**What is ER Diagram?**

**An Entity Relationship Diagram (ER Diagram) is a pictorial representation that explains the relationship between entities to be stored in a database.**

**ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.**

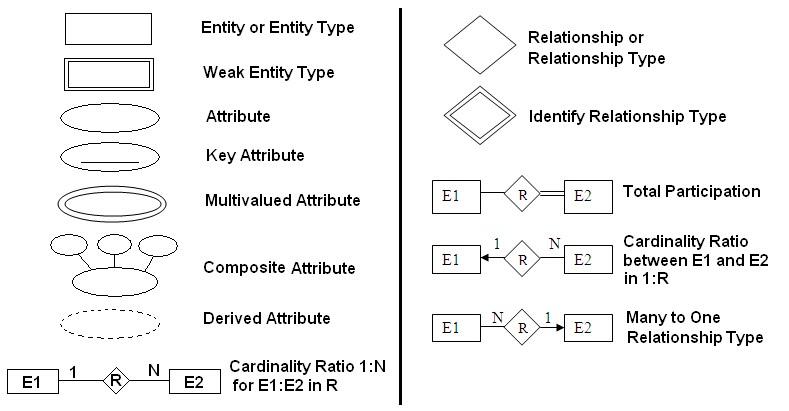
**At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique.**

**The overall logical structure of a database can be expressed graphically by an E-R diagram.**

**The purpose of ER Diagram is to represent the entity framework infrastructure.**

**ER Diagrams consist of different symbols that uses for**

1. **Rectangles to represent entity set**
2. **Double rectangle is used to show weak entity set**
3. **Ellipse/ovals to define attributes**
4. **Diamond shapes to represent relationships.**
5. **Double Diamond to represent weak relationships.**
6. **Double ellipse to show the multivalued attribute.**
7. **Dashed ellipse to show derived attribute.**
8. **Lines denote the link or connection**



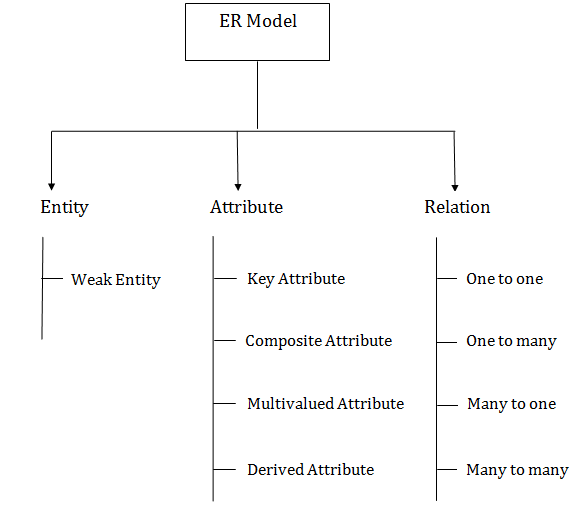
## History of ER models

**ER diagrams are visual tools that are helpful to represent the ER model. Peter Chen proposed ER Diagram in 1971 to create a uniform convention that can be used for relational databases and networks. He aimed to use an ER model as a conceptual modelling approach.**

## Why use ER Diagrams?

* **Helps you to define terms related to entity relationship modelling**
* **Provide a preview of how all your tables should connect, what fields are going to be on each table**
* **Helps to describe entities, attributes, relationships**
* **ER diagrams are translatable into relational tables which allows you to build databases quickly**
* **The database designer gains a better understanding of the information to be contained in the database with the help of ERP diagram**
* **ERD Diagram allows you to communicate with the logical structure of the database to users**

