

# 204222 - Fundamentals of Database Systems

## Chapter 8

# The Enhanced Entity-Relationship (EER) Model

Adapted for 204222

by Areerat Trongratsameethong

# Chapter 8 Outline

- Subclasses, Superclasses, and Inheritance
- Specialization and Generalization
- Constraints and Characteristics of Specialization and Generalization Hierarchies
- Modeling of UNION Types Using Categories
- A Sample UNIVERSITY EER Schema, Design Choices, and Formal Definitions
- Example of Other Notation: Representing Specialization and Generalization in UML Class Diagrams

# The Enhanced Entity-Relationship (EER) Model

- **Enhanced ER (EER) model**
  - Created to design more accurate database schemas: สร้างขึ้นมาเพื่อให้การออกแบบฐานข้อมูลมีความหมายที่ถูกต้องแม่นยำมากขึ้น ช่วยสะท้อนคุณสมบัติของข้อมูล และข้อจำกัดของข้อมูลได้ชัดเจนมากขึ้น
    - Reflect the data properties and constraints more precisely
  - More complex requirements than traditional applications: สำหรับฐานข้อมูลของ application ที่มีความซับซ้อนมากกว่า application แบบดั้งเดิม

# Subclasses, Superclasses, and Inheritance

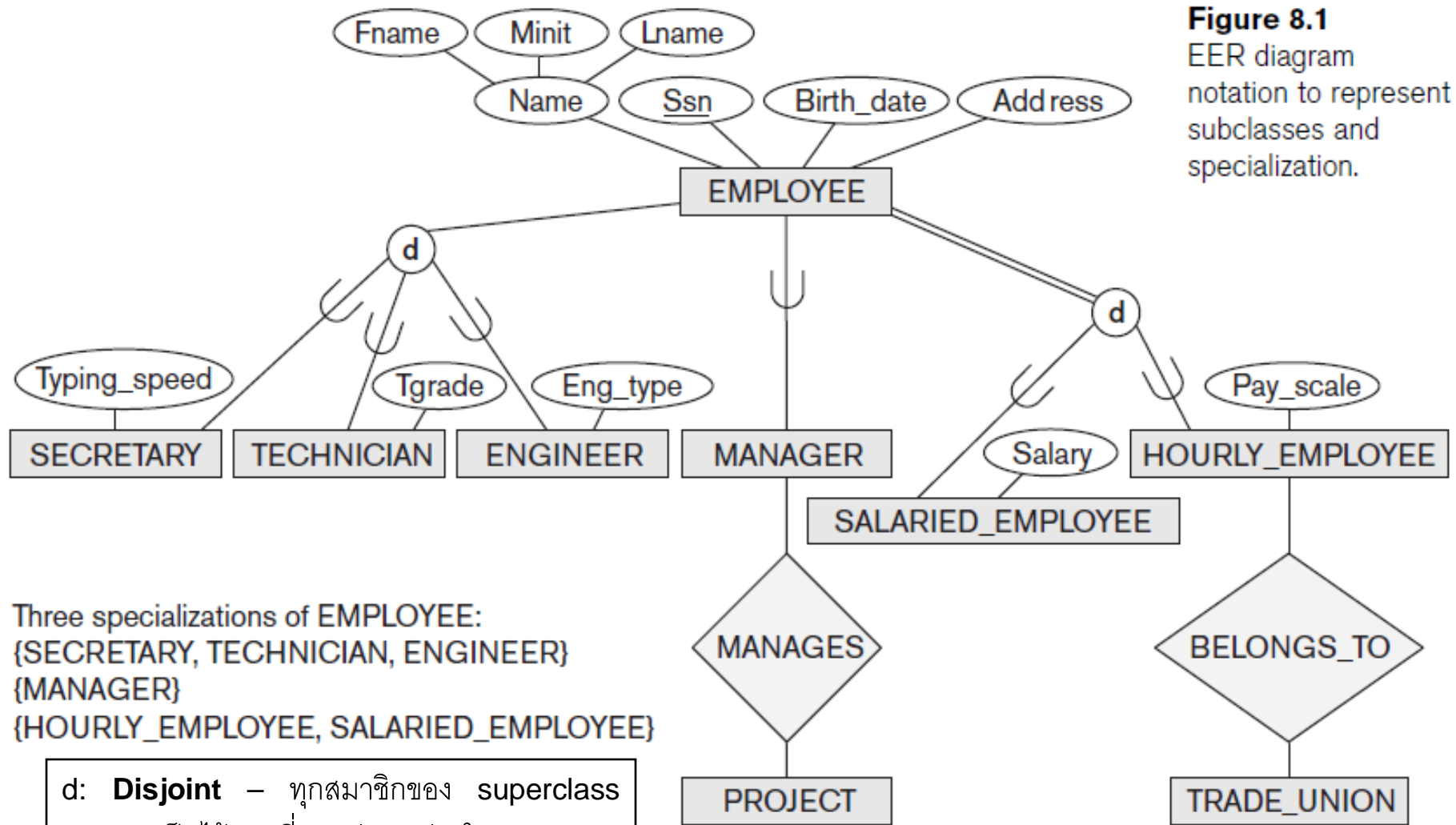
- EER model includes all modeling concepts of the ER model: EER รวม model ทั้งหมดของ ER ไว้ และมี ส่วนที่เพิ่มเติมขึ้นมาสำหรับข้อมูลที่มีความซับซ้อนมากยิ่งขึ้น
- In addition, EER includes:
  - **Subclasses and superclasses**
  - **Specialization and generalization**
  - **Category or union type**
  - **Attribute and relationship inheritance**

