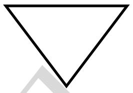
LEC-4: Extended ER Features

1. **Basic ER Features** studied in the **LEC-3,** can be used to model most DB features but when complexity increases, it is better to use some Extended ER features to model the DB Schema.
2. **Specialisation**
    1. In ER model, we may require to subgroup an entity set into other entity sets that are distinct in some way with other entity sets.
    2. **Specialisation** is **splitting** up the entity set into further **sub entity sets** on the basis of their **functionalities**, **specialities** and **features**.
    3. It is a **Top-Down** approach.
    4. e.g., **Person** entity set can be divided into **customer**, **student**, **employee**. Person is **superclass** and other specialised entity sets are **subclasses**.
        1. We have **"is-a"** relationship between superclass and subclass.
        2. Depicted by **triangle** component.



    5. **Why** Specialisation?
        1. Certain attributes may only be applicable to a few entities of the parent entity set.
        2. DB designer can show the distinctive features of the sub entities.
        3. To group such entities we apply Specialisation, to overall **refine** the DB blueprint.
3. **Generalisation**
    1. It is just a **reverse** of Specialisation.
    2. DB Designer, may encounter certain properties of two entities are overlapping. Designer may consider to make a new generalised entity set. That generalised entity set will be a super class.
    3. **"is-a"** relationship is present between subclass and super class.
    4. e.g., **Car**, **Jeep** and **Bus** all have some common attributes, to avoid data repetition for the common attributes. DB designer may consider to Generalise to a new entity set "**Vehicle**".
    5. It is a **Bottom-up** approach.
    6. **Why** Generalisation?
        1. Makes DB more **refined** and **simpler**.
        2. Common attributes are not **repeated**.
4. **Attribute Inheritance**
    1. **Both** Specialisation and Generalisation, has attribute inheritance.
    2. The attributes of higher level entity sets are inherited by lower level entity sets.
    3. E.g., **Customer & Employee** inherit the attributes of **Person**.
5. **Participation Inheritance**
    1. If a parent entity set participates in a relationship then its child entity sets will also participate in that relationship.
6. **Aggregation**
    1. **How to show relationships among relationships?** - Aggregation is the technique.
    2. **Abstraction** is applied to treat relationships as higher-level entities. We can call it Abstract entity.
    3. **Avoid redundancy** by aggregating relationship as an entity set itself.