

# **Entity-Relationship Model**

**Dr. Seema Gupta Bhol**

# Entity Relationship Model

- Models an enterprise as a collection of *entities* and *relationships*
  - ▶ Entity: a “thing” or “object” in the enterprise that is distinguishable from other objects
    - Described by a set of *attributes*
  - ▶ Relationship: an association among several entities
- Represented diagrammatically by an *entity-relationship diagram*:

# ER model

- The ER data model employs three basic concepts:
  - entity sets,
  - relationship sets,
  - attributes.
- The ER model also has an associated diagrammatic representation, the ER diagram, which can express the overall logical structure of a database graphically.

# Entity Sets

- An **entity** is an object that exists and is distinguishable from other objects.
  - Example: specific person, company, event, plant
- An **entity set** is a set of entities of the same type that share the same properties.
  - Example: set of all persons, companies, trees, holidays
- An entity is represented by a set of attributes; i.e., descriptive properties possessed by all members of an entity set.
  - Example:  
$$\text{instructor} = (ID, name, street, city, salary)$$
$$\text{course} = (course\_id, title, credits)$$
- A subset of the attributes form a **primary key** of the entity set; i.e., uniquely identifying each member of the set.

# Entity Sets -- *instructor* and *student*

instructor\_ID instructor\_name

76766	Crick
45565	Katz
10101	Srinivasan
98345	Kim
76543	Singh
22222	Einstein

*instructor*

student-ID student\_name

98988	Tanaka
12345	Shankar
00128	Zhang
76543	Brown
76653	Aoi
23121	Chavez
44553	Peltier

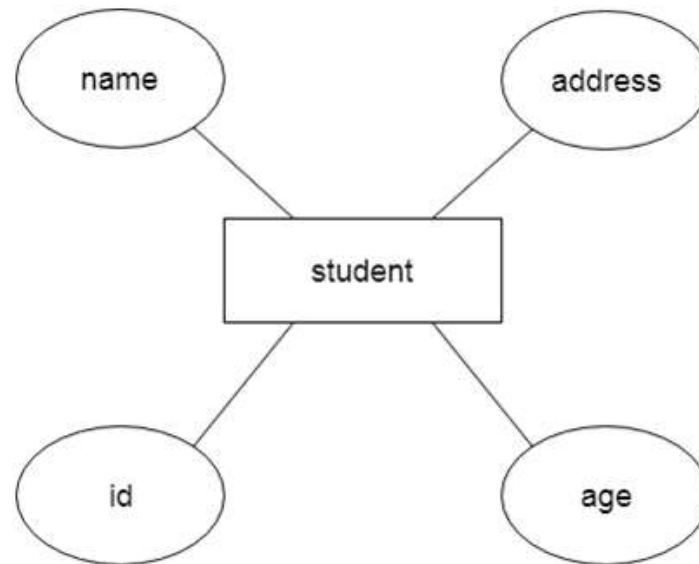
*student*

# Attributes

The attribute is used to describe the property of an entity.

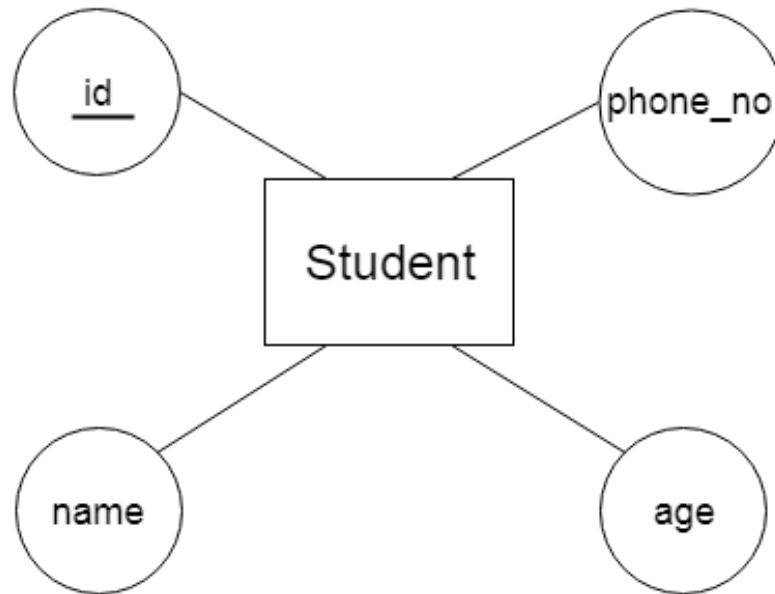
Eclipse is used to represent an attribute.

**For example,** id, age, contact number, name, etc. can be attributes of a student.



# 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.



- Id is key attribute for entity student .

# Attributes

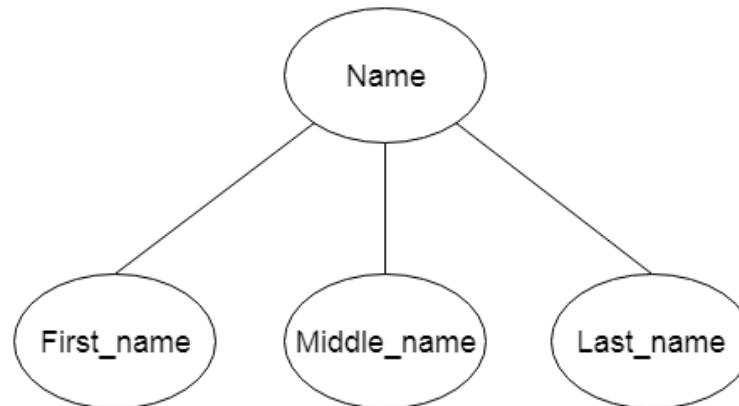
- Attribute types:
  - **Simple** and **composite** attributes.
  - **Single-valued** and **multivalued** attributes
    - ▶ Example: multivalued attribute: *phone\_numbers*
  - **Derived** attributes
    - ▶ Can be computed from other attributes
    - ▶ Example: age, given date\_of\_birth
- **Domain** – the set of permitted values for each attribute



# Composite Attributes

An attribute that composed of many other attributes is known as a composite attribute.