# Subclasses, Superclasses, and Inheritance (cont'd.)

- **Enhanced ER or EER diagrams**
  - Diagrammatic technique for displaying these concepts in an EER schema: EER Diagram ใช้อธิบาย concepts ที่อยู่ใน EER schema
- **Subtype or subclass of an entity type**
  - Subgroupings of entities that are meaningful
  - Represented explicitly because of their significance to the database application
  - ในบางครั้ง ภายใน **entity** อาจประกอบด้วยข้อมูลกลุ่มย่อย ที่มี ความหมายชัดเจน และมีความสำคัญในการนำเสนอข้อมูลดังกล่าวให้ชัดเจน ในบาง Database Application

# Subclasses, Superclasses, and Inheritance (cont'd.)

- Terms for relationship between a superclass and any one of its subclasses: ศัพท์ที่เกี่ยวข้องที่ใช้ใน EER Diagram
  - **Superclass/subclass** or
  - **Supertype/subtype** or
  - **Class/subclass** relationship
- **Type inheritance**
  - Subclass entity inherits all attributes and relationships of superclass
  - **Subclass** สืบทอดทุกคุณสมบัติ และทุกความสัมพันธ์ของ **superclass** (สามารถสืบทอดได้หลายชั้น)



**Figure 8.1**  
EER diagram notation to represent subclasses and specialization.

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

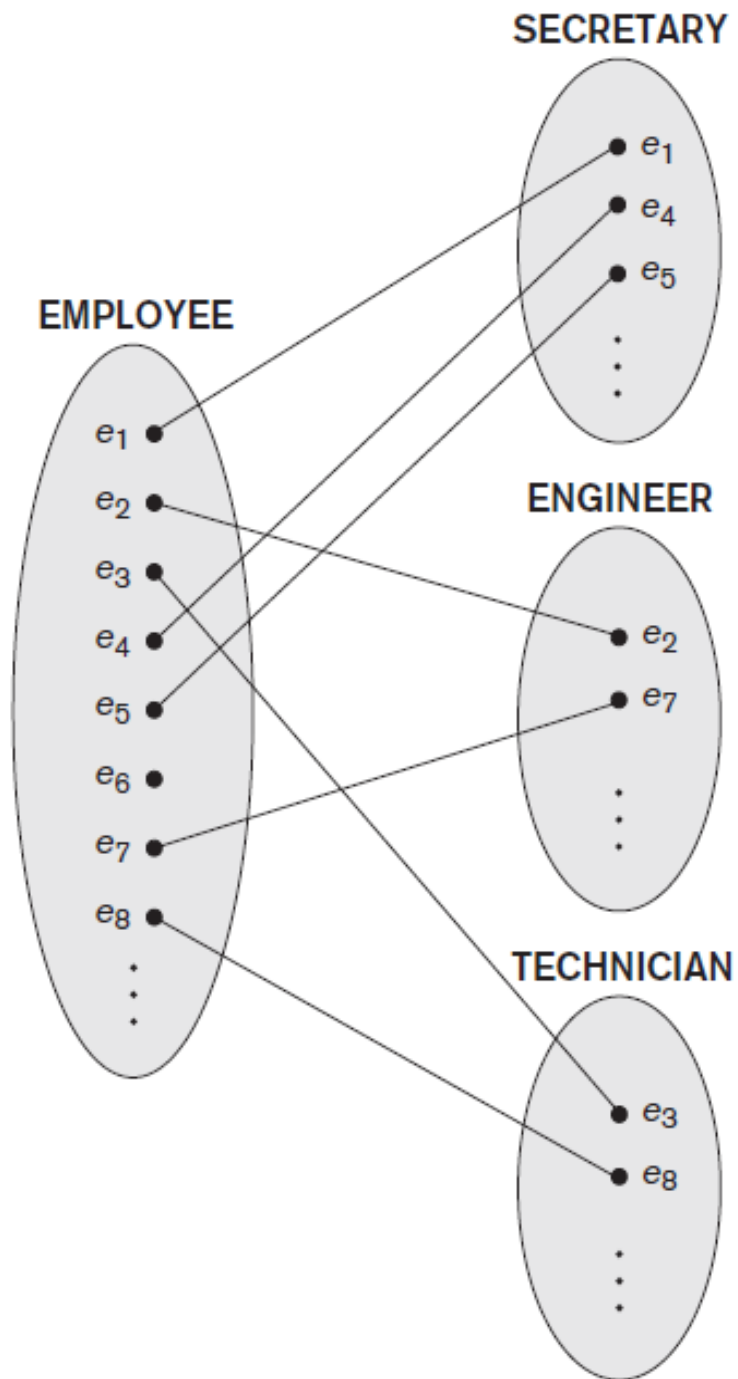
**d: Disjoint** - ทุกสมาชิกของ superclass สามารถเป็นได้มากที่สุดแค่ 1 อย่างใน subclass เช่น พนักงานเป็นได้มากที่สุดแค่ 1 ประเภทใน 3 ประเภท คือ SECRETARY, TECHNICIAN, หรือ ENGINEER

**Note:** MANAGES and BELONGS\_TO → Specific Relationship Type

# Specialization and Generalization

- **Specialization:** สิ่งที่แตกต่างกัน ลักษณะพิเศษ หรือลักษณะเฉพาะ
  - Process of defining a set of subclasses of an entity type
  - Defined on the basis of some distinguishing characteristic of the entities in the superclass
- Subclass can define: subclass สามารถถูกกำหนด attribute และ relationship ที่เฉพาะได้
  - **Specific attributes**
  - **Specific relationship types**





**Figure 8.2**  
Instances of a specialization.

Data of Secretary = Data of Employee (General Data) +  
Specific Data of Secretary

(Specific Data for Secretary e.g. Typing Speed)

# Specialization and Generalization (cont'd.)

- Certain attributes may apply to some but not all entities of the superclass: บาง attribute อาจจะเป็นคุณสมบัติเฉพาะของ subclass
- Some relationship types may be participated in only by members of the subclass: ความสัมพันธ์บางอย่างอาจจะเป็นความสัมพันธ์เฉพาะสำหรับบางสมาชิกใน subclass

