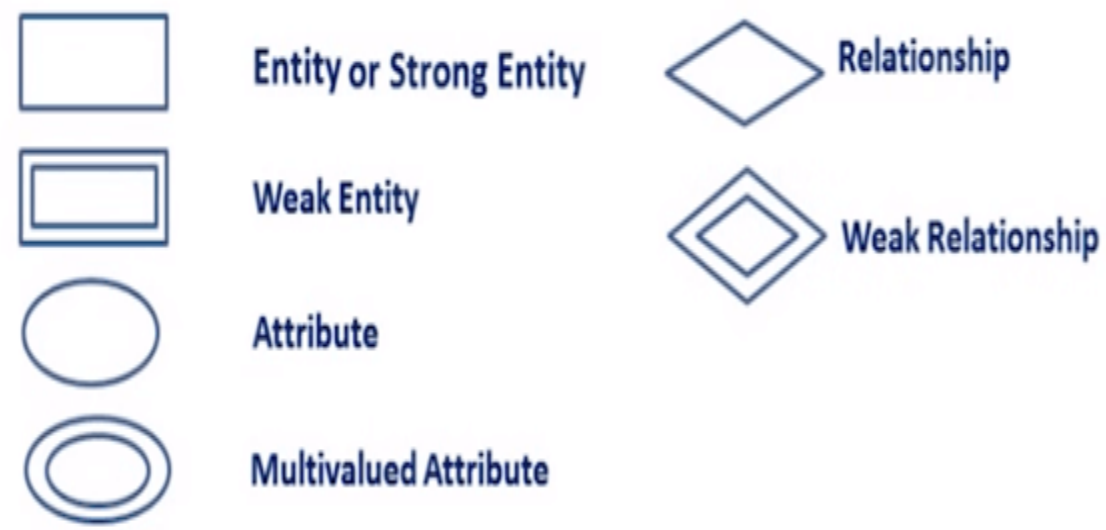
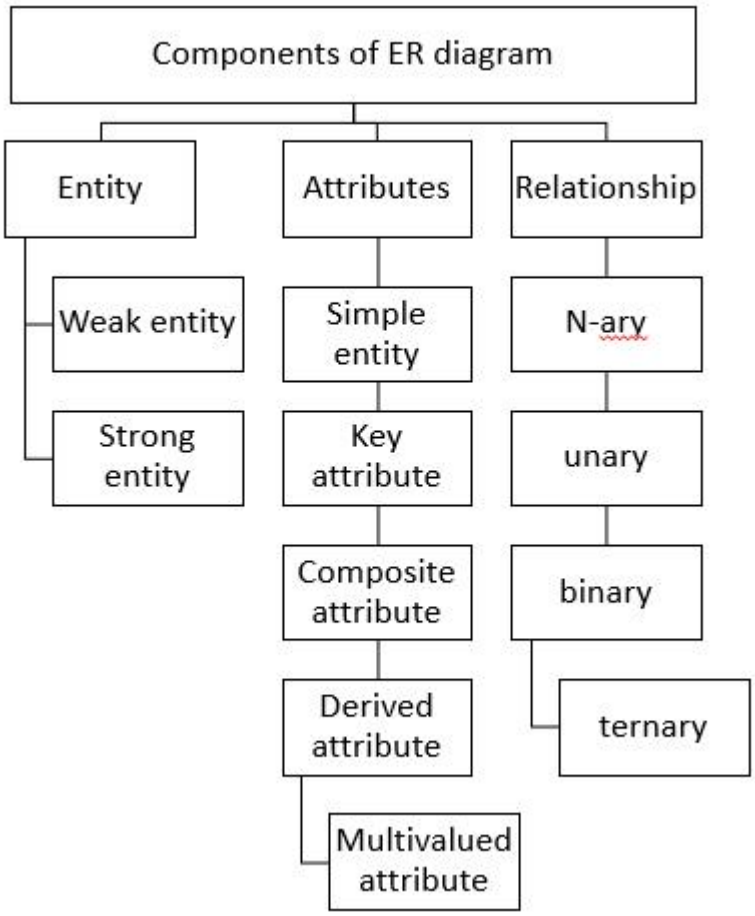
UNIT-II: E-R Modelling

