

## 204321 - Database Systems

### The Enhanced Entity-Relationship (EER) Model

สอนโดย

ผู้ช่วยศาสตราจารย์ ดร.อารีรัตน์ ตรงรัศมีทอง (Section 001)

รองศาสตราจารย์ ดร.สุริ เตชะวุฒิ (Section 002)

Addison-Wesley  
is an imprint of

PEARSON

Copyright © 2016 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

## 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

2

## The Enhanced Entity-Relationship (EER) Model

### • Enhanced ER (EER) model

- EER สร้างขึ้นมาเพื่อให้การออกแบบฐานข้อมูลมีความหมายที่ถูกต้องแม่นยำมากขึ้น ช่วยสะท้อนคุณสมบัติของข้อมูล และข้อจำกัดของข้อมูลได้ชัดเจนมากขึ้น
- EER ใช้สำหรับฐานข้อมูลของแอปพลิเคชันที่มีความซับซ้อนมากกว่าแอปพลิเคชันแบบดั้งเดิม รองรับข้อมูลที่มีลักษณะโครงสร้างไม่ตายตัว (Unstructured Data)

3

## Subclasses, Superclasses, and Inheritance

- EER Model รวมแบบจำลองทั้งหมดของ ER Model ไว้ และมีส่วนที่เพิ่มเติมขึ้นมาสำหรับข้อมูลที่มีความซับซ้อนมากยิ่งขึ้น
- ส่วนที่เพิ่มเติมจาก EER มีดังนี้
  - Subclasses and superclasses
  - Specialization and generalization
  - Category or union type
  - Attribute and relationship inheritance

4

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

- **Enhanced ER or EER diagrams**

- **EER Diagram** ใช้อธิบายแนวคิดที่อยู่ในรูปแบบโครงสร้าง ส่วนที่เพิ่มขยายจาก **ER Diagram**

- **Subtype or subclass of an entity type**

- ในบางครั้ง ภายใน **Entity Type** อาจประกอบด้วยข้อมูลกลุ่มย่อย ที่มีความหมายชัดเจน และมีความสำคัญในการนำเสนอ ข้อมูลดังกล่าวให้ชัดเจนในบางแอปพลิเคชันฐานข้อมูล (Database Application) เช่น Parent-Child

5

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

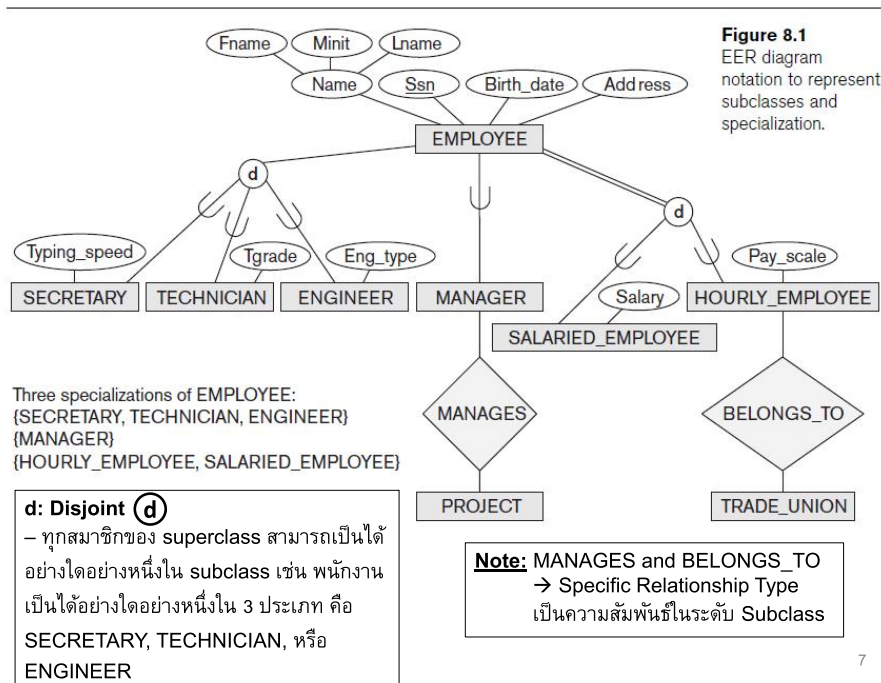
- ศัพท์ที่เกี่ยวข้องกับความสัมพันธ์ระหว่าง Superclass และ Subclass ที่ใช้ใน EER Diagram

