# Entity- Relationship Model (ER Model)

The ER model defines the conceptual view of a database. It works around real-world entities and the associations among them. **At view level, the ER model is considered a good option for designing databases.**

The element in ER diagram are Entity, Attribute, and Relationship. See figure (9-1).

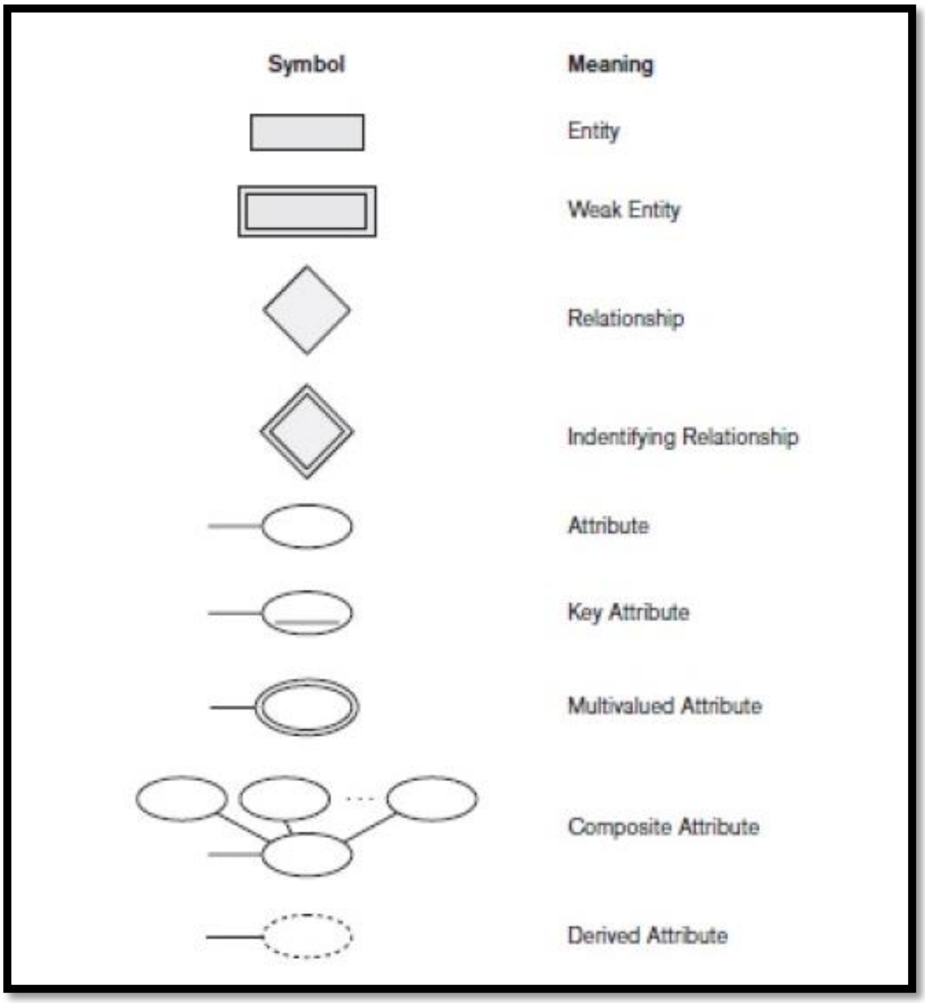


Figure (9-1): The element in ER diagram.

- Entities, defined as tables that hold specific information (data).
- Attributes, each entity has attributes that represented the fields of table.
- Relationships, defined as the associations or interactions between entities.

❖ **Entity**

An *entity* is an object in the real world with an independent existence that can be differentiated from other objects. An entity might be:

- An object with **physical existence** (e.g., a lecturer, a student, a car)
- An object with **conceptual existence** (e.g., a course, a job, a position)

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.

**Entities are represented** by means of **rectangles**. Rectangles are named with the entity set they represent, See figure (9-2).