- ER model stands for an **Entity-Relationship** model. It is a high-level data model.
- The Entity Relationship model was proposed by **Peter Chen** in 1976.
- ER model is a logical representation of an enterprise data. ER model is a diagrammatic representation of logical structure of database.
- E-R model describes relationship among entities and attributes.
- Entity Relationship Diagrams are the best tools to communicate within the entire system.
- These diagrams are the graphical representation of the flow of data and information.
- These diagrams are most commonly used in business organizations to make data travel easy.
- This conceptual database model is an effective way of communicating with the individuals at all the levels.
- The most common use of this diagram is to present the relation of the various tables present in a database.

Following are the main components and its symbols in ER Diagrams:

- **Rectangles:** This Entity Relationship Diagram symbol represents entity types
- **Ellipses :** Symbol represent attributes
- **Diamonds:** This symbol represents relationship types
- **Lines:** It links attributes to entity types and entity types with other relationship types
- **Primary key:** attributes are underlined
- **Double Ellipses:** Represent multi-valued attributes





1. Entity

It may be an object, person, place or event that stores data in a database. In E-R diagram an entity is represented in rectangle form. For example, students, employees, managers, etc.

The entity is pictorially depicted as follows:

Entity set

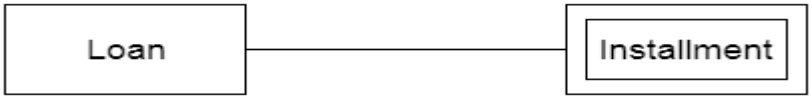
It is a collection of entities of the same type which share similar properties. For example, a group of students in a college and students are an entity set.

Entity is characterised into two types as follows:

a. Strong entity set
b. Weak entity set

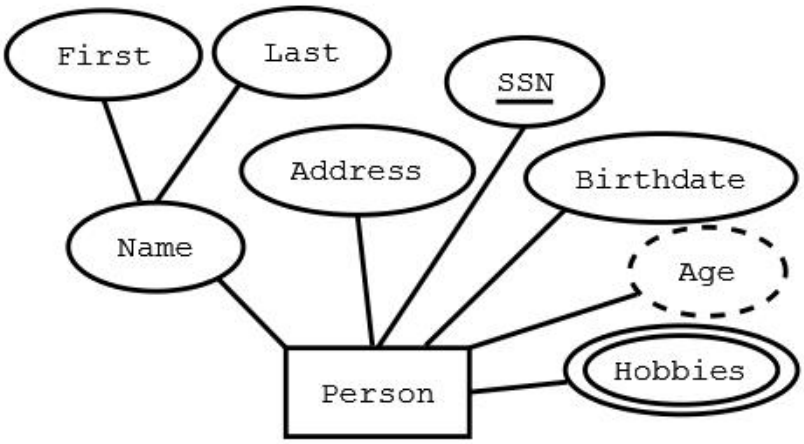
a. **Strong entity set:** The entity types which consist of key attributes or if there are enough attributes for forming a primary key attribute are called a strong entity set. It is represented by a single rectangle.

b. **Weak entity set:** An entity does not have a primary key attribute and depends on another strong entity via foreign key attribute. It is represented by a double rectangle.



2. Attributes

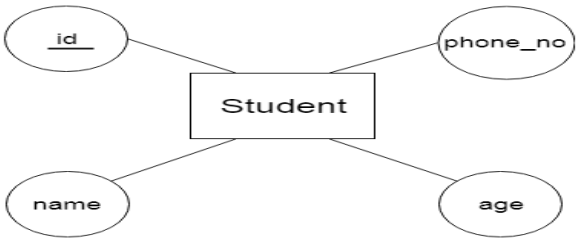
These are the data characteristics of entities or data elements and data fields.



