


Chapter 12

Database Design



ดร.สันติฤทธิ์ นรบิน

เรียบเรียง

อ.วไลลักษณ์ วงษ์รัตน์

Content

- 1) วัตถุประสงค์ของ Database Design
- 2) การทำ Normalization
- 3) แบบจำลองข้อมูลเชิงสัมพันธ์ (Relational Database Model)
- 4) การปรับเปลี่ยนโครงสร้างของข้อมูลให้อยู่ในระดับ Logical
- 5) Relational Schema

1 - วัตถุประสงค์ของ Database Design



วัตถุประสงค์ของ Database Design

- ❑ เพื่อให้ได้ข้อมูลความสัมพันธ์ (Relationship) ของข้อมูลที่ต้องมีในระบบ
- ❑ เพื่อให้ได้ข้อมูลความสัมพันธ์ (Relationship) ตามที่ผู้ใช้กลุ่มต่าง ๆ ต้องการ
- ❑ เพื่อให้เข้าใจถึงแนวทางขั้นตอนในการออกแบบฐานข้อมูล
 - ระดับแนวคิด (Conceptual Database Design)
 - ระดับตรรกะ (Logical Database Design)
 - ระดับกายภาพ (Physical Database Design)

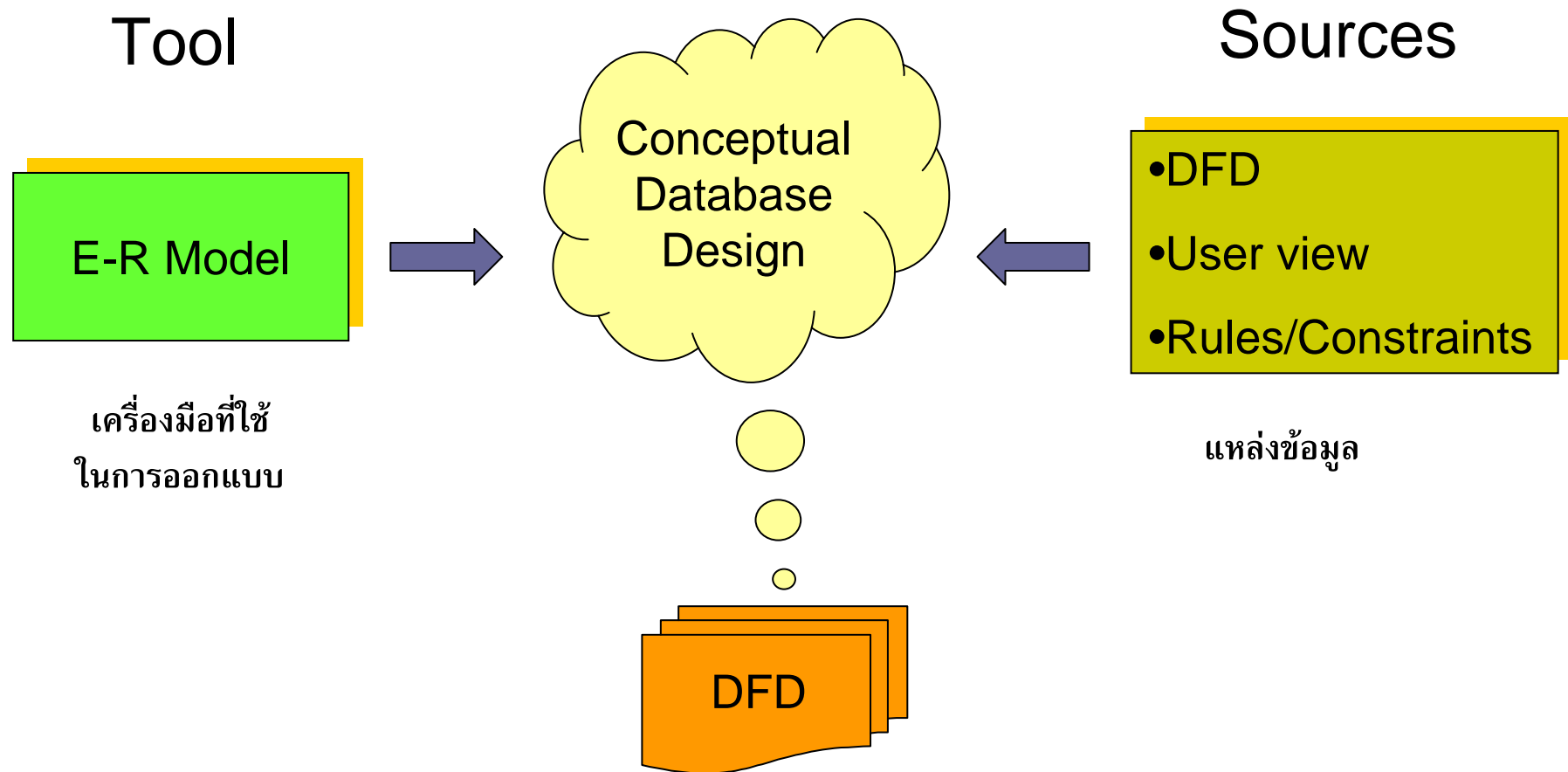
Conceptual Database Design

- ❑ เป็นการออกแบบฐานข้อมูลระดับแนวคิด
- ❑ เป็นการนำข้อมูลที่ได้จากการวิเคราะห์ความต้องการใช้ข้อมูลในระบบ มาออกแบบข้อมูล
- ❑ เพื่อให้ได้เค้าร่างของฐานข้อมูลในระดับแนวคิด ที่ประกอบด้วยโครงสร้างของฐานข้อมูลส่วนหนึ่งที่เป็นความสัมพันธ์กัน

ข้อจำกัดของ Conceptual Database Design

- ❑ ไม่คำนึงถึงปัจจัยทางกายภาพ (Physical)
- ❑ ไม่คำนึงถึงระบบจัดการฐานข้อมูล (DBMS) ที่จะใช้
- ❑ ไม่คำนึงถึงชุดคำสั่งหรือภาษาที่จะนำมาใช้
- ❑ ไม่คำนึงถึงปัจจัยทางด้านฮาร์ดแวร์

Tool and Sources



ลักษณะของข้อมูล

- ❑ แสดงให้เห็นถึงความแตกต่างของข้อมูลความสัมพันธ์ อย่างชัดเจน ถูกต้อง ไม่กำกวม
- ❑ เข้าใจง่าย เพื่อผู้ใช้ที่ไม่มีความรู้ด้านเทคนิคสามารถเข้าใจง่าย
- ❑ มีเฉพาะข้อมูลที่ต้องการให้มีในระบบเท่านั้น
- ❑ ข้อมูลที่ต้องการไม่มีความซ้ำซ้อน
- ❑ การจำลองข้อมูลควรอยู่ในรูปแบบแผนภูมิที่สามารถแปลความหมายได้ง่าย

Logical Database Design

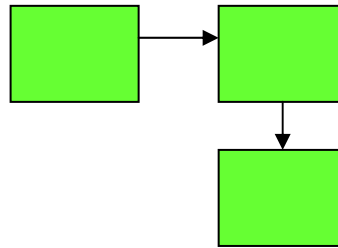
- ❑ เป็นการออกแบบฐานข้อมูลระดับตรรกะ
- ❑ โดยนำข้อมูลที่ได้จากระดับแนวคิดมาวิเคราะห์และออกแบบ
ด้วยแปลงให้อยู่ในรูปแบบการจัดการฐานข้อมูล (DBMS) ที่
เลือกใช้ โดยจะเสริมแนวคิดการทำ Normalization และ
Denormalization

Physical Database Design

- ❑ เป็นการออกแบบฐานข้อมูลระดับกายภาพ
- ❑ โดยนำข้อมูลที่ได้จากระดับตรรกะมากำหนดโครงสร้างข้อมูล และการจัดเก็บ วิธีการเข้าถึงข้อมูล รวมถึงจัดการด้านระบบ ความปลอดภัยของข้อมูลเพื่อให้ฐานข้อมูลทำงานได้อย่างมีประสิทธิภาพ

เค้าร่างของฐานข้อมูลแต่ละระดับ

Conceptual



E-R model



Logical

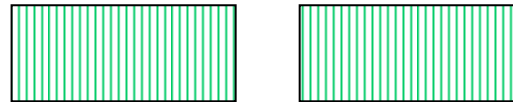
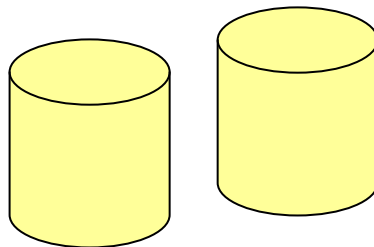


Table definitions

Physical



Database

สรุปขั้นตอนการออกแบบฐานข้อมูล

- ❑ **Conceptual Design** : การอธิบายถึงข้อมูลและความสัมพันธ์กันของข้อมูลในระบบที่วิเคราะห์มา โดยกำหนดว่าจากความต้องการของผู้ใช้ จะต้องมีการมีข้อมูลอะไรบ้าง และข้อมูลแต่ละตัวมีความสัมพันธ์กันอย่างไร และมีข้อจำกัดอะไรบ้าง แสดงโดยใช้ High-level conceptual data model ในที่นี้ใช้ E-R Model

สรุปขั้นตอนการออกแบบฐานข้อมูล (ต่อ)

- ❑ **Logical Design** : จากโครงสร้างของฐานข้อมูลที่ได้จากการออกแบบฐานข้อมูลในระดับ Conceptual แล้วนำโครงสร้างนั้นมาอิงกับ Data Model เพื่อแปลงเป็นตาราง (Relational Data Model) หรือความสัมพันธ์ตามแนวคิดของ Data Model จากนั้นต้องมีการปรับปรุงโครงสร้างของฐานข้อมูลให้มีความซ้ำซ้อนกันน้อยที่สุด โดยใช้ทฤษฎีการทำ Normalization

สรุปขั้นตอนการออกแบบฐานข้อมูล (ต่อ)

- ❑ **Physical Design** : จากนั้นเป็นขั้นตอนการ implement โดยนำ DBMS มาใช้ implement ผลลัพธ์ที่ได้ คือ database schema และขั้นตอนสุดท้าย คือการออกแบบการจัดเก็บข้อมูลลงใน disk เป็นเรื่องของ internal storage structure การเข้าถึงข้อมูล จะทำอย่างไร

ตัวอย่างของฐานข้อมูลเชิงสัมพันธ์

<i>customer-id</i>	<i>customer-name</i>	<i>customer-street</i>	<i>customer-city</i>
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table

<i>customer-id</i>	<i>account-number</i>
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table

<i>account-number</i>	<i>balance</i>
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table

2 - การทำ Normalization



การทำ Normalization

- ❑ Normalization คือ กระบวนการปรับปรุงโครงสร้างข้อมูลของฐานข้อมูลที่มีความซ้ำซ้อนให้อยู่ในรูปแบบที่เป็นบรรทัดฐาน
- ❑ Normal Form มีอยู่ 3 ระดับด้วยกัน คือ
 1. Normal Form ระดับที่ 1 หรือเรียกว่า 1NF (First Normal Form)
 2. Normal Form ระดับที่ 2 หรือเรียกว่า 2NF (Second Normal Form)
 3. Normal Form ระดับที่ 3 หรือเรียกว่า 3NF (Third Normal Form)

