Source of Risk

➢**What is Risk?**

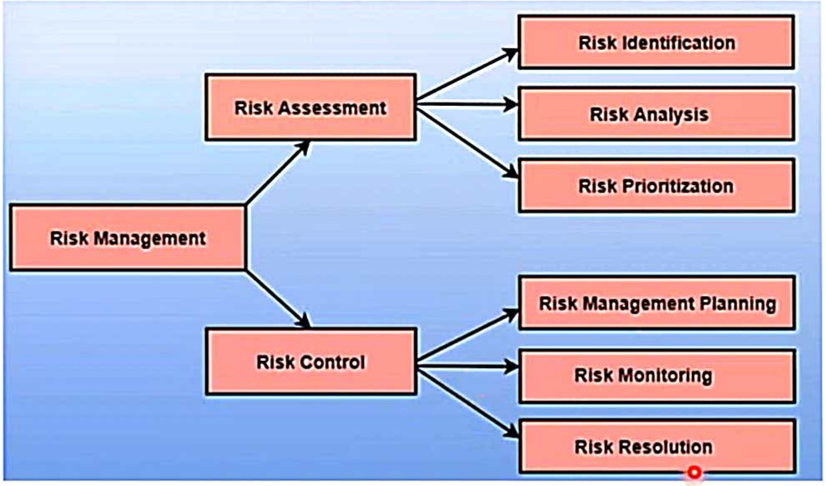
✓"Risk is an uncertain future event with a probability of occurrence and potential for loss"

➢**Source of Risk:**

1. Misunderstanding of customer requirements.
2. Uncontrolled & continuous changing of customer requirements.
3. Unrealistic promises given to the customers.
4. Misunderstanding of real impact of new methodologies.
5. Miscalculation of robustness & extensibility of software design.
6. Miscalculation of Team work & group effectiveness.
7. Wrong budget estimation.

Risk Management

- Risk Management is an important part of project planning activities.
- It involves identifying and estimating the probability of risks with their order of impact on the project.



Risk Identification

- The project organizer needs to find out risk in the project as early as possible.
- So, the impact of risk can be reduced by making effective risk management planning.
- By doing Brainstorming, SWOT Analysis, Casual Mapping & Flowchart methods are used.

➢ **There are different types of risks which can affect a software project:**

1. **Technology risks:** Software or Hardware technologies that are used to develop the system.
2. **People risks:** Risks that are connected with the person in the development team.
3. **Organizational risks:** Organizational environment where the software is being developed.
4. **Tools risks**: Software tools and other support software used to create the system.
5. **Requirement risks:** Changes to customer requirement & process of managing the requirements change.
6. **Estimation risks:** Management of estimation resources required to build the system

Risk Analysis & Projection or Prioritization

In Risk Analysis Process:

1. Identifying the problems causing risk in projects.
2. Identifying the probability of occurrence of problem.
3. Identifying the impact of problem.

➢**The probability of a risk can be categorized as:**

1. **Very Low (0-10%) :** Tolerable Risk (No harm)
2. **Low (10-25%) :** Low Risk (Minor effect)
3. **Moderate (25-50%) :** Medium Risk (Impact on Time)
4. **High (50-75%) :** High Risk (Impact on Time & Budget)
5. **Very high (+75%) : Intolerable Risk** (Impact on Output, Time, Budget & Performance)

Risk Control

Avoid
Accept
Risk
Transfer
Reduce

- It is the process of managing risks to achieve desired outcomes.

1. **Risk Planning:**

- The risk planning technique considers all of the significant risks that have been identified and develop strategies to mitigate them.

➢**There are three main methods to plan for risk management:**

1. **Avoid the risk:** Discussing with client to change requirements, Decrease scope of work, Giving incentives to engineers to avoid the risk of human resources turnover etc.
2. **Transfer the risk:** The risky element developed by third party, Buying insurance cover etc.
3. **Risk reduction:** This means planning method to include the loss due to risk. If there is a risk that some key personnel might leave, new recruitment can be planned.

Risk Control

2. Risk Monitoring:

- This is an ongoing process throughout the project and requires continuous evaluation and assessment of potential risks.
- Any changes in the assumptions made about risks should be identified and appropriate actions are taken to manage those risks.

3. Risk Resolution:

- This process ensures that the project stays on track and risks are controlled within acceptable levels.
- The effectiveness of risk resolution depends on the accuracy of risk identification, analysis, and planning of risk solving .
- It has ability to respond promptly and effectively to any issues that arise during the project.

Risk Mitigation

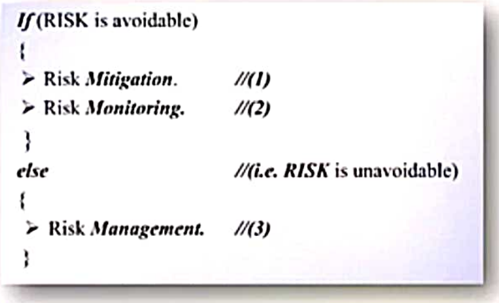
- Risk Mitigation is a technique for avoiding risks (Risk Avoidance).
- It is proactive approach. Apply before risk have generate.

➢**Steps for mitigating the risks as follows:**

1. Communicate with concerned staff to find off probable risk.
2. Removing all causes that are the reason for risk creation.
3. Develop a policy in an organization which will help to continue the project.
4. Controlling the corresponding documents from time to time.
5. Conducting timely reviews to speed up the work.