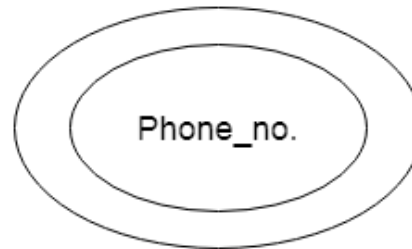
The composite attribute is represented by an ellipse, and those ellipses are connected with an ellipse.



# Multivalued Attribute

An attribute can have more than one value. These attributes are known as a multivalued attribute. The double oval is used to represent multivalued attribute.

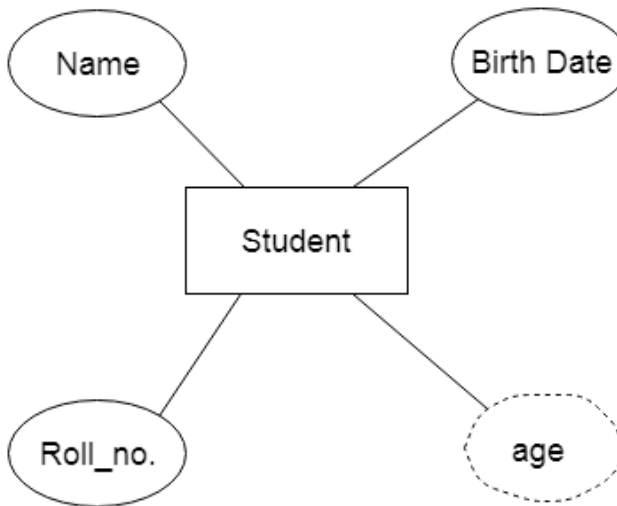
**For example,** a student can have more than one phone number.



# Derived Attribute

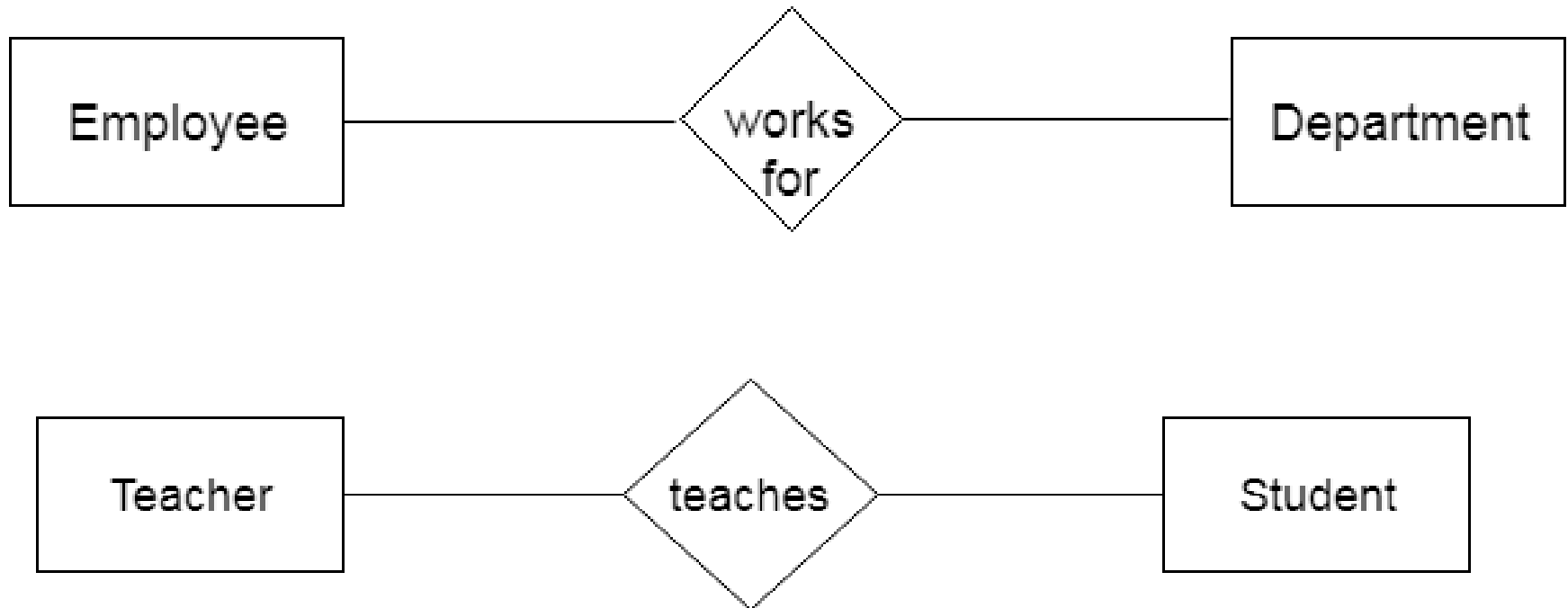
An attribute that can be derived from other attribute is known as a derived attribute. It can be represented by a dashed ellipse.

