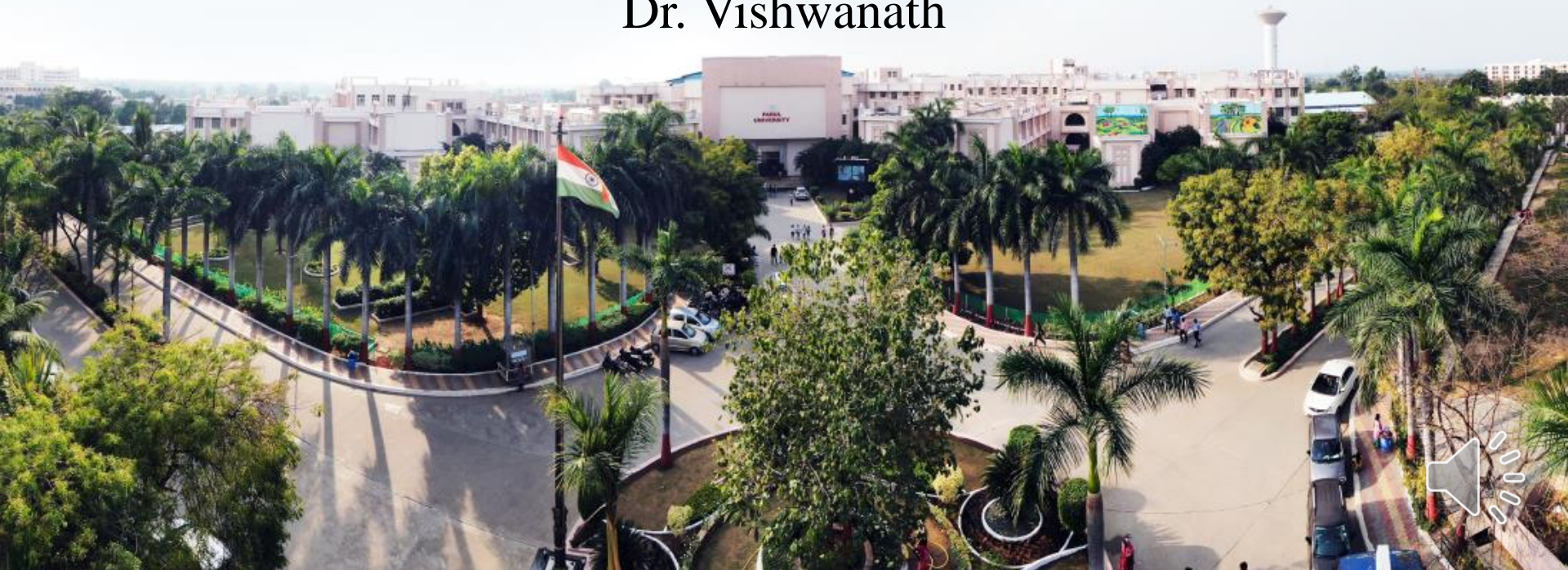


# Database Management System

## Unit 3 : Data Model

Computer Science & Engineering

Dr. Vishwanath



# Outline

- Data Models
- Hierarchical Model
- Network Model
- Relational Model
- Object Oriented Model.
- E-R Diagram
- Introduction to E-R Diagram
- Entities, Attributes & its types,
- Relationships,
- Mapping Cardinalities
- Participation Constraints
- Weak Entity Sets,
- Specialization
- Generalization
- Aggregation.

# What is Data Models?

Data models describe how a database's logical structure is represented.

Data models specify how data is linked to one another, as well as how stored within the system.

The most popular example of a database model is the relational model, which uses a table-based format.

# Type of Database Models

Hierarchical Model

Network Model

Entity-relationship Model

Relational Model

Object-oriented database Model

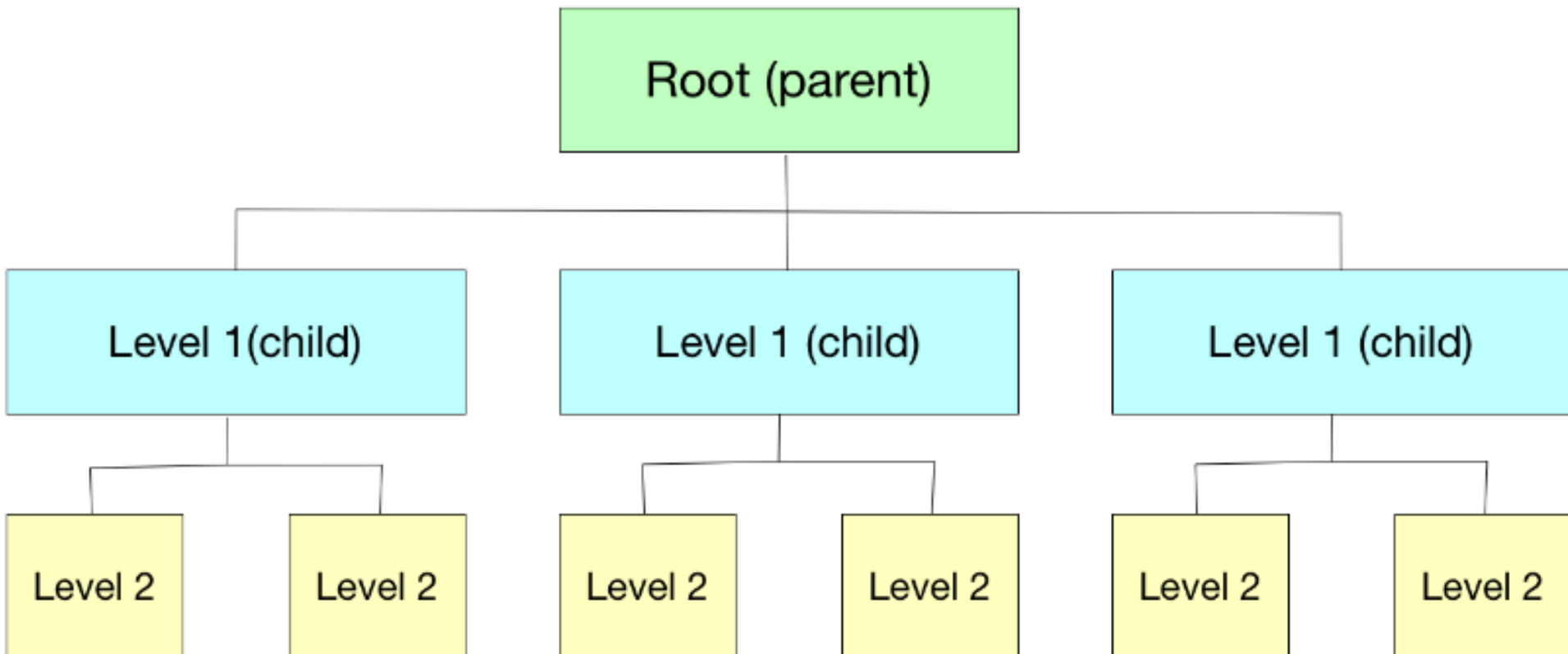
# Hierarchical Model

- The hierarchical model organizes data into a **tree-like structure**, where **each record has a single parent or root**.



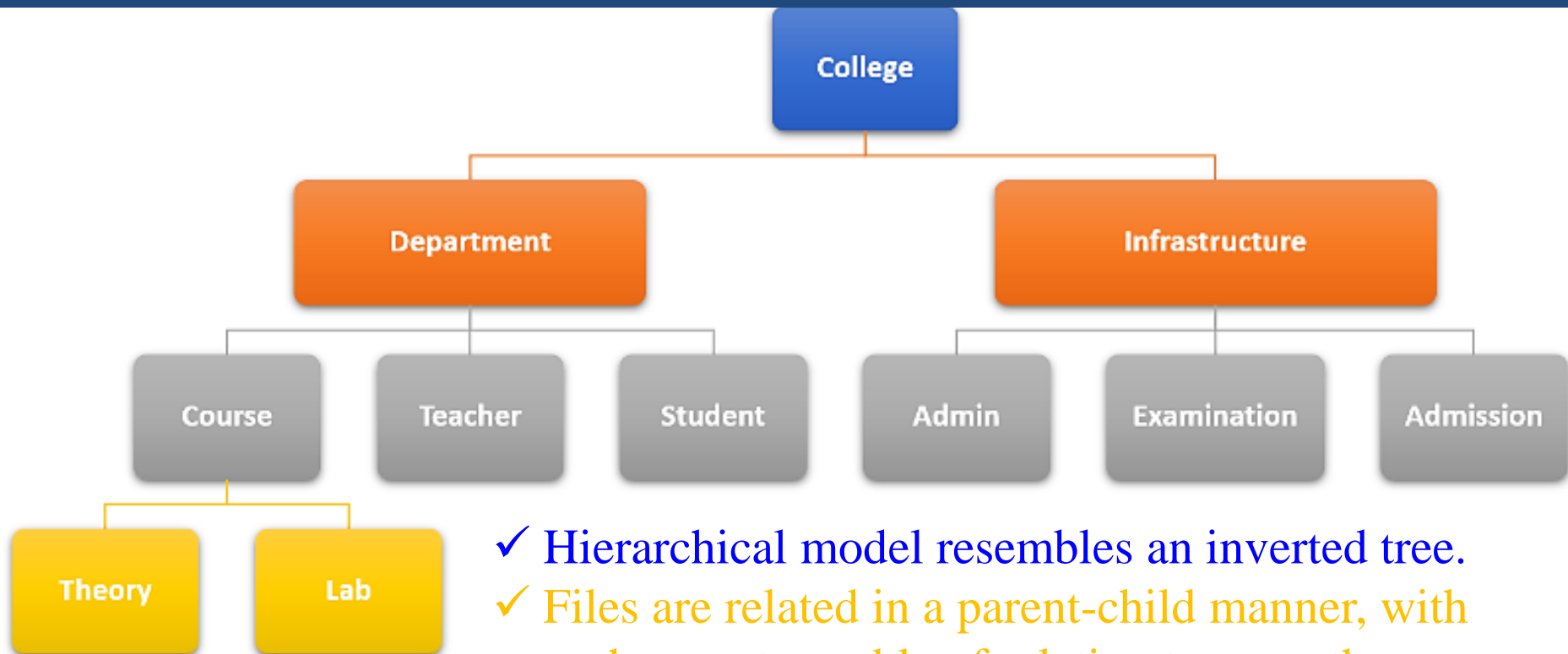
- The hierarchy **starts from the Root data**, and **expands like a tree**, adding **child nodes to the parent nodes**.
- In hierarchical model, data is organized into **tree-like structure** with **one-to-many relationship** between two different types of data, for example, **one department can have many professors and many students**.

# Hierarchical Model





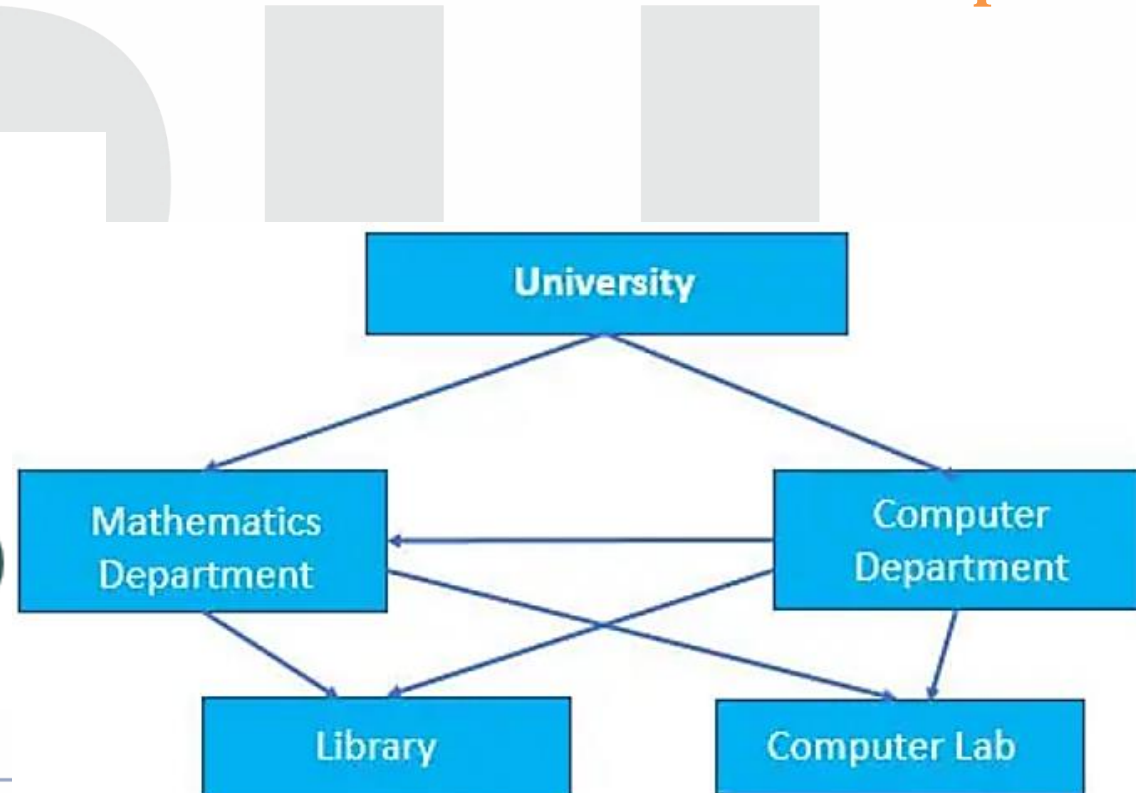
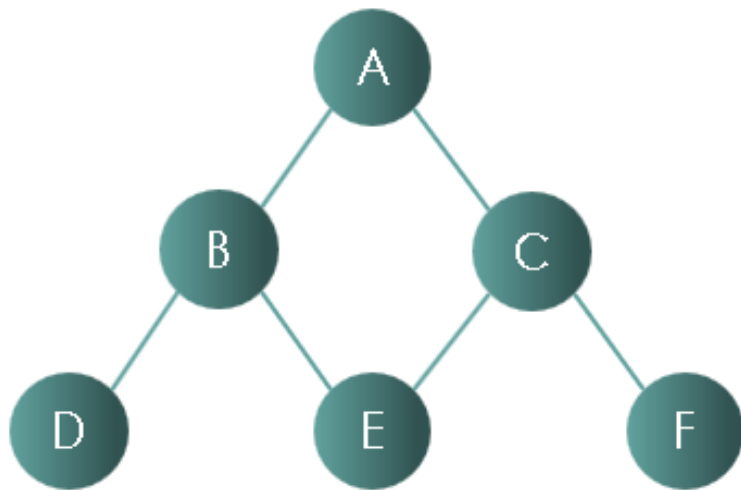
# Hierarchical Model



