Topic 3 & 4
Software Requirements Analysis & Specification

Requirement Engineering

Requirements describe

What not **How**

Produces one large document written in natural language contains a description of what the system will do without describing how it will do it.

Crucial process steps

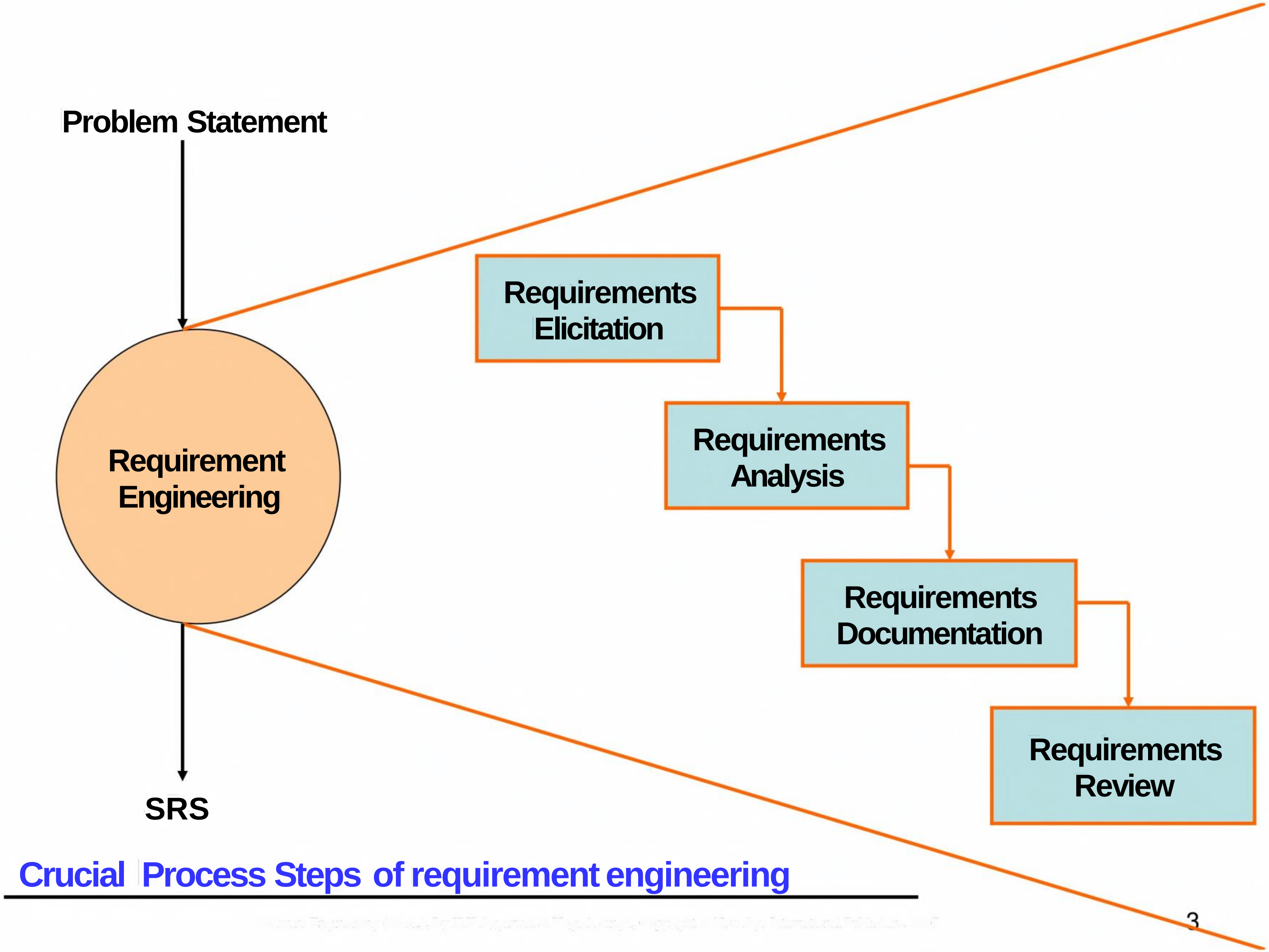
Quality of product m) > Process that creates it

Without well written document

-- Developers do not know what to build

-- Customers do not know what to expect

-- What to validate



Crucial Process Steps of requirement engineering

Requirement Engineering

Requirement Engineering is the disciplined application of proven principles, methods, tools, and notations to describe a proposed system's intended behavior and its associated constraints.

SRS may act as a contract between developer and customer.

State of practice

Requirements are difficult to uncover

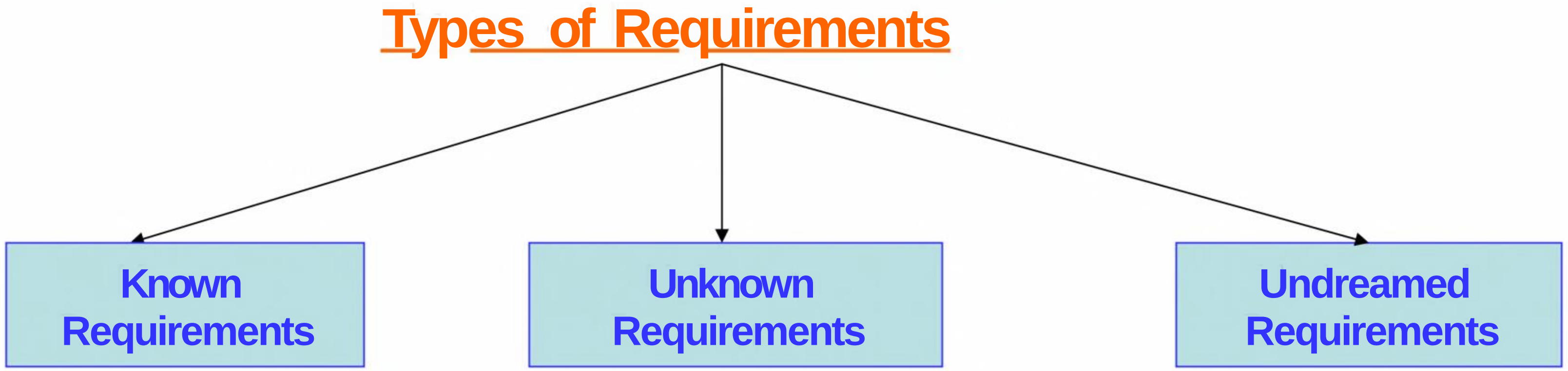
- **Requirements change**
- **Over reliance on CASE Tools**
- **Tight project Schedule**
- **Communication barriers**
- **Market driven software development**
- **Lack of resources**

Requirement Engineering

Example

A University wish to develop a software system for the student result management of its M.Tech. Programme. A problem statement is to be prepared for the software development company. The problem statement may give an overview of the existing system and broad expectations from the new software system.