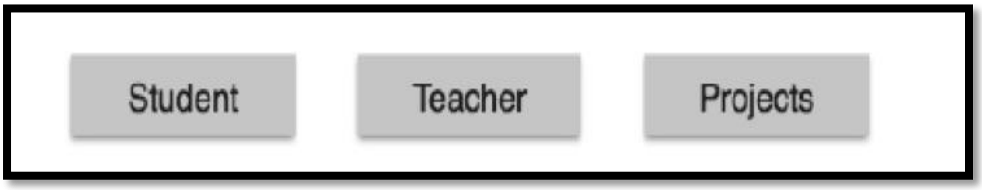


Figure (9-2): The entities .

Entity sets can be broadly classified into:

1- Strong entity.
2- Weak entity.
3- Associative entity.

Strong Entity
Strong entity is one whose existence does not depend on other entity.

Weak Entity
Weak entity is one whose existence depends on other entity. Weak entity does not have primary key.

Example: student takes course. Here student is a strong entity. In this example course is consider as weak entity because, if there are no students to take a particular course. then that course cannot be offered. The COURSE entity depends on the STUDENT entity, See figure (9-3).
.

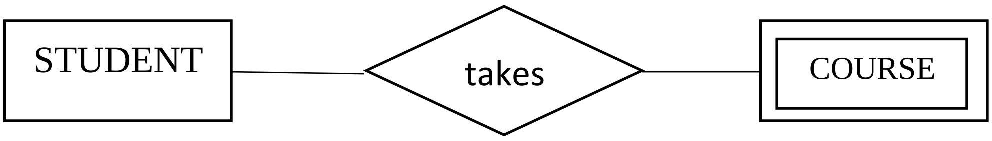
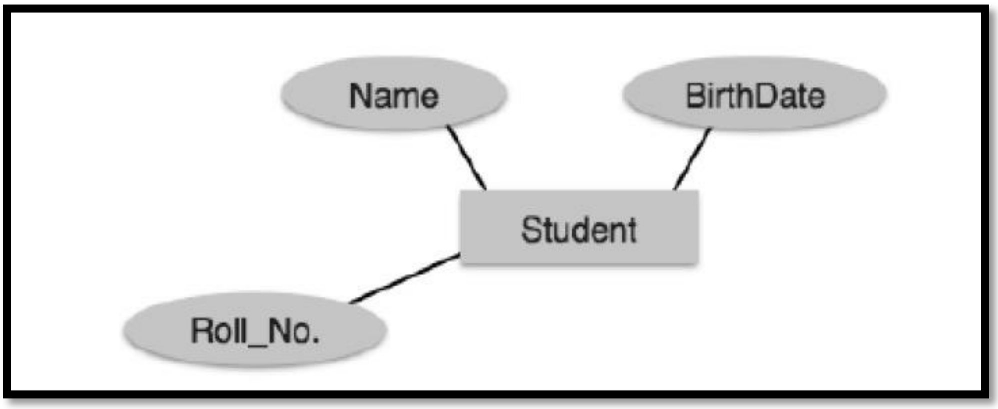


Figure (9-3): The strong entity and weak entity.

### ❖ Attributes

**Each entity is described by a set of attributes**, For example, a student entity may have roll no, name, and birth date as attributes. Each attribute has a name and **represented by an ellipse** Every ellipse represents one attribute and Entity is connected with an attribute with line, shown in figure(9-4).



□

Figure (9-4): The Simple attributes.

### Types of Attributes

Attribute is used to describe the properties of the entity. This attribute can be broadly classified based on value and structure.

**classified based on value:**

**1- Single-value attribute:** Single-value attributes contain single value. For example: Social_Security_Number.

**2- 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. Multivalued attributes are depicted by double ellipse, see figure (9-5).

**3- 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, age can be derived from date_of_birth. Derived attributes are depicted by dashed ellipse, see figure (9-6).

