Sanjivani Rural Education Society's

Sanjivani College of Engineering, Kopargaon-423603

(An Autonomous Institute Affiliated to Savitribai Phule Pune University, Pune)

NACC 'A' Grade Accredited, ISO 9001:2015 Certified

Department of Information Technology

(NBA Accredited)

Database Management System (DBMS)

SY IT

Prof. Bhakti Pawar

Entity-Relationship Model

- ER diagram or **Entity Relationship diagram** is a conceptual model that gives the graphical representation of the logical structure of the database.
- It shows all the constraints and relationships that exist among the different components.

- https://www.gatevidyalay.com/er-diagrams/

Components of ER diagram

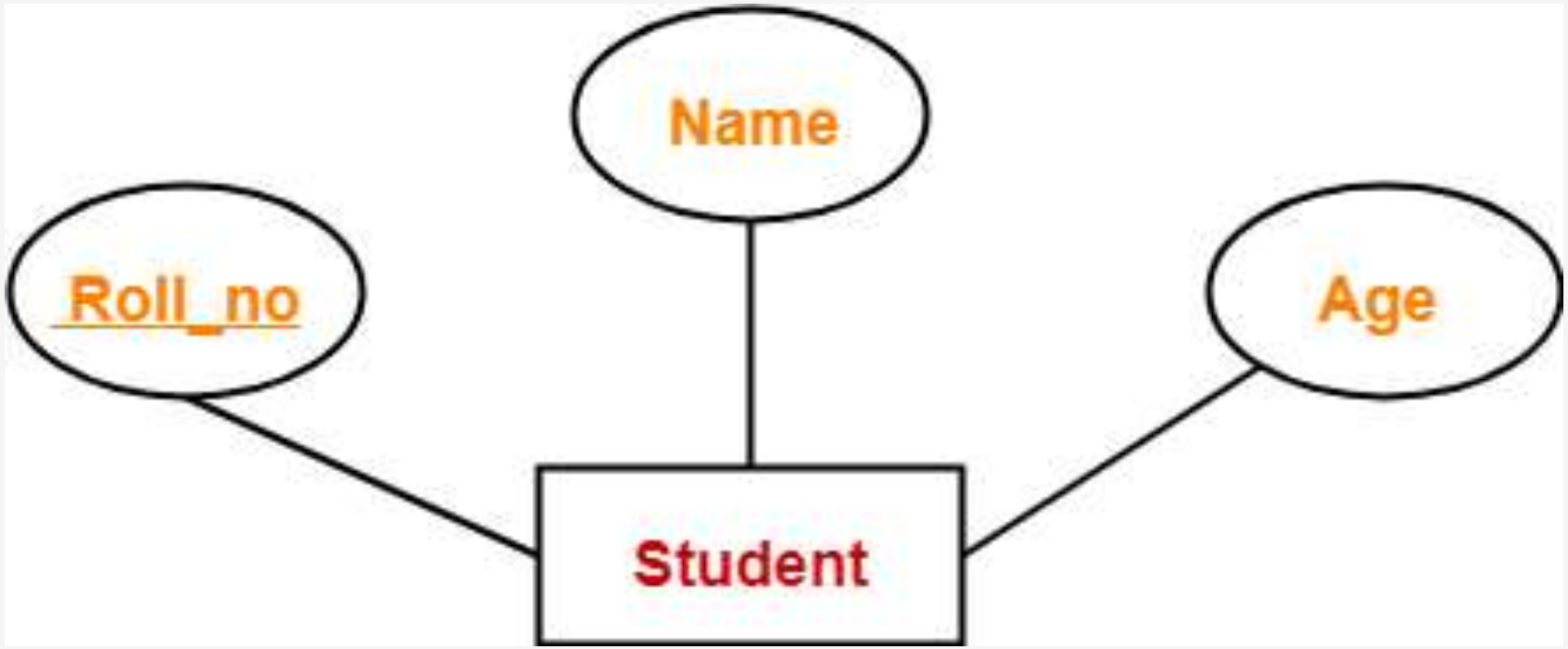
- An ER diagram is mainly composed of following three components-
- Entity Sets
- Attributes
- Relationship Set
- Example-

Consider the following Student table-

Roll_no	Name	Age
1	Akshay	20
2	Rahul	19
3	Pooja	20
4	Aarti	19

This complete table is referred to as "Student Entity Set" and each row represents an "entity".

Representation as ER Diagram-



Roll_no is a primary key that can identify each entity uniquely.
Thus, by using student's roll number, a student can be identified uniquely.

ER Diagram Symbols-

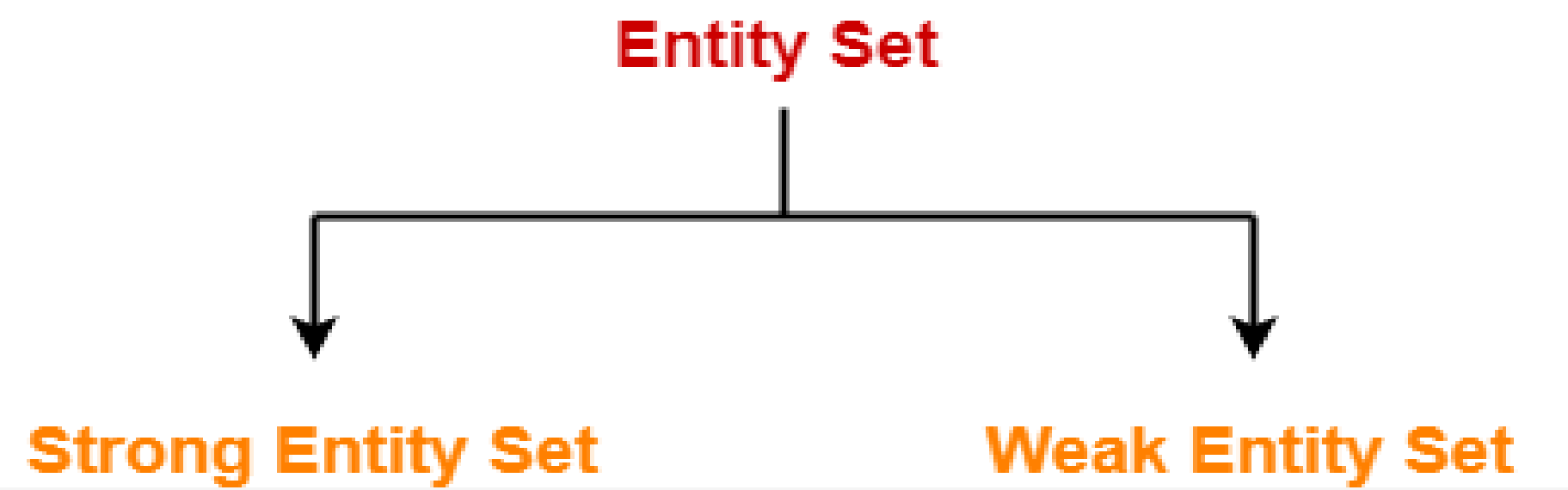
1. For Entity Sets-

An entity set is a set of same type of entities.

An entity refers to any object having-

•Either a physical existence such as a particular person, office, house or car.

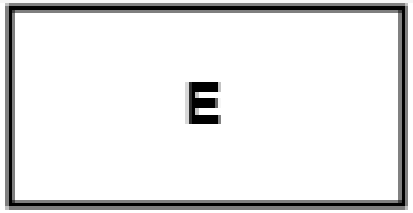
•Or a conceptual existence such as a school or a company.



ER Diagram Symbols-

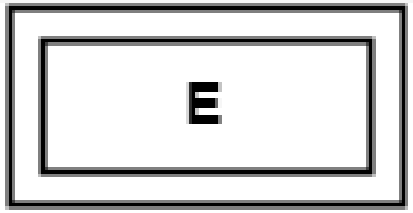
1. Strong Entity Set-

A strong entity set possess its own primary key.
It is represented using a single rectangle.



Strong Entity Set

2. Weak Entity Set-



Weak Entity Set

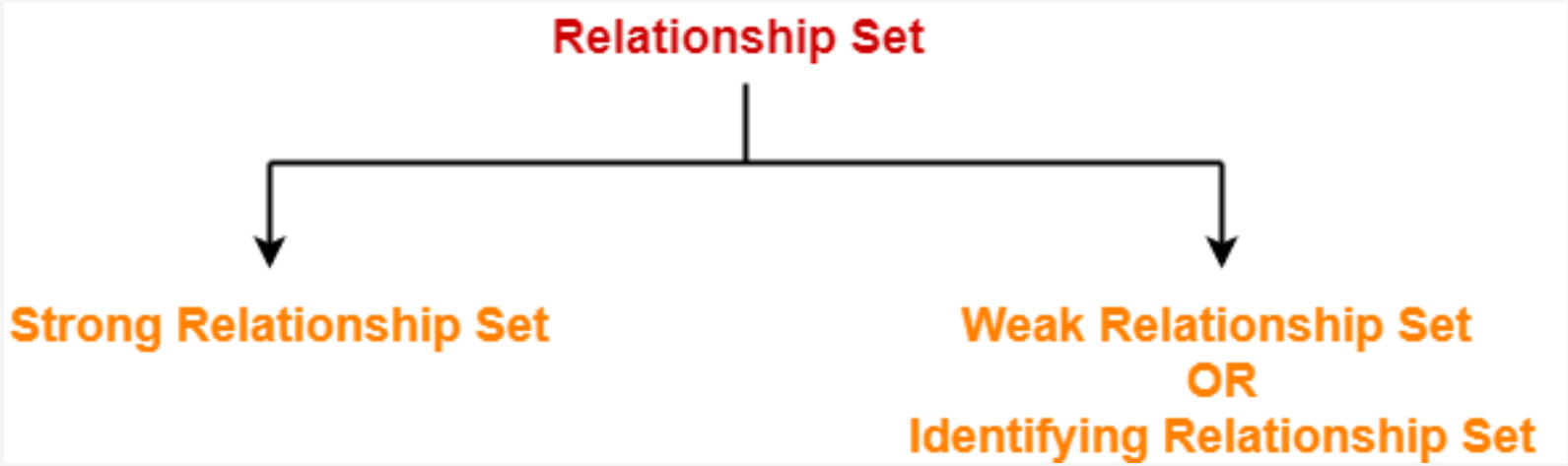
A weak entity set do not possess its own primary key.
It is represented using a double rectangle.

ER Diagram Symbols-

2. For Relationship Sets-

•Relationship defines an association among several entities.
•A relationship set is a set of same type of relationships.

A relationship set may be of the following two types-



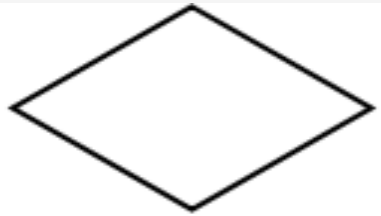
ER Diagram Symbols-

. Strong Relationship Set-

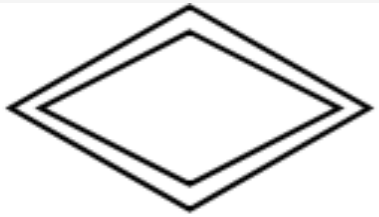
•A strong relationship exists between two strong entity sets.
•It is represented using a diamond symbol.

