

## 204321 - Database Systems

### Relational Database Design by ER and EER-to-Relational Mapping

สอนโดย

ผู้ช่วยศาสตราจารย์ ดร.อารีรัตน์ ตรงรัศมีทอง (Section 001)

รองศาสตราจารย์ ดร.สุรี เตชะวุฒิ (Section 002)

Addison-Wesley  
Is an imprint of

PEARSON

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

2

### Relational Database Design by ER- and EER-to-Relational Mapping

- **Design a relational database schema**

- การออกแบบโครงสร้างฐานข้อมูลเชิงสัมพันธ์ ออกแบบบนพื้นฐานของการออกแบบโครงสร้างเชิงแนวคิด

- หลังจากที่เราได้แบบจำลองเชิงแนวคิดระดับสูง ซึ่งอยู่ในรูปแบบที่คอมพิวเตอร์ไม่สามารถประมวลผลได้ ขั้นตอนต่อไปเป็นการออกแบบในระดับตรรกะ ซึ่งผลลัพธ์ที่ได้คือ **Relation** (ศาสตร์เดียวกับเรื่องความสัมพันธ์ในเชิงคณิตศาสตร์) ซึ่งหากข้อมูลอยู่ในรูปแบบ เซ็ตของความสัมพันธ์เชิงคณิตศาสตร์ คอมพิวเตอร์จะสามารถประมวลผลข้อมูลได้

- **Algorithm** ในการแปลง แบบจำลองเชิงสัมพันธ์ให้อยู่ในรูปแบบ **Relations** ประกอบด้วย 7 ขั้นตอน

3

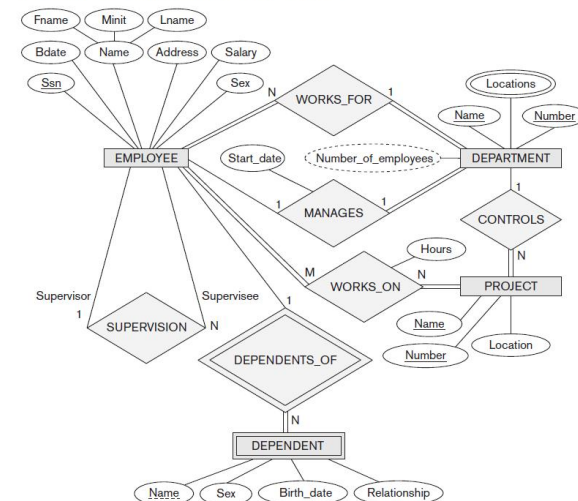
## Outline

- Relational Database Design Using ER-to-Relational Mapping
- Relational Database Design Using EER-to-Relational Mapping

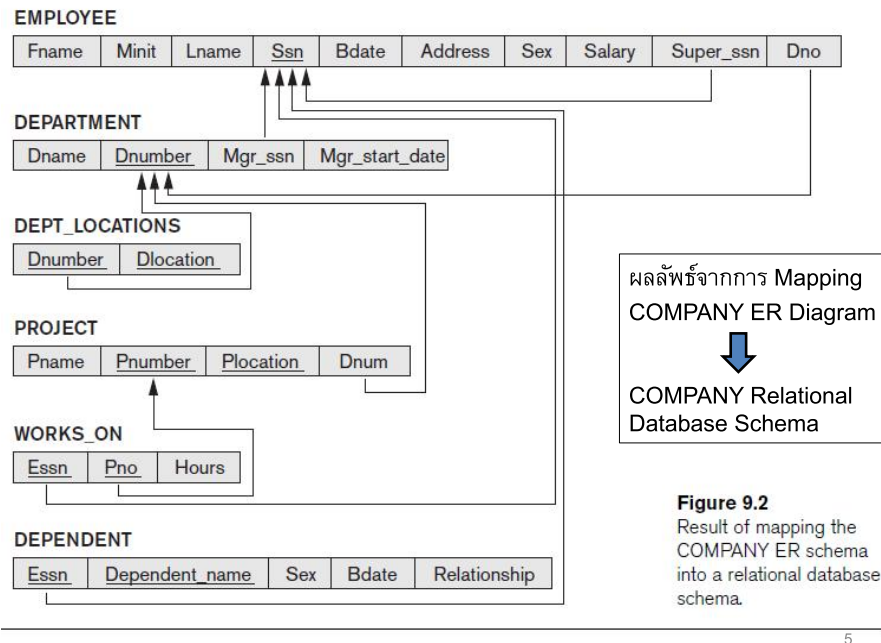
### Relational Database Design Using ER-to-Relational Mapping

Figure 9.1

The ER conceptual schema diagram for the COMPANY database.

