Unit - II E-R Modeling

UNIT-II E-R Modeling:

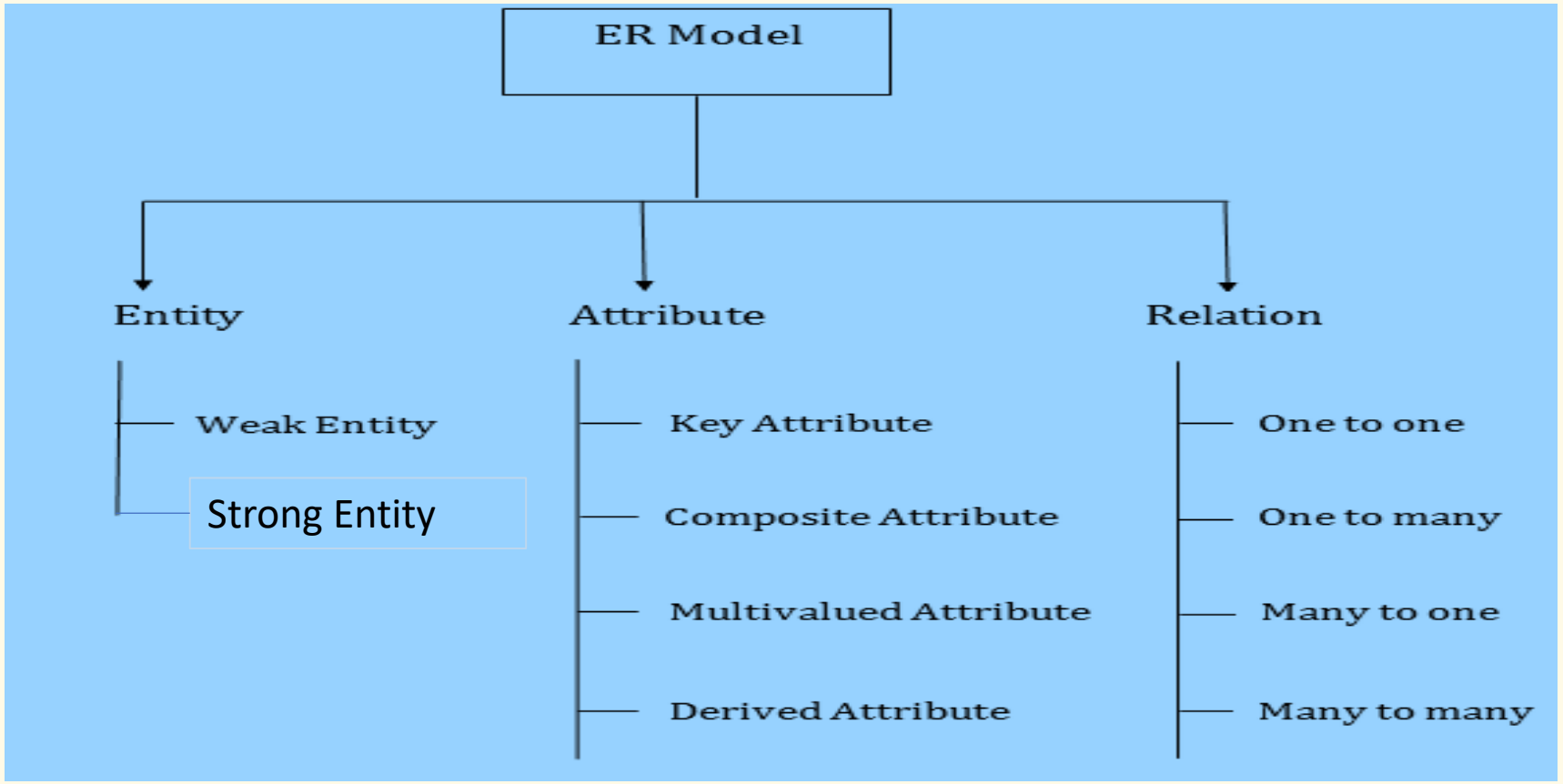
- Entity types, Entity set
- attribute and key relationships
- relation types
- roles and structural constraints
- weak entities
- enhanced E-R and object modeling
- Sub classes; Super classes, inheritance, specialization and generalization.

ER Diagram

ER Diagram is known as Entity-Relationship Diagram, it is used to analyze the structure of the Database. It shows relationships between entities and their attributes. An ER Model provides a means of communication.

ER diagram was developed by **Peter Chen in 1976** to represent database structure.

Component of ER Diagram

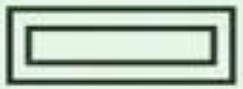


Symbols Used in ER Model

ER Model shows data and its relationships using symbols:

- **Rectangle** → Entity
- **Ellipse** → Attribute
- **Diamond** → Relationship
- **Line** → Connects entities and attributes
- **Double Ellipse** → Multi-valued Attribute
- **Double Rectangle** → Weak Entity

Unit - II E-R Modeling

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

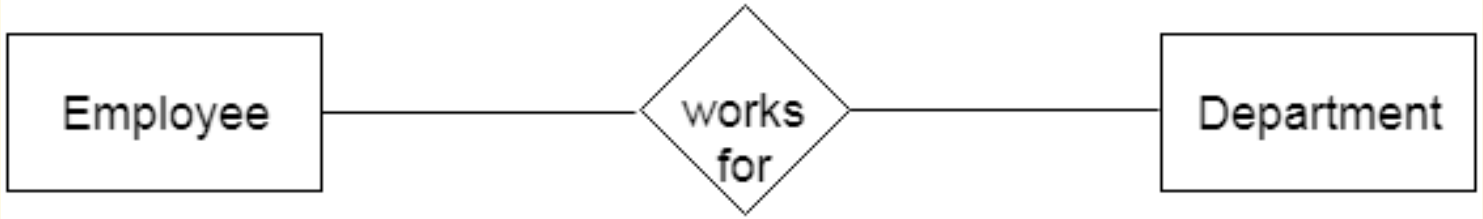
Unit - II E-R Modeling

ER Modeling

ER (Entity-Relationship) modeling is used to **design a database** by showing entities, their attributes, and relationships.

Entity: An entity is any object, person, place, or thing. In ER diagrams, it is shown as a rectangle.
A real-world object that can be identified

Example: In an organization – A student named Ram,
Manager, Product, Employee, Department are entities.



