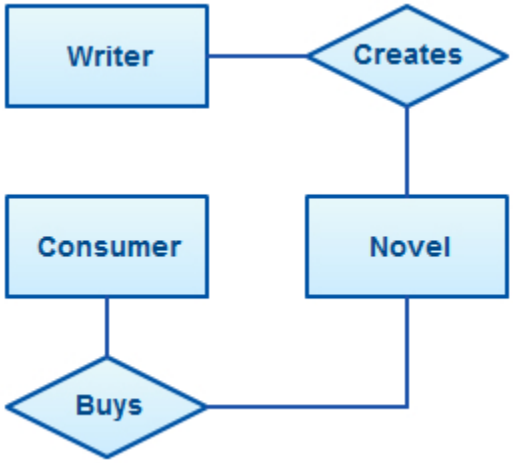
ER Diagram Tutorial (Entity Relationship Diagrams)

This ER diagram tutorial will cover their usage, history, symbols, notations and how to use our ER diagram software to draw them. We've also added some templates for you to get started quickly.

What is an ER diagram?

An Entity Relationship Diagram (ERD) is a visual representation of different entities within a system and how they relate to each other. For example, the elements writer, novel, and a consumer may be described using ER diagrams the following way:



ER diagram with basic objects

History of ER Diagrams

Although data modeling has become a necessity around 1970's there was no standard way to model databases or business processes. Although many solutions were proposed and discussed none were widely adopted.

Peter Chen is credited with introducing the widely adopted ER model in his paper "The Entity Relationship Model-Toward a Unified View of Data". The focus was on entities and relationships and he introduced a diagramming representation for database design as well.

His model was inspired by the data structure diagrams introduced by Charles Bachman. One of the early forms of ER diagrams, Bachman diagrams are named after him.

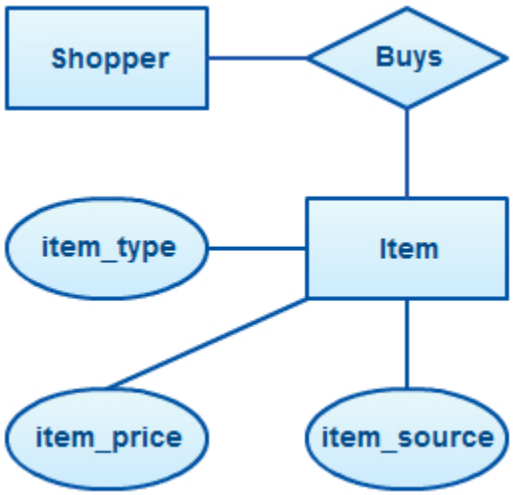
ER Models in Database Design

They are widely used to design relational databases. The entities in the ER schema become tables, attributes and converted the database schema. Since they can be used to visualize database tables and their relationships it's commonly used for database troubleshooting as well.

ER diagrams in software engineering

Entity relationship diagrams are used in software engineering during the planning stages of the software project. They help to identify different system elements and their relationships with each other. It is often used as the basis for data flow diagrams or DFD's as they are commonly known.

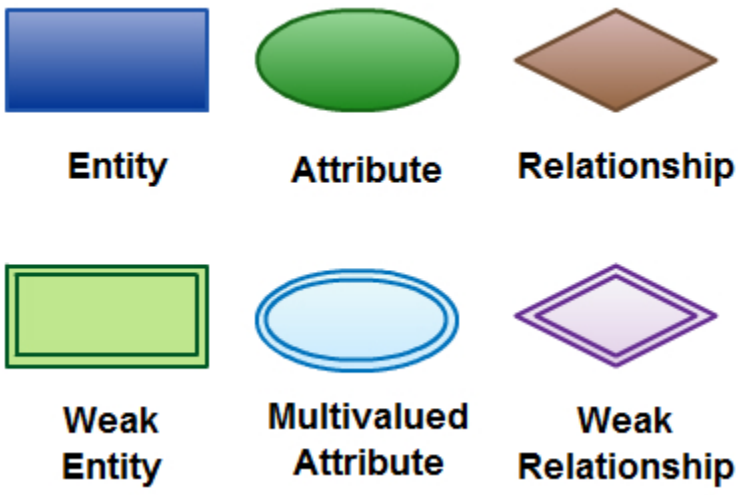
For example, an inventory software used in a retail shop will have a database that monitors elements such as purchases, item, item type, item source and item price. Rendering this information through an ER diagram would be something like this:



ER diagram example with entity having attributes

In the diagram, the information inside the oval shapes are attributes of a particular entity.