- ✓ Hierarchical model resembles an inverted tree.
- ✓ Files are related in a parent-child manner, with each parent capable of relating to more than one child, but each child only being related to one parent.

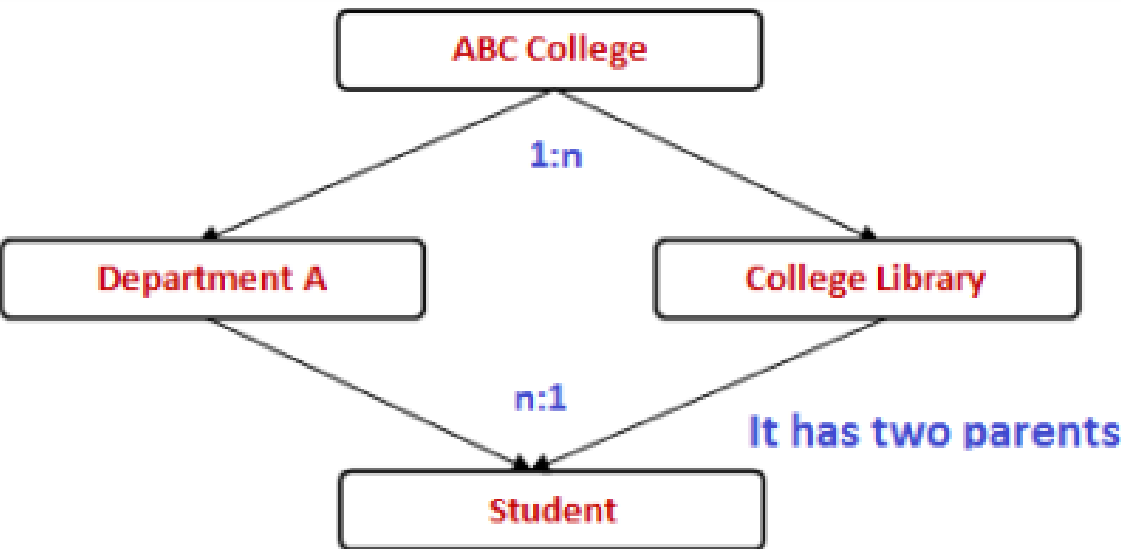
# Network Model

This is an **extension of the hierarchical model**, allowing **many-to-many relationships** in a tree-like structure that **allows multiple parents**.





# Network Model



ABC College has two Child.  
i.e. Department A and College library.

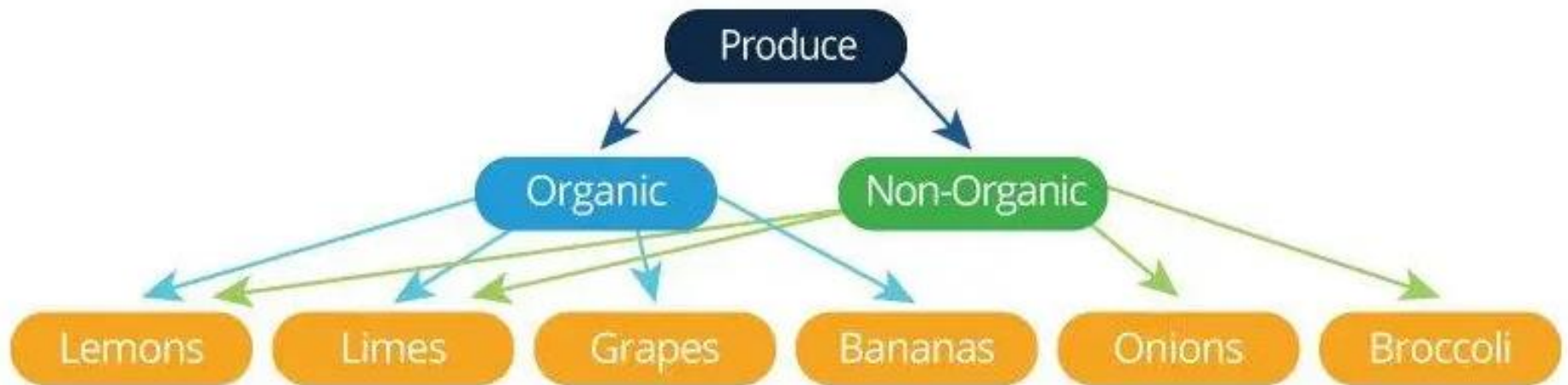
✓ It represents **one to many** relationship.

Even though there is no relation between Department A and College library, a student can be a member of both Department A and College library. This represents **many to one** relationship.

✓ Here, student has two parents which tell us, this is the Network data model.

# Network Model

## Network Database Model



# Relational Model

In this model, **data is organized in two-dimensional tables** and the **relationship is maintained by storing a common field**.

student_id	name	age
1	Akon	17
2	Bkon	18
3	Ckon	17
4	Dkon	18

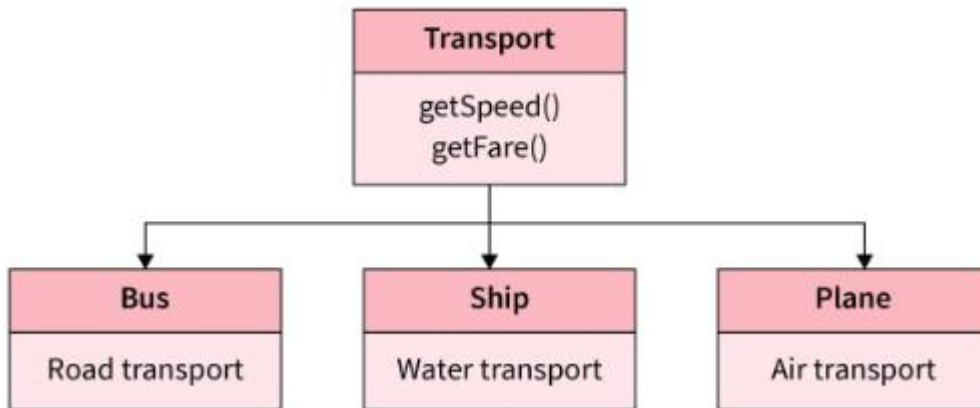
subject_id	name	teacher
1	Java	Mr. J
2	C++	Miss C
3	C#	Mr. C Hash
4	Php	Mr. P H P

student_id	subject_id	marks
1	1	98
1	2	78
2	1	76
3	2	88

# Object-oriented database Model

- Object Oriented Data Model represents the real world problems easily.
- In Object Oriented Data Model, data and their relationships are contained in a single structure.
- Here, we can store pictures, audio, video, and other types of data, which was previously impossible to store with the relational approach.

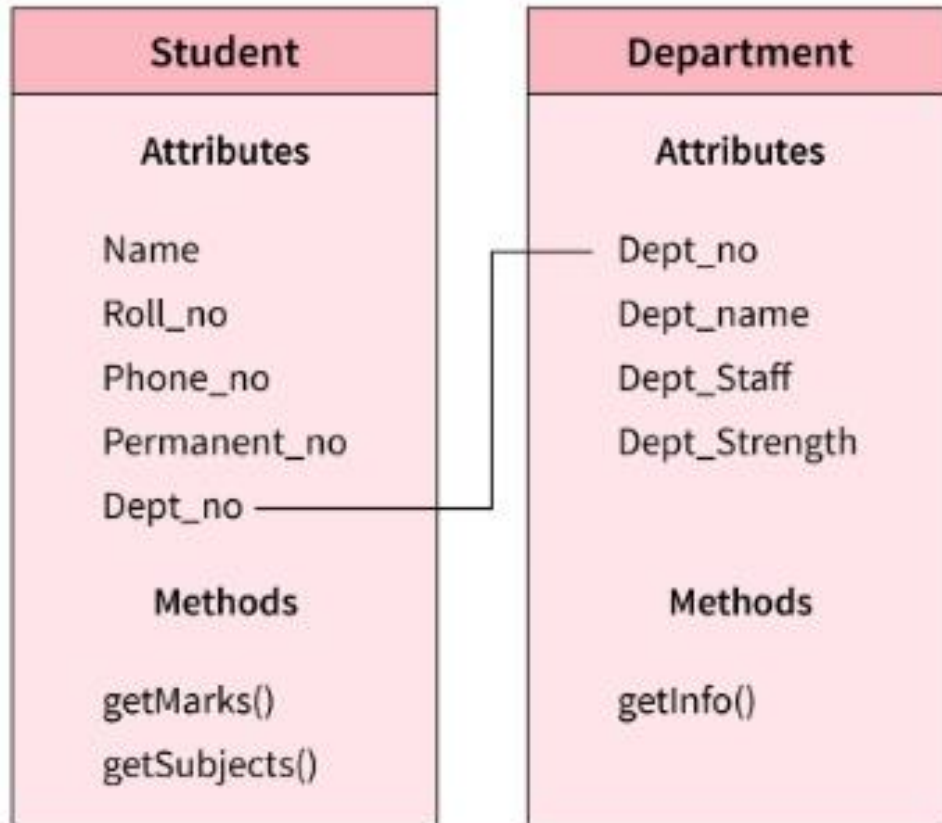
# Object-oriented database Model



- Here Transport, Bus, Ship, and Plane are objects.
- Bus has Road Transport as the attribute.

- Ship has Water Transport as the attribute.
- Plane has Air Transport as the attribute.
- The Transport object is the base object and the Bus, Ship, and Plane objects derive from it.

# Object-oriented database Model



Here, Student and Department are two different objects.

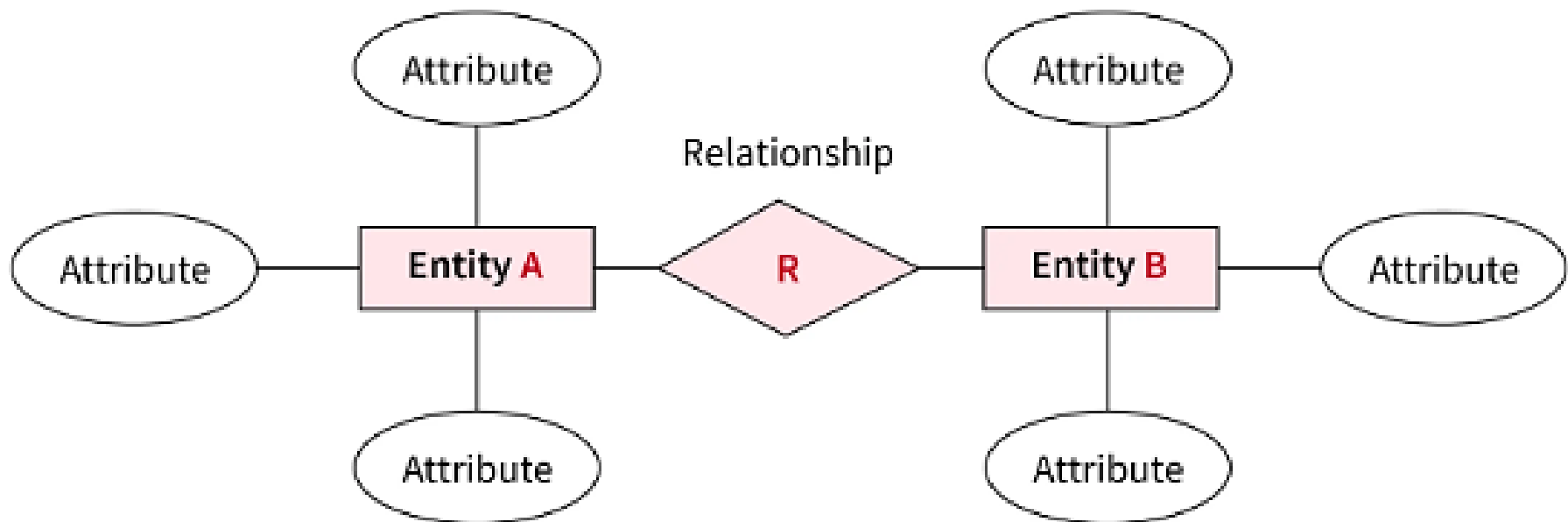
Each one of them has its attributes and methods.

