

# ERD Components

## 1- Entities:

The entity represents a table in the database.

It is the basic element in the system where information is stored.

It can be broad such as: "student", "product", "request", "user", etc.

**And its types:**

### a. Strong Entity:

It is sometimes called the independent entity.

It is the entity that can exist on its own and does not depend on any other entity for its existence.

It has a primary key that distinguishes it.

For example: the entity "student" or "product" is a strong entity because it can exist in the system without the need for another entity to support it.

Representation of the strong entity in the ERD: It is represented by a regular rectangle.

### b. Weak Entity:

It is also called the dependent entity.

It is an entity that cannot exist on its own, and depends on another entity for its existence.

It cannot have an independent primary key, but rather depends on a primary key from another entity (usually a strong entity).

For example: "Order Record" which is based on the "Order" entity to be identified.

Requires a composite key, which includes the primary key of the strong entity and its private key.

Representation of the weak entity in the ERD: It is represented by a double rectangle.

### c. Associate Entity:

It is used to represent complex relationships, especially when there is a "many-to-many" (M ) relationship between entities.

It may contain its own attributes along with the primary keys of the associated entities.

This type is sometimes used to convert an M

relationship to 1

and M:1 relationships to facilitate database implementation.

Representation of the Integral Entity in the ERD: It is represented by a rectangle containing two associated entities plus the added attributes.

#### **d. Subtype Entity:**

It is used when you have a general entity and derive specialized sub-entities from it.

It allows you to split a general entity into specialized entities based on certain properties.

For example: The entity "Vehicle" may contain sub-entities such as "Car", "Truck", and "Motorcycle", where each sub-entity has its own attributes.

Sub-entity representation in ERD: It is represented by a hierarchical relationship (tree), often using a line connecting the general entity and the sub-entities.

#### **e. Aggregation Entity:**

It is used to represent a relationship that contains a set of relationships or entities, and is a more abstract type.

It is used when the relationship between the entities themselves is very complex and needs an additional level of aggregation.

Aggregation Entity representation in ERD: It is usually represented using boxes that overlap or use complex relationships between entities.

### **2- Attributes:** Attributes are the harvest that is being explained.

The speaker in ERD is represented as a circle or oval connected to the entity.

For example, the entity "Student" may have attributes such as: "Name", "University Number", "Age".

#### **Types of Attributes:**

##### **a. Simple Attribute:**

It is an attribute that cannot be divided into smaller attributes or other components.

It contains only one value.

For example: the attribute "age" or "name" may be a simple attribute because it expresses only one value.

Representation in ERD: It is drawn as a single oval connected to the entity.

## **b. Composite Attribute:**

It is an attribute that can be divided into smaller parts, and each part expresses a sub-attribute.

For example: the attribute "address" can consist of sub-attributes such as "street", "city", and "zip code".

Representation in ERD: The primary attribute is drawn as a large oval connected to the entity, and the sub-attributes are connected to the composite attribute as smaller ovals.

## **c. Derived Attribute:**

It is an attribute that is not stored directly in the database, but is calculated based on other data.

For example: The attribute "Age" can be derived from the attribute "Date of Birth".

Representation in ERD: It is represented by a dotted (scattered) oval to indicate that it is derived.

## **d. Single-Valued Attribute:**

It is the attribute that has only one value for each record (instance) of the entity.

For example: The attribute "University Number" for a particular student has only one value.

Representation in ERD: It is represented by a plain oval.

## **e. Multi-Valued Attribute:**

It is the attribute that may have more than one value for each record.

For example: The attribute "Hobbies" for a particular student may have more than one hobby.

Representation in ERD: It is represented by a double oval, which means that the attribute may have multiple values.

## **f. Key Attribute:**

It is the attribute that is used to uniquely identify a record within the entity.

Primary Key is an example of a key attribute.

For example: "National Number" or "ID Number" is a key attribute because it uniquely identifies each individual.

Representation in ERD: The key attribute is indicated by an underline.

## **g. Non-Key Attribute:**

It is the attribute that is not used to uniquely identify the record. They are just additional attributes to describe the entity.

For example: "Address" and "Telephone" are non-key attributes because they do not uniquely identify the entity.

Representation in ERD: It is drawn as a regular oval without underlines.

## **h. Partially Derived Attribute:**

It is an attribute that is partially derived based on data present in the system, but may need external data to complete the derivation.

For example: "Final Total" for products purchased may depend on an external discount or price that may change.

# **2- Relationships:**

Relationships include the primary entities, and show how they interact.

The distinction is represented in the form of a diamond between entities.

**Types of relationships:**

### **a. Degree:**

Definition: It refers to the number of entities that participate in the relationship.

**Types of Degree:**

Binary Relationship: If the relationship includes two entities (which is the most common). For example, a relationship between "student" and "course".

**Ternary Relationship:** If the relationship includes three entities. For example, a relationship between "student", "instructor", and "course" to enroll the student in the course supervised by the instructor.

**Unary Relationship:** If the relationship includes one entity that is related to itself. For example, an "employee" supervises another employee.

**Usage:** The degree of the relationship determines the number of entities participating in the relationship. Most relationships in databases are binary relationships.

## **b. Cardinality:**

**Definition:** It refers to the number of entities that can participate in the relationship from each side.

### **Types of Cardinality:**

- **One-to-one (1:1):** As mentioned earlier, each record in the first entity is related to only one record in the second entity.
- **One to many (1):** A single record in the first entity can be associated with multiple records in the second entity.
- **Many to many (M):** Each record in the first entity can be associated with multiple records in the second entity, and vice versa.
- **Usage:** Multiplicity shows how closely the records in each entity are related to the relationship and helps in understanding the structure of the data better.

## **c. Participation:**

**Definition:** It indicates whether the entity is a mandatory or optional participant in the relationship.

### **Types of Participation:**

- **Total Participation:** It occurs when every entity of the first type must be a participant in the relationship. In this case, every record of the first entity must have an association in the relationship. It is represented by a double line between the entity and the relationship.

**Example:** Every "employee" must have an "ID number", where every employee must have a license associated with it.

- **Partial Participation:** It occurs when the entity has the option to participate in the relationship, where every entity does not need to be associated. It is represented by a single line between the entity and the relationship.

Example: Not every “employee” has a company car, which means that the participation in the “employee - company car” relationship is partial.

## ERD Company Example :

The company is organized into **DEPARTMENTS**. Each department has a unique **name**, a unique **number**, and an employee who manages the department.

We keep track of the start date of the department manager.

A department may have several locations. Each department controls/a number of **PROJECTS**.

Each **project** has a unique **name**, and unique **number** and is located at a single location We store each **EMPLOYEE's** **SSN**, **address**, **salary**, **sex**, and **birthdate**

- Each employee works for one department but may work on several projects. We keep track of the number of hours per week that on employee currently works on each project.
- We also keep track of the direct supervisor of each employee Each employee may have a number of **DEPENDENTS**.

For each **dependent**, we keep track of their name, sex, birth date, and relationship to the employee.

## The Website:

<https://www.lucidchart.com/pages/er-diagrams>