

ระบบสร้างกฎบังคับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับ
ฐานข้อมูลแบบรีเลชันแนล

A NATURAL LANGUAGE-BASED INTEGRITY RULES GENERATOR
FOR RELATION DATABASE ENVIRONMENT



นางมยุรี เลิศเวชกุล

MRS.MAYUREE LERTWATECHAKUL

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
บัณฑิตวิทยาลัย

เลขหมู่..... สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เลขทะเบียน.....28901
วัน, เดือน, ปี..... 1 1 พ.ย. 2540
พ.ศ. 2540
ISBN 974-621-921-9

ลิขสิทธิ์ของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**A NATURAL LANGUAGE-BASED INTEGRITY RULES GENERATOR
FOR RELATION DATABASE ENVIRONMENT**



MAYUREE LERTWATECHAKUL

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

1997

ISBN 974-621-921-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	ระบบสร้างกฎบังคับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนล	๔๖
นักศึกษา	นางมยุรี เลิศเวชกุล	
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ.ดร.ศุภมิตร จิตตะยโสธร	
ระดับการศึกษา	วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า	
ภาควิชา	วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์	
	สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง	
พ.ศ.	2540	

บทคัดย่อ

การออกแบบฐานข้อมูลเป็นการออกแบบการจัดเก็บข้อมูลลงบนฐานข้อมูล ด้วยความถูกต้องปลอดภัย และประหยัดเนื้อที่ในการเก็บข้อมูล ซึ่งประกอบด้วยการวิเคราะห์หาความสัมพันธ์ของข้อมูลและกฎข้อบังคับความถูกต้องของข้อมูล เครื่องมือในการออกแบบฐานข้อมูลที่นิยมใช้กันอย่างแพร่หลายในปัจจุบันนี้มีอยู่หลายวิธี เช่น วิธีการนอร์มัลไลเซชัน, แบบจำลองอีอาร์ และวิธีการของโนแอม เป็นต้น ถึงแม้ว่าวิธีการเหล่านี้จะสามารถช่วยสร้างบรรทัดฐานในการออกแบบฐานข้อมูลสำหรับผู้ออกแบบฐานข้อมูลก็ตามที่ แต่ก็ยังขาดปัจจัยที่ช่วยให้การสื่อสารระหว่างผู้ออกแบบฐานข้อมูลและผู้ปฏิบัติงานจริงในระบบงานเป็นไปได้อย่างตรงไปตรงมา และการแสดงกฎข้อบังคับความถูกต้องของข้อมูลที่ละเอียดครบถ้วน อันเป็นอุปสรรคต่อการออกแบบฐานข้อมูลให้ถูกต้องสมบูรณ์ อีกทั้งการกำหนดกฎข้อบังคับความถูกต้องของข้อมูลบนระบบจัดการฐานข้อมูลส่วนมากก็ยังจะต้องอาศัยการเขียนเป็นโปรแกรมในรูปของ database trigger และ stored procedure ซึ่งยากต่อการกำหนด และดูแลแก้ไขในภายหลัง

งานวิจัยนี้จึงนำเสนอระบบสร้างกฎบังคับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนลที่ยึดถือเอาแนวทางโนแอม และอัลกอริทึม ONF เป็นหลักในการวิเคราะห์ความสัมพันธ์และการออกแบบฐานข้อมูล ระบบนี้จะรับคำสั่งจากผู้ใช้ในรูปของภาษาธรรมชาติไววยากรณ์รูปแบบจำกัดที่ถูกออกแบบให้สามารถเข้าใจได้ง่าย โดยที่ผู้ใช้ไม่จำเป็นจะต้องทราบทฤษฎีพื้นฐานของการออกแบบฐานข้อมูลมาก่อน ผู้ใช้สามารถที่จะกำหนดความสัมพันธ์และกฎข้อบังคับความถูกต้องของข้อมูลด้วยภาษาดังกล่าวได้ในรูปแบบบรรทัดเดียว ซึ่งจะสื่อความเข้าใจได้ดีกว่าวิธีอื่นๆ และระบบสร้างกฎบังคับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนลนี้จะสร้างประโยคคำสั่ง SQL ที่ใช้กำหนดโครงสร้างของตารางฐานข้อมูล และกฎข้อบังคับความถูกต้องของข้อมูลทั้งในรูปของกฎที่เป็นแบบบรรทัดเดียวและ stored

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา^I และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

procedure ซึ่งจะช่วยให้ออกแบบฐานข้อมูลสะดวกเร็ว และสามารถตรวจสอบความถูกต้องได้ง่ายยิ่งขึ้น

•



• • •

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	A Natural Language-Based Integrity Rules Generator for Relational Database Environment
Student	Mrs. Mayuree Lertwatechakul
Thesis Advisor	Assoc. Prof. Dr. Suphamit Chittayasothorn
Level of Study	Master of Engineering in Electrical Engineering
Department	Computer Engineering Department in Faculty of Engineering King Mongkut's Institute of Technology Ladkrabang
Year	1997

ABSTRACT

The design of a database is a process which aims at producing data structures which has minimum redundancy thus increase correctness and reduce the chance to have data inconsistency. Database design methodologies such as the Normalization process, the Entity-Relationships modelling and the NIAM information analysis methodology all lead to the same design result. However, they lack proper user interfaces which ease the design process and improve the communication between the designers and users. Integrity constraints definitions which are definition of validation rules are normally not available in most systems. They are often design in the form of database triggers and stored procedures which are hard to define and maintain.

This thesis presents an automatic integrity constraint definition and enforcement system which is based on the NIAM methodology and the ONF algorithm. The system accepts formal natural language instructions which are easy to understand. The users do not have to know any database design methodologies. They simply define relationships between data items and integrity rules in descriptive language. The system will generate 5NF relational database schemata together with validation rules in SQL, declarative rules and stored procedures.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงลงด้วยดีได้นั้นต้องอาศัยความร่วมมือและความกรุณาจากบุคคลหลายฝ่าย โดยเฉพาะอาจารย์ศุภมิตร จิตตะยโสธร อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งกรุณาช่วยปูพื้นฐานความรู้, ชี้แนะแนวทางในการทำงานวิจัย และให้คำปรึกษาในทุกๆ ด้านตลอดการศึกษาย่างดีเรื่อยมา จึงขอกราบขอบพระคุณท่านอาจารย์เป็นอย่างสูง อีกทั้งต้องขอขอบพระคุณทุกคนในครอบครัวที่ต่างก็ให้ทั้งกำลังใจและช่วยผ่อนเบาภาระของข้าพเจ้าไปได้อย่างมาก ซึ่งก็ได้แก่บิดา, มารดา, น้ำ, สามี และ พี่น้องของข้าพเจ้าและสามี หากข้าพเจ้าไม่ได้รับความช่วยเหลือจากบุคคลในครอบครัวแล้ว คงเป็นเรื่องยากยิ่งที่วิทยานิพนธ์ฉบับนี้จะสำเร็จลงได้

ขอขอบคุณท่านอาจารย์อุทัย ศรีธีระวิโรจน์ หัวหน้าภาควิชาเทคนิคอุตสาหกรรม และอาจารย์และเจ้าหน้าที่ทุกท่านในภาควิชาที่กรุณาช่วยแบ่งเบาภาระงานสอน และธุระอื่นๆ ในภาควิชา

ขอขอบคุณ พี่ เพื่อน และน้องๆ ชาวลาดกระบังทุกคนที่คอยแนะนำ ให้กำลังใจ และช่วยเหลือข้าพเจ้ามาโดยตลอด

ขอขอบคุณสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง โดยเฉพาะภาควิชาวิศวกรรมคอมพิวเตอร์, แผนกสารสนเทศ, คณะวิศวกรรมศาสตร์ และสำนักวิจัยและบริการคอมพิวเตอร์ ที่กรุณาให้ความช่วยเหลือ ทั้งทางด้านความรู้, ตำรา, คู่มือ, และเครื่องมือที่ใช้ในการทำวิทยานิพนธ์ฉบับนี้เป็นอย่างดี

ขอขอบคุณบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง ที่กรุณาให้ทุนอุดหนุนการทำวิทยานิพนธ์

สุดท้ายขอขอบคุณคณาจารย์และเจ้าหน้าที่แผนกบัณฑิตศึกษาคณะวิศวกรรมศาสตร์ และเจ้าหน้าที่บัณฑิตวิทยาลัยทุกท่านเป็นอย่างยิ่ง ที่อำนวยความสะดวกในการติดต่อและสอบถามข้อมูลรวมถึงระเบียบการต่างๆ ทำให้ข้าพเจ้าสามารถดำเนินการศึกษา และทำวิทยานิพนธ์เป็นจนประสบผลสำเร็จ

มยุรี เลิศเวชกุล

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	III
กิตติกรรมประกาศ.....	IV
สารบัญ.....	V
สารบัญตาราง.....	VIII
สารบัญภาพ.....	X
บทที่	
1. บทนำ.....	1
ความเป็นมาและความสำคัญของปัญหา.....	1
ขอบเขตของงานวิจัย.....	3
ประโยชน์ที่คาดว่าจะได้รับ.....	3
วิธีดำเนินงาน และส่วนประกอบของวิทยานิพนธ์.....	5
2. ทฤษฎีที่เกี่ยวข้อง.....	8
ฐานข้อมูลแบบรีเลชันแนล.....	8
การออกแบบฐานข้อมูลแบบรีเลชันแนลด้วยวิธีของไนแอม.....	15
อัลกอริธึม ONF.....	44
3. กฎข้อบังคับความถูกต้องของข้อมูล.....	47
แนวทางการแบ่ง กฎข้อบังคับความถูกต้องของข้อมูลจากงานวิจัยอื่น.....	47
แนวทางการบังคับความถูกต้องของข้อมูลจากงานวิจัยอื่น.....	54
การบังคับกฎข้อบังคับความถูกต้องของข้อมูลบนระบบจัดการฐานข้อมูล.....	55
สรุปแนวทางการแบ่งชนิดกฎข้อบังคับความถูกต้องของข้อมูล.....	61
4. เมตาสกีมมา.....	67
ขั้นตอนการออกแบบเมตาสกีมมา.....	70
5. ภาษาธรรมชาติไวยากรณ์รูปแบบจำกัด.....	81
BNF แสดงโครงสร้างของภาษาธรรมชาติรูปแบบจำกัด.....	86

สารบัญ (ต่อ)

บทที่	หน้า
การแสดงตัวอย่างการกำหนดความสัมพันธ์และกฎข้อบังคับความต้องการของข้อมูล.....	86
6. สถาปัตยกรรมของระบบ.....	95
หน้าที่ของ Syntax Checker.....	95
หน้าที่ของ Schema Analyzer.....	95
หน้าที่ของ Schema Compiler.....	97
หน้าที่ของ Rule Generator.....	97
7. การพัฒนาระบบ.....	98
อุปกรณ์ที่ใช้ในการพัฒนาระบบ.....	98
การพัฒนาระบบในส่วนต่างๆ.....	98
8. การทดสอบและใช้งานระบบ.....	116
วิธีการใช้งานระบบ.....	116
ตัวอย่างข้อมูลที่ใช้ในการทดสอบ.....	125
ตัวอย่างผลลัพธ์ที่ได้จากการทดสอบ.....	128
9. คุณสมบัติในการรักษาความต้องการของข้อมูลที่ระบบจัดการฐานข้อมูลแบบรีเลชันแนลพึงมี.....	133
ปัญหาในการบังคับความต้องการของข้อมูลบนระบบจัดการฐานข้อมูลในปัจจุบัน.....	133
คุณสมบัติที่จำเป็นในการรักษาความต้องการของข้อมูล.....	137
10. สรุปผลและแนะแนวทางการวิจัย.....	140
สรุปผลงานวิจัย.....	140
ปัญหาที่พบในการวิจัยและพัฒนาระบบ.....	142
แนวทางการประยุกต์ใช้.....	143
แนวทางการพัฒนาต่อ.....	144
บรรณานุกรม.....	146
ภาคผนวก.....	147
ภาคผนวก ก.....	148

สารบัญ (ต่อ)

	หน้า
ภาคผนวก ข.....	165
ประวัติผู้เขียน.....	172



สารบัญตาราง

ตารางที่	หน้า
1. แสดงตัวอย่างข้อมูลการเรียนการสอน.....	30
2. แสดงตัวอย่างรายงานภายในระบบงานที่สนใจ.....	32
3. แสดงตัวอย่างข้อมูล.....	37
4. แสดงตัวอย่างข้อมูล.....	39
5. แสดงข้อมูลที่เกิดจากการใช้วิธีโปรเจกชันกับข้อมูลในตารางที่ 4.....	40
6. แสดงข้อมูลที่เกิดจากการใช้วิธีโปรเจกชันกับข้อมูลในตารางที่ 4.....	40
7. แสดงข้อมูลที่เกิดจากการใช้วิธีเนเชอรัลจอยน์กับข้อมูลในตารางที่ 5 และตารางที่ 6....	40
8. แสดงข้อมูลที่เกิดจากการใช้วิธีโปรเจกชันกับข้อมูลในตารางที่ 4.....	40
9. แสดงข้อมูลที่เกิดจากการใช้วิธีเนเชอรัลจอยน์กับข้อมูลในตารางที่ 6 และตารางที่ 8....	40
10. แสดงตารางความสัมพันธ์ที่เกิดจากการใช้อัลกอริทึม ONF ขั้นตอนที่ 1.....	45
11. แสดงตารางความสัมพันธ์ที่เกิดจากการใช้อัลกอริทึม ONF ขั้นตอนที่ 2.....	46
12. แสดงตารางความสัมพันธ์ที่เกิดจากการใช้อัลกอริทึม ONF ขั้นตอนที่ 3.....	46
13. แสดงตารางเมตาสกีมว่า OBJECT.....	76
14. แสดงตารางเมตาสกีมว่า IDENTIFIER.....	76
15. แสดงตารางเมตาสกีมว่า ROLE.....	76
16. แสดงตารางเมตาสกีมว่า CONSTRAINT.....	77
17. แสดงตารางเมตาสกีมว่า IC_ROLE.....	77
18. แสดงตารางเมตาสกีมว่า MANDATORY_ENT.....	77
19. แสดงตารางเมตาสกีมว่า MEMBERSHIP.....	77
20. แสดงตารางเมตาสกีมว่า RANGE.....	77
21. แสดงตารางเมตาสกีมว่า OCCURRENCE.....	78
22. แสดงตารางเมตาสกีมว่า CARDINALITY.....	78
23. แสดงตารางเมตาสกีมว่า CARDINAL_ROLE.....	78
24. แสดงตารางเมตาสกีมว่า EQUALITY.....	78
25. แสดงตารางเมตาสกีมว่า SUBSET.....	78

สารบัญตาราง (ต่อ)

ตารางที่	หน้า
26. แสดงตารางเมตาสกีมนำ SUBTYPE.....	79
27. แสดงตารางเมตาสกีมนำ TRANS_STRING.....	79
28. แสดงตารางเมตาสกีมนำ TRANS_VALUE.....	79
29. แสดงตารางเมตาสกีมนำ DYNAMIC.....	79
30. แสดงตารางเมตาสกีมนำ USER_DEFINED.....	79
31. แสดงตารางเมตาสกีมนำ PK.....	80
32. แสดงตารางฐานข้อมูล DEPT.....	128
33. แสดงตารางฐานข้อมูล SUBJECT.....	129
34. แสดงตารางฐานข้อมูล STUDENT.....	129
35. แสดงตารางฐานข้อมูล ENROLLMENT.....	129
36. แสดงตารางฐานข้อมูล ORDER_WITH_ITEM.....	129
37. แสดงตารางฐานข้อมูล LECTURER.....	130
38. แสดงตารางฐานข้อมูล PRODUCT_HAS_COLOR.....	130
39. แสดงตารางฐานข้อมูล STUDENT_IS_MEMBER_OF_CLUB.....	130
40. แสดงตารางฐานข้อมูล STUDENT_IS_SPONSOR_FOR_CLUB.....	131
41. แสดงตารางฐานข้อมูล MALESTUDENT.....	131
42. แสดงตารางฐานข้อมูล MALESTUDENT_TRAINED_AT_CAMP.....	131
43. แสดงตารางฐานข้อมูล STUDENT_IS_STAFF_OF_SUBJECT_IN.....	131

สารบัญภาพ

หน้า

1. แสดงสถาปัตยกรรมของระบบฐานข้อมูลรีเลชันแนล.....	9
2. แสดงโครงสร้างข้อมูลของฐานข้อมูลแบบรีเลชันแนล.....	10
3. แสดงสัญลักษณ์ชนิดเอนทิตี.....	16
4. แสดงสัญลักษณ์ชนิดเลเบล.....	16
5. แสดงสัญลักษณ์ชนิดความจริง.....	16
6. แสดงสัญลักษณ์ชนิดอ้างอิง.....	17
7. แสดงสัญลักษณ์ชนิดอ้างอิงแบบ one to one.....	17
8. แสดงสัญลักษณ์ชนิดความจริงแบบ one to many.....	17
9. แสดงสัญลักษณ์ชนิดความจริงแบบ one to one.....	17
10. แสดงสัญลักษณ์ชนิดความจริงแบบ many to many.....	18
11. แสดงสัญลักษณ์ชนิดความจริงชนิดยูนารี.....	18
12. แสดงสัญลักษณ์ชนิดความจริงชนิดเทอนารี.....	18
13. แสดงสัญลักษณ์ชนิดความจริงแบบแนสต์.....	19
14. แสดงสัญลักษณ์ inter-uniqueness constraint.....	19
15. แสดงสัญลักษณ์ mandatory role constraint.....	20
16. แสดงสัญลักษณ์ inclusion mandatory role constraint.....	20
17. แสดงสัญลักษณ์ entity type constraint.....	20
18. แสดงสัญลักษณ์ subtype constraint.....	21
19. แสดงสัญลักษณ์ occurrence frequency constraint.....	21
20. แสดงสัญลักษณ์ equality constraint.....	21
21. แสดงสัญลักษณ์ subset constraint.....	22
22. แสดงสัญลักษณ์ exclusion constraint.....	22
23. แสดงสัญลักษณ์ irreflexive constraint.....	23
24. แสดงสัญลักษณ์ asymmetric และ intransitive constraint.....	23
25. แสดงสัญลักษณ์ cardinality และ mandatory entity constraint.....	24
26. แสดงแบบจำลองข้อมูลที่เขียนจากความสัมพันธ์ของข้อมูลในตารางที่ 2.....	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

	หน้า
27. แสดงแบบจำลองข้อมูลตัวอย่าง.....	33
28. แสดงแบบจำลองข้อมูลที่ปรับปรุงจากภาพที่ 27.....	34
29. แสดงแบบจำลองข้อมูลตัวอย่าง.....	34
30. แสดงแบบจำลองข้อมูลที่ปรับปรุงจากภาพที่ 29.....	34
31. แสดงสัญลักษณ์ derived fact type.....	35
32. แสดงแบบจำลองข้อมูลที่เขียนจากความสัมพันธ์ของข้อมูลในตารางที่ 3.....	36
33. แสดงชนิดความจริงที่มีขนาดยาวเกินจริง.....	38
34. แสดงชนิดความจริงที่ปรับปรุงจากภาพที่ 33.....	38
35. แสดงตัวอย่างการเลือกตัวบ่งเอกลักษณ์.....	42
36. แสดงตัวอย่างการเลือกตัวบ่งเอกลักษณ์.....	42
37. แสดงตัวอย่างการเลือกตัวบ่งเอกลักษณ์ที่ถูกต้อง.....	42
38. แสดงแบบจำลองข้อมูลตัวอย่าง.....	45
39. แสดงแบบจำลองเมตาสกิมม่า.....	68
40. แสดงแบบจำลองเมตาสกิมม่า (ต่อ).....	69
41. แสดงแบบจำลองของ mapping information.....	69
42. แสดงตัวอย่าง uniqueness constraint.....	83
43. แสดงตัวอย่าง occurrency frequency constraint.....	83
44. แสดงสถาปัตยกรรมของระบบสร้างกฎบังคับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติ สำหรับฐานข้อมูลแบบรีเลชันแนล.....	96
45. แสดงหน้าจอหลักของโปรแกรม CVS.....	117
46. แสดงหน้าจอเมื่อผู้ใช้คลิกปุ่ม New ในหน้าจอหลัก.....	118
47. แสดงหน้าจอ Open Dialog Box เมื่อผู้ใช้คลิกปุ่ม Open ในหน้าจอหลัก.....	118
48. แสดงหน้าจอเมื่อผู้ใช้เปิดไฟล์ sch.con.....	119
49. แสดงหน้าจอ Save Dialog Box เมื่อผู้ใช้คลิกปุ่ม Save ในหน้าจอหลัก.....	120
50. แสดงหน้าจอเมื่อผู้ใช้คลิกปุ่ม Compile ในหน้าจอหลัก.....	121
51. แสดงหน้าจอแสดงความผิดพลาดที่พบขณะโปรแกรมทำงาน.....	122

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

หน้า

52. แสดงหน้าจอการทำงานของโปรแกรมเสร็จสิ้นสมบูรณ์โดยไม่พบข้อผิดพลาด.....	122
53. แสดงหน้าจอ Save Dialog Box เมื่อผู้ใช้คลิกปุ่ม Save ในหน้าจอแสดงข้อผิดพลาด.....	123
54. แสดงหน้าจอยืนยันความต้องการสร้างไฟล์คำสั่ง.....	123
55. แสดงหน้าจอแสดงและแก้ไขชุดคำสั่งที่โปรแกรมสร้างขึ้น.....	124
56. แสดงหน้าจอ Database Desktop ของ Delphi ที่แสดงข้อมูลที่อยู่ภายในตารางเมตาสกิมมา..	128



บทที่ 1

บทนำ

ความเป็นมาและความสำคัญของปัญหา

ระบบฐานข้อมูลในปัจจุบันนี้นับว่าเป็นสิ่งที่ไม่สามารถจะขาดได้สำหรับทุกๆ ระบบงาน ไม่ว่าจะเป็นงานทางด้านการศึกษา, หรืองานทางด้านบริหารและปฏิบัติการทั้งทางภาครัฐ และเอกชน เนื่องจากระบบฐานข้อมูลมีความสำคัญและความจำเป็นต่อการปฏิบัติงานของทุกๆ หน่วยงาน ดังนั้นระบบฐานข้อมูลที่ดีย่อมที่จะส่งผลให้การดำเนินงานของทุกระบบงานเป็นไปได้อย่างมีประสิทธิภาพ

ในการพัฒนาระบบฐานข้อมูลนั้น การออกแบบฐานข้อมูลนับเป็นขั้นตอนที่สำคัญเป็นอย่างยิ่ง เพราะถ้าหากฐานข้อมูลได้มาจากการออกแบบที่ไม่ดีพอ ก็จะเป็นสาเหตุที่ทำให้การพัฒนาระบบฐานข้อมูลล่าช้าหรือบางกรณีอาจจะทำให้การพัฒนาระบบฐานข้อมูลไม่ประสบผลสำเร็จเลยก็เป็นได้ ซึ่งปัญหานี้มักจะสืบเนื่องมาจากการที่ผู้ออกแบบฐานข้อมูลขาดความรู้ความเข้าใจในระบบงานที่ต้องการจะพัฒนาระบบฐานข้อมูล และไม่สามารถที่จะสื่อสารความเข้าใจของตนกับผู้ปฏิบัติงานจริงในระบบงานนั้นๆ ได้ ถึงแม้ว่าจะมีการพัฒนาแบบจำลองข้อมูลเพื่อให้สามารถสื่อถึงความสัมพันธ์ของข้อมูลให้แสดงออกมาในรูปของแผนภาพแล้วก็ตาม แต่การที่จะให้ฝ่ายผู้ปฏิบัติงานศึกษาและทำความเข้าใจกับแผนภาพเหล่านั้นก็ยังคงเป็นเรื่องยาก ดังนั้นผู้ออกแบบฐานข้อมูลก็ยังคงต้องใช้ความพยายามที่จะสื่อสารกับผู้ใช้งานด้วยคำพูดและเอกสารที่ผู้ใช้งานทำงานอยู่ เพื่อวิเคราะห์ข้อมูลและเขียนแสดงความสัมพันธ์ของข้อมูลเหล่านั้นลงไปเป็นแผนภาพแบบจำลองข้อมูลระดับแนวคิด และเมื่อต้องการที่จะตรวจสอบความถูกต้องของแบบจำลองฐานข้อมูลก็ต้องอาศัยการแปลความหมายต่างๆ ภายในแผนภาพออกเป็นภาษาพูดเพื่อสื่อให้ผู้ใช้งานเข้าใจและสามารถยืนยันความถูกต้องได้ หรือเพื่อการแก้ไขเมื่อพบข้อผิดพลาด ซึ่งการแปลความหมายไปมาระหว่างภาษาพูดกับแผนภาพแบบจำลองข้อมูลในขั้นตอนของการวิเคราะห์, ออกแบบฐานข้อมูล และตรวจสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความถูกต้องนั้น อาจจะเป็นเหตุให้มีข้อมูลบางส่วนตกหล่นไป หรือยังคงปรากฏข้อผิดพลาดอันเนื่องมาจากความเข้าใจคลาดเคลื่อนไม่ตรงกันอยู่

อีกทั้งในปัจจุบันถึงแม้ว่าระบบจัดการฐานข้อมูลจะได้พัฒนาไปมาก ทั้งในส่วนของการบังคับความต้องการของข้อมูล ให้สามารถกำหนดบังคับความต้องการของข้อมูลได้อย่างกว้างขวางยิ่งขึ้น ซึ่งโดยมากจะสามารถกำหนดเป็นกฎในรูปแบบอรรถาธิบายได้สำหรับกฎชนิด primary key, uniqueness, mandatory role, referential integrity, lexical และ entity type แต่กฎอื่นๆ นอกเหนือจากนี้ก็ยังคงต้องอาศัยการกำหนดเป็น database trigger และ stored procedure ซึ่งเป็นการกำหนดในรูปแบบของ procedural ซึ่งการกำหนดกฎข้อบังคับความต้องการของข้อมูลในรูปแบบอรรถาธิบายจะมีข้อดีกว่าแบบ procedural อยู่มากมายหลายประการด้วยกันคือ สามารถกำหนดและดูแลแก้ไขกฎได้โดยง่าย และไม่ต้องเสียเวลาในการทำความเข้าใจกับโครงสร้างและระเบียบในการเขียนและแก้ไขโปรแกรมอีก และยังช่วยลดความเสี่ยงที่อาจจะเกิดจากความผิดพลาดในการเขียนโปรแกรมอีกด้วย

งานวิจัยนี้จึงได้ออกแบบแบบจำลองข้อมูลแบบรีเลชันแนลด้วยภาษาธรรมชาติไวยากรณ์รูปแบบจำกัดขึ้น เพื่อช่วยลดปัญหาในขั้นตอนของการออกแบบฐานข้อมูล และยังได้พัฒนาระบบที่ช่วยสร้าง declarative integrity constraint และ stored procedure สำหรับบังคับความต้องการของข้อมูลบนระบบจัดการฐานข้อมูล ในรูปประโยคคำสั่งแบบอรรถาธิบายเพื่ออำนวยความสะดวกให้แก่ผู้ออกแบบฐานข้อมูลและผู้ดูแลฐานข้อมูลให้สามารถออกแบบ และดูแลแก้ไขฐานข้อมูลและกฎข้อบังคับความต้องการของข้อมูลได้อย่างสะดวกรวดเร็ว และถูกต้องยิ่งขึ้น

นอกจากงานวิจัยนี้จะได้พัฒนาระบบสร้างกฎบังคับความต้องการของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับระบบฐานข้อมูลแบบรีเลชันแนลแล้ว ผลจากการศึกษาชนิดและหน้าที่ของกฎข้อบังคับความต้องการของข้อมูล จึงทำให้เกิดแนวคิดในการวิจัยถึงคุณสมบัติต่างๆ ที่จำเป็นต่อตรวจสอบและรักษาความต้องการของข้อมูลบนระบบจัดการฐานข้อมูล

ดังนั้นการนำเสนอผลงานวิจัยในวิทยานิพนธ์ฉบับนี้ จึงขอแบ่งออกเป็น 2 หัวข้อหลักคือ

- การออกแบบและพัฒนาระบบสร้างกฎบังคับความต้องการของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับระบบฐานข้อมูลแบบรีเลชันแนล
- แนวทางการกำหนดคุณสมบัติในการตรวจสอบและรักษาความต้องการของข้อมูลบนระบบจัดการฐานข้อมูล

ขอบเขตของงานวิจัย

สำหรับระบบสร้างกฎบังคับกับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับระบบฐานข้อมูลแบบรีเลชันแนล ระบบจะรับข้อมูลในรูปของไฟล์ชุดประโยคคำสั่งภาษาธรรมชาติไววยากรณ์จำกัด แล้วทำการตรวจสอบความถูกต้องของไวยากรณ์และวิเคราะห์แยกแยะข้อมูลจากประโยคคำสั่งเหล่านั้นนำไปบรรจุลงในฐานข้อมูลเมตาสกีมม่า และใช้แนวทางของโนแอมและอัลกอร์ริธึม ONF ในการวิเคราะห์ความสัมพันธ์ของข้อมูลและสร้างรีเลชันแนลสกีมม่า ซึ่งผลลัพธ์ของระบบคือ รีเลชันแนลสกีมม่าในรูปของชุดประโยคคำสั่ง SQL DDL (Data Definition Language) และคำสั่งในการกำหนดกฎข้อบังคับกับความถูกต้องของข้อมูลทั้งชนิดที่เป็น declarative integrity constraint และ stored procedure สำหรับโปรแกรมระบบจัดการฐานข้อมูลออราเคิล

สำหรับแนวทางการกำหนดคุณสมบัติในการตรวจสอบและรักษาความถูกต้องของข้อมูลบนระบบจัดการฐานข้อมูลแบบรีเลชันแนลนั้น ได้ศึกษาข้อมูลจากงานวิจัยต่างๆ และลักษณะการบังคับความถูกต้องของข้อมูลของโปรแกรมระบบจัดการฐานข้อมูลแบบรีเลชันแนลชั้นนำหลายโปรแกรม เช่น SQL Server, ORACLE และ INGRES รวมถึงการศึกษาลักษณะความสัมพันธ์ของข้อมูล, กฎข้อบังคับความถูกต้องของข้อมูลจากระบบงานจริงหลายๆ ระบบงาน และการวิเคราะห์ปัญหาต่างๆ ที่อนุมานขึ้น

ประโยชน์ที่คาดว่าจะได้รับ

1. ประโยชน์จากการพัฒนาระบบสร้างกฎบังคับกับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนล

1.1 การออกแบบฐานข้อมูล ช่วยให้การออกแบบฐานข้อมูลเป็นไปอย่างมีประสิทธิภาพ กล่าวคือ สามารถกระทำได้อย่างสะดวกรวดเร็วยิ่งขึ้น และตรวจสอบความผิดพลาดที่อาจเกิดขึ้นได้ง่าย ผู้ออกแบบฐานข้อมูลไม่จำเป็นต้องมีความรู้ในการออกแบบฐานข้อมูล ทำให้ผู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปฏิบัติงานในระบบงานสามารถที่จะมีส่วนร่วมในการออกแบบฐานข้อมูลด้วยได้ ซึ่งจะส่งผลให้ฐานข้อมูลที่ออกแบบได้ตรงกับความต้องการของผู้ใช้มากที่สุด และมีข้อผิดพลาดน้อยที่สุดด้วย

1.2 การบำรุงรักษาฐานข้อมูล เนื่องจากการกำหนดแบบจำลองของข้อมูลและกฎข้อบังคับความถูกต้องของข้อมูลเป็นภาษาธรรมชาติ อันเป็นรูปแบบที่สะดวกต่อการทำความเข้าใจนั้น ทำให้การศึกษารูปแบบข้อมูลที่จะกระทำได้ง่าย หากมีความต้องการเปลี่ยนแปลงกฎเกณฑ์หรือแก้ไขความสัมพันธ์ของข้อมูลใดๆ ก็จะสามารถแก้ไขเปลี่ยนแปลงตารางฐานข้อมูลหรือกฎข้อบังคับความถูกต้องของข้อมูลได้อย่างสะดวก รวดเร็ว

2. ประโยชน์จากแนวทางการตรวจสอบและรักษาความถูกต้องของข้อมูลของระบบจัดการฐานข้อมูล

ข้อมูลที่ได้จากการวิจัยในส่วนนี้ จะเป็นประโยชน์สำหรับผู้ที่มีความสนใจทำวิจัยหรือพัฒนาระบบจัดการฐานข้อมูล เพราะเป็นข้อมูลที่ได้ศึกษามาจากผลงานวิจัยที่เกี่ยวข้องกับการพัฒนาระบบฐานข้อมูล และงานวิจัยที่สนับสนุนการรักษาความถูกต้องของข้อมูลบนฐานข้อมูล , บทความทางวิชาการที่กำหนดคุณสมบัติและหน้าที่ของระบบจัดการฐานข้อมูล รวมทั้งความรู้ที่อนุมานได้จากปัญหาที่ประสบในระหว่างการวิจัยและการพัฒนาระบบฐานข้อมูลจริงด้วยระบบจัดการฐานข้อมูลเชิงพาณิชย์ ซึ่งข้อมูลนี้ไม่ได้เป็นข้อมูลเชิงรวบรวมเท่านั้น แต่เป็นข้อมูลที่วิเคราะห์แล้วว่ามีความเป็นไปได้ในการพัฒนาเป็นคุณสมบัติของระบบจัดการฐานข้อมูล รวมถึงประโยชน์ที่จะได้รับจากคุณสมบัติต่างๆ เหล่านั้นด้วย

และนอกจากข้อมูลนี้จะเป็นประโยชน์ในเชิงของการวิจัยและพัฒนาแล้ว คาดว่าอาจจะเป็นประโยชน์สำหรับผู้คนที่สนใจจะใช้เป็นข้อมูล ในการพิจารณาเปรียบเทียบคุณสมบัติการรักษาความถูกต้องของข้อมูลของระบบจัดการฐานข้อมูลอีกด้วย

วิธีดำเนินการวิจัยและส่วนประกอบของวิทยานิพนธ์

ศึกษาทฤษฎีต่างๆ ที่เกี่ยวข้องกับงานวิจัย

1. ทฤษฎีเกี่ยวกับฐานข้อมูลแบบรีเลชันแนล
2. วิธีการออกแบบฐานข้อมูลแบบรีเลชันแนล วิธีต่างๆ

ศึกษางานวิจัยที่เกี่ยวข้องกับงานวิจัยนี้ ตามหัวข้อต่อไปนี้

1. ภาษาที่ใช้ในการกำหนดแสดงความสัมพันธ์ของข้อมูล
2. ภาษาที่ใช้ในการกำหนดกฎข้อบังคับความถูกต้องของข้อมูล
3. วิธีการแบ่งประเภทของกฎข้อบังคับความถูกต้องของข้อมูล
4. วิธีการบังคับความถูกต้องของข้อมูล

ศึกษาระบบจัดการฐานข้อมูลที่มีใช้แพร่หลายอยู่ในปัจจุบัน ตามหัวข้อต่อไปนี้

1. ภาษาที่ใช้ในการกำหนดแสดงความสัมพันธ์ของข้อมูล
2. ภาษาที่ใช้ในการกำหนดกฎข้อบังคับความถูกต้องของข้อมูล
3. วิธีการแบ่งประเภทของกฎข้อบังคับความถูกต้องของข้อมูล
4. วิธีการบังคับความถูกต้องของข้อมูล

ศึกษาเครื่องมือที่ใช้ในการพัฒนาระบบอัตโนมัติสำหรับบังคับความถูกต้องของข้อมูล

1. ระบบจัดการฐานข้อมูลอรรถาธิ
2. โปรแกรม Delphi

งานวิจัยนี้ได้ดำเนินเริ่มต้นจากการค้นคว้าและศึกษาระบบฐานข้อมูลรีเลชันแนลในด้านต่างๆ ทั้งในเชิงทฤษฎี และข้อมูลที่ได้จากการศึกษาและใช้ระบบจัดการฐานข้อมูลรีเลชันแนลเชิงพาณิชย์ โดยได้พิจารณาคุณสมบัติที่ถูกหยิบยกมาใช้ในการแบ่งชนิดและประเภทของกฎข้อบังคับ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความถูกต้องของข้อมูลจากงานวิจัย และคุณสมบัติในการรักษาความถูกต้องของข้อมูลบนระบบจัดการฐานข้อมูล หลังจากนั้นได้ทำการสรุปแนวทางการแบ่งชนิดของกฎข้อบังคับความถูกต้องของข้อมูลจากข้อมูลที่ศึกษามา และความจำเป็นในการพัฒนาโปรแกรมระบบอัตโนมัติสำหรับบังคับความถูกต้องของข้อมูล โดยได้กำหนดแนวทางของระบบดังกล่าวว่าจะต้องเป็นเครื่องมือที่ช่วยในการออกแบบฐานข้อมูลให้เป็นไปอย่างมีประสิทธิภาพโดยที่ผู้ใช้ไม่จำเป็นจะต้องศึกษาทฤษฎีของการออกแบบฐานข้อมูลมาก่อน และช่วยสร้างคำสั่งที่ใช้สร้างตารางฐานข้อมูลและกฎต่างๆ ทั้งชนิดอรรถาธิบาย และชนิด procedural บนระบบจัดการฐานข้อมูลรีเลชันแนล โดยเบื้องต้นได้กำหนดให้ระบบจัดการฐานข้อมูลอรรถาธิบายเป็นเครื่องมือในการพัฒนาต้นแบบ เนื่องจากระบบจัดการฐานข้อมูลอรรถาธิบายนั้นเป็นโปรแกรมระบบจัดการฐานข้อมูลเชิงพาณิชย์ชั้นแนวหน้า ที่มีคุณสมบัติในการรักษาความถูกต้องของข้อมูลสมบูรณ์ที่สุดโปรแกรมหนึ่ง เมื่อเทียบกับระบบจัดการฐานข้อมูลรีเลชันแนลเชิงพาณิชย์อื่นๆ

จากการศึกษาและสรุปประเภทของกฎข้อบังคับความถูกต้องของข้อมูลได้เป็น 2 ชนิด คือ ชนิดอรรถาธิบาย และชนิด procedural นั้น ทำให้เห็นได้ชัดถึงประโยชน์ของกฎชนิดอรรถาธิบายที่เหนือกว่ากฎชนิด procedural อยู่มาก จึงกำหนดเลือกใช้แนวทางการแบ่งชนิดกฎข้อบังคับความถูกต้องของข้อมูลของ G.M. Nijssen และ T.A. Halpin ที่ได้แบ่งชนิดของกฎข้อบังคับความถูกต้องของข้อมูลไว้มากมายหลายชนิด และได้ให้คำจำกัดความไว้อย่างชัดเจน ซึ่งสามารถนำมาใช้ออกแบบโครงสร้างของคำสั่งเพื่อรองรับการกำหนดกฎแต่ละชนิดนั้นๆ เป็นกฎชนิดอรรถาธิบายได้

โดยโครงสร้างของคำสั่งนี้ได้พิจารณาแล้วเห็นว่า โครงสร้างของประโยคภาษาธรรมชาติเป็นลักษณะของคำสั่งที่เหมาะสมที่สุดในการใช้เป็นเครื่องมือในการติดต่อระหว่างผู้ใช้กับระบบ เนื่องจากผู้ใช้ที่ไม่มีความรู้เกี่ยวกับทฤษฎีฐานข้อมูล มักจะไม่เข้าใจการใช้คำสั่งที่มีโครงสร้างเหมือนประโยคตรรกะทางคณิตศาสตร์ ดังนั้นจึงได้ออกแบบคำสั่งภาษาธรรมชาติไวยากรณ์รูปแบบจำกัดขึ้น โดยแต่ละคำสั่งจะเป็นประโยคภาษาอังกฤษที่เรียบง่ายสามารถทำความเข้าใจได้ง่ายถึงแม้ว่าผู้ใช้จะไม่เชี่ยวชาญการใช้ภาษาอังกฤษก็ตาม โครงสร้างและคำที่กำหนดใช้ในประโยคคำสั่งนั้นไม่ได้ออกแบบมาให้ถูกต้องตรงตามไวยากรณ์ของภาษาอังกฤษ เพราะเน้นให้สามารถใช้งานได้สะดวก และการพัฒนาโปรแกรมไม่ยุ่งยากจนเกินไปนัก

เนื่องจากการเลือกใช้ภาษาธรรมชาติและแนวทางการแบ่งกฎข้อบังคับความถูกต้องของข้อมูลของ G.M. Nijssen และ T.A. Halpin ซึ่งเป็นแนวทางเดียวกับวิธีการออกแบบฐานข้อมูลด้วยแบบจำลองในแอม ดังนั้นเพื่อสะดวกในการศึกษาและพัฒนาระบบสร้างกฎบังคับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนลในภายหลัง จึงได้ออกแบบเมตาสกีมม่าซึ่งเป็นฐานข้อมูลที่ใช้เก็บข้อมูลของแบบจำลองข้อมูลระดับแนวคิด โดยยึดเอาแบบจำลองเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในแอมเป็นหลักสำหรับเป็นต้นแบบของแบบจำลองข้อมูล เพราะแบบจำลองข้อมูลในแอมสามารถรองรับการกำหนดความสัมพันธ์ได้หลายรูปแบบและหลายขนาด รวมทั้งการกำหนดกฎข้อบังคับ ความถูกต้องของข้อมูลต่างๆ หลายชนิด เพื่อให้ตารางเมตาสกิมมาที่ออกแบบได้สามารถที่จะรองรับข้อมูลของแบบจำลองข้อมูลระดับแนวคิดได้อย่างครบถ้วน และสามารถนำข้อมูลในตารางเมตาสกิมมานั้นมาใช้ในการสร้างรีเลชันแนลสกีมาของฐานข้อมูลใดๆ ได้อย่างถูกต้อง

หลังจากนั้นได้ศึกษาโปรแกรมระบบจัดการฐานข้อมูลออรากเคิล ถึงวิธีการใช้คำสั่งกำหนดกฎข้อบังคับความถูกต้องของข้อมูลแบบอรรถาธิบาย และการเขียนชุดคำสั่งใน stored procedure และ database trigger รวมทั้งข้อจำกัดและคุณสมบัติต่างๆ ของกฎข้อบังคับความถูกต้องของข้อมูล, stored procedure และ database trigger โดยละเอียด และศึกษาการพัฒนาโปรแกรมด้วยโปรแกรม Delphi ซึ่งเป็นเครื่องมือสำหรับพัฒนาโปรแกรมประยุกต์สำหรับ Windows การพัฒนาโปรแกรมเป็นไปในลักษณะและรูปแบบของ Virtual Programming กล่าวคือ โปรแกรม Delphi จะมีอ็อบเจกต์ชนิดต่างๆ เตรียมไว้ให้ผู้ใช้งาน โดยอ็อบเจกต์แต่ละชนิดจะมีคุณสมบัติต่างๆ ที่ผู้ใช้งานสามารถกำหนดได้และวิธีการต่างๆ ที่เป็นวิธีมาตรฐานในการกระทำต่อคุณสมบัติต่างๆ ของอ็อบเจกต์ โดยผู้ใช้งานสามารถเขียนชุดคำสั่งเพื่อควบคุมการทำงานของโปรแกรม ที่ถูกเรียกขึ้นมาใช้งานเมื่อเกิดเหตุการณ์ใดๆ ขึ้นต่ออ็อบเจกต์ในขณะที่มีการใช้งานโปรแกรม

โปรแกรม Delphi นี้มีเครื่องมืออยู่หลายชนิด แต่ที่นำมาใช้ในการพัฒนาระบบสร้างกฎข้อบังคับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบออรากเคิล มีเพียง 2 ส่วน คือ ส่วนที่ใช้ในการพัฒนาโปรแกรมประยุกต์ และโปรแกรม Database Desktop โดยโปรแกรม Database Desktop นี้ใช้สำหรับสร้างฐานข้อมูลเมตาสกิมมา เพื่อเก็บข้อมูลที่วิเคราะห์ได้จากประโยคภาษาธรรมชาติที่ใช้กำหนดแบบจำลองข้อมูลระดับแนวคิด และใช้โปรแกรม Delphi ออกแบบหน้าจอส่วนติดต่อกับผู้ใช้ และเขียนโปรแกรมตรวจสอบไวยากรณ์ของประโยคคำสั่ง, วิเคราะห์ประโยค, สร้างคำสั่ง SQL DDL สำหรับสร้างตารางฐานข้อมูล และสร้างกฎชนิดต่างๆ สำหรับบังคับความถูกต้องของข้อมูลจากข้อมูลภายในตารางเมตาสกิมมา

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ฐานข้อมูลแบบรีเลชันแนล

ระบบฐานข้อมูลที่ใช้กันอยู่ในปัจจุบันนี้สามารถแบ่งประเภทตามโมเดลของฐานข้อมูลได้ 3 ประเภทคือ เน็ตเวิร์ค, ไฮราคิ และรีเลชันแนล ซึ่งฐานข้อมูลทั้งสามประเภทที่กล่าวมานี้ฐานข้อมูลแบบรีเลชันแนลเป็นโมเดลฐานข้อมูลชนิดที่ได้รับความนิยมแพร่หลายมากที่สุด เนื่องจากเป็นฐานข้อมูลที่มีความยืดหยุ่นสามารถออกแบบและแก้ไขได้ง่าย แม้จะมีการพัฒนาแอปพลิเคชันของระบบฐานข้อมูลไปแล้วก็ตาม ฐานข้อมูลแบบรีเลชันแนลเป็นฐานข้อมูลที่มีรากฐานมาจากทฤษฎีทางคณิตศาสตร์ เช่น เซต และรีเลชัน ทำให้มีจุดยืนในการพัฒนาและกำหนดมาตรฐาน อันเป็นสาเหตุสำคัญประการหนึ่งที่ทำให้โมเดลฐานข้อมูลแบบรีเลชันแนลนี้เป็นที่ยอมรับกันโดยทั่วไป

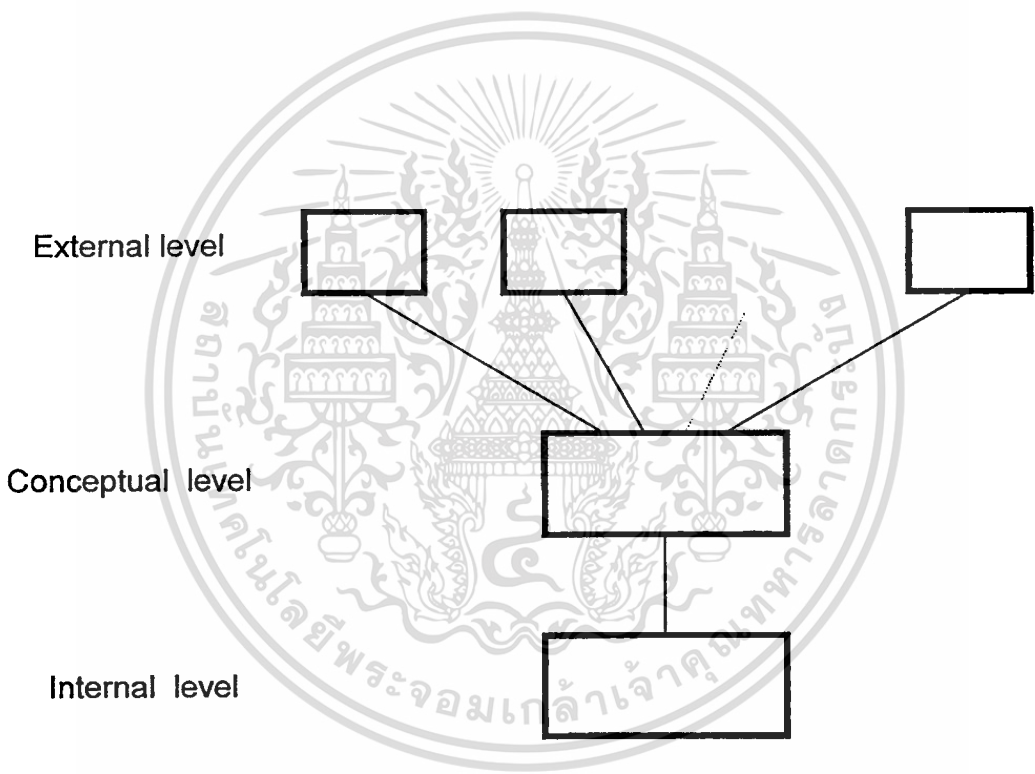
สถาปัตยกรรมของระบบฐานข้อมูลแบบรีเลชันแนล (C.J. Date 1986 : 29) แบ่งออกได้เป็น 3 ระดับคือ

1. **Internal level** เป็นส่วนที่เกี่ยวข้องกับการเก็บข้อมูลลงบนแหล่งเก็บข้อมูลจริง (physical disk)

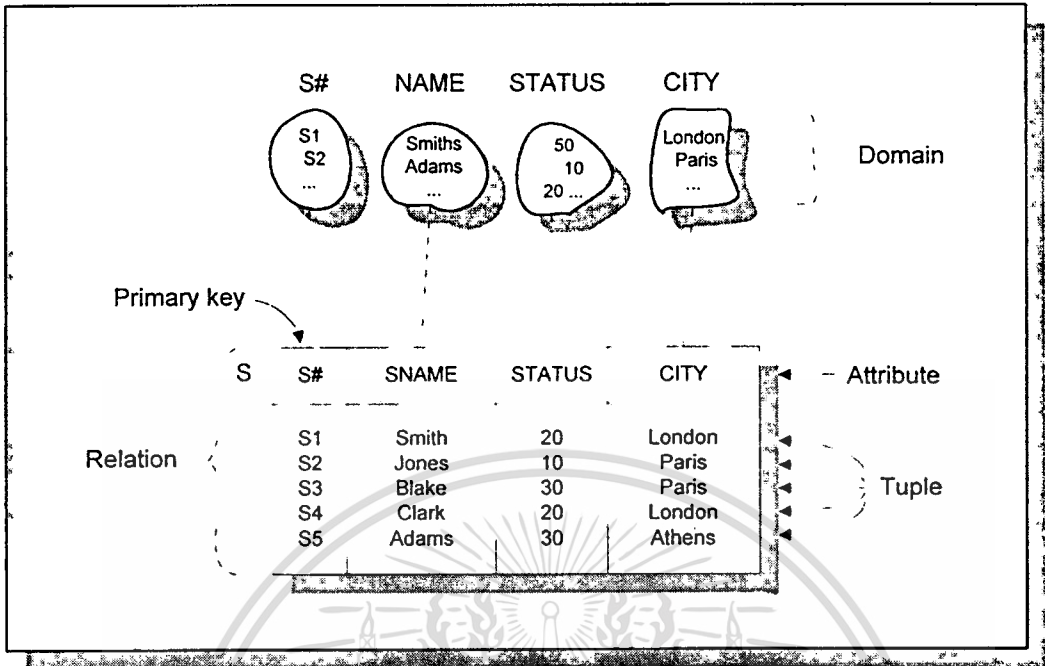
2. **External level** เป็นระดับที่ผู้ใช้งานมองเห็น โดยใน external level นี้อาจจะประกอบไปด้วยหลายๆ external view ที่ประกอบขึ้นจากตารางฐานข้อมูล (table) หรือวิว (view) ต่างๆ โดยที่ผู้ใช้แต่ละคนอาจจะมี external view ที่แตกต่างกันออกไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Conceptual level เป็นระดับที่อยู่ระหว่าง internal level และ external level เป็นระดับที่แสดงโครงสร้างและความสัมพันธ์ของข้อมูลภายในของฐานข้อมูลทั้งหมด



ภาพที่ 1 สถาปัตยกรรมของระบบฐานข้อมูลแบบรีเลชันแนล (C.J. Date 1986 : 30)



ภาพที่ 2 โครงสร้างข้อมูลของฐานข้อมูลแบบรีเลชันแนล (C.J. Date 1986 : 233)

โครงสร้างฐานข้อมูลแบบรีเลชันแนล

ฐานข้อมูลแบบรีเลชันแนลจะแสดงการเก็บข้อมูลในรูปของตารางดังแสดงไว้ในภาพที่ 2 ซึ่งเป็นรูปแบบที่ง่ายต่อการทำความเข้าใจ และสามารถตอบสนองต่อคำถามที่ซับซ้อนได้ดีกว่าโมเดลฐานข้อมูลชนิดอื่น องค์ประกอบสำคัญของระบบจัดการฐานข้อมูลแบบรีเลชันแนลแบ่งออกเป็น 3 ส่วนคือ

- โครงสร้างข้อมูล
- ความถูกต้องของข้อมูล
- การจัดการข้อมูล

1. โครงสร้างข้อมูล (Data Structure) มีองค์ประกอบต่างๆ ดังนี้คือ

1.1 แอททริบิวต์ (Attributes) หมายถึงชื่อคอลัมน์ของตารางความสัมพันธ์ ที่ใช้บ่งชี้ถึงคุณสมบัติของสิ่งต่างๆ ที่มีความสัมพันธ์กันประกอบขึ้นมาเป็นตารางฐานข้อมูล เช่น แอททริบิวต์ S# หมายถึงรหัสของพนักงาน , แอททริบิวต์ SNAME หมายถึงชื่อของพนักงาน และ แอททริบิวต์ CITY หมายถึงชื่อของเมือง โดยที่แอททริบิวต์ดังกล่าวมีความสัมพันธ์กันดังนี้คือ พนักงานที่มีรหัส (S#) QC1203 มีชื่อ (SNAME) ว่า 'John' และอาศัยอยู่ในเมือง (CITY) 'New York'

1.2 โดเมน (Domains) หมายถึงขอบเขตของข้อมูลในแต่ละแอททริบิวต์ เช่น ข้อมูลที่เก็บลงบนแอททริบิวต์ CITY จะต้องเป็นชื่อของเมืองในทวีปยุโรปเท่านั้น

