

## บทที่ 4

### วิธีการรวมแบบจำลองข้อมูล

หลังจากทำการเปรียบเทียบเพื่อหาความสัมพันธ์ต่างๆ จากแบบจำลองข้อมูล 2 ชุดตามวิธีที่ได้กล่าวไว้ในบทที่แล้ว บทนี้จะกล่าวถึงวิธีการรวม โดยจะนำความสัมพันธ์ที่ได้ มาพิจารณาเพื่อทำการรวมเป็นแบบจำลองข้อมูลใหม่ วิธีการรวมสามารถแบ่งได้เป็น 3 ส่วนดังนี้คือ

- 1) การรวมความหมาย (Semantic Integration)
- 2) การรวมแอตทริบิวต์ (Attribute Integration)
- 3) การรวมคลาส (Class Integration)

#### 4.1 การรวมความหมาย

วิธีการรวมความหมายเป็นการรวมความหมายหลายชุดเข้าด้วยกัน เพื่อได้เป็นความหมายใหม่ที่ยังคงครอบคลุมความหมายเดิม โดยความหมายในที่นี้คือ ความหมายของคลาสหรือความหมายของแอตทริบิวต์

เหตุผลในการรวมความหมายของคลาสเกิดได้จากหลายกรณี เช่น จากการรวมคลาสที่มีความสัมพันธ์แบบเท่ากัน ซึ่งความหมายของคลาสใหม่นี้จะได้จากการรวมความหมายจากคลาสเดิม เพื่อที่ความหมายของคลาสใหม่จะยังคงครอบคลุมความหมายของคลาสเดิม หรือการรวมคลาสที่มีความสัมพันธ์แบบซูเปอร์คลาสหรือสับคลาสซึ่งจะได้ซูเปอร์คลาสและสับคลาส โดยที่ความหมายของซูเปอร์คลาสจะต้องมีความหมายที่ครอบคลุมความหมายของสับคลาส ดังนั้นความหมายของซูเปอร์คลาสจะได้จากการรวมความหมายของสับคลาสให้ด้วย เป็นต้น และสำหรับการรวมความหมายของแอตทริบิวต์ จะเกิดขึ้นเมื่อมีการรวมแอตทริบิวต์ทั้งสองเข้าด้วยกัน

วิธีการรวมความหมาย มีดังนี้คือ

ให้  $T_1$  และ  $T_2$  เป็นเทอมที่จะทำการรวมความหมาย ซึ่งเทอมนี้อาจหมายถึงคลาสหรือแอตทริบิวต์ ดังนั้นความหมายที่จะทำการรวมอาจหมายถึงความหมายของคลาสหรือความหมายของแอตทริบิวต์ตามลำดับ

และให้  $T_3$  เป็นเทอมที่ได้จากการรวมความหมายของ  $T_1$  และ  $T_2$

แต่ละความหมายที่มีทั้งใน  $T_1$  และ  $T_2$  จะจะเป็นความหมายของ  $T_3$  โดยค่าของแต่ละความหมายจะได้จากการยูเนียน (Union) ค่าของความหมายนั้นๆ ใน  $T_1$  และ  $T_2$

ตัวอย่างจากการรวมความหมายของคลาส Student และคลาส GradStudent มีดังนี้คือ

Class semantics (คลาส Student) :

Faculty = {'science','engineer'} Major = {'computer','math'}

Class semantics (คลาส GradStudent) :

Faculty = {'science','account','engineer'}

Major = {'computer','stat','math','electronics'} Age = {40...65}

ความหมายรวมที่ได้ :

Faculty = {'science','account','engineer'}

Major = {'computer','math','stat','electronics'}

#### 4.2 การรวมแอตทริบิวต์

ในการเปรียบเทียบแอตทริบิวต์ a ของคลาส A และแอตทริบิวต์ b ของคลาส B และได้ว่าแอตทริบิวต์ทั้งสองมีความสัมพันธ์กัน (ไม่ใช่ความสัมพันธ์แบบแตกต่างกัน) เราจะทำการรวมแอตทริบิวต์ทั้งสองเข้าด้วยกัน โดยจะพิจารณาส่วนประกอบของแอตทริบิวต์ ดังนี้คือ

- 1) ความหมายของแอตทริบิวต์
- 2) ชนิดข้อมูลของแอตทริบิวต์
- 3) ข้อกำหนดคุณภาพของแอตทริบิวต์

##### 4.2.1 ความหมายของแอตทริบิวต์

เมื่อทำการรวมสองแอตทริบิวต์เข้าด้วยกัน จะต้องมีการรวมความหมายของแอตทริบิวต์ทั้งสอง (ตามวิธีในข้อ 4.1) และจากขั้นตอนการเปรียบเทียบแอตทริบิวต์ จะทราบได้ว่าแอตทริบิวต์ทั้งสองควรมีฟังก์ชันในการแปลงความหมายของแอตทริบิวต์หรือไม่ ถ้ามี ผู้ทำการรวมจะต้องสร้างฟังก์ชันเพื่อใช้แปลงความหมายของแอตทริบิวต์นี้ให้เหมือนกัน เช่นในการรวมความหมายของแอตทริบิวต์ height ใน 2 คลาสซึ่งมีความหมายดังนี้

```
height : real <unit-of-height = {'centimetre'}>
```

//คลาส Student (Schema1)

```
height : real <unit-of-height = {'metre'}>
```

//คลาส GradStudent (Schema2)

ซึ่งจากการเปรียบเทียบทั้งสองแอตทริบิวต์นี้จะได้ว่า ทั้งสองมีความสัมพันธ์แบบที่น้องเมื่อทำการรวมแอตทริบิวต์ทั้งสองจะได้

[Attribute] : height

DataType : real ...

<Attribute Semantic> :

Semantic : unit-of-height · Value : centimetre,metre

Map function : true

...

กรณีนี้ผู้ทำการรวมจะต้องสร้างฟังก์ชันเพื่อแปลงค่าของความหมายให้เหมือนกัน (เมื่อนำแบบจำลองข้อมูลรวมไปสร้างเป็นสก็มารวมบนฐานข้อมูลจริง) อาจเป็นฟังก์ชันเพื่อแปลงจากหน่วยเมตรเป็นเซนติเมตรหรือหน่วยเซนติเมตรเป็นเมตร เป็นต้น ตัวอย่างฟังก์ชันการแปลงหน่วยเมตรเป็นเซนติเมตรสำหรับแอตทริบิวต์ height ของคลาส GradStudent ใน Schema2 ให้มีหน่วยเหมือนกับแอตทริบิวต์ height ของคลาส Student ใน Schema1 แสดงได้ดังนี้

```
function Map_metre_height_to_centimetre (height in number) return number is
begin
    return height * 100;
end;
```

#### 4.2.2 ชนิดข้อมูลของแอตทริบิวต์

สำหรับชนิดข้อมูลที่ต่างกันและเป็นชนิดข้อมูลดั้งเดิมของระบบ (Primitive type) ในงานวิจัยนี้จะนำการแปลงชนิดข้อมูล (Type Coercion) จาก [15] มาปรับใช้ดังแสดงในตารางที่ 4.1 โดยชนิดข้อมูลของระบบที่ใช้ ได้แก่ String, Integer และ Real

ตารางที่ 4.1 หลักการแปลงชนิดข้อมูลจาก [15]

	STRING(N1)	INTEGER	REAL
STRING(N2)	STRING(max(N1,N2))		
INTEGER		INTEGER	REAL
REAL		REAL	REAL

กรณีที่แอตทริบิวต์หนึ่งมีชนิดข้อมูลเป็นของระบบ แต่อีกแอตทริบิวต์มีชนิดข้อมูลเป็นคลาสอื่น แอตทริบิวต์รวมที่ได้จะมีชนิดข้อมูลเป็นคลาส ซึ่งหมายความว่าแอตทริบิวต์ที่มีชนิดข้อมูลเป็นของระบบ จะต้องมีการแปลงแอตทริบิวต์เดิมไปเป็นแอตทริบิวต์ต่างๆ ในคลาสนั้น เช่น ในการรวมแอตทริบิวต์ color ซึ่งมีความสัมพันธ์เป็นแบบเท่ากันและมีชนิดข้อมูลต่างกันดังนี้

color : string(15) // Class Car (Schema1)

color : Color // Class Car (Schema2)

โดยที่ Class Color (Schema2) มีแอตทริบิวต์ red\_id,green\_id,blue\_id

เมื่อทำการรวมสองแอตทริบิวต์นี้จะได้เป็น

*[Attribute] : color*

*DataType : Color ...*

*<Map Attribute From> :*

*Attribute :color Class :Car Schema :Schema1*

*DataType :String Size :15 Set : false*

*MapDataTypeFunction : Required*

*Attribute :color Class :Car Schema :Schema2*

*DataType :Color Set : false*

กรณีนี้ผู้ทำการรวมจะต้องสร้างฟังก์ชันเพื่อแปลงค่าแอตทริบิวต์ color ในคลาส Class ใน Schema1 ไปเป็นแอตทริบิวต์ red\_id, green\_id, blue\_id ซึ่งตัวอย่างของฟังก์ชันนี้อาจมีลักษณะดังนี้

```
function map_color_id(gr in varchar) return t_color is
    col t_color;           // t_color is type of ( RED int, GREEN int, BLUE int)
begin
    if(gr='black')then
        col:=t_color(0,0,0);
    elsif (gr='white')then
        col:=t_color(255,255,255);
    elsif (gr='brown')then
        col:=t_color(200,250,250);
    end if;
    return col;
end;
```

สำหรับกรณีที่แอตทริบิวต์ทั้งสองมีชนิดข้อมูลเป็นคลาสทั้งคู่ เมื่อทำการรวมสองแอตทริบิวต์นี้จะทำการพิจารณาดังนี้

- ถ้าคลาสชนิดข้อมูลทั้งสองเป็นซูเปอร์คลาสและสับคลาสกัน จะเลือกคลาสที่เป็นซูเปอร์คลาส เป็นชนิดข้อมูลของแอตทริบิวต์จากการรวม
- ถ้าคลาสชนิดข้อมูลทั้งสองเป็นคลาสพี่น้องกัน จะเลือกชนิดข้อมูลที่ เป็นซูเปอร์คลาสของคลาสพี่น้องทั้งสอง เป็นชนิดข้อมูลของแอตทริบิวต์จากการรวม

