

# Chapter 3: Enhanced Entity- Relationship (EER) Model

cuu duong than cong . com

# Contents

---

- 
- 1 Introduction to Enhanced-ER Model
  - 2 Subclasses, Superclasses, and Inheritance
  - 3 Specialization and Generalization
  - 4 Constraints and Characteristics of Specialization and Generalization Hierarchies
  - 5 Categories
  - 6 Design Choices, and Formal Definitions
-

# Contents

---

- 
- 1 Introduction to Enhanced-ER Model**
  - 2 Subclasses, Superclasses, and Inheritance
  - 3 Specialization and Generalization
  - 4 Constraints and Characteristics of Specialization and Generalization Hierarchies
  - 5 Categories
  - 6 Design Choices, and Formal Definitions
-

# Introduction to Enhanced-ER Model

- **Enhanced ER or Extended ER (EER) model**
  - Created to design more accurate database schemas
    - Reflect the data properties and constraints more precisely
  - More complex requirements than traditional applications

# Introduction to Enhanced-ER Model

- EER model includes all modeling concepts of the ER model
- In addition, EER includes:
  - ❑ **Subclasses and superclasses**
  - ❑ **Specialization and generalization**
  - ❑ **Category or union type**
  - ❑ **Attribute and relationship inheritance**
- **EER diagrams**
  - ❑ Diagrammatic technique for displaying these concepts in an EER schema

# Contents

---

- 
- 1 Introduction to Enhanced-ER Model
  - 2 Subclasses, Superclasses, and Inheritance**
  - 3 Specialization and Generalization
  - 4 Constraints and Characteristics of Specialization and Generalization Hierarchies
  - 5 Categories
  - 6 Design Choices, and Formal Definitions
-

# Subclasses, Superclasses & Inheritance

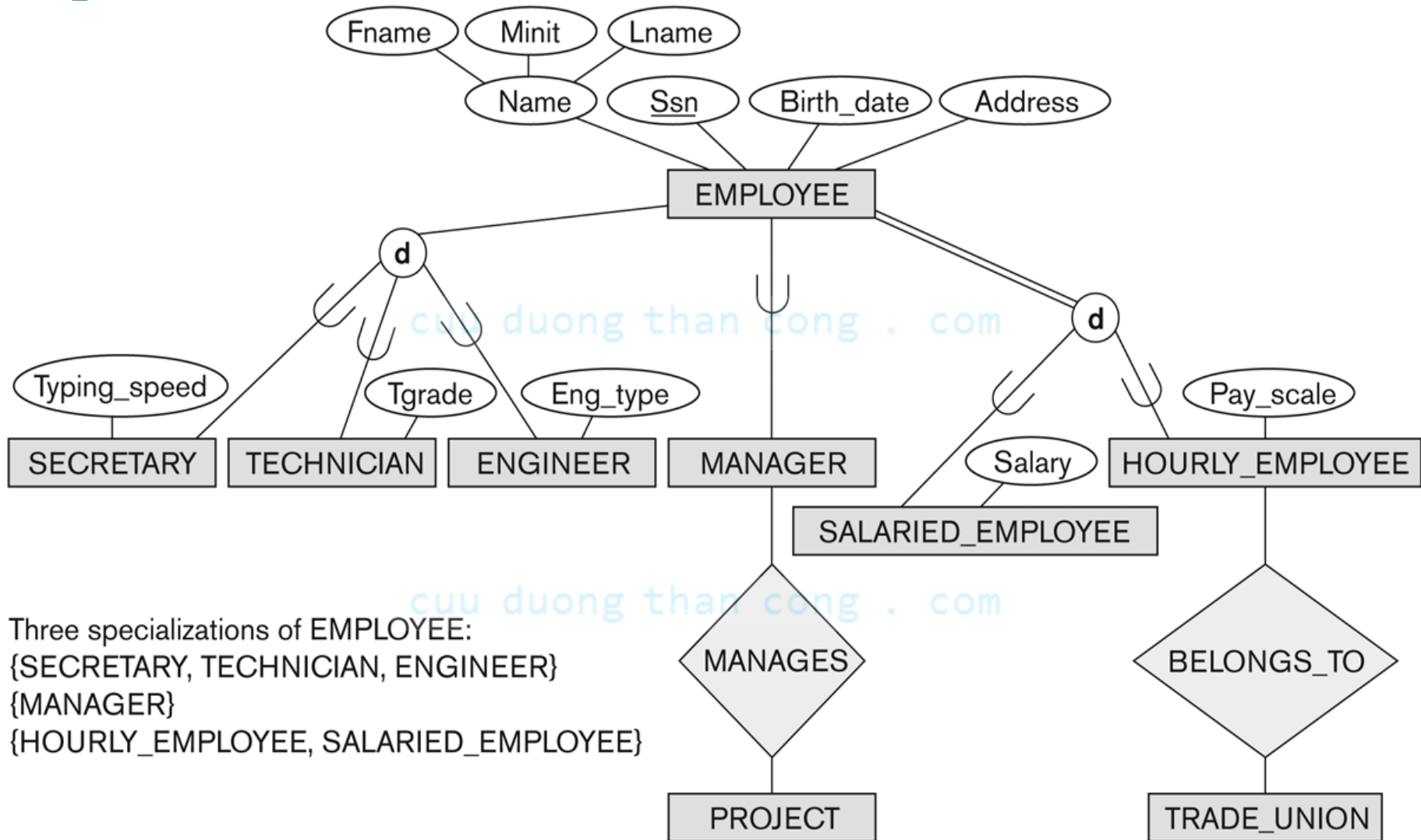
- **Subtype** or **subclass** of an entity type
  - Subgroupings of entities that are meaningful
  - Represented explicitly because of their significance to the database application
- Terms for relationship between a superclass and any one of its subclasses
  - **Superclass/subclass**
  - **Supertype/subtype**
  - **Class/subclass**
- A class/subclass relationship is often called an **IS-A (IS-AN)** relationship.

# Subclasses, Superclasses & Inheritance

- Ex: EMPLOYEE may be further grouped into SECRETARY, ENGINEER, MANAGER, TECHNICIAN, SALARIED\_EMPLOYEE, HOURLY\_EMPLOYEE, and so on.
  - Set of entities in each subgroups is a subset of the EMPLOYEE entity set.
  - Each is called a **subclass** of EMPLOYEE
  - EMPLOYEE is the **superclass** for each of these subclasses



# EER diagram notation to represent subclasses & specialization



Three specializations of EMPLOYEE:  
{SECRETARY, TECHNICIAN, ENGINEER}  
{MANAGER}  
{HOURLY\_EMPLOYEE, SALARIED\_EMPLOYEE}

# Subclasses, Superclasses & Inheritance

- A subclass member is the same as the entity in the superclass, but in a *distinct specific role*
- An entity cannot exist in the database merely by being a member of a subclass; it must also be a member of the superclass
- A member of a superclass can be optionally included as a member of some of its subclasses
- It is not necessary that every entity in a superclass be a member of some subclass
- Superclass/subclass relationship is one-to-one (1:1)

# Subclasses, Superclasses & Inheritance

- Subclass entity **inherits all attributes and relationships** of superclass
- Notice that a subclass, with its own specific (or local) attributes and relationships together with all the attributes and relationships it inherits from the superclass, can be considered an entity type in its own right.