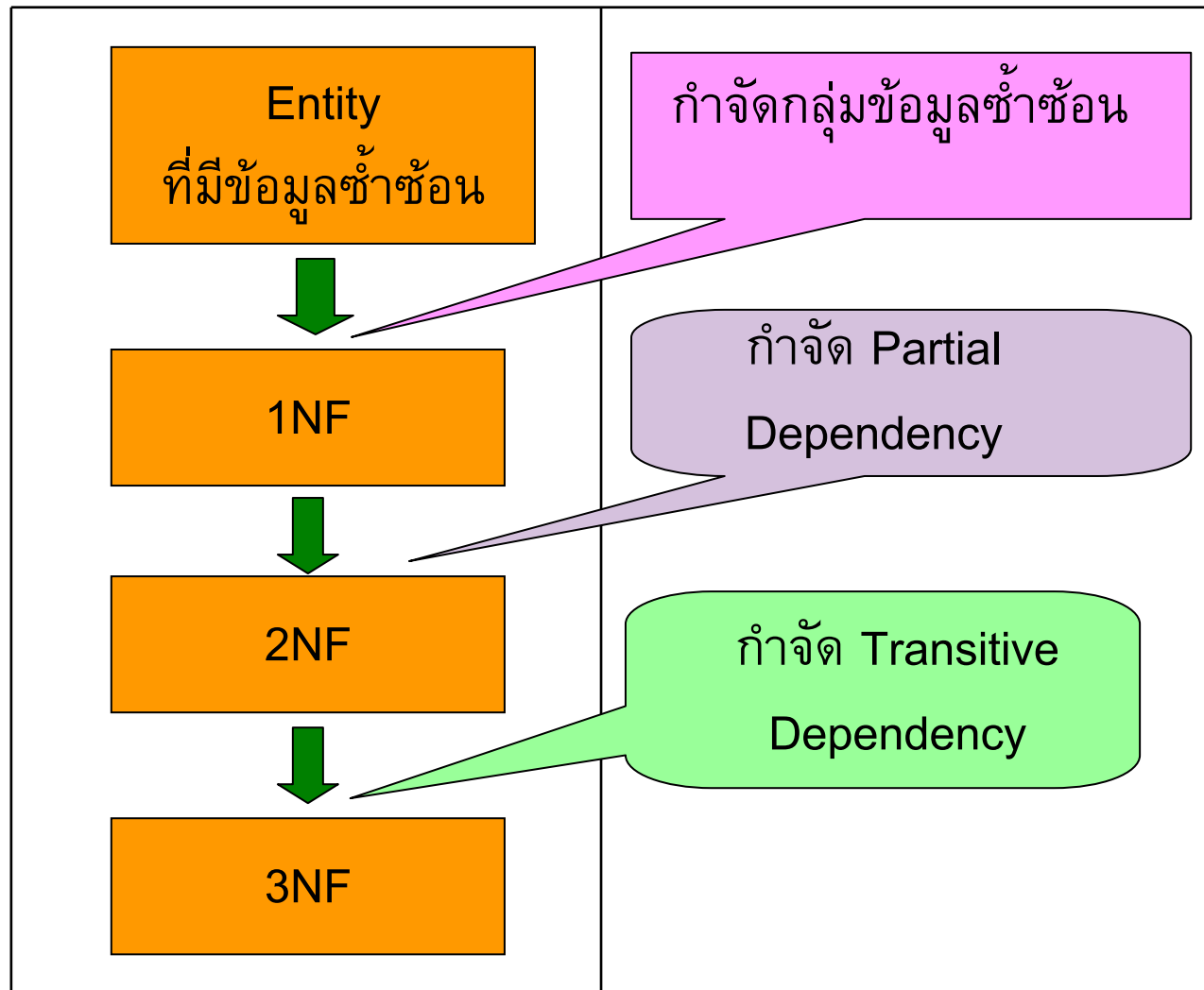
การทำ Normalization (ต่อ)

- นอกจากนี้ยังมีระดับที่ทำให้ Normal Form ระดับที่ 3 มีความแข็งแกร่งขึ้นกว่าเดิม เรียกว่า BCNF (Boyce-Codd Normal Form) ซึ่งพัฒนาขึ้นโดย R.Boyce และ E.F.Codd โดย Normal Form ทุกระดับตั้งอยู่บนพื้นฐานของฟังก์ชันการขึ้นต่อกันระหว่างแอตทริบิวต์ของ Relation (Functional Dependency)

การทำ Normalization (ต่อ)

- Normal Form ในระดับที่สูงขึ้นไปอีกที่อยู่ถัดจาก BCNF ก็ได้ถูกพัฒนาขึ้น คือ Normal Form ระดับที่ 4 (4NF Fourth Normal Form) และระดับที่ 5 (5NF Fifth Normal Form) ซึ่งพัฒนาโดย Fagin (1977, 1979) อย่างไรก็ตามรูปแบบ 4NF และ 5NF ในทางปฏิบัติถือว่าเกิดขึ้นได้ยากมาก

แสดงขั้นตอนของการทำ Normalization



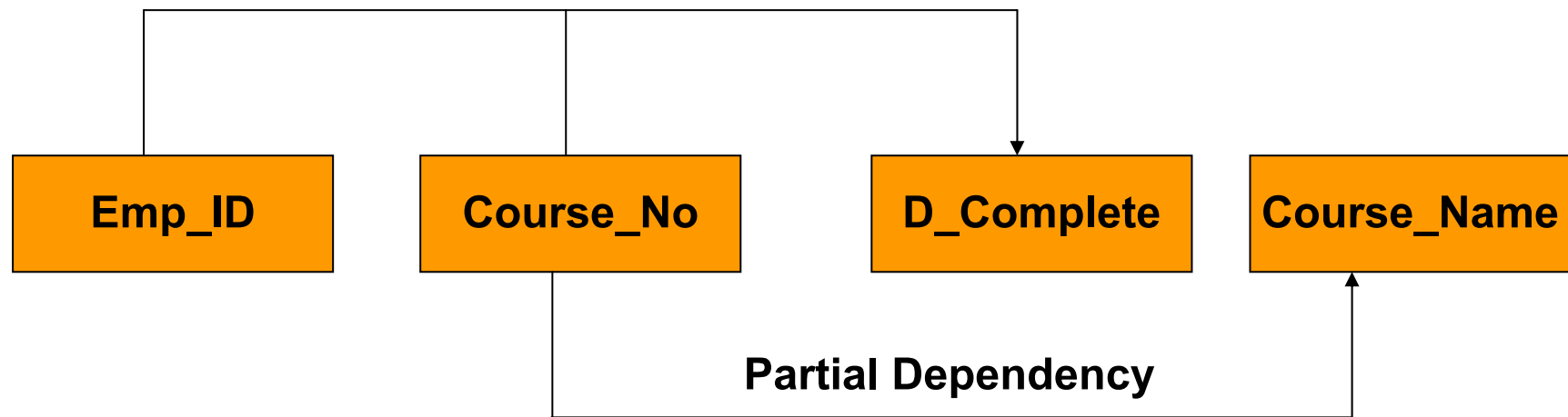
ตัวอย่างตารางข้อมูล

Emp_ID	Emp_Name	Dept	Salary	Course_NO	Course_Name	D-Complete
110	วิลาวลัยย์ ขำคม	Account	15,000	01	Acc PAC	12/0602002
				03	SPSS	30/4/2002
112	อุษาวดี เจริญกุล	Account	15,100	01	Acc PAC	12/0602002
091	นพพร บุญชู	IT	12,000	02	3D Studio max	31/03/2002
				03	SPSS	30/4/2002
010	กสมา ร่มเย็น	IT	11,000	03	SPSS	30/4/2002
001	วนิดา แซ่ลิ้ม	Marketing	12,500	01	Acc PAC	12/06/2002

Normalization

- ❑ Partial Dependency คือความสัมพันธ์ที่ต้องมี Determinant-attribute (Attribute ที่ไม่ใช่ Primary Key) มากกว่า 1 ตัว (มี Primary key/Identifier มากกว่า 1 Attributes นั้นเอง)
- ❑ ความสัมพันธ์ระหว่างค่าของ Attribute เมื่อ Attribute ที่เป็น Determinant บางตัวสามารถระบุค่าของ Attribute อื่นที่ไม่ใช่ Identifier ได้

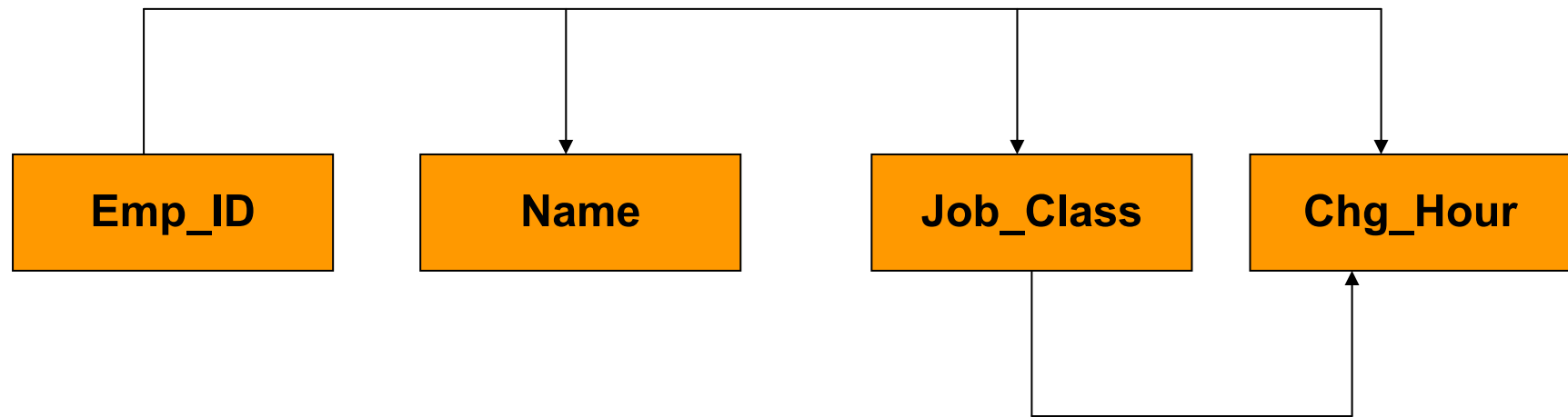
Normalization



Normalization

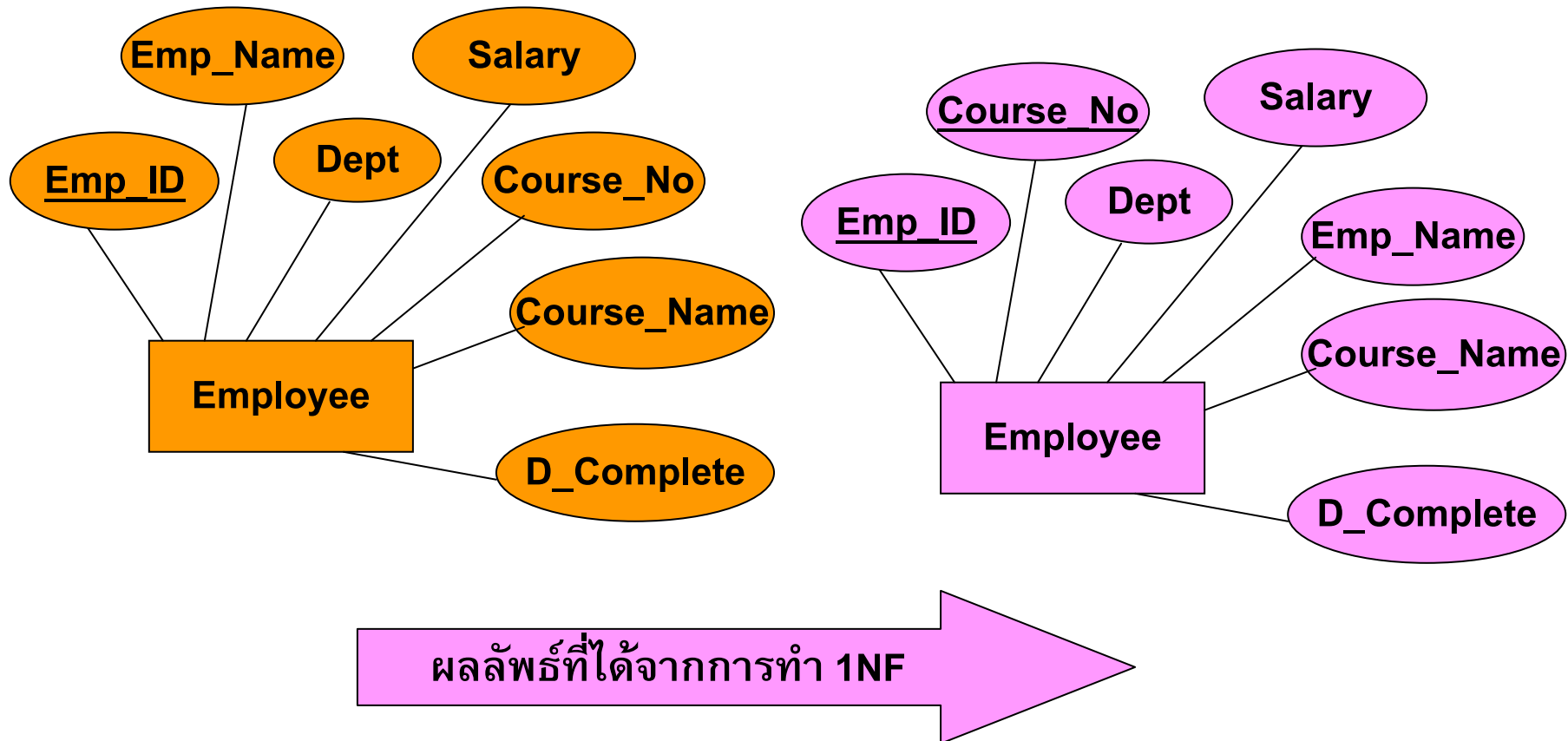
- ❑ Transitive Dependency คือความสัมพันธ์ระหว่าง Attribute ที่ไม่ใช่ Determinant Attribute ไปขึ้นอยู่กับ Attribute อื่นที่ไม่ได้เป็น Determinant Attribute เหมือนกัน

