The Entity Relational Model is a model for identifying entities to be represented in the database and representation of how those entities are related. The ER data model specifies enterprise schema that represents the overall logical structure of a database graphically.

The Entity Relationship Diagram explains the relationship among the entities present in the database. ER models are used to model real-world objects like a person, a car, or a company and the relation between these real-world objects. In short, the ER Diagram is the structural format of the database.

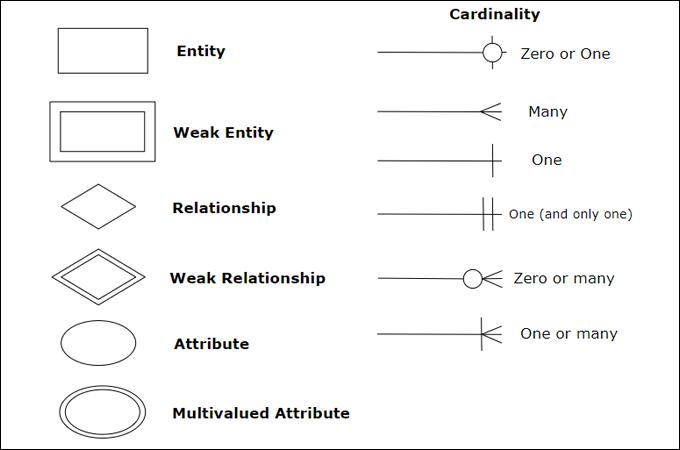
**Why Use ER Diagrams In DBMS?**

* ER diagrams are used to represent the E-R model in a database, which makes them easy to be converted into relations (tables).
* ER diagrams provide the purpose of real-world modeling of objects which makes them intently useful.
* ER diagrams require no technical knowledge and no hardware support.
* These diagrams are very easy to understand and easy to create even for a naive user.
* It gives a standard solution for visualizing the data logically.

**Symbols Used in ER Model**

ER Model is used to model the logical view of the system from a data perspective which consists of these symbols:

* **Rectangles:**Rectangles represent Entities in the ER Model.
* **Ellipses:**Ellipses represent Attributes in the ER Model.
* **Diamond:**Diamonds represent Relationships among Entities.
* **Lines:**Lines represent attributes to entities and entity sets with other relationship types.
* **Double Ellipse:**Double Ellipses represent [Multi-Valued Attributes](https://iotap.geeksforgeeks.org/problems/what-is-the-difference-between-single-valued-and-multi-valued-attributes).
* **Double Rectangle:**Double Rectangle represents a Weak Entity.



*Symbols used in ER Diagram*

**Components of ER Diagram**

ER Model consists of Entities, Attributes, and Relationships among Entities in a Database System.

## Components of ER Diagram

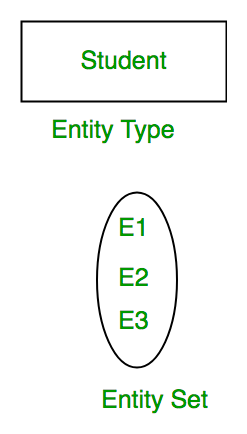
ER Model consists of Entities, Attributes, and Relationships among Entities in a Database System.

*Components of ER Diagram*

### Entity

An Entity may be an object with a physical existence – a particular person, car, house, or employee – or it may be an object with a conceptual existence – a company, a job, or a university course.

**Entity Set:** An Entity is an object of Entity Type and a set of all entities is called an entity set. For Example, E1 is an entity having Entity Type Student and the set of all students is called Entity Set. In ER diagram, Entity Type is represented as:



*Entity Set*

#### 1. Strong Entity

A [Strong Entity](https://www.geeksforgeeks.org/difference-between-strong-and-weak-entity/) is a type of entity that has a key Attribute. Strong Entity does not depend on other Entity in the Schema. It has a primary key, that helps in identifying it uniquely, and it is represented by a rectangle. These are called Strong Entity Types.

#### 2. Weak Entity

An Entity type has a key attribute that uniquely identifies each entity in the entity set. But some entity type exists for which key attributes can’t be defined. These are called [Weak Entity types](https://www.geeksforgeeks.org/weak-entity-set-in-er-diagrams/).

**For Example,** A company may store the information of dependents (Parents, Children, Spouse) of an Employee. But the dependents don’t have existed without the employee. So Dependent will be a **Weak Entity Type** and Employee will be Identifying Entity type for Dependent, which means it is **Strong Entity Type**.

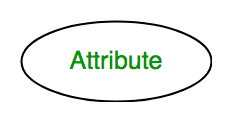
A weak entity type is represented by a Double Rectangle. The participation of weak entity types is always total. The relationship between the weak entity type and its identifying strong entity type is called identifying relationship and it is represented by a double diamond.



*Strong Entity and Weak Entity*

### **Attributes**

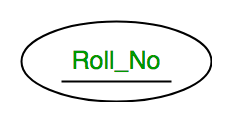
[Attributes](https://www.geeksforgeeks.org/types-of-attributes-in-er-model/) are the properties that define the entity type. For example, Roll\_No, Name, DOB, Age, Address, and Mobile\_No are the attributes that define entity type Student. In ER diagram, the attribute is represented by an oval.



