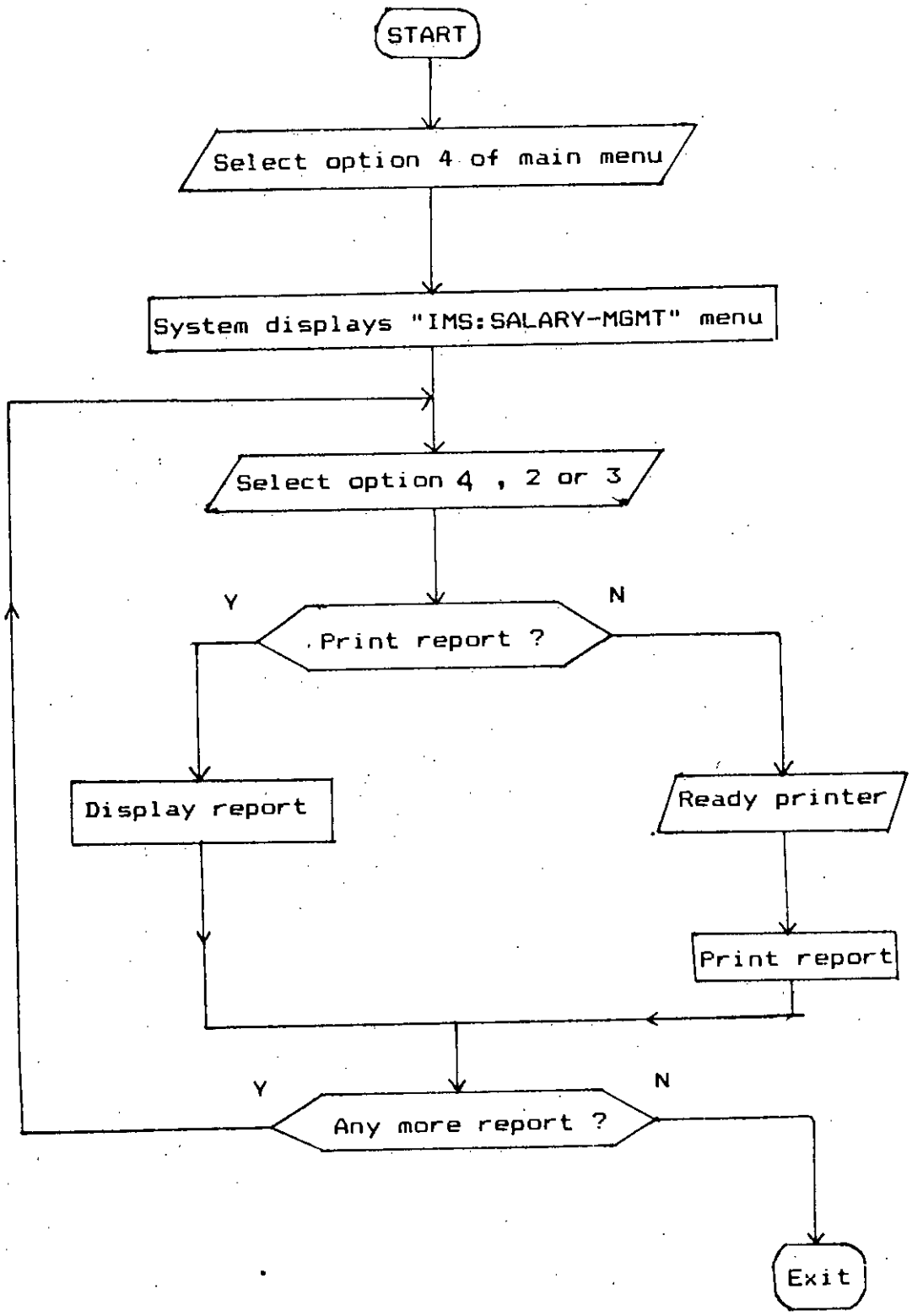


Fig. 4.10a Flow diagram for operation on Salary Management

(e) F.D. for operation on salary billing

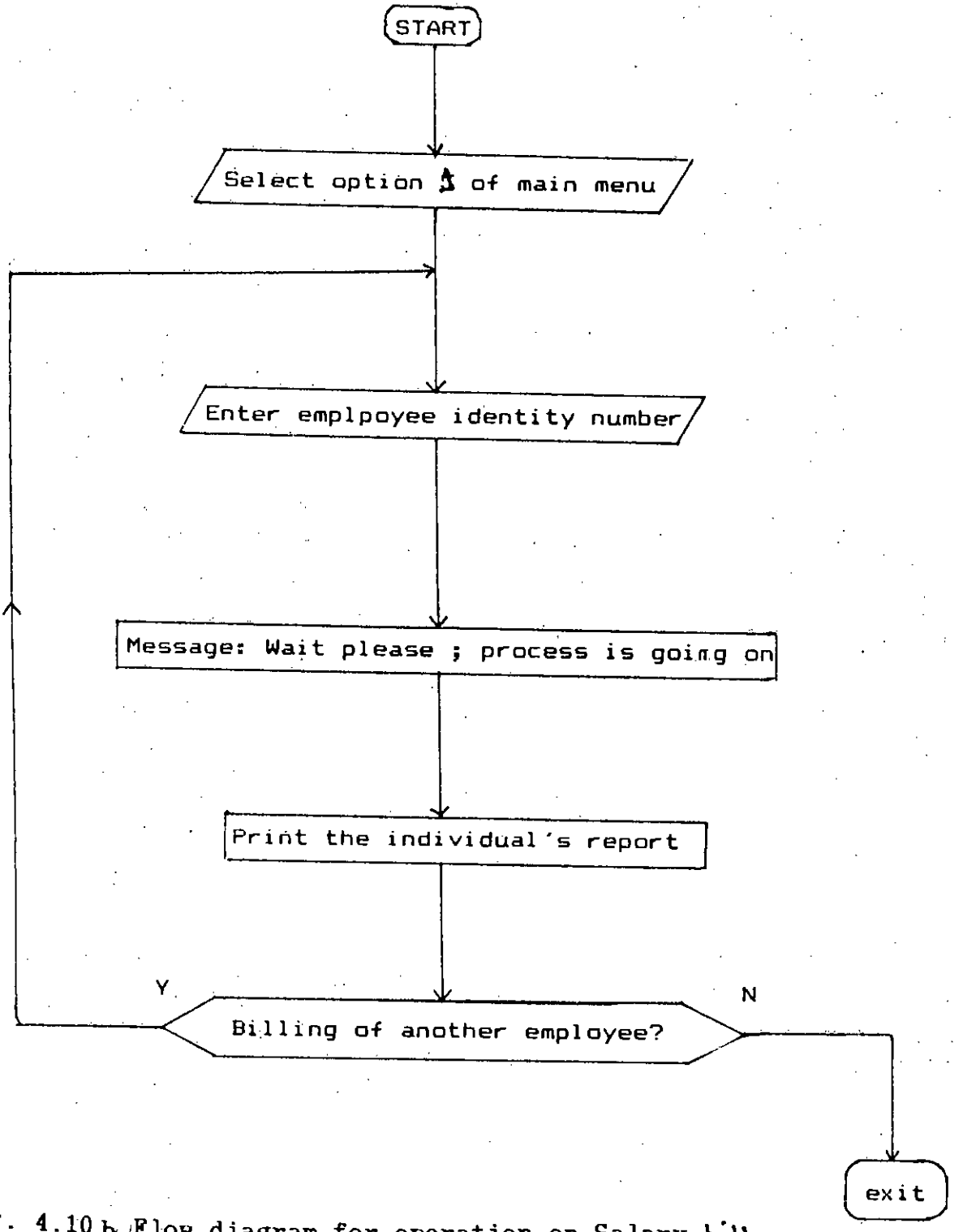


Fig. 4.10 b Flow diagram for operation on Salary billing

4.12 PROCESS DESIGN

Process design is the specification of various processes to be performed by the computing system to implement a specific functional module. In each process, there are some inputs and some outputs. So process design for each functional module is presented as I P O [input,process,output] chart in the follwing sections:

TABLE 5 I P O chart for Personnel Manangement

SYSTEM: I M S		DATE: 15-3-92
MODULE: PERSONNEL MANAGEMENT		AUTHOR:
INPUT	PROCESS	OUTPUT
1.Keyboard queries 2.Data files	1.Accept keyboard queries and note options. 2.Open data file(s) to input data 3.Process on data file(s) according to options and store in temp. file 4.Transfer processed information into report form with heading, date etc. 5.Process to display or print the report. 6.Process next operation.	1.User-selected report on display or printed form

TABLE 6 I P O chart for Income Management

SYSTEM: I M S		DATE: 15-3-92
MODULE: INCOME MANAGEMENT		AUTHOR:
INPUT	**PROCESS**	**OUTPUT**
1.Keyboard queries 2.Data files	1.Accept keyboard queries and note type of report. 2.Open data file(s) to input data 3.Open data file(s) to store data 4.Process input data file(s) according to selected option. 5.Store processed data in output data file(s). 6.Print or display the report. 7.Exit or return to menu.	1.Daily income report. 2.Monthy income report.

TABLE 7 I P O chart for Salary Billing

SYSTEM: I M S		DATE: 15-3-92
MODULE: SALARY BILLING		AUTHOR:
INPUT	**PROCESS**	**OUTPUT**
1.Keyboard queries (Emp-id, month) 2.Data files 3.Policy tables	1.Accept keyboard queries 2.Open data file(s) and store necessary fields to mem-variables 3.Open policy tables 4.Calculate common attributes and store in mem-variables 5.Calculate conditional attribute and store in mem-variables 6.Open output data file and load mem-variables. 7.Print employee's report 8.Process next operation.	1.Salary bills

TABLE 8 I P O chart for Salary Management

SYSTEM: I M S		DATE: 16-3-92
MODULE: SALARY MANAGEMENT		AUTHOR:
INPUT	PROCESS	OUTPUT
1.Data files 2.Keyboard queries	1.Accept keyboard queries 2.Open data files 3.Set printer or display 4.Check validity of input from keyboard. 5.Process data 6.Prepare report form 7.Input data in form 8.Display or print report	1. Monthly deduction & contribution schedule. 2. Consolidated salary report.

TABLE 9 I P O chart for Non-salary heads Management

SYSTEM: I M S		DATE: 16-3-92
MODULE: NON-SALARY HEADS MANAGEMENT		AUTHOR:
INPUT	PROCESS	OUTPUT
1.Keyboard queries 2.Data files	1.Accept keyboard queries 2.Store data from data files 3.Check input from keyboard 4.Prepare report form 5.Process data, store data 6.Display or print report 7.Return	1.Head-wise expenditure statement. 2.Departmental expenditure report

4.13 PROGRAM DESIGN

According to structured approach of program development the following concepts are adopted so that the whole system become a best moduler in form.

NAMING PROGRAM: A naming principle is adopted that will enable to back up the entire system at least the program files with a single command. The easiest way to do this is to begin all programs with the same two or three letters. As for example, in the present system all programs will begin with IM. In this way all the program files can be backed up by a single copy command. As the system design is moduler or structured, the next two letters are used to indicate program module. As for example, IMMM.PRG is the name for main menu program.

PARTS OF PROGRAM: Each program consists of four parts: the preamble, the set up area, the body and the closing section.

The preamble contains the following elements:

1. A comment that describe the system
2. Any copyright or author's notice that is appropriate
3. The name of the program
4. The date of last revision.

As for example the preamble for the main menu program is as follows:

```
*-------------------------------------------------------------
* SYSTEM:I M S          ;  Copyright: Author
* Name: IMMM.PRG        ;  Date     : 15-3-92
*-------------------------------------------------------------
```

The set up area consists of commands that set up the working environment for the program. Here 'SET' commands are issued, files are opened and variables are initiated.

The body of the program consists of the actual commands that do the work of the program. These commands inputs data from database and or screen, process data, create report, edit data file(s) and do whatever else we want the program to accomplish.

The closing section is the final part of the program. This part is used to close files, do any necessary printer ejects and to update global variables as necessary. As a good practice, a footer is added to mark the last line of the program.

4.14 PACKAGE DESIGN

The salient features and characteristics of the developed software package are as follows:

a.MENU DRIVEN: The package is completely menu driven. When the user wants to access to the system, he will see a well decorated opening menu in which he has to enter his valid password. As an authorized user he will get accessed to the main menu. In main menu level, he can select any option except "SYSTEM MAINTENANCE" because working in environment of this option is restricted to technical users. That is, this option placed under further security control. Only authorized technical personnels will get access to the "SYSTEM MAINTENANCE" works.

b.USER-FRIENDLY: The package is user-friendly in that sense that it is menu driven; Headline Box to indicate what is going on in the current session, Promptin Box to prompt what next to be done by the user, error message when something done wrong by the user.

c.ACCESS CONTROL:Access to the system and its maintenance are restricted through password and access category.

d.MODULAR PROGRAM DESIGN: Modular program design gives two major advantages:(1) Better understandibilty since it gives a sturctured program which makes the program self-documenting. That is easy for any one to understand and work with. (2) Easy for maintenance, modification and extention of the system.

e.UTILITY OPERATION: Utility operatios are for better utilization of the database. Following operations are included:

-Entry of new password for new user.

-Edit password and access category.

The utility operations has no such common features as data entry, edit etc. since different utilities has different properties.

4.15 SYSTEM SECURITY AND CONTROL

The plan describing the implementation of system security and control of data, software and hardware is included in the design of the system. The major reasons for implementing system security and control are: protection of program and data by multiple copies maintained off-site; physical security of the computer hardware by taking reasonable precuations against theft,

fire, water, and other catastrophes; and, finally, protection against fraud by using audit trails, passwords, data encryption, and supervision. The plan of the security considerations of the developed system is presented bellow:

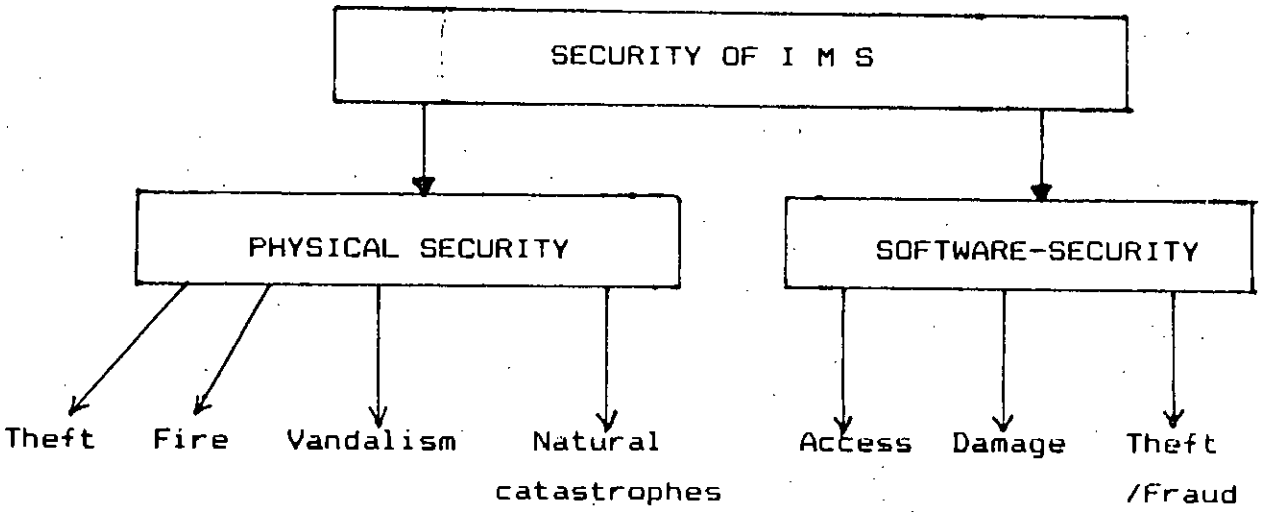


Fig. 4.11 Plan of security considerations

Solutions for some types of security prblems are shown below:

SECURITY AREA	SOLUTIONS	REDUCES EFFECT OF
Computer hardwarwe and installation	Hardware back up	Fire, theft, natural disaster
	Software back up	Accidental erasure
	Data back up	Accidental erasure, vandalism
Data and software	Adit trails	Embezzlement, fraud
	Limit access	Sabotage, vandalism
	Supervise	Fraud, theft, espionage

The process of creating back up copies of database, index and other files is designed into the system so that the user can invoke this process directly, without programmer intervention. Data encryption provides a higher level of security. The data can be coded by using a string entered by the operator, such as a password. So that only proper password and program can be used to decode an encoded field.

CHAPTER FIVE

DEVELOPMENT OF THE PROPOSED I M S

CHAPTER OUTLINE:

Developing the Information Management System is concerned with practical works. A brief description of the various stages of developing work and program flow-charts are presented in this chapter.

