ระบบช่วยการออกแบบฐานข้อมูลแบบสัมพันธ์ ส่วนโครงสร้างข้อมูล

A RELATIONAL DATABASE DESIGNER GENERATOR PART

โดย

นายธนิต แกล้วเดชศรี

นายวิชัย อัจฉริยบุญยงค์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2534

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

007728

ปริญญานิพนธ์ปีการศึกษา 2534
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ผู้จัดทำ

1. นาย ธนิต แกล้วเดชศรี 31.1104

2. นาย วิทัย ตั้งนิรันดุสงค์ 31.1244

..... อาจารย์ที่ปรึกษา
(ดร. ศุภมิตร จิตตะปโสธร)

..... อาจารย์ที่ปรึกษา
(อาจารย์ ธนา หงษ์สุวรรณ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ส่วนโครงสร้างข้อมูล

ธนิต แกล้วเดชศรี
วิชัย อัจฉริยบุญยงค์

ดร.ศุภมิตร จิตตะยโศธร อาจารย์ที่ปรึกษา
อ.ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เล่าวถึง การออกแบบเพื่อสร้างซอฟแวร์ที่พัฒนาให้เป็น ระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ โดยอาศัยแบบจำลองข้อมูลไนแอม โดยมีวัตถุประสงค์เพื่อให้การออกแบบฐานข้อมูลเป็นไปได้อย่างมีประสิทธิภาพ สะดวกรวดเร็วและมีความถูกต้องแน่นอนยิ่งขึ้น

โครงงานนี้เป็นส่วนหนึ่งของโครงงานรวมคือ ระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ เป็นส่วนที่มีหน้าที่ในการสร้างตารางความสัมพันธ์ โดยอาศัยข้อมูลที่อยู่ในรูปของตารางเมต้า (META TABLE) ที่ได้มาจากโครงงานระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ส่วนออกแบบ มาผ่านส่วนของโปรแกรมที่ได้พัฒนาขึ้น โดยใช้ภาษาที่ร่วมกับระบบจัดการฐานข้อมูลออราเคิล ซึ่งรายละเอียดต่างๆ ได้แสดงไว้ในปริญญานิพนธ์ฉบับนี้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A Relaiotnal Database Designer Generator Part

THANIT KLAEWDETSRI
VICHAI ARCHARIYABOONYONG

Dr.SUPHAMIT CHITTAYASOTHON ADVISOR
THANA HONGSUWAN ADVISOR

ABSTRACT

This project will introduce designning for develop a software to be a Relational Database designer by using NIAM with propose to more efficiency ,convenience and correctly database designning

This project is one part of " A Relational Database Designer " by using data ,that is output of " A Relational Database Designer Interface Part " which keep in META TABLE ,passes a part of developed program by using C language and ORACLE Database Management System. Detail presented in this thesis.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณ อาจารย์ ศุภมิตร จิตตะยโศธร ที่เป็นผู้ประสาทวิชาเกี่ยวกับฐานข้อมูล เป็นบุคคลแรก จนเราทั้งมีความรู้ทางด้านนี้พอสมควร นอกจากนี้ท่านยังเป็นอาจารย์ที่ปรึกษาของโครงงานด้วย

ขอขอบคุณ อาจารย์ ธนา หงษ์สุวรรณ ที่ช่วยแก้ปัญหาที่ติดขัดและให้ความช่วยเหลืออย่างดีตลอดมา และเพื่อนๆร่วมอาจารย์ที่ปรึกษาเดียวกันทุกคนที่มีส่วนช่วยแก้ปัญหาบ้าง ในบาง โอกาส

ขอขอบคุณคุณชวัชชัย สุทธิทศธรรม ที่ได้กรุณาช่วยให้คำแนะนำโดยทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
บทที่ 1	บทนำ	1
บทที่ 2	ทฤษฎีและหลักการเบื้องต้น	4
	2.1 ระบบฐานข้อมูล	4
	2.2 ระบบฐานข้อมูลแบบสัมพันธ์	6
	2.3 โมเดลแบบความสัมพันธ์	7
	2.4 โครงสร้างแบบสัมพันธ์	7
	2.5 ความถูกต้องของข้อมูล	11
	2.6 การจัดการข้อมูล	13
	2.7 การออกแบบฐานข้อมูลโดยใช้วิธีนอร์มาไลซ์เซชัน	15
	2.8 การออกแบบฐานข้อมูลโดยใช้เอนติตี้รีเลชันชิฟ	15
	2.9 แบบจำลองข้อมูลไนแอม	16
บทที่ 3	วิธีการวิเคราะห์ข้อมูลของไนส์เซน	17
	3.1 ส่วนประกอบต่าง ๆ ของไนแอม	17
	3.2 ขั้นตอนการออกแบบจำลองข้อมูลไนแอม	22
	3.3 การสร้างเค้าร่างฐานข้อมูลแบบสัมพันธ์ โดยวิธีการแปลงแบบจำลองระดับแนวคิดของไนแอม	50
บทที่ 4	ระบบส่วนหลักแบบฐานข้อมูลแบบสัมพันธ์	53
	4.1 เมต้าสกีมม่า	54
	4.2 เอนติตี้ไนไนแอมคอนเซ็ปชวลเมต้าสกีมม่า	54
	4.3 ความสัมพันธ์ระหว่างเอนติตี้ไทป์ไนไนแอมคอนเซ็ปชวลเมต้าสกีมม่า	57
	4.4 เมต้าสกีมม่าในลักษณะตารางรีเลชันบนฐานข้อมูลแบบสัมพันธ์ออราเคิล	60
	4.5 แมปปิ้งอินเตอร์เมชัน	64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ต่อ)

4.6 แมปปิ้งอินฟอร์เมชันในลักษณะตารางรีเลชันบนฐานข้อมูลแบบสัมพันธ์ออราเคิล 65
4.7 ข้อมูลรายละเอียดของแอตทริบิวท์ต่าง ๆ ในตารางเมต้าสกีมม่าและแมปปิ้งอินฟอร์เมชัน 66
4.8 วัตถุประสงค์ของโครงงานระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ 72
4.9 ส่วนประกอบของโครงงาน 72

บทที่ 5 ระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ส่วนโครงสร้างข้อมูล..... 75
5.1 สภาวะแวดล้อมทางฮาร์ดแวร์และซอฟแวร์ที่ใช้ในการพัฒนา 75
5.2 ส่วนประกอบของระบบ 76
5.3 ขั้นตอนในการพัฒนาโปรแกรม 76
5.4 การออกแบบโครงสร้างของระบบ 77

บทที่ 6 การใช้งานระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์
ส่วนโครงข้อมูล 87
6.1 การใช้งานระบบออกแบบฐานข้อมูลแบบสัมพันธ์ส่วนโครงสร้างข้อมูล 87
6.2 การใช้งานส่วนตรวจสอบความถูกต้องของข้อมูลตามคอนสเตรนท์ 90
6.3 ตัวอย่างระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ส่วนโครงสร้างข้อมูล 90

บทที่ 7 บทสรุปและวิจารณ์..... 98

บทที่ 8 แนวทางการพัฒนาต่อไป..... 99

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ต่อ)

ภาคผนวก

ภาคผนวก ก. โปรสตาร์ซี ก 1

ภาคผนวก ข. การจัดการฐานข้อมูลแบบสัมพันธ์ออราเคิล ข 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป	หน้า
รูปที่ 2.1 สถาปัตยกรรมของระบบฐานข้อมูล	5
รูปที่ 2.2 แสดงตารางความของข้อมูล supplier	7
รูปที่ 2.3 แสดงตัวอย่างตารางที่มีกลุ่มซ้ำและไม่มี	10
รูปที่ 2.5 แสดงตัวอย่างระดับข้อมูลในระบบใช้งาน	14
รูปที่ 3.1 แสดงชนิดเอนติตี้	17
รูปที่ 3.2 แสดงรูปชนิดของเลเบ้ล	18
รูปที่ 3.3 แสดงชนิดของแฟกท์	18
รูปที่ 3.4 แสดงชนิดของเรฟเฟอเรนจ์	18
รูปที่ 3.5 แสดงคอนสเตรนท์แบบยูนีคเนสส์ภายใน	19
รูปที่ 3.6 แสดงคอนสเตรนท์แบบยูนีคเนสส์ภายนอก	19
รูปที่ 3.7 แสดงคอนสเตรนท์แบบแมนดาทอรี่	19
รูปที่ 3.8 แสดงคอนสเตรนท์แบบเอ็กซ์คลูชัน	20
รูปที่ 3.9 แสดงคอนสเตรนท์แบบอิควอลิตี้	20
รูปที่ 3.10 แสดงคอนสเตรนท์แบบสับเซต	21
รูปที่ 3.11 แสดงคอนสเตรนท์แบบออคเคอเรนซ์ฟรีเควนซี่	21
รูปที่ 3.12 แสดงคอนสเตรนท์แบบเรนจ์	21
รูปที่ 3.13 แสดงคอนสเตรนท์แบบเมมเบอร์ชิพ	22
รูปที่ 3.14 แสดงตัวอย่างข้อมูลของตารางเรียน	23
รูปที่ 3.15 แสดงความสัมพันธ์ของข้อมูลในตารางด้วยรูปภาพ	26
รูปที่ 3.16 แสดงแผนภาพจำลองของข้อมูลระดับแนวคิด	27
รูปที่ 3.17 แสดงการเทียบแบบจำลองข้อมูลที่มีเรฟเฟอร์เรนซ์ไทป์ เป็นแบบ 1:1	27
รูปที่ 3.18 แสดง Schema-Based Diagram	28
รูปที่ 3.19 แสดงแฟกท์ไทป์ของผลการเรียน	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ต่อ)

รูป หน้า

รูปที่ 3.20 แสดงการเทียบแฟกท์ไทป์ของผลการเรียนในรูปของ Nesting 29
รูปที่ 3.21 แสดงแผนภาพจำลองข้อมูลระดับแนวคิด 30
รูปที่ 3.22 แสดงแผนภาพจำลองข้อมูลระดับแนวคิด 31
รูปที่ 3.23 แสดงรูปแบบที่ผิดของแบบจำลองข้อมูล 32
รูปที่ 3.24 แสดงผลของการนำขั้นตอนที่ 3 มาใช้แก้ไข 32
รูปที่ 3.25 แสดงตัวอย่างของการใช้งานคอนสเตรนท์แบบยูนีคเนสส์ 34
รูปที่ 3.26 แสดงแบบต่างๆของคอนสเตรนท์แบบยูนีคเนสส์ สำหรับแฟกท์ไทป์แบบไบนารี่ 34
รูปที่ 3.27 แสดงคอนสเตรนท์แบบยูนีคเนสส์ของเทอร์นารี่ 35
รูปที่ 3.28 แสดงแบบจำลองข้อมูลที่ได้จากตาราง 35
รูปที่ 3.29 แสดงคอนสเตรนท์แบบยูนีคเนสส์ภายนอก 36
รูปที่ 3.30 แสดงแบบจำลองข้อมูลที่ได้จากตารางที่ 3.5 38
รูปที่ 3.31 แสดงตัวอย่างของแบบจำลองข้อมูลของผลการแข่งกีฬา 40
รูปที่ 3.32 แสดงการใช้ คอนสเตรนท์แบบเอนติตี้ไทป์ 41
รูปที่ 3.33 แสดงการใช้งานคอนสเตรนท์แบบแมนดาทอรี่ 42
รูปที่ 3.34 แสดงความสัมพันธ์ระหว่างซูเปอร์ไทป์และสับไทป์ 43
รูปที่ 3.35 แสดงการใช้สับไทป์ในแบบจำลองข้อมูล 44
รูปที่ 3.36 แสดงตัวอย่างการใช้ออคเคอเรนซ์ ฟรีเควนซี่ 44
รูปที่ 3.37 แสดงตัวอย่างการใช้งานออคเคอเรนซ์ ฟรีเควนซี่ 45
รูปที่ 3.38 แสดงตัวอย่างของออคเคอเรนซ์ ฟรีเควนซี่ 45
รูปที่ 3.39 แสดงตัวอย่างของแบบจำลอง 47
รูปที่ 3.40 แสดงการอ้างถึงแบบไฮราคี่ 48
รูปที่ 3.41 แสดงแบบจำลองข้อมูลของตารางที่ 3.12 49
รูปที่ 4.1 แผนภูมิแสดงระดับในการแทนข้อมูลในระดับแนวคิด 53
รูปที่ 4.2 แสดง NIAM-Conceptual Metashema 69
รูปที่ 4.3 Mapping Information No1 70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ต่อ)

รูป		หน้า
รูปที่ 4.4	Mapping Information No2	71
รูปที่ 4.5	REDD - System Diagram	74
รูปที่ 5.1	แสดงโครงสร้างของโนดตาราง	79
รูปที่ 5.2	แสดงโครงสร้างของโนดคอลัมน์	80
รูปที่ 5.3	แสดงโครงสร้างของโนดวีเลชั่น	80
รูปที่ 5.4	แสดงโครงสร้างของความสัมพันธ์ของลิงก์ลิสต์	81
รูปที่ 5.5	แสดงความสัมพันธ์ของการเรียกใช้ในแต่ละไฟล์ของโปรแกรมส่วนสร้างความสัมพันธ์	83
รูปที่ 5.6	แสดงความสัมพันธ์ของการเรียกใช้ในแต่ละไฟล์ของโปรแกรมส่วนเอนไวรอนเมนท์	84
รูปที่ 6.1	แสดงตัวอย่างไนแอมรูปที่ 1	91
รูปที่ 6.2	แสดงตัวอย่างไนแอมรูปที่ 2	92
รูปที่ 6.3	แสดงตัวอย่างไนแอมรูปที่ 3	95
รูปที่ 8.1	แสดงการอ้างถึงแบบไฮราคี่แบบมีสับไทป์	100
รูปที่ 8.2	แสดงเรฟเฟอร์เรนซ์ไทป์แบบหาได้	101
รูปที่ 8.3	แสดงเนสท์ 2 ชั้น	102

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 3.1 แสดงตัวอย่างข้อมูลความสัมพันธ์ของรถและคนขับ	25
ตารางที่ 3.2 แสดงตัวอย่างข้อมูลของผลการเรียนนักศึกษา	28
ตารางที่ 3.3 แสดงตัวอย่างข้อมูลรายการขายสินค้า	30
ตารางที่ 3.4 แสดงตัวอย่างข้อมูล	35
ตารางที่ 3.5 แสดงตัวอย่างข้อมูลของผลการเรียน	37
ตารางที่ 3.6 แสดงตัวอย่างข้อมูล	39
ตารางที่ 3.7 แสดง Projection ของข้อมูลในตารางที่ 3.8	39
ตารางที่ 3.8 แสดงผลของการ Join ข้อมูลในตารางที่ 3.8	39
ตารางที่ 3.9 แสดงข้อมูลของคนไข้	42
ตารางที่ 3.10 แสดงตัวอย่างข้อมูล	43
ตารางที่ 3.11 แสดงตัวอย่างของข้อมูล	46
ตารางที่ 3.12 แสดงตัวอย่างข้อมูลของสโมสรกีฬาแห่งหนึ่ง	49
ตารางที่ 4.1 แสดงตารางของเมต้าสกีมม่า	64
ตารางที่ 4.2 แสดงตารางของแมปปิ้งอินฟอร์เมชัน	66
ตารางที่ 4.3 แสดงความหมายของคอลัมน์ต่าง ๆ ในตารางของเมต้า และในแมปปิ้งอินฟอร์เมชัน	68
ตารางที่ 5.1 แสดงวิว ENRELATION	78
ตารางที่ 5.2 แสดงตาราง ENCOLUMN	78
ตารางที่ 5.3 แสดงตาราง FACTENIUNIQ	79
ตารางที่ 5.4 แสดงตาราง CONSTIQUERY	85
ตารางที่ 5.5 แสดงตาราง CONSTIVIEW	85
ตารางที่ 6.1 แสดงตารางในตัวอย่างที่ 1	91
ตารางที่ 6.2 แสดงตารางในตัวอย่างที่ 2	93
ตารางที่ 6.3.1 แสดงตารางในตัวอย่างที่ 3 แบบที่ 1	95
ตารางที่ 6.3.2 แสดงตารางในตัวอย่างที่ 3 แบบที่ 2	96

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

การทำงานในส่วนใหญ่มักจะอาศัยความสามารถของมนุษย์เพื่อใช้งานเสร็จเรียบร้อย เนื่องจากมนุษย์สามารถติดต่อสื่อสารกัน สามารถแลกเปลี่ยนความรู้กันได้ นอกจากนี้มนุษย์ยังมีความสามารถในการเรียนรู้ สามารถเพิ่มพูนประสบการณ์ได้ แต่มนุษย์ก็ยังมีข้อเสียหลายประการ เช่น มนุษย์มีอายุอยู่ได้ไม่นาน มนุษย์เกิดความเบื่อหน่ายได้ง่าย เป็นต้น ดังนั้นเมื่อได้มีการคิดค้นประดิษฐ์อุปกรณ์ใหม่ที่สามารถทำงานแทนมนุษย์ในด้านต่าง อุปกรณ์หนึ่งที่มีความสำคัญในปัจจุบันและจะยังคงมีบทบาทต่อไปในอนาคต ก็คือ คอมพิวเตอร์ ปัจจุบัน คอมพิวเตอร์เข้ามามี บทบาทต่อการดำเนินชีวิตของมนุษย์มากขึ้น คอมพิวเตอร์สามารถใช้งานได้ดี จึงมีการนำคอมพิวเตอร์มาทำงานแทนมนุษย์ในด้านต่างๆ เนื่องจากคอมพิวเตอร์มีข้อดีหลายอย่าง แต่ในงานบางประเภทคอมพิวเตอร์ก็ไม่สามารถทำงานแทนมนุษย์ได้ จึงได้มีการพัฒนาความสามารถของคอมพิวเตอร์ต่อไป

ความสามารถของคอมพิวเตอร์ที่เหนือกว่ามนุษย์ก็คือ ความสามารถในการจัดการกับข้อมูล เช่น การจัดเก็บข้อมูล และการนำข้อมูลที่เก็บไว้มาใช้ เป็นต้น ถึงแม้ว่าคอมพิวเตอร์จะมีความสามารถในการจัดเก็บข้อมูล แต่ถ้าการจัดเก็บข้อมูลไม่เป็นระเบียบ และไม่มีประสิทธิภาพเพียงพอก็จะทำให้การนำข้อมูลกลับมาใช้มีความยุ่งยากซับซ้อน ใช้เวลานาน ก็เท่ากับว่าเป็นการใช้งานคอมพิวเตอร์ไม่เต็มที่ ดังนั้นนักคอมพิวเตอร์จึงได้คิดค้นรูปแบบในการเก็บข้อมูลให้มีประสิทธิภาพมากที่สุด ซึ่งรูปแบบในการจัดเก็บก็มีความแตกต่างกันไปตามการออกแบบ แต่ที่นิยมใช้ในปัจจุบันมีอยู่ด้วยกัน 3 แบบ คือ

1.แบบลำดับชั้น (Hierachical Model)

2.แบบเครือข่าย (Network Model)

3.แบบสัมพันธ์ (Relational Model)

ผลที่ได้จากการจัดเก็บข้อมูลตามรูปแบบต่างๆ ก็คือ ฐานข้อมูล (Database) ในการจัดเก็บข้อมูลและนำข้อมูลจากฐานข้อมูลมาใช้ ก็จำเป็นจะต้องมีซอฟแวร์มาดูแลความเรียบร้อย และความปลอดภัยของข้อมูล เนื่องจาก ถ้าให้มีการใช้งานข้อมูลได้อย่างอิสระก็จะทำให้เกิดความเสียหายแก่ข้อมูลได้ ซอฟแวร์ที่ทำหน้าที่นี้เรียกว่า ระบบจัดการฐานข้อมูล (DBMS : Database Mangement System)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในปัจจุบันรูปแบบที่นิยมกันคือ แบบสัมพันธ์ โดยข้อมูลที่เก็บในระบบตรรก (logical) ของฐานข้อมูลแบบสัมพันธ์ มีลักษณะเป็นตาราง (table) ซึ่งประกอบไปด้วย แถว (row) ซึ่งเรียกว่า ทัพเพิล (tuple) และคอลัมน์ (column) ซึ่งเรียกว่า แอททริบิวท์ (attribute) โดยซอฟแวร์ที่มีหน้าที่ในการจัดการฐานข้อมูลแบบสัมพันธ์ เรียกว่า ระบบจัดการฐานข้อมูลแบบสัมพันธ์ (RDBMS: Relational Database Management System) ได้แก่ ออราเคิล (ORACLE) เป็นต้น

การออกแบบฐานข้อมูลแบบสัมพันธ์นั้น มีวิธีการออกแบบอยู่หลายวิธี ที่นิยมกันมี 3 วิธีคือ

1. วิธีการนอร์มาไลเซชัน (Normalization Method)

2. การใช้เอนทิตี้รีเลชันชิฟ (Entity-relationship Model)

3. วิธีวิเคราะห์ข้อมูลของไนส์เซ่น หรือไนแอม (NIAM : Nijssen's Information Analysis Method)

วิธีการนอร์มาไลเซชัน เป็นวิธีที่นิยมกันแพร่หลาย แต่เป็นวิธีที่ยุ่งยากซับซ้อน ต้องอาศัยความชำนาญของผู้ออกแบบ การใช้เอนทิตี้รีเลชันชิฟ ก็เป็นวิธีที่แพร่หลาย ความยุ่งยากซับซ้อนน้อยกว่าวิธีการนอร์มาไลเซชัน วิธีวิเคราะห์ข้อมูลของไนส์เซ่น เป็นที่วิธีที่นิยมแพร่หลายแถบประเทศออสเตรเลีย ในโครงงานนี้จะใช้วิธีการออกแบบฐานข้อมูลแบบสัมพันธ์ โดยใช้วิธีไนแอม ซึ่งเป็นแบบจำลองข้อมูลที่ใช้ในการออกแบบนิยามข้อมูลระดับแนวคิด โดยอาศัยความสัมพันธ์ของข้อมูล

โครงงานระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ (A Relational Database Designer) หรือมีชื่อย่อว่า เรดด์(REDD) เป็นโครงงานที่กำหนดขึ้น เพื่อพัฒนาระบบการออกแบบฐานข้อมูลแบบสัมพันธ์ได้ง่ายดาย โครงงานของระบบ ประกอบด้วย 2 ส่วน คือ ส่วนออกแบบ และส่วนโครงสร้างข้อมูล แต่ละส่วนแบ่งออกเป็น 1 โครงงาน นั่นคือ โครงงานระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ส่วนออกแบบ และ โครงงานระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ส่วนโครงสร้างข้อมูล ซึ่งในโครงงานนี้เป็นส่วนโครงสร้างข้อมูล โดยงานรวมเป็นการแปลงไนแอมคอนเซปชวลสกีม่า ไปเป็นรีเลชันแนลสกีม่า ซึ่งอินพุท (input) ที่รับจากผู้ใช้เป็นรูปไนแอม จากนั้นโครงงานส่วนออกแบบจะทำการวิเคราะห์และเก็บความสัมพันธ์ของรูปให้อยู่ในตารางเมต้า ซึ่งเป็นการทำงานโดยย่อของโครงงานส่วนออกแบบ ส่วนงานของโครงงานนี้ จะมีอินพุทที่สำคัญ คือ ข้อมูลที่ได้จากโครงงานส่วนออกแบบหรือข้อมูลในตารางเมต้า มาทำการแปลงให้อยู่ในรูปของตารางความสัมพันธ์ ซึ่งเป็นเอาท์พุท (output) ที่สำคัญของโครงงานรวม นอกจากนี้ยังมีอีกส่วนหนึ่งที่ไม่ได้ร่วมอยู่ในโครงงานรวม แต่เป็นส่วนที่ร่วมมือกันระหว่างกลุ่มนี้กับกลุ่มโครงงานการพัฒนากฎในระบบฐานข้อมูล เรื่องคอนสเตรนท์แฮนดลิ้ง (Constraint handling) ซึ่งเป็นส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ทำการตรวจสอบความถูกต้องของข้อมูลในฐานข้อมูลแบบสัมพันธ์ที่ได้รับการกำหนดให้เป็นไปตามคอนสเตรนท์ (constraints) ในไนแอมคอนเซปชวลสกีมา

เนื้อหาโดยย่อของแต่ละบท

บทที่ 2 จะกล่าวถึงทฤษฎีและหลักการที่เกี่ยวข้องกับโครงงานนี้ทั้งหมด ทฤษฎีและหลักการที่สำคัญได้แก่ นิยามของ "ฐานข้อมูล" (Database) ซึ่งเป็นที่เก็บข้อมูลและความสัมพันธ์ของข้อมูล ปัญหาที่สำคัญคือการเก็บข้อมูลนั้นจะเก็บกันอย่างไร จึงจะเปลืองเนื้อที่ในการเก็บน้อยและให้ประสิทธิภาพในการค้นหาที่รวดเร็ว "ระบบฐานข้อมูล" (Database system) เป็นการนำคอมพิวเตอร์มาช่วยในการเก็บข้อมูลและความสัมพันธ์ของข้อมูล หรือฐานข้อมูลนั่นเอง ระบบฐานข้อมูลนั้นดังที่ได้กล่าวมาแล้วว่าด้วยกันหลายรูปแบบ ที่ใช้ในโครงงานนี้

บทที่ 3 วิธีการในการออกแบบฐานข้อมูลมีด้วยกันหลายวิธีดังได้กล่าวมาแล้ว แต่ละวิธีมีข้อดีและข้อเสียที่แตกต่างกันไป โดยวิธีที่ใช้เป็นวิธีที่อ้างอิงในการออกแบบฐานข้อมูลแบบสัมพันธ์ในโครงงานนี้จะเป็นการวิธีไนแอม ซึ่งมีรูปแบบเฉพาะตัว และมีขั้นตอนต่างๆ ในการออกแบบอยู่ด้วยกัน 9 ขั้นตอน ในแต่ละขั้นตอนมีรายละเอียดในการออกแบบเป็นอย่างไร ซึ่งได้กล่าวไว้ในปริญญานิพนธ์นี้แล้ว

บทที่ 4 ระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ ซึ่งได้กล่าวมาบ้างแล้วว่าประกอบด้วยโครงงาน 2 ส่วนคือ ส่วนออกแบบและส่วนโครงสร้างข้อมูล แต่ละขั้นตอนมีการทำงานที่ต่อเนื่องกัน

บทที่ 5 ระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ส่วนโครงสร้างข้อมูล เป็นส่วนหนึ่งของโครงงานระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ โดยเป็นส่วนที่ต่อเนื่องมาจากส่วนออกแบบ ดังที่ได้กล่าวมาแล้วคร่าวๆ ในการออกแบบโครงสร้างของระบบทั้งโครงสร้างของข้อมูลและโครงสร้างของรูทีนการทำงานเพื่อการพัฒนา และการใช้งานโปรแกรม

บทที่ 6 เป็นบทที่แสดงตัวอย่างของการใช้งานระบบ โดยจะมีรูปไนแอมเป็นรูปอินพุท และจะมีรูปตารางที่ผ่านกระบวนการแปลงแล้ว และรายงานในภาษาเอสคิวเอล

บทที่ 7 เป็นบทที่อธิบายสรุปผลการทำงาน และวิจารณ์การทำงานของระบบที่สร้างขึ้น นอกจากนี้ยังมีการกล่าวหาปัญหาที่เกิดขึ้นระหว่างการทำงาน ซึ่งมีปัญหาต่างๆมากมาย

บทที่ 8 เป็นการกล่าวถึงสิ่งที่เป็นการทำงานเพื่อพัฒนาโครงงานให้ทำงานได้ดีขึ้น และแก้ไขปัญหาที่เกิดขึ้นดังที่กล่าวในบทที่ 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการเบื้องต้น

2.1 ระบบฐานข้อมูล (Database System)

ลักษณะสถาปัตยกรรมของระบบฐานข้อมูล ได้ถูกกำหนดให้เป็นมาตรฐานจาก 3 องค์กรหลัก คือ ไอโซ (ISO : International Standard Organizaton) , ไอฟิป (IFIP : International Federation for Information Processing) และ แอนซี (ANSI : American National Standrad Institute) โดยสามารถจัดแบ่งออกเป็น 3 ระดับด้วยกัน คือ

1. นิยามข้อมูลระดับภายนอก (External Schema)
   เป็นโครงสร้างข้อมูลที่ผู้ใช้แต่ละคนมองเห็น
2. นิยามข้อมูลระดับแนวความคิด (Conceptual Schema)
   เป็นโครงสร้างข้อมูลที่นักออกแบบฐานข้อมูลมองเห็นซึ่งจะเป็นข้อมูลทั้งหมดของระบบฐานข้อมูล
3. นิยามข้อมูลระดับภายใน (Internal Schema)
   เป็นโครงสร้างข้อมูลที่จัดเก็บในอุปกรณ์เก็บข้อมูล เช่น ลักษณะการเก็บข้อมูล

ซอฟแวร์ที่มีสถาปัตยกรรมตามมาตรฐานของไอโซ คือ ทำหน้าที่ควบคุมดูแลข้อมูลในฐานข้อมูลทั้ง 3 ระดับดังกล่าว เรียกว่า ระบบจัดการฐานข้อมูล (DBMS : Database Management System)

รายละเอียดส่วนต่างๆเพิ่มเติม ได้แก่

- User คือ ผู้ใช้งานขั้นสุดท้าย (End-user) , นักโปรแกรม (programmer) และโปรแกรมใช้งาน
- Physical Database คือ อุปกรณ์เก็บข้อมูล เช่น ฮาร์ดดิสก์ (hard disk)

ฐานข้อมูลสามารถจำแนกออกเป็นรูปแบบต่างๆ ตามลักษณะการแสดงความสัมพันธ์ระหว่างข้อมูล ในปัจจุบันนี้มีที่นิยมแพร่หลายอยู่ 3 รูปแบบด้วยกัน คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. แบบโครงข่าย (Network Model)

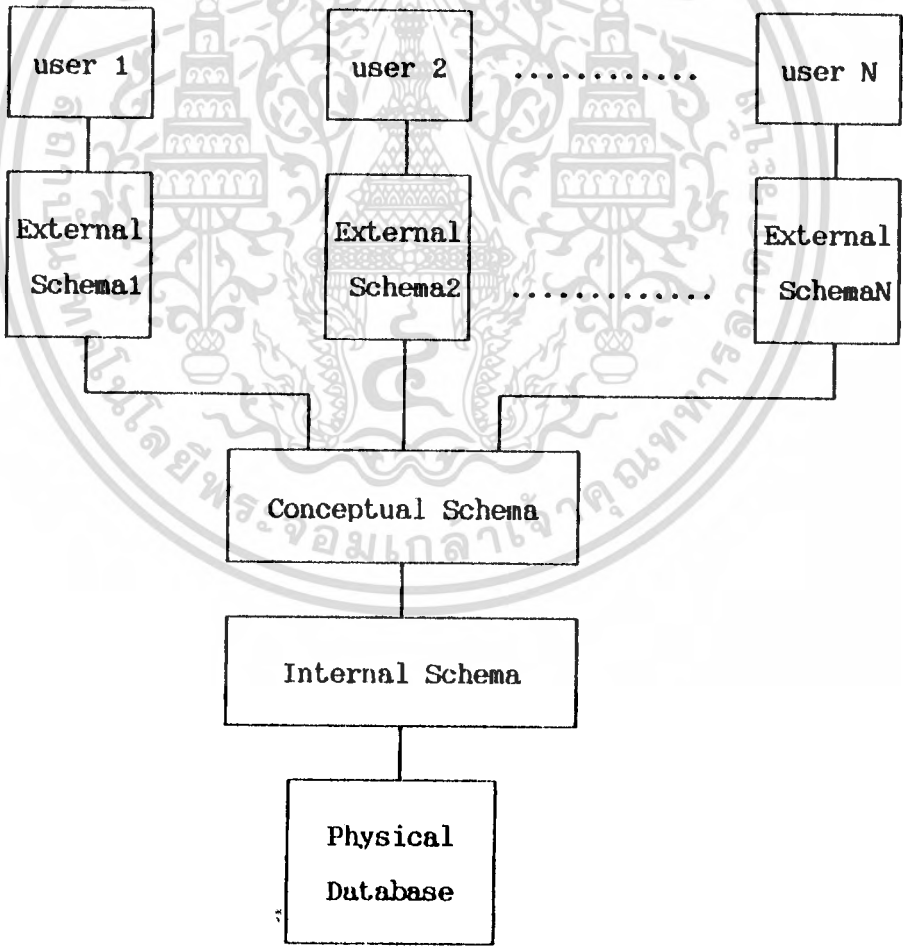
อยู่ในรูปของการรวบรวมเรคอร์ด (record) ต่างๆ และแสดงความสัมพันธ์ระหว่างเอนทิตี้ (entity) อย่างชัดเจน ฐานข้อมูลแบบนี้มีความใกล้เคียงความจริงมาก เนื่องจากความสัมพันธ์ระหว่างข้อมูลของระบบนี้ง่ายต่อการจัดรูปแบบ

2. แบบแผนภูมิต้นไม้ (Hierachical Model)

มีลักษณะเป็นรูปต้นไม้กลับหัว ที่แสดงความสัมพันธ์ระหว่างเอนทิตี้ลดหลั่นกันลงในลักษณะ วันทูเมนี่ (1:n , one to many) ข้อมูลแต่ละเอนทิตี้จะ สัมพันธ์กับข้อมูลที่อยู่เหนือขึ้นไปเพียงเอนทิตี้เดียว

3. แบบสัมพันธ์ (Relational Model)

เป็นการเก็บข้อมูลที่มีลักษณะเหมือนตาราง (table) 2 มิติธรรมดา คือ มีแถว (row) , คอลัมน์ (column) และข้อมูลในทุกแถว แต่ละคอลัมน์ใช้เป็นส่วนหนึ่งของการค้นหาข้อมูลได้



รูปที่ 2.1 สถาปัตยกรรมของระบบฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบจัดการฐานข้อมูล แบ่งภาษาที่ใช้ภายในออกเป็น 2 ส่วนด้วยกัน คือ

1. ภาษาที่ใช้จัดการกับข้อมูล (DML : Data Manipulation Language) โดยตัวภาษามีหน้าที่

1.1 สอบถามหรือเรียกค้นข้อมูล (Query)

1.2 เพิ่มเติมข้อมูล (Insert)

1.3 ลบข้อมูล (Delete)

1.4 แก้ไขเปลี่ยนแปลงข้อมูล (Update)

2. ภาษาที่ใช้ในการกำหนดโครงสร้างของข้อมูล (DDL : Data Definition Language) มี หน้าที่ในการกำหนดโครงสร้างของข้อมูลที่ผู้ใช้แต่ละคนเห็น , โครงสร้างข้อมูลที่นักออกแบบฐานข้อมูลมองเห็น และโครงสร้างข้อมูลที่จัดเก็บในอุปกรณ์เก็บข้อมูล

2.2 ระบบจัดการฐานข้อมูลแบบสัมพันธ์ (Relational Database Management System)

ระบบจัดการฐานข้อมูลแบบสัมพันธ์ได้ถูกคิดค้นและพัฒนาขึ้นมาโดยดร.เอดการ์ เอฟ.คอดด์ (Edger F.Codd) มีการนำเสนอขึ้นมาครั้งแรกในปี ค.ศ. 1969 ขณะที่เขาทำงานอยู่ที่ไอบีเอ็ม (IBM : International Business Machine) ระบบจัดการฐานข้อมูลแบบสัมพันธ์มีพื้นฐานมาจากคณิตศาสตร์ของเซต (Set theory) และมีภาษา SQL (Structure Query Language) เป็นภาษาที่ใช้จัดการกับข้อมูล และภาษาที่ใช้ในการกำหนดโครงสร้างของข้อมูลให้กับฐานข้อมูลแบบสัมพันธ์

ในปัจจุบันฐานข้อมูลส่วนใหญ่เป็นระบบฐานข้อมูลแบบสัมพันธ์ทั้งสิ้น เนื่องจากเป็นโมเดลซึ่งง่ายต่อการทำความเข้าใจ และสามารถตอบสนองต่อคำถาม (query) ที่ซับซ้อนได้ดีกว่าโมเดลชนิดอื่น ตัวอย่างของระบบจัดการฐานข้อมูลแบบสัมพันธ์ ได้แก่

ผลิตภัณฑ์	ผู้ขาย
DB2	IBM
SQL/DS	IBM
INGRES	Relational Technology Inc.
ORACLE	ORACLE Corp.
Rdb/VMS	DEC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 โมเดลแบบสัมพันธ์ (Relational model)

การที่จะกล่าวว่าระบบฐานข้อมูลใด เป็นโมเดลแบบสัมพันธ์นั้น ต้องครบถ้วนในองค์ประกอบทั้งสามส่วน อันได้แก่

1. โครงสร้างข้อมูล (Data structure)

   เป็นไปตามนิยามคุณสมบัติของรีเลชัน (relations) (กล่าวโดยละเอียดในหัวข้อ "โครงสร้างข้อมูลแบบสัมพันธ์")

2. ความถูกต้องของข้อมูล (Data integrity)

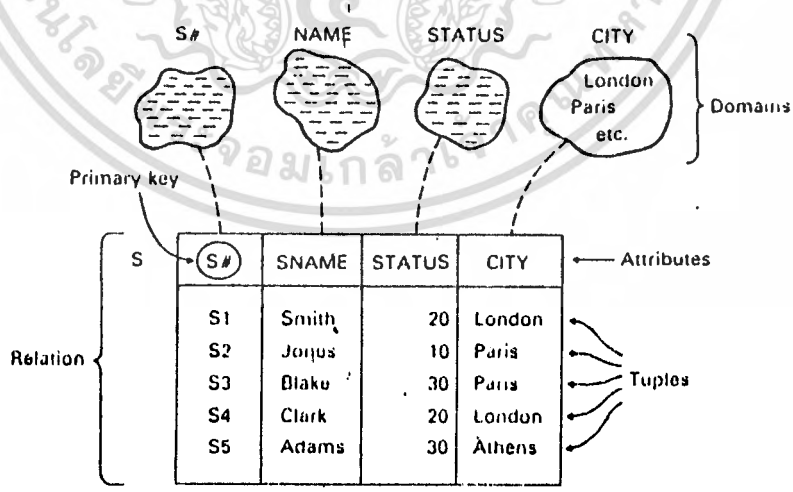
   เป็นไปตามกฎความถูกต้องทั้งสอง (กล่าวโดยละเอียดในหัวข้อ"กฎของความถูกต้องของข้อมูล")

3. การจัดการข้อมูล (Data manipulation)

   เป็นไปตามนิยามการจัดการข้อมูลในรีเลชัน (กล่าวโดยละเอียดในหัวข้อ"การจัดการข้อมูล")

2.4 โครงสร้างข้อมูลแบบสัมพันธ์ (Relational data structure)

2.4.1 ตัวอย่างตารางความสัมพันธ์



รูปที่ 2.2 แสดงตารางความสัมพันธ์ข้อมูล supplier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.2 มีคำเฉพาะที่จะใช้ในการเรียกขานส่วนต่าง ๆ ของตารางข้อมูล ดังนี้

- โดเมน (domain) หมายถึง หน่วยข้อมูลที่เก็บอยู่ในตารางความสัมพันธ์
- แอตตริบิวท์ (attribute) หมายถึง ชื่อคอลัมน์ของตารางความสัมพันธ์
- ทัพเพิล (tuple) หมายถึง แถวข้อมูลใน 1 เรคคอร์ดของตารางความสัมพันธ์
- ไพรมารี่คีย์ (primary key) หมายถึง แอตตริบิวท์หรือกลุ่มแอตตริบิวท์ซึ่งมีข้อมูลที่มีลักษณะเฉพาะ กล่าวคือ ในเวลาใด ๆ ก็ตาม จะไม่ปรากฏว่ามีแถวข้อมูลสองแถวที่อยู่ในแอตตริบิวท์หรือกลุ่มแอตตริบิวท์ดังกล่าว มีข้อมูลเหมือนกันทุกประการ (รายละเอียดเรื่องไพรม่ารี่คีย์อยู่ในหัวข้อ "ความถูกต้องของข้อมูล")

2.4.2 รีเลชั่น (Relations)

หมายถึง ผลคูณคาร์ทีเซียนของโดเมนที่สนใจในแอพพลิเคชั่น ประกอบขึ้นด้วยองค์ประกอบ 2 ส่วน คือ เฮ็ดดิ้ง (heading) และ บอดี้ (body) โดยเรากำหนดให้ D1,D2,....,Dn แทนโดเมนทั้งหลายในระบบงาน

- เฮ็ดดิ้ง (heading) ประกอบขึ้นมาจากเซตที่มีขนาดคงที่ (fixed set) ของแอตตริบิวท์ A1,A2,....,An โดย Ai คือ แอตตริบิวท์ที่แทนข้อมูลในโดเมนที่ Di
- บอดี้ (body) ประกอบขึ้นจากเซตของทัพเพิลที่มีขนาดแปรผันตามเวลาของรีเลชั่น โดยแต่ละทัพเพิลประกอบด้วยค่าข้อมูลของแอตตริบิวท์ต่าง ๆ เช่น Ai:vi แทนค่าที่ i ของแอตตริบิวท์ที่ i

หนึ่งทัพเพิลประกอบขึ้นด้วยค่าข้อมูลของแอตตริบิวท์ที่อ้างอิงโดเมนต่างๆ ในตาราง จากรูปที่ 2.2 เขียนได้เป็น

```
( S#     :  'S1'     )
( SNAME  :  'Smith'  )
( STATUS :  20       )
( CITY   :  'London' )
```

ตารางความสัมพันธ์ที่มีจำนวนแอตตริบิวท์ n แอตตริบิวท์ จะเรียกว่ามีดีกรีขนาด n (n degree) เรียกรีเลชั่นที่มีเพียงแอตตริบิวท์เดียวว่า ยูนารี่ (unary) รีเลชั่นที่มีสองแอตตริบิวท์ว่า ไบนารี่ (binary) และ ขนาดแอตตริบิวท์สามว่า เทอร์นารี่ (ternary) และหากมีดีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดีขนาด n ก็เรียกว่า เอนนารี่ (n-ary)