ER Diagram Symbols and Notations



Elements in ER diagrams

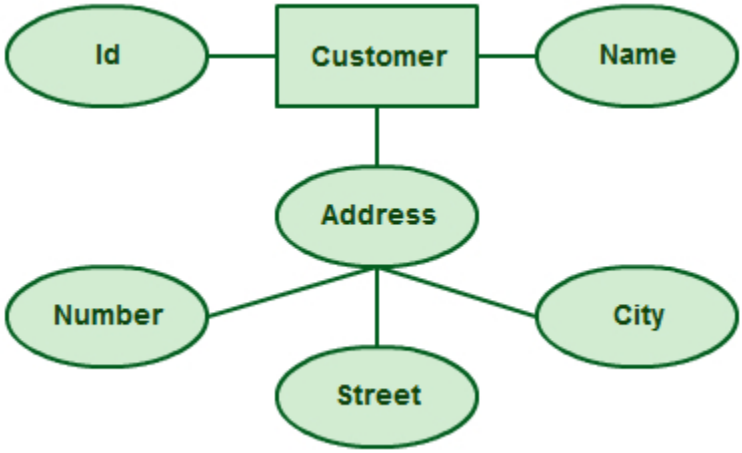
There are three basic elements in an ER Diagram: entity, attribute, relationship. There are more elements which are based on the main elements. They are weak entity, multi valued attribute, derived attribute, weak relationship, and recursive relationship. Cardinality and ordinality are two other notations used in ER diagrams to further define relationships.

Entity

An entity can be a person, place, event, or object that is relevant to a given system. For example, a school system may include students, teachers, major courses, subjects, fees, and other items. Entities are represented in ER diagrams by a rectangle and named using singular nouns.

Attribute

An attribute is a property, trait, or characteristic of an entity, relationship, or another attribute. For example, the attribute Inventory Item Name is an attribute of the entity Inventory Item. An entity can have as many attributes as necessary. Meanwhile, attributes can also have their own specific attributes. For example, the attribute “customer address” can have the attributes number, street, city, and state. These are called composite attributes. Note that some top level ER diagrams do not show attributes for the sake of simplicity. In those that do, however, attributes are represented by oval shapes.



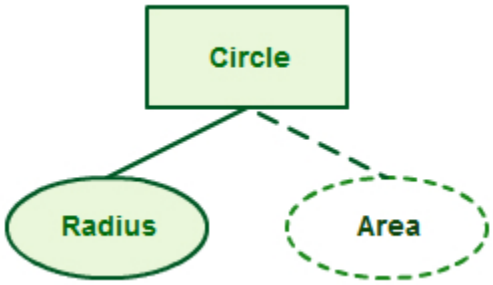
Multivalued Attribute

If an attribute can have more than one value it is called a multi-valued attribute. It is important to note that this is different from an attribute having its own attributes. For example, a teacher entity can have multiple subject values.

Example of a multivalued attribute

Derived Attribute

An attribute based on another attribute. This is found rarely in ER diagrams. For example, for a circle, the area can be derived from the radius.



Derived Attribute in ER diagrams

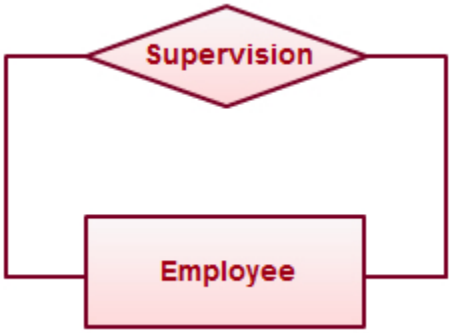
Relationship

A relationship describes how entities interact. 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.

Using Relationships in Entity Relationship Diagrams

Recursive Relationship

If the same entity participates more than once in a relationship it is known as a recursive relationship. In the below example an employee can be a supervisor and be supervised, so there is a recursive relationship.



Example of a recursive relationship in ER diagrams

Cardinality and Ordinality

These two further defines relationships between entities by placing the relationship in the context of numbers. In an email system, for example, one account can have multiple contacts. The relationship, in this case, follows a “one to many” model. There are a number of notations used to present cardinality in ER diagrams. Chen, UML, Crow’s foot, Bachman are some of the popular notations. The following example uses UML to show cardinality.

Cardinality in ER diagrams using UML notation

How to Draw ER Diagrams

Below points show how to go about creating an ER diagram.

- Identify all the entities in the system. An entity should appear only once in a particular diagram. Create rectangles for all entities and name them properly.
- Identify relationships between entities. Connect them using a line and add a diamond in the middle describing the relationship.
- Add attributes for entities. Give meaningful attribute names so they can be understood easily.

Sounds simple right? In a complex system, it can be a nightmare to identify relationships. This is something you’ll perfect only with practice.

ER Diagram Best Practices

- Provide a precise and appropriate name for each entity, attribute, and relationship in the diagram. Terms that are simple and familiar always beats vague, technical-sounding words. In naming entities, remember to use singular nouns. However, adjectives may be used to distinguish entities belonging to the same class (part-time employee and full-time employee, for example). Meanwhile attribute names must be meaningful, unique, system-independent, and easily understandable.
- Remove vague, redundant or unnecessary relationships between entities.
- Never connect a relationship to another relationship.
- Make effective use of colors. You can use colors to classify similar entities or to highlight key areas in your diagrams.