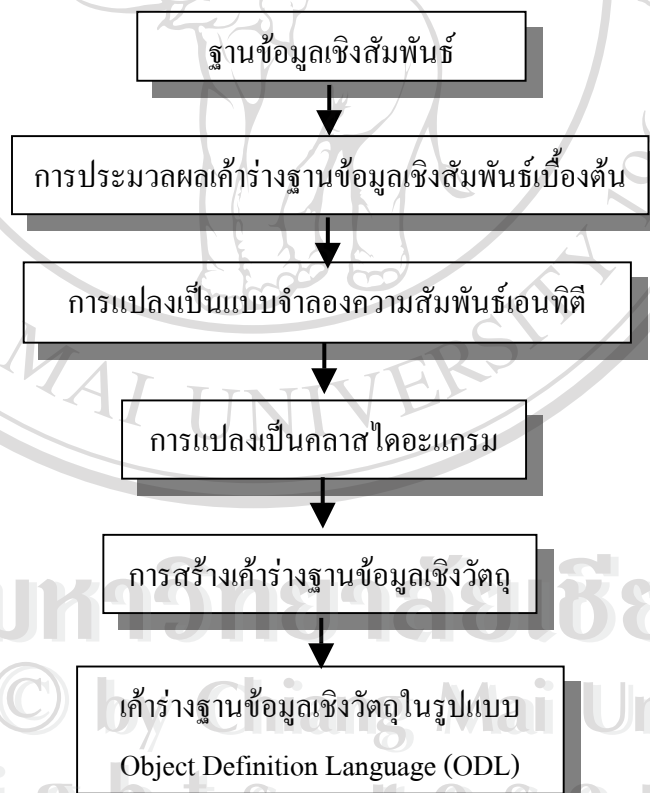


บทที่ 3

การออกแบบระบบ

หลังจากศึกษาเอกสารและงานวิจัยที่เกี่ยวข้องกับการแปลงเค้าร่างฐานข้อมูลเชิงสัมพันธ์เป็นเค้าร่างฐานข้อมูลเชิงวัตถุ เพื่อเป็นพื้นฐานและแนวทางสำหรับดำเนินงานแล้ว ขั้นตอนต่อมาคือการออกแบบและการพัฒนาเครื่องมือ โดยใช้โปรแกรมไมโครซอฟท์วิซวลเบสิกในส่วนของการสร้างเครื่องมือและการประมวลผล และใช้ฐานข้อมูลเชิงสัมพันธ์เป็นข้อมูลนำเข้า มีขั้นตอนการดำเนินการในระบบ ดังรูป 3.1



รูป 3.1 แสดงขั้นตอนการดำเนินการ

ในส่วนแรกของการดำเนินการจะเป็นการนำเข้าฐานข้อมูลเชิงสัมพันธ์ ซึ่งส่วนนำเข้าข้อมูล ผู้ใช้งานสามารถแบ่งส่วนนำเข้าข้อมูลเป็น ดังนี้

แบบที่ 1 ผู้ใช้งานสามารถกำหนดและบันทึกรายละเอียดของเค้าร่างฐานข้อมูลเชิงสัมพันธ์ได้โดยตรง

แบบที่ 2 ผู้ใช้งานสามารถเลือกฐานข้อมูลที่ต้องการจะแปลง

แบบที่ 3 ผู้ใช้งานสามารถเลือกสคริปต์ไฟล์ (Script file) .sql ของ MS SQL Server

แบบที่ 4 ผู้ใช้งานสามารถเลือกเท็กซ์ไฟล์ (Text file) .txt

แบบที่ 5 ผู้ใช้งานสามารถเลือกเอสดีเอฟไฟล์ (SDF file) .sdf

เค้าร่างฐานข้อมูลเชิงสัมพันธ์ที่ได้จากส่วนนำเข้าข้อมูลต่าง ๆ จะถูกนำไปประมวลผลในขั้นตอนอื่น ๆ ของการดำเนินการต่อไป

3.1 การประมวลผลเค้าร่างฐานข้อมูลเชิงสัมพันธ์เบื้องต้น

เป็นการนำข้อมูลเข้ามาพิจารณาและปรับเปลี่ยนเป็นเค้าร่างฐานข้อมูลเชิงสัมพันธ์เบื้องต้น ซึ่งในส่วนนี้มีขั้นตอนย่อยดังแสดงในรูป 3.2



รูป 3.2 แสดงขั้นตอนย่อยของการประมวลผลเค้าร่างฐานข้อมูลเชิงสัมพันธ์เบื้องต้น

3.1.1 การหาเค้าร่างฐานข้อมูลเชิงสัมพันธ์

หลังจากที่รับฐานข้อมูลเชิงสัมพันธ์จากส่วนนำเข้าข้อมูลแล้ว ฐานข้อมูลเชิงสัมพันธ์จะถูกนำมาประมวลผลเบื้องต้น เพื่อหารีเลชัน แอททริบิวต์ คีย์หลัก คีย์นอกและความสัมพันธ์ระหว่างรีเลชัน โดยการประมวลผลดังกล่าวจะแตกต่างกันไปตามลักษณะของวิธีการนำเข้าข้อมูล ดังนี้

(1) ผู้ใช้งานกำหนดรายละเอียดของเค้าร่างฐานข้อมูลเชิงสัมพันธ์ การหาเค้าร่างฐานข้อมูลเชิงสัมพันธ์ทำได้โดยตรงจากการกรอกริเลชัน รายละเอียดของแต่ละแอททริบิวต์ คีย์หลัก คีย์นอก และความสัมพันธ์ระหว่างริเลชัน ของผู้ใช้งาน

(2) ผู้ใช้งานเลือกฐานข้อมูลที่ต้องการจะแปลง การหาเค้าร่างฐานข้อมูลเชิงสัมพันธ์ทำได้ดังนี้

- ใช้ Microsoft ADO Ext. 2.6 For DDL and Security (ADOX) กับฐานข้อมูลแบบ Access ซึ่ง ADOX เป็น Dynamic Link Library (DLL) สำหรับนิยามและจัดการโครงสร้างฐานข้อมูลเชิงสัมพันธ์แบบ Access

- ใช้ Microsoft SQLDMO Object Library กับฐานข้อมูลแบบ MS SQL Server ซึ่ง SQLDMO เป็น Object Library สำหรับนิยามและจัดการโครงสร้างฐานข้อมูลเชิงสัมพันธ์แบบ MS SQL Server

(3) ผู้ใช้งานเลือกสคริปต์ไฟล์ของ MS SQL Server สคริปต์ไฟล์จะถูกผ่านการตรวจสอบและค้นหาเค้าร่างฐานข้อมูลเชิงสัมพันธ์ด้วยคำสำคัญหลัก (Keyword) ดังนี้

- ใช้คำสำคัญหลัก “CREATE TABLE” เพื่อหาริเลชันและแอททริบิวต์

```
CREATE TABLE <table name> (
    <column1> <datatype> <size> <nullable>,
    <column2> <datatype> <size> <nullable>,
    .....
```

)

- ใช้คำสำคัญหลัก “ALTER TABLE... PRIMARY KEY” เพื่อหาคีย์หลัก

```
ALTER TABLE <table name> ...
... PRIMARY KEY ...
```

```
(
    <column name>
```

)

- ใช้คำสั่งหลัก “ALTER TABLE... FOREIGN KEY” เพื่อหาคีย์นอก

ALTER TABLE <table name> ...

... FOREIGN KEY ...

