UNIT 4 FILE AND DATABASE DESIGN

Structure

4.0 INTRODUCTION

After designing the input and output, the designer begins to pay his attention on the work of file designing or how data should be organised around user requirement. How data are organised depends on the data and response requirements that determine hardware configurations. System analyst is responsible for designing the files and selecting their contents, selecting from options available for organising the data. File organisation may be sequential, index sequential, inverted list or random. Each method has its own uses and abuses.

An integrated approach to file design is the database. The general theme is to handle information as an integrated whole, with a minimum of redundancy and improved performance. Various software techniques are applied to manipulate, describe and manage data. Irrespective of type of data structure used, the main objectives of database are accuracy and integrity, successful recovery from failure, privacy and security of data.

4.1 OBJECTIVES

After going through this unit, you will be in a position to learn:

- file concept and its different type
- various methods of selecting data storage medium
  file Organisation
- file design
  database design
  coding system

4.2 SELECTING DATA STORAGE MEDIA

In theory, there is a wide selection of devices and media available for file storage, ranging from punched cards to high speed internal memory. In practice, system analyst is interested with bulk storage devices using magnetic mediasuch as magnetic tape and exchangeable magnetic discs. Magnetic drums and fixed discs are also very helpful but these are generally used for special purpose files. Magnetic card devices, although provide large storage capacity, are slow and not so much reliable. Punched cards and paper tape have become obsolete now-a-days and are not considered for bulk storage of data. Magnetic tape units generally used reels holding plastic tape 0.5 inch wide and 2400 ft. in length. It has magnetizable coating on one side. For reading or writing purpose, the tape is transported from one spool to another. It is not possible at all to read from and immediately write on to one reel of tape. Therefore, when we update a file, it becomes necessary to read the existing file from one reel and to create an entirely new version by writing on the another reel. When dealing with a master file updated from time to time, the file being read is said to be the brought-forward file and the newly created one is the carried-forward file (because the former has been brought forward from a previous updating process and the letter will be carried-forward to a future one). Apart from this, these different versions of master file referred to as 'generations' are very helpful for recovery purposes in case the latest versions get corrupted. Three generations (grandfather, father, son) are commonly created and retained before re-using the oldest version.

The magnetable surface of the tape is densely packed with spots of magnetism measured in bits per inch (bpi) along the length of tape. They are arranged in rows across the tape-commonly either seven or nine spots per mw-each spot representing a bit. These arrangements are also known as seven or nine track tape. Each row or frame represents a character or byte. Data is recorded on the tape in terms of blocks and size of the block is specified.

4.2.1 File Concepts

Files are the heart of a computer application. Before constructing files, we must understand the basic terms used to describe the file hierarchy. The most commonly used terms are data item, record, file and database.

Data Item: A basic or individual element of data is called data item. Each data item is identified by a name and is assigned a value. For example, in a payroll system, Employee Name, Employee Identification Number are the data items assigned the values PANKAJ and 1775 respectively. Data item is sometimes referred to as a field.

Record; The collection of related data items is called a record. For example, in a payroll system, each employee record may have eight separate fields, which are related to print a pay cheque, such as Employee Name, Employee Code, Sex, Designation, Basic Pay, HRA, DA, and Deductions. The analyst also determines the length and type of each field while designing the record. For example, layout of an employeerecord in a payroll system can be as follows:

Name of Data item	Type	Length	Decimal
NAME	C	20	
CODE	N	5.	
SEX	C	1	
DESIGNATION	C	15	
BASIC	N	8	2
HRA	N	6	2
DA	N	6	2
Deduction	N	6	2

It is necessary to distinguish one specific record from another. System analyst select one data item in the record that is Likely to be unique in all the records of a file which is used to identify the record for further processing. This item is called the key field or record key.

File: File is a collection of related records. Each record in a file is included because it pertains to the same entity. In the payroll system, employee file will contain only employee details records and the inventory or stock records will not be kept in the employee file because these are not related to employee details.

Database: The highest level in the hierarchy is the database. It is a set of inter-related files for real time processing. It contains the necessary data for problem solving and can be used for several users who are accessing data concurrently.

4.3 TYPES OF FILE

There are various types of files in which the records are collected and maintained. They are categorised as:

- Master file
- Transaction file
- Table file
- Report file
- Back-up file
- Archival file
- Dump file
- Library file

4.3.1 Master

Master files are the most important type of file. Most file design activities concentrate here, In a business application, these are considered to be very significant because they contain the essential records for maintenance of the organisation's business. A master file can be further categorised. It may be called as a reference master file, in which the records are static or unlikely to change frequently. For example, a product file containing descriptions and codes; a customer file containing name, address and account number are example of reference files. Alternatively, it may be described as a dynamic master file. In this file, we keep records which are frequently changed (updated) as a result of transactions or other events. These two types of master file may be kept as separate files or may be combined, for example, a sales ledger file containing reference data, such as name, address, account number, together with current transaction and balance outstanding for each customer.