Risk Monitoring

- Risk monitoring is an activity used to track a project's progress.
- Performed by Project Manager.



```
If (RISK is avoidable)
{
 ➢ Risk Mitigation.          //(1)
 ➢ Risk Monitoring.          //(2)
}
else                         //(i.e. RISK is unavoidable)
{
 ➢ Risk Management.          //(3)
}
```

➢Objectives of Risk Monitoring Process:

1. To check if predicted risks occur or not.
2. To ensure proper application of risk avoidance are apply or not.
3. To gather information for future risk assessments.
4. To determine which risks generate which problems throughout the project.

Risk Management

- It is reactive approach, Apply after risk have generate.
- It assumes that the mitigation activity failed and the risk generate in reality.
- This task is done by Project manager, They will solve generated risk.
- If the project manager effectively uses project mitigation to remove risks successfully then it is easier to manage the risks

Example:

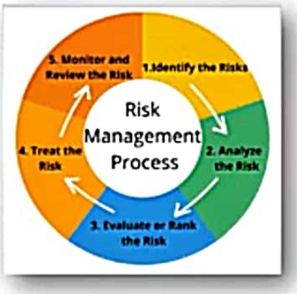
Consider a scenario that many people leaving the organization but

✓If sufficient additional staff is available

✓If current development activity knows to everybody in the team

✓If systematic documentation available

Then any new employee easily understand & start current development activity.



Software Quality is something which depends on lots of factors

Fulfill customer's requirements

- It should do what customer want from it

- It should provide all those feature for which software was developed

Looks Good…

- Software design should be good
- Attractive colors should be used
- Fonts should be attractive and easy to understandable

Maintainability…

- Software should be maintainable.
- If some area of software get defected working of other parts should not be disturbed too much.
- Software should have good fault tolerance.

Value for money…

- Ask a question to yourself “Is the software value for money”.
- Is it saving the time?
- Is it saving the money in any sense?
- Is it making the company/ Organization grow faster?

Quality Assurance

- It check if the implemented techniques and approaches are good to assure quality in software.

- It also final the approaches, techniques to assure the quality in software.

- It deals with the process use to develop the software and try to assure the quality by selecting the quality process, approach and technique.

- Its all about the process and techniques to assure the quality.

Quality Control

- Quality control is the process in we check we verify if we achieved the required quality by applying all the process are not.

- It is the process after the SQA process mostly.

- It also check if all the processes required to achieved and assure the quality are implemented correctly.

- It checks the product we got by following the approaches and try to fix the bugs and defects as well.

Software Measurements

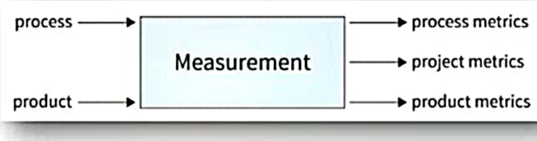
- Software Measurement is indicator of size, quantity amount or dimension of particular attribute of a product or process.
- It helps the project manager & entire software team to take decisions that lead to successful completion of the project by generating quantity result.

➢**Software measurements are of two categories:**

1. **Direct Measures:** It include software processes like Cost & Effort applied, Lines of code produced, Execution speed & Total no. of errors that have been reported.
2. **Indirect Measures:** It include products like Functionality, Quality, Complexity, Reliability, Maintainability and many more.

Software Metrics

- Software Metrics provide measures, functions or formulas for various aspects of software process & product.
- It including measuring software performance, planning work items, measuring productivity, and many other uses.



➢**Software metrics are of three categories:**

1. **Product Metrics:** It estimate Size, Complexity, Quality & Reliability of software.
2. **Process Metrics:** It estimate Faults rate during development, Pattern of testing defect arrival, Time it takes for a fixed operation.
3. **Project Metrics:** It estimate Number of software developers, Cost, Scheduling and Productivity of software.

Size Metrics : LOC

- Size-Oriented Metrics concentrates on the size of the program created.

1. LOC (Lines of Code):

- It is one of the earliest and simpler metrics for calculating the size of the computer program.
- It is generally used in calculating and comparing the productivity of programmers.
- LOC not count comment or blank line in code.

Example:

Total Line of Code (LOC) = 09

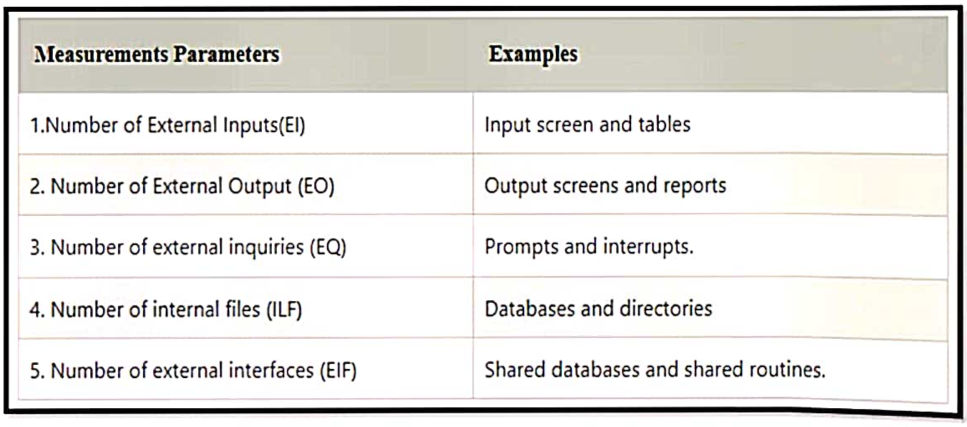
```
//Import header file
#include <iostream>
int main () {
  int num = 10;
//Logic of even number
        if (num % 2 == 0)
        {
          cout<<"It is even number";
        }
  return 0;
}
```