1.3 ทัปเปิล (Tuples) หมายถึง แถวของข้อมูลภายในตารางฐานข้อมูล

1.4 ไพรมารีคีย์ (Primary key) หมายถึง แอททริบิวต์หรือกลุ่มของแอททริบิวต์ที่ทำหน้าที่เป็นตัวชี้เอกลักษณ์ของข้อมูลในแต่ละทัปเปิลของตารางฐานข้อมูล

1.5 รีเลชัน (Relation) หมายถึง ผลคูณคาร์ทีเซียนของโดเมนที่สนใจในแอปพลิเคชัน ประกอบขึ้นด้วยองค์ประกอบ 2 ส่วนคือ เฮดคิง (heading) และ บอดี้ (body)

1.5.1 เฮดคิง ประกอบขึ้นมาจากเซตที่มีขนาดคงที่ของแอททริบิวต์ $A_1, A_2, \dots, A_n$ โดย A_i คือ แอททริบิวต์ที่แทนข้อมูลในโดเมน D_i ($i = 1, 2, \dots, n$)

1.5.2 บอดี ประกอบขึ้นจากเซตของทปเปิดที่มีขนาดแปรผันตามเวลาของรีเลชัน โดยแต่ละทปเปิดประกอบด้วยค่าข้อมูลของแอททริบิวต์ต่างๆที่อ้างอิงกับโดเมนดังจะแสดงต่อไปนี้

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

คุณสมบัติที่สำคัญของฐานข้อมูลแบบรีเลชันแนล มีข้อกำหนดไว้ 4 ประการคือ

- ต้องไม่มีทปเปิดใดที่ซ้ำกันภายในรีเลชัน
- ลำดับของทปเปิดไม่มีความสำคัญในการจัดเก็บและการใช้งานข้อมูล
- ลำดับของแอททริบิวต์ไม่มีความสำคัญในการจัดเก็บและการใช้งานข้อมูล
- ค่าของข้อมูลในแต่ละแอททริบิวต์ต้องเป็นค่าเดี่ยวเสมอ (atomic value)

ต้องไม่มีทปเปิดที่ซ้ำกันภายในรีเลชัน เนื่องจากบอดีของรีเลชันซึ่งประกอบขึ้นจากทปเปิดหลายทปเปิด มีลักษณะเป็นเซตทางคณิตศาสตร์ในรูปของผลคูณคาร์ทีเซียน การซ้ำกันของทปเปิดจึงเปรียบเสมือนการซ้ำกันของสมาชิกภายในเซต ซึ่งผิดนิยามทางคณิตศาสตร์ ดังนั้นจึงกำหนดคุณสมบัติข้อนี้ไว้ เพื่อให้โครงสร้างข้อมูลแบบสัมพันธ์เป็นไปตามกฎทางคณิตศาสตร์ และเนื่องจากคุณสมบัติความเป็นเอกลักษณ์ของแต่ละทปเปิดนี้ ทำให้กล่าวได้ว่า “ทุกๆ รีเลชันจะต้องมีไพรมารีคีย์ (primary key) เสมอ” เพราะอย่างน้อยที่สุด กลุ่มของแอททริบิวต์ที่กำหนดความเป็นเอกลักษณ์ของแต่ละทปเปิด ก็คือเซตของแอททริบิวต์ทั้งหมดภายในรีเลชันนั่นเอง

ลำดับของทปเปิดไม่มีความสำคัญในการจัดเก็บ จากนิยามทางคณิตศาสตร์ที่ถือว่าลำดับของสมาชิกภายในเซตไม่มีความสำคัญ บอดีของรีเลชันซึ่งถือเป็นเซตที่มีสมาชิกเป็นทปเปิดก็เช่นกัน ที่ลำดับในการจัดเก็บของทปเปิดต้องไม่มีผลในการใช้งานรีเลชัน

ลำดับของแอททริบิวต์ไม่มีความสำคัญในการจัดเก็บ จากเหตุผลเดียวกันกับคุณสมบัติในข้อสอง ดังนั้นลำดับของแอททริบิวต์ภายในรีเลชันจึงไม่มีผลต่อการจัดเก็บและการทำงานแต่อย่างใด

ข้อมูลของแต่ละแอททริบิวต์ต้องเป็นค่าเดี่ยวเสมอ (atomic value) หมายถึงการกำหนดแอททริบิวต์นั้น ค่าของแอททริบิวต์ภายในทปเปิดใดๆ จะต้องดึงค่าข้อมูลจากโดเมนที่มันอ้างอิงอยู่มาเพียงค่าเดียวเท่านั้น

2. ความถูกต้องของข้อมูล

การบังคับความถูกต้องของข้อมูลบนระบบฐานข้อมูลแบบรีเลชันอย่างน้อยต้องสามารถบังคับกฎข้อบังคับความถูกต้องของข้อมูลต่อไปนี้ได้คือ

2.1 กฎข้อบังคับความถูกต้องของข้อมูลชนิดเอนติตี้ (Entity type constraint) หมายถึงกฎข้อบังคับความถูกต้องของข้อมูลที่บังคับว่าค่าของข้อมูลภายในแอททริบิวต์ ซึ่งเป็นแอททริบิวต์ของไพรมารีคีย์จะต้องไม่เป็นค่า NULL (ค่า NULL นี้หมายถึงค่าข้อมูลที่ยังไม่พร้อม ซึ่งในทางปฏิบัติหมายถึงยังไม่ได้ใส่ค่าข้อมูล)

2.2 กฎข้อบังคับความถูกต้องของข้อมูลในการอ้างอิง (Referential integrity constraint) ถ้ามีความสัมพันธ์ R2 ซึ่งมีฟอเรนคีย์ FK ที่เกี่ยวข้องกับไพรมารีคีย์ PK ของความสัมพันธ์ R1 แล้ว ทุกๆ ค่าข้อมูลที่ปรากฏอยู่ใน FK ของความสัมพันธ์ R2 จะต้องปรากฏเป็นค่าข้อมูลใน PK ของความสัมพันธ์ R1 เสมอ ยกเว้นกรณีที่ค่าของ FK เป็นค่า NULL

3. การจัดการข้อมูล (Data Manipulation)

ข้อมูลภายในฐานข้อมูลแบบรีเลชันแนล จะต้องถูกจัดการโดยภาษาจัดการข้อมูลของระบบจัดการฐานข้อมูลเท่านั้น โดยภาษาดังกล่าวจะต้องมีคุณสมบัติตามนิยามของ “ความสัมพันธ์ที่สมบูรณ์” (relational complete) กล่าวคือต้องสนับสนุนกลุ่มปฏิบัติการพื้นฐานที่นิยามไว้ในภาษาพีชคณิตสัมพันธ์ (Relational Algebra) หรือ (Relational Calculus) ซึ่งประกอบด้วยปฏิบัติการ 8 อย่าง ที่แบ่งออกได้เป็น 2 กลุ่มดังนี้

3.1 กลุ่มปฏิบัติการพื้นฐาน (Traditional Set Operators) ได้แก่ ผลคูณคาร์ทีเซียน, ยูเนียน, อินเทอร์เซกชัน และดิฟเฟอเรนซ์

3.2 กลุ่มปฏิบัติการพิเศษ (Special Relation Operators) ได้แก่ select, project, join และ divide

ลักษณะการทำงานของตัวปฏิบัติการต่างๆ

SELECT คือ การสร้างความสัมพันธ์จากการเลือกเอาข้อมูลในทปเปิดที่มีคุณสมบัติตามที่ระบุออกมาจากความสัมพันธ์ที่มีอยู่

PROJECT คือ การสร้างความสัมพันธ์โดยเลือกเอาข้อมูลเพียงบางแอททริบิวต์จากจำนวนแอททริบิวต์ทั้งหมดของความสัมพันธ์ใดความสัมพันธ์หนึ่ง

PRODUCT คือ การสร้างความสัมพันธ์ขึ้นใหม่จากความสัมพันธ์ 2 ความสัมพันธ์ใดๆ โดยความสัมพันธ์ใหม่ที่สร้างขึ้นจะประกอบไปด้วยข้อมูลที่เป็นไปได้ทั้งหมดที่เกิดจากการนำทปเปิดของความสัมพันธ์ตั้งต้นมาจับคู่กัน ในลักษณะของผลคูณคาร์ทีเซียน

UNION คือ การสร้างความสัมพันธ์ขึ้นใหม่จาก 2 ความสัมพันธ์ใดๆ โดยความสัมพันธ์ตั้งต้นทั้งสองนั้นจะต้องมีโครงสร้างเหมือนกัน (ชนิดและจำนวนของแอททริบิวต์เหมือนกัน) และความสัมพันธ์ที่สร้างขึ้นใหม่นี้จะเป็นเซตที่ประกอบไปด้วยทปเปิดของความสัมพันธ์ตั้งต้นที่นำมายูเนียนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DIFFERENCE คือ การสร้างความสัมพันธ์ขึ้นใหม่จาก 2 ความสัมพันธ์ใดๆ โดยความสัมพันธ์ดั้งเดิมทั้งสองนั้นจะต้องมีโครงสร้างเหมือนกัน (ชนิดและจำนวนของแอททริบิวต์เหมือนกัน) และความสัมพันธ์ที่สร้างขึ้นใหม่นี้จะประกอบไปด้วยทปเปิดของความสัมพันธ์ที่หนึ่ง แต่เป็นทปเปิดที่ไม่มีปรากฏอยู่ในความสัมพันธ์ที่สอง

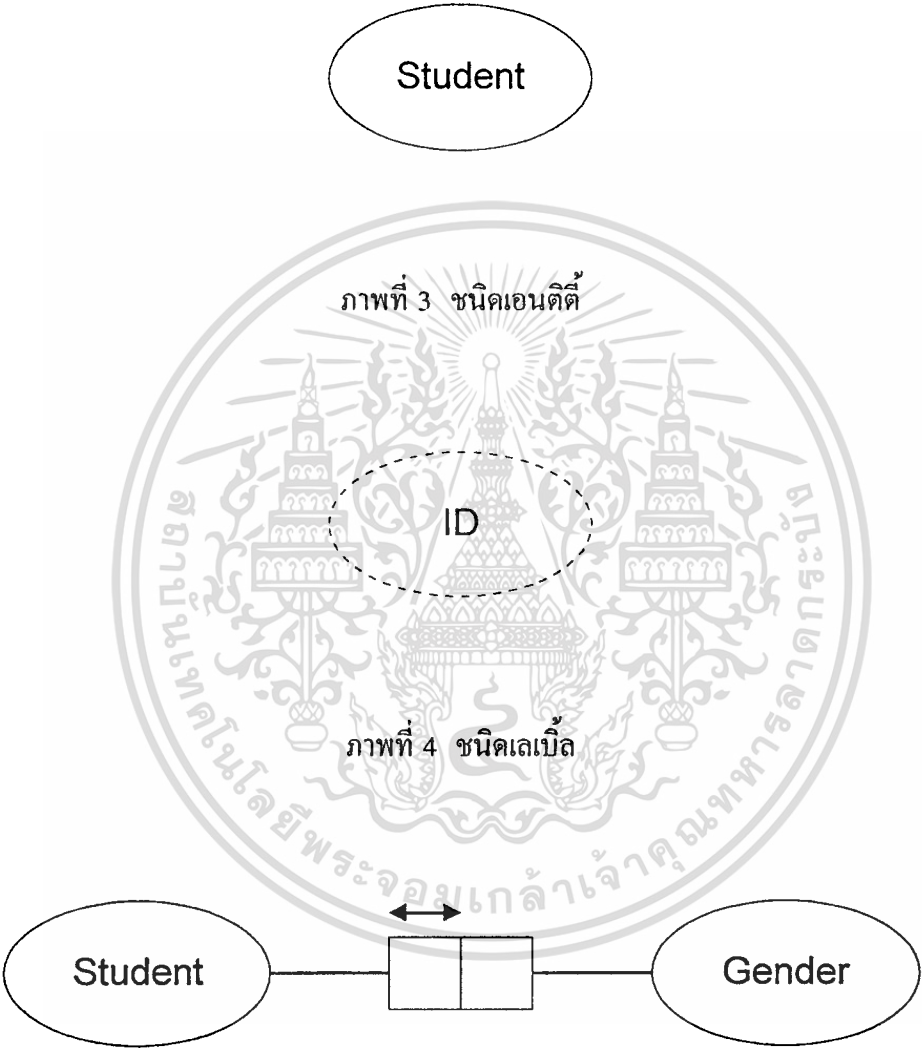
JOIN คือ การสร้างความสัมพันธ์ขึ้นใหม่จาก 2 ความสัมพันธ์ใดๆ โดยความสัมพันธ์ที่สร้างขึ้นใหม่นี้จะประกอบไปด้วยทปเปิดที่เกิดจากการจับคู่กันระหว่างทปเปิดของความสัมพันธ์ดั้งเดิมภายใต้เงื่อนไขที่กำหนด

DEVIDE คือ การสร้างความสัมพันธ์ขึ้นใหม่จากความสัมพันธ์ชนิดไบนารีและความสัมพันธ์ชนิดอรรี โดยความสัมพันธ์ที่สร้างขึ้นใหม่จะมีข้อมูลเพียงแอททริบิวต์เดียวซึ่งเป็นแอททริบิวต์แรกของความสัมพันธ์ชนิดไบนารี และเป็นข้อมูลที่อยู่ในทปเปิดเดียวกับข้อมูลของแอททริบิวต์ที่สองที่ปรากฏค่าข้อมูลอยู่ในความสัมพันธ์ชนิดอรรี

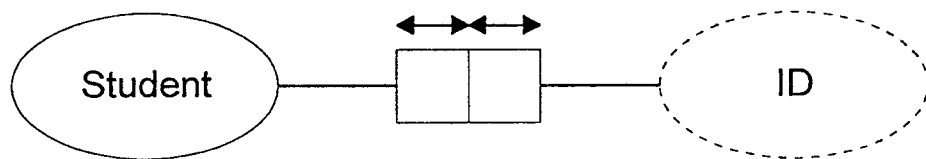
แบบจำลองไนแอม (Nijssen และ Halpin 1989 : 31)

แบบจำลองไนแอม เป็นแบบจำลองข้อมูลที่มีพื้นฐานมาจากภาษารรรมาติและทฤษฎีเซตที่ใช้แผนภาพเป็นสื่อในการแสดงความสัมพันธ์และกฎข้อบังคับความถูกต้องของข้อมูล ซึ่งแบบจำลองไนแอมนี้มีสัญลักษณ์ต่างๆ ที่สามารถใช้แสดงความสัมพันธ์และกฎข้อบังคับความถูกต้องของข้อมูลได้อย่างชัดเจนและกว้างขวางมากที่สุดวิธีหนึ่ง

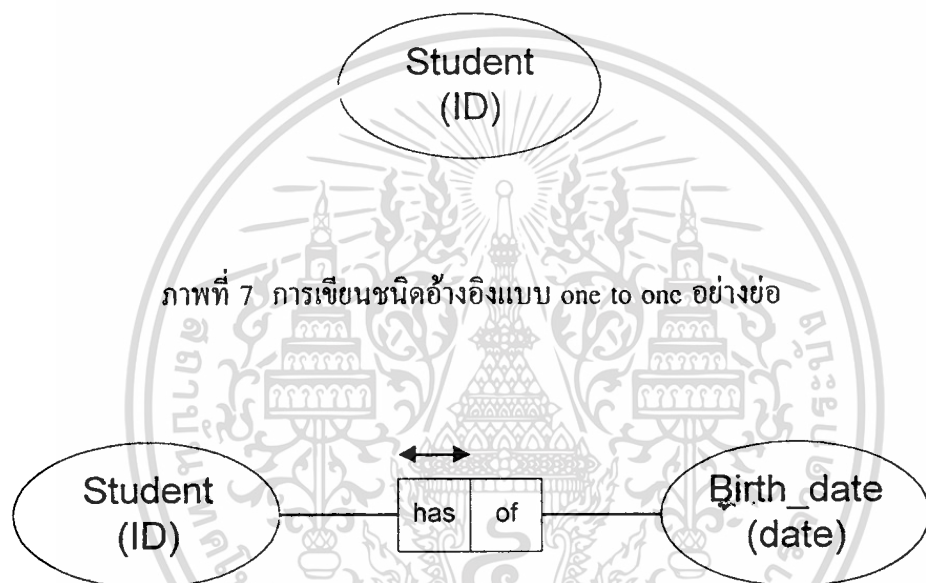
องค์ประกอบของแบบจำลองในแอม



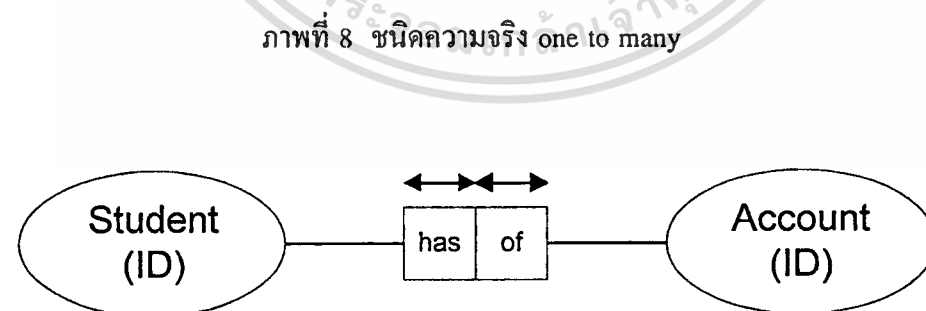
ภาพที่ 5 ชนิดความจริง



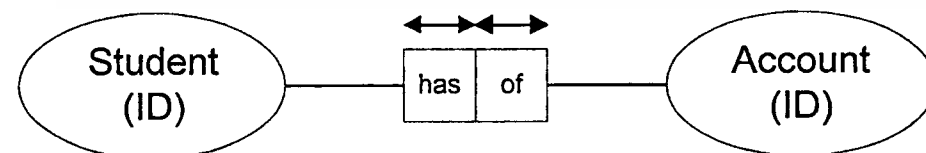
ภาพที่ 6 ชนิดอ้างอิง



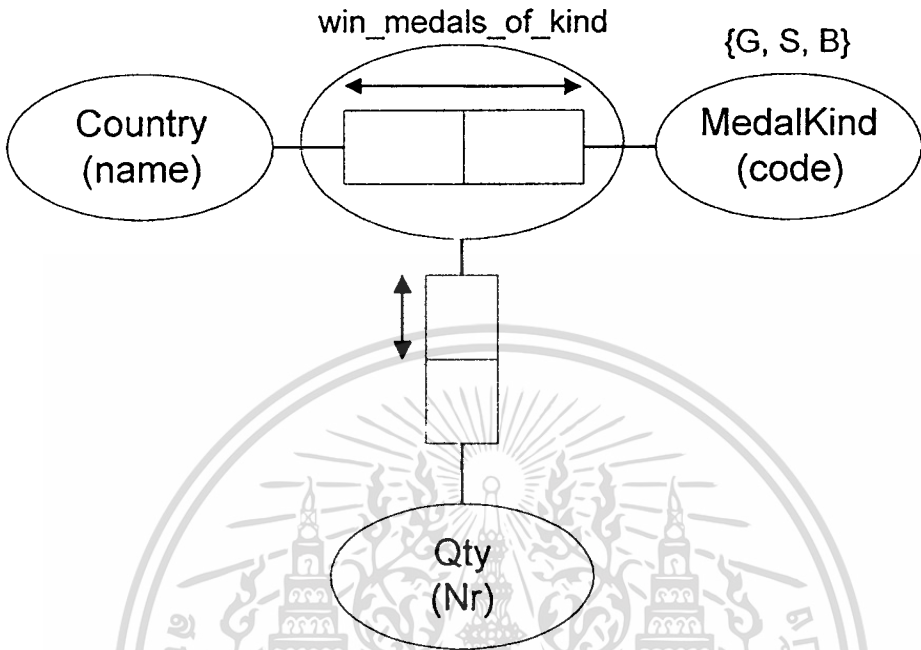
ภาพที่ 7 การเขียนชนิดอ้างอิงแบบ one to one อย่างย่อ



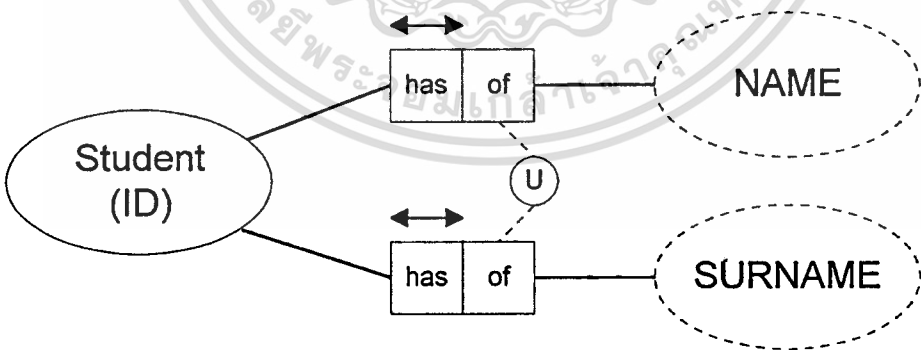
ภาพที่ 8 ชนิดความจริง one to many



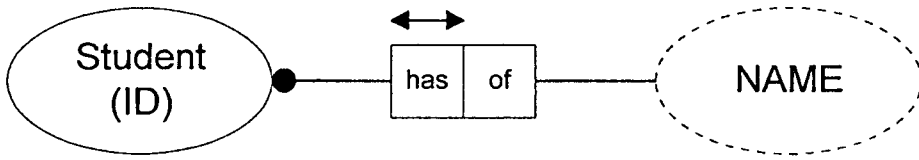
ภาพที่ 9 ชนิดความจริง one to one



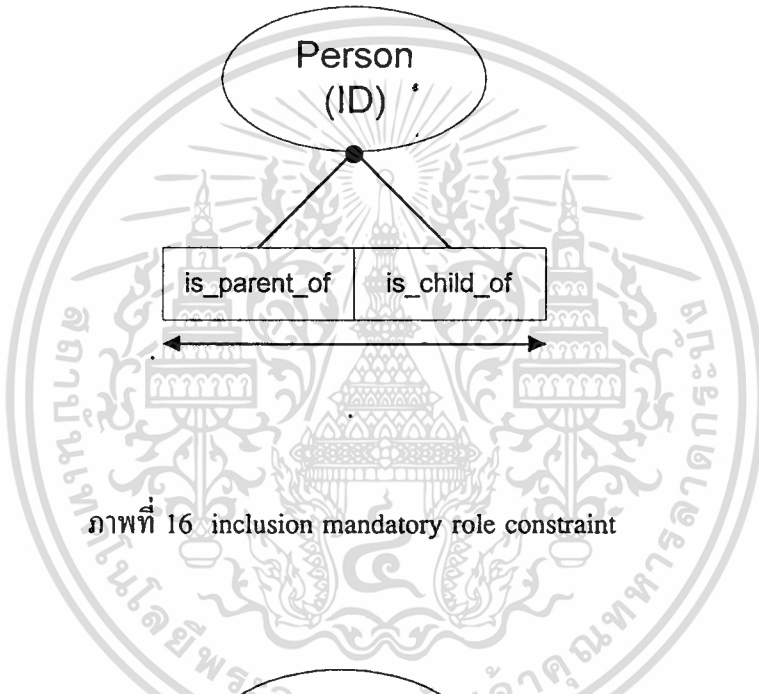
ภาพที่ 13 ชนิดความจริงชนิดเนสต์ (nested)



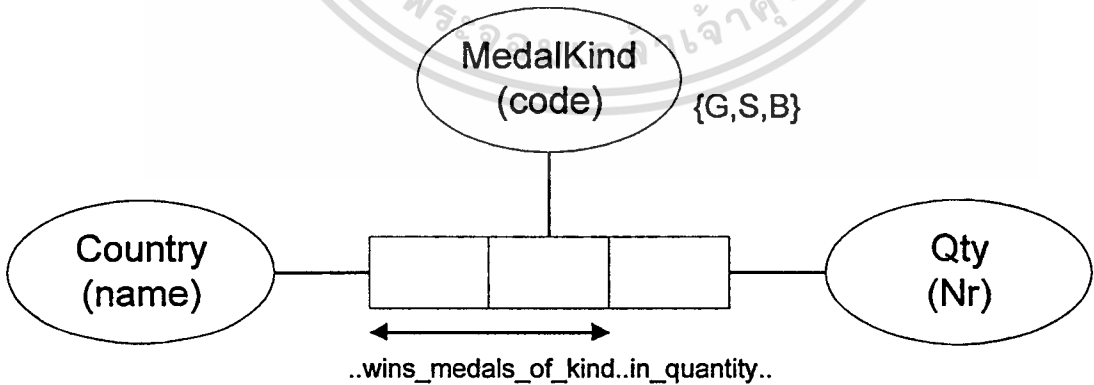
ภาพที่ 14 inter-uniqueness constraint



ภาพที่ 15 mandatory role constraint

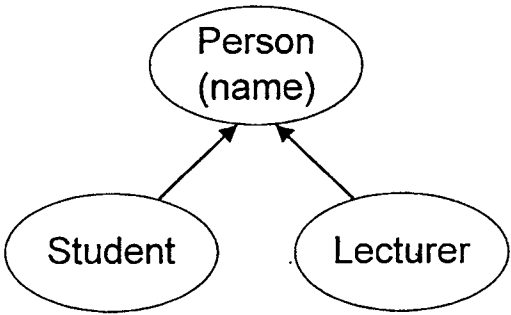


ภาพที่ 16 inclusion mandatory role constraint

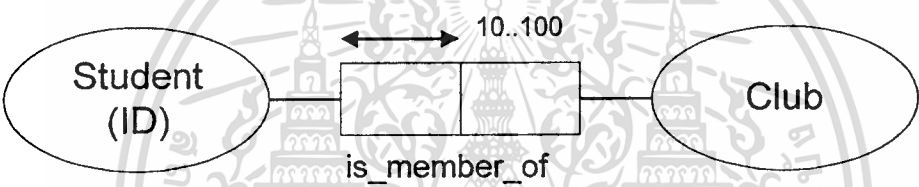


ภาพที่ 17 entity type constraint

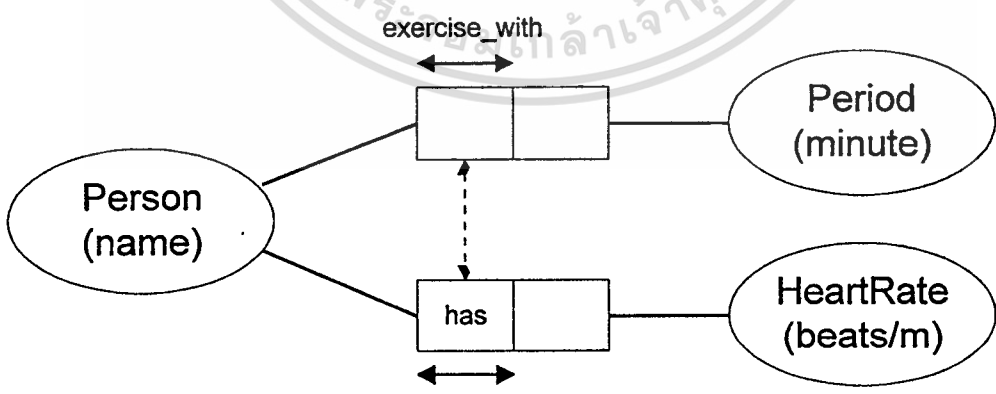
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 18 subtype constraint

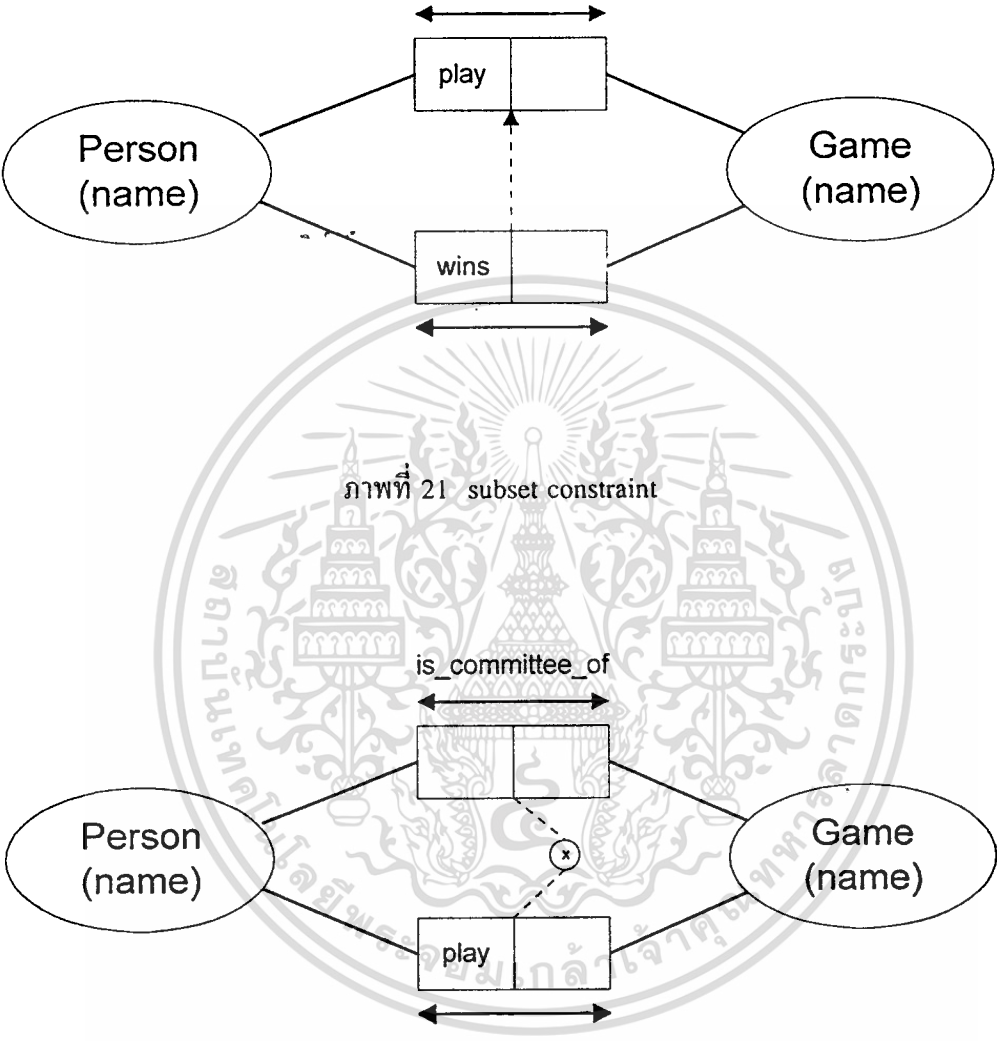


ภาพที่ 19 occurrence frequency constraint



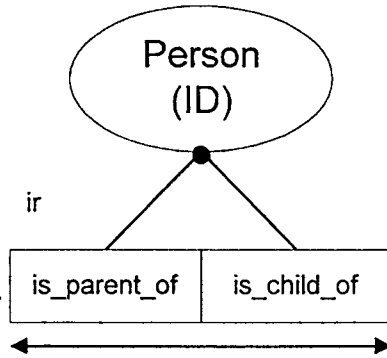
ภาพที่ 20 equality constraint

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

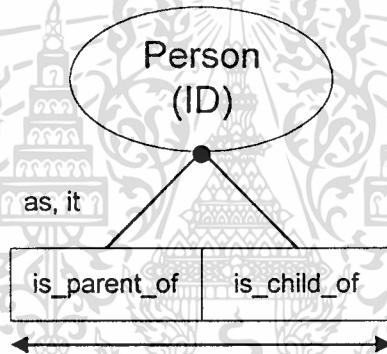


ภาพที่ 22 exclusion constraint

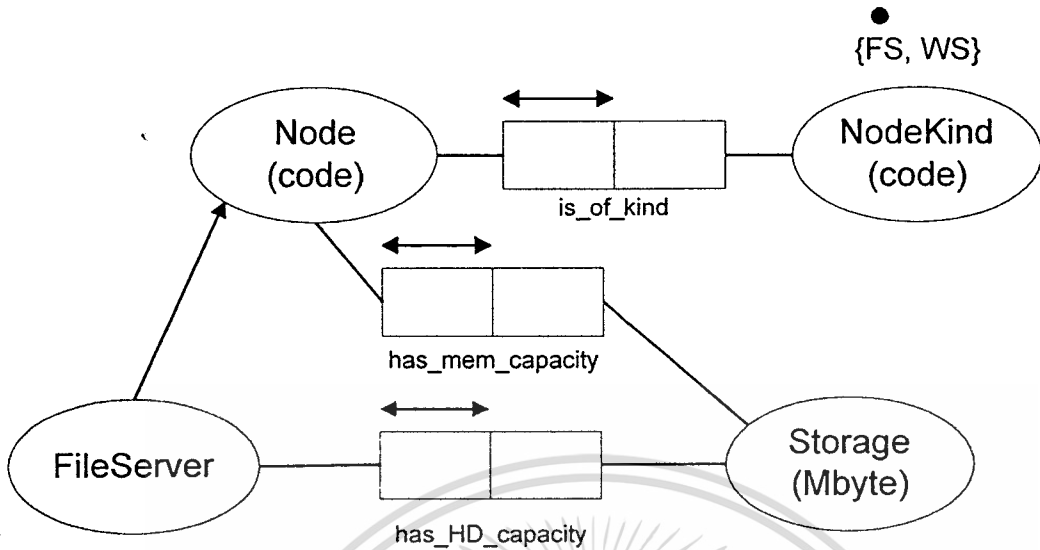
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 23 irreflexive constraint



ภาพที่ 24 asymmetric และ intransitive constraint



ภาพที่ 25 cardinality และ mandatory entity constraint

1. ชนิดเอนติตี้ คือ เซตของสิ่งต่างๆ ที่เราสนใจในระบบ ไม่ว่าสิ่งนั้นจะเป็นรูปธรรมหรือนามธรรม โดยจะใช้รูปวงรีเป็นสัญลักษณ์แทนชนิดเอนติตี้ใดๆ เช่น ชนิดเอนติตี้ Student หมายถึงเซตของนักศึกษา (ดังภาพที่ 3), ชนิดเอนติตี้ Subject หมายถึงเซตของวิชา

2. ชนิดเลเบิ้ล คือ เซตของชื่อที่ใช้อ้างอิงถึงสมาชิกของชนิดเอนติตี้ใดๆ ที่ใช้สัญลักษณ์เป็นรูปวงรีที่มีเส้นรอบวงเป็นเส้นประดังภาพที่ 4 เช่น ชนิดเลเบิ้ล ID หมายถึงเซตของเลขรหัส, ชนิดเลเบิ้ล NAME หมายถึงเซตของชื่อ

3. บทบาท คือความสัมพันธ์ในทิศทางชนิดเอนติตี้สัมพันธ์อยู่ถูกแทนด้วยสัญลักษณ์รูปสี่เหลี่ยม เช่น ในภาพที่ 8 ชนิดเอนติตี้ Student มีบทบาท “enroll_in” กับชนิดเอนติตี้ Subject ซึ่งหมายความว่านักศึกษาแสดงบทบาท “เป็นผู้ลงทะเบียนเรียนในวิชา” ส่วนในทางกลับกันชนิดเอนติตี้ Subject มีบทบาท “enrolled_by” กับชนิดเอนติตี้ Student ซึ่งหมายความว่าชนิดเอนติตี้วิชาแสดงบทบาทเป็น “วิชาที่ถูกลงทะเบียนเรียนโดยนักศึกษา”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ชนิดอ้างอิง (Reference type) คือ เชื้อของความสัมพันธ์ระหว่างสมาชิกของชนิดเอนิตี และสมาชิกของชนิดเลเบิล เช่น ชนิดเอนิตี Student มีความสัมพันธ์ชนิดอ้างอิงกับชนิดเลเบิล ID (ดังภาพที่ 6 และภาพที่ 7) ซึ่งหมายความว่าชนิดเอนิตีนักศึกษาจะใช้รหัสนักศึกษาในการอ้างอิงถึงสมาชิกภายในชนิดเอนิตีของมัน

5. ชนิดความจริง คือ เชื้อของความสัมพันธ์ระหว่างสมาชิกของชนิดเอนิตี ซึ่งอาจจะมีขนาดของชนิดความจริงแตกต่างกันได้ตามจำนวนของบทบาทที่กำหนดอยู่ เช่น ชนิดความจริงชนิดอานารี, ไบนารี และเทอนารี ดังที่แสดงไว้ในภาพที่ 11, ภาพที่ 8 และภาพที่ 12 เป็นต้น ส่วนชนิดความจริงที่มีจำนวนของบทบาทประกอบอยู่มากกว่า 3 บทบาทขึ้นไป จะถูกเรียกรวมกันว่าชนิดความจริงชนิดเอ็นนารี (n-ary)

Nested fact type คือ ความสัมพันธ์ที่มีชนิดเอนิตีเป็นชนิดความจริง เช่นในภาพที่ 13 เป็นการแสดง nested fact type ที่มีชนิดความจริง “Country wins_medals_of_kind MedalKind” และชนิดเอนิตี “Qty” เป็นชนิดเอนิตีของมัน โดยความหมายของชนิดความจริงในภาพที่ 13 นั้นเหมือนกับภาพที่ 12 ทุกประการ

6. Uniqueness constraint คือ กฎที่ใช้บังคับถึงบทบาทที่แสดงเอกลักษณ์ของความสัมพันธ์ใดๆ เพื่อบังคับการจับคู่ระหว่างสมาชิกของชนิดเอนิตีใดๆ กับสมาชิกของชนิดเอนิตี หรือสมาชิกของชนิดเลเบิลว่าเป็นไปในรูปแบบใดดังต่อไปนี้

6.1 ความสัมพันธ์แบบ one to many ดังแสดงไว้ในภาพที่ 8 หมายความว่านักศึกษา

หนึ่งคนจะมีวันเกิดได้เพียงวันเดียวเท่านั้น แต่วันที่ใดๆ อาจจะปรากฏเป็นวันเกิดของนักศึกษามากกว่าหนึ่งคนก็ได้

6.2 ความสัมพันธ์แบบ one to one ดังแสดงไว้ในภาพที่ 9 หมายความว่านักศึกษาหนึ่งคนจะมีรหัสผู้ใช้ของระบบคอมพิวเตอร์ได้เพียงหนึ่งรหัสเท่านั้น และในทางกลับกันรหัสผู้ใช้ใดๆ จะต้องเป็นของนักศึกษาเพียงคนเดียวเช่นกัน

6.3 ความสัมพันธ์แบบ many to many ดังแสดงไว้ในภาพที่ 10 หมายความว่านักศึกษาหนึ่งคนจะสามารถลงทะเบียนเรียนในวิชาต่างๆ ได้หลายวิชา และวิชาใดๆ ก็สามารที่จะรองรับการลงทะเบียนของนักศึกษาได้หลายๆ คน แต่นักศึกษาจะไม่สามารถลงทะเบียนวิชาใดๆ มากกว่าหนึ่งครั้งได้

7. Inter-uniqueness constraint มีหน้าที่เช่นเดียวกับ uniqueness constraint แต่ใช้สำหรับในกรณีที่ชนิดเอนติตี้ใดๆ มีความสัมพันธ์กับชนิดเลเวลมากกว่าหนึ่งชนิด โดยชนิดเลเวลเหล่านั้นสามารถที่จะบ่งเอกลักษณ์ของชนิดเอนติตี้ได้ หรือเป็นการบังคับว่าการจับคู่ของสมาชิกในชนิดเลเวลเหล่านั้นจำเป็นที่จะต้องไม่ซ้ำกัน เช่น ดังภาพที่ 13 หมายความว่านักศึกษาคคนหนึ่งจะมีชื่อได้เพียงชื่อเดียว และมีนามสกุลได้เพียงนามสกุลเดียว โดยชื่อของนักศึกษาคคนหนึ่งคนใดอาจจะซ้ำกับชื่อของนักศึกษาคอีกคนได้ หรืออาจจะมีนักศึกษาคที่มีนามสกุลเหมือนกันได้ แต่ว่าชื่อและนามสกุลของนักศึกษาคเมื่อรวมกันแล้วจะต้องไม่ซ้ำกัน เช่น จะมีนักศึกษาคชื่อ “นภพล” นามสกุล “ใจดี” มากกว่าหนึ่งคนไม่ได้

8. Mandatory role constraint ใช้สัญลักษณ์เป็นจุดทึบเชื่อมระหว่างชนิดเอนติตี้และบทบาทของชนิดเอนติตี้ เพื่อแสดงความหมายว่าสมาชิกทุกตัวของชนิดเอนติตี้ที่มันสัมผัสอยู่จะต้องมีข้อมูลในบทบาทนั้นๆ เช่น ดังภาพที่ 14 บังคับว่านักศึกษาทุกคนจะต้องมีข้อมูลชื่อของนักศึกษาอยู่ในฐานข้อมูล

9. Inclusion mandatory role constraint เป็น mandatory role constraint ที่บังคับกับบทบาทของชนิดเอนติตี้ใดๆ มากกว่า 1 บทบาทขึ้นไป เพื่อแสดงความหมายว่าสมาชิกทุกตัวของชนิดเอนติตี้จะต้องแสดงบทบาทภายในกลุ่มของบทบาทที่ถูกบังคับอยู่อย่างน้อยที่สุดหนึ่งบทบาท เช่นดังภาพที่ 16 หมายความว่า บุคคลใดๆ จะต้องต้องมีข้อมูลบุตรของตนหรือข้อมูลบิดามารดาของตนอย่างน้อยที่สุดหนึ่งข้อมูล

10. Entity type constraint ใช้แสดงการบังคับค่าของข้อมูลที่เป็นไปได้สำหรับสมาชิกชนิดเอนทิตีของชนิดเอนทิตีใดๆ เช่น รหัสของชนิดเอนทิตีระบุรางวัลคือ 'G', 'S' หรือ 'B' เท่านั้นดังแสดงไว้ในภาพที่ 17

11. Subtype constraint เป็นการแสดงกลุ่มของสมาชิกของชนิดเอนทิตีใดๆ ที่มีข้อกำหนดในการแบ่งกลุ่มที่ชัดเจน โดยที่สมาชิกของชนิดเอนทิตีข้อใดข้อหนึ่งนั้นควรจะมีบทบาทเฉพาะของตนที่ชนิดเอนทิตีข้อใดข้อหนึ่งไม่ได้มีส่วนแสดงบทบาทนั้นด้วย เช่น การกำหนดชนิดเอนทิตีข้อใดข้อหนึ่ง Student และชนิดเอนทิตีข้อใดข้อหนึ่ง Lecturer ซึ่งเป็นซับเซตของชนิดเอนทิตี Person ดังแสดงไว้ในภาพที่ 18 เนื่องจากการเก็บข้อมูลของระบบนี้ได้อำหนดความสัมพันธ์เพื่อแสดงข้อมูลส่วนบุคคลของบุคคลทุกบุคคลไว้ทั้งหมด เช่น วันเกิด, เพศ, ที่อยู่ และประเภทของบุคคลนั้นๆ ว่าเป็นอาจารย์หรือนักศึกษา เป็นต้น แต่ในระบบฐานข้อมูลนี้เป็นการเก็บข้อมูลของอาจารย์และนักศึกษาภายในมหาวิทยาลัย ดังนั้นจึงมีข้อมูลบางส่วนที่เป็นข้อมูลเฉพาะสำหรับอาจารย์ เช่น ภาควิชาที่สังกัด, เงินเดือน เป็นต้น และข้อมูลบางส่วนก็เป็นข้อมูลเฉพาะของนักศึกษา เช่น ข้อมูลการลงทะเบียนเรียน และข้อมูลการฝึกงาน เป็นต้น

12. Occurrence frequency constraint เป็นกฎที่ใช้บังคับจำนวนครั้งที่สมาชิกของชนิดเอนทิตีใดๆ ในการแสดงบทบาทใดบทบาทหนึ่ง เช่น ดังภาพที่ 19 แสดงความหมายว่าชมรมใดๆ ต้องรับสมาชิกอย่างน้อยเป็นจำนวน 10 คน แต่จำนวนสมาชิกที่รับได้สูงสุดจะต้องไม่เกิน 100 คน

13. Equality constraint เป็นกฎที่ใช้บังคับการแสดงบทบาท 2 บทบาทใดๆ ของชนิดเอนทิตีที่จะต้องแสดงควบคู่กันไปเสมอ ดังแสดงไว้ในภาพที่ 20 ซึ่งหมายความว่า บุคคลใดๆ จะบันทึกข้อมูลช่วงระยะเวลาของการออกกำลังกายได้ จะต้องบันทึกข้อมูลของอัตราการเต้นของหัวใจควบคู่กันไปด้วย และในทางกลับกันจะบันทึกข้อมูลอัตราการเต้นของหัวใจได้ ก็ต้องบันทึกข้อมูลของช่วงระยะเวลาของการออกกำลังกายร่วมด้วยเช่นกัน

14. Subset constraint เป็นกฎที่มีลักษณะคล้ายกับ equality constraint แต่เป็นการบังคับในทิศทางเดียว เช่นดังภาพที่ 21 หมายความว่า บุคคลใดๆ จะชนะการแข่งขันในเกมสกีพหุชนิดใดได้นั้น จะต้องปรากฏว่าบุคคลคนนั้นเป็นผู้เข้าร่วมแข่งขันในเกมสกีพหุชนิดนั้นๆ ก่อน

15. Exclusion constraint เป็นกฎที่มีลักษณะตรงกันข้ามกับ equality constraint กล่าวคือเป็นการบังคับว่าถ้ามีข้อมูลในความสัมพันธ์หนึ่งต้องไม่มีข้อมูลในอีกความสัมพันธ์ที่ถูกบังคับอยู่ ดังเช่นในภาพที่ 22 หมายความว่า บุคคลที่เป็นกรรมการในการตัดสินเกมสกีพหุชนิดใด จะต้องห้ามเป็นผู้เข้าร่วมแข่งขันในเกมสกีพหุชนิดนั้น ในทางกลับกันก็ถูกบังคับเช่นเดียวกัน

16. Cardinality constraint เป็นกฎที่ใช้บังคับจำนวนของสมาชิกเอนติตี้ที่เป็นไปได้ เช่นดังภาพที่ 23 หมายความว่า โหนดที่มีชนิดเป็นไฟล์เซิร์ฟเวอร์ จะมีได้เพียงโหนดเดียวเท่านั้น

Homogeneous binary โฮโมจีนัสไบนารีหมายถึงชนิดความจริงชนิดไบนารีที่เป็นความสัมพันธ์ระหว่างสมาชิกของชนิดเอนติตี้ชนิดเดียวกัน เช่น “Person is_husband_of Person” และกฎข้อบังคับความถูกต้องชนิดต่อไปนี้เป็นชนิดของกฎข้อบังคับความถูกต้องของข้อมูลที่ใช้บังคับกับชนิดความจริงชนิดโฮโมจีนัสไบนารีเท่านั้น

17. Irreflexive constraint เป็นการบังคับว่าไม่มีสมาชิกใดของชนิดเอนติตี้ที่แสดงบทบาทเป็นทั้งประธานและกรรมของประโยคชนิดความจริงภายในประโยคเดียวกัน เช่น ‘John Smith’ is_parent_of ‘John Smith’ เป็นข้อมูลที่ไม่ถูกต้องตามกฎที่กำหนดไว้ดังภาพที่ 24

18. Asymmetric constraint แสดงว่าถ้าหากสมาชิกใดของชนิดเอนติตี้ถูกจับคู่กันในการแสดงบทบาทที่ถูกบังคับด้วย asymmetric constraint แล้ว สมาชิกทั้งคู่ นั้นไม่สามารถที่จะแสดงบทบาทในทิศทางกลับกันได้ เช่น ถ้าหากปรากฏข้อมูล ‘John Smith’ is_parent_of ‘Henry Smith’

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้ว จะไม่สามารถบรรจุข้อมูลความสัมพันธ์ 'Henry Smith' is_parent_of 'John Smith' ลงไปใน
 ชนิดความจริง "Person is_parent_of Person" ได้อีก เช่นดังที่แสดงการบังคับกฎไว้ในภาพที่ 25

19. Intransitive constraint เป็นการบังคับว่าความสัมพันธ์ของชนิดความจริงที่ถูก
 บังคับนั้นไม่สามารถที่จะถ่ายทอดกันได้ เช่น ถ้าหากมีข้อมูล 'John Smith' is_parent_of 'Henry
 Smith' และ 'Henry Smith' is_parent_of 'Anne Smith' ปรากฏเป็นสมาชิกของชนิดความจริง
 "Person is_parent_of Person" อยู่ จะไม่สามารถบรรจุข้อมูลความสัมพันธ์ 'John Smith'
 is_parent_of 'Anne Smith' ลงไปได้อีก การบังคับกฎข้อนี้แสดงไว้ในภาพที่ 25

แบบจำลองในแอมใช้วิธีการกำหนดชนิดเอนิตีแทนสิ่งต่างๆ ที่เราสนใจภายในระบบงาน
 ไม่ว่าสิ่งนั้นจะเป็นรูปธรรมหรือนามธรรม และสมาชิกของชนิดเอนิตีแต่ละสมาชิกจะสามารถระบุ
 ได้ด้วยชื่อหรือเลเบล โดยความสัมพันธ์ระหว่างชนิดเอนิตีกับชนิดเลเบลจะถูกเรียกว่าชนิดอ้างอิง
 ส่วนความสัมพันธ์ระหว่างชนิดเอนิตีและชนิดเอนิตีจะถูกเรียกว่าชนิดความจริง ชนิดเอนิตี,
 ชนิดอ้างอิง และชนิดความจริงจะถูกบังคับด้วยกฎข้อบังคับความถูกต้องของข้อมูลชนิดต่างๆ แต่
 กฎข้อบังคับความถูกต้องของข้อมูลชนิดที่มีความสำคัญอย่างมากต่อการวิเคราะห์และสร้างรีเลชัน
 แนลสกีมีมากที่สุดคือ uniqueness constraint เพราะเป็นกฎที่ใช้แจกแจงแสดงความสัมพันธ์ระหว่าง
 สมาชิกของชนิดเอนิตีว่ามีความสัมพันธ์กันอย่างไร ซึ่งเป็นข้อมูลพื้นฐานสำคัญสำหรับการสร้าง
 รีเลชันแนลสกีมีมาในขั้นตอนของการใช้อัลกอริธึม ONF ส่วนกฎข้อบังคับความถูกต้องของข้อมูล
 ชนิดอื่นๆ จะถูกพิจารณาอีกครั้งภายหลังขั้นตอนของอัลกอริธึม ONF ซึ่งจะถูกกำหนดเป็นกฎข้อ
 บังคับความถูกต้องของข้อมูลชนิดอรรถาธิบายและ database trigger สำหรับใช้บังคับกับตารางฐาน
 ข้อมูลหรือคอลัมน์ที่เกิดจากการแมปบทบาทภายในแผนภาพในแอมที่ถูกกฎข้อบังคับความถูกต้อง
 ของข้อมูลข้ออื่นๆ บังคับอยู่

ขั้นตอนการออกแบบฐานข้อมูลระดับแนวคิด (Conceptual Schema Design Procedure :CSDP)

ขั้นตอนที่ 1 : เขียนประโยคความจริงขั้นพื้นฐานจากตัวอย่างข้อมูล

ประโยคความจริงขั้นมูลฐานหรือ elementary fact มีหมายความว่าประโยคความจริงที่มีขนาดเล็กย่อยที่สุดไม่สามารถแบ่งแยกให้มีขนาดเล็กลงได้ โดยที่ข้อมูลของความสัมพันธ์ไม่สูญหายไป ซึ่งเป็นประโยคที่ใช้อธิบายถึงข้อมูลและความสัมพันธ์ต่างๆ ภายในระบบงาน โดยอาจจะพิจารณาจากตัวอย่างข้อมูลที่จะเป็นข้อมูลจากตัวอย่างรายงาน หรือแบบฟอร์มข้อมูลใดๆ ก็ได้ เช่น

ตารางที่ 1
ตารางแสดงตัวอย่างข้อมูลการเรียนการสอนในมหาวิทยาลัยแห่งหนึ่ง

Room	Day	Time	Lecturer	Subject	StudentID	StudentNAME
A-502	wednesday	8.30-11.30	Jim Miller	CS311	331207	Allan B
					331256	Jones TS
					331175	Kenny L
					..	..
B-203	tuesday	9.00-12.00	Johnny Walker	CT406	321098	Jenet Jackson
					331175	Kenny L
					320119	Ben Asley
					..	..