(<column name>

)

- ใช้คำสั่งหลัก “FOREIGN KEY...REFERENCES” เพื่อหาความสัมพันธ์

ระหว่างรีเลชัน

... < table name > FOREIGN KEY

(

<column name>

) REFERENCES <referencedtable name>

(

< referencedcolumn name>

)

(4) ผู้ใช้สามารถเลือกเท็กซ์ไฟล์ เท็กซ์ไฟล์จะถูกผ่านการตรวจสอบและค้นหาเค้าร่างฐานข้อมูลเชิงสัมพันธ์ ตามรูปแบบที่กำหนดไว้¹

(5) ผู้ใช้งานเลือกเอสดีเอฟไฟล์ ซึ่งเอสดีเอฟไฟล์จะถูกผ่านการตรวจสอบและค้นหาเค้าร่างฐานข้อมูลเชิงสัมพันธ์ตามรูปแบบที่กำหนดไว้²

3.1.2 การสร้างแผนภาพรีเลชัน

หลังจากการประมวลผลเบื้องต้นเสร็จเรียบร้อยแล้ว รีเลชัน แอททริบิวต์และความสัมพันธ์ระหว่างรีเลชัน จะถูกนำมาสร้างเป็นแผนภาพรีเลชันอย่างอัตโนมัติ

¹ รูปแบบของเท็กซ์ไฟล์ หน้า 67

² รูปแบบของเอสดีเอฟไฟล์ หน้า 68 -70

3.2 การแปลงเป็นแบบจำลองความสัมพันธ์เอนทิตี

การแปลงเค้าร่างฐานข้อมูลเชิงสัมพันธ์ให้เป็นแบบจำลองความสัมพันธ์เอนทิตี มีขั้นตอนย่อยดังแสดงในรูป 3.3



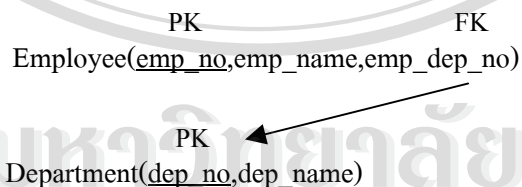
รูป 3.3 แสดงขั้นตอนย่อยของการแปลงเป็นแบบจำลองความสัมพันธ์เอนทิตี

3.2.1 การออกแบบประเภตรีเลชัน

ในขั้นตอนย่อยนี้ รีเลชันต่าง ๆ ของเค้าร่างฐานข้อมูลเชิงสัมพันธ์จะถูกจัดประเภทของรีเลชัน เพื่อนำไปใช้ในการประมวลผลสำหรับขั้นตอนการแปลงเป็นแบบจำลองความสัมพันธ์เอนทิตี โดยได้นิยามกลุ่มรีเลชัน ออกเป็นกลุ่มต่าง ๆ ดังนี้

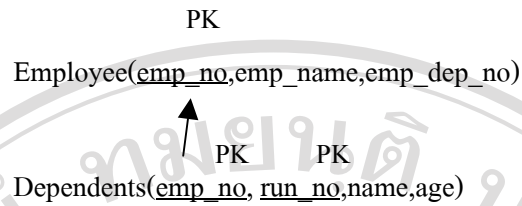
(1) กลุ่มของ Primary relation คือกลุ่มรีเลชันที่คีย์หลัก ไม่มีคีย์ของรีเลชันอื่นประกอบ

เช่น



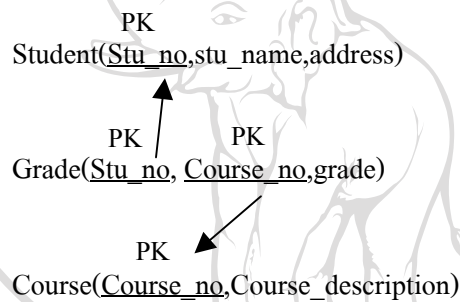
คีย์หลักของรีเลชัน Employee และรีเลชัน Department ต่างก็ไม่มีคีย์ของรีเลชันอื่นประกอบ ดังนั้นรีเลชัน Employee และรีเลชัน Department จึงจัดเป็น Primary relation

(2) กลุ่มของ Weak relation คือกลุ่มรีเลชันที่คีย์หลัก เป็นคีย์ของรีเลชันอื่นและมี partial key ที่ไม่ unique สำหรับ Weak relation นั้น เช่น



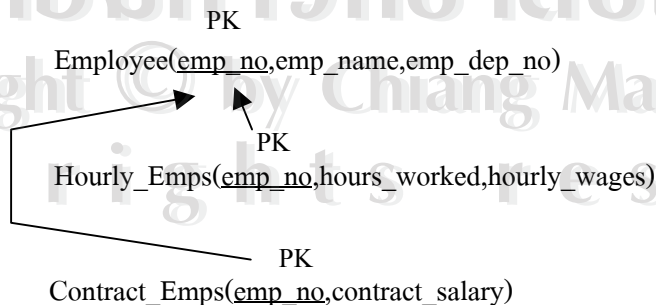
คีย์หลักของรีเลชัน Dependents มีคีย์ emp_no ของรีเลชัน Employee ประกอบ และ run_no ไม่ unique สำหรับรีเลชัน Dependents ดังนั้นรีเลชัน Dependents จึงจัดเป็น Weak relation

(3) กลุ่มของ Secondary relation คือกลุ่มรีเลชันที่แอททริบิวต์ที่ประกอบกันเป็นคีย์หลักทุกตัว มีคีย์ของรีเลชันอื่นประกอบ เช่น



แอททริบิวต์ที่ประกอบเป็นคีย์หลักทุกตัวของรีเลชัน Grade มีคีย์ Stu_no ของรีเลชัน Student และคีย์ Course_no ของรีเลชัน Course ประกอบ ดังนั้นรีเลชัน Grade จึงจัดเป็น Secondary relation

(4) กลุ่มของ ISA relation คือกลุ่มรีเลชันที่แอททริบิวต์ที่ประกอบกันเป็นคีย์หลักทุกตัว มีคีย์ของรีเลชันใดรีเลชันหนึ่งเท่านั้น เช่น



แอททริบิวต์ที่ประกอบเป็นคีย์หลักทุกตัวของรีเลชัน Hourly_Emps และรีเลชัน Contract_Emps มีคีย์ emp_no ของรีเลชัน Employee เป็นคีย์หลัก ดังนั้น Hourly_Emps และ รีเลชัน Contract_Emps จึงจัดเป็น ISA relation

3.2.2 การแปลงเป็นแบบจำลองความสัมพันธ์เอนทิตี

เมื่อรีเลชันต่าง ๆ ที่ถูกจัดประเภทแล้ว จะถูกนำไปสู่กระบวนการแปลงเป็นแบบจำลองความสัมพันธ์เอนทิตี ดังนี้

- (1) กลุ่ม Primary relation จะถูกแปลงเป็นเอนทิตี
- (2) กลุ่ม Weak relation จะถูกแปลงเป็นเอนทิตีแบบอ่อน
- (3) กลุ่ม Secondary relation จะถูกแปลงเป็นความสัมพันธ์ระหว่างเอนทิตีและกำหนดสัดส่วนความสัมพันธ์ระหว่างเอนทิตีเป็น M:N
- (4) กลุ่ม ISA relation จะถูกแปลงเป็นเอนทิตีในแบบเจนอรัลไลเซชัน³
- (5) คีย์นอกที่เกิดขึ้นในแอทริบิวต์ จะถูกแปลงเป็นความสัมพันธ์ระหว่างเอนทิตีและกำหนดสัดส่วนความสัมพันธ์ระหว่างเอนทิตีเป็น 1:N หรือ M:N
- (6) คีย์นอกใด ๆ ได้ทำการอ้างอิงคีย์หลักภายในเอนทิตีเดียวกัน แสดงว่าความสัมพันธ์ของเอนทิตีนั้นเป็นแบบวนซ้ำและกำหนดสัดส่วนความสัมพันธ์ภายในเอนทิตีเป็น 1:N⁴

3.2.3 การสร้างแผนภาพความสัมพันธ์เอนทิตี

เมื่อรีเลชันต่าง ๆ ผ่านกระบวนการแปลงเป็นแบบจำลองความสัมพันธ์เอนทิตีเรียบร้อยแล้ว ระบบจะทำการสร้างแผนภาพความสัมพันธ์เอนทิตี ซึ่งประกอบไปด้วยเอนทิตี แอททริบิวต์ คีย์แอททริบิวต์ ความสัมพันธ์และสัดส่วนความสัมพันธ์ระหว่างเอนทิตีต่าง ๆ อย่างอัตโนมัติ

³ การใส่ข้อบังคับ Disjoint หรือ Overlap ระหว่างเอนทิตีหลักและเอนทิตีย่อย ผู้ใช้ต้องเป็นผู้กำหนดเอง โดยระบบมีตัวเลือกให้ผู้เลือกใช้ได้

