สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การออกแบบและพัฒนาฐานข้อมูลเชิงเวลาด้วยวิธี ทีไนแอม

THE DESIGN AND DEVELOPMENT OF A TEMPORAL DATABASE USING TNIAM

ผุสดี พรผล

PUTSADEE PORAPHOL

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2546

ISBN 974-324-788-2

เลขหมู่.....
เลขทะเบียน 48922
วัน, เดือน, ปี 12 ส.ค. 2547

.b.....
.i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THE DESIGN AND DEVELOPMENT OF A TEMPORAL DATABASE USING TNIAM

PUTSADEE PORNPHOL

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIRMENT FOR THE DEGREE OF

MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2003

ISBN 974-324-788-2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2003

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การออกแบบและพัฒนาฐานข้อมูลเชิงเวลาด้วยวิธีทีไนแอม
นักศึกษา	นางสาวผุสดี พรผล
รหัสประจำตัว	42067014
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
พ.ศ.	2546
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร.ศุภมิตร จิตตะยโศธร

บทคัดย่อ

การออกแบบฐานข้อมูลที่อยู่นอกเหนือการเปลี่ยนแปลงของเวลา วิธีการไนแอม(NIAM : Nijssen's Information Analysis Methodology) ถือว่าเป็นวิธีการที่ดีวิธีการหนึ่ง เพราะเป็นวิธีที่สามารถแสดงความหมาย ความสัมพันธ์ และข้อจำกัดต่างๆของข้อมูล ด้วยแบบจำลองข้อมูลที่ประกอบไปด้วยสัญลักษณ์ต่างๆ และเป็นวิธีที่มีแอลกอรึธึมที่สามารถออกแบบฐานข้อมูลแบบรีเลชันเนลที่อยู่ในรูปของ Fifth Normal Form (5NF)ได้โดยตรง

วิทยานิพนธ์นี้นำเสนอแนวคิดเกี่ยวกับการขยายแบบจำลองข้อมูลไนแอมให้รองรับข้อมูลเชิงเวลา(TNIAM: Temporal Nijssen's Information Analysis Methodology) เป็นเครื่องมือช่วยในการออกแบบฐานข้อมูล โดยขยายแนวคิดจากวิธีการไนแอมให้สามารถรองรับข้อมูลที่เปลี่ยนแปลงไปตามเวลา โดยมีกระบวนการในการออกแบบทั้งในระดับแนวคิดที่จะได้ Conceptual Database Schema และ กระบวนการแปลง Conceptual Database Schema เป็น Relational Database Schema ให้อยู่ในรูปของ Fifth Normal Form (5NF) ได้โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	The Design and Development Of A Temporal Database Using TNIAM
Student	Miss. Putsadee Pornphol
Student ID.	42067014
Degree	Master of Science
Programme	Information Technology
Year	2003
Thesis Advisor	Assoc.Prof.Dr.Suphamit Chittayasothorn

ABSTRACT

Many database applications manage information that varies over time. NIAM(Nijssen's Information Analysis Methodology) is a popular design method which captures the meaning , relationships and integrity constraints of data in symbolic diagram. There exists an algorithm that transforms a NIAM conceptual schema into a fifth normal form relational database schema.

This thesis presents a temporal database design method using the TNIAM (Temporal Nijssen's Information Analysis Methodology). It is a temporal extension to NIAM. There exists an algorithm that transforms a TNIAM conceptual schema into a fifth normal form relational database schema.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดี ด้วยความรู้ต่าง ๆ คำแนะนำ คำปรึกษา และแนวทางการทำวิจัย จาก รองศาสตราจารย์ ดร.ศุภมิตร จิตตะยโศธร ผู้อำนวยการสำนักวิจัย และบริการคอมพิวเตอร์ อาจารย์ผู้ควบคุมวิทยานิพนธ์ ผู้วิจัยรู้สึกซาบซึ้ง และกราบขอบพระคุณในความอนุเคราะห์ และสนับสนุนเป็นอย่างสูง

กราบขอบพระคุณด้วยความเคารพรัก และศรัทธาอย่างสูงสุด ด้วยพระคุณและความรักจาก คุณพ่อ สมพงษ์ และ คุณแม่ชูจิตร พรผล ผู้ให้กำเนิด อบรมสั่งสอน ชี้นำแนวทางการศึกษา ความอดทน และเป็นกำลังใจในทุก ๆ ด้านจนงานวิจัยสำเร็จลุล่วง

ขอขอบพระคุณคณาจารย์ทุกท่าน ที่ประสิทธิประสาทวิชาความรู้ ทำให้สามารถนำความรู้มาใช้ในงานวิจัย ขอบคุณบุคลากรคณะเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัยทุกท่าน และคุณณัฐรียา พริ้งสกุลชัย เลขานุการผู้อำนวยการ สำนักวิจัยฯ สจล. สำหรับการอำนวยความสะดวก และประสานงานต่าง ๆ ด้วยดี

ขอขอบพระคุณ โครงการพัฒนาวิทยาศาสตร์ และวิทยาศาสตร์ประยุกต์(พ.ว.ศ.) สำนักงานสภาสถาบันราชภัฏ ที่สนับสนุนทุนการศึกษา และทำวิจัย

ขอขอบพระคุณสถาบันราชภัฏภูเก็ต ที่สนับสนุนทรัพยากร และเวลาในการทำวิจัย

ขอขอบคุณกำลังใจจากเพื่อนๆ เทคโนโลยีสารสนเทศ รุ่นที่7.1 (IS7) ทุกๆคนที่เป็นกำลังใจและให้ความช่วยเหลือในการติดต่อประสานงาน โดยเฉพาะคุณกอบชัย นิยมธรรม คุณเธียรชัย พัฒนาศิริเวทิน และขอขอบคุณกำลังใจ และการสนับสนุนในทุก ๆ ด้าน จากคุณธีระพงศ์ ตุละ

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

ผุสดี พรผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VII

สารบัญรูป.....VIII

บทที่ 1 บทนำ.....1

1.1 ความเป็นมา และความสำคัญของปัญหา.....1

1.2 ความมุ่งหมาย และวัตถุประสงค์ของการศึกษา.....1

1.3 ทฤษฎี หรือแนวความคิดที่ใช้ในการวิจัย.....2

1.4 งานวิจัยที่เกี่ยวข้อง.....2

1.5 ขอบเขตของการวิจัย.....6

1.6 โครงสร้างวิทยานิพนธ์.....7

1.7 ประโยชน์ที่คาดว่าจะได้รับ.....7

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....8

2.1 ฐานข้อมูลเชิงเวลา.....8

2.2 ประเภทของฐานข้อมูลเชิงเวลา.....10

2.2.1 ฐานข้อมูลวาลิดไทม์.....10

2.2.2 ฐานข้อมูลทรานส์แอกชันไทม์.....10

2.2.3 ฐานข้อมูลไบเทมโพรอล.....10

2.3 แบบจำลองข้อมูลไนแอม.....11

2.3.1 ส่วนประกอบไนแอม.....12

2.3.2 สัญลักษณ์ที่ใช้ในไนแอม.....12

2.3.3 กฎข้อบังคับความถูกต้องของข้อมูลที่ใช้ ในแบบจำลองระดับแนวคิดไนแอม.....14

2.3.4 กระบวนการในการแปลงแบบจำลองระดับแนวคิดไนแอม ให้อยู่ในรูปของโครงสร้างฐานข้อมูลเชิงสัมพันธ์.....19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 3 วิธีการออกแบบฐานข้อมูลเชิงเวลาด้วยวิธีการทีไนแอม.....20

3.1 รูปแบบเวลาที่ใช้ในแบบจำลองทีไนแอม.....20

3.1.1 รูปแบบเวลาของEntity Type.....20

3.1.2 Valid Time.....20

3.1.3 Transaction Time.....20

3.1.4 เวลาที่มีการกำหนดค่าของเวลาที่แน่นอน.....21

3.2 ส่วนประกอบของทีไนแอม.....21

3.2.1 Temporal Entity Type.....21

3.2.2 Temporal Fact Type.....22

3.3 ขั้นตอนการออกแบบฐานข้อมูลเชิงเวลาดวยวิธีการทีไนแอม.....23

3.3.1 ขั้นตอนการเขียนโครงร่างหลัก.....23

3.3.2 ขั้นตอนการเขียนโครงร่างย่อย.....24

3.4 ขั้นตอนการแปลงทีไนแอมเป็นฐานข้อมูลเชิงสัมพันธ์.....27

3.4.1 กรณี Temporal Fact Type มี Life span ระบุอยู่.....27

3.4.2 กรณี Temporal Fact Type มี Valid Time ระบุอยู่.....31

3.4.3 กรณี Temporal Fact Type มี Transaction Time ระบุอยู่.....40

3.4.4 กรณี Temporal Fact Type มี ทั้ง Valid Time และ Transaction Time.....41

3.5 ขั้นตอนการแปลงทีไนแอมเป็นฐานข้อมูลเชิงสัมพันธ์ที่อยู่ในรูปของ Temporal Database.....42

3.5.1 กรณี Temporal Fact Type มี Life span ระบุอยู่.....43

3.5.2 กรณี Temporal Fact Type มี Valid Time ระบุอยู่.....45

3.5.3 กรณี Temporal Fact Type มี Transaction Time ระบุอยู่.....51

3.5.4 กรณี Temporal Fact Type มี ทั้ง Valid Time และ Transaction Time.....51

3.6 ขั้นตอนการแปลงทีไนแอมเป็นฐานข้อมูลเชิงสัมพันธ์ ที่อยู่ในรูปของ Object Relational Database.....51

3.6.1 กรณี Temporal Fact Type มี Life span ระบุอยู่.....51

3.6.2 กรณี Temporal Fact Type มี Valid Time ระบุอยู่.....54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.6.3 กรณี Temporal Fact Type มี Transaction Time ระบุอยู่.....61
3.6.4 กรณี Temporal Fact Type มี ทั้ง Valid Time และ Transaction Time.....61

บทที่ 4 การปฏิบัติการบนฐานข้อมูลเชิงเวลา.....62
4.1 การจัดการกับ Valid Time State Table.....63
4.1.1 Current Modification.....63
4.1.2 Sequence Modification.....70

บทที่ 5 การประยุกต์ TNIAM 75
5.1 ขั้นตอนการออกแบบฐานข้อมูลเชิงเวลาด้วยวิธีการทีไนแอม.....75
5.1.1 ขั้นตอนการเขียนโครงร่างหลัก.....75
5.1.2 ขั้นตอนการเขียนโครงร่างย่อย.....78
5.2 ขั้นตอนการแปลงทีไนแอมเป็นฐานข้อมูลเชิงสัมพันธ์.....81
5.3 แสดงหน้าจอต่าง ๆของระบบบริการข้อมูลนักศึกษาสำหรับอาจารย์ที่ปรึกษา.....81
5.3.1 โครงสร้างของระบบบริการข้อมูลนักศึกษาสำหรับอาจารย์ที่ปรึกษา.....83

บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ.....89
6.1 สรุปผลการวิจัย.....89
6.2 ข้อเสนอแนะ.....89

เอกสารอ้างอิง.....90
หนังสือรับรองผลงานวิจัยที่ได้รับการตีพิมพ์.....92
ประวัติผู้เขียน.....93

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

1.1 ตารางสรุปการใช้รูปแบบเวลาของ Time ER Model.....2

4.1 ตารางข้อมูลDEPARTMENT.....63

4.2 General Scenario ของ DEPARTMENT.....65

4.3 ผลของ Current Delete General Scenario.....65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

1.1 ตัวอย่าง Time ER Diagram.....3

1.2 แสดง E-Relation Employee.....3

1.3 แสดง A-Relation Employee_Salary.....4

1.4 Time ER Diagram แสดง Superclass /Subclass Relationship With Disjoint and Optional.....5

2.1 การจัดเก็บฐานข้อมูลของระบบฐานข้อมูลปกติ.....8

2.2 การจัดเก็บข้อมูลของระบบฐานข้อมูลเชิงเวลา.....8

2.3 แสดงฐานข้อมูลประวัติ.....10

2.4 ฐานข้อมูลย้อนกลับ.....10

2.5 ฐานข้อมูลสแนปช็อท.....11

2.6 ฐานข้อมูลไบเทมโพรอล.....11

2.7 แสดงความสัมพันธ์แบบหนึ่งต่อหลายหน่วย.....14

2.8 แสดงความสัมพันธ์แบบหนึ่งต่อหนึ่งหน่วย.....14

2.9 แสดง Inter fact type uniqueness constraints.....15

2.10 แสดง Mandatory role constraints.....15

2.11 แสดง Inclusion mandatory role constraints.....16

2.12 แสดง Entity type constraints.....16

2.13 แสดง Subset constraints.....17

2.14 แสดง Equality constraints.....17

2.15 แสดง Exclusion constraints.....18

2.16 แสดง Sub Type constraints.....18

3.1 Temporal Entity Type TEACHER.....21

3.2 Temporal Entity Type ระบุค่า Life span ภายนอก Entity Type.....21

3.3 Temporal Fact Type ระบุค่า Valid Time.....22

3.4 Temporal Fact Type ระบุค่า Life span.....22

3.5 Temporal Fact Type ระบุค่า Transaction time.....23

3.6Temporal Fact Type ระบุค่า Valid Time และ Transaction Time.....23

3.7 ตัวอย่างการเขียนโครงร่างหลักที่ไนแอม.....24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ หน้า

3.8 ตัวอย่างการเขียนโครงร่างย่อยทีไนแอม STUDENT.....25

3.9 ตัวอย่างโครงร่างย่อย DEPARTMENT.....25

3.10โครงร่างย่อยCURRICULUMN.....26

3.11 โครงร่างย่อย TEACHER.....26

3.12 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง.....27

3.13 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.12.....28

3.14 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อกลุ่ม.....28

3.15 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.14.....29

3.16 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบกลุ่มต่อกลุ่ม.....29

3.17 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.16.....30

3.18 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง.....31

3.19 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.18.....32

3.20 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง.....33

3.21 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.20.....34

3.22 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อกลุ่ม.....35

3.23 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.22.....36

3.24 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อกลุ่ม.....36

3.25 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.24.....37

3.26 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบกลุ่มต่อกลุ่ม.....38

3.27 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.26.....39

3.28 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบกลุ่มต่อกลุ่ม.....39

3.29 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.28.....40

3.30 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง.....41

3.31 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.30.....41

3.32 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง.....42

3.33 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.32.....42

3.34 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง.....43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ หน้า

3.35 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.34.....43

3.36 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อกลุ่ม.....44

3.37 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.36.....44

3.38 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบกลุ่มต่อกลุ่ม.....45

3.39 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.38.....45

3.40 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง.....46

3.41 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.40.....46

3.42 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง.....47

3.43 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.42.....47

3.44 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อกลุ่ม.....47

3.45 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.44.....48

3.46 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อกลุ่ม.....48

3.47 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.46.....49

3.48 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบกลุ่มต่อกลุ่ม.....49

3.49 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.48.....50

3.50 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบกลุ่มต่อกลุ่ม.....51

3.51 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.50.....50

3.52 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง.....51

3.53 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.52.....52

3.54 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อกลุ่ม.....52

3.55 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.54.....53

3.56 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบกลุ่มต่อกลุ่ม.....53

3.57 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.56.....54

3.58 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง.....54

3.59 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.58.....55

3.60 ตัวอย่างข้อมูลของรีเลชันรูป 3.59.....55

3.61 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง.....55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ หน้า

3.62 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.61.....56

3.63 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อกลุ่ม.....57

3.64 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.63.....57

3.65 ตัวอย่างข้อมูลของรีเลชันรูป 3.64.....57

3.66 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อกลุ่ม.....58

3.67 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.66.....58

3.68 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบกลุ่มต่อกลุ่ม.....59

3.69 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.68.....60

3.70 ตัวอย่างข้อมูลของรีเลชันรูป 3.69.....60

3.71 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบกลุ่มต่อกลุ่ม.....60

3.72 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.71.....61

4.1 การเชื่อมต่อเวลากับฐานข้อมูล.....62

4.2 ตัวอย่างการ INSERT.....64

4.3 ตัวอย่างการ DELETE.....64

4.4 ตัวอย่างการเขียน Current Delete General Scenario.....65

4.5 ตัวอย่างการลบข้อมูลแบบ Current Delete Restricted Scenario.....66

4.6 ตัวอย่างการ Delete ทั้ง General Senario และ Restricted Scenario.....66

4.7 ตัวอย่างคำสั่งการแก้ไขข้อมูลโดยไม่สนใจเวลา.....67

4.8 ตัวอย่างการแก้ไขข้อมูลแบบ Current Update General Scenario.....68

4.9 แสดง 3 กรณีของ Current Update General Scenario.....68

4.10 ตัวอย่างการแก้ไขข้อมูลแบบ Current Update Restricted Scenario.....69

4.11 ตัวอย่าง 3กรณีของ Current Update General Scenario.....69

4.12 ตัวอย่างการเขียนคำสั่ง INSERT ใน Sequence Modification.....70

4.13 ตัวอย่างการเขียนคำสั่ง DELETE ใน Sequence Modification.....71

4.14 แสดง 4 กรณีของ Sequence DELETE.....72

4.15 ตัวอย่างการเขียน Sequence UPDATE.....73

4.16 ภาพแสดงแทนคำสั่ง Sequence UPDATE.....74

5.1 แสดง Temporal Entity Type ที่สนใจของระบบบริการข้อมูลนักศึกษาทั้งหมด.....75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ หน้า

5.2 แสดง Temporal Entity Type ทั้งหมดจากขั้นตอนที่ 1 ที่ระบุ Life Span 76

5.3 แสดง การระบุ Temporal Fact Type ให้กับ Temporal Entity Type จากขั้นตอนที่ 3 ตาม เงื่อนไขที่ระบุ.....77

5.4 แสดง การระบุรูปแบบเวลาให้กับ Temporal Fact Type 77

5.5 แสดง การระบุ Temporal Fact Type ให้กับ Temporal Entity Type DEPARTMENT กับ Label Type DNAME ที่ระบุรูปแบบเวลา Valid Time.....79

5.6 แสดง การระบุ Temporal Fact Type ให้กับ Temporal Entity Type TEACHER กับ Label Type TNAME และ Label Type TSURNAME ที่ระบุรูปแบบเวลา Valid Time.....79

5.7 แสดง การระบุ Temporal Fact Type ให้กับ Temporal Entity Type STUDENT กับ Label Type ทั้งหมด 81

5.8 แสดง 2 รีเลชันที่ได้จากการแปลง TNIAM จากรูปที่ 5.5 82

5.9 แสดง 3 รีเลชันที่ได้จากการแปลง TNIAM จากรูปที่ 5.6.....82

5.10 แสดง 10 รีเลชันที่ได้จากการแปลง TNIAM จากรูปที่ 5.7.....82

5.11แสดงโครงสร้างของระบบงานบริการข้อมูลนักศึกษาสำหรับอาจารย์ที่ปรึกษา.....84

5.12 แสดงเมนูหลัก ซึ่งผู้ใช้สามารถเลือกใช้งานระบบงานได้อย่างอิสระ.....84

5.13 แสดงจอภาพการบันทึกข้อมูลโปรแกรมวิชา.....84

5.14 แสดงจอภาพ การบันทึกข้อมูลอาจารย์ที่ปรึกษา.....85

5.15 หน้าจอการบันทึกข้อมูลนักศึกษา.....85

5.16 หน้าจอรายละเอียดโปรแกรมวิชา.....86

5.17 หน้าจอรายละเอียดอาจารย์.....86

5.18 หน้าจอแสดงรายละเอียดนักศึกษาทั้งหมด.....87

5.19 แสดงรายละเอียดการกู้ยืมเงิน.....87

5.20 แสดงรายละเอียดผลการเรียน.....88

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การออกแบบฐานข้อมูล นับได้ว่าเป็นงานที่มีความสลับซับซ้อน ซึ่งผู้ออกแบบฐานข้อมูลมักต้องการที่จะสร้างให้ระบบมีประโยชน์มีประสิทธิภาพและสมบูรณ์มากที่สุด ขั้นตอนที่สำคัญ และเป็นขั้นตอนแรกของการออกแบบฐานข้อมูล คือ การกำหนดโครงสร้างของฐานข้อมูล โดยอาศัยแบบจำลองข้อมูล (Data Model)

ฐานข้อมูลเชิงเวลาเป็นฐานข้อมูลที่สนับสนุนการเก็บ และการสอบถามข้อมูลที่แปรผันตามเวลา ซึ่งแตกต่างจากฐานข้อมูลแบบเดิม (Conventional Database) ที่ออกแบบมาเพื่อเก็บข้อมูลที่เป็นข้อมูลปัจจุบัน เมื่อมีการแก้ไขข้อมูลก็จะทำให้ค่าของข้อมูลเดิมที่มีอยู่ถูกแทนที่ด้วยข้อมูลใหม่ที่ทำการแก้ไข สำหรับข้อมูลที่จัดเก็บในฐานข้อมูลเชิงเวลาจะแตกต่างจากข้อมูลที่จัดเก็บในฐานข้อมูลแบบเดิม ตรงที่จะมีการเพิ่มส่วนของเวลาติดไปกับส่วนของข้อมูล ซึ่งประกอบด้วย ช่วงเวลาที่ข้อมูลนั้นเป็นจริง(Valid Time) และช่วงเวลาที่ข้อมูลนั้นถูกจัดเก็บในฐานข้อมูล (Transaction Time)

ลักษณะข้อมูลในฐานข้อมูลเชิงเวลาเป็นลักษณะที่มีความเคลื่อนไหว (Dynamic)ซึ่งจะแตกต่างจากข้อมูลในฐานข้อมูลแบบเดิมที่มีลักษณะคงที่ (Static) ดังนั้นแบบจำลองข้อมูลที่ใช้กับการออกแบบฐานข้อมูลแบบเดิมจึงไม่สามารถนำมาใช้ได้กับการออกแบบฐานข้อมูลเชิงเวลา ดังนั้นจึงจำเป็นต้องพิจารณาหาแบบจำลองข้อมูลที่เหมาะสม ในการออกแบบฐานข้อมูลเชิงเวลาและหาวิธีการที่จะช่วยในการออกแบบฐานข้อมูลเชิงเวลา เพื่อให้ได้ระบบที่สมบูรณ์และมีประสิทธิภาพ

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วิธีการออกแบบฐานข้อมูลเชิงเวลา ยังไม่มีวิธีการออกแบบใด ๆ ที่เป็นมาตรฐาน ดังนั้นในงานวิจัยนี้ จึงได้มีการนำเสนอ วิธีการออกแบบฐานข้อมูลเชิงเวลาด้วยวิธีการทีไนแอม ซึ่งเป็นวิธีที่ขยายมาจากวิธีการออกแบบฐานข้อมูล ไนแอม ซึ่งเป็นวิธีที่ได้รับความเชื่อถือ และใช้กันอย่างแพร่หลาย วิธีการออกแบบฐานข้อมูลเชิงเวลาด้วยวิธีการ ทีไนแอม มีกระบวนการที่สามารถแปลงไปอยู่ในรูปของฐานข้อมูลเชิงสัมพันธ์ที่อยู่ในรูป ของ 5NF ได้โดยตรง ซึ่งจะเป็นประโยชน์สำหรับการออกแบบฐานข้อมูลเชิงเวลา ที่มีลักษณะของข้อมูลที่ซับซ้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย

ทฤษฎี หรือแนวความคิดที่นำมาใช้ในการวิจัย ในวิทยานิพนธ์ฉบับนี้ คือทฤษฎีหรือแนวคิด เกี่ยวกับฐานข้อมูลเชิงเวลา การออกแบบฐานข้อมูลเชิงสัมพันธ์ด้วยวิธีการของ NIAM Concept Schema ฐานข้อมูลเชิงสัมพันธ์

1.4 งานวิจัยที่เกี่ยวข้อง

Heidi Gregersen และ คณะนักวิจัย [4][5] ได้นำเสนอการออกแบบฐานข้อมูลเชิงเวลาด้วยวิธีการTemporal ER Model หรือ Time ER Model ซึ่งเป็นแบบจำลอง ที่พัฒนามาจากรูปแบบของ E-R Model ให้มีความสามารถรองรับโครงสร้างของข้อมูลที่มีการเปลี่ยนแปลงไปตามเวลา รูปแบบโครงสร้างของ Temporal ER Model จะทำการจัดเก็บรูปแบบของเวลาที่ประกอบไปด้วย Valid Time, Transaction Time และ Lifespan (Existence Time) ซึ่งเป็นค่าของเวลาที่แสดงให้ทราบว่า Entityนั้นมีการเปลี่ยนแปลงค่าของข้อมูลที่เกิดขึ้นไปตามการเปลี่ยนแปลงของเวลาอยู่อย่างต่อเนื่อง ซึ่งจะแตกต่างจากค่าของ Valid Time การกำหนดรูปแบบของเวลาสำหรับ TimeER Model สรุปได้ดังตารางที่1.1

ตารางที่1.1 สรุปการใช้รูปแบบของเวลาของ Time ER Model

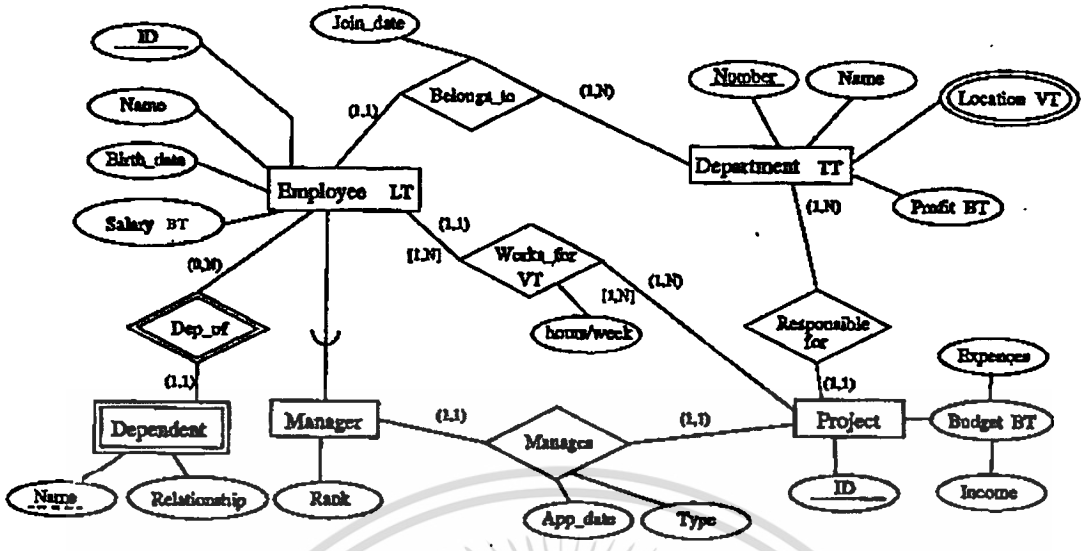
	Entity Type	Relationship Type	Superclass/Subclass Relationship	Attribute
Lifespan	Yes	(Yes)	No	No
Valid Time	No	(Yes)	No	Yes
Transaction Time	Yes	Yes	No	Yes

กรณีของRelationship Typeที่มีค่าLifespan และ Valid Time เป็น (Yes) หมายความว่าต้องพิจารณาจาก Entity Type และ Attribute ที่เกี่ยวข้อง

ส่วนประกอบของ Temporal ER Model มีลักษณะเหมือนกับ E-R Model ที่ใช้ในการออกแบบฐานข้อมูลแบบบเดิม เพียงแต่เพิ่มในส่วนของรายละเอียดของรูปแบบเวลาต่าง ๆ ลงไปในส่วนประกอบนั้นตามเงื่อนไขที่ต้องการ

ขั้นตอนการออกแบบฐานข้อมูลเชิงเวลาด้วย Temporal ER Model ประกอบไปด้วย 3 ขั้นตอน ได้แก่ ขั้นตอนแรก ขั้นตอนการเขียน Temporal ER Model หรือ Time ER Model ขั้นตอนที่สองเป็นขั้นตอนการแปลง Time ER Model ไปเป็น Surrogate based relational model และขั้นตอนที่สามขั้นตอนแปลง Surrogate based relational model เป็น Lexically-based relational

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่1.1 แสดงตัวอย่าง Time ER Diagram of a company database

Relation และ Key Constraint ใน Surrogate based relational model ประกอบด้วย

- E-Relation เป็นรีเลชันที่เกิดจาก Entity Type จาก TimeER Model โดยที่ E-Relationจะมีลักษณะเป็น Single Attribute สำหรับ E-Realtion ที่เกิดจาก Non-Temporal Entity และชื่อ Attribute จะตามด้วย "ϕ" สำหรับ E-Relation ที่เกิดจาก Temporal Entity จะมีลักษณะเช่นเดียวกับ E-Relation ที่เกิดจาก Non-Temporal Entity แต่จะมี Time-Attribute ที่ระบุใน Entity ตัวอย่าง E-Relation แสดงได้ดังรูปที่ 1.2

Employee

Employeeϕ	TT_s	TT_e	LS_s	LS_e
E1	1	8	10	15
E2	9	UC	5	20

รูปที่ 1.2 แสดง E-Relation Employee

- A-Relation ป็นรีเลชันที่เกิดจาก Attribute ของ Entity Type จาก Time ER Model และเป็นรีเลชันที่ประกอบด้วย Attribute มากกว่าหนึ่งตัว A-Relation นอกจากจะประกอบด้วย Attribute ที่มาจาก Entity Type แล้ว ยังประกอบด้วย Attribute จาก E- Relation ที่เป็น Attribute ที่ลงท้ายด้วย "ϕ" เพื่อที่จะทำหน้าที่เป็น Foreign Key อ้างอิงไปยัง E-Relation ที่ A-Relation ขึ้นอยู่ด้วย A-Relation ของE-Relation แต่ละตัวจะแยกเป็นแต่ละA-Relation ระหว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Temporal Attribute และ Non-Temporal Attribute โดย A-Relation ที่เกิดจาก Temporal attribute จะมีการระบุค่าของ Time-Attribute ตามที่ปรากฏใน Time ER Model ดังแสดงในรูปที่1.3

Employee_Salary

Employeeϕ	Salary	TT_s	TT_e	VT_e	VT_e
E1	7K	1	14	10	15
E2	7K	15	UC	5	20

รูปที่ 1.3 แสดง A-Relation Employee_Salary

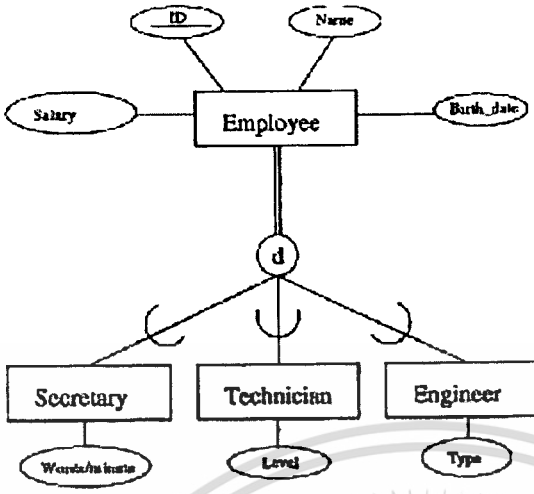
- Key and Constraint ในSurrogate based relational model กรณี E-Relation ที่เกิดจาก Non-Temporal Entity ก็คือค่าของ Attribute เดียวที่ปรากฏอยู่ สำหรับ E-Relation ที่เกิดจาก Temporal Entity ค่าคีย์เกิดจากค่าของ Attribute ที่ระบุด้วย "ϕ" รวมกับ Time-Attribute กรณี A-Relation ค่าคีย์ของA-Relation ที่เกิดจาก Non-Temporal Attribute ก็คือค่า Foreign Key ที่อ้างอิงจาก E-Relation ที่ A-Relation ขึ้นอยู่ด้วย สำหรับ A-Relation ที่เกิดจาก Temporal Attribute ค่าคีย์ก็คือค่า Foreign Key ที่อ้างอิงจาก E-Relation ที่ A-Relation ขึ้นอยู่ด้วย รวมกับ Time-Attribute ทั้ง E-Relation และ A-Relation Attribute ใดที่เป็นคีย์ของรีเลชันนั้นระบุโดยขีดเส้นบน Attributeนั้น สำหรับ Attribute ที่เป็น Foreign Key ระบุโดยใส่ "f.k."ตามหลัง Attribute นั้น และ ระบุโดยขีดใต้ที่ Attribute นั้น

ขั้นตอนการแปลง Time ER Model เป็น Surrogate base relational model ประกอบด้วยขั้นตอนดังนี้

1 ขั้นตอนการแปลง Entity Type (Mapping of entity type) ประกอบด้วย 3 ขั้นตอน

- ขั้นตอนการแปลง Regular Entity type
- ขั้นตอนการแปลง Weak Entity
- ขั้นตอนการแปลงEntity Type Participating in Superclass / Subclass Relationship Types

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.4 Time ER Diagram แสดง Superclass /Subclass Relationship With Disjoint and Optional

รีเลชันที่ได้จากการแปลงจากTimeER Diagram ประกอบด้วย

Employee — Employee_ID_Name_Birth_date_Salary

Employee ϕ	Employee ϕ f.k.	ID u.k.	Name	Birth_date	Salary

Secretary — Secretary_Words/minute

Employee ϕ	Employee ϕ f.k.	Words/minute

Technician — Technician_Level

Employee ϕ	Employee ϕ f.k.	Level

Engineer — Engineer_Type

Employee ϕ	Employee ϕ f.k.	Type

2 ขั้นตอนการแปลง Attribute ของ Entity Type พิจารณาได้สองกรณีดังนี้

- กรณี Non-Attribute Types กรณีนี้ A-Relation ที่ได้ ประกอบไปด้วย Attribute ที่เป็น Non-Temporal โดยที่คีย์ของA-Relation ก็คือ Attribute ที่ลงท้ายด้วย"ϕ" ของE-Relation ที่ A-Relation ขึ้นอ้างอิงอยู่ กรณีที่Non-Temporal Attribute เป็นMultivalued Attribute ก็จะทำการแยกออกเป็น A-Relation อีกรีเลชันหนึ่ง

- กรณี Temporal Attribute Types การสร้าง A-Relation สำหรับ Temporal Attribute ค่าของ Attribute ที่เป็น Time-Attribute จะเป็นชนิดใดขึ้นอยู่กับการระบุประเภทของ Time-Attribute ในTime ER Model ดังนั้นค่าคีย์ของA-Relation กรณีนี้ก็คือค่า Attribute ที่ลงท้ายด้วย "ϕ"ของE-Relation ที่A-Relation อ้างถึงอยู่ รวมกับค่าของTime-Attribute ที่ระบุดังข้อกำหนดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

a) * = VT ดังนั้น T = $\underline{VT_S}$, VT_E

b) * = TT ดังนั้นT = $\underline{TT_S}$, TT_E

c) * = BT ดังนั้น T = $\underline{TT_S}$,TT_E , $\underline{VT_S}$, $\underline{VT_E}$

3. การแปลง Relationship Types พิจารณาได้สองกรณีดังนี้

- กรณี Non-Temporal Binary Relationship type
- กรณี Temporal n-ary Relationship Types

4. การพิจารณา Temporal Constraint ทุกๆเงื่อนไขของเวลาที่จะนำไปใช้กับรีเลชันต่างๆ ที่จะสร้างขึ้นมา ซึ่งประกอบด้วย
   - เงื่อนไขภายใต้ A-Relation
   - เงื่อนไขที่อยู่ระหว่าง A-Relation และ E-Relation
   - เงื่อนไขที่อยู่ระหว่าง E-Relation

ขั้นตอนแปลง Surrogate-based relational model เป็น Lexically-based relational ขั้นตอนในการแปลง Surrogate based relational model เป็น Lexically-based relational ประกอบด้วยสองขั้นตอนดังนี้

ขั้นตอนที่1 นำE-Relation และ A-Relation มาเป็นInput

ขั้นตอนที่2 นำE-Relation และ A-Relation จากขั้นตอนที่1 แปลงมาเป็น Lexically-based relational โดยพิจารณาดังนี้

- กรณี E-Relation แปลงมาเป็น Lexically-based relational โดยนำ Attribute ที่เป็นคีย์ใน Time ER Model มาเป็น Primary Key ของรีเลชัน
- กรณี A-Relation แปลง Attribute ที่เป็น Foreign Key และ Attribute ที่ระบุค่าเป็น "U.K." เป็น Foreign Key

