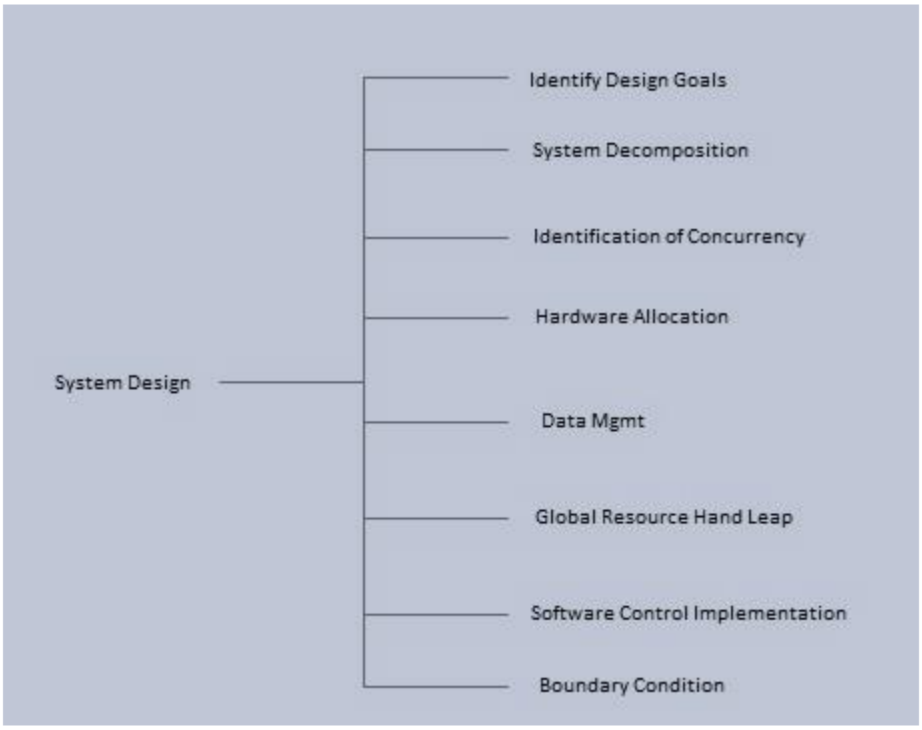
System design is the phase that bridges the gap between problem domain and the existing system in a manageable way. This phase focuses on the solution domain, i.e., *“how to implement?”*

It is the phase where the SRS document is converted into a format that can be implemented and decides how the system will operate.

In this phase, the complex activity of system development is divided into several smaller sub-activities, which coordinate with each other to achieve the main objective of system development.



Inputs to System Design

System design takes the following inputs −

- Statement of work
- Requirement determination plan
- Current situation analysis
- Proposed system requirements including a conceptual data model, modified DFDs, and Metadata (data about data).

Outputs for System Design

System design gives the following outputs −

- Infrastructure and organizational changes for the proposed system.
- A data schema, often a relational schema.
- Metadata to define the tables/files and columns/data-items.
- A function hierarchy diagram or web page map that graphically describes the program structure.
- Actual or pseudocode for each module in the program.
- A prototype for the proposed system.

Types of System Design

Logical Design

Logical design pertains to an abstract representation of the data flow, inputs, and outputs of the system. It describes the inputs (sources), outputs (destinations), databases (data stores), procedures (data flows) all in a format that meets the user requirements.

While preparing the logical design of a system, the system analyst specifies the user needs at level of detail that virtually determines the information flow into and out of the system and the required data sources. Data flow diagram, E-R diagram modeling are used.

Physical Design

Physical design relates to the actual input and output processes of the system. It focuses on how data is entered into a system, verified, processed, and displayed as output.

It produces the working system by defining the design specification that specifies exactly what the candidate system does. It is concerned with user interface design, process design, and data design.

It consists of the following steps −

- Specifying the input/output media, designing the database, and specifying backup procedures.
- Planning system implementation.
- Devising a test and implementation plan, and specifying any new hardware and software.

- Updating costs, benefits, conversion dates, and system constraints.

Architectural Design

It is also known as high level design that focuses on the design of system architecture. It describes the structure and behavior of the system. It defines the structure and relationship between various modules of system development process.

Detailed Design

It follows Architectural design and focuses on development of each module.

Conceptual Data Modeling

It is representation of organizational data which includes all the major entities and relationship. System analysts develop a conceptual data model for the current system that supports the scope and requirement for the proposed system.

The main aim of conceptual data modeling is to capture as much meaning of data as possible. Most organization today use conceptual data modeling using E-R model which uses special notation to represent as much meaning about data as possible.

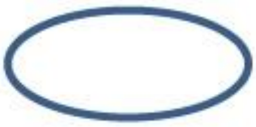
Entity Relationship Model

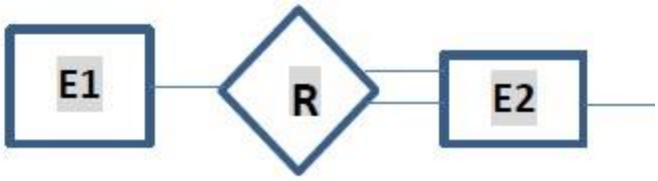
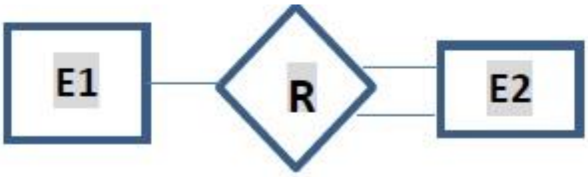
It is a technique used in database design that helps describe the relationship between various entities of an organization.

Terms used in E-R model

- **ENTITY** − It specifies distinct real world items in an application. For example: vendor, item, student, course, teachers, etc.
- **RELATIONSHIP** − They are the meaningful dependencies between entities. For example, vendor supplies items, teacher teaches courses, then supplies and course are relationship.
- **ATTRIBUTES** − It specifies the properties of relationships. For example, vendor code, student name. Symbols used in E-R model and their respective meanings −

The following table shows the symbols used in E-R model and their significance −

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Identity Relationship
	Attributes
	Key Attributes
	Multivalued
	Composite Attribute

	Derived Attributes
	Total Participation of E2 in R
	Cardinality Ratio 1:N for E1:E2 in R

Three types of relationships can exist between two sets of data: one-to-one, one-to-many, and many-to-many.

File Organization

It describes how records are stored within a file.

There are four file organization methods −

- **Serial** − Records are stored in chronological order (in order as they are input or occur). **Examples** − Recording of telephone charges, ATM transactions, Telephone queues.
- **Sequential** − Records are stored in order based on a key field which contains a value that uniquely identifies a record. **Examples** − Phone directories.
- **Direct (relative)** − Each record is stored based on a physical address or location on the device. Address is calculated from the value stored in the record’s key field. Randomizing routine or hashing algorithm does the conversion.
- **Indexed** − Records can be processed both sequentially and non-sequentially using indexes.

Comparision

	Serial	Sequential	Direct	Index
Type of Access	Batch	Batch	Online	Batch or Online
Data Organization	Serial	Sequentially by key value	No particular order	Sequentially and by index
Flexibility in handling inquiries	No	No	Yes	Yes
Availability of up to date Data	No	No	Yes	Yes
Speed Retrieval	Slow	Slow	Very Fast	Fast
Activity	High	High	Low	High
Volatility	Low	Low	High	High
Example	ATM Transition Queue	Payroll process script billing operation	Online reservation and banking transaction	Customer ordering and billing

File Access

One can access a file using either Sequential Access or Random Access. File Access methods allow computer programs read or write records in a file.