- **Superclass/subclass** or
- **Supertype/subtype** or
- **Class/subclass** relationship

- **Type inheritance**

- Subclass สืบทอดทุกคุณสมบัติ (Inherit) และความสัมพันธ์ของ Superclass (สามารถสืบทอดได้หลายชั้น) สัญลักษณ์ที่ใช้แทน Inherit คือ U

6



7

## Specialization and Generalization

- **Specialization**

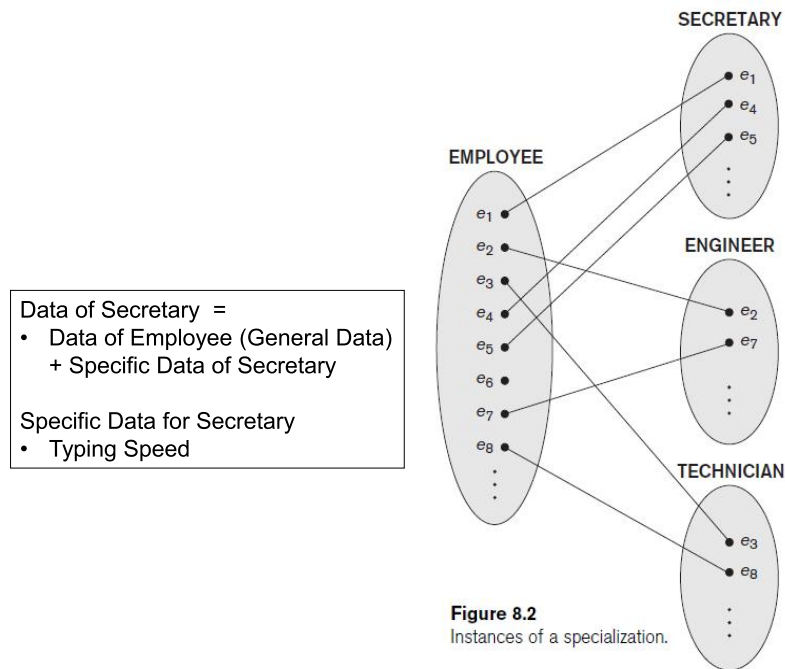
- คุณสมบัติที่แตกต่าง ลักษณะพิเศษ หรือลักษณะเฉพาะ ใน Superclass ให้กำหนดแยกไว้ใน Subclass แต่ละตัว
  - ลักษณะเฉพาะเกิดจาก บาง Attribute เป็นคุณสมบัติของบาง Record ใน Superclass ไม่ใช่คุณสมบัติของทุก Record

- **Subclass**

- Subclass สามารถกำหนดแอททริบิวต์และความสัมพันธ์ที่เฉพาะได้ ความสัมพันธ์อย่างนี้อาจจะเป็นความสัมพันธ์เฉพาะสำหรับสมาชิกใน Subclass

- **Specific attributes**
- **Specific relationship types**

8



9

## Generalization

- **Generalize into a single superclass:**
  - หาสิ่งที่มีร่วมกัน (Common) เก็บไว้ใน Superclass
  - ตั้งชื่อใหม่ที่มีลักษณะทั่วไป (General)
  - สำหรับส่วนที่แตกต่างกันแยกเก็บไว้ใน Subclass และใช้ชื่อเดิม
  - ตัวอย่างเช่น CAR กับ TRUCK นำส่วนที่มีร่วมกันไปไว้ใน Superclass และตั้งชื่อเป็น VEHICLE ส่วนสิ่งที่เฉพาะสำหรับ CAR กับ TRUCK ให้ไว้ที่เดิมซึ่งเป็น Subclass และใช้ชื่อเดิม
- **Generalization**
  - ขบวนการในการกำหนดสิ่งที่อยู่ในลักษณะทั่วไป (Generalized Entity) จากหลายๆ Entity Type ที่มีในระบบให้อยู่ในรูปแบบ Superclass

