

## 204321 - Database Systems

### Data Modeling Using the Entity-Relationship (ER) Model

สอนโดย

ผู้ช่วยศาสตราจารย์ ดร.อารีรัตน์ ตรังศรีทอง (Section 001)

รองศาสตราจารย์ ดร.สุริ เตชะวุฒิ (Section 002)



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

## Outline

- Using High-Level Conceptual Data Models for Database Design
- A Sample Database Application
- Entity Types, Entity Sets, Attributes, and Keys
- Relationship Types, Relationship Sets, Roles, and Structural Constraints
- Weak Entity Types
- Refining the ER Design for the COMPANY Database
- ER Diagrams, Naming Conventions, and Design Issues
- Example of Other Notation: UML Class Diagrams
- Relationship Types of Degree Higher than Two

2

## แบบจำลองข้อมูลโดยใช้แบบจำลองอีอาร์

- **แบบจำลองอีอาร์ (ER Model)**
  - อีอาร์ ย่อมาจาก Entity-Relationship เป็นแบบจำลองข้อมูลที่นิยมสำหรับการออกแบบแบบจำลองเชิงแนวคิดระดับสูง
  - **แบบจำลองเชิงแนวคิด** ใช้สำหรับนำเสนอมุมมองของข้อมูลที่เราสนใจในระดับแนวคิด
    - **ระดับแนวคิด** เป็นระดับที่มนุษย์เข้าใจ แต่คอมพิวเตอร์ไม่สามารถประมวลผลได้ ในบทเรียนนี้ นำเสนอแบบจำลองเชิงแนวคิดในรูปแบบ แผนภาพอีอาร์ (ER Diagram)
    - แบบจำลองเชิงแนวคิด จะต้องถูกแปลงให้อยู่ในรูปที่คอมพิวเตอร์สามารถประมวลผลได้ จึงจะจัดเก็บไว้ในฐานข้อมูลได้
  - นอกจากนี้ยังมีแบบจำลองเชิงแนวคิดในลักษณะอื่น เช่น ยูเอ็มแอล (Unified Modeling Language: UML) ที่ใช้สำหรับนำเสนอแนวคิดในการออกแบบเชิงวัตถุ (Object-oriented Design: OOD)

3

## การใช้แบบจำลองเชิงแนวคิดระดับสูงสำหรับออกแบบฐานข้อมูลภาพรวมขั้นตอนการออกแบบฐานข้อมูล

- 1) การรวบรวมและวิเคราะห์ความต้องการของผู้ใช้ระบบ ผลลัพธ์ที่ได้คือ ฟังก์ชันต่าง ๆ ของแอปพลิเคชันตามความต้องการของผู้ใช้ที่เรียกว่า **Functional Requirements**
- 2) ออกแบบโครงสร้างข้อมูลระดับแนวคิด (**Conceptual Schema**) ที่นำเสนอในรูปแบบ แผนภาพอีอาร์ (ER Diagram)
- 3) การออกแบบเชิงตรรกะ (**Logical Design หรือ Data Model Mapping**) ผลลัพธ์ที่ได้คือ โครงสร้างฐานข้อมูล (**Database Schema**)
- 4) การออกแบบเชิงกายภาพ (**Physical Design Phase**) เพื่อให้การจัดการฐานข้อมูลมีประสิทธิภาพมากยิ่งขึ้น

4

## ขั้นตอนการออกแบบฐานข้อมูล

### 1) การรวบรวมและวิเคราะห์ความต้องการของผู้ใช้ระบบ

- นักออกแบบฐานข้อมูล (Database Designer) จะสัมภาษณ์ผู้ใช้ (User) ที่เป็นคนใช้ฐานข้อมูล เพื่อให้เข้าใจความต้องการของผู้ใช้ และจะได้ดำเนินการจัดทำเอกสารที่เกี่ยวข้องกับฐานข้อมูลได้ถูกต้อง
- การรวบรวมความต้องการของผู้ใช้ ส่วนใหญ่จะใช้วิธีสัมภาษณ์ผู้ใช้ ผลลัพธ์ที่ได้จากการสัมภาษณ์คือ ข้อมูลที่ผู้ใช้ต้องการ
- หลังจากนั้น นักออกแบบฐานข้อมูล จะดำเนินการจัดทำฟังก์ชันต่างๆของแอปพลิเคชันตามความต้องการของผู้ใช้  
**(Functional Requirements)**

5

## ขั้นตอนการออกแบบฐานข้อมูล (ต่อ)

### 2) ออกแบบโครงสร้างข้อมูลระดับแนวคิด

- โครงสร้างฐานข้อมูลในระดับแนวคิด ใช้สำหรับอธิบายข้อมูลตามความต้องการของผู้ใช้ฐานข้อมูล
- ออกแบบรายละเอียดของประเภทเอนทิตี (Entity) ความสัมพันธ์ระหว่าง เอนทิตี และกำหนดข้อบังคับของข้อมูล
- เปลี่ยนจากแบบจำลองข้อมูลระดับสูงไปเป็นแบบจำลองเชิงแนวคิด
  - วิเคราะห์ข้อมูลที่ใช้ในฟังก์ชันต่างๆ และนำเสนอข้อมูลและความสัมพันธ์ของข้อมูลเหล่านี้ในรูปแบบ แผนภาพอีอาร์ หรือแบบจำลองเชิงแนวคิด

6

## ขั้นตอนการออกแบบฐานข้อมูล (ต่อ)

### 3) การออกแบบเชิงตรรกะ

- แปลงแบบจำลองเชิงแนวคิด (แผนภาพอีอาร์) ที่คอมพิวเตอร์ไม่สามารถประมวลผลได้ ให้อยู่ในรูปแบบที่คอมพิวเตอร์สามารถประมวลผลได้
- คือ การแปลงแผนภาพอีอาร์ให้อยู่ในรูปโครงสร้างเชิงสัมพันธ์
- ผลลัพธ์ที่ได้จากขั้นตอนนี้คือ โครงสร้างฐานข้อมูลในวิชานี้อยู่ในรูป โครงสร้างฐานข้อมูลเชิงสัมพันธ์  
**(Relational Database Schema)**