*Attribute*

#### **1. Key Attribute**

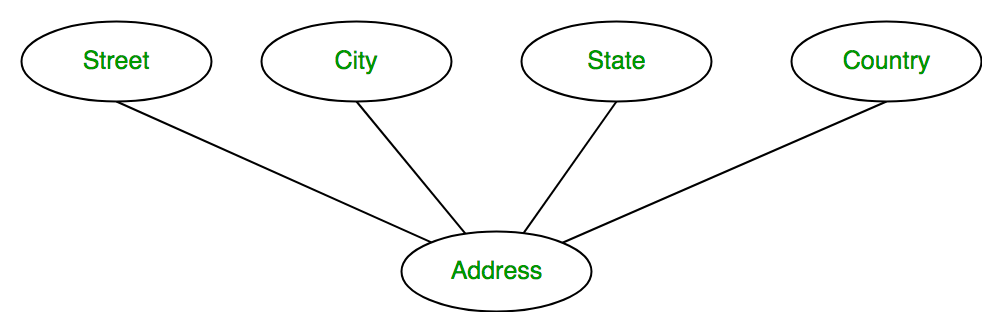
The attribute which **uniquely identifies each entity** in the entity set is called the key attribute. For example, Roll\_No will be unique for each student. In ER diagram, the key attribute is represented by an oval with underlying lines.



*Key Attribute*

#### **2. Composite Attribute**

An attribute **composed of many other attributes** is called a composite attribute. For example, the Address attribute of the student Entity type consists of Street, City, State, and Country. In ER diagram, the composite attribute is represented by an oval comprising of ovals.



*Composite Attribute*

#### **3. Multivalued Attribute**

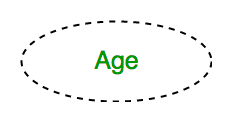
An attribute consisting of more than one value for a given entity. For example, Phone\_No (can be more than one for a given student). In ER diagram, a multivalued attribute is represented by a double oval.



*Multivalued Attribute*

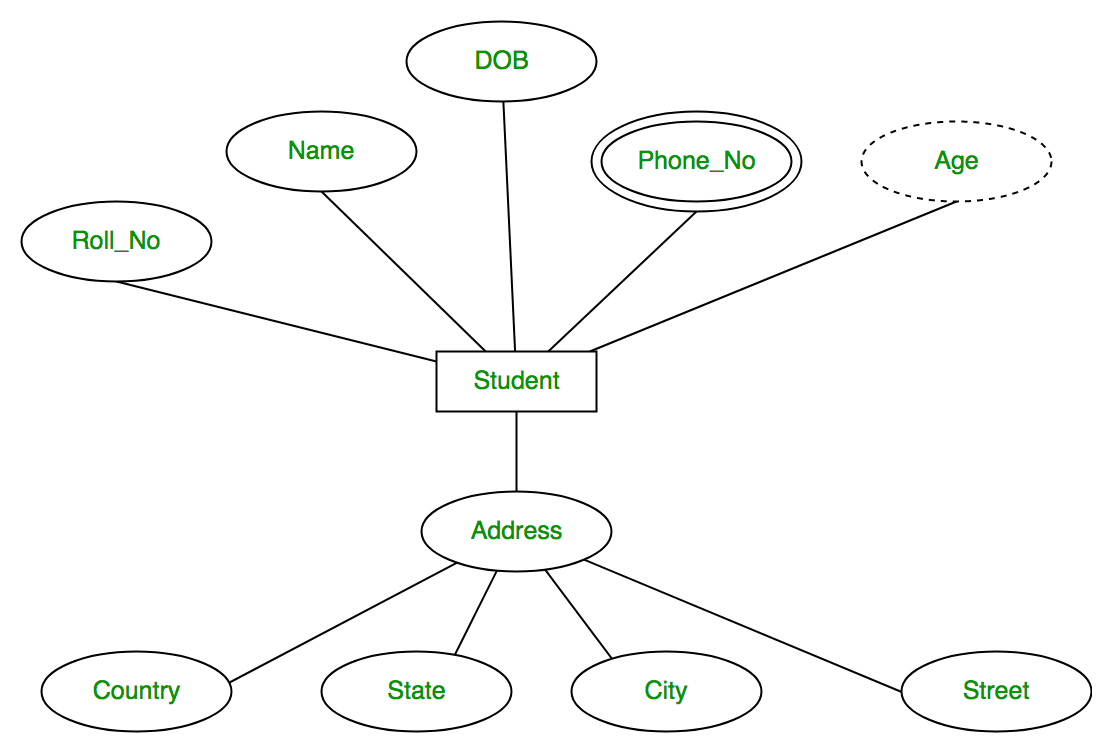
#### **4. Derived Attribute**

An attribute that can be derived from other attributes of the entity type is known as a derived attribute. e.g.; Age (can be derived from DOB). In ER diagram, the derived attribute is represented by a dashed oval.