# Contents

- 
- 1 Introduction to Enhanced-ER Model
  - 2 Subclasses, Superclasses, and Inheritance
  - 3 Specialization and Generalization**
  - 4 Constraints and Characteristics of Specialization and Generalization Hierarchies
  - 5 Categories
  - 6 Design Choices, and Formal Definitions
-

# Specialization and Generalization

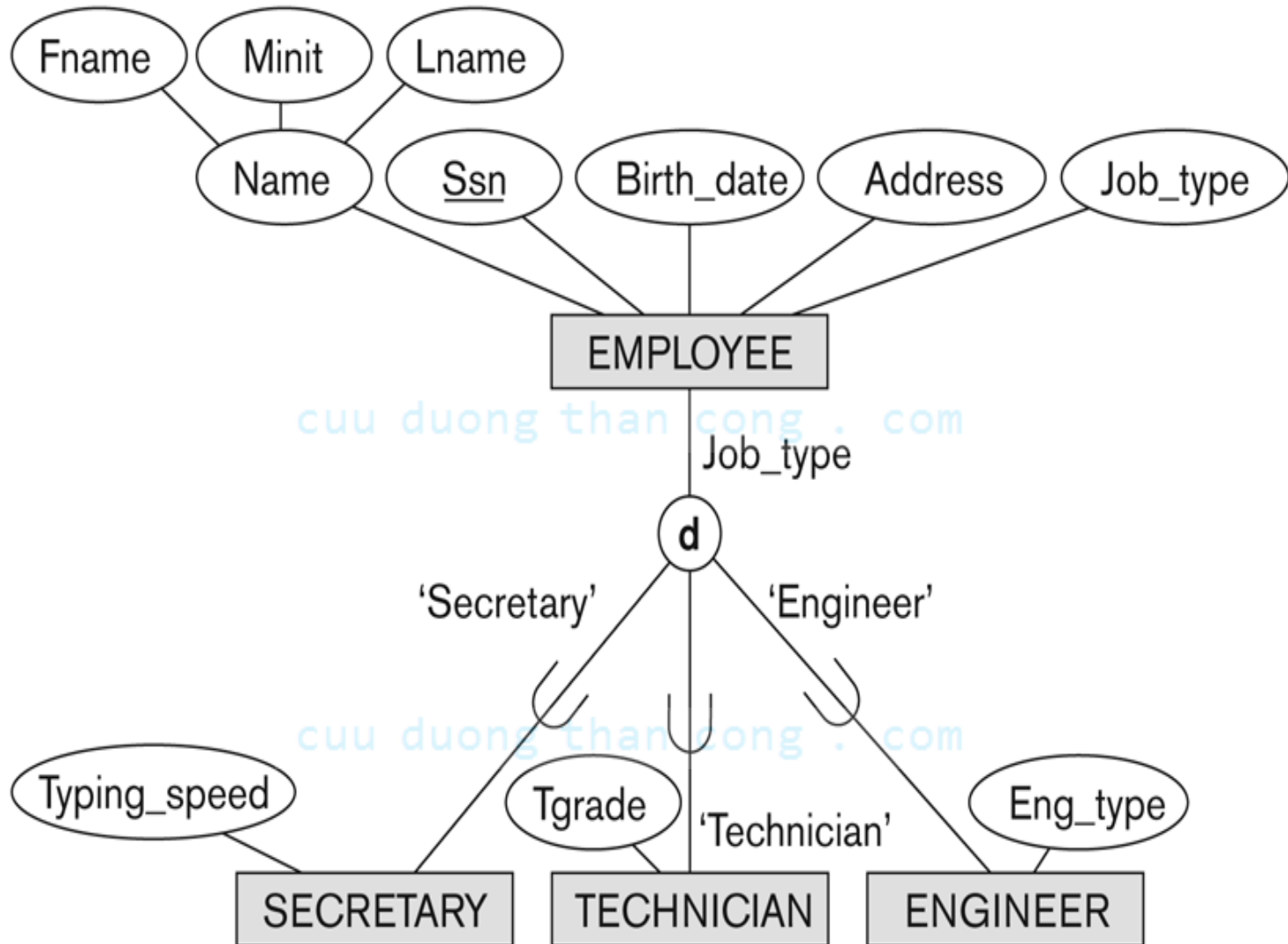
## ■ Specialization

- ❑ Process of defining a set of subclasses of an entity type, called superclass
- ❑ Defined on the basis of some distinguishing characteristic of the entities in the superclass
- ❑ May have several specializations of the same entity type based on different distinguishing characteristics

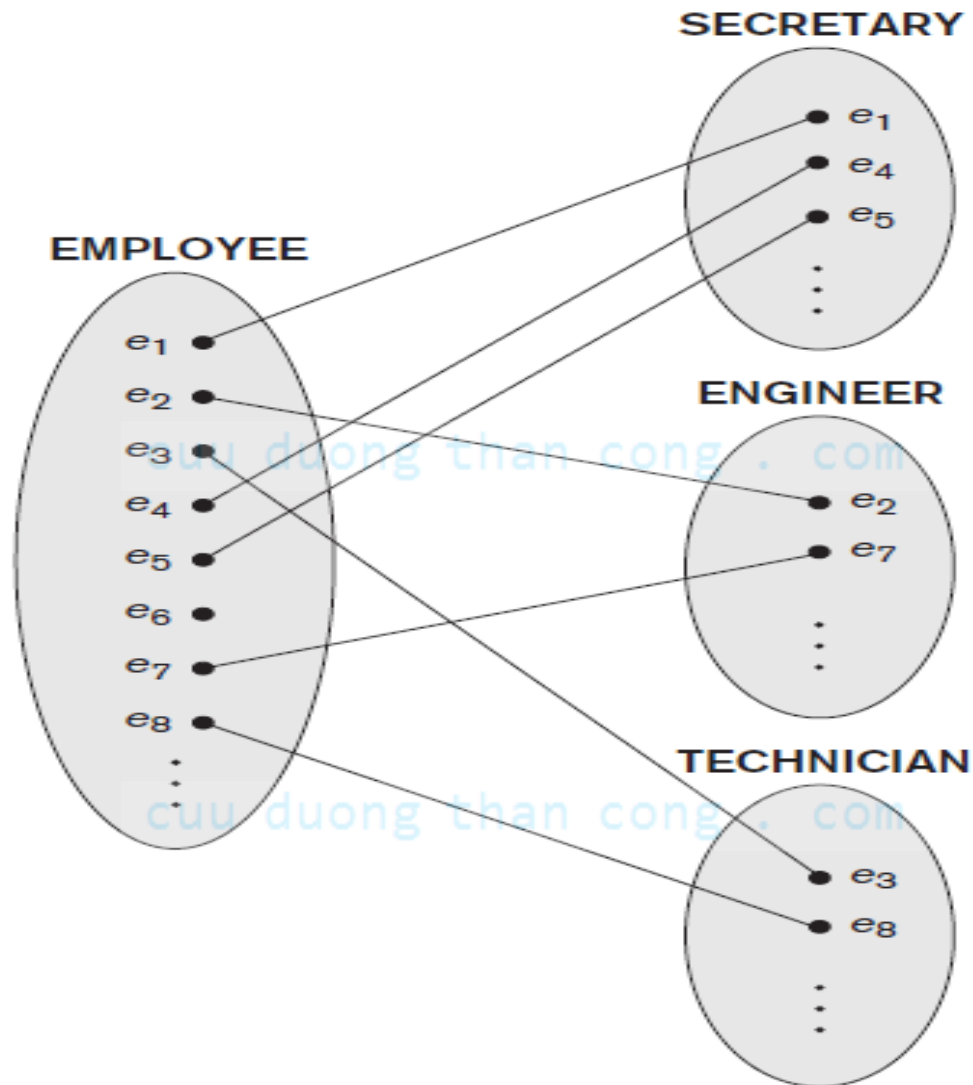
## ■ Subclass can have its own:

