What is Feasibility? Describe the different types of Feasibility.

Ans.: Feasibility is the determination of whether or not a project is worth doing. The process followed in making this determination is called feasibility study. A feasibility study is carried out to select the best system that meets performance requirements. When conducting feasibility study, an analyst can consider 7 types of feasibility:

Technical Feasibility : It is concerned with specifying the equipment and the computer system that will satisfy and support the proposed user requirements. Here we need to consider the configuration of the system which tells the analyst how many work stations are required, how the units are interconnected so that they can operate and communicate smoothly.

Operation Feasibility : It is related to human organizational aspects. The points to be considered here are – what changes will be brought with the system?, what new skills will be required?, do the existing staff members have these skills and can they be trained?

Economic Feasibility : It is the most frequently used technique for evaluating a proposed system. It is also called Cost/Benefit Analysis. It is used to determine the benefits and savings that are expected from the proposed system and compare them with the costs. If benefits are more than the cost, the proposed system is given an OK.

Social Feasibility : It is a determination of whether the proposed system will be acceptable to the people or not. It finds out the probability of the project being accepted by the group of people who are directly affected by the changed system.

Management Feasibility : It is a determination of whether the proposed system is acceptable to the management of the organization. The project may be rejected, if the management does not accept the proposed system.

Legal Feasibility : It is a determination of whether the proposed project is under legal obligation of known Acts, Statutes, etc.

Time Feasibility : It is a determination of whether the project will be completed within a specified time period. If the project takes too much time, it is likely to be rejected.

Q- Explain data modeling?

Data Modeling : It gives answers to questions regarding the data that is to be used in the application. We come to know the data objects, where they are stored, what is the relationship between objects, etc. Data modeling uses an Entity Relationship diagram to solve these questions. An Entity Relationship diagram will focus on all data that are entered, stored, transformed and produced within an application. The data model consists of three interrelated information – data objects, attributes that describe the data objects and relationships that connect data objects to one another.

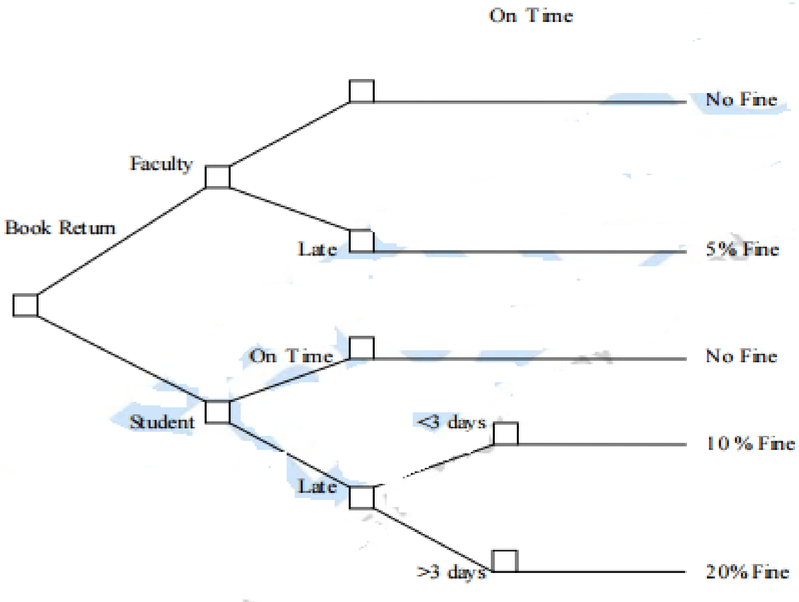
Q- What is data flow diagram?

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. It describes the system‟s data and how the processes transform the data in a graphical manner. Data flow diagrams can be used to provide a clear representation of any business function. It starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. It uses a top-down approach to show all the levels of the functions of the system. Initially a context diagram is drawn, which is a simple representation of the entire system under investigation. This is followed by a level 1 diagram; which provides an overview of the major functional areas of the business. The level 1 diagram identifies the major business processes at a high level and any of these processes can then be analyzed further - giving rise to a corresponding level 2 business process diagram. This process of more detailed analysis can then continue – through level 3, 4 and so on.

Q- Briefly describe a Decision Tree with example.

Ans.: Decision tree are graphical representation methods of representing a sequence of logical decisions. It is mainly used when decisions need to be taken or for defining policies. A decision tree has as many branches as there are logical alternatives. It is easy to construct, easy to read and easy to update. A decision tree is used to identify the strategy most likely to reach a goal. It is also used as a means for calculating probabilities or making financial or number based decisions. A decision making tree is essentially a diagram that represents, in a specially organized way, the decisions, the main external or other events that introduce uncertainty, as well as possible outcomes of all those decisions and events.