คุณสมบัติของรีเลชัน พิจารณาจากข้อกำหนด 4 ประการ คือ

1. ต้องไม่มีทัพเพิลที่ซ้ำกันภายในรีเลชัน
2. ลำดับของทัพเพิลไม่มีความสำคัญในการเก็บ
3. ลำดับของแอตตริบิวท์ไม่มีความสำคัญในการเก็บ
4. ค่าข้อมูลของแต่ละแอตตริบิวท์ต้องเป็นค่าเดี่ยว หรือ อะตอมมิกแวลลู (atomic value)

2.4.2.1 ต้องไม่มีทัพเพิลที่ซ้ำกันภายในรีเลชัน

เนื่องจากบอดี้ของรีเลชันซึ่งประกอบขึ้นจากทัพเพิลหลายทัพเพิล มีลักษณะเป็นเซตทางคณิตศาสตร์ในรูปผลคูณคาร์ทีเซียน การซ้ำกันของทัพเพิลจึงเหมือนกับการซ้ำกันของสมาชิกภายในเซตซึ่งผิดนิยามทางคณิตศาสตร์ ดังนั้นจึงกำหนดคุณสมบัติข้อนี้เพื่อให้โครงสร้างข้อมูลแบบสัมพันธ์เป็นไปตามกฎทางคณิตศาสตร์ และก็เนื่องจากความเป็นหนึ่ง (ไม่มีซ้ำ) ของแต่ละทัพเพิลในรีเลชันนี้เอง ทำให้เราสามารถกล่าวได้ว่า "ทุก ๆ รีเลชันจะต้องมีไพรมารี่คีย์เกิดขึ้นเสมอ" เพราะอย่างน้อยที่สุดกลุ่มแอตตริบิวท์ที่ค่าข้อมูลมีความเป็นหนึ่ง ตามนิยามในหัวข้อ " ตัวอย่างตารางความสัมพันธ์" นั้น ก็คือเซตของแอตตริบิวท์ทั้งหมดในรีเลชันนั่นเอง

2.4.2.2 ลำดับของทัพเพิลไม่มีความสำคัญในการเก็บ

จากนิยามทางคณิตศาสตร์ที่ถือว่าสมาชิกภายในเซต จะอยู่กระจัดกระจายไม่มีลำดับ บอดี้ของรีเลชันซึ่งเป็นเซตที่มีสมาชิกคือ ทัพเพิล จึงไม่ให้ความสำคัญกับลำดับของทัพเพิล กล่าวคือเมื่อมีการลบหรือเพิ่มทัพเพิล ก็ไม่มีผลต่อความถูกต้องของข้อมูลแต่อย่างใด แต่ในทางปฏิบัติเวลาในการค้นหาข้อมูลก็ยังขึ้นอยู่กับตำแหน่งของทัพเพิลเป็นสำคัญ

2.4.2.3 ลำดับของแอตตริบิวท์ไม่มีความสำคัญในการเก็บ

ในนิยามเช่นเดียวกับคุณสมบัติในข้อที่สอง ลำดับของแอตตริบิวท์จึงไม่มีความสำคัญต่อการเก็บข้อมูลเช่นเดียวกัน สังเกตจากตัวอย่างรูปที่ 2.1 ข้อมูล " นายสมิทธ์ซึ่งมีเลขประจำตัว S1 มีสถานะที่ 20 และมีที่อยู่ที่ลอนดอน " สามารถบันทึกในรีเลชันว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

007728

S#	SNAME	STATUS	CITY
S1	Smith	20	London

หรือ

S#	CITY	SNAME	STATUS
S1	London	Smith	20

ก็ได้ข้อมูลความจริงที่ไม่แตกต่างกัน

2.4.2.4 ค่าข้อมูลของแต่ละแอตตริบิวท์ต้องเป็นค่าเดี่ยว หรือ อะตอมมิกแวลลู (atomic value)

ข้อกำหนดสุดท้ายหมายถึง ค่าข้อมูลของแอตตริบิวท์ใด ๆ ต้องเป็นค่าเดี่ยว หรือ อะตอมมิกแวลลู หมายความว่าเมื่อมีการบ่งชี้อรีเลชัน ชื่อแอตตริบิวท์ และทัพเพิ้ลที่ต้องการแล้ว จะต้องได้ค่าข้อมูลแอตตริบิวท์ออกมาเพียงหนึ่งค่าเท่านั้น ข้อกำหนดข้อนี้สามารถกล่าวได้อีกอย่างหนึ่งว่า " รีเลชันใด ๆ จะต้องไม่มีกลุ่มซ้ำ (repeating group) อยู่ภายใน " ตัวอย่างที่แสดงตารางความสัมพันธ์ที่มีกลุ่มซ้ำ และรีเลชันที่มีค่าข้อมูลเป็นอะตอมมิกแวลลู เป็นดังนี้

BEFORE

S#	PQ	
	P#	QTY
S1	P1	300
	P2	200
	P3	400
	P4	200
	P5	100
	P6	100
S2	P1	300
	P2	400
S3	P2	200
S4	P2	200
	P4	300
	P5	400

AFTER

S#	P#	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

รูปที่ 2.3 แสดงตัวอย่างตารางที่มีกลุ่มซ้ำและไม่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปตาราง BEFORE เป็นตารางที่มีกลุ่มซ้ำ เรียกตารางเช่นนี้ว่า ตารางที่ไม่นอร์มาไลซ์ (unnormalized) และตาราง AFTER ว่า ตารางนอร์มาไลซ์ (normalized) ตารางนอร์มาไลซ์นี้เรียกได้อีกอย่างว่า ตารางนอร์มอลฟอร์มที่ 1 (First normal form)

2.5 ความถูกต้องของข้อมูล (Data integrity)

ความถูกต้องของข้อมูลในรีเลชัน ขึ้นอยู่กับพฤติกรรมของข้อมูลที่มีต่อกฎความถูกต้อง (Integrity rules) ซึ่งมีอยู่ 2 กฎอันได้แก่

1. กฎความถูกต้องของเอนติตี้ (entity integrity)
2. กฎความถูกต้องในการอ้างถึง (referencial integrity)

ก่อนที่จะกล่าวถึงรายละเอียดของกฎทั้งสอง จะได้กล่าวถึงความหมายของคำเฉพาะในการอ้างถึงเสียก่อน ดังต่อไปนี้

2.5.1 ไพรมารี่คีย์ (Primary key)

ไพรมารี่คีย์คือกรณีพิเศษของแคนดิเดตคีย์ (candidate key) ความหมายของแคนดิเดตคีย์อธิบายได้ตามนิยามสองประการ เมื่อกำหนดให้ R เป็นรีเลชันใด ๆ A1,A2,...,An เป็นแอตตริบิวท์ของรีเลชันนั้น เซตของแอตตริบิวท์ K เท่ากับ (Ai,Aj,...,Ak) จะเป็นแคนดิเดตคีย์ของ R ได้ก็ต่อเมื่อ

1. นิยามความเป็นหนึ่งเดียว (Uniqueness)

   ในช่วงเวลาขณะใดขณะหนึ่ง จะไม่ปรากฏว่ามีทัพเพิลสองทัพเพิลในรีเลชัน R ที่มีค่าแอตตริบิวท์เท่ากันในแอตตริบิวท์ที่ Ai เท่ากันในแอตตริบิวท์ที่ Aj ... และเท่ากันในแอตตริบิวท์ที่ Ak

2. นิยามขนาดที่เล็กที่สุด (Minimally)

   จะไม่สามารถตัดแอตตริบิวท์ใด ๆ ออกไปจากเซต K ได้ โดยไม่ขัดแย้งกับนิยามความเป็นหนึ่งเดียว

ไพรมารี่คีย์ของรีเลชันใด ๆ ก็คือแคนดิเดตคีย์ตัวหนึ่งของรีเลชันนั้นที่ได้จากการกำหนดของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ออกแบบฐานข้อมูล เนื่องจากเป็นไปได้ว่ารีเลชันหนึ่งจะมีมากกว่าหนึ่งแคนดิเดตคีย์ จากรูปที่ 2.1 จะเห็นได้ว่าแอตตริบิวท์ SNAME ก็เป็นแคนดิเดตคีย์ เช่นกันแต่ในกรณีนี้ได้กำหนดให้เอาแอตตริบิวท์ S# เป็นไพรมารีคีย์ เรียก SNAME ว่าเป็น **อัลเทอร์เนตคีย์** (alternate key)

จากรีเลชัน 3 รีเลชันดังต่อไปนี้

S

S#	SNAME
S1	Smith
S2	John

P

P#	PNAME
P1	Nut
P2	Screw

SP

S#	P#	QTY
S1	P1	10
S1	P2	20
S2	P1	10

ไพรมารีคีย์ในรีเลชัน S คือ S# ไพรมารีคีย์ในรีเลชัน P คือ P# และไพรมารีคีย์ในรีเลชัน SP คือ S# และ P# ในลักษณะเช่นนี้เราเรียกแอตตริบิวท์ S# และ P# ในรีเลชัน SP ว่า **ฟอร์เรนคีย์** (foreign key) ฟอร์เรนคีย์ หมายถึงแอตตริบิวท์ในรีเลชันหนึ่ง ที่มีสมาชิกเทียบเท่ากับแอตตริบิวท์เดียวกันในอีกรีเลชันหนึ่ง เรียกความสัมพันธ์ระหว่างแอตตริบิวท์ที่เป็นฟอร์เรนคีย์ กับไพรมารีคีย์ของรีเลชันอื่นว่า การอ้างถึง หรือเรฟเฟอร์เรนซ์ (reference)

2.5.2 กฎความถูกต้องของเอนติตี้ (entity integrity)

<u>กล่าวว่า</u> เอนติตี้ข้อมูลซึ่งหมายถึงค่าข้อมูลในแอตตริบิวท์ ซึ่งเป็นแอตตริบิวท์ไพรมารีคีย์ของรีเลชันใด ๆ ต้องไม่เป็น null (นัล หมายถึง ไม่บันทึกค่า)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 กฎความถูกต้องในการอ้างถึง (referencial integrity)

กล่าวว่า หาก FK คือแอตตริบิวท์ฟอร์เรนคีย์ของรีเลชัน R2 และ PK คือแอตตริบิวท์ไพรมารีคีย์ของรีเลชัน R1 แล้ว ค่าข้อมูลแต่ละค่าในแอตตริบิวท์ FK ในรีเลชัน R2 จะต้อง ก) มีค่าเท่ากับค่าข้อมูลที่อยู่ในแอตตริบิวท์ PK ในรีเลชัน R1 ในทัพเพิ้ลใดทัพเพิ้ลหนึ่ง หรือ ข) เป็น null ซึ่งหมายถึง ทุกค่าใน FK เป็น null ทั้งหมด

2.6 การจัดการข้อมูล (Data manipulation)

ข้อมูลภายในฐานข้อมูลจะถูกจัดการด้วยภาษาจัดการข้อมูลภายในฐานข้อมูล ภาษาจัดการฐานข้อมูลนี้จะต้องเป็นไปตามนิยามของ รีเลชันแนล คอมพลีท (relational complete) กล่าวคือต้องสนับสนุนกลุ่มปฏิบัติการพื้นฐานที่นิยามในภาษา รีเลชันสัมพันธ์ (Relational Complete) หรือ แคลคูลัสสัมพันธ์ (Relational Calculus) ซึ่งประกอบไปด้วยปฏิบัติการ 8 อย่างโดยแบ่งออกเป็น 2 กลุ่มได้ดังนี้

1. กลุ่มปฏิบัติการพื้นฐาน (Traditional Set Operations) ได้แก่ ผลคูณคาร์ทีเซียน หรือ โปรดัค (CARTESIAN PRODUCT), ยูเนียน (UNION), อินเตอร์เซ็คชัน (INTERSECTION) และ ดิฟเฟอเรนซ์ (DIFFERENCE)

2. กลุ่มปฏิบัติการพิเศษ (Special Relational Operations) ได้แก่ ซีเล็ค (SELECT), โปรเจ็ค (PROJECT), จอยน์ (JOIN) และ ดิไวด์ (DIVIDE)

ลักษณะการทำงานของตัวปฏิบัติการต่าง ๆ

SELECT คือ การสร้างความสัมพันธ์จากการเลือกเอาข้อมูลในทัพเพิล ที่มีคุณสมบัติตามที่ระบุออกมาจากความสัมพันธ์ที่มีอยู่

PROJECT คือ การสร้างความสัมพันธ์โดยการเลือกเอาข้อมูลจากแอททริบิวท์ บางตัวตามที่ได้กำหนด ออกมาจากความสัมพันธ์ที่มีอยู่

PRODUCT คือ การสร้างความสัมพันธ์ขึ้นใหม่จากความสัมพันธ์ 2 ความสัมพันธ์ ซึ่งความสัมพันธ์ที่สร้างขึ้น จะประกอบด้วยข้อมูลที่เป็นไปได้ทั้งหมด ที่เกิดจากการนำข้อมูลในทัพเพิล ของความสัมพันธ์ทั้งสองมาจับคู่กัน

UNION คือ การสร้างความสัมพันธ์ขึ้นใหม่จากความสัมพันธ์ 2 ความสัมพันธ์ โดยความสัม-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พันธ์ที่สร้างขึ้นนี้ จะมีข้อมูลที่ประกอบไปด้วยข้อมูลของทั้งสองความสัมพันธ์

INTERSECT คือ การสร้างความสัมพันธ์จากความสัมพันธ์ 2 ความสัมพันธ์ที่กำหนดให้ โดยความสัมพันธ์ที่ได้ จะประกอบด้วยข้อมูลที่อยู่ในทัพเพิล ที่เหมือนกันของทั้งสองความสัมพันธ์ที่กำหนด

DIFFERENCE คือ การสร้างความสัมพันธ์ที่เกิดจากความสัมพันธ์ 2 ความสัมพันธ์ที่กำหนดให้ ซึ่งความสัมพันธ์ที่เกิดขึ้นใหม่จะประกอบด้วย ข้อมูลในทัพเพิลทั้งหมดที่มีอยู่ในความสัมพันธ์อันแรกที่กำหนดให้ แต่ไม่มีอยู่ในความสัมพันธ์อันที่สอง

JOIN คือ การสร้างความสัมพันธ์ใหม่จากความสัมพันธ์ 2 ความสัมพันธ์ โดยความสัมพันธ์ที่สร้างขึ้นใหม่ จะประกอบด้วย ทัพเพิลที่เกิดจากการจับคู่ระหว่างทัพเพิลของทั้งสองความสัมพันธ์ ซึ่งการจับคู่ของทัพเพิลนี้ จะต้องอยู่ภายใต้เงื่อนไขอย่างหนึ่ง

DIVIDE คือ การสร้างความสัมพันธ์จากความสัมพันธ์ 2 แบบ คือ ความสัมพันธ์แบบไบนารี่ (Binary) และความสัมพันธ์แบบยูนารี่ (Unary) ซึ่งความสัมพันธ์ที่สร้างขึ้นจะประกอบด้วยข้อมูลทั้งหมดของแอททริบิวท์แรกของความสัมพันธ์แบบไบนารี่ ที่มีข้อมูลในแอททริบิวท์ที่สองเหมือนกับข้อมูลทั้งหมดในความสัมพันธ์แบบยูนารี่

External (PL/I)

```
DCL 1 EMPP,
    2 EMP# CHAR(6),
    2 SAL FIXED BIN(31);
```

External (COBOL)

```
01 EMPC.
    02 EMPNO PIC X(6).
    02 DEPTNO PIC X(4).
```

Conceptual

```
EMPLOYEE
    EMPLOYEE_NUMBER     CHARACTER (6)
    DEPARTMENT_NUMBER   CHARACTER (4)
    SALARY              NUMERIC   (5)
```

Internal

```
STORED_EMP      LENGTH=18
   PREFIX       TYPE=BYTE(6),OFFSET=0
   EMP#         TYPE=BYTE(6),OFFSET=6,INDEX=EMPX
   DEPT#        TYPE=BYTE(4),OFFSET=12
   PAY          TYPE=FULLWORD,OFFSET=16
```

รูปที่ 2.5 แสดงตัวอย่างระดับข้อมูลในระบบใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 การออกแบบฐานข้อมูล โดยใช้วิธีนอร์มาไลซ์เซชัน (Normalization)

นอร์มาไลเซชันนั้นคือวิธีการแยกตารางฐานข้อมูลจากตารางรวม ออกเป็นตารางที่เรียกว่า Normal form (NF) ต่างๆ โดยเริ่มที่ตารางที่มีลักษณะของค่าเดี่ยว (Atomic value) หรือ ตารางนอร์มาไลซ์ เรียกว่านอร์มอลฟอร์มที่ 1 (1NF) ใช้นิยามในการค้นหาแอตตริบิวท์ที่เกิดความขึ้นต่อกัน (dependent) กับแอตตริบิวท์อื่น แล้วทำการแยกแอตตริบิวท์ดังกล่าวนั้นออกเป็นตารางในนอร์มอลฟอร์มใหม่ ไม่ว่าจะเป็น การขึ้นต่อกันของฟังก์ชัน (Functional dependency) การขึ้นต่อกันค่าข้อมูลหลายระดับ (Multivalued dependency) หรือการขึ้นต่อกันในลักษณะร่วม (Join dependency) นอร์มอลฟอร์มต่าง ๆ ได้แก่ 2NF , 3NF , BCNF , 4NF , 5NF และ ONF การแปลงจากนอร์มอลฟอร์มหนึ่งไปเป็นอีกนอร์มอลฟอร์มหนึ่ง เป็นกระบวนการที่ค่อนข้างยุ่งยาก ใช้เวลานานในกรณีที่เป็นตารางขนาดใหญ่ และยากต่อการนำไปพัฒนาเป็นโปรแกรม การนอร์มาไลซ์ฐานข้อมูล 1 แอพพลิเคชัน อาจต้องใช้เวลานานนับเป็นปี ทั้งยังไม่อาจรับรองคุณภาพของฐานข้อมูลว่าดีพอ ซึ่งปกติแล้วจะทำการนอร์มาไลซ์ถึงเพียงแค่ BCNF เท่านั้น เพราะการทำมากกว่านี้ทำได้ยากมาก จึงสามารถเกิดความซ้ำซ้อนของข้อมูลได้อีก

2.8 การออกแบบฐานข้อมูล โดยใช้เอนติตี้รีเลชันชิพ (Entity-Relationship Model E/R/Model)

เป็นการใช้แผนภาพในการแทนความสัมพันธ์ของข้อมูล โดยที่ข้อมูลแบ่งออกได้เป็น 2 ชนิดที่สำคัญ คือ

- ชนิดเอนติตี้ เป็นตัวแทนของนาม หรือกลุ่มของนาม แทนได้ด้วยรูปสี่เหลี่ยม
- ชนิดรีเลชันชิพ เป็นการแสดงถึงกริยาการกระทำ หรือกลุ่มของกริยา แทนได้ด้วยสี่เหลี่ยมข้าวหลามตัด

เช่น ในประโยค " นาย ก. ได้รับเงินเดือน 10,000 บาท "

ประกอบด้วย เอนติตี้ 2 เอนติตี้ คือ "นาย ก." และ "เงินเดือน"

และ 1 รีเลชันชิพ คือ "ได้รับ" เป็นต้น

เอนติตี้จะแทนสิ่งที่อยู่ในโลกทั้งหมด เหตการณ์ หรือ บุคคล และ รีเลชันชิพ จะแสดงให้เห็นถึงความสัมพันธ์ของเอนติตี้ ตั้งแต่ 2 เอนติตี้ขึ้นไป คุณสมบัติของเอนติตี้และรีเลชันชิพ เรียกว่า แอตตริบิวท์ส (attributes)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 แบบจำลองข้อมูลไนแอม (The NIAM Conceptual Schema)

ไนแอม (NIAM : Nijssen's Information Analysis Methodology) เป็นวิธีการในวิธีข้อมูลที่ใช้แบบจำลองข้อมูลที่มีลักษณะต่างๆและมีความหมาย เพื่อจุดประสงค์ในการตรวจสอบความถูกต้องของระบบสารสนเทศ คิดค้นขึ้นโดย Prof.G.M.Nijssen ในขณะที่ทำงานอยู่ที่ Control Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
แบบจำลองข้อมูลไนแอม

ไนแอม (NIAM : Nijssen's Information Analysis Methodology) เป็นวิธีการในวิธีข้อมูลที่ใช้แบบจำลองข้อมูลที่มีลักษณะต่างๆและมีความหมาย เพื่อจุดประสงค์ในการตรวจสอบความถูกต้องของระบบสารสนเทศ คิดค้นขึ้นโดย Prof.G.M.Nijssen ในขณะที่ทำงานอยู่ที่ Control Data ไนแอมได้รับการพัฒนาเรื่อยมา โดย Nijssen , Falkenberg และบุคคลอื่นๆ ไนแอมมีลักษณะเป็นแฟคโอเรียนท์โมเดลลิ่ง (fact oriented modeling) ถึงแม้ว่าจะมีความคล้ายคลึงกับ เอนติตี้รีเลชันชิฟโมเดล (Entity-relation model) ที่คิดค้นโดย Chen ในปี 1976 แต่มีความเชื่อว่า ไนแอมเข้าใจง่ายและเป็นธรรมชาติกว่า

3.1 ส่วนประกอบต่างๆของไนแอม

1.เอนติตี้ไทป์ (Entity type) คือเซตของวัตถุที่สนใจ (object) หรือ สิ่งของ (thing) พื้นฐาน ที่เราต้องการกล่าวถึง เอนติตี้อาจจะเป็นวัตถุที่มีตัวตนเช่น คนชื่อ นาย ก. หรือ อาจจะเป็นวัตถุนามธรรม เช่น วิชา CS112 เป็นต้น ตัวอย่างของเอนติตี้ไทป์ได้แก่ เอนติตี้ไทป์ที่เป็นชนิดคน ก็จะเป็นเซตของคนทั้งหมดที่เราต้องการกล่าวถึง

รูปที่ 3.1 แสดงชนิดเอนติตี้

2.เลเบ้ลไทป์ (Label type) คือเซตของสิ่งที่อ้างถึงวัตถุที่เราเฉพาะเจาะจงได้ เช่น ชื่อคน ได้แก่ นางมาร์กาเรต แทตเชอร์ เป็นชื่ออ้างถึงของเอนติตี้นายกรัฐมนตรี , รหัสนักศึกษา เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.2 แสดงรูปชนิดของเลเบิ้ล

3.แฟกท์ไทป์ (Fact type) เป็นสิ่งที่อ้างถึงความสัมพันธ์ระหว่างสิ่งที่เราสนใจ นั่นคือ เป็นสิ่งที่ใช้สำหรับอ้างถึงความสัมพันธ์กันระหว่างเอนติตี้ ซึ่งจะมีความสัมพันธ์กันแบบใดก็ได้

รูปที่ 3.3 แสดงชนิดของแฟกท์

4.เรฟเฟอเรนจ์ไทป์ (Reference type) เป็นสิ่งที่ใช้อ้างความสัมพันธ์ระหว่างเอนติตี้ ไทป์กับเลเบิ้ลไทป์

รูปที่ 3.4 แสดงชนิดของเรฟเฟอเรนจ์

5.คอนสเตรนท์ (Constraint) หรือที่เรียกกันว่า ข้อจำกัดซึ่งเป็นสิ่งที่ใช้ในการกำหนดให้มีการยอมรับข้อมูลของประชากรที่ถูกต้องตามข้อกำหนดให้สามารถเก็บลงในฐานข้อมูล

5.1 คอนสเตรนท์แบบยูนีคเนสส์ (Uniqueness constraint)

สำหรับคอนสเตรนท์แบบยูนีคเนสส์ยังแบ่งออกเป็น 2 อย่างคือ

5.1.1 คอนสเตรนท์แบบยูนีคเนสส์ภายใน (Intra-Fact Category

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

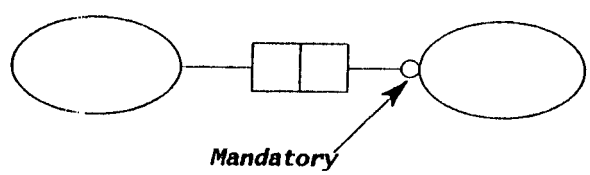
Uniqueness Constraint.) ใช้กำหนดจำนวนโรล (role) ที่น้อยที่สุดสำหรับแฟคส์หนึ่งๆที่จะทำให้ข้อมูลแต่ละแถวของแฟคส์นั้นไม่ซ้ำกัน

รูปที่ 3.5 แสดงคอนสเตรนท์แบบยูนีคเนสส์ภายใน

5.1.2 คอนสเตรนท์แบบยูนีคเนสส์ภายนอก (Extra-Fact Category Uniqueness Constraint.) ใช้อ้างถึงสิ่งที่เราสนใจมาประกอบกัน เช่น การบุคคลอ้างถึงโดยการรวมชื่อกับนามสกุลด้วยกัน

รูปที่ 3.6 แสดงคอนสเตรนท์แบบยูนีคเนสส์ภายนอก

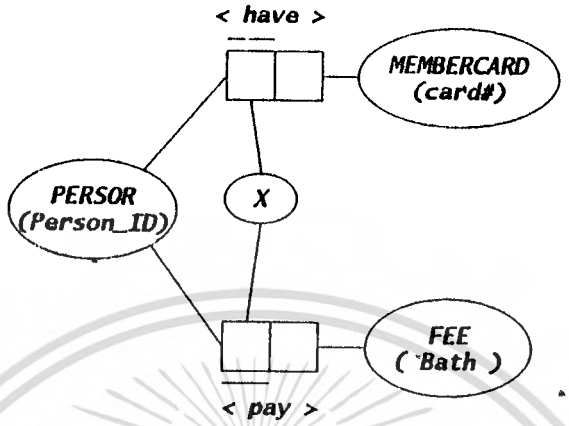
5.2 คอนสเตรนท์แบบแมนดาทอรี่ (Mandatory constraint) อธิบายได้ดังนี้ เมื่อโรลใดมีสัญลักษณ์แมนดาทอรี่ปรากฏอยู่ โดยถ้าเรารู้ค่าข้อมูลของเอนติตี้แล้ว ค่าของข้อมูลของสิ่งยื่นที่มาเกี่ยวข้องกับเอนติตี้ที่มีสัญลักษณ์แมนดาทอรี่นั้น ต้องมีค่าของข้อมูล ตัวอย่างเช่น เมื่อปรากฏชื่อของบุคคลหนึ่ง บุคคลนั้นต้องมีนามสกุลปรากฏอยู่ด้วย



รูปที่ 3.7 แสดงคอนสเตรนท์แบบแมนดาทอรี่

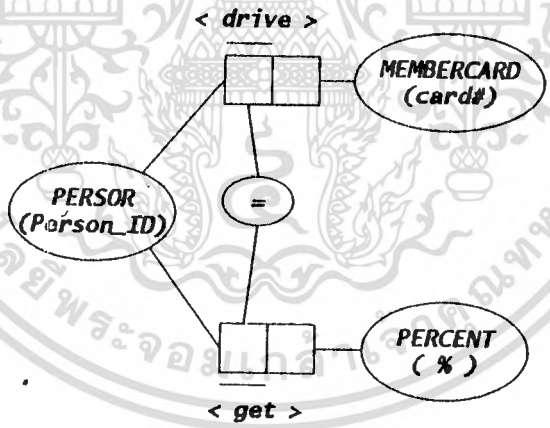
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 คอนสเตรนท์แบบเอ็กซ์คลูชัน (Exclusion constraint) เป็นข้อกำหนดว่า สมาชิกทุกตัวภายในโรลของเอนติตี้หนึ่งจะไม่เป็นสมาชิกภายในอีกโรลหนึ่งของเอนติตี้เดียวกัน



รูปที่ 3.8 แสดงคอนสเตรนท์แบบเอ็กซ์คลูชัน

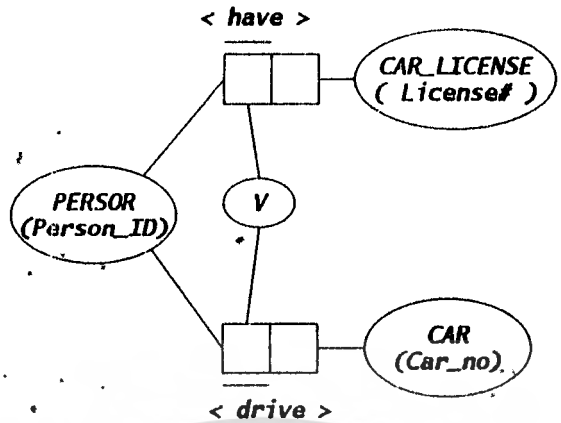
5.4 คอนสเตรนท์แบบอิควอลลิตี้ (Equality constraint) เป็นข้อกำหนดว่า สมาชิกทุกตัวภายในโรลของเอนติตี้หนึ่งต้องเป็นสมาชิกภายในอีกเอนติตี้ในอีกโรลหนึ่งของเอนติตี้เดียวกัน



รูปที่ 3.9 แสดงคอนสเตรนท์แบบอิควอลลิตี้

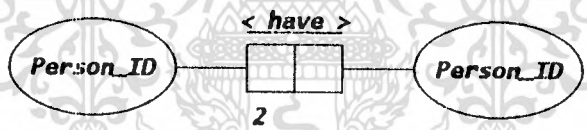
5.5 คอนสเตรนท์แบบสับเซต (Subset constraint) เป็นข้อกำหนดว่า เซตของสมาชิกทุกตัวภายในโรลของเอนติตี้หนึ่งต้องเป็นสับเซตกับเซตของสมาชิกภายในอีกโรลหนึ่งของเอนติตี้เดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



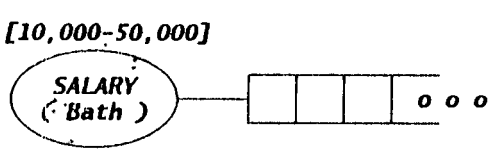
รูปที่ 3.10 แสดงคอนสเตรนท์แบบสับเซต

5.6 คอนสเตรนท์แบบออคเคอเรนซ์ฟรีเควนซี่ (Occurence frequency constraint) เป็นข้อกำหนดความถี่หรือจำนวนครั้งที่เอนติตี้ใดๆที่ต่ออยู่กับโรลที่กำกับด้วยคอนสเตรนท์แบบนี้ จะถูกเก็บข้อมูล



รูปที่ 3.11 แสดงคอนสเตรนท์แบบออคเคอเรนซ์ฟรีเควนซี่

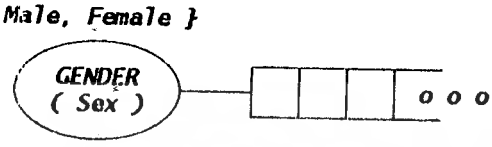
5.7 คอนสเตรนท์แบบเรนจ์ (Range constraint) เป็นคอนสเตรนท์แบบเอนติตี้ไทป์ประเภทหนึ่ง เป็นข้อกำหนดขอบเขตของข้อมูลในแต่ละเอนติตี้ ในแบบช่วงของจำนวนข้อมูล



รูปที่ 3.12 แสดงคอนสเตรนท์แบบเรนจ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.8 <u>คอนสเตรนท์แบบเมมเบอร์ชิพ</u> (Membership constraint) เป็นคอนสเตรนท์แบบเอนติตี้ไทป์ประเภทหนึ่ง เป็นการกำหนดลักษณะของข้อมูลหรือประชากรในเอนติตี้



รูปที่ 3.13 แสดงคอนสเตรนท์แบบเมมเบอร์ชิพ

การสร้างแบบจำลองข้อมูลนี้จะมีพื้นฐานจากการกำหนดตัวอย่างข้อมูลขึ้นมา และนำมาผ่านกระบวนการ (Procedure) ที่ได้กำหนดไว้เป็นขั้นตอนที่แน่นอนแล้ว เราก็จะได้แผนภาพที่มีความหมายในการแทนข้อมูลที่ยกตัวอย่างขึ้นมาได้ และเนื่องจากแบบจำลองข้อมูลนี้คิดขึ้นครั้งแรกโดย Nijssen ดังนั้นจึงเรียกแบบจำลองข้อมูลนี้ว่า NIAM (Nijssen's Information Analysis Methodology)

หลังจากนั้นไนแอมก็ได้นำไปพัฒนาต่อโดย Nijssen และ T.A. Halpin ที่มหาวิทยาลัย Queensland และเนื่องจากวิธีการที่พัฒนาขึ้นนี้จะเน้นที่แฟกท์ไทป์ (Fact Type) เป็นหลัก ดังนั้นอาจจะเรียกแบบจำลองข้อมูลไนแอมว่า fact-oriented modeling และแม้ว่าไนแอมมีความคล้ายคลึงกับโมเดลอีอาร์ (Entity-Relationship Model) และอาจจะมีความสัมพันธ์กันในบางแง่มุมก็ตาม แต่ในความเป็นจริงแล้วแบบจำลองข้อมูลไนแอมนั้นจะสามารถที่จะแสดงความหมายได้ดีกว่า ทำให้สามารถทำความเข้าใจได้ง่ายกว่าและเป็นธรรมชาติกว่า นอกจากนั้นแบบจำลองข้อมูลไนแอมยังสามารถที่จะแปลงรูปไปเป็น 5NF (Fifth Normal Form) ได้ทันที ในขณะที่โมเดลอีอาร์ ทำได้เพียง 3NF เท่านั้น ดังนั้นเราสามารถจะใช้ไนแอมเพื่อช่วยในการออกแบบฐานข้อมูลแบบรีเลชันแนล โดยใช้แทนวิธี "นอร์มาไลซ์เซชัน" ได้อีกด้วย

3.2 <u>ขั้นตอนการลอกแบบแบบจำลองข้อมูลไนแอม</u>

จากที่ได้กล่าวมาข้างต้นเราอาจจะมองว่าสามารถออกแบบได้ง่าย อย่างไรก็ตามหลักการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของการออกแบบ แบบจำลองข้อมูล หากจะทำให้มีความถูกต้องก็จะเป็นงานที่ซับซ้อนไม่น้อย ดังนั้น เราจะแบ่งออกเป็นงาน 9 ขั้นตอนด้วยกัน ดังนี้

ขั้นตอนที่ 1 จากตัวอย่างสู่ "ความจริงพื้นฐาน" (Elementary Facts)

งานในขั้นตอนแรกของการออกแบบ ก็คือเราจะต้องหาตัวอย่างข้อมูลที่ใช้งานอยู่ในระบบ ตัวอย่างข้อมูลที่สำคัญก็คือ รายงาน (Output Report) และแบบฟอร์มที่ใช้ในการรับข้อมูลเข้าสู่ระบบ (Input Form) ซึ่งอาจจะอยู่ในรูปของตารางภาพ หรือข้อความใด ๆ ก็ตาม

Tutorial group	*Time*	*Room*	*Student#*	*Student name*
A	Mon 3 p.m.	CS-718	302156	Bloggs FB
			180064	Fletcher JB
			278155	Jackson M
			334067	Jones EP
			200140	Kawamoto T
..	..	..	..	..
B1	Tue 2 p.m.	E-B18	266010	Anderson AB
			348112	Bloggs FB
..	..	..	..	..

รูปที่ 3.14 แสดงตัวอย่างข้อมูลของตารางเรียน

จากนั้นเราก็จะนำข้อมูลที่หามาได้ ซึ่งในที่นี้ได้แก่ข้อมูลในตารางดังกล่าวมาเขียนให้อยู่ในรูปของ "ความจริงพื้นฐาน" (Elementary Facts) เช่นจากข้อมูลในบรรทัดแรกเราสามารถเขียนได้เป็น

```
Tute group A meets at Time Mon 3 p.m.
Tute group A is held in Room CS-718.
Student 302156 belongs to Tute group A.
Student 302156 has Name 'Bloggs FB'.
```

จะเห็นได้ว่าการเขียนในรูปแบบนี้จะสามารถแสดงความสัมพันธ์ของข้อมูลได้มากกว่าแบบตาราง เช่น จะเห็นได้ว่านักศึกษาที่มีชื่อและเลขประจำตัวที่อยู่ในแถวเดียวกันหมายถึงคนเดียวกัน อย่างไรก็ตาม แม้ว่าข้อมูลจะเป็นแบบอื่นเช่นอาจจะเป็นแผนภูมิต่างๆ แต่งานในขั้นแรกนี้ก็จะต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำมาเขียนเป็น "ความจริงพื้นฐาน" ทั้งหมด

คำว่า "ความจริงพื้นฐาน" นั้นอาจจะถูกมองว่าเป็น "วัตถุ กระทำ สิ่งต่างๆ" (Particular Object Play Particular Role) เช่นประโยคที่ว่า Ann smoke. นั้นเราสามารถมองได้ว่า Ann (Object) กระทำ smoke (Role) นอกจากนั้น "ความจริงพื้นฐาน" ยังจะต้องเป็นข้อมูลพื้นฐานที่ไม่สามารถแบ่งแยกได้

แต่ก่อนที่เราจะกล่าวถึง "ความจริงพื้นฐาน" ในด้านถัดต่อไป เราจะกล่าวถึงนิยามของคำต่างๆ ที่เราจะใช้งานต่อไป โดยเริ่มจากคำว่าเอนติตี้ (Entity) เอนติตี้เป็นหน่วยพื้นฐานที่สุดของระบบข้อมูลที่เราให้ความสนใจ เช่น ในระบบข้อมูลของมหาวิทยาลัย นักศึกษา ภาค-วิชา อาจารย์ ล้วนจัดเป็นเอนติตี้ และเรียกเซ็ทของเอนติตี้ว่าเอนติตี้ไทป์ (Entity Type) ซึ่งมีสมาชิกเป็นตัวอย่างเอนติตี้ (Entity Instance) แต่ละตัวอย่างเอนติตี้จำเป็นต้องมีชื่อเรียกซึ่งอาจเป็นชื่อ, รหัส อย่างใดอย่างหนึ่ง เช่น นักศึกษาจะเรียกโดยใช้รหัสนักศึกษาเป็นชื่อเรียกการกำหนดว่าแต่ละชนิดเอนติตี้ควรใช้อะไรเป็นชื่อเรียก ทำได้โดยระบุเลเบ้ลไทป์ (LabelType) สำหรับเอนติตี้นั้น เช่นชนิดเอนติตี้ Person อาจจะใช้เลเบ้ลไทป์ Surname เป็นชื่อเรียก

ดังนั้นสำหรับข้อมูลที่ว่า "นักศึกษาเลขประจำตัว 311308 ลงเรียนในวิชา CS112"หากจะเขียนให้อยู่ในรูปของ "ความจริงพื้นฐาน" ให้ครบสูตร เราสามารถเขียนให้อยู่ในรูปของประโยคภาษาธรรมชาติได้ดังนี้

The STUDENT with student# '311308'
studies
the SUBJECT with code 'CS112'.

โดยที่ทั้ง STUDENT และ SUBJECT เป็น เอนติตี้ไทป์
student# และ code เป็น เลเบ้ลไทป์
'311308' และ 'CS112' เป็น Instant
และ studies เป็น Role

และจากประโยคข้างต้นจะสังเกตุว่าเราจะเขียนเอนติตี้ไทป์คู่กับเลเบ้ลไทป์เสมอ อย่างไรก็ตาม เราอาจเขียนความสัมพันธ์ข้างต้นเสียใหม่ ดังนี้

The SUBJECT with code 'CS112'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

was studied by

the STUDENT with student# '311308'

ซึ่งจะเห็นได้ว่าการเขียนในทั้งสองรูปแบบสามารถแทนความจริงเดียวกันได้ โดยในประโยคหลังนี้ was studied by เป็นโรล โดยที่เราจะเรียกการเขียนประโยคที่ต่างกันนี้ว่ามี "Surface Structure" ที่ต่างกัน หากแต่มี "Deep Strcuture" อันเดียวกัน ดังนั้นหากจะเขียนความสัมพันธ์ดังกล่าวในรูปแบบใดรูปแบบหนึ่ง ก็คงจะเป็นการไม่ถูกต้องนัก และเพื่อให้สื่อถึงความสัมพันธ์เดียวกันทุกครั้ง เราจะเขียนความจริงนี้ใหม่ในรูปแบบต่อไปนี้

Study ([The STUDENT with student# '311308',studies],

[The SUBJECT with code 'CS112',was studied by])

โดย Study เป็นแฟกท์ไทป์

ที่เราได้กล่าวมาทั้งหมดนั้น เป็นการกล่าวถึงการเปลี่ยนจากตัวข้อมูลที่อยู่ในรูปแบบต่างๆ ให้มาอยู่ในรูปแบบของ "ความจริงพื้นฐาน" ซึ่งเป็นรูปแบบที่สามารถแสดงความสัมพันธ์ของข้อมูลแต่ละตัวได้ดีกว่า

ขั้นตอนที่ 2 แผนภูมิแรกของแบบจำลองข้อมูลระดับแนวคิด

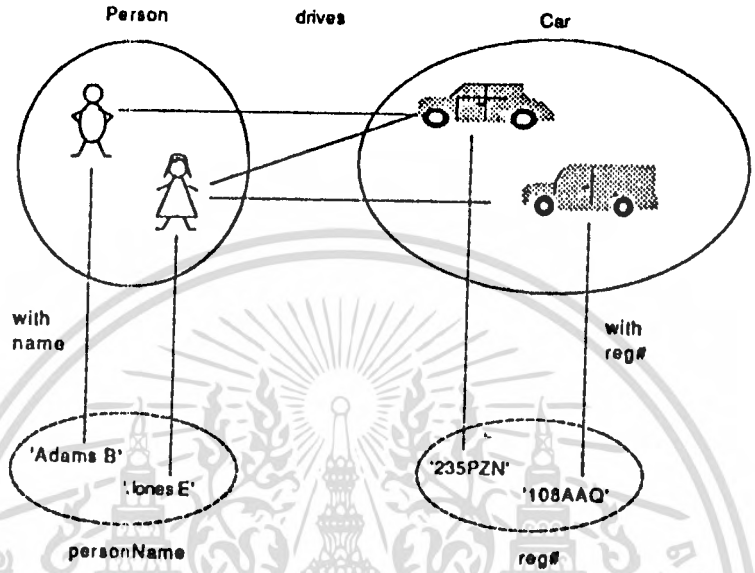
งานหลักในขั้นตอนนี้คือการนำ "ความจริงพื้นฐาน" ที่เราได้มาจากขั้นตอนที่ 1 มาวาดเป็นแผนภาพ ทั้งนี้เพราะได้กล่าวมาแล้วว่าไนแอมเป็นแบบจำลองแทนความหมาย (Semantic Modelling) ดังนั้นการที่อธิบายความสัมพันธ์ของข้อมูลด้วยแผนภาพจะได้ความชัดเจนมากกว่า และเพื่อที่จะสามารถอธิบายได้ชัดเจนยิ่งขึ้น เราจะจะสมมติข้อมูลขึ้นมาชุดหนึ่ง ซึ่งเป็นข้อมูลว่าใครขับรถเลขทะเบียนอะไรดังตารางที่ 3.1

DRIVES

Person	Car
Adams B	235PZN
Jones E	235PZN
Jones E	108AAQ

ตารางที่ 3.1 แสดงตัวอย่างข้อมูลความสัมพันธ์ของรถและคนขับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 แสดงความสัมพันธ์ของข้อมูลในตารางด้วยรูปภาพ

จากข้อมูลในตารางเราสามารถนำมาเขียนเป็น "ความจริงพื้นฐาน" ได้ดังนี้

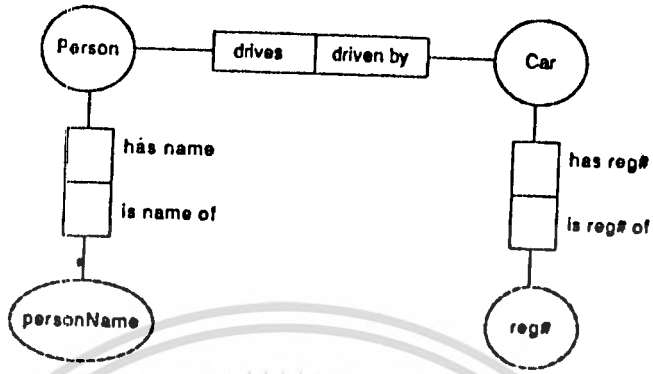
The Person with Name 'Adams B' drives the Car with reg# '235PZN'.

