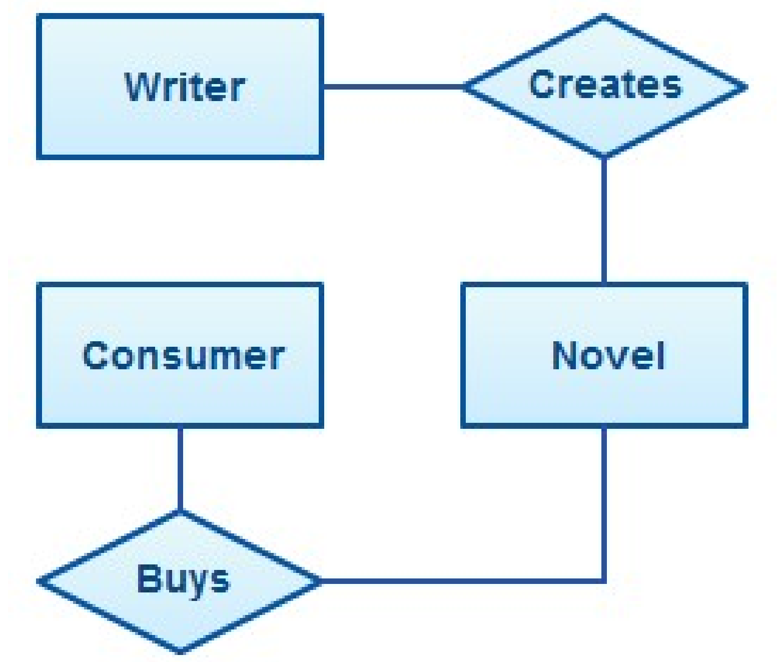
Entity Relationship Model

Entity Relationship Model

- An Entity Relationship Diagram (ERD) is a ***visual representation of different data using conventions*** that ***describe how these data are related to each other***.
- For example, the elements writer, novel, and consumer may be described using ER diagrams this way:



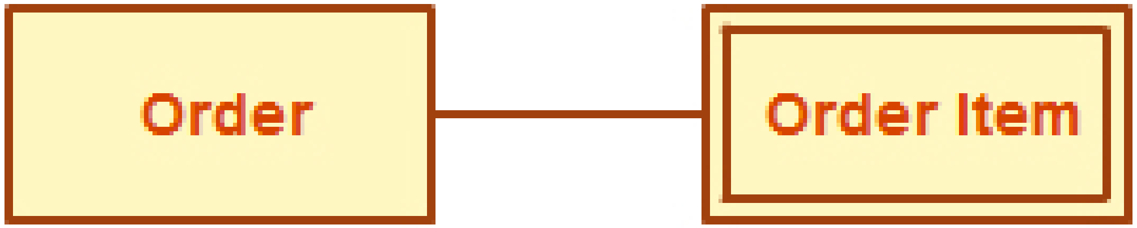
The elements inside **rectangles** are called **entities** while the items inside **diamonds** denote the **relationships** between entities.

Entity

- ***An entity can be a real-world object***, either animate or inanimate, that can be ***easily identifiable***.
  - For example, in a school database, **students, teachers, classes,** and **courses offered** can be considered as **entities**.
  - All these entities have some ***attributes or properties*** that give them their identity.
- An **entity set** is a collection of similar types of entities.
  - An entity set may contain ***entities with attribute sharing similar values.***
  - For example,
  - a Students set may contain all the students of a school;
  - likewise a Teachers set may contain all the teachers of a school from all faculties. Entity sets need not be disjoint.

Weak Entity

- **A weak entity is an entity that depends on the existence of another entity.**
- In more technical terms it can defined as an entity that cannot be identified by its own attributes.
- It uses a foreign key combined with its attributed to form the primary key.
- An entity like **order item** is a good example for this.
  - The order item will be meaningless without an order so it depends on the existence of order.



Attributes

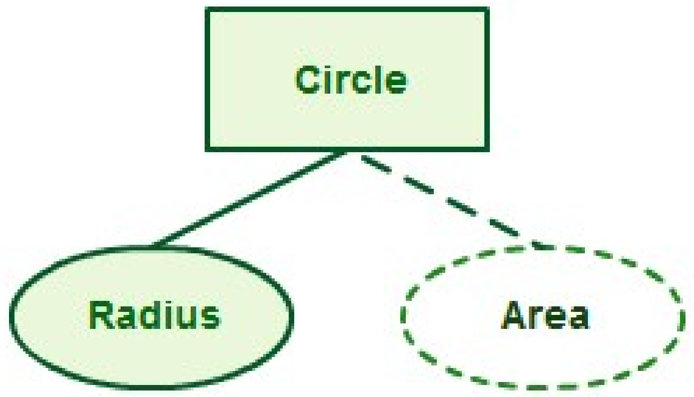
- Entities are represented by means of their ***properties***, called **attributes**.
- All attributes have values.
- For example, a **student entity** may have **name, class,** and **age** as attributes.
- ***There exists a domain or range of values that can be assigned to attributes.***
- For example, a **student's name** cannot be a numeric value. It has to be ***alphabetic***. A **student's age cannot be negative**, etc.

Types of Attributes

- **Simple attribute** − Simple attributes are atomic values, which cannot be divided further. For example, **a student's phone number is an atomic value of 10 digits**.
- **Composite attribute** − Composite attributes are made of more than one simple attribute. For example, **a student's complete name may have first_name and last_name.**

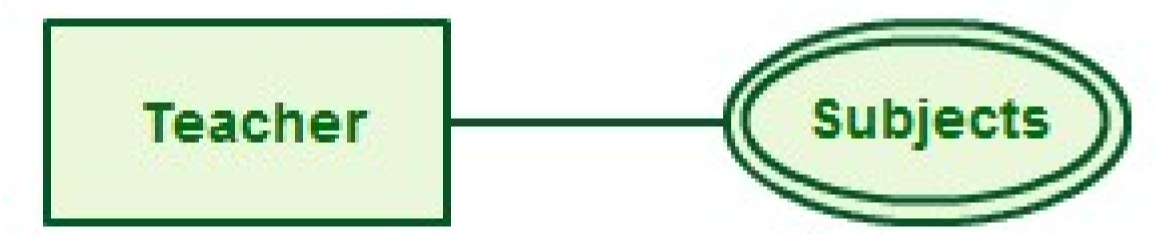
Types of Attributes (contd.,)

- **Derived attribute** − Derived attributes are the attributes that do not exist in the physical database, but their **values are derived from other attributes** present in the database.
- For example, average_salary in a department should not be saved directly in the database, instead it can be derived.
- For another example, age can be derived from data_of_birth.
- For example for a circle the area can be derived from the radius.



Types of Attributes (contd.,)

- **Single-value attribute** − Single-value attributes **contain single value**. For example − Social_Security_Number.
- **Multi-value attribute** − Multi-value attributes may **contain more than one values**. For example, a person can have more than one phone number, email_address, etc.
- For example a teacher entity can have multiple subject values.



