

การพัฒนาาระบบสารสนเทศเชิงเวลาโดยใช้ระบบจัดการฐานข้อมูลเชิงเวลา

**THE DEVELOPMENT OF A TEMPORAL INFORMATION SYSTEM
USING A TEMPORAL DBMS**



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2557

KMITL-2014-EN-M-070-058

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**THE DEVELOPMENT OF A TEMPORAL INFORMATION SYSTEM
USING A TEMPORAL DBMS**

SITTI RUGTANOM

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN COMPUTER ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2014
KMITL-2014-EN-M-070-058**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2014





FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

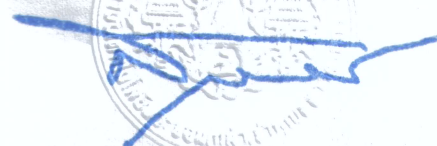
หัวข้อวิทยานิพนธ์ การพัฒนาระบบสารสนเทศเชิงเวลาโดยใช้ระบบจัดการฐานข้อมูลเชิงเวลา
Thesis Title The Development of a Temporal Information System Using a Temporal DBMS
นักศึกษา นายสิทธิ รักถนอม
รหัสประจำตัว 52611002
ปริญญา วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์ รศ.ดร.ศุภมิตร จิตตะยโสธร
หมายเลขวิทยานิพนธ์ KMITL-2014-EN-M-070-058

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.บุญธีร์	เครือตราชู	
ดร.วัชร	ฉัตรวิริยะ	
รศ.ดร.นิพนธ์	เจริญกิจการ	
รศ.ดร.ศุภมิตร	จิตตะยโสธร	

วัน / เดือน / ปี ที่สอบ วันพุธที่ 14 พฤษภาคม พ.ศ. 2557 เวลา 10.00-12.00 น.
สถานที่สอบ ณ อาคาร ECC ห้อง ECC-810

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์ รับรองแล้ว



(ศาสตราจารย์ ดร.สุชัชวีร์ สุวรรณสวัสดิ์)

คณบดี คณะวิศวกรรมศาสตร์

วันที่ 14 พฤษภาคม พ.ศ. 2557

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปเผยแพร่ภายนอกการดำเนินการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การพัฒนาระบบสารสนเทศเชิงเวลาโดยใช้ระบบจัดการฐานข้อมูลเชิงเวลา
นักศึกษา	นายสิทธิ รักถนอม
รหัสประจำตัว	52611002
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
พ.ศ.	2557
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร.ศุภมิตร จิตตะยโสธร

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้ นำเสนอการพัฒนาระบบสารสนเทศเชิงเวลาโดยใช้ระบบจัดการฐานข้อมูลเชิงเวลา นำเสนอการทดลองการใช้ภาษาปฏิบัติการการเข้าถึงข้อมูลที่รองรับเชิงเวลาบน **Workspace Manager** ของโปรแกรมบริหารจัดการฐานข้อมูล **Oracle11g** และนำเสนอการออกแบบฐานข้อมูลเชิงเวลา ซึ่งขั้นตอนการออกแบบจะใช้แบบจำลองที่รองรับข้อมูลเชิงเวลา **Temporal Object-Role Modeling (TORM)** มีการพัฒนาระบบสารสนเทศเชิงเวลาโดยใช้ระบบจัดการฐานข้อมูลเชิงเวลา ที่สามารถใช้งานได้จริง อีกทั้งสามารถตอบคำถามข้อสนเทศที่สัมพันธ์กับเวลา ทั้งในอดีต ปัจจุบัน และอนาคตได้ ทั้งนี้ได้ใช้ระบบงานบุคลากรเป็นกรณีศึกษา สำหรับบันทึกการเปลี่ยนแปลง ข้อมูลชื่อ ข้อมูลที่อยู่ ข้อมูลการศึกษา และข้อมูลการทำงานของบุคลากร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis	The Development of a Temporal Information System using a Temporal DBMS
Student	Mr.Sitti Rugtanom
Student ID.	52611002
Degree	Master of Engineering
Program	Computer Engineering
Year	2014
Thesis Advisor	Assoc. Prof. Dr. Suphamit Chittayasothorn

ABSTRACT

In this thesis, the development of a temporal information system using a temporal Database Management System (DBMS) is proposed. The Temporal Object-Role Modeling (TORM/TNIAM) is used for database design. Oracle 11g Workspace Manager was extensively tested and found to be according to the temporal database principles. This thesis demonstrates the development of temporal information system by using the commercially available temporal database system. A human resource management system is developed as a prototype case study.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา คือ รศ.ดร.ศุภมิตร จิตตะโยธอร์ ที่ให้ความช่วยเหลือ ให้คำชี้แนะ ช่วยแก้ปัญหาตลอดจนให้ความรู้และประสบการณ์ที่ดีแก่ข้าพเจ้า

ขอกราบขอบพระคุณคณาจารย์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกๆ ท่าน ที่ได้ประสิทธิ์ประสาทวิชาความรู้ต่างๆ ให้แก่ข้าพเจ้า

ขอกราบขอบพระคุณ อ.สมชาติ หรั่งเจริญ อาจารย์ประจำภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี และคุณปราณี หรั่งเจริญ ผู้เปรียบประดุจพ่อและแม่คนที่สอง ซึ่งให้คำชี้แนะ ในเส้นทางสายอาชีพวิศวกรรมคอมพิวเตอร์ ตลอดจนให้การอุปการะ ให้ที่พักอาศัย ให้การสนับสนุนทุกๆ ด้าน และกำลังใจเสมอมา

ขอขอบพระคุณคณาจารย์ จากภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี ทุกๆ ท่าน สำหรับคำแนะนำและกำลังใจ

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกๆ ท่าน สำหรับคำแนะนำและกำลังใจเสมอมา

สุดท้ายนี้ ขอกราบขอบพระคุณ บิดา-มารดา ที่เป็นกำลังใจและสนับสนุนข้าพเจ้าในทุกเรื่อง คุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับ บิดา-มารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่าน ที่ได้ประสิทธิ์ประสาทวิชาความรู้และถ่ายทอดประสบการณ์ที่ดีให้แก่ข้าพเจ้า

สิทธิ รักถนอม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	IIIX
สารบัญรูป.....	X
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	2
1.3 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย.....	2
1.4 ขอบเขตของงานวิจัย.....	2
1.5 ขั้นตอนของการศึกษา.....	3
1.6 เครื่องมือและอุปกรณ์ที่ใช้ในงานวิจัย.....	3
1.7 โครงสร้างของวิทยานิพนธ์.....	4
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	5
2.1 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	5
2.1.1 ประเภทและความหมายของเวลา.....	5
2.1.1.1 Valid Time.....	5
2.1.1.2 Transaction Time.....	5
2.1.1.3 User Define Time.....	6
2.1.2 กฎบังคับความถูกต้องในฐานข้อมูลเชิงสัมพันธ์ในรูปแบบข้อมูลเชิงเวลา.....	6
2.1.2.1 Temporal Existential Integrity.....	6
2.1.2.2 Temporal Referential Integrity.....	6
2.2 ความซ้ำซ้อนของข้อมูลแบบเชิงเวลา (Duplication of Temporal Data).....	7
2.2.1 ความซ้ำซ้อนแบบนอนซีควেনซ์ (Nonsequence Duplicate).....	8
2.2.2 ความซ้ำซ้อนแบบค่าเท่ากัน (Value-Equivalent Duplicate).....	8
2.2.3 ความซ้ำซ้อนแบบปัจจุบัน (Current Duplicate).....	9
2.2.4 ความซ้ำซ้อนแบบซีควেনซ์ (Sequenced Duplicate).....	10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.3 Modifying Valid-Time State Tables.....	11
2.3.1 การปรับปรุงข้อมูล ณ เวลาแบบปัจจุบัน (Current Modification).....	13
2.3.2 การเปลี่ยนแปลงแบบซีควเอนซ์ (Sequence Modifications).....	14
2.3.2.1 การเพิ่มข้อมูลแบบซีควเอนซ์ (Sequence Insert).....	14
2.3.2.2 การลบแบบซีควเอนซ์ (Sequence Deletion).....	14
2.3.2.3 การแก้ไขแบบซีควเอนซ์ (Sequence Update).....	18
2.4 Transaction-Time State Tables.....	22
2.5 Bitemporal State Tables.....	24
2.6 Temporal Type and Modules.....	24
2.6.1 Temporal Type.....	24
2.6.2 Temporal Module.....	25
2.7 Temporal Functional Dependencies : TFD.....	25
2.8 Temporal Normalization.....	27
2.8.1 Time Normal Form (TNF).....	27
2.8.2 First Temporal Normal Form (1TNF).....	28
2.8.3 Temporal Boyce-Code Normal Form (TBCNF).....	29
2.8.4 Temporal Third Normal Form (T3NF).....	30
2.8.5 Temporal fourth Normal Form (T4NF).....	30
2.9 Temporal Data Type.....	30
2.9.1 Instant.....	30
2.9.2 Duration.....	30
2.9.3 Period.....	30
2.10 แบบจำลองที่โออาร์เอ็ม (Temporal Object-Role Modeling : TORM).....	30
2.10.1 Temporal Entity Type.....	31
2.10.2 Temporal Fact Type.....	31
2.10.3 Elementary Fact Type.....	32
2.10.4 Referenct Type.....	32
2.10.5 Integrity Constraint.....	32
2.10.6 ขั้นตอนเขียนแบบจำลองที่โออาร์เอ็ม.....	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 3 การทดลองคำสั่งจัดการข้อมูลแบบเชิงเวลา.....	36
3.1 การสร้างตารางแบบเชิงเวลา (Valid Time State Table).....	36
3.2 การทดลองการเพิ่มข้อมูลเวลาแบบปัจจุบัน (Current Insert).....	38
3.2.1 Current Update Case 1, in the General Scenario.....	38
3.2.2 Current Update Case 2, in the General Scenario.....	40
3.2.3 Current Update Case 3, in the General Scenario.....	42
3.3 การทดลองลบข้อมูลแบบซีควเอนซ์ (Sequence Deletion).....	43
3.3.1 การลบข้อมูลแบบซีควเอนซ์ กรณีที่ 1 (Sequence Deletion Case 1).....	43
3.3.2 การลบข้อมูลแบบซีควเอนซ์ กรณีที่ 2 (Sequence Deletion Case 2).....	45
3.3.3 การลบข้อมูลแบบซีควเอนซ์ กรณีที่ 3 (Sequence Deletion Case 3).....	47
3.3.4 การลบข้อมูลแบบซีควเอนซ์ กรณีที่ 4 (Sequence Deletion Case 4).....	48
3.4 การทดลองการปรับปรุงข้อมูลแบบซีควเอนซ์ (Sequence Update).....	50
3.4.1 การปรับปรุงข้อมูลแบบซีควเอนซ์ กรณีที่ 1 (Sequence Update Case 1).....	50
3.4.2 การปรับปรุงข้อมูลแบบซีควเอนซ์ กรณีที่ 2 (Sequence Update Case 2).....	52
3.4.3 การปรับปรุงข้อมูลแบบซีควเอนซ์ กรณีที่ 3 (Sequence Update Case 3).....	54
3.4.4 การปรับปรุงข้อมูลแบบซีควเอนซ์ กรณีที่ 4 (Sequence Update Case 4).....	56
3.6 การทดลอง Temporal Projection and Selection.....	58
3.7 การทดลอง Transaction-Time State Tables.....	69
3.8 ข้อแตกต่างระหว่าง DBMS_WM.MAX_TIME กับ DBMS_WM.UNTIL_CHANGE.....	78
3.9 สรุปผลการทดลองคำสั่งจัดการข้อมูลแบบเชิงเวลา.....	82
บทที่ 4 การออกแบบระบบสารสนเทศเชิงเวลาโดยใช้งานบุคลากรเป็นกรณีศึกษา.....	83
4.1 การออกแบบ ORM ประวัติข้อมูลเฉพาะบุคคล.....	83
4.2 การออกแบบ TORM ประวัติข้อมูลชื่อ-นามสกุล.....	84
4.3 การออกแบบ TORM ประวัติข้อมูลที่อยู่.....	85
4.4 การออกแบบ TORM ประวัติข้อมูลการศึกษา.....	88
4.5 การออกแบบ TORM ประวัติข้อมูลตำแหน่งงาน.....	90
4.6 พจนานุกรมข้อมูล ของระบบงานบุคลากรแบบเชิงเวลา (Data Dictionary).....	92
4.6.1 ตารางระบบงานบุคลากรแบบเชิงเวลา.....	92
4.6.2 พจนานุกรมข้อมูลของตารางแบบสัมพันธ์ (Relation Data Dictionary).....	93

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.6.3	พจนานุกรมข้อมูลของตารางแบบเชิงเวลา (Temporal Data Dictionary)....	96
บทที่ 5	การสอบถามข้อสนเทศเชิงเวลา.....	101
5.1	การสร้างตารางบันทึกข้อมูล.....	101
5.1.1	การสร้างตารางแบบเชิงสัมพันธ์ 13 ตาราง.....	101
5.1.2	การสร้างตารางแบบเชิงเวลา 22 ตาราง.....	103
5.1.3	การทำให้ตารางปกติ (Non-Temporal) เป็นตารางแบบเชิงเวลา (Temporal Table).....	106
5.2	การตอบข้อสนเทศด้านประวัติชื่อ-นามสกุล.....	108
5.3	การตอบข้อสนเทศด้านประวัติที่อยู่.....	111
5.4	การตอบข้อสนเทศด้านประวัติการศึกษา.....	119
5.5	การตอบข้อสนเทศด้านประวัติการทำงาน.....	124
5.6	การยกเลิกตารางแบบเชิงเวลา และการลบตารางทั้งหมด.....	137
บทที่ 6	สรุปผลการวิจัยและข้อเสนอแนะ.....	139
6.1	สรุปผลการทดลอง.....	139
6.2	ข้อเสนอแนะ.....	141
	เอกสารอ้างอิง.....	142
	ภาคผนวก.....	143
	ภาคผนวก ก. การทดลองบันทึกแก้ไขข้อมูลด้านประวัติชื่อ-นามสกุล.....	144
	ภาคผนวก ข. การทดลองบันทึกแก้ไขข้อมูลด้านประวัติที่อยู่.....	154
	ภาคผนวก ค. การทดลองบันทึกแก้ไขข้อมูลด้านประวัติการศึกษา.....	175
	ภาคผนวก ง. การทดลองบันทึกแก้ไขข้อมูลด้านประวัติการทำงาน.....	192
	ภาคผนวก จ. ระบบสารสนเทศเชิงเวลาต้นแบบ.....	223
	ประวัติผู้เขียน.....	256

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 ตัวอย่างข้อมูลในตารางข้อมูลเชิงเวลา.....	6
2.2 สถานะคนไข้ของโรงพยาบาลเด็ก (NICUStatus Table).....	7
2.3 ความซ้ำซ้อนแบบนอนซีเควนซ์.....	8
2.4 ความซ้ำซ้อนแบบค่าเท่ากัน.....	9
2.5 ความซ้ำซ้อนแบบปัจจุบัน.....	9
2.6 ความซ้ำซ้อนแบบซีเควนซ์.....	10
2.7 Valid-Time State Table ผุงั่ว.....	11
2.8 บันทึกเวลาการเปลี่ยนแปลงเพศวั LOT.....	12
2.9 WDS บันทึกตำแหน่งและความสว่างของดวงดาว.....	23
2.10 WDS_TT เป็นตาราง Audit Log ของตาราง WDS.....	23
2.11 Bitemporal Table.....	24
2.12 การเปลี่ยนแปลงข้อมูลของแต่ละเลขบัญชี (ACCOUNTS).....	26
2.13 TRANSACTION-INFO.....	27
2.14 ACCU-INTEREST.....	27
2.15 The EXPENSE Relation.....	28
2.16 The SAL-DEPT Relation.....	28
2.17 The SAL Relation.....	29
2.18 A Temporal Relation.....	29
3.1 WDS_TT เป็นตาราง audit log ของตาราง WDS.....	69
3.2 คำอธิบาย MIN_TIME MAX_TIME และ UNTIL_CHANGED.....	78
4.1 ตารางระบบงานบุคคลากรแบบเชิงสัมพันธ์ (Relation Table).....	92
4.2 ตารางระบบงานบุคคลากรแบบเชิงเวลา (Temporal Table).....	92
4.3 บันทึกประวัติบุคคล (PERSONAL).....	93
4.4 บันทึกหมู่โลหิต (BLOOD).....	94
4.5 บันทึกรหัสค่านำหน้านาม (TITLECODE).....	94
4.6 บันทึกที่อยู่ (ADDRESS).....	94
4.7 บันทึกประวัติการศึกษา (PERSONAL_EDUCATION).....	94
4.8 บันทึกระดับการศึกษา (DEGREE_LEVEL).....	95
4.9 บันทึกวุฒิการศึกษา (QUALIFICATION).....	95
4.10 บันทึกสาขาวิชา (MAJOR).....	95
4.11 บันทึกสถาบัน สังกัดประเทศ (INSTITUTION_COUNTRY).....	95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

ตารางที่	หน้า
4.12	บันทึกประเภทบุคลากร (EMPLOYEE TYPE)..... 95
4.13	บันทึกสายการทำงาน (FIELDTYPE)..... 95
4.14	บันทึกชื่อตำแหน่งทางบริหาร (POSADMIN)..... 96
4.15	บันทึกชื่อตำแหน่งทางวิชาการ (POSACADEMIC)..... 96
4.16	บันทึกคำนำหน้านามของบุคคล (PERSONAL_TITLECODE)..... 96
4.17	บันทึกชื่อของบุคคล (PERSONAL_FNAME)..... 96
4.18	บันทึกนามสกุลของบุคคล (PERSONAL_LNAME)..... 96
4.19	บันทึกชื่อถนน (ROAD_NAME)..... 97
4.20	บันทึกการสังกัดเลขรหัสบ้าน เข้ากับตำบล (ADDRESS_SUBDISTRICT)..... 97
4.21	บันทึกชื่อตำบล (SUBDISTRICT_NAME)..... 97
4.22	บันทึกการสังกัดของตำบล เข้ากับอำเภอ (SUBDISTRICT_DISTRICT)..... 97
4.23	บันทึกชื่ออำเภอ (DISTRICT_NAME)..... 97
4.24	บันทึกการสังกัดของอำเภอ เข้ากับจังหวัด (DISTRICT_PROVINCE)..... 98
4.25	บันทึกชื่อจังหวัด (PROVINCE_NAME)..... 98
4.26	บันทึกการสังกัดของบุคคล เข้ากับรหัสบ้าน (PERSONAL_ADDRESS)..... 98
4.27	บันทึกชื่อสถาบันการศึกษา (INSTITUTION_NAME)..... 98
4.28	บันทึกชื่อประเทศ (COUNTRY_NAME)..... 98
4.29	บันทึกชื่อคณะ (FACULTY)..... 99
4.30	บันทึกชื่อแผนก (DEPARTMENT)..... 99
4.31	บันทึกการสังกัดแผนก เข้ากับคณะ (DEPARTMENT_FACULTY)..... 99
4.32	บันทึกเลขอัตรา (PERSONAL_EMPLOYEE NUMBER)..... 99
4.33	บันทึกบุคลากร เข้ากับประเภทสายการทำงาน (PERSONAL_FIELDTYPE)..... 199
4.34	บันทึกประเภทสายการทำงาน ของบุคลากร (PERSONAL_DEPARTMENT)..... 100
4.35	บันทึกประวัติตำแหน่งทางวิชาการ ของบุคลากร (PERSONAL_POSACADEMIC)..... 100
4.36	บันทึกประวัติตำแหน่งทางบริหาร ของบุคลากร (PERSONAL_POSADMIN)..... 100
4.37	บันทึกประวัติเงินเดือน ของบุคลากร (PERSONAL_SALARY)..... 100
5.1	ข้อมูลสมมุติ ของประชากรที่อาศัยอยู่ใน จ.อุดรธานี (ข้อมูลก่อนปี 2536)..... 113
5.2	ข้อมูลสมมุติ ประวัติการศึกษาของบุคลากร..... 119

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	Current Update Case แบบ General Scenario.....	13
2.2	Sequenced Deletion Case 1-4.....	15
2.3	Sequenced Deletion Case 1.....	15
2.4	Sequenced Deletion Case 2.....	16
2.5	Sequenced Deletion Case 3.....	16
2.6	Sequenced Deletion Case 4.....	16
2.7	ตัวอย่างคำสั่ง SQL มาตรฐานของ Sequenced Deletion.....	17
2.8	Sequenced Update Case 1-4.....	18
2.9	Sequenced Update Case 1.....	19
2.10	Sequenced Update Case 2.....	19
2.11	Sequenced Update Case 3.....	20
2.12	Sequenced Update Case 4.....	20
2.13	ตัวอย่างคำสั่ง SQL มาตรฐานของ Sequenced Update.....	21
2.14	Temporal Entity Type with Own Life Span.....	31
2.15	Label Type DepName.....	31
2.16	Temporal Binary Fact Type.....	32
2.17	Binary Fact Type.....	32
2.18	สัญลักษณ์แทน Inter Fact Type Uniqueness Constraint.....	32
2.19	สัญลักษณ์แทน Intra Fact Type Uniqueness Constraint.....	33
2.20	การใช้ Mandatory Fact Type Constraint.....	33
2.21	ตัวอย่างการเขียนโครงสร้างหลักของระบบสารสนเทศนักศึกษาเชิงเวลา.....	33
2.22	ตัวอย่างการเขียนโครงข่ายย่อยระบบสารสนเทศนักศึกษาเชิงเวลา.....	34
2.23	A Temporal ORM Version of the ORM Conceptual Schema.....	35
2.24	Corresponding Relational Database Schemas of the TORM.....	35
3.1	Current Update Case 1, in the General Scenario.....	38
3.2	Current Update Case 2, in the General Scenario.....	40
3.3	ผลลัพธ์ Current Update case 2, in the General Scenario.....	41
3.4	Current Update Case 3, in the General Scenario.....	42
3.5	ผลลัพธ์ Current Update Case 3, in the General Scenario.....	43
3.6	Sequenced Deletion Case 1.....	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.7 Sequenced Deletion Case 2.....	45
3.8 Sequenced Deletion Case 3.....	47
3.9 Sequenced Deletion Case 4.....	49
3.10 Sequenced Update Case 1.....	50
3.11 Sequenced Update Case 2.....	52
3.12 Sequenced Update Case 3.....	54
3.13 Sequenced Update Case 4.....	56
3.14 ผลการทำงานของเพิ่มข้อมูลตามข้อ 3.7.3.....	70
3.15 ผลข้อมูลจาก WDS WDS_LT และ WDS_HIST ตามข้อ 3.7.4.....	71
3.16 ผลข้อมูลจาก WDS WDS_LT และ WDS_HIST ตามข้อ 3.7.5.....	71
3.17 ผลข้อมูลจาก WDS WDS_LT และ WDS_HIST ตามข้อ 3.7.6.....	72
3.18 ผลข้อมูลจาก WDS WDS_LT และ WDS_HIST ตามข้อ 3.7.7.....	73
3.19 ผลข้อมูลจาก WDS WDS_LT และ WDS_HIST ตามข้อ 3.7.8.....	74
3.20 ผลข้อมูลจาก WDS WDS_LT และ WDS_HIST ตามข้อ 3.7.9.....	75
3.21 ผลข้อมูลจาก WDS WDS_LT และ WDS_HIST ตามข้อ 3.7.10.....	76
3.22 ผลข้อมูลจาก WDS WDS_LT และ WDS_HIST ตามข้อ 3.7.11.....	77
4.1 ORM ประวัติข้อมูลเฉพาะบุคคล.....	83
4.2 โครงร่างตารางประวัติข้อมูลเฉพาะบุคคล.....	83
4.3 TORM ประวัติชื่อ-นามสกุล.....	84
4.4 โครงร่างตารางประวัติข้อมูลชื่อ-นามสกุล.....	84
4.5 TORM ประวัติที่อยู่.....	86
4.6 โครงร่างตารางประวัติที่อยู่.....	87
4.7 TORM ประวัติการศึกษา.....	88
4.8 โครงร่างตารางประวัติการศึกษา.....	89
4.9 TORM ประวัติการทำงาน.....	90
4.10 โครงร่างตารางของประวัติการทำงาน.....	91
5.1 โครงร่างตารางบันทึกชื่อนามสกุล ปัญหาที่ 1.....	108
5.2 โครงร่างตารางบันทึกชื่อนามสกุล ปัญหาที่ 2.....	108
5.3 โครงร่างตารางบันทึกที่อยู่ ปัญหาที่ 1.....	111
5.4 โครงร่างตารางบันทึกที่อยู่ ปัญหาที่ 2.....	112

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
5.5	โครงร่างตารางแบบเชิงเวลา ด้านประวัติการทำงาน.....	125
5.6	ช่วงเวลาที่ถูกบล็อก ดำรงตำแหน่ง ผู้ช่วยศาสตราจารย์.....	128
6.1	โครงร่างตารางที่คอลัมน์ช่วงระยะเวลากำกับแต่ละ Entity Type และข้อมูลตัวอย่าง.....	140
6.2	โครงร่างตารางที่คอลัมน์ช่วงระยะเวลากำกับ Attribute และข้อมูลตัวอย่าง.....	140



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ฐานข้อมูลแบบเชิงเวลา (Temporal Database) เป็นฐานข้อมูลที่สนับสนุนการบันทึกและการสอบถามข้อมูลที่สัมพันธ์กับเวลา ซึ่งแตกต่างจากฐานข้อมูลแบบเดิม (Non-Temporal Database) ที่ออกแบบมาเพื่อบันทึกข้อมูลที่เป็นจริง ณ เวลาปัจจุบัน เมื่อมีการแก้ไขข้อมูลจะทำให้ค่าของข้อมูลเดิมที่มีอยู่ถูกแทนที่ด้วยข้อมูลใหม่ที่ทำการแก้ไข ทำให้ข้อมูลเดิมหายไปจากระบบ

ทั้งนี้ฐานข้อมูลแบบเชิงเวลา มีการเพิ่มส่วนของเวลาที่ข้อมูลนั้นเป็นจริง หรือช่วงเวลาข้อมูลที่ถูกรับบันทึกในฐานข้อมูล ประกอบด้วยแอททริบิวต์เริ่มต้นของข้อมูล (Start Time Attribute) และแอททริบิวต์สิ้นสุดของข้อมูล (End Time Attribute) การเพิ่มแอททริบิวต์ด้านเวลา หากขบวนการออกแบบระบบฐานข้อมูลและภาษาที่ใช้ในการปฏิบัติการเข้าถึงข้อมูล ไม่รองรับข้อมูลแบบเชิงเวลา จะทำให้ไม่สามารถตอบสนองต่อข้อสนเทศเชิงเวลาได้ครบถ้วน

ถึงแม้ว่าฐานข้อมูลแบบเดิม จะสามารถเพิ่มแอททริบิวต์วันเริ่มต้นและวันสิ้นสุดให้กับข้อมูลได้ก็ตาม แต่ภาษาและคำสั่งยังไม่สะดวกในการเข้าถึงข้อมูลแบบเชิงเวลา ดังนั้นในการบันทึกข้อมูล (Insert) การปรับปรุงข้อมูล (Update) การลบข้อมูล (Delete) และการสอบถามข้อมูล (Select) นักพัฒนาโปรแกรมจะต้องเขียนคำสั่ง SQL แบบมาตรฐาน ตรวจสอบช่วงเวลาต่างๆ ที่คาบเกี่ยวกัน ซึ่งในบางครั้งอาจใช้คำสั่ง SQL แบบมาตรฐานถึง 5 คำสั่ง ทำให้เกิดความยุ่งยากในการเขียนคำสั่ง อาจทำให้เกิดข้อผิดพลาดได้ง่าย

โปรแกรมบริหารจัดการฐานข้อมูลออร์าคิล สิบเอ็ดจี (Oracle 11g) มีส่วนคำสั่งที่รองรับฐานข้อมูลและภาษาเชิงเวลา โดยทำงานผ่าน Workspace Manager ในการทำวิจัยในครั้งนี้ ผู้ทำวิจัยได้ทดลองคำสั่งต่างๆ ของออร์าคิล เพื่อทดสอบความสามารถเชิงเวลาในการบันทึกฐานข้อมูลและภาษาเชิงเวลา ซึ่งเป็นไปตามทฤษฎีของฐานข้อมูลเชิงเวลา

จากนั้นได้ ออกแบบและพัฒนาระบบสารสนเทศเชิงเวลาต้นแบบ โดยใช้ระบบงานบุคลากรเป็นกรณีศึกษา ใช้แบบจำลองที่โออาร์เอ็ม (Temporal Object-Role Modeling : TORM/TNIAM) ในการออกแบบฐานข้อมูล และทดลองคำสั่งปฏิบัติการเข้าถึงข้อมูลเชิงเวลา ผ่าน Workspace Manager ซึ่งเป็นระบบสารสนเทศเชิงเวลาต้นแบบที่สามารถใช้คำสั่งปฏิบัติการเชิงเวลาได้จริง

เอกสารนี้เป็นเอกสารลิขสิทธิ์ส่วนตัวสำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วิทยานิพนธ์ฉบับนี้ มุ่งหวังเพื่อศึกษาและออกแบบการพัฒนาระบบสารสนเทศเชิงเวลาโดยใช้ระบบจัดการฐานข้อมูลเชิงเวลา ขั้นตอนการออกแบบฐานข้อมูล ได้ใช้แบบจำลองฐานข้อมูลที่รองรับฐานข้อมูลแบบเชิงเวลา และทดลองการใช้คำสั่งปฏิบัติการเข้าถึงข้อมูลแบบเชิงเวลาผ่าน **Workspace Manager** ของ **Oracle 11g** เพื่อให้ได้ระบบสารสนเทศเชิงเวลาต้นแบบ ที่สามารถสร้างเป็นแอปพลิเคชันที่ใช้งานได้จริง เพื่อการพัฒนาในโอกาสต่อไป

1.3 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย

เพื่อที่จะให้บรรลุผลตามวัตถุประสงค์ ในการสร้างระบบสารสนเทศเชิงเวลา จะต้องเข้าใจหลักการออกแบบและวิเคราะห์ระบบฐานข้อมูลแบบเชิงเวลา ดังต่อไปนี้

- 1.3.1 ประเภทฐานข้อมูลเชิงเวลา ความซับซ้อนข้อมูลเชิงเวลา
- 1.3.2 การออกแบบฐานข้อมูลเชิงเวลา โดยใช้ที่โออาร์เอ็ม
- 1.3.3 การใช้คำสั่งที่รองรับภาษาเชิงเวลา

1.4 ขอบเขตการวิจัย

- 1.4.1 ศึกษาทฤษฎีระบบฐานข้อมูลแบบเชิงเวลา
- 1.4.2 ศึกษาและทดลองคำสั่งปฏิบัติการเข้าถึงข้อมูลแบบเชิงเวลาบน **Workspace Manager** ของ **Oracle 11g**
- 1.4.3 ศึกษาการออกแบบฐานข้อมูลด้วยแบบจำลองที่โออาร์เอ็ม
- 1.4.4 ออกแบบระบบสารสนเทศเชิงเวลา โดยใช้ระบบงานบุคลากรเป็นกรณีศึกษา ด้วยแบบจำลองที่โออาร์เอ็ม
- 1.4.5 ทดลองคำสั่งปฏิบัติการเข้าถึงข้อมูลแบบเชิงเวลาบน **Workspace Manager** ของ **Oracle 11g** เพื่อพัฒนาระบบสารสนเทศเชิงเวลา ตามที่ได้ออกแบบในข้อ 1.4.4
- 1.4.6 แสดงคำสั่งปฏิบัติการเข้าถึงข้อมูลแบบเชิงเวลา ที่ไม่สามารถตอบได้หรือทำได้ยาก ถ้าเป็นฐานข้อมูลแบบเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ขั้นตอนของการศึกษา

1.5.1 กำหนดวัตถุประสงค์และขอบเขตของงานวิจัย ว่าต้องการนำเสนอหรือศึกษางานวิจัยในหัวข้อนี้มากน้อยแค่ไหนและอย่างไร

1.5.2 ศึกษาค้นคว้าทฤษฎีและงานวิจัยที่เกี่ยวข้องกับฐานข้อมูลเชิงเวลา แล้ววิเคราะห์หัวข้อที่ข้อดี และประเด็นที่น่าสนใจของงานวิจัยอื่นๆ ที่สามารถนำมาปรับปรุงและประยุกต์ใช้ เพื่อก่อให้เกิดประโยชน์แก่งานวิจัยนี้ได้

1.5.3 ตั้งสมมติฐานของการศึกษาและกำหนดวางแผนความคิดของงานวิจัย โดยมีการอ้างอิงทฤษฎีหรือหลักการที่เกี่ยวข้อง เพื่อที่จะบรรลุตามวัตถุประสงค์ที่ได้กำหนดไว้

1.5.4 ศึกษาและทดลองชุดคำสั่งของระบบจัดการฐานข้อมูล (DBMS) Oracle 11g ผ่าน Workspace Manager ที่สนับสนุนคำสั่งด้านเชิงเวลา

1.5.5 ออกแบบระบบฐานข้อมูลเชิงเวลา ด้วยแบบจำลองที่โอเอเอ็ม

1.5.6 พัฒนาโปรแกรมสารสนเทศ เพื่อทดสอบบันทึกข้อมูล แก้ไข และลบข้อมูล เพื่อตรวจสอบการทำงานของ Oracle 11g

1.5.7 ทดลองสอบถามข้อสนเทศต่างๆ เชิงเวลา

1.5.8 สรุปผลลัพธ์จากการทดลอง วิเคราะห์และประเมินผลงานวิจัย

1.6 เครื่องมือและอุปกรณ์ที่ใช้ในงานวิจัย

เครื่องมือและอุปกรณ์ที่ใช้ในการวิจัยในครั้งนี้ ได้แก่

1.6.1 เครื่องคอมพิวเตอร์ส่วนบุคคลใช้หน่วยประมวลผลกลาง Intel Core (TM) 2 Duo 2.33 GHz หน่วยความจำ (RAM) 6 จิกะไบท์ (GB) จำนวน 1 เครื่อง

1.6.2 ระบบปฏิบัติการวินโดวส์ 7

1.6.3 ซอฟต์แวร์เจดีเค (JDK) เวอร์ชัน Jdk-6u18-windows-i586

1.6.4 โปรแกรมบริหารจัดการฐานข้อมูลออรากิล สิบเอ็ดจี (Oracle 11g)

1.6.5 โอเจดีบีซี 6 (Ojdbc6)

1.6.6 ซอฟต์แวร์ Tomcat เวอร์ชัน Apache-tomcat-7.0.42-windows-x64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.7 โครงสร้างของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ แบ่งออกเป็น 6 บท แต่ละบทประกอบด้วยเนื้อหาดังต่อไปนี้

บทที่ 1 กล่าวถึงความเป็นมาและความสำคัญของปัญหา ความมุ่งหมายและวัตถุประสงค์ของการศึกษา ทฤษฎีหรือแนวคิดที่ใช้ในการศึกษา ขอบเขตของการศึกษา ขั้นตอนของการศึกษา เครื่องมือและอุปกรณ์ที่ใช้ในงานวิจัย

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

บทที่ 3 การทดลองคำสั่งจัดการข้อมูลแบบเชิงเวลา

บทที่ 4 การออกแบบระบบสารสนเทศเชิงเวลา

บทที่ 5 การสอบถามข้อสนเทศเชิงเวลา

บทที่ 6 บทสรุปผลการวิจัยและข้อเสนอแนะ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ฐานข้อมูลเชิงเวลา (Temporal Database)

ฐานข้อมูลแบบเดิม (Non-Temporal) จะเก็บเฉพาะข้อมูล ณ เวลาใดเวลาหนึ่งเท่านั้น ซึ่งปกติจะเป็นข้อมูลล่าสุดในการเปลี่ยนแปลงข้อมูล ข้อมูลเก่าจะถูกเขียนทับโดยข้อมูลใหม่ การเรียกค้นหาข้อมูล จะเป็นการเรียกค้นหาข้อมูลที่เป็นจริงขณะที่เริ่มค้นหาเท่านั้น ถ้าจะเรียกค้นหาข้อมูลในอดีต จะไม่สามารถทำได้

ฐานข้อมูลเชิงเวลา (Temporal Database) คือ ฐานข้อมูลที่สนับสนุนความสัมพันธ์เชิงเวลาของข้อมูล โดยไม่รวมถึงเวลาที่ผู้ใช้กำหนด (User Define Time) [1]

2.1.1 ประเภทและความหมายของเวลา

ในฐานข้อมูลเชิงเวลานั้น จะมีการระบุเวลาเข้ากับฐานข้อมูล ซึ่งประเภทของเวลาที่ระบุในฐานข้อมูลมี 3 ประเภท ดังนี้

2.1.1.1 Valid Time

คือ เวลาที่แสดงถึงความสัมพันธ์ที่เป็นจริงของข้อมูล (เวลาที่ Fact เป็นจริง) หรือช่วงเวลาที่ยุทธมูลนั้นเป็นจริงในฐานข้อมูล ยกตัวอย่างเช่น นาย ก. บรรจุทำงานตำแหน่งโปรแกรมเมอร์ เมื่อวันที่ 1 มกราคม 2550 และได้รับการเลื่อนตำแหน่งเป็น นักวิเคราะห์ระบบ เมื่อวันที่ 1 มกราคม 2556 ดังนั้น ช่วงเวลา Valid Time ของนาย ก. ที่ทำงานตำแหน่งโปรแกรมเมอร์ คือ 1 มกราคม 2550 - 31 มกราคม 2555 และ Valid Time ของนาย ก. ที่ทำงานตำแหน่งนักวิเคราะห์ระบบ คือ ตั้งแต่วันที่ 1 มกราคม 2556 จนถึงปัจจุบัน

2.1.1.2 Transaction Time

คือ เวลาที่ยุทธมูลนั้นถูกบันทึกลงในฐานข้อมูล โดยส่วนใหญ่จะได้จากโปรแกรมบริหารจัดการฐานข้อมูล (DBMS) ทำหน้าที่บันทึกอัตโนมัติ และจะไม่สามารถแก้ไขข้อมูลในอดีตได้ เนื่องจากจะผิดกฎของเวลาที่ว่า เราไม่สามารถเปลี่ยนแปลงอดีตได้ ยกตัวอย่างเช่น ข้อมูลเงินเดือนของพนักงาน ก. เดือนละ 15,000 บาทต่อเดือน ได้ถูกบันทึกเมื่อวันที่ 1 มกราคม 2550 และต่อมาได้มีการวางแผนว่าจะปรับเงินเดือนให้เป็น 18,000 บาทต่อเดือน ในวันที่ 1 มกราคม 2551 แต่ฝ่ายการเงินได้บันทึกข้อมูลเมื่อวันที่ 30 ธันวาคม 2550 ดังนั้น Transaction Time ของข้อมูลพนักงาน ก. ที่มีเงินเดือน 15,000 บาทต่อเดือน คือ 1 มกราคม 2550 - 30 ธันวาคม 2550 เป็นต้น [2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1.3 User Define Time

คือ เวลาที่อยู่ในรูปแบบที่เป็นส่วนหนึ่งของข้อมูล ระบบจะมองเป็นแค่ข้อมูลธรรมดา ถึงแม้จะมีประเภทข้อมูลเป็นเวลาก็ตาม ส่วนใหญ่จะเป็นข้อมูลที่มีค่าคงที่ ยกตัวอย่าง วันเกิด วันสมรส วันบรรจุเข้าทำงาน

2.1.2 กฎบังคับความถูกต้องในฐานะข้อมูลเชิงเวลา

กฎบังคับความถูกต้องฐานข้อมูลเชิงสัมพันธ์ในรูปแบบฐานข้อมูลเชิงเวลา คือ กฎที่ใช้บังคับความถูกต้องของข้อมูล โดยจะอาศัยส่วนประกอบสองอย่าง คือ ข้อมูลและเวลา มีอยู่ 2 ประเภท คือ Temporal Existential Integrity และ Temporal Referential Integrity Constraints [2]

2.1.2.1 Temporal Existential Integrity

แอทริบิวท์ ที่ใช้เป็นคีย์หลักของข้อมูลจะต้องไม่มีค่าว่า (Null) ในทุกช่วงเวลาของข้อมูล คือ Existential Integrity ในฐานะข้อมูลเชิงสัมพันธ์แบบธรรมดา (Non-Temporal) จะเป็นกฎที่บังคับความสมบูรณ์ของข้อมูลในด้านเดียว คือ ในตารางเดียวกันห้ามมีคีย์หลักซ้ำกัน หรือมีค่าเป็น Null แต่ในฐานะข้อมูลเชิงเวลา กฎนี้จะนำข้อมูลเชิงเวลามาอ้างอิงด้วย ยกตัวอย่างดังแสดงในตารางที่ 2.1 โดยกำหนดให้ E# เป็นคีย์หลัก ซึ่งแม้จะมีคีย์หลักซ้ำกันในสองแถวแรก แต่อยู่คนละช่วงเวลา จึงสามารถบันทึกข้อมูลได้โดยไม่ผิดกฎบังคับความสมบูรณ์ของข้อมูล เป็นต้น

ตารางที่ 2.1 ตัวอย่างข้อมูลในตารางข้อมูลเชิงเวลา

E#	Start	End
121	10	12
121	14	Now
133	23	30
147	18	Now

2.1.2.2 Temporal Referential Integrity

คือ กฎที่ใช้บังคับสำหรับการอ้างอิงข้อมูล โดยข้อมูลที่อ้างอิงนั้น ต้องสัมพันธ์กับข้อมูลของคีย์หลัก และต้องอยู่ในช่วงเวลาของคีย์หลักที่ใช้อ้างอิงด้วย ซึ่งต้องอาศัยส่วนประกอบสองส่วน คือ ข้อมูลและเวลา เพื่อใช้อ้างอิงข้อมูล แบ่งออกเป็น 2 กรณี คือ

1. กรณีที่ทั้งตารางที่เป็นคีย์หลักเป็นตารางที่สนับสนุน Valid Time แล้วตารางที่อ้างอิงถึงข้อมูลต้องเป็นตารางที่สนับสนุน Valid Time แล้วเท่านั้น และข้อมูลที่อ้างอิงถึงนั้นต้องอยู่ในช่วงเวลาของคีย์หลักเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กรณีที่ตารางเป็นคีย์หลัก เป็นตารางที่ไม่สนับสนุน **Valid Time** จะไม่สนใจเวลาสามารถอ้างอิงถึงข้อมูลนั้นได้เลย

2.2 ความซ้ำซ้อนของข้อมูลแบบเชิงเวลา (Duplication of Temporal Data)

การออกแบบระบบฐานข้อมูลที่ดี จะต้องขจัดความซ้ำซ้อนของข้อมูล ถ้าข้อมูลมีความซ้ำซ้อน จะทำให้เกิดปัญหาในการ เพิ่มข้อมูล (Insert) การปรับปรุงข้อมูล (Update) และการลบข้อมูล (Delete) เพื่อแสดงให้เห็นความซ้ำซ้อนแบบต่างๆ ขอยกตัวอย่างตารางสถานะคนไข้ของโรงพยาบาลเด็กแห่งหนึ่ง ดังตารางที่ 2.2

ตารางที่ 2.2 สถานะคนไข้ของโรงพยาบาลเด็ก (NICUStatus Table) [3]

Name	Status	From_date	To_date
Kenneth Robert	Serious	1997-11-19	1997-11-21
Alexis May	Serious	1997-11-19	1997-11-27
Natalie Sue	Serious	1997-11-19	1997-11-25
Kelsey Ann	Serious	1997-11-19	1997-11-26
Brandon James	Serious	1997-11-19	1997-11-26
Nathan Roy	Serious	1997-11-19	1997-11-28
Joel Steven	Critical	1997-11-19	1997-11-20
Joel Steven	Serious	1997-11-20	1997-11-26
Kenneth Robert	Fair	1997-11-21	1998-01-03
Alexis May	Fair	1997-11-27	1998-01-11
Alexis May	Fair	1997-12-02	9999-12-31
Alexis May	Fair	1997-12-02	9999-12-31

ตารางที่ 2.2 เป็นตารางแบบเชิงเวลา (Temporal Table) เพื่อบันทึกสถานะการรักษาของผู้ป่วยเด็กเล็ก ประกอบด้วยคอลัมน์ ชื่อผู้ป่วย (Name) สถานะอาการ (Status) วันที่เริ่มต้นของอาการ (From_date) และวันที่มีการเปลี่ยนแปลงหรือสิ้นสุดของอาการ (To_date)

สถานะอาการของผู้ป่วย มีดังนี้ อาการสาหัส (Serious) ผู้ที่ต้องเฝ้าระวังอย่างใกล้ชิดที่มีโอกาสกลับไปเป็นอาการสาหัสได้ (Critical) และอาการปกติ (Fair)

จากตารางที่ 2.2 Kenneth Robert มีสถานะอาการสาหัส ตั้งแต่วันที่ 1997-11-19 ถึงวันที่ 1997-11-21 ต่อมาเขามีอาการปกติตั้งแต่วันที่ 1997-11-21 ถึงวันที่ 1998-01-03 จะเห็นว่าวัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มต้นของอาการปกติ จะบันทึกเป็นวันสิ้นสุดของอาการสาหัส ดังนั้นเวลาที่ **Fact** ของอาการสาหัส ที่เป็นจริง คือ ช่วงเวลาตั้งแต่วันที่ **1997-11-19** ถึงวันที่ **1997-11-20**

และจากตารางที่ 2.2 นี้ มีความซ้ำซ้อนของข้อมูลอยู่ 4 รูปแบบ คือ ความซ้ำซ้อนแบบนอนซีควেনซ์ (**Nonsequence Duplicate**) ความซ้ำซ้อนแบบค่าเท่ากัน (**Value-Equivalent Duplicate**) ความซ้ำซ้อนแบบปัจจุบัน (**Current Duplicate**) และความซ้ำซ้อนแบบซีควেনซ์ (**Sequenced Duplicate**)

2.2.1 ความซ้ำซ้อนแบบนอนซีควেনซ์ (**Nonsequence Duplicate**)

คือ ความซ้ำซ้อนในแต่ละแถวข้อมูล ของทุกๆ คอลัมน์ มีค่าเหมือนกัน โดยรวมถึงคอลัมน์เวลาด้วย ยกตัวอย่างเช่น สองแถวสุดท้ายของตารางที่ 2.2 จะมีค่าข้อมูลทุกๆ คอลัมน์เหมือนกัน

ตารางที่ 2.3 ความซ้ำซ้อนแบบนอนซีควেনซ์ [3]

Name	Status	From_date	To_date
Alexis May	Fair	1997-12-02	9999-12-31
Alexis May	Fair	1997-12-02	9999-12-31

วิธีแก้ไขปัญหาคือการกำหนดคีย์หลัก **Name Status From_date และ To_date** เป็น PK ส่งผลให้คนไข้ไม่สามารถมีหลายสถานะได้ในช่วงเวลาเดียวกัน แต่อย่างไรก็ตามการป้องกันเช่นนี้ อาจไม่ได้ผล เนื่องจากช่วงเวลาเดียวกันอาจไม่ได้มีวันเริ่มต้นและวันสิ้นสุดเหมือนกัน

```
CREATE TABLE NICUStatus (
...
UNIQUE (Name, Status, from date, to date)
)
```

2.2.2 ความซ้ำซ้อนแบบค่าเท่ากัน (**Value-Equivalent Duplicate**)

คือ ความซ้ำซ้อนในแต่ละแถวข้อมูล ของทุกๆคอลัมน์ มีค่าเหมือนกัน ยกเว้นคอลัมน์เวลาดังตัวอย่างเช่น สามแถวสุดท้ายของตารางที่ 2.2 ดังแสดงในตาราง 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 ความซ้ำซ้อนแบบค่าเท่ากัน [3]

Name	Status
Alexis May	Fair
Alexis May	Fair
Alexis May	Fair

การแก้ไขปัญหาค่าซ้ำซ้อนแบบค่าเท่ากัน สามารถกำหนดคอลัมน์ **Name** และ **Status** เป็น **PK** ส่งผลให้คนไข้มีสถานะเดียวเท่านั้น ไม่สามารถมีหลายสถานะได้

```
CREATE TABLE NICUStatus (
  Name CHAR(15),
  Status CHAR(8),
  from date DATE,
  to date DATE,
  UNIQUE (Name, Status)
)
```

2.2.3 ความซ้ำซ้อนแบบปัจจุบัน (Current Duplicate)

คือ ความซ้ำซ้อนในแต่ละแถวข้อมูลที่ค่าต่างๆ คอลัมน์ ยกเว้นคอลัมน์เวลา จะมีค่าเหมือนกัน ณ เวลาปัจจุบัน ตัวอย่างเช่น สมมุติวันนี้เป็นวันที่ 6 มกราคม 1998 จากตาราง 2.2 สามแถวสุดท้ายจะเป็นความซ้ำซ้อนแบบปัจจุบัน เนื่องจากค่าทุกคอลัมน์ซ้ำกัน ณ เวลาปัจจุบัน ดังแสดงในตารางที่ 2.5

ตารางที่ 2.5 ความซ้ำซ้อนแบบปัจจุบัน [3]

Name	Status
Alexis May	Fair
Alexis May	Fair
Alexis May	Fair

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแก้ไขปัญหาค่าซ้ำซ้อนแบบปัจจุบัน สามารถทำได้โดยตรวจสอบการบันทึกข้อมูล ณ เวลาปัจจุบันให้ถูกต้องก่อน ถึงจะสามารถบันทึกข้อมูลได้ ดังตัวอย่างคำสั่ง SQL ต่อไปนี้

```
CREATE TABLE NICUStatus (
...
CHECK (NOT EXISTS (SELECT N1.SSN
FROM NICUStatus AS N1
WHERE 1 < (SELECT COUNT(Name)
FROM NICUStatus AS N2
WHERE N1.Name = N2.Name AND N1.Status = N2.Status
AND N1.from date <= CURRENT DATE
AND CURRENT DATE < N1.to date
AND N2.from date <= CURRENT DATE
AND CURRENT DATE < N2.to date)))
)
```

2.2.4 ความซ้ำซ้อนแบบซีควเอนซ์ (Sequenced Duplicate)

คือ ความซ้ำซ้อนในแต่ละแถวข้อมูล ที่มีค่าของทุกๆ คอลัมน์ ยกเว้นคอลัมน์เวลา มีค่าเหมือนกันอยู่ในช่วงระยะเวลาหนึ่งที่เหลื่อมซ้อนทับกัน ตัวอย่างเช่น สามแถวสุดท้ายของตารางที่ 2.2 เนื่องจากทั้งสามแถวมีค่าเหมือนกันในช่วงเวลาที่เหลื่อมซ้อนทับกัน โดยผู้ป่วย Alexis May มีการบันทึกอาการ Fair ในช่วงวันที่ 1997-11-27 ถึง 1998-01-11 และบันทึกอาการ Fair ครั้งที่ 2 ในช่วงวันที่ 1997-12-02 ถึง 9999-12-31 ซึ่งจะพบว่า มีช่วงเวลาที่ซ้อนทับกัน

ตารางที่ 2.6 ความซ้ำซ้อนแบบซีควเอนซ์ [3]

Name	Status	From_date	To_date
Alexis May	Fair	1997-11-27	1998-01-11
Alexis May	Fair	1997-12-02	9999-12-31
Alexis May	Fair	1997-12-02	9999-12-31

การแก้ไขปัญหาค่าซ้ำซ้อนแบบซีควเอนซ์ สามารถทำได้โดยกำหนดเงื่อนไขการบันทึกข้อมูล ไม่ให้สามารถบันทึกสถานะผู้ป่วยที่เหมือนกัน ในช่วงหนึ่งๆ ได้ ดังตัวอย่างคำสั่ง SQL ต่อไปนี้

```
CREATE TABLE NICUStatus (
...
CHECK (NOT EXISTS (SELECT N1.Name
FROM NICUStatus AS N1
WHERE 1 < (SELECT COUNT(Name)
FROM NICUStatus AS N2
WHERE N1.Name = N2.Name AND N1.Status = N2.Status
AND N1.from date < N2.to date AND N2.from date < N1.to
date))))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Modifying Valid-Time State Tables

ปัญหาการเขียนคำสั่ง เพื่อรองรับข้อมูลแบบเชิงเวลา ในการเพิ่มข้อมูล (Insert) การลบข้อมูล (Delete) การปรับปรุงข้อมูล (Update) และ สอบถามข้อมูล (Select) ตลอดจนปัญหาการปฏิบัติการเชิงเวลา ในลักษณะเวลา ณ ปัจจุบัน (Current) แต่ละช่วงเวลา (Sequence) และ ไม่สนใจเวลา (Nonsequence) ซึ่ง Richard T. Snodgrass ได้แสดงปัญหาเหล่านั้น โดยใช้ตัวอย่างของข้อมูล เรื่องปศุสัตว์ในประเทศสหรัฐ ซึ่งมีอยู่ช่วงหนึ่ง มีการตรวจพบว่ามีเนื้อแช่แข็งจำนวนหนึ่งติดเชื้อร้ายแรง จึงต้องมีการเรียกคืน (Recall) เนื้อแช่แข็งเหล่านั้น ปัญหาคือ จะต้องระบุให้ได้ว่าเนื้อเหล่านั้นมาจาก โรงฆ่าสัตว์ไหน มาจากปศุสัตว์ฝูงไหน และระบุต่อไปอีกว่า มีฝูงไหนบ้างที่เคยอยู่คอกวัวเดียวกันกับฝูงวัวที่ติดเชื้อ แต่เนื่องจากไม่มีข้อมูลสาเหตุเหล่านี้ การเรียกคืนเนื้อวัวทั้งประเทศทำให้เกิดความเสียหายมาก จึงจำเป็นที่จะต้องเก็บข้อมูลเรื่องเวลาไว้ด้วย

Richard T. Snodgrass ได้เสนอกรณีศึกษา การเพิ่มข้อมูลวัว การเปลี่ยนแปลงแก้ไขข้อมูลเพศวัว การลบข้อมูลวัว โดยใช้ตาราง Lot ซึ่งเป็นตารางแบบ Valid Time State Table ทั้งนี้ได้ยกตัวอย่างคำสั่ง SQL ที่ปฏิบัติการเปลี่ยนแปลงนั้น

ตารางที่ 2.7 Valid-Time State Table ฝูงวัว [3]

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

FDYD_ID (Feed yard) คือ หมายเลขลานให้อาหารสัตว์

LOT_ID_NUM คือ หมายเลขฝูงวัว

PEN_ID คือ หมายเลขคอกวัว

HD_CNT (Head Count) คือ จำนวนวัว

FROM_DATE คือ วันเริ่มต้นในการให้อาหาร

TO_DATE คือ วันสิ้นสุดในการให้อาหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 2.7 ประกอบด้วยลานให้อาหารสัตว์หมายเลข 1 เดียวกันทั้งหมด ผุงวัวหมายเลข 219 คอกวัวที่ 1 มีจำนวนวัว 43 ตัว เริ่มให้อาหารตั้งแต่วันที่ 25 กุมภาพันธ์ 1998 จากนั้นในวันที่ 1 มีนาคม 1998 ได้แบ่งวัวเป็นสองคอก คือคอกวัวที่ 1 มีวัว 20 ตัว และคอกวัวที่ 2 มีวัว 23 ตัว ต่อมาในวันที่ 14 มีนาคม 1998 จึงนำวัวทั้งสองคอก กลับมารวมเป็นคอกวัวที่ 2 มีวัว 43 ตัว ดังเดิม

ตารางที่ 2.8 บันทึกเวลาเปลี่ยนแปลงเพศวัว (LOT) [3]

LOT_ID_NUM	GNDR_CODE	FROM_DATE	TO_DATE
101	c	1998-01-10	1998-03-23
101	s	1998-03-23	9999-12-31
234	c	1998-02-17	9999-12-31
799	s	1998-03-12	9999-12-31

LOT_ID_NUM คือหมายเลขของผุงวัว

GNDR_CODE (Gender Code) คือ สถานะของเพศวัว ใช้อักษรย่อ ดังนี้

C = ลูกวัวเพศผู้ (Bull calf)

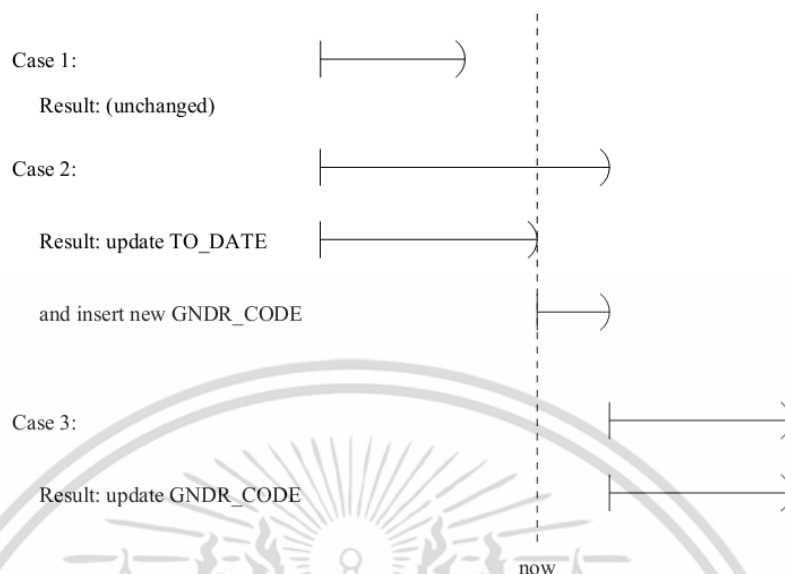
h = วัวสาว (Heifer)

S = วัวเพศผู้ที่ถูกตอน (Steer)

FROM_DATE และ TO_DATE คือ คอลัมน์สำหรับบันทึกวันที่มีการเปลี่ยนแปลงเพศของวัว

จากตารางที่ 2.8 เป็นตารางบันทึกวันเปลี่ยนแปลงเพศวัว เช่น ผุงวัวหมายเลข 101 มีสถานะเป็นลูกวัวเพศผู้ ในช่วงวันที่ 10 มกราคม 1998 ถึงวันที่ 23 มีนาคม 1998 และผุงวัวหมายเลข 101 ดังกล่าวได้ถูกทำการตอนเพศตั้งแต่วันที่ 23 มีนาคม 1998 และมีสถานะการตอนเพศจนถึงปัจจุบัน

2.3.1 การปรับปรุงข้อมูล ณ เวลาแบบปัจจุบัน (Current Modification)



รูปที่ 2.1 Current Update Case แบบ General Scenario [3]

จากรูปที่ 2.1 เป็นกรณีของการปรับปรุงเวลาของการเปลี่ยนแปลง ณ ช่วงเวลาปัจจุบัน ซึ่งสามารถพิจารณาได้ 3 กรณี ดังนี้

กรณีที่ 1 (Case 1) ปรับปรุงข้อมูลในอดีต ณ เวลาปัจจุบัน

ถ้าทำการปรับปรุงข้อมูลในอดีต โดยไม่ได้กำหนดพารามิเตอร์ช่วงเวลา จะเป็นการปรับปรุง ณ ช่วงเวลาปัจจุบัน ผลของการปรับปรุงจะไม่สามารถทำได้ เพราะข้อมูลได้เป็นอดีตไปแล้ว

กรณีที่ 2 (Case 2) ปรับปรุงข้อมูลที่ยังเป็นจริง ณ เวลาปัจจุบัน

ถ้าทำการปรับปรุงข้อมูล ที่ยังเป็นจริงอยู่ โดยไม่ได้กำหนดพารามิเตอร์ช่วงเวลา จะเป็นการปรับปรุง ณ ช่วงเวลาปัจจุบัน ผลของการปรับปรุง จะเกิดแถวข้อมูล 2 แถวข้อมูล (เรียกว่า 1 Insert และ 1 Update)

แถวข้อมูลที่ 1 (ด้านขวาของเส้นประ) เป็นการเพิ่มแถวข้อมูลตามค่าที่ปรับปรุงใหม่ ช่วงเวลาเริ่มต้นจะเป็นเวลา ณ ปัจจุบันที่ปรับปรุง และวันสิ้นสุดจะเป็นวันสิ้นสุดของข้อมูลเดิม

แถวข้อมูลที่ 2 (ด้านซ้ายของเส้นประ) เป็นการปรับปรุงข้อมูลเดิม โดยค่าข้อมูลเป็นค่าเดิม แต่ทำการปรับปรุงเวลาสิ้นสุด จะมีค่าเท่ากับเวลา ณ ปัจจุบันที่ปรับปรุง

กรณีที่ 3 (Case 3) ปรับปรุงข้อมูลในอนาคต ณ เวลาปัจจุบัน

ถ้าทำการปรับปรุงข้อมูล ที่กำหนดช่วงเวลาของอนาคตไว้แล้ว โดยไม่ได้กำหนดพารามิเตอร์ช่วงเวลา จะเป็นการปรับปรุง ณ ช่วงเวลาปัจจุบัน ผลการปรับปรุง ค่าของข้อมูลจะเป็นค่าตามที่ปรับปรุง แต่เวลาของข้อมูลจะเป็นเวลาเดิมก่อนทำการปรับ

การแก้ไขข้อมูล ณ เวลาปัจจุบัน ใน TR-28 ได้แบ่งการแก้ไข ไว้ 2 ประเภท คือ **General Scenario** และ **Restricted Scenario**

General Scenario คือ การยอมให้มีการแก้ไขข้อมูลใน Valid Time State Table ได้ทุกช่วงเวลา ไม่ว่าจะเป็นอดีต ปัจจุบัน หรือ อนาคต

Restricted Scenario คือ การอนุญาตให้แก้ไขข้อมูลใน Valid Time State Table เฉพาะช่วงเวลาปัจจุบันของข้อมูลเท่านั้น ไม่สามารถแก้ไขเปลี่ยนแปลงอดีต ได้

2.3.2 การเปลี่ยนแปลงแบบซีควเอนซ์ (Sequence Modifications)

2.3.2.1 การเพิ่มข้อมูลแบบซีควเอนซ์ (Sequence Insert)

คือ การเพิ่มข้อมูลโดยกำหนดช่วงระยะเวลา พิจารณาการเพิ่มข้อมูล ผุงว้วหมายเลข 426 เข้ามาในฟาร์ม ซึ่งเป็นวัวสาว โดยเข้ามาตั้งแต่วันที่ 26 มีนาคม ถึงวันที่ 14 เมษายน 1998 สามารถเขียนคำสั่ง SQL ได้ดังนี้

```
INSERT INTO LOT
VALUES (426, 'h', DATE '1998-03-26', DATE '1998-04-14')
```

2.3.2.2 การลบแบบซีควเอนซ์ (Sequence Deletion)

บทความวิจัยของ Richard T. Snodgrass TR-28 ได้พิจารณาการเปลี่ยนแปลงของเพศวัว คือ “ผุงว้วหมายเลข 234 ย้ายออกจากฟาร์มใน 3 สัปดาห์แรกของเดือนตุลาคม ซึ่งในช่วงเวลานั้นได้มีการวางแผนที่จะเปลี่ยนผุงว้วหมายเลข 234 ให้เป็นวัวตอนตัวผู้ PA = DATE '1998-10-01' ถึง DATE '1998-10-22'

การเปลี่ยนแปลงลักษณะนี้ จะเกิดกรณีของการเปลี่ยนแปลงข้อมูล 4 กรณี ดังรูปกรณีของการลบแบบซีควเอนซ์ในรูปภาพที่ 2.2 โดยมีรายละเอียด ดังนี้

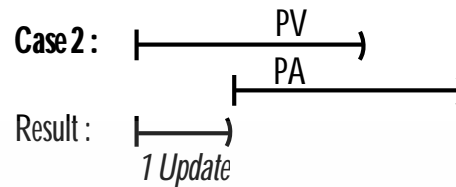
PV คือ ช่วงเวลาของข้อมูลเดิม (Period of Validity)

PA คือ ช่วงเวลาที่ต้องการเปลี่ยนแปลงข้อมูล (Period of Application)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่ 2 (Case 2) : PA เกิดขึ้นและสิ้นสุดที่หลัง PV

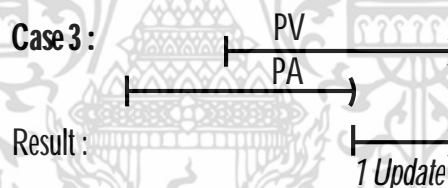
การลบข้อมูลเชิงเวลากรณีที่ 2 นี้ ผลของการลบเป็นการปรับปรุง 1 Update โดยปรับปรุงวันสิ้นสุดของ PV ให้มีค่าเท่ากับวันเริ่มต้นของ PA ส่วนข้อมูลของ PV ยังคงเป็นค่าเดิม และการลบข้อมูลเชิงเวลากรณีที่ 2 จะใช้คำสั่ง SQL มาตรฐาน 1 คำสั่ง ดังตัวอย่าง SQL_2 จากรูปที่ 2.7



รูปที่ 2.4 Sequenced Deletion Case 2

กรณีที่ 3 (Case 3) : PA เกิดขึ้นและสิ้นสุดก่อน PV

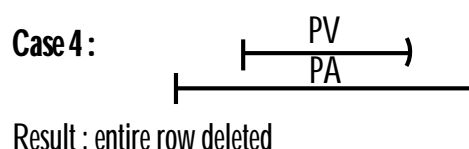
การลบข้อมูลเชิงเวลากรณีที่ 3 นี้ ผลของการลบเป็นการปรับปรุง 1 Update โดยปรับปรุงวันสิ้นสุดของ PV ให้มีค่าเท่ากับวันเริ่มต้นของ PA ส่วนข้อมูลของ PV ยังคงเป็นค่าเดิม และการลบข้อมูลเชิงเวลากรณีที่ 3 จะใช้คำสั่ง SQL มาตรฐาน 1 คำสั่ง ดังตัวอย่าง SQL_3 จากรูปที่ 2.7



รูปที่ 2.5 Sequenced Deletion Case 3

กรณีที่ 4 (Case 4) : PA เกิดขึ้นก่อนและสิ้นสุดที่หลัง PV

การลบข้อมูลเชิงเวลากรณีที่ 4 นี้ ผลของการลบ จะลบแถวข้อมูลของ PA ออกจากฐานข้อมูล และการลบข้อมูลเชิงเวลากรณีที่ 4 จะใช้คำสั่ง SQL มาตรฐาน 1 คำสั่ง ดังตัวอย่าง SQL_4 จากรูปที่ 2.7



รูปที่ 2.6 Sequenced Deletion Case 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ Richard T. Snodgrass ได้แสดงคำสั่ง SQL มาตรฐานเพื่อใช้ลบแบบซีควนซ์ ดังนี้

```
--SQL_1:
INSERT INTO LOT
SELECT LOT_ID_NUM, GNDR_CODE, DATE '1998-10-22', TO_DATE
FROM LOT
WHERE LOT_ID_NUM = 234
      AND FROM_DATE <= DATE '1998-10-01'
      AND TODATE > DATE '1998-10-22'

--SQL_2:
UPDATE LOT
SET TO_DATE = DATE '1998-10-01'
WHERE LOT_ID_NUM = 234
      AND FORM_DATE < DATE '1998-10-01'
      AND TO_DATE >= DATE '1998-10-01'

--SQL_3:
UPDATE LOT
SET TO_DATE = DATE '1998-10-22'
WHERE LOT_ID_NUM = 234
      AND FROM_DATE < DATE '1998-10-22'
      AND TO_DATE >= DATE '1998-10-22'

--SQL_4:
DELETE FROM LOT
WHERE LOT_ID_NUM = 234
      AND FROM_DAET >= DATE '1998-10-01'
      AND TO_DATE <= DATE '1998-10-22'
```

รูปที่ 2.7 ตัวอย่างคำสั่ง SQL มาตรฐานของ Sequenced Deletion

การลบแบบซีควนซ์นี้ จะต้องเขียนคำสั่ง SQL มาตรฐาน ทั้งหมด 4 คำสั่ง เพื่อให้การลบนั้นถูกต้อง และในแต่ละคำสั่งของ SQL มาตรฐาน ผู้เขียนคำสั่งจะต้องเขียนคำสั่งตรวจสอบเงื่อนไขในส่วนของ FROM_DATE และ TO_DATE ให้ถูกต้อง ซึ่งเป็นเรื่องที่ค่อนข้างซับซ้อน เข้าใจยาก หากเขียนเงื่อนไขผิด ส่งผลให้การลบข้อมูลไม่ถูกต้อง

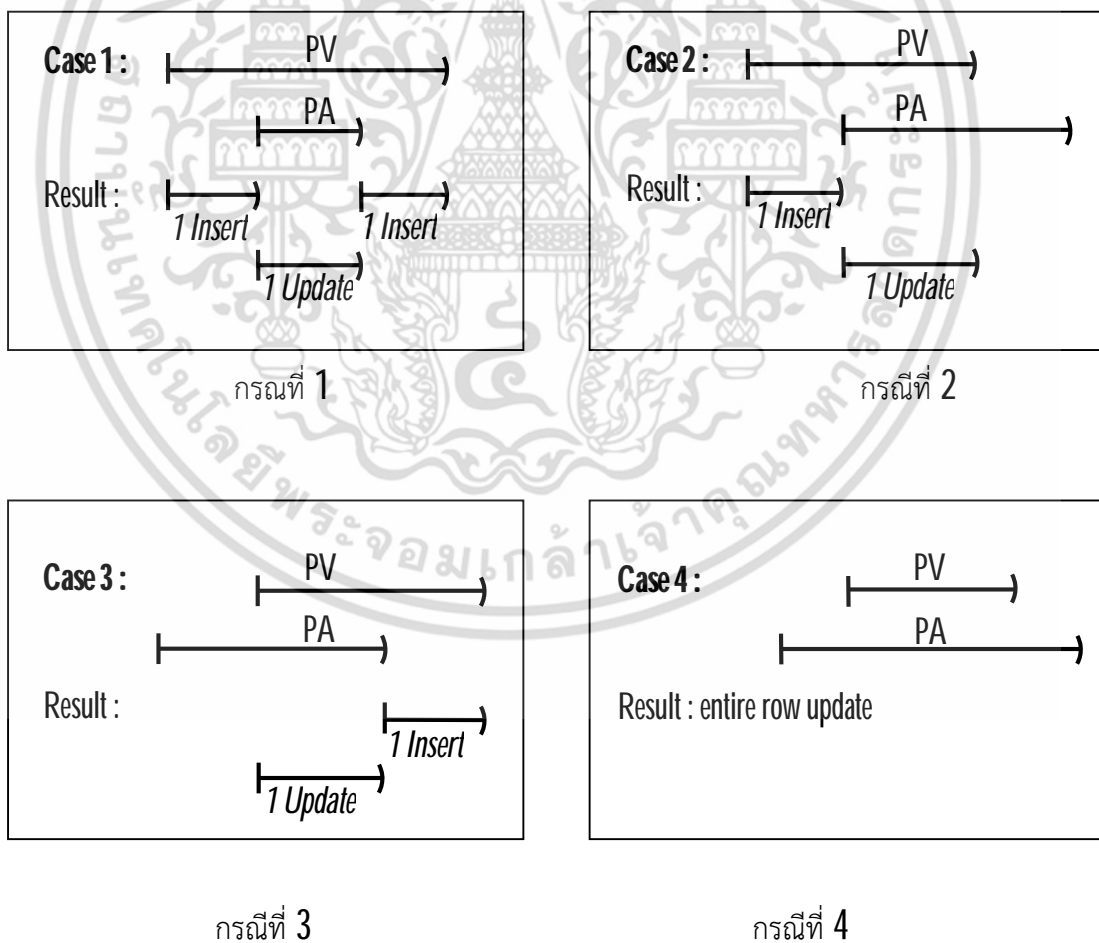
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Richard T. Snodgrass ได้นำเสนอว่า ควรมีคำสั่ง SQL ที่เขียนคำสั่งง่าย เพื่อระบุช่วงเวลา Valid Time Period ที่ต้องการลบ แล้วสามารถใช้คำสั่ง Delete ได้ตามปกติ โดยระบบฐานข้อมูลจะทำการตรวจสอบเงื่อนไขของ FROM_DATE และ TO_DATE ทั้ง 4 กรณีให้อัตโนมัติ ดังตัวอย่างคำสั่งที่เขาได้นำเสนอ คือ

```
VALIDTIME PERIOD `[1998-10-01 - 1998-10-22)'  
DELETE FROM LOT  
WHERE LOT_ID_NUM=234
```

2.3.2.3 การแก้ไขแบบซีควนซ์ (Sequence Update)

พิจารณาการเปลี่ยนแปลงที่ว่า “เปลี่ยนวัลลือท 799 เป็นวัลตอนตัวผู้ เฉพาะในเดือนมีนาคม” การเปลี่ยนแปลงแบบนี้จะต้องแบ่งออกเป็น 4 กรณี ดังแสดงในรูปที่ 2.7



รูปที่ 2.8 Sequenced Update Case1-4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

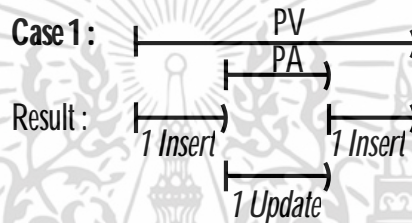
กรณีที่ 1 (Case 1) : PA เกิดขึ้นที่หลังและสิ้นสุดก่อน PV

การปรับปรุงข้อมูลเชิงเวลากรณีที่ 1 เป็นการเพิ่มแถวข้อมูล 2 Insert 1 Update ดังนี้

การเพิ่มข้อมูลแถวที่หนึ่ง วันเริ่มต้นของแถวข้อมูลมีค่าเท่ากับวันเริ่มต้นของ PV และวันสิ้นสุดของแถวข้อมูล มีค่าเท่ากับวันเริ่มต้นของ PA

การเพิ่มข้อมูลแถวที่สอง วันเริ่มต้นของแถวข้อมูลมีค่าเท่ากับวันสิ้นสุดของ PA และวันสิ้นสุดของแถวข้อมูล มีค่าเท่ากับวันสิ้นสุดของ PV

การปรับปรุงแถวข้อมูล จะปรับปรุงวันเริ่มต้นของ PV ให้มีค่าเท่ากับวันเริ่มต้นของ PA และวันสิ้นสุดของ PV มีค่าเท่ากับวันสิ้นสุดของ PA และค่าของข้อมูลจะเท่ากับค่าของ PA ที่ทำการปรับปรุง ทั้งนี้จะใช้คำสั่ง SQL มาตรฐาน 5 คำสั่ง ดังตัวอย่างคำสั่ง SQL_1 - SQL_5 จากรูปที่ 2.13



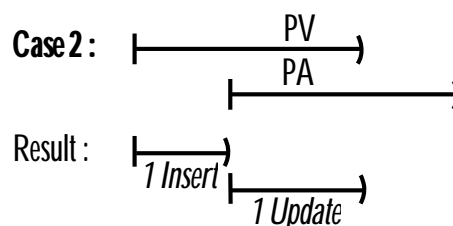
รูปที่ 2.9 Sequenced Update Case 1

กรณีที่ 2 (Case 2) : PA เกิดขึ้นและสิ้นสุดที่หลัง PV

การปรับปรุงข้อมูลเชิงเวลากรณีที่ 2 เป็นการเพิ่มข้อมูล 1 Insert 1 Update ดังนี้

การเพิ่มข้อมูล วันเริ่มต้นของแถวข้อมูลมีค่าเท่ากับวันเริ่มต้นของ PV และวันสิ้นสุดของแถวข้อมูล มีค่าเท่ากับวันเริ่มต้นของ PA

การปรับปรุงแถวข้อมูล จะปรับปรุงวันเริ่มต้นของ PV ให้มีค่าเท่ากับวันเริ่มต้นของ PA และวันสิ้นสุดของ PV มีค่าเท่ากับวันสิ้นสุดของ PA ค่าของข้อมูลเดิมจะเท่ากับค่าของ PA ที่ทำการปรับปรุง ทั้งนี้จะใช้คำสั่ง SQL มาตรฐาน 4 คำสั่ง ดังตัวอย่างคำสั่ง SQL_1 SQL_3 SQL_4 และ SQL_5 จากรูปที่ 2.13



รูปที่ 2.10 Sequenced Update Case 2

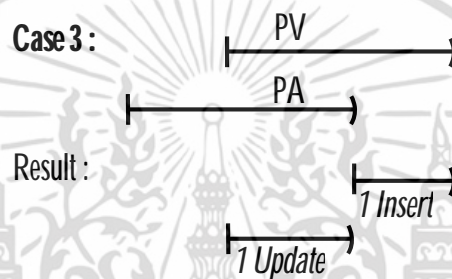
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่ 3 (Case 3) : PA เกิดขึ้นและสิ้นสุดก่อน PV

การปรับปรุงข้อมูลเชิงเวลากรณีที่ 3 เป็นการเพิ่มแถวข้อมูล 1 Insert 1 Update ดังนี้

การเพิ่มข้อมูล (1 Insert) วันเริ่มต้นของแถวข้อมูลมีค่าเท่ากับวันสิ้นสุดของ PA และวันสิ้นสุดของแถวข้อมูล มีค่าเท่ากับวันสิ้นสุดของ PV

การปรับปรุงแถวข้อมูล (1 Update) จะปรับปรุงวันเริ่มต้นของ PV ให้มีค่าเท่ากับวันเริ่มต้นของ PA และวันสิ้นสุดของ PV มีค่าเท่ากับวันสิ้นสุดของ PA ค่าของข้อมูลเดิมจะเท่ากับค่าของ PA ที่ทำการปรับปรุง ทั้งนี้จะใช้คำสั่ง SQL มาตรฐาน 4 คำสั่ง ดังตัวอย่างคำสั่ง SQL_2 SQL_3 SQL_4 และ SQL_5 จากรูปที่ 2.13



รูปที่ 2.11 Sequenced Update Case 3

กรณีที่ 4 (Case 4) : PA เกิดขึ้นก่อนและสิ้นสุดทีหลัง PV

การปรับปรุงข้อมูลเชิงเวลากรณีที่ 4 เป็นการปรับปรุงข้อมูล 1 Update คือปรับปรุงเฉพาะ

ค่าข้อมูลตาม PA แต่วันที่เริ่มต้นและสิ้นสุดยังคงเป็นวันเดิม ทั้งนี้จะใช้คำสั่ง SQL มาตรฐาน 1 คำสั่ง ดังตัวอย่างคำสั่ง SQL_3 จากรูปที่ 2.13



Result : entire row update

รูปที่ 2.12 Sequenced Update Case 4

และ Richard T. Snodgrass ได้แสดงคำสั่ง SQL การแก้ไขข้อมูลแบบซีควเอนซ์ ดังนี้

