สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การถ่ายโอนฐานข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ

A transformation from a relational database to an object-oriented database

T055100

โดย

นายวุฒิธร วุฒิเจียรนัย

นายสมภพ นิธิไพจิตร์

อาจารย์ที่ปรึกษา

รศ.ดร. ศุภมิตร จิตตะยโศธร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขหมู่.......

เลขทะเบียน 55100

วัน,เดือน,ปี - 8 เม.ย. 2548

.b.......

.i.......

ปริญญานิพนธ์ ปีการศึกษา 2546

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การถ่ายโอนฐานข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ

A transformation from a relational database to an object-oriented database

คณะผู้จัดทำ นายวุฒิธร วุฒิเจียรนัย รหัส 43010413

นายสมภพ นิธิไพจิตร์ รหัส 43010449

_______________ อาจารย์ที่ปรึกษา

(รศ.ดร. ศุภมิตร จิตตะยโศธร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การถ่ายโอนฐานข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ

นายวุฒิธร วุฒิเจียรนัย 43010413
นายสมภพ นิธิไพจิตร์ 43010449
รศ.ดร. ศุภมิตร จิตตะยโศธร อาจารย์ที่ปรึกษา
ปีการศึกษา 2546

บทคัดย่อ

ในปัจจุบันเทคโนโลยีทางด้านฐานข้อมูลได้รับการพัฒนาไปอย่างมาก ทั้งในแง่ของประสิทธิการทำงาน ทั้งด้านความเร็ว ความสามารถต่างๆ รวมถึงแบบจำลองข้อมูลที่ใช้ในการเก็บข้อมูลได้รับการพัฒนา เช่นเดียวกัน ซึ่งมีการนำแนวความคิดเชิงวัตถุ (Object-Oriented Concept) ที่มีการใช้งานอย่างกว้างขวางในวิธีการพัฒนาซอฟต์แวร์ โดยนำแนวคิดเชิงวัตถุเข้ามาใช้กับระบบฐานข้อมูล โดยเรียกว่าฐานข้อมูลเชิงวัตถุ (Object-Oriented Database) โดยที่ระบบฐานข้อมูลเชิงวัตถุสามารถจัดการข้อมูลที่มีลักษณะซ้ำซ้อนได้มีประสิทธิภาพมากกว่าฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ที่ได้รับความนิยมในอดีต ทำให้มีการศึกษาการถ่ายโอนฐานข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุได้รับความสนใจเป็นอย่างมาก

โครงงานนี้เป็นโครงการที่ทำ การศึกษาแนวทางในการพัฒนาการถ่ายโอนฐานข้อมูล พัฒนาขั้นตอนในการถ่ายโอนฐานข้อมูล และพัฒนาโปรแกรมเพื่อช่วยในการถ่ายโอนโครงสร้างฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A transformation from a relational database to an object-oriented database

Mr. Wuttitorn Wuttijiaranai 43010413

Mr. Somphop Nithipaichit 43010449

Assoc. Prof. Dr. Suphamit Chittayasothon

Advisor

Academic Year 2003

ABSTRACT

At the present time, Database technology has been developed rapidly. In all efficiency such as performance, speed, capability and especially data model for collecting data. Object-oriented concept is widely used in software development and it can apply to use in the database systems that called object-oriented database. Object-oriented database can manage the complex data better than relational database which were popular in the past. Therefore the topic of this project, a transformation from a relational database to an object-oriented database is interesting.

In this project, it was studying about design relational database and object-oriented database, transform of database, reverse engineering database, develop of process and software in transformation of databases.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และความร่วมมือจากหลาย ๆ ฝ่ายขอขอบพระคุณอาจารย์ศุภมิตร จิตตะยโศธร ซึ่งอาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความใส่ใจดูแลเอาใจใส่ ให้คำปรึกษาอย่างใกล้ชิดด้วยดีตลอดมา ทำให้ผู้จัดทำรู้สึกภูมิใจเป็นอย่างมากที่ได้มีโอกาสได้ทำปริญญานิพนธ์นี้

ขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดู ให้โอกาสในการศึกษา และคอยให้กำลังใจเสมอมาในทุก ๆ ด้านอันหาที่เปรียบมิได้ นอกจากนี้ขอขอบพระคุณคณาจารย์ทุกท่านที่ประสิทธิประสาทวิชาความรู้ให้แก่ข้าพเจ้าตลอดมาตั้งแต่ได้มาศึกษาเล่าเรียนในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังจนมีวันนี้

นายวุฒิธร วุฒิเจียรนัย
นายสมภพ นิธิไพจิตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญภาพ	VII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและความเป็นมา	1
1.2 วัตถุประสงค์ของงานวิจัย	1
1.3 ขอบเขตของงานวิจัย	2
1.4 วิธีการดำเนินงาน	2
บทที่ 2 งานวิจัยที่เกี่ยวข้อง	3
บทที่ 3 แบบจำลองข้อมูลอี-อาร์ (ER model)	5
3.1 แบบจำลองข้อมูลซีแมนติก	5
3.2 แบบจำลองข้อมูลอี-อาร์	6
3.3 การออกแบบฐานข้อมูลด้วยแบบจำลองข้อมูลอี-อาร์	15
3.4. ตัวอย่างการออกแบบฐานข้อมูลด้วยแบบจำลองข้อมูลอี-อาร์	16
3.5 การแปลงฐานข้อมูลที่ออกแบบด้วยอี-อาร์โมเดลเป็นฐานข้อมูลเชิงสัมพันธ์	19
บทที่ 4 แนวคิดเชิงวัตถุ	23
4.1 หลักการของแนวความคิดเชิงวัตถุ	23
บทที่ 5 ฐานข้อมูลเชิงวัตถุ	28
5.1 แนวความคิดของฐานข้อมูลแบบเชิงสัมพันธ์ (RDB)	28
5.2 แนวความคิดของฐานข้อมูลแบบเชิงวัตถุ (OODB)	29
5.3 โครงสร้างของภาษากำหนดรูปแบบเชิงวัตถุ	33
บทที่ 6 การแปลงโครงสร้างฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองข้อมูลอี-อาร์	41
6.1. วิธีการแปลงโครงสร้างฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองข้อมูลอี-อาร์	42
6.2 ขั้นตอนในการแปลงโครงสร้างฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองข้อมูลอี-อาร์	42
6.3 ข้อจำกัดของการใช้วิธีวิศวกรรมย้อนกลับ	46
บทที่ 7 การแปลงอีอาร์ไดอะแกรมไปเป็นสคีมาของฐานข้อมูลแบบเชิงวัตถุ	47
7.1 การแปลงจากตัวสคีมาของฐานข้อมูลแบบเชิงสัมพันธ์	47
7.2 การแปลงจากอีอาร์ไดอะแกรม	49
บทที่ 8 การออกแบบและพัฒนาโปรแกรมในการแปลงโครงสร้างฐานข้อมูลเชิงสัมพันธ์	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปยังฐานข้อมูลเชิงวัตถุ
8.1 การออกแบบโครงสร้างโปรแกรม 55
8.2 การออกแบบโปรแกรม 55
บทที่ 9 ผลการทดสอบ 59
9.1 ทดสอบการแปลงโครงสร้างฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองข้อมูลอี-อาร์ 59
9.2 ทดสอบการแปลงแบบจำลองข้อมูลอี-อาร์ไปยังฐานข้อมูลเชิงวัตถุ 61
บทที่ 10 บทสรุปและวิจารณ์ 63
10.1 สรุปผลการดำเนินงาน 63
10.2 แนวทางในการพัฒนาต่อไป 63
บรรณานุกรม 64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้าที่

ตารางที่ 3.1 เอนทิตีอาจารย์ 8
ตารางที่ 3.2 เอนทิตีตารางสอน 8
ตารางที่ 5.1 แสดงตัวอย่าง ฐานข้อมูลแบบเชิงสัมพันธ์ 29
ตารางที่ 5.2 ตารางเปรียบเทียบคุณสมบัติของฐานข้อมูลเชิงสัมพันธ์ กับฐานข้อมูลเชิงวัตถุ 33
ตารางที่ 6.1 ตัวอย่างโครงสร้างฐานข้อมูลเชิงสัมพันธ์ 42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้าที่

รูปที่ 3.1 เอนทิตีหนังสือ 7

รูปที่ 3.2 ตัวอย่างของเอนทิตีอ่อนแอ 7

รูปที่ 3.3 ตัวอย่างแอททริบิวต์ของนักศึกษา 8

รูปที่ 3.4 ตัวอย่างแอททริบิวต์ผสม 9

รูปที่ 3.5 ตัวอย่างแอททริบิวต์คีย์ 9

รูปที่ 3.6 ตัวอย่างแอททริบิวต์หลายค่า 10

รูปที่ 3.7 ตัวอย่างแอททริบิวต์สืบทอด 10

รูปที่ 3.8 ตัวอย่างความสัมพันธ์ระหว่างเอนทิตีนักศึกษาและเอนทิตีคณะ 11

รูปที่ 3.9 ตัวอย่างความสัมพันธ์ระหว่างเอนทิตีนักศึกษาและเอนทิตีชุดวิชาที่มีแอททริบิวต์ 11

รูปที่ 3.10 ตัวอย่างความสัมพันธ์มากกว่าหนึ่งความสัมพันธ์ 11

รูปที่ 3.11 ตัวอย่างความสัมพันธ์แบบการมีส่วนรวมทั้งหมด 12

รูปที่ 3.12 ตัวอย่างความสัมพันธ์แบบการมีส่วนรวมบางส่วน 12

รูปที่ 3.13 ตัวอย่างความสัมพันธ์แบบหนึ่งต่อหนึ่ง 13

รูปที่ 3.14 ตัวอย่างความสัมพันธ์แบบหนึ่งต่อกลุ่ม 13

รูปที่ 3.15 ตัวอย่างความสัมพันธ์แบบกลุ่มต่อกลุ่ม 13

รูปที่ 3.16 ตัวอย่างความสัมพันธ์แบบเอน-นารี 14

รูปที่ 3.17 ตัวอย่างความสัมพันธ์แบบขึ้นอยู่ 14

รูปที่ 3.18 ตัวอย่างความสัมพันธ์แบบรีเคอซีฟ 15

รูปที่ 3.19 ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีคณะและเอนทิตีภาควิชา 17

รูปที่ 3.20 ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีภาควิชาและเอนทิตีอาจารย์ 17

รูปที่ 3.21 ความสัมพันธ์แบบรีเคอซีฟของเอนทิตีอาจารย์ 17

รูปที่ 3.22 ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีภาควิชาและเอนทิตีชุดวิชา 17

รูปที่ 3.23 ความสัมพันธ์แบบกลุ่มต่อกลุ่มระหว่างเอนทิตีชุดวิชาและเอนทิตีอาจารย์ 18

รูปที่ 3.24 เอนทิตีคณะ 18

รูปที่ 3.25 เอนทิตีภาควิชา 18

รูปที่ 3.26 เอนทิตีอาจารย์ 18

รูปที่ 3.27 เอนทิตีชุดวิชา 19

รูปที่ 3.28 อี-อาร์ไดอะแกรมภาระงานสอนของอาจารย์ 19

รูปที่ 3.29 เอนทิตี 20

รูปที่ 3.30 เอนทิติอ่อนแอ 20

รูปที่ 3.31 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง 20

รูปที่ 3.32 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม 21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.33 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม 21
รูปที่ 3.34 แอททริบิวต์หลายค่า 21
รูปที่ 3.35 ความสัมพันธ์แบบ เอน-นารี่ 22
รูปที่ 4.1 แสดงลำดับชั้นของคลาสแบบ Specialization 25
รูปที่ 5.1 ความสัมพันธ์ระหว่างคลาสในรูปแบบลำดับชั้น 32
รูปที่ 5.2 Subclass Employee ถ่ายทอดคุณสมบัติมาจาก Superclass Person 32
รูปที่ 5.3 ตัวอย่าง class family 34
รูปที่ 5.4 แสดงตัวอย่างรูปแบบของความสัมพันธ์ต่างๆ 37
รูปที่ 6.1 ตัวอย่างเอนทิคีนักศึกษา 43
รูปที่ 6.2 ตัวอย่างเอนทิคีอาจารย์ 43
รูปที่ 6.3 ตัวอย่างเอนทิคีอ่อนแอนักฟุตบอล 44
รูปที่ 6.4 ตัวอย่างความสัมพันธ์หนึ่งต่อกลุ่ม 44
รูปที่ 6.5 ตัวอย่างความสัมพันธ์หนึ่งต่อกลุ่ม 45
รูปที่ 6.6 ตัวอย่างความสัมพันธ์หนึ่งต่อกลุ่ม 45
รูปที่ 6.7 ตัวอย่างแอททริบิวต์หลายค่า 46
รูปที่ 7.1 ตัวอย่าง ER Diagram 47
รูปที่ 7.2 โมเดลของฐานข้อมูลและกระบวนการใช้งาน 49
รูปที่ 7.3 แสดงตัวอย่างรูปแบบของความสัมพันธ์ต่างๆ 50
รูปที่ 7.4 เป็นการแสดงความสัมพันธ์ระหว่างเอ็นติตี้ในอีอาร์ไดอะแกรม 51
รูปที่ 7.5 แสดงความสัมพันธ์ระหว่างเอ็นติตี้ในอีอาร์ไดอะแกรมแบบยูนารี 52
รูปที่ 7.6 แสดงรูปอีอาร์ไดอะแกรมใหม่ที่เราได้แปลงรีเลชั่น Contract เป็นเอ็นติตี้ 53
รูปที่ 7.7 เป็นคลาสในภาษากำหนดเชิงวัตถุที่ได้จากการแปลงรูปที่ 7.6 54
รูปที่ 8.1 โครงสร้างโปรแกรม 55
รูปที่ 9.1 หน้าจอเลือกฐานข้อมูลสัมพันธ์ 59
รูปที่ 9.2 หน้าจอแสดงโครงสร้างฐานข้อมูลเชิงสัมพันธ์ 60
รูปที่ 9.3 หน้าจอแสดงแบบจำลองข้อมูลอี-อาร์ 60
รูปที่ 9.4 หน้าจอเลือกรูปแบบไฟล์ฐานข้อมูลเชิงวัตถุ 61
รูปที่ 9.5 หน้าจอแปลงไฟล์ ODL 61
รูปที่ 9.6 หน้าจอแปลงไฟล์ CDL 62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและความเป็นมา

ในปัจจุบัน เทคโนโลยีเชิงวัตถุในการพัฒนาซอฟต์แวร์ได้มีการขยายตัวไปอย่างกว้างขวาง ทำให้แบบจำลองข้อมูลเชิงวัตถุเป็นที่นิยมและเติมโตขึ้นมาก เพื่อรองรับการพัฒนาซอฟต์แวร์เชิงวัตถุให้สะดวกและรวดเร็วยิ่งขึ้น เนื่องจากแบบจำลองข้อมูลต่างๆที่ไม่ใช้แบบจำลองข้อมูลเชิงวัตถุไม่สอดคล้องกับการพัฒนาโปรแกรมเชิงวัตถุ อีกการพัฒนาซอฟต์แวร์เป็นไปอย่างรวดเร็วทำให้แบบจำลองในอดีตนั้นไม่สามารถรองรับการทำงานได้ แบบจำลองข้อมูลเชิงวัตถุจึงเป็นทางเลือกหนึ่งเพื่อใช้ในการเก็บข้อมูล โดยฐานข้อมูลที่นำแบบจำลองข้อมูลเชิงวัตถุมาใช้นั้นจะเรียกว่า ฐานข้อมูลเชิงวัตถุ (Object-Oriented Database)

เนื่องจากแบบจำลองข้อมูลเชิงสัมพันธ์นั้นได้รับความนิยามอย่างมาก ซึ่งฐานข้อมูลที่นำแบบจำลองข้อมูลเชิงสัมพันธ์มาใช้จะเรียกว่า ฐานข้อมูลเชิงสัมพันธ์ ทำให้ความสนใจเกี่ยวกับการถ่ายโอนฐานข้อมูลระหว่างฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ รวมถึงวิธีการวิศวกรรมย้อนกลับ(Reverse engineering) ฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ เพื่อดึงโครงสร้างของฐานข้อมูลไปยังโมเดลอี-อาร์ จึงได้รับความสนใจเป็นอย่างมาก

อย่างไรก็ดี เนื่องจากฐานข้อมูลเชิงสัมพันธ์และฐานข้อมูลเชิงวัตถุมาจากแบบจำลองข้อมูลที่แตกที่แตกต่างกัน การถ่ายโอนฐานข้อมูลระหว่างฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ จึงเกิดปัญหาจากการที่ข้อมูลโครงสร้างของฐานข้อมูลเชิงสัมพันธ์ ไม่เพียงพอ ทำให้ไม่สามารถถ่ายโอนไปยังฐานข้อมูลเชิงวัตถุได้สมบูรณ์ จึงได้มีการนำวิธีการวิศวกรรมย้อนกลับเพื่อดึงโครงสร้างของฐานข้อมูลไปยังแบบจำลองข้อมูลอี-อาร์ให้ได้โมเดลข้อมูลที่มีความสมบูรณ์พอที่จะนำมาสร้างเป็นโครงสร้างฐานข้อมูลเชิงวัตถุได้อย่างสมบูรณ์ยิ่งขึ้น

โครงงานวิจัยนี้ ได้พัฒนาลำดับวิธีการในการถ่ายโอนฐานข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ โดยนำวิธีวิศวกรรมย้อนกลับเข้ามาใช้ โดยโครงงานได้ทำการทดลองสร้างโปรแกรมเพื่อช่วยในการถ่ายโอนโครงสร้างฐานข้อมูล เพื่อช่วยในการถ่ายโอนฐานข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ MS Access ไปยังฐานข้อมูลเชิงวัตถุ Caché

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 ศึกษาวิธีการถ่ายโอนฐานข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ

1.2.2 พัฒนาลำดับและวิธีการในการถ่ายโอนจากฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ

1.2.3 พัฒนาโปรแกรมเพื่อทดลองการถ่ายโอนฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของงานวิจัย

โครงงานนี้เป็นโครงงานที่มีลักษณะเป็นงานวิจัย เพื่อหาขั้นตอนวิธีการในการถ่ายโอนฐานข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ โดยศึกษาวิธีการถ่ายโอนฐานข้อมูล ศึกษาคุณสมบัติของฐานข้อมูลเชิงวัตถุ การออกแบบฐานข้อมูลเชิงสัมพันธ์ และการออกแบบฐานข้อมูลเชิงวัตถุ ศึกษาวิธีวิศวกรรมย้อนกลับจากฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองข้อมูลอี-อาร์ เพื่อหาวิธีที่เหมาะสมในการถ่ายโอนฐานข้อมูล และพัฒนาโปรแกรมทดสอบเพื่อทดลองขั้นตอนวิธีการในการถ่ายโอนโครงสร้างฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ

1.4 วิธีการดำเนินงาน

1.4.1 ศึกษาคุณลักษณะ คุณสมบัติ และการออกแบบฐานข้อมูลเชิงสัมพันธ์

1.4.2 ศึกษาคุณลักษณะ คุณสมบัติ และการออกแบบฐานข้อมูลเชิงวัตถุ

1.4.3 ศึกษาลำดับวิธีการถ่ายโอนฐานข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ

1.4.4 ศึกษาวิธีวิศวกรรมย้อนกลับจากฐานข้อมูลเชิงสัมพันธ์ไปยัง แบบจำลองข้อมูลอี-อาร์

1.4.5 พัฒนาขั้นตอนวิธีการถ่ายโอนโครงสร้างฐานข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ

1.4.6 พัฒนาโปรแกรมเพื่อทดสอบขั้นตอนวิธีการถ่ายโอนโครงสร้างฐานจากฐานข้อมูลเชิงสัมพันธ์ MS Access ไปยังฐานข้อมูลเชิงวัตถุ Caché

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
งานวิจัยที่เกี่ยวข้อง

1 การวิจัยการย้ายฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ (Migration from relational to object-oriented databases) [1] ได้แสดงลำดับการถ่ายโอนฐานข้อมูล ได้แก่

2.1.1. การถ่ายโอนโครงสร้างของฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ

2.1.2. การถ่ายโอนข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ

2.1.3. การแปลงโปรแกรมที่ใช้งานฐานข้อมูลเชิงสัมพันธ์ไปใช้กับฐานข้อมูลเชิงวัตถุ

และในงานวิจัยได้เสนอโปรแกรมที่ช่วยในการถ่ายโอนฐานข้อมูล แต่จะเน้นในส่วนที่เกี่ยวกับการหาคลาสและซับคลาส เพื่อช่วยในการถ่ายโอนโครงสร้างฐานข้อมูล โดยโปรแกรมจะช่วยในการหาซุปเปอร์คลาสและซับคลาสจากโครงสร้างในฐานข้อมูลเชิงสัมพันธ์

2 การวิจัยการย้ายโครงสร้างฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ (Transformation of Relational Schemas to Object-Oriented Schemas)[2] จะแสดงกฎในการตรวจสอบโครงสร้างของฐานข้อมูลเชิงสัมพันธ์ว่า มาจากรูปแบบ (Type) ใดจากแบบจำลองข้อมูลอี-อาร์

2.1 กฎในการตรวจสอบโครงสร้างของฐานข้อมูลเชิงสัมพันธ์ว่าเป็น ว่าเป็นรูปแบบใดของแบบจำลองข้อมูลอี-อาร์ ได้แก่

2.1.1. กฎการตรวจสอบเอนทิตี

2.1.2. กฎการตรวจสอบแอตทริบิวต์แบบกลุ่ม (Set Attribute)

2.1.3. กฎการตรวจสอบแอตทริบิวต์แบบผสม (Composite Attribute)

2.1.4. กฎการตรวจสอบรีเลชั่นชิพแบบหนึ่งต่อหนึ่งและหนึ่งต่อกลุ่ม (1:1 or 1:M Relationship Type)

2.1.5. กฎการตรวจสอบรีเลชั่นชิพแบบกลุ่มต่อกลุ่ม (M:N Relationship Type)

2.1.6. กฎการตรวจสอบประเภทย่อย (Subtype)

2.2 งานวิจัยนี้มีกฎในการแปลงฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ ได้แก่

2.2.1. กฎการแปลงรีเลชั่นเป็นคลาส

2.2.2. กฎการตรวจสอบรีเลชั่นชิพประเภท คลาสย่อยและคลาสหลัก

2.2.3. กฎการตรวจสอบและแปลงรีเลชั่นไปเป็นแอตทริบิวต์แบบผสม

2.2.4. กฎการตรวจสอบและแปลงรีเลชั่นไปเป็นแอตทริบิวต์แบบกลุ่ม

2.2.5. กฎการตรวจสอบและแปลงรีเลชั่นไปเป็นรีเลชั่นชิพกลุ่มต่อกลุ่มแบบยูนารี่ (Unary M:N Relationship Type)

2.2.6. กฎการตรวจสอบและแปลงรีเลชั่นไปเป็นรีเลชั่นชิพกลุ่มต่อกลุ่มแบบไบนารี่ (Binary M: N Relationship Type)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.7. กฎการขยายการสืบทอด

และงานวิจัยยังเสนอวิธีการเพิ่มเมธอดเข้าไปยังโครงสร้างของฐานข้อมูลเชิงวัตถุ ที่มีการนิยามโครงสร้างในรูปแบบคลาส

3 ส่วนการวิจัยการย้ายโครงสร้างฐานข้อมูลและข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุอีกชิ้นหนึ่ง (The migration of relational schemas and data to object-oriented database systems) [3] จะมีการแบ่งกระบวนการในการถ่ายโอนฐานข้อมูลออกเป็น 2 ส่วน คือ

3.1. การแปลงโครงฐานข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ โดยในส่วนนี้จะมีการเสนอกฎในการแปลงโครงสร้างจากรีเลชั่นไปยังคลาส ซึ่งแบ่ง กฎออกเป็น 2 ส่วนใหญ่ คือ

3.1.1. กฎที่ใช้ในการสร้างคลาส

3.1.2. กฎที่ใช้ในการสร้างแอตทริบิวต์

โดยกฎที่ใช้ค่อนข้างคล้ายคลึงกับงานวิจัยในหัวข้อ 2.2.2 แต่มีการเพิ่มเติมกฎเพิ่มขึ้นมาบ้าง

3.2. การถ่ายโอนข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ สามารถแบ่งเป็น 2 ส่วนใหญ่ คือ

3.2.1. การสร้างอินสแตนท์ของแต่ละคลาส

3.2.2. การกำหนดค่าให้แต่ละอินสแตนท์

และภายในงานวิจัยนี้ยังได้เสนอสถาปัตยกรรมของเครื่องมือเพื่อช่วยในการถ่ายโอนฐานข้อมูลว่าประกอบด้วยส่วนใดบ้าง

4 การวิจัยวิศวกรรมย้อนกลับจากโครงสร้างฐานข้อมูลเชิงสัมพันธ์ไปยังโครงสร้างฐานข้อมูลเชิงวัตถุ (Reverse Engineering Relational Schemas to Object-Oriented Schemas)[4] จะเป็นการใช้วิธีวิธีวิศวกรรมย้อนกลับ เพื่อแปลงโครงสร้างของฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองข้อมูลเอนทิตี-รีเลชั่นชิพ และนำแบบจำลองข้อมูลอี-อาร์ มาสร้างโครงสร้างของฐานข้อมูลเชิงวัตถุ โดยเสนอว่า โครงสร้างของฐานข้อมูลเชิงสัมพันธ์ที่จะนำมาใช้ในการแปลงนั้น จะต้องผ่าน 3 นอร์มอลฟอร์ม เนื่องจากถ้าหากไม่ฐานข้อมูลเชิงสัมพันธ์ไม่ผ่าน 3 นอร์มอลฟอร์มเมื่อนำมาแปลงไปเป็นโครงสร้างของฐานข้อมูลเชิงวัตถุ จะทำให้ได้โครงสร้างของฐานข้อมูลที่ไม่ดี และแสดงวิธีการในการตรวจสอบโครงสร้างของฐานข้อมูลเชิงวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

แบบจำลองข้อมูล อี-อาร์ (ER model)

ในการออกแบบฐานข้อมูล นิยมใช้แบบจำลองข้อมูล (Data model) เพื่อนำเสนอรายละเอียดต่างๆ เกี่ยวกับโครงสร้างและความสัมพันธ์ระหว่างข้อมูลภายในฐานข้อมูล เนื่องจากฐานข้อมูลที่อยู่ในรูปแบบตรรกะ (Logical) ยากต่อการทำความเข้าใจ จึงมีแบบจำลองข้อมูลหลายระดับเพื่อที่จะให้ผู้ใช้ใจแต่ละระดับมีมุมมองสามารถเข้าใจได้ง่ายขึ้น

ในอดีต แบบจำลองในระยะแรกๆ จะมีการนำเสนอเฉพาะรายละเอียดที่เกี่ยวข้องกับโครงสร้างของฐานข้อมูล เช่น การควบคุมบูรณภาพของข้อมูล ซึ่งยังขาดรายละเอียดที่จะนำเสนอความหมายภายในฐานข้อมูล เช่น นักเรียนกับห้องเรียนมีความสัมพันธ์กันอย่างไร ขอบเขตของข้อมูล(Domain) ว่าข้อมูลสามารถมีค่าใดที่เป็นไปได้บ้าง ข้อมูลเป็นคีย์คู่แข่งหรือเป็นคีย์นอก ดังนั้น ในทศวรรษที่ 70 ได้มีการนำเสนอแบบจำลองข้อมูลที่เรียกว่า แบบจำลองข้อมูลซีแมนติก (Semantic model) ขึ้น

3.1 แบบจำลองข้อมูลซีแมนติก

แบบจำลองข้อมูลซีแมนติก เป็นแบบจำลองข้อมูลที่กล่าวถึงแนวคิด หรือความหมายของข้อมูลต่างๆ ในฐานข้อมูล ซึ่งประกอบด้วย เอนทิตี พรอพเพอตี (Property) ความสัมพันธ์ ประเภทย่อย (Subtype)

3.1.1. เอนทิตี หมายถึง สิ่งที่สนใจสามารถระบุได้ในความเป็นจริง และต้องการจัดเก็บข้อมูลที่เกี่ยวข้องด้วย โดยอาจเป็นสิ่งที่เป็นรูปธรรมสามารถจับต้องได้ เช่น นักศึกษาในระบบฐานข้อมูล หรืออาจเป็นนามธรรมที่ไม่สามารถมองเห็นได้ เช่น การลงทะเบียนของนักศึกษา

หากเราทำการนำเอนทิตีที่มีคุณสมบัติที่คล้ายคลึงกัน มารวมกัน เราจะเรียกเอนทิตีนั้นว่า เอนทิตี ไทป์ (Entity Type) ซึ่งหมายถึงกลุ่มของเอนทิตีหรือกลุ่มของข้อมูลที่มีคุณสมบัติเหมือนกัน เช่น เอนทิตีไทป์ของหนังสือในห้องสมุด หมายถึง กลุ่มของหนังสือภายในห้องสมุด เป็นต้น