# Generalization

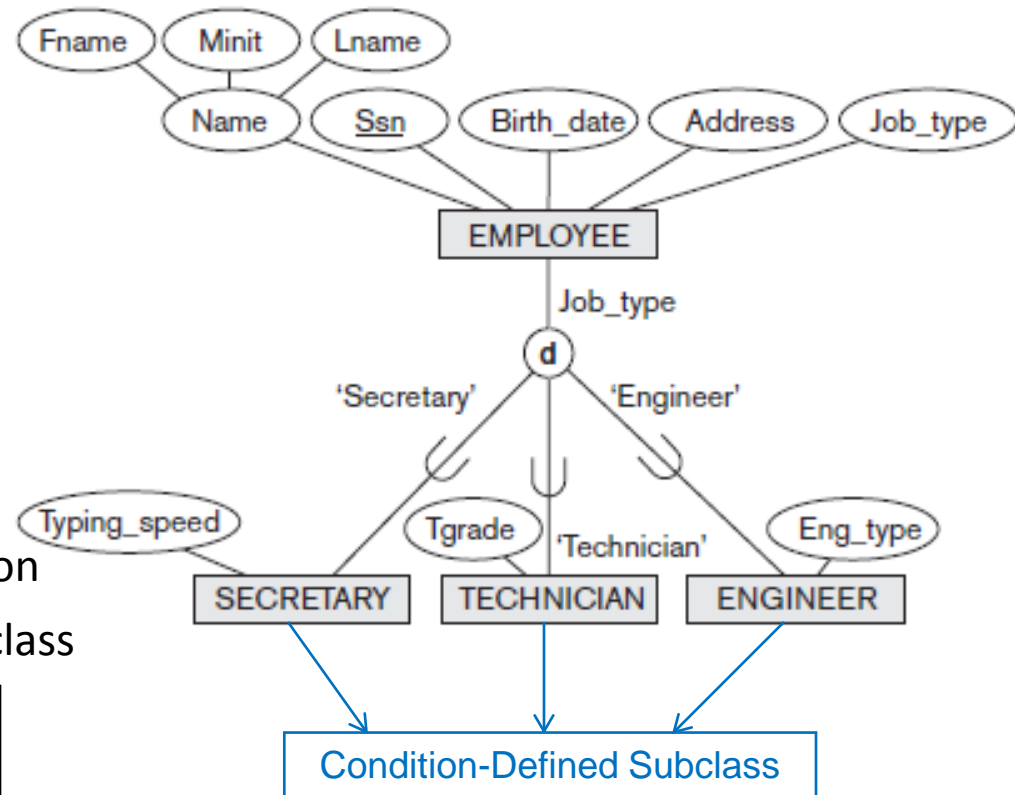
- **Generalize into a single superclass:** หาสิ่งที่มีร่วมกัน (common) เอาไว้ใน superclass แล้วตั้งชื่อใหม่ที่ general สำหรับส่วนที่ต่างกันเอาไว้ใน subclass และใช้ชื่อเดิม เช่น CAR กับ TRUCK เอาส่วนที่มีร่วมกันไปไว้ใน superclass และตั้งชื่อเป็น VEHICLE ส่วนสิ่งที่เหมาะสำหรับ CAR กับ TRUCK ก็ไว้ที่เดิมซึ่งเป็น subclass และใช้ชื่อเดิม
  - Original entity types are special subclasses
- **Generalization**
  - Process of defining a generalized entity type from the given entity types: ขบวนการในการกำหนด generalized entity (superclass) จากหลายๆ entity ที่มีในระบบ

# Constraints and Characteristics of Specialization and Generalization Hierarchies

- Constraints that apply to a single specialization or a single generalization
- Differences between specialization/generalization lattices (**Multiple Inheritance**) and hierarchies (**Single Inheritance**)

# Constraints on Specialization and Generalization

- May be several or one subclass: สิ่งที่ทำให้เกิดคุณสมบัติเฉพาะ specialize อาจเกิดจากคุณลักษณะที่แตกต่างกันของบาง attribute เช่น job\_type ในรูปดังแสดงด้านล่าง
- Determine entity subtype:
  - **Predicate-defined**  
(or condition-defined)  
**subclasses**
  - **Attribute-defined**  
**specialization**
- **User-defined:** do not have a condition for determining membership in a subclass



If all subclasses in a specialization have their membership condition on the same attribute of the superclass, the specialization itself is called an **attribute-defined specialization**, and the attribute is called the **defining attribute of the specialization**

Defining Predicate:  
e.g. Job\_type = 'Secretary' → Secretary Subclass

# Constraints on Specialization and Generalization (cont'd.)

- **Disjointness constraint:** ข้อจำกัดที่เป็นได้บางอย่างใดอย่างหนึ่ง
  - Specifies that the subclasses of the specialization must be disjoint
- **Completeness (or totalness) constraint**
  - May be **total** or **partial**
- Disjointness and completeness constraints are independent