```
--SQL_1:
INSERT INTO LOT
SELECT LOT_ID_NUM, GNDR_CODE, FROM_DATRE, DATE '1998-03-01'
FROM LOT
WHERE LOT_ID_NUM = 799
      AND FROM_DATE < DATE '1998-03-01'
      AND TODATE > DATE '1998-03-01'

--SQL_2:
INSERT INTO LOT
SELECT LOT_ID_NUM, GNDR_CODE, , DATE '1998-04-01', TO_DATE
FROM LOT
WHERE LOT_ID_NUM = 799
      AND FROM_DATE < DATE '1998-04-01'
      AND TODATE > DATE '1998-04-01'

--SQL_3:
UPDATE LOT
SET GNDR_CODE = 's'
WHERE LOT_ID_NUM = 799
      AND FROM_DATE < DATE '1998-04-01'
      AND TODATE > DATE '1998-03-01'

--SQL_4:
UPDATE LOT
SET FROM_DATE = DATE '1998-03-01'
WHERE LOT_ID_NUM = 799
      AND FROM_DATE < DATE '1998-03-01'
      AND TODATE > DATE '1998-03-01'

--SQL_5:
UPDATE LOT
SET TO_DATE = DATE '1998-04-01'
WHERE LOT_ID_NUM = 799
      AND FROM_DATE < DATE '1998-04-01'
      AND TODATE > DATE '1998-04-01'
```

รูปที่ 2.13 ตัวอย่างคำสั่ง SQL มาตรฐานของ Sequenced Update

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแก้ไขแบบซีควนซ์ เป็นตัวอย่างหนึ่ง ที่จะต้องเขียนคำสั่ง SQL มาตรฐาน ทั้งหมด 5 คำสั่ง เพื่อให้การแก้ไขนั้นถูกต้อง และในแต่ละคำสั่งของ SQL มาตรฐาน จะต้องเขียนคำสั่งตรวจสอบเงื่อนไขในส่วนของ FROM_DATE และ TO_DATE ให้ถูกต้อง ซึ่งเป็นเรื่องที่ซับซ้อน และเข้าใจยาก หากเขียนเงื่อนไขผิด จะทำให้การแก้ไขข้อมูลไม่ถูกต้อง ทั้งนี้ Richard T. Snodgrass ได้นำเสนอ คำสั่ง SQL ที่ระบุช่วงเวลา ทำให้เข้าใจและเขียนคำสั่งง่ายขึ้น ดังนี้

```
VALIDTIME PERIOD '[1998-03-01 - 1998-04-01)'  
UPDATE LOT  
SET GNDR_CODE = 's'  
WHERE LOT_ID_NUM=799
```

2.4 Transaction-Time State Tables

Transaction-Time State Tables [3] คือ ตารางที่บันทึกช่วงเวลาบันทึกข้อมูลในฐานข้อมูล ซึ่งเชื่อได้ว่าข้อมูลนั้นเป็นจริง เช่น ตารางบันทึกตำแหน่งและความสว่างปรากฏของดวงดาว ซึ่งในช่วงเวลาหนึ่งเข้าใจว่า มีตำแหน่งและความสว่างปรากฏของดวงดาวเป็นค่าหนึ่ง แต่ภายหลังต่อมา เมื่อเครื่องมือที่ใช้ในการวัดมีความทันสมัยและความละเอียดแม่นยำมากขึ้น จึงพบว่า ตำแหน่งและความสว่างปรากฏของดวงดาวได้เปลี่ยนไปจากที่เคยบันทึกไว้ ดังนั้นจึงมีการแก้ไขข้อเท็จจริงในตารางหลัก และทำการบันทึกค่าที่เป็นอดีตลงในตาราง Transaction-Time State Tables

จากตารางที่ 2.9 ตาราง WDS (Washington Double Star) เป็นตารางหลักที่บันทึกตำแหน่งและความสว่างปรากฏของดวงดาว ณ ข้อมูลที่เป็นจริงในปัจจุบัน โดยมีคอลัมน์ที่สำคัญจำนวน 7 คอลัมน์ ประกอบด้วย

RA_House RA_Min และ RA_Sec คือ คอลัมน์บันทึกตำแหน่งของดวงดาวแบบเส้นลองจิจูด(เส้นนอน) มีหน่วยเป็นเวลาชั่วโมง นาที และวินาที ตามลำดับ โดยที่ 360 องศาเท่ากับ 24 ชั่วโมง หรือ 15 องศา มีค่าเท่ากับ 1 ชั่วโมง (RA ย่อมาจาก Right Ascension)

Dec_Degree Dec_Minute คือ คอลัมน์บันทึกตำแหน่งของดวงดาวแบบเส้นละติจูด (เส้นตั้ง) มีหน่วยเป็นองศา ใช้บอกระยะเชิงมุมของดวงดาวว่าอยู่ห่างจาก เส้นศูนย์สูตรฟ้า (0 องศา) ไปทางทิศเหนือ ระหว่าง 0 ถึง +90 องศา หรือไปทางทิศใต้ ระหว่าง 0 ถึง -90 องศา (Dec ย่อมาจาก Declination)

Discoverer คือ ชื่อของดวงดาว ขึ้นต้นด้วยตัวอักษรขนาด 1-3 ตัวอักษร และตามด้วยเลขของดวงดาวที่ค้นพบ

Mag_First คือ ความสว่างปรากฏของดวงดาว (Mag ย่อมาจาก Apparent Magnitude)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.9 WDS บันทึกตำแหน่งและความสว่างของดวงดาว [3]

RA_Hour	RA_Min	RA_Sec	Dec_Degree	Dec_Minute	Discoverer	Mag_First
00	00	08	75	30	'A 1248'	10.5
05	57	40	00	02	'BU 1190'	6.5
04	13	20	50	32	'CHR 15'	15.5
01	23	70	-09	55	'HJ 3433'	10.5

ตารางที่ 2.10 WDS_TT เป็นตาราง Audit Log ของตาราง WDS [3]

RA_Hour	RA_Min	RA_Sec	Dec_Degree	Dec_Minute	Discoverer	Mag_First	Trans_Start	Trans_Stop
00	00	08	75	30	'A 1248'	12.0	1989-03-12	1992-11-15
00	00	09	75	30	'A 1248'	12.0	1992-11-15	1994-05-18
00	00	09	75	30	'A 1248'	10.5	1994-05-18	1995-07-23
00	00	08	75	30	'A 1248'	10.5	1995-07-23	9999-12-31
05	57	40	00	02	'BU 1190'	6.5	1988-11-08	9999-12-31
04	13	20	50	32	'CHR 15'	15.5	1990-02-09	9999-12-31
01	23	70	-09	55	'HJ 3433'	10.5	1991-03-25	9999-12-31
02	33	10	-09	25	'LDS3402'	10.6	1993-12-19	1996-07-09

ตาราง WDS_TT เป็นตารางแบบ Transaction-Time State Tables สำหรับบันทึกการเปลี่ยนแปลงค่าตำแหน่งและความสว่างปรากฏของดวงดาวที่เป็นอดีต โดยมีการเพิ่มคอลัมน์ Trans_Start และ Trans_Stop เพื่อบันทึกช่วงเวลาที่ค้นพบ หรือช่วงเวลาที่บันทึกเวลาแก้ไขเปลี่ยนแปลงค่าตำแหน่งและความสว่างปรากฏดวงดาวในฐานข้อมูล การบันทึกช่วงเวลานี้เป็นหน้าที่ของ DBMS เป็นตัวจัดการ ผู้ใช้ทั่วไปไม่สามารถเปลี่ยนแปลงค่าเวลาที่บันทึกในตาราง WDS_TT ได้ ซึ่งบางครั้งจะเรียกตารางแบบนี้ว่า **Audit Log**

ถ้ามีการ insert หรือ update ที่ตาราง WDS เมื่อใด DBMS จะทำการเปลี่ยนแปลงแก้ไขค่าในตาราง WDS_TT โดยอัตโนมัติ คือตาราง WDS เป็นตารางบันทึกค่าตำแหน่งและความสว่างปรากฏของดวงดาวในปัจจุบัน ส่วน WDS_TT เป็นตารางบันทึกค่าตำแหน่งและความสว่างปรากฏของดวงดาวที่เป็นอดีตไปแล้ว ซึ่ง DBMS เป็นตัวจัดการโดยบันทึกช่วงเวลาที่มีการแก้ไข

2.5 Bitemporal Tables

Bitemporal Tables [3] คือ ตารางที่บันทึกเวลาการเปลี่ยนแปลงค่าของข้อมูลที่เป็นจริง และบันทึกช่วงเวลาที่ทำการเปลี่ยนแปลงข้อมูลนั้นๆ ในฐานข้อมูล หรือ ตารางที่มีการบันทึกเวลาทั้ง Transaction time และ Valid time ดังตารางที่ 2.11

จากตารางที่ 2.11 จะเห็นว่า ดวงดาว 'A 1248' ได้ถูกบันทึกเวลาที่แก้ไขข้อมูลในฐานข้อมูล (Trans_Start) ในวันที่ 12 มี.ค. 1989 ว่ามีค่าความสว่างปรากฏ 12.0 ซึ่งเชื่อได้เคยมีความสว่างปรากฏนี้ ตั้งแต่ช่วงวันที่ (Valid_From) 14 พ.ค. 1992 จนถึงปัจจุบัน

ต่อมาภายหลัง คือวันที่ 15 พ.ย. 1995 ค้นพบว่าดวงดาว 'A 1248' มีค่าความสว่างปรากฏ อยู่ 2 ช่วงเวลา คือ มีค่าความสว่างปรากฏ 12.0 ตั้งแต่วันที่ 14 พ.ค. 1992 ถึงวันที่ 16 ต.ค. 1994 และมีค่าความสว่างปรากฏ 10.5 ในช่วงวันที่ 16 ต.ค. 1994 จนถึงปัจจุบัน

ตารางที่ 2.11 Bitemporal Table [3]

Discoverer	Mag_First	Trans_Start	Trans_Stop	Valid_From	Valid_To
'A 1248'	12.0	1989-03-12	1995-11-15	1922-05-14	9999-12-31
'A 1248'	12.0	1995-11-15	9999-12-31	1922-05-14	1994-10-16
'A 1248'	10.5	1995-11-15	9999-12-31	1994-10-16	9999-12-31

2.6 Temporal Type and Modules

2.6.1 Temporal Type

นิยาม Temporal Type เป็นการแปลง μ จากเซตของจำนวนเต็มบวก (เวลาขณะหนึ่ง : The time sticks) ถึง 2^R (the set of absolute time sets) ดังนั้นทุกค่าจำนวนเต็มบวก i และ j ซึ่ง $i < j$ โดยมีสองเงื่อนไขดังนี้ [4]

1. $\mu(i) \neq 0$ และ $\mu(j) \neq 0$ แล้ว ค่าจำนวนจริงใน $\mu(i)$ ต้องน้อยกว่าจำนวนจริงใน $\mu(j)$
2. $\mu(i) = 0$ แล้ว $\mu(j) = 0$ เช่นกัน

μ คือ สัญลักษณ์แทนเวลา เช่น วัน เดือน สัปดาห์ และปี ดังนั้นจากนิยาม ประเภทเชิงเวลา จะเป็นช่วงเวลาที่เป็นค่าบวก โดยถ้าเวลาเริ่มต้นและเวลาสิ้นสุดไม่เท่ากับ 0 แล้ว เวลาเริ่มต้นจะต้องมีค่าน้อยกว่าเวลาสิ้นสุดเสมอ เช่น เวลาเริ่มต้นที่ปี 1800 : Year(1) ถึงเวลาสิ้นสุดปี 1801 : Year(2) แต่ถ้าเวลาเริ่มต้นเท่ากับ 0 แล้ว เวลาสิ้นสุดจะต้องเท่ากับ 0 เสมอ คือไม่มีช่วงเวลาเกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2 Temporal module

นิยาม Temporal module scheme จะเขียนอยู่ในรูป (R, μ) โดยที่ R เป็นโครงร่างตาราง และ μ คือ ช่วงระยะเวลา (Temporal type) ทั้งนี้ Temporal Modules เขียนในรูป (R, μ, ϕ) เมื่อ

(R, μ) คือ โครงร่างเชิงเวลา (Temporal Scheme)

ϕ คือ Windowing Function Form \mathbf{N} to $2^{\text{Tup}(R)}$ โดยที่ $\phi(i)=0$ สำหรับ i ซึ่ง $\mu(i)=0$

Time Windowing Function ϕ ในรูปแบบ Temporal Modules (R, μ, ϕ) คือ ช่วงเวลาที่ Fact มีค่าคงที่

2.7 Temporal Functional Dependencies : TFD

นิยาม ให้ X และ Y เป็นเซตจำกัดของ แอทริบิวต์ และ μ คือประเภทเชิงเวลา โดย $\mu(i) \neq 0$ สำหรับ i ดังนั้น $X \rightarrow_{\mu} Y$ เรียกว่า Temporal Function Dependency (TFD)

ฐานข้อมูลเชิงสัมพันธ์ (Relation Database) ตามแบบเดิมจะการบันทึกข้อสารสนเทศที่เป็นค่าคง (Static Information) หรือ ข้อสนเทศในปัจจุบันเพียงอย่างเดียว คือ FD $X \rightarrow Y$ เมื่อใดก็ตามที่ความสัมพันธ์มีสองแถวข้อมูลของแอทริบิวต์เดียวกันกับ X และเป็นแอทริบิวต์เดียวกันกับ Y ยกตัวอย่างเช่น

ตัวอย่างที่ 1 พิจารณาโครงสร้างตาราง คณะ (Faculty) จะบันทึกเลขประกันสังคม (SSn) ชื่อ (Name) ตำแหน่ง (Rank) และแผนกงาน (Dept) จะได้ฟังก์ชัน FD $SSn \rightarrow Rank$ [5] บุคลากรของคณะจะมีสถานะตำแหน่งได้เพียงตำแหน่งเดียว ถึงแม้ว่าเขาจะทำหน้าที่ในหลายแผนกก็ตาม

ในระบบฐานข้อมูลเชิงเวลา ข้อสนเทศจะเปลี่ยนแปลงได้ (Dynamic) แต่ถ้าเป็น FD แบบเดิม ตารางบันทึกข้อมูลปัจจุบัน จะไม่มีการเปลี่ยนแปลง ไม่สอดคล้องกับ ฐานข้อมูลเชิงเวลา เช่นในกรณี คณะ (FACULTY) ถ้าเป็นตารางแบบเชิงเวลาจะบันทึกมากกว่า 2 ความสัมพันธ์ เช่น สถานะตำแหน่งบุคลากรท่านหนึ่งในคณะดำรงตำแหน่ง ผู้ช่วยศาสตราจารย์ ในเวลาต่อมาบุคลากรคนดังกล่าวได้รับให้ดำรงตำแหน่งเป็น รองศาสตราจารย์ เป็นต้น

ตัวอย่างที่ 2 ตารางต่อไปนี้เป็นตารางของข้อมูลเชิงเวลา [5] ชื่อตาราง ACCOUNTS ซึ่งทำการบันทึกช่วงเวลาการเปลี่ยนแปลงข้อมูลของแต่ละเลขบัญชี (AcctNo) จำนวนเงินที่เปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Amount) ยอดเงินคงเหลือ (Balance) ดอกเบี้ยสะสมต่อเดือน (Accumulated interest : AccumInt) และเวลาทำการเปลี่ยนแปลงข้อมูลบัญชี (Time)

ดอกเบี้ยสะสม จะคำนวณ ในอัตรา ร้อยละ 5 ต่อปี โดยคำนวณทุกวัน โดยเริ่มต้นของแต่ละวัน (แต่ไม่เกิดขึ้นในบัญชี จนกระทั่งสิ้นเดือน) สำหรับค่าดอกเบี้ยสะสมใน 1 วันจะไม่มี การเปลี่ยนแปลง จนกระทั่งเปลี่ยนไปอีกวันหนึ่ง ถึงจะเปลี่ยนแปลงดอกเบี้ยสะสม

ค่าเวลา เป็น Timestamp ซึ่งประกอบด้วย เดือน วัน ปี (month / day / year) และรวม บันทึกลงเวลาของวัน (จนถึง วินาที) เพื่อบันทึกการเปลี่ยนแปลงข้อมูล โดยเราจะถือว่า ทุกๆ เวลาที่ทำการเปลี่ยนแปลง จะเป็นค่าช่วงเวลาหรือค่าที่ไม่ซ้ำกัน (Unique Timestamp) ดังตารางที่ 2.12

ตารางที่ 2.12 การเปลี่ยนแปลงข้อมูลของแต่ละเลขบัญชี (ACCOUNTS) [4]

AcctNo	Amount	Balance	AccumInt	Time
1001	+1000	1000	0.00	3/3/93:09:01:00
1001	-500	500	0.14	3/4/93:10:01:55
1001	+200	700	0.14	3/4/93:11:00:00
1001	-315	385	0.14	3/4/93:12:19:03
1001	-255	130	0.14	3/4/93:18:00:00
1001	-10	120	0.19	3/7/93:09:00:00
1001	+100	220	0.19	3/7/93:12:01:40

ตาราง ACCOUNTS แสดงข้อมูลที่ซ้ำซ้อน ด้วยค่าที่เหมือนกันของ AccumInt ซึ่งมีความซ้ำซ้อนอยู่หลายช่วงเวลาในแต่ละวัน ดังนั้นสามารถเขียน FD ที่เกี่ยวข้องกับ ACCOUNTS ได้ดังนี้

AcctNo, Time \rightarrow Amount, Balance, AccumInt.

จากส่วนของ FD เป็นโครงสร้างของ BCNF (คือ ทุกๆ Determinant ด้านซ้ายของ FD ต้องเป็น (Candidate key) และ ความซ้ำซ้อนของตาราง ACCOUNTS คือ ดอกเบี้ยสะสม (AccumInt) สำหรับแต่ละเลขบัญชี (AcctNo) จะเห็นว่าไม่มีการเปลี่ยนแปลงในวันที่ 4 มีนาคม 1993 ดังนั้นภายใต้เวลาในตาราง ACCOUNTS คือ วินาที ไม่ใช่วัน ซึ่งกฎข้อบังคับของความถูกต้อง (Constraint) ไม่สามารถแสดงให้เห็นภาพโดย FD แบบดั้งเดิมได้

จากตัวอย่างที่ 2 และคุณสมบัติของ μ และ TFD สามารถแสดงข้อมูลให้สอดคล้องกับเชิงเวลา โดยขจัดความซ้ำซ้อนของดอกเบี้ยสะสมต่อเดือน โดยแยกเป็น 2 ตารางคือ ตารางบันทึกข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปแบบเวลาที่เป็นวินาที (TRANSACTION-INFO) และตารางบันทึกดอกเบี้ยสะสมต่อเดือนของแต่ละวัน (ACCU-INTEREST) ดังตารางที่ 2.13 และตารางที่ 2.14 และเขียน TFD ดังนี้

TFD AcctNo \rightarrow_{day} AccumInt
 TFD AcctNo \rightarrow_{second} Balance

ตารางที่ 2.13 TRANSACTION-INFO [4]

AcctNo	Amount	Balance	Time
1001	+1000	1000	3/3/93:09:01:00
1001	-500	500	3/4/93:10:01:55
1001	+200	700	3/4/93:11:00:00
1001	-315	385	3/4/93:12:19:03
1001	-255	130	3/4/93:18:00:00
1001	-10	120	3/7/93:09:00:00
1001	+100	220	3/7/93:12:01:40

ตารางที่ 2.14 ACCU-INTEREST [4]

AcctNo	AccumInt	Day
1001	0.00	3/3/93
1001	0.14	3/4/93
1001	0.19	3/7/93

2.8 Temporal Normalization

2.8.1 Time Normal Form (TNF)

นิยาม : ตารางที่ไม่ใช่ Time Normal Form (TNF) ถ้าตารางใดมีแอทริบิวต์เชิงเวลา (Temporal Attribute) อย่างน้อย 1 แอทริบิวต์ ซึ่งมีแถวข้อมูลตั้งแต่ 2 แถวหรือมากกว่า ที่ขอบเขตช่วงเวลาเริ่มต้นและสิ้นสุด (Consecutive Time Intervals : Period) ต่อเนื่องกัน และค่าข้อมูลเท่ากัน คือ มีความซ้ำซ้อนส่วนใดส่วนหนึ่ง ถือว่าไม่เป็น TNF (Non-TNF) [6]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนิยาม ตารางที่ 2.15 ไม่เป็น TNF เพราะมีค่าความซ้ำซ้อนที่แอทริบิวต์ HOTEL และ CITY และไม่สามารถแยกตารางได้ เพราะค่า COST ขึ้นอยู่กับแอทริบิวต์ร่วมกัน (Combination) ระหว่าง HOTEL CITY T_s และ T_e

ตารางที่ 2.15 The EXPENSE Relation [6]

E#	HOTEL	CITY	COST	T_s	T_e
E1	HILTON	LA	950	1	5
E1	SHERATION	LA	900	6	12
E1	SHERATION	NYC	400	13	16
E1	HILTON	SF	300	17	18
E1	HILTON	SF	350	19	20

จากนิยาม ตารางที่ 2.16 ไม่เป็น TNF เพราะมีค่าความซ้ำซ้อนที่ SALARY และ DEPARTMENT

ตารางที่ 2.16 The SAL-DEPT Relation [6]

E#	SALARY	DEPARTMENT	T_s	T_e
E1	20K	ACCOUNTING	1	4
E1	20K	FINANCE	5	8
E1	25K	FINANCE	9	10
E1	25K	MARKETING	11	20
E1	25K	MARKETING	21	25

2.8.2 First Temporal Normal Form (1TNF)

นิยามที่ 1 : ตารางความสัมพันธ์ R ที่ประกอบด้วยแอทริบิวต์ $S, A_1, \dots, A_n, t_s, t_e$ จะเป็น 1TNF ถ้าไม่ปรากฏว่า 2 แถวข้อมูล r^1 และ r^2 ที่มีซุเปอร์คีย์ $s^1 = s^2$ และขอบเขตระยะเวลาที่ $[t_s^1, t_e^1]$ กับ $[t_s^2, t_e^2]$ คาบเกี่ยวกัน

ทั้งนี้ บทความ [6] ได้กล่าวเพิ่มเติมว่า Temporal Normal Form มีความคล้ายคลึงกับ 1NF ของระบบฐานข้อมูลแบบเชิงสัมพันธ์ และถือว่าอ้างอิงเป็น 1TNF ด้วย

จากนิยาม ตารางที่ 2.17 ไม่เป็น 1TNF เพราะที่จุดเวลา 9 และ 10 มีค่าข้อมูลเงินเดือนซ้ำ

ตารางที่ 2.17 The SAL Relation

E#	SALARY	T_s	T_e
E1	20K	1	5
E1	25K	6	10
E1	27K	9	11

นิยามที่ 2 : ตารางความสัมพันธ์ R ที่ประกอบด้วยแอทริบิวต์ $S, A_1, \dots, A_n, t_s, t_e$ จะเป็น 1TNF ถ้าไม่ปรากฏว่า 2 แถวข้อมูล r^1 และ r^2 ที่มีซุเปอร์คีย์ $s^1 = s^2$ และจุดเวลา $t^1 = t^2$

ตารางที่ 2.18 A Temporal Relation

S	T	A
s_1	t_1^1	a_1^1
s_1	t_2^1	a_2^1
\vdots	\vdots	\vdots
s_2	t_1^2	a_1^2
s_2	t_2^2	a_2^2
\vdots	\vdots	\vdots

2.8.3 Temporal Boyce-Codd Normal Form (TBCNF)

ในส่วนของ Temporal BCNF จะมีความคล้ายคลึงกับ Boyce-Codd Normal Form (BCNF) โดย Temporal BCNF (TBCNF) ยังคงไว้หลักการของ BCNF โดยที่ TBCNF จะไม่ยอมรับความซ้ำซ้อนที่นำเสนอด้วย TFD จะทำการแยกส่วนประกอบ อัลกอริทึม ของ TBCNF อยู่ในรูปแบบ Temporal Module Schema ในประการแรก เราจะกล่าวถึง คีย์ (Keys)

$$\forall X \xrightarrow{T} Y \in F^+ (Y \subseteq X \vee X \xrightarrow{T} R)$$

นิยาม ทุกๆ ค่าที่ X กำหนด Y ณ เวลาใด เป็นสมาชิกของ F โดย Y ไม่เป็นซัพเซตของ X หรือ X กำหนด R ณ เวลาใดๆ [1]

2.8.4 Temporal third normal form (T3NF)

นิยาม : (R, F) เป็น T3NF ถ้า ทุกๆ ค่าของ Non-Trivial TFD ที่ Dependencies $X \xrightarrow{T} Y$ ใน F^+ โดยที่ X เป็น Temporal Superkey สำหรับ R หรือ แอทริบิวท์ของ Y เป็นส่วนของ F^+ เซตทุกค่าของ F [1]

2.8.5 Temporal fourth normal form (T4NF)

นิยาม T4NF เหมือนกับ TBCNF แต่ยัง Temporal Multivalued Dependencies [1]

2.9 Temporal Data Type

ชนิดข้อมูลเชิงเวลา สามารถแบ่งเป็น 3 ประเภท ดังนี้ [7]

2.9.1 Instant

คือ จุดของเวลา เช่น วันที่ 1 มกราคม 2557 หรือ เวลา 12.00 น

2.9.2 Duration

คือ ความยาวของเวลา เช่น 2 สัปดาห์ ซึ่งคำนี้จะใช้เช่นเดียวกันใน ISO 8601 และ XML แต่ใน SQL มาตรฐาน จะเรียกว่า "Interval"

2.9.3 Period

คือ ช่วงขอบเขตเริ่มต้นและสิ้นสุดของช่วงระยะเวลา เช่น 4 กรกฎาคม ค.ศ.2008 ถึงวันที่ 7 กรกฎาคม ค.ศ. 2008 ซึ่งคำว่า Period จะใช้ใน SQL/Temporal แต่ใน ISO8601 จะเรียกว่า "Time Interval"

2.10 แบบจำลองที่โออาร์เอ็ม (Temporal Object-Role Modeling : TORM)

วิธีการแบบจำลองที่โออาร์เอ็ม (Temporal Object-Role Modeling : TORM) หรือ ทีไนแอม (TNIAM : Temporal Nijssen's Information Analysis Methodology) เป็นเครื่องมือช่วยในการออกแบบฐานข้อมูล โดยขยายแนวคิดจากวิธีการไนแอมให้สามารถรองรับข้อมูลที่เปลี่ยนแปลงไปตามเวลา โดยมีกระบวนการในการออกแบบทั้งในระดับแนวคิด และกระบวนการแปลงโครงสร้างตารางข้อมูลในระดับแนวคิดเป็นโครงสร้างตารางข้อมูลเชิงสัมพันธ์ที่อยู่ในรูปของนอร์มอลฟอร์มระดับที่ 5 (5NF) [8] [9]

2.10.1 Temporal Entity Type

คือ กลุ่มของนามธรรมหรือเอนทิตีจริง (Real Entity) ซึ่งอาจเป็นสิ่งที่จับต้องได้ หรือไม่ได้ เช่น บุคคล แผนก บริษัท สัญลักษณ์ที่ใช้แทนชนิดเอนทิตี เป็นรูปวงรีเส้นทึบ โดยมีการระบุชื่อชนิดเอนทิตีที่อยู่ภายในวงรี รูปแบบของเวลาที่สนับสนุนชนิดเอนทิตี ได้แก่รูปแบบจำลองของชนิดเอนทิตี และเวลาที่ข้อมูลถูกจัดเก็บลงสู่ฐานข้อมูล การระบุรูปแบบของชนิดเอนทิตี และเวลาที่ข้อมูลถูกจัดเก็บลงสู่ฐานข้อมูลให้กับชนิดเอนทิตี ขึ้นอยู่กับการพิจารณาและการตัดสินใจของผู้ออกแบบฐานข้อมูล โดยจะพิจารณาจากเงื่อนไขที่กำหนดในการออกแบบฐานข้อมูล ชนิดเอนทิตีใดที่มีการระบุรูปแบบเวลาจะมีการระบุรูปแบบเวลาไว้ที่มุมขวาด้านบนของวงรีตามชนิดของเวลาที่ต้องการ เช่น

ชนิดเอนทิตี ที่มีการระบุรูปแบบของเวลาที่สนับสนุนชนิดเอนทิตี ก็จะใช้ "LS" และชนิดเอนทิตีที่มีการระบุค่าเวลาที่ข้อมูลถูกจัดเก็บลงสู่ฐานข้อมูลก็จะใช้ "TT" และสำหรับกรณีชนิดเอนทิตีที่มีการระบุทั้งค่าเวลาของชนิดเอนทิตีและเวลาที่ข้อมูลถูกจัดเก็บลงสู่ฐานข้อมูลก็จะทำการใส่ "LT" ไว้มุมขวาด้านบนของชนิดเอนทิตีนั้น ชนิดเอนทิตีใด มีการระบุรูปแบบของเวลาไว้เรียกชนิดเอนทิตีนั้นว่า เอนทิตีชนิดเชิงเวลา (Temporal Entity Type) ดังแสดงในรูปที่ 2.14 และชนิดเอนทิตีใดไม่มีการระบุรูปแบบของเวลาเรียกชนิดเอนทิตีนั้นว่า เอนทิตีทั่วไปที่ไม่มีการกำหนดชนิดเวลา (Non-Temporal Entity Type)

TEACHER LS
(T#)

รูปที่ 2.14 Temporal Entity Type with Own Life Span

2.10.2 Temporal Fact Type

คือ ความสัมพันธ์ระหว่างเอนทิตีเชิงเวลา ดังตัวอย่างรูปที่ 2.5

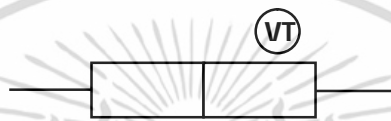


รูปที่ 2.15 Label Type DepName

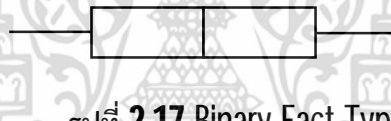
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.3 Elementary Fact Type

เป็นตัวที่ใช้แสดงความสัมพันธ์ระหว่างชนิดแอนติที่สองแอนติที่ขึ้นไป สัญลักษณ์ที่ใช้เป็นรูปสี่เหลี่ยมผืนผ้าติดกันสองรูปขึ้นไป ซึ่งก็จะขึ้นอยู่กับชนิดแอนติที่ที่แสดงความสัมพันธ์กันอยู่ว่าประกอบไปด้วยจำนวนเท่าใด รูปแบบเวลาที่สนับสนุนได้แก่เวลาที่ข้อมูลเป็นจริง การระบุรูปแบบเวลาจะใช้สัญลักษณ์ "VT" และมีวงกลมล้อมรอบ โดยทำการระบุไว้บนอีลิเมนต์อาร์แฟค สำหรับอีลิเมนต์อาร์แฟคที่มีการระบุค่าเวลาเรียกว่า อีลิเมนต์อาร์แฟคชนิดเวลา (Temporal Elementary Fact Type) ดังแสดงในรูปที่ 2.16 สำหรับอีลิเมนต์อาร์แฟคที่ไม่มีการระบุค่าเวลา เรียกว่า อีลิเมนต์อาร์แฟคที่ไม่เป็นชนิดเวลา (Non-Temporal Elementary Fact Type) ดังแสดงได้ที่รูป 2.17



รูปที่ 2.16 Temporal Binary Fact Type



รูปที่ 2.17 Binary Fact Type

2.10.4 ชนิดรีเฟอเรนซ์ (Referent Type) [3] เป็นตัวที่ใช้แสดงความสัมพันธ์ระหว่างชนิดแอนติที่กับชนิดลาเบลที่ขึ้นกับชนิดแอนติที่นั้นๆ

2.10.5 Integrity Constraint [3] เป็นตัวที่ใช้เป็นเงื่อนไขที่จะทำให้ชัดเจนว่า ฐานข้อมูลนั้นถูกต้อง และสมบูรณ์ตลอดเวลา (Consistency and Integrity) ประกอบด้วย

2.10.5.1 Inter Fact Type Uniqueness Constraint เป็นตัวบอกว่าแอนติที่หรือลาเบลที่เชื่อมกันนั้น ร่วมกันเป็นคีย์ร่วม ดังรูปที่ 2.18



รูปที่ 2.18 สัญลักษณ์แทน Inter Fact Type Uniqueness Constraint

2.10.5.2 Intra Fact Type Uniqueness Constraint ดังรูปที่ 2.19



รูปที่ 2.19 สัญลักษณ์แทน Intra Fact Type Uniqueness Constraint

2.10.5.3 Mandatory Fact Type Constraint ตัวอย่างดังรูปที่ 2.20 ทั้งนี้ในการออกแบบถ้าต้องการ Mandatory Fact Type Constraint ที่ชนิดเอนทิตีไหนก็ใส่จุดดำกลมที่ชนิดเอนทิตีนั้น

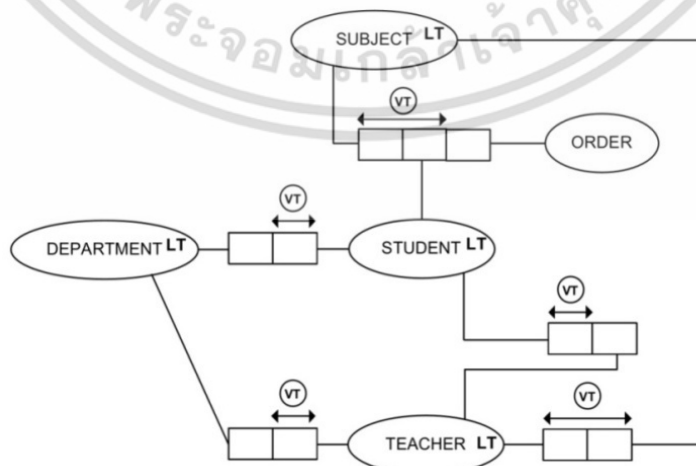


รูปที่ 2.20 การใช้ Mandatory Fact Type Constraint

2.10.6 ขั้นตอนเขียนแบบจำลองที่โออาร์เอ็ม

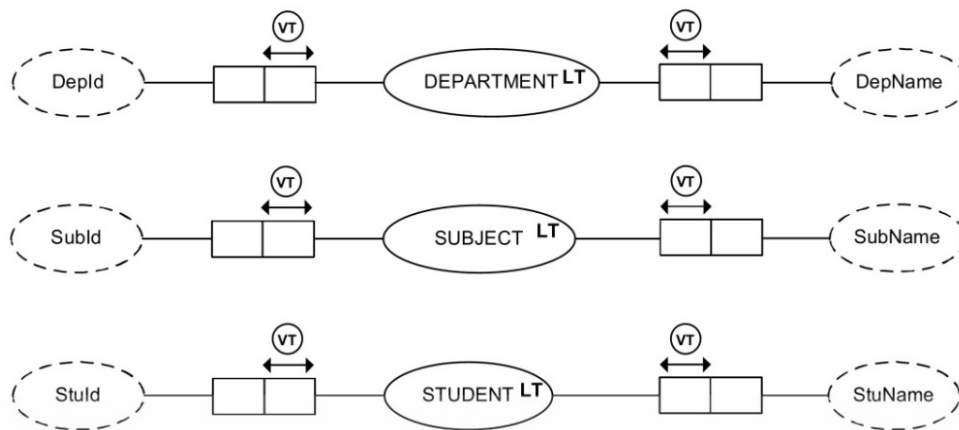
ขั้นตอนการออกแบบฐานข้อมูลเชิงเวลาด้วยแบบจำลองที่โออาร์เอ็ม ประกอบด้วย 2 ขั้นตอนได้แก่ ขั้นตอนการเขียนแบบจำลองที่โออาร์เอ็ม และขั้นตอนการแปลงแบบจำลองโออาร์เอ็ม เป็นแบบโครงร่างตารางเชิงสัมพันธ์

2.10.6.1 ขั้นตอนการเขียนแบบจำลองที่โออาร์เอ็ม ประกอบด้วย 2 ขั้นตอนได้แก่ ขั้นตอนการเขียนโครงร่างหลัก (Main Schema) ดังรูปที่ 2.21 และขั้นตอนการเขียนโครงร่างย่อย (Sub Schema) ดังรูปที่ 2.22



รูปที่ 2.21 ตัวอย่างการเขียนโครงร่างหลักของระบบสารสนเทศนักศึกษาเชิงเวลา [8]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

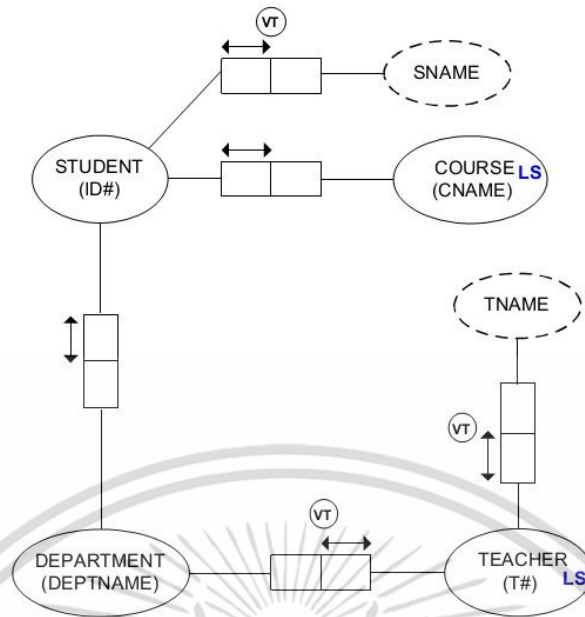


รูปที่ 2.22 ตัวอย่างการเขียนโครงร่างย่อยระบบสารสนเทศนักศึกษาเชิงเวลา [8]

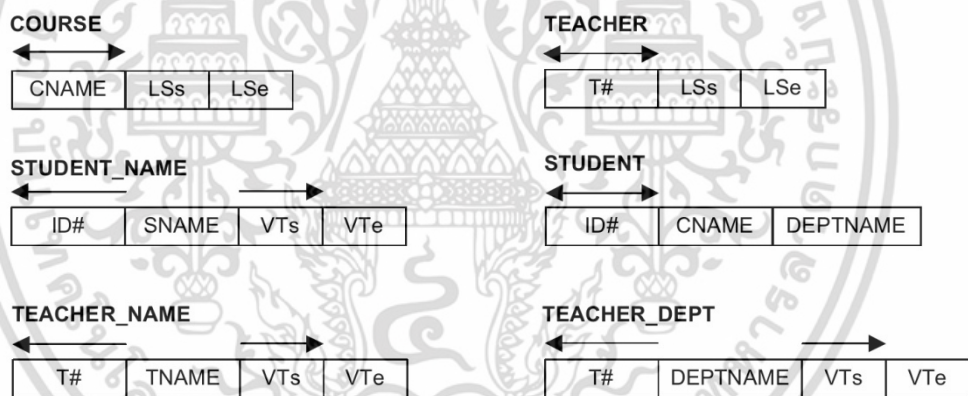
2.10.6.2 ขั้นตอนการแปลงแบบจำลองที่โออาร์เอ็ม เป็น โครงร่างตารางแบบเชิงสัมพันธ์ มี 4 ขั้นตอนดังนี้

1. เขียนโครงร่างตารางสำหรับ **Binary Temporal Fact Type** นำคีย์หลัก (PK) ของ **Entity Type** ที่เกี่ยวข้องกันเข้าร่วมแอทริบิวต์ของโครงร่างตาราง
2. เขียนโครงร่างตารางสำหรับ **Fact Type** ซึ่งเป็น **Intra-Fact Type Uniqueness Constraint** นำคีย์หลัก (PK) ของ **Entity type** ที่เกี่ยวข้องกับ **Fact Type** ร่วมเป็นแอทริบิวต์ในโครงร่างตาราง และถ้ามีการแสดง **Timestamp(VT, TT, BT)** ที่ติดกับ **Temporal Fact Type** ให้เขียนแอทริบิวต์ **Timestamp**
3. เขียนโครงร่างตารางเชิงสัมพันธ์ สำหรับ **N-Ary Fact Type (n>2)** โดยนำคีย์หลัก (PK) ของ **Fact Type** ที่เกี่ยวข้องมาเขียนเข้าด้วยกันกับโครงร่างตารางนี้ และถ้ามีการแสดง **Timestamp(VT, TT, BT)** ที่ติดกับ **Temporal Fact Type** ให้เขียนด้วย
4. เขียนโครงร่างตารางเชิงสัมพันธ์ ส่วน **Non-Temporal** เช่น **Binary Fact Type Reference Type**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.23 A Temporal ORM Version of the ORM Conceptual Schema [9]



รูปที่ 2.24 Corresponding Relational Database Schemas of the TORM ของรูปที่ 2.13 [9]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การทดลองคำสั่งจัดการข้อมูลแบบเชิงเวลา

จากบทที่ผ่านมา จะเห็นว่าการจัดการฐานข้อมูลเชิงเวลา ในส่วนของการเพิ่มข้อมูล การปรับปรุงข้อมูล การลบข้อมูล ตลอดจนการสืบค้นข้อมูล ระบบฐานข้อมูลแบบเดิม (Non-Temporal) สามารถที่จะเพิ่มคอลัมน์วันเริ่มต้นและวันสิ้นสุดของข้อมูลได้ก็ตาม แต่ถ้าใช้คำสั่งในการเข้าถึงข้อมูลเป็นภาษาเอสคิวแคว (SQL) แบบมาตรฐาน จะต้องเขียนคำสั่งตรวจสอบช่วงเวลาต่างๆ ซึ่งบางกรณีจะเขียนคำสั่งถึง 5 คำสั่ง เพื่อให้การปฏิบัติการในการจัดการข้อมูลแบบเชิงเวลาได้ถูกต้อง

ในบทนี้ ผู้ทดลองได้ใช้โปรแกรมบริหารจัดการฐานข้อมูล Oracle11g [8] เพื่อทดลองคำสั่งการจัดการข้อมูลเชิงเวลาต่างๆ ดังนี้

3.1 การสร้างตารางแบบเชิงเวลา (Valid Time State Table)

3.1.1 สร้างตารางชื่อ Positions ประกอบด้วยคอลัมน์ Emp_ID=รหัสพนักงาน และ Job_position=ชื่อตำแหน่งงาน

```
CREATE TABLE positions(  
  emp_id      NUMBER PRIMARY KEY,  
  job_position VARCHAR2(30)  
);
```

CREATE TABLE succeeded.

3.1.2 ใช้คำสั่ง DESC ตรวจสอบดูโครงสร้างของตาราง Positions จะสังเกตมี 2 คอลัมน์ คือ Emp_ID และ Job_posititon

```
DESC positions;
```

Name	Null	Type
EMP_ID	NOT NULL	NUMBER
JOB_POSITION		VARCHAR2(30)

2 rows selected

3.1.3 ทำ EnablesVersioning ให้ตารางสามารถสนับสนุน Valid Time โดยระบบจะเพิ่มคอลัมน์ชื่อ WM_VALID ให้อัตโนมัติ ซึ่งคอลัมน์นี้จะเป็นข้อมูลแบบ WM_PERIOD

```
EXECUTE DBMS_WM.EnableVersioning('positions',  
'VIEW_WO_OVERWRITE',FALSE, TRUE);
```

3.1.4 รูปแบบคำสั่ง EnableVersioning

```
DBMS_WM.EnableVersioning(table_name, hist, isTopology,
validTime, undo_space, validTimeRange)
```

table_name คือ ชื่อตาราง โดยมีความยาวไม่เกิน **25** ตัวอักษร ในตารางจะต้องไม่มีคอลัมน์ที่มีชื่อขึ้นต้นด้วย **WM_** หรือ **WM\$** ชื่อตารางและชื่อคอลัมน์ห้ามมีสัญลักษณ์ พิเศษ เช่น ! ? หรือ * เป็นต้น และชื่อตารางไม่เป็น **case-sensitive**

hist เป็นการกำหนดว่าจะให้มีเขียนทับ หรือไม่เขียนทับ เมื่อมีการเพิ่มปรับปรุงข้อมูล (**Update**) ในตารางต้นฉบับ โดยมีพารามิเตอร์ ดังนี้

NONE ค่า default เป็นการกำหนดไม่ต้องมีการแก้ไขในตารางประวัติ **_hist**

VIEW_W_OVERWRITE เขียนข้อมูลทับ (**with Overwrite**) ในตาราง **_hist** ได้

VIEW_WO_OVERWRITE ไม่เขียนข้อมูลทับ (**without Overwrite**) ในตาราง **_hist**

คือ อนุญาตให้เพิ่มข้อมูลได้

isTopology กำหนดการสนับสนุนแบบ **Spatial** หรือไม่ โดยมีพารามิเตอร์ ดังนี้

TRUE เป็นตารางแบบ **Spatial topology**

FALSE* ไม่เป็นตารางแบบ **Spatial topology**

ValidTime กำหนดการสนับสนุนตารางแบบ **Valid Time State Table** ดังนี้

TRUE เป็นการกำหนดให้เป็นตารางแบบ **Valid Time Support**

FALSE* เป็นการกำหนดเป็นตาราง **Transaction Time Support**

Undo_space กำหนดค่าสูงสุดของ **String** หรือ **Number** สำหรับการคืนค่าในกรณี undo_space WM

ValidTimeRange กำหนดรูปแบบของประเภทของมูล ดังนี้

TRUE * กำหนดให้ **Object Type** เป็นแบบ **WM_PERIOD**

FALSE กำหนดให้ ไม่เป็นแบบ **Objet Type** เป็นแบบ **WM_PERIOD**

หมายเหตุ * เป็นค่าพื้นฐาน (Default)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5 ใช้คำสั่ง DESC เพื่อดูโครงสร้างของตาราง Names จะเห็นว่า มีคอลัมน์ WM_VALID เพิ่มขึ้นมา

```
DESC POSITIONS;
```

Name	Null	Type
EMP_ID		NUMBER
JOB_POSITION		VARCHAR2 (30)
WM_VALID		WM_PERIOD ()

3 rows selected

คอลัมน์ WM_VALID จะมีประเภทข้อมูลเป็นแบบ WMSYS.WM_PERIOD และภายในประเภทข้อมูล นี้จะมี 2 คอลัมน์ย่อยคือ VALIDFROM และ VALIDTILL
 VALIDFROM คือ คอลัมน์บันทึกวันเริ่มต้นของข้อมูล (FROM_DATE)
 VALIDTILL คือ คอลัมน์บันทึกวันสิ้นสุดของข้อมูล (TO_DATE)

3.2 การทดลองการปรับปรุงข้อมูล ณ เวลาปัจจุบัน (Current Modification)

3.2.1 Current Update Case 1, in the General Scenario

จากรูปที่ 3.1 เมื่อมีการปรับปรุงข้อมูลในอดีต ณ เวลาปัจจุบัน จะไม่มีการเปลี่ยนแปลงข้อมูลใดๆ เกิดขึ้น เนื่องจาก ข้อมูลนั้นได้ปรับปรุงเสร็จสิ้นเรียบร้อยแล้วตั้งแต่อดีต

Case 1:



Result : (unchanged)

now

รูปที่ 3.1 Current Update Case 1, in the General Scenario

1. แก้ไขระบบวันที่ของเครื่องคอมพิวเตอร์ เป็นวันที่ 2012-03-05
2. เพิ่มข้อมูลพนักงานรหัส 45 ตำแหน่ง System Analyst ลงในตาราง Positions ตั้งแต่วันที่ 2005-02-16 ถึงวันที่ 2008-01-01

```
INSERT INTO positions VALUES(
  45, 'System Analyst',
  WMSYS.WM_PERIOD(TO_DATE('2005-02-16','YYYY-MM-DD'),
    TO_DATE('2008-01-01','YYYY-MM-DD'))
);
```

1 rows inserted

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ใช้คำสั่ง **SELECT** เพื่อแสดงผลการเพิ่มข้อมูลพนักงานรหัส 45 โดยใช้คำสั่ง **SetValidTime** เพื่อแสดงข้อมูลในอดีตได้ เนื่องจากข้อมูลพนักงานรหัส 45 กลายเป็นอดีตแล้ว

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);
```

```
SELECT emp_id, job_position,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM positions e;
```

anonymous block completed

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
45	System Analyst	2005-02-16	2008-01-01

1 rows selected

หมายเหตุ

DBMS_WM.MIN_TIME คือ เวลายน้อยสุด 1 ม.ค. ค.ศ. -4712 (4712 BCE)

DBMS_WM.MAX_TIME คือ เวลามากสุด 31 ธ.ค. ค.ศ. 9999 (11.59.999999000pm)

Anonymous Block Completed คือ ข้อความแสดงการ Run Script เรียบร้อยแล้ว

Anonymous Block เป็นการแจ้งว่า รูปแบบการเขียนคำสั่ง PL/SQL ไม่ได้ทำการตั้งชื่อ

4. ยืนยันการปรับปรุงข้อมูลด้วยคำสั่ง **COMMIT** และออกจากระบบ (Logout)

```
COMMIT;
```

COMMIT succeeded.

5. เข้าสู่ระบบ (Login) แล้วใช้คำสั่งตรวจสอบวันที่ปัจจุบันของระบบ

```
SELECT sysdate FROM dual;
```

SYSDATE

05-MAR-12

1 rows selected

6. ปรับปรุงข้อมูลพนักงาน 45 ให้เป็นตำแหน่ง **Manager** (ถ้าไม่กำหนดช่วง **Valid Time** จะเป็นการกำหนดให้ปรับปรุงข้อมูล ณ วันปัจจุบันของระบบ)

```
UPDATE positions SET job_position='Manager' WHERE emp_id=45;
```

0 rows updated

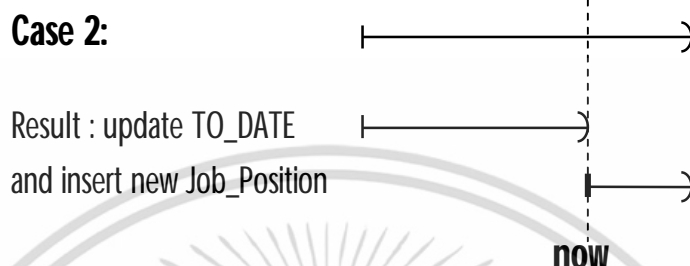
7. สรุปผลการทดลอง จะเห็นว่า ไม่สามารถปรับปรุงข้อมูล (0 rows updated)

เนื่องจากข้อมูลได้ถูกปรับปรุงไปเป็นข้อมูลอดีตแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 Current Update Case 2, in the General Scenario

จากรูปที่ 3.2 เมื่อปรับปรุงข้อมูล ณ เวลาปัจจุบัน จะได้ผลลัพธ์ 1 Insert 1 Update โดยระบบจะทำการเพิ่มข้อมูลและปรับปรุงข้อมูลจากวันปัจจุบัน (Current Date) จนถึงวันที่คาดว่าจะสิ้นสุดของข้อมูล (To_date) จากนั้นจะทำการปรับปรุง To_date ของ PA ให้เป็นวันที่ปัจจุบัน (Current Date)



รูปที่ 3.2 Current Update Case 2, in the General Scenario

1. เพิ่มข้อมูลพนักงานรหัส 46 ตำแหน่ง System Analyst ลงในตาราง Positions ตั้งแต่วันที่ 2005-02-16 ถึงวันที่ 9999-12-31 ด้วยพารามิเตอร์ DBMS_WM.MAX_TIME

```
INSERT INTO positions VALUES(
  46, 'System Analyst',
  WMSYS.WM_PERIOD(TO_DATE('2005-02-16','YYYY-MM-DD'),
    DBMS_WM.MAX_TIME));
```

1 rows inserted

2. ใช้คำสั่ง SELECT เพื่อดูผลการเพิ่มข้อมูลพนักงานรหัส 46

```
SELECT emp_id, job_position,
  TO_CHAR(e.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
  TO_CHAR(e.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM positions e
WHERE emp_id=46;
```

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
46	System Analyst	2005-02-16	9999-12-31

1 rows selected

3. ยืนยันการปรับปรุงข้อมูลด้วยคำสั่ง COMMIT และ Logout ออกจากระบบ

```
COMMIT;
COMMIT succeeded.
```

4. เข้าสู่ระบบ แล้วใช้คำสั่งตรวจสอบวันที่ปัจจุบันของระบบ

```
SELECT sysdate FROM dual;
SYSDATE
-----
05-MAR-12
```

2 rows selected

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ปรับปรุงข้อมูลตำแหน่งพนักงานรหัส 46 ให้เป็นตำแหน่ง Manager

```
UPDATE positions SET job_position='Manager' WHERE emp_id=46;
```

```
1 rows updated
```

6. กำหนดค่าช่วงเวลา Valid Time ตั้งแต่วันที่ DBMS_WS.MIN_TIME ถึงวันที่ DBMS_WM.MAX_TIME แล้วใช้คำสั่ง SELECT เพื่อดูผลการปรับปรุงข้อมูลพนักงานรหัส 46

```
EXECUTE
```

```
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);
```

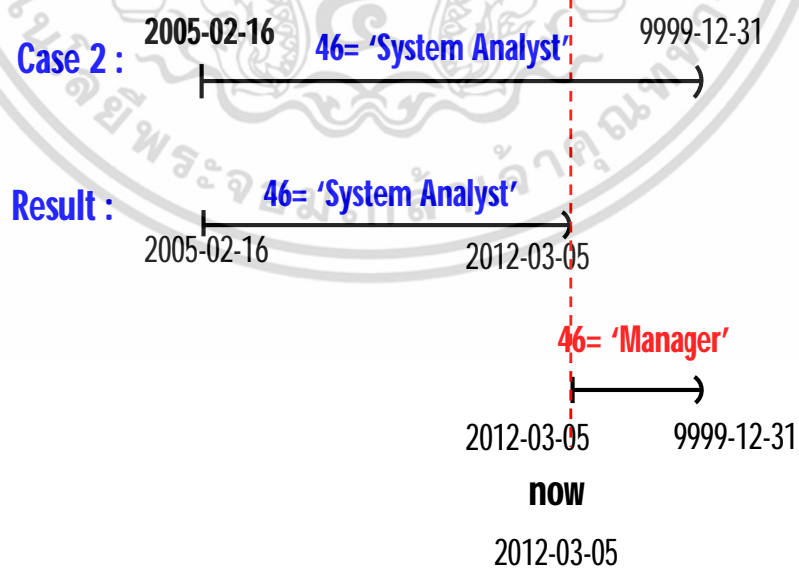
```
SELECT emp_id, job_position,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM positions e
WHERE emp_id=46;
```

```
anonymous block completed
```

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
46	Manager	2012-03-05	9999-12-31
46	System Analyst	2005-02-16	2012-03-05

```
2 rows selected
```

7. สรุปผลการทดลอง ในส่วน Current Update Case 2, in the General Scenario จะได้ผลลัพธ์ 1 Insert 1 Update เป็นไปตามทฤษฎีของ Richard T. Snodgrass ที่ได้เสนอไว้ใน TR-28 [3]

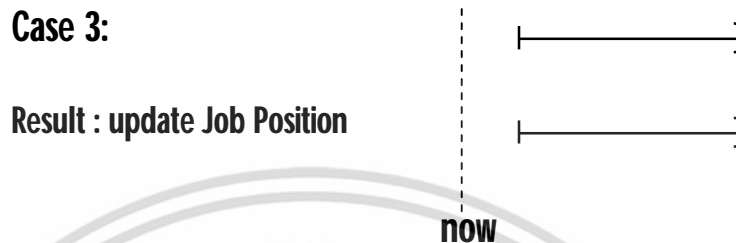


รูปที่ 3.3 ผลลัพธ์ Current Update Case 2, in the General Scenario

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 Current Update Case 3, in the General Scenario

จากรูปที่ 3.4 เมื่อมีการปรับปรุงข้อมูลอนาคต ณ ช่วงเวลาปัจจุบัน ผลของการปรับปรุงข้อมูล จะทำการปรับปรุงเฉพาะค่าของข้อมูลที่ตามการ Update แต่ช่วงเวลายังเป็นเวลาตามเดิม เรียกว่า 1 Update



รูปที่ 3.4 Current Update Case 3, in the General Scenario

1. เพิ่มประวัติข้อมูลพนักงานรหัส 47 ตำแหน่ง System Analyst ลงในตาราง Positions ตั้งแต่วันที่ 2012-06-01 ถึงวันที่ DBMS_WM.MAX_TIME

```
INSERT INTO positions VALUES (
  47, 'System Analyst',
  WMSYS.WM_PERIOD(TO_DATE('2012-06-01','YYYY-MM-DD'),
  DBMS_WM.MAX_TIME));
1 rows inserted
```

2. ใช้คำสั่ง SELECT เพื่อดูผลการเพิ่มข้อมูลพนักงานรหัส 46

```
SELECT emp_id, job_position,
  TO_CHAR(e.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
  TO_CHAR(e.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM positions e
WHERE emp_id=47;
```

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
47	System Analyst	2012-06-01	9999-12-31

1 rows selected

3. ยืนยันการปรับปรุงข้อมูลด้วยคำสั่ง COMMIT และ Logout ออกจากระบบ

```
COMMIT;
COMMIT succeeded.
```

4. เข้าสู่ระบบ แล้วใช้คำสั่งตรวจสอบวันที่ปัจจุบันของระบบ

```
SELECT sysdate FROM dual;
SYSDATE
-----
05-MAR-12
```

1 rows selected

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของบริษัทฯ เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ปรับปรุงข้อมูลตำแหน่งพนักงานรหัส 46 ให้เป็นตำแหน่ง Manager

```
UPDATE positions SET job_position='Manager' WHERE emp_id=47;
```

1 rows updated

6. กำหนดค่าช่วงเวลา Valid Time ตั้งแต่วันที่ DBMS_WS.MIN_TIME ถึงวันที่ DBMS_WM.MAX_TIME แล้วใช้คำสั่ง SELECT เพื่อดูผลการปรับปรุงข้อมูลพนักงานรหัส 47

```
EXECUTE
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

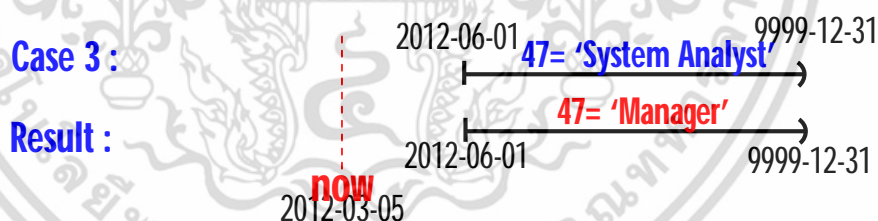
SELECT emp_id, job_position,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM positions e
WHERE emp_id=47;
```

anonymous block completed

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
47	Manager	2012-06-01	9999-12-31

1 rows selected

7. สรุปผลการทดลอง จากการทดลองในส่วน Current Update Case 3, in the General Scenario เป็นไปตามทฤษฎีของ Richard T. Snodgrass ที่ได้เสนอไว้ใน TR-28 [3] คือ จะได้ผลลัพธ์ 1 Update ช่วงเวลาไม่มีการเปลี่ยนแปลง แต่ตำแหน่งงานเปลี่ยนไปจากการปรับปรุง



รูปที่ 3.5 ผลลัพธ์ Current Update Case 3, in the General Scenario

3.3 การทดลองลบข้อมูลแบบซีควนซ์ (Sequence Deletion)

3.3.1 การทดลองลบข้อมูลแบบซีควนซ์ กรณีที่ 1 (Sequence Deletion Case 1)

ในการทดลองนี้ จะเพิ่มข้อมูลพนักงานรหัส 77 ตำแหน่ง='Manager' ตั้งแต่วันที่ 2005-05-01 ถึงวันที่ 31-12-9999 แล้วทำการลบข้อมูลพนักงานรหัส 77 ออกจากระบบ ตั้งแต่วันที่ 2007-02-01 ถึงวันที่ 2007-05-01 ลักษณะของ Sequenced Deletion Case 1 โดย PA เกิดขึ้นที่หลังและสิ้นสุดก่อน PV (PA เป็นส่วนหนึ่งของ PV)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. กำหนดค่าช่วงเวลา Valid Time ตั้งแต่วันที่ DBMS_WS.MIN_TIME ถึงวันที่ DBMS_WM.MAX_TIME แล้วใช้คำสั่ง SELECT เพื่อดูผลการลบข้อมูลพนักงานรหัส 77

```
EXECUTE
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT emp_id, job_position,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM positions e
WHERE emp_id=77;
anonymous block completed
```

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
77	Manager	2005-05-01	2007-01-01
77	Manager	2008-01-01	9999-12-31

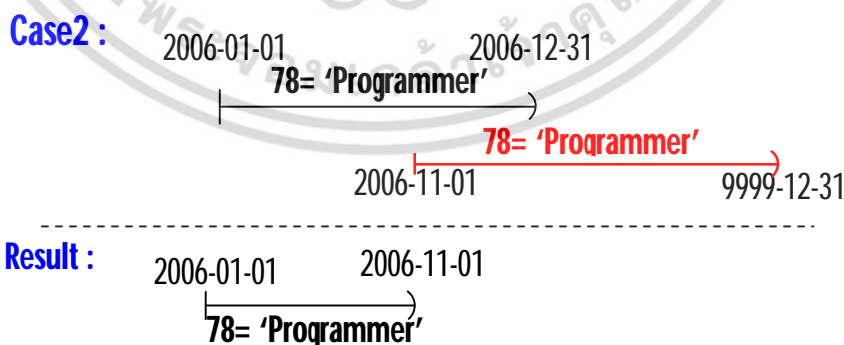
2 rows selected

5. สรุปผลการทดลอง Sequenced Deletion Case 1

จากการทดลองในส่วน Sequenced Deletion Case 1 ผลของ PA ที่เป็นส่วนหนึ่งของ PV จะได้ผลลัพธ์ 1 Update 1 Insert เป็นไปตามทฤษฎีของ Richard T. Snodgrass ที่ได้เสนอไว้ใน TR-28 [3]

3.3.2 การทดลองลบข้อมูลแบบซีควเินซ์ กรณีที่ 2 (Sequence Deletion Case 2)

ในการทดลองนี้ ทำการเพิ่มประวัติข้อมูลพนักงานรหัส 78 ตำแหน่ง='Programmer' ตั้งแต่วันที่ 2006-01-01 ถึงวันที่ 2006-12-31 แล้วทำการลบข้อมูลประวัติพนักงานรหัส 78 ตำแหน่ง 'Programmer' ออกจากระบบ ตั้งแต่วันที่ 2006-11-01 ถึงวันที่ 9999-12-31 ซึ่งเป็นการลบข้อมูลลักษณะของ Sequenced Deletion Case 2 คือ PA เกิดขึ้นและสิ้นสุดที่หลัง PV ดังรูปที่ 3.8



รูปที่ 3.7 Sequenced Deletion Case 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เพิ่มข้อมูลพนักงานรหัส 78 ตำแหน่ง 'Programmer' ลงในตาราง Positions ตั้งแต่วันที่ 2006-01-01 ถึงวันที่ 2006-12-31

```
INSERT INTO positions VALUES(
    78,'Programmer',
    WMSYS.WM_PERIOD(TO_DATE('2006-01-01','YYYY-MM-DD'),
                    TO_DATE('2006-12-31','YYYY-MM-DD')));
```

1 rows inserted

2. กำหนดค่าช่วงเวลา Valid Time ตั้งแต่วันที่ DBMS_WS.MIN_TIME ถึงวันที่ DBMS_WM.MAX_TIME จากนั้นใช้คำสั่ง SELECT เพื่อดูการเพิ่มข้อมูลพนักงานรหัส 78

```
EXECUTE
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT emp_id, job_position,
       TO_CHAR(e.WM_VALID.VALIDFROM,'YYYY-MM-DD') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL,'YYYY-MM-DD') TO_DATE
FROM positions e
WHERE emp_id=78;
```

anonymous block completed

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
78	Programmer	2006-01-01	2006-12-31

1 rows selected

3. กำหนด Valid Time ตั้งแต่วันที่ 2006-11-01 ถึงวันที่ DBMS_WM.MAX_TIME แล้วทำการลบข้อมูลพนักงานรหัส 78 ออกจากระบบ

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('2006-11-01','YYYY-MM-DD'),
                              DBMS_WM.MAX_TIME);
```

```
DELETE FROM positions WHERE emp_id=78;
```

anonymous block completed

1 rows deleted

4. กำหนดค่าช่วงเวลา Valid Time ตั้งแต่วันที่ DBMS_WS.MIN_TIME ถึงวันที่ DBMS_WM.MAX_TIME แล้วใช้คำสั่ง SELECT เพื่อดูผลการลบข้อมูลพนักงานรหัส 78

```
EXECUTE
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT emp_id, job_position,
       TO_CHAR(e.WM_VALID.VALIDFROM,'YYYY-MM-DD') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL,'YYYY-MM-DD') TO_DATE
FROM positions e
WHERE emp_id=78;
```

anonymous block completed

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
78	Programmer	2006-01-01	2006-11-01

1 rows selected

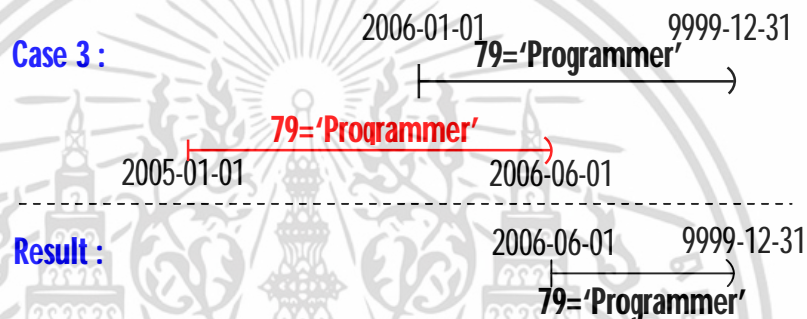
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. สรุปผลการทดลอง Sequenced Deletion Case 2

จากการทดลองในส่วน Sequenced Deletion Case 2 จะได้ผลลัพธ์ 1 Update เป็นไปตามทฤษฎีของ Richard T. Snodgrass ที่ได้เสนอไว้ใน TR-28 [3]

3.3.3 การทดลองลบข้อมูลแบบซีควเอนซ์ กรณีที่ 3 (Sequence Deletion Case 3)

ในการทดลองนี้ ทำการเพิ่มประวัติข้อมูลพนักงานรหัส 79 ตำแหน่ง='Programmer' ตั้งแต่วันที่ 2006-01-01 ถึงวันที่ 9999-12-31 แล้วทำการลบข้อมูลประวัติพนักงานรหัส 79 ตำแหน่ง 'Programmer' ออกจากระบบ ตั้งแต่วันที่ 2005-01-01 ถึงวันที่ 2006-06-01 ซึ่งเป็นการลบข้อมูลลักษณะของ Sequenced Deletion Case 3 คือ PA เกิดขึ้นและสิ้นสุดก่อน PV ดังรูปที่ 3.9



รูปที่ 3.8 Sequenced Deletion Case 3

1. เพิ่มข้อมูลพนักงานรหัส 79 ตำแหน่ง 'Programmer' ลงในตาราง Positions ตั้งแต่วันที่ 2006-01-01 ถึงวันที่ DBMS_WM.MAX_TIME

```
INSERT INTO positions VALUES (
  79, 'Programmer',
  WMSYS.WM_PERIOD(TO_DATE('2006-01-01', 'YYYY-MM-DD'),
    DBMS_WM.MAX_TIME));
```

1 rows inserted

2. กำหนดค่าช่วงเวลา Valid Time ตั้งแต่วันที่ DBMS_WS.MIN_TIME ถึงวันที่ DBMS_WM.MAX_TIME จากนั้นใช้คำสั่ง SELECT เพื่อดูการเพิ่มข้อมูลพนักงานรหัส 79

```
EXECUTE
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT emp_id, job_position,
  TO_CHAR(e.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
  TO_CHAR(e.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM positions e
WHERE emp_id=79;
```

anonymous block completed

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
79	Programmer	2006-01-01	9999-12-31

1 rows selected

เอกสารนี้เป็นเอกสารสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. กำหนดค่า **Valid Time** ตั้งแต่วันที่ 2005-01-01 ถึงวันที่ 2006-06-01 แล้วทำการลบข้อมูลพนักงานรหัส 79 ออกจากระบบ

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('2005-01-01','YYYY-MM-DD'),TO_DATE('2006-06-01','YYYY-MM-DD'));
```

```
DELETE FROM positions WHERE emp_id=79;
```

anonymous block completed
1 rows deleted

4. กำหนดค่าช่วงเวลา **Valid Time** ตั้งแต่วันที่ DBMS_WS.MIN_TIME ถึงวันที่ DBMS_WM.MAX_TIME แล้วใช้คำสั่ง **SELECT** เพื่อดูผลการลบข้อมูลพนักงานรหัส 79

```
EXECUTE  
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);
```

```
SELECT emp_id, job_position,  
        TO_CHAR(e.WM_VALID.VALIDFROM,'YYYY-MM-DD') FROM_DATE,  
        TO_CHAR(e.WM_VALID.VALIDTILL,'YYYY-MM-DD') TO_DATE  
FROM positions e  
WHERE emp_id=79;
```

anonymous block completed

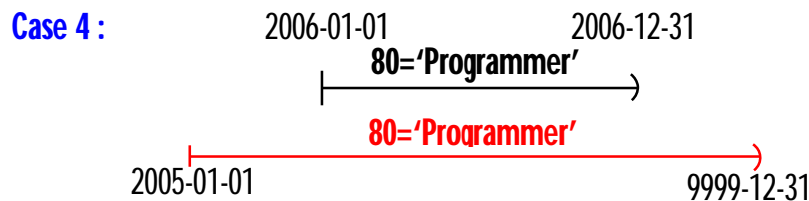
EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
79	Programmer	2006-06-01	9999-12-31

1 rows selected

5. สรุปผลการทดลอง **Sequenced Deletion Case 3**
จากการทดลองในส่วน **Sequenced Deletion Case 3** จะได้ผลลัพธ์ 1 Update เป็นไปตามทฤษฎีของ **Richard T. Snodgrass** ที่ได้เสนอไว้ใน TR-28 [3]

3.3.4 การทดลองลบข้อมูลแบบซีควเอนซ์ กรณีที่ 4 (Sequence Deletion Case 4)

ในการทดลองนี้ ทำการเพิ่มประวัติข้อมูลพนักงานรหัส 80 ตำแหน่ง 'Programmer' ตั้งแต่วันที่ 2006-01-01 ถึงวันที่ 2006-12-31 แล้วทำการลบข้อมูลประวัติพนักงานรหัส 80 ตำแหน่ง 'Programmer' ออกจากระบบ ตั้งแต่วันที่ 2005-01-01 ถึงวันที่ 9999-12-31 ซึ่งเป็นการลบข้อมูลลักษณะของ **Sequenced Deletion Case 4** คือ PA เกิดขึ้นก่อนและสิ้นสุดที่หลัง PV ดังรูปที่ 3.9



Result : entire row deleted

รูปที่ 3.9 Sequenced Deletion Case 4

1. เพิ่มข้อมูลพนักงานรหัส 80 ตำแหน่ง 'Programmer' ลงในตาราง Positions ตั้งแต่วันที่ 2006-01-01 ถึงวันที่ 2006-12-31

```
INSERT INTO positions VALUES (
    80, 'Programmer',
    WMSYS.WM_PERIOD(TO_DATE('2006-01-01', 'YYYY-MM-DD'),
    TO_DATE('2006-12-31', 'YYYY-MM-DD'))
);
1 rows inserted
```

2. กำหนดค่าช่วงเวลา Valid Time ตั้งแต่วันที่ DBMS_WM.MIN_TIME ถึงวันที่ DBMS_WM.MAX_TIME จากนั้นใช้คำสั่ง SELECT เพื่อดูการเพิ่มข้อมูลพนักงานรหัส 80

```
EXECUTE
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT emp_id, job_position,
    TO_CHAR(e.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
    TO_CHAR(e.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM positions e
WHERE emp_id=80;
```

anonymous block completed

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
80	Programmer	2006-01-01	2006-12-31

1 rows selected

3. กำหนดค่า Valid Time ตั้งแต่วันที่ 2005-01-01 ถึงวันที่ DBMS_WM.MAX_TIME แล้วทำการลบข้อมูลพนักงานรหัส 80 ออกจากระบบ

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('2005-01-01', 'YYYY-MM-DD'),
DBMS_WM.MAX_TIME);
```

```
DELETE FROM positions WHERE emp_id=80;
```

anonymous block completed

1 rows deleted

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เพิ่มข้อมูลพนักงานรหัส 45 ตำแหน่ง 'System Analyst' ลงในตาราง Positions ตั้งแต่วันที่ 2005-02-16 ถึงวันที่ DBMS_WM.MAX_TIME

```
INSERT INTO positions VALUES(
  45,'System Analyst',
  WMSYS.WM_PERIOD(TO_DATE('2005-05-16','YYYY-MM-DD'),
    DBMS_WM.MAX_TIME));
```

1 rows inserted

2. กำหนดค่าช่วงเวลา Valid Time ตั้งแต่วันที่ DBMS_WS.MIN_TIME ถึงวันที่ DBMS_WM.MAX_TIME จากนั้นใช้คำสั่ง SELECT เพื่อดูการเพิ่มข้อมูลพนักงานรหัส 45

```
EXECUTE
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT emp_id, job_position,
  TO_CHAR(e.WM_VALID.VALIDFROM,'YYYY-MM-DD') FROM_DATE,
  TO_CHAR(e.WM_VALID.VALIDTILL,'YYYY-MM-DD') TO_DATE
FROM positions e
WHERE emp_id=45;
```

anonymous block completed

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
45	System Analyst	2006-01-01	9999-12-31

1 rows selected

3. กำหนดค่า Valid Time ตั้งแต่วันที่ 2008-01-01 ถึงวันที่ 2010-01-01 แล้วทำการปรับปรุงข้อมูลพนักงานรหัส 45 เป็นตำแหน่ง 'Project Manager'

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('2008-01-01','YYYY-MM-DD'),
  TO_DATE('2010-01-01','YYYY-MM-DD'));

UPDATE positions SET job_position='Project Manager'
WHERE emp_id=45;
```

anonymous block completed

1 rows updated

4. กำหนดค่าช่วงเวลา Valid Time ตั้งแต่วันที่ DBMS_WS.MIN_TIME ถึงวันที่ DBMS_WM.MAX_TIME จากนั้นใช้คำสั่ง SELECT เพื่อดูการปรับปรุงข้อมูลพนักงานรหัส 45

```
EXECUTE
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT emp_id, job_position,
  TO_CHAR(e.WM_VALID.VALIDFROM,'YYYY-MM-DD') FROM_DATE,
  TO_CHAR(e.WM_VALID.VALIDTILL,'YYYY-MM-DD') TO_DATE
FROM positions e
WHERE emp_id=45;
```

anonymous block completed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
45	Project Manager	2008-01-01	2010-01-01
45	System Analyst	2005-05-16	2008-01-01
45	System Analyst	2010-01-01	9999-12-31