Size Metrics : FP

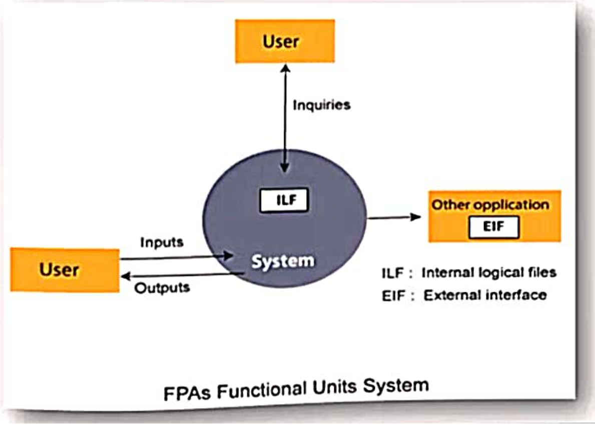
2. FP (Functional Point):

- Function-oriented software metrics measure functionality, what the system performs, is the measure of the system size.
- It find out by counting the number and types of functions used in the applications

FP Attributes



Measurements Parameters	Examples
1.Number of External Inputs(EI)	Input screen and tables
2. Number of External Output (EO)	Output screens and reports
3. Number of external inquiries (EQ)	Prompts and interrupts.
4. Number of internal files (ILF)	Databases and directories
5. Number of external interfaces (EIF)	Shared databases and shared routines.



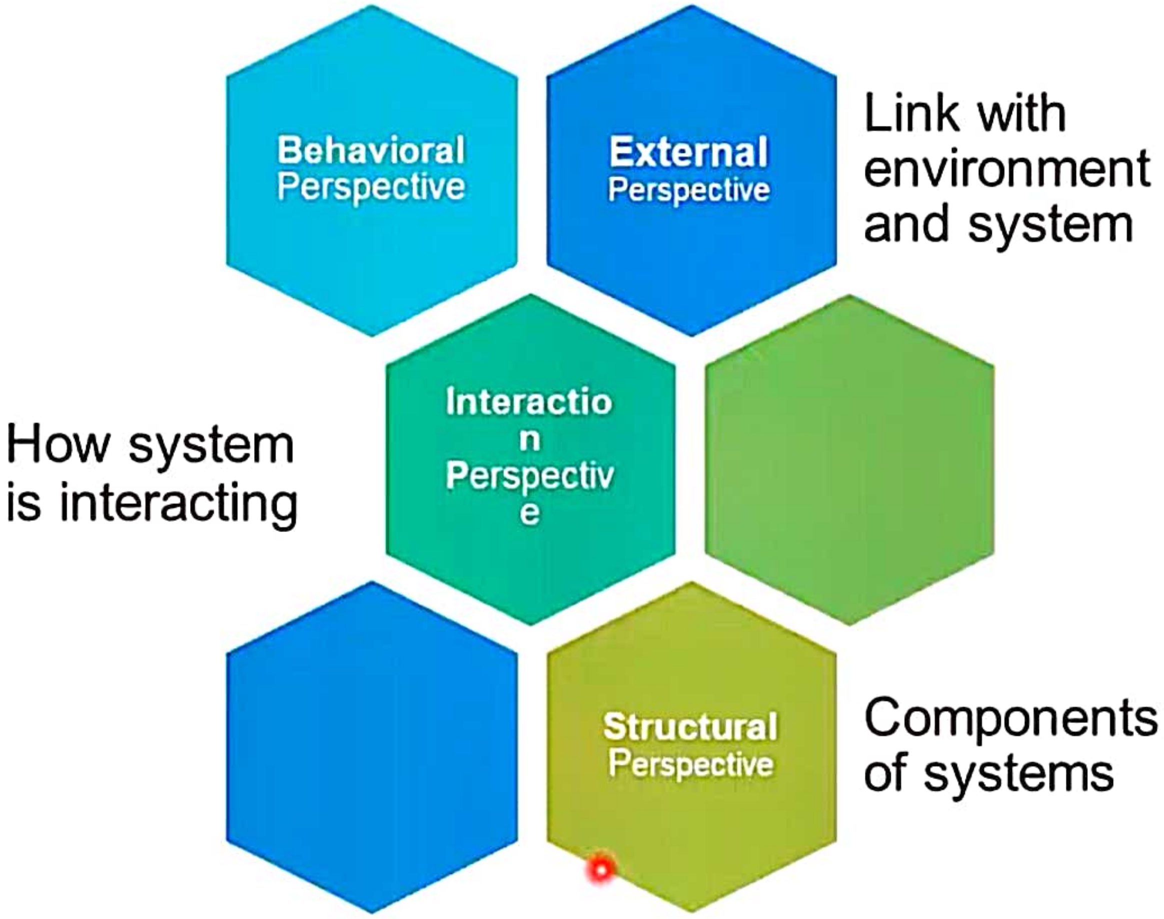
FPAs Functional Units System

System Modeling

- **System modeling** is the process of developing abstract models of a system,
- Each model presenting a different view or perspective of that system.
- Representing a system using some kind of graphical notation,
- Based on notations in the **Unified Modeling Language (UML)**.
- Models help the analyst to
  - Understand the functionality of the system;
  - Used to communicate with customers.