โดยเอนทิตีในที่เราทำการสนใจนั้น เราจะเรียกว่า ตัวแทนของเอนทิตี (Entity instance) ซึ่งเอนทิตีไทป์ใดๆ อาจมีเอนทิตีปรากฏได้เป็นจำนวนมาก

3.1.2. พรอพเพอตี หมายถึง คุณสมบัติหรือคุณลักษณะของเอนทิตี ซึ่งในแต่ละเอนทิตีไทป์เดียวกันจะมีคุณสมบัติต่างๆที่เหมือนกัน เช่น พรอพเพอตีของเอนทิตีไทป์หนังสือ จะประกอบด้วย รหัสหนังสือ ชื่อหนังสือ ประเภทหนังสือ เป็นต้น

ถึงแม้ว่า เอนทิตีที่เป็นสมาชิกของเอนทิตีเดียวกันจะมีคุณสมบัติต่างๆเหมือนกัน แต่ก็จะมีข้อมูลใดในพรอพเพอตี หนึ่งที่เป็นเอกลักษณ์เฉพาะของแต่ละเอนทิตีที่จะแบ่งแยกแต่ละเอนทิตีออกจากกัน เช่น เอนทิตีไทป์หนังสือ จะมีพรอพเพอตีของรหัสหนังสือที่หนังสือทุกเล่มจะไม่มีรหัสหนังสือซ้ำกันเลย ซึ่ง พรอพเพอตีที่สามารถกำหนดเอกลักษณ์เฉพาะให้กับแต่ละเอนทิตีเราจะเรียกว่า ไอเดนทิที (Identity)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งถ้าหากเราพิจารณาดูจะพบว่า แต่ละพรอพเพอตี ก็คือแต่ละแอททริบิวต์ (Attribute) ส่วนเอนทิตีก็คือแต่ละทูเพิล (tuple) ในแบบจำลองข้อมูลเชิงสัมพันธ์ แต่เนื่องจากแบบจำลองข้อมูลซีแมนติกถูกนำมาใช้เพื่อแสดงความเกี่ยวข้องของข้อมูล โดยไม่มีการระบุโครงสร้างของฐานข้อมูลที่ใช้ ดังนั้นจึงใช้คำว่า พรอพเพอตีแม้นคำว่าแอททริบิวต์และคำว่า เอนทิตี แทนคำว่า ทูเพิล

3.1.3. ความสัมพันธ์ หมายถึง แต่ละเอนทิตีไทป์ที่แสดงความสัมพันธ์ระหว่างสองเอนทิตีไทป์ ขึ้นไปเป็นความสัมพันธ์ที่สมาชิกของเอนทิตีไทป์หนึ่ง มีความสัมพันธ์กับเอนทิตีหนึ่ง โดยแต่ละความสัมพันธ์จะถูกระบุด้วยชื่อถึงความสัมพันธ์นั้นๆ เช่น เอนทิติไทป์นักศึกษา มีความสัมพันธ์ เอนทิตีไทป์วิชาเรียน

3.1.4. ประเภทย่อย เมื่อเอนทิตีไทป์มีพรอพเพอตีที่สามารถกำหนดเอกลักษณ์เฉพาะให้กับแต่ละเอนทิตีได้ และข้อมูลทุกเอนทิตีในเอนทิตีไทป์นั้นเป็นข้อมูลส่วนหนึ่งในเอนทิตีหลัก จะเรียกเอนทิตีไทป์นั้นว่าเป็นเอนทิตีไทป์ย่อยของเอนทิตีไทป์หลัก เช่น เอนทิตีไทป์หนังสือคอมพิวเตอร์และเอนทิตีไทป์นิยายเป็นประเภทย่อยของเอนทิตีไทป์นักหนังสือ

3.2 แบบจำลองข้อมูลอี-อาร์ (Entity-Relationship Model)

แบบจำลองข้อมูลอี-อาร์ เป็นแบบจำลองข้อมูลที่ได้รับการประยุกต์และออกแบบมาจากแนวความคิดของแบบจำลองข้อมูลซีแมนติก และได้รับความนิยามอย่างมากสำหรับนำมาใช้เพื่อการออกแบบฐานข้อมูลในระดับแนวความคิด โดยอี-อาร์เป็นผลงานวิจัยของ พีเตอร์ พี เชง (Peter Pin Chen) จากสถาบันเทคโนโลยีแมทซาชูเซท (Massachusetts Institute of Technology) ในปี ค.ศ. 1976

แบบจำลองข้อมูลอี-อาร์ เป็นแบบจำลองข้อมูลซึ่งแสดงถึงโครงสร้างของฐานข้อมูลที่เป็นอิสระจากซอฟต์แวร์ที่ใช้ในการพัฒนาฐานข้อมูล รวมทั้งประกอบด้วยรายละเอียดและความสัมพันธ์ระหว่างข้อมูลในระบบในลักษณะที่เป็นภาพรวม ทำให้เป็นประโยชน์อย่างมากต่อการรวบรวมและวิเคราะห์รายละเอียดตลอดจนความสัมพันธ์ของข้อมูลต่างๆ โดยอี-อาร์โมเดลมีการใช้สัญลักษณ์ต่างๆ ที่เรียกว่า แผนภาพอี-อาร์ (E-R Diagram) แทนรูปแบบของข้อมูลเชิงตรรกะขององค์กร จึงทำให้บุคลากรที่เกี่ยวข้องกับระบบฐานข้อมูลสามารถเข้าใจลักษณะของข้อมูลและความสัมพันธ์ระหว่างข้อมูลได้ง่ายและถูกต้องตรงกัน ระบบที่ได้รับการออกแบบจึงมีความถูกต้องและสอดคล้องกับวัตถุประสงค์ขององค์กร โดยแบบจำลองข้อมูลอี-อาร์ประกอบด้วย 4 ส่วนคือ เอนทิตี แอททริบิวต์ความสัมพันธ์ ประเภทรอง และประเภทหลัก

3.2.1. เอนทิตี (Entity Type) หมายถึง สิ่งที่สนใจสามารถระบุได้ในความเป็นจริง และต้องการเก็บรวบรวมข้อมูลที่เกี่ยวข้องด้วยไว้ใน ฐานข้อมูล โดยตัวอย่างของเอนทิตีต่างๆ เช่น บุคคล สถานที่ สิ่งของ หรือเหตุการณ์ มีดังนี้

- บุคคล เช่น นักเรียน อาจารย์ ผู้ป่วย และ พนักงาน เป็นต้น
- สถานที่ เช่น เขต แขวง จังหวัด ภาค และประเภท เป็นต้น
- วัตถุ เช่น คอมพิวเตอร์ รถยนต์ เครื่องบิน และเครื่องจักร เป็นต้น
- เหตุการณ์ เช่น การลงทะเบียน การยืมหนังสือ เป็นต้น

ซึ่งในแผนภาพ อี-อาร์ จะให้สัญลักษณ์รูปสี่เหลี่ยมผืนผ้า (Rectangle) แทนหนึ่งเอนทิตี โดยมีชื่อเอนทิตีนั้นกำกับอยู่ภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปภาพที่ 3.1 เอนทิตีหนังสือ

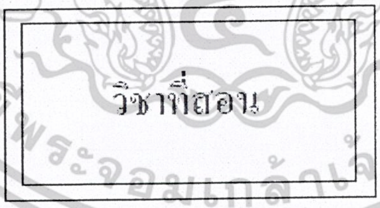
เอนทิตีสามารถจำแนกออกได้เป็น 2 ประเภท คือ เอนทิตีปกติ (Regular entity type) และเอนทิตีอ่อนแอ (Weak entity type)

3.2.1.1 เอนทิตีปกติ หรือ เอนทิตีหลัก (Strong entity type) หมายถึงเอนทิตีที่สนใจและต้องการจัดเก็บข้อมูลที่เกี่ยวข้องไว้ในระบบฐานข้อมูล ซึ่งการคงอยู่ของเอนทิตีนี้ไม่เกี่ยวข้องกับเอนทิตีอื่น โดยเอนทิตีนี้สามารถมีคุณสมบัติแยกความแตกต่าง (Identity) ได้ตัวเอง

ในอี-อาร์ไดอะแกรม สัญลักษณ์ที่ใช้แทนเอนทิตีปกติเป็นเช่นเดียวกับเอนทิตี ดังนั้น เมื่อมีกล่าวถึงเอนทิตีใด ๆ โดยไม่มีการระบุรายละเอียดอื่น จึงมีหมายความถึงเอนทิตีปกตินั่นเอง

3.2.1.2 เอนทิตีอ่อนแอ หมายถึง เอนทิตีที่มีการคงอยู่เกี่ยวข้องกับเอนทิตีอื่นในระบบฐานข้อมูลโดยเอนทิตีอื่นที่มีความสัมพันธ์กับเอนทิตีนี้เรียกว่า เอนทิตีหลัก หรือจากกล่าวว่า เอนทิตีอ่อนแอจะไม่มีความหมายหรือไม่สามารถปรากฏในฐานข้อมูลได้ หากปราศจาก เอนทิตีหลักที่มีความสัมพันธ์เกี่ยวข้องกัน ซึ่งสมาชิกของเอนทิตีอ่อนแอจะสามารถมีคุณสมบัติแยกความแตกต่างได้ก็ต่อเมื่ออาศัย แอททริบิวต์ใดแอททริบิวต์หนึ่งของเอนทิตีหลักมาประกอบแอททริบิวต์นั้น

ในแผนภาพอี-อาร์ จะใช้สัญลักษณ์รูปสี่เหลี่ยมผืนผ้าสองรูปซ้อนกันแทนหนึ่งเอนทิตีอ่อนแอ โดยมีชื่อของเอนทิตีอ่อนแอนั้น กำกับอยู่



รูปที่ 3.2 ตัวอย่างของเอนทิตีอ่อนแอ

ทั้งนี้ ตัวอย่างของเอนทิตีปกติและเอนทิตีอ่อนแออาจอธิบายได้ดังภาพที่ 3.3 คือ ในสถานศึกษา แห่งหนึ่ง อาจารย์แต่ละคนจะมีรหัสอาจารย์ที่ไม่ซ้ำกัน ดังนั้น เอนทิตีอาจารย์จะเป็นเอนทิตีปกติที่มีแอททริบิวต์รหัสอาจารย์เป็นตัวแยกความแตกต่าง

ในเอนทิตีตารางสอน อาจารย์แต่ละคนอาจจะสอนชุดวิชาเดียวกันและหมู่เรียนเดียวกัน หรือบางคนอาจจะสอนชุดวิชาเดียวกันแต่คนละหมู่เรียนก็ได้ ฉะนั้นหากไม่มีเอนทิตีอาจารย์ก็จะไม่สามารถทราบ ด้ว่าอาจารย์คนใด สอนชุดวิชาใด หมู่เรียนใด ในวัน/เวลาใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ เอนทิตีตารางสอนจะสามารถมีคุณสมบัติแยกความแตกต่างได้ก็ต่อเมื่ออาศัย แอททริบิวต์รหัสอาจารย์ของเอนทิตีอาจารย์ซึ่งเป็นเอนทิตีปกติมาประกอบกับแอททริบิวต์วัน/เวลาสอนของ เอนทิตีตารางสอน ฉะนั้น เอนทิตีตารางสอนจะไม่สามารถคงอยู่ได้ หากปราศจากเอนทิตีอาจารย์ ดังนั้น จึง กล่าวได้ว่าเอนทิตีตารางสอนเป็นเอนทิตีอ่อนแอ

รหัส	ชื่อ-สกุล	เพศ	เงินเดือน
Q1011	ศิริภัทรา เหมือนมาลัย	หญิง	8,000
Q1023	สุพิมพ์ ศรีพันธ์วรสกุล	หญิง	9,500
Q1035	ศิริชัย ศรีพรหม	ชาย	12,000

ตารางที่ 3.1 เอนทิตีอาจารย์

รหัส	รหัสชุดวิชาที่	หมู่เรียน	วัน/เวลาสอน	ห้องเรียน
Q1011	729113	700	อ.,พฤ. 08.00-10.00 น.	LH1-201
Q1011	999211	711	จ.,พ. 10.00-12.00 น.	SC4-308
Q1023	999211	712	จ.,พ. 13.00-15.00 น.	SC4-308
Q1023	999211	713	อ.,พฤ. 08.00-10.00 น.	SC4-308
Q1023	999211	714	อ.,พฤ. 10.00-12.00 น.	SC4-308
Q1035	172596	700	จ.,พ. 10.00-12.00 น.	LH1-201
Q1035	729113	700	อ.,พฤ. 08.00-10.00 น.	LH1-201
Q1035	999211	700	ศ. 10.00-12.00 น.	LH1-202
Q1035	999211	715	อ.,พฤ. 13.00-15.00 น.	SC4-308
Q1035	999211	716	จ.,พ. 08.00-10.00 น.	SC4-308

ตารางที่ 3.2 เอนทิตีตารางสอน

3.2.2. แอททริบิวต์ หมายถึง ข้อมูลที่แสดงคุณสมบัติหรือคุณลักษณะของเอนทิตีหรือความสัมพันธ์ เช่น แอททริบิวต์ของหนังสือประกอบด้วย รหัสหนังสือ ชื่อหนังสือ เป็นต้น

ในแผนภาพอี-อาร์ได้ใช้สัญลักษณ์รูปวงรี ที่มีชื่อแอททริบิวต์นั้นกำกับอยู่ภายในแทนหนึ่งแอททริบิวต์เชื่อมต่อกับเอนทิตีที่มีแอททริบิวต์นั้นด้วยเส้นตรง

รูปที่ 3.3 ตัวอย่างแอททริบิวต์ของนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

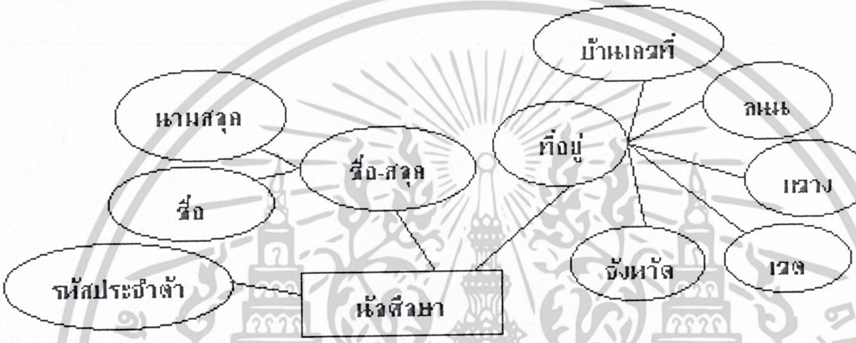
แอททริบิวต์สามารถจำแนกได้เป็น 6 ประเภท ได้แก่

3.2.2.1. แอททริบิวต์อย่างง่าย (Simple attribute) หมายถึง แอททริบิวต์ที่ไม่สามารถแบ่งย่อยลงไปได้อีก เช่น แอททริบิวต์ ชื่อ นามสกุล เพศ เป็นต้น

โดยในแผนภาพ อี-อาร์จะใช้สัญลักษณ์ของแอททริบิวต์ย่างง่าย เหมือนกับแอททริบิวต์ทั่วไป

3.2.2.2. แอททริบิวต์ผสม (Composite attribute) หมายถึงแอททริบิวต์ที่สามารถแบ่งแยกย่อยลงไปอีกได้ เช่น แอททริบิวต์ ชื่อ-สกุล สามารถแบ่งแยกได้เป็น 2 แอททริบิวต์ย่างง่าย คือ แอททริบิวต์ ชื่อ และนามสกุล หรือแอททริบิวต์ที่อยู่ สามารถแบ่งเป็น 5 แอททริบิวต์อย่างง่าย คือแอททริบิวต์ เลขที่บ้าน ถนน แขวง เขต และจังหวัดเป็นต้น

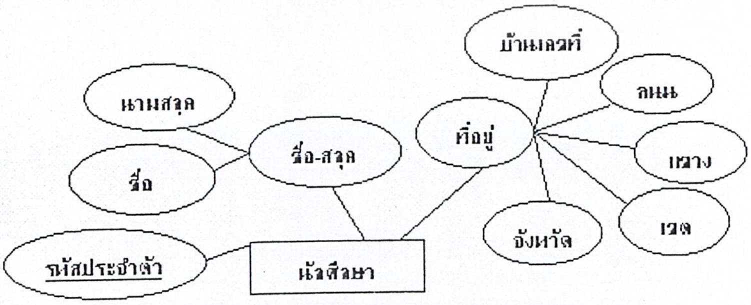
ในแผนภาพ อี-อาร์ จะใช้สัญลักษณ์รูปวงรีที่มีชื่อของแอททริบิวต์นั้นกำกับอยู่ภายในแทนแอททริบิวต์ผสม และเชื่อมต่อกับแอททริบิวต์อย่างง่ายที่จำแนกออกไปด้วยเส้นตรง



รูปที่ 3.4 ตัวอย่างแอททริบิวต์ผสม

3.2.2.3. แอททริบิวต์คีย์ (Key attribute) หมายถึงแอททริบิวต์หรือกลุ่มของแอททริบิวต์ที่มีค่าของข้อมูลในแต่ละสมาชิกจองเอนทิตีไม่ซ้ำกัน ทำให้สามารถระบุความแตกต่างของแต่ละสมาชิกในเอนทิตีได้ เช่น เอนทิตีนักศึกษาประกอบด้วยแอททริบิวต์ รหัสประจำตัว ชื่อ-สกุล และที่อยู่ โดยแอททริบิวต์ที่สามารถบอกความแตกต่างของนักศึกษาแต่ละคนได้ คือ รหัสประจำตัว ดังนั้น แอททริบิวต์รหัสประจำตัวจึงเป็นแอททริบิวต์คีย์ของเอนทิตีนักศึกษา เป็นต้น

ในแผนภาพอี-อาร์ ใช้สัญลักษณ์รูปวงรีซึ่งภายในประกอบด้วยชื่อของแอททริบิวต์ที่มีการขีดเส้นใต้แทนแอททริบิวต์คีย์และเชื่อมต่อกับเอนทิตีนั้นด้วยเส้นตรง



รูปที่ 3.5 ตัวอย่างแอททริบิวต์คีย์

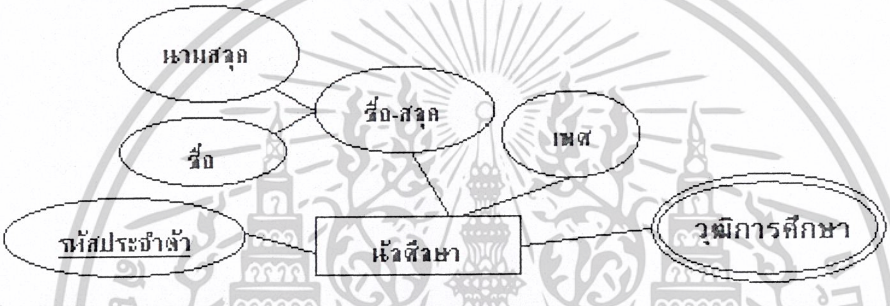
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.4. แอททริบิวต์ค่าเดียว (Single-Valued attribute) หมายถึง แอททริบิวต์ที่มีค่าของแต่ละสมาชิกของเอนทิตีได้เพียงค่าเดียว เช่น บุคคลหนึ่งมีเพศเดียว แอททริบิวต์เพศจึงเป็นแอททริบิวต์ค่าเดียว หรือพนักงานคนหนึ่งมีเงินเดือนเพียงค่าเดียว แอททริบิวต์เงินเดือนจึงเป็นแอททริบิวต์ค่าเดียว

ในแผนภาพอี-อาร์ จะใช้สัญลักษณ์ที่ใช้แทนแอททริบิวต์ค่าเดียวจะเป็นเช่นเดียวกับแอททริบิวต์ อย่างง่าย

3.2.2.5. แอททริบิวต์หลายค่า (Multi-Valued attribute) หมายถึงแอททริบิวต์ที่สามารถมีค่าของข้อมูลในแต่ละสมาชิกของเอนทิตีได้หลายค่า เช่น บุคคลหนึ่งสามารถมีวุฒิการศึกษาได้หลายระดับ แอททริบิวต์วุฒิการศึกษาจึงเป็นแอททริบิวต์หลายค่า หรือบ้านหลังหนึ่งอาจมีหลายเบอร์โทรศัพท์ แอททริบิวต์เบอร์โทรศัพท์จึงเป็น แอททริบิวต์หลายค่า

ในแผนภาพอี-อาร์ จะใช้สัญลักษณ์รูปวงรีซ้อนกัน โดยที่มีชื่อแอททริบิวต์นั้นกำกับอยู่ภายในแทนหนึ่งแอททริบิวต์หลายค่า และเชื่อมต่อกับเอนทิตีที่มีแอททริบิวต์นั้นด้วยเส้นตรง



รูปที่ 3.6 ตัวอย่างแอททริบิวต์หลายค่า

3.2.2.6. แอททริบิวต์สืบทอด (Derived attribute) หมายถึงแอททริบิวต์ที่ค่าของข้อมูลในแต่ละสมาชิกของเอนทิตีได้มาจากการนำ ค่าของข้อมูลในแอททริบิวต์อื่นที่มีอยู่ในแต่ละสมาชิกของเอนทิตีมาทำการคำนวณ ซึ่งโดยทั่วไปไม่จำเป็นต้องจัดเก็บแอททริบิวต์ประเภทนี้เอาไว้ในฐานข้อมูล เนื่องจากแอททริบิวต์ประเภทนี้มีการเปลี่ยนแลงค่าของแต่ละสมาชิกของเอนทิตีทุกครั้ง เมื่อมีการเปลี่ยนแปลงค่าของข้อมูลในแอททริบิวต์ ที่ถูกนำมาคำนวณของแต่ละสมาชิกในเอนทิตี เช่น แอททริบิวต์ของอายุปัจจุบันสามารถคำนวณได้จากแอททริบิวต์ วัน/เดือน/ปี เกิด เป็นต้น

ในแผนภาพอี-อาร์ ใช้สัญลักษณ์รูปวงรีที่เป็นเส้นประ ที่มีชื่อของแอททริบิวต์นั้นกำกับอยู่ภายในแทนหนึ่ง แอททริบิวต์ สืบทอด

รูปที่ 3.7 ตัวอย่างแอททริบิวต์สืบทอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3. ความสัมพันธ์ (Relationship type) หมายถึง เอนทิตีที่แสดงความสัมพันธ์ระหว่างสองเอนทิตีขึ้นไป ซึ่งโดยทั่วไปเป็นความสัมพันธ์ระหว่างเอนทิตีที่มีแอททริบิวต์รวมกัน โดยแสดงความสัมพันธ์ระหว่างเอนทิตีนักศึกษาและเอนทิตีคณะ เป็นต้น

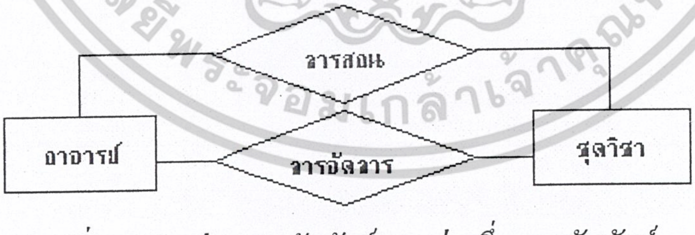
ในอี-อาร์ไดอะแกรม ใช้สัญลักษณ์รูปสี่เหลี่ยมข้าวหลามตัด ที่มีชื่อของความสัมพันธ์นั้นกำกับอยู่ภายในแทนหนึ่งความสัมพันธ์ และเชื่อมความสัมพันธ์กับเอนทิตีเกี่ยวข้องกับความสัมพันธ์นั้นด้วยเส้นตรง

รูปที่ 3.8 ตัวอย่างความสัมพันธ์ระหว่างเอนทิตีนักศึกษาและเอนทิตีคณะ

นอกจากนี้ ความสัมพันธ์ระหว่างเอนทิตีที่มีแอททริบิวต์ร่วมกันจะเป็นตัวกำหนดความสัมพันธ์ขึ้นมา และความสัมพันธ์อาจสร้างขึ้นมาจาก แอททริบิวต์คีย์ของเอนทิตีที่มีความสัมพันธ์กัน โดยความสัมพันธ์นั้นอาจมีแอททริบิวต์ของตัวเองก็ได้ เช่น ความสัมพันธ์การลงทะเบียนแสดงความสัมพันธ์ระหว่างเอนทิตีนักศึกษาและเอนทิตีชุดวิชา รวมทั้งผลการสอบในแต่ละชุดวิชาของนักศึกษาแต่ละคน เป็นต้น

รูปที่ 3.9 ตัวอย่างความสัมพันธ์ระหว่างเอนทิตีนักศึกษาและเอนทิตีชุดวิชาที่มีแอททริบิวต์

ความสัมพันธ์ระหว่างเอนทิตีใด ๆ อาจมีมากกว่าหนึ่งความสัมพันธ์ก็ได้ เช่น ความสัมพันธ์การสอนและความสัมพันธ์การจัดการแสดงความสัมพันธ์ระหว่างเอนทิตีอาจารย์และเอนทิตีชุดวิชา เป็นต้น



รูปที่ 3.10 ตัวอย่างความสัมพันธ์มากกว่าหนึ่งความสัมพันธ์

เอนทิตีที่เกี่ยวข้องกับความสัมพันธ์เรียกว่า การส่วนรวม (Participant) ของความสัมพันธ์ และจำนวนของมีส่วนรวมในความสัมพันธ์เรียกว่า ระดับ (Degree) ของความสัมพันธ์ ทั้งนี้ เอนทิตีซึ่งเป็น มีส่วนรวมของความสัมพันธ์อาจมีส่วนร่วมในความสัมพันธ์ที่สามารถจำแนกได้ 2 ลักษณะ คือ แบบการมีส่วนรวมทั้งหมด (Total Participation) และแบบการมีส่วนรวมบางส่วน (Partial Participation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบการมีส่วนรวมทั้งหมดเป็นความสัมพันธ์ที่ทุกสมาชิกในเอนทีตีหนึ่งจะมีข้อมูลในแอททริบิวต์หนึ่ง ที่มีความสัมพันธ์กับข้อมูลในอีกหนึ่งเอนทิตี เช่น อาจารย์ทุกคนต้องสังกัดคณะใดคณะหนึ่งเท่านั้น ดังนั้น แต่ละสมาชิกในเอนทิตีอาจารย์จะมีความสัมพันธ์กับเอนทิตีคณะ เป็นต้น

ในอี-อาร์ไดอะแกรม การระบุความสัมพันธ์แบบการมีส่วนรวมทั้งหมด ใช้สัญลักษณ์เส้นคู่ เพื่อเชื่อมต่อระหว่าง ความสัมพันธ์กับเอนทิตีที่ทุกสมาชิกมีความสัมพันธ์กับอีกเอนทิตีหนึ่ง

รูปที่ 3.11 ตัวอย่างความสัมพันธ์แบบการมีส่วนรวมทั้งหมด

ความสัมพันธ์แบบการมีส่วนรวมบางส่วน เป็นความสัมพันธ์ที่บางสมาชิกในเอนทีตีหนึ่งเท่านั้นจะมีข้อมูลในแอททริบิวต์ หนึ่งที่มีความสัมพันธ์กับข้อมูลในอีกหนึ่งเอนทิตี เช่น มีนักศึกษาเพียงบางคนเท่านั้นที่เป็นผู้แทนนักศึกษาในแต่ละคณะ ดังนั้น จะมีเพียงบางสมาชิกในเอนทิตีนักศึกษาเท่านั้นที่มีความสัมพันธ์กับเอนทิตีคณะ เป็นต้น

ในอี-อาร์ไดอะแกรม การระบุความสัมพันธ์แบบการมีส่วนรวมบางส่วน ใช้สัญลักษณ์เช่นเดียวกับการแสดงความสัมพันธ์ระหว่างเอนทิตีใด ๆ ที่เกี่ยวข้อง

รูปที่ 3.12 ตัวอย่างความสัมพันธ์แบบการมีส่วนรวมบางส่วน

ความสัมพันธ์มีหลายประเภทขึ้นอยู่กับปัจจัยที่ใช้ในการจำแนกประเภทของความสัมพันธ์ ซึ่งมี 2 ปัจจัย คือ ความสัมพันธ์ระหว่างเอนทิตี และจำนวนเอนทิตีที่เกี่ยวข้อง

3.2.3.1 ความสัมพันธ์ระหว่างเอนทิตี

การจำแนกประเภทของความสัมพันธ์ตามความสัมพันธ์ระหว่างเอนทิตี เป็นการพิจารณาถึงสัดส่วนความสัมพันธ์ระหว่างสมาชิกในเอนทิตีที่เป็นมีส่วนรวมของความสัมพันธ์ซึ่งอาจเรียกอีกอย่าง หนึ่งว่า ค่าความสัมพันธ์ (Cardinality Ratio) วิธีนี้สามารถจำแนกความสัมพันธ์ได้เป็น 3 ประเภท คือ ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (one to one relationship) ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (one to many relationship) และความสัมพันธ์แบบกลุ่มต่อกลุ่ม (many to many relationship)