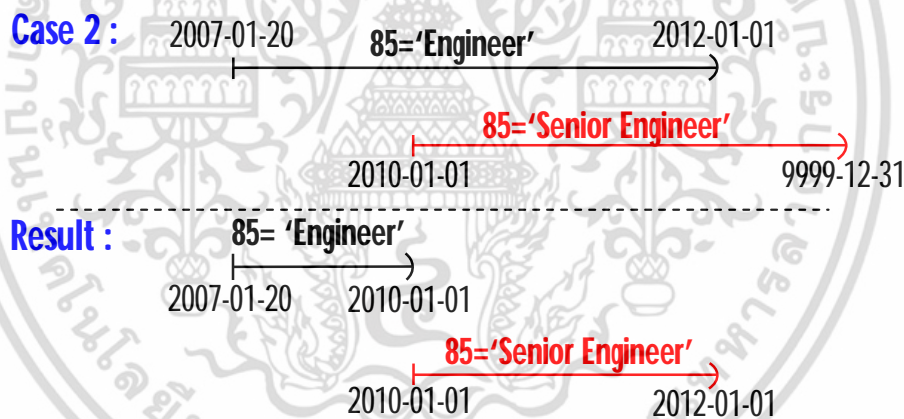
3 rows selected

5. สรุปผลการทดลอง Sequenced Update Case 1

จากการทดลอง Sequenced Update Case 1 จะได้ผลลัพธ์ 2 Insert 1 Update เป็นไปตามทฤษฎีของ Richard T. Snodgrass ที่ได้เสนอไว้ใน TR-28 [3]

3.4.2 การปรับปรุงข้อมูลแบบซีควเอนซ์ กรณีที่ 2 (Sequence Update Case 2)

ในการทดลองนี้ ทำการเพิ่มประวัติข้อมูลพนักงานรหัส 85 ตำแหน่ง 'Engineer' ตั้งแต่วันที่ 2007-01-03 ถึงวันที่ 2012-01-01 แล้วทำการปรับปรุงข้อมูลประวัติพนักงานรหัส 85 เป็นตำแหน่ง 'Senior Engineer' ตั้งแต่วันที่ 2010-01-01 ถึงวันที่ 9999-12-31 ซึ่งเป็นการปรับปรุงข้อมูลลักษณะของ Sequenced Update Case 2 คือ PA เกิดขึ้นและสิ้นสุดที่หลัง PV ดังรูปที่ 3.11



รูปที่ 3.11 Sequenced Update Case 2

1. เพิ่มข้อมูลพนักงานรหัส 85 ตำแหน่ง 'Engineer' ลงในตาราง Positions ตั้งแต่วันที่ 2005-02-16 ถึงวันที่ 2012-01-01

```
INSERT INTO positions VALUES(
  85, 'Engineer',
  WMSYS.WM_PERIOD(TO_DATE('2007-01-20', 'YYYY-MM-DD'),
    TO_DATE('2012-01-01', 'YYYY-MM-DD'))
);
```

1 rows inserted

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กำหนดค่าช่วงเวลา **Valid Time** ตั้งแต่วันที่ **DBMS_WS.MIN_TIME** ถึงวันที่ **DBMS_WM.MAX_TIME** จากนั้นใช้คำสั่ง **SELECT** เพื่อดูการเพิ่มข้อมูลพนักงานรหัส 85

```
EXECUTE
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT emp_id, job_position,
       TO_CHAR(e.WM_VALID.VALIDFROM,'YYYY-MM-DD') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL,'YYYY-MM-DD') TO_DATE
FROM positions e
WHERE emp_id=85;
```

anonymous block completed

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
85	Engineer	2007-01-20	9999-12-31

1 rows selected

3. กำหนดค่าช่วงเวลา **Valid Time** ตั้งแต่วันที่ **2010-01-01** ถึงวันที่ **DBMS_WM.MAX_TIME** แล้วทำการปรับปรุงข้อมูลพนักงานรหัส 85 เป็นตำแหน่ง 'Project Manager'

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('2010-01-01','YYYY-MM-DD'),
DBMS_WM.MAX_TIME);

UPDATE positions SET job_position='Senior Engineer'
WHERE emp_id=85;
```

anonymous block completed
1 rows updated

4. กำหนดค่าช่วงเวลา **Valid Time** ตั้งแต่วันที่ **DBMS_WS.MIN_TIME** ถึงวันที่ **DBMS_WM.MAX_TIME** จากนั้นใช้คำสั่ง **SELECT** เพื่อดูการปรับปรุงข้อมูลพนักงานรหัส 85

```
EXECUTE
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT emp_id, job_position,
       TO_CHAR(e.WM_VALID.VALIDFROM,'YYYY-MM-DD') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL,'YYYY-MM-DD') TO_DATE
FROM positions e
WHERE emp_id=85;
```

anonymous block completed

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
85	Senior Engineer	2010-01-01	2012-01-01
85	Engineer	2007-01-20	2010-01-01

2 rows selected

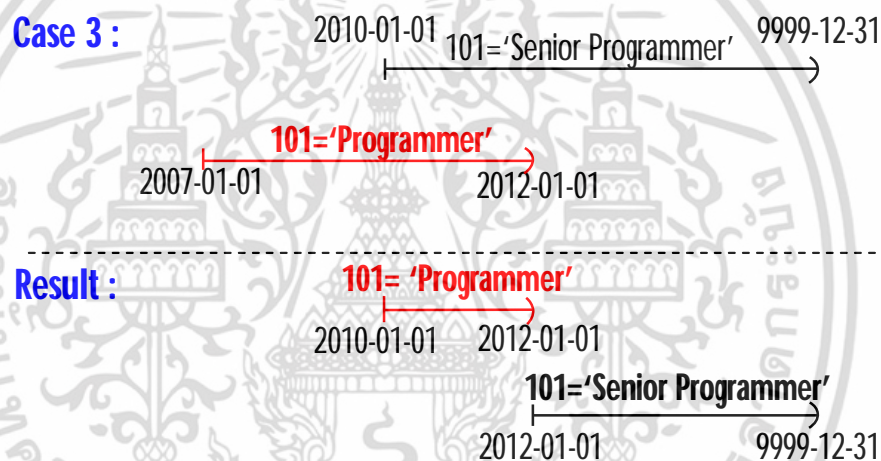
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. สรุปผลการทดลอง Sequenced Update Case 2

จากการทดลอง Sequenced Update Case 2 จะได้ผลลัพธ์ 1 Insert 1 Update เป็นไปตามทฤษฎีของ Richard T. Snodgrass ที่ได้เสนอไว้ใน TR-28 [3]

3.4.3 การปรับปรุงข้อมูลแบบซีควเอนซ์ กรณีที่ 3 (Sequence Update Case 3)

ในการทดลองนี้ ทำการเพิ่มประวัติข้อมูลพนักงานรหัส 101 ตำแหน่ง 'Senior Programmer' ตั้งแต่วันที่ 2010-01-01 ถึงวันที่ 9999-12-31 แล้วทำการปรับปรุงข้อมูลประวัติพนักงานรหัส 101 ย้อนหลังเป็นตำแหน่ง 'Programmer' ตั้งแต่วันที่ 2007-01-01 ถึง 2012-01-01 ซึ่งเป็นการปรับปรุงข้อมูลลักษณะของ Sequenced Update Case 3 คือ PA เกิดขึ้นและสิ้นสุดก่อน PV ดังรูปที่ 3.12



รูปที่ 3.12 Sequenced Update Case 3

1. เพิ่มข้อมูลพนักงานรหัส 101 ตำแหน่ง 'Senior Programmer' ลงในตาราง Positions ตั้งแต่วันที่ 2010-01-01 ถึงวันที่ DBMS_WM.MAX_TIME

```
INSERT INTO positions VALUES(
  101,'Senior Programmer',
  WMSYS.WM_PERIOD(TO_DATE('2010-01-01','YYYY-MM-DD'),
    DBMS_WM.MAX_TIME )
);
```

1 rows inserted

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กำหนดค่าช่วงเวลา **Valid Time** ตั้งแต่วันที่ **DBMS_WS.MIN_TIME** ถึงวันที่ **DBMS_WM.MAX_TIME** จากนั้นใช้คำสั่ง **SELECT** เพื่อดูการเพิ่มข้อมูลพนักงานรหัส 101

```
EXECUTE
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT emp_id, job_position,
       TO_CHAR(e.WM_VALID.VALIDFROM,'YYYY-MM-DD') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL,'YYYY-MM-DD') TO_DATE
FROM positions e
WHERE emp_id=101;
```

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
101	Senior Programmer	2010-01-01	9999-12-31

1 rows selected

3. กำหนดค่า **Valid Time** ตั้งแต่วันที่ **2010-01-01** ถึงวันที่ **DBMS_WM.MAX_TIME** แล้วทำการปรับปรุงข้อมูลพนักงานรหัส 101 เป็นตำแหน่ง 'Project Manager'

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('2010-01-01','YYYY-MM-DD'),
TO_DATE('2012-01-01','YYYY-MM-DD'));

UPDATE positions SET job_position='Programmer'
WHERE emp_id=101;

1 rows updated
```

4. กำหนดค่าช่วงเวลา **Valid Time** ตั้งแต่วันที่ **DBMS_WS.MIN_TIME** ถึงวันที่ **DBMS_WM.MAX_TIME** จากนั้นใช้คำสั่ง **SELECT** เพื่อดูการปรับปรุงข้อมูลพนักงานรหัส 101

```
EXECUTE
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT emp_id, job_position,
       TO_CHAR(e.WM_VALID.VALIDFROM,'YYYY-MM-DD') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL,'YYYY-MM-DD') TO_DATE
FROM positions e
WHERE emp_id=101;
```

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
101	Programmer	2010-01-01	2012-01-01
101	Senior Programmer	2012-01-01	9999-12-31

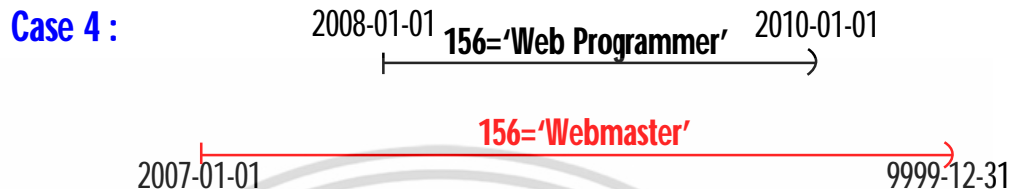
2 rows selected

5. สรุปผลการทดลองการ Sequenced Update Case 3

จากการทดลอง Sequenced Update Case 3 จะได้ผลลัพธ์ 1 Insert 1 Update เป็นไปตามทฤษฎีของ Richard T. Snodgrass ที่ได้เสนอไว้ใน TR-28 [3]

3.4.4 การปรับปรุงข้อมูลแบบซีควเอนซ์ กรณีที่ 4 (Sequence Update Case 4)

ในการทดลองนี้ ทำการเพิ่มข้อมูลพนักงานรหัส 156 ตำแหน่ง 'WEB Programmer' ตั้งแต่วันที่ 2008-01-01 ถึงวันที่ 2010-01-01 แล้วทำการปรับปรุงข้อมูลประวัติพนักงานรหัส 156 เป็นตำแหน่ง 'Webmaster' ตั้งแต่วันที่ 2007-01-01 ถึงวันที่ 9999-12-31 ซึ่งเป็นการปรับปรุงข้อมูลลักษณะของ Sequenced Update Case 4 คือ PA เกิดขึ้นก่อนและสิ้นสุดที่หลัง PV ดังรูปที่ 3.13



Result : entire row updated



รูปที่ 3.13 Sequenced Update Case 4

1. เพิ่มข้อมูลพนักงานรหัส 156 ตำแหน่ง 'WEB Programmer' ลงในตาราง Positions ตั้งแต่วันที่ 2008-01-01 ถึงวันที่ 2010-01-01

```
INSERT INTO positions VALUES (
  156, 'WEB Programmer',
  WMSYS.WM_PERIOD(TO_DATE('2008-01-01', 'YYYY-MM-DD'),
    TO_DATE('2010-01-01', 'YYYY-MM-DD')));
1 rows inserted
```

2. กำหนดค่าช่วงเวลา Valid Time ตั้งแต่วันที่ DBMS_WM.MIN_TIME ถึงวันที่ DBMS_WM.MAX_TIME จากนั้นใช้คำสั่ง SELECT เพื่อดูการเพิ่มข้อมูลพนักงานรหัส 156

```
EXECUTE
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT emp_id, job_position,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM positions e
WHERE emp_id=156;
anonymous block completed
```

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
156	WEB Programmer	2008-01-01	2010-01-01

1 rows selected

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. กำหนดค่า Valid Time ตั้งแต่ 2007-01-01 ถึงวันที่ DBMS_WM.MAX_TIME แล้วทำการปรับปรุงข้อมูลพนักงานรหัส 156 เป็นตำแหน่ง 'Webmaster'

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('2007-01-01','YYYY-MM-DD'), DBMS_WM.MAX_TIME);
```

```
UPDATE positions SET job_position='Webmaster'  
WHERE emp_id=156;
```

anonymous block completed
1 rows updated

4. กำหนดค่าช่วงเวลา Valid Time ตั้งแต่วันที่ DBMS_WM.MIN_TIME ถึงวันที่ DBMS_WM.MAX_TIME จากนั้นใช้คำสั่ง SELECT เพื่อดูการปรับปรุงข้อมูลพนักงานรหัส 156

```
EXECUTE  
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);
```

```
SELECT emp_id, job_position,  
        TO_CHAR(e.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,  
        TO_CHAR(e.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE  
FROM positions e  
WHERE emp_id=156;
```

anonymous block completed

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
156	Webmaster	2008-01-01	2010-01-01

1 rows selected

5. สรุปผลการทดลอง Sequenced Update Case 4

จากการทดลอง Sequenced Update Case 4 จะได้ผลลัพธ์ 1 Update เป็นไปตามทฤษฎีของ Richard T. Snodgrass ที่ได้เสนอไว้ใน TR-28 [3]

3.6 การทดลอง Temporal Projection and Selection

1. กำหนดวันปัจจุบันคือ 2012-05-03 แล้วทำการเพิ่มข้อมูลดังต่อไปนี้

```
INSERT INTO positions VALUES(
  159,'Programmer',
  WMSYS.WM_PERIOD(TO_DATE('2011-08-01','YYYY-MM-DD'),
    TO_DATE('2012-04-30','YYYY-MM-DD')));
INSERT INTO positions VALUES(
  160,'Programmer',
  WMSYS.WM_PERIOD(TO_DATE('2011-08-01','YYYY-MM-DD'),
    TO_DATE('2012-04-30','YYYY-MM-DD')));
INSERT INTO positions VALUES(
  200,'Operator',
  WMSYS.WM_PERIOD(TO_DATE('2012-03-01','YYYY-MM-DD'),
    TO_DATE('2012-07-31','YYYY-MM-DD')));
INSERT INTO positions VALUES(
  201,'Advisor',
  WMSYS.WM_PERIOD(TO_DATE('2012-01-01','YYYY-MM-
DD'),DBMS_WM.UNTIL_CHANGED));

COMMIT;
```

2. ทดลองเขียนคำสั่ง เพื่อแสดงข้อมูลทั้งหมด ตั้งแต่ อดีตจนถึงปัจจุบัน ของตาราง

POSITIONS

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);
```

```
SELECT emp_id, job_position,
  TO_CHAR(e.WM_VALID.VALIDFROM,'YYYY-MM-DD') FROM_DATE,
  TO_CHAR(e.WM_VALID.VALIDTILL,'YYYY-MM-DD') TO_DATE
FROM positions e
```

anonymous block completed

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
45	System Analyst	2005-02-16	2008-01-01
46	Manager	2012-03-05	9999-12-31
47	Manager	2012-06-01	9999-12-31
84	Project Manager	2008-01-01	2010-01-01
85	Senior Engineer	2010-01-01	2012-01-01
101	Programmer	2010-01-01	2012-01-01
156	Webmaster	2008-01-01	2010-01-01
159	Programmer	2011-08-01	2012-04-30
160	Programmer	2011-08-01	2012-04-30
200	Operator	2012-03-01	2012-07-31
201	Advisor	2012-01-01	
46	System Analyst	2005-02-16	2012-03-05
77	Manager	2005-05-01	2007-01-01
77	Manager	2008-01-01	9999-12-31
78	Programmer	2006-01-01	2006-11-01
79	Programmer	2006-06-01	9999-12-31
84	System Analyst	2005-02-16	2008-01-01
84	System Analyst	2010-01-01	9999-12-31
85	Engineer	2007-01-20	2010-01-01
101	Senior Programmer	2012-01-01	9999-12-31

20 rows selected

หมายเหตุ รหัสพนักงาน 201 มีค่าว่างที่คอลัมน์ TO_DATE หมายถึงข้อมูลยังเป็นปัจจุบันอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ทดลองเขียนคำสั่ง เพื่อแสดงข้อมูลว่า “มีพนักงานคนไหนบ้างที่มีตำแหน่งเป็น Manager”

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);
```

```
SELECT emp_id, job_position,
        TO_CHAR(e.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
        TO_CHAR(e.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM positions e
WHERE job_position = 'Manager';
```

anonymous block completed

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
46	Manager	2005-02-16	9999-12-31
47	Manager	2012-06-01	9999-12-31
77	Manager	2005-05-01	2007-01-01
77	Manager	2008-01-01	9999-12-31

4 rows selected

4. ทดลองเขียนคำสั่ง เพื่อแสดงข้อมูลว่า “มีพนักงานคนไหนบ้างที่มีตำแหน่งเป็น Programmer”

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);
```

```
SELECT emp_id, job_position,
        TO_CHAR(e.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
        TO_CHAR(e.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM positions e
WHERE job_position = 'Programmer';
```

anonymous block completed

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
101	Programmer	2010-01-01	2012-01-01
159	Programmer	2011-08-01	2012-04-30
160	Programmer	2011-08-01	2012-04-30
78	Programmer	2006-01-01	2006-11-01
79	Programmer	2006-06-01	9999-12-31

5 rows selected

5. ทดลองเขียนคำสั่ง เพื่อแสดงข้อมูลว่า “พนักงานคนไหนทำงานตำแหน่ง Programmer ในช่วงวันที่ 2006-01-01 ถึง 2006-12-31”

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('2006-01-01', 'YYYY-MM-DD'),
TO_DATE('2006-12-31', 'YYYY-MM-DD'));
```

```
SELECT emp_id, job_position,
        TO_CHAR(e.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
        TO_CHAR(e.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM positions e
WHERE job_position = 'Programmer';
```

anonymous block completed

เอกสารนี้เป็นเอกสารเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
78	Programmer	2006-01-01	2006-11-01
79	Programmer	2006-06-01	9999-12-31

2 rows selected

6. ทดลองเขียนคำสั่ง เพื่อแสดงข้อมูลว่า “มีพนักงานคนไหนบ้าง ที่ปัจจุบันยังทำงานอยู่”

```
EXECUTE DBMS_WM.SetValidTime( );
```

```
SELECT emp_id, job_position,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM positions e;
```

anonymous block completed

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
46	Manager	2012-03-05	9999-12-31
47	Manager	2012-06-01	9999-12-31
159	Programmer	2011-08-01	2012-04-30
160	Programmer	2011-08-01	2012-04-30
200	Operator	2012-03-01	2012-07-31
201	Advisor	2012-01-01	
46	System Analyst	2005-02-16	2012-03-05
77	Manager	2008-01-01	9999-12-31
79	Programmer	2006-06-01	9999-12-31
84	System Analyst	2010-01-01	9999-12-31
101	Senior Programmer	2012-01-01	9999-12-31

11 rows selected

หมายเหตุ จะเห็นว่า พนักงานรหัส 159 160 และ 200 ได้กำหนดวันสิ้นสุดการทำงานของตำแหน่งแล้ว แต่ ณ วันปัจจุบัน (2012-03-05) เขายังทำงานอยู่

7. ทดลองกำหนดช่วง Valid Time เพื่อแสดงข้อมูล พนักงานคนไหนบ้างที่ทำงานในช่วงวันที่ 2011-08-01 ถึงวันที่ 2012-04-30

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('2011-08-01','YYYY-MM-DD'),
TO_DATE('2012-04-30','YYYY-MM-DD'));
```

```
SELECT emp_id, job_position,
        TO_CHAR(e.WM_VALID.VALIDFROM,'YYYY-MM-DD') FROM_DATE,
        TO_CHAR(e.WM_VALID.VALIDTILL,'YYYY-MM-DD') TO_DATE
FROM positions e;
```

anonymous block completed

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
46	Manager	2012-03-05	9999-12-31
85	Senior Engineer	2010-01-01	2012-01-01
101	Programmer	2010-01-01	2012-01-01
159	Programmer	2011-08-01	2012-04-30
160	Programmer	2011-08-01	2012-04-30
200	Operator	2012-03-01	2012-07-31
201	Advisor	2012-01-01	
46	System Analyst	2005-02-16	2012-03-05
77	Manager	2008-01-01	9999-12-31
79	Programmer	2006-06-01	9999-12-31
84	System Analyst	2010-01-01	9999-12-31
101	Senior Programmer	2012-01-01	9999-12-31

12 rows selected

8. ทดลอง การแสดงข้อมูลโดยกำหนดช่วงเวลา “มีพนักงานคนไหนบ้างที่ทำงานเฉพาะวันที่ 2011-08-01 ถึงวันที่ 2012-04-30” โดยใช้ Operators for Valid Time Support

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);
```

```
SELECT emp_id, job_position,
        TO_CHAR(e.WM_VALID.VALIDFROM,'YYYY-MM-DD') FROM_DATE,
        TO_CHAR(e.WM_VALID.VALIDTILL,'YYYY-MM-DD') TO_DATE
FROM positions e
WHERE WM_EQUALS(e.WM_VALID,
                 WM_PERIOD(TO_DATE('2011-08-01','YYYY-MM-DD'),
                           TO_DATE('2012-04-30','YYYY-MM-DD')))=1;
```

anonymous block completed

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
159	Programmer	2011-08-01	2012-04-30
160	Programmer	2011-08-01	2012-04-30

2 rows selected

9. ทดลองเขียนคำสั่ง เพื่อแสดงข้อมูลว่า “มีพนักงานตำแหน่งไหนบ้าง ที่สิ้นสุดการทำงานในแต่ละตำแหน่งแล้ว และได้มีการกำหนดล่วงหน้าของการสิ้นสุดตำแหน่งแล้ว”

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);
```

```
SELECT emp_id, job_position,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM   positions e
WHERE  e.WM_VALID.VALIDTILL != DBMS_WM.MAX_TIME
AND    e.WM_VALID.VALIDTILL IS NOT NULL;
```

anonymous block completed

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
45	System Analyst	2005-02-16	2008-01-01
84	Project Manager	2008-01-01	2010-01-01
85	Senior Engineer	2010-01-01	2012-01-01
101	Programmer	2010-01-01	2012-01-01
156	Webmaster	2008-01-01	2010-01-01
159	Programmer	2011-08-01	2012-04-30
160	Programmer	2011-08-01	2012-04-30
200	Operator	2012-03-01	2012-07-31
46	System Analyst	2005-02-16	2012-03-05
77	Manager	2005-05-01	2007-01-01
78	Programmer	2006-01-01	2006-11-01
84	System Analyst	2005-02-16	2008-01-01
85	Engineer	2007-01-20	2010-01-01

13 rows selected

จากตารางนี้ พนักงานที่สิ้นสุดการทำงานในตำแหน่งต่างๆ แต่อาจจะยังทำงานในตำแหน่งอื่นๆ อยู่ก็ได้เช่น พนักงานรหัส 101 สิ้นสุดการทำงานตำแหน่ง Programmer แต่ปัจจุบันพนักงานรหัส 101 ทำงานอยู่ในตำแหน่ง Senior Programmer (ดูจากข้อที่ 2)

10. ทดลองเขียนคำสั่ง เพื่อแสดงว่า “ณ วันปัจจุบัน มีพนักงานตำแหน่งใดบ้างที่จะถูกพ้นตำแหน่งซึ่งได้กำหนดไว้ล่วงหน้าแล้ว”

```
EXECUTE DBMS_WM.SetValidTime( );
```

```
SELECT emp_id, job_position,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM   positions e
WHERE  e.WM_VALID.VALIDTILL != DBMS_WM.MAX_TIME
AND    e.WM_VALID.VALIDTILL IS NOT NULL;
```

anonymous block completed

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
159	Programmer	2011-08-01	2012-04-30
160	Programmer	2011-08-01	2012-04-30
200	Operator	2012-03-01	2012-07-31
46	System Analyst	2005-02-16	2012-03-05

4 rows selected

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. ทดลองเขียนคำสั่ง เพื่อแสดงข้อมูลว่า “มีพนักงานคนไหนบ้างเคยทำงานในช่วงวันที่ 2010-01-01 ถึงวันที่ 2011-12-31”

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT emp_id, job_position,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM   positions e
WHERE  WM_CONTAINS(e.WM_VALID,
                  WM_PERIOD(TO_DATE('2010-01-01', 'YYYY-MM-DD'),
                            TO_DATE('2011-12-31', 'YYYY-MM-DD')))=1;
```

anonymous block completed

EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
85	Senior Engineer	2010-01-01	2012-01-01
101	Programmer	2010-01-01	2012-01-01
46	System Analyst	2005-02-16	2012-03-05
77	Manager	2008-01-01	9999-12-31
79	Programmer	2006-06-01	9999-12-31
84	System Analyst	2010-01-01	9999-12-31

6 rows selected

12. ทดลองเขียนคำสั่ง เพื่อแสดงข้อมูลว่า “มีพนักงานคนไหนบ้าง ที่ตำแหน่งงานเท่ากัน ในช่วงเวลาที่ WM_CONTAINS(e1, e2)”

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT e1.emp_id, e2.emp_id, e1.job_position,
       TO_CHAR(e1.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
       TO_CHAR(e1.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM   positions e1, positions e2
WHERE  e1.emp_id < e2.emp_id
AND    e1.job_position = e2.job_position
AND    WM_CONTAINS(e1.WM_VALID, e2.WM_VALID)=1;
```

anonymous block completed

EMP_ID	EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
46	47	Manager	2012-03-05	9999-12-31
79	101	Programmer	2006-06-01	9999-12-31
79	159	Programmer	2006-06-01	9999-12-31
79	160	Programmer	2006-06-01	9999-12-31
159	160	Programmer	2011-08-01	2012-04-30
46	84	System Analyst	2005-02-16	2012-03-05
45	84	System Analyst	2005-02-16	2008-01-01

7 rows selected

13. ทดลองเขียนคำสั่ง เพื่อแสดงข้อมูลว่า “มีพนักงานคนไหนบ้าง ที่ตำแหน่งงานเท่ากัน ในช่วงเวลาที่ WM_CONTAINS(e2, e1)”

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT e1.emp_id, e2.emp_id, e1.job_position,
       TO_CHAR(e1.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
       TO_CHAR(e1.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM positions e1, positions e2
WHERE e1.emp_id < e2.emp_id
AND e1.job_position = e2.job_position
AND WM_CONTAINS(e2.WM_VALID,e1.WM_VALID)=1;
```

anonymous block completed

EMP_ID	EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
159	160	Programmer	2011-08-01	2012-04-30
45	46	System Analyst	2005-02-16	2008-01-01
47	77	Manager	2012-06-01	9999-12-31
46	77	Manager	2012-03-05	9999-12-31
45	84	System Analyst	2005-02-16	2008-01-01

5 rows selected

หมายเหตุ ผลคำสั่ง ข้อ 12 และ 13 แสดงให้เห็นว่าลำดับของ WM_CONTAINS(e1, e2) และลำดับ WM_CONTAINS(e2, e1) ผลของคำสั่งไม่เหมือนกัน เพราะลำดับมีความสำคัญ

14. ทดลองเขียนคำสั่ง เพื่อแสดงข้อมูลว่า “มีพนักงานคนไหนบ้าง ที่ตำแหน่งงานเท่ากัน ในช่วงเวลาที่ WM_OVERLAPS(e1, e2)”

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT e1.emp_id, e2.emp_id, e1.job_position,
       TO_CHAR(e1.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
       TO_CHAR(e1.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM positions e1, positions e2
WHERE e1.emp_id < e2.emp_id
AND e1.job_position = e2.job_position
AND WM_OVERLAPS(e1.WM_VALID,e2.WM_VALID)=1;
```

anonymous block completed

EMP_ID	EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
46	47	Manager	2012-03-05	9999-12-31
79	101	Programmer	2006-06-01	9999-12-31
79	159	Programmer	2006-06-01	9999-12-31
101	159	Programmer	2010-01-01	2012-01-01
79	160	Programmer	2006-06-01	9999-12-31
159	160	Programmer	2011-08-01	2012-04-30
101	160	Programmer	2010-01-01	2012-01-01
45	46	System Analyst	2005-02-16	2008-01-01
47	77	Manager	2012-06-01	9999-12-31
46	77	Manager	2012-03-05	9999-12-31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

78          79          Programmer          2006-01-01 2006-11-01
46          84          System Analyst       2005-02-16 2012-03-05
45          84          System Analyst       2005-02-16 2008-01-01
46          84          System Analyst       2005-02-16 2012-03-05

```

14 rows selected

15. ทดลองเขียนคำสั่ง เพื่อแสดงข้อมูลว่า “มีพนักงานคนไหนบ้าง ที่มีตำแหน่งงานเท่ากัน ในช่วงเวลาที่ WM_OVERLAPS(e2, e1)”

```

EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT e1.emp_id, e2.emp_id, e1.job_position,
       TO_CHAR(e1.WM_VALID.VALIDFROM,'YYYY-MM-DD') FROM_DATE,
       TO_CHAR(e1.WM_VALID.VALIDTILL,'YYYY-MM-DD') TO_DATE
FROM positions e1, positions e2
WHERE e1.emp_id < e2.emp_id
AND e1.job_position = e2.job_position
AND WM_OVERLAPS(e2.WM_VALID,e1.WM_VALID)=1;

```

anonymous block completed

EMP_ID	EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
46	47	Manager	2012-03-05	9999-12-31
79	101	Programmer	2006-06-01	9999-12-31
79	159	Programmer	2006-06-01	9999-12-31
101	159	Programmer	2010-01-01	2012-01-01
79	160	Programmer	2006-06-01	9999-12-31
159	160	Programmer	2011-08-01	2012-04-30
101	160	Programmer	2010-01-01	2012-01-01
45	46	System Analyst	2005-02-16	2008-01-01
47	77	Manager	2012-06-01	9999-12-31
46	77	Manager	2012-03-05	9999-12-31
78	79	Programmer	2006-01-01	2006-11-01
46	84	System Analyst	2005-02-16	2012-03-05
45	84	System Analyst	2005-02-16	2008-01-01
46	84	System Analyst	2005-02-16	2012-03-05

14 rows selected

หมายเหตุ คำสั่งข้อ14 และข้อ15 มีผลเหมือนกัน คือ ลำดับของคำสั่ง WM_OVERLAPS() ไม่มีความสำคัญ ได้ผลเหมือนกัน

16. ทดลองเขียนคำสั่ง เพื่อแสดงข้อมูลว่า “มีพนักงานคนไหนบ้าง ที่มีตำแหน่งงานเท่ากัน” แบบตรวจสอบ 4 เงื่อนไข

```

1 EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);
2
3 SELECT e1.emp_id, e2.emp_id, e1.job_position,
4         TO_CHAR(e1.WM_VALID.VALIDFROM,'YYYY-MM-DD') FROM_DATE,
5         TO_CHAR(e1.WM_VALID.VALIDTILL,'YYYY-MM-DD') TO_DATE
6 FROM positions e1, positions e2
7 WHERE e1.emp_id < e2.emp_id
8 AND e1.job_position = e2.job_position
9 AND WM_CONTAINS(e1.WM_VALID,e2.WM_VALID)=1
10 UNION
11 SELECT e1.emp_id, e2.emp_id, e1.job_position,
12        TO_CHAR(e1.WM_VALID.VALIDFROM,'YYYY-MM-DD') FROM_DATE,
13        TO_CHAR(e1.WM_VALID.VALIDTILL,'YYYY-MM-DD') TO_DATE
14 FROM positions e1, positions e2
15 WHERE e1.emp_id < e2.emp_id
16 AND e1.job_position = e2.job_position
17 AND WM_CONTAINS(e2.WM_VALID,e1.WM_VALID)=1
18 UNION
19 SELECT e1.emp_id, e2.emp_id, e1.job_position,
20        TO_CHAR(e1.WM_VALID.VALIDFROM,'YYYY-MM-DD') FROM_DATE,
21        TO_CHAR(e1.WM_VALID.VALIDTILL,'YYYY-MM-DD') TO_DATE
22 FROM positions e1, positions e2
23 WHERE e1.emp_id < e2.emp_id
24 AND e1.job_position = e2.job_position
25 AND WM_OVERLAPS(e1.WM_VALID,e2.WM_VALID)=1
26 UNION
27 SELECT e1.emp_id, e2.emp_id, e1.job_position,
28        TO_CHAR(e1.WM_VALID.VALIDFROM,'YYYY-MM-DD') FROM_DATE,
29        TO_CHAR(e1.WM_VALID.VALIDTILL,'YYYY-MM-DD') TO_DATE
30 FROM positions e1, positions e2
31 WHERE e1.emp_id < e2.emp_id
32 AND e1.job_position = e2.job_position
33 AND WM_OVERLAPS(e2.WM_VALID,e1.WM_VALID)=1;

```

anonymous block completed

EMP_ID	EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
45	46	System Analyst	2005-02-16	2008-01-01
45	84	System Analyst	2005-02-16	2008-01-01
46	47	Manager	2012-03-05	9999-12-31
46	77	Manager	2012-03-05	9999-12-31
46	84	System Analyst	2005-02-16	2012-03-05
47	77	Manager	2012-06-01	9999-12-31
78	79	Programmer	2006-01-01	2006-11-01
79	101	Programmer	2006-06-01	9999-12-31
79	159	Programmer	2006-06-01	9999-12-31
79	160	Programmer	2006-06-01	9999-12-31
101	159	Programmer	2010-01-01	2012-01-01
101	160	Programmer	2010-01-01	2012-01-01
159	160	Programmer	2011-08-01	2012-04-30

13 rows selected

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

17. ทดลองเขียนคำสั่ง เพื่อแสดงข้อมูลว่า “มีพนักงานคนไหนบ้าง ที่มีตำแหน่งงานเท่ากัน” โดยเขียนคำสั่ง **WM_OVERLAPS()** มีลำดับเดียว เนื่องจาก **WM_OVERLAPS()** จะลำดับไหน จะได้ผลเหมือนกัน ผลของคำสั่งจะได้เหมือนกับข้อ 16 เช่นกัน แต่เขียนคำสั่งสั้นกว่า

```

1 EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);
2
3 SELECT e1.emp_id, e2.emp_id, e1.job_position,
4         TO_CHAR(e1.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
5         TO_CHAR(e1.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
6 FROM positions e1, positions e2
7 WHERE e1.emp_id < e2.emp_id
8 AND e1.job_position = e2.job_position
9 AND WM_CONTAINS(e1.WM_VALID,e2.WM_VALID)=1
10 UNION
11 SELECT e1.emp_id, e2.emp_id, e1.job_position,
12        TO_CHAR(e1.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
13        TO_CHAR(e1.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
14 FROM positions e1, positions e2
15 WHERE e1.emp_id < e2.emp_id
16 AND e1.job_position = e2.job_position
17 AND WM_CONTAINS(e2.WM_VALID,e1.WM_VALID)=1
18 UNION
19 SELECT e1.emp_id, e2.emp_id, e1.job_position,
20        TO_CHAR(e1.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
21        TO_CHAR(e1.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
22 FROM positions e1, positions e2
23 WHERE e1.emp_id < e2.emp_id
24 AND e1.job_position = e2.job_position
25 AND WM_OVERLAPS(e1.WM_VALID,e2.WM_VALID)=1;

```

anonymous block completed

EMP_ID	EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
45	46	System Analyst	2005-02-16	2008-01-01
45	84	System Analyst	2005-02-16	2008-01-01
46	47	Manager	2012-03-05	9999-12-31
46	77	Manager	2012-03-05	9999-12-31
46	84	System Analyst	2005-02-16	2012-03-05
47	77	Manager	2012-06-01	9999-12-31
78	79	Programmer	2006-01-01	2006-11-01
79	101	Programmer	2006-06-01	9999-12-31
79	159	Programmer	2006-06-01	9999-12-31
79	160	Programmer	2006-06-01	9999-12-31
101	159	Programmer	2010-01-01	2012-01-01
101	160	Programmer	2010-01-01	2012-01-01
159	160	Programmer	2011-08-01	2012-04-30

13 rows selected

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18. ทดลองเขียนคำสั่ง เพื่อแสดงข้อมูลว่า “มีพนักงานคนไหนบ้าง ที่มีตำแหน่งงานเท่ากัน ณ ช่วงเวลาปัจจุบัน”

```

1 EXECUTE DBMS_WM.SetValidTime( );
2
3 SELECT e1.emp_id, e2.emp_id, e1.job_position,
4        TO_CHAR(e1.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
5        TO_CHAR(e1.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
6 FROM positions e1, positions e2
7 WHERE e1.emp_id < e2.emp_id
8 AND e1.job_position = e2.job_position
9 AND WM_CONTAINS(e1.WM_VALID, e2.WM_VALID)=1
10 UNION
11 SELECT e1.emp_id, e2.emp_id, e1.job_position,
12        TO_CHAR(e1.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
13        TO_CHAR(e1.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
14 FROM positions e1, positions e2
15 WHERE e1.emp_id < e2.emp_id
16 AND e1.job_position = e2.job_position
17 AND WM_CONTAINS(e2.WM_VALID, e1.WM_VALID)=1
18 UNION
19 SELECT e1.emp_id, e2.emp_id, e1.job_position,
20        TO_CHAR(e1.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
21        TO_CHAR(e1.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
22 FROM positions e1, positions e2
23 WHERE e1.emp_id < e2.emp_id
24 AND e1.job_position = e2.job_position
25 AND WM_OVERLAPS(e1.WM_VALID, e2.WM_VALID)=1;

```

anonymous block completed

EMP_ID	EMP_ID	JOB_POSITION	FROM_DATE	TO_DATE
46	47	Manager	2012-03-05	9999-12-31
46	77	Manager	2012-03-05	9999-12-31
46	84	System Analyst	2005-02-16	2012-03-05
47	77	Manager	2012-06-01	9999-12-31
79	159	Programmer	2006-06-01	9999-12-31
79	160	Programmer	2006-06-01	9999-12-31
159	160	Programmer	2011-08-01	2012-04-30

7 rows selected

19. สรุปผลการทดลอง

การใช้คำสั่ง WM_CONTAINS() WM_OVERLAPS() WM_EQUALS() ของ Oracle11g

ทำให้การเขียนคำสั่งง่าย ผู้เขียนคำสั่งไม่ต้องเขียนคำสั่งตรวจสอบ FROM_DATE และ TO_DATE ว่าจะ เป็นแบบไหน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 การทดลอง Transaction-Time State Tables

ในการทดลอง เกี่ยวกับตาราง Transaction-Time State Table จะจำลองโดยการสร้าง ตาราง WDS แล้วใช้คำสั่ง EnableVersioning ทำให้เป็น Transaction-Time State Table ซึ่งใน Oracle11g จะเพิ่มเป็นตาราง WDS_L

จากนั้นทำการเพิ่มข้อมูล (Insert) และปรับปรุงข้อมูล (Update) ตามข้อมูลในตาราง WDS_TT จากการพิจารณาการเปลี่ยนแปลงแก้ไขข้อมูล ตามช่วงเวลา จากคอลัมน์ Trans_Start เพื่อให้เข้าใจง่าย จะใส่ตัวเลขลำดับการเปลี่ยนแปลงแก้ไขเพิ่มในตาราง 2.13 ซึ่งแสดงในตารางที่ 3.1

ตารางที่ 3.1 WDS_TT เป็นตาราง Audit Log ของตาราง WDS [3]

NO	RA_Hour	RA_Min	RA_Sec	Dec_Degree	Dec_Minute	Discoverer	Mag_First	Trans_Start	Trans_Stop
2	00	00	08	75	30	'A 1248'	12.0	1989-03-12	1992-11-15
5	00	00	09	75	30	'A 1248'	12.0	1992-11-15	1994-05-18
7	00	00	09	75	30	'A 1248'	10.5	1994-05-18	1995-07-23
8	00	00	08	75	30	'A 1248'	10.5	1995-07-23	9999-12-31
1	05	57	40	00	02	'BU 1190'	6.5	1988-11-08	9999-12-31
3	04	13	20	50	32	'CHR 15'	15.5	1990-02-09	9999-12-31
4	01	23	70	-09	55	'HJ 3433'	10.5	1991-03-25	9999-12-31
6	02	33	10	-09	25	'LDS3402'	10.6	1993-12-19	1996-07-09

จากนั้นทำการเพิ่มและแก้ไขตามลำดับ สังเกตการเปลี่ยนแปลงข้อมูลใน WDS_LT และการเปลี่ยนแปลงแก้ไขที่วิว WDS (วิวดูค่าปัจจุบัน) และ WDS_HIST (สำหรับดูการเปลี่ยนแปลงข้อมูลของ User)

3.7.1 สร้างตารางชื่อ WDS (Washington Double Start)

```
CREATE TABLE wds (
  ra_hour      NUMBER,
  ra_min       NUMBER,
  ra_sec       NUMBER,
  dec_degree   NUMBER,
  dec_minute   NUMBER,
  discoverer   VARCHAR2(10) PRIMARY KEY,
  mag_first    NUMBER
);
```

CREATE TABLE succeeded.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.2 กำหนด **EnableVersioning** ให้ตาราง **WDS** เป็นตารางแบบ **Transaction Time State Table** โดยกำหนดพารามิเตอร์ ดังนี้

Table_name : 'wds'
 hist : 'VIEW_WO_OVERWRITE' อนุญาตเขียนข้อมูลเพิ่มได้ในตาราง _hist
 isTopology : FALSE กำหนดตารางไม่เป็นตารางแบบ Oracle Spatial Topology
 ValidTime : FALSE กำหนดตารางไม่เป็น Valid Time State Table

```
EXECUTE DBMS_WM.EnableVersioning('wds', 'VIEW_WO_OVERWRITE', FALSE, FALSE);
```

anonymous block completed

3.7.3 ทำการเปลี่ยนระบบวันที่ของเครื่องคอมพิวเตอร์ให้เป็นวันที่ 1988-11-08 จากนั้นเพิ่มข้อมูลดาว 'BU 1190' ลงในตาราง **WDS** แล้วทำการตรวจสอบดูการเปลี่ยนแปลงข้อมูลที่ **WDS**, **WDS_LT**, **WDS_HIST** ตามลำดับ

```
INSERT INTO wds VALUES(05, 57, 40, 00, 02, 'BU 1190', 6.5);
```

1 rows inserted

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST
5	57	40	00	02	2 BU 1190	6.5

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST	VERSION	CREATETIME	RETIRETIME	NEXTVER	DELSTATUS	LTLOCK
5	57	40	00	02	2 BU 1190	6.5	0	08-DEC-88 09:55:28.616000000 PM +07:00	(null)	-1	10 *-1,-9,-1*#SYSMAN#	

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST	WM_WORKSPACE	WM_VERSION	WM_USERNAME	WM_OPTYPE	WM_CREATETIME	WM_RETIRETIME
5	57	40	00	02	2 BU 1190	6.5	LIVE	0	SYSMAN	I	08-DEC-88 09:55:28.616000000 PM +07:00	(null)

รูปที่ 3.14 ผลการทำงานของเพิ่มข้อมูลตามข้อ 3.7.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.4 ทำการเปลี่ยนระบบวันที่ของเครื่องคอมพิวเตอร์ให้เป็นวันที่ 1989-03-12 จากนั้นเพิ่มข้อมูลดาว 'A 1248' ลงในตาราง WDS แล้วทำการตรวจสอบดูการเปลี่ยนแปลงข้อมูลที่ WDS, WDS_LT, WDS_HIST ตามลำดับ

```
INSERT INTO wds VALUES(00, 00, 00, 75, 30, 'A 1248', 12.0);
```

1 rows inserted

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST
1	5	57	40	0	2 BU 1190	6.5
2	0	0	0	75	30 A 1248	12

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST	VERSION	CREATETIME	RETIRETIME	NEXTVER	DELSTATUS	LTLOCK
1	5	57	40	0	2 BU 1190	6.5	0	08-DEC-88 09.55.28.616000000 PM +07:00	(null)	-1	10 *-1,-9,-1*#SYSMAN\$#	
2	0	0	0	75	30 A 1248	12	0	12-MAR-89 10.28.13.869000000 PM +07:00	(null)	-1	10 *-1,-9,-1*#SYSMAN\$#	

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST	WM_WORKSPACE	WM_VERSION	WM_USERNAME	WM_OPTYPE	WM_CREATETIME	WM_RETIRETIME
1	5	57	40	0	2 BU 1190	6.5	LIVE	0	SYSMAN	I	08-DEC-88 09.55.28.616000000 PM +07:00	(null)
2	0	0	0	75	30 A 1248	12	LIVE	0	SYSMAN	I	12-MAR-89 10.28.13.869000000 PM +07:00	(null)

รูปที่ 3.15 ผลข้อมูล WDS WDS_LT และ WDS_HIST ตามข้อ 3.7.4

3.7.5 ทำการเปลี่ยนระบบวันที่ของเครื่องคอมพิวเตอร์ให้เป็นวันที่ 1990-02-09 จากนั้นเพิ่มข้อมูลดาว 'CHR 15' ลงในตาราง WDS แล้วทำการตรวจสอบดูการเปลี่ยนแปลงข้อมูลที่ WDS, WDS_LT, WDS_HIST ตามลำดับ

```
INSERT INTO wds VALUES(04, 13, 20, 50, 32, 'CHR 15', 15.5);
```

1 rows inserted

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST
1	5	57	40	0	2 BU 1190	6.5
2	0	0	0	75	30 A 1248	12
3	4	13	20	50	32 CHR 15	15.5

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST	VERSION	CREATETIME	RETIRETIME	NEXTVER	DELSTATUS	LTLOCK
1	5	57	40	0	2 BU 1190	6.5	0	08-DEC-88 09.55.28.616000000 PM +07:00	(null)	-1	10 *-1,-9,-1*#SYSMAN\$#	
2	0	0	0	75	30 A 1248	12	0	12-MAR-89 10.28.13.869000000 PM +07:00	(null)	-1	10 *-1,-9,-1*#SYSMAN\$#	
3	4	13	20	50	32 CHR 15	15.5	0	09-FEB-90 10.41.20.563000000 PM +07:00	(null)	-1	10 *-1,-9,-1*#SYSMAN\$#	

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST	WM_WORKSPACE	WM_VERSION	WM_USERNAME	WM_OPTYPE	WM_CREATETIME	WM_RETIRETIME
1	5	57	40	0	2 BU 1190	6.5	LIVE	0	SYSMAN	I	08-DEC-88 09.55.28.616000000 PM +07:00	(null)
2	0	0	0	75	30 A 1248	12	LIVE	0	SYSMAN	I	12-MAR-89 10.28.13.869000000 PM +07:00	(null)
3	4	13	20	50	32 CHR 15	15.5	LIVE	0	SYSMAN	I	09-FEB-90 10.41.20.563000000 PM +07:00	(null)

รูปที่ 3.16 ผลข้อมูลจาก WDS WDS_LT และ WDS_HIST ตามข้อ 3.7.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.6 ทำการเปลี่ยนระบบวันที่ของเครื่องคอมพิวเตอร์ให้เป็นวันที่ 1991-03-25 จากนั้นเพิ่มข้อมูลดาว 'HJ 3433' ลงในตาราง WDS แล้วทำการตรวจสอบดูการเปลี่ยนแปลงข้อมูลที่ WDS, WDS_LT, WDS_HIST ตามลำดับ

```
INSERT INTO wds VALUES(01, 23, 70, -09, 55, 'HJ 3433', 10.5);
```

1 rows inserted

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST
1	5	57	40	0	2 BU 1190	6.5
2	0	0	0	75	30 A 1248	12
3	4	13	20	50	32 CHR 15	15.5
4	1	23	70	-9	55 HJ 3433	10.5

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST	VERSION	CREATETIME	RETIETIME	NEXTVER	DELSTATUS	LOCK
1	5	57	40	0	2 BU 1190	6.5	0 08-DEC-88 09.55.28.616000000 PM +07:00	(null)	(null)	-1	10 *-1,-9,-1*#SYSMAN\$#	
2	0	0	0	75	30 A 1248	12	0 12-MAR-89 10.28.13.869000000 PM +07:00	(null)	(null)	-1	10 *-1,-9,-1*#SYSMAN\$#	
3	4	13	20	50	32 CHR 15	15.5	0 09-FEB-90 10.41.20.563000000 PM +07:00	(null)	(null)	-1	10 *-1,-9,-1*#SYSMAN\$#	
4	1	23	70	-9	55 HJ 3433	10.5	0 09-FEB-90 10.47.21.658000000 PM +07:00	(null)	(null)	-1	10 *-1,-9,-1*#SYSMAN\$#	

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST	WM_WORKSPACE	WM_VERSION	WM_USERNAME	WM_OPTYPE	WM_CREATETIME	WM_RETIRETIME
1	5	57	40	0	2 BU 1190	6.5	LIVE	0	SYSMAN	I	08-DEC-88 09.55.28.616000000 PM +07:00	(null)
2	0	0	0	75	30 A 1248	12	LIVE	0	SYSMAN	I	12-MAR-89 10.28.13.869000000 PM +07:00	(null)
3	4	13	20	50	32 CHR 15	15.5	LIVE	0	SYSMAN	I	09-FEB-90 10.41.20.563000000 PM +07:00	(null)
4	1	23	70	-9	55 HJ 3433	10.5	LIVE	0	SYSMAN	I	09-FEB-90 10.47.21.658000000 PM +07:00	(null)

รูปที่ 3.17 ผลข้อมูลจาก WDS WDS_LT และ WDS_HIST ตามข้อ 3.7.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.7 ทำการเปลี่ยนระบบวันที่ของเครื่องคอมพิวเตอร์ให้เป็นวันที่ 1992-11-15 จากนั้นปรับปรุงข้อมูลดาว 'A 1248' ให้ RA_Sec=09 ด้วยคำสั่ง Update แล้วทำการตรวจสอบดูการเปลี่ยนแปลงข้อมูลที่ WDS, WDS_LT, WDS_HIST ตามลำดับ

```
UPDATE wds
SET ra_sec=09
WHERE discoverer='A 1248';
1 rows updated
```

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST
1	5	57	40	0	2 BU 1190	6.5
2	4	13	20	50	32 CHR 15	15.5
3	1	23	70	-9	55 HJ 3433	10.5
4	0	0	9	75	30 A 1248	12

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST	VERSION	CREATETIME	RETIRETIME	NEXTVR	DELSTATUS	LTLOCK
1	5	57	40	0	2 BU 1190	6.5	0	08-DEC-88 09:55:28.616000000 PM +07:00	(null)	-1		10 *-1,-9,-1*#SYSMAN\$#
2	0	0	0	75	30 A 1248	12	0	12-MAR-89 10:28:13.869000000 PM +07:00	15-NOV-92 10:54:07.869000000 PM +07:00	,0,		10 *-1,-9,-1*#SYSMAN\$#
3	4	13	20	50	32 CHR 15	15.5	0	09-FEB-90 10:41:20.563000000 PM +07:00	(null)	-1		10 *-1,-9,-1*#SYSMAN\$#
4	1	23	70	-9	55 HJ 3433	10.5	0	09-FEB-90 10:47:21.658000000 PM +07:00	(null)	-1		10 *-1,-9,-1*#SYSMAN\$#
5	0	0	9	75	30 A 1248	12	0	15-NOV-92 10:54:07.869000000 PM +07:00	(null)	-1		11 *-1,-9,0*#SYSMAN\$#

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST	WM_WORKSPACE	WM_VERSION	WM_USERNAME	WM_OPTYPE	WM_CREATETIME	WM_RETIRETIME
1	5	57	40	0	2 BU 1190	6.5	LIVE	0	SYSMAN	I	08-DEC-88 09:55:28.616000000 PM +07:00	(null)
2	0	0	0	75	30 A 1248	12	LIVE	0	SYSMAN	I	12-MAR-89 10:28:13.869000000 PM +07:00	15-NOV-92 10:54:07.869000000 PM +07:00
3	4	13	20	50	32 CHR 15	15.5	LIVE	0	SYSMAN	I	09-FEB-90 10:41:20.563000000 PM +07:00	(null)
4	1	23	70	-9	55 HJ 3433	10.5	LIVE	0	SYSMAN	I	09-FEB-90 10:47:21.658000000 PM +07:00	(null)
5	0	0	9	75	30 A 1248	12	LIVE	0	SYSMAN	U	15-NOV-92 10:54:07.869000000 PM +07:00	(null)

รูปที่ 3.18 ผลข้อมูลจาก WDS WDS_LT และ WDS_HIST ตามข้อ 3.7.7

จากตาราง WDS และ WDS_LT จะเห็นว่า

- 1) ตาราง WDS มีการปรับปรุงข้อมูลดาว 'A 1248' คอลัมน์ RA_SEC=9 โดยจำนวนแถวทั้งหมดยังคง 4 แถว
- 2) ตาราง WDS_LT มีการปรับปรุงดาว 'A 1248' (แถวที่ 2) RETIRETIME = 15-NOV-92 และ ทำการเพิ่มแถวข้อมูลใหม่ (แถวที่ 5) ดาว 'A 1248', CREATETIME = 15-NOV-92 และ คอลัมน์ RETIRETIME = NULL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.8 ทำการเปลี่ยนระบบวันที่ของเครื่องคอมพิวเตอร์ให้เป็นวันที่ 1993-12-19 จากนั้นเพิ่มข้อมูลดาว 'HJ 3433' ลงในตาราง WDS แล้วทำการตรวจสอบดูการเปลี่ยนแปลงข้อมูลที่ WDS, WDS_LT, WDS_HIST ตามลำดับ

```
INSERT INTO wds VALUES(02, 33, 10, -09, 25, 'LDS3402', 10.6);
```

1 rows inserted

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST
1	5	57	40	0	2 BU 1190	6.5
2	4	13	20	50	32 CHR 15	15.5
3	1	23	70	-9	55 HJ 3433	10.5
4	0	0	9	75	30 A 1248	12
5	2	33	10	-9	25 LDS3402	10.6

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST	VERSION	CREATETIME	RETIRETIME	NEXTVER	DELSTATUS	LOCK
1	5	57	40	0	2 BU 1190	6.5	0	08-DEC-88 09.55.28.616000000 PM +07:00	(null)	-1	10 *-1,-9,-1*SYSMAN\$#	
2	0	0	0	75	30 A 1248	12	0	12-MAR-89 10.28.13.869000000 PM +07:00	15-NOV-92 10.54.07.869000000 PM +07:00	0	10 *-1,-9,-1*SYSMAN\$#	
3	4	13	20	50	32 CHR 15	15.5	0	09-FEB-90 10.41.20.563000000 PM +07:00	(null)	-1	10 *-1,-9,-1*SYSMAN\$#	
4	1	23	70	-9	55 HJ 3433	10.5	0	09-FEB-90 10.47.21.658000000 PM +07:00	(null)	-1	10 *-1,-9,-1*SYSMAN\$#	
5	0	0	9	75	30 A 1248	12	0	15-NOV-92 10.54.07.869000000 PM +07:00	(null)	-1	11 *-1,-9,0*SYSMAN\$#	
6	2	33	10	-9	25 LDS3402	10.6	0	19-DEC-93 11.03.38.700000000 PM +07:00	(null)	-1	10 *-1,-9,-1*SYSMAN\$#	

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST	WM_WORKSPACE	WM_VERSION	WM_USERNAME	WM_OPTYPE	WM_CREATETIME	WM_RETIRETIME
1	5	57	40	0	2 BU 1190	6.5	LIVE		0 SYSMAN	I	08-DEC-88 09.55.28.616000000 PM +07:00	(null)
2	0	0	0	75	30 A 1248	12	LIVE		0 SYSMAN	I	12-MAR-89 10.28.13.869000000 PM +07:00	15-NOV-92 10.54.07.869000000 PM +07:00
3	4	13	20	50	32 CHR 15	15.5	LIVE		0 SYSMAN	I	09-FEB-90 10.41.20.563000000 PM +07:00	(null)
4	1	23	70	-9	55 HJ 3433	10.5	LIVE		0 SYSMAN	I	09-FEB-90 10.47.21.658000000 PM +07:00	(null)
5	0	0	9	75	30 A 1248	12	LIVE		0 SYSMAN	U	15-NOV-92 10.54.07.869000000 PM +07:00	(null)
6	2	33	10	-9	25 LDS3402	10.6	LIVE		0 SYSMAN	I	19-DEC-93 11.03.38.700000000 PM +07:00	(null)

รูปที่ 3.19 ผลข้อมูลจาก WDS WDS_LT และ WDS_HIST ตามข้อ 3.7.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.9 ทำการเปลี่ยนระบบวันที่ของเครื่องคอมพิวเตอร์ให้เป็นวันที่ 1994-05-18 จากนั้นปรับปรุงข้อมูลดาว 'A 1248' ให้ Mag_first=10.5 ด้วยคำสั่ง Update แล้วทำการตรวจสอบดูการเปลี่ยนแปลงข้อมูลที่ WDS, WDS_LT, WDS_HIST ตามลำดับ

```
UPDATE wds
SET mag_first=10.5
WHERE discoverer='A 1248';
```

1 rows updated

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST
1	5	57	40	0	2 BU 1190	6.5
2	4	13	20	50	32 CHR 15	15.5
3	1	23	70	-9	55 HJ 3433	10.5
4	2	33	10	-9	25 LDS3402	10.6
5	0	0	9	75	30 A 1248	10.5

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST	VERSION	CREATETIME	RETIRETIME	NEXTVER	DELSTATUS	LTLOCK
1	5	57	40	0	2 BU 1190	6.5	0	08-DEC-88 09:55:28.616000000 PM +07:00	(null)	-1		10 ^-1,-9,-1*#SYSMAN\$#
2	0	0	0	75	30 A 1248	12	0	12-MAR-89 10:28:13.869000000 PM +07:00	15-NOV-92 10:54:07.869000000 PM +07:00	,0		10 ^-1,-9,-1*#SYSMAN\$#
3	4	13	20	50	32 CHR 15	15.5	0	09-FEB-90 10:41:20.563000000 PM +07:00	(null)	-1		10 ^-1,-9,-1*#SYSMAN\$#
4	1	23	70	-9	55 HJ 3433	10.5	0	09-FEB-90 10:47:21.658000000 PM +07:00	(null)	-1		10 ^-1,-9,-1*#SYSMAN\$#
5	0	0	9	75	30 A 1248	12	0	15-NOV-92 10:54:07.869000000 PM +07:00	18-MAY-94 11:08:33.076000000 PM +07:00	,0		11 ^-1,-9,0*#SYSMAN\$#
6	2	33	10	-9	25 LDS3402	10.6	0	19-DEC-93 11:03:38.700000000 PM +07:00	(null)	-1		10 ^-1,-9,-1*#SYSMAN\$#
7	0	0	9	75	30 A 1248	10.5	0	18-MAY-94 11:08:33.076000000 PM +07:00	(null)	-1		12 ^-1,-9,0*#SYSMAN\$#

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST	WM_WORKSPACE	WM_VERSION	WM_USERNAME	WM_OPTYPE	WM_CREATETIME	WM_RETIRETIME
1	5	57	40	0	2 BU 1190	6.5	LIVE		0 SYSMAN	I	08-DEC-88 09:55:28.616000000 PM +07:00	(null)
2	0	0	0	75	30 A 1248	12	LIVE		0 SYSMAN	I	12-MAR-89 10:28:13.869000000 PM +07:00	15-NOV-92 10:54:07.869000000 PM +07:00
3	4	13	20	50	32 CHR 15	15.5	LIVE		0 SYSMAN	I	09-FEB-90 10:41:20.563000000 PM +07:00	(null)
4	1	23	70	-9	55 HJ 3433	10.5	LIVE		0 SYSMAN	I	09-FEB-90 10:47:21.658000000 PM +07:00	(null)
5	0	0	9	75	30 A 1248	12	LIVE		0 SYSMAN	U	15-NOV-92 10:54:07.869000000 PM +07:00	18-MAY-94 11:08:33.076000000 PM +07:00
6	2	33	10	-9	25 LDS3402	10.6	LIVE		0 SYSMAN	I	19-DEC-93 11:03:38.700000000 PM +07:00	(null)
7	0	0	9	75	30 A 1248	10.5	LIVE		0 SYSMAN	U	18-MAY-94 11:08:33.076000000 PM +07:00	(null)

รูปที่ 3.20 ผลข้อมูลจาก WDS WDS_LT และ WDS_HIST ตามข้อ 3.7.9

จากตาราง WDS และ WDS_LT จะเห็นว่า

1) ตาราง WDS มีการปรับปรุงข้อมูลดาว 'A 1248' คอลัมน์ Mag_first=10.5 โดยจำนวนแถวทั้งหมดยังคง 5 แถว

2) ตาราง WDS_LT มีการปรับปรุงดาว 'A 1248' (แถวที่ 5) RETIRETIME = 18-MAY-94 และ ทำการเพิ่มแถวข้อมูลใหม่ (แถวที่ 7) ดาว 'A 1248' , CREATETIME = 18-MAY-94 และ คอลัมน์ RETIRETIME = NULL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.10 ทำการเปลี่ยนระบบวันที่ของเครื่องคอมพิวเตอร์ให้เป็นวันที่ 1995-07-23 จากนั้นปรับปรุงข้อมูลดาว 'A 1248' ให้ RA_SEC=08 ด้วยคำสั่ง Update แล้วทำการตรวจสอบดูการเปลี่ยนแปลงข้อมูลที่ WDS, WDS_LT, WDS_HIST ตามลำดับ

```
UPDATE wds
SET ra_sec=08
WHERE discoverer='A 1248';
```

1 rows updated

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST
1	5	57	40	0	2 BU 1190	6.5
2	4	13	20	50	32 CHR 15	15.5
3	1	23	70	-9	55 HJ 3433	10.5
4	2	33	10	-9	25 LDS3402	10.6
5	0	0	8	75	30 A 1248	10.5

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST	VERSION	CREATETIME	RETIRETIME	NEXTVR	DELSTATUS	LTLOCK
1	5	57	40	0	2 BU 1190	6.5	0	08-DEC-88 09:55:28.616000000 PM +07:00	(null)	-1	10 *-1,-9,-1*#SYSMAN\$#	
2	0	0	0	75	30 A 1248	12	0	12-MAR-89 10:28:13.869000000 PM +07:00	15-NOV-92 10:54:07.869000000 PM +07:00	,0	10 *-1,-9,-1*#SYSMAN\$#	
3	4	13	20	50	32 CHR 15	15.5	0	09-FEB-90 10:41:20.563000000 PM +07:00	(null)	-1	10 *-1,-9,-1*#SYSMAN\$#	
4	1	23	70	-9	55 HJ 3433	10.5	0	09-FEB-90 10:47:21.658000000 PM +07:00	(null)	-1	10 *-1,-9,-1*#SYSMAN\$#	
5	0	0	9	75	30 A 1248	12	0	15-NOV-92 10:54:07.869000000 PM +07:00	18-MAY-94 11:08:33.076000000 PM +07:00	,0	11 *-1,-9,0*#SYSMAN\$#	
6	2	33	10	-9	25 LDS3402	10.6	0	19-DEC-93 11:03:38.700000000 PM +07:00	(null)	-1	10 *-1,-9,-1*#SYSMAN\$#	
7	0	0	9	75	30 A 1248	10.5	0	18-MAY-94 11:08:33.076000000 PM +07:00	23-JUL-95 11:14:21.494000000 PM +07:00	,0	12 *-1,-9,0*#SYSMAN\$#	
8	0	0	8	75	30 A 1248	10.5	0	23-JUL-95 11:14:21.494000000 PM +07:00	(null)	-1	13 *-1,-9,0*#SYSMAN\$#	

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST	WM_WORKSPACE	WM_VERSION	WM_USERNAME	WM_OPTYPE	WM_CREATETIME	WM_RETIRETIME
1	5	57	40	0	2 BU 1190	6.5	LIVE		0 SYSMAN	I	08-DEC-88 09:55:28.616000000 PM +07:00	(null)
2	0	0	0	75	30 A 1248	12	LIVE		0 SYSMAN	I	12-MAR-89 10:28:13.869000000 PM +07:00	15-NOV-92 10:54:07.869000000 PM +07:00
3	4	13	20	50	32 CHR 15	15.5	LIVE		0 SYSMAN	I	09-FEB-90 10:41:20.563000000 PM +07:00	(null)
4	1	23	70	-9	55 HJ 3433	10.5	LIVE		0 SYSMAN	I	09-FEB-90 10:47:21.658000000 PM +07:00	(null)
5	0	0	9	75	30 A 1248	12	LIVE		0 SYSMAN	U	15-NOV-92 10:54:07.869000000 PM +07:00	18-MAY-94 11:08:33.076000000 PM +07:00
6	2	33	10	-9	25 LDS3402	10.6	LIVE		0 SYSMAN	I	19-DEC-93 11:03:38.700000000 PM +07:00	(null)
7	0	0	9	75	30 A 1248	10.5	LIVE		0 SYSMAN	U	18-MAY-94 11:08:33.076000000 PM +07:00	23-JUL-95 11:14:21.494000000 PM +07:00
8	0	0	8	75	30 A 1248	10.5	LIVE		0 SYSMAN	U	23-JUL-95 11:14:21.494000000 PM +07:00	(null)

รูปที่ 3.21 ผลข้อมูลจาก WDS WDS_LT และ WDS_HIST ตามข้อ 3.7.10

จากตาราง WDS และ WDS_LT จะเห็นว่า

1) ตาราง WDS มีการปรับปรุงข้อมูลดาว 'A 1248' คอลัมน์ RA_SEC=08 โดยจำนวนแถวทั้งหมดยังคง 5 แถว

2) ตาราง WDS_LT มีการปรับปรุงดาว 'A 1248' (แถวที่ 7) RETIRETIME = 23-JUL-95 และ ทำการเพิ่มแถวข้อมูลใหม่ (แถวที่ 8) ดาว 'A 1248', CREATETIME = 23-JUL-95 และ คอลัมน์ RETIRETIME = NULL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.11 ทำการเปลี่ยนระบบวันที่ของเครื่องคอมพิวเตอร์ให้เป็นวันที่ 2012-12-12 จากนั้นทำการลบข้อมูลดาว 'A 1248' ออกจากตาราง WDS แล้วทำการตรวจสอบดูการเปลี่ยนแปลงข้อมูลที่ WDS, WDS_LT, WDS_HIST ตามลำดับ

```
DELETE wds
WHERE discoverer='A 1248';
```

1 rows deleted

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST
1	5	57	40	0	2 BU 1190	6.5
2	4	13	20	50	32 CHR 15	15.5
3	1	23	70	-9	55 HJ 3433	10.5
4	2	33	10	-9	25 LDS3402	10.6

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST	VERSION	CREATETIME	RETIRETIME	NEXTVR	DELSTATUS	LTLOCK
1	5	57	40	0	2 BU 1190	6.5	0	08-DEC-88 09.55.28.616000000 PM +07:00	(null)	-1	10 *-1,-9,-1*#SYSMAN\$#	
2	0	0	0	75	30 A 1248	12	0	12-MAR-89 10.28.13.869000000 PM +07:00	15-NOV-92 10.54.07.869000000 PM +07:00	,0	10 *-1,-9,-1*#SYSMAN\$#	
3	4	13	20	50	32 CHR 15	15.5	0	09-FEB-90 10.41.20.563000000 PM +07:00	(null)	-1	10 *-1,-9,-1*#SYSMAN\$#	
4	1	23	70	-9	55 HJ 3433	10.5	0	09-FEB-90 10.47.21.658000000 PM +07:00	(null)	-1	10 *-1,-9,-1*#SYSMAN\$#	
5	0	0	9	75	30 A 1248	12	0	15-NOV-92 10.54.07.869000000 PM +07:00	18-MAY-94 11.08.33.076000000 PM +07:00	,0	11 *-1,-9,0*#SYSMAN\$#	
6	2	33	10	-9	25 LDS3402	10.6	0	19-DEC-93 11.03.38.700000000 PM +07:00	(null)	-1	10 *-1,-9,-1*#SYSMAN\$#	
7	0	0	9	75	30 A 1248	10.5	0	18-MAY-94 11.08.33.076000000 PM +07:00	23-JUL-95 11.14.21.494000000 PM +07:00	,0	12 *-1,-9,0*#SYSMAN\$#	
8	0	0	8	75	30 A 1248	10.5	0	23-JUL-95 11.14.21.494000000 PM +07:00	(null)	-1	13 *-1,-9,0*#SYSMAN\$#	

RA_HOUR	RA_MIN	RA_SEC	DEC_DEGREE	DEC_MINUTE	DISCOVERER	MAG_FIRST	WM_WORKSPACE	WM_VERSION	WM_USERNAME	WM_OTYPE	WM_CREATETIME	WM_RETIRETIME
1	5	57	40	0	2 BU 1190	6.5	LIVE		0 SYSMAN	I	08-DEC-88 09.55.28.616000000 PM +07:00	(null)
2	0	0	0	75	30 A 1248	12	LIVE		0 SYSMAN	I	12-MAR-89 10.28.13.869000000 PM +07:00	15-NOV-92 10.54.07.869000000 PM +07:00
3	4	13	20	50	32 CHR 15	15.5	LIVE		0 SYSMAN	I	09-FEB-90 10.41.20.563000000 PM +07:00	(null)
4	1	23	70	-9	55 HJ 3433	10.5	LIVE		0 SYSMAN	I	09-FEB-90 10.47.21.658000000 PM +07:00	(null)
5	0	0	9	75	30 A 1248	12	LIVE		0 SYSMAN	U	15-NOV-92 10.54.07.869000000 PM +07:00	18-MAY-94 11.08.33.076000000 PM +07:00
6	2	33	10	-9	25 LDS3402	10.6	LIVE		0 SYSMAN	I	19-DEC-93 11.03.38.700000000 PM +07:00	(null)
7	0	0	9	75	30 A 1248	10.5	LIVE		0 SYSMAN	U	18-MAY-94 11.08.33.076000000 PM +07:00	23-JUL-95 11.14.21.494000000 PM +07:00
8	0	0	8	75	30 A 1248	10.5	LIVE		0 SYSMAN	U	23-JUL-95 11.14.21.494000000 PM +07:00	12-DEC-12 11.22.13.872000000 PM +07:00
9	0	0	8	75	30 A 1248	10.5	LIVE		0 SYSMAN	D	12-DEC-12 11.22.13.872000000 PM +07:00	(null)

รูปที่ 3.22 ผลข้อมูลจาก WDS WDS_LT และ WDS_HIST ตามข้อ 3.7.11

จากตาราง WDS และ WDS_LT จะเห็นว่า

1) ตาราง WDS มีการลบข้อมูลดาว 'A 1248' ออกจากตาราง WDS ทำให้จำนวนแถวทั้งหมดเหลือ 4 แถว

2) ตาราง WDS_LT มีการปรับปรุงดาว 'A 1248' (แถวที่ 8) RETIRETIME = 29-MAR-12 และทำการเพิ่มแถวข้อมูลใหม่ (แถวที่ 9) ดาว 'A 1248', CREATETIME = 29-MAR-12 และคอลัมน์ RETIRETIME = NULL

3.6 การทดลอง Bitemporal Tables

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 ข้อแตกต่างระหว่าง DBMS_WM.MAX_TIME กับ DBMS_WM.UNTIL_CHANGE

ตารางที่ 3.2 คำอธิบาย MIN_TIME MAX_TIME และ UNTIL_CHANGED

Constant	คำอธิบาย
DBMS_WM.MIN_TIME	ค่าเวลาที่น้อยที่สุดเท่าที่ Workspace Manager จะสนับสนุนได้ ซึ่งขณะนี้เริ่มต้นที่ วันที่ 1 มกราคม ค.ศ. -4713 (4713 BCE)
DBMS_WM.MAX_TIME	ค่าเวลาที่มากที่สุดเท่าที่ Workspace Manager จะสนับสนุนได้ ซึ่งขณะนี้วันสุดท้าย คือ วันที่ 31 ธันวาคม ค.ศ. 9999 (ภายในเวลา 11:59.999999000 pm)
DBMS_WM.UNTIL_CHANGED	คือ การคงค่าของเวลา ไว้ที่ DBMS_WM.MAX_TIME จนกว่าจะมีการเปลี่ยนแปลงค่าของข้อมูล

3.8.1 สร้างตารางชื่อ NAMES ประกอบด้วย EMP_ID=รหัสพนักงาน และ EMP_NAME=ชื่อพนักงาน

```
CREATE TABLE names (
  emp_id NUMBER PRIMARY KEY,
  emp_name VARCHAR2(20)
);
```

CREATE TABLE succeeded.

3.8.2 กำหนด EnableVersioning ให้ตารางสามารถสนับสนุน Valid Time โดยจะเพิ่มคอลัมน์ WM_VALID ให้โดยอัตโนมัติ คอลัมน์นี้จะมีชนิดข้อมูล WMSYS.WM_PERIOD

```
EXECUTE DBMS_WM.EnableVersioning('names', 'VIEW_WO_OVERWRITE',
FALSE, TRUE);
```

3.8.3 เพิ่มข้อมูลพนักงานรหัส 45 ชื่อ 'Somchat' ลงในตาราง Names ตั้งแต่วันที่ 2006-01-01 ถึงวันที่ DBMS_WM.MAX_TIME (9999-12-31)

```
INSERT INTO names VALUES (
  45, 'Somchat',
  WMSYS.WM_PERIOD(TO_DATE('2006-01-01', 'YYYY-MM-DD'),
  DBMS_WM.MAX_TIME));
```

1 rows inserted

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.4 เพิ่มข้อมูลพนักงานรหัส 46 ชื่อ 'Sitti' ลงในตาราง names ตั้งแต่วันที่ 2006-01-01 ถึงวันที่ DBMS_WM.UNTIL_CHANGED (Null)

```
INSERT INTO names VALUES(
    46, 'Sitti',
    WMSYS.WM_PERIOD(TO_DATE('2006-01-01', 'YYYY-MM-DD'),
        DBMS_WM.UNTIL_CHANGED)
);
```

1 rows inserted

3.8.5 กำหนด Valid Time ตั้งแต่วันที่ DBMS_WM.MIN ถึงวันที่ DBMS_WM.MAX_TIME จากนั้นใช้คำสั่ง SELECT เพื่อดูการเพิ่มข้อมูลชื่อของพนักงาน

```
EXECUTE
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT emp_id, emp_name,
    TO_CHAR(e.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
    TO_CHAR(e.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM names e;
```

anonymous block completed

EMP_ID	EMP_NAME	FROM_DATE	TO_DATE
45	Somchat	2006-01-01	9999-12-31
46	Sitti	2006-01-01	

2 rows selected

จากการทดลองข้อ 3.8.5 นี้ จะเห็นว่า DBMS_WM.MAX_TIME จะบันทึกค่า 9999-12-31 ส่วน DBMS_WM.UNTIL_CHANGED จะบันทึกค่า Null

3.8.6 กำหนดช่วงระยะเวลา Valid Time ตั้งแต่วันที่ 2010-01-01 ถึงวันที่ DBMS_WM.UNTIL_CHANGED จากนั้นปรับปรุงข้อมูลชื่อของพนักงานรหัส 46 = 'Sittisak'

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('2010-01-01', 'YYYY-MM-DD'), DBMS_WM.UNTIL_CHANGED);
```

```
UPDATE names SET emp_name='Sittisak' WHERE emp_id=46;
```

anonymous block completed

1 rows updated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.7 กำหนดช่วงระยะเวลา Valid Time ตั้งแต่วันที่ DBMS_WM.MIN_TIME ถึงวันที่ DBMS_WM.MAX_TIME จากนั้นใช้คำสั่ง SELECT เพื่อดูการเพิ่มปรับปรุงชื่อของพนักงาน

```
EXECUTE
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT emp_id, emp_name,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM   names e;
```

anonymous block completed

EMP_ID	EMP_NAME	FROM_DATE	TO_DATE
45	Somchat	2006-01-01	9999-12-31
46	Sittisak	2010-01-01	
46	Sitti	2006-01-01	2010-01-01

3 rows selected

3.8.8 กำหนดช่วงเวลา ตั้งแต่วันที่ 2011-06-01 ถึงวันที่ DBMS_WM.UNTIL_CHANGED จากนั้นปรับปรุงข้อมูลชื่อของพนักงานรหัส 45 = 'Sompon'

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('2011-06-01','YYYY-MM-DD'),
DBMS_WM.UNTIL_CHANGED);
```