2. Weak Relationship Set-

•A weak or identifying relationship exists between the strong and weak entity set.
•It is represented using a double diamond symbol.



Strong Relationship Set

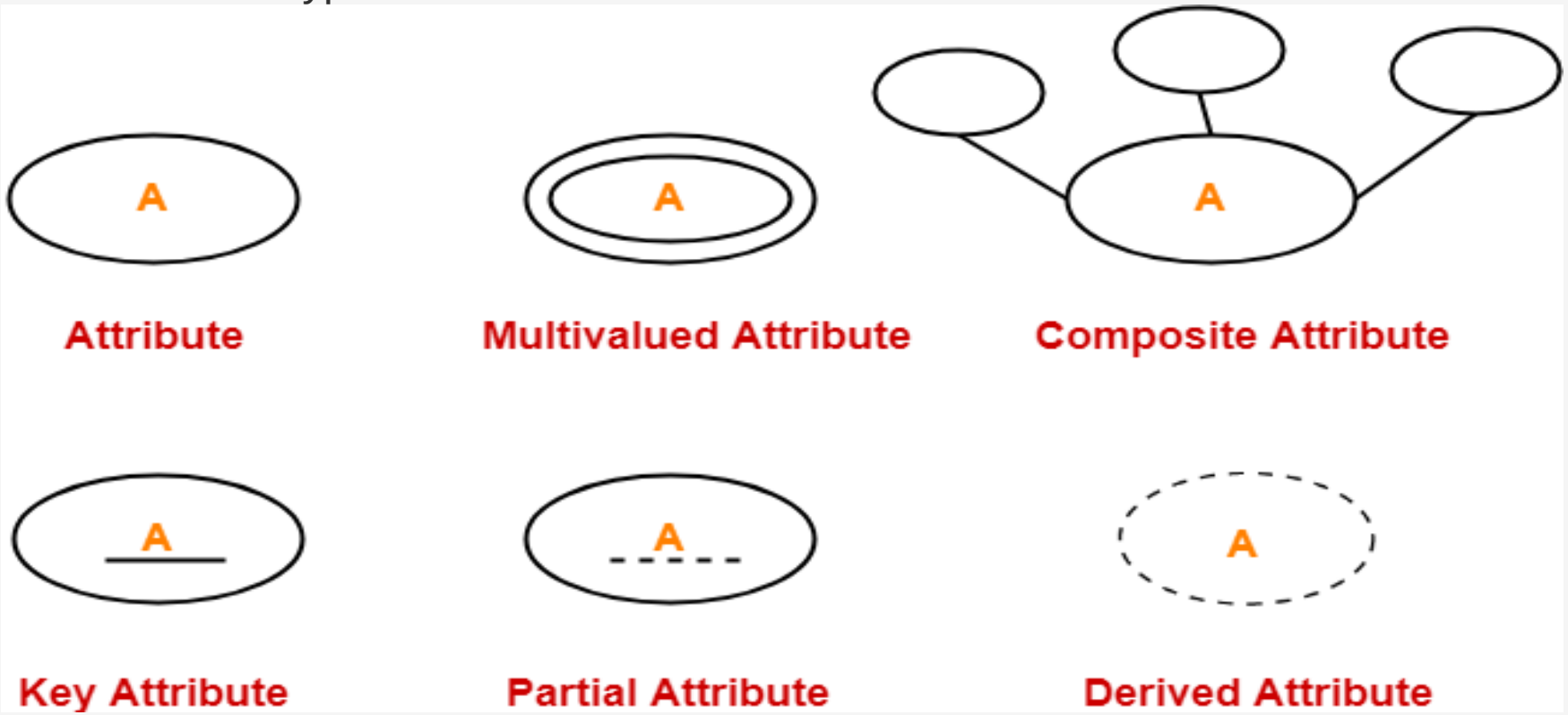


Weak or Identifying Relationship Set

ER Diagram Symbols-

3. For Attributes-

•Attributes are the properties which describes the entities of an entity set.
•There are several types of attributes.

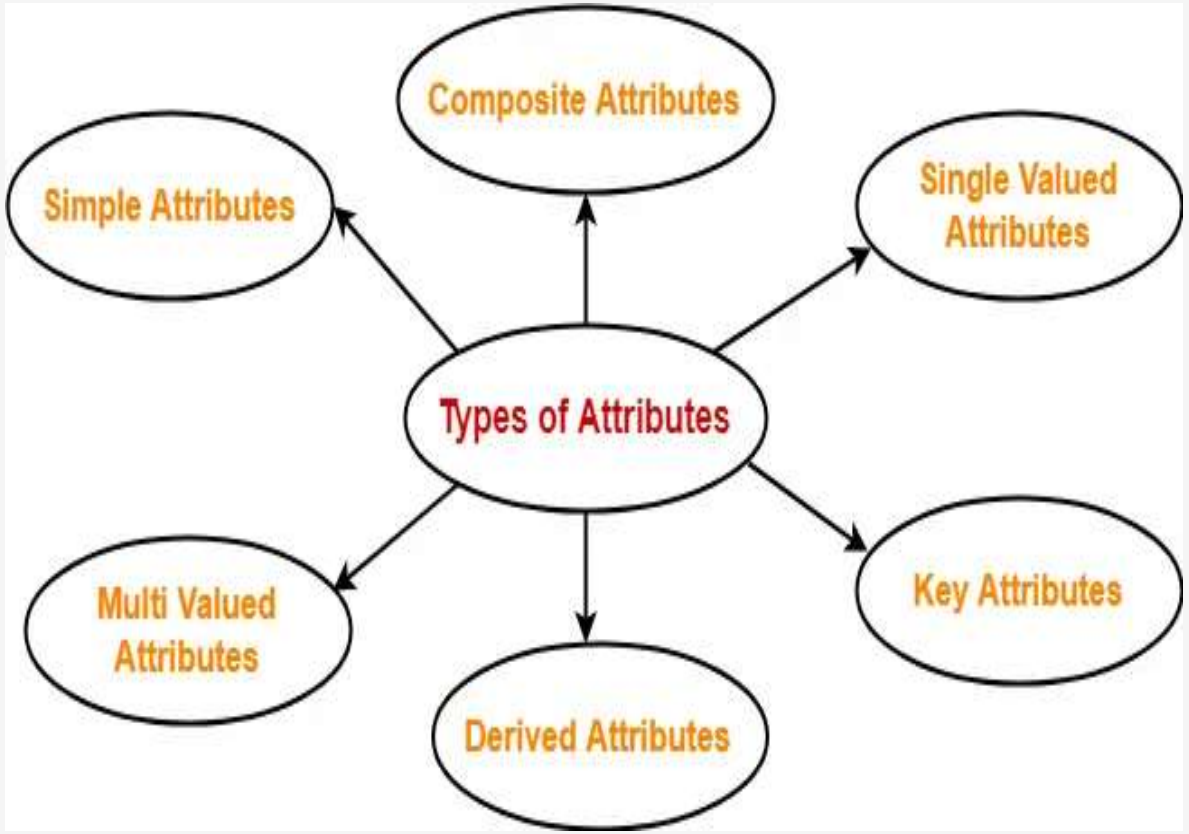


ER Diagram Symbols-

3. For Attributes-

•Attributes are the properties which describes the entities of an entity set.
•There are several types of attributes.

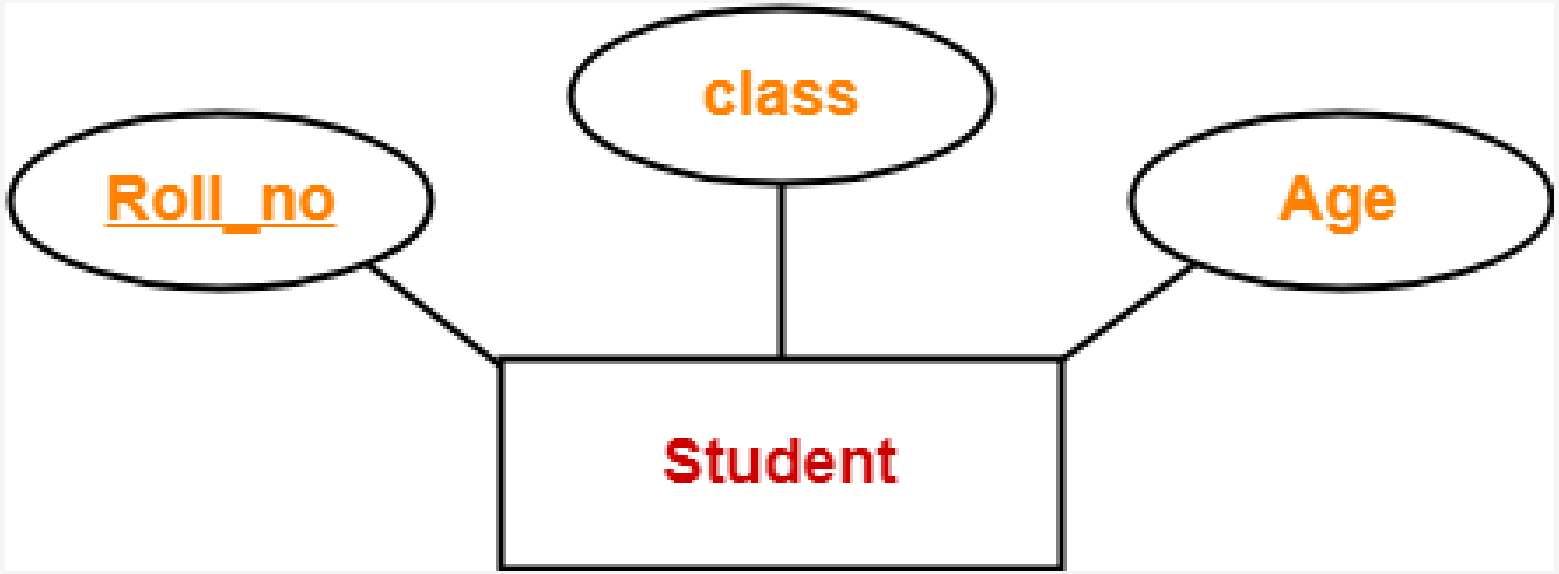
1. Simple attributes
2. Composite attributes
3. Single valued attributes
4. Multi valued attributes
5. Derived attributes
6. Key attributes



ER Diagram Symbols-

1. Simple Attributes-

Simple attributes are those attributes which can not be divided further.