10

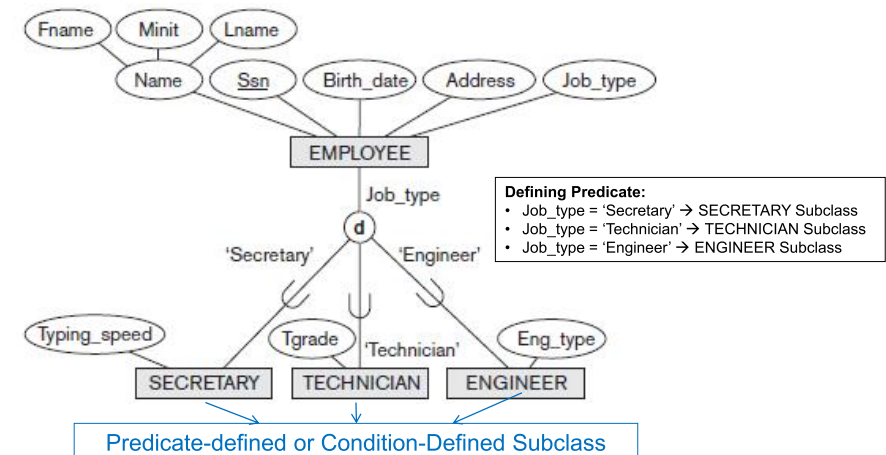
## Constraints on Specialization and Generalization

- สิ่งที่เป็นตัวกำหนด Subclass มีดังนี้
  - Predicate-defined or Condition-defined subclasses
  - Attribute-defined specialization
  - User-defined
- **Predicate-defined/Condition-defined Subclasses**
  - สิ่งที่เป็นตัวกำหนด Subclass คือ ค่าข้อมูลในบางแอตทริบิวต์ของ Superclass เช่น แอตทริบิวต์ job\_type ที่อยู่ใน Superclass ซึ่งเป็นตัวที่กำหนดสมาชิกของ Subclass
  - Subclass ประเภทนี้ เรียกว่า Predicate-defined หรือ Condition-defined Subclasses
  - เงื่อนไขที่เป็นตัวกำหนดแต่ละ Subclass เรียกว่า **Defining Predicate** เช่น เงื่อนไขที่เป็นตัวกำหนด Secretary Subclass คือ Job\_type = 'Secretary' และจะแสดงเงื่อนไขของค่าข้อมูลที่เป็นตัวกำหนดสมาชิกของแต่ละ Subclass ไว้ที่เส้นเชื่อมของแต่ละ Subclass ดังตัวอย่างแสดงในรูปในหน้าถัดไป

11

## Constraints on Specialization and Generalization

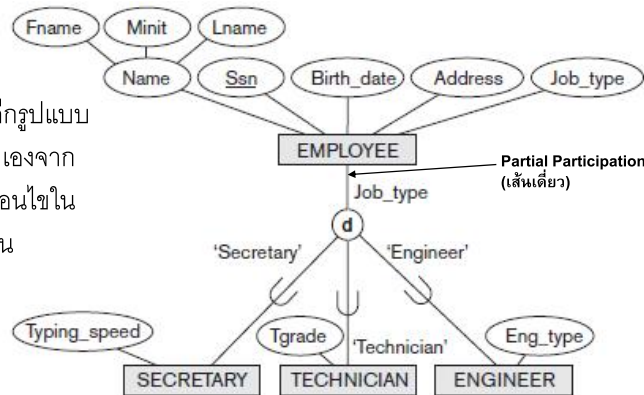
- **Predicate-defined/Condition-defined Subclasses**



12

## Constraints on Specialization and Generalization

- **Attribute-defined specialization:** Attribute เป็นตัวกำหนดที่ทำให้เกิดคุณลักษณะพิเศษ จากรูปด้านล่าง Job\_Type เป็นตัวทำให้เกิดคุณลักษณะพิเศษ ซึ่ง Job\_Type ถูกเรียกว่า Defining Attribute



13

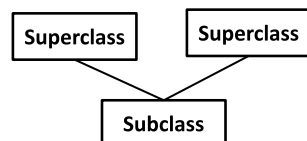
## Constraints on Specialization and Generalization (cont'd.)