They are linked by a common attribute Dept\_no which establishes a relationship between objects.

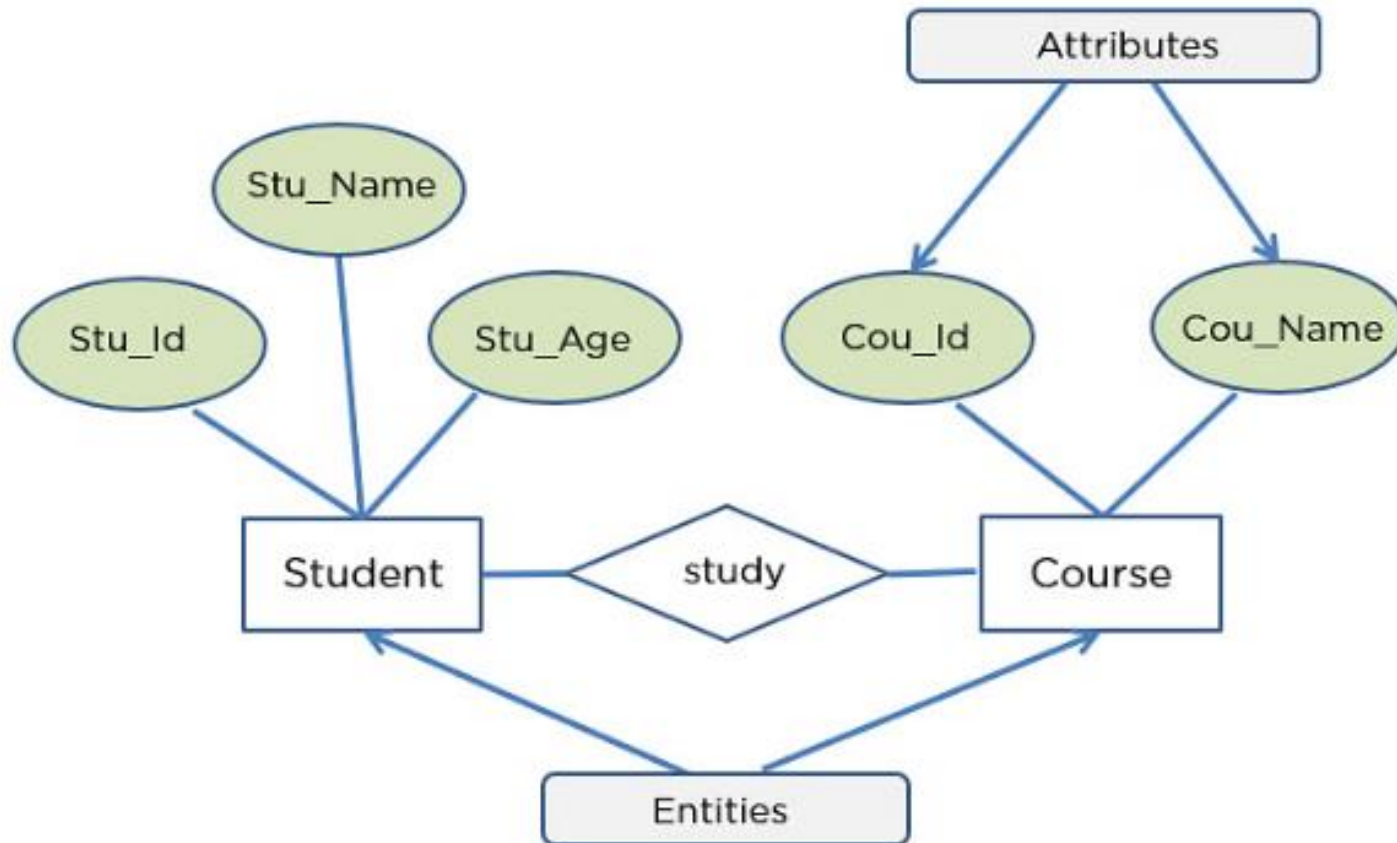


# Entity-relationship Model

- In this database model, **relationships are created by dividing object of interest into entity and its characteristics into attributes.**



# Entity-relationship Model

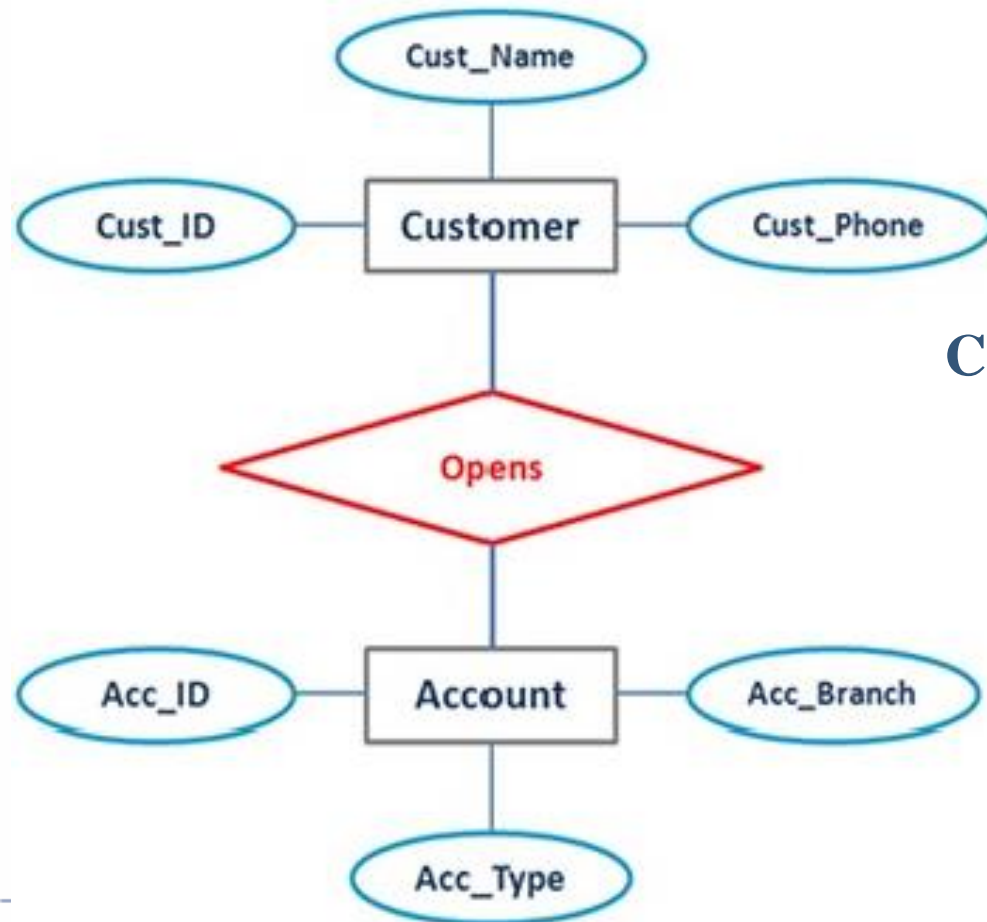


# Entity-relationship Model

It showcases the relationships developed between objects or entities in a system.

## Components of an entity-relationship

1. Entities
2. Attributes
3. Relationships



# Entity-relationship Model

- **Entities** - typically displayed in a rectangle.  
Entities can be represented by objects, persons, concepts, or events that contain data.
- **Attributes** - displayed in a circle or an oval.  
The attributes refer to the characteristics of an entity.
- **Relationships** - relationships among entities are represented in diamond.  
It illustrate; how two or more entities interact with each other.

# E-R Diagram

## What is E-R diagram?

- **E-R diagram:** (Entity-Relationship diagram)
- It is **graphical (pictorial) representation** of database.
- It uses different types of **symbols** to represent different **objects** of database.

# Entity

- An entity is a **person**, a **place** or an **object**.
- An entity is represented by a **rectangle** which contains the name of an entity.
- Entities of a college database are:
  - Student
  - Professor/Faculty
  - Course
  - Department
  - Result
  - Class
  - Subject



Symbol





# Entity Set

- It is a **set (group) of entities** of same type.
- Examples:
  - All persons having an account in a bank
  - All the students studying in a college
  - All the professors working in a college
  - Set of all accounts in a bank

E1

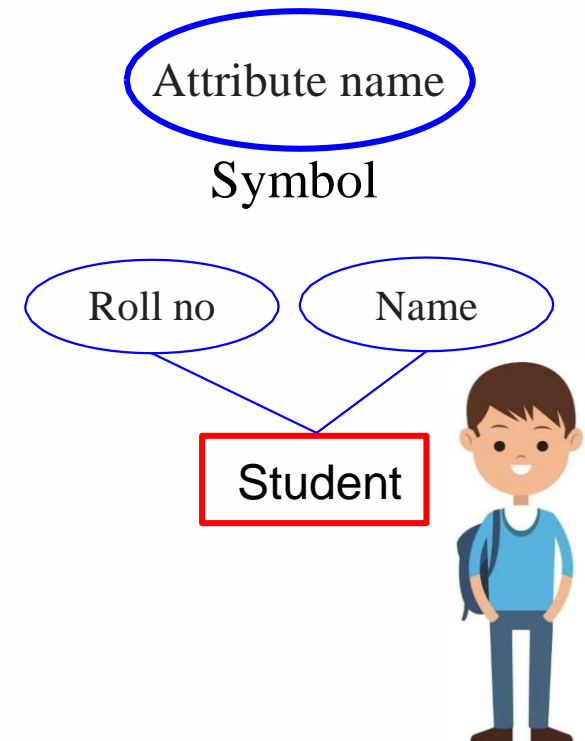
E2

E3



# Attributes

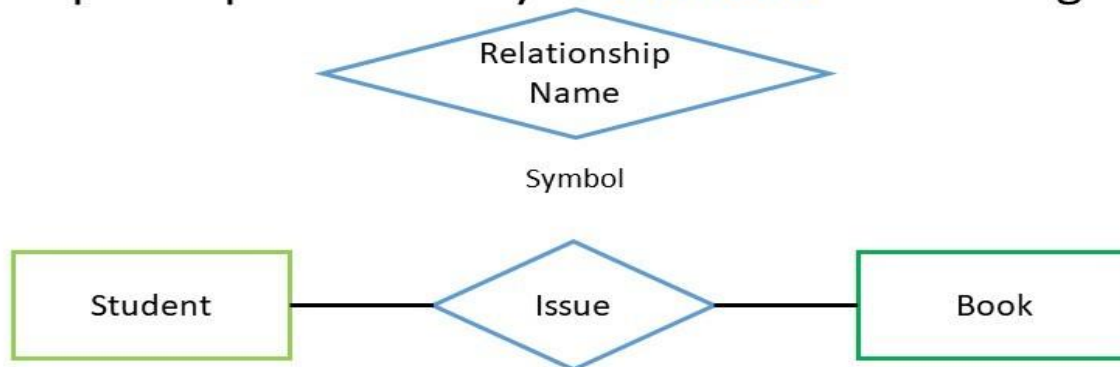
- **Attribute** is details or characteristics about an **entity**.
- An attribute is represented by an oval containing name of an attribute.
- Attributes of Student are:
  - ✓ Student Name
  - ✓ Branch
  - ✓ Semester
  - ✓ Address
  - ✓ Mobile No
  - ✓ Age



# Relationship

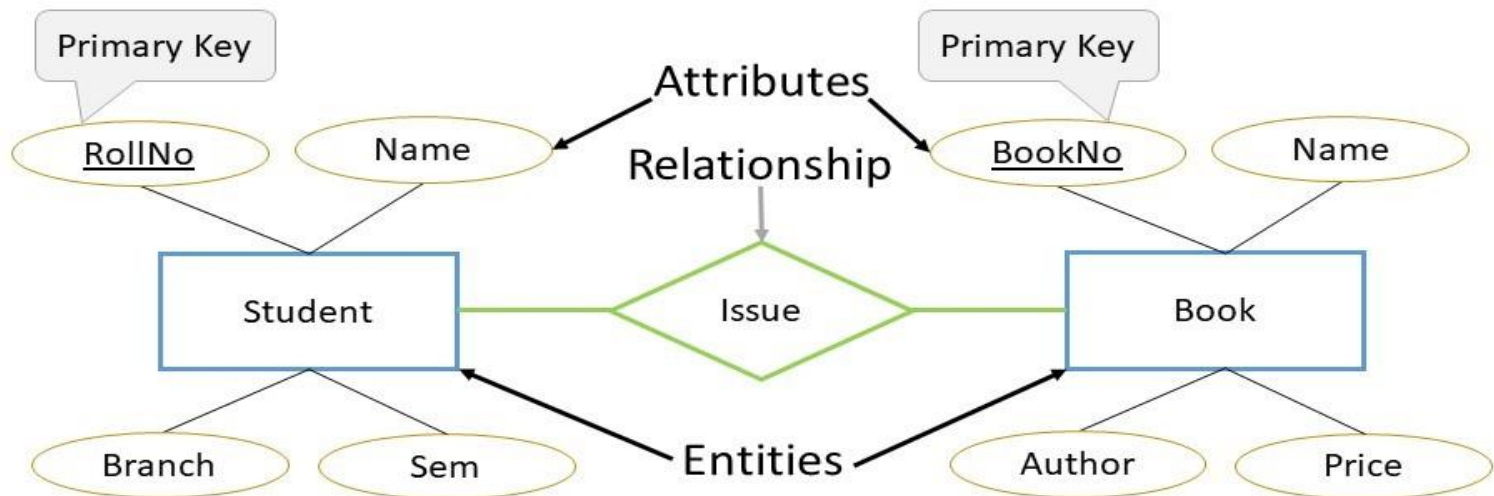
## Relationship

- Relationship is an **association (connection)** between several entities.
- It should be placed between two entities and a line connecting it to an entity.
- A relationship is represented by a **diamond** containing relationship's name.