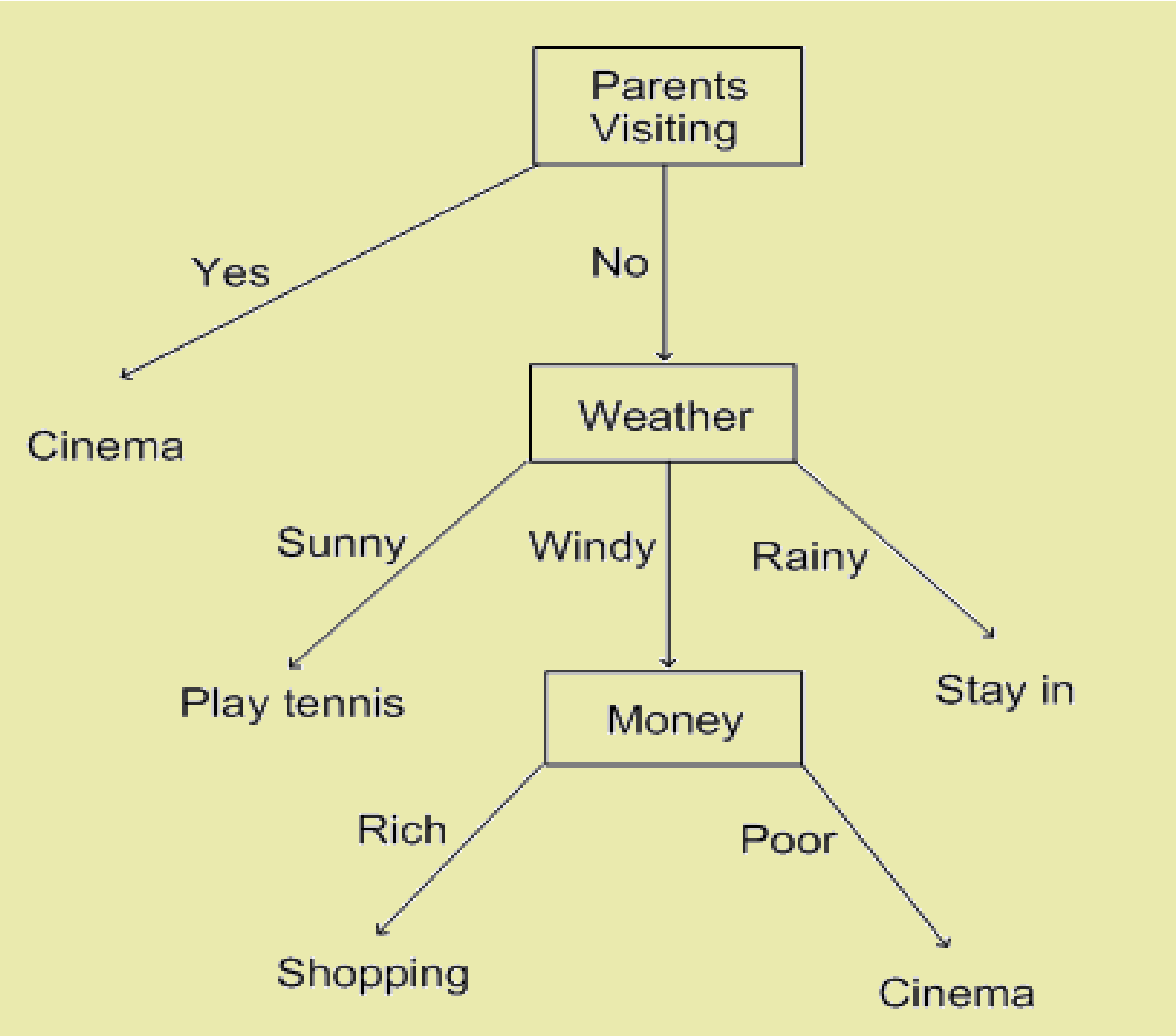
Example : Book return policy in library If a Faculty returns a book late, a fine of 5% of the book rate is charged. If a Student returns a book late by 3 days, fine is 10%, else 20% of book rate.



The root of the tree is the starting point of the decision sequence. The particular branch to be followed depends on the conditions that exist and decision that will be made. Progression from the left to right along a particular branch is the result of making a series of decisions.