5.1 BUILDING OF DEVELOPMENT SCHEDULE

This is the first step of developing a designed information management system. It includes assignment of tasks [which is very very important in commercial projects] and development of program flow diagram. Building of program flow diagram or program structure is made very carefully, following the physical design of the proposed I M S. The program flow diagram shown in figure 5.1 is the total program flow chart for complete physical system.

There are two types of programs; menu programs and processing programs. The opening menu program [om.prg] is designed with two-stage password control in accessing the system. In this stage, the user should enter his valid id-number. For wrong id-number, he will get more chance for retry. After successful entry of id-number, he should also enter his name-code. Three chances are available for wrong entry of name-code. After successful entry, the user will get access to the system that is the opening menu program calls the main menu program [mm.prg]. The main menu program offers six options to the user. Each option represents a functional module of the software system. After selection of any one option, the main menu programs calls the related sub menu program [viz pm.prg, sm.prg, nshm.prg, sysm.prg or im.prg]. The sub-menu program offers a number of basic jobs to be selected by the user. When the user select his job, the sub-menu program calls the related processing program. The processing program then interact with the user for entry of some data. If the user responds properly, the processing program present the output either on display or as printed in paper according to the choice of the user.

5.2 PROGRAMMING

As the system is welldesigned, this step is found less complicated. Menu programs and sub-menu programs are written according to the designed menu structure presented in chapter 4.

Processing program for each basic job defined in design is written according to the I P O chart. Flow-chart for various programs are presented below:

a.Flow chart for opening menu program:

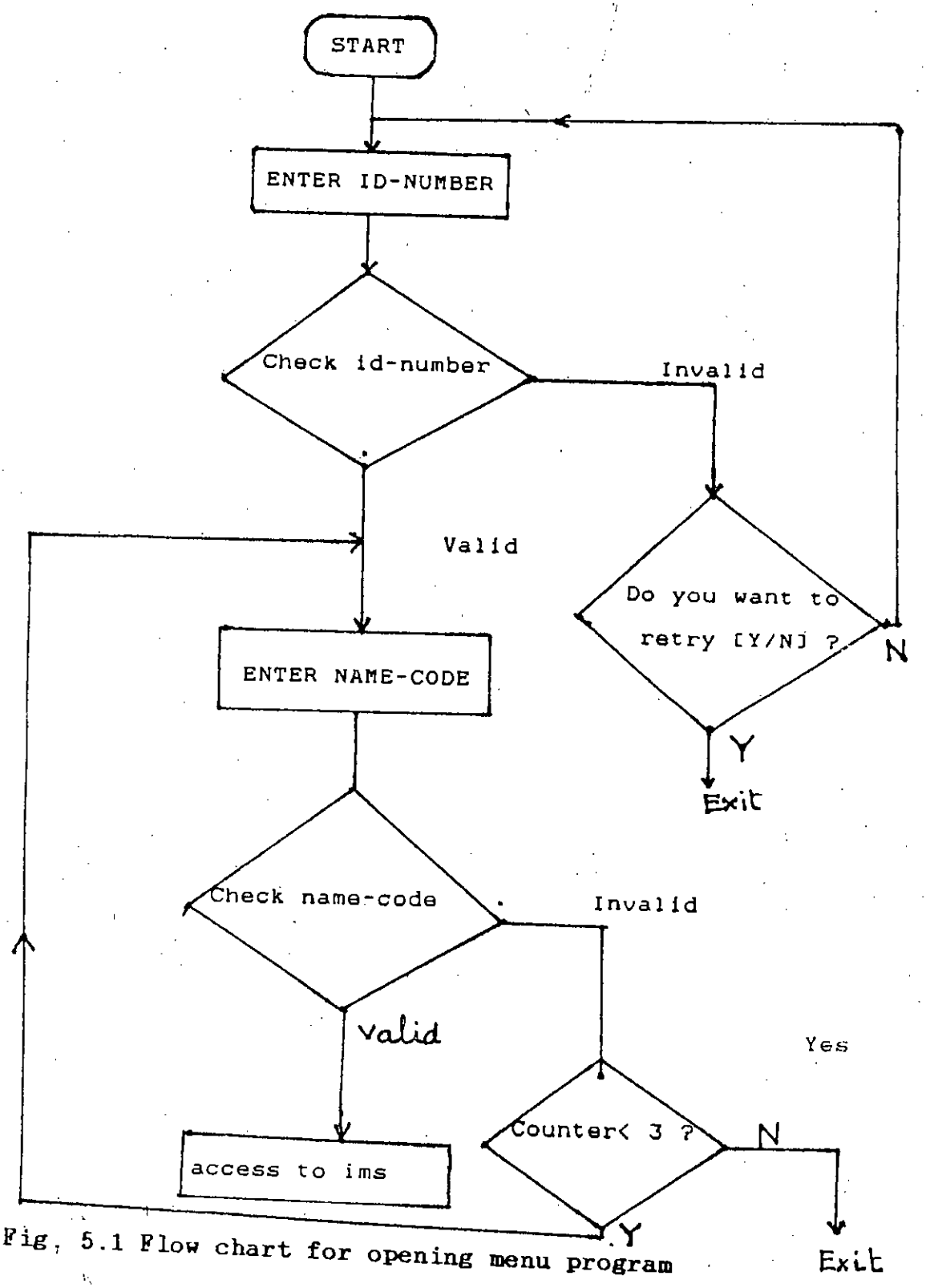


Fig. 5.1 Flow chart for opening menu program

b. Flow chart for main menu program:

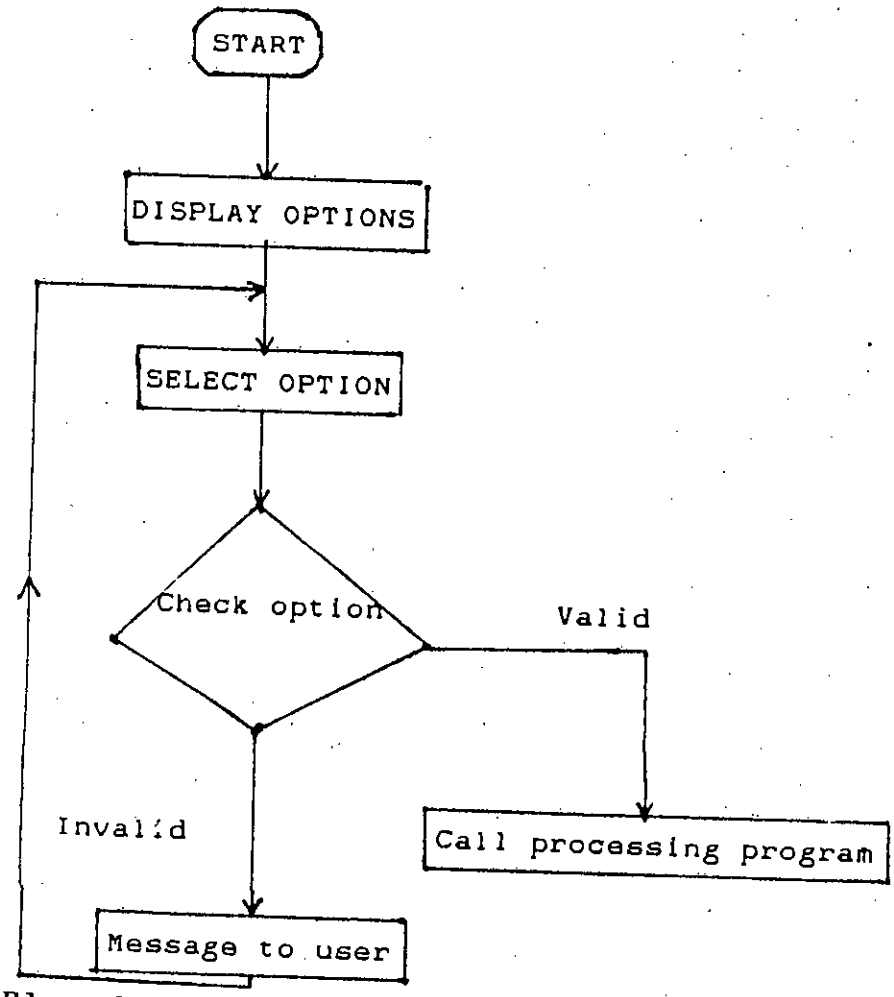


Fig. 5.2 Flow chart for main-menu program

c. Flow chart for sub-menu programs:

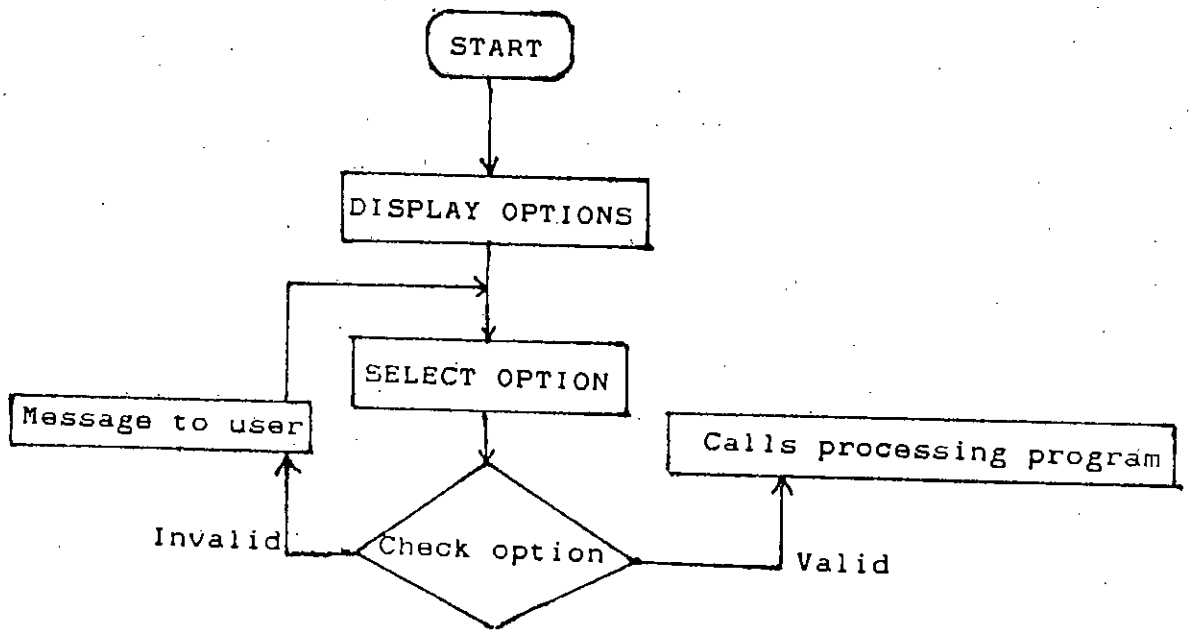


Fig. 5.3 Flow chart for sub-menu programs

d. Flow chart for processing programs of Personnel Management.

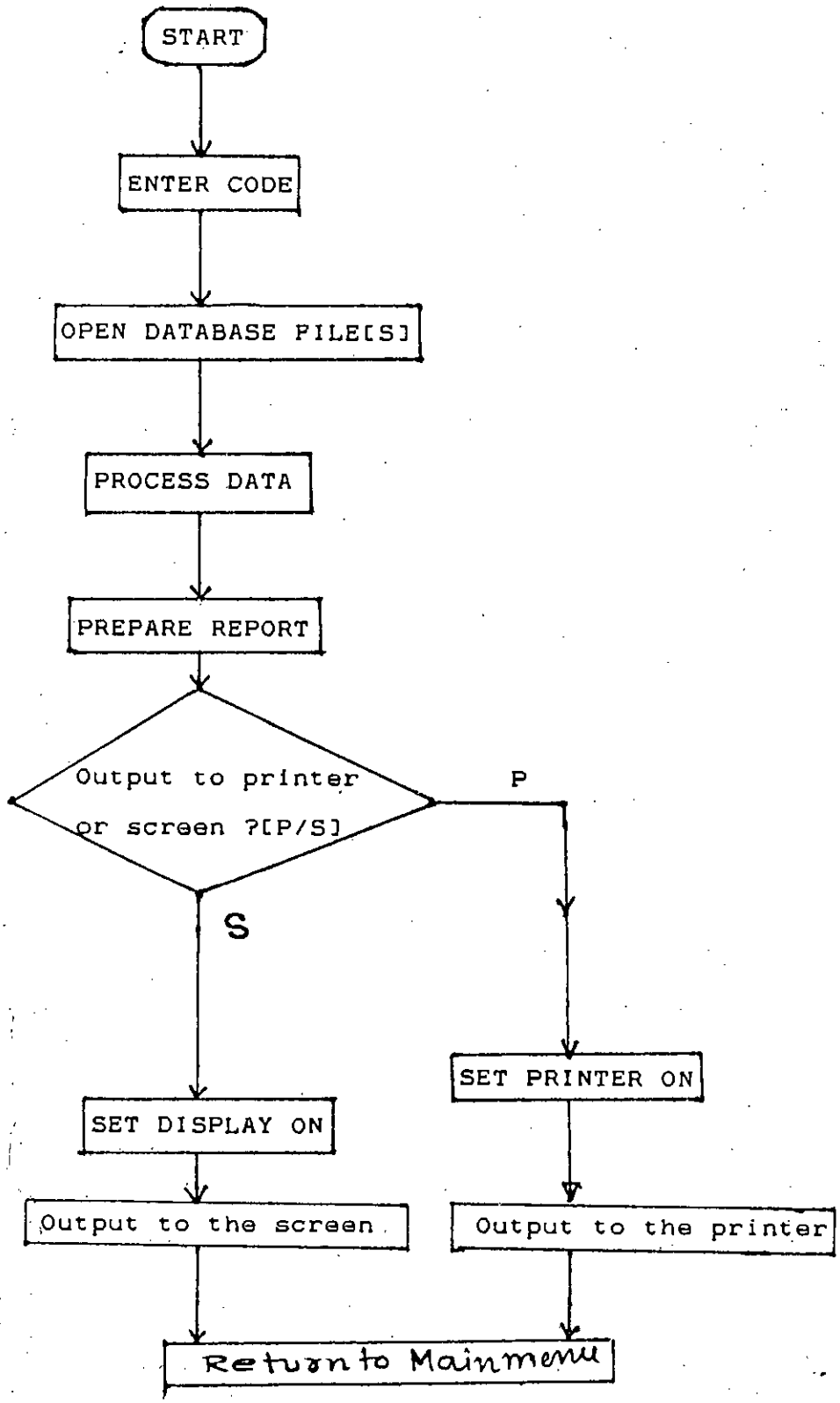


Fig5.4 Flow chart for processing programs of Personnel Management.

e. Flow chart for sbill.prg [Program for salary billing]

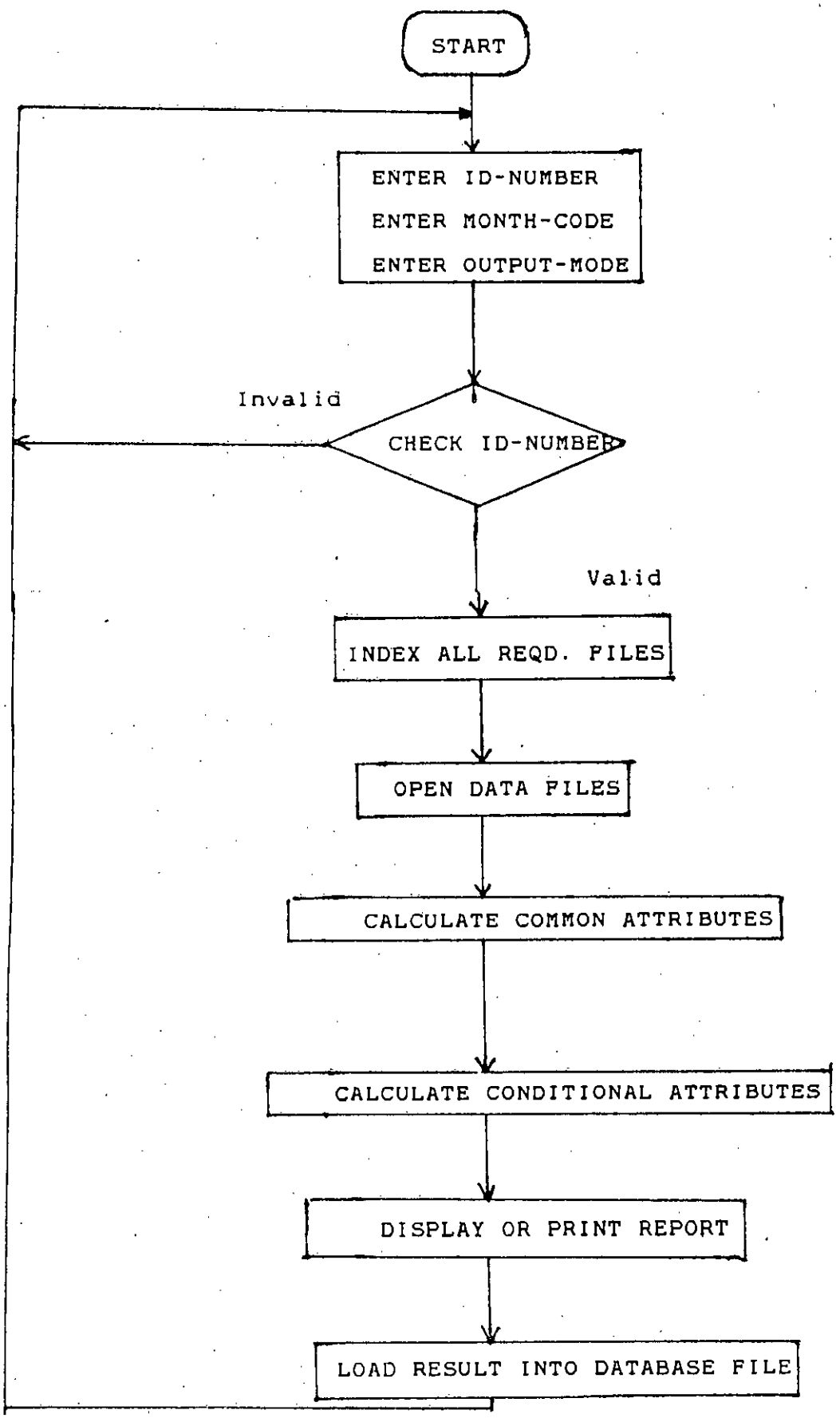


Fig. 5.5 Flow chart for salary billing program

5.2.2 Program listings

There are 29 programs in the physical system. One or more procedures are also involved in most of the programs. Program-name and its function of all programs are presented below:[detail-listing of programs and associated procedures are shown in appendix A-3].

