การพัฒนาระบบจัดการฐานข้อมูลเชิงสัมพันธ์แบบคิวบีอีอิงเวลา

A TEMPORAL QBE EXTENSION TO A RELATIONAL DATABASE MANAGEMENT SYSTEM

ธนัท อร่ามรัศมีวาณิชย์

TANAT ARAMRUSMEVANICH

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2543

ISBN 974-622-975-1

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาระบบจัดการฐานข้อมูลเชิงสัมพันธ์แบบคิวบีอีอิงเวลา

A TEMPORAL QBE EXTENSION TO A RELATIONAL DATABASE MANAGEMENT SYSTEM

T038529

ธนัท อร่ามรัตมีวาณิชย์

TANAT ARAMRUSMEVANICH

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน 38529
วัน, เดือน, ปี 5 ส.ค. 2544

พ.ศ. 2543

ISBN 974-622-976-1

สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A TEMPORAL QBE EXTENSION TO A RELATIONAL DATABASE MANAGEMENT SYSTEM

TANAT ARAMRUSMEVANICH

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING

SCHOOL OF GRADUATE STUDIES

KINGMONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2000

ISBN 974-622-976-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2000

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การพัฒนาระบบจัดการฐานข้อมูล เชิงสัมพันธ์แบบคิวบีอีอิงเวลา

A TEMPORAL QBE EXTENSION TO A RELATIONAL DATABASE MANAGEMENT SYSTEM

ชื่อนักศึกษา นายธนัท อุร่ามรัศมีวาณิชย์

รหัสประจำตัว 36061204

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมไฟฟ้า

อาจารย์ผู้ควบคุมวิทยานิพนธ์ รศ.ดร.ศุภมิตร จิตตะยโศธร

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ดร.วิศิษฏ์	หิรัญกิตติ	
ผศ.ดร.เอื้อน	ปิ่นเงิน	
ผศ.ดร.บุญธีร์	เครือตราชู	
รศ.ดร.บุญวัฒน์	อัตชู	
รศ.ดร.ศุภมิตร	จิตตะยโศธร	

วัน/เดือน/ปี ที่สอบ 26 ตุลาคม 2543 เวลา 11.00-12.00 น.

สถานที่สอบ ณ อาคาร 12 ชั้น ชั้น 4 (ห้อง E12-404)

บัณฑิตวิทยาลัยรับรองแล้ว

(รศ.ดร.บุญวัฒน์ อัตชู)

รักษาราชการแทนคณบดีบัณฑิตวิทยาลัย

วันที่ 17 **เดือน** พฤศจิกายน **พ.ศ.** 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การพัฒนาระบบจัดการฐานข้อมูลเชิงสัมพันธ์แบบคิวบีอีอิงเวลา
นักศึกษา	นายธนัท อร่ามรัศมีวาณิชย์
รหัสประจำตัว	36061204
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2543
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร.ศุภมิตร จิตตะยโศธร

บทคัดย่อ

ระบบจัดการฐานข้อมูลในปัจจุบันได้รับความนิยมเป็นอย่างมาก เนื่องจากมีความถูกต้องของข้อมูลสูง มีการจัดการเกี่ยวกับการควบคุมสถานะของข้อมูลในขณะที่มีผู้ใช้งานหลายๆคนพร้อมกัน มีการกู้ข้อมูลคืนให้อยู่ในสถานะที่ถูกต้องเมื่อเกิดความเสียหายกับระบบ มีการจัดการด้านความปลอดภัยของข้อมูลในระบบฐานข้อมูล มีส่วนติดต่อกับผู้ใช้คือภาษาสอบถามข้อมูลเชิงโครงสร้างที่มีประสิทธิภาพ และเป็นกันเองกับผู้ใช้ มีเครื่องมือที่ใช้อำนวยความสะดวกในการพัฒนา และใช้งานมากมาย

แต่อย่างไรก็ตามรูปแบบฐานข้อมูลเชิงสัมพันธ์ ซึ่งถูกนำเสนอโดย Codd [1] มีโครงสร้างข้อมูลเป็นตาราง 2 มิติคือ แถว (tuple) และ คอลัมน์ (attribute) ซึ่งฐานข้อมูลดังกล่าวถูกเรียกว่าฐานข้อมูลแบบสแนปช็อต(snapshot) ซึ่งไม่สามารถกำหนดมิติทางด้านเวลาให้กับข้อมูลได้ การเปลี่ยนแปลงข้อมูลที่เกี่ยวเนื่องกับเวลา ถูกกระทำเหมือนกับค่าปกติค่าหนึ่งเท่านั้น ทำให้ต้องเสียค่าใช้จ่ายมาก เนื่องจากจะต้องมีคอลัมน์ที่ใช้สำหรับเก็บข้อมูลเกี่ยวกับเวลา ข้อมูลที่จัดเก็บจึงมีความซ้ำซ้อนมาก และภาษาที่ใช้สอบถามข้อมูลจะซับซ้อนและยุ่งยากมาก

วิทยานิพนธ์ฉบับนี้ได้นำเสนอระบบจัดการฐานข้อมูลเชิงสัมพันธ์แบบอิงเวลา โดยการออกแบบและสร้างระบบ TQBE ซึ่งเป็นเครื่องมือที่ช่วยในการจัดการฐานข้อมูลเชิงเวลา โดยไม่จำเป็นต้องแก้ไขระบบฐานข้อมูลที่มีอยู่เดิมแต่ประการใด สถาปัตยกรรมของระบบนี้สามารถประยุกต์ใช้กับระบบจัดการฐานข้อมูลเชิงสัมพันธ์ใดๆก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	A Temporal QBE Extension to a Relational Database Management System
Student	Mr. Tanat Aramrusmevanich
Student ID.	36061204
Degree	Master of Engineering
Programme	Electrical Engineering
Year	2000
Thesis Advisor	Assoc. Prof. Dr. Suphamit Chittayasothorn

ABSTRACT

Nowadays, a database management system is more capable than in the past due to many facility, i.e. the use of integrity of data, concurrency control of many users at a time, recovery of damaged data, data security management, an efficiency of structured query language and many application tools.

However, in existing relational databases, a relation can be visualized as a two dimensional structure of tuples and attributes. This is called 'a snapshot relation' since it captures a snapshot of reality. Time can be considered as its third dimension, turning a relation into a three-dimensional structure like a data value, this will making.

In this thesis we propose a temporal relational database management system by designing and creating the TQBE (TQBE : Temporal Query-By-Example) tools for manipulating the temporal databases and not modifying the database management system. This architecture of this system can be apply to any of relational database management system.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดี ด้วยคำแนะนำและคำปรึกษา รวมทั้งได้รับการตรวจสอบจาก รศ.ดร.ศุภมิตร จิตตะยโศธร ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ผู้ทำวิจัยรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านอาจารย์ และขอกราบขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณคุณพ่อและคุณแม่ ผู้ให้กำเนิดข้าพเจ้า ผู้ให้ความเอื้ออาทรทางการศึกษา ให้การอบรมด้วยดีตลอดมา

ขอขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ทุกท่าน ที่ได้ช่วยแก้ไขวิทยานิพนธ์ฉบับนี้ให้สมบูรณ์มากยิ่งขึ้น

ขอขอบคุณรุ่นพี่ รุ่นน้อง และเพื่อนๆทุกคนที่ช่วยให้กำลังใจ พร้อมทั้งมอบคำแนะนำและคำติชมจนสำเร็จสมบูรณ์ได้ด้วยดี

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

ธนัท อร่ามรัศมีวาณิชย์

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 ความเป็นมาและความสำคัญของปัญหา 1

1.2 ความมุ่งหมายและวัตถุประสงค์ 2

1.3 ทฤษฎีและแนวความคิดที่ใช้ในการวิจัย 2

1.4 ขอบเขตของการวิจัย 2

1.5 ส่วนประกอบของวิทยานิพนธ์ 2

บทที่ 2 หลักการและความหมายของเวลา 3

2.1 ความหมายของเวลา 3

2.1.1 User-defined Time 3

2.1.2 Valid Time 3

2.1.3 Transaction Time 4

2.2 โมเดลของเวลา 4

2.2.1 ประเภทโมเดลของเวลา 5

2.2.2 หน่วยย่อยของเวลา 5

2.2.3 เวลาและเหตุการณ์ 5

2.2.4 ช่วงเวลาและจุดเวลา 5

2.2.5 ลักษณะการเปรียบเทียบสำหรับช่วงเวลา 6

2.3 การระบุเวลากับข้อมูล 6

2.3.1 ความหมายของการระบุเวลา 7

2.3.1.1 การระบุเวลาที่กระทำกับทับเปิล 7

2.3.1.2 การระบุเวลาที่กระทำกับแอททริบิวต์ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.3.2 วิธีการระบุเวลา 9
2.3.2.1 การระบุเวลาด้วยจุดของเวลา 9
2.3.2.2 การระบุเวลาด้วยช่วงเวลา 10
2.3.2.3 การระบุเวลาด้วยส่วนประกอบเชิงเวลา 11

บทที่ 3 ฐานข้อมูลเชิงเวลา 14
3.1 นิยามของฐานข้อมูลตามการสนับสนุนทางด้านเวลา 14
3.1.1 ฐานข้อมูลแบบสแนปช็อต (Snapshot Database) 14
3.1.2 ฐานข้อมูลแบบโรลแบ็ค (Rollback Database) 15
3.1.3 ฐานข้อมูลแบบอิงประวัติ (Historical Database) 15
3.1.4 ฐานข้อมูลแบบอิงเวลา (Bitemporal Database) 16
3.2 สถาปัตยกรรมของฐานข้อมูลเชิงเวลา 17
3.2.1 การรวมฟังก์ชันทางเวลาเข้ากับระบบจัดการฐานข้อมูล 18
3.2.2 การสร้างส่วนสนับสนุนทางเวลาเข้ากับไคเอนท์ของโปรแกรมประยุกต์ ..19
3.2.3 การใช้เซิร์ฟเวอร์เพื่อจัดการฐานข้อมูลเชิงเวลา 20
3.3 การนำไปสู่ฐานข้อมูลเชิงเวลา 20
3.3.1 Upward Compatibility 21
3.3.2 Temporal Upward Compatibility 23
3.3.3 การจัดการและสอบถามข้อมูลแบบเรียงลำดับ 24
3.3.4 การจัดการและสอบถามข้อมูลแบบไม่เรียงลำดับ 26

บทที่ 4 TQBE (Temporal Query-By-Example) 29
4.1 โครงสร้างพื้นฐาน (Basic Structure) 29
4.2 การสอบถามอย่างง่าย (Simple Queries) 30
4.3 การสอบถามข้อมูลแบบมีลำดับ (Queries with Ordering) 32
4.4 การสอบถามหลายความสัมพันธ์ (Queries on Several Relation) 33
4.5 ส่วนกำหนดเงื่อนไข (Condition Box) 34
4.6 ส่วนกำหนดช่วงเวลา (Time Range Box) 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.7 ส่วนกำหนดหน่วยย่อยเวลา (Time Granularity Box) 35
4.8 ส่วนกำหนดประเภทเวลา (Time State Box) 36
4.9 ผลลัพธ์ (Result Relation) 36
4.10 ตัวปฏิบัติการรวม (Aggregate Operations) 37
4.11 การจัดการข้อมูล 38
4.11.1 การเพิ่มข้อมูล 38
4.11.2 การลบข้อมูล 39
4.11.3 การแก้ไขข้อมูล 40

บทที่ 5 สถาปัตยกรรมของระบบ TQBE 41
5.1 สถาปัตยกรรมของ TQBE โดยรวม 41
5.2 ส่วนติดต่อกับผู้ใช้ 41
5.3 หน่วยปรับปรุงโครงสร้างตาราง 43
5.4 หน่วยระบุเวลาเข้ากับตารางฐานข้อมูล 43
5.5 หน่วยสร้างภาษาสอบถามเชิงโครงสร้าง 44
5.6 ส่วนแสดงผลลัพธ์ 44
5.7 หน่วยแปลงฐานข้อมูล 44
5.8 หน่วยควบคุมความถูกต้องของข้อมูล 45

บทที่ 6 การพัฒนาระบบ TQBE 46
6.1 อุปกรณ์ที่ใช้ในการพัฒนาระบบ TQBE 46
6.2 การพัฒนาระบบในส่วนต่าง ๆ 47
6.2.1 การพัฒนาส่วนการสอบถามข้อมูล 47
6.2.2 การพัฒนาส่วนการจัดการข้อมูล 55

บทที่ 7 การทดสอบและประเมินผลระบบ TQBE 61
7.1 การทดสอบระบบ TQBE 61
7.2 เปรียบเทียบระบบ TQBE กับภาษาอื่น ๆ 68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 8 สรุปและแนวทางการวิจัย 70
8.1 สรุป 70
8.2 ข้อเสนอแนะในการวิจัยต่อไป 71

เอกสารอ้างอิง 72
ภาคผนวก คู่มือการใช้ระบบ TQBE 74
ประวัติผู้เขียน 79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 ตัวปฏิบัติการทางเวลาสำหรับช่วงเวลาของ Allen.....7

2.2 ตาราง LOT_LOC.....62

2.3 เปรียบเทียบคุณสมบัติของภาษา TQBE กับภาษาอื่น.....68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

3.1 ฐานข้อมูลแบบสแนปช็อตบนแกนของเวลา....14

3.2 ฐานข้อมูลแบบโรลแบ็คบนแกนของเวลา....15

3.3 ฐานข้อมูลแบบอิงประวัติบนแกนของเวลา....16

3.4 ฐานข้อมูลแบบอิงเวลาบนแกนของเวลา....17

3.5 การรวมฟังก์ชันทางเวลาเข้ากับเคอร์เนลของระบบจัดการฐานข้อมูล....18

3.6 การใช้เซิร์ฟเวอร์เพื่อจัดการฐานข้อมูลเชิงเวลา....19

3.7 การใช้ไคล์เอนท์เพื่อจัดการฐานข้อมูลเชิงเวลา....21

3.8 การสอบถามข้อมูลแบบไม่อิงเวลา....22

3.9 การสอบถามข้อมูลเชิงเวลา....23

3.10 Temporal Upward Compatibility....24

3.11 Snapshot Reducibility ของ q^M ซึ่งสัมพันธ์กับ q....25

3.12 Snapshot Reducible ของการสอบถามเชิงเวลา q' บนความสัมพันธ์เชิงเวลา....26

3.13 การเปลี่ยนแปลงแก้ไขความสัมพันธ์เชิงเวลา....26

3.14 การสอบถามข้อมูลเชิงเวลาแบบไม่เรียงลำดับ q' บนความสัมพันธ์เชิงเวลา....27

3.15 การเปลี่ยนแปลงแก้ไขข้อมูลเชิงเวลาแบบไม่เรียงลำดับ u' บนความสัมพันธ์เชิงเวลา....28

4.1 ตารางโครงร่าง (Skeleton table)....30

5.1 สถาปัตยกรรมของระบบ TQBE โดยรวม....42

5.2 โครงสร้างตารางซึ่งมีส่วนสนับสนุนทางเวลา....43

5.3 ผลลัพธ์ที่ได้จากหน่วยระบุเวลา....44

5.4 ผลลัพธ์ที่ได้จากหน่วยสร้างภาษาสอบถามเชิงโครงสร้าง....44

6.1 ผังขั้นตอนการพัฒนาและการทำงานในส่วน Table....48

6.2 ผังขั้นตอนการพัฒนาและการทำงานในส่วน Column....50

6.3 ผังขั้นตอนการพัฒนาและการทำงานในส่วน Condition....51

6.4 ผังขั้นตอนการพัฒนาและการทำงานในส่วน Time Range....53

6.5 ผังขั้นตอนการพัฒนาและการทำงานในส่วน SQL....54

6.6 ผังขั้นตอนการพัฒนาและการทำงานของการเพิ่มข้อมูลแบบปัจจุบัน (current)....56

6.7 ผังขั้นตอนการพัฒนาและการทำงานของการลบข้อมูลแบบปัจจุบัน (current)....56

6.8 ผังขั้นตอนการพัฒนาและการทำงานของการแก้ไขข้อมูลแบบปัจจุบัน (current)....57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

6.9 ผังขั้นตอนการพัฒนาและการทำงานของการเพิ่มข้อมูลแบบเรียงลำดับ (sequence) 58

6.10 ผังขั้นตอนการพัฒนาและการทำงานของการลบข้อมูลแบบเรียงลำดับ (sequence) 59

6.11 ผังขั้นตอนการพัฒนาและการทำงานของการแก้ไขข้อมูลแบบเรียงลำดับ (sequence) 59

6.12 ผังขั้นตอนการพัฒนาและการทำงานของการจัดการข้อมูลแบบไม่เรียงลำดับ (Nonsequence) 60

ก.1 หน้าจอการสอบถามข้อมูลเชิงเวลาส่วน Table 75

ก.2 หน้าจอการสอบถามข้อมูลเชิงเวลาส่วน Column 75

ก.3 หน้าจอการสอบถามข้อมูลเชิงเวลาส่วน Condition 76

ก.4 หน้าจอการสอบถามข้อมูลเชิงเวลาส่วน Time Range 76

ก.5 หน้าจอการสอบถามข้อมูลเชิงเวลาส่วน SQL 77

ก.6 หน้าจอการสอบถามข้อมูลเชิงเวลาส่วน Output 77

ก.7 หน้าจอการจัดการข้อมูลเชิงเวลา 78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ฐานข้อมูลแบบไม่อิงเวลาได้แก่ฐานข้อมูลเชิงสัมพันธ์ หรือฐานข้อมูลเชิงวัตถุสัมพันธ์ จะเก็บค่าข้อมูล ณ.ขณะเวลาหนึ่งเท่านั้น ซึ่งปกติเป็นค่าของข้อมูล ณ.ปัจจุบัน การเปลี่ยนแปลงข้อมูลจากสถานะหนึ่งไปเป็นอีกสถานะหนึ่ง จะมีการแทนที่ค่าข้อมูลเก่าด้วยค่าข้อมูลใหม่ ส่วนการสอบถามข้อมูลนัยว่าข้อมูลถูกต้องทันสมัยตลอดเวลา นั่นแสดงว่าความจริงทั้งหมดถูกเก็บไว้ในฐานข้อมูลถูกต้อง ณ.ขณะเวลาที่มีการสอบถามข้อมูลเท่านั้น

โปรแกรมประยุกต์จำนวนมากมีความต้องการในการเก็บสถานะเดิมของข้อมูลหรือแม้กระทั่งสถานะของข้อมูลในอนาคต ตัวอย่างของโปรแกรมประยุกต์เช่น โปรแกรมทางด้านการเงินในการเก็บข้อมูลประวัติตลาดหุ้น ทางด้านการประกันเช่นเก็บข้อมูลที่มีผลกระทบกับกรมธรรม์ ระบบจองตั๋วของโรงแรมเช่นการเก็บข้อมูลการจองห้อง ระบบจัดการสารสนเทศทางการแพทย์เช่นเก็บประวัติคนไข้ ระบบสนับสนุนทางการตัดสินใจเช่นการวางแผนสำหรับเหตุการณ์ในอนาคต ในการเพิ่มแอททริบิวต์ของเวลาเข้าไปในโครงสร้างตารางของโมเดลข้อมูลแบบไม่อิงเวลาเพื่อเก็บข้อมูลที่แปรตามเวลานั้นทำได้ไม่ยากนัก แต่ปัญหาใหญ่คือการจัดการข้อมูลเชิงเวลา และการสอบถามข้อมูลที่แปรตามเวลาโดยใช้ภาษาสอบถามเชิงโครงสร้างจะต้องใช้ประโยคคำสั่งที่ยาวและในบางกรณีจะซับซ้อนมาก อีกทั้งการกำหนดกฎบังคับความถูกต้องซึ่งต้องควบคุมสถานะของข้อมูล ณ.ขณะเวลาใดๆนั้นก็ทำได้ยาก ดังรายละเอียดของทฤษฎีเวลา [11]

ระบบจัดการฐานข้อมูลเชิงพาณิชย์ในปัจจุบันยังขาดส่วนสนับสนุนสำหรับข้อมูลเชิงเวลาที่เหมาะสม โปรแกรมประยุกต์ส่วนใหญ่หากมีความต้องการในการจัดการข้อมูลที่แปรเปลี่ยนตามเวลาจะต้องมีการพัฒนาโปรแกรมประยุกต์พิเศษ เพื่อสนับสนุนข้อมูลดังกล่าวทำให้ต้องใช้จำนวนคนมาก ระยะเวลาในการพัฒนายาวนาน และสิ้นเปลืองค่าใช้จ่ายในการพัฒนาสูง

1.2 ความมุ่งหมายและวัตถุประสงค์

วัตถุประสงค์หลักสำหรับงานวิจัยนี้คือการจัดการกับข้อมูลเชิงเวลาแบ่งออกเป็นข้อๆดังนี้

1. เสนอแนวทางในการจัดการกับข้อมูลเชิงเวลาโดยใช้คิวบีอีอิงเวลา
2. ศึกษาทฤษฎีและวิธีการสร้างระบบจัดการฐานข้อมูลเชิงสัมพันธ์แบบคิวบีอีอิงเวลา
3. พัฒนาและปรับปรุงโครงสร้างฐานข้อมูลสำหรับข้อมูลเชิงเวลา
4. แสดงผลที่ได้จากการจัดการและสอบถามข้อมูลเชิงเวลาในรูปแบบภาษาสอบถามเชิงโครงสร้างและผลลัพท์ที่ได้จากการสอบถามข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ทฤษฎีและแนวคิดที่ใช้ในการวิจัย

ระบบจัดการฐานข้อมูลเชิงเวลา TQBE ได้ใช้แนวคิดของการสอบถามแบบใช้ตัวอย่าง (QBE : Query-by-Example) ผสมผสานกับการแนวคิดการจัดการข้อมูลเชิงเวลาในรูปแบบ current sequence และ nonsequence โดยใช้กฎบังคับความถูกต้องในการจัดการและสอบถามข้อมูลเชิงเวลาในการพัฒนาระบบเป็นสำคัญ

1.4 ขอบเขตการวิจัย

สำหรับขอบเขตของงานวิจัยนี้ ได้ตั้งวัตถุประสงค์ของการพัฒนาระบบ TQBE ไว้เพียงการสอบถามข้อมูลและจัดการข้อมูลที่มีการระบุเวลาที่กระทำกับแถว (tuple timestamping) เฉพาะค่า valid time จึงสามารถทราบเฉพาะประวัติของข้อมูลเท่านั้น ยังไม่มีสามารถทราบสถานะใดๆ ของข้อมูลและการเปลี่ยนแปลงฐานข้อมูลได้

1.5 ส่วนประกอบของวิทยานิพนธ์

วิทยานิพนธ์นี้ได้แบ่งออกเป็นหลายบทด้วยกัน ซึ่งประกอบด้วย ทฤษฎีเบื้องต้นที่จะทำความเข้าใจความหมายเวลา ระบบฐานข้อมูลเชิงเวลา การสอบถามแบบใช้ตัวอย่าง การออกแบบและพัฒนาระบบ TQBE โดยแบ่งออกเป็นบทๆดังนี้คือ

บทที่ 2 หลักการและความหมายของเวลา ในบทนี้จะกล่าวถึงความหมายของเวลาและการระบุเวลาเข้ากับข้อมูลด้วยวิธีการต่างๆ

บทที่ 3 ฐานข้อมูลเชิงเวลา บทนี้จะกล่าวถึงนิยามของฐานข้อมูลตามแนวคิดของเวลา สถาปัตยกรรม และการนำไปสู่ฐานข้อมูลเชิงเวลา

บทที่ 4 TQBE (Temporal Query-By-Example) บทนี้อธิบายการสอบถามแบบใช้ตัวอย่างสำหรับฐานข้อมูลเชิงเวลา

บทที่ 5 สถาปัตยกรรมของระบบ TQBE บทนี้อธิบายสถาปัตยกรรมของระบบที่ได้พัฒนาขึ้น

บทที่ 6 การพัฒนาระบบ TQBE ในบทนี้จะอธิบายขั้นตอนการพัฒนาระบบในส่วนต่างๆ อย่างเป็นระบบ พร้อมทั้งแสดงผังงานทั้งหมดช่วยให้เข้าใจวิธีการพัฒนาและการทำงานเป็นขั้นตอน

บทที่ 7 การทดสอบและประเมินผลระบบ TQBE ในบทนี้ได้ใช้ตัวอย่างในการทดสอบระบบ เปรียบเทียบกับระบบอื่นๆ

บทที่ 8 สรุปและแนวทางการวิจัย บทนี้จะสรุปผลการทำงานและอุปสรรคในการทำงานและแนวทางวิจัยต่อไป

ส่วนท้ายของวิทยานิพนธ์จะเป็นภาคผนวก ซึ่งเป็นคู่มือการใช้งานระบบ TQBE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

หลักการและความหมายของเวลา

ในบทนี้จะแนะนำความรู้เบื้องต้นเกี่ยวกับความหมายของเวลา โมเดลของเวลา การระบุเวลา (timestamping) ตัวปฏิบัติการเชิงเวลา และโครงสร้างข้อมูล

2.1 ความหมายของเวลา

ในชีวิตประจำวันเหตุการณ์ต่างๆมักถูกกำหนดเวลาด้วยเวลาเพียงแกนเดียว และมีการอ้างถึงเวลาในรูปแบบของจุดเวลาหรือช่วงเวลา สำหรับฐานข้อมูลเชิงเวลานั้น มีการใช้เวลามากกว่า 1 แกน เพื่อเก็บรูปแบบของความสัมพันธ์ระหว่างเวลากับข้อมูล ในหัวข้อต่อไปจะกล่าวถึงนิยามความหมายของเวลา เพื่อใช้ในการแบ่งประเภทของฐานข้อมูลเชิงเวลาตามลักษณะของประเภทเวลาที่ใช้ในฐานข้อมูลนั้น

2.1.1 User-defined Time

เวลาซึ่งไม่มีการแปลความหมาย ในการระบุเวลาเรียกว่า user-defined time [2] เช่นแอททริบิวต์ ชื่อ Birthdate เป็น user-defined time เพราะผู้ใช้เป็นผู้แปลความหมายของเวลาด้วยตัวเอง ในขณะที่ระบบจัดการฐานข้อมูลมองว่าเป็นเพียงแอททริบิวต์หนึ่งในตารางเท่านั้น

โมเดลข้อมูลทั่วไปสนับสนุน user-defined time โดยผู้ใช้สามารถกำหนดประเภทของข้อมูลเป็น date หรือ timestamp ให้กับค่าแอททริบิวต์ ค่าที่ได้สามารถอ้างถึงเวลาใน อดีต ปัจจุบันและอนาคต ค่าของ user-defined time ถูกเปลี่ยนแปลงและแก้ไขโดยผู้ใช้

ความแตกต่างระหว่างการแปลความหมาย กับการไม่แปลความหมายแอททริบิวต์ของเวลาคือ ในโมเดลข้อมูลเชิงเวลาใช้โครงสร้างข้อมูลของโมเดลข้อมูลแบบไม่อิงเวลา ในการเก็บข้อมูลเชิงเวลาด้วยการเพิ่มค่าแอททริบิวต์ของเวลาเข้าไปในโครงสร้างตาราง(schema) เมื่อมีการสอบถามข้อมูลโดยมีความหมายของคำเชิงเวลา โดยใช้พีชคณิตเชิงเวลาในการเข้าถึงข้อมูลพิเศษที่เป็นแอททริบิวต์ของเวลาและแปลความหมายให้ผลลัพธ์กลับมา ซึ่งไม่สามารถเข้าถึงแอททริบิวต์ของ user-defined time ด้วยวิธีดังกล่าว

2.1.2 Valid Time

โมเดลข้อมูลเชิงเวลาจะต้องสามารถแสดงให้ได้ว่าเมื่อไรที่ความจริง (Fact.) เป็นจริง (True) ตัวอย่างเช่น เราอยากทราบว่าฝากเงินในบัญชีเมื่อไรและถอนเงินอีกเมื่อไร บางครั้งเรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องการบันทึกเหตุการณ์ในอนาคต เช่น จองตั๋วการแสดงล่วงหน้า เวลาที่ถูกบันทึกโดยคำนึงว่าเมื่อไรที่เหตุการณ์นั้นเป็นจริง หรือเกิดขึ้นจริงเรียกว่า valid time โดยที่ค่า valid time จะต้องมีการเปลี่ยนค่าเมื่อมีการเปลี่ยนแปลงแก้ไขหรือเพิ่มข้อมูล

2.1.3 Transaction Time

ในฐานข้อมูลเชิงเวลานอกจากการบันทึกเนื้อหาข้อมูลแล้วยังต้องบันทึกประวัติของข้อมูลอีกด้วย บ่อยครั้งที่ข้อมูลไม่สามารถบันทึกลงในฐานข้อมูลได้ ณ. ขณะเวลานั้นอาจจะเป็นไปได้ว่าเนื่องจากความล่าช้าในการประมวลผลข่าวสาร ดังนั้นจึงเกิดช่วงห่างของเวลาระหว่างข้อมูลที่เป็นจริงในโลกแห่งความเป็นจริงกับข้อมูลที่บันทึกลงในฐานข้อมูลจริงๆ ซึ่งการที่คอยเก็บข้อมูลของเวลาที่ทำการบันทึกลงในฐานข้อมูลเรียกว่า transaction time

ตัวอย่างในการฝากเงินจะเห็นได้ว่า valid time ถูกใช้เพื่อบันทึกประวัติการฝากถอนเงินในธนาคารและสามารถดูประวัติการเปลี่ยนแปลงค่าเหล่านั้นได้ สมมุติว่าพนักงานธนาคารได้บันทึกข้อมูลวันที่ถอนเงินหรือจำนวนเงินที่ผิดพลาด ในกรณีนี้การแก้ไขค่าเกิดผิดพลาด เนื่องจากธนาคารต้องการที่จะทราบว่า เมื่อไรที่บัญชีดังกล่าวถูกแก้ไขเมื่อพนักงานคนนั้นแก้ไขค่า ระบบจะทำการตั้งค่าเก่าของ transaction time ไปเป็นเวลาสิ้นสุดของเวลาปัจจุบัน และค่าเริ่มต้นของ transaction time สามารถที่จะแสดงได้ว่าเป็นการบันทึกประวัติของการแก้ไขข้อมูล หมายความว่าการระบุเวลาข้อมูลโดยใช้ transaction time เป็นประโยชน์ต่อการค้นหาประวัติการจัดการข้อมูลและความผิดพลาดที่เกิดขึ้น ในขณะที่การระบุเวลาด้วย valid time ทำให้สามารถสอบถามข้อมูลที่มีการเปลี่ยนแปลงตามเวลาได้

ค่าของ transaction time จะไม่ช้ากว่าเวลาปัจจุบัน เนื่องจาก transaction time จะสะท้อนถึงเวลาซึ่งมีการจัดการกับฐานข้อมูลจริงๆ ระบบจัดการฐานข้อมูลจะทำการบันทึกค่า transaction time เองโดยอัตโนมัติ