- **Disjointness constraint:**
  - ข้อจำกัดที่เป็นได้อย่างใดอย่างหนึ่งของสมาชิกใน Subclass
- **Completeness (or totalness) constraint**
  - ข้อกำหนด ที่อยู่ในลักษณะ ทั้งหมด (Total) หรือ บางส่วน (Partial)
- ข้อจำกัด Disjointness และ Completeness เป็นอิสระจากกัน คือเป็นคนละส่วนกัน
- ตัวอย่างเช่น พนักงานเป็นสมาชิกได้อย่างใดอย่างหนึ่ง (Disjoint) จาก SECRETARY หรือ TECHNICIAN หรือ ENGINEER
  - หากข้อกำหนด Completeness เป็น Partial พนักงานไม่เป็นอย่างใดอย่างหนึ่งก็ได้ หรือเป็น 1 ใน 3 อย่างก็ได้ (min=0, max=1)
  - หากข้อกำหนด Completeness เป็น Total พนักงานต้องเป็นอย่างน้อย 1 ใน 3 อย่าง และอย่างมาก 1 ใน 3 เช่นกัน (min=1, max=1)

14

## Specialization and Generalization Hierarchies and Lattices

- **Specialization hierarchy: Single Inheritance**
  - ทุก ๆ Subclass สืบทอดคุณสมบัติจาก Superclass เพียงตัวเดียว ลักษณะโครงสร้างคล้ายกับโครงสร้างต้นไม้ (Tree Structure) หรือเรียกว่า (Strict Hierarchy)
- **Specialization lattice: Multiple Inheritance**
  - Subclass สามารถสืบทอดคุณสมบัติจาก Superclass ได้หลายตัว
  - บาง Software ไม่รองรับ Multiple Inheritance เช่น โปรแกรมภาษา Java