Unit - II E-R Modeling

Entity Types

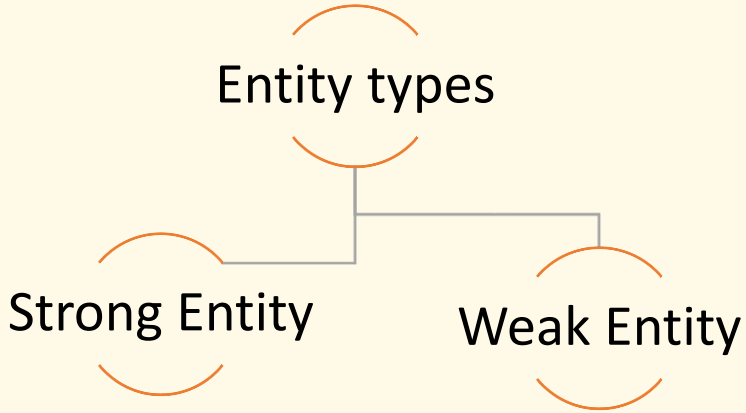
1.Strong Entity

1. Identified by its own attributes.
2. Example: **Customer** (has Customer_ID).
3. **Symbol:** Rectangle

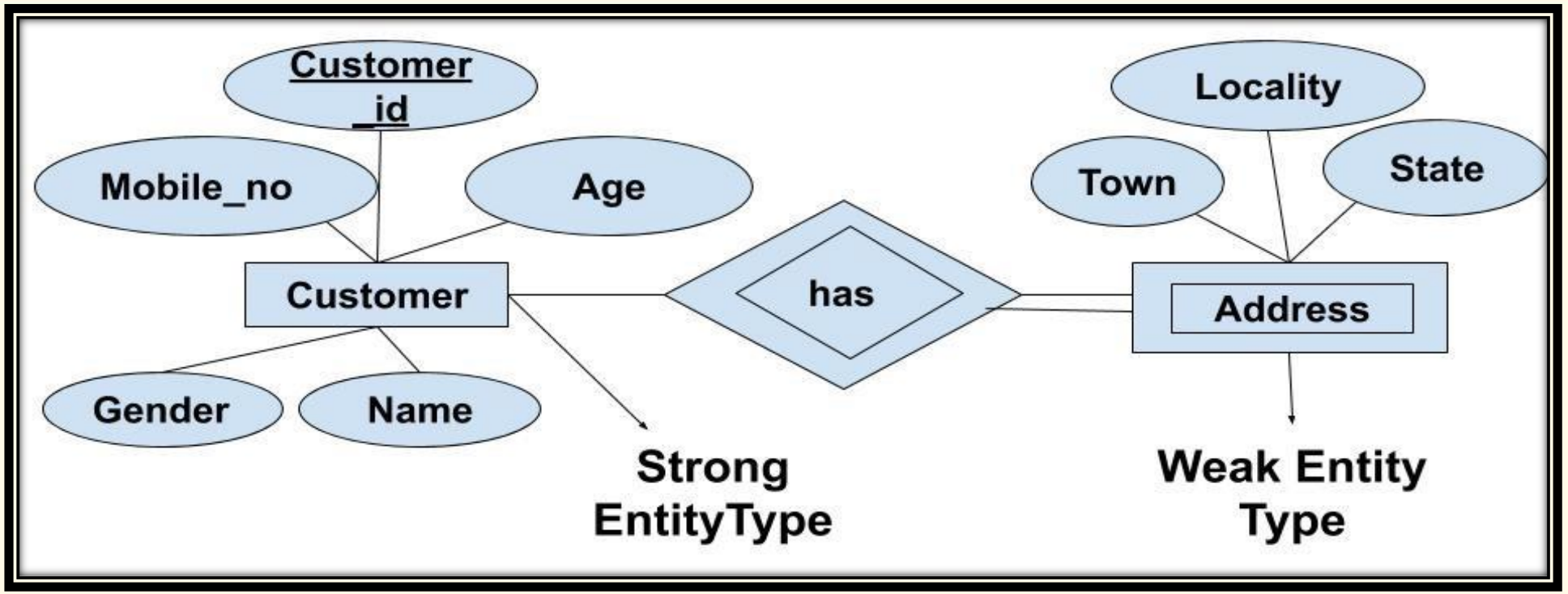
2.Weak Entity

1.This is dependent on the Customer (It cannot exit without a customer).

2. Example: **Dependent** (depends on Customer).
3. Address has attributes
4. **Symbol:** Double Rectangle



Unit - II E-R Modeling

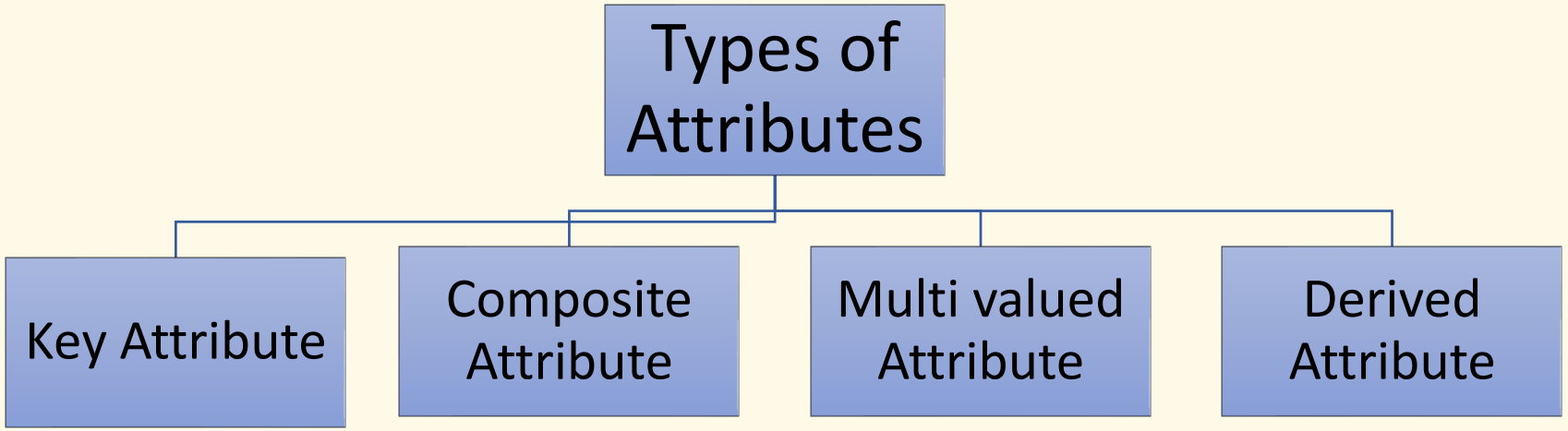


- **Entity Set**
  - Collection of all entities of an entity type stored in the database at a particular time.
  - Example: All students in the database form the **Student Entity Set**.

Entity Set

. Attribute

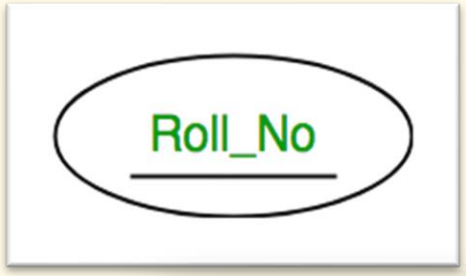
The attribute is used to describe the property of an entity. Eclipse is used to represent an attribute. For example, id, age, contact number, name, etc. can be attributes of a student.



Unit - II E-R Modeling

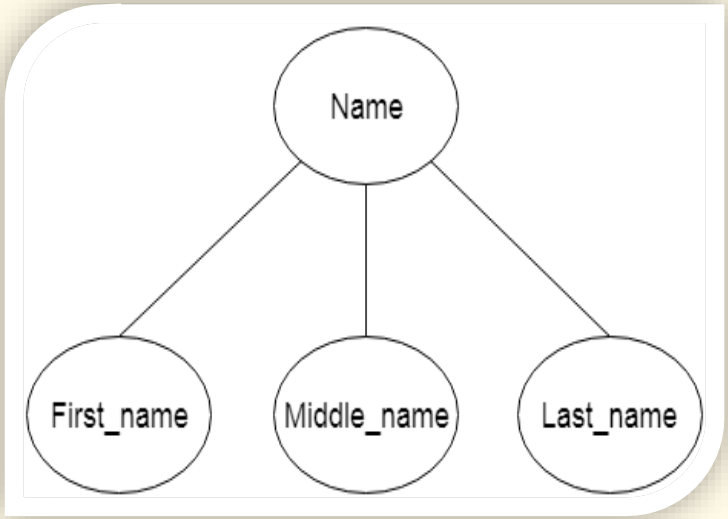
a. Key Attribute

The key attribute is used to represent the main characteristics of an entity. It represents a primary key. The key attribute is represented by an ellipse with the text underlined.



b. Composite Attribute

An attribute that composed of many other attributes is known as a composite attribute. The composite attribute is represented by an ellipse, and those ellipses are connected with an ellipse.