*Derived Attribute*

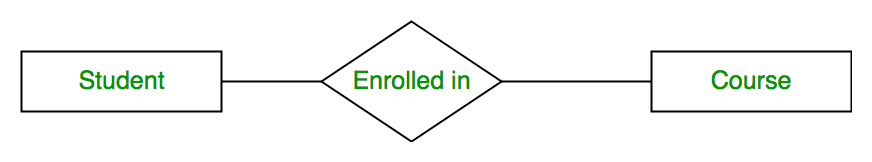
The Complete Entity Type Student with its Attributes can be represented as:



*Entity and Attributes*

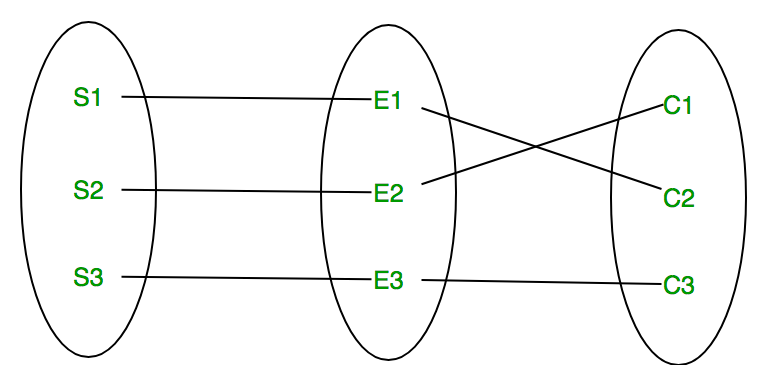
### **Relationship Type and Relationship Set**

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, the relationship type is represented by a diamond and connecting the entities with lines.



*Entity-Relationship Set*

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

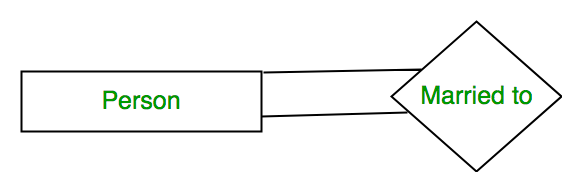


*Relationship Set*

#### **Degree of a Relationship Set**

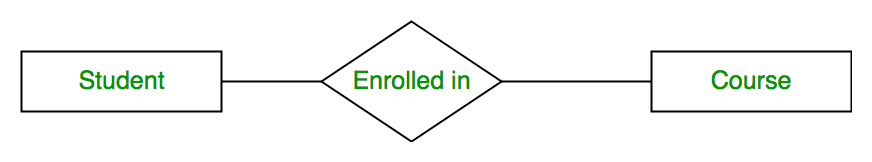
The number of different entity sets participating in a relationship set is called the [degree of a relationship set.](https://www.geeksforgeeks.org/degree-of-relations-in-dbms/)

**1. Unary Relationship:**When there is only ONE entity set participating in a relation, the relationship is called a unary relationship. For example, one person is married to only one person.



*Unary Relationship*

**2. Binary Relationship:**When there are TWO entities set participating in a relationship, the relationship is called a binary relationship. For example, a Student is enrolled in a Course.



*Binary Relationship*

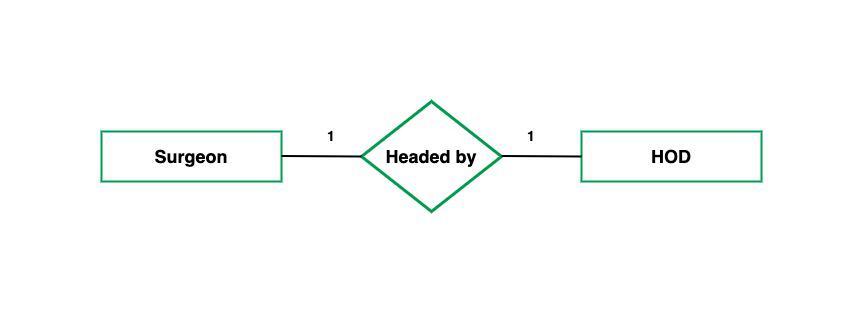
**3. n-ary Relationship:**When there are n entities set participating in a relation, the relationship is called an n-ary relationship.

#### **Cardinality**

The number of times an entity of an entity set participates in a relationship set is known as [cardinality](https://www.geeksforgeeks.org/cardinality-in-dbms/). Cardinality can be of different types:

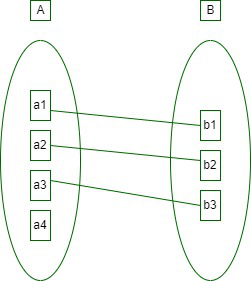
**1. One-to-One:** When each entity in each entity set can take part only once in the relationship, the cardinality is one-to-one. Let us assume that a male can marry one female and a female can marry one male. So the relationship will be one-to-one.

the total number of tables that can be used in this is 2.



*one to one cardinality*

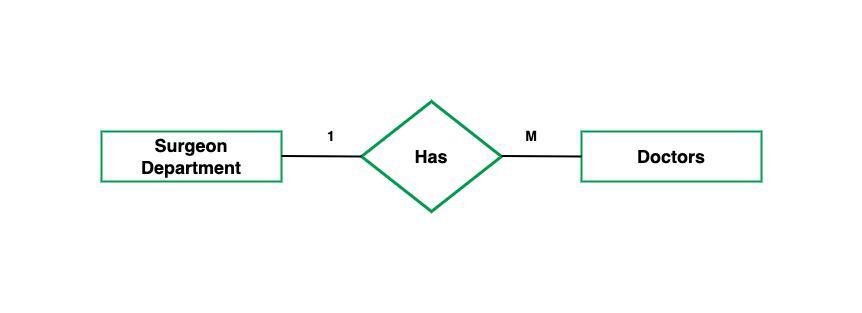
Using Sets, it can be represented as:



*Set Representation of One-to-One*

**2. One-to-Many:** In one-to-many mapping as well where each entity can be related to more than one relationship and the total number of tables that can be used in this is 2. Let us assume that one surgeon deparment can accomodate many doctors. So the Cardinality will be 1 to M. It means one deparment has many Doctors.

total number of tables that can used is 3.