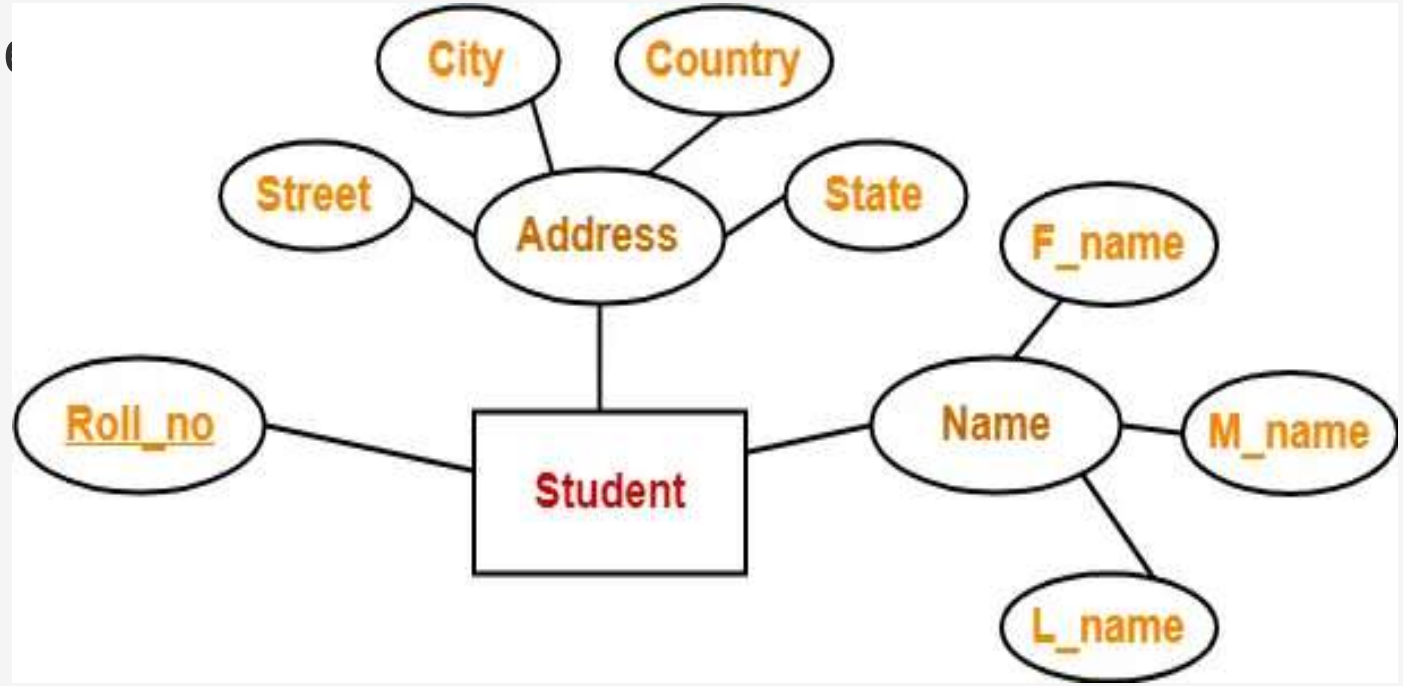
Here, all the attributes are simple attributes as they can not be divided further.

ER Diagram Symbols-

2. Composite Attributes-

Composite attributes are those attributes which are composed of many other simple

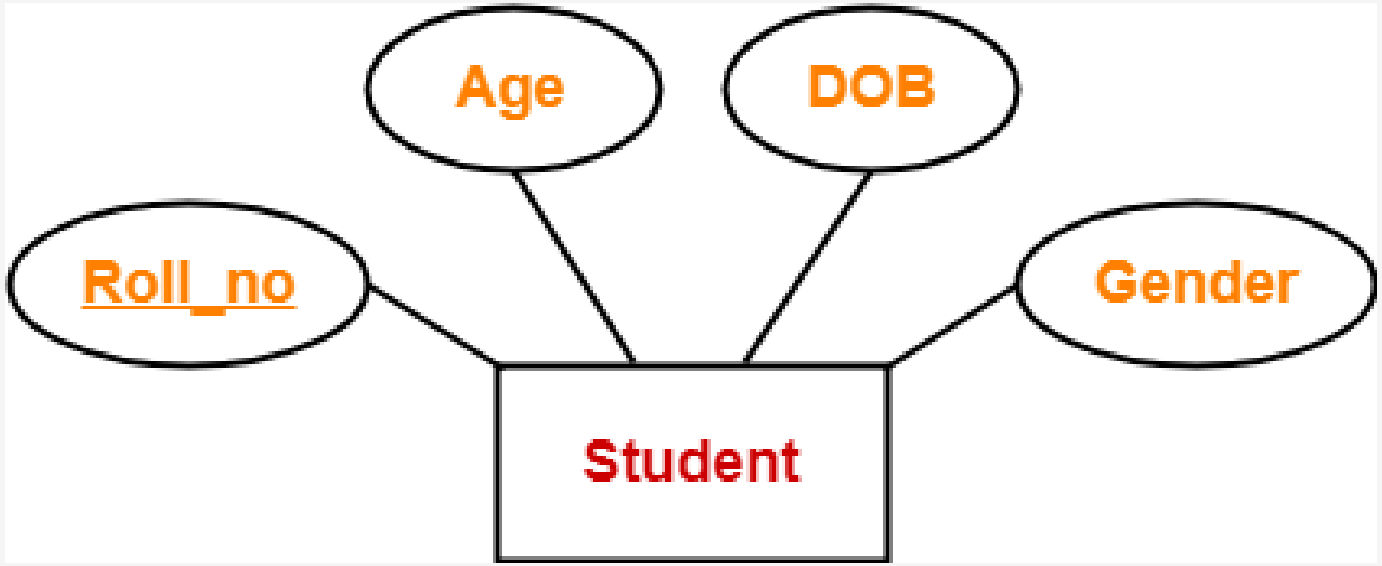
Here, the attributes “Name” and “Address” are composite attributes as they are composed of many other simple attributes.



ER Diagram Symbols-

3. Single Valued Attributes-

Single valued attributes are those attributes which can take only one value for a given entity from an entity set.

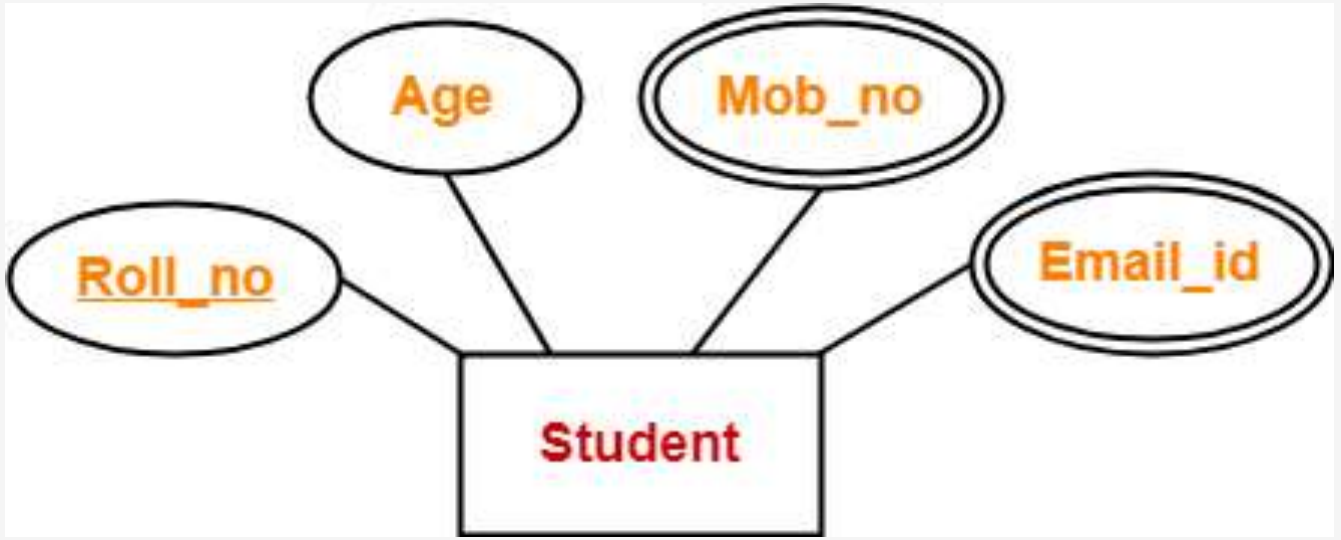


Here, all the attributes are single valued attributes as they can take only one specific value for each entity.

ER Diagram Symbols-

4. Multi Valued Attributes-

Multi valued attributes are those attributes which can take more than one value for a given entity from an entity set.

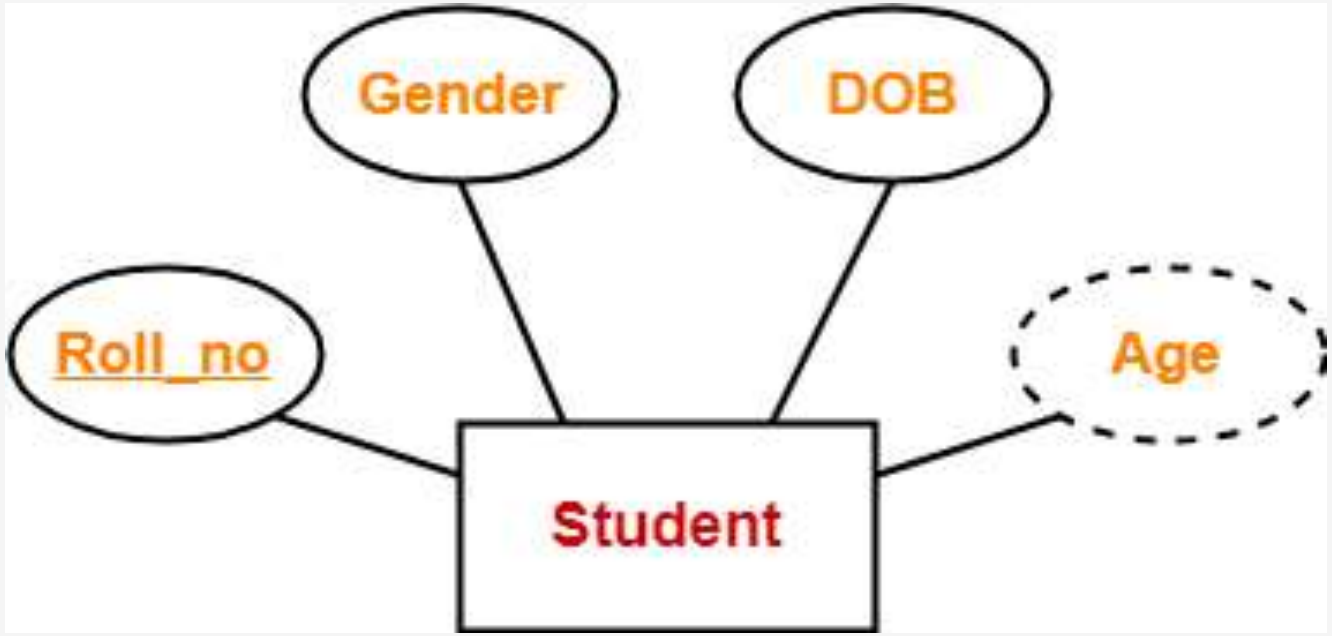


Here, the attributes “Mob_no” and “Email_id” are multi valued attributes as they can take more than one values for a given entity.