Types of attributes

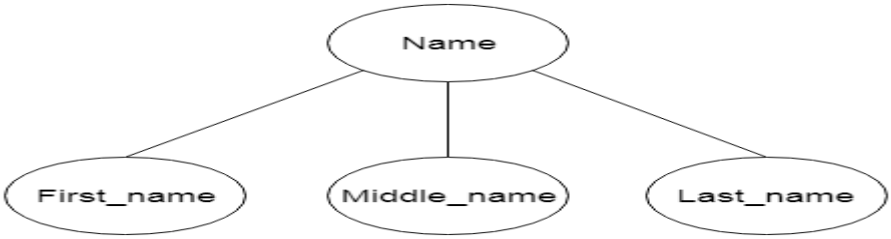
The types of attributes in the Entity Relationship (ER) model are as follows:

1. **Single value attribute** − these attributes contain a single value. For example, age, salary etc.
2. **Key Attributes**- The key attribute is used to represent the main characteristics of an entity. It represents a primary key. The key attribute is represented by an ellipse with the text underlined.

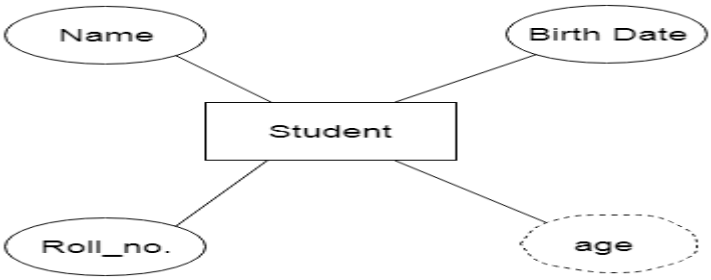


3. **Multivalued attribute** − they contain more than one value of a single entity. For example, phone numbers, Email_Ids, etc.

4. **Composite attribute** − the attributes which can be further divided. For example, **Name** consists of First name, Middle name, last name



5. **Derived attribute** − the attribute that can be derived from other attributes. For example, age can be derived based on DoB.



3. Relationships

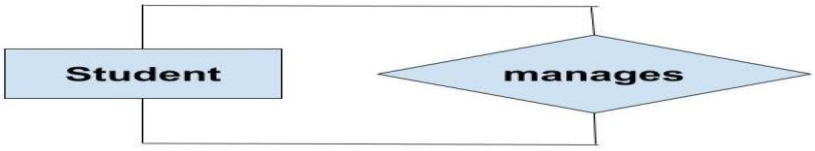
A relationship is used to describe the relation between entities. Diamond or rhombus is used to represent the relationship.

Degree of Relationship: A relationship where a number of different entities set participate is called a degree of a relationship.

It is categorised into the following:

1. **Unary Relationship:** A unary relationship exists when both the participating entity type are the same. When such a relationship is present we say that the degree of relationship is 1.

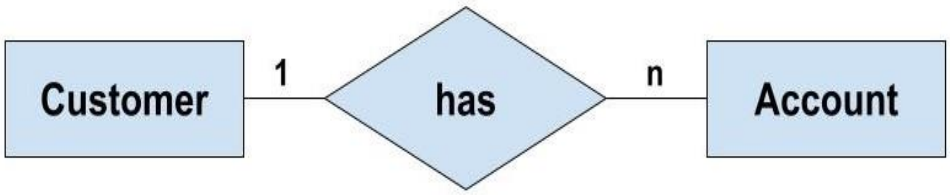
   For example, suppose in a classroom, we have many students who belong to a particular club-like dance club, basketball club etc. and some of them are club leads. So, a particular group of student is managed by their respective club lead. Here, the group is formed from students and also, the club leads are chosen from students. So, the 'Student' is the only entity participating here. We can represent this relationship using the E-R diagram as follows:



Unary Relationship
(degree 1)

2. **Binary Relationship:** A binary relationship exists when exactly two entity type participates. When such a relationship is present we say that the degree is 2. This is the most common degree of relationship. It is easy to deal with such relationship as these can be easily converted into relational tables.

   For example, we have two entity type 'Customer' and 'Account' where each 'Customer' has an 'Account' which stores the account details of the 'Customer'. Since we have two entity types participating we call it a binary relationship. Also, one 'Customer' can have many 'Account' but each 'Account' should belong to only one 'Customer'. We can say that it is a one-to-many binary relationship.