จากตารางข้อมูลตัวอย่างสามารถเขียนเป็นประโยคความจริงขั้นพื้นฐานได้ดังนี้

- Student ‘331207’ has NAME ‘Allan B’
- Subject ‘CS311’ taught by Lecturer ‘Jim Miller’

Subject ‘CS311’ take class in Room ‘A-502’

เอกสารนี้เป็นเอกสารผลงานวิจัยสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Subject 'CS311' take class on day 'wednesday'

Subject 'CS311' begin class at Time 8.30

Subject 'CS311' end class at Time 11.30

โดยทั้งนี้ผู้เขียนประโยคความจริงขึ้นพื้นฐานจำเป็นจะต้องอ้างอิงกับกฎต่างๆ ที่บังคับต่อข้อมูลเหล่านั้นด้วย เช่นดังกรณีการเขียนประโยคความจริงขึ้นพื้นฐานข้างต้นนี้ จะสามารถแสดงข้อมูลได้อย่างละเอียดครบถ้วนก็ต่อเมื่อวิชาใดๆ จะถูกสอนโดยอาจารย์เพียงหนึ่งท่านเท่านั้น และจะสอนเพียงครั้งเดียวในหนึ่งสัปดาห์ ซึ่งถ้าหากความเป็นจริงของระบบเป็นไปในทางกลับกัน กล่าวคือวิชาหนึ่งๆ อาจจะมีอาจารย์สอนได้หลายท่านและในแต่ละสัปดาห์อาจจะมีการเรียนการสอนวิชานั้นได้หลายครั้ง แต่อาจารย์หนึ่งท่านจะสอนวิชานั้นๆ เพียงหนึ่งครั้งเท่านั้นภายในหนึ่งสัปดาห์ ประโยคความจริงที่กล่าวว่า "Subject 'CS311' begin class at Time 8.30" นั้นก็จะมีรายละเอียดของข้อมูลที่ยังไม่เพียงพอต่อการระบุถึงการเรียนการสอนของวิชา 'CS311' ที่อาจารย์ 'Jim Miller' สอน เพราะวิชานี้ อาจจะมีอาจารย์ท่านอื่นสอนในวันและเวลาอื่นก็เป็นได้ ดังนั้นถ้าหากต้องการจะแสดงถึงวันและเวลาเรียนของวิชา 'CS311' ให้ได้อย่างละเอียดชัดเจนจำเป็นจะต้องระบุถึงชื่อของอาจารย์ผู้สอนด้วย เนื่องจากเราทราบจากกฎระเบียบของระบบงานว่า อาจารย์ท่านหนึ่งจะสอนวิชาใดๆ ได้เพียงหนึ่งครั้งต่อสัปดาห์เท่านั้น เพราะเหตุนี้ประโยคความจริงขึ้นพื้นฐานของระบบงานนี้ที่จะกล่าวถึงวันเวลาเรียนและห้องเรียนของวิชา 'CS311' ก็จะเป็น

Subject 'CS311' taught by Lecturer 'Jim Miller' take class in Room 'A-502'

Subject 'CS311' taught by Lecturer 'Jim Miller' take class on day 'wednesday'

Subject 'CS311' taught by Lecturer 'Jim Miller' begin class at Time 8.30

Subject 'CS311' taught by Lecturer 'Jim Miller' end class at Time 11.30

ดังนั้นแสดงว่าประโยคความจริงขึ้นพื้นฐานที่จะใช้บรรยายถึงข้อมูลภายในระบบงานจะต้องบรรจุเอารายละเอียดของสิ่งต่างๆ ที่เราจำเป็นต้องใช้ในการอ้างอิงถึงสิ่งที่เราสนใจในประโยคนั้นๆ ไว้อย่างครบถ้วน

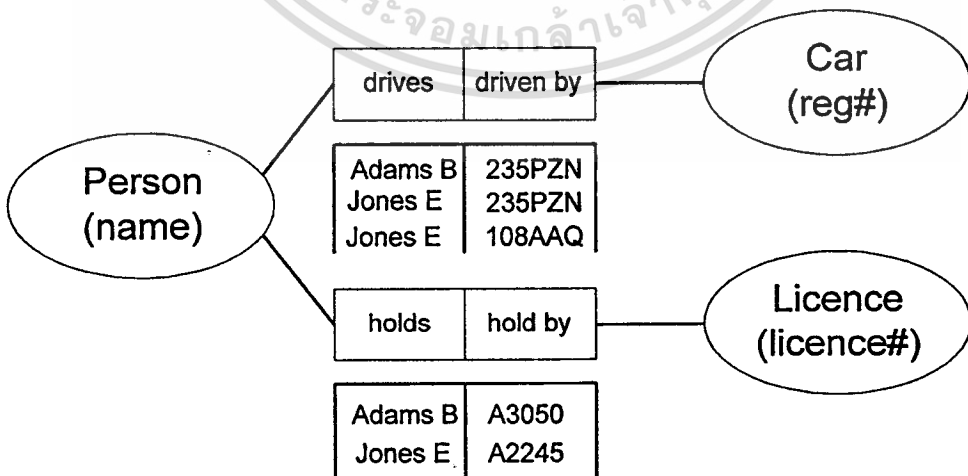
ขั้นตอนที่ 2 : ร่างแบบจำลองข้อมูลระดับแนวคิดและตรรกีสลับแบบจำลองที่ได้ด้วยการแสดงข้อมูลจริงกำกับ

ร่างแผนภาพในแอมเพื่อใช้แสดงแบบจำลองข้อมูลระดับแนวคิดจากประโยชน์ความจริงขั้นพื้นฐานที่วิเคราะห์ได้จากข้อมูลภายในระบบ ให้ประกอบด้วยชนิดเอนติตี้และชนิดเลเบิ้ลต่างๆ รวมทั้งความสัมพันธ์ชนิดความจริงและชนิดอ้างอิงทั้งหมด และตรวจสอบชนิดความจริงภายในแบบจำลองข้อมูลด้วยการพิจารณาตัวอย่างข้อมูลจริงกำกับ เพื่อแสดงว่าชนิดความจริงที่กำหนดขึ้นนั้นสามารถรองรับข้อมูลของระบบได้และสื่อความหมายได้อย่างถูกต้อง เช่น จากรายงานดังตารางที่ 2 สามารถเขียนแผนภาพในแอมได้ดังภาพที่ 26

ตารางที่ 2

ตารางแสดงรายงานข้อมูลภายในระบบงานที่สนใจ

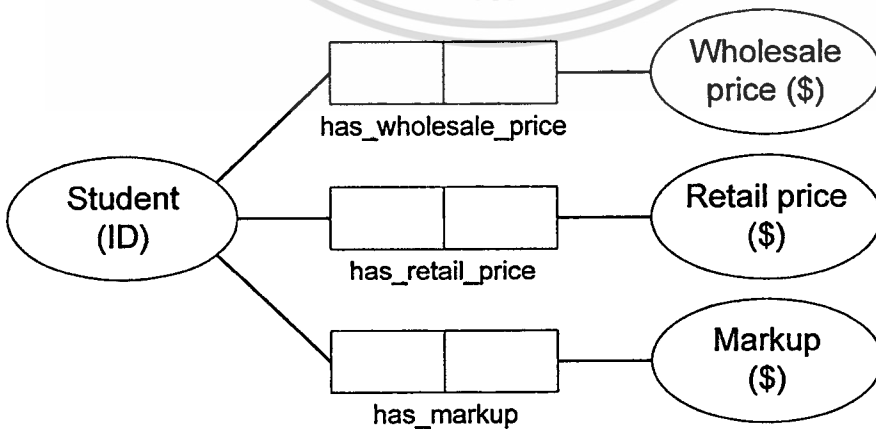
Person	Licence#	CarsDriven
Adams B	A3050	235PZN
Jones E	A2245	235PZN, 108AAQ
...	...	...



ภาพที่ 26 แบบจำลองข้อมูลระดับแนวคิดพร้อมข้อมูลตัวอย่างเขียนแสดงความสัมพันธ์ของข้อมูลจากตารางที่ 2 เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

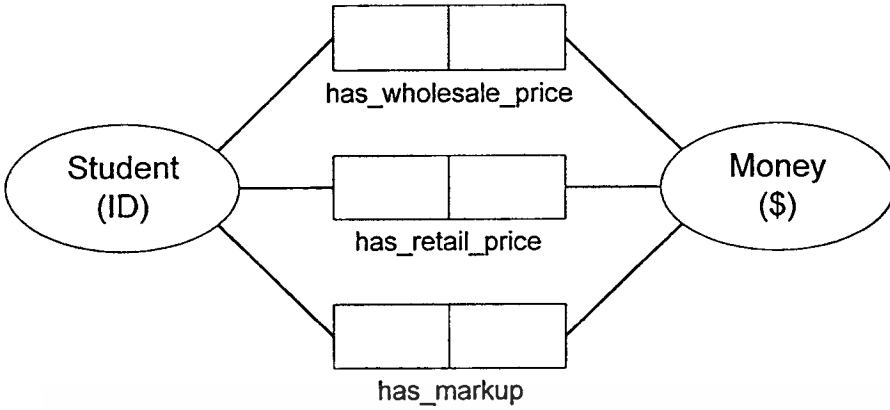
ขั้นตอนที่ 3 : จับกลุ่มชนิดเอนติตี้ที่มีประเภทเหมือนกันและจับกลุ่มบทบาทที่สามารถกำหนดร่วมกันเป็นชนิดความจริงเดียวกันได้ พร้อมทั้งระบุชนิดความจริงที่สามารถทราบได้จากชนิดความจริงอื่นๆ (derived fact type)

สำหรับการเขียนแบบจำลองข้อมูลที่มีจำนวนของชนิดเอนติตี้หรือบทบาทมากเกินไปจนทำให้แผนภาพของแบบจำลองข้อมูลเลอะสับสนและเข้าใจได้ยาก การจัดระเบียบให้ชนิดเอนติตี้ที่มีหน่วยวัดเหมือนกันถูกกำหนดเป็นชนิดเอนติตี้เดียวกัน ก็จะช่วยลดจำนวนของชนิดเอนติตี้ในแผนภาพลงได้มาก ดังเช่นที่แสดงไว้ในภาพที่ 26 และภาพที่ 27 ส่วนการกำหนดชนิดความจริงแบบไบนารี แทนกลุ่มของชนิดความจริงแบบอันวารีก็ช่วยให้แผนภาพดูเข้าใจง่าย และมีผลดีในขั้นตอนของการแปลงเป็นรีเลชันอีกด้วย การลดจำนวนของชนิดความจริงโดยการนำเอาชนิดความจริงที่มีลักษณะของบทบาทเหมือนกันแต่เป็นบทบาทของชนิดเอนติตี้หลายๆ ชนิดที่สามารถกำหนดใหม่เป็นชนิดเอนติตี้เดียวกันได้ก็ช่วยลดขนาดของแผนภาพลงได้เช่นกัน ดังตัวอย่างที่แสดงไว้ในภาพที่ 28 และภาพที่ 29 และการระบุว่าชนิดความจริงใดสามารถหาได้จากชนิดความจริงอื่นนั้น กระทำได้โดยการเติมสัญลักษณ์ '*' ลงเหนือบทบาทที่ต้องการ และอาจจะแสดงสูตรในการหาค่าข้อมูลสำหรับบทบาทนี้ลงไปประกอบด้วยก็ได้ จุดประสงค์ก็เพื่อป้องกันการเกิดปัญหาความไม่มั่นคงของข้อมูล (data inconsistency) และความผิดพลาดอันเกิดจากการแก้ไขข้อมูล (update anomaly) ที่จะเกิดขึ้นแก่ฐานข้อมูล ถ้าหากนำชนิดความจริงที่สามารถหาได้จากชนิดความจริงอื่นๆ มาสร้างเป็นตารางฐานข้อมูล แล้วไม่มีการจัดการที่ดีพอ

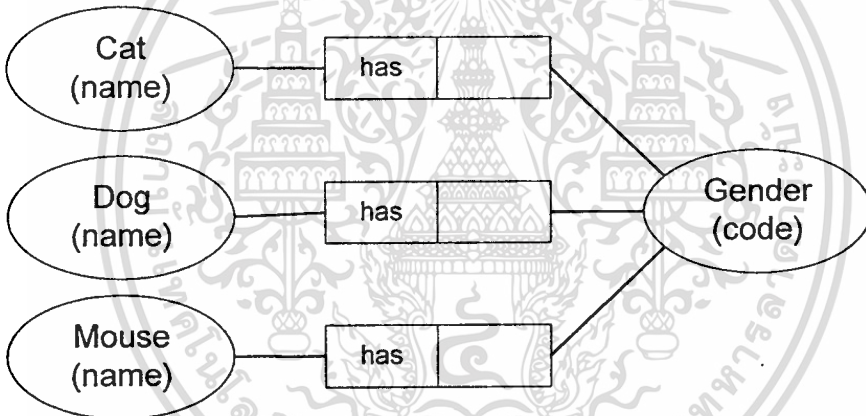


ภาพที่ 27 แผนภาพแสดงแบบจำลองข้อมูลตัวอย่าง

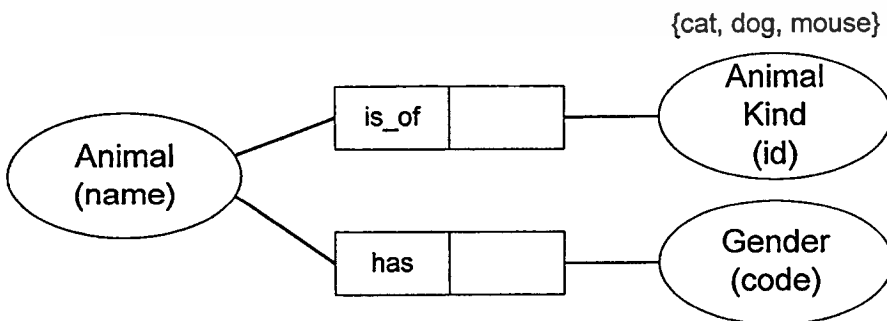
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และห้ามเผยแพร่โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 28 แผนภาพที่ปรับปรุงจากแผนภาพในภาพที่ 27

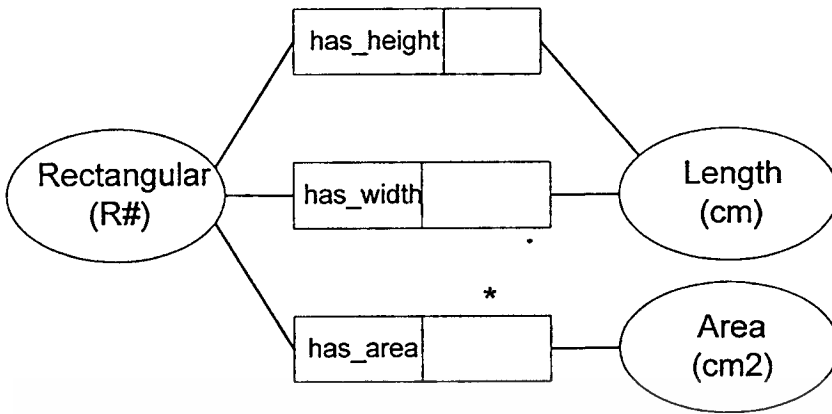


ภาพที่ 29 แผนภาพแสดงแบบจำลองข้อมูลตัวอย่าง



ภาพที่ 30 แผนภาพที่ปรับปรุงจากแผนภาพในภาพที่ 29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 31 แสดงการระบุ derived fact type

ขั้นตอนที่ 4 : ใส่เครื่องหมาย uniqueness constraint

สืบเนื่องมาจากกฎของฐานข้อมูลแบบรีเลชันแนลที่บังคับว่าข้อมูลที่ปรากฏในฐานข้อมูลจะต้องเป็นข้อมูลที่ไม่ซ้ำกัน ดังนั้นอย่างน้อยที่สุดจึงจำเป็นที่จะต้องกำหนดกฎข้อบังคับความถูกต้องของข้อมูลชนิด uniqueness สำหรับทุกชนิดความจริงและชนิดอ้างอิงเพื่อที่จะแสดงว่าในแต่ละชนิดความจริงหรือชนิดอ้างอิงใดๆ นั้น ข้อมูลของชนิดเอนติตี้หรือชนิดเลเบลใดถูกใช้เป็นข้อมูลในการชี้ความเป็นเอกลักษณ์ของความสัมพันธ์นั้นๆบ้าง

ขั้นตอนที่ 5 : ตรวจสอบความถูกต้องของขนาดชนิดความจริง

อาจจะเป็นไปได้ที่ในขณะที่เริ่มต้นกระบวนการออกแบบฐานข้อมูลนั้น ผู้ออกแบบได้กำหนดขนาดของชนิดความจริงผิดไปจากที่ควรจะเป็น เช่น อาจจะมีขนาดที่ยาวเกินไปทำให้การเก็บข้อมูลเป็นไปอย่างซ้ำซ้อนซึ่งจะผิดวัตถุประสงค์ในการออกแบบ หรืออาจจะสั้นเกินไปทำให้สื่อความหมายของข้อมูลในระบบงานได้ไม่ครบถ้วน ซึ่งขั้นตอนที่จะใช้ในการตรวจสอบขนาดของชนิดความจริงมีดังนี้คือ

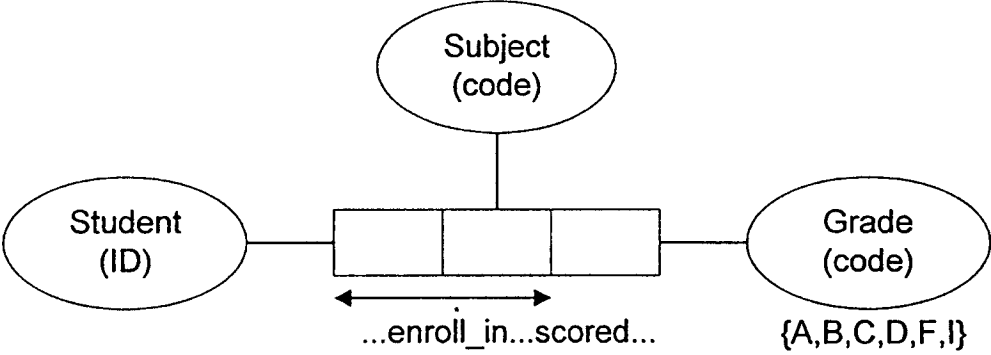
1. ใช้พื้นฐานความรู้เดิมเกี่ยวกับระบบงาน และความรู้สึกรักของผู้ออกแบบในการตรวจสอบว่า อาจจะมีข้อมูลใดขาดหายไปหรือไม่

หากลองแยกชนิดความจริงใดๆ ออกเป็นชนิดความจริงที่ย่อยลง ถ้ามีข้อมูลใดหายไปแสดงว่าชนิดความจริงนั้นมีขนาดที่ถูกต้อง จึงไม่สามารถที่จะแยกออกเป็นชนิดความจริงที่มีขนาดย่อยลงได้ เช่น ตารางที่ 3 และภาพที่ 32 ซึ่งเป็นตารางข้อมูลและแผนภาพที่ใช้แสดงชนิดความจริง “Student enroll_in Subject scored Grade” ซึ่งเป็นชนิดความจริงชนิดเทนารี หากจะลองแยกชนิดความจริงนี้ออกเป็นชนิดความจริงชนิดไบนารีเป็น “Student enroll_in Subject” และ “Student scored Grade” แล้ว เมื่อพิจารณาข้อมูลตัวอย่าง ‘Adams’ enroll_in ‘CS112’ scored ‘B’ จะพบว่าข้อมูลจะถูกแบ่งออกเป็นสองส่วนคือ ‘Adams’ enroll_in ‘CS112’ และ ‘Adams’ scored ‘B’ ซึ่งข้อมูล ‘Adams’ scored ‘B’ นี้ ไม่สามารถชี้ชัดได้ว่า Adams ได้เกรด B จากการเรียนวิชาอะไร เพราะจากข้อมูลตัวอย่างจะเห็นได้ว่า Adams ลงทะเบียนเรียนหลายวิชา และแต่ละวิชาก็ได้เกรดต่างๆ กัน ดังนั้นชนิดความจริง “Student enroll_in Subject scored Grade” มีขนาดที่ถูกต้องดีอยู่แล้ว

ตารางที่ 3

ตารางตัวอย่างข้อมูลผลการเรียนของนักศึกษา

Student	Subject	Grade
Adams	CS112	B
Adams	CS110	A
Adams	PD102	B
Brown	PD102	A
...	...	...

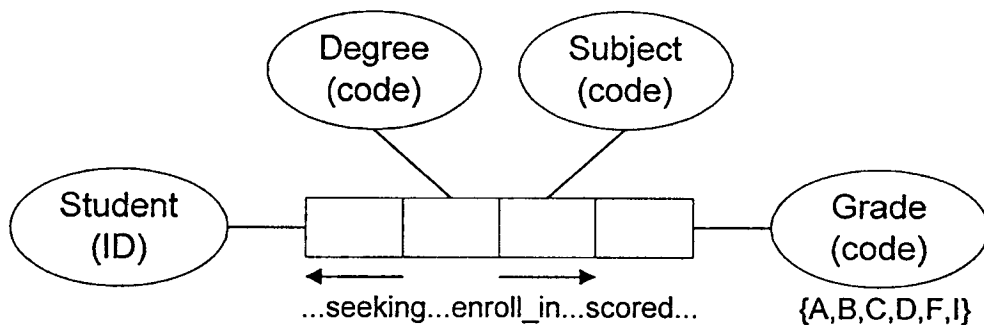


ภาพที่ 32 แผนภาพแสดงแบบจำลองข้อมูลจากความสัมพันธ์ในตารางที่ 3

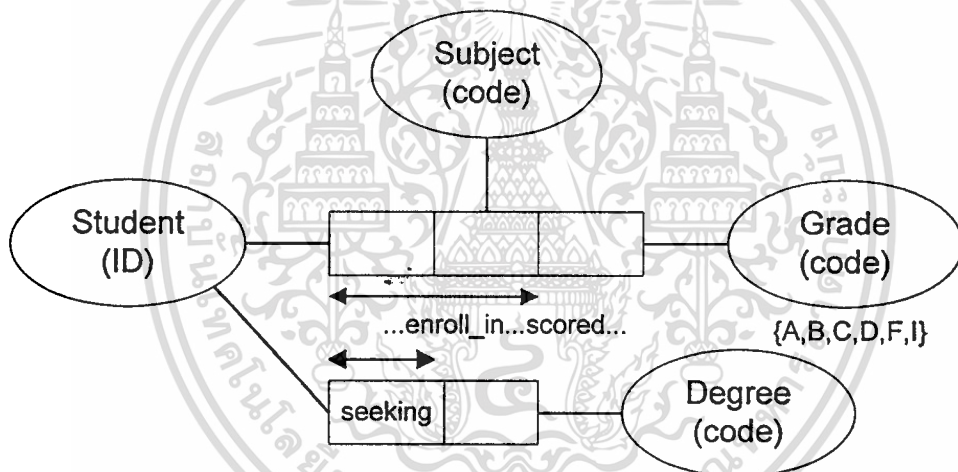
2. แยกชนิดความจริงด้วยการตรวจสอบขนาดของคีย์

2.1 คีย์ (key) คือบทบาทบนชนิดความจริงที่ถูกบังคับด้วย uniqueness constraint เดียวกัน เช่น ภาพที่ 32 ชนิดความจริง “Student enroll_in Subject scored Grade” จะมีชนิดเอนติตี้ Student และชนิดเอนติตี้ Subject เป็นคีย์เช่นเดียวกับภาพที่ 33

2.2 กฎในการแยกชนิดความจริง คือ การที่ชนิดความจริงที่มีขนาดมากกว่าขนาดคีย์ของมันตั้งแต่ 2 บทบาทขึ้นไปสามารถที่จะแยกออกเป็นหลายชนิดความจริงได้ เช่น ในภาพที่ 33 นั้น เป็นชนิดความจริงที่มีขนาดของคีย์เป็น 2 ในขณะที่ชนิดความจริงมีขนาดเป็น 4 ดังนั้นจากการใช้กฎการแยกชนิดความจริง จึงควรแยกชนิดความจริงนี้ออกเป็น 2 ชนิดความจริง คือ “Student enroll_in Subject scored Grade” และ “Student enroll_in Subject for_seeking_in Degree” ซึ่งยังคงมีชนิดเอนติตี้ Student และชนิดเอนติตี้ Subject เป็นคีย์ทั้งคู่ และเมื่อพิจารณาความหมายของชนิดความจริงที่แบ่งแยกออกมาแล้วนั้น ยังคงเห็นได้ว่าข้อมูลของชนิดเอนติตี้ Degree นั้นไม่ได้ขึ้นอยู่กับข้อมูลของชนิดเอนติตี้ Subject ตามที่ใช้ชนิดเอนติตี้ Subject เป็นคีย์แต่อย่างใด ดังนั้นการแปลงชนิดความจริงจากภาพที่ 33 ให้ถูกต้องนั้นจะเป็นไปตามภาพที่ 34



ภาพที่ 33



ภาพที่ 34

8. แยกชนิดความจริงด้วยวิธีโปรเจกชันจอยน์ (projection-join)

แยกชนิดความจริงด้วยวิธีโปรเจกชัน (projection) แล้วทำการรวมชนิดความจริงที่แยกออกมา นั้นกลับไปรวมกันอีกครั้งด้วยวิธีเนเชอรัลจอยน์ (natural join) ถ้าปรากฏว่ามีข้อมูลอื่นเพิ่มขึ้นมา หรือขาดหายไป แสดงว่าชนิดความจริงนั้นไม่อาจที่จะแยกได้ เช่น หากกำหนดชนิดความจริง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

“Student enroll_in Subject for_seeking_of Degree” จากตัวอย่างข้อมูลในตารางที่ 4 เมื่อแยกชนิดความจริงนี้ออกเป็นชนิดความจริง “Student seeking Degree” และ “Student enroll_in Subject” ข้อมูลของชนิดความจริงทั้งสองที่ได้จากวิธีโปรเจกชันจะเป็นไปดังที่แสดงไว้ในตารางที่ 5 และตารางที่ 6 และเมื่อใช้วิธีเนเชอรัลจอยน์กับข้อมูลในตารางทั้งสองแล้วจะได้ข้อมูลดังตารางที่ 7 ซึ่งมีข้อมูลครบถ้วนถูกต้องตามที่มียู่ในตารางที่ 4 ทุกประการ แสดงว่าการกำหนดชนิดความจริง “Student enroll_in Subject for_seeking_of Degree” เป็นชนิดความจริงชนิดเทอนารีนั้นไม่ถูกต้องต้องกำหนดใหม่เป็นชนิดความจริง “Student enroll_in Subject” และชนิดความจริง “Student seeking Degree”

แต่การแยกชนิดความจริง “Student enroll_in Subject scored Grade” เป็นชนิดความจริง “Student enroll_in Subject” และชนิดความจริง “Student scored Grade” ด้วยวิธีโปรเจกชันข้อมูลตัวอย่างจากตารางที่ 4 จะถูกแบ่งออกดังที่แสดงไว้ในตารางที่ 6 และตารางที่ 8 ครั้นทำการจอยน์ 2 ตารางนี้เข้าด้วยกันด้วยวิธีเนเชอรัลจอยน์จะได้ข้อมูลดังตารางที่ 9 ซึ่งพบว่ามีข้อมูลใหม่อีก 5 ข้อมูลถูกเพิ่มเติมขึ้นมา จึงแสดงว่าชนิดความจริง “Student enroll_in Subject scored Grade” ไม่สามารถที่จะแบ่งแยกเป็นชนิดความจริงที่ย่อยกว่านี้ได้

ตารางที่ 4

Student	Degree	Subject	Grade
Adams	BSc	CS112	B
Adams	BSc	CS110	C
Adams	BSc	PD102	B
Brown	BA	CS112	A
Brown	BA	PD102	B
Collins	BSc	CS112	C

ตารางที่ 5

Student	Degree
Adams	BSc
Brown	BA
Collins	BSc

ตารางที่ 6

Student	Subject
Adams	CS112
Adams	CS110
Adams	PD102
Brown	CS112
Brown	PD102
Collins	CS112

ตารางที่ 7

Student	Degree	Subject
Adams	BSc	CS112
Adams	BSc	CS110
Adams	BSc	PD102
Brown	BA	CS112
Brown	BA	PD102
Collins	BSc	CS112

ตารางที่ 9

Student	Subject	Grade
Adams	CS112	B
Adams X	CS112	C
Adams X	CS110	B
Adams	CS110	C
Adams	PD102	B
Adams X	PD102	C
Brown	CS112	A
Brown X	CS112	B
Brown X	PD102	A
Brown	PD102	B
Collins	CS112	C

ตารางที่ 8

Student	Grade
Adams	B
Adams	C
Brown	A
Brown	B
Collins	C

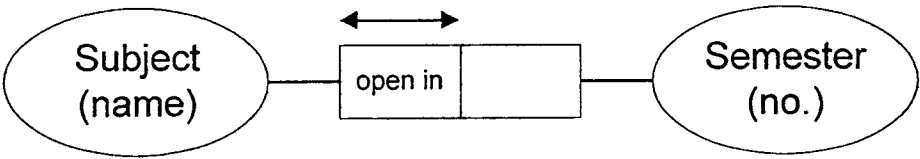
ขั้นตอนที่ 6 : เพิ่มเติมกฎข้อบังคับความถูกต้องของข้อมูลชนิด entity type, mandatory role subtype และ occurrence frequency

สำหรับชนิดเอนทิตีที่เป็นเซตที่จำกัดมีสมาชิกที่สามารถแจกแจงได้ หรือมีข้อกำหนดสำหรับสมาชิกภายในเซตของมัน ให้กำหนดแสดงสมาชิกที่เป็นไปได้หรือข้อจำกัดของสมาชิกชนิดเอนทิตีนั้นๆ ด้วย entity type constraint ดังในภาพที่ 17 และบทบาทที่เป็นบทบาทบังคับของทุกสมาชิกในชนิดเอนทิตีใดๆ ให้เติมเครื่องหมายจุดแสดง mandatory role constraint ดังตัวอย่างในภาพที่ 15

หากชนิดเอนทิตีใดๆ มีกลุ่มของสมาชิกหลายกลุ่มที่ต่างก็มีบทบาทที่เหมือนกันและบทบาทที่แตกต่างกันสามารถแสดงให้เห็นได้อย่างชัดเจนนั้น เพื่อความชัดเจนของแผนภาพแบบจำลองข้อมูลผู้ออกแบบฐานข้อมูลควรที่จะกำหนดชนิดเอนทิตีซ้ำใหม่จากชนิดเอนทิตีนั้น ดังตัวอย่างในภาพที่ 18

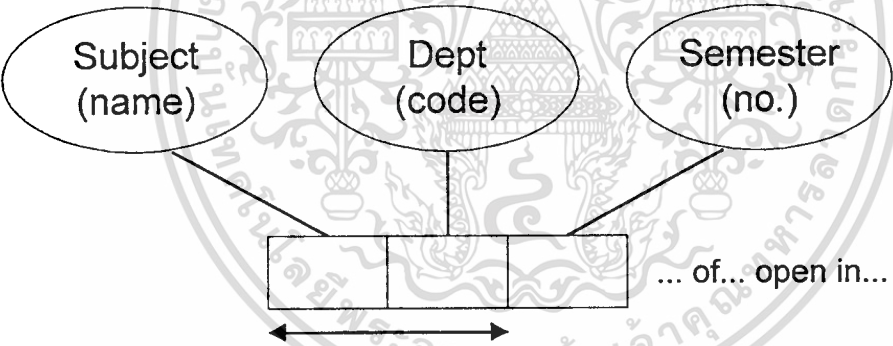
ขั้นตอนที่ 7 : เลือกตัวบ่งเอกลักษณ์ (identifier) สำหรับชนิดเอนทิตี

ในบางกรณีการออกแบบฐานข้อมูลไม่ได้กระทำเป็นระบบใหญ่ในคราวเดียว แต่มีการออกแบบในระดับย่อย เช่น แผนก หรือหน่วยงานย่อยเสียก่อนจึงออกแบบเป็นระบบฐานข้อมูลรวมในภายหลัง การออกแบบฐานข้อมูลเช่นนี้อาจจะเกิดปัญหาในขั้นตอนหลังได้ ถ้าหากไม่มีการตรวจสอบแบบจำลองข้อมูลในระดับแนวคิดเสียก่อน เพราะการตัวบ่งเอกลักษณ์ที่กำหนดไว้ในระบบงานย่อยอาจจะไม่เพียงพอต่อการบ่งถึงเอกลักษณ์ของชนิดเอนทิตีนั้นๆ เมื่อมองจากภาพรวมของระบบงานใหญ่ เช่น ถ้าวิเคราะห์ข้อมูลภายในภาควิชาหนึ่งๆ พบว่าการอ้างถึงชื่อวิชาใดๆ ก็สามารถรู้ได้ว่าวิชานั้นจะเปิดสอนภายในภาคเรียนที่ 1 หรือภาคเรียนที่ 2 ซึ่งความสัมพันธ์ดังกล่าวนี้สามารถเขียนแผนภาพได้ดังภาพที่ 35



ภาพที่ 35

แต่เมื่อมองโดยรวมในระดับขณะนี้ การอ้างถึงชื่อวิชาเพียงอย่างเดียวไม่อาจที่จะระบุได้ว่าวิชานั้นเป็นของภาควิชาใดและจะเปิดสอนในภาคเรียนใด ในกรณีเช่นนี้ เราจะเห็นได้ว่าจำเป็นต้องอ้างถึงชื่อวิชาและภาควิชาควบคู่กันไปเมื่อต้องการทราบข้อมูลที่เกี่ยวข้องกับวิชาใดวิชาหนึ่ง ดังนั้นตัวบ่งเอกลักษณ์ของวิชาที่ควรจะเป็น ชื่อวิชาและชื่อภาควิชา ดังภาพที่ 36

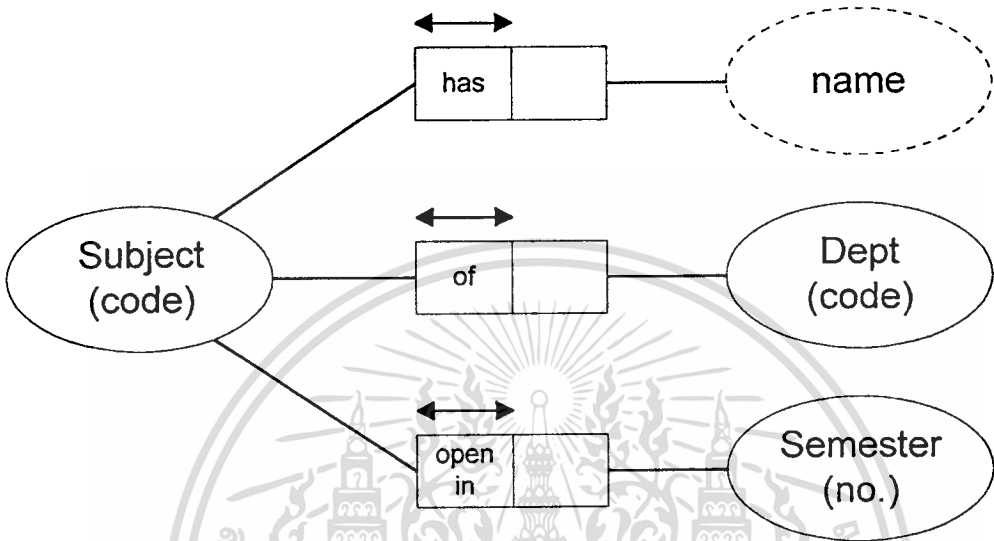


ภาพที่ 36

แต่ในกรณีเดียวกันนี้ ถ้าหากเรานำเอรหัสวิชามาใช้ในการออกแบบด้วย เราจะสามารถทราบข้อมูลที่เกี่ยวข้องกับวิชาใดๆ เช่น ชื่อวิชา, ภาควิชาที่เปิดสอนวิชานั้นๆ, ภาคเรียนที่เปิดสอนได้ทันทีที่อ้างถึงรหัสของวิชานั้น เพราะเหตุว่าภายในขณะนี้ไม่มีวิชาใดที่มีรหัสวิชาใดซ้ำกัน ดังนั้นจะเห็นได้ว่าเราสามารถใช้รหัสวิชาเป็นตัวบ่งเอกลักษณ์ของวิชาได้ ในขณะที่ชื่อวิชาและภาควิชาที่สามารถบ่งเอกลักษณ์ของวิชาได้เช่นกัน แต่การใช้รหัสวิชาเพียงอย่างเดียวจะสะดวกต่อการใช้และความเข้าใจของผู้ใช้ได้มากกว่าดังตัวอย่างในภาพที่ 37 เพราะฉะนั้นการเลือกใช้ตัวบ่งเอกลักษณ์สำหรับชนิด

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอนตีตี้ใดๆ ควรที่จะคำนึงถึงการนำไปใช้ร่วมกับระบบงานใกล้เคียงอื่นๆ ที่กำลังจะพัฒนาขึ้นมา และจะต้องเลือกภายใต้ความสมเหตุสมผลในการใช้งานด้วย



ภาพที่ 37

ขั้นตอนที่ 8 : เพิ่มเติมกฎข้อบังคับความถูกต้องของข้อมูลชนิด equality, exclusion, subset และกฎข้อบังคับความถูกต้องของข้อมูลชนิดอื่นๆ เช่น asymmetric, irreflexive, intransitive, cardinality, mandatory entity constraint

เพิ่มสัญลักษณ์ในแผนภาพเพื่อแสดงกฎข้อบังคับความถูกต้องของข้อมูลในระบบงาน ตามความหมายของชนิดกฎข้อบังคับความถูกต้องของข้อมูลที่อธิบายไว้ในบทก่อน ตามสัญลักษณ์ที่แสดงตัวอย่างไว้ในภาพที่ 20, 21, 22, 23, 24 และภาพที่ 25

ขั้นตอนที่ 9 : ทดสอบว่าแบบจำลองข้อมูลที่ได้ถูกต้องและใช้ได้กับข้อมูลจริงหรือไม่, ตรวจสอบว่าแบบจำลองข้อมูลที่ได้ออกแบบไว้ครบถ้วนสมบูรณ์และกำหนดชนิดความสัมพันธ์ไว้ซ้ำซ้อนหรือไม่

การทดสอบกับข้อมูลจริงนั้น ก็เป็นไปในลักษณะเดียวกับในขั้นตอนที่ 2 โดยการแสดงข้อมูลจริงที่ส่งมาพร้อมกับการพิจารณาแผนภาพแบบจำลองข้อมูล และสังเกตว่าเมื่อบังคับใช้กฎข้อบังคับความถูกต้องของข้อมูลที่กำหนดกับชนิดเอนติตี้และชนิดความจริงใดๆ แล้ว ข้อมูลตัวอย่างจากข้อมูลจริงสามารถที่จะใส่ลงไปในชนิดเอนติตี้และชนิดความจริงที่กำหนดได้หรือไม่ ถ้าหากปรากฏว่าข้อมูลจริงบางข้อมูลไม่สามารถใส่ลงไปได้ หมายความว่าข้อกำหนดใช้กฎข้อบังคับความถูกต้องของข้อมูลนั้นไม่ถูกต้อง เนื่องจากมันได้ทำให้ฐานข้อมูลไม่สามารถรองรับข้อมูลที่มีอยู่จริงได้ ผู้ออกแบบฐานข้อมูลต้องทำการแก้ไขเอกกฎข้อบังคับความถูกต้องของข้อมูลนั้นออก หรือแก้ไขกฎข้อบังคับความถูกต้องของข้อมูลนั้นให้บังคับข้อมูลได้อย่างถูกต้อง

อัลกอริทึม ONF

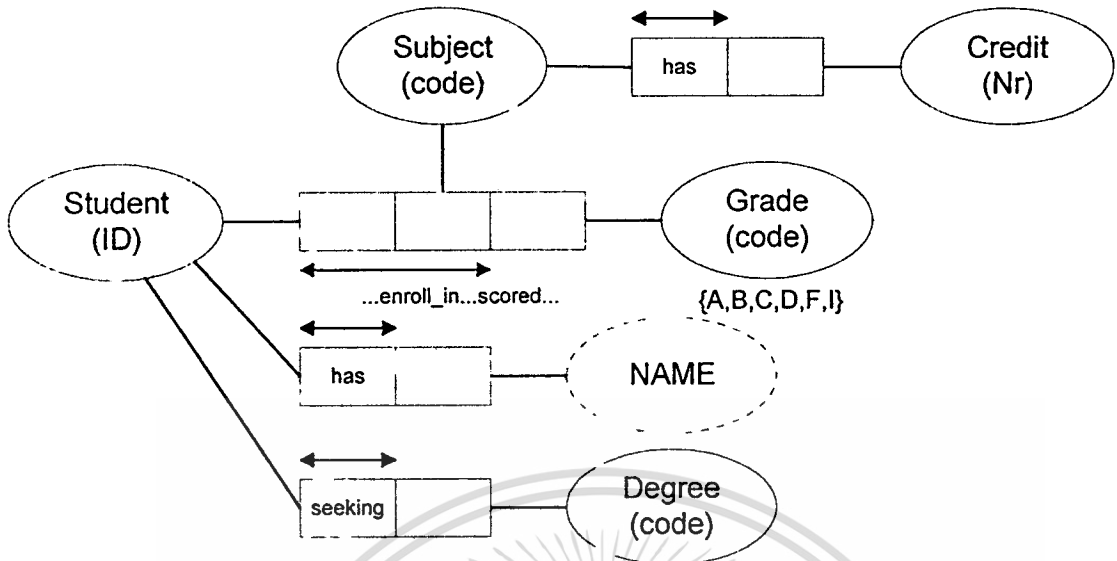
อัลกอริทึม ONF เป็นวิธีที่สามารถช่วยสร้างรีเลชันแนลสกีมาจากแผนภาพในแอมได้โดยตรง โดยที่ตารางภายในรีเลชันแนลสกีมาที่ได้จะอยู่ในรูปของฟิฟท์นอร์มัลฟอร์มทันที ทั้งนี้ไม่ต้องอาศัยการวิเคราะห์ดีเพนเดนซีระหว่างแอททริบิวต์ต่างๆ ภายในตารางเช่นเดียวกับวิธีการนอร์มัลไลเซชัน ซึ่งเป็นวิธีการที่ยุ่งยากซับซ้อน

ขั้นตอนของอัลกอริทึม ONF

1. สร้างตารางความสัมพันธ์หนึ่งตารางสำหรับแต่ละชนิดความจริงที่มีขนาดของคีย์มากกว่า 1 และไม่มีคีย์ร่วมกับชนิดความจริงอื่น ให้เลือกใช้คีย์ที่มีขนาดเล็กที่สุดเป็นไพรมารีคีย์

เช่นตัวอย่างในภาพที่ 38 ชนิดความจริง “Student enroll_in Subject scored Grade” เป็นชนิดความจริงชนิดเอนารี ที่ไม่มีคีย์ร่วมกับชนิดความจริงใดๆ ดังนั้นจึงสร้างเป็นตารางความสัมพันธ์ได้ดังตารางที่ 10 ซึ่งจะใช้แอททริบิวต์ Student_ID และ Subject เป็นไพรมารีคีย์

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 38 แผนภาพแสดงแบบจำลองข้อมูลตัวอย่าง

ตารางที่ 10

ตารางความสัมพันธ์ที่เมืงได้จากภาพที่ 38 ในขั้นตอนที่ 1 ของอัลกอริทึม ONF

Student_ID	Subject	Grade
...	...	...

2. จับกลุ่มชนิดความจริงที่มีคีย์ร่วมกันสร้างเป็นหนึ่งตารางความสัมพันธ์ แล้วใช้คีย์ที่ร่วมกันนั้นเป็นไพรมารีคีย์

ในแผนภาพภาพที่ 38 นั้นมีชนิดความจริงที่ต่างก็ใช้บทบาทของชนิดเอนติตี้ Student เป็นคีย์ดังนี้คือ ชนิดความอ้างอิง “Student has NAME” และชนิดความจริง “Student seeking Degree” ดังนั้นจากกฎข้อที่ 2 ของอัลกอริทึม ONF ชนิดความจริงและชนิดอ้างอิงทั้งสองนี้สามารถสร้างเป็นตารางความสัมพันธ์ได้ดังตารางที่ 11 โดยตารางความสัมพันธ์นี้จะมีแอททริบิวต์ Student_ID เป็นคีย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 11

ตารางความสัมพันธ์ที่เมปได้จากภาพที่ 38 ในขั้นตอนที่ 2 ของอัลกอริธึม ONF

Student_ID	NAME	Degree

8. สำหรับชนิดความจริงที่ยังเหลืออยู่ให้สร้างตารางความสัมพันธ์สำหรับแต่ละชนิดความจริงโดยใช้ชื่อของชนิดความจริงนั้นๆ เป็นไพรมารีคีย์

แผนภาพตัวอย่างในภาพที่ 38 หลังจากผ่านขั้นตอนที่ 1 และ ขั้นตอนที่ 2 ของอัลกอริธึม ONF มาแล้ว ยังคงเหลือความสัมพันธ์ “Subject has Credit” เท่านั้นที่ยังไม่ถูกเมป (map) เป็นตารางความสัมพันธ์ ดังนั้นตารางความสัมพันธ์ที่สร้างได้จากความสัมพันธ์ที่เหลืออยู่คือ ตารางที่ 12 ที่มี Subject เป็นคีย์

ตารางที่ 12

ตารางความสัมพันธ์ที่เมปได้จากภาพที่ 38 ในขั้นตอนที่ 3 ของอัลกอริธึม ONF

Subject	Credit

บทที่ 3

กฎข้อบังคับความถูกต้องของข้อมูล

กฎข้อบังคับความถูกต้องของข้อมูล คือ กฎที่ใช้แสดงสมาชิกที่เป็นไปได้ของชนิดเอนทิตี, ชนิดเลเบล, ชนิดความจริงหรือชนิดอ้างอิงภายในฐานข้อมูลใดๆ

แนวทางการแบ่งชนิดกฎข้อบังคับความถูกต้องของข้อมูลจากงานวิจัยอื่น

1. แนวคิดของ E.F. Codd (1990 : 244-246)

Entity integrity constraint ค่าของคอลัมน์ที่เป็นองค์ประกอบของ primary key จะต้องไม่เป็น NULL

Referential integrity constraint ค่าของ foreign key ในรีเลชันใดๆ จะต้องปรากฏอยู่ใน primary key ของรีเลชันที่เชื่อมโยงกันด้วย referential integrity constraint

Domain integrity constraint ค่าของคอลัมน์ที่มีโดเมนเดียว จะต้องมีชนิดของข้อมูลเหมือนกัน และอยู่ภายใต้ข้อบังคับของโดเมนเหมือนกัน

Column integrity constraint ใช้กับคอลัมน์ที่มีโดเมนเดียวกัน แต่อาจจะมีค่าที่เป็นไปได้ต่างกัน

User-defined integrity constraint เป็นกฎข้อบังคับความถูกต้องของข้อมูลอื่นๆ ที่นอกเหนือจากที่กล่าวไว้ข้างต้น

2. แนวคิดของ Esen Ozkarahan (1990 : 127)

การรักษาความถูกต้องของข้อมูล คือ การป้องกันความผิดพลาดของข้อมูลจากการกระทำของผู้ใช้ที่ได้รับสิทธิในการใช้

Model Specific เป็นกฎข้อบังคับความถูกต้องของข้อมูลพื้นฐานของระบบฐานข้อมูลที่ขึ้นอยู่กับชนิดของแบบจำลองของฐานข้อมูล เช่น

ระบบฐานข้อมูลแบบระดับชั้น (Hierarchy Database)

child node ใดๆ จะมี parent node ได้เพียงโหนดเดียวเท่านั้น

ระบบฐานข้อมูลแบบรีเลชันแนล

- key uniqueness
- entity integrity constraint
- referential integrity constraint

Schema Specific เป็นกฎข้อบังคับความถูกต้องของข้อมูลที่ได้จากการวิเคราะห์และออกแบบฐานข้อมูล เช่น functional dependency และ referential integrity constraint

Application Specific เป็นกฎข้อบังคับความถูกต้องของข้อมูลอื่นๆ ที่ได้จากการวิเคราะห์กฎระเบียบของระบบงาน

3. แนวคิดของ G.M. Nijssen และ T.A. Halpin (1989 : 15-192)

Static integrity constraint เป็นกฎข้อบังคับความถูกต้องของข้อมูลในฐานข้อมูลที่ใช้บังคับกับสถานะของข้อมูล ณ เวลาขณะใดขณะหนึ่ง เช่น ในขณะใดขณะหนึ่ง ประเทศใดๆ จะมีเมืองหลวงได้เพียงเมืองเดียวเท่านั้น โดยกฎข้อบังคับความถูกต้องของข้อมูลชนิด static นี้ยังแบ่งออกเป็นกฎข้อบังคับความถูกต้องของข้อมูลต่างๆ หลายชนิดด้วยกันคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- primary key constraint
- uniqueness constraint
- mandatory role constraint
- entity type constraint (membership constraint & range constraint)
- occurrence frequency constraint
- subtype constraint
- equality constraint
- subset constraint
- exclusion constraint
- asymmetric, irreflexive, intransitive constraint
- cardinality constraint

Dynamic integrity constraint เป็นกฎข้อบังคับกับความถูกต้องของข้อมูลที่เกี่ยวข้องกับการเปลี่ยนแปลงของค่าข้อมูลภายในฐานข้อมูล เช่น เงินเดือนของพนักงานจะต้องไม่มีการปรับลดลง

4. แนวคิดของ Guido Moerkotte และ Peter C. Lockemann (1991 : 670-702)

Internal Constraint

- Model Constraint กฎข้อบังคับกับความถูกต้องของข้อมูลที่แสดงให้เห็นถึงแนวคิดของแบบจำลองข้อมูลที่ใช้
- Schema Constraint กฎข้อบังคับกับความถูกต้องของข้อมูลที่ได้จากการออกแบบฐานข้อมูลภายใต้แนวคิดของแบบจำลองข้อมูล

External Constraint (Consistency constraint)

Consistency constraint หมายถึงกฎข้อบังคับกับความถูกต้องของข้อมูลที่แสดงให้เห็นถึงความหมาย (semantics) ของระบบงานที่ไม่อาจจะแสดงได้โดยแบบจำลองข้อมูล

5. แนวคิดของ David Y.Liu และ Elizabeth A. Unger (1991 : 235-237)

Static Constraint เป็นกฎข้อบังคับความถูกต้องของข้อมูลที่สามารถแสดงได้ด้วย first order predicate logic เพื่อแสดงว่าในสถานะปัจจุบันข้อมูลในฐานข้อมูลสอดคล้องกับกฎข้อบังคับความถูกต้องของข้อมูลที่กำหนดไว้หรือไม่ ซึ่งสามารถตรวจสอบได้จากข้อมูลที่มีอยู่ในฐานข้อมูล

Transition constraint เป็นกฎข้อบังคับความถูกต้องของข้อมูลที่ควบคุม transaction ที่เกี่ยวข้องกับการ update ค่าของข้อมูลที่มีการนำเอาค่าของข้อมูลปัจจุบันและค่าใหม่มาใช้เป็นเงื่อนไขของกฎ เช่น สถานะภาพสมรสของบุคคลใดๆ จะเปลี่ยนจาก 'โสด' เป็น 'หย่า' ไม่ได้

6. แนวคิดของ A. Walshe (1989 : 43-52)

Data Structure Constraint เป็นกฎข้อบังคับความถูกต้องของข้อมูลที่ถูกแสดงโดยโครงสร้างของข้อมูล เช่น เอนิตีใดๆ จะต้องประกอบไปด้วยเซตของแอททริบิวต์ที่กำหนดไว้

Metadata Constraint เป็นกฎข้อบังคับความถูกต้องของข้อมูลที่แสดงให้เห็นถึงความหมายของข้อมูล เช่น referential integrity constraint

Explicit Constraint เป็นกฎข้อบังคับความถูกต้องของข้อมูลชนิดอื่นๆ ที่นอกเหนือไปจาก Data structure constraint และ Metadata constraint ซึ่งยังสามารถแบ่งได้เป็น

- Static Constraint
- Dynamic Constraint

7. แนวคิดของ Christoph F. Eick และ Paul Werstein (1992 : 101-102)

Structural Constraint คือกฎข้อบังคับความถูกต้องของข้อมูลที่สามารถกำหนดได้ด้วยการใช้ไวยากรณ์ของ S-Diagram (S-Diagram คือแบบจำลองข้อมูลระดับแนวคิดสำหรับระบบจัดการฐานข้อมูล DALI เช่น attribute-domain constraint, uniqueness constraint, disjointness constraint, subclass constraint, existence dependency และ functional dependency ดังตัวอย่างที่จะแสดงให้เห็นต่อไป

User-defined constraints กรณีที่กฎข้อบังคับความถูกต้องของข้อมูลไม่สามารถเขียนอยู่ในรูปของ S-Diagram ได้ ผู้ใช้สามารถระบุเป็น Boolean-LISP-functions ได้ โดยระบบจะ return ค่า nil เมื่อเกิดการละเมิดต่อกฎที่สร้างไว้

ตัวอย่างการกำหนดแบบจำลองข้อมูลด้วย S-Diagram

data class Course

attributes: course-id

property: unique #1

type: COURSEID

title

type: TEXT

sections

property: optional, multivalued, onto #2

type: INTEGER

num-sec “number of sections offered”

property: derived using (#(\$kb-match '(section \$SELF \$y))) #3

type: INTEGER

prerequisite

property: multivalued, optional

constraints: (not(\$kb-match '(prerequisite \$ \$)))#4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

type: Course

data class Sections

attributes: section-no

property: unique

constraints: (between \$ 1000 10000)

type: INTEGER

avai-seat

constraints: (> \$ 14)

type: INTEGER

accepted-students

property: multivalued, optional

type: Student

waiting-students

property: multivalued, optional

type: Student

is-taught-by

type: Instructor

constraints: (>= (\$kbget2 \$SELF 'avai-seat)

(#(\$kb-match '(accepted-students \$SELF \$X)))

#5

data class University-Member

attributes: ssn

property: optional, unique

type: SSN #6

name

type: TEXT

birthdate

...

data class Student

subset of University-Member #7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

overlaps with Instructor

attributes:

...

data class Instructor

subset of University-Member

attributes:

...

lexical class SSN

subset of INTEGER

where (and (> \$ 99999999) (> 1000000000 \$))

lexical class COURSEID

กฎข้อที่ 1 เป็นการบังคับ uniqueness constraint บนแอททริบิวต์ course-id

กฎข้อที่ 2 คำสั่ง “onto” เป็นการบังคับว่าแอททริบิวต์ section ไม่สามารถที่จะปรากฏขึ้นได้โดยไม่มี course มาเกี่ยวข้องด้วย

กฎข้อที่ 3 เป็น derivation rule ที่ใช้ในการคำนวณหาจำนวนของ section ของแต่ละ course มาเก็บไว้ในแอททริบิวต์ num-sec

กฎข้อที่ 4 แสดงว่า course ใดๆ ก็ตามไม่สามารถเป็นวิชาบังคับเรียนก่อนของ course เดียวกันนั้น