ความสัมพันธ์แบบหนึ่งต่อหนึ่ง หมายถึง ความสัมพันธ์ที่แต่ละสมาชิกในเอนทิตีหนึ่งมีความสัมพันธ์กับสมาชิกในอีกหนึ่งเอนทิตีเพียงสมาชิกเดียว หรือกล่าวได้ว่า ความสัมพันธ์ดังกล่าวเป็นแบบหนึ่งต่อหนึ่ง เช่น เอนทิตีอาจารย์และเอนทิตีคณะ มีความสัมพันธ์กันแบบหนึ่งต่อหนึ่ง กล่าวคือ แต่ละคณะมีคณบดีเพียงหนึ่งคนเท่านั้น และมีอาจารย์เพียงหนึ่งคนเท่านั้นที่เป็นคณบดี เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในอี-อาร์ไดอะแกรมใช้สัญลักษณ์ 1:1 กำกับเหนือเส้นที่เชื่อมต่อระหว่างความสัมพันธ์ และเอนทิตีที่เกี่ยวข้องกับความสัมพันธ์นั้น

รูปที่ 3.13 ตัวอย่างความสัมพันธ์แบบหนึ่งต่อหนึ่ง

ความสัมพันธ์แบบหนึ่งต่อกลุ่ม หมายถึง ความสัมพันธ์ที่แต่ละสมาชิกในเอนทิตีหนึ่งมี ความสัมพันธ์กับสมาชิกในอีกหนึ่งเอนทิตีมากกว่าหนึ่งสมาชิก หรือกล่าวได้ว่า ความสัมพันธ์ดังกล่าวเป็นแบบหนึ่งต่อกลุ่ม เช่น เอนทิตีคณะและเอนทิตีนักศึกษามีความสัมพันธ์กันแบบหนึ่งต่อกลุ่ม กล่าวคือนักศึกษาแต่ละคนมีสังกัดเพียงคณะเดียว และหนึ่งคณะอาจมีนักศึกษาในสังกัดได้หลายคน

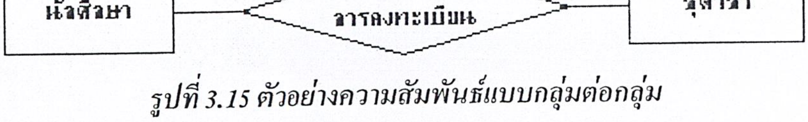
ในอี-อาร์ไดอะแกรมใช้สัญลักษณ์ 1:M กำกับเหนือเส้นที่เชื่อมต่อระหว่างความสัมพันธ์และ เอนทิตีที่เกี่ยวข้องกับความสัมพันธ์ นั้น

รูปที่ 3.14 ตัวอย่างความสัมพันธ์แบบหนึ่งต่อกลุ่ม

จากรูปที่ 3.14 เอนทิตีคณะและเอนทิตีนักศึกษามีความสัมพันธ์กันแบบหนึ่งต่อกลุ่ม และเป็นความสัมพันธ์แบบมีส่วนรวมทั้งหมด

ความสัมพันธ์แบบกลุ่มต่อกลุ่ม หมายถึง ความสัมพันธ์ที่สมาชิกมากกว่าหนึ่งสมาชิก ใน เอนทิตีหนึ่งมีความสัมพันธ์กับสมาชิกในอีกหนึ่งเอนทิตีมากกว่าหนึ่งสมาชิก หรือกล่าวได้ว่า ความสัมพันธ์ ดังกล่าวเป็นแบบกลุ่มต่อกลุ่ม เช่น เอนทิตีนักศึกษาและเอนทิตีชุดวิชามีความสัมพันธ์กันแบบกลุ่มต่อกลุ่ม กล่าวคือนักศึกษาแต่ละคนสามารถลงทะเบียนเรียนได้หลายชุดวิชา และแต่ละชุดวิชาสามารถมีนักศึกษาลงทะเบียนเรียนได้หลายคน

ในอี-อาร์ไดอะแกรม ใช้สัญลักษณ์ M:M กำกับเหนือเส้นที่เชื่อมต่อระหว่างความสัมพันธ์ และเอนทิตีที่เกี่ยวข้องกับความสัมพันธ์ นั้น



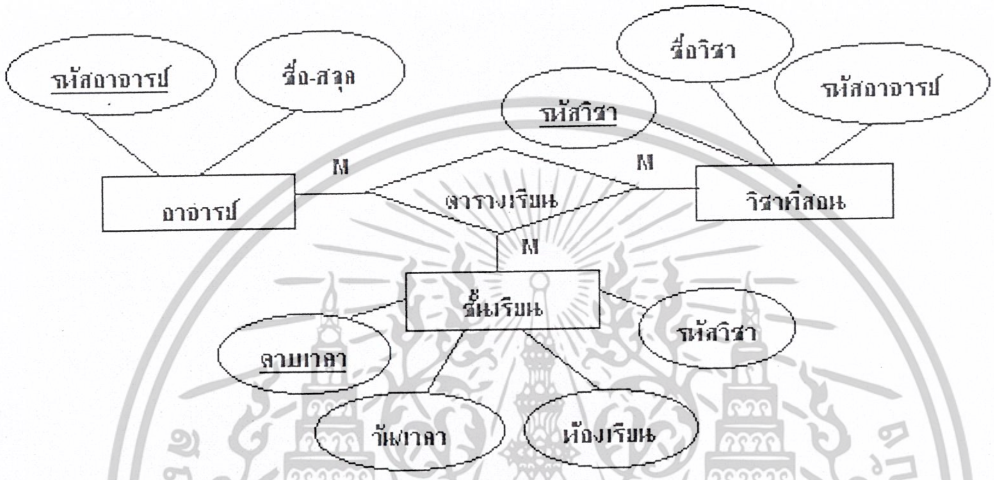
รูปที่ 3.15 ตัวอย่างความสัมพันธ์แบบกลุ่มต่อกลุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3.2 จำนวนเอนทิตีที่เกี่ยวข้อง

การจำแนกประเภทของความสัมพันธ์ตามจำนวนเอนทิตีที่เกี่ยวข้อง เป็นการพิจารณาถึงจำนวนของกามีส่วนรวมใน ความสัมพันธ์ หรือ ระดับของ ความสัมพันธ์ วิธีนี้สามารถจำแนกความสัมพันธ์ ได้ 2 ประเภท คือ ความสัมพันธ์แบบ ไบนารี่ (Binary) และ ความสัมพันธ์แบบ เอนนารี่ (N-ary) ความสัมพันธ์แบบ Binary เป็นความสัมพันธ์ที่พบได้บ่อยที่สุด โดยเป็นความสัมพันธ์ที่แสดงความสัมพันธ์ระหว่างสองเอนทิตีใด ๆ เช่น ความสัมพันธ์ที่ปรึกษาแสดงความสัมพันธ์ระหว่างเอนทิตีอาจารย์ที่ปรึกษากับเอนทิตีนักศึกษา

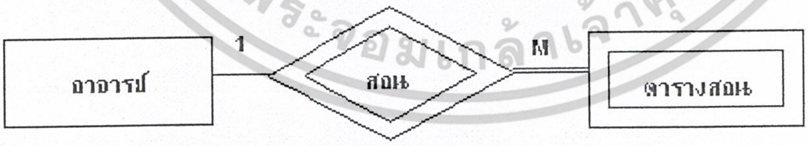
ความสัมพันธ์แบบ N–ary เป็นความสัมพันธ์ที่แสดงความสัมพันธ์ระหว่างสองเอนทิตี ขึ้นไป เช่น ความสัมพันธ์ตารางเรียนแสดงความสัมพันธ์ระหว่างเอนทิตีอาจารย์ เอนทิตีชุดวิชาที่สอน และเอนทิตี ชั้นเรียน



รูปที่ 3.16 ตัวอย่างความสัมพันธ์แบบเอน-นารี่

นอกจากประเภทของความสัมพันธ์ข้างต้น ยังมีลักษณะอื่น ๆ ที่เกี่ยวข้องกับความสัมพันธ์อีก คือ ความสัมพันธ์แบบขึ้นอยู่ (Dependency) ความสัมพันธ์แบบรีเคอซีฟ (Recursive)

ในอี-อาร์ไดอะแกรม สัญลักษณ์ที่ใช้แทนความสัมพันธ์แบบขึ้นอยู่ จะเป็นรูปสี่เหลี่ยมข้าวหลามตัดสองรูปซ้อนกัน



รูปที่ 3.17 ตัวอย่างความสัมพันธ์แบบขึ้นอยู่

จากภาพที่ 3.17 เอนทิตีอาจารย์และเอนทิตีตารางสอนมีความสัมพันธ์กันแบบหนึ่งต่อกลุ่มและ เป็นความสัมพันธ์แบบมีส่วนรวมทั้งหมด

ความสัมพันธ์แบบรีเคอซีพ เป็นความสัมพันธ์ที่เกิดจากเอนทิตีเพียงเอนทิตีเดียวเช่น ในห้องเรียนมีนักศึกษาหลายคน แต่มีนักศึกษาเพียงหนึ่งคนเท่านั้นที่เป็นหัวหน้าห้อง และหัวหน้าห้องเป็นผู้ประสานงานกับนักศึกษาหลายคน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.18 ตัวอย่างความสัมพันธ์แบบรีเคอซีพ

3.2.3.4. ประเภทหลัก และ ประเภทรอง (Supertype and subtype) หากเอนทิตีใดสามารถจำแนกออกเป็นเอนทิตีย่อย ๆ ได้ โดยแต่ละสมาชิกในเอนทิตีย่อยนั้นสามารถ มีคุณสมบัติแบ่งแยกได้ด้วยตัวเอง เอนทิตีหลักที่ถูกนำมาจำแนกจะเรียกว่า ประเภทหลัก ส่วนเอนทิตีย่อยที่จำแนกออกมาจะเรียกว่า ประเภทรองหรืออีกนัยหนึ่งอาจกล่าวได้ว่า เอนทิตีหนึ่งจะเป็นประเภทรอง ของอีกเอนทิตีหนึ่งที่เป็นประเภทหลักก็ต่อเมื่อ ประเภทรอง ประกอบด้วยทุกแอททริบิวต์ที่มีในประเภทหลัก โดยประเภทรอง สามารถมี แอททริบิวต์เพิ่มเติมจาก ประเภทหลักได้ ตัวอย่างเช่น เอนทิตีนักศึกษาประกอบด้วย 4 แอททริบิวต์ คือ รหัสประจำตัว ชื่อ-สกุล เพศ และรหัส คณะ โดยมีแอททริบิวต์รหัสประจำตัวเป็นแบ่งแยก ส่วนเอนทิตีนักศึกษาภาคปกติซึ่งเป็นประเภทรองของประเภทหลักเอนทิตีนักศึกษาอาจประกอบด้วย 6 แอททริบิวต์คือ รหัสประจำตัว ชื่อ-สกุล เพศ รหัสคณะ ผู้สนับสนุนการศึกษา และจำนวนเงินที่ได้รับต่อเดือน โดยมี แอททริบิวต์รหัสประจำตัวเป็นตัวแบ่งแยก และเอนทิตีนักศึกษาภาคพิเศษซึ่งเป็นประเภทรองของประเภทหลักเอนทิตีนักศึกษาอาจประกอบด้วย แอททริบิวต์คือ รหัสประจำตัว ชื่อ-สกุล เพศ รหัสคณะ ตำแหน่งงาน สถานที่ทำงาน และเงินเดือน โดยมี แอททริบิวต์รหัส ประจำตัวเป็นตัวแบ่งแยก เป็นต้น

3.3 การออกแบบฐานข้อมูลด้วยแบบจำลองข้อมูลอี-อาร์

ในการออกแบบฐานข้อมูลจำเป็นต้องการศึกษาคุณสมบัติและความสัมพันธ์ระหว่างข้อมูลที่มีอยู่ในระบบเพื่อให้ได้มาซึ่งโครงสร้างของฐานข้อมูล ซึ่งโดยทั่วไปมักดำเนินการ โดยการใช้แบบจำลองข้อมูล

อี-อาร์โมเดลเป็นแบบจำลองข้อมูลที่ได้รับความนิยมมากในการใช้เป็นเครื่องมือสำหรับงานออกแบบฐานข้อมูล โดยอี-อาร์โมเดลจะเสนอโครงสร้างของฐานข้อมูลในระดับแนวคิดออกมาในรูปของแผนภาพที่มีโครงสร้างง่ายต่อการทำความเข้าใจ ทำให้เห็นภาพรวมของเอนทิตีทั้งหมดและความสัมพันธ์ระหว่างเอนทิตีในระบบฐานข้อมูล

การออกแบบฐานข้อมูลด้วยแบบจำลองข้อมูลอี-อาร์ประกอบด้วย 3 ขั้นตอน คือ

3.3.1. การศึกษารายละเอียดและลักษณะหน้าที่งานของระบบ เป็นการศึกษารายละเอียดและลักษณะหน้าที่งานของระบบเป็นการศึกษาและรวบรวมรายละเอียด เกี่ยวกับลักษณะหน้าที่งานของระบบ ข้อมูลที่เกี่ยวข้อง ขั้นตอนในการทำงาน ตลอดจนข้อกำหนดและสมมติฐานต่าง ๆ ซึ่งอาจทำได้ด้วยการสัมภาษณ์หรือศึกษาจากแบบฟอร์มต่าง ๆ ที่มีการใช้งานอยู่ในระบบงานขณะนั้น

3.3.2. การกำหนดเอนทิตีที่ควรมีในระบบฐานข้อมูล เนื่องจากฐานข้อมูลหนึ่ง ๆ อาจประกอบด้วยเอนทิตีต่าง ๆ ได้จำนวนมาก ดังนั้น ขั้นตอนนี้จึงเป็นการนำรายละเอียดในข้อ 1 มาทำการกำหนดเอนทิตีที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำเป็นต้องมีอยู่ในระบบฐานข้อมูล โดยคำนึงถึงการเป็นเอนทิตีประเภทอ่อนแอ ตลอดจนประเภทหลักหรือประเภทรองด้วย

3.3.3. การกำหนดความสัมพันธ์ระหว่างเอนทิตี เป็นการกำหนดประเภทของความสัมพันธ์ระหว่างเอนทิตี โดยพิจารณาจากข้อกำหนดและสมมติฐานต่าง ๆ ที่ได้ทำการศึกษามาในข้อ 1

3.3.4. การกำหนดคุณลักษณะของเอนทิตี เป็นการกำหนดว่า แต่ละเอนทิตีควรประกอบด้วย แอททริบิวต์ใดบ้าง แอททริบิวต์ใดที่มีคุณสมบัติเป็นแอททริบิวต์คีย์ แอททริบิวต์ผสม แอททริบิวต์สืบถอด

3.3.5. การกำหนดคีย์หลักของแต่ละเอนทิตี เป็นการกำหนดแอททริบิวต์คีย์ของแต่ละเอนทิตีเพื่อให้แต่ละสมาชิกในเอนทิตีสามารถมีคุณสมบัติเป็นเอกลักษณ์เฉพาะได้

3.3.6. การนำสัญลักษณ์ที่ใช้ในอี-อาร์โมเดลมาอธิบายความสัมพันธ์ระหว่างข้อมูล เป็นการนำรายละเอียดในขั้นตอนต่าง ๆ มาพิจารณาทบทวนเพื่อเพิ่มหรือลดเอนทิตี แอททริบิวต์และความสัมพันธ์ต่าง ๆ จากนั้นจึงนำข้อมูลที่ได้จากขั้นตอนทั้งหมดมาเขียนเป็นแบบจำลองเพื่ออธิบายความสัมพันธ์ระหว่างข้อมูลด้วยสัญลักษณ์ต่าง ๆ หรืออี-อาร์ไดอะแกรม ดังนั้น แบบจำลองข้อมูลที่เกิดขึ้นจึงมีความชัดเจน สอดคล้อง ถูกต้อง และเหมาะสมกับองค์ประกอบของงานที่กำลังศึกษาทำให้เป็นที่ยอมรับของทุกฝ่ายที่เกี่ยวข้อง

3.4. ตัวอย่างการออกแบบฐานข้อมูลด้วยแบบจำลองข้อมูลอี-อาร์

เพื่อแสดงให้เห็นถึงขั้นตอนการออกแบบฐานข้อมูลด้วยแบบจำลองข้อมูลอี-อาร์ ในที่นี้ขอนำตัวอย่างภาระงานสอนของอาจารย์ในสถาบันการศึกษาแห่งหนึ่งมาประกอบดังนี้

3.4.1. การศึกษารายละเอียดและลักษณะหน้าที่งานของระบบ

สถาบันการศึกษาแห่งหนึ่งเปิดทำการสอนโดยแบ่งเป็น 3 คณะ คือ คณะเกษตร คณะวิทยาศาสตร์ และคณะศึกษาศาสตร์ ในแต่ละคณะประกอบด้วยภาควิชาต่าง ๆ และในแต่ละภาควิชาประกอบด้วยอาจารย์ได้หลายคน โดยอาจารย์แต่ละคนจะสังกัดภาควิชาใดภาควิชาหนึ่งเท่านั้น และมีอาจารย์เพียงคนเดียวเท่านั้นที่อยู่ในตำแหน่งหัวหน้าภาควิชาทำหน้าที่บริหารงานภาควิชา โดยในแต่ละภาควิชาจะเปิดทำการสอนได้หลายชุดวิชา ซึ่งอาจารย์แต่ละคนจะต้องสอนไม่น้อยกว่า 2 ชุดวิชา

3.4.2. การกำหนดเอนทิตีที่ควรมีในระบบฐานข้อมูล

- เอนทิตีคณะเป็นเอนทิตีที่แสดงรายละเอียดของแต่ละคณะ
- เอนทิตีภาควิชาเป็นเอนทิตีที่แสดงรายละเอียดของแต่ละภาควิชา
- เอนทิตีอาจารย์เป็นเอนทิตีที่แสดงรายละเอียดของอาจารย์แต่ละคน
- เอนทิตีชุดวิชาเป็นเอนทิตีที่แสดงรายละเอียดของแต่ละชุดวิชา

3.4.3. การกำหนดความสัมพันธ์ระหว่างเอนทิตี

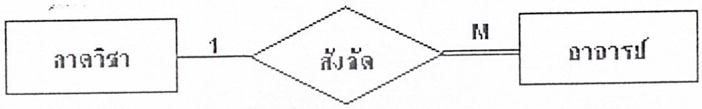
ความสัมพันธ์สังกัด เป็นความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีคณะและเอนทิตีภาควิชา เนื่องจากในแต่ละคณะสามารถมีได้หลายภาควิชา และข้อมูลในเอนทิตีภาควิชาจะมีความสัมพันธ์กับเอนทิตีคณะแบบมีส่วนรวมทั้งหมดเนื่องจากแต่ละภาควิชาจะต้องสังกัดคณะใดคณะหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

รูปที่ 3.19 ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีคณะและเอนทิตีภาควิชา

ความสัมพันธ์สังกัด เป็นความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีภาควิชาและเอนทิตีอาจารย์ เนื่องจากในแต่ละภาควิชาสามารถมีอาจารย์ได้หลายคน และข้อมูลในเอนทิตีอาจารย์จะมี ความสัมพันธ์กับเอนทิตีภาควิชาแบบมีส่วนรวมทั้งหมด เนื่องจากอาจารย์แต่ละคนจะต้องสังกัดภาควิชาใดภาควิชาหนึ่ง

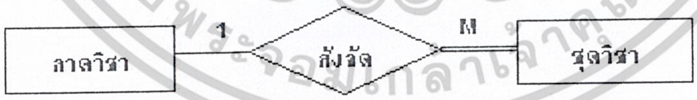


รูปที่ 3.20 ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีภาควิชาและเอนทิตีอาจารย์

ความสัมพันธ์หัวหน้าภาค เป็นความสัมพันธ์แบบรีเคอซีพเนื่องจากในหนึ่งภาควิชามีอาจารย์ได้หลายคน แต่มีอาจารย์เพียงหนึ่งคนเท่านั้นที่ทำหน้าที่หัวหน้าภาควิชา

รูปที่ 3.21 ความสัมพันธ์แบบรีเคอซีพของเอนทิตีอาจารย์

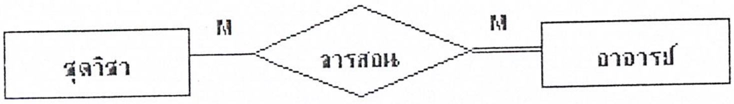
ความสัมพันธ์สังกัด เป็นความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีภาควิชาและเอนทิตี ชุดวิชา เนื่องจากในแต่ละภาควิชาสามารถเปิดสอนได้หลายชุดวิชา และข้อมูลของเอนทิตีชุดวิชามี ความสัมพันธ์กับเอนทิตีภาควิชาแบบมีส่วนรวมทั้งหมดเนื่องจากแต่ละชุดวิชาจะต้องสังกัดภาควิชาใด ภาควิชาหนึ่ง



รูปที่ 3.22 ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนทิตีภาควิชาและเอนทิตีชุดวิชา

ความสัมพันธ์การสอน เป็นความสัมพันธ์แบบกลุ่มต่อกลุ่มระหว่างเอนทิตีชุดวิชาและเอนทิตีอาจารย์ โดยข้อมูลของเอนทิตีอาจารย์มีความสัมพันธ์กับเอนทิตีชุดวิชาแบบส่วนรวมทั้งหมดเนื่องจากในแต่ละชุดวิชาสามารถมีอาจารย์ผู้สอนได้หลายคน และอาจารย์แต่ละคนจะต้องสอนอย่างน้อย 2 ชุดวิชา

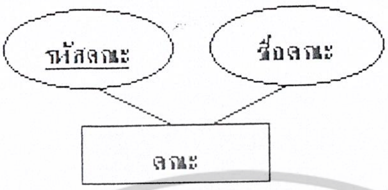
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
55100



รูปที่ 3.23 ความสัมพันธ์แบบกลุ่มต่อกลุ่มระหว่างเอนทิตีชุดวิชาและเอนทิตีอาจารย์

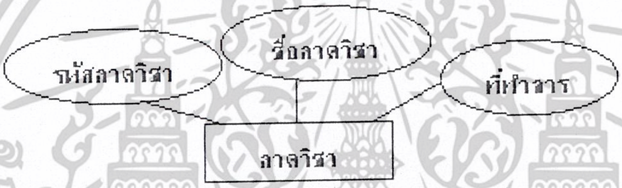
3.4.4. การกำหนดคุณลักษณะของเอนทิตี

- เอนทิตีคณะ ประกอบด้วยแอททริบิวต์รหัสคณะ ชื่อคณะ



รูปที่ 3.24 เอนทิตีคณะ

- เอนทิตีภาควิชา ประกอบด้วยแอททริบิวต์ รหัสภาควิชา ชื่อภาควิชา ที่ทำการ

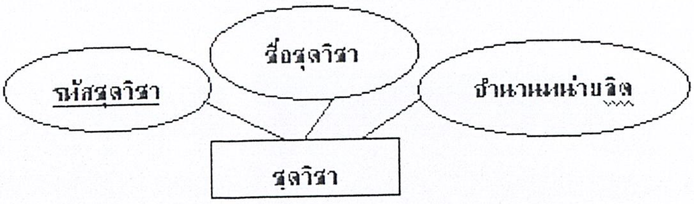


รูปที่ 3.25 เอนทิตีภาควิชา

- เอนทิตีอาจารย์ ประกอบด้วยแอททริบิวต์ รหัสอาจารย์ ชื่ออาจารย์ วันที่เริ่มทำงาน เงินเดือน

รูปที่ 3.26 เอนทิตีอาจารย์

- เอนทิตีชุดวิชา ประกอบด้วยแอททริบิวต์รหัสชุดวิชา ชื่อชุดวิชา จำนวนหน่วยกิต



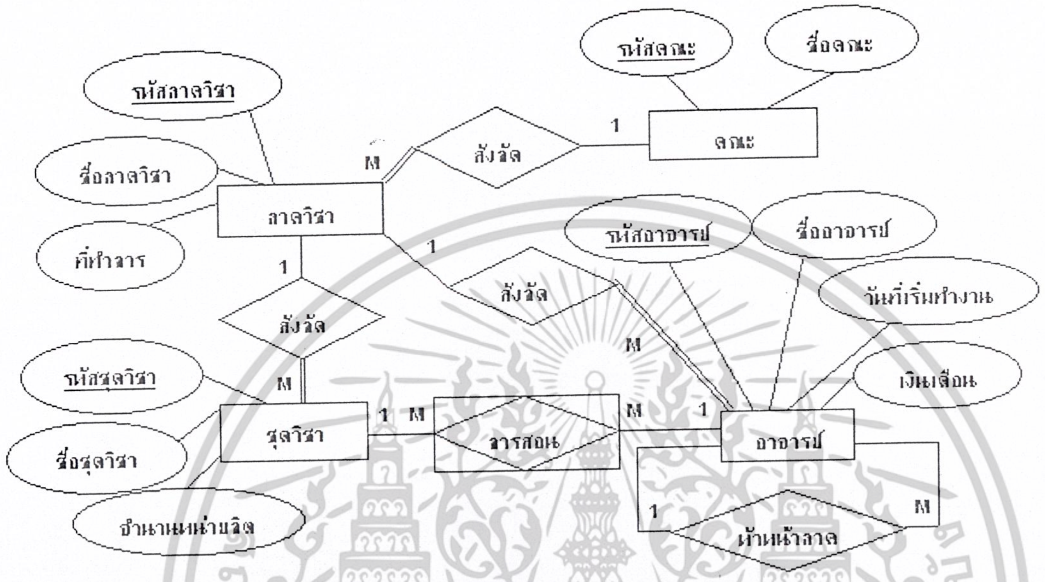
รูปที่ 3.27 เอนทิตีชุดวิชา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5. การกำหนดคีย์หลักของแต่ละเอนทิตี

- เอนทิตีคณะ มีรหัสคณะเป็นคีย์หลัก
- เอนทิตีภาควิชา มีรหัสภาควิชาเป็นคีย์หลัก
- เอนทิตีอาจารย์ มีรหัสอาจารย์เป็นคีย์หลัก
- เอนทิตีชุดวิชา มีรหัสชุดวิชาเป็นคีย์หลัก

3.4.6. การนำสัญลักษณ์ที่ใช้ในอี-อาร์โมเดล มาอธิบายความสัมพันธ์ระหว่างข้อมูล



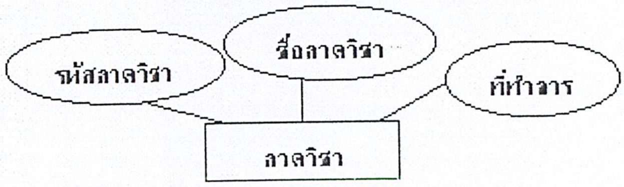
รูปที่ 3.28 อี-อาร์ไดอะแกรมภาระงานสอนของอาจารย์

3.5 การแปลงฐานข้อมูลที่ออกแบบด้วยอี-อาร์โมเดลเป็นฐานข้อมูลเชิงสัมพันธ์

จากแบบจำลองข้อมูลอี-อาร์เป็นการออกแบบฐานข้อมูลในระดับแนวคิดซึ่งจะทำให้ทราบว่า ในฐานข้อมูลนั้นประกอบด้วยเอนทิตีและรายละเอียดอะไรบ้าง ดังนั้น เพื่อให้สามารถใช้งานฐานข้อมูลต่อไปได้ จึงต้องทำการแปลงความสัมพันธ์และอี-อาร์โมเดลให้อยู่ในรูปแบบของแบบจำลองอื่นที่สอดคล้องกับระบบจัดการฐานข้อมูลที่เลือกใช้ ซึ่งในที่นี้จะมุ่งเน้นให้ความสนใจกับฐานข้อมูลเชิงสัมพันธ์เท่านั้น

การแปลงฐานข้อมูลที่ออกแบบด้วยอี-อาร์โมเดลเป็นฐานข้อมูลเชิงสัมพันธ์ ประกอบด้วย 7 ขั้นตอน ดังนี้คือ

3.5.1. ทุกเอนทิตีไทป์ใน อี-อาร์ จะสามารถสร้างรีเลชั่นที่ประกอบด้วยแอททริบิวต์อย่างง่ายหรือแอททริบิวต์ผสม และเลือกแอททริบิวต์คีย์มา เพื่อสร้างเป็นคีย์หลักสำหรับรีเลชั่น

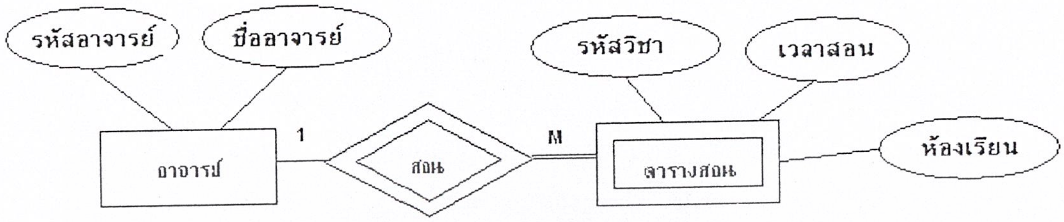


รูปที่ 3.29 เอนทิตี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ภาควิชา (รหัสภาควิชา, ชื่อภาควิชา, ที่ทำการ)