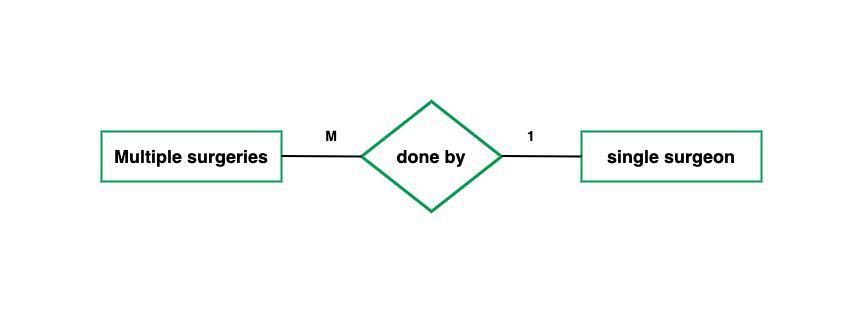
*one to many cardinality*

Using sets, one-to-many cardinality can be represented as:

*Set Representation of One-to-Many*

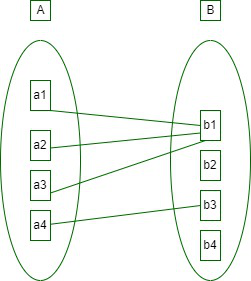
**3. Many-to-One:** When entities in one entity set can take part only once in the relationship set and entities in other entity sets can take part more than once in the relationship set, cardinality is many to one. Let us assume that a student can take only one course but one course can be taken by many students. So the cardinality will be n to 1. It means that for one course there can be n students but for one student, there will be only one course.

The total number of tables that can be used in this is 3.



*many to one cardinality*

Using Sets, it can be represented as:

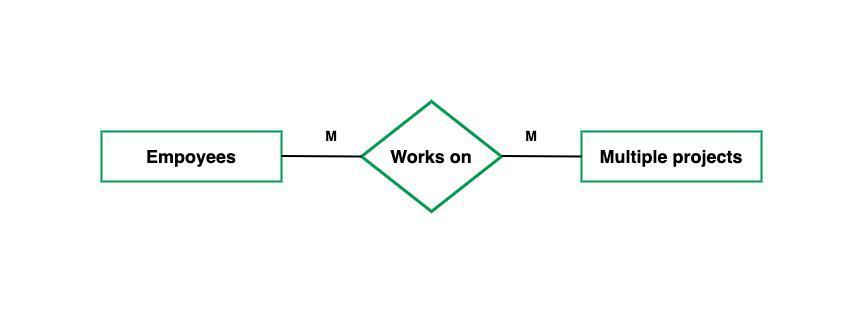


*Set Representation of Many-to-One*

In this case, each student is taking only 1 course but 1 course has been taken by many students.

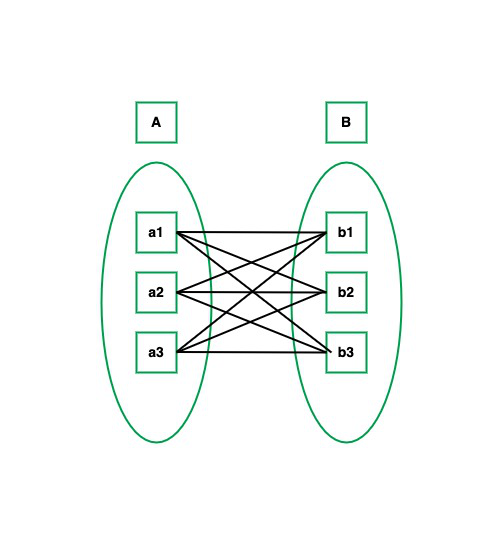
**4. Many-to-Many:**When entities in all entity sets can take part more than once in the relationship cardinality is many to many. Let us assume that a student can take more than one course and one course can be taken by many students. So the relationship will be many to many.

the total number of tables that can be used in this is 3.



*many to many cardinality*

Using Sets, it can be represented as:



*Many-to-Many Set Representation*

In this example, student S1 is enrolled in C1 and C3 and Course C3 is enrolled by S1, S3, and S4. So it is many-to-many relationships.

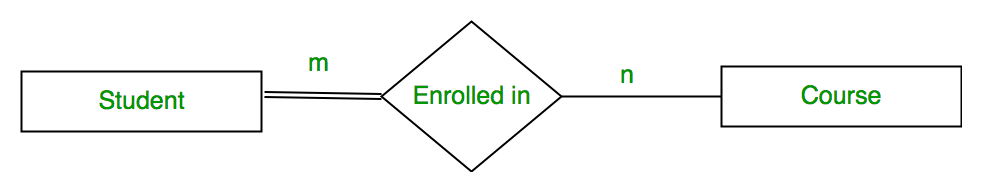
### **Participation Constraint**

[Participation Constraint](https://www.geeksforgeeks.org/structural-constraints-of-relationships-in-er-model/) is applied to the entity participating in the relationship set.

**1. Total Participation –** Each entity in the entity setmust participate in the relationship. If each student must enroll in a course, the participation of students will be total. Total participation is shown by a double line in the ER diagram.

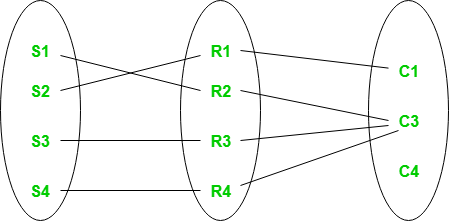
**2. Partial Participation –** The entity in the entity set may or may NOT participate in the relationship. If some courses are not enrolled by any of the students, the participation in the course will be partial.