⁴ M:N ผู้ใช้ต้องเป็นผู้กำหนดเอง โดยระบบมีตัวเลือกให้ผู้เลือกใช้ได้

3.3 การแปลงเป็นคลาสไดอะแกรม

การแปลงแบบจำลองความสัมพันธ์เอนทิตีให้เป็นคลาสไดอะแกรม มีขั้นตอนย่อยดังแสดงในรูป 3.4



รูป 3.4 แสดงขั้นตอนย่อยของการแปลงเป็นคลาสไดอะแกรม

3.3.1 การออกแบบประเภทเอนทิตี

ในขั้นตอนย่อยนี้ เอนทิตีและความสัมพันธ์ระหว่างเอนทิตีจะนำมาจัดประเภท เพื่อนำไปใช้ในการประมวลผลสำหรับขั้นตอนการแปลงเป็นคลาสไดอะแกรม โดยได้นิยามกลุ่มเอนทิตีออกเป็นกลุ่มต่าง ๆ ดังนี้

- (1) กลุ่มของ Primary Entity คือ เอนทิตีแบบธรรมดา
- (2) กลุ่มของ Weak Entity คือ เอนทิตีแบบอ่อน
- (3) กลุ่มของ Relationship Entity คือ กลุ่มเอนทิตีที่แอททริบิวต์ที่ประกอบกันเป็นคีย์หลักทุกตัว มีคีย์หลักของเอนทิตีอื่นประกอบและมีแอททริบิวต์ที่ไม่ใช่คีย์ใด ๆ
- (4) กลุ่มของ Relationship-type1 คือ กลุ่มเอนทิตีที่แอททริบิวต์ที่ประกอบกันเป็นคีย์หลักทุกตัว มีคีย์หลักของเอนทิตีอื่นประกอบและไม่มีแอททริบิวต์ที่ไม่ใช่คีย์ใด ๆ
- (5) กลุ่มของ Relationship-type2 คือกลุ่มความสัมพันธ์ระหว่างเอนทิตีที่เกิดขึ้นระหว่างเอนทิตีแบบธรรมดากับเอนทิตีแบบอ่อน
- (6) กลุ่มของ Relationship-type3 คือกลุ่มความสัมพันธ์ระหว่างเอนทิตีที่เกิดขึ้นในแบบจำลองความสัมพันธ์เอนทิตี นอกเหนือจากข้อ (5) และ ข้อ (6)
- (7) กลุ่มของ ISA Entity คือกลุ่มเอนทิตีที่แอททริบิวต์ที่ประกอบกันเป็นคีย์หลักทุกตัว มีคีย์ของเอนทิตีใดเอนทิตีหนึ่งประกอบเท่านั้น

3.3.2 การแปลงเป็นคลาสไดอะแกรม

เมื่อเอนทิตีและความสัมพันธ์ระหว่างเอนทิตีต่าง ๆ ถูกจัดประเภทแล้ว จะถูกนำไปสู่กระบวนการแปลงเป็นคลาสไดอะแกรม ดังนี้

- (1) กลุ่มของ Primary Entity จะถูกแปลงเป็นคลาส ซึ่งประกอบไปด้วยแอททริบิวต์และเมธอดที่มีชื่อเดียวกันกับชื่อคลาส
- (2) กลุ่มของ Weak Entity จะถูกแปลงเป็นคลาส ซึ่งประกอบไปด้วยแอททริบิวต์และเมธอดที่มีชื่อเดียวกันกับชื่อคลาส
- (3) กลุ่มของ Relationship Entity จะถูกแปลงเป็น แอสโซซิเอชัน คลาส
- (4) กลุ่มของ Relationship-type1 จะถูกแปลงเป็นแอสโซซิเอชันระหว่างคลาส โดยกำหนดปริมาณของคลาสที่สัมพันธ์กัน คือ
 - แปลง M:N โดย M เป็น (1..*) และ N เป็น (1..*)
- (5) กลุ่มของ Relationship-type2 จะถูกแปลงเป็นควอลิไฟ แอสโซซิเอชัน ระหว่างคลาส โดยกำหนดปริมาณของคลาสที่สัมพันธ์กัน คือ
 - แปลง 1:N โดย 1 เป็น (1..1) และ N เป็น (1..*)
- (6) กลุ่มของ Relationship-type3 จะถูกแปลงเป็นแอสโซซิเอชันระหว่างคลาส โดยกำหนดปริมาณของคลาสที่สัมพันธ์กัน คือ
 - แปลง 1:N โดย 1 เป็น (1..1) และ N เป็น (1..*)
 - แปลง M:N โดย M เป็น (1..*) และ N เป็น (1..*)
- (7) กลุ่มของ ISA Entity จะถูกแปลงเป็นซับคลาส โดยเกิดความสัมพันธ์ระหว่างซูเปอร์คลาสและซับคลาสเป็น “Is-a”
- (8) ชื่อความสัมพันธ์ระหว่างเอนทิตี จะถูกแปลงเป็นชื่อของแอสโซซิเอชัน

3.3.3 การสร้างคลาสไดอะแกรม

เมื่อเอนทิตีต่างๆ ผ่านกระบวนการแปลงเป็นคลาสไดอะแกรมเรียบร้อยแล้ว ระบบจะทำการสร้างคลาสไดอะแกรม ซึ่งประกอบไปด้วยคลาส แอททริบิวต์ เมธอด ความสัมพันธ์และปริมาณของความสัมพัทธ์ระหว่างคลาสต่างๆ อย่างอัตโนมัติ

3.4 การสร้างเค้าร่างฐานข้อมูลเชิงวัตถุในรูปแบบภาษานิยามโครงสร้างออบเจกต์

3.4.1 โครงสร้างโดยรวมของภาษานิยามโครงสร้างออบเจกต์

```

Class <ClassName> extends <SuperclassName>
( extent <ExtentName>
  keys <KeyName>
)
{
  attribute <attributeType> <attributeName>;
  ...
  relationship <relationType> <relationName> inverse <inverseSpec>;
  ...
  method_definitions;
  ...
}

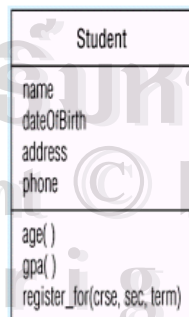
```

3.4.2 ขั้นตอนการสร้างภาษานิยามโครงสร้างออบเจกต์

(1) คลาส สร้าง ODL Class ซึ่งประกอบไปด้วย เซตออบเจกต์ของคลาส คีย์หลัก (มีหรือไม่มีก็ได้) แอททริบิวต์และเมธอด ดังรูป 3.5