7

## ขั้นตอนการออกแบบฐานข้อมูล (ต่อ)

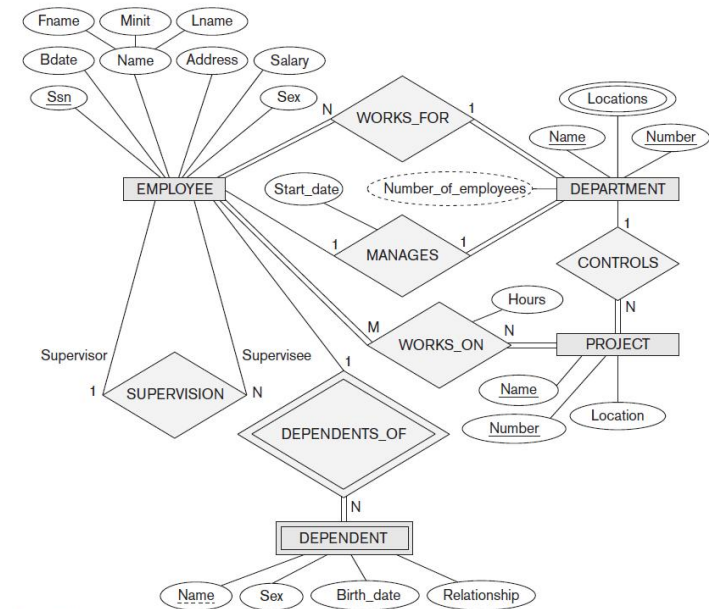
### 4) ขั้นตอนการออกแบบเชิงกายภาพ คือ ขั้นตอนสุดท้าย เป็นการกำหนด

- โครงสร้างของสถานที่ ที่ใช้จัดเก็บข้อมูล
- องค์ประกอบของไฟล์ข้อมูล
- ดัชนีข้อมูล (Index) และเส้นทางในการเข้าถึง ข้อมูล (Access Path)
- พารามิเตอร์ต่างๆที่เกี่ยวข้องกับการออกแบบเชิงกายภาพสำหรับไฟล์ฐานข้อมูล

8

## ตัวอย่างฐานข้อมูล

- ฐานข้อมูลชื่อ **COMPANY** ใช้สำหรับจัดเก็บข้อมูลบริษัท ประกอบด้วยข้อมูลดังต่อไปนี้
  - พนักงาน (Employees) แผนก (Departments) และโครงการ (Projects)
  - บริษัทประกอบด้วย แผนกต่างๆ
  - แผนกควบคุมงานในโครงการต่างๆ
  - ข้อมูลของพนักงานแต่ละคน ประกอบด้วย
    - ชื่อพนักงาน (Employee's name) หมายเลขประกันสังคม (Social Security Number) ที่อยู่ (Address) เงินเดือน (Salary) เพศ (Sex/Gender) และวันเกิด (Birth Date)
    - อีกทั้งมีการจัดเก็บข้อมูลผู้ที่อยู่ภายใต้การอุปการะของพนักงานแต่ละคน เช่น บุตร หรือภรรยา ที่พนักงานต้องอุปการะเลี้ยงดู (มีผลกับเรื่องภาษี)



**Figure 7.2**  
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.

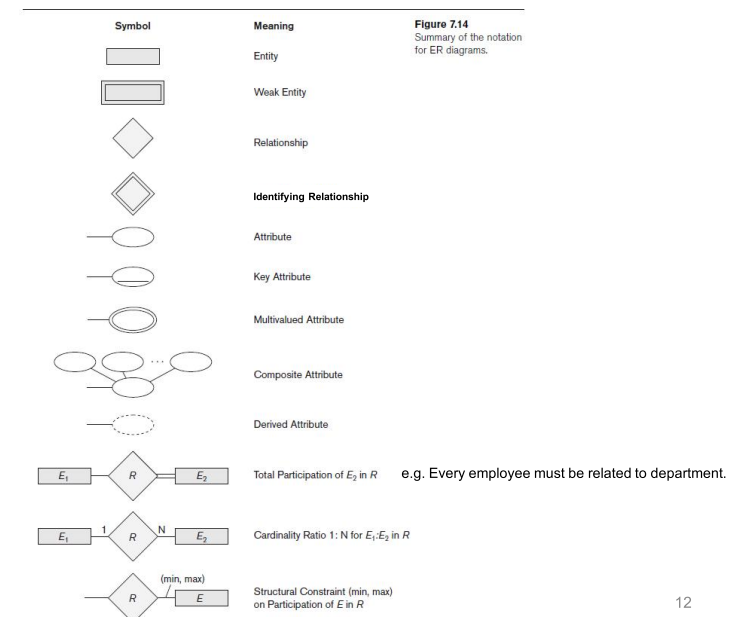
10

## แบบจำลองข้อมูลอีอาร์ (ER Model)

- แบบจำลองอีอาร์ (ER Model) ประกอบด้วย
  - เอนทิตี (Entity):** สิ่งต่างๆที่เกิดขึ้นจริงในโลกที่ดำรงอยู่ได้ด้วยตัวเอง หรือเป็นอิสระจากสิ่งอื่น
  - ความสัมพันธ์ (Relationship):** ความสัมพันธ์ระหว่างเอนทิตี
  - แอททริบิวต์:** คุณสมบัติเฉพาะที่สามารถอธิบายหรือบ่งบอกความเป็นตัวตนของเอนทิตีได้ แอททริบิวต์ มีหลายประเภท ดังนี้
    - เชิงประกอบ (Composite) / เชิงเดี่ยว (Simple หรือ Atomic)
    - มีค่าข้อมูลค่าเดียว (Single-valued) / มีค่าข้อมูลหลายค่า (Multivalued)
    - จัดเก็บข้อมูล (Stored) / คำนวณได้จากแอททริบิวต์ตัวอื่น (Derived)
    - ไม่มีค่าข้อมูล (NULL Values)
    - เชิงซ้อน (Complex)