Why we need system modeling?

- System Modeling helps us to get different perspectives



Behavioral Perspective

External Perspective

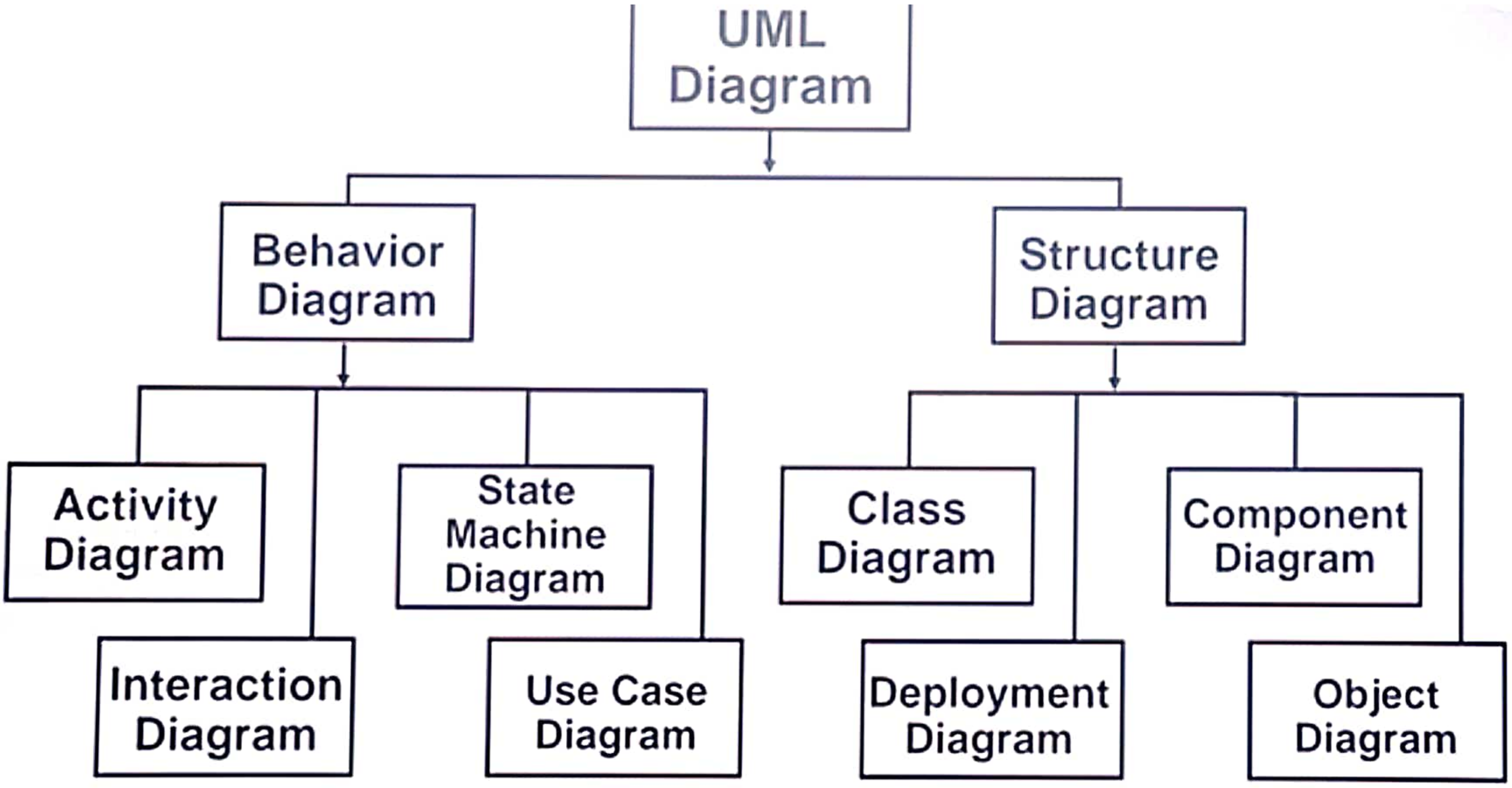
Link with environment and system

How system is interacting

Interactio n Perspectiv e

Structural Perspective

Components of systems



UML Diagram

Behavior Diagram

Structure Diagram

Activity Diagram

Interaction Diagram

State Machine Diagram

Use Case Diagram

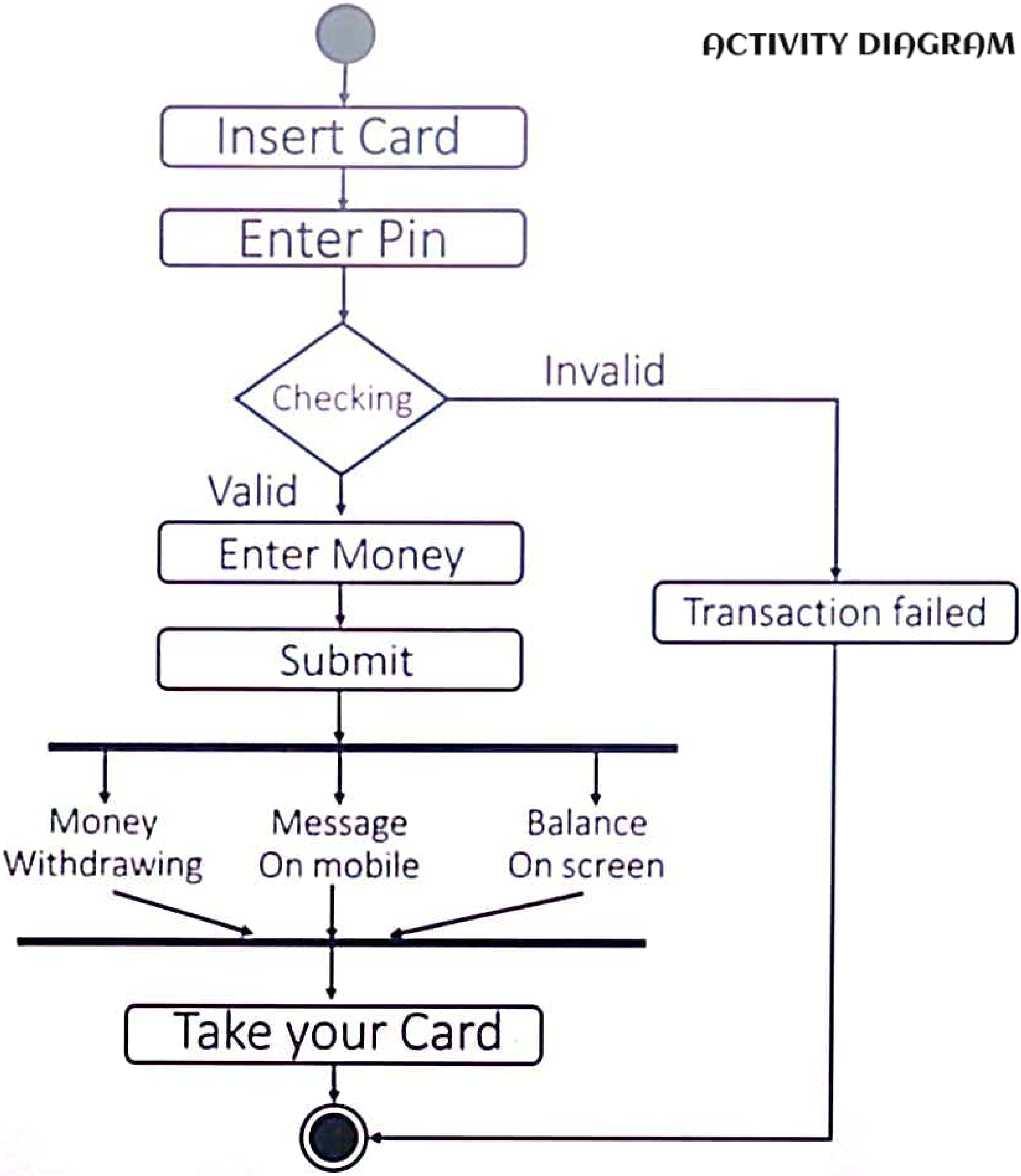
Class Diagram