Normalization



Transitive Dependency

First Normal Form (1NF)



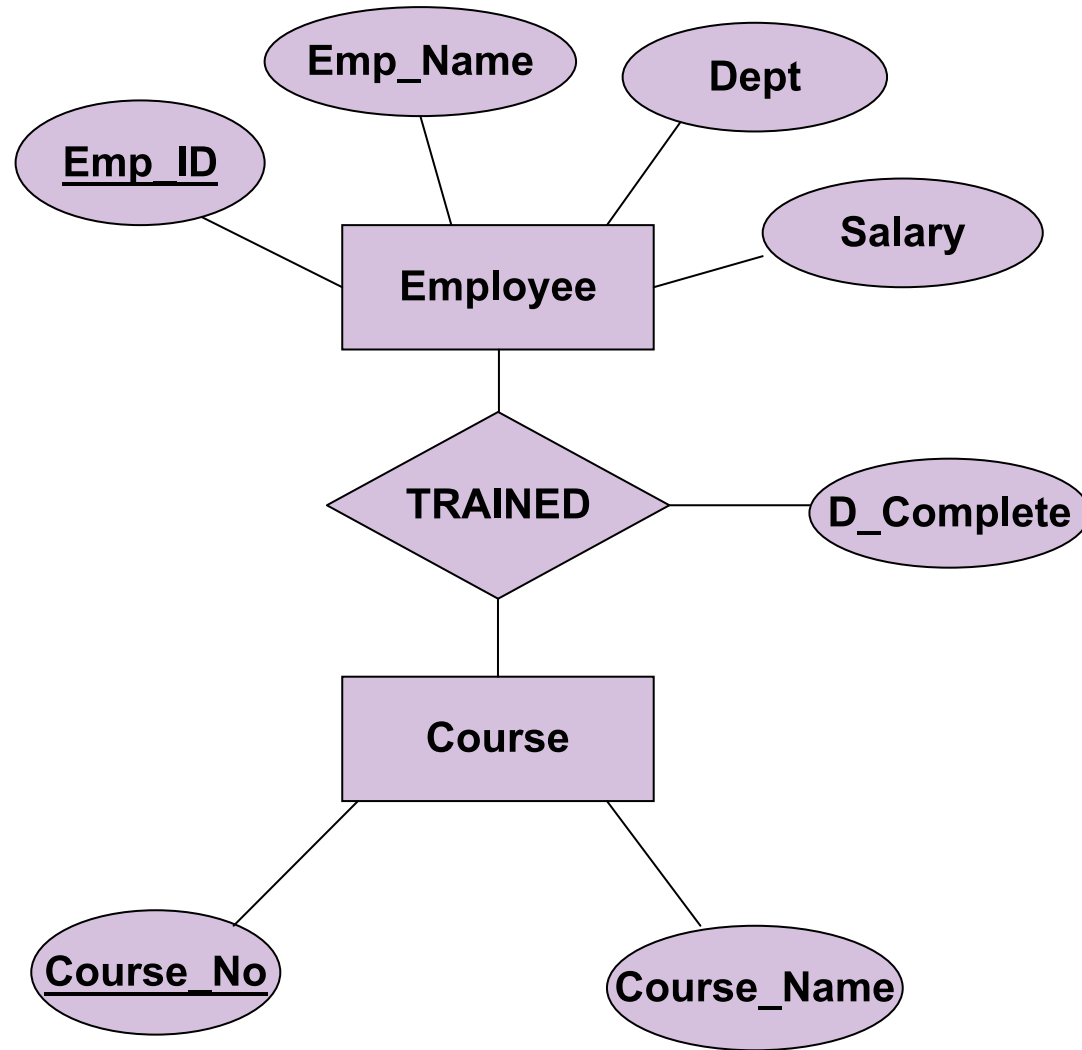
ผลลัพธ์ที่ได้จากการทำ 1NF

Emp_ID	Course_NO	Emp_Name	Dept	Salary	Course_Name	D-Complete
110	01	วิลาวัลย์ ขำคม	Account	15,000	Acc PAC	12/0602002
110	03	วิลาวัลย์ ขำคม	Account	15,000	SPSS	30/4/2002
112	01	อุษาวดี เจริญกุล	Account	15,100	Acc PAC	12/0602002
091	02	นพพร บุญชู	IT	12,000	3D Studio max	31/03/2002
091	03	นพพร บุญชู	IT	12,000	SPSS	30/4/2002
010	03	กสมา ร่มเย็น	IT	11,000	SPSS	30/4/2002
001	01	วนิดา แซ่ลี้ม	Marketing	12,500	Acc PAC	12/06/2002

Second Normal Form (2NF)

- Entity หรือ Relationship จะมีคุณสมบัติเป็น 2 NF ได้เมื่อ
 - 1) Entity นั้นจะต้องมีคุณสมบัติ 1NF
 - 2) Attributes จะต้องไม่มีความสัมพันธ์กันแบบ Partial Dependency กล่าวคือ Nonprime Attribute จะต้องไม่ขึ้นอยู่กับ Identifier ตัวใดตัวหนึ่ง กรณีที่ Identifier นั้นเกิดจาก Attribute มากกว่า 1 Attribute

Second Normal Form (2NF)



Second Normal Form (2NF)

Employee

Emp_ID	Emp_Name	Dept	Salary
001	วนิดา แซ่ลี้	Marketing	12,500
010	กสมา ร่มเย็น	IT	11,000
091	นพพร บุญชู	IT	12,000
110	วิลาวัลย์ ขำคม	Account	15,000
112	อุษาวดี เจริญกุล	Account	15,100

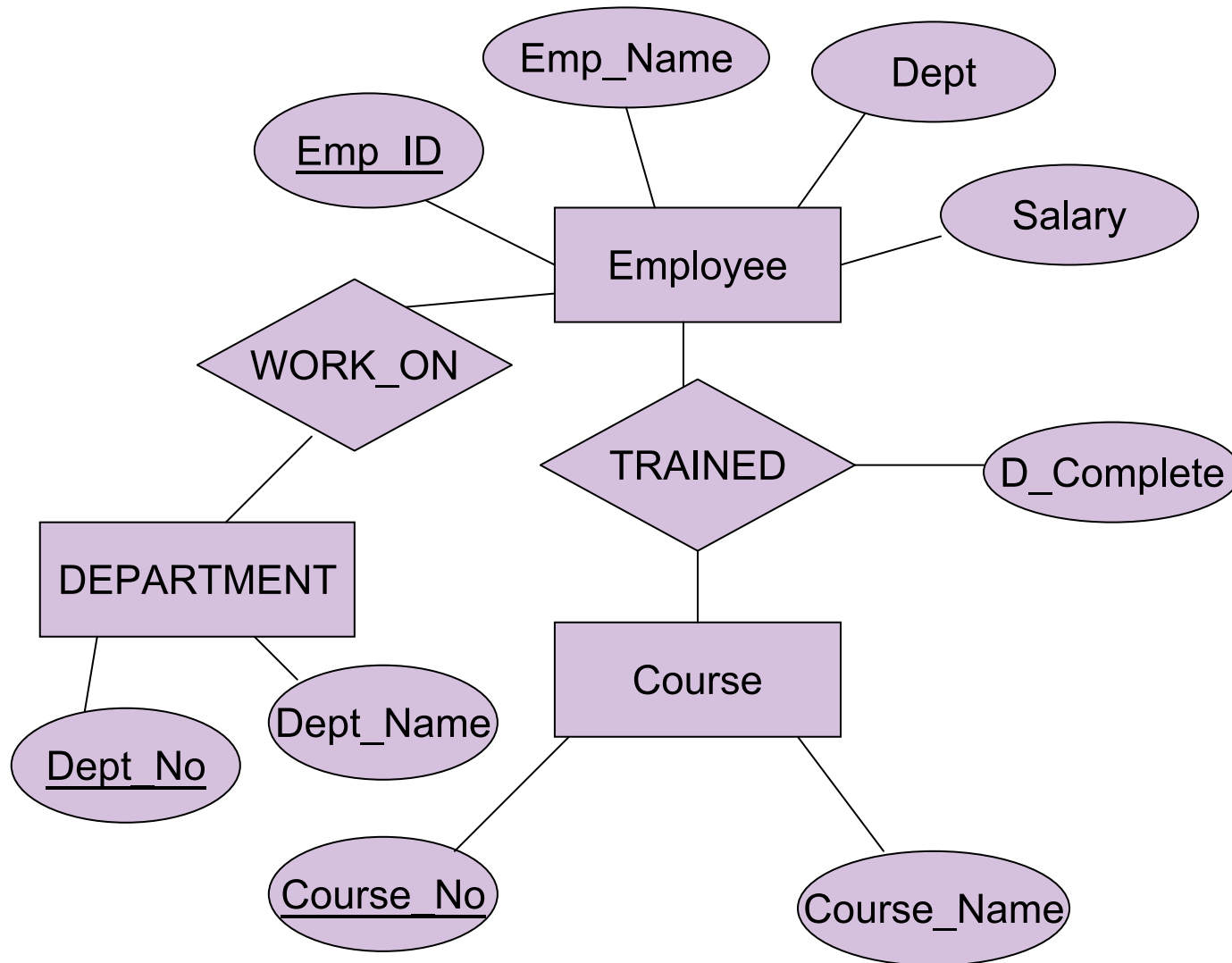
Course

Course_No	Course_Name
01	Acc PAC
02	3D Studio max
03	SPSS

Third Normal Form (3NF)

- Entity หรือ Relationship จะมีคุณสมบัติเป็น 3NF ได้เมื่อ
 - 1) Entity นั้นจะต้องมีคุณสมบัติ 2NF
 - 2) ต้องเป็น Entity ที่ Attributes ไม่มีความสัมพันธ์แบบ Transitive Dependency กล่าวคือเกิดกรณีที่ Nonprime Attribute (หรือ Attribute ที่ไม่ใช่ Identifier นั้นเอง) ไปขึ้นอยู่กับ Nonprime attribute ด้วยกันเอง

Third Normal Form (3NF)



3 – Relational Database Model



แบบจำลองข้อมูลเชิงสัมพันธ์

- ❑ แบบจำลองข้อมูลเชิงสัมพันธ์ (Relational Database Model)
คือ แบบจำลองที่นำเสนอโครงสร้างของข้อมูลที่อยู่ในฐานข้อมูล
เชิงสัมพันธ์ โดยเสนอในรูปแบบตาราง (Table) หรือ
Relations
- ❑ Relations คือ ตาราง 2 มิติที่ใช้บรรจุข้อมูล โดยแต่ละ
Relations จะประกอบด้วยชุดของแถว (Row) เรียกว่า
“Tuple” และ คอลัมน์ (Column)

แสดงตัวอย่าง Relation ในแบบจำลองข้อมูลเชิงสัมพันธ์

Emp_ID	Emp_Name	Emp_Sex	Emp_Dep	Salary
110	วิลาวัลย์ ชำคม	F	โปรแกรมเมอร์	15,000
112	อุษาวดี เจริญกุล	F	โปรแกรมเมอร์	15,100
091	นพพร บุญชู	M	การตลาด	12,000
010	กสมา ร่มเย็น	M	การตลาด	11,000
001	วนิดา แซ่ลี้	F	การตลาด	12,500

EMPLOYEE (Emp_ID, Emp_Sex, Emp_Name, Emp_Dep, Salary)

คุณสมบัติของ Relation

1. ข้อมูลที่อยู่ใน Attribute 1 Attribute และ Row 1 Row จะต้องมีความสัมพันธ์ Atomic คือมีค่าเพียงค่าเดียวเท่านั้น หรือเรียกว่า “Single-Valued”
2. ข้อมูลที่อยู่ใน Attribute เดียวกันจะต้องอยู่ภายในขอบเขตของค่าที่เป็นไปได้ (Domain) เดียวกัน
3. ข้อมูลในแต่ละแถว (Tuple) จะต้องไม่ซ้ำกันเลย เช่น กรณีที่ในแผนงานหนึ่ง มีพนักงานชื่อและนามสกุลเดียวกัน 2 คน ในการเก็บข้อมูลของพนักงานทั้งสองจะต้องไม่มีค่าซ้ำกัน ดังตัวอย่าง