ER Diagram Symbols-

5. Derived Attributes-

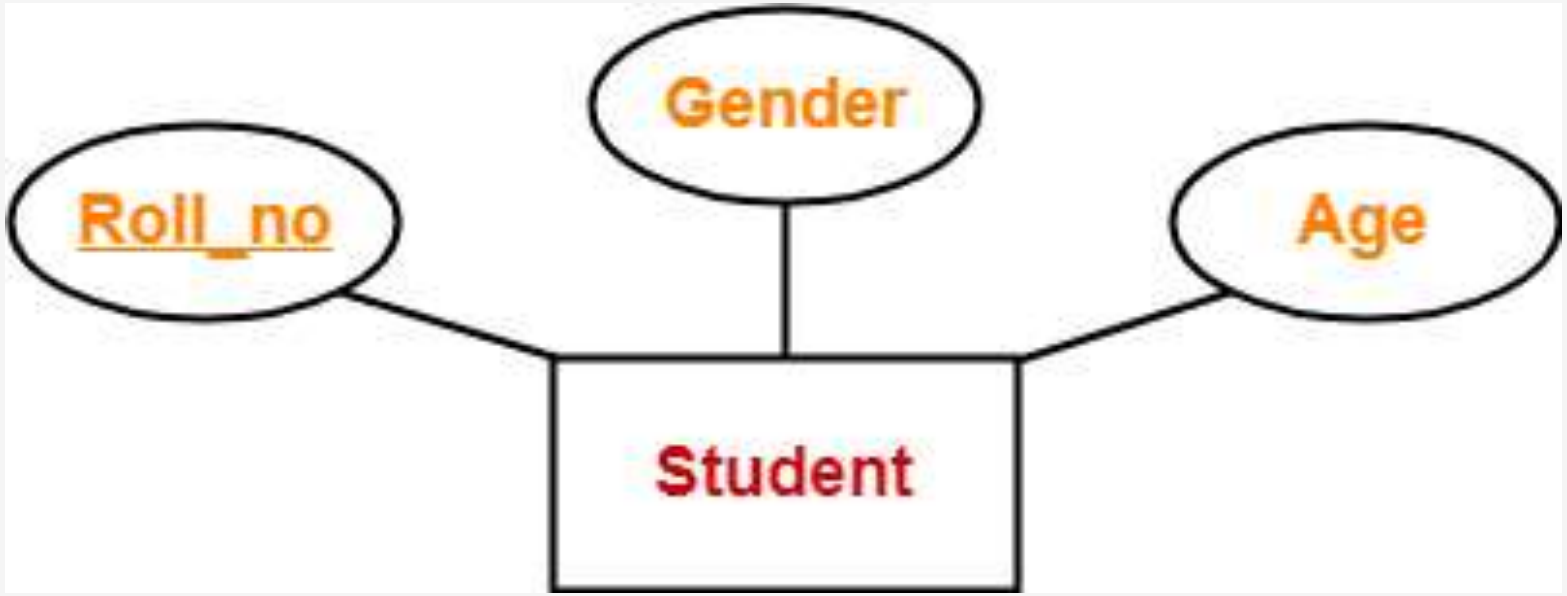
Derived attributes are those attributes which can be derived from other attribute(s).



ER Diagram Symbols-

6. Key Attributes-

Key attributes are those attributes which can identify an entity uniquely in an entity set.

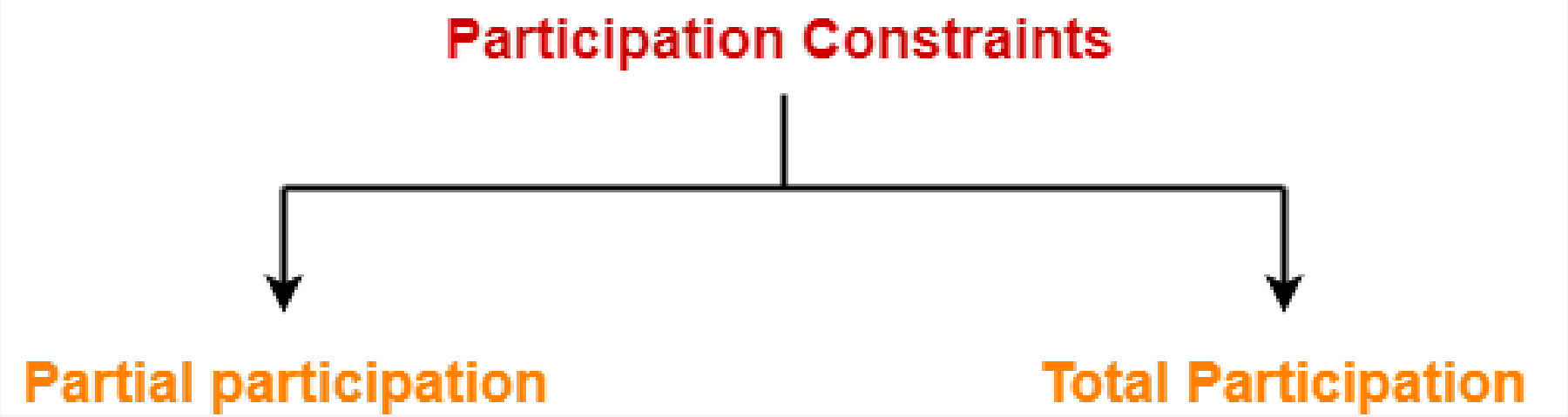


Here, the attribute "Roll_no" is a key attribute as it can identify any student uniquely.

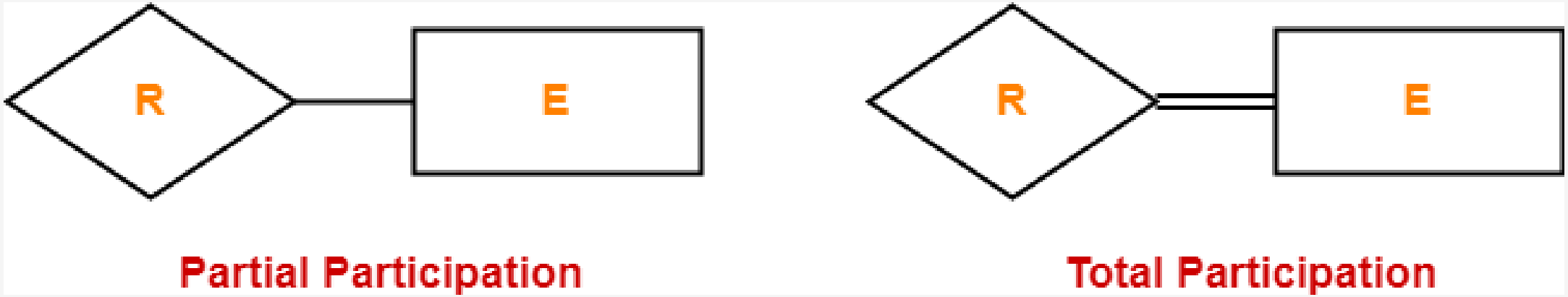
ER Diagram Symbols-

4. For Participation Constraints-

Participation constraint defines the least number of relationship instances in which an entity has to necessarily participate.



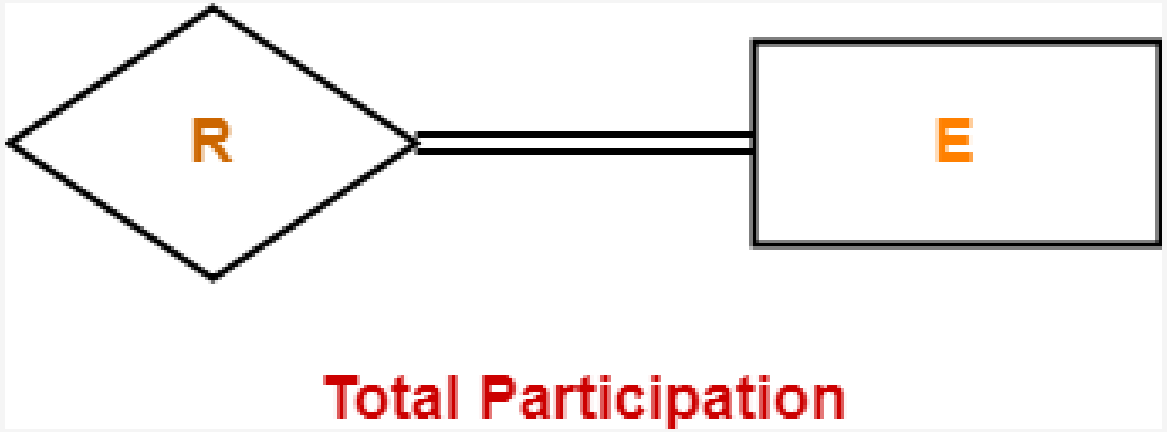
ER Diagram Symbols-



ER Diagram Symbols-

1. Total Participation-

•It specifies that each entity in the entity set must compulsorily participate in at least one relationship instance in that relationship set.
•That is why, it is also called as **mandatory participation.**
•Total participation is represented using a double line between the entity set and relationship set.



ER Diagram Symbols-

Here,

•Double line between the entity set “Student” and relationship set “Enrolled in” signifies total participation.

•It specifies that each student must be enrolled in at least one course.

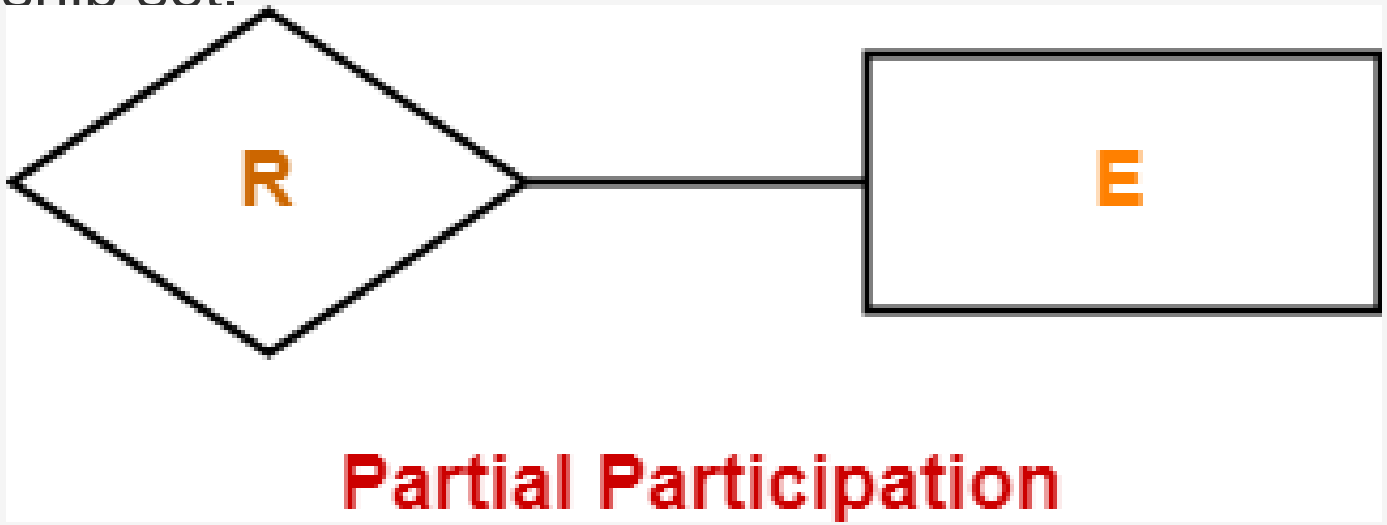
ER Diagram Symbols-

2. Partial Participation-

•It specifies that each entity in the entity set may or may not participate in the relationship instance in that relationship set.