Deployment Diagram

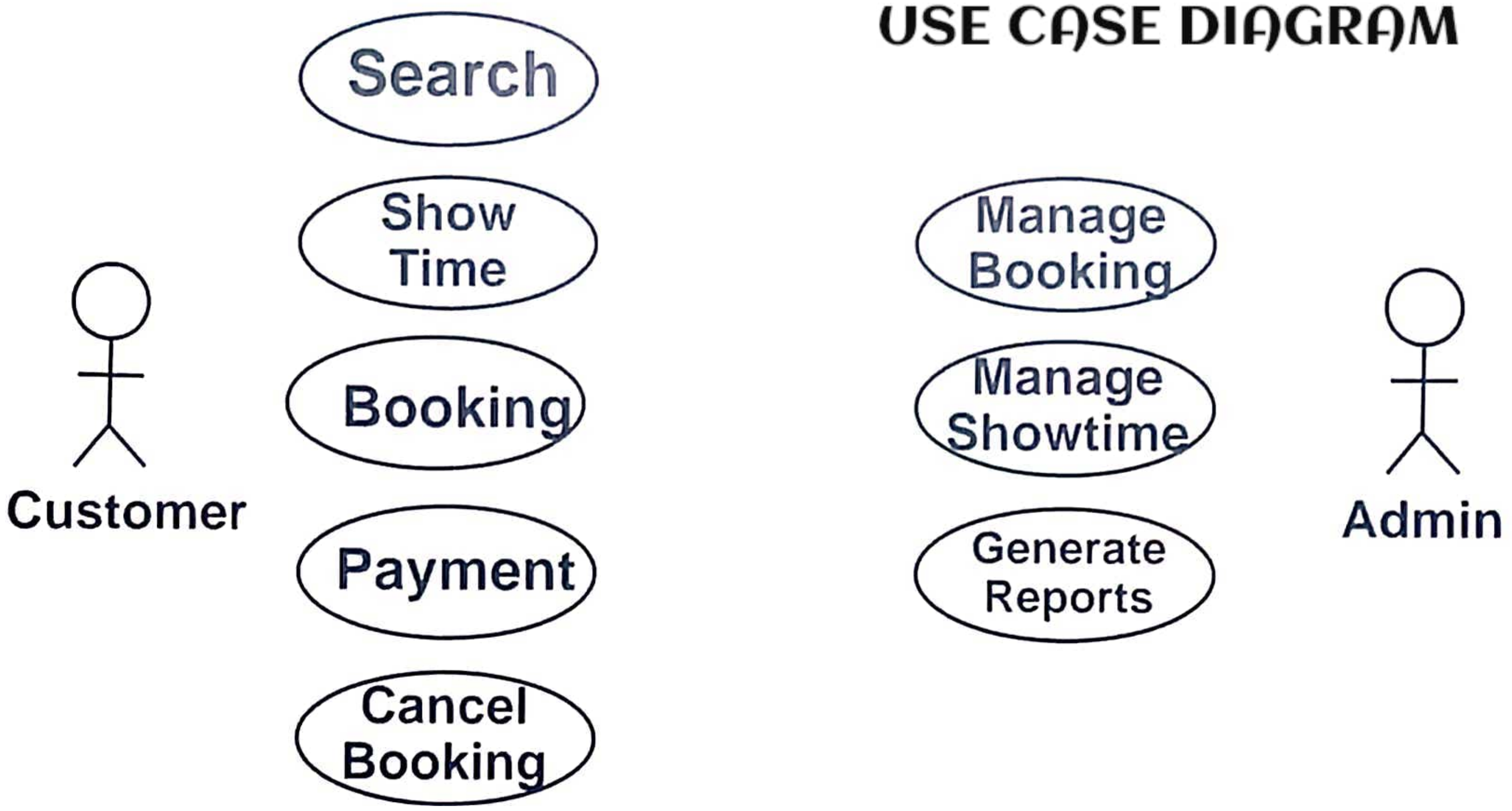
Component Diagram

Object Diagram

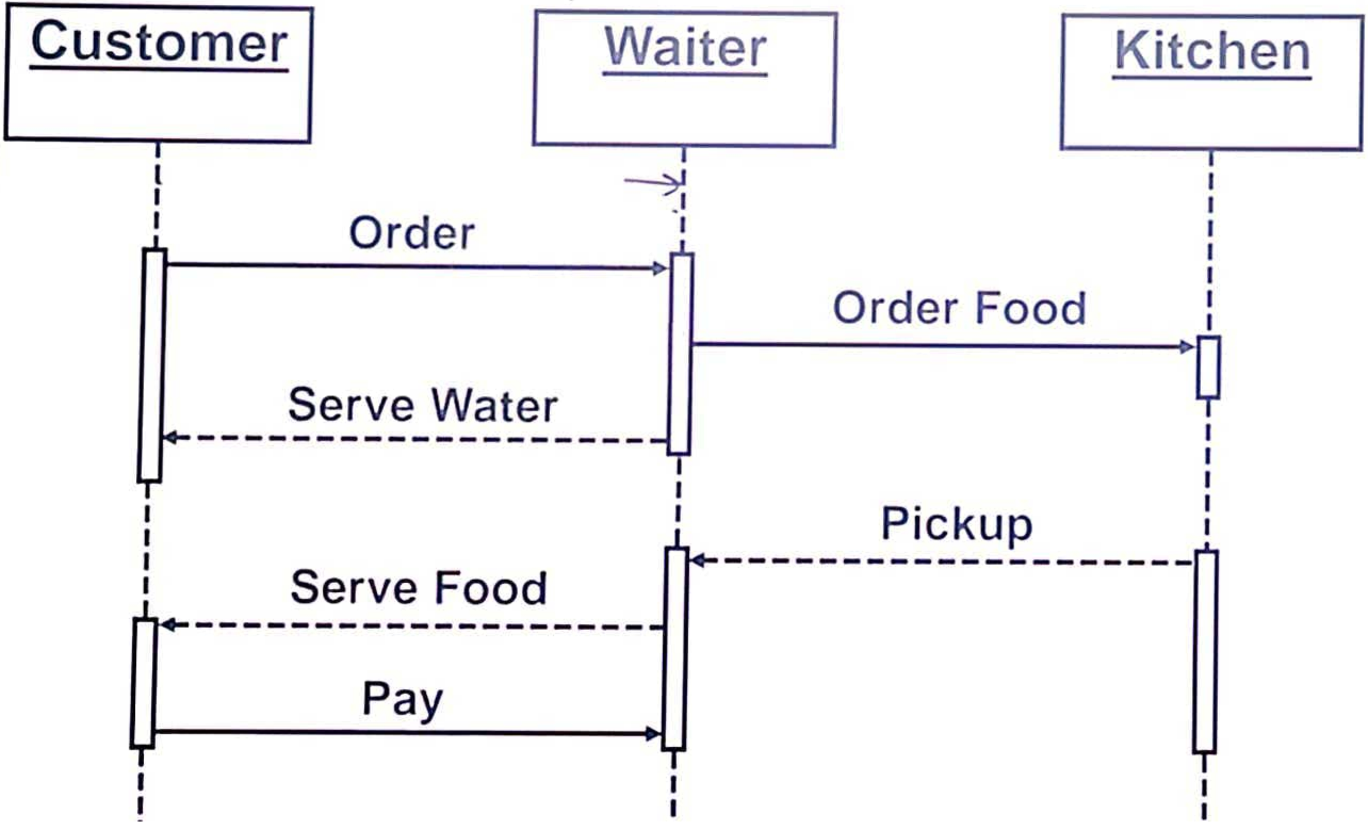
ACTIVITY DIAGRAM

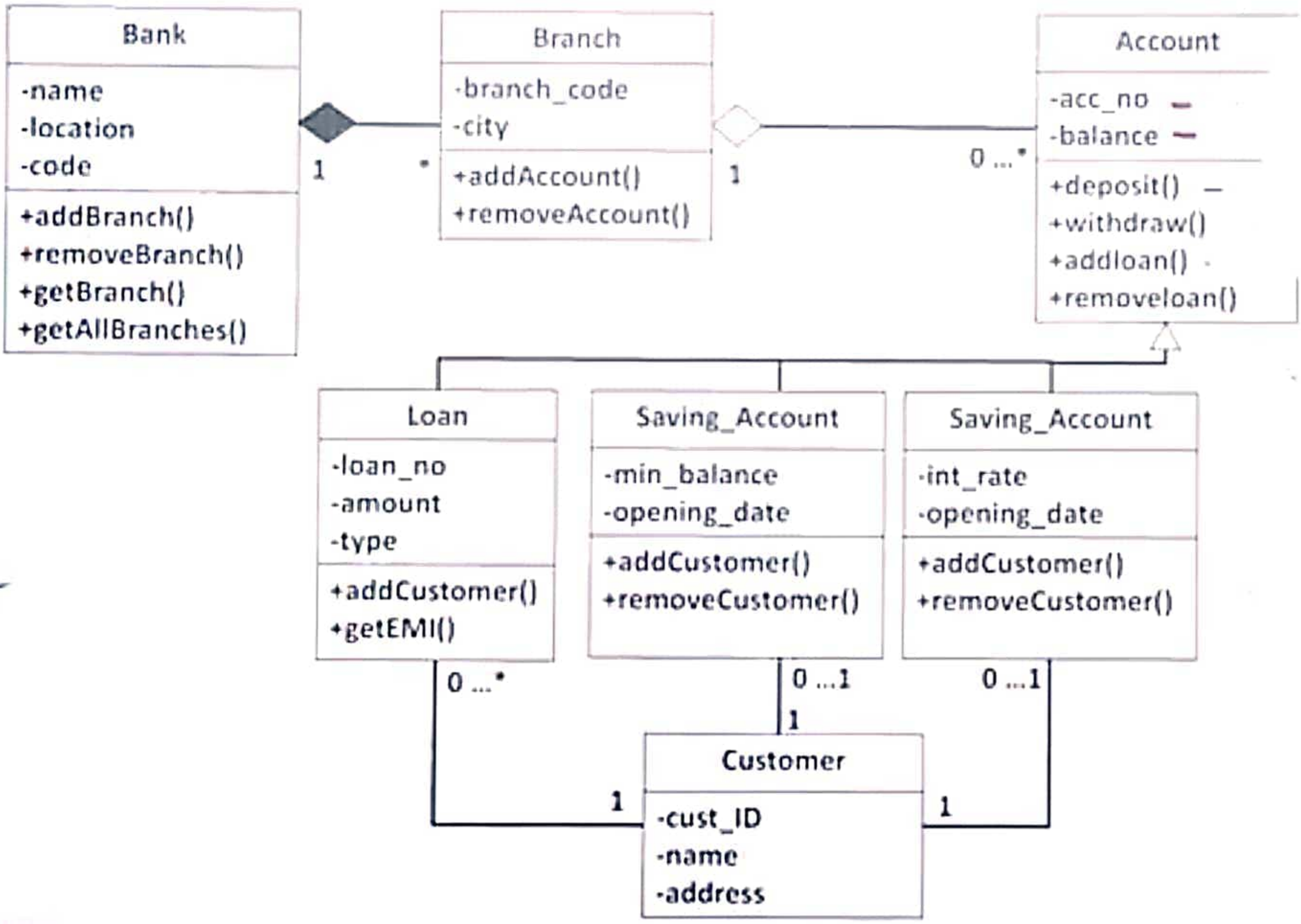


USE CASE DIAGRAM



SEQUENCE DIAGRAM




Bank
-name
-location
-code
+addBranch()
+removeBranch()
+getBranch()
+getAllBranches()
1
*
Branch
-branch_code
-city
+addAccount()
+removeAccount()
1
0 ...*
Account
-acc_no
-balance
+deposit()
+withdraw()
+addloan()
+removeloan()
Loan
-loan_no
-amount
-type
