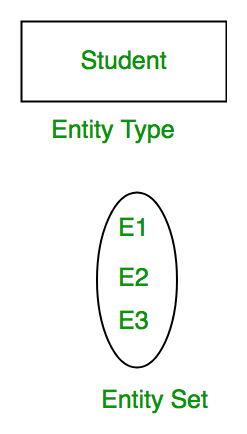
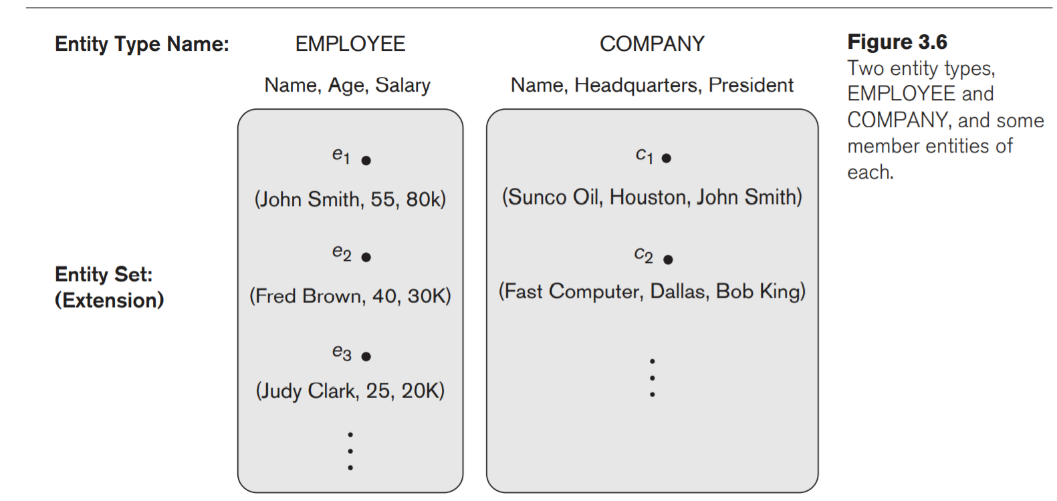
**Entity Types, Entity Sets**

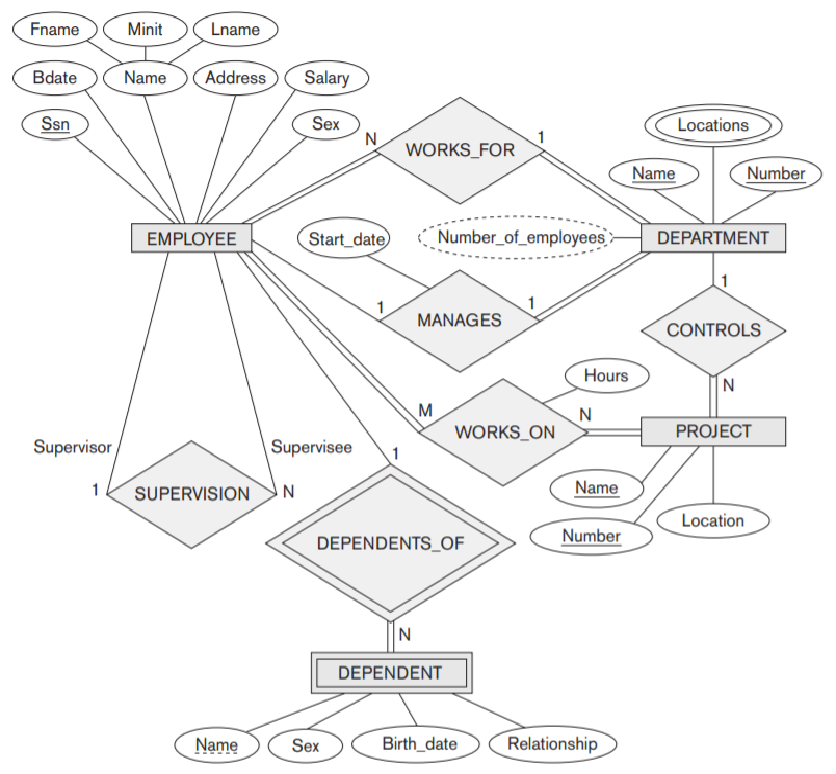
An Entity is an object of Entity Type and set of all entities is called as entity set. e.g.; E1 is an entity having Entity Type Student and set of all students is called Entity Set.

