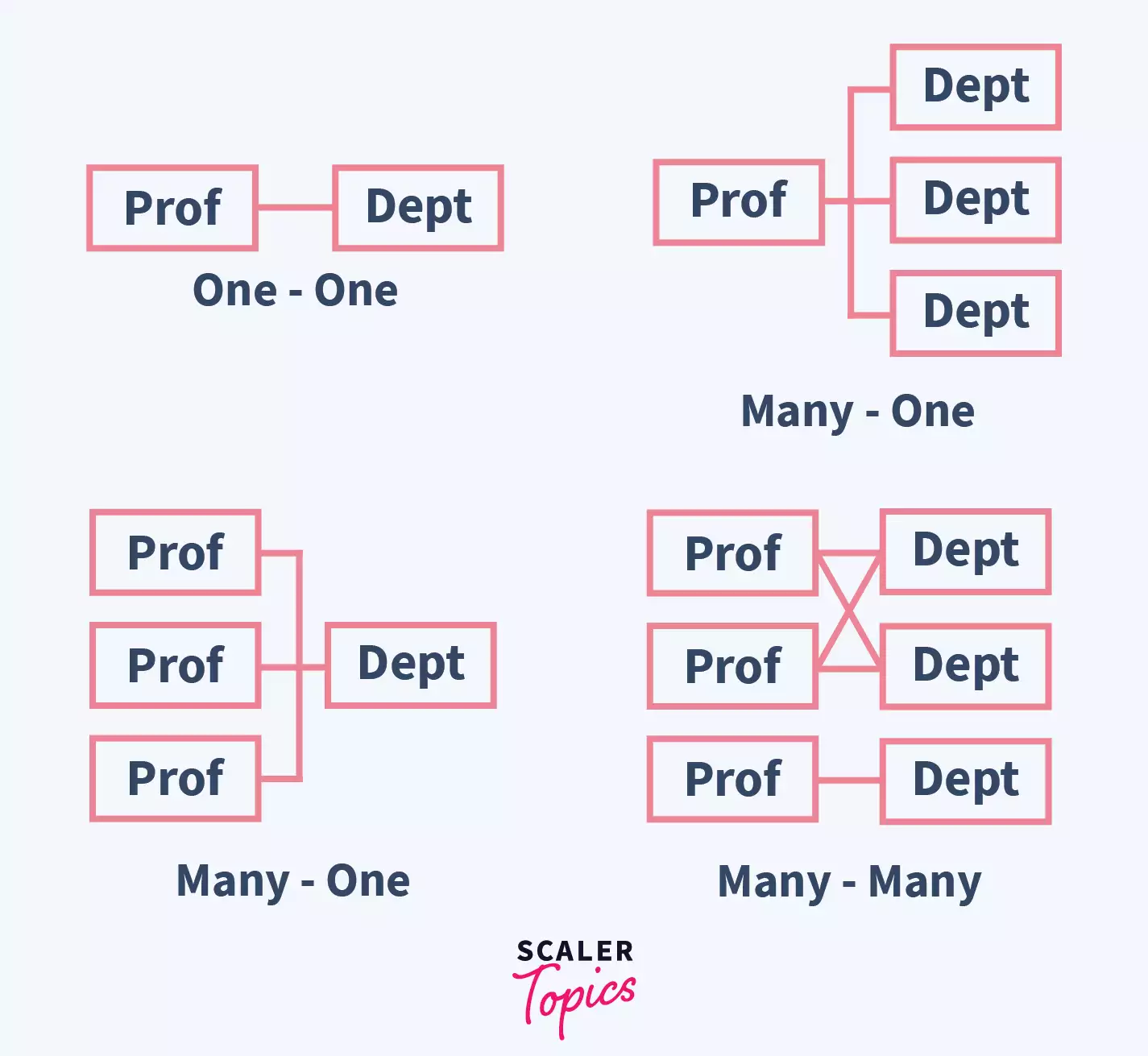
**What is the Association?**

It refers to the relationship between any two entities. Association in java is the relationship that can be established between any two classes. These relationships can be of four types:

1. One-to-One relation
2. One-to-many relation
3. Many-to-one relation
4. Many-to-many relation

Below are the type of relationships/associations that can be possible between them

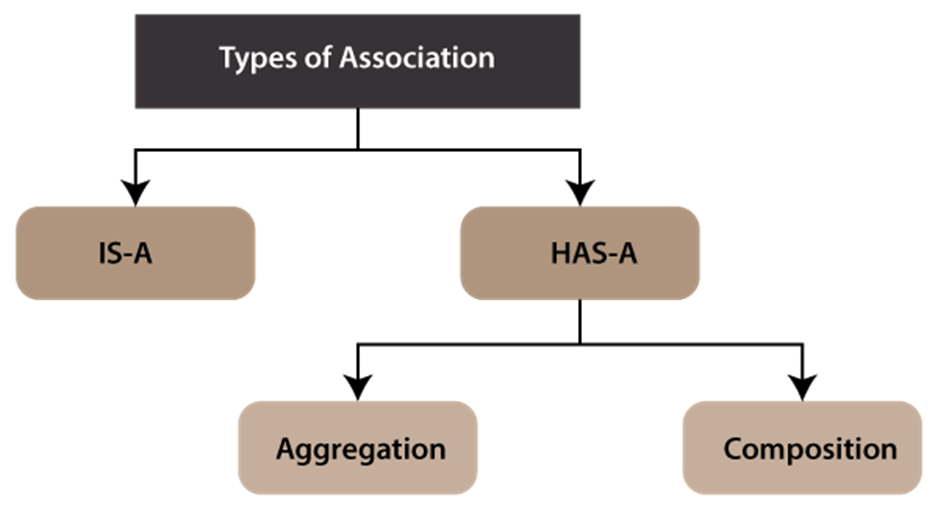
* One professor can only be assigned to work in one department. This forms a **one-to-one** association between the two classes.
* One professor can be assigned to work in multiple departments. This is a **one-to-many** association between the two classes.
* Multiple professors can be assigned to work in one department. This forms a **many-to-one** association between the two classes.
* Multiple professors can be assigned to work in multiple departments. This is the **many-to-many** association between the two classes.
* These associations in java help the objects of one class to communicate with the objects of the other class.



## **Types of Association**

In Java, two types of **Association** are possible:

1. IS-A Association
2. HAS-A Association
   1. Aggregation
   2. Composition



The **HAS-A Association** is further classified into two parts, i.e., Aggregation and Composition.

### **1) Aggregation**

Aggregation is a particular type of Association. It represents the has-a relationship between the two classes. The relationship between the two classes is entirely independent, which means if one of the objects of the class gets deleted, it won’t affect the other.

Let’s take an easy example. A student attends a school. After the completion of his studies, he can quickly leave school. That means the end of the student object would not destroy the School object. We can say that the Student ‘has-a’ relationship with the school.

Eg: AggregationExample

// go through the code

Composition :

* The composition is a design technique in java to implement a **has-a** relationship.
* Java [Inheritance](https://www.geeksforgeeks.org/inheritance-in-java/) is used for code reuse purposes and the same we can do by using composition.
* The composition is achieved by using an instance variable that refers to other objects. If an object contains the other object and the contained object cannot exist without the existence of that object, then it is called composition.
* In more specific words composition is a way of describing reference between two or more classes using instance variable and an instance should be created before it is used.

The benefitsof using Composition is as follows:

1. Composition allows the reuse of code.
2. Java doesn’t support [multiple inheritances](https://www.geeksforgeeks.org/java-and-multiple-inheritance/) but by using composition we can achieve it.

Eg: CompositionExample

Eg: CompositionExample\_2