#### 4.2.3 ข้อกำหนดคุณภาพของแอตทริบิวต์

ข้อกำหนดคุณภาพของแอตทริบิวต์จะพิจารณาเพื่อใช้ในการรวม 2 ลักษณะคือ

- 1) ค่าที่เป็นไปได้ของแอตทริบิวต์
- 2) การขึ้นกับค่าของแอตทริบิวต์อื่น

##### 4.2.3.1 ค่าที่เป็นไปได้ของแอตทริบิวต์

ผลที่ได้จากขั้นตอนการเปรียบเทียบ จะมีการระบุว่าแอตทริบิวต์ทั้งสองจะต้องมีฟังก์ชันการแปลงค่าที่เป็นไปได้หรือไม่ เช่น ในการรวมแอตทริบิวต์ sex และ gender ซึ่งมีความสัมพันธ์แบบความหมายเหมือนกันแต่มีค่าที่เป็นไปได้ต่างกัน

```
sex : string(1) //Class Student (Schema1)
```

```
Integrity constraint : sex = {'F','M'}
```

```
gender : string(10) //Class GradStudent (Schema2)
```

```
Integrity constraint : gender = {'Female','Male'}
```

เมื่อทำการรวมแอตทริบิวต์ทั้งสองจะได้เป็น

```
[Attribute] : sex
```

```
DataType : string Size : 10 Set : false
```

```
<Attribute Constraint> Possible Value : F,Female,M,Male
```

```
MapValueFunction : Required
```

```
<Map Attribute From> :
```

```
Attribute :sex Class :Student Schema :Schema1
```

```
DataType :String Size :1 Set : false
```

```
Possible Value : F,M
```

```
Attribute :gender Class :GradStudent Schema :Schema2
```

```
DataType :String Size :10 Set : false
```

```
Possible Value : Female,Male
```

กรณีนี้แอตทริบิวต์ทั้งสองมีชื่อต่างกัน ชื่อแอตทริบิวต์จากการรวมจะได้จากการเลือกชื่อใดชื่อหนึ่งเป็นชื่อใหม่ และผู้ทำการรวมจะต้องสร้างฟังก์ชันเพื่อแปลงค่าที่เป็นไปได้ของแอตทริบิวต์ไว้เหมือนกัน จากตัวอย่างอาจจะสร้างฟังก์ชันแปลงค่า F และ M เป็น Female และ Male ตามลำดับหรือแปลงค่า Female และ Male เป็น F และ M เป็นต้น ตัวอย่างฟังก์ชันที่ใช้แปลงค่า F และ M เป็น Female และ Male สำหรับแอตทริบิวต์ sex ของคลาส Student ใน Schema1 มีดังนี้

```
function Map_sex_value (sex in varchar) return varchar is
```

```
new_sex varchar;
```

```

begin
  if(sex = 'F') then new_sex = 'Female';
  elsif (sex = 'M') then new_sex = 'Male';
  return new_sex;
end;

```



#### 4.2.3.2 การขึ้นกับค่าของแอตทริบิวต์อื่น

ในขั้นตอนการเปรียบเทียบจะมีการพิจารณาการขึ้นกับค่าของแอตทริบิวต์อื่น และจะมีการระบุว่าต้องการฟังก์ชันการแปลงค่าที่เป็นไปได้สำหรับแอตทริบิวต์ที่มีการขึ้นต่อกันหรือไม่ เช่น คลาส Student ใน Schema1 มีการกำหนดการขึ้นกับค่าของแอตทริบิวต์อื่น สำหรับแอตทริบิวต์ title และ sex ดังนี้

title='Ms'  $\leftrightarrow$  sex='F' และ title='Mr'  $\leftrightarrow$  sex='M' .....(4.1)

หมายความว่า แอตทริบิวต์ title จะขึ้นกับค่าของแอตทริบิวต์ sex เช่นถ้าค่าของแอตทริบิวต์ sex เป็น 'F' แล้ว ค่าของแอตทริบิวต์ title จะเป็น 'Ms' และถ้าค่าของแอตทริบิวต์ sex เป็น 'M' แล้ว ค่าของแอตทริบิวต์ title จะเป็น 'Mr'

และมีการกำหนดค่าที่เป็นไปได้ของแอตทริบิวต์ทั้งสองดังนี้

title={'Ms','Mr'} และ sex={'F','M'}

คลาส GradStudent ใน Schema2 ไม่มีการกำหนดการขึ้นกับค่าของแอตทริบิวต์ สำหรับแอตทริบิวต์ gender และ title แต่มีการกำหนดค่าที่เป็นไปได้ของแอตทริบิวต์ sex ดังนี้

sex={'Female','Male'}

ผลที่ได้จากการเปรียบเทียบจะได้ว่า แอตทริบิวต์ title (คลาส Student/Schema1) กับแอตทริบิวต์ title (คลาส GradStudent/Schema2) มีความสัมพันธ์แบบเท่ากัน ซึ่งสามารถรวมกันได้และแอตทริบิวต์ sex (คลาส Student/Schema1) กับแอตทริบิวต์ gender (คลาส GradStudent/Schema2) มีความสัมพันธ์แบบความหมายเหมือนกัน ซึ่งสามารถรวมกันได้ ทำให้ได้แอตทริบิวต์ในแบบจำลองข้อมูลรวม คือ title และ sex โดยแอตทริบิวต์ sex จะมีฟังก์ชันการแปลงค่าสำหรับค่าที่แตกต่างของแอตทริบิวต์ sex และ gender (ดังตัวอย่าง 4.2.3.1 ซึ่งผู้ทำกรรวมจะเลือกให้ค่าที่เป็นไปได้ของแอตทริบิวต์ sex ในแบบจำลองรวมคือ Female และ Male) แต่สำหรับแอตทริบิวต์ title ซึ่งได้จากแอตทริบิวต์ title (คลาส Student / Schema1) ซึ่งมีการขึ้นกับค่าของแอตทริบิวต์ sex (คลาส Student / Schema1) จากการระบุใน (4.1) ส่งผลให้แอตทริบิวต์ title ขึ้นกับแอตทริบิวต์ sex ในแบบจำลองรวมด้วย ดังนั้นในแบบจำลองข้อมูลรวมจะมีการระบุว่า แอตทริบิวต์ title ควรจะมีฟังก์ชันการแปลงค่าโดยจะพิจารณาพร้อมกับแอตทริบิวต์ที่มีการขึ้นต่อกันด้วย ในที่นี้คือแอตทริบิวต์ sex ซึ่งแสดงได้ดังนี้



```

[Attribute] : title
              DataType : string   Size : 6   Set : false
              MapValueFunction : Required
              AttributeConsideredTogether : Attribute : sex   Class: Student
              <Map Attribute From> :
                Attribute :title   Class :Student   Schema :Schema1
                DataType :String   Size : 5   Set : false
                Possible Value : Mr,Ms
                Attribute :title   Class :GradStudent   Schema :Schema2
                DataType :String   Size : 6   Set : false

```

กรณีนี้ผู้ทำการรวมอาจกำหนดการขึ้นต่อกันระหว่างแอตทริบิวต์ title และ sex ดังแสดงใน (4.2)

title='Miss' ↔ sex='Female' และ title='Mister' ↔ sex='Male' .....(4.2)

ดังนั้นแอตทริบิวต์ title ที่ได้จากการรวมแอตทริบิวต์ title ของคลาส Student (Schema1) กับแอตทริบิวต์ title ของคลาส GradStudent (Schema2) เข้าด้วยกัน ควรมีฟังก์ชันการแปลงค่าโดยการพิจารณาค่าของแอตทริบิวต์ sex ในคลาส Student และคลาส GradStudent เป็นดังนี้

```

function Map_title_value (sex in varchar) return varchar is
    new_title varchar;
begin
    if (sex = 'F') then new_title = 'Miss';           //class Student / Schema1
    elsif (sex = 'M') then new_title = 'Mister';
    elsif (sex = 'Female') then new_title = 'Miss'; //class GradStudent / Schema2
    elsif (sex = 'Male') then new_title = 'Mister'
    return new_title;
end;

```

#### 4.3 การรวมคลาส

ความสัมพันธ์ที่ได้จากการเปรียบเทียบคลาสต่างๆ จากแบบจำลองข้อมูล 2 ชุด จะนำมาใช้ในการพิจารณาเพื่อทำการรวมคลาส รูปที่ 4.1 แสดงขั้นตอนโดยรวมในการรวมคลาสต่างๆ ที่มีความสัมพันธ์กัน ซึ่งความสัมพันธ์ที่ได้จากขั้นตอนการเปรียบเทียบคลาสได้แก่ ความสัมพันธ์แบบเท่ากัน ความสัมพันธ์แบบซูเปอร์คลาส ความสัมพันธ์แบบสับคลาส ความสัมพันธ์แบบพี่น้อง หรือความสัมพันธ์แบบแตกต่างกันนั้น จะแสดงถึงความเหมือนและความแตกต่างในระดับที่แตกต่างกันออกไป ในงานวิจัยนี้จะทำการจัดลำดับความสำคัญ (Priority) ของความสัมพันธ์ที่คลาสหนึ่ง

ในแบบจำลองข้อมูลหนึ่งจะมีต่อคลาสต่างๆ ในอีกแบบจำลองหนึ่งและพิจารณารวมคลาสนั้นตามความสัมพันธ์ที่มีความสำคัญที่สุดก่อน เพราะการรวมในลำดับที่ต่างกันจะได้รูปแบบที่ต่างกันได้ ทั้งนี้การจัดลำดับความสำคัญจากมากไปน้อย ได้แก่ (1) ความสัมพันธ์แบบเท่ากัน (2) ความสัมพันธ์แบบซูเปอร์คลาส/สับคลาส (3) ความสัมพันธ์แบบพี่น้อง (4) ความสัมพันธ์แบบแตกต่างกัน ตัวอย่างเช่น คลาส A ในแบบจำลอง Schema1 มีความสัมพันธ์แบบเท่ากันกับคลาส B ในแบบจำลอง Schema2 และมีความสัมพันธ์แบบซูเปอร์คลาสกับคลาส C ในแบบจำลอง Schema2 เช่นกัน ในการสร้างแบบจำลองรวมจะพิจารณารวม A และ B ตามหลักการรวมความสัมพันธ์แบบ เท่ากันก่อน จากนั้นจึงทำการรวม A กับ C ตามหลักการรวมความสัมพันธ์แบบซูเปอร์คลาส

จากรูปที่ 4.1 วิธีในการรวมคลาสต่างๆ เพื่อได้เป็นแบบจำลองข้อมูลรวม สามารถอธิบายได้เป็นหลายส่วน ดังนี้คือ