+addCustomer()
+getEMI()
Saving_Account
-min_balance
-opening_date
+addCustomer()
+removeCustomer()
Saving_Account
-int_rate
-opening_date
+addCustomer()
+removeCustomer()
0 ...*
0 ...1
0 ...1
1
1
1
Customer
-cust_ID
-name
-address

Behavioral Modeling

Behavioral model describes the interaction in the system. It represents the interaction among the structural diagrams. Behavioral modeling shows the dynamic nature of the system. They consist of the following –

- Activity diagrams
- Sequence Diagram
- Collaboration Diagram
- State-Chart Diagram
- Use case diagrams

All the above show the dynamic sequence of flow in a system.

Structural Modeling

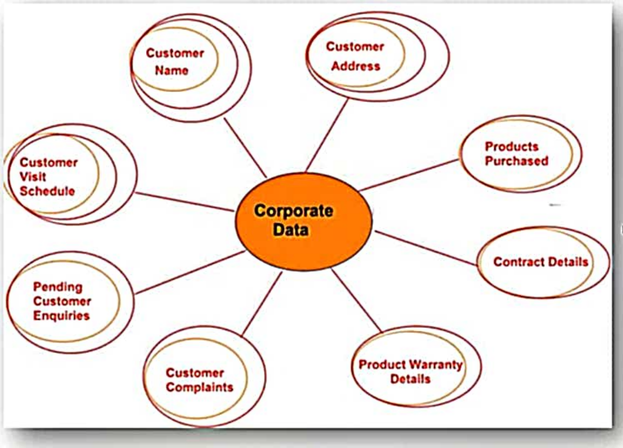
Structural modeling captures the static features of a system. They consist of the following –

- Classes diagrams
- Objects diagrams
- Component diagram
- Deployment diagrams

The structural model never describes the dynamic behavior of the system. Class diagram is the most widely used structural diagram.