คุณสมบัติของ Relation

	Emp_ID	Emp_Name	Emp_Sex	Emp_Dep	Salary
	110	วิลาวัลย์ ขำคม	F	โปรแกรมเมอร์	15,000
	111	วิลาวัลย์ ขำคม	F	โปรแกรมเมอร์	15,000
	112	อุษาวดี เจริญกุล	F	โปรแกรมเมอร์	15,100
	091	นพพร บุญชู	M	การตลาด	12,000
	010	กสมา ร่มเย็น	M	การตลาด	11,000
	001	วนิดา แซ่ลี้	F	การตลาด	12,500

คุณสมบัติของ Relation (ต่อ)

4. ลำดับการจัดเรียงของ Attributes จากซ้ายไปขวา ไม่จำเป็นต้องมีการเรียงลำดับ
5. ลำดับของข้อมูลแต่ละแถวจากบนลงล่าง ไม่จำเป็นต้องมีการเรียงลำดับ

คุณสมบัติของ Relation

Emp_ID	Emp_Name	Emp_Sex	Emp_Dep	Salary
110	วิลาวัลย์ ขำคม	F	โปรแกรมเมอร์	15,000
112	อุษาวดี เจริญกุล	F	โปรแกรมเมอร์	15,100
M	นพพร บุญชู	002	การตลาด ฝ่ายขาย	18,000
010	กสมา ร่มเย็น	M	การตลาด	11,000
001	วนิดา แซ่ลี้	F	การตลาด	12,500

2

2

1

คุณสมบัติของ Relation

	Emp_ID	Emp_Name	Emp_Sex	Emp_Dep	Salary
3,5	110	วิลาวัลย์ ขำคม	F	โปรแกรมเมอร์	15,000
	111	วิลาวัลย์ ขำคม	F	โปรแกรมเมอร์	15,000
	112	อุษาวดี เจริญกุล	F	โปรแกรมเมอร์	15,100
	091	นพพร บุญชู	M	การตลาด	12,000
	010	กสมา ร่มเย็น	M	การตลาด	11,000
	001	วนิดา แซ่ลี้	F	การตลาด	12,500

4 - การปรับเปลี่ยนโครงสร้างของข้อมูล ให้อยู่ในระดับ Logical



การปรับเปลี่ยนโครงสร้างของข้อมูลให้อยู่ในระดับ Logical

□ ในการปรับเปลี่ยนโครงสร้างของข้อมูลให้อยู่ในระดับ Logical มีดังนี้

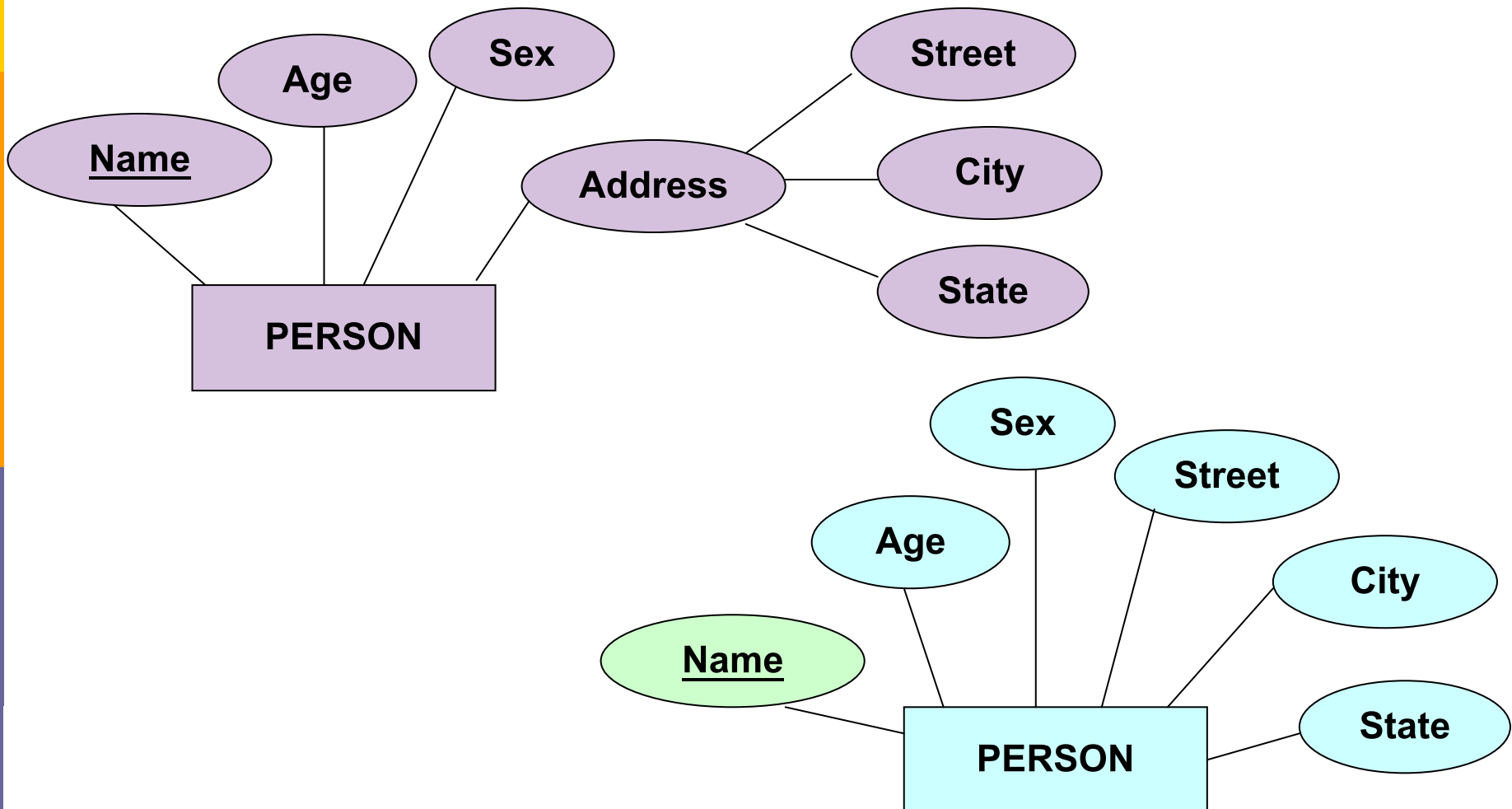
4.1 การกำจัด Composite Attribute

4.2 การกำจัด Multivalued Attribute

4.3 การกำจัด External Identifier

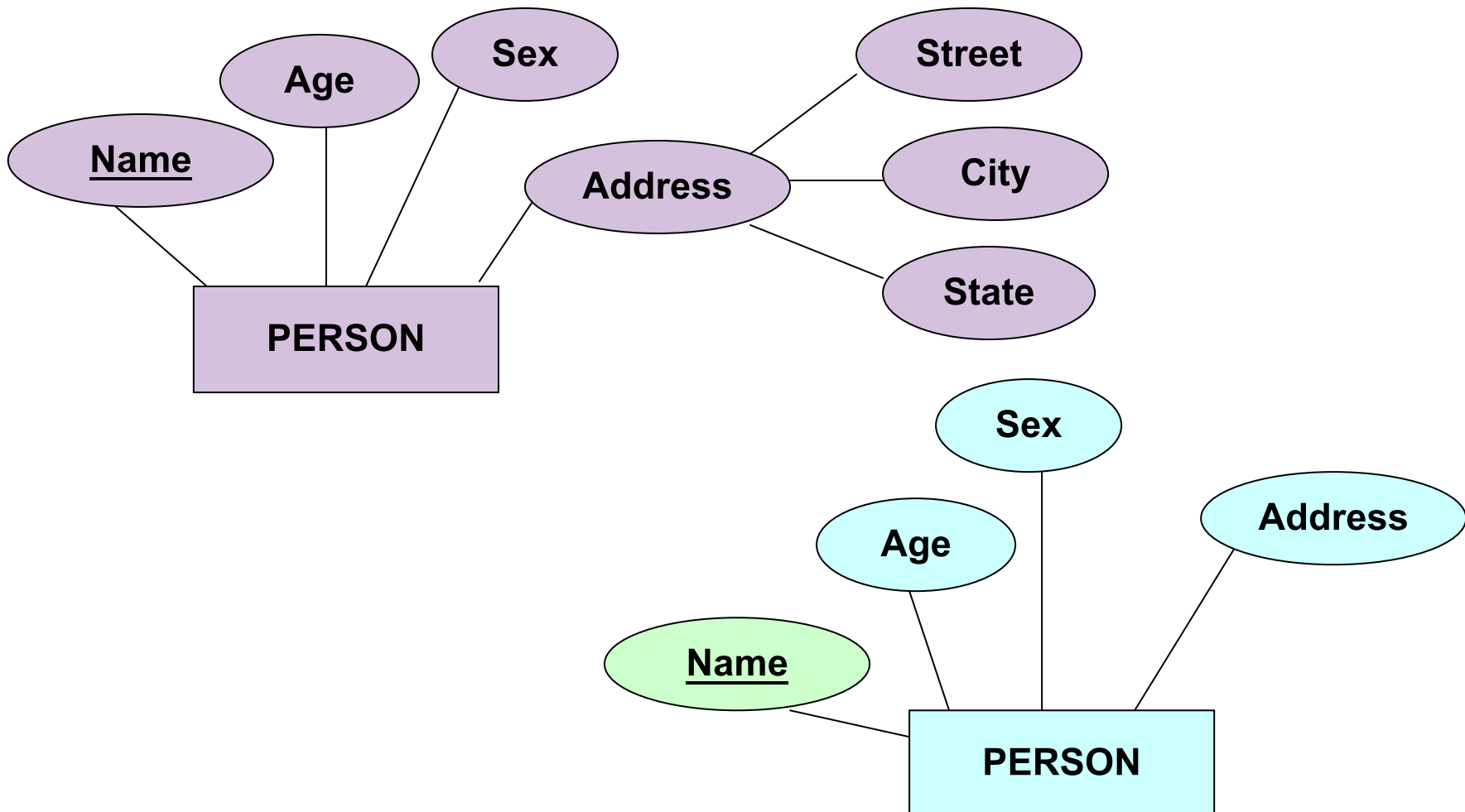
4.1 การกำจัด Composite Attribute

- วิธีที่ 1 แยก Composite Attribute ออกเป็น Attribute ย่อย



การกำจัด Composite Attribute (ต่อ)

- ❑ วิธีที่ 2 ยุบ Composite Attribute ให้เหลือเพียง Attribute เดียว



4.2 การกำจัด Multivalued Attribute

- ❑ กรณีกำจัด Multivalued Attribute ของ Entity จะมีขั้นตอนดังนี้
 - 1) แปลง Multivalued Attribute ของ Entity ไปเป็น Entity ใหม่ โดยที่ Entity ใหม่นี้จะประกอบไปด้วย Attribute ที่ได้มาจาก Multivalued Attribute และ Identifier ของ Entity เดิม
 - 2) รวมทุก Attribute ของ Entity ใหม่เป็น Identifier ของ Entity ใหม่

การกำจัด Multivalued Attribute (ต่อ)

กรณีกำจัด Multivalued Attribute ของ Relationship จะมีขั้นตอนดังนี้

1. แปลง Multivalued Attribute ของ Relationship ไปเป็น Entity ใหม่ โดยที่ Entity ใหม่นี้จะประกอบด้วย Attribute ที่ได้มาจาก Multivalued Attribute และ Identifier ของ Entity ที่มีความสัมพันธ์กับ Relationship นั้น สำหรับการเลือก Identifier มาเป็น Attribute ของ Entity ใหม่จะมีหลักการดังนี้

การกำจัด Multivalued Attribute (ต่อ)