กฎข้อที่ 5 บังคับว่าจำนวนของนักศึกษาที่ลงทะเบียนไว้ จะต้องไม่เกินจำนวนที่นั่งของแต่ละ section

กฎข้อที่ 6 เป็นการแสดง domain constraint ที่กำหนดให้แอททริบิวต์ SSN อยู่ในช่วงของตัวเลขเก้าหลัก

กฎข้อที่ 7 เป็นการแสดง referential integrity constraint ที่ระบุว่าสมาชิกของคลาส Students จะต้องเป็นสมาชิกของ University-Member ด้วย

ซึ่งการระบุกฎข้อบังคับความต้องการของข้อมูลด้วย S-Diagram นี้ต้องอาศัยความรู้ความเข้าใจในฟังก์ชันของ LISP พอสมควร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. แนวคิดของ Stefano Ceri และ J. Widom (1990 : 566-577)

- static constraint : specify conditions that must be satisfied by a single database state
 ระบุเงื่อนไขที่จะต้องเป็นจริงเสมอในสถานะหนึ่งๆ ของฐานข้อมูล

- dynamic constraints : allow expressing conditions over time ordered sequences of states คือกฎข้อบังคับความถูกต้องของข้อมูลที่จะต้องตรวจสอบเงื่อนไขของสถานะฐานข้อมูลตามลำดับเวลาที่กำหนดไว้

- transition constraint : transition constraint that impose restrictions on pairs of states, the before and after states of a transaction. เป็นกรณีเฉพาะของ dynamic constraint กล่าวคือเงื่อนไขจะถูกตรวจสอบจากสถานะฐานข้อมูลที่ติดกันเท่านั้น

แนวทางการบังคับความถูกต้องของข้อมูลจากงานวิจัยอื่น

ในงานวิจัยชิ้นนี้นอกจากต้องศึกษาแนวทางการแบ่งชนิดกฎข้อบังคับความถูกต้องของข้อมูลแล้ว ยังได้ศึกษาแนวทางการบังคับความถูกต้องของข้อมูลด้วย เพื่อนำข้อมูลที่ศึกษาได้มาขยายฐานความรู้และเพิ่มมุมมองเกี่ยวกับกฎข้อบังคับความถูกต้องของข้อมูล เพราะการที่จะสามารถกำหนดคุณสมบัติของกฎข้อบังคับความถูกต้องของข้อมูล เพื่อการแบ่งประเภทให้ได้อย่างกว้างขวางที่สุดแล้ว จำเป็นต้องศึกษากฎข้อบังคับความถูกต้องของข้อมูลอย่างกว้างขวางและรอบด้านด้วย ซึ่งข้อมูลในหัวข้อนี้ได้ศึกษามาเป็นประโยชน์อย่างมากในการเสริมความรู้ว่าคุณสมบัติใดบ้างที่จำเป็นต่อการบังคับความถูกต้องของข้อมูล

นักวิจัยที่มีผลงานการวิจัยเกี่ยวกับการบังคับความถูกต้องของข้อมูลในระบบฐานข้อมูลต่างก็มีความเห็นเป็นหนึ่งเดียวกัน ในเรื่องความจำเป็นที่ระบบจัดการฐานข้อมูลจะต้องทำหน้าที่เป็นผู้ดูแลความถูกต้องของข้อมูลทั้งหมดภายในฐานข้อมูล เนื่องจากเป็นวิถีทางเดียวที่จะสามารถป้องกันการละเมิดกฎข้อบังคับความถูกต้องของข้อมูลของฐานข้อมูลได้สมบูรณ์ หากมีการกำหนดกฎข้อบังคับความถูกต้องของข้อมูลไว้อย่างถูกต้องและครบถ้วน แต่การที่ระบบจัดการฐานข้อมูลเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะสามารถดูแลรักษาความถูกต้องของข้อมูลได้อย่างครบถ้วนหรือไม่ นั่นก็ขึ้นอยู่กับว่าระบบจัดการฐานข้อมูลนั้นๆ มีคุณสมบัติต่างๆ ที่จำเป็นต่อการบังคับความถูกต้องของข้อมูลครบถ้วนหรือไม่อย่างไร โดยคุณสมบัติที่กล่าวถึงนี้นั้นได้ถูกกำหนดแนวทางจากการศึกษาปัญหาของการบังคับความถูกต้องของข้อมูลซึ่งถูกพิจารณาแบ่งออกเป็น 2 ประเด็นหลักๆ คือ

P1) การเลือกกฎข้อบังคับความถูกต้องของข้อมูลที่เกี่ยวข้องมาใช้ตรวจสอบ เมื่อมีการจัดการข้อมูลในฐานข้อมูล

P2) ปฏิกริยาของระบบจัดการฐานข้อมูลที่ตอบสนองต่อการละเมิดกฎข้อบังคับความถูกต้องของข้อมูล

การบังคับความถูกต้องของข้อมูลบนระบบจัดการฐานข้อมูลแบบรีเลชันแนล

ปัจจุบันระบบจัดการฐานข้อมูลแบบรีเลชันแนลได้พัฒนาการบังคับความถูกต้องของข้อมูลให้ขยายขอบเขตออกไปเป็นอันมาก แต่เดิมนั้นบนระบบจัดการฐานข้อมูลแบบรีเลชันแนลในรุ่นเริ่มแรกมีเพียงการบังคับความถูกต้องของข้อมูลไม่กี่ชนิดเท่านั้น ซึ่งกฎข้อบังคับความถูกต้องของข้อมูลที่มีในขณะนั้นทั้งหมดล้วนเป็นกฎข้อบังคับความถูกต้องของข้อมูลขั้นพื้นฐานของระบบจัดการฐานข้อมูลแบบรีเลชันแนล ซึ่งก็คือ

1. Primary Key Constraint
2. Uniqueness Constraint
3. Mandatory Role Constraint

ถึงแม้กระนั้นระบบจัดการฐานข้อมูลในสมัยนั้นก็ยังขาดความสามารถในการบังคับกฎข้อบังคับความถูกต้องของข้อมูลขั้นพื้นฐานที่จำเป็นอย่างยิ่งคือ referential integrity constraint ในขณะที่ระบบจัดการฐานข้อมูลแบบรีเลชันแนลสามารถบังคับความถูกต้องของข้อมูลได้เพียง uniqueness constraint และ mandatory role constraint นั้น ดังนั้นการบังคับกฎข้อบังคับความถูกต้องของข้อมูลชนิดอื่นๆ ก็ต้องตกเป็นหน้าที่ของนักพัฒนาโปรแกรมประยุกต์ ที่จะต้องนำเอากฎข้อบังคับความถูกต้องของข้อมูลไปฝังไว้ในโปรแกรมประยุกต์ ซึ่งการใช้โปรแกรมประยุกต์บังคับความถูกต้องของข้อมูลดังที่กล่าวมานั้น ได้ก่อให้เกิดข้อเสียหลายประการทั้งต่อความถูกต้องของข้อมูลเอง, ความ

ปลอดภัยของข้อมูล และการพัฒนาระบบฐานข้อมูล รวมทั้งการจัดการฐานข้อมูลของผู้ดูแลฐานข้อมูล ดังจะแสดงต่อไปนี้

1. พัฒนาระบบฐานข้อมูลได้ล้ำหน้าและสิ้นเปลืองทรัพยากรแรงงาน

เนื่องจากการพัฒนาโปรแกรมประยุกต์ที่จำเป็นต้องเขียนโปรแกรมเพื่อรองรับการบังคับความต้องการของข้อมูลด้วยนั้น ทำให้ผู้พัฒนาโปรแกรมประยุกต์จะต้องศึกษาและทำความเข้าใจต่อกฎข้อบังคับต่างๆ ของระบบงาน ร่วมกับการศึกษาความต้องการในการใช้งาน โปรแกรมประยุกต์ของผู้ใช้ ทำให้เสียเวลาในขั้นตอนของการทำความเข้าใจ และการเขียนโปรแกรมจะเป็นไปอย่างซับซ้อน ทำให้การตรวจสอบแก้ไขทำได้ยากลำบาก

2. โปรแกรมประยุกต์จะมีขนาดใหญ่

นอกจากจะต้องบรรจุคำสั่งสำหรับรองรับการปฏิบัติการของผู้ใช้แล้ว ยังต้องเขียนโปรแกรมเพื่อดูแลรักษาความถูกต้องของข้อมูลและ การจัดการกับข้อผิดพลาดต่างๆ ที่เกิดจากการใช้งานข้อมูลภายในฐานข้อมูลเองอีกด้วย

3. การเขียนโปรแกรมจะซ้ำซ้อน

เนื่องจากเมื่อเขียนโปรแกรมประยุกต์ใดๆ ที่มีความจำเป็นต้องจัดการกับข้อมูลในตารางฐานข้อมูลตารางใดตารางหนึ่ง โปรแกรมเหล่านั้นจะต้องบรรจุชุดของคำสั่งที่ใช้ในการดูแลความถูกต้องของข้อมูลสำหรับตารางฐานข้อมูลเดียวกันนั้นเข้าไปในทุกๆ โปรแกรม

4. การแก้ไขกฎข้อบังคับความถูกต้องของข้อมูลกระทำได้ยาก

เพราะต้องแก้ไขโปรแกรมประยุกต์ทุกโปรแกรมที่ใช้งานตารางฐานข้อมูลที่ต้องการจะแก้ไขกฎข้อบังคับความถูกต้องของข้อมูล

5. ระบบฐานข้อมูลไม่สามารถที่จะเชื่อถือได้

ถ้าหากมีโปรแกรมประยุกต์ใดๆ ถูกเขียนขึ้นโดยไม่รักษากฎข้อบังคับความถูกต้องของข้อมูลที่ได้วางไว้ (ไม่ว่าจะตั้งใจหรือไม่ก็ตาม) ก็จะทำให้ระบบฐานข้อมูลไม่สามารถที่จะป้องกันการละเมิดกฎข้อบังคับความถูกต้องของข้อมูลที่ได้ถูกละเลยไปได้ และผู้ใช้ของระบบฐานข้อมูลอาจจะจัดการกับข้อมูลโดยตรงโดยไม่ผ่านโปรแกรมประยุกต์ได้ ซึ่งการจัดการข้อมูลในส่วนนี้จะไม่ผ่านการตรวจสอบกฎข้อบังคับความถูกต้องของข้อมูลทำให้ข้อมูลที่มีอยู่ไม่สามารถที่จะเชื่อถือได้

6. ระบบฐานข้อมูลจะขาดประสิทธิภาพที่ดี

เป็นเรื่องยากสำหรับนักพัฒนาโปรแกรมประยุกต์ที่จะต้องเขียนโปรแกรมในส่วนของการบังคับความถูกต้องของข้อมูล ที่ต้องอาศัยความรู้ความเข้าใจในระบบจัดการฐานข้อมูลร่วมด้วย ถ้าหากนักพัฒนาโปรแกรมประยุกต์ขาดความเข้าใจในระบบจัดการฐานข้อมูลที่ใช้อยู่ ก็อาจจะเป็นเหตุให้ระบบฐานข้อมูลที่พัฒนาขึ้นมาทำงานด้วยประสิทธิภาพกว่าที่ควรจะเป็น

7. การโยกย้ายระบบฐานข้อมูลทำได้ลำบาก และใช้เวลานาน

เมื่อมีความจำเป็นในการเปลี่ยนแปลงสภาพแวดล้อมในการทำงานของระบบฐานข้อมูล เช่น การเปลี่ยนรุ่นคอมพิวเตอร์ ซึ่งมักจะต้องเปลี่ยนระบบปฏิบัติการด้วย อันเป็นที่แน่นอนอยู่แล้วว่า โปรแกรมประยุกต์ที่ใช้งานอยู่บนสภาพแวดล้อมหนึ่งนั้น เมื่อมีการเปลี่ยนแปลงสภาพแวดล้อมย่อมไม่สามารถที่จะใช้งานได้ จะต้องมีการแก้ไขโปรแกรม หรือในบางกรณีอาจจะต้องเขียนโปรแกรมประยุกต์ใหม่เลยทีเดียว ซึ่งเหมือนกับการเริ่มต้นพัฒนาระบบใหม่ ที่ใช้เวลาและทรัพยากรแรงงานที่ไม่แตกต่างกัน

ต่างจากการกำหนดกฎข้อบังคับความถูกต้องของข้อมูลด้วยคุณสมบัติที่มีบนระบบจัดการฐานข้อมูล เนื่องจากหากต้องการจะเปลี่ยนแปลงระบบคอมพิวเตอร์ หรือโปรแกรมระบบปฏิบัติการ ระบบจัดการฐานข้อมูลเชิงพาณิชย์ก็มักจะมีหลายเวอร์ชัน โดยแต่ละเวอร์ชันก็สามารถใช้งานได้บนโปรแกรมระบบปฏิบัติการที่แตกต่างกันออกไป ดังนั้นการเปลี่ยนแปลงสภาพแวดล้อมในการทำงานของระบบฐานข้อมูล โดยมากแล้ว จะใช้เครื่องมือในการถ่ายโอนข้อมูลที่เรียกว่าโปรแกรมอิมพอร์ต และโปรแกรมเอ็กซ์พอร์ตเท่านั้น

คุณสมบัติในการบังคับความถูกต้องของข้อมูลของระบบจัดการฐานข้อมูลเชิงพาณิชย์

1. ระบบจัดการฐานข้อมูล ORACLE

กฎข้อบังคับความถูกต้องของข้อมูลในระบบจัดการฐานข้อมูลอราเคิล หมายถึงกลไกที่ใช้ในการป้องกันกระบวนการนำเข้าข้อมูลที่ไม่อนุญาตสำหรับระบบงาน (invalid data entry) เข้าสู่ฐานข้อมูล ซึ่งรวมถึงกฎข้อบังคับของระบบงาน (business rule) ด้วย ถ้าหากคำสั่ง SQL DML ใดที่ละเมิดต่อกฎที่กำหนดไว้ คำสั่งนั้นจะถูกยกเลิกไปในทันที ซึ่งกฎข้อบังคับความถูกต้องของข้อมูลของระบบจัดการฐานข้อมูลอราเคิล แบ่งออกเป็น 2 ชนิดตามคุณสมบัติและวิธีในการกำหนดดังนี้ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Declarative integrity constraint คือประโยคที่ใช้อธิบายถึงคุณสมบัติต่างๆ ของข้อมูลภายในฐานข้อมูลที่จะต้องเป็นจริงเสมอสำหรับข้อมูลทั้งหมด โดยประกอบไปด้วยกฎข้อบังคับความถูกต้องของข้อมูลชนิดต่างๆ ดังต่อไปนี้

NOT NULL constraint

UNIQUE KEY constraint

PRIMARY KEY constraint

FOREIGN KEY constraint (referential integrity constraint)

CHECK constraint เป็น user defined-rule ในลักษณะของ boolean logic เพื่อตรวจสอบค่าของข้อมูลที่จะเก็บลงในฐานข้อมูลว่าเป็นไปตามเงื่อนไขหรือไม่

Database trigger เป็น stored procedure ที่ถูกกระตุ้นให้ทำงานเมื่อเกิดเหตุการณ์ที่กำหนดต่อตารางฐานข้อมูลที่ยังคงอยู่ ซึ่ง database trigger จะใช้กำหนดว่า transaction จะกระทำการต่อข้อมูลภายในฐานข้อมูลได้อย่างไรบ้าง แต่การใช้ database trigger จะไม่สามารถประกันได้ว่าข้อมูลก่อนหน้าที่จะมีการบังคับใช้ database trigger ใดๆ จะต้องถูกต้องตามเงื่อนไขของ database trigger นั้นๆ

โครงสร้างของ database trigger บน ORACLE

Trigger Statement (Event)

AFTER UPDATE OF parts_on_hand ON inventory

Trigger Restriction

WHEN (new.parts_on_hand < new.reorder_point)

Trigger Action

FOR EACH ROW

DECLARE

NUMBER x;