1.5 ขอบเขตการวิจัย

งานวิจัยนี้ ได้กำหนดขอบเขตในการทำงาน โดยแบ่งออกเป็นข้อ ๆ ได้ดังนี้

1. ศึกษาระบบฐานข้อมูลเชิงเวลา
2. ศึกษาการออกแบบฐานข้อมูล NIAM
3. สร้างแบบจำลอง TNIAM ในการออกแบบฐานข้อมูลเชิงเวลา
4. สร้างกระบวนการ ในการแปลงแบบจำลองข้อมูล TNIAM ให้เป็นโครงสร้างฐานข้อมูลเชิงสัมพันธ์ โครงสร้างฐานข้อมูลเชิงสัมพันธ์ที่อยู่ในรูปของ Temporal Database และโครงสร้างฐานข้อมูลเชิงสัมพันธ์ที่อยู่ในรูปของ Object Relational Database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. นำแบบจำลอง TNIAM มาประยุกต์ใช้กับระบบงานจริง
6. นำโครงสร้างของฐานข้อมูลที่ได้จากขั้นตอนที่ 5 พัฒนาเป็นแอปพลิเคชัน

1.6 โครงสร้างวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ กล่าวถึงทฤษฎีที่เกี่ยวข้องกับงานวิจัยดังรายละเอียดดังนี้ บทที่ 2 กล่าวถึงทฤษฎีฐานข้อมูลเชิงเวลา และทฤษีไนแอม บทที่ 3 กล่าวถึงวิธีการออกแบบฐานข้อมูลเชิงเวลาด้วยวิธีการทีไนแอม และ กระบวนการแปลง ทีไนแอมเป็นโครงสร้างฐานข้อมูลเชิงสัมพันธ์ โครงสร้างฐานข้อมูลเชิงสัมพันธ์ที่อยู่ในรูปของ Temporal Database และ โครงสร้างฐานข้อมูลเชิงสัมพันธ์ที่อยู่ในรูปของ Object Relational Database บทที่ 4 อธิบายถึงกระบวนการทำงานของ Temporal Database Operation บทที่ 5 กล่าวถึงการนำทีไนแอมมาประยุกต์ใช้กับระบบงานจริง และ บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ

1.7 ประโยชน์ที่คาดว่าจะได้รับ

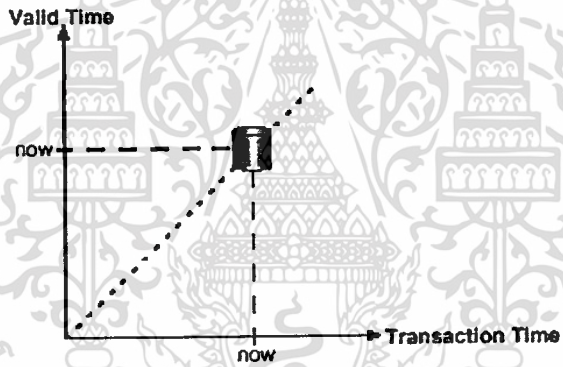
1. มีรูปแบบของการออกแบบฐานข้อมูลเชิงเวลาที่ง่ายต่อการเข้าใจ เนื่องจากมีรูปแบบที่แสดงความสัมพันธ์ของข้อมูลเป็นแบบจำลองที่มีความหมาย และมีเครื่องหมายแสดงความสัมพันธ์ของข้อมูลและข้อจำกัดของข้อมูลได้อย่างชัดเจน
2. มีกระบวนการในการออกแบบทั้งในระดับแนวคิดที่จะได้ Conceptual Database Schema และกระบวนการแปลงConceptual Database Schema เป็นRelational Database Schema ที่อยู่ในรูป Fifth Normal Form ได้โดยตรง
3. สามารถนำไปใช้ได้กับการออกแบบฐานข้อมูลแบบเดิมและการออกแบบฐานข้อมูลเชิงเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

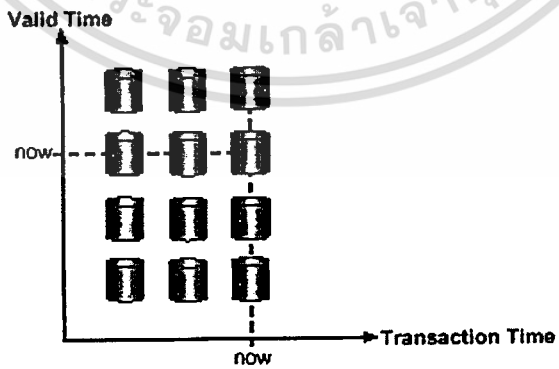
บทที่ 2
ทฤษฎีพื้นฐานที่เกี่ยวข้อง

2.1 ฐานข้อมูลเชิงเวลา (Temporal Database) [11]

ระบบฐานข้อมูลที่ใช้งานกันอยู่ทั่วไปมักจะเป็นระบบฐานข้อมูลเชิงสัมพันธ์ ซึ่งมีการจัดเก็บข้อมูลเพียงสถานะเดียวกันเท่านั้น เมื่อการมีการแก้ไขข้อมูล ข้อมูลเก่าจะถูกลบออกจากฐานข้อมูล และแทนข้อมูลใหม่ลงไปแทน แม้ว่าระบบฐานข้อมูลเดิมนี้จะรองรับโปรแกรมประยุกต์บางโปรแกรมได้ดี แต่ยังไม่มีเพียงพอที่จะใช้งานกับ โปแกรมประยุกต์ที่ต้องการเรียกค้นข้อมูลหลายๆ สถานะ อาทิ โปรแกรมทางการแพทย์ โปรแกรมทางธุรกิจ ดังนั้นเราจึงต้องการระบบฐานข้อมูลที่สามารถเรียกค้นข้อมูลที่แปรผันตามเวลาได้ ซึ่งก็คือระบบฐานข้อมูลที่เก็บข้อมูลหลายๆ สถานะหรือเรียกว่าระบบฐานข้อมูลเชิงเวลา (Temporal Database)นั่นเอง ดังรูปที่ 2.1 และรูปที่2.2



รูปที่ 2.1 การจัดเก็บฐานข้อมูลของระบบฐานข้อมูลปกติ



รูปที่ 2.2 การจัดเก็บข้อมูลของระบบฐานข้อมูลเชิงเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.1 และรูปที่ 2.2 สามารถเห็นความแตกต่างของระบบฐานข้อมูลทั้งสองได้ ข้อมูลที่เก็บอยู่ใน Temporal Database นั้นแตกต่างจากข้อมูลที่ถูกเก็บอยู่ในฐานข้อมูลปกติ ตรงที่ระบบฐานข้อมูลเชิงเวลาเก็บช่วงเวลาไว้กับข้อมูล(Timestamp) เพื่อแสดงว่าข้อมูลถูกเก็บในฐานข้อมูลเมื่อใดหรือข้อมูลเป็นจริงในช่วงเวลาใด โดยเวลาที่เก็บใน Temporal Database มี 3 แบบคือ

2.1.1 Valid Time คือ เวลาที่ข้อมูลเป็นจริง ซึ่งจะแบ่งแยกย่อยออกเป็น

2.1.1.1 Valid Time Start คือ เวลาเริ่มต้นที่ข้อมูลนั้นเป็นจริง

2.1.1.2 Valid Time End คือ เวลาสุดท้ายที่ข้อมูลนั้นเป็นจริง

2.1.2 Transaction Time คือเวลาที่ข้อมูลถูกเก็บในฐานข้อมูล โดยสามารถแบ่งออกเป็น

2.1.2.1 Transaction Time Start คือ เวลาที่เริ่มจัดเก็บข้อมูลลงในฐานข้อมูล

2.1.2.2 Transaction Time End คือ เวลาที่แก้ไขหรือลบข้อมูล

2.1.3 User Defined Time คือ เวลาที่ผู้ใช้งานสร้างขึ้นเพื่อ ใช้งาน ซึ่งผู้ใช้งานต้องจัดการและเรียกค้นเอง ระบบฐานข้อมูลจะไม่จัดการให้อย่างอัตโนมัติ หรือกล่าวอีกนัยหนึ่งว่า เวลาชนิดนี้จะเป็นเวลาที่เป็นจริงเสมอ เช่น วันเกิด วันเข้าทำงาน และวันที่เกษียณ เป็นต้น

สำหรับตัวอย่างของ Valid Time และ Transaction Time นั้นคือ ถ้านาย สมชาย รักการดี เป็นนักศึกษาตั้งแต่วันที่ 1 มิถุนายน ค.ศ. 1994 ถึงวันที่ 30 พฤษภาคม ค.ศ. 1998 แต่ข้อมูลนี้ถูกป้อนเข้าฐานข้อมูลในวันที่ 20 มิถุนายน ค.ศ. 1994 โดยที่ข้อมูลนี้ถูกลบออกจากฐานข้อมูลในวันที่ 25 มิถุนายน ค.ศ. 1998 เราสามารถแจกแจงเวลาได้ดังนี้

- Valid Time Start คือ วันที่ 1 มิถุนายน ค.ศ. 1994
- Valid Time End คือ วันที่ 30 พฤษภาคม ค.ศ. 1998
- Transaction Time Start คือ วันที่ 20 มิถุนายน ค.ศ. 1994
- Transaction Time End คือ วันที่ 25 มิถุนายน ค.ศ. 1998

สังเกตว่าช่วงเวลา 3 ช่วงนี้ไม่จำเป็นต้องเป็นช่วงเวลาเดียวกัน ตัวอย่างเช่น สมมติว่ามี Temporal Database ที่เก็บข้อมูลของศตวรรษที่ 18 valid Time ของข้อมูลคือช่วงเวลาใดๆ ระหว่างปี 1700 และ 1799 ส่วน Transaction Time จะเป็นเวลาที่เราเก็บข้อมลเข้าไปในฐานข้อมูล ถ้า Valid Time Start เป็นอดีตแต่ข้อมูลที่ใส่เข้าไปยังไม่ทราบ Valid Time End เราจึงกำหนดให้ Valid Time End เป็น NOW และให้ Valid Time End เป็น NULL เมื่อ Valid Time Start เป็นอนาคต

$$T_E := \text{NOW}, \text{ if } T_S \leq \text{NOW}$$

$$T_E := \text{NULL}, \text{ if } T_S > \text{NOW}$$

จะเห็นได้ว่า ช่วงเวลา Valid Time และช่วงเวลา Transaction Time ไม่จำเป็นต้องเป็นช่วงเวลาเดียวกันสำหรับข้อมูลตัวเดียวกัน ตัวอย่างเช่น ฐานข้อมูลเก็บชื่อพนักงานคนหนึ่งไว้ตั้งแต่ปี 2540 หากพนักงานคนนี้มีการเปลี่ยนแปลงชื่อในปี 2543 แต่มาตรวจพบว่าชื่อมีการเปลี่ยนแปลงในปี 2544 จะทำให้ Valid Time กับ Transaction Time เป็น 2543 และ 2544 ตามลำดับ ซึ่งมีค่าไม่เท่ากัน

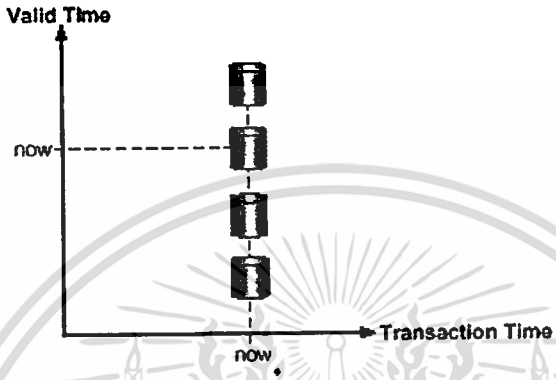
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ประเภทของฐานข้อมูลเชิงเวลา [11]

ช่วงเวลา Valid Time และ Transaction Time จะทำให้เกิดฐานข้อมูลในหลายประเภท ดังนี้

2.2.1 ฐานข้อมูลวาลิดไทม์ (Valid Time Database)

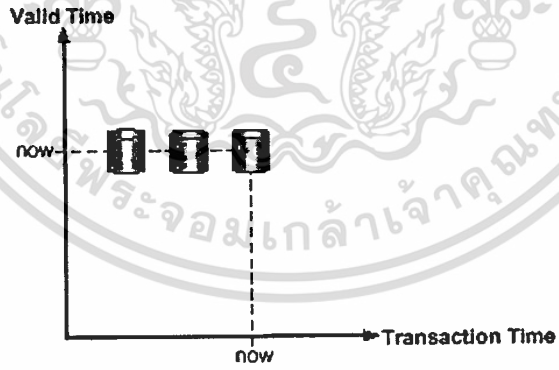
ฐานข้อมูลประวัติ (Historical Database) จะเก็บเฉพาะวาลิดไทม์ ซึ่งจะเก็บข้อมูลในอดีตกับปัจจุบัน แสดงดังรูปที่ 2.3



รูปที่ 2.3 แสดงฐานข้อมูลประวัติ

2.2.2 ฐานข้อมูลทรานส์แอกชันไทม์(Transaction Time)

ฐานข้อมูลย้อนกลับ (Rollback Database) ซึ่งจะเก็บเฉพาะทรานส์แอกชันไทม์ แสดงดังรูปที่ 2.4



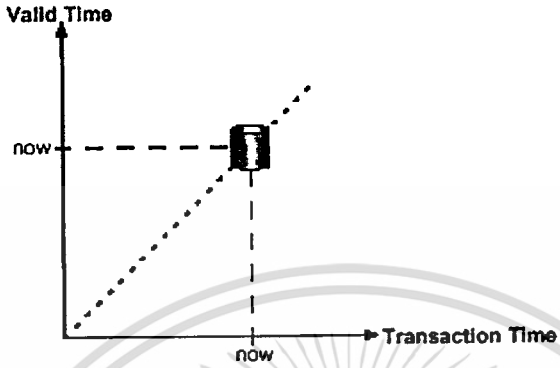
รูปที่ 2.4 ฐานข้อมูลย้อนกลับ

2.2.3 ฐานข้อมูลไบเทมโพรอล (Bitemporal Database)

เป็นฐานข้อมูลที่ เก็บทั้งวาลิดไทม์และทรานส์แอกชันไทม์ จึงทำให้สามารถเก็บข้อมูลในอดีต อนาคต และปัจจุบันได้

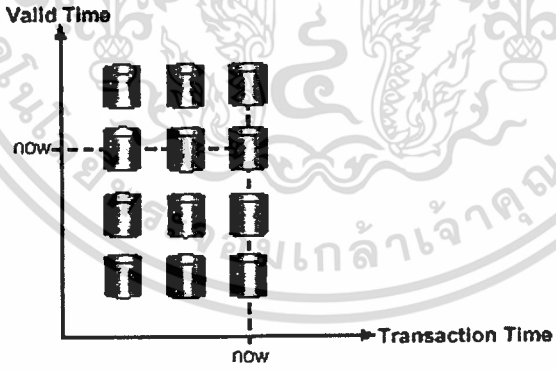
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในระบบจัดการฐานข้อมูลเชิงพานิชย์จะเก็บเฉพาะข้อมูลที่เป็นจริงในปัจจุบันเท่านั้น ฐานข้อมูลชนิดนี้จะเรียกว่าฐานข้อมูลสแนปช็อท (Snapshot Database) โดยฐานข้อมูลสแนปช็อทในแบบของวาลิดไทม์และทรานส์แอกชันไทม์ แสดงดังรูปที่ 2.5



รูปที่ 2.5 ฐานข้อมูลสแนปช็อท

ฐานข้อมูลแบบไบเทมโพรอล อาจจะเป็นได้ทั้งตารางสแนปช็อท คือเก็บตารางวาลิดไทม์ คือเก็บว่าข้อมูลเป็นจริงเมื่อใด โดยมีลักษณะฐานข้อมูล หรือเก็บตารางทรานส์แอกชันไทม์ คือ เก็บว่าข้อมูลถูกบันทึกลงในฐานข้อมูลเมื่อใด หรืออาจจะเป็นตารางไบเทมโพรอล ที่เก็บทั้งวาลิดไทม์และทรานส์แซคชันไทม์ก็ได้ โดยมีลักษณะของฐานข้อมูลดังรูปที่ 2.6



รูปที่ 2.6 ฐานข้อมูลไบเทมโพรอล

2.3 แบบจำลองข้อมูลไนแอม (The NIAM Conceptual Model)

แบบจำลองระดับแนวคิดในแอม [6] นั้น คำว่า NIAM ย่อมาจาก Nijssen's Information Analysis Methodology ซึ่งเป็นรูปแบบหนึ่งในระดับแนวคิด ที่ได้มีการคิดค้นมาตั้งแต่ราวปลายปี ค.ศ. 1977 โดย ศาสตราจารย์ จี เอ็ม ไนเซน ได้เสนอแบบจำลองข้อมูล ไนแอม ซึ่งเป็นแบบจำลองที่มีพื้นฐานมาจาก โครงสร้างภาษาธรรมชาติ ที่ใช้รูปประโยคที่มี ประธาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กริยา และกรรม เป็นแบบจำลองระดับแนวคิด ที่มีความหมายและเครื่องหมาย แสดงความสัมพันธ์ของข้อมูล และข้อจำกัดของข้อมูลได้อย่างชัดเจน นอกจากนี้แล้วยังสามารถแปลงแบบจำลองดังกล่าว เป็นโครงสร้างของฐานข้อมูลเชิงสัมพันธ์ (Relational Database Schema) ซึ่งจะได้ผลลัพธ์ที่อยู่ในรูปของ Fifth Normal Form ดังนั้น จึงมีการนำแบบจำลองระดับแนวคิดไนแอม มาใช้ช่วยในการออกแบบฐานข้อมูลเชิงสัมพันธ์กันอย่างแพร่หลาย เพราะมีการใช้สัญลักษณ์ที่แสดงความสัมพันธ์ของข้อมูลอย่างชัดเจน และง่ายต่อการเข้าใจ

2.3.1 ส่วนประกอบของ ไนแอม

แบบจำลองข้อมูล ไนแอม มีส่วนประกอบพื้นฐานดังนี้

2.3.1.1 Entity Type

Entity Type คือกลุ่มของนามธรรม หรือ Entity จริง ๆ ซึ่งอาจเป็นสิ่ง ที่จับต้องได้ หรือ จับต้องไม่ได้ เช่น PERSON, DEPARTMENT, COMPANY

2.3.1.2 Lable Type

Label Type คือ ตัวที่ใช้บอกความแตกต่าง ของ Entity ภายใน Entity Type (ใช้เป็นคีย์)

2.3.1.3 Elementary Fact Type

Elementary Fact Type คือ ตัวที่ใช้แสดงความสัมพันธ์ ระหว่าง Entity Type 2 Entity Type ขึ้นไปสัญลักษณ์ที่ใช้แทน เป็นรูปสี่เหลี่ยมผืนผ้าติดกันสองรูปขึ้นไป ซึ่งก็ขึ้นอยู่กับ Entity Type ที่แสดงความสัมพันธ์กันอยู่ว่าประกอบไปด้วย Entity Type เท่าใด

2.3.1.4 Subtype

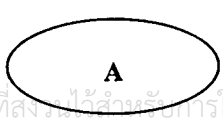
Subtype คือ ตัวที่ใช้ในการกำหนด โครงสร้างลำดับที่ซับซ้อน ของ Entity Type และ Subtype มีการถ่ายทอด Fact Type จาก Supertype ด้วย

2.3.1.5 Constraint

Constraint คือตัวที่ใช้เป็นเงื่อนไขที่จะทำให้แน่ชัดว่า ฐานข้อมูลนั้นถูกต้องและสมบูรณ์ ตลอดเวลา (Consistency and Integrity) Constraint ที่ใช้ใน ไนแอม มีดังนี้

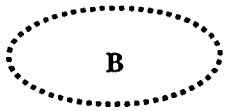
1) Uniqueness Constraint ประกอบด้วย Inter Fact Type และ Intra Fact Type
2) Mandatory Constraint
3) Exclusion Constraint
4) Equality Constraint
5) Subset Constraint

2.3.2 สัญลักษณ์ที่ใช้ใน ไนแอม

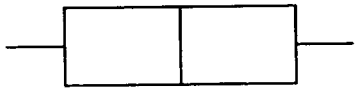


แทน Entity Type A

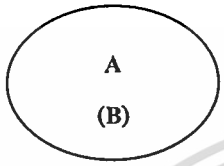
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แทน Label Type B



แทน Binary Fact Type



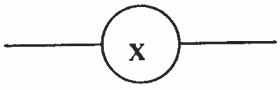
แทน Entity Type A และมี Label Type B

แทน Entity Type C เป็น Subtype ของ Entity Type D ซึ่งเป็น Suppertype

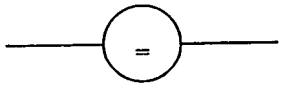
แทน Mandatory Fact Constraint
(ต้องการ Mandatory ที่ Entity Type ไหน ให้ระบุ " ● " ที่ Entity Type นั้น)

แทน Intra Fact Type Uniqueness Constraint

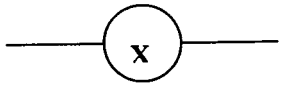
แทน Inter Fact Type Uniqueness Constraint



แทน Exclusion Constraint



แทน Equality Constraint



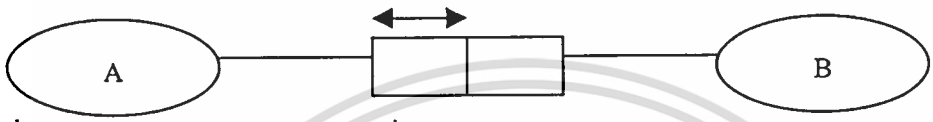
แทน Subset Constraint

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 กฎข้อบังคับความถูกต้องของข้อมูลที่ใช้ในแบบจำลองระดับแนวคิดไนแอม [13]

2.3.3.1 Intra fact type uniqueness constraints (internal uniqueness constraints) เป็นกฎข้อบังคับความถูกต้อง เพื่อทำการกำหนดบทบาท ที่ใช้แสดงเอกลักษณ์ของ ความสัมพันธ์ใดๆ โดยจะกำหนดความสัมพันธ์ระหว่างสมาชิกของ Entity Type หนึ่ง กับสมาชิกของ Entity Typeอื่น หรือกับสมาชิกของ Label Type โดยสามารถกำหนดออกเป็นรูปแบบต่างๆ ได้ดังต่อไปนี้

1) การกำหนดความสัมพันธ์แบบหนึ่งต่อหลายหน่วย (one to many relationship) ซึ่งสามารถแสดงบนแผนภาพได้ดังรูปที่ 2.7



รูปที่ 2.7 แสดงความสัมพันธ์แบบหนึ่งต่อหลายหน่วย

ลักษณะเช่นนี้ สามารถแสดงได้ว่า Entity Type A จะแสดงความสัมพันธ์กับ Entity Type หรือ Label Type B ได้อย่างมากที่สุดเพียงหนึ่งความสัมพันธ์เท่านั้น แต่ในทางกลับกัน Entity Type B หรือ Label Type B จะแสดงความสัมพันธ์ กับ Entity Type A ได้หลายความสัมพันธ์ โดยกฎข้อบังคับความถูกต้อง จะต้องทำการควบคุมไม่ให้เกิดการซ้ำซ้อนของข้อมูลในคอลัมน์ A ขึ้นได้เช่น คนหนึ่งคนจะมีมารดาได้เพียงคนเดียวเท่านั้น แต่ในทางกลับกัน คนเพียงคนเดียวอาจเป็นมารดาของคนหลายคนได้

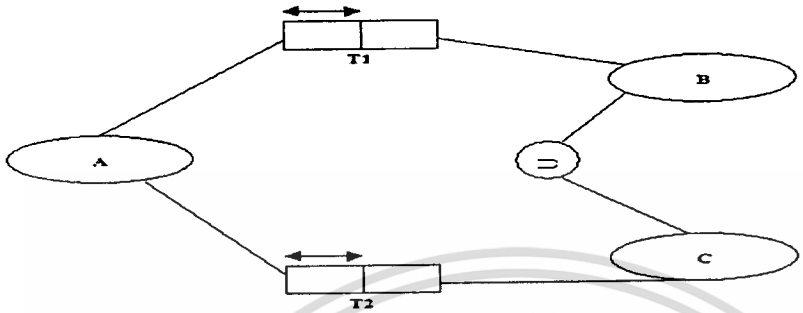
2) การกำหนดความสัมพันธ์ แบบหนึ่งต่อหนึ่งหน่วย (one to one relationship) ซึ่งสามารถแสดงบนแผนภาพได้ดังรูปที่ 2.8

รูปที่ 2.8 แสดงความสัมพันธ์แบบหนึ่งต่อหนึ่งหน่วย

ลักษณะเช่นนี้สามารถแสดงได้ว่า Entity Type A จะแสดงความสัมพันธ์ กับ Entity Type B ได้หลายความสัมพันธ์ และในทางกลับกัน ชนิด Entity Type B ก็จะแสดงความสัมพันธ์กับ Entity Type A ได้หลายความสัมพันธ์เช่นกัน โดยกฎข้อบังคับความถูกต้องจะต้องทำการควบคุมความสัมพันธ์ A และ B ไม่ให้เกิดความซ้ำซ้อนขึ้นได้ เช่น นักศึกษาหนึ่งคน อาจจะลงทะเบียนเรียนได้หลายวิชา และวิชาใดๆ ก็สามารถรองรับนักศึกษาได้หลายคน แต่นักศึกษาหนึ่งคนจะไม่สามารถลงทะเบียนวิชาใดๆ ได้มากกว่า 1 ครั้งของการลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3.2 Inter fact type uniqueness constraints (external uniqueness constraints) เป็นกฎข้อบังคับความถูกต้องที่แสดงให้เห็นว่า Entity Type ใดๆ มีความสัมพันธ์กับ Label Type หรือ Entity Type ได้มากกว่าหนึ่ง โดยในทางกลับกัน Label Type หรือ Entity Type เหล่านั้นสามารถบ่งบอก ถึงลักษณะเฉพาะของ Entity Type นั้นได้ ดังแสดงในรูปที่ 2.9



รูปที่ 2.9 แสดง Inter fact type uniqueness constraints

ลักษณะเช่นนี้สามารถแสดงได้ว่า กฎข้อบังคับความถูกต้องจะทำการควบคุม หากนำ T1 join T2 แล้วผลที่ได้ BC จะไม่เกิดความซ้ำซ้อน เช่น คนหนึ่งคนอาจมีชื่อหรือนามสกุลซ้ำกันได้ แต่ถ้า รวมทั้งชื่อและนามสกุลแล้ว จะไม่เกิดความซ้ำซ้อน ดังนั้นจะสามารถบ่งบอกได้ว่า เป็นการระบุถึงคนใด

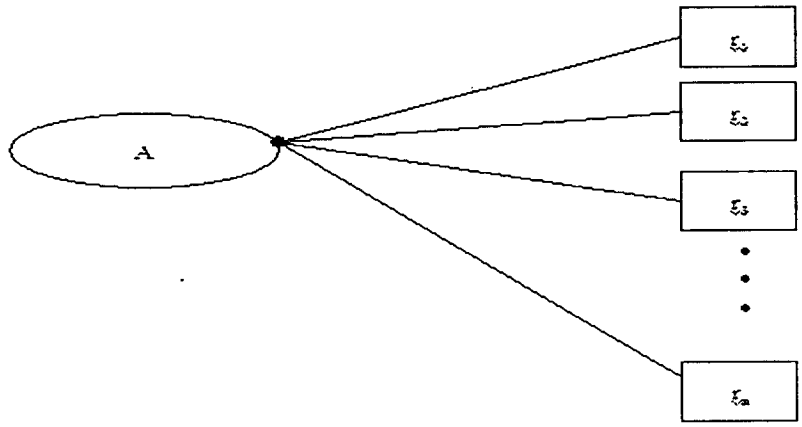
2.3.3.3 Mandatory role constraints เป็นกฎข้อบังคับความถูกต้อง ที่ใช้ในการควบคุมเพื่อแสดงให้เห็นถึง การมีอยู่ของข้อมูล ว่าต้องมีการบันทึกข้อมูลทุกครั้งที่เกิดมีความสัมพันธ์เกิดขึ้น แสดงได้ดังรูปที่ 2.10

รูปที่ 2.10 แสดง Mandatory role constraints

จากรูป จุดทึบที่เชื่อมต่อระหว่าง Entity Type กับ Role นั้น แสดงให้เห็นว่า สมาชิกทุกๆสมาชิกใน Entity Type A จะต้องถูกบันทึกข้อมูลเมื่อมีบทบาท r เกิดขึ้น โดยแสดงให้เห็นว่า pop (A) = pop(r) เช่น นักศึกษาทุกคนต้องมีการบันทึกชื่อ และ นามสกุล เป็นต้น

2.3.3.4 Inclusion mandatory constraints เป็นกฎข้อบังคับ ที่แสดงให้เห็นถึง ทางเลือกของบทบาท ในกลุ่มของความสัมพันธ์ ที่มีอยู่ว่า ต้องมีการบันทึกข้อมูลอย่างน้อยบทบาทใดบทบาทหนึ่ง ของ Entity Type นั้น ดังแสดงในรูปที่ 2.11

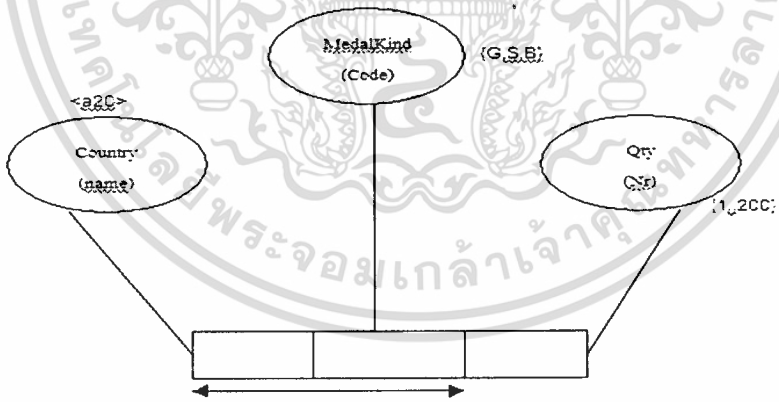
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 แสดง Inclusion mandatory role constraints

จากรูปสามารถแสดงกฎข้อบังคับ ความถูกต้องของข้อมูล คือสมาชิกของชนิด Entity Type A ใด ๆ ต้องมีการบันทึกความสัมพันธ์เกิดขึ้นความสัมพันธ์ใดความสัมพันธ์หนึ่ง ซึ่งแสดงได้ว่า $pop(A) = pop(r_1) \cup pop(r_2) \cup pop(r_3) \cup \ldots \cup pop(r_n)$ เช่น บุคคลใด ๆ จะต้องมีการระบุข้อมูลของบุตร หรือ ข้อมูลของบิดามารดาของบุคคลนั้นๆ อย่างน้อยที่สุดหนึ่งข้อมูล

2.3.3.5 Entity type constraints (Value constraints) เป็นกฎข้อบังคับความถูกต้อง ที่ใช้ในการกำหนดค่า ของสมาชิกภายในเซ็ตของข้อมูล ที่เป็นไปได้ของ Label Type หรือ Entity Type หนึ่ง ๆ รวมไปถึงการกำหนด ชนิดของข้อมูลในเซ็ตด้วย ดังแสดงในรูปที่ 2.12

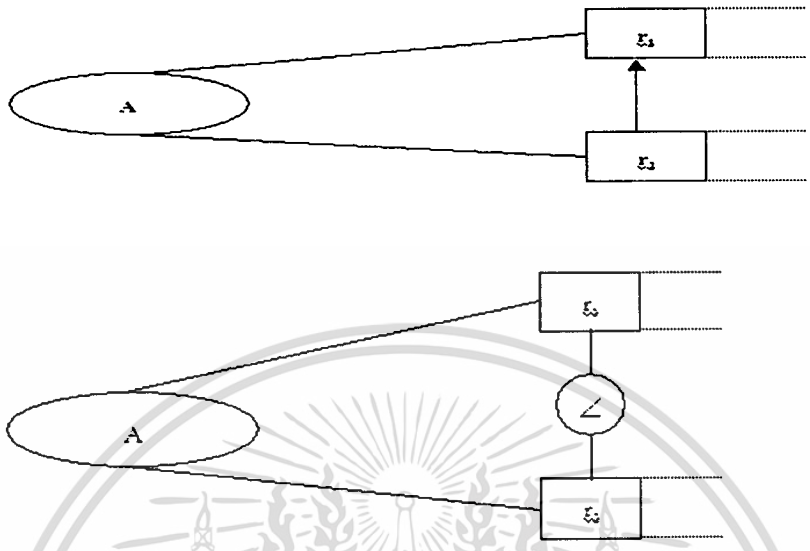


รูปที่ 2.12 แสดง Entity type constraints

จากรูปนั้นจะมีกฎข้อบังคับความถูกต้องของข้อมูล เพื่อทำการระบุชนิดของเหรียญรางวัล ในการแข่งขันกีฬาสามารถแยกออกได้เป็น เหรียญทองแดง , เหรียญเงิน , เหรียญทอง และระบุถึงจำนวนของเหรียญรางวัลที่ได้ว่า ต้องอยู่ในช่วง 1 ถึง 200 เหรียญ รวมทั้งยังสามารถระบุชนิดของข้อมูลได้ด้วย ดังที่แสดงให้เห็นว่า ชื่อประเทศนั้นกำหนดให้จัดเก็บได้มากที่สุด 20 ตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

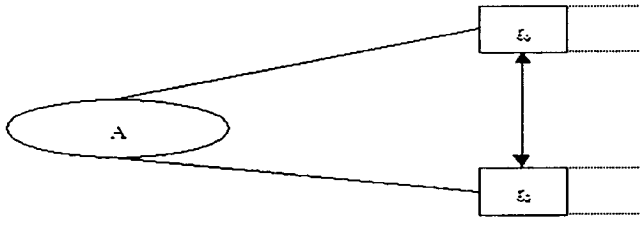
2.3.3.6 Subset constraints เป็นกฎข้อบังคับความถูกต้องของข้อมูล ที่แสดงความสัมพันธ์ที่เป็นส่วนหนึ่งของความสัมพันธ์ที่มีอยู่ แต่จะมีลักษณะความสัมพันธ์ไปในทางเดียว ดังแสดงความสัมพันธ์ได้โดย ใช้สัญลักษณ์คือ A → B ซึ่งสามารถแสดงได้ดังรูปที่ 2.13



รูปที่ 2.13 แสดง Subset constraints

ลักษณะดังกล่าวนี้ แสดงให้เห็นกฎข้อบังคับความถูกต้องของข้อมูลว่า สมาชิกแต่ละตัว ของ Entity Type A หากมีการบันทึกความสัมพันธ์ r_2 แล้วต้องมีการบันทึกความสัมพันธ์ r_1 ด้วย แต่ในทางกลับกัน สมาชิกแต่ละตัวของ Entity Type A หากมีการบันทึกความสัมพันธ์ r_1 แล้ว ไม่จำเป็นต้องมีการบันทึกความสัมพันธ์ r_2 ก็ได้ ดังแสดงได้ว่า $pop(r_2) \subseteq pop(r_1)$ เช่น บุคคลที่ชนะการแข่งขันกีฬา แสดงว่าต้องเป็นนักกีฬา แต่ผู้ที่เป็นนักกีฬาไม่จำเป็นต้องเป็นผู้ชนะการแข่งขันทุกคน

2.3.3.7 Equality constraints เป็นกฎข้อบังคับความถูกต้อง ที่แสดงให้เห็นว่า Entity Type เหล่านั้นจะต้องมีการถูกบันทึกข้อมูลควบคู่กันเสมอไป ใช้สัญลักษณ์แสดงความสัมพันธ์ได้คือ A↔ B ซึ่งสามารถแสดงได้ดังรูปที่ 2.14

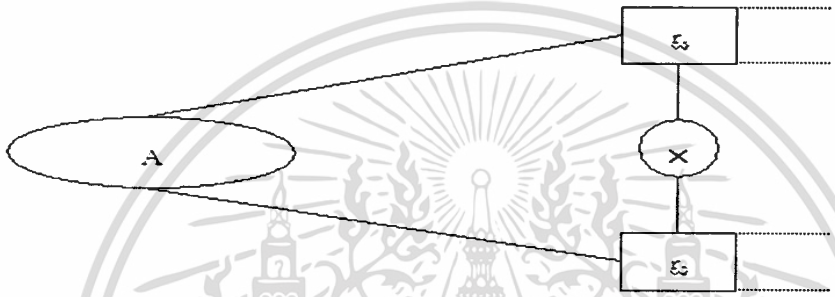


รูปที่ 2.14 แสดง Equality constraints

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะดังกล่าวนี้ สามารถแสดงถึงกฎข้อบังคับความถูกต้องของข้อมูลว่า หากมีการบันทึกข้อมูลความสัมพันธ์ r_1 ก็ต้องมีการบันทึกข้อมูลความสัมพันธ์ r_2 ของสมาชิกของ Entity Type A ด้วย แสดงได้ว่า $pop(r_1) = pop(r_2)$ เช่นหากบุคคลใดจะทำการบันทึกระยะเวลาของการออกกำลังกาย ก็จะต้องทำการบันทึกข้อมูลของอัตราการเต้นของหัวใจด้วย และในทางกลับกัน หากมีการบันทึกข้อมูลอัตราการเต้นของหัวใจ ก็จะต้องทำการบันทึกข้อมูลระยะเวลาการออกกำลังกายด้วยเช่นกัน

