

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ฐานข้อมูลเชิงเวลา

Temporal Database



โดย

นายวุฒิชัย รัฐปฐมวงศ์ 39014499 4D
นายศรรัตน์ เทียงสกุล 39014508 4D



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เลขหมู่.....
เลขทะเบียน..... 37084
เอกสารเป็นเอกสารสงวนลิขสิทธิ์
วัน, เดือน, ปี, 30 ส.ค. 2543

การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
หากกรณีใดที่ผู้ใช้ต้องการนำเอกสารไปใช้ให้ติดต่อแจ้งเจ้าหน้าที่ดูแลห้องสมุดเพื่อตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายงานเรื่อง (ภาษาไทย)
(ภาษาอังกฤษ)

ฐานข้อมูลเชิงเวลา
Temporal Database

จัดทำโดย

นายวุฒิชัย รัษฎามวงศ์ 39014499 4D
นายศระรัน เทียงสกุล 39014508 4D

อาจารย์ที่ปรึกษา



อาจารย์บัณฑิต พิทยา

รายงานฉบับนี้ได้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ลงชื่อ..... *บัณฑิต พิทยา* อาจารย์ที่ปรึกษา
(อาจารย์บัณฑิต พิทยา)

วันที่ ...21... เมษายน / 2563

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

รายงานและผลงาน ฐานข้อมูลที่ขึ้นกับเวลาที่ได้จัดทำสำเร็จลุล่วงได้เป็นอย่างดี ขอกราบ
ขอบพระคุณบิดา-มารดา ที่คอยอบรมสั่งสอนและได้ให้โอกาสการศึกษาแก่ข้าพเจ้าอีกทั้งยังคอยให้กำลังใจ
ใจอยู่เสมอ และทางด้านเนื้อหา ความรู้ในการทำงานครั้งนี้ได้รับการชี้แนะและคำปรึกษาจาก อาจารย์ บัณ
ฑิต พัสยา ซึ่งเป็นอาจารย์ที่ปรึกษา ผู้จัดทำรู้สึกทราบบ้างซึ่งในความอนุเคราะห์จากท่านและขอขอบพระคุณ
เป็นอย่างสูง

ขอขอบคุณ Informix ที่ได้เอื้อเฟื้อผลิตภัณฑ์ต่างๆ ในการทำงานจนสำเร็จลุล่วงไปด้วยดี ขอขอบคุณ
เพื่อนๆ ที่ห้อง database ที่คอยให้คำปรึกษารวมถึง อุปกรณ์ และเครื่องมือต่างๆ อย่างใกล้ชิดเสมอมา

สุดท้ายขอขอบคุณภาควิชาคอมพิวเตอร์ ที่ได้สนับสนุนสถานที่ต่างๆและเครื่องมือ ต่างๆรวมทั้ง
คำปรึกษาจากอาจารย์ทุกท่าน

คุณค่าและประโยชน์อันพึงมีจากรายงานฉบับนี้ ผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน

อ.วิชัย ธีระพงษ์

(นายวิชัย ธีระพงษ์)

อ.วิวัฒน์ ธีระพงษ์

(นายวิวัฒน์ ธีระพงษ์)

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลเชิงเวลา

นายวุฒิชัย รัฐปฐมวงศ์

นายศระรัน เทียงสกุล

อ. บัณฑิต พัสยา อาจารย์ที่ปรึกษา

ภาคเรียนที่ 2 ปีการศึกษา 2542

บทคัดย่อ

ระบบฐานข้อมูลที่ใช้งานกันอยู่ทั่วไป มีการจัดเก็บข้อมูลเพียงค่าข้อมูลเดี่ยวเท่านั้น (ค่าข้อมูลสุดท้าย) มิได้มีการจัดเก็บค่าข้อมูลเก่าหรือข้อมูลในอนาคตไว้เลย แต่โปรแกรมประยุกต์ส่วนใหญ่มักจะต้องการเรียกค้นค่าข้อมูลเก่า ๆ หรือค่าข้อมูลในอนาคต ซึ่งระบบฐานข้อมูลที่มีอยู่ไม่สามารถตอบสนองความต้องการนี้ได้

ระบบฐานข้อมูลเชิงเวลา เป็นระบบฐานข้อมูลที่สามารถจัดเก็บและเรียกค้นข้อมูลเชิงเวลาได้ โดยระบบฐานข้อมูลนี้มีการจัดการเวลาที่ข้อมูลเป็นจริง และเวลาที่ประมวลผลข้อมูลให้อย่างอัตโนมัติ นอกจากนี้ผู้ใช้งานยังสามารถสร้างคอลัมน์ของเวลาเข้าไปได้อีกด้วย แต่ผู้ใช้งานจะต้องจัดการกับคอลัมน์ของเวลานี้เอง

ผู้วิจัยจึงได้สร้าง โปรแกรมประยุกต์ซึ่งมักมีการทำงานคล้ายกับระบบฐานข้อมูลเชิงเวลา โดยมีการจัดเก็บข้อมูลบนฐานข้อมูลเชิงวัตถุสัมพันธ์ และได้นำเอารหัสเฉพาะ (OID) ของระบบฐานข้อมูลเชิงวัตถุสัมพันธ์นี้มาแก้ปัญหาค่าการเปลี่ยนแปลงค่าของคอลัมน์ที่เป็น Key อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Temporal Database

Mr. Wuttichai Ratpathomwong

Mr. Saran Tiengsakul

Assoc. Mr. Bandit Passaya

2nd Semester, Educational Year 1999

Abstract

Database system usually store data in single value (last value of data). They did not cover any past and future data which most of applied program usually need, so normal program can not be response this requirement.

Temporal Database system can be stored and query temporal data. This database system manage valid time and time which automatic processing by system (transaction time). Moreover user can create time column but they must manage time column by themselves.

So I made applied program which work similar temporal database system and store data on object Object relational database and use object identification (OID) of a Object relational database system for solve the key column changing problem.

สารบัญ

		หน้า
	กิตติกรรมประกาศ	I
	บทคัดย่อ	II
	Abstract	III
บทที่ 1	บทนำ	1
	วัตถุประสงค์ของโครงการ	1
	ขอบเขตของโครงการ	1
	วิธีการดำเนินงาน	2
บทที่ 2	ทฤษฎีของ ฐานข้อมูลเชิงเวลา(Temporal Database)	3
	2.1 Temporal Database คืออะไร	3
	2.2 ความซ้ำซ้อนที่เกิดขึ้นในตาราง	4
	2.3 การใช้ SQL ในการป้องกันการเกิด row ที่เป็น value equivalent duplicate	6
	2.4 การใช้ SQL ในการป้องกันการเกิด row ที่เป็น non-sequenced duplicate	6
	2.5 การใช้ SQL ในการป้องกันการเกิด row ที่เป็น Current Duplicated	7
	2.6 การใช้ SQL ในการป้องกันการเกิด row ที่เป็น Sequenced Duplicates	7
	2.7 การเรียกค้น Valid-Time(Querying Valid-Time State Tables)	8
	2.8 การเลือกข้อมูลที่เกี่ยวข้องกับTEMPORAL (TEMPORAL PROJECTION AND SELECTION)	9
	2.8.1 Current querying	9
	2.8.2 Sequence query	9
	2.8.3 Nonsequenced variant	10
	2.9 การ JOIN ที่มีเวลามาเกี่ยวข้อง(TEMPORAL JOINS)	10
	2.9.1 snapshot version	10
	2.9.2 Current version	11
	2.9.3 nonsequenced version	11
	2.9.4 Sequenced version	12
	2.10 การจัดการกับ Valid-Time State Table (Modifying Valid-Time State Tables)	14
	2.10.1 Current Modification	14
	2.10.2 Sequenced Modification	18
	2.10.3 Nonsequence Modification	21
	2.11 Transaction-Time State Tables	22
	2.11.1 Maintaining The Audit Log	22
	2.11.2 Querying The Audit Log	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

(ต่อ)

	หน้า
2.11.3 Modifying The Audit Log	25
2.12 การสนับสนุน Temporal ภายใต้มาตรฐาน SQL	26
2.12.1 VALIDTIME SELECT	26
2.12.2 VALIDTIME <period exp> SELECT	27
2.12.3 NONSEQUENCED VALIDTIME SELECT	27
2.12.4 NONSEQUENCED VALIDTIME <period exp> SELECT	27
2.13 ทฤษฎีการแปลงจาก SQL Temporal เป็น SQL มาตรฐาน	27
2.13.1 Current Validtime	27
2.13.2 Sequence Validtime	28
2.13.3 Nonsequence Validtime	29
2.13.4 Current Transaction time	29
2.13.5 Sequence Transaction time	30
2.13.6 Nonsequenced Transaction	30
2.14 การ Join	31
2.14.1 Current Validtime	31
2.14.2 Sequenced Validtime	32
บทที่ 3 การนำรหัสเฉพาะ(Object Identification: OID) มาประยุกต์ใช้กับข้อมูลที่เปลี่ยนแปลงตามเวลา	35
3.1 การกำหนดค่าเริ่มต้นให้กับชนิดของข้อมูล Serial	35
3.2 การใช้ Serial กับ Integer	35
3.3 ข้อมูลชนิด Serial8(n)	36
3.4 ตัวอย่างการสร้างข้อมูลชนิด Serial	36
3.5 ประโยชน์ของ OID (Object Identification)	37
บทที่ 4 การออกแบบโปรแกรมระบบงานลงทะเบียน	38
4.1 การออกแบบ ERT (Entity Relationship Model) สำหรับระบบงานลงทะเบียนนักศึกษา	38
4.2 รูปแสดงERT model สำหรับระบบงานลงทะเบียนนักศึกษา	40
บทที่ 5 โปรแกรมการตรวจสอบและสอบถามการลงทะเบียนของนักศึกษา	41
5.1 สภาพแวดล้อมของระบบ	42
5.1.1 โปรแกรมที่ใช้ในการติดตั้งเพื่อทำงานกับฐานข้อมูล Informix	42
5.1.2 สถาปัตยกรรมของการเชื่อมต่อ Client และ Server	43
5.2 โปรแกรมจัดการข้อมูลในฐานข้อมูล (Manage Data Application)	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

(ต่อ)

หน้า

5.2.1สถาปัตยกรรมของโปรแกรมจัดการข้อมูลในฐานข้อมูล	43
5.2.2 การทำงานของโปรแกรมจัดการข้อมูลในฐานข้อมูล	44
5.3 โปรแกรมค้นหาข้อมูลโดย SQL และตัวสร้างภาษา SQL (SQL Editor Include SQL Builder)	46
5.3.1 สถาปัตยกรรมของโปรแกรมค้นหาข้อมูลโดย SQL และตัวสร้างภาษา SQL	46
5.3.2 การทำงานของโปรแกรมค้นหาข้อมูลโดย SQL และตัวสร้างภาษา SQL	46
5.4 วิธีการติดต่อกับฐานข้อมูลที่ใช้ในระบบงานลงทะเบียนนักศึกษา	49
5.4.1 การติดต่อกับฐานข้อมูลโดยผ่านระบบจัดการฐานข้อมูลของ อินฟอร์มิทส์	49
5.4.2 ลำดับชั้นของ Data Director Objects	50
5.5 การใช้ Data Director Objects	57
5.5.1 การใช้ properties และ methods	57
5.5.2 การใช้ Visual Basic Code Window	58
5.5.3 Viewing Properties และ methods	58
5.5.4 การใช้ properties	58
5.5.5 การใช้ Method	58
5.5.6 ประโยชน์ของ Properties และ methods	59
5.6 เริ่มต้นการใช้ operation ของ DDO	60
5.7 การส่งคำสั่ง SQL เข้าไป query โดยตรง	69
5.8 การติดต่อกับฐานข้อมูลโดยผ่านระบบจัดการแบบเจ็ต (JET Database Engine)	73
5.8.1 โครงสร้างของระบบจัดการฐานข้อมูลภายนอกผ่าน DAO และ JET	73
5.8.2 องค์ประกอบต่างๆของโครงสร้างข้อมูลใน JET	74
5.8.3 การสร้างพื้นที่ใช้งานระบบจัดการฐานข้อมูล	75
5.8.4 วิธีเปิดฐานข้อมูลที่ต้องการ	76
5.8.5 ตรวจสอบและค้นหาตารางในฐานข้อมูล	76
5.8.6 ออปเจ็ทข้อมูล(RecordSet Object)	78
5.8.7 ขั้นตอนและวิธีการต่างๆที่จะใช้งานออปเจ็ทข้อมูลมีดังนี้	78
5.8.8 การติดต่อบนระบบจัดการฐานข้อมูลภายนอกโดยตรง(ODBC Direct)	79
5.8.9 ออปเจ็ททางค้ำฐานข้อมูลสำหรับการใช้งานแบบ ODBC Direct	80
5.8.10 ออปเจ็ทการติดต่อ (Connection Object)	81
5.8.11 ออปเจ็ทข้อมูล(Recordset Object)	82
5.8.12 เปรียบเทียบข้อดีข้อเสียของระบบจัดการฐานข้อมูลแบบต่างๆ	83
บทที่6 บทสรุป	84
6.1 ข้อเสนอแนะ และ แนวทางการศึกษาต่อ	84

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ
(ต่อ)

	หน้า
6.2 สรุป	84
ภาคผนวก	86
ก. คู่มือการติดตั้งโปรแกรมเพื่อใช้งานฐานข้อมูลนักศึกษา	86
ขั้นตอนการติดตั้งเพื่อใช้งาน โปรแกรม	86
ติดตั้ง โปรแกรมระบบงานทะเบียนนักศึกษา	88
ติดตั้ง ODBC และแก้ไขไฟล์ Host เพื่อติดต่อกับ Database Server	90
ข. วิธีการใช้โปรแกรม	91
โปรแกรมการจัดการข้อมูลในฐานข้อมูลเชิงเวลา	91
โปรแกรม SQL Editor	98
โปรแกรมค้นหาข้อมูลจากคำถามพื้นฐาน	101
ค. Source Code ของโปรแกรม	103
โปรแกรมจัดการฐานข้อมูลเชิงเวลา	103
โปรแกรม SQL Editor	151
โปรแกรม QueryBuilder	180
บรรณานุกรม	201

บทที่ 1

บทนำ

ข้อมูลที่เกิดขึ้นอยู่ในฐานข้อมูลทั่ว ๆ ไป โดยส่วนใหญ่แล้วจะเก็บเฉพาะข้อมูลที่เป็นความจริงในปัจจุบันเท่านั้น ส่วนอดีตและอนาคตของข้อมูลนั้นจะไม่ถูกเก็บไว้ในฐานข้อมูล โดยข้อมูลในอดีตนั้นหมายถึงข้อมูลที่อยู่เคยอยู่ในฐานข้อมูลมาก่อน เมื่อมีการเปลี่ยนแปลงหรือลบข้อมูลออกไปจากฐานข้อมูลจะทำให้ข้อมูลนั้นหายไป แต่ข้อมูลที่ขึ้นกับเวลาที่เก็บอยู่ในฐานข้อมูลที่ขึ้นกับเวลานั้นจะมีการเพิ่มส่วนของเวลาติดไปกับส่วนที่เป็นข้อมูล โดยจะมีการเก็บช่วงเวลาที่เป็นจริงหรือช่วงเวลาที่ยังไม่ถูกเก็บในฐานข้อมูล ซึ่งจะต้องมีการจัดเก็บข้อมูลตามทฤษฎีของฐานข้อมูลที่ขึ้นกับเวลา เพื่อประโยชน์การค้นหาค่าข้อมูลในฐานข้อมูล นอกจากนี้ยังต้องมีการจัดการกับการเชื่อมต่อของเวลากับฐานข้อมูลเพื่อให้สอดคล้องความเข้าใจและไม่เกิดความสับสนต่อการใช้งาน

วัตถุประสงค์ของโครงการ

1. เพื่อศึกษาลักษณะของเวลาที่เก็บอยู่ในฐานข้อมูลเชิงเวลาทั้งแบบ Valid Time และ Transaction Time
2. เพื่อศึกษาวิธีการจัดการกับเวลาที่เก็บอยู่ในฐานข้อมูลเชิงเวลา ทั้งแบบ Valid Time และ Transaction Time
3. เพื่อศึกษาวิธีการในการนำ Object Relational DBMS เช่น Informix เข้ามาจัดการกับฐานข้อมูลเชิงเวลา
4. เพื่อศึกษาวิธีการแปลง ภาษา Temporal SQL ไปเป็น ภาษา SQL ธรรมดาเพื่อที่จะสามารถนำไปใช้ได้กับ Object Relational DBMS ได้
5. เพื่อศึกษา ลักษณะของmodel ที่จะใช้ในการอธิบายฐานข้อมูลเชิงเวลา รวมถึงแผนภาพที่ใช้อธิบายฐานข้อมูลเชิงเวลานี้ด้วย

ขอบเขตของโครงการ

1. ศึกษาความหมายของ ฐานข้อมูลเชิงเวลา รวมถึงวิธีการจัดเก็บเวลาลงในฐานข้อมูล Object Relational Database
2. สร้างระบบที่สามารถจัดการกับฐานข้อมูลเชิงเวลาได้ โดยจะทำหน้าที่จัดการในด้านของการ Insert Delete และ Update
3. สร้าง Application ที่สามารถแปลงภาษา Temporal SQL ไปเป็นภาษา SQL ธรรมดาเพื่อที่จะสามารถไปทำงานบนระบบจัดการข้อมูล Object Relational Database ได้
4. นำระบบจัดการกับฐานข้อมูลและโปรแกรมแปลงภาษา SQL เหล่านั้นไปทดลองใช้กับระบบสารสนเทศนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ทำการทดสอบระบบจัดการฐานข้อมูลที่ได้ กับระบบสารสนเทศนักศึกษา

วิธีการดำเนินงาน

1. ทำการค้นคว้าหาความหมายของ ฐานข้อมูลเชิงเวลา จากงานวิจัยต่างๆที่เกี่ยวข้อง
2. ทำการค้นคว้าหาโครงสร้างของภาษา Temporal SQL จาก ISO รวมถึง Model ที่สามารถอธิบายฐานข้อมูลเชิงเวลาได้
3. ศึกษาระบบจัดการฐานข้อมูล Informix เพื่อจะนำมาใช้เป็นระบบจัดการฐานข้อมูล รวมถึงศึกษาภาษา SQL และชนิดของข้อมูลที่มีอยู่ใน Informix
4. ศึกษาสารสนเทศนักศึกษาโดยนำข้อมูลมาจากจากระบบลงทะเบียนของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังแล้วนำมาดัดแปลงให้เหมาะสมกับระบบจัดการฐานข้อมูลเชิงเวลา
5. ทดสอบระบบสารสนเทศด้วย คำสั่ง SQL ต่างๆทั้งการ Insert Delete และ Update จะต้องสามารถแก้ปัญหาต่างๆที่เกี่ยวกับเวลาได้



บทที่ 2

ทฤษฎีของ ฐานข้อมูลเชิงเวลา(Temporal Database)

2.1 Temporal Database คืออะไร

ข้อมูลที่เก็บอยู่ในฐานข้อมูล ทัวๆไปส่วนใหญ่แล้วจะเก็บเฉพาะข้อมูลที่เป็นความจริงในปัจจุบันเท่านั้น อดีตและอนาคตของข้อมูลนั้นจะไม่ถูกเก็บไว้ในฐานข้อมูล ข้อมูลในอดีตนั้นหมายถึงข้อมูลที่อาจจะเคยอยู่ในdatabase มาก่อนแล้วมีการ update หรือ delete ออกไปจากฐานข้อมูล แล้ว ณเวลาปัจจุบัน

แต่ฐานข้อมูลเชิงเวลานั้น ข้อมูลที่ถูกเก็บอยู่ใน ฐานข้อมูลจะมีความแตกต่างจากข้อมูลที่เก็บในฐานข้อมูลทัวๆไป ตรงที่จะมีการเพิ่มส่วนของเวลาติดไปกับส่วนของข้อมูล เพื่อเก็บช่วงของเวลาที่ข้อมูลนี้เป็นจริง (valid time) หรือช่วงเวลาที่ข้อมูลถูกเก็บอยู่ใน database(transaction time) เวลาที่ถูกจัดเก็บอยู่ใน database นั้นแบ่งได้ 3 ชนิดดังนี้

- Valid Time ของความจริง(fact)หมายถึง เวลาที่ความจริงนั้นเป็นจริง เช่น

Id_number	Ename	Valid_from(mm/dd/yyyy)	Valid_to(mm/dd/yyyy)
508	Saran	03/01/1995	05/20/1998
499	Wuttichai	03/15/1995	12/31/9999

ตารางที่ 1 แสดงตัวอย่างของการเก็บvalid time ลงในฐานข้อมูล

จากตารางที่ 1 ในแถวแรกนั้นมีความหมายว่า นักศึกษาชื่อ saran มีรหัส 508 ตั้งแต่วันที่ 1 มีนาคม 1995 จนถึงวันที่ 20 พฤษภาคม 1998 และในแถวที่สอง มีความหมายว่า นักศึกษาชื่อ Wuttichai มีรหัส 499 ตั้งแต่วันที่ 15 มีนาคม 1995 จนถึงปัจจุบัน ก็ยังเป็นจริงอยู่ตลอดไป

- Transaction Time ของความจริงในฐานข้อมูล หมายถึง เวลาที่ ความจริงนั้นถูกเก็บอยู่ใน ฐานข้อมูลจริงๆ หรืออาจกล่าวง่ายๆได้ว่าเป็นเวลาเก็บไว้เพื่อติดตามการจัดการ(manipulate)กับฐานข้อมูล เช่น

Id_number	Ename	Tx_start(mm/dd/yyyy)	Tx_stop(mm/dd/yyyy)
508	Saran	03/01/1995	05/20/1998
499	Wuttichai	03/15/1995	12/31/9999

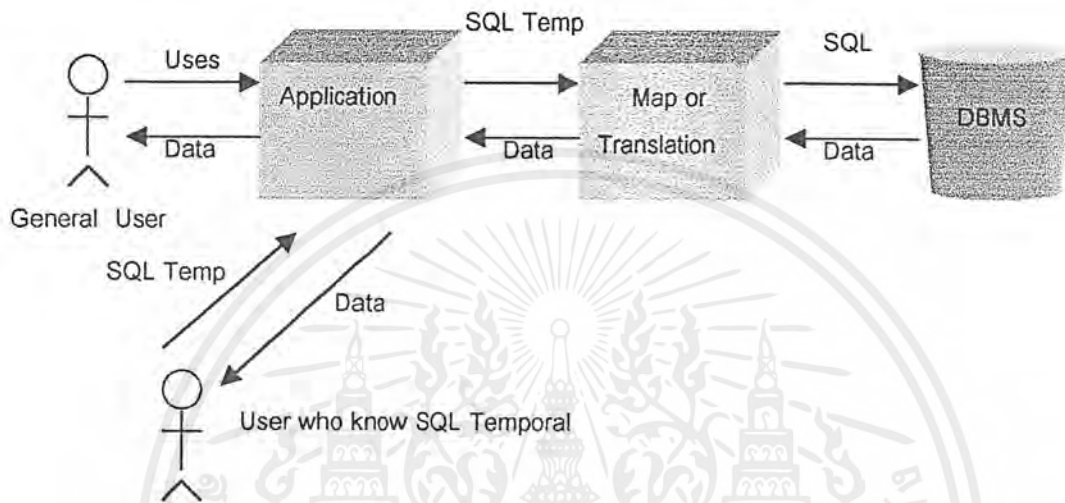
ตารางที่ 2 แสดงตัวอย่างของการเก็บvalid time ลงในฐานข้อมูล

จากตารางที่ 2 ในแถวแรกมีความหมายว่า ความจริงที่นักศึกษาชื่อ saran มีรหัส 508 ถูกเก็บลงในฐานข้อมูลเมื่อ 1 มีนาคม 1995 จนถึงวันที่ 20 พฤษภาคม 1998 และในแถวที่สอง มีความหมายว่า ความจริงที่นักศึกษาชื่อ wuttichai มีรหัส 499 ถูกเก็บลงในฐานข้อมูลตั้งแต่วันที่ 15 มีนาคม 1995 จนถึงปัจจุบันก็ยังอยู่ในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- User Defined Time หมายถึงคุณสมบัติของความจริงที่ไม่ต้องแปลความหมายของวันหรือเวลา อย่างเช่น วันเกิด หรือ วันแรงงาน ซึ่งเป็นคุณสมบัติที่ไม่ต้องการการจัดการทำให้ผู้ใช้จะต้องจัดการเอง ไม่ว่าจะ insert delete หรือ update ซึ่งจะไม่กล่าวถึงในรายงานฉบับนี้

ในการนำเอา Valid Time หรือ Transaction Time มาใช้ได้อย่างมีประสิทธิภาพนั้น จะต้องมีการจัดการกับการเชื่อมต่อของเวลากับฐานข้อมูล เพื่อให้ง่ายต่อความเข้าใจและไม่เกิดความสับสนแก่ผู้ใช้ โดยสามารถทำได้ดังนี้



รูปที่ 1 การเชื่อมต่อเวลากับฐานข้อมูล

โดยที่ ภาษา SQL/Temporal เป็นภาษา SQL ที่สนับสนุนการทำงานที่มีเวลาเข้ามาเกี่ยวข้อง และเข้าใจความหมายของ Valid time และ Transaction time ในตัวภาษาอยู่แล้ว แตกต่างจาก SQL แบบเดิมที่ไม่เข้าใจความหมายของ Valid time และ Transaction time

ใน SQL ที่ไม่สนับสนุน ภาษา Temporal การสร้างตารางและการเก็บข้อมูลอาจจะทำให้เกิดความซ้ำซ้อนขึ้นในตาราง ซึ่งจำเป็นต้องทำการป้องกันมิให้เกิดความซ้ำซ้อน ซึ่งจะได้กล่าวต่อไป

2.2 ความซ้ำซ้อนที่เกิดขึ้นในตาราง

ข้อมูลที่แปรผันตามเวลานั้น ถ้าตั้งกลายเป็นที่แพร่หลายเพราะว่าในทุกๆ 50 คำสั่งของ ภาษา SQL จะมีคำสั่งที่เกี่ยวข้องกับเวลา ในรายงานฉบับนี้จะอธิบายถึงการจัดการเกี่ยวกับ ข้อมูลที่แปรผันกับเวลา โดยใช้ ภาษา SQL โดยในแต่ละหัวข้อจะประกอบไปด้วย คำสั่งพร้อมทั้งตัวอย่างและการอธิบายประกอบ และในหัวข้อท้ายๆจะกล่าวถึงวิธีการใหม่ๆเกี่ยวกับข้อมูลที่เกี่ยวข้องกับเวลาอีกด้วย

ปกติการจัดเก็บข้อมูลลงในฐานข้อมูลจะเป็นการจัดเก็บความจริง(Fact) ของข้อมูลลงไป ในฐานข้อมูล มิได้จัดเก็บช่วงเวลาที่เป็น Fact นั้นเป็นจริง ลงไปด้วย แต่ถ้ามีการจัดเก็บเวลาต่างๆลงไปด้วย สิ่งที่มาคือ ความซ้ำซ้อนที่เกิดขึ้นกับตารางที่ใช้จัดเก็บ

Twin Time

พิจารณาตาราง NICUStatus ซึ่งเป็นตารางที่เก็บสถานะของผู้ป่วยที่รักษาตัวอยู่ที่ Neonatal Intensive care Unit Moines ซึ่งแสดงดังตารางที่ 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	1999-12-31
Alexis May	Fair	1997-12-02	1999-12-31

ตารางที่ 3 คัดทอนมาจาก ตาราง NICUStatus

From_date แสดงว่าผู้ป่วยมีอาการนั้นๆตั้งแต่วันที่

To_date แสดงว่าผู้ป่วยมีอาการเปลี่ยนไปในวันที่

ในแต่ละแถวนั้นจะบอกว่าคนไข้มีอาการเหล่านี้ในช่วงเวลาใด ซึ่งช่วงเวลานี้จะเป็นแบบ “Close-Open” กล่าวคือ From_date จะรวมอยู่ในช่วง แต่ To_date จะไม่รวมอยู่ช่วง เช่น ในแถวสุดท้ายนั้น วันที่ 2 ธันวาคม 1997 จะรวมอยู่ในช่วงเวลาที่ Alexis มีอาการ Fair แต่ในวันที่ 31 ธันวาคม 1999 จะไม่รวมอยู่ในช่วงนั้น

ในกรณีที่แถวไหนที่เป็นจริงในปัจจุบัน To_date จะมีค่าเป็น “forever” ซึ่งใน SQL-92 จะแทนด้วย 9999-12-31 แต่ใน DBMS บางตัวจะแทนปัจจุบันด้วย null value หรือบางตัวใช้เวลาในอดีตเช่น 1899 หรือ อาจใช้เวลาที่มากที่สุดที่เป็นไปได้เช่น 9999 แต่สำหรับ Informix ค่าที่มากที่สุดที่เป็นไปได้คือ 9456 ในตารางนี้ก็เหมือนกับตารางทั่วไปที่ประกอบไปด้วยหลายๆ Columns และ Timestamp ซึ่งเป็นตัวบอกว่าในแต่ละ row นั้นจะเป็นจริงในช่วงเวลาใด

Duplicate ใน SQL นั้นอาจแบ่งออกได้เป็น 4 ประเภท ได้แก่

- Non-sequence Duplicate :เป็นความซ้ำซ้อนที่เกิดขึ้นใน row ใดๆที่เหมือนกัน ในทุกๆ column เช่นใน 2 row สุดท้ายในตารางที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Value-equivalent Duplicate : เป็นความซ้ำซ้อนที่เกิดขึ้นเมื่อตัด column ที่เกี่ยวข้องกับเวลาออกไปแล้ว เช่น ใน 3 row สุดท้ายในตารางที่ 3
- Current Duplicate : เป็นความซ้ำซ้อนที่เกิดขึ้นในระยะเวลาใดเวลาหนึ่งในปัจจุบัน เช่น ถ้ากำหนดให้ปัจจุบันเป็นวันที่ 6 มกราคม 1998 จะได้ว่าเกิดความซ้ำซ้อนใน 3 row สุดท้าย
- Sequence Duplicate : เป็นความซ้ำซ้อนที่เกิดขึ้นบนช่วงเวลาใดๆ เช่น Alexis มีอาการอยู่ในชั้น Fair ใน 3 row สุดท้าย เมื่ออยู่ในช่วงเวลาดังกล่าว ตั้งแต่เดือนธันวาคม 1997 ถึง 11 วันแรกของปี 1998

2.3 การใช้ SQL ในการป้องกันการเกิด row ที่เป็น Value Equivalent Duplicate

```

Create Table NICUStatus(
    Name Char(15),
    Status Char(8),
    From_date Date,
    To_date Date,
    UNIQUE (Name,Status)
);

```

โดยหลักการแล้ว Value-Equivalent สามารถที่จะตอบคำถามที่ว่า “เมื่อใดที่ผู้ป่วยถูกกำหนดว่ามีอาการอยู่ในชั้นไหน

2.4 การใช้ SQL ในการป้องกันการเกิด row ที่เป็น Non-Sequenced Duplicate

```

Create Table NICUStatus(
    Name Char(15),
    Status Char(8),
    From_date Date,
    To_date Date,
    UNIQUE (Name,Status,From_date,To_date)
);

```

ความหมายของ Non-Sequence Duplicate ก็คือ “ผู้ป่วยไม่สามารถที่จะมีสองสถานะ ภายในระยะเวลาเดียวกันได้” อย่างไรก็ตามข้อกำหนดนี้สามารถที่จะยืดหยุ่นได้โดยการเลื่อนวันเริ่มต้นหรือวันสิ้นสุด เพื่อไม่ให้เกิดความเป็น Identical ดังนั้นผู้ป่วยจะได้รับอนุญาตให้มีสถานะเดียวกันได้เป็นหลายๆเวลา

2.5 การใช้ SQL ในการป้องกันการเกิด row ที่เป็น Current Duplicated

เพื่อเป็นการป้องกัน การเกิด Current Duplicate จึงต้องมีบางอย่างมาเกี่ยวข้องกับอีกเล็กน้อย ได้แก่

```

Create Table NICUStatus(
...
check not EXIST (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 <= N1.To_date and
N2.from_date <= Current_date and
Current_date < N2.To_date ))
)

```

ตารางที่ 4 การป้องกันการเกิด Current Duplicate ในตาราง NICUStatus

ความหมายก็คือว่า “ไม่มีผู้ป่วยรายใดในปัจจุบันที่สามารถ ที่มีค่าของสถานะเหมือนกันได้” แต่ทั้งหมดนี้อาจก่อให้เกิดปัญหาที่ว่าตารางดังกล่าวสามารถใช้ได้ในปัจจุบันแต่ไม่สามารถใช้ได้ในอนาคต แม้ว่าจะไม่มีการเปลี่ยนแปลงข้อมูลในตารางเลยก็ตาม วิธีแก้ปัญหานี้ก็คือการเพิ่ม Unique ลงไปด้วย

```

Create Table NICUStatus( ....
UNIQUE (Name,Status,To_date)

```

วิธีนี้สามารถทำงานได้เพราะว่า ข้อมูลปัจจุบันทั้งหมดจะมี To_date ที่เหมือนกันคือ 9999-12-31

2.6 การใช้ SQL ในการป้องกันการเกิด row ที่เป็น Sequenced Duplicates

การป้องกันการเกิด sequence duplicate ก็คล้ายๆกับการป้องกันการเกิด Current duplicate โดยให้ดูในตารางที่ 3 ความหมายของ sequenced duplicate ก็คือ “ไม่มีเวลาใดเลยที่ผู้ป่วยมีสถานะ 2 อย่าง”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Create Table NICUStatus(
...
check (not EXIST (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 ))
)
)

```

ตารางที่ 5 แสดงการป้องกันการเกิด Sequenced Duplicates ในตาราง NICUStatus

2.7 การเรียกค้น Valid-Time (Querying Valid-Time State Tables)

ใน ส่วนที่ 2 นี้จะได้อธิบายถึงการ Select ข้อมูลที่มีเวลาเข้ามาเกี่ยวข้อง ในตัวอย่างจะเป็นตารางของ Dr. Brad De Groot แห่งมหาวิทยาลัย Nebraska ใน Lincoln ซึ่งตั้งอยู่ทางตอนใต้ของเมือง Columbus โดยสนใจในการปรับปรุงและดูแล สุขภาพของวัว ซึ่งหวังที่จะพัฒนาความสัมพันธ์ระหว่าง สุขภาพของวัวที่อยู่ในคอกเดียวกันกับความเสี่ยงในการเกิดโรคระบาด ความสัมพันธ์นี้จะทำให้นำไปสู่ความเข้าใจว่าโรคระบาดสามารถระบาดไปได้ได้อย่างไร โดยที่ Dr Groot ได้รับข้อมูลจาก ฟาร์ม ต่างๆที่ได้เก็บข้อมูลของการขนย้ายวัวประมาณ 55,000 ตัว

วัวพวกนี้จะถูกแยกออกเป็นกลุ่มๆเรียกว่า lots ซึ่งส่วนย่อยๆของ lots จะถูกย้ายจาก pen (คอก) ไปสู่ pen และหนึ่งในตารางของ Dr. Groot ก็คือตาราง LOT_LOC ซึ่งจะเก็บว่า มีวัวจำนวนเท่าไรจากแต่ละ lot ที่อาศัยในแต่ละ pen ของ แต่ละ feed yard (ฟาร์ม) ดังตารางที่ 4

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

ตารางที่ 6 ตัดทอนมาจากตาราง LOT_LOC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางนี้เรียกว่า valid-time state table ซึ่งมันจะเก็บข้อมูลที่เป็นความจริงในช่วงหนึ่งของเวลา จาก From_date ไปถึง To_date และตารางนี้ยังอนุญาตให้มีการเคลื่อนย้ายได้พร้อมๆกันในหนึ่งวันอีกด้วย

2.8 การเลือกข้อมูลที่เกี่ยวข้องกับTEMPORAL (TEMPORAL PROJECTION AND SELECTION)

พิจารณาคำถามที่ไม่เกี่ยวกับเวลาที่ว่า “มีวัวจำนวนเท่าใดจาก lot 219 ใน feed yard 1 อาศัยอยู่ในแต่ละpen” ซึ่งสามารถเขียนได้เป็น SQL ดังนี้

```
Select PEN_ID,HD_CNT
From LOT_LOC
Where FDYD_ID = 1 and LOT_ID_NUM = 219
```

2.8.1 Current Querying

ถ้าเป็นแบบที่เกี่ยวกับ Current คำถามที่ว่า “มีวัวจำนวนเท่าใดจาก lot 219 ใน feed yard 1 ที่ปัจจุบันอาศัยอยู่ในแต่ละpen” แสดงได้ดังตารางที่ 7

SQL	ผลลัพธ์
<pre>Select PEN_ID,HD_CNT From LOT_LOC Where FDYD_ID = 1 and LOT_ID_NUM = 219 and TO_DATE = DATE '9999-12-31'</pre>	<pre>PEN_ID HD_CNT 2 43</pre>

ตารางที่ 7 “มีวัวจำนวนเท่าใดจาก lot 219 ใน Feed yard1 ที่ปัจจุบันอาศัยอยู่ในแต่ละ Pen”

2.8.2 Sequence Querying

“จงแสดงประวัติของวัวจาก lot 219 ในฟาร์มที่ 1 ที่อยู่ในแต่ละคอก” คำถามนี้เราสามารถใช่ Sequenced Querying มาประยุกต์ได้

```
Select PEN_ID, HD_CNT, From_date, To_date
From LOT_LOC
Where FD_YD=1 and LOT_ID_NUM = 219
```

ผลลัพธ์จะปรากฏดังตารางที่ 8

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

ตารางที่ 8 “มีวัวกี่ตัวจาก lot 219 ที่อยู่ในแต่ละคอก”

2.8.3 Nonsequenced Querying

“มีวัวกี่ตัวจาก lot 219 ในฟาร์มที่ 1 ที่ในบางเวลาอยู่ในแต่ละคอก” คำถามนี้ไม่สนใจว่าข้อมูลที่ได้นั้นเป็นจริงเมื่อใดและไม่ได้ถามจำนวนรวมด้วย ซึ่งสามารถเขียนเป็น SQL ได้ดังตารางที่ 9

SQL	ผลลัพธ์	
Select PEN_ID,HD_CNT	PEN_ID	HD_CNT
From LOT_LOC	1	43
Where FDYD_ID = 1	1	20
And LOT_ID_NUM=219	2	23
	2	43

ตารางที่ 9 “มีวัวกี่ตัวจาก lot 219 ในฟาร์มที่ 1 ที่ในบางเวลาอยู่ในแต่ละคอก”

2.9 การ JOIN ที่มีเวลาเกี่ยวข้องกับ (Temporal Joins)

พิจารณาความเกี่ยวข้องกันที่มากขึ้นดังเช่น “มีวัวจาก lot ใดบ้างที่อาศัยอยู่ในคอกเดียวกัน” คำถามดังกล่าวมานี้สามารถเขียนเป็น SQL ได้โดยใช้การ self-join เป็นหลักดังต่อไปนี้

2.9.1 Snapshot Version

```
Select L1.Lot_ID_NUM,L2.LOT_ID_NUM,L1.PEN_ID
From LOT_LOC_SNAPSHOT AS L1, LOT_LOC_SNAPSHOT AS L2
Where L1.LOT_ID_NUM < I2.LOT_ID_NUM AND
      L1.FDYD_ID = L2.FDYD_ID AND
      L1.PEN_ID = L2.PEN_ID
```

ผลลัพธ์จากคำถามนี้จะป็น null ถ้า snapshot ที่เวลาปัจจุบัน เพราะว่า ไม่มี lot ไหนเลขที่อยู่ร่วมกับ lot 219

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.2 Current Version

```

Select L1.Lot_ID_NUM,L2.LOT_ID_NUM,L1.PEN_ID
From LOT_LOC AS L1, LOT_LOC AS L2
Where L1.LOT_ID_NUM < L2.LOT_ID_NUM AND
      L1.FDYD_ID = L2.FDYD_ID AND
      L1.PEN_ID = L2.PEN_ID AND
      L1.TO_DATE = DATE'9999-12-31' AND
      L2.TO_DATE = DATE'9999-12-31'

```

ผลลัพธ์จากคำถามนี้จะ เป็น null เพราะว่า ไม่มี lot ไหนเลขที่ปัจจุบันอยู่ร่วมกับ lot 219

2.9.3 Nonsequenced Version

“lot ไหนบ้างที่อยู่ในคอกเดียวกัน อาจอยู่คนละเวลาก็ได้”

```

Select L1.Lot_ID_NUM,L2.LOT_ID_NUM,L1.PEN_ID
From LOT_LOC AS L1, LOT_LOC AS L2
Where L1.LOT_ID_NUM < L2.LOT_ID_NUM AND
      L1.FDYD_ID = L2.FDYD_ID AND
      L1.PEN_ID = L2.PEN_ID AND

```

ผลลัพธ์จากคำถามนี้จะว่า ทั้ง 3 lot อาศัยอยู่ในคอกที่ 1 เหมือนกัน

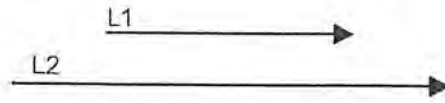
L1	L2	PEN_ID
137	219	1
137	219	1
137	374	1
219	374	1
219	374	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.4 Sequenced Version

“จงแสดงประวัติของการอาศัยร่วมกันในแต่ละคอก” คำถามข้างต้นนี้เป็นสิ่งที่ท้าทายมากในการอธิบายด้วยภาษา SQL ในการ join แบบ sequence นี้จะแบ่งออกเป็น 4 กรณี ขึ้นอยู่กับระยะเวลาที่สัมพันธ์กันได้แก่

- เมื่อ ระยะเวลาของ row L1 อยู่ภายในระยะเวลาของ row L2 ดังรูป



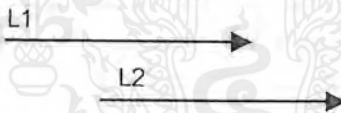
- เมื่อ ระยะเวลาของ row L2 อยู่ภายในระยะเวลาของ row L1 ดังรูป



- เมื่อ ระยะเวลาของ row L1 และ row L2 คาบเกี่ยวกันอยู่ โดยที่ L2 เกิดขึ้นก่อนดังรูป



- เมื่อ ระยะเวลาของ row L1 และ row L2 คาบเกี่ยวกันอยู่ โดยที่ L1 เกิดขึ้นก่อน ดังรูป



จากกรณีดังกล่าวสามารถนำมาเขียนเป็น SQL ได้ดังตารางที่ 10

```

SELECT
L1.LOT_ID_NUM,L2.LOT_ID_NUM,L1.PEN_ID,L1.FROM_DATE,L1.TO_DATE
FROM LOT_LOC AS L1,LOT_LOC AS L2
WHERE L1.LOT_IDNUM < L2.LOT_ID_NUM AND
      L2.FDYD_ID = L2.FDYD_ID AND
      L1.PEN_ID=L2.PEN_ID AND
      L2.FROM_DATE <= L1.FROM_DATE AND
      L1.TO_DATE <= I2.TO_DATE
UNION
SELECT