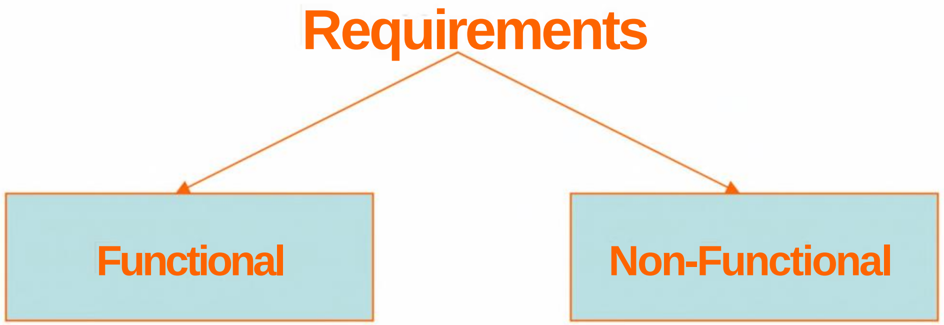
Types ofRequirements



Stakeholder: Anyone who should have some direct or indirect influence on the system requirements.

--- User

--- Affected persons



Types ofRequirements

Functional requirements describe what the software has to do. They are often called product features.

Non Functional requirements are mostly quality requirements. That stipulate how well the software does, what it has to do.

Availability
Reliability
Usability
Flexibility

For Users

Maintainability
Portability
Testability

For Developers

Types ofRequirements

User and system requirements

- User requirement are written for the users and include functional and non functional requirement.
- System requirement are derived from user requirement.
- The user system requirements are the parts of software requirement and specification (SRS) document.

Types ofRequirements

Interface Specification

- Important for the customers.

TYPES OF INTERFACES

- Procedural interfaces (also called Application Programming Interfaces (APIs)).
- Data structures
- Representation of data.

Feasibility Study

Is cancellation of a project a bad news?

As per IBM report, "31% projects get cancelled before they are completed, 53% over-run their cost estimates by an average of 189% & for every 100 projects, there are 94 restarts.

How do we cancel a project with the least work?

m) CONDUCT **A FEASIBIL**TY STUDY

Feasi6ility Study

Technical feasibility

- Is it technically feasible to provide direct communication connectivity through space from one location of globe to another location?

- Is it technically feasible to design a programming language using "Sanskrit"?

Feasibility Study

Feasibility depends upon non technical Issues like:

- Are the project's cost and schedule assumption realistic?
- Does the business model realistic?
- Is there any market for the product?

Feasibility Study

Purpose of feasibility study

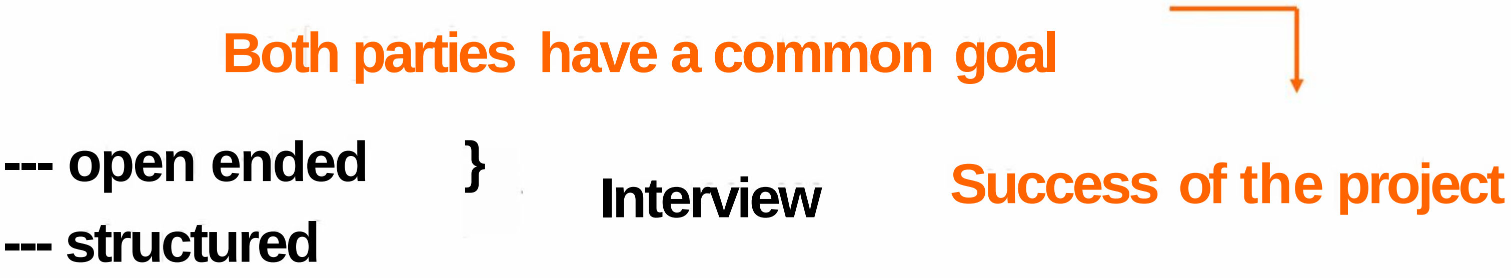
"evaluation or analysis of the potential impact of a proposed project or program."

Focus of feasibility studies

- Is the product concept viable?
- Will it be possible to develop a product that matches the project's vision statement?
- What are the current estimated cost and schedule for the project?

Requirements Elicitation

1.Interviews



Both parties have a common goal

--- open ended }

--- structured

Interview

Success of the project

Selection of stakeholder

1. Entry level personnel
2. Middle level stakeholder
3. Managers
4. Users of the software (Most important)

Requirements Elicitation

5. Possible benefits

6. Satisfied with current policies

7. How are you maintaining the records of previous students?

8. Any requirement of data from other system

9. Any specific problems

10. Any additional functionality

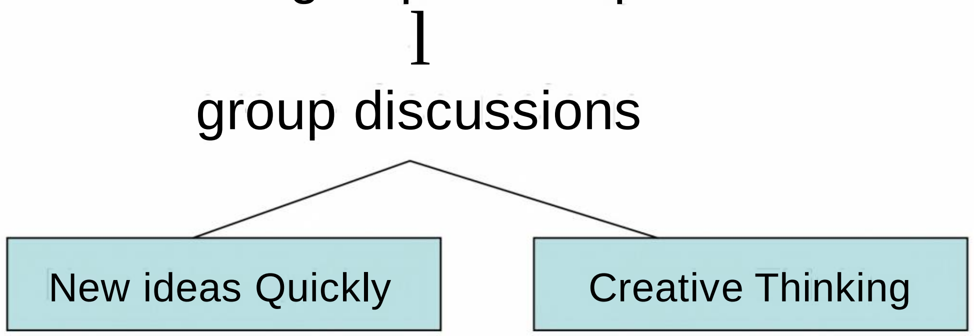
11. Most important goal of the proposed development

At the end, we may have wide variety of expectations from the proposed software.

Requirements Elicitation

2. Brainstorming Sessions

It is a group technique



group discussions

New ideas Quickly

Creative Thinking

Prepare long list of requirements

Categorized
Prioritized
Pruned

*Idea is to generate views ,not to vet them.

Groups

1. Users 2. Middle Level managers 3. Total Stakeholders

Requirements Elicitation

Steps

1. Identify stakeholders
2. List out requirements
3. Degree of importance to each requirement.

Requirements Elicitation

*defines all behavior required of the system, bounding the scope of the system.

Jacobson & others proposed a template for writing Use cases as shown below:

1. Introduction

Describe a quick background of the use case.

2.Actors

List the actors that interact and participate in the use cases.

3.Pre Conditions

Pre conditions that need to be satisfied for the use case to perform.

4. Post Conditions

Define the different states in which we expect the system to be in, after the use case executes.

@Requirements Elicitation

5. Flow of events

5.1 Basic Flow

List the primary events that will occur when this use case is executed.

5.2 Alternative Flows

Any Subsidiary events that can occur in the use case should be separately listed. List each such event as an alternative flow. A use case can have many alternative flows as required.

6.Special Requirements

Business rules should be listed for basic & information flows as special requirements in the use case narration .These rules will also be used for writing test cases. Both success and failures scenarios should be described.

7.Use Case relationships

For Complex systems it is recommended to document the relationships between use cases. Listing the relationships between use cases also provides a mechanism for traceability

Use Case Template

Requirements Elicitation

Use Case Guidelines

1. Identify all users

2. Create a user profile for each category of users including all roles of the users play that are relevant to the system.

3. Create a use case for each goal, following the use case template maintain the same level of abstraction throughout the use case. Steps in higher level use cases may be treated as goals for lower level (i.e. more detailed), sub• use cases.

4. Structure the use case

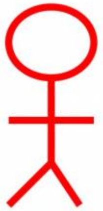
5. Review and validate with users.

Requirements Elicitation

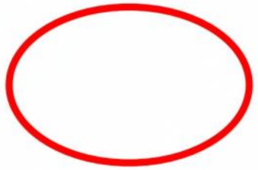
Use case Diagrams

-- represents what happens when actor interacts with a system.