```
UPDATE names SET emp_name='Sompon' WHERE emp_id=45;
```

anonymous block completed

1 rows updated

3.8.9 กำหนด Valid Time ตั้งแต่วันที่ DBMS_WM.MIN ถึงวันที่ DBMS_WM.MAX_TIME จากนั้นใช้คำสั่ง SELECT เพื่อดูการเพิ่มปรับปรุงชื่อของพนักงาน

```
EXECUTE
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT emp_id, emp_name,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'YYYY-MM-DD') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'YYYY-MM-DD') TO_DATE
FROM   names e;
```

anonymous block completed

EMP_ID	EMP_NAME	FROM_DATE	TO_DATE
46	Sittisak	2010-01-01	
45	Sompon	2011-06-01	9999-12-31
46	Sitti	2006-01-01	2010-01-01
45	Somchat	2006-01-01	2011-06-01

4 rows selected

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.10 กำหนดค่า Valid Time ตั้งแต่วันที่ 2012-02-28 ถึงวันที่ DBMS_WM.MAX_TIME จากนั้นปรับปรุงข้อมูลชื่อของพนักงานรหัส 46 = 'Nadthasid'

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('2012-02-28','YYYY-MM-DD'),DBMS_WM.MAX_TIME);
```

```
UPDATE names SET emp_name='Nadthasid' WHERE emp_id=46;
```

anonymous block completed
1 rows updated

3.8.11 กำหนด Valid Time ตั้งแต่วันที่ DBMS_WM.MIN ถึง DBMS_WM.MAX_TIME จากนั้นใช้คำสั่ง SELECT เพื่อดูการเพิ่มปรับปรุงชื่อของพนักงาน

```
EXECUTE  
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);
```

```
SELECT emp_id, emp_name,  
        TO_CHAR(e.WM_VALID.VALIDFROM,'YYYY-MM-DD') FROM_DATE,  
        TO_CHAR(e.WM_VALID.VALIDTILL,'YYYY-MM-DD') TO_DATE  
FROM names e;
```

anonymous block completed

EMP_ID	EMP_NAME	FROM_DATE	TO_DATE
45	Sompon	2011-06-01	9999-12-31
46	Nadthasid	2012-02-28	9999-12-31
46	Sitti	2006-01-01	2010-01-01
45	Somchat	2006-01-01	2011-06-01
46	Sittisak	2010-01-01	2012-02-28

5 rows selected

3.8.12 สรุปผลการทดลอง

- 1) การใช้ค่าคงที่ DBMS_WM.MAX_TIME ระบบจะบันทึกค่า 9999-12-31
- 2) ถ้าใช้ค่า DBMS_WM.UNTIL_CHANGED ระบบจะบันทึกค่าว่าง (Null)
- 3) เมื่อมีการกำหนดช่วงเวลา SetValidTime() ระบบจะทำการเปลี่ยนค่าตำแหน่ง DBMS_WM.UNTIL_CHANGED ให้เป็นค่าเท่ากับ FROM_DATE ให้อัตโนมัติ
- 4) ระบบจะเปลี่ยนค่าตำแหน่ง TO_DATE ให้เป็นค่าว่างเช่นกัน ถ้ากำหนด TO_DATE=DBMS_WM.UNTIL_CHANGED
- 5) ถ้ากำหนด TO_DATE = DBMS_WM.MAX_TIME แล้ว ต่อมาภายหลังกำหนด TO_DATE = DBMS_WM.UNTIL_CHANGED ค่าตำแหน่ง TO_DATE จะเป็นค่า 9999-12-31 เสมอ ไม่สามารถกลับมาเป็นค่าว่าง (Null) ได้อีกต่อไป

3.9 สรุปผลการทดลองคำสั่งจัดการข้อมูลแบบเชิงเวลา

จากการทดลองคำสั่งจัดการข้อมูลแบบเชิงเวลา โดยใช้โปรแกรมบริหารจัดการฐานข้อมูล ออราเคิลสืบเอ็ดจี (Oracle 11g) สามารถปฏิบัติคำสั่งด้านเชิงเวลา ทั้งการเพิ่มข้อมูล (Insert) การปรับปรุงข้อมูล (Update) การลบข้อมูล (Delete) และการสอบถามข้อมูล (Select) ได้ถูกต้องตามคุณสมบัติเชิงเวลา ดังนั้นออราเคิลสืบเอ็ดจี (Oracle 11g) สามารถใช้เป็นเครื่องมือในบันทึกข้อมูล และการพัฒนาระบบสารสนเทศเชิงเวลาได้

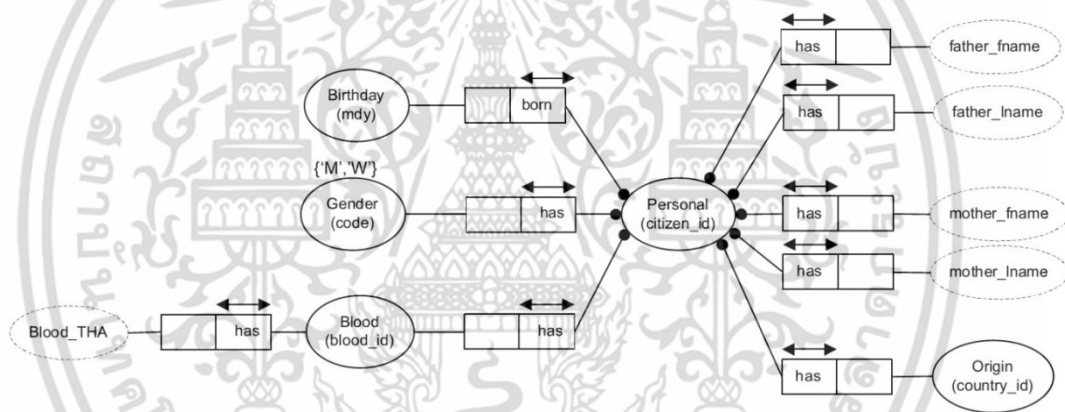


บทที่ 4

การออกแบบระบบสารสนเทศเชิงเวลา

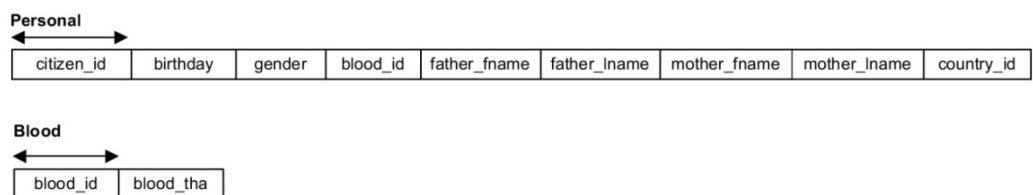
การทำวิทยานิพนธ์ฉบับนี้ ได้ใช้ระบบงานบุคลากรเป็นกรณีศึกษา ในการออกแบบและพัฒนาระบบสารสนเทศเชิงเวลา โดยใช้แบบจำลองที่โออาร์เอ็ม (TORM/NIAM) เป็นแบบจำลองในการออกแบบฐานข้อมูล ซึ่งเป็นวิธีการที่ดีวิธีหนึ่ง เพราะเป็นวิธีที่สามารถแสดงความหมายความสัมพันธ์ และข้อจำกัดต่างๆ ของข้อมูล และเป็นวิธีที่มีอัลกอริทึมที่สามารถออกแบบฐานข้อมูลแบบเชิงสัมพันธ์ (Relation Database) ที่อยู่ในรูปของนอร์มอลฟอร์มระดับที่ 5 (5NF) [8] การออกแบบระบบงานบุคลากรแบบเชิงเวลานี้ จะออกแบบทั้งหมด 5 ส่วน คือ ประวัติข้อมูลเฉพาะบุคคล ประวัติชื่อ-นามสกุล ประวัติที่อยู่ ประวัติการศึกษา และประวัติการทำงาน ดังนี้

4.1 การออกแบบ ORM ประวัติข้อมูลเฉพาะบุคคล



รูปที่ 4.1 ORM ประวัติข้อมูลเฉพาะบุคคล

รูปที่ 4.1 เป็นแบบจำลองโออาร์เอ็ม (ORM) แสดงประวัติข้อมูลบุคคล ที่เป็นข้อมูลเฉพาะไม่สามารถเปลี่ยนแปลงได้ตลอดชีวิต ประกอบด้วย ว.ด.ป.เกิด หมู่อโลहित ชื่อ-นามสกุลของบิดา ชื่อ-นามสกุลของมารดา เชื้อชาติ และเพศ ถึงแม้ปัจจุบัน บางประเทศจะมีกฎหมายรองรับการเปลี่ยนแปลงเพศแล้วก็ตาม แต่สำหรับวิทยานิพนธ์ฉบับนี้ จะสนใจเพศเดิมที่ระบุตอนเกิดครั้งแรก และจากรูปสามารถแปลงเป็นโครงร่างตารางแบบเชิงสัมพันธ์ (Relation Schema) ดังรูปที่ 4.2

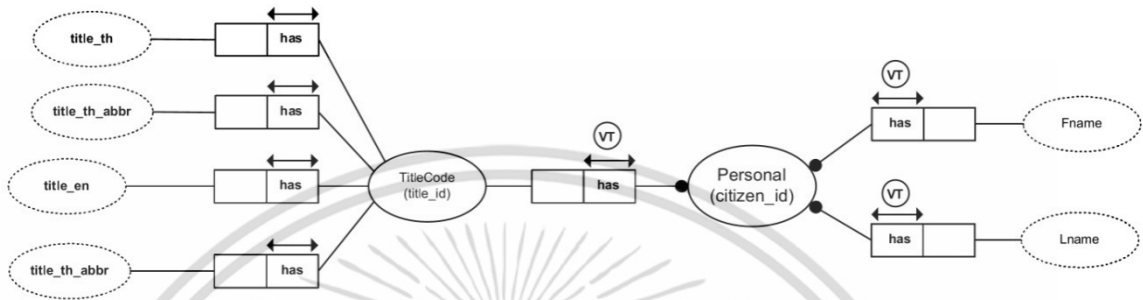


รูปที่ 4.2 โครงร่างตารางประวัติข้อมูลเฉพาะบุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการวิจัยในเพื่อการศึกษาเท่านั้น ขอสงวนสิทธิ์ในไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

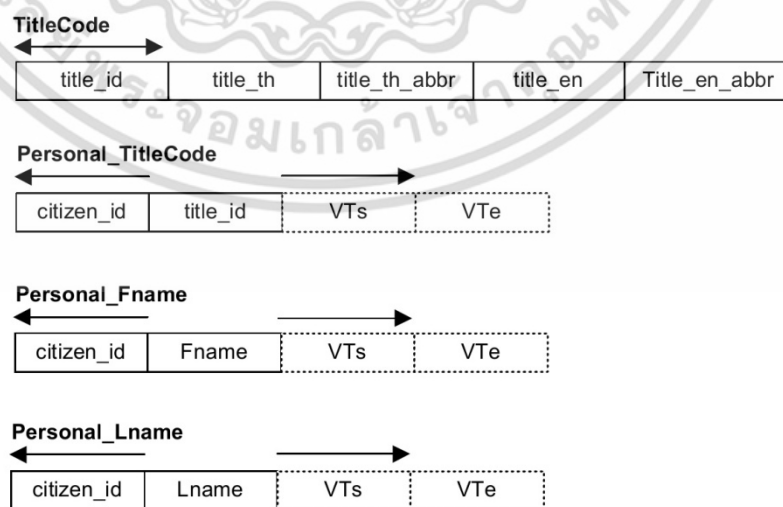
4.2 การออกแบบ **TORM** ประวัติชื่อ-นามสกุล

รูปที่ 4.3 เป็นแบบจำลองที่โออาร์เอ็ม (TORM) แสดงความสัมพันธ์แบบเชิงเวลาของประวัติชื่อ-นามสกุล จะเห็นว่ามี 2 เอนทิตี (Entity) คือ **Personal** และ **TitleCode** โดยเอนทิตี **TitleCode** มีความสัมพันธ์ แบบ **A Fact Type** กับเอนทิตี **Personal**



รูปที่ 4.3 TORM ประวัติชื่อ-นามสกุล

จากรูปที่ 4.3 ปรากฏสัญลักษณ์ VT อยู่ 3 ตำแหน่ง ซึ่งแสดงให้เห็นความสัมพันธ์แบบเชิงเวลา ดังนี้ 1) เอนทิตี **Personal** และเอนทิตี **TitleCode** มีความสัมพันธ์เชิงเวลาแบบข้อเท็จจริง (Temporal Fact Type) 2) เอนทิตี **Personal** และลาเบล **Fname** มีความสัมพันธ์เชิงเวลาอ้างอิง (Temporal Reference Type) และ 3) เอนทิตี **Personal** และลาเบล **Lname** มีความสัมพันธ์แบบเชิงเวลา ดังนั้น สามารถบันทึกเพิ่มและปรับปรุงประวัติการเปลี่ยนแปลง คำนำนาม ชื่อ และนามสกุล โดยแปลงเป็นโครงร่างตารางดังรูปที่ 4.3



รูปที่ 4.4 โครงร่างตารางประวัติชื่อ-นามสกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.4 ประกอบด้วย 5 โครงร่างตาราง ดังนี้

TitleCode เป็นโครงร่างตารางแบบเชิงสัมพันธ์ (Relation Table) มีคีย์หลัก (Primary Key : PK) คือ **Title_id** และมีแอทริบิวต์ 5 แอทริบิวต์ ประกอบด้วย รหัสคำนำหน้านาม (**Title_ID**) คำนำหน้านามภาษาไทย (**Title_th**) คำนำหน้านามภาษาไทยแบบอักษรย่อ (**Title_th_abbrev**) คำนำหน้านามภาษาอังกฤษ (**Title_en**) และคำนำหน้านามภาษาอังกฤษแบบอักษรย่อ (**Title_en_abbrev**)

Personal_TitleCode **Personal_Fname** และ **Persona_Lname** เป็นโครงร่างตารางเชิงเวลาแบบ Valid Time State Table ทั้ง 3 โครงสร้างตาราง จะมีคีย์หลัก (PK) คือ **Citizen_ID** ร่วมกับ VTs โดยแอทริบิวต์ VTs และ VTe มีขอบเป็นเส้นประ เพื่อแสดงให้ทราบว่าไม่ต้องกำหนดคอลัมน์นี้ขณะที่ใช้คำสั่งสร้างตาราง ทั้งนี้ Oracle11g จะสร้างให้อัตโนมัติ เมื่อใช้คำสั่ง **EnableVersioning**

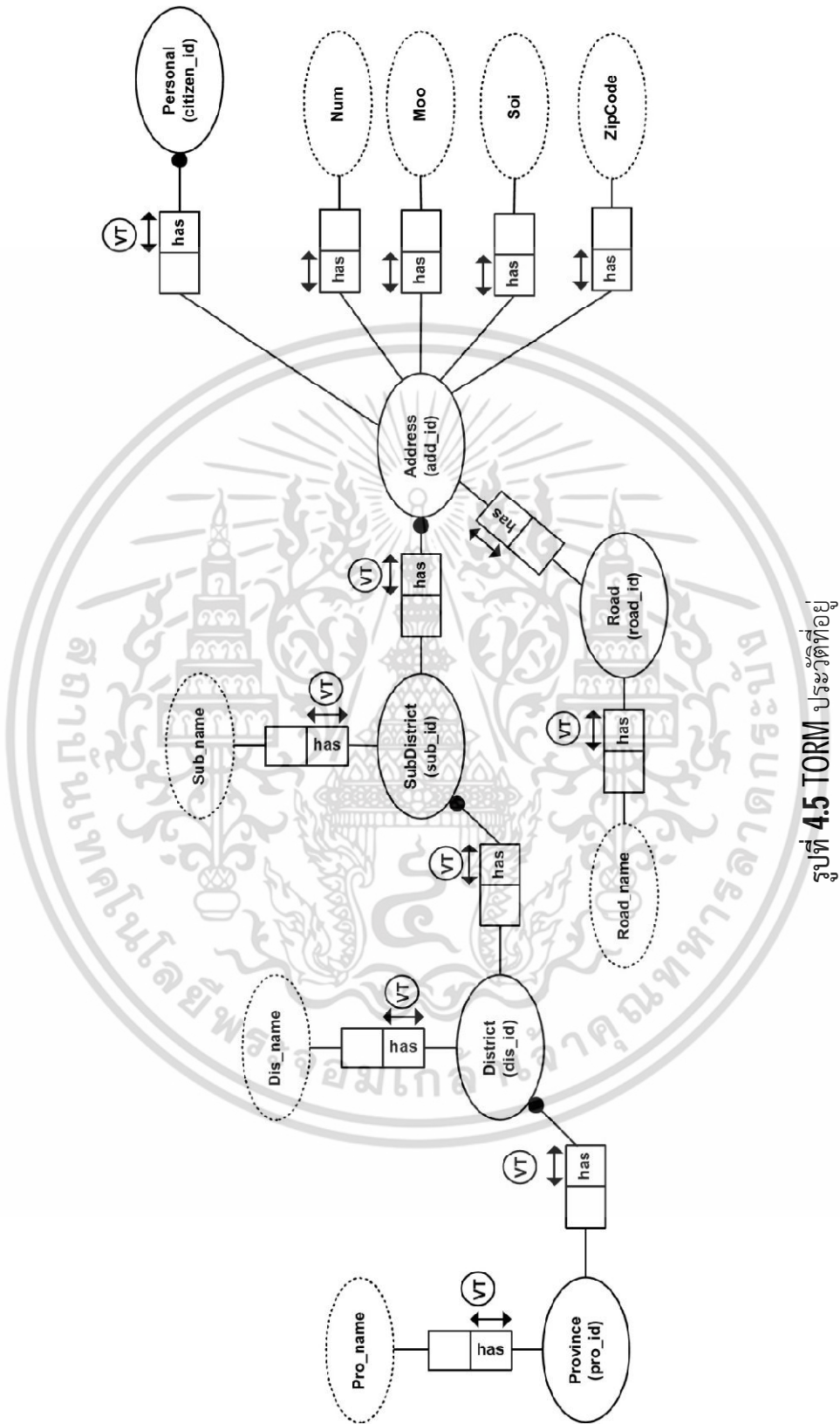
Personal_TitleCode เป็นโครงร่างตารางแบบเชิงเวลา บันทึกการเปลี่ยนแปลงคำนำหน้านาม

Personal_Fname เป็นโครงร่างตารางแบบเชิงเวลา บันทึกการเปลี่ยนแปลงชื่อ

Persona_Lname เป็นโครงร่างตารางแบบเชิงเวลา บันทึกการเปลี่ยนแปลงนามสกุล

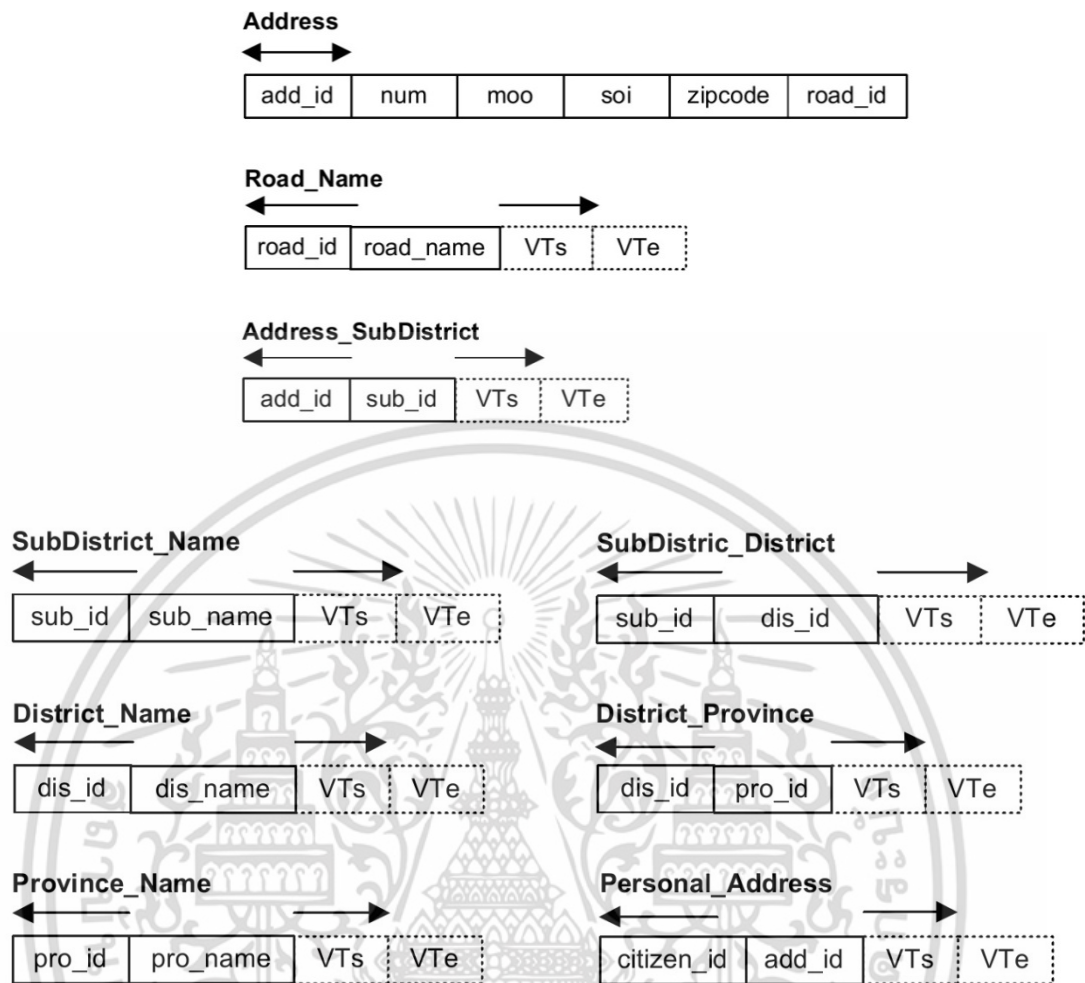
4.3 การออกแบบ **TORM** ประวัติที่อยู่

รูปที่ 4.5 แสดงแบบจำลองทีโออาร์เอ็ม (TORM) ของประวัติที่อยู่ ประกอบด้วย 6 แอทริบิวต์ คือ ข้อมูลบุคคล (**Personal**) ที่อยู่ (**Address**) ถนน (**Road**) ตำบล (**SubDistrict**) อำเภอ (**District**) และ จังหวัด (**Province**) ทั้งนี้มี VT ปรากฏอยู่ 8 ตำแหน่ง สามารถแปลงเป็นโครงร่างตารางดังรูปที่ 4.6



รูปที่ 4.5 TORM ประวัติที่อยู่

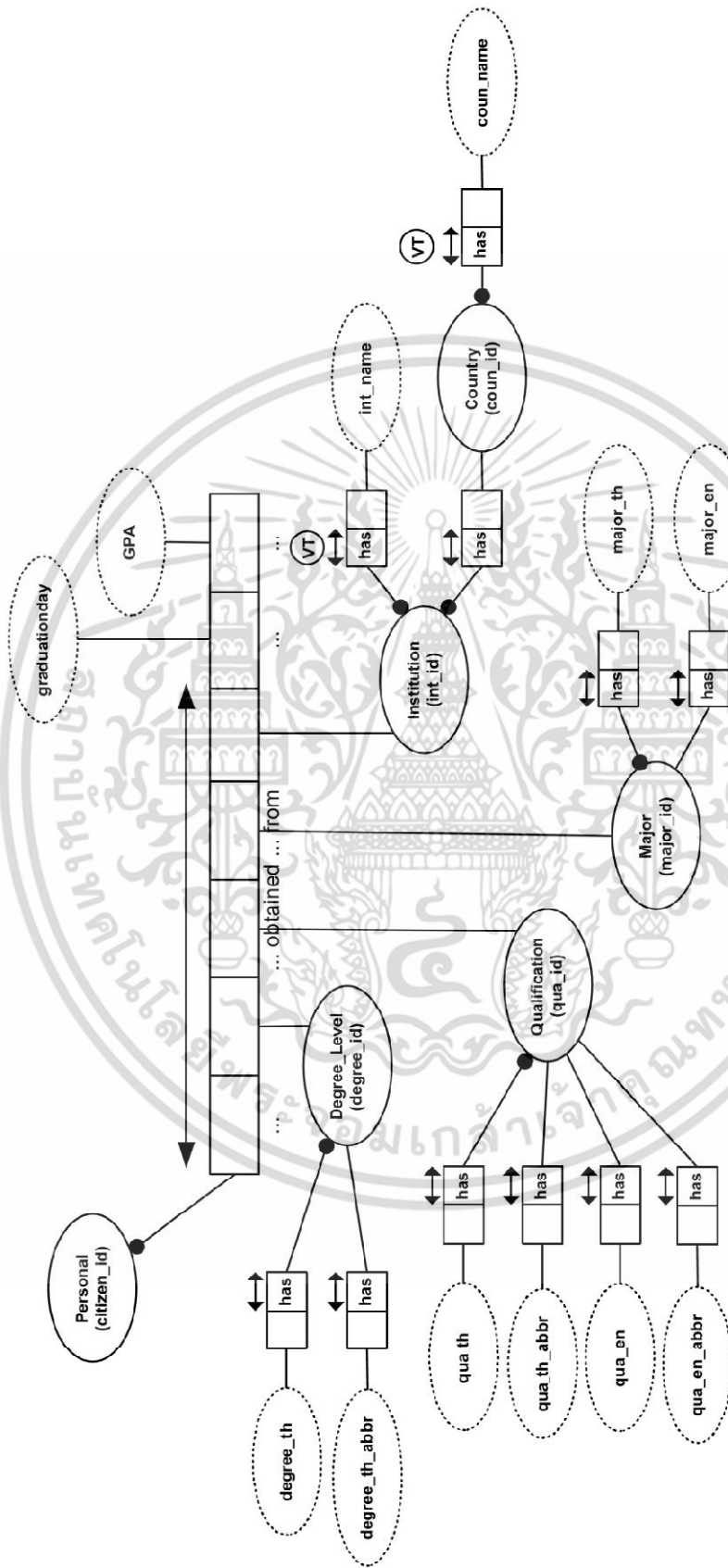
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 โครงร่างตารางประวัติที่อยู่

จากรูปที่ 4.6 จะประกอบด้วยโครงร่างตารางแบบเชิงสัมพันธ์ 1 โครงร่าง คือ Address สำหรับบันทึกประวัติที่อยู่ของบุคลากร และโครงร่างตารางแบบเชิงเวลา 8 โครงร่าง คือ Road_Name Address_SubDistrict Subdistrict_Name SubDistrict_District District_Name District_Province Province_Name และ Personal_Address

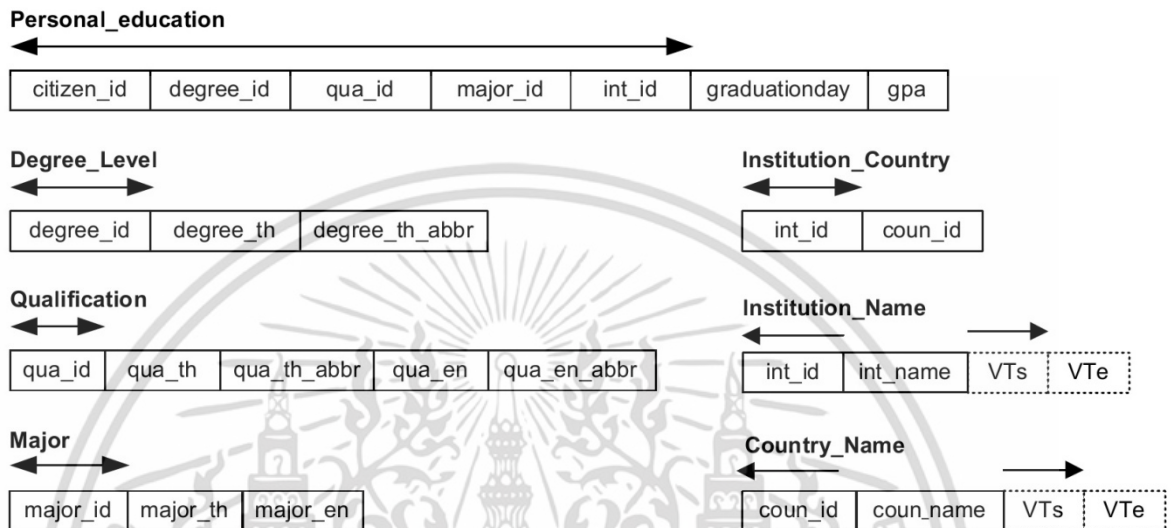
4.4 การออกแบบ **TORM** ประวัติการศึกษา



รูปที่ 4.7 TORM ประวัติการศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.7 แสดงแบบจำลองทีโออาร์เอ็ม (ORM) ของประวัติการศึกษา ประกอบด้วย 6 เอนทิตี คือ ข้อมูลบุคคล (Personal) ระดับการศึกษา (Degree) วุฒิการศึกษา (Qualification) สาขาวิชาเอก (Major) สถาบันการศึกษา (Institution) และประเทศ (Country) สามารถแปลงเป็นโครงสร้างตารางได้ดังรูปที่ 4.8

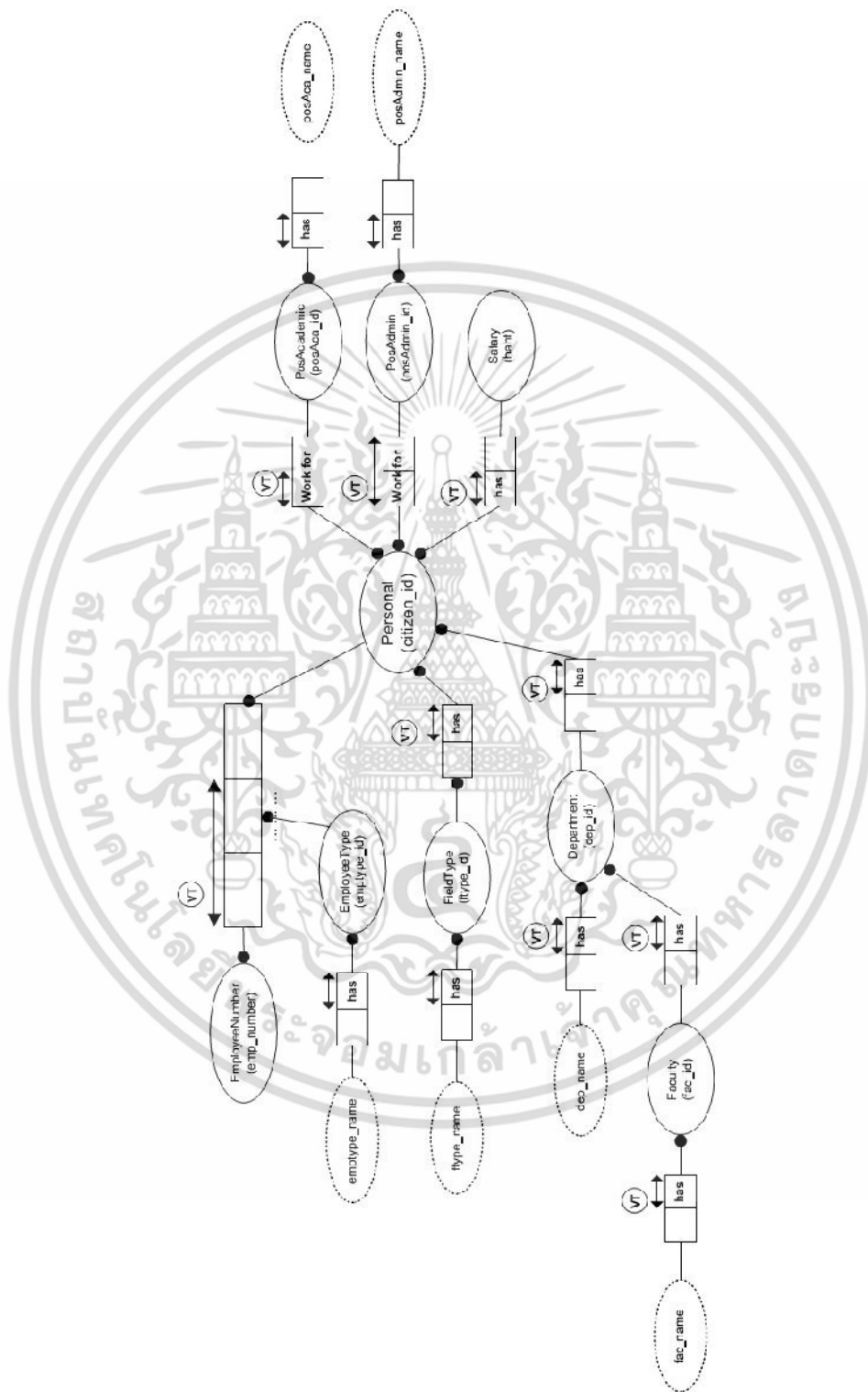


รูปที่ 4.8 โครงสร้างตารางประวัติการศึกษา

จากรูปที่ 4.8 แสดงโครงสร้างตารางของประวัติการศึกษา ประกอบด้วยโครงสร้างตารางแบบเชิงสัมพันธ์ 5 โครงสร้างตาราง คือ Personal_education Degree Qualification Major Institution และโครงสร้างตารางแบบเชิงเวลา 2 โครงสร้างตาราง คือ Institution_Name Country_Name

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

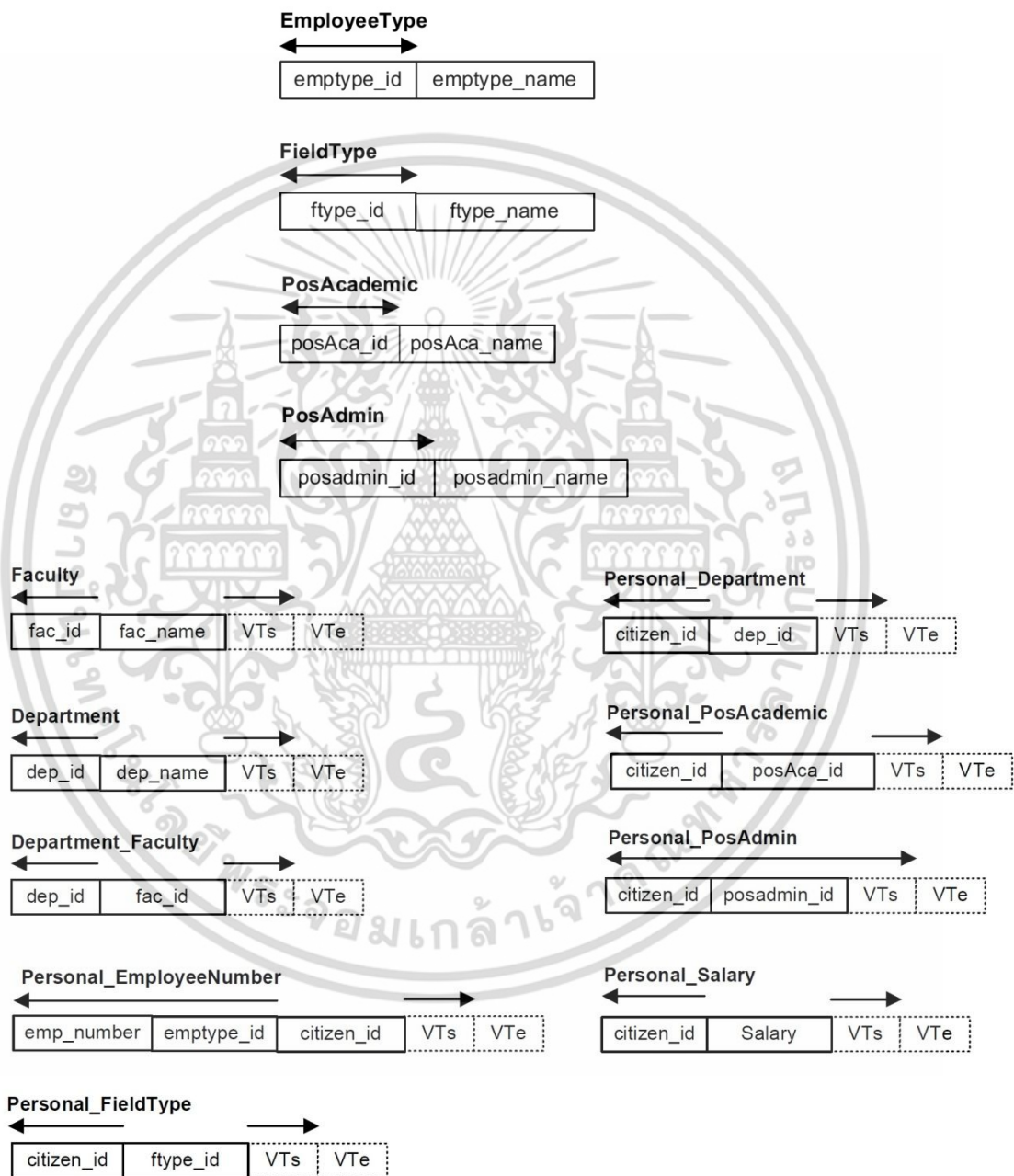
4.5 การออกแบบ **TORM** ประวัติการทำงาน



รูปที่ 4.9 TORM ประวัติการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.9 แสดงแบบจำลองที่โออาร์เอ็ม (TORM) ของประวัติการทำงาน ประกอบด้วย 9 เอนทิตี คือ ข้อมูลบุคคล (Personal) เลขอัตราตำแหน่ง (EmployeeNumber) ประเภทตำแหน่ง (EmployeeType) สายงานการทำงาน (FieldType) แผนก (Department) คณะ (Faculty) ตำแหน่งทางวิชาการ (PosAcademic) ตำแหน่งทางบริหาร (PosAdmin) และเงินเดือน (Salary) สามารถแปลงเป็นโครงสร้างตารางได้ดังรูปที่ 4.10



รูปที่ 4.10 โครงสร้างตารางประวัติการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 พจนานุกรมข้อมูล ของระบบงานบุคลากรแบบเชิงเวลา (Data Dictionary)

4.6.1 ตารางระบบงานบุคลากรแบบเชิงเวลา

ระบบงานบุคลากรแบบเชิงเวลา โดยใช้แบบจำลองที่โออาร์เอ็ม (Temporal ORM) ออกแบบสามารถแปลงเป็นโครงร่างตารางได้ทั้งหมด 35 ตาราง เป็นตารางแบบเชิงสัมพันธ์ 13 ตาราง และตารางแบบเชิงเวลา 22 ตาราง

ตารางที่ 4.1 ระบบงานบุคลากรแบบเชิงสัมพันธ์ (Relation Table)

ลำดับ	ชื่อตาราง	ประเภท	คำอธิบาย
1	PERSONAL	แบบสัมพันธ์	บันทึกประวัติบุคคล
2	BLOOD	แบบสัมพันธ์	บันทึกหมู่โลหิต
3	TITLECODE	แบบสัมพันธ์	บันทึกรหัสค่านำหน้านาม
4	ADDRESS	แบบสัมพันธ์	บันทึกที่อยู่
5	PERSONAL_EDUCATION	แบบสัมพันธ์	บันทึกประวัติการศึกษา
6	DEGREE_LEVEL	แบบสัมพันธ์	บันทึกระดับการศึกษา
7	QUALIFICATION	แบบสัมพันธ์	บันทึกวุฒิการศึกษา
8	MAJOR	แบบสัมพันธ์	บันทึกสาขาวิชา
9	INSTITUTION_COUNTRY	แบบสัมพันธ์	บันทึกสถาบัน สังกัดประเทศ
10	EMPLOYEE_TYPE	แบบสัมพันธ์	บันทึกประเภทบุคลากร
11	FIELDTYPE	แบบสัมพันธ์	บันทึกสายการทำงาน
12	POSADMIN	แบบสัมพันธ์	บันทึกชื่อตำแหน่งทางบริหาร
13	POSACADEMIC	แบบสัมพันธ์	บันทึกชื่อตำแหน่งทางวิชาการ

ตารางที่ 4.2 ระบบงานบุคลากรแบบเชิงเวลา (Temporal Table)

ลำดับ	ชื่อตาราง	ประเภท	คำอธิบาย
1	PERSONAL_TITLECODE	แบบเชิงเวลา	บันทึกค่านำหน้านามของบุคคล
2	PERSONAL_FNAME	แบบเชิงเวลา	บันทึกชื่อของบุคคล
3	PERSONAL_LNAME	แบบเชิงเวลา	บันทึกนามสกุลของบุคคล
4	ROAD_NAME	แบบเชิงเวลา	บันทึกชื่อถนน
5	ADDRESS_SUBDISTRICT	แบบเชิงเวลา	บันทึกการสังกัดเลขรหัสบ้าน เข้ากับตำบล
6	SUBDISTRICT_NAME	แบบเชิงเวลา	บันทึกชื่อตำบล
7	SUBDISTRICT_DISTRICT	แบบเชิงเวลา	บันทึกการสังกัดของตำบล เข้ากับอำเภอ
8	DISTRICT_NAME	แบบเชิงเวลา	บันทึกชื่ออำเภอ
9	DISTRICT_PROVINCE	แบบเชิงเวลา	บันทึกการสังกัดของอำเภอ เข้ากับจังหวัด
10	PROVINCE_NAME	แบบเชิงเวลา	บันทึกชื่อจังหวัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ระบบงานบุคคลากรแบบเชิงเวลา (Temporal Table) (ต่อ)

ลำดับ	ชื่อตาราง	ประเภท	คำอธิบาย
11	PERSONAL_ADDRESS	แบบเชิงเวลา	บันทึกการสังกัดของบุคคล เข้ากับรหัสบ้าน
12	INSTITUTION_NAME	แบบเชิงเวลา	บันทึกชื่อสถาบันการศึกษา
13	COUNTRY_NAME	แบบเชิงเวลา	บันทึกชื่อประเทศ
14	FACULTY	แบบเชิงเวลา	บันทึกชื่อคณะ
15	DEPARTMENT	แบบเชิงเวลา	บันทึกชื่อแผนก
16	DEPARTMENT_FACULTY	แบบเชิงเวลา	บันทึกการสังกัดแผนก เข้ากับคณะ
17	PERSONAL_EMPLOYEENUMBER	แบบเชิงเวลา	บันทึกเลขอัตรา
18	PERSONAL_FIELDTYPE	แบบเชิงเวลา	บันทึกบุคลากร เข้ากับประเภทสายการ ทำงาน
19	PERSONAL_DEPARTMENT	แบบเชิงเวลา	บันทึกประเภทสายการทำงาน ของบุคลากร
20	PERSONAL_POSACADEMIC	แบบเชิงเวลา	บันทึกประวัติตำแหน่งทางวิชาการ ของ บุคลากร
21	PERSONAL_POSADMIN	แบบเชิงเวลา	บันทึกประวัติตำแหน่งทางบริหาร ของ บุคลากร
22	PERSONAL_SALARY	แบบเชิงเวลา	บันทึกประวัติเงินเดือน ของบุคลากร

4.6.2 พจนานุกรมข้อมูลของตารางแบบสัมพันธ์ (Relation Data Dictionary)

ตารางที่ 4.3 บันทึกประวัติบุคคล (PERSONAL)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	CITIZEN_ID	VARCHAR	13	PK		เลขประจำตัวประชาชน
2	BIRTHDAY	DATE				ว.ด.ป.เกิด
3	GENDER	VARCHAR	1			เพศ M=ชาย W=หญิง
4	BLOOD_ID	VARCHAR	2	FK	BLOOD	หมู่โลหิต A, B, AB, O
5	FATHER_FNAME	VARCHAR	30			ชื่อบิดา
6	FATHER_LNAME	VARCHAR	30			นามสกุลบิดา
7	MOTHER_FNAME	VARCHAR	30			ชื่อมารดา
8	MOTHER	VARCHAR	30			นามสกุลมารดา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 บันทึกหมู่โลหิต (BLOOD)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	BLOOD_ID	VARCHAR	2	PK		รหัสหมู่โลหิต
2	BLOOD_TH	VARCHAR	10			ชื่อหมู่โลหิต

ตารางที่ 4.5 บันทึกรหัสค่านำหน้านาม (TITLECODE)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	TITLE_ID	NUMBER		PK		เลขรหัสค่านำหน้านาม
2	TITLE_TH	VARCHAR	100			ค่านำหน้านามภาษาไทย
3	TITLE_TH_ABBR	VARCHAR	30			ค่านำหน้านามภาษาไทย (อักษรย่อ)
4	TITLE_EN	VARCHAR	100			ค่านำหน้านามภาษาอังกฤษ
5	TITLE_EN_ABBR	VARCHAR	30			ค่านำหน้านามภาษาอังกฤษ (อักษรย่อ)

ตารางที่ 4.6 บันทึกที่อยู่ (ADDRESS)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	ADD_ID	NUMBER		PK		เลขรหัสประจำบ้าน
2	NUM	VARCHAR	13			เลขที่บ้าน
3	MOO	VARCHAR	30			หมู่ที่
4	SOI	VARCHAR	50			ชื่อซอย
5	ZIPCODE	VARCHAR	5			เลขรหัสไปรษณีย์
6	ROAD_ID	NUMBER		FK	ROAD_NAME	เลขรหัสถนน

ตารางที่ 4.7 บันทึกประวัติการศึกษา (PERSONAL_EDUCATION)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	CITIZEN_ID	VARCHAR	13	PK		เลขประจำตัวประชาชน
2	DEGREE_ID	NUMBER		PK	DEGREE_LEVEL	เลขรหัสระดับการศึกษา
3	GUA_ID	NUMBER		PK	QUALIFICATION	เลขรหัสวุฒิการศึกษา
4	MAJOR_ID	NUMBER		PK	MAJOR	เลขรหัสสาขาวิชา
5	INT_ID	NUMBER		PK	INSTITUTION	เลขรหัสสถาบันการศึกษา
6	GRADUATION	DATE				วันสำเร็จการศึกษา
7	GPA	NUMBER				GPA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.8 บันทึกระดับการศึกษา (DEGREE_LEVEL)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	DEGREE_ID	NUMBER		PK		เลขรหัสระดับการศึกษา
2	DEGREE_TH	VACHAR	50			ชื่อระดับการศึกษา
3	DEGREE_TH_ABBR	VACHAR	20			ชื่อระดับการศึกษา (อักษรย่อ)

ตารางที่ 4.9 บันทึกวุฒิการศึกษา (QUALIFICATION)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	GUA_ID	NUMBER		PK		เลขรหัสวุฒิการศึกษา
2	GUA_TH	VACHAR	50			ชื่อวุฒิการศึกษาภาษาไทย
3	GUA_TH_ABBR	VACHAR	20			ชื่อวุฒิการศึกษาภาษาไทย(อักษรย่อ)
4	GUA_EN	VACHAR	50			ชื่อวุฒิการศึกษาภาษาอังกฤษ
5	GUA_EN_ABBR	VACHAR	20			ชื่อวุฒิการศึกษาภาษาอังกฤษ(อักษรย่อ)

ตารางที่ 4.10 บันทึกสาขาวิชา (MAJOR)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	MAJOR_ID	NUMBER		PK		เลขรหัสสาขาวิชา
2	MAJOR_TH	VACHAR	50			ชื่อสาขาวิชาภาษาไทย
3	MAJOR_EN	VACHAR	50			ชื่อสาขาวิชาภาษาอังกฤษ

ตารางที่ 4.11 บันทึกสถาบัน สังกัดประเทศ (INSTITUTION_COUNTRY)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	INT_ID	NUMBER		PK		เลขรหัสสถาบันการศึกษา
2	COUN_ID	NUMBER		PK	COUNTRY_NAME	เลขรหัสประเทศ

ตารางที่ 4.12 บันทึกประเภทบุคลากร (EMPLOYEE_TYPE)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	EMPTY_ID	NUMBER		PK		เลขรหัสประเภทบุคลากร
2	EMPTY_NAME	VACHAR	30			ชื่อประเภทบุคลากร

ตารางที่ 4.13 บันทึกสายการทำงาน (FIELDTYPE)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	FTYPE_ID	NUMBER		PK		เลขรหัสสายการทำงาน
2	FTYPE_NAME	VACHAR	30			ชื่อสายการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.14 บันทึกชื่อตำแหน่งทางบริหาร (POSADMIN)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	POSADMIN_ID	NUMBER		PK		เลขรหัสชื่อตำแหน่งทางบริหาร
2	POSADMIN_NAME	VARCHAR	50			ชื่อตำแหน่งทางบริหาร

ตารางที่ 4.15 บันทึกชื่อตำแหน่งทางวิชาการ (POSACADEMIC)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	POSACA_ID	NUMBER		PK		เลขรหัสชื่อตำแหน่งวิชาการ
2	POSACA_NAME	VARCHAR	50			ชื่อตำแหน่งวิชาการ

4.6.3 พจนานุกรมข้อมูลของตารางแบบเชิงเวลา (Temporal Data Dictionary)

ตารางที่ 4.16 บันทึกคำนำหน้านามของบุคคล (PERSONAL_TITLECODE)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	CITIZEN_ID	VARCHAR	13	PK		เลขประจำตัวประชาชน
2	TITLE_ID	NUMBER		FK	TITLECODE	รหัสคำนำหน้านาม
3*	WM_VALID	WM_PERIOD				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

ตารางที่ 4.17 บันทึกชื่อของบุคคล (PERSONAL_FNAME)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	CITIZEN_ID	VARCHAR	13	PK		เลขประจำตัวประชาชน
2	FNAME	VARCHAR	30			ชื่อ
3*	WM_VALID	WM_PERIOD				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

ตารางที่ 4.18 บันทึกนามสกุลของบุคคล (PERSONAL_LNAME)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	CITIZEN_ID	VARCHAR	13	PK		เลขประจำตัวประชาชน
2	LNAME	VARCHAR	30			นามสกุล
3*	WM_VALID	WM_PERIOD				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

หมายเหตุ คอลัมน์ที่มี * ไม่ต้องกำหนดตอนสร้างตาราง เพราะ DBMS จะสร้างให้อัตโนมัติ เมื่อใช้

คำสั่ง `EnableVersioning`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.19 บันทึกชื่อถนน (ROAD_NAME)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	ROAD_ID	NUMBER		PK		เลขรหัสถนน
2	ROAD_NAME	VARCHAR	50			ชื่อถนน
3*	WM_VALID	WM_PERIOD				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

ตารางที่ 4.20 บันทึกการสังกัดเลขรหัสบ้าน เข้ากับตำบล (ADDRESS_SUBDISTRICT)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	ADD_ID	NUMBER		PK		เลขรหัสประจำบ้าน
2	SUB_ID	NUMBER		FK	SUBDISTRICT	เลขรหัสตำบล
3*	WM_VALID	WM_PERIOD				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

ตารางที่ 4.21 บันทึกชื่อตำบล (SUBDISTRICT_NAME)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	SUB_ID	NUMBER		PK		เลขรหัสตำบล
2	SUB_NAME	VARCHAR	30			ชื่อตำบล
3*	WM_VALID	WM_PERIOD				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

ตารางที่ 4.22 บันทึกการสังกัดของตำบล เข้ากับอำเภอ (SUBDISTRICT_DISTRICT)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	SUB_ID	NUMBER		PK		เลขรหัสตำบล
2	DIS_ID	NUMBER		FK	DISTRICT	เลขรหัสอำเภอ
3*	WM_VALID	WM_PERIOD				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

ตารางที่ 4.23 บันทึกชื่ออำเภอ (DISTRICT_NAME)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	DIS_ID	NUMBER		PK		เลขรหัสอำเภอ
2	DIS_NAME	VARCHAR				ชื่ออำเภอ
3*	WM_VALID	WM_PERIOD				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.24 บันทึกการสังกัดของอำเภอ เข้ากับจังหวัด (DISTRICT_PROVINCE)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	DIS_ID	NUMBER		PK		เลขรหัสอำเภอ
2	PRO_ID	NUMBER		FK	PROVINCE	เลขรหัสจังหวัด
3*	WM_VALID	WM_PERIED				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

ตารางที่ 4.25 บันทึกชื่อจังหวัด (PROVINCE_NAME)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	PRO_ID	NUMBER		PK		เลขรหัสจังหวัด
2	PRO_NAME	VACHAR	30			ชื่อจังหวัด
3*	WM_VALID	WM_PERIED				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

ตารางที่ 4.26 บันทึกการสังกัดของบุคคล เข้ากับรหัสบ้าน (PERSONAL_ADDRESS)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	CITIZEN_ID	VACHAR	13	PK		เลขประจำตัวประชาชน
2	ADD_ID	NUMBER		FK	ADDRESS	เลขรหัสประจำบ้าน
3*	WM_VALID	WM_PERIED				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

ตารางที่ 4.27 บันทึกชื่อสถาบันการศึกษา (INSTITUTION_NAME)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	INT_ID	NUMBER		PK		เลขรหัสสถาบันการศึกษา
2	INT_NAME	VACHAR				ชื่อสถาบันการศึกษา
3*	WM_VALID	WM_PERIED				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

ตารางที่ 4.28 บันทึกชื่อประเทศ (COUNTRY_NAME)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	COUN_ID	NUMBER		PK		เลขรหัสประเทศ
2	COUN_NAME	VACHAR				ชื่อประเทศ
3*	WM_VALID	WM_PERIED				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.29 บันทึกชื่อคณะ (FACULTY)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	FAC_ID	NUMBER		PK		เลขรหัสคณะ
2	FAC_NAME	VACHAR	50			ชื่อคณะ
3*	WM_VALID	WM_PERIED				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

ตารางที่ 4.30 บันทึกชื่อแผนก (DEPARTMENT)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	DEP_ID	NUMBER		PK		เลขรหัสแผนก
2	DEP_NAME	VACHAR	50			ชื่อแผนก
3*	WM_VALID	WM_PERIED				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

ตารางที่ 4.31 บันทึกการสังกัดแผนก เข้ากับคณะ (DEPARTMENT_FACULTY)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	DEP_ID	NUMBER		PK		เลขรหัสแผนก
2	FAC_ID	NUMBER		FK	FACULTY	เลขรหัสคณะ
3*	WM_VALID	WM_PERIED				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

ตารางที่ 4.32 บันทึกเลขอัตรา (PERSONAL_EMPLOYEEENUMBER)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	EMP_NUMBER	NUMBER		PK		เลขอัตราตำแหน่ง
2	EMPTYE_ID	NUMBER		PK	EMPLOYEEEYPE	เลขรหัสประเภทบุคลากร
3	CITIZEN_ID	VACHAR	13			เลขประจำตัวประชาชน
4*	WM_VALID	WM_PERIED				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

ตารางที่ 4.33 บันทึกบุคลากร เข้ากับประเภทสายการทำงาน (PERSONAL_FIELDTYPE)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	CITIZEN_ID	VACHAR	13	PK		เลขประจำตัวประชาชน
2	FTYPE_ID	NUMBER		FK	FIELDTYPE	เลขรหัสสายการทำงาน
3*	WM_VALID	WM_PERIED				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.34 บันทึกประเภทสายการทำงาน ของบุคลากร (PERSONAL_DEPARTMENT)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	CITIZEN_ID	VARCHAR	13	PK		เลขประจำตัวประชาชน
2	DEP_ID	NUMBER		FK	DEPARTMENT	เลขรหัสแผนก
3*	WM_VALID	WM_PERIED				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

ตารางที่ 4.35 บันทึกประวัติตำแหน่งทางวิชาการ ของบุคลากร (PERSONAL_POSACADEMIC)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	CITIZEN_ID	VARCHAR	13	PK		เลขประจำตัวประชาชน
2	POSACA_ID	NUMBER		FK	POSACADEMIC	เลขรหัสชื่อตำแหน่งวิชาการ
3*	WM_VALID	WM_PERIED				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

ตารางที่ 4.36 บันทึกประวัติตำแหน่งทางบริหาร ของบุคลากร (PERSONAL_POSADMIN)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	CITIZEN_ID	VARCHAR	13	PK		เลขประจำตัวประชาชน
2	POSADMIN_ID	NUMBER		FK	POSADMIN	เลขรหัสชื่อตำแหน่งทางบริหาร
3*	WM_VALID	WM_PERIED				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

ตารางที่ 4.37 บันทึกประวัติเงินเดือน ของบุคลากร (PERSONAL_SALARY)

คอลัมน์	ชื่อฟิลด์	ประเภท	ขนาด	คีย์	อ้างอิง	คำอธิบาย
1	CITIZEN_ID	VARCHAR	13	PK		เลขประจำตัวประชาชน
2	SALARY	NUMBER				เงินเดือน
3*	WM_VALID	WM_PERIED				เวลาเริ่มต้น (VALIDFROM) เวลาสิ้นสุด (VALIDTILL)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การสอบถามข้อสนเทศเชิงเวลา

ในบทนี้ จะทำการทดสอบการสร้างตาราง ตามที่ได้ออกแบบไว้ในบทที่ 4 และทำการทดลองบันทึกข้อมูลเชิงเวลา 4 ส่วนคือ ประวัติชื่อ-นามสกุล ประวัติที่อยู่ ประวัติการศึกษา ประวัติการทำงาน จากนั้นทำการสอบถามข้อสนเทศเชิงเวลาในด้านต่างๆ โดยปฏิบัติดังต่อไปนี้

5.1 การสร้างตารางบันทึกข้อมูล

หลักจากที่ได้ออกแบบระบบสารสนเทศงานบุคลากรแบบเชิงเวลา โดยใช้แบบจำลองที่โออาร์เอ็ม (TORM) และแปลงเป็นโครงสร้างตารางทั้งหมด 35 ตาราง แบ่งเป็นโครงสร้างตารางแบบเชิงสัมพันธ์ 13 ตาราง และโครงสร้างตารางแบบเชิงเวลา 22 ตาราง ต่อไปนี้จะใช้คำสั่งของออรากิลส์ (Oracle11g) ทำการสร้างตารางทั้ง 35 ตาราง เพื่อทำการทดสอบในส่วนต่างๆ ต่อไปนี้

5.1.1 การสร้างตารางแบบเชิงสัมพันธ์ 13 ตาราง

--1)create personal table

```
CREATE TABLE personal(  
  citizen_id          VARCHAR(13) PRIMARY KEY,  
  birthday            DATE,  
  gender              VARCHAR(1),  
  blood               VARCHAR(2),  
  father_fname       VARCHAR(30),  
  father_lname       VARCHAR(30),  
  mother_fname       VARCHAR(30),  
  mother_lname       VARCHAR(30));
```

--2)create blood table

```
CREATE TABLE blood(  
  blood_id           VARCHAR(2) PRIMARY KEY,  
  blood_th           VARCHAR(10));
```

--3)create titlecode table

```
CREATE TABLE titlecode(  
  title_id           NUMBER PRIMARY KEY,  
  title_th           VARCHAR(100),  
  title_th_abbrev   VARCHAR(30),  
  title_en           VARCHAR(100),  
  title_en_abbrev   VARCHAR(30));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

--4)create address table

```
CREATE TABLE address(
  add_id          NUMBER PRIMARY KEY,
  num             VARCHAR(13),
  moo            VARCHAR(30),
  soi            VARCHAR(50),
  zipcode        VARCHAR(5),
  road_id        NUMBER);
```

--5)create personal_education table

```
CREATE TABLE personal_education(
  citizen_id     NUMBER NOT NULL,
  degree_id     NUMBER NOT NULL,
  qua_id        NUMBER NOT NULL,
  major_id     NUMBER NOT NULL,
  int_id        NUMBER NOT NULL,
  graduationday DATE,
  gpa           NUMBER,
  CONSTRAINT personal_education_pk PRIMARY KEY(citizen_id
, degree_id, qua_id, major_id,int_id));
```

--6)create degree_level table

```
CREATE TABLE degree_level (
  degree_id     NUMBER PRIMARY KEY,
  degree_th     VARCHAR(50),
  degree_th_abbr VARCHAR(20));
```

--7)create qualification table

```
CREATE TABLE qualification (
  qua_id        NUMBER PRIMARY KEY,
  qua_th        VARCHAR(50),
  qua_th_abbr   VARCHAR(20),
  qua_en        VARCHAR(50),
  qua_en_abbr   VARCHAR(20));
```

--8)create major table

```
CREATE TABLE major(
  major_id     NUMBER PRIMARY KEY,
  major_th     VARCHAR(50),
  major_en     VARCHAR(50));
```

--9)create institution_country table

```
CREATE TABLE institution_country (
  int_id       NUMBER PRIMARY KEY,
  coun_id      NUMBER );
```

--10)create institution_country table

```
CREATE TABLE employeetype (
  emptye_id    NUMBER PRIMARY KEY,
  emptye_name  VARCHAR(30));
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

--11)create fieldtype table

CREATE TABLE fieldtype (
    ftype_id          NUMBER PRIMARY KEY,
    ftype_name        VARCHAR(30));

--12)create posAcademic table

CREATE TABLE posAcademic (
    posaca_id         NUMBER PRIMARY KEY,
    posaca_name       VARCHAR(50));

--13)create posAdmin table

CREATE TABLE posAdmin (
    posadmin_id      NUMBER PRIMARY KEY,
    posadmin_name    VARCHAR(50));

commit;

```

5.1.2 การสร้างตารางแบบเชิงเวลา 22 ตาราง

การสร้างตารางแบบเชิงเวลาทั้ง 22 ตาราง จะใช้คำสั่งสร้างตารางได้ตามปกติ แต่ไม่ต้องกำหนดคอลัมน์วันเริ่มต้นและวันสิ้นสุด

```

--1)create personal_titlecode table

CREATE TABLE personal_titlecode(
    citizen_id       VARCHAR(13) PRIMARY KEY,
    title_id         NUMBER );

--2)create personal_fname table

CREATE TABLE personal_fname(
    citizen_id       VARCHAR(13) PRIMARY KEY,
    fname           VARCHAR(30));

--3)create personal_lname table

CREATE TABLE personal_lname (
    citizen_id       VARCHAR2(13) PRIMARY KEY,
    lname           VARCHAR2(30));

--4)create road_name table

CREATE TABLE road_name(
    road_id          NUMBER PRIMARY KEY,
    road_name        VARCHAR(50));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

--5)create address_subdistrict table
CREATE TABLE address_subdistrict(
    add_id          NUMBER PRIMARY KEY,
    sub_id         NUMBER);

--6)create subdistrict_name table
CREATE TABLE subdistrict_name(
    sub_id         NUMBER PRIMARY KEY,
    sub_name      VARCHAR(30));

--7)create subdistrict_district table
CREATE TABLE subdistrict_district(
    sub_id         NUMBER PRIMARY KEY,
    dis_id        NUMBER);

--8)create district_name table
CREATE TABLE district_name(
    dis_id        NUMBER PRIMARY KEY,
    dis_name     VARCHAR(30));

--9)create district_province table
CREATE TABLE district_province(
    dis_id        NUMBER PRIMARY KEY,
    pro_id       NUMBER);

--10)create province_name table
CREATE TABLE province_name(
    pro_id       NUMBER PRIMARY KEY,
    pro_name    VARCHAR(30));

--11)create personal_address table
CREATE TABLE personal_address(
    citizen_id   VARCHAR(13) PRIMARY KEY,
    add_id      NUMBER);

--12)create institution_name table
CREATE TABLE institution_name (
    int_id      NUMBER PRIMARY KEY,
    int_name   VARCHAR(100));

--13)create country_name table
CREATE TABLE country_name (
    coun_id    NUMBER PRIMARY KEY,
    coun_name  VARCHAR(100));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

--14)create faculty table

```
CREATE TABLE faculty (
  fac_id NUMBER PRIMARY KEY,
  fac_name VARCHAR(50));
```

--15)create department table

```
CREATE TABLE department (
  dep_id NUMBER PRIMARY KEY,
  dep_name VARCHAR(50));
```

--16)create department_faculty table

```
CREATE TABLE department_faculty (
  dep_id NUMBER PRIMARY KEY,
  fac_id NUMBER);
```

--17)create personal_employeenumber table

```
CREATE TABLE personal_employeenumber (
  emp_number NUMBER NOT NULL,
  emptye_id NUMBER NOT NULL,
  citizen_id VARCHAR(13),
  CONSTRAINT personal_employeetype_pk PRIMARY KEY(emp_number
, emptye_id));
```

--18)create personal_fieldtype table

```
CREATE TABLE personal_fieldtype (
  citizen_id VARCHAR(13) PRIMARY KEY,
  ftype_id NUMBER);
```

--19)create personal_department table

```
CREATE TABLE personal_department (
  citizen_id VARCHAR(13) PRIMARY KEY,
  dep_id NUMBER);
```

--20)create personal_posacademic table

```
CREATE TABLE personal_posacademic (
  citizen_id VARCHAR(13) PRIMARY KEY,
  posaca_id NUMBER);
```

--21)create personal_posadmin table

```
CREATE TABLE personal_posadmin (
  citizen_id VARCHAR(13),
  posadmin_id NUMBER NOT NULL,
  CONSTRAINT personal_posadmin_pk PRIMARY KEY(citizen_id,
posadmin_id));
```

--22)create personal_salary table

```
CREATE TABLE personal_salary (
  citizen_id VARCHAR(13) PRIMARY KEY,
  salary NUMBER);
```

```
commit;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.3 การทำให้ตารางปกติ (non-Temporal) เป็นตารางเชิงเวลา (Temporal Table)

```

--1)
EXECUTE DBMS_WM.EnableVersioning('personal_titlecode',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

--2)
EXECUTE DBMS_WM.EnableVersioning('personal_fname',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

--3)
EXECUTE DBMS_WM.EnableVersioning('personal_lname',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

--4)
EXECUTE DBMS_WM.EnableVersioning('road_name',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

--5)
EXECUTE DBMS_WM.EnableVersioning('address_subdistrict',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

--6)
EXECUTE DBMS_WM.EnableVersioning('subdistrict_name',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

--7)
EXECUTE DBMS_WM.EnableVersioning('subdistrict_district',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

--8)
EXECUTE DBMS_WM.EnableVersioning('district_name',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

--9)
EXECUTE DBMS_WM.EnableVersioning('district_province',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

--10)
EXECUTE DBMS_WM.EnableVersioning('province_name',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

--11) EXECUTE DBMS_WM.EnableVersioning('personal_address',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

--12) EXECUTE DBMS_WM.EnableVersioning('institution_name',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

--13) EXECUTE DBMS_WM.EnableVersioning('country_name',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

--14) EXECUTE DBMS_WM.EnableVersioning('faculty',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

--15) EXECUTE DBMS_WM.EnableVersioning('department',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

--16) EXECUTE DBMS_WM.EnableVersioning('department_faculty',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

--17) EXECUTE DBMS_WM.EnableVersioning('personal_employeenumber',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

--18) EXECUTE DBMS_WM.EnableVersioning('personal_fieldtype',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

--19) EXECUTE DBMS_WM.EnableVersioning('personal_department',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

--20) EXECUTE DBMS_WM.EnableVersioning('personal_posacademic',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

--21) EXECUTE DBMS_WM.EnableVersioning('personal_posadmin',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

--22) EXECUTE DBMS_WM.EnableVersioning('personal_salary',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

commit;

```

หมายเหตุ การกำหนดให้เป็นตารางเชิงเวลา 22 ตารางใช้เวลาประมาณ 1 นาที 15 วินาที
เฉลี่ยตารางละ 3.4 วินาที ทั้งนี้ขึ้นอยู่กับสมรรถนะของคอมพิวเตอร์แต่ละเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การตอบข้อสนเทศด้านประวัติชื่อ-นามสกุล

การเปลี่ยนแปลงประวัติ คำนำหน้า ชื่อ และนามสกุล เป็นอีกปัญหาหนึ่ง ที่ระบบฐานข้อมูลแบบเดิม (Non-Temporal) ไม่สามารถตอบข้อสนเทศเชิงเวลาได้ เนื่องจากฐานข้อมูลแบบเดิม ออกแบบมาเพื่อเก็บข้อมูล ณ เวลาปัจจุบัน ดังตัวอย่างปัญหาดังนี้

5.2.1 ตัวอย่างปัญหาเชิงเวลา ด้านประวัติชื่อ-นามสกุล

1) ตัวอย่างโครงสร้างตารางบันทึกชื่อนามสกุล ปัญหาที่ 1

รหัสประชาชน	รหัสคำนำหน้านาม	ชื่อ	นามสกุล
Citizen_id	TitleCode_ID	Fname	Lname

รูปที่ 5.1 โครงสร้างตารางบันทึกชื่อนามสกุล ปัญหาที่ 1

จากรูปที่ 5.1 เมื่อมีการแก้ไข คำนำหน้านาม ชื่อ และนามสกุล จะทำให้ค่าข้อมูลเดิมที่มีอยู่ ถูกแทนที่ด้วยข้อมูลใหม่ที่ทำกรแก้ไข คำนำหน้านาม ชื่อ และนามสกุลเดิม จะหายไปจากระบบ ไม่สามารถแสดงข้อมูลในอดีตได้

2) ตัวอย่างโครงสร้างตารางบันทึกชื่อนามสกุล ปัญหาที่ 2

รหัสประชาชน	ลำดับที่	รหัสคำนำหน้านาม	ชื่อ	นามสกุล	วันที่แก้ไข
Citizen_id	Order_Number	TitleCode_ID	Fname	Lname	Date

รูปที่ 5.2 โครงสร้างตารางบันทึกชื่อนามสกุล ปัญหาที่ 2

จากรูปที่ 5.2 มีการเพิ่มลำดับที่เป็นคีย์ร่วมกับรหัสประชาชน สามารถบันทึกประวัติการเปลี่ยนแปลงชื่อนามสกุลได้ แต่ถ้าบุคคลหนึ่งเปลี่ยนนามสกุล 3 ครั้ง แต่ชื่อยังคงเป็นชื่อเดิม จะเกิดความซ้ำซ้อนของคอลัมน์ชื่อ อีกทั้งเมื่อต้องการให้แสดงชื่อ ณ ช่วงเวลาๆ ต่าง จะไม่สามารถแสดงได้

ปัญหาตัวทั้งสองตัวอย่างที่กล่าวมานี้ เป็นปัญหาที่ระบบฐานข้อมูลแบบเดิม (non-Temporal) ตอบข้อสนเทศเชิงเวลาได้ค่อนข้างยุ่งยาก บางครั้งอาจต้องเขียนคำสั่ง SQL มาตรฐานถึง 5 คำสั่ง เพื่อตอบคำถามที่เกี่ยวข้องกับเชิงเวลา การทดลองในส่วนข้อที่ 5.2 นี้ จะทดลองเพื่อตอบคำถามข้อสนเทศเชิงเวลาได้ถูกต้อง

การทดลองการบันทึก ประวัติชื่อ-นามสกุล นี้ จะสมมุติบุคลากรชื่อ น.ส.กนกพร บุญรักษา เลขประจำตัวประชาชน 88888 เข้าบรรจุทำงานเมื่อวันที่ 15 กรกฎาคม 2535 ต่อมาคุณกนกพร ได้สมรสและแจ้งเปลี่ยนนามสกุลตามสามี ชื่อ นางกนกพร ถนอมรัก เมื่อวันที่ 20 กุมภาพันธ์ 2551 ต่อมาในภายหลังคุณกนกพร ได้หย่าร้างกับสามีเมื่อ 26 พฤษภาคม 2556 แจ้งขอเปลี่ยนชื่อและกลับมาใช้นามสกุลเดิม เป็น น.ส.สมปรารถนา บุญรักษา

ทั้งนี้ผู้สนใจสามารถทดลองการปรับปรุง ประวัติชื่อ-นามสกุล ได้ตามเอกสารภาคผนวก ก. หากปฏิบัติตามข้อ 5.1 แล้ว ในภาคผนวก ก. ให้เข้าส่วนการสร้างตาราง และไปปฏิบัติคำสั่งการเพิ่มชื่อ-นามสกุลได้เลย

5.2.2 ทดลองเพิ่มข้อมูล น.ส.กนกพร บุญรักษา เลขประจำตัวประชาชน 88888 เข้าทำงานวันที่ 15 กรกฎาคม 2535 (15-07-1992) โดยเพิ่มข้อมูลในตาราง PERSONAL_TITLECODE PERSONAL_FNAME และ PERSONAL_LNAME ตามลำดับดังนี้

```
INSERT INTO personal_titlecode VALUES(
  '88888',1004,
  WMSYS.WM_PERIOD(TO_DATE('15-07-1992','DD-MM-YYYY'),
    DBMS_WM.MAX_TIME));

INSERT INTO personal_fname VALUES(
  '88888','กนกพร',
  WMSYS.WM_PERIOD(TO_DATE('15-07-1992','DD-MM-YYYY'),
    DBMS_WM.MAX_TIME));

INSERT INTO personal_lname VALUES(
  '88888','บุญรักษา',
  WMSYS.WM_PERIOD(TO_DATE('15-07-1992','DD-MM-YYYY'),
    DBMS_WM.MAX_TIME));

commit;
```

5.2.3 น.ส.กนกพร บุญรักษา ได้แต่งงาน วันที่ 20 กุมภาพันธ์ 2551 (20-02-2008) จึงแจ้งขอเปลี่ยนคำนำหน้านามและนามสกุลตามสามีเป็น นางกนกพร ถนอมรัก

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('20-02-2008','DD-MM-YYYY'), DBMS_WM.MAX_TIME);

UPDATE personal_titlecode SET title_id =1003 WHERE citizen_id = '88888';

UPDATE personal_lname SET lname = 'ถนอมรัก' WHERE citizen_id = '88888';

commit;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.4 นางกนกพร ถนอมรัก (35003) ได้หย่ากับสามี วันที่ 26 พฤษภาคม 2556 (26-05-2013) จึงแจ้งขอเปลี่ยนชื่อและกลับกลับมาใช้นามสกุลเดิม เป็น น.ส.สมปรารถนา บุญรักษา

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('26-05-2013','DD-MM-YYYY'), DBMS_WM.MAX_TIME);

UPDATE personal_titlecode SET title_id =1004 WHERE citizen_id = '88888';

UPDATE personal_fname SET fname = 'สมปรารถนา' WHERE citizen_id = '88888';

UPDATE personal_lname SET lname = 'บุญรักษา' WHERE citizen_id = '88888';
commit;
```

5.2.5 ตรวจสอบประวัติชื่อ-นามสกุล ณ ช่วงเวลาปัจจุบัน หลังจากคุณกนกพร ได้หย่าแล้ว

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT title_th_abbr, fname, lname
FROM ((personal JOIN (personal_titlecode JOIN titlecode ON
personal_titlecode.title_id = titlecode.title_id) ON
personal.citizen_id = personal_titlecode.citizen_id)
JOIN personal_fname ON personal.citizen_id =
personal_fname.citizen_id)
JOIN personal_lname ON personal.citizen_id =
personal_lname.citizen_id
WHERE personal.citizen_id='88888';
```

TITLE_TH_ABBR	FNAME	LNAME
น.ส.	สมปรารถนา	บุญรักษา

1 rows selected

5.2.6 ตรวจสอบประวัติชื่อ-นามสกุล ณ ช่วงเวลา 2 ปีแรกที่เข้าทำงาน

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('1992-07-15','YYYY-MM-DD'),
TO_DATE('1994-07-15','YYYY-MM-DD'));

SELECT title_th_abbr, fname, lname
FROM ((personal JOIN (personal_titlecode JOIN titlecode ON
personal_titlecode.title_id = titlecode.title_id) ON
personal.citizen_id = personal_titlecode.citizen_id)
JOIN personal_fname ON personal.citizen_id =
personal_fname.citizen_id)
JOIN personal_lname ON personal.citizen_id =
personal_lname.citizen_id
WHERE personal.citizen_id='88888';
```

TITLE_TH_ABBR	FNAME	LNAME
น.ส.	กนกพร	บุญรักษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.7 สรุปผลการทดลองเพื่อตอบข้อสนทนาศึ่งเวลาด้านประวัติชื่อ-นามสกุล

จากการทดลอง สามารถเพิ่มข้อมูลชื่อ-นามสกุล (**Insert**) ตามช่วงเวลาที่ต้องการได้ โดยระบุช่วงเวลาเข้าไปในส่วนของการเพิ่มข้อมูล และในการปรับปรุงข้อมูล (**Update**) สามารถกำหนดช่วงเวลาที่ต้องการปรับปรุง โดยใช้คำสั่ง **SetValidTime()** จากนั้นสามารถใช้คำสั่งปรับปรุงได้ตามปกติ

การใช้คำสั่ง **SetValidTime()** ซึ่งเป็นคำสั่งที่รองรับข้อมูลด้านเชิงเวลา ทำให้สะดวกในการใช้คำสั่งนักพัฒนาโปรแกรมไม่ต้องกังวลในการตรวจสอบช่วงเวลาทับซ้อนหรือคาบเกี่ยวกัน เมื่อใช้คำสั่ง **SetValidTime()** แล้วสามารถใช้คำสั่ง **Update Delete** ได้ตามปกติ