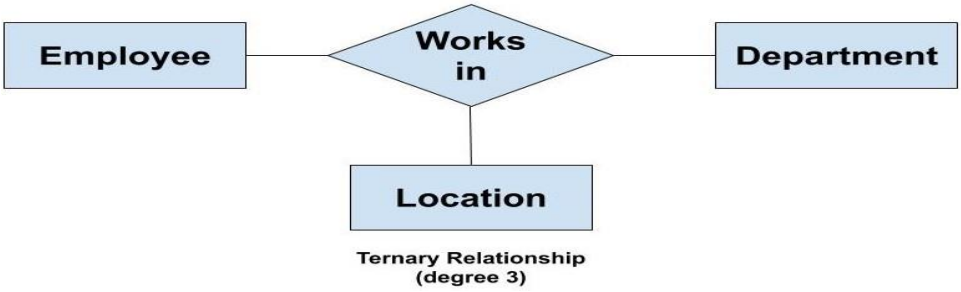


Binary Relationship
(degree 2)

3. **Ternary Relationship:** A ternary relationship exists when exactly three entity type participates. When such a relationship is present we say that the degree is 3. As the number of entity increases in the relationship, it becomes complex to convert them into relational tables.

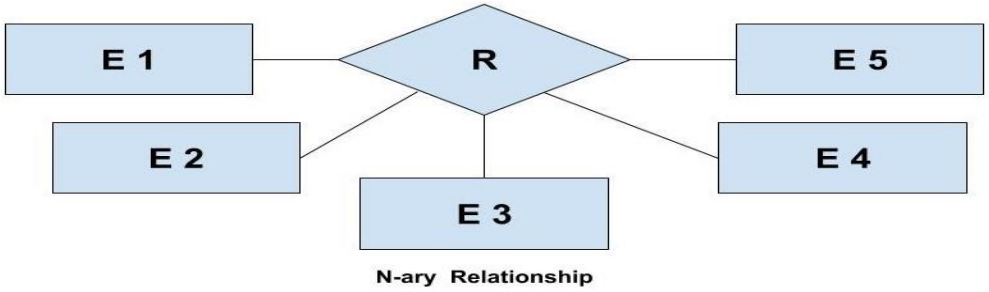
   For example, we have three entity type 'Employee', 'Department' and 'Location'. The relationship between these entities are defined as an employee works in a department,

an employee works at a particular location. So, we can see we have three entities participating in a relationship so it is a ternary relationship. The degree of this relation is 3.



Ternary Relationship (degree 3)

4. **n-ary Relationship:** An N-ary relationship exists when 'n' number of entities are participating. So, any number of entities can participate in a relationship. There is no limitation to the maximum number of entities that can participate.



N-ary Relationship

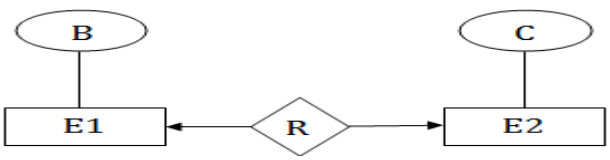
Mapping Constraints

- A mapping constraint is a data constraint that expresses the number of entities to which another entity can be related via a relationship set.
- It is most useful in describing the relationship sets that involve more than two entity sets.

For binary relationship set R on an entity set A and B, there are four possible mapping cardinalities.

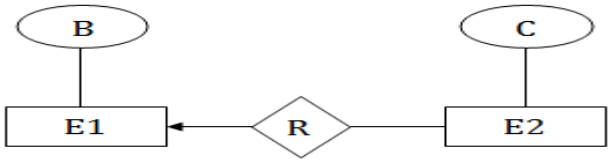
a. One-to-One (1:1)
b. One-to-Many (1:M)
c. Many-to-One (M:1)
d. Many-to-Many (M:M)

a. One-to-One Relationship: When only one instance of an entity is associated with the relationship, then it is known as one to one relationship.