The Person with Name 'Jones E' drives the Car with reg# '235PZN'.

The Person with Name 'Jones E' drives the Car with reg# '108AAQ'.

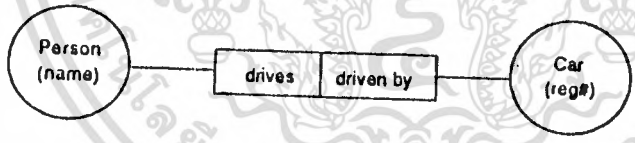
ซึ่งแสดงให้ชัดเจนด้วยรูปดังรูปที่ 3.15 และจาก "ความจริงพื้นฐาน" เราก็นำมาเขียนเป็นแผนภาพที่แสดงแนวคิด (Conceptual Schema Diagram) ได้ดังรูปที่ 3.16 และจากแผนภาพดังกล่าวจะเห็นได้ว่าเราได้แทนเอนติตี้ไทป์โดยใช้วงรีที่เส้นทึบ และแทนเลเบิ้ลไทป์ด้วยวงรีที่เส้นประ และเขียนโรลด้วยรูปสี่เหลี่ยม โดยมีเส้นลากจากเอนติตี้ไปยังโรลเพื่อแสดงความสัมพันธ์ และเราจะเรียกความสัมพันธ์ระหว่างเอนติตี้ไทป์ทั้งสองโดยผ่านโรลทั้งสองว่า **แฟกท์ไทป์ (Fact Type)** และความสัมพันธ์ระหว่างเอนติตี้ไทป์และเลเบิ้ลไทป์ว่า **เรฟเฟอร์เรนซ์ไทป์ (Reference Type)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 แสดงแผนภาพจำลองข้อมูลระดับแนวคิด

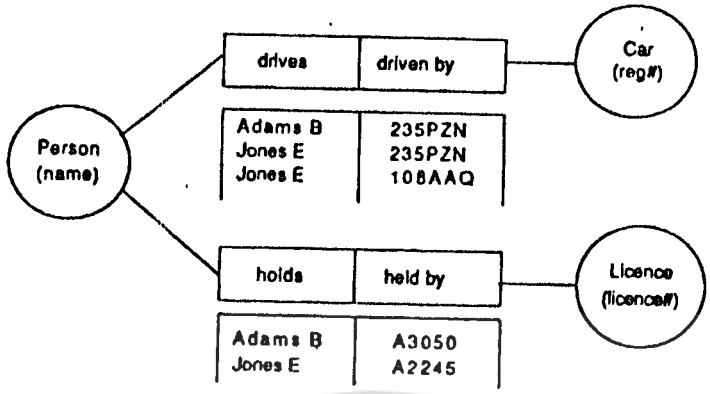
อย่างไรก็ตามจะสังเกตุได้ว่า สมาชิกแต่ละตัวในเอนติตี้ไทป์จะมีความสัมพันธ์กับสมาชิกในเรฟเฟอร์เรนซ์ไทป์กันแบบ 1:1 เช่นในเอนติตี้ไทป์แต่ละ Person จะมีเพียง 1 ชื่อและแต่ละชื่อก็จะหมายถึงคนหนึ่งคน ดังนั้นเราสามารถเขียนแผนภูมิในรูป 3.16 ได้ใหม่ดังในรูปที่ 3.17 โดยไม่เรฟเฟอร์เรนซ์ไทป์ไว้ในวงเล็บเพื่อที่จะเขียนได้ง่ายขึ้น โดยสามารถสื่อความหมายได้คงเดิม



รูปที่ 3.17 แสดงการเขียนแบบจำลองข้อมูลที่มีเรฟเฟอร์เรนซ์ไทป์เป็นแบบ 1:1

สำหรับวิธีการตรวจสอบความถูกต้องของแผนภูมิที่เราวาดนั้น สามารถทำได้โดยการใส่ตัวอย่างลงในแบบจำลองข้อมูล โดยจะเรียกแผนภูมิที่มีตัวอย่างนี้ว่า Schema-Base Diagram ดังรูปที่ 3.18 จะเห็นได้ว่าการเขียนความสัมพันธ์ในรูปแผนภูมิระดับแนวคิดนั้นสามารถมองเห็นความสัมพันธ์ได้ดีกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 แสดง Schema-Based Diagram

Student	*Subject*	*Rating*
Adams	CS100	4
Brown	CS100	4
Brown	CS112	5

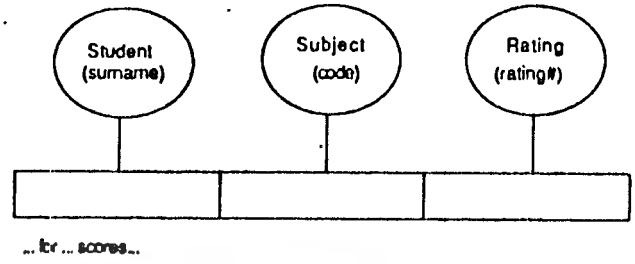
ตารางที่ 3.2 แสดงตัวอย่างข้อมูลของผลการเรียนนักศึกษา

ลองมาดูกันอีกตัวอย่างโดยสมมติข้อมูลดังในตารางที่ 3.2 ซึ่งสามารถเขียนให้อยู่ในรูปของ "ความจริงพื้นฐาน" ได้ดังนี้

Student (surname) 'Adams' for Subject (code) 'CS100'
scores Rating (rating#) 4.

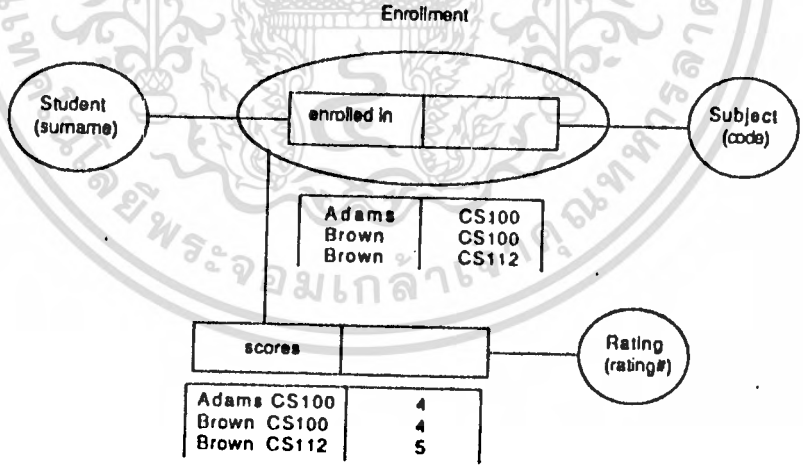
ซึ่งจาก "ความจริงพื้นฐาน" นี้สามารถนำมาเขียนเป็นแผนภูมิระดับแนวคิดได้ดังรูปที่ 3.6 ซึ่งเป็นแฟกท์ไทป์ที่มี 3 โรล และแฟกท์ไทป์ที่มีจำนวนโรลต่างกันนี้ ก็จะมีชื่อเรียกต่างกัน เช่น จะเรียกแผนภูมิที่ประกอบด้วย 2 โรล ว่าแฟกท์ไทป์แบบไบนารี่ (BinaryFact Type) และ 3 โรล ว่า แฟกท์ไทป์แบบเทอร์นารี่ (Ternary Fact Type) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 แสดงแฟกท์ไทป์ของผลการเรียน

อย่างไรก็ตาม การเขียนในลักษณะเช่นนี้นั้น อาจจะไม่เป็นการถูกต้องนักเพราะชนิดเอนติตี้ Rating ถือได้ว่ามิใช่มีความสัมพันธ์กับเพียงชื่อวิชาหรือตัวนักศึกษาอย่างใดอย่างหนึ่งเท่านั้น แต่มีความสัมพันธ์กับทั้งเอนติตี้ไทป์ Student และเอนติตี้ไทป์ Subject เช่นนักศึกษาที่มีชื่อAdams และลงเรียน CS100 เท่านั้น จึงจะมีคะแนนเป็น 4 จะเห็นได้ว่าเป็นความสัมพันธ์ร่วมกันซึ่งเราสามารถเขียนได้ใหม่ในรูปที่ 3.20 และเราเรียกความสัมพันธ์ในลักษณะเช่นนี้ว่าเนสติ้ง



รูปที่ 3.20 แสดงการเขียนแฟกท์ไทป์ของผลการเรียนในรูปของ Nesting

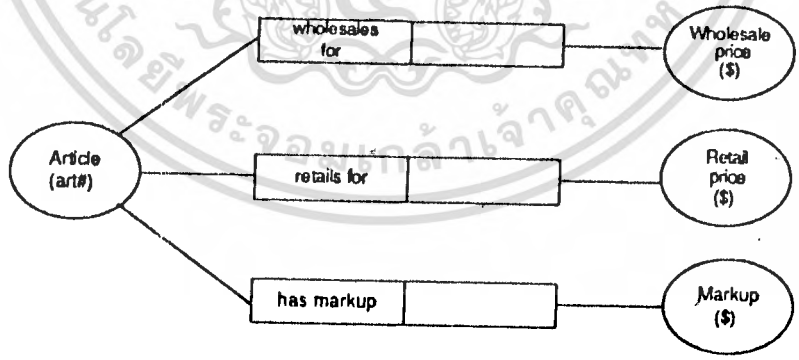
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 3 ปรับปรุงแผนภาพและหาแฟกท์ไทป์แบบหาได้ (derived fact type)

การทำงานในขั้นตอนนี้มีจุดประสงค์เพื่อที่จะหาว่ามีเอนติตี้ไทป์ใดบ้าง ที่เราสามารถที่จะลดลงได้บ้าง เพื่อให้เห็นได้ชัดเจนเราลองมาดูตัวอย่างในตารางที่ 3-4 ซึ่งเป็นรายการขายสินค้าและเงินทอน และจากตารางดังกล่าว เราสามารถนำมาเขียนเป็นแผนภูมิระดับแนวคิดได้ดังรูปที่ 3-8 ซึ่งจากรูป แผนภาพประกอบด้วยเอนติตี้ไทป์ 4 ชนิด หากแต่พิจารณาแต่ละเอนติตี้ไทป์ให้ดีแล้วจะเห็นว่า ที่จริงแล้วเอนติตี้ไทป์ Wholesale price, Retail price, Markup นั้นสามารถจะรวมเป็นเอนติตี้ไทป์เดียวกันได้

Article	*Wholesale price ($)*	*Retail price ($)*	*Markup ($)*
A1	50	75	25
A2	80	130	50
A3	50	70	20
A4	100	130	30

ตารางที่ 3.3 แสดงตัวอย่างข้อมูลรายการขายสินค้า

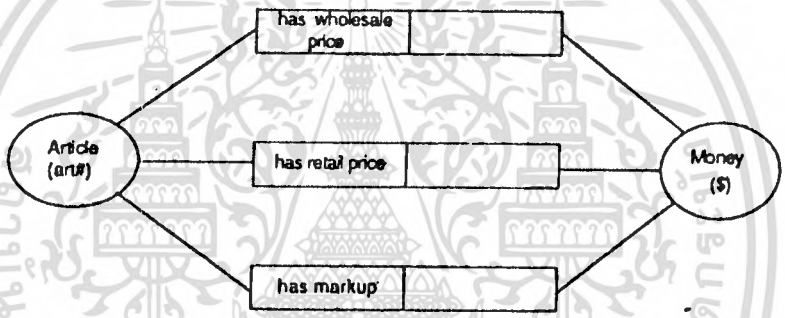


รูปที่ 3.21 แสดงแผนภาพจำลองข้อมูลระดับแนวคิด

และนอกจากนั้นหากเราสังเกตข้อมูลในบรรทัดแรกของตาราง จะเห็นได้ว่าเงินทอนนั้นเป็นข้อมูลที่สามารถหาได้จากเงินที่จ่ายกับราคาขายตามสูตร Markup = Retail price

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-Wholesale price ในลักษณะเช่นนี้หากเราเก็บข้อมูลทั้ง 3 ตัวแล้ว ก็จะถือว่าเป็นการเก็บแบบซ้ำซ้อน และต้องคอยรักษาความถูกต้องของข้อมูลตามความสัมพันธ์อีกด้วย ดังนั้นเพื่อที่จะขจัดปัญหาดังกล่าว เราจะเก็บข้อมูลเพียง Retail price และ Wholesale price เท่านั้น และเก็บMarkup อยู่ในรูปของกฎแสดงความสัมพันธ์แทน การที่เก็บเช่นนี้จะมีข้อดีคือ จะสามารถป้องกันความผิดพลาดได้กล่าวคือ หากเราเก็บข้อมูลทั้งสามตัว หากค่าของเงินทอนของสินค้าตัวหนึ่งไม่เท่ากับค่าเงินที่จ่ายลบด้วยราคาขายก็จะเป็นการขัดแย้งกันของข้อมูล นอกจากนั้นยังประหยัดเนื้อที่ในการเก็บอีกด้วย และเราจะเรียกแฟกท์ไทป์แบบที่เก็บเป็นกฎนี้ว่าแฟกท์ไทป์แบบหาได้(Derived Fact Type) และระบุในแผนภาพโดยเขียนเครื่องหมายดอกจันไว้ที่ด้านล่างของชนิดความจริงนั้น ดังนั้นจากแผนภาพในภาพที่ 3.21 เราสามารถนำมาเขียนใหม่ได้ดังภาพที่ 3.22



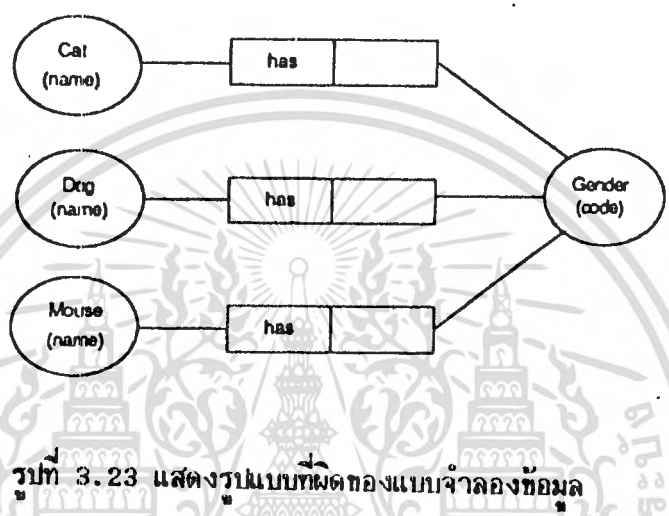
รูปที่ 3.22 แสดงแผนภาพจำลองข้อมูลระดับแนวคิด

ต่อไปเราลองมาดูแผนภาพของแบบจำลองข้อมูลระดับแนวคิดอีกแบบหนึ่งที่แสดงไว้ในรูปที่ 3.23 จากรูปจะเห็นว่ามีเอนติตี้ไทป์ Cat, Dog และ Mouse ซึ่งเอนติตี้ไทป์ทั้งสามนี้ต่างมีเลเบิ้ลไทป์ที่เหมือนกันคือ Name และยังมีแฟกท์ไทป์ที่เหมือนกันอีกด้วย ในกรณีเช่นนี้เราจะรวมเอนติตี้ไทป์ Cat, Dog และ Mouse เข้าด้วยกัน เป็นเอนติตี้ไทป์เดียวคือ Animal ดังแสดงในภาพที่ 3.24

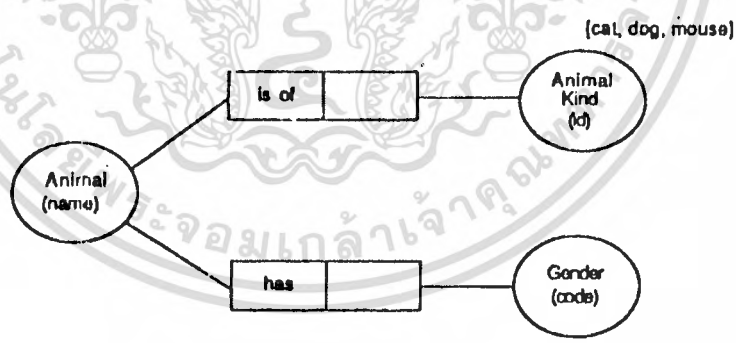
อย่างไรก็ตามจะสังเกตได้ว่าแบบจำลองข้อมูลในภาพที่ 3.23 นั้นอนุญาตให้สัตว์ประเภทต่างๆ มีชื่อที่ซ้ำกันได้ แต่แบบจำลองข้อมูลในภาพที่ 3.24 ไม่อนุญาตให้ทำเช่นนั้น ดังนั้นหากเราจะใช้แบบจำลองข้อมูลในรูปแบบนี้ ก็จะต้องทำการเปลี่ยนชื่อของสัตว์เล็กน้อยเพื่อไม่ให้ซ้ำกันและเพื่อที่จะแก้ปัญหาที่จุดนี้ เราจะสร้างเอนติตี้ไทป์ใหม่ชื่อ AnimalKind โดยกำหนดให้มีชนิดเลเบิ้ลเป็น id เพื่อให้ระบุชนิดของสัตว์ ในลักษณะเช่นนี้ก็จะยอมให้มีชื่อที่ซ้ำกันได้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากแบบจำลองข้อมูลในระดับแนวคิดที่สร้างขึ้นมาใหม่นี้ จะเห็นได้ว่าง่ายกว่าแบบเก่ามาและยังมีแฟกท์ไทป์ที่น้อยกว่าอีกด้วย และนอกจากนั้นหากเรามีชนิดของสัตว์ที่เพิ่มขึ้นเช่นอาจจะเพิ่ม Bird เข้าไป แบบจำลองข้อมูลใหม่นี้ก็ยังสามารถที่จะรองรับได้ ขณะที่แบบเก่าทำไม่ได้ และท้ายสุดเมื่อเราแปลงจากแบบจำลองข้อมูลไปเป็นตารางในฐานข้อมูลแบบสัมพันธ์ก็ยังจะได้จำนวนตารางที่น้อยกว่าอีกด้วย



รูปที่ 3.23 แสดงรูปแบบที่ผิดของแบบจำลองข้อมูล



รูปที่ 3.24 แสดงผลของการนำขั้นตอนที่ 3 มาใช้แก้ไข

จากที่กล่าวมาทั้งหมดในขั้นตอนที่ 3 นี้สามารถที่จะสรุปได้ดังต่อไปนี้

- หากมีเอนติตี้เดียวกันเป็นสมาชิกของ 2 เอนติตี้ไทป์ให้ทำการรวมทั้ง 2 เอนติตี้นั้น
- หากมีเอนติตี้ของเอนติตี้ไทป์ที่ต่างกัน แต่มีความหมายเดียวกัน ให้ทำการรวมทั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 เอนติตี้นั้น

- หากตัวอย่างข้อมูลของเอนติตี้ไทป์ที่ต่างกันมี role ที่เหมือนกัน ให้รวมชนิดเอนติตี้ทั้งสอง และหากจำเป็นให้เพิ่มแฟกท์ไทป์อีก เพื่อรักษาความแตกต่างเดิมไว้
- หากแฟกท์ไทป์ใดที่สามารถที่จะหาได้จากแฟกท์ไทป์อื่นให้ใส่เครื่อง หมายดอกจันไว้ที่ชนิดความจริงนั้น

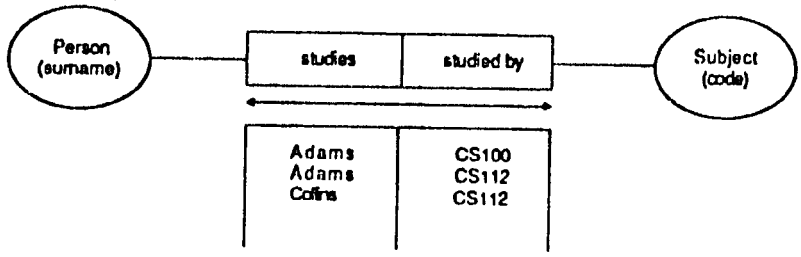
ขั้นที่ 4 คอนสเตรนท์แบบยูนีคเนสส์ (Uniqueness Constraint)

ที่ผ่านมาเราได้ทำการสร้างแผนภาพระดับแนวคิดขึ้น ทั้งในแบบที่เป็นเก็บจริง(store fact type) และแบบหาได้ (derive fact type) สำหรับในขั้นตอนที่เหลือเราจะกล่าวถึงกฎข้อจำกัดหรือคอนสเตรนท์ (constraint) โดยในขั้นตอนนี้เราจะกล่าวถึงคอนสเตรนท์แบบยูนีคเนสส์ (Uniqueness Constriant) ซึ่งเป็นคอนสเตรนท์ที่สำคัญมาก ซึ่งจะมีบทบาทมากในภายหลังในขั้นตอนการแปลงจากฐานข้อมูลระดับแนวคิด (Conceptual Schema) เป็นฐานข้อมูลแบบสัมพันธ์(Relational Schema)

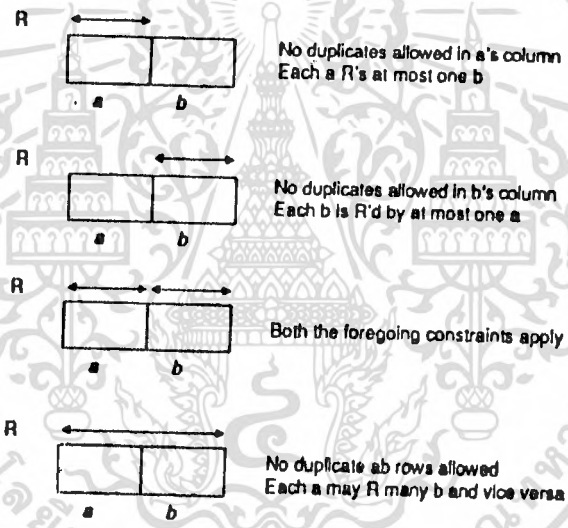
สำหรับกฎข้อบังคับ ซึ่งก็บอกอยู่แล้วโดยมันจะทำหน้าที่ควบคุมความถูกต้องของฐานข้อมูล (static constraints apply to every possible state of database) เช่นอาจจะกำหนดว่า "นักศึกษาแต่ละคนจะสังกัดได้เพียงภาควิชาเดียวเท่านั้น" และโดยเฉพาะคอนสเตรนท์แบบยูนีคเนสส์นั้นทุกๆ แฟกท์ไทป์ (store) จะต้องมีคอนสเตรนท์แบบยูนีคเนสส์ บังคับอยู่อย่างน้อย 1 ตัว เพราะในแต่ละแฟกท์ไทป์นั้นจะต้องไม่มีข้อมูล 2 แถวที่ซ้ำกันเกิดขึ้น

สำหรับการเขียนคอนสเตรนท์แบบยูนีคเนสส์นั้นจะเขียนโดยใช้ลูกศรที่มี 2 หัวเช่น สำหรับแฟกท์ไทป์ของข้อมูลที่ว่า นักศึกษาแต่ละคนจะลงเรียนได้หลายวิชา และแต่ละวิชาก็มีนักศึกษาได้หลายคน จะสามารถเขียน ได้ดังรูปที่ 3.25 ซึ่งจะหมายถึงว่านักศึกษาหนึ่งคนจะลงเรียนในรายวิชาหนึ่งได้เพียงครั้งเดียวเท่านั้น ดังนั้นเมื่อรวมข้อมูลของทั้ง 2 คอลัมน์แล้วจะไม่ซ้ำ(Unique) แต่ข้อมูลในแต่ละคอลัมน์จะมีการซ้ำได้ และเรียกความสัมพันธ์ของแฟกท์ไทป์แบบนี้ว่าแบบ many to many สำหรับความสัมพันธ์ในแบบต่างๆ ของแฟกท์ไทป์แสดงไว้ในรูป 3.26 ซึ่งได้แก่แบบ one to many, many to one, one to one และ many to many ซึ่งจากรูปก็ได้แสดงความหมายไว้อย่างชัดเจนดีอยู่แล้ว จึงไม่กล่าวอะไรมากไปกว่านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



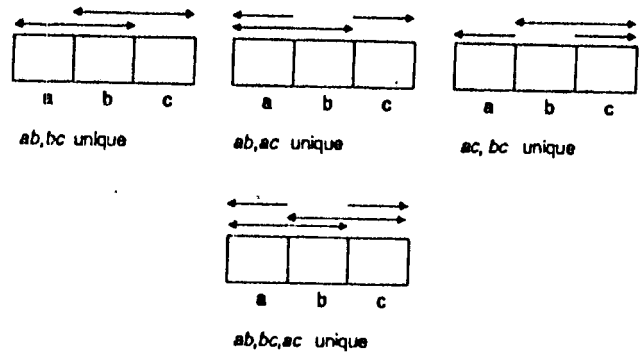
รูปที่ 3.25 แสดงตัวอย่างการใช้งานคอนสเตรนท์แบบยูนีคเนสส์



รูปที่ 3.26 แสดงแบบต่างๆของคอนสเตรนท์แบบยูนีคเนสส์ สำหรับแฟกท์ไทป์แบบไบนารี่

อย่างไรก็ตามสิ่งที่กล่าวมาทั้งหมด เราได้กล่าวถึงเฉพาะกับแบบที่เป็นไบนารี่เท่านั้น หากแต่ในความเป็นจริงแล้วยังมีแฟกท์ไทป์ที่มี role มากกว่า 2 role และในกรณีดังกล่าว เราก็ใช้หลักที่ว่าสำหรับแฟกท์ไทป์ที่มี n role ใดๆ จะสามารถมี role ที่ไม่มี Uniqueness Constriant กำกับได้เพียง 1 role เท่านั้นเช่นแฟกท์ไทป์ที่มี 3 role จะสามารถมี คอนสเตรนท์แบบยูนีคเนสส์ ที่เป็นไปได้ทั้งหมด 4 แบบดังรูปที่ 3.27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

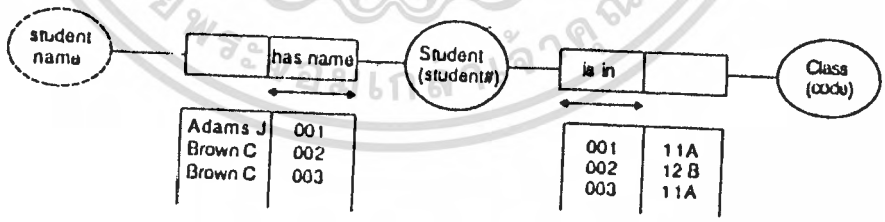


รูปที่ 3.27 แสดงคอนสเตรนท์แบบยูนีคเนสส์ของเทอร์นารี่

เท่าที่ผ่านมา เราได้กล่าวถึงคอนสเตรนท์แบบยูนีคเนสส์ ในแบบที่โรลทุกโรล ที่อยู่ภายใต้คอนสเตรนท์ หนึ่ง จะอยู่ในเฟกท์ไทป์เดียวกันเท่านั้น ต่อไปเราลองมาดูข้อมูลในตารางที่ 3.4 จากตารางจะเห็นได้ว่านักศึกษาแต่ละคนจะมีชื่อเพียงชื่อเดียว และนักศึกษาแต่ละคนจะมีชั้นเรียนเพียงชั้นเรียนเดียว จากข้อมูลดังกล่าวจะสามารถเขียนแผนภาพระดับแนวคิดได้ดังรูปที่ 3.28

Student:	student#	name	class
	001	Adams J	11A
	002	Brown C	12B
	003	Brown C	11A

ตารางที่ 3.4 แสดงตัวอย่างข้อมูล

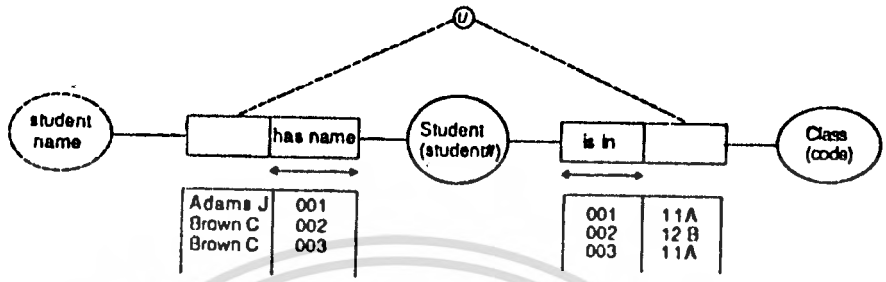


รูปที่ 3.28 แสดงแบบจำลองข้อมูลที่ได้จากตาราง

อย่างไรก็ตามหากสังเกตุให้ดีจากตาราง จะเห็นได้ว่าผลรวมของชื่อและชั้นเรียนก็จะเป็นข้อมูลที่ไม่ซ้ำเช่นกัน และในกรณีเช่นนี้ NIAM ก็อนุญาติให้สามารถสร้างคอนสเตรนท์แบบยูนีคเนสส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระหว่างแฟกท์ไทป์ 2 แฟกท์ไทป์ได้ ดังรูปที่ 3.29



รูปที่ 3.29 แสดงคอนสเตรนท์แบบยูนีคเนสส์ภายนอก

จากรูปจะเห็นได้ว่าเราจะใช้เครื่องหมาย "u" ล้อมด้วยวงกลมและมีเส้นโยงไปยังโรลที่อยู่ภายใต้คอนสเตรนท์แบบยูนีคเนสส์ นั้น การสร้างคอนสเตรนท์ ระหว่างชนิดความจริงนี้เราเรียกว่า Inter-fact-type และเรียกคอนสเตรนท์ที่อยู่ในแฟกท์ไทป์เดียวกันนี้ว่า Intra-fact-type จากที่ผ่านมาทั้งหมดนี้ เราจะเห็นได้ว่าการตรวจสอบความถูกต้องโดยใช้ตัวอย่างในไนแอม นั้นมีความสำคัญมาก ทั้งนี้เพราะจะทำให้ผู้ออกแบบสังเกตุ ข้อมูลบางอย่างเพิ่มเติมได้จากข้อมูลที่ให้มานี้

ขั้นตอนที่ 5 ตรวจสอบอาริตี้ (Arity Checks)

หากในขั้นตอนที่ผ่านมานั้น เราได้ทำการออกแบบอย่างระมัดระวังแล้ว การทำงานในขั้นตอนนี้ นับว่าเป็นสิ่งที่ไม่จำเป็นเลย อย่างไรก็ตาม จากความจริงที่ว่า นักปราชญ์ยังรู้พลั้งนั้นเป็นข้อเตือนใจได้เป็นอย่างดี เพราะแม้ว่าเราจะเป็นผู้ออกแบบที่ชำนาญเพียงใดก็ตาม ก็ยังอาจเกิดการผิดพลาดขึ้นได้

ความผิดพลาดที่เกิดขึ้นนั้นมีอยู่ 2 ประการคือการออกแบบแฟกท์ไทป์ที่สั้นหรือยาวเกินไป ในกรณีของแฟกท์ไทป์ที่ยาวเกินไปนั้น ก็จะต้องทำการแยกแฟกท์ไทป์นั้นออกเป็น 2 แฟกท์ไทป์ และสำหรับแฟกท์ไทป์ที่สั้นเกินไปนั้น ไม่สามารถที่จะบรรจุข้อมูลได้ครบ(information Loss) ดังนั้นก็จะต้องทำการรวม 2 แฟกท์ไทป์เป็นแฟกท์ไทป์เดียว

สำหรับการตรวจสอบความถูกต้องนั้น จริงๆ แล้วอาจจะมีวิธีการมากมาย สุดแต่ใครจะใช้วิธีการอย่างไร หากแต่ในที่นี้เราจะกล่าวถึงเพียง 3 วิธี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ใช้สามัญสำนึก หรือความรู้พื้นฐานในการพิจารณาตัดสินว่ามีข้อมูลสูญหายเพราะการแยกแฟกท์ไทป์ออกมาหรือไม่

2. ใช้กฎของการแยกจากกัน (Splittability Rules) ซึ่งจะกล่าวถึงภายหลัง

3. ใช้วิธีสร้างตัวอย่างขึ้นมาสำหรับแฟกท์ไทป์ และทำการแยกแฟกท์ไทป์นั้นออกโดยวิธี Projection (ซึ่งจะกล่าวถึงในภายหลัง) จากนั้นนำมารวมกันใหม่โดยใช้วิธีNatural Join และหากมีตัวอย่างเพิ่มขึ้นมาจากการ Join ก็แสดงว่าแฟกท์ไทป์นั้นเป็นขนิดความจริงที่ไม่สามารถทำการแยกได้

Person	*Degree*	*Subject*	*Rating*
Adams	BSc	CS112	7
Adams	BSc	CS110	6
Adams	BSc	PD102	7
Brown	BA	CS112	6
Brown	BA	PD102	7
Collins	BSc	CS112	7

ตารางที่ 3.5 แสดงตัวอย่างข้อมูลของผลการเรียน

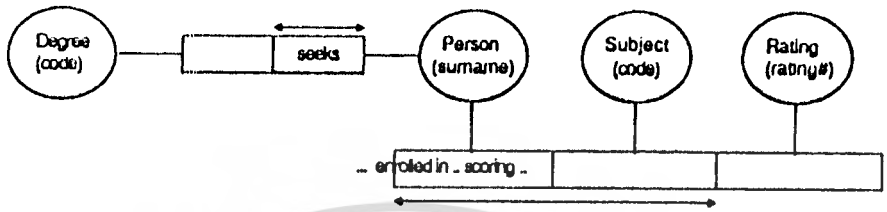
สำหรับวิธีการแรกนั้น อาจจะเป็นวิธีที่ดูไม่ค่อยจะมีหลักการสักเท่าใดนัก แต่จริงๆแล้วนับได้ว่าเป็นวิธีที่ดีวิธีหนึ่ง เพราะเป็นวิธีที่มีผลสอดคล้องกับขั้นตอนการกำหนด Elementary Facts เพื่อให้การทำงานในวิธีนี้ชัดเจนยิ่งขึ้น เราลองมาดูตัวอย่างต่อไปนี้ เราจะเริ่มจากข้อมูลในตารางที่ 3.5 และจากวิธีการที่ผ่านมา เราสามารถสร้าง Elementary Facts ได้ดังนี้

```
Person(surname) 'Adams' seeks Degree(code) 'BSc'.
Person(surname) 'Adams' enrolled in Subject(code) 'CS112'
scoring Rating(Nr) 7.
```

ในที่นี้คำว่า Seeks หมายถึงสำเร็จตามหลักสูตร และจาก Elementary Facts ดังกล่าว เราสามารถนำมาเขียนเป็นแผนภาพระดับแนวคิดได้ดังรูปที่ 3-17 ที่นี้เราก็จะมาลองดูว่าหากเราต้องการจะแยกแฟกท์ไทป์แบบเทอร์นารี่ ในภาพออกเป็น 2 แฟกท์ไทป์จะได้หรือไม่ โดยพิจารณาว่าหากเราทำการแบ่งแล้ว เราก็จะได้ข้อมูลว่า Adams ได้คะแนนเท่ากับ 7และ Adams ลงเรียนในวิชา CS112 ซึ่งจะเห็นได้อย่างชัดเจนว่ามีข้อมูลบางส่วนหายไป(Information

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Loss) นั่นคือข้อมูลที่ว่า Adams ได้คะแนนเท่ากับ 7 ซึ่งไม่ทราบว่าได้คะแนนในวิชาใด ดังนั้น สรุปได้ว่าแฟกท์ไทป์นี้ไม่สามารถจะแยกได้



รูปที่ 3.30 แสดงแบบจำลองข้อมูลที่ได้จากตารางที่ 3.5

1. การใช้กฎการแยก (Splittability Rule)

จริงๆ แล้วเราได้กล่าวถึงกฎที่ว่านี้ไปครั้งหนึ่งแล้ว ในขั้นตอนที่ผ่านมา นั่นคือกฎที่ว่าจะต้องมีโรลที่ไม่อยู่ภายใต้คอนสเตรนท์แบบยูนีค ได้ไม่เกิน 1 เท่านั้น

2. การตรวจสอบโดยใช้ Projection-Join

ในตอนที่แล้ว เราได้กล่าวถึงเงื่อนไขของการแยกแฟกท์ไทป์โดยใช้คอนสเตรนท์แบบยูนีค อย่างไรก็ตามในบางครั้งเงื่อนไขดังกล่าวก็ไม่สามารถใช้งานได้และนอกจากนั้น ผู้ออกแบบบางคนก็อาจจะต้องการวิธีการที่แน่นอนในการตรวจสอบ ซึ่งเราจะได้กล่าวต่อไป

ต่อไปเราก็จะกล่าวถึงหลักการหรือวิธีการที่แน่นอนที่จะใช้ในการแก้ปัญหานี้ได้ ในวิธีนี้เราจะใช้การทำงาน 2 อย่างคือ Projection และ Joining โดยที่ Projection ก็คือการสร้างตารางใหม่ขึ้นมาจากตารางเดิม 1 ตารางโดยเลือกคอลัมน์มาเพียงบางคอลัมน์ ซึ่งในตารางใหม่นี้ บางครั้งก็จะเกิดการซ้ำกันขึ้น ดังนั้นเราก็จะต้องตัดข้อมูลในบางแถวที่มีการซ้ำกันออกไป สำหรับการ Join ก็คือการรวมตาราง 2 ตารางขึ้นมาเป็นตารางใหม่

ทีนี้เราลองมาดูข้อมูลในตารางที่ 3.6 จากรูปจะเห็นได้ว่าเราทำการ Projection ตารางออกเป็น 2 ตาราง ดังแสดงในตารางที่ 3.7 และเมื่อเรานำ 2 ตารางนี้มาทำการ Join กันใหม่ก็จะได้ข้อมูลดังตารางที่ 3.8 ซึ่งจะเห็นได้ว่ามีแถวของข้อมูลเกิดขึ้นใหม่ (แถวที่มีเครื่องหมาย X) ซึ่งข้อมูลเหล่านี้เป็นข้อมูลที่ไม่ได้มีอยู่ในตารางที่เราทำการ Projection มาดังนั้นจะสรุปได้ว่าเราไม่สามารถแยกแฟกท์ไทป์นี้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Person	*Degree*	*Subject*
	Adams	BSc	CS112
	Adams	BSc	CS110
	Adams	BSc	PD102
	Brown	BA	CS112
	Brown	BA	PD102
	Collins	BSc	CS112

ตารางที่ 3.6 แสดงตัวอย่างข้อมูล

Person	*Degree*	*Person*	*Subject*
Adams	BSc	Adams	CS112
Brown	BA	Adams	CS110
Collins	BSc	Adams	PD102
		Brown	CS112
		Brown	PD102
		Collins	CS112

ตารางที่ 3.7 แสดง Projection ของข้อมูลในตารางที่ 3.6

	Person	*Subject*	*Rating*
	Adams	CS112	7
X	Adams	CS112	6
X	Adams	CS110	7
	Adams	CS110	6
	Adams	PD102	7
X	Adams	PD102	6
	Brown	CS112	6
X	Brown	CS112	7
X	Brown	PD102	6
	Brown	PD102	7
	Collins	CS112	7

ตารางที่ 3.8 แสดงผลของการ Join ข้อมูลในตารางที่ 3.6

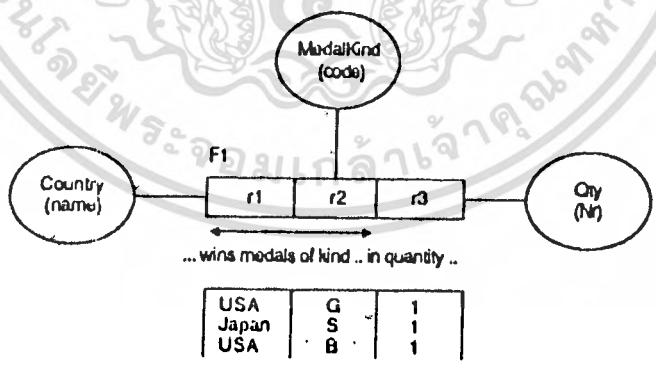
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 6 คอนสเตรนท์อื่น ๆ