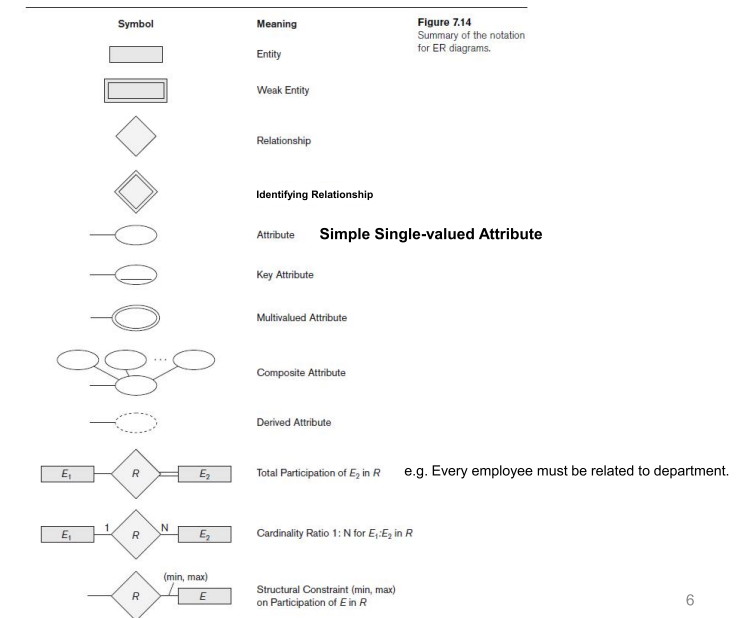


4



5

## ER Diagrams and Naming Conventions



6

## ER-to-Relational Mapping Algorithm

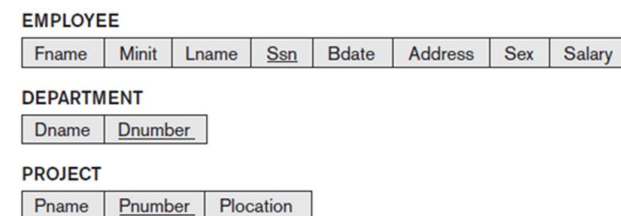
- Step 1: Mapping of Regular Entity Types
  - แปลงเอนทิตีแบบปกติ (Entity Type ที่ไม่ใช่ Weak Entity) ไปเป็น 1 Relation  $R$
  - นำเฉพาะ Simple Attribute(s) ของ Entity ไปไว้ใน  $R$
  - แต่ละ รีเลชัน เรียกว่า **Entity Relation**
    - ข้อมูลแต่ละ Tuple เรียกว่า Entity Instance

7

## ER-to-Relational Mapping Algorithm

- Step 1: Mapping of Regular Entity Types

— รีเลชันที่ได้หลังจากผ่านขั้นตอนที่ 1 มีดังนี้คือ



- **EMPLOYEE:** นำมาเฉพาะ แอททริบิวต์เชิงเดี่ยว (Fname, Minit, และ Lname) ของ แอททริบิวต์เชิงประกอบ Name (Name ไม่ต้องนำมา)
- **DEPARTMENT:** Location เป็นแอททริบิวต์ที่มีค่าหลายค่า (ไม่ใช่แอททริบิวต์เชิงเดี่ยว (ไม่ใช่ Simple Attribute) จึงไม่ต้องนำมา)

8

## ER-to-Relational Mapping Algorithm (cont'd.)

- Step 2: Mapping of Weak Entity Types
  - แปลง Weak Entity Type ไปเป็น 1 Relation  $R$
  - นำเฉพาะ Simple Attributes ของ Weak Entity ไปไว้ใน  $R$
  - ดำเนินการคล้ายกับขั้นตอนที่ 1 แต่ให้นำ Primary Key Attribute(s) ของ Entity ที่เป็น Owner (Entity ที่เชื่อมกับ Weak Entity ด้วย Identifying Relationship) ไปเป็น Foreign Key Attribute(s) ใน Relation  $R$  ด้วย

9

## ER-to-Relational Mapping Algorithm

### Step 2: Mapping of Weak Entity Types

— รีเลชันที่ได้หลังจากผ่านขั้นตอนที่ 2 มีดังนี้คือ

EMPLOYEE							
Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary

DEPENDENT				
Essn	Dependent_name	Sex	Bdate	Relationship

- **Essn:** นำมาจาก EMPLOYEE ซึ่งเป็นคีย์หลัก (Primary Key) ของรีเลชันเจ้าของ และนำมาเป็นคีย์ร่วมใน DEPENDENT รีเลชัน และแอททริบิวต์ทุกตัวใน DEPENDENT เป็นแอททริบิวต์เชิงเดี่ยว
- คีย์หลักของ DEPENDENT รีเลชัน อยู่ในลักษณะ คีย์ผสม ประกอบด้วย 2 แอททริบิวต์ คือ Essn และ Dependent\_name
- สำหรับ **Essn** ใน DEPENDENT มี 2 สถานะ คือ เป็นคีย์ร่วมของคีย์หลัก และเป็นคีย์นอก
- คีย์นอก (Foreign Key) คือคีย์ที่มาจากรีเลชันอื่น (คีย์ที่มาจากต่างแดน) ใช้สำหรับเชื่อมโยงกลับไปยังรีเลชันของคีย์หลัก เพื่อดึงข้อมูลจากฝั่งรีเลชันของคีย์หลัก (Essn ใน DEPENDENT ใช้สำหรับเชื่อมโยงไปยัง EMPLOYEE เพื่อดึงข้อมูลของพนักงาน)

10

## ER-to-Relational Mapping Algorithm (cont'd.)

Figure 9.3

Illustration of some mapping steps.