**For example,** A person's age changes over time and can be derived from another attribute like Date of birth.

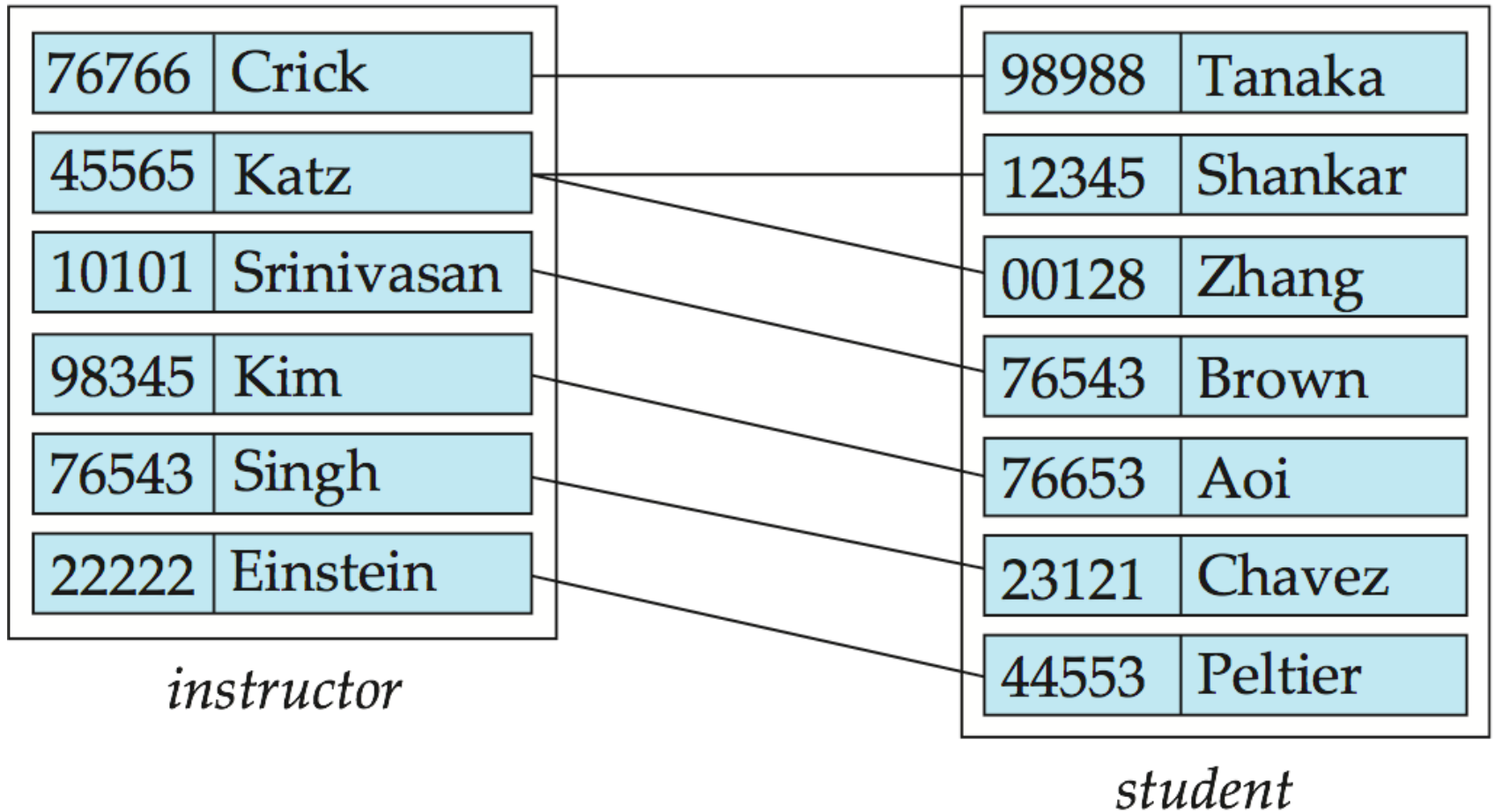


# Relationship

A Relationship is used to describe the relation between entities. Diamond or rhombus is used to represent the relationship.