- 1) การรวมคลาสโดยการพิจารณาความสัมพันธ์
- 2) การเชื่อมความสัมพันธ์แบบแอกกรีเกชันระหว่างคลาสรวมที่ได้จาก 1)
- 3) การรวมค่าที่มีความหมายเหมือนกันหรือค่าที่เป็นส่วนประกอบกันสำหรับแอตทริบิวต์

#### 4.3.1 การรวมคลาสโดยการพิจารณาความสัมพันธ์

วิธีการรวมคลาส สามารถแบ่งได้เป็นหลายกรณีตามลักษณะความสัมพันธ์ที่ได้จากการเปรียบเทียบแต่ละคู่ของคลาส ดังนี้คือ

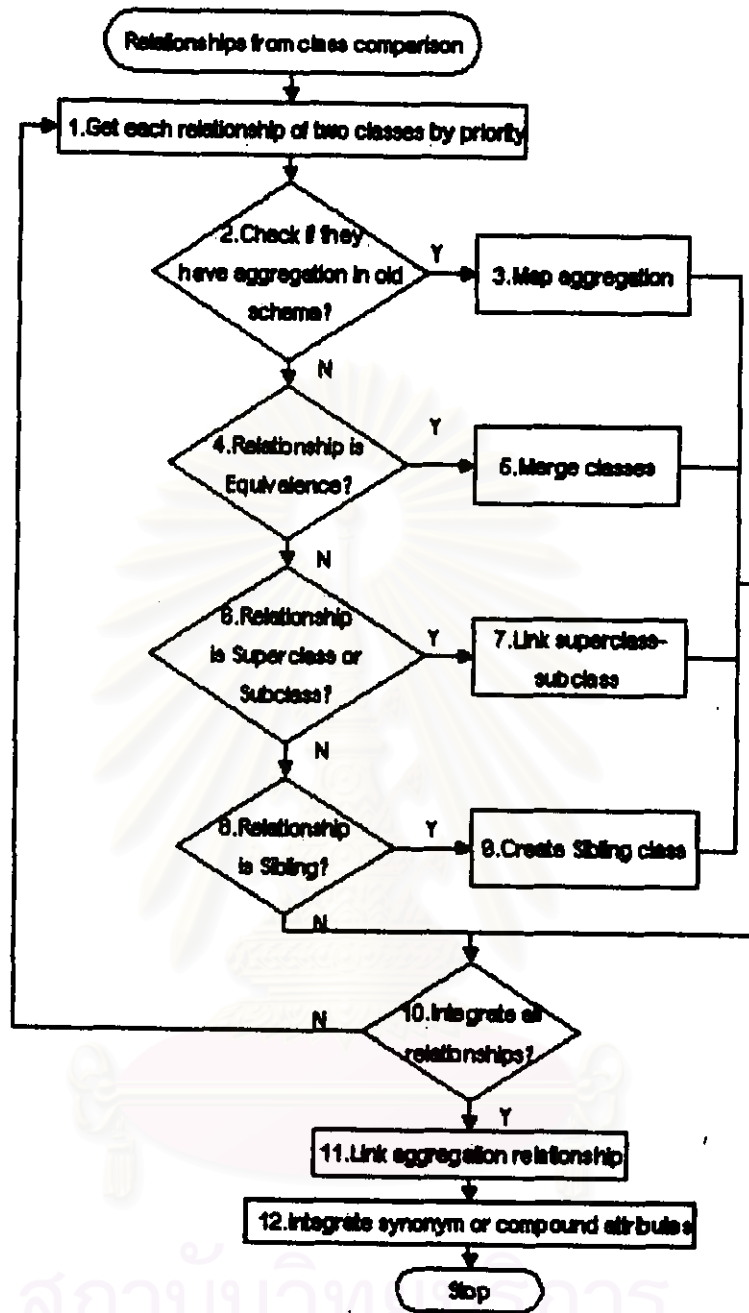
กรณีที่ 1 การรวมคลาสที่มีความสัมพันธ์แบบเท่ากัน (ขั้นตอนที่ 5 จากรูปที่ 4.1)

เมื่อคลาส 2 คลาสมีความสัมพันธ์แบบเท่ากัน จะทำการรวมคลาสทั้งสองเข้าด้วยกันโดยมีขั้นตอน ดังนี้คือ

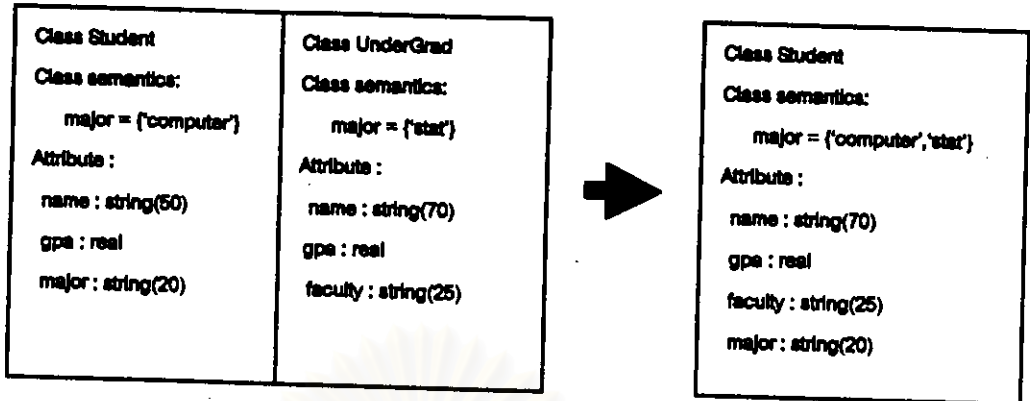
1. รวมความหมายของคลาสทั้งสอง
2. รวมชุดแอตทริบิวต์ของคลาสทั้งสอง โดยพิจารณาความสัมพันธ์ระหว่างแอตทริบิวต์ที่นำมารวมกัน

รูปที่ 4.2 เป็นตัวอย่างการรวมคลาส Student ใน Schema1 และคลาส UnderGrad ใน Schema2 ที่มีความสัมพันธ์แบบเท่ากัน โดยคลาสรวมที่ได้มีรายละเอียดของการรวมดังแสดงในรูปที่ 4.3

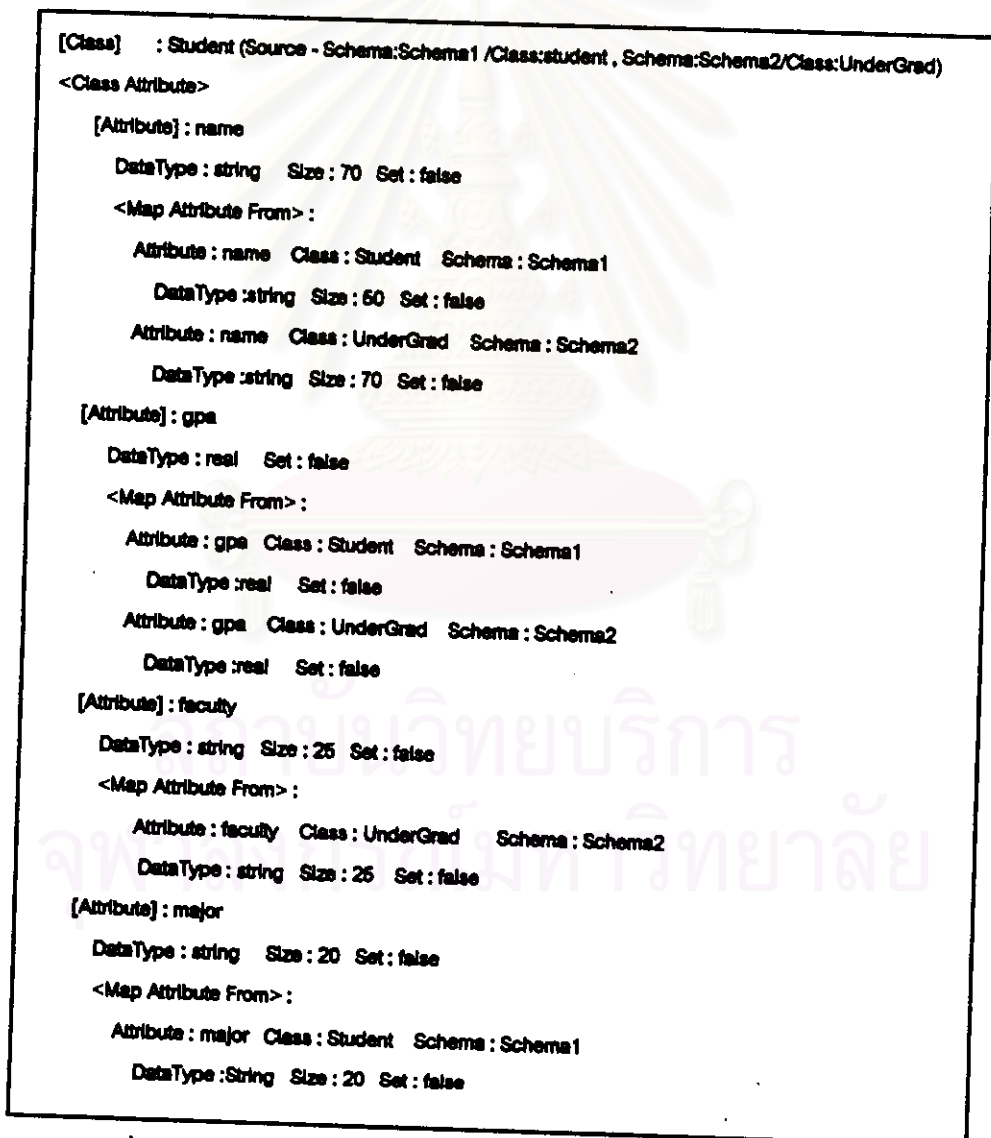




รูปที่ 4.1 ขั้นตอนโดยรวมในการรวมคลาส



รูปที่ 4.2 ตัวอย่างการรวมคลาสที่มีความสัมพันธ์แบบเท่ากัน

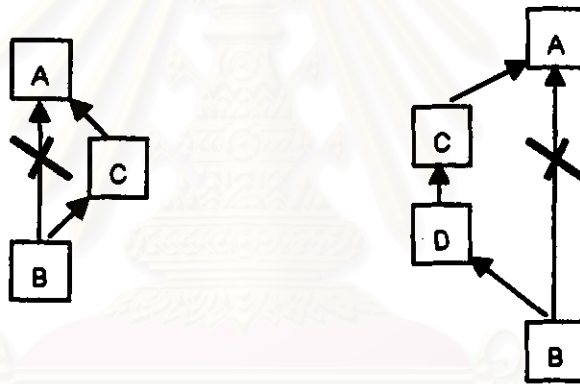


รูปที่ 4.3 รายละเอียดของคลาสรวมที่ได้จากการรวมคลาสในรูปที่ 4.2

**กรณีที่ 2** การรวมคลาสที่มีความสัมพันธ์แบบยูเออร์คลาส/สับคลาส(ขั้นตอนที่ 7 จาก รูปที่ 4.1)

ในกรณีที่คลาส A มีความสัมพันธ์แบบยูเออร์คลาสกับคลาส B หรือคลาส B มีความสัมพันธ์แบบสับคลาสกับคลาส A ขั้นตอนในการรวมคลาสดังนี้มีดังนี้ คือ