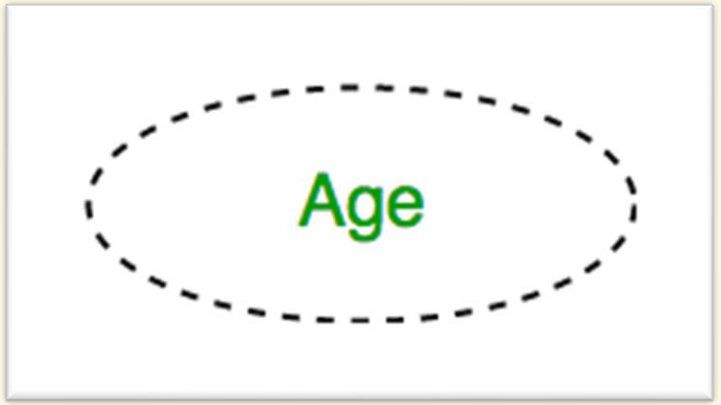


c. Multivalued Attribute

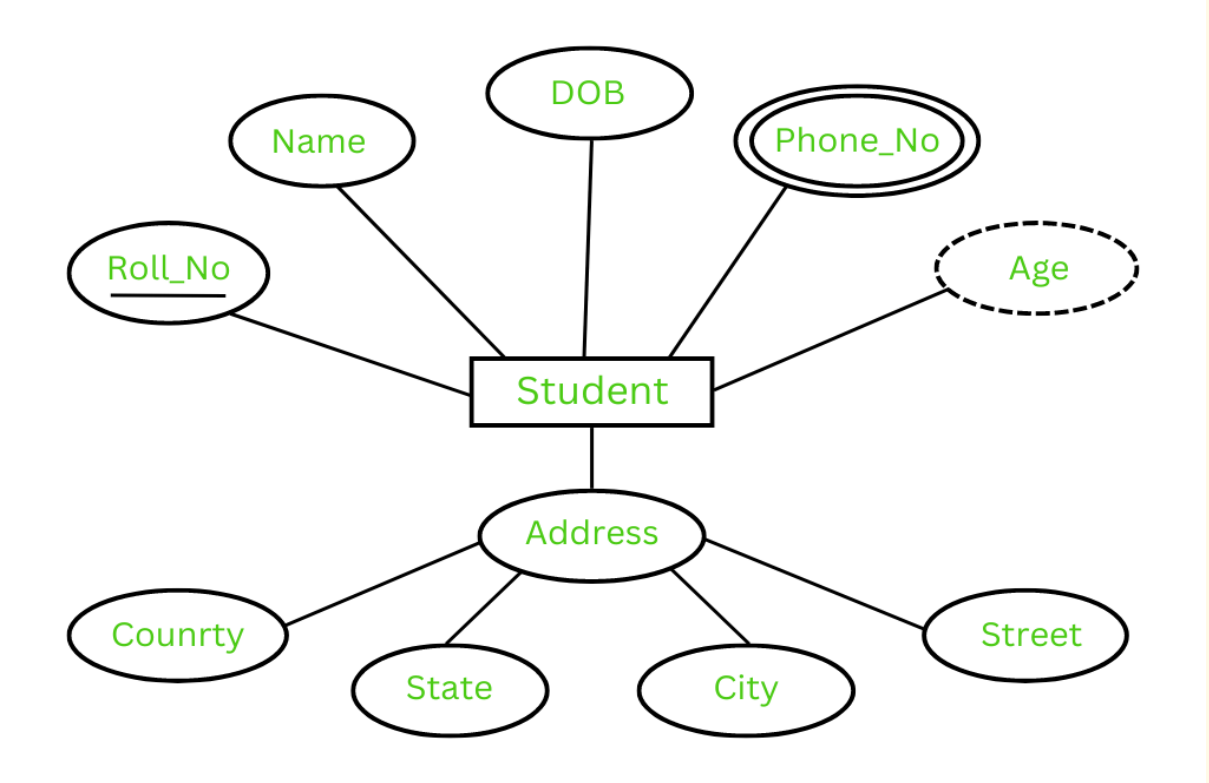
An attribute consisting of more than one value for a given entity.
Can have multiple values for a single entity.
For example, Phone_No (can be more than one for a given student).
In ER diagram, a multivalued attribute is represented by a double oval.

d. Derived Attribute

An attribute that can be derived from other attributes of the entity type is known as a derived attribute. **Example:** Age (calculated from Date of Birth).
In ER diagram, the derived attribute is represented by a dashed oval.

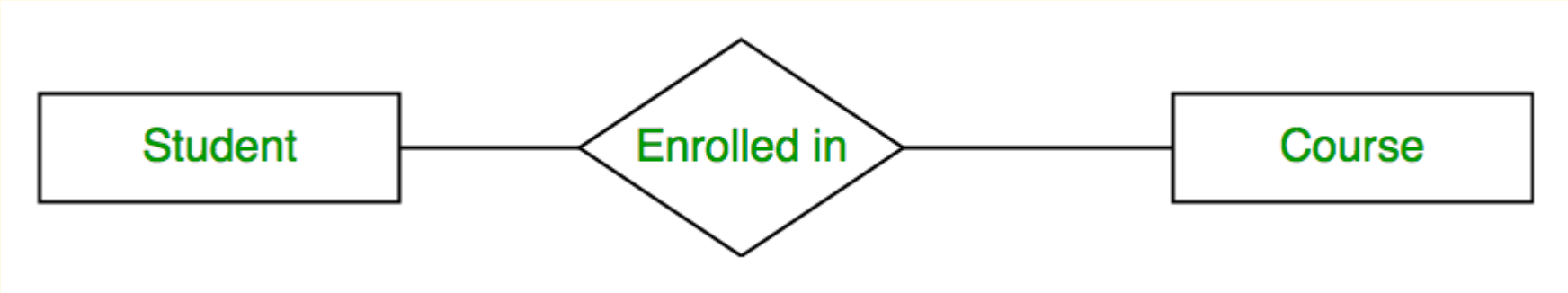


The Complete Entity Type Student with its Attributes can be represented as:



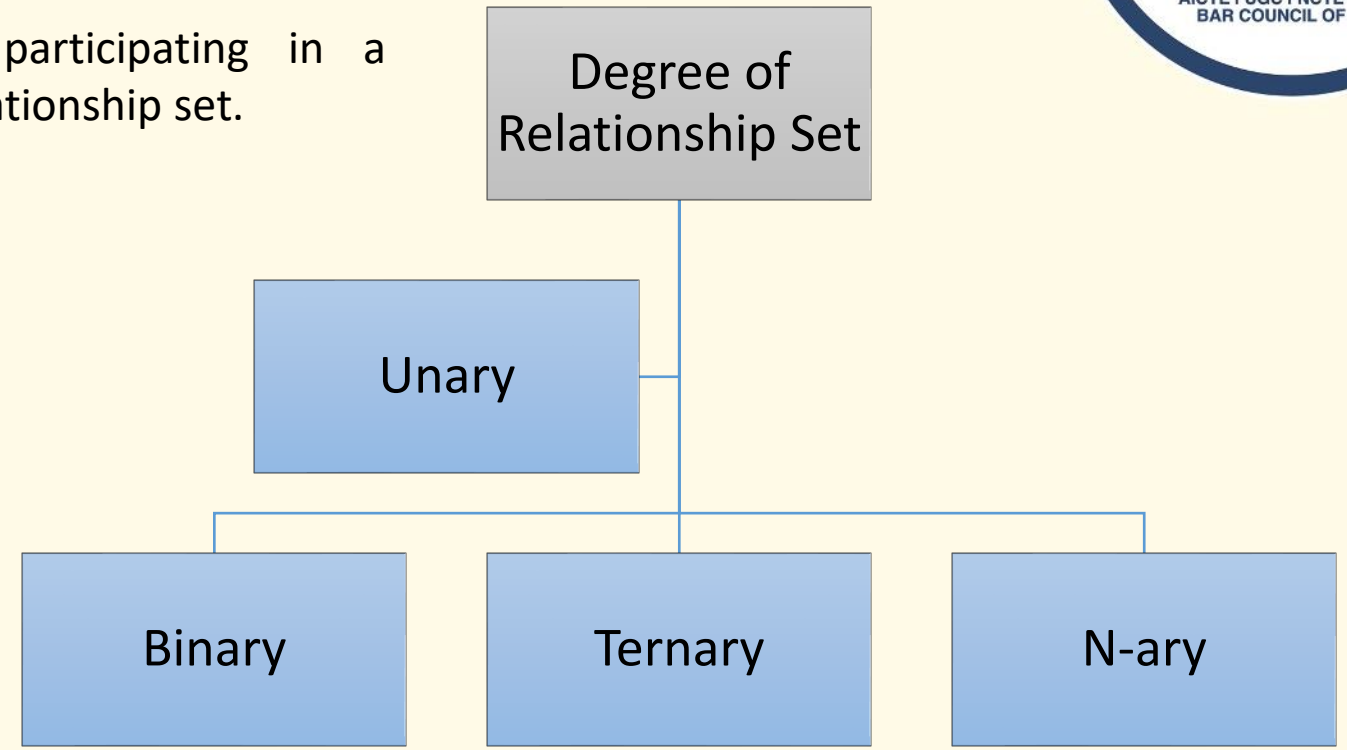
Relationship Type and Relationship Set

A Relationship Type represents the association between entity types. For example, 'Enrolled in' is a relationship type that exists between entity type Student and Course. In ER diagram, the relationship type is represented by a diamond and connecting the entities with lines.