- ❑ Specific attributes (local attributes)
- ❑ Specific relationship types

# Example of a Specialization



# Instances of a specialization



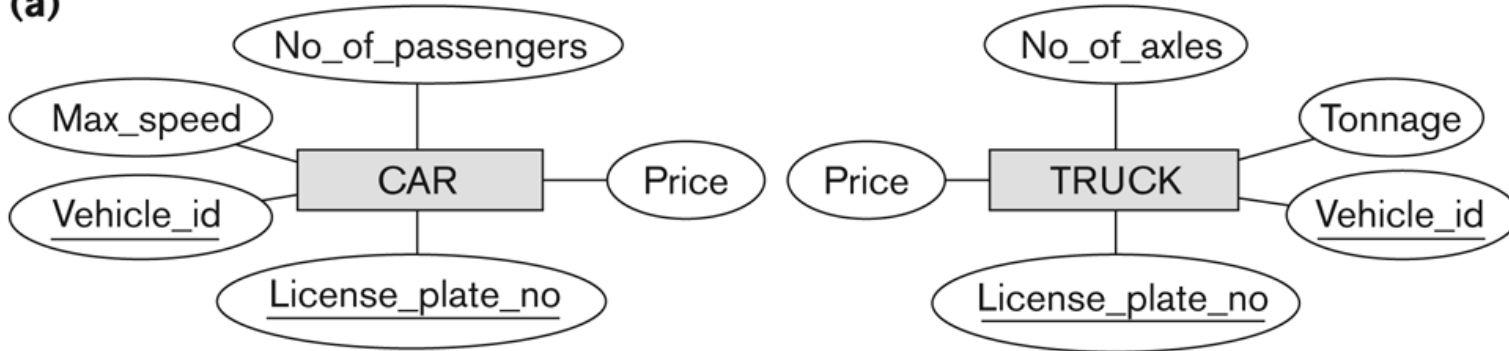
# Generalization

- Reverse process of Specialization
- **Generalize** several entity types which have some common features into a single **superclass**
  - ❑ Original entity types are special subclasses
- **Generalization**
  - ❑ Process of defining a generalized entity type from the given entity types

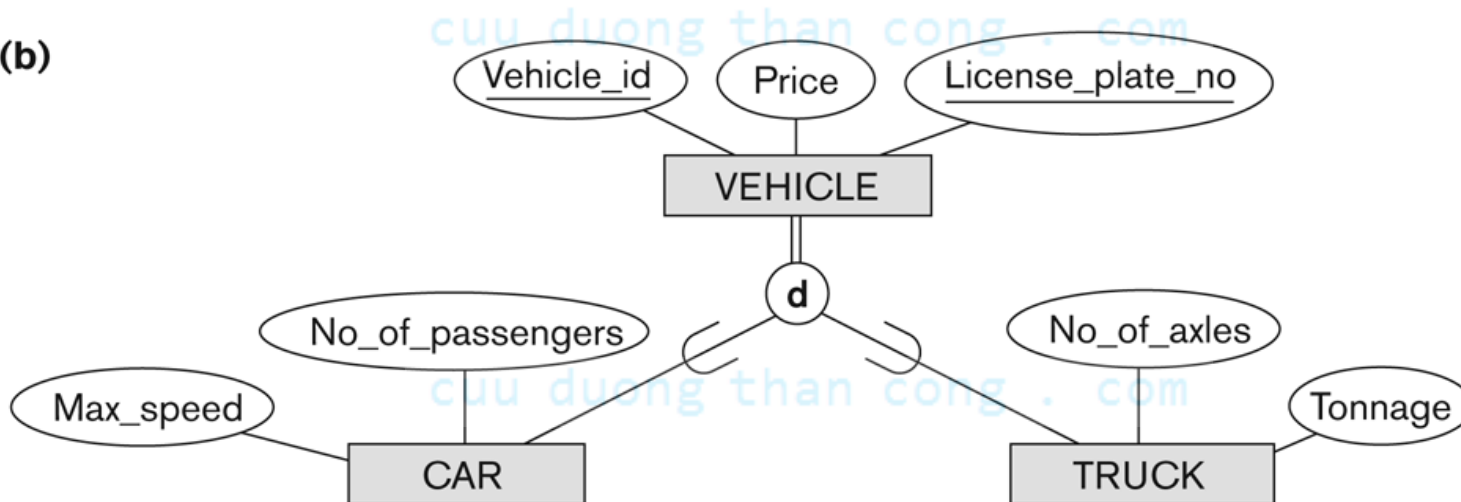


# Example of a Generalization

(a)



(b)



Generalization. (a) Two entity types, CAR and TRUCK.  
(b) Generalizing CAR and TRUCK into the superclass VEHICLE.

# Specialization and Generalization

- Diagrammatic notation sometimes used to distinguish between generalization and specialization
  - Arrow pointing to the generalized superclass represents a generalization
  - Arrows pointing to the specialized subclasses represent a specialization
  - We do not use this notation because it is often subjective as to which process is more appropriate for a particular situation
  - We advocate not drawing any arrows in these situations

# Specialization and Generalization

- Data Modeling with Specialization and Generalization
  - A superclass or subclass represents a set of entities
  - Shown in rectangles in EER diagrams (as are entity types)
  - Sometimes, all entity sets are simply called classes, whether they are entity types, superclasses, or subclasses

# Contents

---

- 
- 1 Introduction to Enhanced-ER Model
  - 2 Subclasses, Superclasses, and Inheritance
  - 3 Specialization and Generalization
  - 4 Constraints and Characteristics of Specialization and Generalization Hierarchies**
  - 5 Categories
  - 6 Design Choices, and Formal Definitions
-

# Constraints and Characteristics of Specialization and Generalization Hierarchies

- Constraints that apply to a single specialization or a single generalization
- Differences between specialization/generalization lattices and hierarchies

cuu duong than cong . com

# Constraints on Specialization and Generalization

- Determine subclass:
  - **Predicate-defined (or condition-defined) subclasses**
  - **Attribute-defined specialization**
  - **User-defined**