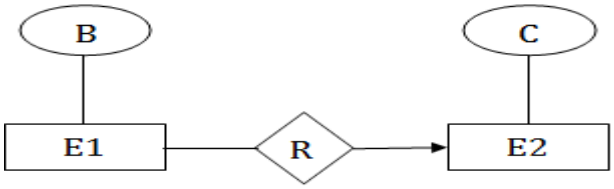
For example: A female can marry to one male, and a male can marry to one female.

b. One-to-many relationship: When only one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then this is known as a one-to-many relationship.



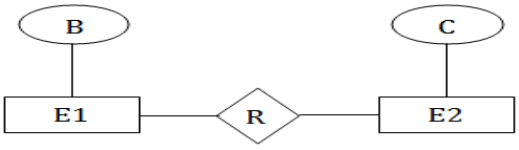
For example, Scientist can invent many inventions, but the invention is done by the only specific scientist.

c. Many-to-one relationship: When more than one instance of the entity on the left, and only one instance of an entity on the right associates with the relationship then it is known as a many-to-one relationship.



For example, Student enrolls for only one course, but a course can have many students.

d. Many-to-many relationship: When more than one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then it is known as a many-to-many relationship.



For example, Employee can assign by many projects and project can have many employees.

Comparison between Strong and Weak Entity Set

Strong Entity Set	Weak Entity Set
1. Strong entity set always has a primary key.	1. It does not have enough attributes to build a primary key.
2. It is represented by a rectangle symbol.	2. It is represented by a double rectangle symbol.
3. It contains a Primary key represented by the underline symbol.	3. It contains a Partial Key which is represented by a dashed underline symbol.
4. The member of a strong entity set is called as dominant entity set.	4. The member of a weak entity set called as a subordinate entity set.
5. Primary Key is one of its attributes which helps to identify its member.	5. In a weak entity set, it is a combination of primary key and partial key of the strong entity set.
6. In the ER diagram the relationship between two strong entity set shown by using a diamond symbol.	6. The relationship between one strong and a weak entity set shown by using the double diamond symbol.
7. The connecting line of the strong entity set with the relationship is single.	7. The line connecting the weak entity set for identifying relationship is double.

Converting E-R model into relational model

A given ER model can be converted into Relational model. A Relational model includes Relations, Tuples, Attributes, Keys, and Foreign keys.

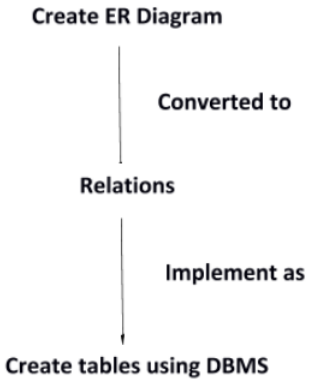
- Relation is a table made from tuples.
- A Tuple is a row of data.
- An Attribute is a characteristic of the relation.

There is a direct mapping between ER model and Relational model.

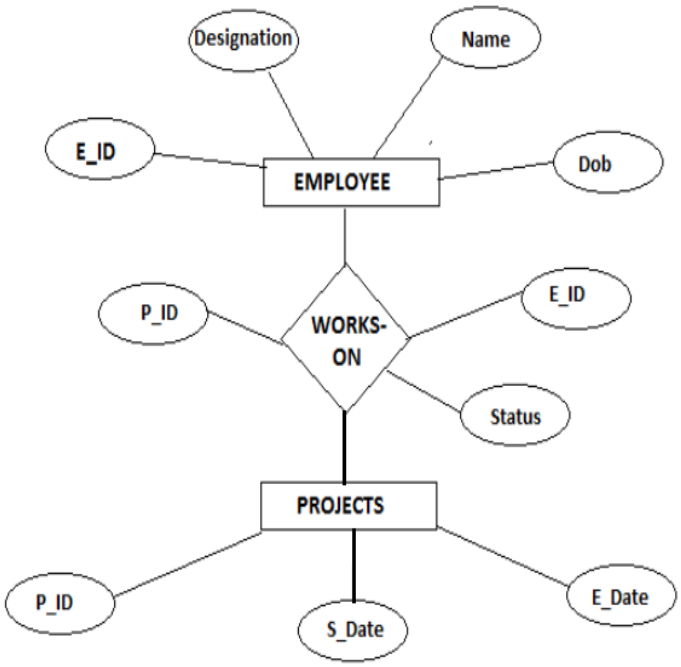
Rules of converting ER model to Relational Model:

- Entity type is converted to a Relation table.
- 1:1 or 1: N relationship type is converted to foreign key.
- M: N relationship type is converted to a relation with two foreign key.
- Simple attribute converted to an attribute.
- Value set converted to a domain.
- Key attribute converted to a primary key.

Overall transformation summary is as follows:



Consider the following **example**:



Now for the above example we can create three relations:

- Employee
- Works_On
- Projects

Transform attributes to fields:

- Employee will have E_ID, Name, Designation and Dob.
- Works_On will have E_ID, Status and P_ID.
- Projects will have P_ID, S_Date and E_Date.

Now we can create tables in DBMS.

Advantages of E-R Model

1. Conceptually E-R model is very simple: ER model is very simple because if we know relationship between entities and attributes, then we can easily draw an ER diagram.
2. Better Visual representation: ER model is a diagrammatic representation of any logical structure of database. By seeing ER diagram, we can easily understand relationship among entities and relationship.
3. Effective communication tool: It is an effective communication tool for database designer.

The clear representation of the data listed under proper headings and tables results in the effective flow of information and communication.

4. Highly integrated with relational model: ER model can be easily converted into relational model by simply converting ER model into tables.
5. Easy conversion to any data: ER model can be easily converted into another data model like hierarchical data model, network data model and so on.
6. Straightforward relation representation: Having designed an E-R diagram for a database application, the relational representation of the database model becomes relatively straightforward.

Disadvantages of E-R Model

1. Limited constraints and specification: The constraints and specifications are limited.
2. Loss of information content: Some information be lost or hidden in ER model.
3. Limited relationship representation: ER model represents limited relationship as compared to another data models like relational model etc.
4. No representation of data manipulation: It is difficult to show data manipulation in ER model.
5. No industry standard for notation.

Enhanced Entity-Relationship Model (EER model)

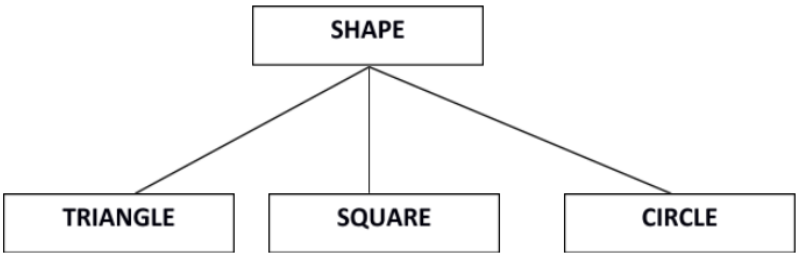
EER is a high-level data model that incorporates the extensions to the original ER model. Enhanced ER Diagrams are high level models that represent the requirements and complexities of complex database.

In addition to ER model concepts EE-R includes −

- Subclasses and Super classes.
- Specialization and Generalization.
- Category or union type.
- Aggregation.

Subclasses and Super class

- Super class is an entity that can be divided into further subtype.
- For **example** − consider Shape super class.

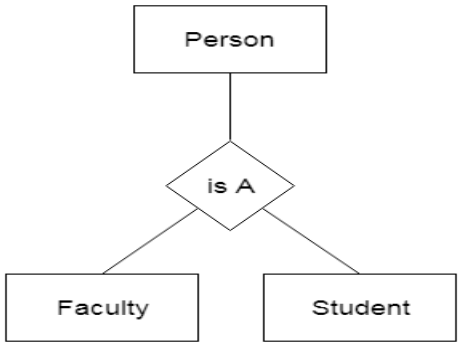


- Super class shape has sub groups: Triangle, Square and Circle.
- Sub classes are the group of entities with some unique attributes. Subclass inherits the properties and attributes from super class.