5.3 การตอบข้อสนทนาศึ่งเวลาด้านประวัติที่อยู่

ระบบฐานข้อมูลแบบเดิม (**non-Temporal Database**) จะไม่มีการบันทึกช่วงเวลาของการเปลี่ยนแปลงข้อมูลประวัติที่อยู่ ว่าข้อมูลนั้นๆ เริ่มต้นเมื่อใด (**Start Time**) สิ้นสุดเวลาใด (**End Time**) ทำให้เมื่อเราต้องการสอบถามข้อมูลที่สัมพันธ์กับเวลา ในอดีต ปัจจุบัน และอนาคต จึงไม่สามารถตอบคำถามเกี่ยวกับเชิงเวลาได้ หรือทำได้ค่อนข้างยาก เพราะต้องเขียนคำสั่งตรวจสอบช่วงระยะเวลาต่างๆ หลายคำสั่ง หาก **DBMS** ที่ใช้ไม่รองรับคำสั่งเชิงเวลา (**Temporal Query**) เช่นกัน ดังตัวอย่างดังต่อไปนี้

5.3.1 ตัวอย่างปัญหาเชิงเวลา ด้านประวัติที่อยู่

1) ตัวอย่างโครงร่างตารางบันทึกที่อยู่ ปัญหาที่ 1

รหัสประชาชน	เลขที่	หมู่ที่	ซอย	ถนน	รหัสตำบล	รหัสอำเภอ	รหัสจังหวัด	รหัสไปรษณีย์
	Num	Moo	Soi	Road	SubDistric_ID	Distric_ID	Province_id	Zipcode

รูปที่ 5.3 โครงร่างตารางบันทึกที่อยู่ ปัญหาที่ 1

จากโครงร่างตารางแสดงในรูปที่ 5.3 นี้ จะเห็นว่า นำคอลัมน์ เลขที่บ้าน หมู่ ซอย ถนน ตำบล อำเภอ จังหวัด รหัสไปรษณีย์ อยู่ร่วมตารางเดียวกันกับเลขรหัสประจำตัวประชาชน (**Citizen_Id**) เมื่อมีการเปลี่ยนแปลงที่อยู่ ก็จะนำข้อมูลที่อยู่ใหม่ ไปบันทึกแทนที่ในข้อมูลเดิม ทำให้ข้อมูลที่อยู่เดิมหายไปจากระบบ ดังนั้น เวลาสอบถามประวัติการเปลี่ยนแปลงที่อยู่ จะแสดงเฉพาะที่อยู่ในปัจจุบันเท่านั้น ไม่สามารถแสดงข้อมูลในอดีตได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ตัวอย่างโครงร่างตารางบันทึกที่อยู่ ปัญหาที่ 2

รหัสบ้าน Add_ID	เลขที่ Num	หมู่ที่ Moo	ซอย Soi	ถนน Road	รหัสตำบล SubDistric_ID	รหัสอำเภอ Distric_ID	รหัสจังหวัด Province_id	รหัสไปรษณีย์ Zipcode
รหัสประชาชน Citizen_id	รหัสบ้าน Add_ID							

รูปที่ 5.4 โครงร่างตารางบันทึกที่อยู่ ปัญหาที่ 2

จากโครงร่างตารางรูปที่ 5.4 นี้ บุคลากรหนึ่งคนสามารถบันทึกข้อมูลที่อยู่ได้หลายที่อยู่ได้ (กำหนดห้ามซ้ำที่อยู่เดิม) และสามารถสอบถามข้อมูลที่อยู่ย้อนหลังของแต่ละบุคคลได้

แต่ถ้าถามข้อสนเทศที่เกี่ยวข้องกับเวลา เช่น ในช่วงปี 2500-2535 มีใครบ้างที่เคยอาศัยอยู่ใน จ.อุดรธานี ตัวอย่างนี้ก็ตอบไม่ได้ และในกรณีชื่อของคอลัมน์เปลี่ยน เช่น ชื่อตำบล ชื่ออำเภอ ชื่อจังหวัด เปลี่ยนไป ก็ไม่สามารถตอบข้อมูลย้อนหลัง ระบบแบบเดิมจะทำการแทนรหัสตำบล อำเภอ จังหวัด เหมือนตัวอย่างในรูปที่ 5.3 ก็จะเข้าปัญหาเดิม คือไม่สามารถตอบคำถามในอดีตได้

3) ตัวอย่าง การจัดตั้งจังหวัดใหม่

จากที่ อ.หนองบัวลำภู เคยเป็นอำเภอหนึ่งใน จ.อุดรธานี ต่อมาได้จัดตั้งเป็น จ.หนองบัวลำภู ตั้งแต่ปี 2536 และ อ.หนองบัวลำภู เปลี่ยนชื่อเป็น “เมืองหนองบัวลำภู” ระบบฐานข้อมูลแบบเดิม (non-Temporal) จะทำการเพิ่มรหัสจังหวัดหนองบัวลำภู และรหัสอำเภอเมืองหนองบัวลำภูเป็นรหัสใหม่ แล้วไปแทนที่หรือไปเพิ่มในข้อมูลเดิม ทำให้เกิดข้อมูลรหัสอำเภอ 2 รหัสที่ต่างกัน ชื่อต่างกัน แต่จริงๆ คืออำเภอเดียวกัน

ถ้าเราถามว่า “จำนวนประชากรทั้งหมดที่เคยอาศัยอยู่ใน จ.อุดรธานี ในช่วงก่อนปี 2536” ถ้าเป็นฐานข้อมูลแบบเดิม (non-Temporal) จะไม่สามารถตอบคำถามนี้ได้ เพราะประชากรที่เคยอาศัยอยู่ จ.อุดรธานี ได้ย้ายสังกัดมาอยู่ จ.หนองบัวลำภู ในฐานข้อมูลได้ถูกแทนที่ด้วยรหัส จ.หนองบัวลำภู ทั้งหมดแล้ว

5.3.2 การทดลองข้อมูลเชิงเวลาด้านที่อยู่

เพื่อความเข้าใจเกี่ยวกับข้อมูลแบบเชิงเวลาที่สัมพันธ์กับที่อยู่ จะสมมุติข้อมูลประชากรที่อาศัยอยู่ใน จ.อุดรธานี จำนวน 10 คน โดยสังกัด อ.เมืองอุดรธานี 6 คน และสังกัด อ.หนองบัวลำภู 4 คน เมื่อจัดตั้งจังหวัดหนองบัวลำภู ในวันที่ 1 ธ.ค. 2536 แล้ว จะทำการย้าย (ปรับปรุงข้อมูล) ประชากรที่อาศัย อ.หนองบัวลำภู ให้สังกัด จ.หนองบัวลำภู และเปลี่ยนเป็น อ.เมืองหนองบัวลำภู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.1 ข้อมูลสมมุติ ของประชากรที่อาศัยอยู่ใน จ.อุตรธานี (ข้อมูลก่อนปี 2536)

ที่	เลขประจำตัวประชาชน	คำ นาม	ชื่อ	สกุล	รหัส บ้าน	เลข ที่	หมู่	ซอย	ตำบล	อำเภอ	จังหวัด	ไปรษณีย์	ว.ค.ป.(ค.ศ) ย้าย/บันทึก
1	11111	นาย	บุญทัน	รักถนอม	1001	1	1		หมากแข้ง	เมืองอุตรธานี	อุตรธานี	41000	15/05/2512 (1969)
2	22222	นาง	อนงค์	รักถนอม	1001	1	1		หมากแข้ง	เมืองอุตรธานี	อุตรธานี	41000	15/05/2512 (1969)
3	33333	นาย	นเรศ	อุตรธรรม	1002	2	1		หมากแข้ง	เมืองอุตรธานี	อุตรธานี	41000	01/01/2501 (1958)
4	44444	นาง	สมใจ	อุตรธรรม	1002	2	1		หมากแข้ง	เมืองอุตรธานี	อุตรธานี	41000	01/01/2501 (1958)
5	55555	นาง	ปราณี	จิตดำรงค์	1003	3	1		หมากแข้ง	เมืองอุตรธานี	อุตรธานี	41000	01/01/2501 (1958)
6	66666	นาย	สิทธิ	รักถนอม	3001	1	3		หนองบัว	หนองบัวลำภู	อุตรธานี	41390	28/02/2519 (1976)
7	77777	นาย	เจษฎา	ถิ่นรักษา	4001	1	4		หมากแข้ง	เมืองอุตรธานี	อุตรธานี	41000	08/08/2508 (1965)
8	88888	น.ส.	กนกพร	บุญรักษา	5001	1	5		หนองบัว	หนองบัวลำภู	อุตรธานี	41390	14/07/2529 (1986)
9	99999	นาง	ปราณี	ทศเจริญ	5001	1	5		หนองบัว	หนองบัวลำภู	อุตรธานี	41390	14/07/2529 (1986)
10	10000	นาย	ปิใหม่	มงคลทัฬ	5001	1	5		หนองบัว	หนองบัวลำภู	อุตรธานี	41390	14/70/2529 (1986)

หมายเหตุ

1. จ.อุตรธานี ถือว่ามีการบันทึกในทะเบียนราษฎร วันที่ 21 มิถุนายน พ.ศ. 2475
2. อ.หนองบัวลำภู แยกจาก จ.อุตรธานี จึงตั้งเป็น จ.หนองบัวลำภู วันที่ 1 ธ.ค. พ.ศ. 2536

5.3.3 ทดลองเพิ่มข้อมูล (Insert) ปรับปรุง (Update) ข้อมูลประชากรใน จ.อุตรธานี ดังเอกสารผนวก ข. ทำการเพิ่มข้อมูล จังหวัด อำเภอ ตำบล โดยกำหนดวันเริ่มต้นข้อมูล 21 มิ.ย. 2475 (1932)

- 1) เพิ่มชื่อจังหวัด ในตาราง province_name
- 2) เพิ่มชื่ออำเภอ ในตาราง district_name
- 3) เพิ่มรหัสอำเภอ สังกัดรหัสจังหวัด ในตาราง district_province
- 4) เพิ่มชื่อตำบล ในตาราง subdistrict_name
- 5) เพิ่มรหัสตำบล สังกัดรหัสอำเภอ ในตาราง subdistrict_district
- 6) เพิ่มชื่อและรหัสถนน ในตาราง road_name
- 7) เพิ่มข้อมูลของรหัสบ้าน ในตาราง address
- 8) เพิ่มรหัสบ้าน สังกัดตำบล ในตาราง address_subdistrict
- 9) เพิ่ม เลขประจำตัวประชาชน เข้ากับรหัสบ้าน ในตาราง personal_address

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.4 เมื่อเพิ่มข้อมูลประชากรทุกคน ตามตารางที่ 5.1 และข้อ 5.3.3 แล้ว ซึ่งประชากรทุกคนสังกัด จ.อุดรธานี ใช้คำสั่ง **SetValidTime()** ตรวจสอบข้อมูลที่อยู่ ณ ช่วงเวลาปัจจุบันได้ดังนี้

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT personal_address.citizen_id, address.num, address.moo,
address soi, road_name.road_name, subdistrict_name.sub_name,
district_name.dis_name, province_name.pro_name, address.zipcode
FROM ((((((personal_address JOIN address ON personal_address.add_id
= address.add_id)
JOIN road_name ON address.road_id = road_name.road_id)
JOIN address_subdistrict ON personal_address.add_id =
address_subdistrict.add_id)
JOIN subdistrict_name ON address_subdistrict.sub_id =
subdistrict_name.sub_id)
JOIN subdistrict_district ON address_subdistrict.sub_id =
subdistrict_district.sub_id)
JOIN district_name ON subdistrict_district.dis_id =
district_name.dis_id)
JOIN district_province ON subdistrict_district.dis_id =
district_province.dis_id)
JOIN province_name ON district_province.pro_id = province_name.pro_id
ORDER BY personal_address.citizen_id;
```

anonymous block completed

CITIZEN_ID	NUM	MOO	SOI	ROAD	SUB_NAME	DIS_NAME	PRO_NAME	ZIPCODE
10000	1	5		หนองบัวลำภู	หนองบัว	หนองบัวลำภู	อุดรธานี	41390
11111	1	1		อุดรคูมฤ	หมากแข้ง	เมืองอุดรธานี	อุดรธานี	41000
22222	1	1		อุดรคูมฤ	หมากแข้ง	เมืองอุดรธานี	อุดรธานี	41000
33333	2	1		อุดรคูมฤ	หมากแข้ง	เมืองอุดรธานี	อุดรธานี	41000
44444	2	1		อุดรคูมฤ	หมากแข้ง	เมืองอุดรธานี	อุดรธานี	41000
55555	3	1		อุดรคูมฤ	หมากแข้ง	เมืองอุดรธานี	อุดรธานี	41000
66666	1	3		หนองบัวลำภู	หนองบัว	หนองบัวลำภู	อุดรธานี	41390
77777	1	4		อุดรคูมฤ	หมากแข้ง	เมืองอุดรธานี	อุดรธานี	41000
88888	1	5		หนองบัวลำภู	หนองบัว	หนองบัวลำภู	อุดรธานี	41390
99999	1	5		หนองบัวลำภู	หนองบัว	หนองบัวลำภู	อุดรธานี	41390

10 rows selected

1,496ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.5 ทำการเพิ่มข้อมูล จ.หนองบัวลำภู โดยกำหนดวันเริ่มต้นข้อมูล 1 ธ.ค. 2536 (1993)

- 1) เพิ่มชื่อจังหวัด หนองบัวลำภู ในตาราง **province_nam**
- 2) ปรับปรุงชื่ออำเภอ เมืองหนองบัวลำภู ในตาราง **district_name**
- 3) ปรับปรุง อ.เมืองหนองบัวลำภู สังกัด จ.หนองบัวลำภู ตาราง**district_province**

5.3.6 เมื่อปรับปรุงข้อมูล จ.หนองบัวลำภูเรียบร้อยแล้ว ใช้คำสั่งตรวจสอบดูข้อมูล ณ ช่วงเวลาปัจจุบัน จะพบว่า มีประชากร 4 คน สังกัด จ.หนองบัวลำภู และมีประชากร 6 คนสังกัด จ.อุดรธานี

```
EXECUTE DBMS_WM.SetValidTime( );
```

```
SELECT personal_address.citizen_id, address.num, address.moo,
address soi, road_name.road_name, subdistrict_name.sub_name,
district_name.dis_name, province_name.pro_name, address.zipcode
FROM personal_address, address, road_name, address_subdistrict,
subdistrict_name, subdistrict_district, district_name,
district_province, province_name
WHERE personal_address.add_id = address.add_id
AND address.road_id = road_name.road_id
AND personal_address.add_id = address_subdistrict.add_id
AND address_subdistrict.sub_id = subdistrict_name.sub_id
AND address_subdistrict.sub_id = subdistrict_district.sub_id
AND subdistrict_district.dis_id = district_name.dis_id
AND subdistrict_district.dis_id = district_province.dis_id
AND district_province.pro_id = province_name.pro_id
ORDER BY personal_address.citizen_id;
```

anonymous block completed

CITIZEN_ID	NUM	MOO	SOI	ROAD	SUB_NAME	DIS_NAME	PRO_NAME	ZIPCODE
10000	1	5		หนองบัวลำภู	หนองบัว	เมืองหนองบัวลำภู	หนองบัวลำภู	41390
11111	1	1		อุดรคูมฤ์	หมากแข้ง	เมืองอุดรธานี	อุดรธานี	41000
22222	1	1		อุดรคูมฤ์	หมากแข้ง	เมืองอุดรธานี	อุดรธานี	41000
33333	2	1		อุดรคูมฤ์	หมากแข้ง	เมืองอุดรธานี	อุดรธานี	41000
44444	2	1		อุดรคูมฤ์	หมากแข้ง	เมืองอุดรธานี	อุดรธานี	41000
55555	3	1		อุดรคูมฤ์	หมากแข้ง	เมืองอุดรธานี	อุดรธานี	41000
66666	1	3		หนองบัวลำภู	หนองบัว	เมืองหนองบัวลำภู	หนองบัวลำภู	41390
77777	1	4		อุดรคูมฤ์	หมากแข้ง	เมืองอุดรธานี	อุดรธานี	41000
88888	1	5		หนองบัวลำภู	หนองบัว	เมืองหนองบัวลำภู	หนองบัวลำภู	41390
99999	1	5		หนองบัวลำภู	หนองบัว	เมืองหนองบัวลำภู	หนองบัวลำภู	41390

10 rows selected

16ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.7 สอบถามจำนวนประชากร ณ ช่วงเวลาปัจจุบัน จะเห็นว่าจำนวนประชากร จะแสดง

ได้ถูกต้อง โดยจำนวนประชากร จ.อุดรธานี = 6 คน และ จ.หนองบัวลำภู = 4 คน

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT province_name.pro_id, province_name.pro_name,
Count(personal_address.citizen_id) AS AMOUNT
FROM personal_address, address_subdistrict, subdistrict_district,
district_province, province_name
WHERE personal_address.add_id = address_subdistrict.add_id
AND address_subdistrict.sub_id = subdistrict_district.sub_id
AND subdistrict_district.dis_id = district_province.dis_id
AND district_province.pro_id = province_name.pro_id
GROUP BY province_name.pro_id , province_name.pro_name;
```

```
anonymous block completed
PRO_ID          PRO_NAME          AMOUNT
-----
41              อุดรธานี          6
39              หนองบัวลำภู      4

2 rows selected
6ms elapsed
```

5.3.8 สอบถามจำนวนประชากร ปี 2500-2535 (1957-1992) จะแสดงจำนวนประชากร

ได้ถูกต้อง คือ จำนวนประชากร ณ ช่วงเวลาดังกล่าว จะมีเฉพาะแค่ จ.อุดรธานี = 10 คน

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('1957','YYYY'),
TO_DATE('1992','YYYY') );

SELECT province_name.pro_id, province_name.pro_name,
Count(personal_address.citizen_id) AS AMOUNT
FROM personal_address, address_subdistrict, subdistrict_district,
district_province, province_name
WHERE personal_address.add_id = address_subdistrict.add_id
AND address_subdistrict.sub_id = subdistrict_district.sub_id
AND subdistrict_district.dis_id = district_province.dis_id
AND district_province.pro_id = province_name.pro_id
GROUP BY province_name.pro_id , province_name.pro_name;
```

```
anonymous block completed
PRO_ID          PRO_NAME          AMOUNT
-----
41              อุดรธานี          10

1 rows selected
7ms elapsed
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.9 สอบถามจำนวนประชากร ทุกช่วงเวลา ตั้งแต่อดีต ถึงปัจจุบัน จะแสดงจำนวนประชากรได้ถูกต้อง คือ จำนวนประชากร จ.อุดรธานี = 10 คน และ จ.หนองบัวลำภู = 4 คน

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT province_name.pro_id, province_name.pro_name,
Count(personal_address.citizen_id) AS AMOUNT
FROM personal_address, address_subdistrict, subdistrict_district,
district_province, province_name
WHERE personal_address.add_id = address_subdistrict.add_id
AND address_subdistrict.sub_id = subdistrict_district.sub_id
AND subdistrict_district.dis_id = district_province.dis_id
AND district_province.pro_id = province_name.pro_id
GROUP BY province_name.pro_id, province_name.pro_name;
anonymous block completed
PRO_ID          PRO_NAME          AMOUNT
-----
41              อุดรธานี         10
39              หนองบัวลำภู     4

2 rows selected

8ms elapsed
```

5.3.10 สอบถามจำนวนประชากรแต่ละอำเภอ ณ ช่วงเวลาปัจจุบัน

```
EXECUTE DBMS_WM.SetValidTime();

SELECT district_name.dis_id, district_name.dis_name,
Count(personal_address.citizen_id) AS AMOUNT
FROM personal_address, address_subdistrict, subdistrict_district,
district_name
WHERE personal_address.add_id = address_subdistrict.add_id
AND address_subdistrict.sub_id = subdistrict_district.sub_id
AND subdistrict_district.dis_id = district_name.dis_id
GROUP BY district_name.dis_id, district_name.dis_name;
anonymous block completed
DIS_ID          DIS_NAME          AMOUNT
-----
4101           เมืองอุดรธานี     6
4102           เมืองหนองบัวลำภู  4

2 rows selected

7ms elapsed
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.11 สอบถามจำนวนประชากรแต่ละอำเภอ ทุกช่วงเวลา ตั้งแต่อดีต จนถึงปัจจุบัน

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT district_name.dis_id, district_name.dis_name,
Count(personal_address.citizen_id) AS AMOUNT
FROM personal_address, address_subdistrict, subdistrict_district ,
district_name
WHERE personal_address.add_id = address_subdistrict.add_id
AND address_subdistrict.sub_id = subdistrict_district.sub_id
AND subdistrict_district.dis_id = district_name.dis_id
GROUP BY district_name.dis_id, district_name.dis_name;
```

anonymous block completed

DIS_ID	DIS_NAME	AMOUNT
4101	เมืองสุรธานี	6
4102	เมืองหนองบัวลำภู	4
4102	หนองบัวลำภู	4

DIS_ID	DIS_NAME	AMOUNT
4101	เมืองสุรธานี	6
4102	เมืองหนองบัวลำภู	4
4102	หนองบัวลำภู	4

3 rows selected

8ms elapsed

5.3.12 สรุปผลการทดลองเพื่อตอบข้อสนทนาศงเวลาด้านประวัติที่อยู่

จากการทดลอง สามารถเพิ่มข้อมูลประวัติที่อยู่ (Insert) ตามช่วงเวลาที่ต้องการได้ โดยระบุช่วงเวลาเข้าไปในส่วนของการเพิ่มข้อมูล และในการปรับปรุงข้อมูล (Update) สามารถกำหนดช่วงเวลาที่ต้องการปรับปรุง โดยใช้คำสั่ง SetValidTime() จากนั้นสามารถใช้คำสั่งปรับปรุงได้ตามปกติ

การใช้คำสั่ง SetValidTime() ซึ่งเป็นคำสั่งที่รองรับข้อมูลด้านเชิงเวลา ทำให้สะดวกในการใช้คำสั่ง นักพัฒนาโปรแกรมไม่ต้องกังวลในการตรวจสอบช่วงเวลาทับซ้อนหรือคาบเกี่ยวกัน เมื่อใช้คำสั่ง SetValidTime() แล้วสามารถใช้คำสั่ง Update Delete ได้ตามปกติ ทั้งนี้สามารถตอบสถิติที่เกี่ยวข้องกับเวลาได้

5.4 การตอบข้อสนเทศด้านประวัติการศึกษา

ข้อมูลประวัติการศึกษา มักพบว่า ชื่อสถาบันการศึกษามีการเปลี่ยนแปลงชื่อ ถ้าเป็นระบบฐานข้อมูลแบบเดิม (non-Temporal) เมื่อมีการเปลี่ยนชื่อสถาบัน จะแทนที่ชื่อเก่าด้วยชื่อใหม่ เช่น สถาบันเทคโนโลยีพระจอมเกล้า ศูนย์นันทบุรี เปลี่ยนชื่อมาแล้วทั้งหมด 5 ครั้ง ปัจจุบันชื่อ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ชื่อสถาบันเดิม 4 ชื่อจะหายไปจากระบบ มีเฉพาะชื่อสถาบันปัจจุบันเท่านั้น

ดังนั้นเมื่อสอบถามข้อสนเทศเชิงเวลา เช่น “มีใครบ้างเคยสำเร็จการศึกษาจากสถาบันเทคโนโลยีพระจอมเกล้า ศูนย์นันทบุรี” หรือ “ให้แสดงระดับการศึกษาและชื่อสถาบัน ณ วันที่สำเร็จการศึกษา” ระบบฐานข้อมูลแบบเดิม (non-Temporal) จะไม่สามารถตอบคำถามนี้ได้ แต่ระบบฐานข้อมูลแบบเชิงเวลา สามารถตอบข้อสนเทศเชิงเวลาได้ ดังการทดลองต่อไปนี้

5.4.1 ข้อมูลสมมุติ ประวัติการศึกษาของบุคลากร

จากตารางที่ 5.2 แสดงข้อมูลสมมุติ ประวัติการศึกษาของบุคลากร จะเห็นว่าผู้ที่สำเร็จการศึกษาจาก มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี มีทั้งหมด 3 คน คือ นายสมชาติ ทดเจริญ นายสิทธิ รักถนอม และ น.ส.กนกพร บุญรักษา แต่ชื่อของสถาบันต่างกัน เพราะสถาบันได้เปลี่ยนชื่อไปแล้วถึง 3 ครั้ง คือ วิทยาลัยเทคโนโลยีและอาชีวศึกษา สถาบันเทคโนโลยีราชมงคล และ มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี ตามลำดับ

ตารางที่ 5.2 ข้อมูลสมมุติ ประวัติการศึกษาของบุคลากร

เลข ปชช	คำ นำหน้า	ชื่อ	นามสกุล	ระดับ	วุฒิ	สาขา	สถานศึกษา	ว.ด.ป. สำเร็จ	GPA
33333	นาย	นรศ	อุดรธรรม	ป.ตรี	วศ.บ.	วิศวกรรมโทรคมนาคม	สถาบันเทคโนโลยีพระจอมเกล้า ศูนย์นันทบุรี	31/03/2517	4.00
				ป.โท	วศ.ม.	วิศวกรรมไฟฟ้า	สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตนันทบุรีลาดกระบัง	31/03/2520	4.00
				ป.เอก	วศ.ด	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	31/03/2550	4.00
35003	นาย	สมชาติ	ทดเจริญ	ป.ตรี	วศ.บ.	วิศวกรรมไฟฟ้า	วิทยาลัยเทคโนโลยีและอาชีวศึกษา	31/03/2525	3.50
				ป.โท	วศ.ม.	วิศวกรรมไฟฟ้า	สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ	31/03/2530	4.00
66666	นาย	สิทธิ	รักถนอม	ป.ตรี	วศ.บ.	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีราชมงคล	31/03/2544	3.00
				ป.โท	วศ.ม.	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	31/03/2557	3.60
88888	น.ส.	กนกพร	บุญรักษา	ป.ตรี	วศ.บ.	วิศวกรรมคอมพิวเตอร์	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	31/03/2550	4.00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.2 ประวัติการเปลี่ยนแปลงชื่อของสถาบันกลุ่มตัวอย่าง

1) สถานศึกษา : สจล. (www.kmitl.ac.th)	วันที่เปลี่ยนชื่อ
วิทยาลัยโทรคมนาคมนนทบุรี	1 พ.ค. 2507
สถาบันเทคโนโลยีพระจอมเกล้า ศูนย์นนทบุรี	24 ส.ค. 2514
สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตนนทบุรีลาดกระบัง	29 มิ.ย. 2517
สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตเจ้าคุณทหารลาดกระบัง	22 เม.ย. 2522
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	20 ก.พ. 2529
2) สถานศึกษา : มจพ. (www.kmutnb.ac.th)	
สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตพระนครเหนือ	24 เม.ย. 2514
สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ	20 ก.พ. 2529
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ	27 ธ.ค. 2550
3) สถานศึกษา : มทร.ธัญบุรี (www.rmutt.ac.th)	
วิทยาลัยเทคโนโลยีและอาชีวศึกษา	27 ก.พ. 2518
สถาบันเทคโนโลยีสยามมงคล	15 ก.ย. 2531
มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	18 ม.ค. 2548

5.4.3 เพื่อความเข้าใจการเปลี่ยนแปลงชื่อ ให้ทำการเพิ่มข้อมูล (Insert) ปรับปรุงข้อมูล (Update) ประวัติด้านการศึกษา ได้ตามผนวก ค. เพื่อตอบสนองประเทศเชิงเวลาต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.4 คำถามเชิงเวลา แสดงประวัติการเปลี่ยนชื่อของสถานศึกษา จัดเรียงลำดับตามรหัส และวันที่เปลี่ยนแปลงชื่อสถาบัน

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);
```

```
SELECT int_id AS "ID", int_name,
TO_CHAR(e.WM_VALID.VALIDFROM, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') FROM_DATE,
TO_CHAR(e.WM_VALID.VALIDTILL, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE
FROM institution_name e
ORDER BY int_id, WM_VALID.VALIDFROM;
```

ID	INT_NAME	FROM_DATE	TO_DATE
1500	สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตพระนครเหนือ	24-04-2514	20-02-2529
1500	สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ	20-02-2529	27-12-2550
1500	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ	27-12-2550	
1600	วิทยาลัยโทรคมนาคมนนทบุรี	01-05-2507	24-04-2514
1600	สถาบันเทคโนโลยีพระจอมเกล้า ศูนย์นนทบุรี	24-04-2514	29-06-2517
1600	สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตนนทบุรีลาดกระบัง	29-06-2517	22-04-2522
1600	สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตเจ้าคุณทหารลาดกระบัง	22-04-2522	20-02-2529
1600	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	20-02-2529	
19100	วิทยาลัยเทคโนโลยีและอาชีวศึกษา	27-02-2518	15-09-2531
19100	สถาบันเทคโนโลยีราชมงคล	15-09-2531	18-01-2548
19100	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	18-01-2548	

11 rows selected
9ms elapsed

ผลของการปฏิบัติคำสั่งเชิงเวลา โดยกำหนดช่วงเวลา **ValidTime** ตั้งแต่เวลา **MIN_TIME** ถึง **MAX_TIME** จะเห็นว่าแต่ละสถาบันในกลุ่มตัวอย่าง มีการเปลี่ยนแปลงชื่ออยู่หลายครั้ง เช่น วิทยาลัยโทรคมนาคมนนทบุรี จัดตั้ง (ประมาณ) เมื่อวันที่ 1 พฤษภาคม 2507 และเปลี่ยนชื่อเป็น “สถาบันเทคโนโลยีพระจอมเกล้า ศูนย์นนทบุรี” เมื่อวันที่ 24 เมษายน 2514 จากนั้นเปลี่ยนชื่อเป็น “สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตนนทบุรีลาดกระบัง” เมื่อวันที่ 29 มิถุนายน 2517 ต่อมาได้ทำการเปลี่ยนชื่ออีกครั้งเป็น “สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตเจ้าคุณทหารลาดกระบัง” เมื่อวันที่ 22 เมษายน 2522 สุดท้ายได้เปลี่ยนชื่อเป็น “สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง” เมื่อวันที่ 20 กุมภาพันธ์ 2529 จนถึงปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.5 แสดงประวัติการศึกษา ณ ช่วงเวลาปัจจุบัน

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT citizen_id, degree_th_abbr AS "ระดับ", qua_th_abbr AS "วุฒิ",
major_th AS "สาขา", int_name AS "สถานศึกษา",
TO_CHAR(graduationday, 'yyyy')+543 AS "พ.ศ."
FROM personal_education, degree_level, qualification, major,
institution_name
WHERE personal_education.degree_id = degree_level.degree_id
AND personal_education.qua_id = qualification.qua_id
AND personal_education.major_id = major.major_id
AND personal_education.int_id = institution_name.int_id
ORDER BY citizen_id , degree_level.degree_id;
```

anonymous block completed

CITIZEN_ID	ระดับ	วุฒิ	สาขา	สถานศึกษา	พ.ศ.
33333	ป.ตรี	วศ.บ.	วิศวกรรมโทรคมนาคม	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	2517
33333	ป.โท	วศ.ม.	วิศวกรรมไฟฟ้า	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	2520
33333	ป.เอก	วศ.ด.	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	2550
35003	ป.ตรี	วศ.บ.	วิศวกรรมไฟฟ้า	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	2525
35003	ป.โท	วศ.ม.	วิศวกรรมไฟฟ้า	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ	2530
66666	ป.ตรี	วศ.บ.	วิศวกรรมคอมพิวเตอร์	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	2544
66666	ป.โท	วศ.ม.	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	2557
88888	ป.ตรี	วศ.บ.	วิศวกรรมคอมพิวเตอร์	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	2550

8 rows selected

5.4.6 แสดงประวัติการศึกษา ณ พ.ศ. ที่บุคลากรสำเร็จการศึกษา (นำวันสำเร็จการศึกษา มาเปรียบเทียบกับช่วงเวลา กับช่วงเวลาการเปลี่ยนแปลงชื่อสถาบัน)

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT citizen_id, degree_th_abbr AS "DEGREE",
qualification.qua_th_abbr AS "วุฒิ", major.major_th as "สาขา",
int_name AS "institution", TO_CHAR(graduationday, 'yyyy')+543 AS
"พ.ศ."

FROM personal_education, degree_level, qualification, major,
institution_name
WHERE personal_education.degree_id = degree_level.degree_id
AND personal_education.qua_id = qualification.qua_id
AND personal_education.major_id = major.major_id
AND personal_education.int_id = institution_name.int_id
AND WM_OVERLAPS(institution_name.WM_VALID,
WM_PERIOD(personal_education.graduationday,
personal_education.graduationday))=1

ORDER BY citizen_id , degree_level.degree_id;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

anonymous block completed

CITIZEN_ID	ระดับ	วุฒิ	สาขา	สถานศึกษา	พ.ศ.
33333	ป.ตรี	ว.ศ.บ.	วิศวกรรมโทรคมนาคม	สถาบันเทคโนโลยีพระจอมเกล้า ศูนย์นนทบุรี	2517
33333	ป.โท	ว.ศ.ม.	วิศวกรรมไฟฟ้า	สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตนนทบุรีลาดกระบัง	2520
33333	ป.เอก	ว.ศ.ด.	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	2550
35003	ป.ตรี	ว.ศ.บ.	วิศวกรรมไฟฟ้า	วิทยาลัยเทคโนโลยีและอาชีวศึกษา	2525
35003	ป.โท	ว.ศ.ม.	วิศวกรรมไฟฟ้า	สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ	2530
66666	ป.ตรี	ว.ศ.บ.	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีราชมงคล	2544
66666	ป.โท	ว.ศ.ม.	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	2557
88888	ป.ตรี	ว.ศ.บ.	วิศวกรรมคอมพิวเตอร์	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	2550

8 rows selected

192ms elapsed

5.4.7 แสดงชื่อสถาบัน ณ วันปัจจุบัน ของบุคลากรทุกคน

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT citizen_id, TO_CHAR(graduationday,'yyyy')+543 AS "YEAR" ,
degree_th_abbr AS "DEGREE", institution_name.int_id AS "INT_ID",
int_name AS "institution"

FROM personal_education, degree_level, institution_name
WHERE personal_education.degree_id = degree_level.degree_id
AND personal_education.int_id = institution_name.int_id

ORDER BY citizen_id , degree_level.degree_id;
```

anonymous block completed

CITIZEN_ID	YEAR	DEGREE	INT_ID	institution
33333	2517	ป.ตรี	1600	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
33333	2520	ป.โท	1600	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
33333	2550	ป.เอก	1600	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
35003	2525	ป.ตรี	19100	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี
35003	2530	ป.โท	1500	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
66666	2544	ป.ตรี	19100	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี
66666	2557	ป.โท	1600	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
88888	2550	ป.ตรี	19100	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

8 rows selected

6ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.7 มีพนักงานที่สำเร็จการศึกษาจากสถาบันใดบ้าง และมีจำนวนเท่าใด ณ เวลาปัจจุบัน

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT personal_education.int_id, institution_name.int_name,
Count(personal_education.citizen_id) AS AMOUNT
FROM personal_education , institution_name
WHERE personal_education.int_id = institution_name.int_id
GROUP BY personal_education.int_id, institution_name.int_name;
```

anonymous block completed

INT_ID	INT_NAME	AMOUNT
1500	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ	1
19100	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	3
1600	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	4

5.4.8 สรุปผลการทดลองเพื่อตอบข้อสนทนาศึ่งเวลาด้านประวัติการศึกษา

จากการทดลอง สามารถเพิ่มข้อมูลประวัติการศึกษา (**Insert**) ตามช่วงเวลาที่ต้องการได้ โดยระบุช่วงเวลาเข้าไปในส่วนของการเพิ่มข้อมูล และในการปรับปรุงข้อมูล (**Update**) สามารถกำหนดช่วงเวลาที่ต้องการปรับปรุง โดยใช้คำสั่ง **SetValidTime()** จากนั้นสามารถใช้คำสั่งปรับปรุงได้ตามปกติ

การใช้คำสั่ง **SetValidTime()** ซึ่งเป็นคำสั่งที่รองรับข้อมูลด้านเชิงเวลา ทำให้สะดวกในการใช้คำสั่งนักพัฒนาโปรแกรมไม่ต้องกังวลในการตรวจสอบช่วงเวลาทับซ้อนหรือคาบเกี่ยวกัน เมื่อใช้คำสั่ง **SetValidTime()** แล้วสามารถใช้คำสั่ง **Update Delete** ได้ตามปกติ ทั้งนี้สามารถตอบสถิติที่เกี่ยวข้องกับเวลาได้ โดยใช้คำสั่ง **WM_OVERLAPS()**

5.5 การตอบข้อสนทนาศึ่งเวลาประวัติการทำงาน

ประวัติการทำงาน มีอีกส่วนหนึ่งที่มีคำถามข้อสนทนาศึ่งเวลา ที่ระบบฐานข้อมูลแบบเดิม (**non-Temporal**) ไม่สามารถตอบได้ หรือตอบได้ค่อนข้างยุ่งยาก ดังตัวอย่างคำถามเชิงเวลา ดังนี้

“แสดงผู้ดำรงตำแหน่งทางวิชาการ ในรอบ 3 เดือนที่ผ่านมา”

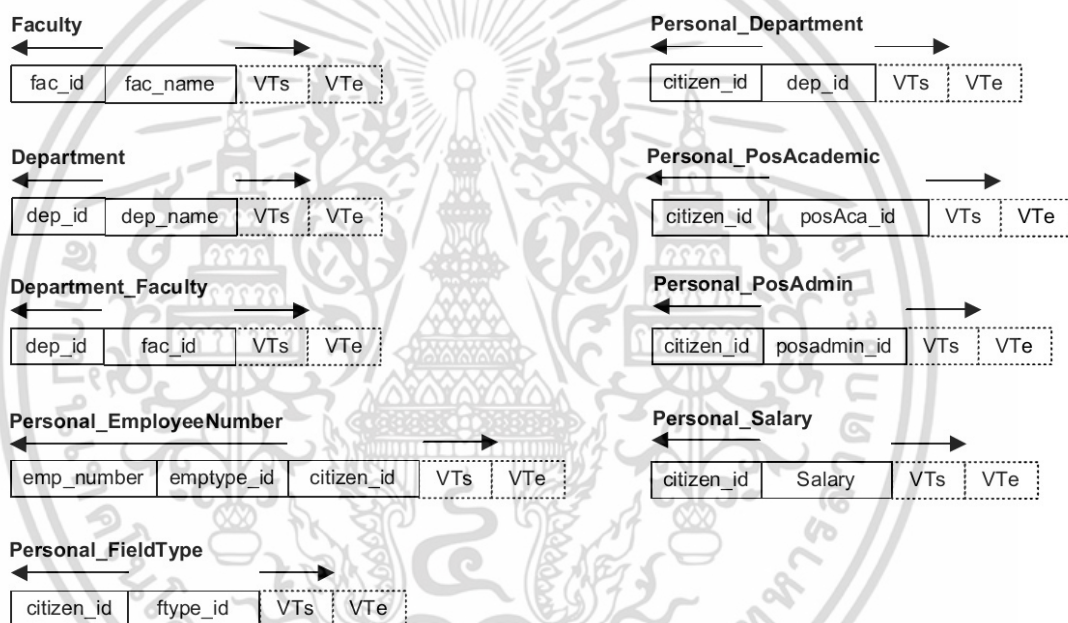
“แสดงจำนวนผู้ดำรงตำแหน่งทางวิชาการในรอบ 1 ปีที่ผ่านมา”

“มีบุคลากรคนไหนบ้างที่ดำรงตำแหน่ง ผู้ช่วยศาสตราจารย์ ในช่วงเวลาเดียวกันกับบุคลากร รหัส 33333 และปัจจุบันยังดำรงตำแหน่ง ผู้ช่วยศาสตราจารย์อยู่ ซึ่งบุคลากร รหัส 33333 ปัจจุบันเขาดำรงตำแหน่ง ศาสตราจารย์ แล้ว”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบฐานข้อมูลแบบเดิมจะตอบคำถามดังกล่าวได้ค่อนข้างยุ่งยาก เพราะไม่มีส่วนบันทึกช่วงเวลาที่แท้จริงของข้อมูล ระบบฐานข้อมูลเชิงเวลาสามารถตอบคำถามดังกล่าวได้ ดังแสดงการทดลองต่อไปนี้

ในส่วนของประวัติการทำงาน จากการออกแบบในบทที่ 4 ข้อที่ 4.5 จะมี Fact อยู่ 9 ประเภท ที่เป็นข้อมูลแบบเชิงเวลา คือ ชื่อคณะ (Faculty) ชื่อแผนก (Department) แผนกสังกัดคณะ (Department_Faculty) เลขอัตราตำแหน่ง (Personal_EmployeeNumber) ประเภทสายการทำงาน (Personal_FiledType) บุคลากรสังกัดแผนก (Personal_Department) ประวัติตำแหน่งทางวิชาการ (Personal_PosAcademic) ประวัติตำแหน่งทางบริหาร (Personal_PosAdmin) ประวัติเงินเดือน (Personal_Salary)



รูปที่ 5.5 โครงร่างตารางแบบเชิงเวลา ด้านประวัติการทำงาน

ข้อมูลที่น่าสนใจในการทดลอง คือ ประวัติตำแหน่งทางวิชาการ ประวัติตำแหน่งทางบริหาร และประวัติเงินเดือน ผู้สนใจสามารถทดลองการบันทึกข้อมูล (Insert) การปรับปรุงข้อมูล (Update) ในส่วนของประวัติการทำงาน ได้ดังตัวอย่างในภาคผนวก ค.

จากการทดลองมีคำถามเชิงเวลา ที่น่าสนใจดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.1 คำถามและคำตอบทางเชิงเวลา ด้านประวัติตำแหน่งทางวิชาการ

5.5.1.1 แสดงประวัติตำแหน่งทางวิชาการของบุคลากรแต่ละท่าน ตั้งแต่อดีต จนถึง

ปัจจุบัน สามารถเขียนคำสั่งได้ ดังนี้

คำสั่งแบบที่ 1

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT personal_posacademic.citizen_id, posacademic.posaca_id,
posacademic.posaca_name,
      TO_CHAR(personal_posacademic.WM_VALID.VALIDFROM, 'DD-MM-
YYYY', 'NLS_CALENDAR=' 'THAI BUDDHA' ' NLS_DATE_LANGUAGE=THAI')
FROM_DATE,
      TO_CHAR(personal_posacademic.WM_VALID.VALIDTILL, 'DD-MM-
YYYY', 'NLS_CALENDAR=' 'THAI BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE

FROM personal_posacademic JOIN posacademic ON
personal_posacademic.posaca_id = posacademic.posaca_id
ORDER BY personal_posacademic.citizen_id ASC ,
personal_posacademic.posaca_id DESC ;
```

คำสั่งแบบที่ 2

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT e.citizen_id, e.posaca_id, pos.posaca_name,
      TO_CHAR(e.WM_VALID.VALIDFROM, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') FROM_DATE,
      TO_CHAR(e.WM_VALID.VALIDTILL, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE

FROM personal_posacademic e , posacademic pos
WHERE e.posaca_id = pos.posaca_id
ORDER BY e.citizen_id ASC , e.posaca_id DESC ;
```

คำสั่งแบบที่ 1 และ 2 ได้ผลลัพธ์ เหมือนกัน ดังนี้

anonymous block completed

1ms elapsed

CITIZEN_ID	POSACA_ID	POSACA_NAME	FROM_DATE	TO_DATE
33333	4	อาจารย์	01-06-2520	01-10-2523
33333	3	ผู้ช่วยศาสตราจารย์	01-10-2523	01-10-2530
33333	2	รองศาสตราจารย์	01-10-2530	01-10-2543
33333	1	ศาสตราจารย์	01-10-2543	
35003	4	อาจารย์	01-06-2525	01-10-2532
35003	3	ผู้ช่วยศาสตราจารย์	01-10-2532	
66666	4	อาจารย์	04-06-2544	
88888	9	นักวิชาการคอมพิวเตอร์	01-06-2550	

8 rows selected

3ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.1.2 แสดงประวัติตำแหน่งทางวิชาการ ณ เวลาปัจจุบัน เขียนคำสั่งได้ ดังนี้

คำสั่งแบบที่ 1

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT personal_posacademic.citizen_id, posacademic.posaca_id,
posacademic.posaca_name,
      TO_CHAR(personal_posacademic.WM_VALID.VALIDFROM, 'DD-MM-
YYYY', 'NLS_CALENDAR=' 'THAI BUDDHA' ' NLS_DATE_LANGUAGE=THAI')
FROM_DATE,
      TO_CHAR(personal_posacademic.WM_VALID.VALIDTILL, 'DD-MM-
YYYY', 'NLS_CALENDAR=' 'THAI BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE

FROM personal_posacademic JOIN posacademic ON
personal_posacademic.posaca_id = posacademic.posaca_id
ORDER BY personal_posacademic.citizen_id ASC ,
personal_posacademic.posaca_id DESC ;
```

คำสั่งแบบที่ 2

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT e.citizen_id, e.posaca_id, pos.posaca_name,
      TO_CHAR(e.WM_VALID.VALIDFROM, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') FROM_DATE,
      TO_CHAR(e.WM_VALID.VALIDTILL, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE

FROM personal_posacademic e , posacademic pos
WHERE e.posaca_id = pos.posaca_id
ORDER BY e.citizen_id ASC , e.posaca_id DESC ;
```

คำสั่งแบบที่ 1 และ 2 ได้ผลลัพธ์ เหมือนกัน ดังนี้

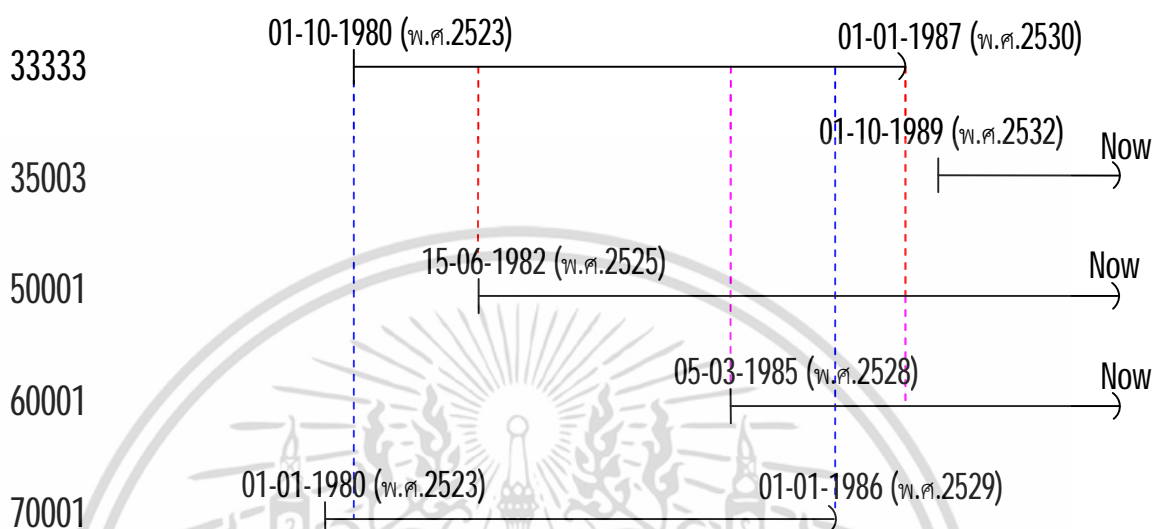
```
anonymous block completed
1ms elapsed
CITIZEN_ID      POSACA_ID      POSACA_NAME      FROM_DATE      TO_DATE
-----
33333          1              ศาสตราจารย์          01-10-2543
35003          3              ผู้ช่วยศาสตราจารย์  01-10-2532
66666          4              อาจารย์              04-06-2544
88888          9              นักวิชาการคอมพิวเตอร์  01-06-2550
```

4 rows selected

4ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.1.3 มีบุคลากรคนไหนบ้าง ดำรงตำแหน่ง ผู้ช่วยศาสตราจารย์ ในช่วงเวลาเดียวกันกับบุคลากรเลขประจำตัวประชาชน 33333 และในปัจจุบันเขายังคงดำรงตำแหน่ง ผู้ช่วยศาสตราจารย์ อยู่ ซึ่งบุคลากรเลขประจำตัวประชาชน 33333 เขาดำรงตำแหน่ง ศาสตราจารย์



รูปที่ 5.6 ช่วงเวลาที่บุคลากร ดำรงตำแหน่ง ผู้ช่วยศาสตราจารย์

จากรูปที่ 5.6 แสดงช่วงเวลา (Time Diagram) ของบุคลากรที่ดำรงตำแหน่ง ผู้ช่วยศาสตราจารย์ จะเห็นว่าบุคลากรเลขประจำตัวประชาชน 35003 ไม่ได้ดำรงตำแหน่ง ผู้ช่วยศาสตราจารย์ ในช่วงเวลาเดียวกันกับบุคลากรเลขประจำตัวประชาชน 33333 เพราะเขาได้รับตำแหน่ง ผู้ช่วยศาสตราจารย์ หลังจากทีบุคลากรเลขประจำตัวประชาชน 33333 สิ้นสุดการดำรงตำแหน่ง ผู้ช่วยศาสตราจารย์ แล้ว

และบุคลากรเลขประจำตัวประชาชน 50001 60001 และ 70001 จะมีช่วงเวลาที่ยังดำรงตำแหน่ง ผู้ช่วยศาสตราจารย์ ที่อยู่ในช่วงเวลาเดียวกันกับบุคลากรรหัส 33333

ตอบคำถามเชิงเวลาข้อ 5.5.1.3

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT e1.citizen_id, e2.citizen_id, pos.posaca_name ,
       TO_CHAR(e2.WM_VALID.VALIDFROM, 'DD-MM-YYYY' , 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') FROM_DATE,
       TO_CHAR(e2.WM_VALID.VALIDTILL, 'DD-MM-YYYY' , 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE
FROM Personal_PosAcademic e1 , Personal_PosAcademic e2 , posacademic
pos
WHERE e1.posaca_id = pos.posaca_id
AND   e1.citizen_id < e2.citizen_id
AND   e1.posaca_id = e2.posaca_id
AND   WM_OVERLAPS(e1.WM_VALID,e2.WM_VALID)=1
AND   e1.posaca_id=3
AND   e1.citizen_id='33333'
AND   e2.WM_VALID.VALIDTILL IS NULL
```

anonymous block completed

CITIZEN_ID	CITIZEN_ID	POSACA_NAME	FROM_DATE	TO_DATE
33333	60001	ผู้ช่วยศาสตราจารย์	05-03-2528	
33333	50001	ผู้ช่วยศาสตราจารย์	15-06-2525	

2 rows selected

5.5.1.4 มีบุคลากรคนไหนบ้าง ดำรงตำแหน่ง ผู้ช่วยศาสตราจารย์ ในช่วงเวลา

เดียวกันกับบุคลากรเลขประจำตัวประชาชน 33333

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT e1.citizen_id, e2.citizen_id, pos.posaca_name ,
       TO_CHAR(e2.WM_VALID.VALIDFROM, 'DD-MM-YYYY' , 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') FROM_DATE,
       TO_CHAR(e2.WM_VALID.VALIDTILL, 'DD-MM-YYYY' , 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE
FROM Personal_PosAcademic e1 , Personal_PosAcademic e2 , posacademic
pos
WHERE e1.posaca_id = pos.posaca_id
AND   e1.citizen_id < e2.citizen_id
AND   e1.posaca_id = e2.posaca_id
AND   WM_OVERLAPS(e1.WM_VALID,e2.WM_VALID)=1
AND   e1.posaca_id=3
AND   e1.citizen_id='33333'
```

anonymous block completed

CITIZEN_ID	CITIZEN_ID	POSACA_NAME	FROM_DATE	TO_DATE
33333	70001	ผู้ช่วยศาสตราจารย์	01-01-2523	01-01-2529
33333	60001	ผู้ช่วยศาสตราจารย์	05-03-2528	
33333	50001	ผู้ช่วยศาสตราจารย์	15-06-2525	

3 rows selected

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.1.5 มีบุคลากรคนไหนบ้าง ที่เคยดำรงตำแหน่งทางวิชาการในช่วงเวลาเดียวกัน

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT e1.citizen_id, e2.citizen_id, pos.posaca_name ,
       TO_CHAR(e2.WM_VALID.VALIDFROM, 'DD-MM-YYYY' , 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') FROM_DATE,
       TO_CHAR(e2.WM_VALID.VALIDTILL, 'DD-MM-YYYY' , 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE
FROM Personal_PosAcademic e1 , Personal_PosAcademic e2 , posacademic
pos
WHERE e1.posaca_id = pos.posaca_id
AND e1.citizen_id < e2.citizen_id
AND e1.posaca_id = e2.posaca_id
AND WM_OVERLAPS(e1.WM_VALID,e2.WM_VALID)=1
ORDER BY e1.citizen_id , e1.posaca_id DESC, e2.citizen_id ,
e2.WM_VALID.VALIDFROM
```

anonymous block completed

CITIZEN_ID	CITIZEN_ID	POSACA_NAME	FROM_DATE	TO_DATE
33333	50001	อาจารย์	02-06-2520	15-06-2525
33333	60001	อาจารย์	24-02-2523	05-03-2528
33333	50001	ผู้ช่วยศาสตราจารย์	15-06-2525	
33333	60001	ผู้ช่วยศาสตราจารย์	05-03-2528	
33333	70001	ผู้ช่วยศาสตราจารย์	01-01-2523	01-01-2529
35003	50001	อาจารย์	02-06-2520	15-06-2525
35003	60001	อาจารย์	24-02-2523	05-03-2528
35003	50001	ผู้ช่วยศาสตราจารย์	15-06-2525	
35003	60001	ผู้ช่วยศาสตราจารย์	05-03-2528	
50001	60001	อาจารย์	24-02-2523	05-03-2528
50001	60001	ผู้ช่วยศาสตราจารย์	05-03-2528	
50001	70001	ผู้ช่วยศาสตราจารย์	01-01-2523	01-01-2529
60001	70001	ผู้ช่วยศาสตราจารย์	01-01-2523	01-01-2529

13 rows selected

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.1.6 คำถามเชิงเวลา แสดงสถิติจำนวนผู้ดำรงตำแหน่งทางวิชาการ ณ เวลา

ปัจจุบัน สามารถเขียนคำสั่งได้ ดังนี้

คำสั่งแบบที่ 1

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT posacademic.posaca_id, posacademic.posaca_name,
Count(personal_posacademic.citizen_id) AS AMOUNT
FROM personal_posacademic JOIN posacademic ON
personal_posacademic.posaca_id = posacademic.posaca_id
GROUP BY posacademic.posaca_id, posacademic.posaca_name
ORDER BY posacademic.posaca_id;
```

คำสั่งแบบที่ 2

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT p.posaca_id, p.posaca_name, Count(e.citizen_id) AS AMOUNT
FROM personal_posacademic e, posacademic p
WHERE e.posaca_id = p.posaca_id
GROUP BY p.posaca_id, p.posaca_name
ORDER BY p.posaca_id;
```

คำสั่งแบบที่ 1 และ 2 ได้ผลลัพธ์ เหมือนกัน ดังนี้

anonymous block completed

2ms elapsed

POSACA_ID	POSACA_NAME	AMOUNT
1	ศาสตราจารย์	1
3	ผู้ช่วยศาสตราจารย์	1
4	อาจารย์	3
9	นักวิชาการคอมพิวเตอร์	1

4 rows selected

3ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.1.7 คำถามเชิงเวลา แสดงสถิติจำนวนผู้ดำรงตำแหน่งทางวิชาการ ในช่วงวันที่

1 ม.ค. 2000 ถึง 1 มิ.ย. 2000 สามารถเขียนคำสั่งได้ ดังนี้

คำสั่งแบบที่ 1

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('01-01-2000','DD-MM-YYYY'),
TO_DATE('01-06-2000','DD-MM-YYYY'));
```

```
SELECT posacademic.posaca_id, posacademic.posaca_name,
Count(personal_posacademic.citizen_id) AS AMOUNT
FROM personal_posacademic JOIN posacademic ON
personal_posacademic.posaca_id = posacademic.posaca_id
GROUP BY posacademic.posaca_id, posacademic.posaca_name
ORDER BY posacademic.posaca_id;
```

คำสั่งแบบที่ 2

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('01-01-2000','DD-MM-YYYY'),
TO_DATE('01-06-2000','DD-MM-YYYY'));
```

```
SELECT p.posaca_id, p.posaca_name, Count(e.citizen_id) AS AMOUNT
FROM personal_posacademic e, posacademic p
WHERE e.posaca_id = p.posaca_id
GROUP BY p.posaca_id, p.posaca_name
ORDER BY p.posaca_id;
```

คำสั่งแบบที่ 1 และ 2 ได้ผลลัพธ์ เหมือนกัน ดังนี้

POSACA_ID	POSACA_NAME	AMOUNT
2	รองศาสตราจารย์	1
3	ผู้ช่วยศาสตราจารย์	3

2 rows selected

3ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.2 คำถามและคำตอบทางเชิงเวลา ด้านประวัติตำแหน่งทางบริหาร

5.5.2.1 ตอบคำถามเชิงเวลา เพื่อแสดงตำแหน่งบริหารของบุคลากรทุกคน ตั้งแต่

อดีต จนถึงปัจจุบัน สามารถเขียนคำสั่งได้ ดังนี้

คำสั่งแบบที่ 1

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT personal_posadmin.citizen_id, personal_posadmin.posadmin_id,
posadmin.posadmin_name,
       TO_CHAR(personal_posadmin.WM_VALID.VALIDFROM, 'DD-MM-
YYYY', 'NLS_CALENDAR=' 'THAI BUDDHA' ' NLS_DATE_LANGUAGE=THAI')
FROM_DATE,
       TO_CHAR(personal_posadmin.WM_VALID.VALIDTILL, 'DD-MM-
YYYY', 'NLS_CALENDAR=' 'THAI BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE

FROM personal_posadmin JOIN posadmin ON personal_posadmin.posadmin_id
= posadmin.posadmin_id
ORDER BY personal_posadmin.citizen_id,
personal_posadmin.WM_VALID.VALIDFROM;
```

คำสั่งแบบที่ 2

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT e.citizen_id, e.posadmin_id, p.posadmin_name,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE
FROM personal_posadmin e , posadmin p
WHERE e.posadmin_id = p.posadmin_id
ORDER BY e.citizen_id, e.WM_VALID.VALIDFROM;
```

คำสั่งแบบที่ 1 และ 2 ได้ผลลัพธ์ เหมือนกัน ดังนี้

anonymous block completed
2ms elapsed

CITIZEN_ID	POSADMIN_ID	POSADMIN_NAME	FROM_DATE	TO_DATE
33333	82	หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์	01-06-2525	
33333	7	รองคณบดีคณะวิศวกรรมศาสตร์	01-06-2529	01-06-2533
33333	4	คณบดีคณะวิศวกรรมศาสตร์	01-06-2533	01-06-2537
33333	2	รองอธิการบดี	01-06-2543	01-06-2547
33333	1	อธิการบดี	01-06-2555	
35003	81	หัวหน้าภาควิชาวิศวกรรมไฟฟ้า	01-06-2525	

6 rows selected

90ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.2.2 ตอบคำถามเชิงเวลา เพื่อแสดงตำแหน่งบริหารของบุคลากรทุกคน ณ เวลา

ปัจจุบัน สามารถเขียนคำสั่งได้ ดังนี้

คำสั่งแบบที่ 1

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT personal_posadmin.citizen_id, personal_posadmin.posadmin_id,
posadmin.posadmin_name,
       TO_CHAR(personal_posadmin.WM_VALID.VALIDFROM, 'DD-MM-
YYYY', 'NLS_CALENDAR=' 'THAI BUDDHA' ' NLS_DATE_LANGUAGE=THAI ')
FROM_DATE,
       TO_CHAR(personal_posadmin.WM_VALID.VALIDTILL, 'DD-MM-
YYYY', 'NLS_CALENDAR=' 'THAI BUDDHA' ' NLS_DATE_LANGUAGE=THAI ') TO_DATE

FROM personal_posadmin JOIN posadmin ON personal_posadmin.posadmin_id
= posadmin.posadmin_id
ORDER BY personal_posadmin.citizen_id,
personal_posadmin.WM_VALID.VALIDFROM;
```

คำสั่งแบบที่ 2

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT e.citizen_id, e.posadmin_id, p.posadmin_name,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI ') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI ') TO_DATE
FROM personal_posadmin e , posadmin p
WHERE e.posadmin_id = p.posadmin_id
ORDER BY e.citizen_id, e.WM_VALID.VALIDFROM;
```

คำสั่งแบบที่ 1 และ 2 ได้ผลลัพธ์ เหมือนกัน ดังนี้

```
anonymous block completed
2ms elapsed
CITIZEN_ID   POSADMIN_ID   POSADMIN_NAME   FROM_DATE   TO_DATE
-----
33333        82            หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์   01-06-2525
33333        1             อธิการบดี      01-06-2555
35003        81            หัวหน้าภาควิชาวิศวกรรมไฟฟ้า         01-06-2525
```

3 rows selected

4ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.3 คำถามและคำตอบทางเชิงเวลา ด้านประวัติเงินเดือน

5.5.3.1 แสดงประวัติเงินเดือนของบุคลากรทุกคน ตั้งแต่อดีต จนถึงปัจจุบัน

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT e.CITIZEN_ID, e.salary,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE

FROM   personal_salary e
ORDER BY e.citizen_id ASC, e.WM_VALID.VALIDFROM ;
```

anonymous block completed

2ms elapsed

CITIZEN_ID	SALARY	FROM_DATE	TO_DATE
33333	15000	01-06-2520	01-06-2523
33333	20000	01-06-2523	01-06-2528
33333	25000	01-06-2528	01-06-2533
33333	30000	01-06-2533	01-06-2538
33333	35000	01-06-2538	01-06-2543
33333	40000	01-06-2543	01-06-2548
33333	45000	01-06-2548	01-06-2553
33333	50000	01-06-2553	
35003	15000	01-06-2525	01-06-2528
35003	20000	01-06-2528	01-06-2533
35003	23000	01-06-2533	01-06-2538
35003	26000	01-06-2538	01-06-2543
35003	29000	01-06-2543	01-06-2548
35003	32000	01-06-2548	01-06-2553
35003	35000	01-06-2553	
66666	15000	04-06-2544	01-06-2548
66666	18000	01-06-2548	01-06-2553
66666	20000	01-06-2553	
88888	15000	01-06-2550	01-06-2553
88888	18000	01-06-2553	

20 rows selected

15ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.3.2 แสดงประวัติเงินเดือนของบุคลากรทุกคน ณ เวลาปัจจุบัน

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT e.CITIZEN_ID, e.salary,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE

FROM   personal_salary e
ORDER BY e.citizen_id ASC, e.WM_VALID.VALIDFROM ;
```

anonymous block completed

2ms elapsed

CITIZEN_ID	SALARY	FROM_DATE	TO_DATE
33333	50000	01-06-2553	
35003	35000	01-06-2553	
66666	20000	01-06-2553	
88888	18000	01-06-2553	

4 rows selected

5ms elapsed

5.5.4 สรุปผลการทดลองเพื่อตอบข้อสงสัยเรื่องเวลาด้านประวัติการทำงาน

จากการทดลอง สามารถเพิ่มข้อมูลประวัติการทำงาน (**Insert**) ตามช่วงเวลาที่ต้องการได้ โดยระบุช่วงเวลาเข้าไปในส่วนของการเพิ่มข้อมูล และในการปรับปรุงข้อมูล (**Update**) สามารถกำหนดช่วงเวลาที่ต้องการปรับปรุง โดยใช้คำสั่ง **SetValidTime()** จากนั้นสามารถใช้คำสั่งปรับปรุงได้ตามปกติ

การใช้คำสั่ง **SetValidTime()** ซึ่งเป็นคำสั่งที่รองรับข้อมูลด้านเชิงเวลา ทำให้สะดวกในการใช้คำสั่งนักพัฒนาโปรแกรมไม่ต้องกังวลในการตรวจสอบช่วงเวลาทับซ้อนหรือคาบเกี่ยวกัน เมื่อใช้คำสั่ง **SetValidTime()** แล้วสามารถใช้คำสั่ง **Update Delete** ได้ตามปกติ ทั้งนี้สามารถตอบสถิติที่เกี่ยวข้องกับเวลาได้ โดยใช้คำสั่ง **WM_OVERLAPS()**

5.6 การยกเลิกตารางแบบเชิงเวลา และการลบตารางทั้งหมด

เนื่องจากการทดลอง ตั้งแต่ข้อ 5.1-5.5 มีข้อมูลที่ต้องทดสอบเป็นจำนวนมาก ดังเอกสาร ภาคผนวก ก ภาคผนวก ข ภาคผนวก ค และ ภาคผนวก ง ถ้าผู้ที่สนใจต้องการลบข้อมูลและทุกตารางเพื่อเริ่มต้นการทดลองใหม่อีกครั้ง จะต้องทำการยกเลิกตารางแบบเชิงเวลาให้เป็นตารางแบบปกติก่อน ด้วยคำสั่ง **DisableVersioning** จึงจะสามารถลบตารางเชิงเวลาด้วยคำสั่ง **Drop table** ได้ สิ่งที่น่าพิศวงคือ เมื่อทำการยกเลิกตารางเชิงเวลาเป็นตารางปกติแล้ว ประวัติข้อมูลต่างๆ จะถูกลบได้

คำสั่งที่ใช้ยกเลิกตารางแบบเชิงเวลา และลบตารางต่างๆ ดังนี้

```
EXECUTE DBMS_WM.DisableVersioning('personal_titlecode');
EXECUTE DBMS_WM.DisableVersioning('personal_fname');
EXECUTE DBMS_WM.DisableVersioning('personal_lname');

EXECUTE DBMS_WM.DisableVersioning('province_name');
EXECUTE DBMS_WM.DisableVersioning('district_name');
EXECUTE DBMS_WM.DisableVersioning('district_province');
EXECUTE DBMS_WM.DisableVersioning('subdistrict_name');
EXECUTE DBMS_WM.DisableVersioning('subdistrict_district');
EXECUTE DBMS_WM.DisableVersioning('road_name');
EXECUTE DBMS_WM.DisableVersioning('address_subdistrict');
EXECUTE DBMS_WM.DisableVersioning('personal_address');

EXECUTE DBMS_WM.DisableVersioning('country_name');
EXECUTE DBMS_WM.DisableVersioning('institution_name');

EXECUTE DBMS_WM.DisableVersioning('faculty');
EXECUTE DBMS_WM.DisableVersioning('department');
EXECUTE DBMS_WM.DisableVersioning('department_faculty');
EXECUTE DBMS_WM.DisableVersioning('personal_employeenumber');
EXECUTE DBMS_WM.DisableVersioning('personal_fieldtype');
EXECUTE DBMS_WM.DisableVersioning('personal_department');
EXECUTE DBMS_WM.DisableVersioning('personal_posacademic');
EXECUTE DBMS_WM.DisableVersioning('personal_posadmin');
EXECUTE DBMS_WM.DisableVersioning('personal_salary');

drop table personal;
drop table blood;
drop table titlecode;
drop table personal_titlecode;
drop table personal_fname;
drop table personal_lname;

drop table province_name;
drop table district_name;
drop table district_province;
drop table subdistrict_name;
drop table subdistrict_district;
drop table road_name;
drop table address;
drop table address_subdistrict;
drop table personal_address;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
drop table personal_education;  
drop table degree_level;  
drop table qualification;  
drop table major;  
drop table institution_country;  
drop table institution_name;  
drop table country_name;
```

```
drop table fieldType;  
drop table posadmin;
```

```
drop table employeetype;  
drop table faculty;  
drop table department;  
drop table department_faculty;  
drop table posacademic;  
drop table personal_employeenumber;  
drop table personal_fieldtype;  
drop table personal_department;  
drop table personal_posacademic;  
drop table personal_posadmin;  
drop table personal_salary;
```

หมายเหตุ การยกเลิกตารางเชิงเวลา 22 ตารางใช้เวลาประมาณ 13 วินาที เฉลี่ยตารางละ 600 ms

ทั้งนี้เวลาความเร็วขึ้นกับสมรรถนะของเครื่องคอมพิวเตอร์ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

6.1 สรุปผลการวิจัย

6.1.1 ในงานวิจัยนี้ได้พัฒนาระบบสารสนเทศเชิงเวลาโดยใช้ระบบจัดการฐานข้อมูลเชิงเวลา ทั้งนี้ได้ใช้ระบบงานบุคลากรเป็นกรณีศึกษา ซึ่งขั้นตอนการออกแบบฐานข้อมูล ได้ใช้แบบจำลองที่โออาร์เอ็ม (TORM) และสามารถแปลงเป็นโครงสร้างตารางได้ทั้งหมด 35 ตาราง แบ่งเป็นตารางแบบเชิงสัมพันธ์ 13 ตาราง และตารางแบบเชิงเวลา 22 ตาราง

6.1.2 จากการทดลองการปฏิบัติการเข้าถึงข้อมูลที่รองรับเชิงเวลา ผ่าน Workspace Manager ของ Oracle11g พบว่า สามารถบันทึกข้อมูล (Insert) สามารถปรับปรุงข้อมูล (Update) สามารถลบข้อมูล (Delete) และสามารถสอบถามข้อมูล (Select) ในลักษณะของเชิงเวลา ณ ช่วงเวลาปัจจุบัน เวลาในอดีต และเวลาในอนาคตได้ถูกต้องตามทฤษฎีของฐานข้อมูลเชิงเวลา [3]

6.1.3 จากการทดลองการปฏิบัติการเข้าถึงข้อมูลที่รองรับเชิงเวลา ผ่าน Workspace Manager ของ Oracle11g พบว่า จำนวนคำสั่งในการเข้าถึงข้อมูลน้อยกว่า SQL มาตรฐาน เช่น การปรับปรุงข้อมูลแบบเชิงเวลา ถ้าใช้คำสั่ง SQL มาตรฐาน จะใช้คำสั่ง 5 คำสั่ง แต่ถ้าเป็นคำสั่งผ่าน Workspace Manager ของ Oracle11g จะใช้คำสั่งเพียง 2 คำสั่ง คือ กำหนด Valid Time และใช้คำสั่ง Update ได้ตามปกติ ทำให้ประหยัดเวลาและง่ายต่อการพัฒนา

6.1.4 ในการออกแบบฐานข้อมูลเชิงเวลา จำเป็นต้องใช้แบบจำลองที่รองรับฐานข้อมูลเชิงเวลาในระดับแอทริบิวต์ และสามารถแปลงรูปแบบตารางรีเลชันเชิงเวลาออกมาอย่างชัดเจน เช่น TORM (อ้างอิงคำอธิบาย : รศ. ดร. ศุภมิตร จิตตะยโสธร) หากไม่ใช้แบบจำลองดังกล่าว อาจเกิดปัญหา 2 ตัวอย่าง ดังนี้

6.1.4.1 คอลัมน์ของช่วงระยะเวลากำกับแต่ละ Entity Type

รูปที่ 6.1 แสดงโครงสร้างตารางที่คอลัมน์ช่วงระยะเวลากำกับ Entity Type และข้อมูลสมมุติตัวอย่าง จากข้อมูลในตาราง จะทราบเวลาที่เปลี่ยนแปลงข้อมูลของแต่ละแถวข้อมูล แต่ไม่ทราบว่าข้อเท็จจริงใดเปลี่ยนแปลง คือ ไม่ทราบเวลาเปลี่ยนแปลงค่านำหน้าเมื่อใด ไม่ทราบเวลาเปลี่ยนแปลงชื่อเมื่อใด ไม่ทราบเวลาเปลี่ยนแปลงนามสกุลเมื่อใด เพราะมีช่วงระยะเวลาที่สัมพันธ์กับ Entity Type เพียงค่าเดียว

Personal_Names

citizen_id	title_id	Fname	Lname	VTs	VTe
88888	น.ส.	กนกพร	บุญรักษา	15-07-1992	20-02-2008
88888	นาง	กนกพร	ถนอมรัก	20-02-2008	26-05-2013
88888	น.ส.	สมปรัตนา	บุญรักษา	26-05-2013	Now

รูปที่ 6.1 โครงร่างตารางที่คอลัมน์ช่วงระยะเวลากำกับแต่ละ Entity Type และข้อมูลตัวอย่าง

6.1.4.2 คอลัมน์ของช่วงระยะเวลากำกับ Attribute โดยไม่แยก Fact

รูปที่ 6.2 แสดงโครงร่างตารางที่คอลัมน์ช่วงระยะเวลาสัมพันธ์กับ Attribute โดยไม่แยก Fact ผลที่เกิดขึ้นจะเป็นลักษณะข้อมูลแบบ Object หรือ Repeating Groups ทำให้การ Insert Update หรือ Delete ได้ยาก และถ้าแปลงรูปโครงร่างตาราง จะยิ่งซับซ้อนเข้าใจมากยิ่งขึ้น เพราะจะกลายเป็นจำนวนแถวข้อมูลซ้ำ (Repeating Groups) และไม่ใช้ฐานข้อมูลเชิงสัมพันธ์

Personal_Names

citizen_id	title_id	VTs	VTe	Fname	VTs	VTe	Lname	VTs	VTe			
88888	น.ส.	26-05-2013	Now	กนกพร	15-07-1992	26-05-2013	บุญรักษา	26-05-2013	Now			
	นาง	20-02-2008	26-05-2013							ถนอมรัก	20-02-2008	26-05-2013
	น.ส.	15-07-1992	20-02-2008							บุญรักษา	15-07-1992	20-02-2008

รูปที่ 6.2 โครงร่างตารางที่คอลัมน์ช่วงระยะเวลากำกับ Attribute และข้อมูลตัวอย่าง

6.1.5 การดำเนินการวิจัยนี้ เป็นการพัฒนาระบบสารสนเทศเชิงเวลาต้นแบบ และเป็นการใช้คำสั่งเพื่อปฏิบัติการเข้าถึงข้อมูลเชิงเวลา ซึ่งยังไม่พบว่า มีใครพัฒนาระบบสารสนเทศเชิงเวลาและปฏิบัติการคำสั่งเชิงเวลาพร้อมทั้งรายงานการออกแบบและพัฒนาโดยละเอียด โดยเฉพาะอย่างยิ่งการออกแบบโดยใช้ **TORM** มาก่อน อีกทั้งการใช้คำสั่งต่างๆ ตามที่ได้ทดลองนั้นพบว่ามีความสะดวกและสะดวกต่อการพัฒนาระบบสารสนเทศแบบเชิงเวลา ในโอกาสต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 ข้อเสนอแนะ

ระบบสารสนเทศเชิงเวลาที่น่าศึกษาค้นคว้าและพัฒนาเพิ่มเติมในโอกาสต่อไป เช่น การพัฒนาระบบสารสนเทศเชิงเวลาที่เกี่ยวข้องกับภูมิศาสตร์ การพัฒนาระบบสารสนเทศเชิงเวลาที่เกี่ยวข้องกับภาพและเสียง เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] Christian S. Jensen, James Clifford, Ramez Elmasri, Shashi K. Gadia, Pat Hayes and Sushil Jajodia editor. "A Glossary of Temporal Database Concepts" **ACM SIGMOD Record**. vol. 23, no. 1, March. 1994. pp. 52-64
- [2] สุรียพร สร้อยสังวาลย์. 2552. "ระบบฐานข้อมูลพีชซีเชิงเวลาที่ใช้พจน์พีชซีเชิงเวลาและภาษา TFSQL." วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- [3] Richard T. Snodgrass. 1998. "**Managing Temporal Data A Five-Part Series.**" A Time Center Technical Report TR-28.
- [4] X. Sean Wang, Claudio Bettini, Alexander Brodsky and Suchil Jajodia. 1997. "**Logical Design for Temporal Database with Multiple Temporal Types.**" ISSE Technical Report : ISSE-TR-94-111.
- [5] X. SEAN WANG, Claudio Bettini, Alexander Brodsky and Suchil Jajodia. "Logical Design for Temporal Databases with Multiple Granularities." **ACM Transactions on Database Systems**, vol. 22, no. 2, June. 1997. pp. 115-170
- [6] A. Segev and A. Shoshani. "The Representation of A Temporal Data Model in the Relational Environment." **Springer Statistical and Scientific Database Management Lecture Notes in Computer Science**, vol. 339, 1989. pp. 39-61
- [7] Terry Halpin. "Temporal Modeling and ORM." **Springer-Verlag Berlin Heidelberg**, vol. 5333, 2008. pp. 688-698
- [8] พุสดี พรผล และศุภมิตร จิตตะยโสธร. "วิธีการออกแบบฐานข้อมูลเชิงเวลา A Temporal Database Design Methodology" วิศวกรรมสาร มก, ปี 2546, ฉบับที่ 48, ธันวาคม 2545-มีนาคม 2546. หน้า 141-153
- [9] Putsadee Pornphol and Suphamit Chittayasothor. "A Temporal Relational and Object Relational Databae Design Technique." **IEEE SoutheastCon**, 2004. pp. 54-59
- [10] Chuck Murray. 2009. "**Workspace Manager Developer's Guide 11gRelease 2 (11.2).**" Oracle11g E11826-02.