- Entity relations after step 1.
- Additional weak entity relation after step 2.
- Relationship relation after step 5.
- Relation representing multivalued attribute after step 6.

(a)	EMPLOYEE
	Fname Minit Lname <u>Ssn</u> Bdate Address Sex Salary
	DEPARTMENT
	Dname <u>Dnumber</u>
	PROJECT
	Pname <u>Pnumber</u> Plocation
(b)	DEPENDENT
	Essn <u>Dependent_name</u> Sex Bdate Relationship
(c)	WORKS_ON
	Essn <u>Pno</u> Hours
(d)	DEPT_LOCATIONS
	<u>Dnumber</u> <u>Dlocation</u>

11

## ER-to-Relational Mapping Algorithm (cont'd.)

### Step 3: Mapping of Binary 1:1 Relationship Types

— For each binary **1:1 relationship type**

- การแปลงความสัมพันธ์แบบ 1:1 ให้เป็น Relation  $R$  ประกอบด้วย 3 แนวทาง ดังนี้
- **Foreign key approach**
  - ใช้แนวทางของคีย์นอก สำหรับความสัมพันธ์แบบ 1:1 ที่การมีส่วนร่วมฝั่งหนึ่งเป็นแบบ Total Participation และอีกฝั่งเป็นแบบ Partial Participation
- **Merged relationship approach**
  - ใช้แนวทางการผสมผสาน สำหรับความสัมพันธ์แบบ 1:1 ที่การมีส่วนร่วมเป็นแบบ Total Participation ทั้งสองฝั่ง
- **Cross reference or relationship relation approach**
  - ใช้แนวทางของความสัมพันธ์ สำหรับความสัมพันธ์แบบ 1:1 ที่การมีส่วนร่วมเป็น Partial Participation ทั้งสองฝั่ง

12

## ER-to-Relational Mapping Algorithm (cont'd.)

### • Step 3: Mapping of Binary 1:1 Relationship Types (cont'd.)

#### • Foreign key approach

- การแปลงโดยการนำ PK ของฝั่ง Partial ไปเป็น FK ของฝั่ง Total
- นำ Simple Attribute(s) ที่อยู่ที่ 1:1 Relationship (ถ้ามี) ไปด้วย

#### • Merged relationship approach

- การแปลงโดยการรวม Relation ทั้งสองฝั่งเป็น 1 Relation
- นำ Simple Attribute(s) ที่อยู่ที่ 1:1 Relationship (ถ้ามี) ไปด้วย และตั้งชื่อ Relation ให้เหมาะสม

#### • Cross reference or relationship relation approach

- การแปลงโดยการสร้าง Relation ใหม่
- นำ PK ทั้งสองฝั่งไปเป็น PK (Composite Key) ในรีเลชันใหม่ ในขณะเดียวกัน PK ของทั้งสองฝั่ง จะเป็น FK ใน Relation ใหม่ด้วย
- นำ Simple Attribute(s) ที่อยู่ที่ 1:1 Relationship (ถ้ามี) ไปด้วย
- ข้อเสียของแนวทางนี้คือ มี Relation เพิ่มขึ้น 1 ตัว ซึ่งส่งผลให้หากต้องการข้อมูลที่เกี่ยวข้องสัมพันธ์กัน ต้องมีการ Join ข้อมูลมากขึ้น 1 ตารางข้อมูล

13

## ER-to-Relational Mapping Algorithm

### • Step 3: Mapping of Binary 1:1 Relationship Types

— รีเลชันที่ได้หลังจากผ่านขั้นตอนที่ 3 โดยใช้ **แนวทางของคีย์นอก**



- ให้นำคีย์หลักฝั่งที่มีส่วนร่วมบางส่วน ไปเป็นคีย์นอกในฝั่งที่มีส่วนร่วมทั้งหมด
- หากมีแอตทริบิวต์ที่มีค่าข้อมูลเชิงเดียวอยู่ที่ความสัมพันธ์ ให้รวมแอตทริบิวต์ดังกล่าวมาไว้ที่ฝั่งที่มีส่วนร่วมทั้งหมด ด้วย

#### EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary
-------	-------	-------	-----	-------	---------	-----	--------

#### DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
-------	---------	---------	----------------

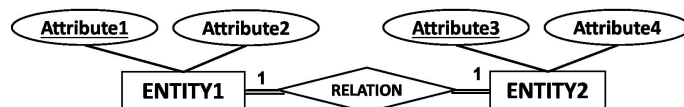
Ssn ถูกเปลี่ยนชื่อเป็น Mgr\_ssn  
Start\_date ถูกเปลี่ยนชื่อเป็น Mgr\_start\_date