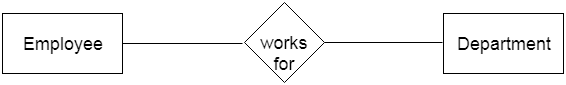
## Component of ER Diagram



### **Entity:**

**An entity may be any object, class, person or place. In the ER diagram, an entity can be represented as rectangles.**

**Consider an organization as an example- manager, product, employee, department etc. can be taken as an entity.**

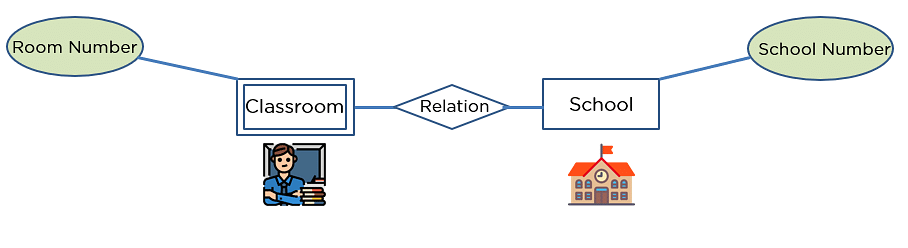


1. **Weak Entity**

**An entity that depends on another entity called a weak entity. The weak entity doesn't contain any key attribute of its own. The weak entity is represented by a double rectangle.**

**The weak entity can be represent as a double rectangle in ER Diagram.**

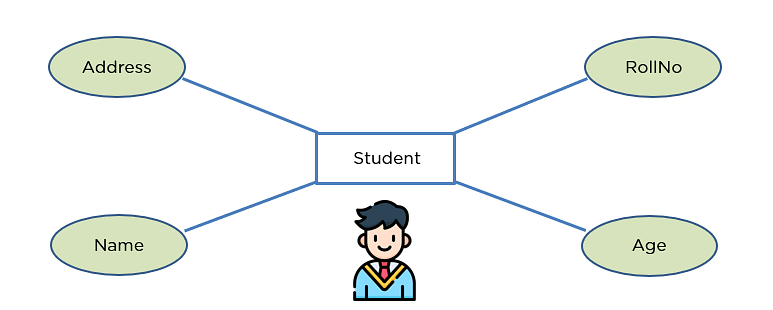
**In the example below, school is a strong entity because it has a primary key attribute - school number. Unlike school, the classroom is a weak entity because it does not have any primary key and the room number here acts only as a discriminator.**



### **2. Attribute**

**The attribute is used to describe the property of an entity. Eclipse is used to represent an attribute.**

**You can illustrate an attribute with an oval shape in an ER diagram.**

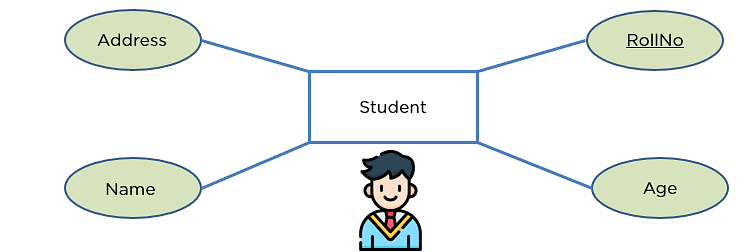


#### **Key Attribute**

**Key attribute uniquely identifies an entity from an entity set.**

**It underlines the text of a key attribute.**

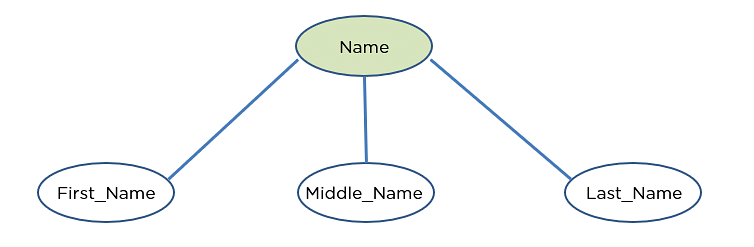
**For example: For a student entity, the roll number can uniquely identify a student from a set of students.**



#### **Composite Attribute**

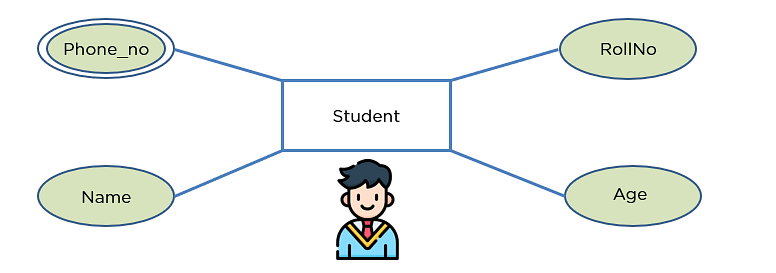
**An attribute that is composed of several other attributes is known as a composite attribute.**