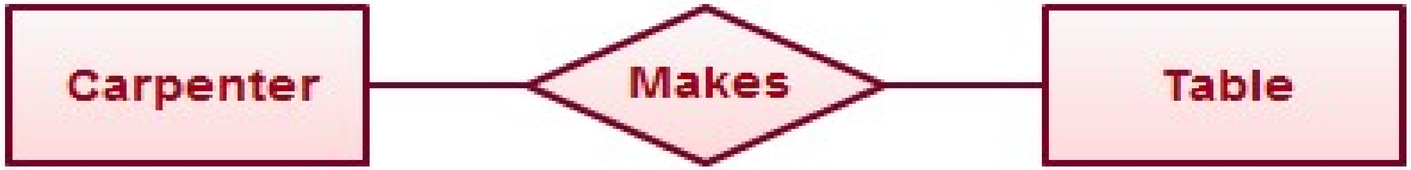
- These attribute types can come together in a way like –
- ***simple single-valued attributes***
- ***simple multi-valued attributes***
- ***composite single-valued attributes***
- ***composite multi-valued attributes***

Entity-Set and Keys

- **Key is an attribute or collection of attributes that uniquely identifies an entity among entity set**.
- For example, the **roll_number** of a student makes him/her identifiable among students.
- **Super Key** − A set of attributes (one or more) that collectively identifies an entity in an entity set.
- **Candidate Key** − A minimal super key is called a candidate key. An entity set may have more than one candidate key.
- **Primary Key** − A primary key is one of the candidate keys chosen by the database designer to uniquely identify the entity set.

Relationship

- The association among entities is called a relationship.
- For example, an employee **works_at** a department, a student **enrolls** in a course.
- Here, Works_at and Enrolls are called relationships.
- For example, the entity "**carpenter**" may be related to the **entity** "**table**" by the **relationship "builds" or "makes**". Relationships are represented by diamond shapes and are labeled using verbs.

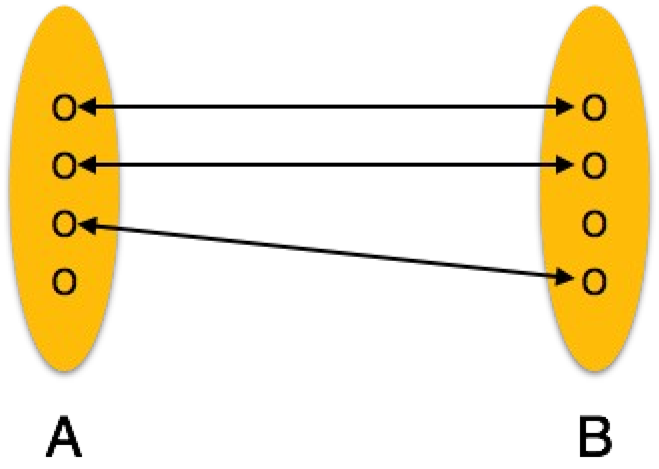


Degree of Relationship

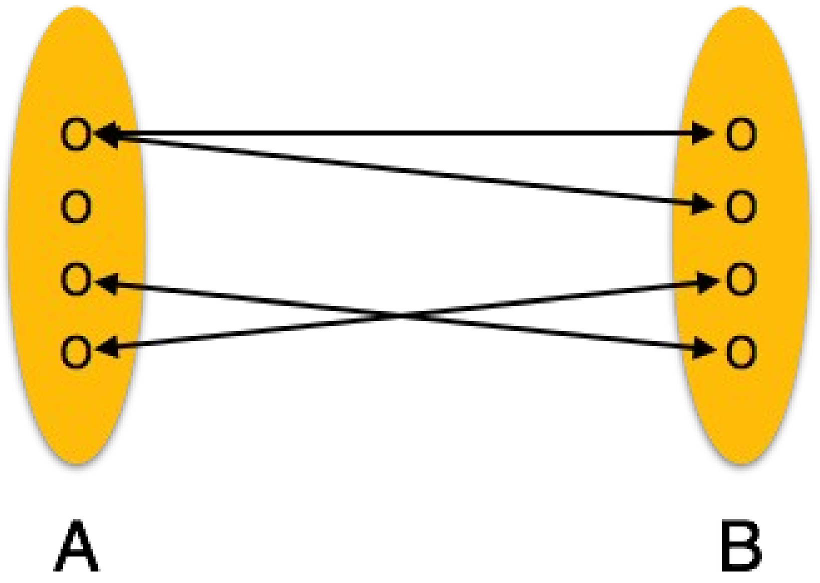
- The number of participating entities in a relationship defines the degree of the relationship.
- Binary = degree 2
- Ternary = degree 3
- n-ary = degree

Mapping Cardinalities

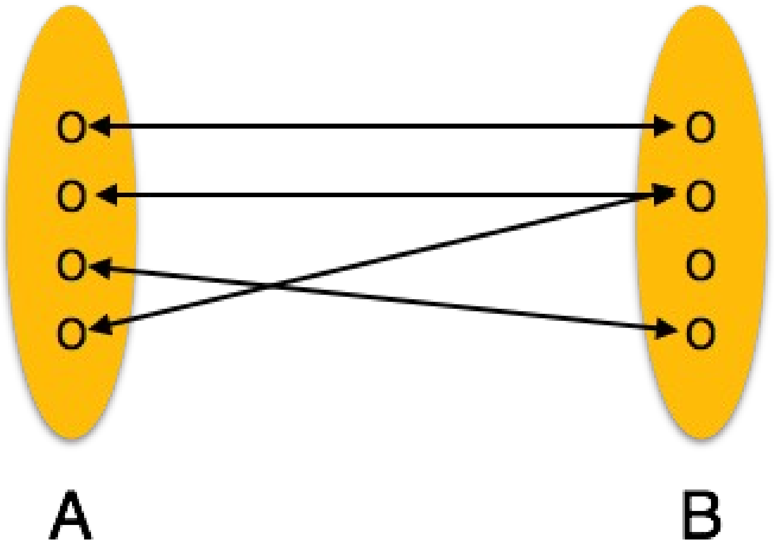
- **Cardinality** defines the number of entities in one entity set, which can be associated with the number of entities of other set via relationship set.
- **One-to-one** − One entity from entity set A can be associated with at most one entity of entity set B and vice versa.



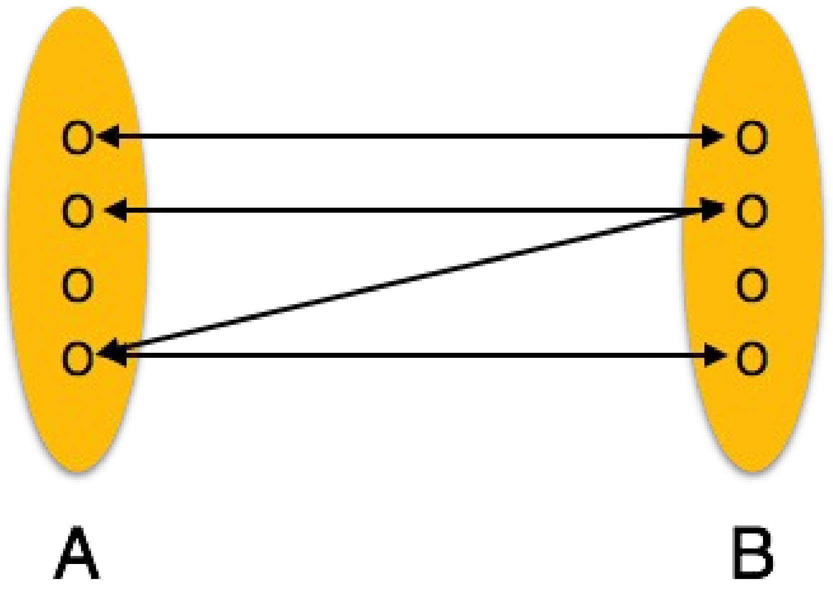
- **One-to-many** − One entity from entity set A can be associated with more than one entities of entity set B however an entity from entity set B, can be associated with at most one entity.



- **Many-to-one** − More than one entities from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A.