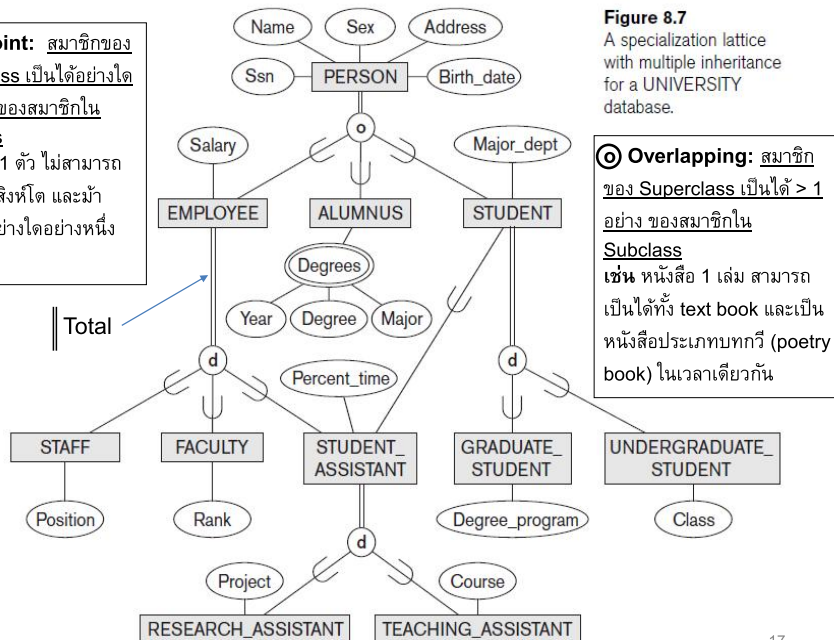
15

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

- **Overlapping constraint:**
  - ข้อกำหนดที่เป็นได้มากกว่าหนึ่งอย่างของสมาชิกใน Subclass
- **Completeness (or totalness) constraint**
  - ข้อกำหนด Overlapping และ Completeness เป็นอิสระจากกัน เช่นเดียวกับกับ Disjointness
- ตัวอย่างเช่น บุคคลเป็นสมาชิกได้หลายอย่าง (Overlapping) จาก EMPLOYEE, ALUMNUS, STUDENT
  - หากข้อกำหนด Completeness เป็น Partial บุคคลไม่เป็นอย่างใดอย่างหนึ่งก็ได้ หรือเป็นได้มากกว่า 1 อย่างก็ได้ (min=0, max=3)
  - หากข้อกำหนด Completeness เป็น Total บุคคลต้องเป็นอย่างน้อย 1 อย่าง หรือเป็นมากกว่า 1 อย่างก็ได้ (min=1, max=3)

16

Ⓢ **Disjoint:** สมาชิกของ Superclass เป็นได้อย่างใดอย่างหนึ่งของสมาชิกใน Subclass  
 เช่น สัตว์ 1 ตัว ไม่สามารถเป็นได้ทั้งสิงห์โต และม้า ต้องเป็นอย่างใดอย่างหนึ่งเท่านั้น



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

Ⓢ **Overlapping:** สมาชิกของ Superclass เป็นได้ > 1 อย่างของสมาชิกใน Subclass  
 เช่น หนังสือ 1 เล่ม สามารถเป็นได้ทั้ง text book และเป็นหนังสือประเภทบทกวี (poetry book) ในเวลาเดียวกัน

17

## Utilizing Specialization and Generalization in Refining Conceptual Schemas

### • Specialization process

- ขบวนการในการกำหนดลักษณะเฉพาะ หรือลักษณะพิเศษของ Subclass ดำเนินการได้ 2 รูปแบบ คือ

### • Top-down conceptual refinement process

- เริ่มจากภาพใหญ่ หรือลักษณะทั่วไปก่อน (Generalization) เช่น ถ้าระบบเกี่ยวกับสัตว์ ให้นึกถึงสัตว์ทั่วไปก่อน (Superclass) แล้วค่อยแยกกลุ่มย่อยของสัตว์แต่ละประเภท (Subclass) เช่น สัตว์บก สัตว์น้ำ สัตว์ปีก เป็นต้น

### • Bottom-up conceptual synthesis

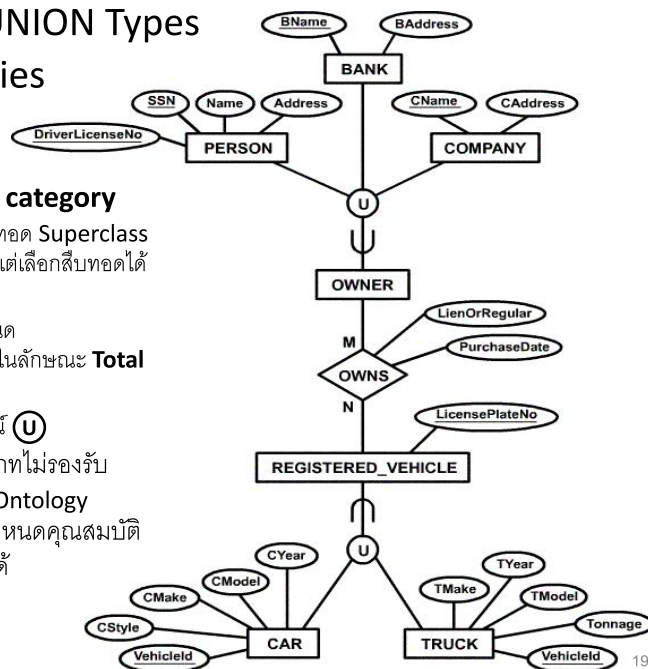
- ในทางกลับกัน ให้นึกถึงลักษณะเฉพาะ หรือลักษณะพิเศษก่อน (Specialization) เช่น ให้นึกถึงสัตว์แต่ละประเภทก่อน (Subclass) ได้แก่ สัตว์บก สัตว์น้ำ สัตว์ปีก แล้วค่อยหาสิ่งที่ร่วมกันของสัตว์เหล่านั้น และแยกส่วนที่ร่วมกันไปเป็น (Superclass)