ในขั้นตอนที่ผ่านมา เราได้กล่าวถึงการสร้างแบบจำลองข้อมูลโดยเริ่มตั้งแต่การสร้าง " ความจริงพื้นฐาน" จนกระทั่งการตรวจสอบความถูกต้องขั้นต้น ซึ่งเมื่อถึงขั้นนี้จะเห็นได้ว่าเราได้กล่าวถึง คอนสเตรนท์ ไปเพียง 1 ชนิดเท่านั้น และจากที่กล่าวไปแล้วว่าไนแอมเป็นแบบจำลองข้อมูลในระดับแนวคิดซึ่งจะเน้นที่ความหมายเป็นสำคัญ และสิ่งที่สำคัญรองลงมาก็คือข้อจำกัดเพื่อความถูกต้อง ดังนั้นสิ่งที่เราจะกล่าวต่อไป ก็คือการสร้างข้อจำกัด (คอนสเตรนท์) ต่างๆเพื่อให้ไนแอมเป็นแบบจำลองข้อมูลที่มีความถูกต้องของข้อมูลมากที่สุด

- คอนสเตรนท์แบบเอนติตี้ไทป์ (entity type constraint)

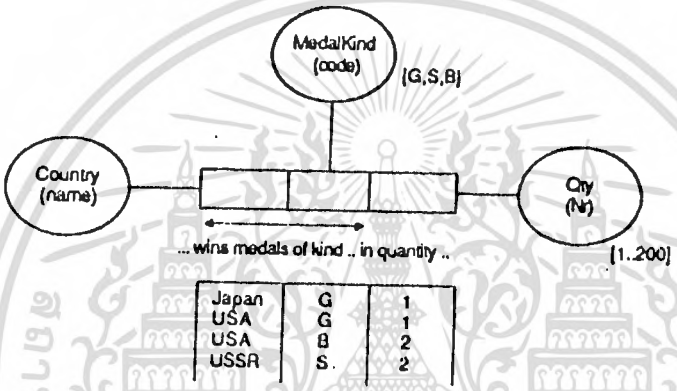
จากรูปที่ 3.31 เป็นแบบจำลองข้อมูลของการได้เหรียญของประเทศต่างๆ จากรูปเอนติตี้ไทป์ Country เป็นเซ็ทของประเทศต่างๆ ทั้งหมดในโลกที่เป็นไปได้ ในกรณีนี้หากเราต้องการที่จะหาว่ามีสมาชิกใดบ้างที่สามารถอยู่ในเอนติตี้นี้ได้ อาจจะเป็นงานที่ยากเพราะมีการตั้งประเทศขึ้นมาใหม่ๆ อยู่เรื่อยๆ อย่างไรก็ตามสำหรับในบางเอนติตี้ไทป์นั้นง่ายมากที่จะระบุถึงสมาชิกของเอนติตี้นั้น เช่น เอนติตี้ของเหรียญที่ใช้ในกีฬาโอลิมปิกก็จะมีเหรียญ ทอง เงิน และบรอนซ์ และในบางครั้งหากเราต้องการให้ข้อมูลที่อยู่ในเอนติตี้ไทป์ MedalKind เป็นได้เพียง(G,S,B) เท่านั้น เราสามารถทำได้โดยการระบุ คอนสเตรนท์แบบเอนติตี้ไทป์ บนเอนติตี้ไทป์นั้น



รูปที่ 3.31 แสดงตัวอย่างของแบบจำลองข้อมูลของผลการแข่งกีฬา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งในกรณีของเอนติตี้เหรียญนี้เราสามารถเขียนแบบจำลองข้อมูลขึ้นใหม่ โดยระบุคอนสเตรนท์แบบเอนติตี้ไทป์ ได้ดังในรูปที่ 3.32 ซึ่งจากรูปจะเห็นได้ว่า เราสามารถระบุเซ็ตของเอนติตี้โดยเขียนเซ็ทไว้ที่ด้านข้างของเอนติตี้ไทป์ โดยที่ในกรณีที่มีสมาชิกจำกัดก็จะใช้เครื่องหมาย () เช่น (G,S,B) หมายถึงสมาชิกของเอนติตี้ไทป์นี้จะต้องเป็นสมาชิกของเซ็ต (G,S,B)ด้วย และหากเป็นช่วงเราจะใช้เครื่องหมาย [] เช่น [1..200] ซึ่งหมายความว่าสมาชิกของเอนติตี้ไทป์นี้จะต้องมีค่าอยู่ในช่วง 1 ถึง 200 เท่านั้น



รูปที่ 3.32 แสดงการใช้ คอนสเตรนท์แบบเอนติตี้ไทป์

- คอนสเตรนท์แบบแมนดาทอรี่ (Mandatory constraint)

ก่อนที่เราจะอธิบายถึงรายละเอียดของ คอนสเตรนท์ นี้ ในขั้นต้นนี้เราจะอธิบายถึงสาเหตุที่เราจะต้องมี คอนสเตรนท์ นี้ โดยเราจะพิจารณาจากตัวอย่างข้อมูลในตารางที่ 3.9 จะเห็นได้ว่ามีข้อมูลอยู่ช่องหนึ่งที่เราใส่เครื่องหมาย "?" ไว้เพื่อแสดงว่ายังไม่มีการบันทึกข้อมูลในตารางนี้ เช่น Brown S ที่จริงแล้วอาจจะมีโทรศัพท์ก็เป็นได้ แต่ยังไม่ได้รับการบันทึก

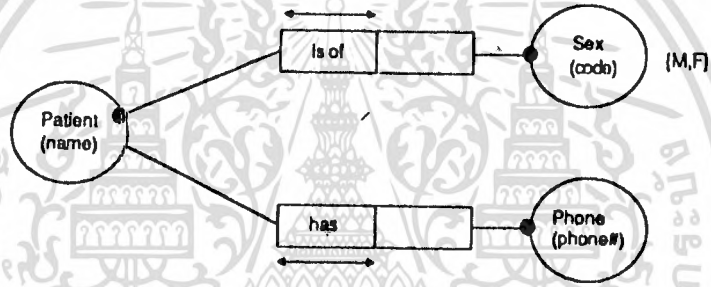
อย่างไรก็ตามหากข้อมูลที่ยังไม่ได้รับการบันทึกนั้น เป็นข้อมูลที่สำคัญเช่นข้อมูลเพศการที่เราจะไม่บันทึกข้อมูลนั้นลงไปก็อาจจะนับได้ว่าเป็นการไม่ถูกต้องนัก และอาจจะเป็นอุปสรรคต่อการนำไปใช้งาน เช่นข้อมูลชั้นปีของนักศึกษา หากไม่ได้รับการบันทึกก็อาจจะทำให้ความหมายอื่นๆ ที่สัมพันธ์กันคลาดเคลื่อนไปได้ ดังนั้นสำหรับในข้อมูลบางตัวที่ระบุให้ "ต้องบันทึก" นั้น เราสามารถทำได้โดยระบุคอนสเตรนท์แบบแมนดาทอรี่ โดยใช้เครื่องหมาย "จุด" ใส่ไว้ที่จุดเชื่อมระหว่างเส้นที่ลากมาจากเอนติตี้ไทป์และ โรล ตรงข้ามของ โรล ที่จะระบุ คอนสเตรนท์เช่นจากรูปที่ 3.32 ซึ่งมีความหมายว่าคนไข้ทุกคนจะต้องระบุเพศ จะเขียนโดยใส่จุดที่จุดเชื่อมระหว่างเอนติตี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Patient และเส้นที่ลากมาจาก โรล ซึ่งเป็นด้านตรงข้ามของ โรล ที่ต้องการระบุ คอนสเตรนท์

Patient	*Sex*	*Phone*
Adams C	F	2057642
Brown S	F	?
Collins T	M	8853020

ตารางที่ 3.9 แสดงข้อมูลของคนไข้

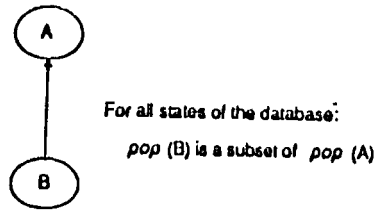


รูปที่ 3.33 แสดงการใช้งานคอนสเตรนท์แบบแมนดาทอรี่

- สับไทป์ (Subtype)

สำหรับเรื่องของสับไทป์นี้เราจะใช้แผนภูมิของออยเลอร์ (Euler'sDiagram) ในการแสดงความสัมพันธ์ระหว่างเอนติตี้ จากที่เราได้กล่าวมาแล้วว่าเอนติตี้เป็นเซ็ทที่นี้ลองสมมติว่าเรามีเอนติตี้ Manager ซึ่งเป็นเซ็ทของผู้จัดการทั้งหมด และเอนติตี้ Employeeซึ่งเป็นเซ็ทของพนักงานที่ทำงานอยู่ และจากความจริงที่ว่าผู้จัดการทุกคนย่อมเป็นพนักงานด้วยเช่นกัน ดังนั้นหากจะเขียนแบบจำลองโดยให้เอนติตี้ Manager และ Employee ไม่เกี่ยวข้องซึ่งกันและกันนั้น ก็คงจะเป็นการผิดความหมายแน่นอน และลักษณะความสัมพันธ์ในลักษณะของสับเซตเช่นนี้เราจะเรียกว่าสับไทป์ ซึ่งจะเขียนได้ดังรูปที่ 3.34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.34 แสดงความสัมพันธ์ระหว่างซูเปอร์ไทป์และสับไทป์

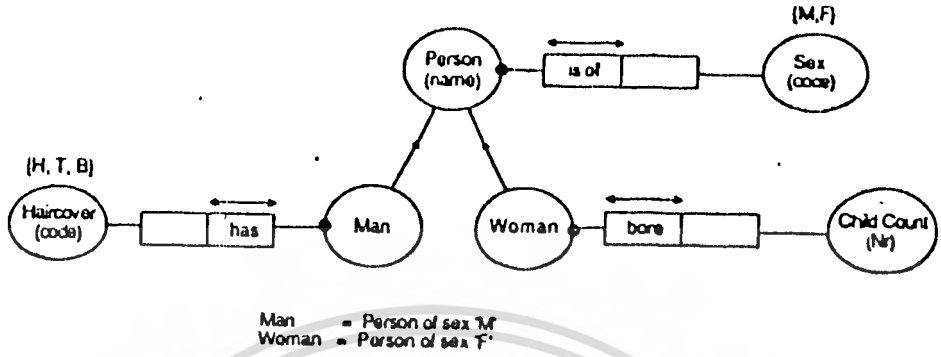
จากรูปหมายถึงว่าเอนติตี้ B เป็นสับเซตของเอนติตี้ไทป์ A และจะเรียก A ว่าซูเปอร์ไทป์ และเรียก B ว่าสับไทป์นั่นย่อมหมายถึงว่าสมาชิกทุกตัวใน B ก็ย่อมจะมีคุณสมบัติของเอนติตี้ A ด้วย เราลองมาดูตัวอย่างกันอีกตัวอย่างหนึ่งโดยอาศัยตารางที่ 3.10 จากตารางจะเห็นได้ว่าเฉพาะเพศชายเท่านั้นจึงจะมีการบันทึก Haircover และเฉพาะเพศหญิงเช่นกันที่มีการบันทึกจำนวนบุตร (NrChilden) ในลักษณะดังกล่าวนี้หากเราใช้วิธีการเทียบในแบบที่ผ่านมาก็ย่อมได้และไม่ผิดหากแต่ไม่สามารถแสดงความหมายได้ครบ เพราะจากที่กล่าวมาว่าไนแอมเป็นแบบจำลองข้อมูลระดับแนวคิดดังนั้นจึงควรจะใช้วิธีการที่จะแสดงความหมายได้ดีกว่าและนั่นก็คือสับไทป์นั่นเองซึ่งจะเทียบเป็นแผนภาพได้ดังรูปที่ 3.35

Person	*Sex*	*Haircover*	*NrChildren*
Jones E	F	-	2
Smith J	M	T	-
Blow J	M	B	-
Lane L	F	-	0
Blossom B	F	-	5

"-" = "not to be recorded"

ตารางที่ 3.10 แสดงตัวอย่างข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

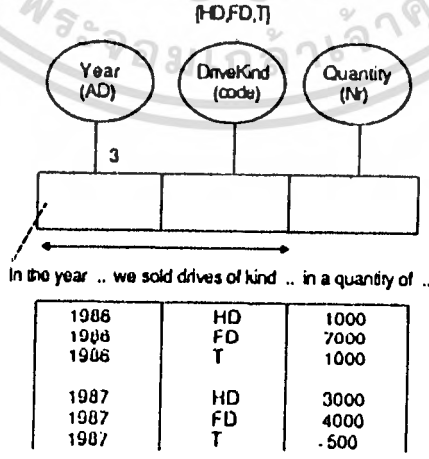


รูปที่ 3.35 แสดงการใช้สับไทป์ในแบบจำลองข้อมูล

- ออคเคอเรนซ์ฟรีเควนซี่ (Occurrence frequencies)

ในตอนนี้เราจะกล่าวถึง คอนสเตรนท์ อีกตัวหนึ่งที่จะใช้ในการระบุจำนวนครั้งที่เอนติตี้ไทป์จะถูกใช้ได้ใน โรล หนึ่งๆ คอนสเตรนท์ นี้มีชื่อเรียกว่า ออคเคอเรนซ์ ฟรีเควนซี่

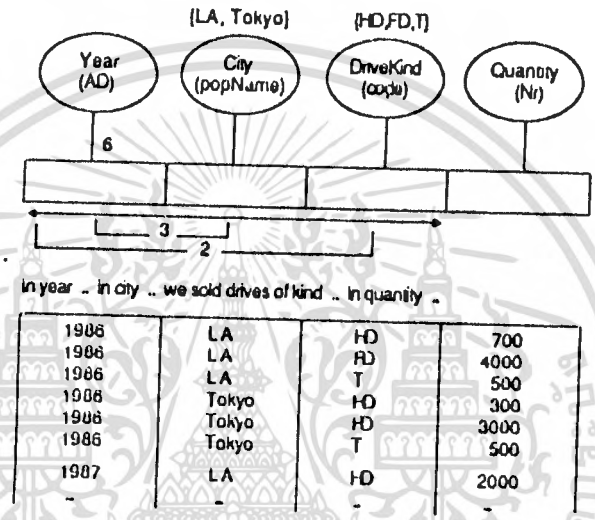
ออคเคอเรนซ์ ฟรีเควนซี่ โดยทั่วไปจะใช้ในการระบุว่าจะมีอินสแตนซ์ ที่เกิดขึ้นในคอลัมน์ที่ระบุไว้เป็นจำนวนเท่าใด สำหรับวิธีระบุ คอนสเตรนท์ นี้เราจะใช้วิธีเทียบตัวเลขไว้เหนือหรือใต้ โรลที่ คอนสเตรนท์ นี้ระบุอยู่ ดังตัวอย่างในรูปที่ 3.36 จะเห็นว่าจะต้องมีการบันทึกการขายไดรฟ์ทั้ง 3 ชนิดในแต่ละปี ดังนั้นจึงต้องระบุออคเคอเรนซ์ ฟรีเควนซี่ 3 ไว้เหนือโรล year ซึ่งจะหมายถึงว่าหากมีการบันทึกปีใดปีหนึ่งลงในโรลนี้ จะต้องบันทึกข้อมูลของไดรฟ์ทั้ง 3 ชนิดพร้อมกัน



รูปที่ 3.36 แสดงตัวอย่างการใช้ออคเคอเรนซ์ ฟรีเควนซี่

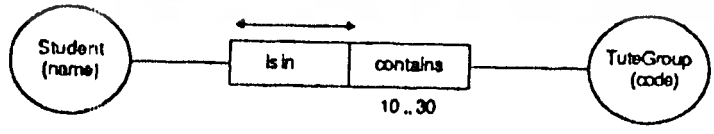
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปเราลองมาดูออคเคอเรนซ์ ฟรีเควนซี่ที่มีความซับซ้อนมากขึ้น จากตัวอย่างในรูปที่ 3.37 เป็นแฟกท์ไทป์ที่เก็บความสัมพันธ์ของไดรฟ์ชนิดต่างๆ ที่ขายได้ในแต่ละปี ในแต่ละเมือง ว่าเป็นจำนวนเท่าใด สำหรับออคเคอเรนซ์ ฟรีเควนซี่ 3 หมายถึงผลรวม(Combiantion) ของ role year และ city ที่เกิดขึ้นในตารางจะต้องมีได้ 3 เท่านั้น ออคเคอเรนซ์ ฟรีเควนซี่ 2 หมายถึงผลรวมของ role year และ DriveKind เกิดขึ้นได้ 2เท่านั้น และออคเคอเรนซ์ ฟรีเควนซี่ 6 หมายถึงจะมี role year เกิดขึ้นได้ครั้งละ 6 เท่านั้น



รูปที่ 3.37 แสดงตัวอย่างการใช้งานออคเคอเรนซ์ ฟรีเควนซี่

อย่างไรก็ตาม นอกจากจะระบุออคเคอเรนซ์ ฟรีเควนซี่ เป็นค่าคงที่แล้วยังสามารถที่จะระบุได้เป็นช่วงอีกด้วยเช่น 1..3 หมายถึงจะมี role นั้นเกิดขึ้นได้ตั้งแต่ 1 ถึง 3เท่านั้น เช่นจากรูปที่ 3.38 จะเป็นแฟกท์ไทป์ที่มีความหมายว่าในแต่ละ TuteGroup จะมีนักศึกษาได้ 10-30 คนเท่านั้น



รูปที่ 3.38 แสดงตัวอย่างของออคเคอเรนซ์ ฟรีเควนซี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ชนิดของข้อมูล (Data type)

ในขั้นตอนก่อนหน้านี้ เราได้กล่าวถึงเรื่องราวต่างๆ ระหว่างเอนติตี้ กับเลเบิ้ล ซึ่งโดยส่วนใหญ่ก็จะเป็นเรื่องที่เกี่ยวกับเอนติตี้ แต่ต่อจากนี้เราจะเน้นแต่เรื่องราวเลเบิ้ลบ้าง ว่ามีรายละเอียดอะไรบ้างที่ควรจะกล่าวเพิ่มเติม

ที่ผ่านมาเราได้ใช้ คอนสเตรนท์แบบเอนติตี้ไทป์ เพื่อใช้บังคับความถูกต้องของข้อมูลในแต่ละเอนติตี้ แต่สำหรับกรณีเลเบิ้ลแล้วเราก็มีข้อบังคับในทำนองนี้เช่นกัน โดยจะใช้ระบุชนิดที่เป็นไปได้ทั้งหมดของชื่อ เช่น ชื่อวิชาก็อาจจะระบุเป็น {aaddd} หมายถึงเซ็ทของตัวอักษร 2 ตัวตามด้วยตัวเลขอีก 3 ตัวเป็นต้น

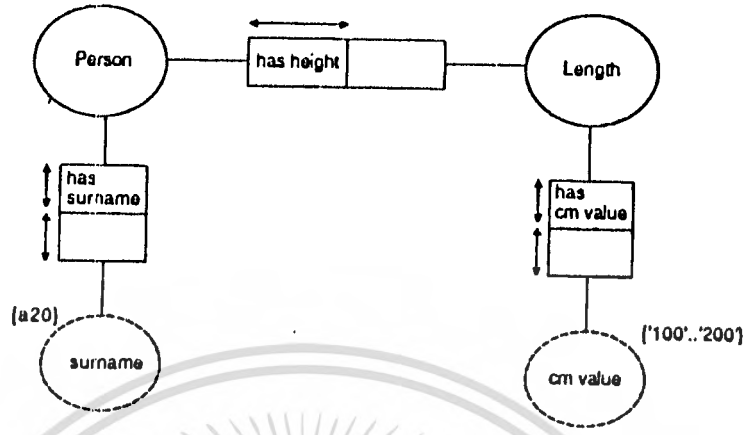
อย่างไรก็ตาม แม้ขณะนี้ก็ยังเป็นข้อถกเถียงกันอยู่ว่าควรจะรวมเอา คอนสเตรนท์แบบเลเบิ้ลไทป์ ไว้ในส่วนของข้อมูลระดับแนวคิด (conceptual level) หรือจะถือว่าอยู่ในส่วนของข้อมูลระดับภายนอก (external level) เช่น ในงานหนึ่งๆ เราอาจจะต้องการรูปแบบของการติดต่อกับผู้ใช้ (User Interface) ที่ต่างกันของวิชาต่างๆ ซึ่งอาจจะอ้างถึงโดยใช้ ตัวอักษร ตัวเลข อาจจะต้องขึ้นต้นด้วยภาษาอังกฤษ หรือภาษาไทย เป็นต้น

และเช่นเดียวกัน เราสามารถที่จะระบุเลเบิ้ลได้โดยอาศัยการให้ตัวอย่าง เช่นจากข้อมูลในตารางที่ 3-12 สำหรับเอนติตี้ Person เราจะใช้ Surname เป็นเลเบิ้ลซึ่งอาจจะระบุโดยเทียบว่า {c20} ซึ่งหมายถึงตัวอักษรจำนวน 20 ตัว และความสูงซึ่งอาจจะระบุโดย{'100'..'200'} หมายถึงจะมีค่าได้ตั้งแต่ 100-200 หรือระบุโดยใช้ [100..200] เป็นต้นดังแสดงในรูปที่ 3.39

Person	*City*	*Height (cm)*	*Chest (cm)*	*Mass (kg)*	*IQ*
Adams	Brisbane	175	100	77	100
Brisbane	Sydney	182	91	80	120
Collins	Sydney	173	80	67	100
Darwin	Darwin	175	95	90	95

ตารางที่ 3.11 แสดงตัวอย่างข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.39 แสดงตัวอย่างของแบบจำลอง

ขั้นตอนที่ 7 การอ้างถึงเลเบิ้ลไทป์ของเอนติตี้ไทป์ (Entity Identification schemes)

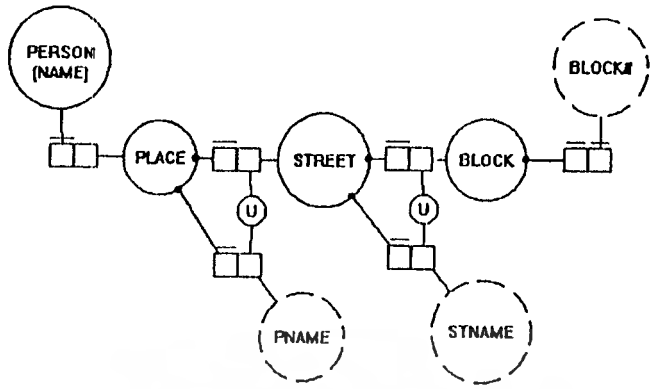
ที่ผ่านมาทั้งหมดเราได้สร้างแผนภาพ โดยสมมติว่าแต่ละเอนติตี้ไทป์มีความสัมพันธ์แบบเรฟเฟอร์เรนซ์ไทป์กับเลเบิ้ลไทป์เพียงเลเบิ้ลไทป์เดียว แต่ในบางครั้งก็มีเอนติตี้ไทป์ก็มีความสัมพันธ์กับหลายๆ เลเบิ้ลไทป์ได้เช่นกัน อย่างไรก็ตามจะต้องมีเรฟเฟอร์เรนซ์ไทป์อย่างน้อยหนึ่งเรฟเฟอร์เรนซ์ไทป์ที่สามารถใช้ในการระบุ (Identification) ถึงแต่ละสมาชิกในเอนติตี้ไทป์ได้ นั่นก็หมายความว่าจะต้องมีเรฟเฟอร์เรนซ์ไทป์อย่างน้อยหนึ่งเรฟเฟอร์เรนซ์ไทป์ที่มีความสัมพันธ์ในแบบ 1:1 กับเอนติตี้ไทป์นั้นเช่น เอนติตี้นักศึกษาอาจจะมีเลเบิ้ลหรือชื่อเรียก 2 แบบได้แก่ รหัสนักศึกษาและชื่อนักศึกษาอย่างไรก็ตาม อย่างน้อยก็จะต้องมีเลเบิ้ลรหัสนักศึกษาที่แต่ละรหัสจะต้องไม่ซ้ำและสามารถใช้ในการระบุถึงแต่ละสมาชิกในเอนติตี้นักศึกษาได้ และเราเรียกรหัสนักศึกษาว่าเป็นไอเดนติไฟเออร์ (Identifier)

ดังนั้นหากจะกล่าวโดยสรุปงานในขั้นตอนนี้ก็คือ การตรวจสอบว่าในแต่ละเอนติตี้จะต้องมี Identifier อย่างน้อยหนึ่ง เพื่อใช้ในการระบุถึงสมาชิกแต่ละตัวของเอนติตี้ไทป์ได้

- การอ้างถึงแบบไฮราคี่ (Hierarchical reference)

ยังมีการอ้างถึงที่สลับซับซ้อนอยู่อีกกรณีหนึ่ง ซึ่งต้องอาศัยการอ้างถึงหลายระดับขึ้น เรียกลักษณะเช่นนี้ว่า การอ้างถึงแบบไฮราคี่ แสดงดังรูปที่ 3.40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.40 แสดงการอ้างถึงแบบไฮราคี่

จากรูปจะเห็นว่าทั้งหมดนั้นเป็นการอ้างถึงของเอนติตี้ไทป์ PLACE เท่านั้น หากแต่มีการอ้างถึงทีละลำดับชั้น เริ่มตั้งแต่ชื่อถนนซึ่งขนานคับเอนติตี้ไทป์ BLOCK เนื่องจากเป็นไปได้ว่า ในต่างบล็อคอาจมีชื่อถนนที่ซ้ำกันได้ เมื่ออยู่ในระดับชั้นของเอนติตี้ไทป์ BLOCK ก็มีการอ้างถึงชื่อบล็อคซึ่งขนานคับเอนติตี้ไทป์ DISTRICT อีก ในเงื่อนไขเดียวกัน คืออาจมีบล็อคเดียวกันในต่างอำเภอได้อีก เช่นนี้เรื่อยไปจนมาสิ้นสุดที่ชื่อของ STATE ซึ่งไม่เกิดกรณีซ้ำซ้อนขึ้นอีก

ในลักษณะนี้เมื่อต้องการอ้างถึงที่อยู่ของคน ๆ หนึ่ง จะต้องอ้างถึงเลเบิ้ลไทป์ต่าง ๆ มากมาย จึงนับเป็นการอ้างถึงในกรณีพิเศษกรณีหนึ่ง กรณีของไฮราคี่นี้อาจปรากฏสับไทป์ขึ้น ณ ที่ใดที่หนึ่งในลำดับชั้นได้ เช่น กรณีที่มีกิ่งกำเภอที่เป็นสับเซตของอำเภออีกที เรียกกรณีเช่นนี้ว่า "การอ้างถึงแบบไฮราคี่ที่มีสับไทป์ (Heirarchical with subtype)"

ขั้นตอนที่ 8 คอนสเตรนท์ต่อไป (Further constraint)

เราได้ผ่านเข้ามายังขั้นตอนที่ 8 ซึ่งด้วยข้อบังคับต่างๆ ซึ่งจะว่าไปแล้วกฎข้อบังคับหรือคอนสเตรนท์ เหล่านั้นก็มีเพื่อรักษาสถานะของฐานข้อมูลให้มีความถูกต้องอยู่เสมอ และแม้ว่าเราจะได้กล่าวถึงคอนสเตรนท์ที่มีความสำคัญไปหลายแบบแล้วก็ตาม แต่ก็ยังมีคอนสเตรนท์ อีกหลายแบบที่ยังต้องกล่าวถึงต่อไป

- คอนสเตรนท์แบบอีควาลิตี้ , แบบเอ็กซ์คลูชัน และ แบบสับเซต

(Equality , Exclusion and Subset constraint)

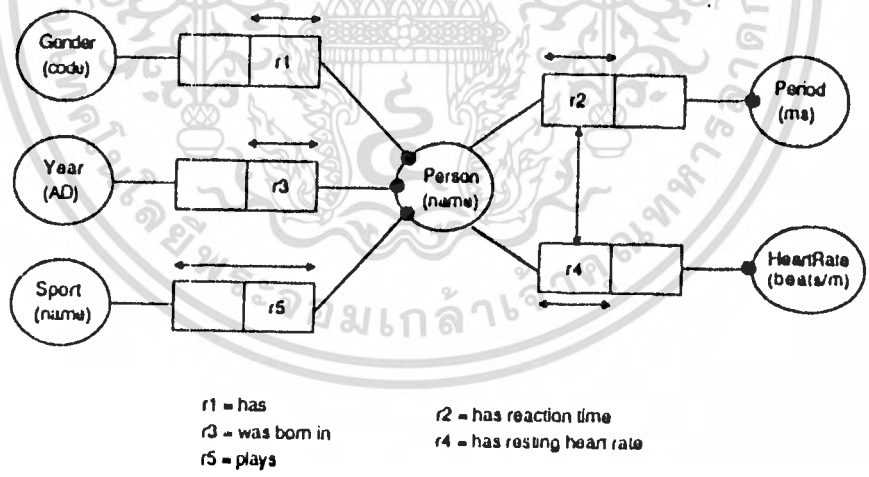
เราจะเริ่มต้นด้วยวิธีการดังเช่นที่ผ่านมาคือ เริ่มด้วยการพิจารณาข้อมูลที่เราสมมติขึ้น เรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลองมาดูตารางที่ 3.12 ซึ่งเป็นตารางข้อมูลของสมาชิกในสโมสรแห่งหนึ่ง และจากข้อมูลดังกล่าว เราสามารถที่จะสร้างแบบจำลองข้อมูลได้ดังรูปที่ 3.41

Member	*Sex*	*BirthYr*	*Sport*	*Reaction Time (ms)*	*Resting Heart Rate (beats/m)*
Anderson PE	M	1940	golf	250	80
Bloggs F	M	1940	golf	?	?
Fit LM	F	1960	aerobics tennis	250	80
Hume PE	F	1946	golf	305	93
Jones T	M	1965	tennis	?	?

ตารางที่ 3.12 แสดงตัวอย่างข้อมูลของสโมสรกีฬาแห่งหนึ่ง



รูปที่ 3.41 แสดงแบบจำลองข้อมูลของตารางที่ 3.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางจะเห็นได้ว่าข้อมูล Reaction time และ Heart Rate นั้นเป็นข้อมูลที่อาจจะใส่หรือไม่ใส่ก็ได้ (Optional) โดยเครื่องหมาย ? จะมีความหมายว่าข้อมูลนั้นไม่ได้รับการบันทึก นอกจากนั้นเราจะสังเกตเพิ่มได้อีกว่าข้อมูลทั้งสองนี้ จะต้องได้รับการบันทึกไว้ด้วยกันเสมอ จะบันทึกเฉพาะข้อมูลในตัวใดตัวหนึ่งไม่ได้ ในกรณีของข้อมูลลักษณะเช่นนี้ ไนแอมสามารถรองรับข้อจำกัดนี้ได้ โดยระบุคอนสเตรนท์แบบอิควาลิตี้ลงบนโรลทั้งสอง โดยใช้เครื่องหมายลูกศร 2 หัว ดังในรูปที่ 3.27 ซึ่งจากรูปก็จะมีความหมายว่า เซ็ตของคนที่ได้รับการบันทึกใน r2 จะต้องเท่ากับเซ็ตของคนที่ได้รับการบันทึกใน r4 หรือสามารถเขียนได้ว่า pop(r2) = pop(r4)

3.3 การสร้างเค้าร่างฐานข้อมูลแบบสัมพันธ์ โดยวิธีแปลงแบบจำลองระดับแนวคิดของไนแอม

(Create relational database schema from NIAM conceptual schema)

ในขั้นตอนก่อนหน้านี้ เราได้ศึกษาถึงวิธีการออกแบบแบบจำลองข้อมูลระดับแนวคิดมาแล้ว หากแต่ในปัจจุบันนี้ยังไม่มีระบบฐานข้อมูลใดที่สามารถใช้งานกับแบบจำลองเช่นนี้ได้ หากแต่เป็นระบบฐานข้อมูลที่เป็นแบบสัมพันธ์

ระบบฐานข้อมูลแบบสัมพันธ์ ได้รับการพัฒนาจนมีประสิทธิภาพสูง และ มีภาษา SQL (Structured Query Language) ซึ่งเป็นภาษาหลักของแบบจำลองข้อมูล (Data Model) ที่เป็นรีเลชันแนล และได้นำไปใช้งานอย่างกว้างขวางในวงการต่างๆ ซึ่งเมื่อไม่นานมานี้ภาษาSQL ก็ได้รับการรับรองให้เป็นภาษามาตรฐาน ทำให้มีการใช้งานภาษานี้ทั้งบนเครื่องเมนเฟรมมินิคอมพิวเตอร์ ไมโครคอมพิวเตอร์ ในระบบปฏิบัติการ (Operating System) ต่างๆ และนอกจากที่ได้กล่าวมาแล้วนั้น SQL ยังสามารถที่จะเชื่อมต่อกับภาษาต่างๆเช่น โคบอล หรือ ซี ได้อีกด้วย ทำให้การใช้งานคล่องตัวยิ่งขึ้น

ดังนั้นต่อไปเราก็จะนำเสนออัลกอริทึมที่ใช้ในการแปลงจาก แบบจำลองข้อมูลระดับแนวคิดของไนแอมให้เป็นเค้าร่างฐานข้อมูลแบบสัมพันธ์

นอลมอลฟอร์มที่เหมาะสม (Optimal Normal Form Algorithm : ONF)

จากที่เราได้กล่าวมาแล้ว เราได้นำตัวอย่างที่เราจะแปลงแบบจำลองข้อมูลในแบบแนวคิดไปเป็นแบบสัมพันธ์ โดยกำหนดรูปแบบของตาราง และการเขียนรูปเพื่อจัดการกับคอนสเตรนท์ต่างๆ ซึ่งต่อไปเราก็จะแนะนำวิธีการที่เป็นระบบในการแปลงแบบจำลองข้อมูลในแบบแนวคิดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นแบบสัมพันธ์

สำหรับแต่ละคอนเซ็ปชวลสกีมม่านั้น เมื่อทำการแปลงเป็น Relational Schema นั้นอาจจะแปลงได้หลายรูปแบบ แต่สำหรับอัลกอริทึม ONF นั้นมีจุดประสงค์เพื่อสร้างรูปแบบที่ง่าย ประหยัด และมีประสิทธิภาพ โดยมีหลักการต่อไปนี้ ประการแรกคือเพื่อให้ได้โครงสร้างข้อมูลที่ง่าย ไม่มีแอทริบิวต์ใดซ้ำ ประการที่สองคือป้องกันการผิดพลาดอันเนื่องมาจากการเปลี่ยนแปลงแก้ไข (Update Anomolies) ข้อมูลไม่มีการซ้ำซ้อน ประการที่ 3 เพื่อให้ได้ประสิทธิภาพ ดังนั้นจะต้องได้จำนวนตารางที่น้อยด้วย เพราะการลดจำนวนตารางก็เท่ากับการลดเวลาที่ใช้ในการเข้าถึงข้อมูล และยังทำให้การทำคิวรี่สามารถทำได้ง่ายขึ้นอีกด้วย

การแปลงเป็นตารางนี้ เมื่อเทียบกับวิธีนอร์มาไลซ์เซชัน ซึ่งเป็นวิธีการสร้างตารางเช่นกันโดยจะเริ่มจากนอร์มอลฟอร์มที่ 1 และสิ้นสุดที่นอร์มอลฟอร์มที่ 5 ในขณะที่ ONF สามารถที่จะสร้างนอร์มอลฟอร์มที่ 5 ได้เลย และสำหรับ ONF Algorithm ก็มีดังต่อไปนี้

1. สำหรับแต่ละแฟกท์ไทป์ที่ไม่ได้มีคีย์มาตรฐาน (simple key) ให้สร้างเป็นตารางแยกออกมา โดยเลือกคีย์ที่สั้นที่สุดของแฟกท์ไทป์นั้นเป็นคีย์หลัก

2. รวมแฟกท์ไทป์ที่มีคีย์มาตรฐานที่เชื่อมกับอ็อบเจ็คไทป์เดียวกันเป็นตารางเดียวกัน และให้อ็อบเจ็คไทป์นั้นเป็นคีย์หลัก

3. สำหรับแฟกท์ไทป์ที่เหลือให้สร้างเป็นตารางแยกต่างหาก

นิยาม : เอนติตี้อาจถูกอ้างถึงได้โดย

ก) เลเบลที่มีต่ออยู่กับตัวเองที่มีแฟกท์ไทป์แบบ 1:1 (one-to-one)

ข) กลุ่มของสิ่งที่ใช้อ้างถึงแบบปฐมภูมิของเอนติตี้ โดยเป็นสิ่งที่มาเกี่ยวข้องกันแบบเนสต์แฟกท์ไทป์ (nest fact type)

ค) กลุ่มของสิ่งที่ใช้อ้างถึงแบบปฐมภูมิของเอนติตี้ โดยเป็นสิ่งที่มาเกี่ยวข้องกันอันเกิดจากความสัมพันธ์กันระหว่างแฟกท์ใด ๆ ที่ถูกครอบคลุมด้วยคอนสเตรนท์แบบยูนีคเนสส์ภายนอก

ถ้าเอนติตี้ถูกอ้างถึงมากกว่า 1 แบบจากข้างบน เราจะกล่าวว่าเอนติตี้นั้นมีชื่อในนัม (synonyms)

นิยาม : สิ่งที่ใช้อ้างถึงเอนติตี้แบบปฐมภูมิ เป็นสิ่งที่ใช้อ้างถึงเพียงอย่างเดียว ถ้าไม่มีชื่อในนัม แต่ถ้ามีชื่อในนัมจะถือว่า สิ่งที่เกิดขึ้นก่อนในนิยามที่ 1 เป็นสิ่งที่ใช้อ้างถึงเอนติตี้แบบปฐมภูมิ

ขั้นตอนที่ใช้ในการแปลง (Transformation Algorithm)

ขั้นตอนการแปลงนิยามข้อมูลระดับแนวคิดของไนเอม ให้เป็นแบบแผนระบบฐานข้อมูลแบบสัมพันธ์ (

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Relational Database Schema) มี 7 ขั้นตอน ดังนี้

1.ทำการแปลงรูปไนเอม ที่มีลักษณะที่เป็นเนสติ้ง (NESTING) ให้อยู่ในรูปของเอนติตี้ที่มีหลายเอนติตี้ย่อย

2.สำหรับเอนติตี้ที่เป็นสับไทป์ สามารถจัดการได้เป็น 2 แบบกล่าวคือ

2.1 เชื่อมโยงแฟกท์ ที่ต่ออยู่กับเอนติตี้ที่เป็นสับไทป์ (subtype) ให้ไปต่ออยู่กับเอนติตี้ที่เป็นซูเปอร์ไทป์ (supertype) อันดับสูงสุด. แล้วจัดแมนดาทอรี่แฟกท์ออกจากโรลที่ถูกเชื่อมโยงนี้

2.2 ตัดความสัมพันธ์ระหว่างเอนติตี้ที่เป็นสับไทป์ กับเอนติตี้ที่เป็นซูเปอร์ไทป์ทิ้ง

3. ไม่ต้องสนใจแฟกท์ที่แสดงถึงตัวที่ใช้ในการอ้างถึงเอนติตี้ หรือแสดงทีนโนนีม เว้นแต่จะมีกล่าวไว้ในขั้นตอนนั้น

4.สร้างความสัมพันธ์ขึ้นมา 1 ความสัมพันธ์สำหรับแต่ละแฟกท์ที่มีคอนสเตรนท์แบบยูนีคเนสส์ครอบคลุมทุกโรลในแฟกท์นั้น

5.สร้างความสัมพันธ์ 1 ความสัมพันธ์ สำหรับแต่ละแฟกท์ที่มีจำนวนโรล n โรล โดยที่ n มากกว่า 2 ซึ่งมีคอนสเตรนท์แบบยูนีคเนสส์ครอบคลุมโรลจำนวน n-1 โรลในแฟกท์นั้น

6.รวมกลุ่มไบนารี่แฟกท์ที่มีคอนสเตรนท์แบบยูนีคเนสส์ครอบคลุมเพียงโรลเดียวบนเอนติตี้ร่วมกันเข้าเป็นความสัมพันธ์จำนวน 1 ความสัมพันธ์ โดยยึดเอนติตี้ร่วมกันนั้นเป็นหลักในการรวมกลุ่ม ถ้าไบนารี่แฟกท์มีคอนสเตรนท์แบบยูนีคเนสส์ครอบคลุมอยู่บนแต่ละโรล ให้ทำการรวมกลุ่มเข้าหาเอนติตี้ที่มีแมนดาทอรี่ปรากฏ อยู่บนโรลของมัน อย่างไรก็ตาม ถ้าไม่มีแมนดาทอรี่ปรากฏอยู่บนโรลใด จะทำการรวมกลุ่มเข้าหาเอนติตี้ใดก็ได้

7.สร้างความสัมพันธ์ขึ้นมา 1 ความสัมพันธ์ สำหรับเอนติตี้ที่มีชันโนนีม แต่ไม่ได้เป็นเอนติตี้ร่วมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

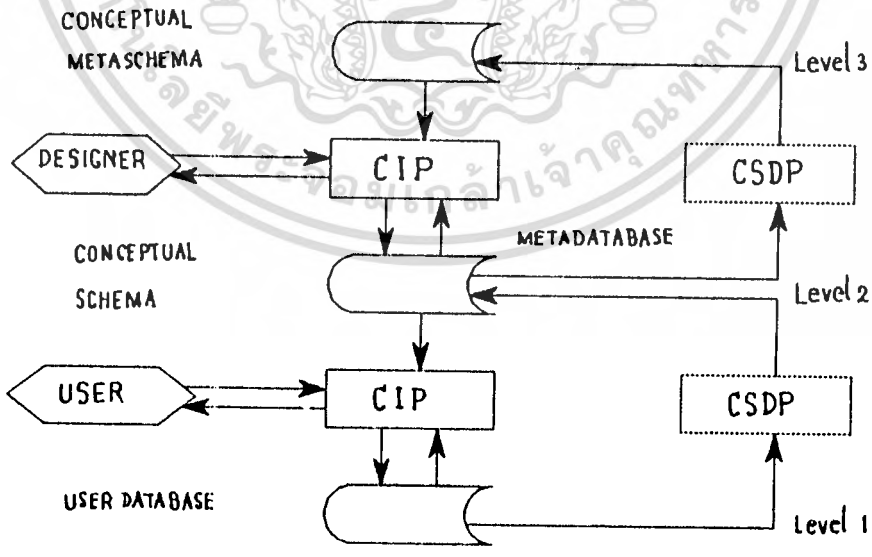
บทที่ 4

ระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์

(A Relational Database Designer Generator Part)