The diagram depicts the ‘Enrolled in’ relationship set with Student Entity set having total participation and Course Entity set having partial participation.



*Total Participation and Partial Participation*

Using Set, it can be represented as,



*Set representation of Total Participation and Partial Participation*

Every student in the Student Entity set participates in a relationship but there exists a course C4 that is not taking part in the relationship.

## How to Draw ER Diagram?

* The very first step is Identifying all the Entities, and place them in a Rectangle, and labeling them accordingly.
* The next step is to identify the relationship between them and pace them accordingly using the Diamond, and make sure that, Relationships are not connected to each other.
* Attach attributes to the entities properly.
* Remove redundant entities and relationships.
* Add proper colors to highlight the data present in the database.

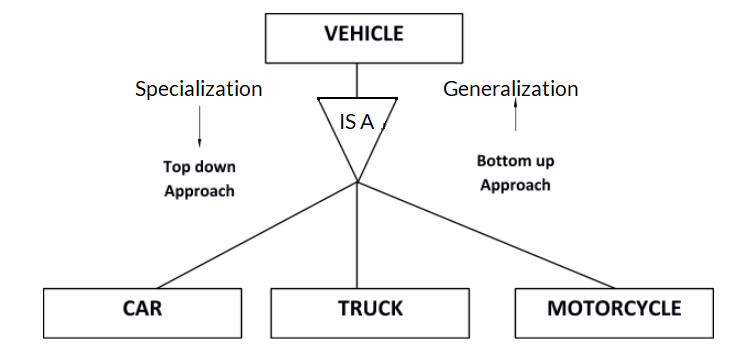
# Subclasses, Superclasses, and Inheritance

The Enhanced Entity Relationship Model contains all the features of the [**Entity Relationship model**](https://www.tutorialspoint.com/what-is-an-entity-relationship-model-in-dbms). In addition to all that, it also contains features of Subclasses, Superclasses and Inheritance.

All of these in detail are as follows −

## Subclasses

A subclass is a class derived from the superclass. It inherits the properties of the superclass and also contains attributes of its own. An example is:



Car, Truck and Motorcycle are all subclasses of the superclass Vehicle. They all inherit common attributes from vehicle such as speed, colour etc. while they have different attributes also i.e Number of wheels in Car is 4 while in Motorcycle is 2.

## Superclasses

A superclass is the class from which many subclasses can be created. The subclasses inherit the characteristics of a superclass. The superclass is also known as the parent class or base class.

In the above example, Vehicle is the Superclass and its subclasses are Car, Truck and Motorcycle.

## Inheritance

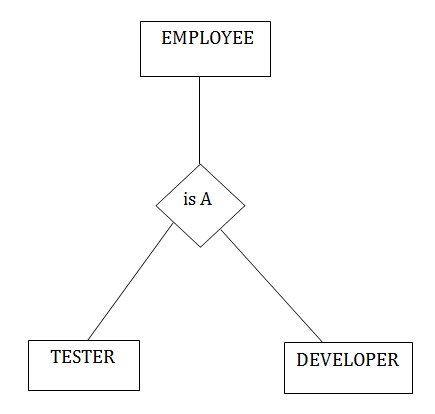
Inheritance is basically the process of basing a class on another class i.e to build a class on a existing class. The new class contains all the features and functionalities of the old class in addition to its own.

The class which is newly created is known as the subclass or child class and the original class is the parent class or the superclass.

# Specialization

* **Specialization is a top-down approach, and it is opposite to Generalization. In specialization, one higher level entity can be broken down into two lower level entities.**
* **Specialization is used to identify the subset of an entity set that shares some distinguishing characteristics.**
* **Normally, the superclass is defined first, the subclass and its related attributes are defined next, and relationship set are then added.**

**For example: In an Employee management system, EMPLOYEE entity can be specialized as TESTER or DEVELOPER based on what role they play in the company.**

****

## Specialization in DBMS

A software developer is a person who can develop software. Now the software can be a web application, a mobile application, or, say, a desktop application. Hence, we can have developers specializing in a particular field, whether it's desktop applications, mobile applications, or desktop applications. This intuition of categorizing or dividing a higher-level entity into multiple lower-level entities of a similar kind is known as *specialization*.