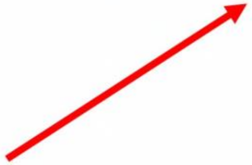
-- captures functional aspect of the system.



Actor



Use Case



Relationship between actors and use case and/or between the use cases.

– Actors appear outside the rectangle.

--Use cases within rectangle providing functionality.

--Relationship association is a solid line between actor & use cases.

Requirements Elicitation

*Use cases should not be used to capture all the details of the system.

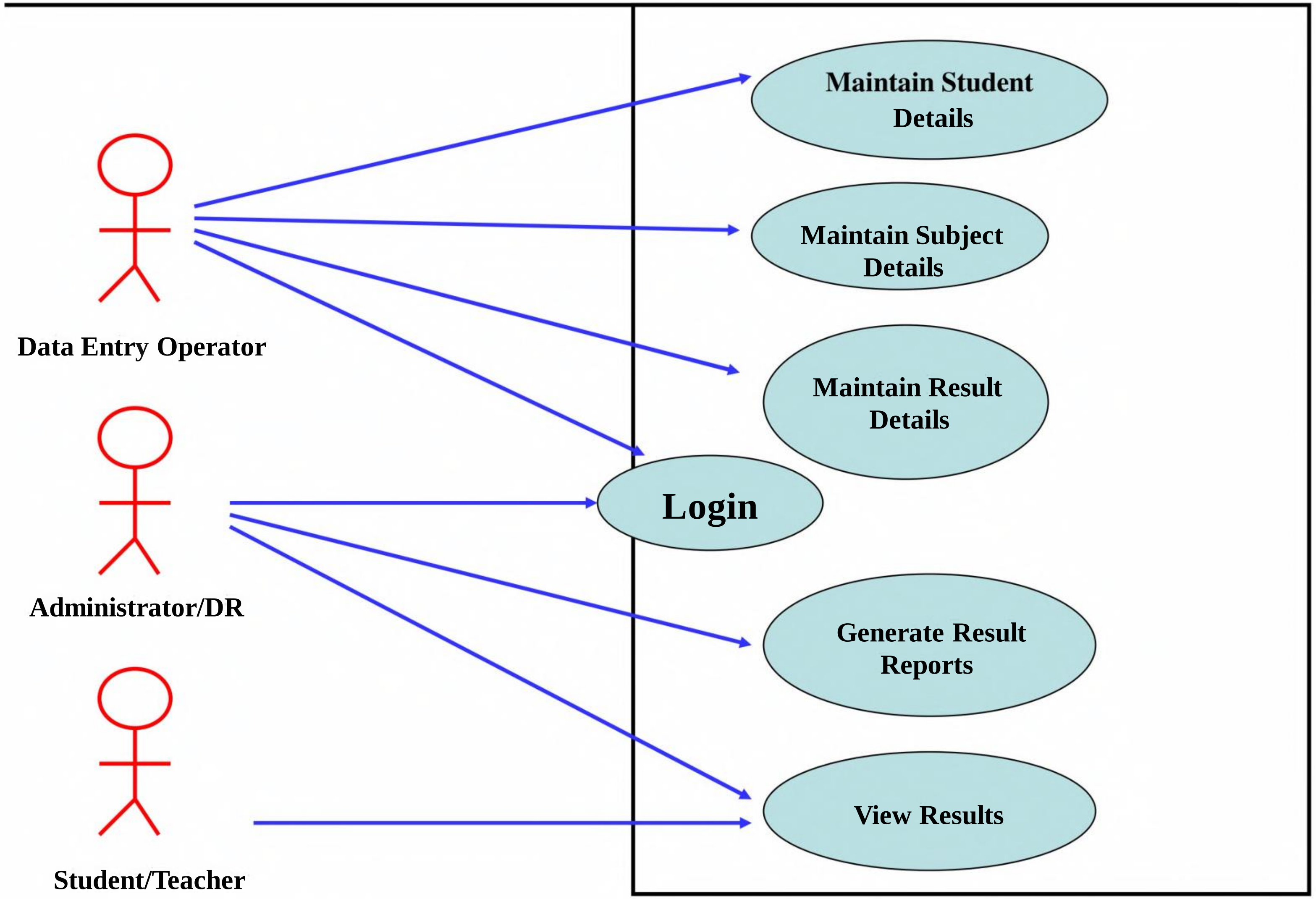
*Only significant aspects of the required functionality

*No design issues

*Use Cases are for "what" the system is, not "how" the system will be designed

* Free of design characteristics

Use case diagram for Result Management System



Requirements Elicitation

1. **Maintain student Details**

   Add/Modify/update students details like name, address.

2. **Maintain subject Details**

   Add/Modify/Update Subject information semester wise

3. **Maintain Result Details**

   Include entry of marks and assignment of credit points for each paper.

4. **Login**

   Use to Provide way to enter through user id & password.

5. **Generate Result Report**

   Use to print various reports

6. **View Result**

   (i) According to course code

   (i) According to Enrollment number/roll number

Requirements Elicitation (Use Case)

Login

1.1 **Introduction :** **This use case describes how a user logs into the Result Management System.**

1.2 **Actors :** (i) **Data Entry Operator**
(ii) **Administrator/Deputy Registrar**

1.3 **Pre Conditions : None**

1.4 **Post Conditions : If the use case is successful, the actor is logged into the system. If not, the system state is unchanged.**

Requirements Elicitation (Use Case)

1.5 **Basic Flow :** This use case starts when the actor wishes to login to the Result Management system.

(i) System requests that the actor enter his/her name and password.

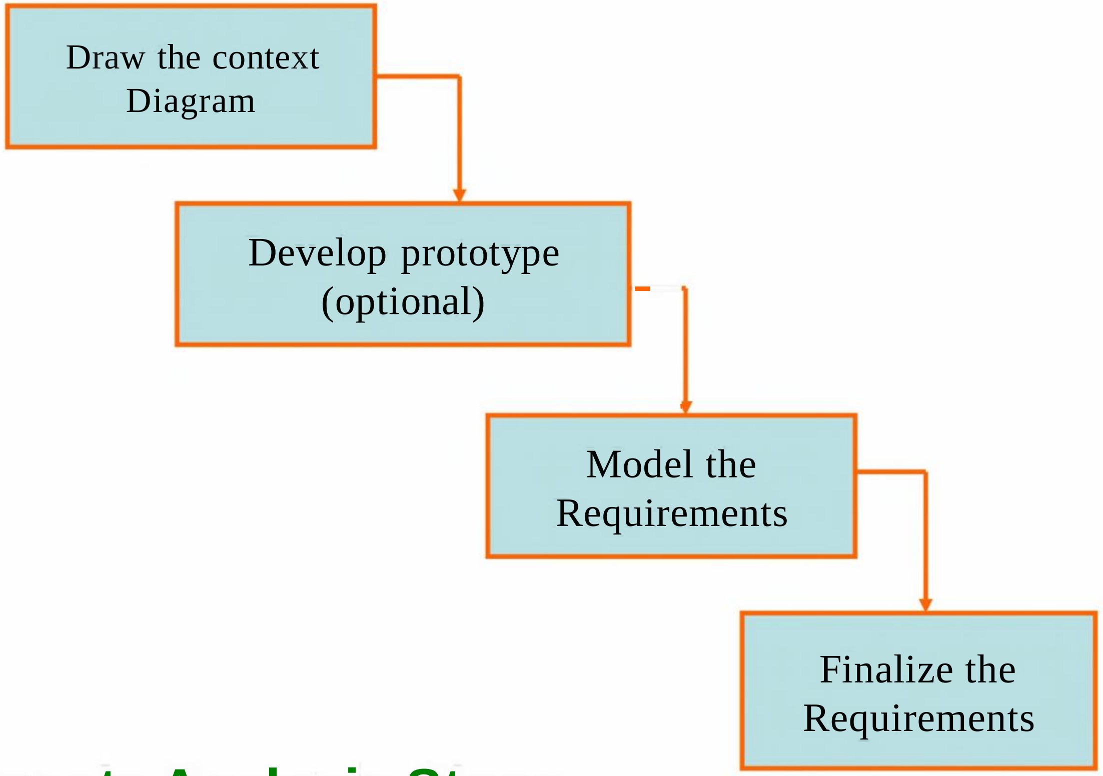
(ii) The actor enters his/her name & password.