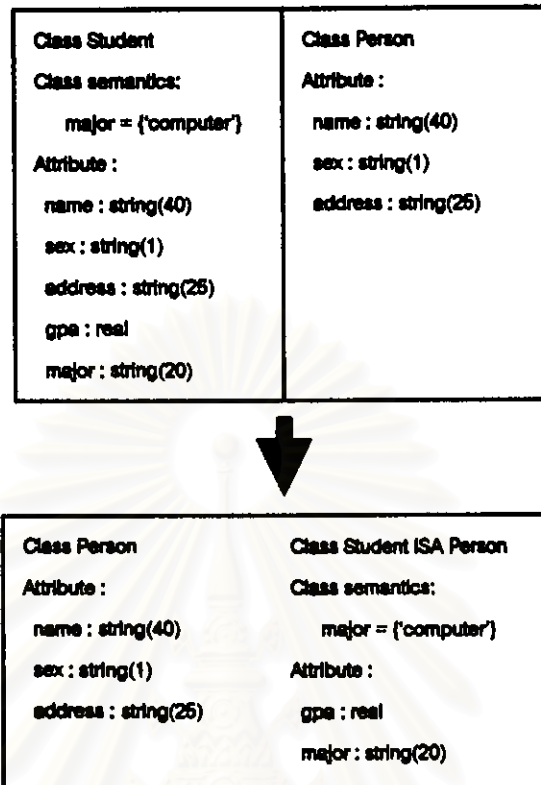
1. เชื่อมความสัมพันธ์โดยให้คลาส A เป็นยูเออร์คลาสของคลาส B
2. ความหมายของคลาส A ได้จากการรวมความหมายเดิมของคลาส A กับ ความหมายของคลาส B โดยที่ความหมายของคลาส B ยังคงเดิม
3. แอตทริบิวท์ของคลาส B ที่มีความสัมพันธ์กับแอตทริบิวท์ของคลาส A จะทำการรวมกันในคลาส A
4. กำจัดเส้นทางการสืบทอดที่ซ้ำซ้อน โดยพิจารณาว่าถ้ามีรูป (loop) ระหว่าง A, B และคลาสอื่นในแบบจำลองรวม จะตัดเส้นทางระหว่าง A และ B ดัง แสดงในรูปที่ 4.4



**รูปที่ 4.4** การตัดความสัมพันธ์การสืบทอดที่ซ้ำซ้อน

**รูปที่ 4.5** เป็นตัวอย่างการรวมคลาส Student และคลาส Person ซึ่งคลาส Student มีความสัมพันธ์แบบสับคลาสกับคลาส Person

จุฬาลงกรณ์มหาวิทยาลัย



**รูปที่ 4.5** ตัวอย่างการรวมคลาสที่มีความสัมพันธ์แบบซูเปอร์คลาส/สับคลาส

**กรณีที่ 3** การรวมคลาสที่มีความสัมพันธ์แบบพี่น้อง (ขั้นตอนที่ 9 จากรูปที่ 4.1)

กรณีที่คลาส A มีความสัมพันธ์แบบพี่น้องกับคลาส B ขั้นตอนในการรวมคลาสทั้งสอง จะพิจารณาว่าถ้าคลาส A และ B ในแบบจำลองข้อมูลรวมไม่มีการสืบทอดจากซูเปอร์คลาส หรือทั้งคลาส A และ B ไม่มีซูเปอร์คลาสโดยตรง (Immediate superclass) ร่วมกัน จะทำตามขั้นตอนดังนี้คือ

1. สร้างคลาสใหม่คือคลาส C ไม่เป็นซูเปอร์คลาสโดยตรงของคลาส A และ B
2. ความหมายของคลาส C ได้จากการรวมความหมายของคลาส A และ B โดยที่ความหมายของคลาส A และ B ยังคงเดิม
3. แอตทริบิวท์ของคลาส A และ B ที่มีความสัมพันธ์กัน จะทำการรวมกันและย้ายไปไว้ในคลาส C
4. หาความสัมพันธ์ของคลาส C กับซูเปอร์คลาสทั้งหลายของคลาส A และ B ซึ่งถ้ามีความสัมพันธ์กันจะทำการรวมกัน โดยพิจารณาจากความสัมพันธ์ในทำนองเดียวกัน

ในกรณีทีคลาส A และ B มีซูเปอร์คลาสโดยตรงร่วมกัน จะไม่ทำการรวมเข้าด้วยกัน เพราะถือว่าคลาส A และ B เป็นคลาสที่น้องกันอยู่แล้ว

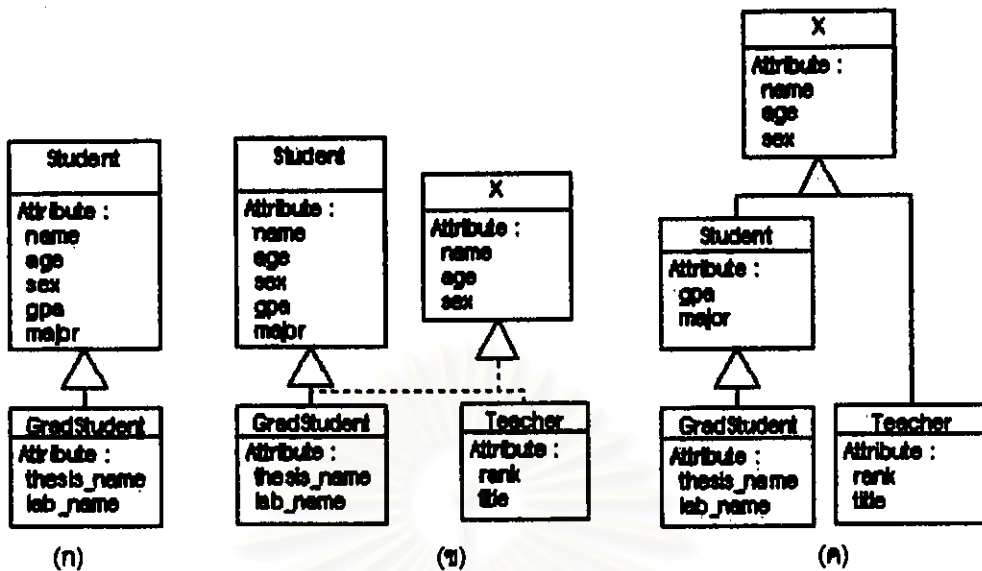
รูปที่ 4.6 เป็นตัวอย่างแบบจำลองข้อมูล 2 ชุด ซึ่งเมื่อทำการเปรียบเทียบคลาสต่างๆ จะได้ความสัมพันธ์ดังแสดงในตารางที่ 4.2 และจะทำการรวมตามขั้นตอนที่แสดงด้วยแผนภาพในรูปที่ 4.7 ซึ่งในที่สุดจะได้แบบจำลองข้อมูลรวมดังรูปที่ 4.8

Schema1	Schema2	
Class GradStudent	Class Student	Class Teacher
Attribute :	Attribute :	Attribute :
name : string(50)	name : string(50)	name : string(50)
age : integer	age : integer	age : integer
sex : string(1)	sex : string(1)	sex : string(1)
gpa : real	gpa : real	rank : string(35)
major : string(20)	major : string(20)	title : string(30)
thesis_name : string(100)		
lab_name : string(30)		

รูปที่ 4.6 ตัวอย่างแบบจำลองข้อมูล 2 ชุดสำหรับการรวมคลาสที่มีความสัมพันธ์แบบที่น้อง

ตารางที่ 4.2 ความสัมพันธ์จากการเปรียบเทียบคลาสในรูปที่ 4.6

	Schema2	Student	Teacher
Schema1			
GradStudent		Subclass	Sibling



รูปที่ 4.7 ขั้นตอนการรวมคลาสโดยแผนภาพจากแบบจำลองข้อมูลในรูปที่ 4.6

Class Person	Class Student ISA Person	Class Teacher ISA Person	Class GradStudent ISA Student
Attribute :	Attribute :	Attribute :	Attribute :
name : string(50)	gpa : real	rank : string(35)	thesis_name : string(100)
age : integer	major : string(20)	title : string(30)	lab_name : string(30)
sex : string(1)			

รูปที่ 4.8 แบบจำลองข้อมูลรวมที่ได้จากการรวมแบบจำลองข้อมูลในรูปที่ 4.6

จากความสัมพันธ์ในตารางที่ 4.2 จะทำการรวมคลาสต่างๆ ดังนี้

- รวมคลาส GradStudent และคลาส Student ด้วยความสัมพันธ์แบบสับคลาส(ดังรูปที่ 4.7 (ก))
- รวมคลาส GradStudent และคลาส Teacher ด้วยความสัมพันธ์แบบพี่น้องโดยมีขั้นตอนการรวมความสัมพันธ์แบบพี่น้องดังนี้
  1. จะสร้างคลาสใหม่ ในที่นี้คือคลาส X ให้เป็นซูเปอร์คลาสของคลาส GradStudent และคลาส Teacher
  2. ความหมายของคลาส X ได้จากการรวมความหมายของคลาส GradStudent และคลาส Teacher
  3. แอดทริบิวท์ของคลาส GradStudent และคลาส Teacher ที่มีความสัมพันธ์กัน จะทำการรวมกันและย้ายมาให้ในคลาส X (ดังรูปที่ 4.7 (ข))



4. นำคลาส X นี้ไปหาความสัมพันธ์กับซูเปอร์คลาสของคลาส GradStudent และ Teacher ซึ่งในที่นี้คือคลาส Student เพียงคลาสเดียว โดยคลาส X มีความสัมพันธ์แบบซูเปอร์คลาสกับคลาส Student จากนั้นก็จะทำการรวมคลาส X กับคลาส Student ด้วยความสัมพันธ์แบบซูเปอร์คลาสซึ่งในที่สุดจะได้เป็นรูป 4.7 (ค) ทั้งนี้ผู้ทำการรวมอาจตั้งชื่อคลาส X ใหม่เป็นคลาส Person สำหรับแบบจำลองข้อมูลรวมที่ได้ ดังแสดงในรูปที่ 4.8