4.3.2 Transaction

A transaction is a temporary file used for two purposes. First of all, it is used to accumulate data about events as they occur. Secondly, it helps in updating master files to reflect the result of current transactions. The term transaction refers to any business event that affects the organisation and about which data is captured Examples of common transactions in the organisation are making purchases, hiring of workers and recording of sales.

4.3.3 Table

A special type of master file is included in many systems to meet specific requirements where data must be referenced repeatedly. Table files are permanent files containing reference data used in processing transactions, updating master file or producing output. As the name implies, these files store reference data in tabular form. Table files conserve memory space and make the program maintenance easier by storing data in a file, that otherwise would be included in programs or master file records.

4.3.4 Report

Report files are collected contents of individual output reports or documents produced by the system. They are created by the system where many reports are produced by the system but printer may not be available for all the reports. This situation frequently arises when the computer carry out three functions - input, processing and output simultaneously, rather than executing each function in sequence. In this case, the computer writes the report contents to a file on a magnetic tape or disk, where it remains until it can be printed. That file is called the report file which contains the unprinted output data. The process of creating it is known as spooling which means that output that cannot be printed when it is produced is spooled into a report file. Then, depending on the availability of printer, the system will be instructed to read the report file and print the output on the printer.

4.3.5 Backup

It is a copy of master, transaction or table file that is made to ensure a copy is available if anything happens to the original.

4.3.6 Archival

These files are copes made for long term storage of data that may be required at a much later date. Usually, archival files are stored far away from the computer centre so that they cannot be easily retrieved for use.

4.3.7 Dump

This is a copy of computer-held data at a particular point of time. This may be a copy of master file to be retained to help recovery in the event of a possible corruption of the master file or it may be part of a program in which error is being traced.

4.3.8 Library

Library file generally contains application programs, utility programs and system software packages.

4.4 FILE ORGANISATION

A file is organised to ensure that records are available for processing. Before a file is created, the application to which the file will be used must be carefully examined. Clearly, a fundamental considemtion in this examination will concern the data to be recorded on the file. But an equally important and less obvious consideration concerns how the data are to be placed on the file.

4.4.1 Sequential

It is the simplest method to store and retrieve data from a file. Sequential organisation simply means storing and sorting in physical on tape or disk. In a sequential organisation a records can be added only at the end of the file. That is in a sequential file, records are stored one after the other without concern for the actual value of the data in the records. It is not possible to insert a record in the middle of the file without re-writing the file. In a sequential file update, transaction records are in the same sequence as in the master file. Records from both files are matched, one record at a time, resulting in an updated master file.

It is a characteristic of sequential files that all records are stored by position; the first one is at the first position, the second one occupies the second position and third is at third and so on. There are no addresses or location assignments in sequential files.

To read a sequential file, the system always starts at the beginning of the file. If the record sought is somewhere in the file, the system reads its ways upto it, one record at a time. For example, if a particular record happens to be the fifteenth one in a file, the system starts at the first one and reads ahead one record at a time until the fifteenth one is reached. It cannot jump directly to the fifteenth one in a sequential file without starting from the beginning.

Using the key field, in a sequential file the records have been arranged into ascending or descending order according to a key field. This key field may be numeric, alphabetic, or a combination of both, but it must be occupy the same place in each record, as it forms the basis for determining the order in which the records will appear on the file,

When we start searching for a particular record in a sequential file the system do not use the physical record key. The system assigns the value of the particular record key as a search key. For example, let a sequential file consists of the records of employee number from 1200 to 1250. Then how to locate or retrieve a record for an employee number 1234? Here employee number 1234 is the search key. When searching for the employee number 1234, the program controls all the processing steps that follows. The first record is read and its employee number is compared with a search key. 1200 versus 1234. Since they do not match, the process is repeated. The employee number for the next record is 1201, and it also does not match the search key. Therefore, the process of reading and comparing records continues until the employee number match the search key. If the file does not contain the employee number 1234, the read and compare process will continue until the end of file is reached. Sequential files are generally maintained a magnetic tape, disk or a mass storage system. The advantages and disadvantages of the Sequential File organisation are compared and given below:

Advantages	Disadvantages
Simple to understand this approach	Entire file must be processed even when the activity rate is low.
Locating a record requires only the record key.	Transactions must be sorted and placed in sequence prior to processing
Efficient and economical if the activity rate is high	Timeliness of data in file deteriorates while batches are being accumulated
Relatively inexpensive I/O media and devices may be used.	
Files may be relatively easy to reconstruct since a good measure of built in backup is usually available.	Data redundancy is typically high since the same data may be stored in several files sequenced on different keys.

4.4.2 Random or Direct

For a proposed system, when the sequential files are assumed as a disadvantage, another file organisation called Direct organisation is used. As with a sequential file, each record in a direct file must contain a key field. However the records need not appear on the file in key field sequence: In addition any record stored on a direct file can be accessed if its location or address is known. All previous records need not to be accessed. The problem, however is to determine how to store the data records so that, given the key field of the desired record, its storage location on the file can be determined. In other words, if the program knows the record key, it can determine the location address of a record and retrieve it independently of any other records in the file.