11

## สัญลักษณ์ที่ใช้ในแผนภาพอีอาร์



12

## แนวทางในการออกแบบเอนทิตีและแอททริบิวต์

### a. การออกแบบจากบนลงล่าง (Top Down Design)

### b. การออกแบบจากล่างขึ้นบน (Bottom Up Design)

13

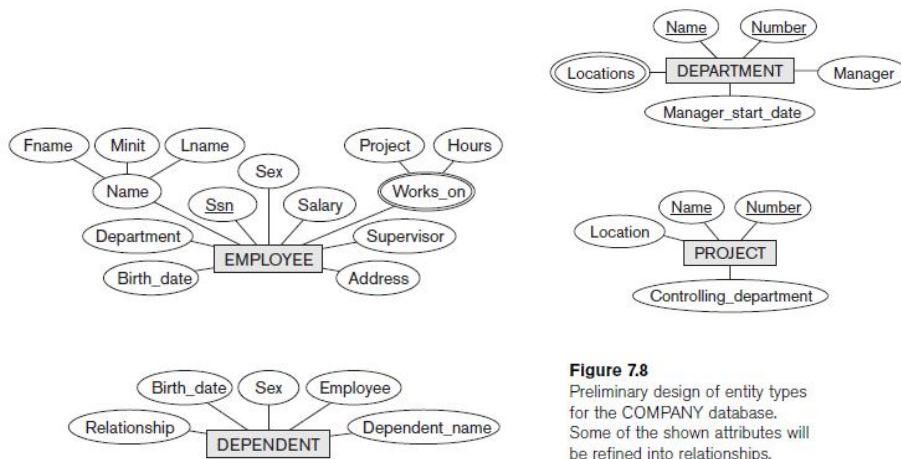
## 1. ออกแบบเอนทิตีและแอททริบิวต์

### a. การออกแบบจากบนลงล่าง

- นึกถึงภาพรวมของสิ่งต่างๆที่อยู่ในระบบที่ต้องการจัดเก็บ โดยที่ยังไม่ต้องนึกถึงรายละเอียดของสิ่งต่างๆเหล่านั้น เช่น ระบบการลงทะเบียนเรียนของนักศึกษา
  - ให้นึกถึง นักศึกษา รายวิชา ผู้สอน คณะ สาขาวิชา เป็นต้น ซึ่งสิ่งต่างๆเหล่านี้คือ **เอนทิตี** ของระบบ
  - หลังจากนั้น ให้นึกถึงคุณสมบัติเฉพาะ หรือ **แอททริบิวต์** ของแต่ละเอนทิตีว่าเราสนใจจะจัดเก็บแอททริบิวต์อะไรบ้างที่สำคัญ และจำเป็นต่อระบบ

14

## ตัวอย่างการออกแบบเอนทิตีและแอททริบิวต์ของฐานข้อมูล COMPANY แบบบนลงล่าง



15

## 1. ออกแบบเอนทิตีและแอททริบิวต์

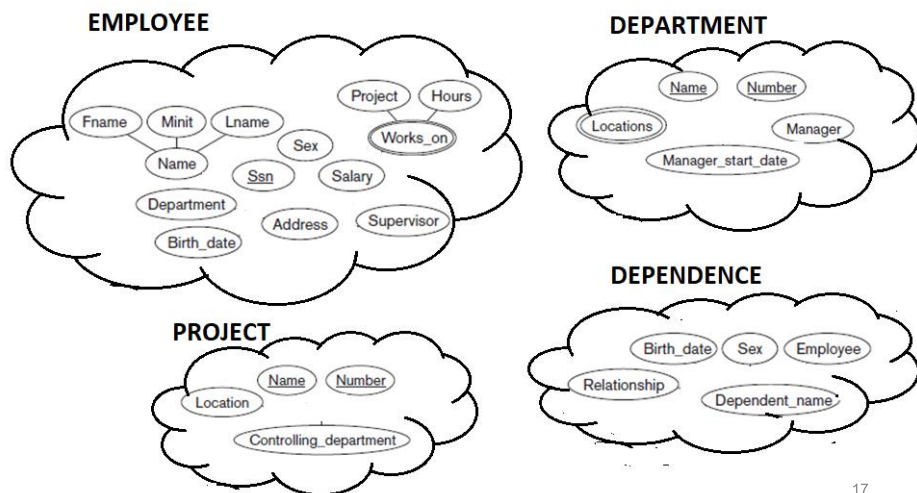
### b. การออกแบบจากล่างขึ้นบน

- ให้นึกถึงสิ่งต่างๆในระบบ และรายละเอียดของสิ่งต่างๆที่ต้องการจัดเก็บในระบบ หรือรายละเอียดต่างๆที่เกี่ยวข้องกับระบบเท่าที่เป็นไปได้ เช่น ระบบการลงทะเบียนเรียนของนักศึกษา
  - ให้นึกถึง นักศึกษา และรายละเอียด (**แอททริบิวต์**) ที่เกี่ยวข้องกับ **นักศึกษา** ไปพร้อมๆกัน ยกตัวอย่างเช่น รหัสนักศึกษา ชื่อ-สกุล ที่อยู่ รายวิชาที่ลงทะเบียน ปีการศึกษา ภาคการศึกษาที่ลงทะเบียน เกรดที่ได้ GPA ของแต่ละภาคการศึกษา ใครเป็นผู้สอนในแต่ละรายวิชา รหัสวิชา ชื่อวิชา หน่วยกิต และข้อมูลอื่นๆที่เกี่ยวข้อง
  - หลังจากนั้น ให้ทำการจัดกลุ่มของรายละเอียดที่กล่าวมาข้างต้น เช่น อะไรที่อยู่ในกลุ่มของนักศึกษา กลุ่มรายวิชา กลุ่มคณะ และให้ตั้งชื่อกลุ่ม ซึ่งจะได้ 1 **เอนทิตี** ต่อ 1 กลุ่ม

16

## ตัวอย่างการออกแบบเอนทิตีและแอททริบิวต์ของฐานข้อมูล

### COMPANY แบบล่างขึ้นบน



17

## การออกแบบเชิงแนวคิดของฐานข้อมูล

หลังจากที่ได้ เอนทิตี และแอททริบิวต์ แล้ว ทั้งสองแนวทาง ให้ดำเนินการตามลำดับขั้นตอนดังต่อไปนี้