BEGIN

SELECT COUNT(*) INTO x

FROM pending_orders

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
WHERE part_no = :new.part_no
IF x = 0
THEN
    INSERT INTO pending_orders
    VALUES (new.part_no, new.reorder_quantity, sysdate);
END IF;
END;
```

2. ระบบจัดการฐานข้อมูล SQL Server

SQL Server

มีคุณสมบัติในการรักษาความถูกต้องของข้อมูลดังต่อไปนี้

- NOT NULL**
- UNIQUE INDEX**
- RULE** เป็น user-defined rule ลักษณะ boolean logic เช่นเดียวกับ CHECK ของ ออราเคิล
- sp_primarykey** เป็น system procedure ที่ใช้ในการกำหนด primary key ของตารางฐานข้อมูลใดๆ
- sp_foreignkey** เป็น system procedure ที่ช่วยรักษา referential integrity constraint
- TRIGGER** เป็นโปรแกรมที่ประกอบไปด้วยคำสั่ง SQL และคำสั่งที่ใช้ในการควบคุม ขั้นตอนการทำงานของโปรแกรมเช่นเดียวกันกับภาษาโปรแกรมอื่นๆ และจะทำงานก็ต่อเมื่อเกิด เหตุการณ์ตรงกับเหตุการณ์ที่กำหนด

3. ระบบจัดการฐานข้อมูล INGRES

มีคุณสมบัติในการรักษาความถูกต้องของข้อมูลดังนี้คือ

NOT NULL

INTEGRITY

เป็น user-defined rule เช่นเดียวกับ CHECK ของออรากเคิล

RULE

เป็นกลไกในการบังคับความถูกต้องของข้อมูล ที่มีลักษณะเหมือนกับ database trigger ทั่วๆ ไป คือเป็นโปรแกรมที่ถูกเรียกขึ้นมาตรวจสอบ หากปรากฏเหตุการณ์ที่ระบุไว้

สรุปแนวคิดในการแบ่งประเภทของกฎข้อบังคับความถูกต้องของข้อมูล

จากการศึกษาค้นคว้าวิธีการแบ่งประเภทของกฎข้อบังคับความถูกต้องของข้อมูลของบุคคลที่มีผลงานทางวิชาการในด้านการวางรากฐานและการออกแบบแบบจำลองข้อมูล (data model) และบุคคลที่มีชื่อเสียงในด้านการออกแบบระบบฐานข้อมูล และจากงานวิจัยต่างๆ รวมทั้งจากประสบการณ์การวิเคราะห์และศึกษากฎข้อบังคับความถูกต้องของข้อมูลจากลักษณะของข้อมูลและกฎระเบียบที่จำเป็นในระบบงานขององค์กรต่างๆ ซึ่งจะขอสรุปผลการศึกษาดังต่อไปนี้คือ

การแบ่งประเภทของกฎข้อบังคับความถูกต้องของข้อมูล พิจารณาได้ใน 2 แง่มุมคือ

1. ในเชิงของการออกแบบฐานข้อมูล

ในการออกแบบฐานข้อมูลนั้นการกำหนดกฎข้อบังคับความถูกต้องของข้อมูลก็ขึ้นอยู่กับ

วิธีที่ใช้ในการออกแบบ และแบบจำลองที่ใช้ เช่น

เอกสารเรียนเอกสารเรียนและเอกสารเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีของการใช้รีเลชันนอร์มัลไลเซชัน (Normalization) นั้น แบบจำลองข้อมูลถูกแสดงด้วย FD (Functional Dependency) Diagram ซึ่งสามารถแสดงกฎข้อบังคับความถูกต้องของข้อมูลได้เพียง FD, JD และ MVD เท่านั้น ซึ่งเป็นกฎที่ใช้แสดงความสัมพันธ์ระหว่างข้อมูล ที่จะแปลงมาอยู่ในรูปของ primary key และ uniqueness key constraint ของรีเลชันแนลสกีมา เมื่อกระบวนการในการออกแบบสิ้นสุดลง

กรณีการออกแบบโดยใช้วิธีการของโนแอม ซึ่งใช้แผนภาพโนแอมเป็นแบบจำลองข้อมูลนั้น ผู้ออกแบบสามารถกำหนดกฎข้อบังคับความถูกต้องของข้อมูลได้หลากหลายประเภท เช่น uniqueness, mandatory, subset, subtype constraint ฯลฯ โดยอาศัยการวิเคราะห์กฎต่างๆ จากความสัมพันธ์ของข้อมูลและกฎระเบียบภายในระบบงานไปพร้อมๆ กัน

2. ในเชิงของการกำหนดกฎข้อบังคับความถูกต้องของข้อมูลบนระบบจัดการฐานข้อมูล

ถ้าหากพิจารณาในแง่ของการกำหนดกฎข้อบังคับความถูกต้องของข้อมูลเพื่อใช้บังคับบนระบบจัดการฐานข้อมูลแล้ว อาจจะแบ่งได้เป็น 2 จำพวก คือ

Declarative integrity constraints (กฎแบบอรรถาธิบาย) เป็นกฎข้อบังคับความถูกต้องของข้อมูลที่สามารถกำหนดโครงสร้างของภาษาคำสั่งให้เขียนบรรยายความหมายของกฎได้อย่างชัดเจน ซึ่งยังสามารถแบ่งออกเป็น

- Model Specific Constraint เช่น กฎข้อบังคับความถูกต้องของข้อมูลชนิด primary key, referential integrity และ entity type constraint สำหรับฐานข้อมูลแบบรีเลชันแนล
- Declarative Integrity Constraint คือกฎข้อบังคับความถูกต้องของข้อมูลที่สามารถกำหนดในรูปแบบของประโยคคำสั่งแบบอรรถาธิบายได้นอกเหนือจากกฎข้อบังคับความถูกต้องของข้อมูลที่เป็น Model Specific

Procedural integrity constraint (กฎแบบมีขั้นตอนการดำเนินการ) เป็นกฎข้อบังคับความถูกต้องของข้อมูลที่ใช้ควบคุมการปฏิบัติการที่จะเกิดขึ้นต่อข้อมูลภายในฐานข้อมูลให้เป็นไปตามขั้นตอนและเงื่อนไขที่ถูกต้อง โดยกฎชนิด procedural นี้จะต้องไม่ใช่กฎข้อบังคับความถูกต้องของข้อมูลแบบอรรถาธิบาย

ซึ่งหากมองโดยรวมแล้วจะเห็นได้ว่ากฎที่สามารถกำหนดบนแบบจำลองข้อมูลที่ใช้ในการออกแบบได้ จะช่วยอำนวยความสะดวกและลดความสับสนในการวิเคราะห์ระบบและการตรวจสอบความถูกต้องของระบบฐานข้อมูลที่ออกแบบไปได้มาก และโดยเฉพาะอย่างยิ่ง ถ้าหากกฎที่กำหนดลงในแบบจำลองข้อมูลนั้น สามารถกำหนดบังคับใช้เป็นกฎข้อบังคับความถูกต้องของข้อมูลบนระบบจัดการฐานข้อมูลได้โดยตรง จะเป็นการลดภาระของผู้พัฒนาระบบฐานข้อมูลในการแปลงกฎข้อบังคับความถูกต้องของข้อมูลจากแบบจำลองข้อมูลเป็นชุดคำสั่งภายใน database trigger ซึ่งความจำเป็นในการเขียน database trigger เพื่อบังคับความถูกต้องของข้อมูลเช่นนี้ อาจจะเกิดข้อผิดพลาดจากการเขียนชุดคำสั่งผิดพลาด หรือเกิดจากการแปลความจากกฎข้อบังคับความถูกต้องของข้อมูลที่กำหนดผิดพลาดไปก็เป็นได้ ดังนั้นจึงขอสรุปความเห็นในการแบ่งกฎข้อบังคับความถูกต้องของข้อมูลสำหรับออกแบบและพัฒนาแบบสร้างกฎข้อบังคับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนลดังนี้คือ

กฎข้อบังคับความถูกต้องของข้อมูลควรแบ่งออกเป็น 2 ประเภท คือ

Declarative integrity constraint เป็นกฎข้อบังคับความถูกต้องของข้อมูลที่กำหนดด้วยคำสั่งที่เป็นอรรถาธิบายได้ โดยกฎข้อบังคับความถูกต้องของข้อมูลทุกข้อที่เป็นกฎพื้นฐานของระบบฐานข้อมูลและกฎที่สามารถแสดงได้ด้วยแบบจำลองข้อมูลควรที่จะมีคำสั่งในการกำหนดกฎข้อบังคับความถูกต้องของข้อมูลแบบอรรถาธิบายรองรับ

Procedural integrity constraint เป็นกฎข้อบังคับความถูกต้องของข้อมูลอื่นๆ ที่นอกเหนือจากกฎข้อบังคับความถูกต้องของข้อมูลแบบอรรถาธิบาย

ทั้งนี้หากจะพิจารณากันโดยละเอียดแล้ว declarative integrity constraint ก็เป็นเพียงกรณีพิเศษของ procedural integrity constraint เท่านั้น เนื่องจากกฎข้อบังคับความถูกต้องของข้อมูลบางเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดที่มีเงื่อนไขที่ไม่ซับซ้อนและมีรูปแบบที่ค่อนข้างตายตัว และหากสามารถที่จะออกแบบให้รูปแบบของภาษาที่ใช้กำหนดกฎข้อบังคับความถูกต้องของข้อมูลเหล่านั้นให้เป็นแบบอรรถาธิบายได้ โดยที่ไม่มีจำนวนรูปแบบที่มากมายจนเกินไป declarative integrity constraint นี้ก็จะให้ประโยชน์ได้หลายด้าน เช่น สะดวกต่อการกำหนดกฎข้อบังคับความถูกต้องของข้อมูล และสะดวกสำหรับผู้ดูแลระบบฐานข้อมูล (DBA : Database Administrator) ในการดูแลกฎข้อบังคับความถูกต้องของข้อมูลและฐานข้อมูลในภายหลัง

การบังคับความถูกต้องของข้อมูล

เนื่องจากงานวิจัยนี้ได้ทำการศึกษาและวิจัยเพื่อออกแบบระบบซึ่งเป็นเครื่องมือที่ช่วยในการออกแบบฐานข้อมูล และช่วยในการกำหนดกฎข้อบังคับความถูกต้องของข้อมูลให้อย่างอัตโนมัติ จากจุดประสงค์ดังกล่าวทำให้ข้อมูลที่ได้จากการศึกษาวิธีการบังคับกฎข้อบังคับความถูกต้องของข้อมูลบนระบบฐานข้อมูลแบบรีเลชันแนลมีน้ำหนักในการนำมาใช้มากกว่า ข้อมูลที่ได้จากการศึกษาวิธีการบังคับความถูกต้องของข้อมูลจากงานวิจัยอื่นๆ เพราะว่าการทำงานที่ระบบสร้างกฎบังคับความถูกต้องของข้อมูล โดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนลจะถูกนำมาใช้อย่างเป็นประโยชน์ได้จริงนั้น จำเป็นที่จะต้องมีการจัดการฐานข้อมูลที่มีมาตรฐานพอสมควรรองรับ เหตุเพราะงานวิจัยชิ้นนี้ไม่ได้เป็นงานวิจัยที่ต่อเนื่องกับโครงการงานวิจัยอื่นๆ ที่มีจุดประสงค์ในการพัฒนาต้นแบบของระบบจัดการฐานข้อมูลและระบบย่อยของระบบจัดการฐานข้อมูลสำหรับดูแลความถูกต้องของข้อมูล ดังนั้นจึงต้องอาศัยคุณสมบัติในการบังคับความถูกต้องของข้อมูลจากระบบจัดการฐานข้อมูลเชิงพาณิชย์ในการบังคับความถูกต้องของข้อมูลตามชนิดของกฎข้อบังคับความถูกต้องของข้อมูลที่ได้แบ่งประเภทไว้ ส่วนข้อมูลการบังคับความถูกต้องของข้อมูลที่ได้จากงานวิจัยอื่นๆ นั้น ก็เป็นประโยชน์ต่อการพิจารณาคุณสมบัติที่จำเป็นต่อการบังคับความถูกต้องของข้อมูลของระบบจัดการฐานข้อมูลซึ่งจะได้แสดงไว้ในบทที่ 9

ด้วยเหตุผลที่กล่าวมาข้างต้น จึงขอสรุปเฉพาะคุณสมบัติในการบังคับข้อมูลของระบบจัดการฐานข้อมูลแบบรีเลชันแนลเชิงพาณิชย์ดังนี้คือ

คุณสมบัติในการบังคับความถูกต้องของข้อมูลของระบบจัดการฐานข้อมูลแบบรีเลชันแนล ในปัจจุบันหากแบ่งออกตามวิธีการและรูปแบบของภาษาที่ใช้ในการกำหนดจะแบ่งได้เป็น 2 ชนิดคือ

1. Declarative integrity constraint

เป็นกฎที่มีรูปแบบของคำสั่งตายตัว ซึ่งอาจจะประกอบไปด้วยเงื่อนไขต่างๆ ตามที่ผู้ใช้งานกำหนดได้ แต่ต้องไม่มีคำสั่งที่ใช้ควบคุมขั้นตอนในการดำเนินการ ซึ่งการบังคับกฎข้อบังคับความถูกต้องของข้อมูลชนิดนี้จะบังคับกับข้อมูลทั้งหมดที่อยู่ในตารางฐานข้อมูลที่ถูกกำหนดบังคับอยู่ หากมีการใช้คำสั่งใดๆ จัดการกับข้อมูลในฐานข้อมูลแล้ว ทำให้ข้อมูลในฐานข้อมูลไม่ตรงกับเงื่อนไขที่กำหนดคำสั่งนั้นก็จะถูกยกเลิกไป โดยที่ข้อมูลในตารางฐานข้อมูลจะยังคงเหมือนข้อมูลก่อนที่คำสั่งนั้นจะเริ่มทำงาน

2. Database trigger

เป็นกฎที่กำหนดอยู่ในรูปของชุดคำสั่งและเงื่อนไขที่มีอาจจะมีการดำเนินการและคำสั่งที่ใช้ในการควบคุมการทำงานของชุดคำสั่ง โดยกฎข้อบังคับความถูกต้องของข้อมูลที่ถูกกำหนดเป็นกฎชนิดนี้จะต้องระบุเหตุการณ์ที่อาจจะเกิดขึ้นแก่ฐานข้อมูล แล้วก่อให้เกิดการละเมิดต่อกฎข้อนั้นๆ ได้ โดยระบบจัดการฐานข้อมูลจะคอยสังเกตการจัดการกับข้อมูลในฐานข้อมูลว่าตรงกับเหตุการณ์ที่กำหนดสำหรับ database trigger ไດ แล้วนำเอาชุดคำสั่งภายใน database trigger นั้นมาตรวจสอบเงื่อนไขตามขั้นตอนที่กำหนด หากข้อมูลหรือผลลัพธ์ของคำสั่งใดที่ก่อให้เกิดปัญหาต่อฐานข้อมูลที่ไม่สามารถยอมรับได้ database trigger จะยกเลิกการทำงานของคำสั่งนั้นเช่นเดียวกับคำสั่งที่ละเมิดกฎชนิดอรรถธิบาย และแสดงข่าวสารบอกแก่ผู้ใช้งานว่าคำสั่งหรือการปฏิบัติการนั้นถูกยกเลิกเพราะเหตุใด (การพิจารณาระดับของปัญหาในการละเมิดกฎแต่ละข้อนั้น ขึ้นอยู่กับวิจารณ์ญาณของผู้ออกแบบและพัฒนาระบบฐานข้อมูล และข่าวสารที่จะแจ้งแก่ผู้ใช้ก็เป็นข้อมูลที่ผู้พัฒนาระบบเป็นผู้กำหนดเช่นเดียวกัน) ซึ่งถ้าหากการละเมิดกฎข้อใดไม่ได้ส่งผลให้เกิดปัญหารุนแรงต่อระบบฐานข้อมูล และมีความจำเป็นที่จะต้องยอมรับการผลลัพธ์ของการกระทำเช่นนั้นไว้ ผู้เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัฒนาระบบฐานข้อมูลที่สามารถที่จะสร้างตารางฐานข้อมูลที่ใช้สำหรับเก็บเหตุการณ์และข้อมูลที่จำเป็นในการอ้างอิงการละเมิดกฏดังกล่าว และยินยอมให้การทำงานของคำสั่งนั้นเกิดผลบนฐานข้อมูล



๕

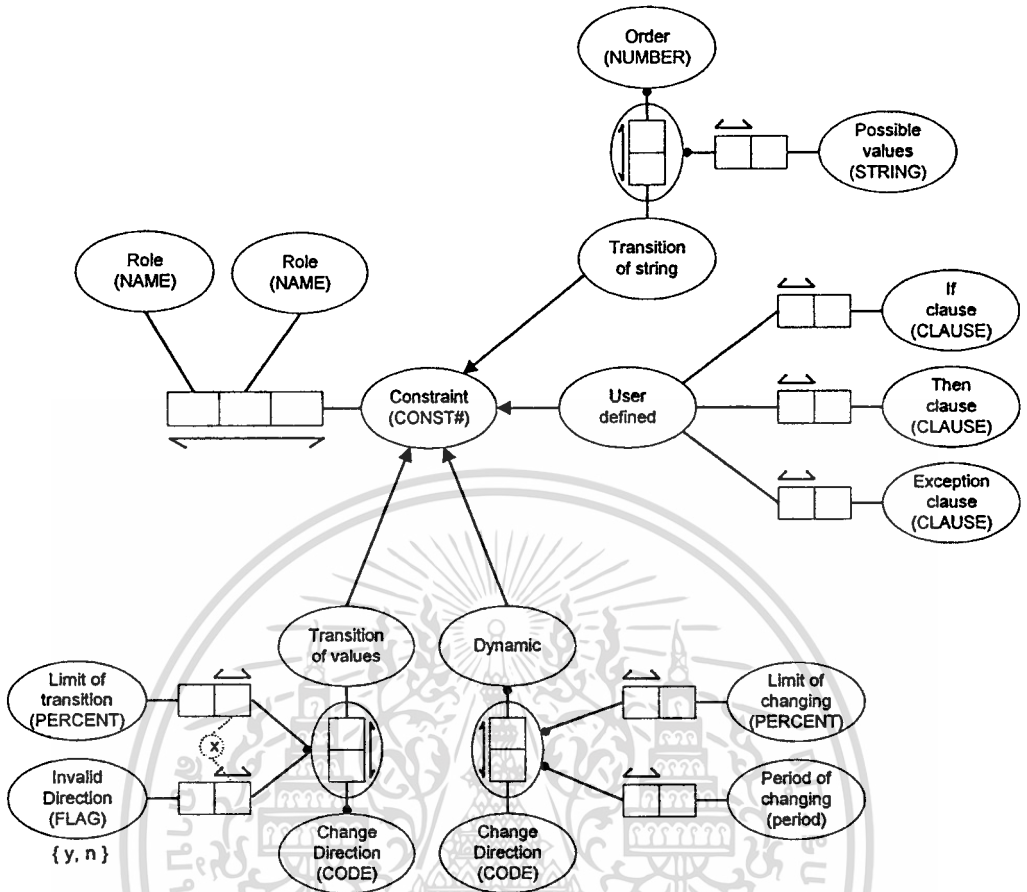
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

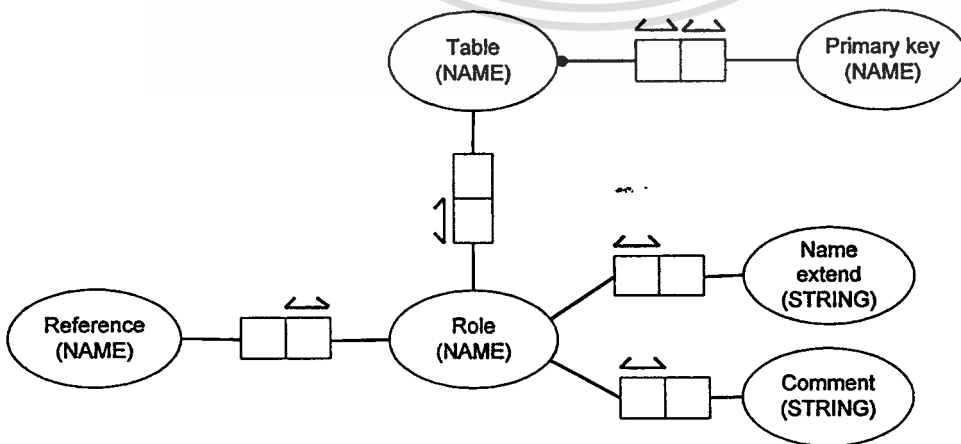
เมตาสกีมา (Metaschema)

ตารางเมตาสกีมาถือเป็นเสมือนหัวใจหลักในการทำงานของระบบสร้างกฎบังคับความถูกต้องของข้อมูลโดยใช้ภาษาราชการสำหรับฐานข้อมูลแบบรีเลชันแนล เพราะการที่จะให้ได้มาซึ่งข้อมูลที่ครบถ้วน ย่อมมีความจำเป็นที่จะต้องสร้างฐานข้อมูลที่มีความสามารถในการเก็บรวบรวมข้อมูลที่จำเป็นต่อการทำงานของระบบได้อย่างถูกต้องสมบูรณ์ เครื่องมือช่วยออกแบบฐานข้อมูลแบบรีเลชันแนลด้วยภาษาราชการนี้ได้ยึดเอาวิธีการออกแบบฐานข้อมูลของวิธีการในแอมเป็นหลัก ซึ่งแต่เดิมวิธีการของในแอมจะแสดงข้อมูลของคอนเซ็ปชวลสกีมาในรูปของแผนภาพในแอม แล้วใช้วิธีที่เรียกว่า Optimal Normal Form Algorithm (ONF) ในการแปลงจากแผนภาพในแอมมาเป็นรีเลชันแนลสกีมาในรูปของตารางความสัมพันธ์ต่างๆ แต่เนื่องจากเครื่องมือช่วยออกแบบฐานข้อมูลแบบรีเลชันแนลด้วยภาษาราชการใช้ประโยชน์ภาษาราชการเป็นเครื่องมือในการออกแบบ ดังนั้นการที่จะนำเอาอัลกอริทึม ONF มาใช้ในการสร้างรีเลชันแนลสกีมาได้ ก็จำเป็นที่จะต้องเก็บข้อมูลของคอนเซ็ปชวลสกีมาให้ได้ถูกต้องครบถ้วนตามที่แผนภาพในแอมสามารถแสดงได้ ซึ่งต้องอาศัยการออกแบบฐานข้อมูลเพื่อเก็บคอนเซ็ปชวลสกีมาจากข้อมูลภายในแผนภาพในแอม โดยฐานข้อมูลนี้จะประกอบไปด้วยตารางความสัมพันธ์ต่างๆ และเนื่องจากตารางความสัมพันธ์เหล่านี้เป็นตารางความสัมพันธ์ที่ใช้เก็บข้อมูลเพื่อการสร้างตารางความสัมพันธ์อีกทีหนึ่งจึงถูกเรียกรวมๆ กันว่า “ตารางเมตาสกีมา”

การออกแบบเมตาสกีมานี้อาศัยการวิเคราะห์ข้อมูลจากโครงสร้างและสัญลักษณ์ของแผนภาพในแอม โดยข้อมูลรายละเอียดต่างๆ ของฐานข้อมูลที่สามารถสื่อได้ด้วยแผนภาพในแอมนั้น เมตาสกีมาก็ต้องสามารถสื่อได้เช่นเดียวกัน แต่เนื่องจากในขั้นตอนการค้นคว้าของงานวิจัยนี้ได้ศึกษาและค้นพบว่าแบบจำลองในแอมยังไม่สามารถแสดงกฎข้อบังคับความถูกต้องของข้อมูลบางชนิดได้ และไม่สามารถระบุรายละเอียดของกฎที่นอกเหนือไปจากสัญลักษณ์ที่มีใช้ในแบบจำลองในแอมได้ อีกทั้งยังศึกษาพบว่ากฎข้อบังคับความถูกต้องของข้อมูลบางกฎในแบบจำลองในแอมสามารถที่จะแสดงรวมกันเป็นกฎๆ เดียวในรูปแบบของภาษาราชการได้ ดังนั้นเพื่อที่จะพัฒนา



ภาพที่ 40 แบบจำลองเมตาสกีมม่า (ต่อ)



ภาพที่ 41 mapping information

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนของการออกแบบเมตาสกีมมา

1. ศึกษาองค์ประกอบและโครงสร้างของแบบจำลองในแอม
2. ออกแบบแบบจำลองข้อมูล เพื่อแสดงความสัมพันธ์ขององค์ประกอบต่างๆ ของแบบจำลองในแอม
3. สร้างตารางเมตาสกีมมาจากแบบจำลองข้อมูลที่ได้
4. ออกแบบฐานข้อมูลจำลองที่มีตัวอย่างของชนิดความสัมพันธ์ และกฎข้อบังคับความถูกต้องของข้อมูลอย่างครบถ้วนมากที่สุด และนำข้อมูลจากฐานข้อมูลจำลองดังกล่าวมาบรรจุในตารางเมตาสกีมมา
5. ตรวจสอบว่าข้อมูลความสัมพันธ์ใด หรือกฎข้อบังคับความถูกต้องของข้อมูลข้อใดไม่สามารถบรรจุลงในตารางเมตาสกีมมาได้
6. แก้ไขแบบจำลองข้อมูลของเมตาสกีมมาใหม่ เพื่อให้สามารถบรรจุข้อมูลของฐานข้อมูลทั้งหมดได้
7. ย้อนไปทำขั้นตอนที่ 5 ใหม่ จนกว่าจะพิสูจน์ได้ว่าตารางเมตาสกีมมาสามารถรองรับข้อมูลของฐานข้อมูล และกฎข้อบังคับความถูกต้องของข้อมูลได้ครบทุกชนิดที่กำหนดขึ้น

ซึ่งในการออกแบบเมตาสกีมมานั้น นอกจากจะต้องแสดงข้อมูลของฐานข้อมูลที่แบบจำลองในแอมสามารถที่จะแสดงได้แล้ว ยังต้องคำนึงถึงความจำเป็นในการแปลงข้อมูลจากแบบจำลองข้อมูลในรูปของภาษาธรรมชาติร่วมด้วย ทำให้การแก้ไขแบบจำลองของเมตาสกีมมาในบางกรณี ต้องเปลี่ยนแปลงแก้ไขรูปแบบของความสัมพันธ์ระหว่างองค์ประกอบของแบบจำลองในแอม เพื่อให้การแสดงความสัมพันธ์และกฎข้อบังคับความถูกต้องของข้อมูลในรูปแบบของภาษาธรรมชาติเป็นไปอย่างกระชับและสื่อความหมายได้ตรงไปตรงมาและชัดเจนที่สุด

การออกแบบเมตาสกีมมาสำหรับเก็บข้อมูลจากแบบจำลองข้อมูลในรูปแบบของประโยคภาษาธรรมชาติไวยากรณ์จำกัดนี้ ได้กำหนดให้ตารางเมตาสกีมมาเก็บข้อมูลของความสัมพันธ์ทุกชนิดในรูปแบบที่เป็นไบนารี โดยพิจารณาว่าชนิดความจริงหรือชนิดอ้างอิงเป็นความสัมพันธ์ (relationship) ที่ประกอบด้วย 2 บทบาท (role) ซึ่งเป็นกิจกรรมของอ็อบเจกต์ 1 หรือ 2 ชนิดใดๆ โดยที่อ็อบเจกต์อาจจะเป็นชนิดเอนทิตี, ชนิดเลเบล, ชนิดความจริง หรือ alias ก็ได้ (เนื่องจากการที่อ็อบเจกต์สามารถเป็นชนิดความจริงได้ ดังนั้นถึงแม้ว่าตารางเมตาสกีมมาจะเก็บข้อมูลความสัมพันธ์ในรูปแบบที่เป็นไบนารี แต่ก็สามารถรองรับข้อมูลความสัมพันธ์ที่มีขนาดมากกว่าไบนารีขึ้นไปได้)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดเอนติตี้ที่กำหนดในแบบจำลองเมตาสกีมามาดังนี้คือ

1. ชนิดเอนติตี้อ็อบเจกต์ (object) เป็นสิ่งต่างๆ ที่สามารถก่อให้เกิดความสัมพันธ์กับสิ่งอื่นๆ ได้ เช่น ชนิดเอนติตี้, ชนิดเลเบล, ชนิดความจริง และ alias

2. ชนิดเอนติตี้บทบาท (role) เป็นกิจกรรมต่างๆ ของอ็อบเจกต์หนึ่งที่ทำต่ออ็อบเจกต์ใดๆ ซึ่งอาจจะเป็นอ็อบเจกต์เดียวกันก็ได้ อันเป็นกิจกรรมที่แสดงให้เห็นถึงความสัมพันธ์ของอ็อบเจกต์ต่างๆ ในแบบจำลองข้อมูล ในกรณีที่ความสัมพันธ์ไม่เป็นชนิดความจริงชนิดอันนารีแล้ว 1 ความสัมพันธ์จะประกอบด้วย 2 บทบาทที่แตกต่างกัน ดังนั้นจึงกล่าวได้ว่า บทบาทเป็นข้อมูลของความสัมพันธ์ที่มีความหลากหลายมากที่สุด ในการออกแบบเมตาสกีม่าจึงได้ยึดเอาบทบาทเป็นแกนในการออกแบบ โดยชื่อของบทบาทจะประกอบไปด้วย ชื่อของอ็อบเจกต์ที่เป็นประธาน, กริยาของอ็อบเจกต์ที่เป็นประธาน และชื่อของอ็อบเจกต์ที่เป็นกรรม จากการกำหนดเช่นนี้ จะทำให้ชื่อของบทบาทนั้นไม่ซ้ำกัน

3. ชนิดเอนติตี้ความสัมพันธ์ (relationship) เป็นกิจกรรมร่วมกันของอ็อบเจกต์ใดๆ ซึ่งอาจจะเป็น ชนิดความจริงหรือชนิดอ้างอิงก็ได้ ความสัมพันธ์ประกอบขึ้นจากบทบาท 1 หรือ 2 บทบาท ชื่อของความสัมพันธ์จะประกอบไปด้วย ชื่อของอ็อบเจกต์ทั้งสองที่มีกิจกรรมร่วมกัน และกริยาหลักของความสัมพันธ์นั้น เช่นเดียวกันกับชื่อของบทบาท

4. ชนิดเอนติตี้กฎข้อบังคับความถูกต้องของข้อมูล (constraint) เป็นกฎที่ใช้บังคับสมาชิกที่เป็นไปได้ของอ็อบเจกต์ใดๆ ชื่อของกฎข้อบังคับความถูกต้องของข้อมูล (CONST#) กำหนดโดยผู้ใช้โดยมีข้อบังคับว่าชื่อของกฎแต่ละข้อจะต้องไม่ซ้ำกัน ภายในแบบจำลองข้อมูลหนึ่งๆ กฎข้อบังคับความถูกต้องของข้อมูลในที่นี้ได้พิจารณาแบ่งออกเป็น 3 ประเภท ตามชนิดของข้อมูลที่กฎใช้บังคับคือ

- กฎข้อบังคับความถูกต้องของข้อมูลที่บังคับกับชนิดของเอนติตี้
- กฎข้อบังคับความถูกต้องของข้อมูลที่บังคับกับบทบาท
- กฎข้อบังคับความถูกต้องของข้อมูลที่บังคับกับความสัมพันธ์

4.1 กฎข้อบังคับความถูกต้องของข้อมูลที่บังคับกับชนิดเอนทิตี เป็นกฎข้อบังคับความถูกต้องของข้อมูลที่บังคับค่าที่เป็นไปได้ของสมาชิกเอนทิตี ได้แก่ กฎข้อบังคับความถูกต้องของข้อมูลชนิด range, membership, cardinality, subtype และ mandatory entity

4.2 กฎข้อบังคับความถูกต้องของข้อมูลที่บังคับกับบทบาท คือกฎข้อบังคับความถูกต้องของข้อมูลที่บังคับกับความสัมพันธ์ใดๆ ทั้งชนิดความจริงและชนิดอ้างอิง แต่บังคับเพียงทิศทางเดียวของความสัมพันธ์นั้นเท่านั้น ได้แก่ กฎข้อบังคับความถูกต้องของข้อมูลชนิด mandatory role, uniqueness, occurrency frequency, equality, subset, cardinality of role, transition และ dynamic

4.3 กฎข้อบังคับความถูกต้องของข้อมูลที่บังคับกับความสัมพันธ์ คือกฎข้อบังคับความถูกต้องของข้อมูลที่บังคับกับชนิดความจริงใดๆ โดยบังคับเหมือนกันทั้งสองทิศทางของความสัมพันธ์นั้น กฎข้อบังคับความถูกต้องของข้อมูลจำพวกนี้ได้แก่ กฎข้อบังคับความถูกต้องของข้อมูลชนิด irreflexive, asymmetric, intransitive และ acyclic

กฎข้อบังคับความถูกต้องของข้อมูลชนิดต่างๆ ต่างถูกกำหนดเป็นชนิดเอนทิตีที่ซับซ้อนของชนิดเอนทิตี Constraint ซึ่งจำแนกได้ด้วยชนิดของกฎข้อบังคับความถูกต้องของข้อมูล (constraint type) นั้นๆ เพื่อที่จะแสดงรายละเอียดในการบังคับข้อมูลของกฎแต่ละชนิดให้เห็นได้อย่างชัดเจนและเป็นประโยชน์ในขั้นตอนการสร้างตารางฐานข้อมูลเมตาสกีมมาในภายหลัง เพราะจะสามารถแก้ปัญหาความสัมพันธ์ของกฎข้อบังคับความถูกต้องของข้อมูลแต่ละชนิดได้เป็นคนละตารางกัน ทำให้สามารถจัดการกับข้อมูลภายในตารางเมตาสกีมมาได้สะดวก โดยกฎข้อบังคับความถูกต้องของข้อมูลแต่ละชนิดมีรายละเอียดดังนี้คือ

Mandatory Entity Constraint มีความสัมพันธ์กับชนิดเอนทิตี Member ซึ่งเก็บเลบิลของสมาชิกชนิดเอนทิตีที่ถูกบังคับว่าจะต้องมีปรากฏเป็นสมาชิกของชนิดเอนทิตีนั้นๆ ซึ่งกฎข้อบังคับความถูกต้องของข้อมูลชนิด mandatory entity ใดๆ อาจจะมี member ได้หลายค่า และ member ใดๆ อาจจะเป็นสมาชิกที่ถูกบังคับของกฎข้อบังคับความถูกต้องของข้อมูลชนิด mandatory entity ได้หลายข้อ

Membership Constraint มีความสัมพันธ์กับชนิดเอนทิตี Member เช่นเดียวกับ mandatory entity constraint แต่เป็นการระบุค่าที่เป็นไปได้ของสมาชิกเอนทิตีที่ถูกบังคับเท่านั้น เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Range Constraint เป็นกฎข้อบังคับความถูกต้องของข้อมูลที่เป็นช่วงไปของ entity type constraint เช่นเดียวกันกับ Membership constraint แต่การระบุค่าทุกค่าที่เป็นไปได้ของสมาชิกชนิดเอนทิตีที่มีชนิดข้อมูลเป็นจำนวนนั้นเป็นเรื่องยุ่งยาก ดังนั้นจึงถูกแบ่งออกเป็นกฎข้อบังคับความถูกต้องของข้อมูลอีกหนึ่งชนิด โดยรายละเอียดของกฎชนิดนี้ก็คือ ค่าสูงสุดหรือค่าต่ำสุดที่เป็นไปได้ของสมาชิกชนิดเอนทิตี โดยอย่างน้อยที่สุดสำหรับการกำหนดกฎชนิดนี้ผู้ใช้ต้องระบุค่าสูงสุดหรือค่าต่ำสุดอย่างน้อยหนึ่งค่าเสมอ

Occurrence frequency Constraint การกำหนดกฎชนิดนี้ ผู้ใช้สามารถกำหนดได้ว่าจะบังคับการปรากฏขึ้นของค่าข้อมูลเป็นเช่นไร เช่น จำนวนครั้งที่บังคับให้เกิด, จำนวนครั้งน้อยที่สุด และจำนวนครั้งมากที่สุดที่ยอมให้เกิดขึ้นได้ ดังตัวอย่างต่อไปนี้ “one Student enroll_in Subject at most 3 Term” ซึ่งหมายความว่า นักศึกษาคนใดคนหนึ่งจะลงทะเบียนวิชาเดิมซ้ำกันได้ไม่เกิน 3 ครั้งเท่านั้น ดังนั้นผู้ใช้จึงต้องกำหนดรายละเอียดในการบังคับข้อมูลทั้งจำนวนครั้งที่ต้องการบังคับ (Frequency) และจุดมุ่งหมายในการบังคับ (Frequency type)

Cardinality Constraint เป็นกฎที่มีรายละเอียดการบังคับข้อมูลคล้ายกับกฎชนิด occurrence frequency เพราะเป็นการบังคับกับจำนวนเช่นเดียวกันแต่ไม่ใช่บังคับจำนวนครั้งของการปรากฏขึ้นของค่าข้อมูลหนึ่งๆ แต่เป็นการบังคับชนิดเอนทิตีถึงจำนวนที่เป็นไปได้ของสมาชิกชนิดเอนทิตี ดังนั้นการเก็บรายละเอียดในการบังคับข้อมูลจึงเหมือนๆ กัน (Qty และ Type of Qty)

Cardinality of Role Constraint เป็นกฎที่บังคับการมีกิจกรรมในบทบาทต่างๆ ของสมาชิกชนิดเอนทิตีใดๆ ว่าสามารถมีกิจกรรมกับบทบาทใดได้กี่บทบาท เช่น “Employee play at least one roles of { ‘Employee own Car’, ‘Employee drive Car’ }” ดังนั้นจึงต้องเก็บข้อกำหนดของกฎชนิดนี้ทั้งจำนวนกิจกรรมที่บังคับ (Number of Role) และ จุดประสงค์ในการบังคับ (Type of Number)

Equality Constraint การกำหนด equality constraint สำหรับบทบาทใดๆ ของชนิดเอนทิตีหนึ่งๆ บทบาททั้งสองที่ถูกบังคับจะต้องมีกฎข้อบังคับความถูกต้องของข้อมูลชนิด equality สัมพันธ์อยู่ และกฎข้อบังคับความถูกต้องของข้อมูลสองข้อนั้นจะถูกแสดงว่าเป็น equality ซึ่งกันและกัน

Subset Constraint ก็มีรายละเอียดเช่นเดียวกับ Equality constraint กล่าวคือ จะต้องรู้ว่าบทบาทใดที่ถูกบังคับเป็นซับเซตของอีกบทบาทหนึ่ง โดยการใช้ความสัมพันธ์และชื่อของกฎข้อบังคับความถูกต้องของข้อมูลโยงไปยังบทบาทที่ถูกบังคับ

Subtype Constraint เป็นการกำหนดข้อจำกัดของสมาชิกชนิดเอนติตี้ซ้ำไต่ไปที่จะประกอบไปด้วยสมาชิกจากชนิดเอนติตี้รูปเปอร์ไทป์ที่มีคุณสมบัติอย่างไรบ้าง โดยชนิดเอนติตี้ Subtype Definition จะเก็บเงื่อนไขในลักษณะของ boolean logic ที่ใช้กำหนดใน WHERE clause ของคำสั่ง SQL

Transition Constraint เป็นกฎที่กำหนดด้วยข้อจำกัดของการเปลี่ยนแปลงค่าของข้อมูล และเนื่องจากเหตุผลในการบังคับกฎชนิดนี้ จึงได้แบ่งกฎข้อบังคับความต้องการของข้อมูลชนิด Transition นี้ออกเป็น 2 ชนิดคือ Transition of String Values และ Transition of Numeric Values เนื่องจากการเปลี่ยนแปลงค่าข้อมูลที่เป็นจำนวนเลขนั้น มีความยืดหยุ่นและครอบคลุมการบังคับได้กว้างกว่าการกำหนดบังคับการเปลี่ยนแปลงค่าข้อมูลที่เป็นข้อความ ซึ่งรายละเอียดในการกำหนดกฎ 2 ชนิดนี้ประกอบไปด้วย

- Transition of String Values เก็บลำดับที่และค่าข้อมูลที่ยังบังคับ
- Transition of Numeric Values เก็บทิศทางของการเปลี่ยนแปลงค่าข้อมูล และข้อจำกัดในการเปลี่ยนแปลง หรือ flag ที่บอกทิศทางในการเปลี่ยนเปลี่ยนนั้นเป็นทิศทางที่ต้องห้ามหรือไม่

Dynamic Constraint มีองค์ประกอบของกฎคล้ายกับ Transition of Numeric Values เพียงแต่มีช่วงระยะเวลาระหว่างการเปลี่ยนแปลงค่าข้อมูลมาใช้บังคับด้วย

User-defined Constraint เนื่องจากเงื่อนไขของกฎสามารถมีได้ไม่จำกัด ดังนั้นข้อมูลที่เก็บลงในตารางเมตาสกีมมา จะเก็บเงื่อนไขลงไปทั้งประโยค ทั้งประโยคที่เป็น if clause ที่เป็นเงื่อนไขที่ทำให้เกิดการตรวจสอบเงื่อนไขต่อไปใน then clause และประโยคคำสั่งที่ปรากฏใน exception clause ที่จะถูกเรียกใช้งานเมื่อมีการละเมิดเงื่อนไขใน then clause

สำหรับ uniqueness constraint ที่ไม่ได้ถูกกำหนดเป็นชนิดเอนติตี้ซ้ำไต่ไปของชนิดเอนติตี้ constraint เนื่องจาก uniqueness constraint เป็นกฎที่บังคับการปรากฏของค่าข้อมูลให้สามารถปรากฏขึ้นได้เพียงหนึ่งครั้งเท่านั้น ซึ่งเป็นข้อกำหนดตายตัวของกฎชนิดนี้ จึงไม่จำเป็นจะต้องสร้างความสัมพันธ์เพื่อเก็บรายละเอียดของกฎชนิดนี้แต่อย่างใด และสำหรับกฎข้อบังคับความต้องการของข้อมูลชนิด mandatory role ก็มีเหตุผลเช่นเดียวกันกับ uniqueness constraint

ดังนั้นจากหลักการแบ่งประเภทของกฎข้อบังคับความต้องการของข้อมูลของแบบจำลอง

ในแอม ทำให้การออกแบบเมตาสกีมมาเพื่อรองรับข้อมูลของกฎข้อบังคับความต้องการของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้เผยแพร่หรือใช้ประโยชน์ในการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดต่างๆ จึงเป็นไปดังที่แสดงไว้ในแผนภาพแบบจำลองข้อมูลเมต้าสกีมา ภาพที่ 39 และภาพที่ 40 โดยกฎข้อบังคับความถูกต้องของข้อมูลแต่ละข้อกำหนดเอกลักษณ์ด้วยชื่อของกฎข้อบังคับความถูกต้องของข้อมูล (CONST#) ส่วนรายละเอียดในการบังคับข้อมูลของกฎข้อบังคับความถูกต้องของข้อมูลชนิดต่างๆ นั้น ได้กำหนดชนิดความจริงต่างๆ ไว้แตกต่างกัน สำหรับกฎข้อบังคับความถูกต้องของข้อมูลแต่ละข้อ เนื่องจากกฎข้อบังคับความถูกต้องของข้อมูลที่ต่างชนิดกัน มักจะมีข้อปลีกย่อยในการบังคับข้อมูลแตกต่างกัน แต่ถึงแม้ว่ากฎข้อบังคับความถูกต้องของข้อมูลบางชนิดในเมต้าสกีมาได้ระบุรายละเอียดในการบังคับข้อมูลไว้ค่อนข้างเหมือนกัน เช่น cardinality constraint, occurrence frequency constraint และ cardinality of role constraint ซึ่งต่างก็มีรายละเอียดในการบังคับกฎเป็นจำนวนครั้งเหมือนกันแต่เป็นจำนวนครั้งสำหรับเหตุการณ์ที่แตกต่างกัน เช่น สำหรับ cardinality constraint

กฎข้อบังคับความถูกต้องของข้อมูลแต่ละชนิดต่างก็ต้องใช้ข้อกำหนดในการบังคับข้อมูลแตกต่างกันออกไป เช่น range constraint และ membership constraint แม้จะเป็นกฎข้อบังคับความถูกต้องของข้อมูลชนิด entity type เหมือนๆ กัน แต่การระบุข้อกำหนดของกฎแตกต่างกัน โดย membership constraint นั้นจะเก็บค่าข้อมูลของสมาชิกที่เป็นไปได้ทุกค่าของชนิดเอนทิตีที่ถูกบังคับ เช่น {'G', 'S', 'B'} จะเก็บทั้งค่า 'G' ค่า 'S' และค่า 'B' ควบคู่ไปกับชื่อของกฎข้อบังคับความถูกต้องของข้อมูล ส่วน range constraint จะเก็บเพียงค่าข้อมูลที่ใช้ในการกำหนดช่วงของค่าที่เป็นไปได้ของสมาชิกชนิดเอนทิตีเท่านั้น เช่น [1...200] จะเก็บค่า 1 และ 200 เพราะสำหรับกฎข้อบังคับความถูกต้องของข้อมูลชนิด range constraint นั้นการจะเก็บค่าที่เป็นไปได้ทั้งหมดของสมาชิกชนิดเอนทิตีเป็นเรื่องที่ไม่อาจจะเป็นไปได้ โดยเฉพาะกับชนิดเอนทิตีที่มีชนิดของข้อมูลเป็นเลขจำนวนจริง เนื่องจากจำนวนสมาชิกของชนิดเอนทิตีจะเป็นอนันต์ ดังนั้นจึงได้ออกแบบตารางเมต้าสกีมาให้รองรับการเก็บข้อกำหนดหรือเงื่อนไขของกฎข้อบังคับความถูกต้องของข้อมูลแต่ละชนิดแยกกันเป็นชนิดละหนึ่งตารางฐานข้อมูล เพื่อความสะดวกในการใช้งาน และการจัดการ ในขั้นตอนของการวิเคราะห์ประโยคแสดงแบบจำลองข้อมูลระดับแนวคิดนั้น เมื่อพบว่าประโยคนั้นเป็นประโยคที่ใช้กำหนดกฎข้อบังคับความถูกต้องของข้อมูลชนิดใด ก็จะแยกข้อกำหนดของกฎข้อบังคับความถูกต้องของข้อมูลข้อนั้นๆ ตามตำแหน่งที่ปรากฏในไวยากรณ์ของภาษา เพื่อนำมาบรรจุลงในตารางเมต้าสกีมาที่เกี่ยวข้อง โดยมากกฎข้อบังคับความถูกต้องของข้อมูลแต่ละชนิดจะมีตารางเมต้าสกีมารองรับข้อกำหนดหรือเงื่อนไขของมันโดยเฉพาะ ชนิดละหนึ่งตาราง แต่เนื่องจากกฎข้อบังคับความถูกต้องของข้อมูลบางชนิดไม่จำเป็นต้องระบุเงื่อนไขในการบังคับข้อมูล เพียงทราบว่ากฎข้อบังคับความถูกต้องของข้อมูลข้อนั้นๆ บังคับกับข้อมูลใดบ้างก็สามารถกำหนดเป็นกฎขึ้นบังคับใช้ได้ เช่น กฎข้อบังคับความถูกต้องของข้อมูลชนิด uniqueness และ mandatory role ดังนั้นเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กฎข้อบังคับความถูกต้องของข้อมูลทั้งสองชนิดนี้จึงไม่มีตารางเมตาสกิมมาสำหรับเก็บข้อกำหนดในการบังคับข้อมูลไว้รองรับ แต่ในขณะเดียวกันกฎข้อบังคับความถูกต้องของข้อมูลชนิด mandatory entity กลับต้องใช้ตารางเก็บข้อกำหนดในการบังคับข้อมูลร่วมกับกฎข้อบังคับความถูกต้องของข้อมูลชนิด membership ที่เป็นเช่นนี้ เนื่องจากการกำหนดกฎข้อบังคับความถูกต้องของข้อมูลชนิด mandatory entity นั้น โดยมากจะควบคู่ไปกับการแจกแจงค่าข้อมูลสมาชิกของชนิดเอนติตี้ที่เป็นไปได้ทั้งหมด

ตารางความสัมพันธ์ที่ได้จากแบบจำลองเมตาสกิมมา และข้อมูลแม่ปึง

ตารางที่ 13

OBJECT

OBJECT_NAME	OBJECT_TYPE
VARCHAR(30)	VARCHAR(10)

ตารางที่ 14

IDENTIFIER

LABEL_NAME	ENTITY_NAME	DATA_TYPE	SIZE
VARCHAR(30)	VARCHAR(30)	VARCHAR(10)	NUMBER

ตารางที่ 15

ROLE

ROLE_NAME	OBJECT_NAME	RELATIONSHIP_NAME	NAME_EXTEND
VARCHAR(100)	VARCHAR(100)	VARCHAR(100)	VARCHAR(20)

TABLE_NAME	COMMENT	REFERENCE
VARCHAR(30)	VARCHAR(30)	VARCHAR(30)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษานั้น มีอนันต์แต่ไม่ใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 16

CONSTRAINT

CONST#	CONST_TYPE	OBJECT_NAME	RELATIONSHIP_NAME
VARCHAR(30)	VARCHAR(10)	VARCHAR(30)	VARCHAR(100)

ตารางที่ 17

IC_ROLE

CONST#	ROLE_NAME
VARCHAR(30)	VARCHAR(100)

ตารางที่ 18

MANDATORY_ENT

CONST#	MEMBER_INSTANCE
VARCHAR(30)	VARCHAR(100)

ตารางที่ 19

MEMBERSHIP

CONST#	MEMBER_INSTANCE
VARCHAR(30)	VARCHAR(100)

ตารางที่ 20

RANGE

CONST#	LOWER_LIMIT	UPPER_LIMIT
VARCHAR(30)	NUMBER	NUMBER

ตารางที่ 21

OCCURRENCE

CONST#	FREQUENCY_TYPE	FREQUENCY
VARCHAR(30)	VARCHAR(10)	NUMBER

ตารางที่ 22

CARDINALITY

CONST#	QUANTITY_TYPE	QUNTITY
VARCHAR(30)	VARCHAR(10)	NUMBER

ตารางที่ 23

CARDINAL_ROLE

CONST#	NUMBER_TYPE	ROLE_NUMBER
VARCHAR(30)	VARCHAR(10)	NUMBER

ตารางที่ 24

EQUALITY

CONST#_ROLE1	CONST#_ROLE2
VARCHAR(30)	VARCHAR(30)

ตารางที่ 25

SUBSET

SUPERSET_CONST#	SUBSET_CONST#
VARCHAR(30)	VARCHAR(30)

ตารางที่ 26

SUBTYPE

CONST#	SUBTYPE_DEFINITION
VARCHAR(30)	VARCHAR(100)

ตารางที่ 27

TRANS_STRING

CONST#	ORDER_NO	POSSIBLE_VALUE
VARCHAR(30)	NUMBER	VARCHAR(30)

ตารางที่ 28

TRANS_VALUE

CONST#	TRANSIT_DIRECTION	LIMIT_OF_TRANSITION	INVALID
VARCHAR(30)	CHAR(1)	NUMBER	CHAR(1)

ตารางที่ 29

DYNAMIC

CONST#	CHANGE_DIRECTION	LIMIT_OF_CHANGE	PERIOD
VARCHAR(30)	CHAR(1)	NUMBER	VARCHAR(10)

ตารางที่ 30

USER_DEFINED

CONST#	IF_CLAUSE	THEN_CLAUSE	ACTION_CLAUSE
VARCHAR(30)	VARCHAR(200)	VARCHAR(200)	VARCHAR(200)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 31

PK

TABLE_NAME	PRIMARYKEY_NAME
VARCHAR(100)	VARCHAR(100)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ภาษาธรรมชาติไวยากรณ์รูปแบบจำกัด

ในการวิเคราะห์และออกแบบข้อมูลนั้น การแสดงความสัมพันธ์หรือกฎข้อบังคับต่างๆ ของระบบงาน ต้องการความละเอียดถี่ถ้วนของความหมายข้อมูลที่ต้องการจะสื่อสารกันเป็นอย่างสูง และสื่อที่จะใช้ในการแสดงข้อมูลที่วิเคราะห์ออกมาได้ ควรจะเป็นสื่อที่ทั้งผู้สื่อและผู้รับทำความเข้าใจได้ง่ายและตรงไปตรงมา ถึงแม้ว่าวิธีการของโนแอมจะใช้แผนภาพเป็นสื่อที่สามารถแสดงความหมายของความสัมพันธ์และกฎข้อบังคับข้อมูลต่างๆ ได้อย่างดีและชัดเจนก็ตาม แต่เนื่องจากสิ่งที่แปลความหมายจากแผนภาพแบบจำลองข้อมูลโนแอม ต้องอาศัยการศึกษาความหมายต่างๆ ภายในแบบจำลองซึ่งต้องใช้เวลานานพอสมควร และเพราะเหตุนี้เองที่อาจเป็นสาเหตุให้ผู้ปฏิบัติงานหมดความพยายามในการทำความเข้าใจกับแบบจำลองข้อมูลได้ อันจะเป็นเหตุให้แบบจำลองข้อมูลขาดการตรวจสอบความถูกต้องอย่างละเอียดรอบคอบได้

ดังนั้นการพัฒนาสื่อที่ใช้ในการแสดงแบบจำลองข้อมูลจึงจำเป็นที่จะต้องพิจารณาถึงความจำเป็นในการทำความเข้าใจของทั้งผู้ออกแบบฐานข้อมูลและผู้ปฏิบัติงานด้วยในคราวเดียวกัน ภาษาธรรมชาติเป็นรูปแบบหนึ่งที่มนุษย์สามารถทำความเข้าใจได้สะดวกง่ายดายที่สุด เนื่องจากภาษาธรรมชาติเป็นภาษาพื้นฐานที่มนุษย์ใช้สื่อสารกันมาเป็นเวลาช้านาน จากเหตุผลดังกล่าวงานวิจัยนี้จึงเลือกพัฒนาแบบจำลองข้อมูลขึ้นในรูปแบบของภาษาธรรมชาติ ซึ่งการออกแบบภาษาธรรมชาติสำหรับแสดงเป็นแบบจำลองข้อมูลในงานวิจัยนี้นั้น ได้เน้นในข้อที่ต้องการให้รูปแบบของภาษาสามารถอ่านทำความเข้าใจได้ง่าย และสามารถแสดงความสัมพันธ์ของข้อมูลและกฎข้อบังคับความถูกต้องของข้อมูลได้อย่างกว้างขวางที่สุดเท่าที่จะทำได้ ซึ่งต้องอาศัยความพยายามในการศึกษารูปแบบของความสัมพันธ์ของข้อมูลและกฎข้อบังคับความถูกต้องของข้อมูลให้ได้มากที่สุด แล้วสรุปนำรายละเอียดที่ศึกษาได้ทั้งหมดมากำหนดโครงสร้างของภาษา และตรวจสอบความสมบูรณ์ของภาษาด้วยการนำฐานข้อมูลจำลอง ที่มีชนิดของความสัมพันธ์และกฎข้อบังคับความถูกต้องของข้อมูลที่หลากหลายมาลองแสดงด้วยภาษาที่ออกแบบไว้ หากพบว่าโครงสร้างของภาษาส่วนใดไม่

เหมาะสมไม่สามารถแสดงความหมายได้อย่างชัดเจน ก็จะทำการปรับปรุงเพื่อให้ได้ภาษาที่ สละสลวยสื่อความหมายได้อย่างชัดเจนและสมบูรณ์ที่สุด

เนื่องจากงานวิจัยมีจุดมุ่งหมายที่จะแก้ไขปัญหาในการออกแบบฐานข้อมูล และการดูแล ความถูกต้องของข้อมูลในฐานข้อมูล ซึ่งได้กำหนดใช้ภาษาธรรมชาติเป็นทางออกของปัญหาดังที่ กล่าวไว้แล้วข้างต้นนั้น ในเบื้องต้นงานวิจัยนี้จึงได้ยึดถือเอาชนิดของกฎข้อบังคับความต้องการของ ข้อมูลที่แบ่งโดย G.M. Nijssen และ T.A. Halpin เป็นหลัก เพราะว่าชนิดของกฎข้อบังคับความต้องการ ของข้อมูลที่ทั้งสองท่านนี้ได้แบ่งแยกเอาไว้แล้ว มีอยู่มากมายหลายชนิด โดยที่แต่ละชนิดนั้นก็ มีลักษณะที่แตกต่างกันสามารถอธิบายได้อย่างชัดเจน จึงเหมาะสมแก่การนำไปกำหนดรูปแบบของ ภาษาเพื่อให้ภาษาที่ออกแบบเป็นแบบบรรทัดที่สมบูรณ์ อันจะทำให้ผู้ดูแลฐานข้อมูลสามารถ กำหนดกฎข้อบังคับความต้องการของข้อมูลได้ง่าย อีกทั้งสามารถดูแลและตรวจสอบกฎข้อบังคับ ความถูกต้องของข้อมูลในภายหลังได้โดยสะดวกอีกด้วย

โครงสร้างของภาษาธรรมชาติที่ออกแบบไว้สำหรับระบบสร้างกฎข้อบังคับความต้องการของ ข้อมูลโดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนลในงานวิจัยชิ้นนี้นั้น สามารถใช้ กำหนดความสัมพันธ์ของข้อมูลทั้งที่เป็นชนิดความจริงและชนิดอ้างอิงได้ ไม่ว่าชนิดความจริงนั้นๆ จะมีขนาดเป็นเท่าใด หรือสามารถจะกำหนดเป็นชนิดความจริงแบบเนสต์ (nested fact type) ได้ ด้วยการกำหนดชื่อเรียก (alias) ให้แก่ชนิดความจริงใดๆ แล้วใช้ชื่อเรียกนั้นเสมือนกับเป็นชนิดเอน ดิตีในการกำหนดชนิดความจริงแบบเนสต์ เช่น สำหรับชนิดความจริง “Student enroll_in Subject in Semester of Year take Grade” เป็นชนิดความจริงชนิดเอนันารีที่มีขนาดเท่ากับ 5 ซึ่งสามารถ กำหนดชื่อเรียกให้แก่ชนิดความจริง “Student enroll_in Subject in Semester of Year” และ กำหนดชนิดความจริงแบบเนสต์ได้ดังต่อไปนี้

Enrollment : Student enroll_in Subject in Semester of Year (กำหนด alias)

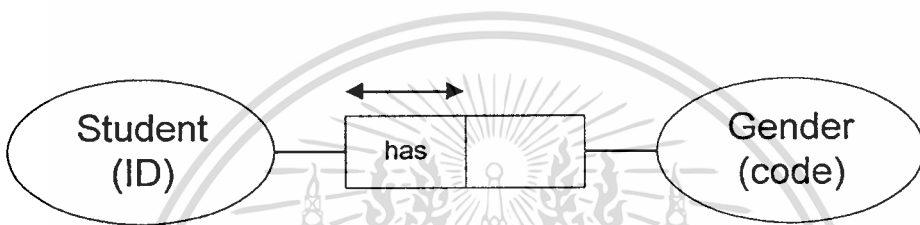
Enrollment take Grade (nested fact type)

ซึ่งประโยคความสัมพันธ์แบบเนสต์ ‘Enrollment take Grade’ จะมีความหมายเหมือนกับประโยค ความสัมพันธ์ ‘Student enroll_in Subject in Semester of Year take Grade’ ทุกประการ

จากการศึกษาและวิเคราะห์ข้อมูลจากระบบงานต่างๆ ได้พบว่าการกำหนดชนิดของกฎข้อ บังคับความต้องการของข้อมูลของ G.M. Nijssen และ T.A. Halpin สามารถครอบคลุมกฎข้อบังคับ ความถูกต้องของข้อมูลที่เควีเคราะห์มาได้ทั้งหมด แต่จากการศึกษาองค์ประกอบและโครงสร้าง ของกฎข้อบังคับความต้องการของข้อมูลแต่ละชนิด ได้ค้นพบว่ากฎบางกฎที่ G.M. Nijssen และ

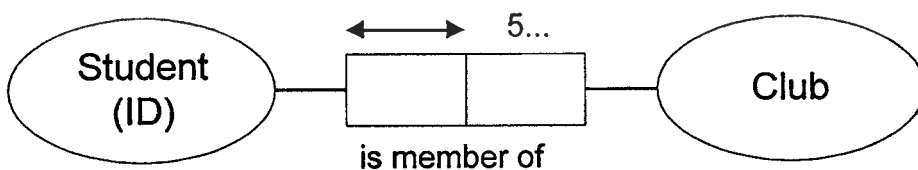
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

T.A. Halpin แบ่งประเภทไว้ว่า มีลักษณะและองค์ประกอบของกฎรวมถึงการบังคับกฎที่ไม่แตกต่างกัน จึงทำให้สามารถจัดกลุ่มของกฎข้อบังคับกับความถูกต้องของข้อมูลเหล่านั้นตามลักษณะดังกล่าวได้เป็นกลุ่มๆ ที่สามารถกำหนดรูปแบบของภาษาเพื่อใช้กำหนดกฎข้อบังคับความถูกต้องของข้อมูลไว้เหมือนกันได้ เช่น Uniqueness และ Occurrence Frequency Constraint เนื่องจากถ้า Uniqueness Constraint บังคับบนบทบาทใด ก็เป็นการบังคับว่าสมาชิกของชนิดเอนทิตีจะสามารถแสดงบทบาทนั้นได้เพียงหนึ่งครั้งเท่านั้น เช่น ภาพที่ 42 หมายความว่า นักศึกษาแต่ละคนจะมีเพศได้เพียงเพศเดียวเท่านั้น



ภาพที่ 42 แสดงตัวอย่าง uniqueness constraint

แต่ในกรณีของ occurrence frequency constraint นั้น ผู้ออกแบบฐานข้อมูลสามารถระบุได้ว่าหากสมาชิกใดๆ ของชนิดเอนทิตีปรากฏมีข้อมูลในบทบาทที่ถูกบังคับด้วย occurrence frequency constraint แล้ว จะต้องปรากฏสมานั้นๆ เป็นจำนวนครั้งที่ถูกบังคับ เช่น ในภาพที่ 43



ภาพที่ 43 แสดงตัวอย่าง occurrence frequency constraint

หมายความว่า นักศึกษาแต่ละคนเป็นสมาชิกของชมรมใดชมรมหนึ่งได้เพียงชมรมเดียวเท่านั้น แต่ว่าชมรมใดๆ ถ้าหากเปิดรับสมาชิก จะต้องสมาชิกตั้งแต่ 5 คนขึ้นไป ซึ่งจะเห็นได้ว่าทั้ง occurrence frequency constraint และ uniqueness constraint ต่างก็บังคับจำนวนครั้งของการปรากฏขึ้นของสมาชิกเอนติตีบนชนิดความจริงเช่นเดียวกัน เพียงแต่ uniqueness constraint เป็นกรณีพิเศษของ occurrence frequency constraint ที่บังคับจำนวนครั้งไว้แน่นอนคือ หนึ่งครั้งเท่านั้น

และกฎข้อบังคับความถูกต้องของข้อมูลชนิด exclusion และ mandatory disjunction ในแผนภาพในแอมต่างก็เป็นกฎข้อบังคับความถูกต้องของข้อมูลชนิดที่ใช้บังคับโอกาสในการแสดงบทบาทใดๆ ที่ถูกบังคับร่วมกับโอกาสในการแสดงบทบาทอื่นๆ ของชนิดเอนติตีเดียวกันนั้น โดย exclusion constraint บังคับให้สมาชิกของชนิดเอนติตีแสดงบทบาทต่างๆ ที่ถูกบังคับได้เพียงอย่างมากที่สุดหนึ่งบทบาท แต่ mandatory disjunction constraint (หรือ inclusion mandatory role constraint) บังคับให้สมาชิกของชนิดเอนติตีแสดงบทบาทต่างๆ ที่ถูกบังคับอย่างน้อยที่สุดหนึ่งบทบาท ดังนั้นจึงได้กำหนดโครงสร้างของภาษาให้รองรับการกำหนดกฎทั้งสองข้อนี้เป็นรูปแบบเดียวกัน และเนื่องจากการใช้ภาษาเขียนเป็นสื่อในการแสดงความหมายของกฎข้อบังคับความถูกต้องของข้อมูลนั้น ทำให้กฎที่สามารถกำหนดได้ด้วยภาษาธรรมชาติไวยากรณ์รูปแบบจำกัดจึงครอบคลุมการบังคับกฎต่างๆ ในระบบได้มากกว่าที่กำหนดได้บนแผนภาพในแอมอยู่มาก เช่น

every Lecturer play one roles of { Lecturer has Address , Lecturer lives_at Room }

Student play at least one roles of { Student is_member_of Club , Student is_member_of Team }

ในขณะที่การใช้ภาษาในการกำหนดชนิดอ้างอิงนั้น จำเป็นต้องกำหนดรูปแบบของภาษาให้แตกต่างกัน สำหรับกรณีการกำหนดชนิดเอนติตีให้เป็นเซตของชื่อที่ใช้อ้างอิงหรือแสดงเอกลักษณ์ของสมาชิกชนิดเอนติตี (reference mode) และกรณีที่ชนิดเอนติตีไม่สามารถใช้แสดงเอกลักษณ์ของสมาชิกชนิดเอนติตีได้หรือไม่ได้เลือกใช้ให้เป็นตัวบ่งเอกลักษณ์ ทั้งนี้เนื่องจากการกำหนดชนิดอ้างอิงในแผนภาพแบบจำลองข้อมูลในแอมนั้น ผู้ใช้สามารถกำหนดลงในแผนภาพได้เหมือนกัน เพียงแต่การกำหนดกฎข้อบังคับความถูกต้องของข้อมูลชนิด uniqueness ที่แสดงความสัมพันธ์ว่าเป็นไปแบบ one to one จะทำให้ทราบได้ว่าชนิดเอนติตีนั้นสามารถใช้เป็นตัวบ่งเอกลักษณ์ได้ ซึ่งในขั้นตอนของการสร้างรีเลชันแนลสกีมาจึงจะเลือกที่จะตัดสินใจใช้ตัวเลือกใดจากตัวบ่งเอกลักษณ์ที่เป็นไปได้ทั้งหมดสำหรับชนิดเอนติตีหนึ่งๆ แต่ในระบบอัตโนมัติสำหรับดูแลความถูกต้องของข้อมูลบนระบบฐานข้อมูลแบบรีเลชันแนลนี้ จำเป็นต้องทราบข้อมูลที่จำเป็นในเอกสารนี้เป็นเอกสารที่สแกนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างรีเลชันแนลสกีมาทั้งหมดจากแบบจำลองข้อมูลระดับแนวคิด เพื่อที่จะใช้ข้อมูลดังกล่าวสร้างรีเลชันแนลสกีมาและกฎต่างๆ ได้อย่างถูกต้องและครบถ้วน จึงจำเป็นต้องอยู่ที่ผู้ใช้จะต้องตัดสินใจหรือกำหนดชนิดอ้างอิงที่อ้างอิงถึงเอกลักษณ์ของชนิดเอนิตีใดๆ แยกจากการระบุชนิดอ้างอิงปกติ

อีกทั้งการระบุกฎของ G.M. Nijssen และ T.A.Halpin เป็นไปในลักษณะครอบคลุมทั้งชนิดเอนิตีหรือชนิดความจริง โดยไม่สามารถระบุข้อมูลให้ละเอียดถึงกลุ่มของสมาชิกชนิดความจริงได้ เช่น กฎข้อบังคับที่ว่าสมาชิกของทุกชมรมจะต้องมากกว่าหรือเท่ากับ 5 คน แต่สำหรับชมรมฟุตบอลแล้วจะต้องมีสมาชิกของชมรมตั้งแต่ 11 คนขึ้นไป เนื่องจากการระบุกฎข้อบังคับความถูกต้องของข้อมูลเป็นข้อความยาวๆ และจำนวนมากๆ ลงในแบบจำลองข้อมูลที่เป็นแผนภาพเป็นเรื่องที่ไม่สะดวก จึงเป็นข้อได้เปรียบอีกประการหนึ่งของการใช้ภาษาธรรมชาติเป็นแบบจำลองข้อมูล เนื่องจากผู้ใช้สามารถบรรยายกฎเกณฑ์ต่างๆ ได้ละเอียดตามที่ต้องการ

BNF (Backus-Naur Form) แสดงไวยากรณ์และโครงสร้างของประโยคคำตั้งภาษาธรรมชาติที่กำหนดขึ้น

relationship --> fact_type | reference_type

fact_type --> object role object; |

object1 role object2 & object2 role object1;

object --> entity | alias | fact_type

role --> name

entity --> name

alias --> name

ตัวอย่างของ fact type

Student enroll_in Subject;

Enrollment take Grade & Grade of Enrollment;

Student enroll_in Subject in Semester of Year;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

reference_type --> entity role label; /

entity role label & label role entity;

ตัวอย่างของ reference type

Student has NAME;

Student has ARMY_ID & ARMY_ID is_military_id_of Student;

alias_statement --> alias : fact_type

ตัวอย่างของการกำหนด alias สำหรับชนิดความจริง

Term : Semester of Year;

Enrollment : Student enroll_in Subject in Term;

lexical --> label datatype(size);

datatype --> char | varchar | number | date

size --> number

ตัวอย่างการกำหนด lexical

ST_ID char(7);

LECT_NAME varchar(30);

NAME varchar(20);

unique_identifier --> entity is uniquely identified by label; /

entity is uniquely identified by (label1, label2 ..., labeln); /

entity(label); /

entity(label1, label2,..., labeln);

ตัวอย่างการกำหนด unique identifier

Student is uniquely identified by ST_ID;

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของสำนักงานคณะกรรมการการอุดมศึกษา ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Lecturer is uniquely identified by (LECT_NAME, LECT_SURNAME);

Lecturer(LECT_NAME, LECT_SURNAME);

subtype_ic --> entity1 = entity2 role entity3 str ; /

entity1 = entity2 role entity3 in { str1, str2,..., strn } ; /

entity1 = entity2 role label str ; /

entity1 = entity2 role label in (str1, str2,..., strn) ;

entity1 = entity2 role entity3 relation n ; /

entity1 = entity2 role entity3 in range of [lower_limit, upperlimit];/

entity1 = entity2 role entity3 in (n1, n2,..., nm) ;

ตัวอย่างการกำหนด subtype constraint

MaleStudent = Student has Gender 'M' ;

NewProduct = Product has CODE in ('qm009' , 'sr688' , 'uj777') ;

HiEmployee = Employee earns Salary > 30000 ;

MidEmployee = Employee earn Salary in range of [10000, 29999] ;

uniqueness_constraint --> ic_name := one entity role quantity label; /

ic_name := one entity role quantity label & one label role quantity entity; /

ic_name := one real_object role quantity single_object; /

ic_name := one real_object role quantity single_object & one single_object role

quantity real_object; /

single_object --> entity / alias

real_object --> entity / alias / fact_type

quantity --> one | many

ตัวอย่างการกำหนด uniqueness constraint

uk_stname := one Student has one NAME;

uk_stgender := one Student has one Gender & one Gender of many Student;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

uk_enroll_gd := one Student enroll_in Subject in Semester of Year take one Grade;

uk_enroll := one Student enroll_in Subject in many Term & one Term of many student;

uk_en_grade := one Enrollment take one Grade & one Grade of many Enrollment;

occurrence_frequency_constraint --> ic_name := one entity role n label; /

ic_name := one entity role n label & one label role n entity; /

ic_name := one real_object role n single_object; /

ic_name := one real_object role n single_object & one single_object role real_object ;

ตัวอย่างการกำหนด occurrence frequency constraint

mem_club := one Club has_member 10 to 100 Student ;

wife_no := one Man has_wife 0 to 4 Women ;

prod_dealer := one Product is_distributed_by 10 Dealer & one Dealer distributes 2 to 5 Product ;

inter-uniqueness_ic --> ic_name := combination of (label1, label2,..., labeln) of entity is unique;

ตัวอย่างการกำหนด inter-uniqueness constraint

uk_st_ns := combination of (NAME,SURNAME) of Student is unique;

mandatory-role_ic --> ic_name := every entity role label /

ic_name := every object role entity

ตัวอย่างการกำหนด mandatory role constraint

md_stname := every Student has NAME

md_stgender := every Student has Gender

md_enroll_gd := every Enrollment take Grade

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

md_winner := every Team win Game with Medal (เวลาชนะจะต้องได้เหรียญ)

membership_ic --> ic_name := label must be in { str1,str2,...,strn };

str --> 'name'

ตัวอย่างการกำหนด membership constraint

mem_medek := MEDAL_CODE must be in { 'G', 'S', 'B' };

*mam_deptname := DEPT_NAME must be in { 'Computer', 'Telecommunication',
'Electronics' };*

range_ic --> ic_name := object role entity in range of [lower_limit , upper_limit] ; /

ic_name := object role entity comparative limit ;

comparative --> greater than | less than | not less than | not greater than

limit --> number

lower_limit --> number

upper_limit --> number

ตัวอย่างการกำหนด range constraint

range_iq := Person has IQ in range of [0 , 200];

range_votes := Candidate win Election with Votes not greater than 200;

mandatory-entity_ic --> ic_name := set of entity1 must be has at least n entity1 role entity2;

str /

n --> number

ตัวอย่างการกำหนด mandatory entity constraint

ent_node := set of Node must be has at least one Node with Node_Kind 'FS';

end_lecttype := set of Lecturer must be has at least 3 Lecturer has Lect_Type 'ประจำ';

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

cardinality_ic --> ic_name := number of entity must be equal to n ; /
ic_name := number of entity must be comparative limit ; /
ic_name := number of entity must be in range of [lower_limit ,
upper_limit] ;

ตัวอย่างการกำหนด cardinality constraint

num_club := number of Club must not be greater than 10 ;
num_committee := number of Committee must not be less than 7 ;
num_candidate := number of Candidate must be in range of [5 , 9] ;

cardinality-of-role_ic --> ic_name := entity play role_qty n roles of { relationship1,
relationship2,..., relationshipn } ; /
ic_name := every entity play role_qty n roles of { relationship1,
relationship2,..., relationshipn } ; /
role_qty --> at most | at least | exactly

ตัวอย่างการกำหนด cardinality of role constraint

role_lect := every Lecturer play one roles of { Lecturer has Address , Lecturer
lives_at Room } ;
role_st := Student play at least one roles of { Student is_member_of Club , Student
is_member_of Team } ;

equality_ic --> ic_name := relation1 iff relation2 ;

ตัวอย่างการกำหนด equality constraint

eq_roomtel := Lecturer lives_at Room iff Lecturer has Internal_Tel ;

subset_ic --> ic_name := relation1 if relation2 ;

ตัวอย่างการกำหนด subset constraint

subset := MaleStudent trained_at Camp if MaleStudent has ARMY_ID ;

transition-of-str_ic --> ic_name := sequence order for update of entity role object is (str1, str2,..., strn) ; /

ic_name := sequence order for update of entity role object is < str1, str2,..., strn > ;

ตัวอย่างการกำหนด transition of string values constraint

tran_pos := sequence order for update of Position of Employee is ('salesman', 'supervisor', 'sale manager') ;

tran_season := sequence order for update of Season of Country is < 'Winter', 'Autumn', 'Summer' > ;

transition-of-value_ic --> ic_name := entity role object is change_direction ; |

ic_name := entity role object is change_direction comparative n ; /

ic_name := entity role object is not change_direction comparative n ; /

ic_name := entity role object is change_direction comparative n

percent ; /

ic_name := entity role object is not change_direction comparative n

percent ; /

change_direction --> decreasingly | increasingly

ตัวอย่างการกำหนด transition of numeric values constraint

chg_price := Price of Product is increasingly ;

tran_hrate := Heart_Rate of Person is not decreasingly more than 5 percent ;

chg_sal := Salary of Employee is not increasingly less than 500 ;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

dynamic_ic --> ic_name := entity role object is change_direction per period ; |

ic_name := entity role object is change_direction comparative n per period ; |

ic_name := entity role object is not change_direction comparative n per period

;/

ic_name := entity role object is change_direction comparative n percent per

period ; |

ic_name := entity role object is not change_direction comparative n percent

per period ; |

ตัวอย่างการกำหนด dynamic constraint

period --> minute / hour / day / week / month / year

dyn_sal := Salary of Employee is not increasingly more than 30 percent per year ;

dyn_hrate := Heart_Rate of Person is not decreasingly more than 5 percent per

hour ;

dyn_tprice := Price of Ticket is increasingly with limit of 3 per month ;

asymmetric --> ic_name := fact_type is asymmetric ;

ตัวอย่างการกำหนด asymmetric constraint

asym_parent := Person is_parent_of Person is asymmetric ;

irreflexive --> ic_name := fact_type is irreflexive ;

ตัวอย่างการกำหนด irreflexive constraint

ir_parent := Person is_parent_of Person is irreflexive ;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

intransitive --> ic_name := fact_type is intransitive ;

ตัวอย่างการกำหนด intransitive constraint

intr := Person is_parent_of Person is intransitive ;

acyclic --> ic_name := fact_type is acyclic ;

ตัวอย่างการกำหนด acyclic constraint

acy_boss := Manager is_boss_of Employee is acyclic ;

user-defined_ic --> ic_name := if statement then statement exception action ;

statement --> condition1 conjunction condition2 conjunction... conjunction conditionm

condition --> entity role object relation n |

entity role object in (n1, n2,..., nm) |

entity role object not in (n1, n2,..., nm) |

entity role object in range of [lower_limit, upper_limit] |

entity role object not in range of [lower_limit, upper_limit] |

entity role object relation str |

entity role object in (str1, str2,..., strn) |

entity role object not in (str1, str2,..., strn) |

entity role object like str_pattern |

entity role object not like str_pattern |

relation --> = | > | >= | < | <= | <>

str_pattern --> alphabet < combine with > wildcard_symbol

alphabet --> a-z | A-Z | _ | # | \$

wildcard_symbol --> _ | %

conjunction --> and | or

action --> print(str)

ตัวอย่างการกำหนด user-defined constraint

```
chk_sal := if Salary of Employee > 3000 then Position of Employee not in ('manager'
, 'director') exception    print('Employee who is not manager or derector cannot earns salary
more than 30000');
```

```
chk_air := if Position of Employee = 'Airhostes' and not (Sex of Employee = 'F' and
Age of Employee < 30) then print('Airhostes must be Female and younger than 30);
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สถาปัตยกรรมของระบบสร้างกฎบังคับความถูกต้องของข้อมูลโดยใช้ภาษา ธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนล

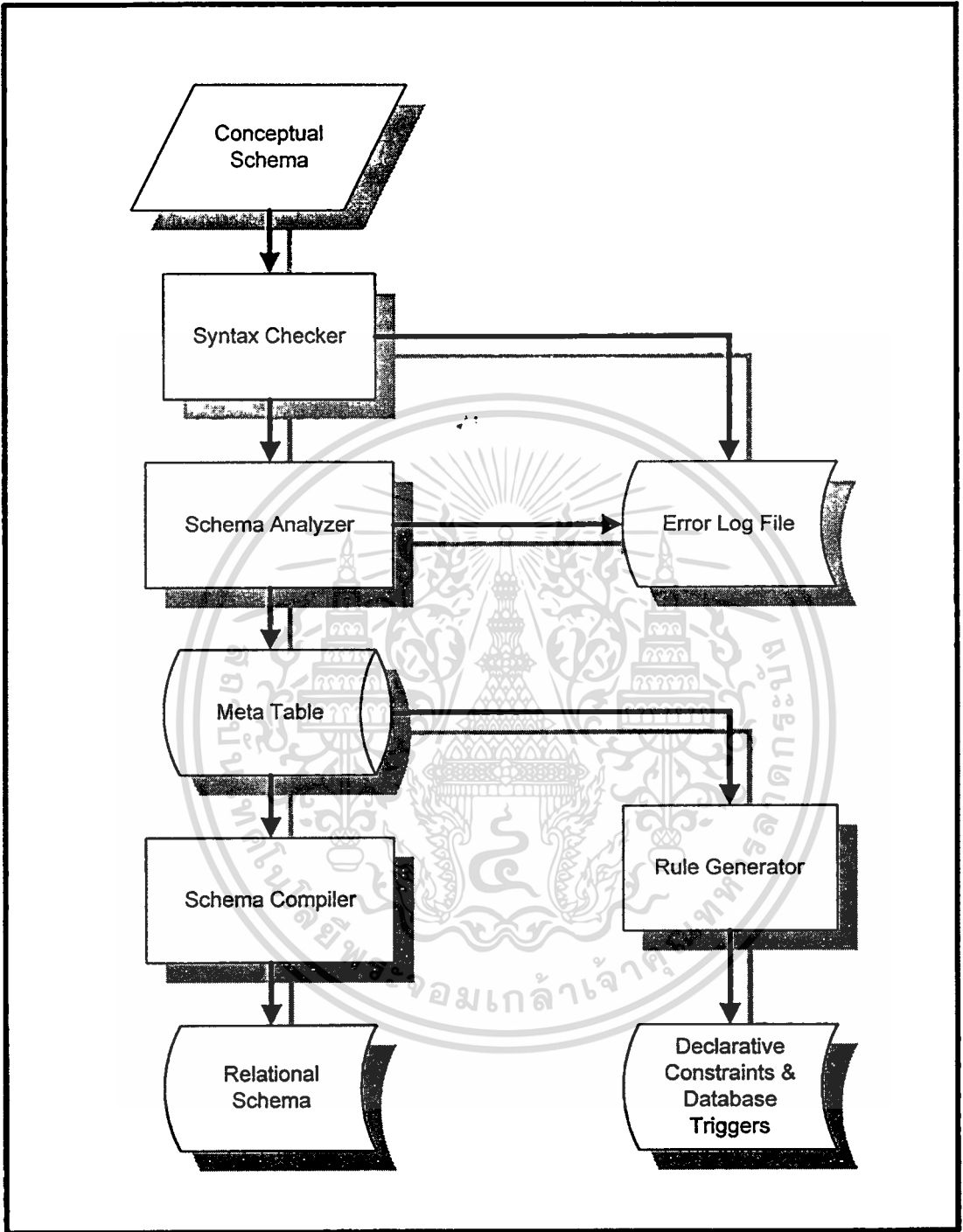
สถาปัตยกรรมของระบบสร้างกฎบังคับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนลประกอบไปด้วย 3 ส่วนหลักๆ คือ Syntax Checker, Schema Analyzer, Schema Compiler และ Rule Generator โดยหน้าที่ของแต่ละส่วนมีดังนี้คือ

1. Syntax Checker จะทำหน้าที่ตรวจสอบไวยากรณ์ของประโยคภาษาธรรมชาติที่ใช้แสดงแบบจำลองข้อมูลระดับแนวคิดซึ่งจะอยู่ในรูปของเท็กซ์ไฟล์ว่าถูกต้องตามไวยากรณ์ที่ได้กำหนดเอาไว้หรือไม่ ถ้าหากพบว่าประโยคใดมีปัญหาเขียนไม่ถูกต้องตามไวยากรณ์ของประโยคภาษาธรรมชาติไวยากรณ์จำกัด ก็จะแสดงประโยคนั้นออกมาพร้อมกับข่าวสารแสดงความผิดพลาดของประโยคนั้นๆ และบันทึกข่าวสารดังกล่าวลงในไฟล์เพื่อให้ผู้ใช้สามารถตรวจสอบความผิดพลาดของประโยคคำสั่งได้ภายหลังอีกด้วย

2. Schema Analyzer เป็นส่วนที่มีหน้าที่สำคัญในการแยกเอาข้อมูลจากประโยคภาษาธรรมชาติที่ใช้แสดงแบบจำลองข้อมูลระดับแนวคิดว่าประกอบไปด้วย ชนิดเอนติตี้ และชนิดเลเบิ้ลใดบ้าง และชนิดเอนติตี้หรือชนิดเลเบิ้ลเหล่านั้นมีความสัมพันธ์กันอย่างไร และมีกฎข้อบังคับความถูกต้องของข้อมูลใดบ้างที่บังคับอยู่บนความสัมพันธ์และชนิดเอนติตี้เหล่านั้น แล้วนำข้อมูลที่วิเคราะห์ได้มาบรรจุลงในตารางเมตาสกีมมา

นอกจากนี้ Schema Analyzer ยังมีหน้าที่เสริมในการกำจัดความสัมพันธ์ที่กำหนดมาอย่างซ้ำซ้อนกันหลายครั้งอีกด้วย จึงทำให้ข้อมูลของแบบจำลองข้อมูลระดับแนวคิดที่ปรากฏในตารางเมตาสกีมมาจะไม่มีข้อมูลที่ซ้ำกัน (หรือไม่ต้องก็ได้ เนื่องจาก constraint บน RDBMS ช่วยกำจัดให้อยู่แล้ว)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 44 สถาปัตยกรรมของระบบสร้างกฎบังคับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติ
สำหรับฐานข้อมูลแบบรีเลชันแนล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Schema Compiler

ทำหน้าที่สร้างรีเลชันแนลสกีมาในรูปของประโยคคำสั่ง SQL จากข้อมูลของแบบจำลองข้อมูลระดับแนวคิดในตารางเมตาสกีมา

4. Rule Generator

ทำหน้าที่สร้างคำสั่งกำหนดกฎข้อบังคับความต้องการของข้อมูลชนิดอรรถาธิบาย , database trigger และ stored procedure ตามข้อมูลของกฎที่ปรากฏอยู่ในตารางเมตาสกีมา



บทที่ 7

การพัฒนาระบบ

อุปกรณ์ที่ใช้ในการพัฒนาระบบ

1. เครื่องคอมพิวเตอร์ ไมโครคอมพิวเตอร์ IBM PC คอมแพคทีเบิ้ล โดยมีหน่วยประมวลผลกลาง (CPU : Central Processing Unit) เป็นชิป 80486 DX-4 ขนาด 16 บิต หน่วยความจำ 32 เมกะไบต์ ฮาร์ดดิสก์ขนาด 1 จิกะไบต์ ฟลอปปีดิสก์ไดรฟ์ขนาด 1.44 นิ้วหนึ่งตัว ซีดีรอมไดรฟ์ 8X หนึ่งตัว จอภาพแบบ super VGA (1024 X 768) 256 สี แป้นพิมพ์ 101 คีย์

2. ซอฟต์แวร์

2.1 โปรแกรมระบบปฏิบัติการ MS-DOS 6.2

2.2 โปรแกรม Microsoft Windows 3.1

2.3 โปรแกรมระบบจัดการฐานข้อมูลรีเลชันแนล Personal Oracle7, SQL*Plus

2.4 โปรแกรม Delphi 1.0

การพัฒนาระบบ

การพัฒนาระบบสร้างกฎบังคับกับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับฐาน

ข้อมูลแบบรีเลชันแนลนั้นได้แบ่งงานออกเป็น 3 ส่วนตามโครงสร้างหลักของสถาปัตยกรรมระบบ

เบื้องต้นก็สืบเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Syntax Checker

ใช้การเขียนโปรแกรม Delphi ให้รับประโยคภาษาธรรมชาติเข้ามาทีละทีละประโยคคำสั่ง ซึ่งแต่ละประโยคคำสั่งเหล่านั้นจะต้องจบด้วยเครื่องหมาย ‘;’ (semicolon) และทำการตรวจสอบชุดประโยคคำสั่งเหล่านั้นว่าถูกต้องตามไวยากรณ์ที่กำหนดไว้หรือไม่ โดยซึ่งถ้าหากพบว่าประโยคใดเขียนไม่ถูกไวยากรณ์ ก็จะแสดงความผิดพลาดของการกำหนดประโยคคำสั่งนั้นออกหน้าจอพร้อมทั้งบันทึกลงในไฟล์บันทึกข้อผิดพลาดตามที่ผู้ใช้กำหนด

2. Schema Analyzer

เป็นส่วนที่ใช้ในการวิเคราะห์ประโยคชนิดความจริงและประโยคชนิดอ้างอิงที่ผู้ใช้กำหนดมา ว่าในแต่ละประโยคประกอบไปด้วยชนิดเอนติตี้หรือชนิดเลเบลใดๆ บ้าง และชนิดเอนติตี้ และชนิดเลเบลเหล่านั้นมีความสัมพันธ์กันอย่างไร และกฎข้อบังคับความถูกต้องของข้อมูลที่กำหนดไว้บังคับกับชนิดเอนติตี้, ชนิดเลเบล, บทบาท หรือความสัมพันธ์ไหนบ้างและบังคับอย่างไร แล้วนำข้อมูลที่วิเคราะห์ได้มา INSERT ลงสู่ตารางเมตาดาทา

จากโครงสร้างของภาษาธรรมชาติไวยากรณ์รูปแบบจำกัดที่ออกแบบขึ้นนั้น ได้กำหนดให้ประโยคความสัมพันธ์ประกอบขึ้นจาก การใช้ชื่อชนิดเอนติตี้และชื่อชนิดเลเบลชนิดต่างๆ มาเขียนแสดงความสัมพันธ์ โดยมีกริยากันสำหรับทุกๆ คู่ของชื่อชนิดเอนติตี้ หรือคู่ของชื่อชนิดเอนติตี้และชื่อชนิดเลเบล ซึ่งชื่อชนิดเอนติตี้, ชื่อชนิดเลเบล และกริยา ต่างจะต้องถูกกำหนดเป็นคำเพียงคำเดียวเท่านั้น คำว่า ‘คำ’ ในที่นี้หมายถึง ชุดของตัวอักษรที่เรียงติดต่อกันไปโดยแต่ละคำจะสามารถแยกออกจากกันได้ด้วย space ดังนั้นกริยาของประโยคความสัมพันธ์ที่ในทางปฏิบัติในภาษาเขียนปกติอาจจะประกอบไปด้วยคำหลายๆ คำได้นั้น เช่น Lecturer is advisor of Student เมื่อนำมากำหนดใช้เป็นกริยาของประโยคความสัมพันธ์ภายใต้ไวยากรณ์ของภาษาธรรมชาติไวยากรณ์รูปแบบจำกัดนี้ ผู้ใช้จะต้องนำคำที่ประกอบกันเป็นกริยานั้น มาเชื่อมต่อกันด้วยอักขระอื่นๆ ที่ภาษาธรรมชาติไวยากรณ์รูปแบบจำกัดยินยอมให้ใช้เช่น Lecturer is advisor_of Student

การวิเคราะห์ประโยคความสัมพันธ์ที่ป้อนเข้าสู่ระบบนั้น จะแยกแต่ละประโยคออกจากกันด้วยเครื่องหมาย ‘;’ สำหรับประโยคที่ไม่พบเครื่องหมาย ‘:=’ ซึ่งเป็นเครื่องหมายที่คั่นระหว่างชื่อเอกสารเป็นเอกสารที่ส่งงานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของกฎข้อบังคับความถูกต้องของข้อมูล และข้อกำหนดของกฎข้อบังคับความถูกต้องของข้อมูลนั้นๆ จะถือว่าเป็นประโยคที่ใช้กำหนดความสัมพันธ์ชนิดความจริง, ชนิดอ้างอิง, ประโยคที่ใช้กำหนด unique identifier , subtype หรือ ชนิดและขนาดของข้อมูลเลเบิลเท่านั้น การวิเคราะห์องค์ประกอบของประโยคเพื่อที่จะนำข้อมูลทีวิเคราะห์ได้ไปบรรจุลงในตารางเมตาด้าสก็มมาเป็นดังนี้คือ

2.1 การวิเคราะห์ประโยคความสัมพันธ์ ทุกประโยคชนิดความจริงจะถูกพิจารณาเป็นประโยคที่ประกอบไปด้วยประธาน, กรรมและกริยา และหนึ่งประโยคจะมีประธาน, กรรม และกริยาเพียงชนิดละหนึ่งเท่านั้น

2.1.1 สำหรับทุกประโยคความจริงขั้นพื้นฐาน ที่ไม่ได้กำหนด inverse role มาให้ (ไม่ปรากฏเครื่องหมาย '&')

2.1.1.1 จะทำการตัดคำ 3 คำแรกจากประโยค เพื่อใช้อ้างอิงเป็น ประธาน, กริยา และกรรม ของชนิดความจริงไบนารี

2.1.1.2 บรรจุข้อมูลลงในตารางเมตาด้าสก็มมา ROLE โดยใช้ชื่อชนิดเอนติตี้ที่เป็นประธานบรรจุลงในคอลัมน์ OBJECT_NAME และประโยคที่ประกอบกันขึ้นจาก ประธาน, กริยา และกรรม ของประโยคบรรจุลงในคอลัมน์ ROLE_NAME และ RELATIONSHIP_NAME

2.1.1.3 สร้าง inverse role โดยบรรจุกรรมของประโยคลงในคอลัมน์ OBJECT_NAME และสร้างประโยคจากการสลับตำแหน่งของประธาน และกรรม และใช้คำว่า by ต่อท้ายกริยา (ยกเว้นกรณีที่กริยาเป็นคำว่า has, have, in, of, for, with ให้ใช้คำว่า of เป็นคำกริยาแทน และกรณีที่คำกริยาประกอบด้วยคำว่า of หรือ for เป็นคำท้าย ให้ตัดคำ of หรือ for ออกก่อนที่จะเติมคำว่า by ลงไป) แล้วนำประโยคที่ได้บรรจุลงในคอลัมน์ ROLE_NAME ส่วนข้อมูลที่บรรจุลงในคอลัมน์ RELATIONSHIP_NAME นั้นใช้ข้อมูลเดียวกับข้อ 2.1.1.2

2.1.1.4 ถ้าหากไม่พบคำต่อไป ให้ไปเริ่มต้นวิเคราะห์ประโยคใหม่

2.1.1.5 ถ้าหากยังพบคำต่อไป ให้ตัดคำจากประโยคมาอีก 2 คำ เพื่อใช้เป็นกริยา และกรรมของประโยคชนิดความจริงไบนารีต่อไป และใช้ข้อมูลที่บรรจุลงในคอลัมน์ RELATIONSHIP_NAME ตามข้อ 2.1.1.2 เป็นประธานของประโยค

2.1.1.6 บรรจุข้อมูลลงในตารางเมตาด้าสก็มมา ROLE โดยใช้ประโยคจากคอลัมน์ RELATIONSHIP_NAME ที่ใช้เป็นประธานของประโยคใหม่ บรรจุลงในคอลัมน์ OBJECT_NAME และบรรจุประโยคใหม่ทั้งประโยคลงในคอลัมน์ ROLE_NAME และ RELATIONSHIP_NAME

2.1.1.7 ย้อนขึ้นไปทำตามขั้นตอนที่ 2.1.1.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 สำหรับประโยคที่พบสัญลักษณ์ ‘&’ ถือว่าเป็นประโยคความสัมพันธ์ที่ผู้กำหนด inverse role มาให้ แต่การกำหนด inverse role นี้ จำกัดให้กำหนดได้เฉพาะสำหรับประโยคชนิดความจริงหรือชนิดเลเบลที่มีขนาดเป็นไบนารีเท่านั้น

2.1.2.1 แยกคำ 3 คำออกเป็น ประธาน, กริยา และกรรมของประโยค

2.1.2.2 บรรจุข้อมูลลงในตารางเมตาสกิมมา ROLE โดยใช้ชื่อของชนิดเอนติตี้ที่เป็นประธานของประโยคบรรจุลงในคอลัมน์ OBJECT_NAME และประโยคที่ประกอบกันขึ้นจากประธาน, กริยา และกรรม บรรจุลงในคอลัมน์ ROLE_NAME และ RELATIONSHIP_NAME

2.1.2.3 แยกคำอีก 3 คำจาก inverse role มาใช้เป็นประธาน, กริยา และกรรมของประโยค inverse role

2.1.2.4 บรรจุข้อมูลลงในตารางเมตาสกิมมา ROLE โดยใช้ชื่อชนิดเอนติตี้ที่เป็นประธานบรรจุลงในคอลัมน์ OBJECT และประโยคที่ประกอบขึ้นจากประธาน , กริยา และกรรมของ inverse role นั้นบรรจุลงในคอลัมน์ ROLE_NAME ส่วนค่าข้อมูลที่บรรจุลงในคอลัมน์ RELATIONSHIP_NAME นั้นใช้ข้อมูลเดียวกับที่บรรจุลงในคอลัมน์ RELATIONSHIP_NAME ตามข้อ 2.1.2.2

2.2 การวิเคราะห์ประโยคกำหนด unique identifier สำหรับชนิดเอนติตี้ทุกชนิด ก่อนที่จะกำหนดใช้ในประโยคความจริงและประโยคอ้างอิง จำเป็นจะต้องกำหนด unique identifier หรือตัวบ่งเอกลักษณ์ของชนิดเอนติตตินั้นๆ เสียก่อน โดยผู้ใช้งานจะต้องกำหนดชื่อของชนิดเอนติตี้ไม่ซ้ำกัน และต้องกำหนดชื่อของชนิดเลเบลไม่ซ้ำกับชื่อที่กำหนดไว้แล้วด้วย โดยประโยคคำสั่งที่ใช้ในการกำหนด unique identifier มี 2 รูปแบบคือ

entity is uniquely identified by (label1 datatype[size], label3 datatype[size],...)