Sequential Access

Every record on the file is processed starting with the first record until End of File (EOF) is reached. It is efficient when a large number of the records on the file need to be accessed at any given time. Data stored on a tape (sequential access) can be accessed only sequentially.

Direct (Random) Access

Records are located by knowing their physical locations or addresses on the device rather than their positions relative to other records. Data stored on a CD device (direct-access) can be accessed either sequentially or randomly.

Types of Files used in an Organization System

Following are the types of files used in an organization system −

- **Master file −** It contains the current information for a system. For example, customer file, student file, telephone directory.
- **Table file −** It is a type of master file that changes infrequently and stored in a tabular format. For example, storing Zipcode.

- **Transaction file** − It contains the day-to-day information generated from business activities. It is used to update or process the master file. For example, Addresses of the employees.
- **Temporary file** − It is created and used whenever needed by a system.
- **Mirror file** − They are the exact duplicates of other files. Help minimize the risk of downtime in cases when the original becomes unusable. They must be modified each time the original file is changed.
- **Log files** − They contain copies of master and transaction records in order to chronicle any changes that are made to the master file. It facilitates auditing and provides mechanism for recovery in case of system failure.
- **Archive files** − Backup files that contain historical versions of other files.

Documentation Control

Documentation is a process of recording the information for any reference or operational purpose. It helps users, managers, and IT staff, who require it. It is important that prepared document must be updated on regular basis to trace the progress of the system easily.

After the implementation of system if the system is working improperly, then documentation helps the administrator to understand the flow of data in the system to correct the flaws and get the system working.

Programmers or systems analysts usually create program and system documentation. Systems analysts usually are responsible for preparing documentation to help users learn the system. In large companies, a technical support team that includes technical writers might assist in the preparation of user documentation and training materials.

Advantages

- It can reduce system downtime, cut costs, and speed up maintenance tasks.
- It provides the clear description of formal flow of present system and helps to understand the type of input data and how the output can be produced.
- It provides effective and efficient way of communication between technical and nontechnical users about system.
- It facilitates the training of new user so that he can easily understand the flow of system.
- It helps the user to solve the problems such as troubleshooting and helps the manager to take better final decisions of the organization system.
- It provides better control to the internal or external working of the system.

Types of Documentations

When it comes to System Design, there are following four main documentations −

- Program documentation
- System documentation
- Operations documentation
- User documentation

Program Documentation

- It describes inputs, outputs, and processing logic for all the program modules.
- The program documentation process starts in the system analysis phase and continues during implementation.
- This documentation guides programmers, who construct modules that are well supported by internal and external comments and descriptions that can be understood and maintained easily.

Operations Documentation

Operations documentation contains all the information needed for processing and distributing online and printed output. Operations documentation should be clear, concise, and available online if possible.

It includes the following information −

- Program, systems analyst, programmer, and system identification.
- Scheduling information for printed output, such as report, execution frequency, and deadlines.
- Input files, their source, output files, and their destinations.
- E-mail and report distribution lists.
- Special forms required, including online forms.
- Error and informational messages to operators and restart procedures.
- Special instructions, such as security requirements.

User Documentation

It includes instructions and information to the users who will interact with the system. For example, user manuals, help guides, and tutorials. User documentation is valuable in training users and for reference purpose. It must be clear, understandable, and readily accessible to users at all levels.

The users, system owners, analysts, and programmers, all put combined efforts to develop a user's guide.

A user documentation should include −

- A system overview that clearly describes all major system features, capabilities, and limitations.
- Description of source document content, preparation, processing, and, samples.
- Overview of menu and data entry screen options, contents, and processing instructions.
- Examples of reports that are produced regularly or available at the user's request, including samples.
- Security and audit trail information.
- Explanation of responsibility for specific input, output, or processing requirements.
- Procedures for requesting changes and reporting problems.
- Examples of exceptions and error situations.
- Frequently asked questions (FAQs).
- Explanation of how to get help and procedures for updating the user manual.

System Documentation

System documentation serves as the technical specifications for the IS(Information Services) and how the objectives of the IS(Information Services) are accomplished. Users, managers and IS owners need never reference system documentation. System documentation provides the basis for understanding the technical aspects of the IS when modifications are made.

- It describes each program within the IS and the entire IS itself.
- It describes the system's functions, the way they are implemented, each program's purpose within the entire IS with respect to the order of execution, information passed to and from programs, and overall system flow.
- It includes data dictionary entries, data flow diagrams, object models, screen layouts, source documents, and the systems request that initiated the project.
- Most of the system documentation is prepared during the system analysis and system design phases.
- During systems implementation, an analyst must review system documentation to verify that it is complete, accurate, and up-to-date, and including any changes made during the implementation process.

Design methodologies aim at the following

- Improve productivity of the analysts and programmers.
- Improve documentation.
- Cut down costs.
- Improve communication among users, analysts, designers and the programmers.
- Simplify the design by segmentation.

STRUCTURED DESIGN:

Structured design is a data flow based methodology. This design partitions a program into small, independent modules. They are arranged in a hierarchy called a top down manner. This method minimizes complexity and makes a problem manageable by subdividing it into smaller segments. This approach is called modularization or decomposition.

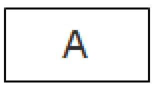
Design is said to be top down if it consists of a hierarchy of modules, with each module having a single entry and single exit subroutine. The primary advantage of this sign

1.critical interfaces are tested first
2.Early versions of the design, though incomplete, are useful enough to resemble the real system.
3.Structuring the design provides control and improves morale
4.The procedural characteristics define the order that determines processing.

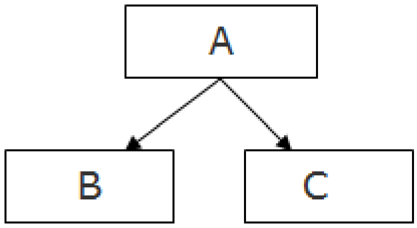
The documentation tool for structured design is the hierarchy or structured chart. It is a graphical tool for representing hierarchy, and it has three elements.

1. Module
2.connection
3.coupling

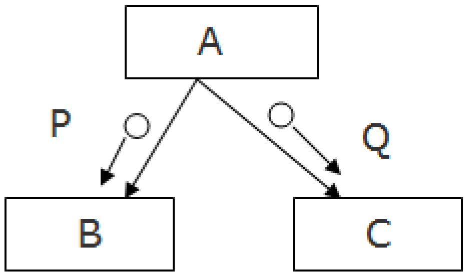
1. The module is represented by a rectangle with a name. It is a contiguous set of statements.



2.The connection is represented by a vector linking two modules. It means one module has called another module



In the above ex the module A calls the module B and it also calls the module C .
3.The couple is represented by an arrow with a circular tail. It represents data items moved from one module to another



FORM – DRIVEN METHODOLOGY:

HIPO is a forms-driven technique. Here standard documents are used to document the information. It consists of a hierarchy chart and an associated set of input/process/output charts. HIPO uses top-down decomposition. It describes the data input and output from processes and defines the data flow composition. The objectives are

1. Provide a structure by which the functions of a system can be understood.
2. State the functions to be performed by the program rather than specifying the program statements to be used to perform the functions.
3. Provide a visual description of input to be used and output to be produced for each level of the diagram.

HIPO makes the transformation of input to output data visible.