- **Many-to-many** − One entity from A can be associated with more than one entity from B and vice versa.

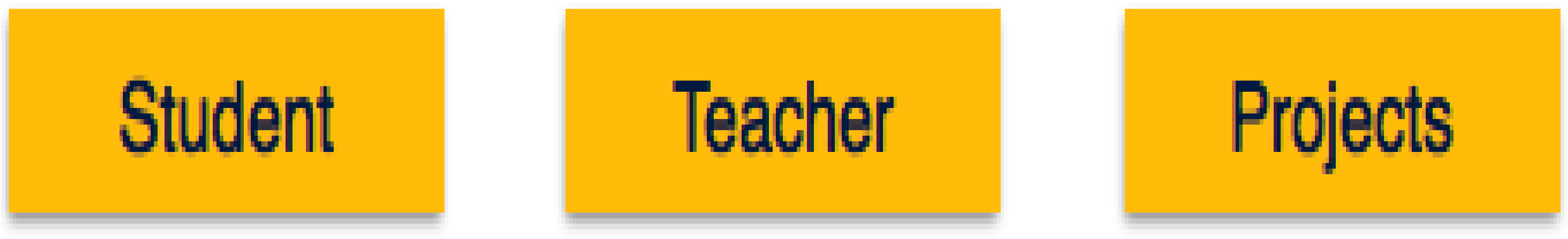


ER Diagram Representation

- Let us now learn how the ER Model is represented by means of an ER diagram.
- Any object, for example, entities, attributes of an entity, relationship sets, and attributes of relationship sets, can be represented with the help of an ER diagram.

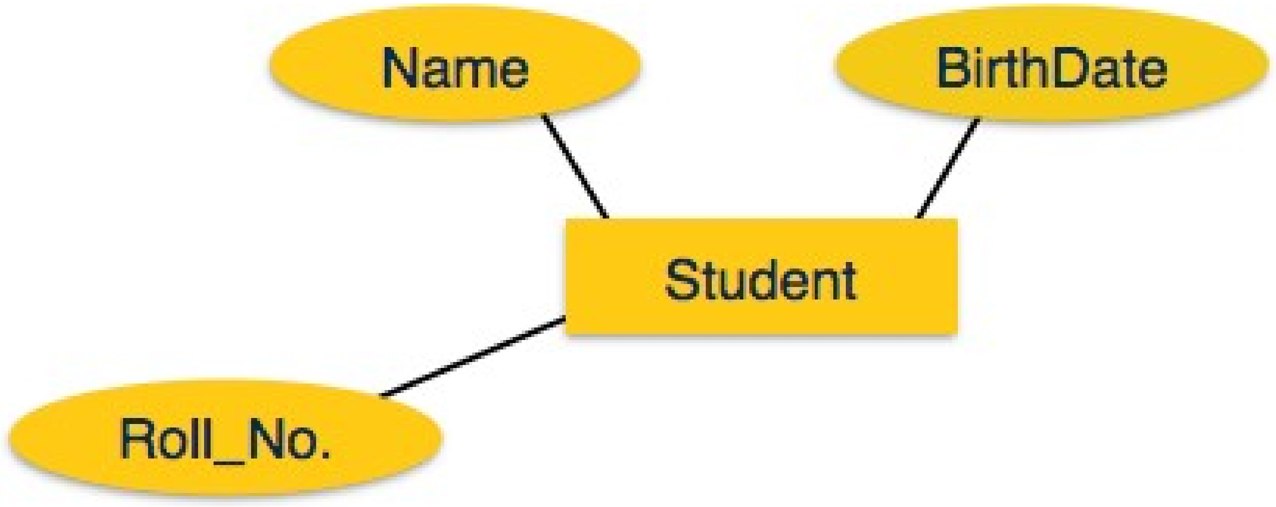
Entity

- Entities are represented by means of rectangles. Rectangles are named with the entity set they represent.