ในงานที่ต้องการใช้ระบบฐานข้อมูลในการเก็บข้อมูล ไม่ว่าจะเป็นฐานข้อมูลที่มีนิยามข้อมูลในระดับแนวคิดแบบใด เช่น แบบข่ายงาน (Network model) แบบลำดับชั้น (Hierarchical model) หรือ แบบสัมพันธ์ (Relational model) ต้องผ่านกระบวนการที่เรียกว่า " การออกแบบฐานข้อมูล (Database design) " ก่อนทั้งสิ้น ในงานที่อาศัยนิยามข้อมูลระดับแนวคิดแบบสัมพันธ์ สามารถออกแบบฐานข้อมูลได้โดยวิธีนอร์มาไลซ์เซชัน วิธีวาดแผนภูมิอีอาร์ หรือ แบบจำลองข้อมูลของไนแอม (NIAM)

เนื่องจากที่ได้กล่าวมาแล้วในบทนำ โครงงานระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ ประกอบไปด้วยโครงงาน 2 ส่วน และมีส่วนโครงสร้างข้อมูลที่เป็นส่วนที่ต้องเชื่อมต่อกันของโครงงานทั้ง 2 ส่วน โครงสร้างข้อมูลดังกล่าวเรียกว่า เมต้าเดต้าเบส ซึ่งหมายถึง ฐานข้อมูล ๆ หนึ่งที่เก็บข้อมูลแทนความสัมพันธ์ของนิยามข้อมูลระดับแนวคิดของไนแอม ในเต้าร่างฐานข้อมูลที่ได้มาจากเมต้าสกีมม่า รูปที่ 4.1 แสดงแผนภูมิความสัมพันธ์ระหว่าง เมต้าสกีมม่า เมต้าเดต้าเบส ผู้ใช้และแบบจำลองนิยามข้อมูลระดับแนวคิดของไนแอม หรือ คอนเซ็ปชวลสกีมม่า



รูปที่ 4.1 แผนภูมิแสดงระดับในการแทนข้อมูลในระดับแนวคิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 เมต้าสกีมม่า

จากรูปที่ 4.1 มีเจ้าหน้าที่ที่เกี่ยวข้องในระบบ 2 คน คือ นักออกแบบฐานข้อมูล (designer) และ ผู้ใช้งานฐานข้อมูล (user) ในระดับที่ 2 , 3 ผู้ออกแบบจะทำการออกแบบคอนเซ็ปชวลสกีมม่าจาก **ตัวประมวลผลข้อมูลระดับแนวคิด** (CIP : Conceptual Information Processor) ซึ่งทำหน้าที่จัดการและดูแลความเปลี่ยนแปลงของข้อมูลในฐานข้อมูล ในลักษณะตอบโต้กับผู้ใช้ ซีไอพีจะเป็นตัวกลางในการจัดการข้อมูลที่ผู้ใช้ (ในระดับนี้คือ นักออกแบบ) ต้องการเปลี่ยนแปลง โดยตรวจสอบจากกฎความถูกต้องที่กำหนดไว้ในคอนเซ็ปชวลเมต้าสกีมม่า (Conceptual metaschema) ซึ่งหมายถึง นิยามข้อมูลระดับแนวคิดที่เก็บข้อมูล กฎ และข้อกำหนดของข้อมูลของคอนเซ็ปชวลสกีมม่าเอง ข้อมูลในฐานข้อมูลที่ได้ในระดับนี้คือข้อมูลของคอนเซ็ปชวลสกีมม่าที่นักออกแบบทำการสร้างขึ้น โดยได้รับการตรวจสอบแล้วว่าถูกต้องตามข้อกำหนดของลักษณะข้อมูล ฐานข้อมูลดังกล่าวนี้เอง ที่เราเรียกว่า อภิฐานข้อมูล หรือ เมต้าเดต้าเบส

เมื่อได้เมต้าเดต้าเบสเรียบร้อยแล้ว ขั้นตอนจะผ่านมาถึงในระดับ 1 , 2 ซึ่งในระดับนี้ผู้ใช้คือผู้ใช้งานฐานข้อมูล เมต้าเดต้าเบสที่ได้จากขั้นแรกนั้น แท้ที่จริงแล้วก็คือคอนเซ็ปชวลสกีมม่าของงานแอพพลิเคชันที่ต้องการฐานข้อมูลใช้งานนั่นเอง (เพียงแต่อยู่ในรูปฐานข้อมูลมิใช่อยู่ในรูปแบบจำลองไนเอม) เช่นเดียวกับในระดับ 2,3 มีตัวประมวลผลซีไอพี ทำหน้าที่ดูแลฐานข้อมูลโดยอ้างอิงตามข้อกำหนดลักษณะข้อมูลที่กำหนดในคอนเซ็ปชวลสกีมม่า ข้อมูลที่ผู้ใช้ทำงานอยู่คือฐานข้อมูลของแอพพลิเคชัน ซึ่งใช้เก็บข้อมูลในฐานข้อมูลของระบบงานต่างๆ ตามจุดประลงค์

หากสังเกตให้ดี จะพบว่าระหว่างระดับที่ 1 กับ 2 และ 2 กับ 3 จะมีกระบวนการออกแบบคอนเซ็ปชวลสกีมม่า (Conceptual Schema Design Procedure) ซึ่งได้กล่าวไว้ในทฤษฎีและหลักการเบื้องต้นบทที่ 2

4.2 เอนติตี้ในไนแอมคอนเซ็ปชวลเมต้าสกีมม่า

(entity in NIAM - Conceptual metaschema)

เนื่องจากในโครงงานนี้ อ้างอิงแบบจำลองข้อมูลไนแอม เป็นนิยามข้อมูลระดับแนวคิดในการออกแบบ เมต้าสกีมม่าที่ใช้ในโครงงานจึงเป็นแบบจำลองระดับแนวคิดของไนแอมด้วย เอนติตี้ไทป์ของเมต้าสกีมม่ามีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. OBJECT TYPE เซตของอ็อบเจ็คไทป์ของไนแอม ใช้ OBJ_NAME ในการอ้างถึง

2. LABEL TYPE เซตของเลเบิ้ลไทป์ของไนแอมซึ่งเป็นสับเซตของอ็อบเจ็คไทป์ ใช้ OBJ_NAME ในการอ้างถึง

3. ENTITY TYPE เซตของเอนติตี้ไทป์ของไนแอมซึ่งเป็นสับเซตของอ็อบเจ็คไทป์ ใช้ OBJ_NAME ในการอ้างถึง

4. KIND_OF_OBJ เซตของประเภทของอ็อบเจ็คของไนแอม ใช้ชี้เฉพาะว่าอ็อบเจ็คที่พิจารณาเป็นเลเบิ้ลหรือเอนติตี้ไทป์ ใช้ KIND ในการอ้างถึง

5. LABEL_ORDER เซตของลำดับของเลเบิ้ลไทป์ ในกรณีที่เอนติตี้ไทป์หนึ่งมีความสัมพันธ์กับเลเบิ้ลไทป์มากกว่าหนึ่งตัว ใช้ ORDER ในการอ้างถึง

6. ROLE เซตของโรลของไนแอม ใช้ ROLE_NAME ในการอ้างถึง

7. ROLE TYPE เซตของชนิดของโรลที่สนใจของไนแอม ใช้ชี้เฉพาะว่าเป็นโรลแบบใด ได้แก่ โรลแบบปกติ และ โรลที่สัมพันธ์กับเนสติ้ง ใช้ TYPE ในการอ้างถึง

8. ROLE ORDER เซตของลำดับของโรลของไนแอม ใช้เพื่อบอกลำดับของโรลในแต่ละแฟกท์ไทป์ หรือ เรฟเฟอร์เรนซ์ไทป์ ใช้ ORDER ในการอ้างถึง

9. NORMAL ROLE เซตของโรลแบบปกติของไนแอม ซึ่งหมายถึงโรลที่สัมพันธ์กับอ็อบเจ็คของไนแอม เป็นสับเซตของ ROLE ใช้ ROLE_NAME ในการอ้างถึง

10. ROLE GROUP เซตของกลุ่มโรลที่ใช้ในการบ่งกลุ่มของโรลที่มีความสัมพันธ์กับคอนสเตรนท์ของไนแอม ใช้ R_GROUP ในการอ้างถึง

11. NESTING ROLE เซตของโรลแบบเนสติ้งของไนแอม ซึ่งหมายถึงโรลที่สัมพันธ์กับเนสติ้งของไนแอม เป็นสับเซตของ ROLE ใช้ ROLE_NAME ในการอ้างถึง

12. RELATIONSHIP TYPE เซตของรีเลชันที่กำหนดในไนแอม มีรีเลชันอยู่ 2 ประเภทหลักคือ แฟกท์ไทป์ และ เรฟเฟอร์เรนซ์ไทป์ ใช้ REL_NAME ในการอ้างถึง

13. FACT TYPE เซตของแฟกท์ไทป์ของไนแอมซึ่งเป็นสับเซตของ RELATIONSHIP TYPE ประกอบด้วยแฟกท์ไทป์ 2 ประเภทคือ แฟกท์ไทป์แบบปกติหรือ นอร์มอลแฟกท์ไทป์ (Normal fact type) และแฟกท์ไทป์แบบหาได้ หรือ ดีไรฟ์วแฟกท์ไทป์ (Derived fact type) ใช้ REL_NAME ในการอ้างถึง

14. NORMAL FACT TYPE เซตของแฟกท์ไทป์แบบปกติของไนแอม ซึ่งเป็นสับเซตของแฟกท์ไทป์ FACT TYPE หมายถึงแฟกท์ไทป์ซึ่งใช้แทนข้อมูลของความสัมพันธ์ที่เก็บในฐานข้อมูลจริงในรูปของตาราง (TABLE) ใช้ REL_NAME ในการอ้างถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

15. DERIVED FACT TYPE เซตของแฟกท์ไทป์แบบหาได้ หรือ ดีไรฟ์วแฟกท์ไทป์ เป็นสับเซตของ FACT TYPE ชนิดที่ข้อมูลของความสัมพันธ์ในฐานข้อมูล ถูกเก็บให้อยู่ในรูปที่ได้มาจากการคำนวณค่าข้อมูลของความสัมพันธ์อื่น ๆ เช่น ดีไรฟ์วแฟกท์ไทป์ AREA (พื้นที่) ได้มาจากการนำเอาแฟกท์ไทป์ WIDTH (กว้าง) และ LENGTH (ยาว) มาคูณกัน เก็บดีไรฟ์วแฟกท์ไทป์ AREA นี้ในรูปของวิว (VIEW) ในฐานข้อมูล ใช้ RELNAME ในการอ้างถึง

16. DERIVED FORMULA เซตของสูตรที่ใช้ในการคำนวณค่าข้อมูลในความสัมพันธ์ที่เป็นแฟกท์ไทป์แบบหาได้ เก็บในลักษณะสตริง สามารถเก็บได้ทั้งสูตรการคำนวณและตัวโปรแกรมภาษา SQL ที่ใช้ในการสร้างวิวฐานข้อมูล ใช้ FORM ในการอ้างถึง

17. KIND_OF_REL เซตของประเภทของรีเลชัน ได้แก่ เรฟเฟอร์เรนซ์ไทป์ (REF) แฟกท์ไทป์แบบหาได้ (F_DERIVED) แฟกท์ไทป์แบบปกติ (F_NORMAL) ใช้ KIND ในการอ้างถึง

18. CONSTRAINT เซตหรือข้อจำกัดหรือคอนสเตรนท์ของไนแอม โดยใช้ CONST# ในการอ้างถึง

19. KIND_OF_CONST เซตของประเภทคอนสเตรนท์ของไนแอม ได้แก่ คอนสเตรนท์แบบยูนีคเนสส์ (UNIQUENESS) แบบแมนดาทอรี่ (MANDATORY) แบบเอ็กซ์คลูชัน(EXCLUSION) แบบสับเซต (SUBSET) แบบอีควาลิตี้ (EQUALITY) แบบเมมเบอร์ชิพ (MEMBERSHIP) แบบเรนจ์ (RANGE) แบบออคเคอเรนซ์ ฟรีเควนซี่ (OCCURRECNCE FREQUENCY) ใช้ KIND ในการอ้างถึง

20. SUBSET CONST เซตของคอนสเตรนท์แบบสับเซตของไนแอม เป็นสับเซตของ CONSTRAINT เนื่องจากเป็นประเภทหนึ่งของคอนสเตรนท์ จึงใช้การอ้างถึงเช่นเดียวกันคือ CONST#

21. OBJECT TYPE CONST เซตของคอนสเตรนท์แบบอ็อบเจ็คไทป์ของไนแอม เป็นสับเซตของ CONSTRAINT เนื่องจากเป็นประเภทหนึ่งของคอนสเตรนท์ที่เกี่ยวข้องอยู่กับอ็อบเจ็คในไนแอม จึงใช้การอ้างถึงเช่นเดียวกันคือ CONST#

22. OCCURRENCE FREQUENCY เซตของคอนสเตรนท์แบบออคเคอเรนซ์ ฟรีเควนซี่ของไนแอม เป็นสับเซตของ CONSTRAINT เนื่องจากเป็นประเภทหนึ่งของคอนสเตรนท์ที่เกี่ยวข้องอยู่กับอ็อบเจ็คในไนแอม จึงใช้การอ้างถึงเช่นเดียวกันคือ CONST#

23. FREQUENCY เซตของค่าความถี่ที่กำหนดในคอนสเตรนท์แบบออคเคอเรนซ์ฟรีเควนซี่ มีทั้งค่าความถี่เริ่มต้น และความถี่สุดท้ายของข้อมูล ใช้ TIME ในการอ้างถึง

24. MEMBERSHIP CONST เซตของคอนสเตรนท์แบบเมมเบอร์ชิพของไนแอม ซึ่งเป็นสับเซตของคอนสเตรนท์แบบอ็อบเจ็คไทป์ เนื่องจากเป็นประเภทหนึ่งของคอนสเตรนท์ที่เกี่ยวข้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อยู่กับอ็อบเจ็คในไนแอม จึงใช้การอ้างถึงเช่นเดียวกับคอนสเตรนท์แบบอ็อบเจ็คไทป์คือ CONST#

25. MEMBER เซตของสมาชิกที่กำหนดอยู่ในคอนสเตรนท์แบบเมมเบอร์ชิพใช้ MEMBER# ในการอ้างถึง

26. RANGE CONST เซตของคอนสเตรนท์แบบเรนจ์ของไนแอม ซึ่งเป็นสับเซตของคอนสเตรนท์แบบอ็อบเจ็คไทป์ เนื่องจากเป็นประเภทหนึ่งของคอนสเตรนท์ที่เกี่ยวข้องอยู่กับอ็อบเจ็คในไนแอม จึงใช้การอ้างถึงเช่นเดียวกับคอนสเตรนท์แบบอ็อบเจ็คไทป์และเช่นเดียวกับคอนสเตรนท์แบบเมมเบอร์ชิพ คือ CONST#

27. LIMIT เซตของค่าจำกัดที่กำหนดอยู่ในคอนสเตรนท์แบบเรนจ์ ค่าจำกัดนี้มีได้ทั้งค่าจำกัดบน และค่าจำกัดล่าง ใช้ LIM ในการอ้างถึง

4.3 ความสัมพันธ์ระหว่างเอนติตี้ไทป์ในไนแอมคอนเซ็ปชวลเมต้าสกีมม่า

(relationship between objects in NIAM - Conceptual metaschema)

4.3.1 ความสัมพันธ์แบบสับเซต

มีอยู่ 4 กลุ่มได้แก่

1. OBJECT TYPE ในไนแอมแบ่งเป็น 2 ประเภทอันได้แก่ LABEL TYPE และ ENTITY TYPE เอนติตี้ทั้งสองจึงเป็นสับไทป์ของ OBJECT TYPE

2. ROLE ในไนแอมแบ่งเป็น 2 ประเภทอันได้แก่ NORMAL ROLE และ NESTING ROLE เอนติตี้ทั้งสองจึงเป็นสับไทป์ของ ROLE

3. RELATIONSHIP TYPE ในไนแอมแบ่งเป็น 2 ประเภทได้แก่ FACT TYPE และ REFERENCE TYPE ในส่วนของเอนติตี้ FACT TYPE จะปรากฏในรูปเมต้าสกีมม่า แต่สำหรับ REFERENCE TYPE จะอยู่ในส่วนแมปปิ้งอินฟอร์เมชัน (Mapping information) ซึ่งจะกล่าวต่อไป ในเอนติตี้ FACT TYPE จะมีแบ่งประเภทเป็นสับไทป์ย่อยลงไปอีกคือ NORMAL FACT TYPE และ DERIVED FACT TYPE

4. CONSTRAINT ในไนแอมแบ่งเป็น 8 ประเภท หากแต่มีเพียง 4 ประเภทที่ปรากฏในรูปไนแอมเมต้าสกีมม่า เนื่องจากคอนสเตรนท์ตัวอื่น ๆ ไม่มีข้อมูลบ่งพิเศษ จะมีความสัมพันธ์ก็แต่เพียงกับ ROLE GROUP เท่านั้น ตัวอย่างคอนสเตรนท์เหล่านี้ เช่น แบบยูนีค แบบแมนดาทอรี่ แบบเอ็กซ์คลูชัน เป็นต้น จากรูปจะมีสับไทป์ 3 สับไทป์ อันได้แก่ SUBSET CONST , OBJECT TYPE CONST และ OCCURENCE FREQ โดยที่เอนติตี้ OBJECT TYPE CONST ยังแบ่งประ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เภทย่อยออกเป็นอีก 2 สับไทป์ ซึ่งได้แก่ MEMBERSHIP CONST และ RANGE CONST

4.3.2. ความสัมพันธ์แบบแฟกท์ไทป์

แบ่งการอธิบายตามกลุ่มสับเซตทั้ง 4 กลุ่ม ดังนี้

1. กลุ่ม OBJECT TYPE

- มีความสัมพันธ์กับ NORMAL ROLE เนื่องจากโรลแบบปกติ (ไม่ใช่โรลแบบเนสติ้ง) ในไนแอมต้องต่ออยู่กับอ็อบเจ็คไทป์ตัวใดตัวหนึ่งเสมอ จะอยู่ลอย ๆ ไม่ได้

- มีความสัมพันธ์กับ KINDJOFJOBJ เพื่อชี้บ่งว่าอ็อบเจ็คไทป์ในไนแอมตัวใด ๆ เป็นอ็อบเจ็คไทป์ประเภทเอนติตี้ไทป์ หรือ เลเบิ้ลไทป์ เรียกเอนติตี้ไทป์อย่าง KIND_OF_OBJ นี้ว่า "ตัวบ่งชนิด" หรือ category

- สับไทป์ ENTITY TYPE มีความสัมพันธ์กับตัวเอง กล่าวคือ ความสัมพันธ์ดังกล่าวนี้เกิดขึ้นในกรณีที่มีการวาดสับไทป์ในไนแอม แฟกท์ไทป์นี้จะเก็บข้อมูลของชื่อเอนติตี้ไทป์ที่เป็นซูเปอร์ไทป์ และชื่อของเอนติตี้ที่เป็นสับไทป์

- สับไทป์ LABEL TYPE มีความสัมพันธ์กับ LABEL ORDER เพื่อใช้บ่งว่าเลเบิ้ลไทป์ตัวใด ๆ ในไนแอมมีลำดับเท่าไรในการอ้างถึง ของเอนติตี้ที่เกี่ยวข้องด้วย เกิดในกรณีที่มีการวาดเอนติตี้ไทป์ในไนแอมที่มีการอ้างถึง (reference) เลเบิ้ลไทป์มากกว่าหนึ่ง หรือกรณีที่มีการอ้างถึงแบบไฮราคี่ (ดังที่กล่าวไว้ในบทที่ 2 ขั้นตอนที่ 7 ของหัวข้อ " ขั้นตอนการออกแบบแบบจำลองข้อมูลไนแอม ")

2. กลุ่ม ROLE

- มีความสัมพันธ์กับเอนติตี้ไทป์ ROLE TYPE เพื่อเป็นตัวบ่งชนิดว่าโรลในไนแอมตัวใด ๆ เป็นโรลประเภท โรลปกติ หรือ โรลแบบเนสติ้ง (ซึ่งหมายถึงโรลที่ต่ออยู่กับเนสติ้ง)

- มีความสัมพันธ์กับ ROLE ORDER เพื่อเก็บหมายเลขกำหนดลำดับของโรลภายในแฟกท์ไทป์ หรือ เรฟเฟอร์เรนซ์ไทป์ ของไนแอม

- มีความสัมพันธ์กับ ROLE GROUP เนื่องจากจากการกำหนดคอนสเตรนท์ในไนแอมจะกระทำบนโรลในรีเลชันชิพไทป์ (แฟกท์ไทป์ และ เรฟเฟอร์ไทป์) ทั้งในลักษณะของการใช้โรลมากกว่า 1 โรล หรือการผนวกโรลเข้าไว้เป็นกลุ่ม เช่น การกำหนดคอนสเตรนท์แบบยูนีคภายนอก ซึ่งต้องอาศัยกลุ่มโรล 2 กลุ่มในการกำหนด แต่ละกลุ่มอาจมีเพียง 1 หรือมากกว่า 1 โรลก็ได้ ความสัมพันธ์อันนี้จึงหมายถึง การกำหนดว่าโรลใดเป็นสมาชิกในกลุ่มโรลใดในไนแอม (หากไม่มีการกำหนดคอนสเตรนท์บนโรล โรลดังกล่าวก็ไม่จำเป็นต้องมีกลุ่ม)

- มีความสัมพันธ์กับ RELATIONSHIP TYPE เนื่องจากในแต่ละรีเลชันชิพไทป์ประกอบด้วยโรลต่างๆ ดังนั้นทุกโรลที่ปรากฏในไนแอมจะต้องมีการบันทึกที่อยู่ในรีเลชันใดรีเล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขันหนึ่งเสมอ

- สับไทป์ NORMAL ROLE มีความสัมพันธ์กับ OBJECT TYPE เพื่อเป็นการบันทึกว่าโรลปกติใดต่ออยู่กับอ็อบเจ็คใดในรูปไนแอม

- สับไทป์ NESTING ROLE มีความสัมพันธ์กับเอนติ้ไทป์ NORMAL FACT TYPE เนื่องจากโรลแบบเนสติ้งต้องต่ออยู่กับเนสติ้ง ซึ่งที่จริงแล้วโดยตัวเนสติ้งเองก็มีฐานะเป็นแฟกท์ไทป์ปกติตัวหนึ่งในรูปไนแอม จึงต้องบันทึกโรลเนสติ้งคู่ไปกับแฟกท์ปกติที่เป็นเนสติ้ง

3. กลุ่ม RELATIONSHIP TYPE

- มีความสัมพันธ์กับ ROLE ในลักษณะที่บอกว่าโรลชื่อใด อยู่ภายในรีเลชันใดในรูปไนแอม

- มีตัวบ่งชนิดคือ KIND_OF_REL ซึ่งจะบ่งชนิดของรีเลชันชิพไทป์ ว่าเป็นแฟกท์หรือเรฟเฟอร์เรนซ์ไทป์ หากเป็นแฟกท์จะเป็นประเภทหาได้หรือประเภทปกติ

- สับไทป์ DERIVED FACT TYPE ซึ่งเป็นสับไทป์ของ FACT TYPE อีกทีหนึ่ง มีความสัมพันธ์กับ DERIVED FORMULA ซึ่งเป็นสูตรที่ใช้ในการหาค่าข้อมูลของแฟกท์ไทป์แบบหาได้ โดยจะเก็บเป็นภาษาสร้างวิวของ SQL หรือ เก็บในลักษณะสูตรคำนวณก็ได้

4. กลุ่ม CONSTRAINT

- มีตัวบ่งชนิดคือ KIND_OF_CONST เพื่อบ่งว่าคอนสเตรนท์ในรูปไนแอมเป็นคอนสเตรนท์ชนิดใด

- มีความสัมพันธ์กับ ROLE GROUP เนื่องจากคอนสเตรนท์เป็นข้อจำกัดที่กระทำอยู่บนโรล ซึ่งเป็นสมาชิกอยู่ในกลุ่มโรล คอนสเตรนท์จึงต้องสัมพันธ์กับกลุ่มโรล แต่ก็ยังมีคอนสเตรนท์บางประเภทที่ไม่ต้องยึดอยู่บนโรล ได้แก่คอนสเตรนท์ในกลุ่มของอ็อบเจ็คไทป์คอนสเตรนท์ที่ยึดอยู่กับอ็อบเจ็คในไนแอมแทน

- สับไทป์ OBJECT TYPE CONST เป็นคอนสเตรนท์ประเภทที่ยึดอยู่กับอ็อบเจ็คในไนแอมเป็นหลัก จึงมีความสัมพันธ์กับ OBJECT_TYPE แทนที่จะสัมพันธ์กับ ROLE GROUP เช่นคอนสเตรนท์ตัวคื่น ๆ

- สับไทป์ OCCURRENCE FREQ เป็นคอนสเตรนท์ที่ต้องมีการกำหนดค่าความถี่บนและค่าความถี่ล่างคือ FREQUENCY

- สับไทป์ MEMBERSHIP CONST เป็นคอนสเตรนท์ที่ต้องมีการประกาศสมาชิกที่จะใช้ จึงมีความสัมพันธ์กับ MEMBER ซึ่งเก็บสมาชิกที่ใช้ในคอนสเตรนท์แบบเมมเบอร์ชิพ

- สับไทป์ RANGE CONST เป็นคอนสเตรนท์ที่ต้องมีการกำหนดค่าลิมิตสูงต่ำ จึงมีความสัมพันธ์กับ LIMIT ที่เก็บค่าลิมิต 2 ค่า คือค่าลิมิตบนและลิมิตล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สับไทป์ SUBSET CONST เป็นคอนสเตรนท์ที่ต้องทำการเก็บกลุ่มโรล ที่เป็นกลุ่มโรลที่มีสมาชิกน้อยกว่าอีกกลุ่มโรลหนึ่ง เรียกกลุ่มโรลนี้ว่า กลุ่มโรลที่เป็นสับเซต จากรูปเมต้าสกีมม่าจะสังเกตเห็นว่ามีความสัมพันธ์อยู่กับ ROLE GROUP ถึง 2 ความสัมพันธ์ โดยที่ความสัมพันธ์หนึ่งเก็บในลักษณะแฟกท์ไทป์ปกติ ได้แก่ความสัมพันธ์ที่ใช้บอกกลุ่มโรลที่เป็นสับเซตกับอีกหนึ่งความสัมพันธ์ที่เป็นแฟกท์ไทป์แบบหาได้ มีสมาชิกคือกลุ่มโรลที่เป็นซูเปอร์เซต เนื่องด้วยเป็นความสัมพันธ์ที่เป็นแฟกท์ไทป์แบบหาได้นี้ จึงต้องมีการกำหนดสูตรที่ใช้ในการหาค่าข้อมูล โดยกรณีนี้จะเป็นภาษา SQL ที่ใช้สร้างวิวในฐานข้อมูล มีความว่า

```
CREATE VIEW SUPERSET_CONST ( CONST# , ROLE_GROUP ) AS
SELECT     T1.CONST# , T1.ROLE_GROUP
FROM       CONST_ROLE T1 , SUBSET_CONST T2
WHERE      T1.CONST# = T2.CONST#
           AND  T1.ROLE_GROUP <> T2.ROLE_GROUP ;
```

ซึ่งหมายถึง ให้หาค่าข้อมูลจากการเชื่อม (join) ระหว่าง 2 ตาราง คือตาราง CONST_ROLE ที่เก็บความสัมพันธ์ระหว่าง CONSTRAINT และ ROLE GROUP และตาราง SUBSET_CONST ที่เก็บความสัมพันธ์ระหว่าง CONSTRAINT และ ROLE GROUP ที่เป็นสับเซต โดยมีเงื่อนไขว่า ค่าข้อมูลที่จะอยู่ในความสัมพันธ์นี้ จะต้องอยู่ในตาราง CONST_ROLE แต่ไม่อยู่ใน SUBSET_CONST

4.4 เมต้าสกีมม่าในลักษณะตารางรีเลชันบนฐานข้อมูลแบบสัมพันธ์คอราเคิล (Relational metaschema on ORACLE RDBMS)

จากไนแอมคอนเซปชวลเมต้าสกีมม่า ที่ได้อธิบายในหัวข้อที่ผ่านมา ได้นำมาทำการแปลงตามอัลกอริทึมการแปลงจากคอนเซ็บชวลสกีมม่า เป็นเค้าร่างตารางฐานข้อมูลแบบสัมพันธ์ ได้ทั้งสิ้น 15 ตาราง กับอีก 1 วิว ทำการสร้างเมต้าสกีมม่าในรูปตารางฐานข้อมูลเหล่านี้ลงบนฐานข้อมูลที่ดูแลโดยระบบจัดการฐานข้อมูล ORACLE ซึ่งใช้เป็นระบบจัดการฐานข้อมูลที่โครงงานนี้อ้างถึง (ดูรายละเอียดในภาคผนวก) เมต้าสกีมม่าเหล่านี้แสดงดังรูปข้างล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(1) TABLE OBJECT_TYPE

OBJECT_NAME	KIND_OF_OBJECT
CHAR(20)	CHAR(10)

(2) TABLE LABEL_TYPE

OBJECT_NAME	TYPE_NAME	LABEL_ORDER
CHAR(20)	CHAR(20)	CHAR(5)

(3) TABLE SUBTYPE

SUB_NAME	SUPER_NAME
CHAR(20)	CHAR(20)

(4) TABLE ROLE

ROLE_NAME	ROLE_TYPE	RELATION_NAME	ROLE_ORDER
CHAR(20)	CHAR(10)	CHAR(10)	CHAR(1)

(5) TABLE OBJECT_ROLE

ROLE_NAME	OBJECT_NAME
CHAR(20)	CHAR(20)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(6) TABLE NESTING

ROLE_NAME	RELATION_NAME
CHAR(20)	CHAR(10)

(7) TABLE RELATION_TYPE

RELATION_NAME	KIND_OF_RELATION
CHAR(10)	CHAR(10)

(8) TABLE ROLE_GROUP

ROLE_NAME	ROLE_GROUP
CHAR(20)	CHAR(10)

(9) TABLE CONSTRAINT

CONST#	KIND_OF_CONST	CONST_NOTE
CHAR(10)	CHAR(10)	CHAR(1)

(10) TABLE CONST_ROLE

CONST#	ROLE_GROUP
CHAR(10)	CHAR(10)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(11) TABLE OBJECT_CONST

CONST#	OBJECT_NAME
CHAR(10)	CHAR(20)

(12) TABLE RANGE_CONST

CONST#	MIN_NUMBER	MAX_NUMBER
CHAR(10)	CHAR(5)	CHAR(5)

(13) TABLE MEMBERSHIP_CONST

CONST#	MEMBERCHAR(10)
CHAR(10)	CHAR(10)

(14) TABLE SUBSET_CONST

CONST#	ROLE_GROUP
CHAR(10)	CHAR(10)

