204320 - Database Management

Chapter 7

Data Modeling Using the Entity-Relationship (ER) Model

Adapted for 204320

by Areerat Trongratsameethong

Addison-Wesley
is an imprint of
PEARSON

Chapter 7 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

Chapter 7 Outline (cont'd.)

- 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

Data Modeling Using the Entity-Relationship (ER) Model

- **Entity-Relationship (ER) model**
  - Popular high-level conceptual data model
- **ER diagrams**
  - Diagrammatic notation associated with the ER model: ER Model ถูกนำเสนอในรูปแบบของ ER Diagram ซึ่งอยู่ในรูปของแบบจำลองข้อมูลในระดับแนวคิด (Conceptual Data Model)
- **Unified Modeling Language (UML):** สำหรับ OO Design

Using High-Level Conceptual Data Models for Database Design

ขั้นตอนการออกแบบฐานข้อมูล

- **Requirements collection and analysis**
  - Database designers จะสัมภาษณ์ user ที่เป็นคนใช้ฐานข้อมูล เพื่อจะได้เข้าใจความต้องการของผู้ใช้ และจะได้ดำเนินการจัดทำเอกสารที่เกี่ยวข้องกับฐานข้อมูลได้ถูกต้อง
  - Result: ผลลัพธ์ที่ได้จากการสัมภาษณ์คือ ข้อมูลที่ผู้ใช้ต้องการ
  - **Functional requirements** of the application: หลังจากนั้นจะดำเนินการจัดทำฟังก์ชันต่างๆของ application ตามความต้องการของผู้ใช้

Using High-Level Conceptual Data Models (cont'd.)

ขั้นตอนการออกแบบฐานข้อมูล (ต่อ)

- **Conceptual schema**
  - ออกแบบฐานข้อมูลในระดับแนวคิด concept เพื่ออธิบายข้อมูลตามความต้องการของผู้ใช้ฐานข้อมูล
  - ออกแบบรายละเอียดของประเภท entity ความสัมพันธ์ระหว่าง entity และกำหนดข้อบังคับของข้อมูล
  - เปลี่ยนจาก high-level data model ไปเป็น conceptual data model

Using High-Level Conceptual Data Models (cont'd.)

ขั้นตอนการออกแบบฐานข้อมูล (ต่อ)

- **Logical design** or **data model mapping**
  - จากการทำ Logical design หรือ data model mapping ผลลัพธ์ที่ได้คือ Database Schema
- **Physical design phase**
  - ในส่วนของ Physical Design เป็นการกำหนดโครงสร้างของที่ที่ใช้จัดเก็บข้อมูล, องค์ประกอบของไฟล์ข้อมูล, index, access path, และ parameters ต่างๆ ที่เกี่ยวข้องกับ physical design สำหรับ database file

A Sample Database Application

- COMPANY
  - Employees, departments, and projects
  - Company is organized into departments
  - Department controls a number of projects
  - Employee: store each employee's name, Social Security number, address, salary, sex (gender), and birth date
  - Keep track of the dependents (บุตรที่อยู่ภายใต้การดูแลของพนักงาน) of each employee

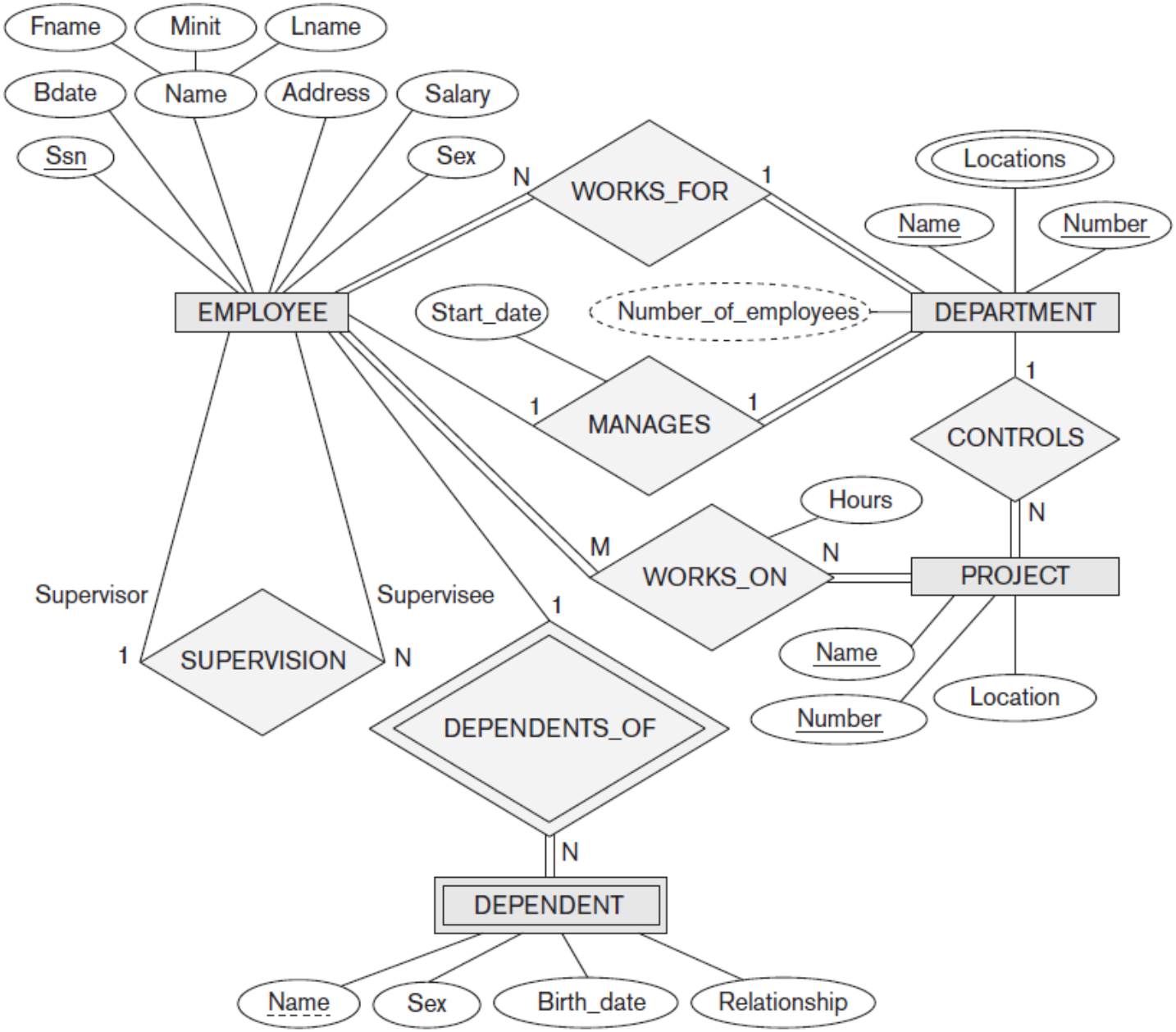


Figure 7.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.