1. OM.PRG ; program for opening menu
2. MM.PRG ; program for main menu
3. PM.PRG ; program for sub-menu - personnel management
4. SM.PRG ; program for sub-menu - salary management
5. NSHM.PRG;program for sub-menu - non-salary heads management
6. SYSM.PRG;program for sub-menu -system maintenance
7. IM.PRG ; program for sub-menu - income management
8. PM1.PRG ; processing program for list of teachers in ... dept.
9. PM2.PRG ; processing program for list employee in scale of ...
10. PM3.PRG ; processing program for list of teachers holing Ph.D.
11. PM4.PRG ;processing program for list of employee on leave
12. PM5.PRG ; processing program for list of adhoc employee
13. PM6.PRG ; processing program for list employee in ... dept.
14. PM7.PRG ; processing program for academic records of .....
15. PM8.PRG ; processing program for address of ....
16. PM9.PRG ; processing program for list of female or non-muslims
17. SBILL.PRG ;processing program for salary billing
18. MDCS.PRG;processing program for monthly deduction and cont. schedule
19. CSR.PRG ; processing program for consolidated salary report
20. DSR.PRG ; processing program for department wise salary report
21. HER.PRG ; processing program for head-wise expenditure statement
22. DER.PRG ; processing program for departmental expenditure statement
23. MAS.PRG ; processing program for monthly adjustment statement
24. DIR.PRG ; processing program for daily income report
25. PIR.PRG ; processing program for periodical income report
26. DE.PRG ; processing program for data entry
27. CDB.PRG ; processing program for change of database structure
28. CSC.PRG ; processing program for change of security code
29. NDX.PRG ; processing program for indexing all database files

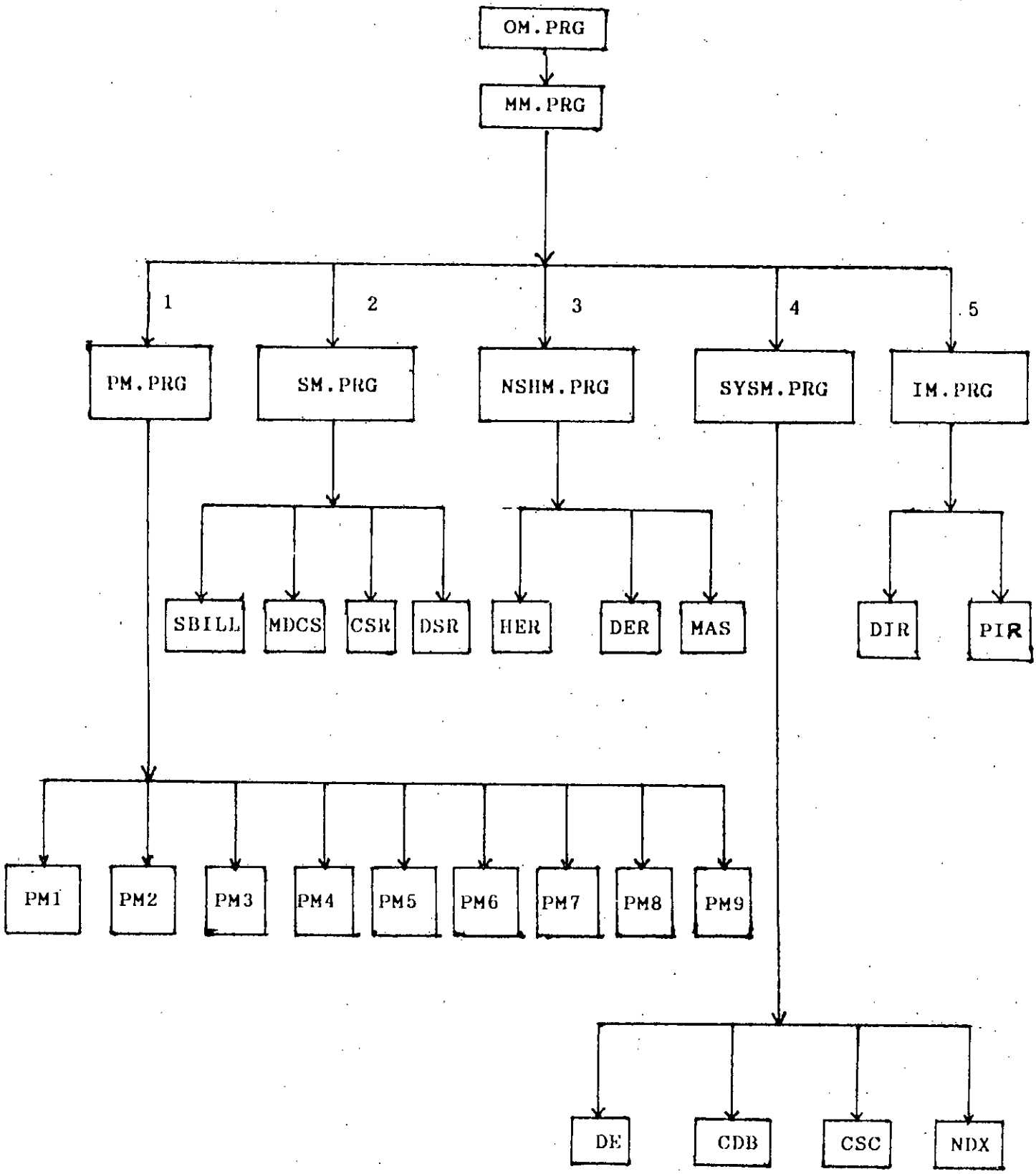


Figure 5.1 Program flow diagram

5.3 TESTING AND LINKING PROGRAMS

In this step, individual programs are coded and tested. This is a tedious but interesting step in developing the information management system. To test a program, related database files are created and loaded with some data. After final debugging, each program is linked to its parent program determined by the program flow diagram [in figure 5.1].

5.4 RUNNING AND TESTING THE SYSTEM

Here the system as a whole is run. But some problems due to erroneous linking and coding are found. So in this stage, modifications and corrections are performed in erroneous system modules. The system is tested for each basic job defined in design. The response of the system for invalid data from keyboard is also studied for each basic job. After a number of iterations, the system is found running successfully.

5.5 SYSTEM AUDIT

In this stage the performance of the developed system is compared with that of the designed system. System performance is tested by implementing all the queries considered in the design. Comparison of the system specifications and performances is found satisfactory and is presented in table18.

TABLE 10 Comparison of performances

Area	Designed System	Developed System
1.User's request	100% fulfilled	100% fulfilled
2.Future extension	Extendible	Extendible
3.System Life time	Ten years	Ten years
4.Utility	For B U E T Dhaka	For B U E T Dhaka
5.Yearly system maintenance cost	Tk=5000.000 only	Tk=5000.00 only
6.Application to other organization	Applicable with some modification.	Applicable with some modification.
7.Unanticipated user 's request.	Can be fulfilled	Can be fulfilled

5.6 CONVERSION, INSTALLATION AND TRAINING

This step is essential for commercial projects but not so for research works. It includes [1] conversion of the source programs into object code to make the system fast and also to maintain privacy and security of programs, [2] installation of the object-coded software system in user's computing system and [3] training of users, so that the user can operate the system successfully and smoothly to get maximum benefit from the system.

CHAPTER SIX

DISCUSSION AND CONCLUSIONS

6.1 UTILITY AND MODIFICATION

The developed information management system can be used for B U E T, Dhaka without any modification. Other universities and academic institutions [like B I Ts] can use this system with least modification. All conventional concepts and theories of developing I M S are discussed and applied. Some new trends are also introduced. So information processing professionals in practical fields and related students will also benefit from this presentation.

6.2 SUGGESTION FOR FUTURE EXTENSION

The design of Information Management System presented in this thesis is outcome of detail study of conventional theories and concepts, various investigations and study of similar works in this field. As a developing country, all government or non-government organizations, bodies, departments, institutions in Bangladesh have been facing serious problems in management of information as discussed in chapter one. This research work was devoted to find a way in solving such a national problem. As the design is more flexible, the present system can be extended to other queries like:

[1] List of Teachers / Officers / Employees / all according to home-district / .....

[2] Allocation of residence among Teachers / Officers / Employees

[3] Service history etc.

As discussed earlier, to improve the performance of such a system, the future direction of works should be involved in finding new trends in analysis and design. To increase life time of such a system, the future designer should keep more allowance in database field structures. To improve overall performance, capacity to handle large volume of textual and graphical data, life time etc., it requires a lot of research and investigations.

6.3 CONCLUSIONS

Finally, it is to be noted that the developed system is more flexible, dynamic and intelligent also. To explore this fact, let us focus on any one module say P M [Personnel Management]. In this system module, the user can generate thousands of academic records, lists, reports etc. The following versatile queries can be formulated in this module.

[1] List of Teachers / Professors / Asso. Professors / Asstt. Professors / Lecturers in the department of ..... joined after a given date or the initial date

[2] List of Asstt. Professors / Asso. Prgoessors / Professors / Teachers holding Ph.D. in the department of .....

[3] List of 3rd class / 3rd(Tech.) / 4th class / all joined after the initial data / agven date.

[4] List of adhoc Teachers / Officers / Employees / all joined after a given date / the initial date.

[5] List of adhoc Teachers / Officers / Employees / all employees on leave joined after a given date / the initial date.

[6] List of employees in the scale .... joined after .....

[7] Academic records of ..... (empid)

[8] Adress of .....(empid)

(9) List of non-muslim / Female , Teachers / Officers /Employees joined after a given date or initial date.

Let us also focus on "SALARY BILLING" ; the module is so dynamic and intelligent that whenever government policies regarding some attributes [like dearness allowance, house rent etc] changes, this module will generate accurate result without any error. In this case, the system manager will bring some modification in related file[s]. So the programs are independent of government policies.

APPENDIX A 1

OFFICE / DEPARTMENT CODES

Sl.No.	Code No.	Name of office / department
1.	01	Vice-Chancellor's office
2.	02	Resistrar's office
3.	03	Comptroller's office
4.	04	Audit office
5.	05	Planning and development office
6.	11	Engg. Faculty office
7.	12	Chem. Engg. dept. office
8.	13	Met. Engg. dept. office
9.	14	Mathematics dept. office
10.	15	Physics dept. office
11	16	Chemistry dept. office
12	21	Civil Engg. faculty office
13	22	Civil Engg. dept. office
14	23	WRE dept. office
15	31	Mech. Engg. faculty office
16.	32	Mech. Engg. dept. office
17.	33	NAME dept. office
18.	34	IPE dept. office
19.	41	Elect. & Electronic Engg. faculty office
20.	42	EEE dept. office
21.	43	CSE dept. office
22.	51	Arch. & Planning faculty office
23.	52	Arch. dept. office
24.	53	U R P dept. office

APPENDIX A 1 [Contd.]

25.	54	Humanities dept. office
26.	61	F I R Institute office
27.	62	I A T office
28.	66	Computer Center office
29.	67	D A E R S office
30.	68	Controller's office
31.	69	Library office
32.	70	Energy Centre office
33.	76	D S W office
34.	77	Ahsan Ullah Hall office
35.	78	Nazrul Islam Hall office
36.	79	Sher-e-Bangla Hall office
37.	80	Titumeer Hall office
38.	81	Sharwardi Hall office
39.	82	Ladies Hall office
40.	83	DR. Rashid Hall office
41	84	Shahid Smritee Hall office
42.	85	Physical education office
43.	86	Central Cafetaria office
44.	87	Auditorium office
45.	96	Engineering office
46.	97	Medical Center office

APPENDIX A2

DATABASE DICTIONARY [model]

a. Catalogs of attributes:

Sl.No.	Name of attribute	Type	Width	Decimal	Allowable values	Description	Derivation
[1]	Empid	N	6	0	0->999999	Identifying attribute	From regis -ter book

b. Catalogs of data:

[1] HA [House allocation status]

Assigned values:

HA=1 for employees entitled to use govt. residence without HR deduction

HA=2 for employees using govt. residence with HR deduction

HA=3 for employees residing outside; they get HR allowance

HA=4 for employees residing in dormatory

c. Catalogs of database files:

[1] Name of file: B4.dbf [Table for calculation of HR allowance]

No. of data records = 4

Field	Field-name	Type	Width	Decimal
1.	BRANGE	N	1	0
2.	RANGE	C	11	0
3.	FAMT	N	5	0
4.	PCBASIC	N	3	2
5.	MINVALUE	N	4	0
6.	MAXVALUE	N	5	0
	Total		= 29 bytes	