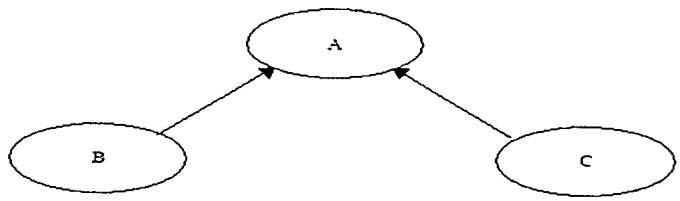
2.3.3.7 Exclusion constraints เป็นกฎข้อบังคับความถูกต้อง ที่มีลักษณะตรงกันข้ามกับ Equality constraints คือแสดงความสัมพันธ์ที่ระบุว่า หากมีความสัมพันธ์แบบหนึ่งเกิดขึ้น จะต้องไม่มีความสัมพันธ์อีกแบบหนึ่งเกิดขึ้นโดยเด็ดขาด ซึ่งสามารถแสดงได้ดังรูปที่ 2.15



รูปที่ 2.15 แสดง Exclusion constraints

ลักษณะดังกล่าวนี้ แสดงให้เห็นกฎข้อบังคับความถูกต้องว่า หากมีการบันทึกข้อมูลความสัมพันธ์ r_1 ของสมาชิกของ Entity Type A ใด จะต้องไม่มีการบันทึกข้อมูลความสัมพันธ์ r_2 ของสมาชิกของ Entity Type A นั้นโดยเด็ดขาด ดังแสดงได้ว่า $pop(r_1) \cap pop(r_2) = \{\}$ เช่น ถ้าบุคคลใดถูกเลือกให้เป็นกรรมการในการตัดสินเกมส์นั้น บุคคลนั้นจะไม่มีสิทธิ์เป็นผู้แข่งขันในเกมส์อย่างเด็ดขาด

2.3.3.8 Sub type constraint เป็นกฎข้อบังคับความถูกต้องที่ระบุถึงการแบ่งกลุ่มของสมาชิกของ Entity Type ที่มีอยู่อย่างชัดเจน ซึ่งสมาชิกของ Entity Type ที่แบ่งแยกออกจาก Entity Type ที่เป็น Super type นั้น จะต้องมีลักษณะและคุณสมบัติที่แตกต่างกันอย่างชัดเจน ดังแสดงในรูปที่ 2.16



รูปที่ 2.16 แสดง Sub Type constraints

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะดังกล่าวนี้ แสดงให้เห็นว่า สมาชิกของ Entity Type A โดยจะเรียกว่า Super Type นั้น สามารถแบ่งออกเป็น 2 กลุ่ม คือ กลุ่มของ Entity Type B และ กลุ่มของ Entity Type C ซึ่งเรียกว่า Sub Type เช่น Entity Type ของบุคคล สามารถแบ่งออกเป็น Sub Type ผู้ชาย และ ผู้หญิงได้

2.3.4 กระบวนการในการแปลงแบบจำลองระดับแนวคิดไนแอม ให้อยู่ในรูปของโครงสร้างฐานข้อมูลเชิงสัมพันธ์ [13]

ประกอบด้วยขั้นตอนดังนี้

1. เปลี่ยนทุกๆชนิดความจริงที่มี 1 บทบาท (unary elementary fact type) ให้อยู่ในรูปของชนิดความจริงแบบ 2 บทบาท
2. ทำการย้ายชนิดความจริงทั้งหมดที่เชื่อมต่อกับ Super Type มาเชื่อมต่อกับ Sub Type ของ Super Type นั้น ๆ แทน
3. สร้างรีเลชันสำหรับแต่ละประโยคมูลฐาน ซึ่งมี intra uniqueness constraints กำกับอยู่บนทุกบทบาท โดยนำแต่ละ Entity Type ที่เกี่ยวข้องกับชนิดความจริงนั้น ๆ มาสร้างเป็น Attribute ของรีเลชันนั้น ๆ ทุก Attribute ที่สร้างขึ้นจะถูกกำหนดให้เป็น mandatory และ Attribute ทั้งหมดที่ถูก intra uniqueness constraints กำกับอยู่นั้นให้ทำการกำหนดเป็นคีย์หลัก(Primary Key) ของรีเลชัน
4. สร้างรีเลชันสำหรับแต่ละ n-ary fact type ที่มีมากกว่า 2 บทบาทขึ้นไป ซึ่งมี intra uniqueness constraints กำกับอยู่ n-1 บทบาท โดยนำแต่ละ Entity Type ที่เกี่ยวข้องกับชนิดความจริงนั้น ๆ มาสร้างเป็น Attribute ของรีเลชันนั้น ๆ ทุก ๆ Attribute ที่สร้างขึ้น จะถูกกำหนดให้เป็น mandatory และ Attribute ทั้งหมดที่ถูก intra uniqueness constraints กำกับอยู่นั้น ให้ทำการกำหนดเป็นคีย์หลัก (Primary Key) ของรีเลชัน
5. ทำการจัดกลุ่มชนิดความจริง ที่มีอยู่ 2 บทบาทที่เหลือ โดยมี intra uniqueness constraints กำกับอยู่ 1 บทบาท อยู่ฝั่งเดียวกัน สำหรับ กรณีที่มี intra uniqueness constraints กำกับอยู่ 1 บทบาททั้งสองฝั่ง ให้ทำการจัดตามฝั่งที่มี mandatory constraints ที่ปรากฏอยู่ หากไม่ปรากฏอยู่ทั้งสองฝั่งให้ทำการจัดตามลำดับของตัวอักษร
6. สร้างรีเลชันตามกลุ่ม ที่ได้จัดไว้ในขั้นตอนที่ 5 โดยทำการสร้าง Attribute ขึ้นจาก Entity Type ที่มีอยู่ และทำการกำหนด mandatory constraints ตาม Entity Typeนั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

แบบจำลองทีไนแอม (TNIAM Model)

การออกแบบฐานข้อมูลเชิงเวลา ด้วยวิธีการทีไนแอม (TNIAM : Temporal NIAM) เป็นแนวคิด ที่ขยายเพิ่มเติม จากแบบจำลองข้อมูลไนแอม โดยมีกระบวนการออกแบบทั้งในระดับแนวคิด ที่จะได้โครงสร้างในระดับแนวคิด และ กระบวนการแปลงโครงสร้างในระดับแนวคิด เป็นโครงสร้างฐานข้อมูลเชิงสัมพันธ์ ที่อยู่ในรูป นอร์มอลฟอร์มระดับที่ 5 (5NF)

3.1 รูปแบบเวลาที่ใช้ในแบบจำลองทีไนแอม

รูปแบบเวลาที่ใช้ในแบบจำลองทีไนแอม ได้แก่ รูปแบบ Life span, Valid Time, Transaction Time และ User Defined Time

3.1.1 รูปแบบเวลา Lifespan [1]

คือระยะเวลาที่ Entity Type นั้นเกิดขึ้นมาจนกระทั่ง Entity Type นั้นจบชีวิตลง ถ้า Entity Type ใดมีการระบุค่าเวลาของ Entity Type หมายความว่าค่ารูปแบบเวลาของ Entity Type จะถูกจัดเก็บในฐานข้อมูล

3.1.2 รูปแบบเวลา Valid Time [2]

รูปแบบเวลา Valid Time หรือช่วงเวลาที่ข้อมูลเป็นจริง คือ เวลาที่ข้อมูลเกิดขึ้นจริง ๆ ซึ่งทุก ๆ ความจริงจะมีค่าเวลาที่ข้อมูลเป็นจริงอยู่ แต่การพิจารณาว่าจะมีการจัดช่วงเก็บเวลา ที่ข้อมูลเป็นจริงในฐานข้อมูลหรือไม่ พิจารณาได้ 3 กรณี ดังนี้

1) จะทำการจัดเก็บช่วงเวลาที่ข้อมูลเป็นจริง เมื่อสามารถทำการจัดเก็บทุก ๆ ค่าที่เป็นจริงถูกต้อง และคงอยู่ตลอดไปของความจริงของข้อมูลนั้น

2) จะไม่ทำการจัดเก็บช่วงเวลาที่ข้อมูลเป็นจริง เมื่อค่าความจริงของข้อมูลไม่มีการเปลี่ยนแปลง

3) ไม่ทำการจัดเก็บเวลาที่ข้อมูลเป็นจริงของความจริงของข้อมูล เมื่อรู้ค่าที่ถูกต้อง แต่ไม่รู้เวลาที่แน่นอน

3.1.3 รูปแบบเวลา Transaction Time [2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบเวลา Transaction Time หรือช่วงเวลาที่ข้อมูลถูกจัดเก็บลงสู่ฐานข้อมูล คือช่วงเวลาที่เป็นจริง ที่ทำการจัดเก็บข้อมูลลงสู่ฐานข้อมูล โดยที่ช่วงเวลาที่ข้อมูลถูกจัดเก็บลงสู่ฐานข้อมูล จะไม่มีการเปลี่ยนแปลงไปตามลำดับของแต่ละการทำงานที่เกิดขึ้น

3.1.4 รูปแบบเวลา User Defined Time [1]

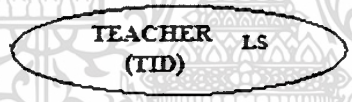
รูปแบบเวลา User Defined Time หมายถึงคุณสมบัติของความจริงที่ไม่ต้องแปลความหมายของวันหรือเวลา อย่างเช่น วันเกิด หรือวันแรงงาน ซึ่งเป็นคุณสมบัติที่ไม่ต้องการการจัดการทำให้ผู้ใช้จะต้องจัดการเอง ไม่ว่าจะเป็น เพิ่ม,ลบ หรือ แก้ไข

3.2 ส่วนประกอบของทีไนแอม

ส่วนประกอบของทีไนแอม ที่ขยายมาจากไนแอม ประกอบด้วย

3.2.1 Temporal Entity Type ประกอบด้วย

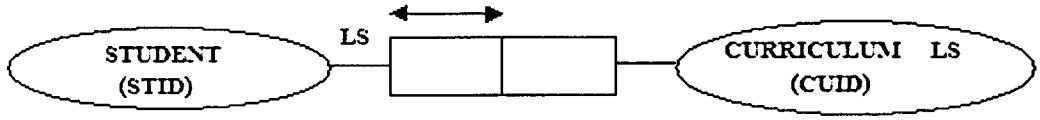
1. Temporal Entity Type ที่ระบุรูปแบบเวลา Life Span ภายใน Entity Type โดยระบุค่า Life span ที่อยู่ในรูปของ "LS" รูปแบบของ Temporal Entity Type แสดงได้ดังรูปที่ 3.1



รูปที่ 3.1 แสดง Temporal Entity Type TEACHER ที่ระบุค่า Life span ไว้ภายใน Entity Type

จากรูปที่ 3.1 อธิบายได้ว่า Life span ที่กำหนดใน Temporal Entity Type TEACHER ก็คือช่วงเวลาของ Life span ของ TEACHER ตั้งแต่ TEACHER เริ่มเข้าทำงานจน TEACHE เกษียณ หรือ หมดสัญญาจ้าง

2. Temporal Entity Type ที่ระบุ Life span อยู่ภายนอก Entity Type โดยระบุที่Fact Type ที่ Temporal Entity Type เกาะติดอยู่ Entity Type ชนิดนี้ เกิดขึ้นเมื่อพบว่า Life span ของ Entity Type ไปขึ้นอยู่กับ Life span ของ Entity Type ที่ไปมีความสัมพันธ์อยู่ด้วย, รูปแบบลักษณะการเกิดของ Temporal Entity Type ชนิดนี้แสดงได้ดังรูปที่ 3.2



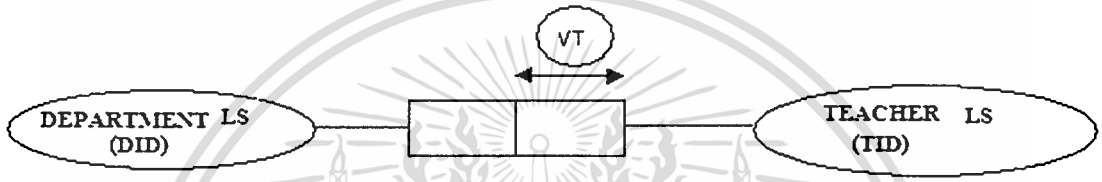
รูปที่ 3.2 แสดง Temporal Entity Type ที่ระบุค่า Life span ไว้ภายนอก Entity Type

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.2 จะพบว่า Entity Type STUDENT ไม่สามารถระบุ Life span ไว้ภายใน Entity Type ได้เหตุผลเพราะ Life span ของ Entity Type STUDENT ไปขึ้นอยู่กับ Life span ของ Temporal Entity Type CURRICURUM (Life span ของนักศึกษาขึ้นอยู่กับ Life span ของหลักสูตรของนักศึกษาแต่ละคน) ดังนั้น Life span ของ Entity Type STUDENT จึงระบุอยู่ภายนอก Entity Type โดยระบุอยู่ที่ Fact Type ที่ Entity Type เกาะติดอยู่

3.2.2 Temporal Fact Type ประกอบด้วย

1 Temporal Fact Type ที่ระบุ Valid Time โดยจะระบุ Valid Time ที่อยู่ในรูปของ "VT" ที่อยู่ในรูปวงกลม รูปแบบของ Fact Type ชนิดนี้แสดงได้ดังรูปที่ 3.3



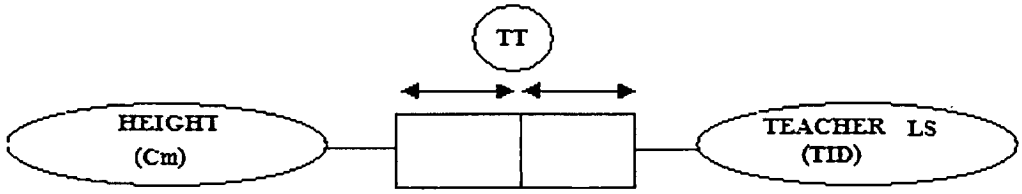
รูปที่ 3.3 แสดง Temporal Fact Type ที่ระบุ Valid Time

2. Temporal Fact Type ที่ระบุ Life span การเกิดของ Temporal Fact Type ชนิดนี้เป็นกรณีเดียวกันกับ Temporal Entity Type ที่ระบุ Life span อยู่ภายนอก Entity Type ลักษณะ Entity Type ชนิดนี้แสดงได้ดังรูปที่ 3.4

รูปที่ 3.4 แสดง Temporal Fact Type ที่ระบุ Life Span

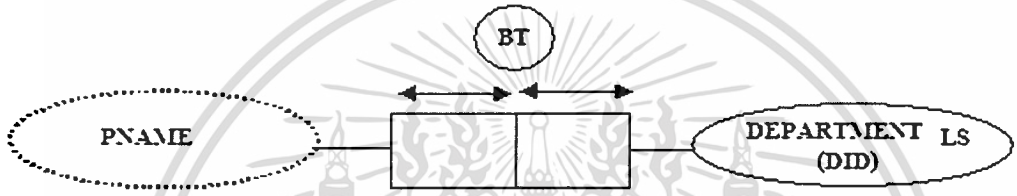
3. Temporal Fact Type ที่ระบุ Transaction Time กรณีนี้จะเกิดขึ้น ขึ้นกับ Application นั้น ๆว่ามีความเป็นไปได้ที่จะเกิดขึ้นได้ เช่น User เคยเชื่อว่าอาจารย์ สมชายมีความสูง 168 เซนติเมตร ซึ่งต่อมา User พบว่า อาจารย์สมชาย มีความสูงเป็น 187 เซนติเมตร เป็นต้น สัญลักษณ์ที่ใช้การระบุค่าเป็น "TT" ที่อยู่ในรูปวงกลม แสดงตัวอย่างดังรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสดง Temporal Fact Type ที่ระบุ Transaction Time

4. Temporal Fact Type ที่ระบุ ทั้ง Valid Time และ Transaction Time กรณีนี้เกิดขึ้นเมื่อต้องการที่จะจัดเก็บ Valid Time เมื่อ Fact นั้นเป็นจริง และระบุ Transaction Time เพื่อจะได้รู้ว่า Fact นั้น เป็นจริงเมื่อใด สัญลักษณ์ที่ใช้ระบุ Valid Time และ Transaction Time ระบุอยู่ในรูปของ "BT" ที่อยู่ในรูปวงกลม แสดงตัวอย่างดังรูปที่ 3.6



รูปที่ 3.6 แสดง Temporal Fact Type ที่ระบุทั้ง Valid Time และ Transaction Time

3.3 ขั้นตอนการออกแบบฐานข้อมูลเชิงเวลา ด้วยวิธีทีไนแอม

ขั้นตอนการออกแบบฐานข้อมูลเชิงเวลา ด้วยวิธีทีไนแอม หรือกระบวนการการเขียนแบบจำลองทีไนแอมสำหรับสำฐานข้อมูลเชิงเวลาประกอบด้วย 2 ขั้นตอนหลัก ได้แก่ ขั้นตอนการเขียนโครงร่างหลัก และขั้นตอนการเขียนโครงร่างย่อย

3.3.1 ขั้นตอนการเขียนโครงร่างหลัก (Main Schema) ประกอบด้วยขั้นตอนดังนี้

1. ระบุ Temporal Entity Type ที่สนใจ

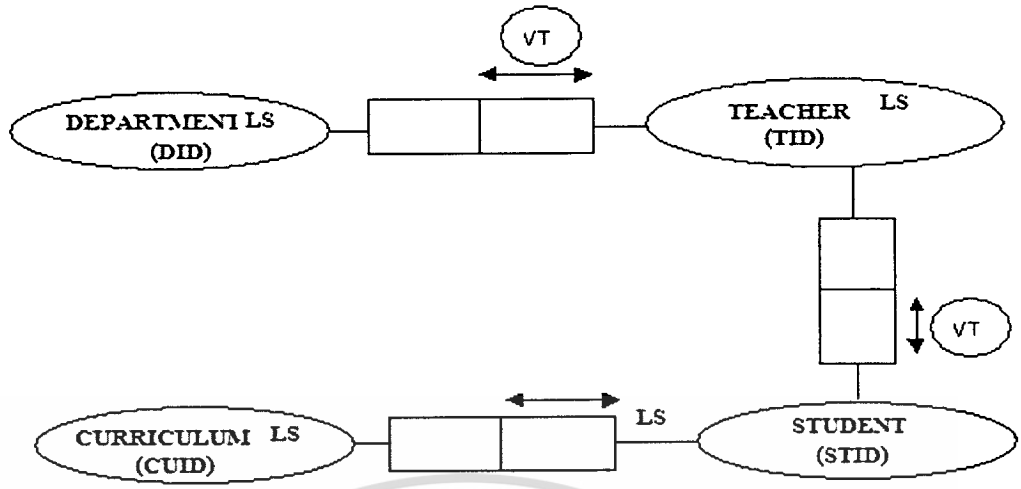
2. พิจารณา Temporal Entity Type จากขั้นตอนที่ 1 ว่าแต่ละ Temporal Entity Type เป็น Temporal Entity Type ชนิดใด

3. ระบุ Life span ให้กับแต่ละ Temporal Entity Type ตามชนิดของ Temporal Entity Type นั้น

4. ระบุ Temporal Fact Type ให้กับ Temporal Entity Type ตามเงื่อนไขที่สนใจ

5. พิจารณาTemporal Fact Type แต่ละตัวว่าเป็น Temporal Fact Type ชนิดใดตามเงื่อนไขจากขั้นตอนที่ 4 เพื่อที่จะทำการระบุรูปแบบเวลา ได้แก่ Valid Time (VT) หรือ Life span (LS) หรือ Transaction Time (TT) หรือทั้ง Valid Time และ Transaction Time (BT) ให้กับแต่ละ Temporal Fact Type

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แสดงตัวอย่างการเขียนโครงร่างหลัก

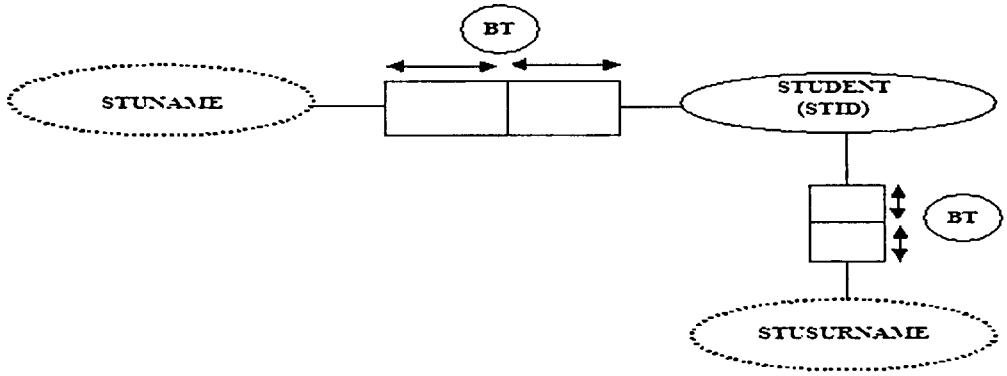
จากรูปที่ 3.7 แสดงตัวอย่างการเขียนโครงร่างหลักของระบบสารสนเทศข้อมูลนักศึกษาสำหรับอาจารย์ที่ปรึกษา ที่ประกอบไปด้วย Temporal Entity Type ที่ระบุ Life span อยู่ภายใน Entity Type ได้แก่ DEPARTMENT , TEACHER , CURRICULUM และ Entity Type ที่ระบุ Life span อยู่ภายนอก Entity Type ได้แก่ Entity Type STUDENT โดยมีเงื่อนไขที่ระบุตาม Temporal Fact Type ได้แก่ Temporal Entity Type DEPARTMENT มีความสัมพันธ์กับ Temporal Entity Type TEACHER แบบหนึ่งต่อกลุ่ม Temporal Entity Type TEACHER มีความสัมพันธ์กับ Temporal Entity Type STUDENT แบบหนึ่งต่อกลุ่ม และ Temporal Entity Type CURRICULUM มีความสัมพันธ์กับ Temporal Entity STUDENT แบบหนึ่งต่อกลุ่ม

3.3.2 ขั้นตอนการเขียนโครงร่างย่อย (Sub Schema) ประกอบด้วยขั้นตอนดังนี้

1. พิจารณา Temporal Entity Type แต่ละตัวจากขั้นตอนการเขียนโครงร่างหลัก ว่า Temporal Entity Type แต่ละตัวประกอบด้วยรายละเอียดอะไรบ้างเพื่อที่จะทำการระบุ Entity Type และ หรือ Label Type ให้กับ Temporal Entity Type นั้น ๆ

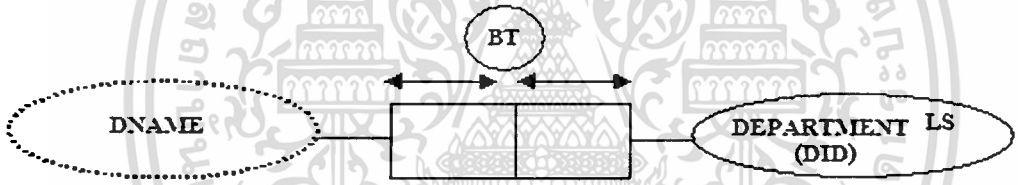
2. ระบุ Temporal Fact Type แต่ละชนิดตามเงื่อนไขที่สนใจ ให้กับ Temporal Entity Type กับ Entity Type และ หรือ Label Type ของ Temporal Entity Type แต่ละตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงตัวอย่างการเขียนโครงร่างย่อยของ Entity Type STUDENT

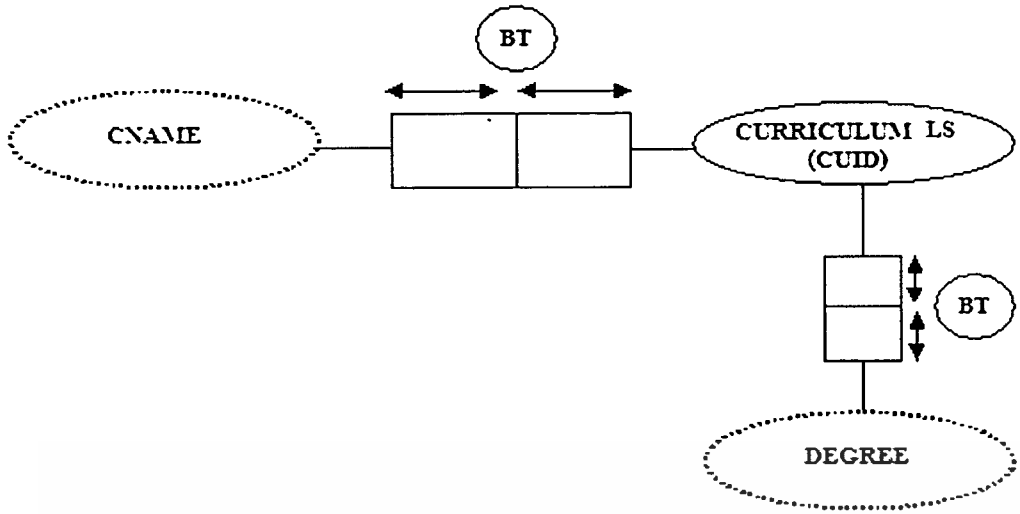
จากรูปที่ 3.8 อธิบายได้ว่า Temporal Entity Type STUDENT (นักศึกษา) มี Label Type STUNAME (ชื่อนักศึกษา) และ Label Type STURNAME (นามสกุลนักศึกษา) ตามเงื่อนไขความสัมพันธ์ที่ระบุที่ Temporal Fact Type คือ นักศึกษาแต่ละคนมีชื่อเพียงชื่อ และ นามสกุลเดียว ณ.เวลาปัจจุบัน (เมื่อ Fact เป็นจริง)



รูปที่3.9 แสดงโครงร่างย่อยของ Entity Type DEPARTMENT

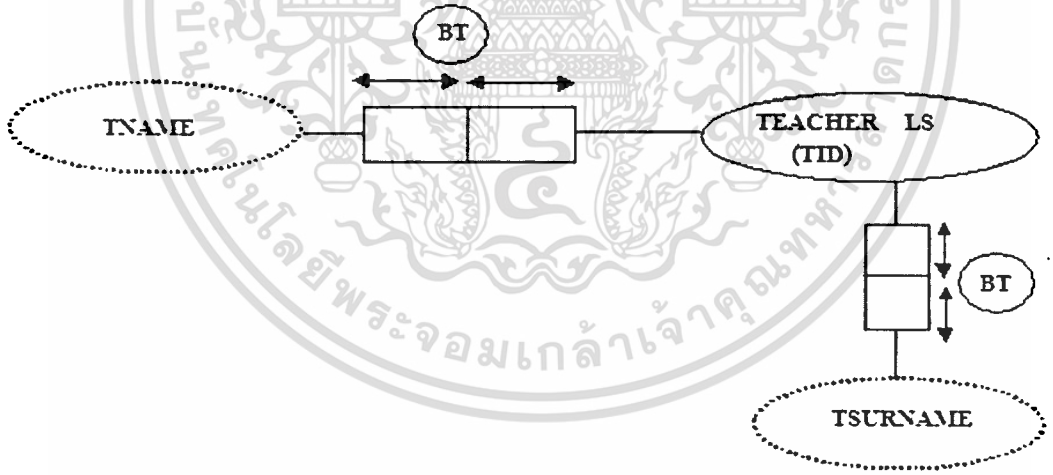
จากรูปที่ 3.9 อธิบายได้ว่า Temporal Entity Type DEPARTMENT (โปรแกรมวิชา) มี Label Type DNAME (ชื่อโปรแกรมวิชา) เงื่อนไขความสัมพันธ์ที่ระบุที่ Temporal Fact Type คือ โปรแกรมวิชา แต่ละโปรแกรมวิชามีชื่อเพียงชื่อเดียว ณ.เวลาปัจจุบัน (เมื่อ Fact เป็นจริง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แสดงโครงร่างย่อยของ Entity Type CURRICULUM

จากรูปที่ 3.10 อธิบายได้ว่า Temporal Entity Type CURRICULUM (หลักสูตร) มี Label Type CNAME (ชื่อหลักสูตร) และ Label Type DEGREE (วุฒิ) ตามเงื่อนไขความสัมพันธ์ที่ระบุที่ Temporal Fact Type คือ แต่ละหลักสูตรมีชื่อหลักสูตรเพียงชื่อเดียว ณ.เวลาปัจจุบัน (เมื่อ Fact เป็นจริง) และ มีวุฒิเดียว ณ.เวลาปัจจุบัน (เมื่อ Fact เป็นจริง) เช่นเดียวกัน



รูปที่ 3.11 แสดงโครงร่างย่อยของ Entity Type TEACHER

จากรูปที่ 3.11 อธิบายได้ว่า Temporal Entity Type TEACHER(อาจารย์) มี Label Type TNAME (ชื่ออาจารย์) และ Label Type TSURNAME (นามสกุลอาจารย์) ตามเงื่อนไขความสัมพันธ์ที่ระบุที่ Temporal Fact Type คือ อาจารย์ที่ปรึกษาแต่ละคนมีชื่อ และ นามสกุล เพียงชื่อและสกุลเดียว ณ.เวลาปัจจุบัน (เมื่อ Fact เป็นจริง)

จากรูปที่ 3.8-3.11 แสดงการเขียนโครงร่างย่อยทั้งหมดจากโครงร่างหลักรูปที่ 3.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

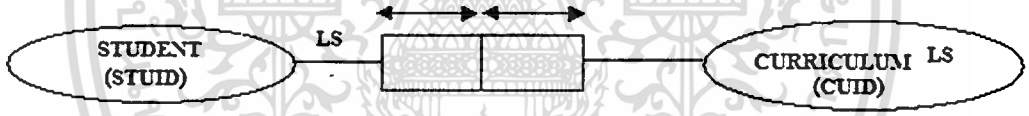
3.4 ขั้นตอนการแปลง ทีไนแอมเป็นฐานข้อมูลเชิงสัมพันธ์

ขั้นตอนการแปลงฐานข้อมูลทีไนแอมเป็นฐานข้อมูลเชิงสัมพันธ์ [7] [8] ประกอบด้วยกรณีต่าง ๆ เมื่อพิจารณาจาก Temporal Fact Type ได้แก่ กรณี Temporal Fact Type ที่มี Life span ระบุอยู่ กรณี Temporal Fact Type ที่มี Valid Time ระบุอยู่ กรณี Temporal Fact Type ที่มี Transaction Time ระบุอยู่ และ กรณีที่ Temporal Fact Type มีทั้ง Valid Time และ Transaction Time ระบุอยู่ หรือเรียกว่า Bitemporal

3.4 .1 กรณี Temporal Fact Type มี Life span (LS) ระบุอยู่

กรณี Temporal Fact Type ที่มี Life span ระบุอยู่ แบ่งการพิจารณาออกได้เป็น 3 กรณี คือ กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบ หนึ่งต่อหนึ่ง กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบ หนึ่งต่อกลุ่ม และ กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบกลุ่มต่อกลุ่ม

3.4.1.1 กรณี Temporal Fact Type ระบุ Uniqueness Constraint แบบหนึ่งต่อหนึ่ง



รูปที่ 3.12 แสดงตัวอย่าง Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อหนึ่ง

จากรูปที่ 3.12 เป็นตัวอย่างกรณีสมมุติที่แสดงให้เห็นถึง Temporal Entity Type 2 ตัว ที่มีความสัมพันธ์กันแบบ หนึ่งต่อหนึ่ง โดยที่ Temporal Fact Type มี Life span ระบุอยู่ ซึ่งความหมายของรูปที่ 3.12 อธิบายได้ว่า Temporal Entity Type CURRICULUM (หลักสูตร) ซึ่งมี CUID เป็น Unique Identifier มีความสัมพันธ์กับ Temporal STUDENT (นักศึกษา) ที่มี STUID เป็น Unique Identifier แบบหนึ่งต่อหนึ่ง ซึ่งหมายถึง ในแต่ละ Life span ของนักศึกษาหนึ่งคน สามารถอยู่ได้เพียงหลักสูตรเดียวในแต่ละ Life span ของหลักสูตร หรือ ในแต่ละ Life span ของหลักสูตรมี Life span ของนักศึกษาเพียงคนเดียว จากกรณีนี้ขั้นตอนของการแปลงเป็นฐานข้อมูลเชิงสัมพันธ์ มีขั้นตอนดังนี้

ขั้นตอนที่ 1 สร้าง 1 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายใน โดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลัก (Primary Key) ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีเลชัน และ กำหนดรูปแบบ Life span ที่ระบุอยู่ในTemporal Entity Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Life span เริ่มต้น (LSs) และ Life span สิ้นสุด (LSe)

ขั้นตอนที่ 2 นำ Unique Identifier ของTemporal Entity Type ที่ Life span ระบุอยู่ภายนอกเป็น Attribute ของรีเลชันในขั้นตอนที่ 1

จากรูปที่ 3.12 แปลงเป็นรีเลชัน ตามขั้นตอนทั้งสองที่กล่าวมาได้รีเลชันCURRICULUM ดังนี้

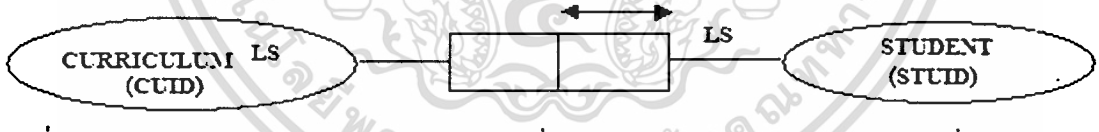
CURRICULUM

CUID	LSs	LSe	STUID
CU1	2540-01-01	9999-12-31	450364578
CU2	2540-01-01	9999-12-31	450364555

รูปที่ 3.13 แสดงรีเลชันที่ได้จากการแปลงทีไนแอมจากรูปที่ 3.12

จากรูปที่ 3.13 อธิบายได้ว่า รหัสหลักสูตร(CUID) CU1 มี Life span เริ่มต้นเมื่อวันที่ 1 มกราคม 2540 และ ปัจจุบัน ก็ยังคงมีLife span อยู่ (กำหนดให้ LSe มีค่าเป็น 9999-12-31 ตามมาตรฐานของ SQL-92) และหลักสูตรนี้มี Life span ของนักศึกษาหนึ่งคนคือนักศึกษารหัส 450364578 เป็นต้น

3.4.1.2 กรณี Temporal Fact Type ระบุ Uniqueness Constraint แบบหนึ่งต่อกลุ่ม



รูปที่ 3.14 แสดงตัวอย่าง Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อกลุ่ม

จากรูปที่ 3.14 เป็นตัวอย่างกรณีสมมุติที่แสดงให้เห็นถึง Temporal Entity Type 2 ตัว ที่มีความสัมพันธ์กันแบบ หนึ่งต่อกลุ่ม (Entity Type CURRICULLUM มีความสัมพันธ์กับ Entity Type STUDENT แบบหนึ่งต่อกลุ่มจาก Entity Type CURRICULLUM ไปยัง Entity Type STUDENT) โดยที่ Temporal Fact Type มี Life span ระบุอยู่ ซึ่งความหมายของรูปที่ 3.14 อธิบายได้ว่า Temporal Entity Type CURRICULUM (หลักสูตร) ซึ่งมี CUID เป็น Unique Identifier มีความสัมพันธ์กับ Temporal STUDENT (นักศึกษา) ที่มี STUID เป็น Unique Identifier แบบหนึ่งต่อกลุ่ม ซึ่งหมายถึง ในแต่ละ Life span ของแต่ละหลักสูตรมีLife span ของนักศึกษาได้หลายคน หรือ ในทางกลับกัน Life span ของนักศึกษาหนึ่งคนปรากฏใน Life span

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของหลักสูตรได้เพียงหลักสูตรเดียว จากกรณีนี้ขั้นตอนของการแปลงเป็นฐานข้อมูลเชิงสัมพันธ์ มีขั้นตอนดังนี้

ขั้นตอนที่ 1 สร้าง 1 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายในโดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลักของรีเลชัน และกำหนดรูปแบบ Life span ที่ระบุอยู่ในTemporal Entity Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Life span เริ่มต้น (LSs) และ Life span สิ้นสุด (LSe)

ขั้นตอนที่ 2 สร้าง 1 รีเลชันสำหรับ Temporal Fact Type โดยนำ Unique Identifier ของ Temporal Entity Type ที่ Life span ระบุอยู่ภายนอกเป็นคีย์หลัก ของรีเลชัน และ นำคีย์หลักของรีเลชันจากขั้นตอนที่ 1 มาเป็นคีย์นอกของรีเลชัน