It would be ideal if the key field could also be the location of the record on the file. This method is known as direct addressing method. This is quite simple method but the requirements af this method often prevents its use. Because of many other factors, this method could not become popular. Hence it is rarely used.

Therefore, before a direct organised file can be created, a formula or method must be devised to convert the key field value for a record to the address or location of the record on the file. This formula or method is generally called an algorithm. Otherwise called the Hashing addressing. Hashing refers to the process of deriving a storage address from a record key. There are many algorithms to determine the storage location using key field. Some of the algorithms are:

Division by Prime: In this procedure, the actual key is divided by any prime number. Here the modular division is used. That is quotient is discarded and the storage location is signified by the remainder. If the key field consists of large number of digits, for instance, 10 digits (e.g. 2345632278) then strip off the first or last 4 digits and then apply the division by prime method.

For example, the key field is 2345632278 strip off first 4 digits. Then the new key is 632278. Divide the new key by a prime number. Let it be 41. The quotient is 15421, remainder is 17. Hence 17 is the storage address.

Various other common algorithms are also given as under

- Folding
- Extraction
- Squaring

The advantages and disadvantages of direct file organisation are as follows:

Advantages	Disadvantages
Immediate access to records for inquiry and updating purposes is possible.	Records in the on-line file may be exposed the risk of a loss of accuracy and a procedure for special backup and reconstruction is required.
Immediate updating of several files as a result of single transaction is possible.	As compared to sequentially organised, this may be less efficient in using the storage space.
Time taken for sorting the transactions can be saved.	Adding and deleting of records is more difficult then with sequential files.
	Relatively expensive hardware and software esources are required.

4.4.3 Indexed

The third way of accessing records stored in the system is through an index. The basic form of an index includes a record key and the storage address for a record. To find a record, when the storage address is unknown it is necessary to scan the records. However, if an index is used, the search will be faster since it takes less time to search an index than an entire file of data.

Indexed file offers the simplicity of sequential file while at the same time offering a capability for direct access. The records must be initially stored on the file in sequential order according to a key field. In addition, as the records are being recorded on the file, one or more indexes are established by the system to associate the key field value(s) with the storage location of the record on the fie. These indexes are then used by the system to allow a record to be directly accessed

To find a specific record when the file is stored under an indexed organisation, the index is searched first to find the key of the record wanted. When it is found, the corresponding storage address is noted and then the program can access the record directly. This method uses a sequential scan of the index, followed by direct access to the appropriate record The index helps to speed up the search compared with a sequential file, but it is slower than the direct addressing.

The indexed files are generally maintained on magnetic disk or on a mass storage system. The primary differences between direct and indexed organised files are as follows:

Records may be accessed from a direct organised file only randomly, where as records may be accessed sequentially or randomly from an indexed organised files.

Direct organised files utilise an algorithm to determine the location of a record, whereas indexed organised files utilize an index to locate a record to be randomly accessed. The advantages and disadvantages of indexed sequential file organisation are as follows:

Advantages	Disadvantages
Permits the efficient and economical use of sequential processing techniques when the activity rate is high.	Less efficient in the use of storage space than some other alternatives.
Permits quick access to records in a relatively efficient way this activity is a small fraction of the total workload.	Access to records may be slower using indexes than when when transform algorithms are used. Relatively expensive hardware and software resources are required.

4.5 FILE DESIGN:

The basic factors to be considered in the selection of file media and file organisation method are:

- the method of processing for updating files,
- size of the file
- file inquiry capabilities
- activity ratio of records in the file
- file volatility and
- the response time.

Each of these factors is briefly described in the following paragraphs.

In batch processing, the sequential method of processing, using magnetic tapes, is employed. This method of updating involves re-creation of the master file every time the file is updated. In order to reduce the set-up time, the updating of the file is'done after accumulation of a fairly large batch of transactions. Fig.1 illustrates the updating procedure.

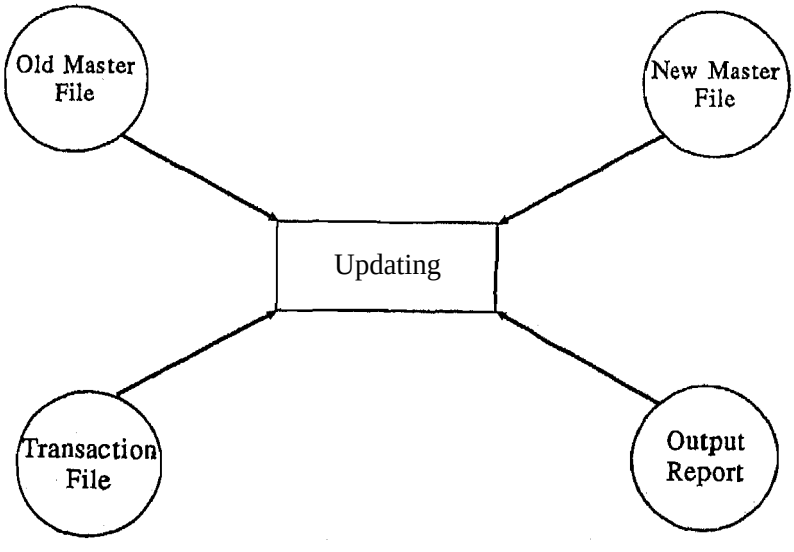

Fig.1: Updating File in Batch Processing

The method of random processing is used to update the files as the transactions occur. This is a widely used method for on- line processing of files through remote terminals. Fig.2 illustrates this method.