**An oval represents the composite attribute, and the composite attribute oval is further connected with other ovals.**



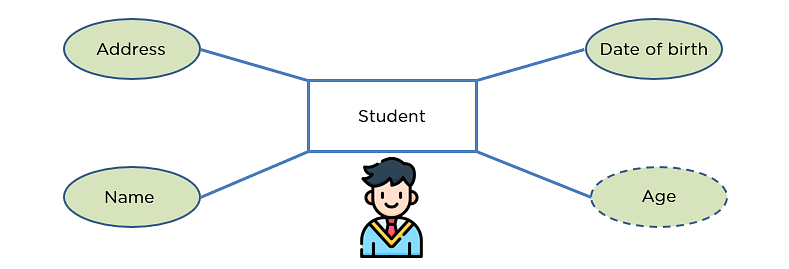
#### **Multivalued Attribute**

**Multi valued attributes can have more than one value at a given instance of time. For example a person can have multiple Phone Numbers,  E-mail  Ids, Addresses, Qualifications, Job Experiences. Multi valued attributes are represented with double outlined ovals in ER diagrams.**



#### **Derived Attribute**

**The attributes whose values can change over time and can be derived from the stored attributes are called Derived Attributes. For example age attribute of a person changes from time to time.  So we can keep this attribute in database as derived attribute.  It can be derived from Date of Birth attribute and current date. You may have seen many times while filling an online form that on filling Date of Birth, Age is automatically calculated. Derived attribute is represented by dashed or dotted outline ovals in the ER diagram.**

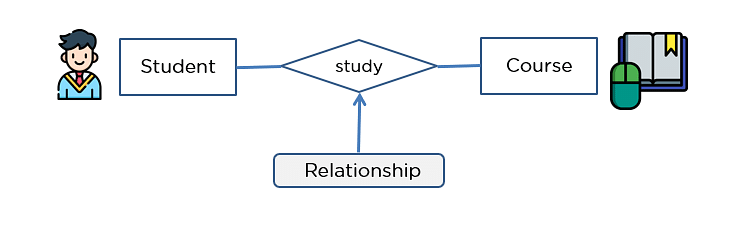


### **Relationship**

**The diamond shape showcases a relationship in the ER diagram.**

**It depicts the relationship between two entities.**

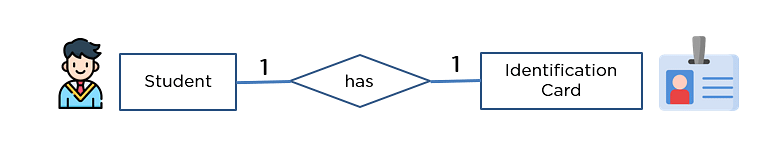
**In the example below, both the student and the course are entities, and study is the relationship between them.**



#### **One-to-One Relationship**

**When a single element of an entity is associated with a single element of another entity, it is called a one-to-one relationship.**

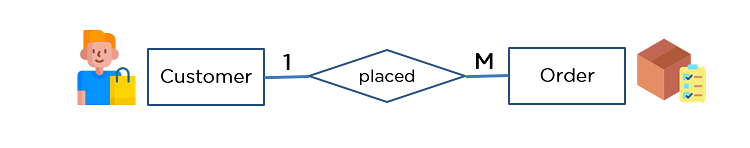
**For example, a student has only one identification card and an identification card is given to one person.**



#### **One-to-Many Relationship**

**When a single element of an entity is associated with more than one element of another entity, it is called a one-to-many relationship**

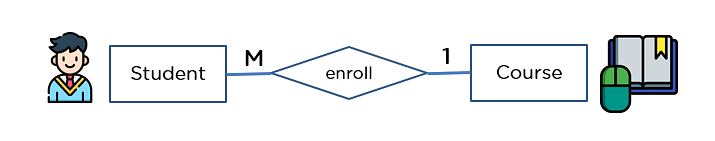
**For example, a customer can place many orders, but an order cannot be placed by many customers.**