2.2 โมเดลของเวลา

มีการใช้โมเดลพื้นฐานของเวลา 3 แบบ [3] ในฐานข้อมูลเชิงเวลา คือแบบต่อเนื่อง (Continuous) แบบพิกัด (Dense) และแบบไม่ต่อเนื่อง (Discrete) คำถามคือแล้วเวลาที่เป็นจุด และเวลาที่เป็นช่วงจะแทนได้อย่างไร อีกทั้งตัวปฏิบัติการและตัวเปรียบเทียบจะกำหนดที่ใด ในหัวข้อนี้จะอธิบายถึงความหมายของเวลาต่างๆหลากหลาย และผลกระทบต่อการกำหนดโมเดลข้อมูลเชิงเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 ประเภทโมเดลของเวลา

โดยธรรมชาติเวลาจะมีลักษณะต่อเนื่อง นักวิทยาศาสตร์ได้สมมุติว่าเวลาได้เริ่มต้นจากการกำเนิดของโลกครั้งแรก ในชีวิตประจำวันมีการแบ่งปริมาณหรือขนาดของเวลาเพื่อกำหนดเวลาด้วยหน่วยย่อยของเวลาแตกต่างกัน เช่น ปี เดือน วัน ชั่วโมง นาที วินาที ซึ่งใช้ปฏิทินเป็นโครงสร้างพื้นฐานของเวลา

โมเดลของเวลาในฐานข้อมูลจำเป็นจะต้องกำหนดเวลาที่แน่นอน ซึ่งไม่ได้หมายความว่าจำเป็นจะต้องมีการใช้เวลาทุกรูปแบบในฐานข้อมูล

ในฐานข้อมูลเชิงเวลามีการใช้โมเดลของเวลา 3 แบบคือ แบบต่อเนื่อง แบบพิกัด และแบบไม่ต่อเนื่อง เวลาจะถูกเข้าพวกเป็นเซตของจำนวนซึ่งสามารถแทนได้ด้วยแกนบนระบบโคออดิเนท

โมเดลของเวลาแบบต่อเนื่อง (continuous) นั้นเป็นเวลาเป็นจำนวนจริง (real number)ในทางคณิตศาสตร์แต่ละค่าของจำนวนจริงคือจุดเวลา (point of time) ดังนั้นระหว่าง 2 จุดใดๆบนแกนเวลาจะมีจุดเวลาจุดอื่นๆอีก โมเดลของเวลาแบบนี้มีความถูกต้องแม่นยำสูงสุด เพราะโมเดลของข้อมูลเชิงเวลานั้นใช้ในเครื่องดิจิตอล

โมเดลของเวลาแบบพิกัด (dense) เวลาเป็นเหมือนเศษส่วน (rational number) และโมเดลของเวลาแบบไม่ต่อเนื่อง (discrete) เวลาเป็นเสมือนจำนวนเต็ม (integer)

ในแง่ของทางเลือกโมเดลทั้ง 3 ไม่สำคัญนัก เมื่อคำนึงถึงการใช้โมเดลของเวลาบนอุปกรณ์ที่คำนวณแบบไม่ต่อเนื่อง(discrete) ฉะนั้นเวลาจึงควรจะเป็นช่วงหรือไม่ต่อเนื่องด้วย

2.2.2 หน่วยย่อยเวลา (Chronons)

เวลาในโมเดลต่างๆสามารถแสดงได้บนแกนเวลา ซึ่งแกนเวลาสามารถแสดงเซ็ตของจุดเวลาหน่วยย่อยซึ่งเล็กที่สุดของเวลาบนแกนเวลาที่เรียกว่า Chronon [4] ดังนั้น Chronon ต่างๆเป็นช่วงเวลาย่อยของช่วงเวลาที่แน่นอนบนแกนเวลา สำหรับโมเดลแบบไม่ต่อเนื่อง Chronon และจุดเวลามองดูเหมือนเป็นสิ่งเดียวกัน ซึ่งสามารถแทนได้ด้วยจำนวนเต็ม (Integer)

2.2.3 จุดเวลาและเหตุการณ์ (Time Instant and Events)

Time Instant คือจุดเวลาบนแกนเวลาในขณะที่ Event คือความจริง ณ เวลานั้น Time Instant เทียบได้กับ Chronon ในหัวข้อที่แล้ว และ Event เกิดขึ้นที่ Chronon t

2.2.4 ช่วงเวลาและส่วนประกอบเชิงเวลา (Time Interval and Temporal Elements)

ฐานข้อมูลเชิงเวลาเก็บความจริงและเวลาที่ความจริงนั้นเป็นจริง ดังนั้นในทางทฤษฎีทุกๆจุดของเวลา Time Instant ของความจริงที่เป็นจริงต้องเก็บค่าในฐานข้อมูลเชิงเวลา ในกรณีที่เซต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของค่าของเวลามีความต่อเนื่องสามารถที่จะแทนค่าด้วยช่วงเวลา(time interval) ดังนั้นจึงสามารถใช้ช่วงเวลากับโมเดลของเวลาซึ่งความจริงนั้นเป็นจริง

ช่วงเวลา (time interval) ประกอบด้วยเวลาเริ่มต้น S และ เวลาสิ้นสุด E หรือแทนด้วยค่าขอบเขตต่ำสุด (lower bound) และค่าขอบเขตสูงสุด (upper bound) ตามลำดับ ค่าเวลา S และ E เป็นค่าที่ต่ำที่สุดและสูงที่สุดในเซตของเวลาที่ต่อเนื่องกัน โดยใช้ตัวปฏิบัติการเปรียบเทียบ $<$ และ $\leq$ ต่อไปนี้เป็นการแสดงความเป็นไปได้ในการกำหนดช่วงเวลา

$$[S-E] = \{ t \mid S \leq t \leq E \}$$

$$(S-E) = \{ t \mid S < t < E \}$$

$$[S-E) = \{ t \mid S \leq t < E \}$$

$$(S-E] = \{ t \mid S < t \leq E \}$$

โดยที่ t แทนค่าจุดเวลา (time instant) ฉะนั้นค่า t สามารถแทนได้ด้วยค่า S-E

สมมุติให้ I_1= {1980-1990), I_2 = [1992-1994) และ I_3= [1975-1996)

$$I_1 \cup I_2 = \{[1980\text{-}1990) , [1992\text{-}1994)\}$$

$$I_1 - I_3 = \{ \}$$

$$I_3 - I_2 = \{[1975\text{-}1992) , [1994\text{-}1996)\}$$

2.2.5 ลักษณะการเปรียบเทียบสำหรับช่วงเวลา (Comparison Predicate for Time Interval)

Allen ได้เสนอพีชคณิตเชิงเวลาเพื่อเปรียบเทียบค่าของช่วงเวลา 13 ข้อ [5] ได้แก่ before, after, meets, met-by, overlaps, ovelaped-by, during, includes, starts, started-by, finish, finished-by และ equal ความสัมพันธ์ระหว่างช่วงเวลา 2 ค่า ความสัมพันธ์ระหว่างคู่ของช่วงเวลา x แทนด้วย $x^- < x^+$ ซึ่งเป็นค่าขอบเขตต่ำสุดและค่าขอบเขตสูงสุดของช่วงเวลาตามลำดับ ดังแสดงในตารางที่ 2.1

2.3 การระบุเวลากับข้อมูล (Timestamping Data)

ในโมเดลข้อมูลเชิงเวลาความจริง (facts) ถูกแทนด้วยข้อมูลซึ่งมีการระบุเวลา ซึ่งแสดงเวลาที่ข้อมูลดังกล่าวเป็นจริง (valid) หรือเก็บลงในฐานข้อมูล คำถามคือหน่วยย่อยของข้อมูลที่จะพิจารณาว่าเป็นค่าเดียวหรือเป็นการรวมของค่าหลายๆค่าในเอนทิตี้ (entity) เดียวกัน หรือว่าเป็นเซตของหลายๆเอนทิตี้ซึ่งรวมกลุ่มกันหรือส่วนของโมเดลในฐานข้อมูล

เมื่อมีการระบุเวลากับข้อมูล จะต้องมีการกำหนดระดับหน่วยย่อยของข้อมูลที่จะระบุเวลาและประเภทของการระบุเวลาที่จะใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ตัวปฏิบัติการทางเวลาสำหรับช่วงเวลาของ Allen

Basic Relation	Symbol	End Points
x before y	$<$	$x^+ < y^-$
y after x	$>$	
x meets y	m	$x^+ = y^-$
y met-by x	$m^{\cup}$	
x overlaps y	O	$x^- < y^- < x^+ , x^+ < y^+$
y overl.-by x	$O^{\cup}$	
x during y	d	$x^- > y^- , x^+ < y^+$
y includes x	$d^{\cup}$	
x starts y	s	$x^- = y^- , x^+ < y^+$
y started-by x	$S^{\cup}$	
x finishes y	F	$x^+ = y^+ , x^- > y^-$
y finished-by x	$F^{\cup}$	
x equal y	$\equiv$	$x^- = y^- , x^+ = y^+$

2.3.1 ความหมายของการระบุเวลา

ระบบจัดการฐานข้อมูลเชิงเวลาจะเก็บความจริง (facts) ซึ่งมีการระบุเวลาด้วยความหมายของคำซึ่งแตกต่างกัน โมเดลของข้อมูลเชิงเวลาที่ได้นำเสนอนี้แตกต่างกันตามระดับของหน่วยย่อยเวลาที่ใช้ หน่วยของข้อมูลอาจจะเป็นค่าแอททริบิวต์จนถึงทับเปิล หรือออบเจ็กต์จนถึงกลุ่มของทับเปิล หรือออบเจ็กต์จนถึงฐานข้อมูลรวมไปจนถึงแบบแผนฐานข้อมูล หรือกฎบังคับจะต้องมีการพิจารณาด้วยการระบุเวลาจะต้องได้รับการสนับสนุนจากโมเดลข้อมูลเชิงเวลา โดยที่มีการตระเตรียมโมเดลข้อมูลเชิงเวลาให้ ส่วนหน่วยย่อยที่จะทำการระบุเวลาขึ้นอยู่กับผู้ใช้

2.3.1.1 การระบุเวลาที่กระทำกับทับเปิล (Tuple Timestamping)

การระบุเวลาที่กระทำกับทับเปิลนี้ปกติใช้ในโมเดลข้อมูลเชิงเวลาสัมพันธ์ซึ่งสนับสนุนความสัมพันธ์แบบ 1NF ในโมเดลข้อมูลมีการระบุเวลาเพิ่มเข้าไปในทุกๆทับเปิลของความสัมพันธ์ ในกรณีของข้อมูลเชิงประวัติ แต่ละทับเปิลจะถูกระบุด้วยค่า valid time ในฐานข้อมูลเชิงเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Bitemporal database) จะระบุเวลาทั้งค่า valid time และ transaction time โดยปกติการระบุเวลาที่กระทำกับทับเปิลโดยใช้โมเดลของฐานข้อมูลแบบขยายส่วนมาจากฐานข้อมูลแบบสแนปช็อต

ข้อเสียของการระบุเวลาโดยวิธีนี้คือ ข้อมูลจะถูกกระจายไปยังทุกๆทับเปิลโดยที่แต่ละทับเปิลแทนสถานะต่างๆในช่วงเวลาที่กำหนด ทำให้เกิดความซ้ำซ้อนของข้อมูลจำนวนมาก (data redundancy) ตัวอย่างเช่น ฐานข้อมูลของพนักงานที่มีการเปลี่ยนแปลงเงินเดือน ซึ่งเป็นคอลัมน์หนึ่งในตารางและมีการระบุเวลาทำให้ฐานข้อมูลในคอลัมน์อื่นๆเช่น เลขที่พนักงาน แผนก ถูกทำให้ซ้ำซ้อนไปด้วย

ตัวอย่างที่ 2.1 ใช้การระบุเวลาที่กระทำกับแถวเพื่อแสดงว่าสมศรีได้ทำงานในแผนกสารสนเทศตั้งแต่ปี 1994 ซึ่งรหัสประจำตัวคือ 100 เงินเดือน 20,000 บาท ข้อมูลดังกล่าวถูกเก็บใน 2 แถวคือ ตั้งแต่ปี 1994 จนถึงปี 1996 จำนวน 1 แถว และตั้งแต่ปี 1996 เป็นต้นไปจำนวน 1 แถวดังแสดง

<100, สมศรี, 15,000, IS> from 1994 until 1996

<100, สมศรี, 20,000, IS> from 1996 until changed

2.3.1.2 การระบุเวลาที่กระทำกับแอททริบิวต์ (Attribute Time-stamping)

การระบุเวลาที่กระทำกับแอททริบิวต์ไม่ทำให้เกิดข้อมูลซ้ำซ้อนเพราะจะทำการเพิ่มเวลาเข้าไปที่แต่ละแอททริบิวต์ เพราะข้อมูลประวัติถูกเก็บแยกกันในแต่ละแอททริบิวต์

โดยการใช้โครงสร้างตารางของฐานข้อมูลซึ่งขยายส่วนความต้องการของการระบุเวลาที่กระทำกับแอททริบิวต์ ซึ่งโมเดลข้อมูลสนับสนุนความสัมพันธ์แบบ NFNF หรือ ออบเจ็กเชิงซ้อน (complex object)ซึ่งแอททริบิวต์ที่ถูกระบุเวลาทั้งหมดอยู่ในรูปเชิงซ้อน คือเก็บค่าของแอททริบิวต์พร้อมกับค่าเวลาที่ระบุ

ตัวอย่างที่ 2.2 โดยใช้การระบุเวลาที่กระทำกับแอททริบิวต์ เราสามารถเก็บค่าข้อมูลจากตัวอย่างที่แล้วเพียงแค่ทับเปิลเดียว

< { |100 from 1994 until changed | },
{ | สมศรี from 1994 until changed | },
{ | 15,000 from 1994 until 1996 |, | 20,000 from 1996 until changed | },
{ | IS from 1994 until changed | } >

ทับเปิลประกอบด้วยประวัติของแต่ละแอททริบิวต์ โดยที่ประวัติของแอททริบิวต์เป็นเซทของค่าเวลาที่ระบุทั้ง 2 ค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 วิธีการระบุเวลา

โมเดลข้อมูลเชิงเวลา แตกต่างกันในแง่การระบุเวลาที่กระทำกับทับเปิลหรือแอททริบิวต์ การใช้แบบแผนฐานข้อมูลแบบขยายส่วน ทางเลือกของการระบุเวลาจะจำกัดภายใต้เงื่อนไขโมเดลข้อมูล

วิธีการระบุเวลาที่กระทำกับข้อมูลคือการระบุเวลาด้วยจุดเวลา time instant การระบุเวลาด้วยช่วงเวลา (time interval) และการระบุเวลาด้วยองค์ประกอบของเวลา (time element)โมเดลข้อมูลเชิงสัมพันธ์แบบอิงเวลาภายใต้เงื่อนไข 1NF ถูกจำกัดการระบุเวลา ซึ่งสามารถอ้างถึงค่าสเกลาร์ที่เพิ่มขึ้นของค่าแอททริบิวต์ของทับเปิล โมเดลข้อมูลซึ่งสนับสนุนความสัมพันธ์แบบ NFNF หรือค่าแอททริบิวต์เชิงวัตถุ สามารถระบุเวลาที่ซับซ้อนได้

2.3.2.1 การระบุเวลาด้วยจุดเวลา (Timestamping with Time Instant)

ข้อมูลที่ถูกระบุเวลาด้วยจุดเวลาปกติเป็นจริง ณ เวลาที่กำหนดเท่านั้นความสัมพันธ์ประกอบด้วย ข้อมูลที่ถูกระบุเวลาด้วยจุดเวลาถูกเรียกว่าตารางของเหตุการณ์ (event tables)การระบุเวลาที่กระทำกับทับเปิลด้วยจุดเวลาทำได้โดยการขยายโครงสร้างตารางของฐานข้อมูลด้วยแอททริบิวต์ประเภทวัน date ที่ซ่อนไว้ ความสัมพันธ์ยังเป็น 1NF

ตัวอย่างที่ 2.3 ตารางความสัมพันธ์ของพนักงานซึ่งไม่อิงเวลา เปลี่ยนให้เป็นตารางอีเวนท์โดยการระบุเวลาที่กระทำกับทับเปิลด้วยค่าจุดเวลา

Emp ID	Name	Salary	Dept	At
100	Maira	15,000	IS	1994
101	Adrain	9,000	IS	1996
102	Alain	9,000	IS	1995
103	Antonia	11,000	IS	1996
104	Gabrio	10,000	IS	1996
105	Andreas	9,000	IS	1993

ข้อมูลการระบุเวลาในตารางอีเวนท์ด้วยจุดเวลานั้นโครงสร้างตารางฐานข้อมูลถูกขยายส่วน จากตาราง แอททริบิวต์ At คือส่วนที่เพิ่มเข้ามาในตารางความสัมพันธ์ ตารางอีเวนท์ Employees ประกอบด้วยแอททริบิวต์ date, name, employment-no และ initial salary

ความสัมพันธ์แบบ NFNF อนุญาติให้ข้อมูลทำการระบุเวลาในระดับแอททริบิวต์ได้ การระบุเวลาที่กระทำกับแอททริบิวต์ด้วยจุดเวลาสามารถที่จะทำได้ แต่ละแอททริบิวต์ถูกเปลี่ยนไปเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอททริบิวต์ที่สลับซับซ้อนด้วยค่าแอททริบิวต์ที่เป็นส่วนขยายคือ At ซึ่งเป็นแอททริบิวต์ประเภท Date

2.3.2.2 การระบุเวลาด้วยช่วงเวลา (Timestamping with Time Intervals)

ดังได้กล่าวไปแล้วว่าการระบุเวลาที่เกี่ยวข้องกับทับเปิล ตัวอย่างเช่น เมื่อการรวมกันของค่าในทับเปิลเป็นจริง โดยใช้ส่วนขยายโครงสร้างตารางฐานข้อมูล การระบุเวลาด้วยช่วงเวลาเป็นความสัมพันธ์แบบ 1NF ที่ขอบเขตต่ำสุด (lower bound)และที่ขอบเขตสูงสุด (upper bound) ของช่วงเวลาถูกแทนด้วยแอททริบิวต์ของเวลาที่เพิ่มขึ้น 2 ค่า ส่วนตารางข้อมูลเชิงเวลา (bitemporal) ต้องใช้แอททริบิวต์ 4 ค่า ใช้กับ valid time 2 ค่าและใช้กับ transaction time อีก 2 ค่า

การระบุเวลาที่กระทำกับทับเปิลด้วยช่วงเวลาทำให้เกิดความซ้ำซ้อนของข้อมูลดังได้กล่าวไปแล้ว การแก้ไขค่าในทับเปิลทำให้เกิดทับเปิลในความสัมพันธ์ขึ้นมาใหม่ ทำให้แอททริบิวต์อื่นๆ ที่ไม่เกี่ยวข้องเกิดความซ้ำซ้อนขึ้นด้วย รวมทั้งทับเปิลซึ่งเป็นจริงระหว่างช่วงเวลาที่ไม่คาบเกี่ยวกันถูกเก็บแยกเวลาแต่ละส่วนยิ่งทำให้ข้อมูลกระจัดกระจายเกิดทับเปิลมากขึ้นทำให้ซ้ำซ้อนยิ่งขึ้น เนื่องจากช่วงเวลาไม่มีคุณสมบัติปิดภายใต้ทฤษฎีของเซตไม่ว่าจะเป็น Intersection Difference หรือ Union ทำให้มีผลกระทบต่อการสอบถามข้อมูล

ตัวอย่างที่ 2.4 ตาราง employees ถูกขยายส่วนและทำการระบุเวลาที่กระทำกับทับเปิลมีแอทริบิวต์เพิ่มจากเดิม 2 ค่าคือ From และ To ซึ่งเก็บค่า valid time ค่าเริ่มต้นและค่าสุดท้ายของเวลาตามลำดับ

Emp ID	Name	Salary	Dept	From	To
100	Marina	15000	IS	1994	1996
100	Marina	20000	IS	1996	∞
101	Adrian	9000	IS	1996	∞
102	Alain	9000	IS	1995	∞
103	Antonia	11000	IS	1996	∞
104	Gabrio	10000	IS	1996	∞
105	Andreas	9000	IS	1993	∞
106	Martin	9000	IS	1989	1996
107	Tom	4500	Math	1988	1991
107	Tom	4500	Math	1993	1995

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการใช้ช่วงเวลาสมมติให้มีคุณสมบัติปิดที่ขอบเขตต่ำสุดและเปิดที่ขอบเขตสูงสุดและใช้เครื่องหมาย ∞ แทนค่าพิเศษคือ until changed สังเกตว่า Martin ลาออกจากมหาวิทยาลัยในปี ค.ศ. 1996 Marina ได้รับเงินเดือนขึ้นในปี1996 ฉะนั้นเธอจึงมีข้อมูล 2 แถวในตาราง Tom มีข้อมูล 2 แถวเช่นกันเนื่องจากทำงานในช่วงเวลาที่ไม่ต่อเนื่องกัน 2 ช่วงเวลา

การระบุเวลาที่กระทำกับแอททริบิวต์โดยใช้ช่วงเวลาแต่ละแอททริบิวต์ของความสัมพันธ์ถูกขยายเพิ่มให้เป็นแอททริบิวต์ที่ซับซ้อนขึ้น โดยการเพิ่ม 2 หรือ 4 แอททริบิวต์ เซตของแต่ละแอททริบิวต์ประกอบด้วยประวัติของแต่ละแอททริบิวต์ในทับเปิล ความซ้ำซ้อนของข้อมูลเกิดขึ้นได้ถ้าค่าเดียวกันปรากฏในหลายช่วงเวลา ระหว่างช่วงเวลาที่ไม่คาบเกี่ยวกัน เช่น Tom ได้รับการว่าจ้างในช่วงเวลาที่ต่างกันแอททริบิวต์อื่นๆจึงมีการซ้ำซ้อนกันในแต่ละช่วงเวลา

ตัวอย่างที่ 2.5 ตาราง Employees ถูกขยายส่วนโดยการระบุเวลาที่กระทำกับแอททริบิวต์ด้วยช่วงเวลา ค่าแอททริบิวต์ที่เพิ่มเข้ามาคือ From และ To ถูกเพิ่มเข้าไปในแต่ละแอททริบิวต์ พิจารณาพนักงานชื่อ Marina ได้รับการขึ้นเงินเดือนแต่ไม่ทำให้เกิดความซ้ำซ้อนของข้อมูล

Emp ID	Name	Salary	Dept
{<100,1994,∞>}	{< Marina,1994,∞>}	{15000,1994,1996 >, <20000,1996,∞>}	{< IS,1994, ∞>}
{<101,1996, ∞>}	{< Adrain,1996, ∞>}	{<9000,1996, ∞>}	{< IS,1994, ∞>}
…	…	…	…
{<107,1985,1991>, <107,1993,1995>}	{< Tom,1985,1991 >, < Tom,1993,1995>}	{<4500,19851991>, <4500,1993,1995>}	{< IS,1985,1991>, <IS,1993,1995>}

2.3.2.3 การระบุเวลาด้วยส่วนประกอบเชิงเวลา (Timestamping with Temporal Elements)

การระบุเวลาด้วยส่วนประกอบเชิงเวลา ข้อมูลเป็นจริงในช่วงเวลาที่ไม่คาบเกี่ยวกันหลายช่วงเวลาได้ รวมทั้งการระบุเวลาด้วยเวลาแบบนี้มีข้อดีคือ ส่วนประกอบเหล่านี้มีคุณสมบัติปิดภายใต้ทฤษฎีของเซตในแง่ตัวปฏิบัติการคือ Intersect Difference และ Union

การขยายโครงสร้างข้อมูลของโมเดลข้อมูลแบบไม่อิงเวลาเพื่อสนับสนุนการระบุเวลาด้วยส่วนประกอบเชิงเวลา เป็นวิธีการเดียวที่ทำได้ถ้าโมเดลข้อมูลสนับสนุนค่าแอททริบิวต์อะตอม ตัวอย่างที่ 2.6 ตาราง employees มีการขยายส่วน ส่วนประกอบเชิงเวลาโดยใช้การระบุเวลาที่กระทำกับทับเปิลด้วย valid time ดังตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Emp_ID	Name	Salary	Dept	Validity
100	Moira	15000	IS	{<1994,1996>}
100	Moira	20000	IS	{<1996, ∞>}
101	Adrian	9000	IS	{<1996, ∞>}
102	Alain	9000	IS	{<1995, ∞>}
103	Antonia	11000	IS	{<1996, ∞>}
104	Garbio	10000	IS	{<1996, ∞>}
105	Andreas	9000	IS	{<1993, ∞>}
106	Martin	9000	IS	{<1989,1996>}
107	Tom	4500	Math	{<1985,1991>,<1993,1995>}

เงินเดือนของพนักงานชื่อ Marina ทำให้เกิดความซ้ำซ้อนของข้อมูล ส่วนนักงานชื่อ Tom มีเพียงแถวเดียวเนื่องจากสามารถเก็บช่วงเวลาที่แตกต่างกันในส่วนประกอบเชิงเวลาได้

การลดความซ้ำซ้อนของข้อมูล ในการระบุข้อมูลที่กระทำกับแอททริบิวต์ทำได้โดยการให้แต่ละแอททริบิวต์ในแบบแผนฐานข้อมูลถูกขยายส่วนด้วยค่าส่วนประกอบเชิงเวลา

ตัวอย่าง 2.7 ตารางพนักงาน employees ถูกขยายส่วนโดยการระบุเวลาที่กระทำกับแอททริบิวต์ด้วยส่วนประกอบเชิงเวลา

Emp_ID	Name	Salary	Dept
{<100, {<1994,∞>}>}	{<Marina, {<1994,∞>}>}	{<15000, {<1994,1996>}>, <20000, {<1996, ∞>}>}	{<IS,{<1994, ∞>}>}
{<101,{<1996,∞>}>}	{<Adrian, {<1996, ∞>}>}	{<9000, {<1996, ∞>}>}	{<IS,{<1996, ∞>}>}
{<107,{<1985,1991>, <1993,1995>}>}	{<Tom, {<1995,1991>, <1993,1995>}>}	{<4500, {<1985,1991>, <1993,1995>}>}	{<IS, {<1985,1991>, <1993,1995>}>}

ดังตัวอย่างข้างต้นการระบุเวลาที่กระทำกับแอททริบิวต์สามารถลดความซ้ำซ้อนลงได้ เช่น Marina มีการเพิ่มเงินเดือน แต่ก็ใช้เพียงแถวเดียว การระบุเวลาด้วยการใช้เวลาประเภทนี้สามารถลดความซ้ำซ้อนลงได้ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การระบุเวลาที่กระทำกับแอททริบิวต์ด้วยส่วนประกอบเชิงเวลา สามารถลดความซ้ำซ้อนได้ทั้งหมด แต่อย่างไรก็ตามตารางความสัมพันธ์ที่ได้ค่อนข้างอ่านยาก อีกทั้งการสอบถามข้อมูลในช่วงเวลาที่แตกต่างกันก็ยังยากตามไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ฐานข้อมูลเชิงเวลา

เป็นที่ยอมรับกันโดยทั่วไปว่า ฐานข้อมูลซึ่งสามารถเก็บบันทึกการเปลี่ยนแปลง ข้อมูลสนับสนุนการสอบถามข้อมูลและจัดการข้อมูลที่แปรตามเวลาได้เรียกว่าฐานข้อมูลเชิงเวลา [6]

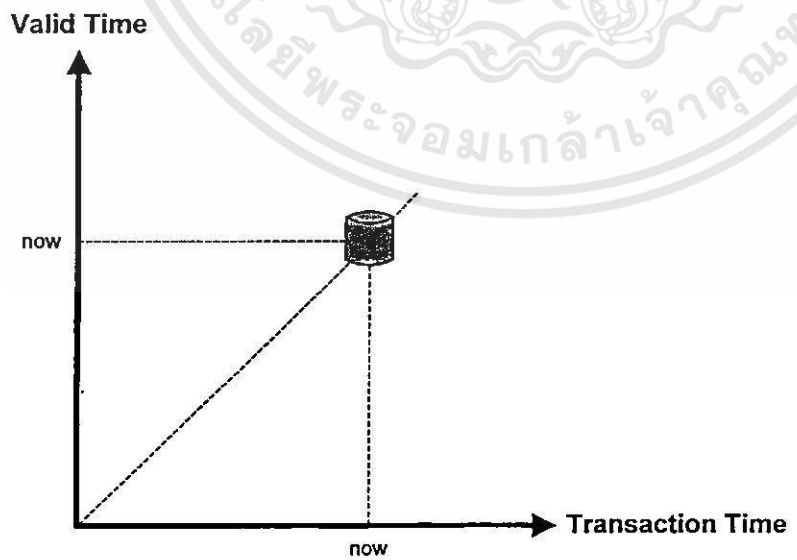
3.1 นิยามของฐานข้อมูลตามการสนับสนุนทางเวลา

ฐานข้อมูลซึ่งแบ่งประเภทตามการสนับสนุนข้อมูลทางเวลาที่เป็น valid time และ transaction time แบ่งออกได้เป็น 4 ชนิด [7] คือ ฐานข้อมูลแบบสแนปช็อต ฐานข้อมูลแบบโรลแบ็ค ฐานข้อมูลแบบอิงประวัติ และฐานข้อมูลแบบอิงเวลา ดังรายละเอียดต่อไปนี้

3.1.1 ฐานข้อมูลแบบสแนปช็อต (Snapshot Databases)

โมเดลเชิงสัมพันธ์แบบสแนปช็อต ฐานข้อมูลจะเก็บรวบรวมความสัมพันธ์แต่ละความสัมพันธ์ประกอบด้วยเซ็ตของแถว (tuple) และคอลัมน์ (attribute) ซึ่งปกติจะแทนด้วยตาราง 2 มิติ การเปลี่ยนแปลงใด ๆ ที่เกิดขึ้นจะกระทำภายใต้ตารางนี้

การเปลี่ยนแปลงและแก้ไขสถานะของฐานข้อมูล โดยใช้การปฏิบัติการ เช่น การเพิ่ม การลบ ซึ่งจะมีผลหลังจากทำการ commit แล้ว ในขบวนการจัดการดังกล่าวข้างต้น สถานะเดิมของฐานข้อมูลจะไม่สามารถรู้ได้ ถ้าปราศจากการสนับสนุนของระบบโปรแกรมประยุกต์จำนวนมาก ได้มีการปรับปรุงเพื่อจะให้สามารถสนับสนุนข้อมูลข่าวสารในลักษณะของ ad-hoc