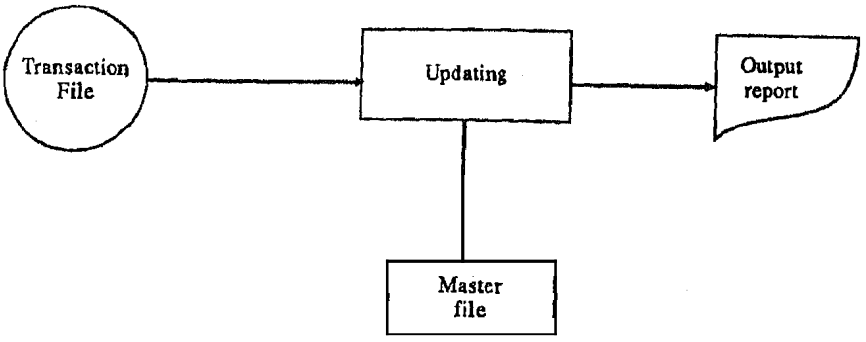

Fig.2: Updating Direct Access File

On-line processing allows for dispersing input/output terminals throughout the organisation, so that various users can have access to the files. This can also be used in a batch processing mode where several files can be updated simultaneously, as illustrated in Fig.3. While the new open order file is created from a batched file of current orders, the customer file and the product file are accessed simultaneously and updated.

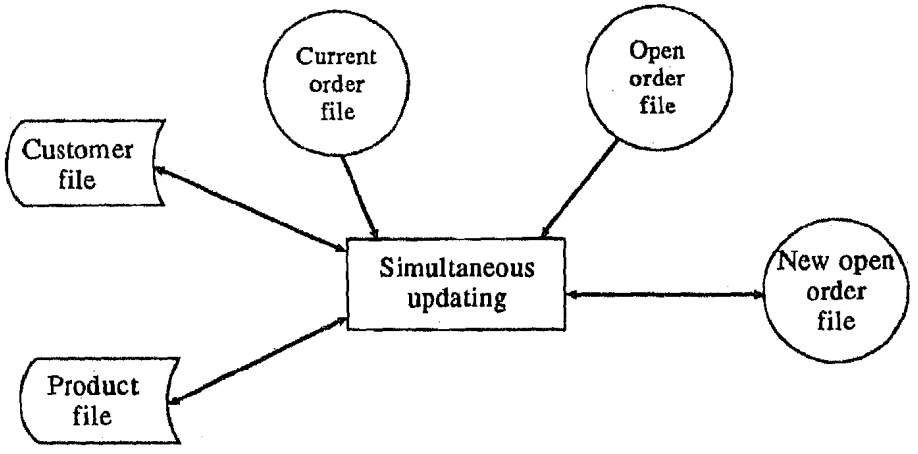

Fig 3: Simultaneous Updating of Multiple Files

Magnetic tape and magnetic disk can accommodate large files. Magnetic tape is more convenient and economical for holding large files, particularly when the records are processed sequentially in a batch mode. The magnetic disk can also hold large volumes of data, but it is relatively expensive. However, it is really suited for on-line processing. In designing the files, the growth potential should be taken into account.

The inquiry capabilities relate to the ease of referring to a specific record in file without any alteration. Direct Access Storage Devices (DASD) can quickly handle file inquiries. A teletype or CRT is used to enter the inquiry specifying the record and the information desired. After the record in the file is accessed, the desired information is communicated by the CPU to the CRT or the teletypewriter. All these steps are completed within a few seconds after entering the inquiry. This inquiry capabilities were severely restricted before the advent of DASD. The inquiry capability of the DASD can be advantageously used without interrupting the normal processing operations.

The time interval between entering an inquiry and getting the reply is called response time. Some applications require a fast response as in the case of airline or hotel reservations, stock quotations in stock exchange. etc. The most suitable medium for such purposes is the DASD.

The activity ratio of a file is an important feature in file design. It is a measure of the proportion of records processed in an updating run. A file with high activity ratio can be processed more economically using the sequential processing method. If more than thirty per cent of the records in a file are used for updating, the activity ratio is considered high. Random or direct accessing is more suitable for updating files whose activity ratio is low, i.e., less than thirty per cent.

Another characteristic of a file is its volatility, which indicates the additions, deletions and changes to the file. When a file is accessed frequently, as in banks, stock exchanges, airlines, etc. in a working period, the file is regarded as highly volatile.

