**Generalization, Specialization and Aggregation in ER Model**

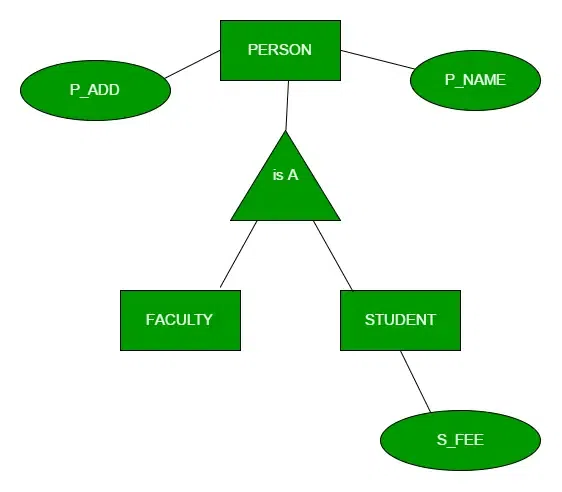
Using the ER model for bigger data creates a lot of complexity while designing a database model, So in order to minimize the complexity Generalization, Specialization, and Aggregation were introduced in the ER model. These were used for data abstraction. In which an abstraction mechanism is used to hide details of a set of objects. In this article we will cover the concept of Generalization, Specialization, and Aggregation with example.

**Generalization**

Generalization is the process of extracting common properties from a set of entities and creating a generalized entity from it. It is a bottom-up approach in which two or more entities can be generalized to a higher-level entity if they have some attributes in common. For Example, STUDENT and FACULTY can be

generalized to a higher-level entity called PERSON as shown in Figure 1. In this case, common attributes like P\_NAME, and P\_ADD become part of a higher [entity](https://www.geeksforgeeks.org/difference-between-entity-and-object/) (PERSON), and specialized [attributes](https://www.geeksforgeeks.org/types-of-attributes-in-er-model/) like S\_FEE become part of a specialized entity (STUDENT).

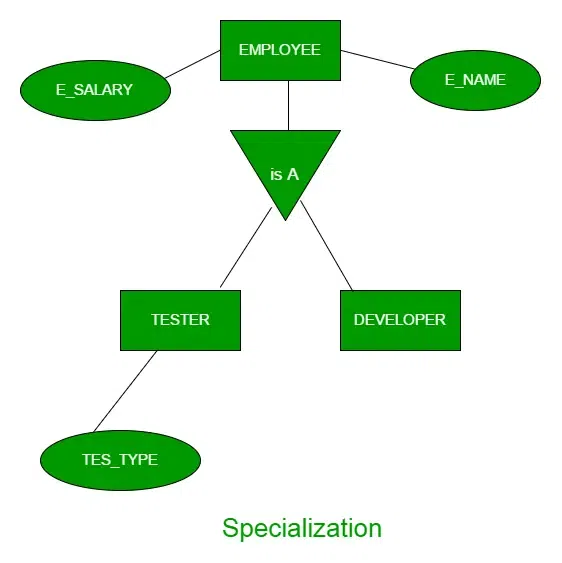
Generalization is also called as ‘ Bottom-up approach”.



**Specialization**

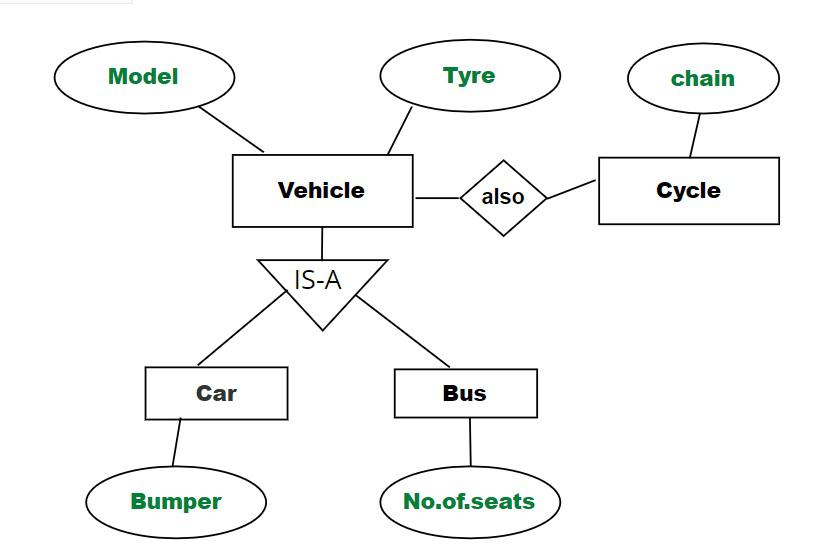
In specialization, an entity is divided into sub-entities based on its characteristics. It is a top-down approach where the higher-level entity is specialized into two or more lower-level [entities](https://www.geeksforgeeks.org/difference-between-entity-entity-set-and-entity-type/). For Example, an EMPLOYEE entity in an Employee management system can be specialized into DEVELOPER, TESTER, etc. as shown in Figure 2. In this case, common attributes like E\_NAME, E\_SAL, etc. become part of a higher entity (EMPLOYEE), and specialized attributes like TES\_TYPE become part of a specialized entity (TESTER).

Specialization is also called as ” Top-Down approch”.



**Inheritance:** It is an important feature of generalization and specialization

* **Attribute inheritance** : It allows lower level entities to inherit the attributes of higher level entities and vice versa. In diagram **Car**entity is an inheritance of **Vehicle**entity ,So Car can acquire attributes of **Vehicle.**Example:car can acquire **Model**attribute of **Vehicle.**
* **Participation inheritance:** Participation inheritance in ER modeling refers to the inheritance of participation constraints from a higher-level entity (superclass) to a lower-level entity (subclass). It ensures that subclasses adhere to the same participation rules in relationships, although attributes and relationships themselves are inherited differently. In diagram Vehicle entity has an relationship with Cycle entity, but it would not automatically acquire the relationship itself with the Vehicle entity. Participation inheritance only refers to the inheritance of participation constraints, not the actual relationships between entities.



**Aggregation**

An ER diagram is not capable of representing the relationship between an entity and a relationship which may be required in some scenarios. In those cases, a relationship with its corresponding entities is aggregated into a higher-level entity. Aggregation is an abstraction through which we can represent relationships as higher-level entity sets.

For Example, an Employee working on a project may require some machinery. So, REQUIRE relationship is needed between the relationship WORKS\_FOR and entity MACHINERY. Using aggregation, WORKS\_FOR relationship with its entities EMPLOYEE and PROJECT is aggregated into a single entity and relationship REQUIRE is created between the aggregated entity and MACHINERY.