#### **Many-to-One Relationship**

**When more than one element of an entity is related to a single element of another entity, then it is called a many-to-one relationship.**

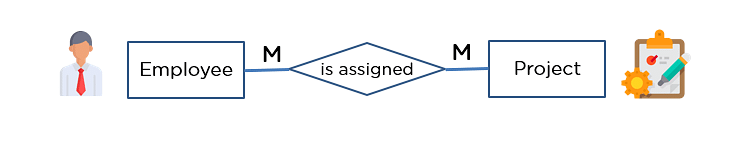
**For example, students have to opt for a single course, but a course can have many** students.



#### **Many-to-Many Relationship**

**When more than one element of an entity is associated with more than one element of another entity, this is called a many-to-many relationship.**

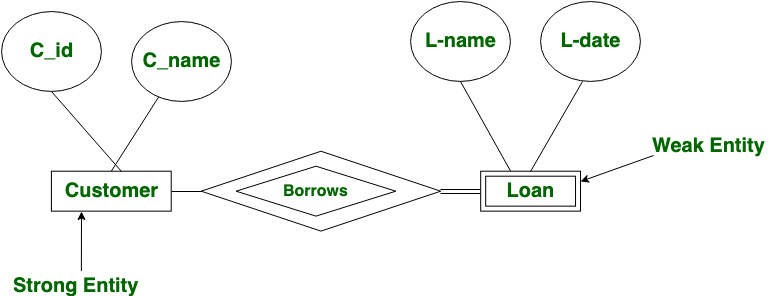
**For example, you can assign an employee to many projects and a project can have many employees.**



### **Comparison Chart**

|  | **Strong Entity** | **Weak Entity** |
| --- | --- | --- |
| 1 | **Strong entity has a primary key.** | **Weak entity has a partial key.** |
| 2 | **Strong entity is independent** | **Weak entity is dependent on a strong entity** |
| 3 | **Strong entity indicated by a single rectangle.** | **Weak entity indicated by a double rectangle.** |
| 4 | **Two strong entity’s relationship is indicated by a single diamond.** | **One strong and one weak entity is indicated by a double diamond.** |
| 5 | **Strong entity may be or may not be participate relationships.** | **Weak entity always participates relationships.** |
| 6 | **In strong entity connecting line is a single line** | **In weak entity connecting line is a double line** |

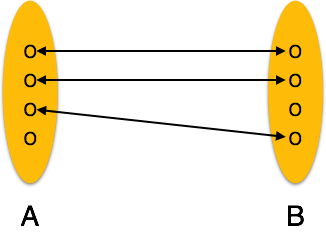
**Example:-**



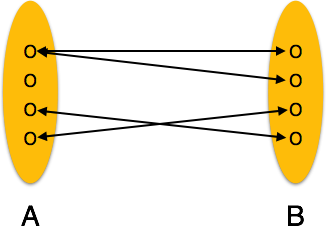
### **Mapping Cardinalities/constraints**

**Cardinality defines the number of entities in one entity set, which can be associated with the number of entities of other set via relationship set.**

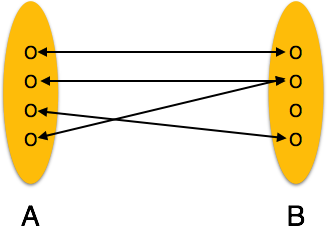
* **One-to-one − One entity from entity set A can be associated with at most one entity of entity set B and vice versa.**



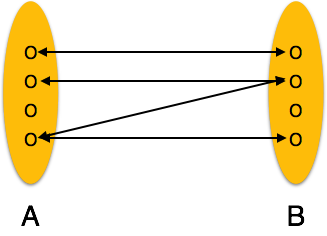
* **One-to-many − One entity from entity set A can be associated with more than one entities of entity set B however an entity from entity set B, can be associated with at most one entity.**



* **Many-to-one − More than one entities from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A.**