•That is why, it is also called as **optional participation.**

•Partial participation is represented using a single line between the entity set and relationship set.



ER Diagram Symbols-

Here,

•Single line between the entity set “Course” and relationship set “Enrolled in” signifies partial participation.

•It specifies that there might exist some courses for which no enrollments are made.

ER Diagram Symbols-

Relationship between Cardinality and Participation Constraints-

Minimum cardinality tells whether the participation is partial or total.

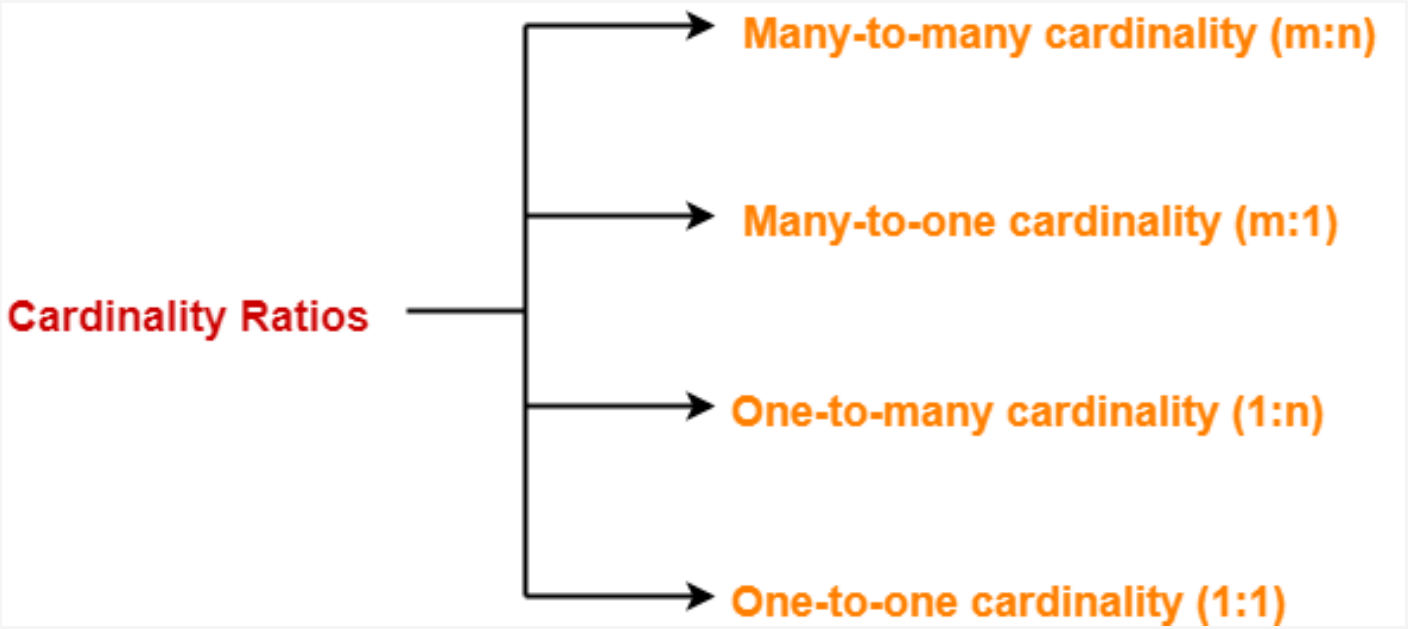
- If minimum cardinality = 0, then it signifies partial participation.
- If minimum cardinality = 1, then it signifies total participation.

Maximum cardinality tells the maximum number of entities that participates in a relationship set.

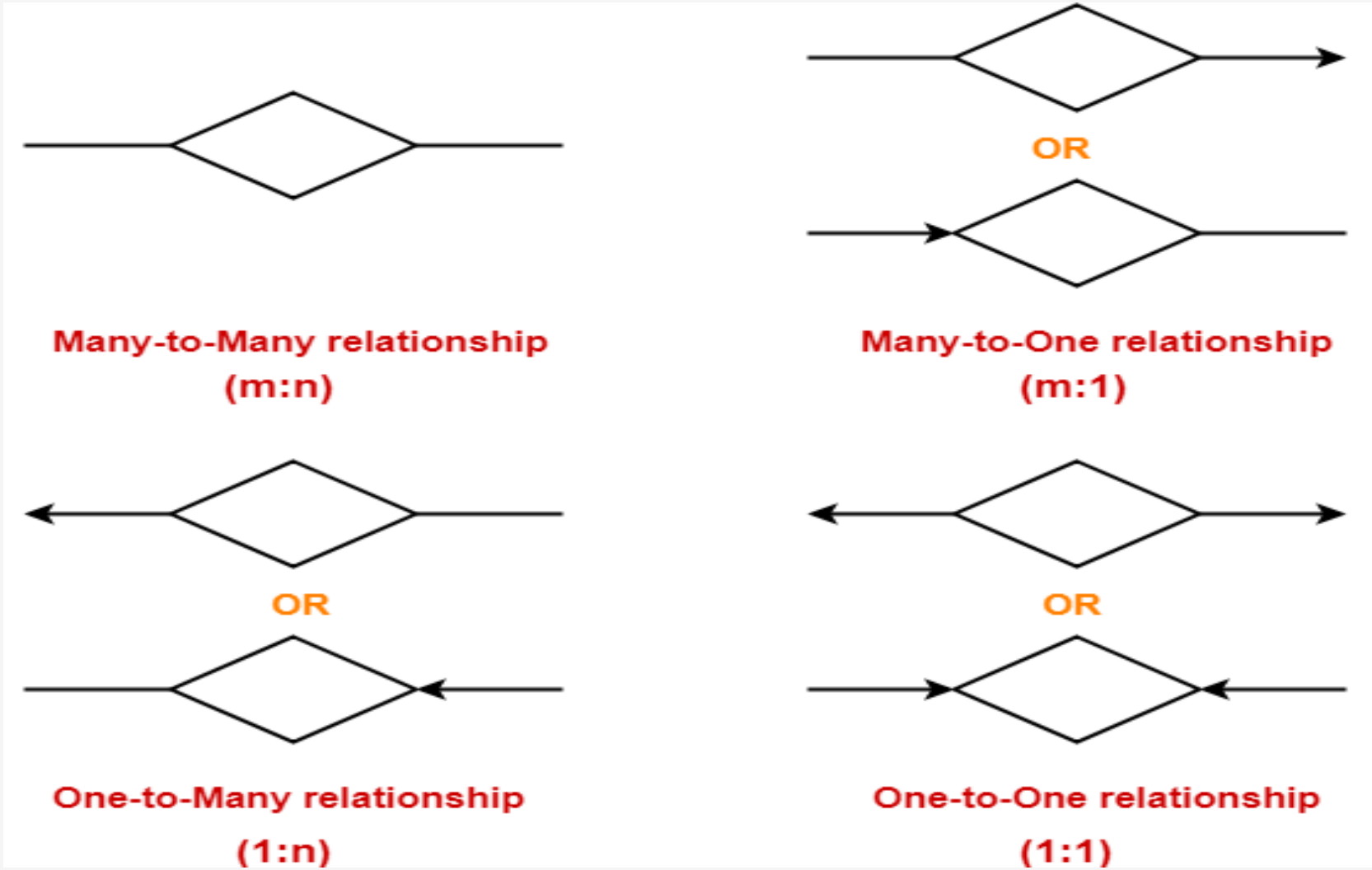
ER Diagram Symbols-

6. For Cardinality Constraints / Ratios-

Cardinality constraint defines the maximum number of relationship instances in which an entity can participate.



ER Diagram Symbols-

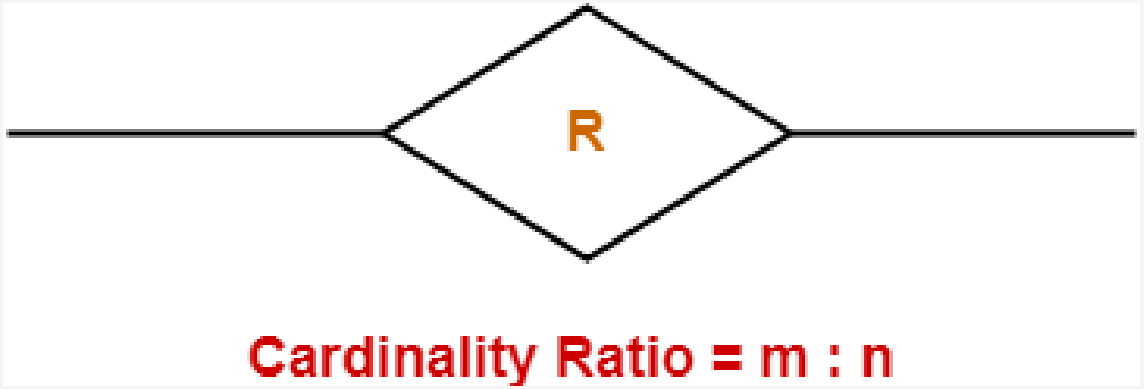


ER Diagram Symbols-

1. Many-to-Many Cardinality-

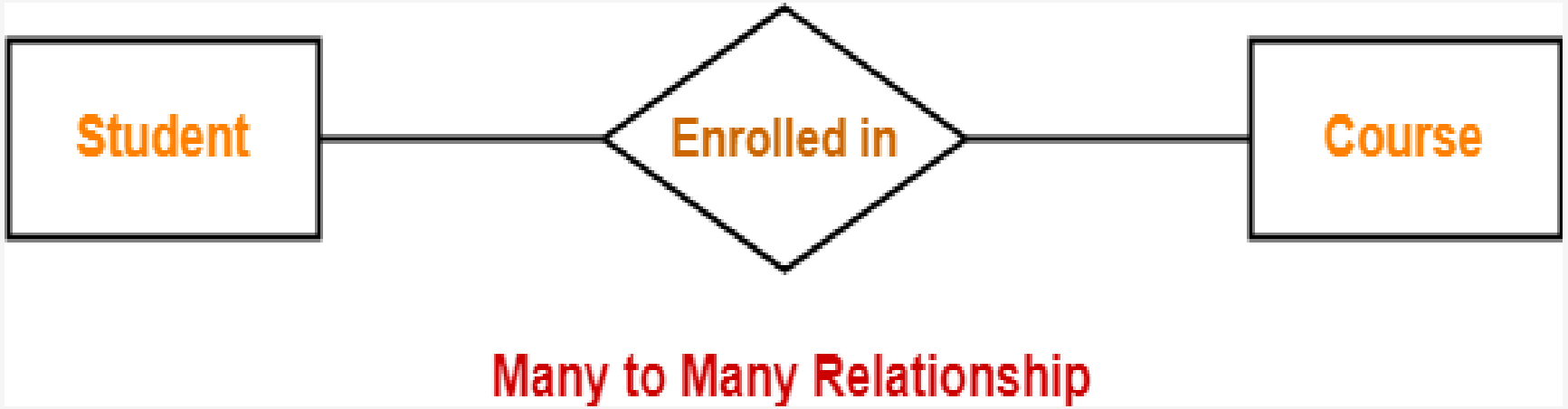
By this cardinality constraint,

- An entity in set A can be associated with any number (zero or more) of entities in set B.
- An entity in set B can be associated with any number (zero or more) of entities in set A.