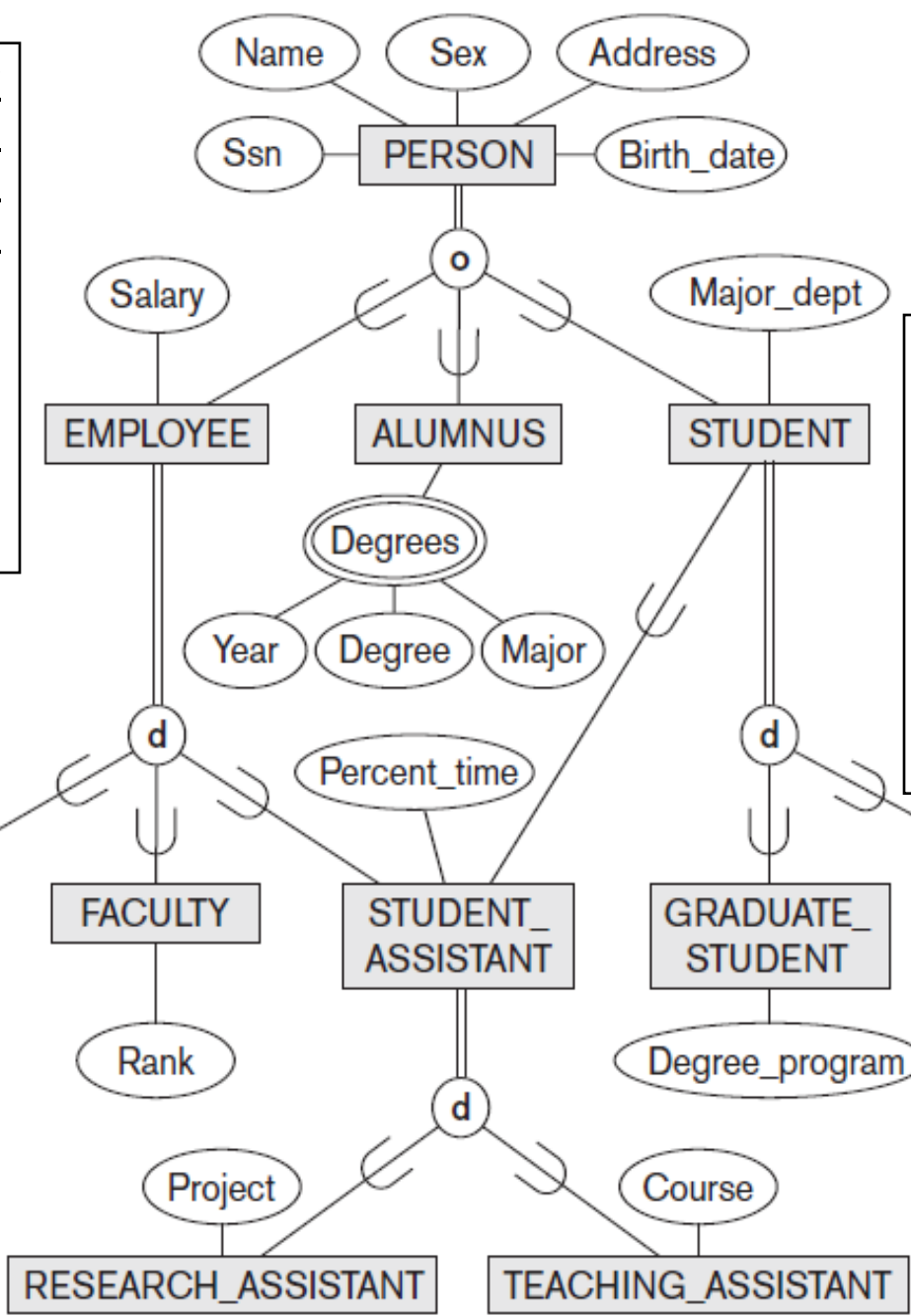
**Total Participation:** Every entity in the superclass must be a member of at least one subclass in the specialization.

**For example,** if every EMPLOYEE must be either an HOURLY\_EMPLOYEE or a SALARIED\_EMPLOYEE.

# Specialization and Generalization Hierarchies and Lattices

- **Specialization hierarchy: Single Inheritance**
  - Every subclass participates as a subclass in only one class/subclass relationship
  - Results in a **tree structure** or **strict hierarchy**
- **Specialization lattice: Multiple Inheritance**
  - Subclass can be a subclass in more than one class/subclass relationship

d: **Disjoint** – every member of the super-class can belong to at most one of the subclasses.  
**For example,** an Animal cannot be a lion and a horse, it must be either a lion, a horse, or a dog.



**Figure 8.7**  
 A specialization lattice with multiple inheritance for a UNIVERSITY database.

**o: Overlapping:** every member of the super-class can belong to more than one of the subclasses.  
**For example,** a book can be a text book, but also a poetry book at the same time.



# Specialization and Generalization Hierarchies and Lattices (cont'd.)

- **Multiple inheritance**

- Subclass with more than one superclass
- If attribute (or relationship) originating in the same superclass inherited more than once via different paths in lattice
  - Included only once in shared subclass