รูปที่ 3.1 ฐานข้อมูลแบบสแนปช็อตบนแกนของเวลา

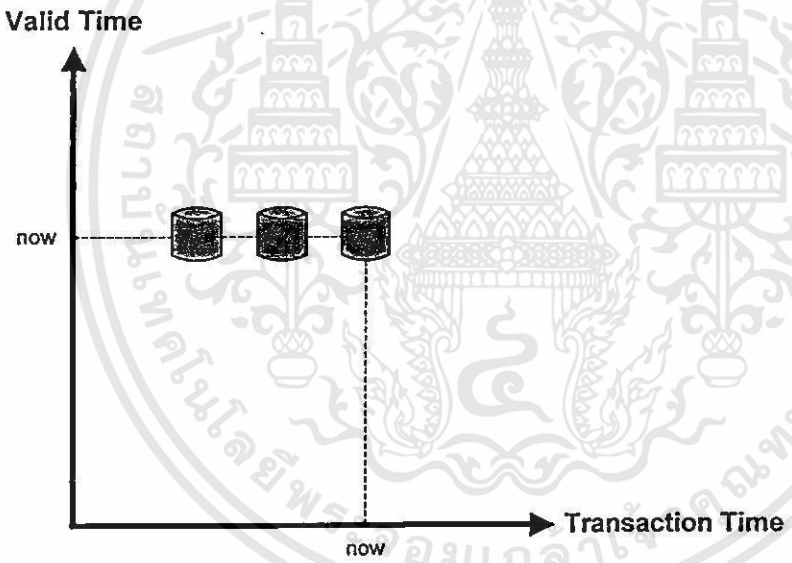
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 ฐานข้อมูลแบบโรลแบ็ค (Rollback Databases)

ฐานข้อมูลซึ่งสามารถสนับสนุนการโรลแบ็คได้ จะต้องมีการเก็บข้อมูลในอดีตและมีดัชนีตามเวลาของฐานข้อมูลแบบสแนปช็อต ความสัมพันธ์ดังกล่าวสามารถอธิบายแนวคิดได้เป็น 3 มิติ ซึ่ง transaction time เป็นเหมือนมิติที่ 3 และสามารถที่จะพิจารณาได้เป็นความสัมพันธ์แบบสแนปช็อตที่เรียงลำดับกัน ซึ่งมี transaction time เป็นดัชนี

โดยการเคลื่อนไปตามแกนเวลา ความสัมพันธ์แบบ snapshot ซึ่งบางเวลาในอดีตสามารถที่จะเรียกและสอบถามมาได้ การเปลี่ยนแปลงให้เป็นฐานข้อมูลแบบ rollback ทำได้โดยการสร้างสถานะล่าสุดของทุก ๆ ข้อมูล เมื่อข้อมูลเสร็จเรียบร้อย จะไม่มีการแก้ไขข้อมูล

มีข้อจำกัด 2 ประการคือ ประการแรก เป็นการเก็บพฤติกรรมของฐานข้อมูลแทนที่จะเก็บประวัติของข้อมูลจริง ๆ ประการที่ 2 คือไม่สามารถทำให้ได้ผลจริงเพราะว่าเกิดความซ้ำซ้อนในข้อมูลมากเกินไป กล่าวคือทับเปิลซึ่งไม่มีการเปลี่ยนแปลงระหว่างสถานะ แต่จะต้องมีข้อมูลซ้ำในสถานะใหม่



รูปที่ 3.2 ฐานข้อมูลแบบโรลแบ็คบนแกนของเวลา

3.1.3 ฐานข้อมูลแบบอิงประวัติ (Historical Database)

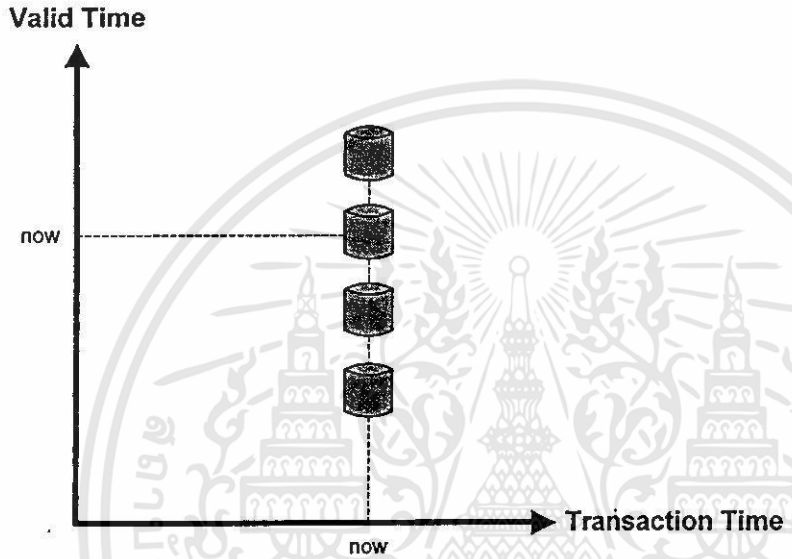
ฐานข้อมูลแบบอิงประวัติจะบันทึกประวัติข้อมูลของทุกความสัมพันธ์ และจะแทน valid time ซึ่งเป็นเวลาที่เก็บข่าวสารจริง ๆ เมื่อพบข้อผิดพลาดขึ้น ก็จะทำการแก้ไขฐานข้อมูลโดยไม่มีการเก็บเรคคอร์ดของการแก้ไขข้อผิดพลาดนั้น

แบบแผนนี้สามารถที่จะแทนได้ด้วย 3 มิติ โดยมิติที่ 3 เป็น valid time แบบนี้จะใกล้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เดียงกับความเป็นจริง แทนที่จะแก้ไขประวัติในกรณีที่เกิดการ rollback อย่างไรก็ตาม เนื่องจากการแก้ไขข้อมูลในความสัมพันธ์เชิงประวัติมีความจำเป็นที่ต้องกำหนด valid time ยิ่งไปกว่านั้นจะต้องมีการจัดการและสอบถามข้อมูล valid time

ข้อแตกต่างระหว่างฐานข้อมูลเชิงประวัติกับฐานข้อมูลแบบ rollback คือแบบอิงประวัติสามารถเปลี่ยนแปลงแก้ไขตามต้องการ ในขณะที่แบบ rollback อนุญาตให้มีการเพิ่มข้อมูลเท่านั้น



รูปที่ 3.3 ฐานข้อมูลแบบอิงประวัติบนแกนของเวลา

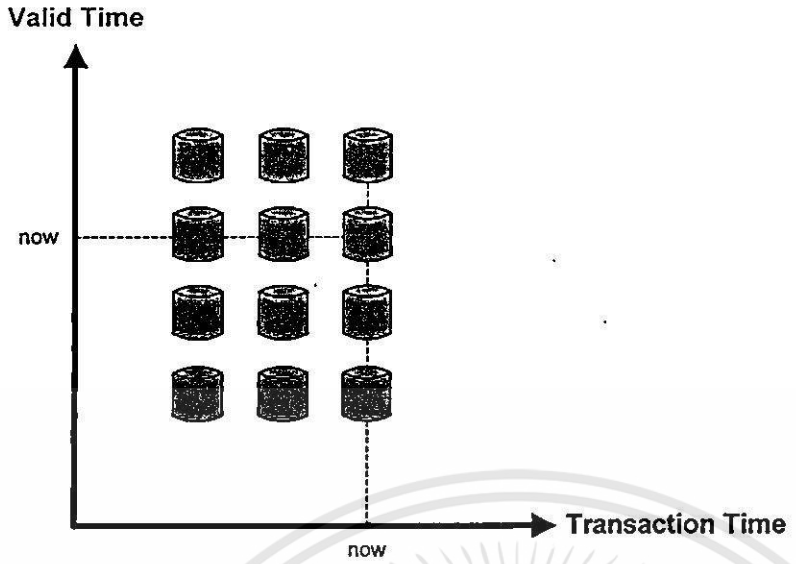
3.1.4 ฐานข้อมูลแบบอิงเวลา (Bitemporal Databases)

ฐานข้อมูลแบบนี้สามารถสนับสนุนทั้ง valid time และ transaction time ในการเก็บข้อมูลอิงเวลา ค่าของแถว (tuple) ถูกต้อง ณ.ขณะเวลาหนึ่งซึ่งเก็บข้อมูลเหตุการณ์ที่เปลี่ยนแปลงในอดีต ดังนั้นจึงมีแกนของเวลา 2 แกนเกี่ยวข้องด้วย ฐานข้อมูลนี้จึงแทนด้วยแกน 4 แกน

ความสัมพันธ์เชิงเวลานั้นสามารถพิจารณาได้ว่าเป็นเสมือนลำดับสถานะเชิงประวัติ แต่ความสัมพันธ์เป็นความสัมพันธ์เชิงประวัติที่สมบูรณ์ การทำ rollback กับความสัมพันธ์เชิงเวลาโดยเลือกจากสถานะของข้อมูลแบบเชิงประวัติที่แตกต่างกัน โดยวิธีการสอบถามข้อมูลเชิงประวัติความสัมพันธ์เชิงเวลาจะถูกเพิ่มเข้าไปเมื่อมีการสร้างสถานะของประวัติใหม่ในแต่ละ transaction ซึ่งต้องกำหนด valid times เช่นเดียวกับในฐานข้อมูลเชิงประวัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 3.4 ฐานข้อมูลแบบอิงเวลาบนแกนของเวลา

3.2 สถาปัตยกรรมของฐานข้อมูลเชิงเวลา

ในอดีตระบบจัดการฐานข้อมูลเชิงเวลาที่ถูกสร้างส่วนใหญ่ ใช้คุณสมบัติพื้นฐานร่วมกับระบบจัดการฐานข้อมูลแบบสแนปช็อต ได้มีการนำเสนอการออกแบบระบบจัดการฐานข้อมูลเชิงเวลาด้วยวิธีการขยายส่วนระบบจัดการฐานข้อมูลแบบสแนปช็อต 3 วิธี และประเมินความเหมาะสมของแต่ละวิธี

ความเหมาะสมของสถาปัตยกรรมแต่ละแบบที่ขยายเพิ่มจากระบบจัดการฐานข้อมูลแบบสแนปช็อต โดยการเพิ่มฟังก์ชันเชิงเวลาเพื่อให้บรรลุตามวัตถุประสงค์ของการออกแบบดังนี้

1. ฟังก์ชันเชิงเวลาต้องมีความสมบูรณ์ โดยที่ประเภทของเวลาที่ใช้ และตัวปฏิบัติการทางเวลาเป็นส่วนแรกที่ต้องพิจารณา
2. สามารถใช้งานกับระบบจัดการฐานข้อมูลเดิมได้
3. สามารถติดตั้งระบบได้ง่าย และสะดวก
4. สามารถใช้คุณสมบัติเดิมของระบบจัดการฐานข้อมูลแบบสแนปช็อตได้โดยไม่ต้องมีการแก้ไขปรับปรุงระบบ
5. มีความถูกต้องของข้อมูล (Data Integrity)
6. ประสิทธิภาพดี (Performance)
7. การใช้ทรัพยากรต้องน้อย
8. สามารถใช้ได้กับหลายแพลทฟอร์ม

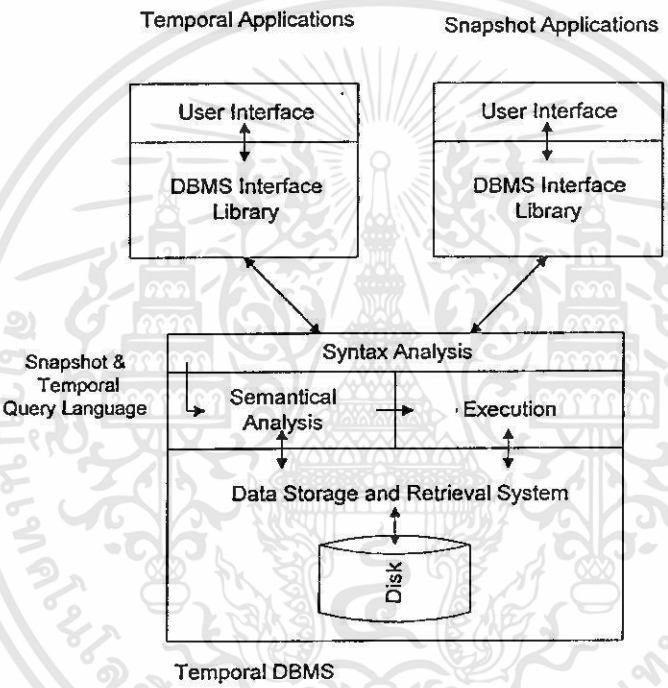
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

38529

สถาปัตยกรรมของฐานข้อมูลเชิงเวลาทั้ง 3 แบบ [8]ได้แก่ การรวมฟังก์ชันทางเวลาเข้ากับระบบจัดการฐานข้อมูลแบบสแนปช็อต กาใช้เซิร์ฟเวอร์เพื่อจัดการฐานข้อมูลเชิงเวลา และการสร้างส่วนสนับสนุนทางเวลาเข้ากับไคลเอนท์ของโปรแกรมประยุกต์โดยตรง

3.2.1 การรวมฟังก์ชันทางเวลาเข้ากับระบบจัดการฐานข้อมูลแบบสแนปช็อต (Integration of Temporal Functionality in the Snapshot DBMS)

สถาปัตยกรรมแบบนี้เป็นการรวมส่วนสนับสนุนทางเวลาเข้ากับระบบจัดการฐานข้อมูลแบบสแนปช็อตโดยตรง โดยการขยายส่วนหรือแก้ไขฐานข้อมูลดังแสดงในรูปที่ 3.5



รูปที่ 3.5 การรวมฟังก์ชันทางเวลาเข้ากับเคอร์เนลของระบบจัดการฐานข้อมูล

1. จะต้องมีการกำหนดชนิดของข้อมูลว่าเป็นจุดเวลา ช่วงเวลา หรือส่วนประกอบเชิงเวลา และโครงสร้างที่เหมาะสม เคอร์เนล (kernel) ของฐานข้อมูลจะต้องมีการเตรียมที่จะนำข้อมูลที่เกิดจากผู้ใช้ให้เก็บข้อมูลในรูปแบบของข้อมูลที่เก็บภายในระบบจัดการฐานข้อมูล
2. ส่วนวิเคราะห์ภาษา (semantical analysis module) เป็นส่วนที่พิจารณาว่าตัวปฏิบัติการนั้นใช้กับข้อมูลแบบไม่อิงเวลา หรือข้อมูลเชิงเวลา
3. ส่วนประมวลผลและสอบถามข้อมูล (execution module) จะต้องรวมตัวปฏิบัติการทางเวลาที่ใช้กับข้อมูลเพื่อรองรับการสอบถามข้อมูลเชิงเวลา

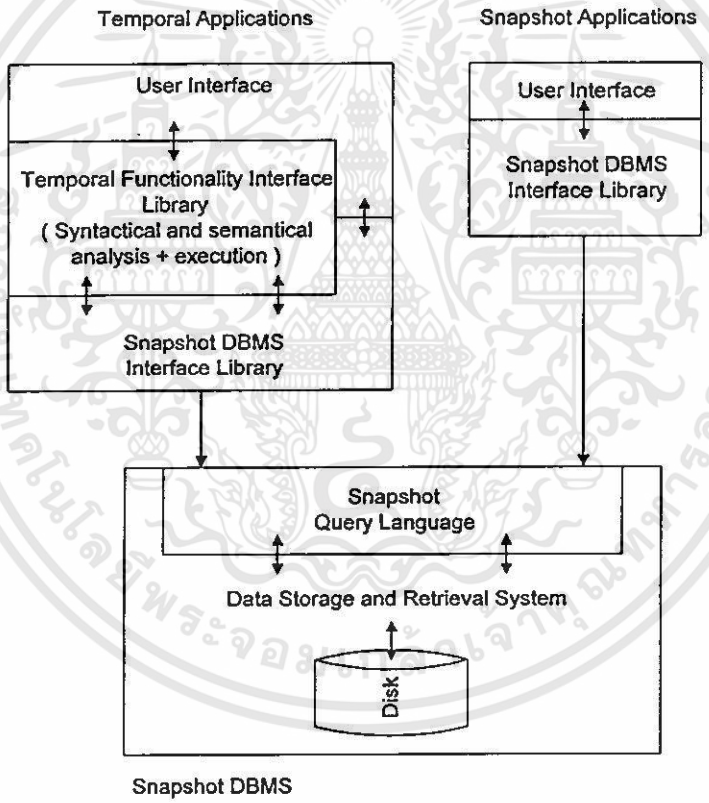
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ภาษากำหนดข้อมูล (Data Definition Language : DDL) และ ภาษาจัดการข้อมูล (Data Manipulation Language : DML) ต้องมีการแก้ไขเพื่อเพิ่มส่วนปฏิบัติการเชิงเวลา

การออกแบบของระบบที่พบว่าใช้วิธีการนี้คือ TempIS

3.2.2 การใช้เซิร์ฟเวอร์เพื่อจัดการฐานข้อมูลเชิงเวลา (The Temporal Server approach)

การใช้เซิร์ฟเวอร์เพื่อจัดการฐานข้อมูลแบบอิงเวลา เป็นวิธีที่ไม่ต้องแก้ไขระบบจัดการฐานข้อมูลแบบสแนปช็อต แต่ต้องมีเซิร์ฟเวอร์แบบอิงเวลา (temporal server) โดยที่เซิร์ฟเวอร์แบบอิงเวลา จะต้องอยู่ระหว่าง โปรแกรมประยุกต์แบบอิงเวลา (temporal application) และระบบจัดการฐานข้อมูลแบบสแนปช็อต ดังแสดงในรูปที่ 3.6



รูปที่ 3.6 การใช้เซิร์ฟเวอร์เพื่อจัดการฐานข้อมูลเชิงเวลา

โปรแกรมประยุกต์ซึ่งเข้าถึงข้อมูลแบบอิงเวลาโดยใช้ไลบารี่ของระบบจัดการฐานข้อมูลแบบอิงเวลา (temporal DBMS interface library) ซึ่งสามารถเรียก แก้ไข ข้อมูลบนเซิร์ฟเวอร์แบบอิงเวลาโดยตรง ซึ่งเซิร์ฟเวอร์รับผิดชอบงานต่างๆดังนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. วิเคราะห์การเรียกข้อมูลของโปรแกรมประยุกต์ว่าใช้ภาษาสอบถามเชิงโครงสร้างแบบอิงเวลา หรือภาษาสอบถามเชิงโครงสร้างปกติ
2. แยกข้อมูลแบบอิงเวลาซึ่ง kernel ของระบบจัดการฐานข้อมูลเดิมไม่สนับสนุนการใช้งาน แล้วทำการแปลงข้อมูลแบบอิงเวลาให้เป็นข้อมูลดิบที่ระบบสามารถจัดเก็บได้
3. แปลงตัวปฏิบัติการเชิงเวลาให้เป็นตัวปฏิบัติการแบบสแนปช็อต และทำการประมวลผลข้อมูลในระบบจัดการฐานข้อมูลแบบสแนปช็อต และส่งผลลัพท์กลับไปยังโปรแกรมประยุกต์ ซึ่งเซิร์ฟเวอร์แบบอิงเวลาใช้ ไลบารี่จากระบบจัดการฐานข้อมูลแบบสแนปช็อต

โปรแกรมประยุกต์ซึ่งต้องการข้อมูลปกติสามารถที่จะติดต่อกับระบบจัดการฐานข้อมูลแบบสแนปช็อต ได้โดยตรง ไม่ต้องผ่านเซิร์ฟเวอร์เพื่อจัดการฐานข้อมูลแบบอิงเวลา

3.2.3 การสร้างส่วนสนับสนุนทางเวลาเข้ากับไคล์เอนท์ของโปรแกรมประยุกต์โดยตรง (Building temporal support into the client application)

วิธีนี้สามารถทำได้โดยการสร้างตัวปฏิบัติการทางเวลาให้กับไคล์เอนท์ของโปรแกรมประยุกต์โดยตรง ดังแสดงในรูปที่ 3.7

โปรแกรมประยุกต์เชิงเวลาสอบถามข้อมูลเชิงเวลา โดยใช้ไลบารี่ของฟังก์ชันเชิงเวลาซึ่งถูกเชื่อมต่อเข้ากับการทำงานของโปรแกรมประยุกต์ ไลบารี่ของฟังก์ชันเชิงเวลามีหน้าที่เตรียมข้อมูลที่ใช้ในการแปลงข้อมูลเชิงเวลาเพื่อให้เป็นข้อมูลที่ได้รับการสนับสนุนจากระบบจัดการฐานข้อมูลแบบสแนปช็อต

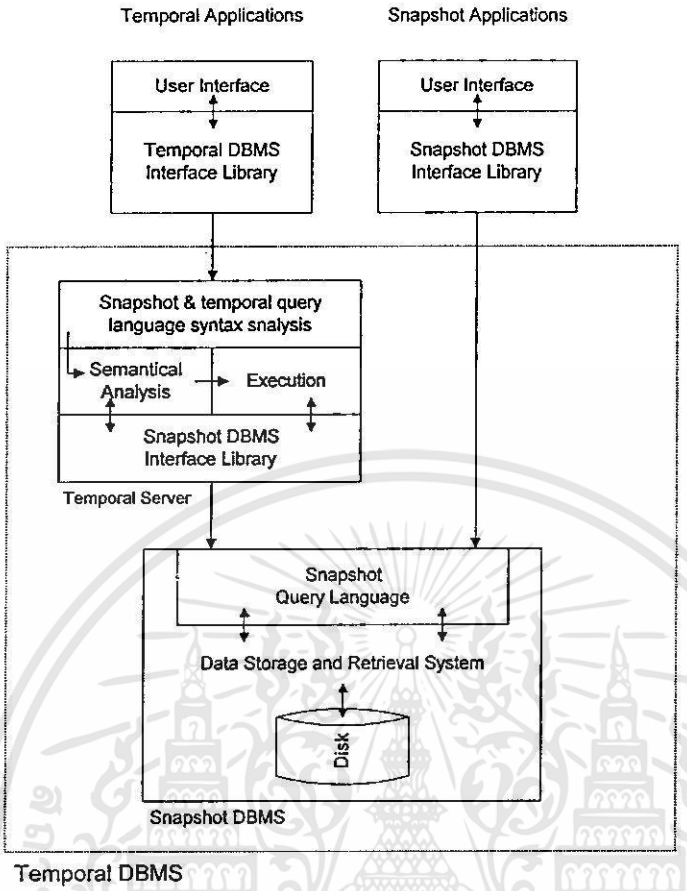
การออกแบบของระบบที่พบว่าใช้โดยวิธีนี้คือ Arcadia GCH-OSQL, Calanda, Chronolog, TempCase, TimeDB และ VT-SQL

3.3 การนำไปสู่ฐานข้อมูลเชิงเวลา

ฐานข้อมูลส่วนใหญ่ในปัจจุบันเป็นฐานข้อมูลเชิงสัมพันธ์ซึ่งมีการสนับสนุนทางเวลาและการจัดการข้อมูลเชิงเวลาน้อยมาก โปรแกรมประยุกต์บางส่วนจำเป็นต้องการใช้ฐานข้อมูลเชิงเวลา การพัฒนาระบบจัดการฐานข้อมูลเดิมไปเป็นระบบจัดการฐานข้อมูลเชิงเวลาและไม่ให้ส่งผลกระทบต่อฟังก์ชันการทำงานของโปรแกรมประยุกต์เดิมสามารถทำได้ด้วยคุณสมบัติของ upward compatibility [9]

การนำไปสู่ฐานข้อมูลเชิงเวลาโดยการเพิ่มโปรแกรมบางส่วนเข้าไปในระบบทำให้ระบบเดิมซึ่งไม่อิงเวลากับระบบงานใหม่ในแบบอิงเวลาสามารถเข้ากันได้อย่างกลมกลืน ซึ่งโปรแกรมประยุกต์ที่มีอยู่เดิมไม่ได้รับผลกระทบต่อการเปลี่ยนแปลงความสัมพันธ์ที่จะกลายเป็นความสัมพันธ์เชิงเวลาด้วยคุณสมบัติของ temporal upward compatibility

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 การใช้ไคล์เอนท์เพื่อจัดการฐานข้อมูลเชิงเวลา

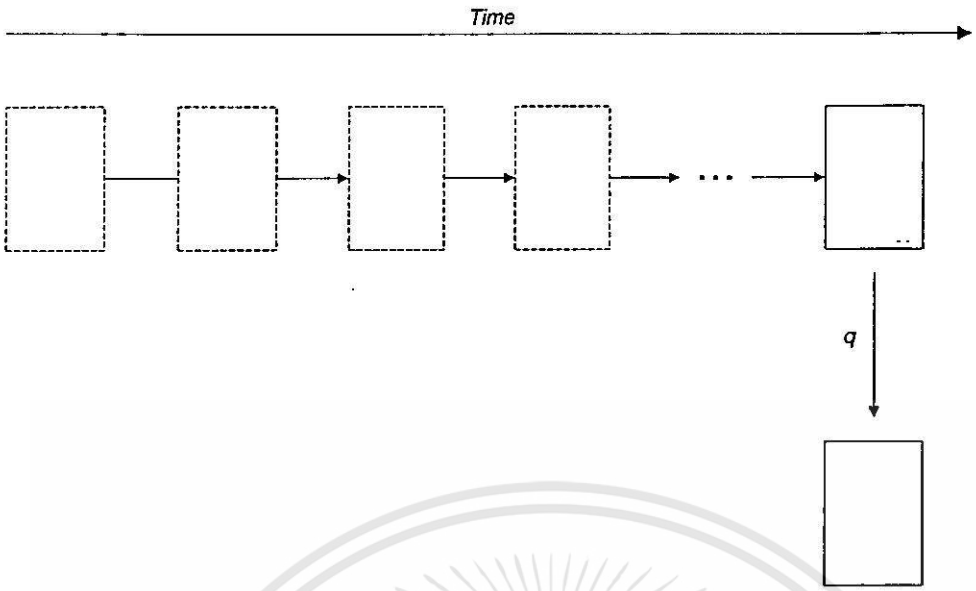
การจัดการข้อมูลและการสอบถามข้อมูลแบบไม่อิงเวลาแต่สามารถให้ผลลัพธ์เสมือนเป็นการสอบถามข้อมูลเชิงเวลาได้ โดยที่โมเดลข้อมูลจะต้องเป็นแบบเรียงลำดับ (segnence) แต่ก็มีข้อมูลอีกจำนวนมากที่โมเดลข้อมูลเป็นแบบไม่เรียงลำดับ จึงต้องมีการกำหนดฟังก์ชันและเพรดดิเคท (predicate) เพิ่มเพื่อการจัดการและสอบถามข้อมูล

3.3.1 Upward Compatibility

ในบางครั้งสิ่งที่สำคัญที่สุดในการย้ายข้อมูลให้ราบรื่นคือจะต้องแน่ใจว่าโปรแกรมประยุกต์ต่างๆ สามารถทำงานได้ในโมเดลข้อมูลใหม่

ในการสำรวจความสัมพันธ์ระหว่างข้อมูลแบบไม่อิงเวลากับข้อมูลเชิงเวลาและการสอบถามข้อมูล รูปที่ 3.8 ตารางข้อมูลแบบไม่อิงเวลาถูกแทนด้วยรูปสี่เหลี่ยม สถานะปัจจุบันของตารางคือรูปสี่เหลี่ยมที่มุมบนด้านขวา เมื่อมีการแก้ไขเปลี่ยนแปลงข้อมูลในตาราง สถานะเดิมจะเปลี่ยนไปจะคงสภาพอยู่เฉพาะสถานะปัจจุบัน ซึ่งสถานะเดิมถูกแทนด้วยรูปปะสี่เหลี่ยม ลูกศรที่ชี้ไปทางด้านขวาแสดงการเปลี่ยนแปลงแก้ไขตารางจากสถานะหนึ่งไปสู่อีกสถานะหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 การสอบถามข้อมูลแบบไม่อิงเวลา

ตัวอย่างที่ 3.1 บริษัทแห่งหนึ่งต้องการเก็บประวัติบุคลากร จึงทำการสร้างตาราง 2 ตาราง คือ ตารางพนักงาน และ ตารางเงินเดือน โดยที่พนักงานทุกคนมีเงินเดือน เมื่อพนักงานทุกคนได้รับเงินเดือนขึ้น 10% ถ้าตารางเงินเดือนไม่มีส่วนสนับสนุนทางเวลาแล้วเงินเดือนเดิมจะถูกบันทึกทับด้วยเงินเดือนใหม่

นิยาม 3.1 กำหนดให้ $M_1 = (DS_1, QL_1)$ และ $M_2 = (DS_2, QL_2)$ เป็นโมเดลของข้อมูล โมเดล M_1 จะเป็น Syntactically Upward Compatible กับโมเดล M_2

ถ้า $\forall db_2 \in DS_2 (db_2 \in DS_1)$ และ

$\forall S_2 \in QL_2 (S_2 \in QL_1)$

เมื่อมีการแปลงจากระบบหนึ่งไปสู่อีกระบบหนึ่ง ที่สำคัญอย่างมากคือโมเดลข้อมูลให้จะมีส่วนประกอบของโมเดลข้อมูลใหม่ ซึ่งโปรแกรมประยุกต์จะยังคงใช้การได้ในสภาพที่ถูกต้อง สำหรับการสอบถามข้อมูลด้วยประโยค s และฐานข้อมูลที่เกี่ยวข้อง db องค์ประกอบทั้ง 2 คือ QL และ DS ของโมเดลข้อมูล M = (DS,QL) กำหนดให้ ⟨⟨s (db)⟩⟩ M เป็นผลลัพธ์ที่ได้จากการกระทำของ s กับ db ในโมเดลข้อมูลของ M ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นิยาม 3.2 กำหนดให้ $M_1 = (DS_1, QL_1)$ และ $M_2 = (DS_2, QL_2)$ เป็นโมเดลข้อมูล โมเดล M_1 เป็น Upward Compatible กับโมเดล M_2

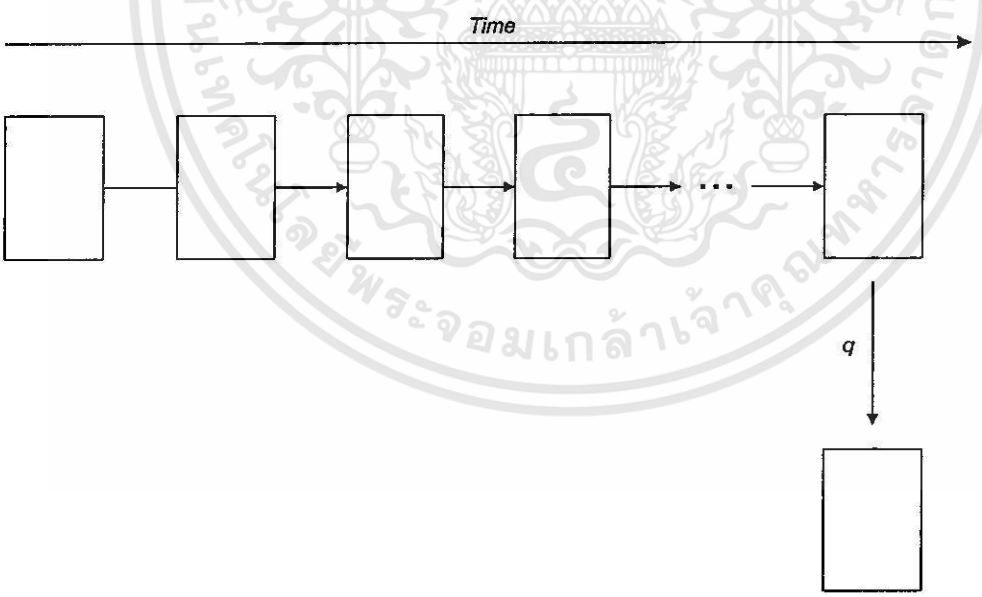
ถ้า M_1 เป็น Syntactically upward compatible กับ M_2