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

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

UNION

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

UNION

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

```

ตารางที่ 10 Lot ไหนบ้างที่อาศัยอยู่ร่วมกันในคอกหนึ่งๆ (Sequenced Version)

ผลลัพธ์ของการค้นหาจะได้เป็นตารางที่ 11 ซึ่งจะไม่มีการ sequenced Duplicate เลย

LOT_ID_NUM	LOT_ID_NUM	PEN_ID	From_date	To_date
219	374	1	1998-02-25	1998-03-01
219	374	1	1998-03-01	1998-03-14

ตารางที่ 11 แสดง lot ของวัวที่อาศัยอยู่ร่วมกันในคอกหนึ่งๆ (sequenced Version)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

ต่อไปนี้เป็น การแสดงการจัดการการเปลี่ยนแปลงกับ Valid-Time State Table ทั้ง current, sequence และ nonsequence โดยจะใช้ตัวอย่างของปศุสัตว์วัว ซึ่งมีการย้ายจากคอกหนึ่งไปยังอีกคอกหนึ่ง โดย GNDR_CODE เป็นการแสดงของวัวในขณะนั้น FROM_DATE และ TO_DATE เป็นคาบเวลาที่แสดงว่าแถวนั้นเป็นจริงอยู่ จากตาราง LOT วันที่ 23 มีนาคม 1998 ใน LOT 101 มีสถานะเป็น Steer , LOT 234 มีสถานะเป็น Claves ซึ่งถ้า TO_DATE มีค่าเป็น forever หมายความว่าแถวนั้นเป็นจริงในปัจจุบัน

ในการ Modification สามารถทำได้โดยแบ่งเป็นประเภทต่าง ๆ ได้ดังนี้

2.10.1 Current Modification

คำสั่ง SQL ต่อไปนี้เป็น การเพิ่มข้อมูลแบบ Current ซึ่ง Statement นี้จะทำการสร้าง Timestamp จากปัจจุบันไปยังอนาคต

```
INSERT INTO LOT
VALUES (433,'h',CURRENT_DATE,DATE '9999-12-31')
```

จะเป็นการลี้ที่สุดถ้าไม่สนใจ Column ที่เป็น Timestamp ซึ่งทำให้เกิดความสับสน ให้พิจารณาว่าถ้า นำ Lot 101 ออกไป จะเป็นการลบ โดยแสดงได้ดังนี้

```
DELETE FROM LOT
WHERE LOT_ID_NUM = 101
```

การลบข้อมูลแบบ Current บน Valid time state table แสดงเป็นคำสั่ง SQL เป็นการ Update โดย Current Deletion จะมีการกระทำจากปัจจุบันจนตลอดกาล

```
UPDATE LOT
SET TO_DATE = CURRENT_DATE
WHERE LOT_ID_NUM = 101
AND TO_DATE = DATE '9999-12-31'
```

LOT_ID_NUM	GNDR_CODE	FORM_DATE	TO_DATE
101	C	1998-01-01	1998-03-23
101	S	1998-03-23	9999-12-31
234	C	1998-02-17	9999-12-31
799	C	1998-03-12	9999-12-31

ตารางที่ 12 ตารางของ LOT

ในที่นี้เราจะพิจารณาเป็น 2 กรณี

1. General scenario - การเปลี่ยนแปลง Valid Time Table จะกระทำได้ทุก ๆ การเปลี่ยนแปลง
2. Restricted scenario - Current Modification เท่านั้นที่สามารถทำได้บนตาราง

Current Deletion ของ Lot 234 หมายความว่าลบ Lot 234 ออกจากตารางไป โดยจะถูก implement โดย general scenario เหมือนกับเป็นการ update และ delete ซึ่งแสดงได้ดังนี้

```

UPDATE LOT
SET TO_DATE = CURRENT_DATE
WHERE LOT_ID_NUM = 234
AND TO_DATE >= CURRENT_DATE
AND FROM_DATE < CURRENT_DATE

DELETE FROM LOT
WHERE LOT_IN_NUM = 234
AND FROM_DATE > CURRENT_DATE

```

โดยสามารถ execute 2 statement นี้ statement ใดก่อนก็ได้

จาก statement ข้างต้นเป็น operation ที่กระทำบนตารางที่ 13 และจะให้ผลเป็นตารางที่ 14 โดยที่ ตารางนี้ข้อมูลที LOT 234 หลังจาก Current date จะถูกลบออกหมด

LOT_ID_NUM	GNDR_CODE	FORM_DATE	TO_DATE
101	C	1998-01-01	1998-03-23
101	S	1998-03-23	9999-12-31
234	C	1998-02-17	1998-10-17
234	S	1998-10-17	9999-12-31
799	C	1998-03-12	9999-12-31

ตารางที่ 13 General Scenario ของ LOT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LOT_ID_NUM	GNDR_CODE	FORM_DATE	TO_DATE
101	C	1998-01-01	1998-03-23
101	S	1998-03-23	9999-12-31
234	C	1998-02-17	1998-07-29
799	C	1998-03-12	9999-12-31

ตารางที่ 14 ผลของ current deletion จากตารางที่ 13

พิจารณาการ update LOT 799 บนตารางที่เป็น nontemporal

```
UPDATE LOT
SET GNDR_CODE = 'S'
WHERE LOT_ID_NUM = 799
```

ในกรณีนี้ current update จะถูกกระทำเหมือนกับการ delete และ insert ในตารางที่ 15 แสดงการเปลี่ยนแปลงบน valid time state table โดยตารางที่ 17 แสดงการเปลี่ยนแปลงบน restrict scenario

Update จะเปลี่ยนค่า current value ที่ขณะนั้น และเพิ่มค่าใหม่เข้าไป update จะต้องเกิดขึ้นหลังจากการ insert

Current Update ใน general scenario แบ่งได้เป็น 3 กรณี

1. ช่วงเวลาที่ fact เป็นจริง (Period of validity) สิ้นสุดไปในอดีต update จะไม่ส่งผลใดๆ ต่อแถวนั้น
2. ถ้าแถวนั้น currently valid ส่วนที่อยู่ก่อนเวลาปัจจุบันจะถูกทำลายและแถวใหม่จะถูกแทรก โดยมี period of validity เริ่มต้นที่วันปัจจุบัน และจะสิ้นสุดที่เวลาเดียวกับแถวเดิม
3. ถ้าแถวนั้นมีจุดเริ่มต้นในอนาคต แถวนั้นจะถูก update ทั้งหมด โดยวิธีดังกล่าวต้องใช้งาน insert 1 ครั้ง และ update 2 ครั้ง (ดูตารางที่ 15) ซึ่งการ update ครั้งที่ 2 จะกระทำเมื่อใดก็ได้ แต่การ update ครั้งแรกจะเกิดขึ้นหลังจากการ insert

```
INSERT INTO LOT
SELECT DISTINCT 799,'S',CURRENT_DATE,DATE '9999-12-31'
FROM LOT
WHERE EXISTS (SELECT *
FROM LOT
WHERE LOT_ID_NUM = 799
AND TO_DATE = DATE '9999-12-31')
UPDATE LOT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SET TO_DATE = CURRENT_DATE
WHERE LOT_ID_NUM = 799
    AND GNDR_CODE <> 'S'
    AND TO_DATE = DATE '9999-12-31'
    
```

ตารางที่ 15 Current update ใน Restrict scenario

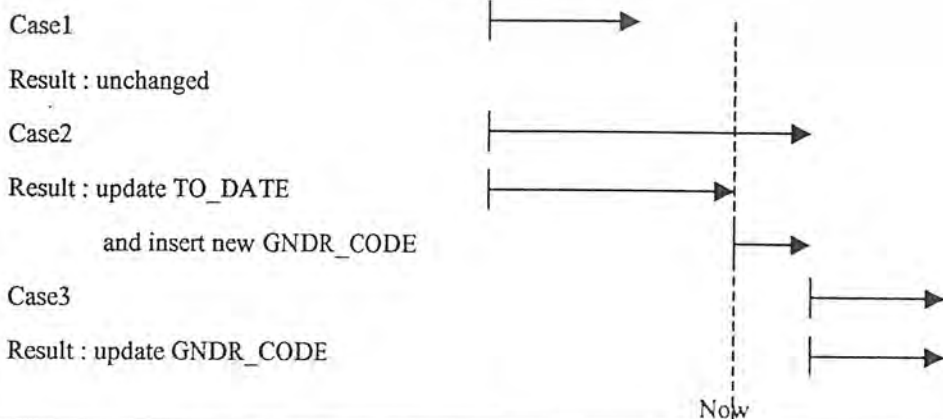
```

INSERT INTO LOT
SELECT LOT_ID_NUM,'S',CURRENT_DATE,TO_DATE
FROM LOT
WHERE LOT_ID_NUM = 799
    AND FROM_DATE <= CURRENT_DATE
    AND TO_DATE > CURRENT_DATE

UPDATE LOT
SET TO_DATE = CURRENT_DATE
WHERE LOT_ID_NUM = 799
    AND GNDR_CODE <> 'S'
    AND FROM_DATE < CURRENT_DATE
    AND TO_DATE > CURRENT_DATE

UPDATE LOT
SET GNDR_CODE = 'S'
WHERE LOT_ID_NUM = 799
    AND FROM_DATE >= CURRENT_DATE
    
```

ตารางที่ 16 Current update ใน General scenario



ตารางที่ 17 Current update ใน general scenario

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาข้อมูลข้างต้นหรือส่งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.2 Sequenced Modification

Current modifying จะมีการกระทำจากปัจจุบันไปสู่อนาคต แต่ sequence modification จะเปลี่ยนแปลงบนคาบเวลาใด ๆ เรียกว่า period of applicability ช่วงเวลานี้สามารถเป็นอดีต ในอนาคต หรือมีการ overlap กับเวลาปัจจุบัน

ในตอนก่อนเราจะทำ sequence modification โดยจะใช้ CURRENT_DATE แทนที่ จุดเริ่มต้นของ period of applicability และจุดสิ้นสุดจะแทนด้วย DATE'9999-12-31'

ใน sequence insertion จะทำให้เกิด period of applicability ตัวอย่างเช่น Lot 426 ซึ่งมี heifer อยู่ อยู่ในตารางตั้งแต่ 26 มีนาคม จนถึง 14 เมษายน

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

Sequence deletion จะมี 4 กรณีที่เป็นไปได้

ช่วงเวลาที่ fact เป็นจริง (Period of validity) ของแถวเดิม ทำการแสดงเหนือ Period of applicability สำหรับการลบ ถ้า

กรณี 1 แถวเดิมจะครอบคลุม period of applicability ทั้งหมด ดังนั้นเราจะไม่มีการแก้ไข ในช่วงคาบเวลาเริ่มต้นและสิ้นสุด ในคาบเวลาเริ่มต้นทำได้โดยการ Set ค่า TO_DATE ไปยังจุดเริ่มต้นของ period of application ส่วนในช่วงคาบเวลาสิ้นสุด จะทำการ insert เพิ่มขึ้น

กรณี 2 เราจะไม่ทำการแก้ไขในส่วนช่วงคาบเวลาเริ่มต้นของ period of validity ของแถวเดิมเท่านั้น ซึ่งสมมาตรกับ

กรณี 3 เราจะไม่ทำการแก้ไขในส่วนสุดท้ายของช่วงเวลา

กรณี 4 เราต้องลบทั้งแถว

ใน กรณี 1 จะมีผลกระทบต่อ 2 statements แรกของตารางที่ 18 ; statement ที่ 2 จะครอบคลุม กรณี 2 ; statement ที่ 3 จะครอบคลุม กรณี 3 และ statement ที่ 4 จะครอบคลุม กรณี 4

```
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 '1998-10-22'
UPDATE LOT
SET TO_DATE = DATE '1998-10-01'
WHERE LOT_ID_NUM = 234
AND FROM_DATE < DATE '1998-10-01'
```

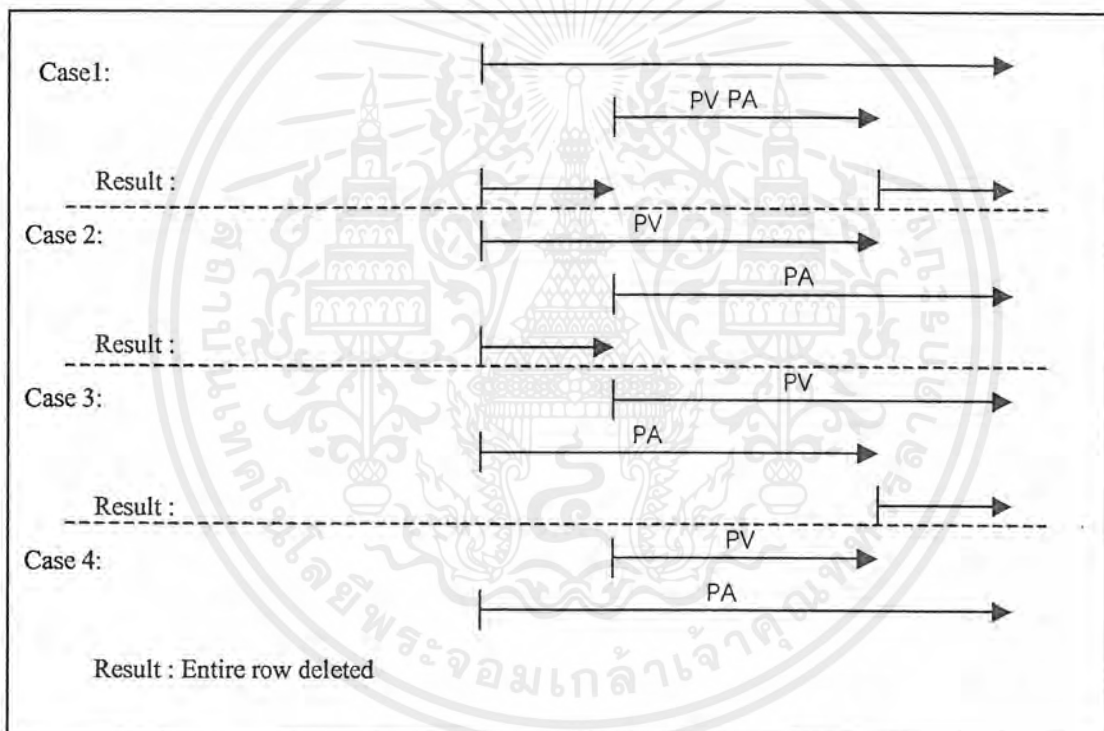
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

AND TO_DATE >= DATE '1998-10-01'
UPDATE LOT
SET FROM_DATE = DATE '1998-10-22'
WHERE LOT_ID_NUM = 234
AND FROM_DATE < DATE '1998-10-22'
AND TO_DATE >= DATE '1998-10-22'
DELETE FROM LOT
WHERE LOT_ID_NUM = 234
AND FROM_DATE >= DATE '1998-10-01'
AND TO_DATE <= DATE '1998-10-22'

```

ตารางที่ 18 Sequence deletion



รูปที่ 2 Sequence Deletion

Sequence update มีอยู่ 4 กรณี

- กรณีที่ 1 เราจะไม่แก้ไขในส่วนเริ่มต้นและสิ้นสุดของ period of applicability และจะส่งผลกระทบต่อส่วนที่ถูก update
- กรณี 2 เราจะไม่ทำการแก้ไขส่วนเริ่มต้น
- กรณี 3 เราจะไม่ทำการแก้ไขส่วนสิ้นสุด
- กรณี 4 เราจะไม่มีแก้ไขส่วน period of validity ซึ่งถูกปกคลุมโดย period of Applicability

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Statement ในตารางที่ 19 แสดง sequence update (period of applicability อยู่ที่ DATE '1998-03-01' ไปยัง DATE '1998-04-01') โดย statement แรกจะแสดง ส่วน initial ของ case1 และ 2 ; Statement ที่ 2 แสดงส่วนท้ายของ case 2 และ 3 ; update แรกจะครอบคลุมทั้ง 4 cases ; update ที่ 2 และ 3 จะเปลี่ยนแปลงจุดเริ่มต้นและสิ้นสุดของเวลาของส่วน update ให้สังเกตว่า update statement ทั้ง 3 จะไม่ส่งผลกับ insert statement ทั้ง 2 statement

```

INSERT INTO LOT
SELECT LOT_ID_NUM,GNDR_CODE,FROM_DATE,DATE '1998-03-01'
FROM LOT
WHERE LOT_ID_NUM = 799
      AND FROM_DATE < DATE '1998-03-01'
      AND TO_DATE > DATE '1998-03-01'

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 TO_DATE > DATE '1998-04-01'

UPDATE LOT
SET GNDR_CODE = 'S'
WHERE LOT_IN_NUM = 799
      AND FROM_DATE < DATE '1998-04-01'
      AND TO_DATE > DATE '1998-03-01'

UPDATE LOT
SET FROM_DATE = DATE '1998-03-01'
WHERE LOT_IN_NUM = 799
      AND FROM_DATE < DATE '1998-03-01'
      AND TO_DATE > DATE '1998-03-01'

UPDATE LOT
SET TO_DATE = DATE '1998-04-01'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

WHERE LOT_IN_NUM = 799
AND FROM_DATE < DATE '1998-04-01'
AND TO_DATE > DATE '1998-04-01'

```

ตารางที่ 19 Sequence update

2.10.3 Nonsequence Modification

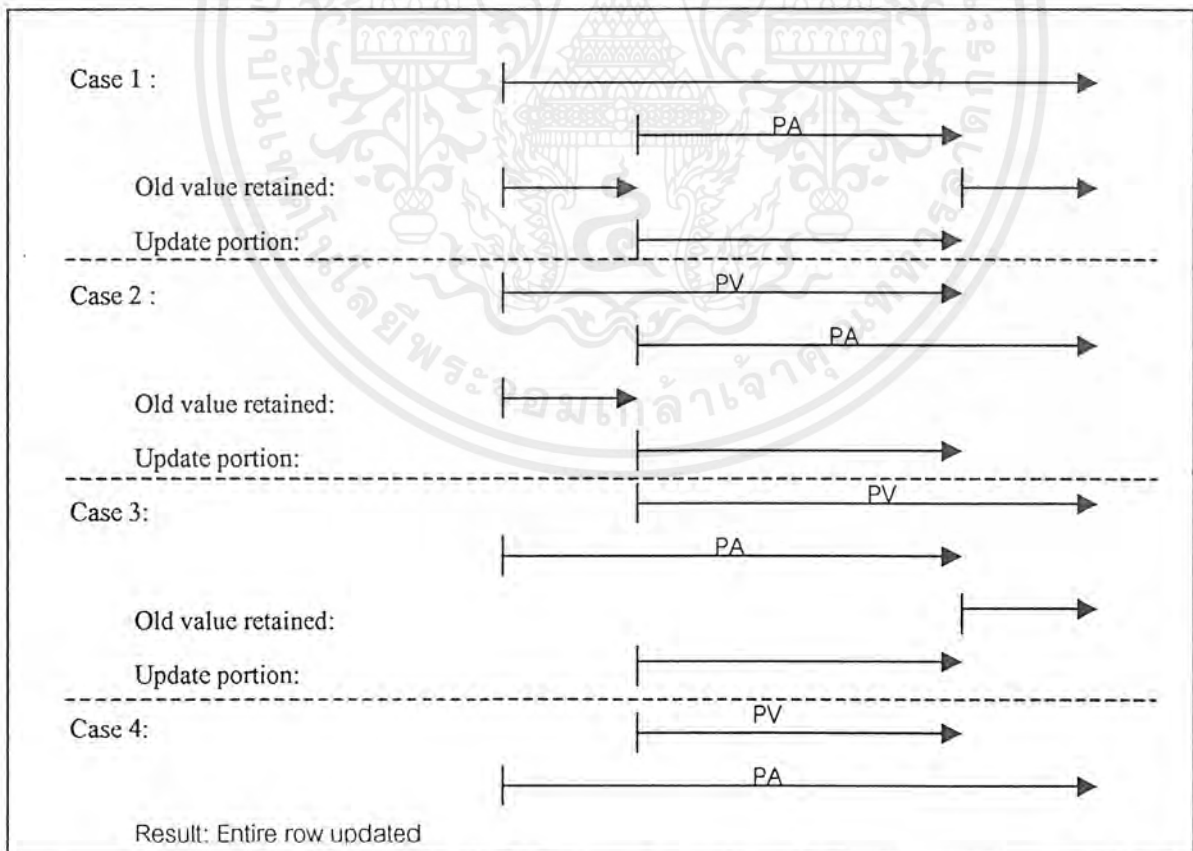
Nonsequence deletion จะกล่าวถึง period of validity ของแถวที่จะทำการลบ ตัวอย่างเช่น “ทำการลบ record ของ Lot 234 ซึ่งมีระยะเวลามากกว่า 3 เดือน”

```

DELETE FROM LOT
WHERE LOT_ID_NUM = 234
AND (TO_DATE-FROM_DATE MONTH)>INTERVAL '3' MONTH

```

Current และ Sequence delete จะกล่าวถึงอะไรที่เกิดขึ้นจริง ๆ เพราะว่า model ของมันเปลี่ยน Statement ของ nonsequence จะตระหนักเกี่ยวกับการนำเสนอเฉพาะอย่าง ในทางกลับกัน ความเกี่ยวข้อง กับ SQL Statement สำหรับ Current และ Sequence จะซับซ้อนมากกว่า nonsequence



รูปที่ 3 Sequence Update

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11 Transaction-Time State Tables

ในที่นี้จะใช้ส่วนหนึ่งของตาราง WDS เป็นตัวอย่าง ตาราง WDS ประกอบด้วย 21 columns แต่เราจะใช้เพียงบางส่วนเท่านั้น

RA จะแสดงถึงการเลื่อนขึ้นทางขวาและ DEC แสดงถึงการเลื่อนลง โดยตารางนี้จะมีการ update ใหม่เมื่อมีการค้นพบใหม่ เพื่อให้เกิดความถูกต้อง ในการติดตามการเปลี่ยนแปลงนี้เราจะทำการสร้างตารางใหม่ขึ้นมาชื่อ WDS_TT ซึ่งมี Column มากขึ้น 2 Column ได้แก่ Trans_Start และ Trans_Stop โดยเราเรียกตารางนี้ว่า Audit log แตกต่างจากรายเดิมซึ่งไม่มี Timestamp Trans_Start จะเป็นตัวบอกว่าเมื่อไหร่ที่แถวนั้นถูก insert เข้ามายังตารางเดิม หรือเมื่อแถวนั้นมีการ update Trans_Stop จะเป็นตัวบอกว่า เมื่อไหร่ที่แถวนั้นถูกลบออกไปจากรายเดิม หรือถูก update

ถ้าเวลาของ Trans_Stop เป็น forever (9999-12-31) ซึ่งให้เห็นว่าแถวนี้ยังอยู่ในตาราง WDS ถ้าปัจจุบัน WDS มี 4 แถว Trans_Stop ของตาราง WDS_TT จะต้องมีค่า forever 4 แถวด้วย

2.11.1 Maintaining The Audit Log

เราสามารถ maintain audit log ได้โดยอัตโนมัติโดยใช้การกำหนดบนตารางเดิม โดยไม่ต้องอาศัย Application อื่น ๆ เพื่อมา maintain ตาราง WDS การใช้วิธีนี้จะทำให้เกิดข้อดีคือจะทำให้ง่ายต่อการหา primary key ของ Audit Log

วิธีนี้จะทำให้เราแน่ใจว่า Audit Log สามารถจับการเปลี่ยนแปลงได้ทุกการเปลี่ยนแปลง เมื่อมีการ insert แถวเข้าไปยังตารางเดิมก็จะมี insert เข้าไปยัง Audit Log ด้วย โดย Trans_Start จะเป็น now (current_date) และ Trans_Stop จะมีค่าเริ่มต้นเป็น forever ถ้ามีการลบแถวใดออกจากตาราง Trans_Stop ของแถวนั้นจะเปลี่ยนค่าเป็น now ใน Audit Log

WDS_TT จะมีขนาดเพิ่มขึ้นเรื่อยๆ การ insert จะเพิ่มแถวเข้าไปในตาราง Audit Log การ delete ก็จะไปเปลี่ยนค่าของ Trans_Stop และการ update ก็จะทำทั้งเพิ่มแถวและเปลี่ยนค่า จะไม่มีแถวใดเลยที่ถูกลบออกจากตาราง WDS_TT

RA_Hour	RA_Min	RA_Sec	Dec_degree	Dec_minute	Discovers	Mag_First
00	00	08	75	30	"A1248"	10.5
05	57	40	00	02	"BU1190"	6.5
04	13	20	50	32	"CHAR15"	15.5
01	23	70	-09	55	"HJ3433"	10.5

ตารางที่ 20 ส่วนหนึ่งของตาราง WDS

RA_Hour	RA_Min	RA_Sec	Dec_degree	Dec_minute	Discover	Mag_first	Trans_start	Trans_stop
00	00	00	75	30	"A1248"	12.0	1989-03-12	1992-11-15
00	00	09	75	30	"A1248"	12.0	1992-11-15	1994-05-18
00	00	09	75	30	"A1248"	10.5	1994-05-18	1995-07-23
00	00	08	75	30	"A1248"	10.5	1995-07-23	9999-12-31
05	57	40	00	02	"BU1190"	6.5	1988-11-08	9999-12-31
04	13	20	50	32	"CHR15"	15.5	1990-02-09	9999-12-31
01	23	70	-09	55	"HU3433"	10.5	1991-03-25	9999-12-31
02	33	10	-09	25	"LDS3402"	10.6	1993-12-19	1996-07-09

ตารางที่ 21 Audit Log (WDS_TT) ของตาราง WDS

2.11.2 Querying The Audit Log

เราเคยอธิบายมาแล้วเกี่ยวกับการ Queries บน Valid time state table : current ,sequence ,nonsequence ซึ่งสามารถนำมาใช้กับ Transaction Time State Table ได้

ในการตัดสินใจสถานะปัจจุบันของ WDS เราสามารถดูได้โดยตรงหรือเอาข้อมูลมาจาก Audit Log

```
SELECT RA_HOUR,RA_MIN,RA_SEC,DEC_DEGREE,
DEC_MINUTE,DISCOVERER,MAG_FIRST
FROM WDS_TT
WHERE Trans_Stop=date'9999-12-31'
```

การใช้ประโยชน์จาก Audit Log เราสามารถสร้างตารางย่อยที่เราสนใจได้ อย่างเช่น เราต้องการจะดูว่ามีอะไรอยู่ในตารางก่อนวันที่ 1 เมษายน โดยสามารถสร้างได้ดังนี้

```
CREATE VIEW WDS_April_1 AS
SELECT RA_HOURS,RA_MIN,RA_SEC,DEC_DEGREE,DEC_MINUTE,
DISCOVERER,MAG_FIRST
FROM WDS_TT
WHERE Trans_Start <= DATE '1994-04-01,
AND DATE'1994-04-01' <Trans_Stop
```

เราสามารถที่จะกำหนด WDS_April_1 เป็นตารางแทนที่จะเป็น VIEW เหตุผลก็คือเราไม่ต้องการที่จะทำการเปลี่ยนแปลงใดๆ บนตาราง WDS อีก การตัดสินใจในการสร้าง WDS_April_1 เป็นการเพิ่มประสิทธิภาพของการ Queries และพื้นที่ของ Disk

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

sequence และ nonsequence queries สามารถทำได้บน Transaction Time State Table พิจารณา Query “เมื่อไรที่มีการบันทึกซึ่ง A1248 มี magnitude ไม่ใช่ 10.5” ในส่วนแรก “เมื่อไรที่มีการบันทึก” เป็นการบอกว่าต้องคิดเป็น Transaction Time และใช้ตาราง WDS_TT ความต้องการนี้ชี้ให้เห็นว่าเป็น sequence query

```
SELECT Mag_First,Trans_Start,Trans_Stop
FROM WDS_TT
WHERE DISCOVERER="A1248"
AND MAG_First <> 10.5
```

ในการ Queries คำถามที่ว่า “เมื่อไรที่มีการบันทึกดาวที่มี magnitude เท่ากับ A1248” ข้อความที่ว่า “เมื่อไรที่มีการบันทึก” เป็นการบ่งบอกว่า Query แบบ Transaction-Time ซึ่งเป็น sequence ในส่วนท้าย (ดาวที่มี magnitude เท่ากับ A1248) ชี้ให้เห็นว่าต้องใช้ Self-join สามารถแสดงเป็น SQL ได้ดังนี้

```
SELECT W2.Discoverer,
GREATEST(W1.Trans_Start,W2.Trans_Start),
LEAST(W1.Trans_Stop,W2.Trans_Stop)
FROM WDS_TT W1, WDS_TT W2
WHERE W1.Discoverer = "A1248"
AND W2.Discoverer <> W1.Discoverer
AND W1.Mag_First = W2.Mag_First
AND GREATEST(W1.Trans_Start,W2.Trans_Start) <
LEAST(W1.Trans_Stop,W2.Trans_Stop)
```

สำหรับ Nonsequence queries บนตาราง Transaction Time สามารถทำได้ โดยตัวอย่างเช่น คำถามว่า “เมื่อไรที่ตำแหน่ง RA_Sec ของ Double Star ถูกต้อง “ แสดงให้เห็นว่าเป็นการซ้อนทับกันของ Transaction time ของ 2 แถว โดยเกี่ยวข้องกับดาว 2 ดวงเดียวกันแต่มีค่า RA_Sec แตกต่างกัน แสดงเป็น SQL ได้เป็น

```
SELECT W1.Discoverer,
W1.RA_Sec AS Old_Value,
W2.RA_Sec AS New_Value,
W1.Trans_Stop AS When_Changed
FROM WDS_TT AS W1, WDS_TT AS W2
```

```

WHERE W1.Discoverer = W2.Discoverer
AND W1.Trans_Stop = W2.Trans_Start
AND W1.RA_Sec <> W2.RA_Sec

```

2.11.3 Modifying The Audit Log

แม้ว่า Queries บน Transaction time table สามารถที่จะทำเป็น current , sequence , nonsequence แต่การเปลี่ยนแปลงไม่สามารถทำได้ โดยความจริงแล้ว Audit Log นั้นจะเปลี่ยนแปลงจากผลกระทบของการเปลี่ยนแปลงตารางเดิมนั้น โดยการเปลี่ยนแปลงที่อนุญาตบน Transaction Time State Table คือ current modifications เท่านั้น

sequence และ nonsequence modifications สามารถเปลี่ยนแปลง State ก่อนหน้าของ Valid Time State Table ซึ่งมันจะขัดกับ semantic ของตาราง เพราะเราไม่สามารถเปลี่ยนแปลงอดีตได้ ด้วยเหตุผลนี้เราจึงห้ามเปลี่ยนแปลง Audit Log เอง ยกเว้นวิธีข้างต้น

Bitemporal Tables

Valid Time และ Transaction Time เป็นเรื่องที่ไม่เกี่ยวข้องกัน แต่ละอย่างเป็นอิสระจากกัน ตารางที่มีการแสดงทั้ง 2 อย่างพร้อมกันเรียกว่า Bitemporal table

Discoverer	Mag_First	Trans_Start	Trans_Stop	Valid_From	Valid_To
"A1248"	12.0	1989-03-12	1995-11-15	1922-05-14	9999-12-31
"A1248"	12.0	1995-11-15	9999-12-31	1922-05-14	1994-10-16
"A1248"	10.5	1995-11-15	9999-12-31	1994-10-16	9999-12-31

ตารางที่ 22 Bitemporal Table (WDS_B)

Bitemporal จะมี Timestamp ที่เป็น Valid time 2 column แสดงช่วงเวลาที่เป็นจริง และมี Timestamp ที่เป็น Transaction time 2 column แสดงเวลาที่ fact นั้นถูกเก็บอยู่ในตาราง โดยการจัดการและเปลี่ยนแปลงสามารถระบุ valid time เพื่อ query แบบ sequence, nonsequence, current อย่างไรก็ตาม Transaction time จะทำได้ที่ current เท่านั้น ส่วนการ Query สามารถทำได้ทั้ง current, sequence และ nonsequence ซึ่งทั้ง 2 อย่างสามารถ Query รวมกันได้ ตัวอย่างเช่น “จงแสดงข้อมูลที่เก็บอยู่ในฐานข้อมูลในปี 1920” แสดงเป็น SQL ได้ดังนี้

```

SELECT B1.Discoverer, B1.Trans_Stop AS WHEN_Changed,
GREATEST (B1.Valid_Form, B2.Valid_Form) AS Valid_Form,
LEAST(B1.Valid_To, B2.Valid_To) AS Valid_To
FROM WDS_B B1, WDS_B B2
WHERE B1.Discoverer = B2.Discoverer

```

```

AND B1.Trans_Stop = B2.Trans_Start
AND GREATEST(B1.Valid_From, B2.Valid_From) < Date'1929-12-31'
AND DATE'1920-01-01' < LEAST(B1.Valid_To, B2.Valid_To)
AND GREATEST(B1.Valid_From, B2.Valid_From) < LEAST(B1.Valid_To,
B2.Valid_to)

```

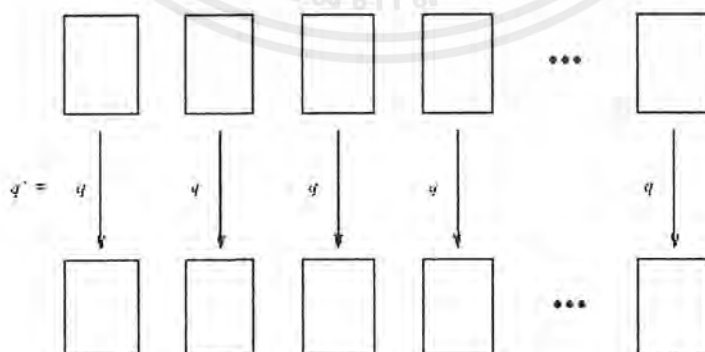
2.12 การสนับสนุน Temporal ในภาษา SQL มาตรฐาน

ในหัวข้อนี้จะอธิบายถึง ภาษา SQL ที่สามารถนำมาจัดการกับ ข้อมูลที่มีเวลาที่เกี่ยวข้อง ถึงแม้ว่า SQL-92 นั้นจะสนับสนุนชนิดของข้อมูลที่เกี่ยวข้องกับเวลาคือ DATE, TIME และ TIMESTAMP แต่ จริงๆ แล้ว SQL-92 ไม่มีความเข้าใจในเกี่ยวกับ ตารางที่มีเวลาแปรผัน จริงๆแล้ว SQL ไม่มีความคิดในหลักการ ของ Current หรือ Sequence query modifications หรือ views รวมถึงไม่รู้ความแตกต่างของ valid time และ transaction time อีกด้วย

โชคดีที่ คณะกรรมการมาตรฐานได้พิจารณาถึงข้อจำกัดดังกล่าวจึงได้มีการสนับสนุน temporal ใน ภาษา SQL-3 และ ผู้ผลิตก็ได้รวมมาตรฐานดังกล่าวเข้าไว้ด้วยในผลิตภัณฑ์ของตัวเอง โดยที่ SQL-3 ได้เพิ่มชนิดของข้อมูลเข้าไปอีกได้แก่ PERIOD(DATE) สำหรับ วัน PERIOD(TIME) สำหรับ วินาที และ PERIOD(TIMESTAMP) สำหรับ มิลลิวินาที SQL-3 นั้น สนับสนุนทั้ง Valid-time, Transaction-time หรืออาจจะทั้ง 2 แบบพร้อมๆกัน(bitemporal)

นอกจากนั้นยังได้มีการเพิ่มทางเลือกทางด้านเวลาเข้าไปอีก 2 คำก็คือ SEQUENCED และ NONSEQUENCED สามารถใช้ร่วมกับ VALIDTIME และ TRANSACTIONTIME ทำให้แบ่งได้เป็น 4 กรณี

2.12.1 VALIDTIME SELECT ใช้กับตารางที่สนับสนุน VALID-TIME ซึ่งจะให้ผลของการ query เป็นแบบ SEQUENCED VALIDTIME SELECT ซึ่งหมายความว่าผลลัพธ์ ของการ query จะได้มาจากแต่ละสถานะของตารางดังรูป ที่ 4

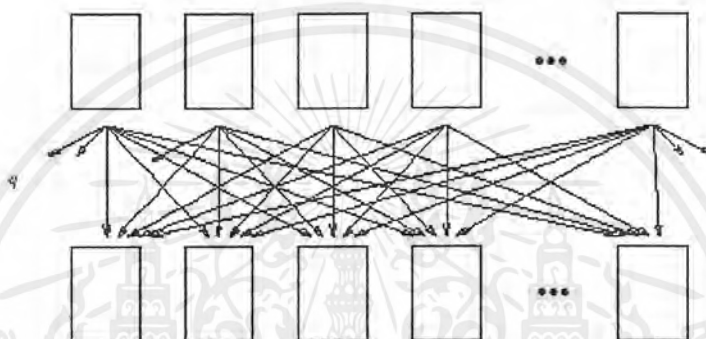


รูปที่ 4 แสดงผลลัพธ์ของการ query แบบ sequenced validtime

จากรูปจะเห็นได้ว่าผลลัพธ์ของการ query ก็คือ q' นั้นจะได้มาจากแต่ละสถานะ(state) ของตาราง input ที่สนับสนุน temporal โดยที่ตาราง input นั้นได้สร้างตาราง output สำหรับในแต่ละสถานะของตาราง เช่น คำถามที่ถามถึงประวัติของเงินเดือนของพนักงาน

2.12.2 VALIDTIME <period exp> SELECT เหมือนกับ VALIDTIME SELECT แต่จะคืนค่า timestamp ภายในระยะเวลา ที่กำหนดใน period เช่น VALIDTIME PERIOD '[1995-01-01 – 1995-12-31] SELECT)

2.12.3 NONSEQUENCED VALIDTIME SELECT มีความเป็นไปได้ว่าผลลัพธ์ของการ query ต้องการสถานะ(state)อื่นๆเพื่อมาประกอบ query ด้วยดังรูป



รูปที่ 5 แสดงผลลัพธ์จากการ query แบบ NONSEQUENCED VALIDTIME

จากรูปจะเห็นได้ว่า q' ซึ่งเป็นผลของการ query นั้น ได้มาจากหลายๆสถานะ(state) ของตาราง input เช่น คำถามที่ว่า “ใครมีเงินเดือนเพิ่มขึ้นบ้าง”

2.15.4 NONSEQUENCED VALIDTIME <period exp> SELECT จะเหมือนกับ NONSEQUENCED VALIDTIME SELECT แต่จะใช้ period เป็น timestamp

2.13 ทฤษฎีการแปลงจาก SQL/Temporal เป็น SQL มาตรฐาน

เนื่องจากฐานข้อมูลที่สัมพันธ์กับเวลา (Temporal Database) มีการเก็บข้อมูลที่เป็นเวลา ซึ่งเวลานี้แบ่งออกได้เป็น 3 ประเภท ได้แก่ Valid time (ช่วงเวลาที่ข้อมูลนั้นเป็นจริง), Transaction Time (ช่วงเวลาที่ข้อมูลนั้นเก็บอยู่ในฐานข้อมูล), User Define time (เวลาที่ไมต้องการแปลความหมายในทาง Temporal) ดังนั้นจึงมี SQL มาตรฐานใหม่ เรียกว่า SQL Temporal เพื่อมาจัดการการ Query ข้อมูลให้สะดวกกว่าการใช้ SQL มาตรฐาน

ซึ่งสามารถทำการ Map จาก SQL Temporal เป็น SQL มาตรฐาน ได้โดยแบ่งเป็นกรณี ดังนี้

2.13.1 Current Validtime มีรูปแบบการ Map ดังนี้

SQL/Temporal	SQL
CURRENTVALID SELECT <COLUMN_NAME> FROM TABLE1 WHERE <EXPRESSION>	SELECT <COLUMN_NAME> FROM TABLE1 WHERE <EXPRESSION> AND TO_DATE = DATE '9999-12-31'

ตัวอย่างของคำถาม Query ประเภทนี้ได้แก่คำถามที่ต้องการความจริงในปัจจุบัน ตัวอย่างเช่น คำถามที่ว่า “มีวัวจำนวนเท่าใดจาก lot 219 ใน feed yard 1 ที่ปัจจุบันอาศัยอยู่ในแต่ละ pen” แสดงการ Map ได้ดังนี้

SQL/Temporal	SQL
CURRENTVALID SELECT PEN_ID, HD_CNT FROM LOT_LOC WHERE FDYD_ID = 1 AND LOT_ID_NUM = 29	SELECT PEN_ID, HD_CNT FROM LOT_LOC WHERE FDYD_ID = 1 AND LOT_ID_NUM = 219 AND TO_DATE = DATE '9999-12-31'

2.13.2 Sequence Validtime มีรูปแบบการ Map ดังนี้

SQL/Temporal	SQL
VALIDTIME SELECT <COLUMN_NAME> FROM TABLE1 WHERE <EXPRESSION>	SELECT <COLUMN_NAME>, <FROM_DATE, TO_DATE> FROM TABLE1 WHERE <EXPRESSION>

ตัวอย่างของคำถามประเภทนี้ได้แก่ “จง list ประวัติของจำนวนวัวจาก lot 219 ใน yard 1 ที่อยู่ในแต่ละ pen” ซึ่งแสดงการ Map ได้ดังนี้

SQL/Temporal	SQL
VALIDTIME SELECT PEN_ID, HD_CNT FROM LOT_LOC WHERE FDYD = 1 AND LOT_ID_NUM = 219	SELECT PEN_ID, HD_CNT, FROM_DATE, TO_DATE FROM LOT_LOC WHERE FD_YD=1 AND LOT_ID_NUM = 219

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13.3 Nonsequence Validtime มีวิธีการ Map ดังนี้

SQL/Temporal	SQL
NONSEQUENCEVALIDTIME SELECT <COLUMN_NAME> FROM TABLE1 WHERE <EXPRESSION>	SELECT <COLUMN_NAME> FROM TABLE1 WHERE <EXPRESSION>

ตัวอย่างของคำถามประเภทนี้ได้แก่ “มีวัวกี่ตัวจาก lot 219 ใน yard 1 ในบางเวลาที่ อยู่ในแต่ละ pen” ซึ่งคำถามนี้ไม่สนใจว่าข้อมูลนั้นเป็นจริงเมื่อใด แสดงการ Map ได้ดังนี้

SQL/Temporal	SQL
NONSEQUENCEVALIDTIME SELECT PEN_ID, HD_CNT FROM LOT_LOC WHERE FDYD_ID = 1 AND LOT_ID_NUM = 219	Select PEN_ID, HD_CNT From LOT_LOC Where FDYD_ID = 1 and LOT_ID_NUM = 219

2.13.4 Current Transaction time มีวิธีการ Map ดังนี้

SQL/Temporal	SQL
CURRENTTRANSACTION SELECT <COLUMN_NAME> FROM TABLE1 WHERE <EXPRESSION>	SELECT <COLUMN_NAME> FROM TABLE1 WHERE <EXPRESSION> AND TRANS_STOP = DATE '9999-12-31'

ตัวอย่างของคำถามประเภทนี้ได้แก่ “ดาวดวงไหนที่มีความสว่างเท่ากับ 11 หรือสว่างกว่าเท่าที่รู้ในปัจจุบัน” (จากตารางของดวงดาว)

SQL/Temporal	SQL
CURRENTTRANSACTION SELECT Discoverer FROM WDS WHERE Mag_First <= 11.0	SELECT Discoverer FROM WDS WHERE Mag_First <= 11.0 AND TRANS_STOP = DATE '9999-12-31'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13.5 Sequence Transaction time มีรูปแบบการ Map ดังนี้

SQL/Temporal	SQL
TRANSACTIONTIME SELECT <COLUMN_NAME> FROM TABLE1 WHERE <EXPRESSION>	SELECT <COLUMN_NAME>, <TRANS_START,TRANS_STOP> FROM TABLE1 WHERE <EXPRESSION>

ตัวอย่างของคำถามประเภทนี้ได้แก่ “เมื่อใดที่ A1248 ถูกบันทึกว่ามีความสว่างแตกต่างไปจาก 10.5” ซึ่งสามารถแสดงการ Map ได้ดังนี้

SQL/Temporal	SQL
TRANSACTIONTIME SELECT Mag_First FROM WDS_TT WHERE Mag_First <> 10.5 AND Discover = 'A1248'	SELECT Mag_First,TRANS_START, TRANS_STOP FROM WDS_TT WHERE Mag_First <> 10.5 AND Discover = 'A1248'

2.13.6 Nonsequenced Transaction มีวิธีการ Map ดังนี้

SQL/Temporal	SQL
NONSEQUENCETRANSACTIONTIME SELECT <COLUMN_NAME> FROM TABLE1 WHERE <EXPRESSION>	SELECT <COLUMN_NAME> FROM TABLE1 WHERE <EXPRESSION>

ตัวอย่างของคำถามประเภทนี้ได้แก่ “เมื่อใดที่ตำแหน่ง RA_Sec ของดาวนั้นเป็นข้อมูลที่ถูกต้อง” แสดงวิธีการ Map ได้ดังนี้