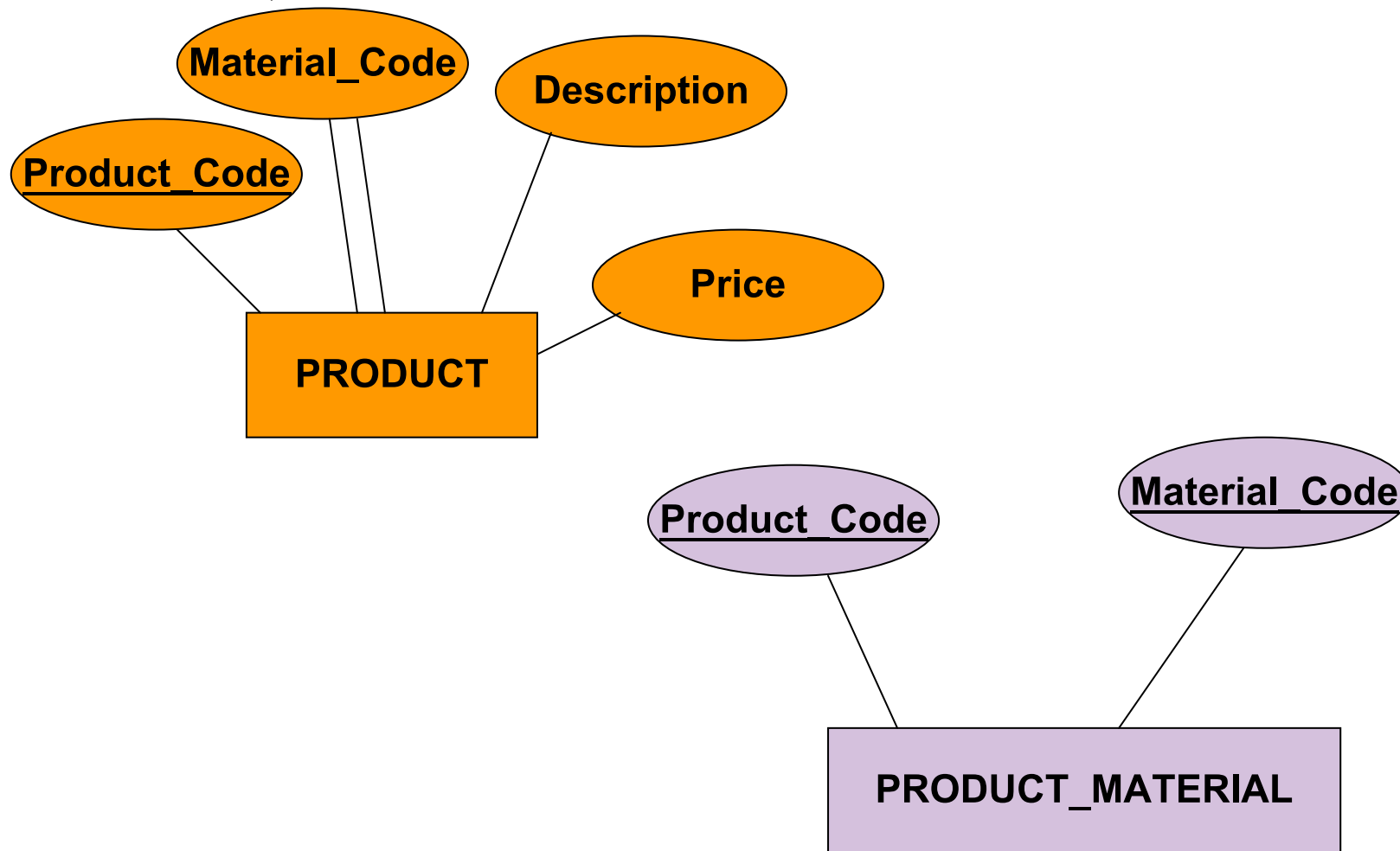
- 1.1 ถ้าเป็น One-to-One Relationship ให้เลือก Identifier จาก Entity ใดก็ได้ที่มีความสัมพันธ์กับ Relationship นั้นมาเป็น Attribute ของ Entity ใหม่
- 1.2 ถ้าเป็น One-to-Many Relationship ให้เลือก Identifier จาก Entity ทางด้าน Many ที่มีความสัมพันธ์กับ Relationship นั้นมาเป็น Attribute ของ Entity ใหม่
- 1.3 ถ้าเป็น Many-to-Many Relationship ให้เลือก Identifier จากทั้ง 2 Entity ที่มีความสัมพันธ์กับ Relationship นั้นมาเป็น Attribute ของ Entity ใหม่

การกำจัด Multivalued Attribute (ต่อ)

2. รวมทุก Attribute ของ Entity ใหม่เป็น Identifier ของ Entity ใหม่

การจัด Multivalued Attribute

เป็น Multivalued attribute

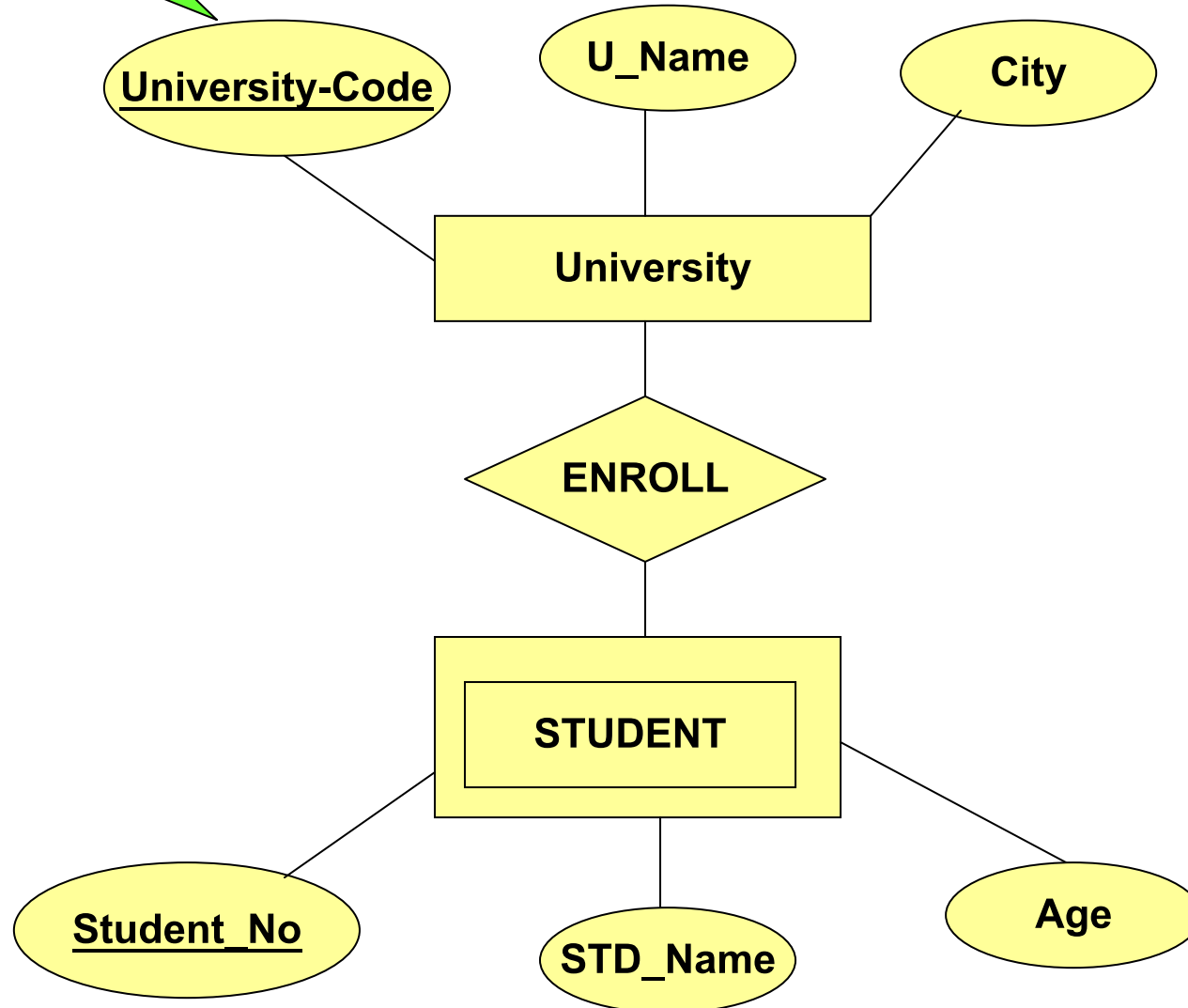


4.3 การกำจัด External Identifier

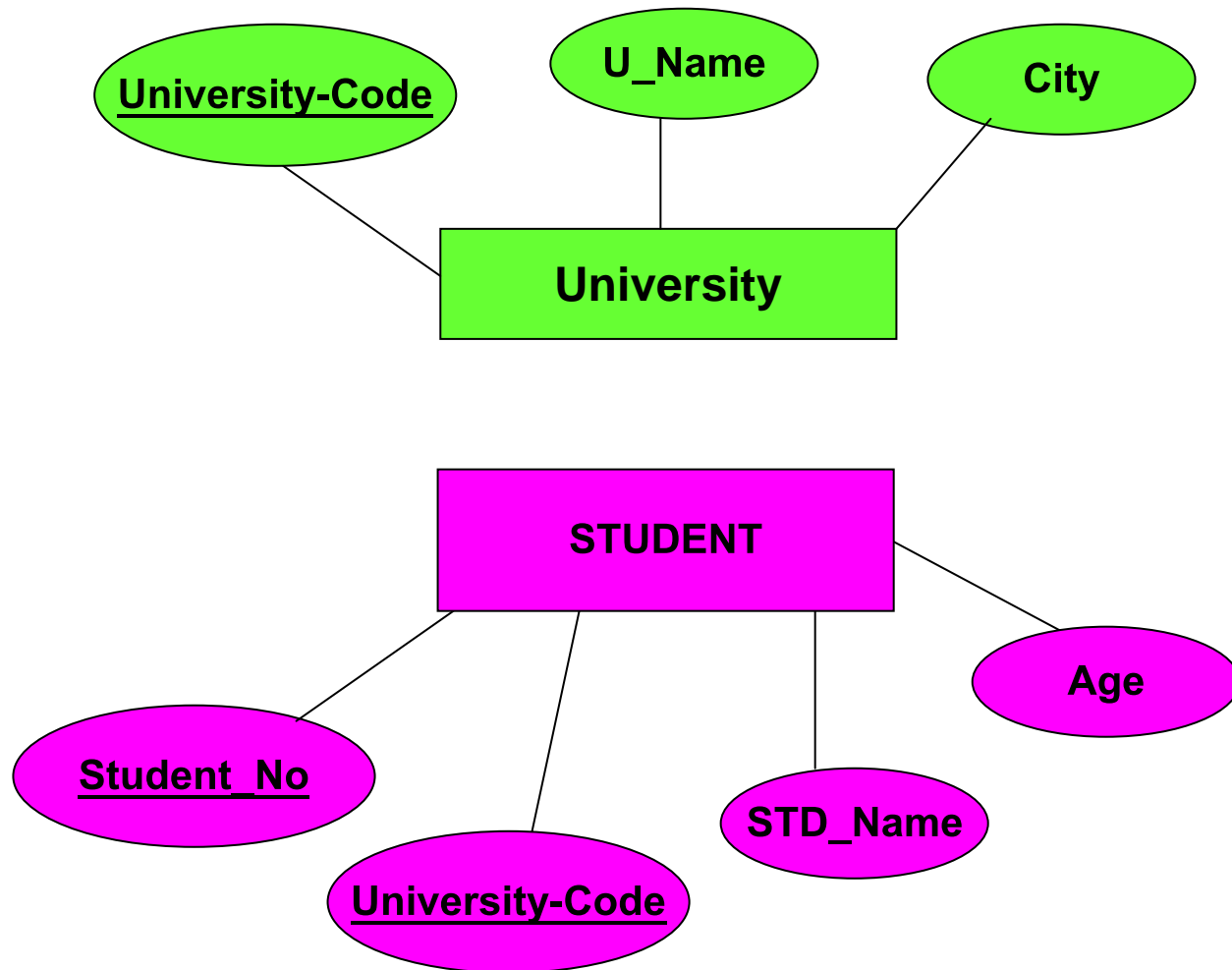
- ❑ เนื่องจาก Relational Model จะไม่ปรากฏ Identifier แบบ External Identifier ดังนั้นจึงต้องแปลง External Identifier ให้อยู่ในรูป Internal Identifier โดยมีขั้นตอนดังนี้
 - 1) แปลง External Identifier ไปเป็น Attribute ใหม่ของ Weak โดย Attribute ใหม่นี้จะทำหน้าที่เป็น Identifier ร่วมกับ Identifier เดิมของ Weak Entity
 - 2) ตัด Relationship ระหว่าง Strong Entity และ Weak Entity ที่

เป็น External Identifier
ของ Entity Student

การจัด External Identifier



การจัด External Identifier



5 – Relational Schema



Relational Schema

- 5.1 ความหมายของ Relational Schema
- 5.2 การแปลง Relationship ไปเป็น Relation กรณีที่เป็น One-to-One Relationship
- 5.3 การแปลง Relationship ไปเป็น Relation กรณีที่เป็น One-to-Many Relationship
- 5.4 การแปลง Relationship ไปเป็น Relation กรณีที่เป็น Many-to-Many Relationship
- 5.5 การแปลง Relationship ไปเป็น Relation กรณีที่เป็น N-ary Relationship
- 5.6 การแปลง Relationship ไปเป็น Relation กรณีที่เป็น Recursive Relationship

5.1 Relational Schema

- ❑ Relational Schema เป็นรูปแบบที่ใช้แสดงถึงโครงสร้างของแต่ละ Entity ใน Relational Model ซึ่งประกอบขึ้นจากเซตของ Attribute ภายใต้ Relation นั้นโดยมีรูปแบบ ดังนี้