14

## ER-to-Relational Mapping Algorithm

### • Step 3: Mapping of Binary 1:1 Relationship Types

— ตัวอย่างการแปลงในขั้นตอนที่ 3 โดยใช้ **แนวทางของการผสาน**



- ให้รวมสองเอนทิตีและความสัมพันธ์ไว้ด้วยกันใน 1 รีเลชัน โดยที่รวมเฉพาะแอตทริบิวต์ที่อยู่ในลักษณะมีค่าข้อมูลเชิงเดียวเท่านั้น (หากในความสัมพันธ์มีแอตทริบิวต์เชิงเดียว อย่าลืมนำมารวมด้วย)

- ให้เลือก Primary Key ที่เหมาะสม (เลือกระหว่าง Attribute1 และ Attribute3 ในตัวอย่างเลือก Attribute1)

#### ENTITY1

Attribute1	Attribute2	Attribute3	Attribute4
------------	------------	------------	------------

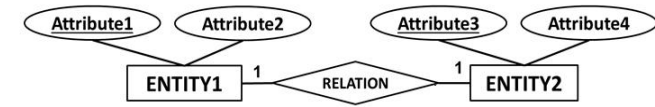
\*\*\*ฐานข้อมูล COMPANY ไม่มีความสัมพันธ์ในลักษณะนี้ จึงไม่เกิดการเปลี่ยนแปลงในขั้นตอนนี้\*\*\*

15

## ER-to-Relational Mapping Algorithm

### • Step 3: Mapping of Binary 1:1 Relationship Types

— ตัวอย่างการแปลงในขั้นตอนที่ 3 โดยใช้ **แนวทางของความสัมพัธ์**



- ให้สร้างรีเลชันจากความสัมพันธ์โดยรวมเฉพาะแอตทริบิวต์เชิงเดียวที่อยู่ในความสัมพันธ์ และรวมคีย์หลักของทั้งสองเอนทิตีไว้ที่รีเลชัน
- ข้อดีของการดำเนินการแนวทางนี้คือ จะจัดเก็บเฉพาะข้อมูลที่มีการเชื่อมโยงข้อมูลทั้งสองฝั่งเท่านั้น
- ข้อเสียคือ มีการสร้างรีเลชันเพิ่ม ทำให้เพิ่มจำนวนรีเลชันในการเชื่อมโยงข้อมูล (Join) มากขึ้น เมื่อต้องการรวมข้อมูลของสองฝั่ง

#### RELATION

Attribute1	Attribute3
------------	------------

Attribute1	Attribute2
------------	------------

Attribute3	Attribute4
------------	------------

\*\*\*ฐานข้อมูล COMPANY ไม่มีความสัมพันธ์ในลักษณะนี้ จึงไม่เกิดการเปลี่ยนแปลงในขั้นตอนนี้\*\*\*

16

## ER-to-Relational Mapping Algorithm (cont'd.)

### • Step 4: Mapping of Binary 1:N Relationship Types

— การแปลงความสัมพันธ์แบบ 1:N ให้เป็น Relation  $R$

- นำ PK ของฝั่ง 1 ไปเป็น FK ในฝั่ง N
- นำ Simple Attribute(s) ที่อยู่ 1:N Relationship (ถ้ามี) ไปด้วย

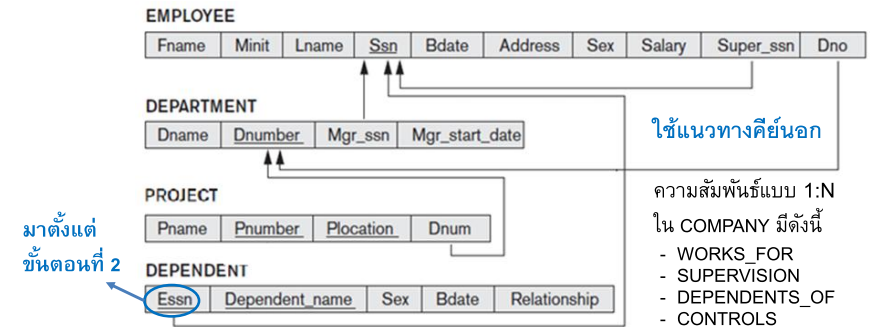
— หรือใช้แนวทางเดียวกับแนวทางที่ 3 ของความสัมพันธ์แบบ 1:1 คือ ใช้แนวทางการสร้าง Relation ใหม่ (แต่ส่วนใหญ่จะใช้วิธีการข้างต้นมากกว่า)

17

## ER-to-Relational Mapping Algorithm (cont'd.)

### • Step 4: Mapping of Binary 1:N Relationship Types

— รีเลชันที่ได้หลังจากผ่านขั้นตอนที่ 4 โดยใช้ **แนวทางคีย์นอก** (นำคีย์หลักในฝั่ง 1 ไปเป็นคีย์นอกในฝั่ง N)