SQL/Temporal	SQL
NONSEQUENCETRANSACTIONTIME SELECT W1.Discoverer , W1.RA_Sec AS Old_Value, W 2.RA_Sec AS New_Value, END(TRANSACTIONTIME(W1)) AS When_Changed	SELECT W1.Discoverer , W1.RA_Sec AS Old_Value, W 2.RA_Sec AS New_Value, END(TRANSACTIONTIME(W1)) AS When_Changed

AS When_Changed FROM WDS_TT AS W1, WDS_TT AS W2 WHERE W1.Discoverer = W2.Discoverer AND W1.RA_Sec <> W2.RA_Sec	FROM WDS_TT AS W1, WDS_TT AS W2 WHERE W1.Discoverer = W2.Discoverer AND W1.RA_Sec <> W2.RA_Sec
---	--

2.14 การ Join

2.14.1 Current Validtime มีวิธีการ map ดังนี้

SQL/Temporal	SQL
CURRENTVALID SELECT <COLUMN_NAME> FROM TABLE1, TABLE2 WHERE <EXPRESSION>	SELECT <COLUMN_NAME> FROM TABLE1, TABLE2 WHERE <EXPRESSION> AND TABLE1.TO_DATE=DATE '9999-12-31' AND TABLE2.TO_DATE=DATE '9999-12-31'

ตัวอย่างของคำถามประเภทนี้ได้แก่ “มี LOT ไหนที่อาศัยอยู่ในคอกเดียวกันในปัจจุบันบ้าง”
ซึ่งสามารถแสดงการ Map ได้ดังนี้

SQL/Temporal	SQL
CURRENTVALID SELECT L1.Lot_ID_NUM, L2.LOT_ID_NUM, L1.PEN_ID FROM LOT_LOC AS L1, LOT_LOC AS L2 WHERE L1.LOT_ID_NUM < L2.LOT_ID_NUM AND L1.FDYD_ID = L2.FDYD_ID AND L1.PEN_ID = L2.PEN_ID	SELECT L1.Lot_ID_NUM, L2.LOT_ID_NUM, L1.PEN_ID FROM LOT_LOC AS L1, LOT_LOC AS L2 WHERE L1.LOT_ID_NUM < L2.LOT_ID_NUM AND L1.FDYD_ID = L2.FDYD_ID AND L1.PEN_ID = L2.PEN_ID AND L1.TO_DATE = DATE '9999-12-31' AND L2.TO_DATE = DATE '9999-12-31'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.14.2 Sequenced Validtime มีวิธีการ Map ดังนี้

SQL/Temporal	SQL
VALIDTIME SELECT <COLUMN_NAME> FROM TABLE1 T1, TABLE2 T2 WHERE <EXPRESSION>	SELECT <COLUMN_NAME>, T1.FROM_DATE, T2.TO_DATE FROM TABLE1 T1, TABLE2 T2 WHERE <EXPRESSION> T2.FROM_DATE <= T1.FROM_DATE AND T1.TO_DATE <= T2.TO_DATE UNION SELECT <COLUMN_NAME>, T1.FROM_DATE, T2.TO_DATE FROM TABLE1 T1, TABLE2 T2 WHERE <EXPRESSION> T1.FROM_DATE > T2.FROM_DATE AND T2.TO_DATE < T1.TO_DATE AND T1.FROM_DATE < T2.TO_DATE UNION SELECT <COLUMN_NAME>, T2.FROM_DATE, T1.TO_DATE FROM TABLE1 T1, TABLE2 T2 WHERE <EXPRESSION> T2.FROM_DATE > T1.FROM_DATE AND T1.TO_DATE < T2.TO_DATE AND T2.FROM_DATE < T1.TO_DATE UNION SELECT <COLUMN_NAME>, T2.FROM_DATE, T2.TO_DATE FROM TABLE1 T1, TABLE2 T2 WHERE <EXPRESSION> T2.FROM_DATE >= T1.FROM_DATE AND T2.TO_DATE <= T2.TO_DATE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างของคำถามประเภทนี้ได้แก่ “จงแสดงประวัติของการอาศัยร่วมกันในแต่ละคอก”
ซึ่งสามารถเขียนเป็น SQL Temporal ได้ดังนี้

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

ซึ่งสามารถ Map เป็น SQL ได้ดังนี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การนำรหัสเฉพาะ(Object Identification: OID) มาประยุกต์ใช้กับข้อมูลที่เปลี่ยนแปลงตามเวลา

เนื่องจากฐานข้อมูลแบบเดิมเป็น Relational Database ซึ่งมี Primary key เป็น External Identify เมื่อมีการเปลี่ยนแปลง key ทำให้ Identify เปลี่ยนแปลงได้ แต่สำหรับฐานข้อมูล Informix ซึ่งเป็น Object Relational สามารถใช้ Internal Identify เป็น key เมื่อมีการเปลี่ยนแปลงข้อมูลใดๆ ไม่มีผลต่อ Identify ทำให้สืบค้นข้อมูลได้ง่ายกว่าวิธีเดิม

โดย ชนิดของข้อมูลของ Informix ที่เป็น Internal Identify ได้แก่ ข้อมูลชนิด Serial(n) ซึ่งมี Column ที่เป็น Serial มีคุณลักษณะดังต่อไปนี้

1. เป็นตัวเลขจำนวนเต็มบวก
2. ไม่ซ้ำซ้อน (UNIQUE)
3. ป้อนค่าให้เองโดยอัตโนมัติ ซึ่งเรียงลำดับจากน้อยไปมาก

ข้อจำกัดของ Column ที่เป็น Serial มีดังต่อไปนี้

1. ในตารางหนึ่ง ๆ สามารถมี Column ที่เป็น Serial ได้เพียง Column เดียว
2. ชนิดของข้อมูลที่เป็น Serial ไม่ได้เป็น column ที่ไม่ซ้ำ (Unique column) โดยอัตโนมัติ ต้องทำการกำหนด unique index ไปยัง column นั้น เพื่อป้องกันการซ้ำกันของ Serial number ถ้าสร้างตารางโดยใช้ Interactive Schema Editor (เช่น SQL Editor ของ Informix) จะมีการกำหนด unique index ไปยัง Serial column นั้น โดยอัตโนมัติ

3.1 การกำหนดค่าเริ่มต้นให้กับชนิดของข้อมูล Serial

ค่าเริ่มต้นของข้อมูลชนิด Serial มีค่าเท่ากับ 1 ซึ่งสามารถที่จะกำหนดค่าเริ่มต้นให้เป็นค่าอื่นได้ แต่ถ้าหากกำหนดค่าเริ่มต้นเป็น 0 แล้ว Database Server จะกำหนดค่าเริ่มต้นของ Column ให้เป็น 1 หลังจากนั้นจะเพิ่มค่าขึ้นเรื่อย ๆ ทีละ 1 เมื่อมีการ insert โดยค่าสูงสุดของ ข้อมูลชนิด Serial มีค่าเท่ากับ 2,147,483,647 ถ้ามากกว่านี้ จะเกิดความผิดพลาด (Syntax Error)

หากมีการกำหนดค่าให้กับ Serial Column (โดยใช้การ Insert) โดยที่ค่านั้นไม่ซ้ำกับค่าเดิมที่อยู่ใน Serial Column ของตาราง Database Server จะกำหนดค่าถัดไปในลำดับให้กับ Serial Column ของแถวนั้น

3.2 การใช้ Serial กับ Integer

Database Server จะมองว่าข้อมูลชนิด Serial เป็นข้อมูล Integer ชนิดพิเศษ ดังนั้นการกระทำทางคณิตศาสตร์ที่ข้อมูลชนิด Integer สามารถทำได้ (+, -, *, /) ก็จะสามารถทำได้ใน ข้อมูลชนิด Serial เช่นเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ข้อมูลชนิด Serial8(n)

ถ้าหากค่าใน Serial Column มีค่ามากกว่า 2,147,483,647 หากใช้ข้อมูลชนิด Serial จะเกิดความผิดพลาด (Syntax Error) ซึ่งจะต้องใช้ข้อมูลชนิด Serial8 โดยจะมีการเก็บเป็นข้อมูลขนาด 8 ไบต์ มีค่าสูงสุดเท่ากับ $2^{63}-1$ หรือเท่ากับ 9,223,372,036,854,775,807 โดยมีข้อจำกัดเหมือนกับข้อมูลชนิด Serial

3.4 ตัวอย่างการสร้างข้อมูลชนิด Serial

สามารถสร้างข้อมูลชนิด Serial จากตัวอย่างการสร้างตาราง ได้ดังนี้

Create table Student	Create table Student_TT
(StdID Char(8),	(StdID Char(8),
TPren Char(20),	TPren Char(20),
EPren Char(20),	EPren Char(20),
TName Char(45),	TName Char(45),
Ename Char(45),	Ename Char(45),
Degree Char(3),	Degree Char(3),
StdType Char(3),	StdType Char(3),
→ Std_serial serial,	→ std_int int,
PRIMARY KEY (StdID)	Valid_From date,
);	Valid_To date,
	TxStart date,
	TxStop date,
	PRIMARY KEY
	(StdID,Valid_From,Valid_To,TxStart,TxStop)
);

สำหรับตาราง student เป็นตารางที่จะเก็บ std_serial ซึ่งจะมีค่าไม่ซ้ำกัน แต่ในตาราง student_tt เป็นตารางที่เก็บค่า std_int ซึ่งสามารถมีค่าซ้ำกันได้ ซึ่งสองตารางนี้จะทำงานสัมพันธ์กันตลอดเวลา ดังที่จะกล่าวในบทต่อไป

3.5 ประโยชน์ของ OID (Object Identification)

ในตาราง audit log นั้น เราไม่สามารถแยกความแตกต่างระหว่างการแก้ไขข้อมูลเดิมกับการเพิ่มข้อมูลใหม่เข้าไป หรืออาจจะกล่าวได้อีกนัยหนึ่งว่า เราไม่สามารถทราบได้ว่าข้อมูลในแถวใหม่ที่ถูกเพิ่มเข้าไปในตาราง

นั้นเกิดจากการแก้ไขข้อมูลเดิม หรือว่าเป็นการเพิ่มข้อมูลแถวใหม่เข้าไปจริงๆ

Stdid	Tpren	Epre	Tname	Ename	Degree	StdType
39014499	นาย	Mr.	วุฒิชัย	Wuttichai	B	B
39014508	นาย	Mr.	วุฒิชัย	Wuttichai	B	B

จากตารางข้างบนจะเห็นได้ว่า เราไม่สามารถบอกได้เลยว่า แถวที่สองนั้นเกิดจากการเปลี่ยนแปลงข้อมูลในแถวแรกตรงส่วนของรหัสนักศึกษา (Stdid) หรือว่ามีกรเพิ่มนักศึกษาคณใหม่เข้าไป

Stdid	Tpren	Epre	Tname	Ename	Degree	StdType	Std_int
39014499	นาย	Mr.	วุฒิชัย	Wuttichai	B	B	1
39014508	นาย	Mr.	วุฒิชัย	Wuttichai	M	A	1

ต่อเมื่อเราเพิ่ม Std_int เข้าไปซึ่งเป็น OID (Object Identification) เข้าไปทำให้เราทราบได้ว่า แถวแรกและแถวที่สองนั้นจริงๆแล้วเป็นคนๆเดียวกัน แต่มีการเปลี่ยนแปลงรหัสนักศึกษา (Stdid) จาก 39014499 ไปเป็น 39014508

ดังนั้นจะเห็นได้ว่าการที่เรานำรหัสเฉพาะมาใช้กับการจัดเก็บข้อมูลแบบ Temporal จะทำให้เราสามารถติดตามการเปลี่ยนแปลงของตารางได้ดีขึ้น

บทที่ 4

การออกแบบโปรแกรมระบบงานลงทะเบียน

4.1 การออกแบบ ERT (Entity Relationship Model) สำหรับระบบงานลงทะเบียนนักศึกษา

ในขั้นตอนการออกแบบนั้นได้ใช้ โมเดลในการออกแบบฐานข้อมูลที่มีชื่อว่า ERT model สำหรับระบบการลงทะเบียนดังรูปที่ 6 ซึ่งมีส่วนที่เป็น Timestamp (Valid time หรือ Transaction time) ติดอยู่กับส่วน Entity โดยมี 3 Entities คือ Registration มีความสัมพันธ์กับ Entity และ Entity ซึ่งสามารถ normalize ได้ตารางดังนี้

ตาราง student

ลำดับที่	ชื่อคอลัมน์	ประเภท	ขนาด	คำอธิบาย	หมายเหตุ
1.	StdID	Char	8	รหัสนักศึกษา	P.K.
2.	TPren	Char	20	คำนำหน้าชื่อภาษาไทย	
3.	EPren	Char	20	คำนำหน้าชื่อภาษาอังกฤษ	
4.	TName	Char	45	ชื่อภาษาไทย	
5.	Ename	Char	45	ชื่อภาษาอังกฤษ	
6.	Degree	Char	1	ระดับ	M – Master D – Doctor
7.	StdType	Char	1	ประเภทนักศึกษา	A – สามัญ B – ทดลองเรียน
8.	Std_serial	Serial	4	Serial ของนักศึกษา	

ตาราง student_tt

ลำดับที่	ชื่อคอลัมน์	ประเภท	ขนาด	คำอธิบาย	หมายเหตุ
1.	StdID	Char	8	รหัสนักศึกษา	P.K.
2.	TPren	Char	20	คำนำหน้าชื่อภาษาไทย	
3.	EPren	Char	20	คำนำหน้าชื่อภาษาอังกฤษ	
4.	TName	Char	45	ชื่อภาษาไทย	
5.	Ename	Char	45	ชื่อภาษาอังกฤษ	
6.	Degree	Char	1	ระดับ	M – Master D – Doctor
7.	StdType	Char	1	ประเภทนักศึกษา	A – สามัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

					B - ทดลองเรียน
8.	Std_int	int	4	Serial ของนักศึกษา	
9.	Valid_from	Date	-		P.K.
10.	Valid_to	Date	-		P.K.
11.	Txstart	Date	-		P.K.
12.	Txstop	Date	-		P.K.

ตาราง Subject

ลำดับที่	ชื่อคอลัมน์	ประเภท	ขนาด	คำอธิบาย	หมายเหตุ
1.	SubId	Char	8	รหัสวิชา	P.K.
2.	EName	Char	75	ชื่อวิชาภาษาอังกฤษ	
3.	Tname	Char	75	ชื่อวิชาภาษาไทย	
4.	Presub1	Char	8	วิชาบังคับก่อน1	
5.	Presub2	Char	8	วิชาบังคับก่อน2	
6.	Credit	Number	2	หน่วยกิตรวม	
7.	Type	Char	1	ประเภทวิชา	A - บรรยาย B - ปฏิบัติ
8.	Sub_serial	Serial	4	Serial ของวิชา	

ตาราง Subject_tt

ลำดับที่	ชื่อคอลัมน์	ประเภท	ขนาด	คำอธิบาย	หมายเหตุ
1.	SubId	Char	8	รหัสวิชา	P.K.
2.	EName	Char	75	ชื่อวิชาภาษาอังกฤษ	
3.	Tname	Char	75	ชื่อวิชาภาษาไทย	
4.	Presub1	Char	8	วิชาบังคับก่อน1	
5.	Presub2	Char	8	วิชาบังคับก่อน2	
6.	Credit	Number	2	หน่วยกิตรวม	
7.	Type	Char	1	ประเภทวิชา	A - บรรยาย B - ปฏิบัติ
8.	Sub_int	Int	4	Serial ของวิชา	
9.	Valid_form	Date	-		P.K.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10.	Valid_to	Date	-		P.K.
11.	Txstart	Date	-		P.K.
12.	Txstop	Date	-		P.K.

ตาราง Registration

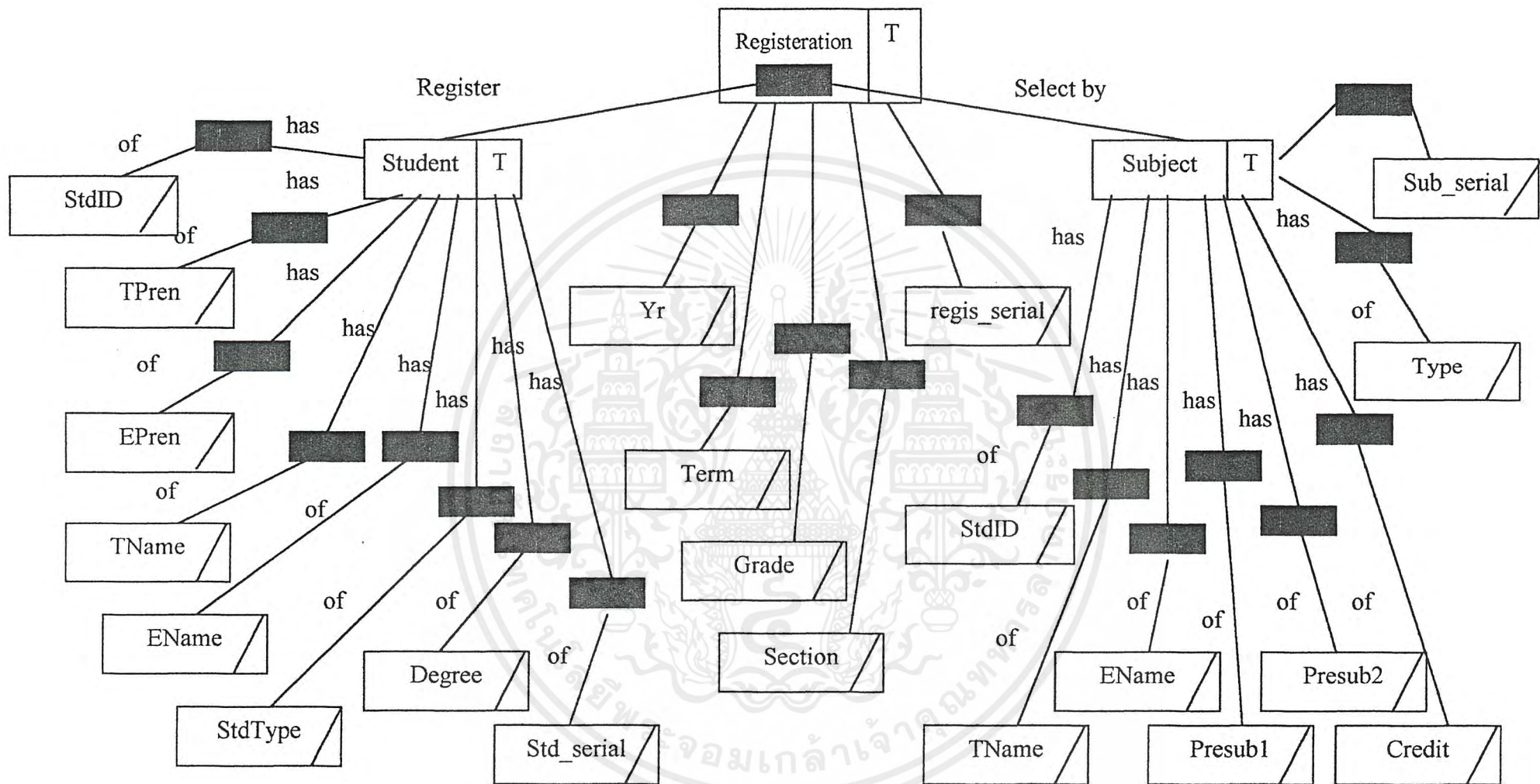
ลำดับที่	ชื่อคอลัมน์	ประเภท	ขนาด	คำอธิบาย	หมายเหตุ
1.	StdId	Char	8	รหัสนักศึกษา	P.K.
2.	SubId	Char	8	รหัสวิชา	P.K.
3.	Yr	Char	4	ปีการศึกษา	P.K.
4.	Term	Char	1	เทอมการศึกษา	P.K.
5.	Grade	Char	2	เกรด	
6.	Section	Char	2	เซ็กชัน	
7.	Regis_serial	Serial	4	Serial ของการลงทะเบียน	

ตาราง Registration_tt

ลำดับที่	ชื่อคอลัมน์	ประเภท	ขนาด	คำอธิบาย	หมายเหตุ
1.	StdId	Char	8	รหัสนักศึกษา	P.K.
2.	SubId	Char	8	รหัสวิชา	P.K.
3.	Yr	Char	4	ปีการศึกษา	P.K.
4.	Term	Char	1	เทอมการศึกษา	P.K.
5.	Grade	Char	2	เกรด	
6.	Section	Char	2	เซ็กชัน	
7.	Regis_int	Int	4	Serial ของการลงทะเบียน	
8.	Valid_form	Date	-		P.K.
9.	Valid_to	Date	-		P.K.
10.	Txstart	Date	-		P.K.
11.	Txstop	Date	-		P.K.

4.2 รูปแสดงERT model สำหรับระบบงานลงทะเบียนนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6 ERT (Entity Relationship Time) Model ของระบบทะเบียนนักศึกษา

บทที่ 5

โปรแกรมการตรวจสอบและสอบถามการลงทะเบียนของนักศึกษา

สำหรับตัวโปรแกรม สามารถแบ่งออกได้เป็น 3 โปรแกรมย่อย ได้แก่

1. โปรแกรมจัดการข้อมูลในฐานข้อมูล (Manage Data Application)
2. โปรแกรมค้นหาข้อมูลโดย SQL และตัวสร้างภาษา SQL (SQL Editor Include SQL Builder)
3. โปรแกรมค้นหาข้อมูลจากคำถามพื้นฐาน

จากการออกแบบฐานข้อมูลในบทที่ผ่านมา มีตารางที่ใช้ในการเก็บข้อมูล 3 ตารางและแต่ละตารางจะมี Audit log ซึ่งเป็นตารางที่เก็บการเปลี่ยนแปลงของข้อมูล โดยแต่ละตารางมีคอตัมส์ ดังนี้

Student

Stdid	TPren	EPren	Tname	EName	Degree	StdType	Std_serial
-------	-------	-------	-------	-------	--------	---------	------------

Student_TT

Stdid	TPren	EPren	Tname	EName	Degree	StdType	Std_int
-------	-------	-------	-------	-------	--------	---------	---------

Valid_from	Valid_to	TxStart	TxStop
------------	----------	---------	--------

Subject

Subid	Ename	PName	Presub1	Presub2	Credit	Type	Sub_serial
-------	-------	-------	---------	---------	--------	------	------------

Subject_TT

Subid	Ename	PName	Presub1	Presub2	Credit	Type	Sub_int
-------	-------	-------	---------	---------	--------	------	---------

Valid_from	Valid_to	TxStart	TxStop
------------	----------	---------	--------

Registration

Stdid	Subid	Yr	Term	Grade	Section	Regis_serial
-------	-------	----	------	-------	---------	--------------

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Registration_TT

Std_int	Sub_int	Yr	Term	Grade	Section	Regis_serial
---------	---------	----	------	-------	---------	--------------

Valid_from	Valid_to	TxStart	TxStop
------------	----------	---------	--------

การที่ต้องมี 6 ตารางก็เพราะว่า จะมีตารางที่คอยเก็บสถานะทั้งหมดของตารางและจะมีตารางที่คอยเก็บสถานะล่าสุดของตาราง ซึ่งทั้งสองตารางนี้จะทำงานประสานกันตลอดเวลาซึ่งก็จะกล่าวต่อไป

5.1 สภาพแวดล้อมของระบบ

5.1.1 โปรแกรมที่ใช้ในการติดตั้งเพื่อทำงานกับฐานข้อมูล Informix

Server

- Informix Universal Server Version 9.14

Client

- Database Administrator
- Informix Connect
- Informix Client Software Developer kits

โดยที่เครื่อง Client จะมีความต้องการของระบบ (System Requirements) ดังนี้

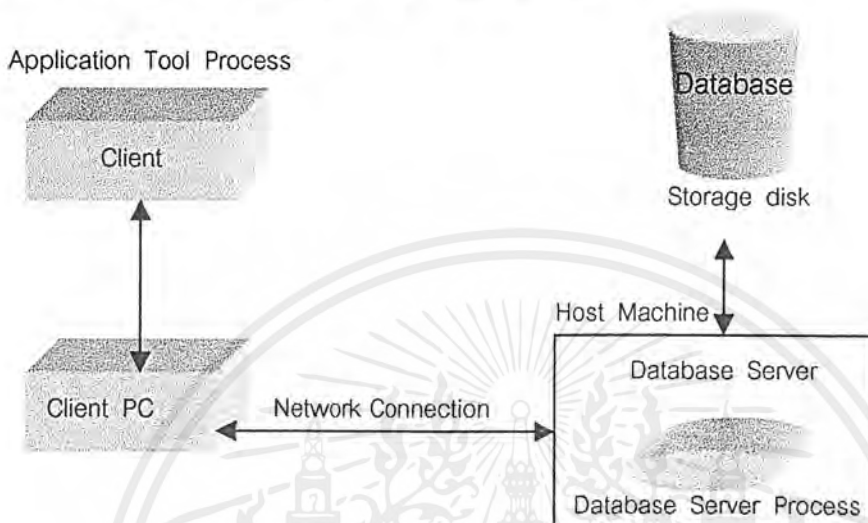
Requirement	Windows 95	Windows NT
Hardware	ไมโครโปรเซสเซอร์ 80486 หรือสูงกว่า	ไมโครโปรเซสเซอร์ 80486 หรือสูงกว่า
Windows Version	Windows 95	Version 4.0
RAM	12 MB	16 MB
Hard-disk	20 MB (หรือน้อยกว่านี้ ขึ้นอยู่กับ Component ที่เลือกทำการติดตั้ง)	20 MB (หรือน้อยกว่านี้ ขึ้นอยู่กับ Component ที่เลือกทำการติดตั้ง)

ตารางที่ 23 ความต้องการของระบบ (System Requirement)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2 สถาปัตยกรรมของการเชื่อมต่อ Client และ Server

Client ของ Informix จะใช้สถาปัตยกรรมแบบ 2 กระบวนการ (Two-process Architecture) ซึ่งเหมาะสมสำหรับการเชื่อมต่อเครือข่ายแบบ Client/Server ดังแสดงตามรูปที่ 7 โดย Database Application จะทำงานเป็นโปรแกรมเดียวบน PC ที่เป็น workstation และมีการสื่อสารผ่านเครือข่ายแยกกับโปรแกรมของ Informix Database Server ที่ Host Machine

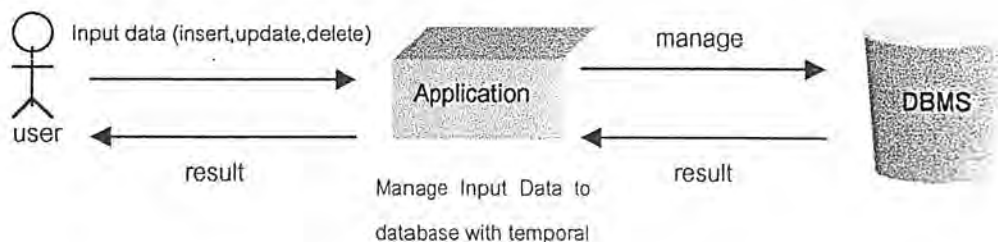


รูปที่ 7 สถาปัตยกรรมของการเชื่อมต่อ Client/Server

5.2 โปรแกรมจัดการข้อมูลในฐานข้อมูล (Manage Data Application)

โปรแกรมจัดการข้อมูลในฐานข้อมูล (Manage Data Application) เป็นโปรแกรมสำหรับการทำการเพิ่มข้อมูล (Insert), ลบข้อมูล (Delete), เปลี่ยนแปลงข้อมูล (Update) ในฐานข้อมูล โดยจะมีการจัดการตามวิธีของ Temporal Database เพื่อให้ผู้ใช้ สามารถค้นหา ข้อมูลโดยใช้ทฤษฎีของ Temporal สำหรับผู้ใช้ โปรแกรม นี้คือเจ้าหน้าที่ที่ทำหน้าที่จัดการกับการเปลี่ยนแปลงข้อมูลในฐานข้อมูล

5.2.1 สถาปัตยกรรมของโปรแกรมจัดการข้อมูลในฐานข้อมูล



รูปที่ 8 สถาปัตยกรรมของ Manage Data Application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 การทำงานของโปรแกรมจัดการข้อมูลในฐานข้อมูล

การทำงานของโปรแกรม แบ่งออกเป็น 3 งานหลัก ได้แก่ การ เพิ่มข้อมูล(Insert), การลบข้อมูล (Delete), การแก้ไขข้อมูล (Update) ในฐานข้อมูล โดยจะมีการจัดการตามทฤษฎีของ Temporal ดังนี้

5.2.2.1 การเพิ่มข้อมูล (Insert) เมื่อทำการเพิ่มข้อมูล โปรแกรมจะทำการจัดการให้ เพิ่มข้อมูล ลงไปทั้งตารางปัจจุบัน (current) และตารางที่เก็บอดีต (Audit log) ตัวอย่างเช่น ต้องการเพิ่มข้อมูลนักศึกษาเข้าไปยังตาราง Student ซึ่งมีรายละเอียดดังต่อไปนี้

Stdid = 39014499
 Tpren = นาย
 Epren = Mr.
 Tname = วุฒิชัย
 Ename = Wuttichai
 Degree = B
 Stdtype = B
 Valid_from = 9/25/1995

เมื่อทำการเพิ่มข้อมูล ลงในตาราง Student ที่ Column Std_serial จะใส่ค่าที่เรียงลำดับโดยอัตโนมัติ ซึ่งถ้าหากยังไม่มีข้อมูลอยู่เลย ที่แถว นี้ Column Std_serial ก็จะมีค่าเป็น 1 จากนั้นก็จะทำการเพิ่มข้อมูล ลงในตาราง Student_TT ด้วย โดยมีข้อมูลเหมือนกับในตาราง student

ส่วนใน column std_int จะมีค่าเหมือนกับ column std_serial สำหรับที่ column ที่เป็นเวลาของตาราง Student_TT จะทำการใส่ค่าดังนี้

Valid_from = วันที่ ที่ user กรอกเข้ามา (สมมติว่าเป็นวันที่ 11/24/1995)
 Valid_to = 12/31/9999
 TxStart = วันที่ ที่ทำการเพิ่มข้อมูลนั้น
 TxStop = 12/31/9999

Stdid	Tpren	Epren	Tname	EName	Degree	StdType	Std_int
39014499	นาย	Mr.	วุฒิชัย	Wuttichai	B	B	1

Valid_from	Valid_to	TxStart	TxStop
9/25/1995	12/31/9999	11/24/1995	12/31/9999

ตาราง Student_TT หลังจากทำการเพิ่มข้อมูล (insert)

สำหรับการเพิ่มข้อมูล ในตารางอื่น ๆ ก็จะเป็นในลักษณะเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2.2 การเปลี่ยนแปลงแก้ไขข้อมูล (Update) ถ้าหากมีการเปลี่ยนแปลงข้อมูล ในตาราง ปัจจุบัน (Current) จะทำการเปลี่ยนแปลงค่าไปตามที่ผู้ใช้ ทำการเปลี่ยนแปลง แต่ในตารางอดีต (Audit log) จะมีการเปลี่ยนแปลงในส่วนของเวลา และมีการเพิ่มแถวใหม่ซึ่งมีค่าตามที่ผู้ใช้ต้องการ ตัวอย่างเช่น จาก การเพิ่มข้อมูลนักศึกษา ลงไปยังตาราง Student ถ้ามีการเปลี่ยนแปลงค่า column ดังนี้

Degree = M

StdType = A

Valid_from = 8/21/1996

ที่ตาราง Student จะมีการเปลี่ยนแปลงค่าไป สำหรับตาราง Student_TT จะมีการเปลี่ยนค่า Valid_to ของแถวนั้น ให้เป็นตามค่า Valid_from ที่ผู้ใช้กรอก เข้ามาใหม่ และเปลี่ยนค่า TxStop เป็นวันที่ที่ทำการแก้ไขข้อมูล จากนั้นจะมีการเพิ่มแถว อีก 1 แถว โดยมีค่าตามที่ผู้ใช้ เปลี่ยน โดยมีค่าของ Column ที่เป็นเวลาดังนี้

Valid_from = 8/21/1996

Valid_to = 12/31/9999

TxStart = วันที่ที่ทำการ update (สมมติว่าเป็นวันที่ 8/21/1996)

TxStop = 12/31/9999

สำหรับ Column Std_int จะนำค่ามาจากแถวเดิม ก่อนที่จะทำการเปลี่ยนแปลงแก้ไข สังเกตได้ ว่าหากมีการแก้ไขข้อมูลแล้ว ค่าของ std_int จะมีค่าเท่ากับค่า std_int ของแถว เดิมก่อนการแก้ไข

Stdid	Tpren	Epre	Tname	Ename	Degree	StdType	Std_int
39014499	นาย	Mr.	วุฒิชัย	Wuttichai	B	B	1
39014499	นาย	Mr.	วุฒิชัย	Wuttichai	M	A	1

Valid_from	Valid_to	TxStart	TxStop
9/25/1995	8/21/1996	11/24/1995	8/21/1996
8/21/1996	12/31/9999	8/21/1996	12/31/9999

ตาราง Student_TT หลังจากทำการ Update

สำหรับการแก้ไขข้อมูล ในตารางอื่นจะเป็นไปในรูปแบบเดียวกัน

5.2.2.3 การลบข้อมูล (Delete) การลบข้อมูล ที่ตารางปัจจุบัน (Current) จะเป็นการลบข้อมูล ออกไปจากฐานข้อมูลจริง ๆ แต่ที่ตารางอดีต (Audit log) จะเป็นเพียงการแก้ไขข้อมูล ในส่วนของเวลา เท่านั้น ตัวอย่างเช่น ทำการลบข้อมูลในตาราง Student จากตัวอย่างข้างต้น โดยผู้ใช้ จะต้องกรอกว่าข้อมูลนั้นเป็นจริงถึงวันที่เท่าไร แล้วจะมีการเปลี่ยนแปลงข้อมูลเวลาดังนี้

Valid_to = วันสุดท้ายที่ข้อมูลนั้นเป็นจริง (สมมติว่าเป็นวันที่ 9/15/1997)

TxStop = วันที่ลบข้อมูลนั้นออกจากฐานข้อมูล(สมมติว่าเป็นวันที่ 10/23/1997)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Stdid	Tpren	Epren	Tname	Ename	Degree	StdType	Std_int
39014499	นาย	Mr.	วุฒิชัย	Wuttichai	B	B	1
39014499	นาย	Mr.	วุฒิชัย	Wuttichai	M	A	1

Valid_from	Valid_to	TxStart	TxStop
9/25/1995	8/21/1996	11/24/1995	8/21/1996
8/21/1996	9/15/1997	8/21/1996	10/21/1997

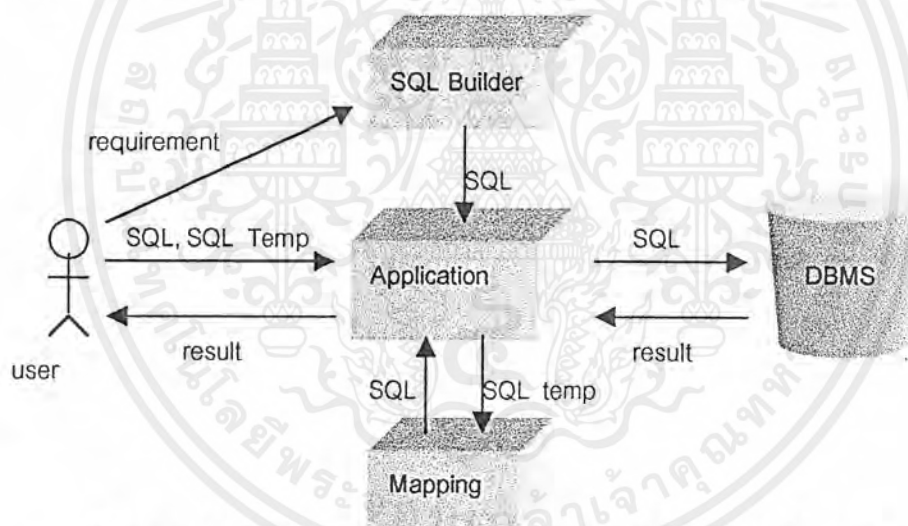
ตาราง Student_TT หลังจากทำการ Delete

สำหรับการลบข้อมูล ในตารางอื่น จะเป็นไปในรูปแบบเดียวกัน

5.3 โปรแกรมค้นหาข้อมูลโดย SQL และตัวสร้างภาษา SQL (SQL Editor Include SQL Builder)

เป็นโปรแกรมสำหรับการ ค้นหาข้อมูลจากฐานข้อมูลขึ้นมา โดยที่สามารถค้นหา โดยภาษา SQL มาตรฐานหรือ SQL/Temporal หรือใช้ตัวสร้างภาษา SQL

5.3.1 สถาปัตยกรรมของโปรแกรมค้นหาข้อมูลโดย SQL และตัวสร้างภาษา SQL



รูปที่ 9 สถาปัตยกรรมของโปรแกรมค้นหาข้อมูลโดย SQL และตัวสร้างภาษา SQL

5.3.2 การทำงานของโปรแกรมค้นหาข้อมูลโดย SQL และตัวสร้างภาษา SQL

โปรแกรมนี้จะทำการค้นหาข้อมูลตามที่ใช้ต้องการ โดยสามารถค้นหาโดยภาษา SQL หรือ SQL/Temporal หรือ ตัวสร้างภาษา (SQL Builder)

ถ้าผู้ใช้ ใช้ภาษา SQL/Temporal ในการค้นหาข้อมูล โปรแกรมจะทำการแปลงจากภาษา SQL Temporal ให้กลายเป็น SQL มาตรฐาน ตามทฤษฎีของฐานข้อมูลที่ขึ้นกับเวลา ซึ่งสามารถแบ่งออกเป็นกรณี ได้ดังต่อไปนี้

5.3.2.1 Current Validtime

เป็นการค้นหาข้อมูลที่เป็นจริงในปัจจุบันในตารางที่เก็บการเปลี่ยนแปลงในอดีต (Audit Log) แสดงได้ดังตัวอย่างต่อไปนี้

SQL Temporal	SQL
CURRENTVALID SELECT * FROM STUDENT_TT	SELECT * FROM STUDENT_TT WHERE Valid_to = '12/31/9999'

5.3.2.2 Sequenced Validtime

เป็นการค้นหาข้อมูลที่เป็นจริงในอดีตจากตาราง ที่เก็บการเปลี่ยนแปลงของข้อมูล (Audit Log) ตัวอย่างเช่น “จงหาประวัติของนักศึกษา รหัส 39014499 แสดง เฉพาะ รหัสและชื่อของนักศึกษา”

SQL/Temporal	SQL
VALIDTIME SELECT Stdid,ENAME FROM STUDENT_TT WHERE Stdid = '39014499'	SELECT Stdid,ENAME,Valid_from,Valid_to FROM STUDENT_TT WHERE Stdid = '39014499'

5.3.2.3 Nonsequenced Validtime

เป็นการค้นหาข้อมูลตามที่ใช้ต้องการ โดยไม่คำนึงถึงช่วงเวลาข้อมูลนั้นเป็นจริง จากตารางที่เก็บการเปลี่ยนแปลงของข้อมูลในอดีต (Audit Log) ตัวอย่างเช่น

SQL/Temporal	SQL
NONSEQUENCEVALIDTIME SELECT Stdid,ENAME FROM STUDENT_TT WHERE Stdid = '39014499'	SELECT Stdid,ENAME FROM STUDENT_TT WHERE Stdid = '39014499'

5.3.2.4 Current Transactiontime

เป็นการค้นหาว่าข้อมูลนั้นถูกเก็บอยู่ในฐานข้อมูลในปัจจุบัน จากตารางที่เก็บการเปลี่ยนแปลงของข้อมูลในอดีต (Audit Log) ตัวอย่างเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SQL/Temporal	SQL
CURRENTTRANSACTION SELECT * FROM STUDENT_TT	SELECT * FROM STUDENT_TT WHERE TxStop = '12/31/9999'

5.3.2.5 Sequenced Transactiontime

เป็นการค้นหาข้อมูลที่เกี่ยวข้องอยู่ในฐานข้อมูลในอดีตจากตารางที่เก็บการเปลี่ยนแปลงของข้อมูล (Audit Log) ตัวอย่างเช่น “เมื่อใดที่นักศึกษารหัส 39014499 ถูกบันทึกว่ามีชื่อคือ Wuttichai ”

SQL/Temporal	SQL
TRANSACTIONTIME SELECT Stdid, Ename FROM STUDENT_TT WHERE Stdid = '39014499' AND Ename = 'Wuttichai'	SELECT Stdid, Ename, TxStart, TxStop FROM STUDENT_TT WHERE Stdid = '39014499' AND Ename = 'Wuttichai'

5.3.2.6 Nonsequenced Transactiontime

เป็นการค้นหาข้อมูลตามที่ใช้ต้องการ โดยไม่คำนึงถึงช่วงเวลาที่ข้อมูลนั้นถูกเก็บในฐานข้อมูล จากตารางที่เก็บการเปลี่ยนแปลงของข้อมูลในอดีต (Audit Log) ตัวอย่างเช่น

SQL/Temporal	SQL
NONSEQUENCETRANSACTIONTIME SELECT Stdid, Ename FROM STUDENT_TT WHERE Stdid = '39014499'	SELECT Stdid, Ename FROM STUDENT_TT WHERE Stdid = '39014499'

5.4 วิธีการติดต่อกับฐานข้อมูลที่ใช้ในระบบงานลงทะเบียนนักศึกษา

การติดต่อกับฐานข้อมูลโดยผ่านวิซวลเบสิก นั้นอาจทำได้หลายวิธีแต่สำหรับในแอปพลิเคชันระบบสืบค้นข้อมูลนักศึกษานั้นมีการใช้อยู่ 3 วิธี ได้แก่

1. ใช้ระบบจัดการฐานข้อมูลของอินฟอร์มิคส์ (informix Data Director for Visual Basic)
2. ใช้ระบบจัดการฐานข้อมูลแบบเจ็ต (JET Database Engine)
3. ใช้ระบบจัดการฐานข้อมูลภายนอกโดยตรง (ODBCDirect Workspace)

ซึ่งในรายงานฉบับนี้จะได้อธิบายถึง ลักษณะของการติดต่อกับฐานข้อมูลในทั้ง 3 ลักษณะรวมถึงเปรียบเทียบข้อดีข้อเสียของแต่ละวิธีด้วย

5.4.1 การติดต่อกับฐานข้อมูลโดยผ่านระบบจัดการฐานข้อมูลของ อินฟอร์มิคส์

(informix Data Director for Visual Basic)

Data Director Object (DDO) เป็นเซ็ทของออปเจกต์ที่ซ่อนความซับซ้อนของการทำงานต่าง ๆ ของฐานข้อมูลทำให้ง่ายต่อการใช้งานและนอกจากนี้ยังมีลักษณะอื่นๆอีกหลายประการที่ควรเลือก DDO ในการใช้สร้างแอปพลิเคชัน ได้แก่

5.4.1.1 ขยายส่วนแอปพลิเคชันต้นแบบ (Extending a prototype Application)

เราไม่ต้องทำการสร้างออปเจกต์ ทั้งหมดในการเขียนโปรแกรม ถ้าเรามีการสร้างออปเจกต์ ไปยังอินเทอร์เน็ตของผู้ใช้ เราสามารถอ้างอิงออปเจกต์นั้นได้ โดยมากแล้วการพัฒนาแอปพลิเคชัน ผู้พัฒนาจะเริ่มต้นโดยการ ใช้ Data Director GUI ในการพัฒนาต้นแบบ หลังจากนั้นจึงพัฒนาแอปพลิเคชัน โดยใช้ DDO ตัวอย่างเช่น ถ้าเราทำการอิมพอร์ต โครงสร้างของ ฐานข้อมูลมาเป็น Model โดยใช้ Model import wizard เราสามารถอ้างอิง Model file ภายใน code แทนที่จะเขียนโปรแกรมเพื่อที่จะสร้าง model file โดยใช้ DDO

5.4.1.2 เพิ่มประสิทธิภาพของแอปพลิเคชัน (Performance)