HIPO uses easy to draw symbols. The procedure to construct the HIPO chart is as follows

1. Begin at the highest level of abstraction.
2. Identify the processing steps that convert the input to the output.
3. Document each element using the HIPO diagram notation and a treelike structure.

4. Identify sub processes and their inputs and outputs. Continue decomposition until the processes cannot be decomposed further.

There are two tools used for drawing the HIPO diagrams

- HIPO Worksheet ,GX20-1970
- HIPO template , GX20- 1971

The template contains the symbols for the HIPO diagrams. HIPO format consist of Visual table of contents shows the structure of the diagram and the relationships of the functions in a hierarchical manner. It also has a legend to show how symbols are to be used.
Overview diagrams describe the major functions and reference the major details diagrams needed to expand the functions adequately. They provide the following

a) The input section that contains the data items used by the process steps.
b) The output section that contains the data items created by the process steps.
c) Process section that contains numbered steps that describe the functions to be performed. Arrows connect then to the output steps and input/output data items.
d) The extended description refers to non-HIPO documentation and code

Detail diagram contains an extended description section that amplifies the process steps and references the code associated with each process steps.

STRUCTURED WALKTHROUGH:

An activity of all the phases of a structured project is a walkthrough. It is an interchange of ideas to agree on the validity of a proposed solution to a problem. The purpose of the design walkthrough is to anticipate as many problems as possible because it is cheaper to make changes in the design phase rather than in during conversion. The objective is to give a maintainable design that is flexible and adaptable.

User Involvement: Walkthroughs are held to review the system test plan, program design and production acceptance. In each case user has to be involved. The amount of success depends on the amount of use involvement.

- User involvement gives the designer feedback.
- It provides the basic understanding of what a candidate system will do and what it will not.
- It paves a way for user acceptance.

It bridges the gap between the designer and the user

FILE ORGANISATION

SAD

A file is organized to ensure that records are available for processing. There are four methods of organizing files:

1) <u>Sequential Organization</u>:

Sequential organization means storing and sorting in physical, contiguous blocks within files on tape or disk. Records are also in sequence within each block. To access a record previous records within the block are scanned. In a sequential organization, records can be added only at the end of the file. It is not possible to insert a record in the middle of the file without rewriting the file.

In a sequential file update, transaction records are in the same sequence as in the master file. Records from both the files are matched, one record at a time, resulting in an updated master file. In a personal computer with two disk drives, the master file is loaded on a diskette into drive A , while the transaction file is loaded on another diskette into drive B. Updating the master file transfers data from drive B to A controlled by the software in memory.

Advantages:

1. Simple to design
2. Easy to program
3. Variable length and blocked records available
4. Best use of storage space

Disadvantages:

1. Records cannot be added at the middle of the file.

2) <u>Indexed Sequential Organization</u>:

Like sequential organization, keyed sequential organization stores data in physically contiguous blocks. The difference is in the use of indexes to locate records. There are three areas in disk storage: prime area, overflow area and index area.

The prime area contains file records stored by key or id numbers. All records are initially stored in the prime area.

The overflow area contains records added to the file that cannot be placed in logical sequence in the prime area

The index area is more like a data dictionary. It contains keys of records and their locations on the disk. A pointer associated with each key is an address that tells the system where to find a record.

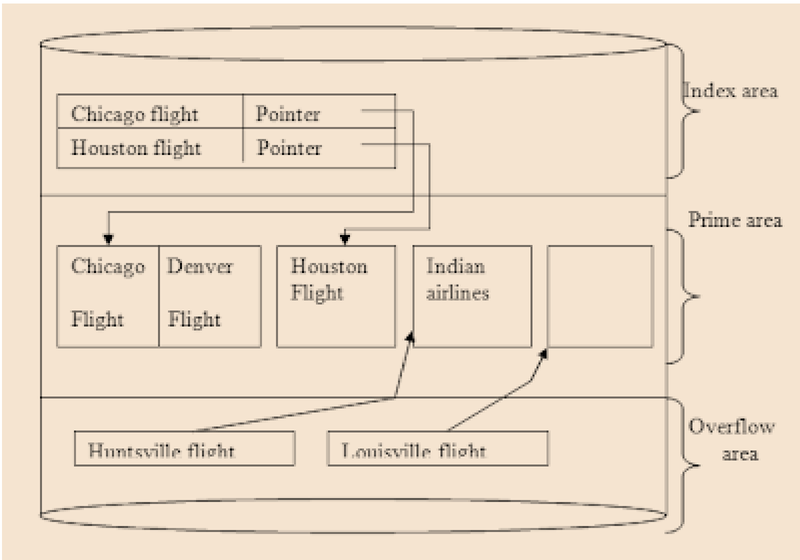
Advantages:

1. Indexed sequential organization reduces the magnitude of the sequential search and provides quick access for sequential and direct processing.

2. Records can be inserted in the middle of the file.

Disadvantages:

1. It takes longer to search the index for data access or retrieval.
2. Unique keys are required
3. Periodic reorganization is required.



3) <u>Inverted List Organization</u>:

Like the indexed- sequential storage method the inverted list organization maintains an index. The two methods differ, however, in the index level and record storage. The indexed sequential method has a multiple index for a given key, whereas the inverted list method has a single index for each key type. In an inverted list, records are not necessarily stored in a particular sequence. They are placed in the data storage area, but indexes are updated for the record key and location. The inverted keys are best for applications that request specific data on multiple keys. They are ideal for static files because additions and deletions cause expensive pointer updating.

Advantages:

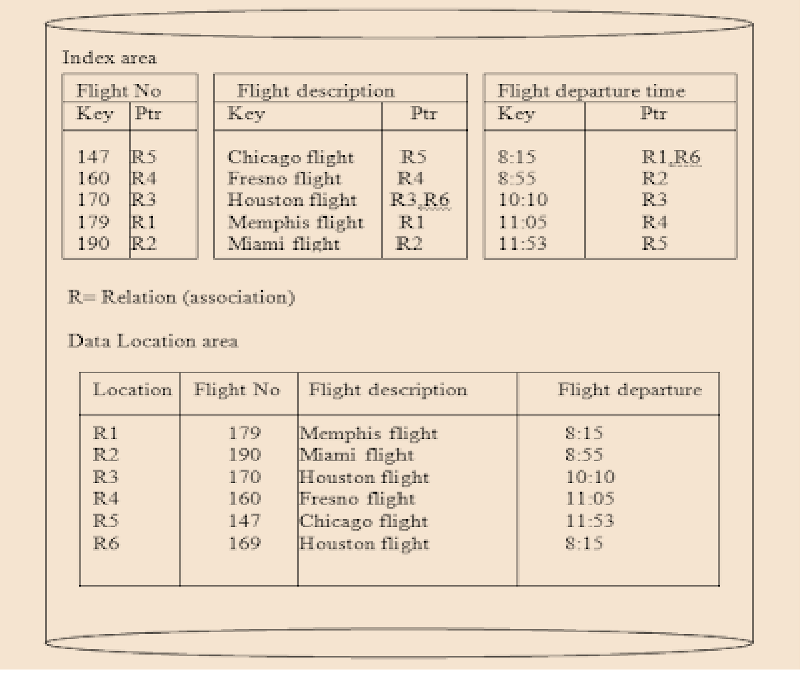
1. Used in applications requesting specific data on multiple keys.

Example:

Data for the flight reservation system.

