Entity-Relationship Notation

An **Entity-Relationship Diagram** (ER Diagram or ERD) is a model used to represent the relationships between entity types in a database.

It serves as an early step for designing a database system.

NOTE: There is no industry standard for ER notations. As such, there are some differences in notation depending on the organization or the drawing tool used.

Entity Types

An **entity type** is a collection of entities that share common properties.

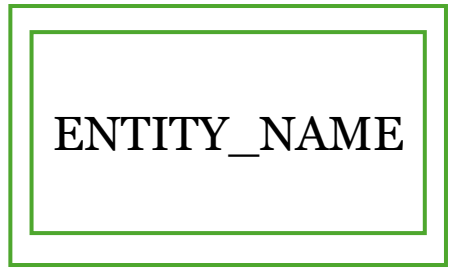
An **entity instance** is a single occurrence of an entity type.

Strong Entity

- A strong entity type exists independently of any other entity type.
- Represented by a rectangle.

Weak Entity

- A weak entity relies on another entity type to exist.
  - The entity type that the weak entity relies on is called the 'owner' or 'owner entity.'
- Represented by a double rectangle.

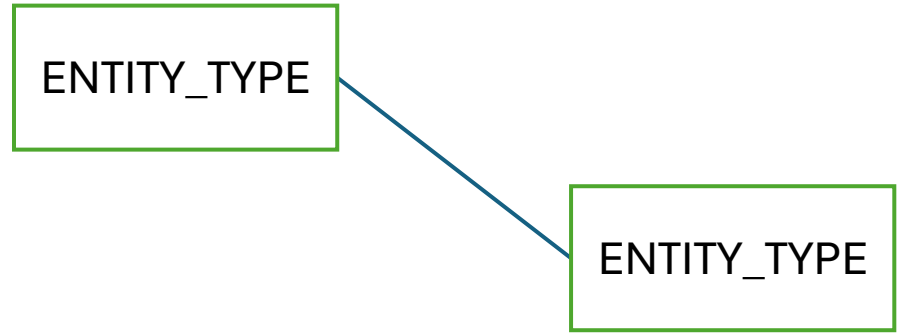


Associative Entity

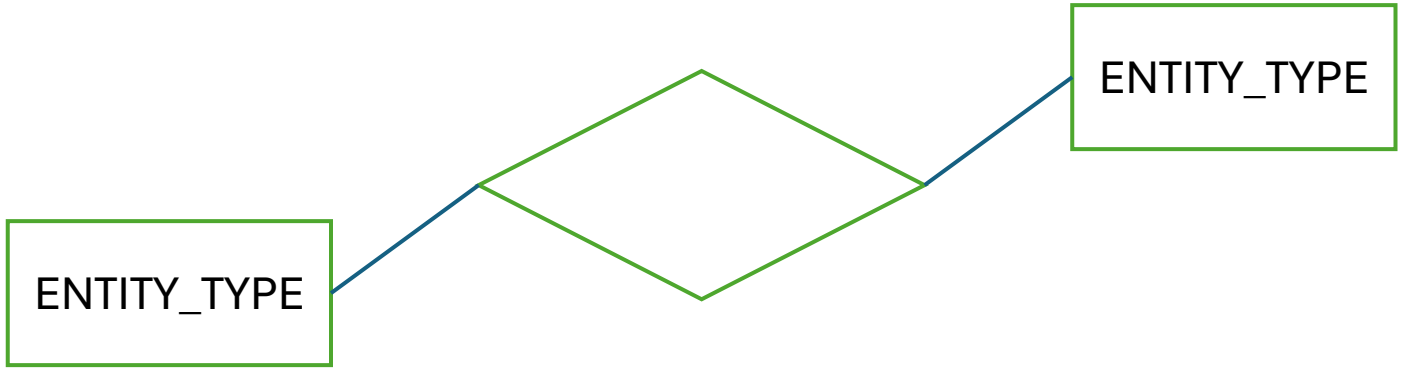
- An associative entity exists to show the relationship between two or more other entities.
- May be used to replace a relationship that contains its own attributes.
- Represented by a rectangle with rounded edges.

Relationships

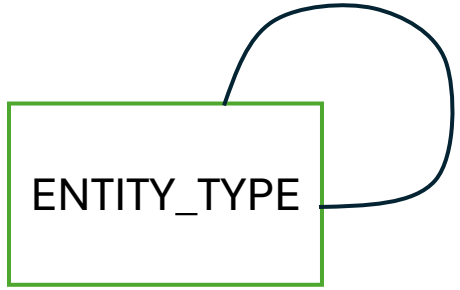
An **entity relationship** may be represented by a line connecting one or more entity types or by both a line and a diamond.