3.5.2.ทุกเอนทิตีไทป์อ่อนแอ จะนำมาสร้างเป็นรีเลชั่น โดยประกอบด้วยแอททริบิวต์อย่างง่ายหรือแอททริบิวต์ผสม และนำคีย์หลักของเอนทิตีไทป์หลักมารวมกับแอททริบิวต์คีย์ของเอนทิตีไทป์อ่อนแอเพื่อนำมาสร้างเป็นคีย์หลักในรีเลชั่น



รูปที่ 3.30 เอนทิติอ่อนแอ

- อาจารย์ (<u>รหัสอาจารย์</u>, ชื่ออาจารย์)

- ตารางสอน (<u>รหัสอาจารย์, รหัสวิชา</u>, เวลาสอน, ห้องเรียน)

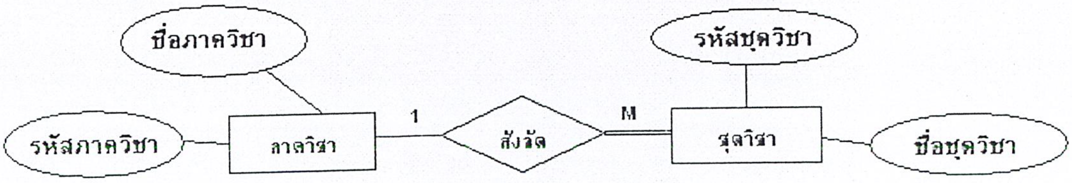
3.5.3. ความสัมพันธ์แบบ 1:1 ซึ่งเป็นความสัมพันธ์กับสองเอนทิตี โดยจะทำการเลือกรีเลชั่นที่สร้างจากเอนทิตีหนึ่งโดยนำคีย์หลักของเอนทิตีไปใส่เป็น คีย์นอกในอีกรีเลชั่นหนึ่ง เพื่อแสดงความสัมพันธ์แบบ 1:1

รูปที่ 3.31 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง

- อาจารย์ (<u>รหัสอาจารย์</u>, ชื่ออาจารย์)

- คณะ (<u>รหัสคณะ</u>, ชื่อคณะ, *รหัสอาจารย์*)

3.5.4. สำหรับความสัมพันธ์แบบ 1:M (ที่ไม่ใช้ความสัมพันธ์แบบอ่อนแอ) ซึ่งเป็นความสัมพันธ์กับสองเอนทิตี โดยจะทำการเลือกรีเลชั่นที่สร้างจากเอนทิตีมีค่าความสัมพันธ์เท่ากับ 1 นำคีย์หลักมาทำเป็นคีย์นอกกับรีเลชั่นที่สร้างจากเอนทิตีที่มีค่าความสัมพันธ์เท่ากับ M



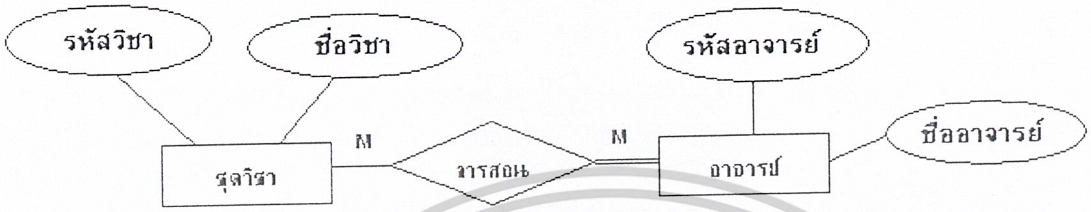
รูปที่ 3.32 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ภาควิชา (<u>รหัสภาควิชา</u>, ชื่อภาควิชา)

- ชุดวิชา (<u>รหัสชุดวิชา</u>, ชื่อชุดวิชา, *รหัสภาควิชา*)

3.5.5. สำหรับความสัมพันธ์แบบ M:N จะทำการสร้างรีเลชั่นใหม่ โดยนำคีย์จากรีเลชั่นจากเอนทิตีที่มีความสัมพันธ์กันมารวมกันสร้างเป็นคีย์หลักให้กับรีเลชั่นใหม่ โดยนำแอททริบิวต์ของความสัมพันธ์มาใส่ไว้ในรีเลชั่น



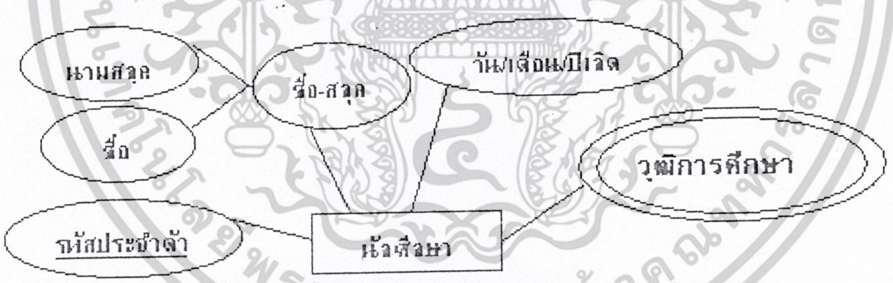
รูปที่ 3.33 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม

- ชุดวิชา (<u>รหัสวิชา</u>, ชื่อวิชา)

- อาจารย์ (<u>รหัสอาจารย์</u>, ชื่ออาจารย์)

- การสอน (<u>รหัสวิชา, รหัสอาจารย์</u>)

3.5.6. ทุกๆแอททริบิวต์หลายค่า จำเป็นที่จะต้องสร้างรีเลชั่นใหม่โดยนำ คีย์หลักของเอนทิตีที่แอททริบิวต์อยู่ นำมารวมกับแอททริบิวต์หลายค่า นำมาสร้างเป็นคีย์หลักในรีเลชั่น



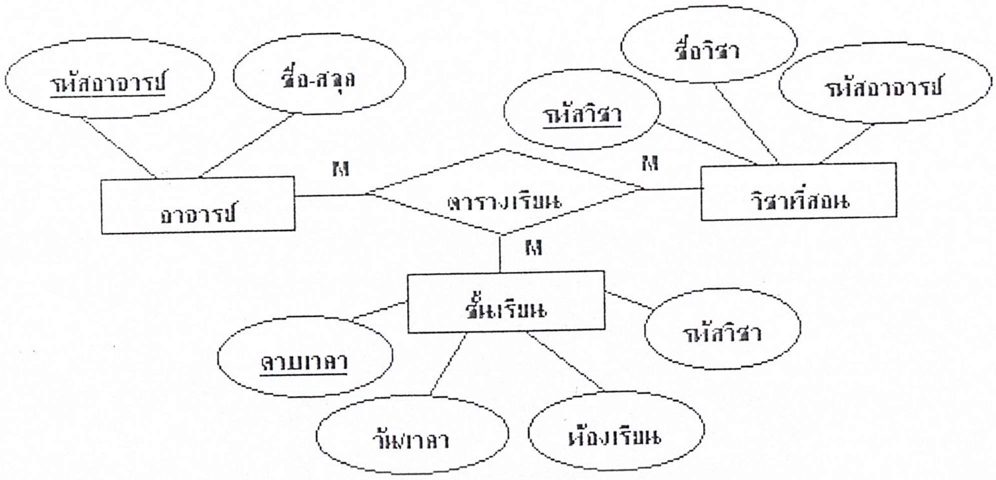
รูปที่ 3.34 แอททริบิวต์หลายค่า

- นักศึกษา (<u>รหัสประจำตัว</u>, ชื่อ, นาสกุล, วันเดือนปี)

- การศึกษา (<u>รหัสประจำตัว, วุฒิการศึกษา</u>)

7. สำหรับความสัมพันธ์แบบเอน-นารี่ ซึ่งมีเอน ทิตีมากกว่า 2 เอนทิตี ซึ่งจะมีรีเลชั่นที่มีความสัมพันธ์มากกว่า 2 รีเลชั่น โดยจะทำการสร้างรีเลชั่นใหม่จากความสัมพันธ์แบบเอน-นารี่ จากนั้นนำคีย์หลักของรีเลชั่นที่เกี่ยวข้องกับความสัมพันธ์มารวมกันเป็นคีย์หลักของรีเลชั่นใหม่ โดยนำแอททริบิวต์ของความสัมพันธ์มาใส่ไว้ในรีเลชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.35 ความสัมพันธ์แบบ เอน-นารี่

- อาจารย์ (<u>รหัสอาจารย์</u>, ชื่อ-สกุล)
- วิชาที่สอน (<u>รหัสวิชา</u>, ชื่อวิชา, รหัสอาจารย์)
- ชั้นเรียน (<u>คาบเวลา</u>, วัน/เวลา, ห้องเรียน, รหัสวิชา)
- ตารางเรียน (<u>รหัสอาจารย์</u>, <u>รหัสวิชา</u>, <u>คาบเวลา</u>)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
แนวคิดเชิงวัตถุ

ในยุคต้น ภาษาที่ใช้สั่งการเป็นภาษาที่สั่งให้คอมพิวเตอร์ทำงานเป็นลำดับของคำสั่ง งานประยุกต์จึงใช้ภาษาเชิงลำดับสั่งงาน ต่อมาเมื่อพบว่างานที่สั่งให้คอมพิวเตอร์ทำมีความซับซ้อนมากขึ้น การสั่งงานทำให้วิธีการเขียนโปรแกรมขึ้นอยู่กับตัวบุคคล เพราะลำดับความคิดที่ถ่ายออกมาเป็นรูปโปรแกรมนั้นยากที่จะทำความเข้าใจได้ โปรแกรมที่พัฒนาจึงขึ้นกับตัวบุคคล ไม่สามารถให้อีกบุคคลหนึ่งดำเนินการตรวจสอบหรือทำความเข้าใจเกี่ยวกับโปรแกรมได้ง่าย

จึงทำให้เกิดแนวความคิดการเขียนโปรแกรมเชิงวัตถุ (Object – Oriented Programming) ขึ้นมา โดยเป็นวิธีใหม่ในการพัฒนาซอฟต์แวร์ (Software) เพื่อเพิ่มประสิทธิภาพในการผลิตซอฟต์แวร์ รูปแบบเชิงวัตถุเป็นการแสดงให้เห็นองค์ประกอบสิ่งต่างๆ (object) ที่เราสนใจ ซึ่งจะมีข้อกำหนดที่กำหนดให้กับองค์ประกอบเหล่านั้น และแสดงความสัมพันธ์ระหว่างองค์ประกอบเหล่านั้น

4.1 หลักการของแนวความคิดเชิงวัตถุ จะประกอบไปด้วยองค์ประกอบเหล่านี้คือ

1. **Object and Object Identifier** หมายถึง ออปเจคใดๆ ที่อยู่ในรูปแบบที่กำหนด และมีการกำหนดค่า id ที่ไม่ซ้ำกัน เพื่อใช้ในการอ้างถึงออปเจคนั้น
2. **Attributes and Methods:** ออปเจคทุกๆ ออปเจคจะมีคุณลักษณะ(state) คือเป็นเซตของค่าข้อมูลของแอตทริบิวต์ของออปเจค และการกระทำ(behavior) เป็นเซตของเมธอด ส่วนของโปรแกรมซึ่งมีการดำเนินการบนลักษณะของออปเจค ซึ่งคุณลักษณะและการกระทำจะถูกห่อหุ้มไว้ในออปเจค และจะถูกอ้างหรือเรียกให้ทำงานจากภายนอก โดยผ่านทางข่าวสาร(message)
3. **Class** หมายถึง การรวมกลุ่มของออปเจคทั้งหมดที่มีลักษณะ และเมธอดที่เหมือนๆ กัน ออปเจคหนึ่งๆ จะถูกสร้างขึ้นจากคลาสเพียงหนึ่งคลาส ซึ่งคลาสมีลักษณะเหมือนกับ abstract data type คลาสอาจจะเป็นข้อมูลพื้นฐานก็ได้ เช่น integer, string, Boolean เป็นต้น
4. **Class Hierarchy and Inheritance** เป็นการสร้างคลาสใหม่(subclass) จากคลาสเดิม(superclass) ที่มีอยู่ก่อน คลาสที่สร้างขึ้นใหม่จะสืบทอดคุณลักษณะ และเมธอดทั้งหมดจาดคลาสเดิม โดยคลาสใหม่สามารถที่จะเพิ่มคุณลักษณะ และเมธอดใหม่ เพิ่มเข้าไปได้ จะแบ่งเป็นสองประเภทคือ Single Inheritance และ Multiple Inheritance

4.1.1.Object Structure

แนวทางเชิงวัตถุมีพื้นฐานมาจากการห่อหุ้มข้อมูล และโปรแกรม(Encapsulating) ดังนั้นการติดต่อกันระหว่างออปเจคจะทำผ่านทางข่าวสาร (message) เสมอ ซึ่งโดยทั่วไปแล้วออปเจคจะเกี่ยวข้องกับ

- เซตของตัวแปรที่บรรจุข้อมูลของออปเจค
- เซตของข่าวสาร ซึ่งออปเจคจะตอบสนองต่อข่าวสารที่ได้รับจากภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เซตของเมธอด ซึ่งแต่ละเมธอดจะเป็นโปรแกรมที่ดำเนินการกับข่าวสาร และส่งข้อมูลกลับ

สิ่งที่ทำให้ต้องมีการใช้ข่าวสาร กับเมธอดระหว่างออปเจคนั้น เนื่องจากในระบบเชิงวัตถุ เราสามารถที่จะแก้ไขนิยามของตัวแปร และเมธอด โดยไม่มีผลกระทบต่อระบบ ซึ่งคุณสมบัตินี้เป็นข้อดีของการเขียนโปรแกรมเชิงวัตถุ

4.1.2.Object Classes

โดยปกติ ในฐานข้อมูลจะมีออปเจคที่คล้ายๆ กัน หมายความว่า ออปเจคมีการตอบสนองต่อข่าวสารเดียวกัน โดยใช้เมธอดเดียวกัน และตัวแปรชื่อเดียวกัน ชนิดข้อมูลเดียวกัน ดังนั้นเราจึงจัดกลุ่มของออปเจคที่มีลักษณะคล้ายๆ กัน แล้วกำหนดเป็นคลาสๆ หนึ่ง และแต่ละออปเจคที่ถกสร้างมาจากคลาสก็เรียกว่า เป็น Instance ของคลาส ออปเจคทุกตัวจะมีลักษณะเหมือนกัน แต่จะแตกต่างกันตรงข้อมูลที่ถูกกำหนดให้กับตัวแปรในออปเจค

ตัวอย่าง เรานิยามคลาส Employee ในรูปแบบของคำสั่งลำลอง ซึ่งแสดงตัวแปรและเมธอด(ในตัวอย่างยังไม่มีการจัดการกับเมธอดของคลาส)

```
Class Employee {
    /* Variables */
    String name;
    String address;
    Date start_date;
    Integer salary;
    /* Methods */
    Integer annual_salary();
    String get_name();
    String get_address();
    Integer set_address(String new_address);
    Integer employment_length();
}
```

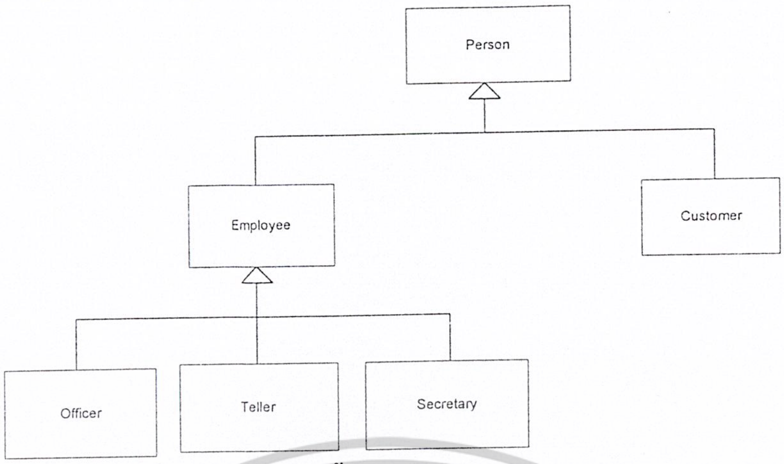
จากนิยามของคลาส Employee จะประกอบไปด้วยตัวแปร name และ address ซึ่งมีชนิดข้อมูลเป็น String start_date มีข้อมูลเป็น Date และ salary มีชนิดข้อมูลเป็น Integer แต่ละออปเจคจะตอบสนองต่อข่าวสาร 5 ข่าวสาร คือ annual_salary, get_name, get_address, set_address และ employment_length โดยเมธอดแต่ละอันมีการกำหนดชนิดข้อมูลไว้ด้านหน้า หมายถึงชนิดของการตอบสนองข้อมูลกลับไปยังเมธอดนั้น

4.1.3. Inheritance

ในระบบฐานข้อมูลเชิงวัตถุจะประกอบด้วยคลาสจำนวนมาก อย่างไรก็ตามคลาสหลายๆ คลาสจะมีลักษณะคล้ายๆ กัน ตัวอย่างเช่น พนักงานของธนาคาร จะมีลักษณะที่คล้ายกับลูกค้า เช่น ชื่อ ที่อยู่ โทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นต้น เพื่อแสดงให้เห็นถึงลักษณะที่คล้ายกันของคลาส เราจะแสดง E-R ไดอะแกรมในลักษณะของลำดับชั้นของ specialization (ความสัมพันธ์แบบ is a)



รูปที่ 4.1 แสดงลำดับชั้นของคลาสแบบ **Specialization**

ตัวอย่าง ลำดับชั้นของคลาสที่กำหนดเป็นคำสั่งลำลอง ได้ดังนี้

```
Class Person {
    String name;
    String address;
}
Class Customer isa Person {
    Integer credit_rating;
}
Class Employee isa Person {
    Date start_date;
    Integer salary;
}
Class Officer isa Employee {
    Integer officer_number;
    Integer expense_account_number;
}
Class Teller isa Employee {
    Integer hours_per_week;
    Integer station_number;
}
Class Secretary isa Employee {
    Integer hours_per_week;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

String manager;

}

คำว่า “is a” ในคำสั่งลำลองใช้สำหรับบอกว่าคลาสนั้นมีลักษณะเช่นเดียวกันกับอีกคลาสหนึ่ง โดยเราเรียกคลาสที่ specialization ว่า Subclass ตัวอย่างเช่น Employee เป็น Subclass ของ Person และ Teller เป็น Subclass ของ Employee ในทางกลับกัน Employee ก็เป็น Superclass ของ Teller

ข้อดีของการสืบทอดคุณสมบัติในระบบเชิงวัตถุคือ มีคุณลักษณะของ code-reuse นั่นคือเมธอดใดๆ ของคลาส A สามารถที่จะถูกเรียกใช้โดยคลาส B ซึ่งเป็น Subclass A ได้ทันที ทำให้ไม่ต้องเขียนคำสั่งในคลาส B อีก

4.1.4.Multiple Inheritance

วิธีการ Multiple inheritance เป็นวิธีการที่ยอมให้คลาสหนึ่งสืบทอดคุณสมบัติ และเมธทอดจากหลายๆ คลาสได้ แต่เมื่อเรามีการสืบทอดแบบหลายคลาส อาจจะมีความคลุมเครือเกิดขึ้นในกรณีที่ตัวแปรหรือเมธทอดที่สืบทอดมาจากหลายคลาสมีชื่อเดียวกัน ทำให้ Subclass เกิดความสับสนในการเรียกใช้งานตัวแปรหรือเมธทอดดังกล่าว

4.1.5. Object Identity

Object identity: ออปเจคจะยังคงรักษาความเป็น identity แม้ว่าข้อมูลของตัวแปรหรือเมธทอดของออปเจคมีการเปลี่ยนแปลง แนวคิดของ identity นี้ไม่ได้มีอยู่ในฐานข้อมูลแบบเชิงสัมพันธ์ ซึ่งในระบบฐานข้อมูลเชิงสัมพันธ์นั้น เรคอร์ดของเทเบิลจะแตกต่างกันด้วยข้อมูลที่ถูกเก็บอยู่

รูปแบบต่างๆ ของ identity

- **Value** หมายถึง ค่าของข้อมูลที่ใช้ในการ identity เช่น Primary key
- **Name** เป็นการกำหนดเพื่อใช้ในการ identity เช่น ชื่อแฟ้มในระบบแฟ้มข้อมูล
- **Built-in** เป็นการกำหนด identity โดยอัตโนมัติให้กับข้อมูล หรือภาษาโปรแกรม โดยผู้ใช้ไม่จำเป็นต้องกำหนด identifier รูปแบบนี้ถูกใช้กับระบบเชิงวัตถุ โดยแต่ละออปเจคจะถูกกำหนด identifier โดยอัตโนมัติ เมื่อออปเจคถูกสร้างขึ้น

ในทางปฏิบัติ Object identity จะถูกกำหนดให้มีค่าไม่ซ้ำกัน เรียกว่า OID ซึ่งค่าของ OID นี้จะไม่สามารถมองเห็นได้โดยผู้ใช้ทั่วไป แต่จะถูกใช้โดยระบบเพื่อระบุออปเจคแต่ละออปเจค

4.1.6. Encapsulation

วิธีการ Encapsulation เป็นวิธีการที่ช่วยปกป้องข้อมูลโดยไม่ให้ออปเจคอื่นเข้ามาทำหารเปลี่ยนแปลงข้อมูลก่อนได้รับอนุญาต โดยการเข้าถึงตัวแปรของออปเจคนั้นจะต้องผ่านทางเมธอดเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.7. Method Overriding and Polymorphism

วิธีการ Overriding เป็นวิธีการที่ Subclass สามารถสร้างเมธทอดที่มีชื่อเดียวกันกับเมธทอดที่สืบทอดมาจาก Superclass ได้ เพื่อทำให้ Subclass นั้นมีความเฉพาะเจาะจงมากขึ้น

การ Polymorphism เป็นการทำให้ออปเจคที่ต่างกันสามารถตอบสนองต่อข่าวสารเดียวกันได้หลายๆ วิธีการ โดยในระบบเชิงวัตถุคำว่า "Polymorphism" จะหมายถึง

1. เมธอดสามารถใช้ชื่อเดียวกันได้ในหลายๆ คลาส
2. ผู้ใช้ส่งข่าวสาร(message) เดียวกัน ไปยังออปเจคจากคลาสต่างๆ กัน ก็ยังได้ผลลัพธ์ที่ถูกต้อง

4.1.8. Abstract Class

Abstract Class เป็นวิธีการที่ช่วยลดความซับซ้อนของคุณสมบัติ โดยแยกเอาเฉพาะสิ่งที่ผู้ใช้งานสนใจออกมา โดยการแยกแยะเอกลักษณะเป็นการแสดงคุณลักษณะที่สำคัญของออปเจค ซึ่งเป็นคุณลักษณะที่ทำให้ออปเจคนั้นๆ แตกต่างจากออปเจคอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
ฐานข้อมูลเชิงวัตถุ

แนวคิดการเขียนโปรแกรมแบบออปเจคเป็นรูปแบบใหม่ในการจัดการกับด้วยการกำหนดสิ่งต่างๆ ให้ใกล้เคียงกับสิ่งต่างๆ ในโลกความเป็นจริงที่เรียกว่า ออปเจค แนวความคิดเชิงวัตถุเริ่มมีใช้ในภาษาโปรแกรมมาก่อน เช่น ภาษาจาวา(Java) และภาษาซีพลัสพลัส (C++) สำหรับแบบจำลองที่นำเสนอโครงสร้างข้อมูลในแนวทางออปเจคเรียกว่า แบบจำลองข้อมูลแบบออปเจค ฐานข้อมูลที่ใช้แบบจำลองข้อมูลซึ่งแสดงข้อมูล และความสัมพันธ์ระหว่างข้อมูลในแนวทางของออปเจค เรียกว่า ฐานข้อมูลเชิงวัตถุ(Object-Oriented Databases: OODB)

ในบทนี้ได้เปรียบเทียบถึงความแตกต่างระหว่างเทคโนโลยีแบบออปเจคกับเทคโนโลยีแบบรีเลชั่นแนวความคิดของฐานข้อมูลแบบเชิงสัมพันธ์ และแนวความคิดของฐานข้อมูลเชิงวัตถุ อยู่ในส่วนของ 1. และ 2. ตามลำดับ

5.1. แนวความคิดของฐานข้อมูลแบบเชิงสัมพันธ์ (RDB)

ฐานข้อมูลแบบเชิงสัมพันธ์จะนำเสนอเป็นรูปแบบของตาราง โดยประกอบด้วยแถวและคอลัมน์ ซึ่งเป็นมุมมองแบบลอจิกคัลของฐานข้อมูลเพื่อแสดงให้ผู้ใช้งานและโปรแกรมเมอร์ดู ส่วนข้อมูลที่เก็บอยู่ในระบบดิสคอมพิวเตอร์นั้นจะเรียกว่ามุมมองแบบฟิสิคัล ฟิลด์ในเรคอร์ดจะเรียกว่าแอททริบิวต์ แต่ละคอลัมน์จะมีชนิดของข้อมูล เช่น ข้อมูลชนิดตัวเลข ข้อมูลชนิดอักขระ ข้อมูลชนิดวันที่ เป็นต้น ซึ่งก่อนที่จะมีการเก็บข้อมูลลงฐานข้อมูลจะต้องมีการตรวจสอบความถูกต้องข้อมูลก่อนเรียกว่า เงื่อนไข มีทั้งเงื่อนไขแบบโดเมน, เงื่อนไขแบบคีย์, เงื่อนไขแบบเอนทิตีอินทีกริตี (Entity integrity) และเงื่อนไขแบบเรฟเฟอเรนเชียวอินทีกริตี (Referential integrity) ซึ่งทำให้แน่ใจได้ว่าจะไม่มีข้อมูลที่ผิดปกติอยู่ในฐานข้อมูล

ฐานข้อมูลแบบเชิงสัมพันธ์ใช้ภาษาเอสคิวแอล (Structure Query Language : SQL) เป็นภาษาดาต้าดิฟฟินิชั่น (Data Definition : DDL) หรือภาษาดาต้าแมนิพพูเลชั่น (Data Manipulation : DML) ซึ่งเอสคิวแอลนั้นจะรวมคำสั่งที่ใช้ในการกำหนดชนิดของข้อมูล, คำสั่งที่ใช้ในการเปลี่ยนแปลงข้อมูล, คำสั่งคิวรี่ และคำสั่งที่ใช้กำหนดเงื่อนไข การคิวรี่นั้นจะแบ่งตามลักษณะของการคิวรี่ ซึ่งเป็นมาตรฐานเดียวกันของภาษาเอสคิวแอลทำให้ผู้ใช้งานง่ายต่อการย้ายดาต้าเบส แอพพลิเคชั่นข้ามระบบฐานข้อมูลของแต่ละบริษัทได้ ตัวอย่างเช่น ผู้ใช้สามารถเข้าถึงข้อมูลที่เก็บไว้แต่ละฐานข้อมูลได้โดยไม่ต้องเปลี่ยนแปลงคำสั่งของเอสคิวแอล ข้อดีอีกอย่างคือการเข้าถึงข้อมูลได้เร็วแม้ว่าฐานข้อมูลจะมีขนาดใหญ่ ข้อเสียหลักของฐานข้อมูลแบบเชิงสัมพันธ์ คือ ไม่สามารถเก็บข้อมูลประเภทรูปภาพได้ ไม่สามารถใช้ชนิดข้อมูลแบบพิเศษได้ เช่น จำนวนเชิงซ้อน, อาร์เรย์ และแอพพลิเคชั่นที่มีการเชื่อมโยงของข้อมูลที่ซับซ้อน (Elmasri และ Navathe 2000)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

time_key (CHAR(10))	fiscal_year (NUMBER(4))	fiscal_quarter (CHAR(10))	Fiscal_month (CHAR(10))
1999Q1M01	1999	Quarter1	January
1999Q1M02	1999	Quarter1	February
1999Q1M03	1999	Quarter1	March
2000Q1M10	2000	Quarter4	October
2000Q1M11	2000	Quarter4	November
2000Q1M12	2000	Quarter4	November
2001Q1M01	2001	Quarter1	January
2001Q1M02	2001	Quarter1	February
2001Q1M03	2001	Quarter1	March
2001Q1M04	2001	Quarter2	April
2002Q1M01	2002	Quarter1	January
2002Q1M02	2002	Quarter1	February
2002Q1M03	2002	Quarter1	March
2002Q1M04	2002	Quarter2	April
2002Q1M05	2002	Quarter2	May
2002Q1M06	2002	Quarter2	June

ตารางที่ 5.1 แสดงตัวอย่าง ฐานข้อมูลแบบเชิงสัมพันธ์