The flight number, description and the departure time are as given as keys. In the data location area, no particular sequence is followed. If a passenger needs information about the Houston flight, the agent requests the record with Houston flight. The DBMS carries a sequential search to find the required record. The output will then be That the flight number is 170 departing at 10.10 A.M and flight number 169 departing at 8.15 A.M.

if the passenger searches for information about a Houston flight that departs at 8.15,then the DBMS searches the table and retrievesR3 and R6. Then it checks the flight departure time and retrieves R6 standing for flight number 169.



4) Direct access organization:

In direct access file organization, records are placed randomly throughout the file. Records need not be in sequence because they are updated directly and rewritten back in the same location. New records are added at the end of the file or inserted in specific locations based on software commands.

Records are accessed by addresses that specify their disk locations. An address is required for locating a record, for linking records, or for establishing relationships. Addresses are of two types:

1. Absolute
2. Relative.

A **absolute address** represents the physical location of the record. It is usually stated in the format of sector/track/record number. One problem with absolute address is that they become invalid when the file that contains the records is relocated on the disk.

A **relative address** gives a record location relative to the beginning of the file. There must be fixed length records for reference. Another way of locating a record is by the number of bytes it is from the beginning of the file. When the file is moved, pointers need not be updated because the relative location remains the same.

Advantages:

1. Records can be inserted or updated in the middle of the file.

2. Better control over record allocation.

Disadvantages:

1. Calculating address required for processing.

2. Impossible to process variable length records.

5) Chaining:

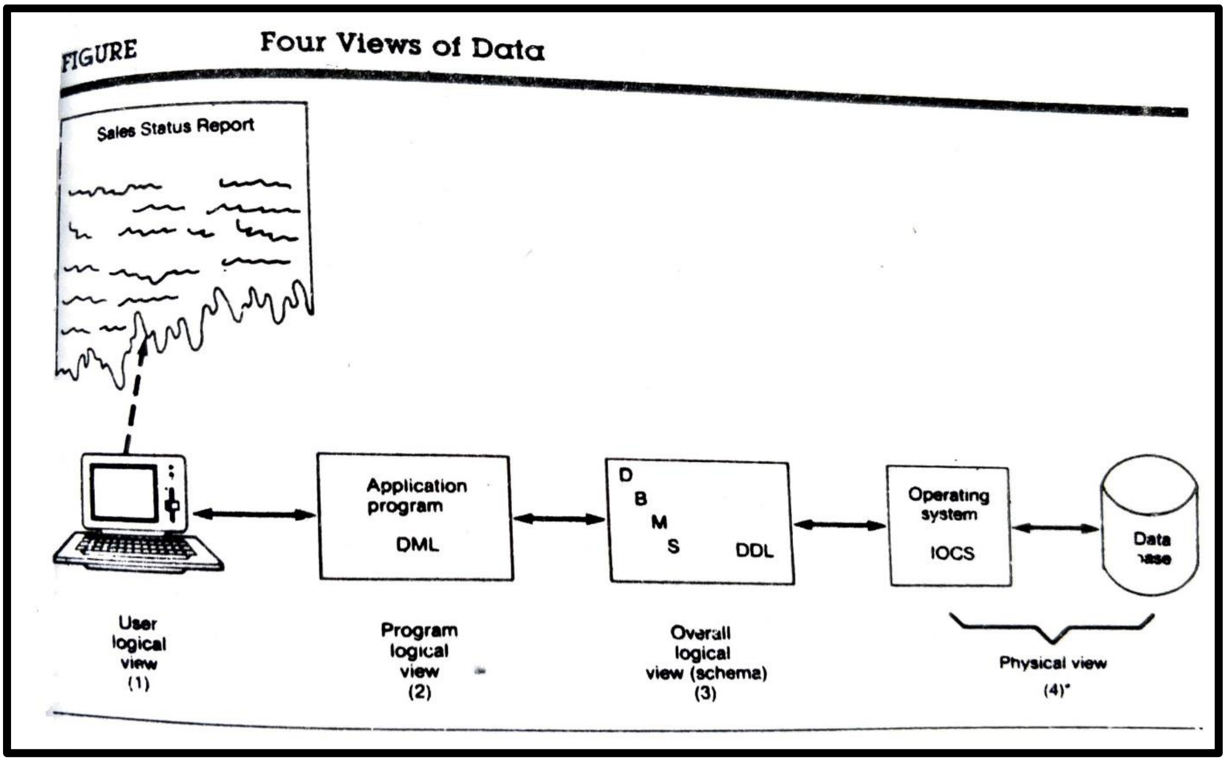
File organization requires that relationships be established among data items. It must show how characters form fields, fields form files and files relate to each other. Establishing relationship is done through chaining. It uses pointers.

Example: The file below contains auto parts that is a indexed sequential file sequenced by part no. A record can be retrieved by part no. To retrieve the next record, the whole file has to be searched. This can be avoided by the use of pointers.

Address	Key 1-Part No	Key 2-Part Type	Pointer
101	014	Muffler	076
102	021	Windshield	061
103	076	Muffler	199
104	061	Windshield	190
201	199	Muffler	END
202	190	Windshield	END

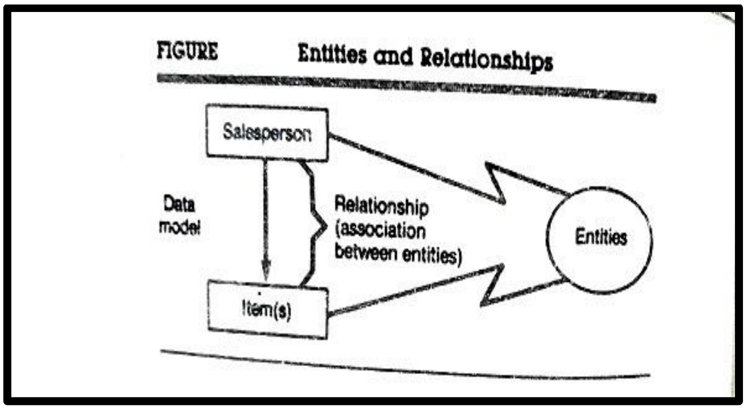
LOGICAL AND PHYSICAL VIEWS OF DATA

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 applications 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 the physical storage. It deals with how data is stored, accessed, or related to other data in storage. **Figure 1** shows four views of data: **three logical** and **one physical**. The **logical views** are the **user view**, the **programmer's views** and **the overall logical view**, called a ***schema***.



Schemas and subschema

The **schema** is the view that helps the DBMS decide what data in storage it should act upon the requested by the application program. An example of a schema is the arrival and departure display of at an airport. Scheduled flights and flight numbers (schema) remain the same, but the actual departure and arrival times may vary. The user's view might be a particular flight arriving or departing at the scheduled time. How the flight actually takes off or lands is one of little concern to the user. The latter view is of *subschema*. It is a programmer (pilot's) view. Many subschemas can be derived from one schema, just as different pilots visualize different views of a landing approach, although all (it is hoped) arrived at the schedule time indicated on the CRT screen display (schema).