The purpose of a relationship is to show how entity types or entity attributes relate to one another.

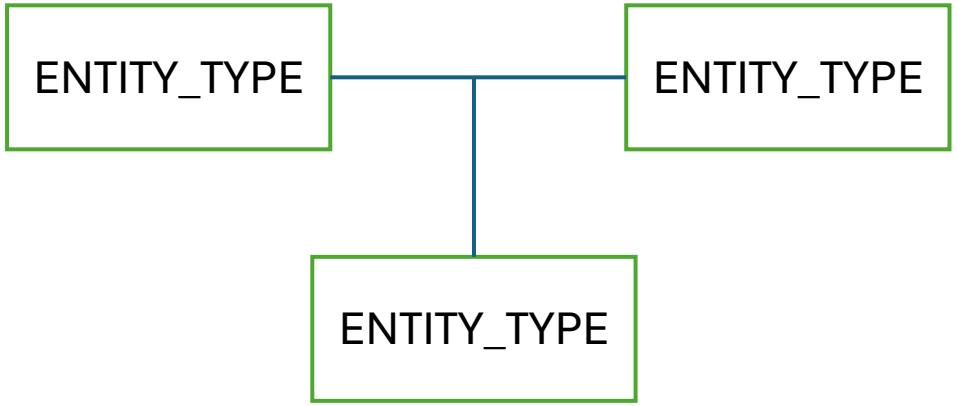


Relationship Degrees



Unary Relationship : A relationship with only one participating entity type.

Binary Relationship : A relationship existing between two entity types.



Ternary Relationship : A relationship existing between three entity types.

Relationship Cardinality

One-to-One 1:1

One-to-Many 1:M

Many-to-One M:1

Many-to-Many M:N or M:M

Optional One

Mandatory One

Optional Many

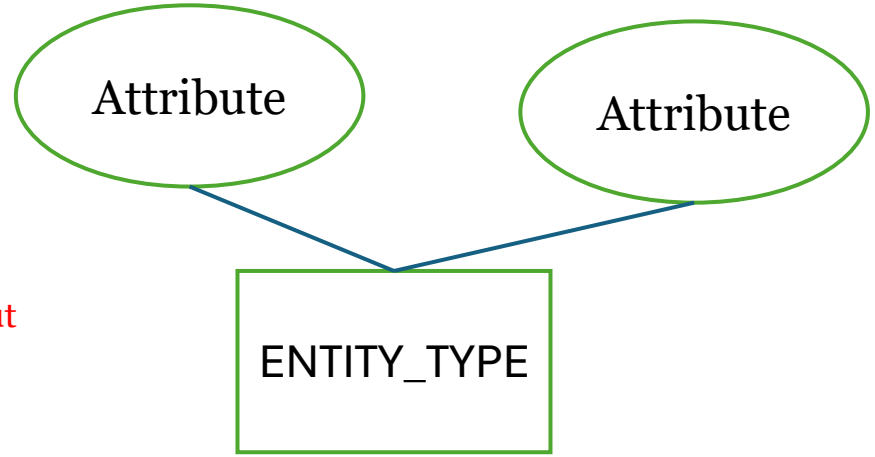
Mandatory Many

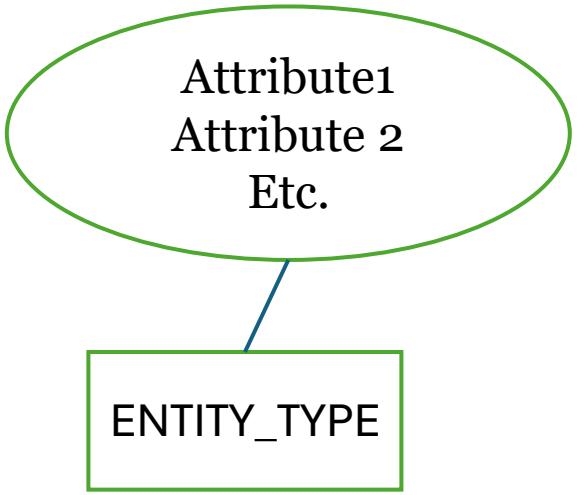
Attributes

An attributed may be represented as an oval connected to an entity type.

A **required attribute** is one that necessitates the input of a value. In an ER diagram, it may be in bold font or have an asterisk (*) located in front.

An **optional attribute** is one that does not necessitate the input of a value. It may be represented by merely by lacking a bold font or an 'o' may be placed in front.