# E-R Diagram Example:

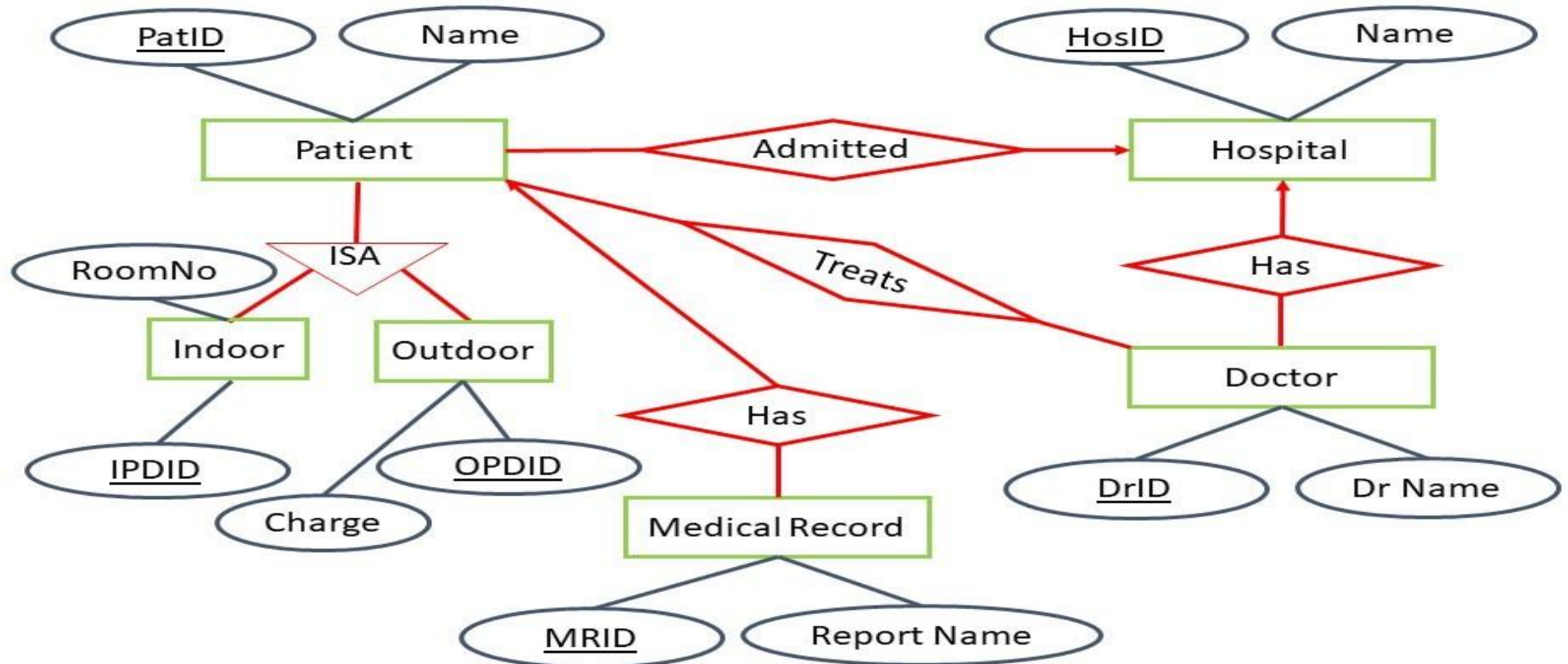
## E-R Diagram of a Library System



- Each and every entity **must have one primary key** attribute.
- **Relationship between 2 entities** is called **binary relationship**.

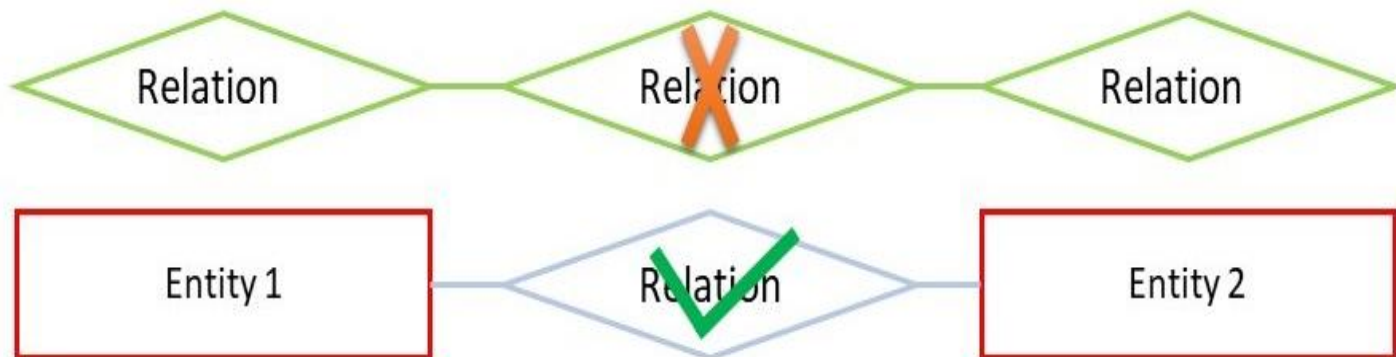
# E-R Diagram Example:

## E-R diagram of Hospital



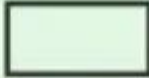




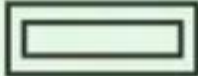
# Limitation of E-R Diagram

In E-R model, we cannot express relationship between two relations.

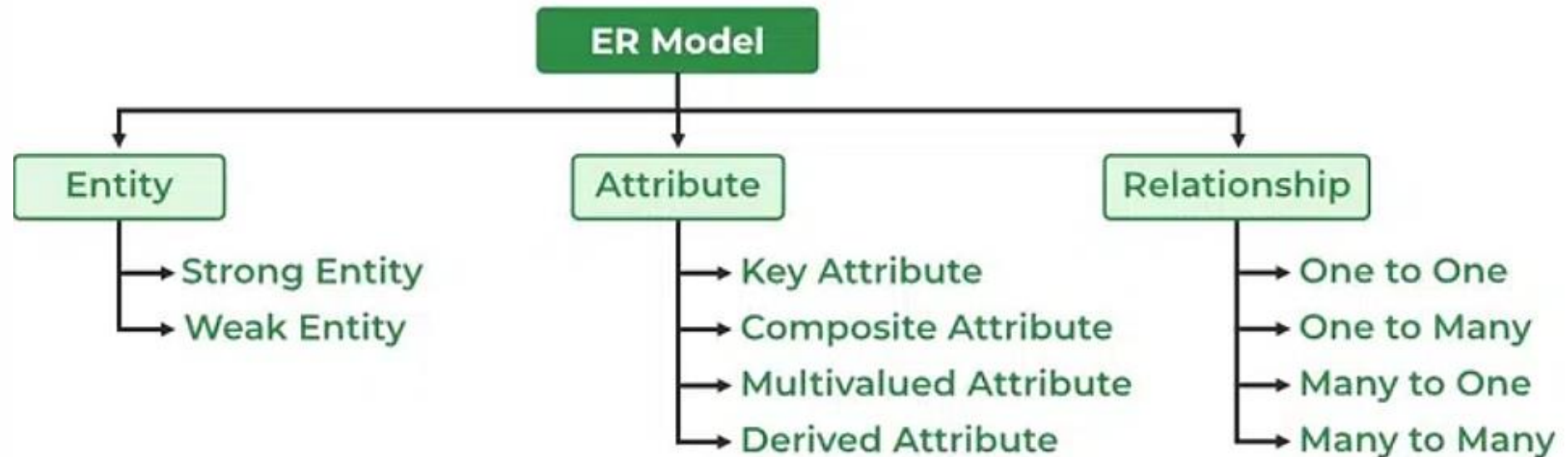




# Symbol of E-R Diagram

Figures	Symbols	Represents
Rectangle		Entities in ER Model
Ellipse		Attributes in ER Model
Diamond		Relationships among Entities
Line		Attributes to Entities and Entity Sets with Other Relationship Types
Double Ellipse		Multi-Valued Attributes
Double Rectangle		Weak Entity

# Components of E-R Diagram

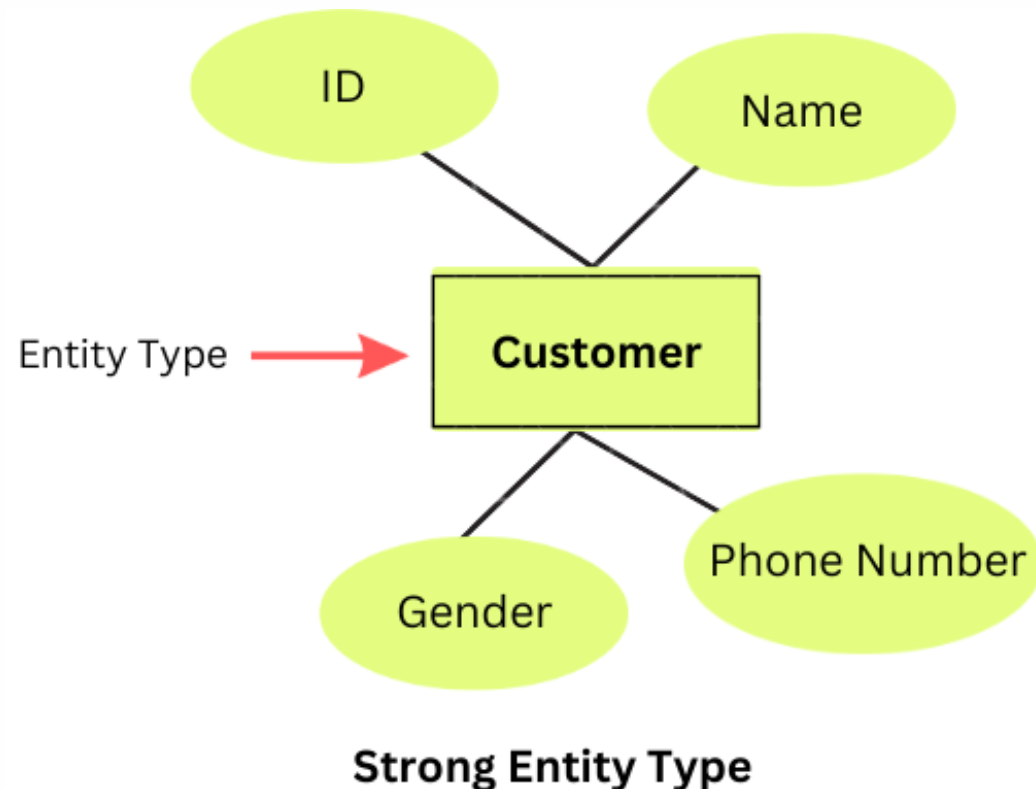


# Types of Entity:

## Strong Entity


❖ **Strong Entity** is an entity that has its own existence and is independent.

- ✓ It is represented by a single rectangle.
- Customer is a strong entity type as it has a unique ID for each customer.