Entity Types, Entity Sets, Attributes, and Keys

- ER model describes data as:
  - Entities: สิ่งต่างๆที่เกิดขึ้นจริงในโลกที่ดำรงอยู่ได้ด้วยตัวเอง หรือเป็นอิสระจากสิ่งอื่น
  - Relationships: ความสัมพันธ์ระหว่าง entity
  - Attributes:
    - คุณสมบัติเฉพาะที่สามารถอธิบาย entity ได้
    - Types of attributes:
      - *Composite* versus *simple* (atomic) *attributes*
      - **Single-valued** versus **multivalued** attributes
      - **Stored** versus **derived** attributes
      - **NULL** values
      - **Complex** attributes

Entity Types, Entity Sets, Keys, and Value Sets

- **Entity type**
  - Collection (or set) of entities that have the same attributes

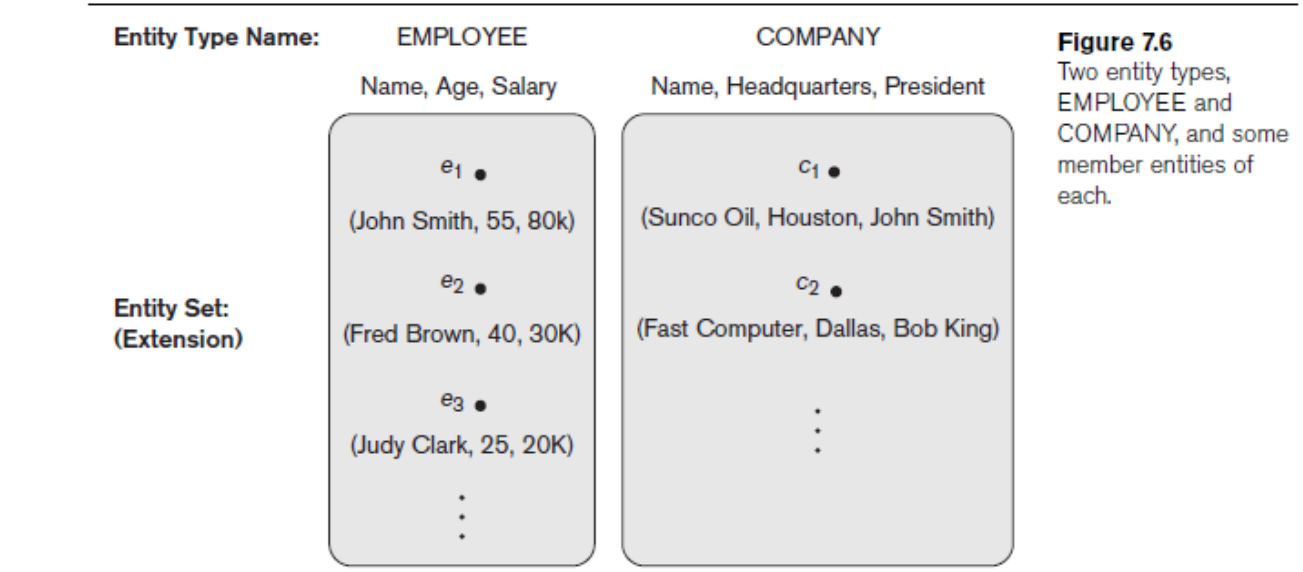


Figure 7.6
Two entity types, EMPLOYEE and COMPANY, and some member entities of each.

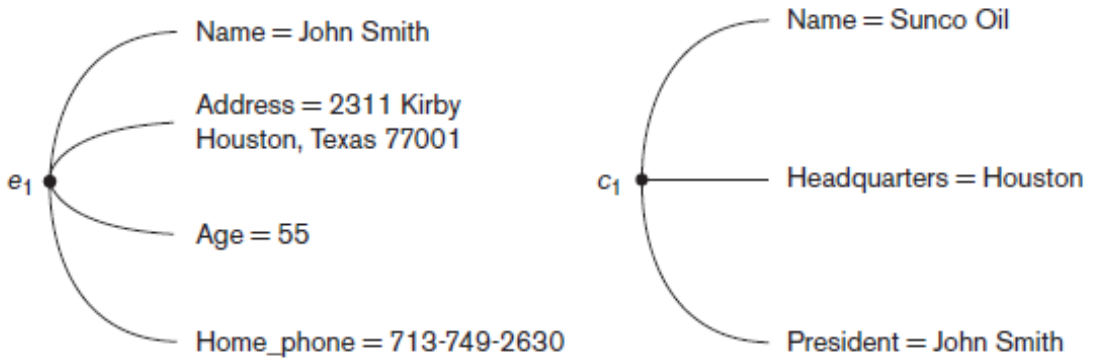


Figure 7.3
Two entities, EMPLOYEE e_1, and COMPANY c_1, and their attributes.

Entity Types, Entity Sets, Keys, and Value Sets (cont'd.)