- **Single inheritance**

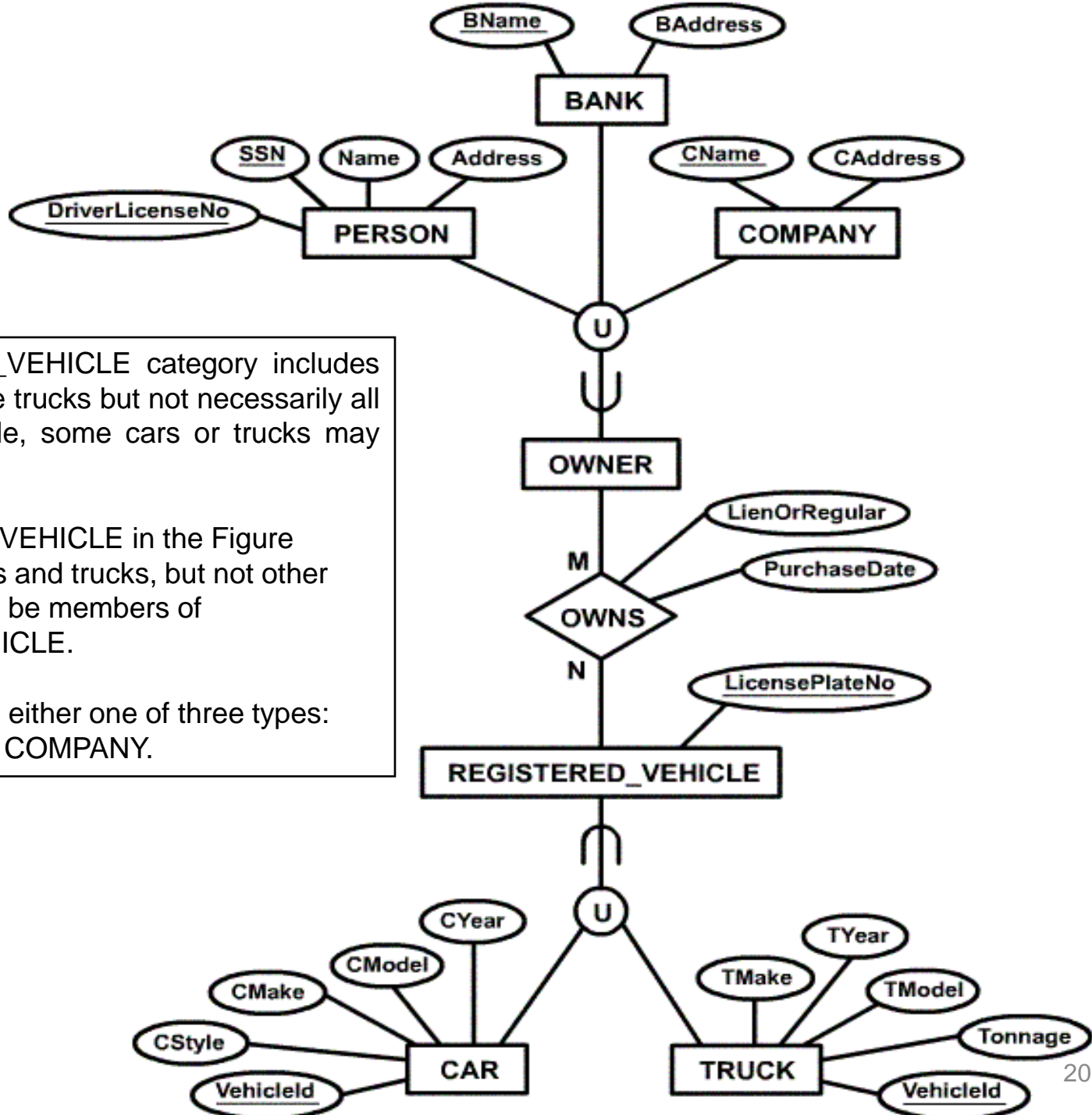
- Some models and languages limited to single inheritance

# 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:** นึกถึงภาพใหญ่ก่อน เช่น ถ้าระบบเกี่ยวกับสัตว์ ให้นึกถึงสัตว์ก่อน แล้วค่อยแยกกลุ่มย่อยของสัตว์แต่ละประเภท เช่น สัตว์บก สัตว์น้ำ สัตว์ปีก เป็นต้น
- **Bottom-up conceptual synthesis:** ในทางกลับกัน ให้นึกถึงสัตว์แต่ละประเภทก่อน เช่น สัตว์บก สัตว์น้ำ สัตว์ปีก แล้วค่อยหาสิ่งที่ร่วมกันของสัตว์เหล่านี้
  - Involves generalization rather than specialization

# Modeling of UNION Types Using Categories

- **Union type** or a **category**
  - 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



The REGISTERED\_VEHICLE category includes some cars and some trucks but not necessarily all of them (for example, some cars or trucks may not be registered).

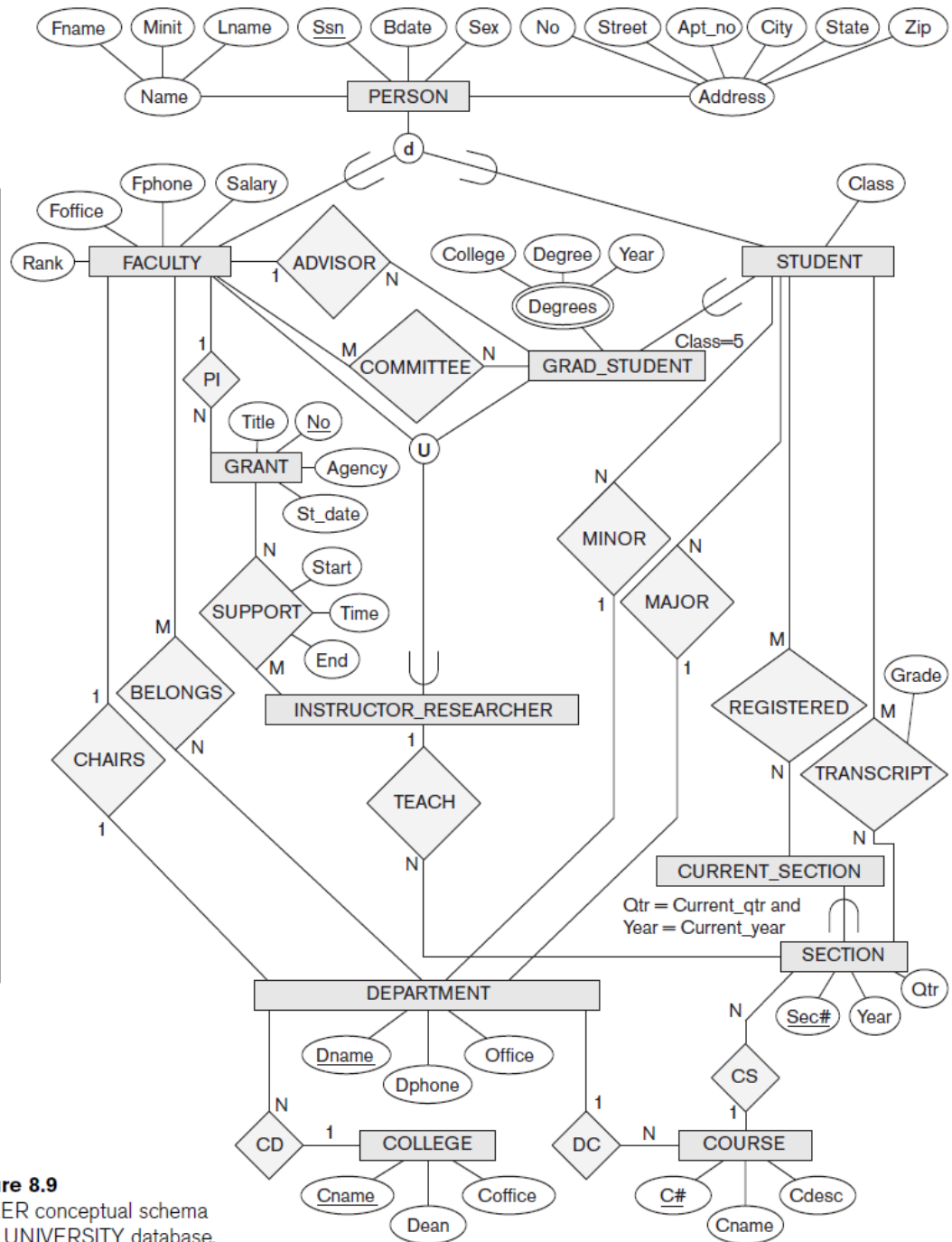
The REGISTERED\_VEHICLE in the Figure implies that only cars and trucks, but not other types of entities, can be members of REGISTERED\_VEHICLE.