### Specialization Characteristics:

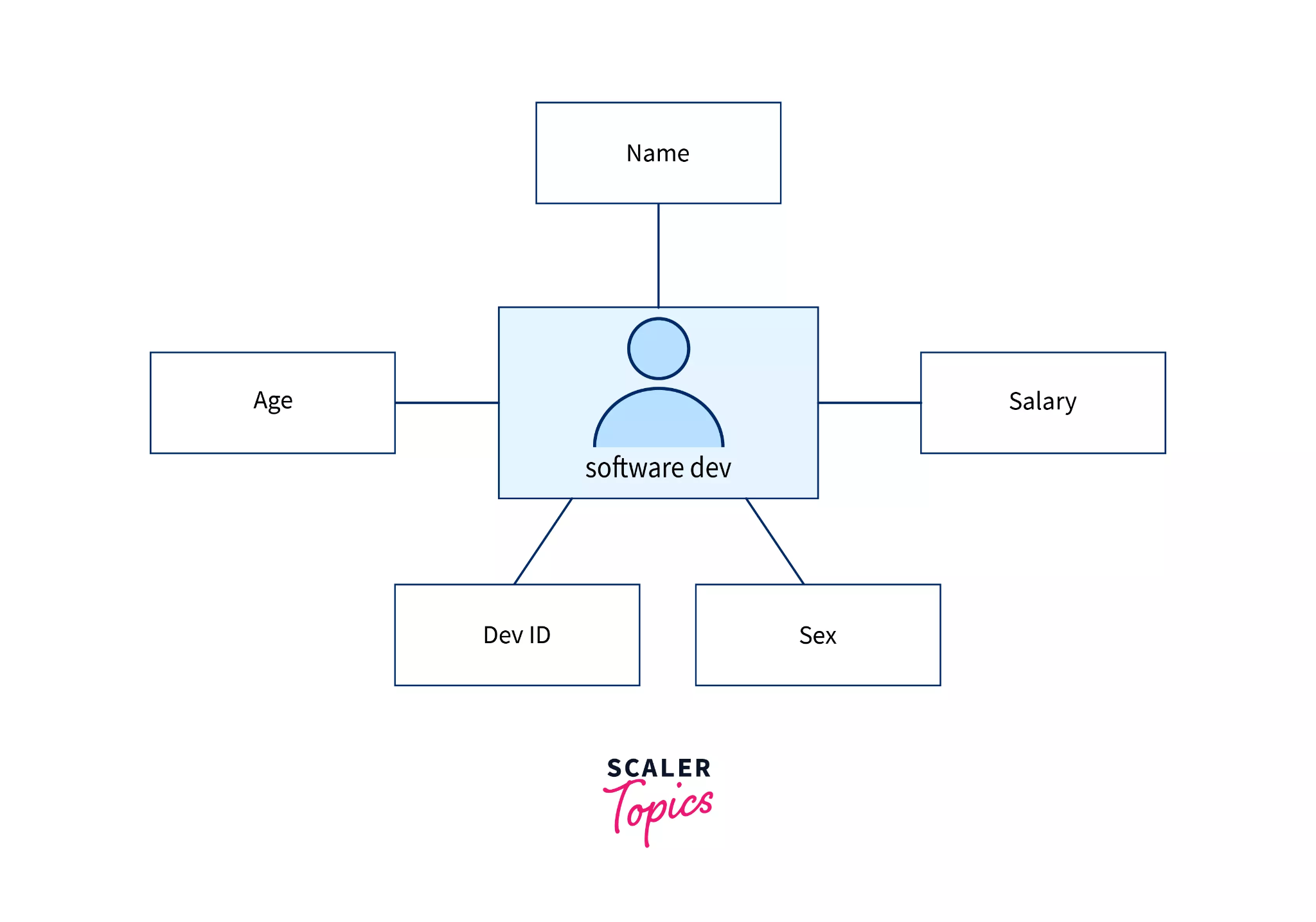
* The top-down technique is used for specialization.
* It divides or specializes in the entity.
* Higher-level entities and lower-level entities can be linked.

**Let us understand it with the help of an example.**

### Entities Before Specialization

Below will be the entities before specialization:

Software Developer: Software developers can have attributes like name, age, salary, etc. We will apply specialization to this entity and will divide it into entities.

****

## Generalization in DBMS

**When someone says, "Social media", what comes to mind?**

**Facebook, Twitter, Instagram, or maybe LinkedIn**

**In short, if we want to talk about Social media in general, we generalize and refer to them as "Social Media". This method of combining multiple lower-level entities of a similar kind and referring to them as a single higher-level entity is what we know as *generalization*.**

### Generalization Characteristics:

* **The bottom-up technique is used for generalization.**
* **It simplifies or generalizes the entities.**
* **Higher-level entities and lower-level entities can be linked.**