(15) VIEW SUPERSET_CONST

CONST#	ROLE_GROUP
CHAR(10)	CHAR(10)

```
CREATE VIEW SUPERSET_CONST AS
SELECT  T1.CONST# , T1.ROLE_GROUP
FROM  CONST_ROLE T1 , SUBSET_CONST T2
WHERE  T1.CONST# = T2.CONST#
AND  T2.ROLE_GROUP <> T1.ROLE_GROUP ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(16) TABLE FREQ]CONST

CONST# CHAR(10)	UPPER_LIMIT CHAR(5)	LOWER_LIMIT CHAR(5)

ตาราง 4.1 แสดงตารางเมต้าสกีมม่า

4.5 แมปปิ้งอินฟอร์เมชั่น (Mapping Information)

ในกระบวนการแมปปิ้งคอนเซ็ปชวลสกีมม่าให้เป็นตารางฐานข้อมูลรีเลชันแนล โดยอาศัยโมดูลการสร้างเค้าร่างฐานข้อมูล (REDD-G : ซึ่งเป็นโมดูลที่พัฒนาขึ้นโดยกลุ่มโครงงานระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ ส่วนโครงสร้างข้อมูล) นั้น จำเป็นต้องมีข้อมูลส่วนหนึ่งไว้ใช้ในกระบวนการแมปปิ้ง และรีเลชันที่ใช้เก็บข้อมูลของตารางฐานข้อมูลที่ได้จากกระบวนการเช่น ชื่อตารางรีเลชันที่ได้ ชื่อวิว เป็นต้น การออกแบบตารางรีเลชันดังกล่าวนี้ ใช้แบบจำลองไนแอมเช่นเดียวกันเรียกคอนเซ็ปชวลสกีมม่านี้ว่า **แมปปิ้งอินฟอร์เมชั่น (Mapping information)** ประกอบด้วย 2 ส่วน คือ 1. ส่วนเก็บการอ้างอิงของเอนติตี้ในแบบจำลองไนแอม และ 2. ส่วนเก็บชื่อของตารางรีเลชันที่ได้จากกระบวนการแมปปิ้ง

4.5.1 เอนติตี้ไทป์ที่เพิ่มเข้ามาในส่วนแมปปิ้งอินฟอร์เมชั่น

1. COLUMN NAME ใช้ในการเก็บชื่อคอลัมน์ของรีเลชันที่ได้จากกระบวนการแมปปิ้ง มีความสัมพันธ์กับ ENTITY TYPE / RELATIONSHIP TYPE ROLE และ LABEL TYPE ในลักษณะของเนสติ้ง เป็นความสัมพันธ์ที่ต้องเกิดขึ้นทุกครั้งที่มีการสร้างไนแอมเพื่อแมปปิ้งเป็นรีเลชัน

2. DATA TYPE ใช้ในการเก็บชนิดข้อมูลของแต่ละแอตตริบิวท์ที่อยู่ในรีเลชันที่ได้จากกระบวนการ ชนิดข้อมูลดังกล่าวนี้ผู้ใช้งานระบบ (นักออกแบบ) จะต้องกำหนดไว้ตั้งแต่เริ่มวาดแบบจำลองไนแอม มิฉะนั้นจะไม่สามารถสร้างรีเลชันได้ ชนิดข้อมูลนี้จะมีความสัมพันธ์กับเลเบ้ลไทป์ LABEL TYPE เนื่องจากเลเบ้ลไทป์เป็นการอ้างถึงของเอนติตี้ในระบบจึงถือเป็นคุณสมบัติที่เอนติตี้ต่าง ๆ พึงจะมี

3. TABLE NAME ใช้ในการเก็บชื่อของรีเลชันที่ได้จากกระบวนการแมปปิ้ง มีความสัมพันธ์กับ REFERENCE TYPE และ NORMAL FACT TYPE เนื่องจากในกระบวนการแมปปิ้งจะอา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ศัพรีเลชันชิพไทป์ทั้งสองเป็นหลักในการสร้างความสัมพันธ์ที่เป็นรีเลชัน นอกจากนี้ยังมีความสัมพันธ์กับ PRIMARY KEY ซึ่งจะกล่าวต่อไป

4. PRIMARY KEY ใช้ในการเก็บชื่อแอตตริบิวท์ที่เป็นไพรมารี่คีย์ของรีเลชันที่ได้ เพื่อบ่งบอกคุณสมบัติของรีเลชันนั้น ๆ

5. VIEW NAME ใช้ในการเก็บชื่อวิวที่ได้จากกระบวนการสร้างวิวจากแฟกท์ไทป์แบบหาได้ของไนแอม

4.6 แมปปิ้งอินฟอร์เมชันในลักษณะตารางรีเลชันบนฐานข้อมูลเบสมันธ์ออราเคิล

(Mapping Information in relational form on ORACLE RDBMS)

(1) TABLE UNIQ]REF

ENTITY_NAME CHAR(20)	RELATION_NAME CHAR(20)	ROLE_NAME CHAR(20)	LABEL_NAME CHAR(20)	COL_NAME CHAR(20)

(2) TABLE NONUNIQ]REF

ENTITY_NAME CHAR(20)	RELATION_NAME CHAR(20)	LABEL_NAME CHAR(20)	COL_NAME CHAR(20)

(3) TABLE NORMAL]FACTTYPE

RELATION_NAME CHAR(20)	TABLE_NAME CHAR(20)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(4) TABLE REFERENCE

RELATION_NAME CHAR(20)	TABLE_NAME CHAR(20)

(5) TABLE PRIMARY_KEY

TABLE_NAME CHAR(20)	COL_NAME CHAR(20)

(6) TABLE DERIVED_FACTTYPE

RELATION_NAME CHAR(10)	VIEW NAME CHAR(20)	DERIVED_DEF CHAR(100)

ตาราง 4.2 แสดงตารางของเมปปิ้งอินฟอร์เมชัน

4.7 อัลกอริทึมละเอียดของแอตทริบิวท์ต่าง ๆ ในตารางเมต้าสกีมม่าและแมปปิ้งอินฟอร์เมชัน

ชื่อแอตทริบิวท์	จำนวนอักขระ	ความหมาย
OBJECT_NAME	20	เป็นตัวเก็บชื่อของ ENTITY หรือ LABEL
KIND_OF_OBJECT	10	จะเก็บชนิดของ OBJECT_NAME ("LABEL","ENTITY")
TYPE_NAME	20	ที่ให้เห็นว่า OBJECT_NAME นี้ เมื่อ Map

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อแอตตริบิวท์	จำนวนอักขระ	ความหมาย
TYPE_NAME (ต่อ)		เป็นตารางแล้ว จะเก็บลักษณะข้อมูลเป็นอย่างไร เช่น C10 จำนวนอักขระ 10 ตัวอักษร, N10 ค่าตัวเลข ขนาด 10 หลัก เป็นต้น
LABEL_ORDER	5	เป็นตัวกำหนดลำดับค่าตัวเลขให้กับ LABEL ต่อ 1 ENTITY (1,2,3,4,5,...)
SUB_NAME	20	ชื่อ Subtype ของ Entity
SUPER_NAME	20	ชื่อ Supertype ของ Entity
ROLE_NAME	20	ชื่อของ ROLE
ROLE_TYPE	10	ชนิดของ ROLE ("NORMAL","NESTING")
ROLE_ORDER	1	กำหนดลำดับของ ROLE (1,2,3,4,....,9)
RELATION_NAME	10	ชื่อของความสัมพันธ์ (จะเก็บชื่อของ FACTTYPE และ ชื่อของ REFERENCETYPE)
KIND_OF_RELATION	10	เป็นชนิดของความสัมพันธ์ ("FTNORMAL", "FTDERIVED","REF")
ROLE_GROUP	10	กลุ่มของ Constraint
CONST#	10	ID ของ Constraint (C1,C2,....,Cn)
KIND_OF_CONST#	10	ชนิดของ Constraint โดยจะมีอยู่ 7 ชนิดดังนี้ (UNIQUE,EXCLUSIV,EQUAL,SUBSET,RANGE, FREQ, MEMBER,MAND)
CONST_NOTE	1	เป็น Flag แสดงว่าจะให้มีการเช็คConstraints หรือ ไม่ (N ไม่เช็ค , Y เช็ค)
MIN_NUMBER	5	ความถี่ล่างที่ใช้ในการกำหนด Occurence freq เก็บเป็นค่าจำนวนเต็ม
MAX_NUMBER	5	ความถี่บนที่ใช้ในการกำหนด Occurence freq. เก็บค่าจำนวนเต็ม

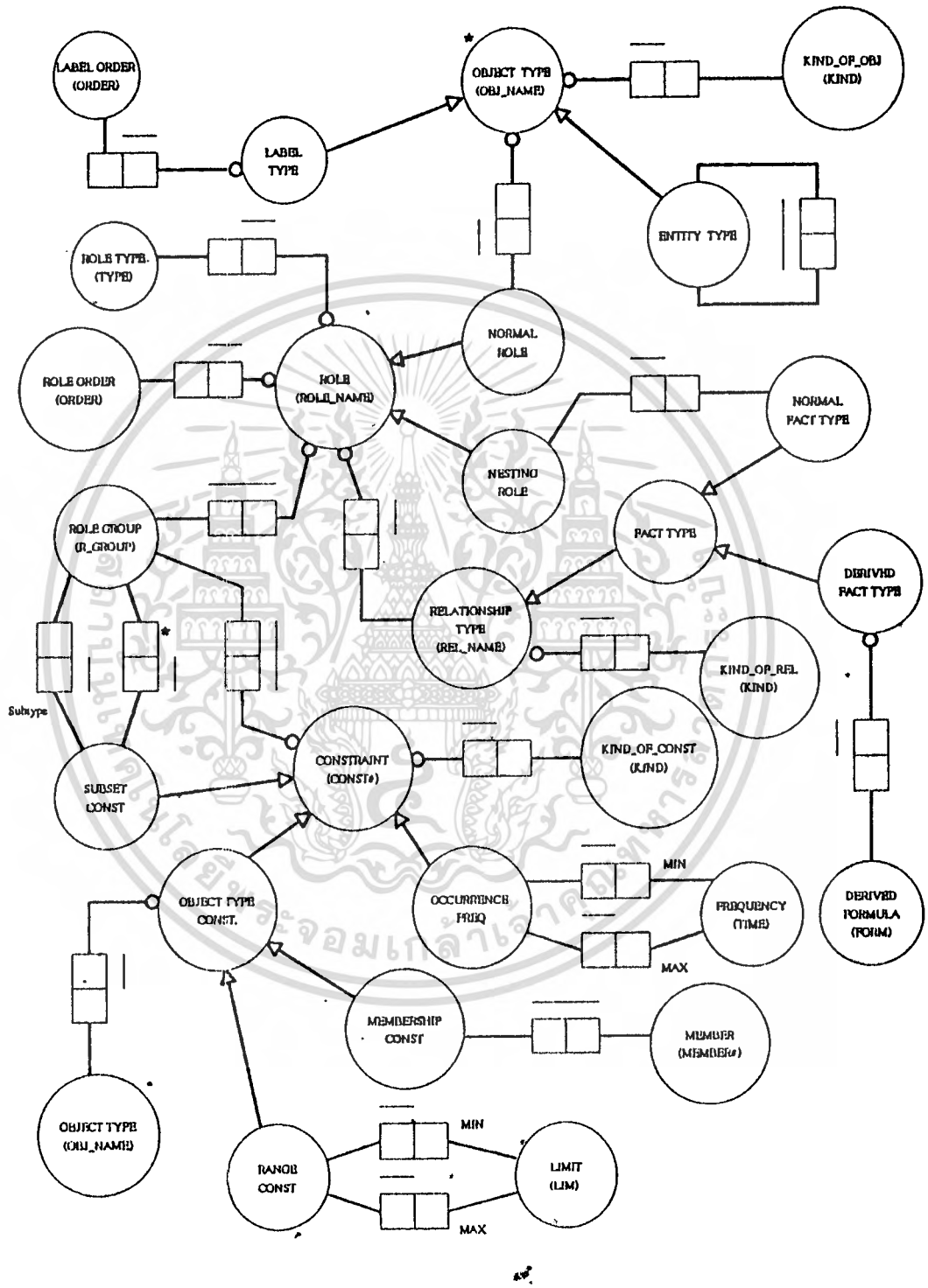
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อแอตตริบิวท์	จำนวนอักขระ	ความหมาย
MEMBER	10	สมาชิกที่กำหนดใน Membership constraint เก็บเป็นสตริงขนาดไม่เกินที่กำหนด
UPPER_LIMIT	5	เป็นค่าตัวเลขที่มากที่สุดที่เก็บได้ในแต่ละ Object
LOWER_LIMIT	5	เป็นค่าตัวเลขที่น้อยที่สุดที่เก็บได้ในแต่ละ Object
ENTITY_NAME	20	ชื่อ Entity
LABEL_NAME	20	ชื่อ Label
COL_NAME	20	ชื่อ Column
TABLE_NAME	20	ชื่อตาราง
VIEW_NAME	20	ชื่อวิว
DERIVED_DEF	100	สูตรที่ใช้ในการ Derived

ตาราง 4.3 แสดงความหมายของคอลัมน์ต่าง ๆ ในตารางของเมต้า และแมปปิ้งอินฟอร์เมชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

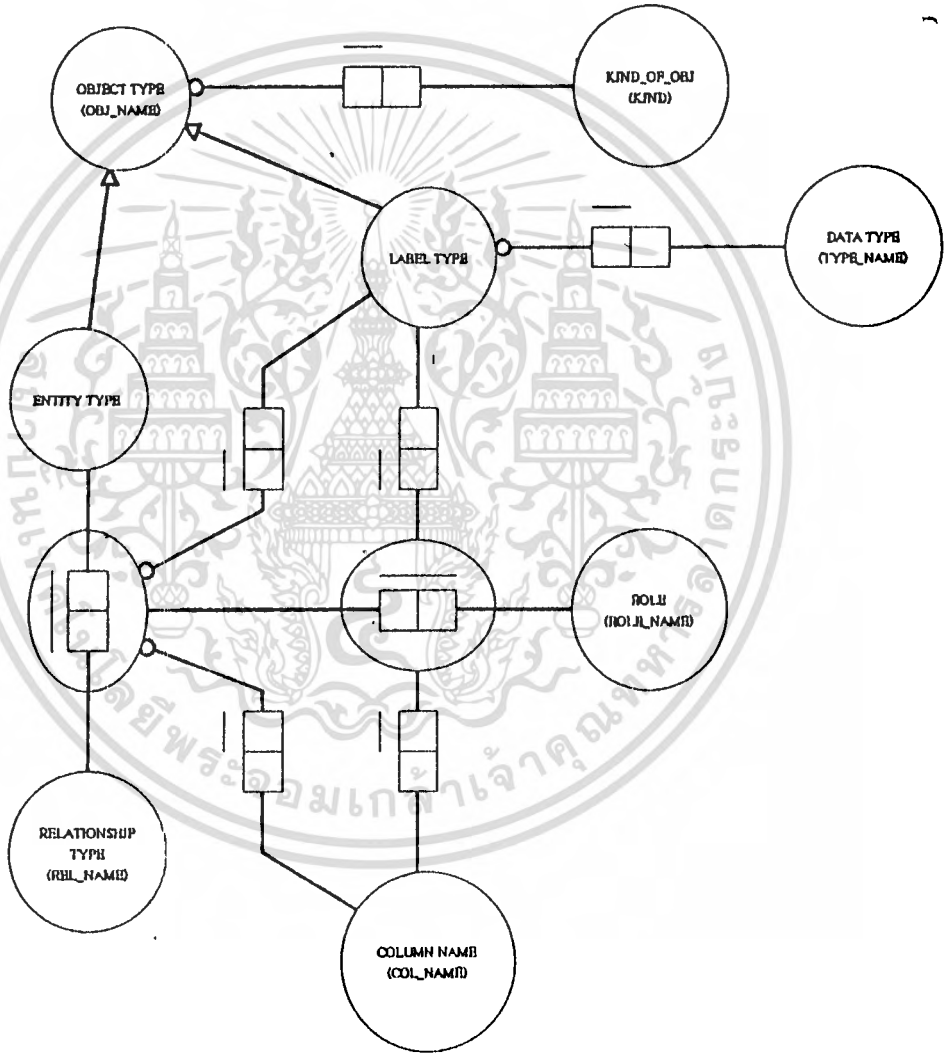
NIAM - CONCEPTUAL METASCHEMA



รูปที่ 4.2 แสดง NIAM-Conceptual Metaschema

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

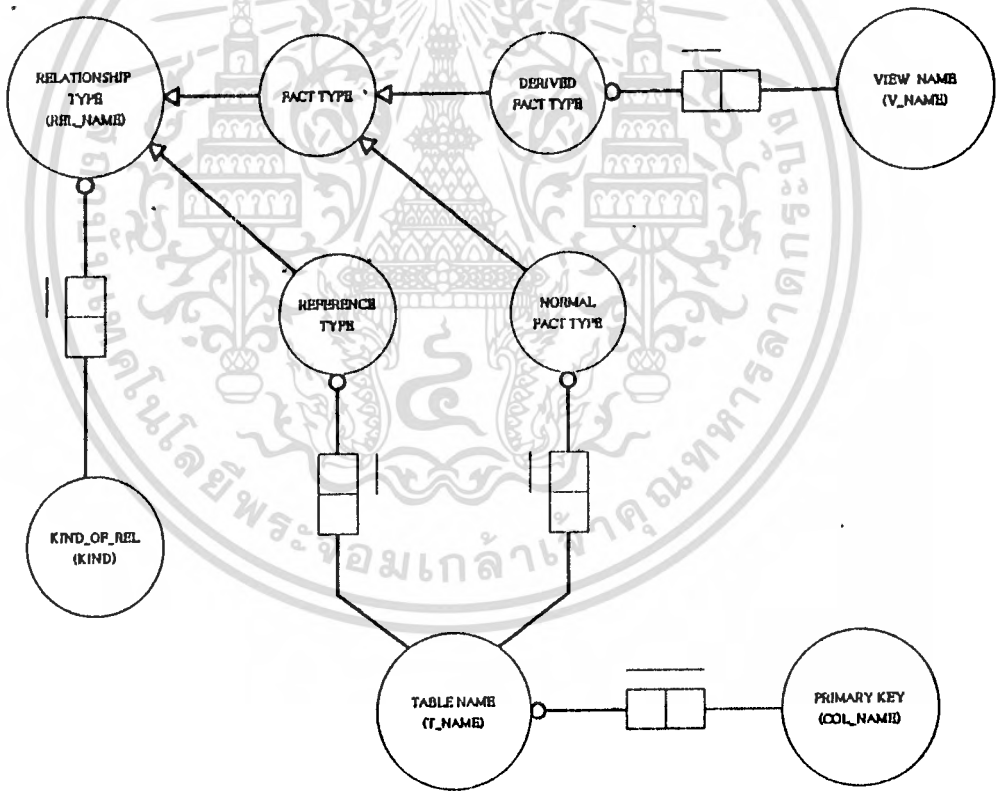
MAPPING INFORMATION NO. 1



รูปที่ 4.3 Mapping Information No1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAPPING INFORMATION NO.2



รูปที่ 4.4 Mapping Information No2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 วัตถุประสงค์ของโครงงานระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์

1. เพื่อเป็นการศึกษาเกี่ยวกับการออกแบบฐานข้อมูล โดยให้มีการทำงานอย่างอัตโนมัติ

2. เพื่อเป็นการวิจัยและทดลองการนำวิธีการออกแบบฐานข้อมูลด้วยวิธีไนแอม มาใช้ในการออกแบบฐานข้อมูลแบบสัมพันธ์

3. เพื่อให้การออกแบบเป็นไปได้อย่างง่ายดาย เป็นเครื่องมือในการออกแบบฐานข้อมูล เนื่องจากในขั้นตอนการออกแบบ อาจเกิดข้อผิดพลาดขึ้นได้

4.9 ส่วนประกอบของโครงงาน

ระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ แสดงด้วยไดอะแกรม ประกอบด้วยกระบวนการจากโมดูลปฏิบัติการ 4 โมดูล อันได้แก่

1. REDD-Gr เป็นกระบวนการรับอินพุทรูปแบบจำลองไนแอมที่สร้างขึ้นภายใต้ระบบช่วยวาด ที่ทำงานภายใต้เอนไวรอนเมนต์ของระบบเชื่อมต่อผู้ใช้แบบกราฟฟิก หรือ จียูไอ (GUI:Graphic User Interface) แบบจำลองไนแอมนี้ถูกออกแบบโดยผู้ใช้งานซึ่งก็คือนักออกแบบฐานข้อมูล โดยอาศัยขั้นตอนการออกแบบคอนเซ็ปชวลสกีมา่ (CSDP)ที่อธิบายไว้ในบทที่ 2 เรื่องหลักการเบื้องต้น เมื่อผู้ใช้ทำการเซฟแบบจำลองไนแอม โปรแกรมช่วยวาดจะสร้างเอาท์พุทออกมา 2 ชนิด ได้แก่

1.1 ไฟล์เดสคริบเตอร์ คือเท็กซ์ไฟล์ซึ่งสร้างขึ้นโดยโปรแกรมช่วยวาด ใช้ในการบอกลักษณะของอ็อบเจ็คต่าง ๆ ของไนแอมที่สร้างขึ้น ไม่ว่าจะเป็น เอนติตี้ไทป์ เลเบิ้ลไทป์ แฟกท์ไทป์ หรือ คอนสเตรนท์ต่าง ๆ รวมทั้งความสัมพันธ์ระหว่างอ็อบเจ็ค เช่น ความสัมพันธ์ในลักษณะการอ้างถึง (reference) ในลักษณะการถ่ายทอด (subtype) หรือ ลักษณะข้อมูลความจริง (fact type) โดยเก็บลักษณะเหล่านี้ในรูปแบบที่กำหนดขึ้น รายละเอียดของรูปแบบไฟล์เดสคริบเตอร์จะอยู่ในบทที่ 5 ต่อไป

1.2 เมต้าไฟล์ คือกราฟฟิกไฟล์ที่เป็นมาตรฐานในการเก็บภาพกราฟฟิกอีกแบบหนึ่งที่ระบบไมโครซอฟต์วินโดว์เตรียมไว้สำหรับผู้พัฒนาโปรแกรม มีลักษณะเป็นไฟล์ที่เก็บคำสั่งกราฟฟิกของระบบ จีดีไอ (GDI:Graphic Device Interface) ของไมโครซอฟต์วินโดว์เอนไวรอนเมนต์บนไมโครซอฟต์วินโดว์ จะถือเสมือนหนึ่งว่าไฟล์ก็เป็นอุปกรณ์ภายนอกตัวหนึ่ง การใช้คำสั่งในการแสดงผลบนจอภาพ ด้วยฟังก์ชันในชุดโปรแกรมพัฒนาระบบของไมโครซอฟต์วิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดว์ สามารถทำให้เกิดรูปภาพบนจอภาพ เช่นเดียวกับการเกิดรูปภาพลงบนไฟล์ เมื่อมีการเปิดไฟล์ก็เหมือนกับการเปิดการทำงานของจอ การโหลดเมต้าไฟล์จากไฟล์ลงบนจอภาพ ก็เหมือนกับการย้ายการแสดงผลจากอุปกรณ์หนึ่งไปยังอีกอุปกรณ์หนึ่ง (รายละเอียดของเมต้าไฟล์อยู่ในภาคผนวก)

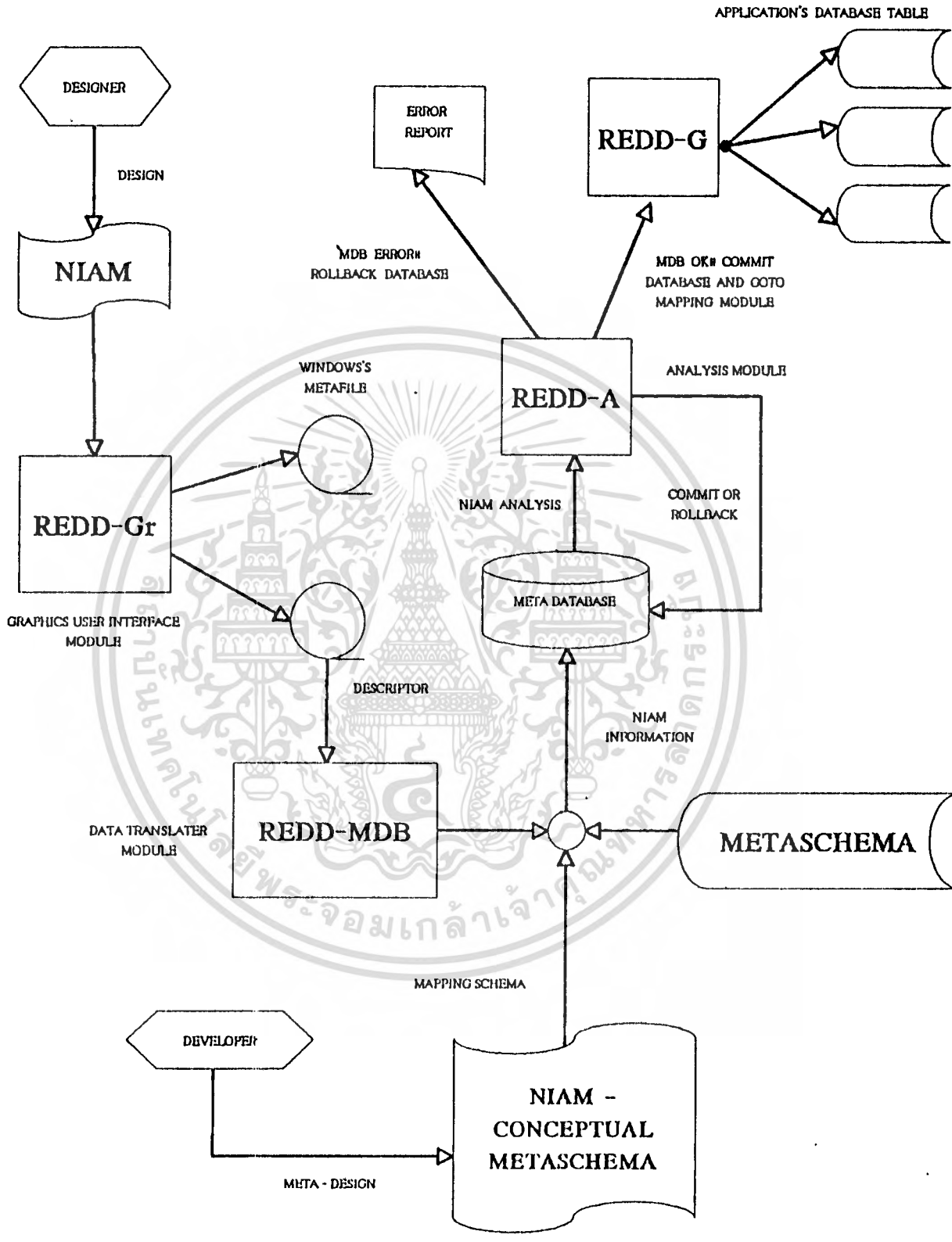
2. REDD-MDB กระบวนการสร้างเมต้าเดต้าเบส รับอินพุทในรูปเดสคริบเตอร์ของแบบจำลองข้อมูลไนแอม ที่ได้จากกระบวนการของโมดูล REDD-Gr กระบวนการนี้จะนำเอาเดสคริบเตอร์ผ่านเข้าพักไว้ในโครงสร้างข้อมูลภายใน (in-core structure) แล้วทำการแปลงข้อมูลให้อยู่ในรูปของข้อมูลในทัพเพิ้ลของรีเลชันฐานข้อมูลของเมต้าสกีมม่า เรียกเหล่าข้อมูลนี้ว่าเมต้าเดต้าเบส จากไดอะแกรมจะพบบุคคลากรเพิ่มขึ้นอีกกลุ่มหนึ่งในบล็อคที่ 8 เรียกว่าผู้พัฒนาเมต้าสกีมม่า ผู้พัฒนากลุ่มนี้จะทำการค้นคว้าหาแบบจำลองระดับแนวคิดของเมต้าสกีมม่าเพื่อใช้แทนข้อมูล เอาท์พุทที่ได้จากกระบวนการแปลงนิยามข้อมูล รวมกับข้อกำหนดเพื่อความถูกต้องของแบบจำลองไนแอม โดยมีรีเลชันฐานข้อมูลเมต้าสกีมม่ารองรับอยู่ ก็คือเมต้าเดต้าเบสในบล็อคที่ 9 นั่นเอง

3. REDD-A กระบวนการวิเคราะห์เมต้าเดต้าเบส เพื่อความถูกต้องของแบบจำลองข้อมูลไนแอม กระบวนการนี้ประกอบขึ้นด้วยกฎความถูกต้อง (NIAM Validation rules) เป็นหลัก การวิเคราะห์จะยึดถือกฎนี้เป็นบรรทัดฐาน หากการวิเคราะห์สัมฤทธิ์ผล คือเมต้าเดต้าเบสไม่มีข้อผิดพลาดร้ายแรง จะส่งต่อการทำงานไปยังกระบวนการสร้างรีเลชันฐานข้อมูลด้วยโมดูล REDD-G ต่อไป แต่หากพบข้อผิดพลาดจะทำการทำลายเมต้าเดต้าเบสเสีย เพื่อไม่ให้นำไปทำการแมปปิ้ง

4. REDD-G เป็นส่วนที่ทำการนำข้อมูลที่เป็นความสัมพันธ์ของแบบจำลองไนแอมที่อยู่ในตารางเมต้า มาทำการผ่านขั้นตอนการแปลงที่มีขั้นตอนเป็นลำดับ โดยในระหว่างการทำการแปลงจำเป็นต้องมีการสร้างตารางเฉพาะเพื่อให้สามารถทำการแปลงได้สะดวก ถ้าในระหว่างการแปลงไม่มีข้อผิดพลาดเกิดขึ้น ก็จะทำให้ได้ตารางความสัมพันธ์ที่เรียกว่า รีเลชันแนลสกีม่า (Relational Schema) เพื่อให้ผู้ใช้สามารถนำไปใช้งานได้ ซึ่งการติดต่อกับผู้ใช้ก็จะทำการผ่านส่วนของเอนไวรอนเมนต์ ให้มีลักษณะยูสเซอร์เฟรนลี่ย์ (user friendly) เพื่อให้ผู้ใช้เข้าใจและใช้งานได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REDD - SYSTEM DIAGRAM



รูปที่ 4.5 REDD - System Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ส่วนโครงสร้างข้อมูล

(A Relational Database Designer Genenator Part)

โครงงานนี้เป็นส่วนหนึ่งของระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ ซึ่งเป็นที่มีหน้าที่สำคัญในการนำข้อมูลของนิยามระดับแนวคิดของไนแอม (NIAM Conceptual Schema) ที่สร้างขึ้นจากผู้ใช้ในส่วนของโครงงานในส่วนออกแบบ ซึ่งจะถูกเก็บอยู่ในตารางเมต้า ซึ่งเป็นตารางที่ได้ออกแบบมาเพื่อให้สามารถเก็บความสัมพันธ์ต่างๆของรูปไนแอมที่ผู้ใช้ต้องการอย่างครบถ้วนและถูกต้องและตารางที่เก็บข้อมูลที่สำคัญที่ใช้ในการแปลงรูปไนแอม (NIAM Mapping Information) มาทำการผ่านโปรแกรมในส่วนนี้เพื่อสร้างตารางที่เป็นความสัมพันธ์ต่างๆของไนแอม ที่ผู้ใช้ได้สร้าง และเพื่อให้ผู้ใช้สามารถนำตารางดังกล่าวที่สร้างขึ้นไปใช้งานได้จริง นั่นคือสามารถนำตารางนั้นไปสร้างให้ผู้ใช้ เพื่อเก็บข้อมูลที่ต้องการได้ โดยอาศัยวิธีขั้นตอนการแปลง (Treansformation algorithms) ที่ได้กล่าวมาแล้ว

5.1 สภาวะแวดล้อมทางฮาร์ดแวร์และซอฟแวร์ที่ใช้ในการพัฒนา

1. เครื่องคอมพิวเตอร์ส่วนบุคคล (Personel Computer) ที่มีไมโครโปรเซสเซอร์ตั้งแต่เบอร์ 80286 ขึ้นไป ที่สามารถอ้างหน่วยความจำได้ถึง 2 เมกะไบต์ขึ้นไป ทั้งนี้เนื่องจากจะต้องทำการรันระบบจัดการฐานข้อมูลออราเคิล (Oracle) ด้วย

2. โปรแกรมระบบจัดการฐานข้อมูลแบบสัมพันธ์ ที่ชื่ออออราเคิล (Oracle)

3. แผ่นแม่เหล็กสำหรับเก็บข้อมูลชนิดแข็ง (Hard disk) ขนาด 40 เมกะไบต์ขึ้นไป

4. โปรแกรมที่ใช้ในการคอมไพล์ประโยคคำสั่งของเอสคิวแอลที่ประกอบอยู่ในโปรแกรมภาษาซี ให้เป็นภาษาซี ชื่อ โปรสตาร์ซีทูล (Pro*c Tool)

5. โปรแกรมสำหรับการคอมไพล์ภาษาซี ชื่อ Microsoft C (ที่ใช้อยู่เป็นเวอร์ชัน 5.1) ที่สามารถคอมไพล์ด้วยโมเดลใหญ่ (large model)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ส่วนประกอบของระบบ

ส่วนของระบบสามารถแบ่งออกได้เป็นส่วนย่อย ดังนี้

1. ส่วนที่ทำหน้าที่ติดต่อกับผู้ใช้ (User interface)

เป็นส่วนที่ทำหน้าที่ในการติดต่อกับผู้ใช้ เช่น การจัดการหน้าจอแสดงผล และกระทำงานทุกอย่างเกี่ยวกับไฟล์ เช่น การอ่านข้อมูลจากแผ่นดิกส์ (Load), การเก็บข้อมูลลงแผ่นดิสก์ (Save) เป็นต้น

2. ส่วนที่ทำหน้าที่ในการจัดการโครงสร้างข้อมูล (Data manipulation)

เป็นส่วนที่สำคัญที่สุดของงานทั้งหมด โดยในส่วนนี้จะเป็นงานเกี่ยวกับการนำข้อมูลที่เก็บอยู่ในตารางเมต้า มาทำการสร้างตารางตามความสัมพันธ์ของรูปไนแอม ซึ่งขั้นตอนการแปลง (Transformation) ที่ได้กล่าวแล้วข้างต้น นอกจากข้อมูลที่อยู่ในตารางเมต้าแล้ว ยังต้องอาศัยข้อมูลที่สำคัญนอกเหนือนั้นอีก เช่น ข้อมูลชื่อตาราง (table name) ข้อมูลชื่อคอลัมน์ (column name) เป็นต้น โดยเรียกรวมว่าแมปปิ้งอินฟอร์เมชัน (Mapping information) ข้อมูลเหล่านี้จะถูกกำหนดเป็นอีกฐานข้อมูลหนึ่ง ซึ่งใช้ในกระบวนการแปลง โดยเฉพาะ ซึ่งในส่วนนี้จะต้องมีการติดต่อกับระบบจัดการฐานข้อมูลแบบสัมพันธ์ออราเคิล เพื่อนำข้อมูลที่เก็บอยู่ในตารางจากฐานข้อมูลของออราเคิลมาใช้ได้

นอกจากนี้ยังมีงานอีกส่วนหนึ่งซึ่งเป็นงานส่วนที่ไม่เกี่ยวข้องกับงานของระบบนี้โดยตรง แต่เป็นการทำงานที่ร่วมกันกับกลุ่มโครงงานการพัฒนากฎในฐานข้อมูล เพื่อทำการสร้างโปรแกรมที่ทำการสร้างคิวรี่ ที่ใช้ในการตรวจสอบความถูกต้องของข้อมูลตามคอนสเตรนท์แต่ละชนิด และทำการตรวจสอบความถูกต้องของข้อมูลตามคิวรี่ด้วยซึ่งข้อมูลที่ต้องนำมาใช้ในการสร้างคิวรี่ เป็นข้อมูลที่อยู่ในตารางเมต้า และข้อมูลที่อยู่ในตารางเฉพาะที่ได้ผ่านการใส่และแก้ไขข้อมูลมาจากระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ส่วนโครงสร้างข้อมูล

5.3 ขั้นตอนในการพัฒนาโปรแกรม

ในการพัฒนาโปรแกรมของระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ส่วนโครงสร้างข้อมูล มีขั้นตอนต่างๆที่จำเป็น และวิธีการค่อนข้างยุ่งยาก ทั้งนี้ เนื่องจากจะต้องทำการติดต่อกับระบบจัดการฐานข้อมูลออราเคิล เพื่อนำข้อมูลที่อยู่ภายใต้การจัดการของออราเคิล และมีโครงสร้างข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนหนึ่งที่มีลักษณะเป็น ตาราง เพื่อให้สามารถเก็บข้อมูลและการนำข้อมูลมาใช้ได้สะดวกขึ้น ดังนั้นจึงมีขั้นตอนการพัฒนาโปรแกรมที่ซับซ้อนกว่า การพัฒนาโปรแกรมด้วยภาษาโครงสร้างธรรมดา ซึ่งสามารถกล่าวได้เป็นขั้นตอนดังนี้

1. ทำการออกแบบโครงสร้างข้อมูลที่จำเป็นต้องใช้ให้มีความกระทัดรัดและสามารถใช้งานได้ เพื่อรองรับโปรแกรมที่ซับซ้อน แบ่งโปรแกรมออกเป็นส่วนๆ เพื่อง่ายต่อการเขียน และการแก้ไขเพิ่มเติม ซึ่งในส่วนนี้จะสามารถแบ่งออกได้เป็น 2 ส่วนคือ

1.1 ส่วนของโปรแกรมที่เป็นภาษาซีล้วนๆ ไม่ได้ติดต่อกับฐานข้อมูล ซึ่งส่วนใหญ่จะเป็นส่วนที่ทำหน้าที่ในการติดต่อกับผู้ใช้และการแสดงผลทางหน้าจอ โดยโปรแกรมในส่วนนี้จะเป็นไฟล์ที่มีนามสกุลเป็น .C

1.2 ส่วนของโปรแกรมที่มีการติดต่อกับฐานข้อมูล ซึ่งส่วนนี้จะมีการเขียนเป็นลักษณะที่เป็นภาษาซี ผสมกับ เอมเบดด์เอสคิวแอล (Embeded SQL) เพื่อให้ในการติดต่อกับฐานข้อมูล โดยโปรแกรมในส่วนนี้จะเป็นไฟล์ที่มีนามสกุลเป็น .PC

2. ทำการพรีคอมไพล์ไฟล์นามสกุล .PC ทั้งหมด ให้อยู่ในรูปของไฟล์นามสกุล .C โดยใช้โปรแกรม PC หรือ พรีคอมไพล์ (Precompile)

3. ทำการคอมไพล์ไฟล์นามสกุล .C ทั้งหมด ให้ได้ผลลัพธ์เป็นไฟล์นามสกุล .OBJ (Object file) โดยใช้โปรแกรม CL ของ Microsoft C

4. ทำการลิงค์ (link) ไฟล์ที่มีนามสกุล .OBJ ทั้งหมดให้ได้ผลลัพธ์ออกมาเป็นไฟล์ที่ใช้งานได้ (Executable file)

5.4 การออกแบบโครงสร้างของระบบ

โครงสร้างของระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ส่วนการออกแบบโครงสร้างข้อมูลของระบบสามารถอธิบายแยกประเภทตามลักษณะการเขียนโปรแกรมได้เป็น 3 แบบด้วยกันคือ

1. โปรแกรมส่วนสร้างความสัมพันธ์ (Generate Relation)
2. โปรแกรมส่วนเอนไวรอนเมนท์ (Environmant Program)
3. โปรแกรมส่วนคอนสเตรนแฮนดิ้ง (Constraint Handing)

1. โปรแกรมส่วนสร้างความสัมพันธ์ (Generate Relation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1 ส่วนโครงสร้างข้อมูล

ส่วนของโครงสร้างข้อมูลของโปรแกรมส่วนสร้างความสัมพันธ์นั้นได้กล่าวไว้แล้วในบทที่ 4 ดังตารางในหัวข้อที่ 4.4 และ 4.6

1.2 ส่วนโครงสร้างของโปรแกรม

จากโปรแกรมที่ถูกสร้างขึ้นมาแล้วนั้นสามารถอธิบายโมดูลส่วนหลัก ๆ ได้ดังนี้

1.2.1 ส่วนกำหนดค่าเริ่มต้น (Initial Process)

จะทำการสร้างตารางชั่วคราวขึ้นมา โดยโปรแกรมจะเรียกใช้ไฟล์ main_0.pc, init11.pc และ ไฟล์ cretab_2.pc ซึ่งหลังจากการผ่านกระบวนการแปลง (Mapping Process) เป็นความสัมพันธ์ของผู้ใช้แล้ว ตารางเหล่านี้ก็จะถูกลบออกจากระบบฐานข้อมูล ซึ่งตารางที่สร้างขึ้นมีโครงสร้างดังนี้

- VIEW ENRELATION

ความสัมพันธ์หลักก็คือ แต่ละเอนติตี้มี ชนิดของรีเลชันใดมาต่อบ้าง

ENTITY	KIND_OF_RELATION
CHAR(20)	CHAR(10)

ตาราง 5.1 แสดงวิว ENRELAITON

- TABLE ENCOLUMN

ตารางนี้จะมีสัมพันธ์ระหว่างระหว่างคอลัมน์ ENTITY_F และ ENTITY_S โดยค่าทั้งสองจะแตกต่างกันในกรณีของรูปไนแอมมีรูปในกรณีของไฮราคี (Case Hirarchy) ค่าของ ENTITY_F จะเป็นค่าต้นของไฮราคี ส่วน ENTITY_S จะเป็นค่าของปลายของไฮราคี ตารางนี้จะมีการใช้โดยชี้หาชื่อคอลัมน์ COL_NAME เมื่อทราบ ENTIY_F และ ROLE_NAME_F

ENTITY_F	ROLE_NAME_F	ENTITY_S	ROLE_NAME_S	REL_NAME	COL_NAME
CHAR(20)	CHAR(20)	CHAR(20)	CHAR(20)	CHAR(10)	CHAR(20)

ตาราง 5.2 แสดงตาราง ENCOLUMN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- TABLE FACTEN]UNIQ

ตารางนี้จะมีความสัมพันธ์หลัก ๆ คือ คอลัมน์ FACT]TYPE, ENTITY และ UNIQ ซึ่งในกระบวนการแปลงจะมีการพิจารณาข้อมูลในตารางนี้เป็นตารางแรก โดยวิเคราะห์จากค่า UNIQ เพื่อตรวจสอบหา FACT]TYPE ในกรณีต่าง ๆ ที่ต้องการ

FACT]TYPE	ENTITY	ROLE]NAME	UNIQ	ROLE]ORDER
CHAR(10)	CHAR(20)	CHAR(20)	CHAR(5)	CHAR(1)

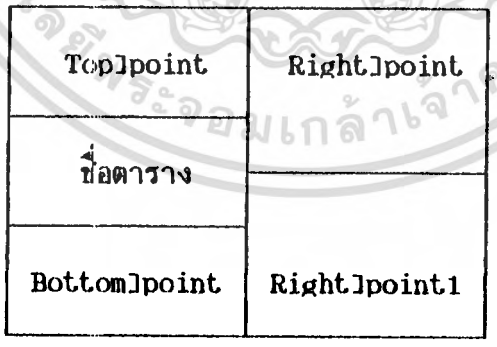
ตาราง 5.3 แสดงตาราง FACTEN]UNIQ

เมื่อสร้างตารางชั่วคราวเพื่อช่วยในการแปลงเสร็จแล้ว ในไฟล์ instab]3.pc และ ไฟล์ instab]4.pc จะทำการนำข้อมูลต่าง ๆ ที่ได้จากตารางเมต้า และ แมปปิ้งอินพอร์เมชัน มาวิเคราะห์และเก็บลงตารางที่สร้างขึ้น

1.2.2 ส่วนแปลง (Mapping Process)

- ทำการสร้างลิงก์ลิสต์โดยเรียกจาก ไฟล์ list]5.pc สำหรับเก็บโครงสร้างของตารางที่แปลงมาได้ไว้ชั่วคราว เพื่อนำมาแสดงออกทางหน้าจอ และนำไปใช้ในส่วนอื่น ๆ อีก โดยโครงสร้างของลิงก์ลิสต์ประกอบไปด้วยโนด 3 ชนิดด้วยกันคือ ซึ่งมีลักษณะดังนี้

1 โนดของตาราง (Table node)



รูปที่ 5.1 แสดงโครงสร้างของโนดตาราง

Top]point เป็นพอยน์เตอร์ที่ชี้ไปยังโนดตารางก่อนหน้านี้

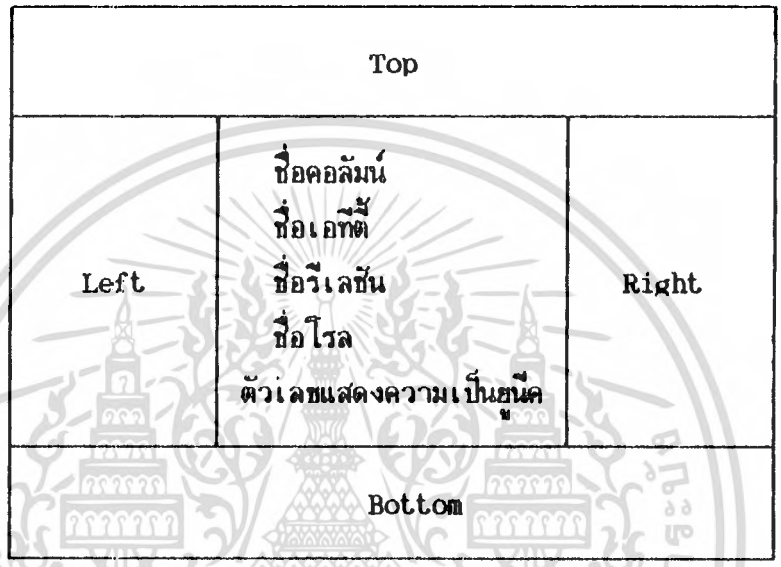
Bottom]ponint เป็นพอยน์เตอร์ที่ชี้ไปยังโนดตารางต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Left_point เป็นพอยน์เตอร์ที่ชี้ไปยังโนดคอลัมน์

Right_point1 เป็นพอยน์เตอร์ที่ชี้ไปยังโนดรีเลชัน

2 โนดคอลัมน์ (Column node)



รูป 5.2 แสดงโครงสร้างของโนดคอลัมน์

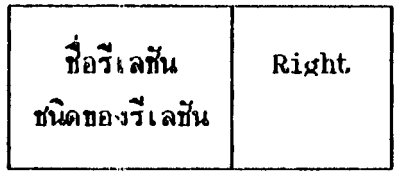
Top เป็นพอยน์เตอร์ที่ชี้ไปยังโนดคอลัมน์ที่อยู่ด้านบน

Bottom เป็นพอยน์เตอร์ที่ชี้ไปยังโนดคอลัมน์ที่อยู่ด้านล่าง

Left เป็นพอยน์เตอร์ที่ชี้ไปยังโนดคอลัมน์ที่อยู่ด้านซ้าย

Right เป็นพอยน์เตอร์ที่ชี้ไปยังโนดคอลัมน์ที่อยู่ด้านขวา

3 โนดรีเลชัน (Reltion node)

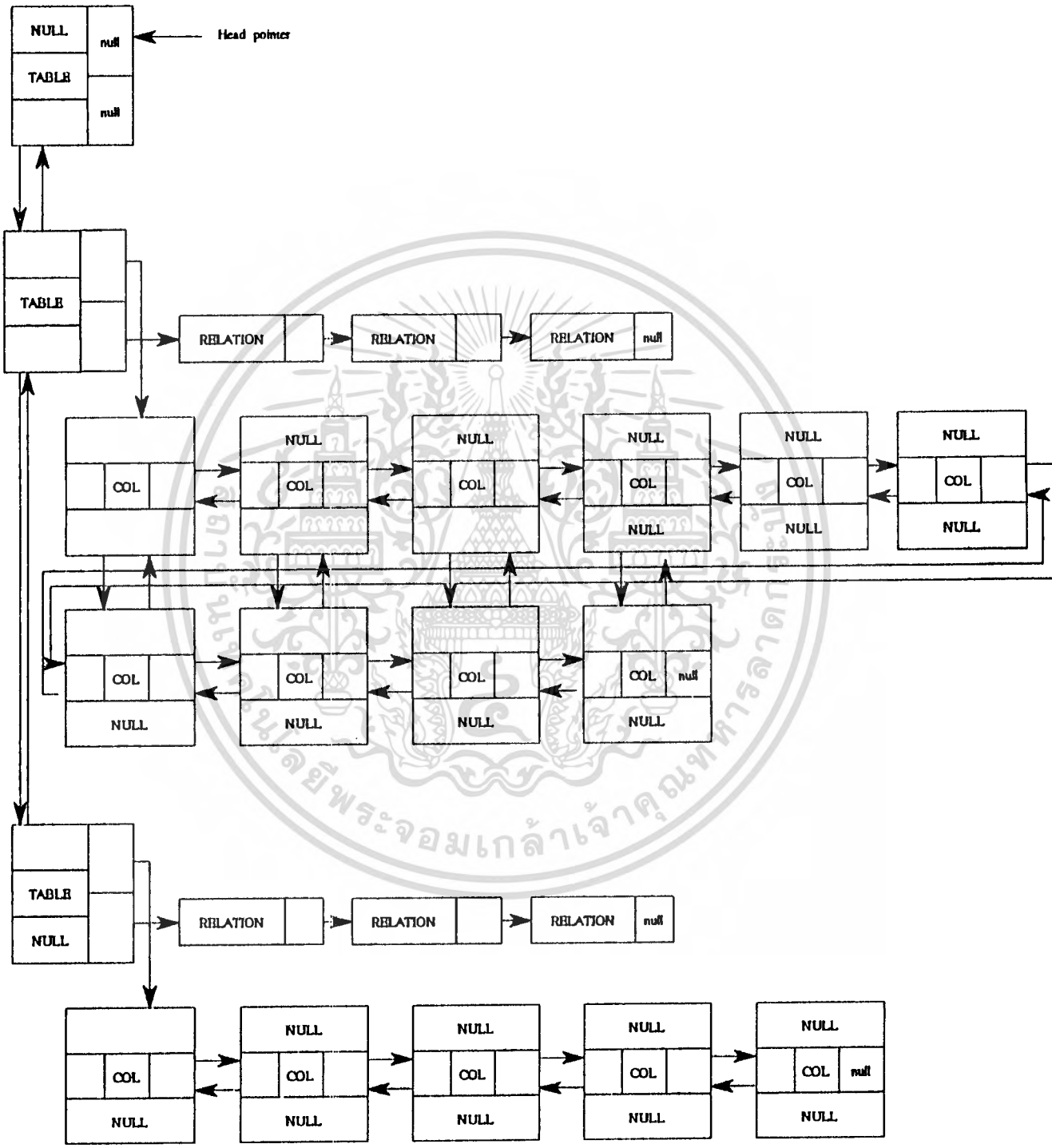


รูป 5.3 แสดงโครงสร้างของโนดรีเลชัน

Right เป็นพอยน์เตอร์ที่ชี้ไปยังโนดรีเลชันตัวต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยลักษณะของโนดทั้ง 3 แบบข้างต้นสามารถนำมาแสดงรวมเข้าด้วยกัน เพื่ออธิบายโครงสร้างของความสัมพันธ์ของผู้ใช้ได้จากกระบวนการแปลงดังตัวอย่างต่อไปนี้



รูป 5.4 แสดงโครงสร้างของความสัมพันธ์ของลิงก์ลิสต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำการสร้าง 1 ความสัมพันธ์สำหรับแฟกท์ไทป์ที่มีคอนสเตรท์ชนิดยูนีคเนสท์คลุมทุกโรล อยู่ในฟังก์ชัน fact_nary() ไฟล์ f_nary_6.pc

- ทำการสร้าง 1 ความสัมพันธ์สำหรับแฟกท์ไทป์ที่มีโรลมากกว่า 2 โรล และมีคอนสเตรท์ชนิดยูนีคเนสท์คลุมทั้งหมดยกเว้น 1 โรล ซึ่งจะอยู่ไปฟังก์ชัน fact_n_1 () ไฟล์ f_n_1_7.pc

- ทำการรวมกลุ่มแฟกท์ไทป์ที่มีโรลเท่ากับ 2 และมีคอนสเตรท์ชนิดยูนีคคลุม 1 โรล ซึ่งจะอยู่ในฟังก์ชัน bi_fact () ไฟล์ bi_fa_8.pc

- ทำการสร้างความสัมพันธ์ที่เหลือจากทั้ง 3 แบบ ที่กล่าวมาแล้ว ซึ่งจะอยู่ในฟังก์ชัน no_common () ไฟล์ no_com_9.pc

จากโครงสร้างของโปรแกรมส่วนสร้างความสัมพันธ์ สามารถเขียนรูปแบบการเรียก (Call) ของแต่ละไฟล์ได้ดังรูป 5.5

2. โปรแกรมส่วนเอนไวรอนเมนท์ (Environment Program)

ในส่วนนี้จะกล่าวเพียงโครงสร้างของโปรแกรมเท่านั้น เนื่องจากโปรแกรมส่วนเอไวรอนเมนท์ไม่มีโครงสร้างข้อมูลของระบบอยู่ ในส่วนโครงสร้างของโปรแกรมส่วนเอนไวรอนเมนท์นั้นจะประกอบไปด้วย ไฟล์ ดังนี้คือ

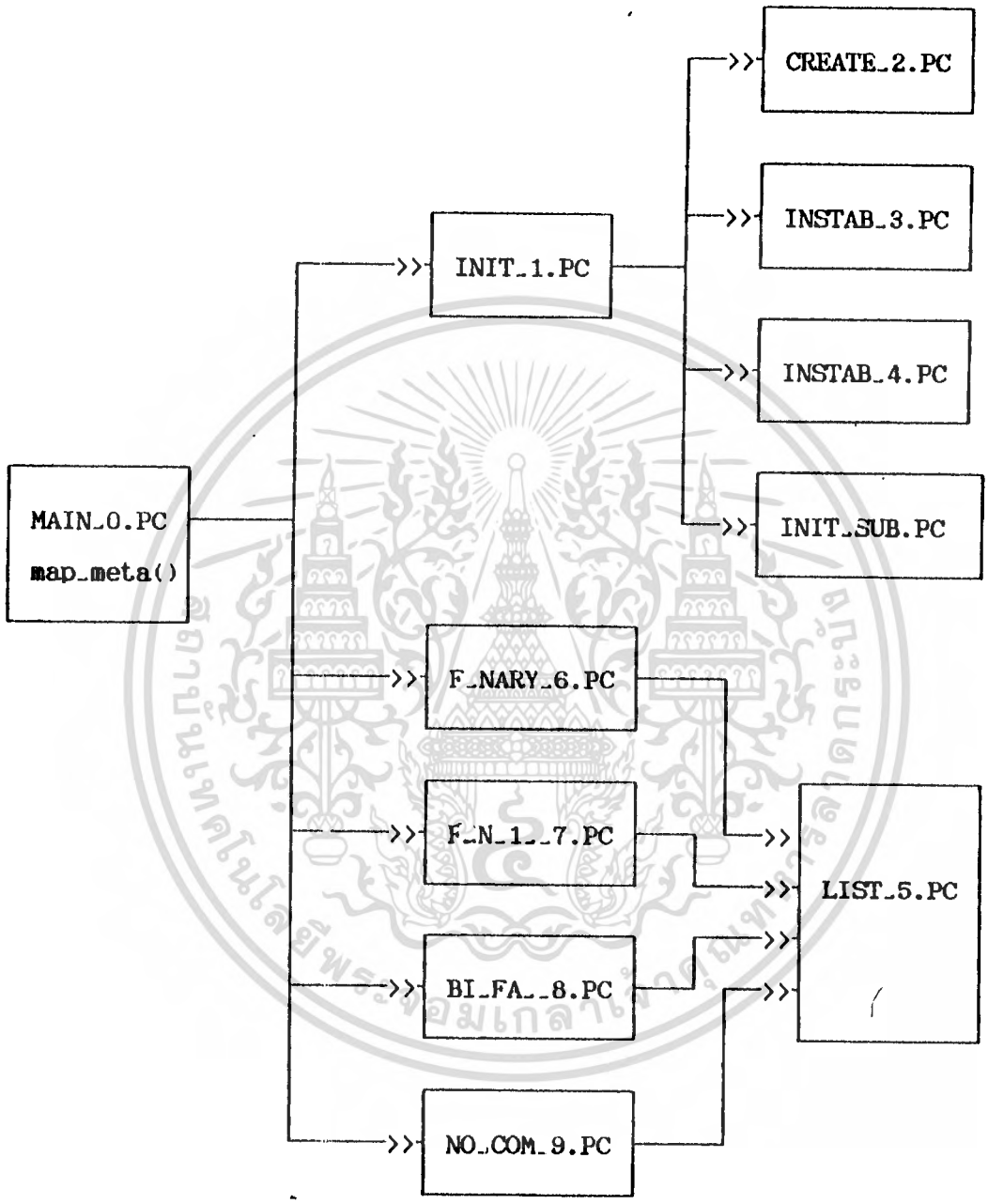
- MENU.C ซึ่งจะเป็นส่วนหลักของระบบงานนี้ คือจะมีฟังก์ชัน main() อยู่ในไฟล์นี้ การทำงานหลัก ๆ ของไฟล์นี้ก็คือ ส่วนที่กำหนดขนาดต่าง ๆ ที่จะแสดงป๊อปอัพ (Pop-up) ซึ่งจะเรียกผ่านไฟล์ POPUP.C และส่วนแสดงพูลดาวน์เมนู (Pull-down Menu) และฟังก์ชันทำงานในพูลดาวน์เมนู ที่เรียกจะถูกส่งไปทำงานในไฟล์ FUNCTION.C

- POPUP.C เป็นไฟล์ที่เก็บฟังก์ชันที่เกี่ยวกับการแสดงป๊อปอัพในแบบต่าง ๆ ซึ่งจะมีฟังก์ชันดังนี้ คือ popShow(), popErase(), popCenter(), popRewrite(), popHilite(), popNormal() และ menubar() ซึ่งรายละเอียดเกี่ยวกับฟังก์ชันเหล่านี้จะกล่าวในคู่มือการพัฒนาโปรแกรมระบบ อีกทีหนึ่ง

- TEXTBOX.C เป็นไฟล์ที่มีฟังก์ชัน textbox() เพียงฟังก์ชันเดียว ซึ่งจะใช้สำหรับการลากกรอบของบล็อก โดยให้มีตัวเลือกสำหรับกรอบที่จะวาด เช่น กรอบเส้นเดี่ยว กรอบเส้นคู่เป็นต้น

- CURSOR เป็นไฟล์ที่ปรับขนาดของเคอร์เซอร์ที่ปรากฏบนหน้าจอ ประกอบด้วยฟังก์ชัน cursoff(), curson() และ curShape ()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 แสดงความสัมพันธ์ของการเรียกใช้ในแต่ละไฟล์ของโปรแกรมส่วนสร้างความสัมพันธ์

- FUNC_MEM.C เป็นไฟล์ที่ประด้วยฟังก์ชันที่จัดการเกี่ยวกับหน่วยความจำสำหรับระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

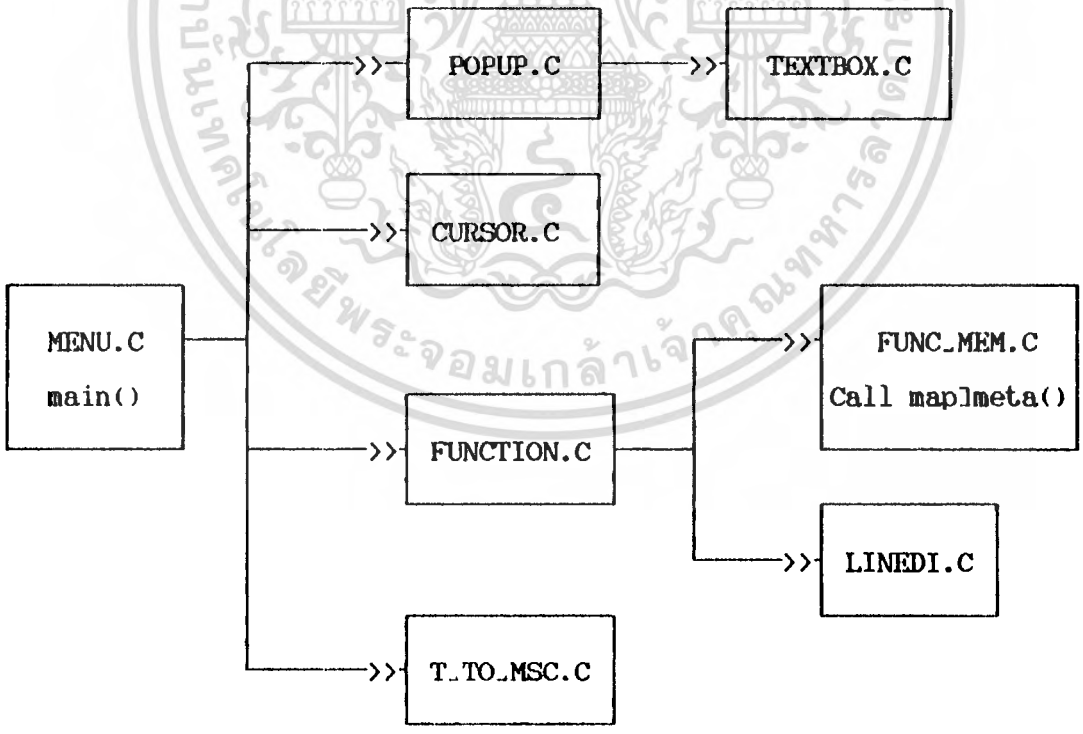
บบโดยเฉพาะ เช่น การดำนวนการแสดงผลของลิงก์ลิสต์ของความสัมพันธ์ของผู้ใช้ที่แปลงได้ออกทางจอภาพ การเขียนรูปเลือก หรือเส้นลงในหน่วนความจำ เป็นต้น

- FUNCTION.C เป็นไฟล์ที่ทำงานใน ฟังก์ชันของพลูดาวน์เมนูทุกฟังก์ชันที่ใช้ผู้ใช้เลือก เช่น การโหลดข้อมูลออกจากแผ่นข้อมูล การเก็บข็อมูลลงไปในแผ่นข้อมูล ส่วนแก้ไขลิงก์ลิสต์แสดงความสัมพันธ์ที่ได้จากกระบวนการแปลง และยังเป็นส่วนที่ไปเรียกกระบวนการแปลง (Call maplmeta()) ด้วย

- T_TO_MSC.C (TURBO TO MICROSOFT) เป็นไฟล์ที่ประกอบไปด้วยชื่อที่เป็นฟังชันของเทอร์โบซี เช่น gotoxy(),wherex(),poke() เป็นต้น ซึ่งในแต่ละฟังก์ชันจะถูกเขียนใหม่ให้สามารถใช้งานได้กับ คอมไพล์เลอร์ของไมโครซอฟต์ซี

- LINEDI.C เป็นโปรแกรมบรรณาธิการเชิงบรรทัด หรือ ไลน์อิดิเตอร์ (Line editor)

ซึ่งความสัมพันธ์ของการเรียกในแต่ละไฟล์ของโปรแกรมส่วนเอนไวรอนเมนท์ สามารถแสดงได้ดังรูปที่ 5.6



รูป 5.6 แสดงความสัมพันธ์ของการเรียกใช้ในแต่ละไฟล์
ของโปรแกรมส่วนเอนไวรอนเมนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. โปรแกรมส่วนคอนสเตรนท์แฮนด์ลิ่ง (Constraint Handing)

3.1 ส่วนโครงสร้างข้อมูล

โครงสร้างข้อมูลที่สำคัญในส่วนนี้ได้แก่ ข้อมูลที่อยู่ในตารางเมต้า และตารางเฉพาะที่ได้สร้างขึ้นมาใช้งานเฉพาะส่วนนี้ ซึ่งประกอบไปด้วยตาราง CONST_QUERY เป็นตารางที่ใช้เก็บข้อมูลที่เป็นเลขของคอนสเตรนท์และคิวรี่ที่สร้างได้ตามการทำงานของรูทีนของคอนสเตรนท์ประเภทนั้น ซึ่งมีอยู่ด้วยกัน 2 คอลัมน์ในกรณีที่คิวรี่ที่สร้างได้มีความยาวเกินกว่า 240 ตัวอักษร และ คอลัมน์สุดท้ายเป็นคอลัมน์ที่เป็นตัวบ่งว่า คอนสเตรนท์นั้นสามารถทำการตรวจสอบได้หรือไม่ โดยมีลักษณะตาราง

CONST#	QUERY1	QUERY2	NOTE
CHAR (10)	CHAR (240)	HAR (240)	CHAR (2)

ตาราง 5.4 แสดงตาราง CONST_QUERY

และ ตาราง CONST_VIEW เป็นตารางที่ใช้เก็บชื่อวิวที่สร้างในระหว่างการสร้างคิวรี่ เพื่อใช้สามารถทำการลบชื่อวิวที่ไม่ต้องการเกี่ยวกับคอนสเตรนท์ตัวนั้นได้ ประกอบด้วยคอลัมน์ 2 คอลัมน์คือ เลขคอนสเตรนท์ หรือ CONST# และ ชื่อวิว หรือ VIEW_NAME โดยมีคอลัมน์ดังต่อไปนี้

CONST#	VIEW_NAME
CHAR (10)	CHAR (20)

ตาราง 5.5 แสดงตาราง CONST_VIEW

3.2 ส่วนโครงสร้างโปรแกรม

3.2.1 ส่วนที่เป็นการสร้างคิวรี่

เพื่อใช้ในการตรวจสอบความถูกต้องของข้อมูลตามคอนสเตรนท์แต่ละชนิด ซึ่งในความรับผิดชอบของโครงงานนี้มีด้วยกัน 6 ชนิด กล่าวคือ

- คอนสเตรนท์แบบเมมเบอร์ชิพ (Membership constraint) จะอยู่ในไฟล์ชื่อ MEMBER.PC ซึ่งประกอบด้วยฟังก์ชันเพียงฟังก์ชันเดียวคือ ฟังก์ชัน member()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คอนสเตรนท์แบบเรนจ์ (Range constraint) จะอยู่ในไฟล์ชื่อ RANGE.PC ซึ่งประกอบด้วยฟังก์ชันเพียงฟังก์ชันเดียว คือ ฟังก์ชัน range_cons()

- คอนสเตรนท์แบบออเตอเรนซี่คอนฟรีเควนซี่ (Occurency frequency constraint) จะอยู่ในไฟล์ชื่อ OCCUREN.PC ซึ่งประกอบด้วยฟังก์ชันเพียงฟังก์ชันเดียวคือ ฟังก์ชัน occur_frq()

- คอนสเตรนท์แบบสับเซต (Subset constraint) จะอยู่ในไฟล์ชื่อ SUBSET.PC ซึ่งประกอบด้วยฟังก์ชัน subset()

- คอนสเตรนท์แบบเอ็กคลูชัน (Exclusion constraint) จะอยู่ในไฟล์ชื่อ EXCLUSION.PC ประกอบด้วยฟังก์ชัน exclusion()

- คอนสเตรนท์แบบอิควาลิตี้ (Equality constraint) จะอยู่ในไฟล์ EQUAL.PC ประกอบด้วยฟังก์ชัน equality()

3.2.2 ส่วนที่ทำการนำคิวรี่ที่สร้างได้มาตรวจสอบข้อมูล แยกเป็น 6 ไฟล์ตามคอนสเตรนท์แต่ละชนิด ดังนี้

- คอนสเตรนท์แบบเมมเบอร์ชิฟ (Membership constraint) จะอยู่ในไฟล์ชื่อ CHKMEM.PC ซึ่งประกอบด้วยฟังก์ชันเพียงฟังก์ชันเดียวคือ ฟังก์ชัน chk_mem()

- คอนสเตรนท์แบบเรนจ์ (Range constraint) จะอยู่ในไฟล์ชื่อ CHK_RAN.PC ซึ่งประกอบด้วยฟังก์ชันเพียงฟังก์ชันเดียว คือ ฟังก์ชัน chk_ran()

- คอนสเตรนท์แบบออคเตอเรนซี่คอนฟรีเควนซี่ (Occurency frequency constraint) จะอยู่ในไฟล์ชื่อ CHK_OCC.PC ซึ่งประกอบด้วยฟังก์ชันเพียงฟังก์ชันเดียวคือ ฟังก์ชัน chk_occ()

- คอนสเตรนท์แบบสับเซต (Subset constraint) จะอยู่ในไฟล์ชื่อ CHK_SUB.PC ซึ่งประกอบด้วยฟังก์ชัน

- คอนสเตรนท์แบบเอ็กคลูชัน (Exclusion constraint) จะอยู่ในไฟล์ชื่อ CHK_EX.PC ประกอบด้วยฟังก์ชัน chk_ex

- คอนสเตรนท์แบบอิควาลิตี้ (Equality constraint) จะอยู่ในไฟล์ CHK_EQU.PC ประกอบด้วยฟังก์ชัน chk_equa()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การใช้งานระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ ส่วนโครงสร้างข้อมูล

การใช้งานระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ ส่วนโครงสร้างข้อมูล แบ่งได้เป็น 2 หัวเรื่องด้วยกันคือ

1. การใช้งานระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ ส่วนโครงสร้างข้อมูล
2. การใช้งานส่วนตรวจสอบความถูกต้องของข้อมูลตามคอนสเตรนท์

6.1 การใช้งานระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ ส่วนโครงสร้างข้อมูล

การเรียกใช้งานระบบนี้จะมีไฟล์ชื่อ REDDG.EXE ซึ่งจะเป็น โปรแกรมระบบที่สามารถทำงานได้ โดยก่อนที่จะเรียกใช้โปรแกรมนี้จะต้องมีการเรียกใช้โปรแกรมระบบจัดการฐานข้อมูลออราเคิลก่อน และต้องมีชื่อผู้ใช้และรหัสผ่าน (Username and Password) นอกจากนี้ในฐานข้อมูลของผู้ที่มีสิทธิ์ใช้ในระบบจัดการฐานข้อมูลของออราเคิลจะต้องมีข้อมูลอยู่ในตารางเมต้า (Meta table) ด้วย

เมื่อเริ่มเรียกโปรแกรม REDDG.EXE แล้วจะปรากฏ โปรแกรมจะให้ผู้ใช้ป้อนชื่อผู้ใช้ และรหัสผ่าน ก่อนเข้าระบบที่แท้จริงของ REDDG.EXE เมื่อเข้าระบบแล้วจะมีพูลดาวน์ ปรากฏให้ผู้ใช้เลือกรายการต่าง ๆ ซึ่งมีอยู่ 6 ส่วนด้วยกัน คือ

1. ส่วนจัดการเกี่ยวกับไฟล์ (File)

รายการในส่วนนี้ประกอบไปด้วย

- ส่วนติดต่อกับโปรแกรมระบบฐานข้อมูลออราเคิล (User ID)

จะให้ผู้ใช้ป้อนชื่อผู้ใช้ และรหัสผ่านเพื่อติดต่อกับฐานข้อมูลออราเคิลที่มีข้อมูลเมต้าอยู่ด้วยซึ่งถือได้ว่าเป็นผู้ใช้ในที่นี้คือผู้ออกแบบระบบฐานข้อมูลนั่นเอง ซึ่งถ้าผู้ใช้ป้อนชื่อผู้ใช้และรหัสผ่านไม่ตรงกัน หรือป้อนผิด ระบบจะให้ผู้ใช้ป้อนใหม่เพิ่มอีก 2 ครั้ง (รวมแล้วป้อนได้ 3 ครั้ง) โดยถ้าป้อนถูกระบบจะอนุญาตให้ผู้ใช้เข้าไปใช้งานได้ แต่ถ้าป้อนชื่อผู้ใช้และรหัสผ่านครบ 3 ครั้งแล้วยังไม่ถูกอีกระบบก็จะออกจากไปสู่ดอสทันที

- ส่วนอ่านข้อมูลมาจากแผ่นดิสก์ (Load)

จะทำการอ่านข้อมูลออกจากแผ่นดิสก์โดยให้ผู้ใช้ป้อนชื่อไฟล์ที่ต้องการดึงออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-87-

มา ซึ่งไฟล์ที่จะดึงออกมาจะต้องมีการจัดเรียงข้อมูลตามแบบที่ระบบต้องการด้วย เมื่อนำข้อมูลออกมาแล้วก็จะนำข้อมูลมาสร้างเป็นลิงก์ลิสต์ของตารางและแสดงออกจากจอภาพทันที

- ส่วนเก็บข้อมูลลงในแผ่นดิสก์ (Save)

จะทำการเก็บข้อมูลของลิงก์ลิสต์ลงในไฟล์ที่ผู้ใช้จะเป็นผู้ตั้ง โดยจะจัดเก็บข้อมูลตามแบบที่ระบบจัดการ (มีฟอร์แมตการเก็บที่แน่นอน)

- ส่วนออกจากระบบไปทำงานในดอส (Dosshell)

ระบบจะหยุดการทำงานชั่วคราวโดยจะเก็บสภาวะที่กำลังทำอยู่ แล้วไปเรียกใช้โปรแกรมจัดการระบบ ซึ่งก็คือโปรแกรมคอสนั่นเอง และเมื่อต้องการกลับมายังระบบอีกที่โดยการพิมพ์คำว่า "EXIT" ระบบก็จะกลับมายังสภาวะเดิมก่อนไปทำงานในโปรแกรมดอส

- ส่วนออกจากระบบ (Quit)

ระบบจะเลิกการทำงานทั้งหมดและออกจากระบบทันที

2. ส่วนจัดการเกี่ยวกับกระบวนการแปลง (Map)

- ส่วนกระบวนการแปลง (Map)

เป็นกระบวนการแปลงจากข้อมูลในเมต้า และข้อมูลในแมปปิ้งอินฟอร์เมชันเป็นความสัมพันธ์ เมื่อผ่านกระบวนการแปลงแล้วก็จะแสดงผลที่ได้ออกทางจอภาพในหน้าต่างส่วนแก้ไข (Edit) โดยข้อมูลในเมต้าและข้อมูลในแมปปิ้งอินฟอร์เมชันจะเป็นข้อมูลของผู้ใช้ที่ติดต่ออยู่กับตัวจัดการฐานข้อมูลขณะนั้น ซึ่งก็คือผู้ออกแบบรูปไนเอมนั่นเอง

- ส่วนแก้ไขลิงก์ลิสต์ที่ได้จากการแปลง (Edit)

เมื่อผ่านกระบวนการแปลงแล้วจะมีรูปความสัมพันธ์ของผู้ใช้แสดงอยู่ในหน้าต่างส่วนแก้ไข และถ้าเลือกใช้ฟังก์ชันนี้ระบบก็จะอนุญาตให้ผู้ใช้เข้าไปแก้ไขได้ 2 ชนิด คือ เปลี่ยนชื่อตาราง และเปลี่ยนชื่อของคอลัมน์ ที่แสดงที่หน้าจอภาพ

- ส่วนเก็บ เปลี่ยนแปลง และแก้ไขข้อมูลในแมปปิ้งอินฟอเมชัน (Assert MI)

จะเป็นส่วนที่เก็บข้อมูลบางส่วนที่ได้จากการความสัมพันธ์ที่ได้จากกระบวนการแปลงแล้ว ลงในส่วนของแมปปิ้งอินฟอร์เมชัน เพื่อที่จะนำไปใช้งานต่อในส่วนของกลุ่มการพัฒนากฎในฐานความรู้ (Rule and Constraint in a knowledge base systemp) อีกทีหนึ่ง

- ส่วนสร้างตารางที่ได้จากการแปลงให้กับผู้ใช้ใด ๆ (Create Table)

ฟังก์ชันนี้จะทำได้ก็ต่อเมื่อความสัมพันธ์ที่แสดงในหน้าต่างส่วนแก้ไข ตรงกันกับที่ข้อมูลในเมต้า และข้อมูลในแมปปิ้งอินฟอร์เมชัน ถ้าไม่ตรงกันก็จะไม่สามารถกระทำในฟังก์ชันนี้ได้ โดยประปรากฏหน้าต่างแสดงข้อความที่ผิดพลาด (Error message) คือ ไม่สามารถหาชนิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของข้อมูลในคอลัมน์ของความสัมพันธ์ที่แปลงมาได้ แต่ถ้าข้อมูลในเมต้าและแมปปิ้งอินฟอร์เมชันตรงกันกับความสัมพันธ์ที่เสดงในหน้าต่างส่วนแก้ไขก็สามารถสร้างตารางได้ โดยจะให้ผู้ใช้ปัจจุบันหรือผู้ออกแบบ เลือกสร้างตารางให้กับผู้ใช้คนใดก็ได้ แต่ก็จะต้องรู้รหัสผ่านของผู้ใช้คนนั้นด้วย

3. ส่วนเซ็ตค่าแฟลกของขั้นตอนในการแปลงเอนติตี้สับไทป์ (Algorithms)

- เซ็ตแฟลกให้มีการเลือกแปลงแบบรวมสับไทป์เข้ากันกับซูเปอร์ไทป์ (Add-on)

ฟังก์ชันนี้จะเซ็ตแฟลกไว้คือ เมื่อทำกระบวนการแปลงถ้าระบบพบสับไทป์ทุกสับไทป์ ก็จะรวมสับไทป์ให้เข้ากันกับซูเปอร์ไทป์ของสับไทป์นั้น ๆ แล้วทำการแปลง

- เซ็ตแฟลกให้มีการเลือกแปลงแบบ แยกสับไทป์ออกจากซูเปอร์ไทป์ (Distributed)

ฟังก์ชันนี้จะเซ็ตแฟลกไว้คือ เมื่อทำกระบวนการแปลงถ้าระบบพบสับไทป์ทุกสับไทป์ ก็จะแยกสับไทป์นั้นออกจากซูเปอร์ไทป์ของสับไทป์นั้น ๆ แล้วทำการแปลง

- เซ็ตแฟลกให้ผู้ออกแบบเลือกว่าในแต่ละซูเปอร์ไทป์จะเลือกการแปลงแบบใด (User-defined)

ฟังก์ชันนี้จะเซ็ตแฟลกไว้คือ เมื่อทำกระบวนการแปลงระบบจะหาซูปเปอร์ไทป์ทุกตัว และนำมาถ้าผู้ใช้ให้ผู้ใช้เลือกว่าจะให้แปลงสำหรับแต่ละซูเปอร์ไทป์เป็นแบบรวมสับไทป์หรือแยกซูปเปอร์ไทป์

4. ส่วนการสร้างรายงานผลที่ได้เมื่อผ่านกระบวนการแปลงแล้ว (Report)

- สร้างรายงานเป็นภาษาเอสคิวเอล (From SQL)

จากความสัมพันธ์ที่ได้จากกระบวนการแปลงเมื่อเลือกฟังก์ชันนี้ จะให้ผู้ใช้ป้อนชื่อไฟล์สำหรับเก็บข้อมูลเป็นภาษาเอสคิวเอล ซึ่งสามารถนำไปใช้หลังจากออกจากโปรแกรมระบบแล้ว

- สร้างรายงานเป็นแบบต่าง ๆ (From Table)

จากความสัมพันธ์ที่ได้จากกระบวนการแปลงเมื่อเลือกฟังก์ชันนี้ จะให้ผู้ใช้ป้อนชื่อไฟล์สำหรับเก็บรูปภาพตารางที่เสดงบนหน้าจอภาพไว้ ซึ่งสามารถนำไปพิมพ์ออกทางเครื่องพิมพ์ได้ ภายหลังโดยใช้คำสั่ง TYPE ตามตัวชื่อไฟล์ในโปรแกรมระบบปฏิบัติการดอสได้

5. ส่วนบอกรายละเอียดของระบบ (About)

เป็นส่วนที่บอกชื่อของระบบเป็นภาษาอังกฤษ ผู้พัฒนาระบบ และปี ค.ศ.ที่ทำเสร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ส่วนบอกรายละเอียดของโครงงาน (Help)

เป็นส่วนที่บอกถึงการรายละเอียดของระบบ ว่าระบบนี้มีไว้เพื่ออะไร และใช้ทำอะไรได้บ้าง

6.2 การใช้งานส่วนตรวจสอบความถูกต้องของข้อมูลตามคอนสเตรนท์

การใช้งานส่วนตรวจสอบความถูกต้องของข้อมูลตามคอนสเตรนท์ แบ่งออกเป็น 2 ส่วน คือ

1. ส่วนที่ทำหน้าที่ในการสร้างคิวรี่ เพื่อใช้ในการตรวจสอบความถูกต้องของข้อมูล โดยในส่วนนี้ได้มีการพัฒนาให้เป็นฟังก์ชัน เพื่อเรียกใช้ได้ โดยเรียกเพียงชื่อฟังก์ชันเท่านั้น จะทำการสร้างคิวรี่ตามคอนสเตรนท์แต่ละประเภทตามแต่ละฟังก์ชัน จนหมดคอนสเตรนท์ประเภทนั้น ๆ เนื่องจากเป็นส่วนที่ทำขึ้นเพื่อให้สามารถใช้งานร่วมกับโครงงานการพัฒนากฎในระบบฐานความรู้ และผลลัพธ์ที่ได้ซึ่งเป็นค่าที่ส่งกลับไปยังรูทีนที่เรียกนั้นมีด้วยกัน 2 ค่า คือ ค่า 0 แสดงว่าสามารถสร้างได้ แต่ถ้าเป็น -1 แสดงว่าไม่สามารถสร้างคิวรี่ได้ โดยที่ผู้ที่ใช้ฟังก์ชันนี้จะต้องสามารถติดต่อนำข้อมูลในตารางเมต้ามาใช้ได้

2. ส่วนที่ทำหน้าที่ในการตรวจสอบความถูกต้องของข้อมูล ตามคิวรี่ที่สร้างขึ้นได้ตามประเภทของคอนสเตรนท์และตามเลขคอนสเตรนท์ โดยมีลักษณะเป็นฟังก์ชันที่เรียกใช้ได้ ซึ่งจะทำการรันตามคิวรี่ ถ้าไม่มีข้อมูลที่ขัดแย้งกับคอนสเตรนท์นั้นๆ ก็จะส่งค่า 0 กลับ ซึ่งอาจจะมีหลายคิวรี่แต่คอนสเตรนท์หนึ่งตัว และจะส่งค่า -1 กลับ ถ้ามีข้อมูลที่ขัดแย้งกับคอนสเตรนท์

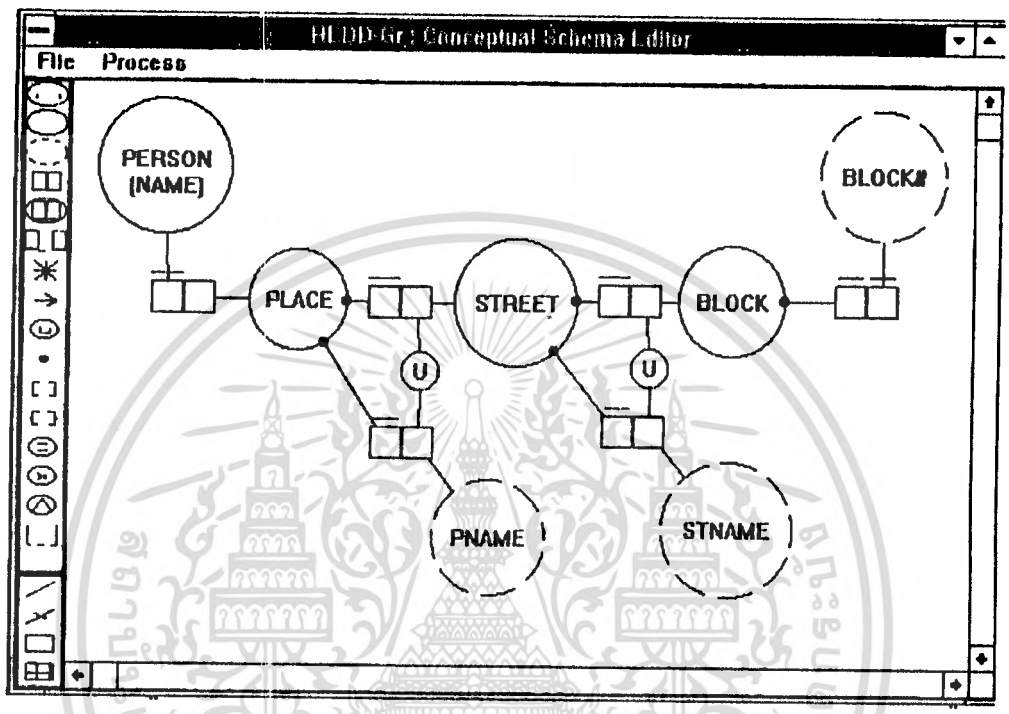
6.3 ตัวอย่างของระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ ส่วนโครงสร้างข้อมูล

ตัวอย่างในส่วนการใช้งานระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ ส่วนโครงสร้างข้อมูลนี้ตัวอย่างจะต้องผ่านโปรแกรมส่วนของระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ ก่อนซึ่งจะเก็บข้อมูลในตารางเมต้า และ ตารางแมปปิ้งอินฟอร์เมชัน จากนั้นระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ ส่วนโครงสร้างข้อมูลก็จะนำมาแปลงและแสดงผล ซึ่งตัวอย่างจะแสดงรูปไนเอม และผลที่ได้เป็นรูปตาราง และ ผลที่เป็นภาษาเอสคิวแอล ซึ่งจะมีตัวอย่างดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 1

แสดงรูปไนแอมดังรูป 6.1



รูปที่ 6.1 ตัวอย่างไนแอมรูปที่ 1

จากรูปไนแอมรูปที่ 6.1 เมื่อผ่านกระบวนการแปลงแล้วรายงานที่ได้มี 2 แบบ ได้แก่ รายงานเป็นรูปตาราง คือ ตาราง 6.1

TABLE T1

F1R1_NAME	F1R2_PNAM	F1R2_STNA	F1R2_BLOC

ตาราง 6.1 แสดงตารางในตัวอย่างที่ 1

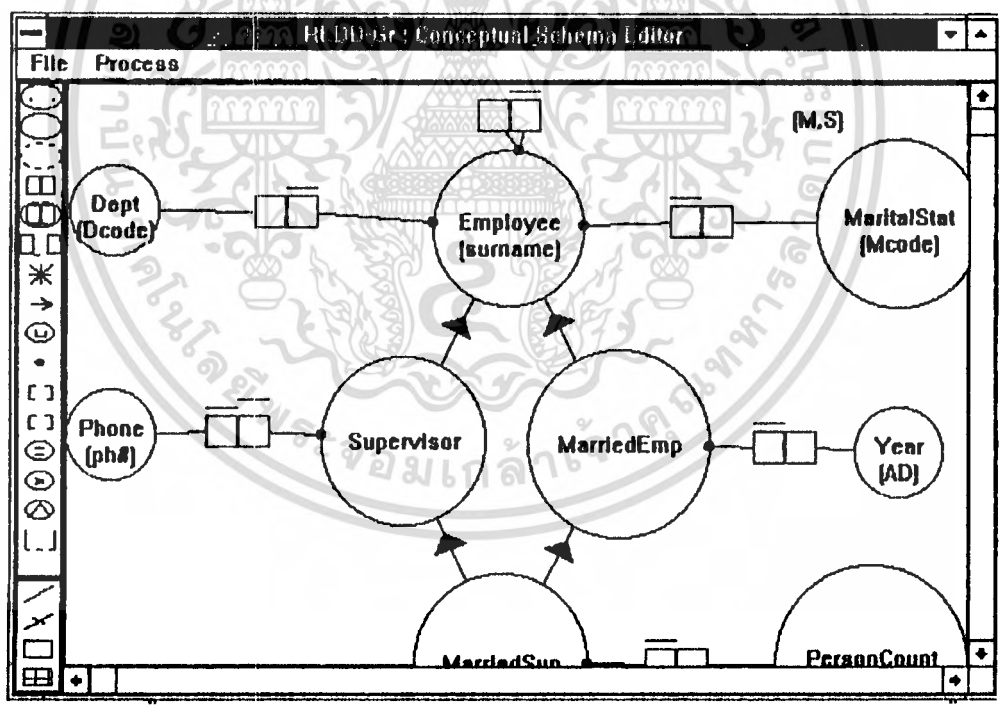
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายงานเป็นภาษาเอสคิวเอล