Before designing the file, the analyst should consider specific aspects relating to a file. These aspects should be recorded on a work sheet as shown below:

File Work Sheet

File Name : Analyst

1. File update: Batch ☐ Direct ☐
2. File Organisation: Sequential ☐
   Indexed Sequential ☐
   Direct ☐
3. Processing frequency: On demand ☐
   Daily ☐
   Weekly ☐
   Monthly ☐
   Any other ☐
4. Activity ratio: Low ☐ Moderate ☐ High ☐
5. Direct access: Yes ☐ No ☐ Occasional ☐
6. Volatility: Low ☐ Moderate ☐ High ☐
7. Record Characteristics:
   Type: Fixed ☐ Variable ☐
   Blocking Factor
   Number of Characters
8. File Dynamics:
   (i) Number of records
   (ii) Yearly additions
   (iii) Yearly deletions
   (iv) Percent growth
9. File media: Tape ☐ Disk ☐
10. Sources of data:
11. *Type* of information required and reported:

The above file work sheet should be accompanied by detailed record layouts, which describe in detail the layout of the records in the file.

Check Your Progress 1

1. Explain the file concept briefly.

.....

.....

.....

.....

.....

2. List out various types of file.

.....

.....

.....

.....

.....

3. List out the three methods of organizing a file.

.....

.....

.....

.....

.....

4. List out the basic factors responsible for selecting the appropriate media and file organisation methods.

.....

.....

.....

.....

.....

4.6 DATABASE DESIGN

As we have discussed that the organization selected for a particular file mainly depends on the nature of the application for which it will be used. Historically, files have been designed based on specific application. Payroll files are created containing all the data pertinent Lo a company's payroll system. Similarly, individual files are created for use with the company's personnel, accounts receivable, inventory, and other systems. If the data contained on these files are not carefully delineated, it is very likely that the same data will appear on several of these files. In other words, these files would contain redundant data. For example, both a company's personnel file and payroll file could contain the name and address of each employee. This would mean that a simple change of address would have to be processed twice and possibly three or four times, depending on the number of other files on which these data appear. Clearly, it would be more practical to have each employee's name and address on one file from which it can be accessed by all programs requiring these data. This would reduce the amount of redundant data and minimise the possibility that data contained on a file might be inaccurate because they were never updated. This is but one of the reasons that database technology was developed.

A DATABASE can be thought of as a set of logically related files organised to facilitate access by one or more applications programs and to minimise data redundancy. In other words, a database can be defined as a stored collection of data, organised on the basis of relationships in the data rather than the convenience of storage structures. It is not a replacement for files.

Some general objectives in establishing a database are as follows:

- Eliminate redundant data as much as possible.
- Integrate existing data files.

- Share data among all users.
- Incorporate changes easily and quickly.
- Simplify the use of data files.
- Lower the cost of storing and retrieving data.
- Improve accuracy and consistency.
- Provide data security from unauthorised use.
- Exercise central control over standards.

In addition to the database itself, a set of programs is necessary to facilitate adding new data as well as modifying and retrieving existing data within a database. This set of programs is referred to as a Data Base Management System (DBMS). A data base system merge data into one pool shared by all systems so that any change automatically affects all relevant systems. The following figures defines the difference between the traditional file systems and database management system.

Fig.4 shows the Traditional file systems in which each system is responsible for its own data.

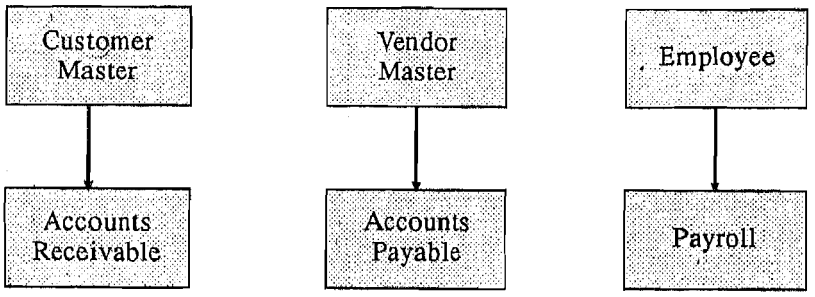

Fig.4: Traditional File System

Fig. 5 shows the Data Base Management Systems in which data is centralised.