และ $\forall db_2 \in DS_2 (\forall s_2 \in QL_2 (\langle\langle s_2 (db_2) \rangle\rangle M_2 = \langle\langle s_2 (db_2) \rangle\rangle M_1))$

แนวคิดดังกล่าวซึ่งทำให้ระบบปัจจุบันที่ใช้โมเดล M_2 ถูกแปลงจากโมเดลข้อมูลเดิมคือ M_1 กับเงื่อนไขที่ว่าฐานข้อมูลที่มีอยู่เดิมทั้งหมดและการสอบถามข้อมูลเดิม สามารถใช้ได้ในระบบงานใหม่ เงื่อนไขที่ 2 คือ การสอบถามที่มีอยู่เดิมทั้งหมดสามารถประมวลผลได้ผลลัพธ์ในระบบใหม่เหมือนกับในระบบเก่า

3.3.2 Temporal Upward Compatibility

ถ้าโปรแกรมประยุกต์ต้องการส่วนสนับสนุนเชิงเวลาสำหรับข้อมูลในตาราง โดยเพิ่มค่า valid-time เข้าไปในตาราง ประวัติข้อมูลจะถูกเก็บรักษาไว้ การแก้ไขและเพิ่มเติมข้อมูล ณ ขณะเวลาใดๆจะถูกเก็บรักษาไว้เช่นกัน จากรูปที่ 3.9 สถานะทางซ้ายสุดเป็นสถานะปัจจุบัน การเพิ่ม ลบ แก้ไขข้อมูล แสดงด้วยเครื่องหมายลูกศร ผลลัพธ์ที่ได้คือตารางซึ่งแทนด้วยรูปสี่เหลี่ยมทางขวามือ



รูปที่ 3.9 การสอบถามข้อมูลเชิงเวลา

เนื่องจากเป็น temporal upward compatibility ของการสอบถามข้อมูล q ทำให้สามารถสอบถามข้อมูลเชิงเวลาได้ต้องการสนับสนุนจาก valid-time

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

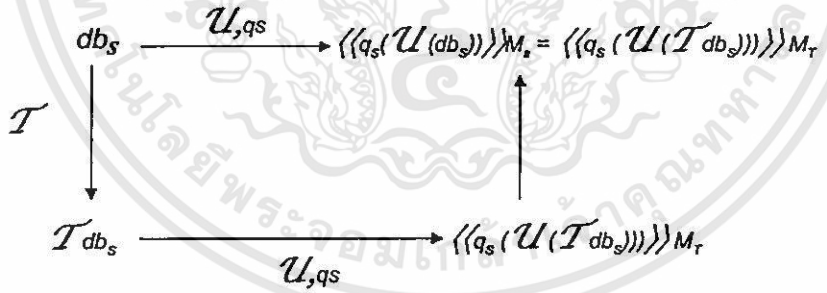
ตัวอย่างที่ 3.2 ทำตารางพนักงานและตารางเงินเดือนให้เป็นตารางเชิงเวลา นั่นหมายความว่าข้อมูลทั้งหมดจะเป็นจริงจากวันนี้เป็นต้นไป การสอบถามข้อมูลกับตารางดังกล่าวหลังจากทำการเพิ่ม valid-time ระบบทำงานได้สมบูรณ์ดังเดิม

นิยาม 3.3 กำหนดให้โมเดลข้อมูลเชิงเวลาและโมเดลข้อมูลแบบสแนปช็อตแทนด้วย $M_T = (DS_T , QL_T)$ และ $MS = (DS_S , QL_S)$ ตามลำดับ และกำหนดให้ $\mathcal{T}$ เป็นตัวปฏิบัติการซึ่งเปลี่ยนประเภทความสัมพันธ์แบบสแนปช็อตไปเป็นความสัมพันธ์เชิงเวลาด้วยการใช้แอททริบิวต์เดิม และกำหนดให้ $\mathcal{U} = u_1 , u_2 , ..., u_n \ (n \geq 0)$ แทนการแก้ไขเปลี่ยนแปลงข้อมูลแบบเรียงลำดับ ดังนั้นโมเดล M_T จะเป็น temporal upward compatible กับโมเดล M_S ถ้า

M_T เป็น upward compatible กับ M_S และ

$$\forall db_S \in DS_S \, (\forall \mathcal{U} (\forall qs \in QL_S ((\langle\langle q_S (\mathcal{U} (db_S))\rangle\rangle M_s = (\langle\langle q_S (\mathcal{U} (\mathcal{T} db_S))\rangle\rangle M_T))))$$

การเปลี่ยนแปลงแก้ไข ($\mathcal{U}$) การสอบถามข้อมูลและผลของการสอบถามข้อมูล (q_S) ในโมเดลข้อมูลเชิงเวลา จากรหัสโปรแกรมเดิมของโมเดลข้อมูลแบบสแนปช็อตไม่ส่งผลกระทบต่อการเปลี่ยนประเภทของความสัมพันธ์จากการแปลงความสัมพันธ์ ($\mathcal{T}$) การแก้ไขข้อมูลในความสัมพันธ์ใหม่ได้ผลลัพธ์เดียวกันกับการแก้ไขข้อมูลในความสัมพันธ์เดิม ดังแสดงในรูปที่ 3.10



รูปที่ 3.10 Temporal Upward Compatibility

3.3.3 การจัดการและสอบถามข้อมูลแบบเรียงลำดับ (Sequenced Queries and Modification)

การทำ Sequentiality มีวัตถุประสงค์หลักคือเพื่อป้องกันการลงทุนในการฝึกอบรมโปรแกรมเมอร์ และลดต้นทุนในการพัฒนาโปรแกรมประยุกต์ในโมเดลข้อมูลเชิงเวลา นั่นแสดงว่าภาษาสอบถามข้อมูลเชิงเวลา TQBE จะต้องสามารถใช้ภาษาสอบถามแบบไม่อิงเวลาได้ ทำให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมเมอร์สามารถเขียนการสอบถามข้อมูลเชิงเวลาได้โดยทันที ความผิดพลาดจึงน้อยลง และไม่ทำให้ประสิทธิภาพการทำงานต่ำลง

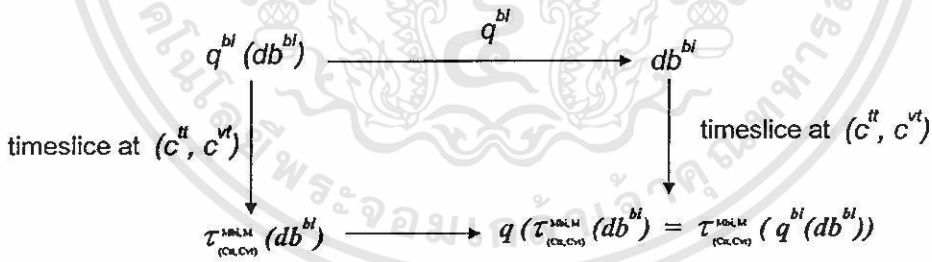
โดยธรรมชาติของ Sequentiality คือการทำ Snapshot reducibility กำหนดให้ r และ r^{bi} แทนความสัมพันธ์แบบสแนปช็อตและความสัมพันธ์เชิงเวลาตามลำดับ ในทำนองเดียวกัน db และ db^{bi} แทนเซ็ตของความสัมพันธ์ของความสัมพันธ์แบบสแนปช็อตและเซ็ตของความสัมพันธ์เชิงเวลาตามลำดับ

กำหนดให้ตัวปฏิบัติการไทม์สไลด์ (timeslice operator) $\tau^{Mbi,M}_{(Ct,Cvt)}$ ของความสัมพันธ์ r^{bi} และในโมเดลข้อมูล M^{bi} และค่าเชิงเวลา (C^{tt} , C^{vt}) โดยคืนค่าความสัมพันธ์แบบสแนปช็อต r ในโมเดลข้อมูล M ประกอบด้วยทัปเปิลทั้งหมด เวลาปัจจุบัน C^{tt} และเป็นจริงที่เวลา C^{vt} อีกด้านหนึ่ง ความสัมพันธ์ r ประกอบด้วยทับเปิลทั้งหมดของความสัมพันธ์เชิงเวลา r^{bi} ซึ่งเกี่ยวข้องกับเวลา (C^{tt} , C^{vt}) แต่ไม่มีค่า valid-time และ transaction time

นิยาม 3.4 กำหนดให้ M = (DS,QL) เป็นโมเดลเชิงสัมพันธ์และ M^{bi} = (DS^{bi} , QL^{bi}) เป็นโมเดลข้อมูลเชิงเวลา โมเดลข้อมูล M^{bi} เป็น Snapshot reducible กับโมเดลข้อมูล M ถ้า

$$\forall q \in QL\,(\exists q^{bi} \in QL^{bi} (\forall\, db^{bi} \in DS^{bi} (\forall c(\tau^{Mbi,M}_{(Ct,Cvt)}\,(q^{bi}\,(db^{bi})) = q\,(\tau^{Mbi,M}_{(Ct,Cvt)}\,(db^{bi}))))))$$

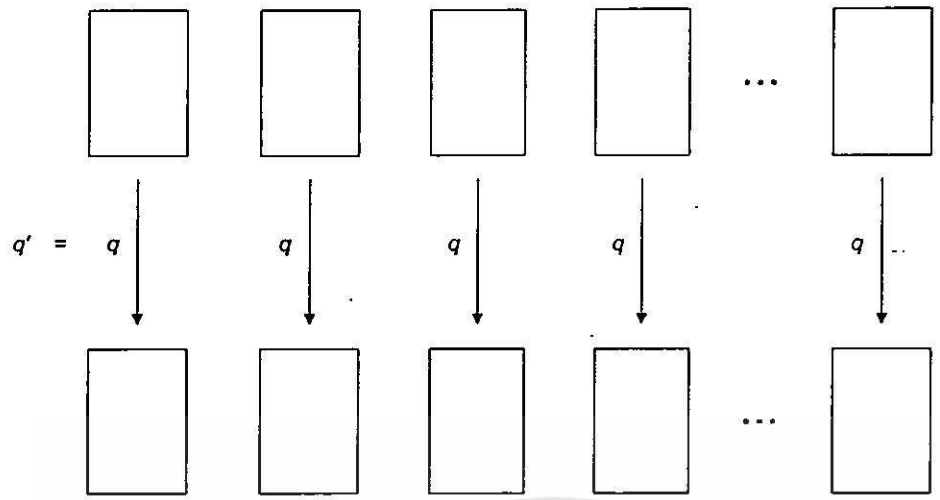
สำหรับการสอบถามข้อมูล q ในโมเดลแบบสแนปช็อต จะต้องมีการสอบถามข้อมูล q^{bi} ในโมเดลฌชิงเวลา ดังนั้นสำหรับทุกๆ db^{bi} และทุกๆค่าของเวลาดังแสดงในรูปที่ 3.11



รูปที่ 3.11 Snapshot Reducibility ของ q^{bi} ซึ่งสัมพันธ์กับ q

สังเกตจากรูปที่ 3.12 แสดง Snapshot Reducibility การสอบถามข้อมูลเชิงเวลา q' เป็น snapshot reducibility กับการสอบถามข้อมูลแบบสแนปช็อต q การสอบถาม q' ประยุกต์ใช้กับความสัมพันธ์เชิงเวลา (ลำดับสถานะที่เป็นรูปสี่เหลี่ยมด้านบน) และผลลัพธ์ในความสัมพันธ์เชิงเวลา ซึ่งแทนด้วยลำดับสถานะรูปสี่เหลี่ยมด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



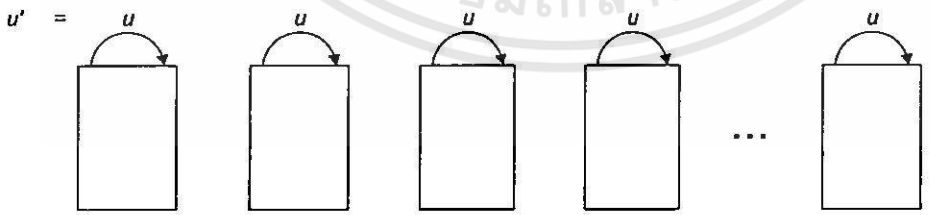
รูปที่ 3.12 Snapshot Reducible ของการสอบถามเชิงเวลา q' บนความสัมพันธ์เชิงเวลา

พิจารณาการสอบถามเชิงเวลา q' ซึ่งลดรูปเป็นการสอบถามแบบสแนปช็อต จากรูปที่ 3.12 แสดงผลจากการสอบถามข้อมูล q' ถูกกำหนดเสมือนเป็นการรวมความสัมพันธ์เชิงเวลาของผลที่ได้จากการหาค่าการสอบถาม q เรียงลำดับสถานะของสแนปช็อตพิจารณา 2 ประการ

ประการแรก สถานะซึ่งเรียงลำดับดังรูปที่ 3.12 เราสามารถใช้นิยาม sequence สำหรับการสอบถาม ซึ่งเป็นไปตามความต้องการแบบ reducibility

ประการที่สองความต้องการในการทำ reducibility เป็นการบังคับให้กำหนดการสอบถามแบบเรียงลำดับ

การสอบถามแบบเรียงลำดับ (sequence) ทำได้กับทั้งเวลาเป็นจุดและช่วงเวลา การลดรูปยังสามารถประยุกต์ใช้กับการแก้ไขข้อมูลดังแสดงในรูปที่ 3.13 การแก้ไขข้อมูลเชิงเวลา ซึ่งจะไปทำลายสถานะการแก้ไขข้อมูลดังแสดงด้วยลูกศรโค้ง



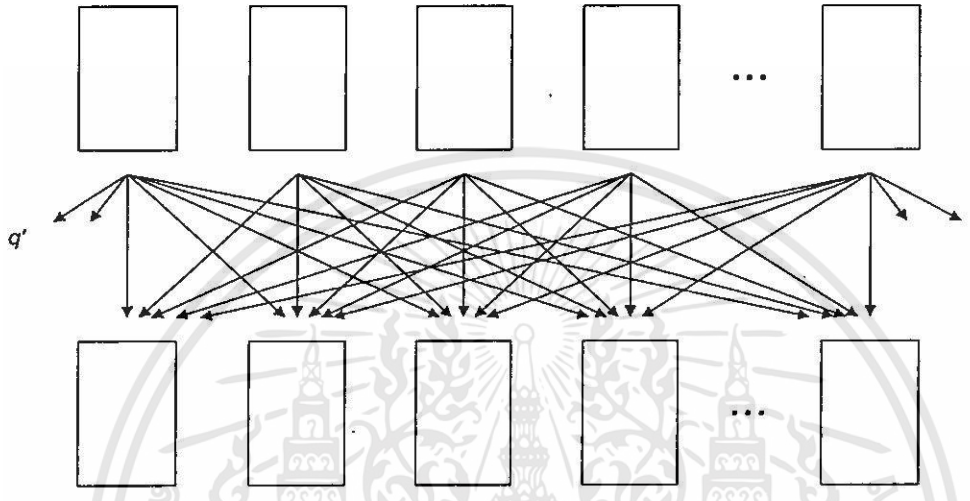
รูปที่ 3.13 การเปลี่ยนแปลงแก้ไขความสัมพันธ์เชิงเวลา

3.3.4 การจัดการและสอบถามข้อมูลแบบไม่เรียงลำดับ (Non-Sequenced Queries and Modification)

ในการสอบถามข้อมูลสแนปช็อตแบบลดรูป ข้อมูลในสถานะต่างๆของผลลัพธ์ที่ได้จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์เชิงเวลา ได้มาจาก์ข้อมูลสถานะต่างๆในเวลาเดียวกันของความสัมพันธ์ แต่ก็มีการสอบถามข้อมูลอีกมากที่ไม่สามารถสอบถามข้อมูลแบบเรียงลำดับได้ ดังนั้นขณะที่ยังไม่มีการกำหนดความต้องการเพิ่มขึ้น ภาษาสอบถามเชิงเวลาควรจะให้มีการสอบถามแบบไม่เรียงลำดับได้ (non-sequenced) การสอบถามดังกล่าวดังแสดงในรูปที่ 3.14 ซึ่งแต่ละสถานะที่เป็นไปได้ทั้งหมดของความสัมพันธ์เดิม

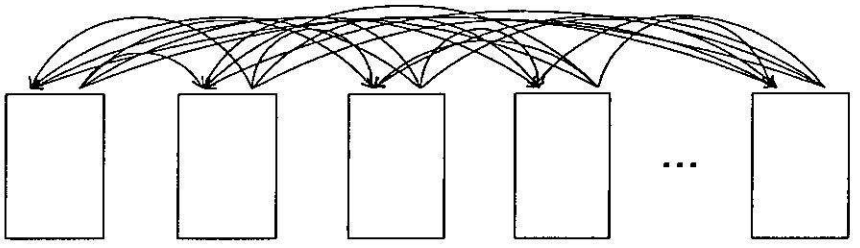


รูปที่ 3.14 การสอบถามข้อมูลเชิงเวลาแบบไม่เรียงลำดับ q′ บนความสัมพันธ์เชิงเวลา

จากรูปที่ 3.14 ความสัมพันธ์เชิงเวลา 2 ความสัมพันธ์ ความสัมพันธ์แรกประกอบด้วยสถานะต่างๆซึ่งแทนด้วยรูปสี่เหลี่ยมด้านบนรูป อีกความสัมพันธ์เป็นผลลัพธ์ที่ได้จากการสอบถามข้อมูล ประกอบด้วยสถานะใดๆแทนด้วยรูปสี่เหลี่ยมด้านล่าง การสอบถามq/ แสดงความเป็นไปได้ทั้งหมดจากการคำนวณ ซึ่งการใช้ข้อมูลแสดงด้วยลูกศรชี้ลง เมื่อไรก็ตามที่การคำนวณของสถานะหนึ่งๆของผลลัพธ์จากความสัมพันธ์จะต้องใช้ข้อมูลจากหลายๆสถานะที่เวลาต่างกัน การสอบถามดังกล่าวจะเป็นการสอบถามแบบไม่เรียงลำดับ (nonsequenced)

แนวคิดของการสอบถามข้อมูลแบบไม่เรียงลำดับนำไปใช้กับการจัดการข้อมูลเช่นการเพิ่ม ลบ แก้ไขข้อมูลแบบไม่เรียงลำดับด้วย แต่ละสถานะของความสัมพันธ์เชิงเวลาที่ถูกเปลี่ยนแปลงแก้ไข โดยใช้ข้อมูลจากสถานะทั้งหมดที่เป็นไปได้ของความสัมพันธ์ก่อนการเปลี่ยนแปลงแก้ไขดังแสดงในรูปที่ 3.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 การเปลี่ยนแปลงแก้ไขข้อมูลเชิงเวลาแบบไม่เรียงลำดับ α' บนความสัมพันธ์เชิงเวลา

การจัดการและการสอบถามข้อมูลเชิงเวลาแบบไม่เรียงลำดับนั้นซับซ้อนกว่าการสอบถามข้อมูลสแนปช็อตแบบลดรูป เนื่องจากได้รับการสนับสนุนจากฟังก์ชันและเพรดดิเคท (predicate) น้อยกว่า ดังนั้นจะต้องจัดการกับความสัมพันธ์เชิงเวลาและคำนวณเวลาเอง ซึ่งจำเป็นจะต้องมีการสร้างส่วนเพิ่มเติมในภาษาสอบถาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

TQBE (Temporal Query-By-Example)

การสอบถามข้อมูลแบบใช้ตัวอย่าง (Query-By-Example : QBE) [10] ถูกพัฒนาขึ้นโดยบริษัท IBM ที่ T.J.watson Research Center เมื่อต้นปี ค.ศ.1970 เดิมทีเป็นทั้งภาษาจัดการข้อมูลและระบบฐานข้อมูล ซึ่งรวมภาษาดังกล่าวไว้ในระบบด้วย แต่ระบบดังกล่าวไม่ได้นำมาใช้ ส่วนภาษาจัดการข้อมูลถูกสร้างให้เป็นส่วนหนึ่งของ QMF (Query Management Facility) ซึ่งคุณสมบัติเด่นที่แตกต่างอย่างเห็นได้ชัด ของ QBE กับภาษาอื่นๆมี 2 ประการคือ

1. QBE มีรูปแบบไวยากรณ์เป็น 2 มิติ ในขณะที่ภาษาสอบถามและภาษาโปรแกรม (programming language) อื่นๆ เป็นแบบมิติเดียว เช่นภาษาสอบถามเชิงโครงสร้าง (SQL) สามารถสอบถามข้อมูลโดยใช้ประโยคคำสั่งเพียงบรรทัดเดียว ซึ่งบางครั้งจะเป็นประโยคที่ยาวมาก ส่วนภาษาแบบ 2 มิติ ต้องการประโยคที่จะแสดงเป็นชุดคำสั่งแบบ 2 มิติ

2. การสอบถามแบบ QBE ถูกแสดง "โดยตัวอย่าง" (by example) แทนที่จะใช้ชุดคำสั่ง (procedure) เพื่อรับค่าคำถามที่ต้องการ ระบบจะประมวลผลหาคำตอบสำหรับคำถามนั้น ทำให้ง่ายและสะดวกต่อการใช้งาน

การสอบถามข้อมูลเชิงเวลาแบบใช้ตัวอย่าง (Temporal Query-By-Example : TQBE) เป็นภาษาสอบถามข้อมูลและจัดการข้อมูล ที่ถูกออกแบบมาสำหรับระบบจัดการฐานข้อมูลเชิงเวลา โดยมีรูปแบบโครงสร้างเป็นแบบกราฟฟิก และ ใช้แนวคิดการสอบถามข้อมูลแบบใช้ตัวอย่าง (QBE) เป็นการผสมผสานกันระหว่าง QBE และกฎบังคับความถูกต้องทางเวลา เพื่อตอบสนองกับการใช้เวลาและการจัดการเวลา ในฐานข้อมูลเชิงเวลาให้เป็นไปได้อย่างถูกต้อง ดังจะได้กล่าวในรายละเอียดต่อไปนี้

4.1 โครงสร้างพื้นฐาน (Basic Structure)

การสอบถามข้อมูลใน QBE ถูกแสดงโดยใช้ตารางโครงร่าง (Skeleton Tables) ตารางเหล่านี้แสดงความสัมพันธ์อย่างมีแบบแผน ดังแสดงในรูปที่ 4.1 ผู้ใช้เลือกตารางโครงร่างเพื่อการสอบถามข้อมูลโดยการใส่แถวตัวอย่าง (example rows) เข้าไปในตารางโครงร่าง แถวตัวอย่างจะประกอบด้วยค่าคงที่ และส่วนประกอบของตัวอย่างซึ่งแท้จริงแล้วคือค่าตัวแปรโดเมน (domain variables) เพื่อหลีกเลี่ยงความสับสนในตัวแปรโดเมนของ QBE จะใช้เครื่องหมายขีดใต้ (underscore) หน้าตัวแปร เช่น _x ส่วนค่าคงที่แสดงไว้โดยไม่มีข้อจำกัด ซึ่งแตกต่างจากภาษาอื่นๆ ซึ่งค่าคงที่จะต้องถูกอ้าง และตัวแปรปรากฏโดยปราศจากข้อจำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table Name	Field 1	Field 2	...	...	Field n

รูปที่ 4.1 ตารางโครงร่าง (Skeleton table)

4.2 การสอบถามอย่างง่าย (Simple Queries)

การสอบถามข้อมูลโดยใช้ตารางโครงร่างเพื่อแสดงผลลัพธ์ที่ต้องการดังตัวอย่าง ตัวอย่างที่ 4.1 ต้องการหาลูกค้าซึ่งมีบัญชีธนาคารอยู่ที่สาขา Perryridge โดยใช้ตารางโครงร่างของความสัมพันธ์ Deposit แล้วใส่ข้อมูลดังต่อไปนี้

Deposit	branch-name	account-number	customer-name	balance
	Perryridge		P._x	

การสอบถามดังกล่าวระบบจะค้นหาทับเปิลในตาราง Deposit ซึ่งมีค่าของแอททริบิวต์ branch-name เป็น Perryridge โดยค่าแอททริบิวต์ customer-name ถูกแทนด้วยตัวแปร x และค่าตัวแปร x ถูกแสดงออกมาโดยคำสั่ง " P." ซึ่งอยู่หน้าตัวแปร _x อีกทั้งสามารถเขียนให้อยู่ในรูปของ domain relational calculus ได้ดังนี้

$$\{ < x > \mid \exists\ b,l,a\ (< b,l,x,a > \in \text{Deposit} \wedge b = \text{"Perryridge"})\}$$

ถ้าตัวแปรในตารางโครงร่างมีเพียงตัวแปรเดียว สามารถที่จะละเว้นการใส่ค่าตัวแปรได้ ทำให้ตำแหน่งที่ตัวแปรเคยอยู่ว่างลงดังแสดงในตาราง

Deposit	branch-name	account-number	customer-name	balance
	Perryridge		P.	

มีคำสั่งที่ใช้ในการกำจัดการซ้ำของแถวข้อมูลโดยอัตโนมัติด้วยการใช้คำสั่ง "ALL" ใส่ไว้หลังคำสั่ง P. ดังแสดง

Deposit	branch-name	account-number	customer-name	balance
	Perryridge		P.ALL	

ในการแสดงข้อมูลของตารางความสัมพันธ์ทั้งหมดในทุกๆแอททริบิวต์สามารถใช้คำสั่งย่อ โดยการวางคำสั่ง P. ที่คอลัมน์ที่เป็นชื่อของตารางดังแสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Deposit	branch-name	account-number	customer-name	balance
P.				

TQBE อนุญาติให้มีการสอบถามข้อมูลที่เกี่ยวกับการเปรียบเทียบทางคณิตศาสตร์ ตัวปฏิบัติการทางคณิตศาสตร์ซึ่งสามารถใช้ใน TQBE ได้แก่ =, <, ≤, >, ≥ และ ¬ ประโยคคณิตศาสตร์จะต้องอยู่ทางด้านขวาของตัวเปรียบเทียบและเว้นที่ว่างทางด้านซ้ายไว้ vu อีกทั้งสามารถใช้ตัวแปรและค่าคงที่ในประโยคคณิตศาสตร์ได้

ตัวอย่างที่ 4.2 ให้หาเลขที่บัญชีของบัญชีธนาคารทั้งหมดซึ่งมียอดเงินมากกว่า 1,200 สามารถเขียนให้อยู่ในรูปของตารางโครงร่างได้ดังนี้

Deposit	branch-name	account-number	customer-name	balance
		P.		>1200

ตัวอย่างที่ 4.3 ให้หาชื่อของสาขาธนาคารทั้งหมดซึ่งไม่อยู่ที่ Brooklyn สามารถเขียนให้อยู่ในรูปของตารางโครงร่างได้ดังนี้

Branch	branch-name	Assets	branch-city
	P.		¬ Brooklyn

ใน TQBE สามารถระบุค่ามากกว่า 1 ค่าให้กับแอททริบิวต์ได้

ตัวอย่างที่ 4.4 ให้หาลูกค้าทั้งหมดที่มีบัญชีอยู่ทั้ง 2 สาขาคือที่สาขา Perryridge และสาขา Redwood จะได้ตารางโครงร่างดังนี้

Deposit	branch-name	account-number	customer-name	balance
	Perryridge		P._x	
	Redwood		_x	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสอบถามในตารางจากตัวอย่างที่ 4.4 ระบบจะต้องหาค่าในแอททริบิวต์ของ customer-name ซึ่งค่าแอททริบิวต์ branch-name ในทับเปิลค่าหนึ่งจะต้องเป็น "Perryridge" ส่วนอีกค่าหนึ่งจะต้องเป็น "Redwood" ซึ่งสามารถเขียนให้อยู่ในรูป domain relational calculus ได้ดังนี้

$$\{ < x > \mid \exists\ b,a,n\ (<b,a,x,n> \in \text{Deposit} \wedge b = \text{"perryridge"}) \wedge \exists\ b,a,n\ (<b,a,x,n> \in \text{Deposit} \wedge b = \text{"Redwood"}) \}$$

ตัวอย่างที่ 4.5 ให้หาลูกค้าซึ่งมีบัญชีอยู่ที่สาขา Perryridge สาขา Redwood หรือทั้ง 2 สาขา ตารางโครงร่างจะเป็นดังนี้

Deposit	branch-name	account-number	Customer-name	balance
	Perryridge		P._x	
	Redwood		P._y	

และเขียนในรูป domain relation calculus ได้ดังนี้

$$\{ < x > \mid \exists\ b,a,n\ (< b,a,x,n > \in \text{Deposit} \wedge b = \text{"Perryridge"}) \vee \exists\ b,a,n\ (< b,a,x,n > \in \text{Deposit} \wedge b = \text{"Redwood"}) \}$$

ตัวอย่างที่ 4.6 ให้หาลูกค้าทั้งหมดซึ่งมีบัญชีอยู่ในสาขาเดียวกันกับที่ Jones มีบัญชีอยู่ สามารถเขียนเป็นตารางโครงร่างได้ดังนี้

Deposit	branch-name	Account-number	Customer-name	balance
	_x		Jones	
	_x		P._y	

4.3 การสอบถามข้อมูลแบบมีลำดับ (Queries with Ordering)

ผู้ใช้สามารถควบคุมลำดับการแสดงผลทับเปิลของความสัมพันธ์ได้โดยการใส่คำสั่ง AO เพื่อเรียงลำดับจากน้อยไปหามาก (ascending order) หรือคำสั่ง DO ในการเรียงลำดับจากมากไปหาน้อย (descending order) ในคอลัมน์ที่ต้องการ

ตัวอย่างที่ 4.7 ต้องการเรียงลำดับชื่อลูกค้าจากน้อยไปหามากซึ่งมีสาขาอยู่ที่ Perryridge จะได้ตารางโครงร่างดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Deposit	Branch-name	account-number	customer-name	balance
	Perryridge		P.AO.	