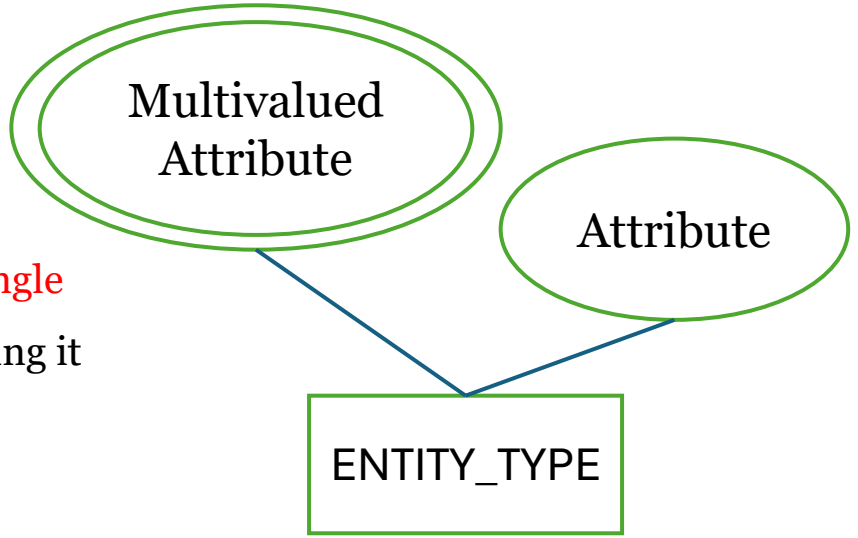
A **composite attribute** is one that can be divided into more specific components.

(for example: Name → First_Name, Middle_Name, Last_Name)

- Antonym: **simple attribute** or **atomic attribute**

A **multivalued attribute** is one that may have more than one value for a single instance of the entity type. It may be written with curly brackets { } surrounding it or as a double oval.

- Antonym: **single-valued attribute**



Enhanced Entity-Relationship Notation

An **Enhanced Entity-Relationship Diagram** (EER Diagram or EERD) is a more advanced version of an ERD.

