Chapter 4, Enhanced Entity-Relationship Modeling

4.1 Subclasses, Superclasses, and Inheritance

- A **Subclass** \( S \) of an entity type \( C \) is a subset of \( C \), i.e., \( S \subseteq C \).
- We call \( C \) a **superclass** of the subclass \( S \).
- All entities in \( S \) have some similar characteristics that all other entities in \( C \) do not possess.
- An entity in a subclass and the corresponding entity in a superclass refer to the same real-world object.
- An entity cannot exist in the database merely by being a member of a subclass; it must also be a member of the superclass.
- For example, an **EMPLOYEE** entity type can be grouped further into **SECRETARY**, **ENGINEER**, **MANAGER**, **TECHNICIAN**, **SALARIED_EMPLOYEE**, **HOURLY_EMPLOYEE**, ... Each of these subgroupings above is a subclass of **EMPLOYEE**, and **EMPLOYEE** is a superclass of each of these subclasses.
- We call the relationship between a superclass and any one of its subclasses a **superclass/subclass** (C/S in short) or **IS-A relationship**.
  
  e.g. **EMPLOYEE/SECRETARY**, **EMPLOYEE/ENGINEER**.
- An entity that is a member of a subclass **inherits** all the attributes of the entity as a member of the superclass and all the relationships in which the superclass participates.
  
  See Figure 4.1 (Figure 4.1 on e3).

4.2 Specialization and Generalization

- **Specialization** is the process of defining a set of subclasses of an entity type.
• There should have some distinguishing characteristic of the entities in the superclass for the specialization process that specializes entities in the superclass into a set of subclasses.
  
  e.g. EMPLOYEE $\rightarrow \{\text{SECRETARY, ENGINEER, TECHNICIAN}\}$ is based on the job type of each entity.

• The C/S relationship should be 1:1 relationship. The main difference is that in a 1:1 relationship type two distinct entities are related, whereas in the C/S relationship only one real-world entity is involved but playing in different roles.

• Two reasons for including C/S relationships and specializations/generalizations in the data model.
  
  − Certain attributes may apply to some but not all entities of the superclass.
  
  − Some relationship types may be participated in only by entities that are members of the subclass.
  
  − See example in Figure 4.1 (Fig 4.1 on e3).

• Generalization is the reverse process of specialization. For several entity types, we suppress the differences among them, identify their common features, and generalize them into a single superclass. e.g. See Figure 4.3 (Fig 4.3 on e3).

### 4.3 Constraints and Characteristics of Specialization and Generalization

• A superclass may have several specializations. e.g. EMPLOYEE $\rightarrow \{\text{SECRETARY, TECHNICIAN, ENGINEER}\}$ and \{SALARIED_EMPLOYEE, HOURLY_EMPLOYEE\} and \{MANAGER\}.

• If there is a condition on values for some attributes of the superclass that can determine exactly which entities are members of a subclass. This kind of subclasses are called predicate-defined subclasses. Also, the condition is the defining predicate of the subclass.
For example, if the EMPLOYEE entity type has a JobType attribute, we can specify the condition of membership for the SECRETARY subclass by the predicate \((\text{JobType} = \text{"Secretary"})\) in the EMPLOYEE entity type. See Figure 4.4 (Fig 4.4 on e3).

- If all subclasses in a specialization have their membership condition on the same attribute of the superclass, then we call this specialization an **attribute-defined specialization**. Also, we call the attribute a **defining attribute** of the specialization.

- **Disjointness Constraint**: specify that the subclasses of the specialization must be disjointed.

- **Completeness Constraint**:
  - **Total**: specify that every entity in the superclass must be a member of some subclasses.
    e.g. EMPLOYEE \(\rightarrow\) \{SALARIED_EMPLOYEE, HOURLY_EMPLOYEE\}.
  - **Partial**: allow an entity not to belong to any of the subclasses. e.g. EMPLOYEE \(\rightarrow\) \{MANAGER\}.
    A superclass that was identified through the generalization process usually is total.

- **Specialization/Generalization Hierarchy**: Every subclass participates as a subclass in only one C/S relationship. See Figure 4.1 (Fig 4.1 on e3).

- **Specialization/Generalization Lattice**: A subclass can be a subclass in more than one C/S relationship. See Figure 4.6, 4.7 (Fig 4.6, 4.7 on e3).

- A subclass inherits the attributes not only of its direct superclass but also of all its predecessor superclasses all the way to the root of the hierarchy or lattice.

- A subclass with more than one superclass is called a **shared subclass**.

### 4.4 Modeling of UNION Types Using Categories
• **Category**: is a collection of objects that is a subset of the UNION of distinct entity types. This is used for modeling a single C/S relationship with more than one superclass.

e.g. \{PERSON, BANK, COMPANY\} → VEHICLE_OWNER

• Compare ENGINEERING_MANAGER in Figure 4.6 (Fig 4.6 on e3) to OWNER in Figure 4.8 (Fig 4.8 on e3).

ENGINEERING_MANAGER is a subset of the intersection of three superclasses; whereas Owner is a subset of the union of three superclasses.

• The inheritance of a category vs. the inheritance of a shared subclass.

OWNER in Figure 4.8 (Fig 4.8 on e3) vs. ENGINEERING_MANAGER in Figure 4.6 (Fig 4.6 on e3).

• Compare REGISTERED_VEHICLE category in Figure 4.8 (Fig 4.8 on e3) to the generalized superclass VEHICLE in Figure 4.3(b) (Fig 4.3(b) on e3).

• A category can be **total** or **partial**. See Figure 4.9 on e3.

• If a category is total, it may be represented alternatively as a specialization. See Figure 4.9(b) on e3.

• **The Notation for EER Model**: See next page.
1. Partial Specialization/Generalization
2. X = 'd' means Disjoint
3. X = 'o' means overlap
4. The subset sign indicates the direction of a C/S relationship

Total Specialization/Generalization

1. Partial Category
2. U means UNION

Total Category