```
select * from films where studio = "MGM"
      into MGM_films ;

select  title, year from parts
      where year < 1940 & ~(role = "Armand")
      into  early_films

select  name, films.title, films.year, role
from     films, parts
where  · films.title = parts.title &
      films.year  = parts.year  &
      studio = "MGM"
into     MGM_stars
```

ตัวอย่าง แสดคำสั่งคิวรี่ด้วยภาษาเอสคิวแอล

5.2. แนวความคิดของฐานข้อมูลแบบเชิงวัตถุ (OODB)

มีลักษณะการเก็บข้อมูลและการเรียกใช้ข้อมูลที่มีรูปแบบแตกต่างจากระบบฐานข้อมูลแบบเชิงสัมพันธ์ คือจะเป็นการมองข้อมูลที่ถูกจัดเก็บในรูปแบบของวัตถุ(Object) ซึ่งวัตถุจะประกอบด้วยแอตทริบิวต์และเมธอด เราสามารถอธิบายลักษณะโดยรวมที่สำคัญๆ ของระบบข้อมูลเชิงวัตถุได้ ดังนี้

5.2.1. Object Identity

จะสนับสนุนไอเดนทิตี(Identity) ของ Object Instance โดยไม่สนใจว่าตัวแปร(Attribute) ของแต่ละออปเจค จะมีค่าเท่าไร และแต่ละออปเจคต้องมีค่า Object Identity เพื่อไว้ใช้อ้างอิงไปยังออปเจคตัวอื่นในแอปพลิเคชั่น สำหรับในระบบฐานข้อมูลที่เป็นแบบเน้นการแสดงค่า(Value-Oriented) จะใช้ค่า หนึ่งที่อยู่ในฐานข้อมูลเป็นค่าหลัก(Key) ในการแทนถึงวัตถุใดวัตถุหนึ่ง ปัญหาที่เกิดขึ้นจากการทำวิธีนี้คือ เมื่อค่าหลักที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการแทนออปเจคมีการเปลี่ยนแปลง จะทำให้เกิดปัญหาการอ้างถึงวัตถุที่ผิดไปจากตัวเดิมได้ และต้องแก้ปัญหาด้วยการแก้ไขค่าหลักที่ใช้ทำการอ้างอิงนี้ทุกตัวเมื่อมีการเปลี่ยนแปลงเกิดขึ้น

ในกรณีของระบบฐานข้อมูลเชิงวัตถุ จะแก้ปัญหานี้ด้วยการแทนค่าที่ใช้ในการอ้างอิงถึงตัวออปเจคใดนั้นเป็นค่าต่างหากค่าหนึ่ง ซึ่งเราเรียกค่านั้นว่า Object Identifier โดยค่านี้จะแตกต่างจากค่าที่เก็บอยู่ในฐานข้อมูลค่าอื่นๆ คือ จะมีค่าที่ไม่ซ้ำกัน ซึ่งค่า Object Identifier ระบบฐานข้อมูลจะใช้ในการอ้างถึง ออปเจคเป็นแบบถาวร คือ ค่าที่ใช้ในการอ้างถึงออปเจคเหล่านั้นจะถูกสร้าง และทำลายโดยตัวระบบจัดการฐานข้อมูลเอง และเราไม่สามารถทำการเปลี่ยนแปลงค่าต่างๆ เหล่านั้นได้

5.2.2. Complex Objects

Complex objects คือออปเจคที่ถูกสร้างมาจากออปเจคพื้นฐานหลายๆ ตัว ด้วยการประยุกต์วิธีการสร้างให้กับออปเจคเหล่านั้น สำหรับออปเจคแต่ละตัวจะมีค่าที่เกี่ยวข้องด้วยอยู่ค่าหนึ่ง ซึ่งอาจเป็นค่าที่มีลักษณะซับซ้อนก็ได้ ซึ่งค่านี้อาจเป็นค่าของ Object Identifier ก็ได้ เป็นผลให้สามารถมีออปเจคสองตัวใดๆ ที่อ้างถึงออปเจคอีกตัวหนึ่งซึ่งเป็นตัวเดียวกันได้ โดยใช้ Object Identifier ที่มีค่าเหมือนกัน และเป็นผลให้เมื่อมีการเปลี่ยนแปลงข้อมูลในตัวออปเจคนั้นๆ ก็จะไม่ส่งผลกระทบไปยังออปเจคตัวอื่นๆ ที่มีการอ้างถึง ซึ่งค่าที่เก็บอยู่ในออปเจคนี้สามารถเรียกได้อีกอย่างว่าเป็นสถานะ(State) ของออปเจคตัวนั้นได้ด้วย Complex Datatype คือข้อมูลที่ซับซ้อน ซึ่งประกอบด้วยการนำเอา Simple Datatype(ประกอบไปด้วย integers, characters, byte string of any length, booleans และ floats) มารวมกันเป็นโครงสร้างของ datatype ตัวใหม่

5.2.3. Method

เมธอด คือ วิธีปฏิบัติการกับข้อมูล ซึ่งแต่ละคลาสจะมีเมธอดของตนเอง โดยการทำงานจะส่งข่าวสาร (Message) ไปยังออปเจคที่ต้องการเพื่อเรียกใช้เมธอดที่อยู่ภายในออปเจคนั้น ออปเจคที่เก็บอยู่ในระบบฐานข้อมูลเชิงวัตถุนั้นจะมีวิธีการใช้งานโดยผ่านทางเมธอดนี้ โดยเมธอดจะประกอบด้วยชื่อ รูปแบบในการใช้งาน (Signature) และส่วนที่เป็นเนื้อของเมธอด

5.2.4. Class

Class คือ กลุ่มของออปเจคที่มีคุณสมบัติพื้นฐานเหมือนกัน การติดต่อกับคลาสกระทำผ่านเมธอดภายในคลาส ข้อมูลที่เก็บอยู่ภายในคลาสเรียกว่า เนื้อข้อมูล (instances) สำหรับออปเจคใดๆ ที่เป็นของคลาสใดคลาสหนึ่งที่อยู่ในฐานข้อมูลจะเรียกคลาสนั้นว่าเป็นคลาสหลัก (Primary Class, Immediate Class, Proper Class) ของออปเจคนั้นๆ ซึ่งจะเป็นคลาสที่ออปเจคตัวนั้นมีความสัมพันธ์ในรูปแบบที่เป็น instance-of ของคลาสนั้น

5.2.5 Encapsulation

แนวความคิดของ Encapsulation มาจาก 1) ความจำเป็นในการแยกส่วนรายละเอียด และส่วนเรียกใช้งานของตัวออปเจค 2) ความจำเป็นสำหรับการทำเป็นหน่วยย่อย Encapsulation คือการรวมกันของโครงสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวแปรและเมธอด เกิดเป็นออปเจคใหม่ที่มีความสามารถในการซ่อนข้อมูลจากระบบภายนอกได้ ทำให้ข้อมูลมีความปลอดภัยขึ้น ซึ่งวิธีการป้องกันนั้นจะอยู่ที่ชนิดของการประกาศข้อมูลภายในคลาส จะมีอยู่ 3 ระดับ

1. **Private** เป็นการประกาศข้อมูลแบบที่ใช้ภายในตัวเอง จะมีผลทำให้ไม่มีคำสั่งใดจากภายนอกสามารถอ้างถึงข้อมูลภายในคลาสนั้นได้เลย นอกจากเมธอดภายในคลาสนั้นเอง การเรียกใช้ก็เรียกผ่านเมธอดของคลาส
2. **Public** เป็นการประกาศข้อมูลแบบสาธารณะ คือมีผลทำให้คลาสอื่นๆ หรือเมธอดอื่นๆ สามารถอ้างถึงข้อมูลภายในคลาสนี้ได้อย่างอิสระ
3. **Protected** เป็นการประกาศข้อมูลแบบสงวนไว้ให้คลาสลูก จะมีผลทำให้มีเฉพาะเมธอดภายในของคลาสนั้นและคลาสที่สืบทอด(Inheritance) เท่านั้นที่สามารถอ้างถึงข้อมูลภายในคลาสได้

5.2.6. Inheritance

ในระบบฐานข้อมูลเชิงวัตถุ แนวคิดของการสืบทอดคุณสมบัติทำให้ออปเจคมีตัวแปร และเมธอดที่เจาะจง(Specialized) ลงไปจากคลาสที่เป็นคลาสต้นแบบ (Superclass) ได้ โดยเราจะเรียกคลาสที่ทำการสืบทอดคุณสมบัติมาจากคลาสต้นแบบว่าคลาสลูก (Subclass) ประโยชน์ของวิธีการสืบทอดคุณสมบัติคือ เราสามารถเพิ่มศักยภาพทางด้านการนำกลับมาใช้ใหม่ (Reusability) และการขยายเพิ่มเติมคุณสมบัติ (Extensibility) ได้

5.2.7. Polymorphism

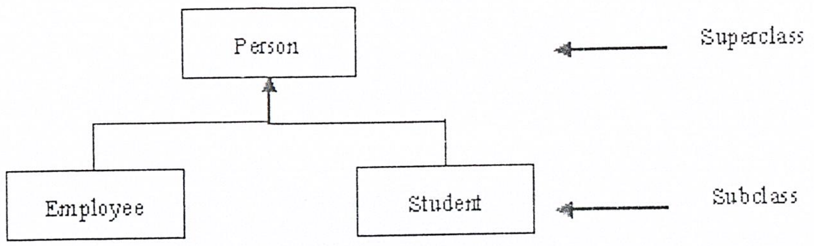
วิธีการ Polymorphism คือรูปแบบหนึ่งของการสืบทอดคุณสมบัติ โดยแบ่งเป็น 2 ลักษณะคือ 1) Overloading หมายถึงการที่เมธอดสองตัวใดๆ ในคลาสเดียวกันที่มีชื่อเหมือนกัน แต่มีรูปแบบการใช้งานต่างกัน 2) Overriding หมายถึงวิธีการที่อ้างถึงตัวแปร หรือเมธอดที่มีชื่อเหมือนกัน แต่มีการเปลี่ยนแปลงส่วนเนื้อภายในของเมธอดที่มีการสืบทอดค่าคุณสมบัตินั้นมาจากคลาสต้นแบบ

5.2.8. Relationship

ความสัมพันธ์ในระบบฐานข้อมูลเชิงวัตถุมี 2 แบบหลักๆ คือ 1) Class-subclass relationship 2) Interobject relationship ซึ่งมี 3 รูปแบบคือ one-to-one, one-to-many และ many-to-many

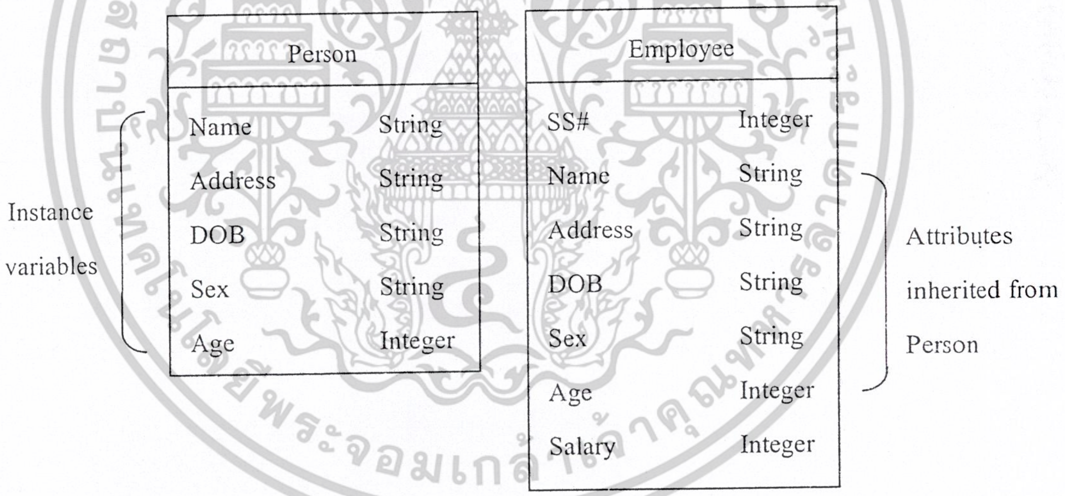
1) Class-subclass relationship เป็นความสัมพันธ์ระหว่างคลาสที่อยู่ในรูปแบบลำดับชั้น ตัวอย่างรูปที่ 5-1 เป็นการแสดงความสัมพันธ์ของ Class-subclass โดยมีคลาส Person เป็นคลาสที่อยู่ระดับเหนือกว่า เรียกว่า Superclass และ Employee กับ Student เป็นคลาสที่อยู่ระดับต่ำกว่า เรียกว่า Subclass จากความสัมพันธ์แสดงให้เห็นว่า คลาส Employee กับ Student ต่างก็เป็น Subclass ของ Superclass Person

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 ความสัมพันธ์ระหว่างคลาสในรูปแบบลำดับชั้น

โดยลักษณะความสัมพันธ์ระหว่างคลาสในรูปแบบ Class-subclass relationship นี้สนับสนุนหลักการของการถ่ายทอดคุณสมบัติ (inheritance) โดยคลาสที่อยู่ระดับสูงกว่าจะถ่ายทอดคุณสมบัติทุกอย่างมายังคลาสที่อยู่ระดับต่ำกว่า และ Subclass สามารถมีคุณสมบัติเพิ่มเติมนอกเหนือจากคุณสมบัติที่ได้รับการถ่ายทอดมาจาก Superclass ตัวอย่างรูปที่ 5-1 แสดงว่าคลาส Employee ได้รับการถ่ายทอดคุณสมบัติมาจากคลาส Person โดยที่ตัวแปร Name, Address, DOB, Sex และ Age ถ่ายทอดมาจาก Superclass Person ส่วน Subclass Employee นั้นมีตัวแปรเพิ่มขึ้นมาอีก 2 ตัวแปร คือ SS# และ Salary



รูปที่ 5.2 Subclass Employee ถ่ายทอดคุณสมบัติมาจาก Superclass Person

2) **Interobject relationship** เป็นความสัมพันธ์ระหว่างออปเจค โดยมี 3 รูปแบบ

- Many-to-many relationship เป็นความสัมพันธ์จากคลาส C ไปคลาส D โดยความสัมพันธ์เป็นเซตของ D ที่มีความเกี่ยวข้องกับแต่ละออปเจคของ C ในทางกลับกัน inverse relationship ก็เป็นเซตของ C ที่มีความเกี่ยวข้องกับแต่ละออปเจคของ D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

● One-to-many relationship เป็นความสัมพันธ์จากคลาส C ไปคลาส D โดยความสัมพันธ์สำหรับแต่ละ C จะได้ค่าที่ไม่ซ้ำกันของ D ในทางกลับกัน inverse relationship จะมีเซตของ C ที่มีความเกี่ยวข้องกับแต่ละออปเจคของ D

● One-to-one relationship เป็นความสัมพันธ์จากคลาส C ไปคลาส D โดยความสัมพันธ์สำหรับแต่ละ C จะได้ค่าที่ไม่ซ้ำกันของ D ในทางกลับกัน inverse relationship แต่ละ D จะได้ค่าที่ไม่ซ้ำกันแต่ละออปเจคของ C

ฐานข้อมูลเชิงสัมพันธ์	ฐานข้อมูลเชิงวัตถุ
1.ใช้ SQL-92	1.ภาษาทางด้านออปเจคจะช่วยลดขนาดของภาษาทางด้านดาต้าเบส
2.เก็บข้อมูลในรูปของตาราง	2.เก็บข้อมูลในรูปของอินสแตรนดาต้าไทป์
3.ไม่รองรับ Complex Attribute	3.รองรับ Complex Attribute
4.ไม่สามารถสร้างโครงสร้างข้อมูลแบบใหม่ได้	4.สามารถสร้างโครงสร้างข้อมูลใหม่ได้
5.เข้าถึงข้อมูลได้ง่าย และรวดเร็ว	5.การเข้าถึงข้อมูลจะต้องผ่านเมธอดของคลาสเท่านั้น
6.ไม่รองรับความสามารถของออปเจค	6.รองรับความสามารถของออปเจค
7.เหมาะกับฐานข้อมูลประเภทโอเปอร์เรชั่นนอลดาต้าเบส	7.เหมาะกับฐานข้อมูลขนาดกลาง

ตารางที่ 5.2 ตารางเปรียบเทียบคุณสมบัติของฐานข้อมูลเชิงสัมพันธ์ กับฐานข้อมูลเชิงวัตถุ

5.3. โครงสร้างของภาษากำหนดรูปแบบเชิงวัตถุ

ระบบฐานข้อมูลเชิงวัตถุได้มีการพัฒนาอย่างต่อเนื่อง ซึ่งระบบฐานข้อมูลเชิงวัตถุนี้มีคุณสมบัติต่างๆ ที่สามารถรองรับข้อมูลที่ซับซ้อนได้ นอกจากนี้ยังมีองค์ประกอบต่างๆ ที่สามารถในงานขนาดใหญ่ได้ แต่เหตุผลประการหนึ่งที่ทำให้ระบบฐานข้อมูลเชิงวัตถุไม่ได้รับความนิยมใช้กันอย่างแพร่หลายก็คือ เพราะยังไม่มีมาตรฐานรองรับระบบฐานข้อมูลเชิงวัตถุนั้นเอง จึงได้มีความพยายามที่จะสร้างมาตรฐานให้กับฐานข้อมูลเชิงวัตถุนี้ โดยมีหน่วยงาน The Object Data Management Group (ODMG) มีหน้าที่ในการสร้างมาตรฐานให้กับระบบฐานข้อมูลเชิงวัตถุ นั่นคือภาษา ODL(Object Definition Language)

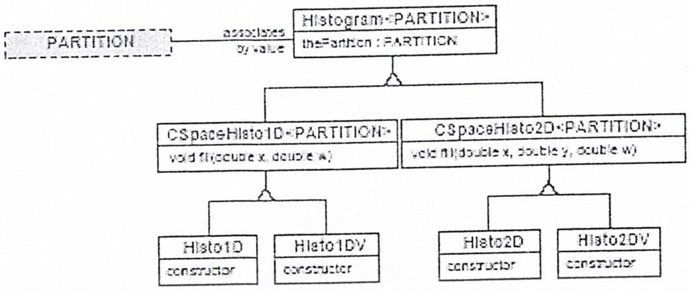
5.3.1. ภาษากำหนดรูปแบบเชิงวัตถุ (Object Definition Language : ODL)

ในมาตรฐาน ODMG นี้ได้กำหนดรูปแบบภาษา ODL ไว้เพื่อเป็นภาษาในการสร้างออปเจค เพื่อนำไปใช้งานในระบบฐานข้อมูลอีกทีหนึ่ง มีรายละเอียดดังนี้

การสร้างคลาสใน ODQL ใน ODL คลาสจะถูกจัดให้อยู่ใน class family (class family คือ logical organization ของคลาส) อย่างเช่น คลาสที่เกี่ยวกับ user และ group อาจจัดอยู่ใน class family เดียวกัน class

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

family สามารถอ้างขึ้นออปเจคใน class family อื่นได้ คลาสสามารถสืบทอดกันได้จากคลาสภายใน class family เดียวกัน แต่สามารถมีแอททริบิวต์(property) ที่อ้างอิงออปเจคใน class family อื่น ดังรูปที่ 5-3



รูปที่ 5.3 ตัวอย่าง class family

เมื่อเรากำหนดคลาส เราจำเป็นต้องกำหนด Class family ด้วยคลาสหนึ่งๆ จะอยู่ใน class family ได้เพียงคลาสเดียวเท่านั้น ชื่อของคลาสจะต้อง unique ภายใน class family หนึ่งๆ แต่เราสามารถมีคลาสชื่อเดียวกันในคนละ class family ได้

ทุกๆ Persistent object ใน ODL จะต้องเป็นอินสเตรนของคลาส และในการกำหนดคลาสนั้นจำเป็นต้องมี

- ชื่อคลาส
- Superclass : โดยจะมีหรือไม่ก็ได้ ทุกๆ user-defined class จะเป็น subclass ของ system class composite โดยอัตโนมัติ Superclass ที่กำหนดใน class-definition จะเป็น immediate superclass ของคลาสที่กำหนด
- Characteristic(method และ property) ของคลาส มีด้วยกัน 3 รูปแบบ

1. Instance characteristic : ได้แก่ instance-level property และ method

2. Class characteristic : ได้แก่ class-level property และ method โดย class-level property จะถูกใช้งานร่วมกัน โดยทุกๆ อินสเตรนของคลาสนั้น เช่น คลาส User มีแอททริบิวต์ average_salary ซึ่งเก็บเงินเดือนเฉลี่ยของพนักงานทุกคนในคลาสนั้น

3. Method on collection : ODQL สนับสนุนเมธอดที่จะทำงานกับ collection ของ ออปเจค หรือของคลาส อย่างเช่น เมธอดที่ทำหน้าที่นับจำนวนของอินสเตรนภายใน collection หรือเมธอดที่ให้อินสเตรนกลับมา ตัวแรกและตัวสุดท้ายใน list และเมธอดที่ทำงานกับ collection of class นั้น คลาสใน collection ควรจะเป็นคลาสนั้น หรือเป็น Subclass ของคลาสนั้น

เราอาจกำหนดค่าเริ่มต้นให้กับ Class-level หรือ instance-level property ได้ และสำหรับ instance-level property นั้นเราสามารถกำหนด constraint อื่นๆ เช่น กำหนดให้แอททริบิวต์นั้นต้องมีค่า (not null) เป็นต้น การกำหนดเมธอดนั้น เราสามารถกำหนดการให้ค่าชนิดของข้อมูลกลับมาด้วย (return type) กำหนดชื่อของเมธอดและพารามิเตอร์ที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างของ ODL ในการสร้างคลาส

```
Interface Person
{
        /* instance properties */
        Attribute String Name;
        Attribute enum Gender{F,M};
        Attribute Date date_birth;
        /* instance operations */
        Integer Age();
}
Interface Employee::Person
{
        /* instance properties */
        Attribute Usigned Integer salary;
        Relationhip Set<Skill> has_skills
                Inverse Skill::belongs_to;
        Relationship Department works_for
                Inverse Department::Employees;
        Relationship Deparment manages
                Inverse Department::Manager;
        /* instance operations */
        Unsigned Integer AvgSalary();
}
Interface Skill
{
        /* instance properties */
        Attribute String name;
        Relationship Set<Employee> belongs_to
                Inverse Employee::has_skills;
}
Interface Department
{
        /* instance properties */
        Attribute String name;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Attribute Integer budget;
Relationship Set<Employee> employees
        Inverse Employee::works_for;
Relationship Employee manager
        Inverse Employee::managers;
/* instance operations */
Unsigned Integer NumOfEmployee
}
```

1. **การประกาศคลาส (Interface Declarations)**

โดยคำว่า "Interface" ใน ODL ใช้ในการประกาศคลาส แล้วตามด้วยชื่อของคลาสที่เราต้องการประกาศ โดยในคลาสจะประกอบไปด้วยแอททริบิวต์, รีเลชั่นชิพ และเมธอด เราสามารถใช้งานแอททริบิวต์ที่เป็น atomic types หรือกำหนดแอททริบิวต์ให้เป็นชนิด structures, collections หรือ collection of structures โดยเราจะใช้คำว่า "Struct" ในการประกาศ ดังในตัวอย่างต่อไปนี้

```
Interface Star
{
    Attribute String name;
    Attribute Struct Addr
            { String street, String city } address;
}
```

2. **การประกาศแอททริบิวต์ใน ODL(Attributes in ODL)**

เป็นการประกาศชนิดของ Properties โดยใช้คำว่า "Attribute" ในการประกาศ เป็นการอธิบายคุณลักษณะของออปเจคตัวนั้นๆ ตัวอย่างเช่น คลาส Person อาจจะประกอบด้วยแอททริบิวต์ name, birthday, address เป็นต้น

3. **การประกาศความสัมพันธ์ใน ODL(Relationships in ODL)**

ส่วนการประกาศความสัมพันธ์ระหว่างคลาสใน ODL เราจะใช้คำว่า "Relationship" ในการประกาศ และในทางกลับกัน เราก็กำหนด inverse relationship เพื่อเป็นแสดงความสัมพันธ์ย้อนกลับระหว่างความสัมพันธ์ของคลาส ซึ่งเราใช้คำว่า "Inverse" ในการอ้างถึงความสัมพันธ์ย้อนกลับนั้น โดยตามด้วยชื่อความสัมพันธ์ของอีกคลาสหนึ่งที่เราต้องการสร้าง inverse relationship เราสามารถแสดงตัวอย่างความสัมพันธ์ระหว่างคลาสได้ดังต่อไปนี้

```
Interface Movie
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Attribute String title;
    Attribute Integer year;
    Attribute Integer length;
    Attribute enum Film { color, blackAndWhite } filmType;
    Relationship Set<Star> stars
            Inverse Star::starredIn;
    Relationship Studio ownedBy
            Inverse Studio::owns;
}
Interface Star
{
    Attribute String name;
    Attribute Struct Addr
            { String street, String city } address;
    Relationship Set<Movie> starredIn
            Inverse Movie::stars;
}
Interface Studio
{
    Attribute String name;
    Attribute String address;
    Relationship Set<Movie> owns
            Inverse Movie::owndBy;
}
```

รูปที่ 5.4 แสดงตัวอย่างรูปแบบของความสัมพันธ์ต่างๆ

4. รูปแบบของความสัมพันธ์

ความสัมพันธ์ระหว่างคลาสมีด้วยกัน 3 รูปแบบ

1. **Many-many relationship** : เป็นความสัมพันธ์จากคลาส C ไปคลาส D โดยความสัมพันธ์เป็นเซตของ D ที่มีความเกี่ยวข้องกับแต่ละออปเจคของ C ในทางกลับกัน inverse relationship ก็เป็นเซตของ C ที่มีความเกี่ยวข้องกับแต่ละออปเจคของ D จากรูปที่ 5-4 stars เป็นความสัมพันธ์แบบ many-many จากคลาส Movie ไปคลาส Star และ starredIn เป็นความสัมพันธ์แบบ many-many จากคลาส Star ไปคลาส Movie โดยทั้งสองความสัมพันธ์ถูกกำหนดเป็นแบบ set

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. **Many-one relationship** : เป็นความสัมพันธ์จากคลาส C ไปคลาส D โดยความสัมพันธ์สำหรับแต่ละ C จะได้ค่าที่ไม่ซ้ำกันของ D ในทางกลับกัน inverse relationship จะมีเซตของ C ที่มีความเกี่ยวข้องกับแต่ละออปเจคของ D จากรูปที่ 5-4 ownedBy เป็นความสัมพันธ์แบบ many-one จากคลาส Movie ไปคลาส Studio เราต้องกำหนด inverse relationship เป็นความสัมพันธ์แบบ one-many จากคลาส D ไปคลาส C โดยในที่นี้ owns เป็นความสัมพันธ์แบบ one-many จากคลาส Studio ไปคลาส Movie

3. **One-one relationship** : เป็นความสัมพันธ์จากคลาส C ไปคลาส D โดยความสัมพันธ์สำหรับแต่ละ C จะได้ค่าที่ไม่ซ้ำกันของ D ในทางกลับกัน inverse relationship แต่ละ D จะได้ค่าที่ไม่ซ้ำกันแต่ละออปเจคของ C

5. ชนิดของตัวแปร **(Types in ODL)** แบ่งได้ 2 แบบ

1. Atomic types : ได้แก่ integer, float, character, character string, boolean และ enumerations สำหรับตัวแปรชนิด enumeration นั้น เป็นการกำหนดรูปแบบของคำหรือข้อความให้มีค่าเท่ากับจำนวนในแบบตัวเลข

2. Interface types : เป็นตัวแปรที่อ้างอิงถึงออปเจคของคลาส ตัวอย่างเช่น ในคลาสหนึ่งมีส่วนประกอบของอีกคลาสหนึ่งอยู่ด้วย

5.3.2.ภาษากำหนดรูปแบบ Cache' (Cache' Definition Language: CDL)

CDL เป็นรูปแบบของไฟล์ที่ใช้ในการสร้างคลาสของระบบฐานข้อมูล Cache' ซึ่งเป็นฐานข้อมูลแบบเชิงวัตถุประเภทหนึ่ง ซึ่งเราสามารถใช้ text edit ในการสร้างไฟล์นี้ ในไฟล์ CDL เราสามารถกำหนดข้อความเพื่อใช้ในการอธิบาย (comment) ได้ 2 แบบดังนี้

1. // หมายถึง เราไม่ต้องการให้โปรแกรมพิจารณาคำสั่งด้านหลังของเครื่องหมาย //

2. /* */ ใช้สำหรับเมื่อเราไม่ต้องการให้โปรแกรมพิจารณาคำสั่งระหว่างเครื่องหมาย /* กับ */

เมื่อเราเขียนคำสั่งตรงตามรูปแบบของไฟล์ CDL เราจะต้องมีสัญลักษณ์ที่ใช้ในการจบคำสั่งด้วยสัญลักษณ์; (semicolon) ส่วนรูปแบบของคำสั่งประกอบด้วยการประกาศที่ใช้งานหลักๆ ดังนี้

1. **การประกาศคลาส (Definition Classes)**

มีรูปแบบการประกาศคลาส ดังนี้

Class {package name.}class name { keywordspecification;… }

โดย Class name คือชื่อของคลาสที่เราต้องการประกาศ

Package name คือชื่อของที่เราประกาศใช้งานให้กับคลาส โดยถ้าชื่อ package นั้นเราไม่ได้ใส่ค่าเข้าไป ตัวแอปพลิเคชั่นจะกำหนดให้ค่าปกติ (default) มีค่าเท่ากับ User.

keywordspecification คือรูปแบบการกำหนดค่าต่างๆ ให้กับคลาสของเรา โดยมีรูปแบบต่างๆ ที่ใช้ในการอธิบายคลาสที่เราสร้าง

2. **การประกาศตัวแปร(Definition of Properties)**

เป็นส่วนที่เราใช้ในการประกาศแอททริบิวต์ โดยมีรูปแบบการประกาศแอททริบิวต์ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Attribute ชื่อแอททริบิวต์ { รายละเอียดของแอททริบิวต์ }

โดยในส่วนของรายละเอียดของแอททริบิวต์ เราสามารถกำหนดรูปแบบต่างๆ ของแอททริบิวต์ได้ ดังต่อไปนี้

1) Calculated : มีความหมายว่า เราต้องคำนวณค่าให้กับแอททริบิวต์นี้ เนื่องจากแอททริบิวต์ไม่ได้เก็บข้อมูลไว้ในหน่วยความจำโดยตรง
2) Initial = expression : เป็นการกำหนดค่าเริ่มต้นให้กับตัวแอททริบิวต์
3) Private : ใช้กำหนดสิทธิในการเข้าถึงค่าแอททริบิวต์
4) Required : เป็นการบอกให้ทราบว่าแอททริบิวต์นี้จะต้องมีค่า(not null)
5) Type = datatypespec : ใช้สำหรับอ้างชนิดของข้อมูลให้กับแอททริบิวต์

3. การประกาศเมธอด(Definition of Methods)

เป็นส่วนที่ใช้ตอบสนองต่อข่าวสารที่เรียกใช้งานออปเจคของคลาส โดยมีรูปแบบการประกาศเมธอด ดังนี้

Method ชื่อเมธอด {รายละเอียดของเมธอด}

โดยในส่วนของรายละเอียดของเมธอด เราสามารถกำหนดรูปแบบต่างๆ ของเมธอดได้ดังต่อไปนี้

1) Call = routinename : ใช้ในการเรียกใช้เมธอดให้ทำงาน
2) Final : ใช้ประกาศเพื่อบอกว่าเมธอดนี้ไม่สามารถเปลี่ยนแปลงส่วนของการจัดการได้อีกแล้ว (Overriding)
3) Expression = expression : ใช้เรียกฟังก์ชั่นของ Cache' ObjectScript
4) Private : ใช้กำหนดสิทธิในการเรียกใช้งานเมธอด
5) Returntype = datatype : ใช้กำหนดค่าชนิดของข้อมูลที่ต้องการส่งค่าออกมาจากเมธอด

ตัวอย่างรูปแบบคลาสของไฟล์ CDL