- **Key** or **uniqueness constraint**
  - Key คือ Attribute(s) ในแต่ละ entity ที่มีค่าข้อมูลที่ไม่ซ้ำกัน (unique)
  - **Key attribute**
    - ทุกๆ entity จะต้องมี key attribute
- **Value sets** (or **domain of values**)
  - ค่าข้อมูลที่ถูกกำหนดให้กับ attribute ในแต่ละ entity

Initial Conceptual Design of the COMPANY Database

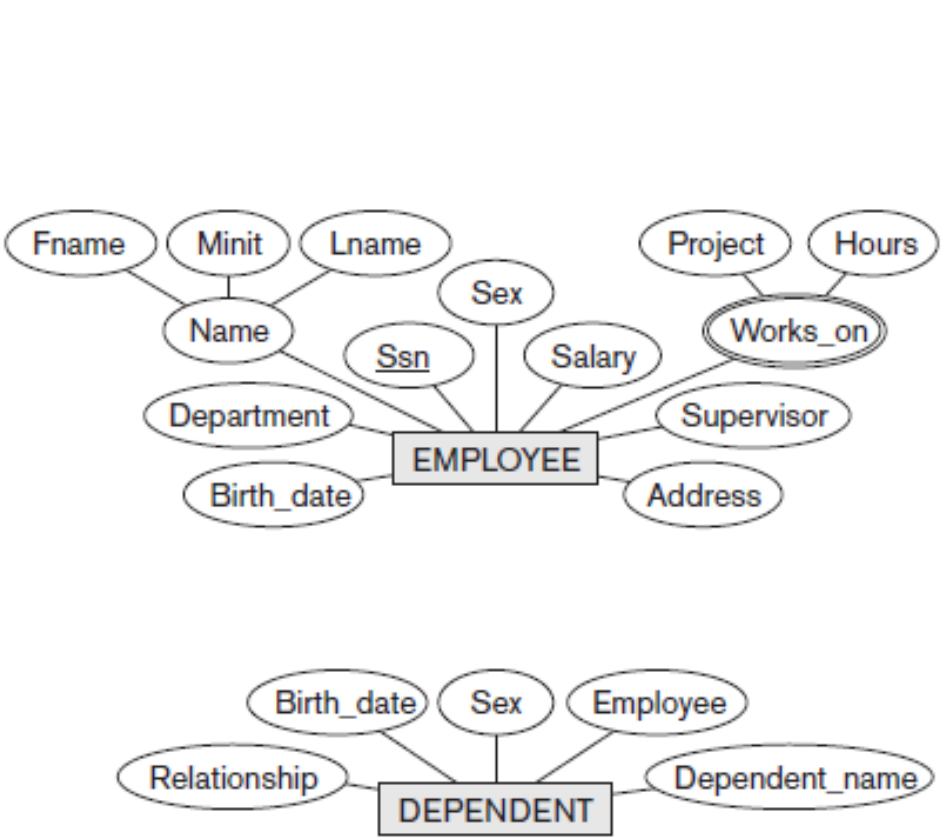


Figure 7.8
Preliminary design of entity types for the COMPANY database. Some of the shown attributes will be refined into relationships.

Relationship Types, Relationship Sets, Roles, and Structural Constraints

- **Relationship**
  - When an attribute of one entity type refers to another entity type: ความสัมพันธ์ระหว่าง entity จะถูกสร้างเมื่อ attribute ใน entity หนึ่งมีการอ้างอิงถึงข้อมูลใน entity อื่น
  - Represent references as relationships not attributes: ให้นำเสนอการอ้างอิงข้อมูลระหว่าง entity ด้วยความสัมพันธ์ ไม่ใช่อ้างอิงในรูปแบบของ attribute แต่ตอนที่คิด และออกแบบให้นึกถึงการเชื่อมโยงระหว่าง 2 entity ว่าเชื่อมโยงกันด้วย attribute ไหน (แต่ไม่ใช่เอา attribute มาเป็น relation)

Relationship Degree

- **Degree** of a relationship type
  - Number of participating entity types
  - **Binary**, **ternary, n-ary**
  - ความสัมพันธ์ระหว่าง **entity** เป็นไปได้ตั้งแต่ 2 ทางขึ้นไป

ตัวอย่าง ternary relationship

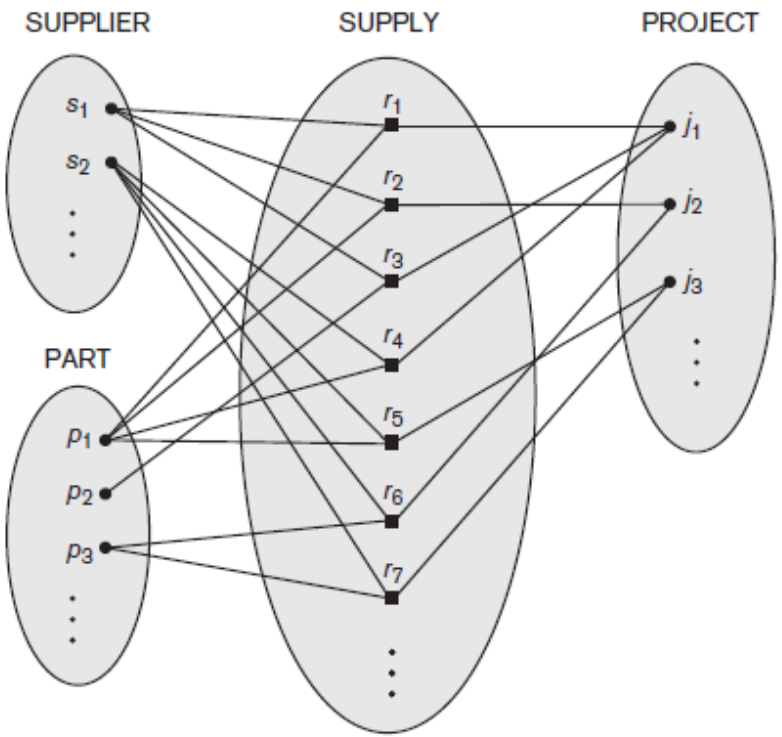


Figure 7.10
Some relationship instances in the SUPPLY ternary relationship set.