หรือ

entity(label1 datatype[size], label3 datatype[size],...)

โดยชนิดเลเบลที่ใช้กำหนดเป็น unique identifier อาจจะมีเพียงหนึ่งชนิด หรือหลายชนิดก็ได้ ซึ่งข้อมูลจากประโยคชนิดนี้จะนำมาบรรจุลงในตารางเมตาสกิมมา OBJECT และ IDENTIFIER

2.3 การวิเคราะห์ประโยคที่ใช้กำหนด alias สำหรับชนิดความจริง alias นั้นเป็น

ชื่อเรียกที่ผู้ใช้งานสามารถกำหนดไว้สำหรับเรียกแทนความสัมพันธ์ใดๆ โดยความสัมพันธ์นั้นจะต้องเอกสารนี้เป็นเอกสารที่สแกนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นชนิดความจริงชนิด many to many เท่านั้น ซึ่งจะมีขนาดของความสัมพันธ์เป็นเท่าใดก็ได้ และผู้ใช้สามารถนำ alias ที่กำหนดขึ้นนี้ไปใช้ในการกำหนดชนิดความจริงอื่นๆ และใช้ในการกำหนดกฎข้อบังคับความถูกต้องของข้อมูลได้เช่นเดียวกับการใช้ชื่อของชนิดเอนติตี้ต่างๆ ไป แต่มีข้อกำหนดว่าจะต้องกำหนด alias เสียก่อนจึงจะสามารถนำไปใช้กำหนดชนิดความจริงและกฎข้อบังคับความถูกต้องของข้อมูลต่างๆ ได้ ข้อมูลที่ได้จากการวิเคราะห์ประโยคกำหนด alias จะนำไปบรรจุไว้ในตารางเมตาสกีมมา ALIAS

2.4 การวิเคราะห์ประโยคที่ใช้กำหนดกฎข้อบังคับความถูกต้องของข้อมูลชนิด

uniqueness การกำหนดกฎข้อบังคับความถูกต้องของข้อมูลชนิดใดๆ ก็ตามผู้ใช้จะกำหนดได้ต่อเมื่อ กำหนดชนิดความจริงและชนิดอ้างอิงไว้ครบถ้วนแล้วเท่านั้น โดยการกำหนด uniqueness constraint นั้น ผู้ใช้ไม่จำเป็นต้องกำหนดก็ได้ในกรณีของความสัมพันธ์ชนิด many to many แต่ในกรณีของความสัมพันธ์ชนิด one to one ผู้ใช้จะต้องกำหนด uniqueness constraint สำหรับทั้ง 2 บทบาทความสัมพันธ์ และในกรณีของความสัมพันธ์ชนิด one to many ผู้ใช้จะกำหนดทั้งสองทิศทางเหมือนกับการกำหนดความสัมพันธ์ชนิด one to one หรือกำหนดเพียงทิศทางเดียวก็ได้ เพราะโดยปกติการวิเคราะห์ประโยคในระบบสร้างกฎข้อบังคับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนล จะถือว่าชนิดความจริงหรือชนิดอ้างอิงที่ผู้ใช้กำหนดเป็นความสัมพันธ์ชนิด many to many ยกเว้นว่าจะมีการกำหนด uniqueness constraint เอาไว้

ในกรณีที่ผู้ใช้ต้องการกำหนด uniqueness constraint ทั้งสองทิศทางของความสัมพันธ์ ให้ผู้ใช้กำหนดในลักษณะเดียวกับการกำหนด inverse role สำหรับความสัมพันธ์ใดๆ โดยวิธีการวิเคราะห์กฎข้อบังคับความถูกต้องของข้อมูลชนิด uniqueness เป็นดังนี้

2.4.1 สำหรับการกำหนด uniqueness constraint ของความสัมพันธ์ชนิด one to many เช่น

uk_stgender := one Student has one Gender & one Gender of many Student;

('& one Gender of many Student' ไม่จำเป็นต้องเขียนก็ได้)

จะใช้ตรวจสอบว่าประโยคใดมีคำว่า one สองคำ จะตัดคำว่า one ออกแล้วใช้ประโยคที่เหลือกำหนดเป็นข้อมูลในตารางเมตาสกีมมา IC_ROLE โดยใช้ชื่อกฎข้อบังคับความถูกต้องของข้อมูลที่ผู้ใช้กำหนดบรรจุลงในคอลัมน์ CONST# ของในตารางเมตาสกีมมา IC_ROLE และ CONSTRAINT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 สำหรับการกำหนด uniqueness constraint ของความสัมพันธ์ชนิด many to many เช่น

uk_enroll := one Student enroll_in many Subject & one Subject enrolled_by many Student;

ในกรณีนี้ทุกประโยคจะพบคำว่า one เพียงคำเดียว ซึ่งการใส่ข้อมูลลงในตารางเมตาสกีมมา IC_ROLE และ CONSTRAINT เป็นดังนี้คือ

2.4.2.1 ใช้ชื่อกฎข้อบังคับความถูกต้องของข้อมูล ที่ผู้ใช้กำหนดและประโยคแรก ที่ตัดคำว่า one และ many ออกแล้วบรรจุลงในตารางเมตาสกีมมา IC_ROLE

2.4.2.2 ใช้ชื่อกฎข้อบังคับความถูกต้องของข้อมูล ที่ผู้ใช้กำหนดและเลือกประโยค ที่เป็น inverse role ของประโยคแรกจากตารางเมตาสกีมมา ROLE บรรจุลงในตารางเมตาสกีมมา IC_ROLE

2.4.2.3 ใช้ชื่อกฎข้อบังคับความถูกต้องของข้อมูล ที่ผู้ใช้กำหนด และคำว่า 'UK' บรรจุลงในตารางเมตาสกีมมา CONSTRAINT

2.4.3 สำหรับการกำหนด uniqueness constraint ของความสัมพันธ์ชนิด one to one เช่น

uk_lec_sp := one Lecturer has one Spouse & one Spouse of one Lecturer;

ในกรณีนี้จะพบประโยคที่มีคำว่า one สองคำถึงสองประโยค ซึ่งการใส่ข้อมูลลงในตารางเมตาสกีมมา IC_ROLE และ CONSTRAINT เป็นดังนี้

2.4.3.1 ใช้ชื่อกฎข้อบังคับความถูกต้องของข้อมูล ที่ผู้ใช้กำหนด และประโยคแรก ที่ตัดคำว่า one ออกแล้วบรรจุลงในตารางเมตาสกีมมา IC_ROLE

2.4.3.2 ใช้ชื่อกฎข้อบังคับความถูกต้องของข้อมูล ที่ผู้ใช้กำหนด และคำว่า 'UK' บรรจุลงในตารางเมตาสกีมมา CONSTRAINT

2.4.3.3 ใช้ชื่อกฎข้อบังคับความถูกต้องของข้อมูล ที่ผู้ใช้กำหนดเติมด้วยเลข 11 ข้างท้ายใช้เป็นชื่อกฎข้อบังคับความถูกต้องของข้อมูลสำหรับอีกบทบาทที่เหลือ และเลือก inverse role ของความสัมพันธ์ของประโยคแรกจากตารางเมตาสกีมมา ROLE มาเป็นข้อมูลที่ใช้บรรจุลงในตารางเมตาสกีมมา IC_ROLE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3.4 ใช้ชื่อกฎข้อบังคับความถูกต้องของข้อมูลที่กำหนดใหม่ดังข้างต้น และคำว่า 'UK' บรรจุลงในตารางเมตาสกิมมา CONSTRAINT

2.4.3.4 เลือกความสัมพันธ์ที่ไม่ปรากฏมีบทบาทอยู่ในข้อมูล ROLE_NAME ของตารางเมตาสกิมมา IC_ROLE โดยกฎข้อบังคับความถูกต้องของข้อมูลข้อที่บังคับบทบาทนั้นๆ เป็นชนิด 'UK' และกำหนดชื่อกฎข้อบังคับความถูกต้องของข้อมูลที่จะใช้บังคับกับบทบาททั้งสองของความสัมพันธ์นั้นๆ ให้โดยอัตโนมัติ (1 ชื่อต่อ 2 บทบาท) แล้วบรรจุข้อมูลมีเลือกได้กับชื่อกฎข้อบังคับความถูกต้องของข้อมูลที่กำหนดลงในตารางเมตาสกิมมา IC_ROLE และ CONSTRAINT

2.5 การวิเคราะห์ประโยคที่ใช้กำหนด inter-uniqueness constraint การวิเคราะห์ประโยคที่ใช้กำหนด inter-uniqueness constraint เป็นการวิเคราะห์ประโยคเพื่อนำข้อมูลทีวิเคราะห์ได้มาบรรจุลงในตารางเมตาสกิมมา โดยการใช้อีกกฎข้อบังคับความถูกต้องของข้อมูลที่ใช้กำหนดกับแต่ละบทบาทที่ปรากฏในวงเล็บสำหรับใช้เป็นข้อมูลแต่ละ row ในตารางเมตาสกิมมา IC_ROLE และใช้ชื่อกฎข้อบังคับความถูกต้องของข้อมูลนั้นๆ กับคำว่า 'UK' เพื่อบรรจุลงในตารางเมตาสกิมมา CONSTRAINT

2.6 การวิเคราะห์ประโยคที่ใช้กำหนดกฎข้อบังคับความถูกต้องของข้อมูลชนิด mandatory role ใช้ชื่อกฎข้อบังคับความถูกต้องของข้อมูลที่ใช้กำหนดและเลือก inverse role ของบทบาทที่แสดงในประโยคกำหนดกฎข้อบังคับความถูกต้องของข้อมูลชนิด mandatory role จากตารางเมตาสกิมมา ROLE เพื่อบรรจุลงในตารางเมตาสกิมมา IC_ROLE และ CONSTRAINT

2.7 การวิเคราะห์ประโยคที่ใช้กำหนดกฎข้อบังคับความถูกต้องของข้อมูลชนิด mandatory entity และกฎชนิดอื่นๆ ลักษณะของการวิเคราะห์ข้อมูลจะคล้ายกับกฎอื่นๆ ที่กล่าวมาแล้ววตอนต้น

2.8 การวิเคราะห์ประโยคที่ใช้กำหนด transition of string values constraint เป็นการนำลำดับของข้อมูลในวงเล็บมาบรรจุลงในตารางเมตาสกิมมา TRANS_STRING โดยจะสร้างลำดับที่สำหรับแต่ละค่าของข้อมูลให้โดยอัตโนมัติ สำหรับกรณีที่ผู้ใช้ใช้วงเล็บ '<....>' ซึ่งหมายถึง

ลำดับการเปลี่ยนแปลงค่าของข้อมูลที่วนรอบมาที่ค่าเริ่มต้นใหม่ได้ จะใช้ค่า 'tr_cy' สำหรับคอลัมน์ CONST_TYPE แต่ถ้าเป็น '(...)' จะใช้ 'tr_str' แทน

2.9 การวิเคราะห์ประโยคที่ใช้กำหนด transition of numeric values constraint จะนำเอาทิศทางการเปลี่ยนแปลงค่าของข้อมูล และข้อจำกัดในการเปลี่ยนแปลงที่แสดงในประโยคบรรจุลงในตารางเมตาสกีม่า TRANS_VALUE สำหรับประโยคที่บ่งชี้ข้อห้ามในการเปลี่ยนแปลงค่าข้อมูล เช่น

sal_inc := Salary of Person is not decreasingly;

ซึ่งหมายความว่าในระบบงานนี้จะไม่มีการลดเงินเดือนของพนักงานโดยเด็ดขาด ดังนั้นสำหรับกฎข้อนี้จึงต้องให้ค่าข้อมูลแก่คอลัมน์ TRANS_DIRECTION และคอลัมน์ INVALID_DIRECTION เป็น 'D' และ 'Y' ตามลำดับ สำหรับกฎที่มีชื่อว่า 'sal_inc'

2.10 การวิเคราะห์ประโยคที่ใช้กำหนด dynamic constraint จะมีลักษณะเหมือนกับการวิเคราะห์ประโยค transition of numeric value เพียงแต่เพิ่มข้อมูลช่วงระยะเวลาของการเปลี่ยนแปลงด้วยเท่านั้น

2.11 การวิเคราะห์ประโยคสำหรับกำหนด user-defined constraint ใช้การตัดเอาประโยคเงื่อนไขที่พบภายใน if clause, then clause และ exception clause ไปเก็บลงในคอลัมน์ IF_CLAUSE, THEN_CLAUSE และ ACTION_CLAUSE ตามลำดับ

3. Schema Compiler

โปรแกรมส่วนนี้พัฒนาขึ้นด้วยโปรแกรม Delphi เช่นเดียวกัน โดยการพัฒนาโปรแกรมได้แบ่งออกเป็น 4 ขั้นตอนคือ

3.1 สร้างข้อมูลแม่ป้าง

3.2 สร้างข้อมูล referential integrity

3.3 สร้างข้อมูล elementary object ของ composit object

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 สร้างคำสั่ง SQL ที่ใช้สร้างตารางฐานข้อมูล และกฎข้อบังคับความถูกต้องของข้อมูลชนิด PRIMARY KEY และ FOREIGN KEY (referential integrity)

โดยขั้นตอนทั้งหมดจะอ้างอิงกับอัลกอริทึมในการสร้างรีเลชันแนลสกีมมาที่จะอธิบายต่อไปนี้

อัลกอริทึมในการสร้างรีเลชันแนลสกีมมา

การสร้างรีเลชันแนลสกีมมาจากข้อมูลที่มีอยู่ในตารางเมตาสกีมมานั้น อาศัยแนวคิดจากอัลกอริทึม ONF ในการแก้ปัญหาความสัมพันธ์เป็นรีเลชัน ซึ่งอัลกอริทึม ONF จำเป็นต้องใช้ข้อมูลดังต่อไปนี้คือ

- ชนิดเอนติตี้ และชนิดเลเบิ้ลที่ประกอบขึ้นเป็นความสัมพันธ์ใดๆ
- คีย์ หมายถึง บทบาทของชนิดเอนติตี้ หรือชนิดเลเบิ้ล ที่ถูกบังคับด้วย uniqueness

constraint

โดยการที่จะสามารถใช้อัลกอริทึม ONF ได้นั้น ต้องสามารถเลือกข้อมูลที่เป็นจากตารางเมตาสกีมมาได้เสียก่อน เนื่องจากข้อมูลที่เก็บอยู่ในตารางเมตาสกีมมานั้นถูกแจกแจงจากประโยชน์มูลฐานออกเป็นข้อมูลย่อยๆ มากมาย เพื่อให้สามารถเก็บรายละเอียดของความสัมพันธ์และกฎข้อบังคับความถูกต้องของข้อมูลที่บังคับได้อย่างครบถ้วน ดังนั้นจึงจำเป็นต้องใช้ประโยคคำสั่ง SELECT ในการเลือกข้อมูลที่ต้องการจากข้อมูลภายในตารางเมตาสกีมมา

ก่อนจะอธิบายถึงชุดประโยคคำสั่ง SQL ที่ใช้ในขั้นตอนการสร้างรีเลชันแนลสกีมมานี้ ต้องขอเกริ่นความหมายของคำว่าอ็อบเจกต์เพื่อให้สามารถเข้าใจคำอธิบายของชุดประโยคคำสั่งได้ง่ายขึ้นดังนี้

object ในที่นี้สามารถจะเป็นได้ทั้ง ชนิดเอนติตี้, ชนิดเลเบิ้ล, alias หรือชนิดความจริง โดย object สามารถพิจารณาแบ่งออกได้เป็น 2 จำพวกคือ elementary object และ composite object

- elementary object คืออ็อบเจกต์ชนิดที่ไม่อาจแบ่งแยกออกเป็นอ็อบเจกต์ย่อยๆ ได้ อีก ได้แก่ ชนิดเอนติตี้ และ ชนิดเลเบิ้ล

- composite object คืออ็อบเจกต์ชนิดที่แบ่งแยกออกเป็นอ็อบเจกต์ย่อยๆ ได้ ซึ่งก็ได้แก่ ชนิดความจริง และ alias เนื่องจากชนิดความจริงประกอบด้วยชนิดเอนติตี้ อย่างน้อย 2 ชนิดขึ้นไป และ alias คือชื่อเรียกแทนชนิดความจริงที่ผู้ใช้กำหนดขึ้น

ที่ต้องแบ่งชนิดของอ็อบเจกต์เป็นเช่นนี้ เนื่องจากว่าโครงสร้างของภาษาที่กำหนดใช้ในระบบสร้างกฎบังคับกับความถูกต้องของข้อมูลโดยใช้ภาษารรรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนล สามารถรองรับชนิดความจริงที่มีขนาดมากกว่าไบนารีได้ และสามารถจะกำหนด alias ให้แก่ชนิดความจริงใดๆ และใช้ alias เหล่านั้นกำหนดเป็นประธานหรือกรรมของประโยคความสัมพันธ์ได้ด้วย และขั้นตอนของการวิเคราะห์ข้อมูลจากประโยคความสัมพันธ์ได้แสดงความสัมพันธ์ที่มีขนาดมากกว่าไบนารีออกเป็นความสัมพันธ์ย่อยๆ โดยถือเสมือนว่าความสัมพันธ์ใดๆ ที่เก็บลงในตารางเมตาสกีมานั้นเป็นความสัมพันธ์ชนิดไบนารีทั้งหมด ซึ่งประธานของประโยคอาจจะเป็นความสัมพันธ์ใดๆ ได้ ดังนั้นการจะประมวลข้อมูลในตารางเมตาสกีมามา เพื่อนำมาใช้ในการสร้างรีเลชันแนลสกีมนั้น จำเป็นที่จะต้องแสดง elementary object ทั้งหมดที่ประกอบขึ้นเป็นตารางความสัมพันธ์ใดๆ ได้ และสามารถแสดงได้ว่าบทบาทใดของชนิดเอนติตี้ใดบ้าง ที่ถูกบังคับด้วยกฎข้อบังคับความถูกต้องของข้อมูลชนิด uniqueness โดยรูปประโยคคำสั่งภาษา SQL ที่ใช้ในการเลือกข้อมูลที่ต้องการออกมาจากตารางเมตาสกีมามา ซึ่งประกอบไปด้วยชุดคำสั่งหลักที่ทำหน้าที่ดังต่อไปนี้คือ

- ALGO1 หาชื่อของความสัมพันธ์ชนิด one to many ทั้งหมด
- ALGO2 หา non-prime attribute (บทบาทของชนิดเอนติตี้ในความสัมพันธ์ที่ไม่ถูกบังคับโดย uniqueness constraint) ของความสัมพันธ์ชนิด one to many
- ALGO3 หาความสัมพันธ์ (relationship name) และคีย์ทั้งสองของความสัมพันธ์ชนิด one to one ทั้งหมด
- ALGO4 หาความสัมพันธ์และคีย์ของความสัมพันธ์ชนิด many to many ทั้งหมด โดยชุดคำสั่ง SQL ดังกล่าวเป็นดังนี้คือ

ALGO1

```
select distinct object_name from role where role_name in
(select role_name from ic_role where const# in
(select const# from constraint where const_type = 'uk' and const# in
(select const# from ic_role group by const# having count(*) = 1)))
and relationship_name not in
(select relationship_name from role where role_name in
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
(select role_name from ic_role where const# in
(select const# from constraint where const_type = 'uk' and const# in
(select const# from ic_role group by const# having count(*) = 1)))
group by relationship_name having count(*) = 2 );
```

ALGO2

```
select object_name, role_name, relationship_name from role t1 where relationship_name in
(select relationship_name from role t2 where object_name = '&1'
and t1.role_name <> t2.role_name and t2.role_name in
(select role_name from ic_role where const# in
(select const# from constraint where const_type = 'uk' and const# in
(select const# from ic_role group by const# having count(*) = 1))))
and relationship_name not in
(select relationship_name from role where role_name in
(select role_name from ic_role where const# in
(select const# from constraint where const_type = 'uk' and const# in
(select const# from ic_role group by const# having count(*) = 1))))
group by relationship_name having count(*) = 2);
```

ALGO3

```
select t1.object_name, t2.object_name ,t1.role_name, t2.role_name ,t1.relationship_name
from role t1, role t2
where t1.relationship_name = t2.relationship_name
and t1.object_name <= t2.object_name and t1.role_name < t2.role_name
and t1.relationship_name in
(select relationship_name from role where role_name in
(select role_name from ic_role where const# in
(select const# from constraint where const_type = 'uk' and const# in
(select const# from ic_role group by const# having count(*) = 1)))
group by relationship_name having count(*) = 2 );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ALGO4