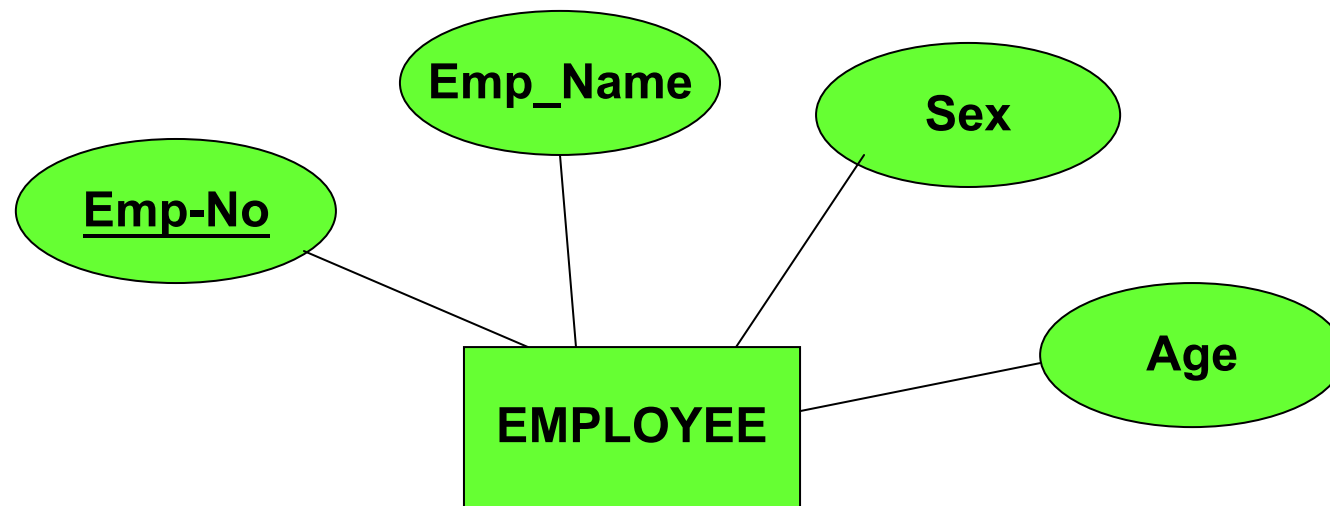
Relation-Name (A1, A2, A3, ..., An)

โดยที่

- Relation-Name หมายถึง ชื่อของ Relation
- A1,A2,...,An หมายถึง รายชื่อ Attribute ภายใต้ Relation นั้น
- เมื่อนำแต่ละ Relation มาประกอบกันจะปรากฏเป็นโครงสร้างทั้งหมดของฐานข้อมูล เรียกว่า Relational Database Schema

การแปลง Entity ไปเป็น Relation

- ในการแปลง Relationship ไปเป็น Relation นั้นจะแปลงตาม Cardinality ของ Relationship



EMPLOYEE (Emp-No, Emp-Name, Sex, Age)

5.2 กรณีที่เป็น One-to-One Relationship

5.2.1 กรณีที่ 1

Entity มีความสัมพันธ์กับแบบ Total Participation

5.2.2 กรณีที่ 2

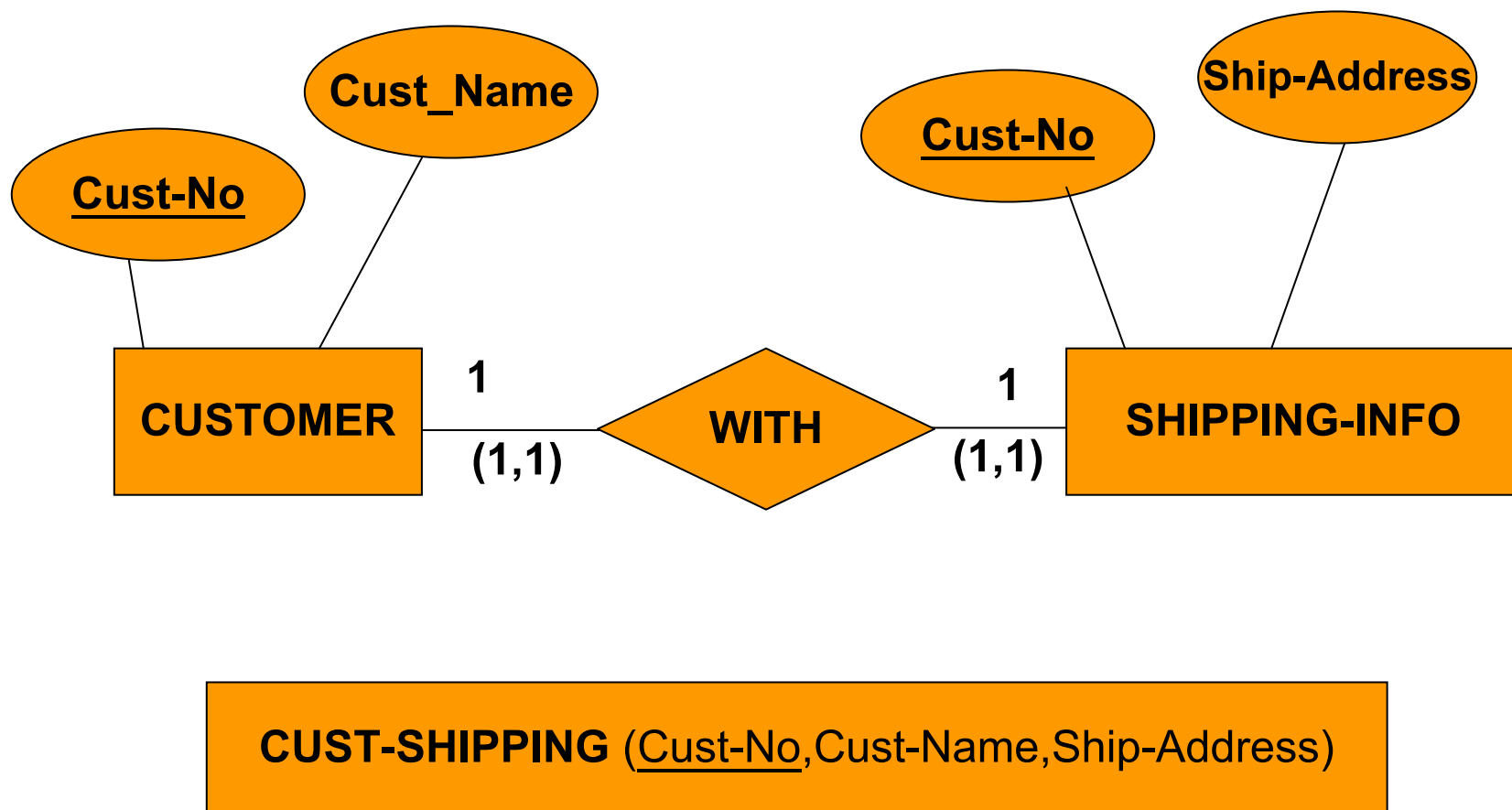
Entity มีความสัมพันธ์กับแบบ Partial Participation

5.2.1 กรณีที่ 1

□ Entity มีความสัมพันธ์กับแบบ Total Participation

คือ สมาชิกทุกตัวใน Entity หนึ่งมีความสัมพันธ์กับอีก Entity หนึ่ง อย่างน้อย 1 ตัว โดยในกรณีนี้ ให้ทำการแปลงทั้ง Entity และ Relationship ที่สัมพันธ์กันเป็น Relation เดียวที่ประกอบขึ้นจาก Attribute ของทั้ง Entity และ Relationship ที่สัมพันธ์กันนั้น พร้อมกับกำหนด Primary key จาก Identifier ของ Entity ใด Entity หนึ่งที่สัมพันธ์กัน

การแปลงโครงร่างของ Relationship ให้อยู่ในรูปของ Relational Schema แบบ Total Participant



5.2.2 กรณีที่ 2

□ Entity มีความสัมพันธ์กับแบบ Partial Participation

คือ สมาชิกบางตัวใน Entity หนึ่งไปมีความสัมพันธ์กับสมาชิกบางตัวของอีก Entity หนึ่ง โดยในกรณีนี้ ให้ทำการแปลงแต่ละ Entity ที่มีสัมพันธ์กันเป็นแต่ละ Relation ที่มี Primary key ซึ่งเลือกจาก Identifier ของ Entity เหล่านั้น

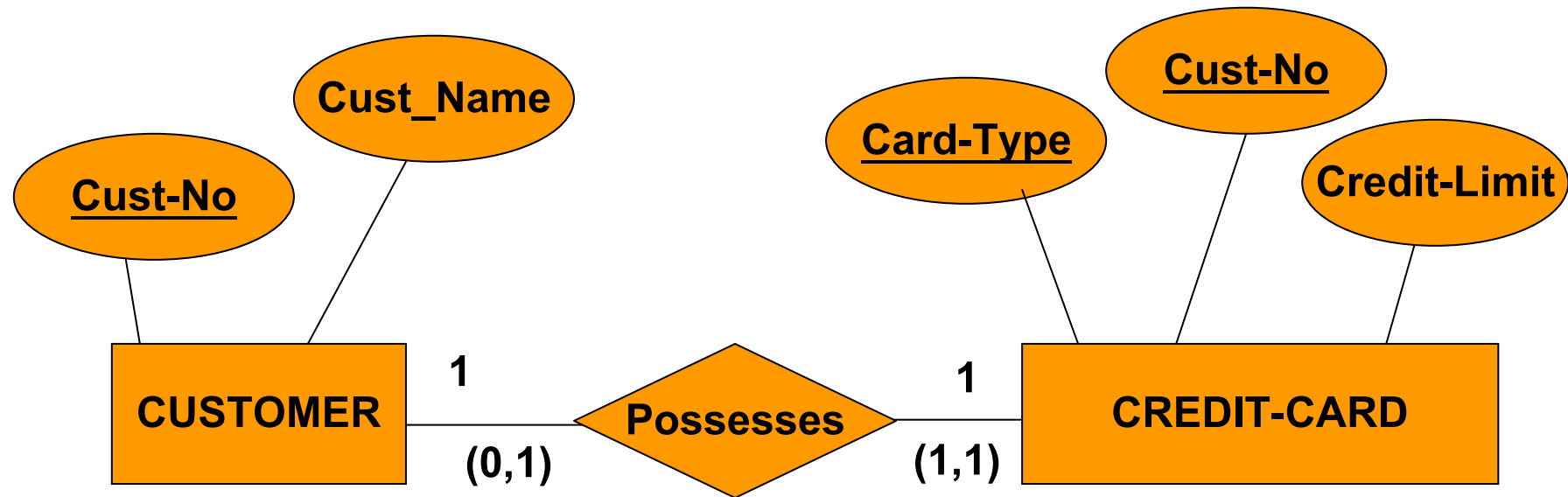
5.2.2 กรณีที่ 2 (ต่อ)

□ ส่วน Relationship สามารถทำได้ 2 วิธีดังนี้

- วิธีที่ 1

แปลง Relationship ไปเป็น Relation ที่ประกอบด้วย Attribute ของ Relationship นั้นเองรวมกับ Attribute ที่เป็น Identifier ของ Entity ที่สัมพันธ์กับ Relationship นั้น ส่วน Primary Key ให้เลือกจาก Identifier ของ Entity ใด Entity หนึ่งที่สัมพันธ์กับ Relationship นั้น

การแปลงโครงร่างของ Relationship ให้อยู่ในรูปของ Relational Schema แบบ Partial Participant



CUSTOMER (Cust-No, Cust-Name)

CREDIT-CARD (Card_Type, Card-No, Credit-Limit)