#### กรณีที่ 4 การรวมคลาสที่มีความสัมพันธ์แบบแตกต่างกัน

คลาสที่มีความสัมพันธ์เป็นแบบแตกต่างกันจะหมายถึงว่าคลาสทั้งสองไม่มีความสัมพันธ์กัน ดังนั้นจึงไม่ต้องทำการรวมกัน

#### กรณีที่ 5 การรวมคลาสที่มีความสัมพันธ์แบบแอกกรีเกชันในแบบจำลองข้อมูลเดิม (ขั้นตอนที่ 3 จากรูปที่ 4.1)

กรณีนี้เกิดจากคลาส A และ B มีความสัมพันธ์แบบเท่ากัน ความสัมพันธ์แบบซูเปอร์คลาส ความสัมพันธ์แบบสับคลาสหรือความสัมพันธ์แบบพี่น้อง แต่คลาส A ในแบบจำลองข้อมูลรวมได้ทำการรวมกับคลาส C (ด้วยความสัมพันธ์แบบเท่ากัน) ซึ่งอยู่ในแบบจำลองเดียวกับ B ก่อนหน้านี้แล้ว และเพราะคลาส C มีความสัมพันธ์แบบแอกกรีเกชันกับคลาส B และคลาส C กับคลาส B ไม่มีความสัมพันธ์ในลักษณะการเป็นซูเปอร์คลาสหรือสับคลาสกัน ดังนั้นเราจะไม่รวมคลาส A และ B ตามความสัมพันธ์ที่ได้จากการเปรียบเทียบ เพราะเราจะยึดตามความสัมพันธ์เดิมที่มีในแบบจำลองข้อมูลเดิมเป็นหลัก แต่จะทำการรวมแอตทริบิวต์ที่มีความสัมพันธ์กันเข้ากับแอตทริบิวต์ของคลาสที่มีความสัมพันธ์แบบแอกกรีเกชันด้วย นั่นคือแอตทริบิวต์ของคลาส A ที่มีความสัมพันธ์กับแอตทริบิวต์ของคลาส B จะทำการรวมไว้ในคลาส B จากนั้นจะทำการเชื่อมความสัมพันธ์ในลักษณะแอกกรีเกชันระหว่างคลาส A และ B ซึ่งจะได้กล่าวถึงในหัวข้อ 4.3.2 ต่อไป

รูปที่ 4.9 แสดงแบบจำลองข้อมูล Schema1 และ Schema2 ซึ่งมีความสัมพันธ์ของคลาสต่างๆ ดังตารางที่ 4.3 โดยจะมีขั้นตอนการรวมดังนี้

- 1) รวมคลาส Teacher จาก Schema1 และ Schema2 ที่มีความสัมพันธ์แบบเท่ากัน
- 2) รวมคลาส Teacher (Schema1) และคลาส Faculty (Schema2) ที่มีความสัมพันธ์แบบสับคลาสแต่คลาส Teacher ของ Schema1 ได้รวมกับคลาส Teacher ของ Schema2 แล้วในขั้น 1) และคลาส Teacher (Schema2) มี

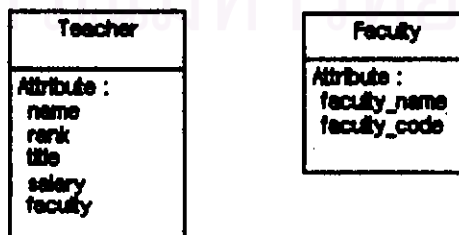
ความสัมพันธ์ในลักษณะแอกกรีเกชันกับคลาส Faculty ใน Schema2 ดังนั้น จะไม่รวมคลาส Teacher (Schema1) กับคลาส Faculty (Schema2) ด้วย ความสัมพันธ์แบบสับคลาสแต่จะทำการรวมแอตทริบิวต์ที่มีความสัมพันธ์กัน ในกรณีนี้คือจะรวมแอตทริบิวต์ faculty\_name และ faculty\_code ของ คลาส Teacher (Schema1) เข้ากับแอตทริบิวต์ faculty\_name และ faculty\_code ในคลาส Faculty ตามลำดับ ซึ่งคลาสต่างๆ ที่ได้สำหรับแบบ จำลองข้อมูลรวม จะได้ดังรูปที่ 4.10 และในที่สุดความสัมพันธ์แบบแอกกรีเกชันระหว่างคลาส Teacher และคลาส Faculty จะถูกเชื่อม โดยจะแสดงใน ส่วนต่อไป

<b>Schema1</b>	<b>Schema2</b>	<b>Class Faculty</b>
<b>Class Teacher</b>	<b>Class Teacher</b>	<b>Class Faculty</b>
<b>Attribute :</b>	<b>Attribute :</b>	<b>Attribute :</b>
name : string(50)	name : string(50)	faculty_name : string(10)
rank : string(10)	rank : string(10)	faculty_code : string(4)
salary : real	title : string(10)	
faculty_name : string(10)	salary : real	
faculty_code : string(4)	faculty : Faculty	

รูปที่ 4.9 ตัวอย่างแบบจำลองข้อมูล 2 ชุดสำหรับการรวมคลาสที่มีความสัมพันธ์แบบแอกกรีเกชันในแบบจำลองเดิม

ตารางที่ 4.3 ความสัมพันธ์จากการเปรียบเทียบคลาสในรูปที่ 4.9

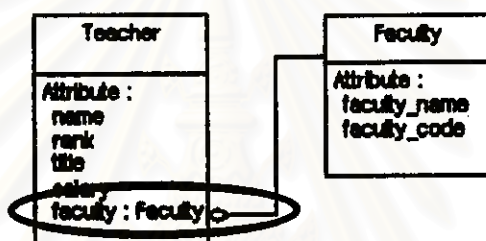
	<b>Schema2</b>	<b>Teacher</b>	<b>Faculty</b>
<b>Schema1</b>			
<b>Teacher</b>		Equivalence	Subclass



รูปที่ 4.10 คลาสต่างๆ ในแบบจำลองข้อมูลรวมที่ได้จากการรวมแบบจำลองข้อมูลในรูปที่ 4.9

#### 4.3.2 การเชื่อมความสัมพันธ์แอกกรีเกชันของคลาส (ขั้นตอนที่ 11 จากรูปที่ 4.1)

หลังจากทำการรวมคลาสต่างๆ ที่มีความสัมพันธ์กันแล้วจะทำการเชื่อมความสัมพันธ์ในลักษณะแอกกรีเกชันโดยพิจารณาว่า คลาสที่มีความสัมพันธ์แบบแอกกรีเกชันในแบบจำลองข้อมูลเดิมได้กลายเป็นคลาสใดในแบบจำลองข้อมูลรวม ซึ่งจะทำให้มีการเชื่อมความสัมพันธ์กับคลาสได้ถูกต้อง รูปที่ 4.11 จะแสดงการเชื่อมความสัมพันธ์แบบแอกกรีเกชันจากการรวมคลาสต่างๆ ที่ได้จากการรวมแบบจำลองข้อมูลในรูปที่ 4.9 ซึ่งกรณีนี้คลาส Teacher ใน Schema2 มีความสัมพันธ์แบบแอกกรีเกชันกับคลาส Faculty ใน Schema2 (ผ่านทางแอตทริบิวต์ faculty) ซึ่งคลาส Teacher และคลาส Faculty นี้ได้กลายเป็นคลาส Teacher และคลาส Faculty ในรูปที่ 4.11 ตามลำดับ ดังนั้นจะทำการเชื่อมความสัมพันธ์แบบแอกกรีเกชันนี้โดยให้ชนิดข้อมูลของแอตทริบิวต์ faculty ในคลาส Teacher เป็นคลาส Faculty



รูปที่ 4.11 แบบจำลองข้อมูลโดยแผนภาพคลาสที่ได้จากขั้นตอนการเชื่อมความสัมพันธ์แบบแอกกรีเกชันจากการรวมแบบจำลองข้อมูลในรูปที่ 4.9

#### 4.3.3 การรวมค่าที่มีความหมายเหมือนกันหรือค่าที่เป็นส่วนประกอบกันสำหรับแอตทริบิวต์ (ขั้นตอนที่ 12 จากรูปที่ 4.1)

เนื่องจากการระบุค่าที่มีความหมายเหมือนกันและค่าที่เป็นส่วนประกอบกันสำหรับแอตทริบิวต์ใดๆ สามารถระบุได้ผ่านความสัมพันธ์แบบแอกกรีเกชัน เช่น

`Schema1.Student.faculty.name = Schema2.GradStudent.faculty_name`

ซึ่งเป็นการระบุค่าที่มีความหมายเหมือนกันจากรูปที่ 3.1 โดยแอตทริบิวต์ faculty ของคลาส Student มีความสัมพันธ์แบบแอกกรีเกชันกับคลาส Faculty\_Detail ดังนั้นจึงต้องทำการรวมคลาสต่างๆ ที่มีความสัมพันธ์กันก่อนจึงจะสามารถรวมแอตทริบิวต์ต่างๆ ที่มีความหมายเหมือนกันหรือเป็นส่วนประกอบกันตามที่ผู้ทำการรวมได้ระบุไว้ได้ถูกต้อง

วิธีการรวมแอตทริบิวต์ที่มีความหมายเหมือนกัน มีดังนี้

ให้ Path1 และ Path2 เป็นเส้นทางการเข้าถึงแอตทริบิวต์ใดๆ ที่มีความหมายเหมือนกัน โดยสามารถผ่านความสัมพันธ์แบบแอกกรีเกชันกับคลาสต่างๆ ตามที่ผู้ทำการรวมได้ระบุไว้

โดยที่ Path1 = `Class.attribute.attribute....attribute` สำหรับแบบจำลองข้อมูลที่ 1