18

## ER-to-Relational Mapping Algorithm (cont'd.)

### • Step 5: Mapping of Binary $M:N$ Relationship Types

— การแปลงความสัมพันธ์แบบ  $M:N$  ให้เป็น Relation  $R$  การดำเนินการเหมือนแนวทางที่ 3 ของการแปลงความสัมพันธ์แบบ 1:1

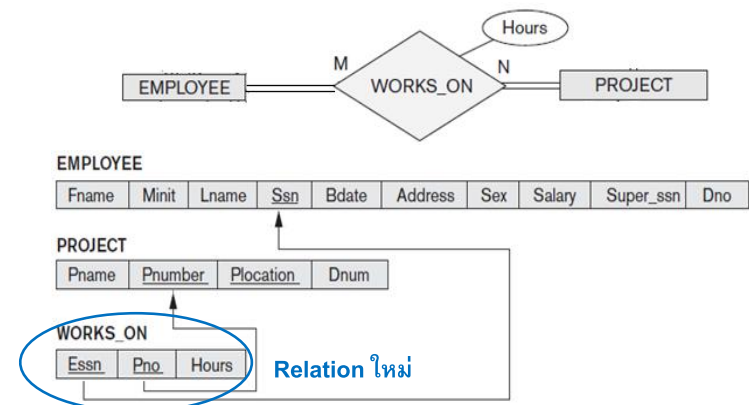
- สร้าง Relation ใหม่
- นำ PK ทั้งสองฝั่งไปเป็น PK (Composite Key) ในรีเลชันใหม่ ในขณะเดียวกัน PK ของทั้งสองฝั่ง จะเป็น FK ใน Relation ใหม่ด้วย
- นำ Simple Attribute(s) ที่อยู่  $M:N$  Relationship (ถ้ามี) ไปด้วย

19

## ER-to-Relational Mapping Algorithm (cont'd.)

### • Step 5: Mapping of Binary $M:N$ Relationship Types

— รีเลชันที่ได้หลังจากผ่านขั้นตอนที่ 5



20

## ER-to-Relational Mapping Algorithm (cont'd.)

### • Step 6: Mapping of **Multivalued Attributes**

— การแปลงแอททริบิวต์ที่มีค่าข้อมูลหลายค่า

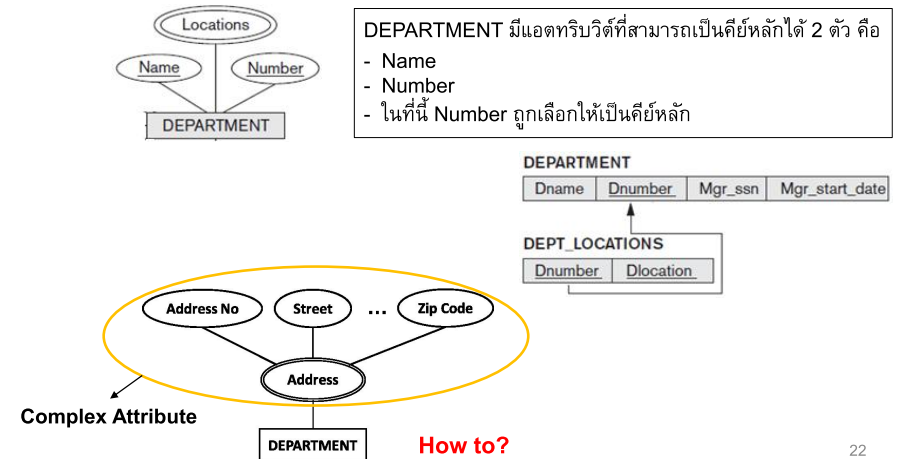
- สร้าง Relation ใหม่ และนำ Multivalued Attribute ไปอยู่ใน Relation ใหม่ด้วย
- นำ PK ของ Relation เดิมที่ Multivalued Attribute เคยอยู่ ไปเป็นคีย์ประกอบร่วมกับคีย์ของ Multivalued Attribute ในรีเลชันใหม่
- หาก Multivalued Attribute อยู่ในรูป Composite Attribute ให้นำ Simple Attributes ของ Multivalued Attribute ไปอยู่ในรีเลชันใหม่

21

## ER-to-Relational Mapping Algorithm (cont'd.)

### • Step 6: Mapping of **Multivalued Attributes**

— รีเลชันที่ได้หลังจากผ่านขั้นตอนที่ 6



22

## ER-to-Relational Mapping Algorithm (cont'd.)

### • Step 7: Mapping of **N-ary Relationship Types**

— การแปลงความสัมพันธ์ตั้งแต่ 3 ทางขึ้นไป ให้เป็น Relation *R* (ดำเนินการเหมือน Step 5 แต่ จำนวนคีย์ใน Composite Key จะมีเท่ากับจำนวน PK ในทุก Entities ที่สัมพันธ์กันรวมกัน)