TQBE มีกลไกการเรียงลำดับและแสดงผลข้อมูลคราวละหลายๆคอลัมน์ โดยใช้เลขจำนวนเต็มใส่ลงในวงเล็บแสดงลำดับความสำคัญก่อนหลัง

ตัวอย่างที่ 4.8 ให้หาลูกค้าทุกคนของธนาคารโดยเรียงลำดับตามชื่อลูกค้าจากน้อยไปมาก แล้วเรียงลำดับยอดคงเหลือจากมากไปน้อย

จะได้ตารางโครงร่างดังนี้

Deposit	branch-name	account-number	customer-name	balance
			P.AO(1)	P.DO(2)

คำสั่ง P.AO(1) กำหนดว่าให้ทำการเรียงลำดับลูกค้าก่อน แล้วจึงเรียงลำดับยอดคงเหลือจากคำสั่ง P.DO(2) เป็นคำสั่งต่อไป

4.4 การสอบถามหลายความสัมพันธ์ (Queries on Several Relation)

เราสามารถที่จะสอบถามข้อมูลพร้อมกันทีละหลายๆความสัมพันธ์ได้ ซึ่งคล้ายคลึงกับการทำคาร์ทีเซียนโปรดักซ์ (cartesian product) หรือการจอย (natural join) ในพีชคณิตสัมพันธ์ การเชื่อมโยงระหว่างความสัมพันธ์ต่างๆ โดยใช้ตัวแปรซึ่งมีค่าเดียวกันบนแอททริบิวต์ในแต่ละความสัมพันธ์

ตัวอย่างที่ 4.9 ต้องการหาชื่อและเมืองของลูกค้าทุกคนซึ่งกู้เงินจากสาขา Perryridge สามารถแสดงได้ดังนี้

Borrow	branch-name	account-number	customer-name	balance
	Perryridge		_x	

Customer	customer-name	street	customer-city
	P._x		P._y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบจะหาทับเปิลในตาราง Borrow ซึ่งมีชื่อสาขาเป็น Perryridge ในแต่ละทับเปิลดังกล่าวระบบจะค้นหาทับเปิลในตาราง Customer ด้วยค่าในแอททริบิวต์ customer-name เดียวกัน แล้วจึงแสดงค่าในแอททริบิวต์ customer-name และ customer-city

ตัวอย่างที่ 4.10 ให้หาชื่อลูกค้าทุกคนซึ่งมีบัญชีอยู่ที่สาขา Perryridge แต่ไม่ได้กู้เงินจากสาขานั้น การสอบถามข้อมูลซึ่งเกี่ยวข้องกับประโยคปฏิเสธใน TQBE ทำได้โดยการใส่เครื่องหมาย not (¬) ข้างใต้ชื่อตารางความสัมพันธ์

Deposit	branch-name	account-number	customer-name	balance
	Perryridge		P._x	

Borrow	branch-name	Loan-number	customer-name	balance
¬	Perryridge		_x	

QBE หาค่า x ทั้งหมดตามตารางโครงร่างซึ่ง

1. มีทับเปิลในตาราง Deposit ซึ่งมี branch-name คือ "Perryridge" และ customer-name เป็นค่าตัวแปรโดเมนของ x
2. ต้องไม่มีทับเปิลในตาราง Borrow ซึ่งมี branch-name เป็น "Perryridge" และ customer-name เป็นค่าตัวแปรโดเมนของ x เช่นเดียวกัน

4.5 ส่วนกำหนดเงื่อนไข (Condition Box)

การกำหนดกฎบังคับต่างๆทั้งหมดบนตัวแปรโดเมนภายในตารางโครงร่างอาจทำได้ไม่สะดวก TQBE จึงได้เพิ่มคุณสมบัติที่เรียกว่าส่วนกำหนดเงื่อนไข (condition box) เพื่อกำหนดกฎบังคับต่างๆโดยเฉพาะ

ตัวอย่างที่ 4.11 ต้องการหา account-number ซึ่งมี balance มีค่าอยู่ระหว่าง 1,300 ถึง 1,500 สามารถเขียนเป็นตารางโครงร่างพร้อมส่วนเงื่อนไขได้ดังนี้

Deposit	Branch-name	account-number	customer-name	balance
		P.		_x

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Conditions
_x ≥ 1300
_x ≤ 1500

เราสามารถใช้ตัวปฏิบัติการทางตรรกะเช่น and or not ซึ่งแทนด้วยสัญลักษณ์ & | และ ¬ ตามลำดับ

ตัวอย่างที่ 4.12 ต้องการหา account-number ซึ่งมี balance มีค่าอยู่ระหว่าง 1,300 ถึง 2,000 ยกเว้นค่า 1,500 สามารถเขียนได้ดังนี้

Deposit	branch-name	account-number	customer-name	balance
		P.		_x

Conditions
_x = (≥ 1300 and ≤ 2000 and ¬ 1500)

4.6 ส่วนกำหนดช่วงเวลา (Time Range Box)

ส่วนกำหนดช่วงเวลา ถูกใช้เพื่อการกำหนดความสัมพันธ์ของช่วงเวลาของตัวแปร โดยใช้ตัวปฏิบัติทางคณิตศาสตร์ หรือตัวปฏิบัติการทางเวลา 13 ตัว ของ Allen คือ before, after, meets, met-by, overlaps, overlapped-by, during, includes, starts, started-by, finishes, finished-by และ equal

ตัวอย่างที่ 4.13 ต้องการข้อมูลช่วงเวลาตั้งแต่วันที่ 26/2/1998 จนถึงวันที่ 13/3/1998 สามารถเขียนเป็นส่วนช่วงเวลาได้ดังนี้

Time Range
1998-02-25 < DATE < 1998-03-14

4.7 ส่วนกำหนดหน่วยย่อยเวลา (Time Granularity Box)

ส่วนกำหนดหน่วยย่อยเวลา ใช้เพื่อกำหนดหน่วยย่อยในการระบุเวลา (timestamping) เช่น day , month , year

ตัวอย่างที่ 4.14 ต้องการกำหนดหน่วยเวลาย่อยเป็น วัน

สามารถเขียนเป็นส่วนหน่วยเวลาได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Time Granularity
DAY

4.8 ส่วนกำหนดประเภทเวลา (Time State Box)

ส่วนกำหนดประเภทเวลา ใช้ในการระบุประเภทเวลา เพื่อการสอบถามข้อมูล โดยระบุว่าเป็นการสอบถามข้อมูล ณ เวลาปัจจุบัน (Currently) หรือแบบเรียงลำดับ (sequence) หรือ สอบถามข้อมูลแบบไม่เรียงลำดับ (Non-sequence) เป็นการระบุประเภทเวลาเป็นแบบปัจจุบัน (Currently)

ตัวอย่างที่ 4.15 ต้องการระบุประเภทเวลาเป็น ณ.ปัจจุบัน

สามารถเขียนเป็นส่วนประเภทเวลาได้ดังนี้

Time State
Current

4.9 ตารางผลลัพธ์ (Result Relation)

การสอบถามข้อมูลก่อนหน้านี้ผลลัพธ์ที่ได้จะปรากฏเพียงแค่แบบแผนความสัมพันธ์เชิงเดี่ยว ถ้าผลลัพธ์จากการสอบถามข้อมูลจะต้องใช้แอททริบิวต์ซึ่งมาจากหลายๆตารางความสัมพันธ์ จึงต้องมีกลไกที่จะสามารถแสดงผลลัพธ์ที่ได้ให้อยู่ในตารางเดียว ซึ่งเป็นตารางที่สร้างขึ้นชั่วคราวเรียกว่าตารางผลลัพธ์ ซึ่งจะแสดงรายละเอียดของแอททริบิวต์ที่เป็นผลจากการสอบถามข้อมูล ด้วยการใช้คำสั่ง “P.” ใส่ลงในตารางผลลัพธ์

ตัวอย่างที่ 4.16 ต้องการหา customer-name, customer-city และaccount-number สำหรับลูกค้าทุกคนซึ่งมีบัญชีอยู่ที่สาขา Perryridge หากใช้พีชคณิตสัมพันธ์มีขั้นตอนการหาคำตอบดังนี้คือ

1. Join ตาราง Deposit และตาราง customer
2. Project customer-name, customer-city และ account-number

แต่ถ้าใช้ TQBE ในการหาคำตอบโดยการ

1. สร้างตารางโครงร่างชื่อ Result ประกอบด้วยแอททริบิวต์ customer-name, customer-city และ account-number
2. สร้างการสอบถามข้อมูล

ผลลัพธ์ที่ได้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Deposit	branch-name	account-number	customer-name	balance
	Perryridge	_z	_x	

Customer	customer-name	Street	customer-city
	_x		_y

Result	customer-name	customer-city	customer-city
P.	_x	_y	_z

4.10 ตัวปฏิบัติการรวม (Aggregate Operation)

QBE จะมีตัวปฏิบัติการรวมได้แก่ AVG, MAX, MIN, SUM และ CNT ตัวปฏิบัติการเหล่านี้จะต้องตามด้วย ALL เพื่อยืนยันว่าต้องทำกับทุกค่าในตาราง

ตัวอย่างที่ 4.17 ให้หายอดเงินรวมของบัญชีที่ชื่อ Janes ถืออยู่

สามารถเขียนเป็นตารางโครงร่างได้ดังนี้

Deposit	branch-name	account-number	customer-name	balance
			Janes	P.SUM.ALL

เพื่อจำกัดความซ้ำซ้อนจะต้องใช้ตัวปฏิบัติการรวม UNQ และตามด้วย ALL

ตัวอย่างที่ 4.18 ให้หาจำนวนลูกค้าทั้งหมดซึ่งมีบัญชีอยู่ที่สาขา Perryridge

สามารถเขียนเป็นตารางโครงร่างได้ดังนี้

Deposit	branch-name	account-number	customer-name	balance
	Perryridge		P.CNT.UNQ.ALL	

TQBE สามารถใช้ฟังก์ชันเพื่อรวมทับเปิลโดยใช้ตัวปฏิบัติการ G ซึ่งจะคล้ายคลึงกับ group by ในภาษา SQL

ตัวอย่างที่ 4.19 ให้หายอด balance เฉลี่ยในแต่ละสาขา

สามารถเขียนเป็นตารางโครงร่างได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Deposit	branch-name	account-number	customer-name	balance
	P.G.			P.AVG.ALL._x

หากต้องการหายอด balance เฉลี่ยของบัญชีในสาขาใดๆซึ่งค่าเฉลี่ยยอด balance ของบัญชีทั้งหมดมากกว่า 1200 ทำได้โดยการเพิ่มส่วนเงื่อนไข

Conditions
AVG.ALL_x > 1200

4.11 การจัดการข้อมูล (Data Manipulation)

การจัดการข้อมูลใน TQBE โดยใช้ตารางโครงร่างอันได้แก่ การเพิ่มข้อมูล การลบข้อมูล และการแก้ไขข้อมูล

4.11.1 การเพิ่มข้อมูล (Insertion)

การเพิ่มข้อมูลเข้าไปในตารางความสัมพันธ์ จะต้องระบุรายละเอียดของค่าแอททริบิวต์ที่ต้องการที่จะเพิ่มในทับเปิล โดยใช้ตัวปฏิบัติการ " I " ใส่ลงใต้ชื่อตารางโครงร่าง ค่าของแอททริบิวต์ที่จะเพิ่มจะต้องเป็นสมาชิกของโดเมนแอททริบิวต์

ตัวอย่างที่ 4.20 ต้องการเพิ่มทับเปิลใหม่ในตาราง Deposit ซึ่งเจ้าของบัญชีคือ Smith มียอดเงิน 1200 ในบัญชีเลขที่ 9732 ที่สาขา Perryridge

จะได้ตารางโครงร่างดังนี้

Deposit	branch-name	account-number	customer-name	balance
I.	Perryridge	9732	Smith	1200

เราสามารถเพิ่มทับเปิลซึ่งมีข้อมูลเพียงบางส่วนได้

ตัวอย่าง 4.21 ต้องการเพิ่มข้อมูลในตาราง Branch ซึ่งเป็นสาขาใหม่โดยมีค่าของ branch-name เข้าเป็น Capital และค่าของ branch-city เป็น Queens

ไม่ได้มีการกำหนดค่าของ assets ดังนั้นค่าของ assets จะเป็น null ดังแสดงตารางโครงร่างดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Branch	branch-name	Assets	branch-city
I.	Capital	-	Queens

ยิ่งไปกว่านั้นเราสามารถเพิ่มทับเปิลเข้าไปในตารางผลลัพธ์ที่ได้จากการสอบถามข้อมูล

ตัวอย่างที่ 4.22 ต้องการเพิ่มเงินในบัญชีเงินฝากจำนวน 200 ให้กับลูกค้าเงินกู้ที่สาขา Perryridge ทุกคนโดยที่เลขที่บัญชีเงินฝากได้มาจากเลขที่บัญชีเงินกู้ได้

จะได้ตารางโครงร่างดังนี้

Deposit	branch-name	account-number	customer-name	balance
I.	Perryridge	_x	_y	200

Borrow	branch-name	loan-number	customer-name	balance
	Perryridge	_x	_y	

4.11.2 การลบข้อมูล (Deletion)

การลบทับเปิลใดๆออกจากตารางความสัมพันธ์ โดยทั่วไปจะคล้ายกับการสอบถามข้อมูล ส่วนที่แตกต่างกันอย่างเห็นได้ชัดคือใช้คำสั่ง " D." ในการลบแทนที่คำสั่ง P. นอกจากการลบข้อมูลทั้งทับเปิลแล้วเรายังสามารถลบค่าในคอลัมน์ที่ต้องการได้ ซึ่งค่าในคอลัมน์ที่ถูกลบจะเป็น null และถูกแทนด้วยเครื่องหมาย "-"

ตัวอย่างที่ 4.23 ต้องการลบบัญชีของลูกค้าชื่อ Smith

จะได้ตารางโครงร่างดังนี้

Deposit	Branch-name	account-number	customer-name	balance
D.			Smith	

ตัวอย่างที่ 4.24 ต้องการลบเงินกู้ทั้งหมดซึ่งมี loan-number มีค่าระหว่าง 1300 ถึง 1500

จะได้ตารางโครงร่างดังนี้

Borrow	Branch-name	Loan-number	customer-name	Amount
D.		_x		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Conditions
_x = (≥ 1300 and ≤ 1500

ตัวอย่างที่ 4.25 ต้องการลบบัญชีทั้งหมดซึ่งมีสาขาตั้งอยู่ใน Brooklyn
จะได้ตารางโครงร่างดังนี้

Deposit	Branch-name	account-number	customer-name	balance
D.	_x			

Branch	branch-name	Assets	branch-city
	_x		Brooklyn

4.11.3 การแก้ไขข้อมูล (Updates)

ในกรณีที่เราต้องการเปลี่ยนแปลงค่าบางค่าในทับเปิลเท่านั้น ไม่ได้เปลี่ยนปลงค่าทั้งหมด เราใช้ตัวปฏิเสธ "U." อย่างไรก็ตามไม่อนุญาติให้ผู้ใช้แก้ไขข้อมูลของฟิลด์ที่เป็นคีย์หลัก (primary key)

ตัวอย่างที่ 4.26 ต้องการแก้ไขค่า asset เป็น 10,000,000 ซึ่งมีค่า branch-name เป็นPerryridge
จะได้ตารางโครงร่างดังนี้

Branch	branch-name	Assets	branch-city
	Perryridge	U.10,000,000	

ฟิลด์ซึ่งว่างในแอททริบิวต์ branch-city แสดงว่าไม่มีการเปลี่ยนแปลงค่าในแอททริบิวต์ดังกล่าว
ในกรณีที่ต้องการเปลี่ยนแปลงค่าในฟิลด์โดยใช้ค่าเก่าของฟิลด์นั้น จะต้องมีการระบุค่าสองแถว แถวแรกเป็นการแสดงค่าเก่าซึ่งต้องแก้ไข ส่วนอีกแถวจะแสดงค่าใหม่ที่ต้องการที่จะเพิ่มในตาราง

ตัวอย่างที่ 4.27 ต้องการเพิ่มดอกเบี้ยเป็น 5% ของยอดทั้งหมด
จะได้ตารางโครงร่างดังนี้

Deposit	branch-name	account-number	customer-name	balance
U.				_x * 1.05
				_x

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สถาปัตยกรรมของระบบ TQBE

ระบบจัดการฐานข้อมูลเชิงเวลา TQBE ที่ได้พัฒนาขึ้นนี้ได้ออกแบบสำหรับใช้กับระบบจัดการฐานข้อมูลเชิงพาณิชย์ในปัจจุบันไม่ว่าจะเป็น ORACLE INFORMIX SYBASE INGRES หรือแม้กระทั่ง Microsoft Access ทำให้ฐานข้อมูลดังกล่าวสามารถสนับสนุนข้อมูลที่แปรเปลี่ยนตามเวลาได้ ในการเก็บข้อมูลทั้งในอดีต ปัจจุบัน และอนาคต ระบบ TQBE เป็นทั้งระบบจัดการและภาษาสอบถามข้อมูล พัฒนาขึ้นโดยมีรูปแบบโครงสร้างเป็นกราฟฟิก ตามแนวคิดของ QBE (Query-By-Example) [10] อีกทั้งสนับสนุนข้อมูลที่เกี่ยวเนื่องกับเวลาและเงื่อนไขต่างๆที่เกี่ยวข้องกับเวลา

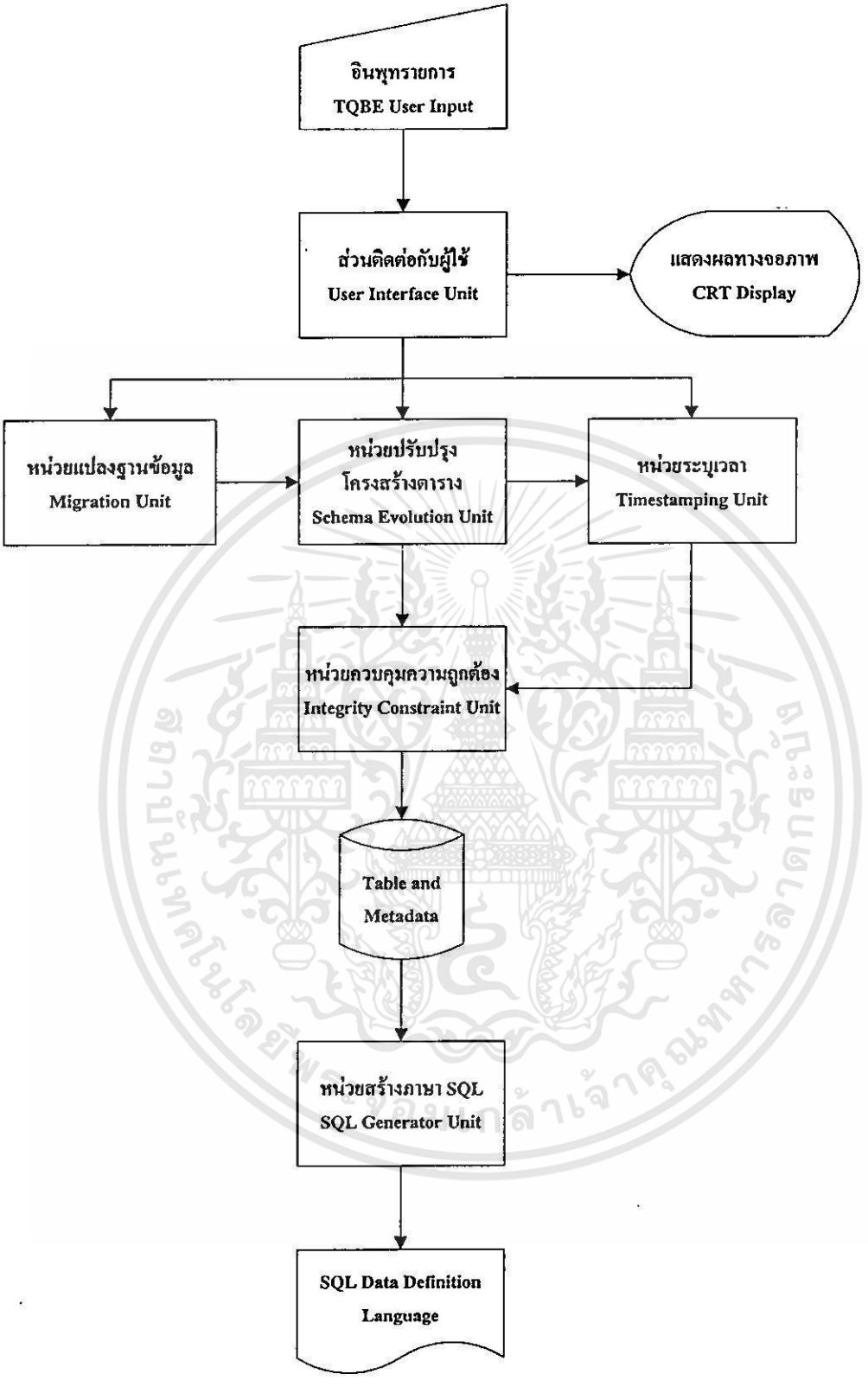
5.1 สถาปัตยกรรมของระบบ TQBE โดยรวม

ในการออกแบบระบบ TQBE เพื่อให้ระบบจัดการฐานข้อมูลเดิมสามารถสนับสนุนข้อมูลและปฏิบัติการเชิงเวลาได้นั้น เป็นการออกแบบและพัฒนาระบบที่ส่วนไคล์เอนท์ของโปรแกรมประยุกต์โดยตรง ประกอบด้วยส่วนทำงานหลักๆ 7 ส่วนคือ ส่วนติดต่อกับผู้ใช้ หน่วยปรับปรุงแบบแผนโครงสร้างตาราง หน่วยระบุเวลาเข้ากับตารางฐานข้อมูล หน่วยควบคุมความถูกต้องของข้อมูล หน่วยสร้างภาษาสอบถามเชิงโครงสร้าง ส่วนแสดงผลลัพธ์ และหน่วยแปลงฐานข้อมูลแบบไม่อิงเวลาเป็นฐานข้อมูลเชิงเวลา ดังแสดงในรูปที่ 5.1

5.2 ส่วนติดต่อกับผู้ใช้

ส่วนติดต่อผู้ใช้งานนี้ เป็นโปรแกรมส่วนที่ผู้ใช้งานเป็นผู้เรียกใช้ในการเข้าถึงข้อมูล โดยส่วนนี้จะมีหน้าที่ในการรับข้อมูล จากการเลือกฐานข้อมูล เลือกตาราง เลือกแอททริบิวต์ เพื่อการสอบถามข้อมูลในส่วนที่ผู้ใช้ต้องการ ผู้ใช้ยังสามารถระบุเงื่อนไขพิเศษอื่นๆเพิ่มเติมไม่ว่าจะเป็นเงื่อนไขทางเซ็ท เงื่อนไขทางคณิตศาสตร์ กระทั่งเงื่อนไขทางเวลาในการทำงานของ TQBE ในส่วนติดต่อผู้ใช้ส่วนนี้ผ่านทางไมโครซอฟท์วินโดว์ ลักษณะการทำงานโดยทั่วไปจะเป็นการคลิ๊กเลือกสิ่งที่ผู้ใช้ต้องการ หลังจากการเลือกเสร็จเรียบร้อยแล้ว ระบบสามารถแสดงผลลัพธ์ที่ได้ให้โดยทันที ทำให้การใช้งานง่ายและสะดวกกับผู้ใช้งานมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 สถาปัตยกรรมของระบบ TQBE โดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 หน่วยปรับปรุงโครงสร้างตาราง

ในการสร้างแอททริบิวต์ของเวลาเข้ากับความสัมพันธ์ของตารางกระทำใน 2 ส่วนคือ กระทำในขณะที่สร้างตาราง และกระทำกับตารางข้อมูลเดิม เป็นการเพิ่มแอททริบิวต์ของเวลา 2 ค่า เข้าไปในแบบแผนโครงสร้างของตาราง (schema) สำหรับตารางที่ต้องการส่วนสนับสนุนทางเวลา คือค่าแอททริบิวต์ FROM_DATE และค่าแอททริบิวต์ TO_DATE ซึ่งประเภทของแอททริบิวต์จะเป็น date ซึ่งมีรายละเอียดโครงสร้างตาราง ดังแสดงในรูปที่ 5.2

Field 1	Field 2	Field 3	...	...	FROM_DATE	TO_DATE

Field Name	Data Type
FDYD_ID	Text
LOT_ID_NUM	Text
PEN_ID	Text
HD_CNT	Number
FROM_MOVE_ORDER	Text
TO_MOVE_ORDER	Text
RECORD_DATE	Date
FROM_DATE	Date
TO_DATE	Date

รูปที่ 5.2 โครงสร้างตารางซึ่งมีส่วนสนับสนุนทางเวลา

5.4 หน่วยระบุเวลาเข้ากับตารางฐานข้อมูล (Timestamping Unit)

การระบุเวลาของระบบ TQBE นั้นเป็นการระบุเวลาด้วยวิธีการระบุเวลาที่กระทำกับทับเปิล และใช้ช่วงเวลาในการระบุเวลา โดยที่ขอบเขตล่าง (lower bound) ให้ทำการระบุเวลาที่แอททริบิวต์ from_date ส่วนขอบเขตบน (upper bound) ให้ทำการระบุเวลาที่แอททริบิวต์ to_date และเวลาที่ทำการระบุนั้นเป็นค่า valid time ด้วย หน่วยย่อยเวลา (time granularity) เป็นวันแทนด้วยปี/เดือน/วัน วันที่ ณ. ปัจจุบันได้มาจากวันที่ของระบบคอมพิวเตอร์ ส่วนวันที่ที่เป็น ∞ แทนด้วยค่า '9999-12-31' ดังแสดงในรูปที่ 5.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FDYD_ID	LOT_ID_NUM	PEN_ID	HD_CNT	FROM_DATE	TO_DATE
1	137	1	17	1998-02-07	1998-02-18
1	219	1	43	1998-02-25	1998-03-01
1	219	1	20	1998-03-01	1998-03-14
1	219	2	23	1998-03-01	1998-03-14
1	219	2	43	1998-03-14	9999-12-31
1	374	1	14	1998-02-20	9999-12-31

รูปที่ 5.3 ผลลัพธ์ที่ได้จากหน่วยระบุเวลา

5.5 หน่วยสร้างภาษาสอบถามเชิงโครงสร้าง

ทำหน้าที่สร้างภาษาสอบถามเชิงโครงสร้าง โดยการนำข้อมูลมาจากส่วนติดต่อกับผู้ใช้ ซึ่งผู้ใช้ได้ทำการเลือกตารางความสัมพันธ์ คอลัมน์ เงื่อนไขต่างๆ นำมาวิเคราะห์และประมวลผลเพื่อสร้างคำสั่งภาษาสอบถามเชิงโครงสร้าง (SQL) เพื่อส่งให้ระบบจัดการฐานข้อมูล เพื่อสอบถามข้อมูลในลำดับต่อไป ดังแสดงในรูปที่ 5.4

```
SELECT L1.LOT_ID_NUM , L2.LOT_ID_NUM , L1.PEN_ID
FROM   LOT_LOC AS L1 , LOT_LOC AS L2
WHERE L1.LOT_ID_NUM < L2.LOT_ID_NUM
        AND L1.PEN_ID = L2.PEN_ID
        AND L1.TO_DATE = DATE ' 9999-12-31 '
        AND L2.TO_DATE = DATE ' 9999-12-31 '
```

รูปที่ 5.4 ผลลัพธ์ที่ได้จากหน่วยสร้างภาษาสอบถามเชิงโครงสร้าง

5.6 ส่วนแสดงผลลัพธ์

ทำหน้าที่สร้างผลลัพธ์ที่ได้แสดงออกทางจอภาพด้วยการสอบถามข้อมูลจากฐานข้อมูลซึ่งติดต่ออยู่โดยใช้ภาษาสอบถามเชิงโครงสร้างที่ได้มาจากหน่วยสร้างภาษาสอบถามเชิงโครงสร้าง

5.7 หน่วยแปลงฐานข้อมูล

ในการใช้ระบบจัดการฐานข้อมูลเดิม ซึ่งเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์หรือระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ ระบบดังกล่าวมีโปรแกรมประยุกต์ใช้งานอยู่ เราสามารถเปลี่ยน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แปลงฐานข้อมูลเดิมให้เป็นฐานข้อมูลเชิงเวลา โดยไม่ส่งผลกระทบต่อโปรแกรมประยุกต์ที่มีอยู่เดิม อีกทั้งยังสามารถจัดเก็บและสอบถามข้อมูลที่แปรตามเวลาได้ โดยใช้หน่วยแปลงฐานข้อมูลในระบบ TQBE หน่วยแปลงฐานข้อมูลจะทำการอ่านตารางทุกตารางในฐานข้อมูลแล้วทำการขยายส่วนโครงสร้างตารางให้สามารถสนับสนุนข้อมูลเชิงเวลาได้โดยการเพิ่มแอททริบิวต์สำหรับเวลาโดยเฉพาะ

5.8 หน่วยควบคุมความถูกต้องข้อมูล (Data Integrity Constraints)

5.8.1 ส่วนบังคับความถูกต้องของผลลัพธ์ (Column Constraint)

ในการแสดงชื่อคอลัมน์ที่ปรากฏของลิสต์บ็อกซ์ ในส่วนติดต่อกับผู้ใช้ เพื่อให้ผู้ใช้ได้ทำการเลือกคอลัมน์ที่ต้องการในระบบ TQBE นั้น ระบบจะทำการตรวจสอบชื่อแอททริบิวต์ที่แสดงต่อผู้ใช้ว่าเป็นแอททริบิวต์ของระบบ ซึ่งแอททริบิวต์ของระบบคือ แอททริบิวต์ FROM_DATE และแอททริบิวต์ TO_DATE ถ้าตรวจสอบแล้วพบว่าเป็นแอททริบิวต์ของระบบ ระบบ TQBE จะไม่แสดงแอททริบิวต์ดังกล่าวปรากฏต่อผู้ใช้ ส่วนแอททริบิวต์อื่นๆจะปรากฏให้เห็นตามปกติ

5.8.2 ส่วนบังคับความถูกต้องของประเภทข้อมูล (Data Constraint)

การสอบถามข้อมูลและการจัดการกับข้อมูลเชิงเวลาไม่ว่าจะเป็นข้อมูลในแบบ Current, Current, History หรือ Sometime ซึ่งเป็นการแทนข้อมูลในแบบ Sequence และ Nonsequence ตามลำดับ จะต้องมีส่วนควบคุมความถูกต้องเพื่อให้ข้อมูลที่ได้ถูกต้องเป็นไปตามลักษณะของข้อมูลแต่ละประเภทรายละเอียดดังแสดงในบทที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การพัฒนาระบบ TQBE

โดยปกติของการพัฒนาโปรแกรมเพื่อวัตถุประสงค์สำหรับการใช้งานใดก็ตาม จำเป็นจะต้องมีเครื่องมือสำหรับช่วยในการพัฒนาเพื่อให้บรรลุตามวัตถุประสงค์นั้น สำหรับการพัฒนาระบบ TQBE ก็เช่นกันต้องใช้เครื่องมือหลายชนิดทั้งฮาร์ดแวร์และซอฟท์แวร์ ในบทนี้จะเป็นการเสนอและอธิบายเกี่ยวกับรายละเอียดของอุปกรณ์ที่ใช้ในการพัฒนาระบบโปรแกรมรวมถึงหน่วยโปรแกรมต่างๆซึ่งเป็นส่วนประกอบสำคัญของระบบ