```
CREATE TABLE T1 (
        F1R1_NAME           CHAR(20),
        F1R2_PNAME          CHAR(20),
        F1R2_STNAME         CHAR(20),
        F1R2_BLOCK#         CHAR(20) );
```

ตัวอย่างที่ 2

แสดงรูปไนแอมดังรูป 6.2



รูปที่ 6.2 ตัวอย่างไนแอมรูปที่ 2

จากรูปไนแอมรูปที่ 6.2 เมื่อผ่านกระบวนการแปลงแล้วรายงานที่ได้มี 2 แบบ ได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>รายงานเป็นรูปตาราง</u> คือ ตาราง 6.2

TABLE T1

F1R2_SURN	F1R1_SURN	F2R1_DCOD	F4R2_MCOD

TABLE T2

F5R1_SURN	F5R2_AD

TABLE T3

F6R1_SURN	F6R2_NR

TABLE T4

F3R1_PH#	F3R2_SURN

ตาราง 6.2 แสดงตารางในตัวอย่างที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

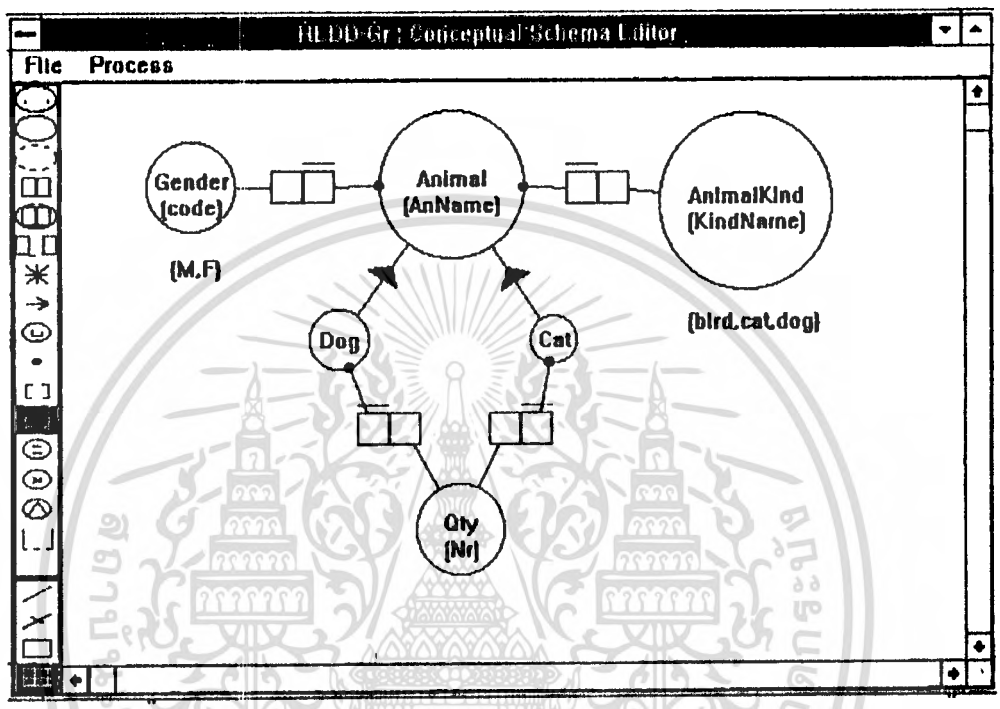
รายงานเป็นภาษาเอสคิวเอล

```
CREATE TABLE T1 (
        F1R2_SURNAME        CHAR(20),
        F1R1_SURNAME        CHAR(20),
        F2R1_DCODE          CHAR(20),
        F4R2_MCODE          CHAR(20) );

CREATE TABLE T2 (
        F5R1_SURNAME        CHAR(20),
        F5R2_AD             CHAR(20) );

CREATE TABLE T3 (
        F6R1_SURNAME        CHAR(20),
        F6R2_NR             CHAR(20) );

CREATE TABLE T4 (
        F3R1_PH#            CHAR(20),
        F3R2_SURNAME        CHAR(20) );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 3

แสดงรูปไนเอมดังรูป 6.3



รูปที่ 6.3 ตัวอย่างไนเอมรูปที่ 3

จากรูปไนเอมรูปที่ 6.3 เมื่อผ่านกระบวนการแปลงแล้วรายงานที่ได้มี 2 แบบ ได้แก่

รายงานรูปตารางแบบที่ 1 คือ ตาราง 6.3.1 เป็นการแปลงแบบรวมสับไทป์รวมเข้ากับซูเปอร์ไทป์

TABLE T1

F1R2_ANAM	F1R1_CODE	F2R2_KNAM	F3R2_NR	F4R1_NR

ตาราง 6.3.1 แสดงตารางในตัวอย่างที่ 3 แบบที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายงานเป็นภาษาเอสคิวเอล แบบที่ 1 เป็นรายงานภาษาเอสคิวแอล ที่เกิดจากการแปลงแบบรวมสับไทป์รวมเข้ากับซูเปอร์ไทป์

```
CREATE TABLE T1 (
        F1R2_ANAME          CHAR(20),
        F1R1_CODE           CHAR(20),
        F2R2_KNAME          CHAR(20),
        F3R2_NR             CHAR(20),
        F4R1_NR             CHAR(20) );
```

รายงานรูปตารางแบบที่ 2 คือ ตาราง 6.3.2 เป็นการแปลงแบบรวมสับไทป์แยกกันกับซูเปอร์ไทป์

