Hibernate Mapping

Hibernate Mapping

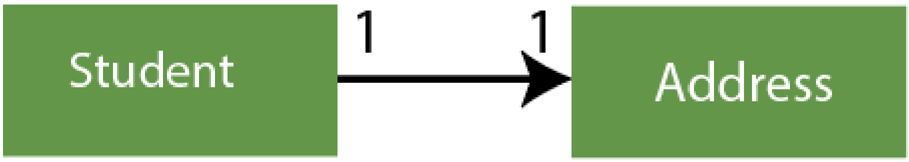
Association- It means that two or more things are related to each other by some specific relation. In other words, it specifies how the objects are associated with each other.

Hibernate mapping is one of the essential features of Hibernate. In Hibernate, it tells about how the objects of persistent classes are associated with each other. Hibernate provides four types of association mapping, which are listed below:

- One To One
- One To Many
- Many To One
- Many To Many

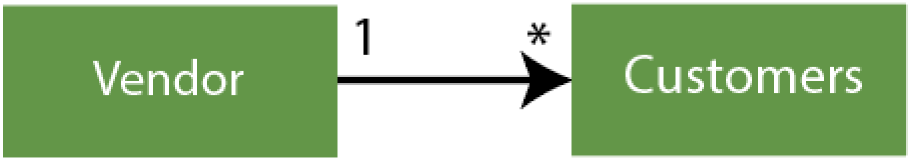
The types of association mapping can be uni-directional or bi-directional.

One To One



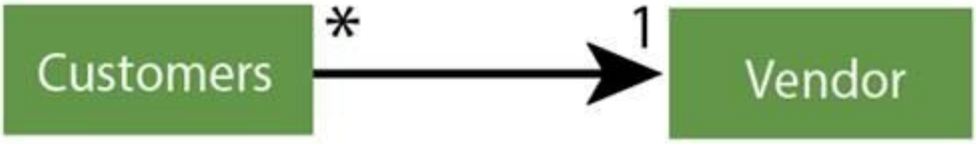
Two objects are said to be in a One-to-One relationship when one object has only one occurrence in the other object. In other words, only one object of a persistent class is related to only one object of another persistent class. To achieve a One-to-One relationship between two objects, the primary key value of a parent object and the child object must be same.

One To Many



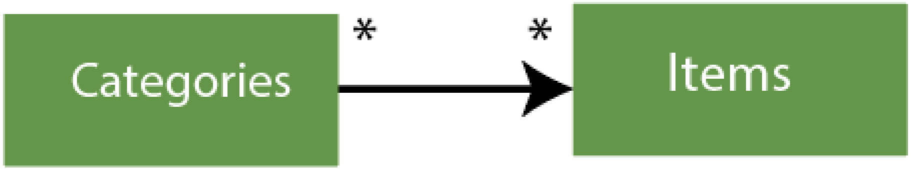
In One-to-Many association mapping, only one object of one persistent class is related to more than one objects of another persistent class. It is a relationship in which one (**parent**) is related to many (**children**). For example, *clothing* is a parent class of child classes such as *men clothing*, *women clothing,* and *kids clothing.* It is a 1 to N relationship. To achieve a One-to-Many relationship between two objects, we need to take a collection property such as Set, Map, and List in the parent class.

Many To One



A Many-to-One association mapping is the reverse of One-to-Many association mapping. For example, many (**vehicles**)are associated with one (**person**), i.e., a person may have more than one vehicle. It is one of the most commonly used association mapping. In Hibernate, Many-to-One association mapping is applied from child class object to parent class object. It is an N to 1 relationship.

Many To Many



In Many-to-Many association mapping, many objects of one persistent class are associated with many objects of another persistent class. For example, many (**students**) are related to many (**courses**) and vice-versa as many-to-many is a bidirectional association mapping. To achieve many-to-many mapping, we need to declare collection property in both persistent classes.