```
Class User.Person
{
        description = "General person class";
        super = %Library.Persistent;
        persistent;
        /* Definition of the properties */
        attribute Name { type = %Library.String; required; }
        attribute Firstname { type = %Library.String; }
        attribute DOB { type = %Library.Date; }
        attribute Age { type = %Library.Integer; calculated;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        sqlcomputecode =
        {
                :Set {AgeInYears}=$Horolog-{DOB}\365.2425;
        }
        sqlcomputecode; sqlfieldname = AgeInYear; }
    /* Definition of the methods */
    method AgeGet()
    {
        returntype = %Library.Integer;
        expression = $Horolog-..DOB\365.2425;
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การแปลงโครงสร้างฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองข้อมูลอี-อาร์

การออกแบบฐานข้อมูลจะออกแบบนั้น โดยส่วนใหญ่จะทำการเก็บรายละเอียดโครงสร้างของระดับแนวความคิดด้วยแบบจำลองข้อมูลอี-อาร์ ซึ่งเป็นโครงสร้างต่อการทำความเข้าใจเนื่องจากใช้แผนภาพในการแสดง ทำให้เห็นภาพรวมของเอนทิตีทั้งหมดและความสัมพันธ์ระหว่างเอนทิตีในฐานข้อมูล

เมื่อเราได้แบบจำลองข้อมูลอี-อาร์แล้วจะทำการแปลงไปยังโครงสร้างฐานข้อมูลเชิงสัมพันธ์ ซึ่งการที่ใช้แบบจำลองข้อมูลอี-อาร์จะช่วยทำให้ได้โครงสร้างของฐานข้อมูลเชิงสัมพันธ์ที่ดี ช่วยลดปัญหาต่างๆ เช่น การซ้ำซ้อนของข้อมูล ความคัดแย้งของข้อมูล เป็นต้น นอกจากนี้ การออกแบบโครงร่างของฐานข้อมูลในระดับแนวความคิดยังแสดงข้อจำกัดและกฎเกณฑ์ของข้อมูลด้วย

ถึงแม้ว่า จะทำการแปลงโครงสร้างฐานข้อมูลเชิงสัมพันธ์มาจากแบบจำลองข้อมูลอี-อาร์ แต่ยังจำเป็นที่จะต้องมีกระบวนการทำให้เป็นรูปแบบที่เป็นบรรทัดฐาน (Normalization process) เนื่องจากแบบจำลองข้อมูลอี-อาร์นั้น ทำการวิเคราะห์ความสัมพันธ์ของข้อมูลในระดับ เอนทิตีเท่านั้น หากทว่ายังขาดการวิเคราะห์ความสัมพันธ์ระหว่างแอททริบิวต์ในแต่ละรีเลชั่นจึงจำเป็นต้องมีกระบวนการทำให้เป็นรูปแบบที่เป็นบรรทัดฐานอีกขั้นหนึ่ง เพื่อทำการวิเคราะห์ความสัมพันธ์ของข้อมูลในระดับแอททริบิวต์ของแต่ละรีเลชั่น เพื่อให้ความซ้ำซ้อนของข้อมูลในแต่ละรีเลชั่นลดลง ซึ่งจะช่วยลดปัญหาข้อมูลขาดความถูกต้องสมบูรณ์ และปัญหาที่เกิดจากการปรับปรุง เพิ่มเติม และลบข้อมูล

แต่เนื่องจากของจำกัดของฐานข้อมูลเชิงสัมพันธ์ ซึ่งถ้าหากนำมาประยุกต์ใช้ในบ้างงานแล้วจะทำให้ยากในการพัฒนาเช่น การใช้ฐานข้อมูลเชิงสัมพันธ์กับการพัฒนาซอฟต์แวร์เชิงวัตถุ การใช้ฐานข้อมูลเชิงสัมพันธ์กับการประยุกต์งานด้านจีไอเอส (GIS) ซึ่งแบบจำลองข้อมูลอื่นมีความเหมาะสมกว่า เช่น แบบจำลองข้อมูลเชิงวัตถุ จึงมีการศึกษาเกี่ยวกับการถ่ายโอนฐานข้อมูลจากแบบจำลองข้อมูลหนึ่งไปยังแบบจำลองข้อมูลหนึ่ง โดยวิธีนิยมได้แก่ วิธีการวิศวกรรมย้อนกลับ โดยจะทำการแปลงโครงสร้างฐานข้อมูลไปยังแบบจำลองอี-อาร์ และนำอี-อาร์ที่ได้มาทำการแปลงเป็นโครงสร้างฐานข้อมูลอื่นต่อไป เนื่องจากการแปลงโครงสร้างของฐานข้อมูลไปยังแบบจำลองข้อมูลอี-อาร์ ที่เป็นโครงสร้างข้อมูลในระดับแนวความคิดนั้น ทำให้เราสามารถแก้ไข ปรับปรุงรายละเอียดโครงสร้างข้อมูลก่อน เพื่อให้โครงสร้างข้อมูลที่ได้นั้น เหมาะสมกับโครงสร้างของฐานข้อมูลปลายทาง เพื่อให้สามารถใช้ประสิทธิภาพของฐานข้อมูลได้อย่างเต็มที่

ในการแปลงโครงสร้างฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองข้อมูลอี-อาร์นั้น เนื่องจากในการออกแบบฐานจากแบบจำลองข้อมูลอี-อาร์ ไปยังโครงสร้างฐานข้อมูลเชิงสัมพันธ์นั้น จะทำให้ศูนย์เสียรายละเอียดของโครงสร้างของ อี-อาร์ไป ทำให้เมื่อเราทำการนำโครงสร้างของฐานข้อมูลเชิงสัมพันธ์มาทำแปลงย้อนกลับไปเป็นแบบจำลองข้อมูลอี-อาร์ จึงไม่สามารถได้แบบจำลองอี-อาร์ที่สมบูรณ์ซึ่งจำเป็นต้องได้รายละเอียดของผู้ใช้ เพื่อที่จะทำให้แบบจำลองข้อมูลอี-อาร์สมบูรณ์ยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1. วิธีการแปลงโครงสร้างฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองข้อมูลอี-อาร์

ในการแปลงโครงสร้างฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองข้อมูลอี-อาร์ เราจะได้ข้อมูลจากโครงสร้างของฐานข้อมูลเชิงสัมพันธ์จาก รีเลชั่น แอททริบิวต์ แอททริบิวต์คีย์ คีย์หลัก คีย์นอก ตัวอย่างเช่น

รหัสประจำตัว	ชื่อ-สกุล	เพศ
42020018	ทิพวรรณ์ วงศ์อินทร์ตา	หญิง
42020152	แรงราม พลจันทร์	ชาย
42020665	เชาว์วิทย์ วิชิตอำพล	ชาย
42020674	จริญญา ลีฟุม	หญิง
42020699	นงลักษณ์ คะณา	หญิง

รีเลชั่นนักศึกษา

รหัสประจำตัว	รหัสอาจารย์
42020018	Q1011
42020152	Q1023
42020665	Q1035
42020674	Q1011
42020699	Q1023

รีเลชั่นที่ปรึกษา

รหัสอาจารย์	ชื่อ-สกุล	เพศ	เงินเดือน
Q1011	ศิริภัทรา เหมือนมาลัย	หญิง	8,000
Q1023	สุพิมพ์ ศรีพันธ์วรสกุล	หญิง	9,500
Q1035	ศิริชัย ศรีพรหม	ชาย	12,000

รีเลชั่นอาจารย์

ตารางที่ 6.1 ตัวอย่างโครงสร้างฐานข้อมูลเชิงสัมพันธ์

จากตารางที่ 6.1 จะมีโครงสร้างของฐานข้อมูลเชิงสัมพันธ์ ดังนี้

- นักศึกษา (<u>รหัสประจำตัว</u>, ชื่อ-สกุล, เพศ)
- อาจารย์ (<u>รหัสอาจารย์</u>, ชื่อ-สกุล, เพศ, เงินเดือน)
- ที่ปรึกษา (<u>รหัสประจำตัว, รหัสอาจารย์</u>)

นักศึกษามีความสัมพันธ์กับอาจารย์คือเป็นที่ปรึกษา โดยในรีเลชั่นนักศึกษาจะมีรหัสนักประจำตัวเป็นแอททริบิวต์คีย์ ในรีเลชั่นอาจารย์มีแอททริบิวต์รหัสอาจารย์เป็นคีย์หลัก และในรีเลชั่นที่ปรึกษามีรหัสประจำตัวและรหัสอาจารย์รวมกันเป็นคีย์หลัก ซึ่งเป็นคีย์นอก มีรหัสอาจารย์เป็นคีย์นอกที่มาจากรีเลชั่นอาจารย์ และมีรหัสประจำตัวเป็นคีย์นอกที่มาจากรีเลชั่นนักศึกษา

ในการแปลงโครงสร้างฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองข้อมูลอี-อาร์ เราสามารถใช้รายละเอียดของแต่ละรีเลชั่น มาแปลงเป็น เอนทิตี ความสัมพันธ์ แอททริบิวต์ ประเภทต่างได้ โดยมีลำดับขั้นตอนในการแปลงดังนี้

6.2 ขั้นตอนในการแปลงโครงสร้างฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองข้อมูลอี-อาร์

ในการแปลงโครงสร้างฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองข้อมูลอี-อาร์ จะใช้หลักการวิศวกรรมย้อนกลับในการแปลงโครงสร้าง โดยขั้นตอนในการแปลงจะมีขั้นตอนดังต่อไปนี้

6.2.1. การตรวจสอบเอนทิตีไทป์ จะทำการตรวจสอบรีเลชั่นที่คุณสมบัติดังต่อไปนี้

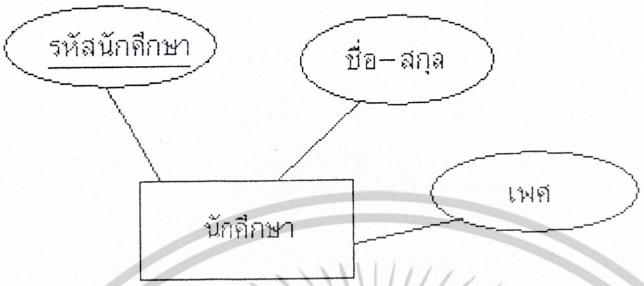
- รีเลชั่นที่มีคีย์หลักเกิดจากแอททริบิวต์เพียงแอททริบิวต์เดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รีเลชั่นที่มีคีย์หลักประกอบด้วยแอททริบิวต์ตั้งแต่สองแอททริบิวต์ขึ้นไป และ แอททริบิวต์ที่เป็นส่วนประกอบของคีย์หลักไม่เป็นคีย์นอก
- แอททริบิวต์ทั้งหมดในแต่ละรีเลชั่นมาแปลงเป็นแอททริบิวต์ของเอนทิตี

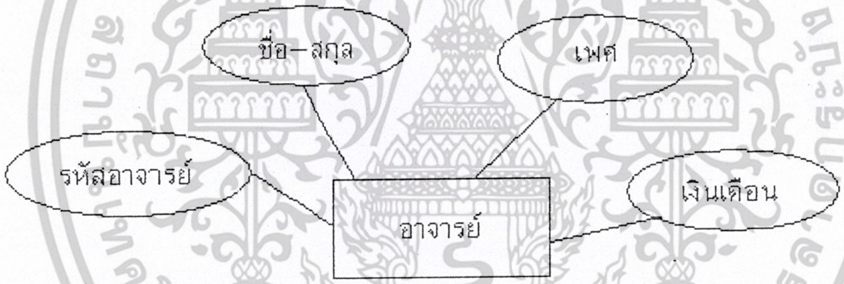
ตัวอย่าง การรีเลชั่นที่ตรวจสอบเป็นเอนทิตี

- รีเลชั่น **นักศึกษา** (<u>รหัสประจำตัว</u>, ชื่อ-สกุล, เพศ)



รูปที่ 6.1 ตัวอย่างเอนทิตีนักศึกษา

- รีเลชั่น **อาจารย์** (<u>รหัสอาจารย์</u>, ชื่อ-สกุล, เพศ, เงินเดือน)



รูปที่ 6.2 ตัวอย่างเอนทิตีอาจารย์

6.2.2 การตรวจสอบเอนทิตีไทป์อ่อนแอ ทำการตรวจสอบรีเลชั่นที่มีคุณสมบัติดังต่อไปนี้

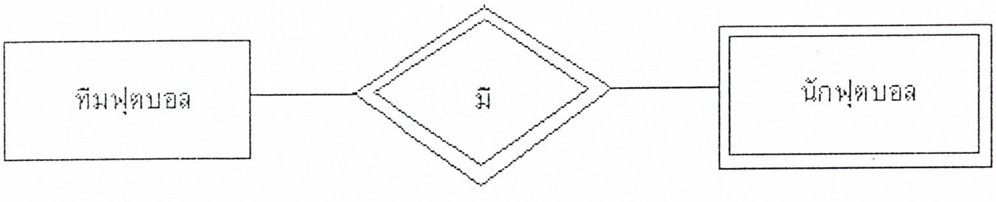
- รีเลชั่นที่มีคีย์หลักประกอบด้วยแอททริบิวต์ตั้งแต่สองแอททริบิวต์ขึ้นไป โดยที่มีแอททริบิวต์ที่เป็นส่วนประกอบของคีย์หลักอย่างน้อยหนึ่งแอททริบิวต์เป็นคีย์นอก และมีแอททริบิวต์อย่างน้อยหนึ่งไม่เป็นคีย์นอก
- รีเลชั่นใดที่ตรวจสอบแล้วเป็นเอนทิตีไทป์ โดยที่มีคีย์หลักเป็นคีย์นอกของรีเลชั่นที่เป็นเอนทิตีไทป์อ่อนแอ แสดงว่ารีเลชั่นนั้นเป็นเอนทิตีไทป์หลักที่มีความสัมพันธ์กับรีเลชั่นที่เอนทิตีไทป์อ่อนแอ
- แอททริบิวต์ทั้งหมดในแต่ละรีเลชั่นมาแปลงเป็นแอททริบิวต์ของเอนทิตี

ตัวอย่าง การรีเลชั่นที่ตรวจสอบเป็นเอนทิตีอ่อนแอ

- รีเลชั่น **ทีมฟุตบอล** (<u>รหัสทีม</u>, ชื่อทีม, สังกัด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รีเลชั่น นักฟุตบอล (<u>รหัสทีม, รหัสนักฟุตบอล</u>, ชื่อนักฟุตบอล, อายุ)



รูปที่ 6.3 ตัวอย่างเอนทิตีอ่อนแอนักฟุตบอล

6.2.3 การตรวจสอบความสัมพันธ์แบบหนึ่งต่อหนึ่ง **(1:1)** หรือ หนึ่งต่อกลุ่ม **(1:M)** ตรวจสอบรีเลชั่นที่มีคุณสมบัติดังนี้

- รีเลชั่นที่เป็นเอนทิตีหรือเอนทิตีอ่อนแอ ซึ่ง มีแอททริบิวต์ที่มีคีย์หลักเป็นคีย์นอกกับรีเลชั่นอื่น โดยที่รีเลชั่นที่มีคีย์นอกนั้นเป็นเอนทิตีด้วย และคีย์นอกนั้นจะต้องไม่เป็นคีย์หลักในรีเลชั่นนั้น แสดงว่า รีเลชั่นทั้งสองมีความสัมพันธ์กันแบบหนึ่งต่อหนึ่งหรือ หนึ่งต่อกลุ่มโดยที่รีเลชั่นที่มีคีย์นอกจะมีระดับต่อความสัมพันธ์เป็นกลุ่ม

- แอททริบิวต์ทั้งหมดในแต่ละรีเลชั่นมาแปลงเป็นแอททริบิวต์ของเอนทิตี

ตัวอย่าง การรีเลชั่นที่ตรวจสอบเป็นความสัมพันธ์แบบหนึ่งต่อหนึ่งหรือ หนึ่งต่อกลุ่ม

- รีเลชั่น คณะ (<u>รหัสคณะ</u>, ชื่อคณะ)

-รีเลชั่น ภาควิชา (<u>รหัสภาควิชา</u>, ชื่อภาควิชา, ที่ทำการ, *รหัสคณะ*)

รูปที่ 6.4 ตัวอย่างความสัมพันธ์หนึ่งต่อกลุ่ม

6.2.4. การตรวจสอบความสัมพันธ์แบบกลุ่มต่อกลุ่ม (M:N) ทำการตรวจสอบรีเลชั่นที่มีคุณสมบัติดังนี้

- รีเลชั่นที่มีคีย์หลักประกอบด้วยแอททริบิวต์ตั้งแต่สองแอททริบิวต์ขึ้นไป โดยที่คีย์หลักทั้งหมดเป็นคีย์นอก ที่ได้มาจากคีย์หลักของรีเลชั่นสองรีเลชั่น แสดงว่ารีเลชั่นเป็นความสัมพันธ์กันแบบกลุ่มต่อกลุ่ม

ตัวอย่าง การรีเลชั่นที่ตรวจสอบเป็นความสัมพันธ์แบบกลุ่มต่อกลุ่ม

- รีเลชั่น ชุดวิชา (<u>รหัสวิชา</u>, ชื่อชุดวิชา, จำนวนหน่วยกิต, *รหัสภาควิชา*)
- รีเลชั่น อาจารย์ (<u>รหัสอาจารย์</u>, ชื่ออาจารย์, เงินเดือน, *รหัสภาควิชา*)
- รีเลชั่น การสอน (<u>รหัสชุดวิชา, รหัสอาจารย์</u>)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

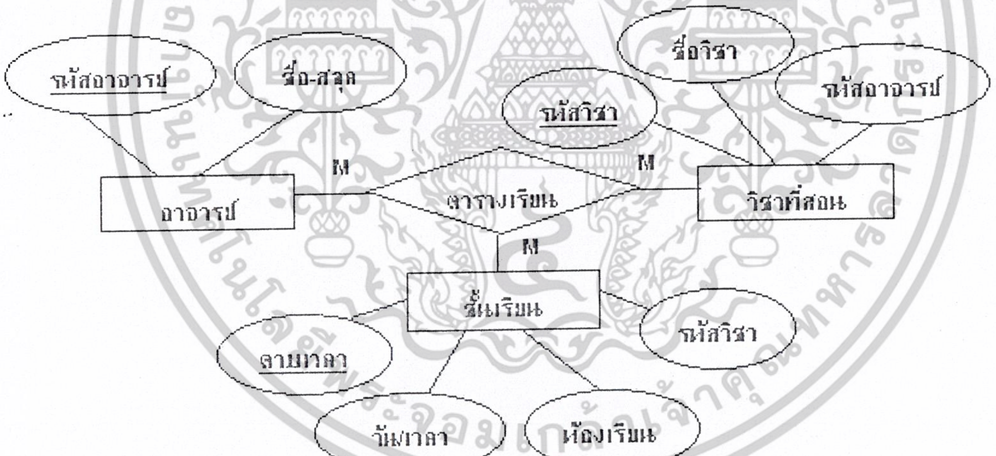
รูปที่ 6.5 ตัวอย่างความสัมพันธ์หนึ่งต่อกลุ่ม

6.2.5. การตรวจสอบความสัมพันธ์แบบเทอนารี (ternary) และเอน-นารี่ (N-ary) ทำการตรวจสอบรีเลชั่นที่มีคุณสมบัติดังนี้

- รีเลชั่นที่มีคีย์หลักประกอบด้วยแอททริบิวต์ตั้งแต่สามแอททริบิวต์ขึ้นไป โดยที่คีย์หลักทั้งหมดเป็นคีย์นอก ที่ได้มาจากคีย์หลักของสามรีเลชั่น แสดงว่ารีเลชั่นเป็นความสัมพันธ์กันแบบเทอนารี แต่ถ้าคีย์นอกมาจากคีย์หลักของรีเลชั่นมากกว่าสามรีเลชั่น แสดงว่ารีเลชั่นเป็นความสัมพันธ์แบบเอน-นารี่

ตัวอย่าง การรีเลชั่นที่ตรวจสอบเป็นความสัมพันธ์แบบเอน-นารี่

- รีเลชั่น อาจารย์ (<u>รหัสอาจารย์</u>, ชื่อ-สกุล)
- รีเลชั่น วิชาที่สอน (<u>รหัสวิชา</u>, ชื่อวิชา, *รหัสอาจารย์*)
- รีเลชั่น การสอน (คาบเวลา,วัน/เวลา.ห้องเรียน,รหัสวิชา)
- รีเลชั่น ตารางเรียน (<u>รหัสวิชา, รหัสอาจารย์, คาบเวลา</u>)



รูปที่ 6.6 ตัวอย่างความสัมพันธ์หนึ่งต่อกลุ่ม

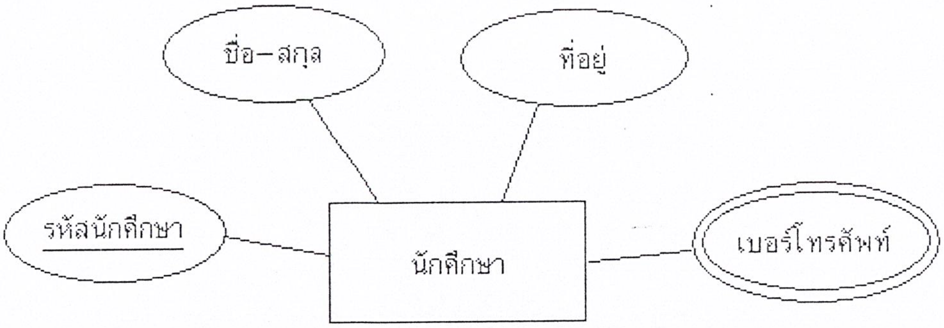
6.2.6 การตรวจสอบแอททริบิวต์หลายค่า (Multivalue attribute) ทำการตรวจสอบรีเลชั่นที่มีคุณสมบัติดังนี้

- รีเลชั่นที่มีแอททริบิวต์ทั้งหมดแอททริบิวต์เป็นคีย์หลัก มีแอททริบิวต์อย่างน้อยหนึ่งแอททริบิวต์ที่เป็นคีย์นอก แสดงว่า แอททริบิวต์ที่ไม่ใช้คีย์นอกเป็นแอททริบิวต์แบบหลายค่าของเอนทิตีที่คีย์หลักเป็นคีย์นอกของรีเลชั่น

ตัวอย่าง การรีเลชั่นที่ตรวจสอบแอททริบิวต์หลายค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รีเลชั่น นักศึกษา (<u>รหัสนักศึกษา</u>, ชื่อ-สกุล, ที่อยู่)
- รีเลชั่น เบอร์โทรศัพท์นักศึกษา (<u>รหัสนักศึกษา, เบอร์โทรศัพท์</u>)



รูปที่ 6.7 ตัวอย่างแอททริบิวต์หลายค่า

6.3 ข้อจำกัดของการใช้วิธีวิศวกรรมย้อนกลับ

เนื่องจากการในการแปลงแบบจำลองอี-อาร์ไปยังฐานข้อมูลเชิงสัมพันธ์นั้น ทำให้รายละเอียดบ้างอย่างหายไป ซึ่งทำให้เกิดข้อจำกัดในการทำวิศวกรรมย้อนกลับ ทำให้แบบจำลองอี-อาร์ที่ได้จากวิศวกรย้อนกลับนั้นมีรายละเอียดไม่ครบถ้วน ได้แก่

- ไม่สามารถตรวจสอบแอททริบิวต์ผสมได้
- ไม่สามารถตรวจสอบความมีส่วนรวมทั้งหมดหรือความมีส่วนรวมบ้างส่วนของความสัมพันธ์
- ความสัมพันธ์แบบหนึ่งต่อหนึ่ง หรือหนึ่งต่อกลุ่ม ไม่สามารถพิสูจน์ได้ว่าแอททริบิวต์ใดที่อยู่ในรีเลชั่นที่ เป็นแอททริบิวต์ของความสัมพันธ์

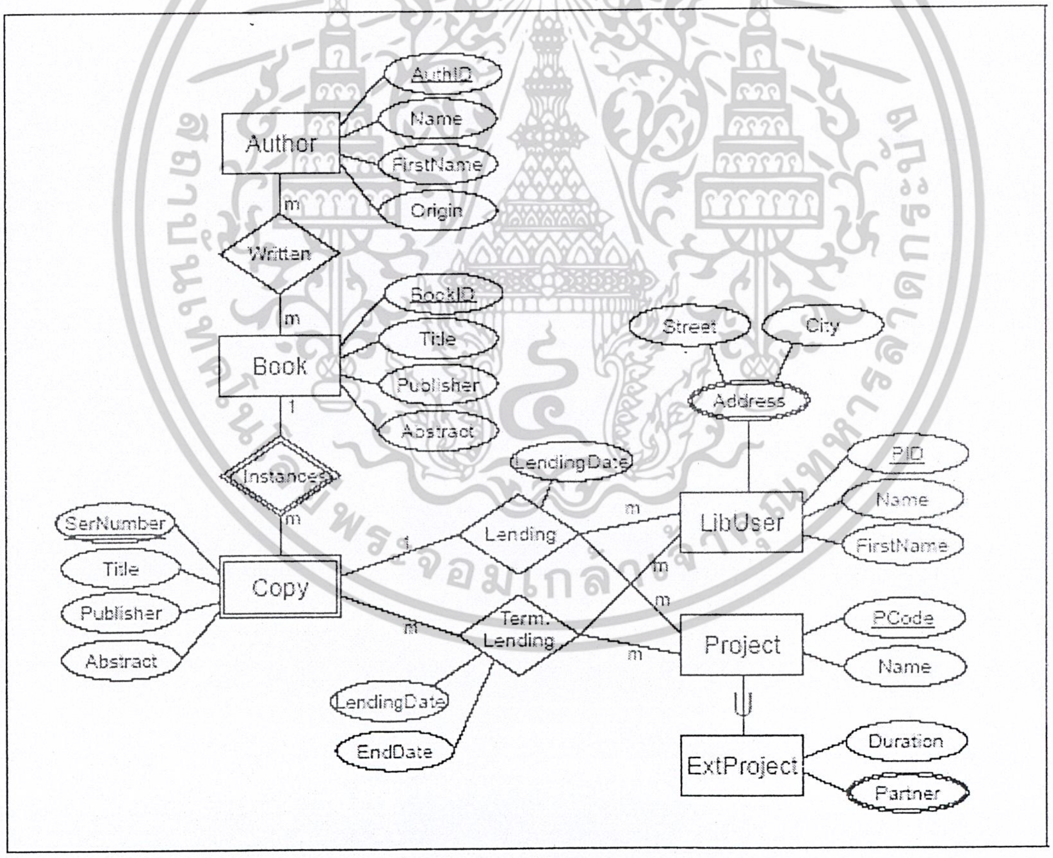
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
การแปลงอีอาร์ไดอะแกรมไปเป็นสคีมาของฐานข้อมูลแบบเชิงวัตถุ