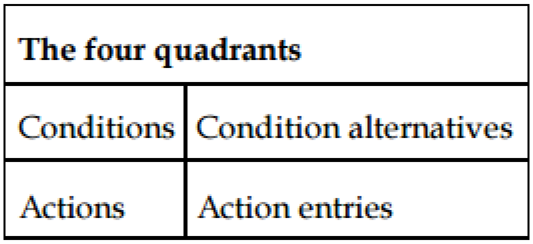
Example: **Imagine you only ever do four things at the weekend: go shopping, watch a movie, play tennis or just stay in. What you do depends on three things: the weather (windy, rainy or sunny); how much money you have (rich or poor) and whether your parents are visiting. You say to your yourself: if my parents are visiting, we'll go to the cinema. If they're not visiting and it's sunny, then I'll play**

**tennis, but if it's windy, and I'm rich, then I'll go shopping. If they're not visiting, it's windy and I'm poor, then I will go to the cinema. If they're not visiting and it's rainy, then I'll stay in.**



Q-What are Decision Tables? Explain with example.

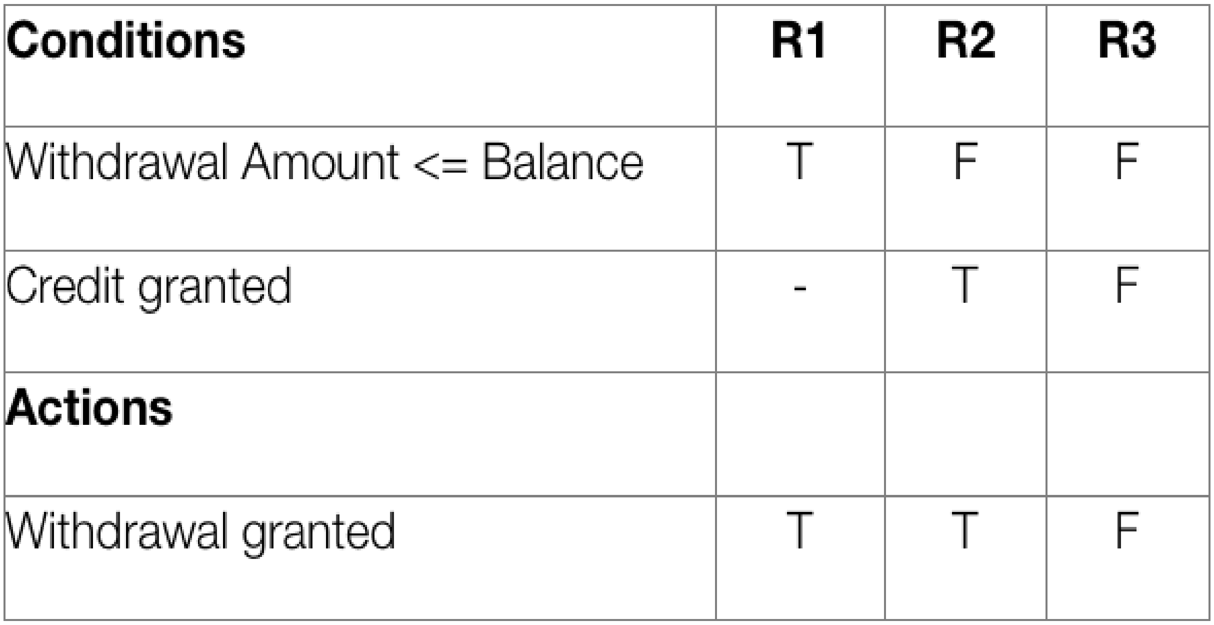
Ans.: Decision tables are a precise yet compact way to model complicated logic. Decision tables, like if-then-else and switch-case statements, associate conditions with actions to perform. But, unlike the control structures found in traditional programming languages, decision tables can associate many independent conditions with several actions in an elegant way. Decision tables are typically divided into four quadrants, as shown below



| The four quadrants | |
|---|---|
| Conditions | Condition alternatives |
| Actions | Action entries |

A decision table lists causes and effects in a matrix. Each column represents a unique combination. The following are various components of a Decision table: Condition stubs: This portion of table describes the conditions or factors that will affect the decision or policy making of the organisation. Action stubs: This portion describes the possible policy actions or decisions in the form of statements. Rule: Rules describe which actions are to be taken under a specific combination of conditions. Decision tables use a standard format and handle combinations of conditions in a very concise manner. Decision table also provides technique for identifying policy incompleteness and contradictions.

Example-A customer requests a cash withdrawal. One of the business rules for the ATM is that the ATM machine pays out the amount if the customer has sufficient funds in their account or if the customer has the credit granted.



| **Conditions** | **R1** | **R2** | **R3** |
|---|---|---|---|
| Withdrawal Amount <= Balance | T | F | F |
| Credit granted | - | T | F |
| **Actions** | | | |
| Withdrawal granted | T | T | F |

Q- What is Structured English?