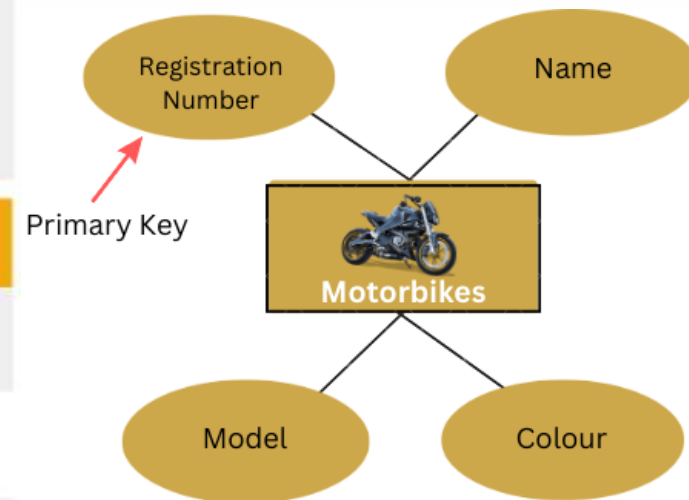
# Types of Entity:

## Strong Entity



**Motorbikes**

	REGISTRATION NUMBER	NAME	MODEL	COLOUR
Entity 1 →	UP515	Yamaha	R15S	Black
Entity 2 →	UP340	Honda	SP125	Red
Entity 3 →	UP414	KTM	RC8	Orange



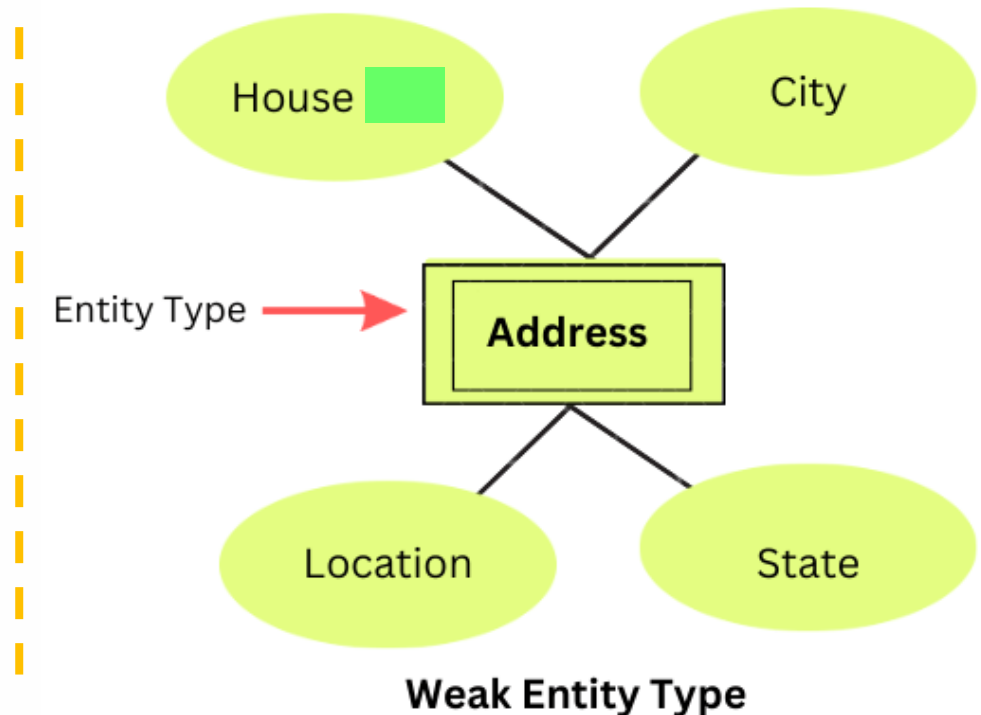
E-R diagram of strong entity

# Types of Entity:

## Weak Entity

❖ **Weak Entity** is an entity that does not have its own existence and relies on a strong entity for its existence.

- ✓ It is represented by double rectangles.
- "Address" is a weak entity type with attributes such as House No., City, Location, and State.

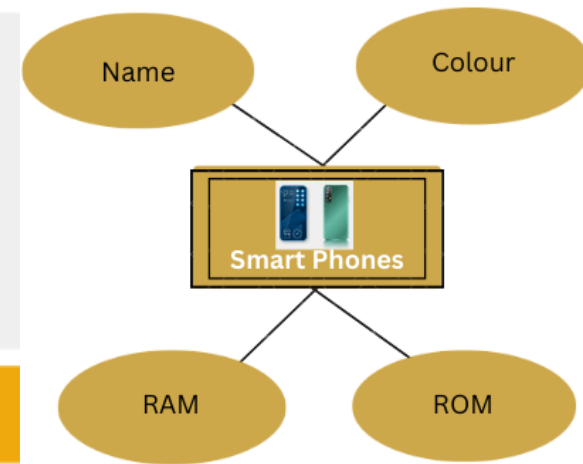


# Types of Entity:

## Weak Entity



	NAME	COLOUR	RAM	ROM
Entity 1 →	Samsung	Sea Green	4GB	64GB
Entity 2 →	Vivo	White	6GB	128GB
Entity 3 →	Redmi	Black	4GB	128GB

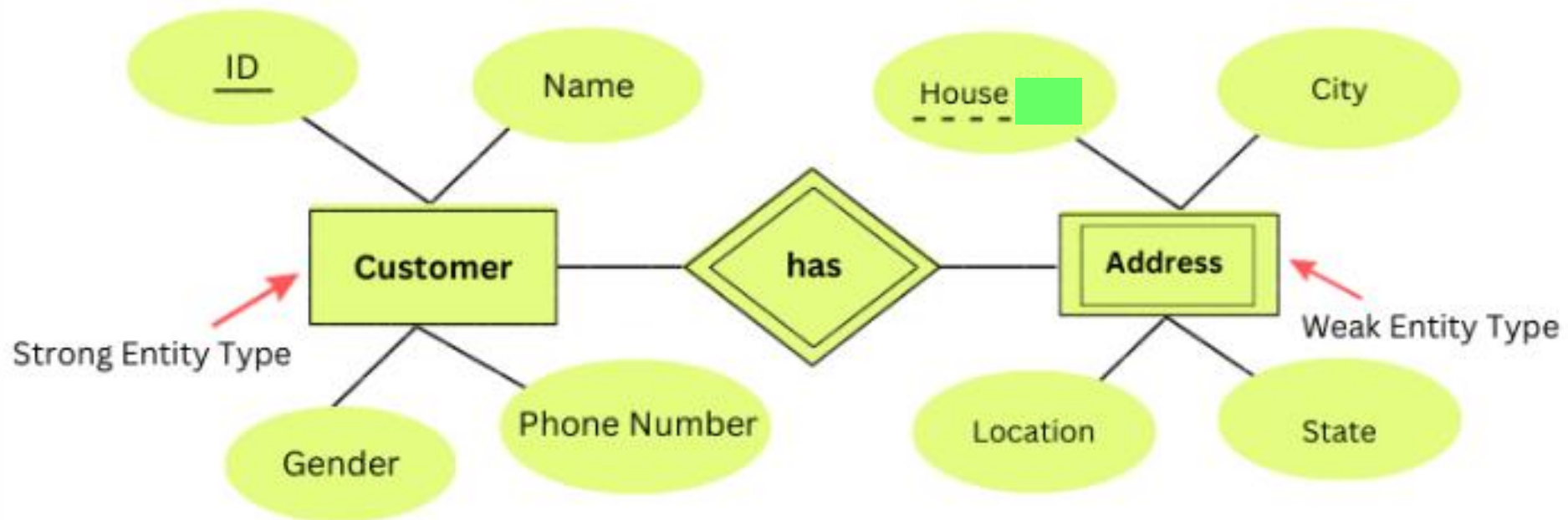


E-R diagram of  
weak entity



# Example of Strong Entity & Weak Entity

The relationship between a strong and a weak entity type is known as an **identifying relationship** and is represented by **double diamond**.



# Strong Entity & Weak Entity

## Strong Entity

Operates independently

Possesses a primary key

Represented by a single rectangle in E-R diagram

Relationship in-between is denoted by a single diamond

## Weak Entity


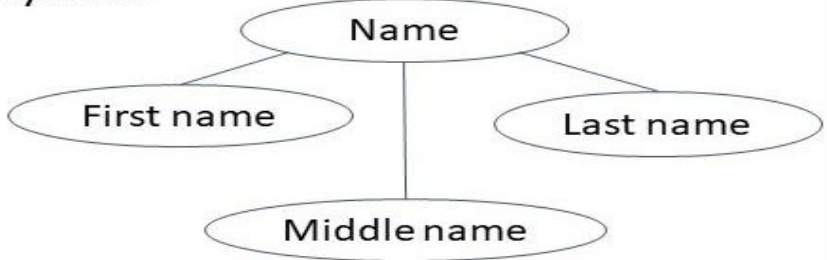
Requires support from a strong entity

Lacks a primary key



Represented by double rectangle in diagram

Relationship in-between is denoted by a double diamond



# Types of Attribute

Simple Attribute	Composite Attribute
<b>Cannot be divided</b> into subparts	<b>Can be divided</b> into subparts
E.g. RollNo, CPI	E.g. Name (first name, middle name, last name)  Address (street, road, city)
Symbol 	Symbol 

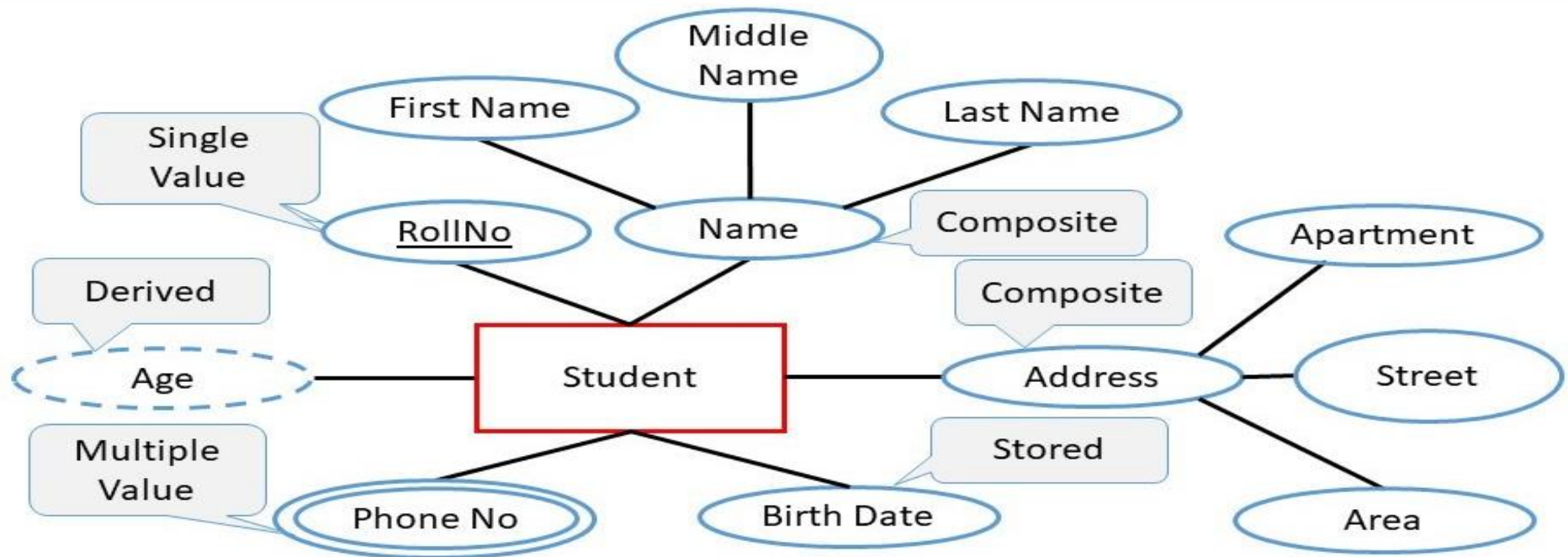
# Types of Attribute

Single-valued Attribute	Multi-valued Attribute
Has <b>single value</b>	Have <b>multiple value</b>
E.g. Rollno, CPI	E.g. Phoneno (person may have multiple phone nos)  EmailID (person may have multiple emails)
Symbol 	Symbol 

# Types of Attribute

Stored Attribute	Derived Attribute
It's value is <b>stored manually</b> in database	It's value is <b>derived</b> or <b>calculated</b> from other attributes.
E.g. Birthdate	E.g. Age (can be calculated using current date and birthdate)
Symbol 	Symbol 