Role Names and Recursive Relationships

- **Role names** and recursive relationships
  - Role name signifies role that a participating entity plays in each relationship instance: ชื่อ relationship แต่ละชื่อ ใช้บอกบทบาทการมีส่วนร่วม หรือมีความเกี่ยวข้องกันระหว่าง entity แต่ละคู่ หรือแต่ละกลุ่ม (ในกรณี ternary ขึ้นไป)
- **Recursive** relationships
  - Same entity type participates more than once in a relationship type in different roles
  - Must specify role name
  - กรณีที่ 1 คู่ของ entity มีความสัมพันธ์มากกว่า 1 แบบ เช่น ความสัมพันธ์แบบ 2 ทาง ไป-กลับ ที่ไม่เหมือนกัน ให้กำหนดชื่อให้ทั้งสองความสัมพันธ์เพื่อบอกความแตกต่าง

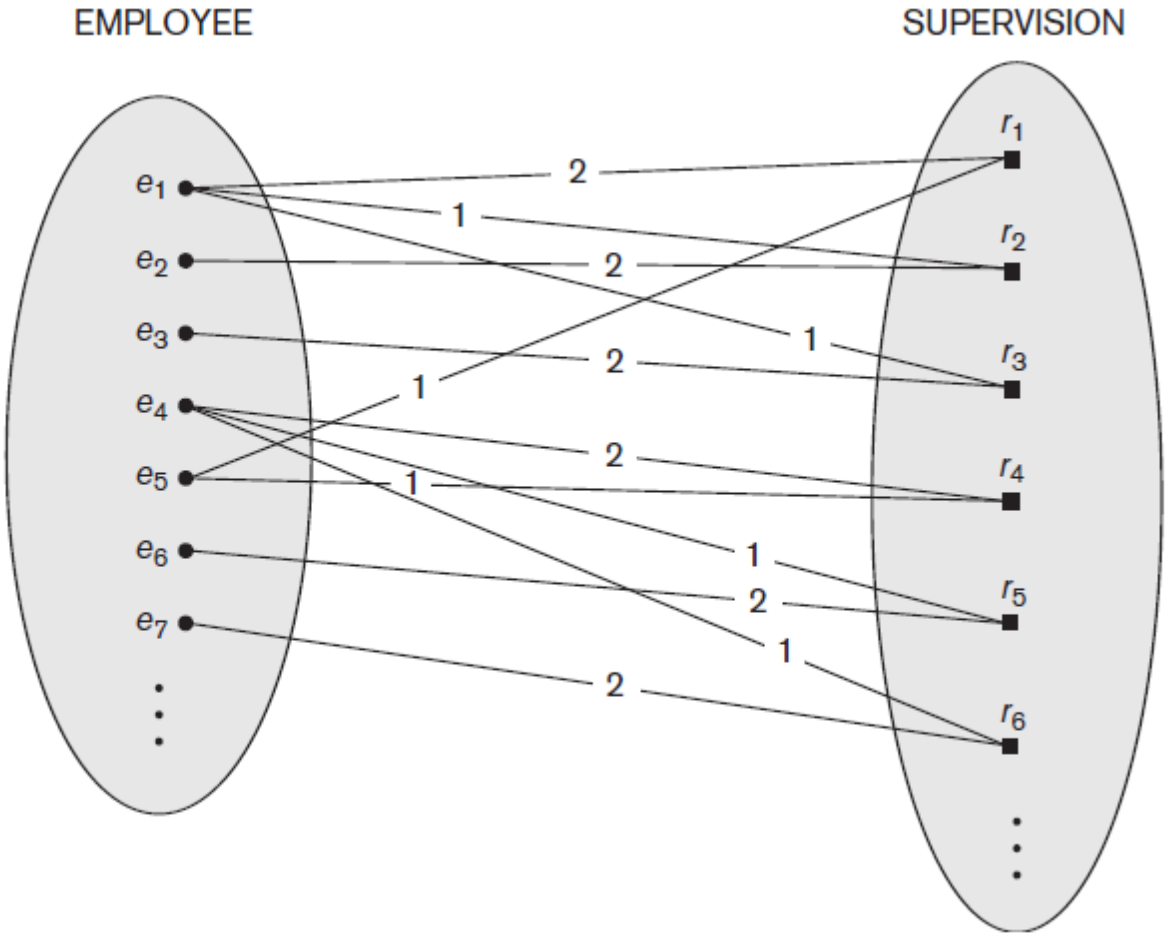


Figure 7.11
A recursive relationship SUPERVISION between EMPLOYEE in the *supervisor* role (1) and EMPLOYEE in the *subordinate* role (2).

<u>บทบาท</u>

1 = supervisor
2 = supervisee

e.g. r_1 แสดงความสัมพันธ์ระหว่าง e_5 และ e_1

- e_5 is a supervisor of e_1,
- e_1 is a supervisee of e_5

Constraints on Binary Relationship Types

- **Cardinality ratio** for a binary relationship
  - Specifies maximum number of relationship instances that entity can participate in
    - 1:1 called One-to-One Relationship
    - 1:N called One-to-Many Relationship
    - M:N called Many-to-Many Relationship
- **Participation constraint**
  - Specifies whether existence of entity depends on its being related to another entity
  - **Total Participation**: Every employee must be related to department (พนักงานทุกคนจะต้องมีแผนกสังกัด). EMPLOYEE work_for DEPARTMENT (total ทั้งไปและกลับ)
  - **Partial Participation:** EMPLOYEE supervison EMPLOYEE พนักงานบางคน supervise (ให้คำปรึกษาในฐานะ supervisor) แต่พนักงานทุกคนไม่ใช่ supervisor

Weak Entity Types

- Do not have key attributes of their own: คือ entity ที่ไม่มี key attribute เป็นของตัวเอง แต่ key attribute มาจาก entity ที่สัมพันธ์กัน
  - Identified by being related to specific entities from another entity type
- **Identifying relationship:** ใช้เชื่อมโยงความสัมพันธ์ระหว่าง weak entity กับ entity ที่สัมพันธ์กัน
  - Relates a weak entity type to its owner
- Always has a total participation constraint: สำหรับ Identifying relationship จะอยู่ในรูปความสัมพันธ์แบบ total participation