ในการสร้างสคีมาของฐานข้อมูลเชิงวัตถุ เราสามารถทำได้ 2 แบบหลักๆ คือ 1. การแปลงจากตัวสคีมาของฐานข้อมูลแบบเชิงสัมพันธ์ไปเป็นสคีมาของฐานข้อมูลแบบเชิงวัตถุ 2. การแปลงจากอีอาร์ไดอะแกรมไปเป็นสคีมาของฐานข้อมูลแบบเชิงวัตถุ โดยวิธีแรกเป็นการใช้วิธีทางคณิตศาสตร์เข้ามาช่วยสร้างสคีมาของคลาสในฐานข้อมูลแบบเชิงวัตถุ ส่วนวิธีที่สองเป็นการสร้างคลาสโดยพิจารณาจากเอ็นติตี้ไทพของอีอาร์ไดอะแกรม ซึ่งวิธีนี้เราจะได้ผลลัพธ์ของคลาสที่สมบูรณ์กว่าวิธีแรก

7.1. การแปลงจากตัวสคีมาของฐานข้อมูลแบบเชิงสัมพันธ์

ในวิธีการนี้ เราต้องมีความรู้ในเรื่องของอีอีอาร์ไดอะแกรมอยู่บ้าง โดยต้องรู้วิธีการแปลงจากโมเดลอีอีอาร์ไปเป็นสคีมาของฐานข้อมูลแบบเชิงวัตถุ



รูปที่ 7.1 ตัวอย่าง ER Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.1.1. อีอีอาร์โมเดล (EER model)

อีอีอาร์โมเดลเป็นการแสดงเอ็นติตี้ไทพ และความสัมพันธ์ระหว่างเอ็นติตี้ไทพ ซึ่งได้ขยายเพิ่มความสัมพันธ์บางอย่างมาจากอีอาร์โมเดล อย่างเช่น ความสัมพันธ์แบบการสืบทอดคุณสมบัติจากเอ็นติตี้ไทพพ่อ(Supertype) มายังเอ็นติตี้ไทพลูก (Subtype) โดยความสัมพันธ์แบบนี้เราใช้คำว่า "is a" แทนการสืบทอดคุณสมบัติ ตัวอย่าง ถ้าเอ็นติตี้ไทพ E2 เป็นเอ็นติตี้ไทพลูกของเอ็นติตี้ไทพ E1 แล้วแอททริบิวต์และรีเลชั่นต่างๆ ที่มีอยู่ในเอ็นติตี้ไทพ E1 จะถูกถ่ายทอดมาให้กับเอ็นติตี้ไทพ E2

7.1.2. รีเลชั่นนอลโมเดล (Relational model)

เอ็นติตี้และรีเลชั่นชิพระหว่างเอ็นติตี้เป็นรูปแบบโมเดลของรีเลชั่น โดยแต่ละรีเลชั่นมาจากการกำหนดกลุ่มของแอททริบิวต์ และส่วนรีเลชั่นชิพระหว่างเอ็นติตี้จะมีการใช้งานกำหนดแอททริบิวต์ร่วมกัน ความสัมพันธ์ที่สำคัญอย่างหนึ่งคือ กฎข้อบังคับความสัมพันธ์ภายในรีเลชั่นหรือเรียกว่า กฎข้อบังคับในการอ้างถึง(Referential integrity) หมายความว่า ถ้าแอททริบิวต์ A ของรีเลชั่น R1 ถูกกำหนดให้เป็นคีย์นอก (foreign key) โดยอ้างอิงถึงคีย์ในรีเลชั่น R2 แล้วทุกค่าของแอททริบิวต์ A ต้องมีค่าตรงกลับค่าคีย์ที่อยู่ในรีเลชั่น R2

7.1.3. ออปเจค-ออเรียนเตทโมเดล (Object-oriented model)

ในรูปแบบโมเดลเชิงวัตถุ มีการกำหนดคลาสในรูปแบบของแอททริบิวต์ ซึ่งเป็นตัวอธิบายโครงสร้างของตัวออปเจคในคลาส และมีเมธอดเป็นตัวอธิบายพฤติกรรมของออปเจค

โดยแอททริบิวต์ในคลาสของ โมเดลเชิงวัตถุจะสนับสนุนชนิดของข้อมูลได้หลากหลายกว่าโมเดลเชิงสัมพันธ์ โดยจะมีชนิดข้อมูล เช่น integer, char, byte, character of string เป็นต้น รวมถึงชนิดของข้อมูลที่ซับซ้อน ซึ่งมาจากการที่เราประกาศคลาส หรือชนิดของข้อมูลที่มาจากการรวมชนิดของข้อมูลต่างๆ มาประกอบกันเรียกว่า composite attribute

คลาสในรูปแบบโมเดลเชิงวัตถุนั้น สามารถจัดการได้แบบคลาสลำดับชั้น (Class hierarchy) โดยคลาสลูก (Subclass) จะสืบทอดคุณสมบัติต่างๆ มาจากคลาสพ่อของมัน (Superclass)

7.1.4. การแปลงรีเลชั่นนอลสคีมาไปเป็นออปเจค-ออเรียนเตทสคีมา

เป็นวิธีการแปลงรีเลชั่นนอลสคีมาไปเป็นออปเจค-ออเรียนเตทสคีมาโดยการใช้คณิตศาสตร์ระดับสูงในการคำนวณ ซึ่งมีกฎหลักๆ อยู่ 7 ข้อ (มาจากเอกสารของ Weiyi Meng, Aqueo Kamada, Yu-His Chang) แต่กล่าวโดยคร่าวๆ ได้ว่า

1. กำหนดประเภทของรีเลชั่นนอลสคีมาเพื่อใช้เป็นข้อมูลในการแปลงเป็นคลาส ซึ่งต้องได้รับข้อมูลเพิ่มจากผู้ใช้ระบบเพื่อแบ่งแยกประเภทของรีเลชั่นนอลสคีมา
2. แปลงรีเลชั่นนอลสคีมาไปเป็นคลาส
3. นำรีเลชั่นนอลสคีมาหรือแอททริบิวต์ในแต่ละรีเลชั่นนอลสคีมาแปลงไปเป็นแอททริบิวต์ของแต่ละคลาส

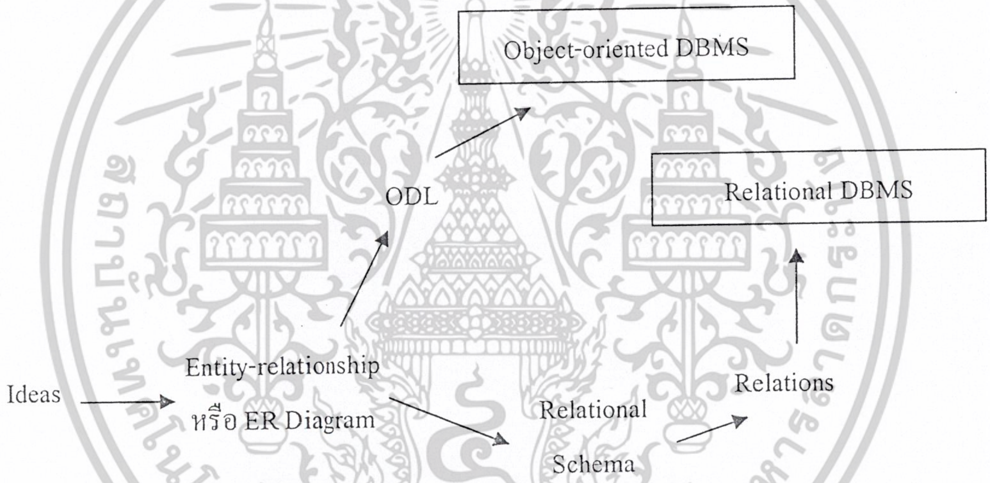
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยในการแปลงรีเลชั่นนอลสคีมาไปเป็นออปเจค-ออเรียนเตทสคีมานั้น มีปัญหาหลักๆ ด้วยกัน 2 ข้อที่เราต้องนำมาพิจารณาในการแปลงสคีมา

1. การแปลงอีอาร์ไดอะแกรมไปยังรีเลชั่นนอลสคีมาทำให้รูปแบบประโยค(Semantic) บางอย่างหายไป ทำให้เมื่อเราต้องการทำการแปลงรีเลชั่นนอลสคีมาไปยังออปเจค-ออเรียนเตทสคีมาหรืออีอาร์ไดอะแกรม จำเป็นต้องได้รับข้อมูลเพิ่มเติมบางอย่าง
2. รีเลชั่นนอลสคีมาที่ทำการแปลงควรจะได้รับการออกแบบมาอย่างดี หรือ ควรจะผ่านคุณสมบัติของนอร์มัลฟอร์มที่ 3 (3NF) เป็นอย่างน้อย

7.2. การแปลงจากอีอาร์ไดอะแกรม

เป็นวิธีการที่ใช้ในการนำโครงสร้างของอีอาร์ไดอะแกรมมาใช้ในการพิจารณาสร้างเป็นคลาสในออปเจค-ออเรียนเตทสคีมา ซึ่งวิธีการนี้เราจะได้คุณสมบัติต่างๆ ของคลาสและความสัมพันธ์ต่างๆ ระหว่างคลาสได้อย่างถูกต้องตามหลักการของออปเจค-ออเรียนเตทสคีมา



รูปที่ 7.2 โมเดลของฐานข้อมูลและกระบวนการใช้งาน

จากรูปที่ 7.2 เป็นการแสดงกระบวนการออกแบบ โดยเราเริ่มจากหาแนวความคิดเกี่ยวกับข้อมูลที่ต้องการสร้างเป็นโมเดล แนวความคิดเหล่านั้นจะถูกแปลให้อยู่ในรูปของอีอาร์ไดอะแกรม แล้วหลังจากนั้นก็นำอีอาร์ไดอะแกรมที่ได้แปลงเป็นรูปแบบภาษาที่ใช้ออกแบบโครงสร้างของฐานข้อมูลได้ 2 อย่างหลักๆ คือ รีเลชั่นนอลสคีมา หรือภาษากำหนดเชิงวัตถุ (Object Definition Language : ODL) โดยหลังจากแปลงเป็นรีเลชั่นนอลสคีมาก็จะทำการแปลให้อยู่ในรูปของรีเลชั่นนอล โดยใช้งานกับระบบฐานข้อมูลแบบเชิงสัมพันธ์ (Relational DBMS : RDBMS) หรือหลังจากแปลงเป็นภาษากำหนดเชิงวัตถุเพื่อนำไปใช้งานกับระบบฐานข้อมูลแบบเชิงวัตถุ(Object-oriented DBMS : OODBMS)

การแปลงจากอีอาร์ไดอะแกรมไปเป็นภาษากำหนดเชิงวัตถุนั้น มีส่วนหลักๆ ในการพิจารณา 3 ส่วน

1. กลุ่มของเอ็นติตี้ในอีอาร์ไดอะแกรม ซึ่งถูกแปลงให้เป็นคลาสในภาษากำหนดเชิงวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. แอททริบิวต์ ซึ่งใช้เป็นค่าในการอธิบายคุณสมบัติของคลาส โดยแอททริบิวต์ในอีอาร์ไดอะแกรม และในภาษากำหนดเชิงวัตถุ จะนำมาจากแนวคิดเดียว ก็คือ กำหนดแอททริบิวต์ของเอ็นติตี้ในไดอะแกรมให้เป็นแอททริบิวต์ของคลาสในภาษากำหนดเชิงวัตถุ

3. ความสัมพันธ์ระหว่างเอ็นติตี้ ซึ่งใช้ในการแสดงความสัมพันธ์ระหว่างเอ็นติตี้ตั้งแต่ 2 เอ็นติตี้ขึ้นไป โดยความสัมพันธ์ระหว่างเอ็นติตี้ในอีอาร์ไดอะแกรม กับความสัมพันธ์ของคลาสในภาษากำหนดเชิงวัตถุจะมีลักษณะคล้ายๆ กัน แต่อย่างไรก็ตาม

(a) ในอีอาร์ไดอะแกรม เรากำหนดให้ชื่อแอททริบิวต์ให้ใช้แสดงความสัมพันธ์ระหว่างเอ็นติตี้ แต่ในขณะที่ความสัมพันธ์ของคลาสในภาษากำหนดเชิงวัตถุจะกำหนดโดยใช้คำว่า “relationship” และประกาศความสัมพันธ์ย้อนกลับด้วยคำว่า “inverse” ให้กับคลาสในภาษากำหนดเชิงวัตถุ ดังแสดงในรูปที่ 7-3

```
Interface Movie
{
    Attribute String title;
    Attribute Integer year;
    Attribute Integer length;
    Attribute enum Film { color, blackAndWhite } filmType;
    Relationship Set<Star> stars
            Inverse Star::starredIn;
    Relationship Studio ownedBy
            Inverse Studio::owns;
}
Interface Star
{
    Attribute String name;
    Attribute Struct Addr
            { String street, String city } address;
    Relationship Set<Movie> starredIn
            Inverse Movie::stars;
}
```

รูปที่ 7.3 แสดงตัวอย่างรูปแบบของความสัมพันธ์ต่างๆ

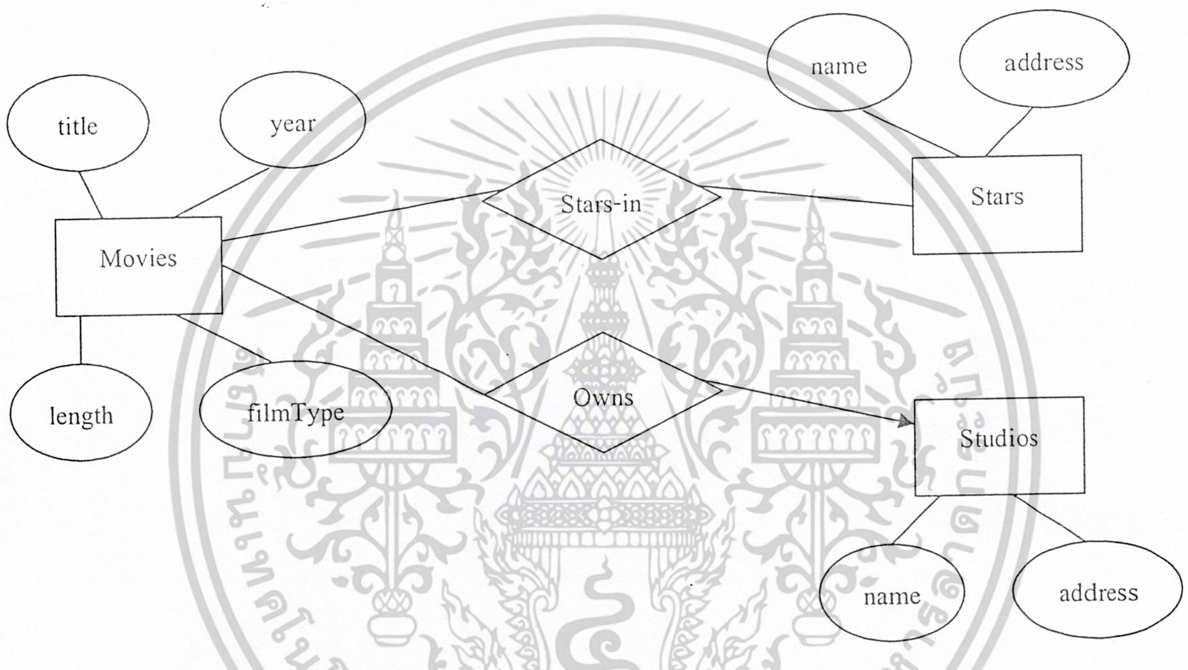
(b) ในอีอาร์ไดอะแกรม ความสัมพันธ์ระหว่างเอนทิตีแบบหลายเอนทิตี (Many-to-many relationship) นั้นเวลาแปลงเป็นรีเลชั่นนอลสคีมา เราจะแปลงได้ 3 รีเลชั่น คือจะมีรีเลชั่นของเอ็นติตี้ 2 รีเลชั่น และมีรีเลชั่นที่สร้างจากคีย์หลักของทั้งสองเนติตี้อีก 1 รีเลชั่น แต่ในภาษากำหนดเชิงวัตถุ เราจะแปลงได้เป็น 2 คลาสเท่านั้น คือ 2 คลาสของเอ็นติตี้ในอีอาร์ไดอะแกรม และทั้งสองคลาสต้องมีการกำหนดความสัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระหว่างกันด้วย โดยใช้คำว่า "relationship" ในการประกาศความสัมพันธ์ ตัวอย่างในรูปที่ 7-4 ดูได้จากความสัมพันธ์ระหว่างคลาส Movie กับคลาส Star ซึ่งเป็นความสัมพันธ์แบบหลายคลาส โดยในคลาส Movie ใช้ตัวแปร stars ในการแสดงความสัมพันธ์ระหว่าง และในคลาส Star มีตัวแปร starredIn เพื่อประกาศเป็นความสัมพันธ์ย้อนกลับของตัวแปร stars ในคลาส Movie

(c) ความสัมพันธ์ในอีอาร์ไดอะแกรมสามารถแสดงความสัมพันธ์ระหว่างเอ็นติตี้ได้มากกว่า 2 เอ็นติตี้ แต่ในความสัมพันธ์ของคลาสในภาษากำหนดเชิงวัตถุจะแสดงความสัมพันธ์ระหว่างคลาสได้เพียง 2 คลาสเท่านั้น ซึ่งหมายความว่า ภาษากำหนดเชิงวัตถุนั้นไม่สนับสนุนความสัมพันธ์แบบยูนารี (U-nary relationship)

ตัวอย่างการแปลงความสัมพันธ์ของคลาสในภาษากำหนดเชิงวัตถุ



รูปที่ 7.4 เป็นการแสดงความสัมพันธ์ระหว่างเอนทิตีในอีอาร์ไดอะแกรม

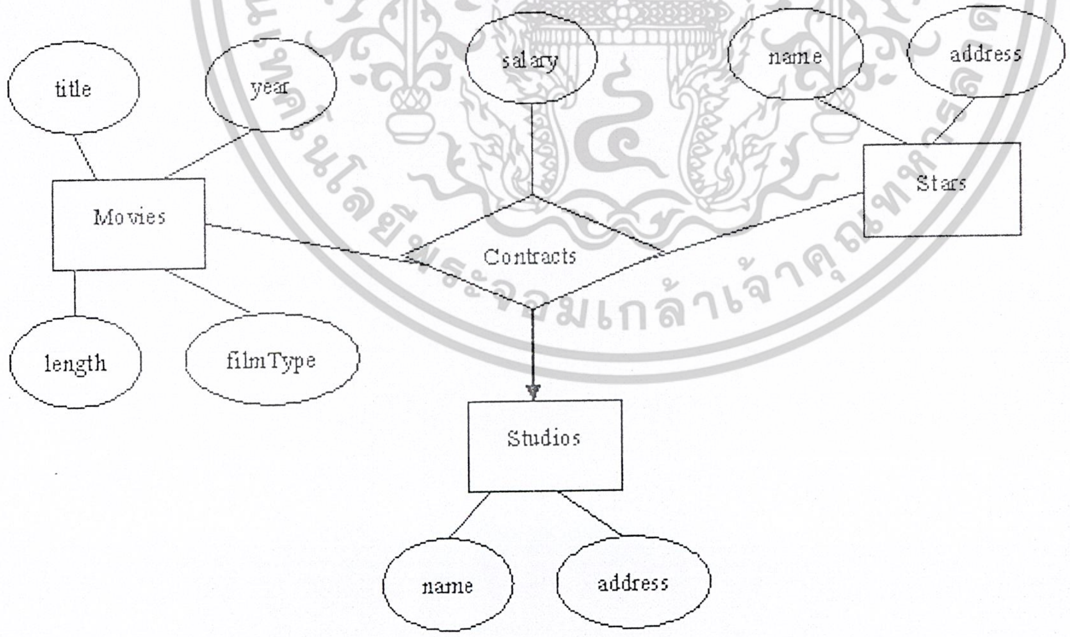
เราสามารถแปลงเป็นคลาสในภาษากำหนดเชิงวัตถุได้ ดังนี้คือ

```
Interface Movie
{
    Attribute String title;
    Attribute Integer year;
    Attribute Integer length;
    Attribute enum Film { color, blackAndWhite } filmType;
    Relationship Set<Star> stars
            Inverse Star::starredIn;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Relationship Studio ownedBy
            Inverse Studio::owns;
}
Interface Star
{
    Attribute String name;
    Attribute Struct Addr
            { String street, String city } address;
    Relationship Set<Movie> starredIn
            Inverse Movie::stars;
}
Interface Studio
{
    Attribute String name;
    Attribute String address;
    Relationship Set<Movie> owns
            Inverse Movie::owndBy;
}
```

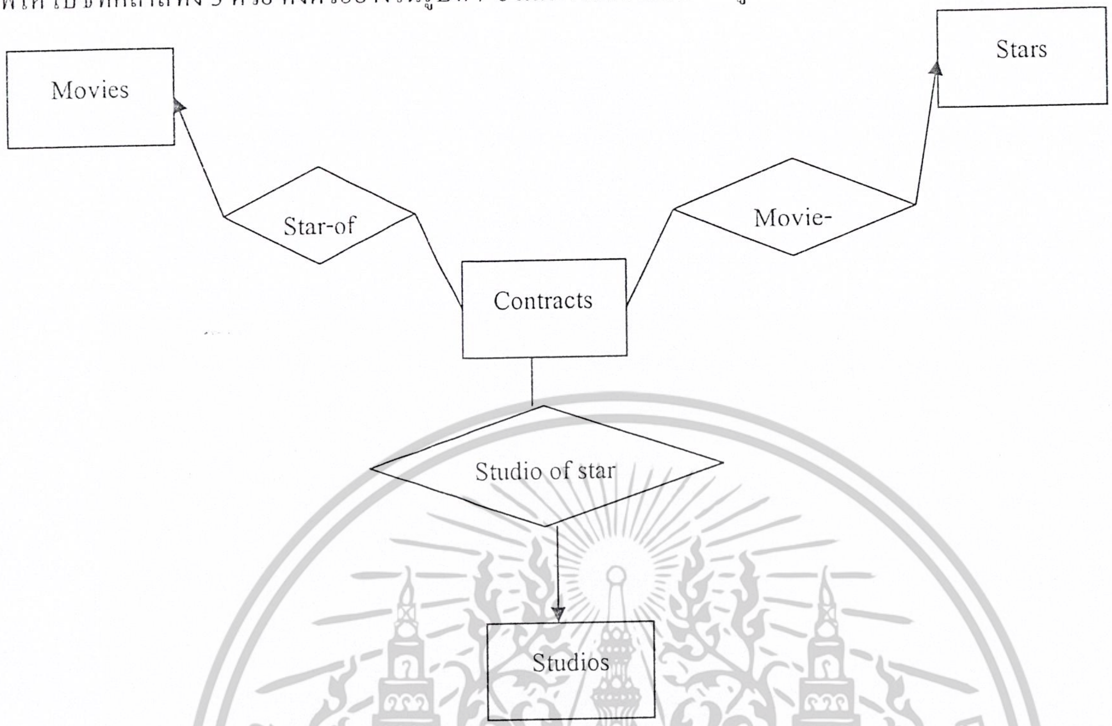
ตัวอย่างการแปลงความสัมพันธ์แบบยูนารี(U-nary relationship)



รูปที่ 7.5 แสดงความสัมพันธ์ระหว่างเอนทิตีในอีอาร์ไดอะแกรมแบบยูนารี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราต้องแปลงรีเลชั่น Contract ให้กลายเป็นคลาสก่อน จากนั้นก็สร้างความสัมพันธ์ระหว่างคลาส Contract กับคลาสทั้ง 3 คลาส ได้แก่ คลาส Movie, คลาส Studio และคลาส Star และต้องมีการประกาศรีเลชั่นชิพให้ไปชี้ที่คลาสทั้ง 3 ด้วย ดังตัวอย่างในรูปที่ 7-6 และตัวอย่างคลาสในรูปที่ 7-7



รูปที่ 7.6 แสดงรูปอีอาร์ไดอะแกรมใหม่ที่เราได้แปลงรีเลชั่น Contract เป็นเอนทิตี

```
Interface Movie
{
    Attribute String title;
    Attribute Integer year;
    Attribute Integer length;
    Attribute enum Film { color, blackAndWhite } filmType;
    Relationship Set<Contract> contracts
            Inverse Contract::movieOfStar;
}
Interface Star
{
    Attribute String name;
    Attribute Struct Addr
            { String street, String city } address;
    Relationship Set<Contract> starredIn
            Inverse Contract::stars;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Interface Studio
{
    Attribute String name;
    Attribute String address;
    Relationship Set<Contract> owns
            Inverse Contract::ownerOfStar;
}
Interface Contract
{
    Attribute Integer salary;
    Relationship Movie movieOfStar
            Inverse Movie::contracts;
    Relationship Star stars
            Inverse Star::starredIn;
    Relationship Studio ownerOfStar
            Inverse Studio::owns;
}
```

รูปที่ 7.7 เป็นคลาสในภาษากำหนดเชิงวัตถุที่ได้จากการแปลงรูปที่ 7.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8
การออกแบบและพัฒนาโปรแกรมในการแปลงโครงสร้างฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ

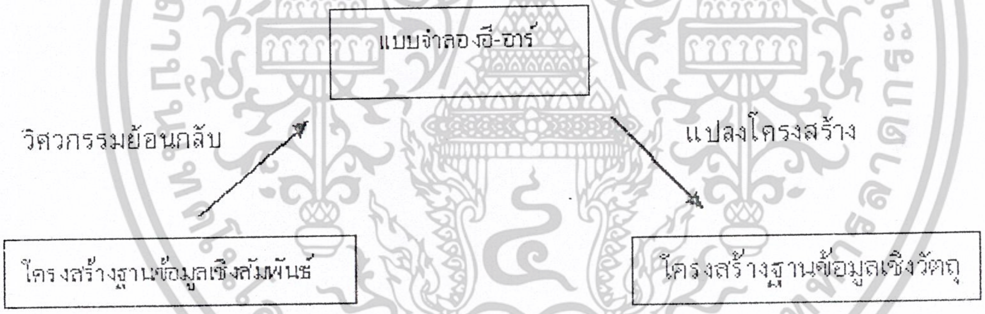
8.1 การออกแบบโครงสร้างโปรแกรม

ในการออกแบบโครงสร้างของโปรแกรมแปลงฐานข้อมูลนั้น สามารถแบ่งส่วนการทำงานเป็น 3 ส่วนหลักๆ เพื่อที่จะสามารถนำแต่ละส่วนของโปรแกรมในโครงงานนำไปประยุกต์ได้สะดวก ได้แก่

8.1.1. ส่วนวิศวกรรมย้อนกลับฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองข้อมูลอี-อาร์ โดยในส่วนนี้จะทำการแปลงโครงสร้างฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองข้อมูลอี-อาร์

8.1.2. ส่วนติดต่อผู้ใช้งานเพื่อปรับปรุงแบบจำลองข้อมูลอี-อาร์ เพื่อให้ผู้ใช้สามารถปรับปรุงแบบจำลองข้อมูลอี-อาร์ ให้มีความถูกต้องตามความต้องการมากขึ้น เนื่องจากในการทำวิศวกรรมย้อนกลับฐานข้อมูลเชิงสัมพันธ์นั้น ไม่สามารถแปลงโครงสร้างไปเป็นอี-อาร์ได้ถูกต้องทั้งหมด จึงจำเป็นที่จะต้องได้รับข้อมูลจากผู้ใช้เพื่อให้แบบจำลองข้อมูลอี-อาร์มีความสมบูรณ์ยิ่งขึ้น

8.1.3. ส่วนแปลงแบบจำลองข้อมูลอี-อาร์ไปยังโครงสร้างข้อมูลเชิงวัตถุ



รูปที่ 8.1 โครงสร้างโปรแกรม

8.2 การออกแบบโปรแกรม

8.2.1 ในส่วนวิศวกรรมย้อนกลับจากฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองข้อมูลอี-อาร์ ในส่วนนี้มีขั้นตอนการทำงานดังนี้

- ติดต่อกับฐานข้อมูลเชิงสัมพันธ์ เพื่อดึงรายละเอียดโครงสร้างของฐานข้อมูลเพื่อใช้ในการทำวิศวกรรมย้อนกลับ โดยข้อมูลที่ได้จะได้แก่
    - ตารางต่างๆในฐานข้อมูล
    - แอททริบิวต์ที่อยู่ภายในตาราง
    - คีย์หลักที่อยู่ภายในตาราง
    - คีย์นอกที่อยู่มีความสัมพันธ์อื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ออกแบบ เมต้าอี-อาร์เพื่อทำการใช้ในการออกแบบฐานข้อมูล เพื่อใช้เก็บรายละเอียดของอี-อาร์ที่ได้จากการทำวิศวกรรมย้อนกลับ

- เมื่อได้รายละเอียดต่างๆของฐานข้อมูลเชิงสัมพันธ์แล้วจะทำวิศวกรรมย้อนกลับเพื่อให้ได้แบบจำลองข้อมูลอี-อาร์ โดยมีขั้นตอนดังนี้