The OWNER can be either one of three types: PERSON, BANK, or COMPANY.

# A Sample UNIVERSITY EER Schema, Design Choices, and Formal Definitions

- The UNIVERSITY Database Example
  - UNIVERSITY database
    - Students and their majors
    - Transcripts, and registration
    - University's course offerings

**u: Union** – a subclass/super-class relationship can have more than one super-class, and the subclass inherits from at most one of the super-classes (i.e. the subclass `INSTRUCTOR_RESEARCHER` will inherit the relationships and attributes associated with either `FACULTY` or `Grad_STUDENT`, but not both). Each super class may have different primary keys, or the same primary key. All members of the super-classes are not members of the super-class. For example, a `INSTRUCTOR_RESEARCHER` can be a `FACULTY`, or a `Grad_STUDENT`, but not both. And all `FACULTY`s and `Grad_STUDENT`s are not `INSTRUCTOR_RESEARCHER`s.



**Figure 8.9**  
An EER conceptual schema for a UNIVERSITY database.

# Design Choices for Specialization/Generalization

- Many specializations and subclasses can be defined to make the conceptual model accurate: specializations and subclasses ทำให้การออกแบบในระดับแนวคิดมีความถูกต้องแม่นยำชัดเจนขึ้น
- If subclass has few specific attributes and no specific relationships
  - Can be merged into the superclass
  - ในกรณีที่มีเพียง 1 subclass และมี attribute เฉพาะน้อย และไม่มีความสัมพันธ์เฉพาะเจาะจงกับ entity อื่น สามารถรวม subclass ดังกล่าวเข้ากับ superclass ได้

# Design Choices for Specialization/Generalization (cont'd.)

- 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
  - ในกรณีที่ทุก **subclass** มี **attribute** เฉพาะน้อย และไม่มีความสัมพันธ์เฉพาะเจาะจงกับ **entity** อื่น สามารถรวม **subclass** ดังกล่าวเข้ากับ **superclass** ได้ และกำหนด **attribute** เพิ่มเติมเพื่อใช้เป็นตัวที่กำหนดความแตกต่างของ **subclass**



# Design Choices for Specialization/Generalization (cont'd.)

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

# Formal Definitions for the EER Model

## Concepts

- **Class**
  - Set or collection of entities
  - Includes any of the EER schema constructs of group entities
- **Subclass**
  - Class whose entities must always be a subset of the entities in another class
- **Specialization**
  - Set of subclasses that have same superclass

# Formal Definitions for the EER Model

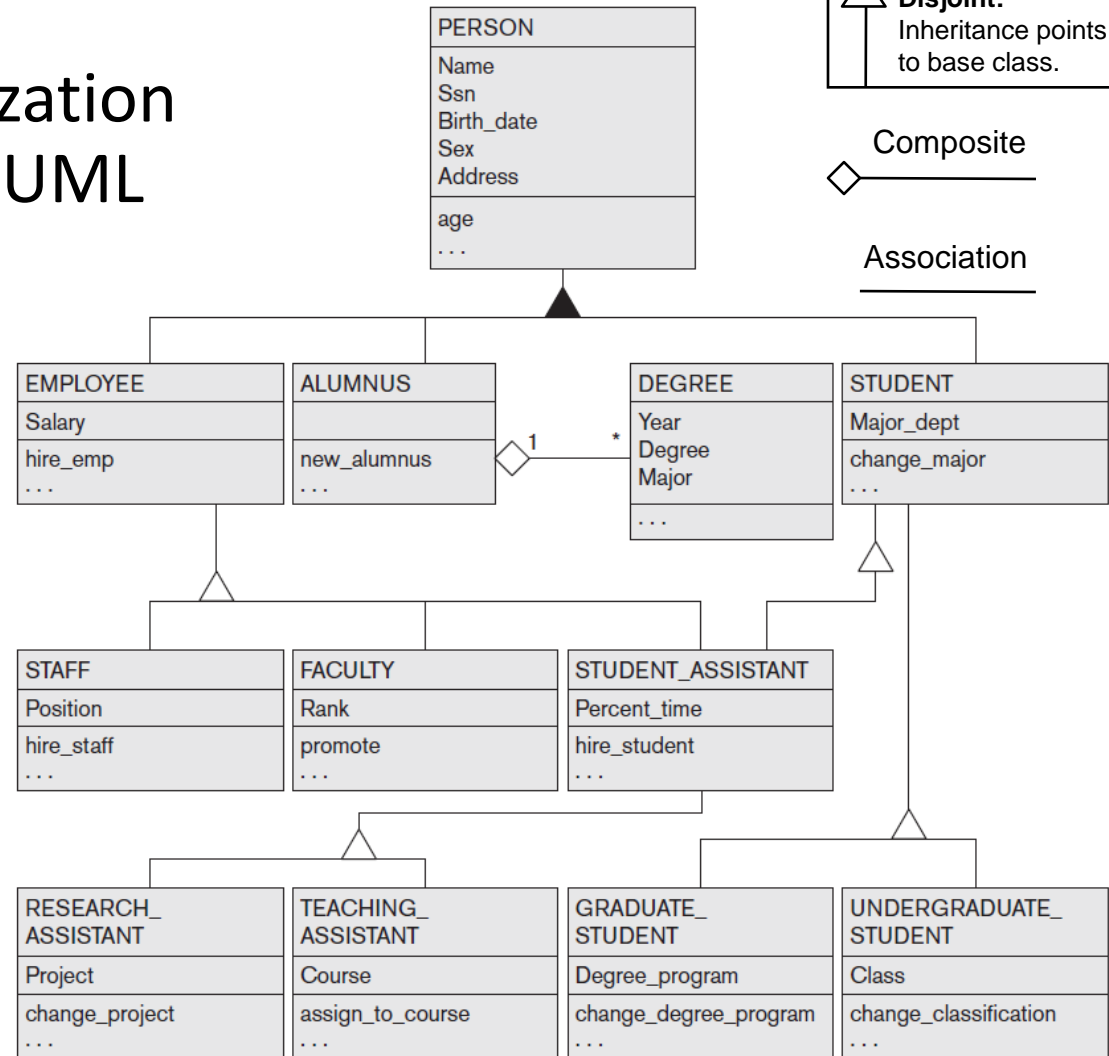
## Concepts (cont'd.)