1. ให้วิเคราะห์เอนทิตีที่ได้ทั้งหมดว่า ควรจะอยู่ในลักษณะ เอนทิตีแบบปกติ หรือเอนทิตีแบบอ่อน
2. แต่ละเอนทิตี ให้จำแนกประเภทของแอททริบิวต์แต่ละตัว และให้พิจารณาว่า แอททริบิวต์ใดสามารถเป็นคีย์ได้
3. หาความสัมพันธ์ระหว่างเอนทิตี และตั้งชื่อความสัมพันธ์ให้เหมาะสม
4. กำหนดค่าอัตราส่วนความสัมพันธ์
5. วิเคราะห์ข้อกำหนดของการมีส่วนร่วม เป็นความสัมพันธ์ในลักษณะมีส่วนร่วมบางส่วน (Partial Participation) หรือความสัมพันธ์ในลักษณะมีส่วนร่วมทั้งหมด (Total Participation)
6. แปลงอัตราส่วนความสัมพันธ์ และลักษณะของการมีส่วนร่วม ให้อยู่ในรูป (min, max) → ข้อ 7 จะทำหรือไม่ทำก็ได้

18

## 2. วิเคราะห์ประเภทเอนทิตี

- สิ่งที่ต้องคิดถึงสิ่งแรกในการออกแบบเชิงแนวคิดคือ เอนทิตี ซึ่งใช้นำเสนอ สิ่งที่เกิดขึ้น สิ่งที่เราสนใจ หรือสิ่งที่สำคัญที่ต้องการจัดเก็บลงในฐานข้อมูล
  - ไม่จำเป็นต้องจัดเก็บทุกสิ่งที่เกิดขึ้น เช่น ในระบบการขายสินค้าออนไลน์ ถ้าไม่ได้สนใจเกี่ยวกับโปรโมชั่น (Promotion) ก็ไม่จำเป็นต้องจัดเก็บข้อมูลโปรโมชั่นเข้าไปในฐานข้อมูล เป็นต้น
- เอนทิตี แบ่งออกเป็น 2 ประเภท คือ
  - เอนทิตีแบบปกติ (Regular Entity) หรือเรียกว่า เอนทิตีแบบแข็ง (แข็งแรง) หรือ Strong Entity
  - เอนทิตีแบบอ่อน (อ่อนแอ) หรือ Weak Entity

19

## 2. วิเคราะห์ประเภทเอนทิตี (ต่อ)

- เอนทิตีแบบปกติ คือ เอนทิตีที่ดำรงอยู่ได้ด้วยตัวเอง หรือเป็นอิสระจากเอนทิตีอื่น เช่น เอนทิตีพนักงาน เอนทิตีแผนก เป็นต้น
- เอนทิตีแบบอ่อน คือ เอนทิตีที่ไม่สามารถอยู่ได้ด้วยตัวเอง ถ้าไม่มีเอนทิตีที่เป็นเจ้าของ (Owner) หรือเอนทิตีที่เป็นต้นกำเนิด (Parent) ก็จะไม่มี เอนทิตีแบบอ่อน เช่น
  - ถ้าไม่มีรายการสั่งซื้อสินค้า (Order) ก็จะไม่มียาวละเอียดการสั่งซื้อสินค้า (Order Item) เป็นต้น
  - เอนทิตีแบบอ่อน จะไม่มี คีย์แอททริบิวต์ (Key Attribute) เป็นของตัวเอง แต่จะมีคีย์แอททริบิวต์ของเอนทิตีต้นกำเนิด ร่วมกับ คีย์แอททริบิวต์ของตนเอง จึงจะสามารถแยกแยะความเป็นเอกลักษณ์ของตนเองได้

20

### 3. จำแนกประเภทของแอททริบิวต์

- **แอททริบิวต์** คือ คุณสมบัติเฉพาะที่สามารถใช้อธิบายเอนทิตีได้ เช่น
  - คุณสมบัติเฉพาะที่บ่งบอกถึงนักเรียน คือ รหัสนักเรียน ชื่อ-สกุลนักเรียน GPA เป็นต้น
  - ทั้งนี้ให้นึกถึงเฉพาะคุณสมบัติที่เราสนใจ หรือสำคัญต่อระบบของเราเท่านั้น เช่น
    - รหัสนักเรียน ชื่อ-สกุล มีความจำเป็นและสำคัญในระบบงานทะเบียนนักเรียน จึงจำเป็นต้องจัดเก็บในฐานข้อมูล
    - หาก งานอดิเรก และค่าขนม ไม่จำเป็นกับระบบงานทะเบียนนักเรียน คุณสมบัติเหล่านี้ก็ไม่จำเป็นต้องจัดเก็บ

21

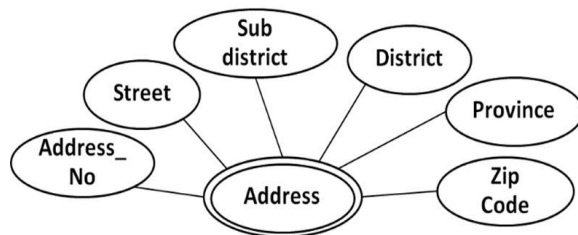
### 3. จำแนกประเภทของแอททริบิวต์

- **ประเภทแอททริบิวต์**
  - **แอททริบิวต์เชิงเดี่ยว (Simple or Atomic)** คือ แอททริบิวต์ที่ไม่สามารถแบ่งแยกได้อีก เช่น รหัสนักศึกษา รหัสไปรษณีย์ เป็นต้น
  - **แอททริบิวต์เชิงประกอบ (Composite Attribute)** คือ แอททริบิวต์ที่สามารถแบ่งแยกออกเป็นส่วนที่ย่อยลงไปได้อีก เช่น ที่อยู่ สามารถแบ่งย่อยออกเป็น บ้านเลขที่ ถนน ตำบล อำเภอ จังหวัด รหัสไปรษณีย์ เป็นต้น
  - **แอททริบิวต์ที่มีค่าข้อมูลได้หลายค่า (Multivalued Attribute)** เช่น หมายเลขโทรศัพท์ของผู้ปกครอง ซึ่งโรงเรียนอาจจะสนใจจัดเก็บทั้ง หมายเลข บิดา และมารดา เป็นต้น
  - **แอททริบิวต์ที่สามารถหาค่าข้อมูลได้จากแอททริบิวต์อื่น (Derived Attribute)** เช่น อายุ สามารถคำนวณจาก วันที่ปัจจุบัน - วันเกิด เป็นต้น