APPENDIX- A 3

PROGRAM LISTINGS

```
* I M S for B U E T Dhaka
* Opening menu program with security control: OM.PRG
* Date: 12-1-92
* Set working environment & close all open files
SET HEADING OFF
SET SAFETY OFF
SET STATUS OFF
SET TALK OFF
CLEAR ALL
SET CONSOLE ON
SET DEVICE TO SCREEN
SET PRINT OFF
CLOSE ALL
* Body of the program
* Initialization of variables
mn=" "
NAME = "    "
CLEAR ALL
CLEAR
* Creation of a double lined box to hold the heading
@ 3,14 TO 11,68 DOUBLE
@ 5,16 SAY "     WELCOME TO INFORMATION MANAGEMENT SYSTEM"
@ 7,16 SAY "                          OF                              "
@ 9,16 say " BANGLDESH UNIVERSITY OF ENGG. & TECHNOLOGY, DHAKA"
@ 10,18 SAY "                                                  "
@ 12,2 SAY "                                                          "
* To accept user's valid ID-NO
@ 14,2 say "PLEASE ENTER YOUR USER-ID  AND PRESS ENTER KEY"
* To make the input invisible for security purpose
set color to n/n
ACCEPT " " TO mn
set color to w/n
* To open database file containing valid ID-Numbers.
USE PASS INDEX PASS
```

```
* To check the validity of ID-NO entered by the user.
seek &mn
if found()
  pwcnt=0
* If the user enters valid ID-NO, then he has to enter his valid
* name-code; three chances are given to enter valid name-code.
  do while pwcnt<3
        ?
        ?"PLEASE ENTER YOUR NAME-CODE AND PRESS ENTER KEY"
* To make the input invisible for security purpose
        SET COLOR TO n/n
        ACCEPT " " TO NAME
        set color to w/n
        if nam=name
* If the user successfully enters his valid name-code, then
* main menu program is called to offer system options.
            use
            do mm.prg
            exit
        else
* If the user fails to enter valid name-code, then the system
* displays a message and the user is advised to retry
            ?"INVALID SIGN-ON; RETRY"
            pwcnt=pwcnt+1
        endif
   enddo
 else
*  If  the  user fails to enter valid ID-NO, then  a  message  is
* diplayed and the user can either retry or quit.
   ?"INVALID ID-NUMBER"
   ans=" "
   wait "Do you want to retry ?[Y/N]" to ans
   ans=upper(ans)
   if ans="Y"
```

```
    clear
    do om
   endif
endif
* End of program for opening menu
*****************************************************************
* I M S : MAIN MENU PROGRAM
* Program-name:MM.PRG
* Date: 15-1-92
DO WHILE .T.
CLEAR
* Creation of a double lined box to hold the heading
@ 2,0 TO 18,79 DOUBLE
@ 4,20 SAY [INFORMATION MANAGEMENT SYSTEM : MAIN-MENU]
@ 6,1 TO 6,78 DOUBLE
* To display various options
@ 8,15 SAY [1. PERSONNEL MANAGEMENT]
@ 9,15 SAY [2. INCOME MANAGEMENT]
@ 10,15 SAY [3. SALARY MANAGEMENT]
@ 11,15 SAY [4. NON-SALARY HEADS MGMT.]
@ 12,15 SAY [5. SYSTEM MAINTENANCE]
@ 14,15 SAY [0. EXIT ]
* Initialize a variable to store the user's option.
STORE 0 TO selectnum
@ 16,45 SAY [SELECT :    ]
@ 16,55 GET selectnum PICTURE "9" RANGE 0,6
READ
* Calling of related sub-menu program according to user-
* selected option.
DO CASE
   CASE selectnum = 0
      RETURN
   CASE selectnum = 1
      CLEAR ALL
```

```
      DO PM.PRG
   CASE selectnum = 2
      CLEAR ALL
      DO IM.PRG
   CASE selectnum = 3
      CLEAR ALL
      DO SM.PRG
   CASE selectnum = 4
      CLEAR ALL
      DO NSHM.PRG
   CASE selectnum = 5
      CLEAR ALL
      DO SC.PRG
   ENDCASE
ENDDO
* End of program for main menu
*************************************************************************
* SYSTEM : I M S : MODULE : PERSONNEL MGMT.
* Program-name:IMSPM.PRG
* Date: 20-1-92
DO WHILE .T.
CLEAR
* Creation of a double lined box to hold the heading
@ 2,0 TO 16,79 DOUBLE
@ 4,25 SAY [I M S : PERSONNEL MANAGEMENT      ]
@ 6,1 TO 6,78 DOUBLE
* To display various options
@ 8,3 SAY "1. List of Teachers in [dept-code]"
@ 8,40 say "2. List of Employee in [dept-code]"
@ 9,3 say "3. List of Teachers holding Ph.D."
@ 9,40 say [4. List of Employee on leave]
@ 10,3 say [5. List of Adhoc employee]
@ 10,40 say "6. List of Employee in [scale-code]"
@ 11,3 say "7. Academic records of [emp-id]"
```

```
@ 11,40 say "8. Address of [emp-id]"
@ 12,3 say [9. List of Female(1) / Non muslim(2) Employee]
@ 14,3 SAY [0.EXIT ]
* Initialize a variable to store the user's option.
STORE 0 TO selectnum
@ 14,45 SAY [SELECT :    ]
@ 14,55 GET selectnum PICTURE "9" RANGE 0,9
READ
* Calling of related sub-menu program according to user-
* selected option.
DO CASE
   CASE selectnum = 1
      clear all
      DO PM1.PRG
   CASE selectnum = 0
      CLEAR ALL
      RETURN
   CASE selectnum=2
      clear all
      do pm2.prg
   CASE selectnum=3
      clear all
      do pm3.prg
   CASE selectnum=4
      clear all
      do pm4.prg
   CASE selectnum=5
      clear all
      do pm5.prg
   CASE selectnum=6
      clear all
      do pm6.prg
   CASE selectnum=7
      clear all
```

```
      do pm7.prg
   CASE selectnum=8
      clear all
      do pm8.prg
   CASE selectnum=9
      clear all
      do pm9.prg
ENDCASE
ENDDO
* End of program for sub-menu "personnel management"
*****************************************************************
* I M S : INCOME MANAGEMENT
* Program-name:IM.PRG      DATE : 12-12-92
DO WHILE .T.
CLEAR
* Creation of a double lined box to hold the heading
@ 2,0 TO 18,79 DOUBLE
@ 4,25 SAY [I M S : INCOME MANAGEMENT ]
@ 6,1 TO 6,78 DOUBLE
* To display various options
@ 8,20 SAY [1.DAILY INCOME REPORT]
@ 10,20 SAY [2.PERIODICAL INCOME REPORT]
@ 12,20 SAY [0.EXIT]
* Initialize a variable to store the user's option.
STORE 0 TO selectnum
@ 15,45 SAY [SELECT :    ]
@ 15,55 GET selectnum PICTURE "9" RANGE 0,2
READ
* Calling of related sub-menu program according to user-
* selected option.
DO CASE
   CASE selectnum = 0
      CLEAR ALL
```

```
    CASE selectnum = 1
       CLEAR ALL
       DO DIR.PRG
    CASE selectnum = 2
       CLEAR ALL
       DO MIR.PRG
    ENDCASE
ENDDO
* End of program for sub-menu " income management "
*************************************************************************
* I M S : SALARY MANAGEMENT
* Program-name:IMSSM.PRG
* Date: 28-1-92
CLOSE ALL
DO WHILE .T.
CLEAR
* Creation of a double lined box to hold the heading
@ 2,0 TO 18,79 DOUBLE
@ 4,25 SAY [I M S : SALARY MANAGEMENT ]
@ 6,1 TO 6,78 DOUBLE
* To display various options
@ 8,20 SAY [1.SALARY BILLING ]
@ 9,20 SAY [2.CONSOLIDATED SALARY REPORT ]
@ 10,20 SAY [3.MONTHLY DEDUCTION & CONTRIBUTION]
@ 11,20 SAY [4.DEPT-WISE SALARY REPORT]
@ 13,20 SAY [0.EXIT]
* Initialize a variable to store the user's option.
STORE 0 TO selectnum
@ 16,45 SAY [SELECT :    ]
@ 16,55 GET selectnum PICTURE "9" RANGE 0,4
READ
* Calling of related sub-menu program according to user-
* selected option.
DO CASE
```

```
    CASE selectnum = 0
       CLEAR ALL
       RETURN
    CASE selectnum = 1
       CLEAR ALL
       DO SBILL.PRG
    CASE selectnum = 2
       CLEAR ALL
       DO CSR.PRG
    CASE selectnum = 3
       clear all
       do mdcs.prg
    CASE selectnum = 4
       CLEAR ALL
       DO DSR.PRG
    ENDCASE
ENDDO
* End of program for sub-menu " salary management "
*****************************************************************
* I M S : NON-SALARY HEADS MANAGEMENT
* Program-name:IMSNSHM.PRG
* Date: 30-1-92
CLOSE ALL
DO WHILE .T.
CLEAR
* Creation of a double lined box to hold the heading
@ 2,0 TO 18,79 DOUBLE
@ 4,23 SAY [I M S :NON-SALARY HEADS MANAGEMENT]
@ 6,1 TO 6,78 DOUBLE
* To display various options
@ 8,15 SAY [1.HEAD-WISE EXPENDITURE STATEMENT]
@ 9,15 SAY [2.DEPT-WISE EXPENDITURE STATEMENT]
@ 11,15 SAY [0.EXIT]
* Initialize a variable to store the user's option.
```

```
STORE 0 TO selectnum
@ 16,45 SAY [SELECT :    ]
@ 16,55 GET selectnum PICTURE "9" RANGE 0,3
READ
* Calling of related sub-menu program according to user-
* selected option.
DO CASE
   CASE selectnum = 0
      CLEAR ALL
      RETURN
   CASE selectnum = 1
      CLEAR ALL
      DO HER.PRG
   CASE selectnum = 2
      CLEAR ALL
      DO DER.PRG
   ENDCASE
ENDDO
* End of program for sub-menu " non-salary heads management "
*************************************************************************
* I M S : SYSTEM MAINTENANCE
* Program-name:IMSSYSM.PRG
* Date: 15-2-92
DO WHILE .T.
CLEAR
* Creation of a double lined box to hold the heading
@ 2,0 TO 18,79 DOUBLE
@ 4,25 SAY [I M S : SYSTEM MAINTENANCE]
@ 6,1 TO 6,78 DOUBLE
* To display various options
@ 8,20 SAY [1. DATA ENTRY]
@ 9,20 SAY [2. CHANGE OF DATABASE STRUCTURE]
@ 10,20 SAY [3. CHANGE OF SECURITY CODE]
@ 11,20 say [4. INDEXING DATABASE FILES]
```

```
@ 12,20 say [5. DATA EDIT]
@ 13,20 SAY [0.EXIT ]
* Initialize a variable to store the user's option.
STORE 0 TO selectnum
@ 16,45 SAY [SELECT :    ]
@ 16,55 GET selectnum PICTURE "9" RANGE 0,5
READ
* Calling of related sub-menu program according to user-
* selected option.
DO CASE
   CASE selectnum = 0
      CLEAR ALL
      RETURN
   CASE selectnum = 1
      CLEAR ALL
      DO DE.PRG
   CASE selectnum = 2
      CLEAR ALL
      DO CDB.PRG
   CASE selectnum = 3
      CLEAR ALL
      USE PASS
      SET STATUS ON
      EDIT
      SET STATUS OFF
   CASE selectnum = 4
      CLEAR ALL
      do indx.prg
   CASE selectnum = 5
      clear all
      do dedit.prg
   ENDCASE
ENDDO
* End of program for sub-menu " system maintenance "
```

```
* pm1.prg : Program for list of Teacher in ... dept.
* System: IMS ; Module: PM
*Declaration of variables
public mpost,mcode,mdepartment,mempid,msubcat,mbasic,mdate
clear
set talk off
close all
* Setting of procedure file ; to call related procedure
set procedure to pfpm
* Initialization of variables
mdc=00
mpost="                    "
store DATE() - 52000 TO mdate
@ 2,2 say "Enter dept.code no.: " get mdc PICTURE "99"
READ
* Calling of the procedure to check the validity of dept-code
do check_dc
if mn =0
  ?"No such department in BUET"
  ?
*Calling of the procedure to display the list of valid dept-codes
  do disp_dc
  mdc=00
  @ 3,2 say "Enter valid dept-code :" get mdc PICTURE "99"
  read
endif
* Calling of the procedure to check the validity of dept-code
do check_dc
clear
* If the user again fails to enter valid dept-code, then such a
* a message is displayed and he is advised to get out.
if mn=0
  TEXT
```

```
do case
     case &mscat=0
          msubcat=2
          mpost="Professor"
    case &mscat=1
          msubcat=3
          mpost="Associate Professor"
     case &mscat=2
          msubcat=4
          mpost="Assistant Professor"
     case &mscat=3
          msubcat=5
          mpost="Lecturer"
     case &mscat=4
          msubcat=00
     otherwise
        ?"Illegal entry"
        wait
        return
 endcase
wait "Output to the Printer or Screen [P/S] ? " TO ans
ans=upper(ans)
if ans="P"
 set print on
 ? chr(27) + chr(15)
endif
clear
? "            Please wait !    "
x=mdc*10000
* Initialization of variables
mempid=000000
select a
use m1 index m1_id
rein
```

```
select b
use m2 index m2_id
rein
set safe off
erase zz.dbf
set safe on
select a
JOIN WITH b TO zz FOR empid = b-> empid FIELDS
empid,ename,basic,subcat,catcode,doj
clear
?"  BANGLADESH UNIVERSITY OF ENGG. & TECHNOLOGY, DHAKA."
?
?"List of Teachers in dept. of ",mdepartment
?
?"ID-NO  NAME                                          POST"
?
use zz
go top
* Initialization of variables
cnt=0
tcnt=0
if msubcat=00
do while .not. eof()
if empid<=(x+9999) .and. empid>x .and. catcode=1 .and. doj>mdate
 msubcat=subcat
  do case
    case msubcat = 1
       mpost = "Professor"
    case msubcat = 2
       mpost="Professor"
    case msubcat=3
       mpost="Associate Prof."
    case msubcat=4
       mpost="Assistant Prof."
```

```
    case msubcat=5
        mpost="Lecturer"
  endcase
  ?EMPID," ",ENAME," ",mpost
  ?
  tcnt=tcnt+1
  cnt=cnt+1
 endif
 skip
 if cnt>6
    cnt=0
    wait
 endif
enddo
else
do while .not. eof()
 if empid<=(x+9999) .and. empid>x .and. catcode=1 .and.
subcat=msubcat .and. doj>mdate
?empid," ",ename," ",mpost
?
cnt=cnt+1
tcnt=tcnt+1
endif
skip
   if cnt>6
      cnt=0
      wait
    endif
enddo
endif
?"Total number of teachers = ",tcnt
?
res= " "
wait "Observation complete ? [Y]:" to res
```

```
ans=upper(ans)
if ans="P"
   ? chr(27) + chr(18)
   set print off
endif
close all
* eop
*****************************************************************
* pm2.prg program for list of employee in ... dept.
* system :IMS ; Module: PM
clear
clear all
*Declaration of variables
public mdepartment,mdc,mdate,mcat,mscat,mclass
* Setting of procedure file ; to call related procedure
set procedure to pfpm
set talk off
@ 2,2 say " WELCOME TO LIST GENERATION"
* Initialization of variables
mdc=00
mclass="                    "
ans=" "
@ 4,1 SAY "Please enter DEPT-CODE: " get mdc picture "99"
read
* Calling of the procedure to check the validity of dept-code
do check_dc
if mn=0
  ?"No such department in BUET"
  ?
*Calling of the procedure to display the list of valid dept-codes
  do disp_dc
  mdc=00
  @ 2,18 say "Enter valid dept-code :" get mdc PICTURE "99"
  read
```

```
endif
* Calling of the procedure to check the validity of dept-code
do check_dc
* If the user again fails to enter valid dept-code, then such a
* a message is displayed and he is advised to get out.
if mn=0
   clear
   TEXT
         You have failed to enter valid dept-code.

                        Please get out.
   ENDTEXT
   wait
   return
endif
clear
do pfpm3 with mdepartment
store date() - 52000 to mdate
@ 2,2 say " Enter the following optional condition or press
enter"
@ 4,2 say "Who have joined after the date :" get mdate
read
clear
@ 2,2 say "Select any one of the following options"
?
TEXT
            0. 3rd class

            1. 3rd class ( Technical )

            2. 4th class

            3. all
ENDTEXT
```

```
* Initialization of variables
mscat=0
mcat=00
accept "Select :" to mscat
if &mscat>3
   ?"Illegal entry"
   wait
   accept "Select [ type 0,1,2 or 3 and press Enter ] " to mscat
endif
do case
    case &mscat=0
            mcat=6
            mclass="3rd class"
    case &mscat=1
            mcat=7
            mclass="3rd class(Technical)"
    case &mscat=2
            mcat=8
            mclass="4th class"
    case &mscat=3
            mcat=00
    otherwise
        ?" Illegal entry"
        wait
        return
    endcase
wait "Output to Printer or Screen [P/S] ? " TO ans
ans=upper(ans)
if ans="P"
   set print on
   ? chr(27) + chr(15)
else
endif
x=mdc*10000
```

```
clear
?"       BANGLADESH UNIVERSITY OF ENGG. AND TECHNOLOY, DHAKA."
?
? "       List of Employees in the department / office  of"
?
? "             ",mdepartment
?
? "ID-NO   NAME                                DATE OF        CLASS"
? "                                              JOINING"
? " -----  ----------------------------    ----------   -------"
select a
use m1 index m1_id
select b
use m2 index m2_id
set safety off
select a
join with b to z1 for empid = b-> empid fields empid,ename,basic
,catcode,doj
set safe on
use z1
go top
* Initialization of variables
cnt=0
tcnt=0
if mcat=00
do while .not. eof()
 if empid>x .and. empid<=(x+9999) .and. catcode>5 .and. doj>mdate
   cnt=cnt+1
   tcnt=tcnt+1
   ?
   mcat=catcode
   do case
        case mcat=6
             mclass="3rd class"
```

```
        case mcat=7
             mclass="3rd class(Technical)"
        case mcat=8
             mclass="4th class"
    endcase
   ?EMPID," ",ENAME,doj," ",mclass
 endif
 skip
 if cnt>6
    cnt=0
    wait
 endif
enddo
else
do while .not. eof()
if empid>x .and. empid<=(x+9999) .and. catcode=mcat .and.
doj>mdate
?
cnt=cnt+1
tcnt=tcnt+1
?empid," ",ename,doj," ",mclass
endif
skip
  if cnt>6
     cnt=0
     wait
   endif
enddo
endif
?
?" Total number of employees  = ",tcnt
res=" "
?
wait "Observation complete ?[Y]: " to res
```

```
res = upper(res)
if ans="P"
? chr(27) + chr(18)
set print off
endif
ERASE Z1
close all
*eop
*****************************************************************
* System : I M S ; Module : P M
* Program-name: pm3.prg
set safe off
clear
clear all
*Declaration of variables
public mdepartment,mdc,mdate
* Setting of procedure file ; to call related procedure
set procedure to pfpm
set talk off
@ 2,2 say " WELCOME TO LIST GENERATION"
* Initialization of variables
mdc=00
ans=" "
@ 4,1 SAY "Please enter DEPT-CODE: " get mdc picture "99"
read
* Calling of the procedure to check the validity of dept-code
do check_dc
if mn=0
 ?"No such department in BUET"
 ?
* Calling of the procedure to display the list of valid
* dept-codes.
 do disp_dc
 mdc=00
```

```
 @ 2,18 say " Enter valid dept-code :" get mdc PICTURE "99"
 read
endif
* Calling of the procedure to check the validity of dept-code
do check_dc
* If the user again fails to enter valid dept-code, then such a
* a message is displayed and he is advised to get out.
if mn=0
   clear
   TEXT
            You have failed to enter valid dept-code.

                        Please get out.
   ENDTEXT
   wait
   return
endif
do pfpm3 with mdepartment
clear
store date() - 52000 to mdate
@ 2,2 say " Enter the following optional codition or press enter"
@ 4,2 say "Who have joined after the date :" get mdate
read
clear
@ 2,2 say "Select any one of the following options"
?
text
                0. List of Professors

                1. List of Associate Professors

                2. List of Assistant Professors

                3. All
```

```
endtext
* Initialization of variables
mscat=0
msubcat=00
accept "Select : " to mscat
if &mscat>3
   ?"Illegal entry"
   wait
   accept "Select [ type 0,1,2 or 3 and press Enter ]" to mscat
endif
do case
    case &mscat=0
        mpost="Professor"
        msubcat=2
    case &mscat=1
        mpost="Associate Professor"
        msubcat=3
    case &mscat=2
        msubcat=4
        mpost="Assistant Professor"
    case &mscat=3
        msubcat=00
    otherwise
        ?"Illegal entry"
        wait
        return
endcase
wait "Output to Printer or Screen [P/S] ? " TO ans
ans=upper(ans)
if ans="P"
   set print on
   ? chr(27) + chr(15)
else
endif
```

```
x=mdc*10000
clear
? "    BANGLADESH UNIVERSITY OF ENGG. & TECHNOLOGY, DHAKA."
?
? "List of Teachers holding Ph.D. degree in the department of"
?
? "           ",mdepartment
?
? "ID-NO     NAME                          DATE OF JOINING      POST"
?" -----    ----------------------         -----------       -----"
select a
use m1 index m1_id
select b
use m2 index m2_id
select a
join with b to zz for empid = b-> empid
use zz
go top
* Initialization of variables
tcnt=0
cnt=0
if msubcat=00
do while .not. eof()
if empid>x .and. empid<=(x+9999) .and. phdyr>0
  do case
    case subcat=2
        mpost="Professor"
    case subcat=3
        mpost="Associate Professor"
    case subcat=4
        mpost="Assistant Professor"
  endcase
   ?
   ?EMPID," ",ENAME,DOJ," ",mpost
```

```
    tcnt=tcnt+1
    cnt=cnt+1
 endif
  skip
  if cnt>6
     cnt=0
     wait
  endif
 enddo
 else
 do while .not. eof()
 if empid>x .and. empid<=(x+9999) .and. phdyr>0 .and.
 subcat=msubcat
 ?
 ?empid," ",ename,doj," ",mpost
 tcnt=tcnt+1
 cnt=cnt+1
 endif
 skip
 if cnt>6
    cnt=0
    wait
 endif
 enddo
 endif
 ?" Total number of teachers = ",tcnt
 res=" "
 ?
 wait "Observation complete ?[Y]: " to res
 res=upper(res)
 if ans="P"
 chr(27) + chr(18)
 set print off
 endif
```

```
close all
*****************************************************
* pm4.prg : Program for List of employee on leave
* System: IMS  module: PM
set safe off
erase zz.dbf
clear
set talk off
clear all
@ 2,2 say " WELCOME TO LIST GENERATION "
* Initialization of variable
store date() - 52000 to mdate
select a
use m1 index m1_id
select b
use leave index leave
select a
join with b to zz.dbf for empid = b->empid
use zz.dbf
index on empid to z_id
use
select c
use zz index z_id
select d
use m2 index m2_id
select c
join with d to z1 for empid = d->empid
use z1
clear
@ 2,2 say "Enter the following optional condition or press enter"
@ 4,2 say "Who have joined after the date :" get mdate
read
clear
@ 2,2 say "Select any one of the following options"
```

```
text
                0. Teachers

                1. Officers

                2. 3rd class

                3. 4th class
endtext
* Initialization of variable
mscat=00
accept "Select : " to mscat
if &mscat>3
?"Illegal entry"
wait
accept " Select again :" to mscat
endif
if &mscat>3
?"You have failed to enter valid option ; Please get out"
wait
return
endif
go top
wait "Output to Printer or Screen [P/S] ? " TO ans
ans=upper(ans)
if ans="P"
   set print on
   ? chr(27) + chr(15)
endif
clear
?"       BANGLADESH UNIVERSITY OF ENGG. & TECHNOLOGY, DHAKA."
?
?"              List of employees on leave"
```

```
?"ID-NO     NAME                                          Starting    Expired"
?"                                                          Date        Date"
?"------    ---------------------------                 -------     -------"
* Initialization of variables
cnt=0
tcnt=0
do case
case &mscat=0
    do while .not. eof()
       if catcode=1
       ?
       ?EMPID," ",ENAME,sdate," ",edate
       cnt=cnt+1
       tcnt=tcnt+1
       endif
       skip
       if cnt>6
       cnt=0
       wait
       endif
     enddo
case &mscat=1
   do while .not. eof()
      if catcode=2 .or. catcode=3 .or. catcode=4 .or. catcode=5
      ?
      ?empid," ",ename,sdate," ",edate
      cnt=cnt+1
      tcnt=tcnt+1
      endif
      skip
      if cnt>6
         cnt=0
         wait
```

```
        endif
      onddo
  case &mscat=2
      do while .not. eof()
         if catcode=6 .or. catcode=7
         ?
         ?empid," ",ename,sdate," ",edate
         cnt=cnt+1
         tcnt=tcnt+1
         endif
         skip
         if cnt>6
            cnt=0
            wait
         endif
      enddo
 case &mscat=3
      do while .not. eof()
        if catcode=8
        ?
        ?empid," ",ename,sdate," ",edate
        cnt=cnt+1
        tcnt=tcnt+1
        endif
        skip
        if cnt>6
           cnt=0
           wait
        endif
     enddo
otherwise
    ?"Illegal entry"
    wait
    return
```

```
endcase
erase z_id.ndx
?
?" Total number of employees on leave =",tcnt
res=" "
?
wait "Observation complete ? [Y]:"
res=upper(res)
if ans="P"
chr(27) + chr(18)
set print off
endif
close all
************************************************************
* pm5.prg : Program for list of adhoc employee
* System : I M S ; Module : P M
clear
set talk off
clear all
@ 2,2 say " WELCOME TO LIST GENERATION "
store date() - 52000 to mdate
@ 2,2 say "Enter the following optional condition or press Enter"
@ 4,2 say "Who have joined after the date :" get mdate
read
clear
@ 2,2 say "Select any one of the following options"
?
TEXT
                0. Teachers

                1. Officers

                2. 3rd class
```