- **Generalization**
  - Generalized entity type or superclass
- **Predicate-defined**
  - Predicate on the attributes of  $C$  is used to specify which entities in  $C$  are members of  $S$
- **User-defined**
  - Subclass that is not defined by a predicate
- **Category**
  - Class that is a subset of the union of  $n$  defining superclasses
- **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)



**Overlapping:**  
Inheritance points to base class.

**Disjoint:**  
Inheritance points to base class.

Composite

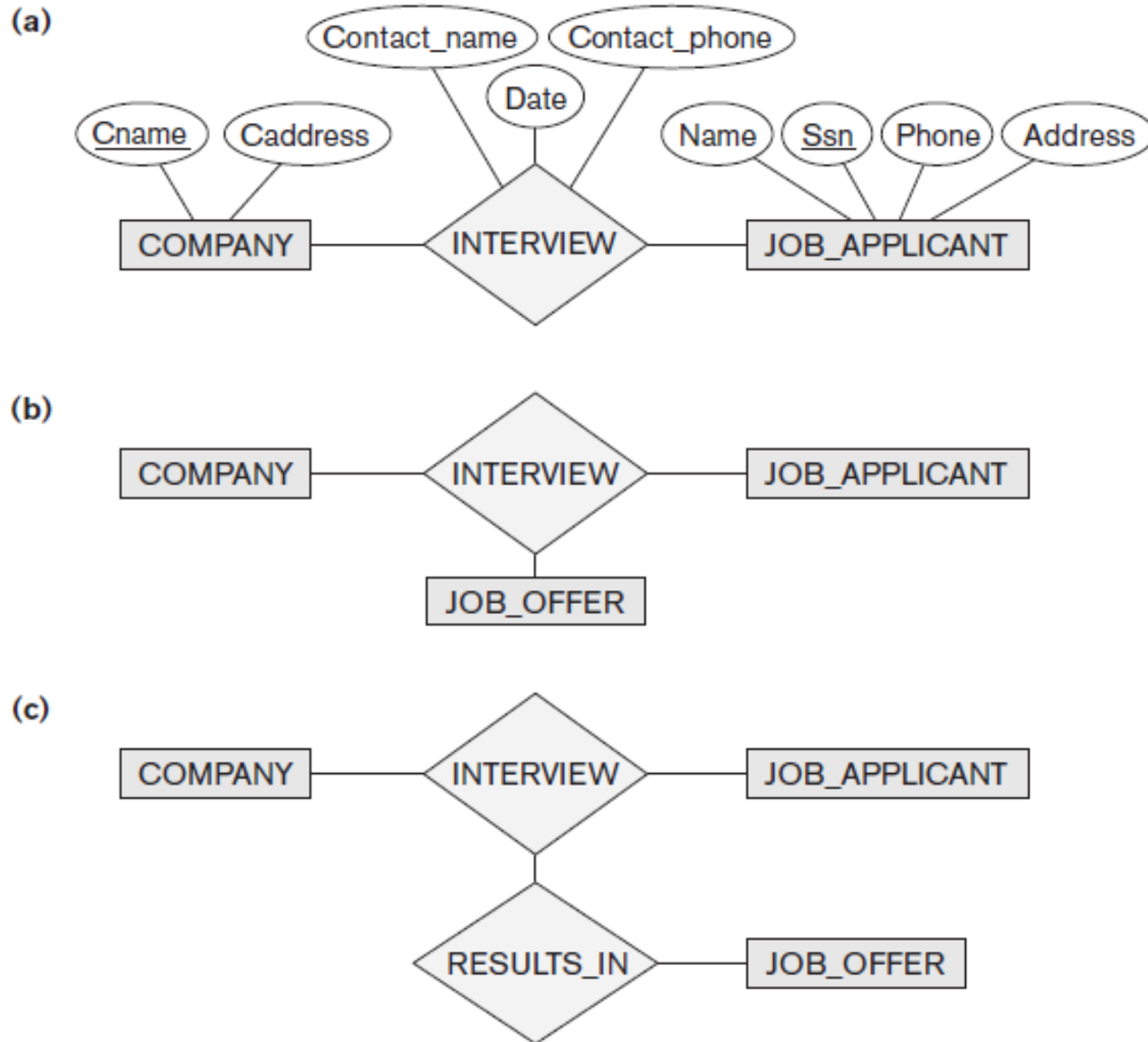
Association

**Figure 8.10**  
A UML class diagram corresponding to the EER diagram in Figure 8.7, illustrating UML notation for specialization/generalization.