22

### 3. จำแนกประเภทของแอททริบิวต์

- **ประเภทแอททริบิวต์ (ต่อ)**
  - **แอททริบิวต์เชิงซ้อน (Complex Attribute)** คือ แอททริบิวต์ที่เกิดจาก แอททริบิวต์เชิงประกอบ รวมกับ แอททริบิวต์ที่มีค่าข้อมูลหลายค่า เช่น บางระบบต้องการจัดเก็บข้อมูลที่อยู่ (Address) มากกว่า 1 ที่ และข้อมูลที่อยู่เป็นแอททริบิวต์เชิงประกอบ ดังตัวอย่างแสดงด้านล่าง



23

### 3. จำแนกประเภทของแอททริบิวต์

- **ประเภทแอททริบิวต์ (ต่อ)**
  - **คีย์แอททริบิวต์ (Key attribute)** คือ แอททริบิวต์ในเอนทิตีที่มีค่าข้อมูลไม่ซ้ำกัน เช่น รหัสนักเรียนของแต่ละคนจะมีค่าไม่ซ้ำกัน ในทุก ๆ เอนทิตีจะต้องมีแอททริบิวต์ที่เป็นคีย์แอททริบิวต์
    - คีย์ (Key) คือกฎข้อบังคับของข้อมูลที่มีค่าไม่ซ้ำกัน (Uniqueness Constraint) หรือเรียกอีกอย่างหนึ่งว่า ข้อมูลเอกลักษณ์
  - แต่ละเอนทิตีจะต้องมีแอททริบิวต์ที่มีค่าข้อมูลไม่ซ้ำกัน

24



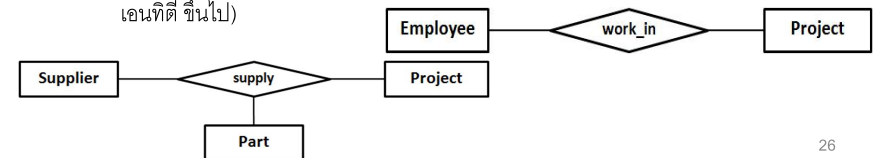
## 4. วิเคราะห์ความสัมพันธ์ระหว่างเอนทิตี

- **ความสัมพันธ์ (Relationship)** ใช้แสดงความสัมพันธ์ที่เกิดขึ้นระหว่างเอนทิตี โดยตั้งชื่อความสัมพันธ์ให้สอดคล้องกับความต้องการในการจัดเก็บข้อมูลในฐานข้อมูล เช่น ถ้ามีเอนทิตี ลูกค้า กับ สินค้า
  - ให้เราตั้งคำถามก่อนว่าต้องการทราบหรือไม่ว่ามีลูกค้าคนใดซื้อสินค้าอะไรในระบบ ถ้าต้องการทราบ เราก็สร้างความสัมพันธ์ระหว่าง ลูกค้า กับ สินค้า ดังนี้คือ
    - ลูกค้า ชื่อ สินค้า
  - และให้พิจารณาว่ามีข้อมูลที่ต้องจัดเก็บเพิ่มเติมเกี่ยวกับการซื้อสินค้าหรือไม่ เช่น
    - ลูกค้าซื้อสินค้าวันไหน ชื่อร้าน เป็นต้น
  - และข้อมูลการซื้อสินค้าจะถูกเชื่อมโยงกับข้อมูลลูกค้า โดยปกติแล้วความสัมพันธ์เกิดขึ้นเนื่องจากแอททริบิวต์ในเอนทิตีหนึ่ง มีการอ้างอิงถึงข้อมูลในเอนทิตีอื่น เช่น ลูกค้ารหัส xxx ซื้อสินค้ารหัส yyy เป็นต้น

25

## 4. วิเคราะห์ความสัมพันธ์ระหว่างเอนทิตี

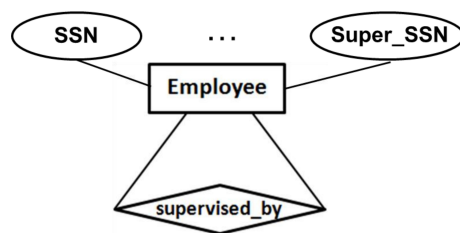
- **ความสัมพันธ์ (ต่อ)**
  - ให้วิเคราะห์ว่าความสัมพันธ์ระหว่างเอนทิตี เป็น **ความสัมพันธ์แบบกี่ฝั่ง/ทาง (Relation Degree)** เช่น
    - 2 (Binary Relationship) คือความสัมพันธ์แบบ 2 ทาง
    - 3 (Ternary Relationship) คือความสัมพันธ์แบบ 3 ทาง
    - $n$  ( $n$ -ary Relationship) โดยที่  $n$  คือ ตัวเลขจำนวนเอนทิตีที่สัมพันธ์กัน,  $n \geq 4$
  - **การตั้งชื่อความสัมพันธ์**
    - ควรตั้งตามบทบาทหน้าที่ของความสัมพันธ์
    - นอกจากใช้บทบาทแล้ว ชื่อความสัมพันธ์ยังบอกถึงการมีส่วนร่วม หรือมีความเกี่ยวข้องกันระหว่างเอนทิตีแต่ละคู่ หรือแต่ละกลุ่ม (กรณีที่เป็นความสัมพันธ์ตั้งแต่ 3 เอนทิตี ขึ้นไป)