- ตรวจสอบเอนทิตีไทป์
- ตรวจสอบเอนทิตีไทป์อ่อนแอ
- ตรวจสอบรีเลชั่นชิพไทป์
- ตรวจสอบเอนนารี่รีเลชั่นชิพ
- ตรวจสอบแอททริบิวต์หลายค่า

- บันทึกรายละเอียดของแบบจำลองข้อมูลอี-อาร์ลงในฐานข้อมูลที่ได้ออกแบบเอาไว้เพื่อนำไปใช้ต่อไป

8.2.2 ส่วนติดต่อผู้ใช้งานเพื่อปรับปรุงแบบจำลองข้อมูลอี-อาร์

ในส่วนนี้จะเป็นการนำข้อมูลที่เก็บเอาไว้ในฐานข้อมูลซึ่งเป็นรายละเอียดเกี่ยวกับแบบจำลองอี-อาร์มาแสดงเพื่อให้ผู้ใช้เพิ่มเติมรายละเอียดเกี่ยวกับแบบจำลองข้อมูลให้ครบถ้วน โดยเมื่อผู้ใช้ทำการแก้ไขควรมีการตรวจสอบความถูกต้องของแบบจำลองข้อมูลว่ามีรายละเอียดครบถ้วนสมบูรณ์หรือไม่เพื่อที่ผู้ใช้ได้

8.2.3. ส่วนแปลงแบบจำลองข้อมูลอี-อาร์ไปยังโครงสร้างข้อมูลเชิงวัตถุ

ในการออกแบบโปรแกรมของส่วนการแปลงแบบจำลองข้อมูลอี-อาร์ไปยังฐานข้อมูลเชิงวัตถุนั้น เราจะต้องใช้งานฐานข้อมูลที่ใช้เก็บรายละเอียดต่างๆ เกี่ยวกับแบบจำลองข้อมูลอี-อาร์ ซึ่งเราเรียกฐานข้อมูลตัวนี้ว่า อี-อาร์เมต้าสคีมา โดยในรายละเอียดของอี-อาร์เมต้าสคีมามีทั้งหมด 6 ตาราง ดังนี้คือ

1. ตาราง EntityType
   - EID เป็นข้อมูลชนิด AutoNumber
   - Ename เป็นข้อมูลชนิด Text
   - Etype เป็นข้อมูลชนิด Text
   - ESuper เป็นข้อมูลชนิด Text

   โดยมีคีย์หลักคือ EID
2. ตาราง EntityAttribute
   - AID เป็นข้อมูลชนิด AutoNumber
   - EID เป็นข้อมูลชนิด Number
   - Aname เป็นข้อมูลชนิด Text
   - DataType เป็นข้อมูลชนิด Text
   - AttrType เป็นข้อมูลชนิด Text

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Array เป็นข้อมูลชนิด Yes/No
- Primarykey เป็นข้อมูลชนิด Yes/No

โดยมีคีย์หลักคือ AID และมีคีย์นอกคือ EID มาจากตาราง EntityType

3. ตาราง StructureAttribute
   - Struct_id เป็นข้อมูลชนิด AutoNumber
   - Struct_name เป็นข้อมูลชนิด Text
   - Aname เป็นข้อมูลชนิด Text
   - DataType เป็นข้อมูลชนิด Text
   - AttrType เป็นข้อมูลชนิด Text
   - Array เป็นข้อมูลชนิด Yes/No

   โดยมีคีย์หลักคือ Struct_id
4. ตาราง Relationship
   - RID เป็นข้อมูลชนิด AutoNumber
   - Rname เป็นข้อมูลชนิด Text

   โดยมีคีย์หลักคือ RID
5. ตาราง RelationshipAttribute
   - AID เป็นข้อมูลชนิด AutoNumber
   - RID เป็นข้อมูลชนิด Number
   - Aname เป็นข้อมูลชนิด Text
   - DataType เป็นข้อมูลชนิด Text
   - AttrType เป็นข้อมูลชนิด Text
   - Array เป็นข้อมูลชนิด Yes/No

   โดยมีคีย์หลักคือ AID และมีคีย์นอกคือ RID มาจากตาราง RelationshipAttribute
6. ตาราง Cardinality
   - CID เป็นข้อมูลชนิด AutoNumber
   - RID เป็นข้อมูลชนิด Number
   - EID เป็นข้อมูลชนิด Number
   - Cardinality เป็นข้อมูลชนิด Text

   โดยมีคีย์หลักคือ CID และมีคีย์นอกคือ RID กับ EID เป็น Combine key

ส่วนในการออกแบบโปรแกรม ผู้ใช้งานสามารถที่จะสร้างไฟล์ฐานข้อมูลแบบเชิงวัตถุได้ 2 รูปแบบด้วยกัน คือ ไฟล์ ODL กับไฟล์ CDL เหตุที่เราต้องทำเป็นสองแบบ เนื่องจากโดยโครงงานนี้เราใช้งาน Cache' Object DBMS เป็นฐานข้อมูลแบบเชิงวัตถุ ซึ่งใช้ไฟล์ CDL ในการสร้างคลาสของฐานข้อมูล โดยไฟล์ CDL จะมีไม่สนับสนุนคุณสมบัติบางอย่างของมาตรฐานของระบบฐานข้อมูลเชิงวัตถุดังในไฟล์ ODL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. การสร้างคลาสในไฟล์ ODL

1.1 ส่วนของการใส่ชื่อโปรเจคของฐานข้อมูล

เรากำหนดให้ผู้ใช้จะต้องใส่ข้อมูลชื่อโปรเจคของฐานข้อมูลก่อนที่จะทำการแปลงแบบจำลองข้อมูลอี-อาร์ไปยังฐานข้อมูลเชิงวัตถุ ถ้าผู้ใช้ไม่ได้ใส่ชื่อโปรเจคแล้ว ตัวโปรแกรมก็จะไม่สร้างคลาสของไฟล์ ODL ให้

1.2 ส่วนของการสร้างคลาส และติดต่อกับผู้ใช้

ในส่วนนี้ ตัวโปรแกรมจะสร้างคลาสทั้งหมด และกำหนดความสัมพันธ์ระหว่างคลาสต่างๆ โดยจะแสดงคลาส และความสัมพันธ์ระหว่างคลาสทั้งหมดไว้ในกล่องช่องว่าง และผู้ใช้สามารถตรวจสอบคลาสและความสัมพันธ์ระหว่างคลาสทั้งหมดที่โปรแกรมได้ทำการแปลงมา ผู้สามารถแก้ไขหรือเพิ่มเติมข้อมูลบางอย่างให้กับคลาสได้อีก

1.3 ส่วนของการใส่ชื่อและพาธของไฟล์

ผู้ใช้ต้องใส่ชื่อและพาธของไฟล์ ที่ต้องการสร้างคลาสในฐานข้อมูลเชิงวัตถุ โดยผู้ใช้ต้องใส่ข้อมูลชื่อที่มีนามสกุล .ODL ได้เท่านั้น ถ้าผู้ใช้ใส่ชื่อไฟล์หรือพาธที่ไม่ใช่ไฟล์ .ODL ตัวโปรแกรมก็จะแจ้งข้อผิดพลาดให้ผู้ใช้ทราบ

1.4 ส่วนของการสร้างไฟล์ฐานข้อมูล

สุดท้ายผลลัพธ์ที่ได้จากการเป็นไฟล์ .ODL

2. การสร้างคลาสในไฟล์ CDL

2.1 ส่วนของการใส่ชื่อโปรเจคของฐานข้อมูล

เรากำหนดให้ผู้ใช้จะต้องใส่ข้อมูลชื่อโปรเจคของฐานข้อมูลก่อนที่จะทำการแปลงแบบจำลองข้อมูลอี-อาร์ไปยังฐานข้อมูลเชิงวัตถุ ถ้าผู้ใช้ไม่ได้ใส่ชื่อโปรเจคแล้ว ตัวโปรแกรมก็จะไม่สร้างคลาสของไฟล์ CDL ให้

2.2 ส่วนของการสร้างคลาส และติดต่อกับผู้ใช้

ในส่วนนี้ ตัวโปรแกรมจะสร้างคลาสทั้งหมด และกำหนดความสัมพันธ์ระหว่างคลาสต่างๆ โดยจะแสดงคลาส และความสัมพันธ์ระหว่างคลาสทั้งหมดไว้ในกล่องช่องว่าง และผู้ใช้สามารถตรวจสอบคลาสและความสัมพันธ์ระหว่างคลาสทั้งหมดที่โปรแกรมได้ทำการแปลงมา ผู้สามารถแก้ไขหรือเพิ่มเติมข้อมูลบางอย่างให้กับคลาสได้อีก

2.3 ส่วนของการใส่ชื่อและพาธของไฟล์

ผู้ใช้ต้องใส่ชื่อและพาธของไฟล์ ที่ต้องการสร้างคลาสในฐานข้อมูลเชิงวัตถุ โดยผู้ใช้ต้องใส่ข้อมูลชื่อที่มีนามสกุล .CDL ได้เท่านั้น ถ้าผู้ใช้ใส่ชื่อไฟล์หรือพาธที่ไม่ใช่ไฟล์ .CDL ตัวโปรแกรมก็จะแจ้งข้อผิดพลาดให้ผู้ใช้ทราบ

2.4 ส่วนของการสร้างไฟล์ฐานข้อมูล

สุดท้ายผลลัพธ์ที่ได้จากการเป็นไฟล์ .CDL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9

ผลการทดลองและทดสอบ

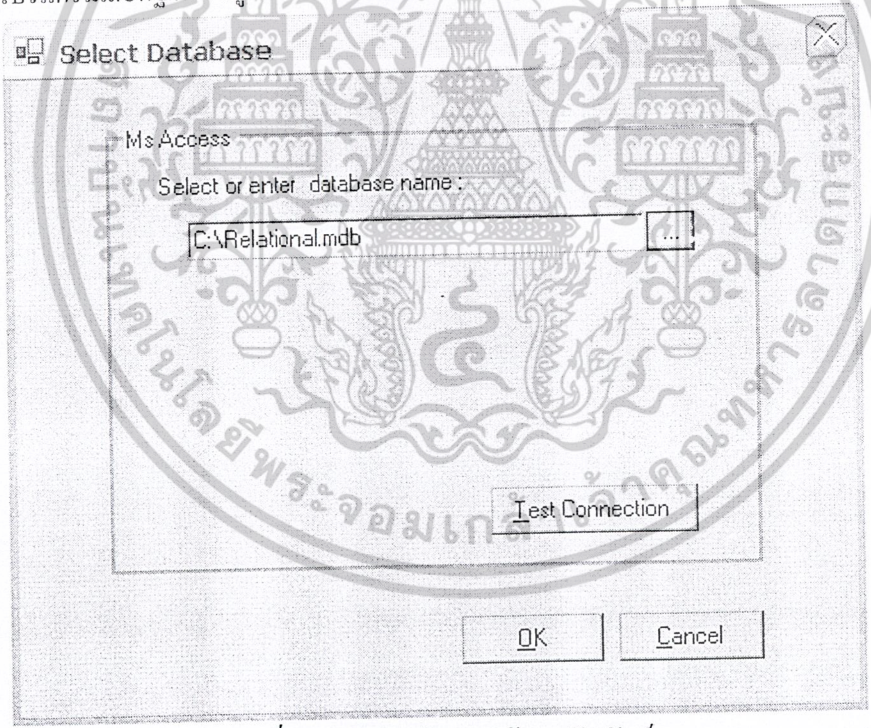
ทำการทดสอบโปรแกรมที่พัฒนาโดยนำโครงสร้างฐานข้อมูลเชิงสัมพันธ์รูปแบบต่างๆมาทดสอบ ซึ่งโปรแกรมที่โดยแบ่งการทดสอบออกเป็น 2 ส่วนคือ

1. ส่วนแปลงโครงสร้างฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองข้อมูลอี-อาร์
2. ส่วนแปลงแบบจำลองข้อมูลอี-อาร์ไปยังฐานข้อมูลเชิงวัตถุ

การทดสอบนั้น จะการแปลงโครงสร้างฐานข้อมูลเชิงสัมพันธ์ MS Access ไปเป็นแบบจำลองข้อมูลอี-อาร์ และนำแบบจำลองข้อมูลอี-อาร์ที่ได้มาแปลงเป็นโครงสร้างฐานข้อมูลเชิงวัตถุ Cache

9.1 ทดสอบการแปลงโครงสร้างฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองข้อมูลอี-อาร์

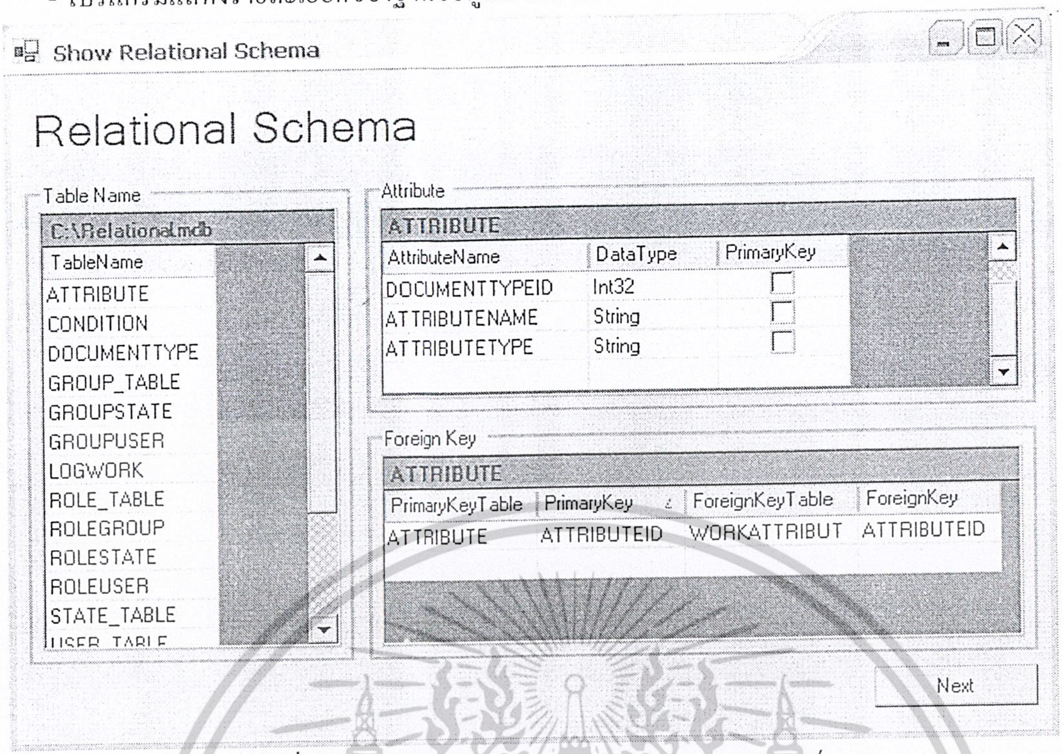
- ใช้โปรแกรมเลือกฐานข้อมูลเชิงสัมพันธ์ ที่ต้องการแปลง



รูปที่ 9.1 หน้าจอเลือกฐานข้อมูลสัมพันธ์

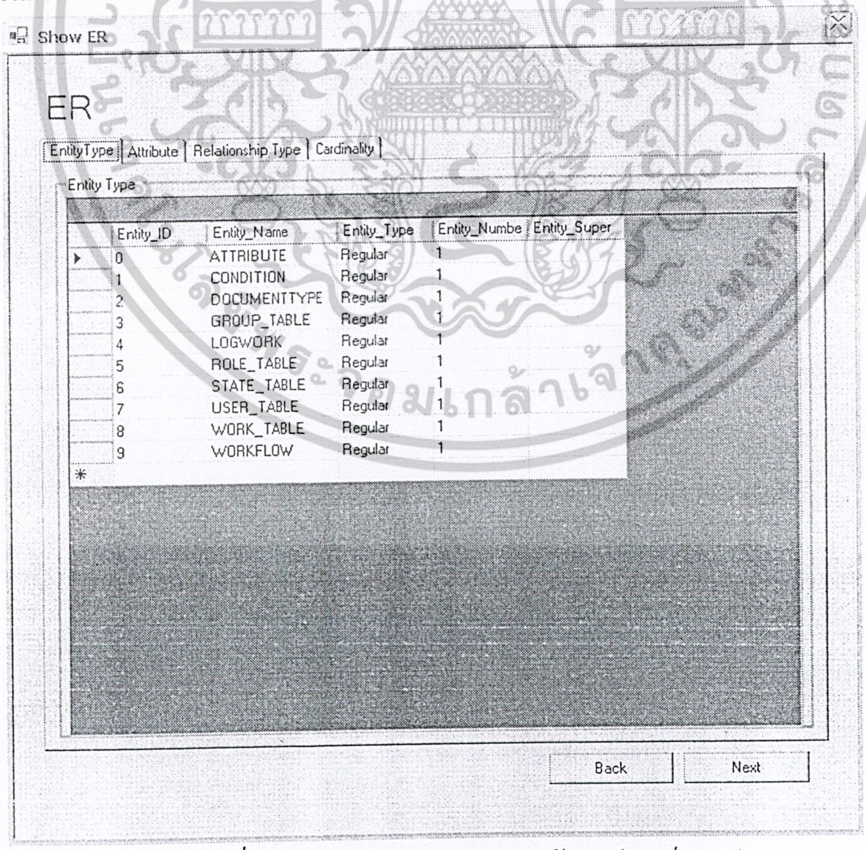
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โปรแกรมแสดงรายละเอียดของฐานข้อมูลเชิงสัมพันธ์



รูปที่ 9.2 หน้าจอแสดงโครงสร้างฐานข้อมูลเชิงสัมพันธ์

- โปรแกรมแสดงรายละเอียดของแบบจำลองอี-อาร์ ที่แปลงจากฐานข้อมูลเชิงสัมพันธ์



รูปที่ 9.3 หน้าจอแสดงแบบจำลองข้อมูลอี-อาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

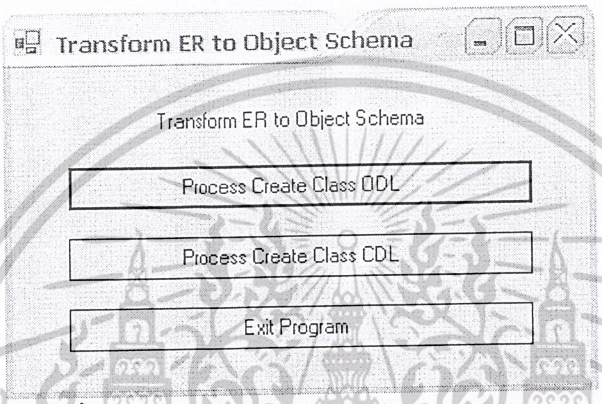
- ผลการทดสอบ

หลังจากทดลองการแปลงโครงสร้างฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองข้อมูลอี-อาร์ สามารถตรวจสอบ เอนทิตี เอนทิตีอ่อนแอ ความสัมพันธ์แบบหนึ่งต่อหนึ่ง ความสัมพันธ์แบบหนึ่งต่อกลุ่ม ความสัมพันธ์แบบกลุ่มต่อกลุ่ม แอททริบิวต์ปกติ แอททริบิวต์หลายค่า ได้

9.2 ทดสอบการแปลงแบบจำลองข้อมูลอี-อาร์ไปยังฐานข้อมูลเชิงวัตถุ

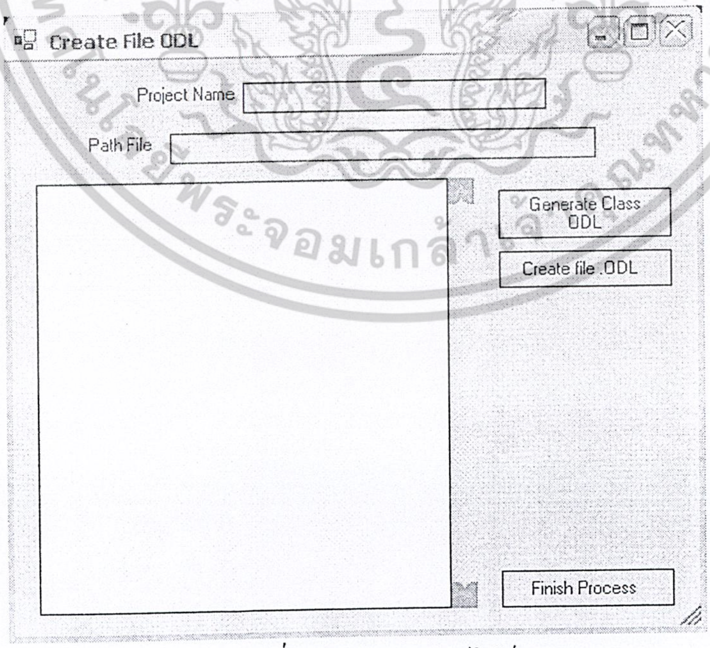
หลังจากที่ได้แบบจำลองข้อมูลอี-อาร์แล้ว ทำการแปลงไปเป็นฐานข้อมูลเชิงวัตถุ

- ใช้โปรแกรมเลือกโหมดรูปแบบของไฟล์ฐานข้อมูลเชิงวัตถุที่ต้องการแปลง โดยมีให้เลือก 2 แบบ คือ ODL กับ CDL



รูปที่ 9.4 หน้าจอเลือกรูปแบบไฟล์ฐานข้อมูลเชิงวัตถุ

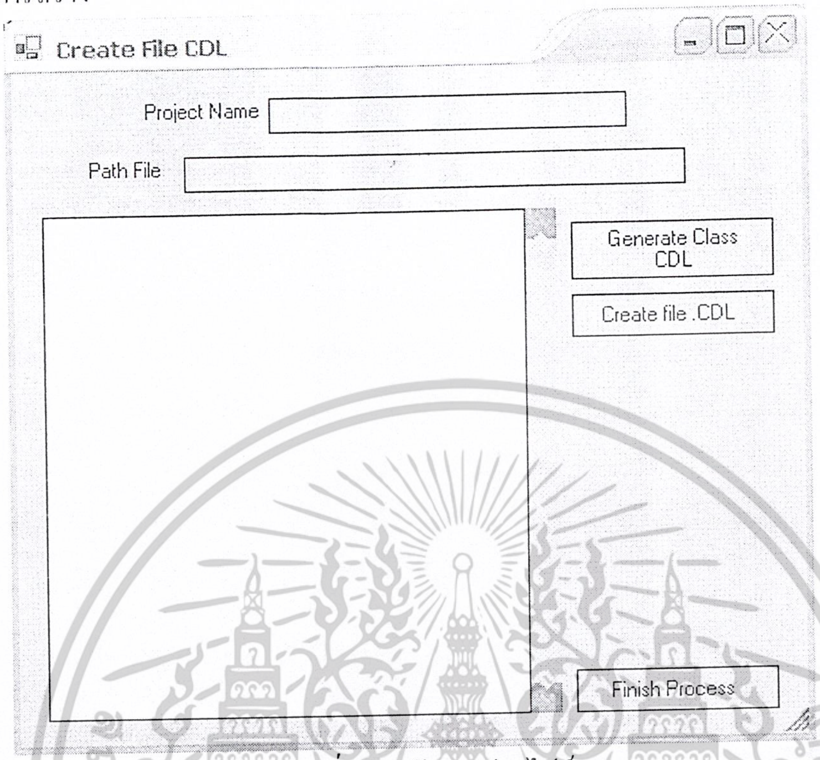
1) โหมดการแปลงไฟล์ ODL โดยจะเข้าหน้าจอ Create File ODL เราใช้ชื่อของโปรเจค และพาธของไฟล์ ODL ก่อนการสร้าง



รูปที่ 9.5 หน้าจอแปลงไฟล์ ODL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) โหมดการแปลงไฟล์ CDL โดยจะเข้าหน้าจอ Create File CDL เราใช้ชื่อของโปรเจค และพาธของไฟล์ ODL ก่อนการสร้าง



รูปที่ 9.6 หน้าจอแปลงไฟล์ CDL

- ผลการทดสอบ

หลังจากทดลองการแปลงแบบจำลองข้อมูลอี-อาร์ไปเป็นฐานข้อมูลเชิงวัตถุ โปรแกรมนี้สามารถสร้างไฟล์ฐานข้อมูลเชิงวัตถุได้สองแบบคือ ไฟล์ ODL กับไฟล์ CDL โดยไฟล์ ODL เป็นไฟล์มาตรฐานของการสร้างคลาสในฐานข้อมูลเชิงวัตถุ แต่ไฟล์ CDL เป็นไฟล์ที่ใช้สร้างคลาสใน Cache' Object DBMS ซึ่งเรานำมาใช้ในการทดลองนี้ เพราะในคลาสของ CDL จะไม่สนับสนุนคุณสมบัติบางอย่างของมาตราฐานของการสร้างคลาสในฐานข้อมูลเชิงวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 10
บทสรุปและวิจารณ์

10.1 สรุปผลการดำเนินงาน

ในโครงการนี้ได้ทำการศึกษาการถ่ายโอนฐานข้อมูล ศึกษาการถ่ายโอนโครงสร้างของฐานข้อมูล การศึกษาหลักการวิศวกรรมย้อนกลับ การศึกษาการออกแบบฐานข้อมูลเชิงสัมพันธ์ และฐานข้อมูลเชิงวัตถุ และพัฒนาการถ่ายโอนฐานข้อมูล เพื่อให้ได้ขั้นตอนที่มีประสิทธิภาพในการถ่ายโอนฐานข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ

โครงการได้พัฒนาโปรแกรม เพื่อช่วยในการถ่ายโครงสร้างฐานข้อมูลเพื่อ เป็นการทดสอบแนวขั้นตอนในการถ่ายโอนฐานข้อมูล และมีเครื่องมือในเพื่อช่วยในการประหยัดเวลา โดยโปรแกรมที่ทำการพัฒนาสามารถประยุกต์ใช้ในการทำวิศวกรรมย้อนกลับฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองข้อมูลอี-อาร์ และถ้าหากมีอี-อาร์ก็สามารถแปลงไปเป็นโครงสร้างฐานข้อมูลเชิงวัตถุได้

10.2 แนวทางในการพัฒนาต่อไป

ในส่วนของการแปลงฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองข้อมูลอี-อาร์ นั้นอาจเพิ่มความสามารถในการตรวจสอบแอททริบิวต์ผสม รวมถึงการพัฒนาเพิ่มความสามารถจนสามารถแปลงฐานข้อมูลเชิงสัมพันธ์ไปยังแบบจำลองอีอี-อาร์

สามารถพัฒนาส่วนให้สามารถเพื่อจำนวนฐานข้อมูลเชิงสัมพันธ์ทำสามารถแปลงได้ และปรับปรุงหน้าจอการเชื่อมต่อกับผู้ใช้ ซึ่งอาจพัฒนาเป็นแผนภาพอี-อาร์ เพื่อสามารถปรับปรุงแบบจำลองข้อมูลอี-อาร์ เพื่อให้ได้แบบจำลองที่ตรงตามความต้องการของผู้ใช้ และสะดวกในการใช้งาน

สามารถนำบ้างส่วนโปรแกรมไปพัฒนาเพื่อประยุกต์ใช้กับงานด้านวิศวกรรมย้อนกลับได้ โดยอาจมีการพัฒนาโปรแกรมเพิ่มเพื่อสามารถแปลงจากฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงสัมพันธ์ที่ต่างยี่ห้อได้

อีกทั้งในส่วนการแปลงอี-อาร์ไปยังฐานข้อมูลเชิงวัตถุนั้น สามารถพัฒนาโปรแกรมเพิ่มเติมเพื่อรองรับฐานข้อมูลเชิงวัตถุหลายยี่ห้อได้ หรืออาจจะรองรับฐานข้อมูลเชิงวัตถุสัมพันธ์ด้วย

แนวทางในการวิจัยต่อไป การศึกษาการถ่ายโอนข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลเชิงวัตถุ การแปลงโปรแกรมที่ใช้กับฐานข้อมูลเชิงสัมพันธ์เพื่อใช้กับฐานข้อมูลเชิงวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] S. Monk., J. A. Mariani., B. Elgalal. and H. Campbell. 1996. Migration from relational to object-oriented databases. Information and Software Technology., Vol.38, pp. 467-475

[2] W. Meng., A. Kamada. and Y. Chang. 1995. Transformation of Relational Schemas to Object-Oriented Schemas. : IEEE

[3] A. Behm., A. Geppert. and K. R. Dittrich. 1997. On the migration of relational schemas and data to object-oriented database systems. In Proc. Of the 5th International Conf. on Re-Technologies in Information Systems Klagenfurt Austria December 1997

[4] S. Ramanathan. and J. Hodges. Reverse Engineering Relational to Object-Oriented Schemas.

[5] R. Elmasri and S.B. Navathe. 2000. Fundamentals of Database Systems. Addison Wesley Longman, 3 editions

[6] มหาวิทยาลัยสุโขทัยธรรมาธิราช. 2544. เอกสารการสอนชุดวิชาการจัดการระบบฐานข้อมูล หน่วยที่ 8 – 15. นนทบุรี. สำนักพิมพ์สุโขทัยธรรมาธิราช

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้