(iii) System validates name & password, and if finds correct allow the actor to logs into the system.

Requirements naysis

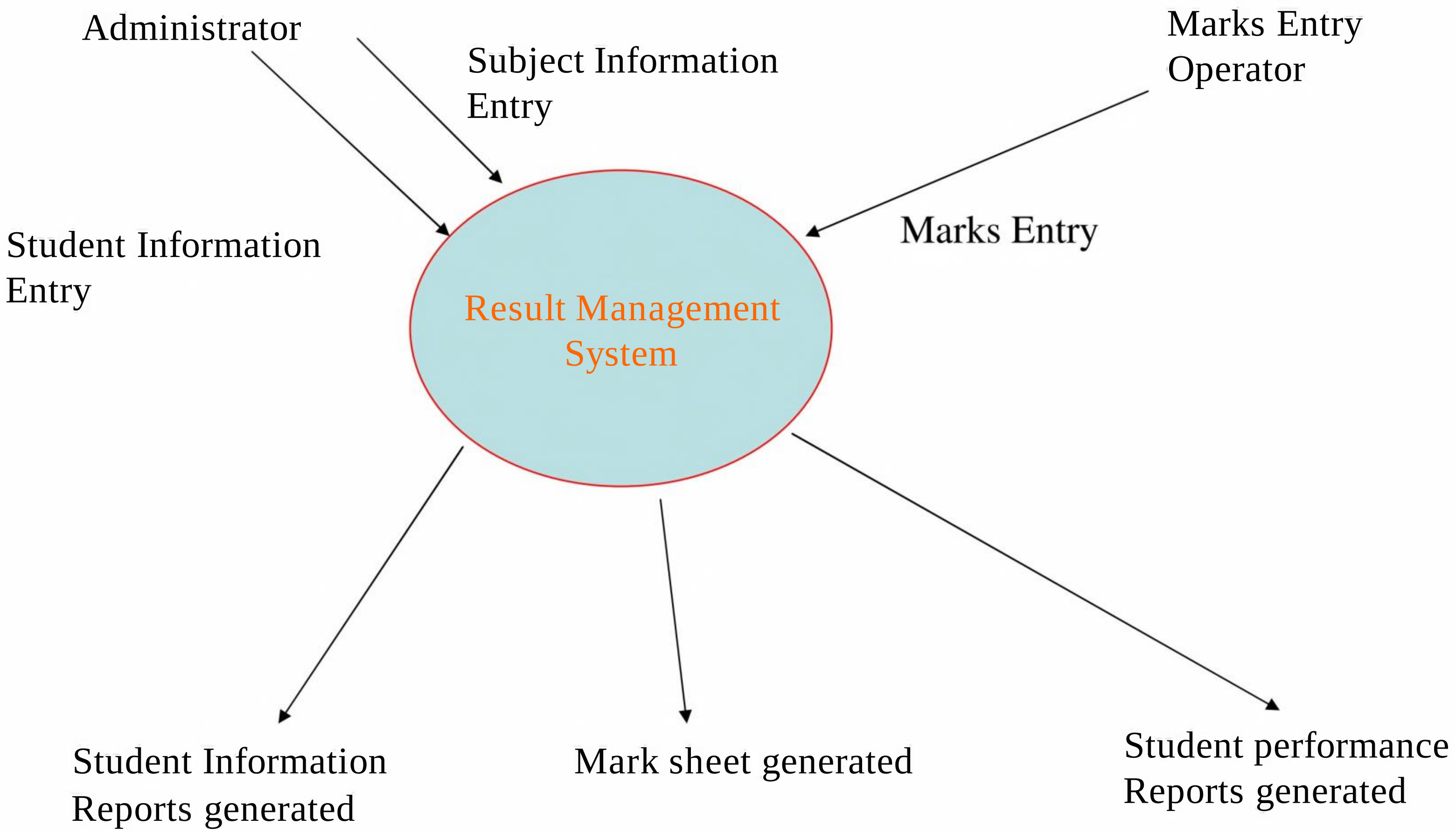
We analyze, refine and scrutinize requirements to make consistent & unambiguous requirements.

Steps



Requirements Analysis Steps

Requirements naysis


Administrator
Subject Information Entry
Marks Entry Operator
Student Information Entry
Marks Entry
Result Management System
Student Information Reports generated
Mark sheet generated
Student performance Reports generated

Requirements analysis

Data Flow Diagrams

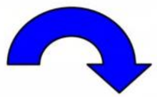
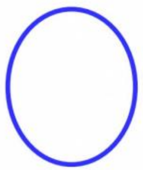
DFD show the flow of data through the system.

--All names should be unique

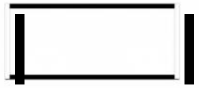
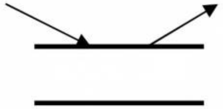
-- It is not a flow chart

-- Suppress logical decisions

-- Defer error conditions & handling until the end of the analysis

Symbol	Name	Function
	Data Flow	Connect process
	Process	Perform some transformation of its input data to yield output data.