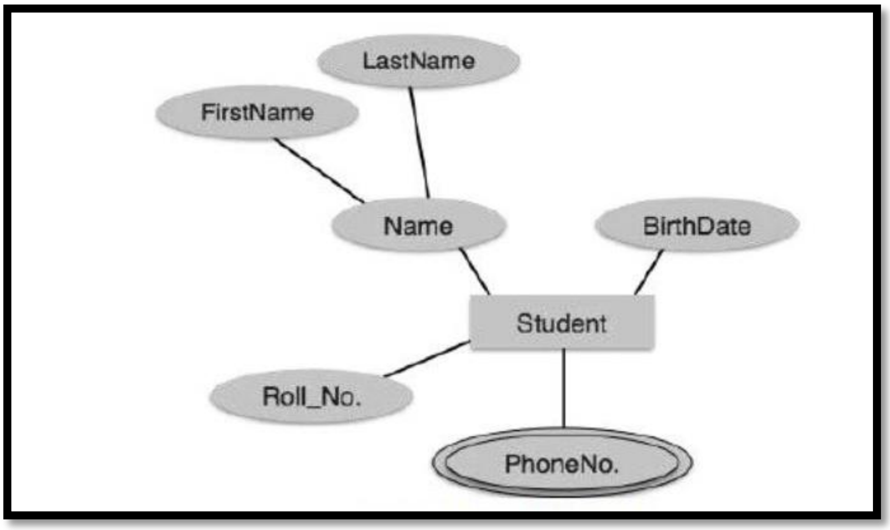


Figure (9-5): The Multi-value attributes .

□

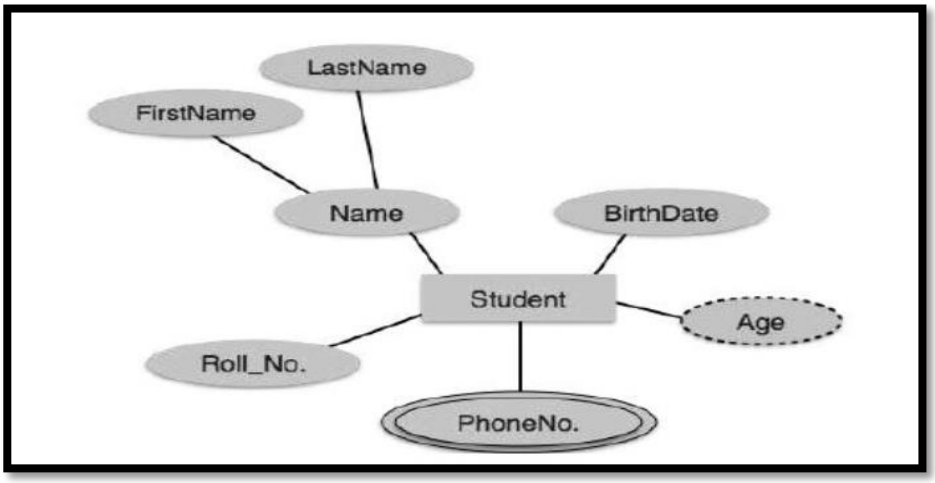


Figure (9-6): The Derived attributes.

**classified based on structure:**

□ **1- 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.

**2- 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. 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, see figure (9-7).

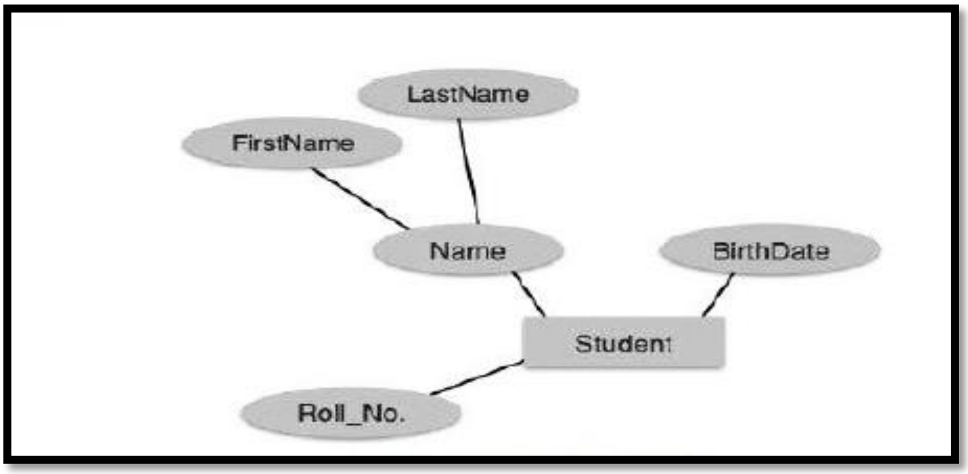


Figure (9-7): The Composite attributes.