จากรูปที่ 3.14 แปลงเป็นรีเลชันตามขั้นตอนทั้งสองที่กล่าวมาได้รีเลชันCURRICULUM และ รีเลชัน STUDENT ดังนี้

CURRICURUM

CUID	LSs	Lse
CU1	2540-01-01	9999-12-31
CU2	2540-01-01	9999-12-31

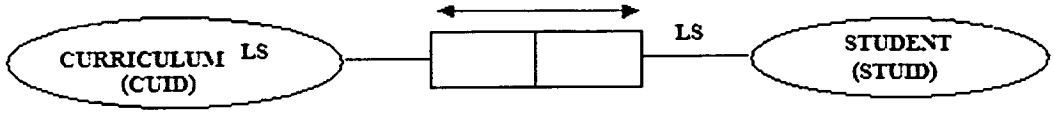
STUDENT

STUID	CUID
450364578	CU1
450364555	CU1
450364579	CU2

รูปที่ 3.15 แสดง 2 รีเลชันที่ได้จากการแปลงทีไนแอม จากรูปที่ 3.14

จากรูปที่ 3.15 อธิบายได้ว่า รหัสหลักสูตร(CUID) CU1 มี Life span เริ่มต้นเมื่อวันที่ 1 มกราคม 2540 และ ปัจจุบัน ก็ยังคงมีLife span อยู่ (กำหนดให้ LSe มีค่าเป็น 9999-12-31) และหลักสูตรนี้มี Life span ของนักศึกษาสองคนคือนักศึกษารหัส 450364578 และ นักศึกษารหัส 450364555เป็นต้น

3.4.1.3 กรณี Temporal Fact Type ระบุ Uniqueness Constraint แบบกลุ่มต่อกลุ่ม



รูปที่ 3.16 แสดงตัวอย่าง Temporal Fact Type ที่มีความสัมพันธ์ แบบกลุ่มต่อกลุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.16 เป็นตัวอย่างกรณีสมมุติที่แสดงให้เห็นถึง Temporal Entity Type 2 ตัว ที่มีความสัมพันธ์กันแบบ กลุ่มต่อกลุ่ม โดยที่ Temporal Fact Type มี Life span ระบุอยู่ ซึ่งความหมายของรูปที่ 3.16 อธิบายได้ว่า Temporal Entity Type CURRICULUM (หลักสูตร) ซึ่งมี CUID เป็น Unique Identifier มีความสัมพันธ์กับ Temporal STUDENT (นักศึกษา) ที่มี STUID เป็น Unique Identifier แบบกลุ่มต่อกลุ่ม ซึ่งหมายถึง ในแต่ละ Life span ของแต่ละหลักสูตรมีLife span ของนักศึกษาได้หลายคน หรือ ในทางกลับกัน Life span ของนักศึกษาแต่ละคนสามารถปรากฏใน Life span ของหลักสูตรได้หลายหลักสูตรเดียว จากกรณีนี้ขั้นตอนของการแปลงเป็นฐานข้อมูลเชิงสัมพันธ์ มีขั้นตอนดังนี้

ขั้นตอนที่ 1 สร้าง 1 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายใน โดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลักของรีเลชัน และกำหนดรูปแบบ Life span ที่ระบุอยู่ในTemporal Entity Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Life span เริ่มต้น (LSs) และ Life span สิ้นสุด (LSe)

ขั้นตอนที่ 2 สร้าง 1 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายนอก โดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลักของรีเลชัน

ขั้นตอนที่ 3 สร้าง 1 รีเลชันสำหรับ Temporal Fact Type โดยนำคีย์หลักจากขั้นตอนที่หนึ่งมาเป็นคีย์นอก และ นำคีย์หลักจากขั้นตอนที่ 2 มาเป็นคีย์นอก สำหรับคีย์หลักของรีเลชันนี้คือแอททริบิวท์ทั้งสองที่กล่าวมารวมกันเป็นคีย์หลักของรีเลชัน

จากรูปที่ 3.16 แปลงเป็นรีเลชัน ตามขั้นตอนทั้งสามที่กล่าวมาได้รีเลชัน CURRICULUM รีเลชัน STUDENT และ รีเลชัน CURICURUM_STUDENT ดังนี้

CURRICULUM

CUID	LSs	Lse
CU1	2540-01-01	9999-12-31
CU2	2540-01-01	9999-12-31

STUDENT

STUID
450364578
450364555

CURRICULUM_STUDENT

CUID	STUID
CU1	450364578
CU2	450364578
CU2	450364555

รูปที่ 3.17 แสดงรีเลชัน 3 รีเลชัน ที่แปลงจากทีไนแอม จากรูปที่ 3.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.17 อธิบายได้ว่า รหัสหลักสูตร(CUID) CU2 มี Life span เริ่มต้นเมื่อวันที่ 1 มกราคม 2540 และ ปัจจุบัน ก็ยังคงมีLife span อยู่ (กำหนดให้ LSe มีค่าเป็น 9999-12-31) และหลักสูตรนี้มี Life span ของนักศึกษาสองคนคือนักศึกษารหัส 450364578 และ นักศึกษารหัส 450364555และ ขณะเดียวกัน นักศึกษารหัส 450364578 ยังคงมี Life span ของนักศึกษาที่อยู่ใน Life span ของ รหัสหลักสูตร CU1 ด้วย เป็นต้น

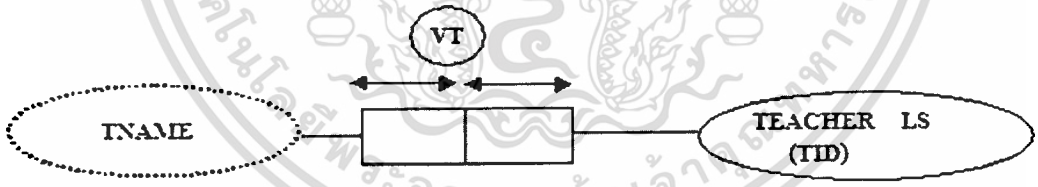
3.4 .2 กรณี Temporal Fact Type มี Valid Time (VT) ระบุอยู่

กรณี Temporal Fact Type ที่มี Valid Time ระบุอยู่ แบ่งการพิจารณาออกได้เป็น 3 กรณี คือ กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบ หนึ่งต่อหนึ่ง กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบ หนึ่งต่อกลุ่ม และ กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบกลุ่มต่อกลุ่ม

3.4.2.1 กรณี Temporal Fact Type ระบุ Uniqueness Constraint แบบหนึ่งต่อหนึ่ง

กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อหนึ่ง พิจารณาได้ 2 กรณี ได้แก่ กรณี Temporal Entity Type มีความสัมพันธ์กับ Entity Type หรือ Label Type กรณีที่สอง กรณี Temporal Entity Type มีความสัมพันธ์กับ Temporal Entity Type

1. กรณี Temporal Entity Type มีความสัมพันธ์กับ Entity Type หรือ Label Type



รูปที่ 3.18 แสดงตัวอย่าง Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อหนึ่ง

จากรูปที่ 3.18 เป็นตัวอย่างกรณีสมมุติที่แสดงให้เห็นถึง Temporal Entity Type TEACHER และ Label Type TNAME มีความสัมพันธ์กันแบบ หนึ่งต่อหนึ่ง โดยที่ Temporal Fact Type มี Valid Time ระบุอยู่ และเป็น Temporal Entity Type ที่เกิดจากขั้นตอนการออกแบบโครงร่างหลักของทีไนแอม ซึ่งความหมายของรูปที่ 13.8 อธิบายได้ว่า Temporal Entity Type TEACHER (อาจารย์) ซึ่งมี TID เป็น Unique Identifier มีความสัมพันธ์กับ Label Type TNAME (ชื่ออาจารย์) แบบหนึ่งต่อหนึ่ง ซึ่งหมายถึง ในแต่ละ Life span ของอาจารย์หนึ่งคนมีชื่ออาจารย์ได้เพียงชื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เดียวในแต่ละ Valid Time จากกรณีนี้ขั้นตอนของการแปลงเป็นฐานข้อมูลเชิงสัมพันธ์ มีขั้นตอนดังนี้

ขั้นตอนที่ 1 สร้าง 1 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายในโดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลัก(Primary Key) ของรีเลชัน และ กำหนดรูปแบบ Life span ที่ระบุอยู่ในTemporal Entity Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Life span เริ่มต้น (LSs) และ Life span สิ้นสุด (LSe)

ขั้นตอนที่ 2 สร้าง 1 รีเลชัน สำหรับ Temporal Fact Type โดยนำชื่อของ Label Type หรือ Unique Identifier ของ Entity Type ที่ระบุอยู่เป็น Attribute ของรีเลชันและ นำคีย์หลักจากขั้นตอนที่ 1 เป็นคีย์นอก (Foreign Key) และ กำหนดรูปแบบของ Valid Time ที่ระบุอยู่บน Temporal Fact Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Valid Time เริ่มต้น (VTs) และ Valid Time สิ้นสุด สำหรับคีย์หลักของรีเลชันก็คือ Attribute ที่เป็นคีย์นอกรวมกับ Valid Time เริ่มต้น

จากรูปที่ 3.18 แปลงเป็นรีเลชัน ตามขั้นตอนทั้งสองที่กล่าวมาได้รีเลชัน TEACHER และ TEACHER_NAME ดังนี้

TEACHER

TID	LSs	LSe
T1	2540-01-01	9999-12-31
T2	2540-01-01	9999-12-31

TEACHER_NAME

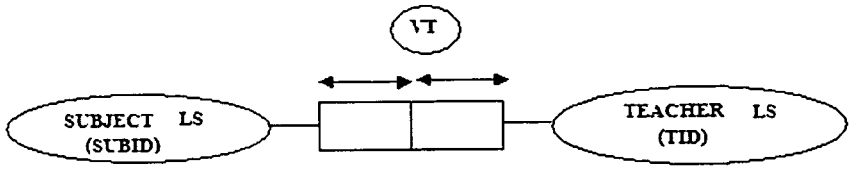
TID	TNAME	VTs	Vte
T1	SOMCHAI	2540-01-01	2544-01-01
T1	SOMPONG	2544-01-01	9999-12-31
T2	SUCHADA	2540-01-01	9999-12-31

รูปที่ 3.19 แสดงรีเลชัน 2 รีเลชัน ที่แปลงจากรูปที่ในแอมที่ 3.18

จากรูปที่ 3.19 รีเลชัน TEACHER มี Attribute TID เป็นคีย์หลักของรีเลชัน และ รีเลชัน TEACHER_NAME มี TID และ VTs เป็นคีย์หลักของรีเลชัน อธิบายได้ว่า รหัสอาจารย์(TID) T1 มี Life span เริ่มต้นเมื่อวันที่ 1 มกราคม 2540 จนถึงปัจจุบัน ก็ยังคงมีLife span อยู่ (กำหนดให้ LSe มีค่าเป็น 9999-12-31) และรหัสอาจารย์นี้มีชื่ออาจารย์คือ " SOMPONG" เพียงชื่อเดียวเมื่อเวลาปัจจุบัน (VTe มีค่าเป็น 9999-12-31) ซึ่งเมื่ออดีตที่ผ่านมาคือตั้งแต่ 40-01-01 จนถึง 44-01-01 อาจารย์ที่มีรหัส T1 เคยชื่อ "SOMCHAI" มาก่อนเป็นต้น

2. กรณี Temporal Entity Type มีความสัมพันธ์กับ Temporal Entity Type

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.20 แสดงตัวอย่าง Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อหนึ่ง

จากรูปที่ 3.20 เป็นตัวอย่างกรณีสมมุติที่แสดงให้เห็นถึง Temporal Entity Type TEACHER และ Temporal Entity Type SUBJECT ที่มีความสัมพันธ์กันแบบ หนึ่งต่อหนึ่ง ลักษณะการเกิด Temporal Fact Type กรณีนี้เกิดขึ้นในขั้นตอนการเขียนโครงร่างหลักของทีไนแอม ซึ่งความหมายของรูปที่ 3.20 อธิบายได้ว่า Temporal Entity Type TEACHER (อาจารย์) ซึ่งมี TID เป็น Unique Identifier มีความสัมพันธ์กับ Temporal Entity Type SUBJECT (วิชา) แบบหนึ่งต่อหนึ่ง ซึ่งหมายถึง ในแต่ละ Life span ของอาจารย์หนึ่งคนมี Life span ของวิชาได้เพียง Life span เดียว จากกรณีนี้ขั้นตอนของการแปลงเป็นฐานข้อมูลเชิงสัมพันธ์ มีขั้นตอนดังนี้

ขั้นตอนที่ 1 สร้าง 2 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายในโดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลัก ของรีเลชัน และกำหนดรูปแบบ Life span ที่ระบุอยู่ในTemporal Entity Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Life span เริ่มต้น (LSs) และ Life span สิ้นสุด (LSe)

ขั้นตอนที่ 2 สร้าง 1 รีเลชัน สำหรับ Temporal Fact Type โดยนำคีย์หลักจากรีเลชันทั้งสองในขั้นตอนที่ 1 มาเป็นคีย์นอก และ กำหนดรูปแบบของ Valid Time ที่ระบุอยู่บน Temporal Fact Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Valid Time เริ่มต้น (VTs) และ Valid Time สิ้นสุด สำหรับคีย์หลักของรีเลชันก็คือแอทริบิวท์ที่เป็นคีย์นอกแอททริบิวท์ใด Attribute หนึ่งรวมกับ Valid Time เริ่มต้น

จากรูปที่ 3.20 แปลงเป็นรีเลชันตามขั้นตอนทั้งสองที่กล่าวมาได้รีเลชัน TEACHER รีเลชัน SUBJECT และ รีเลชัน TEACHER_SUBJECT ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TEACHER

TID	LSs	LSe
T1	2540-01-01	9999-12-31
T2	2540-01-01	9999-12-31

SUBJECT

SUBID	LSs	Lse
CS105	2530-01-01	9999-12-31
CS444	2530-01-01	9999-12-31

TEACHER_SUBJECT

TID	SUBID	VTs	VTe
T1	CS105	2540-01-01	2544-01-01
T1	CS444	2544-01-01	9999-12-31
T2	CS444	2540-01-01	2544-01-01
T2	CS105	2544-01-01	9999-12-31

รูปที่ 3.21 แสดง 3 รีเลชันที่ได้จากการแปลงทีไนแอมรูปที่ 3.20

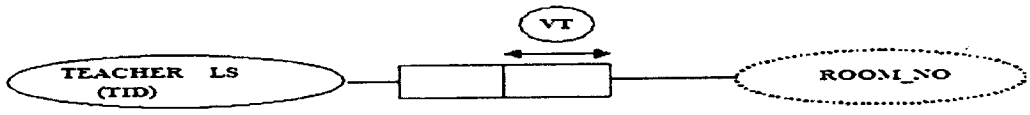
จากรูปที่ 3.21 รีเลชัน TEACHER มี แอทริบิวท์ TID เป็นคีย์หลักของรีเลชัน รีเลชัน TEACHER_NAME มี TID และ VTs เป็นคีย์หลักของรีเลชัน และ รีเลชัน TEACHER_SUJECT มี TID และ VTs เป็นคีย์หลักของรีเลชัน อธิบายได้ว่า รหัสอาจารย์(TID) T1 มี Life span เริ่มต้นเมื่อวันที่ 1 มกราคม 2540 จนถึงปัจจุบัน ก็ยังคงมีLife span อยู่ (กำหนดให้ LSe มีค่าเป็น 9999-12-31) รหัสวิชา (CS105) มี Life span ตั้งแต่ วันที่ 1 มกราคม 2530 จนถึงปัจจุบัน และรหัสอาจารย์ T1มีความสัมพันธ์กับรหัสวิชา CS105 แบบหนึ่งต่อหนึ่งในอดีตที่ผ่านมาคือ ตั้งแต่ Valid Time เริ่มต้นที่วันที่ 1 มกราคม 2540 จนถึง Valid Time สิ้นสุดที่ วันที่ 1 มกราคม 2544 และ รหัสอาจารย์ T1มีความสัมพันธ์กับรหัสวิชา CS44 แบบหนึ่งต่อหนึ่งในปัจจุบันคือ ตั้งแต่ Valid Time เริ่มต้นที่วันที่ 1 มกราคม 2544 จนถึง Valid Time สิ้นสุดเป็น "9999-12-31" (เป็นค่าปัจจุบัน) เป็นต้น

3.4.2.2 กรณี Temporal Fact Type ระบุ Uniqueness Constraint แบบหนึ่งต่อกลุ่ม

กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อกลุ่มพิจารณาได้ 2 กรณี ได้แก่ กรณี Temporal Entity Type มีความสัมพันธ์กับ Entity Type หรือ Label Type กรณีที่สอง กรณี Temporal Entity Type มีความสัมพันธ์กับ Temporal Entity Type

1 กรณี Temporal Entity Type มีความสัมพันธ์กับ Entity Type หรือ Label Type

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.22 แสดง Temporal Entity Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อกลุ่ม

จากรูปที่ 3.22 เป็นตัวอย่างกรณีสมมุติที่แสดงให้เห็นถึง Temporal Entity Type TEACHER และ Label Type ADDRESS มีความสัมพันธ์แบบ หนึ่งต่อกลุ่ม โดยที่ Temporal Fact Type มี Valid Time ระบุอยู่ และเป็น Temporal Entity Type ที่เกิดจากขั้นตอนการออกแบบโครงร่างหลักของทีไนแอม ซึ่งความหมายของรูปที่ 3.22 อธิบายได้ว่า Temporal Entity Type TEACHER (อาจารย์) ซึ่งมี TID เป็น Unique Identifier มีความสัมพันธ์กับ Label Type ADDRESS (ที่อยู่อาจารย์) แบบหนึ่งต่อกลุ่ม ซึ่งหมายถึง ในแต่ละ Life span ของอาจารย์หนึ่งคนมีที่อยู่ได้หลายที่ในแต่ละ Valid Time จากกรณีนี้ขั้นตอนของการแปลงเป็นฐานข้อมูลเชิงสัมพันธ์ มีขั้นตอนดังนี้

ขั้นตอนที่ 1 สร้าง 1 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายในโดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลัก ของรีเลชัน และกำหนดรูปแบบ Life span ที่ระบุอยู่ในTemporal Entity Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Life span เริ่มต้น (LSs) และ Life span สิ้นสุด (LSe)

ขั้นตอนที่ 2 สร้าง 1 รีเลชัน สำหรับ Temporal Fact Type โดยนำชื่อของ Label Type หรือ Unique Identifier ของ Entity Type ที่ระบุอยู่เป็น Attribute ของรีเลชันและ นำคีย์หลักจากขั้นตอนที่ 1 เป็นคีย์นอก และ กำหนดรูปแบบของ Valid Time ที่ระบุอยู่บน Temporal Fact Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Valid Time เริ่มต้น (VTs) และ Valid Time สิ้นสุด สำหรับคีย์หลักของรีเลชันก็คือ Attribute ที่เป็นชื่อของ Label Type รวมกับ Valid Time เริ่มต้น

จากรูปที่ 3.22 แปลงเป็นรีเลชันตามขั้นตอนทั้งสองที่กล่าวมาได้รีเลชัน TEACHER และ TEACHER_ADDRESS ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TEACHER

TID	LSs	LSe
T1	2540-01-01	9999-12-31
T2	2540-01-01	9999-12-31

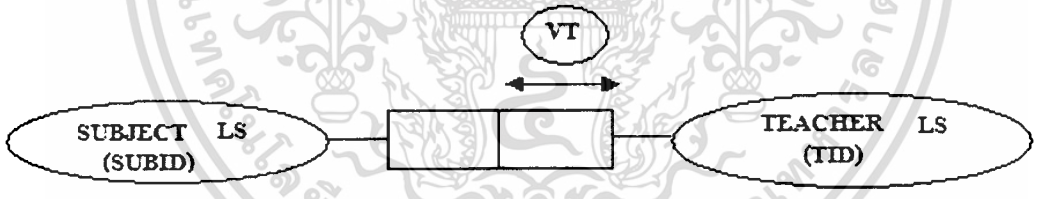
TEACHER_ROOM

TID	ROOM_NO	VTs	VTe
T1	ROOM1	2540-01-01	2544-01-01
T1	ROOM2	2544-01-01	9999-12-31
T1	ROOM3	2544-01-01	9999-12-31
T2	ROOM1	2544-01-01	9999-12-31

รูปที่ 3.23 แสดง 3 รีเลชันที่ได้จากการแปลง รูปทีไนแอมที่ 3.22

จากรูปที่ 3.23 รีเลชัน TEACHER มี Attribute TID เป็นคีย์หลักของรีเลชัน **รีเลชัน** TEACHER_ROOM มี ROOM_NO และ VTs เป็นคีย์หลักของรีเลชัน อธิบายได้ว่า รหัสอาจารย์ (TID) T1 มี Life span เริ่มต้นเมื่อวันที่ 1 มกราคม 2544 จนถึงปัจจุบัน ก็ยังคงมีLife span อยู่ **โดยที่** รหัสอาจารย์ T1มีความสัมพันธ์กับที่อยู่แบบหนึ่งต่อกลุ่ม เช่น รหัสอาจารย์ T1มีที่อยู่ ทั้ง ROOM2 และ ROOM3 ในแต่ละ Valid Time ในปัจจุบัน และในอดีตรหัสอาจารย์ T1 มีที่อยู่ที่ ROOM1 ตั้งแต่ Valid Time เริ่มต้นที่ วันที่ 1 มกราคม 2540 จนถึง Valid Time สิ้นสุดที่วันที่ 1 มกราคม 2544 เป็นต้น

2. กรณี Temporal Entity Type มีความสัมพันธ์กับ Temporal Entity Type



รูปที่ 3.24 แสดง Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อกลุ่ม

จากรูปที่ 3.24 เป็นตัวอย่างกรณีสมมุติที่แสดงให้เห็นถึง Temporal Entity Type TEACHER และ Temporal Entity Type SUBJECT ที่มีความสัมพันธ์กันแบบ หนึ่งต่อกลุ่ม ลักษณะการเกิด Temporal Fact Type กรณีนี้เกิดขึ้นในขั้นตอนการเขียนโครงร่างหลักของทีไนแอม ซึ่งความหมายของรูปที่ 3.24 อธิบายได้ว่า Temporal Entity Type TEACHER (อาจารย์) ซึ่งมี TID เป็น Unique Identifier มีความสัมพันธ์กับ Temporal Entity Type SUBJECT (วิชา) แบบหนึ่งต่อกลุ่มซึ่งหมายถึง ในแต่ละ Life span ของอาจารย์หนึ่งคนมี Life span ของวิชาได้หลาย Life span จากกรณีนี้ขั้นตอนของการแปลงเป็นฐานข้อมูลเชิงสัมพันธ์ มีขั้นตอนดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 1 สร้าง 2 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายในโดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลักของรีเลชัน และกำหนดรูปแบบ Life span ที่ระบุอยู่ในTemporal Entity Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Life span เริ่มต้น (LSs) และ Life span สิ้นสุด (LSe)

ขั้นตอนที่ 2 สร้าง 1 รีเลชัน สำหรับ Temporal Fact Type โดยนำคีย์หลักจากรีเลชันทั้งสองในขั้นตอนที่ 1 มาเป็นคีย์นอก และ กำหนดรูปแบบของ Valid Time ที่ระบุอยู่บน Temporal Fact Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Valid Time เริ่มต้น (VTs) และ Valid Time สิ้นสุด สำหรับคีย์หลักของรีเลชันก็คือ Attribute ที่เป็นคีย์นอกฝั่งกลุ่ม (Many) รวมกับ Valid Time เริ่มต้น

จากรูปที่ 3.24 แปลงเป็นรีเลชัน ตามขั้นตอนทั้งสองที่กล่าวมาได้รีเลชัน TEACHER รีเลชัน SUBJECT และ รีเลชัน TEACHER_SUBJECT ดังนี้

SUBJECT

SUBID	LSs	LSe
CS105	2530-01-01	9999-12-31
CS444	2530-01-01	9999-12-31

TEACHER

TID	LSs	Lse
T1	2540-01-01	9999-12-31
T1	2540-01-01	9999-12-31

SUBJECT_TEACHER

SUBID	TID	VTs	Vte
CS105	T1	2540-01-01	9999-12-31
CS444	T2	2540-01-01	2544-01-01
CS105	T2	2544-01-01	9999-12-31

รูปที่ 3.25 แสดง 3 รีเลชัน ที่ได้จากการแปลงรูปทีไนแอมที่ 3.24

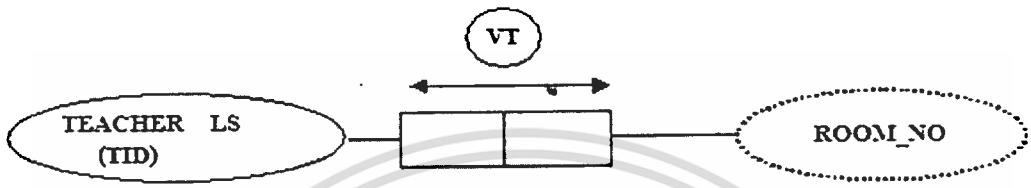
จากรูปที่ 3.25 รีเลชัน TEACHER มี Attribute TID เป็นคีย์หลักของรีเลชันรีเลชัน SUBJECT มี SUBID เป็นคีย์หลักของรีเลชัน และ รีเลชัน SUJECT_TEACHER มี TID และ VTs เป็นคีย์หลักของรีเลชัน อธิบายได้ว่า รหัสอาจารย์(TID) T1 มี Life span เริ่มต้นเมื่อวันที่ 1 มกราคม 2540 จนถึงปัจจุบัน ก็ยังคงมีLife span อยู่ รหัสวิชา (CS105) มี Life span ตั้งแต่ วันที่ 1 มกราคม 2530 จนถึงปัจจุบัน และรหัสวิชา CS105 มีความสัมพันธ์กับรหัสอาจารย์ T1 และ T2 แบบหนึ่งต่อกลุ่ม ในแต่ละ Valid Time เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2.3 กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบกลุ่มต่อกลุ่ม

กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบกลุ่มต่อกลุ่มพิจารณาได้ 2 กรณี ได้แก่ กรณี Temporal Entity Type มีความสัมพันธ์กับ Entity Type หรือ Label Type กรณีที่สอง กรณี Temporal Entity Type มีความสัมพันธ์กับ Temporal Entity Type

1. กรณี Temporal Entity Type มีความสัมพันธ์กับ Entity Type หรือ Label Type



รูปที่ 3.26 แสดง Temporal Fact Type ที่ระบุUniqueness Constraint แบบกลุ่มต่อกลุ่ม

จากรูปที่ 3.26 เป็นตัวอย่างกรณีสมมุติที่แสดงให้เห็นถึง Temporal Entity Type TEACHER และ Label Type ADDRESS มีความสัมพันธ์แบบ กลุ่มต่อกลุ่ม โดยที่ Temporal Fact Type มี Valid Time ระบุอยู่ และเป็น Temporal Entity Type ที่เกิดจากขั้นตอนการออกแบบโครงร่างหลักของทีไนแอม ซึ่งความหมายของรูปที่ 3.26 อธิบายได้ว่า Temporal Entity Type TEACHER (อาจารย์) ซึ่งมี TID เป็น Unique Identifier มีความสัมพันธ์กับ Label Type ADDRESS (ที่อยู่อาจารย์) แบบกลุ่มต่อกลุ่ม ซึ่งหมายถึง ในแต่ละ Life span ของอาจารย์หนึ่งคนมีที่อยู่ได้หลายที่ในแต่ละ Valid Time และในทางกลับกัน ในแต่ละ Valid Time ของที่อยู่สามารถมีอาจารย์ได้หลายคน จากกรณีนี้ขั้นตอนของการแปลงเป็นฐานข้อมูลเชิงสัมพันธ์ มีขั้นตอนดังนี้

ขั้นตอนที่ 1 สร้าง 1 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายในโดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลัก ของรีเลชัน และกำหนดรูปแบบ Life span ที่ระบุอยู่ในTemporal Entity Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Life span เริ่มต้น (LSs) และ Life span สิ้นสุด (LSe)

ขั้นตอนที่ 2 สร้าง 1 รีเลชัน สำหรับ Temporal Fact Type โดยนำชื่อของ Label Type หรือ Unique identifier ที่ระบุอยู่เป็น Attribute ของรีเลชันและ นำคีย์หลักจากขั้นตอนที่ 1 เป็นคีย์นอก และ กำหนดรูปแบบของ Valid Time ที่ระบุอยู่บน Temporal Fact Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Valid Time เริ่มต้น (VTs) และ Valid Time สิ้นสุด สำหรับคีย์หลักของรีเลชันก็คือแอทริบิวท์ที่เป็นคีย์นอกรวมกับชื่อของ Label Type และ Valid Time เริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.26 แปลงเป็นรีเลชัน ตามขั้นตอนทั้งสองที่กล่าวมาได้รีเลชัน TEACHER และ TEACHER_ADDRESS ดังนี้

TEACHER

TID	LSs	LSe
T1	2540-01-01	9999-12-31
T2	2540-01-01	9999-12-31

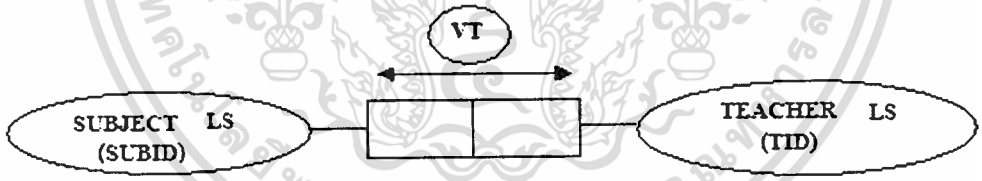
TEACHER_ADDRESS

TID	ROOM_NO	VTs	Vte
T1	ROOM1	2540-01-01	2544-01-01
T1	ROOM2	2544-01-01	9999-12-31
T1	ROOM3	2544-01-01	9999-12-31
T2	ROOM2	2540-01-01	9999-12-31

รูปที่ 3.27 แสดง 3 รีเลชันที่ได้จากการแปลง รูปทีไนแอมที่ 3.26

จากรูปที่ 3.27 รีเลชัน TEACHER มี Attribute TID เป็นคีย์หลักของรีเลชัน รีเลชัน TEACHER_ADDRESS มี TID,ADDRESSและ VTs เป็นคีย์หลักของรีเลชัน อธิบายได้ว่า รหัสอาจารย์(TID) T1 มี Life span เริ่มต้นเมื่อวันที่ 1 มกราคม 2540 จนถึงปัจจุบัน ก็ยังคงมีLife span อยู่ โดยที่รหัสอาจารย์มีความสัมพันธ์กับที่อยู่อาจารย์ (ROOM_NO) เป็นแบบกลุ่มต่อกลุ่ม ในแต่ละ Valid Time

2. กรณี Temporal Entity Type มีความสัมพันธ์กับ Temporal Entity Type



รูปที่ 3.28 แสดงตัวอย่าง Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบกลุ่มต่อกลุ่ม

จากรูปที่ 3.28 เป็นตัวอย่างกรณีสมมุติที่แสดงให้เห็นถึง Temporal Entity Type TEACHER และ Temporal Entity Type TEACHER มีความสัมพันธ์แบบ กลุ่มต่อกลุ่ม โดยที่ Temporal Fact Type มี Valid Time ระบุอยู่ ซึ่งความหมายของรูปที่ 3.28 อธิบายได้ว่า Temporal Entity Type TEACHER (อาจารย์) ซึ่งมี TID เป็น Unique Identifier มีความสัมพันธ์กับ Temporal Entity Type SUBJECT (วิชา) แบบกลุ่มต่อกลุ่ม ซึ่งหมายถึง ในแต่ละ Life span ของอาจารย์หนึ่งคนมีสามารถสอนได้หลายวิชาในแต่ละ Valid Time และในทางกลับกัน ในแต่ละ Life span ของวิชามารถมีอาจารย์สอนได้หลายคนในแต่ละ Valid Time จากกรณีนี้ขั้นตอนของการแปลงเป็นฐานข้อมูลเชิงสัมพันธ์ มีขั้นตอนดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 1 สร้าง 2 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายใน โดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลักของรีเลชัน และกำหนดรูปแบบ Life span ที่ระบุอยู่ในTemporal Entity Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Life span เริ่มต้น (LSs) และ Life span สิ้นสุด (LSe)

ขั้นตอนที่ 2 สร้าง 1 รีเลชัน สำหรับ Temporal Fact Type โดยนำคีย์หลักจากรีเลชันทั้งสองในขั้นตอนที่ 1 มาเป็นคีย์นอก และ กำหนดรูปแบบของ Valid Time ที่ระบุอยู่บน Temporal Fact Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Valid Time เริ่มต้น (VTs) และ Valid Time สิ้นสุด สำหรับคีย์หลักของรีเลชันก็คือแอทริบิวท์ที่เป็นคีย์นอกทั้งสองรวมกับ Valid Time เริ่มต้น

จากรูปที่ 3.28 แปลงเป็นรีเลชัน ตามขั้นตอนทั้งสองที่กล่าวมาได้รีเลชัน TEACHER รีเลชัน SUBJECT และ รีเลชัน TEACHER_SUBJECT ดังนี้

TEACHER

TID	LSs	Lse
T1	2540-01-01	9999-12-31
T2	2540-01-01	9999-12-31

SUBLECT

SUBID	LSs	Lse
CS105	2530-01-01	9999-12-31
CS105	2530-01-01	9999-12-31

TEACHER_SUBJECT

TID	SUBID	VTs	VTe
T1	CS105	2540-01-01	9999-12-31
T1	CS444	2540-01-01	9999-12-31
T2	CS444	2540-01-01	9999-12-31
T2	CS105	2544-01-01	9999-12-31

รูปที่ 3.29 แสดง 3 รีเลชันที่ได้จากการแปลงรูปทีไนแอมที่ 3.28

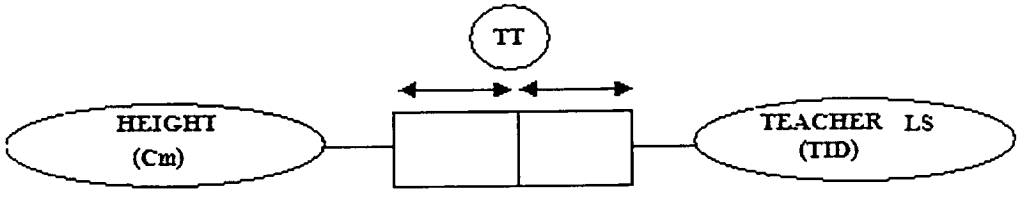
จากรูปที่ 3.29 อธิบายได้ว่า ในแต่ละ Life span ของรหัสอาจารย์(TID) มีความสัมพันธ์กับ แต่ละ Life span ของรหัสวิชา แบบกลุ่มต่อกลุ่ม ในแต่ละ Valid Time

3.4.3 กรณี Temporal Fact Type มี Transaction Time (TT) ระบุอยู่

สำหรับกรณีนี้มีลักษณะการแปลงเป็นรีเลชันที่อยู่ในรูปของ SQL มาตรฐานทุกๆเงื่อนไขเหมือนกับกรณีTemporal Fact Type ที่มี Valid Time ระบุอยู่ สำหรับTemporal Fact Type ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบุ Transaction Time นี้จะเกิดขึ้น ขึ้นกับApplication นั้น ๆว่ามีความเป็นไปได้ที่จะเกิดขึ้นได้ เช่น User เคยเชื่อว่าอาจารย์ สมชายมีความสูง 168 เซนติเมตร ซึ่งต่อมา User พบว่า อาจารย์สมชาย มีความสูงเป็น 187 เซนติเมตร เป็นต้น แสดงตัวอย่างดังรูปที่ 3.30



รูปที่ 3. 30 แสดง Temporal Fact Typeที่ระบุ Uniqueness Constraint แบบหนึ่งต่อหนึ่ง ที่ระบุ Transaction Time (TT)

จากรูปที่ 3.30 แสดงตัวอย่างการแสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง ระหว่าง Temporal Entity Type TEACHER(อาจารย์) และ Entity Type HEIGHT(ความสูง) โดยมีความหมายว่า ในแต่ละ Life span ของอาจารย์แต่ละคน มีความสูงได้ค่าเดียวในแต่ละ Transaction Time ดังที่ได้กล่าวมาแล้วว่า กรณีของการระบุ Transaction Time บน Temporal Fact Type นั้นขึ้นอยู่กับ Application นั้น ๆว่ามีความเป็นไปได้ที่จะเกิดขึ้นได้ เช่นเดียวกับตัวอย่างที่ได้กล่าวมา จากรูปที่ 3.30 แปลงอยู่ในรูปของรีเลชัน ซึ่งมีขั้นตอนเช่นเดียวกับขั้นตอนการแปลง Temporal Fact Type ที่ระบุ Valid Time ในกรณีที่ Temporal Entity Type มีความสัมพันธ์กับ Entity Type หรือ Label Type