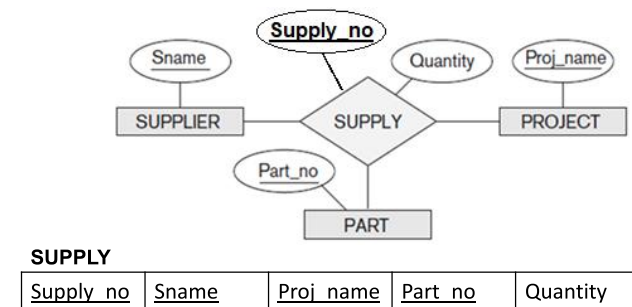
- สร้าง Relation ใหม่
- นำ PK ของทุกฝั่งไปเป็น PK (Composite Key) ในรีเลชันใหม่ ในขณะเดียวกัน PK ของทุกฝั่ง จะเป็น FK ใน Relation ใหม่ ด้วย
- นำ Simple Attribute(s) ที่อยู่ที่ความสัมพันธ์แบบหลายทาง Relationship (ถ้ามี) ไปด้วย

23

## ER-to-Relational Mapping Algorithm (cont'd.)

### • Step 7: Mapping of **N-ary Relationship Types**

— ตัวอย่างการแปลงในขั้นตอนที่ 7



\*\*\*ฐานข้อมูล COMPANY ไม่มีความสัมพันธ์ในลักษณะนี้ จึงไม่เกิดการเปลี่ยนแปลงในขั้นตอนนี้\*\*\*

24

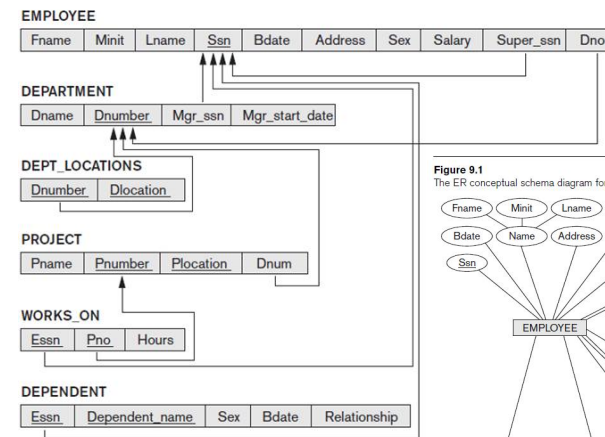


# Discussion and Summary of Mapping for ER Model Constructs

**Table 9.1** Correspondence between ER and Relational Models

ER MODEL	RELATIONAL MODEL
Entity type	Entity relation
1:1 or 1:N relationship type	Foreign key (or <i>relationship</i> relation)
M:N relationship type	<i>Relationship</i> relation and <i>two</i> foreign keys
<i>n</i> -ary relationship type	<i>Relationship</i> relation and <i>n</i> foreign keys
Simple attribute	Attribute
Composite attribute	Set of simple component attributes
Multivalued attribute	Relation and foreign key
Value set	Domain
Key attribute	Primary (or secondary) key

25



**Figure 9.1**  
The ER conceptual schema diagram for the COMPANY database.

26

## Mapping EER Model Constructs to Relations

- Extending ER-to-relational mapping algorithm:  
ขั้นตอนที่ 8 เป็นส่วนที่ขยายจากการแปลงอีอาร์เพื่อรองรับการแปลงความสัมพันธ์ที่อยู่ในลักษณะการสืบทอด Inheritance ในลักษณะ Disjoint หรือ Overlapping
- Step 8:** การแปลงขั้นตอนที่ 8 ประกอบด้วย 4 ทางเลือกคือ 8A – 8D
  - Option 8A:** สร้าง Relations จากทั้ง Superclass และ Subclass
  - Option 8B:** สร้าง Relations จาก Subclass เท่านั้น
  - Option 8C:** สร้างเพียง 1 Relation กับ 1 Attribute ที่ใช้แยกแยะข้อมูล (Discriminating Attribute)
  - Option 8D:** สร้างเพียง 1 Relation กับ Attribute หลายตัวที่อยู่ในประเภทบูลีนแอดทริบิวต์ (Boolean Type Attribute) เพื่อใช้ระบุว่า Tuple เป็นของคลาสย่อยหรือไม่

27

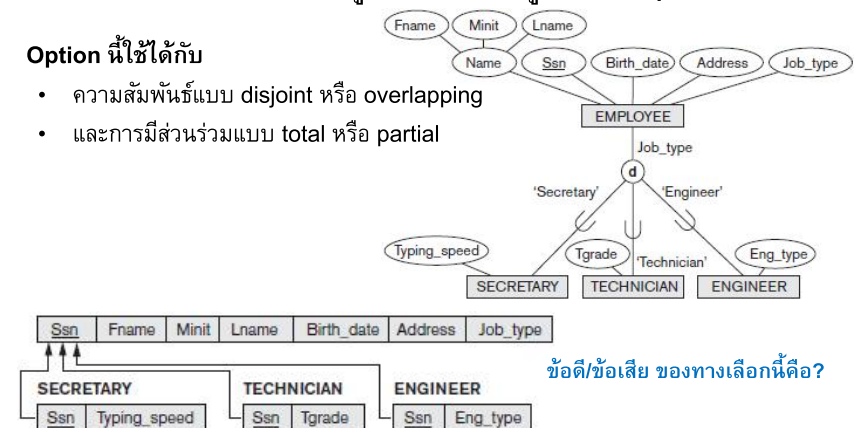
## Mapping of Specialization or Generalization