ER Diagram Symbols-

Example-



Here,

•One student can enroll in any number (zero or more) of courses.

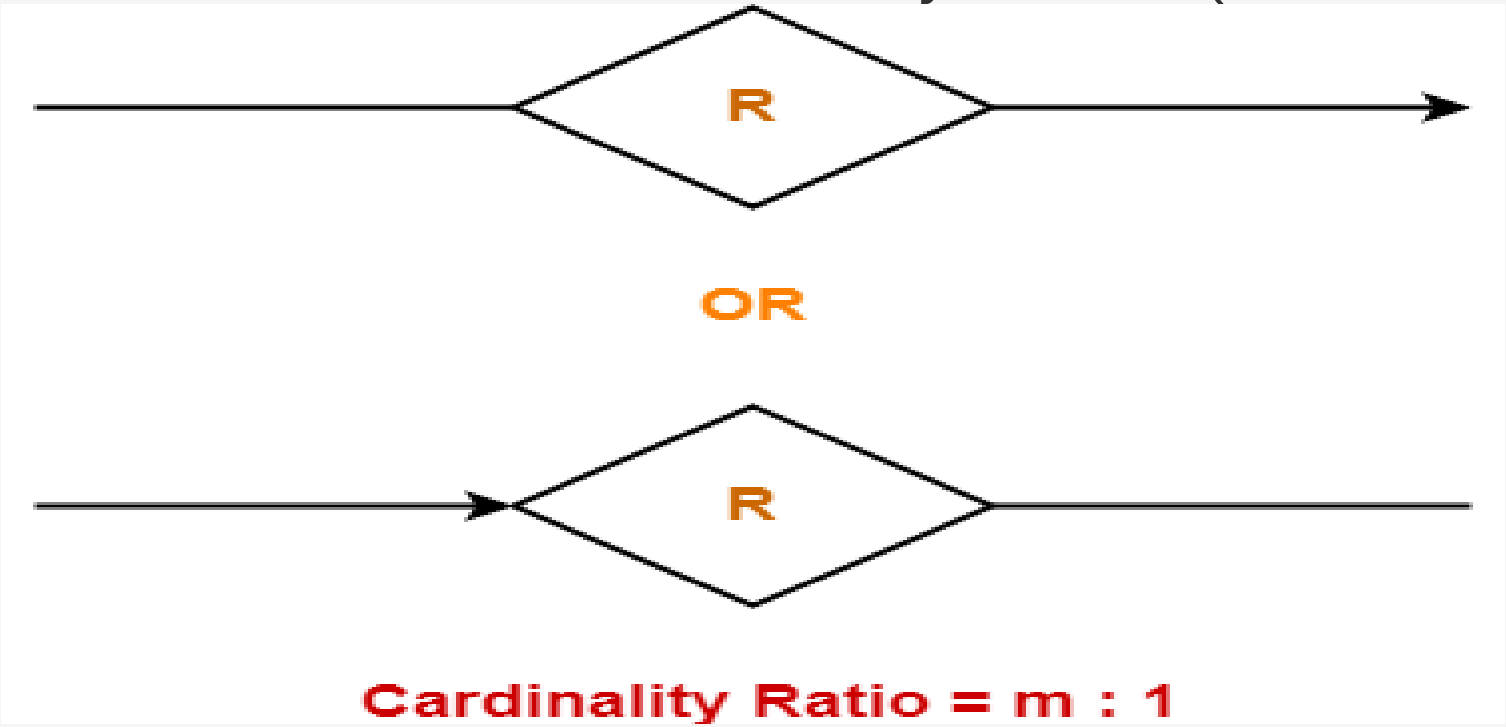
•One course can be enrolled by any number (zero or more) of students.

ER Diagram Symbols-

2. Many-to-One Cardinality-

By this cardinality constraint,

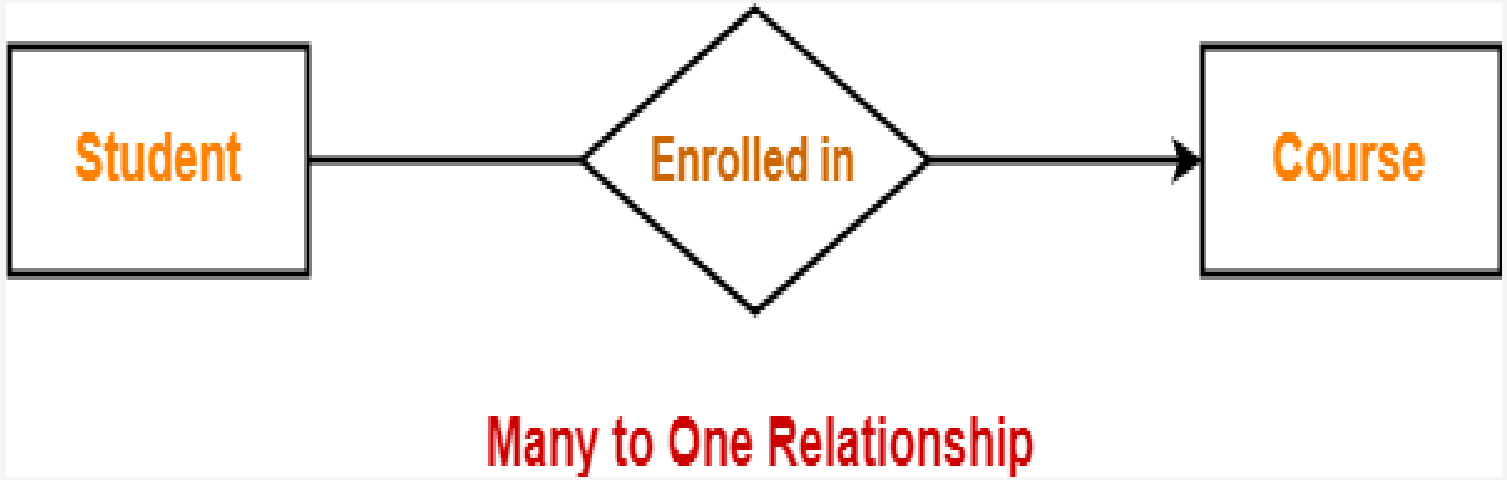
- An entity in set A can be associated with at most one entity in set B.
- An entity in set B can be associated with any number (zero or more) of entities in set A.



Representation as ER Diagram-

Example-

Consider the following ER diagram-



Here,

- One student can enroll in at most one course.
- One course can be enrolled by any number (zero or more) of students.

Representation as ER Diagram-

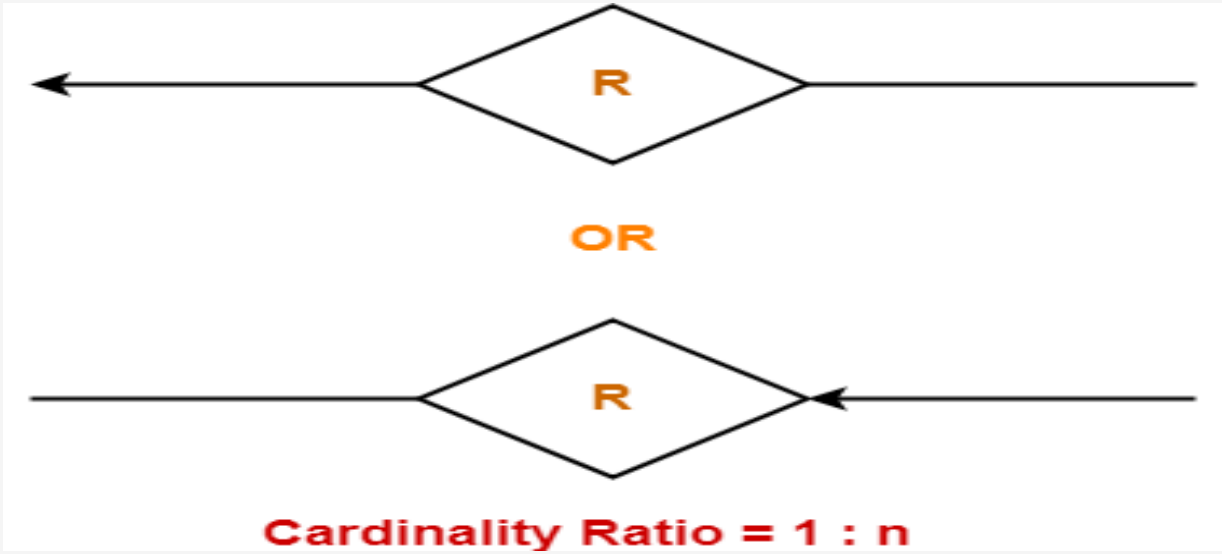
3. One-to-Many Cardinality-

By this cardinality constraint,

•An entity in set A can be associated with any number (zero or more) of entities in set B.

•An entity in set B can be associated with at most one entity in set A.