ในบางกรณีประสิทธิภาพของ application นั้นสามารถทำให้เพิ่มขึ้นโดยการกำจัดการเชื่อมโยงแบบ drag และ drop ตัวอย่างเช่น สมมติว่า application นี้มี form ซึ่ง user ต้องการที่จะ access เดือนละครั้งเท่านั้น ถ้าใช้วิธี drag และ drop ในการสร้างการเชื่อมโยงระหว่าง object บน form และ column จะมีการสร้างการเชื่อมโยง ในฐานข้อมูล ทุก ๆ ครั้งที่มีการ Start application ซึ่งไม่ทำเป็นเพราะจะได้ข้อมูลที่ใช้ไม่บ่อย ถ้าเราใช้ DDO แทน code นี้จะถูก execute เมื่อ form นั้นถูก active โดย user เท่านั้น ไม่ใช่ทุก ๆ ครั้งที่ application นี้ start up

5.4.1.3 เพิ่มการติดต่อกับ โปรแกรมแบบอื่นๆได้ (Access ไปยัง Software Environment อื่น ๆ)

การเขียนโปรแกรมโดยใช้ DDO นั้นมีพื้นฐานมาจาก Component Object Model (COM) สิ่งนี้เองที่ทำให้ code ของ DDO ทำงานได้ใน software ต่าง ๆ ซึ่งสนับสนุน COM ตัวอย่างเช่น เราสามารถเขียน DDO code ใน Excel application โดยใช้ฐานข้อมูล Informix นอกจากนั้น DDO API ยังอนุญาตให้ผู้พัฒนาทำงานใน 3GL environment ตัวอย่างเช่น C++ และ J++

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-สามารถใช้ Data type ชนิดพิเศษสำหรับ unconventional data

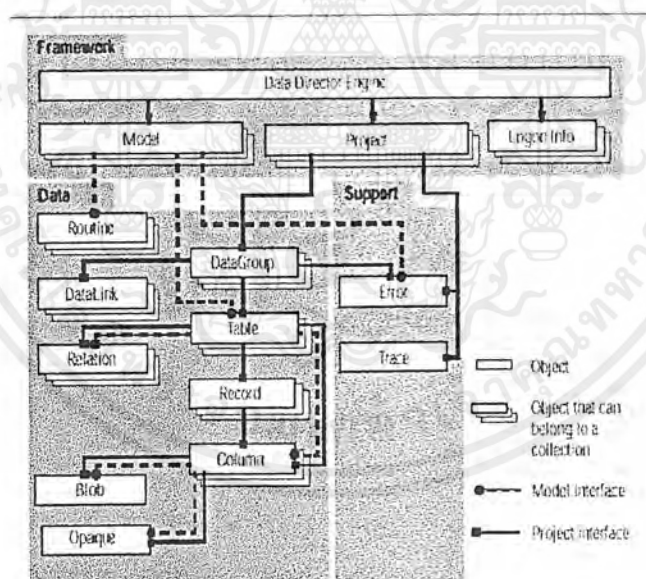
เราจะใช้ DDO ในการแสดงค่าของ opaque data (user define) ใน application ถ้าข้อมูลนั้นนำเสนอเป็น character string ได้ไม่ดี บ่อยครั้งที่ Opaque data type จะเก็บข้อมูลที่เป็น unconventional เช่น sound , video และ color เราสามารถใช้ DDO ในการทำ operation บนข้อมูล เช่น อ่านและเขียนจาก buffer หรือ file

5.4.1.4 มีความยืดหยุ่นสูงกว่าเมื่อเวลารันไทม์(Runtime)

ในการใช้ DDO สามารถที่จะทำการเปลี่ยน model ระหว่าง runtime ตัวอย่างเช่น การเพิ่มตารางใหม่หรือตารางเสมือน แต่ถ้าใช้ GUI เมื่อเวลา runtime application จะสามารถใช้เฉพาะ model ที่ได้กำหนดไว้เมื่อ Designime เท่านั้น

5.4.2 ลำดับชั้นของ Data Director Objects

DDO ประกอบด้วยออบเจกต์ที่สามารถโปรแกรมได้ซึ่งจะมีการคิดต่อไปยังออบเจกต์ที่โปรแกรมได้อื่น ๆ เป็นลำดับชั้นออบเจกต์ที่เป็น “owns” จะทำการควบคุมออบเจกต์อื่นๆ เมื่อมันอยู่ในลำดับชั้นที่สูงกว่าออบเจกต์ที่เป็นเจ้าของเรียกว่า “parent object” และออบเจกต์ที่ถูกเป็นเจ้าของเรียกว่า “child object” โดยที่ object หนึ่งๆ สามารถมีเจ้าของได้มากกว่า 1 ตัว ตัวอย่างเช่น Error object นั้นถูกเป็นเจ้าของโดย Model, Project และ Datagroup Object



รูปที่ 10 แสดงลำดับชั้นของ DDO

5.4.4 Objects and Collections

Collection ประกอบด้วยอินสแตนซ์ของออบเจกต์มากมาย เช่น Table, Project และ Model ตัวอย่างเช่น Datagroup object สามารถเป็นสมาชิกของ Datagroups collection ซึ่งประกอบด้วยบาง Datagroup objects โดยปกติ Collections จะถูกอ้างในรูปพหูพจน์เสมอ (ตัวอย่างเช่น Model object, Models collection)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Collection มี unique set ของคุณสมบัติและเมทอธที่ แตกต่างจากสมาชิกของมัน และมันยังไม่สามารถมี คุณสมบัติและ เมทอธที่เดียวกันได้

เราสามารถทำแอคชั่นต่างๆ เหล่านี้กับ collections

- แก้ไขแต่ละออปเจกต์ใน collection
- ลบแต่ละออปเจกต์ออกจาก collection
- นับจำนวนออปเจกต์ใน collection

Collection เป็นเจ้าของโดย Engine, Project, Model, Datagroup, Table, Relation, Routine และ Record objects

Collection เป็นเจ้าของแต่ละ instance ของ objects ซึ่งประกอบด้วยตัวมันเอง เช่น Table collection เป็นเจ้าของ Table objects , Record collection เป็นเจ้าของ Record objects เป็นต้น

5.4.5 ชนิดของ Objects

Data Director Objects สามารถแบ่งได้เป็น 3 ชนิด ได้แก่

- Framework objects
- Data objects
- Support objects

5.4.6 Framework objects

แต่ละ application ที่เราสร้างโดยใช้ Data Director และ DDO จะมีการจัดการที่เป็นแบบแผน และมีโครงสร้าง ฐานข้อมูลที่เราใช้งานอยู่ก็จะมีโครงสร้างเช่นเดียวกัน Framework objects จะนำเสนอโครงสร้างเหล่านี้ และให้การ access ไปยัง object อื่น ๆ

ต่อจากนี้จะทำการแนะนำและข้อมูลรายละเอียดสำหรับแต่ละ framework objects

- Engine

Engine objects เป็น root objects เราจะต้องทำการสร้าง object นี้ก่อนที่จะทำการสร้าง Data Director Objects อื่น ๆ สำหรับ Engine object สามารถควบคุม Project และ Model objects เราสามารถที่จะสร้างหลาย ๆ project และ model โดยใช้ Engine objects เพียงตัวเดียวได้ ดังนั้นจะมีเพียง Engine object เดียวเท่านั้นที่อนุญาตสำหรับการ run แต่ละครั้งของ Visual Basic หรือ Data Director

เราสามารถทำ action เหล่านี้ได้กับ Engine object

- สร้าง Project objects
- สร้าง Model objects
- สร้าง LogonInfo objects

Engine Object ไม่สามารถถูกเป็นเจ้าของโดย Object อื่น ๆ

Engine Object เป็นเจ้าของ Project Object, Model Object และ LogonInfo Object

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- LogonInfo

LogonInfo object จะเป็นตัวกำหนดชนิดของฐานข้อมูล, username, password, server ID และ database name ซึ่งใช้เมื่อต้องการที่จะ Logging on ไปยัง database

เราสามารถกระทำ action ต่าง ๆ เหล่านี้กับ LogonInfo Object ได้

- Log on ไปยังฐานข้อมูล
- Import ฐานข้อมูลมาเป็น Model

LogonInfo Object ถูกเป็นเจ้าของโดย Engine object

LogonInfo Object ไม่สามารถเป็นเจ้าของ objects อื่น ๆ

- Project

Project object รวบรวม DataGroup, Table, Column และ Relation Objects ซึ่งจะมีการติดต่อกันโดย Data Director project (.ddx) file

เราสามารถที่จะกระทำ action เหล่านี้กับ Project Object

- สร้าง Datagroup Objects
- Enable และ Disable การควบคุม Runtime Error

Project Object ถูกเป็นเจ้าของโดย Engine Object

Project Object เป็นเจ้าของ Datagroup Object, Error Object และ Trace Object

- Model

Model Object จะทำการ access และ update ข้อมูลของ Data Director Model มันจะขึ้นตรงกับ Model (.mlt หรือ .mlx) file

เราสามารถกระทำ action เหล่านี้กับ Model Object

- import โครงสร้างของฐานข้อมูลมาเป็น Model
- Copy และ Save Model
- สร้าง Table Object
- สร้าง LogonInfo Object

Model Object ถูกเป็นเจ้าของโดย Engine Object

Model Object เป็นเจ้าของ Table Object, Routine Object และ Error Object

5.4.7 Data Objects

แต่ละฐานข้อมูลที่เราใช้งานนั้นจะประกอบด้วยส่วนต่าง ๆ ที่แน่นอน ตัวอย่างเช่น Tables, Columns และ rows ซึ่ง Data Object จะทำการนำเสนอและ Enable ให้เราทำงานในส่วนต่าง ๆ ของฐานข้อมูลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- DataGroup

DataGroup Object จะขึ้นตรงกับ Data Director DataGroup และเป็นกลุ่มที่มีความสัมพันธ์กับ Table และ Datalinks ของฐานข้อมูล

เราสามารถที่จะกระทำ action ต่อไปนี้ให้กับ DataGroup Object

- สร้าง LogonInfo Object
- สร้าง DataLink objects
- สร้าง Routine Objects
- Log on และ Log Off ฐานข้อมูล
- Set Option ต่าง ๆ สำหรับการรายงานความผิดพลาด
- Execute SQL Statements
- Set Option ต่าง ๆ สำหรับ record - level locking

- DataLink

DataLink Object ขึ้นตรงกับ Data Director DataLink ซึ่งจะทำหน้าที่ในการสร้างการติดต่อ ระหว่าง column ของฐานข้อมูล และ Visual Basic Control การเชื่อมต่อนี้จะเป็นตัวบ่งบอกว่าข้อมูลใดที่ปรากฏบน application form

เราสามารถกระทำ action ดังต่อไปนี้กับ DataLink Object

- Copy DataLink
- Set คุณสมบัติต่าง ๆ ให้กับ DataLink
- บ่งบอกว่า DataLink นี้เป็น Real หรือ Virtual DataLink (Virtual DataLink จะเป็นการชี้ Column แต่ไม่ใช่ Visual Basic Control)

DataLink Object ถูกเป็นเจ้าของโดย DataGroup Object

DataLink Object ไม่ได้เป็นเจ้าของ Object ใด ๆ

- Table

Table Object ขึ้นตรงกับ Table ของฐานข้อมูล ซึ่ง Table จะประกอบด้วย Column และ row โดย Column จะประกอบด้วยรายละเอียดต่าง เช่น Lastname, Firstname, address และ phone number ส่วน Rows หรือ Record จะประกอบด้วยข้อมูลต่าง ๆ ภายใต้อายัดละเอียดของ Column

เราสามารถที่จะกระทำ action ต่อไปนี้กับ Table Object

- สร้าง Relation Objects
- สร้าง Column Objects
- สร้าง Record Objects
- เป็นตัวนำทางไปยัง record ใน result set
- เป็นตัวนำทางไปยัง Bookmark Record
- กำหนดและ execute stored procedures, functions และ user define routines
- Set ค่า parameters ต่าง ๆ ใน SQL Statements

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table Object ถูกเป็นเจ้าของโดย DataGroup และ Model Objects

Table Object เป็นเจ้าของ Columns และ Relation Collections ถ้า Table object เป็นส่วนหนึ่งของ DataGroup มันจะเป็นเจ้าของ Record Collection ด้วย

- Column

Column object ขึ้นตรงกับ column ของฐานข้อมูล ซึ่ง Column เป็นหน่วยย่อย มาตรฐานของฐานข้อมูล Column จะประกอบด้วยรายละเอียดต่าง ๆ ตัวอย่างเช่น Lastname, Firstname, address, phone number

เราสามารถที่จะทำ action ดังต่อไปนี้กับ Column Object

- ทำการ Access binary large objects (ค่าของ blob) จากฐานข้อมูล
- ทำการ Access user define (opaque) data type จากฐานข้อมูล
- Set Option สำหรับค่าของ Blob
- Set ชนิดของข้อมูลใน column, ขนาด และวิธีการในการจัดเรียง
- กำหนด Column ที่เป็น primary key

Column object ถูกเป็นเจ้าของโดย Table และ Record Objects

Column object เป็นเจ้าของ Blob object และ Opaque object

- Record

Record Object ขึ้นตรงกับ record ของฐานข้อมูล ซึ่งประกอบด้วยข้อมูลที่ถูกกำหนดโดยรายละเอียดของ column

เราสามารถที่จะกระทำ action ต่าง ๆ ดังต่อไปนี้กับ Record Object

- Copy Record
- Clear ค่า Change flags จาก record
- Reset ข้อมูลใน record ให้เป็นค่าเดิม
- Lock Record ขณะที่มีการเปลี่ยนแปลง

Record Object ถูกเป็นเจ้าของโดย Table Object

Record Object เป็นเจ้าของ Column Object

- Relation

Relation object จะกำหนดความสัมพันธ์ระหว่างตาราง 2 ตาราง

เราสามารถที่จะกระทำ action ต่าง ๆ ดังต่อไปนี้กับ Relation Object

- ชื่อตำแหน่งไปยังจุดเริ่มต้นหรือจุดหมายของตาราง ซึ่งเป็นการสร้างความสัมพันธ์
- ชื่อตำแหน่งไปยังจุดเริ่มต้นหรือจุดหมายของ column ซึ่งเป็นการสร้างความสัมพันธ์
- เพิ่มจุดเริ่มต้นและจุดหมายของ column ไปในความสัมพันธ์
- ชื่อว่าเป็นความสัมพันธ์แบบใด

Relation object ถูกเป็นเจ้าของโดย Table Object

Relation object ไม่ได้เป็นเจ้าของ object ใด ๆ เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Blob

Blob object นั้นจะนำเสนอ Binary Large Object (blob) โดยปกติแล้ว blob จะเก็บค่าของข้อมูลขนาดใหญ่ เช่น bitmap file ซึ่งเราต้องการ access เป็นหน่วย ๆ หนึ่ง

Blob object จะเกี่ยวข้องกับ column และ record คือ ถ้ามีการย้ายจาก record หนึ่งไปยังอีก record หนึ่ง Blob object จะชี้ไปยังค่าของ blob สำหรับ record นั้น

เราสามารถที่จะกระทำ action ต่าง ๆ ดังต่อไปนี้กับ Blob object

- Retrieve blob data
- Set ข้อมูลที่อยู่ใน Blob object
- Save และ Load blob objects จาก files

Blob object ถูกเป็นเจ้าของโดย Column object

Blob object ไม่ได้เป็นเจ้าของ Object ใด ๆ เลย

- Opaque

Opaque Object นำเสนอ user define data type ใน Informix Server ซึ่งมี Data Option ที่เป็นสากล user define data type เป็น data type ที่ถูกสร้างโดยผู้เขียนโปรแกรมในการนำเสนอข้อมูลชนิดพิเศษ เช่น ข้อมูลที่เป็น Time-series

Opaque object นั้นเกี่ยวข้องกับ Column และ Record ซึ่งถ้ามีการย้ายจาก record หนึ่งไปยังอีก record หนึ่ง Opaque Object จะมีความเกี่ยวข้องกับค่าใน record นั้น ๆ

เราสามารถกระทำ action ต่าง ๆ ดังต่อไปนี้กับ Opaque Object

- Retrieve user define data
- Save และ Load Opaque objects จาก application buffer
- Save และ Load Opaque objects จาก files
- ตรวจสอบว่า Opaque object อยู่ในความยาวที่จำกัดหรือไม่
- ตรวจสอบว่า Opaque object เป็น null หรือไม่
- เปรียบเทียบระหว่าง 2 object ว่ามี Opaque data type เดียวกันหรือไม่

Opaque object ถูกเป็นเจ้าของโดย Column object

Opaque object ไม่ได้เป็นเจ้าของ object ใด ๆ เลย

- Routine

Routine Object นำเสนอ ชนิดต่าง ๆ ของ user หรือ system define function หรือ Store procedure ในฐานข้อมูล การใช้ Routine Object เราสามารถเพิ่ม Store procedure และ user define routine จากฐานข้อมูลไปยัง Model Object

Routine object ถูกเป็นเจ้าของโดย Model object

หมายเหตุ เราจะได้ Routine object มาจาก Datagroup object ไม่ใช่ Model object เพื่อที่จะเรียก routine โดย routine object จะถูกส่งค่ากลับโดย Model object ซึ่งเป็นที่เก็บข้อมูลของ routine ไม่ได้หมายความว่า จะ execute routine

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Routine ไม่ได้เป็นเจ้าของ object ใด ๆ เลย

5.4.8 Support Objects

แต่ละ project ที่ทำงานโดยใช้ Data Director ต้องการที่จะกระทำ “housekeeping” function เพื่อที่จะเก็บหนทางที่จะทำให้เกิดความผิดพลาด และตำแหน่งของ file ซึ่ง Support Object จะทำหน้าที่ในการกระทำ function เหล่านี้

- Error

Error object เป็นตัวบ่งบอกชนิด, เวลา และ text ของ error

Error object ถูกเป็นเจ้าของโดย Project, Model และ DataGroup objects

Error object ไม่สามารถเป็นเจ้าของ object ใด ๆ เลย

- Trace

Trace Object ขึ้นตรงกับ Data Director Trace file ซึ่ง Trace file นี้เป็นบันทึกของคำสั่งใน Data Director ที่ถูก execute

เราสามารถที่จะกระทำ action ดังต่อไปนี้กับ Trace Object

- Enable หรือ Disable Tracing

- เพิ่ม Trace File

Trace object ถูกเป็นเจ้าของโดย Project object

Trace object ไม่สามารถเป็นเจ้าของ object ใด ๆ ได้

Parameters ของ DDO

ใน Data Director เราสามารถใช้ parameters ได้ 2 ชนิด

- Binding parameters สามารถถูกแทนได้โดยค่าต่าง ๆ ที่มีชนิดของข้อมูลที่เข้ากันได้ ที่เวลา runtime

- Replacement parameters เพื่อชี้ไปยัง column ของตาราง

Binding parameters

Binding parameters หรือ SQL parameters นั้นมีความสัมพันธ์กับ ค่าคงที่หรือตัวแปรของ SetParam function และจะถูกนำเสนอโดย question mark (?)

ตัวอย่างของ SQL Statement ซึ่งมี binding parameters ได้แก่

```
INSERT INTO customer (customer_num, lname, phone)
VALUES (?, ?, ?);
```

ที่เวลา runtime Application จะแทนค่าต่าง ๆ ซึ่งเหมาะสมกับ column ของฐานข้อมูล ซึ่งมีความสัมพันธ์กับ parameters โดยการเรียก SetParam และมีการทำ operation เป็น loop จนกว่าจะเสร็จครบทุกแถว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บาง ODBC driver ไม่สนับสนุน SQLDescribeParam ซึ่งเป็น Level – two ODBC function ที่ทำการอธิบายชนิดของ parameters และชนิดของข้อมูลเมื่อเวลา runtime

ถ้าเราใช้ Model viewer ในการสร้างตารางเสมือน จาก SQL Statement โดยใช้ binding parameter และ ODBC driver ที่ใช้ไม่ support SQLDescribeParam เรายังคงสามารถใช้ parameters โดยการกำหนด varDataType, varDataSize, varParamType parameters เมื่อเรียก SetParam

- varDataType เป็น string ที่กำหนดชนิดของข้อมูลของ parameter เช่น varchar หรือ numeric
- varDataSize เป็นเลขจำนวนเต็มซึ่งกำหนดขนาดของ string parameters
- varParamType เป็น integer ที่กำหนดชนิดของ Parameter ดังนี้
 1. Input Parameter
 2. Input/Output Parameter
 3. Result column
 4. Output parameter
 5. Return Value

ถ้าเราไม่ทำการกำหนด parameter เหล่านี้ Data Director จะ set parameter เป็นค่า default

5.4.9 Replacement Parameters

Replacement parameters อนุญาตให้ application ทำการ execute virtual table บนค่าของ column ใน physical table โดย Replacement parameters จะนำเสนอโดยใช้ Double colon (::) ตัวอย่างของ SQL Statement โดยใช้ Replacement parameter คือ

```
SELECT * FROM orders
WHERE customer_num = ::customer_custnum
```

ที่เวลา runtime Replacement parameters ::customer_custnum จะถูกแทนโดยค่าของ column customer.customer_num ภายใน record ปัจจุบัน เหมือนกับ binding parameter ซึ่งจะมีการทำเป็น Loop จนกระทั่งเสร็จ

5.5 การใช้ Data Director Objects

5.5.1 การใช้ properties และ methods

ประสิทธิภาพของ DDO นั้นอยู่ที่การซ่อนความซับซ้อนของการ access ข้อมูลต่าง ๆ ให้อยู่ในรูปของ properties และ methods ของ objects ซึ่งเราสามารถใช่วิธี methods และ properties ได้โดยอ้างถึงมัน ทำให้เราสามารถลดความยาวของ code ลงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนอื่นต้องทำการสร้าง object พื้นฐาน โดยใช้ DDO methods และ properties ซึ่งทำได้โดย syntax ง่าย ๆ Data Director Object แต่ละตัวมี methods และ properties ทำให้เรา execute function ต่าง ๆ ได้

5.5.2 การใช้ Visual Basic Code Window

การใช้ Visual Basic code window เขียนโปรแกรมเพื่อที่จะสร้าง, access, และ set properties และ methods สำหรับ Data Director Objects ในการ access code window ให้ทำการเลือก form, object หรือ module มาตรฐานอื่น ๆ จากหน้าต่าง Visual Basic project แล้ว click ปุ่ม View code

5.5.3 Viewing Properties และ methods

Data Director properties และ methods สามารถดูได้โดยใช้ Visual Basic Object Browser ซึ่ง Properties และ Methods เป็น interfaces พื้นฐานที่ทำให้เราสามารถดู DDO ใน application ได้

Properties เป็น attribute ของ object ซึ่งจะเป็นตัวกำหนด state ของ object ตัวอย่างเช่น properties ชื่อ Visible สามารถ set เป็น true เพื่อที่จะทำให้ object นั้นมองเห็น แต่ถ้าเป็น false ก็จะทำให้ object นั้นหายไป

Methods เป็น attribute ของ object ซึ่งจะกระทำ action ต่าง ๆ ของ object ตัวอย่างเช่น Methods ชื่อ Resize จะเกี่ยวข้องกับการเปลี่ยนแปลงขนาดของ object

การดู Data Director Objects properties และ method สามารถทำได้ดังนี้
เลือก View → Object Browser จาก menu bar ของ visual basic ซึ่ง Object Browser dialog box จะปรากฏขึ้น

จาก Libraries / Projects List box เลือก DDO

5.5.4 การใช้ properties

การใช้ properties ของ object เราจะอ้างถึงได้โดยบอก object name และ properties name ซึ่งมีรูปแบบดังนี้

```
ObjectName.PropertyName
```

Properties จะไม่มี parameter แต่สามารถจะ set ให้เป็นค่าที่เหมาะสม ดัง syntax ดังต่อไปนี้

```
ObjectName.PropertyName = Value
```

5.5.5 การใช้ Method

ในการใช้ Methods ของ object เราสามารถอ้างถึงได้โดยการใช้ syntax ที่บอก object name, method name และ method parameters ซึ่งมีรูปแบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ObjectName.MethodName(Parameter, [Parameter]...)
```

Optional parameter เขียนแทนด้วย []

ถ้า Method นั้นไม่มี parameter ใด ๆ เลย จะเขียนเป็น Statement ได้ดังนี้

```
ObjectName.MethodName
```

5.5.6 ประโยชน์ของ Properties และ methods

Properties และ Methods มีมากมายสำหรับแต่ละ Object แต่จะมีเพียงบาง Method และ properties เท่านั้นซึ่งนำมาใช้ให้เป็นประโยชน์ได้เป็นพิเศษ

- **Parent Property** Data Director Object แต่ละตัวมี Parent property ซึ่ง Parent property ของ object จะทำให้เราสามารถเคลื่อนย้ายลำดับชั้นของ DDO โดยไม่มีการอ้างอิงที่แน่นอนไปยัง object ที่เราต้องการ access ตัวอย่างเช่น

```
ObjectSource.[[ParentObjectName]...].ObjectDestination.PropertyMethodName
```

Parent property สามารถที่จะถูกเรียกจนกระทั่งสูงสุดของลำดับของ DDO (Engine Object)

- **Application Property** Data Director Object แต่ละตัวจะมี application property โดย application property นี้จะส่งค่าอ้างอิงไปยัง Engine Object ซึ่งการอ้างอิงนี้ทำให้เราสามารถเขียนโปรแกรมกระโดดไปยัง Engine Object ได้โดยตรง ไม่มีการเรียกหลาย ๆ ครั้งไปยัง Parent property

- **Return** จำนวนของ Table หรือ Column นับจำนวนของ Table ภายใน Datagroup ซึ่งเขียนเป็น code ได้ดังนี้

```
oDataGroup.Tables.Count
```

นับจำนวนของ column ในตารางเขียนเป็น code ได้ดังนี้

```
oDatagroup.Tables("NameOfTable").Columns.Count
```

- **Return** ชื่อตาราง ต้องการชื่อของตารางภายใน DataGroups collection เขียนเป็น code ได้ดังนี้

```
oDataGroup.Table(1).Name
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Clone method** จะทำการสร้าง object อีกตัวหนึ่ง รวมถึงทุก ๆ สิ่งของ object นั้น Clone method นั้นใช้ได้กับ Model และ LogonInfo Objects

5.6 เริ่มต้นการใช้ operation ของ DDO

ก่อนที่จะใช้ Data Director Object จะต้องมีการสร้าง instance ของ object นั้น ซึ่งมี 2 step ในการเริ่มต้นการใช้ DDO

1. สร้าง Engine Object ซึ่งเป็น Top - Level object ทำหน้าที่ access Data Director Object ตัวอื่น ๆ ทุกตัว

2. สร้าง Project object ซึ่งเป็น object ที่มีหน้าที่สร้างและ access Data Director (.ddx) project file และ object ทุก ๆ ตัวที่อยู่ภายใต้ Project Object

ขั้นตอนที่ 1 สร้าง Engine Object

ขั้นตอนแรกจะใช้ DDO ในการสร้าง Engine object ซึ่งต้องทำอย่างเป็นขั้นตอนเพราะว่า Engine object เป็น Top-level object ในลำดับชั้นของ DDO โดย Data Director Object ตัวอื่น ๆ ทุกตัวจะทำการ access ผ่านไปยัง Engine Object

ในการสร้าง Engine Object จะใช้ syntax ดังนี้ (สำหรับ 2 บรรทัดแรกเป็นการสมมติว่าได้มีการประกาศตัวแปรชื่อว่า oEngine เป็น Class DdoEngine ไว้แล้ว)

```
Set oEngine = CreateObject("DataDirector.Engine")
Set oEngine = New ddoEngine
Dim oEngine As New ddoEngine
```

ขั้นตอนที่ 2 สร้าง Project Object

หลังจากสร้าง Engine Object แล้ว จะทำการสร้าง Project Object ซึ่งต้องทำเป็นขั้นตอน เพราะ object อื่น ๆ ในลำดับชั้นของ DDO จะ access ผ่าน Project Object ยกเว้น Model Object และ LogonInfo Object

การสร้าง project object ใหม่ จะใช้ CreateProject method ของ Engine Object

```
Dim oProject As ddpProject
Set oProject = oEngine.CreateProject
OProject.Name = "Sport"
```

Code นี้จะทำการสร้าง project ใหม่ ด้วยชื่อ sport

หมายเหตุ จากข้างต้นแต่ละครั้งที่มีการเรียก CreateProject Data Director จะทำการสร้าง Project object ใหม่ทุกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้ามีการสร้าง Visual basic project ไว้แล้ว คือมีการออกแบบส่วนที่เป็น interface และ save project เราสามารถที่จะสร้าง project object โดยอ้างอิง Data Director Project file ที่มีอยู่แล้วได้ ซึ่งทำได้โดย code ดังนี้

```
Dim oProject As ddoProject
Set oProject = oEngine.CreateProject("c:\Program Files\Microsoft Visual
Basic\sport.ddx")
```

Code ตามข้างต้นนี้จะทำการสร้าง Project Object จาก Data Director project file ชื่อ sport.ddx เมื่อบอกชื่อของ project ที่มีอยู่แล้ว Data Director จะทำการสร้าง DataGroup Object ขึ้นมา โดยอัตโนมัติจากข้อมูลใน project file เราสามารถใช้ properties และ methods ของ DataGroup object ได้ เราสามารถใช้ syntax นี้ได้

```
Dim oProject As ddoProject
Set oProject = oEngine.CurrentProject
```

Code ข้างต้นนี้จะบอก Data Director ให้ใช้ project ที่มีอยู่แล้ว โดยไม่ต้องบอก path name ของ project

จาก code ดังต่อไปนี้แสดงให้เห็นว่าเราสามารถ access สมาชิกของ project collection ในกรณีนี้จะต้องมีการเจาะจง project ใน collection

```
Dim oEngine As ddoEngine
Dim oProject1 As ddoproject
Dim oProject2 As ddoproject
Set oEngine = New ddoEngine
oEngine.CreateProject("sports.ddx")
oEngine.CreateProject("sports.ddx")
Set oProject1 = oEngine.Project(1)
Set oProject2 = oEngine.Project(2)
```

ขั้นตอนที่ 3 สร้าง Additional Objects

หลังจากที่เราทำการสร้าง Data Director Object พื้นฐานแล้ว เราสามารถสร้าง object ในลำดับชั้นของ DDO อื่น ๆ ได้ โดยแต่ละ object จะสร้างโดยใช้ Method จาก parent object ตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น ถ้าเราสร้าง DataGroup Object โดยใช้ CreateDataGroup method ของ Project Object ซึ่งเป็น Parent ของ DataGroup Object

- การสร้าง Model Object

หลังจากที่เราทำการสร้าง Engine Object เราสามารถที่จะสร้าง Model Object เพื่อที่จะ access Table และ Column ในฐานข้อมูล ตัวอย่างเช่น ใช้ข้อมูลใน model สร้าง DataLink ระหว่าง database column และ Visual Basic Control

เราจะใช้ CreateModel method ของ Engine Object ในการสร้าง Model object ดังแสดงในตัวอย่าง ซึ่งตัวอย่างนี้ใช้ pathname ของ model file เป็น parameter สำหรับ CreateModel method

```
Dim oModel As ddoModel
Set oModel = oEngine.CreateModel("c:\Program Files\Microsoft Visual
Basic\sports.mlt")
```

จาก code ข้างต้นจะทำการสร้าง Model Object โดยใช้ Model file sports.mlt ที่มีอยู่แล้ว ซึ่ง file นี้จะสร้างมาจาก Model Import Wizard

ถ้าทำการสร้าง model เมื่อทำการออกแบบ application เราสามารถอ้างอิงถึง model ที่มีอยู่แล้วใน code ได้ดังแสดงตามตัวอย่าง

```
Dim oModel As ddoModel
Set oModel = oDataGroup.Model
```

เมื่อทำการบอกชื่อของ model file แล้ว Data Director จะทำการสร้าง Table, Column และ Relation Objects จากข้อมูลต่าง ๆ ที่อยู่ใน Model file

- การสร้าง Model Collection

จาก code ดังต่อไปนี้ จะทำการสร้าง collection สำหรับ 2 Models

```
Dim oEngine As ddoEngine
Dim oModel1 As ddoModel
Dim oModel2 As ddoModel
Set oEngine = New ddoEngine
OEngine.CreateModel("book.mlt")
OEngine.CreateModel("sports.mlt")
Set oModel1 = oEngine.Models(1)
Set oModel2 = oEngine.Models(2)
```

- การสร้าง DataGroup Object

หลังจากสร้าง Project Object เราสามารถสร้าง DataGroup object เพื่อทำการ access และใช้ข้อมูลในฐานข้อมูลในเวลา runtime

การสร้าง DataGroup Object ทำได้โดยการใช้ CreateDataGroup method ของ Project object ดังแสดงในตัวอย่าง ซึ่งใช้ทุก parameter ของ CreateDataGroup method

```
Dim oDataGroup As ddoDataGroup
Set oDataGroup = oProject.CreateDataGroup ("customer", "DGCustomers",
"c:\Program Files\Microsoft Visual Basic\sports.mlt")
```

code นี้จะสร้าง DataGroup object ชื่อว่า DGCustomers ซึ่งมาจาก model file sports.mlt โดยจะต้องบอก Table เหมือนเป็น Master table สำหรับ DataGroup นี้

ถ้ามีการสร้าง DataGroup ไว้แล้วเมื่อตอน design time ไปยัง Data Director user interface เราสามารถอ้างถึง DataGroup ที่มีอยู่แล้วได้ใน code ดังตัวอย่างต่อไปนี้ โดยส่งผ่านชื่อของ DataGroup ไปยัง DataGroup Collection ของ Engine Object

```
Dim oDataGroup As ddoDataGroup
Set oDataGroup = oProject.DataGroup("MyExistingDataGroup")
```

- Logging On เข้าไปยัง Databases

ในส่วนนี้จะนำเสนอ code ตัวอย่างซึ่งจะแสดงว่าจะทำการ setup LogonInfo object อย่างไร และ Log on ไปยัง Database โดยผ่าน Model หรือ DataGroup object สำหรับ Model Object เราจะเจาะจงข้อมูลต่าง ๆ ของการ log on ในการ import โครงสร้างของฐานข้อมูลมาเป็น Model เมื่อเวลา runtime เราจะกำหนดข้อมูลในการ log on สำหรับ DataGroup object เพื่อ access และ ใช้ข้อมูลในฐานข้อมูลผ่าน DataGroup

ในการ Log on นั้นสามารถทำได้หลายวิธี ได้แก่

- Set up logon parameters โดยใช้ LogonInfo Object
- ใช้ import method ของ Model object
- ใช้ Logon method ของ DataGroup Object โดยใช้ LogonInfo object หรือกำหนด Logon parameters โดยตรง

- การสร้าง LogonInfo Object

ในการกำหนดข้อมูลในการ Logon สำหรับฐานข้อมูล ใช้ CreateLogonInfo method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของ model หรือ DataGroup Object ในการ setup LogonInfo object

มีข้อสำคัญบางประการสำหรับการออกแบบซึ่งควรจำไว้เมื่อใช้ LogonInfo Object

- ถ้า log on ไปยังฐานข้อมูลโดยใช้ DataGroup object และไม่ส่งผ่านข้อมูลในการ Logon หรือไม่ได้ส่งผ่าน parameters ทั้งหมดสำหรับ LogonInfo Object แล้ว Data Director จะใช้ parameters เดียวกันกับ DataGroup ซึ่งใช้สำหรับ LogonInfo object เมื่อมีการ import model
- ถ้าใช้ค่า default ของข้อมูลจาก Model LogonInfo object เมื่อ log on ไปยัง DataGroup แล้ว Data Director จะไม่ให้ password เป็นค่า default
- ถ้าผ่านการ logon สำหรับ DataGroup ครั้งหนึ่งแล้ว Data Director จะใช้ข้อมูลเดิมนี้เป็นค่า default สำหรับการ Logon ครั้งต่อไป ซึ่งสามารถลบค่า default เหล่านี้ได้โดยการส่งผ่าน parameters ค่าใหม่ สำหรับ DataGroup logon ตามที่ต้องการ

ในการสร้าง LogonInfo object จะใช้ CreateLogonInfo method ของ Model object

หรือ DataGroup object

LogonInfo object จะมี parameters ดังต่อไปนี้

- varUser
- varPassword
- varServer
- varDatabase
- varDBMSName (อ้างอิง database driver โดยมากจะเป็น ODBC)
- varDatasource
- varName

ถ้าใช้ทุก parameters สำหรับ LogonInfo object แล้ว จะมี syntax ดังนี้

```
Dim oLogonInfo As ddoLogonInfo
Set oLogonInfo = oDataGroup.CreateLogonInfo("User", "Password",
"Server", "Database", "ODBC", "DataSource")
```

การ set up LogonInfo object สำหรับ DataGroup นั้นแสดงได้ดัง code ตัวอย่างซึ่งต้องมีการสร้าง object ต่าง ๆ ดังนี้ไว้แล้ว

- Engine object (DataDirector.Engine)
- Project object (sports.ddx)
- DataGroup object (DGCustomer)

สำหรับจุดประสงค์ของตัวอย่างนี้ ซึ่งมี username(admin), password(admin), database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(admin) , database(sports) , DBMS name (ODBC) , และ DataSource (sports) เมื่อ logon ไปยังฐานข้อมูลเมื่อเวลา runtime Data Director จะทำการ Log on อยู่เบื้องหลังโดยไม่มีกรขึ้น prompt

```
Dim oLogonInfo As ddoLogonInfo
Set oLogonInfo = oDataGroup.CreateLogonInfo ("admin", "admin", "sports"
, "ODBC", "sports")
```

การใช้ public function สำหรับ username และ password ซึ่งเราสามารถกำหนด public function เพื่อที่จะ set username และ password ใน LogonInfo object โดยจากตัวอย่างจะใช้ public function SetLogonInfo มันจะทำการตรวจสอบว่า LogonInfo object มีการสร้างขึ้นแล้วหรือยัง ถ้ายังมันจะสร้างขึ้นมา object หนึ่ง ถ้า object นั้นมีอยู่แล้ว code นี้จะ set username และ password parameters ไปยัง object นั้น

```
Public Function SetLogonInfo (szName As String , saPassword As String ,
oLogonInfo As object) As Boolean
On Error GoTo Error
If oLogonInfo Is Nothing Then
Set oLogonInfo = oDataGroup.CreateLogonInfo(szName , szPassword ,
"sports" , "ODBC" , "sports")
Else
oLogonInfo.User = szName
oLogonInfo.Password = saPassword
End If
SetLogonInfo = true
Exit Function
Error:
SetLogonInfo = false
End Function
```

- การทำ Actual Logon

ในการทำ Actual Logon operation จะใช้ Import method สำหรับ model object หรือ logon method สำหรับ DataGroup object

- การจับโครงสร้างของฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่ทำการสร้าง LogonInfo object ใช้ import method ของ Model object ทำการ import โครงสร้างของฐานข้อมูลมาเป็น model file เมื่อทำการ import database model แล้ว Data Director จะทำการ logon , import database และทำการ logoff โดยอัตโนมัติ

oModel.Import

ซึ่ง code นี้จะเอา parameters ที่ประกอบกันใน LogonInfo object และใช้มัน Logon ไปยัง Database โดยใช้ Import method จาก Model ชื่อ oModel

- การ logging on โดยผ่าน DataGroup

หลังจากที่เราสร้าง LogonInfo Object ใช้ Logon method ของ DataGroup object ในการ logon ไปยังฐานข้อมูล

oDataGroup.Logon

ซึ่ง code นี้จะเอา parameters ที่ประกอบกันใน LogonInfo object และใช้มัน Logon ไปยัง Database โดยใช้ Logon method จาก DataGroup ชื่อ oDataGroup

- การ Logging on โดยไม่ใช้ LogonInfo Object

เราสามารถทำการ Log on ไปยัง database โดยไม่ต้องใช้ LogonInfo Object ซึ่งเราจะกำหนด logon parameters โดยการให้ Logon method ของ DataGroup object ดังนี้

oDatagroup.Logon "MyName", "MyPassword"

เราสามารถที่จะกำหนด Server , Database , DBMS name และ Data source ได้ถ้าต้องการ สังเกตว่า method นี้ไม่ต้องใช้เครื่องหมาย ()

- การ Loggingon โดยไม่ใช้ model information ใด ๆ เลย

บางครั้งจำเป็นที่ต้องบังคับให้ Data Director logon โดยไม่ใช้ Model information ใด ๆ เลย เราสามารถทำได้โดย Set DataGroup property LogOnUsingModel เป็น False

การทำงานกับข้อมูล ในหัวข้อนี้จะรวมถึงตัวอย่าง โปรแกรมที่อธิบายการทำงานกับข้อมูล โดยใช้ การโปรแกรมอย่างเดี่ยวซึ่งรวมถึงการ สร้าง Datalinks, Querying Data และการแสดงข้อมูล

- การสร้าง Datalinks โดยใช้ DDO

DataLinks คือการเชื่อมต่อที่เชื่อมกันระหว่างcolumnของฐานข้อมูลและคอนโทรลของ visual basic และยังสามารถสร้างการเชื่อมต่อเสมือนซึ่งคิดค้กับcolumn แต่ไม่เชื่อมกับ คอนโทรลของ visual basic จริงๆ การเชื่อมต่อเสมือนนี้จะทำให้คุณสามารถquery ข้อมูลโดยใช้การ โปรแกรมอย่างเดี่ยว โดยปราศจากการแสดงข้อมูลในคอนโทรล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อคุณสร้าง DataLinks แล้วคุณสามารถตั้งค่า Data Path ของตารางและcolumn ได้ ซึ่ง Data Path นี้จะเป็นตัวระบุถึง ตารางเริ่มต้นที่จะเป็น

อย่างไรก็ตามเป็นไปได้ที่จะสร้าง Datalink กับ column ที่ไม่สามารถเข้าถึงได้จาก master table ซึ่ง column นี้จะถูก query เมื่อ datagroup ถูก query แต่จะไม่มีการจริงในตาราง

โดยใช้คำสั่ง CreateLink ของ DataGroup Object ดังนี้

```
Set DataLink = DataGroup.CreateLink _
  (VarColumn As Variant,[varControl As Variant]) As Object
```

การสร้าง DataPath มีสามวิธีที่สามารถระบุ datapath สำหรับ datalink ได้คือ

- ระบุ ตารางและ column คั้งเดิมของมัน
- ระบุถึงบางส่วนของpath และ ให้ data director อ้างถึงความสัมพันธ์บน model โดยตรง
- ระบุpath ทั้งหมดโดยเริ่มต้นจาก ตารางแม่

ตัวอย่างต่อไปนี้จะแสดงถึงวิธีสร้าง datalinks และ virtual datalink และยังคงจะแสดงวิธีที่ต่าง ๆ กัน

ในการset datapath อีกด้วย

การระบุถึงตารางและcolumn คั้งเดิม

```
Set olinkCustID = oDataGroup.CreateLink _
  ("customer.customer_num", form1.text)
```

การระบุถึง Path บางส่วน

```
Set olinkOrderIDD = oDataGroup.CreateLink _
  ("customer\orders.order_num",form1.text)
```

การระบุถึง Path ทั้งหมด

```
Set olinkCustID = oDataGroup.CreateLink _
  ("customer.customer_num\orders.customer_num", form1.text)
```

การสร้าง Virtual Links

คำสั่งต่อไปนี้จะทำการสร้าง Half Link ของ Column Customer_num ในตาราง customer

```
Set olnkCustID = oDataGroup.Create("customer.customer_num")
```

หรืออาจจะใช้ path แบบfull ก็ได้ดังนี้

```
Set olnkOrderID = oDataGroup.CreateLink("cutomer\orders.order_num")
```

การ Query ข้อมูล จะเกี่ยวข้องกับการดึงข้อมูลจากฐานข้อมูล โดยปกติเพื่อใช้ในการแสดงข้อมูล ใน หน้าจอโปรแกรม ด้วยคำสั่งของ DataDirector Object นั้น ข้อมูลส่วนใหญ่จะกระทำกับ ตารางและ ออปเจต DataGroup