### Option 8A: Multiple relations superclass and subclasses.

- สร้าง Relations สำหรับทุก Superclass และทุก Subclass
  - นำ PK ของ Superclass ไปไว้ใน Relation ของทุก Subclass สำหรับเชื่อมโยงไปยัง Attribute ที่อยู่ใน Relation ที่อยู่ในระดับ Superclass

### Option นี้ใช้ได้กับ

- ความสัมพันธ์แบบ disjoint หรือ overlapping
- และการมีส่วนร่วมแบบ total หรือ partial



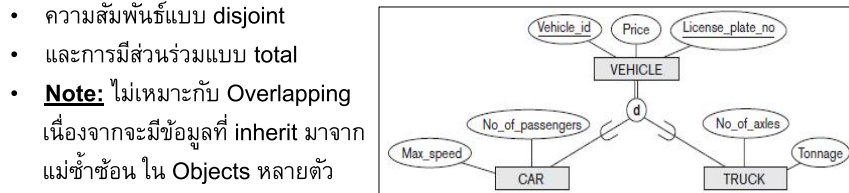
ข้อดี/ข้อเสีย ของทางเลือกนี้คือ?

28

## Mapping of Specialization or Generalization

### Option 8B: Multiple relations—subclass relations only.

- สร้าง Relations สำหรับทุก Subclass
- นำ Attribute ทุกตัวใน Superclass ไปไว้ในทุก Relation ของ Subclass
- Option นี้ใช้ได้กับ (เป็นอย่างใดอย่างหนึ่งแน่ ๆ)



CAR				
Vehicle_id	License_plate_no	Price	Max_speed	No_of_passengers

TRUCK				
Vehicle_id	License_plate_no	Price	No_of_axles	Tonnage

ข้อดี/ข้อเสีย ของทางเลือกนี้คือ?

29

## Mapping of Specialization or Generalization

### Option 8C: Single relation with one type attribute

- สร้าง Relation เดียวเท่านั้น
- นำ Attribute ทุกตัวใน Superclass และ Subclass มารวมกัน
- เพิ่ม Discriminating Attribute (ถ้ามี)
- Option นี้ใช้ได้กับ



ข้อดี/ข้อเสีย ของทางเลือกนี้คือ?

EMPLOYEE									
Ssn	Fname	Minit	Lname	Birth_date	Address	Job_type	Typing_speed	Tgrade	Eng_type

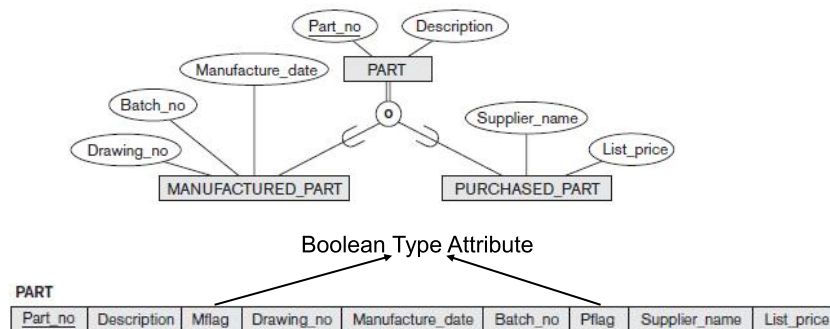
Discriminating Attribute

30

## Mapping of Specialization or Generalization

### Option 8D: Single relation with multiple type attributes

- สร้าง Relation เดียวเท่านั้น
- นำ Attribute ทุกตัวใน Superclass และ Subclass มารวมกัน
- เพิ่ม Boolean Type Attribute เพื่อบอกว่า เป็นอะไรได้บ้างใน Subclass
- Option นี้ใช้ได้กับ
  - ความสัมพันธ์แบบ overlapping หรือจะใช้กับ disjoint ก็ได้



Boolean Type Attribute

PART								
Part_no	Description	Mflag	Drawing_no	Manufacture_date	Batch_no	Pflag	Supplier_name	List_price

ข้อดี/ข้อเสีย ของทางเลือกนี้คือ?