Requirements naysis

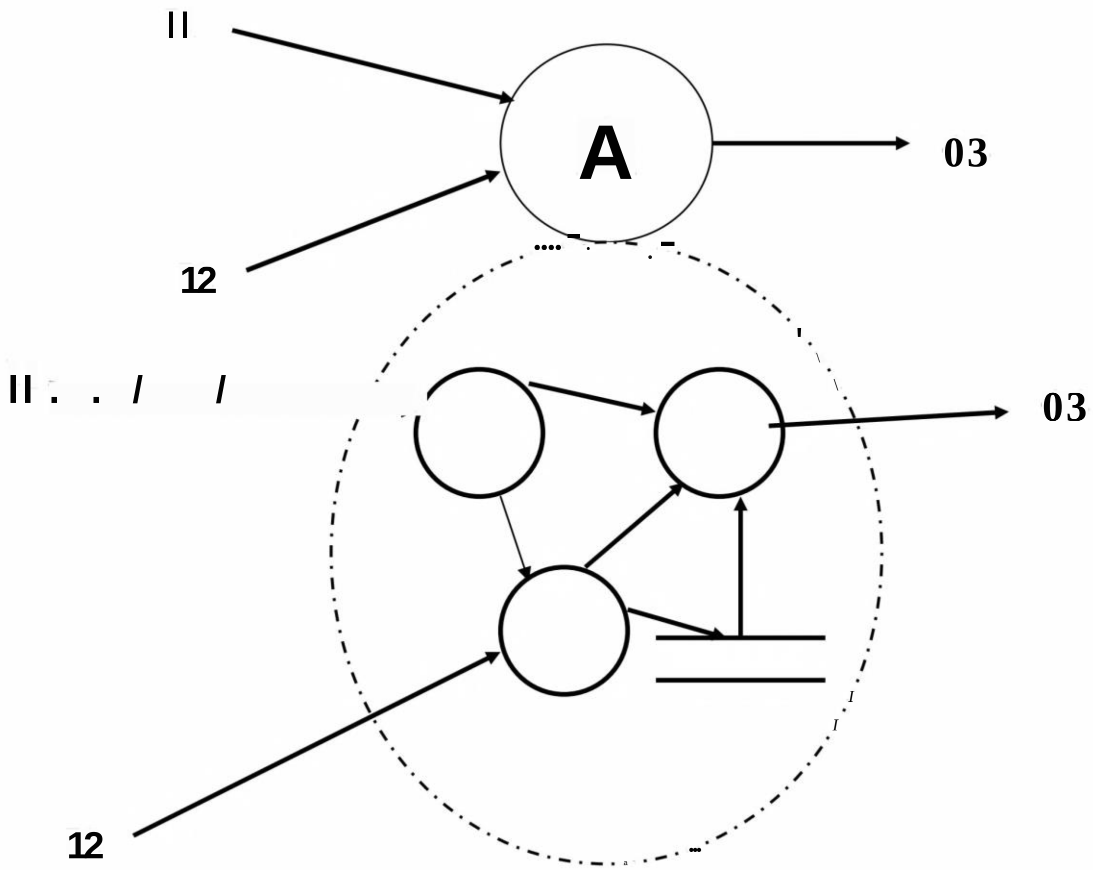
Symbol	Name	Function
	Source or sink	A source of system inputs or sink of system outputs
	Data Store	A repository of data, the arrowhead indicate net input and net outputs to store

Leveling

DFD represent a system or software at any level of abstraction.

A level O DFD is called fundamental system model or context model represents entire software element as a single bubble with input and output data indicating by incoming & outgoing arrows.

Requirements naysis



Data Dictionaries

DFD → DD

Data Dictionaries are simply repositories to store information about all data items defined in DFD.

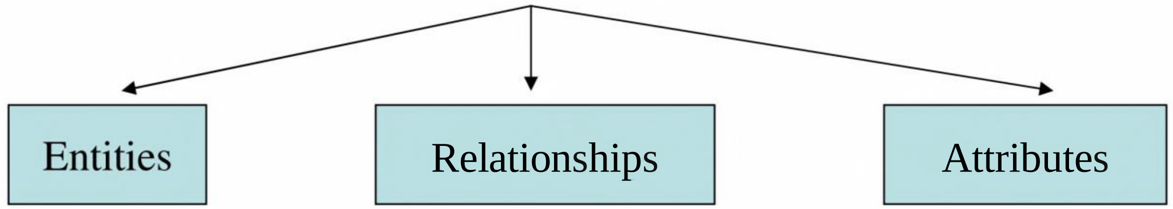
Includes :

Name of data item
Aliases (other names for items)
Description/Purpose
Related data items
Range of values
Data flows
Data structure definition

Entity-Relationsfiip Diagrams

Entity-Relationship Diagrams

It is a detailed logical representation of data for an organization and uses three main constructs.



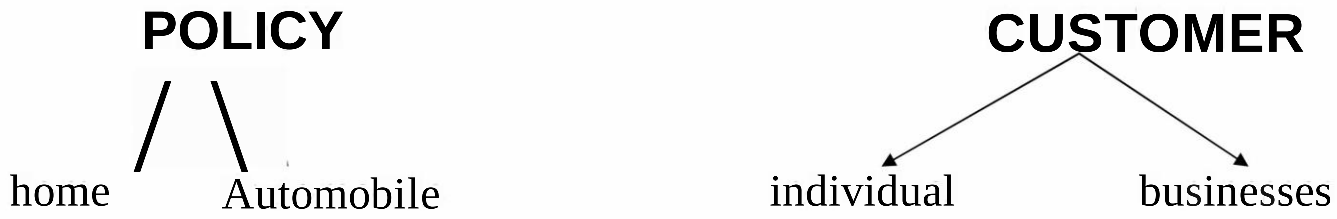
Entities

Fundamental thing about which data may be maintained. Each entity has its own identity.

Entity Type is the description of all entities to which a common definition and common relationships and attributes apply.

Entity-Relationsfiip Diagrams

Consider an insurance company that offers both home and automobile insurance policies .These policies are offered to individuals and businesses.