Refining the ER Design for the COMPANY Database

- Change attributes that represent relationships into relationship types
- Determine cardinality ratio and participation constraint of each relationship type

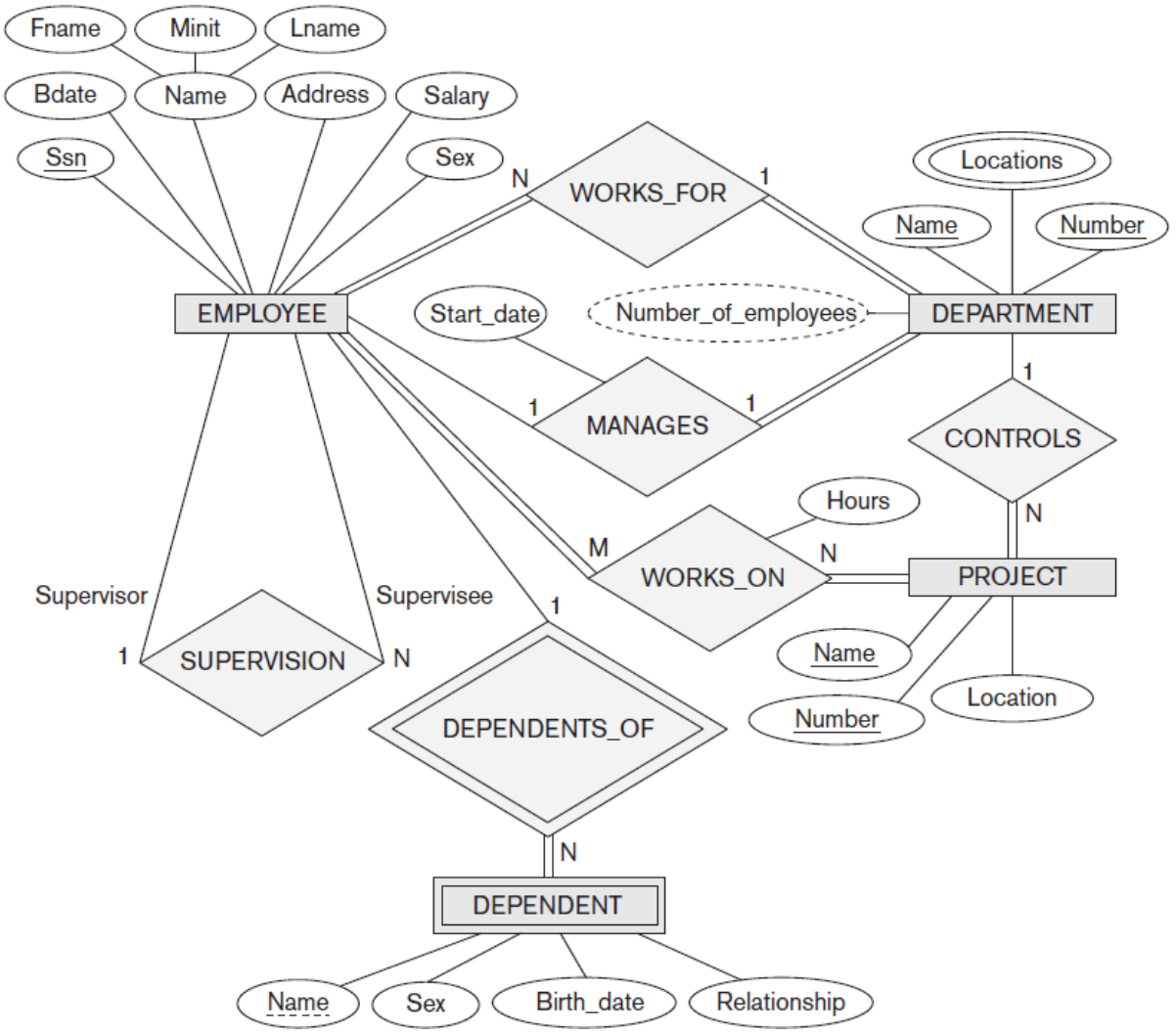


Figure 7.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.