6.1 อุปกรณ์ที่ใช้ในการพัฒนาระบบ TQBE

6.1.1 เครื่องไมโครคอมพิวเตอร์ IBM PC

เครื่องไมโครคอมพิวเตอร์ IBM PC คอมแพทติเบิลใช้เป็นฮาร์ดแวร์หลักในการพัฒนาระบบ TQBE ประกอบด้วยหน่วยประมวลผลกลาง (Central Processing Unit) ขนาด 32 บิทของบริษัท อินเทล เบอร์ Pentium Celelon 333 MHz หน่วยความจำขนาด 48 เมกกะไบท์ (MB) ฮาร์ดดิสก์ขนาด 4 จิกกะไบท์ (GB) ฟลอปปี้ดิสก์ขนาด 1.44 เมกะไบท์ ซีดีรอมความเร็ว 48x จอสี Super VGA ที่ระดับความละเอียด 1024 x 768 และ 1.6 ล้านสี แป้นพิมพ์เอาท์คีย์ไมโครซอฟท์เมาท์ 1 ตัว ต่อเข้ากับระบบอินพุท เอ๊าท์พุท แบบอนุกรมตามมาตรฐาน RS-232

6.1.2 ระบบปฏิบัติการไมโครซอฟท์วินโดว์

ระบบปฏิบัติการไมโครซอฟท์วินโดว์ใช้เป็นระบบปฏิบัติการบนเครื่อง IBM PC ผลิตโดยบริษัทไมโครซอฟท์ โดยรุ่นที่ใช้ในการพัฒนาระบบ TQBE นี้เป็นระบบปฏิบัติการไมโครซอฟท์วินโดว์ 98 ไทย

6.1.3 โปรแกรมไมโครซอฟท์ Visual Basic สำหรับวินโดว์

โปรแกรมไมโครซอฟท์ Visual Basic สำหับวินโดว์ใช้เป็นซอฟท์แวร์แปลภาษา Visual Basic ที่เขียนขึ้นเพื่อให้สามารถพัฒนาระบบประมวลผล และใช้งานบนไมโครซอฟท์วินโดว์ได้ ผลิตโดยบริษัทไมโครซอฟท์ โดยรุ่นที่ใช้ในการพัฒนาระบบ TQBE นี้เป็นโปรแกรมไมโครซอฟท์ Visual Basic 6.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1.4 ระบบจัดการฐานข้อมูลไมโครซอฟท์ Access

ระบบจัดการฐานข้อมูลไมโครซอฟท์ Access ใช้เป็นซอฟท์แวร์ระบบจัดการฐานข้อมูลสำหรับเครื่อง IBM PC ผลิตโดยบริษัทไมโครซอฟท์ โดยรุ่นที่ใช้ในการพัฒนาระบบ TQBE นี้เป็นระบบจัดการฐานข้อมูลไมโครซอฟท์ Access 2.0

6.2 การพัฒนาระบบในส่วนต่างๆ

ระบบจัดการฐานข้อมูลเชิงเวลา TQBE ถูกพัฒนาขึ้นโดยซอฟท์แวร์แปลภาษาไมโครซอฟท์วิชอลเบสิค (Microsoft Visual Basic) รุ่น 6.0 โดยใช้หลักการและแนวคิดของการสอบถามข้อมูลเชิงเวลาแบบใช้ตัวอย่าง (Temporal-Query-By-Example : TQBE) ระบบ TQBE ประกอบด้วยส่วนการทำงานหลัก 2 ส่วนหลักคือ ส่วนการสอบถามข้อมูลเชิงเวลา และส่วนการจัดการข้อมูลเชิงเวลา ดังรายละเอียดต่อไปนี้

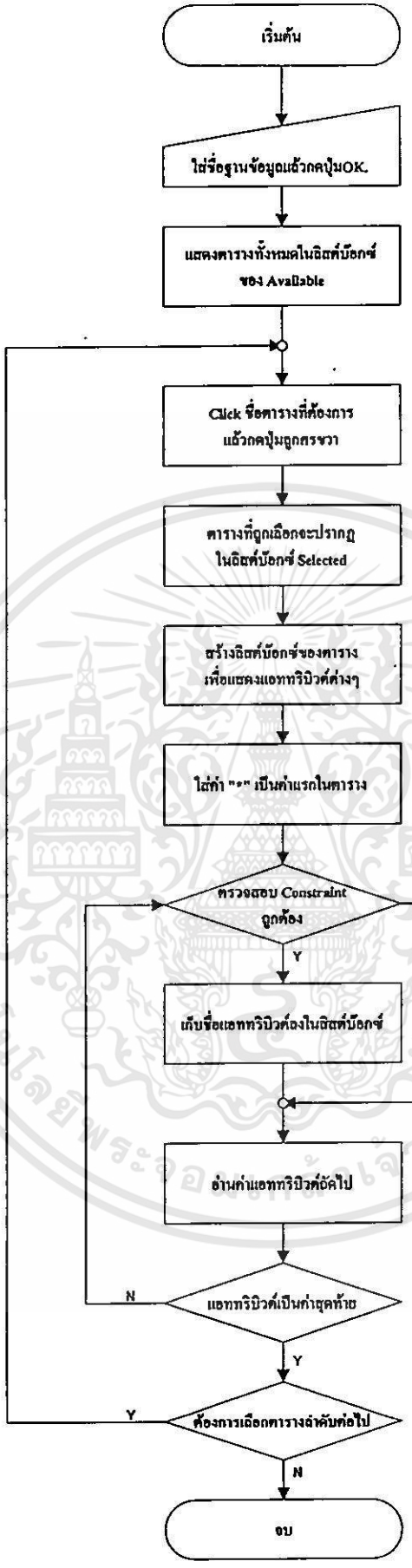
6.2.1 การพัฒนาส่วนการสอบถามข้อมูล

ตามที่ได้กล่าวไปเมื่อตอนต้นแล้วว่าระบบTQBE ได้รับแนวคิดมาจาก QBE การสอบถามข้อมูล จึงเป็นการสอบถามข้อมูลซึ่งสะดวกและง่ายต่อการใช้งาน โดยผู้ใช้เพียงคลิ๊กเลือกรายการต่างๆซึ่งปรากฏอยู่ในแท็บ (tab) ต่างๆในระบบ TQBE อันได้แก่ Table, Column, Condition, Time Range, SQL และ Output

- **Table**

เป็นส่วนติดต่อกับผู้ใช้งานส่วนแรกที่พัฒนาขึ้นเพื่อใช้ในการเลือกฐานข้อมูล ตาราง และความสัมพันธ์ เมื่อผู้ใช้ทำการเลือกฐานข้อมูลเป็นที่เรียบร้อย จะปรากฏตารางทั้งหมดในลิสต์บ๊อกซ์ (list box) ของ Available จากนั้นทำการเลือกตารางจะปรากฏตารางที่เลือกในลิสต์บ๊อกซ์ของ SELECTED ทางด้านขวามือ เมื่อทำการเลือกตารางเสร็จเรียบร้อยจะปรากฏแบบแผนโครงสร้างตารางที่แท็บของ Column, Condition และ Time Range สำหรับผู้ใช้ในการเลือกรายละเอียดของแอททริบิวต์ที่ต้องการสำหรับใช้ในส่วนอื่นๆต่อไป จากนั้นผู้ใช้สามารถเลือกความสัมพันธ์ระหว่างตารางหรือภายในตารางเดียวกันได้ เมื่อผู้ใช้คลิ๊กปุ่ม Relationships จะปรากฏตารางในการจับคู่ความสัมพันธ์ตามต้องการ ขั้นตอนการทำงานดังแสดงในรูป 6.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.1 ผังขั้นตอนการพัฒนาและการทำงานในส่วน Table

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Column

ในส่วนนี้จะแสดงตารางทั้งหมดที่เลือกมาจากแท็บ Table ตารางทุกตารางอยู่ในรูปของลิสต์บอกซ์ ซึ่งแต่ละลิสต์บ๊อกซ์จะแทนตารางแต่ละตาราง ภายในลิสต์บ๊อกซ์จะแสดงรายละเอียดของแอททริบิวต์ต่างๆภายในตารางซึ่งได้มาจากการทำงานในส่วนของแท็บ Table เช่นกัน ระบบจะทำการซ่อนแอททริบิวต์ที่ระบบใช้เพื่อการระบุเวลา (timestamping) ผู้ใช้จะไม่เห็นและไม่สามารถเลือกได้

ในการพัฒนาส่วนของแท็บใน Column นี้ทำให้ผู้ใช้สามารถเลือกแอททริบิวต์ใดๆภายในตารางโดยการ double-click แอททริบิวต์นั้นๆ ทำให้แอททริบิวต์ดังกล่าวปรากฏในส่วนของกริดบ๊อกซ์ (grid box)

ภายในกริดบ๊อกซ์มีการแสดงรายละเอียดที่จำเป็นซึ่งเป็นข้อมูลในการประมวลผลในลำดับต่อไปนี้ได้แก่ค่า Field , Table , Sort , Show , Criteria , Or และ Function หลังจากทำการเลือกแอททริบิวต์ในตารางโดย double-click แอททริบิวต์ที่ต้องการแล้ว แอททริบิวต์ดังกล่าวจะปรากฏชื่อในช่อง Field พร้อมทั้งชื่อตารางของแอททริบิวต์นั้นๆจะปรากฏในช่อง Table ระบบจะทำการระบุให้เป็นการเลือกในช่อง Show ด้วยการแสดงค่ากากบาท (x) ในเบื้องต้น แต่ระบบให้ผู้ใช้สามารถเลือกที่จะไม่แสดงผลแอททริบิวต์ดังกล่าวได้โดยการ click เพื่อทำการยกเลิกการเลือก (disable)

เมื่อทำการ click ที่ช่อง sort ระบบจะแสดง combo box ให้ผู้ใช้ได้เลือกว่าเป็นการ sort แบบ Ascending หรือ Descending

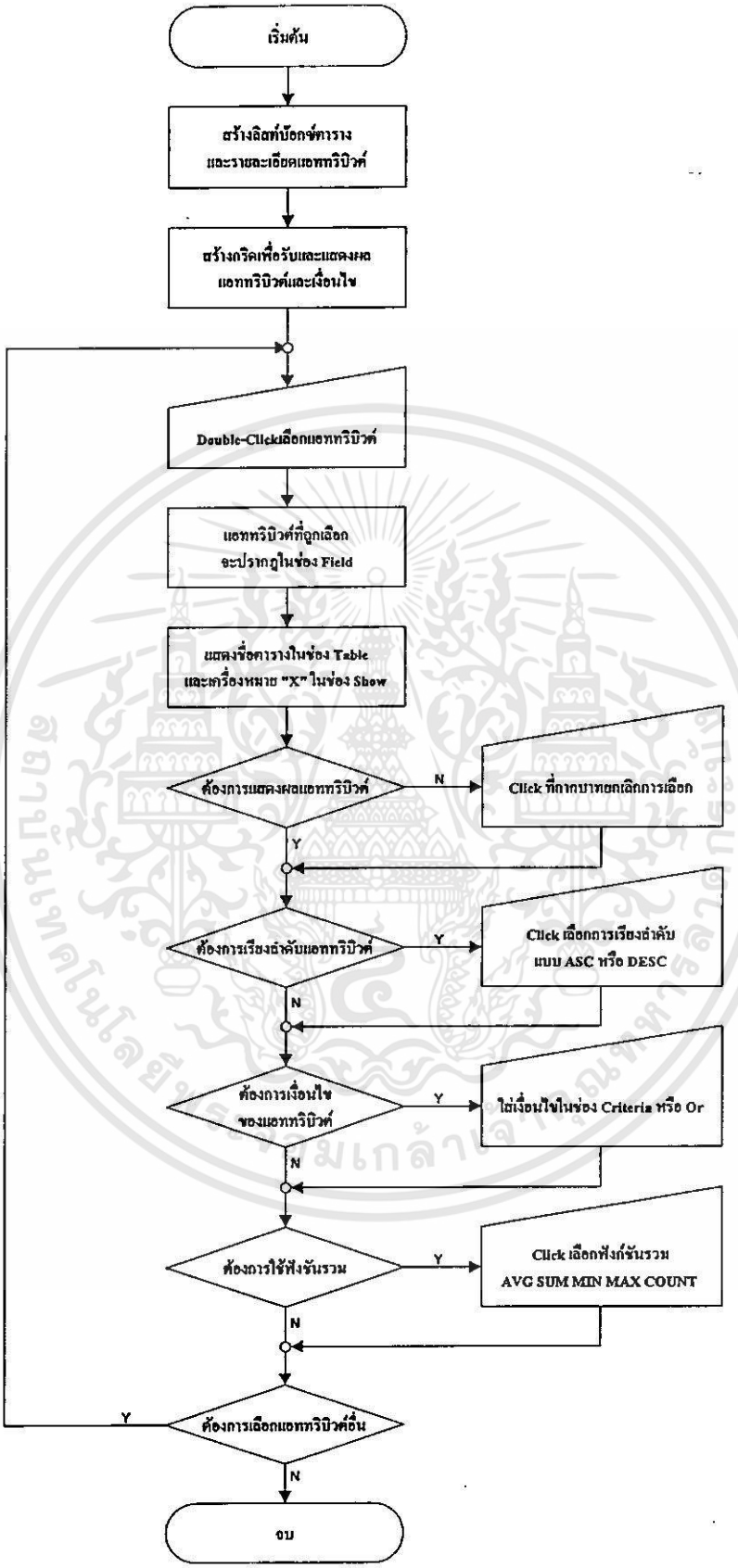
ในส่วนของ Criteria และ Or รับจะรับข้อมูลเกี่ยวกับเงื่อนไขเฉพาะของแอททริบิวต์นั้นๆ ที่ช่อง Function ระบบจะสร้าง combo box เพื่อแสดงค่าฟังก์ชันรวม (Aggregate function) ได้แก่ AVG , SUM , MIN , MAX , COUNT สำหรับเป็นส่วนหนึ่งเพื่ออำนวยความสะดวกแก่ผู้ใช้ ขั้นตอนการทำงานดังแสดงในรูป 6.2

- Condition

ส่วนนี้จะนำเอาแอททริบิวต์ทั้งหมดที่ผู้ใช้เลือกจากแท็บ Column มาแสดงในลิสต์บ๊อกซ์ของ Field ซึ่งทุกๆแอททริบิวต์จะนำหน้าด้วยชื่อตารางตามด้วยจุด (.) และชื่อแอททริบิวต์ตามลำดับ

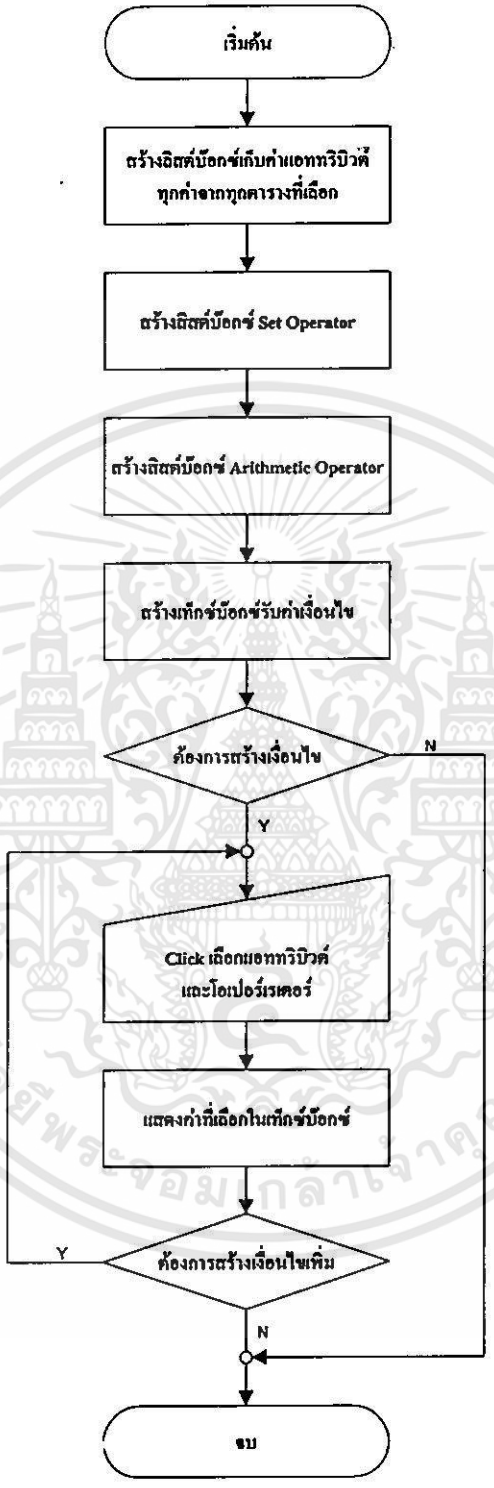
เมื่อมีการตัวปฏิบัติการซึ่งจำเป็นสำหรับเงื่อนไขที่กระทำระหว่างแอททริบิวต์ แบ่งออกเป็น 2 ประเภท คือ ตัวปฏิบัติการทางเซ็ทและตรรกะได้แก่ UNION, INTERSECT,DIFFERENCE , AND , OR , NOT และตัวปฏิบัติการทางคณิตศาสตร์และเปรียบเทียบได้แก่ + , - , * , / , > , < , = , < > , (,) ผู้ใช้สามารถคลิ๊กเลือกได้ตามต้องการ ส่วนที่เลือกจะปรากฏใน text box ด้านล่าง ขั้นตอนการทำงานดังแสดงในรูปที่ 6.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.2 ผังขั้นตอนการพัฒนาและการทำงานในส่วน Column

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.3 ผังขั้นตอนการพัฒนาและการทำงานในส่วน Condition

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Time Range

ในส่วนนี้จะมีลิสต์บ๊อกซ์ของ Field ซึ่งมีรายละเอียดของแอททริบิวต์ทั้งหมดของทุกตารางที่เลือกจากแท็บ Column เช่นเดียวกันกับในแท็บ Condition แต่จะมีลิสต์บ๊อกซ์ของเวลาและตัวเลือกแบบเรดิโอของเวลาเพื่อบังคับและควบคุมการสอบถามข้อมูลเชิงเวลา รวมทั้งมีตัวปฏิบัติการเชิงเวลาทั้ง 13 ตัวของ Allen ได้แก่ BEFORE , AFTER , MEET , MET-BY , OVER-LAP , OVERLAP-BY , DURING , INCLUDES , START , STARTED-BY , FINISH , FINISHED-BY และ EQUAL สำหรับช่วงเวลา

ตัวเลือกแบบเรดิโอที่ระบบสร้างขึ้นสำหรับผู้ใช้นั้นมีอยู่ 2 ส่วนคือ ส่วนที่เป็น Time Unit ส่วนนี้จะเป็นการระบุการใช้หน่วยย่อยเวลาของผู้ใช้ว่าเป็น DAY , MONTH หรือ YEAR อีกส่วนคือ Time State ส่วนนี้จะเป็นการระบุวิธีการสอบถามข้อมูลว่าเป็นแบบใดระบบจะมีให้ผู้ใช้ได้เลือก 3 แบบคือ Current , History และ Sometime ซึ่งจะแทนการสอบถามข้อมูล ณ. เวลาปัจจุบัน (current) แบบเรียงลำดับ (sequence) และแบบไม่เรียงลำดับ (Non-Sequence) [11] ตามลำดับ

ระบบจะทำการประมวลผลจากการเลือกของผู้ใช้ว่าต้องการการสอบถามข้อมูลเชิงเวลาในรูปแบบใด ตามที่กำหนดไว้ในตัวเลือกแบบเรดิโอ เก็บค่าไว้ในตัวแปร SelTemp และตัวแปร WhrTemp โดยที่ระบบจะทำการพิจารณาว่าถ้าค่าตัวเลือกแบบเรดิโอเป็น current ระบบจะกำหนดค่า TO_DATE = '9999-12-31' (ณ . ปัจจุบัน) ให้กับตัวแปร WhrTemp แต่ถ้าค่าตัวเลือกเรดิโอเป็น History ระบบจะกำหนดค่า FROM_DATE และ TO_DATE เพิ่มเข้าไปในตัวแปร SelTemp หากค่าตัวเลือกเรดิโอเป็น Sometime ระบบจะผ่านส่วนนี้ไป ขั้นตอนการทำงานดังแสดงในรูปที่ 6.4

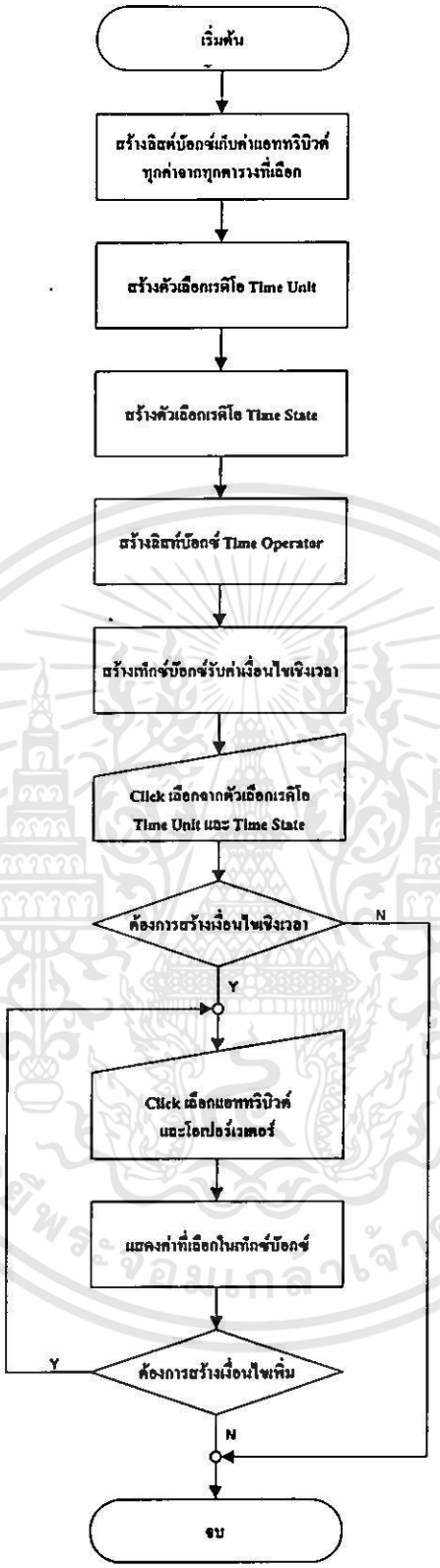
- SQL

ส่วนของ SQL นี้จะมีเฉพาะเท็กซ์บ๊อกซ์เพื่อรับค่าจากการประมวลผลของระบบจากข้อมูลทั้งหมดที่ผู้ใช้อินพุทข้อมูลโดยการเลือกจากแท็บต่างๆตั้งแต่แท็บ Table , Column , Condition และ Time Range ซึ่งมีลำดับขั้นตอนการทำงานดังนี้

ที่แท็บ Column ระบบจะทำการอ่านค่าแอททริบิวต์ทั้งหมดที่ปรากฏในช่อง Field ของแต่ละตารางในช่อง Table เก็บค่าที่อ่านได้ไว้ในตัวแปร FrmTemp จากนั้นตรวจสอบว่ามีค่าในช่อง Sort หรือไม่ ถ้ามีพิจารณาว่าเป็นแบบเรียงลำดับจากน้อยไปหามาก (ASC) หรือเรียงลำดับจากมากไปหาน้อย (DESC) เก็บค่าที่ได้ในตัวแปร OrdTemp แล้วทำการอ่านค่าในช่อง Criteria และ Or เก็บค่าที่ได้ไว้ในตัวแปร WhrTemp ในส่วนของฟังก์ชันรวม (Aggregate Function) ในช่อง Function ระบบจะทำการนำค่าฟังก์ชันที่เลือกมารวมกับแอททริบิวต์นั้นประกอบเป็นฟังก์ชันเก็บค่าที่ได้ในตัวแปร SelTemp เพื่อใช้ในการสร้างภาษา SQL รวมกับส่วนอื่นๆต่อไป

ที่แท็บ Condition ระบบจะทำการอ่านค่าจากเท็กซ์บ๊อกซ์ซึ่งผู้ใช้เลือกจากลิสต์บ๊อกซ์ Field ลิสต์บ๊อกซ์ Set Operator และลิวต์บ๊อกซ์ Arithmetic Operator ค่าที่ได้เก็บไว้ในตัวแปร CondTemp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



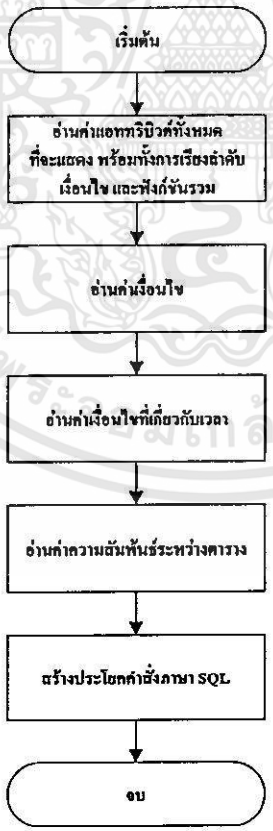
รูปที่ 6.4 ผังขั้นตอนการพัฒนาและการทำงานในส่วน Time Range

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่แท็บ Time Range ระบบจะทำการพิจารณาค่าตัวเลือกแบบเรดิโอ โดยถ้าตัวเลือกที่เลือกเป็น Current จะมีการสร้างประโยค TO_DATE = '9999-12-31' เข้าไปเก็บไว้ในตัวแปร WhrTemp ซึ่งต้องพิจารณาก่อนว่าตัวแปรนี้มีค่าอยู่หรือไม่ ถ้ามีระบบจะทำการเพิ่มค่าเข้าไปในตัวแปร WhrTemp แต่ถ้ายังไม่มีค่าในตัวแปรนี้ก็จะทำการสร้างค่าใหม่ให้กับตัวแปร WhrTemp หากตัวเลือกเรดิโอมีค่าเป็น History ระบบจะทำการเพิ่มค่าแอททริบิวต์ 2 ค่าคือ FROM_DATE และค่า TO_DATE ให้กับตัวแปร SelTemp แต่ถ้าหากตัวเลือกเรดิโอเป็นค่า Sometime ระบบจะคงสภาพการทำงานให้เป็นไปตามปกติ

ที่แท็บ Table ระบบจะพิจารณาค่า Relationships ว่ามีหรือไม่. ถ้ามีจะอ่านค่าในคอลัมน์ Table และคอลัมน์ Related Table เพื่อพิจารณาการจอย (join) ในตารางเดียวกัน(self-join) หรือคนละตารางกัน

ขั้นตอนสุดท้ายเป็นการรวบรวมค่าที่เก็บไว้ในตัวแปรทั้งหมดที่ได้จากแต่ละขั้นตอนได้แก่ ตัวแปร SelTemp , FrmTemp , WhrTemp , OrdTemp มาประกอบกับความสัมพันธ์ระหว่างตารางหรือในตารางเดียวกัน สร้างเป็นประโยคสอบถามเชิงโครงสร้างตามต้องการ ขั้นตอนการทำงานดังแสดงในรูปที่ 6.5



รูปที่ 6.5 ผังขั้นตอนการพัฒนาและการทำงานในส่วน SQL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Output

ในส่วนนี้จะนำประโยคสอบถามเชิงโครงสร้าง (SQL) ซึ่งได้จากขั้นตอนการสร้างภาษา SQL มาใช้เพื่อสอบถามข้อมูลจากระบบจัดการฐานข้อมูลผลลัพธ์ที่ได้จะเป็นทั้งการสอบถามข้อมูลแบบไม่อิงเวลา และการสอบถามข้อมูลเชิงเวลา ขึ้นอยู่กับผู้ใช้ว่าต้องการสอบถามข้อมูลแบบใด จากการเลือกและกำหนดค่าต่างๆในแต่ละขั้นตอนที่ผ่านมา

6.2.2 การพัฒนาในส่วนการเพิ่ม/ลบ/แก้ไขข้อมูล

การเพิ่ม/ลบ/แก้ไขข้อมูลเชิงเวลาในระบบ TQBE ใช้แนวคิดการสอบถามแบบใช้ตัวอย่างเช่นเดียวกัน เพื่อให้ผู้ใช้มีความสะดวกและง่ายต่อการจัดการข้อมูล โดยการระบุเวลาตามความต้องการที่ผู้ใช้เลือกในตัวเลือกเรดิโอทั้ง 2 คือ Time Unit และ Time State และมีปุ่มสั่งการ (command button) เพื่อเป็นการให้ระบบได้ทราบว่าต้องการทำกระบวนการในการ เพิ่ม ลบ หรือแก้ไขข้อมูล ทำให้เกิดขั้นตอนน้อยที่สุดและได้ผลลัพธ์สูงสุด ในระบบ TQBE นี้ได้แบ่งกระบวนการ 3 กระบวนการคือ การเพิ่มข้อมูล การลบข้อมูล และการแก้ไขข้อมูล ในแบบปัจจุบัน (current) เรียงลำดับ (sequenced) และไม่เรียงลำดับ (nonsequence) [11] ระบบ TQBE จะทำการอ่านข้อมูลทุกทับเปิลจากตารางซึ่งผู้ใช้เลือกก่อนหน้านี้ มาแสดงในกริดและสร้างปุ่มคำสั่งในการเพิ่ม/ลบ/แก้ไขข้อมูล ขั้นตอนการทำงานดังแสดงในรูปที่ 6.6

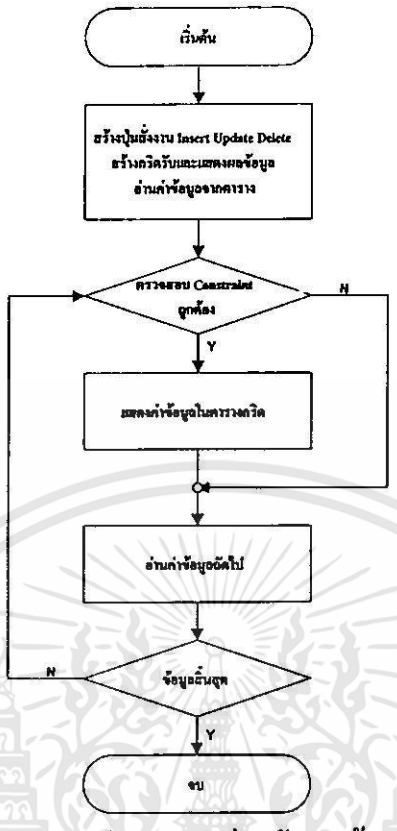
- การจัดการข้อมูลปัจจุบัน (Current Modification)

การจัดการข้อมูลในการเพิ่ม/ลบ/แก้ไข ข้อมูลเชิงเวลาในปัจจุบัน ระบบจะทำการระบุช่วงเวลาจากปัจจุบันจนถึงตลอดกาลให้เองโดยอัตโนมัติ