3. 4th class

4. all

```
endtext
* Initialization of variable
mscat=0
accept "Select : " to mscat
if &mscat>4
 ?"Illegal entry"
 wait
 accept "Select again :" to mscat
endif
if &mscat>4
?"You have failed to enter valid option; please get out"
wait
return
endif
wait "Output to Printer or Screen [P/S] ? " TO ans
ans=upper(ans)
if ans="P"
   set print on
   ? chr(27) + chr(15)
else
endif
select a
use m1 index m1_id
select b
use m2 index m2_id
erase zz.dbf
select a
join with b to zz for empid= b->empid
use zz
clear
?"       BANGLADESH UNIVERSITY OF ENGG. & TECHNOLOGY, DHAKA."
```

```
?"                LIST OF ADHOC EMPLOYEES"
?
?"ID-NO      NAME                                        DATE OF JOINING"
?"------     ------------------------------------        ---------------"
GO TOP
* Initialization of variables
cnt=0
tcnt=0
do case
    case &mscat=0
    do while .not. eof()
       if jobstatus = "A" .and. catcode=1
       ?EMPID,"  ",ENAME,DOJ
       cnt=cnt+1
       tcnt=tcnt+1
       ENDIF
       skip
       if cnt>6
          cnt=0
          wait
       endif
   enddo
   case &mscat=1
   do while .not. eof()
      if jobstatus ="A" .and. catcode=2 .or. catcode=3 .or.
        catcode=4 .or. catcode=5
       ?
       ?empid," ",ename,doj
       cnt=cnt+1
       tcnt=tcnt+1
       skip
       if cnt>6
           cnt=0
```

```
        wait
    endif
    enddo
case &mscat=2
do while .not. eof()
    if jobstatus="A" .and. catcode=6 .or. catcode=7
    ?
    ?empid," ",ename,doj
    cnt=cnt+1
    tcnt=tcnt+1
    endif
    skip
    if cnt>6
        cnt=0
        wait
    endif
enddo
case &mscat=3
do while .not. eof()
    if jobstatus="A" .and. catcode=8
    ?
    ?empid," ",ename,doj
    cnt=cnt+1
    tcnt=tcnt+1
    endif
    skip
    if cnt>6
        cnt=0
        wait
    endif
enddo
case &mscat=4
do while .not. eof()
    if jobstatus="A"
```

```
        ?ompid," ",ename,doj
        cnt=cnt+1
        tcnt=tcnt+1
        endif
        skip
        if cnt>6
           cnt=0
            wait
        endif
    enddo
endcase
?
?" Total number of adhoc employees = ",tcnt
res=" "
?
wait "Observation complete ? [Y] :" to res
res=upper(res)
if ans="P"
chr(27) + chr(18)
set print off
endif
close all
* pm6.prg : Program for list of employee in  scale....
* System: IMS ; Module: PM
*Declaration of variables
public mscale,msc
clear
close all
set talk off
* Setting of procedure file ; to call related procedure
set procedure to pfpm
* Initialization of variables
msc=00
```

```
@ 2,2 say "Enter scale code no.: " get msc PICTURE "99"
READ
if msc=0 .OR. msc>20
  ?"Scale code is not vaild"
  ?"Please enter valid scale code [ 1 .... 20]"
  wait
  return
endif
store DATE() - 52000 to mdate
@ 2,2 say "Enter the following optional condition or press enter"
@ 4,2 say "Who have joined after the date :" get mdate
read
```



```
wait "Output to the Printer or Screen [P/S] ? " TO ans
ans=upper(ans)
if ans="P"
 set print on
 ? chr(27) + chr(15)
endif
mempid=000000
erase zz.dbf
clear
?"      BANGLADESH UNIVERSITY OF ENGG. AND TECHNOLOGY, DHAKA."
?
?"                List employee in scale of ",msc
?
?"ID-NO    NAME                          DATE OF BIRTH  DISTRICT"
?"-----    ----------------------------    -----------  -------"
select a
use m1 index m1_id
select b
use m2 index m2_id
SET TALK OFF
```

```
select a
join with b to zz for empid =b-> empid
use zz
go top
* Initialization of variables
cnt=0
tcnt=0
do while .not. eof()
 if catcode > 5 .and. scale = msc
  ?
  tcnt=tcnt+1
  cnt=cnt+1


                                        Page-
 ?EMPID," ",ENAME,DOB," ",district
 endif
 skip
 if cnt>6
    cnt=0
    wait
 endif
enddo
?
?" Total number of employee in scale of",msc," = ",tcnt
res=" "
?
wait "Observation complete ? [Y] :"
res=upper(res)
if ans="P"
   chr(27) + chr(18)
   set print off
endif
close all
* End of program for list of employee in [scale-code]
```

```
*************************************************************************
* pm7.prg : Program for academic record generation
clear
set talk off
clear all
@ 2,2 say " WELCOME TO ACADEMIC RECORD GENERATION "
* Initialization of variables
mid=000000
ans=" "
@ 4,1 SAY "Please enter the ID-NO: " get mid picture "999999"
read
use m1 index m1_id
```



```
locate for empid=mid
if eof()
  ?"No such ID-NO found"
  wait
  return
endif
wait "Output to Printer or Screen [P/S] ? " TO ans
ans=upper(ans)
if ans="P"
   set print on
   ? chr(27) + chr(15)
else
endif
mname=ename
clear
?" Academic records of ",ename
?
?"                          ID-NO  :",mid
?
? "Examination    Div./Class/GPA    Year of passing"
```

```
?" S S C                ",sscdiv,"              ",sscyr
?
?" H S C                ",hscdiv,"              ",hscyr
?
?" Graduation           ",gradcl,"              ",gradyr
?
?" Master's       ",MGPA,"Out of",mt,"       ",myr
?
?" Ph. D.                                  ",PHDYR
?
res=" "
```



```
wait "Observation complete ? [Y] :"
res=upper(res)
if ans="P"
chr(27) + chr(18)
set print off
endif
close all
* End of program for academic record generation
*************************************************************
* pm8.prg : Program for address generation
clear
set talk off
clear all
@ 2,2 say " WELCOME TO ADDRESS GENERATION "
* Initialization of variables
mid=000000
ans=" "
@ 4,1 SAY "Please enter the ID-NO: " get mid picture "999999"
read
use m1 index m1_id
```

```
locate for empid=mid
if eof()
  ?"No such ID-NO found"
  wait
  return
endif
wait "Output to Printer or Screen [P/S] ? " TO ans
ans=upper(ans)
if ans="P"
   set print on
   ? chr(27) + chr(15)
else
endif
                                        Page-
clear
? "NAME              :",ENAME
?
? "FATHER'S NAME     :",FNAME
?
? "PERMANENT ADDRESS:",PERADDRESS
?
? "PRESENT ADDRESS   :",PRADDRESS
res=" "
?
wait "Observation complete ? [Y] :" to res
res=upper(res)
if ans="P"
chr(27) + chr(18)
set print off
endif
close all
* End of program for address of [ empid ]
*************************************************************
* pm9.prg : Program for list of female/non muslim employees
```

```
close all
set talk off
set safe off
*Declaration of variables
public mattr,mval,mtitle,mtitle0,mscat
* Initialization of variables
store space(20) to mattr
store space(2) to mval
erase zz.dbf
mtitle="
mtitle0="
clear
clear all
@ 2,2 say " WELCOME TO LIST GENERATION "
mid=0
ans=" "
store date() - 52000 to mdate
@ 4,1 SAY "Please enter CHOICE-NO: " get mid picture "9"
@ 6,1 say "[ Enter '1' for female ; '2' for non-muslim]"
read
do case
case mid=1
    mattr="SEX"
    mval="F"
    mtitle="LIST OF FEMALE EMPLOYEES IN BUET, DHAKA"
    mtitle0="Total number of female employees"
case mid=2
    mattr="religion"
    mval="N"
    mtitle="LIST OF NON-MUSLIM EMPLOYEES IN BUET, DHAKA"
    mtitle0="Total number of non-muslim employees"
otherwise
    ?"Illegal entry"
    wait
```

```
    return
endcase
clear
@ 2,2 say "Enter the following optional condition or press enter"
@ 4,2 say "Who have joined after the date :" get mdate
read
clear
@ 2,2 say "Select any one of the following options"
?
text
                0. Teachers

                1. Officers

                2. 3rd class

                3. 4th class

                4. All
endtext
* Initialization of variable
mscat=0
accept "Select :" to mscat
if &mscat>4
?"Illegal entry"
wait
accept "Select again :" to mscat
endif
if &mscat>4
?"You have failed to enter valid option; please get out"
wait
return
endif
wait "Output to Printer or Screen [P/S] ? " TO ans
```

```
ans=upper(ans)
if ans="P"
   set print on
   ? chr(27) + chr(15)
else
endif
clear
@ 2,6 say "Please wait ! processing is going on ....."
select a
use m1 index m1_id
select b
use m2 index m2_id
select a
join with b to zz for empid = b-> empid
set procedure to pfpm
use zz
clear
go top
@ 1,2 SAY "      &mtitle      "
 ?
 ? "ID-NO.   NAME                                DATE OF JOINING"
 ?" ------   --------------------------      ---------------"
* Initialization of variables
tcnt=0
cnt=0
do case
    case &mscat=0
       do while .not. eof()
        if &mattr=mval
           if catcode=1
              ?
              ?empid," ",ename,doj
              cnt=cnt+1
              tcnt=tcnt+1
```

```
        endif
      endif
      skip
      if cnt>6
          cnt=0
          wait
      endif
     enddo
case &mscat=1
    do while .not. eof()
    if &mattr=mval
    if catcode=2 .or. catcode=3 .or. catcode=4 . or. catcode=5
       ?
       ?empid," ",ename,doj
       cnt=cnt+1
       tcnt=tcnt+1
       endif
     endif
     skip
     if cnt>6
         cnt=0
         wait
     endif
    enddo
 case &mscat=2
    do while .not. eof()
     if &mattr=mval
        if catcode=6 .or. catcode=7
        ?
        ?empid," ",ename,doj
        cnt=cnt+1
        tcnt=tcnt+1
        endif
      endif
```

```
      skip
      if cnt>6
          cnt=0
          wait
      endif
    enddo
case &mscat=3
    do while .not. eof()
    if &mattr=mval
       if catcode=8
       ?
       ?empid," ",ename,doj
       cnt=cnt+1
       tcnt=tcnt+1
       endif
    endif
    skip
    if cnt>6
         cnt=0
         wait
    endif
    enddo
case &mscat=4
    do while .not. eof()
    if &mattr=mval
         ?
         ?empid," ",ename,doj
         cnt=cnt+1
         tcnt=tcnt+1
    endif
    skip
     if cnt>6
         cnt=0
         wait
```

```
        endif
       enddo
endcase
?"  &mtitle0 =",tcnt
if ans="P"
chr(27) + chr(18)
set print off
endif
res=" "
?
wait "Observation complete ? [Y] :" to res
res=upper(res)
close procedure
close all
* End of program for list of non-muslims or female
*************************************************************
* Procedure file for personnel management system
* Program-name PFPM.PRG ; Calling program-name:PMn.PRG
* Procedure to retrieve department-name according to department
* code entered by the user.
procedure pfpm3
parameter mdept
do case
case mdc=12
 mdept="CHEMICAL ENGINEERING"
CASE MDC=13
 mdept="METALURGICAL ENGINEERING"
case mdc=14
 mdept="MATHEMATICS"
case mdc=15
 mdept="PHYSICS"
case mdc=16
 mdept="CHEMISTRY"
case mdc=22
```

```
 mdept="CIVIL ENGINEERING"
case mdc=23
 mdept="WATER RESOURCE ENGINEERING"
case mdc=32
 mdept="MECHANICAL ENGINEERING"
case mdc=33
 mdept="NAVAL ARCHITECHTURE AND MARINE ENGG."
case mdc=34
 mdept="INDUSTRIAL AND PRODUCTION ENGG."
case mdc=42
 mdept="ELECTRICAL & ELECTRONIC ENGG."
case mdc=43
 mdept="COMPUTER SCIENCE & ENGINEERING"
case mdc=52
 mdept="ARCHITECHTURE"
case mdc=53
 mdept="URBUN & REGIONAL PLANNING"
case mdc=54
 mdept="HUMANITIES"
else
  ? " Illegal entry ! "
endcase
*************
* Procedure to retrieve data for list of non-muslim or
* female employee in BUET
procedure pfpm9
parameters mattr,mvalue
use m1 index m1_id
go top
do while .not. eof()
   if mattr="mvalue"
    ?"empid,"  ",ename,doj,"    ",district
   endif
   skip
```

```
enddo
use
* Procedure to check the validity of department code entered
* by the user.
procedure check_dc
public mn
use dc
mn=0
locate for DN=mdc
if eof()
   mn=0
 else
   mn=1
endif
use
return
* Procedure to display the list of valid department codes
* [ To help the user ]
procedure disp_dc
TEXT
 The followings are the valid dept-code in B U E T, Dhaka.

12 for ChemE        22 for CE          42 for EEE
13 for MetE         23 for WRE         43 for CSE
14 for Math         32 for ME          52 for ARCH
15 for Phy          33 for NAME        53 for URP
16 for Chem         34 for IPE         54 for HUM
ENDTEXT
return
*************************************************************************
```

```
* SALARY BILLING; SBILL.PRG ; SYSTEM  IMS
close all
* Declaration of variables
public melectric,mmedical,mda,mgid,mtrans,mmedicine,mhbl,mgpa,;
       mtel,menta,mgas,id,temp,mbasic,maddcharge,mgass,;
       moclub,moasso,mclub3,masso3,masso3t,masso4
public mclub4,mbfd,mhr,mhrd,mhalla,mheada,mdeana,mfestia,;
       mempid,mename,ga,na,nd,mx
* Setting of procedure file
set procedure to sb1
set safety off
set talk off
* Initialization of variables
na=000000
ga=000000
nd=000000
mx=0
clear
do input_id          ; procedure to input employee's id-no.
use m1 index m1_id   ;
locate for empid=id
mename=ename
use
do input_m2          ;to check the validity of id-no entered
if mx=1              ; if id-no. is invalid, then returns
return               ; to menu.
endif
@ 16,1 say "You should index all reqd. files once in a month"
Wait "Do you want to index all the reqd. files ? [Y/N]:" to ans
ans = upper(ans)
if ans = "Y"
 do make_ndx         ; procedure to index all database files.
```

```
endif
wait " The month is of festival allounce[ Y/N ]:" to ff
ff=upper(ff)
if ff="Y"
 mfestia=mbasic      ; calculation of festival allowance
 clear
 @ 4,40 say "Festival allowance="
 @ 4,60 say mfestia
else
 clear
 mfestia=0.00
endif
ga=mbasic+mfestia          ; calculation of gross amount
@ 1,1 say "Salary bill for"
@ 1,17 say mename
@ 1,53 say "ID-No= "
@ 1,60 say mempid
@ 3,1 say "PAYS AND ALLOWANCES:"
@ 4,5 say "Basic Pay        ="
@ 4,25 say mbasic
* Calculation of electric bill
do cal_bill with "a1","a1_id",melectric
@ 16,5 say "Electric bill   ="
@ 16,25 say  melectric
nd=melectric                    ; calculation of net deduction
@ 9,1 say "DEDUCTIONS:"
do cal_gas with mgass           ; calculation of gas bill
@ 15,40 say "Gas bill           = "
@ 15,60 say mgass
nd=nd+mgass
do cal_med                      ; calculation of medical allowance
@ 8,5 say "Medical allc.    = "
```

```
@ 8,25 say mmedical
@ 5,5 say "P. A.              = "
@ 5,25 say mpa
@ 5,40 say "P. P.                ="
@ 5,60 say mpp
@ 15,5 say "G. P. F.          = "
@ 15,25 say mgpf
ga=ga+mmedical+mpa+mpp          ; calculation of gross amount
nd=nd+mgpf
* calculation of transport bill
do cal_bill with  "a3","a3_id",mtrans
@ 12,5 say "Transport bill  ="
@ 12,25 say mtrans
* calculation of telephone bill
do cal_bill with "a4","a4_id",mtel
@ 14,5 say "Telephone bill  ="
@ 14,25 say mtel
* calculation of house building loan
do cal_bill with "a5","a5_id",mhbl
@ 10,5 say "House bldg. loan= "
@ 10,25 say mhbl
* calculation of medicine bill
do cal_bill with "a2","a2_id",mmedicine
@ 11,5 say "Medicine bill   = "
@ 11,25 say mmedicine
* calculation of entertainment allowance
do cal_ent with "a7","a7_id",menta
nd=nd+mtrans+mtel+mhbl+mmedicine
@ 7,5 say "Entertainment    ="
@ 7,25 say menta
do cal_da       ; calculation of D.A.
@ 6,5 say "D. A.              ="
```

```
@ 6,25 say mda
ga=ga+menta+mda
do cal_gid
@ 12,40  say "G. I. D.          ="
@ 12,60 say mgid
do cal_bfd
@ 11,40 say "B. F. D.           ="
@ 11,60 say mbfd
do cal_gpa with "a6","a6_id",mgpa
@ 10,40 say "GPA                ="
@ 10,60 say mgpa
nd=nd+mgid+mbfd+mgpa
@ 7,40 say "Session allowance ="
@ 7,60 say msesa
ga=ga+msesa
do case
 case mha=2
 mhrd=mbasic*0.075
 mhr=0.00
 case mha=3
 if mbasic>=6000
  mhr= mbasic*0.35
  mhrd=0.00
 endif
 if mbasic>=4000 .and. mbasic<6000
  mhr=mbasic*0.40
 endif
 if mbasic<4000
  mhr=mbasic*0.50
 endif
 case mha=4
  mhr=mbasic*.5
```

```
  mhrd=200.00
 case mha=1
  mhr=0.00
  mhrd=0.00
endcase
@ 6,40 say "House rent        ="
@ 6,60 say mhr
@ 13,5 say "H R D             ="
@ 13,25 say mhrd
nd=nd+mhrd
do case          ; calculation of charge allowances
   case maddcharge=0
    mheada=0.00
    mdeana=0.00
    mchargea=0.00
   case maddcharge=1
    mchargea=400.00
    @ 13,40 say "Charge allowance  ="
    @ 13,60 say mchargea
    mheada=0.00
    mdeana=0.00
    @ 13,40 say "Charge allowance  ="
    @ 13,60 say mdeana
   case maddcharge=2
    mchargea=600.00
    @ 13,40 say "Charge allowance  ="
    @ 13,60 say mchargea
    mheada=0.00
    mdeana=0.00
   case maddcharge=3
    mheada=450.00
    @ 13,40 say "Head allowance    ="
```

```
    @ 13,60 say mheada
    mchargea=0.00
    mdeana=0.00
   case maddcharge=4
    mdeana=600.00
    @ 13,40 say "Dean allowance    ="
    @ 13,60 say mdeana
    mheada=0
    mchargea=0
endcase
use
ga=ga+mheada+mdeana+mchargea
do case          ; calculation of deductions for club, asso
 case mcatcode=1,2,3
  do common2 with "Oclub",moclub
  do common2 with "Oasso",moasso
  @ 14,40 say "Club              ="
  @ 14,60 say moclub
  @ 13,40 say "Association       ="
  @ 13,60 say moasso
  nd=nd+moclub+moasso
  case mcatcode=4,5
  do common2 with "Oclub",moclub
  do common2 with "Oasso",moasso
  do common2 with "Club3",mclub3
 case mcatcode=6
  do common with mcatcode,"b9",mconveya
  do common2 with "Club3",mclub3