การ Query ตาราง เราสามารถได้รับข้อมูล Query จาก Table, Record หรือ Datalink Object ได้ ตัวอย่างด้านล่างนี้จะแสดงถึงวิธีการ query ข้อมูลจาก ตาราง customer ใน oDataGroup object

```
Set QueryTable = odataGroup.Tables("customer")
QueryTable.QueryData
```

การ Query ข้อมูลในโหมด QBE มี 3 ขั้นตอนดังนี้
บอกให้ Data Director รู้ว่าเราต้องการ execute โดยใช้ QBE โดยใช้คำสั่ง

```
DataGroup.StartQBE = True
```

สถานะล่าสุดของ DataGroup จะถูกบันทึกเก็บไว้แล้วตารางนั้นๆจะเปลี่ยน โหมดเป็น โหมด QBE ทำการกำหนดเงื่อนไขของ QBE ด้วยคำสั่งต่อไปนี้

```
oDataGroup.StartQBE = True
oCustomers.Columns("Name").Value='A'
```

Excute ด้วยคำสั่ง

```
oDataGroup.StartQBE = True หรือคำสั่ง
oCustomers.Tables(index).QueryData
```

ผลลัพธ์ของการ Query จะอยู่ใน DataGroup และหลังจาก Query แล้ว โหมดของการ QBE จะอยู่ใน โหมด False อย่างอัตโนมัติ
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างต่อไปนี้จะแสดงถึงวิธี query ข้อมูลใน โหมด QBE โดยชื่อตาราง จะส่งผ่านมาจากตัวแปร szTableName

```
Public Function QueryCustomer(szTableName as string) as
Object
Dim oTable as object
On error resume next
oDataGroup.StartQBE = True
oDataGroup.Tables("customer").Columns("lname").Value =
szCustName
Set QueryCustomer = oDataGroup.Tables("customer")
QueryCustomer.QueryData
End Function
```

5.7 การส่งคำสั่ง SQL เข้าไป query โดยตรง

การสร้าง Virtual Table นั้นจะอนุญาตให้คุณสามารถส่งคำสั่ง SQL เข้าไปทำงานได้โดยตรงโดยผ่าน Datadirector Object และ virtual table ยังสามารถใช้ในการ adhocquery, subquery, group-by และ stored procedures

ตัวอย่างต่อไปนี้จะสร้าง virtual table ชื่อว่า oVTable และจะทำการ query คำสั่ง ด้วย

```
Function QueryCustomer(szName as String) as Object
Dim oVTable as object
Set oVTable = oDataGroup.CreateVirtualTable()
oVTable.QueryCommand =
"SELECT * from customers where name = ?"
oVTable.SetParam QUERY_CMD, 1, szName
oVTable.ExecuteQueryCommand
End Function
```

การแสดงผลข้อมูล ตัวอย่างต่อไปนี้จะแสดงวิธีการในการแสดงข้อมูลที่ query แล้วผ่านทาง วินโดวส์

```
Private Sub DisplayCustomer(Record As Object)
    tbCust_Id = Record.Columns("customer_num").Value
    tbName = Record.Columns("lname").Value
    tbContact = Record.Columns("fname").Value
    tbAddress1 = Record.Columns("address1").Value
    tbAddress2 = Record.Columns("address2").Value
    tbCity = Record.Columns("city").Value
    tbState = Record.Columns("state").Value
    tbZip = Record.Columns("postal_code").Value
    tbPhone = Record.Columns("phone").Value
End Sub
```

การ Set เงื่อนไข

ตัวอย่างต่อไปนี้จะสร้างเงื่อนไขของการ query จากพนักงานที่อาศัยอยู่ใน San Francisco เท่านั้น

```
oDataGroup.Tables("employee").ClearCondition
oDataGroup.Tables("employee").SetCondition "CITY =
'SAN FRANCISCO'"
oDataGroup.Tables("employee").QueryData
```

การเพิ่มเงื่อนไข

เราควรใช้ method Addcondition สำหรับ Table Object ในการเพิ่มเงื่อนไขของการ query แทน Setcondition เพราะว่า การ Setcondition นั้นจะทำการ set เงื่อนไขทับของเดิมที่มีอยู่

```
oDataGroup.Tables("employee").AddCondition
("POSITION = 'SALES REPRESENTATIVE'")
oDataGroup.Tables("employee").QueryData
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตั้งค่าลำดับของ record ในตาราง

เราสามารถกำหนดให้มีการเรียงลำดับของ record ตาม Column ใดๆก็ได้ และแบบ ASCENDING หรือ DESCENDING ก็ได้ดังนี้

```
oDataGroup.Tables("customer").Columns("customer_num").Sort =
ASCENDING
oDataGroup.Tables("customer").Columns("customer_num").
SortOrder = 1
```

การจัดการกับ Data Grid

ตัวอย่างต่อไปนี้แสดงถึงวิธีในการจัดการกับ ข้อมูล โดยผ่าน Data Grid

```
Private Sub DisplayOrders()
TDBGridOrders.Refresh
End Sub

Private Sub TDBGridOrders_UnboundReadData(ByVal RowBuf As
RowBuffer, StartLocation As Variant, ByVal ReadPriorRows As
Boolean)
Dim iCurRow As Integer ' The current row in the grid
Dim iRowCount As Integer ' The row count for this row buffer
Dim irowsFetched As Integer ' Number of rows fetched
Dim incr As Integer ' Direction of display in grid
Dim i As Integer
Dim j As Integer
If bQueried Then ' bQueried is global boolean
If bClearGrid Then ' bClearGrid is also global boolean
RowBuf.RowCount = 1
RowBuf.Value(1, 0) = ""
RowBuf.Value(1, 1) = ""
RowBuf.Value(1, 2) = ""
RowBuf.Value(1, 3) = ""
RowBuf.Value(1, 4) = ""
```

```

Exit Sub
End If
irowsFetched = 0
If ReadPriorRows Then incr = -1
Else incr = 1
End If
If IsNull(StartLocation) Then
If ReadPriorRows Then
iCurRow = iRowCount - 1
Else iCurRow = 0
End If
Else
iCurRow = StartLocation + incr
End If
oOrders.Tables("orders").LastRecord
iRowCount = oOrders.Records.Count - 1 ' oOrders is also global
RowBuf.RowCount = iRowCount ' Set number of rows into RowBuf
oOrders.Tables("orders").FirstRecord
Do
If iCurRow < 0 Or iCurRow >= iRowCount Then Exit Do
RowBuf.Value(i, 0) = oOrders.Columns("order_num").Value
RowBuf.Value(i, 1) =
oOrders.Columns("order_date").Value
RowBuf.Value(i, 2) =
oOrders.Columns("customer_num").Value
RowBuf.Value(i, 3) =
oOrders.Columns("ship_instruct").Value
RowBuf.Value(i, 4) = oOrders.Columns("backlog").Value
RowBuf.Value(i, 5) = oOrders.Columns("po_num").Value
RowBuf.Value(i, 6) = oOrders.Columns("ship_date").Value
RowBuf.Value(i, 7) = oOrders.Columns("ship_weight").Value
RowBuf.Value(i, 8) = oOrders.Columns("ship_charge").Value
RowBuf.Value(i, 9) = oOrders.Columns("paid_date").Value
RowBuf.Bookmark(i) = iCurRow

```

```

iCurRow = iCurRow + incr
irowsFetched = irowsFetched + 1
Loop While oOrders.NextRecord <> False
RowBuf.RowCount = irowsFetched
End If
End Sub

```

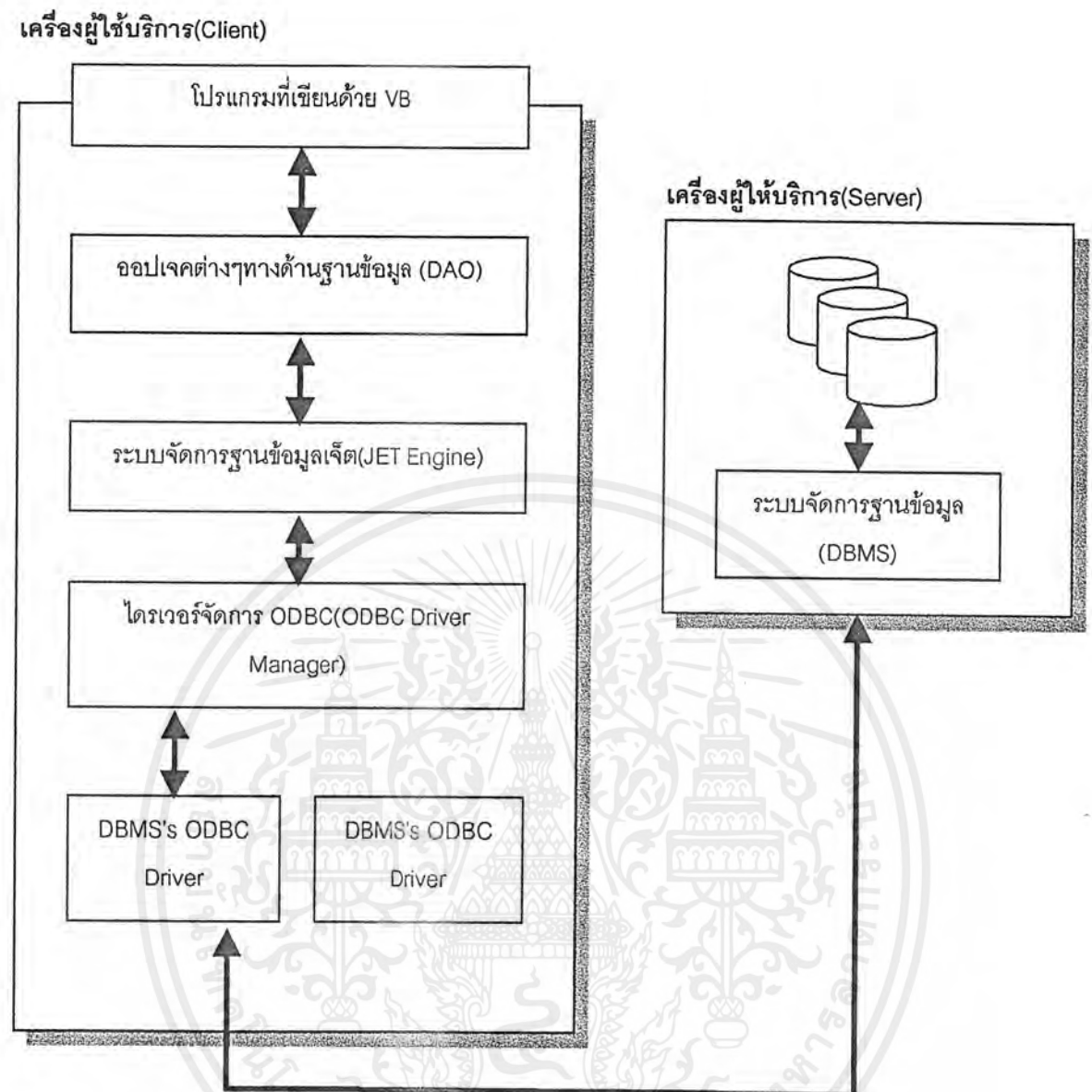
5.8 การติดต่อฐานข้อมูลโดยผ่านระบบจัดการแบบเจ็ต (JET Database Engine)

ระบบการจัดการฐานข้อมูล โดยใช้ JET นั้นทางบริษัท ไมโครซอฟท์ ได้นำมาใช้ในการจัดการฐานข้อมูลของ MS-Access เป็นหลัก โดยเริ่มคืบหน้าเอา JET Engine 1.1 มาใส่รวมไว้ในวิซวลเบสิกเวอร์ชัน 3.0 เป็นต้นมาช่วยให้วิซวลเบสิกสามารถติดต่อกับฐานข้อมูลได้หลากหลายขึ้น เช่น ฐานข้อมูล Ms_Access, dBase หรือ Foxpro เป็นต้นพร้อมๆกับการสร้าง ODBC(Open DataBase Connect) เพื่อให้เป็นมาตรฐานในการติดต่อกับฐานข้อมูลภายนอกบนเครื่องเซิร์ฟเวอร์(Server) อื่นๆเช่น SQL Server, Oracle, Informix และอื่นๆได้อีก

จากนั้นได้มีการปรับปรุงมาเป็น JET Engine เวอร์ชันต่างๆ ซึ่งได้เพิ่มความสามารถที่สำคัญจากเดิมอีกมากมายจนถึงเวอร์ชัน 3.51 ซึ่งมาพร้อมกับวิซวลเบสิกเวอร์ชัน 6.0

5.8.1 โครงสร้างของระบบจัดการฐานข้อมูลภายนอกผ่าน DAO และ JET

DAO(Data Access Object) เป็นตัวแปรชนิดพิเศษที่วิซวลเบสิกมีมาให้เพื่อใช้ติดต่อกับฐานข้อมูล เพื่อให้เกิดความเข้าใจในตัว DAO ง่ายขึ้น จึงจะขอเปรียบเทียบกับตัวแปร โดยทั่วไปเช่น ตัวแปรแบบ Integer หรือ String เป็นต้น กล่าวคือ ตัวแปร โดยทั่วไปจะใช้เก็บข้อมูลในหน่วยความจำแต่ DAO นี้สามารถใช้เพื่อเป็นตัวแทนของฐานข้อมูล ตาราง หรือ คิวรี ที่อยู่ในดิสก์ หรือใช้เก็บคิวรี ที่อยู่ในหน่วยความจำก็ได้ รวมทั้งสามารถตรวจสอบความผิดพลาดของข้อมูลได้มากกว่าที่ Database Engine มีมาให้ และยังสามารถค้นหาข้อมูลที่ต้องการได้โดยไม่ต้องแสดงบนหน้าจอ เช่น การค้นหาข้อมูลราคาสินค้าจากตาราง Product เป็นต้น



รูปที่ 11 แสดงการทำงานกับระบบจัดการฐานข้อมูลภายนอกผ่าน DAO และ JET

5.8.2 องค์ประกอบต่างๆของโครงสร้างข้อมูลใน JET

1. TableDefs Collection เป็นองค์ประกอบหนึ่งที่เก็บข้อมูลของโครงสร้างหรือองค์ประกอบของตารางต่างๆ ในฐานข้อมูลนั้นซึ่งแต่ละตารางในฐานข้อมูลเกี่ยวกับโครงสร้างของตารางนั้นที่ใช้ในการอธิบายรายละเอียดถึงโครงสร้างของตารางว่าประกอบไปด้วยฟิลด์อะไรบ้าง มีทั้งหมดกี่ฟิลด์ แต่ละฟิลด์จะเก็บข้อมูลในลักษณะใด เป็นตัวเลขหรือตัวอักษร เป็นต้นโดยข้อมูลของโครงสร้างของตารางจะประกอบไปด้วยองค์ประกอบย่อยอีก องค์ประกอบคือ

- 1.1 Fields Definition เป็นส่วนที่เก็บรายละเอียดว่า ตารางนั้นๆ ประกอบไปด้วยฟิลด์อะไรบ้างและแต่ละฟิลด์มีลักษณะเป็นอย่างไร เช่นเป็นตัวเลข หรือตัวอักษร หรืออื่นๆ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 Indexes Definition หากตารางนั้นมีครรรชนีประกอบเพื่อช่วยในการค้นหาตำแหน่งของข้อมูลในตารางทำได้เร็วขึ้นข้อมูลอธิบายหรือกำหนดลักษณะของครรรชนีต่างๆ ของตารางจะเก็บไว้ในส่วนนี้ เช่นครรรชนีมีชื่อเรียกว่าอะไร ประกอบด้วยฟิลด์อะไรบ้าง เป็นครรรชนีที่มีค่าซ้ำกันได้หรือไม่

2. QueryDefs Connection จะเป็นส่วนหรือองค์ประกอบที่เก็บโปรแกรมย่อยที่ใช้ในการทำงานกับข้อมูลในฐานะข้อมูล(สำหรับระบบจัดการฐานข้อมูลโดยทั่วไปจะเรียกว่า Stored Procedure แต่ในฐานะข้อมูลของเจ็ต จะใช้ศัพท์ว่า QueryDef) สำหรับการเรียกใช้ในภายหลัง จะเป็นคำสั่งใดๆของ SQL ก็ได้ไม่ว่าเป็นคำสั่ง Select, Insert ,Delete หรือ Update ซึ่งโดยมากจะใช้ในการเก็บคำสั่งที่ใช้บ่อยๆเพื่อเป็นการแยกคำสั่งที่ใช้ในการจัดการกับฐานข้อมูล ออกจากตัวโปรแกรม เป็นการผูกคำสั่งในการจัดการกับฐานข้อมูลเข้าไว้กับฐานข้อมูลนั้น ช่วยลดความยุ่งยากของการโปรแกรมได้เป็นอย่างดี แต่ชุดคำสั่งที่ใช้ได้ใน QueryDef ตำกรับระบบจัดการฐานข้อมูลเจ็ตนี้จะไม่ซับซ้อนหรือมีความสามารถมากมายเหมือนกับระบบจัดการฐานข้อมูลขนาดใหญ่ๆ

3. Relations Connlection เป็นส่วนหรือองค์ประกอบที่เก็บข้อมูลที่อยู่อธิบายถึงโครงสร้างความสัมพันธ์(Relation) ระหว่างตารางในฐานะข้อมูล โดยแต่ละความสัมพันธ์จะมีองค์ประกอบที่ใช้เก็บข้อมูลของความสัมพันธ์หนึ่งองค์ประกอบ ซึ่งจะอธิบายว่าความสัมพันธ์นั้นเป็นความสัมพันธ์ระหว่างตารางใดกับตารางใด และมีฟิลด์ใดบ้างที่ประกอบกันเป็นความสัมพันธ์นั้น โดยที่ระบบจัดการฐานข้อมูลเจ็ต จะใช้ข้อมูลในความสัมพันธ์เหล่านี้เป็นส่วนหนึ่งใจการระวังรักษาความถูกต้องให้กับข้อมูลในฐานะข้อมูลสำหรับประเภทของฐานข้อมูลขึ้นมาใช้งานแบ่งได้เป็น

- ฐานข้อมูลตระกูล xBase ทั้งหมดเช่น dBase และ FoxPro เป็นต้น
- ฐานข้อมูลของ Excel (.xls)
- ฐานข้อมูลภายนอก เช่น Informix, Oracle, SQL Server และอื่นๆผ่านทาง ODBC(Open Database Connect)

สำหรับในรายงานฉบับนี้จะอธิบายเฉพาะส่วนของการทำงานกับฐานข้อมูลภายนอกผ่าน ODBC เท่านั้น เพราะเป็นส่วนที่มีความเกี่ยวข้องกับรายงานฉบับนี้

5.8.3 การสร้างพื้นที่ใช้งานระบบจัดการฐานข้อมูล

โดยปกติแล้วการทำงานกับฐานข้อมูลทุกอย่างในวิซวลเบสิก จะถูกจัดการ โดยระบบจัดการฐานข้อมูล หรือ JET ดังนั้นโปรแกรมที่ต้องการทำงานกับฐานข้อมูลจะต้องกำหนดรายละเอียดและวิธีติดต่อกับระบบจัดการข้อมูลเสียก่อน ซึ่งทำได้โดยการสร้างออปเจ็คพื้นฐานที่ทำงานกับระบบจัดการฐานข้อมูล (Workspace Object) โดยปกติแล้ววิซวลเบสิกจะสร้างพื้นที่ทำงานกับระบบจัดการฐานข้อมูลขึ้นมาให้โดยอัตโนมัติ แล้วหนึ่งออปเจ็ค ที่เป็นมาตรฐานสำหรับการทำงานกับฐานข้อมูลทั่วไป

เราสามารถเรียกออปเจ็คพื้นที่ทำงานกับระบบจัดการฐานข้อมูลที่เป็นมาตรฐานนี้ขึ้นมาใช้ได้เลย โดยการอ้างถึงออปเจ็คลำดับแรกในออปเจ็คกลุ่มของออปเจ็คพื้นที่ทำงานกับระบบจัดการฐานข้อมูล (Workspace Collection) ดังในตัวอย่างนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim wksDefault As Workspace
Set wksDefault = DBEngine.Workspaces(0)
```

5.8.4 วิธีเปิดฐานข้อมูลที่ต้องการ

หลังจากได้ออปเจ็คพื้นที่ทำงานกับระบบจัดการฐานข้อมูลแล้ว ก่อนจะสามารถอ่านโครงสร้างหรือจัดการใดๆกับข้อมูลนั้นได้ ฐานข้อมูลที่ต้องการนั้นจะต้องถูกเปิด (Open) ก่อนในลักษณะเช่นเดียวกับการเปิดเพิ่มข้อมูลปกติทั่วไปก่อนที่จะทำการอ่านหรือเขียนเพิ่มข้อมูลนั้น

การเปิดฐานข้อมูลขึ้นมาเพื่อใช้งาน ทำได้โดยการเรียกใช้ฟังก์ชันการทำงาน Open Database จากออปเจ็คพื้นที่ทำงานกับฐานข้อมูลแล้วกำหนดค่าที่ได้ให้กับตัวแปรออปเจ็คฐานข้อมูล โดยมีรูปแบบของการเรียกใช้ดังนี้

```
Set database_obj = workspace_obj.OpenDatabase (dbname, exclusive, readonly, connect)
```

โดยการกำหนดค่าของ dbname และ connect ดังนี้
 dbname ให้กำหนดค่าเป็น "" หรือค่าคงที่อักษรที่มีความยาวเป็นศูนย์
 connect ให้ระบุตามรูปแบบดังต่อไปนี้

```
"ODBC;DSN=datasource;DATABASE=dbname;UID=user;PWD=passwd;"
```

โดยที่	
datasource	คือชื่อของ Data Source Name ของที่ได้กำหนดไว้ใน ODBC
dbname	เป็นชื่อของฐานข้อมูลที่ต้องการทำงานด้วย จะระบุในกรณีที่ต้องการจะให้เปิดฐานข้อมูลอื่นที่ต่างจากที่กำหนดไว้ใน Data Source ที่เรียกใช้
user	เป็นชื่อของผู้ใช้ เพื่อให้ระบบจัดการฐานข้อมูลนั้นตรวจสอบสิทธิการใช้งาน เปรียบเหมือนการ login เข้าสู่ระบบจัดการฐานข้อมูลนั้น
passwd	เป็นรหัสผ่านสำหรับผู้ใช้ เพื่อให้ระบบจัดการฐานข้อมูลตรวจสอบ

5.8.5 ตรวจสอบและค้นหาตารางในฐานข้อมูล

การตรวจสอบค้นหาว่าในฐานข้อมูลประกอบไปด้วยตารางอะไรบ้างนั้น ทำได้โดยการเข้าไปค้นหาในออปเจ็คข้อกำหนดตาราง (TableDefs Connection) ของออปเจ็คฐานข้อมูล ดังนั้นจึงควรที่จะทำความเข้าใจออปเจ็คนี้ให้ดี

5.8.5.1. ออปเจ็คของกลุ่มออปเจ็คข้อกำหนดตาราง (TableDefs Collection) เป็นออปเจ็คที่จะเก็บข้อมูลที่แสดงว่าฐานข้อมูลนั้นๆ ประกอบด้วยตารางอะไรบ้างในออปเจ็คฐานข้อมูลหนึ่งๆ จะมีออปเจ็คข้อ

กำหนดตารางเพียงหนึ่งออปเจ็กเท่านั้น ออปเจ็กข้อกำหนดตารางจะถูกจัดเก็บเรียงตามลำดับในลักษณะเช่นเดียวอะเรย์ อยู่ภายในออปเจ็กกลุ่มนี้ ซึ่งมีวิธีการกำหนดตัวแปรออปเจ็กข้อกำหนดตารางดังนี้

```
Dim tabledefs_object As TableDefs
```

5.8.5.2.ออปเจ็กข้อกำหนดตาราง(TableDef Object) เป็นออปเจ็กย่อยที่ประกอบอยู่ในออปเจ็กกลุ่มของออปเจ็กข้อกำหนดตาราง โดยที่ทุกๆตารางในฐานข้อมูล จะมีออปเจ็กข้อกำหนดตารางหนึ่งออปเจ็ก เพื่อใช้บอกรายละเอียดเกี่ยวกับตารางนั้น เช่นประกอบไปด้วยฟิลด์หรือคอลัมน์อะไรบ้าง ซึ่งมีวิธีการกำหนดตัวแปรออปเจ็กข้อกำหนดตารางดังนี้

```
Dim tabledef_object As TableDef
```

ตรวจสอบและค้นหาฟิลด์ในตาราง

เมื่อเราค้นหาตารางที่ต้องการได้แล้ว หากต้องการจะตรวจสอบว่าตารางที่สนใจนั้นประกอบด้วยฟิลด์อะไรบ้างและเป็นฟิลด์สำหรับข้อมูลประเภทใด สามารถทำได้โดยการค้นหาใจออปเจ็กกลุ่มของออปเจ็กข้อกำหนดฟิลด์(Field Collection) ที่อยู่ในออปเจ็กข้อกำหนดตารางที่สนใจ

5.8.5.3.ออปเจ็กกลุ่มของออปเจ็กข้อกำหนดฟิลด์(Field Collection)

ทุกตารางจะประกอบด้วยฟิลด์หรือคอลัมน์ ดังนั้นในแต่ละออปเจ็กข้อกำหนดตารางจะมีออปเจ็กของออปเจ็กข้อกำหนดฟิลด์(Field Collection) และในออปเจ็กข้อกำหนดเท่านั้น ในลักษณะเช่นเดียวกับที่ออปเจ็กกลุ่มของออปเจ็กข้อกำหนดตารางที่ได้อธิบายข้างต้น การกำหนดตัวแปรออปเจ็กของออปเจ็กข้อกำหนดฟิลด์ทำได้ดังนี้

```
Dim fields_object As Fields
```

โดยที่ `fields_object` คือตัวแปรที่ต้องการกำหนดให้เป็นประเภทออปเจ็กกลุ่มของออปเจ็กข้อกำหนดฟิลด์

5.8.5.4.ออปเจ็กข้อกำหนดฟิลด์(Field Object)

เป็นออปเจ็กย่อยที่ประกอบอยู่ในออปเจ็กกลุ่มของออปเจ็กข้อกำหนดฟิลด์ โดยที่ทุกๆฟิลด์ของตารางนั้นๆจะมีออปเจ็กข้อกำหนดฟิลด์หนึ่งออปเจ็ก เพื่อใช้บอกรายละเอียดเกี่ยวกับฟิลด์นั้น เช่น เป็นฟิลด์สำหรับข้อมูลประเภทใด มีขนาดเท่าใด เป็นต้น การกำหนดตัวแปรออปเจ็กข้อกำหนดฟิลด์

```
Dim field_object As Field
```

5.8.6 ออปเจ็ทข้อมูล(RecordSet Object)

ออปเจ็ทข้อมูล หมายถึงกลุ่มของข้อมูลที่สนใจหรือต้องการจะใช้งานเช่นตาราง หรือส่วนของตาราง หรืออาจเป็นข้อมูลที่มาจากรายมากกว่าหนึ่งตารางรวมกันได้ ขึ้นอยู่กับขั้นตอนที่กำหนดค่าให้กับออปเจ็ทข้อมูล การจัดการ และทำงานกับข้อมูลในฐานะข้อมูล ก็คือการจัดการ และทำงานกับข้อมูลที่อยู่ใ้ออปเจ็ทข้อมูลต่าง ๆ นั้นเอง

ออปเจ็ทข้อมูลแบ่งออกเป็นประเภทที่แตกต่างกันได้ 3 ประเภทหลักๆแต่ละประเภทก็จะเหมาะสมกับการใช้งานลักษณะต่างๆกัน ซึ่งพอจะสรุปได้ดังนี้

1. ออปเจ็ทข้อมูลประเภทตาราง(Table Type RecordSet Object) ออปเจ็ทข้อมูลประเภทตารางนี้จะเก็บข้อมูลทั้งหมดทุกๆเรคอร์ดและทุกๆฟิลด์ของตารางที่สนใจ เช่นข้อมูลในตาราง Student ทั้งหมดเป็นต้น
2. ออปเจ็ทข้อมูลประเภทไดนาเซ็ท(DynaSet Type RecordSet Object) เป็นออปเจ็ทข้อมูลที่มีข้อมูลทั้งหมดจากราย ข้อมูลบางส่วนของตาราง หรือข้อมูลจากรายหลายๆตารางมารวมกันก็ได้ และค่าเสนอให้เข้าใจว่าเป็นอีกตารางหนึ่ง คำว่า DynaSet ในวิซวลเบสิกก็มีลักษณะเช่นเดียวกับ View ที่พบในระบบจัดการข้อมูลขนาดใหญ่ทั่วไป
3. ออปเจ็ทข้อมูลประเภทสแน็ปช็อต(SnapShot Type RecordSet Object) สแน็ปช็อตจะมีลักษณะเหมือนกับออปเจ็ทข้อมูลประเภทไดนาเซ็ททุกประการทั้งในเรื่องของการสร้างและการนำไปใช้งาน ยกเว้นแต่เพียงว่าสามารถอ่านหรือค้นหาข้อมูลที่อยู่ใ้ออปเจ็ทข้อมูลประเภทสแน็ปช็อตได้เพียงอย่างเดียวเท่านั้น ไม่สามารถเพิ่มเติม ลบ หรือแก้ไขได้

5.8.7 ขั้นตอนและวิธีการต่างๆที่จะใช้งานออปเจ็ทข้อมูลมีดังนี้

กำหนดตัวแปร

```
Dim recordset_object As RecordSet
```

กำหนดค่าให้กับตัวแปรออปเจ็ทข้อมูล

```
Set recordset_object = db_object.OpenRecordSet (source [,type [,options [,lockedits]])
```

โดยที่

recordset_object เป็นตัวแปรประเภทออปเจ็ทข้อมูลที่ต้องการจะกำหนดค่าให้

db_object เป็นตัวแปรประเภทออปเจ็ทฐานข้อมูลที่ได้ถูกกำหนดค่าหรือเปิดไว้แล้ว

source เป็นตัวแปรหรือค่าคงที่ที่เป็นตัวอักษร เพื่อใช้ในการระบุถึงแหล่งของข้อมูลสำหรับกำหนดให้ออปเจ็ทข้อมูลนี้มีค่าเป็นได้ 3 กรณี คือดังนี้

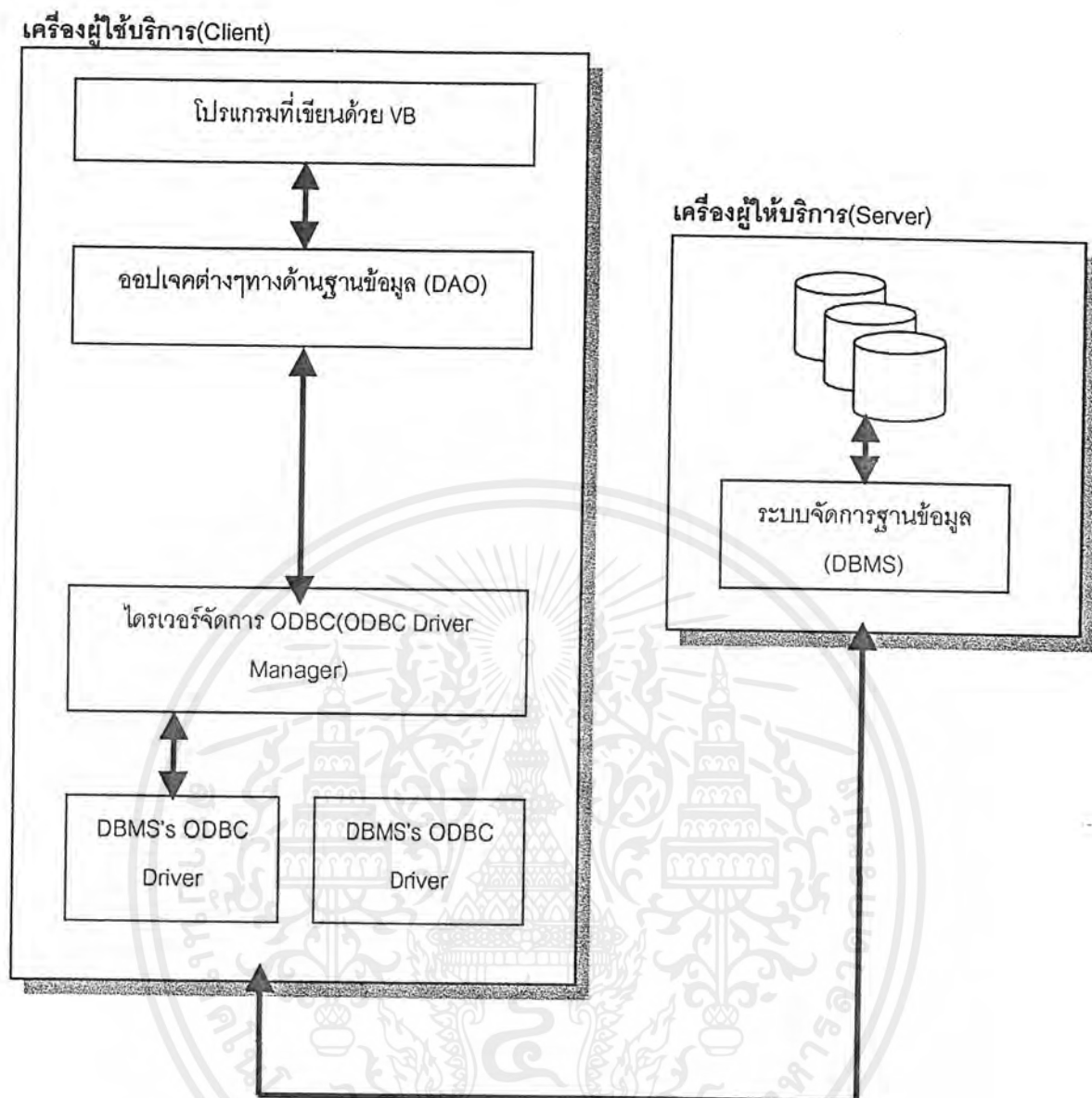
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

source	<p>เป็นตัวแปรหรือค่าคงที่ที่เป็นตัวอักษร เพื่อใช้ในการระบุถึงแหล่งของข้อมูลสำหรับกำหนดให้ออปเจ็คข้อมูลนี้มีค่าเป็นได้ 3 กรณีคือดังนี้</p> <ol style="list-style-type: none"> 1. เป็นชื่อของตาราง ในกรณีที่ต้องการจะให้ออปเจ็คข้อมูลนี้ที่จะกำหนดค่านี้หมายถึงข้อมูลทั้งหมดในตารางนั้นๆ 2. เป็นคำสั่ง select ของ SQL ในกรณีที่ต้องการจะให้ออปเจ็คข้อมูลนี้มีข้อมูลตามผลลัพธ์ที่ได้จากคำสั่ง Select นี้ 3. เป็นชื่อของโปรแกรมย่อยในฐานข้อมูล(QueryDef) ในฐานข้อมูลนี้มีข้อมูลตามผลลัพธ์ที่ได้จากคำสั่ง Select นี้
type	<p>เป็นค่าคงที่ที่ใช้ในการระบุถึงประเภทของออปเจ็คข้อมูลที่ต้องการจะกำหนดขึ้น มีรายละเอียดดังนี้</p>

5.8.8 การติดต่อระบบจัดการฐานข้อมูลภายนอกโดยตรง(ODBC Direct)

อย่างที่กล่าวไปแล้วว่า โดยปกติแล้วในการทำงานกับฐานข้อมูลต่างๆ วิศวกรเบสิกจะเรียกกระบวนการจัดการฐานข้อมูลเจ็ด ขึ้นมาจัดการให้ทั้งหมด นั่นเป็นการสิ้นเปลืองที่เกิดขึ้นในทรัพยากรของระบบ เพราะต้องมีขั้นตอนของการตรวจสอบคำสั่ง และเปลี่ยนคำสั่งให้เหมาะสมกับระบบจัดการฐานข้อมูลนั้นอีกทีหนึ่ง และไม่สามารถจะเรียกใช้ความสามารถต่างๆของระบบจัดการฐานข้อมูลภายนอกนั้นได้อย่างเต็มที่ เพราะว่าถูกจำกัดอยู่ด้วยความสามารถของคำสั่ง SQL ของระบบจัดการฐานข้อมูลเจ็ด

จึงทำให้วิศวกรเบสิกได้เพิ่มความสามารถในการส่งคำสั่ง SQL ไปให้กับระบบจัดการฐานข้อมูลภายนอกโดยตรง ด้วยการเรียกใช้ฟังก์ชันการทำงานพร้อมกำหนดค่าตัวเลือกสำหรับลักษณะการทำงานเป็น dbPassThrough แต่ก็ยังเป็นเพียงจุดเดียวที่ใช้ได้ ดังแสดงในรูป



รูปที่ 12 แสดงการทำงานกับระบบจัดการฐานข้อมูลภายนอกโดยตรง

การเรียกใช้ออปเจตต่างๆในการจัดการฐานข้อมูลภายนอกโดยตรงจะใช้ RDO ซึ่งก็คือ ออปเจตต่างๆทางด้านฐานข้อมูลภายนอก (Remote Data Object หรือ RDO) ซึ่งมีสองวิธีคือ

1. การเรียกใช้ออปเจตต่างๆทางด้านฐานข้อมูลภายนอกโดยตรง
2. การเรียกใช้ผ่านออปเจตต่างๆทางด้านฐานข้อมูล(DAO)

5.8.9 ออปเจตทางด้านฐานข้อมูลสำหรับการใช้งานแบบ ODBC Direct

1. ออปเจตพื้นที่ทำงานกับระบบจัดการฐานข้อมูลภายนอก (Workspace Object) อย่างที่ได้อธิบายไปในบทก่อนๆแล้วว่า การทำงานกับฐานข้อมูลภายนอกทั้งหมดจะต้องเริ่มต้นจากออปเจตพื้นที่ทำงานกับระบบจัดการฐานข้อมูลนี้ทั้งนั้น หากต้องการกำหนดชนิดให้กับออป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เจ็ทพื้นที่ทำงานกับระบบจัดการฐานข้อมูลเป็นประเภทพื้นที่ทำงานกับระบบจัดการฐานข้อมูลผ่าน ODBC โดยตรง นั้นทำได้สองวิธี

-วิธีแรก กำหนดค่าเริ่มต้นที่คุณสมบัติชื่อ DefaultType ของออปเจ็ทจัดการฐานข้อมูลให้มีค่าเป็น dbUseODBC ตัวอย่างเช่น

```
Dim wksODBC As Workspace
DbEngine.DefaultType = dbUseODBC
Set wksODBC = DBEngine.Workspaces(0)
```

-วิธีสอง กำหนดประเภทในขั้นตอนกำหนดสร้างออปเจ็ทพื้นที่ทำงานกับระบบจัดการฐานข้อมูล ด้วยฟังก์ชันการทำงาน CreateWorkspace ซึ่งมีรูปแบบของการเรียกใช้ดังนี้

```
Set workspace_obj = CreateWorkspace(name, user, password, type)
```

Workspace_obj: เป็นตัวแปรประเภทออปเจ็ทพื้นที่ทำงานติดต่อกับระบบจัดการฐานข้อมูลที่ต้องการจะกำหนดค่าให้

Name: เป็นตัวแปรหรือค่าคงที่ประเภทตัวอักษรสำหรับกำหนดชื่อให้กับออปเจ็ทที่จะกำหนดสร้างนี้

User: เป็นตัวแปรหรือค่าคงที่ประเภทตัวอักษรสำหรับกำหนดค่าเริ่มต้นของชื่อผู้ใช้ให้กับออปเจ็ทนี้เพื่อใช้ในการติดต่อกับระบบฐานข้อมูลในภายหลัง

Password: เป็นตัวแปรหรือค่าคงที่ประเภทตัวอักษรสำหรับกำหนดค่าเริ่มต้นของรหัสผ่านของผู้ใช้นี้เพื่อใช้ในการติดต่อกับระบบฐานข้อมูลในภายหลัง

Type: สำหรับระบุประเภทของออปเจ็ทที่จะกำหนดสร้างนี้ โดยมีค่าเป็น dbUseJet และ dbUseODBC

5.8.10 ออปเจ็ทการติดต่อ (Connection Object)

สำหรับการทำงานกับฐานข้อมูลภายนอกผ่าน ODBC โดยตรงนี้จะไม่มีการใช้ออปเจ็ทฐานข้อมูล (Database Object) แต่จะใช้ออปเจ็ทการติดต่อ (Connection Object) ซึ่งหมายถึงลักษณะของการติดต่อกับฐานข้อมูลภายนอก โดยมีความหมายเปรียบได้กับออปเจ็ทฐานข้อมูล (Database Object) ในออปเจ็ทพื้นที่ทำงานกับระบบจัดการข้อมูล ประเภทเจ็ทออปเจ็ทการติดต่อนี้จะอยู่รวมกันภายในออปเจ็ทกลุ่มของออปเจ็ทการติดต่อ (Connections Collection) ภายในออปเจ็ทพื้นที่ทำงานกับระบบจัดการฐานข้อมูลผ่าน ODBC โดยตรง (ODBCDirect Workspace Object) เช่นเดียวกับออปเจ็ททางค้ำฐานข้อมูลอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบของการเรียกใช้ฟังก์ชันการทำงาน OpenConnection มีดังนี้

```
set connection_obj = workspace_obj.OpenConnection(name, options, readonly, connect)
```

โดยที่

- connection_obj** เป็นตัวแปรประเภทออปเจ็กต์ที่ทำงานกับระบบจัดการฐานข้อมูลผ่าน ODBC โดยตรง (ODBCDirect Workspace Object) ที่ได้มีการกำหนดค่าให้เรียบร้อยแล้ว
- name** เป็นตัวแปรหรือค่าคงที่ประเภทตัวอักษรที่ใช้ระบุชื่อของออปเจ็กต์ข้อกำหนดการติดต่อ (Connection Object) ที่จะกำหนดสร้างนี้
- options** เป็นค่าคงที่ที่ใช้ระบุลักษณะพิเศษเพิ่มเติมของออปเจ็กต์การติดต่อ(Connection Object)นี้
- readonly** เป็นตัวแปรหรือค่าคงที่ทางตรรกะ(Boolean) เพื่อใช้ระบุการทำงานผ่านออปเจ็กต์นี้เป็น การอ่านข้อมูลอย่างเดียวหรือเปลี่ยนแปลงแก้ไขข้อมูลก็ได้ โดยหากมีค่าเป็น true หมายถึง ต้องการทำงานในลักษณะอ่านอย่างเดียว
- Connect** เป็นตัวแปรหรือค่าคงที่ประเภทตัวอักษรที่ใช้ระบุประเภทของฐานข้อมูลซึ่งกรณีนี้จะ เป็นประเภท ODBC ได้เพียงประการเดียว พร้อมทั้งข้อมูลประกอบเพิ่มเติมเช่น ชื่อของ ODBC ที่ได้กำหนดไว้เพื่ออ้างถึงฐานข้อมูล และตำแหน่ง รวมทั้งชื่อ และรหัสของผู้ใช้ ที่ให้แสดงตนเข้าสู่ระบบจัดการฐานข้อมูลภายนอก เป็นต้น

5.8.11 ออปเจ็กต์ข้อมูล(Recordset Object)

ในแต่ละออปเจ็กต์ข้อกำหนดการติดต่อ(Connection Object) นั้นอาจจะมีการกำหนดใช้ออปเจ็กต์ข้อมูล(Recordset Object) ซึ่งเป็นที่เก็บข้อมูลที่สนใจที่ได้เลือกค้นหาจากออปเจ็กต์ข้อมูล(Recordset Object) ซึ่งเป็นที่เก็บข้อมูลที่สนใจที่ได้เลือกค้นหาจากตารางต่างๆในฐานข้อมูล ได้มากกว่าหนึ่งออปเจ็กต์ มีออปเจ็กต์ข้อมูลบางประเภทใช้ได้กับเฉพาะการทำงานกับระบบจัดการฐานข้อมูลเจ็ด และมีบางประเภทใช้ได้กับเฉพาะการทำงานแบบ ODBCDirect เท่านั้น