26

## 4. วิเคราะห์ความสัมพันธ์ระหว่างเอนทิตี

- **ความสัมพันธ์ (ต่อ)**
  - ให้วิเคราะห์ว่าความสัมพันธ์ระหว่างเอนทิตี เป็น **ความสัมพันธ์แบบวงกลับ (Recursive Relationship)** หรือไม่ คือ การที่เอนทิตีเดียวกันมีความสัมพันธ์กันเอง
  - เกิดจากการที่มี 2 แอททริบิวต์ มีลักษณะเหมือนกัน แต่มีบทบาทต่างกัน เช่น เอนทิตีพนักงาน มีแอททริบิวต์ หมายเลขประกันสังคม 2 หมายเลข
    - หมายเลขประกันสังคมของพนักงาน (SSN)
    - หมายเลขประกันสังคมของหัวหน้างาน (Super\_SSN)



27

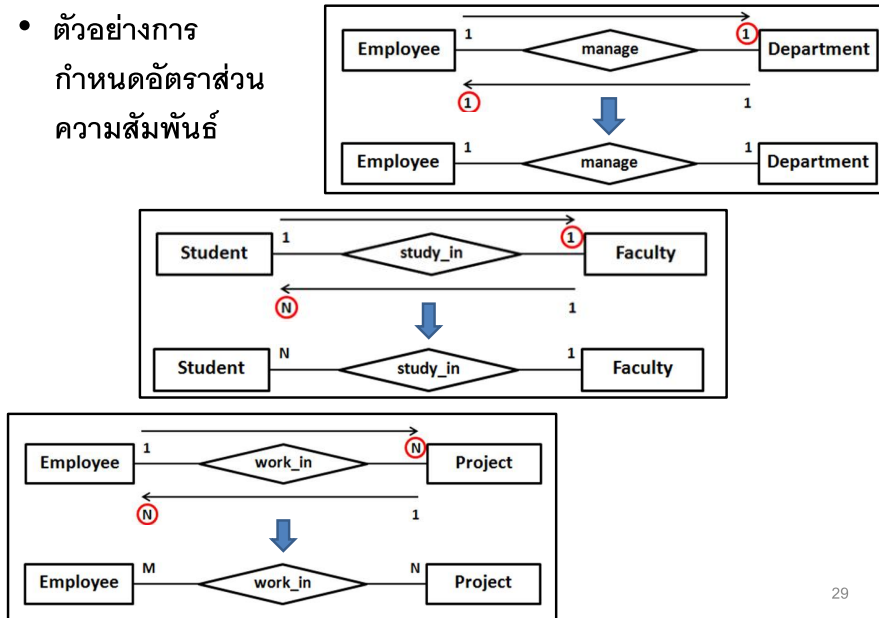
## 5. กำหนดค่าอัตราส่วนความสัมพันธ์

- **อัตราส่วนความสัมพันธ์ระหว่างข้อมูลของสองเอนทิตี (Cardinality Ratio)**
  - ให้วิเคราะห์อัตราส่วนความสัมพันธ์ระหว่างข้อมูลของเอนทิตีในแต่ละฝั่ง โดยระบุค่าอัตราส่วนที่มากที่สุดที่สัมพันธ์กับฝั่งตรงข้าม ซึ่งค่าที่เป็นไปได้ มีดังนี้คือ
    - One-to-one หรือ 1:1 คือความสัมพันธ์แบบ **หนึ่งต่อหนึ่ง**
    - One-to-many หรือ 1:N คือความสัมพันธ์แบบ **หนึ่งต่อหลาย**
    - Many-to-many หรือ M:N คือความสัมพันธ์แบบ **หลายต่อหลาย**
  - ในการกำหนดอัตราส่วนความสัมพันธ์ ให้ทำสองฝั่ง คือ จากซ้ายไปขวา และจากขวาไปซ้าย และคงไว้ซึ่งตัวเลขที่อยู่ปลายลูกศรทั้งสองด้าน
  - สำหรับอัตราส่วนความสัมพันธ์แบบ **N:M** ไม่ควรระบุจำนวนตัวเลขเป็น **N** ทั้งสองฝั่ง เนื่องจากค่าจำนวนตัวเลขที่อยู่ในแต่ละฝั่งไม่จำเป็นต้องมีค่าเท่ากัน จึงควรระบุฝั่งหนึ่งเป็น **M** และอีกฝั่งหนึ่งเป็น **N**

28

## 5. กำหนดค่าอัตราส่วนความสัมพันธ์

- ตัวอย่างการกำหนดอัตราส่วนความสัมพันธ์



29

## 6. วิเคราะห์ข้อกำหนดของการมีส่วนร่วม

- ข้อกำหนดของการมีส่วนร่วม (Participation Constraint)

- ความสัมพันธ์แบบมีส่วนร่วมทั้งหมด (Total Participation) ใช้สัญลักษณ์ **เส้นคู่** เช่น ความสัมพันธ์ระหว่าง พนักงาน บริหาร แผนก โดยที่ผังแผนกเป็นความสัมพันธ์แบบมีส่วนร่วมทั้งหมด หมายความว่า ทุกๆแผนกจะต้องมีผู้จัดการ (พนักงานที่อยู่อีกผังหนึ่ง) มาบริหาร
- ความสัมพันธ์แบบมีส่วนร่วมบางส่วน (Partial Participation) ใช้สัญลักษณ์ **เส้นเดี่ยว** เช่น ความสัมพันธ์ระหว่างพนักงาน บริหาร แผนก โดยที่ผังพนักงานเป็นความสัมพันธ์แบบมีส่วนร่วมบางส่วน หมายความว่า แผนกถูกบริหารโดย พนักงานบางคน (หรือ มีพนักงานบางคน (ไม่ใช่ทุกคน) ที่เป็นผู้จัดการ)