  do common2 with "Asso3",masso3
  do common with "b11",revenue
 case mcatcode=7
  do common with mcatcode,"b9",mconveya
```

```
   do common2 "Club3",mclub3
   do common2 "asso3t",masso3t
   do common with mcatcode,"b11",mrev
   do common with mcatcode,"b5",mwashinga
   do cyclea
  case mcatcode=8
   do common with mcatcode,"b9",mconveya
   do common2 with "club4",mclub4
   do common2 with "asso4",masso4
   do common with mcatcode,"b11",revenue
   do common with mcatcode,"b5",mwashinga
   do cyclea
endcase
na=ga-nd
@ 18,2 say "GROSS AMOUNT="
@ 18,16 say ga
@ 18,40 say "NET DEDUCTION="
@ 18,55 say nd
@ 20,5 say "NET AMOUNT="
@ 20,20 say na
close procedure
res=" "
@ 22,2 say "Observation complete ?[Y]" get res
read
set procedure to sb2        ; pocedure to input these data in
 z=mcatcode                 ; respective database file
 do case
  case z = 1
   do stortsal
  case z<6 .and. z>1
   do storosal
  case z<9 .and. z>5
```

```
   do storesal
endcase
close all
* eop
*************************************************************************
* Procedure For indexing various dbase files
procedure make_ndx
use m1
index on empid to m1_id
use
use m2
index on empid to m2_id
use
use b3
index on brange to b3_brange
use
use a1
index on empid to a1_id
use
use b11
index on gas to b11_gas
use
use b1
index on head to b1_head
use
use a3
index on empid to a3_id
use
use a4
index on empid to a4_id
use
use a5
```

```
index on empid to a5_id
use
use a2
index on empid to a2_id
use
use a7
index on empid to a7_id
use
use b4
index on brange to b4_brange
use
use a6
index on empid to a6_id
use
use b7
index on catcode to b7_ccode
use
use b5
index on catcode to b5_ccode
use
return
*********************************************************
Procedure calculate ; to calculate value of an attribute
parameters basic,pc,max,min,famount
public amount
am1=pc*basic
am1=iif( am1<min,min,am1 )
am1=iif( am1>max,max,am1 )
amount = am1 + fixedamt
use
return
```

```
* Procedure for taking employee's id from keyboard
Procedure input_id
public id,mmonth
id=000000
mmonth="       "
@ 1,9 say "WELCOME TO SALARY BILLING ......."
@ 3,1 say "Please enter employee's id # :" get id picture "999999"
@ 5,1 SAY "Please enter the billing month[e.g Jan/93]:" get mmonth
mmonth=upper(mmonth)
read
temp=id
return
*Procedure for taking record fields into mem. variables
Procedure input_m2
public mbasic,maddcharge,mbrange,mgas,mgpf,mha,msesa,mpa,mpp,;
       mdutystatus,mempid,mcatcode
use m2 index m2_id
locate for empid=id .and. upflag="$"
if eof()
    ?"No such empid # found"
    wait "Press any key to continue with menu:" to ans
    mx=1
else
     mcatcode=catcode
     mempid=empid
     mbasic=basic
     maddcharge=addcharge
     mbrange=brange
     mgas=gas
     mgpf=gpf
     mha=ha
     msesa=sesa
```

```
        mpa=pa
        mpp=pp
        mdutystatus=dutystatus
endif
use
return
*Procedure to calculate bill directly from files's record
Procedure cal_bill
parameters file,ndxfile,v_bill
use &file index &ndxfile
seek(id)
if found()
     v_bill = amt
else
     v_bill = 0
endif
use
return
*Procedure to calculate entertainment allowance
procedure cal_ent
prameters file,ndxfile,v_bill
use &file index &ndxfile
seek(id)
if found()
     if catcode=1 .and. subcat=1
          v_bill=amt
     else
          v_bill=0
     endif
else
   v_bill=0
endif
```

```
use
return
*Procedure to calculate D.A.
procedure cal_da
public mda
use b3 index b3_brange
locate for brange=mbrange
da1=pcbasic1*mbasic
da2=pcbasic2*mbasic
da3=pcbasic3*mbasic
da1=iif( da1>maxvalue1,maxvalue1,da1 )
da1=iif( da1<minvalue1,minvalue1,da1 )
da2=iif( da2>maxvalue2,maxvalue2,da2 )
da2=iif( da2<minvalue2,minvalue2,da2 )
da3=iif( da3>maxvalue3,maxvalue3,da3 )
da3=iif( da3<minvalue3,minvalue3,da3 )
mda=da1+da2+da3+fixedamt
use
return
* GID
procedure cal_gid
use m2 index m2_id
seek(id)
if found()
  tmp_code = catcode
else
  tmp_code = 0
endif
use
use b7 index b7_ccode
seek(tmp_code)
if found()
```

```
  am=pcbasic*mbasic
  am=iif( am>maxvalue,maxvalue,am )
  am=iif( am<minvalue,minvalue,am )
  am=am+fixedamt
  mgid=am
else
  mgid=0
endif
use
return
*B F D
procedure cal_bfd
use b12 index b12_ccod
seek(mcatcode)
bf=pcbasic*mbasic
bf=iif( bf>maxvalue,maxvalue,bf )
bf=iif( bf<minvalue,minvalue,bf )
bf=bf+fixedamt
mbfd=bf
use
return
*GPA
procedure cal_gpa
parameters file,ndxfile,v_bill
use &file index &ndxfile
seek(id)
if found()
   v_bill=floan+sloan+tloan+foloan
else
   v_bill=0
endif
use
```

```
return
*GAS BILL
procedure cal_gas
parameter v_bill
use b11 index b11_gas
locate for gas=mgas
am=pcbasic*mbasic
am=iif( am>maxvalue,maxvalue,am )
am=iif( am<minvalue,minvalue,am )
am=am+fixedamt
v_bill=am
use
return
*MA
procedure cal_med
public mmedical
use b1 index b1_head
locate for head="Medical"
mmedical=fixedamt
v_bil=pcbasic*mbasic
v_bil=iif( v_bil<minvalue,minvalue,v_bil )
v_bil=iif( v_bil>maxvalue,maxvalue,v_bil )
mmedical=v_bil+mmedical
use
return


* COMMON
procedure common
parameters mcatcode,dbname,mvar
use &dbname
index on catcode to &dbname
```

```
seek mcatcode
mvar=pcbasic*mbasic
mvar = iif( mvar<minvalue,minvalue,mvar )
mvar = iif( mvar>maxvalue,maxvalue,mvar )
mvar = mvar + fixedamt
use
return
*
procedure common2
parameters fldstr,mfldstr
use b1 index b1_head
find &fldstr
mfldstr=pcbasic*mbasic
mfldstr= iif( mfldstr<minvalue,minvalue,mfldstr )
mfldstr= iif( mfldstr>maxvalue,maxvalue,mfldstr )
mfldstr = mfldstr + fixedamt
use
return
* CYCLE AD.
procedure cyclea
use a9 index a9_id
seek(id)
if found()
 mcyclea = cyclea
else
 mcyclea = 0
endif
use
return
* Procedure to store calculated salary attributes into esalary
procedure storesal
use esalary
```

```
dele for month="mmonth" .and. empid=mempid
pack
append blank
replace empid with id
replace catcode with mcatcode
replace month with mmonth
replace basic with mbasic
replace da with mda
replace hr with mhr
replace medical with mmedical
replace conveya with mconveya
replace festivala with mfestia
replace washinga with mwashinga
replace sesa with msesa
replace hrd with mhrd
replace elect with melectric
replace gas with mgas
replace trans with mtrans
replace medicine with mmedicine
replace telephone with mtel
replace gpf with mgpf
replace gpa with mgpa
replace hbl with mhbl
replace gid with mgid
replace bfd with mbfd
replace club4 with mclub4
replace asso3 with masso3
replace asso3t with masso3t
replace asso4 with masso4
replace club3 with mclub3
replace cyclea with mcyclea
replace rev with mrev
```

```
replace eida with meida
use
return
* Procedure to store calculated salary attributes into osalary
procedure stortsal
use tsalary
dele for month="mmonth" .and. empid=mempid
pack
append blank
replace empid with id
replace catcode with mcatcode
replace month with mmonth
replace basic with mbasic
replace pp with mpp
replace da with mda
replace hr with mhr
replace medical with mmedical
replace deana with mdeana
replace heada with mheada
replace chargea with mchargea
replace festivala with mfestia
replace sesa with msesa
replace hrd with mhrd
replace elect with melectric
replace gas with mgas
replace trans with mtrans
replace medicine with mmedicine
replace telephone with mtel
replace gpf with mgpf
replace gpa with mgpa
replace hbl with mhbl
replace gid with mgid
```

```
replace bfd with mbfd
replace oclub with moclub
replace oasso with moasso
use
return
*******************************
procedure storosal
use osalary
dele for month="mmonth" .and. empid=mempid
pack
append blank
replace catcode with mcatcode
replace empid with id
replace month with mmonth
replace basic with mbasic
replace da with mda
replace hr with mhr
replace medical with mmedical
replace chargea with mchargea
replace festivala with mfestia
replace sesa with msesa
replace hrd with mhrd
replace elect with melectric
replace gas with mgas
replace trans with mtrans
replace medicine with mmedicine
replace telephone with mtel
replace gpf with mgpf
replace gpa with mgpa
replace hbl with mhbl
replace gid with mgid
replace bfd with mbfd
```

```
replace oclub with moclub
replace oasso with moasso
replace club3 with mclub3
use
return
*************************************************************************
*System: I M S          Module: Salary Management
*Program:MDCS.prg ; Monthly deduction & contribution schedule
*Procedure file: PFMDCS.PRG
close all
public mdc,mmonth,mcatcode
set safety off
set exact on
set PROCEDURE TO pfpm
select a
use m1 index m1_id
select e
use b1 index b1_head
select f
use b9 index b9_ccode
clear
select g
use Jan1993
* Initialization of variables
mdc = 0
mmonth="       "
fname="        "
ans=" "
mcatcode=0
@ 1,2 say "Enter dept code:" get mdc picture "99"
read
do check_dc
```

```
if mn=0
 ?"No such dept. in BUET "
 ?
 do disp_dc
 mdc=00
 mn=0
 @ 3,2 say "Enter valid dept-code:" get mdc PICTURE "99"
 read
endif
do check_dc
clear
if mn=0
text
                        You have failed to enter valid dept-code

                                    Please get out.
endtext
wait
return
endif
clear
@ 2,2 say "Enter month code [Jan/93] :" get mmonth
read
use month index month
locate for mcode="&mmonth"
if eof()
 ?"Illegal entry ; retry"
  wait
  clear
  @ 2,2 say "Enter valid month code [Jan/93]:" get mmonth
  read
endif
```

```
locate for mcode="&mmonth"
if eof()
  ?"Illegal entry ; "
  wait
  return
endif
close procedure
set procedure to pfmdcs
wait "Output to the Printer or Screen [P/S] ?" TO ans
ans=upper(ans)
if ans="P"
  set print on
*    Initialization of printer to CONDENSED MODE
  ? chr(27) + chr(15)
else
  if ans <> "S"
      ?"Illegal entry"
  endif
endif
select h
use ded_con
copy struc to zz.dbf
select c
use a5
select d
use a6
select b
use m2 index m2_id
go top
x= mdc*10000
locate for empid>=x .AND. empid<=(x+9999) .AND. upflag="$"
if eof()
```

```
   ?"No record for department code = ",mdc
else
 clear
 @ 0,0
? "                DEDUCTION AND CONTRIBUTION SCHEDULE"
? "                ----------------------------------"
? " DEPARTMENT CODE    :" ,mdc
? " FOR THE MONTH      :" ,mmonth
?
? "EMPID    BASIC    GPF  GPA  TOTAL   PENSION  HBL   BFD  CLUB
ASSO   REMARK"
cnt=0
do while .not. eof()
  if empid>=x .AND. empid<=(x+9999) .AND. upflag="$"
      mempid = empid
      mbasic=basic
      mgpf=gpf
      mcatcode=catcode
      select C
      locate for empid=mempid .AND. flag="$"
      mhbl=amt
      select D
      locate for empid=mempid .AND.  flag="$"
      mgpa=flamt+slamt+tlamt+folamt
      mtotal=mgpa+mgpf
      mpension=mbasic*0.1
      select E
      if mcatcode>=1 .AND. mcatcode<=5
         m1head="oclub"
         m2head="oasso"
      endif
     do case
```

```
        case  mcatcode=6
              m1head="club3
              m2head="asso3"
        case mcatcode=7
              m1head="club3"
              m2head="asso3t"
        case mcatcode=8
              m1head="club4"
              m2head="asso4"
     endcase
     mclub=0
     masso=0
     seek m1head
     do pfmdcs with mclub
     seek m2head
     do pfmdcs with masso
     mbfd=0
     select F
     locate for catcode=mcatcode
     do pfmdcs with mbfd
     select h
     append blank
     replace empid with mempid,basic with mbasic,gpf with mgpf,;
          gpa with mgpa,total with mtotal,pension with mpension,;
          hbl with mhbl,bfd with mbfd,club with mclub,;
          asso with masso
     ?
     ?EMPID,"  ",BASIC,GPF,GPA,TOTAL,"    ",PENSION,HBL,BFD,"
",CLUB,ASSO
   cnt=cnt+1
   endif
      select B
```

```
        skip
        if cnt>6
           cnt=0
           wait
        endif
     enddo
  endif
  close all
  close procedure
  res=" "
  ?
  wait "Observation complete ? [Y] :"
  res=upper(res)
  if ans="p"
  * turn CONDENSED MODE off
   chr(27) + chr(18)
   set print off
  endif
  *eop
```

```
*CSR.PRG; Program for consolidated salary report
*System I M S; Module: S M
close all
public mst,mst_t,msao,msao_t,mseo,mseo_t,mstc,mstc_t,msfc,;
       mmsfc_t,mbt,mbao,mbeo,mbtc,mbfc,mmonth,
clear
* Initialization of variables
mst=00000000
mseo=0000000
msao=0000000
mstc=0000000
msfc=0000000
mst_t=00000000
mseo_t=0000000
msao_t=0000000
mstc_t=0000000
msfc_t=0000000
mbt=000000000
mbao=00000000
mbeo=00000000
mbtc=00000000
mbfc=00000000
use budget index budget
go top
* calculation of budgets for various heads
do while .not. eof()
do case
   case head=201
    mbao=amt
   case head=202
    mbt=amt
   case head=203
```

```
      mbeo=amt
     case head=204
      mbtc=amt
     case head=205
      mbfc=amt
endcase
skip
enddo
accept "Enter current month [Jan/93]:" to mmonth
use tsalary index tsal_id
go top
* calculation of total salary for teachers
do while .not. eof()
if month=mmonth
  mst=mst+basic
endif
skip
enddo
use osalary index osal_id
go top
* calculation of total salary for officers
do while .not. eof()
if month=mmonth
do case
   case catcode=2
        msao=msao+basic
   case catcode=4
        msao=msao+basic
   case catcode=3
        mseo=mseo+basic
   case catcode=5
        mseo=mseo+basic
```

```
endcase
skip
enddo
use esalary index esal_id
go top
* calculation of total salary for employees
do while .not. eof()
if month=mmonth
do case
   case catcode=6
   mstc=mstc+basic
   case catcode=7
   mstc=mstc+basic
   case catcode=8
   msfc=msfc+basic
endcase
endif
skip
enddo
clear
* Report heading
?"  CONSOLIDATED SALARY REPORT FOR THE MONTH  ",mmonth
?
?"HEAD    HEAD-NAME          BUDGET        PAID IN    PAID UPTO "
?"                                         THE MONTH  THE MONTH"
?
?"201  Salary of ad.officers",mbao,msao,msao_t
?"202  Salary of teachers   ",mbt,mst,mst_t
?"203  Salary of ed.officers",mbeo,mseo,mseo_t
?"204  Salary of 3rd class  ",mbtc,mstc,mstc_t
?"205  Salary of 4th class  ",mbfc,msfc,msfc_t
?
```

```
res=" "
wait "Observation Complete ? [Y]:" to res
res=upper(res)
close all
* eop
*************************************************************
* SYSTEM : I M S ; Module : S M
* Program-name : MSS ; for monthly salary statements
public mc,mdc,mmonth
clear
close all
mdc=0
mmonth="       "
clear
@ 2,2 say "Enter dept-code for salary statements" get mdc PICTURE
"99"
read
@ 4,2 say "Enter the month-code [Jan/93] :" get mmonth
read
clear
TEXT
Enter your choice:
Salary statement for 1. Teachers
                     2. Officers
                     3. Employees
                     0. Return
ENDTEXT
mc=0
@ 14,2 say "Select:" get  mc PICTURE "9"
read
do case
   case mc = 0
```

```
   wait
endif
enddo
res=" "
wait "Observation complete ? [Y] " to res
res=upper(res)
close all
* eop
**************************
```

```
if cnt>6
   cnt=0
   wait
endif
enddo
res=" "
wait "Observation complete ? [Y] " to res
res=upper(res)
close all
*eop
************************************************************
* SYSTEM : I M S ; MOdule : S M
* program-name : msso.prg ; program for monthly salary statement
* for officers
clear
set talk off
use osalary
x=mdc*10000
?" MONTHLY SALARY STATEMENT FOR OFFICERS ( PAYS AND ALLOWANCES )"
?
?" FOR THE MONTH : ",mmonth
?"ID-NO   BASIC PP MEDICAL DA H.RENT  CHARGEA SESSION"
cnt=0
do while .not. eof()
if empid>x .and. empid<= (x+9999) .and. month=mmonth
?
?empid,basic,pp,medical," ",da,hr,"  ",chargea,"  ",sesa
cnt=cnt+1
endif
skip
if cnt>6
   cnt=0
```

```
endif
skip
if cnt>6
   cnt=0
   wait
endif
enddo
res=" "
wait "Observation complete ? [Y] " to res
res=upper(res)
close all
* eop
*********************************************************
* SYSTEM : I M S ; MOdule : S M
* program-name : msse.prg ; program for monthly salary statement
* for employees
clear
set talk off
use esalary
x=mdc*10000
?" MONTHLY SALARY STATEMENT FOR EMPLOYEES( PAYS AND ALLOWANCES )"
?
?" FOR THE MONTH : ",mmonth
?"ID-NO   BASIC PP MEDICAL DA H.RENT  CHARGEA SESSION"
cnt=0
do while .not. eof()
if empid>x .and. empid<= (x+9999) .and. month=mmonth
?
?empid,basic,pp,medical," ",da,hr," ",chargea,"  ",sesa
cnt=cnt+1
endif
skip
```

```
          clear
          return
     case mc = 1
          clear
          do msst.prg
     case mc = 2
          clear
          do msso.prg
     case mc = 3
          clear
          do msse.prg
endcase
* eop
************************************************************
* SYSTEM : I M S ; MOdule : S M
* program-name : msst.prg ; program for monthly salary statement
* for teachers
clear
set talk off
use tsalary
x=mdc*10000
?" MONTHLY SALARY STATEMENT FOR TEACHERS ( PAYS AND ALLOWANCES )"
?
?" FOR THE MONTH : ",mmonth
?"ID-NO   BASIC PP MEDICAL DA H.RENT  DEANA  HEADA CHARGEA SESSION"