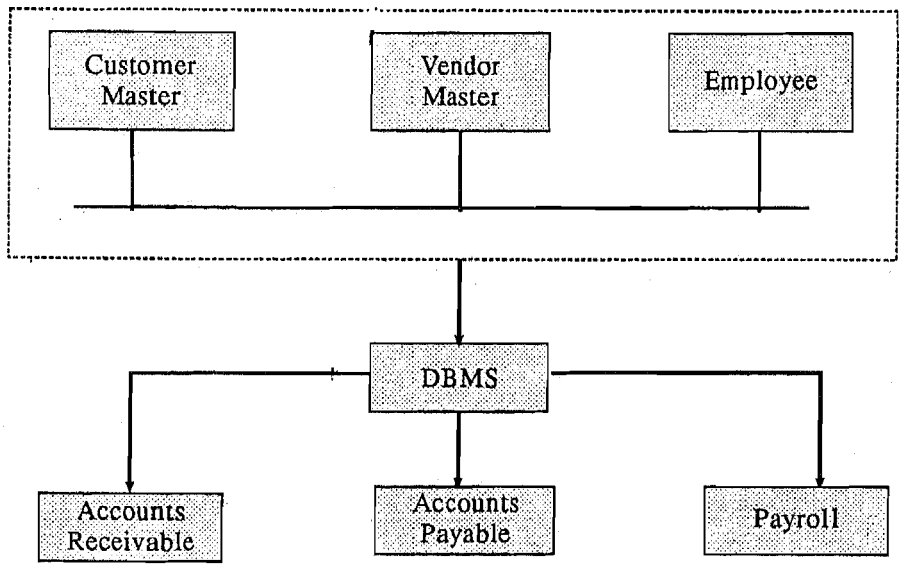

Fig.5: Data Base Management System

Specific advantages of data base are:

1. **File Consolidation:** Pooling data reduces redundancy and inconsistency and promotes cooperation among different users. Since data bases link records together logically, a data change in one system will cascade through all the other system using the data.

2. **Program and** file independence: This feature separates the definition of the files from their programs, allowing a programmer to concentrate on the logic of the program instead of precisely how to store and retrieve data.

3. **Access Versatility:** Users can remeve data in many ways. They enjoy the best of both worlds - sequential access for reporting data in a prescribed order and random access for rapid retrieval of a specific record.

4. **Data Security:** Usually a DBMS includes a password system that controls access to sensitive data. By limiting their access to read-only, write-only, or specified records, or even fields in records, passwords can prevent certain users from retrieving unauthorised data.

5. **Program** Development. Programmers must use standard names for data items rather than invent their own from program to program. This allows the programmer to focus on desired function,

6. Program Maintenance: Changes and repairs to a system are relatively easy.

7. Special Information: Special-purposereport generators can produce reports with minimum effort.

4.6.1 Logical and Physical view of Data

In database design, several views of data must be considered along with the persons who use them. In addition to data structuring, where relationships are reflected between and within entities, we need to identify the application program's logical views of data within an overall Logical data structure. The logical view is what the data look like, regardless of how they are stored. The physical view is the way data exist in physical storage. It deals with how data are stored, accessed, or related to other data in storage. There are four views of data out of which three are logical and one is physical. The logical views are the user's view, the programmer's view and the overall logical view, called a schema.

4.6.2 SCHEMA

Once a database system has been designed, it will be possible to identify each type of data item, data aggregate, record and set by a name or code. It will be possible to state which data item types go together to make data aggregate types and record types, and to identify which record types are members and owners of set types. A coded set of tables describing this information and stored in the computer system on direct access devices is called a SCHEMA. It is a description of the data structure which is separate from the data itself. The schema describes the areas, their identifiers and page sizes, and indicates how these are related to the records and sets. In other systems, a different set of tables is used for this.

The schema therefore, is the view of the data, the overall logical data structure which is held by the DBMS. Each time a program requires data, the DBMS will look up in the schema for the details of the structure of the data requested. For example if the program requires an occurrence of a set, the DBMS will look up in the schema which record types are required, how to find the relevant records given a certain key by the program, and perhaps also which areas the pages containing the relevant data are stored in.

4.6.3 Sub-schema

In a database system, it is not always possible to allow programmers to write the data division of their choice for reasons of security or control. It is more useful to provide the programmer with a standard description of the logical data to be used in a particular application. All references to data within the program will be for this description, which is called a SUBSCHEMA and is similar to the Schema in structure. The DBMS has the job of matching data requests on a subschema and data requests based on the schema:

4.7 TYPES OF DATABASE

In conventional file systems, groups of bytes constitute a field, one or more fields make a record, and two or more records make a file. In a database environment, a group of bytes constitutes a data item or segment, a collection of segments a data entry, and a series of data entries a data set. The complete collection of data sets is the database itself. With traditional processing of files, records are not automatically related, so a programmer must be concerned with record relationships. Often the files are stored and processed by record key, just as we sorted the transaction file. Data bases relate data sets in one of three models: hierarchical, network, or relational.

4.7.1 Hierarchical Model

In a hierarchical structure, sometimes referred to as a tree structure, the stored data get more and more detailed as one branches further and further out on the tree, Each segment, or node, may be subdivided into two or more subordinate nodes, which can be further subdivided into two or more additional nodes. However, each node can have only one "parent" from which it emanates. The Fig.6 shows the hierarchical structure.