```

select t1.object_name, t2.object_name, t1.role_name, t2.role_name, t1.relationship_name
from role t1, role t2
where t1.role_name < t2.role_name and t1.relationship_name = t2.relationship_name
and t1.object_name <= t2.object_name and t1.role_name in
(select role_name from ic_role where const# in
(select const# from constraint where const_type = 'uk' and const# in
(select const# from ic_role group by const# having count(*) = 2)))
and t1.relationship_name not in
(select distinct object_name from role where role_name in
(select role_name from ic_role where const# in
(select const# from constraint where const_type = 'uk' and const# in
(select const# from ic_role group by const# having count(*) = 1))))
and t1.relationship_name not in
(select object_name from role where role_name in
(select role_name from ic_role where const# in
(select const# from constraint where const_type = 'uk' and const# in
(select const# from ic_role group by const# having count(*) = 2))))
and t1.relationship_name not in
(select relationship_name from alias where alias in
(select object_name from role where role_name in
(select role_name from ic_role where const# in
(select const# from constraint where const_type = 'uk' and const# in
(select const# from ic_role group by const# having count(*) = 2))))))
order by t1.relationship_name;

```

ข้อมูลที่เป็นผลลัพธ์จากการทำงานของชุดคำสั่ง SQL ข้างต้นนำมาใช้ในขั้นตอนของอัลกอริทึมในการสร้างรีเลชันแนลสกีมา โดยอัลกอริทึมในการสร้างรีเลชันแนลสกีมานี้จะแตกต่างจากอัลกอริทึม ONF ในเรื่องของรายละเอียดและขั้นตอน เนื่องจากในระบบนี้ตารางเมตาสกีมาได้ถูกเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกแบบให้เก็บความสัมพันธ์ชนิดต่างๆ ในรูปของความสัมพันธ์ชนิดไบนารีเท่านั้น แต่อาศัยหลักการเดียวกับอัลกอริทึม ONF ในการจับกลุ่มความสัมพันธ์ย่อยๆ ที่มีคีย์ร่วมกัน ทำให้ขั้นตอนการสร้างรีเลชันแนลสกีมมาจะเป็นดังต่อไปนี้

3.1 ขั้นตอนที่ 1 สร้างข้อมูลแม่ปิ้ง เป็นการสร้างข้อมูลเพื่อแสดงว่า role ใดถูกแม่ปิ้งเป็นคอลัมน์ใดในตารางความสัมพันธ์ใด และ role ใดถูกเลือกเป็นไพรมารีคีย์ หรือ คีย์คู่แข่ง (candidate key)

3.1.1 ใช้ ALGO1 หา คีย์ของความสัมพัธ์ชนิด one to many ทั้งหมด

3.1.2 ใส่ข้อมูลไพรมารีคีย์ (PRIMARY_KEY) และชื่อตาราง (TABLE_NAME) ลงในตารางเมต้าสกีมมา PK ด้วยชื่อของคีย์ที่ได้จากข้อ 3.1.1

3.1.3 ใช้ ALGO2 หา non-prime attribute ทั้งหมดสำหรับแต่ละคีย์ของความสัมพัธ์ชนิด one to many นั้น

3.1.4 ใส่ข้อมูลชื่อตาราง (TABLE_NAME) ด้วยชื่อของคีย์ลงในตารางเมต้าสกีมมา ROLE โดยตรวจสอบก่อนว่าชื่อของอ็อบเจกต์ทั้งหมดของแต่ละคีย์เหมือนกันหรือไม่ และเหมือนกับชื่อของคีย์หรือไม่ ถ้าพบว่าชื่อซ้ำกันก็ให้ใช้ คำกริยา ของข้อมูลชื่อความสัมพันธ์ (RELATIONSHIP_NAME) มาบรรจุลงในคอลัมน์ (NAME_EXTEND)

3.1.5 สำหรับอ็อบเจกต์ที่เป็น non-prime attribute ของความสัมพันธ์ที่มีคีย์เป็นชนิดเอนติตี้ซัพไทป์ให้ใส่ข้อมูลลงในคอลัมน์ (COMMENT) ด้วยชื่อของชนิดเอนติตี้ซัพไทป์นั้น

3.1.6 ใช้ ALGO4 หา คีย์ทั้งสอง (key1, key2) ของความสัมพันธ์ชนิด one to one

3.1.7 ตรวจสอบว่า key1 ปรากฏเป็นคีย์ของความสัมพัธ์ชนิด one to many หรือไม่

3.1.8 ตรวจสอบว่า key2 ปรากฏเป็นคีย์ของความสัมพัธ์ชนิด one to many หรือไม่

3.1.9 ถ้าพบว่าทั้ง key1 และ key2 ต่างก็เป็นคีย์ของความสัมพัธ์ชนิด one to many อื่นๆ ให้ตรวจสอบชนิดของอ็อบเจกต์ที่เป็น key1 และ key2

3.1.9.1 ถ้าคีย์ key1 หรือ key2 เป็นอ็อบเจกต์ชนิดเอนติตี้เพียงอ็อบเจกต์เดียว ให้เลือกคีย์ที่เป็นอ็อบเจกต์ชนิดเอนติตี้นั้นเป็นไพรมารีคีย์

3.1.9.2 ถ้าคีย์ key1 หรือ key2 เป็นอ็อบเจกต์ชนิดเอนติตี้หรือชนิดเลเบลหรือชนิดความจริง หรือชนิด alias ทั้งคู่ หรือเป็นอ็อบเจกต์ชนิดความจริง และชนิด alias ให้เลือกไพรมารีคีย์จากคีย์ที่มี non-prime attribute มากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.9.3 ลบข้อมูลในตารางเมตาสกีมมา PK สำหรับข้อมูลที่มีชื่อตารางเหมือนกับชื่อของ key1 หรือ key2 ที่ไม่ได้ถูกเลือกเป็นไพรมารีคีย์ (candidate key)

3.1.9.4 เช็ดค่าข้อมูลในคอลัมน์ TABLE_NAME ของ non-prime attribute ของ candidate key ให้เท่ากับชื่อของไพรมารีคีย์ และกรณีที่มีชื่อของอ็อบเจกต์นั้นซ้ำกับชื่ออ็อบเจกต์ของ non-prime attribute อื่นของไพรมารีคีย์เดียวกัน ให้ใช้คำกริยาของความสัมพันธ์มาบรรจุลงในคอลัมน์ NAME_EXTEND

3.1.9.5 ใส่ข้อมูลในคอลัมน์ TABLE_NAME ของ candidate key ด้วยชื่อของไพรมารีคีย์ และต้องตรวจสอบชื่อของอ็อบเจกต์เช่นกันว่า ซ้ำกับอ็อบเจกต์อื่นที่ถูกแม่ปบนตารางความสัมพันธ์เดียวกัน (TABLE_NAME) หรือไม่ และใส่ข้อมูลลงในคอลัมน์ COMMENT ด้วยค่า 'CK'

3.1.10 ถ้าพบว่า key1 หรือ key2 เท่านั้นที่เป็นคีย์ของความสัมพันธ์ชนิด one to many ให้นำใส่ข้อมูลชื่อคอลัมน์ด้วยชื่อของอ็อบเจกต์ และชื่อตารางด้วยชื่อคีย์ของความสัมพันธ์ชนิด one to many นั้น

3.1.11 ถ้าไม่พบว่า key1 หรือ key2 เป็นคีย์ของความสัมพันธ์ one to many ใดๆ ให้ตรวจสอบชนิดของอ็อบเจกต์ของคีย์ทั้งสองและเลือกไพรมารีคีย์ จากคีย์ที่มีชนิดอ็อบเจกต์เป็นชนิดเอนติตี้ หรือเลือกใช้ key1 ในกรณีที่คีย์ทั้งสองมีชนิดของอ็อบเจกต์เหมือนกัน

3.1.12 ใส่ข้อมูลลงในตารางเมตาสกีมมา PK ทั้งสองคอลัมน์ด้วยชื่อของไพรมารีคีย์ที่เลือกไว้

3.1.13 ใส่ค่าข้อมูลชื่อคอลัมน์และชื่อตารางของ candidate key ของความสัมพันธ์ชนิด one to one ข้างต้น ด้วยชื่อของ อ็อบเจกต์ และชื่อของไพรมารีคีย์ และเช็ดค่าในคอลัมน์ COMMENT ให้เท่ากับ 'CK'

3.1.14 เช็ดให้ TABLE_NAME เท่ากับ '-' สำหรับ role ต่างๆ ที่มีอ็อบเจกต์เป็นไพรมารีคีย์ของตารางความสัมพันธ์ทั้งหมดที่กำหนดขึ้น

3.1.15 เช็ดให้ TABLE_NAME เท่ากับ '-' สำหรับ role ต่างๆ ที่มีอ็อบเจกต์เป็น candidate key โดย role นั้นยังไม่มีค่าข้อมูล TABLE_NAME อยู่ (null)

3.1.16 ใช้ ALGO5 หาความสัมพันธ์ชนิด many to many และอ็อบเจกต์ ทั้งสองที่เป็นคีย์ความสัมพันธ์นั้น

3.1.17 ตรวจสอบว่าความสัมพันธ์ชนิด many to manyใด ปรากฏเป็นไพรมารีคีย์หรือ candidate key ของตารางความสัมพันธ์สัมพันธ์ที่กำหนดไว้แล้ว ให้ใส่ข้อมูลในคอลัมน์ TABLE_NAME ด้วยคำว่า 'key'

3.1.18 ตรวจสอบว่าความสัมพันธ์ชนิด many to many ใด ปรากฏเป็นส่วนหนึ่งของ คีย์ความสัมพันธ์ชนิด many to many อื่น ให้ใส่ข้อมูลในคอลัมน์ TABLE_NAME ด้วย 'nnkey'

3.1.19 สำหรับความสัมพันธ์ชนิด many to many ที่เหลือ ให้แม่เป็นหนึ่งตาราง ความสัมพันธ์ต่อหนึ่งความสัมพันธ์ โดยใส่ข้อมูลลงในตารางเมตาสกีมมา PK ทั้งสองคอลัมน์ด้วย ชื่อของความสัมพันธ์.

3.1.20 ใส่ข้อมูลใน TABLE_NAME บนตารางเมตาสกีมมา ROLE ของ role ที่ ประกอบเป็นคีย์ของความสัมพันธ์ชนิด many to many ด้วยชื่อของความสัมพันธ์

3.2 ขั้นตอนที่ 2 สร้างข้อมูล referential integrity

3.2.1 เลือกข้อมูลของ role ที่มีอ็อปเจ็คต์ปรากฏเป็นไพมารีย์คีย์ของตารางความสัมพันธ์ที่กำหนดขึ้นจากตารางเมตาสกีมมา ROLE ยกเว้น role ที่มีค่าข้อมูลในคอลัมน์ TABLE เท่ากับ '-', 'key' หรือ 'nnkey'

3.2.2 กำหนดข้อมูลในคอลัมน์ REFERENCE บนตารางเมตาสกีมมา ROLE ของ role นั้นให้มีค่าเท่ากับชื่อของตารางความสัมพันธ์ที่อ้างอิงถึง

3.3 ขั้นตอนที่ 3 สร้างข้อมูล elementary object ของแต่ละ composite object

3.3.1 สำหรับอ็อปเจ็คต์ที่เป็นชนิดความจริง หรือชนิด alias ให้หา 2 อ็อปเจ็คต์ ใดๆ ที่ประกอบขึ้นเป็นชนิดความจริงนั้นๆ (หรือชนิดความจริงที่อ้างอิงด้วยชื่อ alias นั้น)

3.3.2 ตรวจสอบว่าอ็อปเจ็คต์ทั้งสองที่ได้จากข้อ 1 เป็นอ็อปเจ็คต์ชนิดเอนติตี้ หรือ ชนิดเลเบิลหรือไม่

3.3.3 ถ้าพบว่าอ็อปเจ็คต์ที่แยกออกมายังไม่เป็น elementary object ให้ย้อนกลับไป ทำตามวิธีการในข้อที่ 1 และ 2 จนกว่าอ็อปเจ็คต์ทั้งหมดที่แยกออกมา จะเป็นชนิดเอนติตี้หรือชนิด เลเบิลเท่านั้น

3.3.4 นำข้อมูลที่ใส่ลงในตารางเมตาสกีมมา COMPOS_OBJ

3.4 ขั้นตอนที่ 4 สร้างคำสั่ง SQL ที่ใช้สร้างตารางฐานข้อมูล และกฎข้อบังคับความถูกต้องของข้อมูลชนิด PRIMARY KEY และ FOREIGN KEY (referential integrity)

3.4.1 สร้างคำสั่ง CREATE TABLE โดยใช้ข้อมูลในคอลัมน์ TABLE_NAME ในตารางเมตาสกีมมา PK เป็นชื่อของตารางฐานข้อมูล

3.4.2 ตรวจสอบว่าไพรมารีชีของตารางความสัมพันธ์นั้นเป็น elementary object หรือไม่

3.4.2.1 ถ้าใช่ ให้ใช้ข้อมูลชื่อเลเบล, ชนิดข้อมูล และขนาดข้อมูล จากตารางเมตาสกีมมา IDENTIFIER ของไพรมารีชีของแต่ละตารางความสัมพันธ์กำหนดเป็นคอลัมน์ที่กำหนดในตอนต้นของแต่ละตารางฐานข้อมูล

3.4.2.2 ถ้าไม่ใช่ ให้ใช้ข้อมูลจากตารางเมตาสกีมมา COMPOS_OBJ เพื่อเลือกข้อมูลชื่อเลเบล, ชนิดข้อมูล และขนาดข้อมูล จากตารางเมตาสกีมมา IDENTIFIER ของไพรมารีชีกำหนดเป็นคอลัมน์ที่กำหนดในตอนต้นของแต่ละตารางฐานข้อมูล

3.4.3 เลือกข้อมูล non-prime attribute ของตารางความสัมพันธ์เพื่อนำมากำหนดเป็นคอลัมน์อื่นๆ ของตารางฐานข้อมูลที่เฝ้าจากตารางความสัมพันธ์นั้นๆ โดยวิธีการกำหนดชื่อ, ชนิดข้อมูล และขนาดของคอลัมน์ก็ใช้วิธีเดียวกับที่ใช้ในการกำหนดคอลัมน์ของไพรมารีชีดังแสดงไว้ในข้อ 2 เพียงแต่ถ้าหากพบว่าค่าข้อมูลในคอลัมน์ NAME_EXTEND ของตารางเมตาสกีมมา ROLE ไม่เป็นค่า null ให้ใช้ค่านั้นมาเพิ่มหน้าชื่อเลเบลเพื่อที่จะนำมากำหนดเป็นชื่อของคอลัมน์

3.4.4 กำหนดกฎข้อบังคับความถูกต้องของข้อมูลชนิด NOT NULL สำหรับคอลัมน์ที่เฝ้าจาก role ที่ถูกบังคับด้วย mandatory role constraint

3.4.5 สร้างคำสั่ง ALTER TABLE table_name ADD CONSTRAINT ic_name PRIMARY KEY ด้วยการกำหนดชื่อของตารางฐานข้อมูลและคอลัมน์ที่กำหนดจากอ็อปเจ็คต์ที่เป็นไพรมารีชี

3.4.6 สร้างคำสั่ง ALTER TABLE table_name ADD CONSTRAINT ic_name FOREIGN KEY สำหรับ role ที่พบว่าข้อมูลในคอลัมน์ REFERENCE บนตารางเมตาสกีมมา ROLE ไม่เป็นค่า null ด้วยการกำหนดชื่อของคอลัมน์ไพรมารีชีของตารางฐานข้อมูล และชื่อของตารางฐานข้อมูล ที่อ้างอิงถึงด้วยชื่ออ็อปเจ็คต์ของ role นั้น และค่าข้อมูลในคอลัมน์ REFERENCE

4. rule generator

กฎข้อบังคับความต้องการของข้อมูลแต่ละชนิดจะถูกกำหนดรูปแบบและโครงสร้างของกฎที่ใช้ในการบังคับไว้ โดยได้มีการศึกษาว่ากฎข้อบังคับความต้องการของข้อมูลแต่ละชนิดเมื่อมีความจำเป็นในการรักษาความต้องการของข้อมูลต่างลักษณะกัน จะมีโครงสร้างของกฎที่กำหนดเป็นกฎชนิดอรรถาธิบายและกฎชนิด procedural ต่างกันอย่างไร โครงสร้างของกฎแต่ละชนิดถูกแสดงไว้ภาคผนวก ก. โดยที่ตัวอักษรที่บ่งหมายถึงเป็นวลีและคำที่กำหนดใช้อย่างตายตัวสำหรับกฎข้อบังคับความต้องการของข้อมูลแต่ละชนิด แต่ตัวอักษรเอียงหมายถึงคำหรือวลีที่เปลี่ยนแปลงไปตามรายละเอียดของกฎข้อบังคับความต้องการของข้อมูลแต่ละข้อ

ขั้นตอนในการสร้างคำสั่งสำหรับกำหนดกฎข้อบังคับความต้องการของข้อมูลชนิดต่างๆ

4.1 สร้างคำสั่งสำหรับสร้างกฎข้อบังคับความต้องการของข้อมูลชนิดอรรถาธิบาย

(CHECK constraint) สร้างคำสั่ง ALTER TABLE table_name ADD CONSTRAINT ic_name CHECK constraint สำหรับคอลัมน์ที่เฝ้ามาจากชนิดเอนติตี้ที่ถูกบังคับด้วยกฎข้อบังคับความต้องการของข้อมูลชนิด entity type โดยใช้ข้อมูลในคอลัมน์ CONST# เป็นชื่อกฎข้อบังคับความต้องการของข้อมูล และข้อมูลอื่นๆ ในตารางเมตาสกิมมา MEMBERSHIP และ RANGE ในการกำหนดรายละเอียดของกฎ

4.2 สร้างคำสั่งสำหรับกำหนดกฎข้อบังคับความต้องการของข้อมูลชนิด procedural

สร้างคำสั่ง CREATE PROCEDURE และคำสั่ง CREATE TRIGGER เพื่อกำหนด stored procedure และ database trigger สำหรับตารางฐานข้อมูลเอนติตี้, role หรือ relation ที่ถูกบังคับด้วยกฎข้อบังคับความต้องการของข้อมูลอยู่ ซึ่งเลือกข้อมูลได้จากตารางเมตาสกิมมา IC_OBJ, IC_ROLE และ IC_REL โดย

4.2.1 กำหนดชื่อของ stored procedure จาก CONST#

4.2.2 ใช้ข้อมูลที่อ่านได้จากชื่อของคอลัมน์ OBJECT และ NAME_EXTEND ของตารางเมตาดัสมิมา ROLE มาประกอบเป็นชื่อของคอลัมน์ที่ถูกบังคับ สำหรับกฎที่บังคับกับบทบาทหรือความสัมพันธ์

4.2.3 นำข้อมูลจากตารางเมตาดัสมิมา ที่เก็บรายละเอียดของกฎข้อบังคับความต้องการของข้อมูลชนิดต่างๆ มาบรรจุลงในแพทเทิร์นของ stored procedure ที่เกี่ยวข้องกับกฎแต่ละชนิด

4.2.4 กำหนดชื่อของ database trigger จากชื่อของตารางฐานข้อมูลรวมกับตัวอักษรและคำต่อไปนี้ A, B และ INS, UPD, DEL และ ROW

4.2.5 กำหนดเรียกใช้ stored procedure ภายใน database trigger ตามชนิดของกฎข้อบังคับความต้องการของข้อมูลที่วิเคราะห์ไว้



บทที่ 8

การทดสอบและใช้งานระบบ

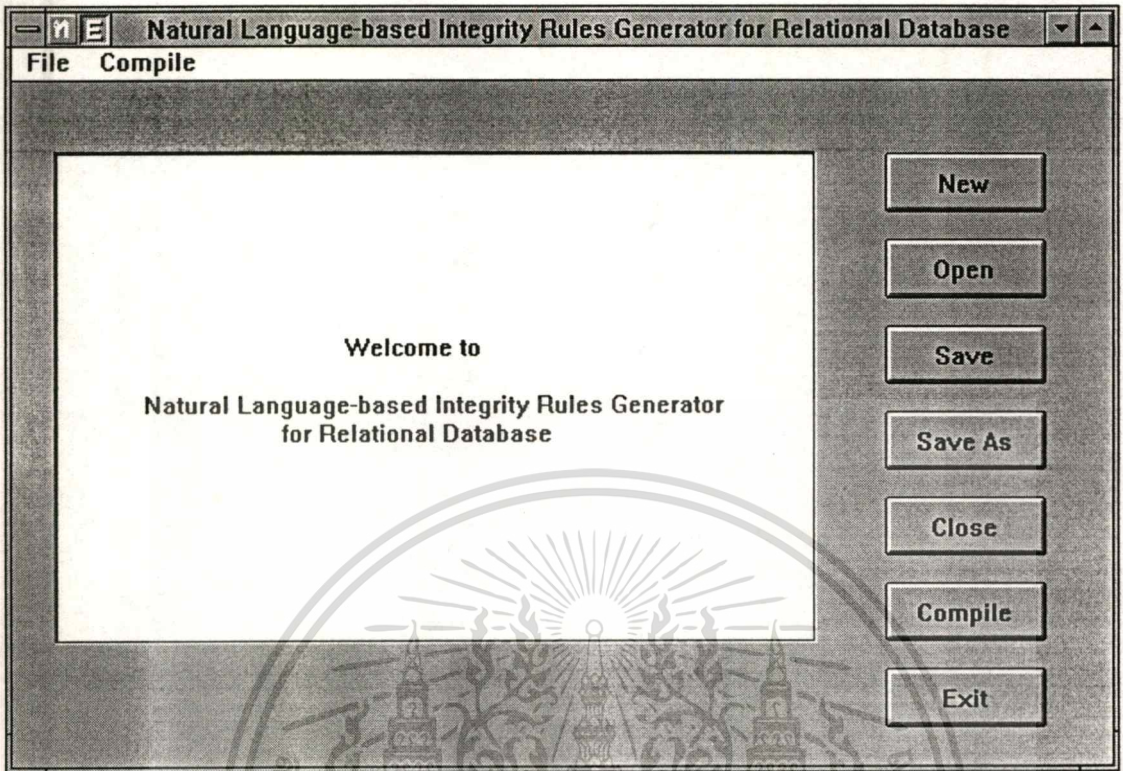
การใช้งานระบบ

ระบบสร้างกฎบังคับกับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนล เป็นโปรแกรมที่พัฒนาโดยโปรแกรม Delphi ซึ่งเป็นเครื่องมือที่ใช้พัฒนาโปรแกรมประยุกต์สำหรับฐานข้อมูลบน Windows ดังนั้นเมื่อต้องการใช้งานระบบสร้างกฎบังคับกับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนล ผู้ใช้จำเป็นต้องมีปัจจัยต่างๆ ต่อไปนี้คือ

1. เครื่องไมโครคอมพิวเตอร์ IBM PC คอมแพคทิเบิล ที่มี CPU ตั้งแต่รุ่น 386DX ขึ้นไป และต้องมีหน่วยความจำอย่างน้อย 8 เมกะไบต์
2. โปรแกรม Windows เวอร์ชัน 3.1 ขึ้นไป
3. โปรแกรม Delphi เวอร์ชัน 1.0 ในส่วนของ Database Engine
4. เอ็กซিকิวทีฟไฟล์ของโปรแกรม cvs.exe

ขั้นตอนการใช้งานระบบ

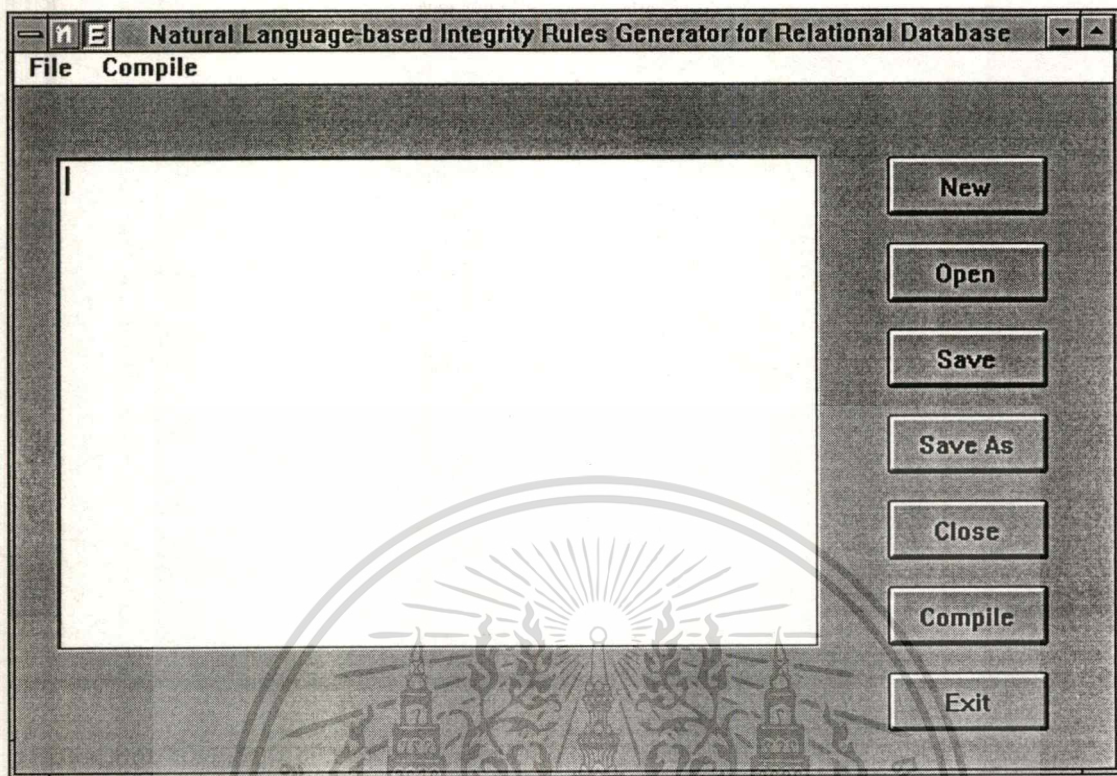
1. เปิดเครื่องคอมพิวเตอร์ และเรียกใช้งานโปรแกรม Windows
2. ใช้เมาส์ดับเบิลคลิกที่ไอคอนของโปรแกรม CVS จะปรากฏวินโดว์ดังภาพที่ 45



ภาพที่ 45 หน้าจอหลัก

ซึ่งการใช้งานโปรแกรม CVS นี้จะมีลักษณะเดียวกับการใช้งานโปรแกรม Windows ทุกประการ คือผู้ใช้สามารถเลือกการทำงานได้จากเมนู หรือใช้การคลิกปุ่มต่างๆ ที่กำหนดไว้บนหน้าจอ ด้วยการใช้นิ้วเหล่านี้ผู้ใช้ไม่จำเป็นต้องเปิดเมนูเพื่อเลือกคำสั่งการทำงานก็ได้ เนื่องจากผู้ออกแบบระบบได้สร้างปุ่มต่างๆ ไว้ครบสำหรับทุกฟังก์ชันการใช้งาน ดังนั้นจึงขออธิบายการใช้งานของระบบ ด้วยการใช้นิ้วเหล่านี้เท่านั้น ซึ่งปุ่มต่างๆ มีความหมายและผลลัพธ์ดังนี้คือ

NEW หมายถึงคำสั่งให้เคลียร์หน้าจอของโปรแกรม เพื่อรับการป้อนข้อมูล เมื่อผู้ใช้คลิกที่ปุ่ม New ผลลัพธ์จะเป็นดังที่แสดงไว้ในภาพที่ 46

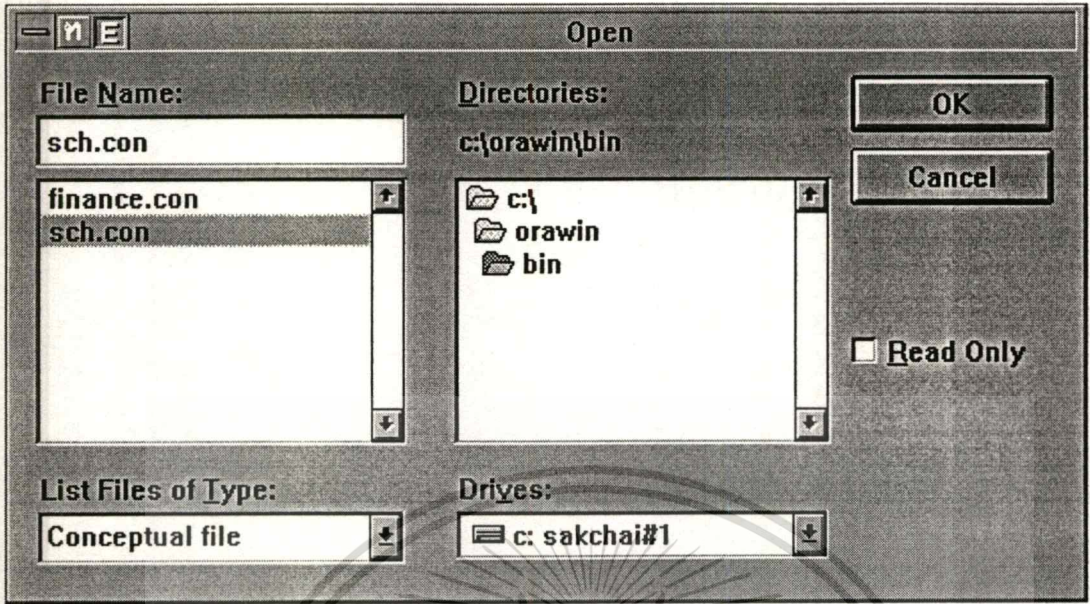


ภาพที่ 46 หน้าจอหลังจากที่ผู้ใช้คลิกปุ่ม New

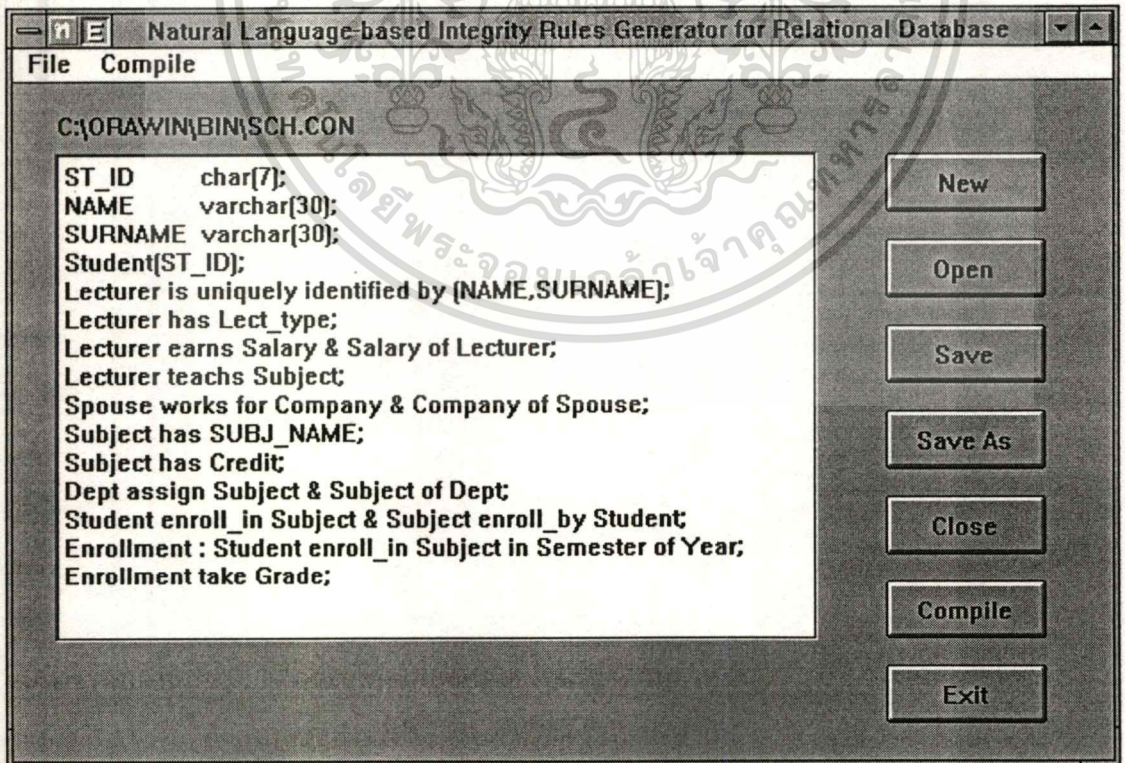
OPEN หมายถึง คำสั่งที่ใช้เรียก Open Dialog Box สำหรับเปิดไฟล์ conceptual schema ซึ่งเป็นไฟล์ข้อความที่บรรจุประโยคคำสั่งภาษาธรรมชาติ เช่นเดียวกับการเปิดไฟล์ด้วยโปรแกรม Windows อื่นๆ การคลิกที่ปุ่ม Open จะได้ผลลัพธ์ดังหน้าจอที่แสดงไว้ในภาพที่ 47

และเมื่อผู้ใช้เปิดไฟล์ที่มีข้อมูลอยู่แล้ว ข้อความต่างๆ ภายในไฟล์ที่เปิด จะปรากฏอยู่บนหน้าจอของโปรแกรม ดังที่แสดงไว้ในภาพที่ 48 ซึ่งผู้ใช้สามารถที่จะเพิ่มเติม หรือแก้ไขข้อมูลได้ตามที่ต้องการ

SAVE หมายถึง คำสั่งที่ใช้ในการเรียก Save Dialog Box สำหรับเก็บข้อมูลที่แก้ไขบนหน้าจอของโปรแกรมลงสู่ไฟล์ที่ต้องการ ซึ่งเมื่อผู้ใช้คลิกปุ่ม Save จะปรากฏหน้าจอ ดังภาพที่ 49

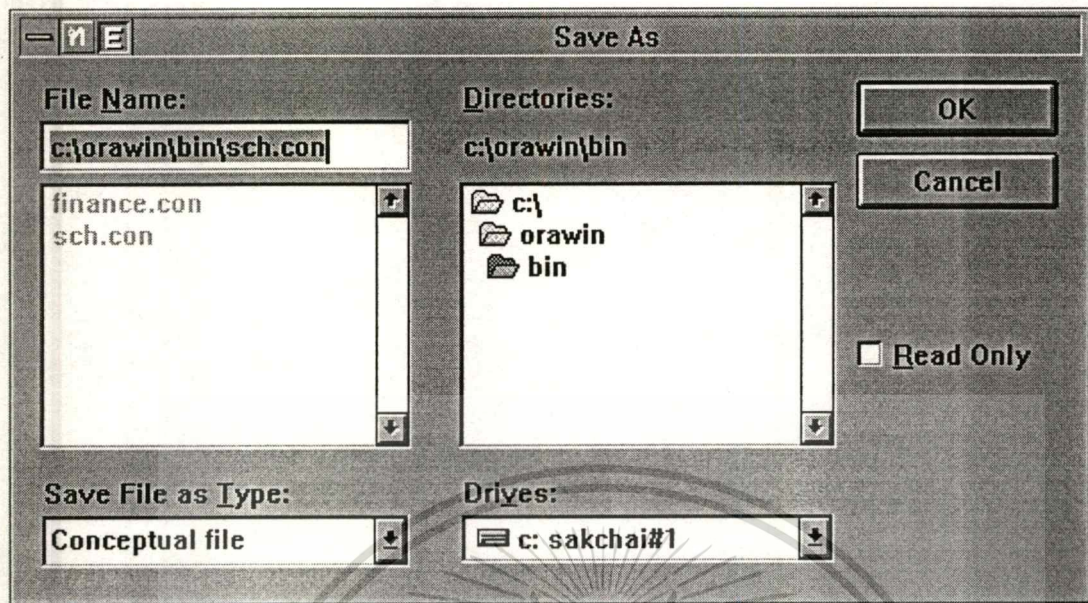


ภาพที่ 47 Open Dialog Box ที่ปรากฏเมื่อผู้ใช้คลิกปุ่ม Open



ภาพที่ 48 หน้าจอหลังจากที่ผู้ใช้เปิดไฟล์ sch.con

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 49 Save Dialog Box ที่ปรากฏขึ้นเมื่อผู้ใช้คลิกปุ่ม Save

SAVE AS

เป็นคำสั่งที่ใช้เรียก Save Dialog Box เช่นเดียวกับปุ่ม Save

CLOSE

เป็นคำสั่งที่ใช้คลิกหน้าจอ พร้อมทั้งปิดไฟล์ที่ถูกเปิดไว้

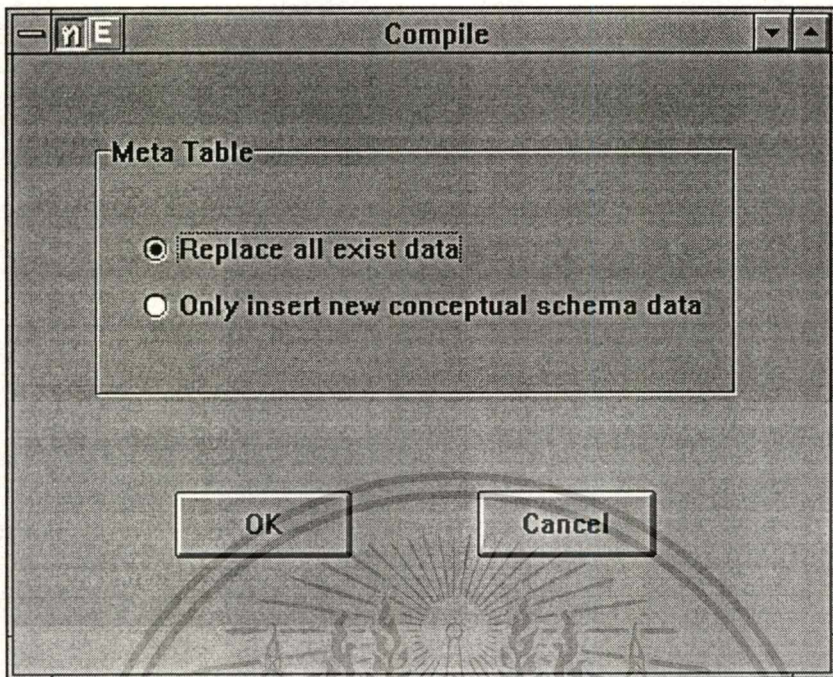
COMPILE

เป็นคำสั่งให้ตรวจสอบ และวิเคราะห์ชุดประโยคคำสั่งภายในไฟล์ที่เปิด

หรือประโยคคำสั่งที่พิมพ์ขึ้นใหม่บนหน้าจอ เพื่อนำข้อมูลที่วิเคราะห์ได้ไปบรรจุไว้ในตารางเมตาสกีมมา และสร้างคำสั่ง SQL DDL สำหรับสร้างตารางฐานข้อมูล และคำสั่งที่ใช้ในการกำหนดกฎข้อบังคับความถูกต้องของข้อมูลชนิดอรรถาธิบาย และคำสั่งที่ใช้ในการกำหนด stored procedure และ database trigger ที่ใช้บังคับกฎชนิดที่ไม่มีคำสั่งแบบอรรถาธิบายไว้รองรับ เมื่อผู้ใช้คลิกปุ่ม Compile จะปรากฏหน้าจอ ดังภาพที่ 50 เพื่อให้ผู้ใช้กำหนดทางเลือกว่าต้องการที่จะลบข้อมูลเดิมที่มีอยู่ภายในตารางเมตาสกีมมาก่อนที่จะบรรจุข้อมูล conceptual schema ใหม่ลงไปหรือไม่ ซึ่งถ้าหากผู้ใช้เลือกการเพิ่มข้อมูลเข้าไปในตารางเมตาสกีมมา โดยไม่ลบข้อมูลก่อน การนำข้อมูลใหม่ อาจจะมีข้อมูลที่ซ้ำกับข้อมูลในส่วนของไพรมารีเอ็ลลียของแต่ละตารางได้ ซึ่งจะทำให้ข้อมูลเหล่านั้นไม่สามารถบรรจุลงในตารางเมตาสกีมมาได้ ดังนั้นจึงขอแนะนำว่าผู้ใช้ควรจะเลือกทางเลือกในการลบข้อมูลเก่าทิ้งเสียก่อน ยกเว้นกรณีที่ผู้ใช้ต้องการเพิ่มเติมข้อมูลลงในแบบจำลองข้อมูลเก่าเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

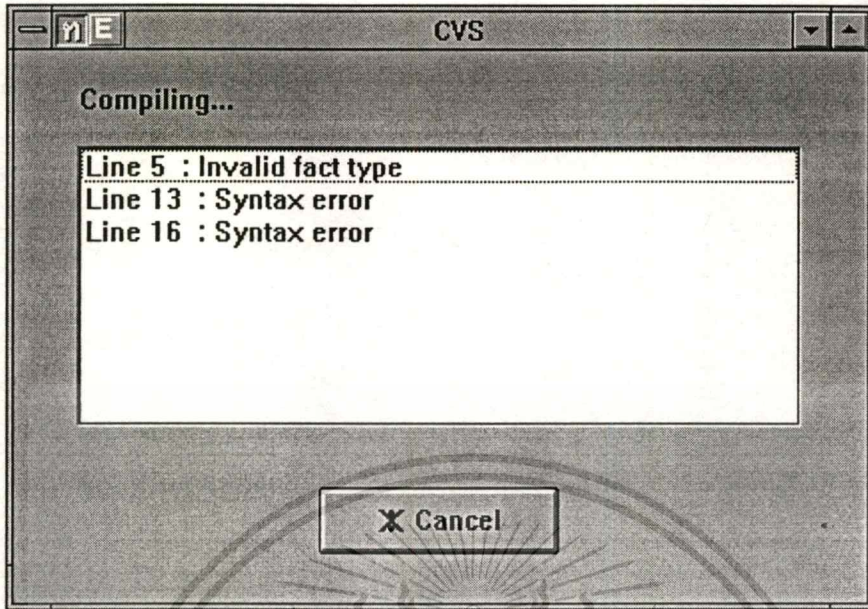


ภาพที่ 50 หน้าจอทางเลือกในการลบข้อมูลในเมตาสกีมมาที่ปรากฏเมื่อผู้ใช้คลิกปุ่ม Compile

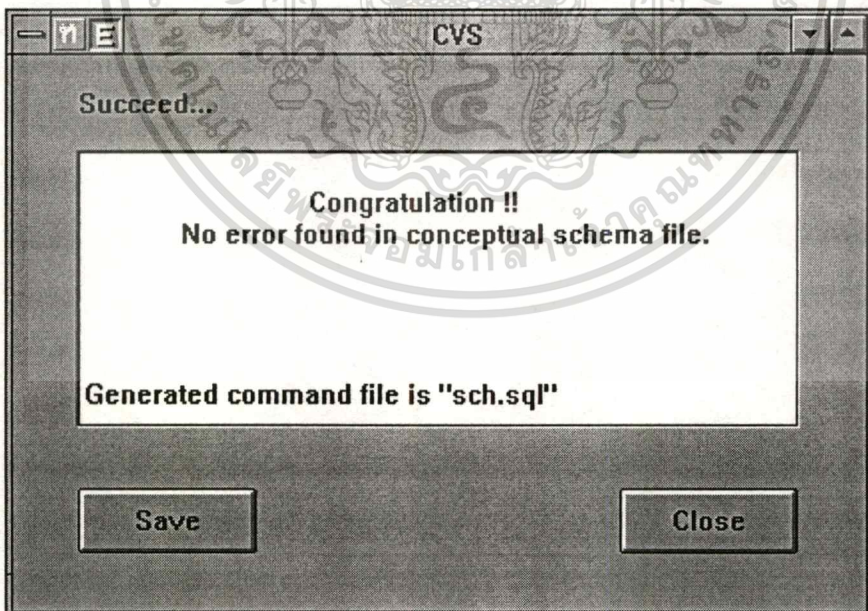
ซึ่งเมื่อผู้ใช้เลือกทางเลือกลบแล้ว ถ้าหากต้องการยกเลิกการทำงานแล้วกลับไปหน้าจอหลัก ให้ผู้ใช้คลิกปุ่ม Cancel แต่ถ้ายังคงต้องการให้โปรแกรมทำงานต่อไปให้คลิกปุ่ม OK ซึ่งการคลิกปุ่ม OK นี้จะปรากฏหน้าจอที่แสดงผลลัพธ์ของการทำงานของโปรแกรกดังภาพที่ 51

หน้าจอภาพที่ 51 นี้ จะแสดงข้อความที่แสดงความผิดพลาดอันเกิดจากการกำหนดประโยคคำสั่งภาษาธรรมชาติไวยากรณ์รูปแบบจำกัดไว้ไม่ถูกต้อง หรือข้อความแสดงความผิดพลาดอื่นๆ อันเกิดจากการทำงานของโปรแกรม ขณะที่โปรแกรมทำงานอยู่นี้หากผู้ใช้ต้องการยกเลิกการทำงานและกลับสู่หน้าจอหลักในทันที กระทำได้โดยการคลิกปุ่ม Cancel แต่ถ้าหากโปรแกรมทำงานจนเสร็จสิ้นโดยไม่พบข้อผิดพลาดใดๆ จะปรากฏหน้าจอภาพที่ 52

หากผู้ใช้ต้องการที่จะอ่านข้อความแสดงความผิดพลาดที่เกิดขึ้น ผู้ใช้ก็สามารถอ่านได้โดยใช้ลูกศรขึ้น-ลง บนแป้นพิมพ์เพื่อเลื่อนดูข้อความก่อนหรือหลังได้ และถ้าผู้ใช้ต้องการเก็บข้อความเหล่านั้นไว้ในไฟล์ ผู้ใช้สามารถทำได้ด้วยการคลิกปุ่ม Save ซึ่งจะปรากฏหน้าจอภาพที่ 53 หรือคลิกปุ่ม Close เมื่อต้องการปิดหน้าจอนี้ เพื่อไปสู่หน้าจอถัดไปซึ่งจะเป็นหน้าจอภาพที่ 54

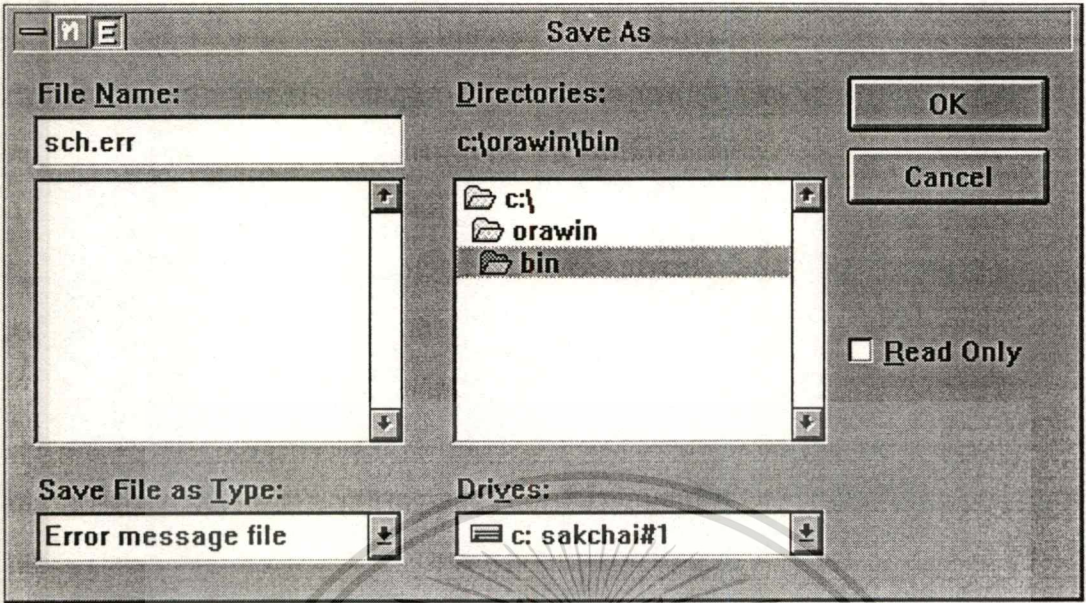


ภาพที่ 51 หน้าจอที่แสดงความผิดพลาดที่พบขณะโปรแกรมทำงาน

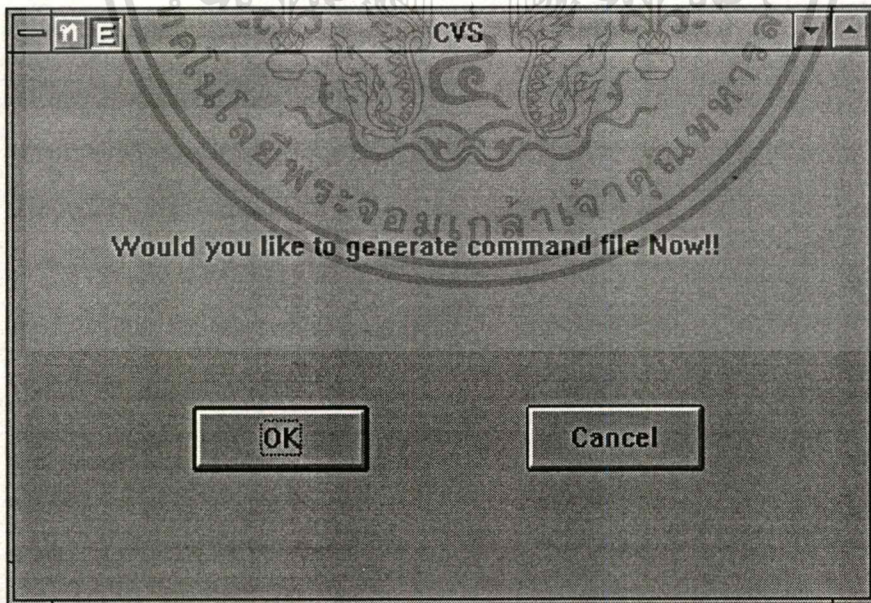


ภาพที่ 52 หน้าจอที่แสดงว่าโปรแกรมทำงานเสร็จสิ้นโดยไม่พบข้อผิดพลาดใดๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 53 หน้าจอที่ปรากฏเมื่อผู้ใช้งานต้องการเก็บข้อความแสดงข้อผิดพลาดลงในไฟล์

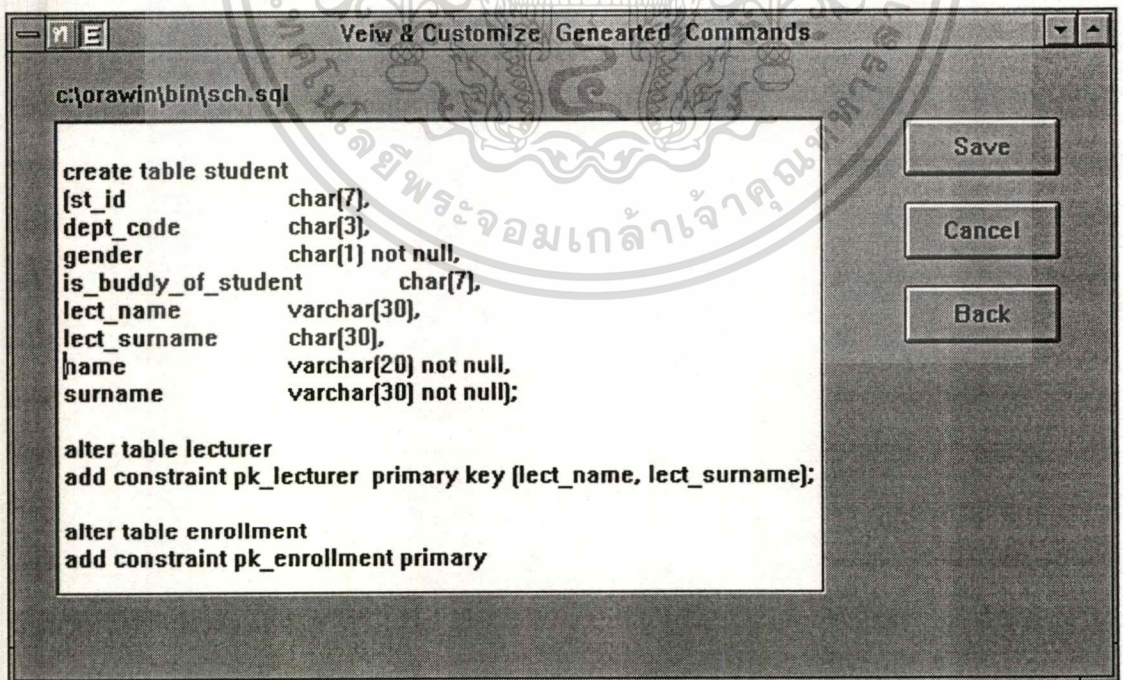


ภาพที่ 54 หน้าจอยืนยันความต้องการสร้างไฟล์คำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในหน้าจอภาพที่ 54 เป็นหน้าจอที่ให้ผู้ใช้นั่นความต้องการในการสร้างไฟล์คำสั่งที่จะนำไปใช้สร้างตารางฐานข้อมูล และกฎข้อบังคับความถูกต้องของข้อมูลต่างๆ ซึ่งถ้าหากผู้ใช้พบว่าเกิดข้อผิดพลาดในการเขียนประโยคคำสั่งภาษาธรรมชาติ หรือในการกำหนดความสัมพันธ์หรือกฎต่างๆ และผู้ใช้ไม่ต้องการที่จะสร้างไฟล์คำสั่งในทันที แต่ต้องการที่จะแก้ไขชุดประโยคคำสั่งภาษาธรรมชาติเสียก่อน ให้ผู้ใช้คลิกปุ่ม Cancel เพื่อกลับสู่หน้าจอหลัก หรือถ้าหากผู้ใช้ต้องการสร้างไฟล์คำสั่งเนื่องจากไม่พบข้อผิดพลาดในการกำหนดแบบจำลองข้อมูลด้วยประโยคภาษาธรรมชาติ หรือข้อผิดพลาดที่พบไม่มีผลมากนัก ให้ผู้ใช้คลิกปุ่ม OK เพื่อไปสู่หน้าจอการทำงานถัดไป ซึ่งจะแสดงไว้ดังภาพที่ 55

ในหน้าจอนี้จะปรากฏผลลัพธ์ของโปรแกรม คือ ชุดคำสั่งที่เกิดจากการนำข้อมูลแบบจำลองข้อมูลระดับแนวคิดที่ปรากฏอยู่ในตารางเมตาสกีมมา มาสร้างเป็นคำสั่งที่ใช้ในการสร้างตารางฐานข้อมูล และกฎข้อบังคับความถูกต้องของข้อมูลชนิดต่างๆ ซึ่งผู้ใช้สามารถที่จะเลื่อนขึ้นเลื่อนลงเพื่อตรวจสอบดูชุดคำสั่งที่โปรแกรมสร้างขึ้น และสามารถแก้ไขได้ เมื่อผู้ใช้ต้องการจะเก็บคำสั่งที่แก้ไขแล้วลงไฟล์เดิมที่โปรแกรมสร้างหรือเก็บลงไฟล์ใหม่ สามารถกระทำได้ด้วยการคลิกปุ่ม Save หรือหากต้องการยกเลิกการแก้ไข ทำได้ด้วยการคลิกปุ่ม Cancel และหากต้องการกลับสู่หน้าจอหลักอีกครั้งหนึ่งให้คลิกปุ่ม Back



ภาพที่ 55 หน้าจอแสดงและแก้ไขชุดคำสั่งที่โปรแกรมสร้างขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกลับมาที่หน้าจอหลัก และผู้ใช้ต้องการจะออกจากโปรแกรม โดยให้โปรแกรมสิ้นสุดการทำงาน ให้คลิกปุ่ม Exit

ตัวอย่างข้อมูลที่ใช้ในการทดสอบ

ขั้นตอนในการทดสอบการทำงานของระบบสร้างกฎบังคับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนลนี้ ได้อาศัยแบบจำลองข้อมูลตัวอย่างที่ออกแบบขึ้นเพื่อใช้ในการทดสอบโดยเฉพาะ แบบจำลองข้อมูลตัวอย่างนี้ได้ถูกออกแบบให้มีความหลากหลายของชนิดความจริงคือ มีชนิดความสัมพันธ์ตั้งแต่ชนิดยูนิอาร์, ไบนารี ไปจนถึงชนิดเอ็นนารี และได้กำหนดกฎข้อบังคับกับความถูกต้องของข้อมูลชนิดต่างๆ ไว้อย่างครบถ้วน เพื่อทดสอบการทำงานของ Schema Analyzer ว่าจะสามารถวิเคราะห์ข้อมูลจากประโยคความจริง และประโยคอ้างอิงที่มีขนาดต่างๆกัน และนำไปบรรจุลงในตารางเมตาสกีมมาได้อย่างถูกต้องหรือไม่ และสามารถวิเคราะห์ประโยคกำหนดกฎข้อบังคับกับความถูกต้องของข้อมูลชนิดต่างๆ ได้ถูกต้องหรือไม่

และเพื่อทดสอบ Schema Compiler ตารางความสัมพันธ์ที่สร้างจากแบบจำลองข้อมูลตัวอย่าง จำเป็นที่จะต้องเกิดจากวิธีการแม่ปความสัมพันธ์ที่แตกต่างกัน โดยการทำงานของ Schema Compiler จะเป็นไปตามหลักการของอัลกอริทึม ONF คือรวมเอาความสัมพันธ์ที่มีคีย์ร่วมกันมาสร้างเป็นหนึ่งตารางความสัมพันธ์ แต่รายละเอียดและขั้นตอนในการดำเนินงานแตกต่างจากอัลกอริทึม ONF ดังที่กล่าวไว้แล้วในบทที่ 7 เนื่องจากข้อมูลของความสัมพันธ์ที่ถูกเก็บไว้ภายในตารางเมตาสกีมมา เป็นความสัมพันธ์ที่มีขนาดสูงสุดคือชนิดไบนารีของ object 2 ชนิดใดๆ เท่านั้น โดย object ทั้งสองชนิดนั้นอาจจะเป็น ชนิดเอนติตี้, ชนิดเลเบล หรือ ชนิดความจริงใดๆ ก็ได้ ซึ่งต่างจากความสัมพันธ์ที่ถูกแสดงโดยแบบจำลองในแอม ที่สามารถพิจารณาเห็นได้ในทันทีว่าเป็นความสัมพันธ์ที่มีขนาดเป็นเท่าใด ดังนั้นอัลกอริทึมในการสร้างตารางความสัมพันธ์จากข้อมูลภายในตารางเมตาสกีมมาจึงมีขั้นตอนที่ซับซ้อนมากกว่าขั้นตอนของอัลกอริทึม ONF อยู่มาก

แบบจำลองข้อมูลตัวอย่างที่ใช้ในการทดสอบเป็นไฟล์ข้อความที่มีเนื้อหาภายในดังที่แสดงไว้ในภาคผนวก ข. แต่ได้ยกบางส่วนมาเป็นตัวอย่างดังนี้คือ

ST_ID char(7);

NAME varchar(20);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SURNAME varchar(30);

LECT_NAME varchar(30);

LECT_SURNAME varchar(30);

SUBJ_CODE char(5);

SUBJ_NAME varchar(50);

.....

.....

Grade(GRADE);

Budget(BUDGET_AMOUNT);

Product(PRODUCT_ID);

Color(COLOR_NAME);

Order(ORDER_NO);

Item(ITEM_NO);

.....

.....

Student enroll_in Subject & Subject is_enrolled_by Student;

Student enroll_in Subject in Semester of Year;

Student enroll_in Subject in Semester of Year take Grade;

Enrollment : Student enroll_in Subject in Semester of Year;

Enrollment has Status;

Semester of Year;

Term : Semester of Year;

Lecturer has Lect_type;

Lecturer earns Salary & Salary of Lecturer;

Lecturer teaches Subject;

Spouse works for Company & Company of Spouse;

Subject has SUBJ_NAME;

.....

.....

uk_stname := one Student has one NAME & one NAME of many Student;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

uk_stsur := one Student has one SURNAME & one SURNAME of many Student;
 uk_gender := one Student has one Gender;
 uk_enr := one Student enroll_in many Subject & one Subject is_enrolled_by many Student;
 uk_enroll := one Student enroll_in Subject in Semester of many Year & one Year of many Student enroll_in Subject in Semester;
 uk_fullname := combination of (NAME, SURNAME) of Student is unique;
 nn_stdept := every Student is_in Dept;
 nn_deptname := every Dept has DEPT_NAME;
 mem_club := one Club has 5 to 100 Student;
 army_camp := MaleStudent trained_at Camp if MaleStudent has ARMY_ID;
 val_grade := GRADE must be in {'A','B+','B','C+','C','D+','D','F','T'};
 val_gender := GENDER must be in ('M','F');
 val_lectype := LECT_TYPE must be in ('ประจำ','พิเศษ');
 val_credit := Subject has Credit in range of [0,5];
 ir_advisor := Student is_advisee_of Lecturer is irreflexive;
 acy_advisor := Student is_advisee_of Lecturer is acyclic;
 asym_advisor := Student is_advisee_of Lecturer is asymmetric;
 need_lectype := set of Lecturer must be has at least 3 Lecturer has Lect_type 'ประจำ';
 sel_lect := every Lecturer play exactly 1 roles of { Lecturer lives_at Address, Lecturer occupies_lodging_at Room};
 lect_office := every Lecturer play at least one roles of { Lecturer is_in Dept, Lecturer works_for Company}
 dual_lect := Lecturer occupies_lodging_at Room iff Lecturer has_int_tel Tel;
 sal_inc := Salary of Lecturer is not decreasingly;

ซึ่งประโยคข้างต้นนี้ ประกอบไปด้วยประโยคคำสั่งที่ใช้กำหนดความสัมพันธ์ และกฎข้อ
 บังคับความถูกต้องของข้อมูลชนิดต่างๆ เมื่อนำมาใช้เป็นอินพุทของโปรแกรมแล้ว ข้อมูลที่เป็นผล
 ลัพท์ของโปรแกรมและปรากฏอยู่ภายในตารางเมตาสกีม่าจะเป็นดังตัวอย่างที่ปรากฏในภาพที่ 56

Database Desktop					
File Edit View Table Record Properties Utilities Window Help					
Table : ROLE.DB					
ROLE	Role	Object	Relation	Name_extend	
6	Club is sponsored by Student	Club	Student is_sponsor_for Club		Student is
7	Club of Student	Club	Student is_member_of Club		Student is
8	Color of Product	Color	Product has Color		key
9	Company is_office_of Lecturer	Company	Lecturer works_for Company	works_for	Lecturer
10	Company of Spouse	Company	Spouse works_in Company	works_in	Lecturer
11	Company research_with Lecturer	Company	Lecturer research_for Company	research_for	Lecturer
12	Credit of Subject	Credit	Subject has Credit		Subject
13	DEPT_NAME has Budget	DEPT_NAME	DEPT_NAME has Budget		-
14	DEPT_NAME of Dept	DEPT_NAME	Dept has DEPT_NAME		Dept
15	Dept has DEPT_NAME	Dept	Dept has DEPT_NAME		-
16	Dept has Dept_HEAD	Dept	Dept has Dept_HEAD		-
17	Dept of Lecturer	Dept	Lecturer is_in Dept		Lecturer
18	Dept of Student	Dept	Student is_in Dept		Student
19	Dept of Subject	Dept	Subject of Dept		Subject
20	Dept_HEAD of Dept	Dept_HEAD	Dept has Dept_HEAD		Dept
21	Enroll make_in Term	Enroll	Enroll make_in Term		key
22	Enrollment has Status	Enrollment	Enrollment has Status		-
23	Enrollment take Grade	Enrollment	Enrollment take Grade		-

Record 23 of 95

ภาพที่ 56 หน้าจอของ Database Desktop ที่แสดงข้อมูลภายในตารางเมตาดัสม้า ROLE

และไฟล์ชุดคำสั่งที่ได้จากการทำงานของโปรแกรม เมื่อพิจารณาในแง่ของการออกแบบฐานข้อมูลโดยเลือกพิจารณาเฉพาะคำสั่งที่ใช้ในการสร้างตารางฐานข้อมูล , primary key และ foreign key แล้วพบว่าตารางฐานข้อมูลที่สร้างได้จากคำสั่งดังกล่าวจะมีลักษณะเป็นดังตารางต่อไปนี้ โดยตารางทั้งหมดอยู่ในรูปของฟิฟท์นอร์มัลฟอร์มตามจุดประสงค์ของโปรแกรม

ตารางที่ 32
DEPT

DEPT_CODE	DEPT_NAME	LECT_NAME	LECT_SURNAME	BUDGET_AMOUNT
CHAR(3)	CHAR(30)	VARCHAR(30)	VARCHAR(30)	NUMBER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 33
SUBJECT

SUBJ_CODE	SUBJ_NAME	CREDIT	DEPT_CODE
CHAR(5)	VARCHAR(50)	NUMBER	CHAR(3)

ตารางที่ 34
STUDENT

ST_ID	ST_NAME	ST_SURNAME	GENDER	DEPT_CODE
CHAR(7)	VARCHAR(20)	VARCHAR(30)	CHAR(1)	CHAR(3)

LECT_NAME	LECT_SURNAME	IS_BUDDY_OF_STUDENT
VARCHAR(30)	VARCHAR(30)	CHAR(7)

ตารางที่ 35
ENROLLMENT

ST_ID	SUBJ_CODE	SEMESTER	YEAR	GRADE	STATUS
CHAR(7)	CHAR(5)	NUMBER	NUMBER	CHAR(2)	CHAR(1)

ตารางที่ 36
ORDER_WITH_ITEM

ORDER_NO	ITEM_NO	PRODUCT_ID	QTY
NUMBER	NUMBER	CHAR(9)	NUMBER

ตารางที่ 37
LECTURER

LECT_NAME	LECT_SURNAME	LECT_TYPE	DEPT_CODE.	SAL
VARCHAR(30)	VARCHAR(30)	VARCHAR(10)	CHAR(3)	NUMBER
WORKS_FOR_COMPANY_NAME		RESEARCH_WITH_COMPANY_NAME		TEL_NO
VARCHAR(30)		VARCHAR(30)		CHAR(7)
ROOM_NO	ADDRESS	SP_NAME	SP_SURNAME	
CHAR(4)	VARCHAR(100)	VARCHAR(30)	VARCHAR(30)	
WORKS_AT_COMPANY_NAME				
VARCHAR(30)				

ตารางที่ 38
PRODUCT_HAS_COLOR

PRODUCT_ID	COLOR_NAME	QTY
CHAR(9)	VARCHAR(20)	NUMBER

ตารางที่ 39
STUDENT_IS_MEMBER_OF_CLUB

ST_ID	CLUB_NAME
CHAR(7)	VARCHAR(20)

ตารางที่ 40

STUDENT_IS_SPONSOR_FOR_CLUB

ST_ID	CLUB
CHAR(7)	VARCHAR(20)

ตารางที่ 41

MALESTUDENT

ST_ID	ARMY_ID
CHAR(7)	CHAR(10)

ตารางที่ 42

MALESTUDENT_TRAINED_AT_CAMP

ST_ID	CAMP_NAME
CHAR(7)	VARCHAR(40)

ตารางที่ 43

STUDENT_IS_STAFF_OF_SUBJECT_IN

ST_ID	SUBJ_CODE	SEMESTER	YEAR
CHAR(7)	CHAR(5)	NUMBER	NUMBER

และเมื่อนำมาใช้สร้างตารางฐานข้อมูล, กฎชนิดอรรถาธิบาย, stored procedure และ database trigger บนระบบจัดการฐานข้อมูลออราเคิลแล้วพบว่า สำหรับตารางฐานข้อมูลที่มีกฎข้อบังคับความถูกต้องของข้อมูลที่ยังบังคับกับกลุ่มของข้อมูล เช่น ตาราง STUDENT_IS_MEMBER_OF_CLUB นั้นจะไม่สามารถบรรจุข้อมูลลงไปได้ เนื่องจาก stored procedure ที่บังคับกฎข้อบังคับความถูกต้องของข้อมูลชนิด occurrence frequency กับตารางนี้ บังคับว่าจำนวนของ row ข้อมูลของชมรมใดชมรมหนึ่งจะต้องปรากฏอยู่ในตารางอย่างน้อยที่สุด 5 row ตามที่กฎระบุไว้ แต่เนื่องจาก database trigger ที่เรียกใช้ stored procedure นั้นมีคุณสมบัติตรวจสอบการทำงานของผลลัพธ์ของคำสั่งทีละคำสั่ง ดังนั้นเมื่อใช้คำสั่ง INSERT เพื่อป้อนข้อมูล 1 row เข้าสู่ตารางดังกล่าว จึงเป็นการละเมิดกฎในทันที ส่วนตารางฐานข้อมูลอื่นๆ ที่ถูกบังคับด้วยกฎในลักษณะเดียวกันนี้ก็เช่นเดียวกัน เช่น ตาราง LECTURER ซึ่งมีกฎข้อบังคับความถูกต้องของข้อมูลชนิด mandatory entity บังคับให้ข้อมูลของอาจารย์ต้องปรากฏอาจารย์ที่มีชนิดเป็นอาจารย์ประจำอยู่อย่างน้อยที่สุด 3 คน

ส่วนตารางฐานข้อมูลอื่นๆ นั้นจะไม่มีปัญหาในการนำเข้าข้อมูล ถ้าหากข้อมูลที่บรรจุลงสู่ตารางไม่ละเมิดต่อกฎที่กำหนดไว้ ซึ่งถ้าหากข้อมูลที่นำเข้าสู่ตารางฐานข้อมูลนั้นละเมิดต่อกฎที่วางไว้จะปรากฏข้อความแสดงความผิดพลาดของคำสั่ง INSERT และเมื่อทดลองใช้คำสั่ง UPDATE และ DELETE กับข้อมูลภายในตารางฐานข้อมูลต่างๆ ก็ให้ผลเช่นเดียวกันคือ จะยอมรับการทำงานของคำสั่ง ถ้าหากผลลัพธ์ของคำสั่งนั้นไม่ละเมิดต่อกฎข้อบังคับความถูกต้องของข้อมูลที่กำหนดไว้ แต่ละยกเลิกการทำงานของคำสั่ง และแสดงข้อความแสดงความผิดพลาดของคำสั่งต่อผู้ใช้ ถ้าหากคำสั่งนั้นก่อให้เกิดผลลัพธ์ที่ละเมิดต่อกฎข้อบังคับความถูกต้องของข้อมูล

บทที่ 9

คุณสมบัติในการรักษาความถูกต้องของข้อมูลที่ระบบจัดการฐานข้อมูล เชิงพาณิชย์พึงมี

ปัญหาในการบังคับความถูกต้องของข้อมูลบนระบบจัดการฐานข้อมูลในปัจจุบัน

จากการศึกษาการใช้คุณสมบัติในการบังคับความถูกต้องของข้อมูลของระบบจัดการฐานข้อมูลเชิงพาณิชย์ในปัจจุบันนั้น ถึงแม้จะพบว่าระบบจัดการฐานข้อมูลได้พัฒนาคุณสมบัติในการดูแลรักษาความถูกต้องของข้อมูลไปได้มากก็ตาม แต่การกำหนดกฎข้อบังคับความถูกต้องของข้อมูลเพื่อบังคับความถูกต้องของข้อมูลให้เป็นไปตามที่ต้องการอย่างครบถ้วนนั้น ยังเป็นเรื่องลำบาก เนื่องจากระบบจัดการฐานข้อมูลส่วนมากจะมีกฎข้อบังคับความถูกต้องของข้อมูลชนิดอรรถธิบายไม่มากนัก ซึ่งส่วนใหญ่จะเป็นกฎข้อบังคับความถูกต้องของข้อมูลชนิด primary key, unique key, not null, referential integrity และชนิด entity type ดังนั้นโดยมากแล้วผู้ใช้จำเป็นต้องเขียน stored procedure และ database trigger เพื่อบังคับกฎข้อบังคับความถูกต้องของข้อมูลชนิดที่ระบบจัดการฐานข้อมูลไม่มีคำสั่งเฉพาะมารองรับ ซึ่งเป็นในกรณีที่ระบบจัดการฐานข้อมูลมีคุณสมบัติในการกำหนด database trigger และ stored procedure ไม่เช่นนั้นผู้ใช้จะต้องเขียนโปรแกรมประยุกต์ให้สามารถจัดการการรักษาความถูกต้องของข้อมูลเอง ซึ่งข้อเสียจากการรักษาความถูกต้องของข้อมูลโดยใช้โปรแกรมประยุกต์นั้นมีหลายประการดังที่กล่าวไว้แล้วในบทที่ 3 แต่การใช้ database trigger และ stored procedure สามารถลดปัญหาในการกำหนดกฎข้อบังคับความถูกต้องของข้อมูลไปได้ส่วนหนึ่งเท่านั้น ซึ่งก็ยังคงมีข้อเสียอยู่ นั่นคือ

1. การเขียน stored procedure และ database trigger ทำได้ลำบาก

ผู้ใช้จำเป็นต้องศึกษาโครงสร้าง, วิธีการเขียน และวิธีการบังคับใช้ของ database trigger เหล่านั้น และการเขียนกฎข้อบังคับความถูกต้องของข้อมูลในลักษณะ procedural เช่นนี้ ย่อมสร้างความเข้าใจให้แก่ผู้ที่มีหน้าที่ต้องสานงานต่อได้ลำบาก ต่างจากการกำหนดกฎข้อบังคับความถูกต้องของข้อมูลในรูปแบบที่เป็นอรรถธิบาย เนื่องจากกฎข้อบังคับความถูกต้องของข้อมูลที่มีคำสั่งเฉพาะรองรับนั้นผู้ใช้สามารถทราบความหมายและวิธีการบังคับของกฎได้โดยง่าย จากคู่มือการใช้ระบบจัดการฐานข้อมูล

นอกจากนี้ยังได้ศึกษาพบว่า เมื่อต้องการกำหนดกฎข้อบังคับความถูกต้องของข้อมูลเพื่อใช้บังคับความถูกต้องของข้อมูลตามที่วิเคราะห์ได้จากระบบงาน และเพื่อบังคับความถูกต้องของข้อมูลตามกฎข้อบังคับความถูกต้องของข้อมูลชนิดต่างๆ ที่ได้กำหนดไว้ ระบบจัดการฐานข้อมูลในปัจจุบัน ต่างก็ไม่สามารถรองรับความจำเป็นในการกำหนดกฎเหล่านั้นได้ทั้งหมด ทั้งนี้เนื่องจากคุณสมบัติและข้อจำกัดของ database trigger และ stored procedure ของโปรแกรมระบบจัดการฐานข้อมูลแต่ละโปรแกรม ดังจะได้แสดงไว้ต่อไป

1.1 การใช้ database trigger บนระบบจัดการฐานข้อมูลแบบรีเลชันแนลในปัจจุบัน ไม่สามารถรองรับการละเมิดกฎข้อบังคับความถูกต้องของข้อมูลภายใน transaction ได้

database trigger ทั้งหมดจะทำการตรวจสอบความถูกต้องของข้อมูล หรือความถูกต้องในการปฏิบัติการต่อข้อมูล ก็ต่อเมื่อเกิดเหตุการณ์ใดๆ ต่อข้อมูลในฐานข้อมูลที่ตรงกับเหตุการณ์ที่ระบุไว้ใน database trigger ซึ่งเหตุการณ์เหล่านั้นมักจะเป็นเหตุการณ์ที่บ่งบอกถึงการจัดการข้อมูลของผู้ใช้ เช่น เมื่อมีผู้ใช้ใช้คำสั่ง UPDATE กับข้อมูลในคอลัมน์ SALARY ของตารางฐานข้อมูล LECTURER หรือ เมื่อมีการนำข้อมูลใส่ลงในตารางฐานข้อมูล STUDENT_IS_MEMBER_OF_CLUB เป็นต้น แต่จากความเป็นจริงและคุณสมบัติของกฎข้อบังคับความถูกต้องของข้อมูล เมื่อพิจารณาพร้อมกับ transaction แล้ว กฎข้อบังคับความถูกต้องของข้อมูลจะต้องสามารถละเมิดได้ภายใน transaction ใดๆ แต่เมื่อจบ transaction แล้ว ฐานข้อมูลจะต้องอยู่ในสถานะที่ถูกต้องเสมอ แต่จากคุณสมบัติการทำงานของ database trigger ที่เป็นเช่นนี้ทำให้การตรวจสอบความถูกต้องของข้อมูลกระทำกับทุกๆ คำสั่งภายใน transaction (ในโปรแกรมระบบจัดการฐานข้อมูลบางโปรแกรมเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น ระบบจัดการฐานข้อมูลออรากเคิล สามารถที่จะตรวจสอบผลลัพธ์ของคำสั่งจัดการข้อมูลที่ละ row ได้ในขณะที่คำสั่งนั้นยังประมวลผลไม่สิ้นสุดลง) เมื่อพบว่าคำสั่งใดละเมิดกฎที่ได้กำหนดไว้ คำสั่งนั้นก็จะถูกยกเลิกไปทันที ดังนั้นจึงไม่มีทางที่ผู้ใช้จะละเมิดกฎข้อบังคับความถูกต้องของข้อมูลภายใน transaction ตามคุณสมบัติในการรักษาความถูกต้องของข้อมูลได้

1.2 การกำหนด database trigger จำเป็นต้องวิเคราะห์เหตุการณ์ต่างๆ ที่เกิดขึ้นกับฐานข้อมูล แล้วเกิดผลกระทบต่อความถูกต้องของข้อมูล

ความยุ่งยากที่เกิดจากการที่กฎข้อบังคับความถูกต้องของข้อมูลชนิดต่างๆ ที่ต้องกำหนด ข้อบังคับความถูกต้องของข้อมูลแต่ละข้อไว้ใน database trigger หลายชนิด และ database trigger แต่ละโปรแกรมจะซับซ้อน เนื่องจากต้องประกอบไปด้วยชุดคำสั่งที่ใช้รักษากฎข้อบังคับความถูกต้องของข้อมูลหลายข้อ

database trigger ถูกแบ่งชนิดตามประเภทของเหตุการณ์ที่กำหนดให้มันทำงาน เป็นต้นว่า BEFORE INSERT trigger, AFTER DELETE trigger หรือ AFTER INSERT OR UPDATE trigger และยังกำหนดแยกออกเป็นชนิด row trigger และ statement trigger ซึ่ง database trigger แต่ละชนิดจะถูกกำหนดให้บังคับกับตารางฐานข้อมูลใดๆ ได้เพียงตารางละ 1 ชนิดเท่านั้น ซึ่งเป็นการจำกัดที่ทำให้การกำหนด database trigger ก่อนข้างจะยุ่งยาก เนื่องจากจะต้องวิเคราะห์ว่ากฎข้อบังคับความถูกต้องของข้อมูลแต่ละข้อที่ต้องการจะใช้บังคับนั้น จะต้องนำไปกำหนดไว้ภายใน database trigger ชนิดใดบ้าง โดยกฎข้อบังคับความถูกต้องของข้อมูล 1 ข้ออาจจะต้องกำหนดไว้ใน database trigger หลายชนิด เช่น ตารางฐานข้อมูลดังตารางที่ 44 หากเราต้องการกำหนดว่าผลรวมเงินเดือนของลูกจ้างช่วยงานวิจัยของงานวิจัยแต่ละโครงการ จะต้องไม่เกินงบประมาณในการจ้างของแต่ละโครงการนั้นๆ ซึ่งจะพิจารณาจากกฎดังกล่าวได้ว่า ต้องการตรวจสอบไม่ให้เกิดการใช้เงินในการจ้างผู้ร่วมวิจัยสำหรับงานวิจัยแต่ละโครงการเกินจำนวนเงินงบประมาณที่มี ดังนั้นเหตุการณ์ที่เกิดขึ้นกับฐานข้อมูลที่มีผลกระทบต่อจำนวนเงินเดือนรวมนี้คือ การจ้างลูกจ้างใหม่ และการเพิ่มเงินเดือน ซึ่งเกิดจากการ INSERT ข้อมูลใหม่ลงในตารางฐานข้อมูล EMPLOYEE และ การ UPDATE คอลัมน์ SALARY ของตาราง EMPLOYEE เช่นเดียวกัน ดังนั้นจำเป็นจะต้องกำหนดกฎข้อบังคับความถูกต้องของข้อมูลข้อนี้ลงไปใน database trigger ทั้ง 2 ชนิด (กรณีที่ไม่ได้กำหนด database trigger 2 ชนิดนี้ไว้รวมกัน) คือ AFTER INSERT และ AFTER UPDATE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 44

EMPLOYEE	PROJECT	POSITION	SALARY
Tom Hang	WS101	project leader	70,000
Alan Miller	WS101	researcher	38,000
Don King	WS101	researcher	34,500
Dolly Parton	EC207	project header	66,000
Tina Turner	EC107	research	43,000

3. การกำหนดกฎข้อบังคับความถูกต้องของข้อมูลเป็นไปอย่างสับสนและ ซ้ำซ้อน ซึ่งความซ้ำซ้อนของการเขียนชุดคำสั่งอาจทำให้เกิดข้อผิดพลาดได้

เนื่องจากเหตุผลในข้อ 2. ที่ตารางฐานข้อมูลหนึ่งๆ สามารถกำหนด database trigger ชนิดหนึ่งชนิดใดได้เพียง 1 ชนิดเท่านั้น ทำให้ผู้ใช้อาจจะเกิดความสับสนและหากผู้ใช้ไม่ได้กำหนดเนื้อหาของกฎข้อบังคับความถูกต้องของข้อมูลแยกกันเอาไว้ใน stored procedure แล้ว และหากระบบงานมีความจำเป็นที่จะต้องกำหนด database trigger แยกชนิดกัน เช่น หากต้องการกำหนด database trigger ให้ตรวจสอบเมื่อมีความเปลี่ยนแปลงเกิดขึ้นต่อคอลัมน์ SALARY ในตารางฐานข้อมูล EMPLOYEE จะต้องกำหนด database trigger ชนิด AFTER INSERT OR DELETE OR UPDATE สำหรับกฎข้อนั้น แต่หากในระบบงานเดียวกันนี้ มีความจำเป็นในการกำหนด database trigger ให้แยกชนิดกัน เช่น กฎข้อบังคับความถูกต้องของข้อมูลที่ตรวจสอบเฉพาะเมื่อมีการลบข้อมูลของลูกจ้างออกจากตารางฐานข้อมูล EMPLOYEE จึงต้องกำหนด database trigger เป็นชนิด AFTER DELETE นั้น ทำให้กฎข้อบังคับความถูกต้องของข้อมูลที่ตรวจสอบค่าของคอลัมน์ SALARY ต้องกำหนดแยกเป็น AFTER INSERT OR UPDATE และ AFTER DELETE และคำสั่งภายใน database trigger ทั้งสองชนิดจึงมีบางส่วนที่ซ้ำซ้อนกัน และบางส่วนของที่แตกต่างกัน ซึ่งส่วนที่ซ้ำซ้อนกันนั้นนอกจากจะเปลืองเนื้อที่บนแหล่งเก็บข้อมูลแล้ว อาจจะเป็นไปได้ที่ผู้ใช้เขียนชุดคำสั่งผิดพลาดทำให้การควบคุมความถูกต้องของข้อมูลผิดพลาดไปได้

คุณสมบัติที่จำเป็นในการรักษาความถูกต้องของข้อมูล

ระบบจัดการฐานข้อมูล จำเป็นที่จะต้องมียุคสมบัติในการบังคับความถูกต้องของข้อมูล เพื่อป้องกันปัญหาที่สืบเนื่องมาจากการรักษาความถูกต้องของข้อมูลด้วยโปรแกรมประยุกต์ แต่ถึงแม้ว่าโปรแกรมระบบจัดการฐานข้อมูลจะพัฒนาคุณสมบัติการบังคับความถูกต้องของข้อมูลให้สามารถกำหนดได้ทั้งชนิดที่เป็นอรรถาธิบาย และชนิดที่เป็น procedural ที่ผู้ใช้สามารถกำหนดคำสั่ง, เงื่อนไข และควบคุมขั้นตอนการดำเนินการได้อย่างตามใจชอบ ทำให้สามารถบังคับกฎข้อบังคับความถูกต้องของข้อมูลได้อย่างหลากหลายก็ตาม แต่จากการศึกษาและการใช้โปรแกรมระบบจัดการฐานข้อมูลเชิงพาณิชย์ปัจจุบันในการรักษาความถูกต้องของข้อมูล ทำให้ได้ตระหนักถึงปัญหาที่เกิดจากการขาดคุณสมบัติในการรักษาข้อมูลของโปรแกรมระบบจัดการฐานข้อมูลเชิงพาณิชย์ในปัจจุบัน จึงขอรวบรวมและแสดงคุณสมบัติที่พบว่าจำเป็นต่อการบังคับความถูกต้องของข้อมูลของระบบจัดการฐานข้อมูลดังต่อไปนี้

1. กฎข้อบังคับความถูกต้องของข้อมูลต้องสามารถที่จะละเมิดได้ภายใน transaction ใดๆ

เนื่องจาก transaction คือ ลำดับ ของกิจกรรมที่ก่อให้เกิดการเปลี่ยนแปลงของข้อมูลภายในฐานข้อมูล [Codd E.F.] ที่ผลลัพธ์ของ transaction จะต้องทำให้ฐานข้อมูลอยู่ในสถานะที่ถูกต้อง ดังนั้นถึงแม้ว่าผลลัพธ์ของคำสั่งใดๆ ใน transaction จะละเมิดกฎข้อบังคับความถูกต้องของข้อมูลที่กำหนดไว้ก็ตาม แต่ถ้าหากผลลัพธ์จากคำสั่งอื่นทำให้การละเมิดกฎข้อบังคับความถูกต้องของข้อมูลนั้นๆหมดสิ้นไป ซึ่งทำให้เมื่อจบ transaction ข้อมูลภายในฐานข้อมูลยังคงมีความถูกต้องตามกฎข้อบังคับความถูกต้องของข้อมูลที่ได้วางไว้ ระบบจัดการฐานข้อมูลจึงควรที่จะยินยอมให้การทำงานของคำสั่งที่ก่อให้เกิดการละเมิดกฎข้อบังคับความถูกต้องของข้อมูลนั้นมีผลต่อข้อมูลได้

2. การบังคับความถูกต้องของข้อมูลควรมีคุณสมบัติดังต่อไปนี้

จากการศึกษาลักษณะและขั้นตอนการทำงานรวมทั้งกฎระเบียบต่างๆ ในระบบงานหลายๆ ระบบ สามารถสรุปลักษณะของการบังคับกฎข้อบังคับความถูกต้องของข้อมูล (Integrity constraint enforcement) ได้ดังนี้คือ

2.1 กฎข้อบังคับความถูกต้องของข้อมูลที่ทำงานเมื่อเกิดเหตุการณ์ใดๆ ขึ้น ต่อข้อมูลภายในฐานข้อมูล (trigger by data access action)

2.2 กฎข้อบังคับความถูกต้องของข้อมูลที่ทำงานตามวันเวลาที่ตั้งไว้ (trigger by clock)

โดยกฎข้อบังคับความถูกต้องของข้อมูลทั้งสองชนิดข้างต้นนี้ยังแบ่งออกตามช่วงระยะเวลาที่มีอำนาจในการควบคุมความถูกต้องของข้อมูลได้เป็น

- กฎข้อบังคับความถูกต้องของข้อมูล ที่มีอำนาจบังคับต่อข้อมูลที่มีอยู่ทั้งหมดตลอดเวลา เช่น primary key, uniqueness, mandatory role, entity type, referential integrity constraint ฯลฯ

- กฎข้อบังคับความถูกต้องของข้อมูล ที่มีอำนาจครอบคลุมต่อข้อมูลเฉพาะในช่วงเวลาที่กำหนดไว้ เช่น ในช่วงเทศกาลลดราคาสินค้า ระหว่างวันที่ 10 - 14 กุมภาพันธ์ 2540 สินค้าทุกชนิดจะจำหน่ายในราคาที่ลดลงจากราคาปกติ 15%

- กฎข้อบังคับความถูกต้องของข้อมูลที่มีอำนาจครอบคลุมการจัดการข้อมูลเฉพาะข้อมูลในส่วนที่ถูก INSERT เข้าสู่ฐานข้อมูล ในช่วงที่เวลาที่กฎข้อบังคับความถูกต้องของข้อมูลข้อนั้นกำลังถูกบังคับใช้อยู่ เช่น นัก ศึกษาที่เริ่มเข้าศึกษาในปีการศึกษาใด จะถูกบังคับด้วยกฎระเบียบที่กำหนดใช้ในปีการศึกษานั้นๆ เท่านั้น กฎระเบียบที่ดั่งขึ้นใหม่ในภายหลังไม่มีผลกระทบต่อนักศึกษาที่เข้ารับการศึกษาก่อนหน้านั้น

เนื่องจากความถูกต้องและความสมบูรณ์ของข้อมูลมีผลกระทบโดยตรงต่อการทำงานทั้งในระดับปฏิบัติการและระดับบริหาร ดังนั้นกฎข้อบังคับความถูกต้องของข้อมูลไม่ควรที่จะถูกละเมิด เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ ไม่ว่าจะเป็นการกระทำนั้นจะกระทำผ่านโปรแกรมแอปพลิเคชันหรือกระทำโดยตรงต่อข้อมูลในฐานข้อมูลด้วยคำสั่ง SQL ก็ตาม ยกเว้นแต่ในกรณีที่ผู้ใช้ผู้นั้นได้รับการยกเว้นให้สามารถละเมิดกฎข้อนั้นๆ ได้ ซึ่งคุณสมบัติที่กล่าวมานี้มีความจำเป็นต่อระบบงานโดยทั่วๆ ไปอยู่มาก แต่กลับไม่มีซอฟต์แวร์ระบบจัดการฐานข้อมูลใดที่สามารถรองรับความต้องการนี้ได้ ซึ่งคงต้องอาศัยการพัฒนาและการค้นคว้าเรื่องการดูแลความถูกต้องของข้อมูลต่อไป เพื่อส่งเสริมให้ระบบช่วยออกแบบฐานข้อมูลและระบบจัดการฐานข้อมูลได้รับการพัฒนาให้มีคุณสมบัติสามารถรองรับความต้องการและความจำเป็นในการดูแลทั้งความถูกต้องและความปลอดภัยของข้อมูลได้เพิ่มขึ้น



๑๖

บทที่ 10

สรุปผลและแนวทางการวิจัย

สรุปผลงานวิจัย

ระบบสร้างกฎบังคับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนลนี้ จะเป็นประโยชน์อย่างมากแก่หน่วยงานที่ต้องการจะพัฒนาระบบฐานข้อมูลขึ้นเพื่อเพิ่มประสิทธิภาพในการทำงานและลดโอกาสที่จะเกิดปัญหาข้อมูลผิดพลาด ซึ่งจะส่งผลกระทบต่อการบริหารงานและความน่าเชื่อถือของหน่วยงาน โดยเฉพาะอย่างยิ่งกับหน่วยงานที่ขาดแคลนบุคลากรที่มีความรู้ความเชี่ยวชาญในการออกแบบและพัฒนาระบบฐานข้อมูล เพราะวาระบบอัตโนมัติมีคำสั่งที่ทำให้ผู้ใช้สามารถกำหนดสิ่งต่างๆ ที่สนใจในระบบงานและความสัมพันธ์ของสิ่งเหล่านั้นได้ในรูปแบบของภาษาธรรมชาติซึ่งเป็นรูปแบบที่ง่ายต่อการทำความเข้าใจ อีกทั้งยังมีโครงสร้างของภาษาธรรมชาติไวยากรณ์รูปแบบจำกัดที่ใช้ในการกำหนดกฎข้อบังคับความถูกต้องของข้อมูลที่แสดงความหมายของกฎได้อย่างชัดเจน การกำหนดความสัมพันธ์และกฎข้อบังคับความถูกต้องของข้อมูลชนิดต่างๆ ด้วยภาษาธรรมชาติไวยากรณ์รูปแบบจำกัด จะเป็นเสมือนเครื่องมือสื่อสารระหว่างผู้ออกแบบฐานข้อมูลและผู้ปฏิบัติงานจริง ที่ช่วยให้การตรวจสอบความถูกต้องของแบบจำลองข้อมูลเป็นไปโดยสะดวกและมีความเข้าใจที่ถูกต้องตรงกันทั้งสองฝ่าย

นอกจากจะเป็นเครื่องมือในการออกแบบแบบจำลองข้อมูลแล้ว ระบบอัตโนมัติสำหรับบังคับความถูกต้องของข้อมูลจะสร้างชุดคำสั่งที่ใช้ในการสร้างตารางฐานข้อมูลและกฎข้อบังคับความถูกต้องของข้อมูลชนิดต่างๆ ตามรีเลชันแนลสกีม่าและกฎที่วิเคราะห์ได้จากชุดประโยคภาษาธรรมชาติโดยอัตโนมัติให้แก่ผู้ใช้ ทำให้ผู้ใช้ไม่จำเป็นต้องเขียนประโยคคำสั่ง SQL DDL หรือคำสั่งที่ใช้ในการกำหนดกฎข้อบังคับความถูกต้องของข้อมูลชนิดอรรถธิบาย หรือเขียนโปรแกรมใน database trigger เอง จึงช่วยให้การพัฒนาระบบฐานข้อมูลเป็นไปโดยรวดเร็ว และลดความผิดพลาดในการเขียนโปรแกรมคำสั่งใน database trigger ที่อาจจะส่งผลกระทบต่อระบบจัดการฐานข้อมูลไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถป้องกันการละเมิดกฎข้อบังคับความถูกต้องของข้อมูลของระบบงาน และอาจจะทำให้การนำเข้าหรือแก้ไขข้อมูลที่ถูกต้องไม่สามารถที่จะกระทำได้อีกเป็นไปได้อีก

ประโยชน์ของการออกแบบระบบฐานข้อมูลด้วยระบบอัตโนมัติสำหรับบังคับความถูกต้องของข้อมูลบนฐานข้อมูลแบบรีเลชันแนลนี้ยังมีประโยชน์สำคัญอีกประการหนึ่งคือ เมื่อเกิดการเปลี่ยนแปลงของระบบงานซึ่งจำเป็นจะต้องแก้ไขระบบฐานข้อมูลเดิมที่มีอยู่ ถึงแม้ว่าการแก้ไขระบบฐานข้อมูลนี้จะไม่ได้กระทำโดยผู้ออกแบบฐานข้อมูลคนเดิม ผู้ออกแบบฐานข้อมูลคนใหม่สามารถที่จะศึกษาความสัมพันธ์ของสิ่งต่างๆ ภายในระบบงาน และกฎต่างๆ จากชุดประโยคภาษาธรรมชาติเดิมที่ใช้ในการออกแบบคราวก่อน ได้โดยไม่ต้องเสียเวลาวิเคราะห์ข้อมูลและความสัมพันธ์ทั้งหมดใหม่ ซึ่งการวิเคราะห์ข้อมูลและความสัมพันธ์ของข้อมูลนับว่าเป็นขั้นตอนที่ใช้เวลามากที่สุดในการออกแบบและพัฒนาระบบฐานข้อมูลที่ดี และสามารถแก้ไขชุดประโยคภาษาธรรมชาติเดิมเฉพาะส่วนที่ต้องการแก้ไขหรือเพิ่มเติมได้ง่าย ดังนั้นระบบสร้างกฎบังคับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนล นอกจากจะอำนวยความสะดวกและความปลอดภัยในการออกแบบระบบฐานข้อมูลในปัจจุบันแล้ว ยังเอื้อประโยชน์ในการแก้ไขปรับปรุงระบบฐานข้อมูลในภายหน้าอีกด้วย

ผลที่ได้จากการศึกษาชนิดของกฎข้อบังคับความถูกต้องของข้อมูลและวิธีการบังคับความถูกต้องของข้อมูล ทั้งจากการศึกษางานวิจัยอื่นๆ และจากประสบการณ์และปัญหาในการออกแบบและพัฒนาระบบฐานข้อมูลจริง และปัญหาในการพัฒนาระบบสร้างกฎบังคับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนลนี้สำหรับใช้กับโปรแกรมระบบจัดการฐานข้อมูลแบบรีเลชันแนลเชิงพาณิชย์ในปัจจุบัน ทำให้งานวิจัยนี้สามารถสรุปปัญหาการบังคับความถูกต้องของข้อมูลบนระบบจัดการฐานข้อมูลเชิงพาณิชย์ และคุณสมบัติต่างๆ ที่จำเป็นต้องการบังคับความถูกต้องของข้อมูลได้ดังที่แสดงไว้ในบทที่ 9 ซึ่งผู้ที่สนใจอาจจะนำข้อสรุปดังกล่าวมาใช้ในการวิจัยและพัฒนาระบบจัดการฐานข้อมูลให้มีคุณสมบัติในการรักษาความถูกต้องของข้อมูลให้สมบูรณ์ยิ่งขึ้น หรือนำมาใช้เป็นเกณฑ์ในการตัดสินใจเลือกใช้ระบบจัดการฐานข้อมูลเชิงพาณิชย์ก็ได้เช่นเดียวกัน

ปัญหาที่พบในการวิจัยและพัฒนาระบบ

เนื่องจากระบบอัตโนมัติสำหรับบังคับควบคุมความต้องการของข้อมูลได้ถูกออกแบบมาเพื่อใช้กับโปรแกรมระบบจัดการฐานข้อมูลออราเคิล ซึ่งเป็นโปรแกรมระบบจัดการฐานข้อมูลแบบรีเลชันแนลที่มีความสามารถค่อนข้างสูงเมื่อเทียบกับโปรแกรมระบบจัดการฐานข้อมูลเชิงพาณิชย์ โปรแกรมอื่นๆ ถึงแม้จะเป็นโปรแกรมระบบจัดการฐานข้อมูลที่มีความสามารถสูงก็ตาม แต่ก็ยังมีข้อจำกัดหลายประการ ทำให้การกำหนดกฎข้อบังคับความต้องการของข้อมูลเพื่อบังคับใช้ในระบบฐานข้อมูลนั้นต้องเป็นไปตามข้อกำหนดและข้อจำกัดของโปรแกรมระบบจัดการฐานข้อมูลออราเคิลด้วย เช่น ข้อบังคับในการกำหนด database trigger ที่โปรแกรมระบบจัดการฐานข้อมูลออราเคิลอนุญาตให้กำหนดได้เพียง 1 โปรแกรมต่อหนึ่งชนิดของ database trigger ที่กำหนดให้เก็ตรายงานข้อมูลใดๆ ทำให้การบังคับควบคุมความต้องการของข้อมูลตามกฎหมายที่กำหนดไว้ ต้องแก้ปัญหานี้ด้วยการเขียนโปรแกรมที่ทำหน้าที่จัดการตรวจสอบความต้องการของข้อมูลไว้ใน stored procedure และเรียกใช้ stored procedure เหล่านั้นผ่าน database trigger ซึ่งอาจจะต้องใช้ database trigger เพียงหนึ่งชนิดหรือหลายชนิดก็ได้ สำหรับกฎข้อบังคับความต้องการของข้อมูลแต่ละข้อ

และนอกจากข้อจำกัดดังกล่าวแล้ว การขาดคุณสมบัติบางประการในการรักษาความต้องการของข้อมูลของโปรแกรมระบบจัดการฐานข้อมูลออราเคิลเอง เป็นสาเหตุที่ทำให้กฎข้อบังคับความต้องการของข้อมูลบางชนิดที่กำหนดขึ้น ไม่สามารถที่จะใช้บังคับได้จริง เช่นการที่โปรแกรมระบบจัดการฐานข้อมูลออราเคิลไม่ยินยอมให้เกิดการละเมิดกฎข้อบังคับความต้องการของข้อมูลที่กำหนดขึ้นได้ภายใน transaction ซึ่งทำให้กฎข้อบังคับความต้องการของข้อมูลชนิดที่บังคับข้อมูลในลักษณะที่เป็นกลุ่ม (กลุ่มของข้อมูลหมายถึงข้อมูลหลายๆ row) อาทิเช่น กฎชนิด cardinality, occurency frequency, equality และ cardinality of role ไม่สามารถใช้บังคับได้จริงในทางปฏิบัติ เนื่องจากหากกำหนดใช้กฎดังกล่าวก่อนที่จะตรารายงานข้อมูลจะมีข้อมูล ผู้ใช้จะไม่สามารถนำเข้าข้อมูลได้ หรือแม้กระทั่งการบังคับใช้กฎเหล่านี้หลังจากที่ได้บรรจุข้อมูลลงในตารางฐานข้อมูลแล้ว และข้อมูลเหล่านั้นมีความถูกต้องเป็นไปตามกฎข้อบังคับความต้องการของข้อมูลที่บังคับไว้ก็ตาม ผู้ใช้ก็จะไม่สามารถแก้ไขหรือนำเข้าข้อมูลที่จำเป็นและถูกต้องสำหรับระบบงานลงบนฐานข้อมูลได้เช่นกัน

แนวทางการประยุกต์ใช้

ดังที่ได้กล่าวไว้ในเบื้องต้นแล้วว่าปัจจัยที่สำคัญในการออกแบบและพัฒนาระบบฐานข้อมูลนั้น ต้องอาศัยรากฐานจากการออกแบบฐานข้อมูลที่ดี โดยฐานข้อมูลที่ออกแบบได้จะต้องสามารถรองรับข้อมูลที่จำเป็นต่อระบบงานได้อย่างถูกต้องและครบถ้วน ภายใต้การดูแลรักษาความถูกต้องของข้อมูลด้วยกฎข้อบังคับความถูกต้องของข้อมูลชนิดต่างๆ ที่วิเคราะห์ได้จากระบบงาน ซึ่งการออกแบบฐานข้อมูลให้สมบูรณ์ได้ ผู้ออกแบบฐานข้อมูลจะต้องมีความเข้าใจต่อข้อมูลในระบบงาน และสามารถแสดงความเข้าใจนั้นออกมาในรูปของแบบจำลองข้อมูลระดับแนวคิดได้อย่างถูกต้อง ซึ่งโดยมากแล้วผู้ที่ทำหน้าที่รับผิดชอบในการพัฒนาระบบฐานข้อมูลมักจะเป็นบุคคลคนละกลุ่มกับผู้ที่ทำหน้าที่ปฏิบัติงานจริงอยู่ในระบบงาน ซึ่งเป็นผู้ที่มีความรู้ความเข้าใจในข้อมูลและระบบงานในส่วนที่แต่ละบุคคลทำงานเกี่ยวข้องอยู่ดีที่สุด ดังนั้นการออกแบบฐานข้อมูลที่มีดีต้องอาศัยการประสานความคิดและความเข้าใจระหว่างบุคคลสองกลุ่มนี้อย่างมีประสิทธิภาพ ซึ่งเครื่องมือที่ใช้ในการประสานความคิดและความเข้าใจดังกล่าวมักจะเป็นวิธีการและเครื่องมือในการออกแบบฐานข้อมูลนั่นเอง

ระบบสร้างกฎข้อบังคับความถูกต้องของข้อมูล โดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนลนับว่าเป็นเครื่องมือที่มีประสิทธิภาพในการสื่อความเข้าใจระหว่างนักออกแบบฐานข้อมูลและผู้ปฏิบัติงานในระบบงานจริงอยู่มาก เนื่องจากใช้ภาษาธรรมชาติเป็นเครื่องมือในการสื่อสารระหว่างระบบและผู้ใช้งาน โดยนักออกแบบฐานข้อมูลสามารถใช้ความเข้าใจของตนเองที่ได้จากการวิเคราะห์ระบบงาน เขียนเป็นประโยคแสดงความสัมพันธ์และกฎข้อบังคับความถูกต้องของข้อมูลได้ด้วยภาษาธรรมชาติไวยากรณ์รูปแบบจำกัดที่สามารถเข้าใจได้ง่าย และให้ผู้ปฏิบัติงานจริงช่วยตรวจสอบชุดประโยคเหล่านั้นว่าถูกต้องครบถ้วนหรือไม่ ซึ่งผู้ปฏิบัติงานจริงสามารถอ่านและทำความเข้าใจกับประโยคดังกล่าวได้โดยไม่ต้องศึกษาทฤษฎีพื้นฐานการออกแบบฐานข้อมูลและวิธีการออกแบบฐานข้อมูลวิธีต่างๆ จึงทำให้การออกแบบฐานข้อมูลดำเนินไปอย่างมีประสิทธิภาพภายใต้การตรวจสอบความถูกต้องที่เชื่อถือได้

หลังจากที่นักออกแบบฐานข้อมูลได้ออกแบบแบบจำลองข้อมูลระดับแนวคิดในรูปของชุดประโยคภาษาธรรมชาติไวยากรณ์รูปแบบจำกัดและตรวจสอบความถูกต้องกับผู้ปฏิบัติงานจริงเรียบร้อยแล้ว ไฟล์ชุดคำสั่งที่ระบบสร้างกฎข้อบังคับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนลสร้างให้นั้น ผู้ใช้สามารถนำไปใช้สร้างตารางฐานข้อมูลและกฎข้อบังคับความถูกต้องของข้อมูลต่างๆ ได้ทันที หรือถ้าหากผู้ใช้มีความประสงค์จะเปลี่ยนแปลงคุณสมบัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของคอลัมน์, ตารางฐานข้อมูล หรือกฎต่างๆ ผู้ใช้สามารถใช้โปรแกรมที่กซ์อิดิตเตอร์แก้ไขได้โดยตรงภายในไฟล์ชุดคำสั่ง โดยจะต้องระวังแก้ไขทุกคำสั่งที่เกี่ยวข้องกับสิ่งที่ถูกเปลี่ยนแปลงแก้ไขให้เป็นไปในทิศทางเดียวกันด้วย

หรือนักออกแบบฐานข้อมูลอาจจะให้ผู้ปฏิบัติงานจริงในระบบงานเป็นผู้เขียนประโยคชุดคำสั่งภาษาธรรมชาติเอง โดยให้ผู้ใช้แต่ละหน่วยงานเขียนเฉพาะส่วนที่เกี่ยวข้องกับการทำงานของแผนกตนเท่านั้น ภายหลังจากได้ทำการอบรมชี้แจงโครงสร้างคุณสมบัติของภาษาธรรมชาติไวยากรณ์รูปแบบจำกัดให้แก่ผู้ใช้เสียก่อน แล้วจึงตรวจสอบชุดประโยคคำสั่งทั้งหมดที่ได้จากทุกหน่วยงานว่ามีข้อมูลใดขัดแย้งกันอยู่ แล้วจึงประชุมผู้ที่เกี่ยวข้องทุกฝ่ายเพื่อการตรวจสอบความถูกต้องและแก้ไขข้อขัดแย้งเหล่านั้น แต่วิธีการนี้มักจะเสียเวลาและยุ่งยากในการขอความร่วมมือมากกว่าวิธีแรกอยู่มาก

แนวทางการพัฒนาต่อ

ถึงแม้ระบบสร้างกฎบังคับความถูกต้องของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนลจะมีประโยชน์ในการออกแบบและพัฒนาระบบฐานข้อมูลดังที่กล่าวมา แต่ข้อจำกัดของระบบนี้ก็คือ ชุดของคำสั่งที่ใช้ในการกำหนดกฎข้อบังคับความถูกต้องของข้อมูลชนิดต่างๆ เป็นคำสั่งที่ไม่สามารถใช้ในการบังคับความถูกต้องของข้อมูลบนโปรแกรมระบบจัดการฐานข้อมูลโปรแกรมอื่นๆ ได้ เนื่องจากโปรแกรมระบบจัดการฐานข้อมูลแต่ละโปรแกรมได้รูปแบบของคำสั่งที่ใช้ในการกำหนดกฎข้อบังคับความถูกต้องของข้อมูลแบบอรรถธิบาย, stored procedure และ database trigger ไว้แตกต่างกัน ทำให้ชุดคำสั่งที่ใช้ในการกำหนดกฎข้อบังคับความถูกต้องของข้อมูลสำหรับโปรแกรมระบบจัดการฐานข้อมูลอรรถธิบายไม่สามารถจะนำไปใช้กับโปรแกรมระบบจัดการฐานข้อมูลอื่นได้ ดังนั้นหากผู้ใดประสงค์ที่จะนำแนวทางของงานวิจัยนี้ไปใช้ในการพัฒนาระบบอัตโนมัติที่ช่วยในการออกแบบและพัฒนาระบบฐานข้อมูลสำหรับโปรแกรมระบบจัดการฐานข้อมูลอื่นแล้ว ก็สามารถที่จะกระทำได้ โดยต้องศึกษาข้อมูลและความสัมพันธ์ภายในตารางเมตาดัสมม่า และโครงสร้างภาษาและวิธีการบังคับความถูกต้องของข้อมูลบนโปรแกรมระบบจัดการฐานข้อมูลชนิดนั้น เพื่อที่จะสามารถแปลงข้อมูลภายในตารางเมตาดัสมม่าเป็นคำสั่งที่ใช้ในการกำหนดกฎข้อบังคับความถูกต้องของข้อมูลสำหรับโปรแกรมระบบจัดการฐานข้อมูลชนิดนั้นๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบสร้างกฎบังคับความต้องการของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนลนี้ได้ยืดยาวถึงขีดจำกัดของกฎข้อบังคับความต้องการของข้อมูลที่ G.M. Nijssen และ T.A. Halpin ได้แบ่งประเภทไว้และนำมาแสดงในแบบจำลองข้อมูลในแอมเท่านั้น โดยได้ออกแบบโครงสร้างของภาษาธรรมชาติไว้อย่างรัดกุมรูปแบบจำกัดให้สามารถรองรับความหมายของกฎข้อบังคับความต้องการของข้อมูลแต่ละชนิดได้มากกว่าที่ทนายทั้งสองนี้กำหนดไว้ และที่แบบจำลองในแอมสามารถแสดงได้ โดยความสามารถของระบบในการสร้างกฎข้อบังคับความต้องการของข้อมูลนี้ ขึ้นอยู่กับโครงสร้างของภาษาคำสั่งที่ได้ออกแบบขึ้น และตารางเมตาสกิมมาที่ใช้รองรับข้อมูลจากภาษาที่ออกแบบขึ้น ดังนั้นโปรแกรมระบบสร้างกฎบังคับความต้องการของข้อมูลโดยใช้ภาษาธรรมชาติสำหรับฐานข้อมูลแบบรีเลชันแนลสามารถที่จะขยายขีดความสามารถออกไปได้อีก ด้วยการปรับปรุงโครงสร้างของภาษาและตารางเมตาสกิมมา เพื่อให้ผู้ใช้สามารถกำหนดกฎข้อบังคับความต้องการของข้อมูลด้วยรายละเอียดที่มากขึ้นกว่าเดิมได้ และปรับปรุงโปรแกรมให้สามารถสร้าง stored procedure และ database trigger สำหรับบังคับกฎเหล่านั้น บนระบบจัดการฐานข้อมูลที่ต้องการได้



บรรณานุกรม

- Ceri, S. and Widom, J. "Deriving production rules for constraint maintenance." In Proceedings of the 16th International Conference on Very Large Data Bases. VLDB Endowment, (1990) : 566-577
- Codd, E.F. The Relational Model for Database Management Version 2, U.S.A. : Addison-Wesley, 1990.
- David Y. Liu and Elizabeth A. Unger, "An Integrity Model Based on Knowledge and Belief," IEEE Transaction on Database, (1991) : 34-78.
- Esen Ozkarahan. Database Management Concepts, Design, and Practice, U.S.A. : Prentice-Hall, 1990.
- Ingres Corporation. INGRES Database Administrator's Guide for the UNIX Operating System Release 6.4, (December 1991)
- Microsoft Corporation. Transact-SQL Reference, Microsoft SQL Server, Client-Server Database Management System Version 4.2 for the Microsoft Windows NT Operating System.
- Moerkotte, G. and Lockemann, P. C. "Reactive consistency control in deductive databases." ACM Transactions on Database Systems, Vol.16, No.4, (1991) : 670-702.
- Nijssen, G.M. and Halpin, T.A. Conceptual Schema and Relational Database Design. A fact oriented approach. Prentice-Hall, 1989.
- Oracle Corporation. ORACLE 7 Server Concepts Manual. (December 1992)
- Oracle Corporation. ORACLE 7 Server SQL Language Reference Manual. (December 1992)
- Oracle Corporation. PL/SQL User's Guide and Reference. (December 1992)
- Walshe, A. "Formal Methods for Database Language Design and Constraint Handling," Software Engineering Journal, (January 1989)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ก.

**โครงสร้างคำสั่งสำหรับกำหนดกฎนิรณาย, store procedure และ
database trigger สำหรับ กฎข้อบังคับความถูกต้องของข้อมูลชนิดต่างๆ**

PRIMARY KEY