Path2 = Class.attribute.attribute....attribute สำหรับแบบจำลองข้อมูลที่ 2

1. หากคลาสเริ่มต้นจาก Path1 และ Path2 ว่าเป็นคลาสใดในแบบจำลองข้อมูลรวม โดย A เป็นคลาสเริ่มต้นจาก Path1 และ B เป็นคลาสเริ่มต้นจาก Path2
2. ถ้า A ประกอบด้วยแอตทริบิวต์จากทั้ง Path1 และ Path2 และ Path1 ไม่มีการผ่านความสัมพันธ์แบบแอกกรีเกชันแล้ว จะทำการรวมแอตทริบิวต์สุดท้ายจาก Path1 เข้ากับแอตทริบิวต์สุดท้ายใน Path2 และลบแอตทริบิวต์สุดท้ายของ Path1 ออก
3. ถ้า A ประกอบด้วยแอตทริบิวต์จากทั้ง Path1 และ Path2 และ Path2 ไม่มีการผ่านความสัมพันธ์แบบแอกกรีเกชันแล้ว
  - ถ้า A เป็นคลาสเดียวกับ B ในแบบจำลองข้อมูลรวม จะทำการรวมแอตทริบิวต์สุดท้ายจาก Path2 กับแอตทริบิวต์สุดท้ายใน Path1 และลบแอตทริบิวต์สุดท้ายของ Path2 ออก
  - ถ้า B เป็นซูเปอร์คลาสของ A ในแบบจำลองข้อมูลรวม จะทำการรวมแอตทริบิวต์สุดท้ายจาก Path2 กับแอตทริบิวต์สุดท้ายใน Path1 และลบแอตทริบิวต์สุดท้ายของ Path2 ออกพร้อมทั้งย้าย Path1 ไปที่ B.
  - ถ้า A เป็นคลาสที่น้องกับ B ในแบบจำลองข้อมูลรวม จะทำการรวมแอตทริบิวต์สุดท้ายจาก Path2 กับแอตทริบิวต์สุดท้ายใน Path1 และลบแอตทริบิวต์สุดท้ายของ Path2 ออกพร้อมทั้งย้าย Path1 ไปที่ซูเปอร์คลาสของ A และ B
4. ถ้า B ประกอบด้วยแอตทริบิวต์จากทั้ง Path1 และ Path2 และ Path2 ไม่มีการผ่านความสัมพันธ์แบบแอกกรีเกชันแล้ว จะทำการรวมแอตทริบิวต์สุดท้ายจาก Path2 เข้ากับแอตทริบิวต์สุดท้ายใน Path1 และลบแอตทริบิวต์สุดท้ายของ Path2 ออก
5. ถ้า B ประกอบด้วยแอตทริบิวต์จากทั้ง Path1 และ Path2 และ Path1 ไม่มีการผ่านความสัมพันธ์แบบแอกกรีเกชันแล้ว
  - ถ้า A เป็นคลาสเดียวกับ B ในแบบจำลองข้อมูลรวม จะทำการรวมแอตทริบิวต์สุดท้ายจาก Path1 กับแอตทริบิวต์สุดท้ายใน Path2 และลบแอตทริบิวต์สุดท้ายของ Path1 ออก
  - ถ้า A เป็นซูเปอร์คลาสของ B ในแบบจำลองข้อมูลรวม จะทำการรวมแอตทริบิวต์สุดท้ายจาก Path1 กับแอตทริบิวต์สุดท้ายใน Path2 และลบแอตทริบิวต์สุดท้ายของ Path1 ออกพร้อมทั้งย้าย Path2 ไปที่ A
  - ถ้า A เป็นคลาสที่น้องกับ B ในแบบจำลองข้อมูลรวม จะทำการรวมแอตทริบิวต์สุดท้ายจาก Path1 กับแอตทริบิวต์สุดท้ายใน Path2 และลบแอตทริบิวต์สุดท้ายของ Path1 ออกพร้อมทั้งย้าย Path2 ไปที่ซูเปอร์คลาสของ A และ B

ขั้นตอนวิธีที่ใช้ในการรวม 2 แอตทริบิวต์ที่มีความหมายเหมือนกันดังกล่าวจะอาศัยหลักการพิจารณาเส้นทางในการเข้าถึงแอตทริบิวต์ทั้งสองจากการระบุของผู้ทำการรวม จากนั้นก็จะทำการรวมแอตทริบิวต์ปลายทางที่เหมือนกัน และสุดท้ายจะตัดเส้นทางที่เข้าช้อนนี้ออกไปเส้นทางหนึ่ง

รูปที่ 4.12 (ก) เป็นแผนภาพคลาสที่ได้จากการรวม Schema1 และ Schema2 ในรูปที่ 3.1 และรูปที่ 4.12 (ข) แสดงการรวมแอตทริบิวต์จากการระบุค่าเหมือน

Schema1.Student.faculty.name = Schema2.GradStudent.faculty\_name

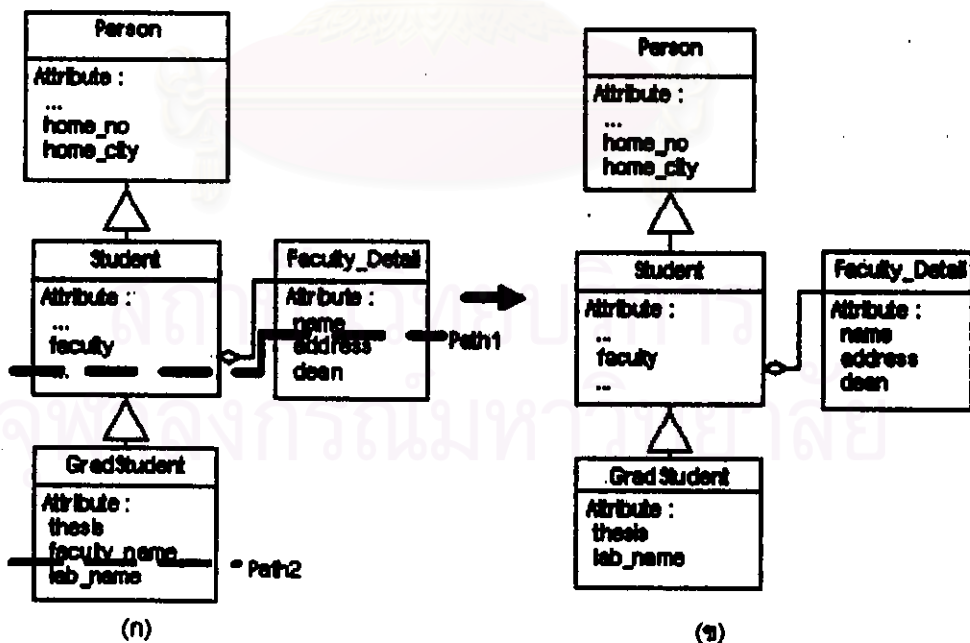
ซึ่งจะได้ว่า Path1 = Student.faculty.name

Path2 = GradStudent.faculty\_name

คลาสเริ่มต้นจาก Path1 ในแบบจำลองข้อมูลรวมคือ คลาส Student

คลาสเริ่มต้นจาก Path2 ในแบบจำลองข้อมูลรวมคือ คลาส GradStudent

จะได้ว่าแอตทริบิวต์ซึ่งแทนข้อมูลชื่อคณะในคลาส GradStudent สามารถเข้าถึงได้สองเส้นทางตาม Path1 และ Path2 ซึ่งมีการระบุค่าเหมือนกัน ดังนั้นจากกฎข้อ 4 ข้างต้น จะทำการรวมแอตทริบิวต์ faculty\_name (คลาส GradStudent) กับแอตทริบิวต์ name (คลาส Faculty\_Detail) จากนั้นก็จะลบแอตทริบิวต์ faculty\_name ในคลาส GradStudent ซึ่งผลที่ได้จะแสดงดังรูปที่ 4.12 (ข)



รูปที่ 4.12 ตัวอย่างการรวมแอตทริบิวต์ที่มีความหมายเหมือนกัน จากแบบจำลองข้อมูลในรูปที่ 3.1

ขั้นตอนวิธีที่ใช้ในการรวมแอดทรีวิฟท์ที่เป็นส่วนประกอบกันจะทำได้ในลักษณะเดียวกัน แต่จะเป็นการรวมส่วนประกอบทั้งหลายเข้ากับแอดทรีวิฟท์หลัก โดยมีวิธีดังนี้

ให้ Path1 และ Path2 เป็นเส้นทางในการเข้าถึงแอดทรีวิฟท์ที่เป็นส่วนประกอบกัน โดยสามารถผ่านความสัมพันธ์แบบแอกกรีเกชัน ตามที่ผู้ทำการรวมได้ระบุไว้

โดยที่ Path1 = Class.attribute.attribute....attribute สำหรับแบบจำลองข้อมูลที่ 1

Path2 = Class.attribute.attribute....attribute +

Class.attribute.attribute....attribute + ... สำหรับแบบจำลองข้อมูลที่ 2

1. หากคลาสเริ่มต้นจาก Path1 และ Path2 ว่าเป็นคลาสใดในแบบจำลองข้อมูลรวม โดย A เป็นคลาสเริ่มต้นจาก Path1 และ B เป็นคลาสเริ่มต้นจาก Path2
2. ถ้า A ประกอบด้วยแอดทรีวิฟท์จากทั้ง Path1 และ Path2 และ Path1 ไม่มีการผ่านความสัมพันธ์แบบแอกกรีเกชันแล้ว จะทำการรวมชุดแอดทรีวิฟท์สุดท้ายจาก Path2 เข้ากับแอดทรีวิฟท์สุดท้ายใน Path1 ในลักษณะการเป็นส่วนประกอบ และย้ายแอดทรีวิฟท์สุดท้ายใน Path1 ไปที่คลาสของแอดทรีวิฟท์สุดท้ายใน Path2 และลบแอดทรีวิฟท์สุดท้ายของ Path1 ออก
3. ถ้า A ประกอบด้วยแอดทรีวิฟท์จากทั้ง Path1 และ Path2 และ Path2 ไม่มีการผ่านความสัมพันธ์แบบแอกกรีเกชันแล้ว
  - ถ้า A เป็นคลาสเดียวกับ B ในแบบจำลองข้อมูลรวม จะทำการรวมชุดแอดทรีวิฟท์สุดท้ายจาก Path2 เข้ากับแอดทรีวิฟท์สุดท้ายใน Path1 ในลักษณะการเป็นส่วนประกอบและลบแอดทรีวิฟท์สุดท้ายของ Path2 ออก
  - ถ้า B เป็นซูเปอร์คลาสของ A ในแบบจำลองข้อมูลรวม จะทำการรวมชุดแอดทรีวิฟท์สุดท้ายจาก Path2 เข้ากับแอดทรีวิฟท์สุดท้ายใน Path1 ในลักษณะการเป็นส่วนประกอบและลบแอดทรีวิฟท์สุดท้ายของ Path2 ออกพร้อมทั้งย้าย Path1 ไปที่ B
  - ถ้า A เป็นคลาสที่น้องกับ B ในแบบจำลองข้อมูลรวม จะทำการรวมชุดแอดทรีวิฟท์สุดท้ายจาก Path2 เข้ากับแอดทรีวิฟท์สุดท้ายใน Path1 ในลักษณะการเป็นส่วนประกอบและลบแอดทรีวิฟท์สุดท้ายของ Path2 ออกพร้อมทั้งย้าย Path1 ไปที่ซูเปอร์คลาสของ A และ B
4. ถ้า B ประกอบด้วยแอดทรีวิฟท์จากทั้ง Path1 และ Path2 และ Path2 ไม่มีการผ่านความสัมพันธ์แบบแอกกรีเกชันแล้ว จะทำการรวมชุดแอดทรีวิฟท์สุดท้ายจาก Path2 เข้ากับแอดทรีวิฟท์สุดท้ายใน Path1 ในลักษณะการเป็นส่วนประกอบ และลบแอดทรีวิฟท์สุดท้ายของ Path2 ออก