การเรียกใช้ฟังก์ชันการทำงาน OpenRecordset ของออปเจ็กต์การติดต่อ โดยมีรูปแบบของการเรียกใช้ดังนี้

```
Set recordset_obj = object.OpenRecordset(source,type ,option, lockedits)
```

5.8.12 เปรียบเทียบข้อดีข้อเสียของระบบจัดการฐานข้อมูลแบบต่างๆ

ข้อเสียของการใช้การติดต่อผ่านระบบจัดการฐานข้อมูลแบบเจ็ต

- สิ้นเปลืองการใช้ทรัพยากรของระบบ เช่น หน่วยความจำ
- สิ้นเปลืองเวลาที่ต้องใช้ในการเปลี่ยนคำสั่งในการทำงานกับฐานข้อมูลไปเป็นคำสั่งที่เหมาะสมกับระบบจัดการฐานข้อมูลภายนอกนั้น
- ไม่สามารถจะใช้ประโยชน์ในความสามารถอื่นๆของระบบจัดการฐานข้อมูลนั้นได้อย่างเต็มที่
- วิศวกรเบสิกไม่อนุญาตให้สร้างออปเจ็คข้อกำหนดของโปรแกรมย่อยในฐานข้อมูล (QueryDef Object) เนื่องจากความสามารถที่แตกต่างกันของโปรแกรมย่อยในฐานข้อมูล (Stored Procedure) ของระบบจัดการฐานข้อมูลแต่ละระบบ

ข้อดีของการใช้การติดต่อผ่านระบบจัดการฐานข้อมูลแบบเจ็ต

- ช่วยให้นักพัฒนาโปรแกรมทั้งหลายสามารถใช้รูปแบบและคำสั่งต่างๆสำหรับการจัดการฐานข้อมูลเหมือนกันไม่ว่าจะทำงานกับฐานข้อมูลใด
- เพิ่มความยืดหยุ่นต่อการเปลี่ยนแปลงฐานข้อมูลที่ใช้ในโปรแกรม (Portability) ของโปรแกรมดีขึ้น
- ลดระยะเวลาในการพัฒนาโปรแกรมลงได้เนื่องจากใช้ภาษาที่เหมือนกัน

ข้อเสียของการใช้การติดต่อผ่านระบบจัดการฐานข้อมูลภายนอกโดยตรง

- จะเสียความยืดหยุ่นต่อการเปลี่ยนแปลงฐานข้อมูลที่ใช้ในโปรแกรม (Portability) ของโปรแกรมไป

ข้อดีของการใช้การติดต่อผ่านระบบจัดการฐานข้อมูลภายนอกโดยตรง

- ประหยัดการใช้พื้นที่ในหน่วยความจำ เพราะจะไม่มีกรนำระบบจัดการฐานข้อมูลเจ็ตไปเก็บไว้ในหน่วยความจำของระบบ หากไม่มีการใช้หรือทำงานกับฐานข้อมูลแบบเจ็ตด้วยพร้อมกัน
- เพิ่มความสามารถในการทำงานกับฐานข้อมูลในลักษณะที่ไม่ต้องรอให้ทำงานเสร็จเสียก่อนหรือที่เรียกว่าเป็นการทำงานแบบ Asynchronous ช่วยให้ระหว่างที่ระบบจัดการฐานข้อมูลกำลังทำงานตามคำสั่งอยู่นั้น โดยเฉพาะอย่างยิ่งคำสั่งที่ต้องเสียเวลาอย่างมาก เราสามารถจะให้โปรแกรมไปทำงานอย่างอื่นก่อนโดยไม่หยุดรอผลลัพธ์ของคำสั่ง จึงทำให้โปรแกรมสามารถจะทำงานได้มากขึ้น
- อนุญาตให้มีการกำหนดสร้างออปเจ็คข้อกำหนดของโปรแกรมย่อยในฐานข้อมูล (QueryDef Object) ขึ้นมาใช้งานได้ และสามารถจะใช้ออปเจ็คนี้เรียกโปรแกรมย่อยในฐานข้อมูล (Stored Procedure) ที่อยู่บนฐานข้อมูลภายนอกนั้นทำได้ พร้อมทั้งการรับส่งพารามิเตอร์ (Parameter) ระหว่างโปรแกรมกับโปรแกรมย่อยในฐานข้อมูล (Stored Procedure) โดยที่ไม่สามารถทำในระบบฐานข้อมูลเจ็ต (JET)

บทที่ 6

บทสรุป

6.1 ข้อเสนอแนะ และ แนวทางการศึกษาต่อ

จากการศึกษาและทำวิจัยในโครงการ Temporal Database ทำให้ทราบถึงปัญหาและแนวทางการแก้ไข ซึ่งสามารถรวบรวมเป็นข้อเสนอแนะได้ดังนี้

1. จากการที่ผู้ทำวิจัยได้ทดลองเขียนโปรแกรมติดต่อกับฐานข้อมูล โดยใช้วิธีการในการเชื่อมต่อแบบต่าง ๆ พบว่า ในการติดต่อกับการใช้ Informix Data Director for visual basic version 3.5 นั้นมีปัญหาในการ Save ข้อมูลที่ถูกแก้ไข ซึ่งจะ compile ผ่านเมื่ออยู่ในโปรแกรม Visual Basic แต่เมื่อ compile เป็นจุด exe แล้วไม่สามารถ run ได้ ซึ่งทางผู้วิจัยได้รับคำแนะนำเพื่อแก้ปัญหานี้โดยการใช้ Informix Data Director for Visual Basic version ใหม่ในการติดต่อกับฐานข้อมูล แต่ยังมีได้มีการทดลองใช้แต่อย่างไร เนื่องจากทางภาควิชากำลังติดต่อกับบริษัท Informix เพื่อขอ Software ตัวนี้มาทำการศึกษา
2. สำหรับผู้ที่ต้องการทำการวิจัยต่อ ควรจะมีการศึกษาทฤษฎีต่าง ๆ ของ Temporal Database ให้เข้าใจเป็นอย่างดีเสียก่อน หลังจากนั้นค่อยนำเอาทฤษฎีที่ได้ทำศึกษามาลงมือปฏิบัติ เพื่อประโยชน์ในเรื่องของเวลาและความถูกต้องของสิ่งที่นำไปปฏิบัติ

สำหรับแนวทางการศึกษาต่อในโครงการ Temporal Database มีดังต่อไปนี้

1. ทำการพัฒนาตัว Map จาก ภาษา SQL Temporal ให้เป็น SQL มาตรฐาน ให้มีความสมบูรณ์ถูกต้องมากยิ่งขึ้น
2. ทำการพัฒนาโปรแกรมทะเบียนของนักศึกษา เพื่อนำไปใช้งานจริงต่อไปในอนาคต

6.2 สรุป

จากการทดลองและทำการวิจัยโครงการในหัวข้อฐานข้อมูลเชิงเวลา (Temporal) ทำให้สามารถสรุปผลการทดลองออกเป็น 2 ส่วน ด้วยกัน ได้แก่

1. ทางทฤษฎี

ในฐานข้อมูลเชิงเวลาจะมีการจัดเก็บข้อมูลตามทฤษฎีของ Temporal Database ดังที่ได้กล่าวมาแล้วข้างต้น ซึ่งจะมีการจัดเก็บส่วนที่เป็นเวลาลงไปด้วย สามารถแบ่งเวลาออกได้เป็น 3 ประเภท ได้แก่ Validtime (ช่วงเวลาที่ข้อมูลนั้นเป็นจริง), Transaction time (ช่วงเวลาที่ข้อมูลนั้นถูกเก็บอยู่ในฐานข้อมูล), User define time (เวลาที่ไม่ต้องมีการแปลความหมายในทาง temporal)

นอกจากนี้ยังมีการศึกษาการค้นหาข้อมูลจากฐานข้อมูลเชิงเวลา โดยใช้ SQL มาตรฐานใหม่ที่เรียกว่า SQL/Temporal ซึ่งช่วยให้สามารถค้นหาข้อมูลได้สะดวกมากขึ้น และได้มีการ

2. ทางปฏิบัติ

ได้ทำการพัฒนา Software งานทะเบียนของนักศึกษา ตามทฤษฎีของ Temporal ซึ่งสามารถแบ่งออกได้เป็น 3 โปรแกรม โดยแต่ละโปรแกรมมีหน้าที่และข้อจำกัดในการใช้งานดังต่อไปนี้

- โปรแกรมจัดการข้อมูลในฐานข้อมูลเชิงเวลา
ทำหน้าที่ในการจัดการการเก็บข้อมูลในฐานข้อมูลตามทฤษฎีของ Temporal
- โปรแกรมตอบคำถามอย่างง่าย
เป็นโปรแกรมที่ใช้ในการตอบคำถามอย่างง่ายที่ผู้ใช้มักถามบ่อย ๆ
- SQL Editor

ทำหน้าที่ในการค้นหาข้อมูลจากฐานข้อมูล โดยผู้ใช้งานสามารถป้อนคำสั่งภาษา SQL หรือ SQL/Temporal ก็ได้ หากผู้ใช้งานป้อนภาษา SQL/Temporal ก็จะมีการ map ให้เป็นคำสั่งภาษา SQL ซึ่งในการ map นี้ยังมีข้อจำกัดในการแปลง กล่าวคือ จากทฤษฎีการ map ทั้งหมดที่ได้ทำการศึกษา มีทั้งสิ้น 8 รูปแบบ ตามที่ได้กล่าวมาแล้วข้างต้น โดยใน SQL Editor นี้สามารถ map ได้ 6 รูปแบบ สำหรับ 2 รูปแบบ ที่ไม่สามารถ map ได้ นั้นเป็นการค้นหาที่มีการ join กันระหว่างตารางตั้งแต่ 2 ตารางขึ้นไป เนื่องจากไม่สามารถตั้งคำถามในการค้นหาข้อมูลให้ตรงกับทฤษฎีทั้ง 2 รูปแบบนั้นได้

นอกจากนั้นใน SQL Editor ยังมีเครื่องมือช่วยในการสร้างภาษา SQL (SQL Builder) สำหรับผู้ที่ไม่รู้จักภาษา SQL อีกด้วย

ภาคผนวก

ก. คู่มือการติดตั้งโปรแกรมเพื่อใช้งานฐานข้อมูลนักศึกษา

เครื่องที่ทำหน้าที่เป็น Client จะมีความต้องการของระบบ (System Requirements) ดังนี้

Requirement	Windows 95	Windows NT
Hardware	ไมโครโปรเซสเซอร์ 80486 หรือสูงกว่า	ไมโครโปรเซสเซอร์ 80486 หรือสูงกว่า
Windows Version	Windows 95	Version 4.0
RAM	12 MB	16 MB
Hard-disk	20 MB (หรือน้อยกว่านี้ ขึ้นอยู่ กับ Component ที่เลือกทำ การติดตั้ง)	20 MB (หรือน้อยกว่านี้ ขึ้นอยู่ กับ Component ที่เลือกทำการ ติดตั้ง)

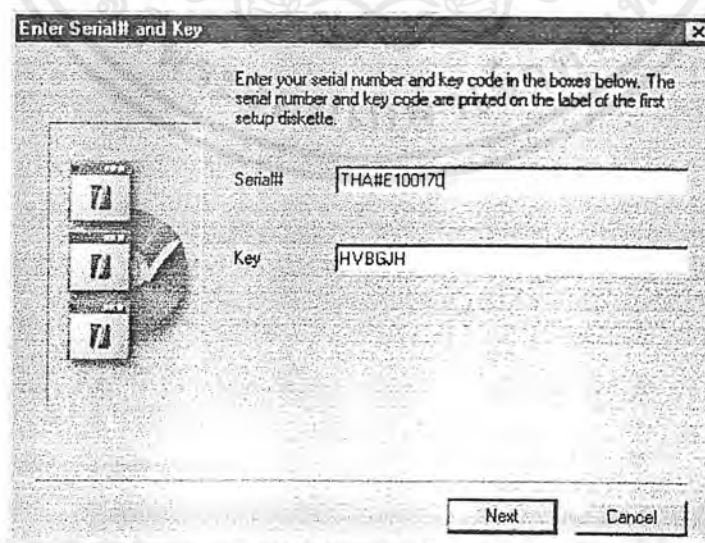
ตารางที่ 24 ความต้องการของระบบ

ขั้นตอนการติดตั้งเพื่อใช้งานโปรแกรม

สามารถแบ่งออกได้เป็น 3 ส่วนดังนี้

1. ติดตั้งโปรแกรมที่ใช้ในการติดต่อกับฐานข้อมูล Informix ได้แก่ โปรแกรม **Informix Connect**

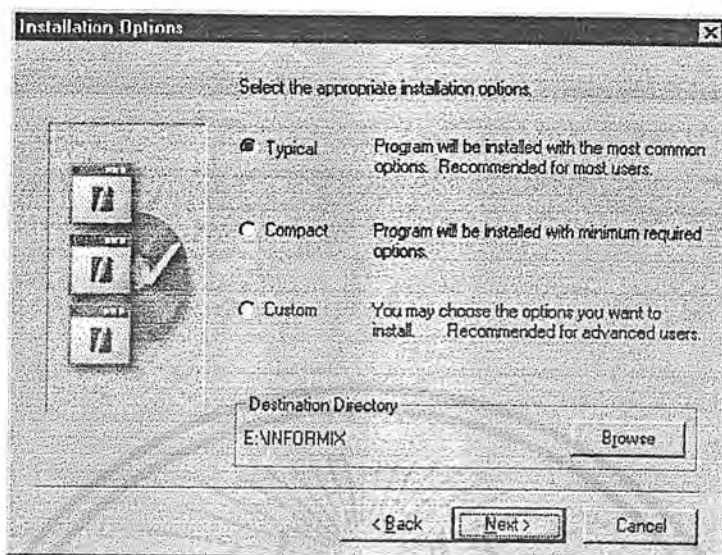
1.1 ทำการรัน File Setup.exe ขึ้นมาหลังจากนั้นทำการใส่ค่า Serial# และ Key แล้วกดปุ่ม Next



รูปที่ 13

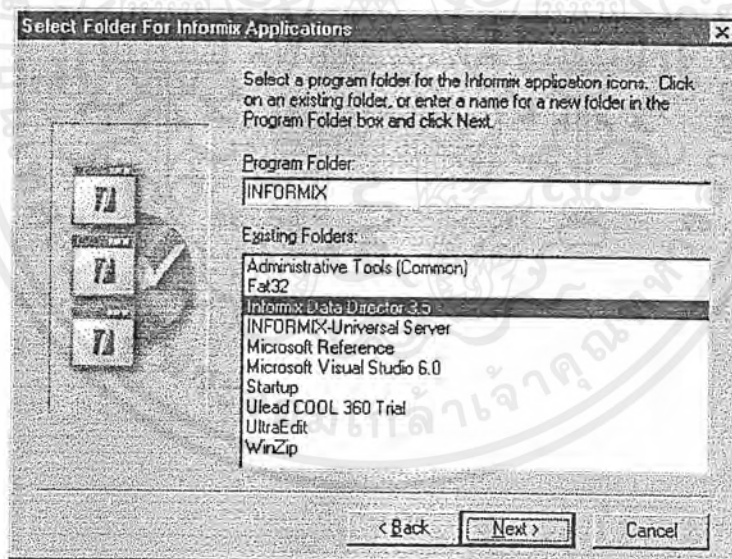
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.2 หลังจากนั้นโปรแกรมจะถามว่าจะเลือกแบบใด ให้เลือกแบบ Typical และทำการเลือก Destination Directory จากนั้นคลิกปุ่ม Next



รูปที่ 14

- 1.3 ต่อมาจะเป็นหน้าจอในการเลือก program folder ทำการเลือกแล้วคลิกปุ่ม Next แล้วรอนจนกระทั่งการติดตั้งโปรแกรมเสร็จเรียบร้อย

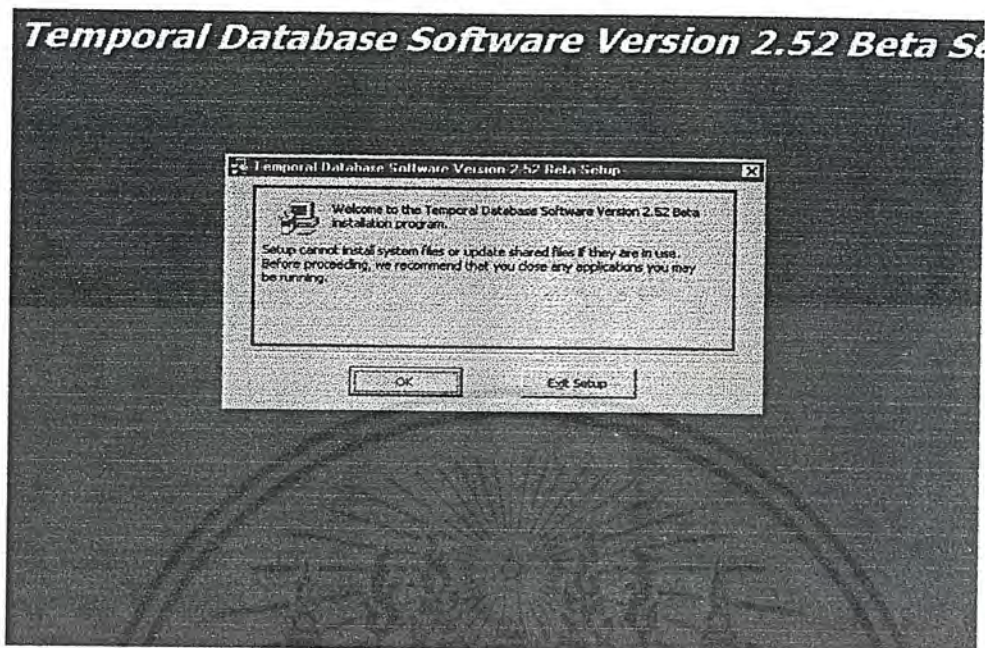


รูปที่ 15 .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

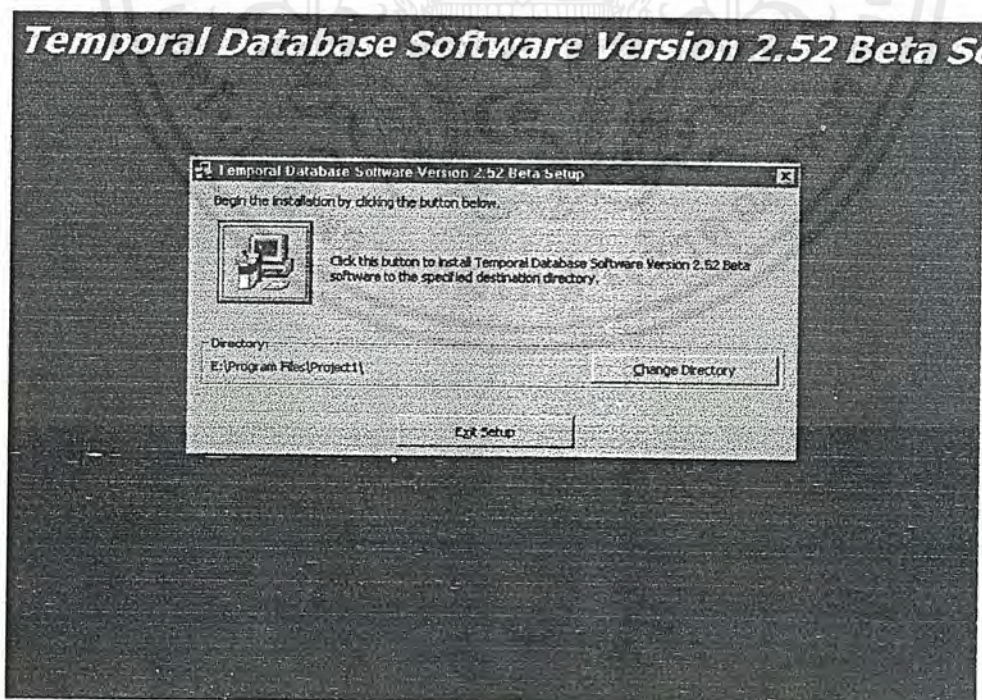
2. ติดตั้งโปรแกรมระบบงานทะเบียนนักศึกษา

2.1 ทำการรัน File\Packages\Setup.exe แล้วทำการกด ok เพื่อเริ่มการติดตั้ง



รูปที่ 16

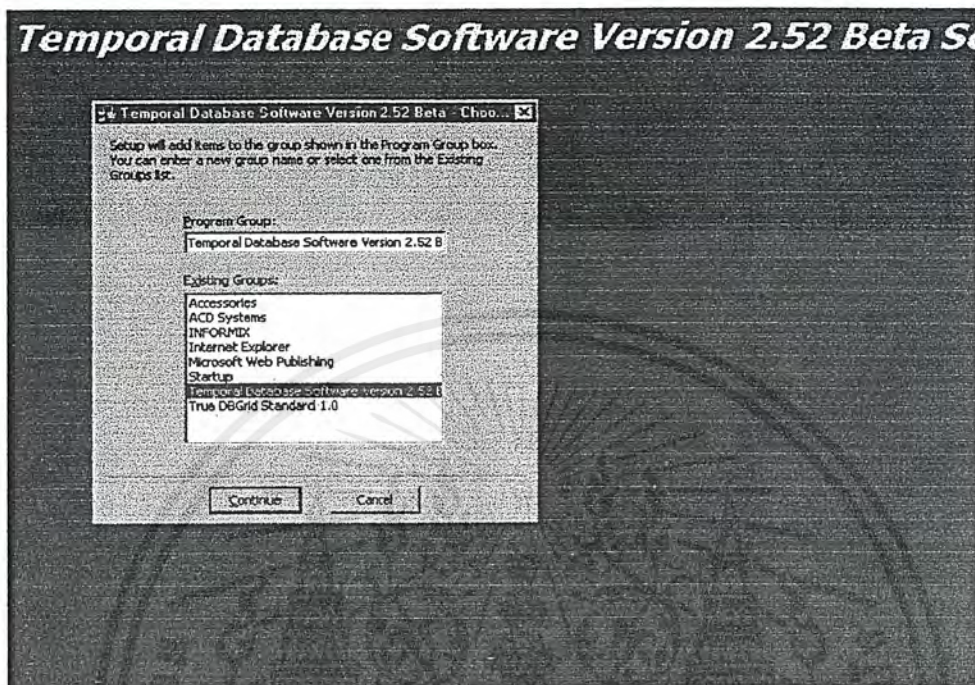
2.2 หลังจากนั้นให้ทำการเลือก Directory ที่ต้องการจะติดตั้งโปรแกรมลงไป แล้วกดปุ่ม install



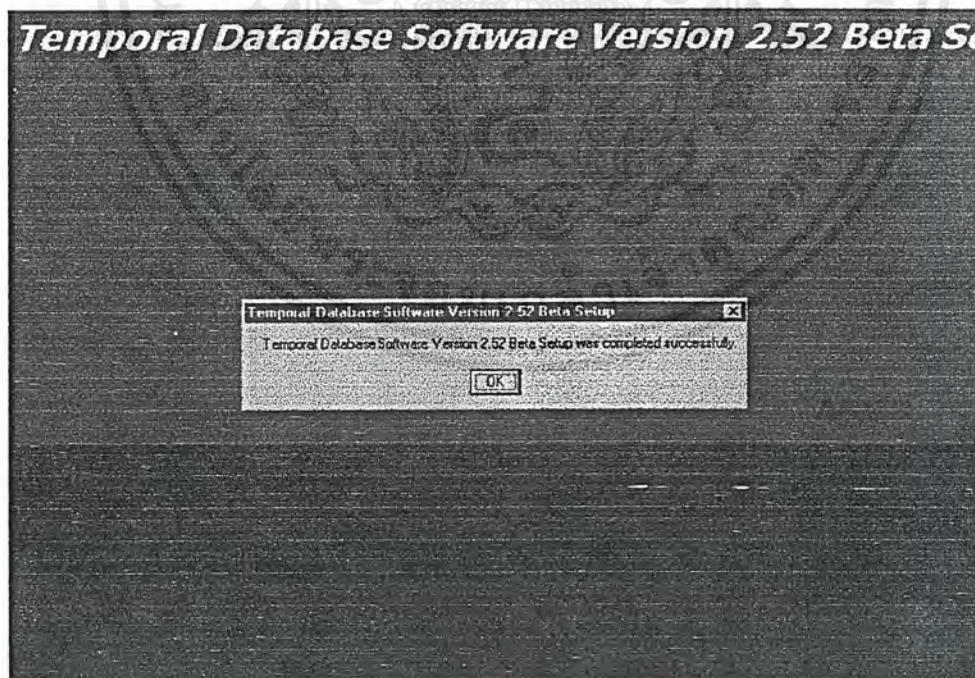
รูปที่ 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2.3 จากนั้นทำการเลือก program group ในการติดตั้ง กดปุ่ม Continue แล้วรอนกระทั่งการติดตั้งเสร็จสมบูรณ์ดังรูปที่ ซึ่งในระหว่างการติดตั้งอาจมีการฟ้อง error บางให้กดปุ่ม ignore ข้ามไป



รูปที่ 18



รูปที่ 19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ติดตั้ง ODBC และแก้ไขไฟล์ Host เพื่อติดต่อกับ Database Server

- 3.1 ไปที่ Start → Setting → Control Panel → ODBC หลังจากนั้น ที่ Tab User DSN ให้กดปุ่ม Add แล้วเลือกที่ INFORMIX 2.70 32 BIT จากนั้นกดปุ่ม Finish ซึ่งจะขึ้นหน้าจอดังนี้

รูปที่ 20

- 3.2 ให้ทำการกรอกข้อมูลตามรูปที่ 20 โดยที่ค่า Server และ Host จะมีชื่อตาม Database Server ที่ได้ทำการติดตั้งขึ้น สำหรับ PWD = informix จากนั้นกดปุ่ม OK

- 3.3 ทำการแก้ไข file Hosts ซึ่งหาได้โดยทำการ Find Files or Folders โดยทำการเพิ่มบรรทัดที่เป็น IP address และ ชื่อของ Server ดังตัวอย่างต่อไปนี้

```
10.99.1.10      ol_sengtn
```

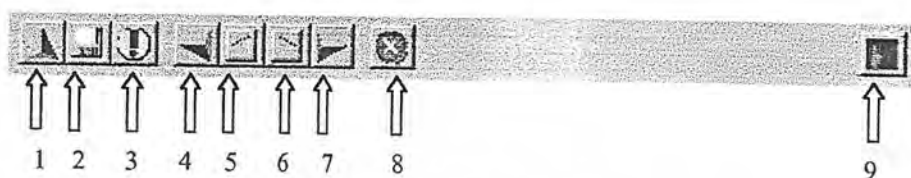
- 3.4 ตรวจสอบ file Services ซึ่งหาได้โดยทำการ Find Files or Folders ว่ามีบรรทัดที่มีข้อความดังต่อไปนี้หรือไม่ ถ้าไม่มีให้ใส่เพิ่มลงไป

```
turbo 1526/tcp      # INFORMIX-Universal Server
```

ข. วิธีการใช้โปรแกรม

1. โปรแกรมการจัดการข้อมูลในฐานข้อมูลเชิงเวลา

การใช้งานโปรแกรมจัดการข้อมูลในฐานข้อมูลนี้ จะต้องเริ่มจากสร้างฐานข้อมูลขึ้นมาใหม่ ซึ่งในแต่ละตารางจะไม่มีข้อมูลใคอยู่เลย โดยก่อนที่จะเริ่มใช้งานนั้นต้องทำความเข้าใจกับปุ่มต่าง ๆ ดังต่อไปนี้



หมายเลขปุ่ม	การทำงาน
9	ออกจากโปรแกรม
8	ลบแถวนั้นออกจากตาราง
7	เลื่อนไปยังแถวท้ายสุดของตาราง
6	เลื่อนไปยังแถวถัดไป
5	เลื่อนไปยังแถวก่อนหน้า
4	เลื่อนไปยังแถวแรกสุดของตาราง
3	ทำการ Clear ค่าใน Field ต่าง ๆ
2	บันทึกการเปลี่ยนแปลง
1	เพิ่มข้อมูลใหม่ (Insert)

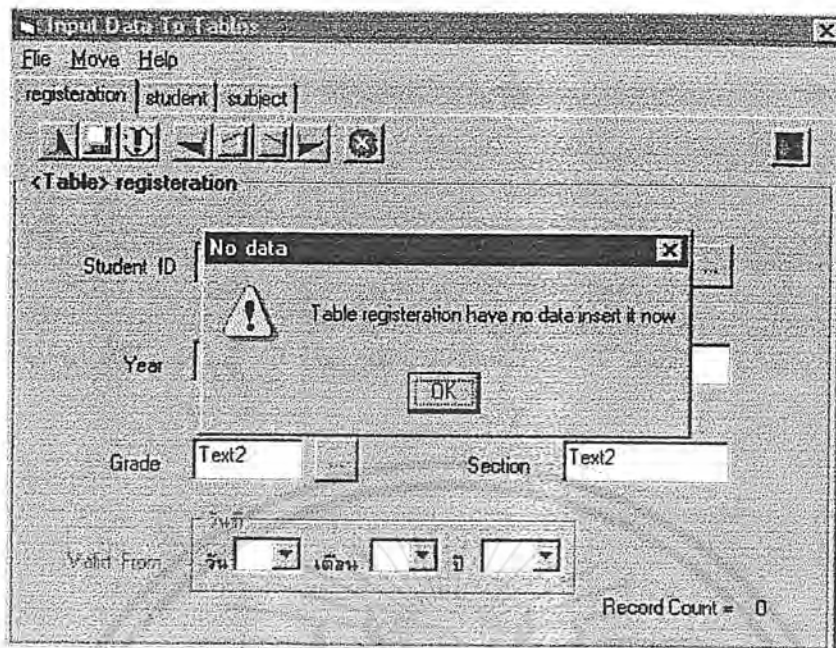
ตารางที่ 25 การทำงานของปุ่มต่าง ๆ

วิธีการใช้โปรแกรม

สามารถแบ่ง Operation การทำงานได้เป็น 3 ส่วน หลัก ๆ ดังนี้

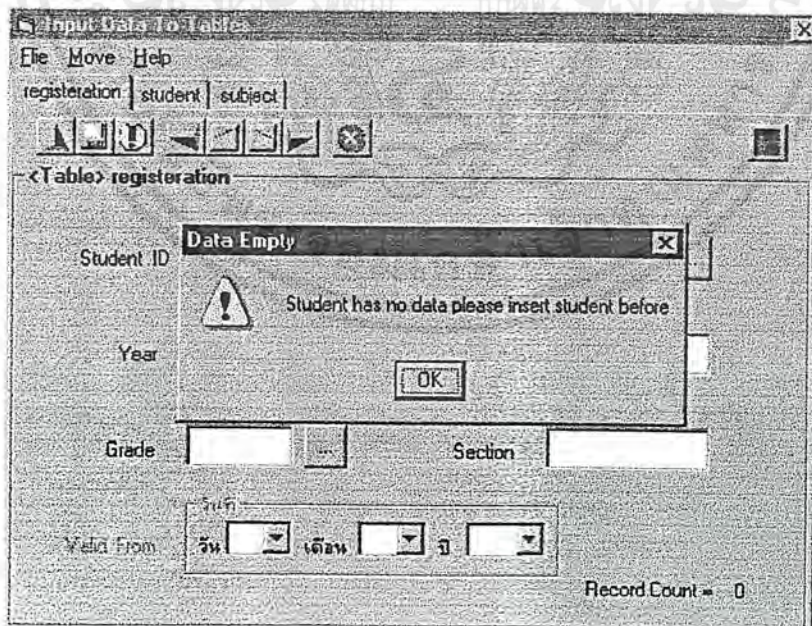
1. การเพิ่มข้อมูล (insert)

1.1 เมื่อเริ่มโปรแกรมจะเห็นว่ามี 3 Tabstrip ซึ่งเป็นของแต่ละตารางที่ผู้ป้อนข้อมูลต้องใช้ โดยเริ่มต้นจะอยู่ที่ตาราง Registration เนื่องจากในฐานข้อมูลไม่มีข้อมูลใคอยู่เลยจึงมีความขึ้นดังรูปที่ 21



รูปที่ 21

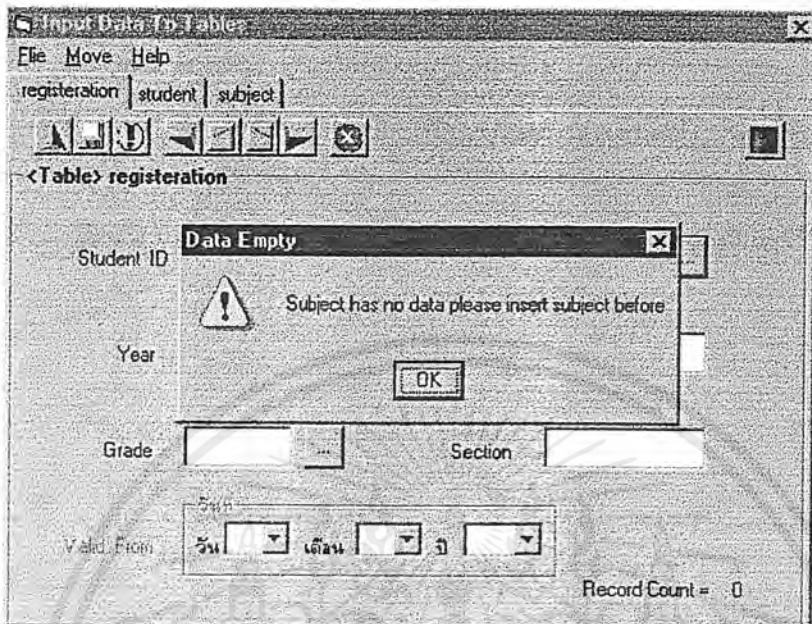
1.2 จากรูปที่ 1 เมื่อคลิกปุ่ม OK แล้วจะเหมือนกับเป็นการอนุญาตให้เพิ่มข้อมูล เมื่อคลิกที่ปุ่ม Browse ของ Student ID จะปรากฏข้อความเตือนว่ายังไม่มีข้อมูลนักศึกษาคนใดเลย เป็นดังรูปที่ 22



รูปที่ 22

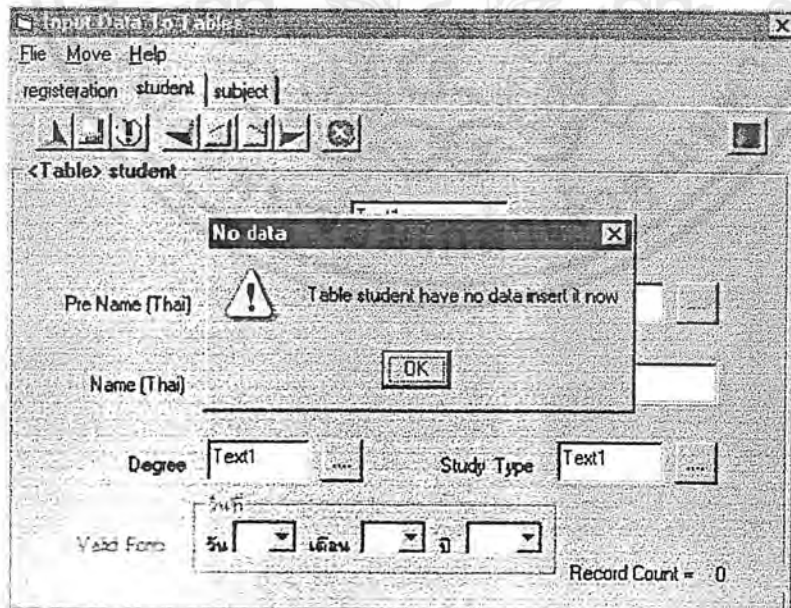
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 หลังจากนั้นถ้าคลิกที่ปุ่ม Browse ของ Subject ID ก็จะปรากฏข้อความเตือนว่าตาราง Subject ยังไม่มีข้อมูลใดๆ เลย ดังรูปที่ 23



รูปที่ 23

1.4 จากนั้นให้ทำการเปลี่ยนแท็บไปที่ตาราง Student จะมีข้อความเตือนว่ายังไม่มีข้อมูลของนักศึกษา ให้คลิก OK เพื่อทำการเพิ่มข้อมูลลงในตาราง



รูปที่ 24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ทำการใส่ข้อมูลที่ต้องการลงในตาราง รวมถึงใส่วันที่ที่ข้อมูลนั้นเป็นจริงด้วย ดังรูปที่ 25

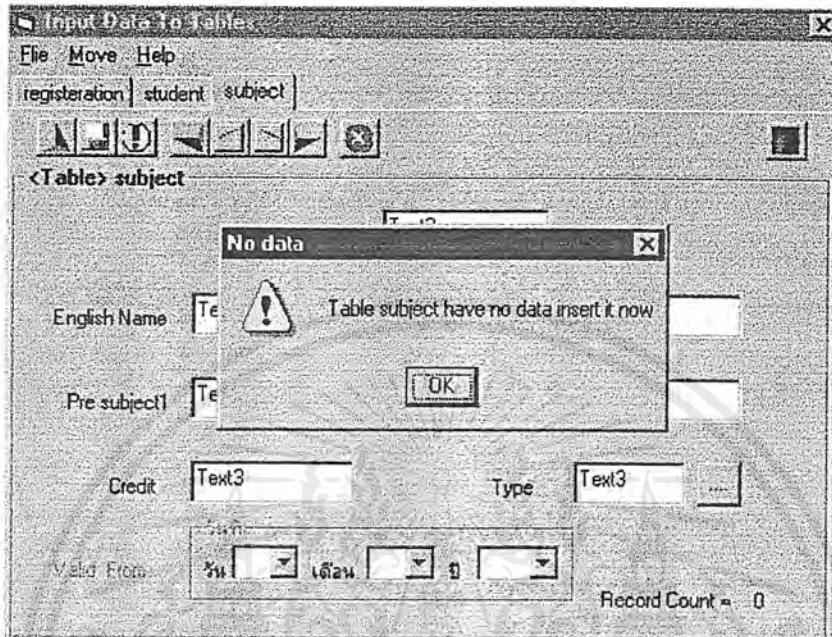
รูปที่ 25

1.6 จากนั้นทำการคลิกปุ่มเพิ่มข้อมูลอีกครั้งเพื่อทำการบันทึกข้อมูลลงไปในฐานข้อมูล โดยโปรแกรมจะถามว่าต้องการเปลี่ยนแปลงค่าใช่หรือไม่ให้ตอบ OK ดังรูปที่ 26

รูปที่ 26

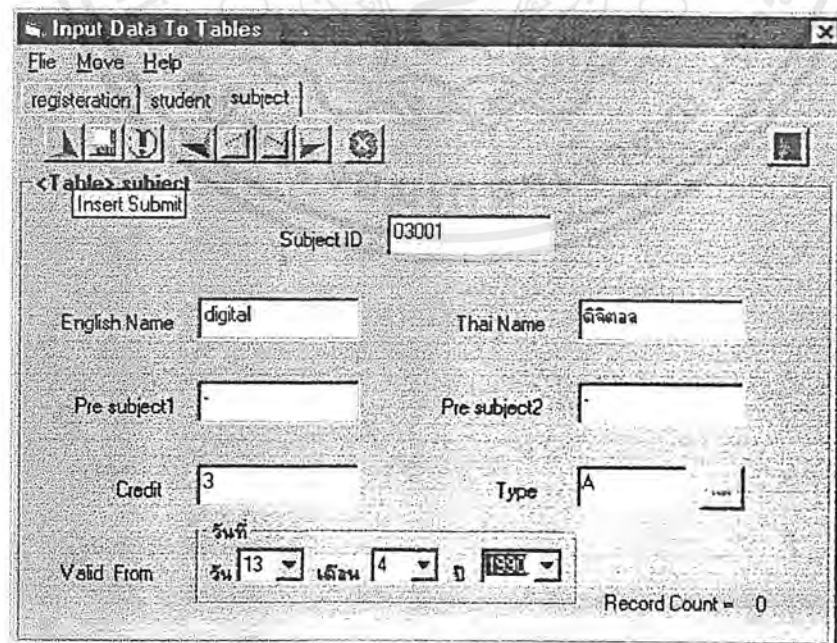
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.7 หลังจากทำการเพิ่มข้อมูลเข้าไปยังตาราง Student แล้ว ให้ทำการเพิ่มข้อมูลไปยังตาราง Subject โดยคลิกที่แท็บ Subject จะปรากฏดังรูปที่ 27 เนื่องจากยังไม่มีข้อมูลใด ๆ อยู่ในตารางเลย จากนั้นกดปุ่ม OK เพื่อทำการเพิ่มข้อมูล



รูปที่ 27

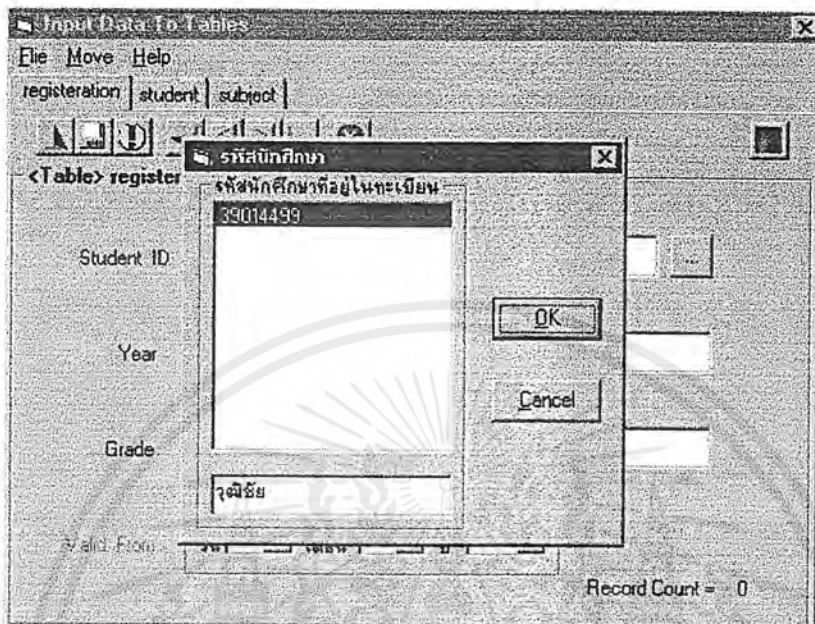
1.8 ให้ทำการใส่ข้อมูลที่ต้องการแล้วกดปุ่มเพิ่มข้อมูล โปรแกรมก็จะถามยืนยันความต้องการเช่นเดียวกับที่ตาราง Student



รูปที่ 28

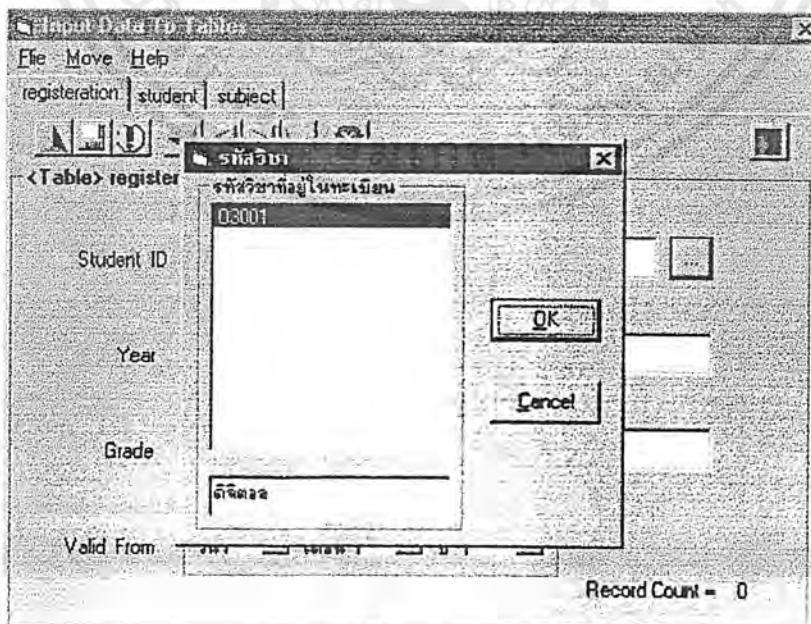
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.9 หลังจากทำการเพิ่มข้อมูลในตาราง Student และ Subject แล้ว จึงสามารถทำการเพิ่มข้อมูลในตาราง registration โดยการคลิกปุ่ม Browse ของที่ Student ID จะปรากฏหน้าจอ ดังรูปที่ 29 ซึ่งเป็นรหัสนักศึกษาที่อยู่ในทะเบียน



รูปที่ 29

1.10 หลังจากนั้นคลิกที่ปุ่ม Browse ของ Subject ID ก็จะปรากฏรหัสวิชาที่มีอยู่ในฐานข้อมูลดังรูปที่ 30 หลังจากนั้นให้ทำการกรอกข้อมูลอื่น ๆ รวมถึงเวลาที่ข้อมูลนั้นเป็นจริงลงไป แล้วกดปุ่มเพิ่มข้อมูล (Insert)



รูปที่ 30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การเปลี่ยนแปลงแก้ไขข้อมูล (Update)

2.1 ทำการแก้ไขข้อมูลที่ต้องการดังรูปที่ 31 แล้วทำการใส่เวลาเริ่มต้นที่ข้อมูลที่เปลี่ยนแปลงเป็นจริง

Input Data To Tables

File Move Help

registration student subject

<Table> student

Save Update data

Student ID 39014499

Pre Name (Thai) นาย Pre Name (English) Mr.