Specialization and Generalization

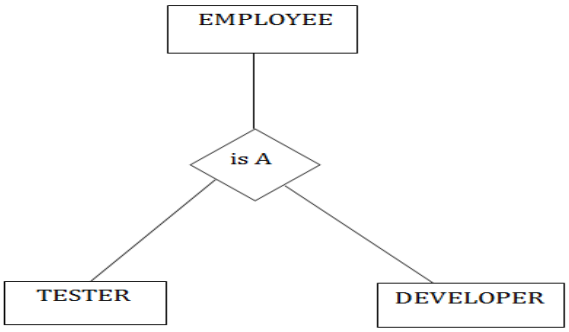
- **Generalization** is like a bottom-up approach in which two or more entities of lower level combine to form a higher level entity if they have some attributes in common.
- In generalization, entities are combined to form a more generalized entity, i.e., subclasses are combined to make a superclass.

For example, Faculty and Student entities can be generalized and create a higher level entity Person.



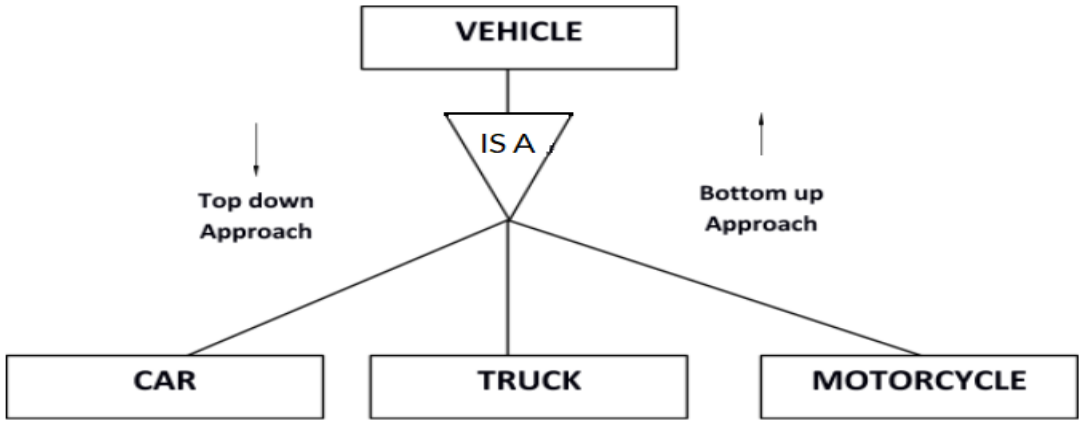
- **Specialization** is a top-down approach, and it is opposite to Generalization. In specialization, one higher level entity can be broken down into two lower level entities.
- Specialization is used to identify the subset of an entity set that shares some distinguishing characteristics.
- Normally, 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.



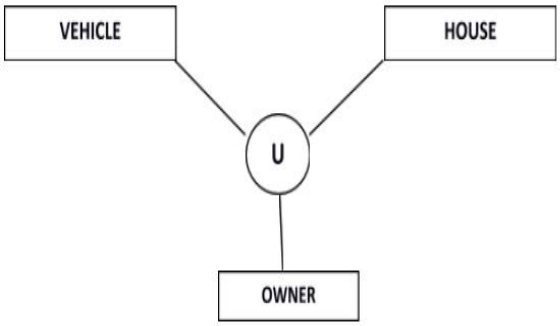
For example, in the bellow diagram we have 3 sub entities Car, Truck and Motorcycle. The three entities can be generalized into one super class named as **Vehicle (Generalization).**

Specialization is a process of identifying subsets of an entity that share some different characteristic. It is a top down approach in which one entity is broken down into low level entity i.e., Vehicle entity can be a Car, Truck or Motorcycle (**Specializations**).



Category or Union

- Relationship of one super or sub class with more than one super class.