5. ถ้า B ประกอบด้วยแอตทริบิวต์จากทั้ง Path1 และ Path2 และ Path1 ไม่มีการผ่านความสัมพันธ์แบบแอกกรีเกชันแล้ว
- ถ้า A เป็นคลาสเดียวกับ B ในแบบจำลองข้อมูลรวม จะทำการรวมชุดแอตทริบิวต์สุดท้ายจาก Path2 เข้ากับแอตทริบิวต์สุดท้ายใน Path1 ในลักษณะการเป็นส่วนประกอบและย้ายแอตทริบิวต์สุดท้ายจาก Path1 ไปที่คลาสของแอตทริบิวต์สุดท้ายของ Path2 และลบแอตทริบิวต์สุดท้ายของ Path1 ออก
  - ถ้า A เป็นซูเปอร์คลาสของ B ในแบบจำลองข้อมูลรวม จะทำการรวมชุดแอตทริบิวต์สุดท้ายจาก Path2 เข้ากับแอตทริบิวต์สุดท้ายใน Path1 ในลักษณะการเป็นส่วนประกอบ และย้ายแอตทริบิวต์สุดท้ายจาก Path1 ไปที่คลาสของแอตทริบิวต์สุดท้ายของ Path2 และลบแอตทริบิวต์สุดท้ายของ Path1 ออกพร้อมทั้งย้าย Path2 ไปที่ A
  - ถ้า A เป็นคลาสที่น้องกับ B ในแบบจำลองข้อมูลรวม จะทำการรวมชุดแอตทริบิวต์สุดท้ายจาก Path2 เข้ากับแอตทริบิวต์สุดท้ายใน Path1 ในลักษณะการเป็นส่วนประกอบและลบแอตทริบิวต์สุดท้ายของ Path1 ออกพร้อมทั้งย้าย Path2 ไปที่ซูเปอร์คลาสของ A และ B

รูปที่ 4.13 (ก) เป็นแผนภาพคลาสที่ได้จากการรวม Schema1 และ Schema2 ในรูปที่ 3.1 และรูปที่ 4.13 (ข) แสดงการรวมแอตทริบิวต์จากการระบุค่าที่เป็นส่วนประกอบกัน

Schema1.Student.address = Schema2.GradStudent.home\_no  
+ Schema2.GradStudent.home\_city

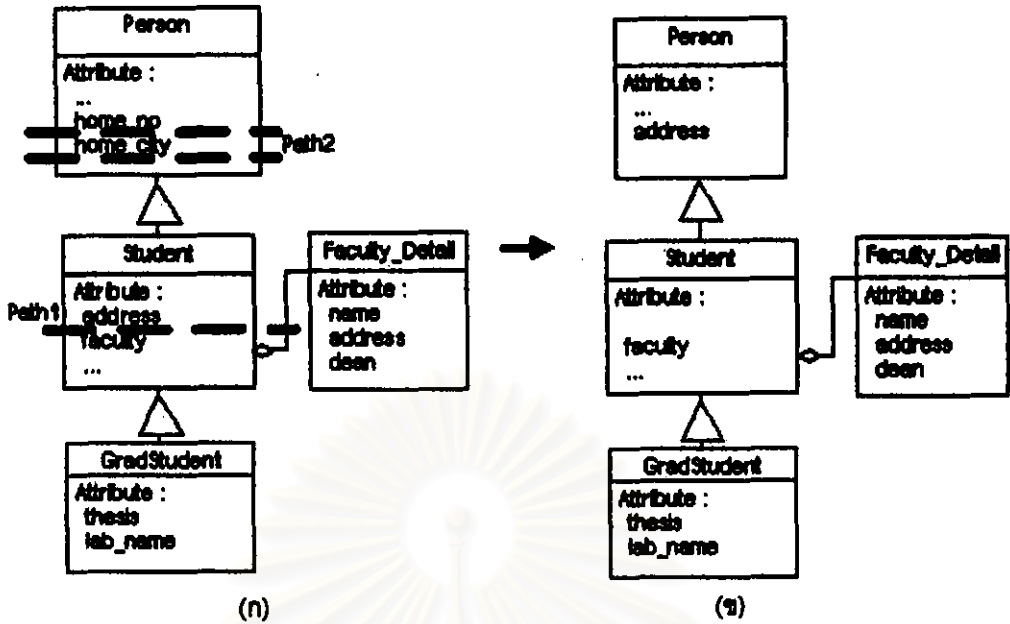
ซึ่งจะได้ว่า Path1 = Student.address

Path2 = GradStudent.home\_no + GradStudent.home\_city

คลาสเริ่มต้นจาก Path1 ในแบบจำลองข้อมูลรวมคือ คลาส Student

คลาสเริ่มต้นจาก Path2 ในแบบจำลองข้อมูลรวมคือ คลาส GradStudent

จะได้ว่าแอตทริบิวต์ซึ่งแทนที่อยู่ในคลาส Student สามารถเข้าถึงได้สองเส้นทางตาม Path1 และ Path2 ซึ่งมีการระบุว่าเป็นส่วนประกอบกัน ดังนั้นจากกฎข้อ 2 ข้างต้น จะทำการรวมแอตทริบิวต์ home\_no และแอตทริบิวต์ home\_city เข้ากับแอตทริบิวต์ address จากนั้นก็ย้ายแอตทริบิวต์ address มาไว้ในคลาส Person ซึ่งผลที่ได้จะแสดงดังรูปที่ 4.13 (ข)



รูปที่ 4.13 ตัวอย่างการรวมแอตทริบิวต์ที่เป็นส่วนประกอบกัน จากแบบจำลองข้อมูลในรูปที่ 3.1

เมื่อทำการรวมค่าที่มีความหมายเหมือนกันและค่าที่เป็นส่วนประกอบกันสำหรับแอตทริบิวต์ ตามที่ผู้ทำการรวมได้ระบุไว้แล้ว ในที่สุดจะได้เป็นแบบจำลองข้อมูลรวมที่สมบูรณ์

4.4 ตัวอย่างการรวมแบบจำลองข้อมูล

ในส่วนนี้จะเป็นการรวมแบบจำลองข้อมูล Schema1 และ Schema2 ในรูปที่ 3.1 ซึ่งจากการเปรียบเทียบแบบจำลองทั้งสองในบทที่ 4 จะได้ความสัมพันธ์ของคลาสต่างๆ ดังตารางที่ 4.4

ตารางที่ 4.4 ความสัมพันธ์จากการเปรียบเทียบคลาสในรูปที่ 3.1

	Schema2	Person	GradStudent
Schema1			
	Student	Subclass	Superclass
	Faculty_Detail	Disjoint	Disjoint

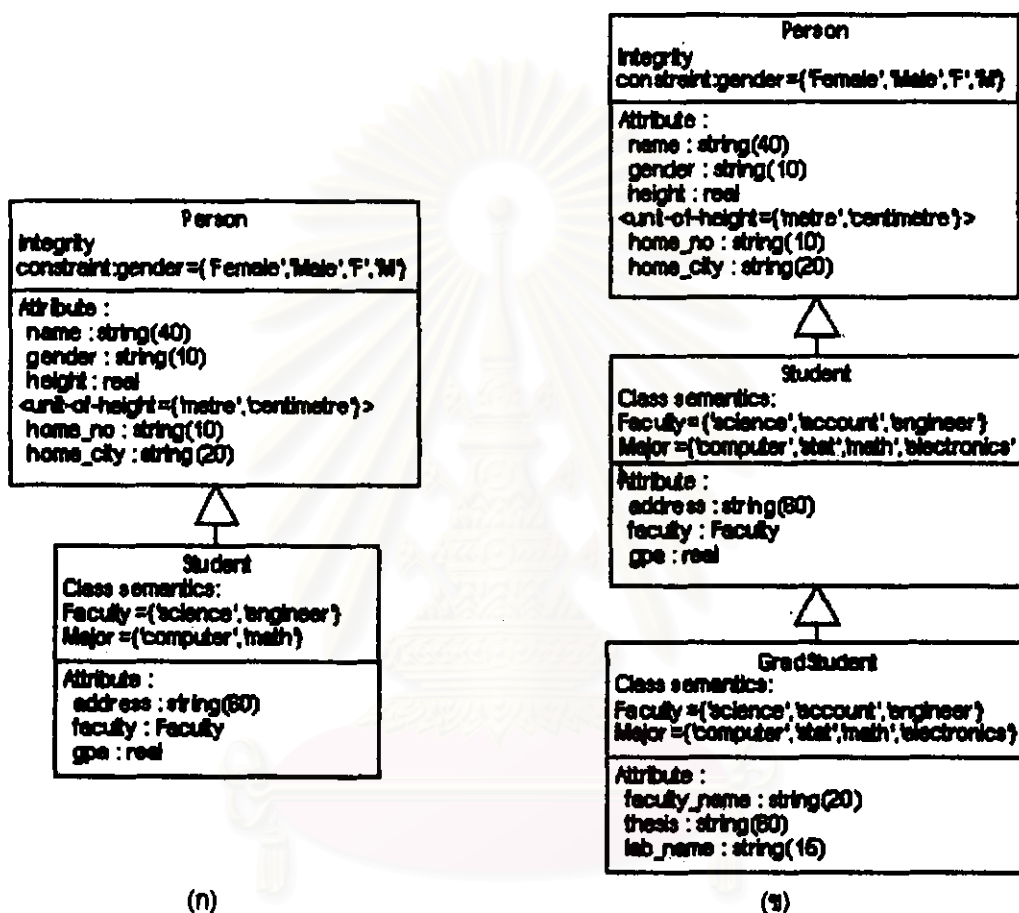
ขั้นตอนที่ใช้ในการรวมเป็นแบบจำลองข้อมูลรวมมีดังนี้