POLICY

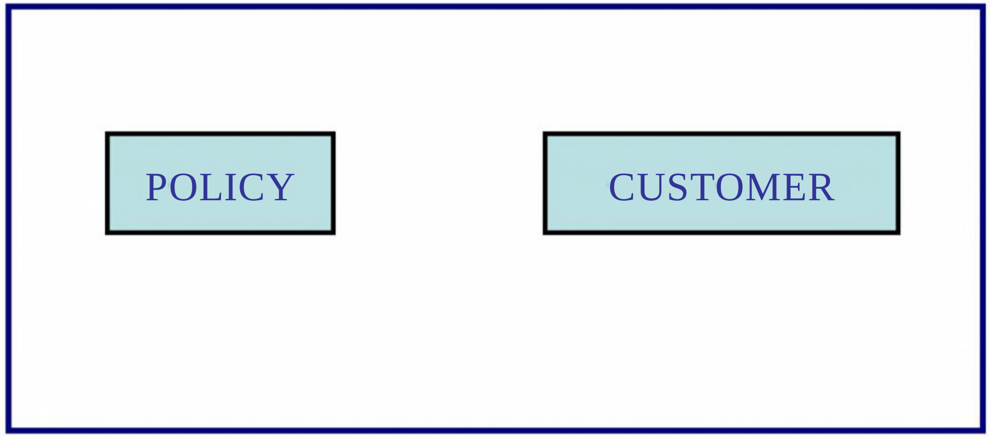
home

Automobile

CUSTOMER

individual

businesses



Entity-Relationsfiip Diagrams

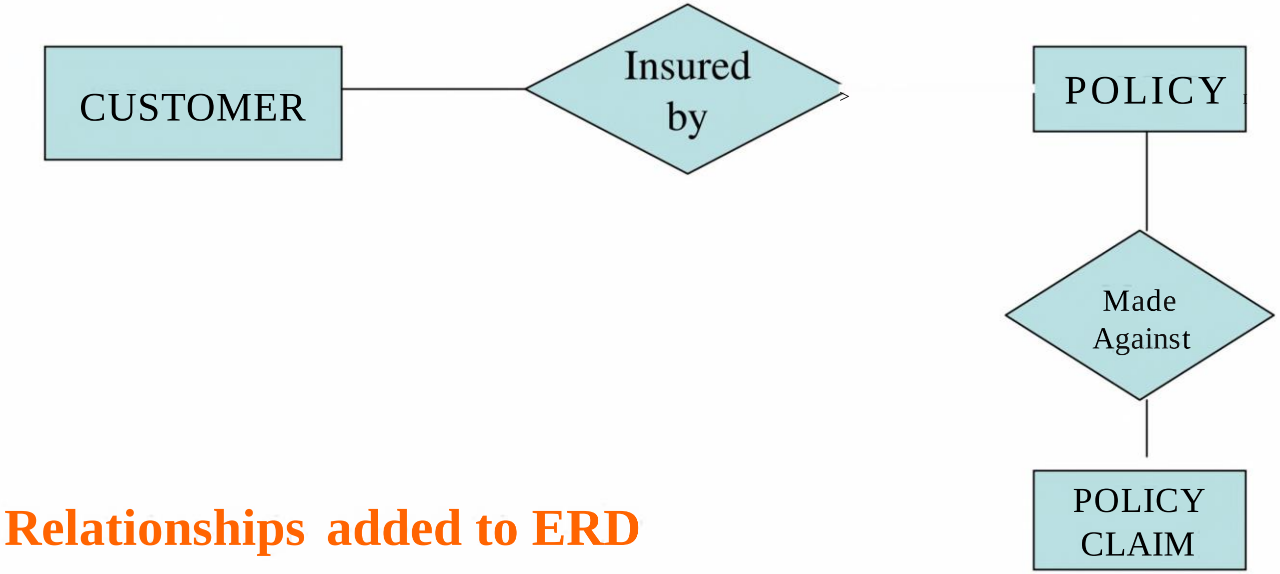
Relationships

A relationship is a reason for associating two entity types.

Binary relationships ________, involve two entity types

A CUSTOMER is insured by a POLICY. A POLICY CLAIM is made against a POLICY.

Relationships are represented by diamond notation in a ER diagram.

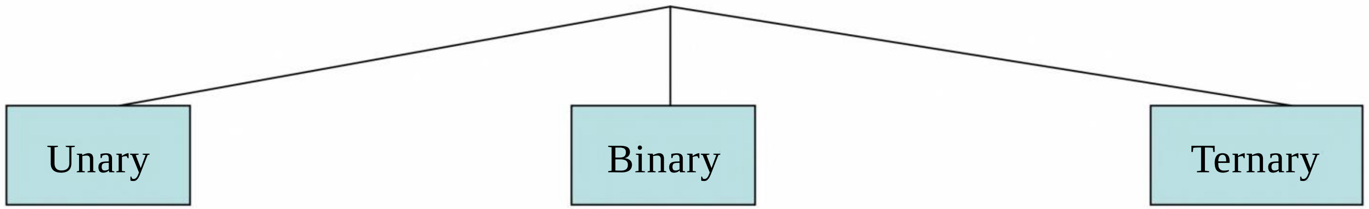


Relationships added to ERD

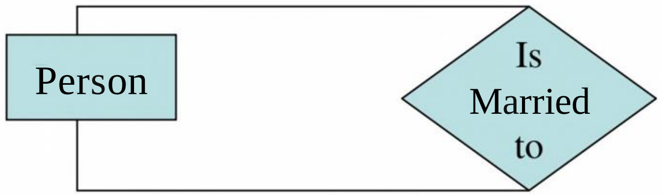
Entity-Relationship Diagrams

Degree of relationship

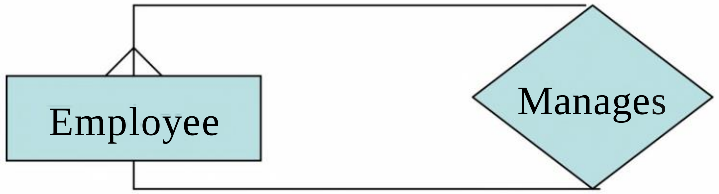
It is the number of entity types that participates in that relationship.