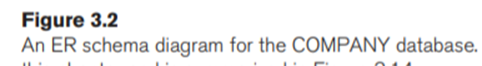


An entity type defines a collection (or set) of entities that have the same attributes. Each entity type in the database is described by its name and attributes. Figure 3.6 shows two entity types: EMPLOYEE and COMPANY, and a list of some of the attributes for each.



An entity type is represented in ER diagrams (see Figure 3.2) as a rectangular box enclosing the entity type name. Attribute names are enclosed in ovals and are attached to their entity type by straight lines. Composite attributes are attached to their component attributes by straight lines. Multivalued attributes are displayed in double ovals.





The collection of entities of a particular entity type is grouped into an entity set, which is also called the extension of the entity type.

**Key Attributes of an Entity Type.**

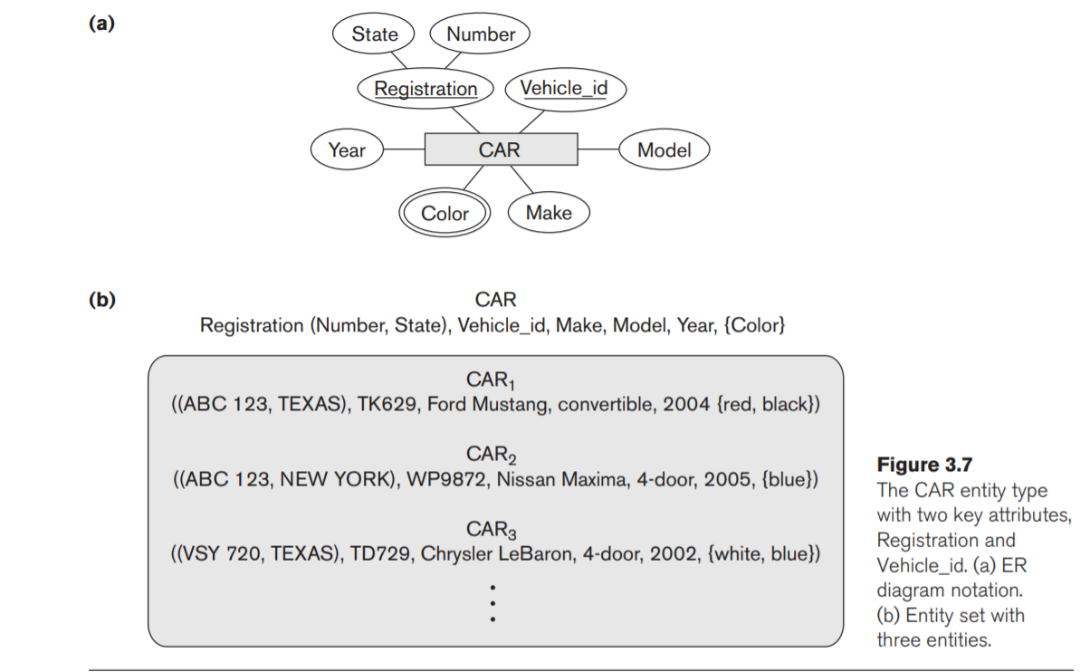
An important constraint on the entities of an entity type is the key or uniqueness constraint on attributes.

Key is an attribute or collection of attributes that uniquely identifies an entity among entity set.

For example, the roll\_number of a student makes him/her identifiable among students.