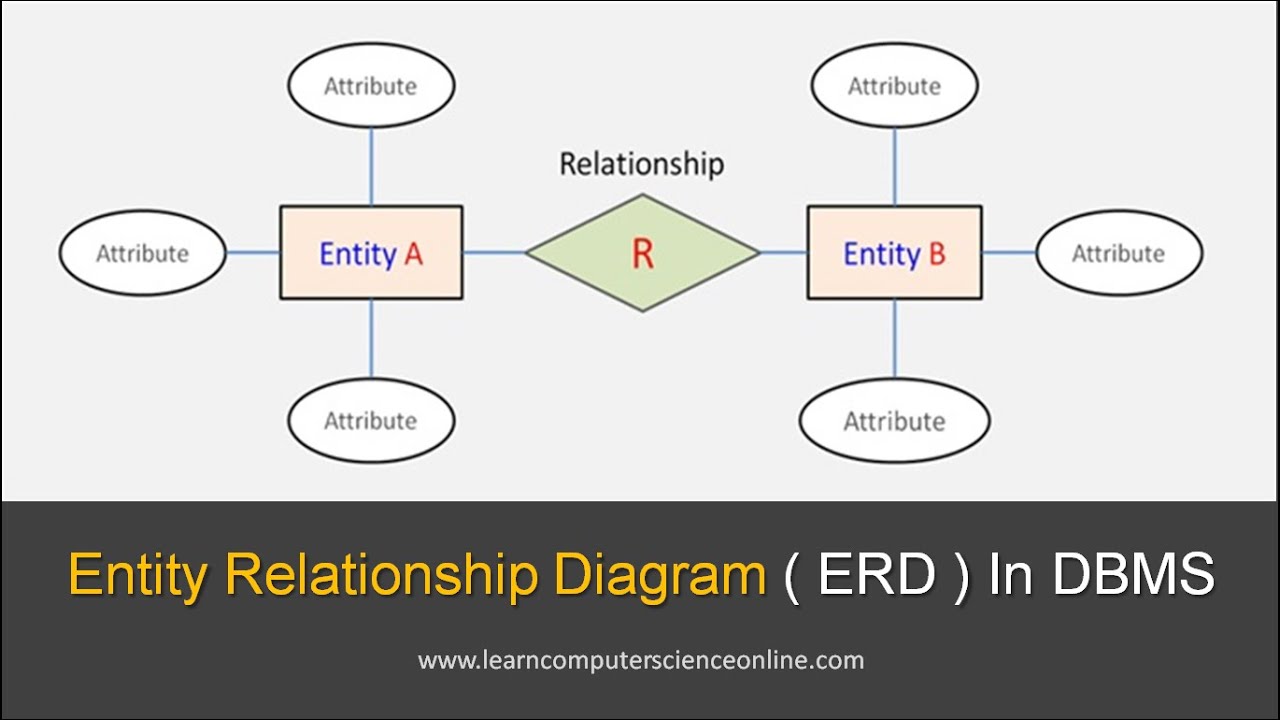


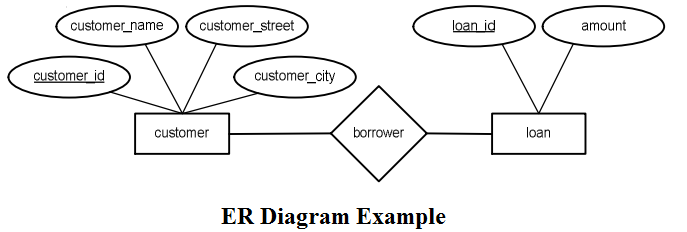
* **Many-to-many − One entity from A can be associated with more than one entity from B and vice versa.**

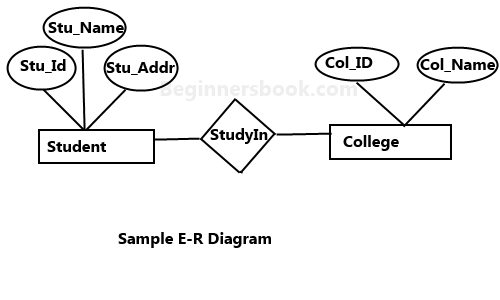


**The main difference between relation and relationship in DBMS is that relation refers to a table in a relational database while relationship refers to how two tables are connected together in a relational model based database. A database is a collection of related data.**

**Example of E-R Diagram**

1. 