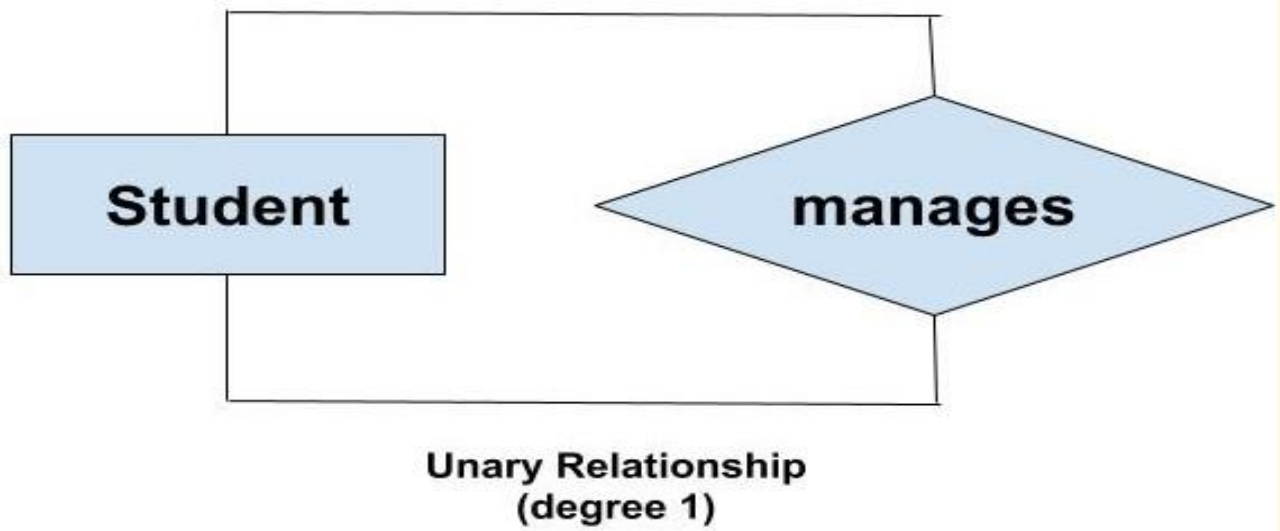


Degree of a Relationship Set

The number of different entity sets participating in a relationship set is called the degree of a relationship set.

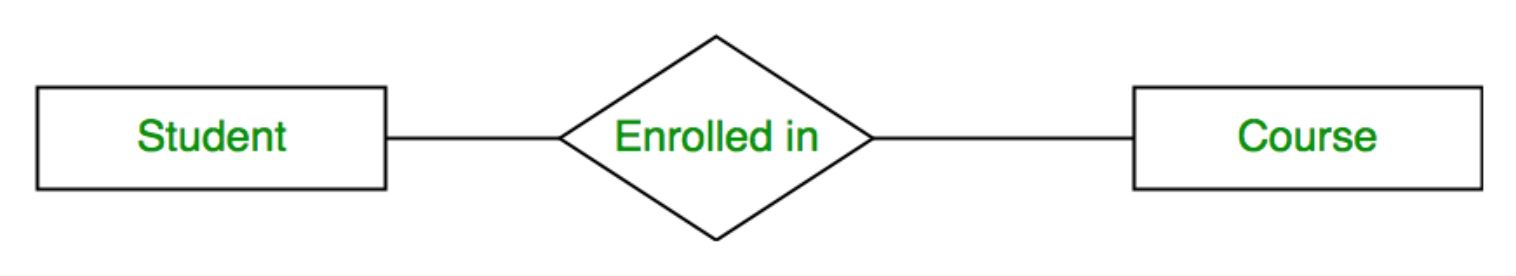


1. Unary Relationship: When there is only ONE entity set participating in a relation, the relationship is called a unary relationship. For example, one person is married to only one person.

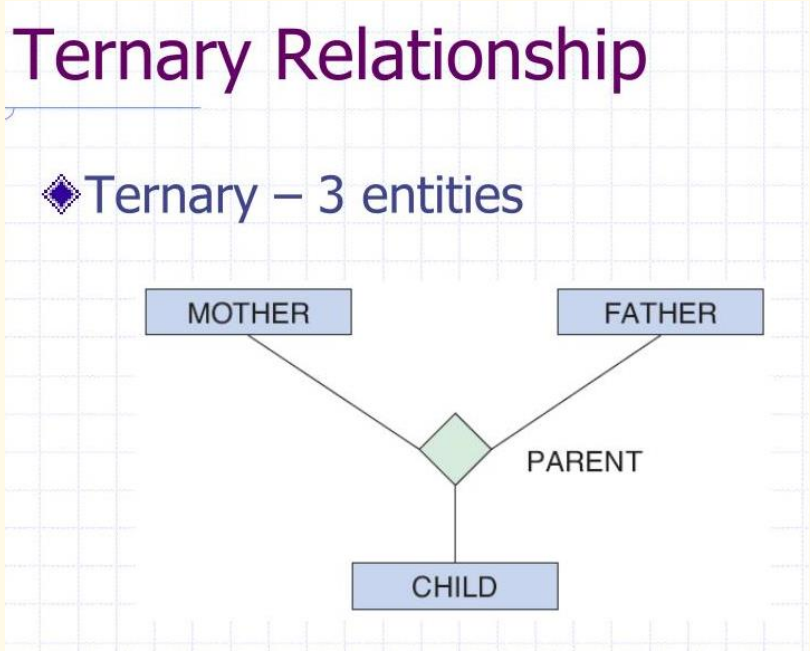


Unary Relationship
(degree 1)

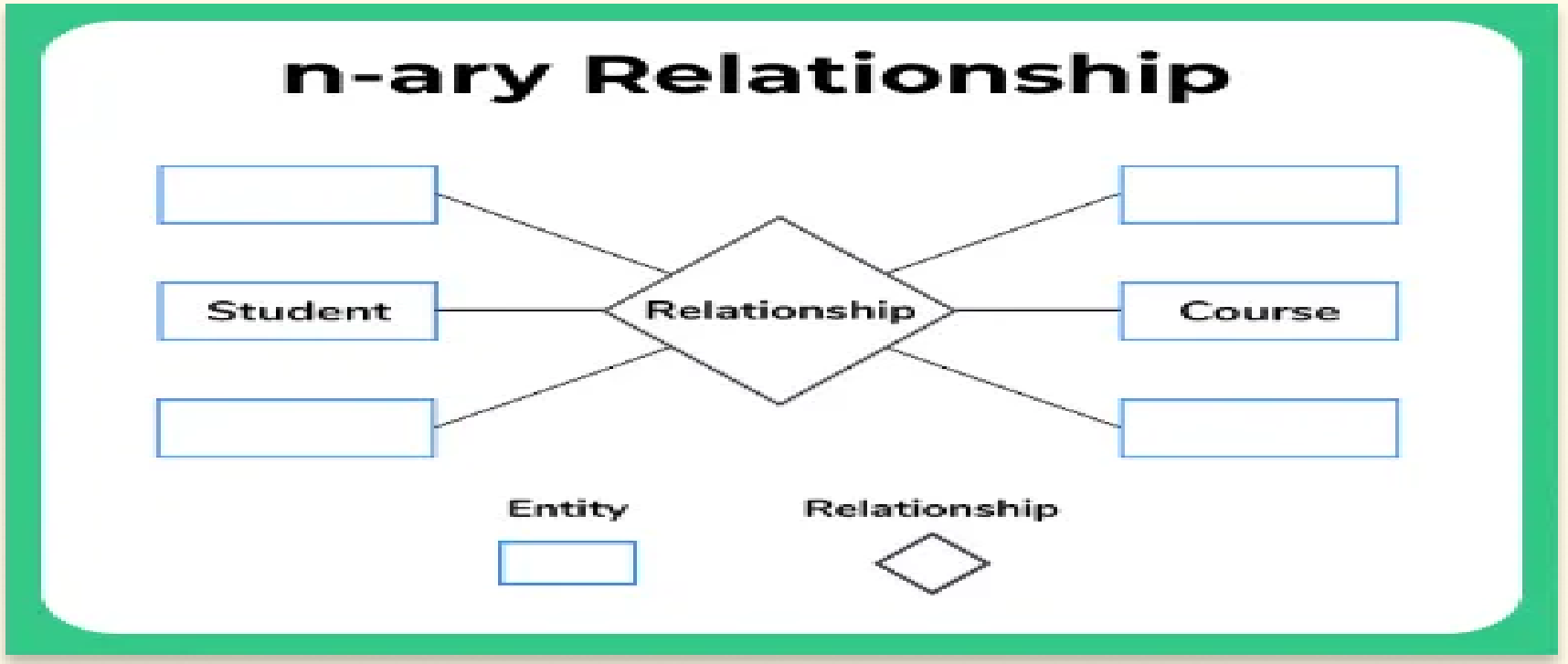
2. Binary Relationship: When there are TWO entities set participating in a relationship, the relationship is called a binary relationship. For example, a Student is enrolled in a Course.