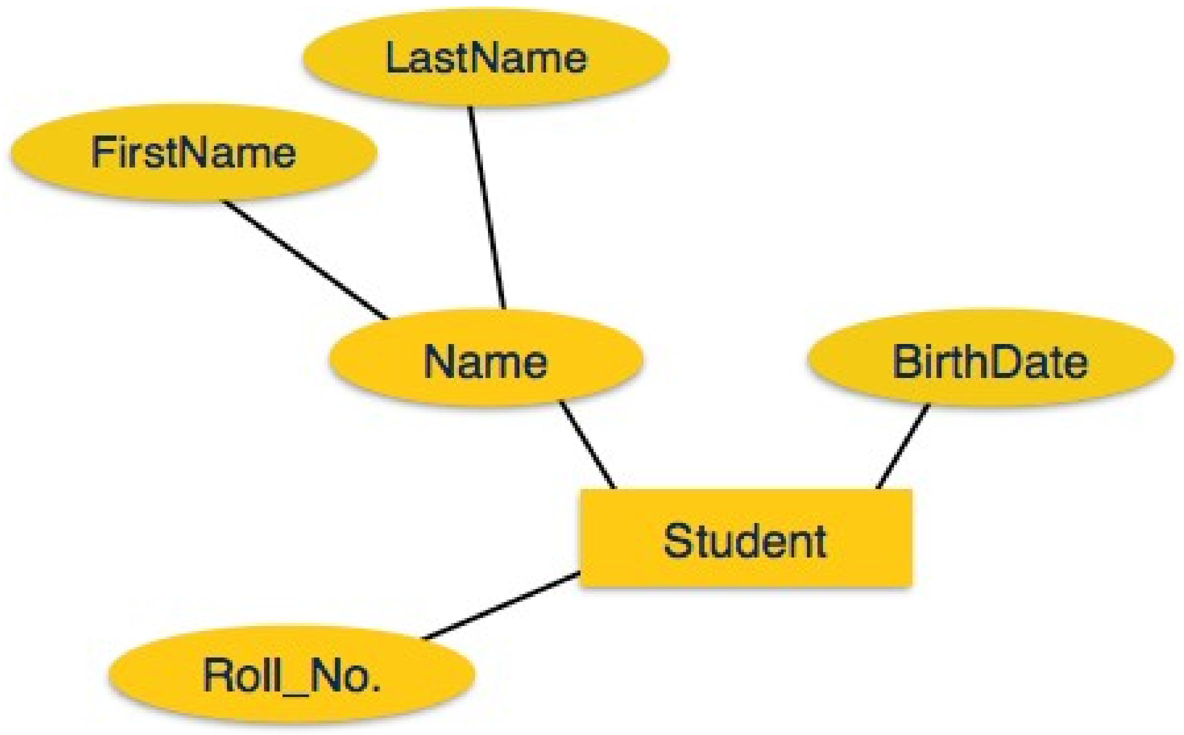


Attributes

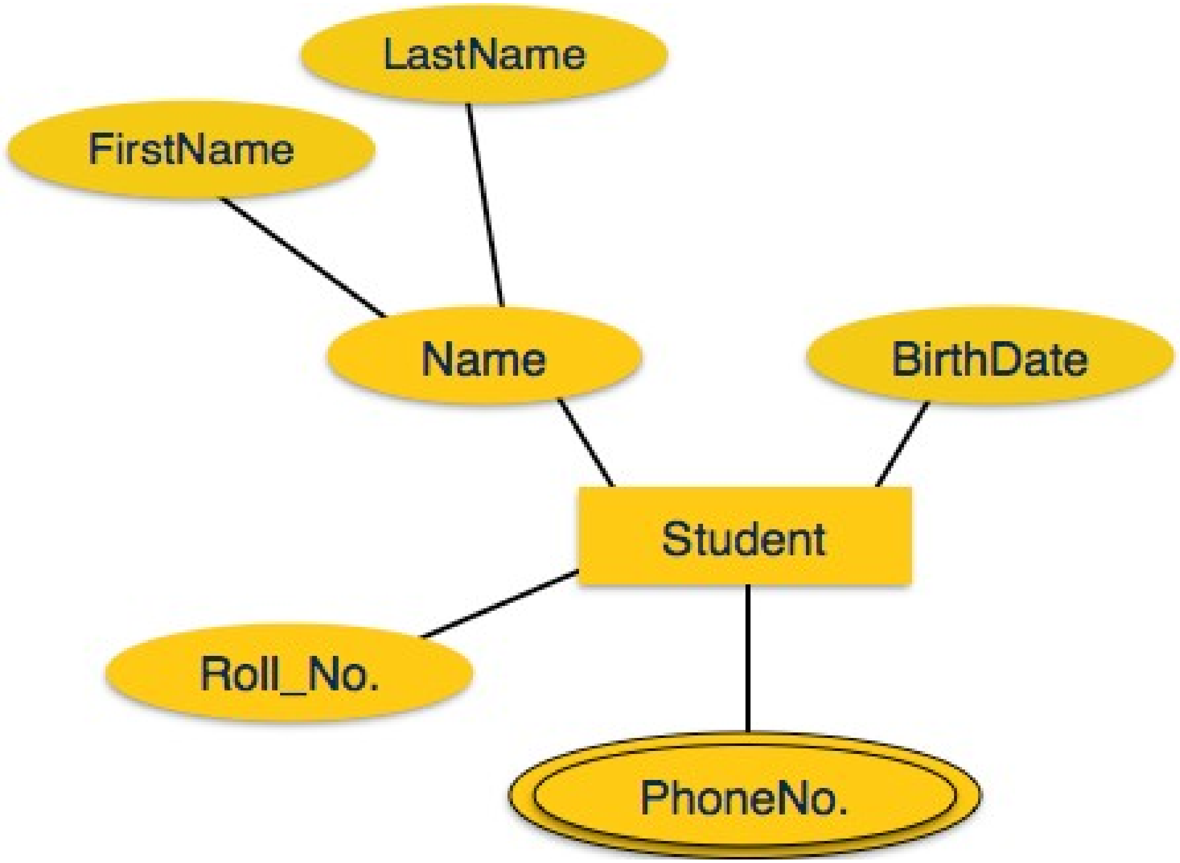
- Attributes are the properties of entities. Attributes are represented by means of ellipses.
- Every ellipse represents one attribute and is directly connected to its entity (rectangle).



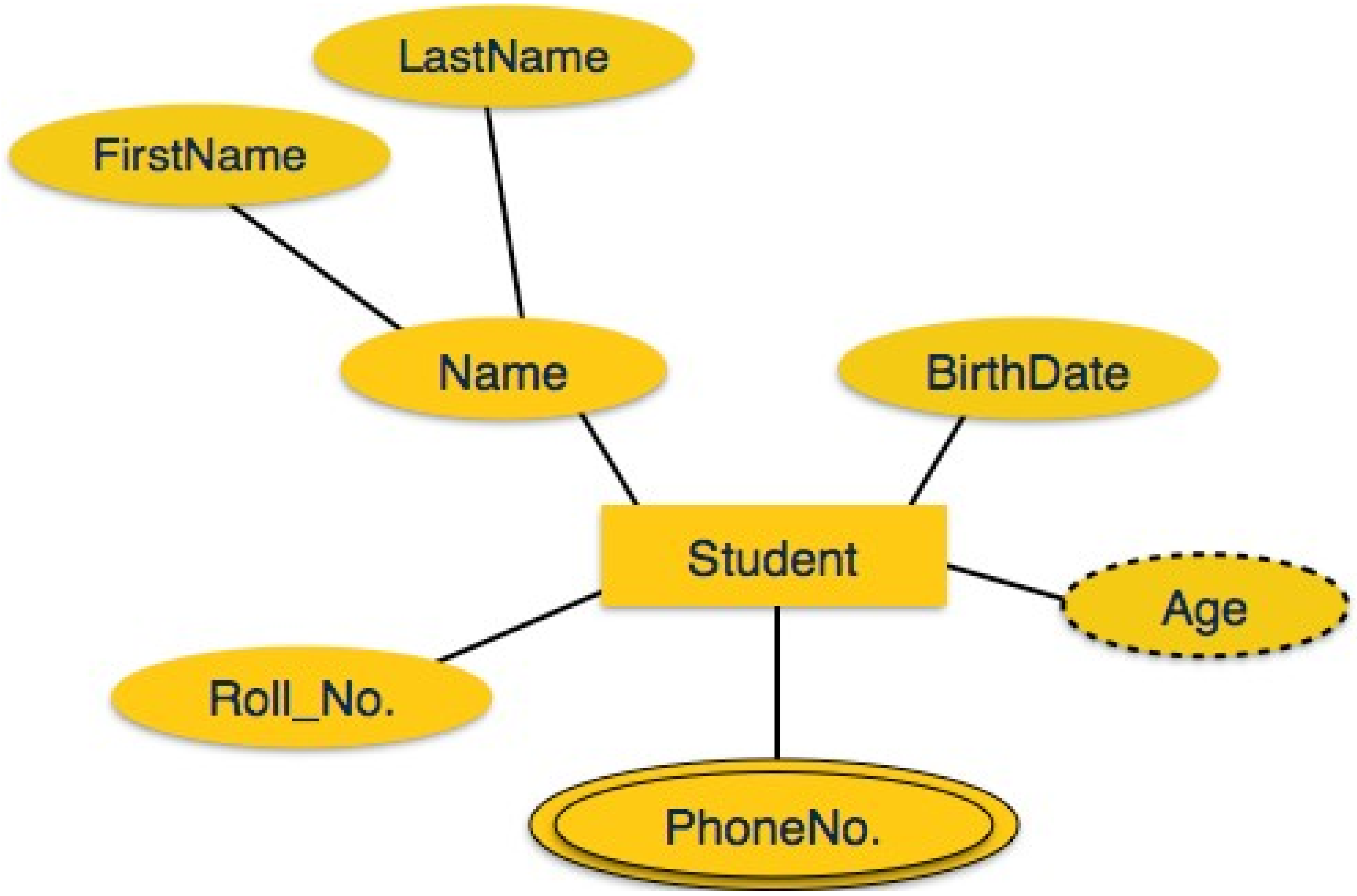
- If the attributes are **composite**, they are **further divided in a tree like structure**. Every node is then connected to its attribute. That is, composite attributes are represented by ellipses that are connected with an ellipse.



- **Multivalued** attributes are depicted by double ellipse.