ER Diagrams, Naming Conventions, and Design Issues

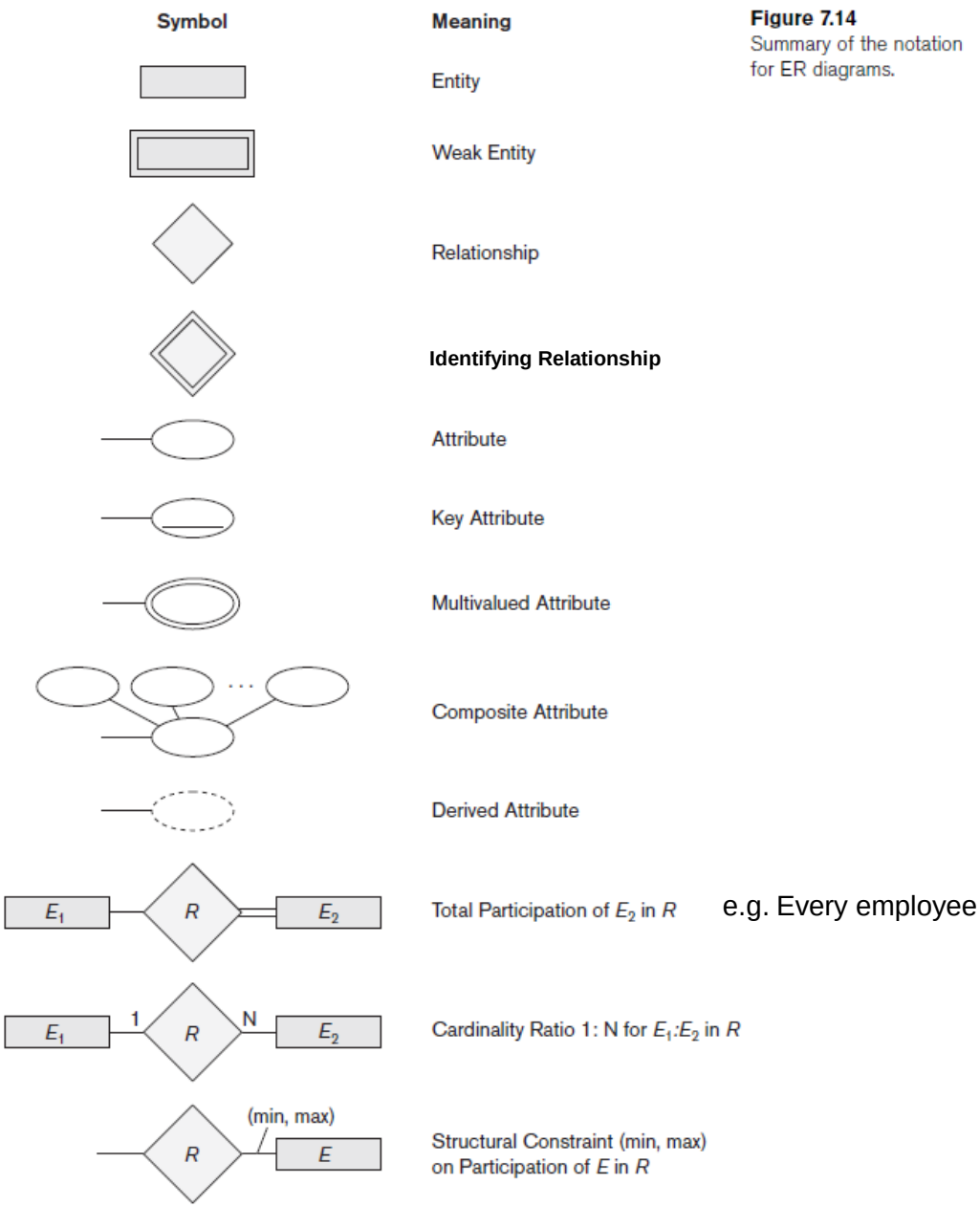


Figure 7.14
Summary of the notation for ER diagrams.

e.g. Every employee must be related to department.

Proper Naming of Schema Constructs

- ตั้งชื่อให้สื่อความหมายให้ชัดเจน
- สำหรับชื่อ entity ให้ใช้คำนาม
- สำหรับชื่อ relationship ให้ใช้กริยา
- ชื่อ binary relationship ให้ตั้งในลักษณะที่อ่านจากซ้ายไปขวาและบนลงล่าง เช่น EMPLOYEE work_for DEPARTMENT

Design Choices for ER Conceptual Design

- Model concept first as an attribute
  - Refined into a relationship if attribute is a reference to another entity type
  - ในการออกแบบ ER ตอนแรกให้นึกถึง attribute ในแต่ละ entity ต่อจากนั้นให้ดูว่า attribute ใน entity หนึ่งมีความสัมพันธ์กับข้อมูลใน entity อื่นหรือไม่ หากมีความสัมพันธ์กัน ให้สร้างความสัมพันธ์โดยดูจาก attribute ที่มีความเกี่ยวข้องกัน
- Attribute that exists in several entity types may be elevated to an independent entity type:
  - attribute ที่มีส่วนร่วมอยู่ในหลายๆ entity เช่น attribute ของ department ให้แยก attribute ดังกล่าวออกจาก entity ต่างๆ และนำไปสร้างเป็น department entity แทน
  - ในทางกลับกัน ถ้าใน entity มีเพียง 1 attribute และสัมพันธ์กับ 1 entity เท่านั้น ให้ย้าย attribute ดังกล่าวไปไว้ใน entity ที่มีความสัมพันธ์กัน และยุบ entity ที่มี 1 attribute ดังกล่าว