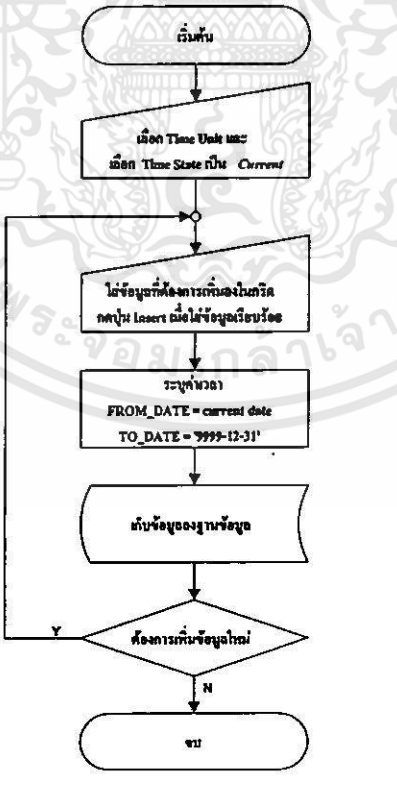
- การเพิ่มข้อมูล ผู้ใช้เลือกตัวเลือกเรดิโอ Time State และ Time Unit แล้วกรอกข้อมูลอื่นๆ เมื่อใส่ข้อมูลเรียบร้อยแล้วบันทึกโดยการกดปุ่ม Insert ระบบจะทำการเพิ่มแถวใหม่เข้าไปในตารางพร้อมกับใส่ค่าแอททริบิวต์ของ FROM_DATE เป็นวันเวลาปัจจุบัน และใส่ค่าแอททริบิวต์ของ TO_DATE เป็น '9999-12-31' (ตลอดกาล) ขั้นตอนการทำงานดังแสดงในรูปที่ 6.7

- การลบข้อมูล การลบข้อมูลในข้อมูลเชิงเวลาคือการแก้ไขข้อมูลโดยการระบุค่า TO_DATE ของข้อมูลที่ต้องการลบโดยตรวจสอบค่า TO_DATE เดิมจะต้องเป็น '9999-12-31' ให้เปลี่ยนเป็นค่าวันปัจจุบัน (current date) ขั้นตอนการทำงานดังแสดงในรูปที่ 6.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

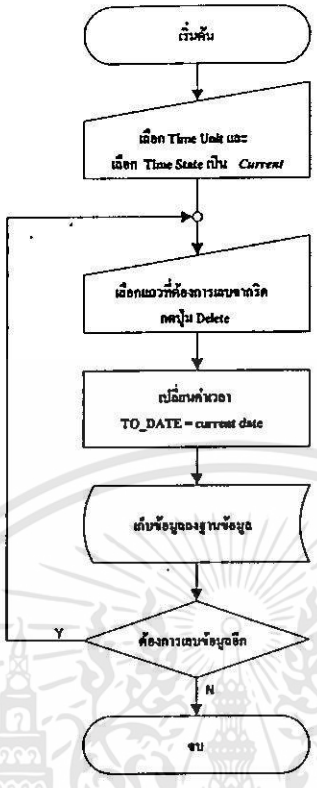


รูปที่ 6.6 ผังขั้นตอนการพัฒนาและการทำงานของส่วนจัดการข้อมูล



รูปที่ 6.7 ผังขั้นตอนการพัฒนาและการทำงานของการเพิ่มข้อมูลแบบปัจจุบัน (current)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



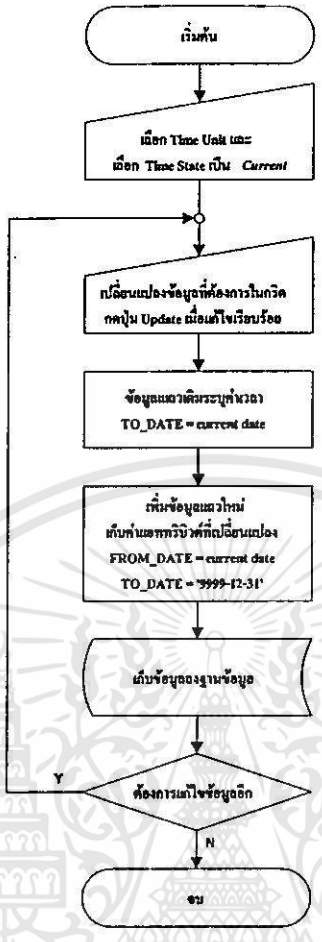
รูปที่ 6.8 ผังขั้นตอนการพัฒนาและการทำงานของการลบข้อมูลแบบปัจจุบัน (current)

- การแก้ไขข้อมูลเชิงเวลาเป็นการลบข้อมูลเชิงเวลาและการเพิ่มข้อมูลเชิงเวลา ในขณะเดียวกันโดยที่ข้อมูลเดิมจะต้องทำการลบเชิงเวลา หรือแก้ไขค่าของแอททริบิวต์ TO_DATE ให้เป็นวันที่ปัจจุบัน (current date) แล้วเพิ่มข้อมูลใหม่โดยให้ค่าของแอททริบิวต์ที่มีการเปลี่ยนแปลงปรากฏในข้อมูลที่เพิ่มเข้ามาแล้วให้ระบบระบุค่าเวลา FROM_DATE เป็นวันที่ปัจจุบัน (current date) และให้ค่า TO_DATE เป็น '9999-12-31' (forever) ขั้นตอนการทำงานดังแสดงในรูปที่ 6.9

- **การจัดการข้อมูลแบบเรียงลำดับ (Sequenced Modification)**

จากหัวข้อที่แล้วการเพิ่ม/ลบ/แก้ไขข้อมูลแบบปัจจุบันจะใช้วันที่จากปัจจุบันจนถึงตลอดกาล แต่การเพิ่ม/ลบ/แก้ไขข้อมูลแบบเรียงลำดับกระทำเป็นช่วงเวลาที่กำหนด ซึ่งช่วงเวลาดังกล่าวอาจเป็นช่วงเวลาในอดีต ช่วงเวลาในอนาคต หรือช่วงเวลาที่ตาบเกี่ยวกับช่วงเวลาปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.9 ผังขั้นตอนการพัฒนาและการทำงานของการแก้ไขข้อมูลแบบปัจจุบัน (current)

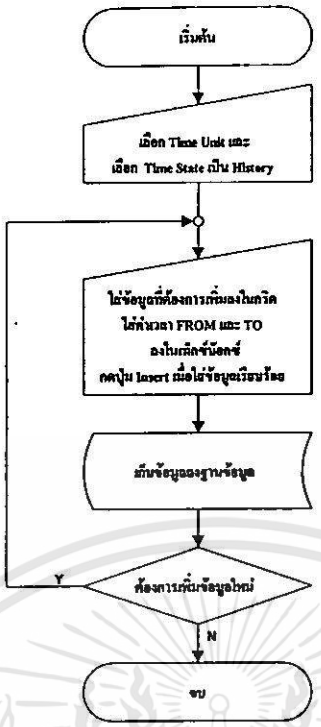
- การเพิ่มข้อมูลแบบเรียงลำดับ การเพิ่มข้อมูลแบบนี้เป็นการเพิ่มข้อมูลเชิงเวลาปกติ ซึ่งเมื่อเพิ่มข้อมูลจะมีการใส่ค่า FROM_DATE และ TO_DATE ซึ่งผู้ใช้กำหนดในเท็กซ์บ๊อกซ์ที่ระบบเตรียมไว้ให้ ซึ่งเท็กซ์บ๊อกซ์ดังกล่าวจะปรากฏไว้ให้เมื่อผู้ใช้ได้คลิ๊กเลือกตัวเลือกเรดิโอเป็น History เมื่อผู้ใช้ใส่ข้อมูลแอททริบิวต์อื่นๆเสร็จเรียบร้อยพร้อมทั้งใส่ค่า FROM_DATE และ TO_DATE จากนั้นกดปุ่ม Insert ระบบจะทำการบันทึกข้อมูลเพิ่มลงในฐานข้อมูล ขั้นตอนการทำงานดังแสดงในรูปที่ 6.10

- การลบข้อมูลแบบเรียงลำดับ การลบข้อมูลแบบนี้มี 4 กรณีด้วยกันขึ้นอยู่กับช่วงเวลาที่ข้อมูลเป็นจริง (period of validity : PV) กับช่วงเวลาที่ต้องการลบ (period of applicability : PA) ซึ่งผู้ใช้ระบุในเท็กซ์บ๊อกซ์ FROM และ TO

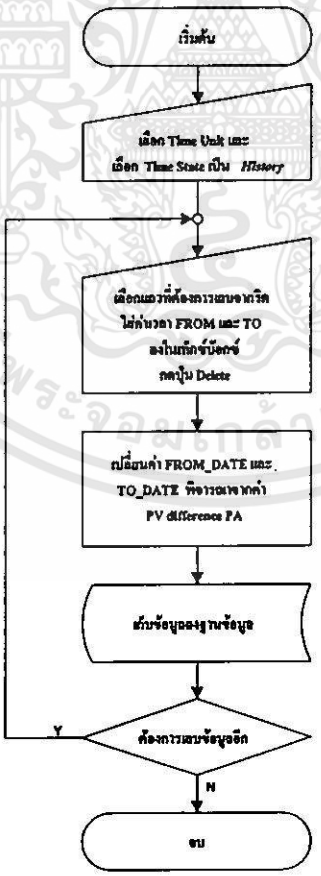
กรณีที่ 1 ค่า PV ครอบคลุม PA ผลที่ได้คือเก็บข้อมูลส่วนที่ไม่อยู่ใน PA ไว้

กรณีที่ 2 ค่า PV มีช่วงเวลาที่คาบเกี่ยวกับค่า PA โดยช่วงเวลา PV เกิดขึ้นก่อนดังนั้นจะต้องทำการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.10 ผังขั้นตอนการพัฒนาและการทำงานของการเพิ่มข้อมูลแบบเรียงลำดับ (sequence)



รูปที่ 6.11 ผังขั้นตอนการพัฒนาและการทำงานของการลบข้อมูลแบบเรียงลำดับ (sequence)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

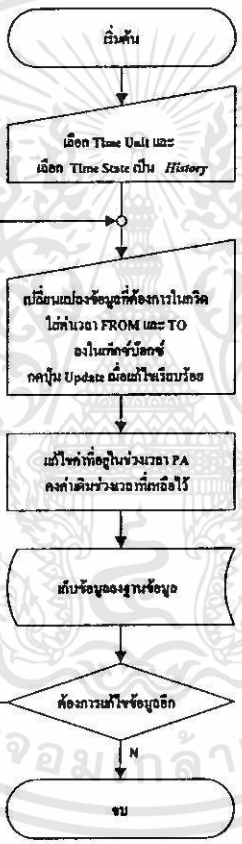
เก็บค่า PV ก่อนเกิดค่า PA

กรณีที่ 3 ช่วงเวลาคาบเกี่ยวกันแต่ค่า PA เกิดขึ้นก่อน ดังนั้นจะต้องเก็บค่า PV ส่วนที่เหลือหลังจากช่วงเวลาของ PA

กรณีที่ 4 ค่า PA ครอบคลุมค่า PV จะทำการลบทั้งหมด

ขั้นตอนการทำงานดังแสดงในรูปที่ 6.11

- การแก้ไขข้อมูลแบบเรียงลำดับ การแก้ไขข้อมูลแบบนี้มี 4 กรณี เช่นเดียวกับการลบข้อมูลแบบเรียงลำดับ (sequence) โดยทำการแก้ไขค่าในช่วงเวลา PA ซึ่งเป็นช่วงเวลาที่ต้องการแก้ไขข้อมูล ส่วนช่วงเวลาที่เหลือจะคงค่าเดิมไว้ ขั้นตอนการทำงานดังแสดงในรูปที่ 6.12



รูปที่ 6.12 ผังขั้นตอนการพัฒนาและการทำงานของการแก้ไขข้อมูลแบบเรียงลำดับ (sequence)

- **การจัดการข้อมูลแบบไม่เรียงลำดับ (Nonsequenced Modification)**

การเพิ่ม/ลบ/แก้ไขข้อมูลแบบไม่เรียงลำดับ กระทำกับค่าแอททริบิวต์ที่ใช้ในการระบุเวลาเสมือนกับเป็นค่าแอททริบิวต์ปกติ ดังนั้นผู้ใช้จะต้องกำหนดค่าFROM_DATE และ TO_DATE และตัวปฏิบัติการในการจัดการกับแอททริบิวต์ดังกล่าวเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

การทดสอบและประเมินผลระบบ TQBE

.ในบทนี้จะเป็นตัวอย่างการสอบถามข้อมูลเชิงเวลาโดยใช้เครื่องมือที่พัฒนาขึ้น

7.1 การทดสอบระบบ TQBE

พิจารณาจากตาราง LOT_LOC ซึ่งเป็นตารางเกี่ยวกับฝูงวัว มีการแสดงรายละเอียดของจำนวนวัวใน แต่ละคอกที่วัวแต่ละฝูงอยู่ การเคลื่อนย้ายจากคอกหนึ่งไปยังอีกคอกหนึ่ง และช่วงเวลาที่ค่าของข้อมูลเป็นจริง ซึ่งมีรายละเอียดโครงสร้างของตารางดังนี้ [11]

LOT_LOC (FDYD_ID , LOT_IDNYM , PEN_ID , HD_CNT , FROM_DATE ,TO_DATE)

ซึ่งค่าในตารางเป็นจริง ณ ช่วงเวลาขณะหนึ่งเท่านั้น โดยแต่ละแอตทริบิวต์มีความหมายดังนี้

FDYD_ID	แทน	หมายเลขลานให้อาหารสัตว์
LOT_ID_NUM	แทน	หมายเลขฝูงวัว
PEN_ID	แทน	หมายเลขคอก
HD_CNT	แทน	จำนวนวัว
FROM_DATE	แทน	valid time start
TO_DATE	แทน	valid time end

โดยมีหน่วยย่อยเวลา (time granulanty) เป็นวัน (day) และค่าวันที่เป็น "9999-12-31" หมายถึง ตลอดไป (forever) ดังแสดงในตารางที่ 7.1

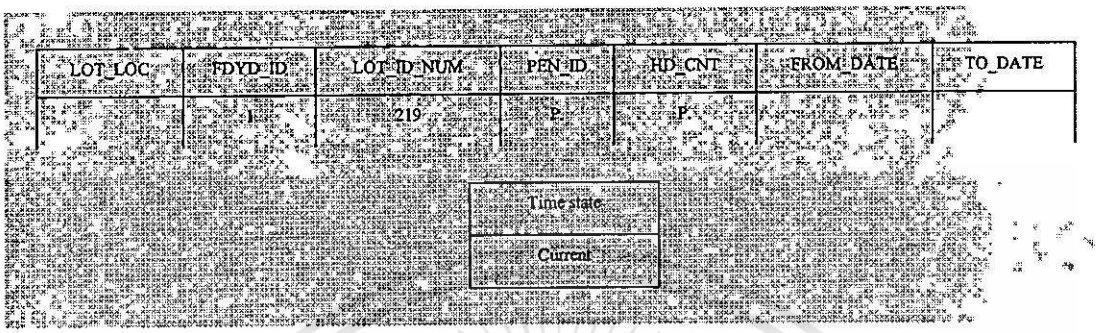
ตารางที่ 7.1 ตาราง LOT_LOC

FDYD_ID	LOT_ID_NUM	PEN_ID	HD_CNT	FROM_DATE	TO_DATE
1	137	1	17	1998-02-07	1998-02-18
1	219	1	43	1998-02-25	1998-03-01
1	219	1	20	1998-03-01	1998-03-14
1	219	2	23	1998-03-01	1998-03-14
1	219	2	43	1998-03-14	9999-12-31
1	374	1	14	1998-02-20	9999-12-31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 7.1 ให้หาจำนวนวัวจากฝูงหมายเลข 219 ในลานให้อาหารสัตว์หมายเลข 1 ในแต่ละคอก ณ ปัจจุบัน ?

โดยใช้ TQBE ต้องการข้อมูล ณ.ปัจจุบัน จึงระบุค่า Time State เป็น Current จะได้



เมื่อระบุ Time State เป็น Currently ระบบจะทำการระบุ TO_DATE = DATE '9999-12-31' ให้เองโดยอัตโนมัติ และผลลัพธ์ที่ได้ในรูปของ SQL คือ

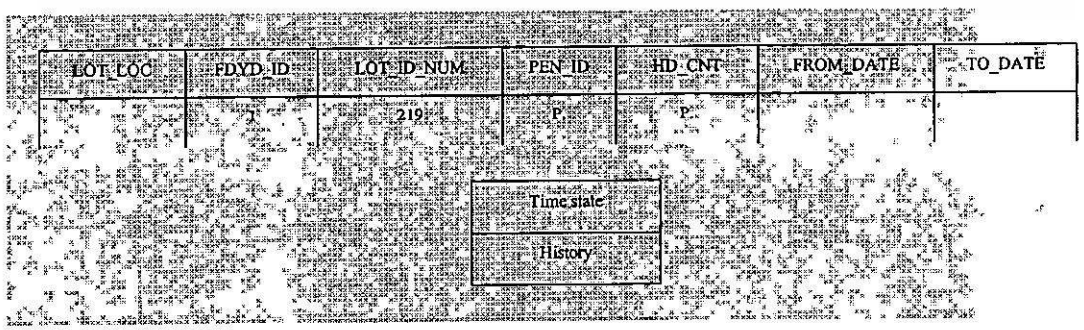
```
SELECT  PEN_ID , HD_CNT
FROM    LOT_LOC
WHERE   FDYD_ID = 1 AND LOT_ID_NUM = 219
        AND TO_DATE = DATE ' 9999-12-31'
```

ผลลัพธ์ที่ได้คือ

PEN_ID	HD_CNT
2	43

ตัวอย่างที่ 2. ให้หาประวัติ (history) ของจำนวนวัวจากฝูงหมายเลข 219 ในลานให้อาหารสัตว์หมายเลข 1 ในแต่ละคอก?

โดยใช้ TQBE ต้องการประวัติ จึงระบุค่า Time State เป็น History จะได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

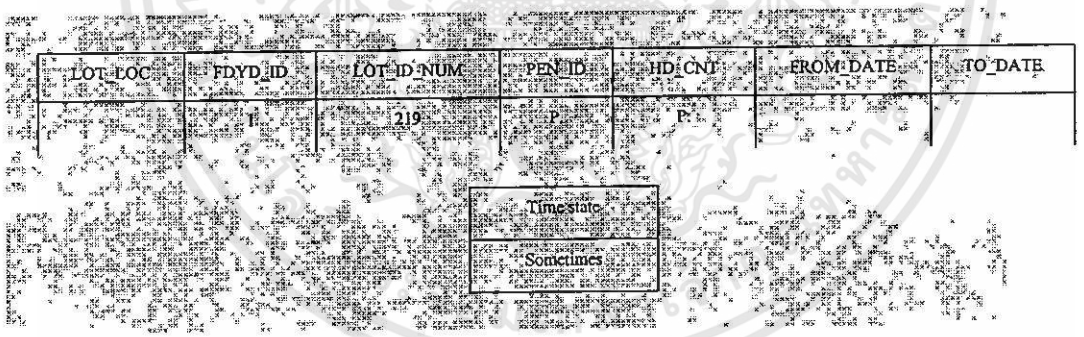
เมื่อระบุ Time State เป็น History ระบบจะทำการแสดงค่า FROM_DATE และ TO_DATE เพิ่มขึ้นอีก 2 คอลัมน์ให้เองโดยอัตโนมัติ และผลลัพธ์ที่ได้ในรูปของ SQL [11] คือ

```
SELECT   PEN_ID , HD_CNT , FROM_DATE , TO_DATE
FROM     LOT_LOC
WHERE    FDYD_ID = 1 AND LOT_ID_NUM = 219
```

ผลลัพธ์ที่ได้คือ

PEN_ID	HD_CNT	FROM_DATE	TO_DATE
1	43	1998-02-25	1998-03-01
1	20	1998-03-01	1998-03-14
2	23	1998-03-01	1998-03-14
2	43	1998-03-14	9999-12-31

ตัวอย่างที่ 3. ให้หาจำนวนวัวในแต่ละคอกจากฝูงหมายเลข 219 ในลานให้อาหารสัตว์หมายเลข 1 ในบางเวลา (sometimes) ?
โดยใช้ TQBE ต้องการข้อมูลบางเวลา จึงระบุค่า Time State เป็น Sometimes จะได้



เมื่อระบุ Time State เป็น Sometimes ระบบจะทำการแสดงค่า Nosequence และผลลัพธ์ที่ได้ในรูปของ SQL คือ

```
SELECT   PEN_ID , HD_CNT
FROM     LOT_LOC
WHERE    FDYD_ID = 1 AND LOT_ID_NUM = 219
```

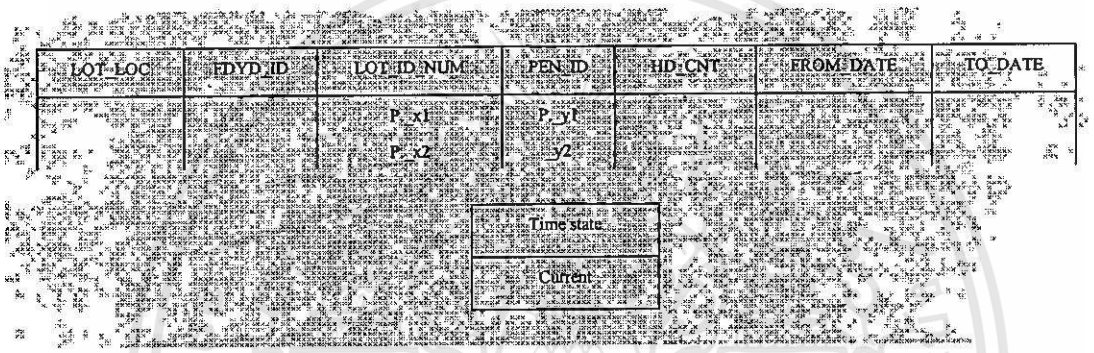
ผลลัพธ์ที่ได้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PEN_ID	HD_CNT
1	43
1	20
2	23
2	43

ตัวอย่างที่ 4. ให้หาฝูงวัวที่อยู่ในคอกเดียวกัน ณ.ปัจจุบัน ?

จากคำถามที่ว่าอยู่ในคอกเดียวกัน แสดงว่าเป็น self-join โดยใช้ TQBE จะได้



เมื่อระบุ Time State เป็น Current ระบบจะทำการระบุ TO_DATE = DATE '9999-12-31' ให้เองโดยอัตโนมัติ และผลลัพธ์ที่ได้ในรูปของ SQL คือ

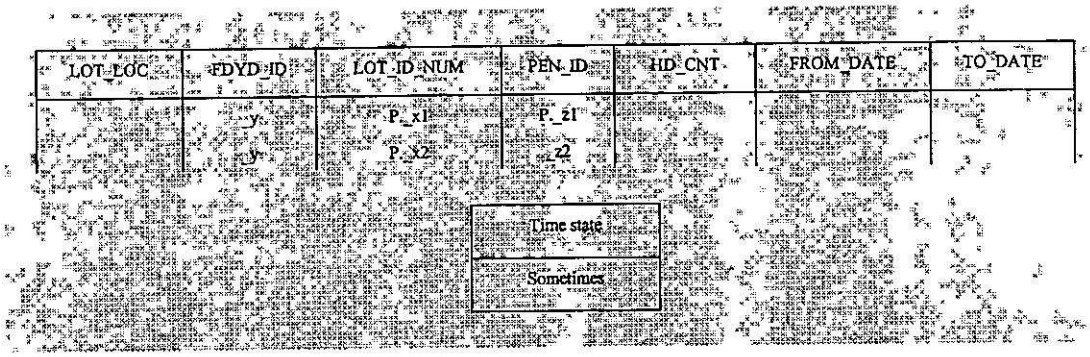
```
SELECT  L1.LOT_ID_NUM , L2.LOT_ID_NUM , L1.PEN_ID
FROM    LOT_LOC AS L1 , LOT_LOC AS L2
WHERE   L1.LOT_ID_NUM < L2.LOT_ID_NUM
        AND L1.PEN_ID = L2.PEN_ID
        AND L1.TO_DATE = DATE ' 9999-12-31 '
        AND L2.TO_DATE = DATE ' 9999-12-31 '
```

ผลลัพธ์คือ จะไม่มีข้อมูลในตารางสามารถแสดงให้ปรากฏ

ตัวอย่างที่ 5. ให้หาฝูงวัวที่อยู่ในคอกเดียวกัน ปรากฏในเวลาที่ต่างกัน ?

จากคำถามที่ว่าอยู่ในคอกเดียวกัน แสดงว่าเป็น self-join โดยใช้ TQBE จะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เมื่อระบุ Time State เป็น Sometimes ระบบจะทำการแสดงค่า Nonsequence และและผลลัพธ์ที่ได้ในรูปของ SQL คือ

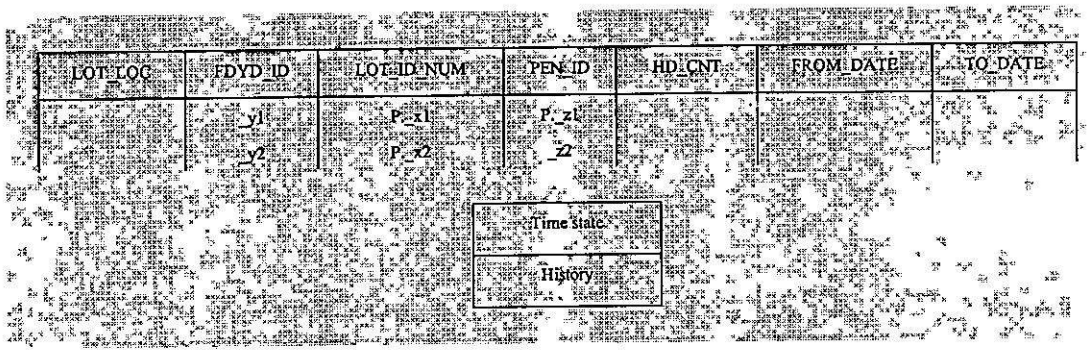
```
SELECT  L1.LOT_ID_NUM , L2.LOT_ID_NUM , L1.PEN_ID
FROM    LOT_LOC AS L1 , LOT_LOC AS L2
WHERE   L1.LOT_ID_NUM < L2.LOT_ID_NUM
        AND FDYD_ID = L2.FDYD_ID
        AND L1.PEN_ID = L2.PEN_ID
```

ผลลัพธ์ที่ได้คือ

L1	L2	PEN_ID
137	219	1
137	219	1
137	374	1
219	374	1
219	374	1

ตัวอย่างที่ 6. ให้หาประวัติของฝูงวัวที่อยู่ในคอกเดียวกัน ?