```
alter table table_name add constraint pk_table_name primary key (pk_attribute1,
pk_attribute2,...);
```

UNIQUE KEY

```
alter table table_name add constraint uk_table_name unique (uk_attribute1,
uk_attribute2..);
```

REFERENTIAL INTEGRITY CONSTRAINT

```
alter table table_name1 add constraint ic_name foreign key (col_name1, col_name2,... )
reference table_name2 (col1,col2, ...);
```

MEMBERSHIP CONSTRAINT

```
alter table table_name add constraint ic_name check (col_name in (str1, str2, ... , strn));
```

RANGE CONSTRAINT

```
alter table table_name add constraint ic_name check (col_name < upper_limit and
col_name > lower_limit) ;
```

OCCURENCY FREQUENCY IC

create or replace trigger af_ins_row_lect

after insert on *lecturer* (update, delete)

for each row

begin

 occur_name(:new.lecturer_name,:new.lecturer_surname);

end;

และ

create or replace procedure occur_<combination of subject of role> is

 cursor c1 is

 select count(*) from *table_of_roles*

 group by *combination of subject of role*;

 mem_no integer;

 status integer;

begin

 status := 0;

 open c1;

 loop

 fetch c1 into mem_no;

 exit when c1%notfound;

 if (mem_no < min_limit or mem_no > max_limit)

 then raise_application_error (-20620,' Occurency frequency of distinct
(combination of subject of role) must be in range of min_limit to max_limit');

 end if;

 end loop;

end;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ACYCLIC CONSTRAINT

create or replace procedure *acyclic* (e1 in char, e2 in char) is

subj1 char(10);

obj1 char(10);

obj2 char(10);

obj3 char(10);

subj2 char(10);

obj4 char(10);

obj5 char(10);

obj6 char(10);

cursor c1 is select child from parent where parent = subj1;

cursor c2 is select child from parent where parent = obj1;

cursor c3 is select child from parent where parent = obj2;

cursor c4 is select parent from parent where child = subj2;

cursor c5 is select parent from parent where child = obj4;

cursor c6 is select parent from parent where child = obj5;

begin

subj1 := e1;

open c1;

loop

fetch c1 into obj1;

exit when c1%notfound;

open c2;

loop

fetch c2 into obj2;

exit when c2%notfound;

if (intrans(obj2,e1) = 1)

then raise_application_error(-20668,'parent of child is acyclic');

```

end if;
open c3;
loop
    fetch c3 into obj3;
    exit when c3%notfound;
    if (intrans(obj3, e1) = 1)
    then raise_application_error(-20669,'parent of child is acyclic');
    end if;
end loop;
close c3;
end loop;
close c2;
end loop;
close c1;
subj2 := e2;
open c4;
loop
    fetch c4 into obj4;
    exit when c4%notfound;
    open c5;
    loop
        fetch c5 into obj5;
        exit when c5%notfound;
        if (intrans(e2,obj5) = 1)
        then raise_application_error(-20668,'parent of child is acyclic');
        end if;
        open c6;
        loop
            fetch c6 into obj6;

```

```

        exit when c6%notfound;

        if (intrans(e2,obj6) = 1)

            then raise_application_error(-20669,'parent of child is acyclic');

        end if;

    end loop;

    close c6;

end loop;

close c5;

end loop;

close c4;

end;

```

IRREFLEXIVE

```

alter table table_name add constraint ic_name check (col_name1 <> col_name2) ;

```

INTRANSITIVE CONSTRAINT

create or replace procedure *intran* (e1 in char, e2 in char) is

```

    subj    parent.parent%type;

    obj1    parent.child%type;

    obj2    parent.child%type;

    obj3    parent.child%type;

    cursor c1 is select child from parent where parent = subj;

    cursor c2 is select child from parent where parent = obj1;

    cursor c3 is select child from parent where parent = obj2;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
    subj := e1;
    open c1;
    loop
        fetch c1 into obj1;
        exit when c1%notfound;
        open c2;
        loop
            fetch c2 into obj2;
            exit when c2%notfound;
            if (obj2 = e2)
            then raise_application_error(-20668,'parent of child is intransitive');
            end if;
            open c3;
            loop
                fetch c3 into obj3;
                exit when c3%notfound;
                if (obj3 = e2)
                then raise_application_error(-20669,'parent of child is intransitive');
                end if;
            end loop;
        end loop;
        close c3;
    end loop;
    close c2;
end loop;
close c1;
end;

```

ASYMMETRIC CONSTRAINT

create or replace procedure *asym* (e1 in char, e2 in char) is

 cursor c1 is select child from parent where parent = e2;

 cursor c2 is select parent from parent where child = e1;

 subj parent.parent%type;

 obj parent.child%type;

begin

 subj := e1;

 open c1;

 loop

 fetch c1 into obj;

 exit when c1%notfound;

 if (obj = subj)

 then raise_application_error(-20666,'Parent of Child is asymmetric');

 end if;

 end loop;

 close c1;

 obj := e2;

 open c2;

 loop

 fetch c2 into subj;

 exit when c2%notfound;

 if (subj = obj)

 then raise_application_error(-20667,'Child of Parent is asymmetric');

 end if;

 end loop;

 close c2;

end;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CARDINALITY CONSTRAINT

create or replace procedure *cardi_club* is

```
    num    number(2);
```

begin

```
    select count(distinct club) into num
```

```
    from memberclub;
```

```
    if ((num > 4) or (num <2))
```

```
    then raise_application_error(-20640,'number of club cannot more than 4 and
```

```
not least than 2');
```

```
    end if;
```

end;

CARDINALITY_OF_ROLE CONSTRAINT

create or replace procedure *car_of_role* (*lect_name* in char, *lect_surname* in char, *myroom*
in char, *myaddr* in char) is

```
    role_no  integer := 0;
```

begin

```
    if (myroom is not null)
```

```
    then role_no := role_no + 1;
```

```
    end if;
```

```
    if (myaddr is not null)
```

```
    then role_no := role_no + 1;
```

```
    end if;
```

```
    if (chk_teach(lect_name,lect_surname) = 1)
```

```
    then role_no := role_no + 1;
```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end if;

if role_no < 2

then raise_application_error(-20655,'Lecturer play at least two roles of Room,
Address, Teaching');

end if;

end;

```

MANDATORY ENTITY CONSTRAINT

```

create or replace procedure man_ent_lect is

    mem_no          integer;

begin

    select count(*) into mem_no from lecturer where lect_type = 'ประจำ';

    if (mem_no < min_no)

    then raise_application_error(-20630,'set of lecturer must be has at least 2 lecturer
with lect_type ประจำ');

    end if;

    select count(*) into mem_no from lecturer where lect_type = 'พิเศษ';

    if (mem_no < 1)

    then raise_application_error(-20640,'set of lecturer must be has at least 1 lecturer
with lect_type ,พิเศษ');

    end if;

end;

```

DYNAMIC CONSTRAINT

สำหรับ Dynamic constraint นั้น จำเป็นจะต้องสร้างตารางฐานข้อมูล log table เพื่อรองรับข้อมูล ก่อนทำการเปลี่ยนแปลงไว้ด้วย และกำหนดชุดคำสั่งไว้ภายใน row level trigger ชนิด BEFORE DELETE , BEFORE INSERT และ BEFORE UPDATE เพื่อทำการเก็บข้อมูลที่เป็นอดีตไว้ภายใน log table ด้วย โดย log table จะมีโครงสร้างดังนี้คือ

```
log_file_name
( col_rowid    row_id,
  logtime      date,
  col_name     col%type)
```

โดย col_rowid จะเก็บ rowid ของข้อมูลแต่ละ row และ logtime จะทำหน้าที่เก็บวันเวลาที่ ค่า ข้อมูลใน row นั้นถูกเปลี่ยนแปลงไปจากค่าที่เก็บไว้ในคอลัมน์ col_name

create or replace trigger lec

after update of salary on lecturer

for each row

when (new.salary > old.salary)

declare

log **date;**

sal **number(7,2);**

interval **real;**

cursor c1 is select logtime, salary from sal_log where sal_row = :new.rowid;

begin

open c1;

loop

fetch c1 into log,sal ;

exit when c1%notfound;

if ((:new.salary - sal)/sal > 0.3)

then

select months_between(sysdate,log) into interval from dual;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่ภายนอก
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (interval < 12.0)
            then raise_application_error(-20677,'Salary cannot increase more
than 30 percent');
        end if;
    end if;
end loop;
close c1;
insert into sal_log values (:new.rowid,sysdate,:new.salary);
end;
```

TRANSITION_OF_NUMERIC_VALUES CONSTRAINT

```

create or replace procedure transition (lect_name in char ,lect_surname in char, oldsal in
number, newsal in number)
is
begin
    if (newsal < oldsal)
        then raise_application_error (-20880,'Salary cannot decrease');
    end if;
end;
```

TRANSITION_OF_STRING_VALUES CONSTRAINT

```

create or replace procedure tr_str (myold_pos in char,mynew_pos in char) is
begin
    if (myold_pos = 'saleman')
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    then raise_application_error(-20655,'transition of position for salesman is
supervisor');

    end if;

end if;

if (myold_pos = 'supervisor')
then if (mynew_pos <> 'sale manager')
    then raise_application_error(-20656,'transition of position for supervisor is
supervisor');

    end if;

end if;

if (myold_pos = 'sale manager')
then if (mynew_pos <> 'sale manager')
    then raise_application_error(-20657,'transition of position for sale manager
is invalid');

    end if;

end if;

end;
```

CYCLICABLE_TRANSITION_OF_STRING CONSTRAINT

create or replace procedure tr_cyc(myold_season in char,mynew_season in char) is

begin

if (myold_season = 'winter')

then if (mynew_season not in ('winter','autumn'))

**then raise_application_error(-20655,'transition of season for winter is
autumn');**

end if;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end if;

if (myold_season = 'autumn')

then if (mynew_season not in ('autumn', 'summer'))

    then raise_application_error(-20656,'transition of season for autumn is
summer');

    end if;

end if;

if (myold_season = 'summer')

then if (mynew_season not in ('summer','winter'))

    then raise_application_error(-20657,'transition of season for summer is
winter');

    end if;

end if;

end;

```

MANDATORY ROLE CONSTRAINT

กรณีอยู่บนตารางฐานข้อมูลเดียวกัน

```
alter table name_log modify (col_name not null);
```

กรณีอยู่ต่างตารางฐานข้อมูล

```
alter table table_name modify (col_name not null);
```

และ

```
create or replace trigger af_ins_row_name
```

```
after insert on table_name
```

```
for each row
```

```
begin
```

```
    mand_name(:new.pk_attribute1,:new.pk_attribute2);
```

end;

เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ

create or replace procedure *mand_teach*(*pk1* in char ,*pk2* in char) is

mem_no integer;

begin

select count(*) into *mem_no* from *table* where *col_pk1* = *pk1*

and *col_pk2* = *pk2*;

if *mem_no* = 0

then raise_application_error(-20600,'every entity must be role entity');

end if;

end;

EQUALITY CONSTRAINT

กรณีอยู่บนตารางฐานข้อมูลเดียวกัน

create or replace procedure *equa* (*myroom* in char, *mytel* in char) is

begin

if ((*mytel* is null and *myroom* is not null) or (*mytel* is not null and *myroom* is null))

then raise_application_error ('-20660','have Tel iff have Room');

end if;

end;

กรณีอยู่ต่างตารางฐานข้อมูล

create or replace procedure *subsettt*(*myarmy_id* in char, *mystudent_id* in char) is

mycamp training.camp%type;

begin

select *camp* into *mycamp*

from *training*

where *student_id* = *mystudent_id*;

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (mycamp is null and myarmy_id is not null) or (mycamp is not null and
myarmy_id is null)

        then raise_application_error(-20670,'Only training Student has Amry_ID
can training at camp');

        end if;

end;
```

และ

```

create or replace procedure subsett(mycamp in char, mystudent_id in char) is

    myarmy_id    student.army_id%type;

begin

    select army_id into myarmy_id
    from student
    where student_id = mystudent_id;

    if (myarmy_id is null and mycamp is not null) or (myarmy_id is not null and
mycamp is null)

    then raise_application_error(-20680,'Only Student has Amry_ID can
training at camp');

    end if;

end;
```

SUBSET CONSTRAINT

```

create or replace procedure subsett(mycamp in char, mystudent_id in char) is

    myarmy_id    student.army_id%type;

begin

    if mycamp is not null
```

then เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

select army_id into myarmy_id
from student where student_id = mystudent_id;
if myarmy_id is null
then raise_application_error(-20690,'Subset invalid');
end if;
end if;
end;
```

SUBTYPE CONSTRAINT

กรณีอยู่บนตารางฐานข้อมูลเดียวกัน

```

create or replace procedure subtype(myarmy_id in char, mygender in char) is
begin
    if (myarmy_id is not null and mygender = 'F')
    then raise_application_error(-20700,'Female cannot has army_id');
    end if;
end;
```

กรณีอยู่ต่างตารางฐานข้อมูล

```

create or replace procedure subbttype(mystudent_id in char) is
    mygender      student.gender%type;
begin
    select gender into mygender from student where student_id = mystudent_id;
    if (mygender = 'F')
    then raise_application_error(-20710,'Female cannot has army_id');
    end if;
end;
```



```

ST_ID char(7);
NAME varchar(20);
SURNAME varchar(30);
LECT_NAME varchar(30);
LECT_SURNAME varchar(30);
SUBJ_CODE char(5);
SUBJ_NAME varchar(50);
CREDIT number(1);
DEPT_CODE char(3);
DEPT_NAME char(30);
GENDER_CODE char(1);
CLUB_NAME varchar(20);
CAMP_NAME varchar(40);
ARMY_ID char(10);
STATUS char(1);
SP_NAME varchar(30);
SP_SURNAME varchar(30);
SEMESTER number(1);
YEAR number(4);
BUDGET_AMOUNT number(6);
PRODUCT_ID char(9);
PRODUCT_NAME varchar(30);
COLOR_NAME varchar(20);
ORDER_NO number(7);
ITEM_NO number(3);
QTY number(7);
GRADE char(2);
ADDRESS varchar(100);
TEL_NO char(7);
ROOM_NO char(4);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMPANY_NAME varchar(30);

SAL number(5);

LECT_TYPE varchar(10);

Student(ST_ID);

Lecturer is uniquely identified by (LECT_NAME, LECT_SURNAME);

Subject is uniquely identified by SUBJ_CODE;

Credit(CREDIT);

Dept(DEPT_CODE);

Gender(GENDER_CODE);

Club(CLUB_NAME);

Camp(CAMP_NAME);

Spouse(SP_NAME, SP_SURNAME);

Address(ADDRESS);

Room(ROOM_NO);

Tel(TEL_NO);

Semester(SEMESTER);

Year(YEAR);

Grade(GRADE);

Budget(BUDGET_AMOUNT);

Product(PRODUCT_ID);

Color(COLOR_NAME);

Order(ORDER_NO);

Item(ITEM_NO);

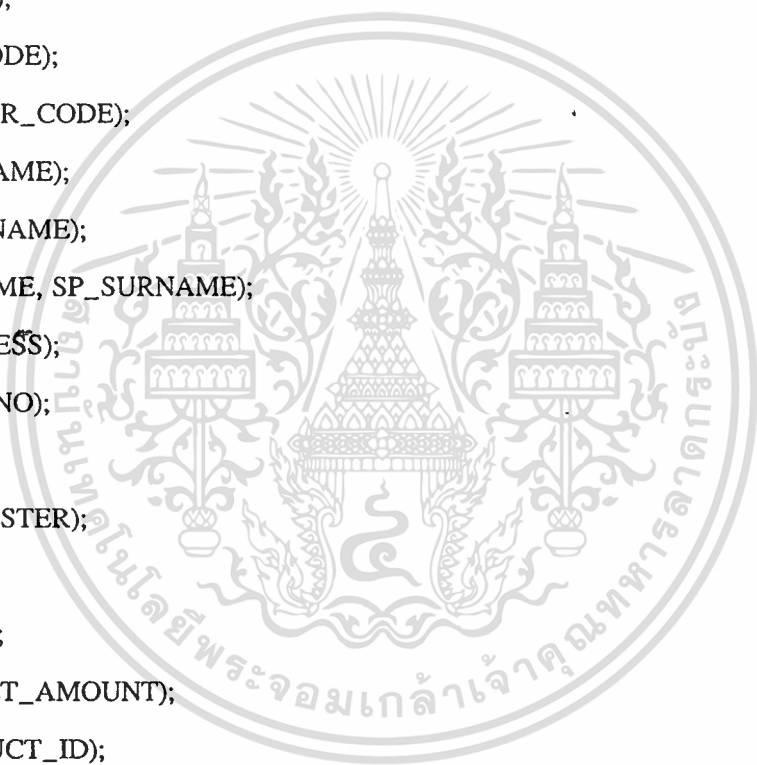
Qty(QTY);

Status(STATUS);

Company(COMPANY_NAME);

Salary(SAL);

Lect_type(LECT_TYPE);



Student has NAME;

Student has SURNAME;

Student has Gender;

Student enroll_in Subject & Subject is_enrolled_by Student;

Student enroll_in Subject in Semester of Year;

Student enroll_in Subject in Semester of Year take Grade;

Enrollment : Student enroll_in Subject in Semester of Year;

Enrollment has Status;

Semester of Year;

Term : Semester of Year;

Enrollment take Grade;

MaleStudent = Student has Gender 'M';

MaleStudent has ARMY_ID;

MaleStudent trained_at Camp;

Student is_member_of Club & Club is_political_of Student;

Student is_sponsor_of Club;

Student is_staff_of Subject;

Student is_staff_of Subject in Term;

Student is_in Dept;

Student is_advisee_of Lecturer & Lecturer is_advisor_of Student;

Student is_buddy_of Student & Student is_take_care_by Student;

Dept has DEPT_NAME;

Dept has Budget;

Lecturer is_in Dept;

Lecturer married_with Spouse & Spouse of Lecturer;

Lecturer lives_at Address & Address of Lecturer;

Lecturer occupies_lodging_at Room & Room of Lecturer;

Lecturer has_int_tel_with Tel & Tel of Lecturer;

Lecturer works_for Company;

Lecturer research_with Company;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Lecturer has Lect_type;

Lecturer earns Salary & Salary of Lecturer;

Lecturer teaches Subject;

Spouse works for Company & Company of Spouse;

Subject has SUBJ_NAME;

Subject has Credit;

Dept assign Subject & Subject of Dept;

Order with Item & Item of Order;

Product has Color;

Product has Color stock_with Qty;

Order with Item order_for Product;

Order with Item order_with Qty;

uk_stname := one Student has one NAME & one NAME of many Student;;

uk_stsur := one Student has one SURNAME & one SURNAME of many Student;

uk_gender := one Student has one Gender;

uk_enr := one Student enroll_in many Subject & one Subject is_enrolled_by many Student;

uk_enroll := one Student enroll_in Subject in Semester of many Year & one Year of Student enroll_in Subject in Semester;

uk_grade := one Student enroll_in Subject in Semester of Year take one Grade;

uk_en_sta := one Enrollment has one Status;

uk_en_grd := one Enrollment has one Grade;

uk_army := one MaleStudent has one ARMY_ID & one ARMY_ID of one MaleStudent;

uk_camp := one MaleStudent trained_at many Camp;

uk_club := one Student is_member_of many Club & one Club is_political_of one Student;

uk_spclub := one Student is_sponsor_of many Club;

uk_stdept := one Student is_in one Dept;

uk_advisor := one Student is_advisee_of one Lecturer & one Lecturer is_advisor_of many Student;

uk_buddy := one Student is_buddy_of one Student;

๒.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

uk_deptname := one Dept has one DEPT_NAME & one DEPT_NAME of one Dept;

uk_deptbud := one Dept has one Budget;

uk_lectdept := one Lecturer is_in one Dept;

uk_lectsp := one Lecturer married_with one Spouse & one Spouse of one Lecturer;

uk_addr := one Lecturer lives_at one Address & one Address of many Lecturer;

uk_room := one Lecturer occupies_lodging_at one Room;

uk_tel := oneLecturer has_int_tel_with one Tel;

uk_wcomp := one Lecturer works_for one Company;

uk_rscomp := one Lecturer research_with one Company;

uk_lactype := one Lecturer has one Lect_type;

uk_sal := one Lecturer earns one Salary & one Salary of many Lecturer;

uk_tsc := one Lecturer teaches many Subject;

uk_sp_comp := one Spouse works for one Company;

uk_subjname := one Subject has one SUBJ_NAME;

uk_credit := one Subject has one Credit;

uk_subjdept := one Subject of one Dept;

uk_orditem := one Order with many Item & one Item of manyOrder;

uk_stck_qty := one Product has Color stock_with one Qty;

uk_ord_prd := one Order with Item order_for one Product;

uk_ord_qty := one Order with Item order_with one Qty;

uk_fullname := combination of (NAME, SURNAME) of Studnet is unique;

nn_stname := every Student has NAME;

nn_stsur := every Student has SURNAME;

nn_gender := every Student has Gender;

nn_stat := every Enrollment has Status;

nn_grade := every enrollment has Grade;

nn_stdept := every Student is_in Dept;

nn_dept_name := every Dept has DEPT_NAME;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

nn_lectype := every Lecturer has Lect_type;

nn_subjname := every Subject has SUBJ_NAME;

nn_credit := every Subject has Credit;

nn_subjdept := every Subject of Dept;

nn_stckqty := every Product has Color stock_with Qty;

nn_ordprd := every Order with Item order_for Product;

nn_ordqty := every Order with Item order_with Qty;

mein_club := one Club has 5 to 100 Srtudent;

army_camp := MaleStudent trained_at Camp if MaleStudent has ARMY_ID;

val_grade := GRADE mustbe in ('A', 'B+', 'B', 'C+', 'C', 'D+', 'D', 'F', 'I');

val_gender := GENDER must be in ('M', 'F');

val_lectype := LECT_TYPE must be in ('ประจำ', 'พิเศษ');

val_credit := Subject has Credit in range of [0, 5];

ir_advisor := Student is_advisee_of Lecturer is irreflexive;

asym_advisor := Student is_advisee_of Lecturer is iasymmetric;

acy_advisor := Student is_advisee_of Lecturer is acyclic;

need_lectype := set of Lecvturer mustbe has at least 3 Lecturer has Lectype 'ประจำ';

sel_lect := every Lecturer play exactly 1 roles of { 'Lecturer lives_at Address', 'Lecturer occupies_lodging_at Room' };

lect_office := every Lecturer play at least one roles of { 'Lecturer is_in Dept', 'Lecturer works_for Company' };

dual_lect := Lecturer occupies_lodging_at Room iff Lecturer has_int_tel Tel;

ประวัติผู้เขียน

นางมยุรี เลิศเวชกุล เกิดวันที่ 21 พฤศจิกายน 2510 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาระดับอุดมศึกษาในหลักสูตรวิศวกรรมศาสตรบัณฑิต จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2531 และเข้ารับราชการตำแหน่งอาจารย์ประจำคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง ตั้งแต่ปี พ.ศ. 2536 ปัจจุบันดำรงตำแหน่งอาจารย์ระดับ 5 สังกัดภาควิชาเทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์