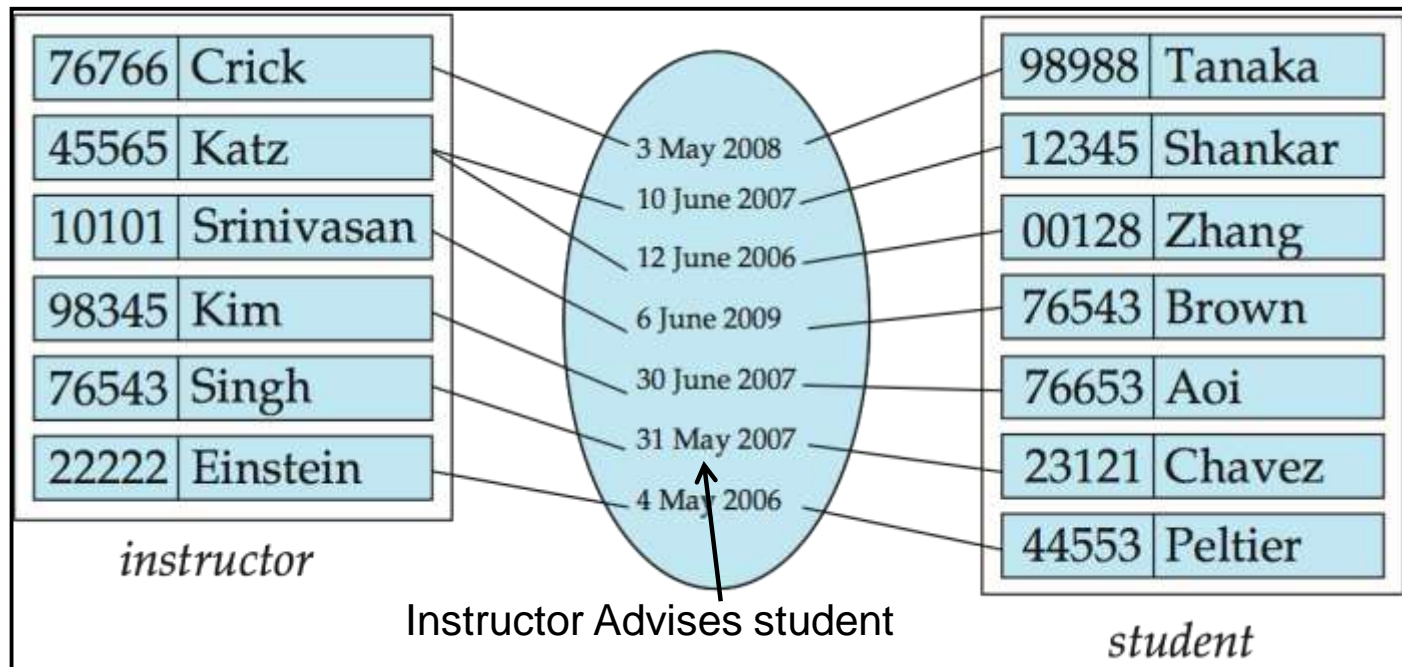


# Relationship Set *advisor*



# Relationship Sets

- An attribute can also be associated with a relationship set.
- For instance, the *advisor* relationship set between entity sets *instructor* and *student* may have the attribute *date* which tracks when the student started being associated with the advisor



# Relationship Sets

- A **relationship** is an association among several entities

Example:

44553 (Peltier)	<u>advisor</u>	22222 (Einstein)
<i>student</i> entity	relationship set	<i>instructor</i> entity

- A **relationship set** is a mathematical relation among  $n \geq 2$  entities, each taken from entity sets

$$\{(e_1, e_2, \dots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$$

where  $(e_1, e_2, \dots, e_n)$  is a relationship

- Example:

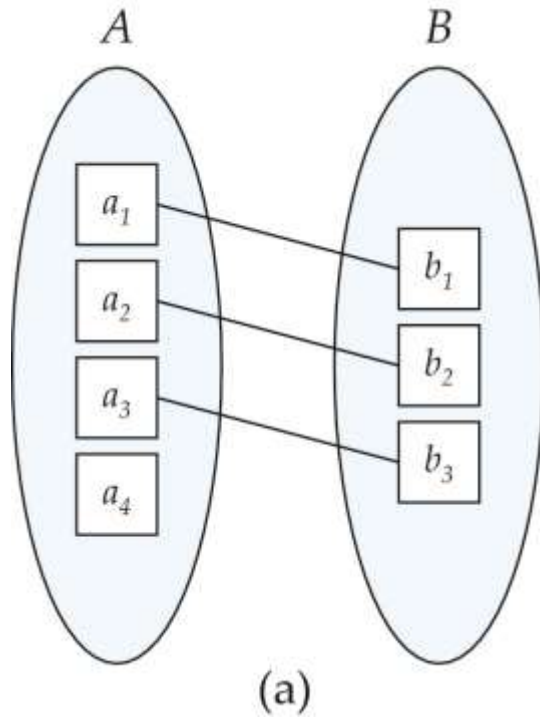
$$(44553, 22222) \in \text{advisor}$$

# Mapping Cardinality Constraints

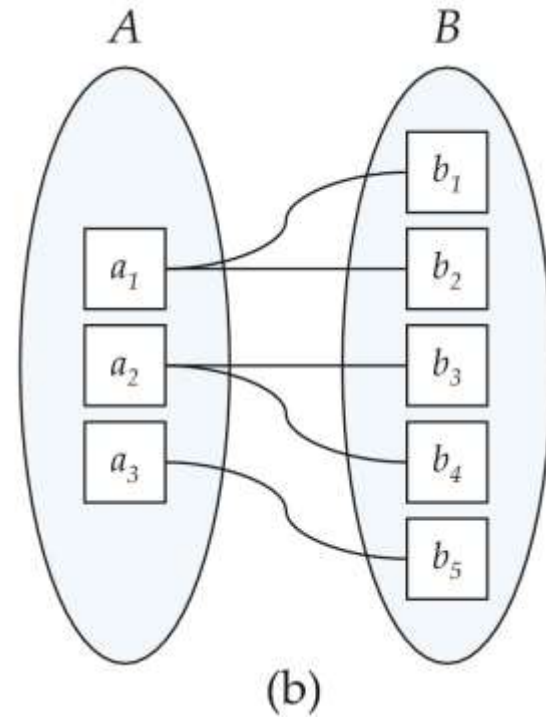
- Express the number of entities to which another entity can be associated via a relationship set.
- Most useful in describing binary relationship sets.
- For a binary relationship set the mapping cardinality must be one of the following types:
  - One to one
  - One to many
  - Many to one
  - Many to many



# Mapping Cardinalities



One to one



One to many

Note: Some elements in  $A$  and  $B$  may not be mapped to any elements in the other set

# One-to-One Relationship

When only one instance of an entity is associated with the relationship, then it is known as one to one relationship.

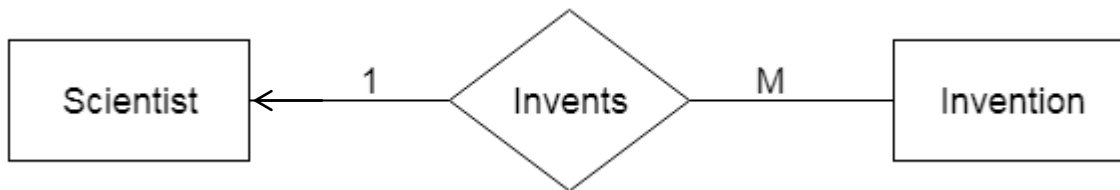
**For example,** A female can marry to one male, and a male can marry to one female.