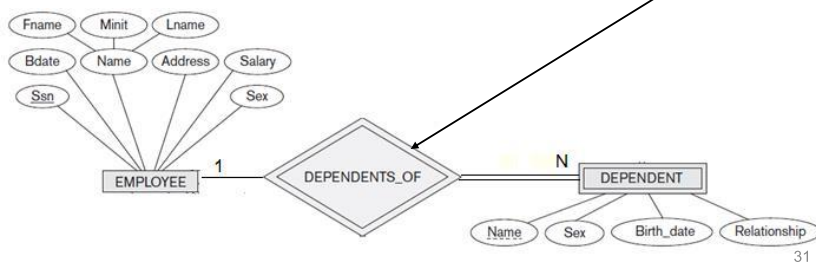
30

## 6. วิเคราะห์ข้อกำหนดของการมีส่วนร่วม

- ข้อกำหนดของการมีส่วนร่วมทางฝั่งที่เป็นเอนทิตีแบบอ่อนจะเป็นแบบมีส่วนร่วมทั้งหมดเสมอ เนื่องจาก ถ้าไม่มีเอนทิตีเจ้าของก็จะมีเอนทิตีแบบอ่อน ดังนั้น เมื่อมีข้อมูลเอนทิตีแบบอ่อนจึงถูกเชื่อมโยง หรือมีส่วนร่วมกับข้อมูลฝั่งเจ้าของทุกข้อมูล

- ความสัมพันธ์ระหว่าง เอนทิตีแบบอ่อน กับ เอนทิตีที่เป็นเจ้าของ เรียกว่า

Identifying Relationship ใช้สัญลักษณ์ **ความสัมพันธ์เส้นคู่**



31

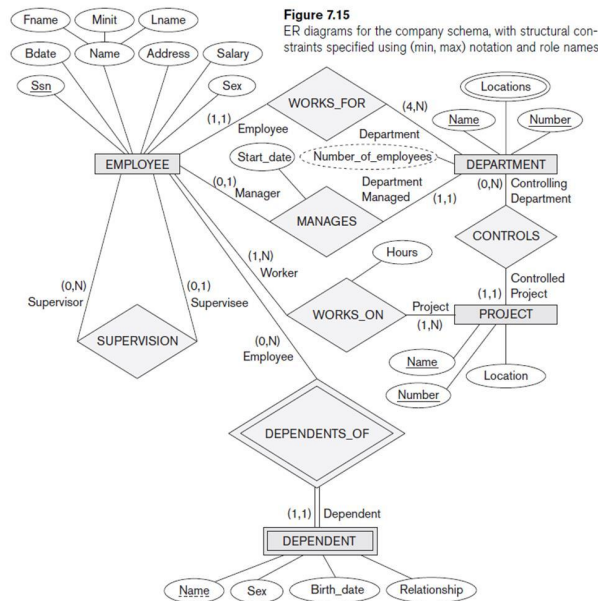
## 7. แปลงอัตราส่วนความสัมพันธ์ และการมีส่วนร่วมให้อยู่ในรูป (min, max)

- (min, max) ใช้บ่งบอกค่าต่ำสุด และสูงสุด ที่เป็นไปได้ของอัตราส่วนความสัมพันธ์ระหว่างเอนทิตี
- ความสัมพันธ์ในลักษณะมีส่วนร่วมทั้งหมด (เส้นคู่) เนื่องจาก มีส่วนร่วมของข้อมูลกับอีกฝั่งหนึ่งทุกข้อมูล แสดงว่า อย่างน้อยต้องมีส่วนร่วมอย่างน้อย 1 ข้อมูล เพราะฉะนั้น ค่าต่ำสุดที่เป็นไปได้คือ ตั้งแต่ 1 เป็นต้นไป ส่วนค่าสูงสุดคงค่าเดิมที่อยู่บนเส้น (ค่าเดิมที่อยู่บนเส้น คือค่าการมีส่วนร่วมที่เป็นไปได้สูงสุด)
- ความสัมพันธ์ในลักษณะมีส่วนร่วมบางส่วน (เส้นเดี่ยว) เนื่องจาก มีส่วนร่วมของข้อมูลกับอีกฝั่งหนึ่งแค่บางส่วน ดังนั้น อาจจะมีส่วนร่วม หรือไม่มีส่วนร่วมเลยก็ได้ เพราะฉะนั้น ค่าต่ำสุดที่เป็นไปได้คือ ตั้งแต่ 0 เป็นต้นไป ส่วนค่าสูงสุดคงค่าเดิมที่อยู่บนเส้น
- หลังจากกำหนดค่า ต่ำสุด และสูงสุด บนเส้นความสัมพันธ์ แล้วให้นำค่า (min, max) สลับฝั่งกันของแต่ละคู่

32



## ตัวอย่างแผนภาพอีอาร์หลังจากกำหนดค่า (min, max)



33

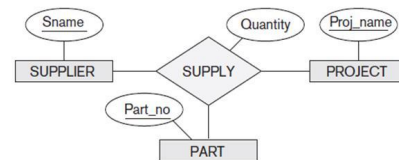
## การตั้งชื่อองค์ประกอบต่างๆในแผนภาพอีอาร์

- การตั้งชื่อให้กับองค์ประกอบต่างๆที่อยู่ในแผนภาพอีอาร์ ควรตั้งชื่อให้เหมาะสม โดยมีแนวทางในการตั้งชื่อดังนี้
  - การตั้งชื่อ เอนทิตี ความสัมพันธ์ แอททริบิวต์ และส่วนอื่นๆ ควรตั้งชื่อให้สื่อความหมายให้ชัดเจน
  - การตั้งชื่อเอนทิตี ควรใช้คำนาม
  - การตั้งชื่อความสัมพันธ์ ควรใช้คำกริยา
  - การตั้งชื่อความสัมพันธ์ระหว่างสองเอนทิตี ให้ตั้งในลักษณะที่อ่านจากซ้ายไปขวา และอ่านจากบนลงล่าง เช่น
    - พนักงาน ทำงานใน โครงการ (Employee work\_on Project)
    - โครงการ ถูกกำหนดให้ พนักงาน (Project is\_assigned\_to Employee)

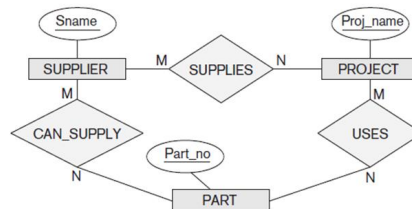
34

## ประเภทของความสัมพันธ์ที่มีมากกว่า 2 ทาง/ฝั่ง

- ในการออกแบบฐานข้อมูล หากความสัมพันธ์เชื่อมโยงมากกว่า 2 เอนทิตีจริง ให้นำเสนอในรูปความสัมพันธ์แบบ 3 ทาง หรือมากกว่า