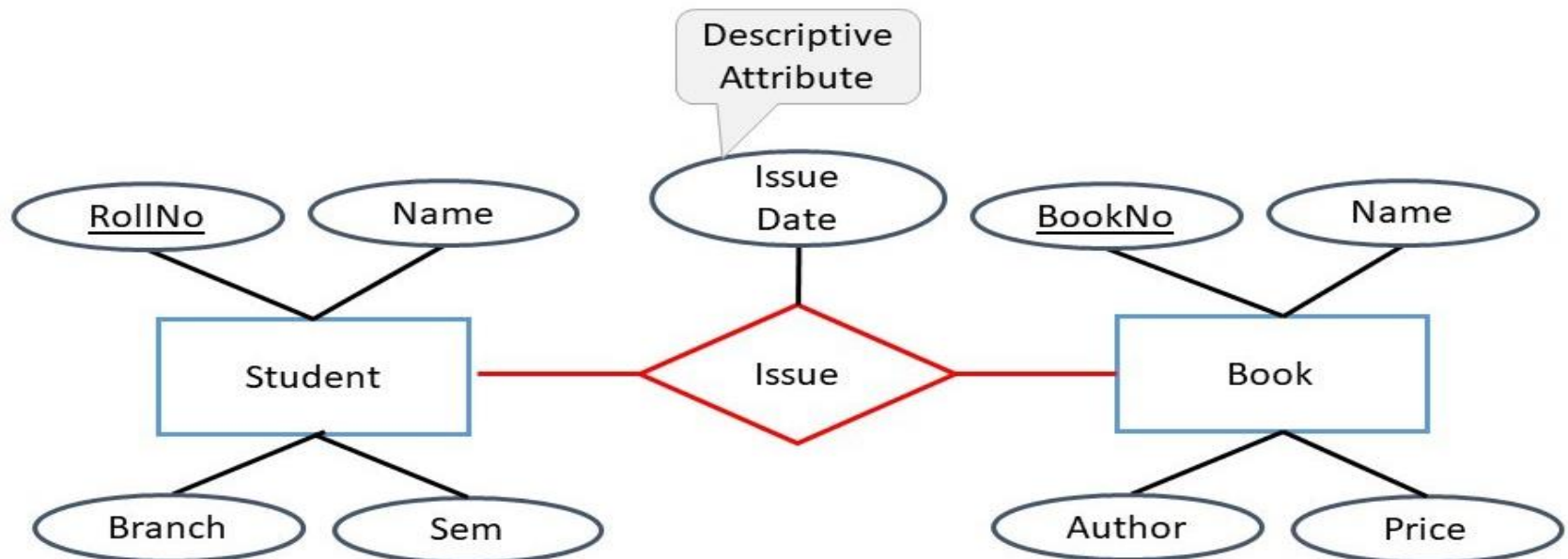
# Entity with all types of Attributes:





# Descriptive Attribute

- **Attributes of the relationship** is called descriptive attribute.

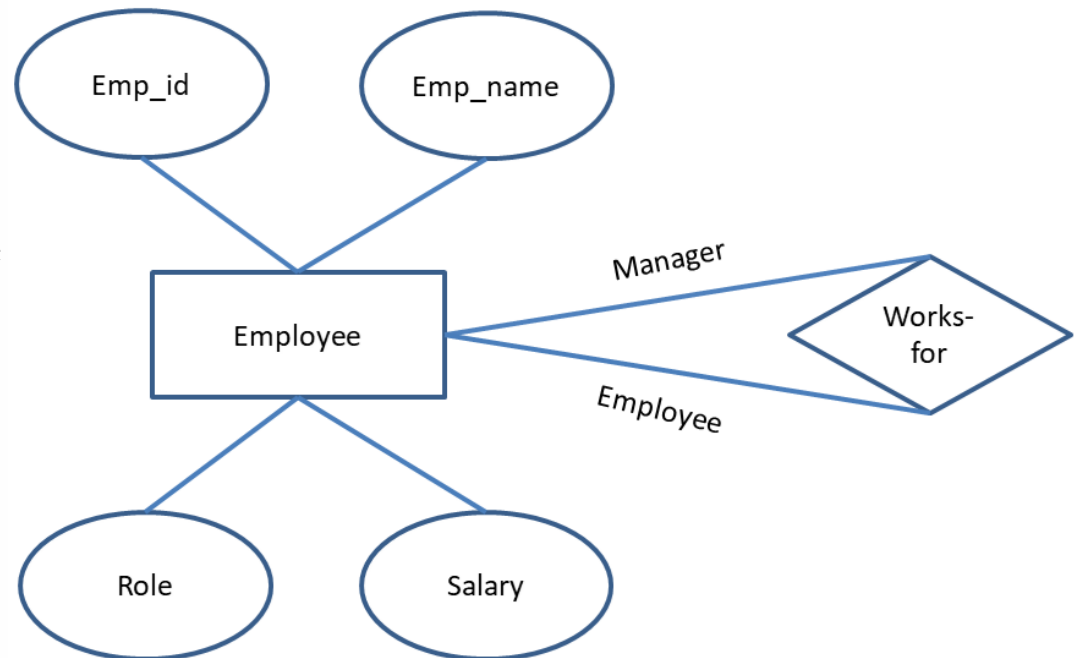




# Roles in E-R Diagram

Roles are mentioned when the relationship meaning needs to clarify.

Example: The labels “manager” and “worker” are called roles; they specify how employee entities interact via the works-for relationship set.



Roles are indicated in E-R diagrams by labeling the lines that connect diamonds to rectangles.

# Mapping Cardinality/Cardinality Ratio

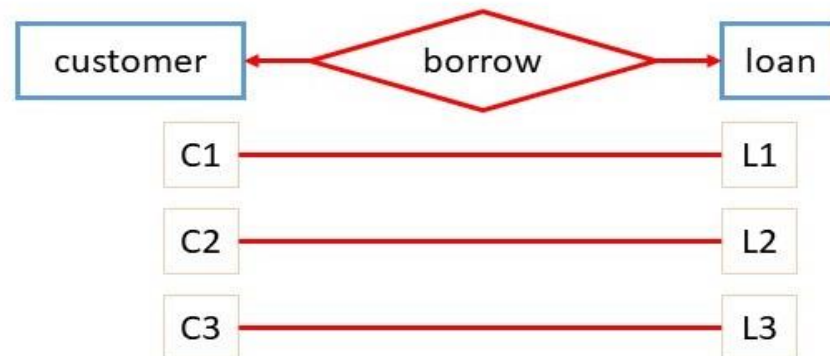
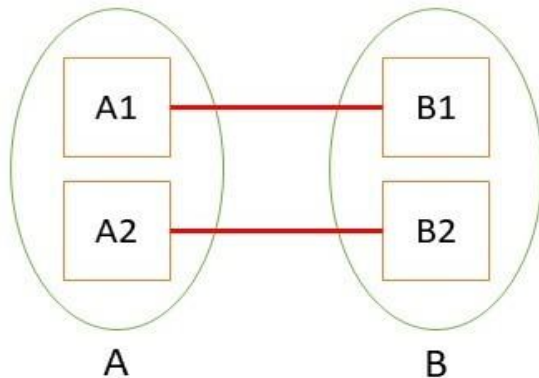
It shows the number of instances of one entity that can be linked to the number of instances of another entity via a certain kind of relationship such as:

1. One to one
2. One to many
3. Many to one
4. Many to many

# One to One Relationship

## One-to-One relationship (1 – 1)

- An entity in **A** is associated with only one entity in **B** and an entity in **B** is associated with only one entity in **A**.

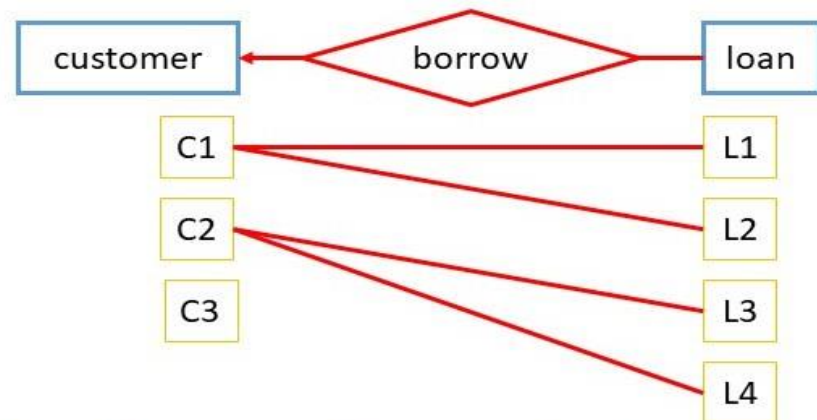
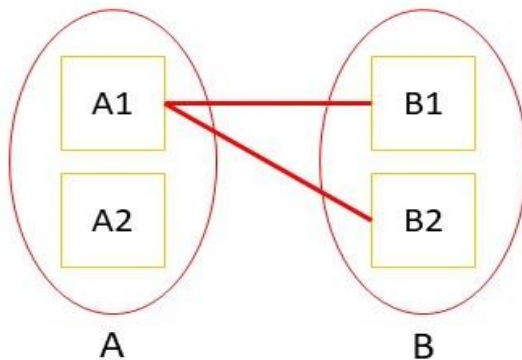


- Example:** A **customer** is connected with only one **loan** using the relationship **borrower** and a **loan** is connected with only one **customer** using **borrower**.

# One to Many Relationship

## One-to-Many relationship (1 – N)

- An entity in **A** is associated with more than one entities in **B** and an entity in **B** is associated with only one entity in **A**.

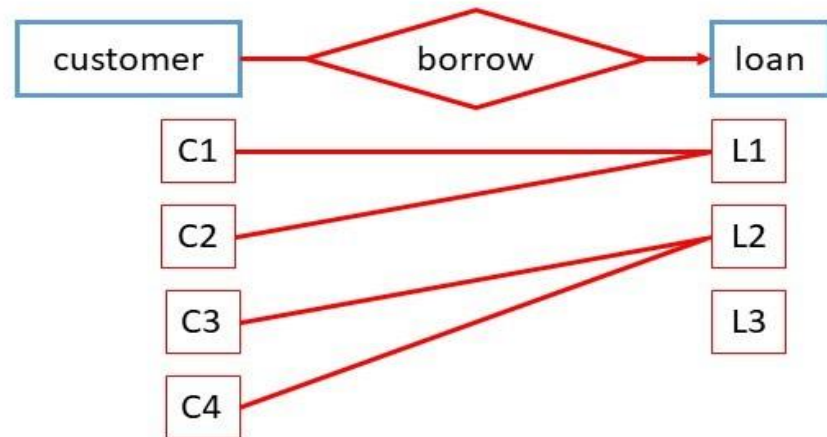
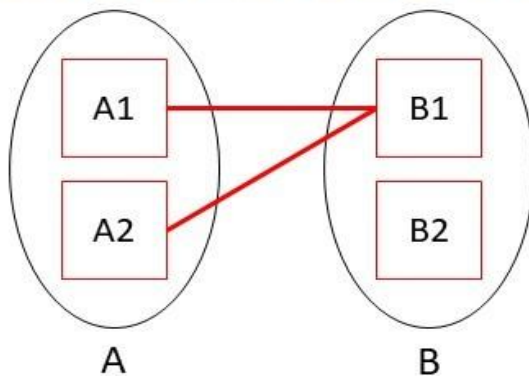


- **Example:** A loan is connected with only one customer using borrower and a customer is connected with more than one loans using borrower.

# Many to One Relationship

## Many-to-One relationship ( $N - 1$ )

- An entity in **A** is associated with only one entity in **B** and an entity in **B** is associated with more than one entities in **A**.



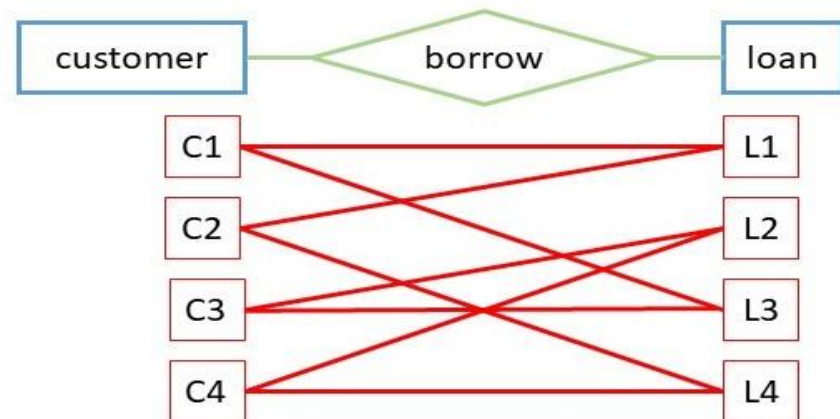
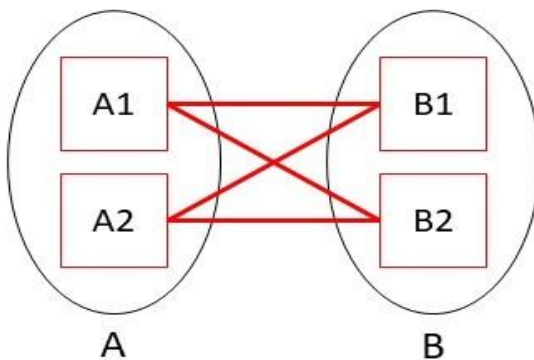
- Example:** A **loan** is connected with more than one **customer** using borrower and a **customer** is connected with only one **loan** using borrower.