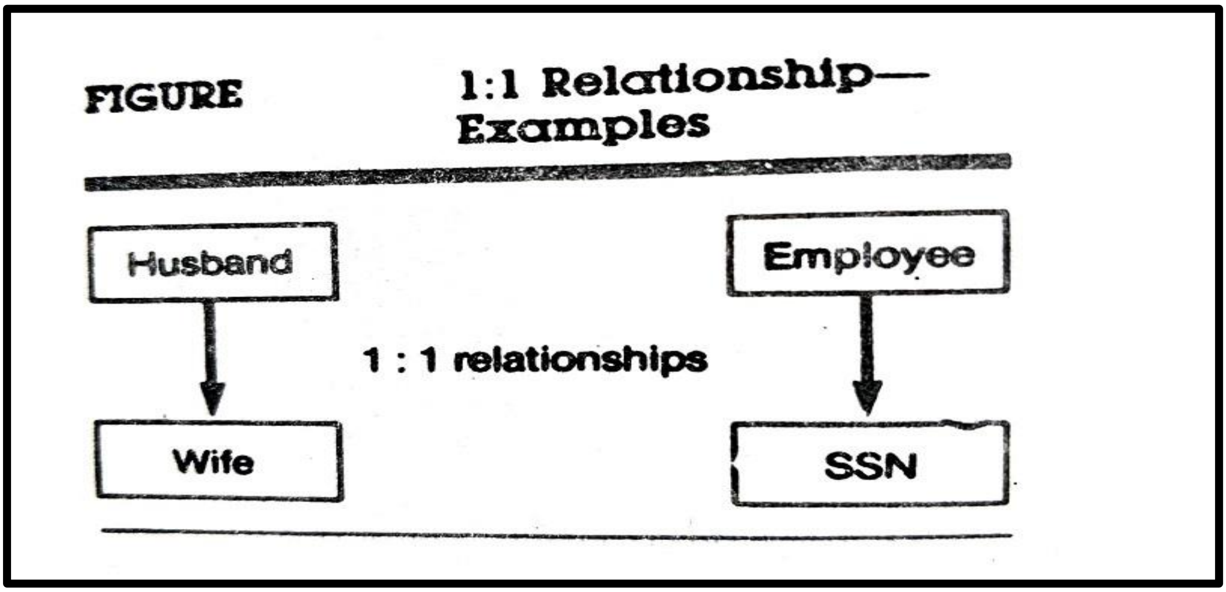
Different application programmers visualise different schemas. The relationships among the schemas, subschemas, and physical structure are provided by the software.

Data Structure

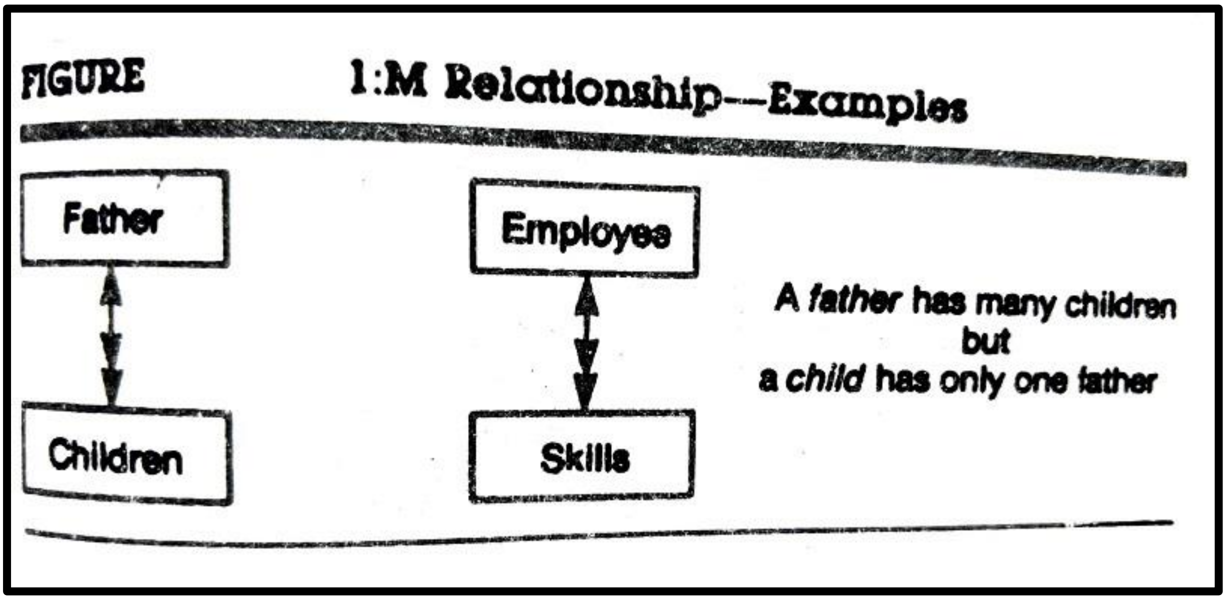
Data are structured according to the data model. In our example see figure 2, sales items are linked to the salesperson who sold them. The salesperson is called an **Entity** and the item sold is also an entity. An entity is a conceptual representation of an object. Relationships between entities make up a data structure. A data model represents a data structure that is described to the DBMS in DDL.

Types of Relationships

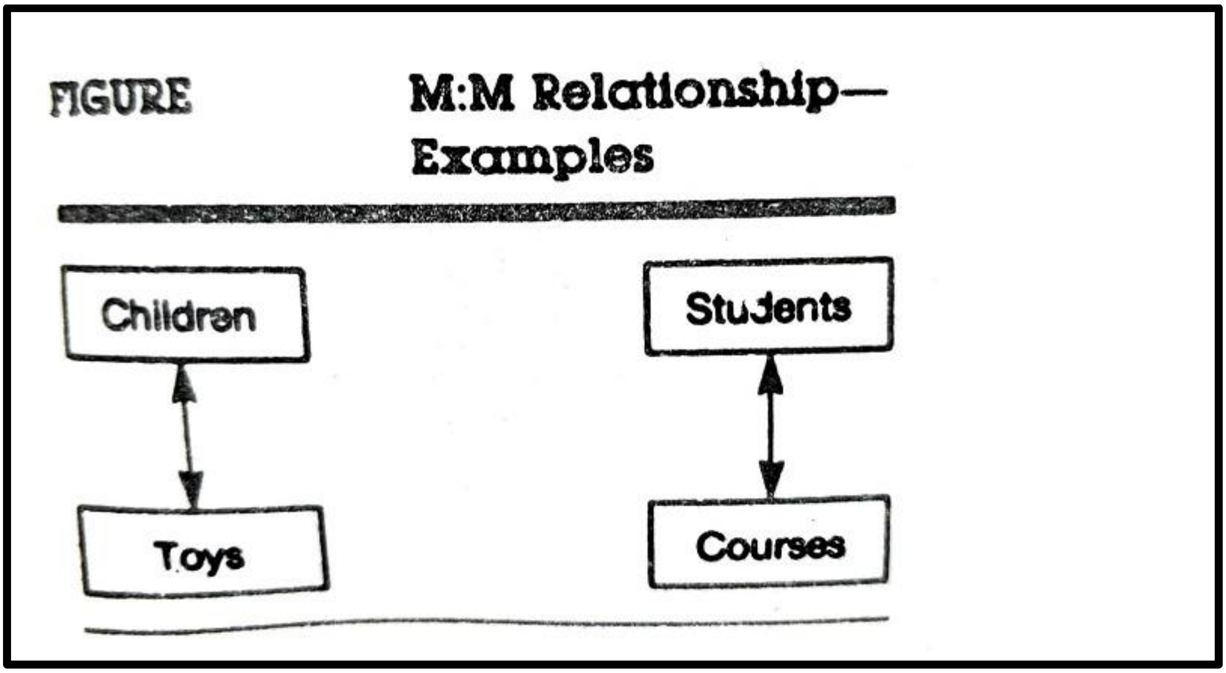
Three types of relationships exist among entities: one-to-one, one-to-many, and many-to-many relationships.