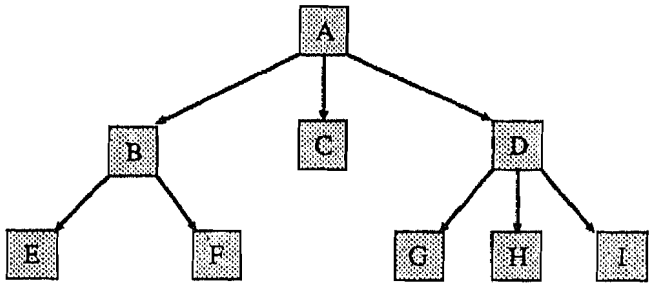

Fig.6: Hierarchical Structure

4.7.2 Network Model

The network related data sets are similar to hierarchical ones, except that a node may have more than one parent. Thus a hierarchical DBMS is a subset of network DBMS. The trade off between the simplicity of design of a hierarchical structure and the storage efficiency of a network structure is a very important consideration in database implementation. The Fig.7 shows the Network structure.

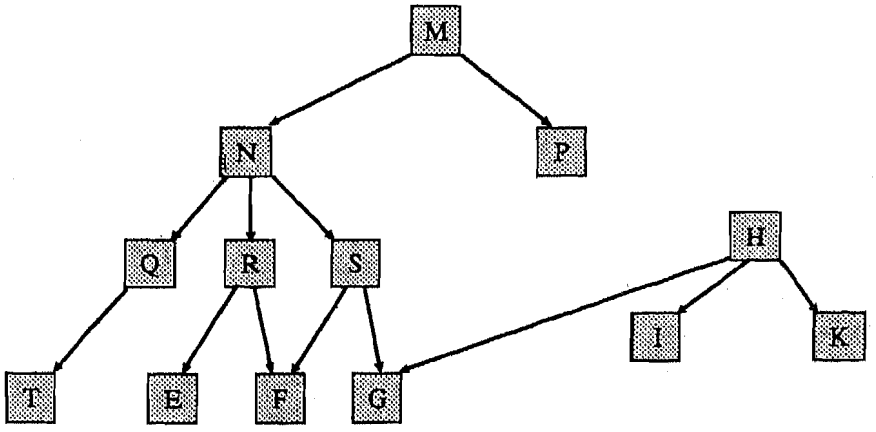

Fig.7: Network Structure

4.7.3 Relational Model

The relational structure, however, organises the data in terms of two dimensional tables. That is, Relational data sets order data in a table of rows and columns and differ markedly from their hierarchical or network counterparts. There are no parent or node data sets as shown in Fig.8. In a relational database management systems, we have the same concept of files, records, and fields. Files are represented by two-dimensional tables, each of which is called a "relation". Records, which can be visualized as rows in the table, are called "Luples". Fields can be visualised as columns, and are called by attribute names, or domains.

For example, note that in the supplier table in Fig.8 we have three luples, or rows, and three attribute names or columns. If we need to know the name of the supplier of blue chairs, the relational DBMS searches the type and color columns of the Furniture, Table and finds supplier number 30, and then it scans the supplier table for number 30, which turns out to be PANKAJ'S. Since each "record" is a row in the table and each "field" a column, an inventory system of 1600 Luples, each with 5 attributes, would create a table of 1600 rows and 5 columns.

FURNITURE

Product Number	Type	Color	Quantity in stock	Supplier Number
2589	Table	White	4	26
2892	Chair	Blue	6	30
3471	Chair	Eight Green	20	133
3678	Desk	Brown	9	150
3689	Stool	Brown	25	159

SUPPLIER

Supplier Number	Supplier Name	Amount of Purchases This year
30	PANKAJ	26,035.00
26	DINESH	13,960.00
159	RAJESH	75,286.00

Fig.8: Relational Structure

A relational DBMS can perform the following basic operations:

- create or delete tables
- update, insert, or delete rows
- add or delete columns
- copy data from one table into another
- retrieve or query a table, row or column
- print, recognize, or read a table or row
- join or combine tables based on a value in a table.

Since the relational structure organises the data in terms of two dimensional tables, they offer great flexibility and a high degree of data security. The relational structure uses relatively little memory or secondary storage. Unfortunately, the process of creating the tables is a rather elaborate procedure. Another disadvantage of this structure is that it generally requires more time to access information than either of the other two structures. This is because much more information must be searched in order to answer queries posed to the systems. In addition, some implementation use a fixed amount of storage for each field, resulting in insufficient storage utilisation. In spite of these disadvantages. the relational structure has gained rapid acceptance and is currently the most popular of the three structures. Many experts predict that it will eventually replace the others completely,

4.8 CODING SYSTEM

The information systems are designed with space, time, and cost savings in mind. The coding systems are used to reduce the input, control errors and speed up the entire process. So coding systems are methods in which conditions, words, ideas or relationships are expressed by a code. A code is an ordered collection of symbols designed to provide unique identification of an entity or attribute. It may be a brief number, title or symbol. The main purpose of codes is to facilitate the identification and retrieval of items of information from the system.

4.9 TYPES OF CODE

There are many possible coding structures. The main types of codes are described below:

4.9.1 Classification Code

Classification is the best described as the establishment of categories of entities, types and attributes in a way that brings like or similar items together according to pre-determined relationships. A classification is by nature an ordered systematic structure. So the

classification code places separate entities Bike events, people, or objects, into distinct groups called classes. A code is used to identify one class from another. Using this code the user classifies the event into one of several possible categories and records the code. Classification codes vastly simplify the input process because only a single-digit code is required. The need for writing lengthy descriptions or making judgements is eliminated.