# Constraints on Specialization and Generalization

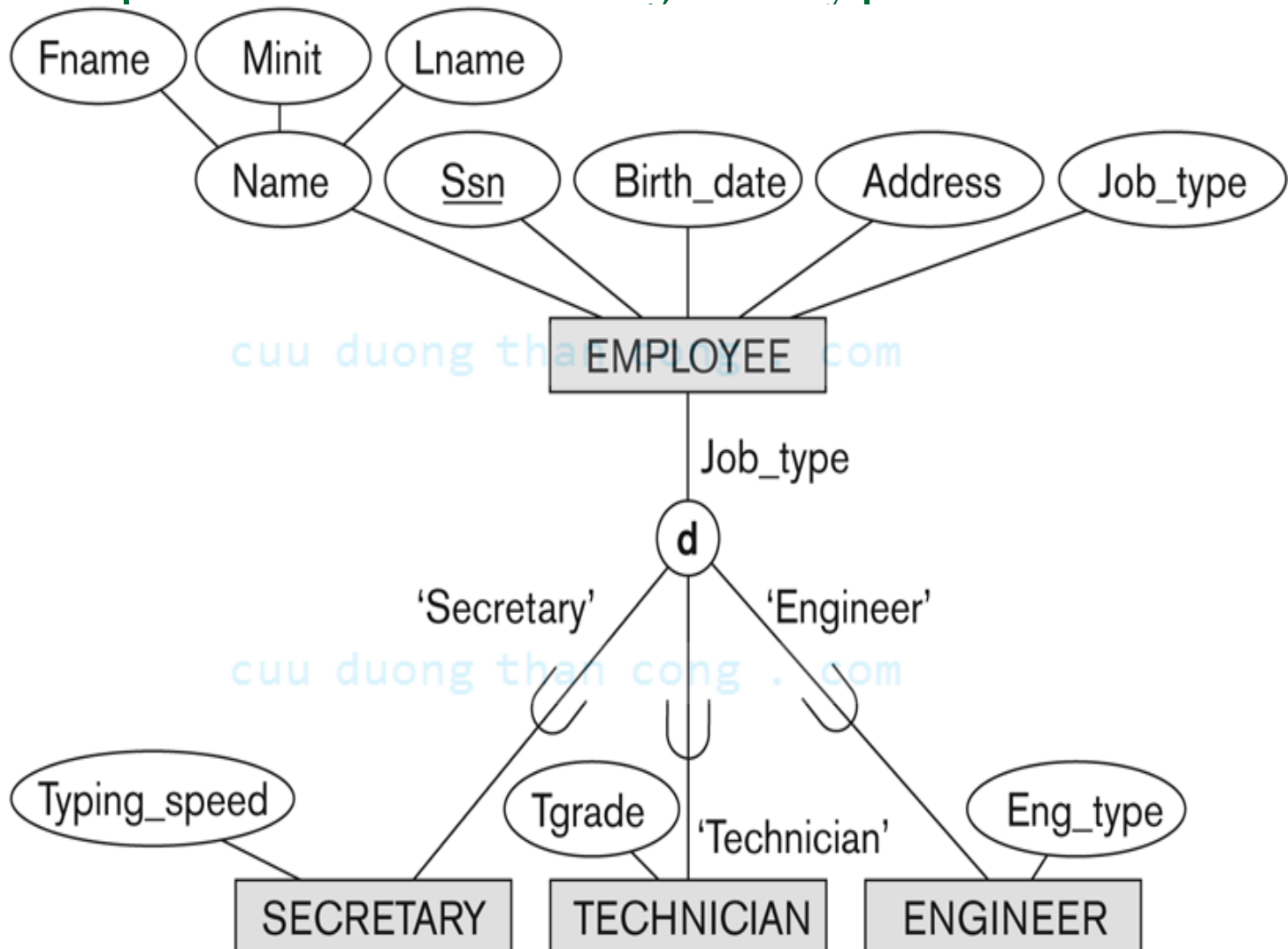
- If we can determine exactly those entities that will become members of each subclass by a condition, the subclasses are called ***predicate-defined (or condition-defined)*** subclasses
  - Condition is a constraint that determines subclass members
  - Display a predicate-defined subclass by writing the predicate condition next to the line attaching the subclass to its superclass

# Constraints on Specialization and Generalization

- If all subclasses in a specialization have membership condition on same attribute of the superclass, specialization is called an ***attribute defined-specialization***
  - Attribute is called the defining attribute of the specialization
  - Ex: JobType is the defining attribute of the specialization {SECRETARY, TECHNICIAN, ENGINEER} of EMPLOYEE



# EER diagram notation for an attribute-defined specialization on JobType



# Constraints on Specialization and Generalization

- If no condition determines membership, the subclass is called ***user-defined***
  - Membership in a subclass is determined by the database users by applying an operation to add an entity to the subclass
  - Membership in the subclass is specified individually for each entity in the superclass by the user

# Constraints on Specialization and Generalization

- Two basic conditions apply to a specialization/ generalization: **disjointness** and **completeness** constraints
- **Disjointness constraint**
  - May be **disjointed** or **overlap**
- **Completeness (or totalness) constraint**
  - May be **total** or **partial**
- Disjointness and completeness constraints are independent

# Constraints on Specialization and Generalization

## ■ Disjointness Constraint:

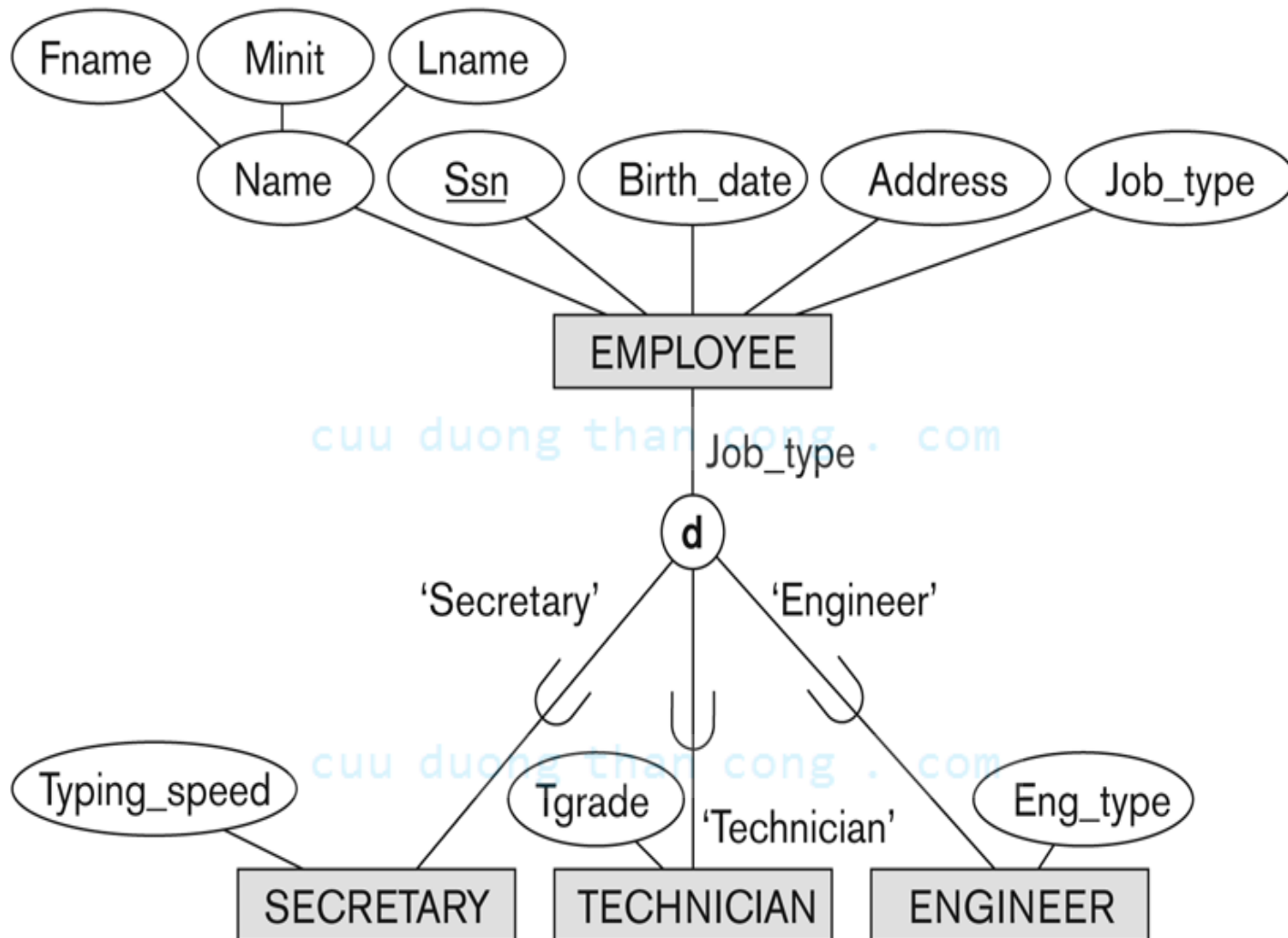
- ❑ Specifies that the subclasses of the specialization must be disjoint (an entity can be a member of at most one of the subclasses of the specialization)
- ❑ Specified by **d** in EER diagram
- ❑ If not disjointed, overlap; that is the same entity may be a member of more than one subclass of the specialization
- ❑ Specified by **o** in EER diagram

