# Enhanced-Entity-Relationship (EER) Model

**Notes by Mannan Ul Haq**

## EER Diagram Concepts

An Enhanced Entity-Relationship (EER) diagram is a visual tool used to display more complex database schemas that include advanced data modeling concepts beyond the basic Entity-Relationship (ER) model. The EER model incorporates the following key concepts:

1. **Sub Class and Super Class**
2. **Specialization and Generalization**

These concepts help in creating more accurate and detailed database schemas.

## Key Concepts Explained

## A. Sub Class and Super Class

- **Super Class**: An entity type that is the parent to one or more subtypes. For instance, a "Shape" super class might have subtypes like "Square," "Circle," and "Triangle."
- **Sub Class**: A more specific entity type that inherits attributes and relationships from its super class. For example, "Square," "Circle," and "Triangle" are sub classes of the "Shape" super class.
- **Inheritance**: Sub classes inherit attributes and behaviors from their super class.

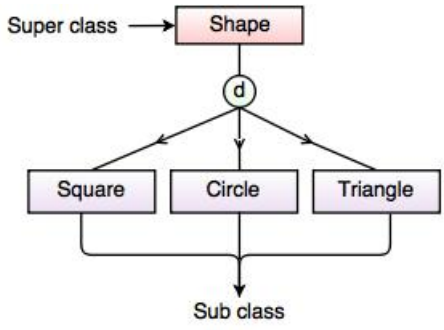



Fig. Super class/Sub class Relationship

## B. Specialization and Generalization

- **Generalization**: A bottom-up approach where multiple lower-level entities are combined into a higher-level entity. It identifies common features among entities.
  - **Example**: Animals (generalized from Tiger, Lion, Elephant).

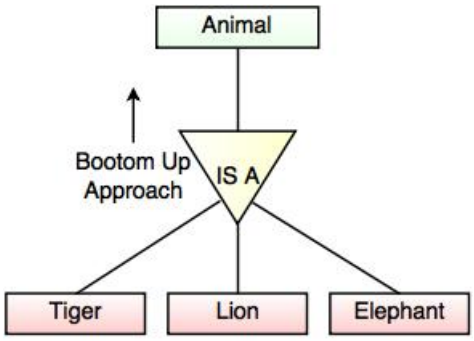



Fig. Generalization

- **Specialization**: A top-down approach where a higher-level entity is divided into more specific sub-entities based on their unique characteristics.
  - **Example**: Employee can be specialized into Developer or Tester.

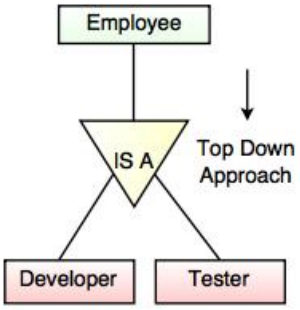



Fig. Specialization

## C. Category or Union

**Union (Category)**: Represents a relationship where a sub class has more than one super class. It can involve total or partial participation.

- **Example**: A "Car Owner" category can be a person, a bank (possesses the car), or a company. The "Owner" sub class is a union of the "Person," "Bank," and "Company" super classes.

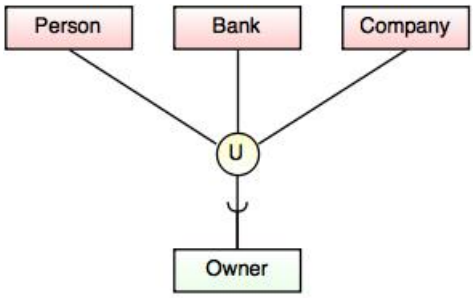



Fig. Categories (Union Type)

## D. Aggregation

- Represents a relationship where a relationship itself is treated as an entity.
- Abstraction of a relationship between objects, viewing the relationship as a whole.
- Example: The relationship between `College` and `Course` can be aggregated into an entity that relates to `Student`.

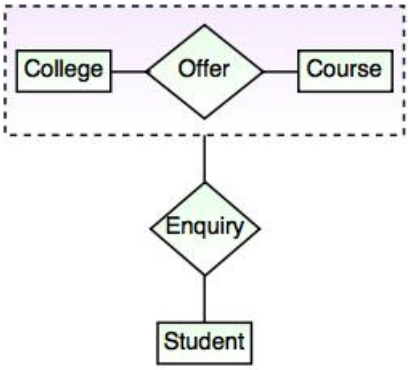
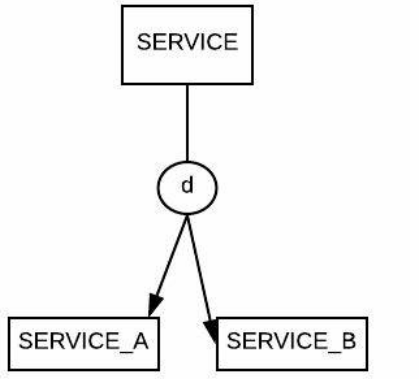


Fig. Aggregation

## E. Disjoint and Overlapping

**Disjoint (Exclusive)**:

- Sub classes that do not share common members.
- An entity instance can belong to only one sub class.
- Example: A `Vehicle` can be either a `Car` or a `Truck`, but not both.
- Denoted by a "d" in the diagram.




**Overlapping**:

- Sub classes that can share common members.
- An entity instance can belong to more than one sub class.
- Example: A `Person` can be both an `Employee` and a `Student`.

- Denoted by an "o" in the diagram.