A one-to-one (1:1) relationship is an association between two entities. For example, in our culture, a husband is allowed one wife (at a time) and vice versa, and an employee has one social security number (see figure 2)



A one-to-many (1: M) relationship describes an entity that may have two or more entities related to it. For example, a father may have many children's and an employee may have many skills. (see figure 3)



A many-to-many (M:M) relationship describes entities that may have many relationships both directions. For example, children may have many toys, and students may have many courses (see figure 4)

INPUT OUTPUT FORM DESIGN

Input Design

Inaccurate input data are the most common cause of errors in data processing. Errors entered by data entry operators can be controlled by input design. *Input design* is the process of converting user-originated inputs to a computer-based format. In the system design phase, the expanded data flow diagram identifies logical data flows, data stores, sources, and destination. A system flowchart specifies master files (data base), transaction files, and computer programs. Input data are collected and organized into groups of similar data. Once identified, appropriate input media are selected for processing.

Input Data

The goal of Designing input data is to make data entry as easy, logical, and free from errors as possible. In entering data, operators need to know the following:

1. The allocated space for each field.
2. Field sequence, which must match that in the source document.
3. The format in which data fields are entered; for example, filling out the date field is required through the edited format mm/dd/yy.

When we approach input data design, we design the source documents that capture the data and then select the media used to enter them into the computer. Let us elaborate on each step.

Source Documents

Source data are captured initially on original paper and or a source document. for example, a cheque return against an account is a source document. When it reaches the bank, its encoded with special magnetic ink character reader (MICR) so that it can be processed by a reader that is part of information system of the bank. Therefore, source documents initiate a processing cycle as soon as they are entered into the system.

Source documents may be entered into the system from punch cards, from diskettes, or even directly through the keyboard. A source document, may or may not be retained in the candidate system. Thus, each source document may be evaluated in terms of (1) it is continued use in the candidate system, (2) the extent of modification for the candidate system, and (3) replacement by an alternative source document.

A source document should be logical and easy-to-understand. Each area in the form should be clearly identified and should specify the user what to write and where to write it. For example, a field as simple as date of birth may be returned in four different ways:

1. 12 January 2021

2. Jan. 12, 2021

3.1/12/2021

4. 12/1/2021 (European style)

Unless it is clear in a source document that two digits are allowed for the month, day, and year (MM/DD/YY), we could expect such combinations of responses.

Input Media and Devices

Source data are input into the system in a variety of ways. The following media and devices are suitable for operation:

1. ***Punch cards*** are either 80 or 96 columns wide. Data are arranged in a sequential and logical order. Operators use a keypunch card to copy data from source documents onto cards. This means that the source document and card design must be considered simultaneously.

2. ***Key-to-diskette*** is modelled after the keypunch process. A diskette replaces the card and stores up to 325,000 characters of data-equivalent to the data stored in 4,050 cards. Like cards, data on diskettes are stored in sequence and in batches. The approach to source document and diskette design is similar to that of the punch card. Data must be in sequence and logically cohesive.

3. ***MICR*** translates the special fonts printed in magnetic ink on checks into direct computer input.

4. ***Mark sensing readers*** automatically convert pencil marks in predetermined locations on a card to punched holes on the same card.

5. ***Optical character recognition*** (OCR) readers are similar MICR readers, except that the recognize pencil, ink, or characters by their configuration (shape) rather than their magnetic pattern. They are often used in remote locations as free-standing input preparation devices or direct input media to the system.

6. ***Optical bar code readers*** detect combination of marks that represents data. The most widely known system is **U**niversal **P**roduct **C**ode (UPC) which codes retail items in stores. Automatic tag reading is a major breakthrough in speeding up customer services and eliminating costly data input errors at the point of sale. It is virtually impossible for sales clerk to enter in correct merchandise information such as department and class type data. Automatic tag reading is the ideal way to collect unit inventory information fast, accurately, and economically.

7. ***Cathode ray tube*** (CRT) ***screens*** are used for online data entry. CRT screen displays 20,40, or 80 characters simultaneously on a television-like screen. They show as many as 24 lines of data.

OUTPUT DESIGN

Computer output is the most important and direct source of information to the user. Efficient, intelligible output design should improve the system's relationships with the user and help in decision making. A major form of output is a hard copy from the printer. Printouts should be designed around the output requirements of the user. The output devices to consider depend on factors such as compatibility of the device with the system, response time requirements, expected print quality, and number of copies needed. The following media devices are available for providing computer-based output:

1. MICR readers.

2. Line, matrix, and daisy wheel printers.

3. Computer output microfilm (COM).

4. CRT screen display.

5. Graph plotters.

6. Audio response.

In addition to deciding on the output device, the systems analyst must consider the print format and the editing for the final printout. Editing ranges from suppressing unwanted zeros to merging selected records to produce new figures. In either, the task of output preparation is critical, requiring skill and ability to align user requirements with the capabilities of the system in operation.

The standards for printed output suggest the following:

1. Give each output a specific name or title.

2. Provide a sample of the output layout including areas where printing may appear and the location of each field.

3. State whether each output field is to include significant zeros, spaces between fields and alphabetic or any other data.

4. Specify the procedure for proving the accuracy of output data.

In online application, information is displayed on the screen. The layout sheet for displayed output is similar to the layout chart used for designing input. Areas for displaying the information are blocked out, leaving the rest of the screen blank or for system status information. Allowing the user review sample screens can be extremely important because the user is ultimate judge of the quality of the output and, in turn, the success (or failure) of the system. For example, the following shows editing output for a student birthdate:

DISPLAY: DATE OF BIRTH (mm/dd/yy) 01/13/2021

RESPONSE: MONTH EXCEEDS 12

SUGGESTS A RETRY: DATE OF BIRTH (mm/dd/yy)

FORMS DESIGN

As we know that data provide the basis for or information systems. Without data there is no system, but data must be provided in the right form for input and the information produced must be in a format acceptable to the user. In either case, it is still data-the basic element of a printed form.

What is a form?

People read from forms, write on forms, and spend hours handling forms and filling forms. The data the forms carry come from people, and the informational output of the system goes to people. So, the form is a tool with a message; it is the physical carrier of data - of information. It also can constitute authority for action. For example, a purchase order says BUY, a customer's order says SHIP, and a pay check says PAY TO THE ORDER OF, each form is a request for action. It provides information for making decisions and improving operations.

With this in mind, it is hard to imagine a business operating without using forms. They are the vehicles for most communications and blueprint for many activities. As important as the printed form is, however, the majority of forms are designed by poorly trained people. People are puzzled by confusing forms; they ask for directions on how to read them and how to fill them out. When a form is poorly designed, it is a poor (and costly) administrative tool.

Classification of forms

A printed form is generally classified by what it does in the system. There are three primary classifications: action, memory, and report forms.

An ***Action form*** requests the user to do something - get action. (Examples are purchase orders and shop orders.)

A ***Memory form*** is a record of historical data that remains in a file, is used for reference, and serves as control on key details. (Examples are inventory records, purchase records and bond registers.)

A ***Report form*** guide supervisors and other administrator in their activities. It provides data on a project or a job. (Examples are profit and loss statements and sales analysis report.)

Figure is a summary of the characteristics and examples of these forms.