31

## Mapping of Shared Subclasses (Multiple Inheritance)

- Apply any of the options discussed in step 8 to a shared subclass

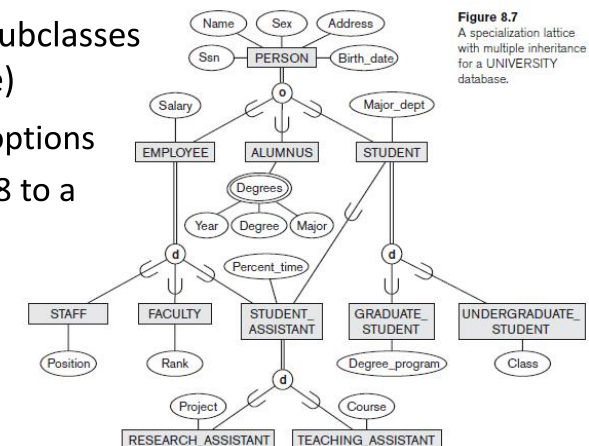


Figure 8.7  
A specialization lattice with multiple inheritance for a UNIVERSITY database.

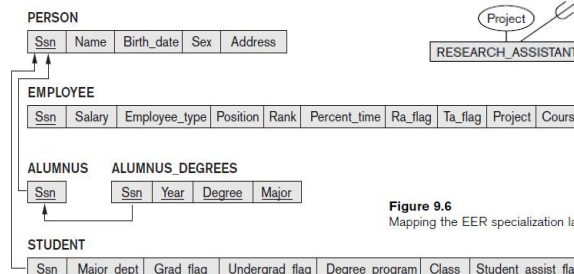


Figure 9.6  
Mapping the EER specialization lattice in Figure 8.8 using multiple options.

32



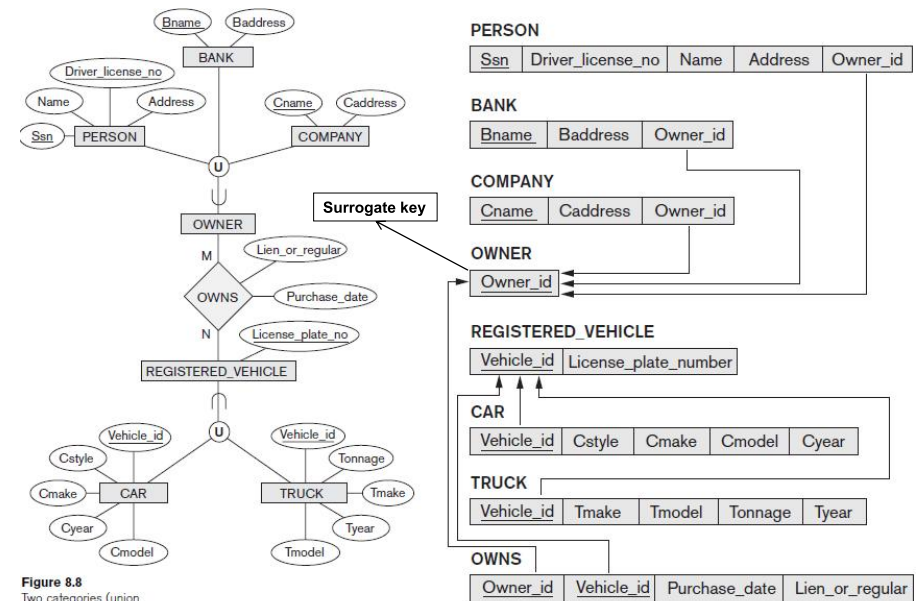
## Mapping of Categories (Union Types)

### • Step 9: Mapping of Union Types (Categories)

— ให้สร้าง Relations จากทั้ง Superclass และ Subclass

- ในกรณีที่ Superclasses มี PK ที่แตกต่างกัน ให้สร้าง Key Attribute ตัวใหม่เรียกว่า **Surrogate key** ที่ Relation ในฝั่ง Subclass และนำ **Surrogate key** ไปเป็น FK ที่ Relation ในฝั่ง Superclass
- ในกรณีที่ Superclasses มี PK ที่ไม่แตกต่างกัน ไม่ต้องสร้าง **Surrogate key** ให้นำ PK ของ Superclass ไปเป็น PK ในฝั่ง Subclass และ PK ฝั่ง Superclass เป็นทั้ง PK และ FK ในเวลาเดียวกัน

33



**Figure 8.8**  
Two categories (union types): **OWNER** and **REGISTERED\_VEHICLE**.

**Figure 9.7**  
Mapping the EER categories (union types) in Figure 8.8 to relations.

34

## EER Mapping Guideline

- กรณีที่ออกแบบแบบจำลองเชิงแนวคิดในลักษณะ EER ให้ดำเนินการ **Step 8** และ **Step 9** ก่อน แล้วจึงดำเนินการ **Step 1-7**
- เนื่องจากใน **Step 8** และ **9** อาจจะมีการแตก Entity เพิ่มเติม จะได้ไม่ต้องย้อนกลับไปตรวจสอบ **Step 1-7** สำหรับ Entity ที่เกิดขึ้นใหม่

35

## Summary

- Map conceptual schema design in the ER model to a relational database schema
  - Algorithm for ER-to-relational mapping
  - Illustrated by examples from the COMPANY database
- Include additional steps in the algorithm for mapping constructs from EER model into relational model

36