# Specialization and Generalization

## Aggregation and Association

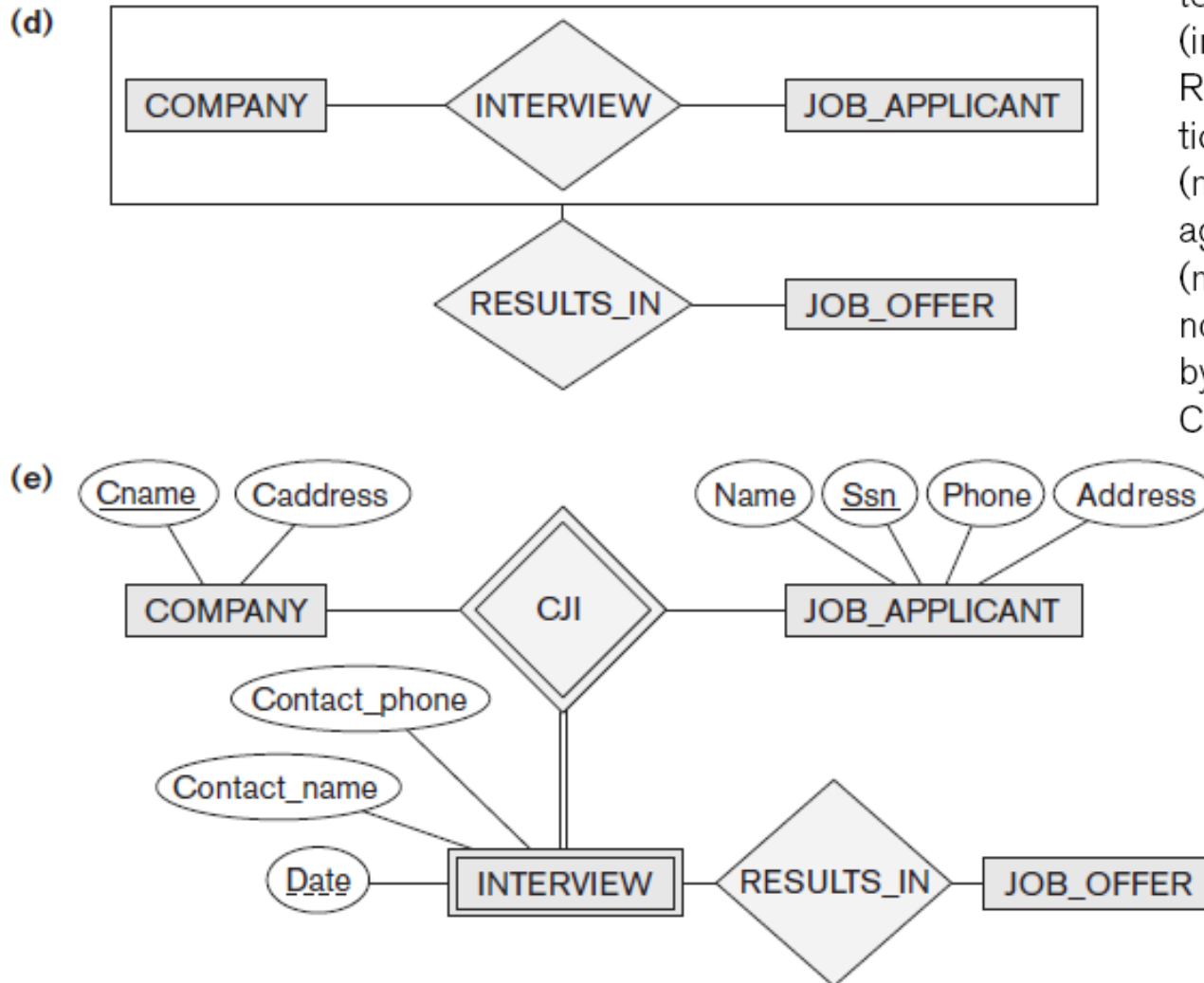
- **Specialization**
  - Classify a class of objects into more specialized subclasses
- **Generalization**
  - Generalize several classes into a higher-level abstract class
  - Includes the objects in all these classes
- **Aggregation**
  - Abstraction concept for building composite objects from their component objects
- **Association**
  - Associate objects from several independent classes
- **Main structural distinction**
  - When an association instance is deleted
    - Participating objects may continue to exist



**Figure 8.11**  
 Aggregation. (a) The relationship type INTERVIEW. (b) Including JOB\_OFFER in a ternary relationship type (incorrect). (c) Having the RESULTS\_IN relationship participate in other relationships (not allowed in ER). (d) Using aggregation and a composite (molecular) object (generally not allowed in ER but allowed by some modeling tools). (e) Correct representation in ER.

**Figure 8.11**

Aggregation. (a) The relationship type INTERVIEW. (b) Including JOB\_OFFER in a ternary relationship type (incorrect). (c) Having the RESULTS\_IN relationship participate in other relationships (not allowed in ER). (d) Using aggregation and a composite (molecular) object (generally not allowed in ER but allowed by some modeling tools). (e) Correct representation in ER.



# Summary

- Enhanced ER or EER model
  - Extensions to ER model that improve its representational capabilities
  - Subclass and its superclass
  - Category or union type
- Notation and terminology of UML for representing specialization and generalization