FIGURE Three Classes of Forms—A Summary

Class	Characteristics	Examples
Action	1. Orders, instructs, authorizes 2. Achieves results 3. Goes from one place (person) to another	Application form Purchase order Sales slip Shop order Time card
Memory	1. Represents historical data 2. Data generally used for reference 3. Stationary and remains in one place, usually in a file 4. Serves as control on certain details	Bond register Inventory record Journal sheet Purchase record Stock ledger
Report	1. Summary picture of a project 2. Provides information about job or details that need attention 3. Used by a manager with authority to effect change 4. Used as a basis for decision making	Balance sheet Operating statement Profit and loss statement Sales analysis Trial balance

Requirements of Forms Design

Forms design follows analysing forms, evaluating present documents, and creating new or improved forms. Bear in mind that detailed analysis occurs only after the problem definition stage and the beginning of designing the candidate system. Since the purpose of a form is to communicate effectively through forms design, there are several major requirements:

1. Identification and wording: The form title must clearly identify its purpose. Columns and rows should be labelled to avoid confusion. The form should also be identified by firm name or code number to make it easy to reorder.

2. Maximum readability and use: The form must be easy to use and fill out. It should be legible, intelligible and uncomplicated. Ample writing space must be provided for inserting data. This means analysing for adequate space and balancing the overall form layout, administration, and use.

3. Physical factors: the form's composition, colour, layout bracket (margins, space, etc.), and paper stock should lend themselves to easy reading. pages should be numbered when multi-page reports are being generated for the user.

4. Order of data items: the data requested should reflect a logical sequence. Related data should be in adjacent positions. Data copied from the source document should be in the same sequence or both the forms. Much of this design take place in the form's analysis phase.

5. Ease of data entry: if use for data entry, the form should be field positions indicated under each column of data and should have some indication of where decimal points are (used broken vertical lines).

6. Size and arrangement: the form must be easily stored and field. It should provide for signatures. Important item must be in a prominent location on the form.

7. Use of instructions: The instructions that accompany a form should clearly show how it is used and handled.

8. Efficiency consideration: the form must be cost-effective. This means eliminating unnecessary data entry and facilitating reading lines across the form. To illustrate, if a poorly design form causes 10 supervisors to waist 30 seconds each than 5 minutes and lost because of the form. If the form uses 10,000 of these forms per year than 833 hours of lost time could have been saved by a better forms design.

9. Type of report: form design should also consider whether the content is executive summary intermediate managerial information, for supporting data. The user requirements for each type of often determine the final form design.

Types of Forms

Forms are classified into several categories: flat forms, unit-set/snap out forms, continuous strip/fanfold forms, NCR paper, and pre-printed forms. These types are described briefly.

Flat Forms

A flat form is single-copy form prepared manually or by a machine and printed on any grade of paper. For additional copies of the original, carbon paper is inserted between copies. It is the easiest form to design, print, and reproduce; it has a low-volume use; and it is the least expensive. Often a pad of the flat forms is printed identical to the original copy of a unit set.

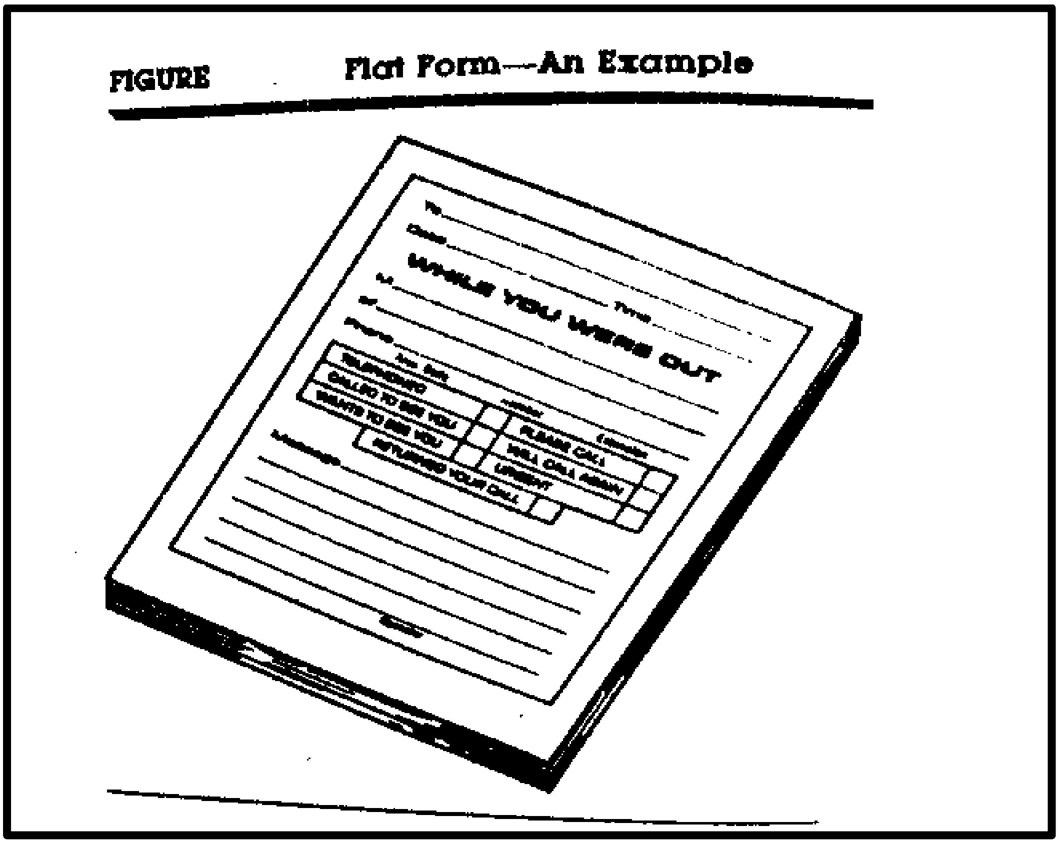


Figure of *Flat Forms*

Unit-Set/Snap out Forms

These forms have an original copy and several copies with one-time carbon paper interleaved between them. The set is glued into a unit (thus unit set) for easy handling. The carbon paper is approximately 3/8 inch shorter than the copies. The copies are perforated at the glue margin for tearing out, call to the carbon is not perforated. Because of the perforation and the shorter carbon, the forms can be easily snapped and (thus, the name snap out form) after completion.