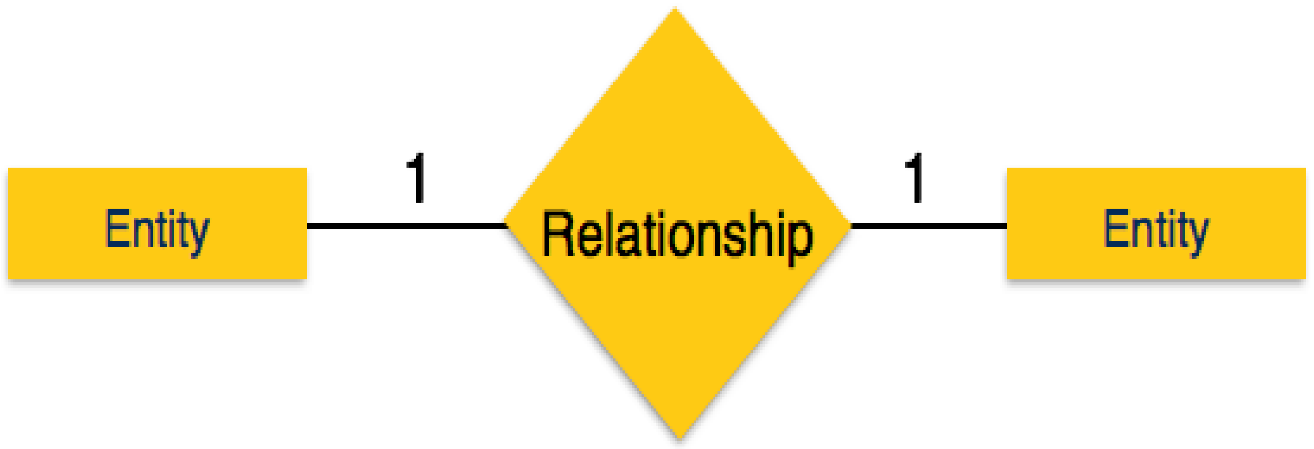
- **Derived** attributes are depicted by dashed ellipse.



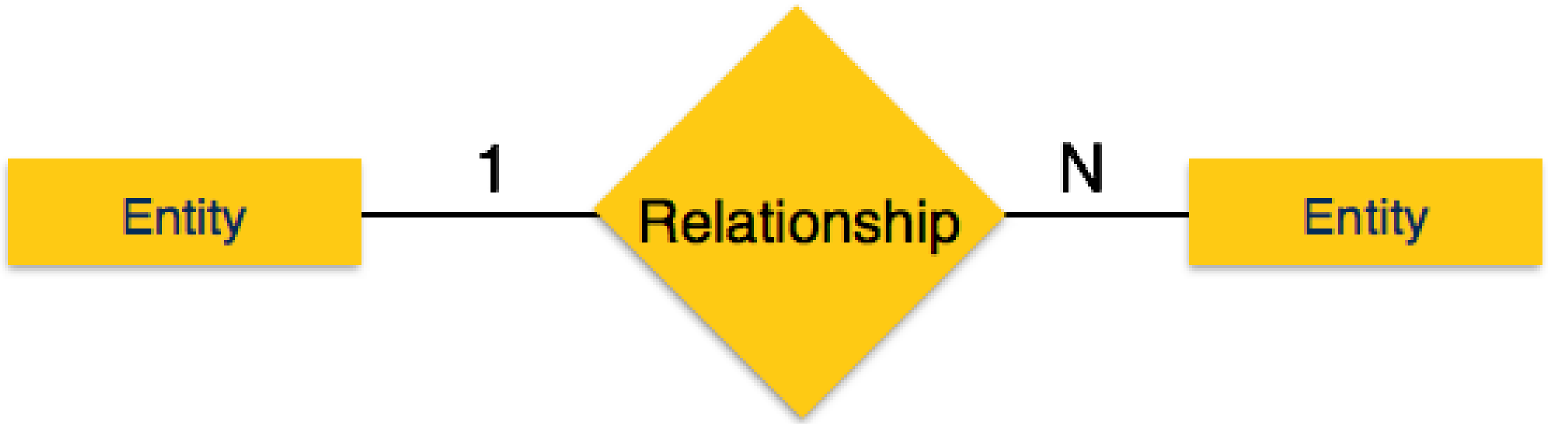
Relationship

- Relationships are represented by **diamond-shaped box**.
- Name of the relationship is written inside the diamond-box.
- All the entities (rectangles) participating in a relationship, are connected to it by a line.
- Binary Relationship and Cardinality
- A relationship where two entities are participating is called a **binary relationship**. Cardinality is the number of instance of an entity from a relation that can be associated with the relation.

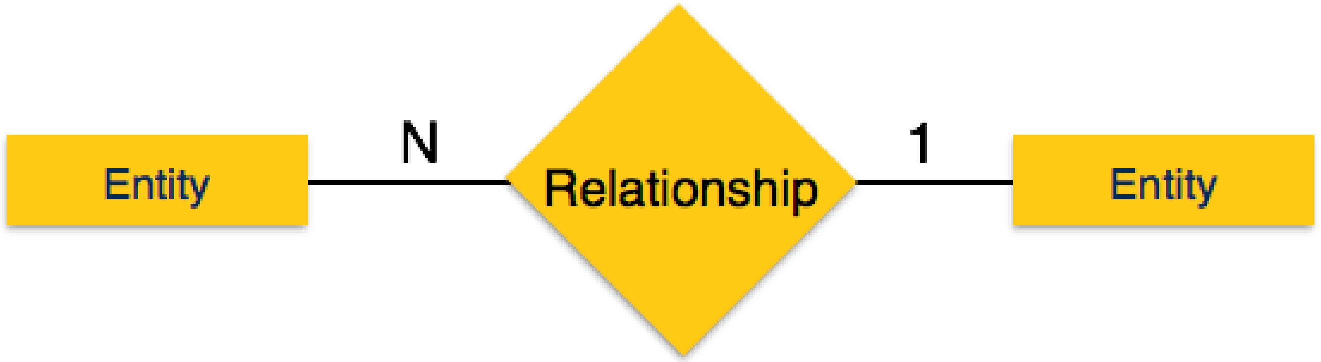
- **One-to-one** – When only one instance of an entity is associated with the relationship, it is marked as '1:1'. The following image reflects that only one instance of each entity should be associated with the relationship. It depicts one-to-one relationship.



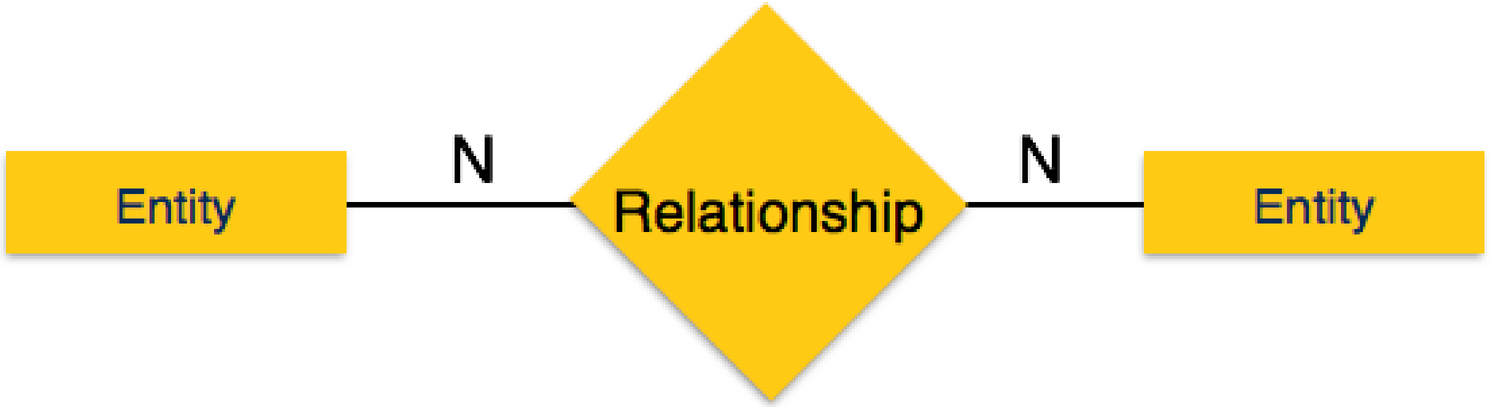
- **One-to-many** – When more than one instance of an entity is associated with a relationship, it is marked as **'1:N'.** The following image reflects that only ***one instance of entity on the left and more than one instance of an entity on the right can be associated with the relationship***. It depicts one-to-many relationship.