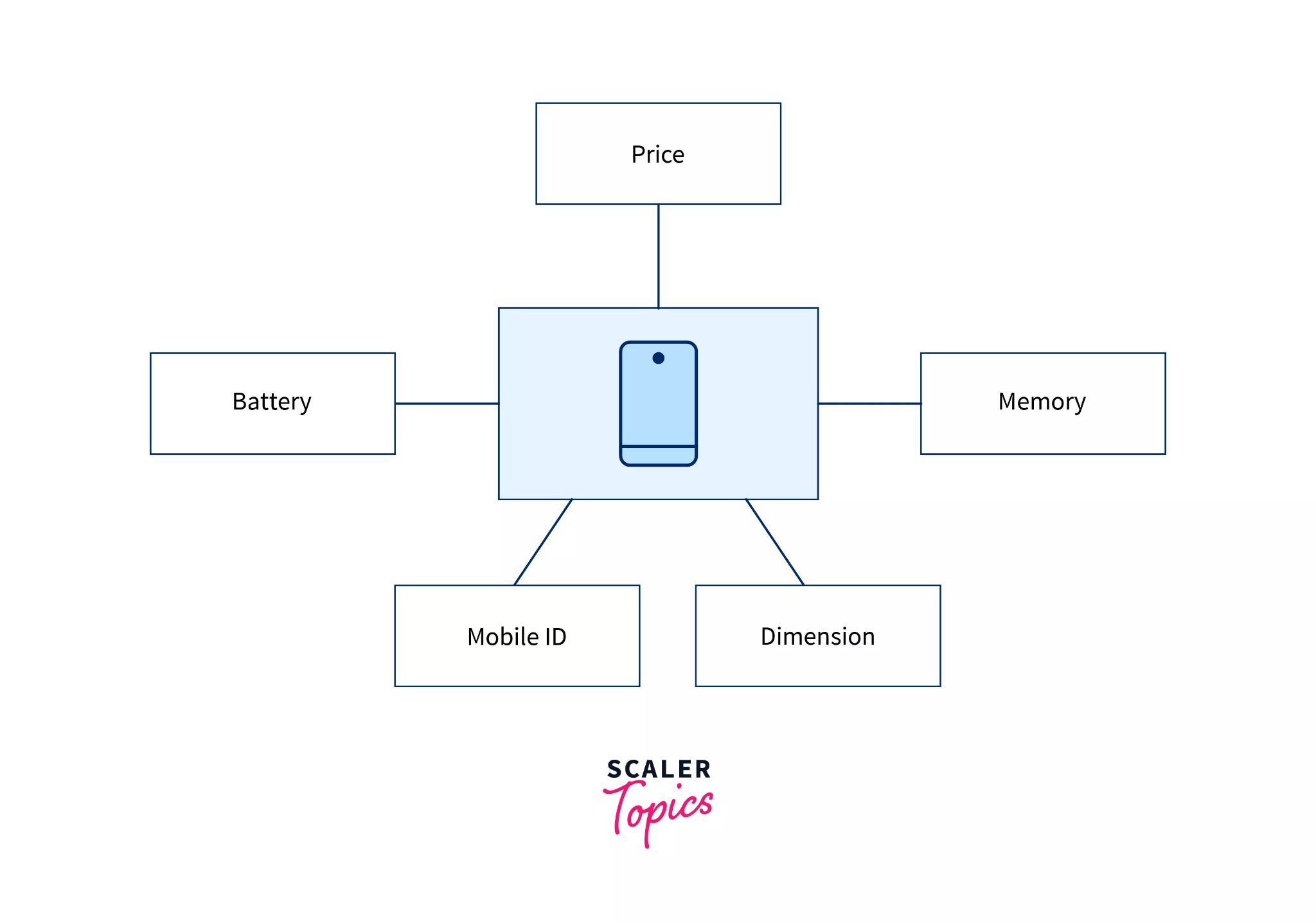
**Let us take an example.**

**Let's say we have entities as mobile and laptop and we need to generalize it based on the common entities between them so before generalization, we will have two separate entities for mobiles and laptops.**