# Constraints on Specialization and Generalization

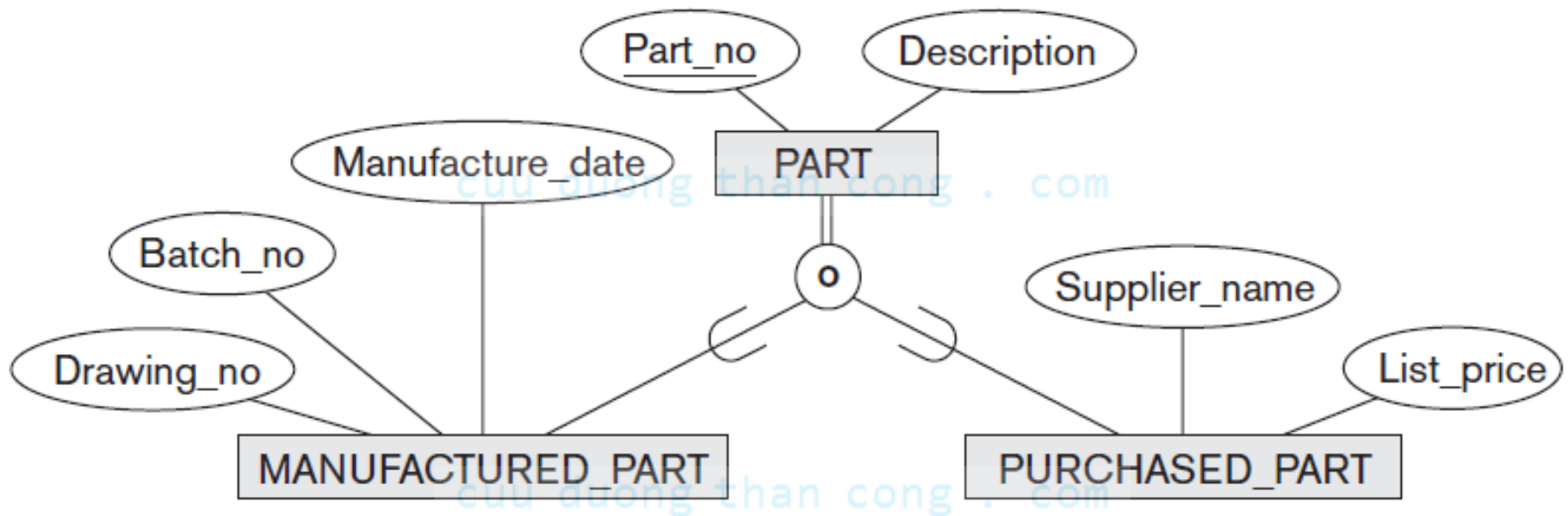
## ■ **Completeness Constraint:**

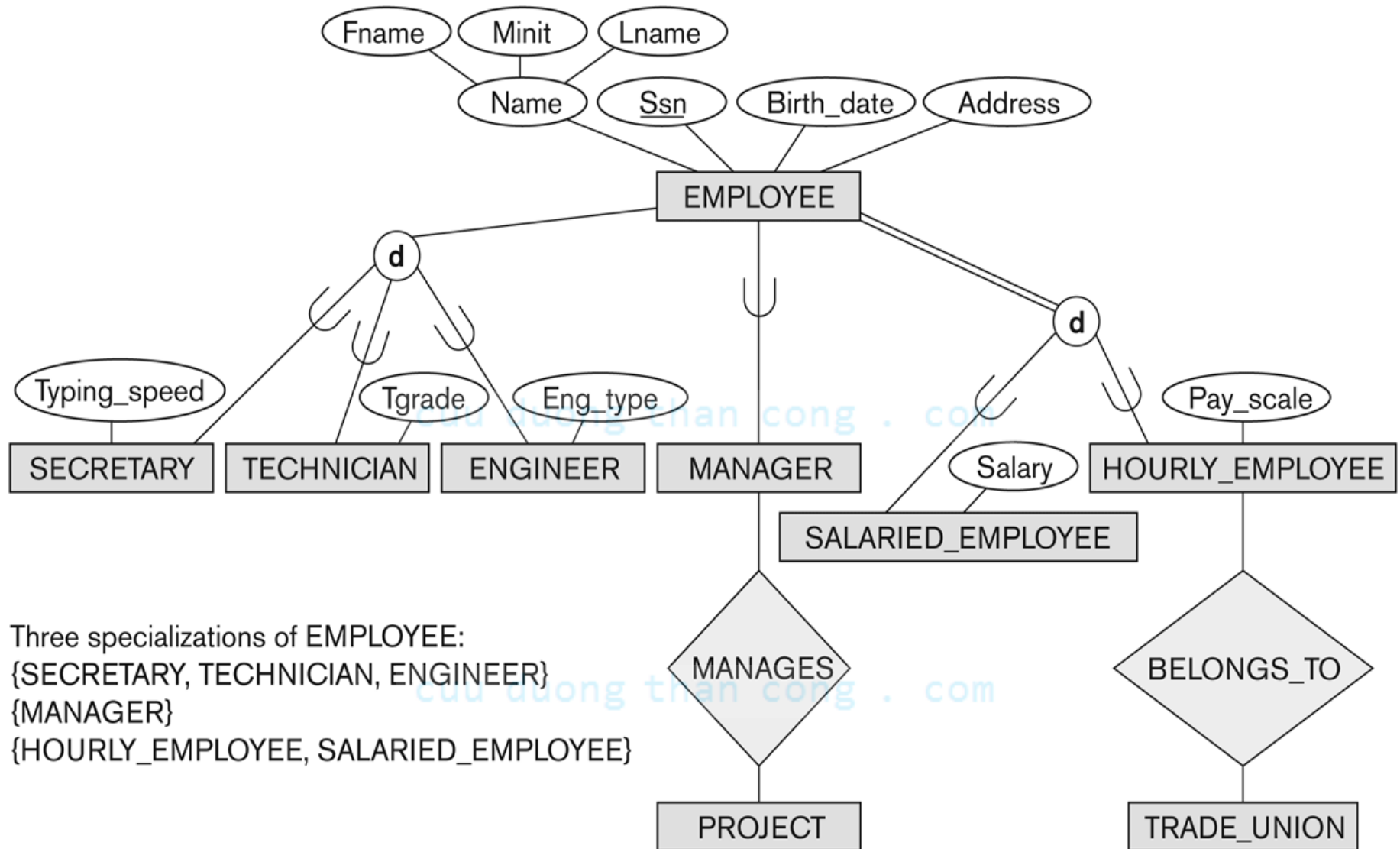
- ❑ **Total** specifies that every entity in the superclass must be a member of some subclass in the specialization/ generalization: Shown in EER diagrams by a **double line**
- ❑ **Partial** allows an entity not to belong to any of the subclasses: Shown in EER diagrams by a **single line**