3. Ternary Relationship: When there are three entity sets participating in a relationship, the relationship is called a ternary relationship.



4. N-ary Relationship: When there are n entities set participating in a relationship, the relationship is called an n-ary relationship.
When a relationship connects **three or more entity sets**.



Roles of Structural Constraints in ER Model

Structural constraints define **rules on relationships** in ER diagrams. They specify how many entities can participate in a relationship and maintain **data integrity**.

Main Roles

1.Cardinality (Mapping) Constraint

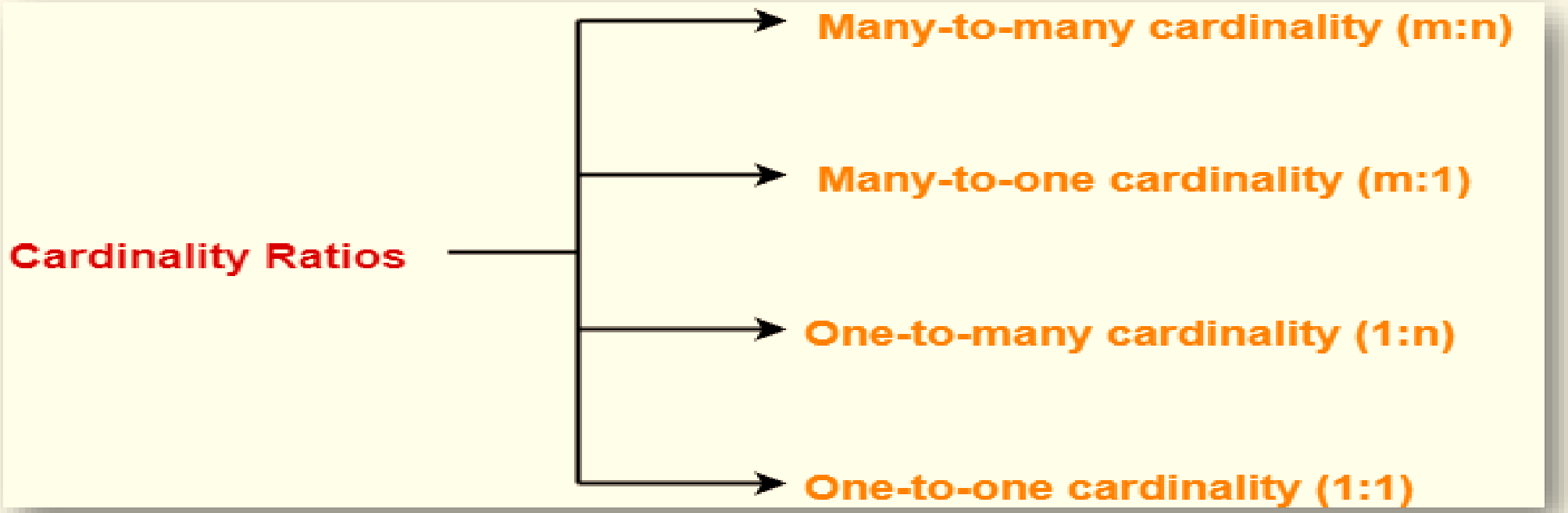
1. Defines the **maximum number** of entity occurrences that can be related.
2. Example: *One-to-One (1:1), One-to-Many (1:N), Many-to-Many (M:N)*.

2.Participation Constraint

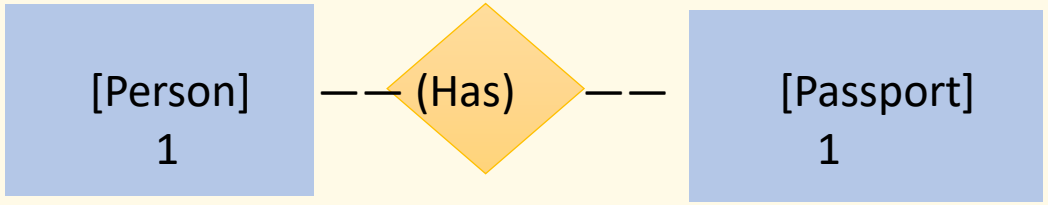
1. Defines the **minimum number** of entity occurrences that must participate (Total or Partial).
2. Example: *Employee must belong to at least one department (Total participation)*.

What is Cardinality?

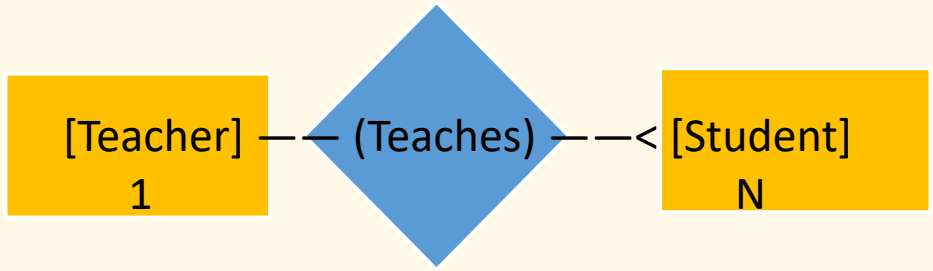
The number of times an entity of an entity set participates in a relationship set is known as cardinality. Cardinality can be of different types:



- **One-to-One (1:1)** → One person has **one passport**.
- **One-to-Many (1:N)** → One teacher teaches **many students**.
- **Many-to-One (N:1)** → Many students study in **one college**.
- **Many-to-Many (M:N)** → Students can take **many courses**, and each course has many students.

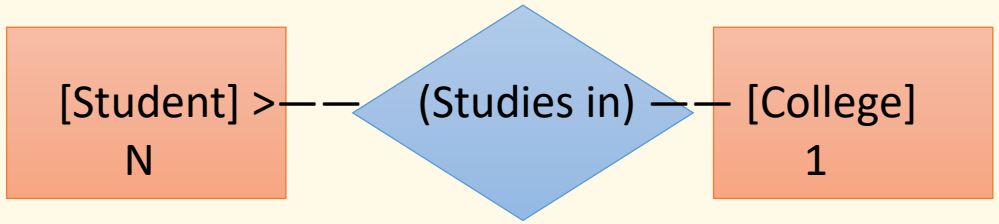


One-to-One (1:1)

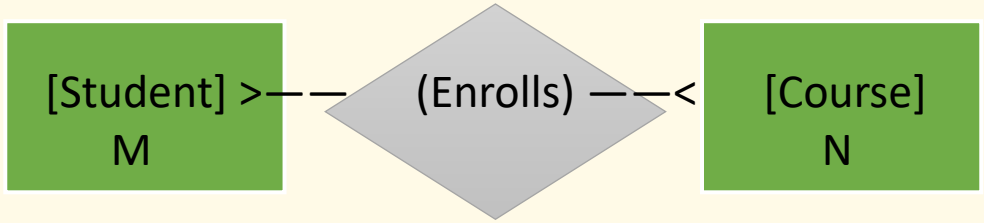


One-to-Many (1:N)

Unit - II E-R Modeling



- **Many-to-One (N:1)**



- **Many-to-Many (M:N)**

✅ 1. Super Class

A general entity that stores the common features of multiple related entities.
It acts like a parent class.
Example : Vehicle is a superclass because all vehicles have some common properties like vehicle no, color, and model.

2. Sub Class

A specific entity created from a superclass.
It contains extra attributes apart from what it inherited.
Example : Car and Bike are subclasses of Vehicle.
Car may have no_of_seats, and Bike may have engine_type.

3. Inheritance

The process of copying attributes and relationships from superclass to subclass.