UML

ODL



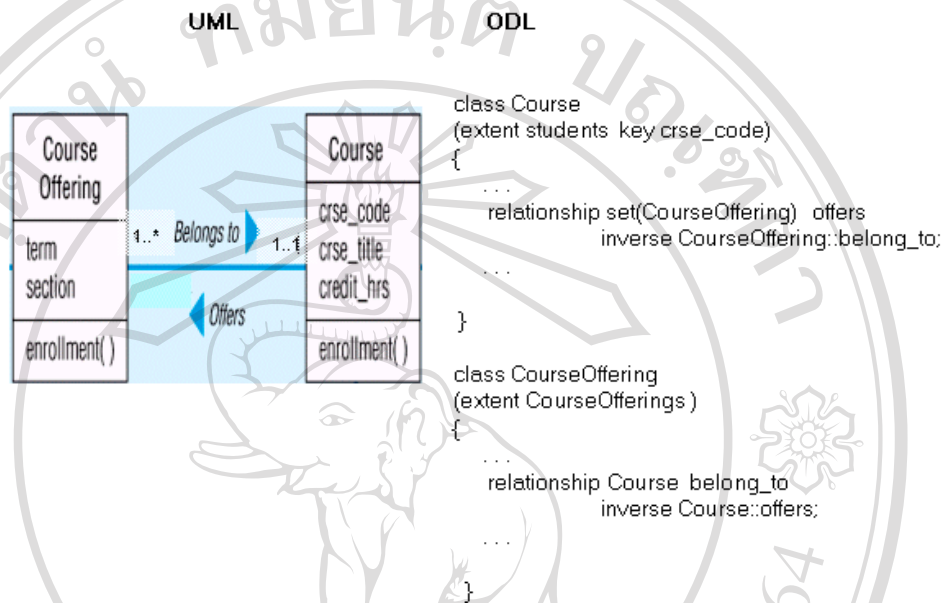
```

class Student
(extent students key name)
{
  attribute string name;
  attribute date dateOfBirth;
  attribute string address;
  attribute string phone;
  age()
  gpa()
  boolean register_for(string crse, short sec, string term)
}

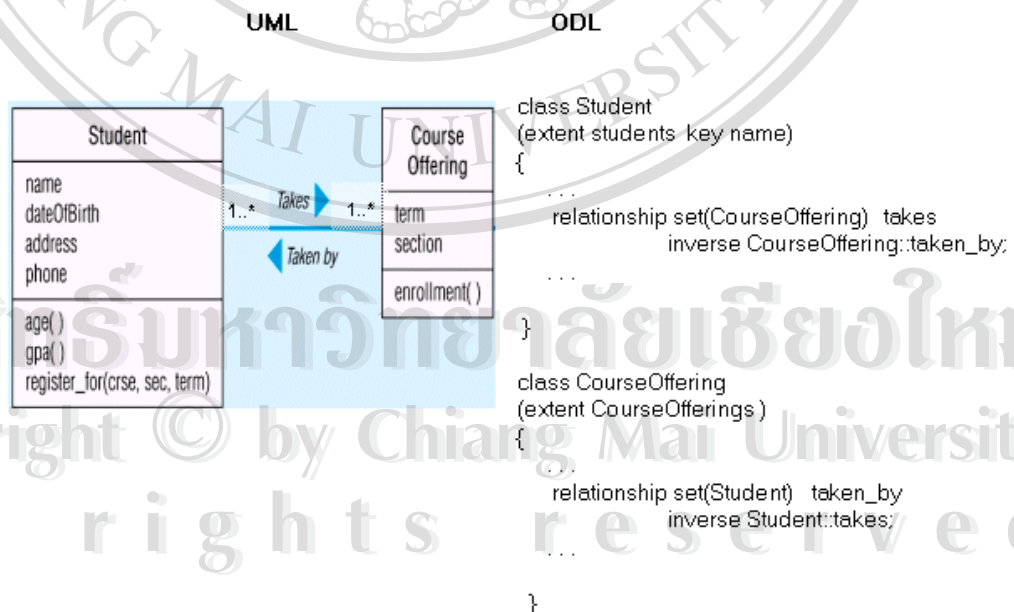
```

รูป 3.5 แสดงการสร้าง ODL Class

(2) แอสโซซิเอชันที่ไม่มีนัยศัพท์วิวัตน์ สร้างความสัมพันธ์ระหว่างคลาสที่เกิดความสัมพันธ์ ดังรูป 3.6 และรูป 3.7

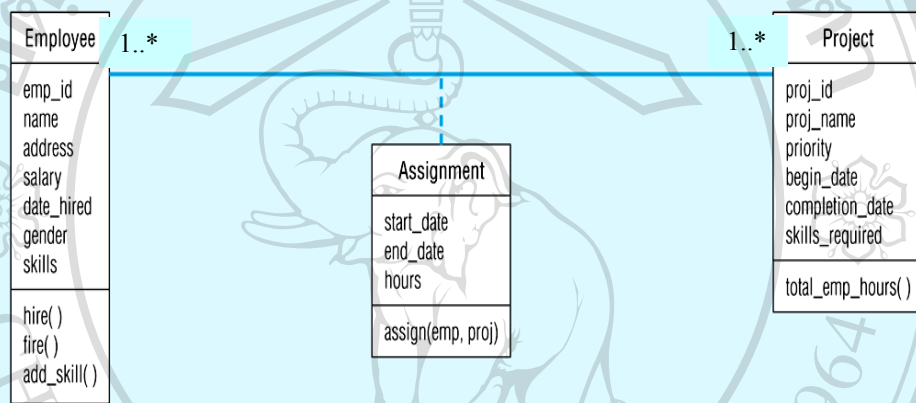


รูป 3.6 แสดง ODL สำหรับแอสโซซิเอชันแบบหนึ่งต่อกลุ่ม

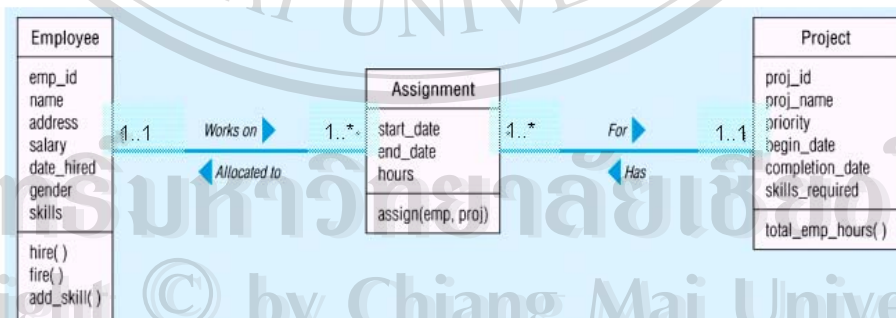


รูป 3.7 แสดง ODL สำหรับแอสโซซิเอชันแบบกลุ่มต่อกลุ่ม

(3) แอสโซซิเอชันที่มีนัยยะแอททริบิวต์และปริมาณความสัมพันธ์ระหว่างคลาสเป็นกลุ่มต่อกลุ่ม สร้าง ODL Class ของแอสโซซิเอชัน คลาส ซึ่งประกอบไปด้วย เซตออบเจกต์ของคลาส แอททริบิวต์และเมธอด หลังจากนั้นสร้างความสัมพันธ์ขึ้นใหม่ระหว่างแอสโซซิเอชัน คลาสกับคลาสทั้งสองที่เกิดความสัมพันธ์กัน โดยปริมาณความสัมพันธ์ระหว่างแอสโซซิเอชัน คลาสกับคลาสทั้งสองจะเป็นหนึ่งต่อกลุ่ม ดังรูป 3.8 – 3.10



รูป 3.8 แสดงแอสโซซิเอชัน คลาส



รูป 3.9 แสดงการสร้างความสัมพันธ์ใหม่ระหว่างแอสโซซิเอชัน คลาสกับคลาส

ODL

```

class Employee
(extent Employees key emp_id)
{
...
relationship set(Assignment) work_on
inverse Assignment::allocated_to;
...
}

class Project
(extent Projects key proj_id)
{
...
relationship set(Assignment) has
inverse Assignment::for;
...
}

class Assignment
(extent Assignments)
{
...
relationship Employee allocated_to
inverse Employee:: work_on;
relationship Project for
inverse Project::has;
...
}

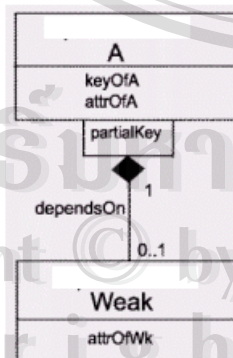
```

รูป 3.10 แสดง ODL สำหรับแอสโซซิเอชัน คลาส

(4) ควอลิไฟ แอสโซซิเอชัน สร้าง ODL Class สำหรับคลาส ซึ่งประกอบไปด้วยแอททริบิวต์ คีย์หลักที่รวมคีย์ของคลาสที่เป็นเจ้าของคลาสนั้นและเมธอด หลังจากนั้นสร้างความสัมพันธ์ระหว่างคลาสที่เป็นเจ้าของและคลาสนั้น ดังรูป 3.11