# One-to-many relationship

When only one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then this is known as a one-to-many relationship.

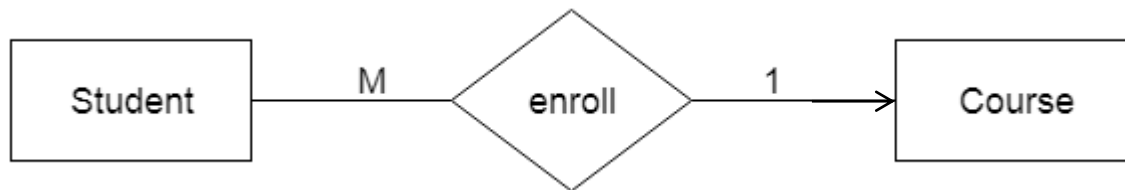
**For example,** Scientist can invent many inventions, but the invention is done by the only specific scientist.



# Many-to-one relationship

When more than one instance of the entity on the left, and only one instance of an entity on the right associates with the relationship then it is known as a many-to-one relationship.

**For example,** Student enrolls for only one course, but a course can have many students.



# Many-to-many relationship

When more than one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then it is known as a many-to-many relationship.

**For example,** Employee can assign by many projects and project can have many employees.



# Degree of a Relationship Set

## ■ Binary relationship

- involve two entity sets (or degree two).
- most relationship sets in a database system are binary.

## ■ Ternary Relationships :

between more than three entity sets .

- ▶ Example: *students* work on research *projects* under the guidance of an *instructor*.
- ▶ relationship *proj\_guide* is a ternary relationship between *instructor*, *student*, and *project*

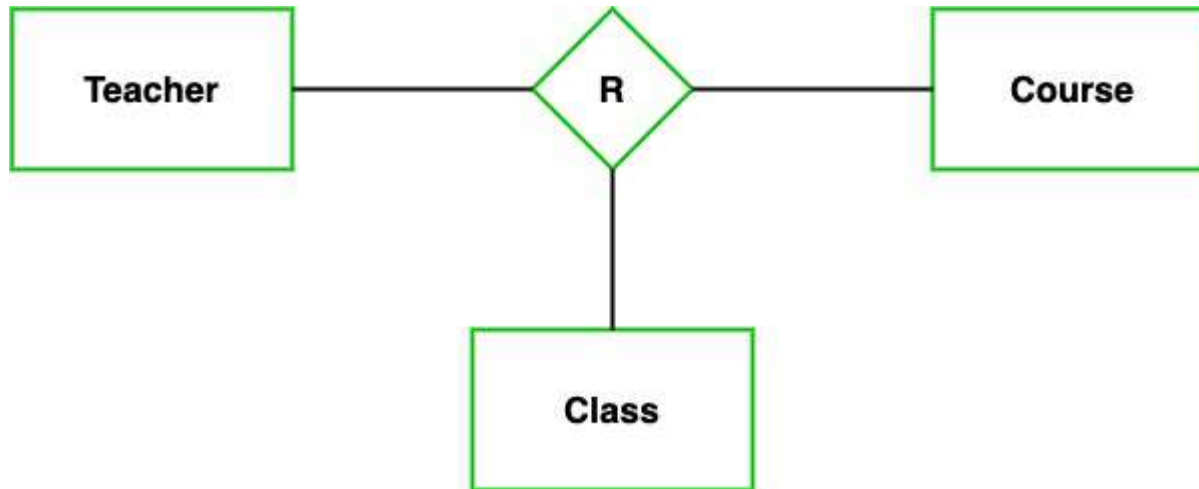
# Ternary relationship

- In the Ternary relationship, there are three types of entity associates.
- Ternary relationship exists when there are three types of entity and we call them a degree of relationship is 3.

**Example:** Consider three entity types 'Teacher', 'Course', and 'Class'.

The relationship between these entities is defined as the teacher teaching a particular course, also the teacher teaches a particular class.