Alternative Notations for ER Diagrams

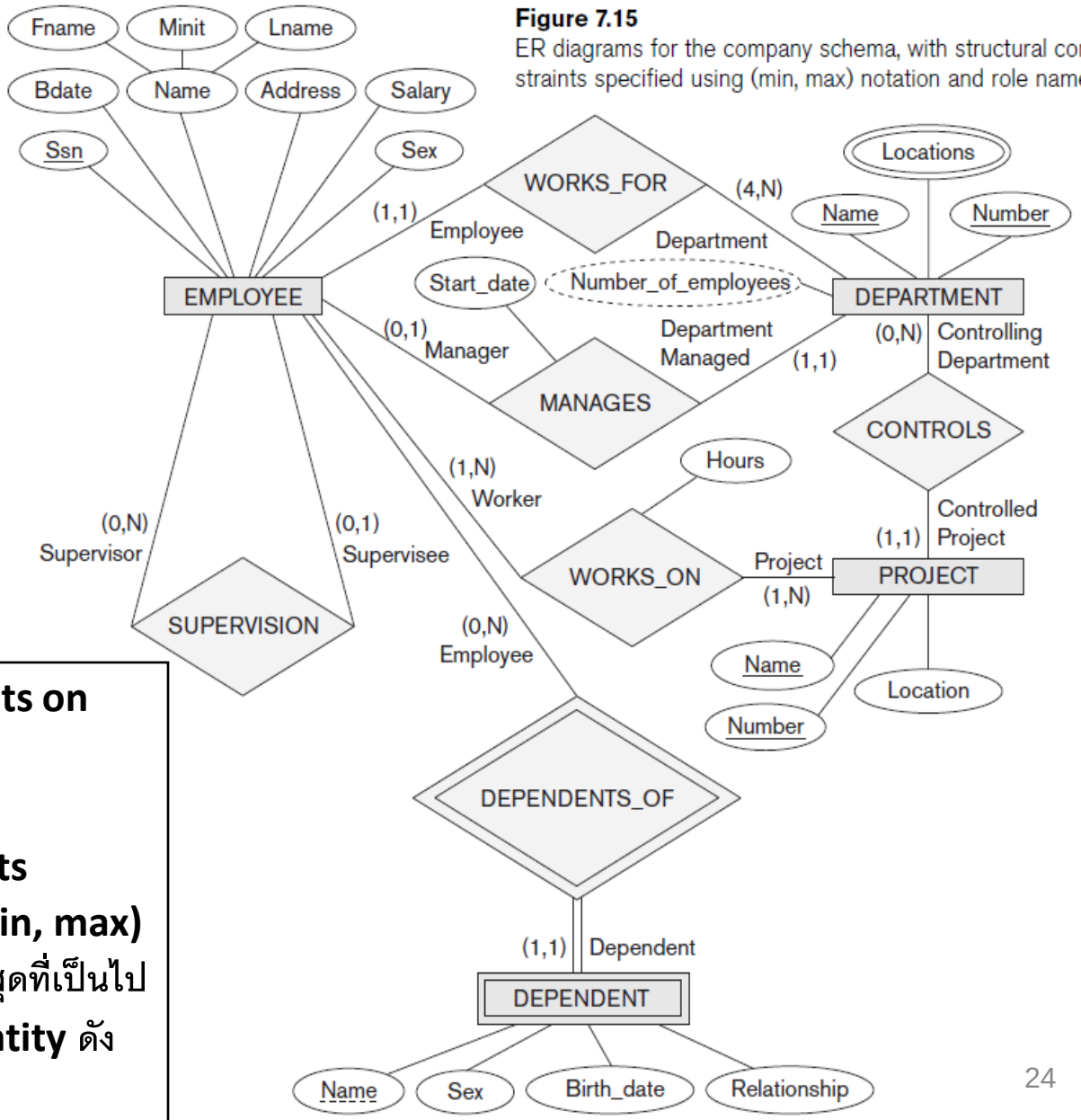


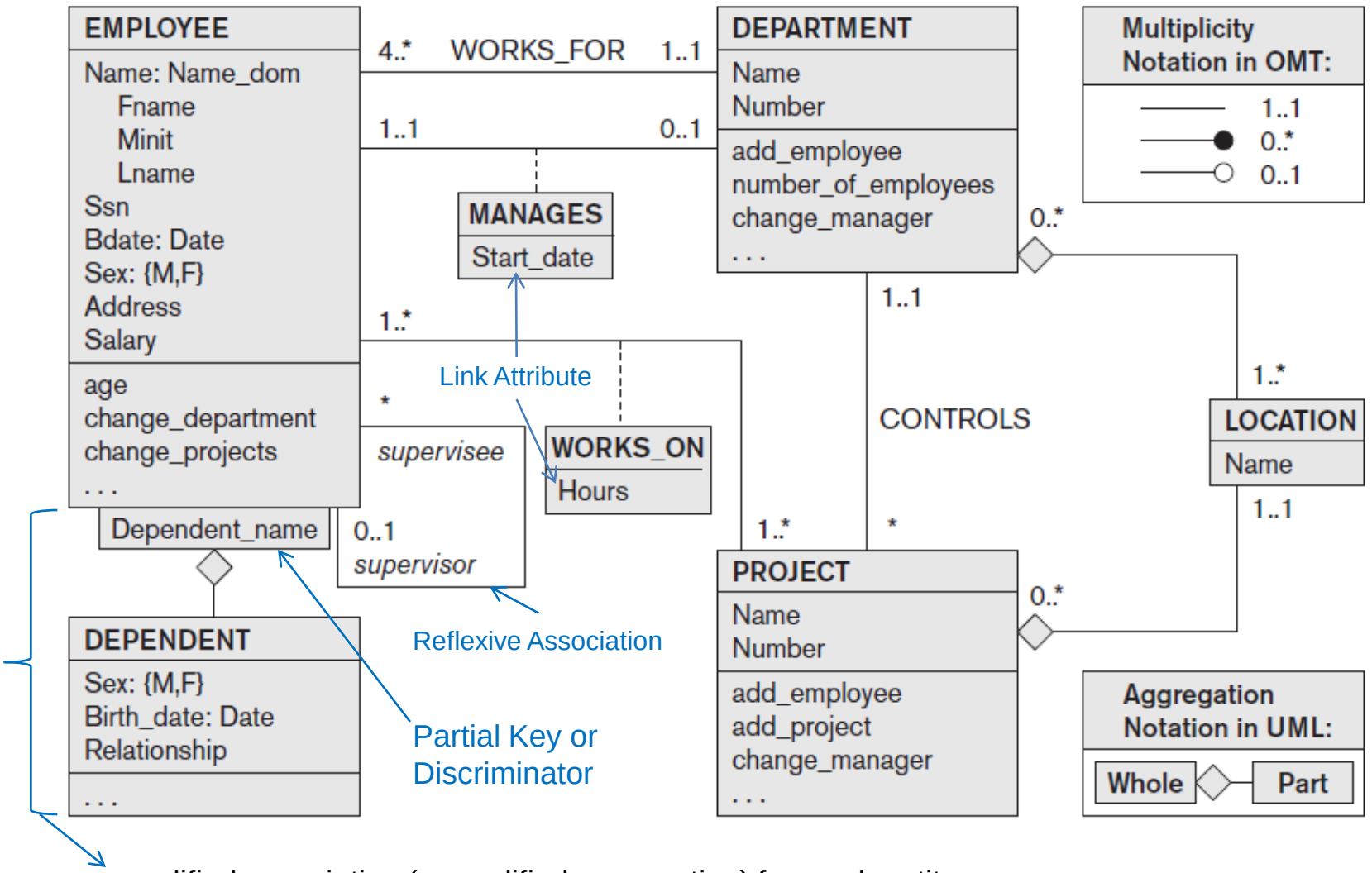
Figure 7.15
ER diagrams for the company schema, with structural constraints specified using (min, max) notation and role names.