UML

ODL



```

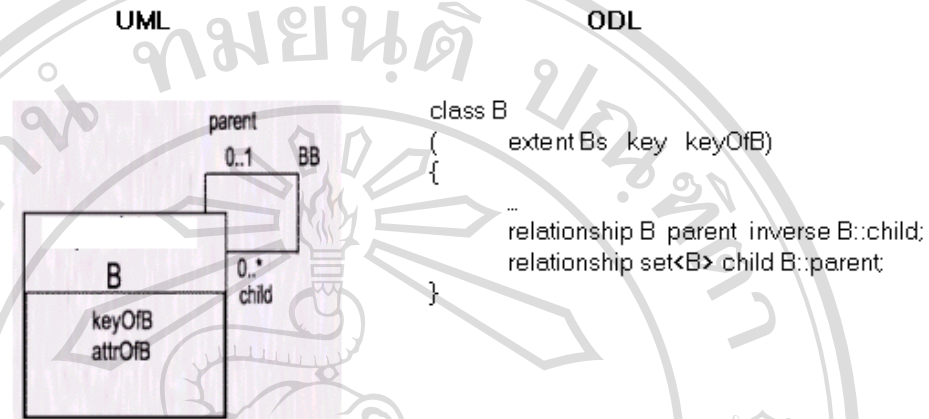
class A
{
    extent As key keyOfA
    {
        attribute string keyOfA;
        relationship set<Weak> linkToWeak
        inverse Weak::linkToOwner;
    }
}

class Weak
{
    extent Weaks key (partialKey, keyOfA)
    {
        attribute string partialKey;
        attribute string keyOfA;
        relationship A linkToOwner
        inverse A::linkToWeak;
    }
}

```

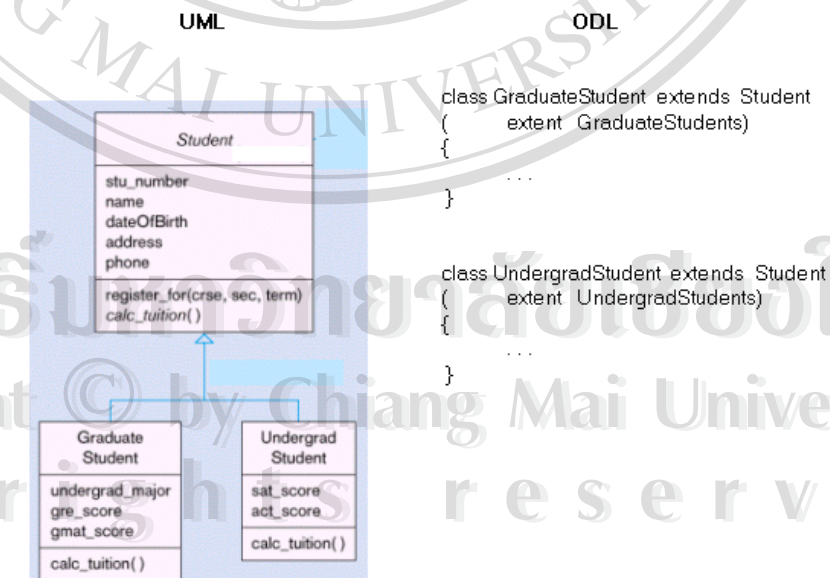
รูป 3.11 แสดง ODL สำหรับควอลิไฟ แอสโซซิเอชัน

- (5) แอสโซซิเอชันแบบวนซ้ำ สร้างความสัมพันธ์ภายในคลาสเดียวกัน ดังรูป 3.12



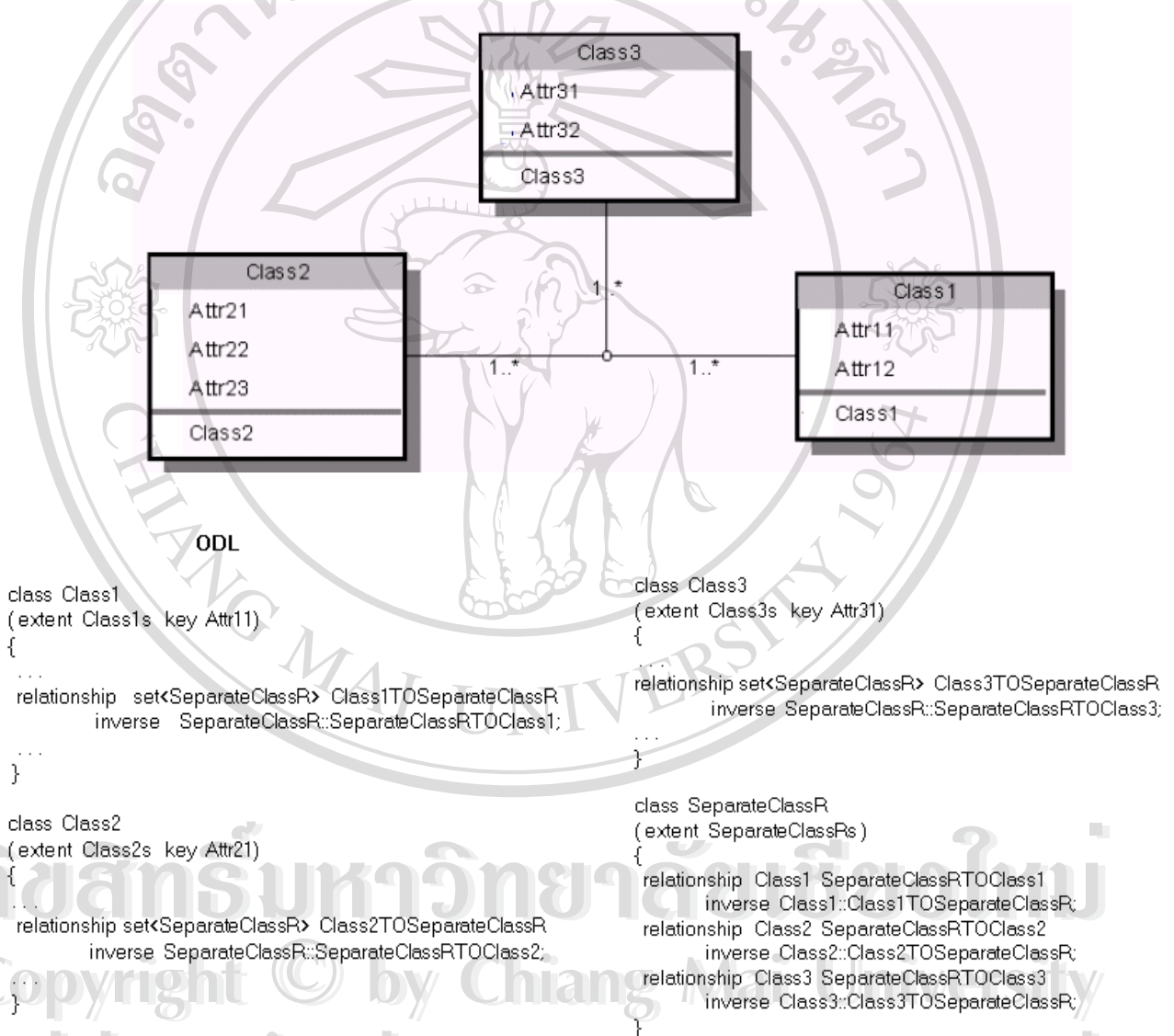
รูป 3.12 แสดง ODL สำหรับ Recursive แอสโซซิเอชันแบบวนซ้ำ

- (6) เจเนอรัลไลเซชันหรือ “Is-a” ชั้นคลาสจะถูกสร้างเป็น ODL Class โดยหลังจากการประกาศชื่อคลาส จะต้องระบุซูเปอร์คลาสที่ถ่ายทอดคุณสมบัติ ดังรูป 3.13



รูป 3.13 แสดง ODL สำหรับเจเนอรัลไลเซชัน

(7) เอ็นนารี แอสโซซิเอชัน กรณีที่เกิดความสัมพันธ์ระหว่างคลาสมากกว่าสองคลาสขึ้นไปให้สร้างคลาสใหม่ขึ้นมา หลังจากนั้นสร้างความสัมพันธ์ขึ้นระหว่างคลาสที่สร้างใหม่กับคลาสเดิมที่มีอยู่แล้ว โดยปริมาณความสัมพันธ์ระหว่างคลาสใหม่กับคลาสที่มีอยู่แล้วจะเป็นหนึ่งต่อกลุ่ม ดังรูป 3.14