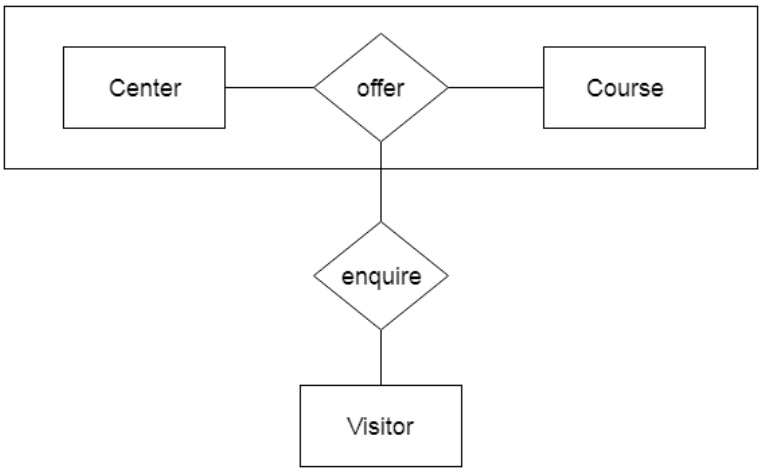


- Owner is the subset of two super class: Vehicle and House.

Aggregation

In aggregation, the relation between two entities is treated as a single entity. In aggregation, relationship with its corresponding entities is aggregated into a higher level entity.

For example: Centre entity offers the Course entity act as a single entity in the relationship which is in a relationship with another entity visitor. In the real world, if a visitor visits a coaching Centre then he will never enquiry about the Course only or just about the Centre instead he will ask the enquiry about both.

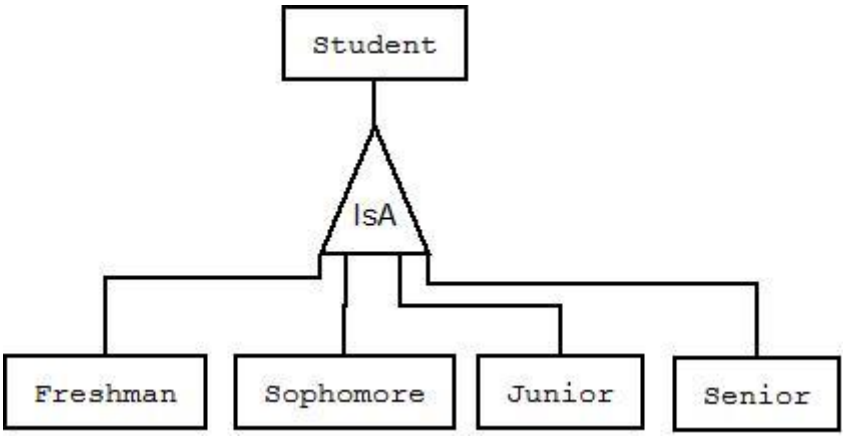


IsA Relationship and Attribute Inheritance

This relationship is called ***IsA***. Some texts will call this an IsA relationship, but **do not confuse** with the concept of relationship between entities.

- Freshman IsA Student, an eagle IsA bird
- The two entities represented by IsA are always descriptions of the same real-world object
- Typically used in databases to be implemented as Object Oriented Models.

- The upper entity type (connected to the apex of the IsA triangle) is the more abstract/general entity type (super type) from which the lower entities inherit its attributes.



Properties of IsA

1. Inheritance - All attributes of the supertype apply to the subtype.

- E.g., *An a*ttribute of **Student** applies to **Freshman**
- **The subtype *inherits* all attributes of its supertype.**
- The key of the supertype is also the key of the subtype

2. Transitivity - This property creates a hierarchy of IsA relationships

- **Student** is subtype of **Person**,
  Freshman is subtype of **Student**,
  therefore **Freshman** is also a subtype of **Person**

References

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan,"Database System Concepts", Seventh Edition.
2. Jeffrey A. Hoffer, V. Ramesh, Heikki Topi, "Modern Database Management", Tenth Edition.
3. Raghu Ramakrishnan, "Database Management Systems".
4. https://onlinecourses.nptel.ac.in/noc19_cs46/
5. https://nptel.ac.in/courses/106/106/106106220/