### ❖ Relationship:

The association among entities is called a relationship.

Relationships can be thought of as verbs, linking two or more nouns.

Examples : a ***supervises*** relationship between an department and employee.

: a ***performs*** relationship between an artist and a song.

**Types of Relationships**

Below are descriptions of the various types of relationships. In the relational model, tables can be related as any of:

- **many-to-many**
- **many-to-one** (rev. **one-to-many**)
- **one-to-one**

For example, considering a database designed to keep track of hospital records. Such a database could have many tables like:

- a *Doctor* table full of doctor information
- a *Patient* table with patient information
- and a *Department* table with an entry for each department of the hospital.

In that model:

- There is a **many-to-many** relationship between the records in the doctor table and records in the patient table (Doctors have many patients, and a patient could have several doctors);
- a **one-to-many** relation between the department table and the doctor table (each doctor works for one department, but one department could have many doctors).
- **one-to-one** relationship is mostly used to split a table in two in order to optimize access or limit the visibility of some information. In the hospital example, such a relationship could be used to keep apart doctor's personal or administrative information.

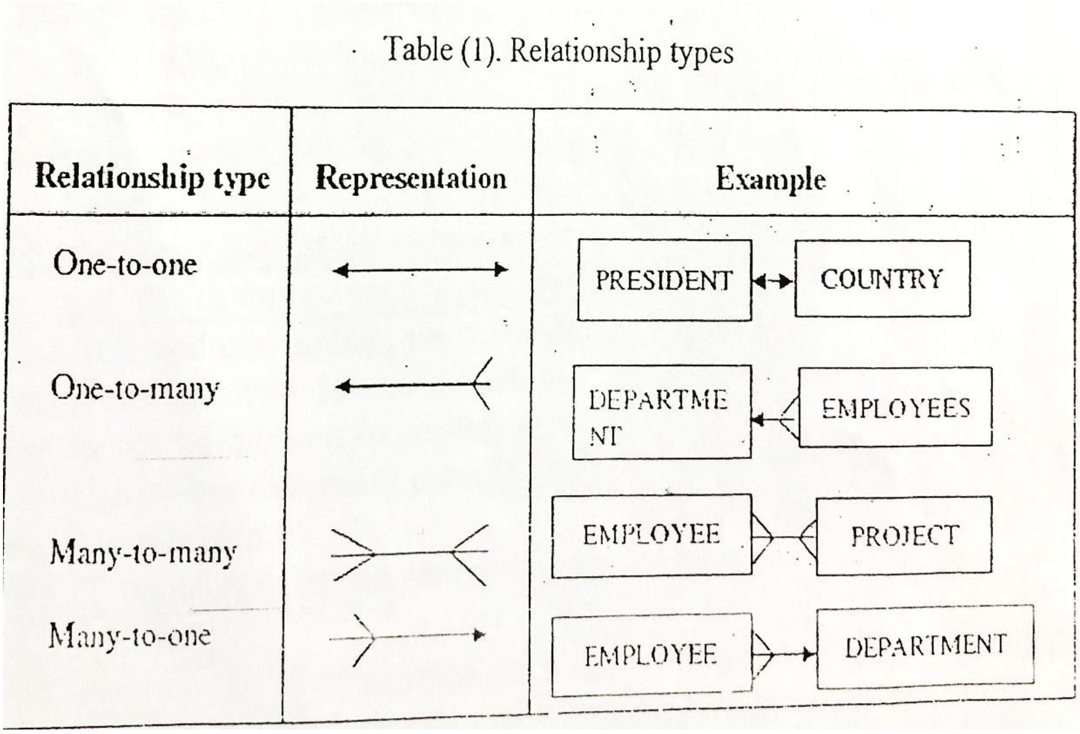


Table (1). Relationship types

| Relationship type | Representation | Example |
|---|---|---|
| One-to-one | ←——→ | PRESIDENT ↔ COUNTRY |
| One-to-many | ←——< | DEPARTMENT ←< EMPLOYEES |
| Many-to-many | >——< | EMPLOYEE >—< PROJECT |
| Many-to-one | >——→ | EMPLOYEE >→ DEPARTMENT |