ขั้นที่ 1 การรวมคลาสโดยการพิจารณาความสัมพันธ์

จากลำดับความสัมพันธ์ของคลาสจะทำการรวมดังนี้

- รวมคลาส Student (Schema1) และคลาส Person (Schema2) ที่มีความสัมพันธ์แบบสับคลาส ดังรูปที่ 4.14 (ก)

- รวมคลาส Student (Schema1) และคลาส GradStudent (Schema2) ที่มีความสัมพันธ์แบบยูเอชคลาส ดังรูปที่ 4.14 (ข)
- ผลที่ได้จากการรวมในขั้นนี้ คือแบบจำลองข้อมูลรวมในรูปที่ 4.14 (ง) พร้อมทั้งคลาส Faculty\_Detail ของ Schema1 ที่ไม่มีการรวมกับคลาสใดๆ ใน Schema2 ด้วย

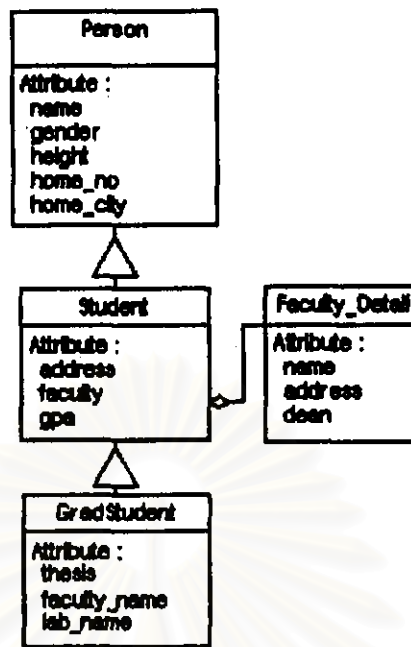


รูปที่ 4.14 (ก) การรวมคลาส Student และคลาส Person ที่มีความสัมพันธ์แบบสับคลาส

(ข) การรวมคลาส Student และคลาส GradStudent ที่มีความสัมพันธ์แบบยูเอชคลาส

## ขั้นที่ 2 การเชื่อมความสัมพันธ์แบบเอกกริเทรชันกับคลาสรวมที่ได้จากขั้นที่ 1

ในขั้นนี้จะพิจารณาว่าคลาสที่มีความสัมพันธ์แบบเอกกริเทรชันในแบบจำลองข้อมูลเดิมได้กลายเป็นคลาสใดในแบบจำลองข้อมูลรวม ซึ่งจะทำให้มีการเชื่อมความสัมพันธ์ได้ถูกต้อง จากตัวอย่างนี้จะได้ว่าคลาส Student มีความสัมพันธ์แบบเอกกริเทรชันกับคลาส Faculty\_Detail โดยผ่านความสัมพันธ์กับแอตทริบิวต์ faculty ใน Schema1 ดังนั้นจะทำการเชื่อมความสัมพันธ์แบบเอกกริเทรชันซึ่งจะได้ดังรูปที่ 4.15



รูปที่ 4.15 แบบจำลองข้อมูลรวมโดยแผนภาพคลาสที่ได้  
จากขั้นตอนการเชื่อมความสัมพันธ์แบบเอกกรีเกชัน

### ขั้นที่ 3 การรวมค่าที่มีความหมายเหมือนกันหรือค่าที่เป็นส่วนประกอบกันสำหรับแอตทริบิวต์

การระบุค่าที่มีความหมายเหมือนกันและการระบุค่าที่เป็นส่วนประกอบกันสำหรับแอตทริบิวต์ สำหรับตัวอย่างนี้คือ

ค่าที่มีความหมายเหมือนกัน :

Schema1.Student.sex = Schema2.GradStudent.gender

Schema1.Student.faculty.name = Schema2.GradStudent.faculty\_name

ค่าที่เป็นส่วนประกอบกัน :

Schema1.Student.address = Schema2.GradStudent.home\_no

+ Schema2.GradStudent.home\_city

จากนั้นจะทำการรวมค่าที่มีความหมายเหมือนกันและค่าที่เป็นส่วนประกอบกันตามขั้นตอนในหัวข้อ 4.3.3 ซึ่งในที่สุดจะได้แบบจำลองข้อมูลรวมที่สมบูรณ์โดยมีรายละเอียดของการรวมดังแสดงในรูป 4.16

```

[Class] : Person (Source - Schema:schema2/Class:Person)
<Class Attribute>
  [Attribute] : name
    DataType : string Size : 40 Set : false
  <Map Attribute From> :
    Attribute :name Class :Student Schema :schema1
      DataType :String Size :40 Set : false
    Attribute :name Class :Person Schema :schema2
      DataType :String Size :40 Set : false
  [Attribute] : gender
    DataType : string Size : 10 Set : false
  <Attribute Constraint> Possible Value:F,Female,M,Male
  MapValueFunction : Required
  <Map Attribute From> :
    Attribute :sex Class :Student Schema :schema1
      DataType :String Size :1 Set : false
      Possible Value : F,M
    Attribute :gender Class :Person Schema :schema2
      DataType :String Size :10 Set : false
      Possible Value : Female, Male
  [Attribute] : height
    DataType : double Size : 0 Set : false
  <Attribute Semantic> :
    Semantic : unit-of-height Value : centimetre, metre
    Map function : true
    For Semantic :
      - unit-of-height = centimetre(Schema:schema1 Class:Student Attribute:height)
      unit-of-height = metre(Schema:schema2 Class:Person Attribute:height)
  <Map Attribute From> :
    Attribute :height Class :Student Schema :schema1
      DataType :Real Size :0 Set : false
    Attribute :height Class :Person Schema :schema2
      DataType :Real Size :0 Set : false

```

**รูปที่ 4.16** รายละเอียดแบบจำลองข้อมูลรวมที่ได้จากการรวมแบบจำลองข้อมูลในชั้นการรวมค่าที่มีความหมายเหมือนกันและค่าที่เป็นส่วนประกอบกันสำหรับแอตทริบิวต์ (มีต่อ)

```

(Attribute) : address
  DataType : string  Size : 60  Set : false
  <Map Attribute From> :
    Attribute :address  Class :Student  Schema :schema1
    DataType :String  Size :60  Set : false
  Compound Attributes :
    home_no  Class :Person  Schema :schema2
    DataType :String  Size :10  Set : false
    home_city  Class :Person  Schema :schema2
    DataType :String  Size :20  Set : false
[Class] : Student (Source - Schema:schema1/Class:Student)
Parent : Person
<Class Semantic>
  Semantic : Faculty  Value : account,engineer,science
  Map function : false
  Semantic : Major  Value : computer,electronics,math,stat
  Map function : false
<Class Attribute>
  (Attribute) : gpa
  DataType : double  Size : 0  Set : false
  <Map Attribute From> :
    Attribute :gpa  Class :Student  Schema :schema1
    DataType :Real  Size :0  Set : false
    Attribute :gpa  Class :GradStudent  Schema :schema2
    DataType :Real  Size :0  Set : false
  (Attribute) : faculty
  DataType : Faculty_Detail  Set : false
  <Map Attribute From> :
    Attribute :faculty  Class :Student  Schema :schema1
    DataType :Faculty_Detail  Set : false
[Class] : GradStudent (Source - Schema:schema2/Class:GradStudent)
Parent : Student
<Class Semantic>
  Semantic : Faculty  Value : science,account,engineer
  Map function : false
  Semantic : Major  Value : computer,stat,math,electronics
  Map function : false

```

รูปที่ 4.16 (ต่อ) รายละเอียดของแบบจำลองข้อมูลรวมที่ได้จากการรวมแบบจำลองข้อมูลใน  
ชั้นการรวมค่าที่มีความหมายเหมือนกันและค่าที่เป็นส่วนประกอบกันสำหรับแอตทริบิวท์ (มีต่อ)



```

<Class Attribute>
  [Attribute]: thesis
    DataType: string Size: 60 Set: false
  <Map Attribute From>:
    Attribute: thesis Class: GradStudent Schema: schema2
    DataType: String Size: 60 Set: false
  [Attribute]: lab_name
    DataType: string Size: 15 Set: false
  <Map Attribute From>:
    Attribute: lab_name Class: GradStudent Schema: schema2
    DataType: String Size: 15 Set: false
[Class] : Faculty_Detail (Source - Schema: schema1/Class: Faculty_Detail)
<Class Attribute>
  [Attribute]: name
    DataType: string Size: 20 Set: false
  <Map Attribute From>:
    Attribute: name Class: Faculty_Detail Schema: schema1
    DataType: String Size: 20 Set: false
    Attribute: faculty_name Class: GradStudent Schema: schema2
    DataType: String Size: 20 Set: false
  [Attribute]: address
    DataType: string Size: 25 Set: false
  <Map Attribute From>:
    Attribute: address Class: Faculty_Detail Schema: schema1
    DataType: String Size: 25 Set: false
  [Attribute]: dean
    DataType: string Size: 40 Set: false
  <Map Attribute From>:
    Attribute: dean Class: Faculty_Detail Schema: schema1
    DataType: String Size: 40 Set: false

```

รูปที่ 4.16 (ต่อ) รายละเอียดของแบบจำลองข้อมูลรวมที่ได้จากการรวมแบบจำลองข้อมูลใน  
ขั้นการรวมค่าที่มีความหมายเหมือนกันและค่าที่เป็นส่วนประกอบกันสำหรับแอตทริบิวต์

จากวิธีการเปรียบเทียบในบทที่ 3 และวิธีการรวมตามที่ได้กล่าวถึงในบทนี้ จะได้แบบ  
จำลองข้อมูลรวม แต่จะยังไม่สามารถสรุปได้ว่าแบบจำลองข้อมูลรวมที่ได้จะถูกดึงตามหลักการ  
ออกแบบแบบจำลองข้อมูลเชิงวัตถุที่ดีหรือไม่ ดังนั้น ในการทดสอบจะนำแบบจำลองข้อมูลรวมไป  
เปรียบเทียบกับผลการรวมจากงานวิจัยอื่น นอกจากนี้จะนำแบบจำลองข้อมูลรวมที่ได้ไปทดลอง  
สร้างบนฐานข้อมูลจริง เพื่อดูว่าการเข้าถึงข้อมูลจากแบบจำลองข้อมูลรวมจะยังให้ข้อมูลครบถ้วน  
เหมือนกับการเข้าถึงข้อมูลจากแบบจำลองข้อมูลเดิมทั้งสองที่ละชั้นหรือไม่ ซึ่งรายละเอียดในเรื่อง  
ของการทดสอบผลนี้จะกล่าวถึงในบทต่อไป