# Many to Many Relationship

## Many-to-Many relationship (N – N)

- An entity in **A** is associated with more than one entities in **B** and an entity in **B** is associated with more than one entities in **A**.



- Example:** A **customer** is connected with more than one **loan** using **borrower** and a **loan** is connected with more than one **customer** using **borrower**.

# Participation Constraints

It specifies the number of instances of an entity can participate in a relationship set.

## Types of Participation Constraints

1.

Total Participation

2.

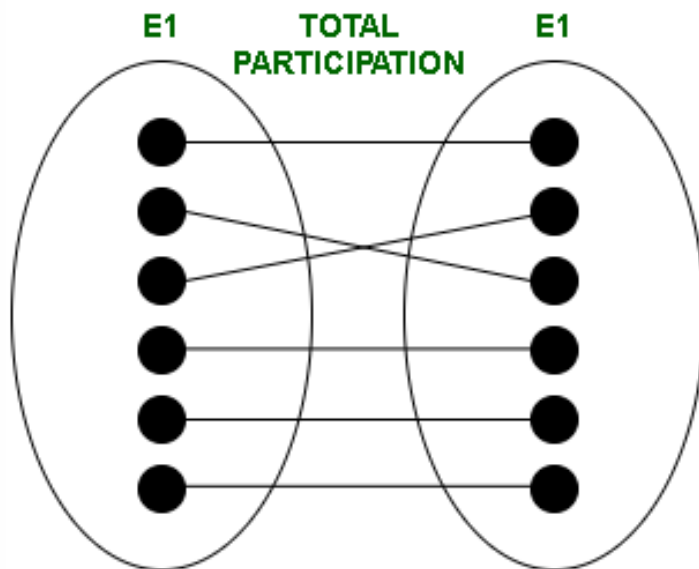
Partial Participation



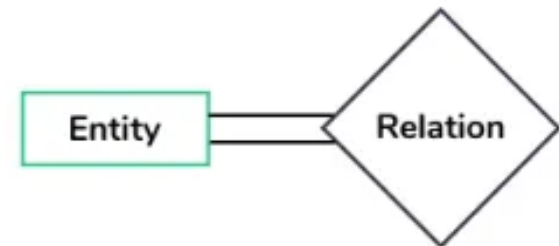
# Participation Constraints

If every entity participates in at least one relationship, it is called **Total participation**.

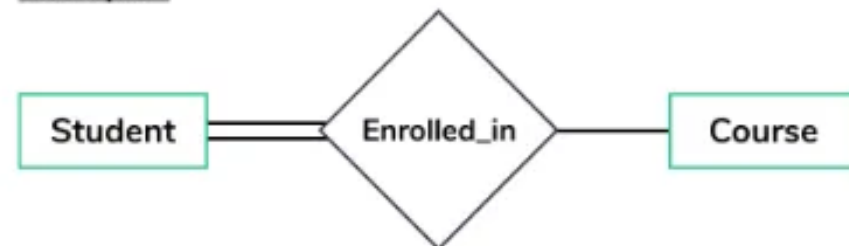
It is represented using a double line between the entity set and relationship set.



Representation :



Example :

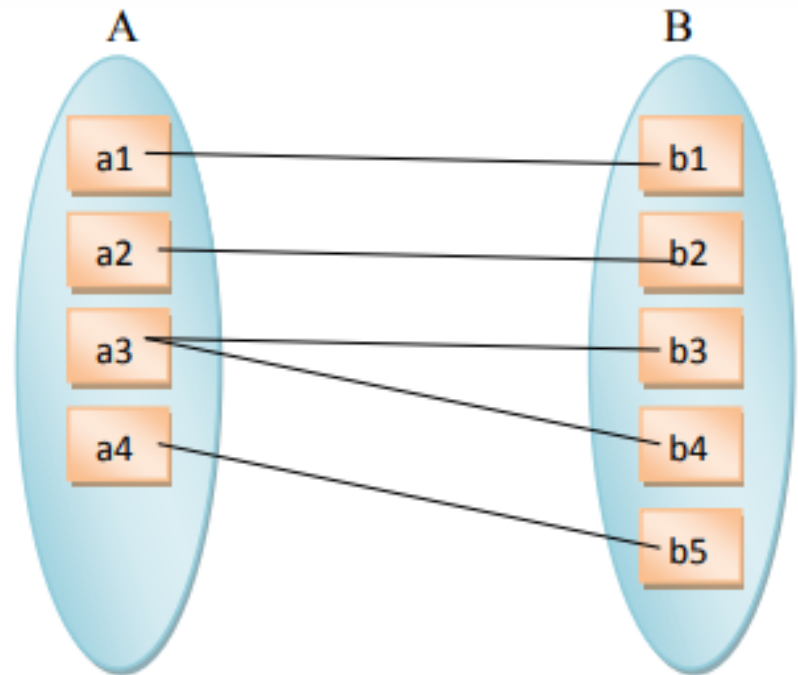


# Participation Constraints

The participation of entity set A in the relationship set is **total** because every entity of A participates in the relationship set.

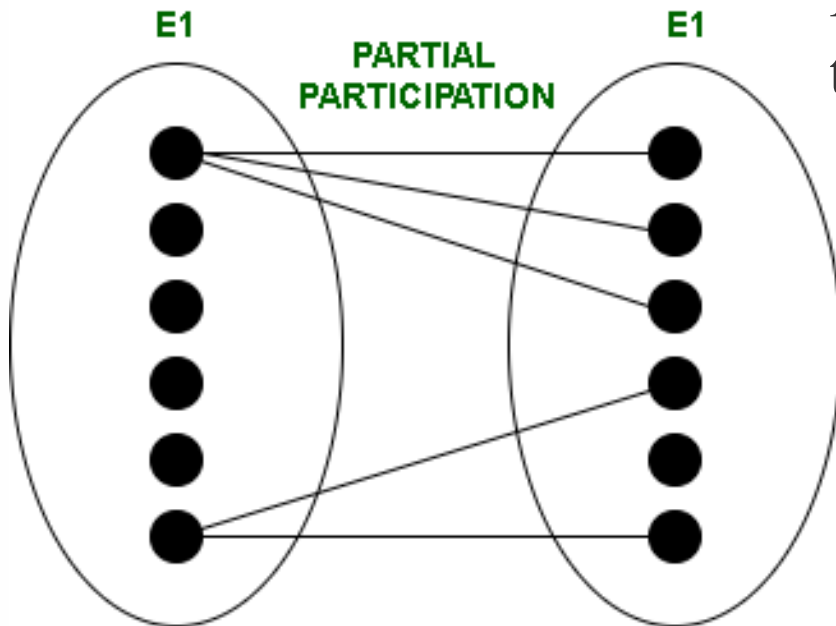
and

The participation of entity set B in the relationship set is also **total** because every entity of B also participates in the relationship set.



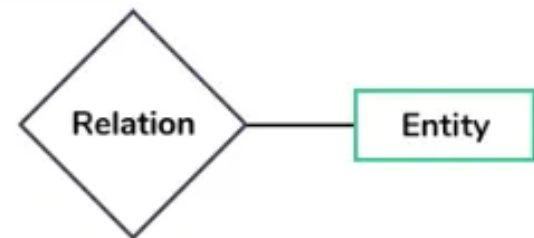
# Participation Constraints

If only some entities participate in relationships, it is called **Partial participation**.

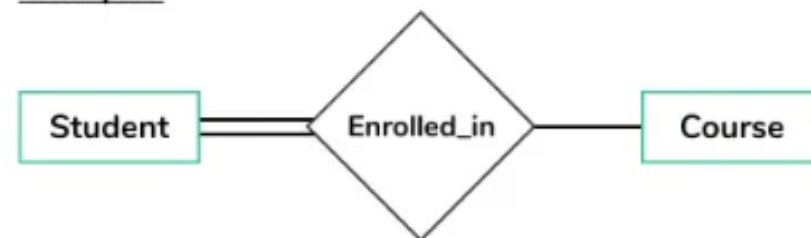


It is represented by a single line between the entity set and relationship set.

Representation :



Example :

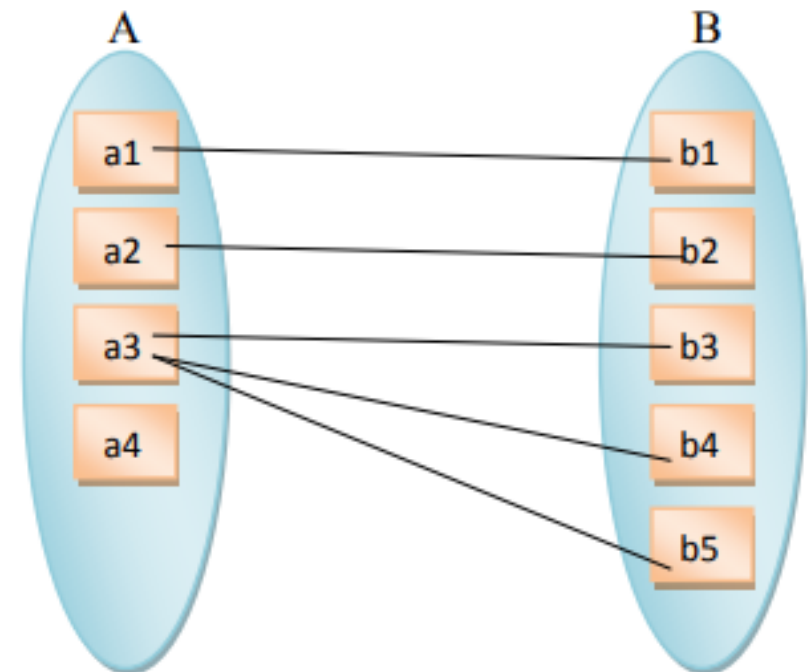


# Participation Constraints

The participation of entity set A in the relationship set is **partial** because only some entities of A participate in the relationship set.

while

The participation of entity set B in the relationship set is **total** because every entity of B participates in the relationship set.



## Generalization, Specialization and Aggregation in ER Model

Using the ER model for bigger data creates a lot of complexity while designing a database model.

In order to minimize the complexity, some new concepts were added in the ER model.

**These new concepts are:**

- Generalization
- Specialization
- Aggregation

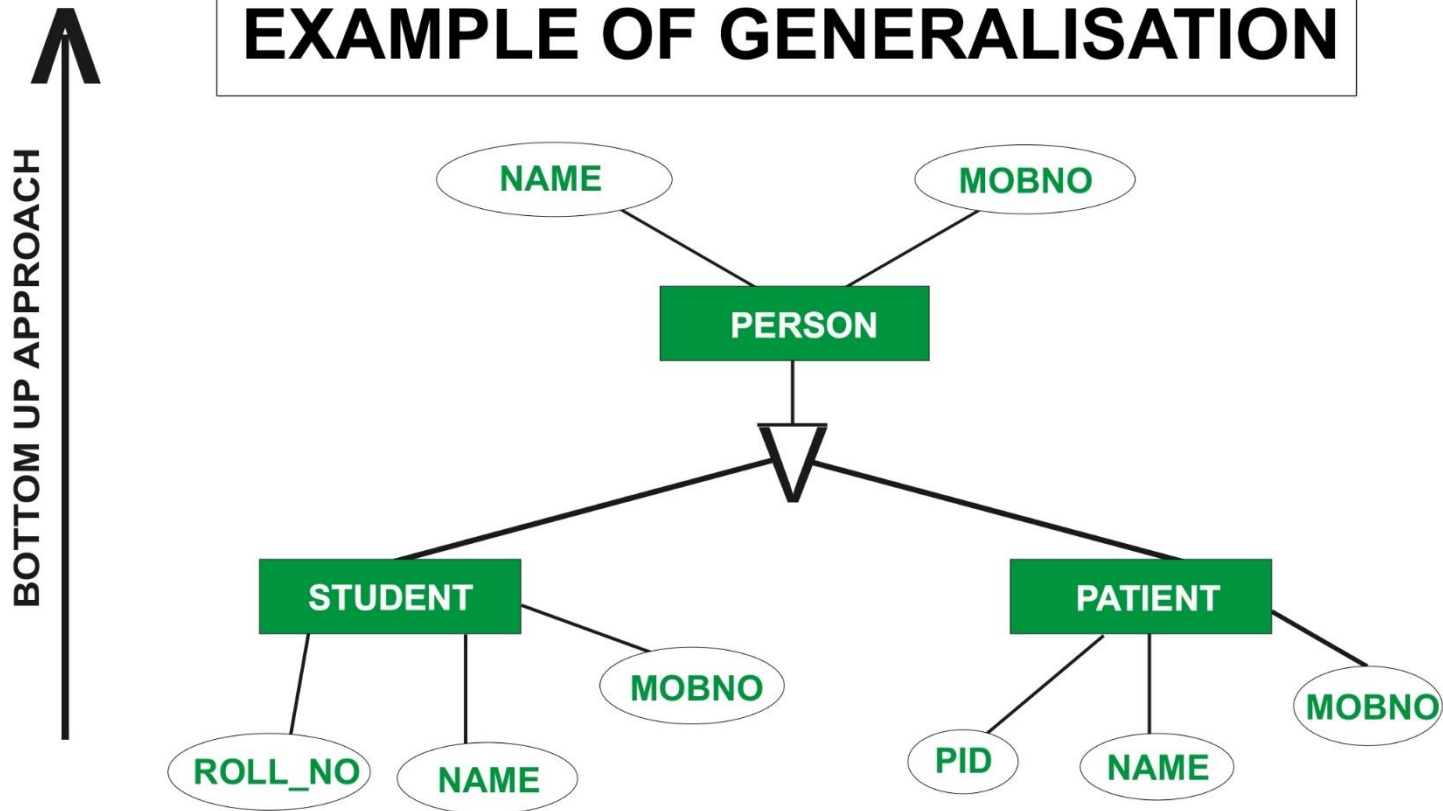
# Generalization

Generalization is a process of generalizing an entity which contains generalized attributes or properties of generalized entities.