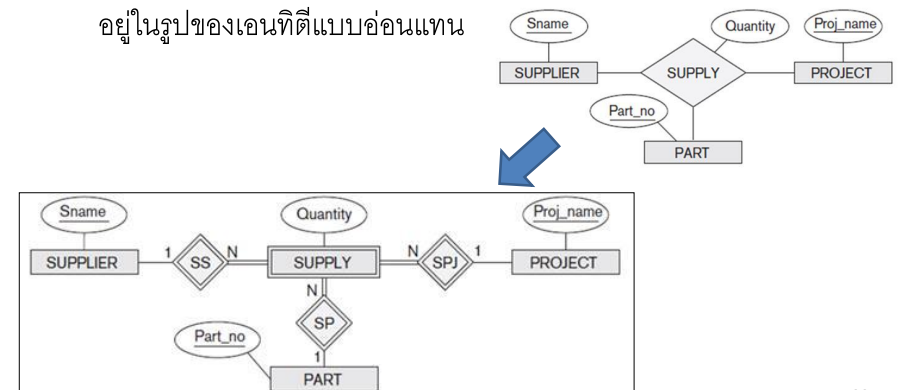
- หากความสัมพันธ์ของเอนทิตีแต่ละคู่เป็นความสัมพันธ์ที่แตกต่างกัน ให้แยกความสัมพันธ์ออกเป็นความสัมพันธ์แบบ 2 ทาง จำนวน 3 คู่



35

## ประเภทของความสัมพันธ์ที่มีมากกว่า 2 ทาง/ฝั่ง

- เครื่องมือที่ใช้ในการออกแบบฐานข้อมูลเชิงสัมพันธ์บางตัว อนุญาตให้ออกแบบความสัมพันธ์ในลักษณะความสัมพันธ์แบบ 2 ทางเท่านั้น
- ในกรณีที่มีความสัมพันธ์มากกว่า 2 ทาง ให้นำเสนอความสัมพันธ์ให้อยู่ในรูปของเอนทิตีแบบอ่อนแทน



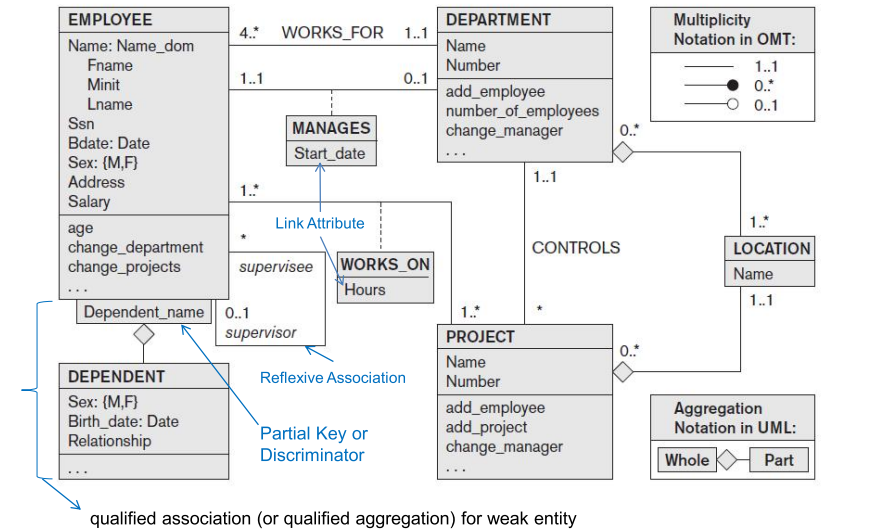
36

## Example of Other Notation: UML Class Diagrams

- UML methodology
  - การออกแบบโดยใช้แนวทาง UML มีการใช้อย่างแพร่หลายในการออกแบบซอฟต์แวร์
  - ใน UML มีหลายๆ Diagrams สำหรับใช้ในวัตถุประสงค์ที่แตกต่างกัน
- UML class diagrams
  - ผลลัพธ์ปลายทางของการออกแบบโดยใช้แนวทาง UML คือ Class Diagram
  - Entity ใน ER Diagram เทียบเท่ากับ Class ใน UML Class Diagram

37

**Figure 7.16**  
The COMPANY conceptual schema in UML class diagram notation.



38

## Example of Other Notation: UML Class Diagrams

- ในแต่ละ Class แบ่งออกเป็น 3 ส่วน
  - ส่วนบนสุดคือ ชื่อ Class
  - ส่วนกลางคือ Attributes หรือ Data
  - ส่วนสุดท้ายคือ Operation หรือ Method
- **Associations:** ความสัมพันธ์ระหว่าง Classes
- **Relationship instances:** ความสัมพันธ์ที่มีแอททริบิวต์เกิดขึ้นจากความสัมพันธ์ระหว่าง Classes แทนด้วยเส้นประ
- **Binary association:** ความสัมพันธ์แบบ 2 ทาง
  - สามารถอยู่ในรูปแบบใดแบบหนึ่งก็ได้ใน 2 แบบข้างต้น
- **Link attribute:** Attribute ที่อยู่ในกล่องที่เชื่อมกับเส้นประ

39

## Example of Other Notation: UML Class Diagrams (cont'd.)

- **Multiplicities:** ใช้สำหรับระบุการมีส่วนร่วม
  - min..max: asterisk (\*) ใช้แสดงจำนวนสูงสุดที่อยู่ในลักษณะไม่จำกัดจำนวนของค่า max
- **Types of relationships:** ประเภทของความสัมพันธ์
  - **Association**
  - **Aggregation:** ความสัมพันธ์แบบกลุ่ม
- **Unidirectional and bidirectional associations**
  - **Unidirectional:** ใช้ลูกศรเป็นตัวบอกความสัมพันธ์จากไหนไปไหน
  - **Bidirectional:** ไม่มีลูกศร เป็นความสัมพันธ์แบบสองทาง
- **Model weak entities using qualified association:** ใช้ Partial Key ในการเชื่อมโยง

40