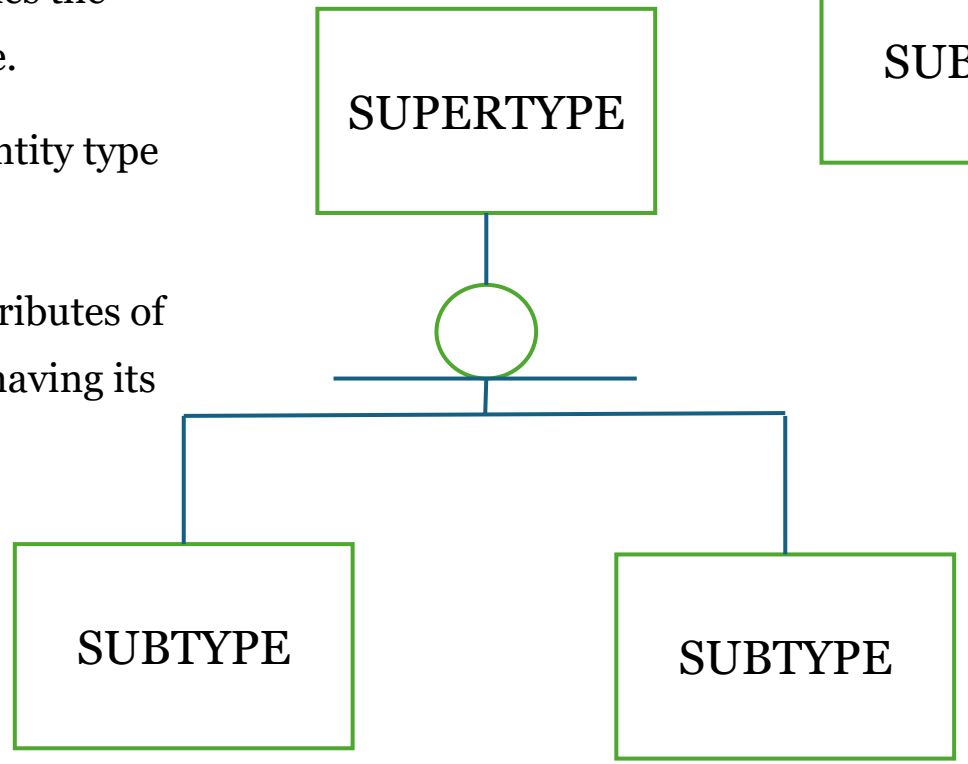
Supertypes and Subtypes

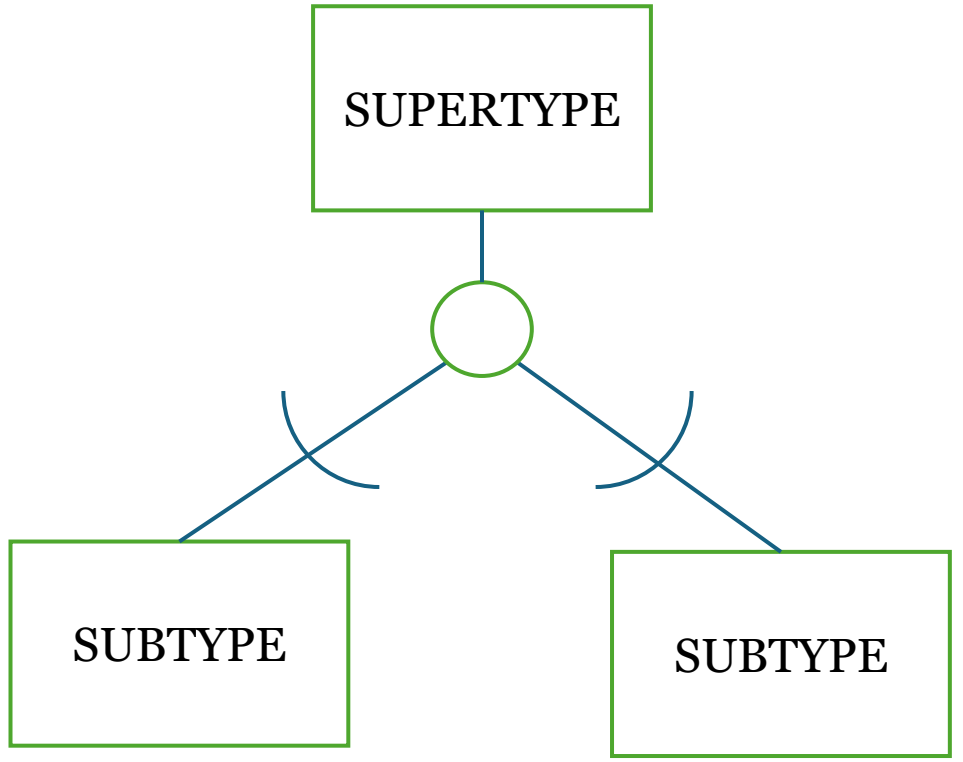
A **supertype** refers to a generic entity type.

- May have more than one subtype.
- Has an attribute called a **subtype discriminator** that identifies the target subtype of an instance.

A **subtype** is a more specific entity type derived from a supertype.

- A subtype inherits all the attributes of its supertype in addition to having its own unique attributes.

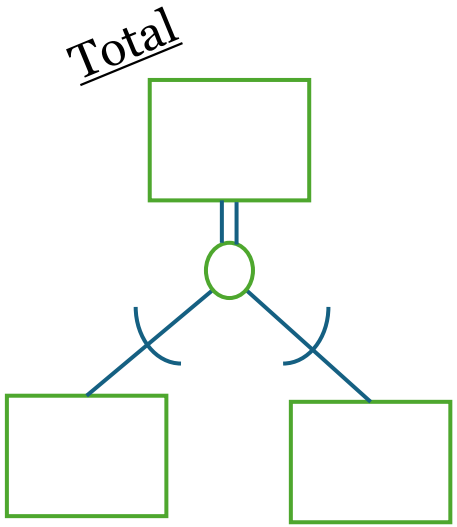


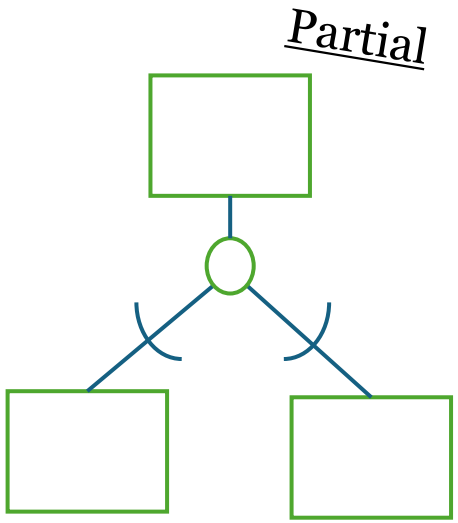


Specialization: A top-down process where subtypes are created from existing supertypes.

Generalization: A bottom-up process where a supertype is created to unite existing entity types with similar properties.

Constraints





Completeness Constraint: Determines whether the instance of a supertype must also belong to a subtype.

Total Specialization- An instance must belong to a subtype.

Partial Specialization- An instance may or may not belong to a subtype.

Disjointness Constraint: Determines whether an instance can belong to more than one subtype simultaneously.

Disjoint Rule- Allows an instance to belong to only one subtype.

Overlap Rule- Allows an instance to belong to more than one subtype simultaneously.