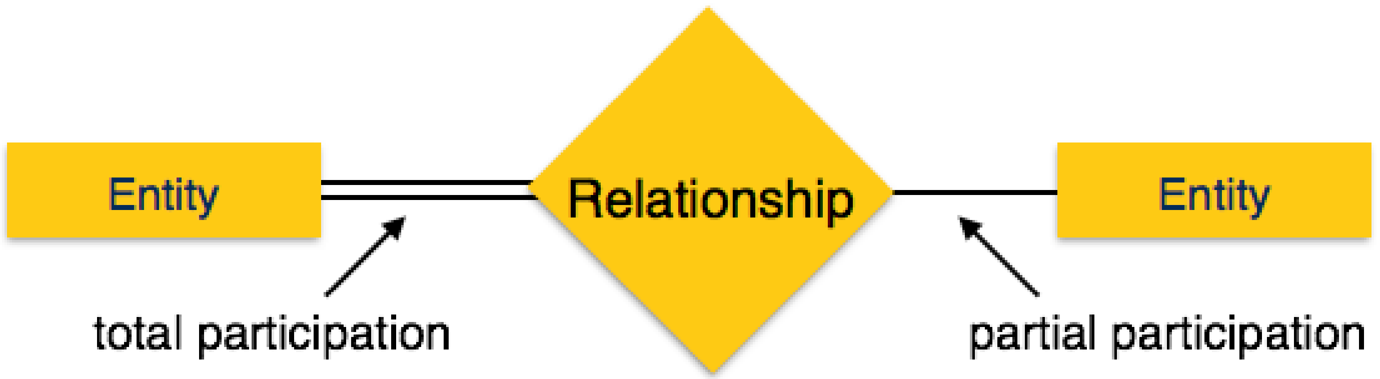
- **Many-to-one** – When ***more than one instance of entity is associated with the relationship***, it is marked as 'N:1'. The following image reflects that more than one instance of an entity on the left and only one instance of an entity on the right can be associated with the relationship. It depicts many-to-one relationship.



- **Many-to-many** – The following image reflects that ***more than one instance of an entity on the left and more than one instance of an entity on the right can be associated with the relationship***. It depicts many-to-many relationship.



- Participation Constraints
- **Total Participation** − Each entity is involved in the relationship. Total participation is represented by double lines.
- **Partial participation** − Not all entities are involved in the relationship. Partial participation is represented by single lines.

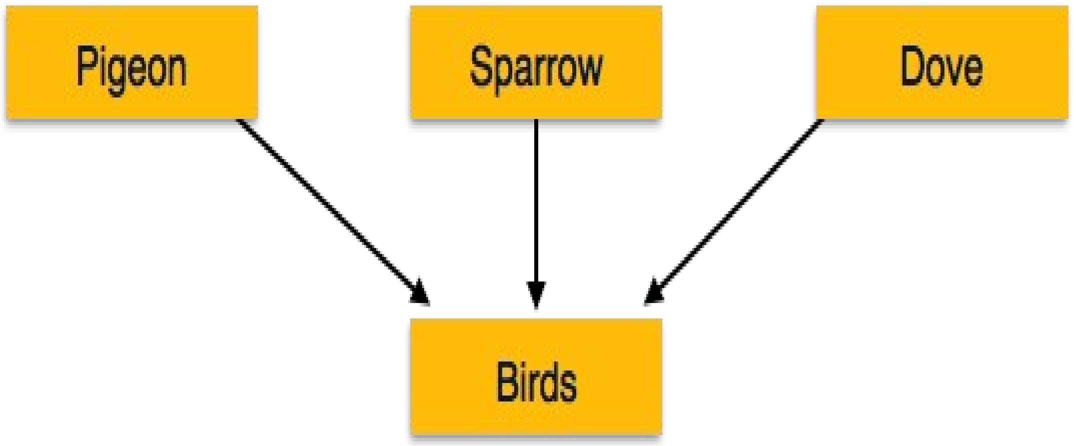


Generalization Aggregation

- The ER Model has the power of expressing database entities in a conceptual ***hierarchical manner***.
- As the hierarchy ***goes up***, it ***generalizes the view of entities***, and as we **go deep** in the hierarchy, ***it gives us the detail of every entity included.***
- Going up in this structure is called **generalization**, where entities are clubbed together to represent a more generalized view.
- For example, a particular student named Mira can be generalized along with all the students.
- The entity shall be a student, and further, the student is a person. The reverse is called **specialization** where a person is a student, and that student is Mira.

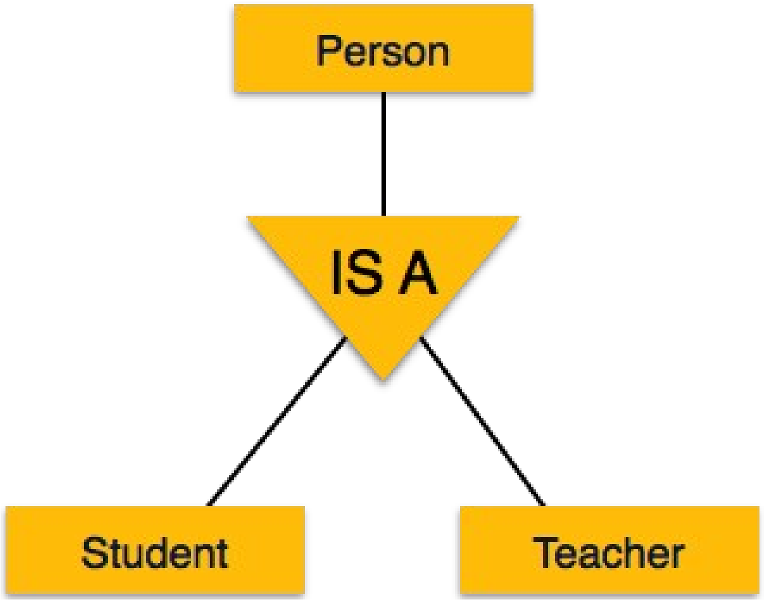
Generalization

- The process of generalizing entities, where the generalized entities contain the properties of all the generalized entities, is called generalization.
- In generalization, a number of entities are brought together into one generalized entity based on their similar characteristics. For example, pigeon, house sparrow, crow and dove can all be generalized as Birds.