4.9.2 Function Code'

Function codes state the activities or work to be performed without spelling out all of the details in narrative statements. Analysts use this type of code frequently in transaction data to tell the system how to process the data. For example, the design for file processing the codes given like the following to perform certain activities.

A — To add record
D — To delete record
U — To update record
E — To edit record

otherwise,

1 — Addition of records
2 — Modification of records
3 — Deletion of records

4.9.3 Card Code

Card codes allow the program to distinguish between the type of card and to determine whether the contents of a specific card are correct. These card codes are used in the punched cards only which is used for batch process.

4.9.4 *Sequence* Code

Sequence codes are numbers or letters assigned in series. They tell the order in which events have occurred. This code is simple to use and apply, For example, employee numbers might be assigned consecutively to employees as they hired. It makes no provision for classifying groups of like items according to specific characteristics. An advantage of the sequence code is that it can cover an unlimited number of items by using the fewest possible code digits. As new items occur they are simply assigned to the next higher unused number in.

4.9.5 Significant-digit *Subset Code*

A well conceived coding scheme, using subcodes within larger codes or numbers, can provide a wealth of information to users. Suppose item number will be assigned to the different materials and products a firm stocks or sells. One way is to assign numbers in sequence, starting with the first and going through to the last one. Or a prefix can be added to the identification nurnbers to further describe the type of item. For example:

PL — refers the product is Plastic and

ST — refers the product is steel.

The codes can be divided into subsets or subcodes, or characters that are part of the identification number that have special meaning. The subcodes tell the user additional information about the item. For example, in a bank examination, the registration number is assigned to each candidate which gives many informations,

Let the assigned numbet is 11021978

11 — centre where the examination is to be held

02 — The post number for which the candidate has applied.

1978 — The number assigned to the candidate.

A frequent method of coding is by abbreviation of the name of an entity or attribute. The main ways of doing this are by mnemonic codes and acronyms.

4.9.6 Mnemonic Code

Mnemonic code construction is characterised by the use of either letters or numbers, or letters and numbers combinations, which describe the items coded, the combinations having been derived from descriptions of the items themselves. Mnemonic codes produce fewer errors than other types of code where the number of items is relatively small and stable. For example, M and F are more reliable for Male and Female than 1 and 2, and Y and N for Yes and No respectively. Another example for combination of letters and numbers is, to describe 16 inch color television set, a use code is **TV-CL-16.** Also the unit of measure codes are frequently mnemonic codes. For example, cm -centimeter, m - meter, km - kilometer

4.9.7 Acronym

The acronym is a particular type of mnemonic representation formed from the first letter or letters of several words. An acronym often becomes a word in itself. For example,

MCA — Master of Computer Applications

RADAR — Radio Detecting and Ranging

Check Your Progress 2

1. List out the specific advantages of database.

.....

.....

.....

.....

.....

2. List out various types of database structure.

.....

.....

.....

.....

.....

3. What do you understand by Coding System?

.....

.....

.....

.....

.....

4. List out various types of code.

.....

.....

.....

.....

.....

4.10 SUMMARY

In this unit, we have discussed file concept and several categories of data files. Three most important methods of file organisation have also been explained in a systematic way. Basic factors to be considered in the selection of file media and file organisation have been pointed out in this unit Database design, its various types are also included. Coding design also play significant role in business data processing system. Different types of code have been defined.

4.11 MODEL ANSWERS

Check Your Progress 1

1. File is a collection of related records. Each record in a file pertains to the same entity.

2. Various types of files are:
   (i) Master (ii) Transaction (iii) Table
   (iv) Report (v) Back-up (vi) Archival
   (vii) Library (viii) Dump

3. Three methods are:
   (i) Sequential (ii) Random (iii) Indexed

4. The basic factors to be considered in the selection of file media and file organisation method are:
   (i) Method of processing for updating files
   (ii) Size of the file
   (iii) File inquiry capabilities
   (vi) Activity ratio of records in the file
   (v) File volatility
   (vi) Response time

Check Your Progress 2

1. Specific advantages of database are:
   (i) File consolidation.
   (ii) Program and file independence
   (iii) Access versatility
   (iv) Data security
   (v) Program development
   (vi) Easier way of program maintenance

2. (i) Hierarchical (ii) Network (iii) Relational

3. The coding systems are used to reduce the input, control errors and speed up the entire process. Coding systems are considered as methods in which conditions, words, ideas or relationships are expressed by a code. A code is an ordered collection of symbols designed to provide unique identification of an entity or attribute. It may be a brief number, title or symbol.

4. Various types of code are:
   (i) Classification code (ii) Function code
   (iii) Card code (iv) Sequence code
   (v) Significant digit subset code (vi) Mnemonic code
   (vii) Acronym