# Ternary relationship





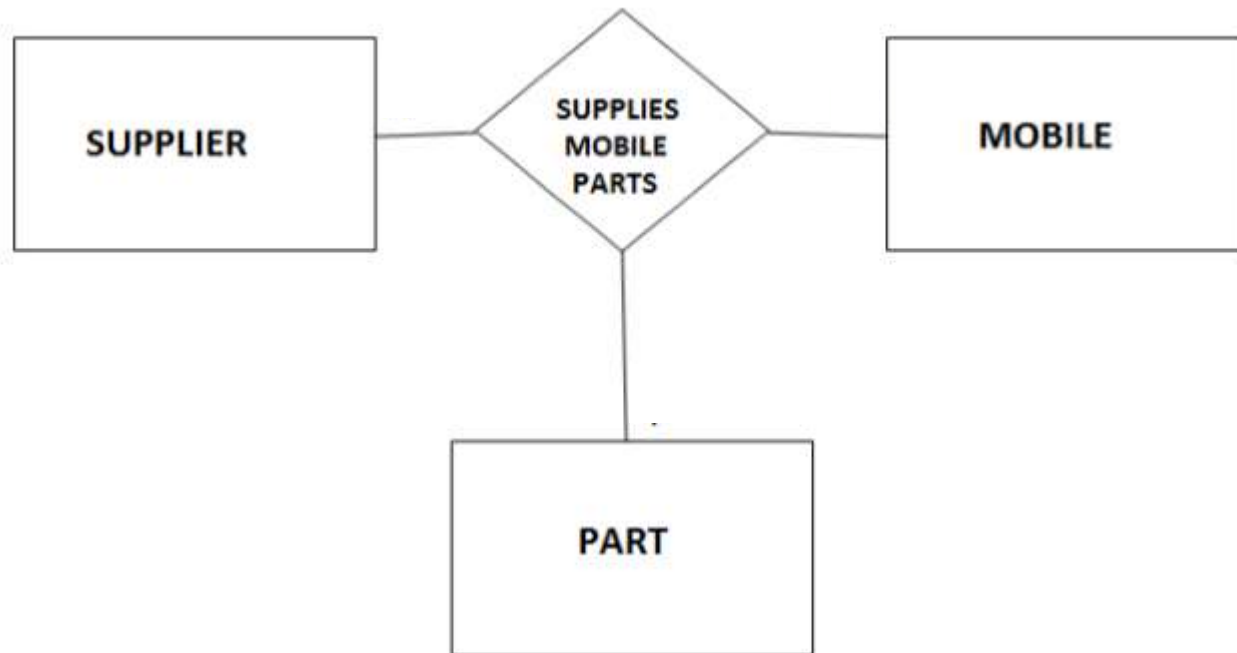
## Example 2

Three different entities involved:

Mobile - Manufactured by company.

Part - Mobile Part which company get from Supplier.

Supplier - Supplier supplies Mobile parts to Company.

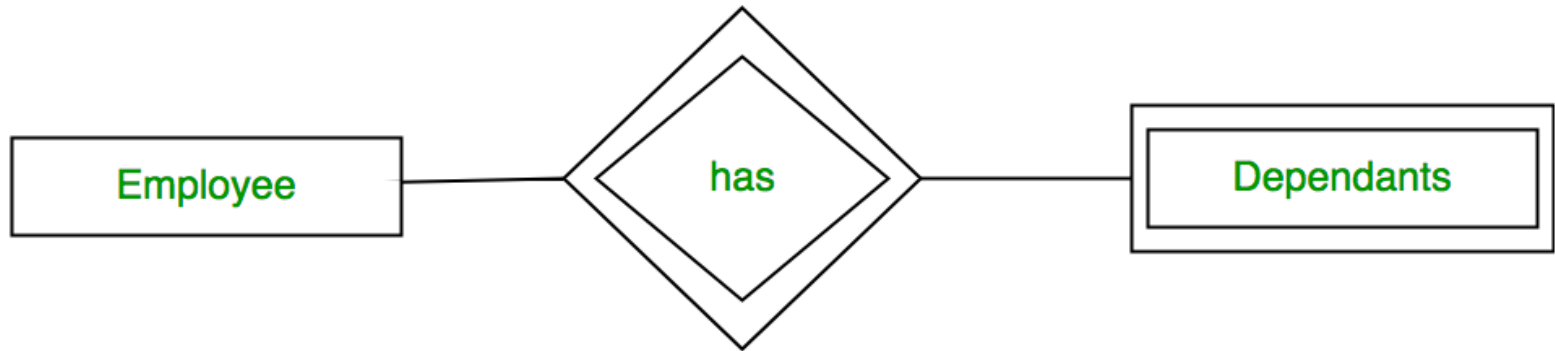


# weak entity sets

- An entity type should have a key attribute which uniquely identifies each entity in the entity set, but there exists some entity type for which key attribute can't be defined. These are called Weak Entity type.
- The entity sets which do not have sufficient attributes to form a primary key are known as **weak entity sets** and the entity sets which have a primary key are known as strong entity sets.
- As the weak entities do not have any primary key, they cannot be identified on their own, so they depend on some other entity (known as owner entity).
- Weak entity types have partial keys. Partial Keys are set of attributes with the help of which the tuples of the weak entities can be distinguished and identified.