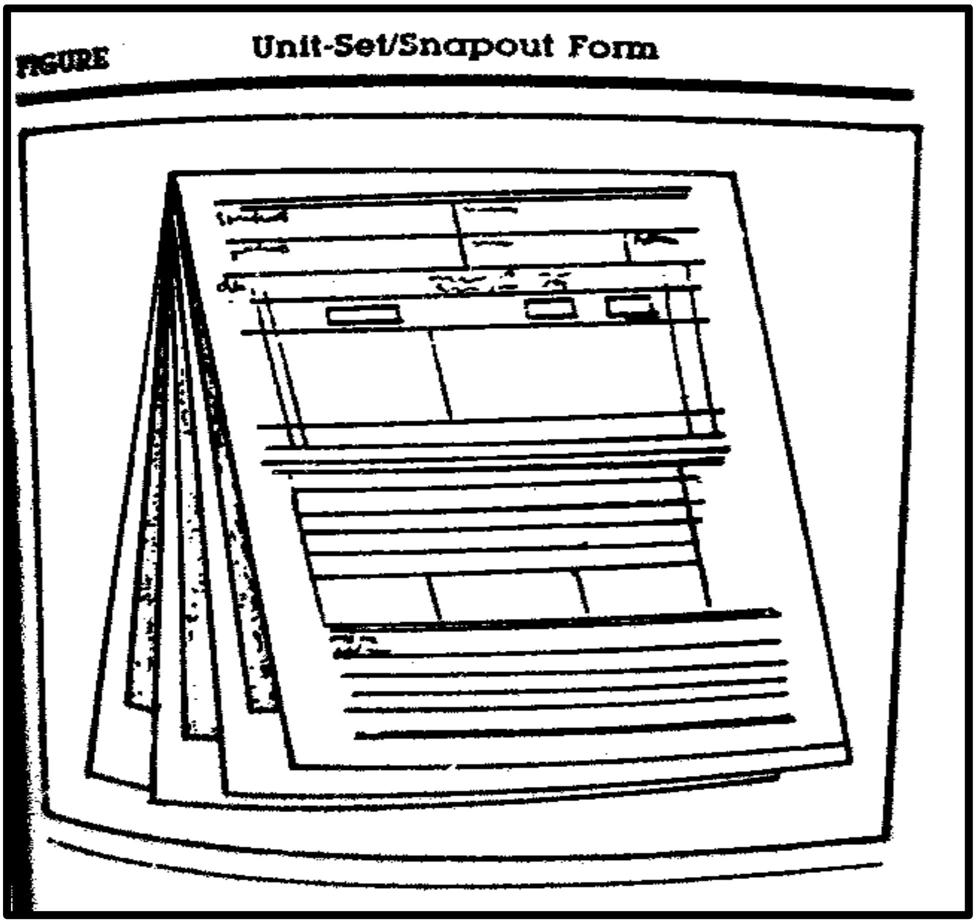


Figure of *Unit-Set/Snap out Forms*

Continuous Strip/Fanfold Forms

These are multiple-unit forms join together in a continuous strip with perforations between each pair of forms. One-time carbon is interleaved between copies, which are stacked in a fanfold arrangement. Note the pin-feed holes punched in both margins for mounting the forms onto the sprocket wheels of the printer. The device eliminates individual insertion of forms. The fanfold is the least expensive construction for large-volume use. Computer printouts are invariably produced on them; they are virtually part of system design.

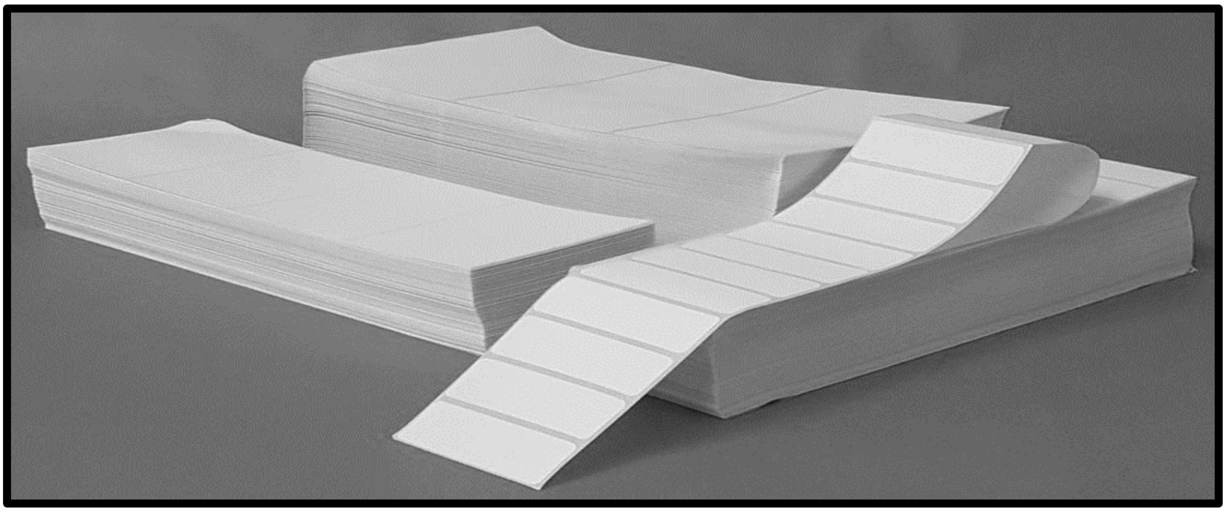


Figure of *Continuous Strip/Fanfold Forms*

NCR (no carbon required) Paper

Several copies can be made by pressing a chemical undercoating on the top sheet into a clay like coating on the top of the second sheet. The writing (or printing) pressure forms an image by the coating material. The same process applies to the back of the second sheet for producing a carbon copy on the face of the succeeding sheet, and so on (see figure).

NCR paper has many applications in sales books, check books, inventory tickets, and deposit slips. It offers cleaner, clearer, and longer-lasting copies than carbon-interleaved forms. No carbon means no smears or smudges.

One problem is the sensitivity of the chemical: it shows every unintended scratch. Other disadvantages are difficulty with erasures and high cost. NCR paper costs as much as 25% more than carbon-interleaved forms. Considering the labour saving of the NCR process, however, cost may be well justified in the long run.

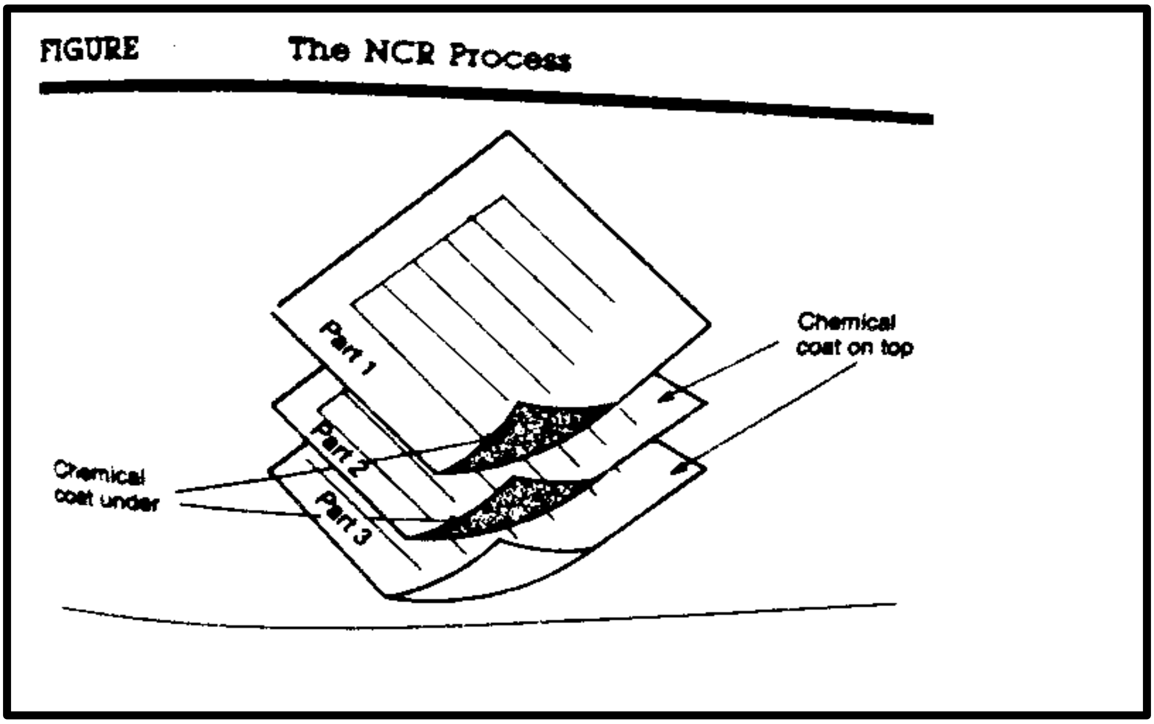


Figure of *NCR (no carbon required) Paper*