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ก.
การทดลองบันทึกแก้ไขข้อมูลด้านประวัติชื่อ-นามสกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

การทดลองบันทึกแก้ไขข้อมูลด้านประวัติชื่อ-นามสกุล

การทดลองบันทึกประวัติชื่อ-นามสกุล นี้ จะสมมุติบุคคลากรชื่อ น.ส.กนกพร บุญรักษา เลขประจำตัวประชาชน 88888 เข้าบรรจุทำงานเมื่อวันที่ 15 กรกฎาคม 2535 ต่อมาคุณกนกพร ได้สมรสและแจ้งเปลี่ยนนามสกุลตามสามี ชื่อ นางกนกพร ถนอมรัก เมื่อวันที่ 20 กุมภาพันธ์ 2551 ต่อมาในภายหลังกคุณกนกพร ได้หย่าร้างกับสามีเมื่อ 26 พฤษภาคม 2556 แจ้งขอเปลี่ยนชื่อและกลับมาใช้นามสกุลเดิม เป็น น.ส.สมปรารถนา บุญรักษา

1. สร้างตาราง personal blood titlecode personal_titlecode personal_fname personal_lname

--1)create personal table

```
CREATE TABLE personal(
  citizen_id          VARCHAR(13) PRIMARY KEY,
  birthday            DATE,
  gender              VARCHAR(1),
  blood_id            VARCHAR(2),
  father_fname       VARCHAR(30),
  father_Lname       VARCHAR(30),
  mother_fname       VARCHAR(30),
  mother_Lname       VARCHAR(30));
```

--2)create blood table

```
CREATE TABLE blood(
  blood_id            VARCHAR(2) PRIMARY KEY,
  blood_th            VARCHAR(10));
```

--3)create titlecode table

```
CREATE TABLE titlecode(
  title_id            NUMBER PRIMARY KEY,
  title_th            VARCHAR(100),
  title_th_abbr       VARCHAR(30),
  title_en            VARCHAR(100),
  title_en_abbr       VARCHAR(30));
```

--4)create personal_titlecode table

```
CREATE TABLE personal_titlecode(
  citizen_id          VARCHAR(13) PRIMARY KEY,
  title_id            NUMBER );
```

--5)create personal_fname table

```
CREATE TABLE personal_fname(
  citizen_id          VARCHAR(13) PRIMARY KEY,
  fname              VARCHAR(30));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

--6)create personal_lname table

```
CREATE TABLE personal_lname (
  citizen_id      VARCHAR2(13) PRIMARY KEY,
  lname           VARCHAR2(30));
commit;
```

2. กำหนดให้ตาราง personal_titlecode personal_fname personal_lname เป็นแบบ เชิงเวลา

```
EXECUTE DBMS_WM.EnableVersioning('personal_titlecode',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

EXECUTE DBMS_WM.EnableVersioning('personal_fname',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

EXECUTE DBMS_WM.EnableVersioning('personal_lname',
'VIEW_WO_OVERWRITE', FALSE, TRUE);
commit;
```

3. เพิ่มข้อมูลเบื้องต้นในตาราง PERSONAL และตาราง TITLECODE

--1) เพิ่มข้อมูลค่านำหน้าในตาราง titlecode

```
INSERT INTO PERSONAL VALUES ('33333', TO_DATE('23-04-
1991', 'DD-MM-YY'), 'M', 'A', 'ประทีป', 'อุดรธรรม', 'ปราวณี', 'อุดรธรรม');
INSERT INTO PERSONAL VALUES ('35003', TO_DATE('23-04-
1991', 'DD-MM-YY'), 'M', 'A', 'มงคล', 'ทตเจริญ', 'จรรยา', 'ทตเจริญ');
INSERT INTO PERSONAL VALUES ('66666', TO_DATE('28-02-
1976', 'DD-MM-YY'), 'M', 'O', 'บุญทัน', 'รักถนอม', 'อนงค์', 'ไผ่ชู');
INSERT INTO PERSONAL VALUES ('88888', TO_DATE('03-05-
1992', 'DD-MM-YY'), 'W', 'B', 'กิตติรัก', 'บุญรักษา', 'ดวงใจ', 'บุญรักษา');
```

--2) เพิ่มข้อมูลค่านำหน้าในตาราง titlecode

```
INSERT INTO titlecode VALUES(1000, 'เด็กชาย', 'ด.ช.', 'Master', 'Mstr.'
);
INSERT INTO titlecode VALUES(1001, 'เด็กหญิง', 'ด.ญ.', 'Miss', 'Ms.'
);
INSERT INTO titlecode VALUES(1002, 'นาย', 'นาย', 'Mister', 'Mr.'
);
INSERT INTO titlecode VALUES(1003, 'นาง', 'นาง', 'Missis',
'Mrs.');
```

```
INSERT INTO titlecode VALUES(1004, 'นางสาว', 'น.ส.', 'Miss', 'Ms.'
);
```

```
commit;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทดลองเพิ่มข้อมูล น.ส.กนกพร บุญรักษา เลขประจำตัวประชาชน 88888 เข้าทำงานวันที่ 15 กรกฎาคม 2535 (15-07-1992) โดยเพิ่มข้อมูลในตาราง PERSONAL_TITLECODE PERSONAL_FNAME และ PERSONAL_LNAME ตามลำดับดังนี้

```

INSERT INTO personal_titlecode VALUES(
  '88888',1004,
  WMSYS.WM_PERIOD(TO_DATE('15-07-1992','DD-MM-YYYY'),
    DBMS_WM.MAX_TIME ));

INSERT INTO personal_fname VALUES(
  '88888','กนกพร',
  WMSYS.WM_PERIOD(TO_DATE('15-07-1992','DD-MM-YYYY'),
    DBMS_WM.MAX_TIME ));

INSERT INTO personal_lname VALUES(
  '88888','บุญรักษา',
  WMSYS.WM_PERIOD(TO_DATE('15-07-1992','DD-MM-YYYY'),
    DBMS_WM.MAX_TIME ));

commit;

```

5. แสดงข้อมูลชื่อ-นามสกุลของ น.ส.กนกพร บุญรักษา เลขประจำตัวประชาชน 88888 ณ เวลาปัจจุบัน

```

EXECUTE DBMS_WM.SetValidTime( );

SELECT title_th_abbr, fname, lname
FROM ((personal JOIN (personal_titlecode JOIN titlecode ON
personal_titlecode.title_id = titlecode.title_id) ON
personal.citizen_id = personal_titlecode.citizen_id)
JOIN personal_fname ON personal.citizen_id =
personal_fname.citizen_id)
JOIN personal_lname ON personal.citizen_id =
personal_lname.citizen_id
WHERE personal.citizen_id='88888';

```

TITLE_TH_ABBR	FNAME	LNAME
น.ส.	กนกพร	บุญรักษา

1 rows selected
4ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. น.ส.กนกพร บุญรักษา ได้แต่งงาน วันที่ 20 กุมภาพันธ์ 2551 (20-02-2008) จึงแจ้งขอเปลี่ยนคำนำหน้านามและนามสกุลตามสามีเป็น นางกนกพร ฌนอมรัก

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('20-02-2008','DD-MM-YYYY'), DBMS_WM.MAX_TIME);

UPDATE personal_titlecode SET title_id =1003 WHERE citizen_id = '88888';

UPDATE personal_lname SET lname = 'ฌนอมรัก' WHERE citizen_id = '88888';

commit;
```

7. ตรวจสอบดูการเปลี่ยนแปลงค่าในตาราง PERSONAL_TITLECODE

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT citizen_id, title_id,
       TO_CHAR(e.WM_VALID.VALIDFROM,'DD-MM-YYYY') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL,'DD-MM-YYYY') TO_DATE
FROM personal_titlecode e
-----
CITIZEN_ID    TITLE_ID  FROM_DATE    TO_DATE
-----
88888         1003     20-02-2008   31-12-9999
88888         1004     15-07-1992   20-02-2008

2 rows selected
```

8. ตรวจสอบดูการเปลี่ยนแปลงค่าในตาราง PERSONAL_FNAME

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT citizen_id, fname,
       TO_CHAR(e.WM_VALID.VALIDFROM,'DD-MM-YYYY') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL,'DD-MM-YYYY') TO_DATE
FROM personal_fname e
-----
CITIZEN_ID    FNAME      FROM_DATE    TO_DATE
-----
88888         กนกพร     15-07-1992   31-12-9999

1 rows selected
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. ตรวจสอบดูการเปลี่ยนแปลงค่าในตาราง PERSONAL_LNAME

```
EXECUTE
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT citizen_id, lname,
       TO_CHAR(e.WM_VALID.VALIDFROM,'DD-MM-YYYY') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL,'DD-MM-YYYY') TO_DATE
FROM personal_lname e
```

CITIZEN_ID	LNAME	FROM_DATE	TO_DATE
88888	ถนนมรก	20-02-2008	31-12-9999
88888	บุญรักษา	15-07-1992	20-02-2008

2 rows selected

10. ตรวจสอบประวัติชื่อ-นามสกุล ณ ช่วงเวลาปัจจุบัน หลังจากที่คุณกนกพร ได้แต่งงานแล้ว

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT title_th_abbr, fname, lname
FROM ((personal JOIN (personal_titlecode JOIN titlecode ON
personal_titlecode.title_id = titlecode.title_id) ON
personal.citizen_id = personal_titlecode.citizen_id)
JOIN personal_fname ON personal.citizen_id =
personal_fname.citizen_id)
JOIN personal_lname ON personal.citizen_id =
personal_lname.citizen_id
WHERE personal.citizen_id='88888';
```

TITLE_TH_ABBR	FNAME	LNAME
นาง	กนกพร	ถนนมรก

1 rows selected

11. นางกนกพร ถนนมรก (88888) ได้หย่ากับสามี วันที่ 26 พฤษภาคม 2556 (26-05-2013)

จึงแจ้งขอเปลี่ยนชื่อและกลับกลับมาใช้นามสกุลเดิม เป็น น.ส.สมปรารถนา บุญรักษา

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('26-05-2013','DD-MM-
YYYY'), DBMS_WM.MAX_TIME);
```

```
UPDATE personal_titlecode SET title_id =1004
WHERE citizen_id ='88888';
```

```
UPDATE personal_fname SET fname ='สมปรารถนา'
WHERE citizen_id ='88888';
```

```
UPDATE personal_lname SET lname ='บุญรักษา'
WHERE citizen_id ='88888';
commit;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. ตรวจสอบประวัติชื่อ-นามสกุล ณ ช่วงเวลาปัจจุบัน หลังจากที่คุณกนกพร ได้หย่าแล้ว

-- แบบที่ 1

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT title_th_abbr, fname, lname
FROM ((personal JOIN (personal_titlecode JOIN titlecode ON
personal_titlecode.title_id = titlecode.title_id) ON
personal.citizen_id = personal_titlecode.citizen_id)
JOIN personal_fname ON personal.citizen_id =
personal_fname.citizen_id)
JOIN personal_lname ON personal.citizen_id =
personal_lname.citizen_id
WHERE personal.citizen_id='88888';
```

-- แบบที่ 2

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT title_th_abbr, fname, lname
FROM personal, (personal_titlecode JOIN titlecode ON
personal_titlecode.title_id = titlecode.title_id), personal_fname,
personal_lname
WHERE personal.citizen_id = personal_titlecode.citizen_id
AND personal.citizen_id = personal_fname.citizen_id
AND personal.citizen_id = personal_lname.citizen_id
AND personal.citizen_id='88888';
```

TITLE_TH_ABBR	FNAME	LNAME
น.ส.	สมปรารถนา	บุญรักษา

1 rows selected

13. ตรวจสอบประวัติชื่อ-นามสกุล ณ ช่วงเวลา 2 ปีแรกที่เข้าทำงาน

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('1992-07-15','YYYY-MM-DD'),
TO_DATE('1994-07-15','YYYY-MM-DD'));
```

```
SELECT title_th_abbr, fname, lname
FROM ((personal JOIN (personal_titlecode JOIN titlecode ON
personal_titlecode.title_id = titlecode.title_id) ON
personal.citizen_id = personal_titlecode.citizen_id)
JOIN personal_fname ON personal.citizen_id =
personal_fname.citizen_id)
JOIN personal_lname ON personal.citizen_id =
personal_lname.citizen_id
WHERE personal.citizen_id='88888';
```

TITLE_TH_ABBR	FNAME	LNAME
น.ส.	กนกพร	บุญรักษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14. ดูข้อมูลตาราง personal_titlecode personal_fname personal_lname ของพนักงานรหัส 88888

```

EXECUTE
DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT citizen_id, title_id,
       TO_CHAR(e.WM_VALID.VALIDFROM,'DD-MM-YYYY') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL,'DD-MM-YYYY') TO_DATE
FROM personal_titlecode e WHERE citizen_id= '88888'
ORDER BY e.WM_VALID.VALIDFROM;

SELECT citizen_id, fname,
       TO_CHAR(e.WM_VALID.VALIDFROM,'DD-MM-YYYY') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL,'DD-MM-YYYY') TO_DATE
FROM personal_fname e WHERE citizen_id= '88888'
ORDER BY e.WM_VALID.VALIDFROM;

SELECT citizen_id, lname,
       TO_CHAR(e.WM_VALID.VALIDFROM,'DD-MM-YYYY') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL,'DD-MM-YYYY') TO_DATE
FROM personal_lname e WHERE citizen_id= '88888'
ORDER BY e.WM_VALID.VALIDFROM;

```

CITIZEN_ID	TITLE_ID	FROM_DATE	TO_DATE
88888	1004	15-07-1992	20-02-2008
88888	1003	20-02-2008	26-05-2013
88888	1004	26-05-2013	31-12-9999

3 rows selected

CITIZEN_ID	FNAME	FROM_DATE	TO_DATE
88888	กนกพร	15-07-1992	26-05-2013
88888	สมปราวรณา	26-05-2013	31-12-9999

2 rows selected

CITIZEN_ID	LNAME	FROM_DATE	TO_DATE
88888	บุญรักษา	15-07-1992	20-02-2008
88888	ถนอมรัก	20-02-2008	26-05-2013
88888	บุญรักษา	26-05-2013	31-12-9999

3 rows selected

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

15. เพิ่มข้อมูลหมู่โลหิต

```

INSERT INTO blood VALUES ('A', 'เอ');
INSERT INTO blood VALUES ('B', 'บี');
INSERT INTO blood VALUES ('AB', 'เอบี');
INSERT INTO blood VALUES ('O', 'โอ');

```

16. เพิ่มข้อมูลชื่อของบุคคลอื่นๆ เพื่อใช้ในการทดลองในส่วนงานอื่น

-1) เพิ่มนายเนต

```

INSERT INTO personal_titlecode VALUES(
  '33333',1002,
  WMSYS.WM_PERIOD(TO_DATE('01-06-1977', 'DD-MM-YYYY'),
    DBMS_WM.MAX_TIME ));

INSERT INTO personal_fname VALUES(
  '33333', 'เนต',
  WMSYS.WM_PERIOD(TO_DATE('01-06-1977', 'DD-MM-YYYY'),
    DBMS_WM.MAX_TIME ));

INSERT INTO personal_lname VALUES(
  '33333', 'อุดมธรรม',
  WMSYS.WM_PERIOD(TO_DATE('01-06-1977', 'DD-MM-YYYY'),
    DBMS_WM.MAX_TIME ));

```

-2) เพิ่มนายสมชาติ ทดเจริญ

```

INSERT INTO personal_titlecode VALUES(
  '35003',1002,
  WMSYS.WM_PERIOD(TO_DATE('01-06-1982', 'DD-MM-YYYY'),
    DBMS_WM.MAX_TIME ));

INSERT INTO personal_fname VALUES(
  '35003', 'สมชาติ',
  WMSYS.WM_PERIOD(TO_DATE('01-06-1982', 'DD-MM-YYYY'),
    DBMS_WM.MAX_TIME ));

INSERT INTO personal_lname VALUES(
  '35003', 'ทดเจริญ',
  WMSYS.WM_PERIOD(TO_DATE('01-06-1982', 'DD-MM-YYYY'),
    DBMS_WM.MAX_TIME ));

```

-3) เพิ่มนายสิทธิ ภัทถนอม

```

INSERT INTO personal_titlecode VALUES(
  '66666',1002,
  WMSYS.WM_PERIOD(TO_DATE('04-06-2001', 'DD-MM-YYYY'),
    DBMS_WM.MAX_TIME ));

INSERT INTO personal_fname VALUES(
  '66666', 'สิทธิ',
  WMSYS.WM_PERIOD(TO_DATE('04-06-2001', 'DD-MM-YYYY'),
    DBMS_WM.MAX_TIME ));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
INSERT INTO personal_lname VALUES(
  '66666', 'รักถนนม',
  WMSYS.WM_PERIOD(TO_DATE('04-06-2001', 'DD-MM-YYYY'),
    DBMS_WM.MAX_TIME ));
```

```
commit;
```

17. สรุปผลการทดลองบันทึกแก้ไขข้อมูลด้านประวัติชื่อ-นามสกุล

จากการทดลอง สามารถเพิ่มข้อมูลชื่อ-นามสกุล (Insert) ตามช่วงเวลาที่ต้องการ โดยระบุช่วงเวลาเข้าไปในส่วนของการเพิ่มข้อมูลได้ตั้งข้อ 2 และในการปรับปรุงข้อมูล (Update) สามารถกำหนดช่วงเวลาที่ต้องการปรับปรุง โดยใช้คำสั่ง SetValidTime() จากนั้นสามารถใช้คำสั่งปรับปรุงได้ตามปกติ

การใช้คำสั่ง SetValidTime() ซึ่งเป็นคำสั่งที่รองรับข้อมูลด้านเชิงเวลา ทำให้สะดวกในการใช้คำสั่งนักพัฒนาโปรแกรมไม่ต้องกังวลในการตรวจสอบช่วงเวลาที่ยับซ้อนหรือคาบเกี่ยวกัน เมื่อใช้คำสั่ง SetValidTime() แล้วสามารถใช้คำสั่ง Update Delete ได้ตามปกติ

18. คำสั่งสำหรับลบตารางแบบเชิงเวลา และทุกตาราง เพื่อเริ่มต้นการทดลองตั้งแต่ข้อ 1 ถึง 14

```
EXECUTE DBMS_WM.DisableVersioning('personal_titlecode');
EXECUTE DBMS_WM.DisableVersioning('personal_fname');
EXECUTE DBMS_WM.DisableVersioning('personal_lname');
```

```
drop table personal;
drop table blood;
drop table titlecode;
drop table personal_titlecode;
drop table personal_fname;
drop table personal_lname;
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

การทดลองบันทึกแก้ไขข้อมูลด้านประวัติที่อยู่

1. ทำไมต้องใช้ระบบฐานข้อมูลแบบเชิงเวลา?

ระบบฐานข้อมูลแบบปกติ (non-Temporal) จะไม่มีการบันทึกช่วงเวลาของการเปลี่ยนแปลงข้อมูล ว่าข้อมูลนั้นๆ เริ่มต้นเมื่อใด (Start Time) สิ้นสุดเวลาใด (End Time) ทำให้เมื่อเราต้องการสอบถามข้อมูลที่สัมพันธ์กับเวลา ในอดีต ปัจจุบัน และอนาคต จึงไม่สามารถตอบได้ หรือทำได้ค่อนข้างยาก จะต้องเขียนคำสั่งตรวจสอบช่วงระยะเวลาต่างๆ หลายคำสั่ง หาก DBMS ที่ใช้ไม่รองรับคำสั่งเชิงเวลา (Temporal Query) เช่นกัน ดังตัวอย่างดังต่อไปนี้

1) ตัวอย่างโครงสร้างตารางบันทึกที่อยู่ ปัญหาที่ 1

รหัสประชาชน	เลขที่	หมู่ที่	ซอย	ถนน	รหัสตำบล	รหัสอำเภอ	รหัสจังหวัด	รหัสไปรษณีย์
	Num	Moo	Soi	Road	SubDistric_ID	Distric_ID	Province_id	Zipcode

จากโครงสร้างตารางที่ 1) นี้ จะเห็นว่า นำคอลัมน์ เลขที่บ้าน หมู่ ซอย ถนน ตำบล อำเภอ จังหวัด รหัสไปรษณีย์ อยู่ร่วมตารางเดียวกันกับเลขรหัสประจำตัวประชาชน (Citizen_Id) เมื่อมีการเปลี่ยนแปลงที่อยู่ ก็ให้นำข้อมูลที่อยู่ใหม่ ไปบันทึกแทนที่ ในข้อมูลเดิม ทำให้ข้อมูลที่อยู่เดิมหายไปจากระบบ ดังนั้น เวลาสอบถามประวัติการเปลี่ยนแปลงที่อยู่ ก็จะเห็นเฉพาะที่อยู่ในปัจจุบันเท่านั้น ไม่สามารถสอบถามข้อมูลในอดีตได้

2) ตัวอย่างโครงสร้างตารางบันทึกที่อยู่ ปัญหาที่ 2

รหัสบ้าน	เลขที่	หมู่ที่	ซอย	ถนน	รหัสตำบล	รหัสอำเภอ	รหัสจังหวัด	รหัสไปรษณีย์
Add_ID	Num	Moo	Soi	Road	SubDistric_ID	Distric_ID	Province_id	Zipcode
รหัสประชาชน	รหัสบ้าน							
Citizen_id	Add_ID							

จากโครงสร้างที่ 2) นี้ ในหนึ่งคนสามารถบันทึกข้อมูลที่อยู่ได้หลายที่อยู่ได้ (กำหนดให้ห้ามซ้ำที่อยู่เดิม) และสามารถสอบถามข้อมูลที่อยู่ย้อนหลังของแต่ละบุคคลได้

แต่ถ้าถามข้อสนเทศที่เกี่ยวข้องกับเวลา เช่น ในช่วงปี 2500-2535 มีใครบ้างที่เคยอาศัยอยู่ใน จ.อุดรธานี ตัวอย่างนี้ก็จะตอบไม่ได้ และในกรณีชื่อของคอลัมน์เปลี่ยน เช่น ชื่อตำบล ชื่ออำเภอ ชื่อจังหวัด เปลี่ยนไป ก็จะไม่สามารถตอบข้อมูลย้อนหลัง ระบบแบบเดิมจะทำการแทนรหัสตำบล อำเภอ จังหวัด เหมือนตัวอย่างที่ 1) ก็จะเข้าปัญหาเดิม คือ ไม่สามารถตอบคำถามในอดีตได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

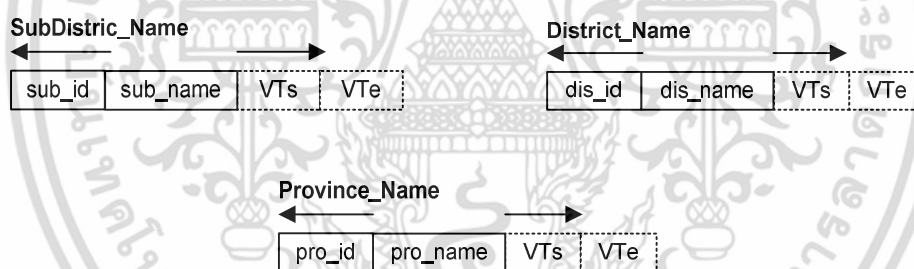
3) ตัวอย่าง การจัดตั้งจังหวัดใหม่

จากที่ อ.หนองบัวลำภู เคยเป็นอำเภอหนึ่งใน จ.อุดรธานี ได้แยกตัวเป็น จ.หนองบัวลำภู ตั้งแต่ปี 2536 และ อ.หนองบัวลำภู เปลี่ยนชื่อเป็น “เมืองหนองบัวลำภู” ระบบฐานข้อมูลแบบเดิม (non-Temporal) จะทำการเพิ่มรหัสจังหวัดหนองบัวลำภู และรหัสอำเภอเมืองหนองบัวลำภูเป็นรหัสใหม่ แล้วไปแทนที่หรือไปเพิ่มในข้อมูลเดิม ทำให้เกิดข้อมูลรหัสอำเภอ 2 รหัสที่ต่างกัน ชื่อต่างกัน แต่จริงๆ คืออำเภอเดียวกัน

ถ้าเราถามว่า จำนวนประชากรทั้งหมดที่เคยอาศัยอยู่ใน จ.อุดรธานี ในช่วงก่อนปี 2536 ถ้าเป็นข้อมูลแบบเดิม (non-Temporal) จะไม่สามารถตอบคำถามนี้ได้ เพราะประชากรที่เคยอาศัยอยู่ จ.อุดรธานี และย้ายสังกัดมาอยู่ จ.หนองบัวลำภู ในฐานข้อมูลได้ถูกแทนที่ด้วยรหัส จ.หนองบัวลำภู แล้ว

วิธีการแก้ปัญหา

จากตัวอย่างปัญหาที่กล่าวมานี้ จะแก้ไขได้โดย ต้องใช้ระบบฐานข้อมูลแบบเชิงเวลา (Temporal Database) โดยจะต้องทำการแยกข้อมูลทุก Fact ที่มีการเปลี่ยนแปลงแบบเชิงเวลา ออกเป็นตารางแบบ Valid Time ซึ่งประกอบด้วย รหัสของข้อมูล ชื่อข้อมูลที่สามารถเปลี่ยนแปลงได้ วันที่เริ่มต้นข้อมูล (VTs) และวันที่สิ้นสุดข้อมูล (VTe) ตัวอย่างเช่น



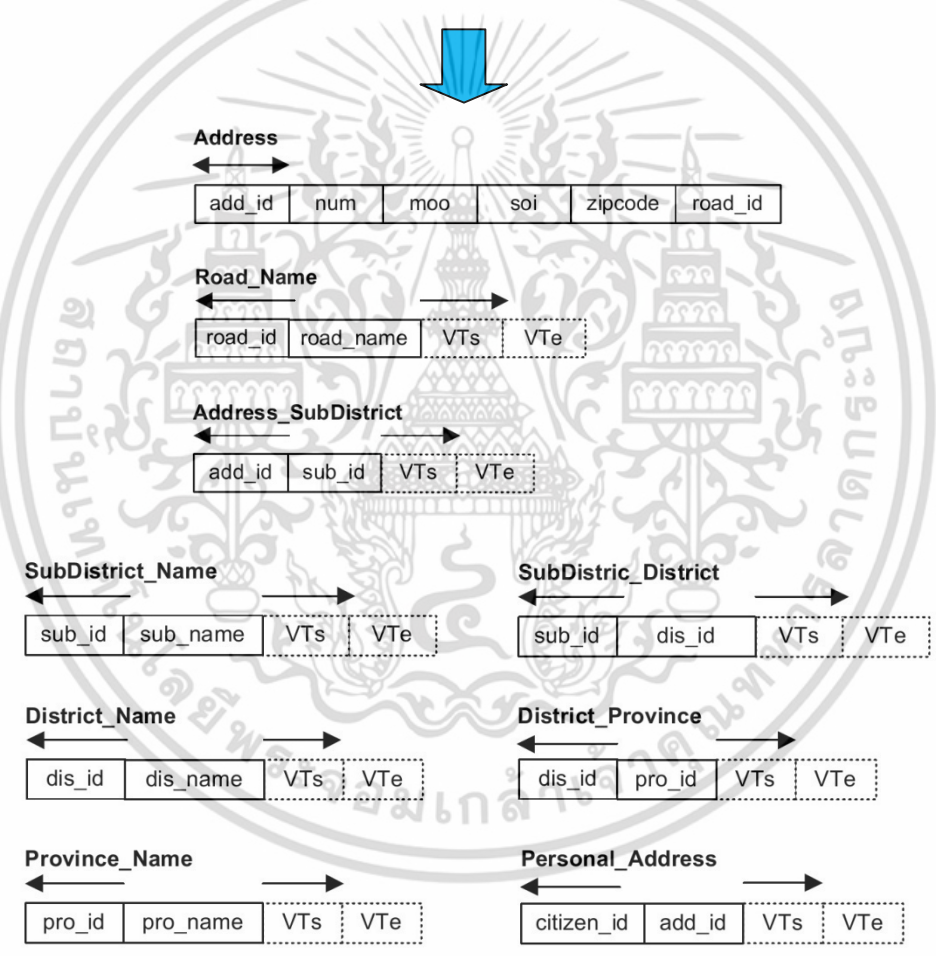
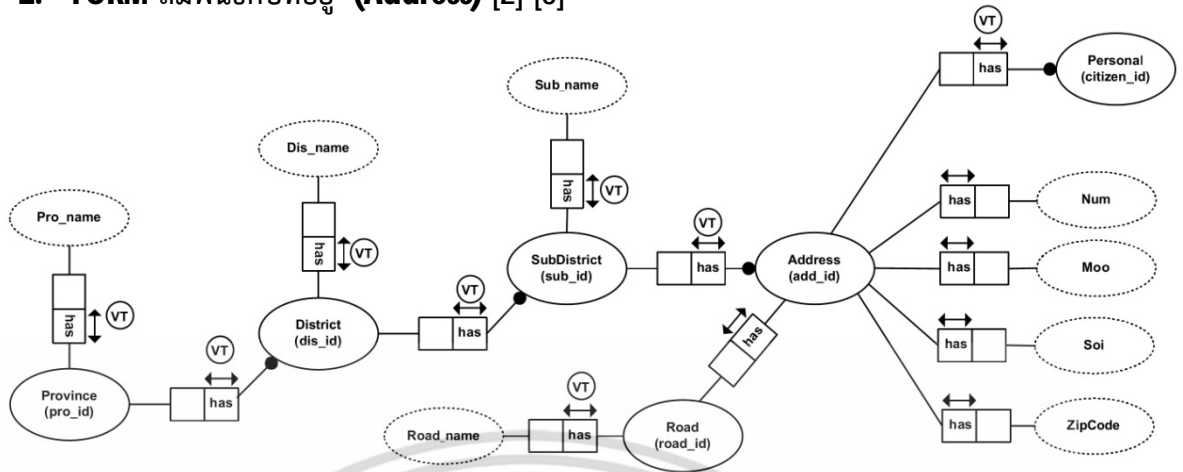
นอกจากที่จะเป็นระบบฐานข้อมูลแบบเชิงเวลาแล้ว จำเป็นที่จะต้องใช้โปรแกรมบริหารจัดการฐานข้อมูล (DBMS) ที่รองรับฐานข้อมูลแบบเชิงเวลา เช่น Oracle11g ซึ่งโปรแกรมบริหารจัดการนี้ จะมีส่วนดำเนินการและคำสั่งให้เราสามารถเพิ่ม ลบ ปรับปรุงข้อมูล ให้อัตโนมัติ เช่น จะเพิ่มคอลัมน์ VTs VTe ให้อัตโนมัติ จะทำการตรวจสอบช่วงเวลาที่เกี่ยวข้องกันของข้อมูลด้วยคำสั่ง ValidTime ให้อัตโนมัติ ทำให้เราไม่ต้องเขียนโปรแกรมตรวจสอบเวลาของข้อมูลที่เกี่ยวข้องกัน

หากเราใช้โปรแกรมบริหารจัดการฐานข้อมูลที่ไม่รองรับแบบเชิงเวลา เราสามารถเพิ่มคอลัมน์ VTs VTe ได้ด้วยตนเอง แต่เวลาเราจะเพิ่มข้อมูล ลบข้อมูล ปรับปรุงข้อมูล เราจะต้องเขียนคำสั่ง SQL ตรวจสอบคาบเวลาที่สัมพันธ์กัน ซึ่งในบางครั้งเราจะต้องใช้คำสั่ง SQL มาตรฐานถึง 5 คำสั่ง เพื่อที่จะทำให้การ เพิ่มข้อมูล ลบข้อมูล ปรับปรุงข้อมูลถูกต้อง ดังที่ Richard T. Snodgrass

ในกล่าวถึงปัญหานี้ไว้ในเอกสาร [Managing Temporal Data A Five-Part Series \[1\]](#)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ORM สัมพันธ์กับที่อยู่ (Address) [2] [3]



คำถามด้านเชิงเวลา (Temporal Query)

- 1) แสดงที่อยู่ ชื่อตำบล ชื่ออำเภอ ชื่อจังหวัด ของพนักงาน ณ ช่วงเวลาปัจจุบันได้
- 2) แสดงที่อยู่ ชื่อตำบล ชื่ออำเภอ ชื่อจังหวัด ของพนักงาน ณ ช่วงเวลาในอดีตได้
- 3) จำนวนประชากร ที่เคยอาศัยอยู่ จ.อุดรธานี ช่วงปี **2500-2535** และ **2536-ปัจจุบัน**
- 4) จำนวนประชากร ที่อาศัยอยู่ จ.หนองบัวลำภู ช่วงปี **2536-ปัจจุบัน**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ตารางรหัสจังหวัด ที่ใช้ในปัจจุบัน (โปรแกรมทะเบียนนักเรียน ของ สพฐ)

จากตารางต่อไปนี้ แสดงรหัสจังหวัดของประเทศไทย โดยอิงตามรหัสไปรษณีย์ การแบ่งตามภูมิภาคจะใช้เลข 1-9 แล้วตามด้วยลำดับของจังหวัดในแต่ละภูมิภาค เมื่อมีจังหวัดใหม่เกิดขึ้น จะเพิ่มเลขรหัสต่อท้าย หรือเติมในรหัสจังหวัดที่ยังว่าง เช่น จังหวัดสระแก้ว (รหัส 27)

กทม. ปริมาณและ

ภาคกลางตอนล่าง

รหัส	ชื่อจังหวัด
10	กรุงเทพมหานคร
11	สมุทรปราการ
12	นนทบุรี
13	ปทุมธานี
14	พระนครศรีอยุธยา
15	อ่างทอง
16	ลพบุรี
17	สิงห์บุรี
18	ชัยนาท
19	สระบุรี

ภาคตะวันออก

รหัส	ชื่อจังหวัด
20	ชลบุรี
21	ระยอง
22	จันทบุรี
23	ตราด
24	ฉะเชิงเทรา
25	ปราจีนบุรี
26	นครนายก
27	สระแก้ว

ภาคอีสาน

รหัส	ชื่อจังหวัด
30	นครราชสีมา
31	บุรีรัมย์
32	สุรินทร์
33	ศรีสะเกษ
34	อุบลราชธานี
35	ยโสธร
36	ชัยภูมิ
37	อำนาจเจริญ
39	หนองบัวลำภู
40	ขอนแก่น
41	อุดรธานี
42	เลย
43	หนองคาย
44	มหาสารคาม
45	ร้อยเอ็ด
46	กาฬสินธุ์
47	สกลนคร
48	นครพนม
49	มุกดาหาร

ภาคเหนือตอนบน

รหัส	ชื่อจังหวัด
50	เชียงใหม่
51	ลำพูน
52	ลำปาง
53	อุตรดิตถ์
54	แพร่
55	น่าน
56	พะเยา
57	เชียงราย
58	แม่ฮ่องสอน

ภาคเหนือตอนล่าง

รหัส	ชื่อจังหวัด
60	นครสวรรค์
61	อุทัยธานี
62	กำแพงเพชร
63	ตาก
64	สุโขทัย
65	พิษณุโลก
66	พิจิตร
67	เพชรบูรณ์

ภาคตะวันตก

รหัส	ชื่อจังหวัด
70	ราชบุรี
71	กาญจนบุรี
72	สุพรรณบุรี
73	นครปฐม
74	สมุทรสาคร
75	สมุทรสงคราม
76	เพชรบุรี
77	ประจวบคีรีขันธ์

ภาคใต้

รหัส	ชื่อจังหวัด
80	นครศรีธรรมราช
81	กระบี่
82	พังงา
83	ภูเก็ต
84	สุราษฎร์ธานี
85	ระนอง
86	ชุมพร
90	สงขลา
91	สตูล
92	ตรัง
93	พัทลุง
94	ปัตตานี
95	ยะลา
96	นราธิวาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. สมมุติตัวอย่างประชากร เป็นกรณีศึกษาด้านที่อยู่แบบเชิงเวลา

เพื่อความเข้าใจเกี่ยวกับข้อมูลแบบเชิงเวลา ที่สัมพันธ์กับที่อยู่ จะสมมุติข้อมูลประชากรที่อาศัยอยู่ใน จ.อุตรธานี จำนวน 10 คน โดยสังกัด อ.เมืองอุตรธานี 6 คน และสังกัด อ.หนองบัวลำภู 4 คน เมื่อจัดตั้งจังหวัดหนองบัวลำภู ในวันที่ 1 ธ.ค. 2536 แล้ว จะทำการย้าย (ปรับปรุงข้อมูล) ประชากรที่อาศัย อ.หนองบัวลำภู ให้สังกัด จ.หนองบัวลำภู และเปลี่ยนเป็น อ.เมืองหนองบัวลำภู

ตาราง 4.1 ข้อมูลสมมุติตัวอย่างของประชากรที่อาศัยอยู่ใน จ.อุตรธานี (ข้อมูลก่อนปี 2536)

ที่	เลขประจำตัวประชาชน	คำนาม	ชื่อ	สกุล	รหัสบ้าน	เลขที่	หมู่	ซอย	ตำบล	อำเภอ	จังหวัด	ไปรษณีย์	ว.ด.ป. (ค.ศ) ย้าย/บันทึก
1	11111	นาย	บุญทัน	รักถนอม	1001	1	1		หมากแข้ง	เมืองอุตรธานี	อุตรธานี	41000	15/05/2512 (1969)
2	22222	นาง	อนงค์	รักถนอม	1001	1	1		หมากแข้ง	เมืองอุตรธานี	อุตรธานี	41000	15/05/2512 (1969)
3	33333	นาย	นเรศ	อุตรธรรม	1002	2	1		หมากแข้ง	เมืองอุตรธานี	อุตรธานี	41000	01/01/2501 (1958)
4	44444	นาง	สมใจ	อุตรธรรม	1002	2	1		หมากแข้ง	เมืองอุตรธานี	อุตรธานี	41000	01/01/2501 (1958)
5	55555	นาง	ปราณี	จิตดำรงค์	1003	3	1		หมากแข้ง	เมืองอุตรธานี	อุตรธานี	41000	01/01/2501 (1958)
6	66666	นาย	สิทธิ	รักถนอม	3001	1	3		หนองบัว	หนองบัวลำภู	อุตรธานี	41390	28/02/2519 (1976)
7	77777	นาย	เจษฎา	สินรักษา	4001	1	4		หมากแข้ง	เมืองอุตรธานี	อุตรธานี	41000	08/08/2508 (1965)
8	88888	น.ส.	กนกพร	บุญรักษา	5001	1	5		หนองบัว	หนองบัวลำภู	อุตรธานี	41390	14/07/2529 (1986)
9	99999	นาง	ปราณี	ทศเจริญ	5001	1	5		หนองบัว	หนองบัวลำภู	อุตรธานี	41390	14/07/2529 (1986)
10	10000	นาย	ปิใหม่	มงคลทัพ	5001	1	5		หนองบัว	หนองบัวลำภู	อุตรธานี	41390	14/70/2529 (1986)

หมายเหตุ

- 1) จังหวัดอุตรธานี ถือว่ามีการบันทึกในทะเบียนราษฎร วันที่ 21 มิถุนายน พ.ศ. 2475 (ค.ศ. 1932) [4]
- 2) อำเภอหนองบัวลำภู แยกตัวจากจังหวัดอุตรธานี จัดตั้งเป็นจังหวัดหนองบัวลำภู วันที่ 1 ธันวาคม พ.ศ. 2536 (ค.ศ. 1993) [5]

ตัวอย่างคำถามและคำตอบด้านเชิงเวลา (Temporal Query)

- 1) จำนวนประชากรที่อาศัยอยู่ใน จ.อุตรธานี และ จ.หนองบัวลำภู ในช่วงเวลาปัจจุบัน
ตอบ จ.อุตรธานี มีประชากร **6** คน
จ.หนองบัวลำภู มีประชากร **4** คน
- 2) จำนวนประชากรที่เคยอาศัยใน จ.อุตรธานี ในช่วงปี 2550-2535
ตอบ มีประชากรที่เคยอาศัย จ.อุตรธานี จำนวน **10** คน
- 3) จำนวนประชากร ที่เคยอยู่อำเภอหนองบัวลำภู
- 4) จำนวนประชากร ที่อยู่อำเภอเมืองหนองบัวลำภู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ทดลองสมมุติข้อมูลในตาราง TORM ที่ได้ออกแบบในข้อ 2

จากการแปลงรูป TORM ในข้อ 2 และข้อมูลตัวอย่างในข้อ 4 ได้ตารางแบบเชิงเวลา 8 ตาราง ตารางแบบปกติ 1 ตาราง เพื่อความเข้าใจง่าย ผู้ทดลอง ได้สมมุติข้อมูลลงในแต่ละตาราง ดังนี้

ตาราง 5.1 บันทึกชื่อจังหวัด (Province_Name)

รหัสจังหวัด	ชื่อจังหวัด	เริ่มต้น	สิ้นสุด
41	อุดรธานี	21/06/1932	
39	หนองบัวลำภู	01/12/1993	

ตาราง 5.2 บันทึกชื่ออำเภอ (District_Name)

รหัสอำเภอ	ชื่ออำเภอ	เริ่มต้น	สิ้นสุด
4101	เมืองอุดรธานี	21/06/1932	
4102	หนองบัวลำภู	21/06/1932	01/12/1993
4102	เมืองหนองบัวลำภู	01/12/1993	

ตาราง 5.3 บันทึกอำเภอสังกัดจังหวัด (District_Province)

รหัสอำเภอ	รหัสจังหวัด	เริ่มต้น	สิ้นสุด
4101	41	21/06/1932	
4102	41	21/06/1932	01/12/1993
4102	39	01/12/1993	

ตาราง 5.4 บันทึกชื่อตำบล (SubDistrict_Name)

รหัสตำบล	ชื่อตำบล	เริ่มต้น	สิ้นสุด
410101	หมากแข้ง	21/06/1932	
410102	หนองบัว	21/06/1932	

ตาราง 5.5 บันทึกตำบลสังกัดอำเภอ (SubDistrict_District)

รหัสตำบล	รหัสอำเภอ	เริ่มต้น	สิ้นสุด
410101	4101	21/06/1932	
410102	4102	21/06/1932	

ตาราง 5.6 บันทึกชื่อถนน (Road_Name)

รหัสถนน	ชื่อถนน	เริ่มต้น	สิ้นสุด
11	อุดรสุขุมิ	21/06/1932	
12	หนองบัวลำภู	21/06/1932	01/12/1993
12	หนองบัวลำภู-อุดรธานี	01/12/1993	

ตารางที่ 5.7 บันทึกรหัสประจำบ้าน (Address)

รหัสบ้าน	เลขที่	หมู่	ซอย	รหัสไปรษณีย์	รหัสถนน
1001	1	1		41000	11
1002	2	1		41000	11
1003	3	1		41000	11
3001	1	3		41390	12
4001	1	4		41000	11
5001	1	5		41390	12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 5.8 บันทึกรหัสบ้านสังกัดตำบล(Address_Subdistrict)

รหัสบ้าน	รหัสตำบล	เริ่มต้น	สิ้นสุด
1001	410101	21/06/1932	
1002	410101	21/06/1932	
1003	410101	21/06/1932	
3001	410102	21/06/1932	
4001	410101	21/06/1932	
5001	410102	21/06/1932	

ตารางที่ 5.9 พนักงานมีที่อยู่ (Personal_Address)

เลขประจำตัวประชาชน	รหัสประจำบ้าน	เริ่มต้น	สิ้นสุด
11111	1001	15/05/1969	
22222	1001	15/05/1969	
33333	1002	01/01/1958	
44444	1002	01/01/1958	
55555	1003	01/01/1958	
66666	3001	28/02/1976	
77777	4001	08/08/1965	
88888	5001	14/07/1986	
99999	5001	14/07/1986	
10000	5001	14/70/1986	

ตาราง 5.1 - 5.6 และ 5.8-5.9 เป็นตารางแบบ Valid Time State Table จะเห็นว่า Fact ทุกค่าจะถูกแยกออกเป็นแต่ละตาราง โดยจะมีรหัสข้อมูล ชื่อข้อมูล วันที่เริ่มต้นข้อมูล และวันที่สิ้นสุดข้อมูล ส่วนตาราง 5.7 เป็นตารางแบบปกติ (non-Temporal)

การเพิ่มข้อมูล จะต้องระบุวันที่เริ่มต้น ถึงวันที่สิ้นสุดข้อมูล หากข้อมูลยังเป็นปัจจุบันจะบันทึกได้ 2 ค่าคือ แบบวันที่ 31-12-9999 (ออรากิล ใช้คำสั่ง MAX_TIME) หรือบันทึกเป็นค่า Null (ออรากิลใช้คำสั่ง UNTIL_CHANGE)

การปรับปรุงข้อมูล วันที่สิ้นสุดของข้อมูลเก่า จะเป็นวันเริ่มต้นของข้อมูลใหม่ ดังตารางที่ 5.2 จะเห็นว่า อ.หนองบัวลำภู และ อ.เมืองหนองบัวลำภู จะใช้รหัสอำเภอเหมือนเดิม คือ 4102 แต่ข้อมูล อ.หนองบัวลำภู จะสิ้นสุดวันที่ 01/12/1993 และ อ.เมืองหนองบัวลำภู จะเริ่มต้นวันที่ 01/12/1993 จนถึงปัจจุบัน (Null)

6. ทดลองคำสั่ง **Oracle 11g** ตามการออกแบบ **TORM** ด้านที่อยู่

6.1 สร้างตารางจำนวน 9 ตาราง ดังนี้ 1. Address 2. Road_name 3. Address_subdistrict
4. Subdistrict_name 5. Subdistrict_district 6. District_name 7. District_province 8.
Province_name และ 9. Personal_Address [6]

```

CREATE TABLE address(
  add_id          NUMBER PRIMARY KEY,
  num            VARCHAR(13),
  moo           VARCHAR(30),
  soi           VARCHAR(50),
  zipcode       VARCHAR(5),
  road_id       NUMBER);

CREATE TABLE road_name(
  road_id        NUMBER PRIMARY KEY,
  road_name     VARCHAR(50));

CREATE TABLE address_subdistrict(
  add_id         NUMBER PRIMARY KEY,
  sub_id        NUMBER);

CREATE TABLE subdistrict_name(
  sub_id        NUMBER PRIMARY KEY,
  sub_name     VARCHAR(30));

CREATE TABLE subdistrict_district(
  sub_id        NUMBER PRIMARY KEY,
  dis_id       NUMBER);

CREATE TABLE district_name(
  dis_id       NUMBER PRIMARY KEY,
  dis_name    VARCHAR(30));

CREATE TABLE district_province(
  dis_id       NUMBER PRIMARY KEY,
  pro_id      NUMBER);

CREATE TABLE province_name(
  pro_id      NUMBER PRIMARY KEY,
  pro_name   VARCHAR(30));

CREATE TABLE personal_address(
  citizen_id  VARCHAR(13) PRIMARY KEY,
  add_id     NUMBER);

commit;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 ใช้คำสั่ง EnableVersioning ทำให้ตาราง เป็นตารางแบบเชิงเวลา ยกเว้นตาราง Address

```
EXECUTE DBMS_WM.EnableVersioning('road_name',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

EXECUTE DBMS_WM.EnableVersioning('address_subdistrict',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

EXECUTE DBMS_WM.EnableVersioning('subdistrict_name',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

EXECUTE DBMS_WM.EnableVersioning('subdistrict_district',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

EXECUTE DBMS_WM.EnableVersioning('district_name',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

EXECUTE DBMS_WM.EnableVersioning('district_province',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

EXECUTE DBMS_WM.EnableVersioning('province_name',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

EXECUTE DBMS_WM.EnableVersioning('personal_address',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

commit;
```

6.3 ทำการเพิ่มข้อมูล จังหวัด อำเภอ ตำบล โดยกำหนดวันเริ่มต้นข้อมูล 21 มิ.ย. 2475 (1932)

- 1) เพิ่มชื่อจังหวัด อุดรธานี (41) ในตาราง province_name
- 2) เพิ่มชื่ออำเภอ เมืองอุดรธานี(4101)หนองบัวลำภู(4102) ในตาราง district_name
- 3) เพิ่มรหัสอำเภอ 4101 4102 สังกัดรหัสจังหวัด 41 ในตาราง district_province
- 4) เพิ่มชื่อตำบล หมากแข้ง(410101) หนองบัว(410102) ในตาราง subdistrict_name
- 5) เพิ่มรหัสตำบล 410101 สังกัดรหัสอำเภอ 4101 และเพิ่มรหัสตำบล 410102 สังกัดรหัสตำบล 4102 ในตาราง subdistrict_district
- 6) เพิ่มชื่อและรหัสถนน ในตาราง road_name

--1) เพิ่มชื่อจังหวัด

```
INSERT INTO province_name VALUES( 41, 'อุดรธานี', WMSYS.WM_PERIOD
(TO_DATE(' 21-06-1932', 'DD-MM-YYYY'), DBMS_WM.MAX_TIME));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

--2) เพิ่มชื่ออำเภอ

```
INSERT INTO district_name VALUES(4101, 'เมืองอุดรธานี',
WMSYS.WM_PERIOD(TO_DATE('21-06-1932', 'DD-MM-YYYY')),
DBMS_WM.MAX_TIME));
```

```
INSERT INTO district_name VALUES(4102, 'หนองบัวลำภู',
WMSYS.WM_PERIOD(TO_DATE('21-06-1932', 'DD-MM-YYYY')),
DBMS_WM.MAX_TIME));
```

--3) เพิ่ม รหัสอำเภอ สังกัด รหัสจังหวัด

```
INSERT INTO district_province VALUES(4101,41,
WMSYS.WM_PERIOD(TO_DATE('21-06-1932', 'DD-MM-YYYY')),
DBMS_WM.MAX_TIME));
```

```
INSERT INTO district_province VALUES(4102,41,
WMSYS.WM_PERIOD(TO_DATE('21-06-1932', 'DD-MM-YYYY')),
DBMS_WM.MAX_TIME));
```

--4) เพิ่มชื่อตำบล

```
INSERT INTO subdistrict_name VALUES(410101, 'หมากแข้ง',
WMSYS.WM_PERIOD(TO_DATE('21-06-1932', 'DD-MM-YYYY')),
DBMS_WM.MAX_TIME));
```

```
INSERT INTO subdistrict_name VALUES(410102, 'หนองบัว',
WMSYS.WM_PERIOD(TO_DATE('21-06-1932', 'DD-MM-YYYY')),
DBMS_WM.MAX_TIME));
```

--5) เพิ่ม รหัสตำบล สังกัด รหัสอำเภอ

```
INSERT INTO subdistrict_district VALUES(410101,4101,
WMSYS.WM_PERIOD(TO_DATE('21-06-1932', 'DD-MM-YYYY')),
DBMS_WM.MAX_TIME));
```

```
INSERT INTO subdistrict_district VALUES(410102,4102,
WMSYS.WM_PERIOD(TO_DATE('21-06-1932', 'DD-MM-YYYY')),
DBMS_WM.MAX_TIME));
```

--6) เพิ่ม ชื่อถนน

```
INSERT INTO road_name VALUES(11, 'อุดรคูขลุ่ย',
WMSYS.WM_PERIOD(TO_DATE('21-06-1932', 'DD-MM-YYYY')),
DBMS_WM.MAX_TIME));
```

```
INSERT INTO road_name VALUES(12, 'หนองบัวลำภู',
WMSYS.WM_PERIOD(TO_DATE('21-06-1932', 'DD-MM-YYYY')),
DBMS_WM.MAX_TIME));
```

```
commit;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 ทำการเพิ่มข้อมูล รหัสประจำบ้าน โดยกำหนดวันเริ่มต้นข้อมูล 21 มิ.ย. 2475 (1932)

- 1) เพิ่มข้อมูลของรหัสบ้าน ในตาราง **address**
- 2) เพิ่มรหัสบ้าน สังกัดตำบล ในตาราง **address_subdistrict**

```
--1) เพิ่ม รหัสประจำบ้าน
INSERT INTO address VALUES(1001, '1', '1', '', '41000', 11);
INSERT INTO address VALUES(1002, '2', '1', '', '41000', 11);
INSERT INTO address VALUES(1003, '3', '1', '', '41000', 11);
INSERT INTO address VALUES(3001, '1', '3', '', '41390', 12);
INSERT INTO address VALUES(4001, '1', '4', '', '41000', 11);
INSERT INTO address VALUES(5001, '1', '5', '', '41390', 12);

--2) เพิ่ม รหัสประจำบ้าน สังกัด รหัสตำบล
INSERT INTO Address_subdistrict VALUES(1001,410101,
WMSYS.WM_PERIOD(TO_DATE('21-06-1932','DD-MM-YYYY'),
DBMS_WM.MAX_TIME));
INSERT INTO Address_subdistrict VALUES(1002,410101,
WMSYS.WM_PERIOD(TO_DATE('21-06-1932','DD-MM-YYYY'),
DBMS_WM.MAX_TIME));
INSERT INTO Address_subdistrict VALUES(1003,410101,
WMSYS.WM_PERIOD(TO_DATE('21-06-1932','DD-MM-YYYY'),
DBMS_WM.MAX_TIME));
INSERT INTO Address_subdistrict VALUES(3001,410102,
WMSYS.WM_PERIOD(TO_DATE('21-06-1932','DD-MM-YYYY'),
DBMS_WM.MAX_TIME));
INSERT INTO Address_subdistrict VALUES(4001,410101,
WMSYS.WM_PERIOD(TO_DATE('21-06-1932','DD-MM-YYYY'),
DBMS_WM.MAX_TIME));
INSERT INTO Address_subdistrict VALUES(5001,410102,
WMSYS.WM_PERIOD(TO_DATE('21-06-1932','DD-MM-YYYY'),
DBMS_WM.MAX_TIME));

commit;
```

6.5 ทำการเพิ่ม เลขประจำตัวประชาชน เข้ากับรหัสบ้าน ดังในตารางที่ 4.1

```
INSERT INTO personal_address VALUES('11111', 1001,
WMSYS.WM_PERIOD(TO_DATE('15-05-1969','DD-MM-YYYY'),
DBMS_WM.MAX_TIME));
INSERT INTO personal_address VALUES('22222', 1001,
WMSYS.WM_PERIOD(TO_DATE('15-05-1969','DD-MM-YYYY'),
DBMS_WM.MAX_TIME));
INSERT INTO personal_address VALUES('33333', 1002,
WMSYS.WM_PERIOD(TO_DATE('01-01-1958','DD-MM-YYYY'),
DBMS_WM.MAX_TIME));
INSERT INTO personal_address VALUES('44444', 1002,
WMSYS.WM_PERIOD(TO_DATE('01-01-1958','DD-MM-YYYY'),
DBMS_WM.MAX_TIME));
INSERT INTO personal_address VALUES('55555', 1003,
WMSYS.WM_PERIOD(TO_DATE('01-01-1958','DD-MM-YYYY'),
DBMS_WM.MAX_TIME));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

INSERT INTO personal_address VALUES('66666', 3001,
WMSYS.WM_PERIOD(TO_DATE('28-02-1976','DD-MM-YYYY'),
DBMS_WM.MAX_TIME));
INSERT INTO personal_address VALUES('77777', 4001,
WMSYS.WM_PERIOD(TO_DATE('08-08-1965','DD-MM-YYYY'),
DBMS_WM.MAX_TIME));
INSERT INTO personal_address VALUES('88888', 5001,
WMSYS.WM_PERIOD(TO_DATE('14-07-1986','DD-MM-YYYY'),
DBMS_WM.MAX_TIME));
INSERT INTO personal_address VALUES('99999', 5001,
WMSYS.WM_PERIOD(TO_DATE('14-07-1986','DD-MM-YYYY'),
DBMS_WM.MAX_TIME));
INSERT INTO personal_address VALUES('10000', 5001,
WMSYS.WM_PERIOD(TO_DATE('14-07-1986','DD-MM-YYYY'),
DBMS_WM.MAX_TIME));
commit;

```

7. สอบถามข้อมูล ณ ช่วงเวลาปัจจุบัน จะเห็นว่าประชากรตัวอย่างทุกคน สังกัดจังหวัดอุดรธานี

```
EXECUTE DBMS_WM.SetValidTime( );
```

```

SELECT personal_address.citizen_id, address.num, address.moo,
address soi, road_name.road_name, subdistrict_name.sub_name,
district_name.dis_name, province_name.pro_name, address.zipcode
FROM ((((((personal_address JOIN address ON personal_address.add_id
= address.add_id)
JOIN road_name ON address.road_id = road_name.road_id)
JOIN address_subdistrict ON personal_address.add_id =
address_subdistrict.add_id)
JOIN subdistrict_name ON address_subdistrict.sub_id =
subdistrict_name.sub_id)
JOIN subdistrict_district ON address_subdistrict.sub_id =
subdistrict_district.sub_id)
JOIN district_name ON subdistrict_district.dis_id =
district_name.dis_id)
JOIN district_province ON subdistrict_district.dis_id =
district_province.dis_id)
JOIN province_name ON district_province.pro_id = province_name.pro_id
ORDER BY personal_address.citizen_id;

```

anonymous block completed

CITIZEN_ID	NUM	MOO	SOI	ROAD	SUB_NAME	DIS_NAME	PRO_NAME	ZIPCODE
10000	1	5		หนองบัวลำภู	หนองบัว	หนองบัวลำภู	อุดรธานี	41390
11111	1	1		อุดรคูมณี	หมากแข้ง	เมืองอุดรธานี	อุดรธานี	41000
22222	1	1		อุดรคูมณี	หมากแข้ง	เมืองอุดรธานี	อุดรธานี	41000
33333	2	1		อุดรคูมณี	หมากแข้ง	เมืองอุดรธานี	อุดรธานี	41000
44444	2	1		อุดรคูมณี	หมากแข้ง	เมืองอุดรธานี	อุดรธานี	41000
55555	3	1		อุดรคูมณี	หมากแข้ง	เมืองอุดรธานี	อุดรธานี	41000
66666	1	3		หนองบัวลำภู	หนองบัว	หนองบัวลำภู	อุดรธานี	41390
77777	1	4		อุดรคูมณี	หมากแข้ง	เมืองอุดรธานี	อุดรธานี	41000
88888	1	5		หนองบัวลำภู	หนองบัว	หนองบัวลำภู	อุดรธานี	41390
99999	1	5		หนองบัวลำภู	หนองบัว	หนองบัวลำภู	อุดรธานี	41390

10 rows selected

1,496ms elapsed
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. เพิ่มข้อมูลจังหวัด และปรับปรุงอำเภอ ตำบล โดยกำหนดวันเริ่มต้นข้อมูล 1 ธ.ค. 2536 (1993)
- 1) เพิ่มชื่อจังหวัด นongบัวลำภู(39) ในตาราง province_name
 - 2) ปรับปรุงชื่ออำเภอ รหัส 4102 เป็นอำเภอ เมืองหนองบัวลำภู ในตาราง district_name
 - 3) ปรับปรุงอำเภอเมืองหนองบัวลำภู (1402) สังกัดจังหวัดหนองบัวลำภู (39) ในตาราง district_province

```
--1)เพิ่มชื่อจังหวัด
INSERT INTO province_name VALUES( 39,'หนองบัวลำภู',
WMSYS.WM_PERIOD(TO_DATE('01-12-1993','DD-MM-YYYY'),
DBMS_WM.MAX_TIME));

--2)ปรับปรุงชื่ออำเภอ

EXECUTE DBMS_WM.SetValidTime(TO_DATE('01-12-1993','DD-MM-
YYYY'), DBMS_WM.MAX_TIME);

UPDATE district_name SET dis_name ='เมืองหนองบัวลำภู' WHERE
dis_id=4102;

--3)ปรับปรุงอำเภอเมืองหนองบัวลำภู(1402) สังกัดจังหวัดหนองบัวลำภู(39)
UPDATE district_province SET pro_id =39 WHERE dis_id=4102;
commit;
```

9. สอบถามข้อสนเทศเชิงเวลา (Temporal Query)

9.1 ตรวจสอบดูผลการปรับปรุงในตาราง district_name จะเห็นว่า รหัสอำเภอ 1402 จะมีชื่ออำเภอ ปรากฏ 2 แถว คือ หนองบัวลำภู และ เมืองหนองบัวลำภู

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT dis_id, dis_name,
TO_CHAR(e.WM_VALID.VALIDFROM,'DD-MM-YYYY') FROM_DATE,
TO_CHAR(e.WM_VALID.VALIDTILL,'DD-MM-YYYY') TO_DATE
FROM district_name e
```

anonymous block completed

DIS_ID	DIS_NAME	FROM_DATE	TO_DATE
4101	เมืองอุดรธานี	21-06-1932	31-12-9999
4102	เมืองหนองบัวลำภู	01-12-1993	31-12-9999
4102	หนองบัวลำภู	21-06-1932	01-12-1993

3 rows selected

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.2 ตรวจสอบผลการปรับปรุงในตาราง district_province จะเห็นว่า รหัสอำเภอ 1402 จะสังกัดรหัสจังหวัด 2 รหัส คือ 41 และ 39

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);
```

```
SELECT dis_id, pro_id,
       TO_CHAR(e.WM_VALID.VALIDFROM,'DD-MM-YYYY') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL,'DD-MM-YYYY') TO_DATE
FROM district_province e
```

DIS_ID	PRO_ID	FROM_DATE	TO_DATE
4101	41	21-06-1932	31-12-9999
4102	39	01-12-1993	31-12-9999
4102	41	21-06-1932	01-12-1993

3 rows selected

9.3 สอบถามที่อยู่ของประชากรแต่ละคน ณ ช่วงเวลาปัจจุบัน โดยเขียนคำสั่งอีกลักษณะหนึ่ง ซึ่งแตกต่างจากข้อ 7 แต่ก็ได้คำตอบถูกต้องเหมือนกัน

```
EXECUTE DBMS_WM.SetValidTime( );
```

```
SELECT personal_address.citizen_id, address.num, address.moo,
address soi, road_name.road_name, subdistrict_name.sub_name,
district_name.dis_name, province_name.pro_name, address.zipcode
FROM personal_address, address, road_name, address_subdistrict,
subdistrict_name, subdistrict_district, district_name,
district_province, province_name
WHERE personal_address.add_id = address.add_id
AND address.road_id = road_name.road_id
AND personal_address.add_id = address_subdistrict.add_id
AND address_subdistrict.sub_id = subdistrict_name.sub_id
AND address_subdistrict.sub_id = subdistrict_district.sub_id
AND subdistrict_district.dis_id = district_name.dis_id
AND subdistrict_district.dis_id = district_province.dis_id
AND district_province.pro_id = province_name.pro_id
ORDER BY personal_address.citizen_id;
```

anonymous block completed

CITIZEN_ID	NUM	MOO	SOI	ROAD	SUB_NAME	DIS_NAME	PRO_NAME	ZIPCODE
10000	1	5		หนองบัวลำภู	หนองบัว	เมืองหนองบัวลำภู	หนองบัวลำภู	41390
11111	1	1		อุครคุษฎี	หมากแข้ง	เมืองอุครธานี	อุครธานี	41000
22222	1	1		อุครคุษฎี	หมากแข้ง	เมืองอุครธานี	อุครธานี	41000
33333	2	1		อุครคุษฎี	หมากแข้ง	เมืองอุครธานี	อุครธานี	41000
44444	2	1		อุครคุษฎี	หมากแข้ง	เมืองอุครธานี	อุครธานี	41000
55555	3	1		อุครคุษฎี	หมากแข้ง	เมืองอุครธานี	อุครธานี	41000
66666	1	3		หนองบัวลำภู	หนองบัว	เมืองหนองบัวลำภู	หนองบัวลำภู	41390
77777	1	4		อุครคุษฎี	หมากแข้ง	เมืองอุครธานี	อุครธานี	41000
88888	1	5		หนองบัวลำภู	หนองบัว	เมืองหนองบัวลำภู	หนองบัวลำภู	41390
99999	1	5		หนองบัวลำภู	หนองบัว	เมืองหนองบัวลำภู	หนองบัวลำภู	41390

10 rows selected

16ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.4 สอบถามจำนวนประชากร ณ ช่วงเวลาปัจจุบัน จะเห็นว่าจำนวนประชากร จะแสดงได้ถูกต้อง โดยจำนวนประชากร จ.อุดรธานี = 6 คน และ จ.หนองบัวลำภู = 4 คน

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT province_name.pro_id, province_name.pro_name,
Count(personal_address.citizen_id) AS AMOUNT
FROM ((personal_address JOIN address_subdistrict ON
personal_address.add_id = address_subdistrict.add_id)
JOIN subdistrict_district ON address_subdistrict.sub_id =
subdistrict_district.sub_id)
JOIN district_province ON subdistrict_district.dis_id =
district_province.dis_id)
JOIN province_name ON district_province.pro_id = province_name.pro_id
GROUP BY province_name.pro_id, province_name.pro_name;
```

```
anonymous block completed
PRO_ID          PRO_NAME          AMOUNT
-----
41              อุดรธานี         6
39              หนองบัวลำภู     4

2 rows selected
8ms elapsed
```

เขียนสั้งแบบที่ 2 สามารถตอบคำตอบข้อ 9.4 ได้ผลลัพธ์เหมือนกัน

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT province_name.pro_id, province_name.pro_name,
Count(personal_address.citizen_id) AS AMOUNT
FROM personal_address, address_subdistrict, subdistrict_district,
district_province, province_name
WHERE personal_address.add_id = address_subdistrict.add_id
AND address_subdistrict.sub_id = subdistrict_district.sub_id
AND subdistrict_district.dis_id = district_province.dis_id
AND district_province.pro_id = province_name.pro_id
GROUP BY province_name.pro_id , province_name.pro_name;
```

```
anonymous block completed
PRO_ID          PRO_NAME          AMOUNT
-----
41              อุดรธานี         6
39              หนองบัวลำภู     4

2 rows selected
9ms elapsed
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.5 สอบถามจำนวนประชากร ปี 2500-2535 (1957-1992) จะแสดงจำนวนประชากรได้ถูกต้อง คือ จำนวนประชากร ณ ช่วงเวลาดังกล่าว จะมีเฉพาะแค่ จ.อุดรธานี = 10 คน

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('1957','YYYY'),
TO_DATE('1992','YYYY') );

SELECT province_name.pro_id, province_name.pro_name,
Count(personal_address.citizen_id) AS AMOUNT
FROM ((personal_address JOIN address_subdistrict ON
personal_address.add_id = address_subdistrict.add_id)
JOIN subdistrict_district ON address_subdistrict.sub_id =
subdistrict_district.sub_id)
JOIN district_province ON subdistrict_district.dis_id =
district_province.dis_id)
JOIN province_name ON district_province.pro_id = province_name.pro_id
GROUP BY province_name.pro_id, province_name.pro_name;
```

```
anonymous block completed
PRO_ID          PRO_NAME          AMOUNT
-----
41              อุดรธานี          10
1 rows selected

8ms elapsed
```

เขียนคำสั่งแบบที่ 2 สามารถตอบข้อ 9.5 ได้ผลลัพธ์เหมือนกัน

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('1957','YYYY'),
TO_DATE('1992','YYYY') );

SELECT province_name.pro_id, province_name.pro_name,
Count(personal_address.citizen_id) AS AMOUNT
FROM personal_address, address_subdistrict, subdistrict_district,
district_province, province_name
WHERE personal_address.add_id = address_subdistrict.add_id
AND address_subdistrict.sub_id = subdistrict_district.sub_id
AND subdistrict_district.dis_id = district_province.dis_id
AND district_province.pro_id = province_name.pro_id
GROUP BY province_name.pro_id , province_name.pro_name;
```

```
anonymous block completed
PRO_ID          PRO_NAME          AMOUNT
-----
41              อุดรธานี          10
1 rows selected

7ms elapsed
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.6 สอบถามจำนวนประชากร ทุกช่วงเวลา ตั้งแต่อดีต ถึง ปัจจุบัน จะแสดงจำนวนประชากร
ได้ถูกต้อง คือ จำนวนประชากร จ.อุดรธานี = 10 คน และ จ.หนองบัวลำภู = 4 คน

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT province_name.pro_id, province_name.pro_name,
Count(personal_address.citizen_id) AS AMOUNT
FROM ((personal_address JOIN address_subdistrict ON
personal_address.add_id = address_subdistrict.add_id)
JOIN subdistrict_district ON address_subdistrict.sub_id =
subdistrict_district.sub_id)
JOIN district_province ON subdistrict_district.dis_id =
district_province.dis_id)
JOIN province_name ON district_province.pro_id = province_name.pro_id
GROUP BY province_name.pro_id, province_name.pro_name;
```

```
anonymous block completed
PRO_ID          PRO_NAME          AMOUNT
-----
41              อุดรธานี          10
39              หนองบัวลำภู      4

2 rows selected

8ms elapsed
```

เขียนคำสั่งแบบที่ 2 สามารถตอบข้อ 9.6 ได้ผลลัพธ์เหมือนกัน

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT province_name.pro_id, province_name.pro_name,
Count(personal_address.citizen_id) AS AMOUNT
FROM personal_address, address_subdistrict, subdistrict_district,
district_province, province_name
WHERE personal_address.add_id = address_subdistrict.add_id
AND address_subdistrict.sub_id = subdistrict_district.sub_id
AND subdistrict_district.dis_id = district_province.dis_id
AND district_province.pro_id = province_name.pro_id
GROUP BY province_name.pro_id, province_name.pro_name;
```

```
anonymous block completed
PRO_ID          PRO_NAME          AMOUNT
-----
41              อุดรธานี          10
39              หนองบัวลำภู      4

2 rows selected

8ms elapsed
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.7 สอบถามจำนวนประชากรแต่ละอำเภอ ณ ช่วงเวลาปัจจุบัน

```
EXECUTE DBMS_WM.SetValidTime();
```

```
SELECT district_name.dis_id, district_name.dis_name,
Count(personal_address.citizen_id) AS AMOUNT
FROM ((personal_address JOIN address_subdistrict ON
personal_address.add_id = address_subdistrict.add_id)
JOIN subdistrict_district ON address_subdistrict.sub_id =
subdistrict_district.sub_id)
JOIN district_name ON subdistrict_district.dis_id =
district_name.dis_id
GROUP BY district_name.dis_id, district_name.dis_name;
```

anonymous block completed

DIS_ID	DIS_NAME	AMOUNT
4101	เมืองอุดรธานี	6
4102	เมืองหนองบัวลำภู	4

2 rows selected

7ms elapsed

เขียนคำสั่งแบบที่ 2 สามารถตอบคำถามข้อ 9.7 ได้ผลลัพธ์เหมือนกัน

```
EXECUTE DBMS_WM.SetValidTime();
```

```
SELECT district_name.dis_id, district_name.dis_name,
Count(personal_address.citizen_id) AS AMOUNT
FROM personal_address, address_subdistrict, subdistrict_district ,
district_name
WHERE personal_address.add_id = address_subdistrict.add_id
AND address_subdistrict.sub_id = subdistrict_district.sub_id
AND subdistrict_district.dis_id = district_name.dis_id
GROUP BY district_name.dis_id, district_name.dis_name;
```

anonymous block completed

DIS_ID	DIS_NAME	AMOUNT
4101	เมืองอุดรธานี	6
4102	เมืองหนองบัวลำภู	4

2 rows selected

7ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.8 สอบถามจำนวนประชากรแต่ละอำเภอ ทุกช่วงเวลา ตั้งแต่อดีต จนถึงปัจจุบัน

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);
```

```
SELECT district_name.dis_id, district_name.dis_name,
Count(personal_address.citizen_id) AS AMOUNT
FROM ((personal_address JOIN address_subdistrict ON
personal_address.add_id = address_subdistrict.add_id)
JOIN subdistrict_district ON address_subdistrict.sub_id =
subdistrict_district.sub_id)
JOIN district_name ON subdistrict_district.dis_id =
district_name.dis_id
GROUP BY district_name.dis_id, district_name.dis_name;
```

anonymous block completed

DIS_ID	DIS_NAME	AMOUNT
4101	เมืองสุพรรณ	6
4102	เมืองหนองบัวลำภู	4
4102	หนองบัวลำภู	4

3 rows selected

8ms elapsed

เขียนคำสั่งแบบที่ 2 สามารถตอบคำถามข้อ 9.8 ได้ผลลัพธ์เหมือนกัน

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);
```

```
SELECT district_name.dis_id, district_name.dis_name,
Count(personal_address.citizen_id) AS AMOUNT
FROM personal_address, address_subdistrict, subdistrict_district ,
district_name
WHERE personal_address.add_id = address_subdistrict.add_id
AND address_subdistrict.sub_id = subdistrict_district.sub_id
AND subdistrict_district.dis_id = district_name.dis_id
GROUP BY district_name.dis_id, district_name.dis_name;
```

anonymous block completed

DIS_ID	DIS_NAME	AMOUNT
4101	เมืองสุพรรณ	6
4102	เมืองหนองบัวลำภู	4
4102	หนองบัวลำภู	4

3 rows selected

6ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. สรุปผลการทดลอง

จากการทดลอง สามารถบันทึกเปลี่ยนแปลงที่อยู่สัมพันธ์กับเวลาได้ และสอบถามข้อมูลพื้นที่ และสถิติที่สัมพันธ์เชิงเวลา ทั้งในอดีต ปัจจุบัน ได้อย่างถูกต้องตามแต่ละช่วงเวลา

11. การยกเลิกตารางแบบเชิงเวลาและลบตาราง เพื่อเริ่มต้นการทดลองด้านประวัติที่อยู่

การยกเลิกตารางแบบเชิงเวลา จะใช้คำสั่ง **DisableVersioning** และลบตาราง ที่สัมพันธ์ **Address** ทั้งหมด เพื่อเริ่มต้นการทดลอง

```
EXECUTE DBMS_WM.DisableVersioning('province_name');
EXECUTE DBMS_WM.DisableVersioning('district_name');
EXECUTE DBMS_WM.DisableVersioning('district_province');
EXECUTE DBMS_WM.DisableVersioning('subdistrict_name');
EXECUTE DBMS_WM.DisableVersioning('subdistrict_district');
EXECUTE DBMS_WM.DisableVersioning('road_name');
EXECUTE DBMS_WM.DisableVersioning('address_subdistrict');
EXECUTE DBMS_WM.DisableVersioning('personal_address');

drop table province_name;
drop table district_name;
drop table district_province;
drop table subdistrict_name;
drop table subdistrict_district;
drop table road_name;
drop table address;
drop table address_subdistrict;
drop table personal_address;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

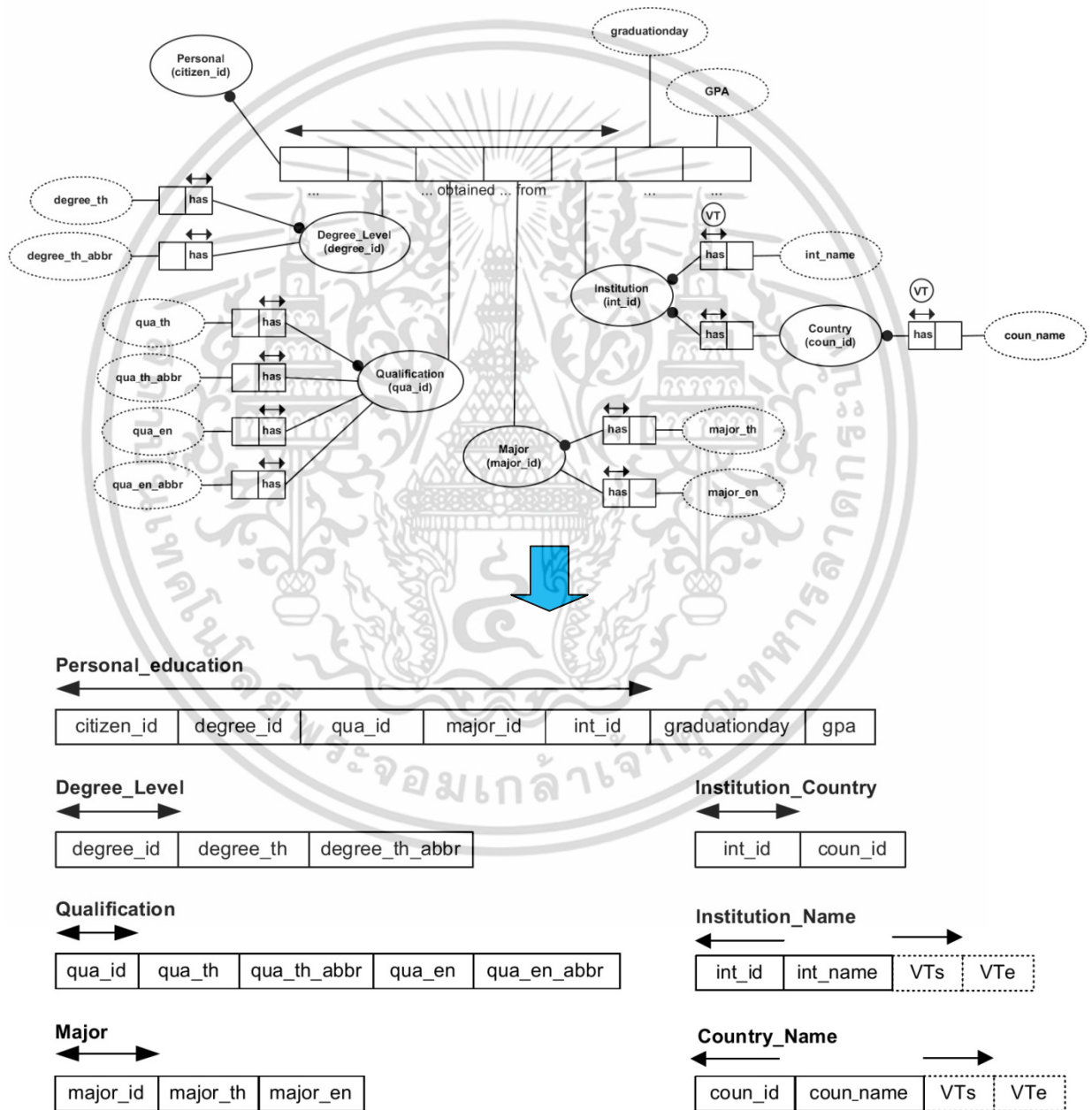


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

การทดลองบันทึกแก้ไขข้อมูลด้านประวัติการศึกษา

1. **TORM** ประวัติการศึกษา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ตารางข้อมูลตัวอย่างสมมุติ ประวัติทางการศึกษา

เลข ปชช	คำ นำหน้า	ชื่อ	นามสกุล	ระดับ	วุฒิ	สาขา	สถานศึกษา	ว.ด.ป. สำเร็จ	GPA
33333	นาย	นเรศ	อุดรธรรม	ป.ตรี	วศ.บ.	วิศวกรรมโทรคมนาคม	สถาบันเทคโนโลยีพระ จอมเกล้า ศูนย์นนทบุรี	31/03/2517	4.00
				ป.โท	วศ.ม.	วิศวกรรมไฟฟ้า	สถาบันเทคโนโลยีพระ จอมเกล้า วิทยาเขต นนทบุรีสาทรกระบ้ง	31/03/2520	4.00
				ป. เอก	วศ.ด	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีพระ จอมเกล้าเจ้าคุณทหาร ลาดกระบ้ง	31/03/2550	4.00
35003	นาย	สมชาติ	ทตเจริญ	ป.ตรี	วศ.บ.	วิศวกรรมไฟฟ้า	วิทยาลัยเทคโนโลยีและ อาชีวศึกษา	31/03/2525	3.50
				ป.โท	วศ.ม.	วิศวกรรมไฟฟ้า	สถาบันเทคโนโลยีพระ จอมเกล้าพระนครเหนือ	31/03/2530	4.00
66666	นาย	สิทธิ	รักถนอม	ป.ตรี	วศ.บ.	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีราช มงคล	31/03/2544	3.00
				ป.โท	วศ.ม.	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีพระ จอมเกล้าเจ้าคุณทหาร ลาดกระบ้ง	31/03/2557	3.60
88888	น.ส.	กนกพร	บุญรักษา	ป.ตรี	วศ.บ.	วิศวกรรมคอมพิวเตอร์	มหาวิทยาลัยเทคโนโลยี ราชมงคลธัญบุรี	31/03/2550	4.00

จากตารางข้อมูลตัวอย่างสมมุตินี้ ผู้ที่สำเร็จการศึกษาจาก มหาวิทยาลัยเทคโนโลยีราชมงคล
ธัญบุรี มีทั้งหมด 3 คน คือ นายสมชาติ ทตเจริญ นายสิทธิ รักถนอม และ น.ส.กนกพร บุญรักษา แต่
ชื่อของสถาบันต่างกัน เพราะสถาบันได้เปลี่ยนชื่อไปแล้วถึง 3 ครั้ง คือ วิทยาลัยเทคโนโลยีและ
อาชีวศึกษา สถาบันเทคโนโลยีราชมงคล และมหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

สถานศึกษา : สจล. (www.kmitl.ac.th)

- | | |
|--|----------------------|
| 1. วิทยาลัยโทรคมนาคมนนทบุรี | 1 พ.ค. 2507 (1964) |
| 2. สถาบันเทคโนโลยีพระจอมเกล้า ศูนย์นนทบุรี | 24 ส.ค. 2514 (1971) |
| 3. สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตนนทบุรีสาทรกระบ้ง | 29 มิ.ย. 2517 (1974) |
| 4. สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตเจ้าคุณทหารลาดกระบ้ง | 22 เม.ย. 2522 (1979) |
| 5. สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบ้ง | 20 ก.พ. 2529 (1986) |

สถานศึกษา : มทร.ธัญบุรี (www.rmutt.ac.th)

- | | |
|---------------------------------------|---------------------|
| 1. วิทยาลัยเทคโนโลยีและอาชีวศึกษา | 27 ก.พ. 2518 (1975) |
| 2. สถาบันเทคโนโลยีราชมงคล | 15 ก.ย. 2531 (1988) |
| 3. มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี | 18 ม.ค. 2548 (2005) |

สถานศึกษา : มจพ. (www.kmutnb.ac.th)

- | | |
|---|----------------------|
| 1. สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตพระนครเหนือ | 24 เม.ย. 2514 (1971) |
| 2. สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ | 20 ก.พ. 2529 (1986) |
| 3. มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ | 27 ธ.ค. 2550 (2007) |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ทดลองสมมุติข้อมูลในตาราง **TORM**

ตาราง 3.1 บันทึกรายประวัติการศึกษา (Personal_Education)

citizen_id	degree_id	qua_id	major_id	int_id	graduationday	gpa
33333	40	159	22	1600	31/03/1974	4.0
33333	60	209	12	1600	31/03/1977	4.0
33333	80	204	12	1600	31/03/2007	4.0
35003	40	206	11	19100	31/03/1982	3.5
35003	60	209	11	1500	31/03/1987	4.0
66666	40	206	12	19100	31/03/2001	3.0
66666	60	209	12	1600	31/03/2014	3.6
88888	40	206	12	19100	31/03/2007	4.0

ตาราง 3.2 บันทึกรายตัวอย่างระดับการศึกษา (Degree_Level)

degree_id	degree_th	degree_th_abbr
40	ปริญญาตรี	ป.ตรี
50	ประกาศนียบัตรบัณฑิต	ป.บัณฑิต
60	ปริญญาโท	ป.โท
70	ประกาศนียบัตรบัณฑิตชั้นสูง	ป.บัณฑิตชั้นสูง
80	ปริญญาเอก	ป.เอก

ตาราง 3.3 บันทึกรายตัวอย่างวุฒิการศึกษา (Qualification)

qua_id	qua_th	qua_th_abbr	qua_en	qua_en_abbr
206	วิศวกรรมศาสตรบัณฑิต	วศ.บ.	Bachelor of Engineering	B.Eng.
209	วิศวกรรมศาสตรมหาบัณฑิต	วศ.ม.	Master of Engineering	M.Eng.
204	วิศวกรรมศาสตรดุษฎีบัณฑิต	วศ.ด.	Doctor of Engineering	D.Eng.
159	วิทยาศาสตร์บัณฑิต	วท.บ.	Bachelor of Science	B.S.