cnt=0
do while .not. eof()
if empid>x .and. empid<= (x+9999) .and. month=mmonth
?
?empid,basic,pp,medical," ",da,hr," ",deana," ",heada,"
",chargea,"  ",sesa
cnt=cnt+1
```

```
* DE.PRG ;Program for data entry
* System : I M S ; Module : Sys. Maintenance
* Declaration of variable
public mfname
clear
close all
?"                WELCOME TO I M S: DATA ENTRY "
?
?"Please enter any one db. file from the list presented below:"
?
?"  [For more information, consult with DATABASE DICTIONARY]"
TEXT


     B1   B2   B3   B4   B5   B6   B7   B8   B9   B10   B11   B12

     A1   A2   A3   A4   A5   A6   A7   A9   M1   M2    E3.   EXP

     BUDGET     VOUCHER     DEXP
ENDTEXT
mfname="          "
@ 20,2 SAY "Enter file-name     : " get mfname
READ
use &mfname           ; to open desired database file
set status on
append                ; data entry in append mode
* End of program for data entry
*************************************************************
* CDB.PRG ;Program for modification of database structure
* System : I M S ; Module : Sys. Maintenance
* Declaration of variable
public mfname
```

```
clear
close all
?"       WELCOME TO I M S: MODIFICATION OF DATABASE STRUCTURE"
?
?" Please enter any one db. file from the list presented below:"
?
?"    [For more information, consult with DATABASE DICTIONARY]"
TEXT


      B1   B2   B3   B4   B5   B6   B7   B8   B9   B10    B11   B12

      A1   A2   A3   A4   A5   A6   A7   A9   M1   M2     E3    EXP

    BUDGET     VOUCHER     DEXP    TSALARY    ESALARY   OSALARY
ENDTEXT
mfname="          "
@ 20,2 SAY "Enter file-name     : " get mfname
READ
use &mfname           ; to open desired database file
set status on
modi struc
* End of program for modification of database structure
* CSC.PRG ;Program for change of security number
* System : I M S ; Module : Sys. Maintenance
clear
close all
?"       WELCOME TO I M S: SYSTEM MAINTENANCE"
use pass
edit
index on id to pass
close all
```

```
* INDX.PRG ;Program for indexing data base file
* System : I M S ; Module : Sys. Maintenance
* Declaration of variables
public mfname,mfld,tfname
clear
close all
?"            WELCOME TO I M S: INDEXING OF DATABASE FILE"
?
?" Please enter any one db. file from the list presented below:"
?
?"  [ For more information, consult with DATABASE DICTIONARY ]"
TEXT


    B1   B2   B3   B4   B5   B6   B7   B8   B9   B10    B11   B12

    A1   A2   A3   A4   A5   A6   A7   A9   M1   M2     E3    EXP

    BUDGET     VOUCHER     DEXP
ENDTEXT
* Initialization of variables
mfname="         "
mfld="         "
tfname="         "
@ 20,2 SAY "Enter file-name     : " get mfname
READ
use &mfname           ; to open desired database file
set status on
disp struc            ; to display structure of db. file
res=" "
wait "Observation complete ? [Y] " to res
res=upper(res)
```

```
clear
@ 2,2 say "Enter target-field name: " get mfld
read
@ 4,2 say "Enter target-file name: " get tfname
read
index on &mfld to &tfname

* DEDIT.PRG ;Program for data edit
* System : I M S ; Module : Sys. Maintenance
* Declation of variables
public mfname,mfld,mvalue
clear
close all
?"                WELCOME TO I M S: DATA-EDIT "
?
?" Please enter any one db. file from the list presented below:"
?
?"   [For more information, consult with DATABASE DICTIONARY]"
TEXT


     B1   B2   B3   B4   B5   B6   B7   B8   B9   B10    B11   B12

     A1   A2   A3   A4   A5   A6   A7   A9   M1   M2     E3    EXP

     BUDGET     VOUCHER     DEXP
ENDTEXT
mfname="          "
mfld="          "
mvalue="          "
@ 20,2 SAY "Enter file-name     : " get mfname
READ
```

```
use &mfname             ; to open selected file
set status on
disp struc              ; to display structure of the db. file
res=" "
wait "Observation complete ? [Y] " to res
res=upper(res)
clear
@ 2,2 say "Enter the name of target-field :" get mfld
read
@ 4,2 say "Enter the value of target-field :" get mvalue
read
locate for &mfld=&mvalue
if eof()
 ?"No such value of target-field is found ; retry "
 wait
 return
endif
edit
close all
* eop
* DIR.PRG ; program for daily income report
* SYSTEM : I M S ; Module : I M
set talk off
set status off
clear
* Declation of variables
public ma21,ma22,ma23,ma41,ma42,ma43,ma44,ma45,ma46;
       mheadno,mheadname,mflag,mdate
close all
set PROCEDURE TO PFDIR
MDATE=DATE()
@ 6,1 say "Enter Date:" get mdate PICTURE  "!"
```

```
READ
* Initialization of variables


mheadno=000
mheadname="XXXXXXXXXXXXXXXXXXXX"
mflag="$"
ma21=0000000
ma22=0000000
ma23=0000000
ma41=0000000
ma42=0000000
ma43=0000000
ma44=0000000
ma45=0000000
ma46=0000000
close all
@ 13,12 say "Please wait ! processing is going on....."
do pfdir with ma21,21
DO pfdir with ma22,22
Do pfdir with ma23,23
Do pfdir with ma41,41
Do pfdir with ma42,42
Do pfdir with ma43,43
Do pfdir with ma44,44
do pfdir with ma45,45
do pfdir with ma46,46
wait "Output to Printer or Screen [P/S] ?" to  mm
mm=upper(mm)
close procedure
set procedure to pfapdir
use dir
```

```
delete for flag="$"
pack
if ma21 > 0
   do pfapdir with 21,"Tution fee",ma21
endif
if ma22>0
    do pfapdir with 22,"Admission fee",ma22
endif
if ma23>0
   do pfapdir with 23,"Registration fee",ma23
endif
if ma41>0
   do pfapdir with 41,"Examination fee",ma41
endif
if ma42>0
   do pfapdir with 42,"Mark sheet fee",ma42
endif
if ma43>0
   do pfapdir with 43,"Certificate fee",ma43
endif
if ma44>0
   do pfapdir with 44,"Question fee",ma44
endif
if ma45>0
   do pfapdir with 45,"Paper_sale",ma45
endif
if ma46>0
   do pfapdir with 46,"Admission form",ma46
endif
close PROCEDURE
if mm="P"
   set print on
```

```
    ? chr(27) + chr(15)
 else
   if mm <> "S"
     ?"Illegal entry"
   endif
 endif
 clear
 ? "         I M S: INCOME MANAGEMENT"
 ?
 ?  "        DAILY INCOME STATEMMENT"
 ?  "       ========================="
 ?
 ? "FOR THE DATE: ",MDATE
 ? "HEAD NO.      NAME OF HEAD                              AMOUNT"
 ? "-------       ------------------                        --------"
 go top
 DO WHILE .NOT. eof()
   ?headno,"          ",headname,"          ",amt
   skip
 enddo
 if mm="P"
   ? chr(27) + chr(18)
   set print off
 endif
 res=" "
 ?
 wait "Observation complete ? [Y] :" to res
 res=upper(res)
 CLOSE ALL
 * End of program for daily income report
 ************************************************************
 * mir.prg ; program for periodical income report
```

```
* system : I M S ; module : I M
set talk off
set status off
clear
* Declaration of variables
public ma21,ma22,ma23,ma41,ma42,ma43,ma44,ma45,ma46;
       mheadno,mheadname,mflag,mmonth,sdate,edate
close all
set PROCEDURE TO PFMIR
sdate=DATE()
edate=DATE()
@ 6,1 say "Enter starting date:" get sdate PICTURE "!"
@ 8,1 say "Enter ending date:" get edate PICTURE "!"
READ
* Initialization of variables
mheadno=000
mheadname="XXXXXXXXXXXXXXXXXXXX"
mflag="$"
ma21=0000000
ma22=0000000
ma23=0000000
ma41=0000000
ma42=0000000
ma43=0000000
ma44=0000000
ma45=0000000
ma46=0000000
close all
@ 13,12 say "Please wait ! processing is going on....."
do pfmir with ma21,21
DO pfmir with ma22,22
Do pfmir with ma23,23
```

```
Do pfmir with ma41,41
Do pfmir with ma42,42
Do pfmir with ma43,43
Do pfmir with ma44,44
Do pfmir with ma45,45
Do pfmir with ma46,46
wait "Output to Printer or Screen [P/S] ?" to  mm
mm=upper(mm)
close procedure
set procedure to pfapdir
use dir
delete for flag="$"
pack
if ma21 > 0
   do pfapdir with 21,"tution fee",ma21
endif
if ma22>0
    do pfapdir with 22,"admission fee",ma22
endif
if ma23>0
   do pfapdir with 23,"registration fee",ma23
endif
if ma41>0
   do pfapdir with 41,"examination fee",ma41
endif
if ma42>0
   do pfapdir with 42,"mark sheet fee",ma42
endif
if ma43>0
   do pfapdir with 43,"certificate fee",ma43
endif
if ma44>0
```

```
   do pfapdir with 44,"question fee",ma44
endif
if ma45>0
   do pfapdir with 45,"paper_sale",ma45
endif
if ma46>0
   do pfapdir with 46,"admission form",ma46
endif
close PROCEDURE
if mm="P"
   set print on
   ? chr(27) + chr(15)
else
  if mm <> "S"
    ?"Illegal entry"
  endid
endif
clear
? "         I M S: INCOME MANAGEMRNT OF B U E T DHAKA"
?
?  "              PERIODICAL INCOME STATEMENT"
?  "              ==========================="
?
? "FOR THE PERIOD: ",sdate," TO ",edate
?
? "HEAD NO.      NAME OF HEAD                              AMOUNT"
? "-------       ------------------                        --------"
go top
DO WHILE .NOT. eof()
  ?headno,"         ",headname,"          ",amt
  skip
enddo
```

```
?
res=" "
wait "Observation complete ? [Y]:" to res
res=upper(res)
CLOSE ALL
* End of program for periodical income report
*****************************************************************
PROCEDURE PFAPDIR
PARAMETERS HNO,HNA,AM
mheadno=HNO
mheadname=HNA
mflag="$"
append blank
replace headno with mheadno
replace headname with mheadname
replace amt with AM
replace flag with mflag
return
*****************************************************************
procedure pfmir
parameters X,N
use voucher
index on headno to voucher
go top
do while .not. eof()
   if date<=edate .and. date>=sdate .and. headno=N
      X=X+amt
   endif
   skip
enddo
close all
erase voucher.ndx
```

```
return
*************************************************************
PROCEDURE PFDIR
PARAMETER X,N
use voucher
index on headno to voucher.ndx
go top
do while .not. eof()
  if date = mdate .and. headno=N
     X = X + amt
  else
  endif
  skip
enddo
close all
erase voucher.ndx
return
procedure pfmir
PARAMETERS X,N
use voucher
index on headno to voucher
go top
do while .not. eof()
   if date >= sdate .and. date <= edate .and. headno = N
      X = X + amt
    endif
    skip
enddo
close all
erase voucher.ndx
return
*************************************************************
```

```
procedure pfapdir
PARAMETERS HNO,HNA,AM
mheadno=HNO
mheadname=HNA
mflag = "$"
append blank
replace headno with mheadno
replace headname with mheadname
replace amt with AM
return
* End of procedure file for income management
*****************************************************************
* HER.PRG ; Program for Headwise Expenditure Report
* System: I M S; Module: NSHM
set talk off
clear
* Declaration of variables
public ma401,ma402,ma403,ma404,ma405,ma411,ma412,ma413,ma423,;
       ma424,ma425,mheadno,mheadname,mflag,mmonth
close all
set procedure to pfnshm
mmonth="xxxxxx"
@ 2,5 say "Please enter month-code:[Jan/93] " get mmonth
read
mheadno=0
mheadname="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
ma401=00000000
ma402=00000000
ma403=00000000
ma404=00000000
ma405=00000000
ma411=00000000
```

```
ma412=00000000
ma413=00000000
ma423=00000000
ma424=00000000
ma425=00000000
@ 13,12 say "Please wait ! processing is going on ....."
do pfher with ma401,401
do pfher with ma402,402
do pfher with ma403,402
do pfher with ma404,404
do pfher with ma405,405
do pfher with ma411,411
do pfher with ma412,412
do pfher with ma413,413
do pfher with ma423,423
do pfher with ma424,424
do pfher with ma425,425
wait "Output to the Printer or Screen [P/S] ?" to mm
mm=upper(mm)
use her.dbf
delete for flag="$"
pack
if ma401>0
  do pfapher with 401,"Municipal tax",ma401
endif
if ma402>0
  do pfapher with 402,"WASA-tax",ma402
endif
if ma403>0
  do pfapher with 403,"Electricity cost",ma403
endif
if ma404>0
```

```
  do pfapher with 404,"Gas cost",ma404
endif
if ma405>0
  do pfapher with 405,"Development tax",405
endif
if ma411>0
  do pfapher with 411,"Fuel cost",411
endif
if ma412>0
  do pfapher with 412,"Repair & maintenance",412
endif
if ma413>0
  do pfapher with 413,"Purchase of transport",413
endif
if ma423>0
  do pfapher with 423,"Repair of office-machines",423
endif
if ma424>0
  do pfapher with 424,"Maintenance of computers",424
endif
if ma425>0
  do pfapher with 425,"Purchase of office-machines",425
endif
close procedure
if mm="P"
  set print on
  ? chr(27) + chr(15)
else
  if mm<> "S"
    ?"Illegal entry"
  endif
endif
```

```
clear
?"        I M S : NON-SALARY HEADS MANAGEMENTS"
?
? "      Monthly Headwise Expenditure Statement"
?
? " For the month :",mmonth
?
? "Head No. Name of Head                              Amount"
? "-------  ----------------------                  ---------"
go top
do while .not. eof()
  ?headno,"    ",headname,"     ",amt
  skip
enddo
res=" "
?
wait "Observation complete ? [Y] :"
res=upper(res)
if mm="P"
  ? chr(27) + chr(18)
  set print off
endif
close all
*****************************************************************
* DER.PRG ; Program for department-wise Expenditure Report
* System: I M S; Module: NSHM
set talk off
clear
public ma306,ma307,ma308,ma309,ma310,ma453,ma454,ma455,;
       ma456,ma458,mheadno,mheadname,mflag,mmonth
close all
set procedure to pfnshm
```

```
mmonth="xxxxxx"
@ 2,5 say "Please enter month-code:[Jan/93] " get mmonth
read
mheadno=0
mheadname="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
ma306=00000000
ma307=00000000
ma308=00000000
ma309=00000000
ma310=00000000
ma453=00000000
ma454=00000000
ma455=00000000
ma456=00000000
ma458=00000000
@ 13,12 say "Please wait ! processing is going on ....."
do pfher with ma306,306
do pfher with ma307,307
do pfher with ma308,308
do pfher with ma309,309
do pfher with ma310,310
do pfher with ma453,453
do pfher with ma454,454
do pfher with ma455,455
do pfher with ma456,456
do pfher with ma458,458
wait "Output to the Printer or Screen [P/S] ?" to mm
mm=upper(mm)
use her.dbf
delete for flag="$"
pack
if ma306>0
```

```
  do pfapher with 306,"Entertainment",ma306
endif
if ma307>0
  do pfapher with 307,"Travel allounce",ma307
endif
if ma308>0
  do pfapher with 308,"Telephone cost",ma308
endif
if ma309>0
  do pfapher with 309,"Newspapers & Journals",ma309
endif
if ma310>0
  do pfapher with 310,"Miscellaneous",310
endif
if ma453>0
  do pfapher with 453,"Study-tour/Training",453
endif
if ma454>0
  do pfapher with 454,"Educational instruments",454
endif
if ma455>0
  do pfapher with 455,"Student's projects",455
endif
if ma456>0
  do pfapher with 456,"Seminar/Conference/Deligation",456
endif
if ma458>0
  do pfapher with 458,"Post graduate research",458
endif
close procedure
if mm="P"
  set print on
```

```
  ? chr(27) + chr(15)
else
  if mm<> "S"
    ?"Illegal entry"
  endif
endif
clear
?"       I M S : NON-SALARY HEADS MANAGEMENTS"
?
? "      Monthly Depertment-wise Expenditure Statement"
?
? " For the Month :",mmonth
?
? "Head No. Name of Head                                Amount"
? "-------  ----------------------                    ---------"
use her
go top
do while .not. eof()
  ?headno,"    ",headname,"     ",amt
  skip
enddo
res=" "
?
wait "Observation complete ? [Y] :"
res=upper(res)
if mm="P"
  ? chr(27) + chr(18)
  set print off
endif
close all
*****************************************************************
```

```
procedure pfher
parameters X,N
erase exp.ndx
use exp.dbf
index on headno to exp.ndx
go top
do while .not. eof()
    if month=mmonth .and. headno=N
     X=X+amt
    endif
    skip
enddo
close all
return
*******************************
procedure pfapher
parameters HNO,HNA,AM
mheadno=HNO
mheadname=HNA
mflag="$"
append blank
replace headno with mheadno, headname with mheadname
replace amt with AM, flag with mflag
return
* eop
*************************************************************
* SYSTEM : I M S ; Module : NSHM
* Program-name : MAS.PRG ; for monthly adjustment statement
set talk off
set status off
set safe off
clear
```

```
public mmonth,sdate,edate
close all
mmonth="      "
@ 6,1 say "Enter Month-code[Jan/93]:" get mmonth
READ
use month index month
locate for mcode = "&mmonth"
if eof()
 ?"Illegal entry ; retry"
 wait
 return
endif
sdate=DATE()
edate=DATE()
sdate=d1
edate=d2
select a
use refund
index on hno to refund
select b
use head
index on hno to head
select a
mm=" "
join with b to zz for hno = b->hno fields hno,date,hname,amt
wait "Output to the printer or screen [P/S] ?" to mm
mm=upper(mm)
if mm="P"
   set print on
   ? chr(27) + chr(15)
endif
use zz
```

```
clear
?"BANGLADESH UNIVERSITY OF ENGG. AND TECHNOLOGY, DHAKA"
?
?"MONTHLY ADJUSTMENT STATEMENT FOR THE MONTH : ",mmonth
?
?"Head-no  Head-name                                Refund amount"
cnt=0
go top
do while .not. eof()
  if date > sdate .and. date < edate
  ?
  cnt=cnt+1
  ? HNO,"    ",hname,"    ",amt
  endif
  skip
  if cnt>7
     cnt=0
     wait
  endif
enddo
if mm="P"
   ? chr(27) + chr(18)
   set print off
else
  if mm <> "S"
    ?"Illegal entry"
  endif
endif
res=" "
?
wait "Observation complete ? [Y] :" to res
res=upper(res)
CLOSE ALL
```