TEACHER

TID	LSs	Lse
T1	2540-01-01	9999-12-31
T2	2540-01-01	9999-12-31

TEACHER_HEIGHT

TID	Cm	TTs	TTe
T1	168	2540-01-01	2544-01-01
T1	187	2544-01-01	9999-12-31
T2	170	2540-01-01	9999-12-31

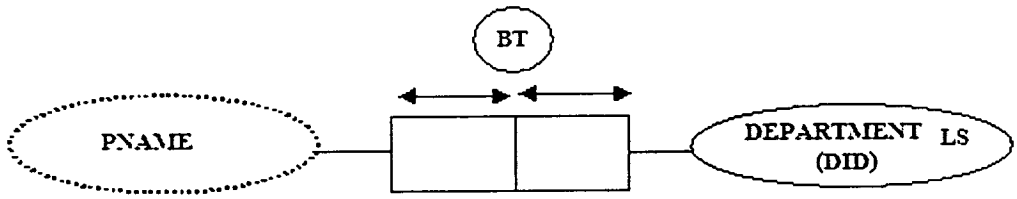
รูปที่ 3.31 แสดงรีเลชัน 2 รีเลชัน ที่แปลงจากรูปทีไนแอมที่ 3.30

3.4.4 กรณี Temporal Fact Type มีทั้ง Valid Time และ Transaction Time (BT) ระบุอยู่

สำหรับกรณีนี้มีลักษณะเช่นเดียวกับขั้นตอนการแปลง Temporal Fact Type ที่มี Valid Time ระบุอยู่ เพียงแต่กรณีนี้สนใจทั้งเวลาเมื่อ Fact นั้นเป็นจริง(Valid Time) และเวลาที่จัดเก็บข้อมูลลงสู่ฐานข้อมูล (Transaction Time) ซึ่งการระบุลักษณะของ Valid Time และ Transaction

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Time ที่ระบุอยู่บน Temporal Fact Type ก็จะอยู่ในรูปของ Valid Time เริ่มต้น (VTs) Valid Time สิ้นสุด (VTe) Transaction Time เริ่มต้น (TTs) และ Transaction Time สิ้นสุด (TTe)



รูปที่ 3.32 แสดง Temporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง ที่ระบุทั้ง Valid Time และ Transaction Time

จากรูปที่ 3.32 แปลงเป็นรีเลชัน ซึ่งมีขั้นตอนการแปลงเช่นเดียวกับ ขั้นตอนการแปลงTemporal Fact Type ที่มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง และมี Valid Time ระบุอยู่ในกรณีที่ Temporal Entity Type มีความสัมพันธ์กับ Entity Type หรือ Label Type

DEPARTMENT

TID	LSs	LSe
D1	2530-01-01	9999-12-31
D2	2530-01-01	9999-12-31

DEPARTMENT_NAME

TID	NAME	VTs	Vte	TTs	TTe
D1	Computer Science	2530-01-01	2546-01-01	2546-01-01	2546-01-01
D1	Information Technology	2546-01-01	9999-12-31	2546-01-01	9999-12-31

รูปที่ 3.33 แสดงรีเลชัน 2 รีเลชัน ที่แปลงจากรูปทีไนแอมที่ 3.32

3.5 ขั้นตอนการแปลง ทีไนแอมเป็นฐานข้อมูลเชิงสัมพันธ์ที่อยู่ในรูปของ Temporal Database ประกอบด้วย

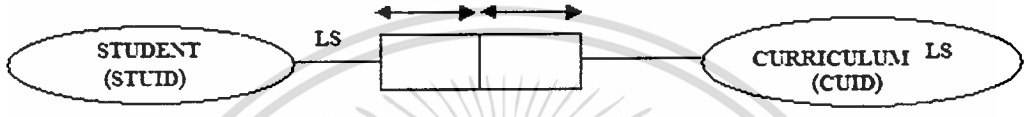
ขั้นตอนการแปลงฐานข้อมูลทีไนแอมเป็นTemporal Database [1][9] จะมีรูปแบบเช่นเดียวกับการแปลงทีไนแอม เป็นฐานข้อมูลเชิงสัมพันธ์เพียงแต่ฐานข้อมูลเชิงสัมพันธ์จะไม่การพิจารณารูปแบบเวลาที่ระบุอยู่ ประกอบด้วยกรณีต่าง ๆ เมื่อพิจารณาจาก Temporal Fact Type ได้แก่ กรณี Temporal Fact Type ที่มี Life span ระบุอยู่ กรณี Temporal Fact Type ที่มี Valid Time ระบุอยู่ กรณี Temporal Fact Type ที่มี Transaction Time ระบุอยู่ และ กรณีที่ Temporal Fact Type มีทั้ง Valid Time และ Transaction Time ระบุอยู่ หรือเรียกว่า Bitemporal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 .1 กรณี Temporal Fact Type มี Life span (LS) ระบุอยู่

กรณี Temporal Fact Type ที่มี Life span ระบุอยู่ แบ่งการพิจารณาออกได้เป็น 3 กรณี คือ กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบ หนึ่งต่อหนึ่ง กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบ หนึ่งต่อกลุ่ม และ กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบกลุ่มต่อกลุ่ม

3.5.1.1 กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อหนึ่ง



รูปที่ 3.34 แสดงตัวอย่าง Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อหนึ่ง

จากกรณีนี้ขั้นตอนของการแปลงเป็นฐานข้อมูลเชิงสัมพันธ์ที่อยู่ในรูปของTemporal Database มาตรฐาน มีขั้นตอนดังนี้ คือ สร้าง 1 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายใน โดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลักของรีเลชัน และ นำ Unique Identifier ของTemporal Entity Type ที่ Life span ระบุอยู่ภายนอกเป็น Attribute ของรีเลชัน

จากรูปที่ 3.34 แปลงเป็นรีเลชันที่อยู่ในรูปของTemporal Database ตามขั้นตอนทั้งสองที่กล่าวมาได้รีเลชันCURRICULUM ดังนี้

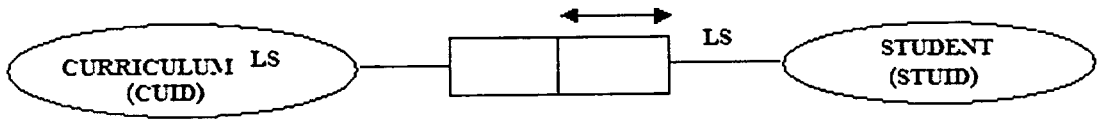
CURRICULUM

CUID	STUID
CU1	450364578
CU2	450364555

รูปที่ 3.35 แสดงรีเลชัน CURRICULUM ที่แปลงจากรูปที่ 3.34 อยู่ในรูปของTemporal Database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.1.2 กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อกลุ่ม



รูปที่ 3.36 แสดงตัวอย่าง Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อกลุ่ม

จากกรณีนี้ขั้นตอนของการแปลงเป็นฐานข้อมูลเชิงสัมพันธ์ที่อยู่ในรูปของTemporal Database มีขั้นตอนดังนี้

ขั้นตอนที่ 1 สร้าง 1 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายใน โดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลักของรีเลชัน

ขั้นตอนที่ 2 สร้าง 1 รีเลชันสำหรับ Temporal Fact Type โดยนำ Unique Identifier ของ Temporal Entity Type ที่ Life span ระบุอยู่ภายนอกเป็นคีย์หลักของรีเลชัน และ นำคีย์หลักของรีเลชันจากขั้นตอนที่ 1 มาเป็นคีย์นอกของรีเลชัน

จากรูปที่ 3.36 แปลงเป็นรีเลชันที่อยู่ในรูปของTemporal Database ตามขั้นตอนทั้งสองที่กล่าวมาได้รีเลชันCURRICULUM และ รีเลชัน STUDENT ดังนี้

CURRICURUM

CUID
CU1
CU2

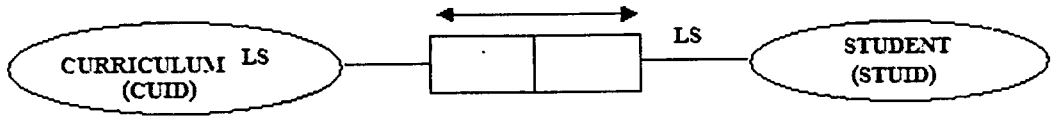
STUDENT

STUID	CUID
450364578	CU1
450364555	CU1
450364579	CU2

รูปที่ 3.37 แสดง 2 รีเลชัน ที่แปลงจากรูปที่ 3.36 อยู่ในรูป ของTemporal Database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.1.3 กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบกลุ่มต่อกลุ่ม



รูปที่ 3.38 แสดงตัวอย่าง Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบกลุ่มต่อกลุ่ม

จากกรณีนี้ขั้นตอนของการแปลงเป็นฐานข้อมูลเชิงสัมพันธ์ที่อยู่ในรูปของTemporal Database มีขั้นตอนดังนี้

ขั้นตอนที่ 1 สร้าง 1 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายใน โดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลักของรีเลชัน

ขั้นตอนที่ 2 สร้าง 1 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายนอก โดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลักของรีเลชัน

ขั้นตอนที่ 3 สร้าง 1 รีเลชันสำหรับ Temporal Fact Type โดยนำคีย์หลักจากขั้นตอนที่หนึ่งและขั้นตอนที่ 2 มาเป็นคีย์นอก สำหรับคีย์หลักของรีเลชันนี้คือแอททริบิวท์ทั้งสองที่กล่าวมารวมกันเป็นคีย์หลักของรีเลชัน

จากรูปที่ 3.38แปลงเป็นรีเลชันที่อยู่ในรูปของTemporal Database ได้รีเลชันCURRICULUM รีเลชัน STUDENT และ รีเลชัน CURICURUM_STUDENT ดังนี้

CURRICULUM

CUID
CU1
CU2

STUDENT

STUID
450364578
450364555

CURRICULUM_STUDENT

CUID	STUID
CU1	450364578
CU2	450364578
CU2	450364555

รูปที่ 3.39 แสดงรีเลชัน 3 รีเลชัน ที่แปลงจากรูปที่ 3.38 อยู่ในรูป ของ Temporal Database

3.5 .2 กรณี Temporal Fact Type มี Valid Time (VT) ระบุอยู่

กรณี Temporal Fact Type ที่มี Valid Time ระบุอยู่ แบ่งการพิจารณาออกได้เป็น 3 กรณี คือ กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบ หนึ่งต่อหนึ่ง กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบ หนึ่งต่อกลุ่ม และ กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบกลุ่มต่อกลุ่ม

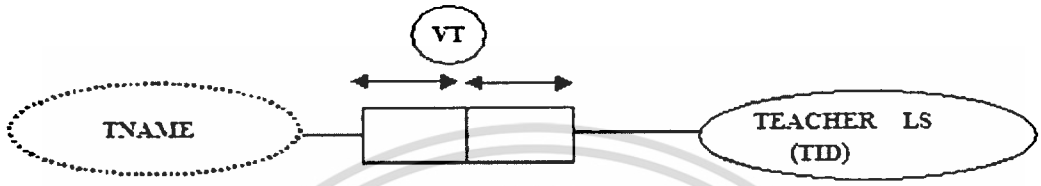
ซึ่งประกอบด้วยกรณีต่าง ๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2.1 กรณี Temporal Fact Type มีความสัมพันธ์ แบบหนึ่งต่อหนึ่ง

กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อหนึ่ง พิจารณาได้ 2 กรณี ได้แก่ กรณี Temporal Entity Type มีความสัมพันธ์กับ Entity Type หรือ Label Type กรณีที่สอง กรณี Temporal Entity Type มีความสัมพันธ์กับ Temporal Entity Type

1. กรณี Temporal Entity Type มีความสัมพันธ์กับ Entity Type หรือ Label Type



รูปที่ 3.40 แสดงตัวอย่าง Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อหนึ่ง

จากกรณีนี้ขั้นตอนของการแปลงเป็นฐานข้อมูลเชิงสัมพันธ์ที่อยู่ในรูปของ SQLมาตรฐาน มีขั้นตอนดังนี้คือ สร้าง 1 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายใน โดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลักของรีเลชัน และนำชื่อของ Label Type หรือ Unique Identifier ของ Entity Type ที่ระบุอยู่เป็น Attribute ของรีเลชันด้วย

จากรูปที่ 3.40 แปลงเป็นรีเลชันที่อยู่ในรูปของ Temporal Database ตามขั้นตอนที่กล่าวมาได้รีเลชัน TEACHER_NAME ดังนี้

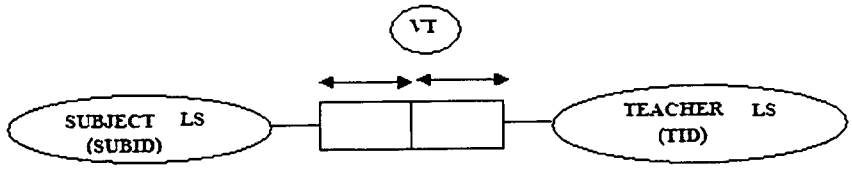
TEACHER_NAME

TID	TNAME
T1	SOMCHAI
T1	SOMPONG
T2	SUCHADA

รูปที่ 3.41 แสดงรีเลชันรีเลชัน ที่แปลงจากรูปที่ 3.40 อยู่ในรูปของTemporal Database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กรณี Temporal Entity Type มีความสัมพันธ์กับ Temporal Entity Type



รูปที่ 3.42 แสดงตัวอย่าง Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อหนึ่ง

จากกรณีนี้ขั้นตอนของการแปลงเป็นฐานข้อมูลเชิงสัมพันธ์ที่อยู่ในรูปTemporal Database มีขั้นตอนดังนี้ คือ สร้าง 1 รีเลชัน สำหรับ Temporal Fact Type โดยนำ Unique Identifier ของ Temporal Entity Type มาเป็นแอทริบิวท์ของรีเลชัน สำหรับคีย์หลักของรีเลชันก็คือ Attributeใด Attribute หนึ่ง

จากรูปที่ 3.42 แปลงเป็นรีเลชันที่อยู่ในรูปของ Temporal Database ตามขั้นตอนที่กล่าวมาได้รีเลชัน TEACHER_SUBJECT ดังนี้

TEACHER_SUBJECT

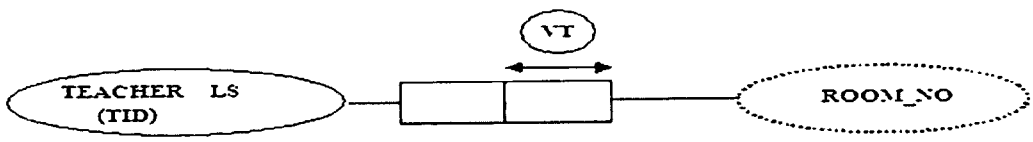
TID	SUBID
T1	CS105
T2	CS444

รูปที่ 3.43 แสดงรีเลชันที่ได้จากการแปลงรูปที่ 3.42 อยู่ในรูปของTemporal Database

3.5.2.2 กรณี Temporal Fact Type ระบุ Uniqueness Constraint แบบหนึ่งต่อกลุ่ม

กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อกลุ่ม พิจารณาได้ 2 กรณี ได้แก่ กรณี Temporal Entity Type มีความสัมพันธ์กับ Entity Type หรือ Label Type กรณีที่สอง กรณี Temporal Entity Type มีความสัมพันธ์กับ Temporal Entity Type

1 กรณี Temporal Entity Type มีความสัมพันธ์กับ Entity Type หรือ Label Type



รูปที่ 3.44 แสดง Temporal Entity Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อกลุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากกรณีนี้ขั้นตอนของการแปลงเป็นฐานข้อมูลเชิงสัมพันธ์ที่อยู่ในรูปของ Temporal Database มีขั้นตอนดังนี้

ขั้นตอนที่ 1 สร้าง 1 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายในโดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลัก

ขั้นตอนที่ 2 สร้าง 1 รีเลชัน สำหรับ Temporal Fact Type โดยนำชื่อของ Label Type หรือ Unique Identifier ของ Entity Type ที่ระบุอยู่เป็นคีย์หลักของรีเลชัน

จากรูปที่ 3.44 แปลงเป็นรีเลชันที่อยู่ในรูปของTemporal Database ตามขั้นตอนทั้งสองที่กล่าวมาได้รีเลชัน TEACHER และ TEACHER_ROOM ดังนี้

TEACHER

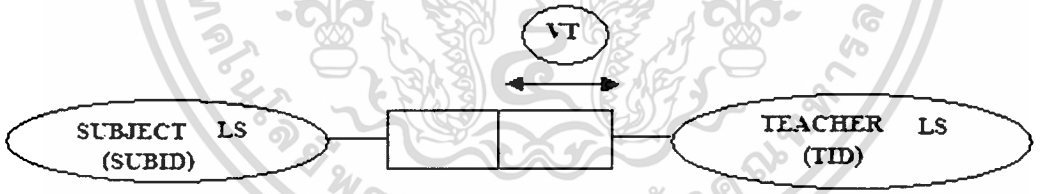
TID
T1
T2

TEACHER_ROOM

TID	ROOM_NO
T1	ROOM1
T1	ROOM2
T2	ROOM3

รูปที่ 3.45 แสดง 3 รีเลชันที่ได้จากการแปลง รูปที่ 3.44 ที่อยู่ในรูปTemporal Database

2. กรณี Temporal Entity Type มีความสัมพันธ์กับ Temporal Entity Type



รูปที่ 3.46 แสดง Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อกลุ่ม

จากกรณีนี้ขั้นตอนของการแปลงเป็นฐานข้อมูลเชิงสัมพันธ์ที่อยู่ในรูปของTemporal Database มีขั้นตอนดังนี้

ขั้นตอนที่ 1 สร้าง 1 รีเลชันสำหรับ Temporal Entity Type ที่เกาะติดกับTemporal Fact Type ด้านที่ไม่ได้ระบุรูปแบบ Valid Time โดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลักของรีเลชัน

ขั้นตอนที่ 2 สร้าง 1 รีเลชัน สำหรับ Temporal Fact Type ที่เกาะติดกับ Temporal Fact Type ด้านที่ระบุรูปแบบ Valid Time โดยนำคีย์หลักจากรีเลชันในขั้นตอนที่ 1 มาเป็นคีย์นอก สำหรับคีย์หลักของรีเลชันก็คือUnique Identifier ของ Temporal Entity Type

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.46 แปลงเป็นรีเลชันที่อยู่ในรูปของTemporal Database ตามขั้นตอนทั้งสองที่กล่าวมาได้รีเลชัน TEACHER และ รีเลชัน SUBJECT ดังนี้

SUBJECT

SUBID
CS105
CS444

TEACHER

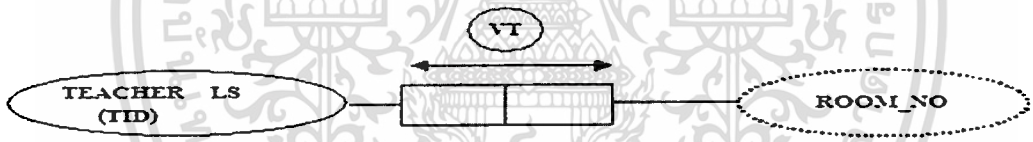
SUBID	TID
CS105	T1
CS105	T2

รูปที่ 3.47 แสดง 3 รีเลชัน ที่ได้จากการแปลงรูปที่ 3.46 ในรูปของTemporal Database

3.5.2.3 กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบกลุ่มต่อกลุ่ม

กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบกลุ่มต่อกลุ่ม พิจารณาได้ 2 กรณี ได้แก่ กรณี Temporal Entity Type มีความสัมพันธ์กับ Entity Type หรือ Label Type กรณีที่สอง กรณี Temporal Entity Type มีความสัมพันธ์กับ Temporal Entity Type

1. กรณี Temporal Entity Type มีความสัมพันธ์กับ Entity Type หรือ Label Type



รูปที่ 3.48 แสดง Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบกลุ่ม ต่อกลุ่ม

จากกรณีนี้ขั้นตอนของการแปลงเป็นฐานข้อมูลเชิงสัมพันธ์ที่อยู่ในรูปของTemporal Database มีขั้นตอนดังนี้

ขั้นตอนที่ 1 สร้าง 1 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายในโดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลักของรีเลชัน

ขั้นตอนที่ 2 สร้าง 1 รีเลชัน สำหรับ Temporal Fact Type โดยนำชื่อของ Label Type หรือ Unique identifier ที่ระบุอยู่เป็น Attribute ของรีเลชันและ นำคีย์หลักจากขั้นตอนที่ 1 เป็นคีย์นอก สำหรับคีย์หลักของรีเลชันก็คือ Attribute ที่เป็นคีย์นอกรวมกับชื่อของ Label Type หรือ Unique Identifier

จากรูปที่ 3.48 แปลงเป็นรีเลชันที่อยู่ในรูปของTemporal Database ตามขั้นตอนทั้งสองที่กล่าวมาได้รีเลชัน TEACHER และ TEACHER_ROOM ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TEACHER

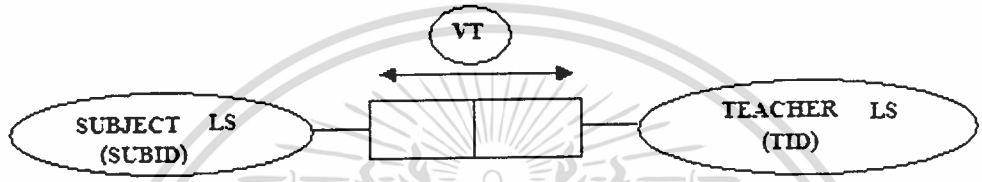
TID
T1
T2

TEACHER_ROOM

TID	ADDRESS
T1	ROOM1
T1	ROOM2
T2	ROOM2

รูปที่ 3.49 แสดง 2 รีเลชันที่ได้จากการแปลง รูปที่ 3.48 ที่อยู่ในรูปTemporal Database

2. กรณี Temporal Entity Type มีความสัมพันธ์กับ Temporal Entity Type



รูปที่ 3.50 แสดงตัวอย่าง Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบกลุ่มต่อกลุ่ม

จากกรณีนี้ขั้นตอนของการแปลงเป็นฐานข้อมูลเชิงสัมพันธ์ที่อยู่ในรูปของ Temporal Database มีขั้นตอนดังนี้

ขั้นตอนที่ 1 สร้าง 2 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายในโดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลักของรีเลชัน

ขั้นตอนที่ 2 สร้าง 1 รีเลชัน สำหรับ Temporal Fact Type โดยนำคีย์หลักจากรีเลชันทั้งสองในขั้นตอนที่ 1 มาเป็นคีย์นอก และรวมกันเป็นคีย์หลักของรีเลชัน

จากรูปที่ 3.49 แปลงเป็นรีเลชันที่อยู่ในรูปของTemporal Database ตามขั้นตอนทั้งสองที่กล่าวมาได้รีเลชัน TEACHER รีเลชัน SUBJECT และ รีเลชัน TEACHER_SUBJECT ดังนี้

TEACHER

TID
T1
T2

SUBLECT

SUBID
CS105
CS105

TEACHER_SUBJECT

TID	SUBID
T1	CS105
T1	CS444
T2	CS444
T2	CS105

รูปที่ 3.51 แสดง 3 รีเลชันที่ได้จากการแปลงรูปที่ 3.50 ที่อยู่ในรูปของTemporal Database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.3 กรณี Temporal Fact Type มี Transaction Time (TT) ระบุอยู่ สำหรับกรณีนี้มีลักษณะการแปลงเป็นรีเลชันที่อยู่ในรูปของTemporal Database ทุกๆเงื่อนไขเหมือนกับกรณี Temporal Fact Type ที่มี Valid Time ระบุอยู่ สำหรับTemporal Fact Type ที่ระบุ Transaction Time นี้จะเกิดขึ้น ขึ้นกับApplication นั้น ๆว่ามีความเป็นไปได้ที่จะเกิดขึ้นได้

3.5.4 กรณี Temporal Fact Type มีทั้ง Valid Time และ Transaction Time (BT) ระบุอยู่ สำหรับกรณีนี้มีลักษณะเช่นเดียวกับขั้นตอนการแปลง Temporal Fact Type ที่มี Valid Time ระบุอยู่ เพียงแต่กรณีนี้สนใจทั้งเวลาเมื่อ Fact นั้นเป็นจริง(Valid Time) และเวลาที่จัดเก็บข้อมูลลงสู่ฐานข้อมูล (Transaction Time)

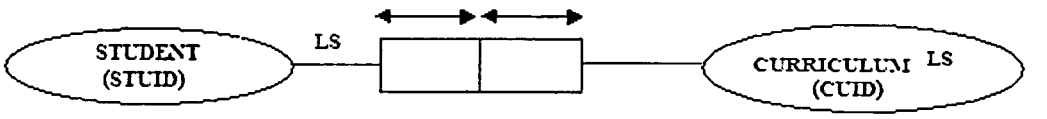
3.6 ขั้นตอนการแปลง ทีไนแอมเป็นฐานข้อมูลเชิงสัมพันธ์ที่อยู่ในรูปของ Object Relational Database

ขั้นตอนการแปลงฐานข้อมูลทีไนแอมเป็นฐานข้อมูลเชิงสัมพันธ์ที่อยู่ในรูปของ Object Relational Database [8] [11] [13]ประกอบด้วยกรณีต่าง ๆ เมื่อพิจารณาจาก Temporal Fact Type ได้แก่ กรณี Temporal Fact Type ที่มี Life span ระบุอยู่ กรณี Temporal Fact Type ที่มี Valid Time ระบุอยู่ กรณี Temporal Fact Type ที่มี Transaction Time ระบุอยู่ และ กรณีที่ Temporal Fact Type มีทั้ง Valid Time และ Transaction Time ระบุอยู่ หรือเรียกว่า Bitemporal

3.6 .1 กรณี Temporal Fact Type มี Life span (LS) ระบุอยู่

กรณี Temporal Fact Type ที่มี Life span ระบุอยู่ แบ่งการพิจารณาออกได้เป็น 3 กรณี คือ กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบ หนึ่งต่อหนึ่ง กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบ หนึ่งต่อกลุ่ม และ กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบกลุ่มต่อกลุ่ม

3.6.1.1 กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อหนึ่ง



รูปที่ 3.52 แสดงตัวอย่าง Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากกรณีนี้ขั้นตอนของการแปลงเป็น Object Relational Database มีขั้นตอนคือ สร้าง 1 รีเลชัน สำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายใน โดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลักของรีเลชัน และ กำหนดรูปแบบ Life span ที่ระบุอยู่ในTemporal Entity Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Life span เริ่มต้น (LSs) และ Life span สิ้นสุด (LSe) และนำ Unique Identifier ของ Temporal Entity Type ที่ Life span ระบุอยู่ภายนอกเป็น Attribute ของรีเลชันด้วย

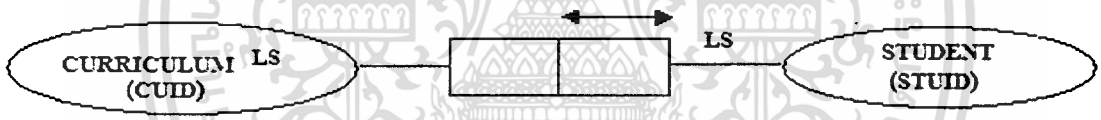
จากรูปที่ 3.52 แปลงเป็นObject Relational Database ตามขั้นตอนที่กล่าวมาได้รีเลชัน ดังนี้

CURRICULUM

CUID	LSs	LSe	STUID
CU1	2540-01-01	9999-12-31	450364578
CU2	2540-01-01	9999-12-31	450364555

รูปที่ 3.53 แสดงรีเลชันที่แปลงจากรูปที่ 3.52 อยู่ในรูปของ Object Relational Database

3.6.1.2 กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อกลุ่ม



รูปที่ 3.54 แสดงตัวอย่าง Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อกลุ่ม

จากกรณีนี้ขั้นตอนของการแปลงเป็น Object Relational Database มีขั้นตอนคือ

ขั้นตอนที่ 1 สร้าง 1 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายในโดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลักของรีเลชัน และกำหนดรูปแบบ Life span ที่ระบุอยู่ในTemporal Entity Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Life span เริ่มต้น (LSs) และ Life span สิ้นสุด (LSe)

ขั้นตอนที่ 2 สร้าง 1 รีเลชันสำหรับ Temporal Fact Type โดยนำ Unique Identifier ของ Temporal Entity Type ที่ Life span ระบุอยู่ภายนอกเป็นคีย์หลักของรีเลชัน และ นำคีย์หลักของรีเลชันจากขั้นตอนที่ 1 มาเป็นคีย์นอกของรีเลชัน

จากรูปที่ 3.54 แปลงเป็น Object Relational Database ตามขั้นตอนทั้งสองที่กล่าวมาได้รีเลชัน CURRICULUM และ รีเลชัน STUDENT ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CURRICURUM

CUID	LSs	Lse
CU1	2540-01-01	9999-12-31
CU2	2540-01-01	9999-12-31

STUDENT

STUID	CUID
450364578	CU1
450364555	CU1
450364579	CU2

รูปที่ 3.55 แสดง 2 รีเลชัน ที่แปลงจากรูปที่ 3.54 อยู่ในรูป Object Relational Database

3.6.1.3 กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบกลุ่มต่อกลุ่ม

รูปที่ 3.56 แสดงตัวอย่าง Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบกลุ่มต่อกลุ่ม

จากกรณีนี้ขั้นตอนของการแปลงเป็น Object Relational Database มีขั้นตอนดังนี้

ขั้นตอนที่ 1 สร้าง 1 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายใน โดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลักของรีเลชัน และกำหนดรูปแบบ Life span ที่ระบุอยู่ในTemporal Entity Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Life span เริ่มต้น (LSs) และ Life span สิ้นสุด (LSe)

ขั้นตอนที่ 2 สร้าง 1 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายนอก โดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลักของรีเลชัน

ขั้นตอนที่ 3 สร้าง 1 รีเลชันสำหรับ Temporal Fact Type โดยนำคีย์หลักจากขั้นตอนที่หนึ่งมาเป็นคีย์นอก และ นำคีย์หลักจากขั้นตอนที่ 2 มาเป็นคีย์นอก สำหรับคีย์หลักของรีเลชันนี้คือแอททริบิวท์ทั้งสองที่กล่าวมารวมกันเป็นคีย์หลักของรีเลชัน

จากรูปที่ 3.56 แปลงเป็น Object Relational Database ตามขั้นตอนทั้งสามที่กล่าวมาได้รีเลชัน CURRICULUM รีเลชัน STUDENT และ รีเลชัน CURICURUM_STUDENT ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CURRICULUM

CUID	LSs	Lse
CU1	2540-01-01	9999-12-31
CU2	2540-01-01	9999-12-31

STUDENT

STUID
450364578
450364555

CURRICULUM_STUDENT

CUID	STUID
CU1	450364578
CU2	450364578
CU2	450364555

รูปที่ 3.57 แสดง 3 รีเลชัน ที่แปลงจากรูปที่ 3.56 อยู่ในรูปObject Relational Database

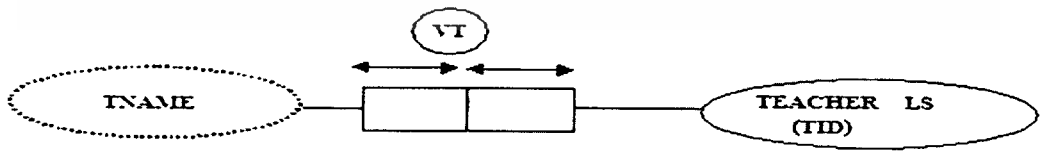
3.6.2 กรณี Temporal Fact Type มี Valid Time (VT) ระบุอยู่

กรณี Temporal Fact Type ที่มี Valid Time ระบุอยู่ แบ่งการพิจารณาออกได้เป็น 3 กรณี คือ กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบ หนึ่งต่อหนึ่ง กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบ หนึ่งต่อกลุ่ม และ กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบกลุ่มต่อกลุ่ม

3.6.2.1 กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อหนึ่ง

กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อหนึ่ง พิจารณาได้ 2 กรณี ได้แก่ กรณี Temporal Entity Type มีความสัมพันธ์กับ Entity Type หรือ Label Type กรณีที่สอง กรณี Temporal Entity Type มีความสัมพันธ์กับ Temporal Entity Type

1. กรณี Temporal Entity Type มีความสัมพันธ์กับ Entity Type หรือ Label Type



รูปที่ 3.58 แสดงตัวอย่าง Temporal Fact Type ที่ระบุ uniqueness แบบหนึ่งต่อหนึ่ง

จากกรณีนี้ขั้นตอนของการแปลงเป็นObject Relational Database มีขั้นตอนคือ สร้าง 1 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายใน โดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลักของรีเลชัน และ กำหนดรูปแบบ Life span

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ระบุอยู่ในTemporal Entity Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Life span เริ่มต้น (LSs) และ Life span สิ้นสุด (LSe) และนำชื่อของ Label Type หรือ Unique Identifier ของ Entity Type ที่ระบุอยู่เป็น Attribute ของรีเลชัน รวมทั้งกำหนดรูปแบบของ Valid Time ที่ระบุอยู่บน Temporal Fact Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Valid Time เริ่มต้น (VTs) และ Valid Time สิ้นสุด

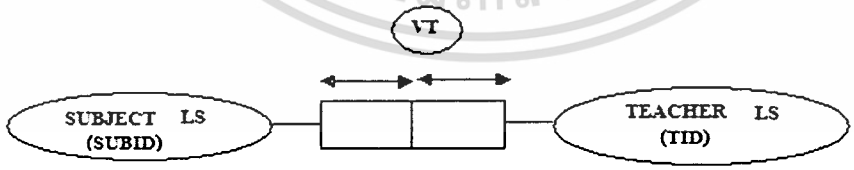
จากรูปที่ 3.58 แปลงเป็นObject Relational Database ตามขั้นตอนที่กล่าวมาได้รีเลชัน TEACHER ดังนี้

รูปที่ 3.59 แสดงรีเลชัน STUDENT ที่แปลงจากรูปที่ 3.58 อยู่ในรูป Object Relational Database

TID	LSs	LSe	TNAME	VTs	VTe
T1	2540-01-01	9999-12-31	SOMCHAI	2540-01-01	2544-01-01
			SOMPONG	2544-01-01	9999-12-31
T2	2540-01-01	9999-12-31	SUCHADA	2540-01-01	9999-12-31

รูปที่ 3.60 แสดงตัวอย่างการระบุข้อมูลใน รีเลชัน STUDENT จากรูปที่ 3.59

2. กรณี Temporal Entity Type มีความสัมพันธ์กับ Temporal Entity Type



รูปที่ 3.61 แสดงตัวอย่าง Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อหนึ่ง

จากกรณีนี้ขั้นตอนของการแปลงเป็นObject Relational Database มีขั้นตอนดังนี้

ขั้นตอนที่ 1 สร้าง 2 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายในโดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลักของรีเลชัน และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดรูปแบบ Life span ที่ระบุอยู่ในTemporal Entity Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Life span เริ่มต้น (LSs) และ Life span สิ้นสุด (LSe)

ขั้นตอนที่ 2 สร้าง 1 รีเลชัน สำหรับ Temporal Fact Type โดยนำคีย์หลักจากรีเลชันทั้งสองในขั้นตอนที่ 1 มาเป็นคีย์นอก และ กำหนดรูปแบบของ Valid Time ที่ระบุอยู่บน Temporal Fact Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Valid Time เริ่มต้น (VTs) และ Valid Time สิ้นสุด สำหรับคีย์หลักของรีเลชันก็คือ Attribute ที่เป็นคีย์นอกแอททริบิวท์ใด Attribute หนึ่งรวมกับ Valid Time เริ่มต้น

จากรูปที่ 3.61 แปลงเป็นรีเลชันที่อยู่ในรูปของ Object Relational Database ตามขั้นตอนทั้งสองที่กล่าวมาได้รีเลชัน TEACHER รีเลชัน SUBJECT และ รีเลชัน TEACHER_SUBJECT ดังนี้

TEACHER

TID	LSs	Lse
T1	2540-01-01	9999-01-01
T2	2540-01-01	9999-01-01

SUBJECT

SUBID	LSs	LSe
CS105	2530-01-01	9999-01-01
CS444	2530-01-01	9999-01-01

TEACHER_SUBJECT

TID	SUBID	VTs	Vte
T1	CS105	2540-01-01	2544-01-01
T1	CS444	2544-01-01	9999-01-01
T2	CS444	2540-01-01	2544-01-01
T2	CS105	2544-01-01	9999-01-01