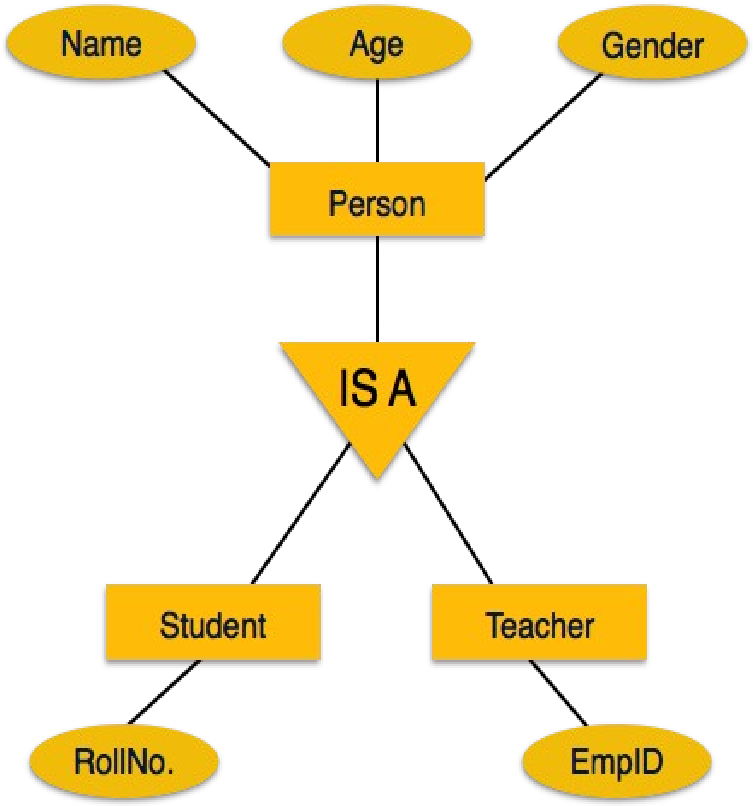
Specialization

- Specialization is the opposite of generalization. In specialization, a group of entities is divided into sub-groups based on their characteristics.
- Take a group '**Person**' for example. A person has **name, date of birth, gender**, etc. These properties are common in all persons, human beings. But in a **company,** persons can be identified as **employee, employer, customer, or vendor,** based on what role they play in the company.
- Similarly, in a **school database**, persons can be specialized as **teacher, student, or a staff**, based on what role they play in school as entities.



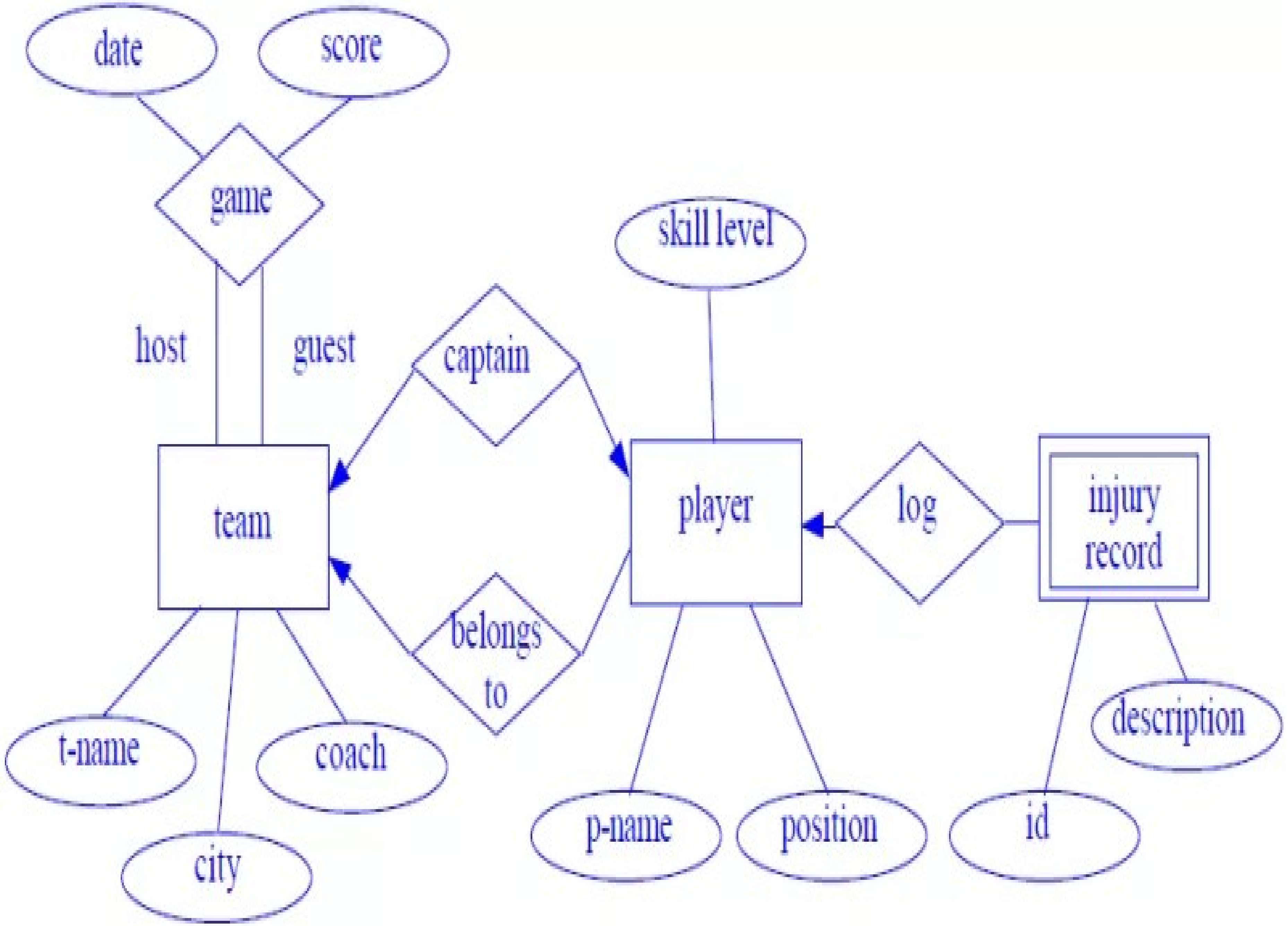
Inheritance

- We use all the above features of ER-Model in order to create classes of objects in object-oriented programming.
- ***The details of entities are generally hidden*** from the user; this process known as **abstraction**.
- Inheritance is an important feature of ***Generalization and Specialization***. It allows lower-level entities to inherit the attributes of higher-level entities.
- For example, the attributes of a Person class such as name, age, and gender can be inherited by lower-level entities such as Student or Teacher.