Unary relationship



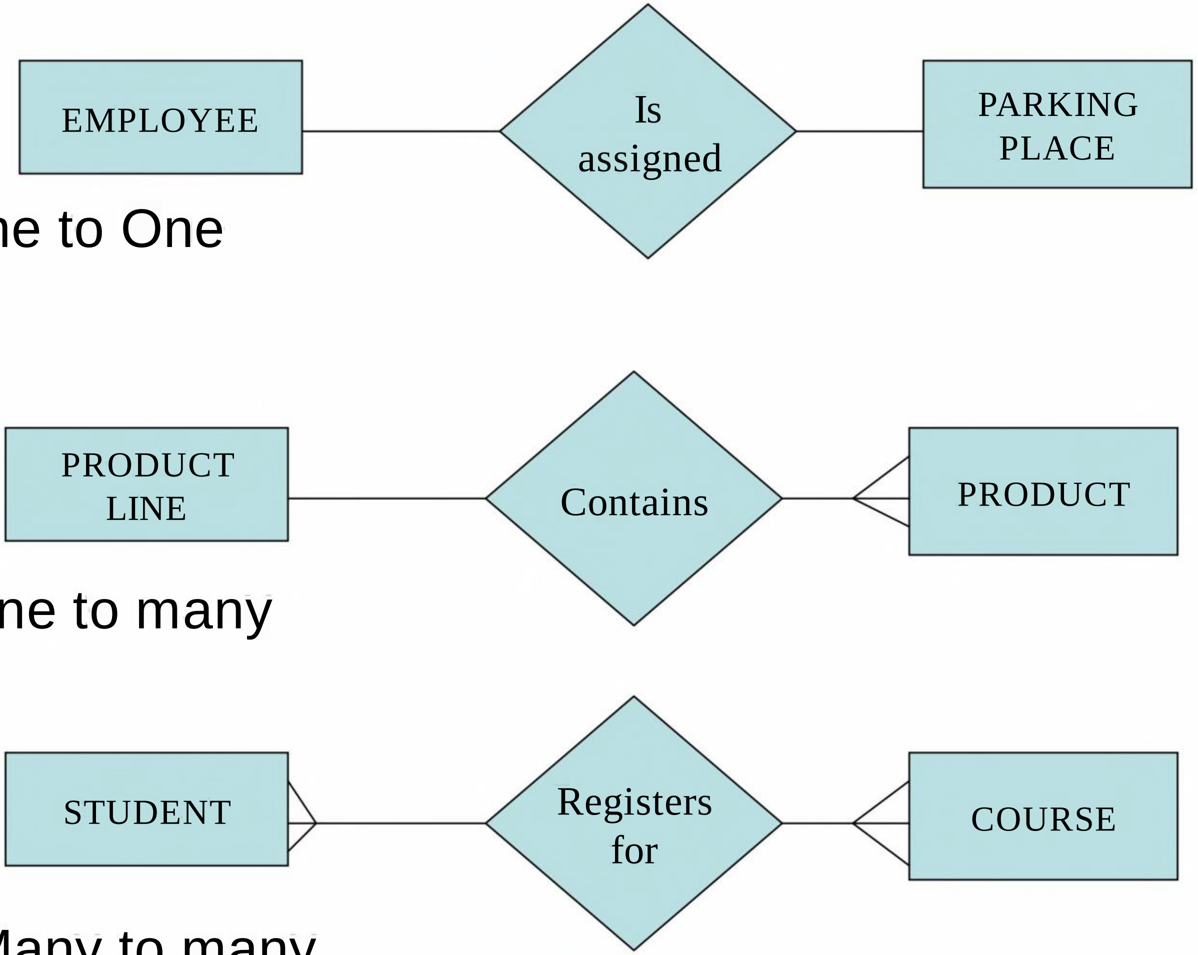
One to One



One to many

Entity-Relationship Diagrams

Binary Relationship



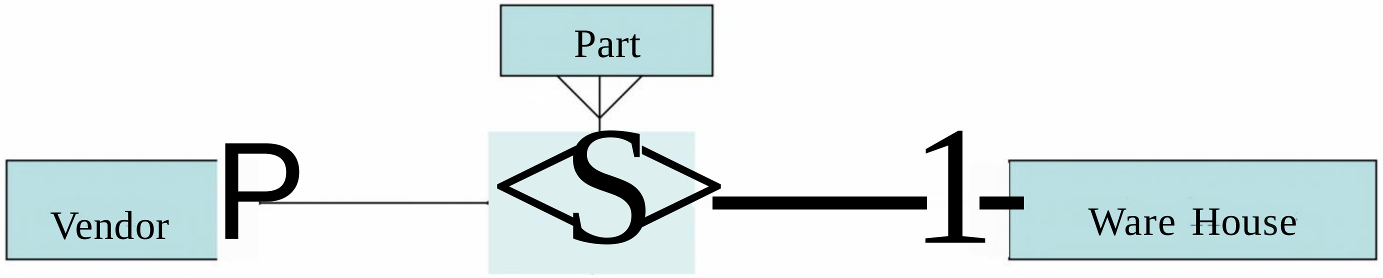
One to One

One to many

Many to many

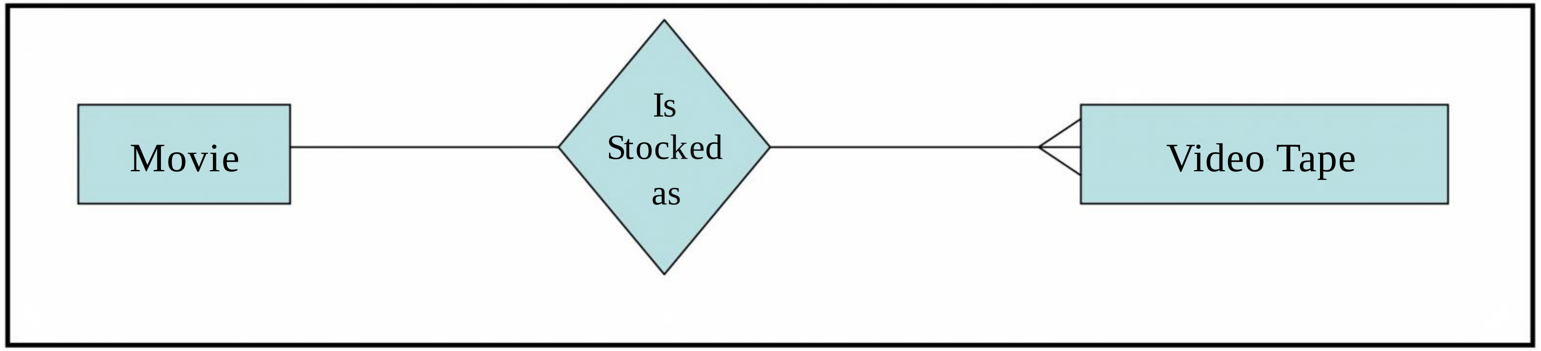
Entity-Relationsfiip Diagrams

Ternary relationship



Cardinalities and optionality

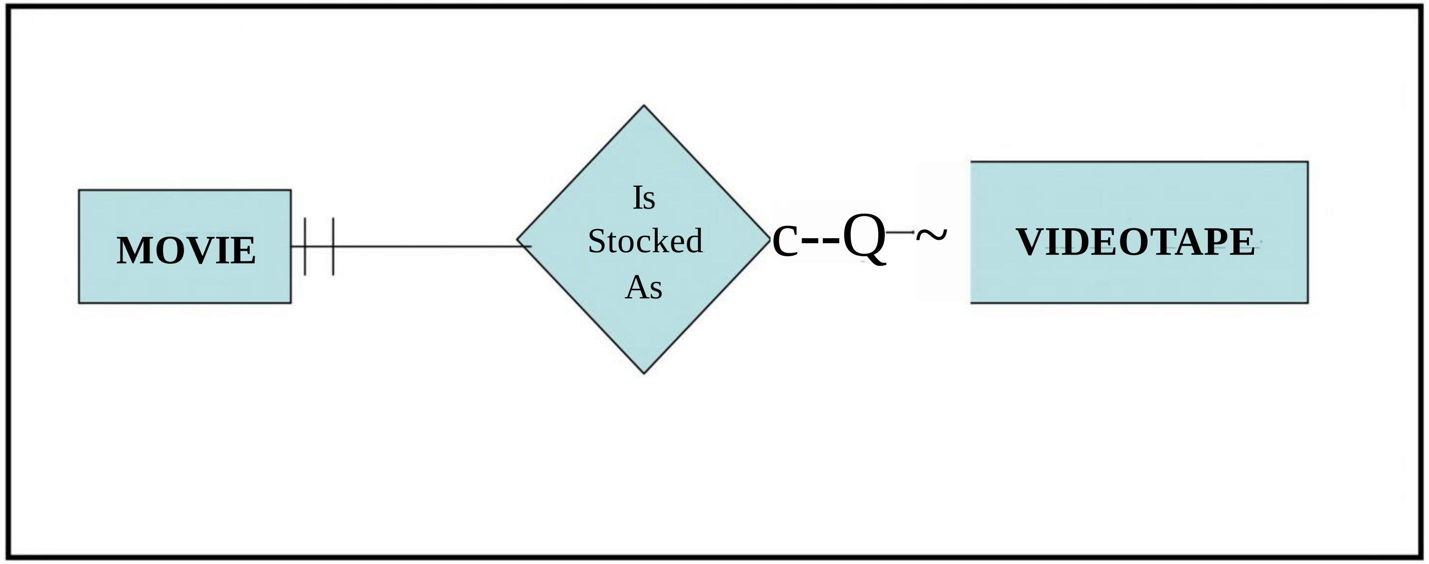
Two entity types A,B, connected by a relationship.
The cardinality of a relationship is the number of instances of entity B that can be associated with each instance of entity A



Entity Relationship Diagrams

Minimum cardinality is the minimum number of instances of entity B that may be associated with each instance of entity A.

Minimum no. of tapes available for a movie is zero.We say VIDEO TAPE is an optional participant in the is-stocked-as relationship.

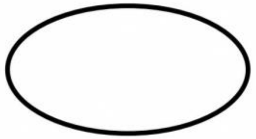


Entity-Relationsfiip Diagrams

Attributes

Each entity type has a set of attributes associated with it.

An attribute is a property or characteristic of an entity that is of interest to organization.



Attribute

Entity-Relationsfiip Diagrams

A candidate key is an attribute or combination of attributes that uniquely identifies each instance of an entity type.