# Example of Disjoint Partial Specialization



# Example of Overlapping Total Specialization







# Constraints on Specialization and Generalization

- Hence, we have four types of specialization / generalization:
  - Disjoint, total
  - Disjoint, partial
  - Overlapping, total
  - Overlapping, partial
- Note: Generalization is usually total because the superclass is derived from the subclasses

# Specialization and Generalization Hierarchies and Lattices

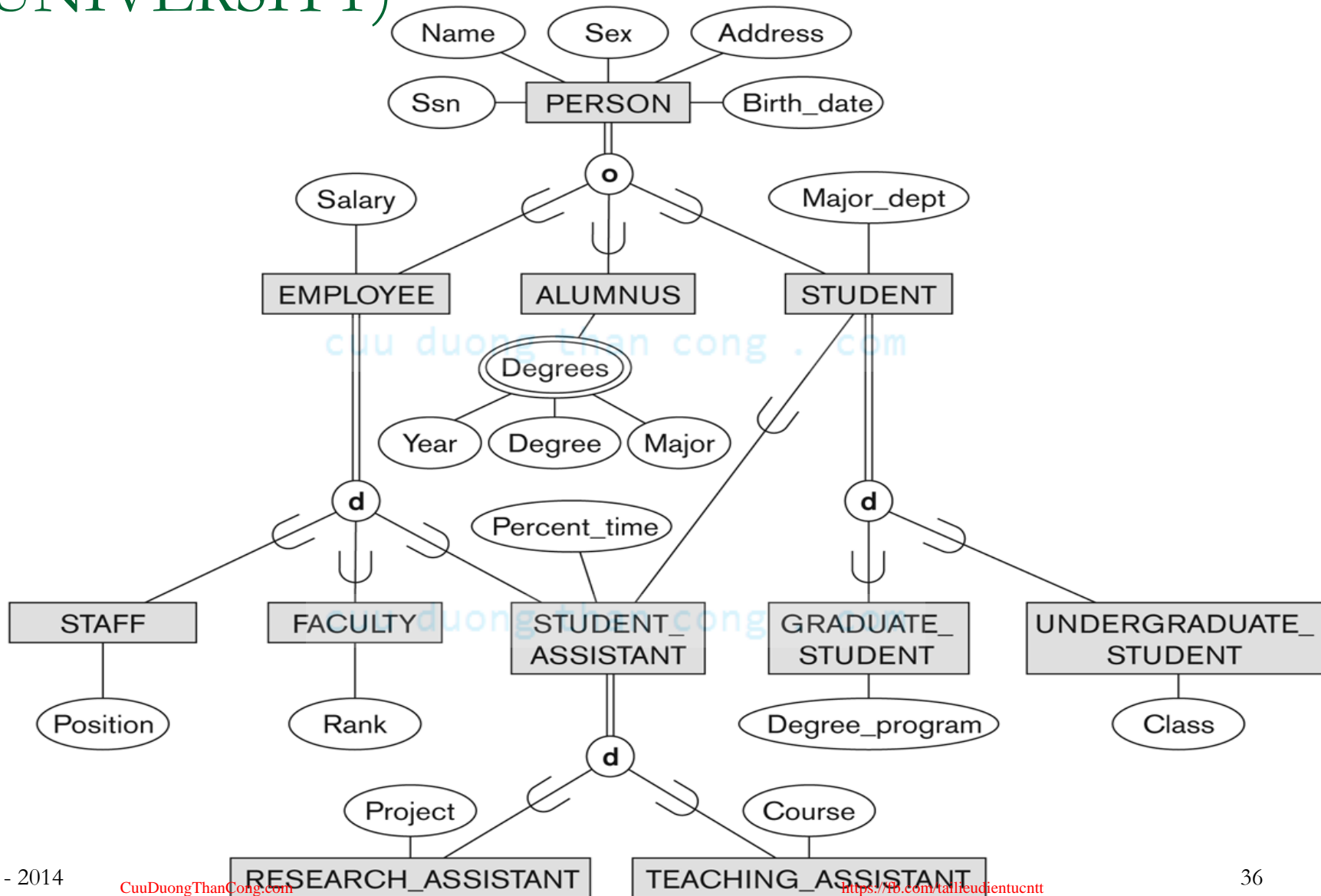
- A subclass may itself have further subclasses specified on it, forming a hierarchy or a lattice.
- **Specialization hierarchy**
  - ❑ Every subclass participates as a subclass in **only one** class/subclass relationship
  - ❑ Results in a **tree** structure or strict hierarchy
  - ❑ **Single inheritance**
- **Specialization lattice**
  - ❑ Subclass can be a subclass in **more than one** class/subclass relationship
  - ❑ **Multiple inheritance**

# Specialization and Generalization

## Hierarchies and Lattices

- In a lattice or hierarchy, a subclass inherits attributes not only of its direct superclass, but also of all its predecessor superclasses
- A subclass with more than one superclass is called a **shared subclass**
- In case of multiple inheritance, if attribute (or relationship) originating in the same superclass inherited more than once via different paths in lattice
  - Included only **once** in shared subclass

# Specialization / Generalization Lattice Example (UNIVERSITY)



# Utilizing Specialization and Generalization in Refining Conceptual Schemas

## ■ Specialization process

- Start with entity type then define subclasses by successive specialization
- **Top-down** conceptual refinement process

## ■ **Bottom-up** conceptual synthesis

- Involves generalization rather than specialization

# Contents

---

- 1 Introduction to Enhanced-ER Model
- 2 Subclasses, Superclasses, and Inheritance
- 3 Specialization and Generalization
- 4 Constraints and Characteristics of Specialization and Generalization Hierarchies
- 5 Categories**
- 6 Design Choices, and Formal Definitions