What is Data Modeling?

- A data model is an abstract view of the data referred to in the product being developed.
- Data Modeling explaining & visualizing how the data be used by the software and defining data objects that will be stored in a database. **Example:**



Unstructured Data

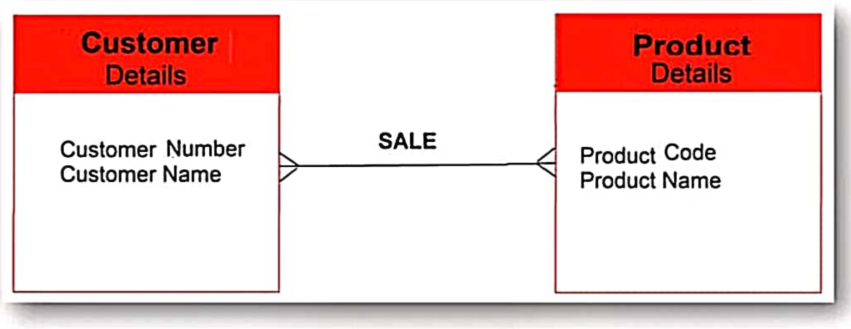
Customer Data	Product Data	Sale Data	Task Data
Customer id	Product Id	Customer Id	Task Id
Customer Name	Product Name	Product Id	Task Type (Sale,Complaint,Enquiry)
Address	Unit of Measure	Sale date	Schedule date
Contact No	Std Warranty period	Warranty upto	Task Status
Mail id	Features	Contract details	Allocated Resource Id
			Task detail

Structured Data

Types of Data Models

Type 1: Conceptual Data Model

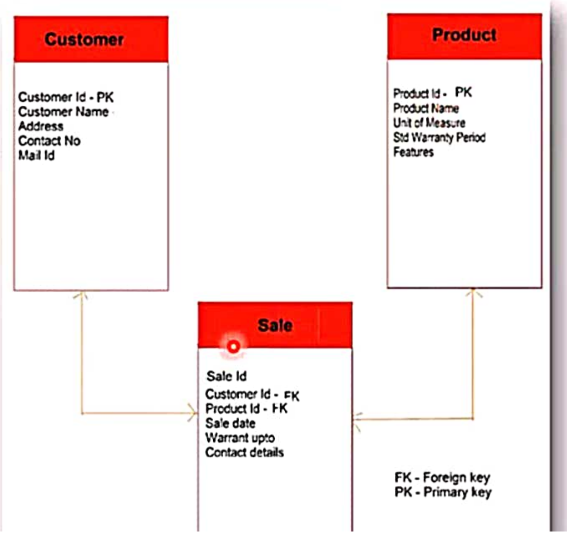
- It gives a front view of the of each data entity and not a technical detail of data.
- It presents all the **data entities** referred to by the business and their characteristics called **attributes** and the dependency or the connection between entities called **Relationships.**
- It makes communication easy and clear as all stakeholders & reducing communication gaps.



Types of Data Models

Type 2: Logical Data Model

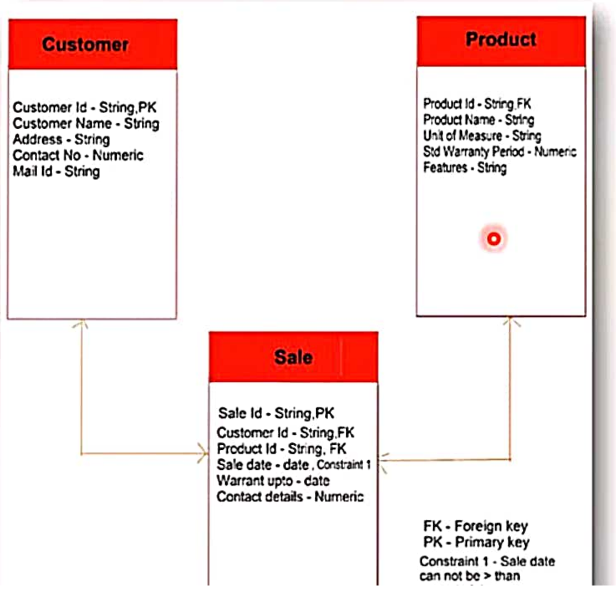
- It gives a detailed description of each data entity their attributes & relationship between two entities giving business purpose to each data.
- The relationship is presented more explicitly with details like Primary key, Foreign key & Parent-child dependent entity type.



Types of Data Models

Type 3: Physical Data Model

- A physical data model is the layout of the actual database with all its components.
- It gives a technical view of the data i.e., the table name, column name, data type, constraints, indexes, primary key, triggers, stored procedures, etc.
- It is developed by using databases SQL, ORACLE etc



Customer

Customer Id - String,PK
Customer Name - String
Address - String
Contact No - Numeric
Mail Id - String

Product

Product Id - String,FK
Product Name - String
Unit of Measure - String
Std Warranty Period - Numeric
Features - String

Sale

Sale Id - String,PK
Customer Id - String,FK
Product Id - String, FK
Sale date - date , Constraint 1
Warrant upto - date
Contact details - Numeric

FK - Foreign key
PK - Primary key
Constraint 1 - Sale date can not be > than