Symbol Used-



Representation as ER Diagram-

Example-

Consider the following ER diagram-
Here,

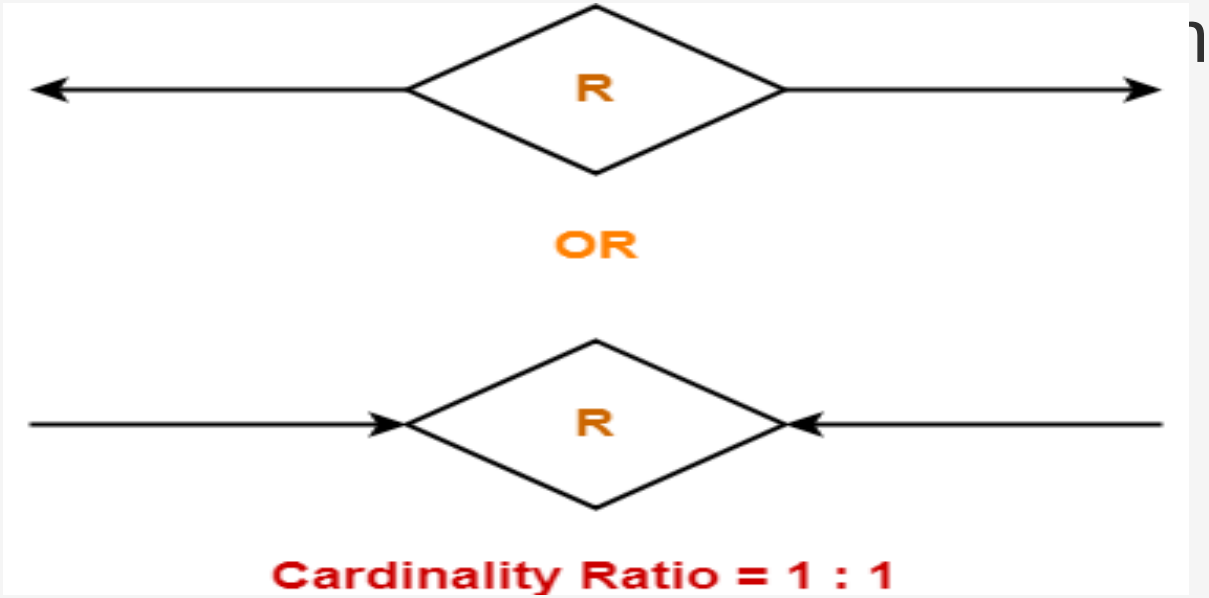
- One student can enroll in any number (zero or more) of courses.
- One course can be enrolled by at most one student.

Representation as ER Diagram-

4. One-to-One Cardinality-

By this cardinality constraint,

- An entity in set A can be associated with at most one entity in set B.
- An entity in set ne entity in set A.
- **Symbol Used-**



ER Diagram Symbols-

Example-

Consider the following ER diagram-

Here,

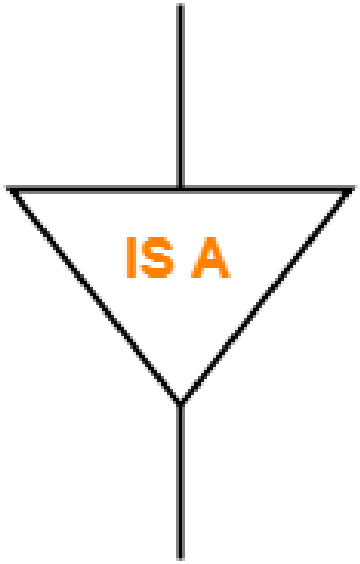
- One student can enroll in at most one course.
- One course can be enrolled by at most one student.

ER Diagram Symbols-

ER Diagram Symbols-

5. For Specialization and Generalization-

- Generalization is a process of forming a generalized super class by extracting the common characteristics from two or more classes.
- Specialization is a reverse process of generalization where a super class is divided into sub classes by assigning the specific characteristics of sub classes to them.



IS A specialization or generalization

Representation as ER Diagram-

Entity-Relationship Model

- Design Process
- Modeling
- Constraints
- E-R Diagram
- Design Issues
- Weak Entity Sets
- Extended E-R Features
- Design of the Bank Database
- Reduction to Relation Schemas
- Database Design

Design Approaches

- Entity Relationship Model (covered in this chapter)
  - 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:*
- Normalization Theory
- Formalize what designs are bad, and test for them

ER model -- Database Modeling

- The ER data mode was developed to facilitate database design by allowing specification of an enterprise schema that represents the overall logical structure of a database.
- The ER model is very useful in mapping the meanings and interactions of real-world enterprises onto a conceptual schema. Because of this usefulness, many database-design tools draw on concepts from the 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:
- instructor = (ID, name, street, city, salary)
  course= (course_id, title, credits)
- A subset of the attributes form a primary key of the entity set; i.e., uniquely identifiying 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

Relationship Sets

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

  Example:

  44553 (Peltier) — *advisor* — 22222 (Einstein)

  student entity — relationship set — *instructor* entity

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

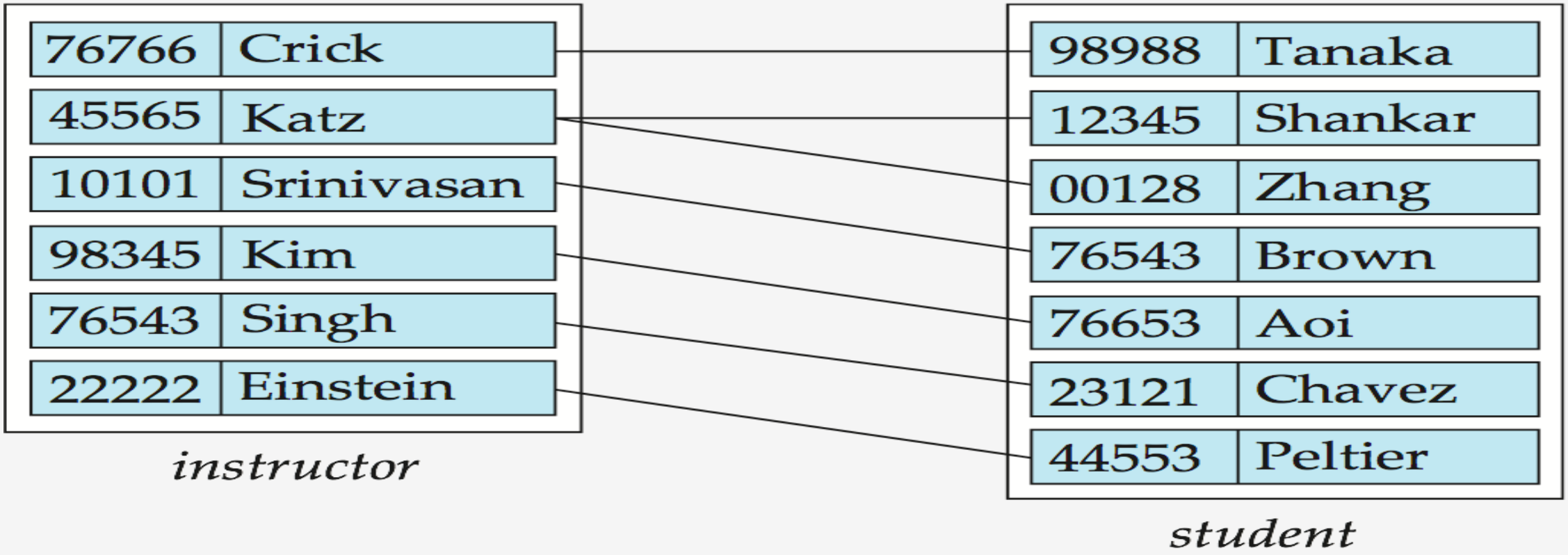
$$\{(e_1, e_2, \ldots e_n) \mid e_1 \in E_1, e_2 \in E_2, \ldots, e_n \in E_n\}$$

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

  - Example:

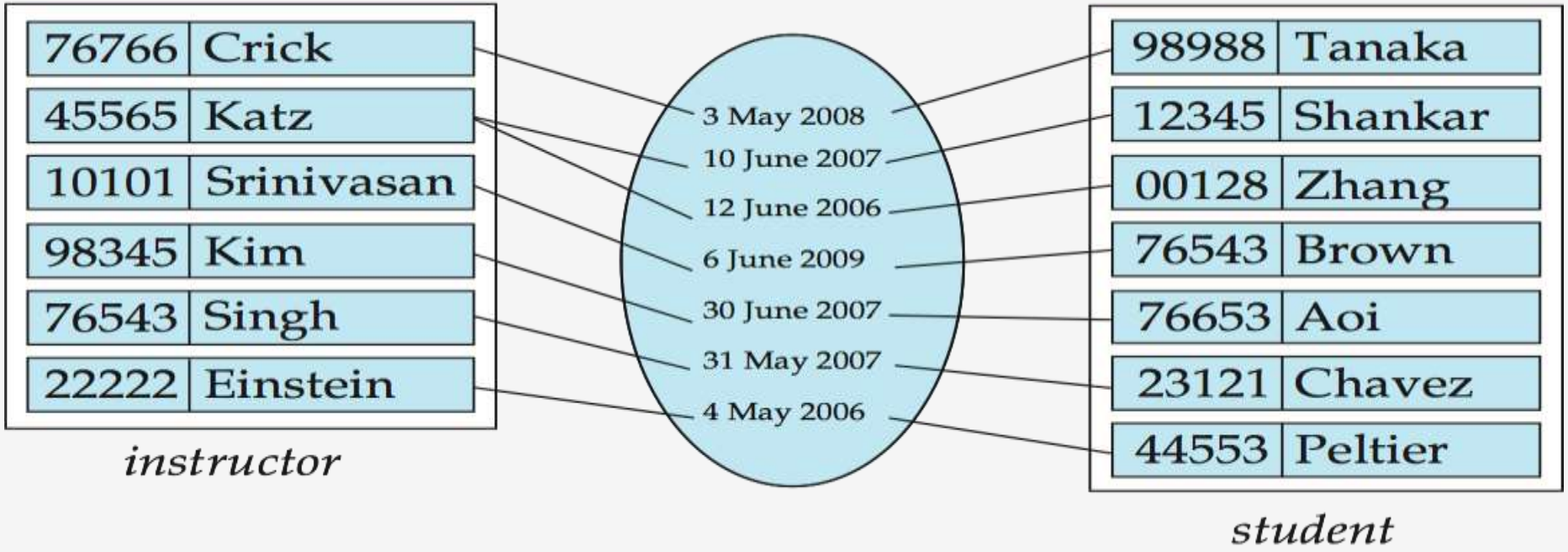
$$(44553,22222) \in advisor$$

Relationship Set advisor



Relationship Sets (Cont.)