# Categories

## ■ Category or Union type

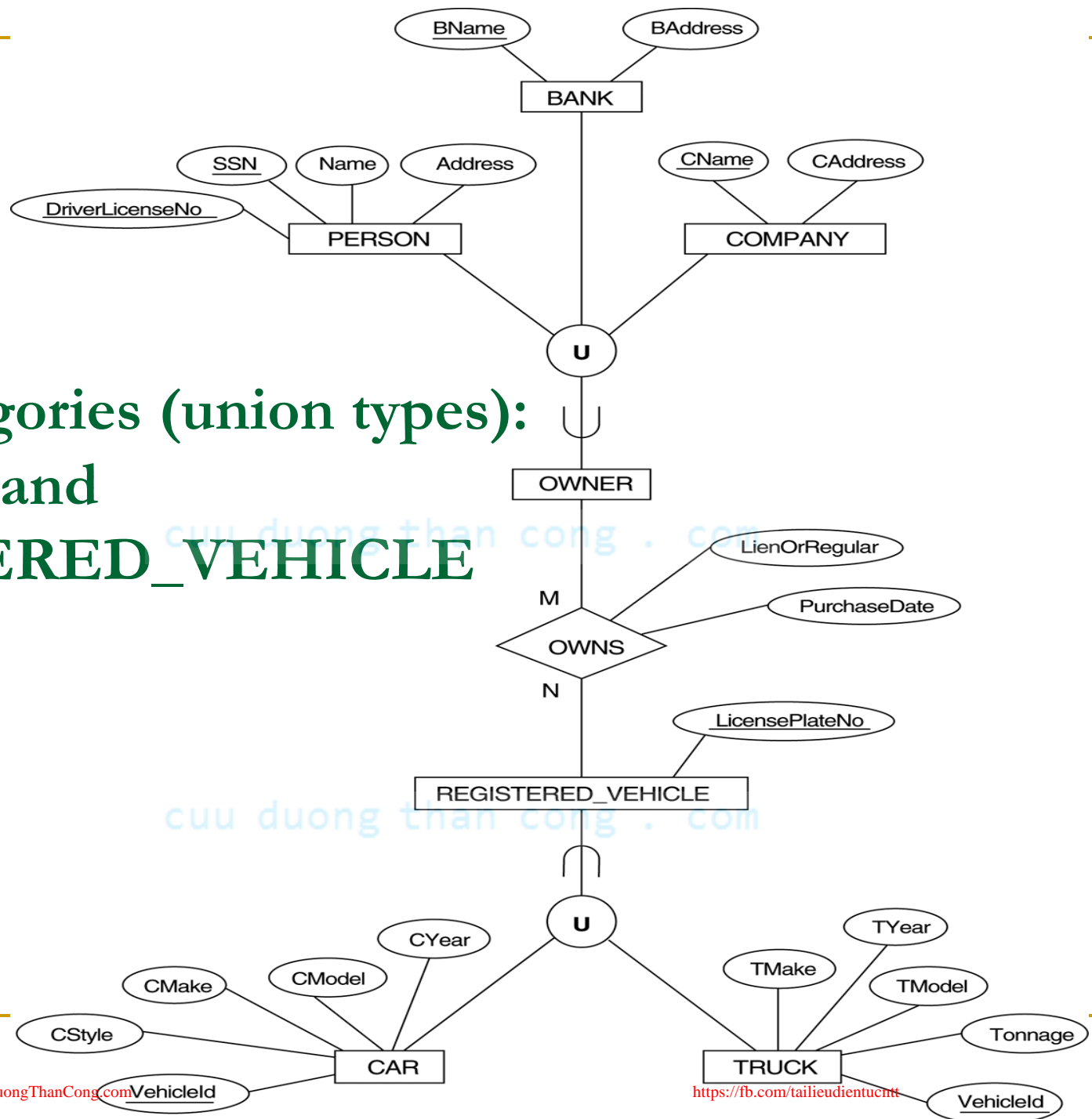
- ❑ Represents a single superclass/subclass relationship with more than one superclass
- ❑ Subclass represents a collection of objects that is a subset of the UNION of distinct entity types
- ❑ Attribute inheritance works more selectively
- ❑ Category can be **total** or **partial**

## ■ Some modeling methodologies do not have union types

# Categories

- Example: Database for vehicle registration, vehicle owner can be a person, a bank (holding a lien on a vehicle) or a company.
  - Category (subclass) OWNER is a subset of the union of the three superclasses COMPANY, BANK, and PERSON
  - A category member must exist in **only one** of its superclasses
- Note: The difference from shared subclass, which is a subset of the intersection of its superclasses (shared subclass member must exist in all of its superclasses)





Two categories (union types):  
**OWNER** and  
**REGISTERED\_VEHICLE**

# Contents

---

- 1 Introduction to Enhanced-ER Model
- 2 Subclasses, Superclasses, and Inheritance
- 3 Specialization and Generalization
- 4 Constraints and Characteristics of Specialization and Generalization Hierarchies
- 5 Categories
- 6 Design Choices, and Formal Definitions**

# Design Choices for Specialization/Generalization

- Many specializations and subclasses can be defined to make the conceptual model accurate
- If subclass has few specific attributes and no specific relationships
  - Can be merged into the superclass

# Design Choices for Specialization/Generalization

- If all the subclasses of a specialization/generalization have few specific attributes and no specific relationships
  - Can be merged into the superclass
  - Replace with one or more type attributes that specify the subclass or subclasses that each entity belongs to

cuu duong than cong . com

# Design Choices for Specialization/Generalization

- Union types and categories should generally be avoided
- Choice of disjoint/overlapping and total/partial constraints on specialization/generalization
  - Driven by rules in miniworld being modeled

cuu duong than cong . com

# Formal Definitions for the EER Model Concepts

## ■ Class C

- ❑ Set or collection of entities
- ❑ Includes any of the EER schema constructs of group entities
- ❑ Can be entity type, subclass, superclass, or category
- ❑ Note: The definition of *relationship type* in ER/EER should have 'entity type' replaced with 'class' to allow relationships among classes in general

# Formal Definitions for the EER Model Concepts

## ■ Subclass $S$ :

- ❑ Inherits all the attributes and relationship of a class  $C$
- ❑ Set of entities must always be a subset of the set of entities of the other class  $C$ :  $S \subseteq C$
- ❑  $C$  is called the superclass of  $S$
- ❑ A superclass/subclass relationship exists between  $S$  and  $C$

# Formal Definitions for the EER Model Concepts

- **Specialization Z:**  $Z = \{S1, S2, \dots, Sn\}$  is a set of subclasses with same superclass  $G$ ;
  - $G/Si$  is a superclass/subclass relationship,  $i=1..n$
  - $G$  is called a generalization of the subclasses  $\{S1, S2, \dots, Sn\}$
  - $Z$  is total if:
    - $S1 \cup S2 \cup \dots \cup Sn = G$ ;
    - Otherwise,  $Z$  is partial
  - $Z$  is disjoint if:
    - $Si \cap Sj$  empty-set for  $i \neq j$ ;
    - Otherwise,  $Z$  is overlapping



# Formal Definitions for the EER Model Concepts

- **Generalization:**
  - Generalized entity type or superclass
- Subclass  $S$  of  $C$  is **predicate defined** if predicate (condition)  $p$  on attributes of  $C$  is used to specify membership in  $S$ ; that is,  $S = C[p]$ , where  $C[p]$  is the set of entities in  $C$  that satisfy condition  $p$
- A subclass not defined by a predicate is called **user-defined**