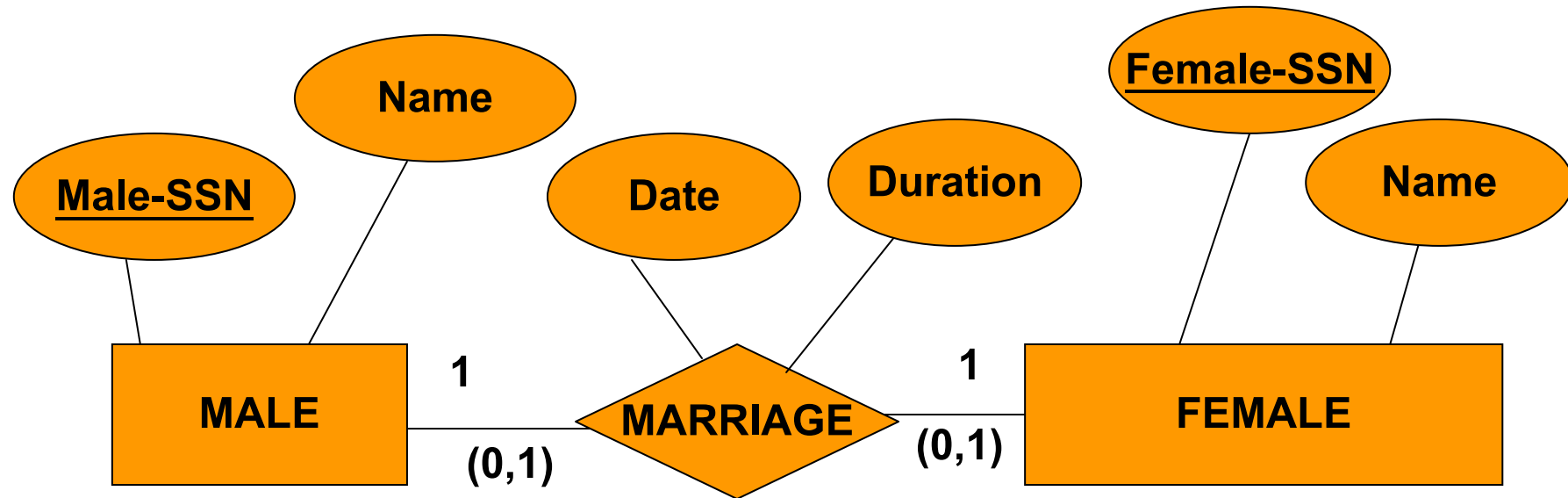
POSSESSES-CARD (Card-Type, Card-No, Cust-No)

5.2.2 กรณีที่ 2 (ต่อ)

- วิธีที่ 2

วิธีนี้สามารถทำได้เช่นเดียวกับวิธีที่ 1 แต่การเลือก Primary Key ให้เลือกจาก Identifier ของทุก Entity ที่สัมพันธ์กับ Relationship นั้น ดังตัวอย่าง

การแปลงโครงร่างของ Relationship ให้อยู่ในรูปของ Relational Schema แบบ Partial Participant



MALE (Male-SSN, Name)

FEMALE (Female-SSN, Name)

MARRIAGE (Male-SSN, Female-SSN, Date, Duration)

5.3 กรณีที่เป็น One-to-Many Relationship

5.3.1 กรณีที่ 1

Entity มีความสัมพันธ์กับแบบ Total Participation

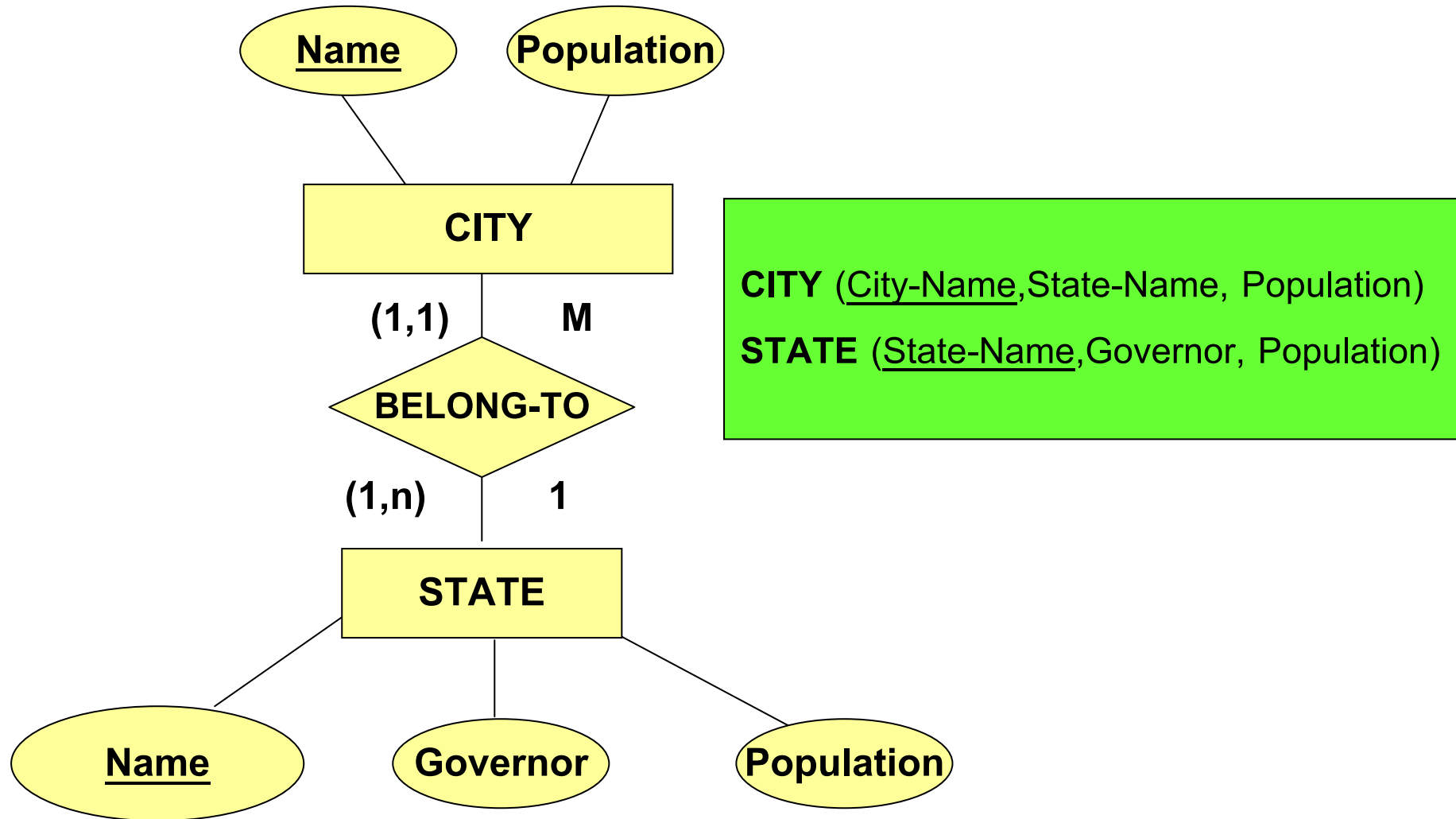
5.3.2 กรณีที่ 2

Entity มีความสัมพันธ์กับแบบ Partial Participation

5.3.1 กรณีที่ 1

- Entity มีความสัมพันธ์กับแบบ Total Participation คือ ให้ย้ายทุก Attribute ของ Relationship รวมทั้งสำเนา Identifier ของ Entity ทางด้าน One ไปเป็น Attribute ของ Entity ทางด้าน Many จากนั้นจึงแปลงทั้ง 2 Entity ไปเป็น Relation ดังตัวอย่าง

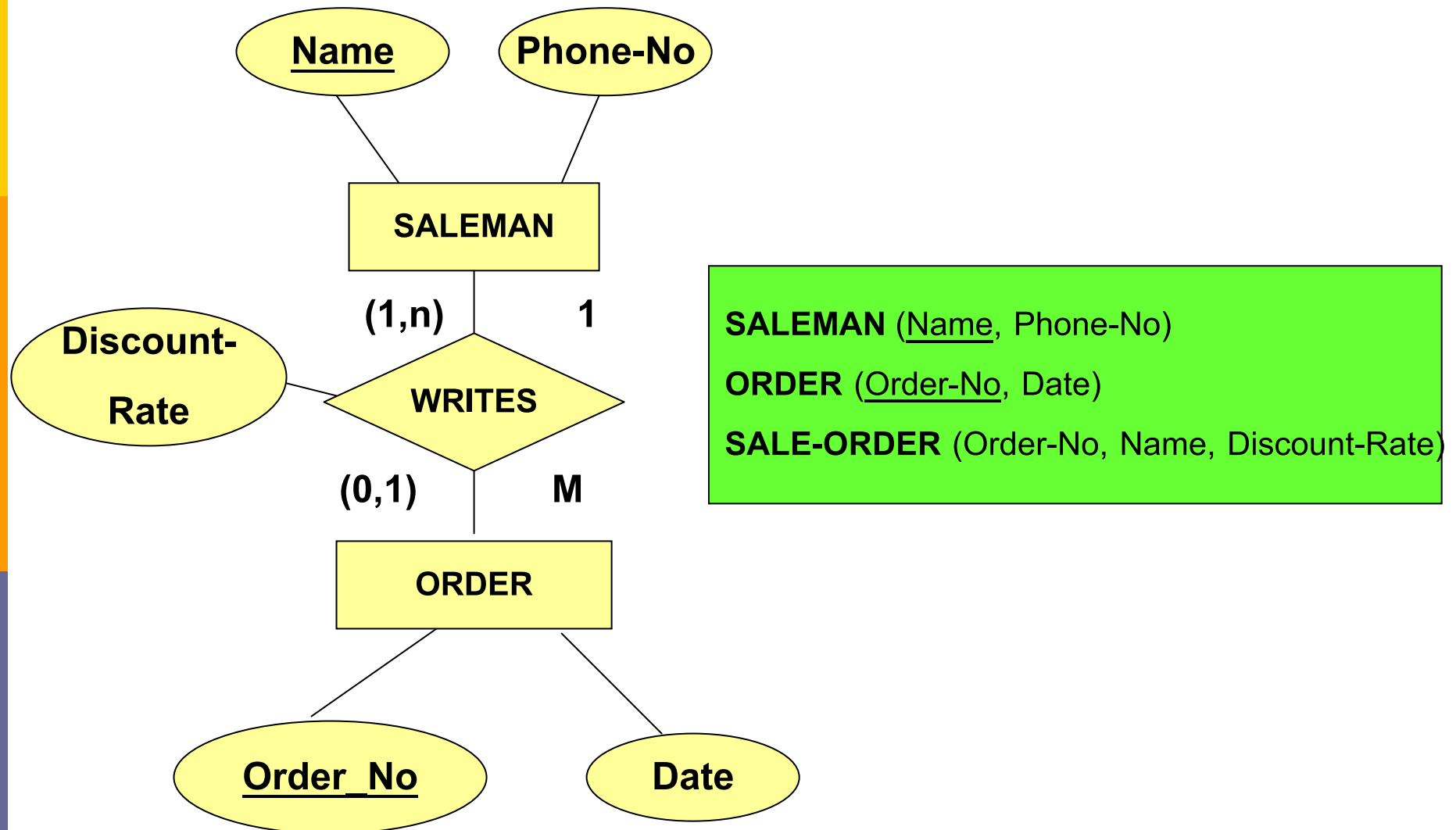
การแปลงโครงร่างของ Relationship ให้อยู่ในรูปของ Relational Schema



5.3.2 กรณีที่ 2

- Entity มีความสัมพันธ์กับแบบ Partial Participation
ให้แปลงทั้ง 2 Entity และ Relationship ไปเป็นแต่ละ Relation
แต่ Relation ของ Relationship จะประกอบด้วย Attribute ของ
Relationship เองรวมกับ Identifier ของทั้ง 2 Entity ที่สัมพันธ์
กับ Relationship นั้น ส่วน Primary Key ให้เลือกจาก
Identifier ของ Entity ใด Entity หนึ่งที่สัมพันธ์กับ
Relationship นั้น ดังตัวอย่าง