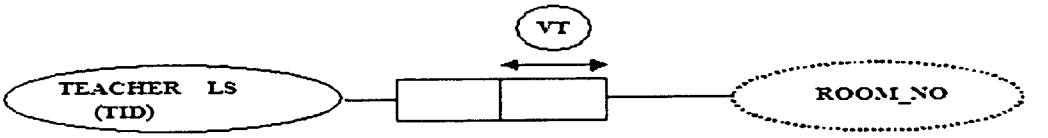
รูปที่ 3.62 แสดง 3 รีเลชันที่ได้จากการแปลงรูปที่ 3.61 อยู่ในรูปของObject Relational Database

3.6.2.2 กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อกลุ่ม

กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อกลุ่ม พิจารณาได้ 2 กรณี ได้แก่ กรณี Temporal Entity Type มีความสัมพันธ์กับ Entity Type หรือ Label Type กรณีที่สอง กรณี Temporal Entity Type มีความสัมพันธ์กับ Temporal Entity Type

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1 กรณี Temporal Entity Type มีความสัมพันธ์กับ Entity Type หรือ Label Type



รูปที่ 3.63 แสดง Temporal Entity Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อกลุ่ม

จากกรณีนี้ขั้นตอนของการแปลงเป็นObject Relational Database มีขั้นตอนดังนี้ คือ สร้าง 1 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายใน โดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลักของรีเลชัน และ กำหนดรูปแบบ Life span ที่ระบุอยู่ในTemporal Entity Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Life span เริ่มต้น (LSs) และ Life span สิ้นสุด (LSe) และ นำชื่อของ Label Type หรือ Unique Identifier ของ Entity Type ที่ระบุอยู่เป็น Attribute ของรีเลชัน และ กำหนดรูปแบบของ Valid Time ที่ระบุอยู่บน Temporal Fact Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Valid Time เริ่มต้น (VTs) และ Valid Time สิ้นสุด

จากรูปที่ 3.63 แปลงเป็นObject Relational Database ตามขั้นตอนที่กล่าวมาได้รีเลชัน TEACHER ดังนี้

TID	LSs	LSe	ROOM_NO	VTs	VTe

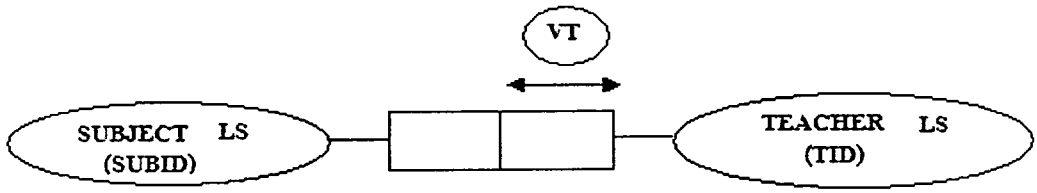
รูปที่ 3.64 แสดงรีเลชัน TEACHER ที่แปลงจากรูปที่ 3.63 อยู่ในรูป Object Relational Database

TID	LSs	LSe	ROOM_NO	VTS	VTE
T1	2540-01-01	9999-12-31	ROOM1	2540-01-01	2544-01-01
			ROOM2	2544-01-01	9999-12-31
			ROOM3	2544-01-01	9999-12-31
T2	2540-01-01	9999-12-31	ROOM1	2544-01-01	9999-12-31

รูปที่ 3.65 แสดงตัวอย่างข้อมูล รีเลชัน TEACHER จากรูปที่ 3.63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กรณี Temporal Entity Type มีความสัมพันธ์กับ Temporal Entity Type



รูปที่ 3.66 แสดง Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบหนึ่งต่อกลุ่ม

จากกรณีนี้ขั้นตอนของการแปลงเป็นObject Relational Database มีขั้นตอนดังนี้

ขั้นตอนที่ 1 สร้าง 2 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายในโดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลัก ของรีเลชัน และกำหนดรูปแบบ Life span ที่ระบุอยู่ในTemporal Entity Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Life span เริ่มต้น (LSs) และ Life span สิ้นสุด (LSe)

ขั้นตอนที่ 2 สร้าง 1 รีเลชัน สำหรับ Temporal Fact Type โดยนำคีย์หลักจากรีเลชันทั้งสองในขั้นตอนที่ 1 มาเป็นคีย์นอก และ กำหนดรูปแบบของ Valid Time ที่ระบุอยู่บน Temporal Fact Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Valid Time เริ่มต้น (VTs) และ Valid Time สิ้นสุด สำหรับคีย์หลักของรีเลชันก็คือ Attribute ที่เป็นคีย์นอกฝั่งกลุ่ม (Many) รวมกับ Valid Time เริ่มต้น

จากรูปที่ 3.66 แปลงเป็น Object Relational Database ตามขั้นตอนทั้งสองที่กล่าวมาได้รีเลชัน TEACHER รีเลชัน SUBJECT และ รีเลชัน TEACHER_SUBJECT ดังนี้

SUBJECT

SUBID	LSs	LSe
CS105	2530-01-01	9999-12-31
CS444	2530-01-01	9999-12-31

TEACHER

TID	LSs	Lse
T1	2540-01-01	9999-12-31
T1	2540-01-01	9999-12-31

SUBJECT_TEACHER

SUBID	TID	VTs	VTe
CS105	T1	2540-01-01	9999-12-31
CS444	T2	2540-01-01	2544-01-01
CS105	T2	2544-01-01	9999-12-31

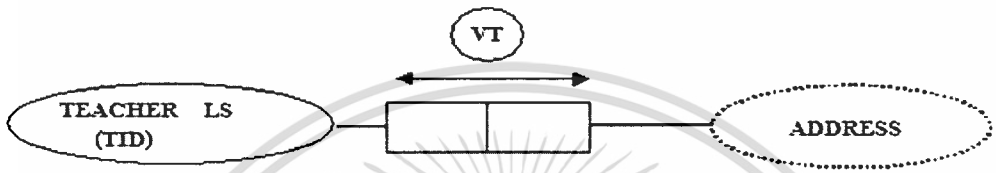
รูปที่ 3.67 แสดง 3 รีเลชัน ที่ได้จากการแปลงรูปที่ 3.66 ในรูปของ Object Relational Database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.2.3 กรณี Temporal Fact Type มีความสัมพันธ์แบบกลุ่มต่อกลุ่ม

กรณี Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบกลุ่มต่อกลุ่ม พิจารณาได้ 2 กรณี ได้แก่ กรณี Temporal Entity Type มีความสัมพันธ์กับ Entity Type หรือ Label Type กรณีที่สอง กรณี Temporal Entity Type มีความสัมพันธ์กับ Temporal Entity Type

1. กรณี Temporal Entity Type มีความสัมพันธ์กับ Entity Type หรือ Label Type



รูปที่ 3.68 แสดง Temporal Fact Type ระบุ Uniqueness Constraint แบบกลุ่ม ต่อกลุ่ม

จากกรณีนี้ขั้นตอนของการแปลงเป็น Object Relational Database มีขั้นตอนดังนี้ คือ สร้าง 1 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายใน โดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลักของรีเลชัน และ กำหนดรูปแบบ Life span ที่ระบุอยู่ในTemporal Entity Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Life span เริ่มต้น (LSs) และ Life span สิ้นสุด (LSe) และ นำชื่อของ Label Type หรือ Unique identifier ที่ระบุอยู่เป็น Attribute ของรีเลชัน และ กำหนดรูปแบบของ Valid Time ที่ระบุอยู่บน Temporal Fact Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Valid Time เริ่มต้น (VTs) และ Valid Time สิ้นสุด

จากรูปที่ 3.68 แปลงเป็นObject Relational Database ตามขั้นตอนที่กล่าวมาได้รีเลชัน TEACHER ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

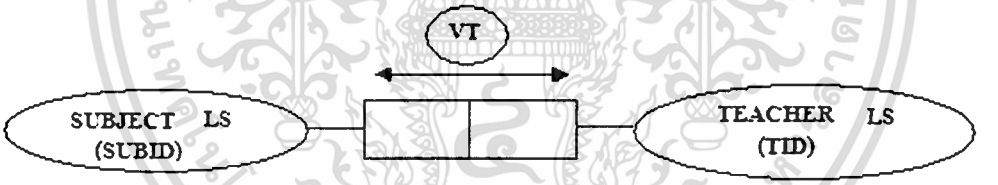
TID	LSs	LSe	ROOM_NO	VTs	VTe

รูปที่ 3.69 แสดงรีเลชันที่ได้จากการแปลง รูปที่ 3.68 ที่อยู่ในรูปObject Relational Database

TID	LSs	LSe	ROOM_NO	VTs	VTe
T1	2540-01-01	9999-12-31	ROOM1	2540-01-01	2544-01-01
			ROOM2	2544-01-01	9999-12-31
			ROOM3	2545-01-01	9999-12-31
T2	2540-01-01	9999-12-31	ROOM2	2540-01-01	9999-12-31

รูปที่ 3.70 แสดงตัวอย่างการระบุข้อมูลใน รีเลชัน TEACHER จากรูปที่ 3.69

2. กรณี Temporal Entity Type มีความสัมพันธ์กับ Temporal Entity Type



รูปที่ 3.71 แสดงตัวอย่าง Temporal Fact Type ที่ระบุ Uniqueness Constraint แบบกลุ่มต่อกลุ่ม

จากกรณีนี้ขั้นตอนของการแปลงเป็นObject Relational Database มีขั้นตอนดังนี้

ขั้นตอนที่ 1 สร้าง 2 รีเลชันสำหรับ Temporal Entity Type ที่มี Life span ระบุอยู่ภายในโดยกำหนดให้ Unique Identifier ของ Temporal Entity Type นั้นเป็นคีย์หลัก ของรีเลชัน และกำหนดรูปแบบ Life span ที่ระบุอยู่ในTemporal Entity Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Life span เริ่มต้น (LSs) และ Life span สิ้นสุด (LSe)

ขั้นตอนที่ 2 สร้าง 1 รีเลชัน สำหรับ Temporal Fact Type โดยนำคีย์หลักจากรีเลชันทั้งสองในขั้นตอนที่ 1 มาเป็นคีย์นอก และ กำหนดรูปแบบของ Valid Time ที่ระบุอยู่บน Temporal Fact Type เป็น Attribute ของรีเลชันที่กำหนดอยู่ในรูปของ Valid Time เริ่มต้น (VTs) และ Valid

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Time สิ้นสุด สำหรับคีย์หลักของรีเลชันก็คือ Attribute ที่เป็นคีย์นอกทั้งสองรวมกับ Valid Time เริ่มต้น

จากรูปที่ 3.71 แปลงเป็นObject Relational Database ตามขั้นตอนทั้งสองที่กล่าวมาได้รีเลชัน TEACHER รีเลชัน SUBJECT และ รีเลชัน TEACHER_SUBJECT ดังนี้

TEACHER

TID	LSs	Lse
T1	2540-01-01	9999-21-31
T2	2540-01-01	9999-21-31

SUBLECT

SUBID	LSs	LSe
CS105	2530-01-01	9999-21-31
CS105	2530-01-01	9999-21-31

TEACHER_SUBJECT

TID	SUBID	VTs	Vte
T1	CS105	2540-01-01	9999-21-31
T1	CS444	2540-01-01	9999-21-31
T2	CS444	2540-01-01	9999-21-31
T2	CS105	2544-01-01	9999-21-31

รูปที่ 3.72 แสดง 3 รีเลชันที่ได้จากการแปลงรูปที่ 3.71 เป็น Object Relational Database

3.6.3 กรณี Temporal Fact Type มี Transaction Time (TT) ระบุอยู่ สำหรับกรณีนี้มีลักษณะการแปลงทีไนแอมเป็นObject Relational Database ทุกๆเงื่อนไขเหมือนกับกรณีTemporal Fact Type ที่มี Valid Time ระบุอยู่เพียงแต่เปลี่ยนจากรูปแบบ Valid Time เป็น รูปแบบ Transaction Time สำหรับTemporal Fact Type ที่ระบุ Transaction Time นี้จะเกิดขึ้น ขึ้นกับ Application นั้น ๆว่ามีความเป็นไปได้ที่จะเกิดขึ้นได้

3.6.4 กรณี Temporal Fact Type มีทั้ง Valid Time และ Transaction Time (BT) ระบุอยู่

สำหรับขั้นตอนการแปลงทีไนแอม เป็น Object Relational Database กรณีนี้มีลักษณะเช่นเดียวกับขั้นตอนการแปลงแปลงทีไนแอมของ Temporal Fact Type ที่มี Valid Time ระบุอยู่ เพียงแต่กรณีนี้สนใจทั้งเวลาเมื่อ Fact นั้นเป็นจริง(Valid Time) และเวลาที่จัดเก็บข้อมูลลงสู่ฐานข้อมูล (Transaction Time) ซึ่งการระบุลักษณะของ Valid Time และ Transaction Time ที่ระบุอยู่บน Temporal Fact Type ก็จะอยู่ในรูปของ Valid Time เริ่มต้น (VTs) Valid Time สิ้นสุด (VTe) Transaction Time เริ่มต้น (TTs) และ Transaction Time สิ้นสุด (TTe)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

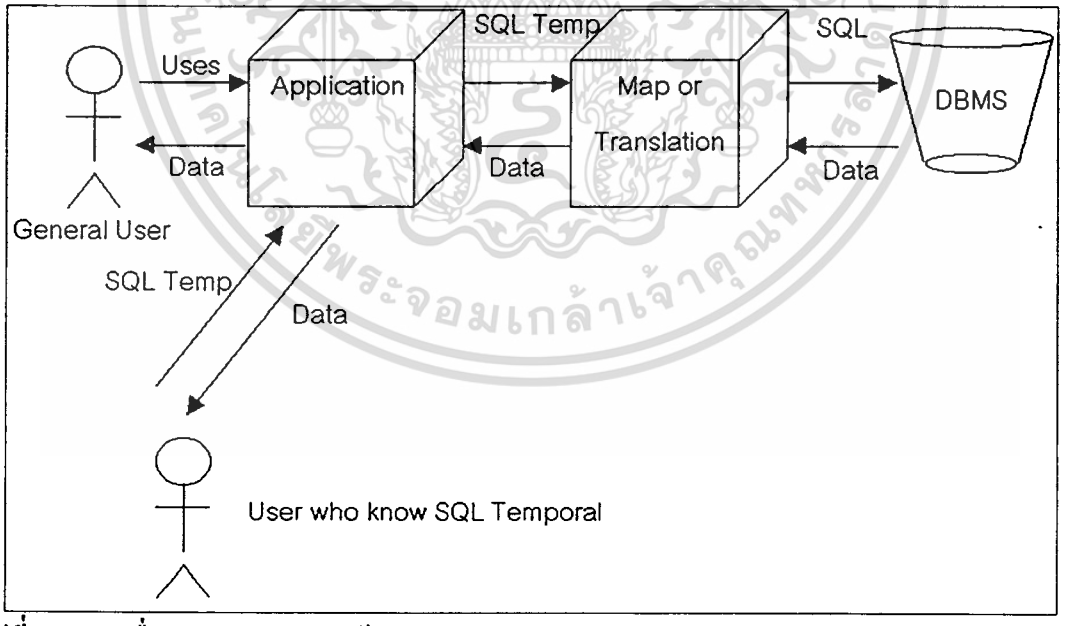
บทที่ 4

การปฏิบัติการบนฐานข้อมูลเชิงเวลา

(Temporal Database Operation)

ข้อมูลที่ถูกเก็บอยู่ในฐานข้อมูลเชิงเวลาจะมีความแตกต่างจากข้อมูลที่เก็บในฐานข้อมูลทั่ว ๆ ไป ตรงที่จะมีการเพิ่มส่วนของเวลาติดไปกับส่วนของข้อมูล เพื่อเก็บช่วงของเวลาที่ข้อมูลนี้เป็นจริง (Valid time) หรือช่วงเวลาที่ข้อมูลถูกเก็บอยู่ในฐานข้อมูล (Transaction time) เวลาที่ถูกเก็บอยู่ในฐานข้อมูลนั้นแบ่งได้ 3 ชนิดดังได้แก่ Valid Time ของความจริง(Fact) , Transaction Time ของความจริงในฐานข้อมูล และ User Defined Time ซึ่งหมายถึง คุณสมบัติของความจริงที่ไม่ต้องแปลความหมายของวันหรือเวลา อย่างเช่น วันเกิด หรือแรงงาน ซึ่งเป็นคุณสมบัติที่ไม่ต้องการการจัดการทำให้ผู้ใช้จะต้องจัดการเอง ไม่ว่าจะเป็น Insert ,Delete และ Update

ในการนำเอา Valid Time หรือ Transaction Time มาใช้ได้อย่างมีประสิทธิภาพนั้น จะต้องมีการจัดการกับการเชื่อมต่อของเวลากับฐานข้อมูล เพื่อให้ง่ายต่อความเข้าใจและไม่เกิดความสับสนแก่ผู้ใช้โดยสามารถทำได้ดังรูปที่ 4.1 [9]



รูปที่ 4.1 การเชื่อมต่อเวลากับฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาษา SQL/Temporal เป็นภาษา SQL ที่สนับสนุนการทำงานที่มีเวลาเข้ามาเกี่ยวข้องและเข้าใจความหมายของ Valid time และ Transaction time ในตัวภาษาอยู่แล้ว แตกต่างจาก SQL แบบเดิมที่ไม่เข้าใจความหมายของ Valid time และ Transaction time

4.1 การจัดการกับ Valid-Time State Table (Modifying Valid-Time State Tables)

การแสดงการจัดการการเปลี่ยนแปลงเกี่ยวกับ Valid-Time State Table ทั้ง current และ sequence ในงานวิจัยนี้จะใช้ตัวอย่างข้อมูลโปรแกรมวิชาจากระบบการบริการข้อมูลนักศึกษาสำหรับอาจารย์ที่ปรึกษา ซึ่งแต่ละโปรแกรมวิชามีชื่อเรียกแตกต่างกันในแต่ละช่วงเวลา โดยใช้ DEP_NAME(ชื่อโปรแกรมวิชา) เป็นการแสดงการเปลี่ยนแปลง และใช้ FROM_ DATE และ TO_ DATE เป็นคาบเวลาที่แสดงว่าแถวนั้นเป็นจริงอยู่ จากตัวอย่างข้อมูลในตารางที่ 4.1 จะพบว่าที่ DEP_NO ที่ 101 DEP_NAME เป็น การศึกษาเคมี ตั้งแต่ (FROM_DATE) วันที่ 1 มกราคม 2541 จนถึง วันที่ 23 มีนาคม 2541 และตั้งแต่ วันที่ 23 มีนาคม 2541 จนถึงปัจจุบัน(TO_DATE มีค่าเป็น 999-12-31) DEP_NO 101 มี DEP_NAME เป็น เคมี และสำหรับรหัสโปรแกรมวิชา 234 มีชื่อโปรแกรมวิชา คอมพิวเตอร์ ตั้งแต่วันที่ 17 กุมภาพันธ์ 2541 จนถึงปัจจุบัน การแทนค่า TO_DATE ที่เป็นค่าเป็นจริงอยู่ในปัจจุบันใน SQL-92 จะแทนด้วย 9999-12-31 ซึ่งใน DBMS บางตัวจะแทนปัจจุบันด้วย null value หรือบางตัวใช้เวลาในอดีตเช่น 1899 หรือ อาจใช้เวลาที่มากที่สุดที่เป็นไปได้ เช่น 9999 เป็นต้น สำหรับงานวิจัยนี้เลือกใช้การแทนค่า TO_DATE เช่นเดียวกับ SQL-92 ในการจัดการ(Modification)กับข้อมูลทั้งในลักษณะของข้อมูลทั่ว ๆ ไป และ การจัดการกับข้อมูลใน Valid time state table อธิบายได้ดังนี้

ตารางที่ 4.1 ตารางDEPARTMENT

DEP_NO	DEP_NAME	FROM_DATE	TO_DATE
101	การศึกษาเคมี	2541-01-01	2541-03-23
101	เคมี	2541-03-23	9999-12-31
234	คอมพิวเตอร์	2541-02-17	9999-12-31
799	ฟิสิกส์	2541-03-12	9999-12-31

4.1.1 Current Modification

เป็นลักษณะของการจัดการหรือการกระทำกับข้อมูลจากปัจจุบันไปสู่การจัดการหรือการกระทำกับข้อมูลในอนาคต ประกอบด้วย Current INSERT , Current DELETE และ Current UPDATE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1.1 Current INSERT

คำสั่งแบบ Current INSERT ลักษณะคำสั่งเป็นการสร้างTimestamp จากปัจจุบันไปยังอนาคต ซึ่งลักษณะคำสั่ง Current INSERT ของ Valid Time State เหมือนกับคำสั่งการเพิ่มข้อมูลทั่ว ๆไป ตัวอย่างเช่น ต้องการเพิ่มข้อมูล "433,'คอมพิวเตอร์ศึกษา',CURRENT _DATE,DATE '9999-12-31'" ลงในตารางDEPARTMENT เขียนคำสั่งได้ดังรูปที่ 4.2

```
INSERT INTO  DEPARTMENT
VALUES (433,'คอมพิวเตอร์ศึกษา',CURRENT _DATE,DATE '9999-12-31')
```

รูปที่ 4.2 แสดงตัวอย่างการ INSERT

4.1.1.2 Current DELETE

คำสั่ง Current DELETE ในแนวคิดถ้าไม่สนใจเวลาที่เกาะติดอยู่กับข้อมูล ลักษณะคำสั่งก็เหมือนกับคำสั่งการลบโดยทั่ว ๆ เช่นถ้าต้องการลบ DEP_NO 101 ออกไปจากตารางที่ 4.1 โดยที่ไม่สนใจคอลัมน์ที่เป็นเวลา มีลักษณะคำสั่งเป็นดังนี้

```
DELETE FROM  DEPARTMENT
        WHERE  DEP_NO = 101
```

รูปที่ 4.3 แสดงตัวอย่างการ DELETE

ในคำสั่ง Current DELETE ของ Current Time State Table พิจารณาเป็น 2 กรณี ได้แก่ Current DELETE General scenario และ Current DELETE Restricted scenario

1. Current DELETE General scenario

คำสั่งการลบข้อมูลแบบ Current DELETE General scenario สำหรับ Time State Table ก็คือการ UPDATE และ DELETE ข้อมูล ตัวอย่างเช่น ถ้าต้องการ Current DELETE General scenario ข้อมูล DEP_NO 234 จากตารางที่ 4.2 แสดงคำสั่งได้ดังรูปที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 General Scenario ของ DEPARTMENT

DEP_NO	DEP_NAME	FROM_DATE	TO_DATE
101	การศึกษาเคมี	2541-01-01	2541-03-23
101	เคมี	2541-03-23	9999-12-31
234	คอมพิวเตอร์	2541-02-17	2541-10-17
234	วิทยาการคอมพิวเตอร์	2541-10-17	9999-12-31
799	ฟิสิกส์	2541-03-12	9999-12-31

```
UPDATE DEPARTMENT
        SET TO_DATE = CURRENT _DATE
        WHERE DEP_NO = 234
        AND TO_DATE >= CURRENT_DATE
        AND FROM _DATE < CURRENT _DATE

DELETE FROM DEPARTMENT
        WHERE DEP_NO = 234
        AND FROM _DATE > CURRENT _DATE
```

รูปที่ 4.4. แสดงตัวอย่างการเขียน Current DELETE General scenario

จากคำสั่งที่แสดงในรูปที่ 4.4 ซึ่งแสดง Current DELETE General scenario ซึ่งการทำงานประกอบด้วยคำสั่ง DELETE และ UPDATE ในการทำงานตามลักษณะของ Current DELETE General scenario นั้นสามารถสามารถทำคำสั่ง DELETE หรือ UPDATE คำสั่งใดก่อนก็ได้

ตารางที่ 4.3 ผลของ Current DELETE General scenario จากตารางที่ 4.2

DEP_NO	DEP_NAME	FROM_DATE	TO_DATE
101	การศึกษาเคมี	2541-01-01	2541-03-23
101	เคมี	2541-03-23	9999-12-31
234	คอมพิวเตอร์	2541-02-17	2541-07-29
799	การศึกษาฟิสิกส์	2541-03-12	9999-12-31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Current DELETE Restricted scenario

การลบข้อมูลแบบ Current DELETE Restricted scenario บน Valid Time state table ก็คือการ UPDATE ข้อมูล โดย Current DELETE จะมีการกระทำจากปัจจุบันจนตลอดกาล ดังรูปที่ 4.5

```
UPDATE DEPARTMENT
        SET TO_DATE = CURRENT _DATE
        WHERE DEP_NO = 101
        AND TO_DATE = DATE '9999-12-31'
```

รูปที่ 4.5 แสดงตัวอย่างคำสั่งการลบข้อมูลแบบ Current DELETE Restricted scenario

Current Modification	
DELETE	
Restricted Scenario	**General Scenario**
UPDATE DEPARTMENT SET TO_DATE = CURRENT _DATE WHERE DEP_NO = 101 AND TO_DATE = DATE '9999-12-31'	**UPDATE** DEPARTMENT SET TO_DATE = CURRENT _DATE WHERE DEP_NO = 234 AND TO_DATE >= CURRENT_DATE AND FROM _DATE < CURRENT _DATE **DELETE** DEPARTMENT WHERE DEP_NO= 234 AND FROM _DATE > CURRENT _DATE

รูปที่ 4.6 แสดงตัวอย่างการ Current Modification ของการ DELETE ทั้ง 2 กรณี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1.3 Current UPDATE

คำสั่งในการแก้ไขข้อมูลในแนวคิดถ้าไม่สนใจเวลาที่เกาะติดอยู่กับข้อมูลแล้ว ลักษณะคำสั่งก็เหมือนกับคำสั่งการแก้ไขโดยทั่ว ๆ เช่นถ้าต้องการแก้ไข DEP_NO 799 ของ DEPARTMENT จากตารางที่ 4.1 ให้เป็นข้อมูลใหม่ ลักษณะคำสั่งดังรูปที่ 4.7

```
UPDATE DEPARTMENT
        SET DEP_NAME = 'ฟิสิกส์'
        WHERE DEP_NO = 799
```

รูปที่ 4.7 แสดงตัวอย่างคำสั่งของการแก้ไขข้อมูลในกรณีที่ไม่สนใจเวลา

คำสั่ง Current UPDATE ของ Current Time State Table พิจารณาเป็น 2 กรณี ได้แก่ Current UPDATE General scenario และ Current UPDATE Restricted scenario

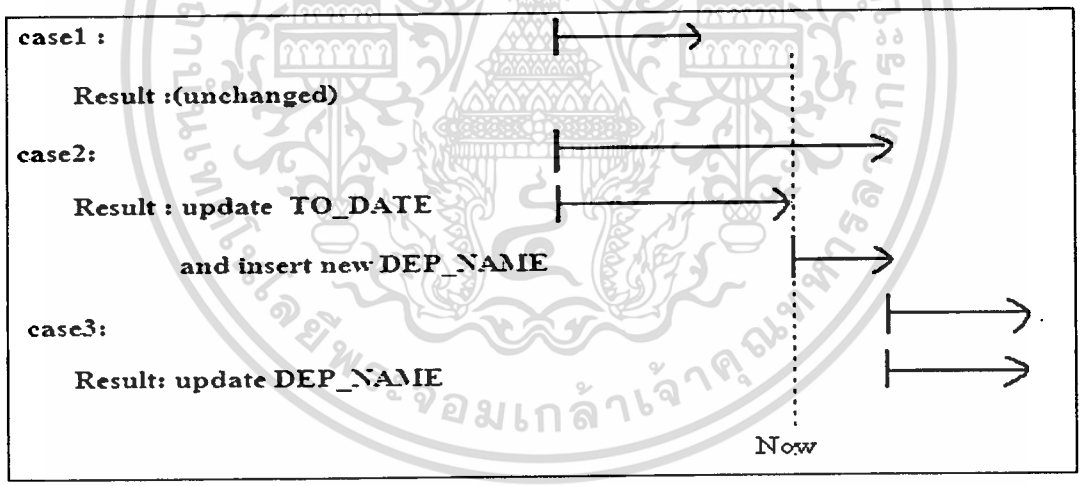
1. Current UPDATE General scenario

Current UPDATE ใน general scenario ประกอบด้วย 3 กรณีได้แก่ กรณีที่ 1 เมื่อช่วงเวลาที่ fact เป็นจริง (Period of validity) สิ้นสุดไปในอดีตซึ่งการ UPDATE จะไม่ส่งผลใด ๆ ต่อแถวนั้น กรณีที่ 2 ถ้าแถวนั้น currently valid ส่วนที่อยู่ก่อนเวลาปัจจุบันจะถูกทำลายและแถวใหม่จะถูกแทรก โดยมี period of validity เริ่มต้นที่วันปัจจุบัน และจะสิ้นสุดที่เวลาเดียวกับแถวเดิม กรณีที่ที่ 3 ถ้าแถวนั้นมีจุดเริ่มต้นในอนาคต แถวนั้นจะถูก UPDATE ทั้งหมด โดยวิธีดังกล่าวต้องใช้การ INSERT 1 ครั้ง และUPDATE 2 ครั้ง ซึ่งการ UPDATE ครั้งที่ 2 จะกระทำเมื่อใดก็ได้ แต่การ UPDATE ครั้งแรกจะเกิดขึ้นหลังจากการ INSERT ตัวอย่างคำสั่งในการ UPDATE แบบ General scenario แสดงได้ดังรูปที่ 4.8 และรูปที่ 4.9 แสดงทั้ง 3 กรณีของการ Current UPDATE General scenario

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
INSERT INTO DEPARTMENT
        SELECT DEP_NO,'ฟิสิกส์',CURRENT_DATE,TO_DATE
        FROM DEPARTMENT
        WHERE DEP_NO = 799
        AND FROM_DATE <= CURRENT_DATE
        AND TO_DATE > CURRENT_DATE
UPDATE DEPARTMENT
        SET TO_DATE = CURRENT_DATE
        WHERE DEP_NO = 799
        AND DEP_NO <> 'ฟิสิกส์'
        AND FROM_DATE < CURRENT_DATE
        AND TO_DATE > CURRENT_DATE
UPDATE DEPARTMENT
        SET DEP_NO = 'ฟิสิกส์'
        WHERE DEP_NO = 799
        AND FROM_DATE >= CURRENT_DATE
```

รูปที่ 4.8 แสดงตัวอย่างคำสั่งของการแก้ไขข้อมูลแบบ Current Update General scenario



รูปที่ 4.9 แสดง 3 กรณีของ Current UPDATE General scenario

2. Current UPDATE Restrict scenario

คำสั่งการ Current UPDATE แบบ Restrict scenario บน Valid Time state table ก็คือการ INSERT และ UPDATE ข้อมูล ตัวอย่างการเขียนคำสั่งในการแก้ไขข้อมูลแบบ Restrict scenario แสดงได้ดังรูปที่ 4.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
INSERT INTO DEPARTMENT
        SELECT DISTINCT 799,'ฟิสิกส์',CURRENT_DATE,DATE '9999-12-31'
        FROM  DEPARTMENT
        WHERE EXISTS (SELECT *
                      FROM  DEPARTMENT
                      WHERE  DEP_NO = 799
                             AND TO_DATE = DATE '9999-12-31')
UPDATE  DEPARTMENT
        SET TO_DATE = CURRENT_DATE
        WHERE  DEP_NO = 799
                      AND  DEP_NAME <> 'ฟิสิกส์'
                      AND TO_DATE = DATE '9999-12-31'
```

รูปที่ 4.10 แสดงตัวอย่างคำสั่งของการแก้ไขข้อมูลแบบ Current UPDATE Restrict scenario

Current Modification	
UPDATE	
Restricted Scenario	**General Scenario**
INSERT INTO DEPARTMENT SELECT DISTINCT 799,'ฟิสิกส์',CURRENT_DATE,DATE '9999-12-31' FROM DEPARTMENT WHERE EXISTS (SELECT * FROM DEPARTMENT WHERE DEP_NO = 799 AND TO_DATE = DATE '9999-12-31') **UPDATE** DEPARTMENT SET TO_DATE = CURRENT_DATE WHERE DEP_NO = 799 AND DEP_NAME <> 'ฟิสิกส์' AND TO_DATE = DATE '9999-12-31'	**INSERT** INTO DEPARTMENT SELECT DEP_NO,'ฟิสิกส์',CURRENT_DATE,TO_DATE FROM DEPARTMENT WHERE DEP_NO = 799 AND FROM_DATE <= CURRENT_DATE AND TO_DATE > CURRENT_DATE **UPDATE** DEPARTMENT SET TO_DATE = CURRENT_DATE WHERE DEP_NO = 799 AND DEP_NAME <> 'ฟิสิกส์' AND FROM_DATE < CURRENT_DATE AND TO_DATE > CURRENT_DATE **UPDATE** DEPARTMENT SET DEP_NAME = 'ฟิสิกส์' WHERE DEP_NO = 799 AND FROM_DATE >= CURRENT_DATE

รูปที่ 4.11 แสดงตัวอย่างการเขียน Current Modification ของการ Update ทั้ง 2 กรณี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 Sequenced Modification

Current Modification มีการกระทำจากปัจจุบันไปสู่อนาคต แต่ sequence modification จะเปลี่ยนแปลงบนคาบเวลาใด ๆ เรียกว่า period of applicability ช่วงเวลานี้สามารถเป็นอดีต ในอนาคตหรือมีการคาบเกี่ยว (Overlap) กับเวลาปัจจุบันก็ได้ การจัดการกับข้อมูลแบบ sequence modification จะใช้ CURRENT_DATE (วันที่ปัจจุบัน) แทนที่จุดเริ่มต้นของคาบเวลาใด ๆ และจุดสิ้นสุดแทนด้วย DATE'9999-12-31'(วันที่ในอนาคต) การจัดการกับข้อมูลแบบ Sequence modification ประกอบด้วย

4.1.2.1 Sequence INSERT

คำสั่งในการเพิ่มข้อมูลแบบ Sequence จะทำให้เกิดการเปลี่ยนแปลงบนคาบเวลาใด ๆ ตัวอย่างเช่นถ้าต้องการเพิ่มข้อมูล โปรแกรมวิชาที่ DEP_NO เป็น 426 ชื่อโปรแกรมวิชา (DEP_NAME) เป็น ชีววิทยา ตั้งแต่ (FROM_DATE) 26 มีนาคม 2541 จนถึง (TO_DATE) 14 เมษายน 2542 ลักษณะคำสั่งในการ INSERT จะมีลักษณะดังรูปที่ 4.12

```
INSERT INTO  DEPARTMENT
        VALUES (426,'ชีววิทยา',DATE'2541-03-26',DATE'2541-04-14)
```

รูปที่ 4.12 แสดงตัวอย่างการเขียนคำสั่ง INSERT ในรูปแบบ Sequence Modification

4.1.2.2 Sequence DELETE

สำหรับการจัดการกับข้อมูลแบบ Sequence Modification ประกอบด้วย 4 กรณีที่เป็นไปได้ ดังเงื่อนไขที่ว่าช่วงเวลาที่ fact เป็นจริง (Period of validity) ของแถวเดิม ทำการแสดงเหนือลำดับของการเปลี่ยนแปลงบนคาบเวลาใด ๆ(period of applicability) สำหรับการลบ ถ้า

กรณี 1 แถวเดิมจะครอบคลุมลำดับของการเปลี่ยนแปลงบนคาบเวลาใด ๆ ทั้งหมด ดังนั้นจะไม่มีการแก้ไข ในช่วงคาบเวลาเริ่มต้นและสิ้นสุด ในคาบเวลาเริ่มต้นทำได้โดยการกำหนดค่า TO_DATE ไปยังจุดเริ่มต้นของลำดับของการเปลี่ยนแปลงบนคาบเวลาใด ๆ ส่วนในช่วงคาบเวลาสิ้นสุด จะทำการINSERTเพิ่มขึ้น

กรณี 2 จะไม่ทำการแก้ไขในส่วนช่วงคาบเวลาเริ่มต้นของ ลำดับของการเปลี่ยนแปลงบนคาบเวลาใด ๆ ของแถวเดิมนั้น ซึ่งสมมาตรกัน

กรณี 3 จะไม่ทำการแก้ไขในส่วนสุดท้ายของช่วงเวลา

กรณี 4 ต้องทำการลบทั้งแถว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างคำสั่ง Sequence DELETE แสดงได้ดังรูปที่ 4.13 และรูปที่ 4.14 แสดงทั้ง 4 กรณีของ Sequence DELETE