18

## Modeling of UNION Types Using Categories

### – Union type or a category

- ใช้นำเสนอ การสืบทอด Superclass ที่มีมากกว่า 1 ตัว แต่เลือกสืบทอดได้เพียง 1 ตัว
- สามารถ มีข้อกำหนด Completeness ในลักษณะ Total หรือ Partial ได้
- แทนด้วยสัญลักษณ์ Ⓢ
- แบบจำลองบางประเภทไม่รองรับ Union Types แต่ใน Ontology สามารถรองรับการกำหนดคุณสมบัติแบบ Union Types ได้



19

## 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

20



### Ⓢ Union – การสืบทอดจาก

Superclass ที่มีมากกว่า 1 ตัว แต่เลือกสืบทอดได้เพียง 1 ตัว

เช่น subclass ชื่อ

INSTRUCTOR\_RESEARCHER สามารถเลือก inherit จาก

Superclass 2 ตัวคือ FACULTY หรือ Grad\_STUDENT แต่เลือก inherit ได้เป็น 1 ตัว

- แต่ละ super class อาจจะมี PK ที่เหมือนกัน หรือแตกต่างกันก็ได้

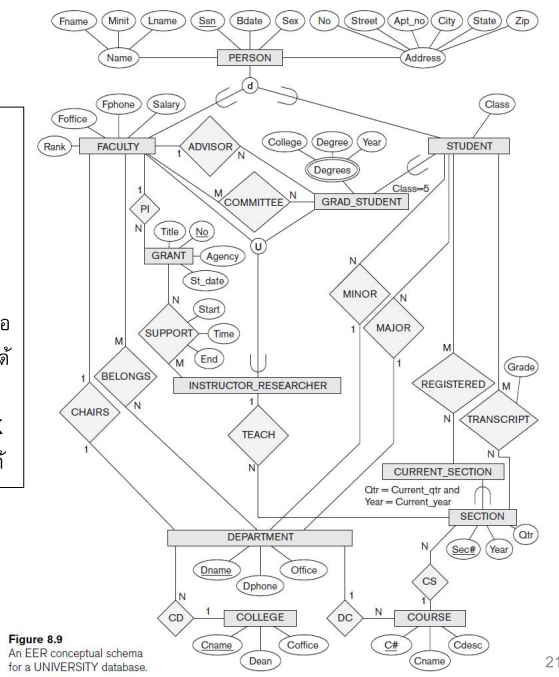


Figure 8.9  
An EER conceptual schema  
for a UNIVERSITY database.

21

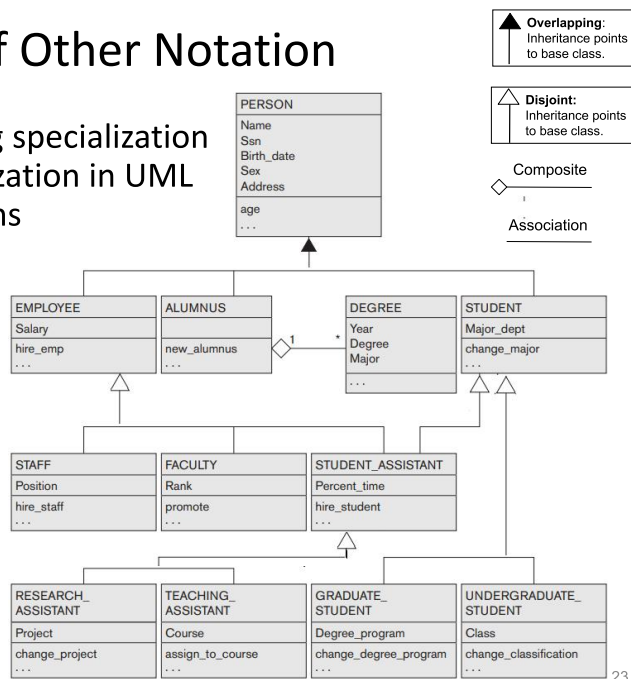
## Design Choices for Specialization/Generalization