- ✓ The new entity that is created will contain the common features.
- Generalization is a Bottom up process.

# Generalization

## EXAMPLE OF GENERALISATION





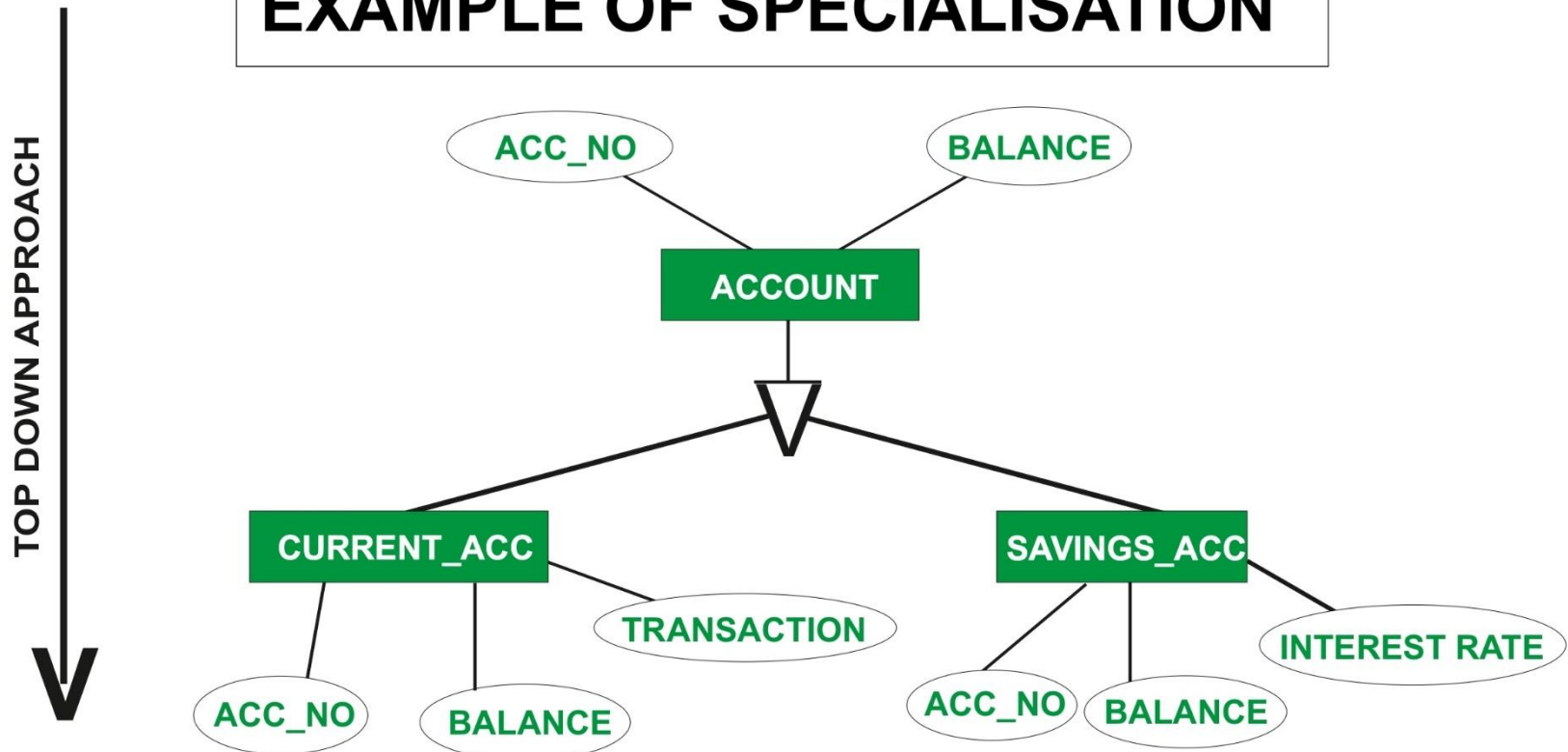
# Specialization

In Specialization things are broken down into smaller things to simplify it further.

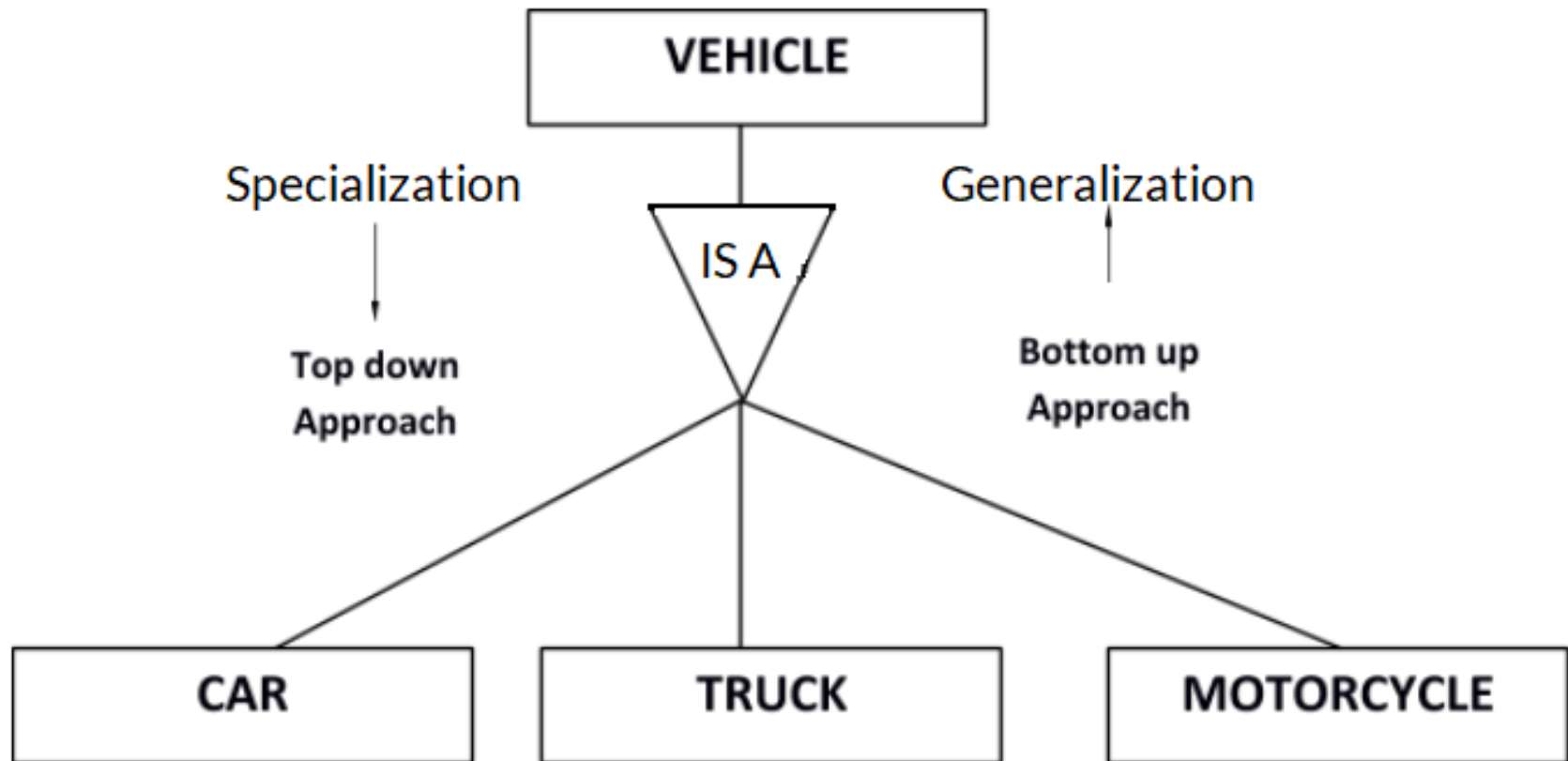
- Specialization is opposite of Generalization.
- ✓ It is a top down approach.

# Specialization

## EXAMPLE OF SPECIALISATION



# Generalization & Specialization



# Generalization & Specialization

## Generalization

works in Bottom-Up approach

size of schema gets reduced

It is normally applied to group of entities.

It is a process of creating groupings from various entity sets

## Specialization

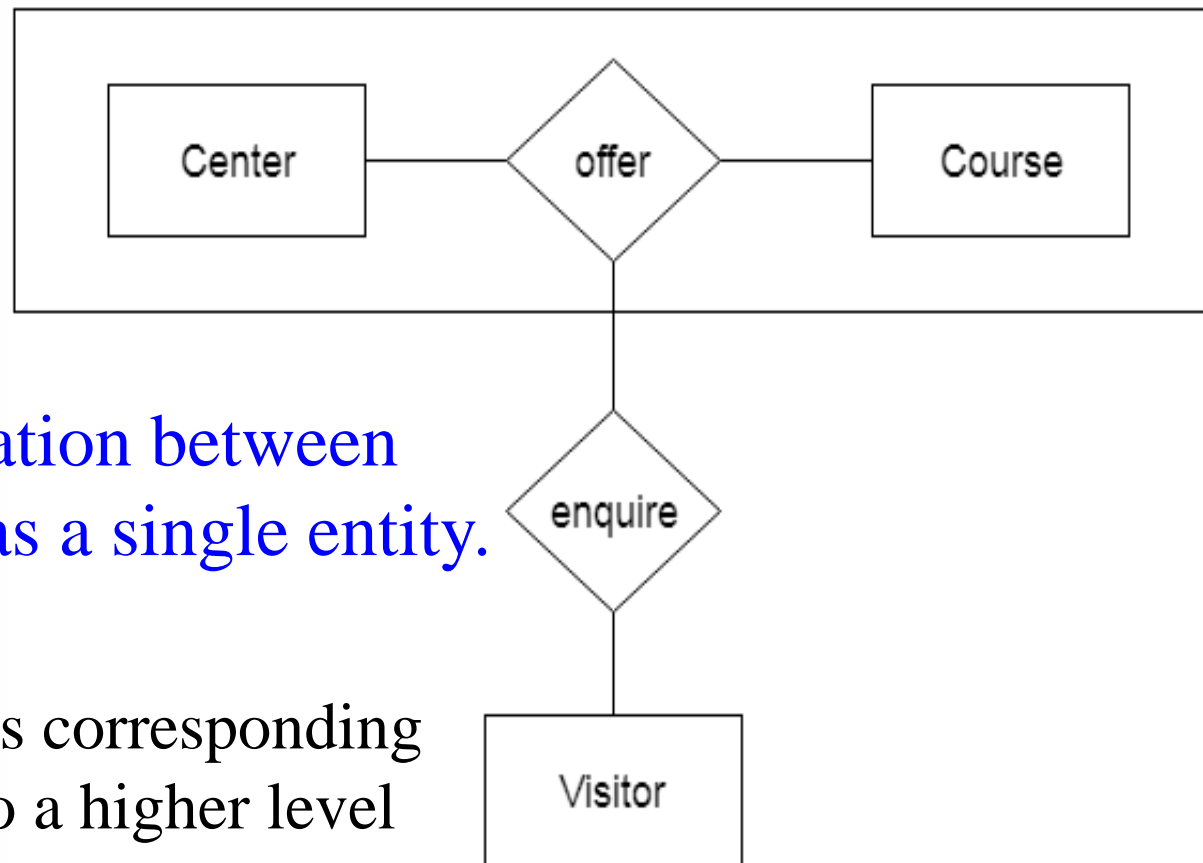
works in top-down approach

size of schema gets increased

It can be applied to a single entity

It is a process of creating subgrouping within an entity set

# Aggregation



In aggregation, the relation between two entities is treated as a single entity.

Here, relationship with its corresponding entities is aggregated into a higher level entity.

**Thank You!!!**

# x DIGITAL LEARNING CONTENT

0



## Parul<sup>®</sup> University



[www.paruluniversity.ac.in](http://www.paruluniversity.ac.in)