Name (Thai) วุฒิชัย Name (English) Wuttichai

Degree M Study Type B

วันที่ วัน 11 เดือน 9 ปี 1997

Valid Form

Record Count = 1

เปลี่ยนจาก A เป็น B

รูปที่ 31

2.2 จากนั้นกดปุ่ม Save Update Data เพื่อทำการบันทึกการเปลี่ยนแปลงข้อมูล จะได้ผลลัพธ์ดังรูปที่ 32 จากนั้นกด OK เพื่อยืนยันการเปลี่ยนแปลง

Input Data To Tables

File Move Help

registration student subject

<Table> student

Save data

Are you sure to save change to data

OK Cancel

Pre Name (Thai) นาย Pre Name (English) Mr.

Name (Thai) วุฒิชัย Name (English) Wuttichai

Degree M Study Type B

วันที่ วัน 11 เดือน 9 ปี 1997

Valid Form

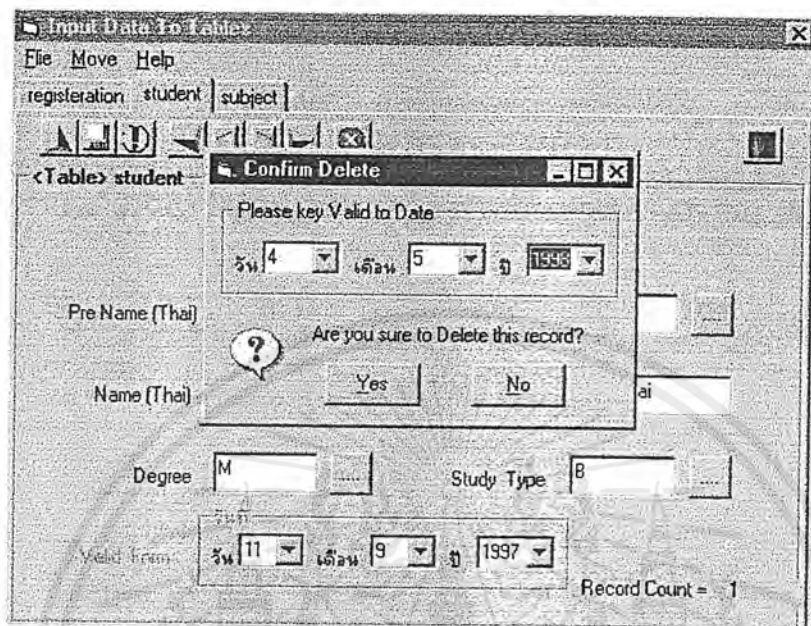
Record Count = 1

รูปที่ 32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การลบข้อมูล (Delete)

3.1 เมื่อเลื่อนไปยังแถวที่ต้องการจะลบแล้วคลิกปุ่มลบข้อมูลออกจากฐานข้อมูล จะปรากฏ หน้าจอดังรูปที่ 33

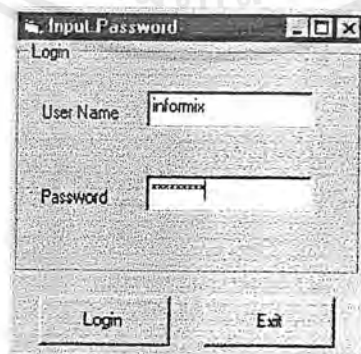


รูปที่ 33

3.2 โปรแกรมจะให้กรอกเวลาสิ้นสุดที่ข้อมูลนั้น แล้วยืนยันการลบข้อมูลโดยคลิกปุ่ม Yes เป็นอันเสร็จการลบข้อมูล

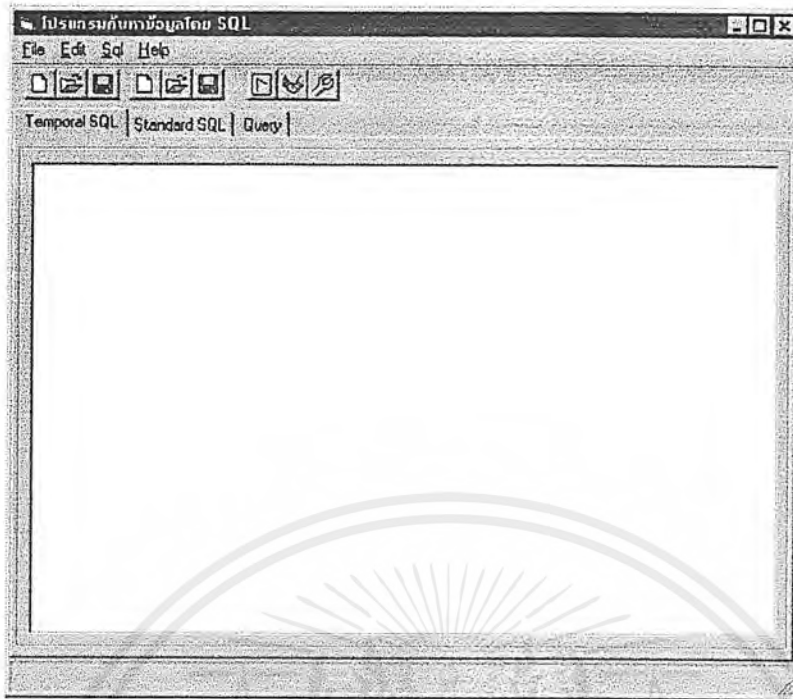
2. โปรแกรม SQL Editor

การทำงานกับ โปรแกรมค้นหาข้อมูลจะต้องเริ่มจากการLogin เข้าสู่ระบบโดยป้อนชื่อผู้ใช้และรหัสผ่านก่อนดังรูปที่ 34



รูปที่ 34

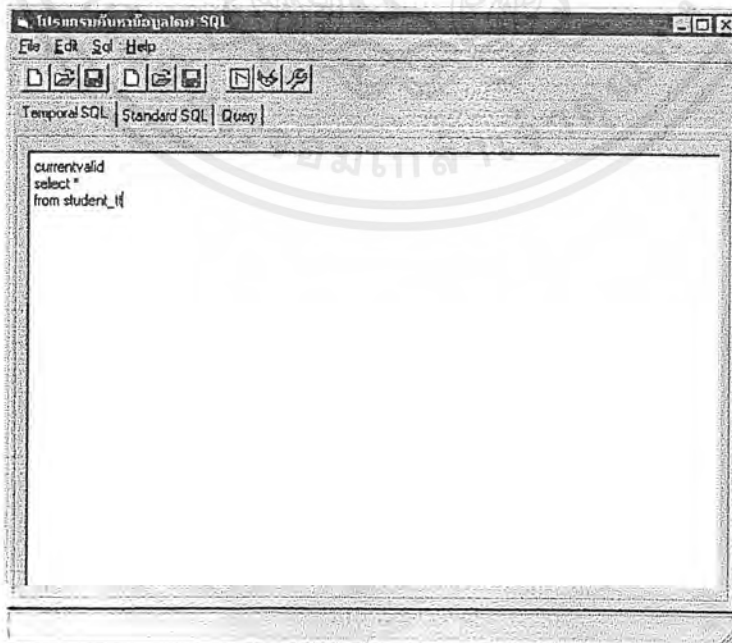
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 35

เมื่อเข้าสู่ระบบได้แล้วจะเห็นหน้าจอที่จะให้ป้อนภาษา Temporal SQL และ SQL ซึ่งเราสามารถป้อนภาษา ได้ทั้งสองแบบ ดังรูปที่ 35

เมื่อเราป้อนคำสั่ง Temporal SQL ลงในแท็บแรกตามทฤษฎีของ Temporal Database จากนั้นไปที่ menu => Sql => map Temp SQL to SQL เพื่อทำการ map จากภาษา sql ไปเป็น SQL มาตรฐานดังตัวอย่างดังรูปที่ 36

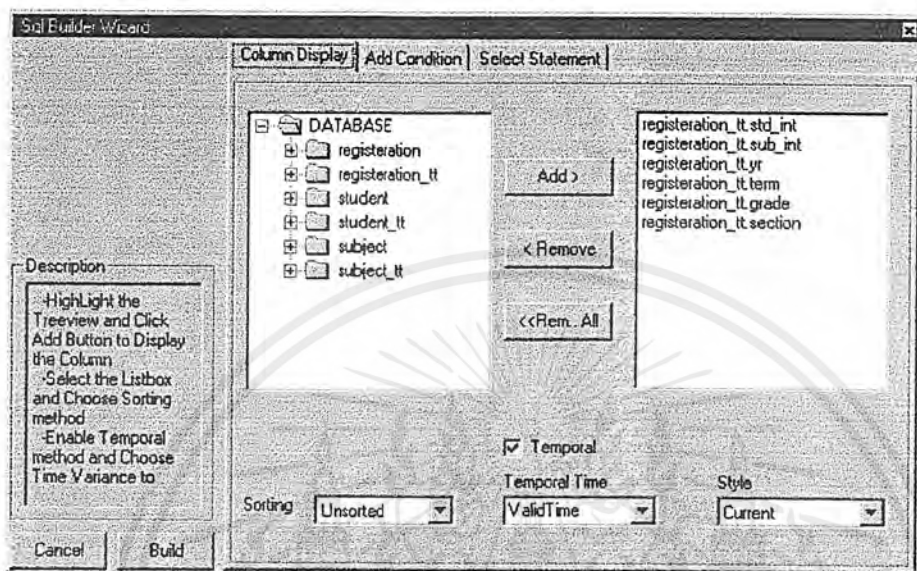


รูปที่ 36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และขั้นตอนสุดท้ายก็คือ ทำการกดปุ่มเลือกที่ menu => sql => Query เพื่อทำการแสดงผลการค้นหาวางหน้าจอ

นอกจากนี้ยังมีวีซาร์ด ที่ช่วยในการสร้างคำสั่งSQL ได้โดยไม่ต้องรู้โครงสร้างของภาษานี้ซึ่งทำโดยไปที่ menu => sql => Query Builder จะได้หน้าจอดังรูปที่ 37

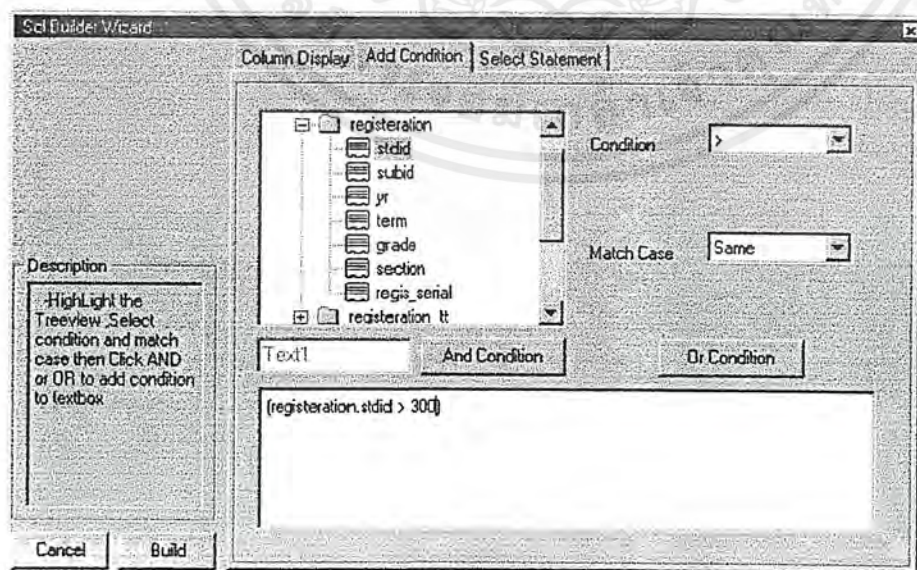


รูปที่ 37

หน้าจอแรกนี้จะเป็นหน้าจอให้เราเลือกคอลัมน์ที่ต้องการแสดงออกทางหน้าจอแสดงผล และยังเป็นหน้าจอที่ให้เราเลือกรูปแบบของการแสดงเวลาได้อีกด้วย

จากนั้นให้เราคลิกที่แท็บที่สองจะเป็นส่วนของเงื่อนไขที่ต้องการกรองข้อมูลที่ไม่ต้องการออกไป

ดังรูป 38

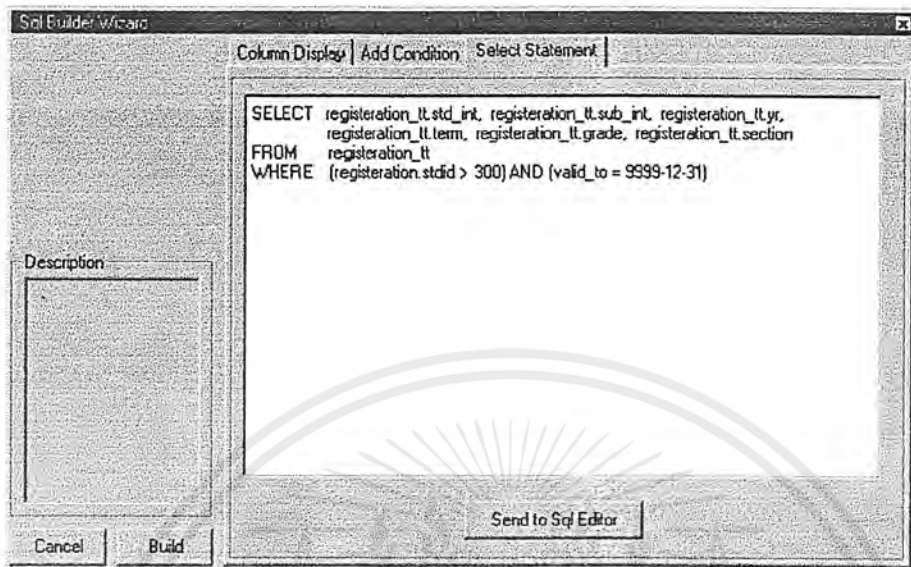


รูปที่ 38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่ใส่เงื่อนไขเรียบร้อยแล้ว ก็ให้คลิกที่ปุ่ม Build เพื่อทำการสร้างคำสั่ง SQL ออกมาดังรูปที่

39



รูปที่ 39

หลังจากที่เราได้ คำสั่ง Sql แล้วเราสามารถส่ง ไปยัง หน้าจอหลัก โดยการคลิกที่ปุ่ม Send to Sql Editor

6.2.3 โปรแกรมค้นหาข้อมูลจากคำถามพื้นฐาน

การทำงานกับระบบค้นหาข้อมูลจากหน้าจอเลือกคำถามดังรูปที่ 40



รูปที่ 40

ทำการเลือกคำถามที่ต้องการถาม จากนั้นก็ทำการ ใส่รายละเอียดของคำถามดังรูปที่ 41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 41

ขั้นตอนสุดท้ายก็ทำการคลิกที่ปุ่ม ค้นหา เพื่อทำการค้นหาข้อมูลตามที่ต้องการดังรูปที่ 42

stcid	tpren	t1.tname	epren	t1.ename	degree	stdtype	subid	t2.tname	t2.ename	p
▶ 101	นาย	นิเทศ	Mr.	Nipon	B	B	1233	วิศวกรรมเครื่องกล	Electrical Machine	-

รูปที่ 42

นอกจากนี้ยังมีคำถามอื่นที่สามารถ Query ได้มีดังนี้

1. ประวัติการลงทะเบียนของ ในวิชา ...
2. ประวัติของนักศึกษา
3. ประวัติของรายวิชา
4. ข้อมูลปัจจุบันของการลงทะเบียน ... ของวิชา
5. ข้อมูลปัจจุบันของนักศึกษา
6. ข้อมูลปัจจุบันของรายวิชา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค. Source Code ของโปรแกรม

1. โปรแกรมจัดการฐานข้อมูลเชิงเวลา

Source Code ในส่วนของ Input.frm

```

Public ii As Integer
Public counts As Integer
Dim databas As Database
Dim work As Workspace
Public record_regis As Recordset
Public record_std As Recordset
Public record_sub As Recordset
Public record_regisTT As Recordset
Public record_stdTT As Recordset
Public record_subTT As Recordset
Dim sql1 As String
Dim str(3) As String
Public i1, j1, k1, con As Integer
Dim chksave As Boolean

Private Sub Command1_Click()
    Form2.Show
    Form2.Frame1.ZOrder 0
    Form2.Caption = "Pre name (Thai)"
    Form1.Enabled = False
End Sub

Private Sub Command2_Click()
    Form2.Frame2.ZOrder 0
    Form2.Show
    Form2.Caption = "Pre name (English)"
    Form1.Enabled = False
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Command3_Click()
    Form2.Frame3.ZOrder 0
    Form2.Show
    Form2.Caption = "Select Degree"
    Form2.Frame3.Caption = "Degree"
    Form1.Enabled = False
End Sub

```

```

Private Sub Command4_Click()
    Form2.Frame4.ZOrder 0
    Form2.Option1.Value = True
    Form2.Show
    Form2.Caption = "Select Study type"
    Form2.Frame4.Caption = "Study type"
    Form1.Enabled = False
End Sub

```

```

Private Sub Command5_Click()
    Form2.Frame5.ZOrder 0
    Form2.Show
    Form2.Caption = "Select Subject type"
    Form2.Frame5.Caption = "Subject type"
    Form1.Enabled = False
End Sub

```

```

Private Sub Command6_Click()
    Form2.Frame6.ZOrder 0
    Form2.Show
    Form2.Caption = "Select Grade"
    Form2.Frame6.Caption = "Grade"
    Form1.Enabled = False
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Command7_Click()
Dim aa, i, j As Integer
  If record_std.RecordCount <> 0 Then
    Form3.List1.Clear
    record_std.MoveLast
    aa = record_std.RecordCount
    record_std.MoveFirst

    For i = 0 To aa - 1
      If j = 0 Then
        record_std.MoveFirst
        j = 1
      Else: record_std.MoveNext
      End If
      Form3.List1.AddItem record_std.Fields("stdid").Value
    Next i
    Form3.List1.Selected(0) = True
    record_std.MoveFirst
    Form1.Enabled = False
    With Form3
      .Caption = "เอน์เ็น;เอ็;เอ๋"
      .Frame1.ZOrder 0
      .Text1.Text = record_std.Fields("tname").Value
      .Visible = True
      .Show
    End With
  Else
    MsgBox "Student has no data please insert student before", vbOKOnly + vbExclamation, "Data
Empty"
  End If
End Sub

```

```
Private Sub Command8_Click()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

Combo3(1).Enabled = True
Combo3(2).Enabled = True
Label12(0).Enabled = True
Label12(1).Enabled = True
Label12(2).Enabled = True

```

```
End Sub
```

```
Public Sub regisTimeEn_false()
```

```

Label2(6).Enabled = False
Frame4.Enabled = False
Combo3(0).Enabled = False
Combo3(1).Enabled = False
Combo3(2).Enabled = False
Label12(0).Enabled = False
Label12(1).Enabled = False
Label12(2).Enabled = False

```

```
End Sub
```

```
Public Sub stdTimeEn_true()
```

```

Label1(7).Enabled = True
Frame3.Enabled = True
Combo2(0).Enabled = True
Combo2(1).Enabled = True
Combo2(2).Enabled = True
Label11(0).Enabled = True
Label11(1).Enabled = True
Label11(2).Enabled = True

```

```
End Sub
```

```
Public Sub stdTimeEn_false()
```

```

Label1(7).Enabled = False
Frame3.Enabled = False
Combo2(0).Enabled = False
Combo2(1).Enabled = False
Combo2(2).Enabled = False
Label11(0).Enabled = False
Label11(1).Enabled = False

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Label11(2).Enabled = False
End Sub
Public Sub subTimeEn_true()
Label3(7).Enabled = True
Frame2.Enabled = True
Combo1(0).Enabled = True
Combo1(1).Enabled = True
Combo1(2).Enabled = True
Label10(0).Enabled = True
Label10(1).Enabled = True
Label10(2).Enabled = True
End Sub
Public Sub subTimeEn_false()
Label3(7).Enabled = False
Frame2.Enabled = False
Combo1(0).Enabled = False
Combo1(1).Enabled = False
Combo1(2).Enabled = False
Label10(0).Enabled = False
Label10(1).Enabled = False
Label10(2).Enabled = False
End Sub

Private Sub delregis_Click()
Dim msg As String
Dim aa1, aa2, i As Integer
If (TabStrip1.SelectedItem.Index - 1) = 0 Then
record_regis.MoveLast
aa1 = record_regis.RecordCount
record_regisTT.MoveLast
aa2 = record_regisTT.RecordCount
msg = MsgBox("Are you sure to delete all data in registration", vbOKCancel + vbExclamation,
"Confirm Delete All Data")
If msg = vbOK Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

For i = 1 To aa1
    record_regis.Delete
    If i <> aa1 Then
        record_regis.MovePrevious
    End If
Next i
For i = 1 To aa2
    record_regisTT.Delete
    If i <> aa2 Then
        record_regisTT.MovePrevious
    End If
Next i

Label5.Caption = 0
clear_click
i1 = 0
MsgBox "Table registration have no data insert it now", vbOKOnly + vbExclamation, "No
data"
insert_click
Else:
End If
End If
End Sub

```

```

Private Sub delstd_Click()
Dim msg As String
Dim aa1, aa2, i As Integer
If (TabStrip1.SelectedItem.Index - 1) = 1 Then
    record_std.MoveLast
    aa1 = record_std.RecordCount
    record_stdTT.MoveLast
    aa2 = record_stdTT.RecordCount
    msg = MsgBox("Are you sure to delete all data in student", vbOKCancel + vbExclamation,
"Confirm Delete All Data")

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If msg = vbOK Then
    For i = 1 To aa1
        record_std.Delete
        If i <> aa1 Then
            record_std.MovePrevious
        End If
    Next i
    For i = 1 To aa2
        record_stdTT.Delete
        If i <> aa2 Then
            record_stdTT.MovePrevious
        End If
    Next i

    Label7.Caption = 0
    clear_click
    j1 = 0
    MsgBox "Table student have no data insert it now", vbOKOnly + vbExclamation, "No data"
    insert_click
Else:
    End If
End If
End Sub

```

```

Private Sub delsub_Click()
    Dim msg As String
    Dim aa1, aa2, i As Integer
    If (TabStrip1.SelectedItem.Index - 1) = 2 Then
        record_sub.MoveLast
        aa1 = record_sub.RecordCount
        record_subTT.MoveLast
        aa2 = record_subTT.RecordCount
        msg = MsgBox("Are you sure to delete all data in subject", vbOKCancel + vbExclamation,
"Confirm Delete All Data")

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If msg = vbOK Then
    For i = 1 To aa1
        record_sub.Delete
        If i <> aa1 Then
            record_sub.MovePrevious
        End If
    Next i
    For i = 1 To aa2
        record_subTT.Delete
        If i <> aa2 Then
            record_subTT.MovePrevious
        End If
    Next i

    Label9.Caption = 0
    clear_click
    k1 = 0
    MsgBox "Table subject have no data insert it now", vbOKOnly + vbExclamation, "No data"
    insert_click
Else:
    End If
End If
End Sub

Private Sub Form_Activate()
    Dim j As Integer
    Dim k, l, m, kk, che As Integer
    If con = 0 Then
        For j = 0 To Data1.Database.TableDefs.Count - 1
            If (Data1.Database.TableDefs(j).Attributes And _
                (dbSystemObject + dbHiddenObject)) = 0 Then
                texts1.Text = Data1.Database.TableDefs(j).Name
                If che = 0 Then
                    texts1.Text = texts1.Text + "*"
                End If
            End If
        Next j
    End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

k = texts1.Find(".")
texts1.SelStart = k + 1
texts1.SelLength = 30
texts1.Text = texts1.SelText
kk = texts1.Find("*")
texts1.SelStart = 0
texts1.SelLength = kk
TabStrip1.Tabs.Add ii, , texts1.SelText
str(m) = texts1.SelText
m = m + 1
l = ii - 1
With Frame1(l)
    .Caption = "<Table> " + texts1.SelText
    .FontBold = True
    .Height = Form1.Height - 1450
    .Width = Form1.Width - 190
    .Left = 30
    .Top = 800
    .Visible = True
End With
ii = ii + 1
che = 1
Else: che = 0
End If

End If
Next j
con = 1
If (record_regis.RecordCount > 0) Then
    record_regis.MoveLast
    Label5.Caption = record_regis.RecordCount
    record_regis.MoveFirst
End If

If (record_std.RecordCount > 0) Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

record_std.MoveLast
Label7.Caption = record_std.RecordCount
record_std.MoveFirst
End If
If (record_sub.RecordCount <> 0) Then
    record_sub.MoveLast
    Label9.Caption = record_sub.RecordCount
    record_sub.MoveFirst
End If
If (TabStrip1.SelectedItem.Index - 1 = 0) And (record_regis.RecordCount = 0) Then
    MsgBox "Table registration have no data insert it now", vbOKOnly + vbExclamation, "No data"
    insert_click
End If
Call fillvalue
End If
Unload frmSplash
frmSplash.Hide
End Sub

Private Sub Form_Load()
Dim str As String
Dim year As Integer
regisTimeEn_false
stdTimeEn_false
subTimeEn_false
delregis.Enabled = True
delstd.Enabled = False
delsub.Enabled = False

Combo1(2).Clear
str = Format(Date, "yyyy")
year = str
For i = year - 15 To year - 15
    Combo1(2).AddItem i

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Combo2(2).AddItem i
```

```
Combo3(2).AddItem i
```

```
Next i
```

```
Set work = DBEngine.Workspaces(0)
```

```
Set databas = work.OpenDatabase("", False, False, _
```

```
"ODBC;DSN=student_data;DATABASE=student_data;UID=informix;PWD=informix;")
```

```
Set record_regis = databas.OpenRecordset("registration", dbOpenDynaset)
```

```
Set record_std = databas.OpenRecordset("student", dbOpenDynaset)
```

```
Set record_sub = databas.OpenRecordset("subject", dbOpenDynaset)
```

```
Set record_regisTT = databas.OpenRecordset("registration_TT", dbOpenDynaset)
```

```
Set record_stdTT = databas.OpenRecordset("student_TT", dbOpenDynaset)
```

```
Set record_subTT = databas.OpenRecordset("subject_TT", dbOpenDynaset)
```

```
Frame1(0).ZOrder 0
```

```
Data1.Connect = "ODBC;DSN=student_data;"
```

```
ii = 1
```

```
End Sub
```

```
Public Sub Save_link()
```

```
Dim chkcombo As Integer
```

```
'format time in informix is month/date/year
```

```
If (TabStrip1.SelectedItem.Index - 1) = 0 Then
```

```
record_regis.Fields("stdid").Value = Text2(0).Text
```

```
record_regis.Fields("subid").Value = Text2(1).Text
```

```
record_regis.Fields("yr").Value = Text2(2).Text
```

```
record_regis.Fields("term").Value = Text2(3).Text
```

```
record_regis.Fields("grade").Value = Text2(4).Text
```

```
record_regis.Fields("section").Value = Text2(5).Text
```

```
record_regis.Update
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

 to regisTT
record_std.MoveFirst
record_std.Move Form3.a1      'a1 is index of student
record_regisTT.Fields("std_int").Value = record_std.Fields("std_serial").Value

record_sub.MoveFirst
record_sub.Move Form3.a2      'a2 is index of subject
record_regisTT.Fields("sub_int").Value = record_sub.Fields("sub_serial").Value

record_regisTT.Fields("yr").Value = Text2(2).Text
record_regisTT.Fields("term").Value = Text2(3).Text
record_regisTT.Fields("grade").Value = Text2(4).Text
record_regisTT.Fields("section").Value = Text2(5).Text

'-----
chkcombo = Combo3(1).ListIndex
Text2(6).Text = Combo3(1).List(chkcombo)
Text2(6).Text = Text2(6).Text + "/"
chkcombo = Combo3(0).ListIndex
Text2(6).Text = Text2(6).Text + Combo3(0).List(chkcombo)
Text2(6).Text = Text2(6).Text + "/"
chkcombo = Combo3(2).ListIndex
Text2(6).Text = Text2(6).Text + Combo3(2).List(chkcombo)
'-----

record_regisTT.Fields("Valid_from").Value = Text2(6).Text
record_regisTT.Fields("Valid_to").Value = "12/31/9999"
record_regisTT.Fields("TxStart").Value = Format(Date, "mm/dd/yyyy")
record_regisTT.Fields("TxStop").Value = "12/31/9999"

record_regis.MoveLast
record_regisTT.Fields("regis_int").Value = record_regis.Fields("regis_serial").Value

record_regisTT.Update

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End If

If (TabStrip1.SelectedItem.Index - 1) = 1 Then

record_std.Fields("stdid").Value = Text1(0).Text

record_std.Fields("tpren").Value = Text1(1).Text

record_std.Fields("epren").Value = Text1(2).Text

record_std.Fields("tname").Value = Text1(3).Text

record_std.Fields("ename").Value = Text1(4).Text

record_std.Fields("degree").Value = Text1(5).Text

If Form2.Option1.Value = True Then

record_std.Fields("stdtype").Value = "A"

Elseif Form2.Option2.Value = True Then

record_std.Fields("stdtype").Value = "B"

End If

record_stdTT.Fields("stdid").Value = Text1(0).Text

record_stdTT.Fields("tpren").Value = Text1(1).Text

record_stdTT.Fields("epren").Value = Text1(2).Text

record_stdTT.Fields("tname").Value = Text1(3).Text

record_stdTT.Fields("ename").Value = Text1(4).Text

record_stdTT.Fields("degree").Value = Text1(5).Text

If Form2.Option1.Value = True Then

record_stdTT.Fields("stdtype").Value = "A"

Elseif Form2.Option2.Value = True Then

record_stdTT.Fields("stdtype").Value = "B"

End If

'-----

chkcombo = Combo2(1).ListIndex

Text1(7).Text = Combo2(1).List(chkcombo)

Text1(7).Text = Text1(7).Text + "/"

chkcombo = Combo2(0).ListIndex

Text1(7).Text = Text1(7).Text + Combo2(0).List(chkcombo)

Text1(7).Text = Text1(7).Text + "/"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

chkcombo = Combo2(2).ListIndex
Text1(7).Text = Text1(7).Text + Combo2(2).List(chkcombo)
'-----

record_stdTT.Fields("valid_from").Value = Text1(7).Text
record_stdTT.Fields("valid_to").Value = "12/31/9999"
record_stdTT.Fields("TxStart").Value = Format(Date, "mm/dd/yyyy")
record_stdTT.Fields("TxStop").Value = "12/31/9999"

record_std.Update
record_std.MoveLast
record_stdTT.Fields("std_int").Value = record_std.Fields("std_serial").Value
record_stdTT.Update
End If

If (TabStrip1.SelectedItem.Index - 1) = 2 Then
    record_sub.Fields("subid").Value = Text3(0).Text
    record_sub.Fields("ename").Value = Text3(1).Text
    record_sub.Fields("tname").Value = Text3(2).Text
    record_sub.Fields("presub1").Value = Text3(3).Text
    record_sub.Fields("presub2").Value = Text3(4).Text
    record_sub.Fields("credit").Value = Text3(5).Text
    record_sub.Fields("type").Value = Text3(6).Text

    record_subTT.Fields("subid").Value = Text3(0).Text
    record_subTT.Fields("ename").Value = Text3(1).Text
    record_subTT.Fields("tname").Value = Text3(2).Text
    record_subTT.Fields("presub1").Value = Text3(3).Text
    record_subTT.Fields("presub2").Value = Text3(4).Text
    record_subTT.Fields("credit").Value = Text3(5).Text
    record_subTT.Fields("type").Value = Text3(6).Text

'-----

chkcombo = Combo1(1).ListIndex
Text3(7).Text = Combo1(1).List(chkcombo)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Text3(7).Text = Text3(7).Text + "/"
chkcombo = Combo1(0).ListIndex
Text3(7).Text = Text3(7).Text + Combo1(0).List(chkcombo)
Text3(7).Text = Text3(7).Text + "/"
chkcombo = Combo1(2).ListIndex
Text3(7).Text = Text3(7).Text + Combo1(2).List(chkcombo)

```

```

record_subTT.Fields("valid_from").Value = Text3(7).Text
record_subTT.Fields("valid_to").Value = "12/31/9999"
record_subTT.Fields("TxStart").Value = Format(Date, "mm/dd/yyyy")
record_subTT.Fields("TxStop").Value = "12/31/9999"

record_sub.Update
record_sub.MoveLast
record_subTT.Fields("sub_int").Value = record_sub.Fields("sub_serial").Value
record_subTT.Update

```

End If

End Sub

Private Sub Save_change()

Dim i1, i2, aaa1, aaa2, aa1, aa2, i, j As Integer

Dim str As String

Dim stdtext, subtext As String

If (TabStrip1.SelectedItem.Index - 1) = 0 Then

```

record_regis.Fields("stdid").Value = Text2(0).Text
record_regis.Fields("subid").Value = Text2(1).Text
record_regis.Fields("yr").Value = Text2(2).Text
record_regis.Fields("term").Value = Text2(3).Text
record_regis.Fields("grade").Value = Text2(4).Text
record_regis.Fields("section").Value = Text2(5).Text
record_regis.Update

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

chkcombo = Combo3(1).ListIndex
Text2(6).Text = Combo3(1).List(chkcombo)
Text2(6).Text = Text2(6).Text + "/"
chkcombo = Combo3(0).ListIndex
Text2(6).Text = Text2(6).Text + Combo3(0).List(chkcombo)
Text2(6).Text = Text2(6).Text + "/"
chkcombo = Combo3(2).ListIndex
Text2(6).Text = Text2(6).Text + Combo3(2).List(chkcombo)

```

```

'-----

record_regisTT.FindLast "regis_int=" & record_regis.Fields("regis_serial").Value
record_regisTT.Edit
record_regisTT.Fields("Valid_to").Value = Text2(6).Text
record_regisTT.Fields("TxStop").Value = Format(Date, "mm/dd/yyyy")
record_regisTT.Update
stdtext = record_regisTT.Fields("std_int").Value
subtext = record_regisTT.Fields("sub_int").Value
record_regisTT.AddNew

If Form3.chk1 Then
    record_std.MoveFirst
    record_std.Move Form3.a1
    record_regisTT.Fields("std_int").Value = record_std.Fields("std_serial").Value
Else
    record_regisTT.Fields("std_int").Value = stdtext
End If

```

```

If Form3.chk2 Then
    record_sub.MoveFirst
    record_sub.Move Form3.a2
    record_regisTT.Fields("sub_int").Value = record_sub.Fields("sub_serial").Value
Else
    record_regisTT.Fields("sub_int").Value = subtext
End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

record_regisTT.Fields("yr").Value = Text2(2).Text
record_regisTT.Fields("term").Value = Text2(3).Text
record_regisTT.Fields("grade").Value = Text2(4).Text
record_regisTT.Fields("section").Value = Text2(5).Text

```

```

record_regisTT.Fields("regis_int").Value = record_regis.Fields("regis_serial").Value
record_regisTT.Fields("Valid_from").Value = Text2(6).Text
record_regisTT.Fields("Valid_to").Value = "12/31/9999"
record_regisTT.Fields("TxStart").Value = Format(Date, "mm/dd/yyyy")
record_regisTT.Fields("TxStop").Value = "12/31/9999"
record_regisTT.Update

```

End If

```

If (TabStrip1.SelectedItem.Index - 1) = 1 Then
record_std.Fields("stdid").Value = Text1(0).Text
record_std.Fields("tpren").Value = Text1(1).Text
record_std.Fields("epren").Value = Text1(2).Text
record_std.Fields("tname").Value = Text1(3).Text
record_std.Fields("ename").Value = Text1(4).Text
record_std.Fields("degree").Value = Text1(5).Text

```

```

If Form2.Option1.Value = True Then

```

```

record_std.Fields("stdtype").Value = "A"

```

```

ElseIf Form2.Option2.Value = True Then

```

```

    record_std.Fields("stdtype").Value = "B"

```

End If

```

record_std.Update

```

```

'-----

```

```

chkcombo = Combo2(1).ListIndex

```

```

Text1(7).Text = Combo2(1).List(chkcombo)

```

```

Text1(7).Text = Text1(7).Text + "/"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

chkcombo = Combo2(0).ListIndex
Text1(7).Text = Text1(7).Text + Combo2(0).List(chkcombo)
Text1(7).Text = Text1(7).Text + "/"
chkcombo = Combo2(2).ListIndex
Text1(7).Text = Text1(7).Text + Combo2(2).List(chkcombo)
'-----

record_stdTT.FindLast "std_int = " & record_std.Fields("std_serial").Value
record_stdTT.Edit
record_stdTT.Fields("Valid_to").Value = Text1(7).Text
record_stdTT.Fields("TxStop").Value = Format(Date, "mm/dd/yyyy")
record_stdTT.Update

record_stdTT.AddNew
record_stdTT.Fields("stdid").Value = Text1(0).Text
record_stdTT.Fields("tpren").Value = Text1(1).Text
record_stdTT.Fields("epren").Value = Text1(2).Text
record_stdTT.Fields("tname").Value = Text1(3).Text
record_stdTT.Fields("ename").Value = Text1(4).Text
record_stdTT.Fields("degree").Value = Text1(5).Text
If Form2.Option1.Value = True Then
record_stdTT.Fields("stdtype").Value = "A"
ElseIf Form2.Option2.Value = True Then
record_stdTT.Fields("stdtype").Value = "B"
End If

record_stdTT.Fields("valid_from").Value = Text1(7).Text
record_stdTT.Fields("valid_to").Value = "12/31/9999"
record_stdTT.Fields("TxStart").Value = Format(Date, "mm/dd/yyyy")
record_stdTT.Fields("TxStop").Value = "12/31/9999"

record_stdTT.Fields("std_int").Value = record_std.Fields("std_serial").Value
record_stdTT.Update

End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If (TabStrip1.SelectedItem.Index - 1) = 2 Then
    record_sub.Fields("subid").Value = Text3(0).Text
    record_sub.Fields("ename").Value = Text3(1).Text
    record_sub.Fields("tname").Value = Text3(2).Text
    record_sub.Fields("presub1").Value = Text3(3).Text
    record_sub.Fields("presub2").Value = Text3(4).Text
    record_sub.Fields("credit").Value = Text3(5).Text
    record_sub.Fields("type").Value = Text3(6).Text
    record_sub.Update

'-----
chkcombo = Combo1(1).ListIndex
Text3(7).Text = Combo1(1).List(chkcombo)
Text3(7).Text = Text3(7).Text + "/"
chkcombo = Combo1(0).ListIndex
Text3(7).Text = Text3(7).Text + Combo1(0).List(chkcombo)
Text3(7).Text = Text3(7).Text + "/"
chkcombo = Combo1(2).ListIndex
Text3(7).Text = Text3(7).Text + Combo1(2).List(chkcombo)
'-----

record_subTT.FindLast "sub_int = " & record_sub.Fields("sub_serial").Value
record_subTT.Edit
record_subTT.Fields("Valid_to").Value = Text3(7).Text
record_subTT.Fields("TxStop").Value = Format(Date, "mm/dd/yyyy")
record_subTT.Update

record_subTT.AddNew
record_subTT.Fields("subid").Value = Text3(0).Text
record_subTT.Fields("ename").Value = Text3(1).Text
record_subTT.Fields("tname").Value = Text3(2).Text
record_subTT.Fields("presub1").Value = Text3(3).Text
record_subTT.Fields("presub2").Value = Text3(4).Text

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

record_subTT.Fields("credit").Value = Text3(5).Text
record_subTT.Fields("type").Value = Text3(6).Text
record_subTT.Fields("valid_from").Value = Text3(7).Text
record_subTT.Fields("valid_to").Value = "12/31/9999"
record_subTT.Fields("TxStart").Value = Format(Date, "dd/mm/yyyy")
record_subTT.Fields("TxStop").Value = "12/31/9999"

record_subTT.Fields("sub_int").Value = record_sub.Fields("sub_serial").Value
record_subTT.Update

```

```
End If
```

```
End Sub
```

```
Private Sub Form_Terminate()
```

```

record_regis.Close
record_std.Close
record_sub.Close
record_regisTT.Close
record_stdTT.Close
record_subTT.Close
databas.Close

```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
End
```

```
End Sub
```

```
Private Sub mnuabout_Click()
```

```

AboutForm.Visible = True
AboutForm.Show

```

```
End Sub
```

```
Private Sub mnuexit_Click()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End
```

```
End Sub
```

```
Private Sub mnufirst_Click()
```

```
    first_click
```

```
End Sub
```

```
Private Sub mnulast_Click()
```

```
    last_click
```

```
End Sub
```

```
Private Sub mnunext_Click()
```

```
    next_click
```

```
End Sub
```

```
Private Sub mnuprevious_Click()
```

```
    previous_click
```

```
End Sub
```

```
Private Sub TabStrip1_Click()
```

```
    Frame1(TabStrip1.SelectedItem.Index - 1).ZOrder 0
```

```
    If (TabStrip1.SelectedItem.Index - 1 = 0) Then
```

```
        delregis.Enabled = True
```

```
        delstd.Enabled = False
```

```
        delsub.Enabled = False
```

```
        chksave = False
```

```
        If (record_regis.RecordCount = 0) Then
```

```
            MsgBox "Table registration have no data insert it now", vbOKOnly + vbExclamation, "No data"
```

```
            i1 = 0
```

```
            insert_click
```

```
        End If
```

```
    End If
```

```
    If (TabStrip1.SelectedItem.Index - 1 = 1) Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

delstd.Enabled = True
delregis.Enabled = False
delsub.Enabled = False
chksave = False

If (record_std.RecordCount = 0) Then
MsgBox "Table student have no data insert it now", vbOKOnly + vbExclamation, "No data"
j1 = 0
insert_click
End If

```

```
End If
```

```
If (TabStrip1.SelectedItem.Index - 1 = 2) Then
```

```

delsub.Enabled = True
delregis.Enabled = False
delstd.Enabled = False
chksave = False

If (record_sub.RecordCount = 0) Then
MsgBox "Table subject have no data insert it now", vbOKOnly + vbExclamation, "No data"
k1 = 0
insert_click
End If

```

```
End If
```

```
Call fillvalue
```

```
End Sub
```

```
Public Sub insert_click()
```

```
Dim msg1 As String
```

```
If (TabStrip1.SelectedItem.Index - 1) = 0 Then
```

```
If i1 = 0 Then
```

```

regisTimeEn_true
record_regis.AddNew
record_regisTT.AddNew
clear_click
chksave = True

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

i1 = 1
Else
    If (Text2(0).Text = "") Or (Text2(1).Text = "") Or (Text2(2).Text = "") Or (Text2(3).Text = "") Or
(Combo3(0).Text = "") Or (Combo3(1).Text = "") Or (Combo3(2).Text = "") Then
        MsgBox "Student_id, Subject_id, Year and Term is Primary Key can't have null value OR Not
Enter Valid from Date", vbOKOnly + vbExclamation, "Input Error"
    Else
        Form1.Enabled = False
        Form5.Option1.Value = True
        Form5.Visible = True
        Form5.Show
        chksave = False
    End If
End If
End If