An entity type usually has one or more attributes whose values are distinct for each individual entity in the entity set. Such an attribute is called a key attribute, and its values can be used to identify each entity uniquely.

For example, the Name attribute is a key of the COMPANY entity type in Figure 3.6 because no two companies are allowed to have the same name. In ER diagrammatic notation, each key attribute has its name underlined inside the oval.



Some entity types have more than one key attribute. For example, each of the Vehicle\_id and Registration attributes of the entity type CAR (Figure 3.7) is a key in its own right. The Registration attribute is an example of a composite key formed from two simple component attributes, State and Number.

An entity type may also have no key, in which case it is called a weak entity type.

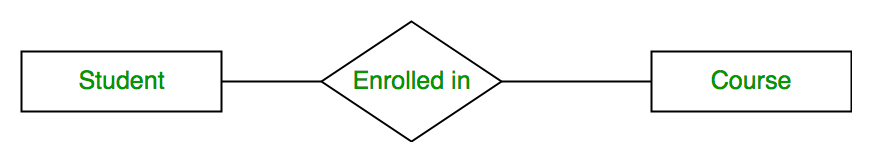
Relationship

The association among entities is called a relationship. For example, an employee works\_at a department, a student enrolls in a course. Here, Works\_at and Enrolls are called relationships.

Relationship Type and Relationship Set:



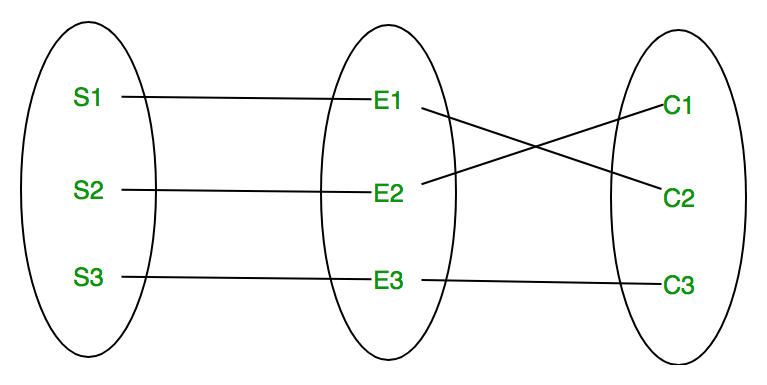
A relationship type represents the association between entity types. For example,‘Enrolled in’ is a relationship type that exists between entity type Student and Course. In ER diagram, relationship type is represented by a diamond and connecting the entities with lines.





A set of relationships of same type is known as relationship set. The following relationship set depicts S1 is enrolled in C2, S2 is enrolled in C1 and S3 is enrolled in C3.





Degree of Relationship



The number of participating entities in a relationship defines the degree of the relationship.

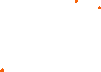
Unary= degree 1



Binary = degree 2



Ternary = degree 3

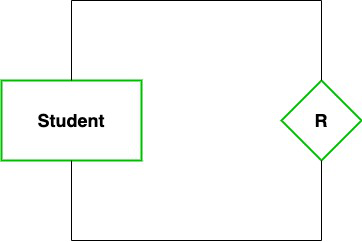


n-ary = degree n

**Unary**

In this type of relationship, both the associating entity type are the same. So, we can say that unary relationships exist when both entity types are the same and we call them the degree of relationship is 1. Or in other words, in a relation only one entity set is participating then such type of relationship is known as a unary relationship.

**Example:** In a particular class, we have many students, there are monitors too. So, here class monitors are also students. Thus, we can say that only students are participating here. So, the degree of such type of relationship is 1.



**Binary (degree 2)**

In a Binary relationship, there are two types of entity associates. So, we can say that a Binary relationship exists when there are two types of entity and we call them a degree of relationship is 2. Or in other words, in a relation when two entity sets are participating then such type of relationship is known as a binary relationship. This is the most used relationship and one can easily be converted into a relational table.

