What is an ER Diagram?

**Helpful Terminology**

* Entity - A thing about which we are storing the information.
  + Some examples that could be entities: a person, a building, a restaurant
  + An entity can be pretty much anything we are wanting to store information or data about
* Relationship - The connection between Entities
  + There are different types of relationships between entities (one-to-one, one-to-many, many-to-one) but don't worry about these right now.
* Attributes - Data categories which describe our entity.
  + An example: If you have an entity of a Building, some attributes might be Name, Address, Number of Floors, etc.
* Primary Key - a unique identifier designated to uniquely identify each table record
  + Example: a person's social security number, a car's VIN number
  + Note: Primary Keys cannot be null (This doesn't matter for drawing an ER diagram just an important piece of information to know.)
* Foreign Key - acts like a cross-reference between tables because it references the primary key of another table

**What is an Entity Relationship Diagram or ERD?**

The Entity Relationship Diagram or ERD visualizes the entity or entities and their relationships and attributes. The ER diagram illustrates a logical structure of the database(s). It serves as a form of documentation. It's also helpful if you are working with a client who is not knowledgeable in databases to be able to explain what is happening and how information flows and is connected in layman's terms.

**Why Make an ERD?**

* It can help you as a developer discuss with your client their wishes and make sure you are understanding each other clearly.
* It can serve to documentation the development structure
* It can help to design a new database, gather design requirements or re-engineer and existing database
* Debug, analyze, and troubleshoot

One of the goals, beyond documentation and clear communication, is to build a database that is logically is connecting information. What do I mean by this? You don't want to have redundant information stored in multiple tables - there's just no need for it. The ER diagram helps to know that all information is getting into the database without numerous overlaps because that just means more storage which ultimately costs more money for the organization. You also don't want related information that can "live" in one table scattered across many.

The database we'll be working with this semester is fairly simple and definitely small, but when you get into the real-world that won't be the case. Understanding the way information is connected and related will help you organize and be proficient in database creation and management.

 Common Symbols Used When Creating an ERD



These are both symbols for Entity. Remember, an Entity is just an object or concept you want to store information about. The single line entity box is primarily the type of entity we will use in this class; however, it is good to know there is another type of Entity, which is represented by the double-lined box - a Weak Entity. This is an Entity that must be defined by a foreign key relationship with another Entity and cannot be defined with a Primary Key or it's own attributes alone.

An example - kind of silly but it works - let's say you made a database to store information about employees and their cars so you know the types of cars that should be in the parking lot. You can have the Employee Entity (single line box) and the Car Entity which we are making a weak entity (double line box) because the only information we will have about the car is the color, make, and model. See none of that information about the car is specific enough to reference only one car. Now, on the other hand, if we had the VIN number of each employees car, we would be able to make it a regular entity instead of weak because it would have a unique identifier or Primary Key.



This is the symbol for relationship, which shows how two entities share information. Example: Two entities Student and School, the relationship might be ATTEND. The **Student** (Entity) **ATTENDS** (the relationship) the **School** (Entity).



Attributes describe entities.

Example: Person (Entity) Name (attribute), Age (attribute), Address (attribute)

<https://www.w3schools.in/dbms/database-normalization/>