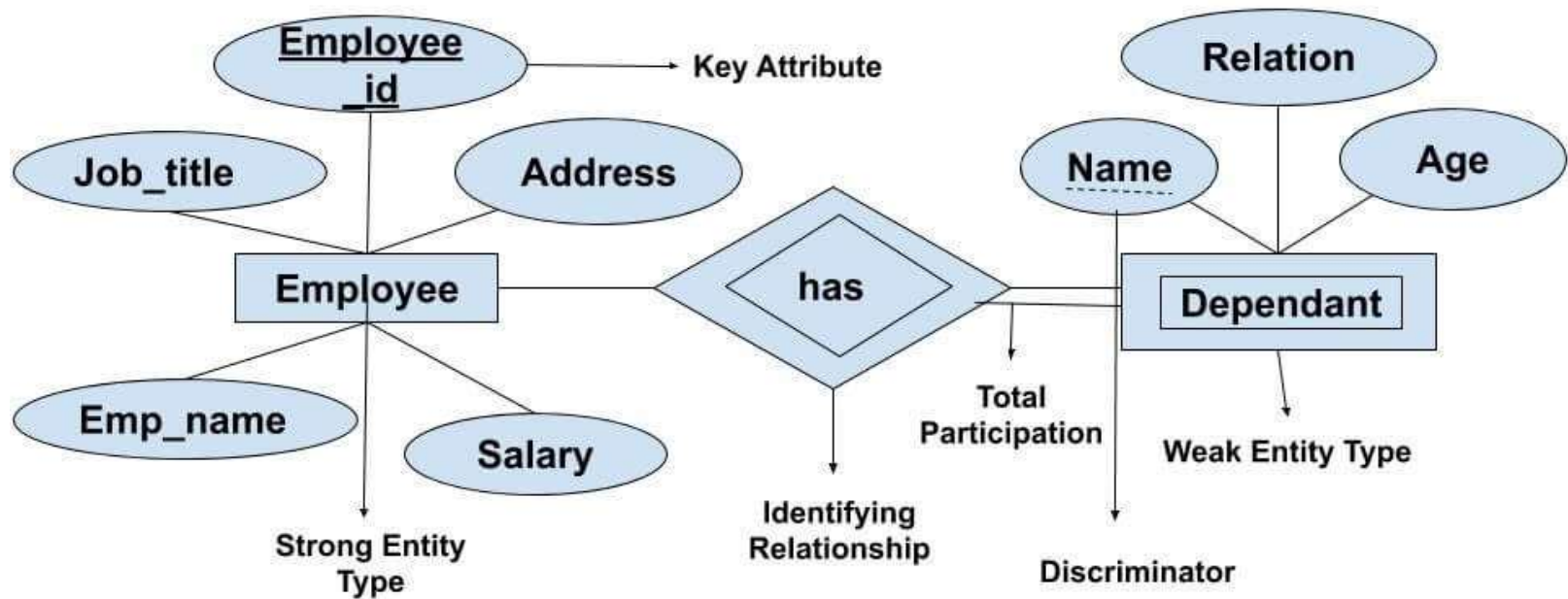
# Weak Entity Sets

**Weak entities** are represented with **double rectangular** box in the ER Diagram and the identifying relationships are represented with double diamond. Partial Key attributes are represented with dotted lines.



we use the identifying entity, along with extra attributes called **discriminator** to uniquely identify a weak entity. An entity set that is not a weak entity set is termed a **strong entity set**.