## Specialization

**Specialization is a top-down approach in which a higher-level entity is divided into multiple specialized lower-level entities. Then this process is known as Specialization. Specialization is usually used to find subsets of an entity that has a few different or additional attributes.**

**Steps for Implementing the Specialization**

**If any user wants to implement the specialization concept for breaking the entity, then we have to follow the following steps one by one:**

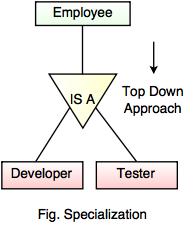
**Step 1:** **The first step for each user is to identify the entity for breaking it into two or more entities.**

**Step 2:** **After that, the user has to examine the attributes of the identified higher-level entity**

**Step 3:** **Now, the user has to define the subclasses or lower-level entities, which consist of the attributes of the higher-level entity. The lower-level entities are termed as specialized entities.**

**Entity set - employee**

**Employee can be divided into 2 entity set- Developer and Tester**



**Generalization**

**When several entity sets are grouped together to form one entity set, then this process is known as Generalization.**

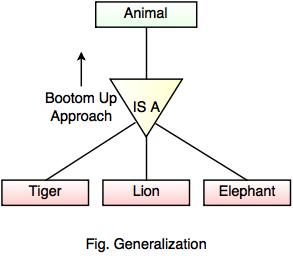
**e.g. if in the above example we grouped the entity set Developer and Tester entity set to form the entity set Employee.**

**Developer**

**+ =**

**Employee**

**Tester**



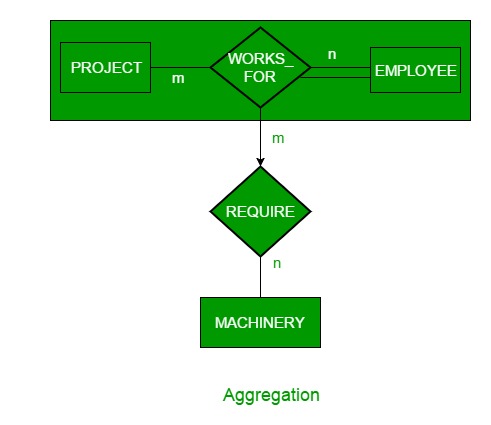
**Difference Between Specialization and Generalization**

|  |  |
| --- | --- |
| **Specialization** | **Generalization** |
| **1. This concept of a database system works in a top-down manner.** | **1. This concept of a database system works in a bottom-up manner.** |
| **2. In the Specialization concept, schema size gets increased.** | **2. In the generalization concept, schema size gets reduced.** |
| **3. This technique is applied to the single higher-level entity.** | **3. This technique is applied to the multiple lower-level entities.** |
| **4. Inheritance process occurs in this mechanism because the properties of the superclass are shared with the subclasses.** | **4. Inheritance process does not occur in this mechanism.** |
| **5. In this technique, the higher-level entity of the ER diagram may not have the entities of the lower level.** | **5. The higher-level entity in this technique must have lower-level entities.** |
| **6. This mechanism splits the higher-level entity and form new entities with common properties.** | **6. This mechanism takes the common features of multiple lower-level entities and forms a new higher entity.** |

**Aggregation**

**Aggregation :- Aggregation is the abstraction of the relationship of the low level entities. One relationship between two entity set can be treated as one entity set and name of the such relation.**

**Aggregation in DBMS (Database Management System) is a process of combining two or more entities to form a more meaningful new entity. This Aggregation process is done when the entities don't make sense on their own without applying the aggregation process**



**Here we see that, it is difficult to draw the E-R Diagram of three entity set. Therefore here we will use the concept of aggregate the two entity set( Employee , project) and then we will draw the E-R Diagram as follows**

**So we see here how aggregation is performed.**

**There are two entity set included in work entity set.**

**Work**

**Uses**

**Machine**

**Fig -Aggregation**