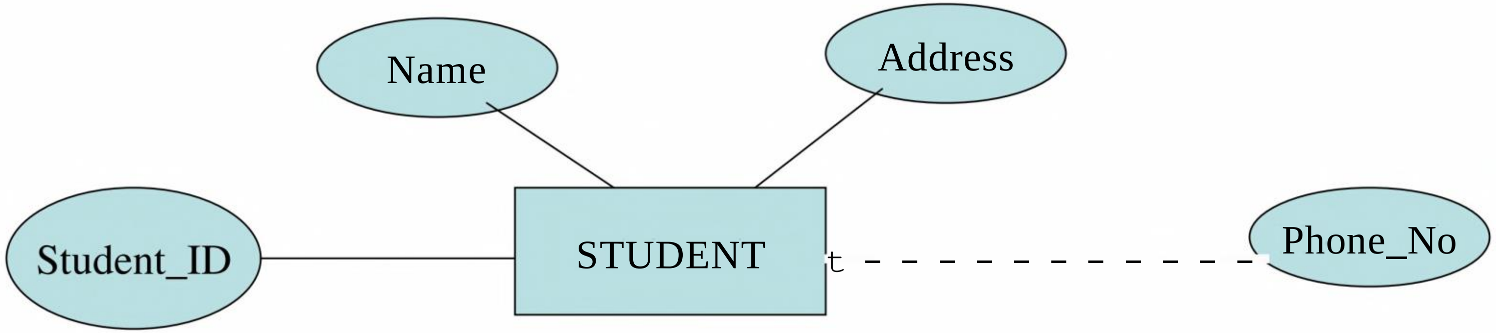
Student_ID → Candidate Key

If there are more candidate keys, one of the key may be chosen as the Identifier.

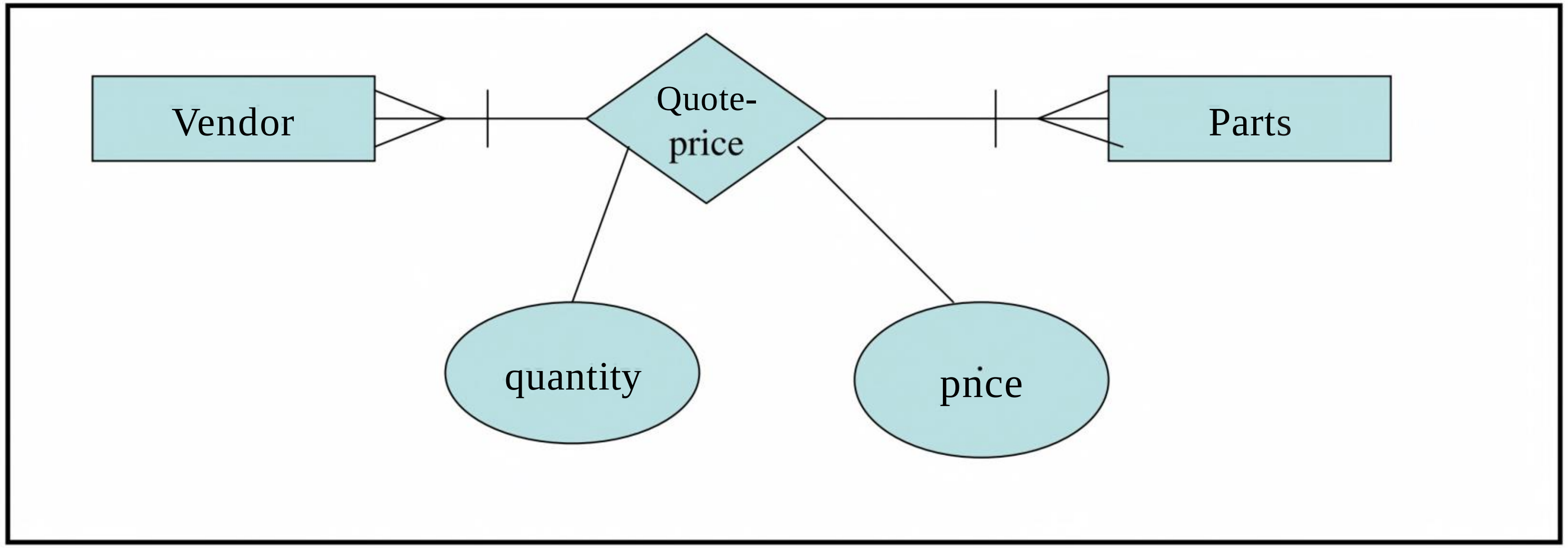
It is used as unique characteristic for an entity type.

Identifier

Entity-Relationsfiip Diagrams



Vendors quote prices for several parts along with quantity of parts.
Draw an E-RR diagram.



Approaches to problem analysis

1. List all inputs, outputs and functions.
2. List all functions and then list all inputs and outputs associated with each function.

Structured requirements definition (SRO)

Step1

Define a user level DFD. Record the inputs and outputs for each individual in a DFD.

Step2

Define a combined user level DFD.

Step3

Define application level DFD.

Step4

Define application level functions

Exercises

3.1 Discuss the significance and use of requirement engineering. What are the problems in the formulation of requirements?

3.2 Requirements analysis is unquestionably the most communication intensive step in the software engineering process. Why does the communication path frequently break down ?

3.3 What are crucial process steps of requirement engineering ? Discuss with the help of a diagram.

3.4 Discuss the present state of practices in requirement engineering. Suggest few steps to improve the present state of practice.

3.5 Explain the importance of requirements. How many types of requirements are possible and why ?

3.6 Describe the various steps of requirements engineering. Is it essential to follow these steps ?

3.7 What do you understand with the term "requirements elicitation" ? Discuss any two techniques in detail.

3.8 List out requirements elicitation techniques. Which one is most popular and why?

Exercises

3.9 Describe facilitated application specification technique (FAST) and compare this with brainstorming sessions.

3.10 Discuss quality function deployment technique of requirements elicitation. Why an importance or value factor is associated with every requirement ?

3.11. Explain the use case approach of requirements elicitation. What are use-case guidelines ?

3.12. What are components of a use case diagram. Explain their usage with the help of an example.

3.13. Consider the problem of library management system and design the following:

(i) Problem statement

(ii) Use case diagram

(iii) Use cases.

Exercises

3.14. Consider the problem of railway reservation system and design the following:

(i) Problem statement

(ii) Use case diagram

(iii) Use cases.

3.15. Explain why a many to many relationship is to be modeled as an associative entity ?

3.16. What are the linkages between data flow and E-R diagrams ?

3.17. What is the degree of a relationship ? Give an example of each of the relationship degree.

3.18. Explain the relationship between minimum cardinality and optional and mandatory participation.

3.19. An airline reservation is an association between a passenger, a flight, and a seat. Select a few

pertinent attributes for each of these entity types and represent a reservation in an E-R diagram.

Exercises

3.20. A department of computer science has usual resources and usual users for these resources. A software is to be developed so that resources are assigned without conflict. Draw a DFD specifying the above system.

3.21. Draw a DFD for result preparation automation system of B. Tech. courses (or MCA program) of any university. Clearly describe the working of the system. Also mention all assumptions made by you.

3.22. Write short notes on

(i) Data flow diagram

(ii) Data dictionary.

3.23. Draw a DFD for borrowing a book in a library which is explained below: "A borrower can borrow a book if it is available else he/she can reserve for the book if he/she so wishes. He/she can borrow a maximum of three books".

3.24. Draw the E-R diagram for a hotel reception desk management.

Explain why, for large software systems development, is it recommended that prototypes should be "throw-away" prototype ?