ตาราง 3.4 บันทึกรายตัวอย่างสาขา (Major)

major_id	major_th	major_en
11	วิศวกรรมไฟฟ้า	Electrical Engineering
12	วิศวกรรมคอมพิวเตอร์	Computer Engineering
13	วิศวกรรมโทรคมนาคม	Telecommunication Engineering
21	เทคโนโลยีชีวภาพ	Biotechnology
22	วิทยาการคอมพิวเตอร์	Computer science

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.5 บันทึกสถานศึกษา (Institution_Country)

Int_id	coun_id
1600	10
1500	10
19100	10

ตาราง 3.6 บันทึกชื่อสถาบัน (Institution_name)

int_id	int_name	VTs	VTe
1600	วิทยาลัยโทรคมนาคมนนทบุรี	01/05/1964	24/04/1971
1600	สถาบันเทคโนโลยีพระจอมเกล้า ศูนย์นนทบุรี	24/04/1971	29/06/1974
1600	สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตนนทบุรีลาดกระบัง	29/06/1974	22/04/1979
1600	สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตเจ้าคุณทหารลาดกระบัง	22/04/1979	20/02/1986
1600	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	20/02/1986	
1500	สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตพระนครเหนือ	24/04/1971	20/02/1986
1500	สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ	20/02/1986	27/12/2007
1500	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ	27/12/2007	
19100	วิทยาลัยเทคโนโลยีและอาชีวศึกษา	27/02/1975	15/09/1988
19100	สถาบันเทคโนโลยีสยามมงคล	15/09/1988	18/01/2005
19100	มหาวิทยาลัยเทคโนโลยีสยามมงคลธัญบุรี	18/01/2005	

ตาราง 3.7 บันทึกชื่อประเทศ (country_name)

coun_id	coun_name	VTs	VTe
10	สยาม	06/04/1782	24/06/1939
10	ไทย	24/06/1939	

วันสถาปนากรุงรัตนโกสินทร์ 6 เมษายน 2325 (1782)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทดลองคำสั่ง **Oracle 11g** ตามการออกแบบ **TORM** ประวัติการศึกษา

4.1 สร้างตาราง จำนวน 7 ตาราง ดังนี้ 1. Personal_Education 2. Degree 3. Qualification 4. Major 5. Institution 6. Institution_name 7. count_name

```

CREATE TABLE personal_education(
  citizen_id          NUMBER NOT NULL,
  degree_id           NUMBER NOT NULL,
  qua_id              NUMBER NOT NULL,
  major_id            NUMBER NOT NULL,
  int_id              NUMBER NOT NULL,
  graduationday       DATE,
  gpa                 NUMBER,
  CONSTRAINT personal_education_pk PRIMARY KEY(citizen_id
, degree_id, qua_id, major_id,int_id));

CREATE TABLE degree_level (
  degree_id           NUMBER PRIMARY KEY,
  degree_th           VARCHAR(50),
  degree_th_abbr     VARCHAR(20));

CREATE TABLE qualification (
  qua_id              NUMBER PRIMARY KEY,
  qua_th              VARCHAR(50),
  qua_th_abbr        VARCHAR(20),
  qua_en              VARCHAR(50),
  qua_en_abbr        VARCHAR(20));

CREATE TABLE major (
  major_id            NUMBER PRIMARY KEY,
  major_th           VARCHAR(50),
  major_en            VARCHAR(50));

CREATE TABLE institution_country (
  int_id              NUMBER PRIMARY KEY,
  coun_id             NUMBER );

CREATE TABLE institution_name (
  int_id              NUMBER PRIMARY KEY,
  int_name            VARCHAR(100));

CREATE TABLE country_name (
  coun_id             NUMBER PRIMARY KEY,
  coun_name           VARCHAR(100));

commit;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ใช้คำสั่ง EnableVersioning ทำให้ตาราง institution_name และตาราง country_name

เป็นตารางแบบเชิงเวลา

```
EXECUTE DBMS_WM.EnableVersioning('institution_name',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

EXECUTE DBMS_WM.EnableVersioning('country_name',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

commit;
```

4.3 ทำการเพิ่มข้อมูลตัวอย่างระดับการศึกษา (Degree) ตามตารางที่ 3.2

```
INSERT INTO degree_level VALUES (40, 'ปริญญาตรี', 'ป.ตรี');
INSERT INTO degree_level VALUES (50, 'ประกาศนียบัตรบัณฑิต', 'ป.บัณฑิต');
INSERT INTO degree_level VALUES (60, 'ปริญญาโท', 'ป.โท');
INSERT INTO degree_level VALUES (70, 'ประกาศนียบัตรบัณฑิตชั้นสูง', 'ป.บัณฑิต
ชั้นสูง');
INSERT INTO degree_level VALUES (80, 'ปริญญาเอก', 'ป.เอก');

commit;
```

4.4 ทำการเพิ่มข้อมูลตัวอย่างวุฒิการศึกษา (Qualification) ตามตารางที่ 3.3

```
INSERT INTO qualification VALUES (206, 'วิศวกรรมศาสตรบัณฑิต', 'วศ.บ.',
' Bachelor of Engineering ', 'B.Eng. ');

INSERT INTO qualification VALUES (209, 'วิศวกรรมศาสตรมหาบัณฑิต', 'วศ.ม.',
' Master of Engineering ', 'M.Eng. ');

INSERT INTO qualification VALUES (204, 'วิศวกรรมศาสตรดุษฎีบัณฑิต', 'วศ.ด.',
' Doctor of Engineering ', 'D.Eng. ');

INSERT INTO qualification VALUES (159, 'วิทยาศาสตร์บัณฑิต', 'วท.บ.',
' Bachelor of Science ', 'B.S. ');

commit;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 ทำการเพิ่มตัวอย่างสาขา (Major) ตามตารางที่ 3.4

```

INSERT INTO major VALUES( 11, 'วิศวกรรมไฟฟ้า', 'Electrical Engineering' );
INSERT INTO major VALUES( 12, 'วิศวกรรมคอมพิวเตอร์', 'Computer Engineering' );
INSERT INTO major VALUES( 13, 'วิศวกรรมโทรคมนาคม', 'Telecommunication
Engineering' );
INSERT INTO major VALUES( 21, 'เทคโนโลยีชีวภาพ', 'Biotechnology' );
INSERT INTO major VALUES( 22, 'วิทยาการคอมพิวเตอร์', 'Computer science' );

commit;

```

4.5 ทำการเพิ่มตัวอย่างชื่อประเทศ (country_name) ตามตารางที่ 3.7

- 1) ทำการเพิ่มชื่อประเทศสยาม ในตาราง country_name
- 2) ทำการปรับปรุงชื่อประเทศสยาม เป็นชื่อประเทศไทย

```

-- 1) เพิ่มชื่อประเทศสยาม
INSERT INTO country_name VALUES( 10, 'สยาม', WMSYS.WM_PERIOD
(TO_DATE('06-04-1782', 'DD-MM-YYYY'), DBMS_WM.MAX_TIME) );

-- 2) ปรับปรุงชื่อประเทศสยาม เป็นชื่อประเทศไทย
EXECUTE DBMS_WM.SetValidTime(TO_DATE('24-06-1939', 'DD-MM-
YYYY'), DBMS_WM.MAX_TIME);

UPDATE country_name SET coun_name = 'ไทย' WHERE coun_id = 10;
commit;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ทำการเพิ่มตัวอย่างชื่อสถานศึกษา ดังนี้

--1) เพิ่มชื่อ สถานศึกษา ทั้ง 3 สถาบัน

```
INSERT INTO institution_name VALUES ( 1600, 'วิทยาลัยโทรคมนาคมนนทบุรี',
WMSYS.WM_PERIOD (TO_DATE('01-05-1964', 'DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));
```

```
INSERT INTO institution_name VALUES ( 1500, 'สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตพระนครเหนือ',
WMSYS.WM_PERIOD (TO_DATE('24-04-1971', 'DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));
```

```
INSERT INTO institution_name VALUES ( 19100, 'วิทยาลัยเทคโนโลยีและอาชีวศึกษา',
WMSYS.WM_PERIOD (TO_DATE('27-02-1975', 'DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));
```

--2) ปรับปรุงชื่อ สจล. ครั้งที่ 1

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('24-04-1971', 'DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED);
```

```
UPDATE institution_name SET int_name = 'สถาบันเทคโนโลยีพระจอมเกล้า ศูนย์
นนทบุรี' WHERE int_id =1600;
```

--3) ปรับปรุงชื่อ สจล. ครั้งที่ 2

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('29-06-1974', 'DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED);
```

```
UPDATE institution_name SET int_name = 'สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขต
นนทบุรีลาดกระบัง' WHERE int_id =1600;
```

--4) ปรับปรุงชื่อ สจล. ครั้งที่ 3

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('22-04-1979', 'DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED);
```

```
UPDATE institution_name SET int_name = 'สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขต
เจ้าคุณทหารลาดกระบัง' WHERE int_id =1600;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

--5)ปรับปรุงชื่อ สจล. ครั้งที่ 4 และปรับปรุงชื่อ สจพ. ครั้งที่ 1

      EXECUTE      DBMS_WM.SetValidTime(TO_DATE('20-02-1986','DD-MM-
YYYY'), DBMS_WM.UNTIL_CHANGED);

      UPDATE institution_name SET int_name = 'สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ
ทหารลาดกระบัง' WHERE int_id =1600;

      UPDATE institution_name SET int_name = 'สถาบันเทคโนโลยีพระจอมเกล้าพระนคร
เหนือ' WHERE int_id =1500;

--6)ปรับปรุงชื่อ ส.ราชมงคล ครั้งที่ 1

      EXECUTE      DBMS_WM.SetValidTime(TO_DATE('15-09-1988','DD-MM-
YYYY'), DBMS_WM.UNTIL_CHANGED);

      UPDATE institution_name SET int_name = 'สถาบันเทคโนโลยีราชมงคล' WHERE
int_id =19100;

--7)ปรับปรุงชื่อ ส.ราชมงคล ครั้งที่ 2 เป็นมหาวิทยาลัย

      EXECUTE      DBMS_WM.SetValidTime(TO_DATE('18-01-2005','DD-MM-
YYYY'), DBMS_WM.UNTIL_CHANGED);

      UPDATE institution_name SET int_name = 'มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี'
WHERE int_id =19100;

--8)ปรับปรุงชื่อ สจพ. เป็นมหาวิทยาลัย

      EXECUTE      DBMS_WM.SetValidTime(TO_DATE('27-12-2007','DD-MM-
YYYY'), DBMS_WM.UNTIL_CHANGED);

      UPDATE institution_name SET int_name = 'มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระ
นครเหนือ' WHERE int_id =1500;

commit;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. แสดงประวัติการเปลี่ยนชื่อของสถานศึกษา จัดเรียงลำดับตามรหัส และวันที่เปลี่ยนแปลงชื่อ

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);
```

```
SELECT int_id AS "ID", int_name,
TO_CHAR(e.WM_VALID.VALIDFROM, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') FROM_DATE,
TO_CHAR(e.WM_VALID.VALIDTILL, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE
FROM institution_name e
ORDER BY int_id, WM_VALID.VALIDFROM;
```

ID	INT_NAME	FROM_DATE	TO_DATE
1500	สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตพระนครเหนือ	24-04-2514	20-02-2529
1500	สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ	20-02-2529	27-12-2550
1500	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ	27-12-2550	
1600	วิทยาลัยโศภนคมนาคนนทบุรี	01-05-2507	24-04-2514
1600	สถาบันเทคโนโลยีพระจอมเกล้า ศูนย์นนทบุรี	24-04-2514	29-06-2517
1600	สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตนนทบุรีลาดกระบัง	29-06-2517	22-04-2522
1600	สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตเจ้าคุณทหารลาดกระบัง	22-04-2522	20-02-2529
1600	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	20-02-2529	
19100	วิทยาลัยเทคโนโลยีและอาชีวศึกษา	27-02-2518	15-09-2531
19100	สถาบันเทคโนโลยีราชมงคล	15-09-2531	18-01-2548
19100	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	18-01-2548	

11 rows selected

9ms elapsed

7. ทำการเพิ่มรหัสสถานศึกษา ลงในตาราง Institution

```
INSERT INTO institution_country VALUES( 1600,10);
INSERT INTO institution_country VALUES( 1500,10);
INSERT INTO institution_country VALUES(19100,10);
```

8. ทำการเพิ่มประวัติการศึกษา ลงในตาราง Personal_Education

```
INSERT INTO Personal_Education VALUES('33333',40,206,13,
1600, TO_DATE('31-03-1974', 'DD-MM-YYYY'),4.0);
INSERT INTO Personal_Education VALUES('33333',60,209,11,
1600, TO_DATE('31-03-1977', 'DD-MM-YYYY'),4.0);
INSERT INTO Personal_Education VALUES('33333',80,204,12,
1600, TO_DATE('31-03-2007', 'DD-MM-YYYY'),4.0);
INSERT INTO Personal_Education VALUES('35003',40,206,11,
19100, TO_DATE('31-03-1982', 'DD-MM-YYYY'),3.5);
INSERT INTO Personal_Education VALUES('35003',60,209,11,
1500, TO_DATE('31-03-1987', 'DD-MM-YYYY'),4.0);
INSERT INTO Personal_Education VALUES('66666',40,206,12,
19100, TO_DATE('31-03-2001', 'DD-MM-YYYY'),3.0);
INSERT INTO Personal_Education VALUES('66666',60,209,12,
1600, TO_DATE('31-03-2014', 'DD-MM-YYYY'),3.6);
INSERT INTO Personal_Education VALUES('88888',40,206,12,
19100, TO_DATE('31-03-2007', 'DD-MM-YYYY'),4.0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. แสดงประวัติการศึกษา ในช่วงเวลาปัจจุบัน

คำสั่งแบบที่ 1

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT citizen_id, degree_th_abbr AS "ระดับ", qua_th_abbr AS "วุฒิ",
major_th AS "สาขา", int_name AS "สถานศึกษา",
TO_CHAR(graduationday, 'yyyy')+543 AS "พ.ศ."
FROM ((personal_education JOIN degree_level ON
personal_education.degree_id = degree_level.degree_id)
JOIN qualification ON personal_education.qua_id =
qualification.qua_id)
JOIN major ON personal_education.major_id = major.major_id )
JOIN institution_name ON personal_education.int_id =
institution_name.int_id
ORDER BY citizen_id , degree_level.degree_id;
```

คำสั่งแบบที่ 2

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT citizen_id, degree_th_abbr AS "ระดับ", qua_th_abbr AS "วุฒิ",
major_th AS "สาขา", int_name AS "สถานศึกษา",
TO_CHAR(graduationday, 'yyyy')+543 AS "พ.ศ."
FROM personal_education, degree_level, qualification, major,
institution_name
WHERE personal_education.degree_id = degree_level.degree_id
AND personal_education.qua_id = qualification.qua_id
AND personal_education.major_id = major.major_id
AND personal_education.int_id = institution_name.int_id
ORDER BY citizen_id , degree_level.degree_id;
```

anonymous block completed	CITIZEN_ID	ระดับ	วุฒิ	สาขา	สถานศึกษา	พ.ศ.
	33333	ป.ตรี	วศ.บ.	วิศวกรรมโทรคมนาคม	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	2517
	33333	ป.โท	วศ.ม.	วิศวกรรมไฟฟ้า	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	2520
	33333	ป.เอก	วศ.ด.	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	2550
	35003	ป.ตรี	วศ.บ.	วิศวกรรมไฟฟ้า	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	2525
	35003	ป.โท	วศ.ม.	วิศวกรรมไฟฟ้า	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ	2530
	66666	ป.ตรี	วศ.บ.	วิศวกรรมคอมพิวเตอร์	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	2544
	66666	ป.โท	วศ.ม.	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	2557
	88888	ป.ตรี	วศ.บ.	วิศวกรรมคอมพิวเตอร์	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	2550

8 rows selected

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. แสดงชื่อสถาบัน ณ ปี พ.ศ. ที่สำเร็จ ของบุคลากรทุกคน (นำวันสำเร็จการศึกษามาเปรียบเทียบกับ
เชิงเวลากับ ช่วงเวลาการเปลี่ยนแปลงชื่อสถาบัน)

คำสั่งแบบที่ 1

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT citizen_id, TO_CHAR(graduationday,'yyyy')+543 AS "YEAR" ,
degree_th_abbr AS "DEGREE", institution_name.int_id AS "INT_ID",
int_name AS "institution"

FROM ((personal_education JOIN degree_level ON
personal_education.degree_id = degree_level.degree_id)
JOIN institution_name ON personal_education.int_id =
institution_name.int_id)

WHERE WM_OVERLAPS(institution_name.WM_VALID,
WM_PERIOD(personal_education.graduationday,
personal_education.graduationday))=1

ORDER BY citizen_id , degree_level.degree_id;
```

คำสั่งแบบที่ 2

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT citizen_id, TO_CHAR(graduationday,'yyyy')+543 AS "YEAR" ,
degree_th_abbr AS "DEGREE", institution_name.int_id AS "INT_ID",
int_name AS "institution"

FROM personal_education, degree_level, institution_name
WHERE personal_education.degree_id = degree_level.degree_id
AND personal_education.int_id = institution_name.int_id
AND WM_OVERLAPS(institution_name.WM_VALID,
WM_PERIOD(personal_education.graduationday,
personal_education.graduationday))=1

ORDER BY citizen_id , degree_level.degree_id;
```

CITIZEN_ID	YEAR	DEGREE	INT_ID	institution
33333	2517	ป.ตรี	1600	สถาบันเทคโนโลยีพระจอมเกล้า ศูนย์นนทบุรี
33333	2520	ป.โท	1600	สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตนนทบุรีลาดกระบัง
33333	2550	ป.เอก	1600	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
35003	2525	ป.ตรี	19100	วิทยาลัยเทคโนโลยีและอาชีวศึกษา
35003	2530	ป.โท	1500	สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ
66666	2544	ป.ตรี	19100	สถาบันเทคโนโลยีราชมงคล
66666	2557	ป.โท	1600	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
88888	2550	ป.ตรี	19100	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

8 rows selected

8ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. แสดงชื่อสถาบัน ณ วันปัจจุบัน ของบุคลากรทุกคน

คำสั่งแบบที่ 1

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT citizen_id, TO_CHAR(graduationday,'yyyy')+543 AS "YEAR" ,
degree_th_abbr AS "DEGREE", institution_name.int_id AS "INT_ID",
int_name AS "institution"

FROM ((personal_education JOIN degree_level ON
personal_education.degree_id = degree_level.degree_id)
JOIN institution_name ON personal_education.int_id =
institution_name.int_id)

ORDER BY citizen_id , degree_level.degree_id;
```

คำสั่งแบบที่ 2

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT citizen_id, TO_CHAR(graduationday,'yyyy')+543 AS "YEAR" ,
degree_th_abbr AS "DEGREE", institution_name.int_id AS "INT_ID",
int_name AS "institution"

FROM personal_education, degree_level, institution_name
WHERE personal_education.degree_id = degree_level.degree_id
AND personal_education.int_id = institution_name.int_id

ORDER BY citizen_id , degree_level.degree_id;
```

CITIZEN_ID	YEAR	DEGREE	INT_ID	institution
33333	2517	ป.ตรี	1600	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
33333	2520	ป.โท	1600	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
33333	2550	ป.เอก	1600	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
35003	2525	ป.ตรี	19100	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี
35003	2530	ป.โท	1500	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
66666	2544	ป.ตรี	19100	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี
66666	2557	ป.โท	1600	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
88888	2550	ป.ตรี	19100	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

8 rows selected

6ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. แสดงวุฒิการศึกษา ของพนักงาน เลขประจำตัวประชาชน 6666

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('1989','YYYY'),
TO_DATE('2004','YYYY') );

SELECT citizen_id, degree_th_abbr AS "ระดับ", qua_th_abbr AS "วุฒิ",
major_th AS "สาขา", int_name AS "สถานศึกษา",
TO_CHAR(graduationday,'yyyy')+543 AS "พ.ศ."
FROM ((personal_education JOIN degree_level ON
personal_education.degree_id = degree_level.degree_id)
JOIN qualification ON personal_education.qua_id =
qualification.qua_id)
JOIN major ON personal_education.major_id = major.major_id )
JOIN institution_name ON personal_education.int_id =
institution_name.int_id
WHERE citizen_id = '66666'
ORDER BY citizen_id , degree_level.degree_id;
```

anonymous block completed

CITIZEN_ID	ระดับ	วุฒิ	สาขา	สถานศึกษา	พ.ศ.
66666	ป.ตรี	วศ.บ.	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีราชมงคล	2544
66666	ป.โท	วศ.ม.	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	2547

2 rows selected

13. มีพนักงานที่สำเร็จการศึกษาจากสถาบันใดบ้าง และเป็นจำนวนเท่าใด ณ เวลาปัจจุบัน คำสั่งแบบที่ 1

```
SELECT personal_education.int_id, institution_name.int_name,
Count(personal_education.citizen_id) AS AMOUNT
FROM personal_education JOIN institution_name ON
personal_education.int_id = institution_name.int_id
GROUP BY personal_education.int_id, institution_name.int_name;
```

คำสั่งแบบที่ 2

```
SELECT personal_education.int_id, institution_name.int_name,
Count(personal_education.citizen_id) AS AMOUNT
FROM personal_education , institution_name
WHERE personal_education.int_id = institution_name.int_id
GROUP BY personal_education.int_id, institution_name.int_name;
```

INT_ID	INT_NAME	AMOUNT
1500	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ	1
19100	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	3
1600	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14. แสดงประวัติการศึกษา ของบุคลากร ณ วันสำเร็จการศึกษา

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT citizen_id, degree_th_abbr AS "DEGREE",
qualification.qua_th_abbr AS "วุฒิ", major.major_th as "สาขา" ,
int_name AS "institution" , TO_CHAR(graduationday,'yyyy')+543 AS
"พ.ศ."

FROM personal_education, degree_level, qualification, major,
institution_name
WHERE personal_education.degree_id = degree_level.degree_id
AND personal_education.qua_id = qualification.qua_id
AND personal_education.major_id = major.major_id
AND personal_education.int_id = institution_name.int_id
AND WM_OVERLAPS(institution_name.WM_VALID,
WM_PERIOD(personal_education.graduationday,
personal_education.graduationday))=1

ORDER BY citizen_id , degree_level.degree_id;
```

anonymous block completed

CITIZEN_ID	ระดับ	วุฒิ	สาขา	สถานศึกษา	พ.ศ.
33333	ป.ตรี	วศ.บ.	วิศวกรรมโทรคมนาคม	สถาบันเทคโนโลยีพระจอมเกล้า ศูนย์นนทบุรี	2517
33333	ป.โท	วศ.ม.	วิศวกรรมไฟฟ้า	สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตนนทบุรีลาดกระบัง	2520
33333	ป.เอก	วศ.ด.	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	2550
35003	ป.ตรี	วศ.บ.	วิศวกรรมไฟฟ้า	วิทยาลัยเทคโนโลยีและอาชีวศึกษา	2525
35003	ป.โท	วศ.ม.	วิศวกรรมไฟฟ้า	สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ	2530
66666	ป.ตรี	วศ.บ.	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีราชมงคล	2544
66666	ป.โท	วศ.ม.	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	2557
88888	ป.ตรี	วศ.บ.	วิศวกรรมคอมพิวเตอร์	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	2550

8 rows selected

192ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

15. สรุปผลการทดลอง

จากการทดลอง สามารถบันทึกข้อมูล ปรับปรุงข้อมูล และสอบถามข้อมูลประวัติการศึกษา ที่สัมพันธ์กับเวลาได้ถูกต้อง

16. คำสั่งการยกเลิกตารางแบบเชิงเวลาและลบตาราง เพื่อเริ่มต้นการทดลอง

การยกเลิกตารางแบบเชิงเวลา จะใช้คำสั่ง **DisableVersioning** และลบตาราง ที่สัมพันธ์ **Address** ทั้งหมด เพื่อเริ่มต้นการทดลอง

```
EXECUTE DBMS_WM.DisableVersioning('country_name');
EXECUTE DBMS_WM.DisableVersioning('institution_name');

drop table personal_education;
drop table degree_level;
drop table qualification;
drop table major;
drop table institution_country;
drop table institution_name;
drop table country_name;

commit;
```

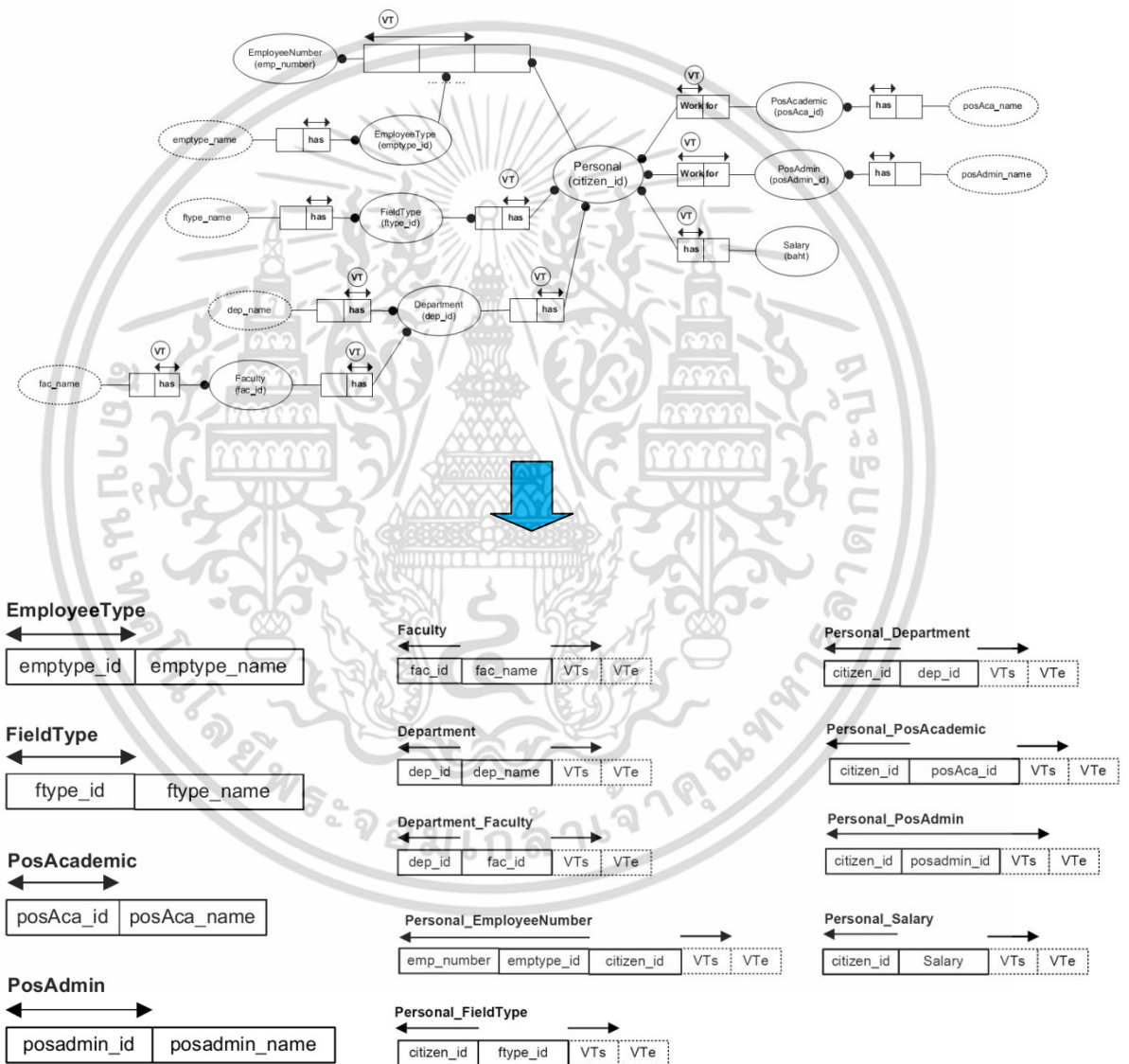


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง.

การทดลองบันทึกแก้ไขข้อมูลด้านประวัติการทำงาน

1. **TORM** ประวัติการทำงาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ทดลองสมมติข้อมูลในตาราง TORM

ตาราง 3.1 บันทึกชื่อประเภทตำแหน่ง (EmployeeType)

emptype_id	emptype_name
1	ข้าราชการ
2	พนักงานมหาวิทยาลัย
3	พนักงานราชการ
4	ลูกจ้างประจำ
5	ลูกจ้างชั่วคราว

ตาราง 3.2 บันทึกชื่อประเภทสายงาน (FieldType)

fieldtype_id	fieldtype_name
1	สายวิชาการ
2	สายสนับสนุน

ตาราง 3.3 บันทึกชื่อคณะ (Faculty)

fac_id	fac_name	VTs	VTe
1	คณะวิศวกรรมศาสตร์	24-04-1971	31-12-9999
2	คณะสถาปัตยกรรมศาสตร์	24-04-1971	31-12-9999
3	คณะครุศาสตร์อุตสาหกรรม	24-04-1971	31-12-9999
4	กองกลาง	24-04-1971	31-12-9999
5	กองการเจ้าหน้าที่	24-04-1971	31-12-9999

ตาราง 3.4 บันทึกชื่อแผนก (Department)

dep_id	dep_name	VTs	VTe
101	ภาควิชาวิศวกรรมไฟฟ้า	24-04-1971	31-12-9999
102	ภาควิชาวิศวกรรมอิเล็กทรอนิกส์	24-04-1971	31-12-9999
103	ภาควิชาวิศวกรรมคอมพิวเตอร์	24-04-1971	31-12-9999
104	ภาควิชาวิศวกรรมโทรคมนาคม	24-04-1971	31-12-9999
105	ภาควิชาวิศวกรรมเครื่องกล	24-04-1971	31-12-9999
701	แผนกคอมพิวเตอร์	24-04-1971	31-12-9999

ตาราง 3.5 บันทึกแผนก สังกัดคณะ (Department_Faculty)

dep_id	fac_id	VTs	VTe
101	1	24-04-1971	31-12-9999
102	1	24-04-1971	31-12-9999
103	1	24-04-1971	31-12-9999
104	1	24-04-1971	31-12-9999
105	1	24-04-1971	31-12-9999
701	5	24-04-1971	31-12-9999

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.6 บันทึกชื่อตำแหน่งทางวิชาการ (PosAcademic)

posAca_id	posAca_name
1	ศาสตราจารย์
2	รองศาสตราจารย์
3	ผู้ช่วยศาสตราจารย์
4	อาจารย์
5	เจ้าหน้าที่บริหารงานทั่วไป
6	นักวิชาการเงินและบัญชี
7	นักวิชาการพัสดุ
8	นักวิชาการศึกษา
9	นักวิชาการคอมพิวเตอร์
10	บุคลากร

ตาราง 3.7 บันทึกชื่อตำแหน่งทางบริหาร (PosAdmin)

posAdmin_id	posAdmin_name
1	อธิการบดี
2	รองอธิการบดี
3	ผู้ช่วยอธิการบดี
4	คณบดีคณะวิศวกรรมศาสตร์
5	คณบดีคณะสถาปัตยกรรมศาสตร์
6	คณบดีคณะครุศาสตร์อุตสาหกรรม
7	รองคณบดีคณะวิศวกรรมศาสตร์
8	รองคณบดีคณะสถาปัตยกรรมศาสตร์
9	รองคณบดีคณะครุศาสตร์อุตสาหกรรม
10	ผู้อำนวยการกองกลาง
11	ผู้อำนวยการกองการเจ้าหน้าที่
81	หัวหน้าภาควิชาวิศวกรรมไฟฟ้า
82	หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์

ตาราง 3.8 ข้อมูลตัวอย่างของ Personal_fldtype

citizen_id	ftype_id	VTs	VTe
33333	1	01-06-1977	
35003	1	01-06-1982	
66666	1	04-06-2001	
88888	2	01-06-2007	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.9 ข้อมูลตัวอย่างของ Personal_department

citizen_id	dep_id	VTs	VTe
33333	103	01-06-1977	
35003	102	01-06-1982	
66666	103	04-06-2001	
88888	701	01-06-2007	

ตาราง 3.10 ข้อมูลตัวอย่างของ Personal_PosAcademic

citizen_id	posAca_id	VTs	VTe
33333	4 (อาจารย์)	01-06-1977	01-10-1980
33333	3 (ผศ.)	01-10-1980	01-10-1987
33333	2 (รศ.)	01-10-1987	01-10-2000
33333	1 (ศ.)	01-10-2000	
35003	4	01-06-1982	01-10-1989
35003	3	01-10-1989	
50001	4	02-06-1977	15-06-1982
50001	3	15-06-1982	
60001	3	24-02-1981	05-03-1985
60001	4	05-03-1985	
66666	4	04-06-2001	
70001	3	01-01-1980	
88888	9 (นักวิชาการคอมฯ)	01-06-2007	

ตาราง 3.11 ข้อมูลตัวอย่างของ Personal_Admin

citizen_id	posAdmin_id	VTs	VTe
33333	82 (หัวหน้าภาคคอมฯ)	01-06-1982	
33333	7 (รองคณบดีคณะวิศวะ)	01-06-1986	01-06-1990
33333	4 (คณบดีคณะวิศวะ)	01-06-1990	01-06-1994
33333	2 (รองอธิการบดี)	01-06-2000	01-06-2004
33333	1 (อธิการบดี)	01-06-2012	
35003	81 (หัวหน้าภาคไฟฟ้า)	01-10-2000	01-10-2004

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.12 ข้อมูลตัวอย่างของ Personal_Salary

citizen_id	salary	VTs	VTe
33333	15000	01-06-1977	01-06-1980
33333	20000	01-06-1980	01-06-1985
33333	25000	01-06-1985	01-06-1990
33333	30000	01-06-1990	01-06-1995
33333	35000	01-06-1995	01-06-2000
33333	40000	01-06-2000	01-06-2005
33333	45000	01-06-2005	01-06-2010
33333	50000	01-06-2010	
35003	15000	01-06-1982	01-06-1985
35003	20000	01-06-1985	01-06-1990
35003	23000	01-06-1990	01-06-1995
35003	26000	01-06-1995	01-06-2000
35003	29000	01-06-2000	01-06-2005
35003	32000	01-06-2005	01-06-2010
35003	35000	01-06-2010	
66666	15000	04-06-2001	01-06-2005
66666	18000	01-06-2005	01-06-2010
66666	20000	01-06-2010	
88888	15000	01-06-2007	01-06-2010
88888	18000	01-06-2010	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทดลองคำสั่ง **Oracle 11g** ตามการออกแบบ **TORM** ประวัติการการทำงาน

4.1 สร้างตารางที่เกี่ยวข้องกับประวัติการทำงาน

```

CREATE TABLE employeetype (
  emptyype_id          NUMBER PRIMARY KEY,
  emptyype_name        VARCHAR(30));

CREATE TABLE fieldtype (
  ftype_id            NUMBER PRIMARY KEY,
  ftype_name          VARCHAR(30));

CREATE TABLE posAcademic (
  posaca_id           NUMBER PRIMARY KEY,
  posaca_name         VARCHAR(50));

CREATE TABLE posAdmin (
  posadmin_id         NUMBER PRIMARY KEY,
  posadmin_name       VARCHAR(50));

CREATE TABLE faculty (
  fac_id              NUMBER PRIMARY KEY,
  fac_name            VARCHAR(50));

CREATE TABLE department (
  dep_id              NUMBER PRIMARY KEY,
  dep_name            VARCHAR(50));

CREATE TABLE department_faculty (
  dep_id              NUMBER PRIMARY KEY,
  fac_id              NUMBER);

CREATE TABLE personal_employeenumner (
  emp_number          NUMBER NOT NULL,
  emptyype_id         NUMBER NOT NULL,
  citizen_id          VARCHAR(13),
  CONSTRAINT personal_employeetype_pk PRIMARY KEY(emp_number
, emptyype_id));

CREATE TABLE personal_fieldtype (
  citizen_id          VARCHAR(13) PRIMARY KEY,
  ftype_id            NUMBER);

CREATE TABLE personal_department (
  citizen_id          VARCHAR(13) PRIMARY KEY,
  dep_id              NUMBER);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CREATE TABLE personal_posacademic (
  citizen_id      VARCHAR(13) PRIMARY KEY,
  posaca_id      NUMBER);

CREATE TABLE personal_posadmin (
  citizen_id      VARCHAR(13),
  posadmin_id    NUMBER NOT NULL,
  CONSTRAINT    personal_posadmin_pk PRIMARY KEY(citizen_id,
posadmin_id));

CREATE TABLE personal_salary (
  citizen_id      VARCHAR(13) PRIMARY KEY,
  salary         NUMBER);

commit;

```

4.2 ใช้คำสั่ง EnableVersioning ทำให้ตารางที่บันทึกข้อมูลเชิงเวลา เป็นตาราง Temporal

```

EXECUTE DBMS_WM.EnableVersioning('faculty',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

EXECUTE DBMS_WM.EnableVersioning('department',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

EXECUTE DBMS_WM.EnableVersioning('department_faculty',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

EXECUTE DBMS_WM.EnableVersioning('personal_employeenumber',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

EXECUTE DBMS_WM.EnableVersioning('personal_fieldtype',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

EXECUTE DBMS_WM.EnableVersioning('personal_department',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

EXECUTE DBMS_WM.EnableVersioning('personal_posacademic',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

EXECUTE DBMS_WM.EnableVersioning('personal_posadmin',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

EXECUTE DBMS_WM.EnableVersioning('personal_salary',
'VIEW_WO_OVERWRITE', FALSE, TRUE);

commit;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ทำการเพิ่มข้อมูลตัวอย่าง

--1) เพิ่มชื่อสายการทำงาน

```
INSERT INTO fieldtype VALUES ( 1, 'สายวิชาการ' );
INSERT INTO fieldtype VALUES ( 2, 'สายวิชาสนับสนุน' );
```

--2) เพิ่มชื่อประเภทตำแหน่ง

```
INSERT INTO employeetype VALUES ( 1, 'ข้าราชการ' );
INSERT INTO employeetype VALUES ( 2, 'พนักงานมหาวิทยาลัย' );
INSERT INTO employeetype VALUES ( 3, 'พนักงานราชการ' );
INSERT INTO employeetype VALUES ( 4, 'ลูกจ้างประจำ' );
INSERT INTO employeetype VALUES ( 5, 'ลูกจ้างชั่วคราว' );
```

--3) เพิ่มชื่อทางตำแหน่งวิชาการ

```
INSERT INTO posAcademic VALUES ( 1, 'ศาสตราจารย์' );
INSERT INTO posAcademic VALUES ( 2, 'รองศาสตราจารย์' );
INSERT INTO posAcademic VALUES ( 3, 'ผู้ช่วยศาสตราจารย์' );
INSERT INTO posAcademic VALUES ( 4, 'อาจารย์' );
INSERT INTO posAcademic VALUES ( 5, 'เจ้าหน้าที่บริหารงานทั่วไป' );
INSERT INTO posAcademic VALUES ( 6, 'นักวิชาการเงินและบัญชี' );
INSERT INTO posAcademic VALUES ( 7, 'นักวิชาการพัสดุ' );
INSERT INTO posAcademic VALUES ( 8, 'นักวิชาการศึกษา' );
INSERT INTO posAcademic VALUES ( 9, 'นักวิชาการคอมพิวเตอร์' );
INSERT INTO posAcademic VALUES ( 10, 'บุคลากร' );
```

--4) เพิ่มชื่อตำแหน่งทางบริหาร

```
INSERT INTO posAdmin VALUES ( 1, 'อธิการบดี' );
INSERT INTO posAdmin VALUES ( 2, 'รองอธิการบดี' );
INSERT INTO posAdmin VALUES ( 3, 'ผู้ช่วยอธิการบดี' );
INSERT INTO posAdmin VALUES ( 4, 'คณบดีคณะวิศวกรรมศาสตร์' );
INSERT INTO posAdmin VALUES ( 5, 'คณบดีคณะสถาปัตยกรรมศาสตร์' );
INSERT INTO posAdmin VALUES ( 6, 'คณบดีคณะครุศาสตร์อุตสาหกรรม' );
INSERT INTO posAdmin VALUES ( 7, 'รองคณบดีคณะวิศวกรรมศาสตร์' );
INSERT INTO posAdmin VALUES ( 8, 'รองคณบดีคณะสถาปัตยกรรมศาสตร์' );
INSERT INTO posAdmin VALUES ( 9, 'รองคณบดีคณะครุศาสตร์อุตสาหกรรม' );
INSERT INTO posAdmin VALUES ( 10, 'ผู้อำนวยการกองกลาง' );
INSERT INTO posAdmin VALUES ( 11, 'ผู้อำนวยการกองการเจ้าหน้าที่' );
INSERT INTO posAdmin VALUES ( 81, 'หัวหน้าภาควิชาวิศวกรรมไฟฟ้า' );
INSERT INTO posAdmin VALUES ( 82, 'หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์' );
```

commit;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

--5) เพิ่มชื่อคณะและหน่วยงาน

```
INSERT INTO faculty VALUES(1, 'คณะวิศวกรรมศาสตร์', WMSYS.WM_PERIOD
(TO_DATE('24-04-1971', 'DD-MM-YYYY'), DBMS_WM.MAX_TIME));
```

```
INSERT INTO faculty VALUES(2, 'คณะสถาปัตยกรรมศาสตร์', WMSYS.WM_PERIOD
(TO_DATE('24-04-1971', 'DD-MM-YYYY'), DBMS_WM.MAX_TIME));
```

```
INSERT INTO faculty VALUES(3, 'คณะครุศาสตร์อุตสาหกรรม', WMSYS.WM_PERIOD
(TO_DATE('24-04-1971', 'DD-MM-YYYY'), DBMS_WM.MAX_TIME));
```

```
INSERT INTO faculty VALUES(4, 'กองกลาง', WMSYS.WM_PERIOD
(TO_DATE('24-04-1971', 'DD-MM-YYYY'), DBMS_WM.MAX_TIME));
```

```
INSERT INTO faculty VALUES(5, 'กองการเจ้าหน้าที่', WMSYS.WM_PERIOD
(TO_DATE('24-04-1971', 'DD-MM-YYYY'), DBMS_WM.MAX_TIME));
```

commit;

--6) เพิ่มชื่อแผนกต่างๆ

```
INSERT INTO department VALUES(101, 'ภาควิชาวิศวกรรมไฟฟ้า',
WMSYS.WM_PERIOD (TO_DATE('24-04-1971', 'DD-MM-YYYY'),
DBMS_WM.MAX_TIME));
```

```
INSERT INTO department VALUES(102, 'ภาควิชาวิศวกรรมอิเล็กทรอนิกส์',
WMSYS.WM_PERIOD (TO_DATE('24-04-1971', 'DD-MM-YYYY'),
DBMS_WM.MAX_TIME));
```

```
INSERT INTO department VALUES(103, 'ภาควิชาวิศวกรรมคอมพิวเตอร์',
WMSYS.WM_PERIOD (TO_DATE('24-04-1971', 'DD-MM-YYYY'),
DBMS_WM.MAX_TIME));
```

```
INSERT INTO department VALUES(104, 'ภาควิชาวิศวกรรมโทรคมนาคม',
WMSYS.WM_PERIOD (TO_DATE('24-04-1971', 'DD-MM-YYYY'),
DBMS_WM.MAX_TIME));
```

```
INSERT INTO department VALUES(105, 'ภาควิชาวิศวกรรมเครื่องกล',
WMSYS.WM_PERIOD (TO_DATE('24-04-1971', 'DD-MM-YYYY'),
DBMS_WM.MAX_TIME));
```

```
INSERT INTO department VALUES(701, 'แผนกคอมพิวเตอร์', WMSYS.WM_PERIOD
(TO_DATE('24-04-1971', 'DD-MM-YYYY'), DBMS_WM.MAX_TIME));
```

commit;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
--7) เพิ่มแผนกสังกัดคณะ department_faculty
INSERT INTO department_faculty VALUES(101,1,WMSYS.WM_PERIOD
(TO_DATE('24-04-1971','DD-MM-YYYY'),DBMS_WM.MAX_TIME));

INSERT INTO department_faculty VALUES(102,1,WMSYS.WM_PERIOD
(TO_DATE('24-04-1971','DD-MM-YYYY'),DBMS_WM.MAX_TIME));

INSERT INTO department_faculty VALUES(103,1,WMSYS.WM_PERIOD
(TO_DATE('24-04-1971','DD-MM-YYYY'),DBMS_WM.MAX_TIME));

INSERT INTO department_faculty VALUES(104,1,WMSYS.WM_PERIOD
(TO_DATE('24-04-1971','DD-MM-YYYY'),DBMS_WM.MAX_TIME));

INSERT INTO department_faculty VALUES(105,1,WMSYS.WM_PERIOD
(TO_DATE('24-04-1971','DD-MM-YYYY'),DBMS_WM.MAX_TIME));

INSERT INTO department_faculty VALUES(701,5,WMSYS.WM_PERIOD
(TO_DATE('24-04-1971','DD-MM-YYYY'),DBMS_WM.MAX_TIME));

commit;

--8.1) เพิ่มเลขอัตรา
INSERT INTO personal_employeenumber VALUES(01,1,'33333',
WMSYS.WM_PERIOD(TO_DATE('01-06-1977','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));
INSERT INTO personal_employeenumber VALUES(02,1,'35003',
WMSYS.WM_PERIOD(TO_DATE('01-06-1982','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));
INSERT INTO personal_employeenumber VALUES(03,1,'66666',
WMSYS.WM_PERIOD(TO_DATE('04-06-2001','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));
INSERT INTO personal_employeenumber VALUES(01,2,'88888',
WMSYS.WM_PERIOD(TO_DATE('01-06-2007','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));

--8.2) เพิ่มข้อมูลตารางบุคลากร กับสาขาการทำงาน personal_filetype
INSERT INTO personal_fieldtype VALUES('33333',1,WMSYS.WM_PERIOD
(TO_DATE('01-06-1977','DD-MM-YYYY'),DBMS_WM.UNTIL_CHANGED));
INSERT INTO personal_fieldtype VALUES('35003',1,WMSYS.WM_PERIOD
(TO_DATE('01-06-1982','DD-MM-YYYY'),DBMS_WM.UNTIL_CHANGED));
INSERT INTO personal_fieldtype VALUES('66666',1,WMSYS.WM_PERIOD
(TO_DATE('04-06-2001','DD-MM-YYYY'),DBMS_WM.UNTIL_CHANGED));
INSERT INTO personal_fieldtype VALUES('88888',2,WMSYS.WM_PERIOD
(TO_DATE('01-06-2007','DD-MM-YYYY'),DBMS_WM.UNTIL_CHANGED));

commit;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
--9) เพิ่มข้อมูลตารางบุคลากร สังกัดแผนก Personal_department
INSERT INTO Personal_department VALUES('33333',103,
WMSYS.WM_PERIOD (TO_DATE('01-06-1977','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));
INSERT INTO Personal_department VALUES('35003',102,
WMSYS.WM_PERIOD (TO_DATE('01-06-1982','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));
INSERT INTO Personal_department VALUES('66666',103,
WMSYS.WM_PERIOD (TO_DATE('04-06-2001','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));
INSERT INTO Personal_department VALUES('88888', 701,
WMSYS.WM_PERIOD (TO_DATE('01-06-2007','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));

commit;
```

4.4 เพิ่มข้อมูลประวัติและปรับปรุง ตำแหน่งทางวิชาการ ของบุคลากร

```
--1) เพิ่มข้อมูลในตาราง Personal_PosAcademic
INSERT INTO Personal_PosAcademic VALUES('33333',4,
WMSYS.WM_PERIOD (TO_DATE('01-06-1977','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));
INSERT INTO Personal_PosAcademic VALUES('35003',4,
WMSYS.WM_PERIOD (TO_DATE('01-06-1982','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));
INSERT INTO Personal_PosAcademic VALUES('66666',4,
WMSYS.WM_PERIOD (TO_DATE('04-06-2001','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));
INSERT INTO Personal_PosAcademic VALUES('88888',9,
WMSYS.WM_PERIOD (TO_DATE('01-06-2007','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));
commit;
```

```
--2) ปรับปรุงข้อมูลของ ศศ. รศ. ศ. ของ 33333 ใน Personal_PosAcademic ตามลำดับ
EXECUTE DBMS_WM.SetValidTime(TO_DATE('01-10-1980','DD-MM-
YYYY'), DBMS_WM.UNTIL_CHANGED);
UPDATE Personal_PosAcademic SET posAca_id =3 WHERE
citizen_id = '33333';

EXECUTE DBMS_WM.SetValidTime(TO_DATE('01-10-1987','DD-MM-
YYYY'), DBMS_WM.UNTIL_CHANGED);
UPDATE Personal_PosAcademic SET posAca_id =2 WHERE
citizen_id = '33333';

EXECUTE DBMS_WM.SetValidTime(TO_DATE('01-10-2000','DD-MM-
YYYY'), DBMS_WM.UNTIL_CHANGED);
UPDATE Personal_PosAcademic SET posAca_id =1 WHERE
citizen_id = '33333';

commit;
```

```
--3) ปรับปรุงข้อมูลของ ศศ. ของ 35003 ตามลำดับ
EXECUTE DBMS_WM.SetValidTime(TO_DATE('01-10-1989','DD-MM-
YYYY'), DBMS_WM.UNTIL_CHANGED);
UPDATE Personal_PosAcademic SET posAca_id =3 WHERE
citizen_id = '35003';

commit;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 แสดงประวัติตำแหน่งทางวิชาการ สามารถตอบคำถามเชิงเวลาต่างๆ ได้ดังนี้

4.5.1 แสดงประวัติตำแหน่งทางวิชาการของบุคลากรแต่ละท่าน ตั้งแต่อดีต จนถึง

ปัจจุบัน สามารถเขียนคำสั่งได้ ดังนี้

คำสั่งแบบที่ 1

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT personal_posacademic.citizen_id, posacademic.posaca_id,
posacademic.posaca_name,
       TO_CHAR(personal_posacademic.WM_VALID.VALIDFROM, 'DD-MM-
YYYY', 'NLS_CALENDAR=' 'THAI BUDDHA' ' NLS_DATE_LANGUAGE=THAI')
FROM_DATE,
       TO_CHAR(personal_posacademic.WM_VALID.VALIDTILL, 'DD-MM-
YYYY', 'NLS_CALENDAR=' 'THAI BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE

FROM personal_posacademic JOIN posacademic ON
personal_posacademic.posaca_id = posacademic.posaca_id
ORDER BY personal_posacademic.citizen_id ASC ,
personal_posacademic.posaca_id DESC ;
```

คำสั่งแบบที่ 2

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT e.citizen_id, e.posaca_id, pos.posaca_name,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE

FROM personal_posacademic e , posacademic pos
WHERE e.posaca_id = pos.posaca_id
ORDER BY e.citizen_id ASC , e.posaca_id DESC ;
```

คำสั่งแบบที่ 1 และ 2 ได้ผลลัพธ์ เหมือนกัน ดังนี้

```
anonymous block completed
1ms elapsed
```

CITIZEN_ID	POSACA_ID	POSACA_NAME	FROM_DATE	TO_DATE
33333	4	อาจารย์	01-06-2520	01-10-2523
33333	3	ผู้ช่วยศาสตราจารย์	01-10-2523	01-10-2530
33333	2	รองศาสตราจารย์	01-10-2530	01-10-2543
33333	1	ศาสตราจารย์	01-10-2543	
35003	4	อาจารย์	01-06-2525	01-10-2532
35003	3	ผู้ช่วยศาสตราจารย์	01-10-2532	
66666	4	อาจารย์	04-06-2544	
88888	9	นักวิชาการคอมพิวเตอร์	01-06-2550	

8 rows selected

3ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.2 แสดงประวัติตำแหน่งทางวิชาการ ณ เวลาปัจจุบัน สามารถเขียนคำสั่งได้ ดังนี้

คำสั่งแบบที่ 1

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT personal_posacademic.citizen_id, posacademic.posaca_id,
posacademic.posaca_name,
       TO_CHAR(personal_posacademic.WM_VALID.VALIDFROM, 'DD-MM-
YYYY', 'NLS_CALENDAR=' 'THAI BUDDHA' ' NLS_DATE_LANGUAGE=THAI')
FROM_DATE,
       TO_CHAR(personal_posacademic.WM_VALID.VALIDTILL, 'DD-MM-
YYYY', 'NLS_CALENDAR=' 'THAI BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE

FROM personal_posacademic JOIN posacademic ON
personal_posacademic.posaca_id = posacademic.posaca_id
ORDER BY personal_posacademic.citizen_id ASC ,
personal_posacademic.posaca_id DESC ;
```

คำสั่งแบบที่ 2

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT e.citizen_id, e.posaca_id, pos.posaca_name,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE

FROM personal_posacademic e , posacademic pos
WHERE e.posaca_id = pos.posaca_id
ORDER BY e.citizen_id ASC , e.posaca_id DESC ;
```

คำสั่งแบบที่ 1 และ 2 ได้ผลลัพธ์ เหมือนกัน ดังนี้

```
anonymous block completed
1ms elapsed
CITIZEN_ID      POSACA_ID      POSACA_NAME      FROM_DATE      TO_DATE
-----
33333          1              ศาสตราจารย์          01-10-2543
35003          3              ผู้ช่วยศาสตราจารย์  01-10-2532
66666          4              อาจารย์              04-06-2544
88888          9              นักวิชาการคอมพิวเตอร์ 01-06-2550
```

4 rows selected

4ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.3 มีบุคลากรคนไหนบ้าง ดำรงตำแหน่ง ผู้ช่วยศาสตราจารย์ ในช่วงเวลาเดียวกันกับ บุคลากรรหัส 33333 และในปัจจุบันเขายังคงดำรงตำแหน่ง ผู้ช่วยศาสตราจารย์ อยู่ ซึ่งบุคลากร รหัส 33333 เขาดำรงตำแหน่ง ศาสตราจารย์

--1) เพิ่มข้อมูลบุคลากร 50001 60001 70001 ดังนี้

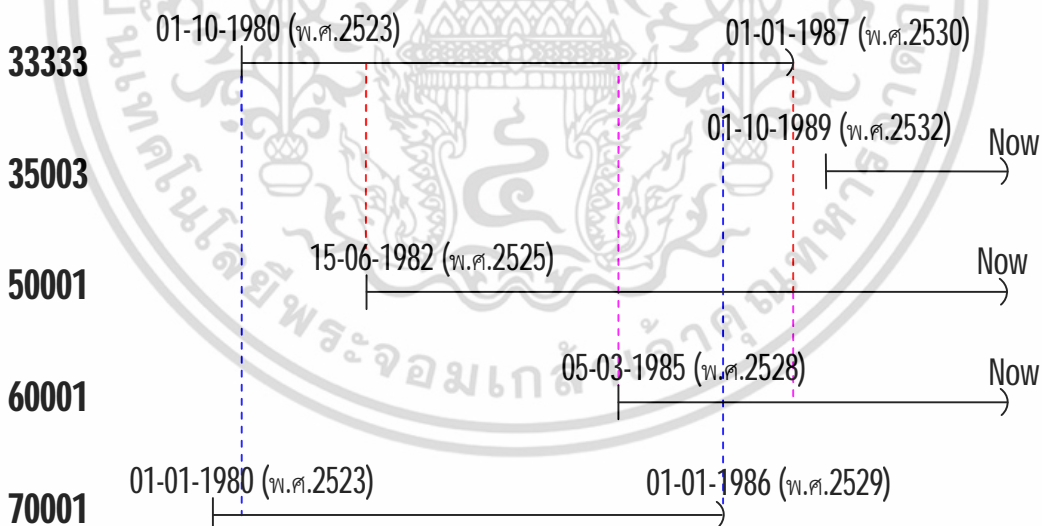
```
INSERT INTO Personal_PosAcademic VALUES ('50001',4,
WMSYS.WM_PERIOD (TO_DATE('02-06-1977','DD-MM-YYYY'), TO_DATE('15-06-1982','DD-MM-YYYY')));
```

```
INSERT INTO Personal_PosAcademic VALUES ('50001',3,
WMSYS.WM_PERIOD (TO_DATE('15-06-1982','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));
```

```
INSERT INTO Personal_PosAcademic VALUES ('60001',4,
WMSYS.WM_PERIOD (TO_DATE('24-02-1980','DD-MM-YYYY'), TO_DATE('05-03-1985','DD-MM-YYYY')));
```

```
INSERT INTO Personal_PosAcademic VALUES ('60001',3,
WMSYS.WM_PERIOD (TO_DATE('05-03-1985','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));
```

```
INSERT INTO Personal_PosAcademic VALUES ('70001',3,
WMSYS.WM_PERIOD (TO_DATE('01-01-1980','DD-MM-YYYY'), TO_DATE('01-01-1986','DD-MM-YYYY')));
```



รูปแสดงช่วงเวลาที่พนักงาน ดำรงตำแหน่ง ผศ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอบคำถามเชิงเวลา ข้อ 4.5.3

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT e1.citizen_id, e2.citizen_id, pos.posaca_name ,
       TO_CHAR(e2.WM_VALID.VALIDFROM, 'DD-MM-YYYY' , 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') FROM_DATE,
       TO_CHAR(e2.WM_VALID.VALIDTILL, 'DD-MM-YYYY' , 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE
FROM Personal_PosAcademic e1 , Personal_PosAcademic e2 , posacademic
pos
WHERE e1.posaca_id = pos.posaca_id
AND e1.citizen_id < e2.citizen_id
AND e1.posaca_id = e2.posaca_id
AND WM_OVERLAPS(e1.WM_VALID,e2.WM_VALID)=1
AND e1.posaca_id=3
AND e1.citizen_id='33333'
AND e2.WM_VALID.VALIDTILL IS NULL
```

anonymous block completed

CITIZEN_ID	CITIZEN_ID	POSACA_NAME	FROM_DATE	TO_DATE
33333	60001	ผู้ช่วยศาสตราจารย์	05-03-2528	
33333	50001	ผู้ช่วยศาสตราจารย์	15-06-2525	

2 rows selected

4.5.4 มีบุคลากรคนไหนบ้าง ดำรงตำแหน่ง ผู้ช่วยศาสตราจารย์ ในช่วงเวลาเดียวกันกับ

บุคลากรรหัส 33333

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT e1.citizen_id, e2.citizen_id, pos.posaca_name ,
       TO_CHAR(e2.WM_VALID.VALIDFROM, 'DD-MM-YYYY' , 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') FROM_DATE,
       TO_CHAR(e2.WM_VALID.VALIDTILL, 'DD-MM-YYYY' , 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE
FROM Personal_PosAcademic e1 , Personal_PosAcademic e2 , posacademic
pos
WHERE e1.posaca_id = pos.posaca_id
AND e1.citizen_id < e2.citizen_id
AND e1.posaca_id = e2.posaca_id
AND WM_OVERLAPS(e1.WM_VALID,e2.WM_VALID)=1
AND e1.posaca_id=3
AND e1.citizen_id='33333'
```

anonymous block completed

CITIZEN_ID	CITIZEN_ID	POSACA_NAME	FROM_DATE	TO_DATE
33333	70001	ผู้ช่วยศาสตราจารย์	01-01-2523	01-01-2529
33333	60001	ผู้ช่วยศาสตราจารย์	05-03-2528	
33333	50001	ผู้ช่วยศาสตราจารย์	15-06-2525	

3 rows selected

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.5 มีบุคลากรคนไหนบ้าง ที่เคยดำรงตำแหน่งทางวิชาการเท่ากัน

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME,DBMS_WM.MAX_TIME);

SELECT e1.citizen_id, e2.citizen_id, pos.posaca_name ,
       TO_CHAR(e2.WM_VALID.VALIDFROM, 'DD-MM-YYYY' , 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') FROM_DATE,
       TO_CHAR(e2.WM_VALID.VALIDTILL, 'DD-MM-YYYY' , 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE
FROM Personal_PosAcademic e1 , Personal_PosAcademic e2 , posacademic
pos
WHERE e1.posaca_id = pos.posaca_id
AND e1.citizen_id < e2.citizen_id
AND e1.posaca_id = e2.posaca_id
AND WM_OVERLAPS(e1.WM_VALID,e2.WM_VALID)=1
ORDER BY e1.citizen_id , e1.posaca_id DESC, e2.citizen_id ,
e2.WM_VALID.VALIDFROM
```

anonymous block completed

CITIZEN_ID	CITIZEN_ID	POSACA_NAME	FROM_DATE	TO_DATE
33333	50001	อาจารย์	02-06-2520	15-06-2525
33333	60001	อาจารย์	24-02-2523	05-03-2528
33333	50001	ผู้ช่วยศาสตราจารย์	15-06-2525	
33333	60001	ผู้ช่วยศาสตราจารย์	05-03-2528	
33333	70001	ผู้ช่วยศาสตราจารย์	01-01-2523	01-01-2529
35003	50001	อาจารย์	02-06-2520	15-06-2525
35003	60001	อาจารย์	24-02-2523	05-03-2528
35003	50001	ผู้ช่วยศาสตราจารย์	15-06-2525	
35003	60001	ผู้ช่วยศาสตราจารย์	05-03-2528	
50001	60001	อาจารย์	24-02-2523	05-03-2528
50001	60001	ผู้ช่วยศาสตราจารย์	05-03-2528	
50001	70001	ผู้ช่วยศาสตราจารย์	01-01-2523	01-01-2529
60001	70001	ผู้ช่วยศาสตราจารย์	01-01-2523	01-01-2529

13 rows selected

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.6 คำถามเชิงเวลา แสดงสถิติจำนวนผู้ดำรงตำแหน่งทางวิชาการ ณ เวลาปัจจุบัน สามารถเขียนคำสั่งได้ ดังนี้

คำสั่งแบบที่ 1

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT posacademic.posaca_id, posacademic.posaca_name,
Count(personal_posacademic.citizen_id) AS AMOUNT
FROM personal_posacademic JOIN posacademic ON
personal_posacademic.posaca_id = posacademic.posaca_id
GROUP BY posacademic.posaca_id, posacademic.posaca_name
ORDER BY posacademic.posaca_id;
```

คำสั่งแบบที่ 2

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT p.posaca_id, p.posaca_name, Count(e.citizen_id) AS AMOUNT
FROM personal_posacademic e, posacademic p
WHERE e.posaca_id = p.posaca_id
GROUP BY p.posaca_id, p.posaca_name
ORDER BY p.posaca_id;
```

คำสั่งแบบที่ 1 และ 2 ได้ผลลัพธ์ เหมือนกัน ดังนี้

```
anonymous block completed
2ms elapsed
POSACA_ID      POSACA_NAME      AMOUNT
-----
1              ศาสตราจารย์      1
3              ผู้ช่วยศาสตราจารย์  3
4              อาจารย์          1
9              นักวิชาการคอมพิวเตอร์  1
```

4 rows selected

3ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.7 คำถามเชิงเวลา แสดงสถิติจำนวนผู้ดำรงตำแหน่งทางวิชาการ ในช่วงวันที่ 1 ม.ค. 2000 ถึง 1 มิ.ย. 2000 สามารถเขียนคำสั่งได้ ดังนี้

คำสั่งแบบที่ 1

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('01-01-2000','DD-MM-YYYY'),
TO_DATE('01-06-2000','DD-MM-YYYY') );
```

```
SELECT posacademic.posaca_id, posacademic.posaca_name,
Count(personal_posacademic.citizen_id) AS AMOUNT
FROM personal_posacademic JOIN posacademic ON
personal_posacademic.posaca_id = posacademic.posaca_id
GROUP BY posacademic.posaca_id, posacademic.posaca_name
ORDER BY posacademic.posaca_id;
```

คำสั่งแบบที่ 2

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('01-01-2000','DD-MM-YYYY'),
TO_DATE('01-06-2000','DD-MM-YYYY') );
```

```
SELECT p.posaca_id, p.posaca_name, Count(e.citizen_id) AS AMOUNT
FROM personal_posacademic e, posacademic p
WHERE e.posaca_id = p.posaca_id
GROUP BY p.posaca_id, p.posaca_name
ORDER BY p.posaca_id;
```

คำสั่งแบบที่ 1 และ 2 ได้ผลลัพธ์ เหมือนกัน ดังนี้

POSACA_ID	POSACA_NAME	AMOUNT
2	รองศาสตราจารย์	1
3	ผู้ช่วยศาสตราจารย์	3

2 rows selected

3ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 เพิ่มข้อมูลประวัติและปรับปรุง ตำแหน่งทางบริหาร

```
--1) เพิ่มข้อมูลในตาราง Personal_PosAdmin
      INSERT INTO Personal_PosAdmin VALUES('33333',82,
WMSYS.WM_PERIOD (TO_DATE('01-06-1982','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));

      INSERT INTO Personal_PosAdmin VALUES('35003',81,
WMSYS.WM_PERIOD (TO_DATE('01-06-1982','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));

commit;

--2) แต่งตั้งพนักงาน 33333 ดำรงตำแหน่ง รองคณบดี
      INSERT INTO Personal_PosAdmin VALUES('33333',7,WMSYS.WM_PERIOD
(TO_DATE('01-06-1986','DD-MM-YYYY'),DBMS_WM.UNTIL_CHANGED));

--3) ปรับปรุงพนักงาน 33333 สิ้นสุดการดำรงตำแหน่ง รองคณบดี
      EXECUTE DBMS_WM.SetValidTime(TO_DATE('01-06-1990','DD-MM-
YYYY'),DBMS_WM.UNTIL_CHANGED);

      DELETE FROM Personal_PosAdmin WHERE citizen_id='33333' and
posadmin_id=7;

commit;

--4) แต่งตั้งพนักงาน 33333 ดำรงตำแหน่ง คณบดี
      INSERT INTO Personal_PosAdmin VALUES('33333',4,WMSYS.WM_PERIOD
(TO_DATE('01-06-1990','DD-MM-YYYY'),DBMS_WM.UNTIL_CHANGED));

commit;

--5) พนักงาน 33333 สิ้นสุดตำแหน่ง 4 (คณบดีคณะวิศวกรรมศาสตร์)
      EXECUTE DBMS_WM.SetValidTime(TO_DATE('01-06-1994','DD-MM-
YYYY'),DBMS_WM.UNTIL_CHANGED);

      DELETE FROM Personal_PosAdmin WHERE citizen_id='33333' and
posadmin_id=4;

commit;

--6) เพิ่มพนักงาน 33333 รับตำแหน่ง 2 รองอธิการบดี
      INSERT INTO Personal_PosAdmin VALUES('33333',2,
WMSYS.WM_PERIOD (TO_DATE('01-06-2000','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));

commit;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

--7) พนักงาน 33333 สิ้นสุดตำแหน่ง 2 รองอธิการบดี

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('01-06-2004','DD-MM-YYYY'), DBMS_WM.UNTIL_CHANGED);
```

```
DELETE FROM Personal_PosAdmin WHERE citizen_id='33333' and posadmin_id=2;
```

```
commit;
```

--8) พนักงาน 33333 รับตำแหน่ง 1 อธิการบดี

```
INSERT INTO Personal_PosAdmin VALUES('33333',1,
WMSYS.WM_PERIOD (TO_DATE('01-06-2012','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));
```

```
commit;
```

หมายเหตุ

การสิ้นสุดตำแหน่งทางบริหาร นอกจากจะใช้คำสั่ง **SetValidTime** และตามด้วยคำสั่ง **Delete** แล้ว ยังสามารถใช้คำสั่ง **Update** ที่แอตทริบิวของวันสิ้นสุดได้ โดยไม่ต้องผ่านคำสั่ง **SetValidTime** ดังตัวอย่างคำสั่งใน 5) สามารถเขียนอีกคำสั่งหนึ่งได้ดังนี้

--5) พนักงาน 33333 สิ้นสุดตำแหน่ง 4 (คณบดีคณะวิศวกรรมศาสตร์)

```
EXECUTE DBMS_WM.SetValidTime();
UPDATE Personal_PosAdmin e SET e.WM_VALID.VALIDTILL=
TO_DATE('01-06-1994','DD-MM-YYYY') WHERE citizen_id = '33333' AND
posadmin_id=4;
```

```
commit;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 ตอบคำถามเชิงเวลา เกี่ยวกับตำแหน่งบริหาร

4.7.1 ตอบคำถามเชิงเวลา เพื่อแสดงตำแหน่งบริหารของบุคลากรทุกคน ตั้งแต่อดีตจนถึงปัจจุบัน สามารถเขียนคำสั่งได้ ดังนี้

คำสั่งแบบที่ 1

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT personal_posadmin.citizen_id, personal_posadmin.posadmin_id,
       posadmin.posadmin_name,
       TO_CHAR(personal_posadmin.WM_VALID.VALIDFROM, 'DD-MM-
YYYY', 'NLS_CALENDAR=' 'THAI BUDDHA' ' NLS_DATE_LANGUAGE=THAI ')
FROM_DATE,
       TO_CHAR(personal_posadmin.WM_VALID.VALIDTILL, 'DD-MM-
YYYY', 'NLS_CALENDAR=' 'THAI BUDDHA' ' NLS_DATE_LANGUAGE=THAI ') TO_DATE
FROM personal_posadmin JOIN posadmin ON personal_posadmin.posadmin_id
= posadmin.posadmin_id
ORDER BY personal_posadmin.citizen_id,
personal_posadmin.WM_VALID.VALIDFROM;
```

คำสั่งแบบที่ 2

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT e.citizen_id, e.posadmin_id, p.posadmin_name,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI ') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI ') TO_DATE
FROM personal_posadmin e, posadmin p
WHERE e.posadmin_id = p.posadmin_id
ORDER BY e.citizen_id, e.WM_VALID.VALIDFROM;
```