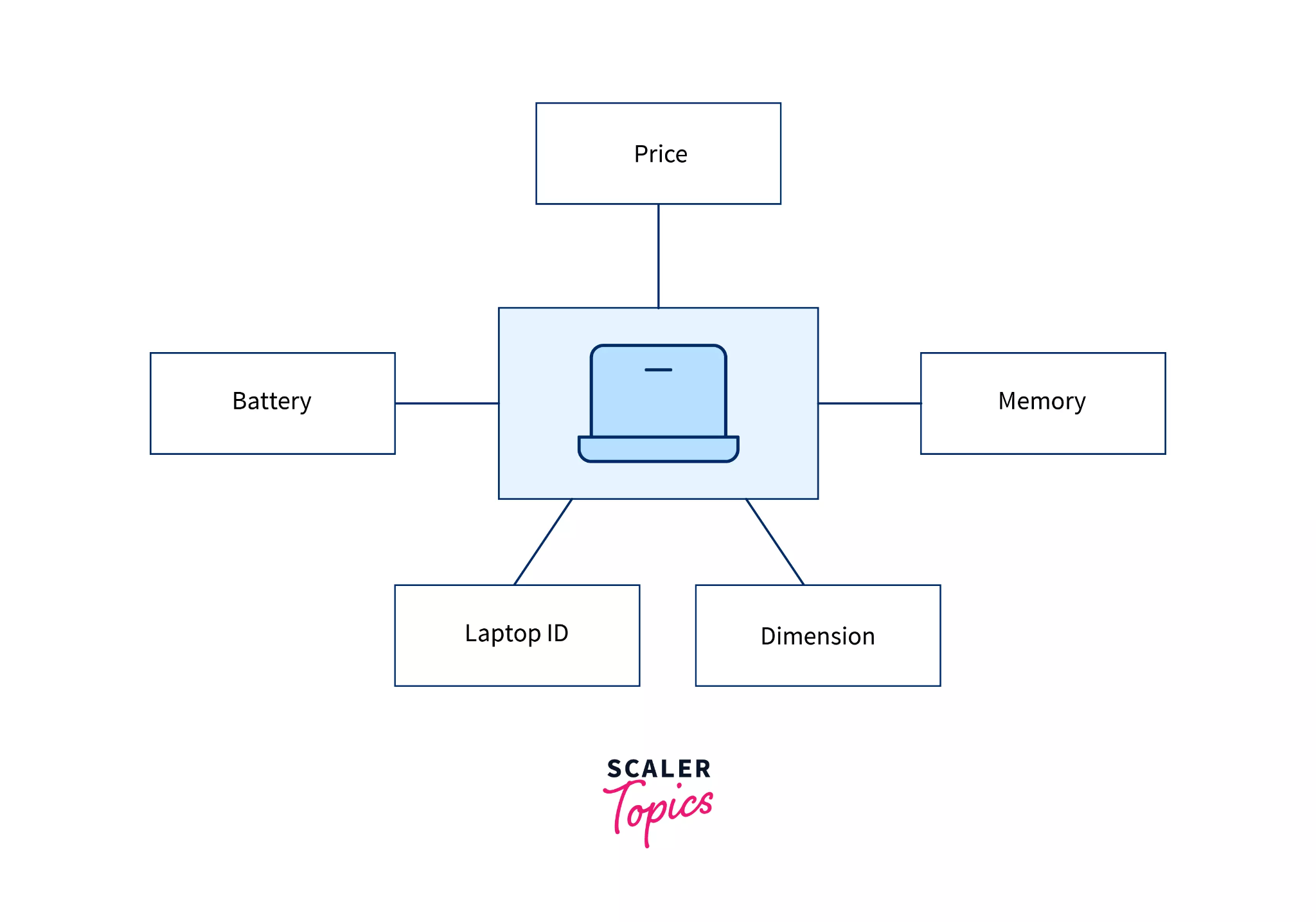
### Entities before Generalization

**Below will be the entities before generalization:**

**Mobile Entity: Mobile entity can have attributes like mobile\_ID, price, memory, battery, and dimension to which we can apply the generalization procedure.**

****

**Laptop Entity: Laptop entity can have attributes like laptop\_ID, price, memory, battery, and dimension to which we can apply the generalization procedure.**

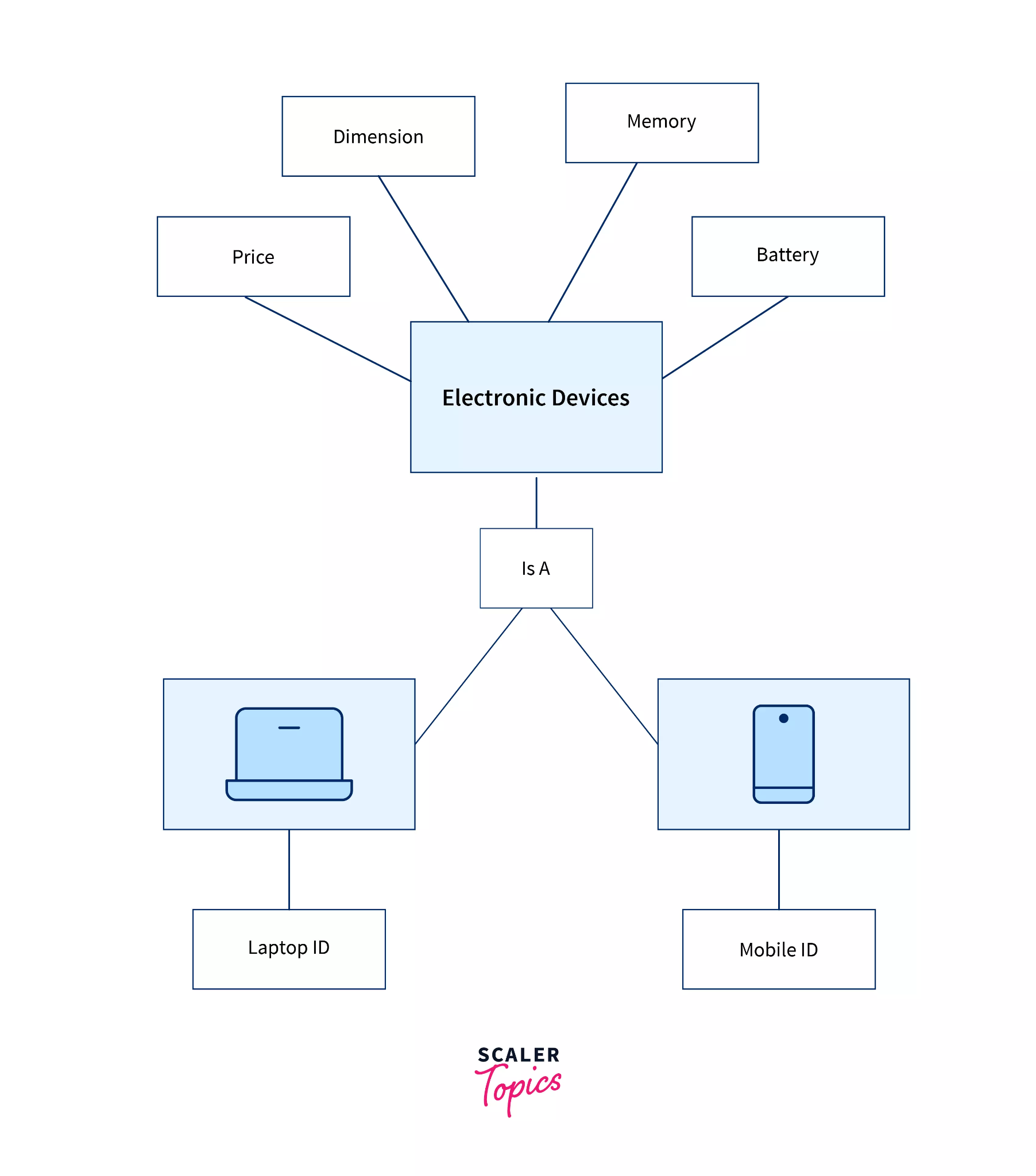
****

### ER Diagram after Generalization

**We can see that for laptops and mobiles some attributes like price, memory, battery, and dimension are the same so we can apply generalization and can group them into an entity called Electronic Devices.**

**The**

***Electronic* *Devices* entity will be having these common attributes and the laptop and mobile entity will be related to them using "Is A" relationship table, so using generalization we can represent ER diagram by:**

****