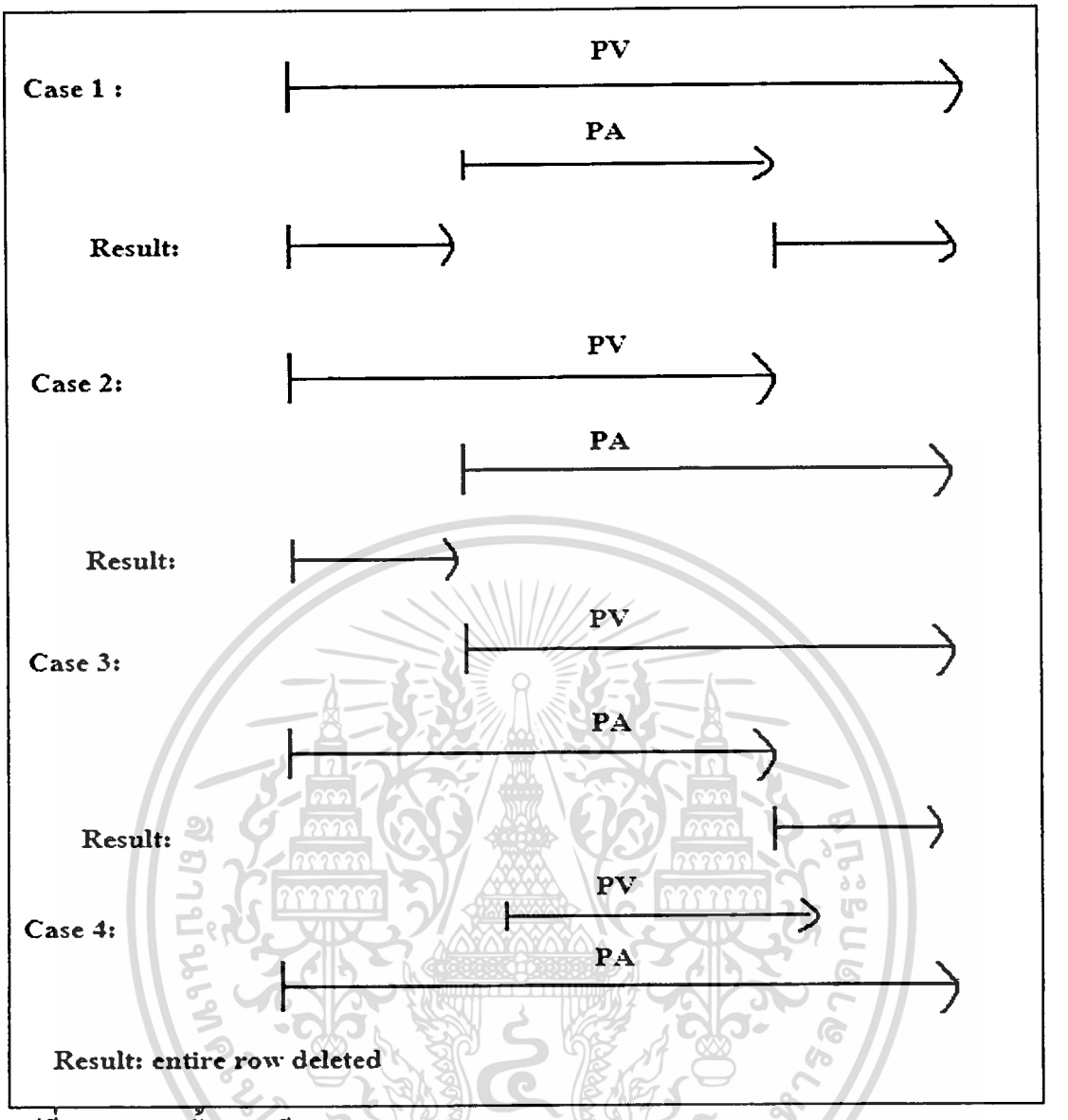
```
INSERT INTO DEPARTMENT
        SELECT DEP_NO, DEP_NAME ,DATE '2541-10-22',TO_DATE
        FROM DEPARTMENT
        WHERE DEP_NO = 234
                AND FROM_DATE <=        DATE '2541-10-22'
        UPDATE DEPARTMENT
        SET TO_DATE = DATE '2541-10-01'
        WHERE DEP_NO < 234
                AND FROM_DATE < '2541-10-01'
                AND TO_DATE >= DATE '2541-10-01'
        UPDATE DEPARTMENT
        SET FROM_DATE = DATE '2541-10-22'
        WHERE DEP_NO = 234
          AND FROM_DATE < DATE '2541-10-22'
                AND TO_DATE >= DATE'2541-10-22'
        DELETE FROM DEPARTMENT
        WHERE DEP_NO = 234
                AND FROM_DATE >= DATE '2541-10-01'
 AND TO_DATE <= DATE '2541-10-22'
```

รูปที่ 4.13 แสดงตัวอย่างคำสั่งในการทำ Sequence DELETE

เมื่อพิจารณาคำสั่ง Sequence Delete เปรียบเทียบกับทั้ง 4 กรณีของ Sequence Deletion สรุปได้ว่า

- กรณี 1ของ Sequence Delete จะมีผลกระทบต่อ 2 statements (INSERT INTO DEPARTMENT)แรกของคำสั่ง Sequence Deletion
- statement ที่ 2(UPDATE DEPARTMENT) จะครอบคลุม กรณี 2
- statement ที่ 3(UPDATE DEPARTMENT)จะครอบคลุมกรณี 3 และ
- statement ที่ 4 (DELETE FROM DEPARTMENT)จะครอบคลุมกรณี 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงทั้ง 4 กรณีของ Sequence Deletion