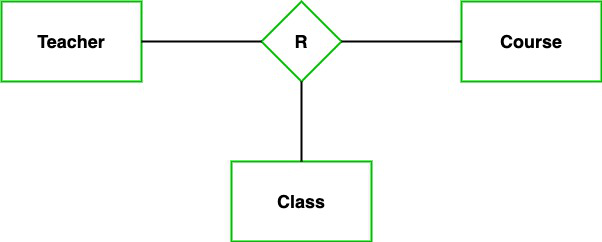
**Example:**We have two entity types ‘Student’ and ‘ID’ where each ‘Student’ has his ‘ID’. So, here two entity types are associating we can say it is a binary relationship. Also, one ‘Student’ can have many ‘daughters’ but each ‘daughter’ should belong to only one ‘father. We can say that it is a one-to-many binary relationship.



**Ternary(degree 3)**

In the Ternary relationship, there are three types of entity associates. So, we can say that a Ternary relationship exists when there are three types of entity and we call them a degree of relationship is 3. Since the number of entities increases due to this, it becomes very complex to turn E-R into a relational table. Now let’s understand with the examples.

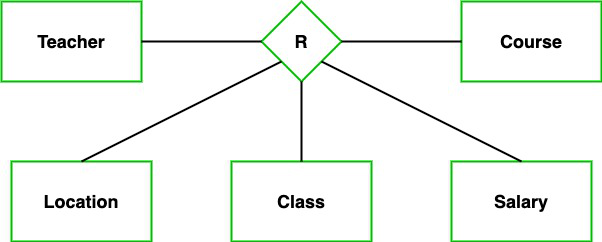
**Example:** We have three entity types ‘Teacher’, ‘Course’, and ‘Class’. The relationship between these entities is defined as the teacher teaching a particular course, also the teacher teaches a particular class. So, here three entity types are associating we can say it is a ternary relationship.



**N-ary (n degree)**

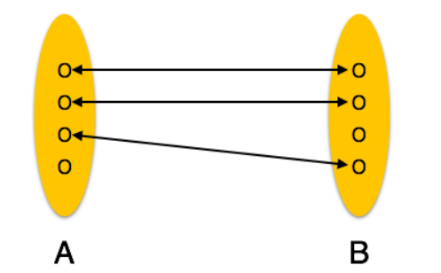
In the N-ary relationship, there are n types of entity that associates. So, we can say that an N-ary relationship exists when there are n types of entities. There is one limitation of the N-ary relationship, as there are many entities so it is very hard to convert into an entity, rational table. So, this is very uncommon, unlike binary which is very much popular.

**Example:**We have 5 entities Teacher, Class, Location, Salary, Course. So, here five entity types are associating we can say an n-ary relationship is 5.



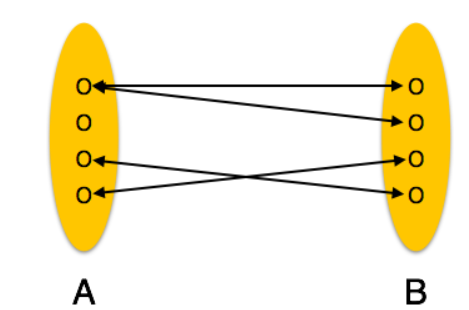
**Relationship Types**

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



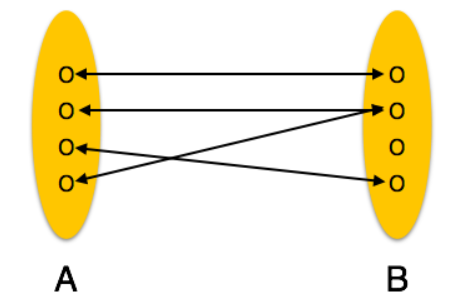


One-to-many − One entity from entity set A can be associated with more than one entity of entity set B however an entity from entity set B, can be associated with at most one entity.



Many-to-one − More than one entity 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.