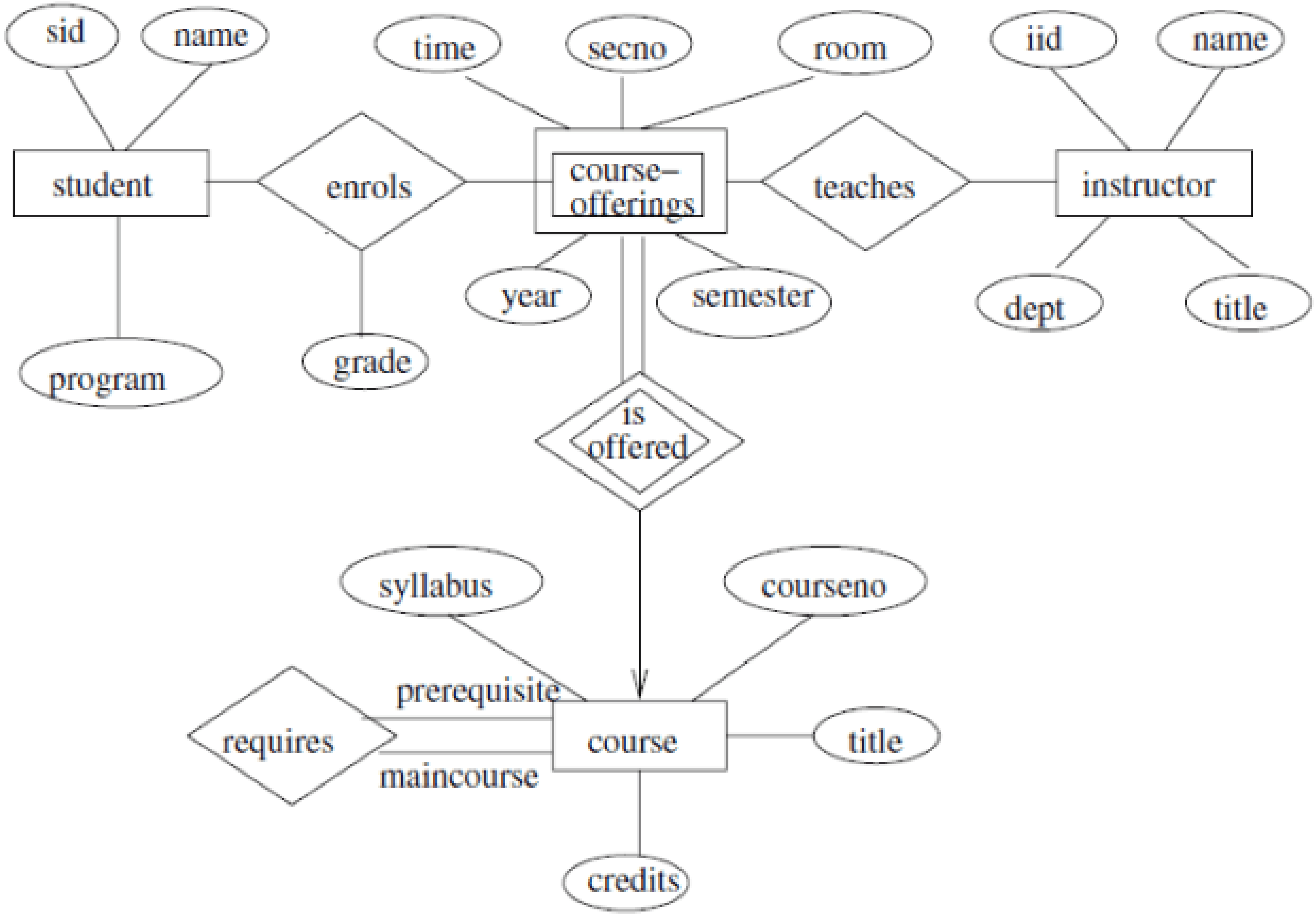


ER Diagram Example

- Suppose you are given the following requirements for a simple database for the National Hockey League (NHL):
- the NHL has many teams,
- each team has a name, a city, a coach, a captain, and a set of players,
- each player belongs to only one team,
- each player has a name, a position (such as left wing or goalie), a skill level, and a set of injury records,
- a team captain is also a player,
- a game is played between two teams (referred to as host_team and guest_team) and has a date (such as May 11th, 1999) and a score (such as 4 to 2).
- Construct a clean and concise ER diagram for the NHL database.


date
score
game
host
guest
captain
skill level
team
player
log
injury record
belongs to
t-name
coach
city
description
p-name
position
id

- Question 2:
- A university registrar’s office maintains data about the following entities:
- courses, including number, title, credits, syllabus, and prerequisites;
- course offerings, including course number, year, semester, section number, instructor(s), timings, and classroom;
- students, including student-id, name, and program;
- instructors, including identi-cation number, name, department, and title.
- Further, the enrollment of students in courses and grades awarded to students in each course they are enrolled for must be appropriately modeled. Construct an E-R diagram for the registrar’s of-ce.Document all assumptions that you make about the mapping constraints.



E-R diagram for a university.

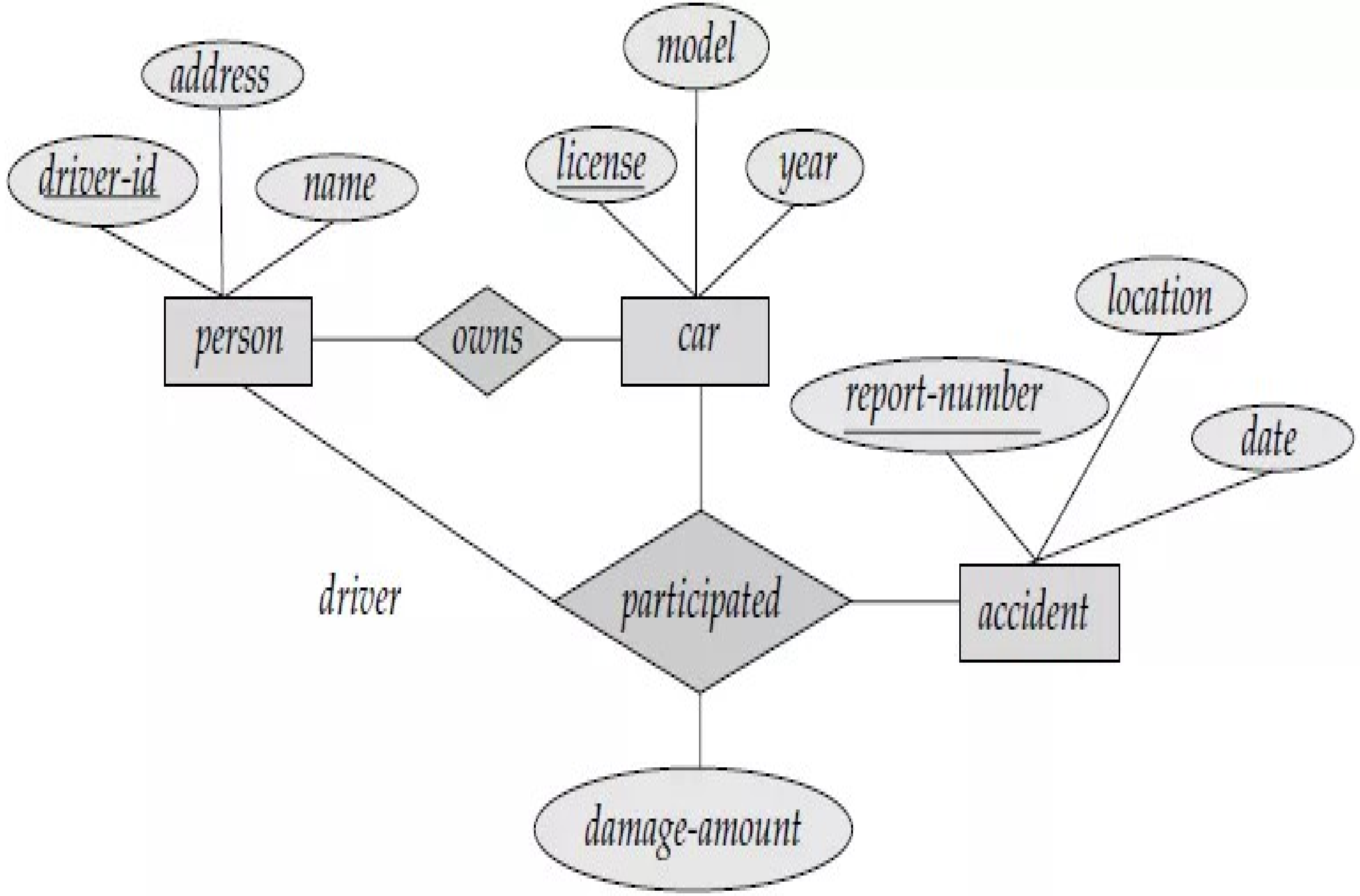
- **Question 3:**

  (a) Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.

  (b) Construct appropriate tables for the above ER Diagram ?

Car insurance tables:

- person (driver-id, name, address)
- car (license, year,model)
- accident (report-number, date, location)
- participated(driver-id, license, report-number, damage-amount)



E-R diagram for a Car-insurance company.