Ans.: Structured English or pseudo code or program design language (PDL) uses the vocabulary of English and the syntax of a structured programming. Structured English looks like a modern programming language. The difference between structured English and a real programming language is in the use of narrative text which is placed within the structured English statements. Structured English cannot be compiled. It should have the following characteristics: o A fixed syntax of keywords used for structured constructs, data declaration o A free syntax of natural language that describes processing o Data declaration facilities that include simple(array) and complex(linked list or tree) data structures o Facility to declare subprograms and call them Decisions in Structured English are made through IF, THEN, ELSE, SO, etc.

The following are the guidelines usage of Structured English notation:

1. Avoid computer programming language verbs such a move, open or close.

2. The statement used in the Structured English Notation should always specify the formula to be used.

Structured English notation is based on the principles of structured programming. Process specification logic consists of a combination of sequences of one or more imperative sentences with decision and repetition constructs.

Q- What are Structure Charts? Describe.

Ans.: Structure Chart : A hierarchical diagram showing the relationships between the modules of a computer program. A module is the basic component of a structure chart and is used to identify a function. Modules are relatively simple and independent components. Higher-level modules are “control” modules that control the flow of execution. Lower level modules are “worker bee” modules and contain the program logic to actually perform the functions. The vertical lines connecting the modules indicate the calling structure from the high-level modules to the lower-level modules. The little arrows next to the lines show the data that is passed between modules and represent the inputs and outputs of each module. At the structure chart level, we are not concerned with what is happening inside the module yet. We only want to know that somehow it does the function indicated by its name using the input data and producing the output data. A program call is when one module invokes a lower-level module to perform a needed service or calculation. Program call: The transfer of control from a module to a subordinate module to perform a requested service. The arrows with the open circle, called data couples, represent data being passed into and out of the module. A data couple can be an individual data item (e.g., a flag or a customer account number) or a higher-level data structure (e.g., an array, record, or other data structure). The arrow with the darkened circle is a “flag.” A flag is purely internal information that is used between modules to indicate some result.

## User interface design

User interface design is concerned with the dialogue between a user and the computer. It is concerned with everything from starting the system or logging into the system to the eventually presentation of desired inputs and outputs. The overall flow of screens and messages is called a dialogue. The following are various guidelines for user interface design: i) The system user should always be aware of what to do next. ii) The screen should be formatted so that various types of information, instructions and messages always appear in the same general display area. iii) Messages, instructions or information should be displayed long enough to allow the system user to read them. iv) Use display attributes sparingly. v) Default values for fields and answers to be entered by the user should be specified. vi) A user should not be allowed to proceed without correcting an error. vii) The system user should never get an operating system message or fatal error.

## Data Dictionary

A Data Dictionary consists of data about data. The major elements of data dictionary are data flows, data stores and processes. The data dictionary stores details and descriptions of these elements. It does not consist of actual data in the database. But, DBMS cannot access data in database with out accessing data dictionary. If analysts want to know the other names by which a data item is referenced in the system or where it is used in the system, they should be able to find the answers in properly developed data dictionary. Data dictionaries are hidden from users so that data in it is not tampered. Analysts use data dictionaries for the following reasons:

1. To manage the detail in large systems.

2. To communicate a common meaning for all system elements.

3. To document the features of the system.

4. To facilitate analysis of the details in order to evaluate characteristics and determine changes that should be made to the system.

5. To locate errors and omissions in the system.

The dictionary contains two types of descriptions for the data flowing through the system: Data elements and Data structures. Data elements are grouped together to make up a data structure.

Data elements are recorded in data dictionary at the fundamental data level. Each item is identified by a data name, description, alias and length and has specific values that are permissible for it in the system. A data structure is a set of data items that are related to one another and then collectively describe a component in the system. Data is arranged in accordance with one of the relationships namely sequence, selection, iteration and optional relationship.

# Pseudocode

A pseudocode does not conform to any programming language and expresses logic in plain English.

- It may specify the physical programming logic without actual coding during and after the physical design.
- It is used in conjunction with structured programming.
- It replaces the flowcharts of a program.

# Guidelines for Selecting Appropriate Tools

Use the following guidelines for selecting the most appropriate tool that would suit your requirements −

- Use DFD at high or low level analysis for providing good system documentations.
- Use data dictionary to simplify the structure for meeting the data requirement of the system.
- Use structured English if there are many loops and actions are complex.
- Use decision tables when there are a large number of conditions to check and logic is complex.
- Use decision trees when sequencing of conditions is important and if there are few conditions to be tested.