รูป 3.14 แสดง ODL สำหรับเอ็นนารี แอสโซซิเอชัน

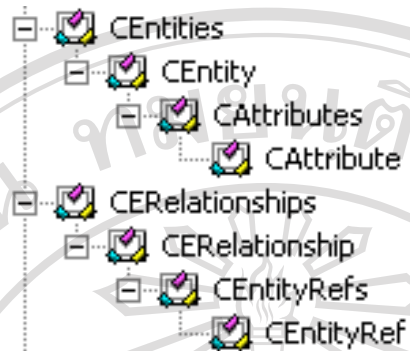
เพื่อประโยชน์ในการจัดเก็บและการประมวลผลอย่างสะดวกและรวดเร็วในแต่ละชั้น
ตอน คำร่างฐานข้อมูลเชิงสัมพันธ์ แบบจำลองความสัมพันธ์เอนทิตี คลาสไดอะแกรมและคำร่างฐาน
ข้อมูลเชิงวัตถุจึงถูกจัดเก็บในรูปของออบเจกต์ โดยประกอบไปด้วยโครงสร้างที่มีลักษณะเป็นลำดับ
ชั้นต่าง ๆ ดังรูป 3.15



- [-] CRelationships
 - [-] CRelationship
- [-] CTables
 - [-] CTable
 - [-] CColumns
 - [-] CColumn

รูป 3.16 แสดงโครงสร้างสำหรับจัดเก็บเค้าร่างฐานข้อมูลเชิงสัมพันธ์

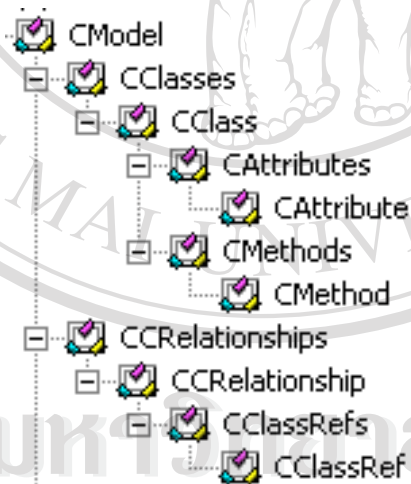
3.5.2 ส่วนจัดเก็บแบบจำลองความสัมพันธ์เอนทิตีประกอบไปด้วยคลาสต่างๆ ดังรูป 3.17



รูป 3.17 แสดงโครงสร้างสำหรับจัดเก็บแบบจำลองความสัมพันธ์เอนทิตี

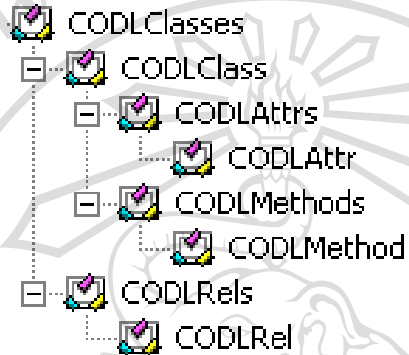
3.5.3 ส่วนจัดเก็บคลาสไดอะแกรม

ประกอบไปด้วยคลาสต่าง ๆ ดังรูป 3.18



รูป 3.18 แสดงโครงสร้างสำหรับจัดเก็บคลาสไดอะแกรม

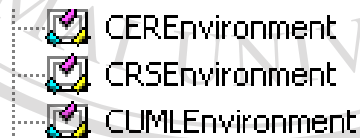
3.5.4 ส่วนจัดเก็บเค้าร่างฐานข้อมูลเชิงวัตถุ ประกอบไปด้วยคลาสต่างๆ ดังรูป 3.19



รูป 3.19 แสดงโครงสร้างสำหรับจัดเก็บเค้าร่างฐานข้อมูลเชิงวัตถุ

3.6 การออกแบบโครงสร้างสำหรับจัดเก็บสภาพแวดล้อมของแผนภาพต่างๆ

สภาพแวดล้อมของแผนภาพ คือ ส่วนที่ใช้สำหรับกำหนดรูปแบบของแต่ละไดอะแกรมให้เป็นไปตามความต้องการของผู้ใช้งาน ประกอบไปด้วยคลาสต่างๆ ดังรูป 3.20



รูป 3.20 แสดงโครงสร้างสำหรับจัดเก็บสภาพแวดล้อมของแผนภาพต่างๆ

3.7 คำอธิบายคลาสต่างๆ

คลาสต่างๆ ที่ใช้ในการจัดเก็บข้อมูลเค้าร่างฐานข้อมูลเชิงสัมพันธ์ แบบจำลองความสัมพันธ์เอนทิตี คลาสไดอะแกรม เค้าร่างฐานข้อมูลเชิงวัตถุและสภาพแวดล้อม อธิบายหน้าที่การใช้งานได้ ดังตาราง 3.1 และ ตาราง 3.2

ตาราง 3.1 แสดงคำอธิบายการใช้งานของคลาส

คลาส	การใช้งาน
CModel	เป็นคอลเลกชันคลาสที่ใช้เก็บกลุ่มของออบเจกต์ที่เกิดขึ้นในไดอะแกรมต่าง ๆ
CTables	เป็นคอลเลกชันคลาสที่ใช้เก็บกลุ่มของออบเจกต์รีเลชัน กลุ่มของออบเจกต์แอททริบิวต์และกลุ่มของออบเจกต์ความสัมพันธ์ระหว่างรีเลชันสำหรับเค้าร่างฐานข้อมูลเชิงสัมพันธ์
CTable	เป็นคลาสที่ใช้เก็บรายละเอียดของรีเลชัน
CColumns	เป็นคอลเลกชันคลาสที่ใช้เก็บกลุ่มของออบเจกต์แอททริบิวต์
CColumn	เป็นคลาสที่ใช้เก็บรายละเอียดของแอททริบิวต์
CRelationships	เป็นคอลเลกชันคลาสที่ใช้เก็บกลุ่มของออบเจกต์ความสัมพันธ์ระหว่างรีเลชัน
CRelationship	เป็นคลาสที่ใช้เก็บรายละเอียดของความสัมพันธ์ระหว่างรีเลชัน
CEntities	เป็นคอลเลกชันคลาสที่ใช้เก็บกลุ่มของออบเจกต์เอนทิตี กลุ่มของออบเจกต์แอททริบิวต์และกลุ่มของออบเจกต์ความสัมพันธ์ระหว่างเอนทิตีสำหรับแบบจำลองความสัมพันธ์เอนทิตี
CEntity	เป็นคลาสที่ใช้เก็บรายละเอียดของเอนทิตี
CAttributes	เป็นคอลเลกชันคลาสที่ใช้เก็บกลุ่มของออบเจกต์แอททริบิวต์
CAttribute	เป็นคลาสที่ใช้เก็บรายละเอียดของแอททริบิวต์
CERelationships	เป็นคอลเลกชันคลาสที่ใช้เก็บกลุ่มของออบเจกต์ความสัมพันธ์ระหว่างเอนทิตี
CERelationship	เป็นคลาสที่ใช้เก็บรายละเอียดของความสัมพันธ์ระหว่างเอนทิตี
CEntityRefs	เป็นคอลเลกชันคลาสที่ใช้เก็บกลุ่มของออบเจกต์เอนทิตีที่เกิดความสัมพันธ์ในความสัมพันธ์ระหว่างเอนทิตีหนึ่ง ๆ
CEntityRef	เป็นคลาสที่ใช้เก็บรายละเอียดของเอนทิตีที่เกิดความสัมพันธ์ในความสัมพันธ์ระหว่างเอนทิตีหนึ่ง ๆ

ตาราง 3.2 แสดงคำอธิบายการใช้งานของคลาส (ต่อ)