# Formal Definitions for the EER Model Concepts

## ■ Category or UNION type T

- Class that is a subset of the union of n defining superclasses

$$D1, D2, \dots, Dn, n > 1: T \subseteq (D1 \cup D2 \cup \dots \cup Dn)$$

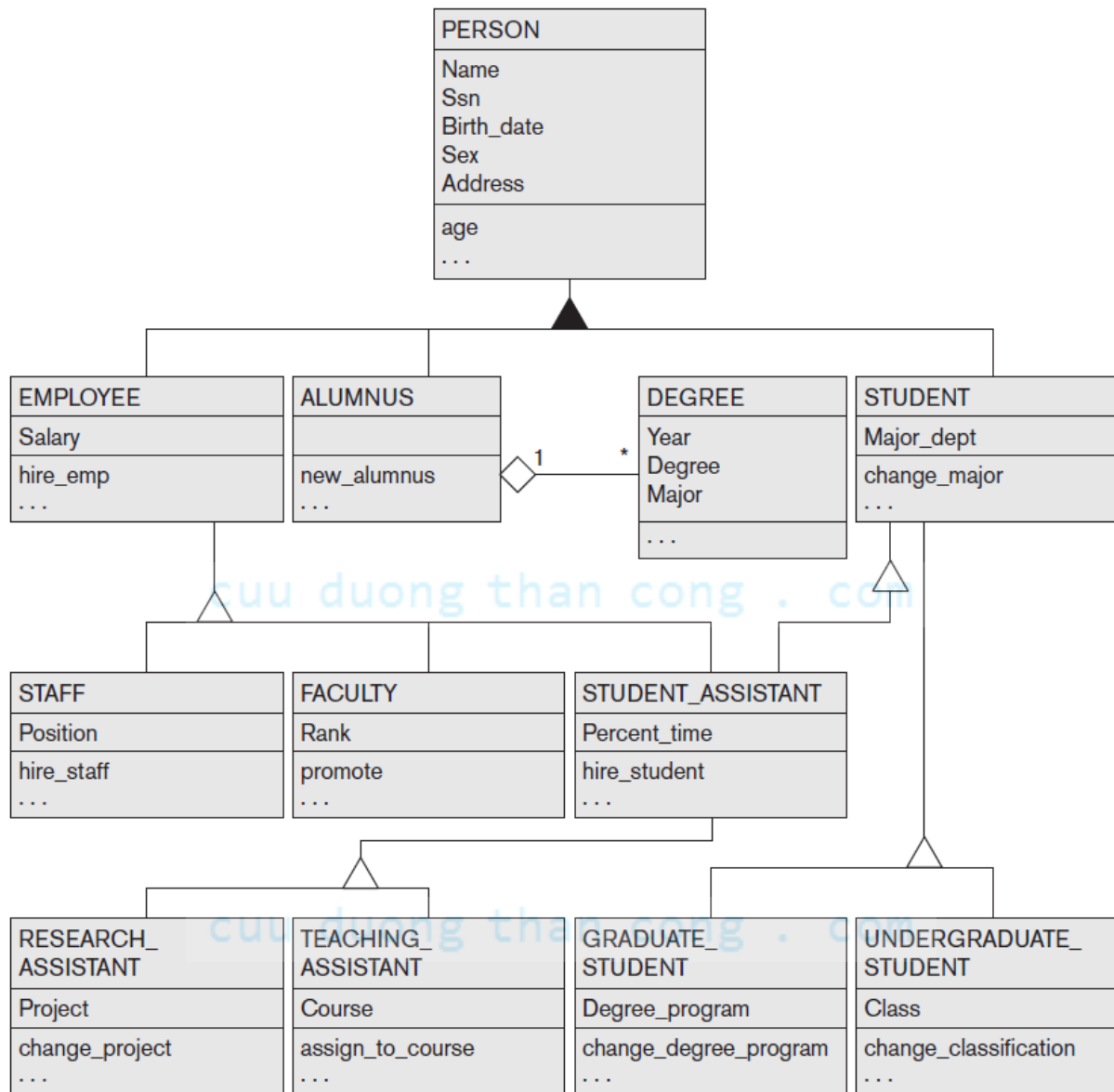
- Can have a predicate  $p_i$  on the attributes of  $D_i$  to specify entities of  $D_i$  that are members of T.
- If a predicate is specified on every  $D_i$ :  $T = (D1[p1] \cup D2[p2] \cup \dots \cup Dn[pn])$

## ■ Relationship type

- Any class can participate in a relationship

# Example of Other Notation

- Representing specialization and generalization in UML class diagrams
  - Basic notation
    - See Figure 8.10
  - Base class
    - Root superclass
  - Leaf classes
    - Subclasses (leaf nodes)



**Figure 8.10**

A UML class diagram corresponding to the EER diagram in Figure 8.7, illustrating UML notation for specialization/generalization.

# Contents

---

- 
- 1 Introduction to Enhanced-ER Model
  - 2 Subclasses, Superclasses, and Inheritance
  - 3 Specialization and Generalization
  - 4 Constraints and Characteristics of Specialization and Generalization Hierarchies
  - 5 Categories
  - 6 Design Choices, and Formal Definitions
-

Q & A

cuu duong than cong . com

# Exercise 1: University Database

**Modify the UNIVERSITY diagram by classifying:**

- **MODULES** as either:
  - ❑ UNDERGRAD\_MODULES or
  - ❑ GRAD\_MODULES
- **LECTURERS** as either:
  - ❑ JUNIOR\_PROFESSORS or
  - ❑ SENIOR\_PROFESSORS.
- Include appropriate attributes for these new entity types.
- Establish relationships indicating that **junior lecturers teach undergraduate modules** while **senior lecturers teach graduate modules**.

# Exercise 2

- A non-profit organization depends on a number of different types of **persons** for its successful operation. The organization is interested in the following attributes for all of these persons: Social Security Number, Name, Address, City, State and Telephone.
- Three types of persons are of interest: **employees, volunteers and donors**. Employees have only a Date\_Hired attribute, and volunteers have only a Skill attribute. Donors have a relationship (named **Donates**) with an **Item**. A donor must have donated one or more Items, and an Item can only be donated by one donor. Attributes of item includes an identity and a description.
- There are persons other than employees, volunteers and donors who are of interest to the organization, so a person does not have to belong to one of these groups. A person may also belong to one or more of these groups, at any one time.



# Exercise 3

- Attic Antiques buys and sells one-of-a-kind antiques of all kinds (e.g. furniture, china, clothing, etc.). Each item is uniquely identified by a serial number, and is also characterized by asking price and condition.
- Attic works with several individuals who sell and buy items from the store. Some clients only sell items to Attic, others only buy items, and some both buy and sell.
- Attic keeps track of its clients through the assigning of client numbers. They also keep track of clients' names and addresses.
- When Attic sells an item to a client, they need to keep track of the actual selling price, the date of the sale, and the sales tax. When Attic buys an item, they wish to track the purchase cost, condition at the time of purchase, and the date.

# Review questions

- 1) Discuss the two main types of **constraints on specializations and generalizations**.
- 2) What is the difference between a ***specialization hierarchy*** and a ***specialization lattice***?
- 3) How does a ***category*** differ from a regular ***shared subclass***? What is a category used for? Illustrate your answer with examples.