- 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



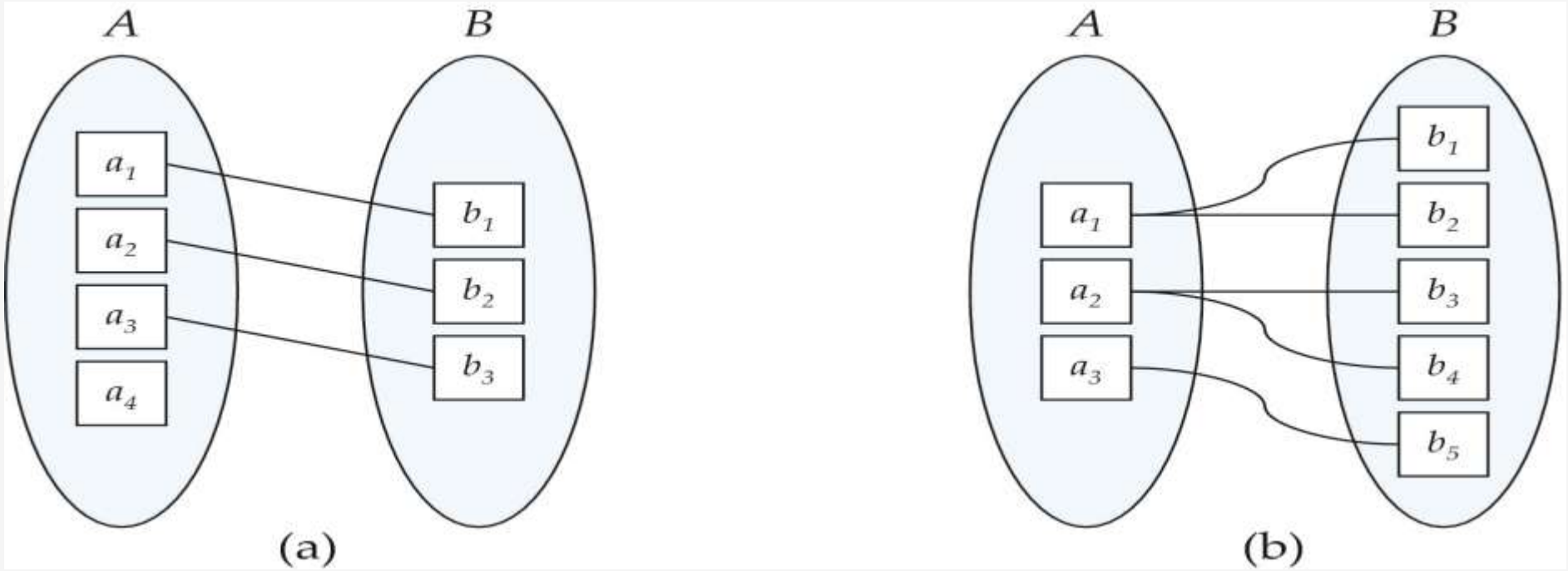
Degree of a Relationship Set

- binary relationship
  - involve two entity sets (or degree two).
  - most relationship sets in a database system are binary.
- Relationships between more than two entity sets are rare. Most relationships are binary. (More on this later.)
  - Example: *students* work on research *projects* under the guidance of an *instructor*.
  - relationship *proj_guide* is a ternary relationship between *instructor*, *student,* and *project*

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

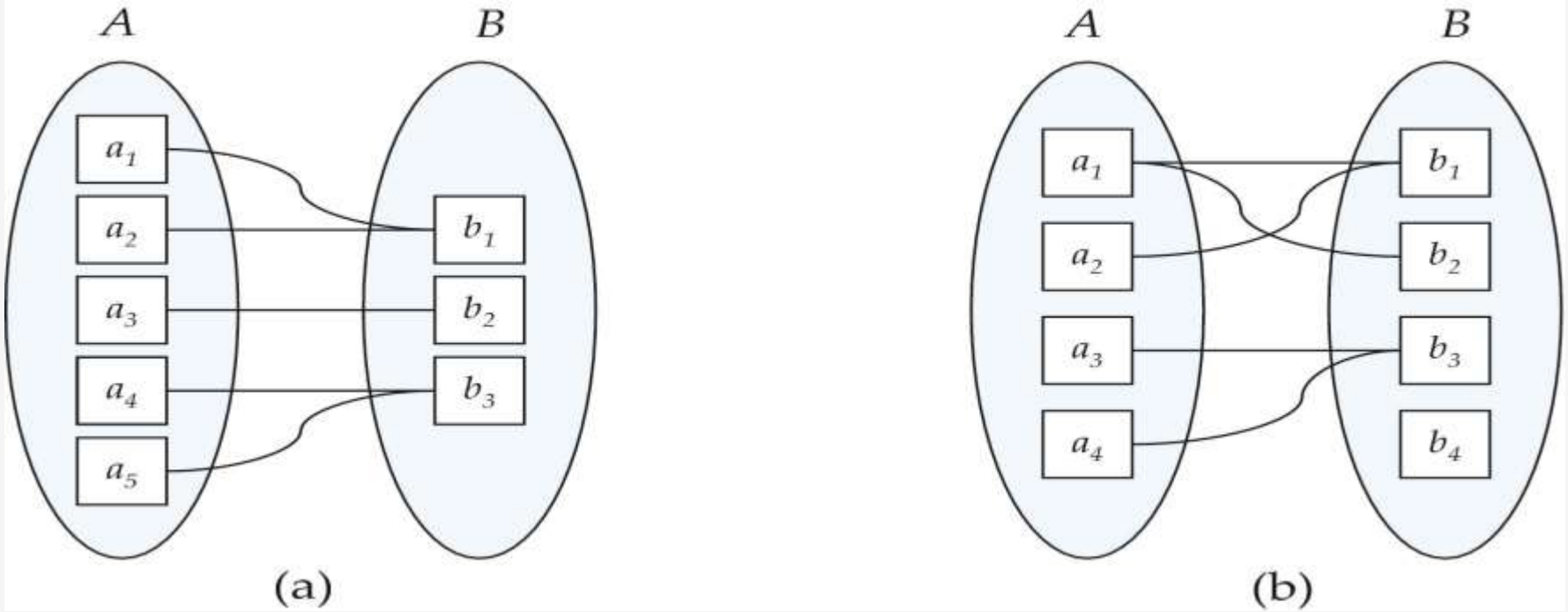


One to one

One to many

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

Mapping Cardinalities



many to one

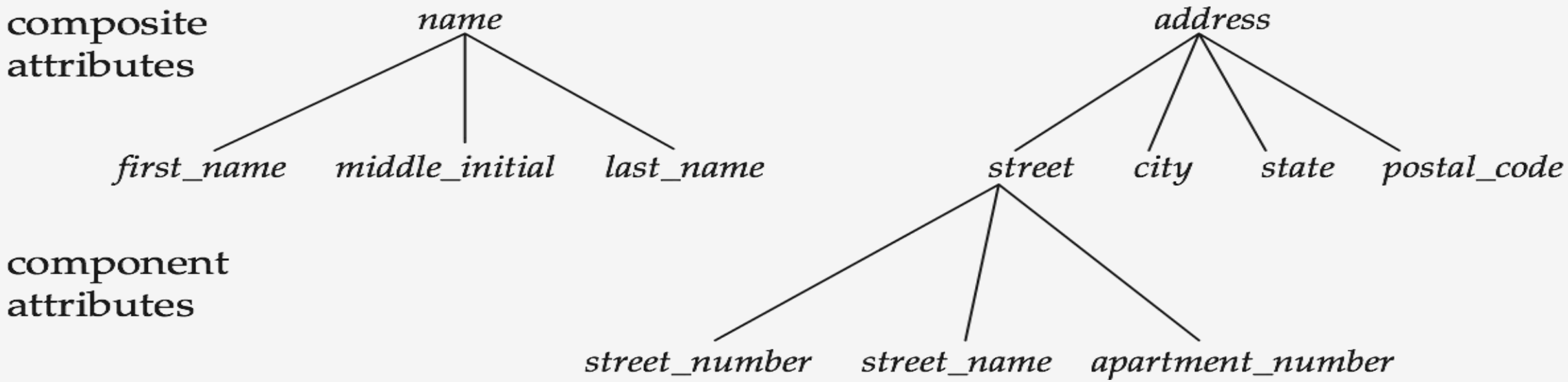
many to many

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

Complex 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



Redundant Attributes

- Suppose we have entity sets:
  - *instructor*, with attributes: *ID*, *name*, *dept_name, salary*
  - *department,* with attributes: *dept_name, building, budget*
- We model the fact that each instructor has an associated department using a relationship set *inst_dept*
- The attribute *dept_name* appears in both entity sets. Since it is the primary key for the entity set *department*, it replicates information present in the relationship and is therefore redundant in the entity set *instructor* and needs to be removed.
- BUT: when converting back to tables, in some cases the attribute gets reintroduced, as we will see later.

Weak Entity Sets

- Consider a *section* entity, which is uniquely identified by a *course_id*, *semester, year*, and *sec_id*.
- Clearly, section entities are related to course entities. Suppose we create a relationship set *sec_course* between entity sets *section* and *course*.
- Note that the information in *sec_course* is redundant, since *section* already has an attribute *course_id*, which identifies the course with which the section is related.
- One option to deal with this redundancy is to get rid of the relationship *sec_course*; however, by doing so the relationship between *section* and *course* becomes implicit in an attribute, which is not desirable

Weak Entity Sets (Cont.)

- An alternative way to deal with this redundancy is to not store the attribute *course_id* in the *section* entity and to only store the remaining attributes *section_id*, *year*, and *semester.* However, the entity set *section* then does not have enough attributes to identify a particular *section* entity uniquely; although each *section* entity is distinct, sections for different courses may share the same *section_id*, *year*, and *semester*.
- To deal with this problem, we treat the relationship *sec_course* as a special relationship that provides extra information, in this case, the *course_id*, required to identify *section* entities uniquely.
- The notion of **weak entity set** formalizes the above intuition. A weak entity set is one whose existence is dependent on another entity, called its **identifying entity**; instead of associating a primary key with a weak entity, 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 Sets (Cont.)

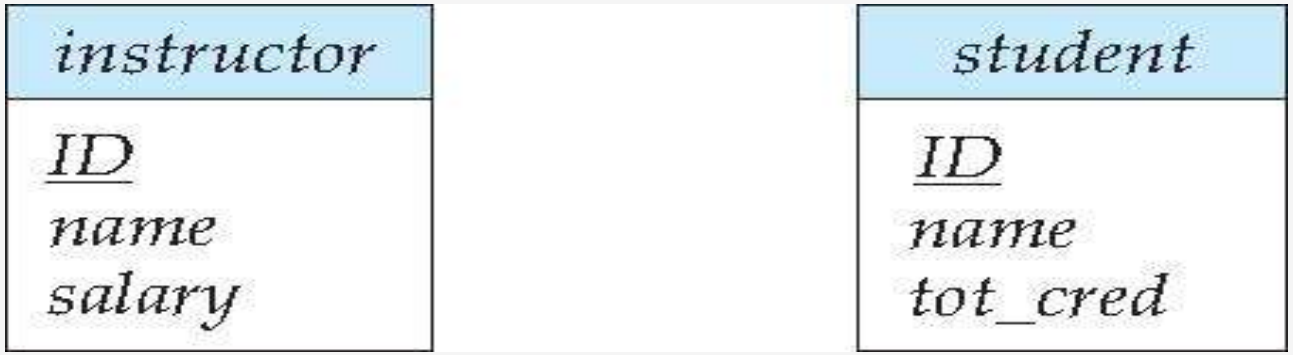
- Every weak entity must be associated with an identifying entity; that is, the weak entity set is said to be **existence dependent** on the identifying entity set. The identifying entity set is said to **own** the weak entity set that it identifies. The relationship associating the weak entity set with the identifying entity set is called the **identifying relationship**.
- Note that the relational schema we eventually create from the entity set *section* does have the attribute *course_id*, for reasons that will become clear later, even though we have dropped the attribute *course_id* from the entity set *section.*

E-R Diagrams

Entity Sets

Entities can be represented graphically as follows:

- **Rectangles represent entity sets.**
- **Attributes listed inside entity rectangle**
- **Underline indicates primary key attributes**



Relationship Sets

- ✓ Diamonds represent relationship sets.