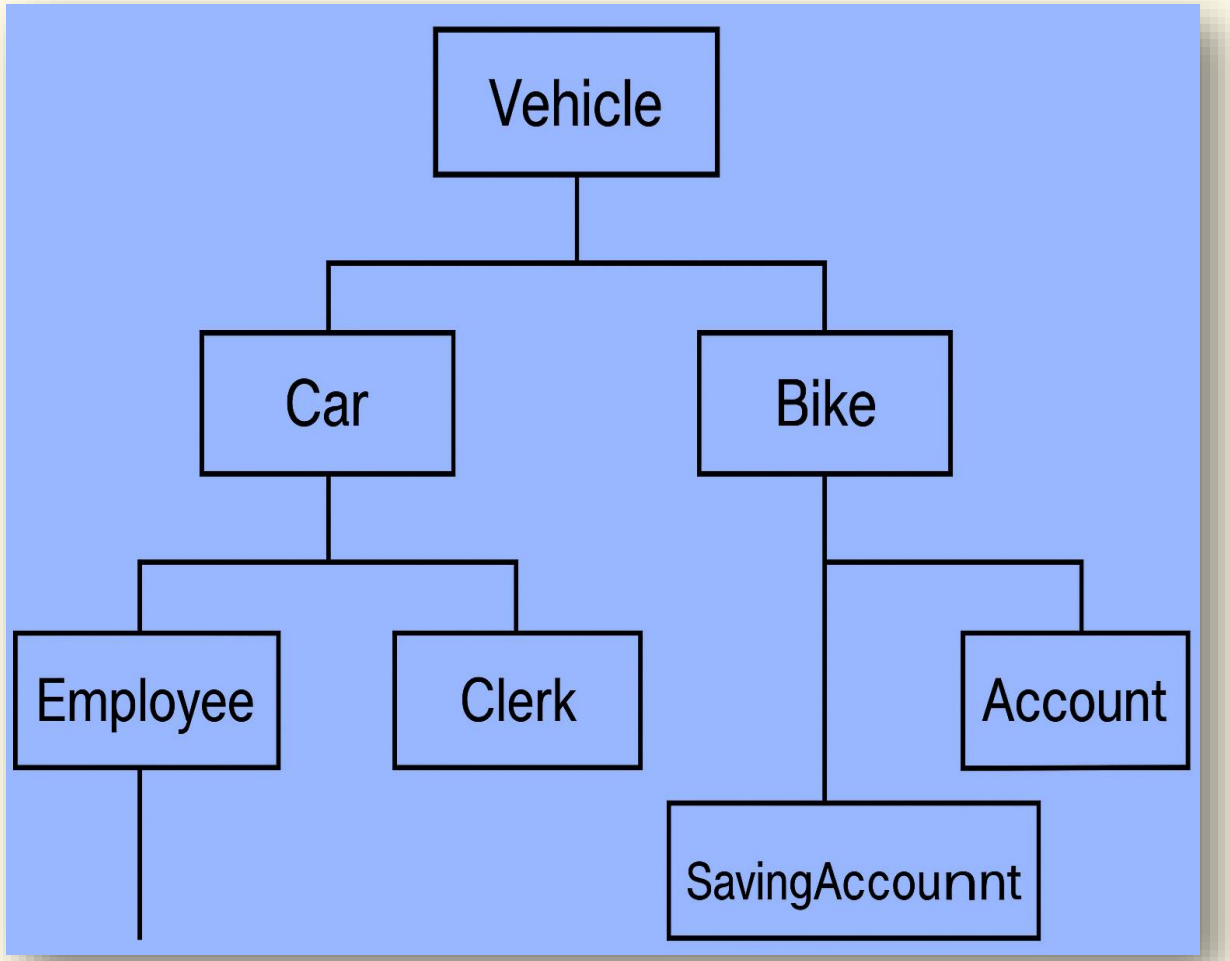
It saves duplication because common attributes are defined only once in the superclass.

Example: Car and Bike both inherit vehicle _no and color from Vehicle.

4. Specialization

A Top-Down approach: Start with one broad entity and create more specific entities (subclasses) based on some differences.

Example : From Employee superclass, create subclasses Manager, Clerk, and Engineer.



5. Generalization

A Bottom-Up approach: Start with several similar entities and combine them into one superclass.

Example : Combine Car and Bike into a single superclass called Vehicle.

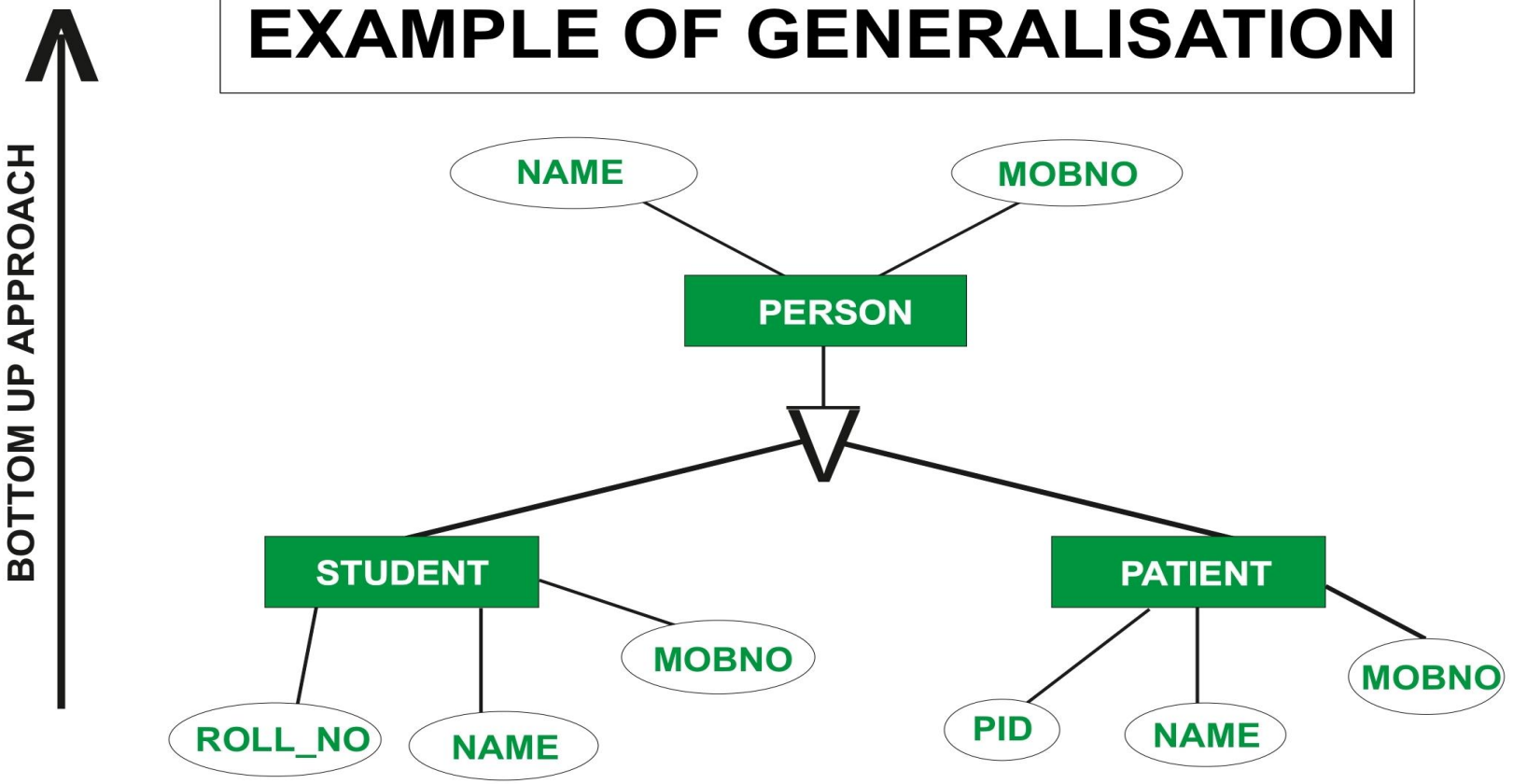
Example of Generalization

Consider two entities Student and Patient.
These two entities will have some characteristics of their own.

For example, the Student entity will have Roll_No, Name, and Mob_No while the patient will have PId, Name, and Mob_No characteristics.

Now in this example Name and Mob_No of both Student and Patient can be combined as a Person to form one higher-level entity and this process is called as Generalization Process.