คลาส	การใช้งาน
CClasses	เป็นคอลเลกชันคลาสที่ใช้เก็บกลุ่มของออบเจ็กต์คลาส กลุ่มของออบเจ็กต์แอททริบิวต์ กลุ่มของออบเจ็กต์เมธอดและกลุ่มของออบเจ็กต์ความสัมพันธ์ระหว่างคลาสสำหรับคลาสโคโอะแกรม
CClass	เป็นคลาสที่ใช้เก็บรายละเอียดของคลาส
CMethods	เป็นคอลเลกชันคลาสที่ใช้เก็บกลุ่มของออบเจ็กต์เมธอด
CMethod	เป็นคลาสที่ใช้เก็บรายละเอียดของเมธอด
CCRelationships	เป็นคอลเลกชันคลาสที่ใช้เก็บกลุ่มของออบเจ็กต์ความสัมพันธ์ระหว่างคลาส
CCRelationship	เป็นคลาสที่ใช้เก็บรายละเอียดของความสัมพันธ์ระหว่างคลาส
CClassRefs	เป็นคอลเลกชันคลาสที่ใช้เก็บกลุ่มของออบเจ็กต์คลาสที่เกิดความสัมพันธ์ในความสัมพันธ์ระหว่างคลาสหนึ่งๆ
CClassRef	เป็นคลาสที่ใช้เก็บรายละเอียดของคลาสที่เกิดความสัมพันธ์ในความสัมพันธ์ระหว่างคลาสหนึ่งๆ
CODLClasses	เป็นคอลเลกชันคลาสที่ใช้เก็บกลุ่มของออบเจ็กต์คลาส กลุ่มของออบเจ็กต์แอททริบิวต์ กลุ่มของออบเจ็กต์เมธอดและกลุ่มของออบเจ็กต์ความสัมพันธ์ระหว่างคลาสสำหรับเค้าร่างฐานข้อมูลเชิงวัตถุ
CODLClass	เป็นคลาสที่ใช้เก็บรายละเอียดของคลาส
CODLAttrs	เป็นคอลเลกชันคลาสที่ใช้เก็บกลุ่มของออบเจ็กต์แอททริบิวต์
CODLAttr	เป็นคลาสที่ใช้เก็บรายละเอียดของแอททริบิวต์
CODLMethods	เป็นคอลเลกชันคลาสที่ใช้เก็บกลุ่มของออบเจ็กต์เมธอด
CODLMethod	เป็นคลาสที่ใช้เก็บรายละเอียดของเมธอด
CODLRels	เป็นคอลเลกชันคลาสที่ใช้เก็บกลุ่มของออบเจ็กต์ความสัมพันธ์ระหว่างคลาส

ตาราง 3.3 แสดงคำอธิบายการใช้งานของคลาส (ต่อ)

คลาส	การใช้งาน
CODLRel	เป็นคลาสที่ใช้เก็บรายละเอียดของความสัมพันธ์ระหว่างคลาส
CRSEnvironment	เป็นคลาสที่ใช้เก็บรายละเอียดรูปแบบของแผนภาพรีเลชัน
CEREnvironment	เป็นคลาสที่ใช้เก็บรายละเอียดรูปแบบของแผนภาพอี-อาร์
CUMLEnvironment	เป็นคลาสที่ใช้เก็บรายละเอียดรูปแบบของคลาสไดอะแกรม

3.8 การออกแบบโครงสร้างเท็กซ์ไฟล์

เท็กซ์ไฟล์(*.txt) เป็นไฟล์ที่ใช้สำหรับจัดเก็บเค้าร่างฐานข้อมูลเชิงสัมพันธ์ ไฟล์ชนิดนี้จะเกิดขึ้นภายหลังจากผู้ใช้งานได้ทำการกำหนดรายละเอียดและบันทึกเค้าร่างฐานข้อมูลเชิงสัมพันธ์เรียบร้อยแล้ว โดยเท็กซ์ไฟล์จะมีรูปแบบ ดังนี้

RS

<database name>,<DBMS name>

TABLE

<table name>,<database name>

COLUMN

<column1>,<datatype>,<size>,<precisionscale>,<nullable>,<Default>,<primarykey>,<foreignkey>,<description>

<column2>,<datatype>,<size>,<precisionscale>,<nullable>,<Default>,<primarykey>,<foreignkey>,<description>

....

COLEND

TEND

TABLE

....

TEND

....

TEND

RELATION

<table name>,<column name>,<referencedtable name>,

<referencedcolumn name>,<relationship name>

...

REND

RSEND

โดยได้กำหนดคำสำคัญหลักเพื่อใช้ในการตรวจสอบและค้นหาเค้าร่างฐานข้อมูลเชิงสัมพันธ์ ดังตาราง 3.4

ตาราง 3.4 ความหมายของคำสำคัญหลักที่ใช้ในเท็กซ์ไฟล์

คำสำคัญหลัก	ความหมาย
RS ... RSEND	จุดเริ่มต้นและจุดสิ้นสุดของเค้าร่างฐานข้อมูลเชิงสัมพันธ์
TABLE ... TEND	จุดเริ่มต้นและจุดสิ้นสุดของรีเลชัน
COLUMN... COLEND	จุดเริ่มต้นและจุดสิ้นสุดของแอททริบิวต์
RELATION... REND	จุดเริ่มต้นและจุดสิ้นสุดของความสัมพันธ์ระหว่างรีเลชัน

3.9 การออกแบบโครงสร้างเอสดีเอฟไฟล์

เอสดีเอฟไฟล์ (*.sdf) เป็นไฟล์ที่ใช้สำหรับเค้าร่างฐานข้อมูลเชิงสัมพันธ์ แบบจำลองความสัมพันธ์เอนทิตีและคลาสไดอะแกรม ไฟล์ชนิดนี้จะเกิดขึ้นเมื่อผู้ใช้นั่งที่กแผนภาพต่าง ๆ หลังจากมีการสร้างแผนภาพรีเลชันหรือแบบจำลองความสัมพันธ์เอนทิตีหรือคลาสไดอะแกรม โดยเอสดีเอฟไฟล์ จะมีรูปแบบ ดังนี้

MODEL

<DBMS name>,<database name>

RS

TABLE

<table name>,<database name>

COLUMN

```

<column1>,<datatype>,<size>,<PrecisionScale>,<nullable>,<Default>,<primarykey>,<foreignkey>,<description>
<column2>,<datatype>,<size>,<PrecisionScale>,<nullable>,<Default>,<primarykey>,<foreignkey>,<description>
...
COLEND
TEND

```

TABLE

```

...
TEND

```

RELATION

```

<table name>,<column name>,<referencedtable name>,<referencedcolumn name>,<relationship name>
...
REND
RSEND

```

ER

ENTITY

```

<Entity name>,<Entity type>

```

ATTRIBUTE

```

<attribute1>,<datatype>,<size>,<PrecisionScale>,<nullable>,<Default>,<primarykey>,<foreignkey>,<description>,<visibility>

```

<attribute2>,<datatype>,<size>,<PrecisionScale>,<nullable>,<Default>,
<primarykey>,<foreignkey>,<description>,<visibility>

...

ATTREND

ENEND

ENTITY

...

ENEND

REL

<relationship name>,< relationship shortname>,<relationship type>

REF

<relationship name>,<entity name>,<referencedentity name>,<constraint>,<entity type>

REFEND

RELEND

EREND

UML

CLASS

<class name>,<class type>

ATTRIBUTE

<attribute1>,<datatype>,<size>,<PrecisionScale>,<nullable>,<Default>,<primarykey>,<foreignkey>,<description>,<visibility>

<attribute2>,<datatype>,<size>,<PrecisionScale>,<nullable>,<Default>,<primarykey>,<foreignkey>,<description>,<visibility>

...

ATTREND

METHOD

<method1>,<agruments>,<return value>,<visibility>

<method2>,<agruments>,<return value>,<visibility>

...

METHODEND

CLASSEND

CLASS

...