จากคำถามที่ว่าอยู่ในคอกเดียวกัน แสดงว่าเป็น self-join โดยใช้ TQBE จะได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อระบุ Time State เป็น History ระบบจะทำการแสดงค่า FROM_DATE และ TO_DATE เพิ่มขึ้นอีก 2 คอลัมน์ให้เองโดยอัตโนมัติ และผลลัพธ์ที่ได้ในรูปของ SQL [1] คือ

```
SELECT L1.LOT_ID_NUM , L2.LOT_ID_NUM , L1.PEN_ID , L1.FROM_DATE ,
       L1.TO_DATE
FROM   LOT_LOC AS L1 , LOT_LOC AS L2
WHERE L1.LOT_ID_NUM < L2.LOT_ID_NUM
       AND L1.FDYD_ID = L2.FDYD_ID
       AND L1.PEN_ID = L2.PEN_ID
       AND L2.FROM_DATE <= L1.FROM_DATE
       AND L1.TO_DATE <= L2.TO_DATE
UNION
SELECT L1.LOT_ID_NUM , L2.LOT_ID_NUM , L1.PEN_ID , L1.FROM_DATE ,
       L2.TO_DATE
FROM   LOT_LOC AS L1 , LOT_LOC AS L2
       WHERE L1.LOT_ID_NUM < L2.LOT_ID_NUM
       AND L1.FDYD_ID = L2.FDYD_ID
       AND L1.PEN_ID = L2.PEN_ID
       AND L1.FROM_DATE > L2.FROM_DATE
       AND L2.TO_DATE < L1.TO_DATE
       AND L1.FROM_DATE < L2.TO_DATE
UNION
SELECT L1.LOT_ID_NUM , L2.LOT_ID_NUM , L1.PEN_ID , L2.FROM_DATE ,
       L1.TO_DATE
FROM   LOT_LOC AS L1 , LOT_LOC AS L2
WHERE L1.LOT_ID_NUM < L2.LOT_ID_NUM
       AND L1.FDYD_ID = L2.FDYD_ID
       AND L1.PEN_ID = L2.PEN_ID
       AND L2.FROM_DATE > L1.FROM_DATE
       AND L1.TO_DATE < L2.TO_DATE
       AND L2.FROM_DATE < L1.TO_DATE
UNION
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SELECT L1.LOT_ID_NUM , L2.LOT_ID_NUM , L1.PEN_ID , L2.FROM_DATE ,
       L2.TO_DATE
FROM   LOT_LOC AS L1 , LOT_LOC AS L2
WHERE  L1.LOT_ID_NUM < L2.LOT_ID_NUM
       AND L1.FDYD_ID = L2.FDYD_ID
       AND L1.PEN_ID = L2.PEN_ID
       AND L2.FROM_DATE >= L1.FROM_DATE
       AND L2.TO_DATE <= L1.TO_DATE
```

และสามารถแปลให้เป็นภาษา SQL-92 ได้ดังนี้

```
SELECT L1.LOT_ID_NUM , L2.LOT_ID_NUM , L1.PEN_ID
       CASE WHEN L1.FROM_DATE > L2.FROM_DATE
            THEN L1.FROM_DATE
            ELSE L2.FROM_DATE END.
       CASE WHEN L1.TO_DATE > L2.TO_DATE
            THEN L2.TO_DATE
            ELSE L1.TO_DATE END
FROM   LOT_LOC AS L1.LOT_LOC AS L2
WHERE L1.LOT_ID_NUM < L2.LOT_ID_NUM
       AND L1.FDYD_ID = L2.FDYD_ID
       AND L1.PEN_ID = L2.PEN_ID
       AND ( CASE WHEN L1.FROM_DATE > L2.FROM_DATE
            THEN L1.FROM_DATE
            ELSE L2.FROM_DATE END ) <
                 ( CASE WHEN L1.TO_DATE > L2.TO_DATE
                      THEN L2.TO_DATE
                      ELSE L1.TO_DATE END )
```

ผลลัพธ์ที่ได้คือ

LOT_ID_NUM	LOT_ID_NUM	PEN_ID	FROM_DATE	TO_DATE
219	374	1	1998-02-25	1998-03-01
219	374	1	1998-03-01	1998-03-14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 เปรียบเทียบระบบ TQBE กับภาษาอื่นๆ

เปรียบเทียบ TQBE กับภาษาสอบถามเชิงเวลา 3 ชนิด ได้แก่ TQUEL , HTQUEL และ HQUEL

TQUEL [12] เป็นส่วนขยายของภาษาสอบถาม QUELใน INGRES [13] ภาษา TQUEL มีการสร้างคำใหม่คือ valid , when และ as of อีกทั้งยังอนุญาตให้มีตัวปฏิบัติการทางเวลาที่ใช้คือ overlap , extend และ precede ตัวปฏิบัติการ overlap ใน TQUEL จะเหมือนกันกับใน TQBE มี when clause เพิ่มจาก where clause เพื่อใช้เป็นกฎเกณฑ์ของปริมาณเชิงเวลา TQUEL อนุญาตให้ช่วงเวลาที่แตกต่างกันในการเพิ่มและลบข้อมูล อย่างไรก็ตาม valid clause ไม่สามารถกำหนดค่า Difference และ Union ของช่วงเวลาได้ เมื่อเวลาไม่ต่อเนื่องกัน ฟังก์ชันรวม (aggregate function) สามารถใช้ได้ใน TQUEL และ TQBE สามารถใช้ฟังก์ชันรวมได้เช่นกัน การทำ optimize ในขบวนการสอบถามข้อมูลถูกจำกัด เนื่องจากไม่มีพีชคณิตสัมพันธ์ที่เกี่ยวข้อง

HTQUEL เป็นส่วนย่อยที่ขยายมาจาก QUEL [14] ซึ่ง HTQUEL สามารถใช้ predicate ทางเวลาใน WHEN clause เพื่อกำหนดเวลาที่อ้างอิงถึงในแบบ snapshot มีฟังก์ชั่นที่สำคัญในการเข้าถึงจุดเวลาหรือช่วงเวลาในโดเมนของเวลา HTQUEL มีคำสั่งที่ใช้สำหรับจัดการกับโดเมนของเวลา HTQUEL สามารถใช้ Intersection , Union และ Difference กับโดเมนของเวลาได้ เวลาที่กำหนดในผลลัพธ์ถูกซ่อนไว้ ซึ่งต่างจากใน TQUEL และ TQBE ฟังก์ชั่นรวมยังไม่สามารถใช้ได้ใน HTQUEL ส่วนการใช้พีชคณิตสัมพันธ์มีความสมบูรณ์ดี

HQUEL เป็นส่วนขยายของภาษา QUEL [15] อีกภาษาหนึ่ง ซึ่งมีพื้นฐานมาจากโมเดลของข้อมูลเชิงประวัติซึ่งสามารถใช้ตัวปฏิบัติการทางเซต และ ตัวปฏิบัติการเปรียบเทียบกับช่วงเวลาที่อ้างถึงได้ ดังนั้นจึงสามารถใช้ Intersection , Union และ Difference ใน WHERE clause เป็นตัวปฏิบัติการได้ ฟังก์ชันรวมขึ้นอยู่กับพีชคณิตเชิงประวัติ และได้ทำการสรุปการเปรียบเทียบ ดังแสดงในตารางที่ 2

ตารางที่ 7.2 เปรียบเทียบคุณสมบัติของภาษา TQBE กับภาษาอื่น

Language	TQBE	HQUEL	HTQUEL	TQUEL
Temporal Operators	X	X	X	X
Aggregate Operation	X	X	-	X
Noncontiguous Interval	X	X	X	X
Implementation	X	-	-	X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TQBE เป็นระบบจัดการและภาษาสอบถามข้อมูลสำหรับฐานข้อมูลเชิงเวลา โดยใช้การสอบถามตามแนวคิดของ QBE (Query-By-Example) ประยุกต์ใช้กับข้อมูลเชิงเวลาเพื่อให้ได้โครงสร้าง รวมทั้งได้สร้างความหมายของคำ และ รูปแบบที่ใช้กับข้อมูลเชิงเวลาได้เหมาะสม รวมทั้งได้สร้างส่วนที่ใช้ในการแปลงภาษา TQBE ให้เป็นภาษาสอบถามเชิงโครงสร้าง (SQL) ซึ่งยุ่งยากและสลับซับซ้อนเพื่อสนับสนุนข้อมูลเชิงเวลาให้เองโดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8
สรุปและแนวทางการวิจัย

8.1 สรุป

จากการนำข้อมูลและข่าวสารมาใช้งานในด้านต่างๆมากมาย ทำให้ระบบจัดการฐานข้อมูลถูกนำมาใช้งานมากขึ้นตามไปด้วย ทั้งนี้เพื่อความสะดวกรวดเร็วในการจัดการและสอบถามข้อมูล จากสาเหตุที่มีการใช้กันอย่างแพร่หลายนี้ ระบบจัดการฐานข้อมูลจึงได้รับการพัฒนาอย่างต่อเนื่องตลอดมาจากระบบจัดการฐานข้อมูลแบบไม่อิงเวลา จนถึงระบบจัดการฐานข้อมูลที่เกี่ยวเนื่องกับเวลา โดยมีมิติของเวลาเข้ามาเกี่ยวข้องซึ่งเรียกว่า ระบบจัดการฐานข้อมูลเชิงเวลา

ระบบจัดการฐานข้อมูลเชิงเวลา TQBE ได้พัฒนาขึ้นเพื่อตอบสนองความต้องการของฐานข้อมูลและโปรแกรมประยุกต์เดิมที่ต้องการการสนับสนุนข้อมูลเชิงเวลา โดยมีรูปแบบโครงสร้างเป็นกราฟฟิกตามแนวคิดของ QBE ซึ่งสามารถสนับสนุนข้อมูลต่างๆที่เกี่ยวเนื่องกับเวลา และเงื่อนไขต่างๆที่เกี่ยวข้องกับเวลา

โมเดลของข้อมูลที่ใช้ในระบบ TQBE นี้เป็นแบบเชิงเวลา ซึ่งมีการระบุเวลาแบบเป็นช่วงเวลาที่กระทำกับแถว (tuple timestamping) และมีการสร้างประเภทของเวลาใหม่คือ Current, History และ Sometime เพื่อสนับสนุนข้อมูลเวลาในรูปแบบ Current, Sequence และ Nonsequence ตามลำดับ

การสอบถามข้อมูลเชิงเวลาแบบใช้ตัวอย่าง (Temporal Query-By-Example : TQBE) เป็นภาษาสอบถามข้อมูลและจัดการข้อมูล ที่ถูกออกแบบมาสำหรับระบบจัดการฐานข้อมูลเชิงเวลา โดยมีรูปแบบโครงสร้างเป็นแบบกราฟฟิก และ ใช้แนวคิดการสอบถามข้อมูลแบบใช้ตัวอย่าง (QBE) เป็นการผสมผสานกันระหว่าง QBE และกฎบังคับความถูกต้องทางเวลา เพื่อตอบสนองกับการใช้เวลาและการจัดการเวลา ในฐานข้อมูลเชิงเวลาให้เป็นไปได้อย่างถูกต้อง ประกอบด้วยส่วนสำคัญดังนี้

โครงร่างความสัมพันธ์ (Relation Skeleton) ซึ่งจะแสดงแบบแผนฐานข้อมูล (database schema) ประกอบด้วย คอลัมน์ของแต่ละแอตทริบิวต์ของความสัมพันธ์ ชื่อตาราง ชื่อแอตทริบิวต์ รวมทั้งแถวว่างสำหรับกำหนดแถวตัวอย่าง

ส่วนเงื่อนไข (Condition Box) เป็นตารางโครงร่าง ซึ่งประกอบด้วยคอลัมน์เดียวแต่หลายแถว แต่ละแถวประกอบด้วย บูลลีน (Boolean), ฟังก์ชันรวม (Aggregate function), ตัวปฏิบัติการทางเซต, ตัวปฏิบัติการเปรียบเทียบทางเซต, ตัวปฏิบัติการทางตรรกะ และตัวปฏิบัติการทางคณิตศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนช่วงเวลา (Time Range Box) เป็นตารางโครงร่าง ถูกใช้เพื่อการกำหนดความสัมพันธ์ของช่วงเวลาของตัวแปร โดยใช้ตัวปฏิบัติทางคณิตศาสตร์ หรือตัวปฏิบัติการทางเวลา 13 ตัว ของ Allen คือ before , after , meets , met-by , overlaps , overlapped-by , during , includes , starts , started-by , finishes , finished-by และ equal

ส่วนหน่วยเวลา (Time Granularity Box) เป็นตารางโครงร่างเพื่อกำหนดหน่วยย่อยในการระบุเวลา (timestamp)

ส่วนประเภทเวลา (Time State Box) เป็น ตารางโครงร่างเพื่อการสอบถามข้อมูล โดยระบุว่าเป็นการสอบถามข้อมูล ณ เวลาปัจจุบัน (currently) หรือแบบเรียงลำดับ (sequence) หรือ สอบถามข้อมูลแบบไม่เรียงลำดับ (nonsequence)

สำหรับสถาปัตยกรรมของระบบจัดการฐานข้อมูลเชิงเวลา TQBE ประกอบด้วยส่วนทำงานหลักๆ 7 ส่วนคือ ส่วนติดต่อกับผู้ใช้ หน่วยปรับปรุงแบบแผนโครงสร้างตาราง หน่วยระบุเวลาเข้ากับตารางฐานข้อมูล หน่วยควบคุมความถูกต้องของข้อมูล หน่วยสร้างภาษาสอบถามเชิงโครงสร้าง ส่วนแสดงผลลัพธ์ และหน่วยแปลงฐานข้อมูลแบบไม่อิงเวลาเป็นฐานข้อมูลเชิงเวลา

8.2 ข้อเสนอแนะในการวิจัยต่อไป

สำหรับขอบเขตของงานวิจัยนี้ ได้ตั้งวัตถุประสงค์ของการพัฒนาระบบ TQBE ไว้เพียงการสอบถามข้อมูลและจัดการข้อมูลที่มีการระบุเวลาเฉพาะค่า valid time จึงทราบเฉพาะประวัติของข้อมูลเท่านั้น ยังไม่มีสามารถทราบสถานะใดๆของข้อมูลและการเปลี่ยนแปลงฐานข้อมูล ดังนั้นแนวทางในการทำวิจัยครั้งต่อไปควรจะมีเพิ่มการระบุเวลาที่เป็น transaction time เข้ากับโครงสร้างของตารางข้อมูลด้วย ซึ่งต้องอยู่บนพื้นฐานของกฎบังคับความถูกต้องของข้อมูลเชิงเวลาเป็นสำคัญ

ระบบ TQBE ที่ได้พัฒนาขึ้นนี้ยังขาดความสามารถอีกหลายอย่างเช่น หน่วยสร้างฟอร์ม หน่วยสร้างรายงาน หน่วยสร้างเมนู และหน่วยสร้างโปรแกรมประยุกต์ ดังนั้นในการวิจัยและพัฒนาในครั้งต่อไปควรเพิ่มการพัฒนาในส่วนดังกล่าวให้ระบบมีฟังก์ชันการทำงานและประสิทธิภาพมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] E.F.Codd. "Relational Completeness of Data Base Sublanguage." Data Base System,Courant Computer Symposia Series, Vol 6, Prentice-Hall, EngleWood Cliffs, NJ, 1972.

[2] R. Snodgrass and I. Ahn. "A Temporal Databases." IEEE Computer, 1986, 19(9), pp. 35-42.

[3] S.K.Gadia and J.H.Vaishnav. **A Query Language for a Homogeneous Temporal Database.** In Proseeding of the International Conferrence on Principle of Database System, 1985 , pages 51-56.

[4] A. Tansel, J. Clifford, S. Gadia, S. Jajodia, A. Segev, and R. Snodgrass. **Temporal Database: Theory, design, and Implementation.** Benjamin/Cummings Publiching Company, 1993.

[5] Allen, J.F. "Maintaining knowledge about temporal intervals." Communications of the ACM, 26(11):832-843, 1983.

[6] C.S. Jensen, J. Clifford, R. Elmasri, S.K. Gadia, P.Hayes and S. Jajodia [eds]. "A Glossary of Temporal Database Concepts." ACM SIGMOD Record, March, 1994. 23(1) : 52-64.

[7] R. Snodgrass and I.Ahn. "A Taxonomy of Time in Databases." In S. Navathe, editor, Proceedings of the ACM SIGMOD International Conference on Management of Data, 1985. pp. 236-246.

[8] C.Vassilakis, P.Geogiadis and A.Sotiropoulou. "A Comparative Study of Temporal DBMS Architectures." IEEE Transaction on Knowledge and Data Engineering,1996.

[9] J. Bair, M. Bohlen, C. Jensen and R.Snodgrass. "Notion of Upward Compatibility of Temporal Query Languages." A TimeCenter Technical Report, April 2, 1997, TR-6.

[10] M.M. Zloof. "Query-by-Example : a data base language." IBM System J., 16(4):342-343, 1977.

[11] R. Snodgrass. "Managing Temporal Data A Five-Part Series." A TimeCenter Technical Report, TR-28 : 1-21, September 3, 1998.

[12] R.Snodgrass. "A temporal query language." ACM Trans. Database Syst., vol. 12, no. 2, June 1987.

[13] M. Stonebraker, E. Wong, P. Kreps, and G. Held. "The design and implementation of INGRES." ACM Trans. Database Syst., vol. 1. No. 3, Sept, 1976.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[14] S.K. Gadia and J.H. Vaishnav. "A query language for a homogeneous temporal database." in Proc. ACM PODS Conf., 1985.

[15] A.U. Tansel and M.E. Arkun. "HQUEL:A historical query language." in Proc. 3 rd Int. Workshop Statistical Sci. Database Management, Luxembourg, 1986.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

คู่มือการใช้งานระบบ TQBE

ระบบจัดการและสอบถามข้อมูลเชิงเวลา TQBE ซึ่งได้พัฒนาขึ้นนี้เป็นระบบซึ่งช่วยให้ผู้ใช้สามารถสอบถามข้อมูลเชิงเวลา อีกทั้งยังสามารถทำการเพิ่มข้อมูล ลบข้อมูล และแก้ไขข้อมูลที่เกี่ยวเนื่องกับเวลาได้ซึ่งรายละเอียดการใช้งาน ระบบจะได้อธิบายในลำดับต่อไป

1. การเข้าสู่ระบบ

เริ่มต้นเข้าระบบวินโดว์ คลิ๊กที่ไอคอน TQBE เพื่อเรียกระบบ ระบบ TQBE จะแสดงบนจอภาพพร้อมที่จะใช้งาน

2. รายละเอียดของระบบ

ระบบ TQBE ประกอบด้วยรายการหลัก 3 ส่วนดังต่อไปนี้คือ การสอบถามข้อมูลเชิงเวลา การจัดการข้อมูลเชิงเวลา และการแปลงฐานข้อมูล

2.1 การสอบถามข้อมูลเชิงเวลา

ในส่วนนี้ประกอบด้วยแท็บ 6 ส่วนหลักดังนี้คือ ส่วน Table , Column , Condition , Time Range , SQL และOutput

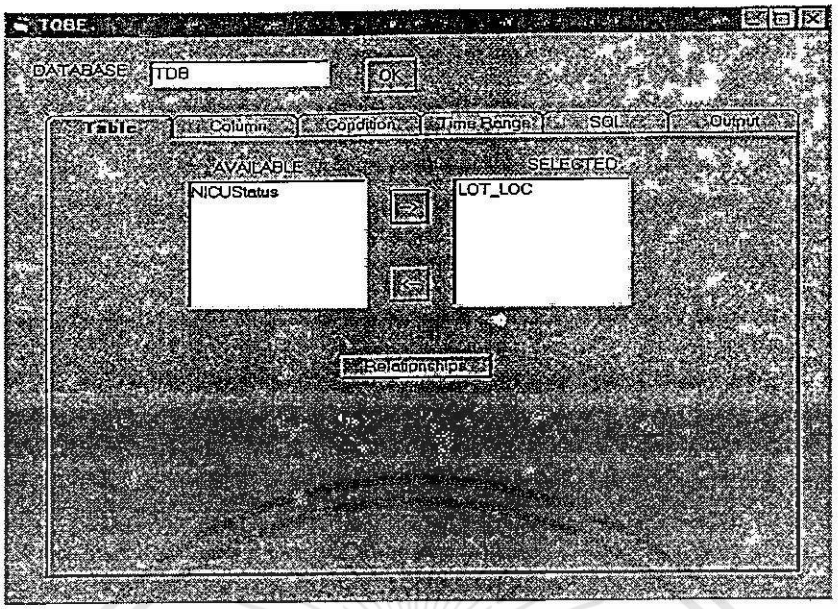
- แท็บ Table

  เมื่อผู้ใช้ใส่ชื่อฐานข้อมูลเรียบร้อยแล้ว กดปุ่ม OK จะปรากฏชื่อตารางในส่วนของ Available เลือกมาจากตารางที่ต้องการ โดยการกดลูกศรขวา หากมีความสัมพันธ์ระหว่างตารางให้คลิ๊กปุ่ม Relationship แล้วเลือกคอลัมน์ที่มีความสัมพันธ์ระหว่างกัน ดังแสดงในรูปที่ ก.1

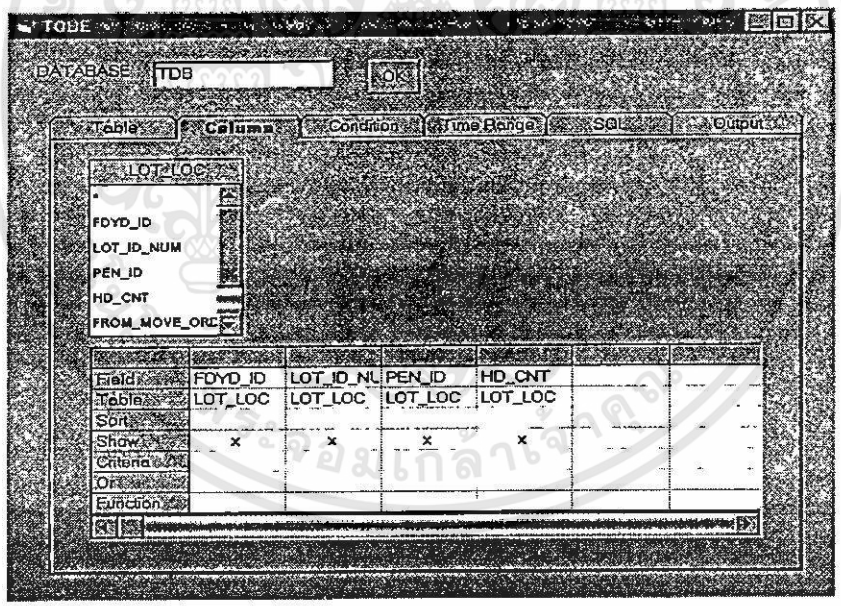
- แท็บ Column

  เลือกฟิลด์ที่ต้องการแสดงโดย double-click ในตารางที่ปรากฏ ผู้ใช้สามารถเลือกการเรียงลำดับข้อมูลจากช่อง Sort และสามารถใส่ค่าเงื่อนไขในช่อง Criteria และ Or อีกทั้งสามารถเลือกฟังก์ชันรวมในช่อง Function ดังแสดงในรูปที่ ก.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.1 หน้าจอการสอบถามข้อมูลเชิงเวลาส่วน Table

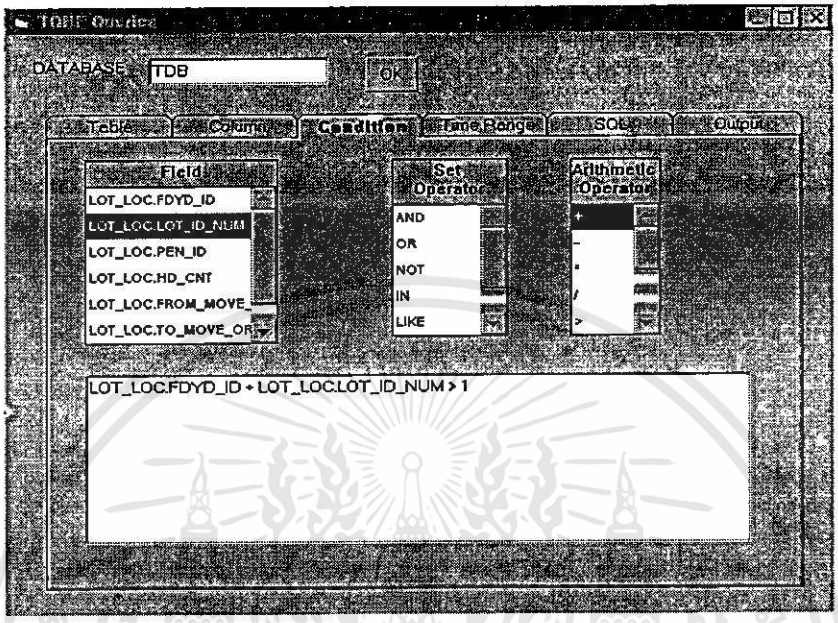


รูปที่ ก.2 หน้าจอการสอบถามข้อมูลเชิงเวลาส่วน Column

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แท็บ Condition

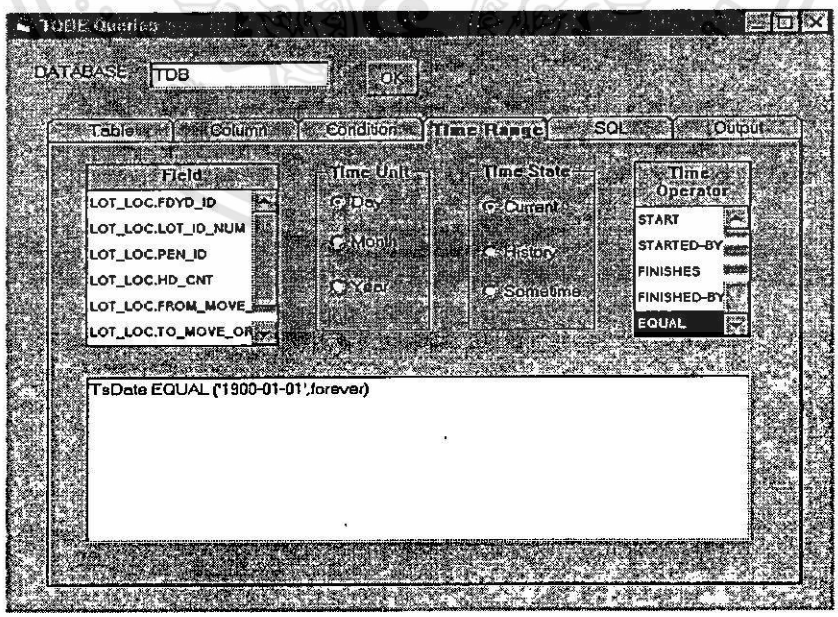
ผู้ใช้สามารถกำหนดเงื่อนไขทางเซ็ทและทางคณิตศาสตร์ โดยการเลือกข้อมูลจาก บ๊อกซ์ Field, บ๊อกซ์ Set Operator และบ๊อกซ์ Arithmetic Operator ดังแสดงในรูปที่ ก.3



รูปที่ ก.3 หน้าจอการสอบถามข้อมูลเชิงเวลาส่วน Condition

- แท็บ Time Range

ผู้ใช้สามารถกำหนดเงื่อนไขทางเวลาได้ โดยการเลือกจากบ๊อกซ์ Field, บ๊อก Time State, Time Granularity และ Time Operator ดังแสดงในรูปที่ ก.4

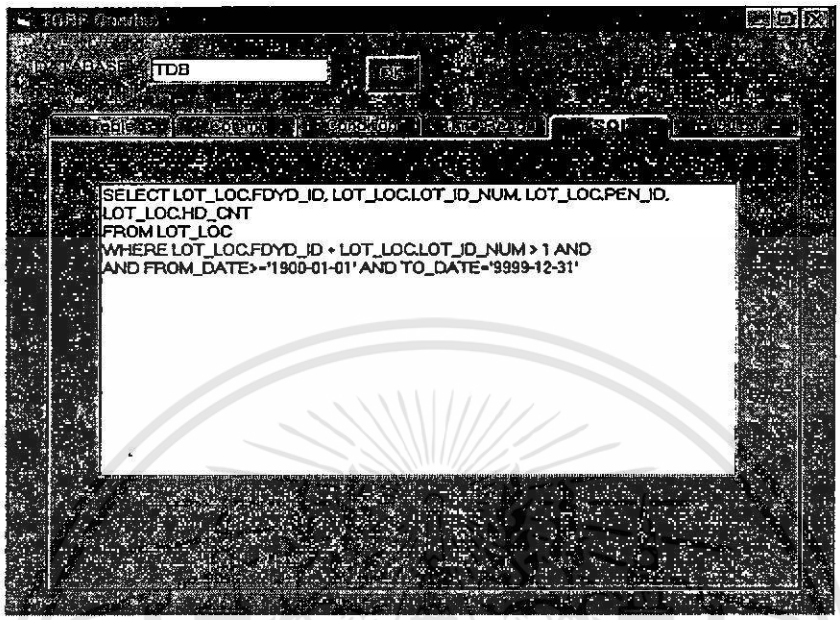


รูปที่ ก.4 หน้าจอการสอบถามข้อมูลเชิงเวลาส่วน Time Range

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แท็บ SQL

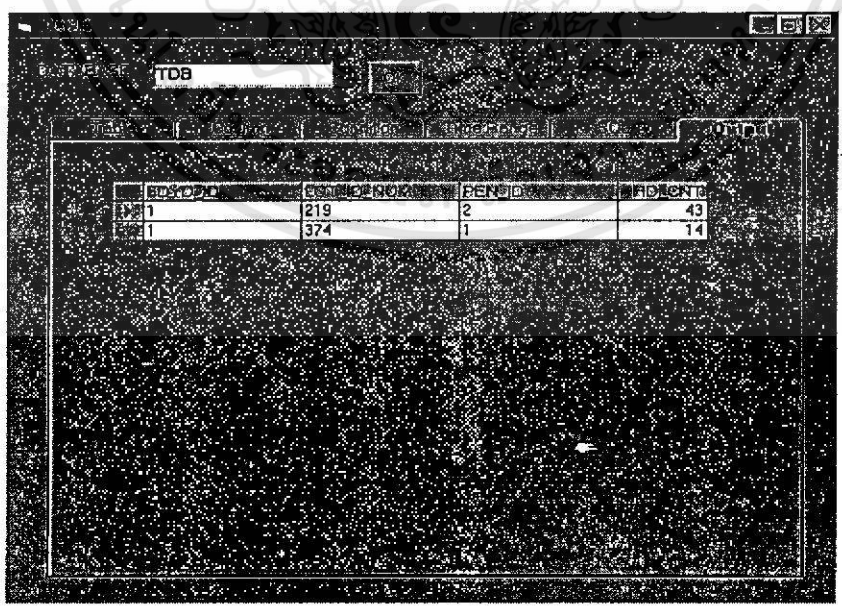
  เมื่อกดแท็บนี้ระบบจะทำการสร้างภาษาสอบถามเชิงโครงสร้างดังแสดงในรูปที่ ก.5



รูปที่ ก.5 หน้าจอการสอบถามข้อมูลเชิงเวลาส่วน SQL

- แท็บ Output

  เมื่อกดแท็บนี้ระบบจะแสดงผลที่ได้จากการสอบถสมข้อมูลดังแสดงในรูปที่ ก.6

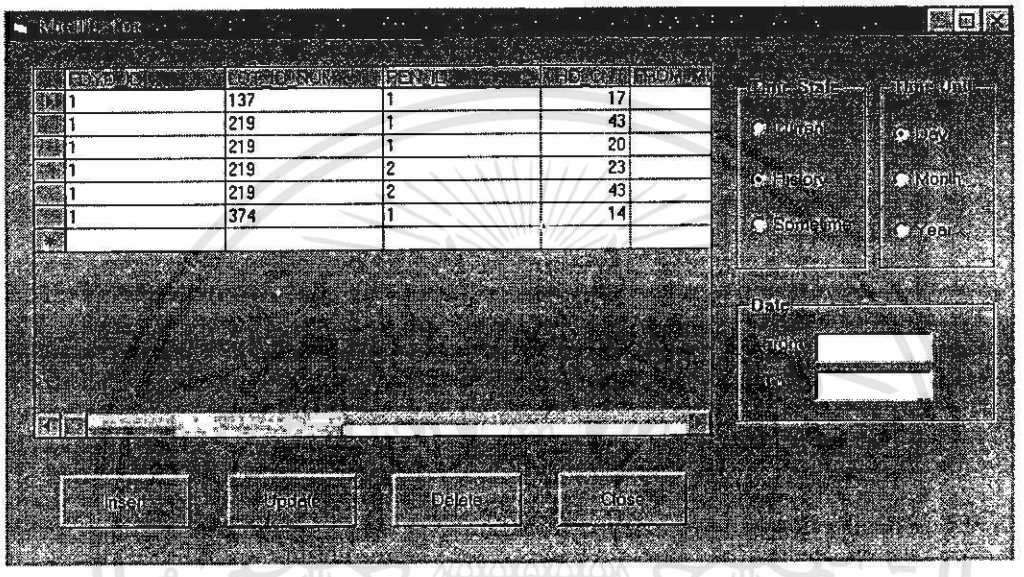


รูปที่ ก.6 หน้าจอการสอบถามข้อมูลเชิงเวลาส่วน Output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การจัดการข้อมูลเชิงเวลา

ในส่วนนี้ผู้ใช้สามารถทำการเพิ่มข้อมูล ลบข้อมูล และแก้ไขข้อมูลเชิงเวลาได้ โดยการระบุค่า Time Unit และเลือกค่า Time State ว่าตัวการเพิ่ม/ลบ/แก้ไขข้อมูลในแบบ Current, History, หรือ Sometime ซึ่งจะแทนข้อมูลในแบบ Current, Sequence และNonsequence ตามลำดับ เมื่อคลิ๊กเลือก Time State แล้วระบบจะปรากฏเท็กซ์บ๊อกซ์สำหรับใส่วันที่ FROM_DATE และ TO_DATE จากนั้นกดปุ่มคำสั่ง Insert, Update หรือ Delete ตามต้องการ ดังแสดงในรูปที่ ก.7



รูปที่ ก.7 หน้าจอการจัดการข้อมูลเชิงเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

นายธนัท อร่ามรัศมีวาณิชย์เกิดเมื่อวันที่ 24 เมษายน 2507 ที่กรุงเทพฯ สำเร็จการศึกษาวิทยาศาสตร์บัณฑิต(สถิติประยุกต์) จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง คณะครุศาสตร์อุตสาหกรรมและวิทยาศาสตร์ ปีการศึกษา 2530 เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เมื่อปีพ.ศ.2536

ปัจจุบันทำงานอยู่ที่ บริษัทสยามนิปปอนเทรดดิ้งจำกัด ในตำแหน่งผู้จัดการทั่วไป

บทความวิชาการที่ได้รับการตีพิมพ์

1. ธนัท อร่ามรัศมีวาณิชย์, ศุภมิตร จิตตะยโศธร. "การพัฒนาระบบ QBE บนฐานข้อมูลเชิงเวลา." วิศวกรรมสาร มก. ฉบับที่ 40 ปีที่ 14 เดือน เมษายน – กรกฎาคม 2543.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้