- Specialization และ Subclass ทำให้การออกแบบในระดับแนวคิดมีความถูกต้องแม่นยำชัดเจนขึ้น
- ในกรณีที่มีเพียง 1 Subclass และมี Attribute เฉพาะน้อย และ ไม่มีความสัมพันธ์เฉพาะเจาะจงกับ Entity อื่น สามารถรวม Subclass ดังกล่าวเข้ากับ Superclass ได้
- ในกรณีที่ทุก Subclass มี Attribute เฉพาะน้อย และ ไม่มีความสัมพันธ์เฉพาะเจาะจงกับ Entity อื่น สามารถรวม Subclass ดังกล่าวเข้ากับ Superclass ได้ และกำหนด Attribute เพิ่มเติมจาก Attribute ของแต่ละ Subclass
- ถ้าเป็นไปได้ ให้หลีกเลี่ยง Union Types

22

## Example of Other Notation

- Representing specialization and generalization in UML class diagrams



23

## Aggregation and Association

### • Aggregation

— แนวคิดในการสร้าง Object ที่อยู่ในรูปองค์ประกอบ (Composite Object) เป็นการรวมหลาย Object ไว้ด้วยกัน

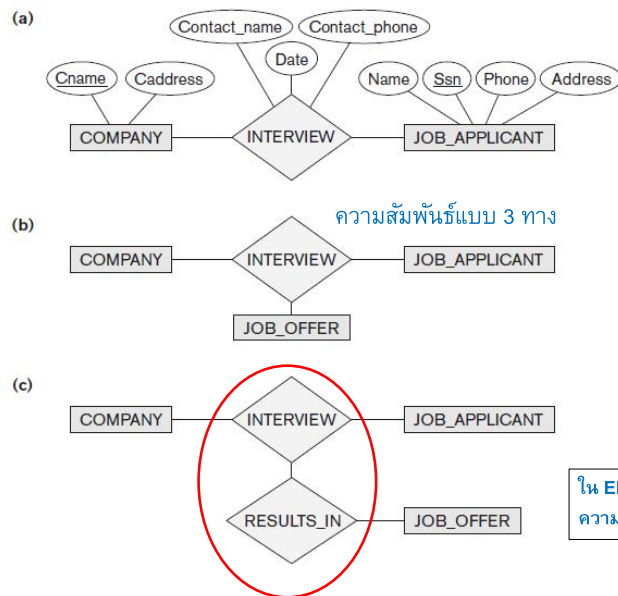
### • Association

— การเชื่อมโยง Object จากคลาสอิสระหลายคลาส

### • สิ่งที่ Association แตกต่างจาก Aggregation

— **Association:** หาก Instance ของ Class ที่เชื่อมโยงกัน ด้านใดด้านหนึ่ง ถูกลบ Instance ของ Class อีกด้านหนึ่งจะยังคงอยู่ ไม่ถูกลบไปด้วย

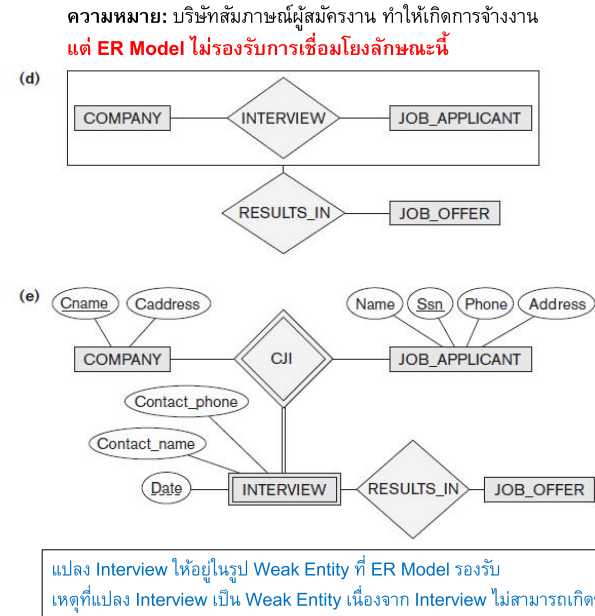
24



**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.

25



**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.

26

## 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

27