CLASSEND

REL

<relationship name>,< relationship shortname>,<relationship type>

REF

<relationship name>,<entity name>,<referencedentity name>,<constraint>,<entity type>

REFEND

RELEND

UMLEND

ตาราง 3.5 ความหมายของคำสำคัญหลักที่ใช้ในเอสดีเอฟไฟล์

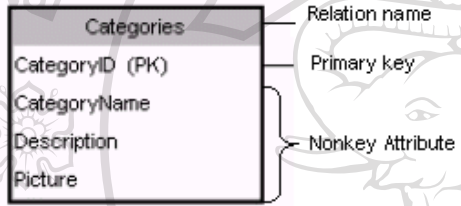
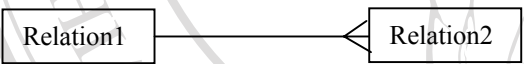
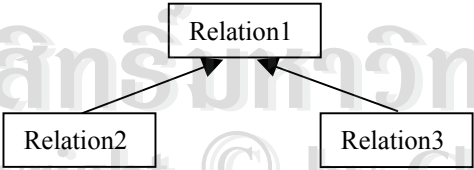
คำสำคัญหลัก	ความหมาย
MODEL	จุดเริ่มต้นของไดอะแกรมต่าง ๆ
RS ... RSEND	จุดเริ่มต้นและจุดสิ้นสุดของเค้าร่างฐานข้อมูลเชิงสัมพันธ์
TABLE ... TEND	จุดเริ่มต้นและจุดสิ้นสุดของรีเลชัน
COLUMN... COLEND	จุดเริ่มต้นและจุดสิ้นสุดของแอททริบิวต์
RELATION... REND	จุดเริ่มต้นและจุดสิ้นสุดของความสัมพันธ์ระหว่างรีเลชัน
ER...EREND	จุดเริ่มต้นและจุดสิ้นสุดของแผนภาพอี-อาร์
ENTITY... ENEND	จุดเริ่มต้นและจุดสิ้นสุดของเอนทิตี
ATTRIBUTE...ATTREND	จุดเริ่มต้นและจุดสิ้นสุดของแอททริบิวต์
REL... RELEND	จุดเริ่มต้นและจุดสิ้นสุดของความสัมพันธ์ระหว่างเอนทิตี
REF... REFEND	จุดเริ่มต้นและจุดสิ้นสุดของเอนทิตีที่เกิดความสัมพันธ์ขึ้นในความสัมพันธ์ระหว่างเอนทิตีหนึ่ง ๆ
UML... UMLEND	จุดเริ่มต้นและจุดสิ้นสุดของคลาสไดอะแกรม
CLASS... CLASSEND	จุดเริ่มต้นและจุดสิ้นสุดของคลาส
ATTRIBUTE...ATTREND	จุดเริ่มต้นและจุดสิ้นสุดของแอททริบิวต์
METHOD... METHODEND	จุดเริ่มต้นและจุดสิ้นสุดของเมธอด
REL... RELEND	จุดเริ่มต้นและจุดสิ้นสุดของความสัมพันธ์ระหว่างคลาส
REF... REFEND	จุดเริ่มต้นและจุดสิ้นสุดของคลาสที่เกิดความสัมพันธ์ขึ้นในความสัมพันธ์ระหว่างคลาสหนึ่ง ๆ

3.10 การออกแบบสัญลักษณ์ที่ใช้ในแผนภาพต่าง ๆ

เป็นการออกแบบสัญลักษณ์ที่ใช้ในแผนภาพรีเลชัน แผนภาพความสัมพันธ์เอนทิตีและคลาสไดอะแกรม โดยได้กำหนดสัญลักษณ์ที่ใช้ในแต่ละแผนภาพ ดังนี้

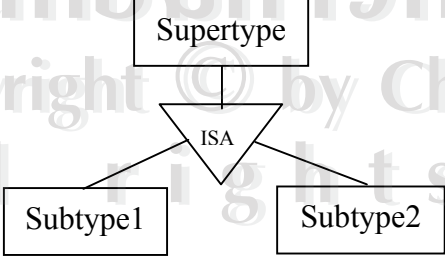
3.10.1 สัญลักษณ์สำหรับแผนภาพรีเลชัน สามารถแทนได้ดังตาราง 3.6

ตาราง 3.6 สัญลักษณ์สำหรับแผนภาพรีเลชัน

สัญลักษณ์	ความหมาย
	รีเลชัน แทนด้วยรูปสี่เหลี่ยมผืนผ้า โดยแบ่งเป็น 2 ส่วนคือ ชื่อรีเลชันและแอททริบิวต์ คีย์แอททริบิวต์ แทนด้วยชื่อแอททริบิวต์ที่เป็นคีย์แอททริบิวต์ ตามด้วยวงเล็บพีเค (PK)
	ความสัมพันธ์แบบหนึ่งต่อกลุ่ม แทนความสัมพันธ์ด้วยเส้นตรง ปลายเส้นตรงด้านกลุ่มจะแตกเป็นสามเส้น
	ความสัมพันธ์แบบกลุ่มต่อกลุ่ม แทนความสัมพันธ์ด้วยเส้นตรง ปลายเส้นตรงทั้งสองด้านแตกเป็นสามเส้น
	ความสัมพันธ์แบบเจเนอรัลไลเซชัน แทนความสัมพันธ์ด้วยเส้นตรง โดยหัวลูกศรชี้ไปยังรีเลชันหลัก

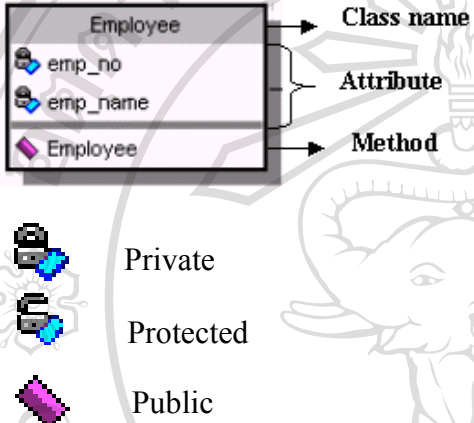
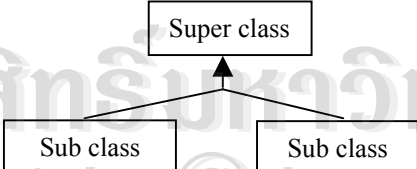
3.10.2 สัญลักษณ์สำหรับแผนภาพความสัมพันธ์เอนทิตี สามารถแทนได้ดังตาราง 3.7

ตาราง 3.7 สัญลักษณ์สำหรับแผนภาพความสัมพันธ์เอนทิตี

สัญลักษณ์	ความหมาย
	เอนทิตีแบบธรรมดา แทนด้วยสี่เหลี่ยมผืนผ้า
	เอนทิตีแบบอ่อน แทนด้วยสี่เหลี่ยมผืนผ้าขอบหนา
	แอททริบิวต์ แทนด้วยวงรี
	คีย์แอททริบิวต์ แทนด้วยวงรีและขีดเส้นใต้ชื่อแอททริบิวต์นั้น
	ความสัมพันธ์ระหว่างเอนทิตี แทนด้วยสี่เหลี่ยมขนมเปียกปูน
	ความสัมพันธ์ระหว่างเอนทิตีแบบธรรมดา กับแบบอ่อน แทนด้วยสี่เหลี่ยมขนมเปียกปูนขอบหนา
	ความสัมพันธ์แบบหนึ่งต่อกลุ่ม แทนหนึ่งด้วยตัวเลข 1 และแทนกลุ่มด้วย M
	ความสัมพันธ์แบบกลุ่มต่อกลุ่ม แทนกลุ่มด้วย M
	ความสัมพันธ์แบบเจเนอรัลไลเซชัน แทนความสัมพันธ์ด้วยสามเหลี่ยม ISA

3.10.3 สัญลักษณ์สำหรับคลาสไคอะแกรม สามารถแทนได้ดังตาราง 3.8

ตาราง 3.8 สัญลักษณ์สำหรับคลาสไคอะแกรม

สัญลักษณ์	ความหมาย
	คลาส แทนด้วยรูปสี่เหลี่ยมผืนผ้าโดยแบ่งเป็น 3 ส่วนคือ ชื่อคลาส แอททริบิวต์และเมธอด
	ปริมาณความสัมพันธ์ระหว่างคลาสแบบหนึ่งต่อกลุ่ม แทน 1 ด้วย 1..1 และแทนกลุ่มด้วย 1..M
	ปริมาณความสัมพันธ์ระหว่างคลาสแบบกลุ่มต่อกลุ่ม แทนกลุ่มด้วย 1..M
	ความสัมพันธ์แบบเจเนอรัลไลเซชัน แทนความสัมพันธ์ด้วยเส้นตรง โดยหัวลูกศรชี้ไปยังซูเปอร์คลาส



ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่

Copyright © by Chiang Mai University
All rights reserved