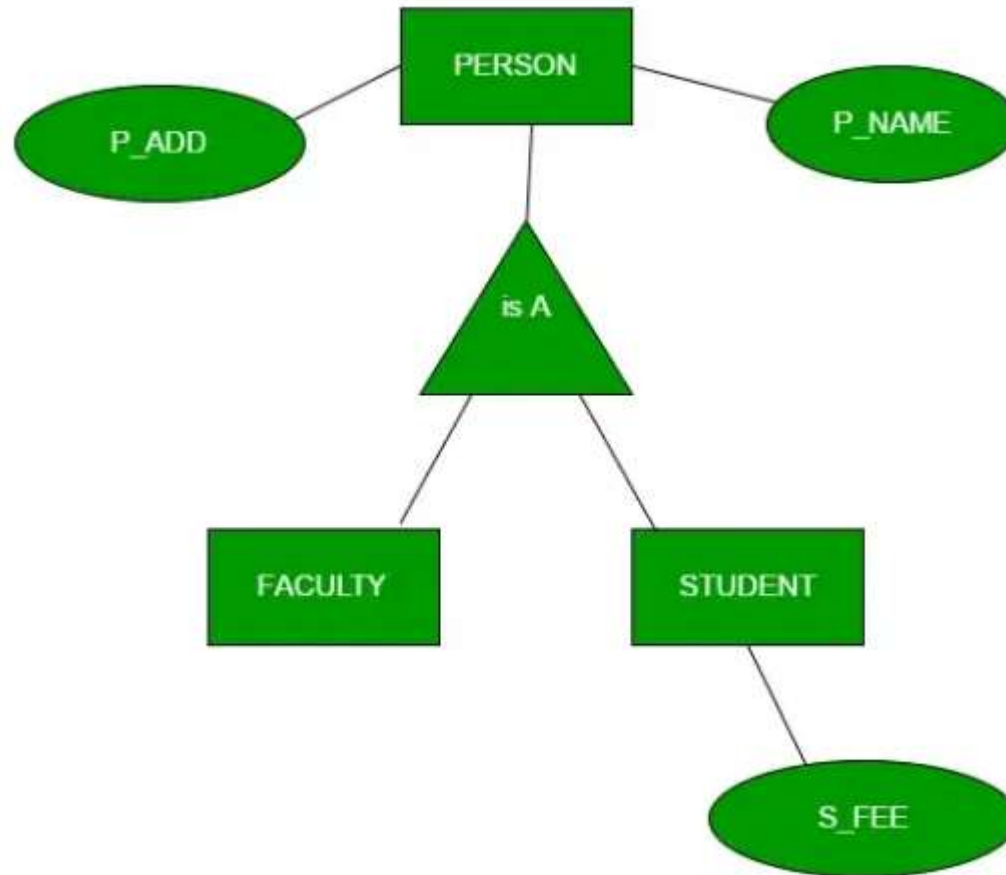
# Weak entity set



# Generalization

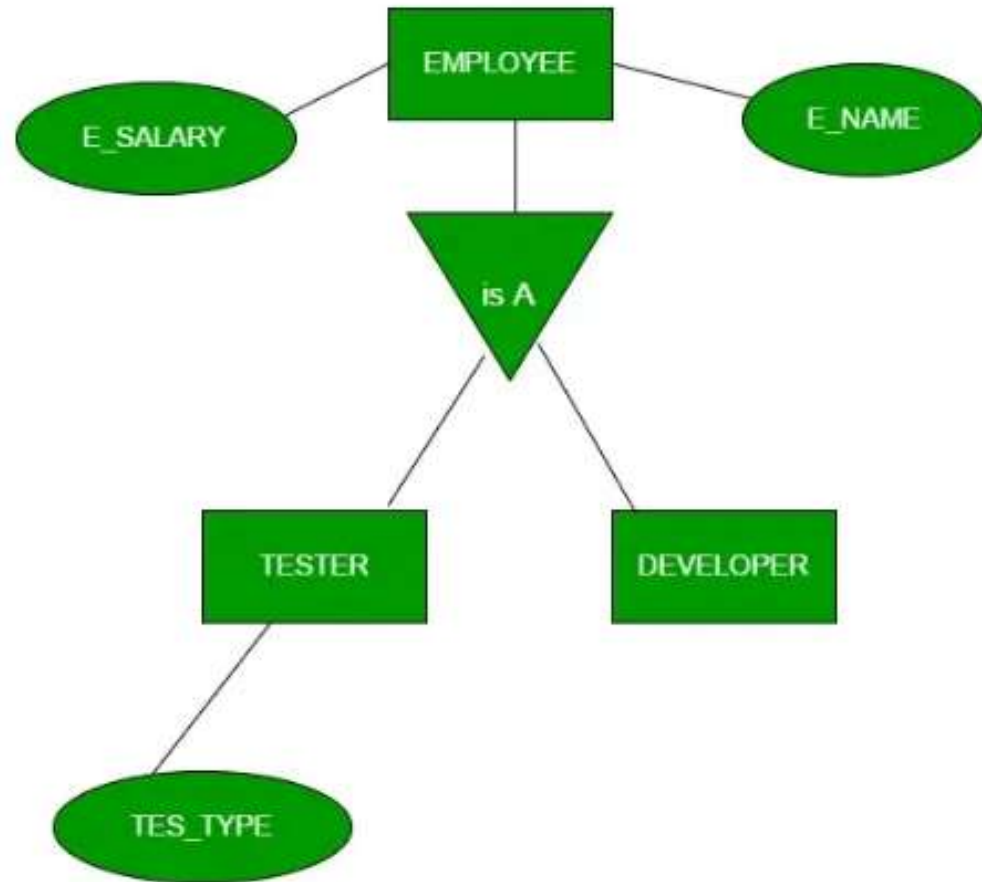
- Generalization is like a bottom-up approach in which two or more entities of lower level combine to form a higher level entity if they have some attributes in common.
- In generalization, an entity of a higher level can also combine with the entities of the lower level to form a further higher level entity.
- **For example**, Faculty and Student entities can be generalized and create a higher level entity Person.

# Generalization



# Specialization

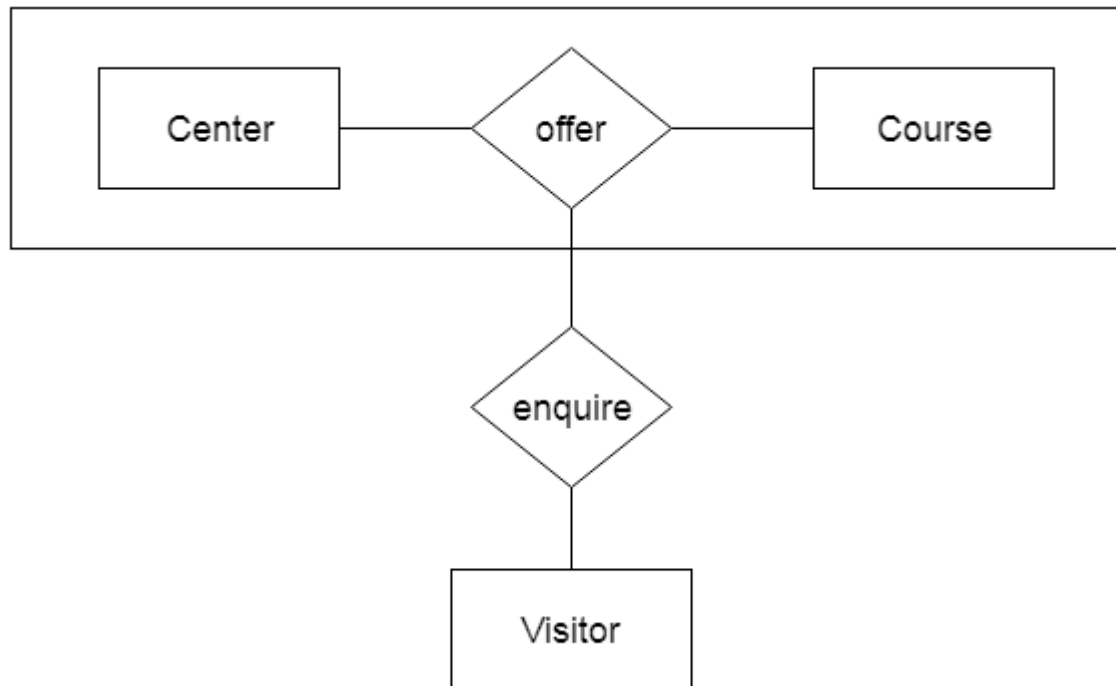
- Specialization is a top-down approach, and it is opposite to Generalization. In specialization, one higher level entity can be broken down into two lower level entities.
- Specialization is used to identify the subset of an entity set that shares some distinguishing characteristics.
- **For example:** In an Employee management system, EMPLOYEE entity can be specialized as TESTER or DEVELOPER based on what role they play in the company.





# Aggregation

- In aggregation, the relation between two entities is treated as a single entity. In aggregation, relationship with its corresponding entities is aggregated into a higher level entity.
- **For example:** Center entity offers the Course entity act as a single entity in the relationship which is in a relationship with another entity visitor. In the real world, if a visitor visits a coaching center then he will never enquiry about the Course only or just about the Center instead he will ask the enquiry about both.



# Aggregation

A manager not only manages the employee working under them but he has to manage the project as well.

In these cases the relationship of two entities acts as one entity.

The relationship “Works-On” between “Employee” & “Project” acts as one entity that has a relationship “Manages” with the entity “Manager”.