คำสั่งแบบที่ 1 และ 2 ได้ผลลัพธ์ เหมือนกัน ดังนี้

anonymous block completed

2ms elapsed

CITIZEN_ID	POSADMIN_ID	POSADMIN_NAME	FROM_DATE	TO_DATE
33333	82	หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์	01-06-2525	
33333	7	รองคณบดีคณะวิศวกรรมศาสตร์	01-06-2529	01-06-2533
33333	4	คณบดีคณะวิศวกรรมศาสตร์	01-06-2533	01-06-2537
33333	2	รองอธิการบดี	01-06-2543	01-06-2547
33333	1	อธิการบดี	01-06-2555	
35003	81	หัวหน้าภาควิชาวิศวกรรมไฟฟ้า	01-06-2525	

6 rows selected

90ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7.2 ตอบคำถามเชิงเวลา เพื่อแสดงตำแหน่งบริหารของบุคลากรทุกคน ณ เวลาปัจจุบัน สามารถเขียนคำสั่งได้ ดังนี้

คำสั่งแบบที่ 1

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT personal_posadmin.citizen_id, personal_posadmin.posadmin_id,
posadmin.posadmin_name,
       TO_CHAR(personal_posadmin.WM_VALID.VALIDFROM, 'DD-MM-
YYYY', 'NLS_CALENDAR=' 'THAI BUDDHA' ' NLS_DATE_LANGUAGE=THAI ')
FROM_DATE,
       TO_CHAR(personal_posadmin.WM_VALID.VALIDTILL, 'DD-MM-
YYYY', 'NLS_CALENDAR=' 'THAI BUDDHA' ' NLS_DATE_LANGUAGE=THAI ') TO_DATE

FROM personal_posadmin JOIN posadmin ON personal_posadmin.posadmin_id
= posadmin.posadmin_id
ORDER BY personal_posadmin.citizen_id,
personal_posadmin.WM_VALID.VALIDFROM;
```

คำสั่งแบบที่ 2

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT e.citizen_id, e.posadmin_id, p.posadmin_name,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI ') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI ') TO_DATE
FROM personal_posadmin e , posadmin p
WHERE e.posadmin_id = p.posadmin_id
ORDER BY e.citizen_id, e.WM_VALID.VALIDFROM;
```

คำสั่งแบบที่ 1 และ 2 ได้ผลลัพธ์ เหมือนกัน ดังนี้

anonymous block completed
2ms elapsed

CITIZEN_ID	POSADMIN_ID	POSADMIN_NAME	FROM_DATE	TO_DATE
33333	82	หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์	01-06-2525	
33333	1	อธิการบดี	01-06-2555	
35003	81	หัวหน้าภาควิชาวิศวกรรมไฟฟ้า	01-06-2525	

3 rows selected

4ms elapsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 เพิ่มข้อมูลประวัติเงินเดือน

สมมุติการเพิ่มเงินเดือน มีการปรับเงินเดือนทุกๆ 5 ปี ดังตารางที่ 3.12

```
--1) เพิ่มข้อมูลในตาราง Personal_Salary
      INSERT INTO Personal_Salary VALUES('33333',15000,
WMSYS.WM_PERIOD (TO_DATE('01-06-1977','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));

commit;

--2) ปรับฐานเงินเดือน ณ ปี วันที่ 1 มิถุนายน 1980
      EXECUTE DBMS_WM.SetValidTime(TO_DATE('01-06-1980','DD-MM-
YYYY'), DBMS_WM.UNTIL_CHANGED);
      UPDATE Personal_Salary SET salary= 20000 WHERE citizen_id
='33333';

commit;

--3) บรรจุแต่งตั้งบุคลากร 35003 อัตราเงินเดือน 15000
      INSERT INTO Personal_Salary VALUES('35003', 15000,
WMSYS.WM_PERIOD (TO_DATE('01-06-1982','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));

--4) ปรับฐานเงินเดือน ณ ปี วันที่ 1 มิถุนายน 1985
      EXECUTE DBMS_WM.SetValidTime(TO_DATE('01-06-1985','DD-MM-
YYYY'), DBMS_WM.UNTIL_CHANGED);
      UPDATE Personal_Salary SET salary= 25000 WHERE citizen_id
='33333';

      UPDATE Personal_Salary SET salary= 20000 WHERE citizen_id
='35003';

commit;

--5) ปรับฐานเงินเดือน ณ ปี วันที่ 1 มิถุนายน 1990
      EXECUTE DBMS_WM.SetValidTime(TO_DATE('01-06-1990','DD-MM-
YYYY'), DBMS_WM.UNTIL_CHANGED);
      UPDATE Personal_Salary SET salary= 30000 WHERE citizen_id
='33333';

      UPDATE Personal_Salary SET salary= 23000 WHERE citizen_id
='35003';

commit;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

--6) ปรับฐานเงินเดือน ณ วันที่ 1 มิถุนายน 1995
      EXECUTE DBMS_WM.SetValidTime(TO_DATE('01-06-1995','DD-MM-
YYYY'), DBMS_WM.UNTIL_CHANGED);
      UPDATE Personal_Salary SET salary= 35000 WHERE citizen_id
='33333';

      UPDATE Personal_Salary SET salary= 26000 WHERE citizen_id
='35003';

commit;

--7) ปรับฐานเงินเดือน ณ วันที่ 1 มิถุนายน 2000
      EXECUTE DBMS_WM.SetValidTime(TO_DATE('01-06-2000','DD-MM-
YYYY'), DBMS_WM.UNTIL_CHANGED);
      UPDATE Personal_Salary SET salary= 40000 WHERE citizen_id
='33333';

      UPDATE Personal_Salary SET salary= 29000 WHERE citizen_id
='35003';

commit;

--8) บรรจุแต่งตั้งบุคลากร 66666 อัตราเงินเดือน 15000
      INSERT INTO Personal_Salary VALUES('66666', 15000,
WMSYS.WM_PERIOD (TO_DATE('04-06-2001','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));
commit;

--9) ปรับฐานเงินเดือน ณ วันที่ 1 มิถุนายน 2005
      EXECUTE DBMS_WM.SetValidTime(TO_DATE('01-06-2005','DD-MM-
YYYY'), DBMS_WM.UNTIL_CHANGED);
      UPDATE Personal_Salary SET salary= 45000 WHERE citizen_id
='33333';

      UPDATE Personal_Salary SET salary= 32000 WHERE citizen_id
='35003';

      UPDATE Personal_Salary SET salary= 18000 WHERE citizen_id
='66666';

commit;

--10) บรรจุแต่งตั้งบุคลากร 88888 อัตราเงินเดือน 15000

      INSERT INTO Personal_Salary VALUES('88888', 15000,
WMSYS.WM_PERIOD (TO_DATE('01-06-2007','DD-MM-YYYY'),
DBMS_WM.UNTIL_CHANGED));
commit;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
--11) ปรับฐานเงินเดือน ณ วันที่ 1 มิถุนายน 2010
      EXECUTE DBMS_WM.SetValidTime(TO_DATE('01-06-2010','DD-MM-
YYYY'), DBMS_WM.UNTIL_CHANGED);
      UPDATE Personal_Salary SET salary= 50000 WHERE citizen_id
='33333';

      UPDATE Personal_Salary SET salary= 35000 WHERE citizen_id
='35003';

      UPDATE Personal_Salary SET salary= 20000 WHERE citizen_id
='66666';

      UPDATE Personal_Salary SET salary= 18000 WHERE citizen_id
='88888';

commit;
```

4.9 คำถามเชิงเวลา เกี่ยวกับประวัติเงินเดือน

4.9.1 แสดงประวัติเงินเดือนของบุคลากรทุกคน ตั้งแต่อดีต จนถึงปัจจุบัน

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT e.CITIZEN_ID, e.salary,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE

FROM   personal_salary e
ORDER BY e.citizen_id ASC, e.WM_VALID.VALIDFROM ;
```

anonymous block completed
2ms elapsed

CITIZEN_ID	SALARY	FROM_DATE	TO_DATE
33333	15000	01-06-2520	01-06-2523
33333	20000	01-06-2523	01-06-2528
33333	25000	01-06-2528	01-06-2533
33333	30000	01-06-2533	01-06-2538
33333	35000	01-06-2538	01-06-2543
33333	40000	01-06-2543	01-06-2548
33333	45000	01-06-2548	01-06-2553
33333	50000	01-06-2553	
35003	15000	01-06-2525	01-06-2528
35003	20000	01-06-2528	01-06-2533
35003	23000	01-06-2533	01-06-2538
35003	26000	01-06-2538	01-06-2543
35003	29000	01-06-2543	01-06-2548
35003	32000	01-06-2548	01-06-2553
35003	35000	01-06-2553	
66666	15000	04-06-2544	01-06-2548
66666	18000	01-06-2548	01-06-2553
66666	20000	01-06-2553	
88888	15000	01-06-2550	01-06-2553
88888	18000	01-06-2553	

20 rows selected

เอกสารนี้ 15ms elapsed ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.9.2 แสดงประวัติเงินเดือนของบุคลากรทุกคน ณ เวลาปัจจุบัน

```
EXECUTE DBMS_WM.SetValidTime( );

SELECT e.CITIZEN_ID, e.salary,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE
FROM   personal_salary e
ORDER BY e.citizen_id ASC, e.WM_VALID.VALIDFROM ;
```

anonymous block completed

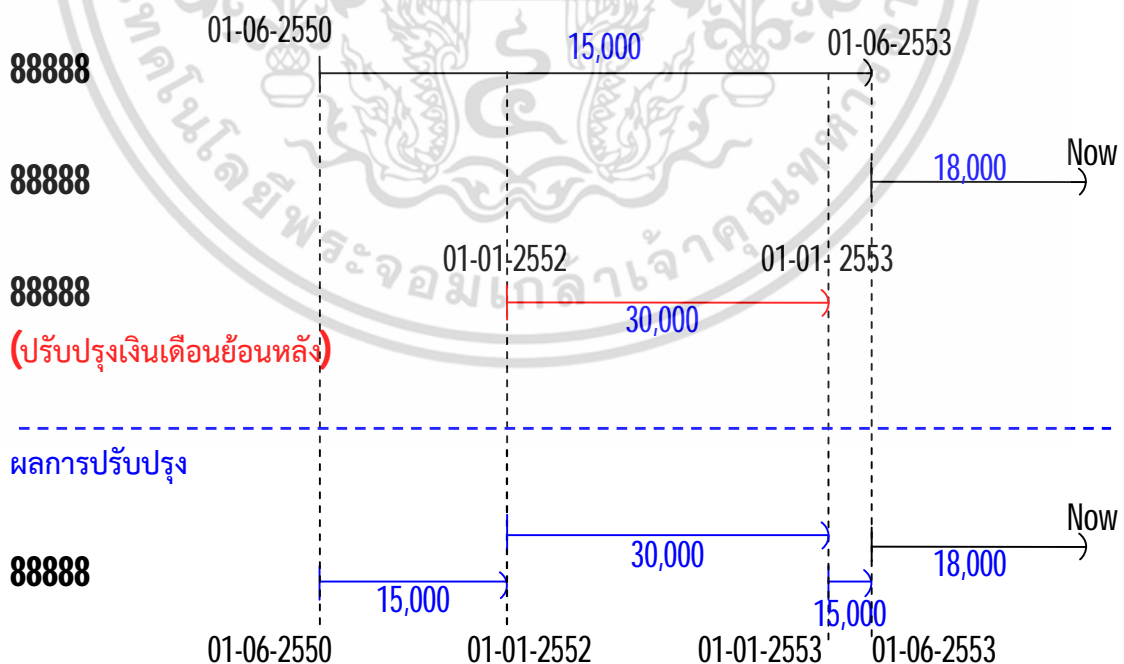
2ms elapsed

CITIZEN_ID	SALARY	FROM_DATE	TO_DATE
33333	50000	01-06-2553	
35003	35000	01-06-2553	
66666	20000	01-06-2553	
88888	18000	01-06-2553	

4 rows selected

5ms elapsed

4.9.3 แสดงการปรับเงินเดือนย้อนหลัง บุคลากรรหัส 88888 ได้ไปช่วยราชการที่สามจังหวัดชายแดนภาคใต้ ในช่วงวันที่ 1 ม.ค. 2552 ถึง 31 ธ.ค. 2552 ซึ่งเขาได้รับเงินเดือน 15,000 บาท ต่อมาในหลายหลังได้รับอนุมัติให้ปรับเงินเดือนย้อนหลังเป็น 30,000 บาท จารูปต่อไปนี้แสดงช่วงเวลาการได้รับเงินเดือนและการปรับเงินเดือนย้อนหลังของบุคลากรรหัส 88888



รูปแสดงการปรับปรุงเงินเดือนย้อนหลังของบุคลากรรหัส 88888

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.9.3.1 แสดงข้อมูลเงินเดือนของบุคลากรรหัส 88888 ก่อนทำการปรับปรุงเงินเดือนย้อนหลัง

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT e.CITIZEN_ID, e.salary,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE
FROM   personal_salary e
WHERE  citizen_id = '88888'
ORDER BY e.citizen_id ASC, e.WM_VALID.VALIDFROM ;
```

anonymous block completed

CITIZEN_ID	SALARY	FROM_DATE	TO_DATE
88888	15000	01-06-2550	01-06-2553
88888	18000	01-06-2553	

2 rows selected

4.9.3.2 กำหนดช่วงระยะเวลาที่ทำงานปรับปรุงเงินเดือนย้อนหลัง และใช้คำสั่ง Update

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('01-01-2009', 'DD-MM-YYYY'),
TO_DATE('01-01-2010', 'DD-MM-YYYY'));

UPDATE Personal_Salary SET salary= 30000 WHERE citizen_id = '88888';
```

anonymous block completed

1 rows updated

4.9.3.3 แสดงผลของการปรับปรุงเงินเดือนย้อนหลัง จะเห็นว่า แถวข้อมูลเป็น 4 แถวข้อมูล

และมีการปรับปรุงเงินเดือนและช่วงเวลา

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);

SELECT e.CITIZEN_ID, e.salary,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE
FROM   personal_salary e
WHERE  citizen_id = '88888'
ORDER BY e.citizen_id ASC, e.WM_VALID.VALIDFROM ;
```

anonymous block completed

CITIZEN_ID	SALARY	FROM_DATE	TO_DATE
88888	15000	01-06-2550	01-01-2552
88888	30000	01-01-2552	01-01-2553
88888	15000	01-01-2553	01-06-2553
88888	18000	01-06-2553	

4 rows selected

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ทดลองตรวจสอบการเข้าช้อนข้อมูลหรือไม่ ว่าเป็นไปตามนิยาม **1TNF**

ตารางที่ 5.1 The SAL Relation

E#	SALARY	T_s	T_e
E1	20K	1	5
E1	25K	6	10
E1	27K	9	11

จากตารางที่ 5.1 จะไม่เป็นไปตามนิยามของ **1TNF** เพราะมีค่าซ้ำซ้อนของเงินเดือนที่เวลา 9

5.1 เพิ่มข้อมูลเงินเดือน 20K และ 25K

```
INSERT INTO Personal_Salary VALUES('E1',20000,
WMSYS.WM_PERIOD (TO_DATE('01-01-2000','DD-MM-YYYY'), TO_DATE('01-05-
2000','DD-MM-YYYY')));
```

```
INSERT INTO Personal_Salary VALUES('E1',25000,
WMSYS.WM_PERIOD (TO_DATE('01-06-2000','DD-MM-YYYY'), TO_DATE('01-10-
2000','DD-MM-YYYY')));
```

```
commit;
```

```
2 rows inserted
commit succeeded.
```

5.2 ตรวจสอบการเพิ่มข้อมูลเงินเดือน

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);
```

```
SELECT e.CITIZEN_ID, e.salary,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'DD-MM-YYYY', 'NLS_CALENDAR=' 'THAI
BUDDHA' ' NLS_DATE_LANGUAGE=THAI') TO_DATE
```

```
FROM personal_salary e
ORDER BY e.citizen_id ASC, e.WM_VALID.VALIDFROM ;
```

```
anonymous block completed
```

```
CITIZEN_ID      SALARY          FROM_DATE      TO_DATE
-----
E1              20000          01-01-2543    01-05-2543
E1              25000          01-06-2543    01-10-2543
```

```
2 rows selected
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ทดลองเพิ่มข้อมูลเงินเดือน **27K** ช่วงวันที่ **01-09-2000** ถึง **01-11-2000** จะเห็นว่าไม่สามารถเพิ่มข้อมูลได้ เพราะไม่เป็นไปตามนิยาม

```
INSERT INTO Personal_Salary VALUES('E1',27000,
WMSYS.WM_PERIOD (TO_DATE('01-09-2000','DD-MM-YYYY'), TO_DATE('01-11-2000','DD-MM-YYYY')));
```

Error starting at line 2 in command:

```
INSERT INTO Personal_Salary VALUES('E1',27000,
WMSYS.WM_PERIOD (TO_DATE('01-09-2000','DD-MM-YYYY'), TO_DATE('01-11-2000','DD-MM-YYYY')))
```

Error report:

```
SQL Error: ORA-20010: unique key violation
ORA-06512: at "TEMPORAL.OVM_INSERT_257", line 13
ORA-04088: error during execution of trigger
'TEMPORAL.OVM_INSERT_257'
```

5.4 ทดลองปรับปรุงข้อมูลเงินเดือน **27K** ช่วงวันที่ **01-09-2000** ถึง **01-11-2000** จะเห็นว่าสามารถแสดงการปรับปรุงข้อมูลเป็น **0** แถว

```
EXECUTE DBMS_WM.SetValidTime(TO_DATE('01-09-2010','DD-MM-YYYY'), TO_DATE('01-11-2010','DD-MM-YYYY'));
UPDATE Personal_Salary SET salary= 27000 WHERE citizen_id='E1';
```

anonymous block completed
0 rows updated

5.5 ตรวจสอบผลการปรับปรุงข้อมูล จะเห็นว่าไม่มีการปรับปรุงข้อมูล ซึ่งเป็นไปตามนิยาม

```
EXECUTE DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME);
SELECT e.CITIZEN_ID, e.salary,
       TO_CHAR(e.WM_VALID.VALIDFROM, 'DD-MM-YYYY', 'NLS_CALENDAR=' THAI BUDDHA ' NLS_DATE_LANGUAGE=THAI') FROM_DATE,
       TO_CHAR(e.WM_VALID.VALIDTILL, 'DD-MM-YYYY', 'NLS_CALENDAR=' THAI BUDDHA ' NLS_DATE_LANGUAGE=THAI') TO_DATE
```

```
FROM personal_salary e
ORDER BY e.citizen_id ASC, e.WM_VALID.VALIDFROM ;
```

anonymous block completed

CITIZEN_ID	SALARY	FROM_DATE	TO_DATE
E1	20000	01-01-2543	01-05-2543
E1	25000	01-06-2543	01-10-2543

2 rows selected

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. สรุปผลการทดลอง

จากการทดลอง สามารถบันทึก ปรับปรุง ข้อมูลประวัติการทำงาน ประเภทการทำงาน การสังกัดแผนก

7. การยกเลิกตารางแบบเชิงเวลาและลบตาราง เพื่อเริ่มต้นการทดลอง

การยกเลิกตารางแบบเชิงเวลา จะใช้คำสั่ง **DisableVersioning** และลบตาราง ที่สัมพันธ์ **Address** ทั้งหมด เพื่อเริ่มต้นการทดลอง

```
EXECUTE DBMS_WM.DisableVersioning('faculty');
EXECUTE DBMS_WM.DisableVersioning('department');
EXECUTE DBMS_WM.DisableVersioning('department_faculty');
EXECUTE DBMS_WM.DisableVersioning('personal_employeenumber');
EXECUTE DBMS_WM.DisableVersioning('personal_fieldtype');
EXECUTE DBMS_WM.DisableVersioning('personal_department');
EXECUTE DBMS_WM.DisableVersioning('personal_posacademic');
EXECUTE DBMS_WM.DisableVersioning('personal_posadmin');
EXECUTE DBMS_WM.DisableVersioning('personal_salary');

drop table fieldType;
drop table posadmin;
drop table employeetype;
drop table faculty;
drop table department;
drop table department_faculty;
drop table posacademic;
drop table personal_employeenumber;
drop table personal_fieldtype;
drop table personal_department;
drop table personal_posacademic;
drop table personal_posadmin;
drop table personal_salary;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ. ระบบสารสนเทศเชิงเวลาต้นแบบ

1. ขั้นตอนการพัฒนาาระบบสารสนเทศเชิงเวลาต้นแบบ

1.1 ติดตั้ง Oracle11g

วิธีการติดตั้งศึกษาได้จาก

<http://ruzzv.wordpress.com/2013/05/28/2-%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%95%E0%B8%B4%E0%B8%94%E0%B8%95%E0%B8%B1%E0%B9%89%E0%B8%87-oracle-database-11g-release-1/>

1.2 ติดตั้ง Tomcat7

วิธีการติดตั้งศึกษาได้จาก

http://www.atom.rmutphysics.com/charud/oldnews/0/286/17/9/pic1/prg_note/server/TomcatInstall.html

1.3 ติดตั้ง JDBC

1) ดาวน์โหลดไฟล์ `ojdbc6.jar`

ดาวน์โหลดได้จาก : <http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-111060-084321.html>

2) สำเนาไฟล์ `ojdbc6.jar` ที่ดาวน์โหลดเรียบร้อยแล้ว ไปไว้ในโฟลเดอร์ของจาวา ที่ `C:\Program Files (x86)\Java\jdk1.6.0_18\jre\lib\ext`

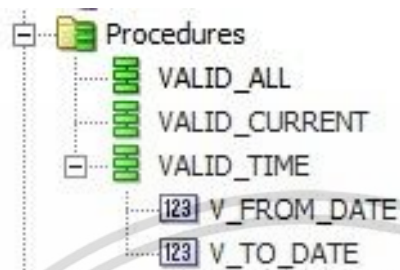
1.4 สร้างรหัสผู้ใช้ที่ Oracle11g ซึ่งในการพัฒนาาระบบสารสนเทศเชิงเวลาต้นแบบ ใช้ชื่อและรหัสคือ `temporal/temporal`

1.5 สร้างตารางตามที่ได้ออกแบบไว้

1.6 พัฒนาระบบด้วยภาษา JSP

2. ในการเขียนคำสั่ง JSP เพื่อทำการติดต่อกับฐานข้อมูล Oracle11g จะมีบางคำสั่งที่เรียกใช้ผ่าน Workspace Manager ซึ่งเป็นคำสั่งพิเศษ ไม่ใช่คำสั่งใน SQL มาตรฐาน จำเป็นต้องเขียน โพรซีเจอร์ (Procedures) ฝังอยู่ที่ Oracle11g จำนวน 3 โพรซีเจอร์ ดังนี้

2.1 Procedures in Oracel11g



1) valid_all procedures

```
create or replace procedure valid_all as
begin
  DBMS_WM.SetValidTime(DBMS_WM.MIN_TIME, DBMS_WM.MAX_TIME );
end valid_all;
```

2) Valid_current procedures

```
create or replace procedure valid_current as
begin
  DBMS_WM.SetValidTime( );
end valid_current;
```

3) Valid_time procedures

```
create or replace procedure valid_time(
  v_FROM_DATE in varchar,
  v_TO_DATE in varchar
) as
begin
  DBMS_WM.SetValidTime(TO_DATE(v_FROM_DATE, 'MM/DD/YYYY'), TO_DATE(v_TO_DATE, 'MM/DD/YYYY'));
end valid_time;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ตัวอย่างการเรียกโพรซีเจอร์ ใน JSP ทั้ง 3 โพรซีเจอร์

```
// Call VALID_CURRENT Procedures
CallableStatement sql = conn.prepareCall("{call VALID_CURRENT( )}");
sql.execute();

Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("SELECT      PRO_ID,
PRO_NAME,      TO_CHAR(e.WM_VALID.VALIDFROM, 'DD-MM-YYYY') FROM_DATE,
TO_CHAR(e.WM_VALID.VALIDTILL, 'DD-MM-YYYY')   TO_DATE      FROM
PROVINCE_NAME e ORDER BY PRO_ID, e.WM_VALID.VALIDFROM  ");
```

```
//Call VALID_ALL Procedures
CallableStatement sql = conn.prepareCall("{call VALID_ALL( )}");
sql.execute();

Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("SELECT PRO_ID,   PRO_NAME,
TO_CHAR(e.WM_VALID.VALIDFROM, 'DD-MM-YYYY') FROM_DATE,
TO_CHAR(e.WM_VALID.VALIDTILL, 'DD-MM-YYYY') TO_DATE FROM
PROVINCE_NAME e ORDER BY PRO_ID, e.WM_VALID.VALIDFROM  ");
```

```
//Call VALID_TIME Procedures
CallableStatement sql = conn.prepareCall("{call VALID_TIME(' " +
v_FROM_DATE + "',' " + v_TO_DATE + "')}");
sql.execute();

Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("SELECT PRO_ID,   PRO_NAME,
TO_CHAR(e.WM_VALID.VALIDFROM, 'DD-MM-YYYY') FROM_DATE,
TO_CHAR(e.WM_VALID.VALIDTILL, 'DD-MM-YYYY') TO_DATE FROM
PROVINCE_NAME e " + v_PRO_ID + " ORDER BY PRO_ID,
e.WM_VALID.VALIDFROM  ");
```

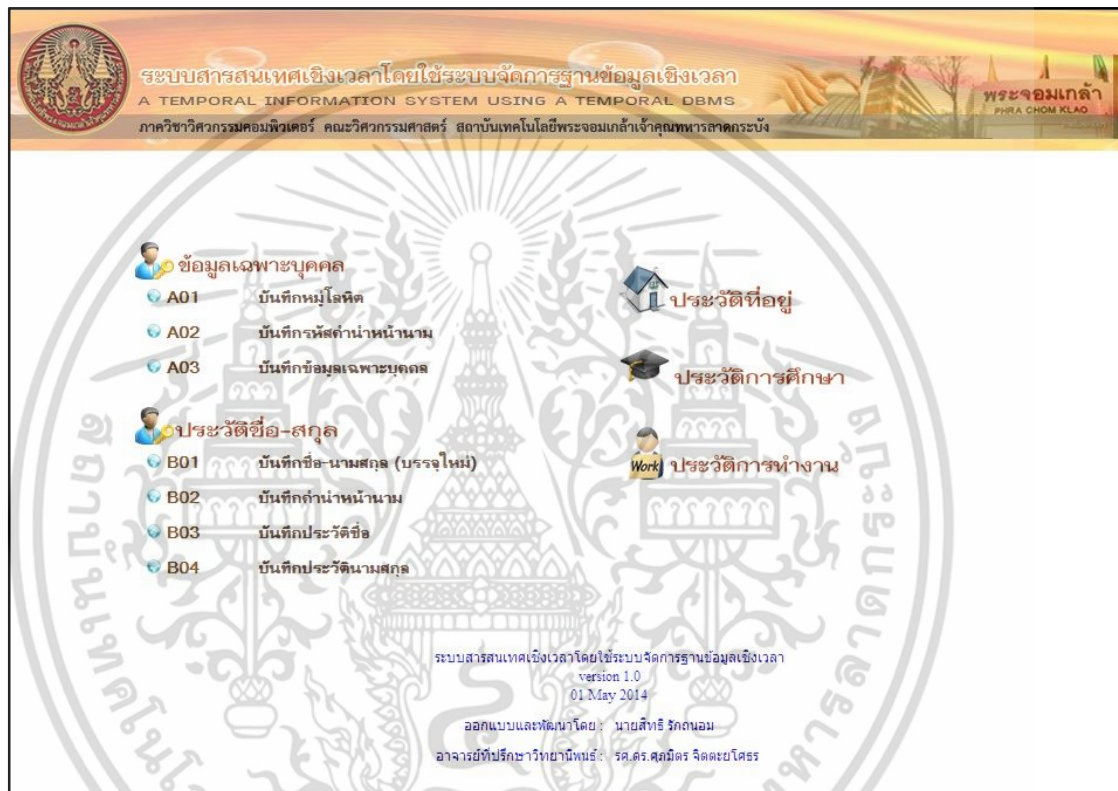
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงผลหน้าจอการทำงานของระบบ

1. การเข้าใช้ระบบ <http://161.246.6.131:80800/Temporal>

หมายเหตุ เป็น URL ที่ใช้ในการทำวิจัย

2. หน้าจอเมนูประกอบด้วย 5 ส่วน คือ ข้อมูลเฉพาะบุคคล ประวัติชื่อ-สกุล ประวัติที่อยู่ ประวัติการศึกษา และประวัติการทำงาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ความหมายของปุ่มคำสั่งต่างๆ



บันทึกข้อมูล (Save)



ปรับปรุงข้อมูล (Edit)



ลบข้อมูล (Delete)



ลบแถวข้อมูล (Delete row)



แสดงข้อมูลทุกช่วงเวลา ตั้งแต่อดีต-ปัจจุบัน (All time)



แสดงข้อมูล ณ เวลาปัจจุบัน (Current time)



แสดงข้อมูลตามช่วงเวลาที่กำหนด (Valid time)



ออกจากหน้าจอเมนู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ข้อมูลเฉพาะบุคคล

4.1 เมนูหลักของข้อมูลเฉพาะบุคคล



4.2 เมนู A01 บ้านทีกหมูโลหิต

บ้านทีกข้อมูลหมูโลหิต

รหัสหมูโลหิต*
Group blood ID

ชื่อหมูโลหิตภาษาไทย
Group blood in Thai

รหัสหมูโลหิต Group blood ID	ชื่อหมูโลหิต Group blood in Thai	แก้ไข Edit	ลบข้อมูล Delete
A	เอ		
B	บี		
AB	เอบี		
O	โอ		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 เมนู A02 บันทึกรหัสคำนำหน้านาม



บันทึกรหัสคำนำหน้านาม			
รหัสคำนำหน้า* Title ID	<input type="text"/>		
คำนำหน้าภาษาไทย Title in Thai	<input type="text"/>	คำนำหน้าภาษาไทย-อักษรย่อ Title in Thai-abbreviation	<input type="text"/>
คำนำหน้าภาษาอังกฤษ Title in English	<input type="text"/>	คำนำหน้าภาษาอังกฤษ-อักษรย่อ Title in English-abbreviation	<input type="text"/>

	รหัสคำนำหน้า Title ID	คำนำหน้าไทย Title in Thai	คำนำหน้าไทย-อักษรย่อ Title in Thai-abbr	คำนำหน้าอังกฤษ Title in English	คำนำหน้าอังกฤษ-อักษรย่อ Title in English-abbr	แก้ไข Edit	ลบข้อมูล Delete
1	1000	เด็กชาย	ค.ช.	Master	Mstr.		
2	1001	เด็กหญิง	ค.ญ.	Miss	Ms.		
3	1002	นาย	นาย	Mister	Mr.		
4	1003	นาง	นาง	Missis	Mrs.		
5	1004	นางสาว	น.ส.	Miss	Mis.		

4.4 เมนู A03 บันทึกข้อมูลเฉพาะบุคคล



บันทึกข้อมูลส่วนบุคคล			
เลขประจำตัวประชาชน* Identification Number	<input type="text"/>	ว.ต.ป. เกิด Birthday	<input type="text"/>
เพศ Gender	ชาย ▼	หมู่โลหิต Group blood	A ▼
ชื่อบิดา Father first name	<input type="text"/>	นามสกุลบิดา Father last name	<input type="text"/>
ชื่อมารดา Mother first name	<input type="text"/>	นามสกุลมารดา Mother last name	<input type="text"/>

	เลขประจำตัวประชาชน Identification Number	ว.ต.ป. เกิด Birthday	เพศ Gender	หมู่โลหิต Group blood	ชื่อบิดา Father first name	นามสกุลบิดา Father last name	ชื่อมารดา Mother first name	นามสกุลมารดา Mother last name	แก้ไข Edit	ลบข้อมูล Delete
1	33333	23-04-1991	M	A	ประทีป	อุตรธรรม	ปราณี	อุตรธรรม		
2	35003	23-04-1991	M	A	มงคล	ทตเจริญ	จรวย	ทตเจริญ		
3	66666	28-02-1976	M	O	บุญทิน	รักถนอม	อนงค์	ไพชู		
4	88888	03-05-1992	W	B	กิตติกร	บุญรักษา	ดวงใจ	บุญรักษา		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ประวัติชื่อ-นามสกุล

5.1 เมนูหลัก ประวัติชื่อ-สกุล



5.2 เมนู B01 บันทึกชื่อ-นามสกุล (บรรจุใหม่)

แสดงทุกช่วงเวลา แสดง ณ ปัจจุบัน แสดง ณ ช่วงเวลา บันทึกข้อมูล กลับไปที่เมนูหลัก

All Time Current Time Valid Time Insert data Close windows

บันทึกชื่อ-นามสกุล (บรรจุใหม่)			
เลขประจำตัวประชาชน* Identification Number	<input type="text"/>	ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
ตำแหน่งนำหน้านาม Title name	ค.ช. <input type="text"/>	ว.ด.ป. สิ้นสุด To date	<input type="text"/>
ชื่อ First name	<input type="text"/>		
นามสกุล Last name	<input type="text"/>		

	เลขประจำตัวประชาชน Identification Number	ตำแหน่งนำหน้า Title name	ชื่อ First name	นามสกุล Last name
1	88888	น.ส.	สมปรารถนา	บุญรักษา
2	33333	นาย	นเรศ	อุตรธรรม
3	35003	นาย	สมชาติ	ทศเจริญ
4	66666	นาย	สิทธิ	รักษนอม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 เมนู B02 บันทึกคำนำหน้านาม



บันทึกประวัติคำนำหน้า			
เลขประจำตัวประชาชน* Identification Number	<input type="text"/>	ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
คำนำหน้า Title	ด.ช. ▼	ว.ด.ป. สิ้นสุด To date	<input type="text"/>

	เลขประจำตัวประชาชน Identification Number	คำนำหน้า Title	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	33333	นาย	01-06-1977	31-12-9999
2	35003	นาย	01-06-1982	31-12-9999
3	66666	นาย	04-06-2001	31-12-9999
4	88888	น.ส.	26-05-2013	31-12-9999

5.4 เมนู B03 บันทึกประวัติชื่อ



บันทึกประวัติชื่อ			
เลขประจำตัวประชาชน* Identification Number	<input type="text"/>	ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
ชื่อ First name	<input type="text"/>	ว.ด.ป. สิ้นสุด To date	<input type="text"/>

	เลขประจำตัวประชาชน Identification Number	ชื่อ First name	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	33333	นเรศ	01-06-1977	31-12-9999
2	35003	สมชาติ	01-06-1982	31-12-9999
3	66666	สิทธิ์	04-06-2001	31-12-9999
4	88888	สมปรารถนา	26-05-2013	31-12-9999

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 B04 บันทึกประวัตินามสกุล



บันทึกประวัตินามสกุล			
เลขประจำตัวประชาชน* Identification Number	<input type="text"/>	ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
นามสกุล Last name	<input type="text"/>	ว.ด.ป. สิ้นสุด To date	<input type="text"/>

	เลขประจำตัวประชาชน Identification Number	ชื่อ Last name	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	33333	ฉัตรธรรม	01-06-1977	31-12-9999
2	35003	ทศเจริญ	01-06-1982	31-12-9999
3	66666	รักถนอม	04-06-2001	31-12-9999
4	88888	มณูรักษา	26-05-2013	31-12-9999

6. ประวัติที่อยู่

6.1 เมนูหลัก ประวัติที่อยู่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 เมนู C01 บันทึกชื่อจังหวัด



บันทึกชื่อจังหวัด			
รหัสจังหวัด* Province ID	<input type="text"/>	ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
ชื่อจังหวัด Province name	<input type="text"/>	ว.ด.ป. สิ้นสุด To date	<input type="text"/>

	รหัสจังหวัด Province ID	ชื่อจังหวัด Province name	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	39	หนองบัวลำภู	01-12-1993	null
2	41	อุดรธานี	21-06-1932	31-12-9999

6.3 เมนู C02 บันทึกชื่ออำเภอ



บันทึกชื่ออำเภอ			
รหัสอำเภอ* District ID	<input type="text"/>	ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
ชื่ออำเภอ District name	<input type="text"/>	ว.ด.ป. สิ้นสุด To date	<input type="text"/>

	รหัสอำเภอ District ID	ชื่ออำเภอ District name	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	4101	เมืองอุดรธานี	21-06-1932	31-12-9999
2	4102	หนองบัวลำภู	21-06-1932	01-12-1993
3	4102	เมืองหนองบัวลำภู	01-12-1993	31-12-9999

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 เมนู C03 บันทึกชื่อตำบล



บันทึกชื่อตำบล			
รหัสตำบล* Subdistrict ID	<input type="text"/>	ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
ชื่อตำบล Subdistrict name	<input type="text"/>	ว.ด.ป. สิ้นสุด To date	<input type="text"/>

	รหัสตำบล Subdistrict ID	ชื่อตำบล Subdistrict name	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	410101	หมากแข้ง	21-06-1932	31-12-9999
2	410102	หนองบัว	21-06-1932	31-12-9999

6.5 เมนู C04 บันทึกอำเภอ สังกัดจังหวัด



บันทึกอำเภอ สังกัดจังหวัด			
อำเภอ* District	<input type="text" value="เมืองอุดรธานี"/>	ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
สังกัดจังหวัด* Province	<input type="text" value="อุดรธานี"/>	ว.ด.ป. สิ้นสุด To date	<input type="text"/>

	อำเภอ District	สังกัดจังหวัด Province	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	เมืองอุดรธานี	อุดรธานี	21-06-1932	31-12-9999
2	เมืองหนองบัวลำภู	หนองบัวลำภู	01-12-1993	31-12-9999

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.6 เมนู C05 บันทึกตำบล สังกัดอำเภอ



บันทึกตำบล สังกัดอำเภอ			
ตำบล* Subdistrict	หมากแข้ง	ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
สังกัดอำเภอ* District	เมืองอุดรธานี	ว.ด.ป. สิ้นสุด To date	<input type="text"/>

	ตำบล Subdistrict	สังกัดอำเภอ District	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	หมากแข้ง	เมืองอุดรธานี	21-06-1932	31-12-9999
2	หนองบัว	เมืองหนองบัวลำภู	21-06-1932	31-12-9999

6.7 เมนู C06 บันทึกชื่อถนน



บันทึกชื่อถนน			
รหัสถนน* Road ID	<input type="text"/>	ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
ชื่อถนน Road name	<input type="text"/>	ว.ด.ป. สิ้นสุด To date	<input type="text"/>

	รหัสถนน Road ID	ชื่อถนน Road name	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	11	อุดรธานี	21-06-1932	31-12-9999
2	12	หนองบัวลำภู	21-06-1932	31-12-9999

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.7 เมนู C07 บันทึกรหัสบ้าน



บันทึกรหัสบ้านเลขที่			
รหัสบ้าน* Address ID	<input type="text"/>	บ้านเลขที่ Number	<input type="text"/>
หมู่ Moo	<input type="text"/>	ซอย Soi	<input type="text"/>
ถนน Road	อุตรดิตถ์	รหัสไปรษณีย์ Zipcode	<input type="text"/>

	รหัสบ้าน Address ID	เลขที่ Number	หมู่ Moo	ซอย Soi	ถนน Road	รหัสไปรษณีย์ Zipcode	แก้ไข Edit	ลบข้อมูล Delete
1	1001	1	1	null	อุตรดิตถ์	41000		
2	1002	2	1	null	อุตรดิตถ์	41000		
3	1003	3	1	null	อุตรดิตถ์	41000		
4	3001	1	3	null	หนองบัวลำภู	41390		
5	4001	1	4	null	อุตรดิตถ์	41000		
6	5001	1	5	null	หนองบัวลำภู	41390		

6.9 เมนู C08 บันทึกรหัสบ้าน สังกัดตำบล



บันทึกรหัสบ้าน สังกัดตำบล			
รหัสบ้าน* Address ID	<input type="text"/>	ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
สังกัดตำบล* Subdistrict	หมากแข้ง	ว.ด.ป. สิ้นสุด To date	<input type="text"/>

	รหัสบ้าน Address ID	สังกัดตำบล Subdistrict	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	1001	หมากแข้ง	21-06-1932	31-12-9999
2	1002	หมากแข้ง	21-06-1932	31-12-9999
3	1003	หมากแข้ง	21-06-1932	31-12-9999
4	3001	หนองบัว	21-06-1932	31-12-9999
5	4001	หมากแข้ง	21-06-1932	31-12-9999
6	5001	หนองบัว	21-06-1932	31-12-9999

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.10 เมนู C09 บันทึกบุคคล กบรหัสบ้าน



บันทึกบุคคล กบรหัสบ้าน			
เลขประจำตัวประชาชน* Identification Number	<input type="text"/>	ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
รหัสบ้าน Address ID	<input type="text"/>	ว.ด.ป. สิ้นสุด To date	<input type="text"/>

	เลขประจำตัวประชาชน Identification Number	รหัสบ้าน Address ID	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	10000	5001	14-07-1986	31-12-9999
2	11111	1001	15-05-1969	31-12-9999
3	22222	1001	15-05-1969	31-12-9999
4	33333	1002	01-01-1958	31-12-9999
5	44444	1002	01-01-1958	31-12-9999
6	55555	1003	01-01-1958	31-12-9999
7	66666	3001	28-02-1976	31-12-9999
8	77777	4001	08-08-1965	31-12-9999
9	88888	5001	14-07-1986	31-12-9999
10	99999	5001	14-07-1986	31-12-9999

6.11 เมนู C10 แสดงข้อมูลเขต ที่อยู่บุคลากร



แสดงข้อมูลเขต ที่อยู่บุคลากร			
เลขประจำตัวประชาชน* Identification Number	<input type="text"/>	ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
		ว.ด.ป. สิ้นสุด To date	<input type="text"/>

เลขประจำตัว ประชาชน Iden Number	รหัสบ้าน Addres ID	เลขที่ Number	หมู่ Moo	ซอย Soi	ถนน Road	ตำบล Subdistrict	อำเภอ District	จังหวัด Province	รหัส ไปรษณีย์ Zipcode	
1	10000	5001	1	5	null	หนองบัวสำถุ	หนองบัว	เมืองหนองบัวสำถุ	หนองบัวสำถุ	41390
2	11111	1001	1	1	null	ลุดรคชยั	หมากคั้ง	เมืองลุดรคชานั	ลุดรคชานั	41000
3	22222	1001	1	1	null	ลุดรคชยั	หมากคั้ง	เมืองลุดรคชานั	ลุดรคชานั	41000
4	33333	1002	2	1	null	ลุดรคชยั	หมากคั้ง	เมืองลุดรคชานั	ลุดรคชานั	41000
5	44444	1002	2	1	null	ลุดรคชยั	หมากคั้ง	เมืองลุดรคชานั	ลุดรคชานั	41000
6	55555	1003	3	1	null	ลุดรคชยั	หมากคั้ง	เมืองลุดรคชานั	ลุดรคชานั	41000
7	66666	3001	1	3	null	หนองบัวสำถุ	หนองบัว	เมืองหนองบัวสำถุ	หนองบัวสำถุ	41390
8	77777	4001	1	4	null	ลุดรคชยั	หมากคั้ง	เมืองลุดรคชานั	ลุดรคชานั	41000
9	88888	5001	1	5	null	หนองบัวสำถุ	หนองบัว	เมืองหนองบัวสำถุ	หนองบัวสำถุ	41390
10	99999	5001	1	5	null	หนองบัวสำถุ	หนองบัว	เมืองหนองบัวสำถุ	หนองบัวสำถุ	41390

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.12 เมนู C11 แสดงสถิติจำนวนประชากร

1) เมื่อคลิกปุ่ม แสดง ณ ปัจจุบัน (Current Time) จะแสดงสถิติจำนวนประชากร ณ เวลาปัจจุบัน จากรูปจะเห็นว่า มีจำนวนประชากรทั้งสิ้น 10 คน โดยอาศัย จ.อุตรธานี 6 คน และอาศัย จ.หนองบัวลำภู คน



แสดงสถิติจำนวนประชากร

รหัสจังหวัด* Province ID	<input type="text"/>
ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
ว.ด.ป. สิ้นสุด To date	<input type="text"/>

รหัสจังหวัด Province ID	ชื่อจังหวัด Province name	จำนวนประชากร Person	
1	41	อุตรธานี	6
2	39	หนองบัวลำภู	4

2) เมื่อคลิกที่ปุ่ม แสดงทุกช่วงเวลา (All Time) จะแสดงสถิติในอดีต จนถึงปัจจุบัน จะเห็นว่า จ.อุตรธานี มีประชากร 10 คน ซึ่งรวมประชากรอาศัย อ.หนองบัวลำภู ที่เคยสังกัด จ.อุตรธานี ด้วย และในขณะเดียวกัน ก็แสดงจำนวนประชากร ของ จ.หนองบัวลำภู 4 คน



แสดงสถิติจำนวนประชากร

รหัสจังหวัด* Province ID	<input type="text"/>
ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
ว.ด.ป. สิ้นสุด To date	<input type="text"/>

	รหัสจังหวัด Province ID	ชื่อจังหวัด Province name	จำนวนประชากร Person
1	41	อุตรธานี	10
2	39	หนองบัวลำภู	4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ประวัติการศึกษา

7.1 เมนูหลัก ประวัติการศึกษา



- 📍 D01 บันทึกชื่อประเทศ
- 📍 D02 บันทึกชื่อสถานศึกษา
- 📍 D03 บันทึกสถานศึกษา สังกัดประเทศ
- 📍 D04 บันทึกระดับการศึกษา
- 📍 D05 บันทึกวุฒิการศึกษา
- 📍 D06 บันทึกสาขาวิชาเอก
- 📍 D06 บันทึกประวัติการศึกษาของบุคคลากร
- 📍 D07 แสดงข้อสนเทศประวัติการศึกษา
- 📍 D08 แสดงสถิติจำนวนบุคคลากรด้านการศึกษา

7.2 เมนู D01 บันทึกชื่อประเทศ



บันทึกชื่อประเทศ			
รหัสประเทศ* Country ID	<input type="text"/>	ว.ต.ป. เริ่มต้น* From date	<input type="text"/>
ชื่อประเทศ Country name	<input type="text"/>	ว.ต.ป. สิ้นสุด To date	<input type="text"/>

	รหัสประเทศ Country ID	ชื่อประเทศ Country name	ว.ต.ป. เริ่มต้น From date	ว.ต.ป. สิ้นสุด To date
1	10	สยาม	06-04-1782	24-06-1939
2	10	ไทย	24-06-1939	31-12-9999

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3 เมนู D02 บันทึกชื่อสถานศึกษา

1) ถ้าคลิกปุ่มคำสั่ง แสดง ณ ปัจจุบัน (Current Time) จะแสดงชื่อสถาบัน ณ ปัจจุบัน



บันทึกชื่อสถานศึกษา			
รหัสสถานศึกษา* Institution ID	<input type="text"/>	ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
ชื่อสถานศึกษา Institution name	<input type="text"/>	ว.ด.ป. สิ้นสุด To date	<input type="text"/>

	รหัสสถานศึกษา Institution ID	ชื่อสถานศึกษา Institution name	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	1500	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ	27-12-2007	null
2	1600	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	20-02-1986	null
3	19100	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	18-01-2005	null

2) ถ้าคลิกปุ่มคำสั่ง แสดงทุกช่วงเวลา (All Time) จะแสดงชื่อสถาบันทั้งหมด รวมทั้งชื่อปัจจุบัน และชื่อในอดีตด้วย จากรูปจะเห็นว่า สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง รหัส 1600 มีการเปลี่ยนชื่อมาแล้วทั้งหมด 5 ครั้ง

	รหัสสถานศึกษา Institution ID	ชื่อสถานศึกษา Institution name	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	1500	สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตพระนครเหนือ	24-04-1971	20-02-1986
2	1500	สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ	20-02-1986	27-12-2007
3	1500	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ	27-12-2007	null
4	1600	วิทยาลัยโทรคมนาคมธนบุรี	01-05-1964	24-04-1971
5	1600	สถาบันเทคโนโลยีพระจอมเกล้า ศูนย์ธนบุรี	24-04-1971	29-06-1974
6	1600	สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตธนบุรี ลาดกระบัง	29-06-1974	22-04-1979
7	1600	สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตเจ้าคุณทหาร ลาดกระบัง	22-04-1979	20-02-1986
8	1600	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	20-02-1986	null
9	19100	วิทยาลัยเทคโนโลยีและอาชีวศึกษา	27-02-1975	15-09-1988
10	19100	สถาบันเทคโนโลยีราชมงคล	15-09-1988	18-01-2005
11	19100	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	18-01-2005	null

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.4 เมนู D03 บันทึกสถานศึกษา สังกัดประเทศ



บันทึกสถาบัน สังกัดประเทศ	
สถาบัน* Institution	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ประเทศ* Country	ไทย

	รหัสสถาบัน Institution ID	สถาบัน Institution	ประเทศ Country	แก้ไข Edit	ลบข้อมูล Delete
1	1500	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ	ไทย		
2	1600	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	ไทย		
3	19100	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	ไทย		

7.5 เมนู D04 บันทึกระดับการศึกษา



บันทึกรหัสระดับการศึกษา	
รหัสระดับการศึกษา* Level ID	<input type="text"/>
ชื่อระดับการศึกษา Level name	<input type="text"/>
ชื่อย่อระดับการศึกษา Level name-abbreviation	<input type="text"/>

	รหัส Level ID	ชื่อระดับการศึกษา Level name	ชื่อย่อระดับการศึกษา Level name-abbreviation	แก้ไข Edit	ลบข้อมูล Delete
1	40	ปริญญาตรี	ป.ตรี		
2	50	ประกาศนียบัตรบัณฑิต	ป.บัณฑิต		
3	60	ปริญญาโท	ป.โท		
4	70	ประกาศนียบัตรบัณฑิตชั้นสูง	ป.บัณฑิตชั้นสูง		
5	80	ปริญญาเอก	ป.เอก		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.6 เมนู D05 บันทึกวุฒิการศึกษา



บันทึกวุฒิการศึกษา			
รหัสวุฒิ* Qualification ID	<input type="text"/>		
ชื่อวุฒิภาษาไทย Qualification in Thai	<input type="text"/>	ชื่อวุฒิภาษาไทย-อักษรย่อ Qualification in Thai-abbreviation	<input type="text"/>
ชื่อวุฒิภาษาอังกฤษ Qualification in English	<input type="text"/>	ชื่อวุฒิภาษาอังกฤษ-อักษรย่อ Qualification in English-abbreviation	<input type="text"/>

	รหัสวุฒิ Qualifi ID	ชื่อวุฒิภาษาไทย Qualification in Thai	ชื่อวุฒิภาษาไทย-ย่อ Qualification in Thai-abbr	ชื่อวุฒิภาษาอังกฤษ Qualification in English	ชื่อวุฒิภาษาอังกฤษ-ย่อ Qualification in Eng-abbr	แก้ไข Edit	ลบข้อมูล Delete
1	159	วิทยาศาสตร์บัณฑิต	วท.บ.	Bachelor of Science	B.S.		
2	204	วิศวกรรมศาสตรดุษฎีบัณฑิต	วศ.ด.	Doctor of Engineering	D.Eng.		
3	206	วิศวกรรมศาสตรบัณฑิต	วศ.บ.	Bachelor of Engineering	B.Eng.		
4	209	วิศวกรรมศาสตรมหาบัณฑิต	วศ.ม.	Master of Engineering	M.Eng.		

7.7 เมนู D06 บันทึกสาขาวิชาเอก



บันทึกสาขาวิชาเอก	
รหัสสาขาวิชาเอก* Major ID	<input type="text"/>
ชื่อสาขาวิชาเอก-ภาษาไทย Major name of Thai	<input type="text"/>
ชื่อสาขาวิชาเอก-ภาษาอังกฤษ Major name of English	<input type="text"/>

	รหัส Major ID	ชื่อสาขาวิชาเอก-ภาษาไทย Major name of Thai	ชื่อสาขาวิชาเอก-ภาษาอังกฤษ Major name of English	แก้ไข Edit	ลบข้อมูล Delete
1	11	วิศวกรรมไฟฟ้า	Electrical Engineering		
2	12	วิศวกรรมคอมพิวเตอร์	Computer Engineering		
3	13	วิศวกรรมอิเล็กทรอนิกส์และ โทรคมนาคม	Electronic and Telecommunication Engineering		
4	21	เทคโนโลยีชีวภาพ	Biotechnology		
5	22	วิทยาการคอมพิวเตอร์	Computer science		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.8 เมนู D07 บันทึกประวัติการศึกษาของบุคลากร



บันทึกประวัติการศึกษาของบุคลากร			
เลขประจำตัวประชาชน* Identification Number	<input type="text"/>		
ระดับการศึกษา* Level	ป.ตรี	สถาบันการศึกษา* Institution	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
วุฒิการศึกษา* Qualification	วศ.บ.	ผลการเรียนเฉลี่ย GPA	<input type="text"/>
สาขาวิชาเอก* Major	วิศวกรรมไฟฟ้า	วันที่สำเร็จการศึกษา Graduation Day	<input type="text"/>

เลขประจำตัวประชาชน	ระดับ Level	วุฒิ Qualification	สาขาวิชาเอก Major	สถาบันการศึกษา Institution	GPA	วันที่สำเร็จ Graduation Day	แก้ไข Edit	ลบข้อมูล Delete	
1	33333	ป.ตรี	วศ.บ.	วิศวกรรมโทรคมนาคม	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	4	31-03-1974		
2	33333	ป.โท	วศ.ม.	วิศวกรรมไฟฟ้า	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	4	31-03-1977		
3	33333	ป.เอก	วศ.ด.	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	4	31-03-2007		
4	35003	ป.ตรี	วศ.บ.	วิศวกรรมไฟฟ้า	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	3.5	31-03-1982		
5	35003	ป.โท	วศ.ม.	วิศวกรรมไฟฟ้า	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ	4	31-03-1987		
6	66666	ป.ตรี	วศ.บ.	วิศวกรรมคอมพิวเตอร์	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	3	31-03-2001		
7	66666	ป.โท	วศ.ม.	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	3.6	31-03-2014		
8	88888	ป.ตรี	วศ.บ.	วิศวกรรมคอมพิวเตอร์	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	4	31-03-2007		


7.9 เมนู D08 แสดงข้อสนเทศประวัติการศึกษา

แสดงข้อสนเทศประวัติการศึกษา			
เลขประจำตัวประชาชน* Identification Number	<input type="text"/>	ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
รหัสสถานศึกษา* Institution ID	<input type="text"/>	ว.ด.ป. สิ้นสุด To date	<input type="text"/>

เลขประจำตัวประชาชน ID Number	ระดับ Level	วุฒิ Qualification	สาขาวิชาเอก Major	สถาบันการศึกษา Institution	GPA	วันที่สำเร็จ Graduation Day	
1	33333	ป.ตรี	วศ.บ.	วิศวกรรมโทรคมนาคม	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	4	31-03-1974
2	33333	ป.โท	วศ.ม.	วิศวกรรมไฟฟ้า	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	4	31-03-1977
3	33333	ป.เอก	วศ.ด.	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	4	31-03-2007
4	35003	ป.ตรี	วศ.บ.	วิศวกรรมไฟฟ้า	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	3.5	31-03-1982
5	35003	ป.โท	วศ.ม.	วิศวกรรมไฟฟ้า	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ	4	31-03-1987
6	66666	ป.ตรี	วศ.บ.	วิศวกรรมคอมพิวเตอร์	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	3	31-03-2001
7	66666	ป.โท	วศ.ม.	วิศวกรรมคอมพิวเตอร์	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	3.6	31-03-2014
8	88888	ป.ตรี	วศ.บ.	วิศวกรรมคอมพิวเตอร์	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	4	31-03-2007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.10 เมนู D09 แสดงสถิติจำนวนบุคลากรด้านการศึกษา



แสดงทุกช่วงเวลา
All Time



แสดง ณ ปัจจุบัน
Current Time



กลับไปที่เมนูหลัก
Close windows

แสดงสถิติจำนวนบุคลากรด้านการศึกษา			
รหัสสถาบัน Institution id	<input type="text"/>	ว.ด.ป. เริ่มต้น From date	<input type="text"/>
		ว.ด.ป. สิ้นสุด To date	<input type="text"/>

	รหัสสถาบัน ID	สถาบันการศึกษา Institution	จำนวนบุคลากร Person
1	1500	มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ	1
2	19100	มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี	3
3	1600	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	4

8. ประวัติการทำงาน

8.1 เมนูหลัก ประวัติการทำงาน



Work) **ประวัติการทำงาน**

-  E01 **บันทึกประเภทบุคลากร**
-  E02 **บันทึกประเภทสายการทำงาน**
-  E03 **บันทึกรหัสตำแหน่งวิชาการ**
-  E04 **บันทึกรหัสตำแหน่งบริหาร**
-  E05 **บันทึกรหัสคณะ**
-  E06 **บันทึกรหัสภาควิชา/แผนกงาน**
-  E07 **บันทึกภาควิชา สังกัดคณะ**
-  E08 **บันทึกอัตรากำลัง กับบุคลากร**
-  E09 **บันทึกบุคลากร กับ ประเภทสายงาน**
-  E10 **บันทึกบุคลากร สังกัดภาควิชา/แผนก**
-  E11 **บันทึกบุคลากร กับ ตำแหน่งทางวิชาการ**
-  E12 **บันทึกบุคลากร กับ ตำแหน่งทางบริหาร**
-  E13 **บันทึกบุคลากร กับ เงินเดือน**
-  E14 **แสดงสถิติตำแหน่งทางวิชาการ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.2 เมนู E01 บันทึกประเภทบุคลากร



บันทึกประเภทบุคลากร	
รหัสประเภทบุคลากร* Employee type id	<input type="text"/>
ชื่อประเภทบุคลากร Employee type name	<input type="text"/>

	รหัสประเภทบุคลากร Employee type id	ชื่อประเภทบุคลากร Employee type name	แก้ไข Edit	ลบข้อมูล Delete
1	1	ข้าราชการ		
2	2	พนักงานมหาวิทยาลัย		
3	3	พนักงานราชการ		
4	4	ลูกจ้างประจำ		
5	5	ลูกจ้างชั่วคราว		

8.3 เมนู E02 บันทึกประเภทสายการทำงาน



บันทึกประเภทสายการทำงาน	
รหัสประเภทสายการทำงาน* Field type id	<input type="text"/>
ชื่อประเภทสายการทำงาน Field type name	<input type="text"/>

	รหัสประเภทสายการทำงาน Field type id	ชื่อประเภทสายการทำงาน Field type name	แก้ไข Edit	ลบข้อมูล Delete
1	1	สายวิชาการ		
2	2	สายวิชาสนับสนุน		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.4 เมนู E03 บันทึกรหัสตำแหน่งวิชาการ



บันทึกรหัสตำแหน่งวิชาการ	
รหัสตำแหน่ง* Position academic id	<input type="text"/>
ชื่อตำแหน่งทางวิชาการ Position academic name	<input type="text"/>

	รหัสตำแหน่ง Position academic id	ชื่อตำแหน่งทางวิชาการ Position academic name	แก้ไข Edit	ลบข้อมูล Delete
1	1	ศาสตราจารย์		
2	2	รองศาสตราจารย์		
3	3	ผู้ช่วยศาสตราจารย์		
4	4	อาจารย์		
5	5	เจ้าหน้าที่บริหารงานทั่วไป		

8.5 เมนู E04 บันทึกรหัสตำแหน่งบริหาร



บันทึกรหัสตำแหน่งบริหาร	
รหัสตำแหน่ง* Position admin id	<input type="text"/>
ชื่อตำแหน่งบริหาร Position admin name	<input type="text"/>

	รหัสตำแหน่ง Position admin id	ชื่อตำแหน่งบริหาร Position admin name	แก้ไข Edit	ลบข้อมูล Delete
1	1	อธิการบดี		
2	2	รองอธิการบดี		
3	3	ผู้ช่วยอธิการบดี		
4	4	คณบดีคณะวิศวกรรมศาสตร์		
5	5	คณบดีคณะสถาปัตยกรรมศาสตร์		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.6 เมนู E05 บันทึกรหัสคณะ



บันทึกชื่อคณะ			
รหัสคณะ* Faculty id	<input type="text"/>	ว.ด.ป. เริ่มต้น From date	<input type="text"/>
ชื่อคณะ Country name	<input type="text"/>	ว.ด.ป. สิ้นสุด To date	<input type="text"/>

รหัสคณะ Faculty id	ชื่อคณะ Faculty name	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	คณะวิศวกรรมศาสตร์	24-04-1971	31-12-9999
2	คณะสถาปัตยกรรมศาสตร์	24-04-1971	31-12-9999
3	คณะครุศาสตร์อุตสาหกรรม	24-04-1971	31-12-9999
4	กองกลาง	24-04-1971	31-12-9999
5	กองการเจ้าหน้าที่	24-04-1971	31-12-9999

8.7 เมนู E06 บันทึกรหัสภาควิชา/แผนกงาน



บันทึกชื่อภาควิชา/แผนกงาน			
รหัส* Department id	<input type="text"/>	ว.ด.ป. เริ่มต้น From date	<input type="text"/>
ชื่อภาควิชา/แผนกงาน Department name	<input type="text"/>	ว.ด.ป. สิ้นสุด To date	<input type="text"/>

รหัส Department id	ชื่อภาควิชา/แผนกงาน Department name	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	ภาควิชาวิศวกรรมไฟฟ้า	24-04-1971	31-12-9999
2	ภาควิชาวิศวกรรมอิเล็กทรอนิกส์	24-04-1971	31-12-9999
3	ภาควิชาวิศวกรรมคอมพิวเตอร์	24-04-1971	31-12-9999
4	ภาควิชาวิศวกรรมโทรคมนาคม	24-04-1971	31-12-9999
5	ภาควิชาวิศวกรรมเครื่องกล	24-04-1971	31-12-9999
6	แผนกคอมพิวเตอร์	24-04-1971	31-12-9999

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.5 เมนู E07 บันทึกภาควิชา สังกัดคณะ



บันทึกภาควิชา สังกัดคณะ			
ภาควิชา* Department	<input type="text" value="ภาควิชาวิศวกรรมไฟฟ้า"/>	ว.ด.ป. เริ่มต้น From date	<input type="text"/>
สังกัดคณะ Faculty	<input type="text" value="คณะวิศวกรรมศาสตร์"/>	ว.ด.ป. สิ้นสุด To date	<input type="text"/>

	ภาควิชา Department	สังกัดคณะ Faculty	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	ภาควิชาวิศวกรรมไฟฟ้า	คณะวิศวกรรมศาสตร์	24-04-1971	31-12-9999
2	ภาควิชาวิศวกรรมอิเล็กทรอนิกส์	คณะวิศวกรรมศาสตร์	24-04-1971	31-12-9999
3	ภาควิชาวิศวกรรมคอมพิวเตอร์	คณะวิศวกรรมศาสตร์	24-04-1971	31-12-9999
4	ภาควิชาวิศวกรรมโทรคมนาคม	คณะวิศวกรรมศาสตร์	24-04-1971	31-12-9999
5	ภาควิชาวิศวกรรมเครื่องกล	คณะวิศวกรรมศาสตร์	24-04-1971	31-12-9999
6	แผนกคอมพิวเตอร์	กองการเจ้าหน้าที่	24-04-1971	31-12-9999

8.9 เมนู E08 บันทึกอัตรากำลัง กับบุคลากร



บันทึกเลขอัตรา กับบุคลากร			
เลขอัตรา* Employee number	<input type="text"/>	ว.ด.ป. เริ่มต้น From date	<input type="text"/>
ประเภทบุคลากร* Employee type	<input type="text" value="ข้าราชการ"/>	ว.ด.ป. สิ้นสุด To date	<input type="text"/>
เลขประจำตัวประชาชน Identification Number	<input type="text"/>		

	เลขอัตรา Employee number	ประเภทบุคลากร Employee type	เลขประจำตัวประชาชน Citizen id	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	1	ข้าราชการ	33333	01-06-1977	null
2	2	ข้าราชการ	35003	01-06-1982	null
3	3	ข้าราชการ	66666	04-06-2001	null
4	1	พนักงานมหาวิทยาลัย	88888	01-06-2007	null

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.10 เมนู E09 บันทึกบุคลากร กับ ประเภทสายงาน



บันทึกบุคลากร กับ ประเภทสายงาน			
เลขประจำตัวประชาชน* Identification Number	<input type="text"/>	ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
ประเภทสายงาน Field type	สายวิชาการ ▼	ว.ด.ป. สิ้นสุด To date	<input type="text"/>

	เลขประจำตัวประชาชน Identification Number	ประเภทสายงาน Field type	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	33333	สายวิชาการ	01-06-1977	null
2	35003	สายวิชาการ	01-06-1982	null
3	66666	สายวิชาการ	04-06-2001	null
4	88888	สายวิชาสนับสนุน	01-06-2007	null

8.11 เมนู E10 บันทึกบุคลากร สังกัดภาควิชา/แผนก



บันทึกบุคลากร สังกัดภาควิชา/แผนก			
เลขประจำตัวประชาชน* Identification Number	<input type="text"/>	ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
ภาควิชา/แผนก Department	ภาควิชาวิศวกรรมไฟฟ้า ▼	ว.ด.ป. สิ้นสุด To date	<input type="text"/>

	เลขประจำตัวประชาชน Identification Number	ภาควิชา/แผนก Department	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	33333	ภาควิชาวิศวกรรมคอมพิวเตอร์	01-06-1977	null
2	35003	ภาควิชาวิศวกรรมอิเล็กทรอนิกส์	01-06-1982	null
3	66666	ภาควิชาวิศวกรรมคอมพิวเตอร์	04-06-2001	null
4	88888	แผนกคอมพิวเตอร์	01-06-2007	null

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.12 เมนู E11 บันทึกบุคลากร กับ ตำแหน่งทางวิชาการ

1) ถ้าคลิกปุ่มคำสั่ง แสดง ณ ปัจจุบัน (Current Time) จะแสดงตำแหน่งทางวิชาการ ณ เวลาปัจจุบัน



บันทึกบุคลากร กับ ตำแหน่งทางวิชาการ			
เลขประจำตัวประชาชน* Identification Number	<input type="text"/>	ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
ตำแหน่งวิชาการ Position Academic	ศาสตราจารย์	ว.ด.ป. สิ้นสุด To date	<input type="text"/>

	เลขประจำตัวประชาชน Identification Number	ตำแหน่งวิชาการ Position Academic	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	33333	ศาสตราจารย์	01-10-2000	null
2	35003	ผู้ช่วยศาสตราจารย์	01-10-1989	null
3	66666	อาจารย์	04-06-2001	null
4	88888	นักวิชาการคอมพิวเตอร์	01-06-2007	null

2) ถ้าคลิกปุ่มคำสั่ง แสดงทุกช่วงเวลา (All Time) จะแสดงตำแหน่งทางวิชาการ ทั้งหมด ตั้งแต่อดีต จนถึงปัจจุบัน จากรูปจะเห็นว่า เลขประจำตัวประชาชน 33333 ได้รับตำแหน่งทางวิชาการมาแล้ว 4 ครั้ง ตามลำดับ คือ อาจารย์ ผู้ช่วยศาสตราจารย์ รองศาสตราจารย์ และ ศาสตราจารย์ ในปัจจุบัน

	เลขประจำตัวประชาชน Identification Number	ตำแหน่งวิชาการ Position Academic	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	33333	อาจารย์	01-06-1977	01-10-1980
2	33333	ผู้ช่วยศาสตราจารย์	01-10-1980	01-10-1987
3	33333	รองศาสตราจารย์	01-10-1987	01-10-2000
4	33333	ศาสตราจารย์	01-10-2000	null
5	35003	อาจารย์	01-06-1982	01-10-1989
6	35003	ผู้ช่วยศาสตราจารย์	01-10-1989	null
7	66666	อาจารย์	04-06-2001	null
8	88888	นักวิชาการคอมพิวเตอร์	01-06-2007	null

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.13 เมนู E12 บันทึกบุคลากร กับ ตำแหน่งทางบริหาร


1) ถ้าคลิกปุ่มคำสั่ง แสดง ณ ปัจจุบัน (Current Time) จะแสดงตำแหน่งทางบริหาร ณ เวลาปัจจุบัน จากรูปจะเห็นว่า บุคลากรเลขประจำตัวประชาชน 33333 ดำรงตำแหน่งในปัจจุบัน 2 ตำแหน่ง คือ อธิการบดี และ หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์

						
แสดงทุกช่วงเวลา All Time	แสดง ณ ปัจจุบัน Current Time	แสดง ณ ช่วงเวลา Valid Time	ลบข้อมูล Delete	บันทึกข้อมูล Insert data	ปรับปรุงข้อมูล Update data	กลับไปเมนูหลัก Close windows

บันทึกบุคลากร กับ ตำแหน่งทางบริหาร				
เลขประจำตัวประชาชน* Identification Number	<input type="text"/>	ว.ด.ป. เริ่มต้น* From date	<input type="text"/>	
ตำแหน่งบริหาร Position Admin	อธิการบดี	ว.ด.ป. สิ้นสุด To date	<input type="text"/>	

	เลขประจำตัวประชาชน Identification Number	ตำแหน่งบริหาร Position Admin	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	33333	หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์	01-06-1982	null
2	33333	อธิการบดี	01-06-2012	null
3	35003	หัวหน้าภาควิชาวิศวกรรมไฟฟ้า	01-06-1982	null

2) ถ้าคลิกปุ่มคำสั่ง แสดงทุกช่วงเวลา (All Time) จะแสดงตำแหน่งทางบริหาร ตั้งแต่อดีตจนถึงปัจจุบัน จากรูปจะเห็นว่า บุคลากรเลขประจำตัวประชาชน 33333 เคยดำรงตำแหน่งมาแล้ว 3 ตำแหน่ง และในปัจจุบัน เขาดำรงตำแหน่ง 2 ตำแหน่ง

						
แสดงทุกช่วงเวลา All Time	แสดง ณ ปัจจุบัน Current Time	แสดง ณ ช่วงเวลา Valid Time	ลบข้อมูล Delete	บันทึกข้อมูล Insert data	ปรับปรุงข้อมูล Update data	กลับไปเมนูหลัก Close windows

บันทึกบุคลากร กับ ตำแหน่งทางบริหาร				
เลขประจำตัวประชาชน* Identification Number	<input type="text"/>	ว.ด.ป. เริ่มต้น* From date	<input type="text"/>	
ตำแหน่งบริหาร Position Admin	อธิการบดี	ว.ด.ป. สิ้นสุด To date	<input type="text"/>	

	เลขประจำตัวประชาชน Identification Number	ตำแหน่งบริหาร Position Admin	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	33333	หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์	01-06-1982	null
2	33333	รองคณบดีคณณะวิศวกรรมศาสตร์	01-06-1986	01-06-1990
3	33333	คณบดีคณณะวิศวกรรมศาสตร์	01-06-1990	01-06-1994
4	33333	รองอธิการบดี	01-06-2000	01-06-2004
5	33333	อธิการบดี	01-06-2012	null
6	35003	หัวหน้าภาควิชาวิศวกรรมไฟฟ้า	01-06-1982	null

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.14 เมนู E13 บันทึกบุคลากร กับ เงินเดือน

1) ถ้าคลิกปุ่มคำสั่ง แสดง ณ ปัจจุบัน (Current Time) จะแสดงอัตราเงินเดือน ณ เวลาปัจจุบัน



บันทึกบุคลากร กับ อัตราเงินเดือน			
เลขประจำตัวประชาชน* Identification Number	<input type="text"/>	ว.ต.ป. เริ่มต้น* From date	<input type="text"/>
อัตราเงินเดือน Salary	<input type="text"/>	ว.ต.ป. สิ้นสุด To date	<input type="text"/>

	เลขประจำตัวประชาชน Identification Number	อัตราเงินเดือน Salary	ว.ต.ป. เริ่มต้น From date	ว.ต.ป. สิ้นสุด To date
1	33333	50000	01-06-2010	null
2	35003	35000	01-06-2010	null
3	66666	20000	01-06-2010	null
4	88888	18000	01-06-2010	null

2) ถ้าคลิกปุ่มคำสั่ง แสดงทุกช่วงเวลา (Time All) จะแสดงอัตราเงินเดือน ตั้งแต่อดีต จนถึงปัจจุบัน จากรูปในหน้าต่อไป จะแสดงอัตราเงินเดือนที่ได้รับทั้งหมดของบุคลากร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บันทึกบุคลากร กับ อัตราเงินเดือน			
เลขประจำตัวประชาชน* Identification Number	<input type="text"/>	ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
อัตราเงินเดือน Salary	<input type="text"/>	ว.ด.ป. สิ้นสุด To date	<input type="text"/>

	เลขประจำตัวประชาชน Identification Number	อัตราเงินเดือน Salary	ว.ด.ป. เริ่มต้น From date	ว.ด.ป. สิ้นสุด To date
1	33333	15000	01-06-1977	01-06-1980
2	33333	20000	01-06-1980	01-06-1985
3	33333	25000	01-06-1985	01-06-1990
4	33333	30000	01-06-1990	01-06-1995
5	33333	35000	01-06-1995	01-06-2000
6	33333	40000	01-06-2000	01-06-2005
7	33333	45000	01-06-2005	01-06-2010
8	33333	50000	01-06-2010	null
9	35003	15000	01-06-1982	01-06-1985
10	35003	20000	01-06-1985	01-06-1990
11	35003	23000	01-06-1990	01-06-1995
12	35003	26000	01-06-1995	01-06-2000
13	35003	29000	01-06-2000	01-06-2005
14	35003	32000	01-06-2005	01-06-2010
15	35003	35000	01-06-2010	null
16	66666	15000	04-06-2001	01-06-2005
17	66666	18000	01-06-2005	01-06-2010
18	66666	20000	01-06-2010	null
19	88888	15000	01-06-2007	01-06-2010
20	88888	18000	01-06-2010	null

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.15 เมนู E14 แสดงสถิติตำแหน่งทางวิชาการ

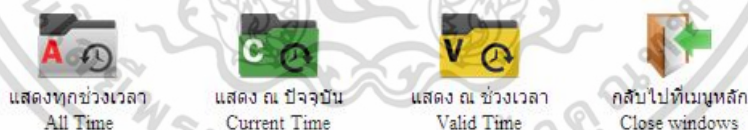
1) ถ้าคลิกปุ่มคำสั่ง แสดง ณ ปัจจุบัน (Current Time) จะแสดงสถิติตำแหน่งทางวิชาการ ณ เวลาปัจจุบัน



แสดงสถิติตำแหน่งทางวิชาการ	
ว.ด.ป. เริ่มต้น* From date	<input type="text"/>
ว.ด.ป. สิ้นสุด To date	<input type="text"/>

	ตำแหน่งวิชาการ Position Academic	จำนวน Person
1	ศาสตราจารย์	1
2	ผู้ช่วยศาสตราจารย์	1
3	อาจารย์	1
4	นักวิชาการคอมพิวเตอร์	1

2) ถ้ากำหนดช่วงเวลาที่ต้องการแสดงผล โดยกำหนดวันเริ่มต้น และวันสิ้นสุด แล้วคลิกปุ่มคำสั่ง แสดง ณ ช่วงเวลา (Valid Time) จะได้ข้อมูลสถิติตำแหน่งทางวิชาการ ณ เวลาที่กำหนด ดังรูปที่ปรากฏนี้



แสดงสถิติตำแหน่งทางวิชาการ	
ว.ด.ป. เริ่มต้น* From date	<input type="text" value="01/01/2000"/>
ว.ด.ป. สิ้นสุด To date	<input type="text" value="01/06/2000"/>

สถิติระหว่าง 01/01/2000 ถึงวันที่ 01/06/2000

	ตำแหน่งวิชาการ Position Academic	จำนวน Person
1	รองศาสตราจารย์	1
2	ผู้ช่วยศาสตราจารย์	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-นามสกุล นายสิทธิ รักถนอม
 วัน เดือน ปีเกิด 28 กุมภาพันธ์ 2519
 ที่อยู่ 16/06927 หมู่ 10 ถ.พหลโยธิน ต.คลองหนึ่ง อ.คลองหลวง
 จ.ปทุมธานี 12120
 โทร. 08-4025-5136
 อีเมลล์ tosit@hotmail.com

ประวัติการศึกษา **2538** ประกาศนียบัตรวิชาชีพ (ปวช.) สาขาช่างอิเล็กทรอนิกส์
 วิทยาลัยเทคนิคชลบุรี อ.เมือง จ.ชลบุรี
2540 ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.) สาขาช่างเทคนิคคอมพิวเตอร์
 สถาบันเทคโนโลยีราชมงคล วิทยาเขตพระนครเหนือ จ.กรุงเทพมหานคร
2544 วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์
 สถาบันเทคโนโลยีราชมงคล ต.คลองหก อ.ธัญบุรี จ.ปทุมธานี
2552 เข้าศึกษาต่อระดับปริญญาโท หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต
 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

สถานที่ทำงาน อาจารย์ประจำสาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
 มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี ต.คลองหก อ.ธัญบุรี จ.ปทุมธานี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้