4.1.2.3 Sequence UPDATE

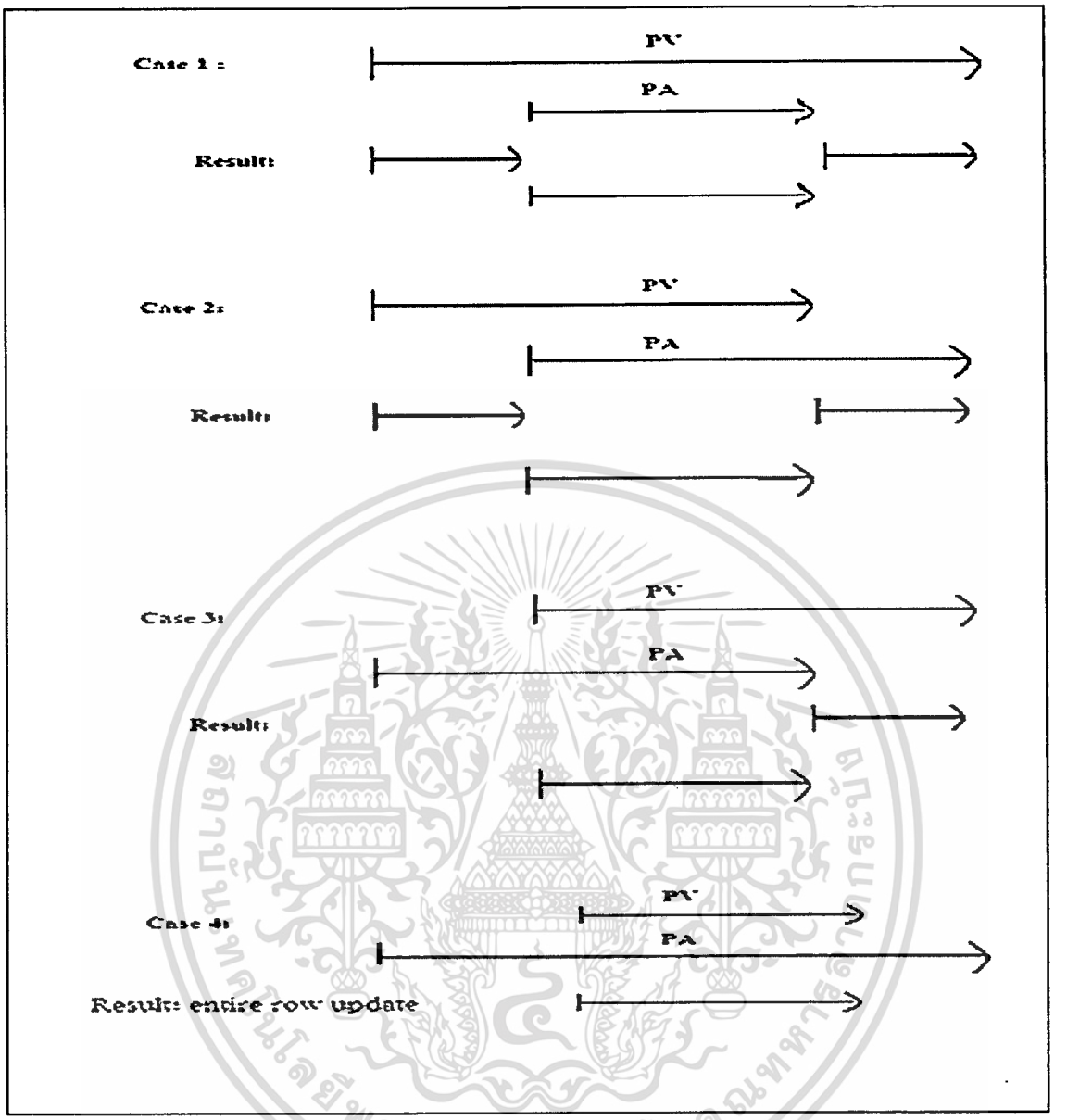
ตัวอย่างคำสั่งในการทำ Sequence UPDATE มีลักษณะดังรูปที่ 4.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
INSERT INTO DEPARTMENT
        SELECT DEP_NO,DEP_NAME,FROM_DATE,DATE '2541-03-01'
        FROM  DEPARTMENT
        WHERE  DEP_NO = 799
           AND FROM_DATE < DATE '2541-03-01'
           AND TO_DATE > DATE '2541-03-01'
INSERT INTO  DEPARTMENT
        SELECT  DEP_NO,DEP_NAME,DATE '2541-04-01',TO_DATE
        FROM  DEPARTMENT
        WHERE  DEP_NO = 799
           AND FROM_DATE < DATE '2541-04-01'
           AND TO_DATE > DATE '2541-04-01'
UPDATE   DEPARTMENT
        SET  DEP_NAME = 'ฟิสิกส์'
        WHERE  DEP_NO = 799
           AND FROM_DATE < DATE '2541-04-01'
           AND TO_DATE > DATE '2541-03-01'
UPDATE  DEPARTMENT
        SET  FROM_DATE = DATE '2541-03-01'
        WHERE  DEP_NO = 799
           AND FROM_DATE < DATE '2541-03-01'
           AND TO_DATE > DATE '2541-03-01'
UPDATE  DEPARTMENT
        SET TO_DATE = DATE '2541-04-01'
        WHERE  DEP_NO = 799
           AND FROM_DATE < DATE '2541-04-01'
           AND TO_DATE > DATE '2541-04-01'
```

รูปที่ 4.15 ตัวอย่างการเขียนคำสั่ง Sequence UPDATE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 ภาพแสดงแทนคำสั่ง Sequence Update

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การประยุกต์ทีไนแอม

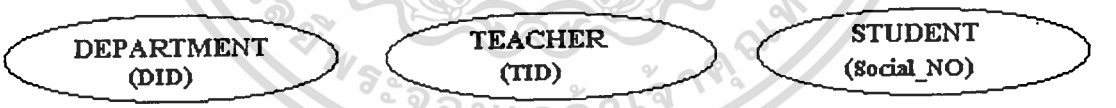
การประยุกต์ทีไนแอม ในบทที่ 5 นี้ เป็นการนำเอา กระบวนการออกแบบฐานข้อมูลเชิงเวลาด้วยทีไนแอมที่ได้นำเสนอในบทที่ 3 มาประยุกต์ใช้กับข้อมูลจริง ซึ่งระบบงานที่นำข้อมูลมานำเสนอ เป็นระบบงานบริการข้อมูลนักศึกษาสำหรับอาจารย์ที่ปรึกษา ของคณะวิทยาศาสตร์ สถาบันราชภัฏภูเก็ต ที่ใช้ฐานข้อมูลเชิงเวลา โดยใช้กระบวนการทีไนแอมในการออกแบบฐานข้อมูล ซึ่งแสดงรายละเอียดทุก ๆ ขั้นตอนตามกระบวนการของทีไนแอม ดังนี้

5.1 ขั้นตอนการออกแบบฐานข้อมูลเชิงเวลาด้วยวิธีการทีไนแอม

ขั้นตอนการออกแบบฐานข้อมูลเชิงเวลาของระบบบริการข้อมูลนักศึกษาสำหรับอาจารย์ที่ปรึกษา คณะวิทยาศาสตร์ และเทคโนโลยี สถาบันราชภัฏภูเก็ต ประกอบด้วย

5.1.1 ขั้นตอนการเขียนโครงร่างหลัก (Main Schema) ประกอบด้วย

ขั้นตอนที่ 1 ระบุ Temporal Entity Type ที่สนใจ สำหรับระบบการบริการข้อมูลนักศึกษาสำหรับอาจารย์ที่ปรึกษา Temporal Entity Type ที่สนใจได้แก่ โปรแกรมวิชา (DEPARTMENT) อาจารย์ที่ปรึกษา (TEACHER) และ นักศึกษา (STUDENT) แสดง Temporal Entity Type ที่สนใจแต่ละ Entity Type ได้ดังรูปที่ 5.1



รูปที่ 5.1 แสดง Temporal Entity Type ที่สนใจของระบบบริการข้อมูลนักศึกษาทั้งหมด

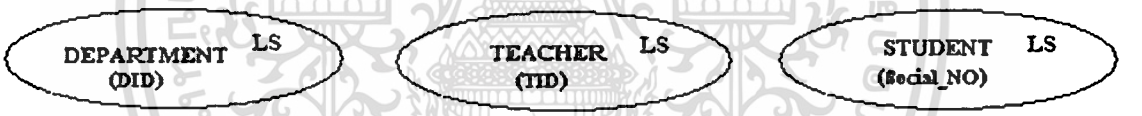
ขั้นตอนที่ 2 พิจารณา Temporal Entity Type จากขั้นตอนที่ 1 ว่าแต่ละ Temporal Entity Type เป็น Temporal Entity Type ชนิดที่ระบุ Life Span (LS) อยู่ภายใน Entity Type หรือ Temporal Entity Type ที่ระบุ Life Span อยู่ภายนอก Entity Type ดังได้กล่าวไว้ในหัวข้อที่ 3.2.1.1 และสำหรับระบบบริการข้อมูลนักศึกษาสำหรับอาจารย์ที่ปรึกษา Temporal Entity Type ที่ได้จากขั้นตอนที่ 1 พิจารณารูปแบบของเวลาที่จะระบุได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Temporal Entity Type DEPARTMENT ระบุรูปแบบเวลา Life Span อยู่ภายใน Entity Type เพราะว่า Temporal Entity Type DEPARTMENT สามารถที่จะระบุช่วงเวลาของการมีชีวิตได้ในตัวEntity Type ได้เองโดยไม่ขึ้นกับ Entity Type อื่น Life Span ที่กำหนดใน Temporal Entity Type DEPARTMENT คือ ช่วงเวลาของ Life span ของ Temporal Entity Type DEPARTMENT ตั้งแต่เริ่มมีโปรแกรมวิชา จนกระทั่งโปรแกรมวิชานั้นถูกยกเลิกไป

- Temporal Entity Type TEACHER ระบุรูปแบบเวลา Life Span อยู่ภายใน Entity Type เพราะว่า Temporal Entity Type TEACHER สามารถที่จะระบุช่วงเวลาของการมีชีวิตได้ในตัวEntity Type ได้เองโดยไม่ขึ้นกับ Entity Type อื่น Life Span ที่กำหนดใน Temporal Entity Type TEACHER คือ ช่วงเวลาของ Life span ของ Temporal Entity Type TEACHER ตั้งแต่อาจารย์เริ่มเข้าทำงาน จนกระทั่งอาจารย์เกษียณ หรือหมดสัญญาจ้าง

- Temporal Entity Type STUDENT ระบุรูปแบบเวลา Life Span อยู่ภายใน Entity Type เพราะว่า Temporal Entity Type STUDENT สามารถที่จะระบุช่วงเวลาของการมีชีวิตได้ในตัวEntity Type ได้เองโดยไม่ขึ้นกับ Entity Type อื่น Life Span ที่กำหนดใน Temporal Entity Type STUDENT คือ ช่วงเวลาของ Life span ของ Temporal Entity Type STUDENT ตั้งแต่นักศึกษาเริ่มเข้าศึกษาในสถาบัน จนกระทั่งนักศึกษาจบการศึกษา



รูปที่ 5.2 แสดง Temporal Entity Type ทั้งหมดจากขั้นตอนที่ 1 ที่ระบุ Life Span

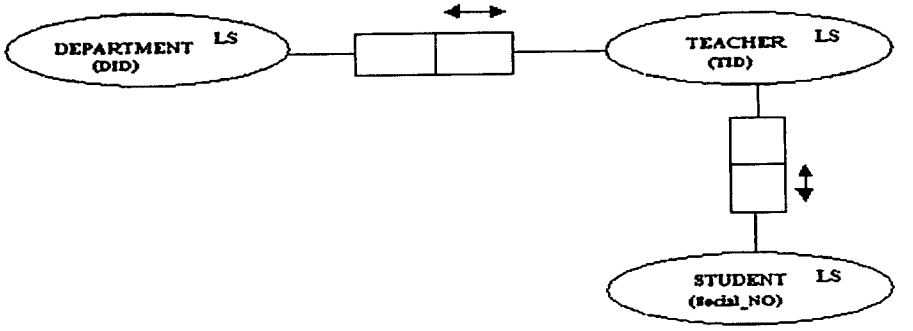
ขั้นตอนที่ 3 ระบุ Life Span ให้กับแต่ละให้กับแต่ละ Temporal Entity Type ตามชนิดของ Temporal Entity Type ที่ได้จากขั้นตอนที่ 2 ดังนั้นจากขั้นตอนที่ 2 จะได้ Temporal Entity Type ที่ระบุ Life Span ตามรูปแบบที่ได้กล่าวไว้ในหัวข้อที่ 3.2.1.1 ได้ดังรูปที่ 5.2

ขั้นตอนที่ 4 ระบุ Temporal Fact Type ให้กับ Temporal Entity Type ตามเงื่อนไขที่สนใจสำหรับระบบบริการข้อมูลนักศึกษา มีเงื่อนไขที่สนใจดังนี้

- อาจารย์แต่ละท่านสังกัดโปรแกรมวิชาได้เพียงหนึ่งโปรแกรมวิชาในแต่ละช่วงเวลา
- นักศึกษาแต่ละคนมีอาจารย์ที่ปรึกษาได้เพียงคนเดียวในแต่ละช่วงเวลา

จากเงื่อนไขที่กำหนดมาให้สามารถนำมาระบุ Temporal Fact Type ให้กับTemporal Entity Type จากขั้นตอนที่ 3 ได้ดังรูปที่ 5.3

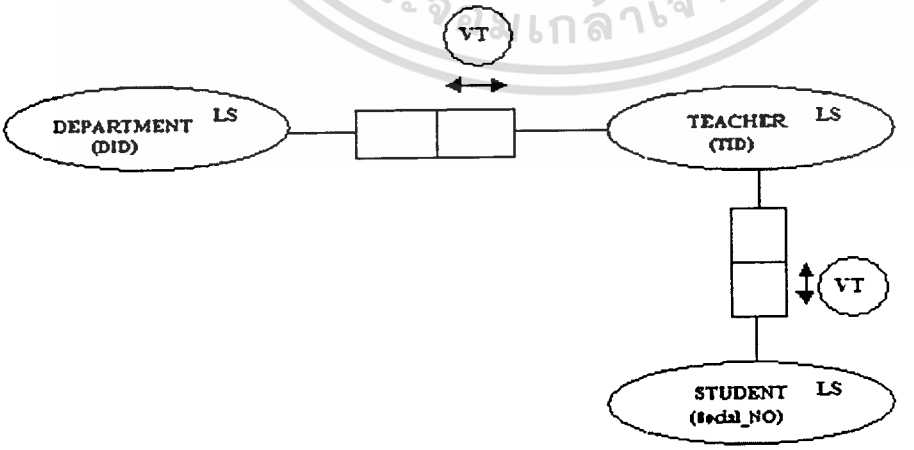
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 แสดง การระบุ Temporal Fact Type ให้กับ Temporal Entity Type จากขั้นตอนที่ 3 ตาม เงื่อนไขที่ระบุ

จากรูปที่ 5.3 อธิบายได้ว่า Temporal Entity Type DEPARTMENT ระบุ Temporal Fact Type กับ Temporal Entity Type TEACHER และระบุ Uniqueness Constraint (↔) บน Temporal Fact Type ด้าน Temporal Entity Type TEACHER ซึ่งมีความหมายว่า Temporal Entity Type DEPARTMENT มีความสัมพันธ์กับ Temporal Entity Type TEACHER แบบหนึ่งต่อกลุ่ม (One To Many) ซึ่งเช่นเดียวกับ Temporal Entity Type TEACHER มีความสัมพันธ์กับ Temporal Entity Type STUDENT แบบหนึ่งต่อกลุ่ม

ขั้นตอนที่ 5 พิจารณา Temporal Fact Type จากขั้นตอนที่ 4 ว่าเป็น Temporal Fact Type ชนิดใด ตามชนิดของTemporal Fact Type ที่อธิบายในหัวข้อ 3.2.1.2 จากขั้นตอนที่ 4 ระบุรูปแบบเวลาให้กับ Temporal Fact Type แต่ละชนิดตามรูปแบบที่สนใจ ซึ่งในระบบบริการข้อมูลนักศึกษาสำหรับอาจารย์ที่ปรึกษานี้เลือกใช้รูปแบบของ Valid Time ซึ่งแสดงได้ดังรูปที่ 5.4



รูปที่ 5.4 แสดง การระบุรูปแบบเวลาให้กับ Temporal Fact Type

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 5.4 แสดงให้เห็นโครงร่างหลักที่ได้จากการออกแบบโดยวิธีทีไนแอม

5.1.2 ขั้นตอนการเขียนโครงร่างย่อย (Sub Schema) ประกอบด้วย

ขั้นตอนที่ 1 พิจารณา Temporal Entity Type แต่ละตัวจากโครงร่างหลักว่า Temporal Entity Type แต่ละตัวนั้นประกอบด้วยรายละเอียดอะไรบ้าง เพื่อที่จะทำการระบุ Entity Type และหรือ Label Type ให้กับ Temporal Entity Type

จากโครงร่างหลักที่ได้จากรูปที่ 5.4 นำ Temporal Entity Type มาระบุรายละเอียดตามรายละเอียดที่ต้องการได้ดังนี้

- Temporal Entity Type DEPARTMENT รายละเอียดที่ต้องการระบุคือ Label Type ชื่อโปรแกรมวิชา (DNAME)

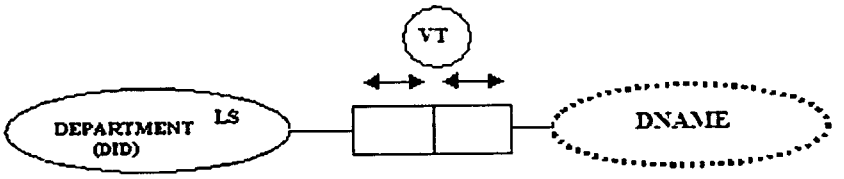
- Temporal Entity Type TEACHER รายละเอียดที่ต้องการระบุคือ Label Type ชื่ออาจารย์ (TNAME) และ นามสกุล (TSURNAME)

- Temporal Entity Type STUDENT รายละเอียดที่ต้องการระบุคือ Label Type รหัสนักศึกษา ชื่อนักศึกษา นามสกุล ชื่อบิดา ชื่อมารดา ชื่อผู้ปกครอง ภูมิลำเนา เพศ ห้องเรียน เกรดเฉลี่ย ชื่อทุนที่ได้รับ จำนวนทุนที่ได้รับ และ จำนวนเงินกู้ที่ได้รับ

ขั้นตอนที่ 2 ระบุ Temporal Fact Type ตามเงื่อนไขที่สนใจให้กับ Temporal Entity Type กับ Entity Type และหรือ Label Type ของ Temporal Entity Type แต่ละตัว สำหรับการระบุ Temporal Fact Type ให้กับ Entity Type และหรือ Label Type ของ Temporal Entity Type แต่ละตัวจะต้องพิจารณาด้วยว่า Entity Type และหรือ Label Type นั้นมีการเปลี่ยนแปลงค่าในแต่ละช่วงเวลาหรือไม่

- ระบุ Temporal Fact Type ให้กับ Temporal Entity Type DEPARTMENT กับ Label Type DNAME จะพบว่าเงื่อนไขความสัมพันธ์ระหว่าง Temporal Entity DEPARTMENT กับ Label Type DNAME มีความสัมพันธ์แบบ หนึ่งต่อหนึ่ง หมายความว่าโปรแกรมวิชามีชื่อโปรแกรมวิชาเพียงชื่อเดียว ณ. เวลาหนึ่ง แต่อาจจะเปลี่ยนแปลงชื่อโปรแกรมวิชาได้เมื่อเวลาเปลี่ยนไป ตัวอย่างเช่น โปรแกรมวิชาที่มีรหัสโปรแกรมเป็น "P1" ที่ Life Span เริ่มต้นที่ วันที่ 1 มกราคม 2535 จนถึงปัจจุบัน มีชื่อโปรแกรมวิชา "การศึกษาเคมี" ตั้งแต่ Valid Time เริ่มต้นที่ วันที่ 1 มกราคม 2535 จนถึง วันที่ 1 มกราคม 2540 และเปลี่ยนชื่อโปรแกรมวิชาใหม่เป็น " เคมี" ตั้งแต่ Valid Time เริ่มต้น ที่ วันที่ 1 มกราคม 2540 จนถึงปัจจุบัน เป็นต้น ดังนั้น การระบุ Temporal Fact Type ให้กับ Temporal Entity Type DEPARTMENT กับ Label Type DNAME โดยเลือกรูปแบบเวลาที่ระบุบน Temporal Fact Type เป็น Valid Time แสดงได้ดังรูปที่ 5.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 แสดง การระบุ Temporal Fact Type ให้กับ Temporal Entity Type DEPARTMENT กับ Label Type DNAME ที่ระบุรูปแบบเวลา Valid Time

- การระบุ Temporal Fact Type ให้กับ Temporal Entity Type TEACHER กับ Label Type TNAME และ Label Type TSURNAME จะพบว่าเงื่อนไขความสัมพันธ์ระหว่าง Temporal Entity TEACHER กับ Label Type TNAME มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง และ Temporal Entity Type TEACHER กับ Label Type TSURNAME มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง หมายความว่าอาจารย์แต่ละคนมีชื่อเพียงชื่อเดียว ณ. เวลาหนึ่ง และมีนามสกุลเพียงนามสกุลเดียว ในแต่ละช่วงเวลาเช่นเดียวกัน แต่อาจจะเปลี่ยนแปลงชื่อ และนามสกุลได้เมื่อเวลาเปลี่ยนไป ดังนั้น การระบุ Temporal Fact Type ให้กับ Temporal Entity Type TEACHER กับ Label Type TNAME และ Label Type TSURNAME โดยเลือกรูปแบบเวลาที่ระบุบน Temporal Fact Type เป็น Valid Time แสดงได้ดังรูปที่ 5.6

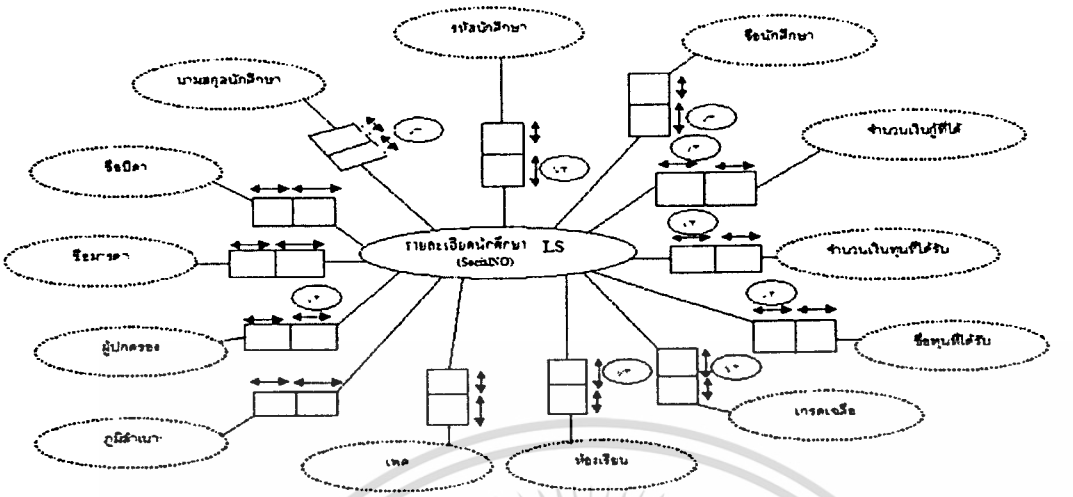
รูปที่ 5.6 แสดง การระบุ Temporal Fact Type ให้กับ Temporal Entity Type TEACHER กับ Label Type TNAME และ Label Type TSURNAME ที่ระบุรูปแบบเวลา Valid Time

- การระบุ Temporal Fact Type ให้กับ Temporal Entity Type STUDENT กับ Label Type รหัสนักศึกษา (SID) ชื่อนักศึกษา (SNAME) นามสกุล (SSURNAME) ชื่อบิดา (SFATHER) ชื่อมารดา (SMOTHER) ชื่อผู้ปกครอง(SRULER) ภูมิลำเนา (SDOMICILE) เพศ (SSEX) ห้องเรียน (SROOM) เกรดเฉลี่ย (SGRADE) ชื่อทุนที่ได้รับ (SNFOUNDATION) จำนวนทุนที่ได้รับ (SCFOUNDATION) และ จำนวนเงินกู้ที่ได้รับ (SCLOAN) จะพบว่าเงื่อนไขความสัมพันธ์ระหว่าง Temporal Entity STUDENT กับ Label Type รหัสนักศึกษา มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง ในแต่ละช่วงเวลา และมีโอกาสเปลี่ยนแปลงได้เมื่อเวลาเปลี่ยนไป ความสัมพันธ์ระหว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Temporal Entity STUDENT กับ Label Type ชื่อนักศึกษา มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง ในแต่ละช่วงเวลา และมีโอกาสเปลี่ยนแปลงได้เมื่อเวลาเปลี่ยนไป ความสัมพันธ์ระหว่าง Temporal Entity STUDENT กับ Label Type นามสกุล มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง ในแต่ละช่วงเวลา และมีโอกาสเปลี่ยนแปลงได้เมื่อเวลาเปลี่ยนไป ความสัมพันธ์ระหว่าง Temporal Entity STUDENT กับ Label Type ชื่อบิดา มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง และไม่มีโอกาสเปลี่ยนแปลงไปแต่ละช่วงเวลา ความสัมพันธ์ระหว่าง Temporal Entity STUDENT กับ Label Type ชื่อมารดา มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง และไม่มีโอกาสเปลี่ยนแปลงไปแต่ละช่วงเวลา ความสัมพันธ์ระหว่าง Temporal Entity STUDENT กับ Label Type ผู้ปกครอง มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง ในแต่ละช่วงเวลา และมีโอกาสเปลี่ยนแปลงได้เมื่อเวลาเปลี่ยนไป ความสัมพันธ์ระหว่าง Temporal Entity STUDENT กับ Label Type ชื่อภูมิลำเนา มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง และไม่มีโอกาสเปลี่ยนแปลงไปแต่ละช่วงเวลา ความสัมพันธ์ระหว่าง Temporal Entity STUDENT กับ Label Type เพศ มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง และไม่มีโอกาสเปลี่ยนแปลงไปแต่ละช่วงเวลา ความสัมพันธ์ระหว่าง Temporal Entity STUDENT กับ Label Type ห้องเรียน มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง ในแต่ละช่วงเวลา และมีโอกาสเปลี่ยนแปลงได้เมื่อเวลาเปลี่ยนไป ความสัมพันธ์ระหว่าง Temporal Entity STUDENT กับ Label Type เกรดเฉลี่ย มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง ในแต่ละช่วงเวลา และมีโอกาสเปลี่ยนแปลงได้เมื่อเวลาเปลี่ยนไป ความสัมพันธ์ระหว่าง Temporal Entity STUDENT กับ Label Type ชื่อทุนที่ได้รับ มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง ในแต่ละช่วงเวลา และมีโอกาสเปลี่ยนแปลงได้เมื่อเวลาเปลี่ยนไป ความสัมพันธ์ระหว่าง Temporal Entity STUDENT กับ Label Type จำนวนทุนที่ได้รับ มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง ในแต่ละช่วงเวลา และมีโอกาสเปลี่ยนแปลงได้เมื่อเวลาเปลี่ยนไป และความสัมพันธ์ระหว่าง Temporal Entity STUDENT กับ Label Type จำนวนเงินกู้ที่ได้รับ มีความสัมพันธ์แบบหนึ่งต่อหนึ่ง ในแต่ละช่วงเวลา และมีโอกาสเปลี่ยนแปลงได้เมื่อเวลาเปลี่ยนไป การระบุ Temporal Fact Type ให้กับ Temporal Entity Type STUDENT กับ Label Type แต่ละตัวที่ได้กล่าวมา ซึ่งจะพบว่ามีทั้ง Label Type ที่มีค่าเปลี่ยนแปลงไปตามเวลา และ Label Type ที่มีค่าคงที่ไม่มีการเปลี่ยนแปลงไปตามเวลา โดยเลือกรูปแบบเวลาที่ระบุบน Temporal Fact Type เป็น Valid Time แสดงได้ดังรูปที่ 5.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.7 แสดง การระบุ Temporal Fact Type ให้กับ Temporal Entity Type STUDENT กับ Label Type ทั้งหมด

5.2 ขั้นตอนการแปลงทีไนแอม เป็นฐานข้อมูลเชิงสัมพันธ์

สำหรับขั้นตอนการแปลงทีไนแอม เป็นฐานข้อมูลเชิงสัมพันธ์ จะพิจารณาไปพร้อม ๆ กัน ทั้งโครงร่างหลัก และโครงร่างย่อย โดยจะพิจารณาแยกเป็นกรณีต่าง ๆ ตามชนิดของ Temporal Fact Type และในระบบบริการข้อมูลนักศึกษาสำหรับอาจารย์ที่ปรึกษา ที่ใช้เป็นตัวอย่างประกอบในการนำเสนอนั้น อาจจะไม่ครอบคลุมทุกกรณีดังที่ได้อธิบายในหัวข้อ 3.4 เหตุผลเพราะว่าบางกรณีที่นำเสนอไม่ได้ครอบคลุมทุกกรณีของขั้นตอนการแปลงทีไนแอม เป็นฐานข้อมูลเชิงสัมพันธ์

5.2.1 กรณี Temporal Fact Type มี Life Span ระบุอยู่ สำหรับตัวอย่างที่นำเสนอนี้จะพบว่ากรณีนี้ไม่มีทั้งในโครงร่างหลัก และโครงร่างย่อย ของทีไนแอม

5.2.2 กรณี Temporal Fact Type มี Valid Time ระบุอยู่ ซึ่งประกอบด้วยกรณีย่อยดังต่อไปนี้

5.2.2.1 กรณี Temporal Fact Type มีความสัมพันธ์แบบ หนึ่ง ต่อ หนึ่ง ประกอบด้วย

- กรณี Temporal Entity Type มีความสัมพันธ์กับ Entity Type และหรือ Label Type เมื่อพิจารณาจากโครงร่างหลัก และโครงร่างย่อย ของทีไนแอม มีรูปแบบในกรณีนี้ ในโครงร่างย่อยที่แสดงในรูปที่ 5.5 รูปที่ 5.6 และ รูปที่ 5.7 ซึ่งสามารถแปลงเป็นฐานข้อมูลเชิงสัมพันธ์ ได้ดังนี้ รูปที่ 5.8 ซึ่งแปลงมาจากทีไนแอมรูปที่ 5.5 รูปที่ 5.9 ซึ่งแปลงมาจากทีไนแอมรูปที่ 5.6 และ รูปที่ 6.10 ซึ่งแปลงมาจากทีไนแอมรูปที่ 5.7 ตามกระบวนการที่นำเสนอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในบทที่ 3 หัวข้อที่ 3.4.2.1 ในกรณี Temporal Fact Type มีความสัมพันธ์กับ Entity Type และหรือ Label Type

DEPARTMENT

DID	LSs	Lse

DEPARTMENT_NAME

DID	DNAME	VTs	VTe

รูปที่ 5.8 แสดง 2 รีเลชันที่ได้จากการแปลง TNIAM จากรูปที่ 5.5

TEACHER

TID	LSs	Lse

TEACHER_NAME

TID	TNAME	VTs	VTe

TEACHER_SURNAME

TID	TSURNAME	VTs	VTe

รูปที่ 5.9 แสดง 3 รีเลชันที่ได้จากการแปลง TNIAM จากรูปที่ 5.6

STUDENT

STUID	SFATHER	SMOTHER	SSEX	SDOMICILE	LSs	LSe

STUDENT_ID

STUID	SID	VTS	Vte

STUDENT_NAME

STUID	SNAME	VTS	VTe

STUDENT_SURNAME

STUID	SSURNAME	VTS	VTe

STUDENT_RULER

STUID	SRULER	VTS	VTe

STUDENT_ROOM

STUID	SROOM	VTS	VTe

STUDENT_GRADE

STUID	SGRADE	VTS	VTe

STUDENT_NFOUNDATION

STUID	SNFOUNDATION	VTS	VTe

STUDENT_CFOUNDATION

STUID	SCFOUNDATION	VTS	VTe

STUDENT_LOAN

STUID	SLOAN	VTS	VTe

รูปที่ 5.10 แสดง 10 รีเลชันที่ได้จากการแปลง TNIAM จากรูปที่ 5.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2.2 กรณี Temporal Fact Type มีความสัมพันธ์แบบ หนึ่ง ต่อ กลุ่ม

- กรณี Temporal Entity Type มีความสัมพันธ์ กับ Entity Type และ Label Type สำหรับตัวอย่างที่นำเสนอนี้จะพบว่ากรณีนี้ไม่มีทั้งในโครงร่างหลัก และโครงร่างย่อย ของทีไนแอม

- กรณี Temporal Entity Type มีความสัมพันธ์ กับ Temporal Entity Type ในกรณีนี้จะพบว่ามีอยู่ในโครงร่างหลักของทีไนแอม แต่จากกระบวนการที่อธิบายในบทที่ 3 ในหัวข้อการแปลงทีไนแอมเป็นฐานข้อมูลเชิงสัมพันธ์ ในกรณี Temporal Fact Type มีความสัมพันธ์แบบหนึ่งต่อกลุ่ม ในประเภทของกรณีTemporal Entity Type มีความสัมพันธ์กับTemporal Entity Type ก็จะพบว่า Temporal Entity Type ที่ปรากฏในโครงร่างหลักนั้นเมื่อทำการแปลงเป็นฐานข้อมูลเชิงสัมพันธ์ จะได้รีเลชันที่ DEPARTMENT รีเลชัน TEACHER และ รีเลชัน STUDENT ซึ่งจะพบว่าซ้ำกับรีเลชันที่ได้จากการแปลงในหัวข้อ 5.2.2.1 ในกรณี Temporal Entity Type มีความสัมพันธ์กับ Entity Type และหรือ Label Type ดังนั้นถ้าเจอกรณีแบบนี้ก็ให้ทำการตัดรีเลชันที่ซ้ำนั้นทิ้ง

5.2.2.3 กรณี Temporal Fact Type มีความสัมพันธ์แบบ กลุ่ม ต่อ กลุ่ม สำหรับตัวอย่างที่นำเสนอนี้จะพบว่ากรณีนี้ไม่มีทั้งในโครงร่างหลัก และโครงร่างย่อย ของทีไนแอม ในแต่ละกรณีย่อย

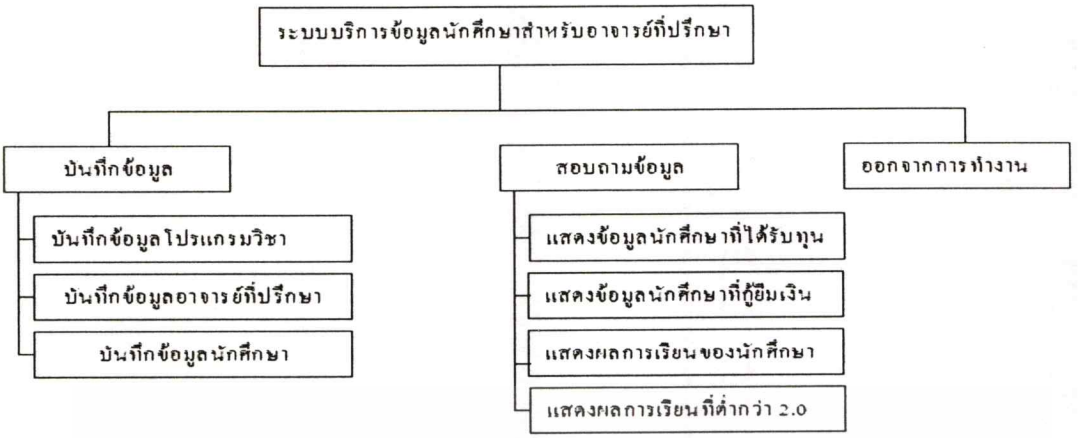
5.3 แสดงหน้าจอต่างๆ ของระบบการบริการข้อมูลนักศึกษา สำหรับอาจารย์ที่ปรึกษา

แสดงหน้าจอการทำงานของระบบการบริการข้อมูลนักศึกษาสำหรับอาจารย์ที่ปรึกษา บนฐานข้อมูลเชิงเวลา ที่ได้จากการออกแบบด้วยกระบวนการทีไนแอม และแปลงจากทีไนแอมเป็นฐานข้อมูลเชิงสัมพันธ์

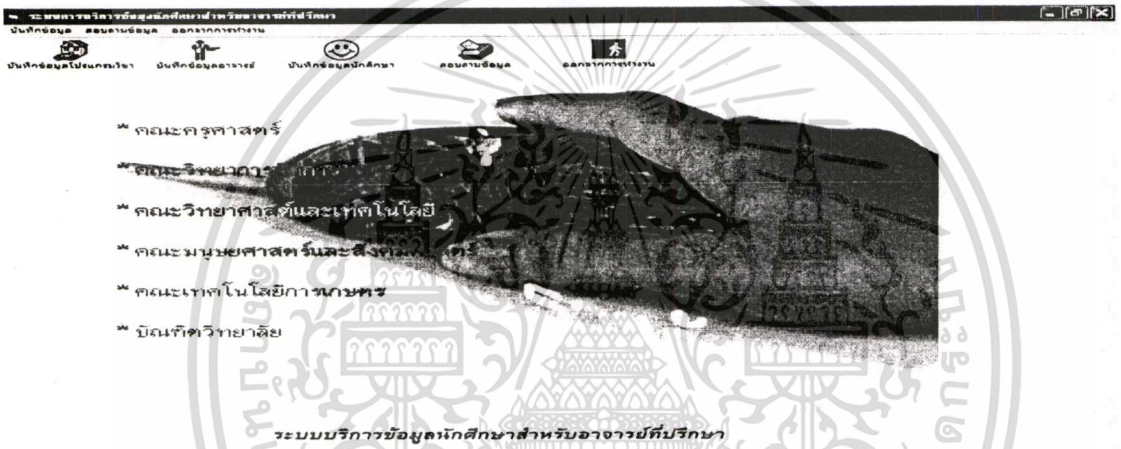
5.3.1 โครงสร้างโปรแกรมระบบการบริการข้อมูลนักศึกษาสำหรับอาจารย์ที่ปรึกษา

โครงสร้างการทำงานของระบบบริการข้อมูลนักศึกษา มีรายละเอียดดังแสดงในรูปที่ 5.11

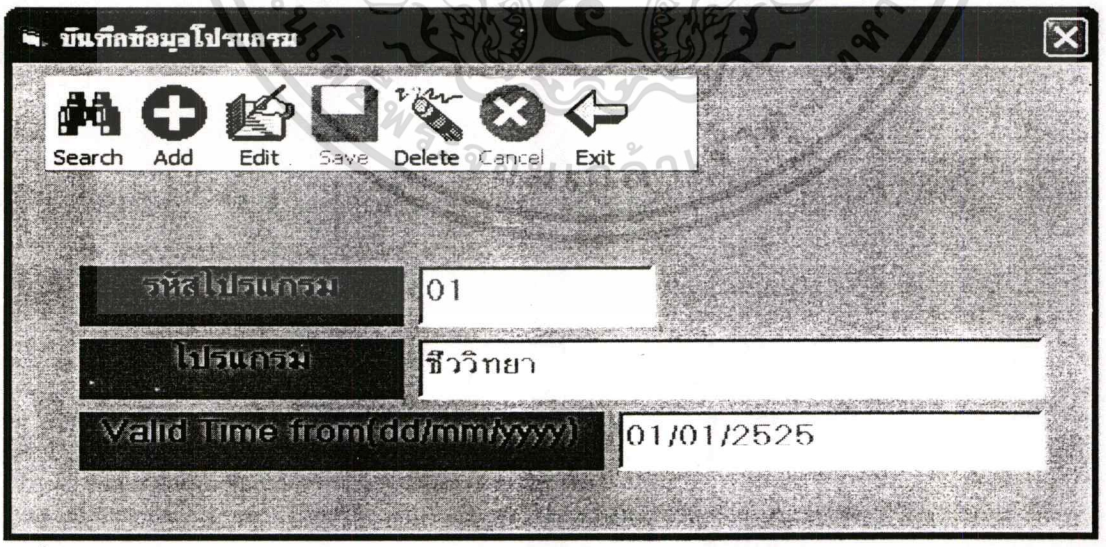
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.11 แสดงโครงสร้างของระบบงานบริการข้อมูลนักศึกษาสำหรับอาจารย์ที่ปรึกษา



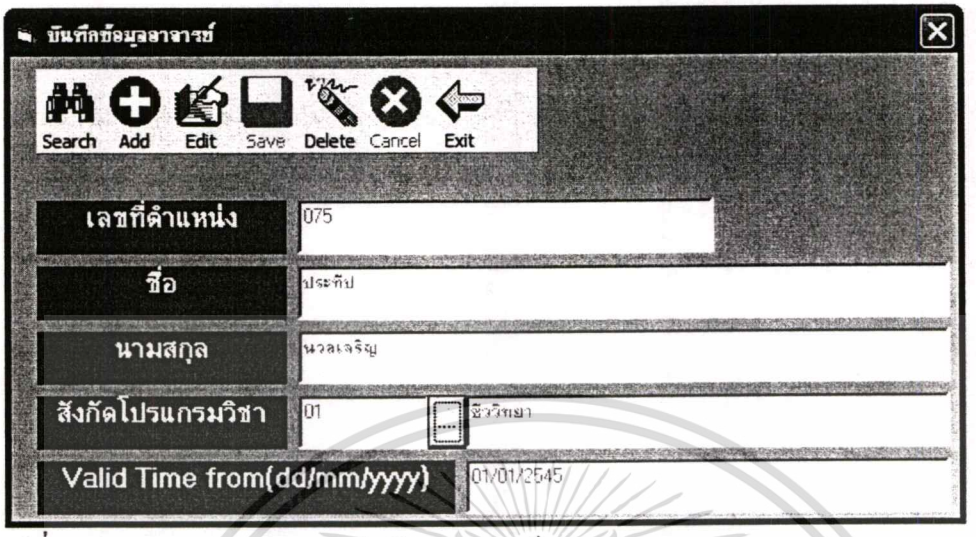
รูปที่ 5.12 แสดงเมนูหลัก ซึ่งผู้ใช้สามารถเลือกใช้งานระบบงานได้อย่างอิสระ



รูปที่ 5.13 แสดงจอภาพ การบันทึกข้อมูลโปรแกรมวิชา

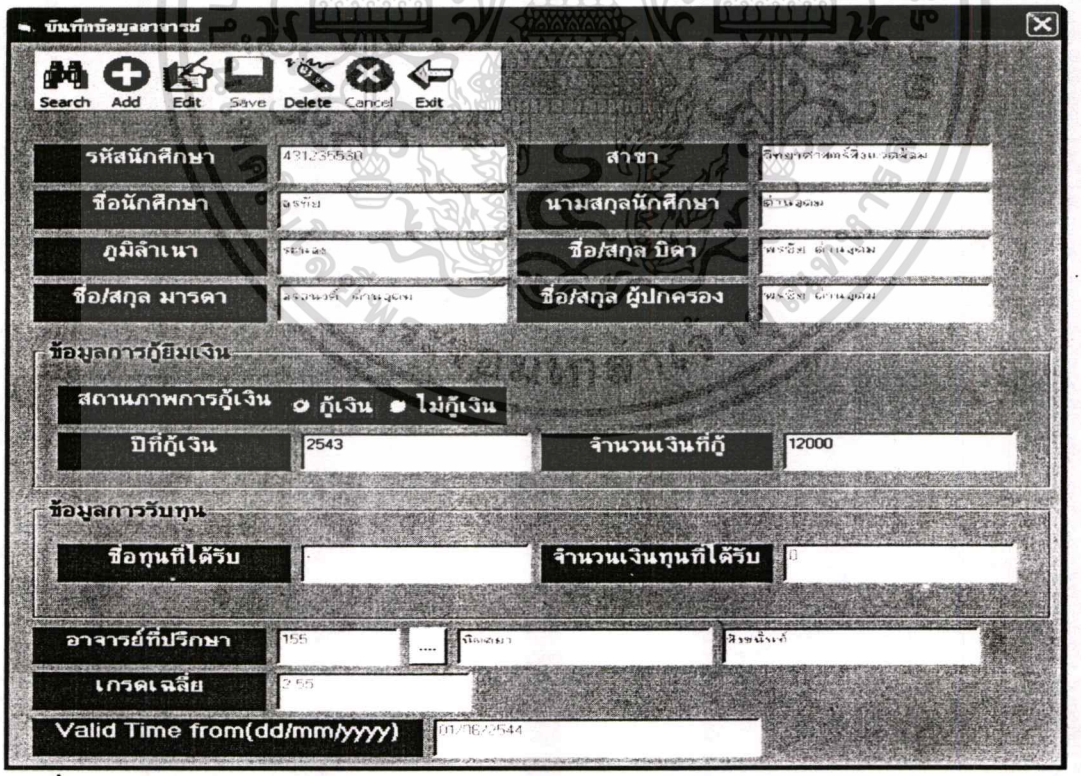
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 5.13 เป็นการแสดงจอภาพการบันทึกข้อมูลของโปรแกรมวิชา ซึ่งผู้ใช้งาน สามารถ ค้นหาข้อมูล บันทึกข้อมูล แสดงข้อมูล หรือ ออกจากการทำงานได้ ตามความเป็นจริง



รูปที่ 5.14 แสดงจอภาพ การบันทึกข้อมูลอาจารย์

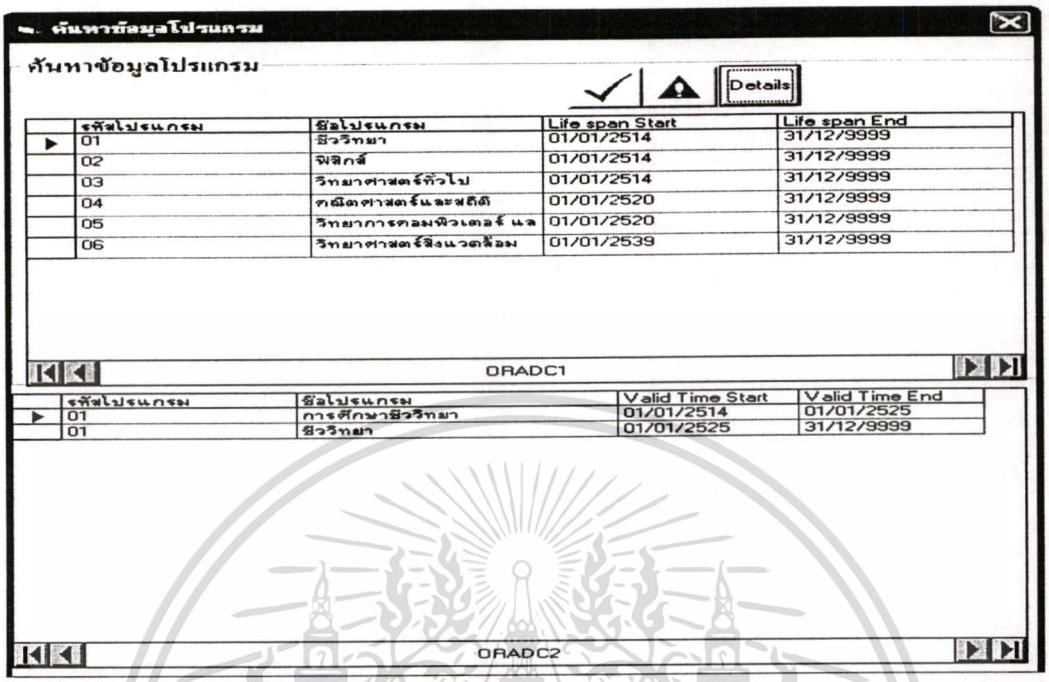
จากรูปที่ 5.14 เป็นการแสดงจอภาพการบันทึกข้อมูลของอาจารย์ที่ปรึกษา ซึ่งผู้ใช้งาน สามารถ ค้นหาข้อมูล บันทึกข้อมูล แสดงข้อมูล หรือ ออกจากการทำงานได้ ตามความเป็นจริง



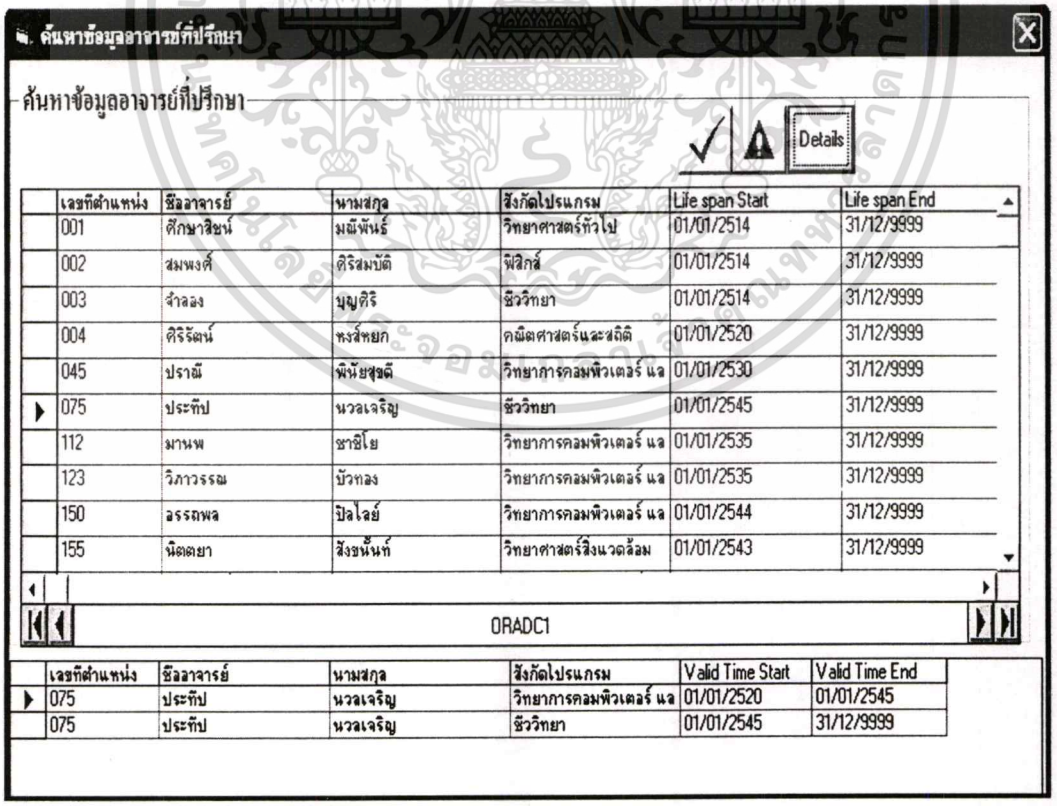
รูปที่ 5.15 หน้าจอการบันทึกข้อมูลนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 5.15 แสดงจอภาพการบันทึกข้อมูลของนักศึกษา ซึ่งผู้ใช้งาน สามารถ ค้นหาข้อมูล บันทึกข้อมูล แสดงข้อมูล หรือ ออกจากการทำงานได้ ตามความเป็นจริง

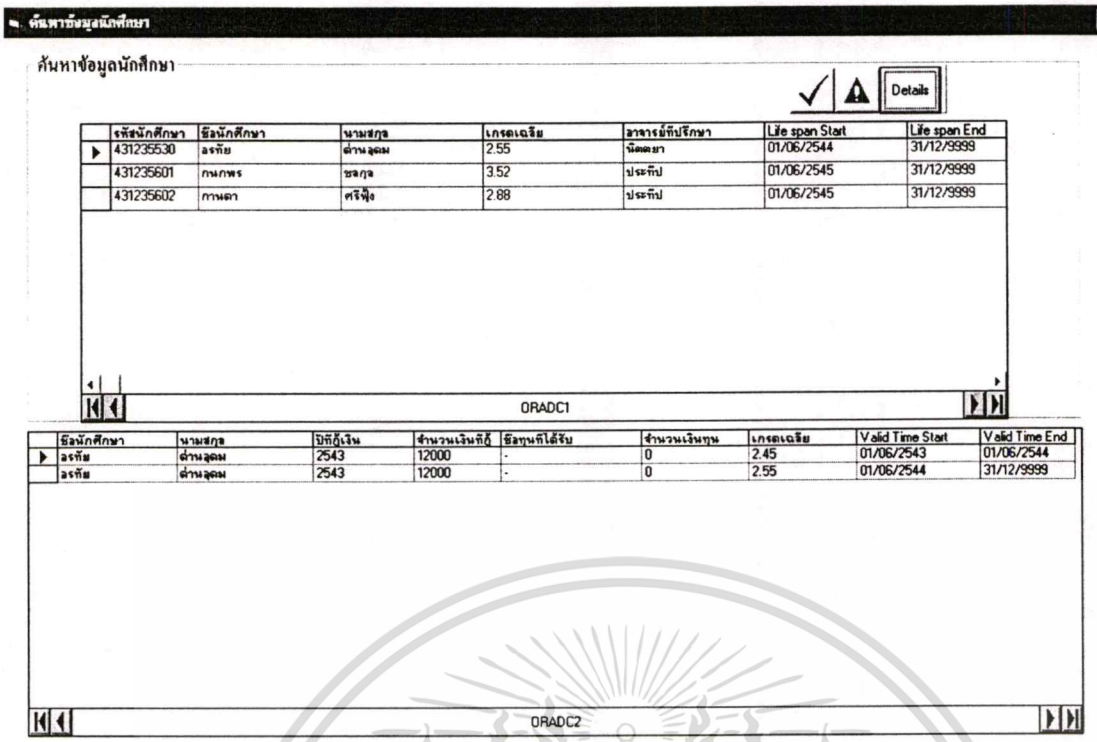


รูปที่ 5.16 หน้าจอการแสดงการค้นหาข้อมูลโปรแกรม

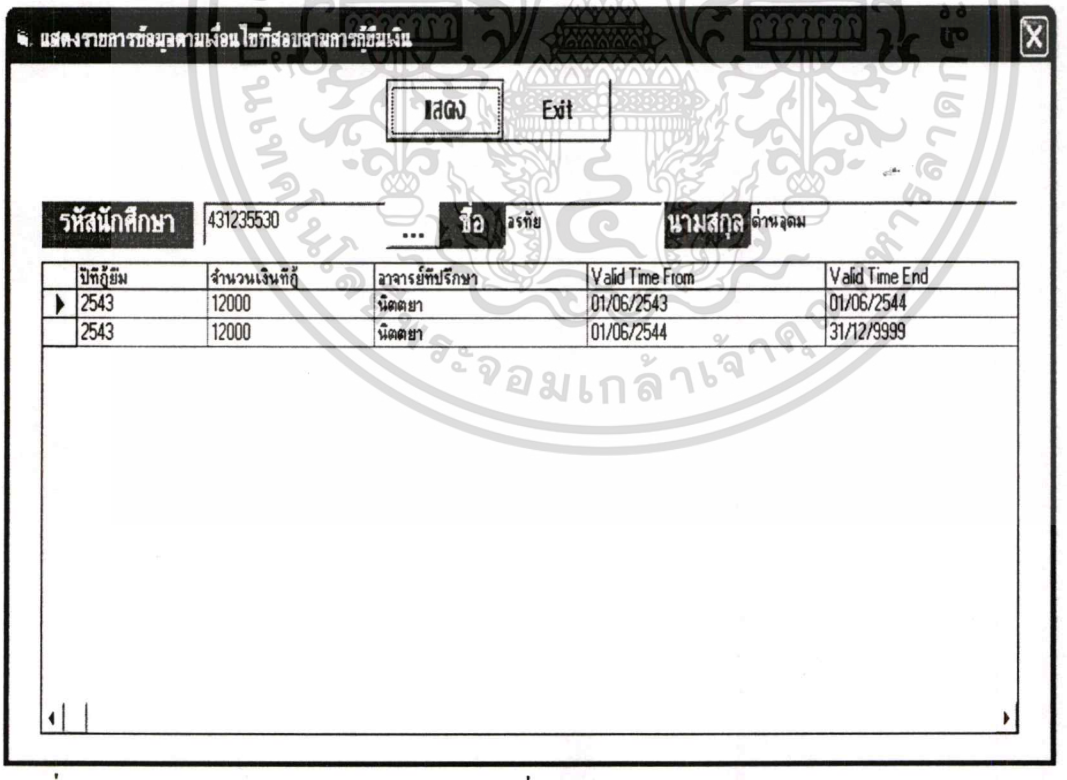


รูปที่ 5.17 หน้าจอแสดงข้อมูลการค้นหาข้อมูลอาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

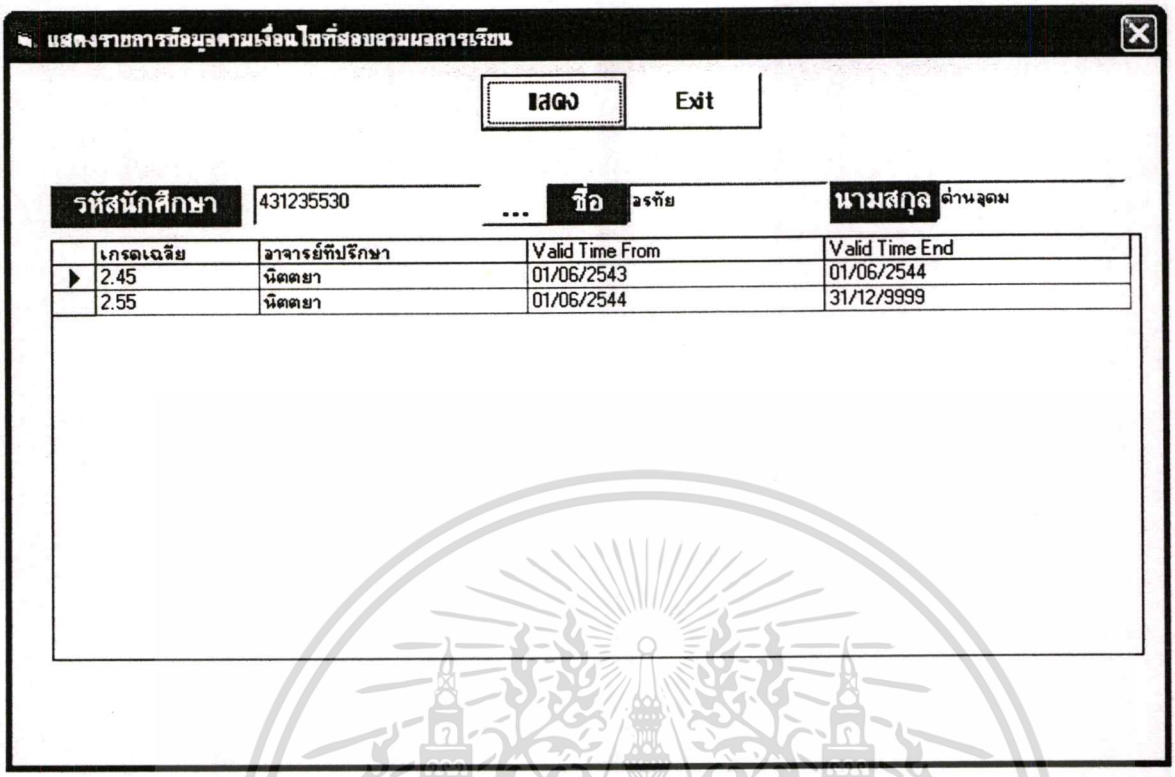


รูปที่ 5.18 แสดงหน้าจอการค้นหาข้อมูลนักศึกษา



รูปที่ 5.19 หน้าจอแสดงรายละเอียด นักศึกษาที่กู้ยืมเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.20 หน้าจอแสดงผลการเรียนของนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปผลการวิจัย และข้อเสนอแนะ

6.1 สรุปผลการวิจัย

ไนแอม (NIAM) เป็นแบบจำลองข้อมูลระดับแนวความคิด ที่มีพื้นฐานมาจากวิธี Binary Relationship โดยที่ มีแนวความคิดเกี่ยวกับโครงสร้างว่าประกอบด้วย Entity Type , Label Type ,Subtype , Elementary Fact Type และ Constraint และมีการแทน (Represent) ด้วยรูปภาพ ซึ่งง่ายต่อการอ่าน และเข้าใจในการออกแบบ

ในงานวิจัยนี้ ผู้วิจัยได้นำเสนอโมเดลทีไนแอม (TNIAM) ซึ่งเป็นโมเดลสำหรับการออกแบบฐานข้อมูลเชิงเวลา โดยโมเดลทีไนแอมนี้ได้พัฒนามาจากไนแอม ซึ่งได้เพิ่มสัญลักษณ์บางอย่างและเวลาเข้าไปในโครงสร้างของไนแอม เพื่อให้สามารถรองรับการออกแบบฐานข้อมูลเชิงเวลาได้

นอกจากการนำเสนอโมเดลทีไนแอมแล้ว งานวิจัยนี้ ได้นำเสนอขั้นตอน และกระบวนการ ในการแปลงโมเดลทีไนแอม เป็นฐานข้อมูลเชิงสัมพันธ์ในรูปแบบต่าง ๆ ที่อยู่ในรูปของ 5NF ได้โดยตรง และรวมทั้งงานวิจัยนี้ ได้นำทีไนแอม ไปประยุกต์ใช้กับระบบงานจริง และพัฒนาเป็นแอฟลิเคชัน

6.2 ข้อเสนอแนะ

ในส่วนของวิทยานิพนธ์ฉบับนี้ ที่ผู้วิจัยนำเสนอวิธีการออกแบบฐานข้อมูลเชิงเวลาด้วยวิธีการทีไนแอม และ กระบวนการแปลงทีไนแอม เป็นฐานข้อมูลเชิงสัมพันธ์ ฐานข้อมูลเชิงสัมพันธ์ที่อยู่ในรูปของ Temporal Database และ ฐานข้อมูลเชิงสัมพันธ์ที่อยู่ในรูปของ Object Relational Database นั้น รวมทั้งการนำทีไนแอมไปประยุกต์ใช้กับระบบงานจริง ดังที่ได้นำเสนอไปแล้วนั้น สำหรับกรณีศึกษาที่ผู้วิจัยนำเสนอยังคงเน้นในรูปของเวลา Lifespan และ Valid Time ซึ่งยังมีรูปแบบของเวลาที่ผู้วิจัยยังไม่ได้นำเสนอ และเป็นรูปแบบเวลาที่น่าสนใจได้แก่ Transaction time และ Bitemporal ซึ่งถ้าผู้วิจัยท่านใดสนใจหัวข้อนี้ ก็นับว่าเป็นเรื่องน่าสนใจ และได้รับการค้นคว้าวิจัยต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] Abdullah Uz Tanssel , James Clifford , Shashi Gadia , Sushi Jajodia , Arie Segev, Richard Snodgrass.**Temporal Database Theory Design and Implement**. California ,The benjamin /cumming publishing , Inc., 1993.

[2] Christian S.Jensen, Richard T. Snodgrass, and Michale D. Soo. "Extending Existing Dependency Theory to Temporal Database". **IEEE Transaction on Knowledge and Data Engineering.** Vol. 8, No. 4, 1996, pp. 563-577.

[3] H Gregersen and J. S. Jensen. "Temporal Entity-Relationship models - a survey". **IEEE Transactions on Knowledge and Data Engineering**. 11(3):464--497, 1999.

[4] Heidi Gregersen , Leo Mark and Christian S. Jensen " Mapping Temporal ER Diagrams to Relational Schemas." **A TIME CENTER TECHNIAL REPORT**. December 4, 1998.

[5] Heidi Gregersen , Christian S. Jensen "Conceptual Modeling of Time-Varying Information." A TIME CENTER TECHNIAL REPORT. December 4, 1998.

[6] Nijssen G.M., Halpin, T.A. **Conceptual Schema and Relational Database Design**. New York : Prentice Hall , 1989.

[7] Suphamit Chittayasothorn. 2001. "Database Design and SQL". **Seminar in Database Theory and Design.** Bongkok, July 2001.

[8] Suphamit Chittayasothorn. 2001. "Database Concepts and Models".**Cache Object Database Technology Conference 2001.** Bangkok. June 2001.

[9] Time Center. " **Temporal Database**"[Online]. Available : http://www.cs.auc.dk/TimeCenter/pub.htm.

[10] X.Sean Wang "Logical Design for Temporal Database". **ACM Transaction on Database System.** Vol. 22, June 1997, pp. 155-170.

[11] บัณฑิต พัสยา. "แบบจำลองฐานข้อมูลเชิงแนวคิดทีโอโอไนแอม และการแปลงรูปเป็นฐานข้อมูลเชิงวัตถุ" วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2544

[12] พันตรีหญิงรัชนีพร ผดุงผล. "เครื่องมือออกแบบฐานข้อมูลเชิงวัตถุ โดยใช้ NIAM++" วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2533 [9]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[13] สุธีรา พันธ์ธีรานุรักษ์. "แบบจำลองฐานข้อมูลเชิงแนวคิดโอโอไนแอมและการแปลงรูปเป็นฐานข้อมูลเชิงวัตถุ" วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ ทม 0407.01/๖๗ ๔๗

คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตบางเขน
ตู้ ป.ณ. 1032 ไปรษณีย์เกษตรศาสตร์
กรุงเทพมหานคร 10903

๒๖ สิงหาคม 2545

เรื่อง ตอบรับบทความทางวิชาการ

เรียน นางสาวผุสดี พรผล

ตามที่ท่านได้ส่งบทความเรื่อง "วิธีการออกแบบฐานข้อมูลเชิงเวลา" มาเพื่อลงตีพิมพ์ในวารสาร "วิศวกรรมสาร มก." ดังที่ทราบแล้วนั้น

ฝ่ายบรรณาธิการวารสาร "วิศวกรรมสาร มก." ได้รับพิจารณาบทความของท่านแล้วเห็นสมควรรับตีพิมพ์ บทความของท่าน โดยจะนำลงพิมพ์ในวารสาร"วิศวกรรมสาร มก." ฉบับที่ 48 ประจำเดือน ธันวาคม 2545 – มีนาคม 2546 ทั้งนี้เมื่อวารสารแล้วเสร็จ คณะฯ จะจัดส่งวารสารฉบับดังกล่าวให้ท่านต่อไป

จึงเรียนมาเพื่อโปรดทราบ

ขอแสดงความนับถือ

(นางสาวชุติมา เทพเฉลิม)
หัวหน้างานบริการวิชาการและวิจัย
ผู้ช่วยบรรณาธิการวารสารวิศวกรรมสาร มก.

คณะวิศวกรรมศาสตร์
โทร. 0-2942-8555 ต่อ 1126
โทรสาร 0-2579-2775

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

นางสาวผุสดี พรผล เกิดเมื่อวันที่ 18 มีนาคม 2512 ที่จังหวัดพังงา สำเร็จการศึกษาวิทยาศาสตรบัณฑิต สาขา ศาสตร์คอมพิวเตอร์ จากมหาวิทยาลัยรามคำแหง ปีการศึกษา 2538

ปี 2534- 2538 ทำงานตำแหน่งโปรแกรมเมอร์ บริษัท Synchron Cooperation จำกัด

ปี 2538-2540 ทำงานตำแหน่ง System Analyst บริษัท Solution21 จำกัด

ปี 2540 เข้ารับราชการ ตำแหน่ง อาจารย์ 1 ระดับ 3 สังกัดโปรแกรมวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ และ เทคโนโลยี สถาบันราชภัฏภูเก็ต

ปี 2542 ได้รับอนุมัติให้ลาศึกษาต่อในระดับปริญญาโท สาขาเทคโนโลยีสารสนเทศ ณ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง โดยทุนพัฒนาอาจารย์จากโครงการพัฒนาด้านวิทยาศาสตร์ และวิทยาศาสตร์ประยุกต์ (พ.ว.ศ.) จากสำนักงานสภาสถาบันราชภัฏ สาขาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ระดับปริญญาโท ปัจจุบันเป็นข้าราชการอาจารย์ 1 ระดับ 5 สังกัดโปรแกรมวิชาวิทยาการคอมพิวเตอร์ และ คอมพิวเตอร์ศึกษา คณะวิทยาศาสตร์และเทคโนโลยี สถาบันราชภัฏภูเก็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้