EXAMPLE OF GENERALISATION



Aggregation in ER Model :

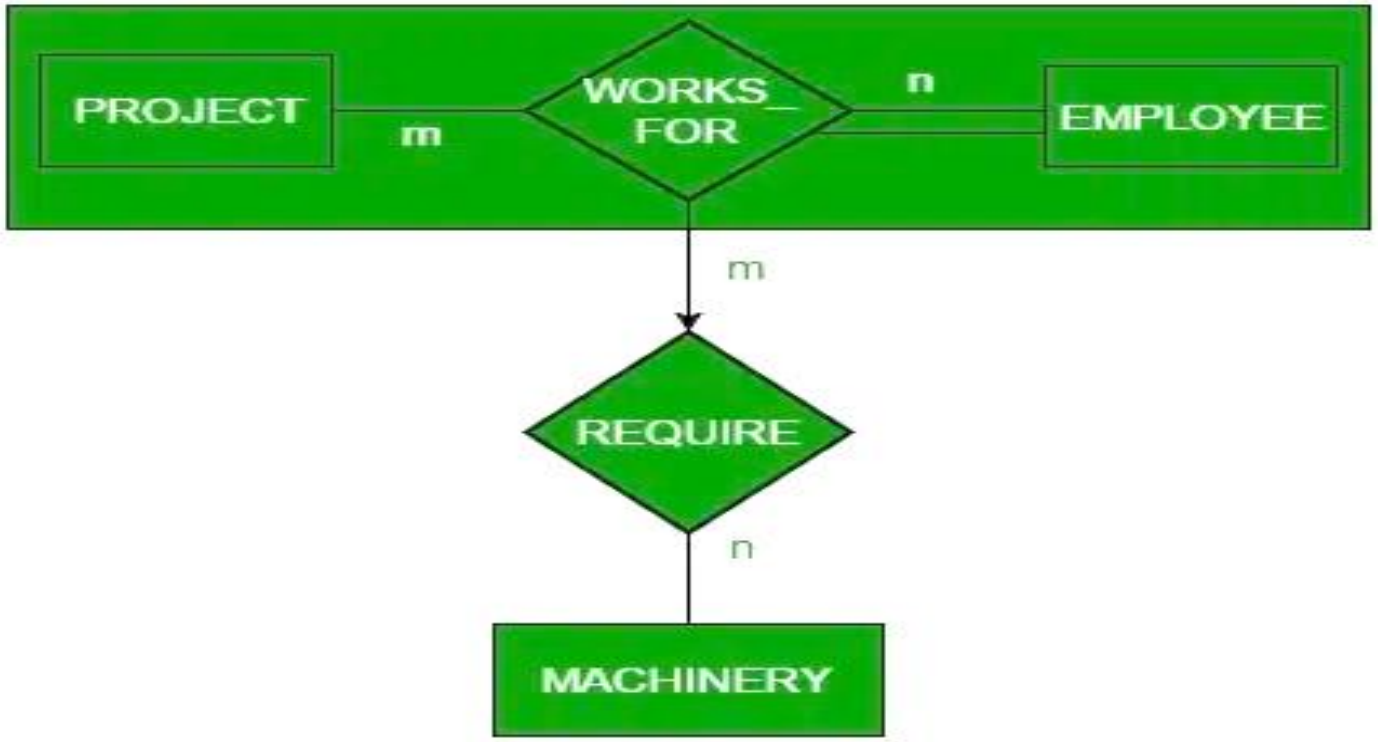
Aggregation is a concept used when we need to show a relationship between a **relationship** and an **entity**. It is used to represent **"Has-a"** or **"Part-of"** relationships.

✅ **Example:**

•Suppose **Employee** works on a **Project**, and **Project** is controlled by a **Department**.

•Here, the relationship **"Works_On"** between Employee and Project is treated as an entity and connected to Department.

Unit - II E-R Modeling



Aggregation

For Example, an Employee working on a project may require some machinery. So, REQUIRE relationship is needed between the relationship WORKS_FOR and entity MACHINERY. Using aggregation, WORKS_FOR relationship with its entities EMPLOYEE and PROJECT is aggregated into a single entity and relationship REQUIRE is created between the aggregated entity and MACHINERY.

Generalization	**Specialization**
Works in **Bottom-Up** approach	Works in **Top-Down** approach
Combines two or more entities into a **superclass**	Divides one entity into **subclasses**
Schema size decreases	**Schema size increases**
Focus: **Common features**	Focus: **Unique features**
Example: Car + Bike → Vehicle	Example: Employee → Teacher, Clerk

Schema size decreases = Combining many entities into one (Generalization), so fewer tables.
Schema size increases = Splitting one entity into many (Specialization), so more tables.

Advantages of ER Model

1.Simple and Easy to Understand – Uses diagrams, so concepts are clear.

2.Visual Representation – Shows entities, attributes, and relationships graphically.

3.Better Communication – Helps users and developers discuss system requirements easily.

4.Foundation for Database Design – Acts as a blueprint for relational databases.

5.Captures Real-World Relationships – Represents how data is related in reality.

Disadvantages of ER Model

1. **Not for Complex Queries** – Cannot show detailed operations like joins.

2. **No Standardization** – Different designers may create different ER diagrams for the same system.
3. **Ignores Implementation Details** – Does not consider performance, indexing, etc.

4. **Difficult for Large Systems** – Becomes complex and hard to manage with many entities.

5. **Lacks Support for Some Constraints** – Cannot fully represent all business rules.

CONCEPT DESIGN WITH E-R MODEL

AIM: To Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong and weak
entities. Indicate the type of relationships (total/partial). Incorporate generalization, aggregation and specialization
etc wherever required.