TABLE T1

F1R2_ANAM	F1R1_CODE	F2R2_KNAM

TABLE T2

F4R2_ANAM	F4R1_NR

TABLE T3

F3R1_ANAM	F3R2_NR

ตาราง 6.3.2 แสดงตารางในตัวอย่างที่ 3 แบบที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายงานเป็นภาษาเอสคิวเอล แบบที่ 2 เป็นรายงานภาษาเอสคิวแอล ที่เกิดจากการแปลงแบบแยกสับไทป์กับซูเปอร์ไทป์

```
CREATE TABLE T1 (
        F1R2_ANAME          CHAR(20),
        F1R1_CODE           CHAR(20),
        F2R2JKNAME          CHAR(20) );

CREATE TABLE T2 (
        F4R2_ANAME          CHAR(20),
        F4R1_NR             CHAR(20) );

CREATE TABLE T3 (
        F3R1_ANAME          CHAR(20),
        F3R2_NR             CHAR(20) );
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
บทสรุปและวิจารณ์

โครงงานระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ ส่วนโครงสร้างข้อมูล เป็นการพัฒนาโปรแกรมเพื่อให้สามารถทำการเชื่อมต่อกับส่วนออกแบบ ในการนำข้อมูลจากส่วนออกแบบมาใช้งาน ซึ่งสามารถใช้งานและเชื่อมต่อกันได้ แต่ก็ยังทำงานบนฮาร์ดแวร์ที่ได้กำหนดไว้ ซึ่งสามารถปรับปรุงต่อไปในการย้ายโปรแกรมไปบนเครื่องที่มีเอนไวรอนเมนต์ต่างๆ เพื่อให้สามารถทำงานได้ทุกเอนไวรอนเมนต์

จากการวิเคราะห์ระบบการออกแบบฐานข้อมูลแบบสัมพันธ์ส่วนโครงสร้างข้อมูล สามารถสรุปปัญหาต่างๆที่เกิดขึ้นได้ดังนี้

โครงงานระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ แบ่งออกเป็น 2 ส่วนดังที่ได้กล่าวมาแล้ว คือ ส่วนออกแบบ และ ส่วนโครงสร้างข้อมูล ซึ่งเป็นงานของโครงงานนี้ จะเห็นว่าต้องมีการติดต่อสื่อสารทำความเข้าใจระหว่าง 2 กลุ่ม ซึ่งจะพบอุปสรรคต่างๆ เช่น

- ปัญหาในการออกแบบที่จะต้องอาศัยโครงสร้างหลักเหมือนกัน จะพบว่า ในระหว่างการพัฒนา อาจจำเป็นจะต้องทำการแก้ไขโครงสร้างเพื่อให้ง่าย และเพื่อให้กระชับมากที่สุด ถ้าเกิดการแก้ไขในอีกกลุ่มหนึ่ง ก็จำเป็นที่อีกกลุ่มหนึ่งจะต้องทำการแก้ไขด้วย ซึ่งก็ต้องมีการแก้ไขในส่วนที่ได้ทำการพัฒนาไปแล้วด้วย จึงเกิดการยุ่งยาก เพราะจะต้องทำการแก้ไขอยู่เรื่อยๆ

- ปัญหาจากการพัฒนาโครงงาน เนื่องจากแบบจำลองข้อมูลไนเอมนั้น สามารถเขียนหรือกำหนดความสัมพันธ์ต่าง ๆ ที่ไม่ผิดความหมายได้มากกมาย ซึ่งในระหว่างการพัฒนาอาจจะมีบางรูปแบบที่ไม่ได้พิจารณา แต่เมื่อได้พัฒนาเสร็จเรียบร้อยแล้ว กลับพบว่า ยังมีบางกรณีที่ยังไม่ครอบคลุม ก็ต้องทำการแก้ไขโปรแกรมกันใหม่ ซึ่งจะพบได้บ่อย

- ปัญหาทางด้านเครื่องมือ จะพบว่าจากการกำหนดเอนไวรอนเมนต์ จำเป็นจะต้องใช้ฮาร์ดแวร์ที่มีคุณสมบัติตามที่กำหนด ซึ่งเป็นเอนไวรอนเมนต์ที่ไม่ค่อยพบโดยปกติในเครื่องคอมพิวเตอร์ทั่วไป นอกจากนี้การพัฒนาโครงงานนี้มีหลายส่วน ซึ่งจะต้องแยกทำการพัฒนา แต่ไม่สามารถหาเครื่องมือมาช่วยในการพัฒนาได้ งานจึงดำเนินไปได้ค่อนข้างจำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8
แนวทางการพัฒนาต่อไป
(Further Research)

ในบทนี้จะได้กล่าวถึง สิ่งที่ได้มีการกำหนดในเบื้องต้นว่าจะทำ แต่เนื่องจากข้อจำกัดทางเวลา ทำให้ไม่สามารถทำได้ตามความประสงค์ จึงบรรจุสิ่งเหล่านี้ไว้เป็นแนวทางในการพัฒนาต่อสำหรับผู้พัฒนาโครงงานนี้ต่อไป ได้แก่

1. **รวมเอาโครงงานระบบช่วยออกแบบ ฯ ในส่วนออกแบบ และ ส่วนโครงสร้าง ให้อยู่ในเอนไวรอนเมนต์แบบกราฟฟิกเดียวกัน โดยเรียกใช้ผ่านเมนูของระบบรวม**

เนื่องจากระบบช่วยออกแบบฐานข้อมูลแบบสัมพันธ์ เป็นระบบงานรวมซึ่งต้องดำรงสภาพแวดล้อมแบบกราฟฟิก การเรียกใช้โมดูลการแปลงนิยามข้อมูล โมดูลวิเคราะห์แบบจำลอง และโมดูลสร้างฐานข้อมูลของกลุ่มโครงสร้าง จะกระทำผ่านเมนูเรียกใช้ในสภาพแวดล้อมของระบบรวม และการแสดงผลของโมดูลเหล่านี้ ควรจะอยู่ในรูปแบบกราฟฟิกด้วยเพื่อความเป็นอันหนึ่งอันเดียวกัน

2. **ให้สามารถรองรับกฎความถูกต้อง (Integrity constraint) ของฐานข้อมูล อันได้แก่ กฎความถูกต้องของเอนติตี้ Entity Integrity และกฎความถูกต้องในการอ้างถึง (Referencial integrity) ได้**

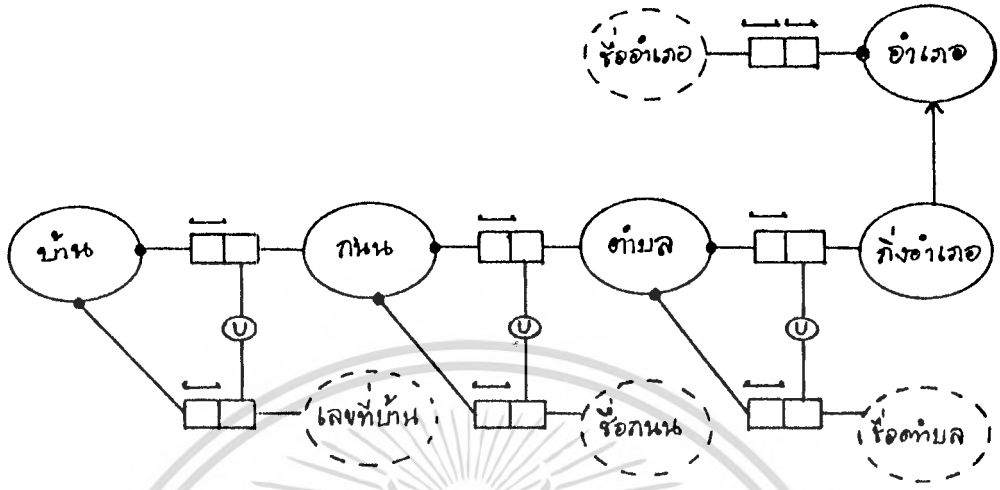
กฎความถูกต้องถือเป็นเงื่อนไขหลักข้อหนึ่ง ที่ใช้ในการตัดสินว่าระบบจัดการฐานข้อมูลนั้นเป็นระบบแบบสัมพันธ์หรือไม่ การกำหนดไพรมารี่คีย์บนรีเลชั่นที่ได้จากโมดูลการสร้างรีเลชันต้องเป็นไปตามกฎความถูกต้องของเอนติตี้ และเมื่อมีการเปลี่ยนแปลงแก้ไขค่าข้อมูลที่เป็นฟอร์เรนคีย์ ระบบจะต้องดูแลความถูกต้องให้กับไพรมารี่คีย์ของรีเลชันที่เกี่ยวข้อง ตามกฎความถูกต้องในการอ้างถึงด้วย

3. **ให้สามารถรองรับและสามารถแมปปิ้งไปแอบที่มีเอนติตี้ไทป์ที่มีการอ้างถึงหรือเนสติ้งแบบไฮราคี่ที่มีสับไทป์ได้**

ในการอ้างถึงของเอนติตี้ไทป์แบบไฮราคี่นั้น ได้มีตัวอย่างแล้วในรูปที่ 5.5 แต่ในกรณีที่จะกล่าวถึงต่อไปนี้ เป็นกรณีพิเศษของการอ้างถึงแบบไฮราคี่ ซึ่งเรียกว่าไฮราคี่ที่มีสับไทป์ดัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงในรูปที่ 8.1



รูปที่ 8.1 การอ้างถึงแบบไฮราคี่แบบมีสับไทป์

4. สามารถตรวจสอบหรือวิเคราะห์ความถูกต้องเชิงความหมายหรือ semantic analysis ของแบบจำลองข้อมูลในเอนทองระบบใช้งานฐานข้อมูลได้ โดยวิธีตรวจสอบโดยกำหนดประชากร (Population checking)

เป็นวิธีตรวจสอบความถูกต้องเชิงความหมายที่ค่อนข้างง่าย โดยจะทำการตรวจสอบความสัมพันธ์ที่เป็นแฟกท์ไทป์แบบเทอร์นารี่ขึ้นไป โดยการสุ่มประชากรตามคอนสเตรนท์ที่กำหนดอยู่แล้วทำการแยกเทอร์นารี่นั้นออกมา เป็น 2 รีเลชัน หากผลลัพธ์ที่ได้จากการรวม (join) ระหว่าง 2 รีเลชันดังกล่าวเหมือนกับรีเลชันต้นแบบ ก็ถือว่าไม่สามารถแยก (split) รีเลชันที่เป็นเทอร์นารี่นี้ออกได้ แต่หากไม่ใช่ จะสามารถแยกออกได้เป็นไบนารี่ 2 รีเลชัน

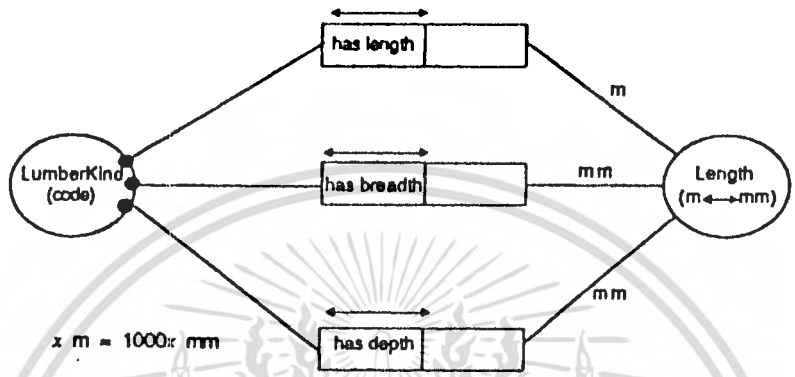
5. ให้สามารถรองรับไนเอนที่มีแฟกท์ไทป์ชนิดหาได้ (Derived fact type)

ในขณะนี้ส่วนเมต้าสกีมม่าของโครงงานได้สร้างขึ้นรองรับแฟกท์ไทป์ชนิดหาได้อยู่แล้ว เหลือแต่เพียงการพัฒนากระบวนการ ที่ใช้ในการประมวลผลสูตรที่ใช้ในการหาค่าหรือ Derived formula เพื่อให้ได้ค่าข้อมูลตามที่ต้องการ สูตรการหานี้เป็นได้ตั้งแต่ สูตรคำนวณธรรมดาจนถึงภาษา SQL ที่ใช้ในการสร้างวิวลักษณะต่าง ๆ ตามที่ตัวโปรแกรมจะกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ให้สามารถรองรับไนแอมที่มีเรฟเฟอร์เรนซ์ไทป์ชนิดหาได้ (Derived reference type)

ลักษณะของเรฟเฟอร์เรนซ์ไทป์แบบหาได้ แสดงดังรูปที่ 8.2



รูปที่ 8.2 เรฟเฟอร์เรนซ์ไทป์แบบหาได้

จากรูปแบบจำลอง LumberKind มีความสัมพันธ์แบบแฟกท์ไทป์ 3 ความสัมพันธ์ โดยกระทำกับ Length ซึ่งมีการอ้างถึงด้วยหน่วยความยาว 2 หน่วย คือ เมตรกับมิลลิเมตร ทั้งนี้ที่แฟกท์ไทป์จะต้องกำหนดว่าจะใช้การอ้างถึงในหน่วยใด กรณีเช่นนี้จะเห็นได้ว่าเรฟเฟอร์เรนซ์ไทป์ที่แฝงอยู่ในเอนติตี้ไทป์ Length มีอยู่เพียงหนึ่งเดียวคือ มิลลิเมตร โดยที่จะสามารถหาการอ้างถึงในหน่วยเมตร ได้จากการคำนวณสูตร เมตร = 1000 * มิลลิเมตร เรียกเรฟเฟอร์เรนซ์ไทป์ลักษณะนี้ว่า เรฟเฟอร์เรนซ์ไทป์แบบหาได้ (Derived reference type)

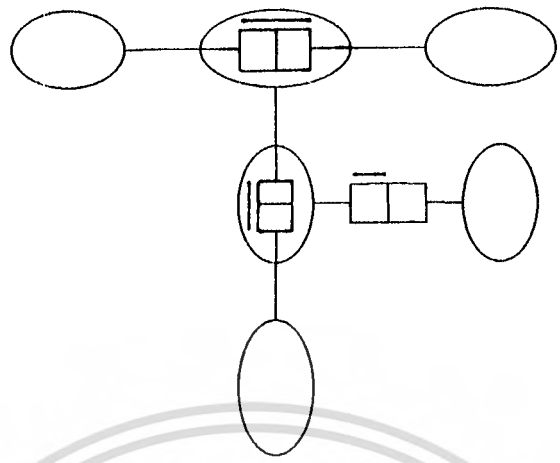
7. ให้สามารถรองรับเลเบิ้ลไทป์ที่มีชนิดของข้อมูลได้มากกว่าที่เป็นอยู่

โครงงานนี้ครอบคลุมเพียงแค่ชนิดข้อมูล 2 ประเภทคือ สตริง (character) และ จำนวน (number) ซึ่งในทางปฏิบัติควรมีมากกว่าที่เป็นอยู่

8. ให้สามารถรองรับไนแอมที่มีเนสท์แฟกท์ไทป์ได้มากกว่า 1 ชั้น (Nest fact type)

โครงงานนี้ครอบคลุมเพียงเนสท์แฟกท์ไทป์เพียงชั้นเดียวเท่านั้น ซึ่งต่อไปควรให้ไนแอมสามารถรอบรับเนสท์แฟกท์ไทป์ได้มากกว่า 1 ชั้นซึ่งจะมีผลทำให้การออกแบบฐานข้อมูลมีขนาดเล็กลงไปด้วย ตัวอย่างเนสท์แฟกท์ไทป์ที่มี 2 ชั้นแสดงดังรูป 7.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 8.3 แสดงเนสท์ 2 ชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อ้างอิง

[1] G.M.Nijssen and T.A.Halpin , "Conceptual Schema and Relational Database Design A Fact Oriented Approach", PRENTICE HALL , 1989.

[2] C.J.Date , "An Introduction to Database Systems" , Addison-Wesley, 1986.

[3] ORACLE , "PRO*C USER'S GUIDE" , ORACLE , 1987.

[4] James T.Perry and Joseph G.Latter , "UNDERSTRANDING ORACLE" , TECH PUBLICATIONS , 1989.

[5] N.Prabhakaran , "GENERATOR OF RELATIONAL DATABASE SCHEMATA AND ITS APPLICATIONS" , Department of Computer Science Univercity of Queensland , 1984.

[6] C.M.R.Leung and G.M.Nijssen , "From a NIAM Conceptual Schema into the optimal SQL Relational Database Schema" , Department of Computer Science Univercity of Queensland , 1987.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
โปรสตาร์ซี (Pro*c)

ระบบจัดการฐานข้อมูลแบบสัมพันธ์ของออราเคิล (ORACLE Relational Database Management System) ได้ทำการสร้างโปรแกรมที่ใช้ในการเปลี่ยนประโยคคำสั่งของเอสคิวแอลที่ประกอบอยู่ในโปรแกรมภาษาซี ซึ่งจะนำหน้าด้วย "EXEC SQL" ให้เป็นประโยคคำสั่งในภาษาซี ซึ่งสามารถจัดการฐานข้อมูลของออราเคิลได้ และเรียกวิธีนี้ว่า "การพรีคอมไพล์" (Precompiler) และจากนั้นจึงนำผลที่ได้ มาทำการคอมไพล์หรือลิงก์ โดยใช้คอมไพเลอร์ของภาษาสูงที่ใช้ เพื่อให้ได้โปรแกรมที่สามารถปฏิบัติการได้ ซึ่งโปรแกรมที่ใช้ในการพรีคอมไพล์นี้คือ "โปรสตาร์ซีทูล" (Pro*c Tool)

ดังนั้น เราสามารถสรุปขั้นตอนในการเขียนโปรแกรมโปรสตาร์ซี ได้ดังนี้

1. เขียนโปรแกรมโปรสตาร์ซี
2. ทำการพรีคอมไพล์ โดยใช้โปรสตาร์ซีทูล
3. ทำการคอมไพล์แฟ้มข้อมูลที่ได้จากการคอมไพล์
4. ทำการลิงค์แฟ้มข้อมูลที่เกี่ยวข้องกัน
5. รันโปรแกรม

1. ประโยคคำสั่งของเอสคิวแอลในการปฏิบัติการและการกำหนด (Execute and Declarative SQL Statements)

ประโยคคำสั่งของเอสคิวแอลที่ใช้ในการโปรแกรมโปรสตาร์ซี สามารถแบ่งได้ 2 ประเภทคือ

1. ประโยคคำสั่งปฏิบัติการ คือคำสั่งที่เกี่ยวกับการจัดการข้อมูล (Data Manipulation Language) , การกำหนดข้อมูล (Data Definition Language) และการควบคุมข้อมูล (Data Control Language) ซึ่งการปฏิบัติการของประโยค คำสั่งเหล่านี้จะมีผลต่อส่วนการติดต่อข้อมูลของเอสคิวแอล (SQL Communication Area) หรือเรียกอีกอย่างว่าเอสคิวแอลซีเอ (SQLCA) โดยจะมีการแสดงรหัสสถานะที่ได้จากผลของประโยคคำสั่งการปฏิบัติการที่ส่วนนี้ นอกจากนี้ ยังมีผลต่อการเริ่มต้น หรือการสิ้นสุดของส่วนการเก็บการเรียงลำดับของประโยคคำสั่งเอสคิวแอล (Logical Unit of Work)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ประโยคคำสั่งการกำหนด คือ ประโยคคำสั่งที่จะไม่มีการแสดงรหัสสถานะของการปฏิบัติที่ส่วนการติดต่อข้อมูลของเอสคิวแอล และส่วนการเก็บการเรียงลำดับของประโยคคำสั่งของเอสคิวแอล

2. ส่วนประกอบในโปรแกรมโปรสตาร์ซี (Parts of the Pro*C Program)

โปรแกรมโปรสตาร์ซี สามารถแบ่งได้เป็น 2 ส่วน คือ

1. ส่วนโครงสร้างการประยุกต์ (Application Prologue) คือส่วนกำหนดตัวแปรที่ใช้ในการติดต่อกับฐานข้อมูล , การเตรียมพร้อม และการเริ่มต้นเพื่อที่จะติดต่อกับฐานข้อมูล

2. ส่วนตัวโปรแกรมประยุกต์ (Application body) คือ ส่วนของโปรแกรมที่ประโยคคำสั่งต่างๆของเอสคิวแอล ในการจัดการฐานข้อมูลปรากฏอยู่

3. ส่วนโครงสร้างการประยุกต์

สามารถแบ่งออกได้เป็น 3 ส่วน คือ

1. ส่วนกำหนดตัวแปร (DECLARE section) คือ ส่วนที่ใช้กำหนดตัวแปรและชนิดในภาษาที่ที่จะใช้ในการติดต่อกับฐานข้อมูล มีรูปแบบในการใช้โดยจะเริ่มต้นด้วยประโยค

```
EXEC SQL BEGIN DECLARE SECTION;
```

และจบด้วยประโยค

```
EXEC SQL END DECLARE SECTION;
```

ระหว่างประโยคทั้ง 2 จะเป็นการกำหนดตัวแปรและชนิดของตัวแปร ซึ่งตัวแปรที่กำหนดในส่วนนี้สามารถแบ่งได้เป็น ตัวแปรหลัก (Host Vaiable) และตัวแปรบ่งชี้ (Indicator Variable) ตัวแปรหลักเป็นตัวแปรที่เก็บค่าข้อมูลต่างๆที่ใช้ในการติดต่อกับฐานข้อมูล ส่วนตัวแปรบ่งชี้จะใช้ในการเพิ่มข้อมูลในคอลัมน์ที่ไม่ทราบค่า (NULL) และสามารถแสดงให้ทราบว่าข้อมูลที่ตัวแปรหลักได้รับจากฐานข้อมูลมีลักษณะอย่างไร เช่น ไม่ทราบค่า หรือ มีการตัดข้อมูลส่วนทั้งเป็นต้น

สำหรับชนิดของตัวแปรหลักนั้น จะเหมือนกับชนิดของตัวแปรในภาษาซี แต่ที่เพิ่มขึ้นมาคือประเภท วาร์ชาร์ (VARCHAR) ซึ่งมีลักษณะเป็นแบบเรคอร์ด (Record)

สำหรับข้อผิดลาด (error) ที่เกิดขึ้นในส่วนการกำหนดตัวแปรนี้ ในกรณีที่ผู้ใช้มิได้กำหนดตัวแปรที่ใช้ในโปรแกรม แต่มีการเรียกใช้ จะมีการแสดงข้อความผิดพลาด คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Undeclared host variable <a> at line <b> in file <c>

2. INCLUDE SQLCA จะมีรูปแบบการใช้คือ

EXEC SQL INCLUDE SQLCA;

ประโยคคำสั่งนี้ จะเป็นการกำหนดให้โปรสตาร์ซี รวมส่วนของการติดต่อข้อมูลกับเอสคิวแอลเข้าไว้ในโปรแกรม โดยขณะที่ทำการพรีคอมไพล์ ออราเคิลจะทำหน้าที่เปลี่ยนหรือแทนที่ตัวแปรหลักในโปรแกรมด้วยตัวแปรที่ได้จากแฟ้มข้อมูลที่นำมารวม และหน้าที่สำคัญของการติดต่อข้อมูลกับเอสคิวแอล อีกอย่างหนึ่ง นอกเหนือจากการติดต่อกับออราเคิล ก็คือ การแสดงข้อผิดลาดและข้อระวังต่างๆ (Warning) ที่เกิดขึ้นในการปฏิบัติคำสั่งของเอสคิวแอลโดยจะแสดงในรูป

- sqlca.sqlcode : ค่ามากกว่า 0 จะแสดงถึงการกระทำคำสั่ง
  ค่าเท่ากับ 0 แสดงว่าทำคำสั่งได้สมบูรณ์
  ค่าน้อยกว่า 0 เกิดการผิดพลาดขึ้น

- sqlca.sqlwarn : จะประกอบด้วยอาร์เรย์ (array) ของแฟลกส์ (flags) 8 ตัว ซึ่งแต่ละตัวก็จะแสดงถึงลักษณะของข้อระวังที่แตกต่างกันออกไป

นอกจากจะสามารถเรียกใช้เอสคิวแอลได้แล้ว ยังสามารถใช้คำสั่งเฉพาะหรือติดต่อกับออราเคิลได้โดยตรง โดยใช้คำสั่ง

EXEC SQL INCLUDE ORACA;

3. การติดต่อกับออราเคิล (Connecting to ORACLE) มีรูปแบบการใช้คือ

EXEC SQL CONNECT (:oracleid> IDENTIFIED BY <:oraclepasswd>

หรือ

EXEC SQL CONNECT <:oracleid>

โดยที่ oracleid อยู่ใน <:oracleid>/<:oraclepasswd> จะเป็นส่วนที่ต้องใช้เพื่อให้โปรแกรมสามารถติดต่อกับระบบจัดการฐานข้อมูลแบบสัมพันธ์ของออราเคิลได้ ซึ่งจะทำให้สามารถเข้าถึงฐานข้อมูลของออราเคิลได้

4. ตัวโปรแกรม (Application Body)

เป็นส่วนที่ภาษาหลัก (Host) และภาษา Embeded รวมกันอยู่ ลักษณะโดยทั่วไปของโปรแกรม คือ

- ภาษาหลัก จะเป็นตัวจัดการเกี่ยวกับการแสดงผล (display) และรูปแบบการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่างๆของโปรแกรม เช่น เมนู เป็นต้น

- ภาษา Embeded จะทำงานในด้านการจัดการเกี่ยวกับข้อมูล รวมทั้งการเรียกใช้คำสั่งของเอสคิวแอลและออราเคิลด้วย ซึ่งการเรียกใช้นั้น จะต้องมี "EXEC SQL" นำหน้าก่อนเสมอ

5. การถามตอบกับเอมเบดด์ (Query)

เป็นส่วนหนึ่งในตัวโปรแกรม (Application Body) ซึ่งจะใช้ในการเรียกข้อมูลมาใช้ หรือ เก็บข้อมูลต่างๆ

สำหรับคำสั่งที่ใช้ในการสอบถาม (Query) จะประกอบด้วย

- SELECT	- INTO
- FROM	- WHERE
- CONNECT	- UNION
- INTERSECT	- MINUS
- GROUP BY	- HAVING
- ORDER BY	

สำหรับตัวแปรที่ใช้ในการสอบถามนั้น มาจาก 2 ที่ คือ จากตาราง (table) ในภาษาเอมเบดด์ และจากตัวแปรในภาษาหลัก ซึ่งตัวแปรในภาษาหลักที่ใช้ในการสอบถามจะต้องมีเครื่องหมาย ":" (colon) นำหน้าชื่อตัวแปรเสมอ

ลักษณะการสอบถาม มี 2 แบบ คือ

1.การสอบถามที่ให้ผลลัพธ์ออกมาเพียง 1 แถว (Query which return SINGLE ROW only) เป็นการสอบถามที่จะต้องอ้างกับค่าที่มีเพียง 1 แถวในตารางเท่านั้น (Unique index) ซึ่งถ้าให้ค่ามากกว่า1 แถวจะแสดงข้อผิดพลาดออกมา

2.การสอบถามที่ให้ผลลัพธ์มากกว่า 1 แถว (Query which return MULTIOLE ROWS) การสอบถามลักษณะนี้มักจะใช้กับการเรียกข้อมูลที่มีเป็นกลุ่มในตาราง ซึ่งเมื่อกระทำการสอบถามนี้แล้ว เอสคิวแอลจะให้ผลลัพธ์ ทั้งหมดออกมาในครั้งเดียว ดังนั้น การใช้การสอบถามแบบนี้จึงจำเป็นต้องเตรียมเนื้อที่ส่วนหนึ่งในออราเคิลหรือเอสคิวแอล เพื่อที่จะใช้ในการเก็บผลลัพธ์นั้นไว้ แล้วจึงเรียกออกมาใช้ตามที่ต้องการ ซึ่งพื้นที่นั้นเรียกว่า เคอร์เซอร์ (Cursor)

เคอร์เซอร์มีลักษณะการใช้ ดังนี้คือ

- การประกาศเคอร์เซอร์ (DECLARE CURSOR) เพื่อกำหนดพื้นที่ , ชื่อ และการสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถามที่ต้องการ

รูปแบบ : EXEC SQL DECLARE <cursorname> CURSOR FOR [Query];

- การเปิดเคอร์เซอร์ (OPEN CURSOR) เพื่อเปิดให้สามารถเรียกใช้เคอร์เซอร์ได้

รูปแบบ : EXEC SQL OPEN <cursorname>;

- การเฟทช์ (FETCH) ให้เคอร์เซอร์แสดงผลลัพธ์ตัวต่อไป

รูปแบบ : EXEC SQL FETCH <cursorname> INTO <HostVar>;

- การปิดเคอร์เซอร์ (CLOSE CURSOR) เป็นการยกเลิกเคอร์เซอร์ที่ระบุออกไป

รูปแบบ : EXEC SQL CLOSE <cursorname>;

- ตำแหน่งของเคอร์เซอร์ปัจจุบัน (CURRENT OF CURSOR) ใช้ในการอ้างถึงตำแหน่งแถวปัจจุบันในเคอร์เซอร์ที่มีการเฟทช์ข้อมูล เพื่อนำมาใช้ในการแก้ไขหรือลบข้อมูล ตำแหน่งของเคอร์เซอร์ปัจจุบันสามารถใช้แทนด้วย โรว์ไอดี (ROWID)

6. <u>คอมมิท และ โรลแบค</u> (Commit and Rollback)

ในการทำงานของโปรแกรม คำสั่งที่เป็นภาษาเอสคิวแอลแต่ละคำสั่ง จะถูกออราเคิลมองเป็นส่วนย่อย(logical unit of work) ซึ่งในแต่ละส่วนนี้ จะถูกประมวลผลเป็นลำดับขั้นไปจนจบ หรืออาจมีการถูกยกเลิกกลางคันก็ได้ สำหรับการยกเลิก unit of work นั้นเกิดได้จาก 2 กรณี คือ

1. ผู้ใช้ (user) ยกเลิกเอง

2. ระบบ (system) ไม่สามารถทำงานต่อไปได้ เช่น เกิดเดดล็อก (Deadlock) ขึ้น และการจบของ unit of work มี 2 แบบ คือ

- การคอมมิทเวอร์ก (Commit work) เป็นการจบ unit of work โดยให้ผ่านการเก็บการเปลี่ยนแปลงทั้งหมดที่เกิดขึ้นไว้ในฐานข้อมูล มีรูปแบบการใช้คือ

EXEC SQL COMMIT WORK [RELEASE];

โดยที่ทางเลือก RELEASE จะเป็นการคืนเนื้อที่ในหน่วยความจำทั้งหมดและออกจากระบบ (log off) ซึ่งจะเป็นการจบ unit of work สุดท้าย

- โรลแบคเวอร์ก (Rollback Work) เป็นการจบ unit of work เช่นกัน แต่จะทำการยกเลิกการแก้ไขข้อมูลทั้งหมด จะใช้ในกรณีที่เกิดการผิดพลาด ในการทำงานของโปรแกรม มีรูปแบบการใช้ลักษณะเดียวกับคอมมิท คือ

EXEC SQL ROLLBACK WORK [RELEASE]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. การแสดงความผิดพลาด (Error and Warning)

หน้าที่สำคัญอีกประการหนึ่งของส่วนติดต่อข้อมูลของเอสคิวแอล คือ เป็นส่วนที่จะกระทำเกี่ยวกับการแสดงความผิดพลาดของโปรแกรม หรือในส่วนต่างๆของเอสคิวแอล โดยที่ลักษณะของส่วนการติดต่อข้อมูลของเอสคิวแอล

8. การตรวจข้อผิดพลาด (Error Detection)

ในการตรวจสอบข้อผิดพลาด จะใช้คำสั่ง WHENEVER ซึ่งจะทำการตรวจสอบสถานะของส่วนการติดต่อข้อมูลของเอสคิวแอลทุกๆครั้งที่กระทำคำสั่งเอสคิวแอล มีรูปแบบการใช้ ดังนี้

```
EXEC SQL WHENEVER [SQLERROR]   [STOP]
                  [SQLWARNING] [CONTINUE]
                  [NOT FOUND]  [GOTO label]
```

โดยที่

- SQLERROR : จะถูก set เมื่อ sqlca.sqlcode เป็นลบ
- SQLWARNING : จะถูก set เมื่อ sqlca.sqlwarn[0] = "W"
- NOT FOUND : จะถูก set เมื่อ sqlca.sqlcode = +1403 (no row found)
- STOP : หยุดการทำงานของโปรแกรม และ โรลแบค
- CONTINUE : ทำงานต่อไป ไม่ว่า sqlca จะเป็นอย่างไร
- GOTO label : ข้ามไปทำงานที่ label

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

ระบบจัดการฐานข้อมูลแบบสัมพันธ์ออราเคิล

(ORACLE Relational Database Management System)

โดยทั่วไประบบจัดการฐานข้อมูลแบบสัมพันธ์ (RDBMS:Relational Database Management System) เป็นตัวเชื่อมต่อระหว่าง หน่วยเก็บข้อมูลแบบกายภาพ กับข้อมูลทางตรรกะ ในทางปฏิบัติ จะแบ่งออกเป็นกลุ่ม ๆ ตามลักษณะการกระทำกับข้อมูลบนเครื่องมือ (Tools) โดยสามารถใช้เครื่องมือเหล่านี้ในการทำงานดังต่อไปนี้

- ให้นิยามของฐานข้อมูล (Define a database)
- สอบถามข้อมูลจากฐานข้อมูล (Query tho database)
- เพิ่มเติม แก้ไข และลบข้อมูล (Add,edit and delete data)
- เปลี่ยนแปลงโครงสร้างของฐานข้อมูล (Modify the structure of the database)
- รักษาความปลอดภัยแก่ข้อมูลจากการใช้งาน (Secure data from public access)
- ติดต่อสื่อสารระหว่างระบบเครือข่าย (Communicate within networks)
- ส่งข้อมูลออก และรับข้อมูลเข้า (Export and import data)

เนื่องจากระบบจัดการฐานข้อมูลแบบสัมพันธ์อนุญาตให้ผู้ใช้ ควบคุมข้อมูลได้เต็มที่ ทำให้มีผลิตภัณฑ์รองรับการจัดการฐานข้อมูลแบบสัมพันธ์เกิดขึ้นมากมาย ซึ่งความสามารถเหล่านี้ ทำให้ผู้ใช้สามารถเลือกทำงานตามที่ต้องการได้

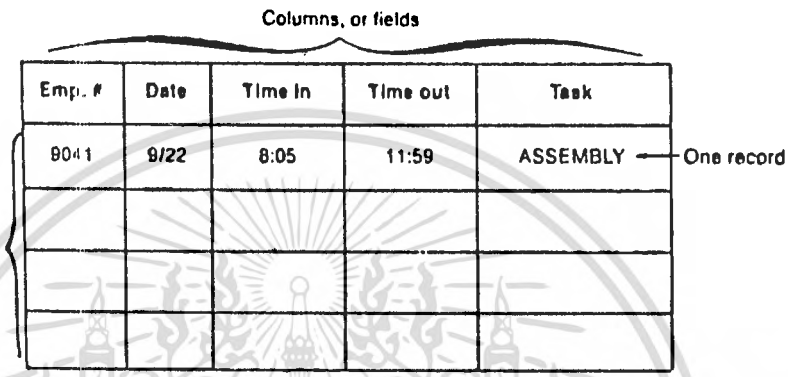
ในปัจจุบัน ระบบจัดการฐานข้อมูลแบบสัมพันธ์ ต้องการที่จะให้ผู้ใช้มากกว่า 1 คนใช้งานกับฐานข้อมูลเดียวกัน โดยมีความปลอดภัยเท่าที่ควร ดังนั้นโปรแกรมประยุกต์เกี่ยวกับฐานข้อมูลได้มีการพัฒนามาเป็นระบบที่มีผู้ใช้หลายคน (Multiuser) ORACLE จะมีเครื่องมือช่วยหลาย ๆ แบบเพื่อจัดการข้อมูลและการพัฒนาโปรแกรมประยุกต์ สำหรับเวอร์ชั่น 5.1 ORACLE ได้แบ่งกลุ่มของโปรแกรมกำหนดหน้าที่ ออกเป็นผลิตภัณฑ์ ซึ่งมีหลายทางที่จะเข้าไป และใช้ระบบตามที่ต้องการ

1 การจัดการข้อมูลของ ORACLE

ใน ORACLE ข้อมูลทั้งหมดจะถูกเก็บและแสดงในรูปของตาราง (Table) ในแต่ละตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะประกอบไปด้วยคอลัมน์ (Columns) หรือบางทีเรียกว่า ฟิลด์ (fields) และแถว (Rows) แต่ละแถวข้อมูลจะเรียกว่า 1 เรกคอร์ด จากตารางผู้ใช้อาจจะเลือกสับเซตของแถวหรือคอลัมน์ แต่ผลลัพธ์จะแสดงออกมาบนหน้าจอ หรือถูกพิมพ์ออกมาเป็นรูปแบบของตารางด้วย รูปที่ 1 แสดงโครงสร้างพื้นฐานของตาราง



รูปที่ ข(1)

วิว (View) ได้มาจากตารางซึ่งผู้ใช้สามารถสร้างเพื่อจุดประสงค์สำหรับการแสดงผล ถึงแม้ว่าวิวจะมองดูเหมือนกับว่าเป็นตารางจริง ๆ แต่ในฐานข้อมูล วิวจะเก็บในลักษณะนิยามเท่านั้น ด้วยเหตุผลนี้ วิวก็จะถูกอ้างว่าเป็นตารางเสมือน (Virtual Tables) โดยตารางที่ถูกวิวถ่ายทอดเรียกว่า ตารางพื้นฐาน (Base Tables) วิวอาจจะเกิดจากการรวมกันของ 2 ตารางพื้นฐาน หรือเป็นซับเซตของตารางพื้นฐาน ผู้ใช้สามารถใช้วิวเพื่อที่จะเข้าถึงตารางได้ และเพื่อทำให้งานที่มีความยุ่งยากให้มีความง่ายขึ้น

เนื่องจาก ORACLE เป็นระบบจัดการฐานข้อมูลแบบสัมพันธ์ ผู้ใช้จึงสามารถที่จะเชื่อมโยงข้อมูลที่ถูกเก็บไว้ในหลาย ๆ ตารางเพื่อเพิ่มประสิทธิภาพในการทำงานและเพื่อการหลีกเลี่ยงการทำซ้ำ การเลือก (Selection) เป็นกระบวนการเพื่อสร้างตารางใหม่ซึ่งประกอบไปด้วยกลุ่มของแถวจากตารางใด ๆ ซึ่งตรงกับเกณฑ์ที่ต้องการ การออกแบบ (Projection) เป็นกระบวนการเพื่อสร้างตารางจากกลุ่มของคอลัมน์จากตารางใด ๆ ซึ่งตรงกับเกณฑ์ที่ต้องการ การร่วมกัน (Join) จะสร้างตารางใหม่ซึ่งเป็นการเชื่อมกันของแถวทั้งหมดใน 2 ตาราง โดยไม่รวมแถวที่มีข้อมูลเหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 การเข้าถึงข้อมูลของ ORACLE

ส่วนแก่น (Core) ของ ORACLE คือ SQL (Structured Query Language) เป็นภาษาซึ่งผู้ใช้สามารถติดต่อกับ ORACLE SQL ประกอบไปด้วยกลุ่มคำพื้นฐานของภาษาอังกฤษ เช่น Select และ Create เป็นต้น ซึ่งผู้ใช้สามารถใช้คำสั่งโครงสร้างเพื่อที่จะเข้าถึงและจัดการข้อมูลที่ถูกเก็บไว้ในฐานของมูลแบบสัมพันธ์

ORACLE ใช้ SQL เพื่อที่จะเข้าถึง (Access) เปลี่ยนแปลง (Modify) และแสดงผล (Display) ข้อมูล ORACLE ยังสามารถใช้เครื่องมือ (Tools) อื่นในการกระทำกับข้อมูลได้เช่นกัน

3 เหตุผลของการเลือก ORACLE

ORACLE มีมาหลายปีในวงการระบบฐานข้อมูลแบบสัมพันธ์ ถึงแม้จะไม่มีความเป็นฟังก์ชันและระบบที่ใช้กับไมโครคอมพิวเตอร์ เหมือนกับ dBASE IV แต่ ORACLE มีความสามารถในการใช้งานแบบผู้ใช้หลายคน การเข้าถึงการควบคุม และ SQL-Compatibility ก็ถูกเพิ่มเข้ามา นอกจากนี้ ORACLE ยังมีข้อดีสำหรับการใช้งาน คือ

3.1 ORACLE ให้ความปลอดภัยแก่ผู้ใช้

ORACLE มีหลายลักษณะเพื่อที่จะให้ความแน่นอนในความถูกต้องของฐานข้อมูล เมื่อมีการขัดจังหวะ (Interruption) เกิดขึ้นในขณะที่ทำงานอยู่ การกลับคืนสู่ระดับเดิม (Rollback) สามารถที่จะเริ่มให้ฐานข้อมูลนี้ไปยังตำแหน่งที่ก่อนที่จะเกิดความผิดพลาด

ORACLE มีฟังก์ชันมากมายสำหรับการรักษาความปลอดภัยของข้อมูล คำสั่ง Grant และ Revoke จะจำกัดการเข้าถึงของข้อมูลลงไปถึงระดับแถวและคอลัมน์ วิว ก็เป็นรูปแบบหนึ่งสำหรับการจำกัดการเข้าถึงตารางพื้นฐานในระบบฐานข้อมูล ซึ่งผู้ใช้จะพบว่ามีหลายทางที่จะควบคุมการเข้าถึงฐานข้อมูลใน ORACLE

3.2 ORACLE สามารถดำเนินงานได้อย่างเหมาะสม

ORACLE ได้ถูกปรับปรุงอย่างต่อเนื่อง เพื่อที่จะให้มีความสามารถดำเนินงานได้อย่างเหมาะสม บนฐานข้อมูลที่มีขนาดใหญ่ เนื่องจากระบบฐานข้อมูลแบบสัมพันธ์มีข้อเสียในเรื่องของความล่าช้าในการเข้าถึงข้อมูลเป็นอย่างมาก ORACLE ได้พัฒนาตัวเองอย่างต่อเนื่อง ซึ่งผลก็คือ ชุดคำสั่งประจำสามารถ ค้นวิถีทางที่ดีที่สุดที่จะไปยังข้อมูล อย่างรวดเร็ว โดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทคนิคในการจัดกลุ่มของ ORACLE เพื่อที่จะเก็บข้อมูลลงบนดิสก์ ก็เป็นอีกคุณสมบัติหนึ่ง และยังมีฟังก์ชั่นเพิ่มเติมช่วยควบคุมการติดฐานข้อมูลที่มีความซับซ้อนด้วย

3.3 ORACLE ใช้ชุดคำสั่ง SQL

ORACLE ใช้ชุดคำสั่ง SQL ซึ่งใกล้เคียงกับมาตราฐาน ANSI และมีความเข้ากันได้กับ DB2 ของ IBM และ DS/SQL

ผู้ใช้จะได้รับประโยชน์ของการปรับปรุงอย่างมากของ ORACLE เพื่อให้เข้าใกล้มาตราฐาน SQL ORACLE ได้เพิ่มคำสั่งรูปแบบของการรายงานเพื่อที่จะขยายความสามารถในการรายงานผลของผลลัพธ์ และเพื่อที่จะให้มีการแสดงผลการรายงานอย่างต่อเนื่อง นอกจากนี้ ORACLE ได้เพิ่มฟังก์ชันทางสถิติ (Statistical) ทางคณิตศาสตร์ (Arithmetic) ชุดอักขระ (String) และวันที่เวลา (Date/Time) ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้