- **Specify structural constraints on relationships**
  - **cardinality ratio** และ **participation constraints** สามารถถูกแทนที่ด้วยค่า **(min, max)** ที่ใช้บ่งบอกค่าต่ำสุด และสูงสุดที่เป็นไปของความสัมพันธ์ระหว่าง **entity** ดังแสดงในรูปที่ 7.15

Example of Other Notation: UML Class Diagrams

- UML methodology
  - Used extensively in software design: ใช้อย่างกว้างขวางใน software design
  - Many types of diagrams for various software design purposes: มีหลายๆ diagrams สำหรับวัตถุประสงค์ที่แตกต่างกันของ software ที่แตกต่างกัน
- UML class diagrams
  - Entity in ER corresponds to an object in UML: Entity ใน ER diagram เทียบเท่ากับ object ใน UML diagram

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



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

- **Class** includes three sections: ในแต่ละ class แบ่งออกเป็น 3 ส่วน
  - Top section gives the class name: ส่วนบนสุดคือ ชื่อ class
  - Middle section includes the attributes: ส่วนกลางคือ attributes
  - Last section includes operations that can be applied to individual objects: ส่วนสุดท้ายคือ operation หรือ method

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

- **Associations**: relationship types
- **Relationship instances**: links
- Binary association
  - Represented as a line connecting participating classes
  - May optionally have a name
- Link attribute
  - Placed in a box connected to the association's line by a dashed line

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

- **Multiplicities**: min..max, asterisk (*) indicates no maximum limit on participation
- Types of relationships: **association** and **aggregation**
- Distinguish between **unidirectional** and **bidirectional** associations
  - Unidirectional: ใช้ลูกศรเป็นตัวบอกความสัมพันธ์จากไหนไปไหน
  - Bidirectional: ไม่มีลูกศร เป็นความสัมพันธ์แบบสองทาง
- Model weak entities using **qualified association**

Choosing between Binary and Ternary (or Higher-Degree) Relationships

- Some database design tools permit only binary relationships
  - Ternary relationship must be represented as a weak entity type
  - No partial key and three identifying relationships
- Represent ternary relationship as a regular entity type
  - By introducing an artificial or surrogate key
  - เปลี่ยน relationship ให้เป็น entity และสร้าง key ขึ้นมาใหม่ เช่น จากรูปที่ 7.17 a สร้าง entity ชื่อ SUPPLY และสร้าง key Supply_Id ขึ้นมาใหม่

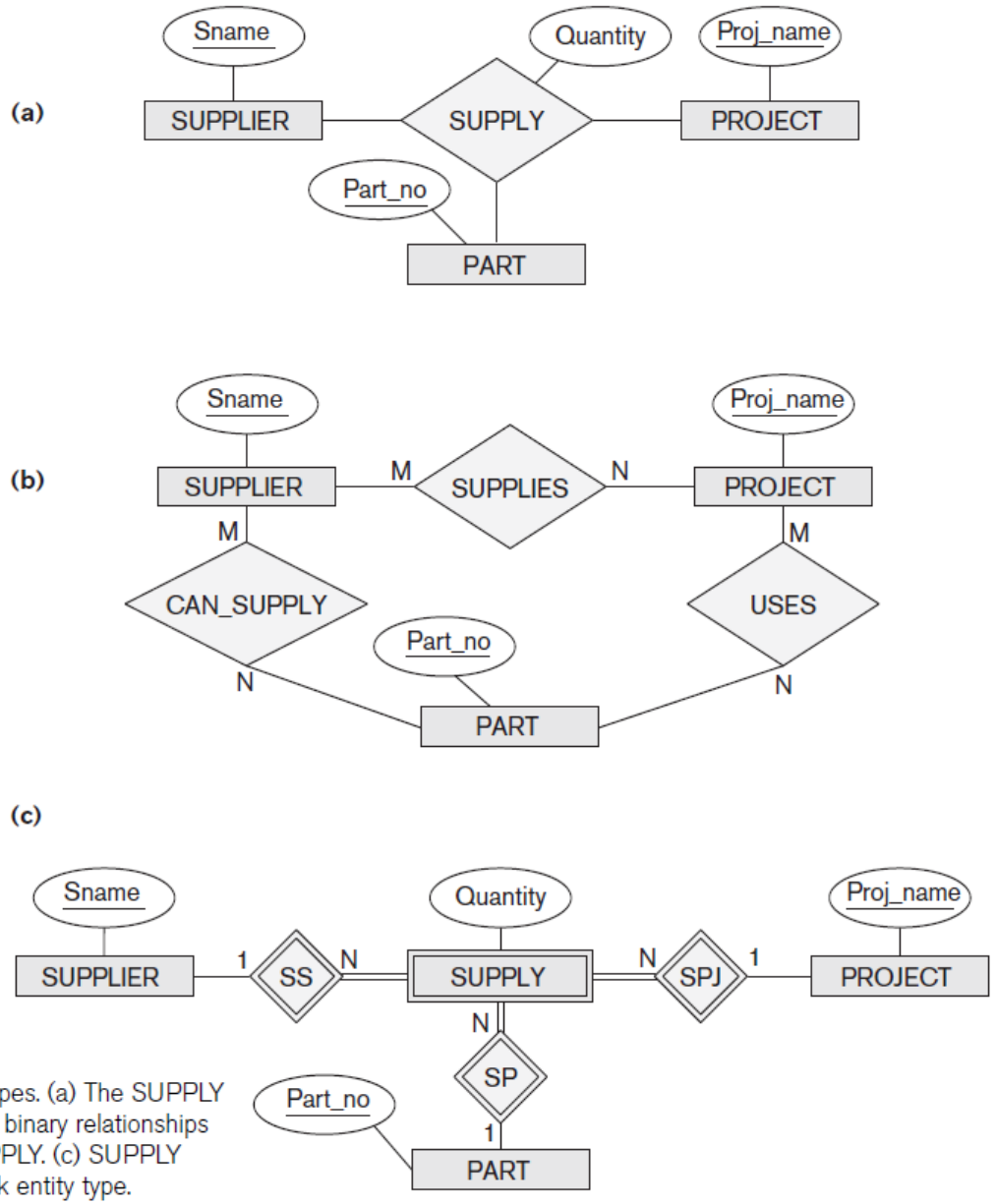


Figure 7.17
Ternary relationship types. (a) The SUPPLY relationship. (b) Three binary relationships not equivalent to SUPPLY. (c) SUPPLY represented as a weak entity type.

Summary

- Basic ER model concepts of entities and their attributes
  - Different types of attributes
  - Structural constraints on relationships
- ER diagrams represent E-R schemas
- UML class diagrams relate to ER modeling concepts