REFERENCES

[1] Gio-Wiederhold, *"Database design"* International Student Edition McGRAW HILL International book Co. 1983.

[2] James Martin, *"Computer Database Organization"* Prentice Hall of India Pvt. Ltd. 2nd Edition, 1986.

[3] John L. Pfaliz, *"Computer data structure"* International Student Edition, McGRAW-HILL Kogakosha ltd. 1984.

[4] Carl Townsend, *" Mastering dBASEIII Plus"* SYBEX Tech. Publication, Singapore, 1985.

[5] Meiliuar Page Jones, *"Practical Guide To Structured System Design"* Yourdon Press, England, 1980.

[6] Gadwell D.J.L., *"State of the Art Report: Databse 2nd Generation"* Series 10 No.7 , Pergmon Infotech Ltd. 1983

[7] Codd E. F., *"Courent Computer Science Symposia"* Volume 6, 1972.

[8] Samet P.A., *"Query Language: Aunifield approach"* Heyden and Sons Ltd.

[9] Elbra & Tony, *"Database for the small Computer Users"* N C C Publications, 1982.

[10] Ullman & D. Jaffrey, *"Principles of Database Systems"* Galgotia Publications, New Delhi, 1985.

[11] Mayne & Alan, *"Database Management Systems"* Technical review, National Computing Centre, England, 1981.

[12] J. A. Vasta, *"Understanding Database Management Systems"* Wadsworth Publishing Co. Belmount, California,1989.

[13] L. S. Orillia, *"Computers and Informations"* Third Edition, McGrAW Hill Book Co., New York, 1988.

[14] W. S. Davis, *"Information Processing Systems"* 2nd Edition, Addition-Wesley Publishing Co., California, 1988.

[15] S. L. Mandell, *"Computers & Information Processing"* 4th Edition, West Publishing Co., Los Angeles, 1988.

[16] John Avitabile, *"Computers & the Information Society"* McGRAW Hill Book Co., New York, 1988.

[17] DR. Mustafa A. Kamal, *"Systems analysis, design & development with structured approach"* Dept. of Computer Information Systems, Central Missouri State University, U.S.A.,1987.

[18] T.J.Teorey, D.Yang, and J.P.Fry , *"Logical design methodology for Relational Databases"* Computing Surveys, Vol.18, No.2, June, 1986.

[19] Mohammad Nur-Ul-Alam,*"Design of a Relational Database for a wide varities of microcomputer chips"* M.Sc.Engg.(CSE) Thesis, Dept. of CSE, B U E T, Dhaka, 1988.

[20] Monzor Murshed Kabir, *"Database System Design for resourse management in an industry"* P.G.D.(CSE) Project, Dept. of C S E, B U E T, Dhaka, 1991.