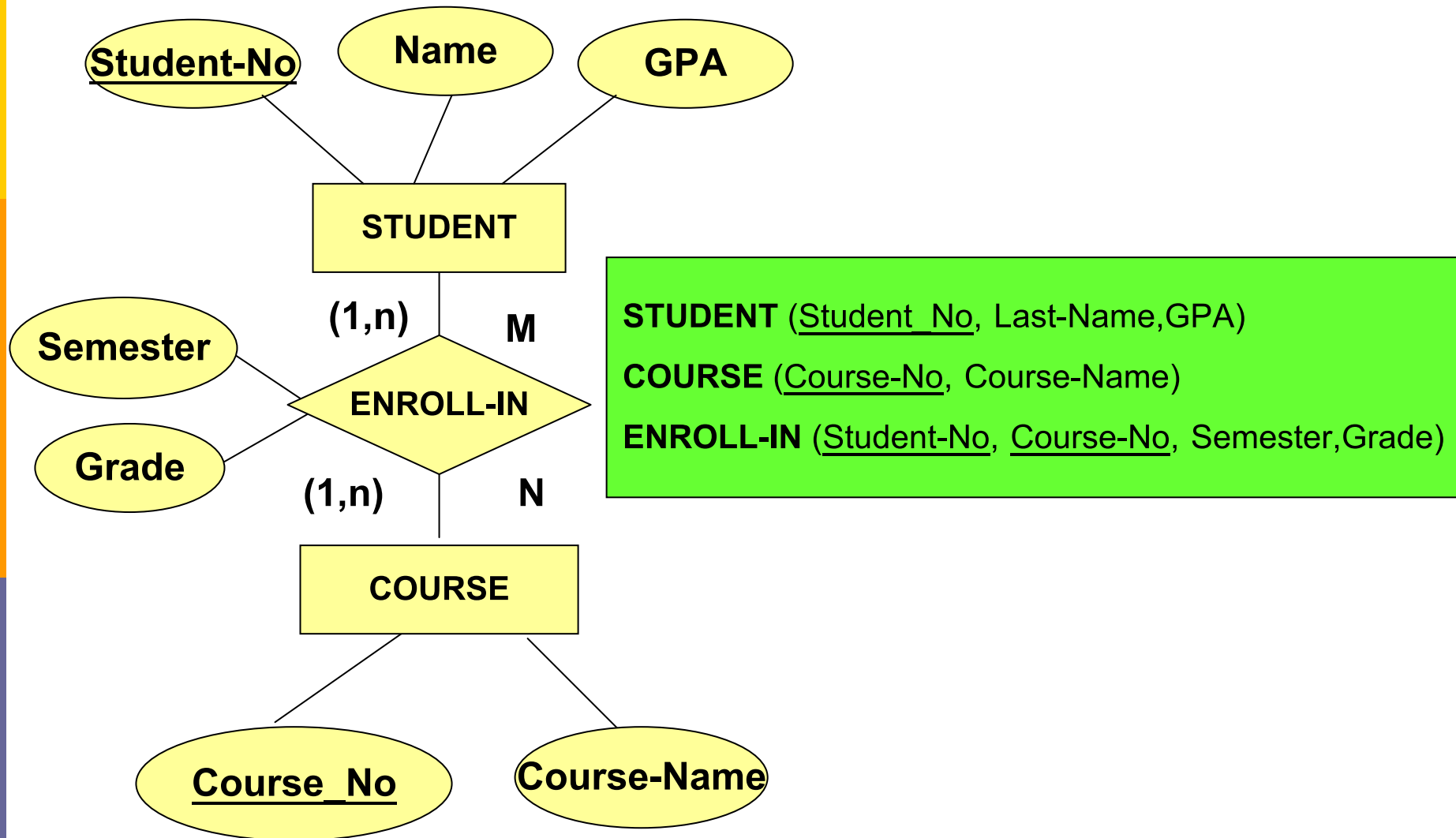
การแปลงโครงร่างของ Relationship ให้อยู่ในรูปของ Relational Schema



5.4 กรณีที่เป็น Many-to-Many Relationship

- ให้ทำการแปลงทั้ง 2 Entity และ Relationship ไปเป็นแต่ละ Relation แต่ Relation ของ Relationship จะประกอบด้วย Attribute ของ Relationship เอง รวมกับ Identifier ของทั้ง 2 Entity ที่สัมพันธ์กับ Relationship นั้น ส่วน Primary Key ให้เลือกจาก Identifier ของทั้ง 2 Entity ที่สัมพันธ์กับ Relationship นั้น ดังตัวอย่าง

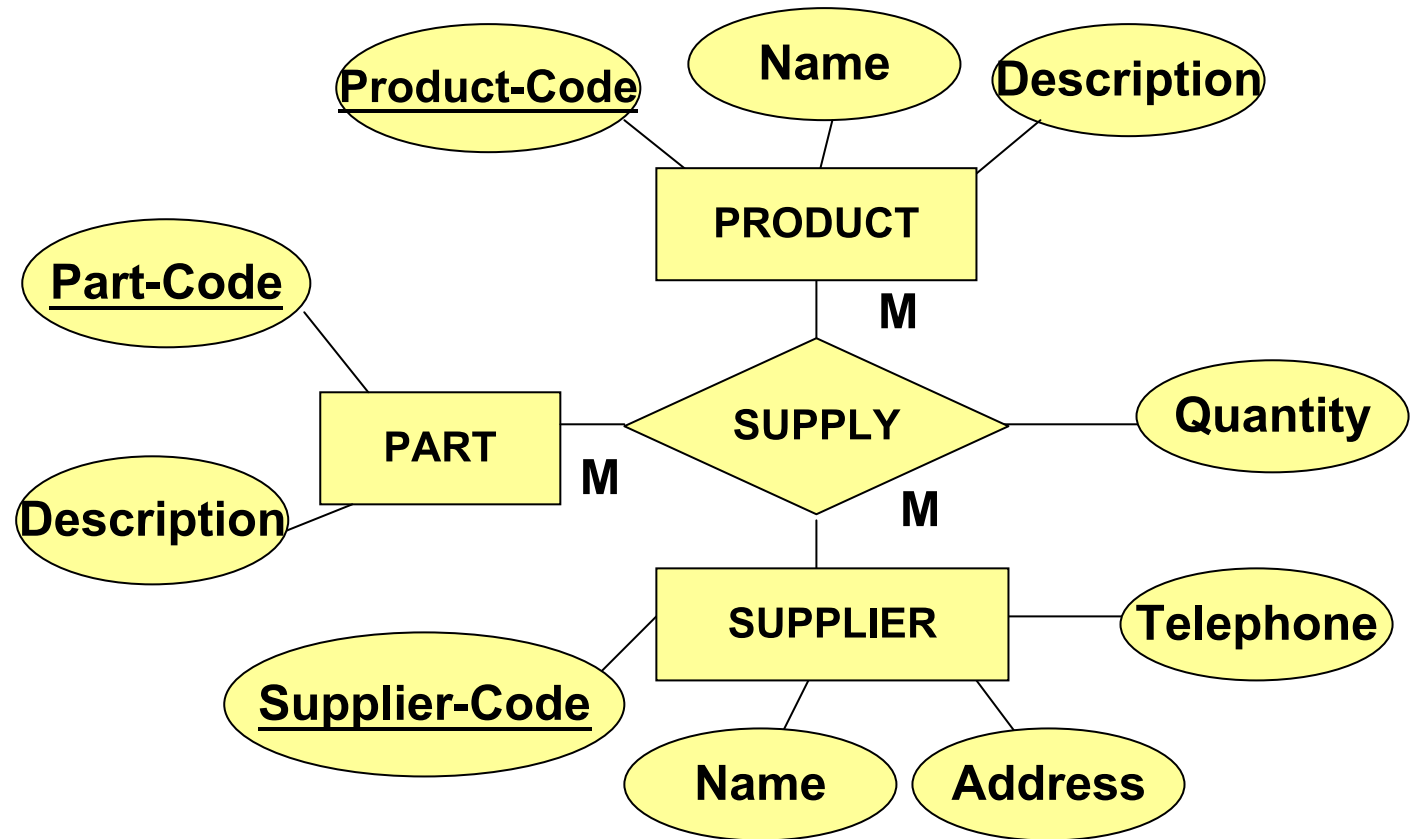
การแปลงโครงร่างของ Relationship ให้อยู่ในรูปของ Relational Schema



5.5 กรณีที่เป็น N-ary Relationship

- ให้แปลงทุก Entity และ Relationship ไปเป็นแต่ละ Relation แต่ Relation ของ Relationship จะประกอบด้วย Attribute ของ Relationship เอง รวมกับ Identifier ของทุก Entity ที่สัมพันธ์กับ Relationship นั้น ส่วน Primary Key ให้เลือกจาก Identifier ของทุก Entity ที่สัมพันธ์กับ Relationship นั้น ดังตัวอย่าง

การแปลงโครงร่างของ Relationship ให้อยู่ในรูปของ Relational Schema



PRODUCT (Product-Code, Name, Description)

PART (Part-Code, Description)

SUPPLIER (Supplier-Code, Name, Address, Telephone)

SUPPLY (Product-Code, Part-Code, Supplier-Code, Quantity)

5.6 กรณีที่เป็น Recursive Relationship

- ให้แปลงทุก Entity และ Relationship ไปเป็นแต่ละ Relation แต่ Relation ของ Relationship จะประกอบขึ้นจาก Identifier ของ Entity ที่สัมพันธ์กับ Relationship นั้นจำนวน 2 ชุด ซึ่งแต่ละชุดจะให้ชื่อตาม Role ของ Entity ใน Relationship นั้นรวมกับ Attribute ของ Relationship เอง ส่วนการเลือก Primary Key จะสามารถทำได้ 2 กรณีดังนี้

5.6 กรณีที่เป็น Recursive Relationship (ต่อ)

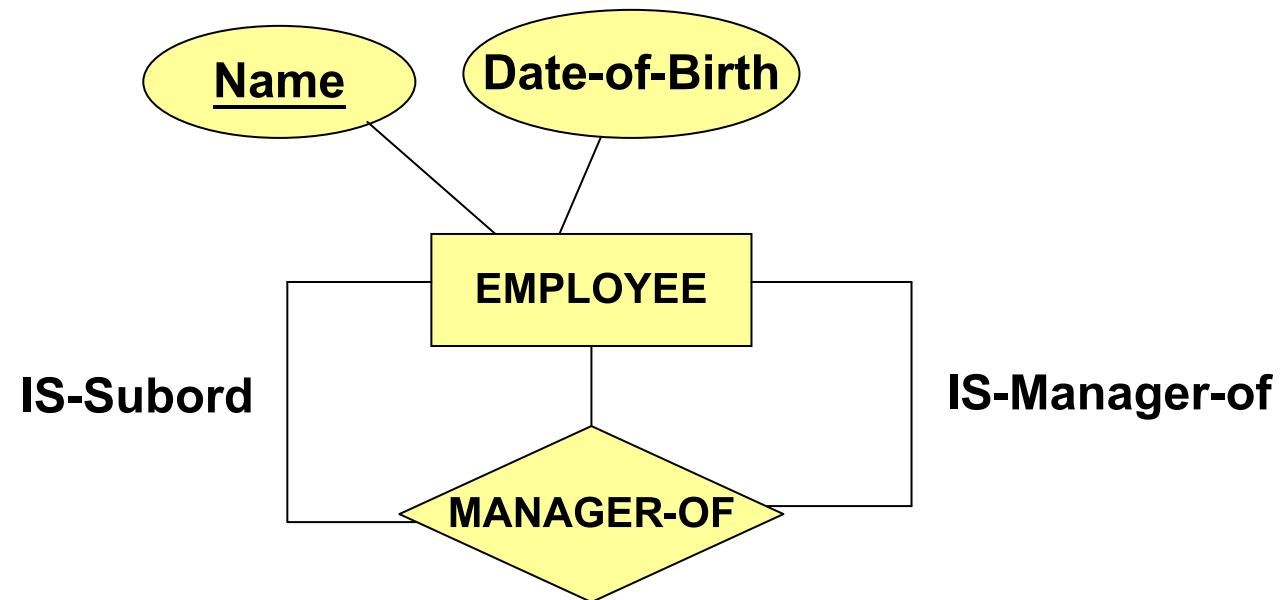
5.6.1 กรณีเป็น One-to-Many Relationship

ให้เลือก Primary Key จาก Identifier ทางด้าน Many

5.6.2 กรณีเป็น Many-to-Many Relationship

ให้เลือก Primary Key 2 ด้าน รวมกัน

Recursive Relationship



กรณีที่เป็น Recursive Relationship (ต่อ)

1. กรณีกำหนดให้ลูกจ้างแต่ละคนสามารถมีนายจ้างได้หลายคน ความสัมพันธ์จึงอยู่ในลักษณะ Many-to-Many ดังนั้นทั้ง Attribute “NAME-OF-MANAGER” และ “NAME-OF-SUBORDINATE” จึงถูกกำหนดให้เป็น Primary Key ของ Relation ดังนี้

EMPLOYEE (Name, Date-Of-Birth)

MANAGER-OF (Name-Of-Manager, Name-of-Subordinate)

กรณีที่เป็น Recursive Relationship (ต่อ)

2. กรณีกำหนดให้ลูกจ้างแต่ละคนสามารถมีนายจ้างได้เพียงคนเดียว ความสัมพันธ์จึงอยู่ในลักษณะ One-to-Many ดังนั้น Attribute “Name-Of-Subordinate” จึงถูกกำหนดให้เป็น Primary Key ของ Relation เพียง Attribute ดังนี้

EMPLOYEE (Name, Date-Of-Birth)

MANAGER-OF (Name-Of-Subordinate, Name-of-Manager)

กรณีที่เป็น Recursive Relationship (ต่อ)

- ❑ แต่บางกรณี ถ้าค่าของ Attribute “Name-of-Subordinate” ของ Relation “Manager-Of” และ Attribute “Name” ของ Relation “Employee” มีค่าเดียวกัน สามารถยุบ Relation “Manager-Of” เพื่อไปเป็น Attribute ของ Relation “Employee”

EMPLOYEE (Name, Date-Of-Birth, Name-Of-Manager)