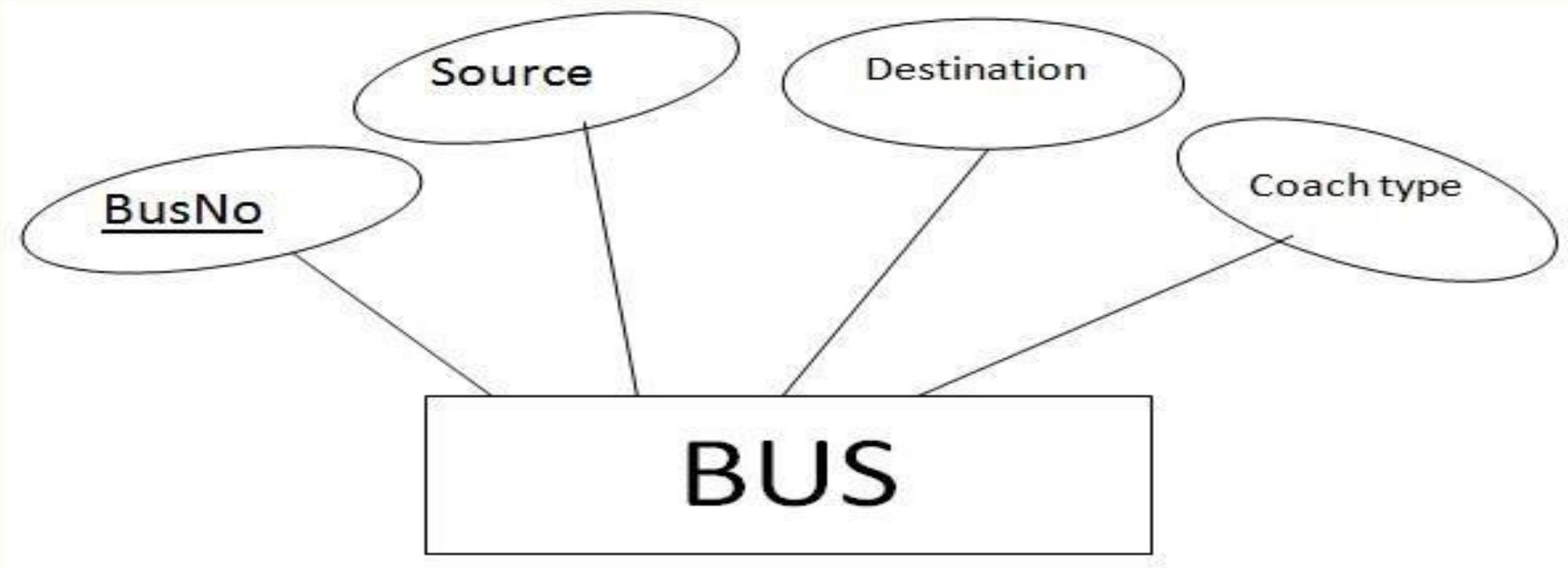
E-R Model

- Bus
- Bus No
- Source
- Destination
- Coach Type

E-R Model

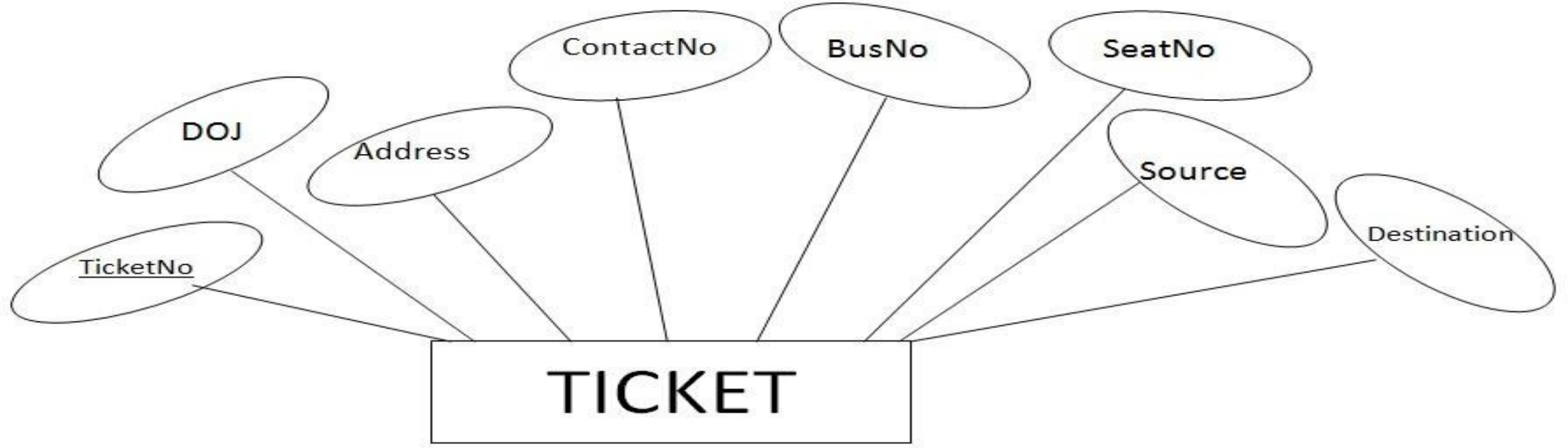
Entity: Bus

Attributes: Bus No, Source, Destination, Coach Type.



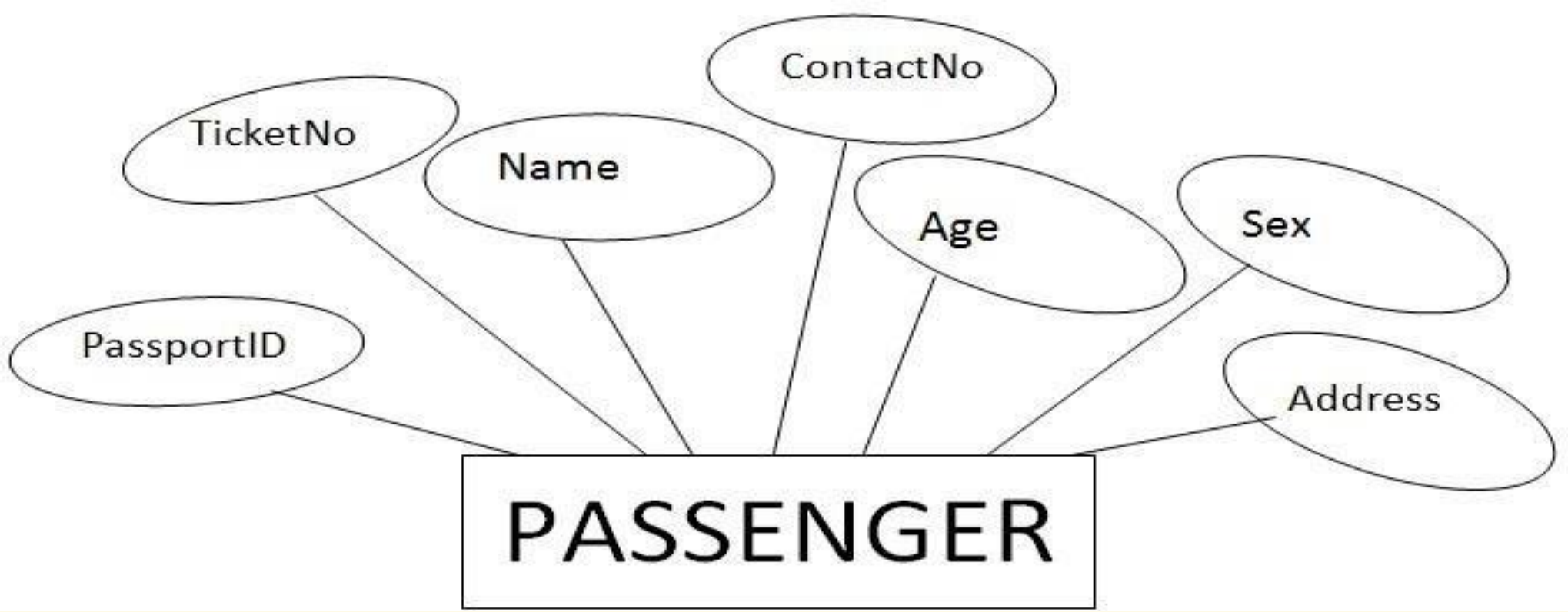
Unit - II E-R Modeling

Entity: Ticket
Attributes: TicketNo, DOJ, Address, ContactNo, BusNo, SeatNo, Source, Destination

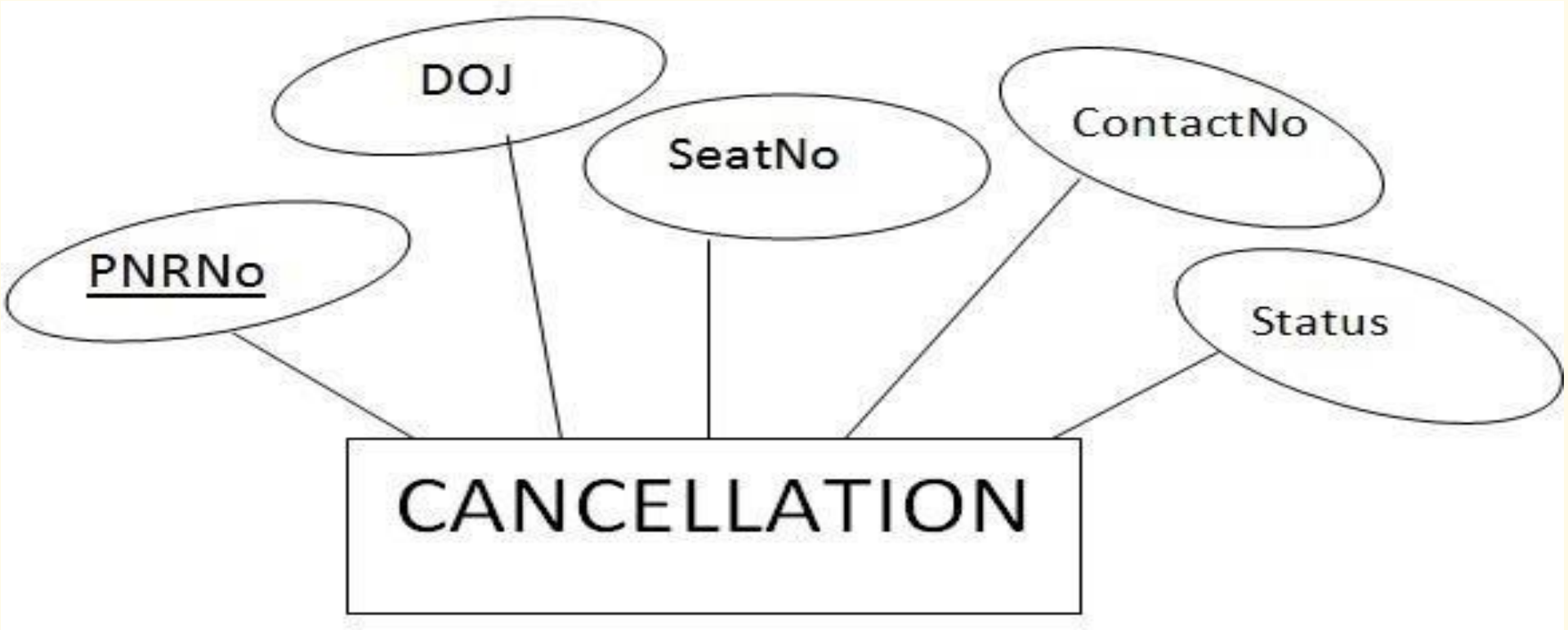


Unit - II E-R Modeling

Entity: Passenger
Attributes: Passport ID, Ticket No, Name, Contact No, Age, Sex, Address



Entity: Cancellation
Attributes: PNR No, DOJ, Seat No, Contact No, Status



E-R Diagram of Library Management System