If (TabStrip1.SelectedItem.Index - 1) = 1 Then
If j1 = 0 Then
    stdTimeEn_true
    record_std.AddNew
    record_stdTT.AddNew
    clear_click
    j1 = 1
    chksave = True
Else
    If (Text1(0).Text = "") Or (Combo2(0).Text = "") Or (Combo2(1).Text = "") Or (Combo2(2).Text
= "") Then
        MsgBox "Student_id is Primary Key can't have null value OR Not Enter Valid from Date",
vbOKOnly + vbExclamation, "Input Error"
    Else
        Form1.Enabled = False
        Form5.Option1.Value = True
        Form5.Visible = True
        Form5.Show

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        chksave = False
    End If
End If
End If

If (TabStrip1.SelectedItem.Index - 1) = 2 Then
    If k1 = 0 Then
        subTimeEn_true
        record_sub.AddNew
        record_subTT.AddNew
        clear_click
        chksave = True
        k1 = 1
    Else
        If (Text3(0) = "") Or (Combo1(0).Text = "") Or (Combo1(1).Text = "") Or (Combo1(2).Text = "")
Then
            MsgBox "Subject_id is Primary Key can't have null value", vbOKOnly + vbExclamation, "Input
Error"
        Else
            Form1.Enabled = False
            Form5.Option1.Value = True
            Form5.Visible = True
            Form5.Show
            chksave = False
        End If
    End If
End If
End If

Toolbar1(0).Buttons.Item(3).Enabled = False
'chksave = True
End Sub

Public Sub clear_click()
    Dim i As Integer

    If (TabStrip1.SelectedItem.Index - 1) = 0 Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

For i = 0 To 6
    Text2(i).Text = ""
Next i
End If

```

```

If (TabStrip1.SelectedItem.Index - 1) = 1 Then
    For i = 0 To 7
        Text1(i).Text = ""
    Next i
End If

```

```

If (TabStrip1.SelectedItem.Index - 1) = 2 Then
    For i = 0 To 7
        Text3(i).Text = ""
    Next i
End If

```

```
End Sub
```

```
Private Sub first_click()
```

```

If (TabStrip1.SelectedItem.Index - 1) = 0 Then
    If record_regis.RecordCount <> 0 Then
        record_regis.MoveFirst
        Text2(6).Enabled = False
        i1 = 0
        Call fillvalue
    Else: MsgBox "No data in registration Please insert before", vbOKOnly + vbExclamation, "Data
Empty"

```

```
        End If
```

```
    End If
```

```
    If (TabStrip1.SelectedItem.Index - 1) = 1 Then
```

```
        If record_std.RecordCount <> 0 Then
```

```
            Text1(7).Enabled = False
```

```
            record_std.MoveFirst
```

```
            j1 = 0
```

```
            Call fillvalue
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Else: MsgBox "No data in student Please insert before", vbOKOnly + vbExclamation, "Data
Empty"
End If
End If
If (TabStrip1.SelectedItem.Index - 1) = 2 Then
If record_sub.RecordCount <> 0 Then
Text3(7).Enabled = False
record_sub.MoveFirst
k1 = 0
Call fillvalue
Else: MsgBox "No data in subject Please insert before", vbOKOnly + vbExclamation, "Data
Empty"
End If
End If
chksave = False
End Sub
Private Sub previous_click()
If (TabStrip1.SelectedItem.Index - 1) = 0 Then
If record_regis.RecordCount <> 0 Then
record_regis.MovePrevious
If record_regis.BOF Then
MsgBox "This record is beginning of table", vbOKOnly + vbExclamation
record_regis.MoveFirst
End If
i1 = 0
Call fillvalue
Else: MsgBox "No data in registration Please insert before", vbOKOnly + vbExclamation, "Data
Empty"
End If
End If

If (TabStrip1.SelectedItem.Index - 1) = 1 Then
If record_std.RecordCount <> 0 Then
record_std.MovePrevious

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If record_std.BOF Then
    MsgBox "This record is beginning of table", vbOKOnly + vbExclamation
    record_std.MoveFirst
End If
j1 = 0
Call fillvalue
Else: MsgBox "No data in student Please insert before", vbOKOnly + vbExclamation, "Data
Empty"
End If
End If

If (TabStrip1.SelectedItem.Index - 1) = 2 Then
    If record_sub.RecordCount < 0 Then
        record_sub.MovePrevious
    If record_sub.BOF Then
        MsgBox "This record is beginning of table", vbOKOnly + vbExclamation
        record_sub.MoveFirst
    End If
    k1 = 0
    Call fillvalue
    Else: MsgBox "No data in subject Please insert before", vbOKOnly + vbExclamation, "Data
Empty"
    End If
End If
chksave = False
End Sub

Private Sub next_click()
    If (TabStrip1.SelectedItem.Index - 1) = 0 Then
        If record_regis.RecordCount < 0 Then
            record_regis.MoveNext
        If record_regis.EOF Then
            MsgBox "This record is end of table", vbOKOnly + vbExclamation
            record_regis.MoveLast
        End If
    End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

i1 = 0
Call fillvalue
Else: MsgBox "No data in registration Please insert before", vbOKOnly + vbExclamation, "Data
Empty"
End If
End If

If (TabStrip1.SelectedItem.Index - 1) = 1 Then
If record_std.RecordCount <> 0 Then
record_std.MoveNext
If record_std.EOF Then
MsgBox "This record is end of table", vbOKOnly + vbExclamation
record_std.MoveLast
End If
j1 = 0
Call fillvalue
Else: MsgBox "No data in student Please insert before", vbOKOnly + vbExclamation, "Data
Empty"
End If
End If

If (TabStrip1.SelectedItem.Index - 1) = 2 Then
If record_sub.RecordCount <> 0 Then
record_sub.MoveNext
If record_sub.EOF Then
MsgBox "This record is end of table", vbOKOnly + vbExclamation
record_sub.MoveLast
End If
k1 = 0
Call fillvalue
Else: MsgBox "No data in subject Please insert before", vbOKOnly + vbExclamation, "Data
Empty"
End If
End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

chksave = False
End Sub
Private Sub last_click()
    If (TabStrip1.SelectedItem.Index - 1) = 0 Then
        If record_regis.RecordCount <> 0 Then
            record_regis.MoveLast
            i1 = 0
            Call fillvalue
        Else: MsgBox "No data in registration Please insert before", vbOKOnly + vbExclamation, "Data
Empty"
        End If
    End If
    If (TabStrip1.SelectedItem.Index - 1) = 1 Then
        If record_std.RecordCount <> 0 Then
            record_std.MoveLast
            j1 = 0
            Call fillvalue
        Else: MsgBox "No data in student Please insert before", vbOKOnly + vbExclamation, "Data
Empty"
        End If
    End If
    If (TabStrip1.SelectedItem.Index - 1) = 2 Then
        If record_sub.RecordCount <> 0 Then
            record_sub.MoveLast
            k1 = 0
            Call fillvalue
        Else: MsgBox "No data in subject Please insert before", vbOKOnly + vbExclamation, "Data
Empty"
        End If
    End If
    chksave = False
End Sub
Private Sub save_click()
    Dim msg2 As String

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

'On Error GoTo err_enter
If (TabStrip1.SelectedItem.Index - 1) = 0 Then
    If (Combo3(0).Text = "") Or (Combo3(1).Text = "") Or (Combo3(2).Text = "") Then
        MsgBox "Please enter valid form date", vbOKOnly + vbExclamation, "Please enter"
    Else
        msg2 = MsgBox("Are you sure to save change to data", vbOKCancel + vbQuestion, "Save data")
        If msg2 = vbOK Then
            record_regis.Edit
            record_regisTT.Edit
            Save_change
            Combo3(0).ListIndex = 0
            Combo3(1).ListIndex = 0
            Combo3(2).ListIndex = 10
            regisTimeEn_false
        End If
    End If
End If
If (TabStrip1.SelectedItem.Index - 1) = 1 Then
    If (Combo2(0).Text = "") Or (Combo2(1).Text = "") Or (Combo2(2).Text = "") Then
        MsgBox "Please enter valid form date", vbOKOnly + vbExclamation, "Please enter"
    Else
        msg2 = MsgBox("Are you sure to save change to data", vbOKCancel + vbQuestion, "Save data")
        If msg2 = vbOK Then
            record_std.Edit
            record_stdTT.Edit
            Save_change
            Combo2(0).ListIndex = 0
            Combo2(1).ListIndex = 0
            Combo2(2).ListIndex = 10
            stdTimeEn_false
        End If
    End If
End If
If (TabStrip1.SelectedItem.Index - 1) = 2 Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If (Combo1(0).Text = "") Or (Combo1(1).Text = "") Or (Combo1(2).Text = "") Then
    MsgBox "Please enter valid form date", vbOKOnly + vbExclamation, "Please enter"
Else
    msg2 = MsgBox("Are you sure to save change to data", vbOKCancel + vbQuestion, "Save data")
    If msg2 = vbOK Then
        record_sub.Edit
        record_subTT.Edit
        Save_change
        Combo1(0).ListIndex = 0
        Combo1(1).ListIndex = 0
        Combo1(2).ListIndex = 10
        subTimeEn_false
    End If
End If
End If
'err_enter: MsgBox Err.Description, vbOKOnly + vbExclamation, "Input Error"
End Sub

Private Sub delete_click()
Dim msg2 As String
    If (TabStrip1.SelectedItem.Index - 1) = 0 Then
        If record_regis.RecordCount <> 0 Then
            Form1.Enabled = False
            Form4.Visible = True
            Form4.Show
        Else: MsgBox "No data in registration Please insert before", vbOKOnly + vbExclamation, "Data
Empty"
        End If
    End If
End If

    If (TabStrip1.SelectedItem.Index - 1) = 1 Then
        If record_std.RecordCount <> 0 Then
            Form1.Enabled = False
            Form4.Visible = True

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Form4.Show
```

```
Else: MsgBox "No data in student Please insert before", vbOKOnly + vbExclamation, "Data  
Empty"
```

```
End If
```

```
End If
```

```
If (TabStrip1.SelectedItem.Index - 1) = 2 Then
```

```
    If record_sub.RecordCount <> 0 Then
```

```
        Form1.Enabled = False
```

```
        Form4.Visible = True
```

```
        Form4.Show
```

```
    Else: MsgBox "No data in subject Please insert before", vbOKOnly + vbExclamation, "Data  
Empty"
```

```
    End If
```

```
End If
```

```
End Sub
```

```
Private Sub fillvalue()
```

```
Dim temp As String
```

```
If (TabStrip1.SelectedItem.Index - 1) = 0 Then
```

```
    Text2(0).Text = Format$(record_regis.Fields("stdid").Value)
```

```
    Text2(1).Text = Format$(record_regis.Fields("subid").Value)
```

```
    Text2(2).Text = Format$(record_regis.Fields("yr").Value)
```

```
    Text2(3).Text = Format$(record_regis.Fields("term").Value)
```

```
    Text2(4).Text = Format$(record_regis.Fields("grade").Value)
```

```
    Text2(5).Text = Format$(record_regis.Fields("section").Value)
```

```
    regisTimeEn_false
```

```
End If
```

```
If (TabStrip1.SelectedItem.Index - 1) = 1 Then
```

```
    Text1(0).Text = Format$(record_std.Fields("stdid").Value)
```

```
    Text1(1).Text = Format$(record_std.Fields("tpren").Value)
```

```
    Text1(2).Text = Format$(record_std.Fields("epren").Value)
```

```
    Text1(3).Text = Format$(record_std.Fields("tname").Value)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Text1(4).Text = Format$(record_std.Fields("ename").Value)
Text1(5).Text = Format$(record_std.Fields("degree").Value)
TextBox1.Text = Format$(record_std.Fields("stdtype").Value)
If TextBox1.Find("A") <> -1 Then
    Text1(6).Text = Form2.Option1.Caption
ElseIf TextBox1.Find("B") <> -1 Then
    Text1(6).Text = Form2.Option2.Caption
End If
stdTimeEn_false
End If

```

```

If (TabStrip1.SelectedItem.Index - 1) = 2 Then
    Text3(0).Text = Format$(record_sub.Fields("subid").Value)
    Text3(1).Text = Format$(record_sub.Fields("ename").Value)
    Text3(2).Text = Format$(record_sub.Fields("tname").Value)
    Text3(3).Text = Format$(record_sub.Fields("presub1").Value)
    Text3(4).Text = Format$(record_sub.Fields("presub2").Value)
    Text3(5).Text = Format$(record_sub.Fields("credit").Value)
    Text3(6).Text = Format$(record_sub.Fields("type").Value)
    subTimeEn_false
End If
Toolbar1(0).Buttons.Item(3).Enabled = False
End Sub

```

```

Private Sub Text1_Change(Index As Integer)
    stdTimeEn_true
    If chksave = False Then
        Toolbar1(0).Buttons.Item(3).Enabled = True
    End If
End Sub

```

```

Private Sub Text2_Change(Indcx As Integer)
    regisTimeEn_true
    If chksave = False Then
        Toolbar1(0).Buttons.Item(3).Enabled = True
    End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End If
```

```
End Sub
```

```
Private Sub Text3_Change(Index As Integer)
```

```
    subTimeEn_true
```

```
    If chksave = False Then
```

```
        Toolbar1(0).Buttons.Item(3).Enabled = True
```

```
    End If
```

```
End Sub
```

```
Private Sub Toolbar1_ButtonClick(Index As Integer, ByVal Button As MSComctlLib.Button)
```

```
    Select Case Button.Key
```

```
        Case "insert"
```

```
            insert_click
```

```
        Case "clear"
```

```
            clear_click
```

```
        Case "first"
```

```
            first_click
```

```
        Case "previous"
```

```
            previous_click
```

```
        Case "next"
```

```
            next_click
```

```
        Case "last"
```

```
            last_click
```

```
        Case "save"
```

```
            save_click
```

```
        Case "delete"
```

```
            delete_click
```

```
        Case "exit"
```

```
            Form_Terminate
```

```
    End
```

```
End Select
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source Code ในส่วนของ Form2.frm

```

Private Sub Command1_Click()
    If Form2.Caption = "Pre name (Thai)" Then
        Form1.Enabled = True
        Form1.Text1(1) = List1.Text
        Form1.Text1(1).SetFocus
    End If
    If Form2.Caption = "Pre name (English)" Then
        Form1.Enabled = True
        Form1.Text1(2) = List2.Text
        Form1.Text1(2).SetFocus
    End If
    If Form2.Caption = "Select Degree" Then
        Form1.Enabled = True
        Form1.Text1(5) = List3.Text
        Form1.Text1(5).SetFocus
    End If
    If Form2.Caption = "Select Study type" Then
        Form1.Enabled = True
        If Option1.Value = True Then
            Form1.Text1(6) = Option1.Caption
        End If
        If Option2.Value = True Then
            Form1.Text1(6) = Option2.Caption
        End If
        Form1.Text1(6).SetFocus
    End If
    If Form2.Caption = "Select Subject type" Then
        Form1.Enabled = True
        Form1.Text3(6) = List5.Text
        Form1.Text3(6).SetFocus
    End If
    If Form2.Caption = "Select Grade" Then
        Form1.Enabled = True

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Form1.Text2(4) = List6.Text
Form1.Text2(4).SetFocus
End If
Form2.Hide
Form1.Enabled = True
Form1.Show
End Sub

```

```

Private Sub Command2_Click()
Form2.Hide
Form1.Enabled = True
Form1.Show
End Sub

```

```

Private Sub Form_Load()
List1.Selected(0) = True
List2.Selected(0) = True
List3.Selected(0) = True
List5.Selected(0) = True
End Sub

```

```

Private Sub Form_Terminate()
Form1.Show
Form1.Enabled = True
End Sub

```

```

Private Sub Form_Unload(Cancel As Integer)
Form1.Show
Form1.Enabled = True
End Sub

```

Source Code ในส่วนของ Form3.frm

```

Public a1 As Integer
Public a2 As Integer

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Public chk1, chk2 As Boolean
Private Sub Command1_Click()
    If Form3.Caption = "เอนิเฌเฌเฌเฌเฌเฌ" Then
        Form1.Text2(0).Text = List1.Text
        Form1.Enabled = True
        Form1.Text2(0).SetFocus
        chk1 = True
    End If
    If Form3.Caption = "เอนิเฌเฌเฌเฌเฌเฌ" Then
        Form1.Text2(1).Text = List2.Text
        Form1.Enabled = True
        Form1.Text2(1).SetFocus
        chk2 = True
    End If
End Sub

Private Sub Command2_Click()
    Form3.Hide
    Form1.Enabled = True
    Form1.Show
End Sub

Private Sub Form_Terminate()
    Form1.Enabled = True
    Form1.Show
End Sub

```

```

Private Sub Form_Unload(Cancel As Integer)
    Form1.Enabled = True
    Form1.Show
End Sub

```

```

Private Sub List1_Click()
    Dim ii As Integer

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ii = List1.ListIndex
a1 = ii
Form1.record_std.MoveFirst
Form1.record_std.Move ii
Text1.Text = ii
Text1.Text = Form1.record_std.Fields("tname").Value
End Sub

```

```

Private Sub List2_Click()
Dim ii As Integer
ii = List2.ListIndex
a2 = ii
Form1.record_sub.MoveFirst
Form1.record_sub.Move ii
Text2.Text = ii
Text2.Text = Form1.record_sub.Fields("tname").Value
End Sub

```

Source Code ในส่วนของ Form5.frm

```

Private Sub Command1_Click()
If (Form1.TabStrip1.SelectedItem.Index - 1) = 0 Then
Form1.Save_link
Form1.record_regis.MoveLast
Form1.Label5.Caption = Form1.record_regis.RecordCount

If Option1.Value = True Then
Form1.record_regis.AddNew
Form1.record_regisTT.AddNew
Form1.clear_click
Form5.Visible = False
Form5.Hide
Form1.Enabled = True
Form1.Show

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Else: Form5.Visible = False
    Form5.Hide
    Form1.Enabled = True
    Form1.Show
    Form1.i1 = 0
    Form1.regisTimeEn_false
End If
    Form1.Combo3(0).ListIndex = 0
    Form1.Combo3(1).ListIndex = 0
    Form1.Combo3(2).ListIndex = 10
End If

If (Form1.TabStrip1.SelectedItem.Index - 1) = 1 Then
    Form1.Save_link
    Form1.record_std.MoveLast
    Form1.Label7.Caption = Form1.record_std.RecordCount

If Option1.Value = True Then
    Form1.record_std.AddNew
    Form1.record_stdTT.AddNew
    Form1.clear_click
    Form5.Visible = False
    Form5.Hide
    Form1.Enabled = True
    Form1.Show
Else: Form5.Visible = False
    Form5.Hide
    Form1.Enabled = True
    Form1.Show
    Form1.j1 = 0
    Form1.stdTimeEn_false
End If
    Form1.Combo2(0).ListIndex = 0
    Form1.Combo2(1).ListIndex = 0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Form1.Combo2(2).ListIndex = 10
```

```
End If
```

```
If (Form1.TabStrip1.SelectedItem.Index - 1) = 2 Then
```

```
Form1.Save_link
```

```
Form1.record_sub.MoveLast
```

```
Form1.Label9.Caption = Form1.record_sub.RecordCount
```

```
If Option1.Value = True Then
```

```
Form1.record_sub.AddNew
```

```
Form1.record_subTT.AddNew
```

```
Form1.clear_click
```

```
Form5.Visible = False
```

```
Form5.Hide
```

```
Form1.Enabled = True
```

```
Form1.Show
```

```
Else: Form5.Visible = False
```

```
Form5.Hide
```

```
Form1.Enabled = True
```

```
Form1.Show
```

```
Form1.k1 = 0
```

```
Form1.subTimeEn_false
```

```
End If
```

```
Form1.Combo1(0).ListIndex = 0
```

```
Form1.Combo1(1).ListIndex = 0
```

```
Form1.Combo1(2).ListIndex = 10
```

```
End If
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
Form5.Visible = True
```

```
Form5.Hide
```

```
Form1.Enabled = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Form1.Show
End Sub
```

Source Code ในส่วนของ delete.frm

```
Private Sub Command1_Click()
Dim chkcombo As Integer
If (Form1.TabStrip1.SelectedItem.Index - 1) = 0 Then
chkcombo = Combo1(1).ListIndex
Form1.Text2(6).Text = Combo1(1).List(chkcombo)
Form1.Text2(6).Text = Form1.Text2(6).Text + "/"
chkcombo = Combo1(0).ListIndex
Form1.Text2(6).Text = Form1.Text2(6).Text + Combo1(0).List(chkcombo)
Form1.Text2(6).Text = Form1.Text2(6).Text + "/"
chkcombo = Combo1(2).ListIndex
Form1.Text2(6).Text = Form1.Text2(6).Text + Combo1(2).List(chkcombo)

Form1.record_regisTT.FindLast "regis_int = " & Form1.record_regis.Fields("regis_serial").Value
Form1.record_regisTT.Edit
Form1.record_regisTT.Fields("Valid_to").Value = Form1.Text2(6).Text
Form1.record_regisTT.Fields("TxStop").Value = Format(Date, "mm/dd/yyyy")
Form1.record_regisTT.Update
Form1.record_regis.Delete
If Form1.record_regis.RecordCount <> 0 Then
Form1.record_regis.MoveLast
Form1.Label5.Caption = Form1.record_regis.RecordCount
Form1.clear_click
Form1.record_regis.MoveFirst
Else: MsgBox "No data in registration table please insert data", vbOKOnly + vbExclamation,
"Data Empty"
il = 0
Form1.clear_click
Form1.insert_click
End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End If

If (Form1.TabStrip1.SelectedItem.Index - 1) = 1 Then

 chkcombo = Combo1(1).ListIndex

 Form1.Text1(7).Text = Combo1(1).List(chkcombo)

 Form1.Text1(7).Text = Form1.Text1(7).Text + "/"

 chkcombo = Combo1(0).ListIndex

 Form1.Text1(7).Text = Form1.Text1(7).Text + Combo1(0).List(chkcombo)

 Form1.Text1(7).Text = Form1.Text1(7).Text + "/"

 chkcombo = Combo1(2).ListIndex

 Form1.Text1(7).Text = Form1.Text1(7).Text + Combo1(2).List(chkcombo)

 Form1.record_stdTT.FindLast "std_int = " & Form1.record_std.Fields("std_serial").Value

 Form1.record_stdTT.Edit

 Form1.record_stdTT.Fields("Valid_to").Value = Form1.Text1(7).Text

 Form1.record_stdTT.Fields("TxStop").Value = Format(Date, "mm/dd/yyyy")

 Form1.record_stdTT.Update

 Form1.record_std.Delete

 If Form1.record_std.RecordCount <> 0 Then

 Form1.record_std.MoveLast

 Form1.Label7.Caption = Form1.record_std.RecordCount

 Form1.record_std.MoveFirst

 Else: MsgBox "No data in student table please insert data", vbOKOnly + vbExclamation, "Data

Empty"

 i1 = 0

 Form1.clear_click

 Form1.insert_click

 End If

End If

If (Form1.TabStrip1.SelectedItem.Index - 1) = 2 Then

 chkcombo = Combo1(1).ListIndex

 Form1.Text3(7).Text = Combo1(1).List(chkcombo)

 Form1.Text3(7).Text = Form1.Text3(7).Text + "/"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

chkcombo = Combo1(0).ListIndex
Form1.Text3(7).Text = Form1.Text3(7).Text + Combo1(0).List(chkcombo)
Form1.Text3(7).Text = Form1.Text3(7).Text + "/"
chkcombo = Combo1(2).ListIndex
Form1.Text3(7).Text = Form1.Text3(7).Text + Combo1(2).List(chkcombo)

Form1.record_subTT.FindLast "sub_int = " & Form1.record_sub.Fields("sub_serial").Value
Form1.record_subTT.Edit
Form1.record_subTT.Fields("Valid_to").Value = Form1.Text3(7).Text
Form1.record_subTT.Fields("TxStop").Value = Format(Date, "mm/dd/yyyy")
Form1.record_subTT.Update
Form1.record_sub.Delete
If Form1.record_sub.RecordCount <> 0 Then
    Form1.record_sub.MoveLast
    Form1.Label9.Caption = Form1.record_sub.RecordCount
    Form1.record_sub.MoveFirst
Else: MsgBox "No data in subject table please insert data", vbOKOnly + vbExclamation, "Data
Empty"
    i1 = 0
    Form1.clear_click
    Form1.insert_click
End If
End If
Form4.Hide
Form4.Visible = False
Form1.Enabled = True
Form1.Show
End Sub

Private Sub Command2_Click()
    Form4.Hide
    Form4.Visible = False
    Form1.Enabled = True
    Form1.Show

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

Source Code ในส่วนของ About.frm

VERSION 5.00

Begin VB.Form Form6

Caption = "About"
 ClientHeight = 3945
 ClientLeft = 3255
 ClientTop = 2745
 ClientWidth = 4710
 LinkTopic = "Form6"
 ScaleHeight = 3945
 ScaleWidth = 4710
 Visible = 0 'False

Begin VB.CommandButton Command1

Caption = "&OK"
 Height = 375
 Left = 1800
 TabIndex = 7
 Top = 3480
 Width = 975

End

Begin VB.Frame Frame1

Caption = "By"
 BeginProperty Font
 Name = "MS Sans Serif"
 Size = 8.25
 Charset = 222
 Weight = 700
 Underline = 0 'False
 Italic = 0 'False
 Strikethrough = 0 'False

EndProperty

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Height    = 1935
Left      = 240
TabIndex  = 2
Top       = 1440
Width     = 4215

```

```
Begin VB.Label Label6
```

```

Caption    = "Email : s9014508@kmitl.ac.th"
ForeColor  = &H00FF0000&
Height     = 375
Left       = 960
TabIndex   = 6
Top        = 1440
Width      = 2295

```

```
End
```

```
Begin VB.Label Label5
```

```

Caption    = "Saran Theingsakul 39014508 4D"
BeginProperty Font
    Name     = "MS Sans Serif"
    Size     = 8.25
    Charset  = 222
    Weight   = 700
    Underline = 0 'False
    Italic   = 0 'False
    Strikethrough = 0 'False
EndProperty

```

```
EndProperty
```

```

Height     = 375
Left       = 600
TabIndex   = 5
Top        = 1080
Width      = 3375

```

```
End
```

```
Begin VB.Label Label4
```

```

Caption    = "Email : s9014499@kmitl.ac.th"
ForeColor  = &H00FF0000&

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Height    = 375
Left      = 960
TabIndex  = 4
Top       = 720
Width     = 2655

```

```
End
```

```
Begin VB.Label Label3
```

```
Caption    = "Wuttichai Ratpathomwong 39014499 4D "
```

```
BeginProperty Font
```

```

Name      = "MS Sans Serif"
Size      = 8.25
Charset   = 222
Weight    = 700
Underline = 0 'False'
Italic    = 0 'False'
Strikethrough = 0 'False'

```

```
EndProperty
```

```

Height    = 375
Left      = 240
TabIndex  = 3
Top       = 360
Width     = 3615

```

```
End
```

```
End
```

```
Begin VB.Label Label2
```

```
Caption    = "Version 1.0"
```

```
BeginProperty Font
```

```

Name      = "MS Sans Serif"
Size      = 8.25
Charset   = 222
Weight    = 700
Underline = 0 'False'
Italic    = 0 'False'
Strikethrough = 0 'False'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
EndProperty
```

```
Height = 375
```

```
Left = 1800
```

```
TabIndex = 1
```

```
Top = 1080
```

```
Width = 975
```

```
End
```

```
Begin VB.Label Label1
```

```
Caption = "Registration and Information System Temporal Database Software "
```

```
BeginProperty Font
```

```
Name = "MS Sans Serif"
```

```
Size = 9.75
```

```
Charset = 222
```

```
Weight = 700
```

```
Underline = 0 'False'
```

```
Italic = 0 'False'
```

```
Strikethrough = 0 'False'
```

```
EndProperty
```

```
Height = 495
```

```
Left = 360
```

```
TabIndex = 0
```

```
Top = 360
```

```
Width = 3975
```

```
End
```

```
Begin VB.Shape Shape1
```

```
BorderColor = &H00C00000&
```

```
BorderWidth = 2
```

```
FillColor = &H0000FF00&
```

```
Height = 735
```

```
Left = 240
```

```
Top = 240
```

```
Width = 4215
```

```
End
```

```
End
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Attribute VB_Name = "Form6"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Private Sub Command1_Click()
    Form6.Visible = False
End Sub

```

2. โปรแกรม SQL Editor

Source Code ในส่วนของ Login.frm

```

Private Sub Command1_Click()
'On Error Resume Next
'EditForm.Data1.Connect =
"ODBC;DSN=student_dsn;DATABASE=database_student;UID=informix;PWD=informix;HOST=islalnd;SERVER=ol_island;SERVICE=turbo;PROTOCOL=olsocetp;"
EditForm.Data1.Connect = "ODBC;DSN=student_data;" &
"UID=" & User.Text & ";PWD=" & Passwprd.Text & ";"

EditForm.Show
LoginForm.Hide
End Sub

Private Sub Command2_Click()
End
End Sub

Private Sub Form_Load()
    EditForm.Hide
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source Code ในส่วนของ SqlEditor.frm

```
Public Change1 As Boolean
```

```
Public Change2 As Boolean
```

```
Dim chkhavetable As Boolean
```

```
Public Sub resize()
```

```
On Error Resume Next
```

```
tabstrip1.Width = EditForm.Width - 120
```

```
tabstrip1.Height = EditForm.Height - 1550
```

```
Frame1(0).Width = EditForm.Width - 240
```

```
Frame1(0).Height = EditForm.Height - 2000
```

```
Frame1(1).Width = EditForm.Width - 240
```

```
Frame1(1).Height = EditForm.Height - 2000
```

```
Frame1(2).Width = EditForm.Width - 240
```

```
Frame1(2).Height = EditForm.Height - 2000
```

```
Inputsql.Width = Frame1(0).Width - 240
```

```
Inputsql.Height = Frame1(0).Height - 350
```

```
OutputSql.Width = Frame1(1).Width - 240
```

```
OutputSql.Height = Frame1(1).Height - 350
```

```
TDBGridS1.Width = Frame1(2).Width - 240
```

```
TDBGridS1.Height = Frame1(2).Height - 920
```

```
Data1.Width = TDBGridS1.Width
```

```
Data1.Top = TDBGridS1.Top + TDBGridS1.Height + 200
```

```
End Sub
```

```
Private Sub changeTableName()
```

```
    Inputsql.Text = Inputsql.Text
```

```
    If Inputsql.Find("student_tt") = "-1" And Inputsql.Find("subject_tt") = "-1" And Inputsql.Find
```

```
    ("registration_tt") = "-1" Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If Inputsql.Find("student") <> "-1" Then
    Inputsql.SelText = "student_tt"
    chkhavetable = True
ElseIf Inputsql.Find("subject") <> "-1" Then
    Inputsql.SelText = "subject_tt"
    chkhavetable = True
ElseIf Inputsql.Find("registration") <> "-1" Then
    Inputsql.SelText = "registration_tt"
    chkhavetable = True
End If
Else: chkhavetable = True
End If
Inputsql.Text = Inputsql.Text
End Sub

Private Sub mapTsqlToSql()
Dim pos1 As Long
Dim pos2 As Long
Dim string1 As String
Dim where As String
Dim chk, chk2, chk3 As Integer
chk = 0
chkhavetable = False
Inputsql.Text = Inputsql.Text
OutputSql.Text = ""
where = "where"
'If Check1.Value = 0 Then
' MsgBox "This statement is not a temporal SQL", vbOKOnly + vbExclamation, "Warning"
'Else
'sequence valid query
If Inputsql.Find("validtime") = "-1" Then
Else: Inputsql.Text = Inputsql.Text
pos1 = Inputsql.Find("validtime")

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

changeTableName
Inputsql.SelStart = pos1 + 11
Inputsql.SelLength = 10000
Inputsql.SetFocus
string1 = Inputsql.SelText
If Inputsql.Find("*") = "-1" Then
    text3.Text = Inputsql.SelText
    pos2 = text3.Find("from")
    text3.SelStart = 0
    text3.SelLength = pos2 - 1
    OutputSql.Text = text3.SelText
    OutputSql.Text = OutputSql.Text + ",valid_from,valid_to " + Chr(10)
    text3.SelStart = pos2
    text3.SelLength = 10000
    OutputSql.Text = OutputSql.Text + text3.SelText
    Temp = True
Else: OutputSql.Text = string 1
    Temp = True
End If
chk = 1
End If
'nonsequence valid query
If Inputsql.Find("nonsequencevalid") = "-1" Then
Else: Inputsql.Text = Inputsql.Text
    pos1 = Inputsql.Find("nonsequencevalid")
    changeTableName
    Inputsql.SelStart = pos1 + 18
    Inputsql.SelLength = 10000
    Inputsql.SetFocus
    OutputSql.Text = Inputsql.SelText
    Temp = True
    chk = 1
End If
'current valid query

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

text3.Text = Inputsql.Find(wher)
Inputsq1.Text = Inputsq1.Text
chk2 = Inputsq1.Find("currentvalid")
Inputsq1.Text = Inputsq1.Text
If (chk2 <> -1) Then
  If text3.Text = "-1" Then
    pos1 = Inputsq1.Find("currentvalid")
    changeTableName
    Inputsq1.SelStart = pos1 + 14
    Inputsq1.SelLength = 10000
    Inputsq1.SetFocus
    OutputSql.Text = Inputsq1.SelText
    OutputSql.Text = OutputSql.Text + " where valid_to = #12/31/9456# "
    chk = 1
  Else
    pos1 = Inputsq1.Find("currentvalid")
    changeTableName
    Inputsq1.SelStart = pos1 + 14
    Inputsq1.SelLength = 10000
    Inputsq1.SetFocus
    OutputSql.Text = Inputsq1.SelText
    OutputSql.Text = OutputSql.Text + " and valid_to = #12/31/9456# "
    chk = 1
  End If
End If
'sequence transaction query
If Inputsq1.Find("transactiontime") = "-1" Then
Else: Inputsq1.Text = Inputsq1.Text
  pos1 = Inputsq1.Find("transactiontime")
  changeTableName
  Inputsq1.SelStart = pos1 + 16
  Inputsq1.SelLength = 10000
  Inputsq1.SetFocus
  string1 = Inputsq1.SelText

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If Inputsql.Find("*") = "-1" Then
    text3.Text = Inputsql.SelText
    pos2 = text3.Find("from")
    text3.SelStart = 0
    text3.SelLength = pos2 - 1
    OutputSql.Text = text3.SelText
    OutputSql.Text = OutputSql.Text + ",txstart,txstop " + Chr(10)
    text3.SelStart = pos2
    text3.SelLength = 10000
    OutputSql.Text = OutputSql.Text + text3.SelText
    Temp = True
Else: OutputSql.Text = string I
    Temp = True
End If
chk = 1
End If
'nonsequence transaction query
If Inputsql.Find("nonsequencetransaction") = "-1" Then
Else: Inputsql.Text = Inputsql.Text
    pos1 = Inputsql.Find("nonsequencetransaction")
    changeTableName
    Inputsql.SelStart = pos1 + 24
    Inputsql.SelLength = 10000
    Inputsql.SetFocus
    OutputSql.Text = Inputsql.SelText
    Temp = True
    chk = 1
End If
'current transaction query
text3.Text = Inputsql.Find(where)
Inputsql.Text = Inputsql.Text
chk3 = Inputsql.Find("currenttransaction")
Inputsql.Text = Inputsql.Text
If (chk3 <> -1) Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If text3.Text = "-1" Then
    pos1 = Inputsql.Find("currenttransaction")
    changeTableName
    Inputsql.SelStart = pos1 + 20
    Inputsql.SelLength = 10000
    Inputsql.SetFocus
    OutputSql.Text = Inputsql.SelText
    OutputSql.Text = OutputSql.Text + " where txstop = #12/31/9456# "
    chk = 1

```

```

Else
    pos1 = Inputsql.Find("currenttransaction")
    changeTableName
    Inputsql.SelStart = pos1 + 20
    Inputsql.SelLength = 10000
    Inputsql.SetFocus
    OutputSql.Text = Inputsql.SelText
    OutputSql.Text = OutputSql.Text + " and txstop = #12/31/9456# "
    chk = 1

```

```
End If
```

```
End If
```

```
If chk <> 1 Then
```

```
    MsgBox "This Statement is not format of temporal SQL", vbOKOnly + vbExclamation,
"Warning"
```

```
    ElseIf chkhavetable = False Then
```

```
        MsgBox "Table that you refer not have in database", vbOKOnly + vbExclamation, "Table"
```

```
        OutputSql.Text = ""
```

```
    Else: tabstrip1.Tabs(2).Selected = True
```

```
        Frame1(1).ZOrder 0
```

```
        OutputSql.SetFocus
```

```
    End If
```

```
'End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

Private Sub Form_Load()

'Data1.Connect =

"ODBC;DSN=student_dsn;DATABASE=student_data;UID=informix;PWD=informix;HOST=island;SERVER=ol_island;SERVICE=turbo;PROTOCOL=olsoctcp;"

LoginForm.Show

EditForm.Hide

Change1 = False

Change2 = False

Dim i As Integer

resize

' The name of the TabStrip is "tabstrip1."

' The Frame control is named "frame1."

For i = 0 To Frame1.Count - 1

With Frame1(i)

.Move tabstrip1.ClientLeft, _

tabstrip1.ClientTop, _

tabstrip1.ClientWidth, _

tabstrip1.ClientHeight

End With

Next i

' Bring the first fraTab control to the front.

Frame1(0).ZOrder 0

Unload frmSplash

frmSplash.Hide

End Sub

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Form_Resize()
```

```
    resize
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
    If CheckSave() = False Then End
```

```
End Sub
```

```
Private Sub InputSql_Change()
```

```
    Change2 = True
```

```
End Sub
```

```
Private Sub mapsql_Click()
```

```
    mapTsqlToSql
```

```
End Sub
```

```
Private Sub mnuAbout_Click()
```

```
    AboutForm.Show
```

```
End Sub
```

```
Private Sub mnuBuilder_Click()
```

```
    Wizard.Show
```

```
End Sub
```

```
Private Sub mnuExit_Click()
```

```
    If CheckSave() = False Then End
```

```
End Sub
```

```
Private Sub mnuLogin_Click()
```

```
    EditForm.Hide
```

```
    LoginForm.Show
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub mnuNewstd_Click()
```

```
    Inputsql.Text = ""
```

```
End Sub
```

```
Private Sub mnuOpenStd_Click()
```

```
    Dialog1.DialogTitle = "Open Standard SQL"
```

```
    Dialog1.ShowOpen
```

```
    OutputSql.FileName = Dialog1.FileName
```

```
End Sub
```

```
Private Sub mnuOpenTemp_Click()
```

```
    Dialog1.DialogTitle = "Open Temporal SQL"
```

```
    Dialog1.ShowOpen
```

```
    Inputsql.FileName = Dialog1.FileName
```

```
End Sub
```

```
Private Sub mnuQuery_Click()
```

```
    On Error GoTo Err_Enter
```

```
    If Len(OutputSql.Text) = 0 Then
```

```
        MsgBox "ãÁèÁŒ Statement ã' æ àÁÁ", vbOKOnly, "Error SQL"
```

```
        OutputSql.SetFocus
```

```
        Exit Sub
```

```
    Else
```

```
        With Data1
```

```
            .RecordSource = OutputSql.Text
```

```
            .Refresh
```

```
            .Caption = "Return " & str(.Recordset.RecordCount) & " records"
```

```
        End With
```

```
        tabstrip1.Tabs(3).Selected = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Frame1(2).ZOrder 0

Exit Sub
End If

Err_Enter:
If Err.Description = "Object required" Then

Else: MsgBox Err.Description, vbOKOnly + vbExclamation, "Error in SQL"
OutputSql.SetFocus
End If

End Sub

Private Sub mnuStandard_Sql_Click()
'Create NewFile
Dialog1.DialogTitle = "Save as Standard SQL"
Dialog1.DefaultExt = ".sql"
Dialog1.ShowSave
Dim filename1 As String
filename1 = Dialog1.FileName

Dim fso As New FileSystemObject
Dim objFil As File
Dim objts As TextStream
On Error GoTo Error_Hand
fso.CreateTextFile (filename1)
Set objFil = fso.GetFile(filename1)
Set objts = objFil.OpenAsTextStream(ForWriting)
objts.Write OutputSql.Text
objts.Close
Error_Hand:
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub mnuNewTemp_Click()
```

```
    OutputSql.Text = ""
```

```
End Sub
```

```
Private Sub mnuTemporal_Sql_Click()
```

```
    Dialog1.DialogTitle = "Save as Temporal SQL"
```

```
    Dialog1.DefaultExt = ".tql"
```

```
    Dialog1.ShowSave
```

```
Dim filename1 As String
```

```
    filename1 = Dialog1.FileName
```

```
Dim fso As New FileSystemObject
```

```
Dim objFil As File
```

```
Dim objts As TextStream
```

```
On Error GoTo Error_Hand
```

```
    fso.CreateTextFile (filename1)
```

```
    Set objFil = fso.GetFile(filename1)
```

```
    Set objts = objFil.OpenAsTextStream(ForWriting)
```

```
    objts.Write Inputsql.Text
```

```
    objts.Close
```

```
Error_Hand:
```

```
End Sub
```

```
Private Sub OutputSql_Change()
```

```
    Change1 = True
```

```
End Sub
```

```
Private Function CheckSave() As Boolean
```

```
Dim Response
```

```
CheckSave = False
```

```
If Change1 Then
```

```
    Response = MsgBox("Do you want to save Standard Sql?". vbYesNoCancel, "Save Comfirm")
```

```
    If Response = vbYes Then
```

```
        mnuStandard_Sql_Click
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ElseIf Response = vbNo Then
ElseIf Response = vbCancel Then
    CheckSave = True
End If
End If

```

```

If Change2 Then
    Response = MsgBox("Do you want to save Temporal Sql?", vbYesNoCancel, "Save Comfirm")
    If Response = vbYes Then
        mnuTemporal_Sql_Click
    ElseIf Response = vbNo Then
    ElseIf Response = vbCancel Then
        CheckSave = True
    End If
End If

endd:
End Function

```

```

Private Sub TabStrip1_Click()
    Frame1(tabstrip1.SelectedItem.Index - 1).ZOrder 0
End Sub

```

```

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)

```

```

    Select Case Button.Key
    Case "NewTemporal"
        mnuNewTemp_Click
    Case "OpenTemporal"
        mnuOpenTemp_Click
    Case "SaveTemporal"
        mnuTemporal_Sql_Click

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Case "NewStandard"
    mnuNewstd_Click
Case "OpenStandard"
    mnuOpenStd_Click
Case "SaveStandard"
    mnuStandard_Sql_Click
Case "query"
    mnuQuery_Click
Case "map"
    mapsq_Click
Case "builder"
    mnuBuilder_Click
End Select
End Sub

```

Source Code ในส่วนของ Wizard.frm

```

Private mNode As Node
Dim keylist As Integer
Dim KeyIn As Integer
Dim IsAddTree As Boolean
Public Newline As String
Private help1 As String
Private help2 As String
Private help3 As String

'Cutdot will Cut pure TableName from x
Private Function CutDot(X As String) As String
Dim start_pos, end_pos As Long
    CutDot = X
    Temp.Text = X
    start_pos = Temp.Find(".") + 1
    Temp.SelStart = start_pos
    end_pos = 1000

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Temp.SelLength = end_pos - start_pos
CutDot = Temp.SelText
Temp.SelLength = Temp.SelLength - 2
CutDot = Temp.SelText

```

End Function

'Cutdot will Cut pure Columnname from x

Private Function cutdotpost(X As String) As String

Dim start_pos, end_pos As Long

cutdotpost = X

Temp.Text = X

end_pos = Temp.Find(".")

start_pos = 0

Temp.SelStart = start_pos

Temp.SelLength = end_pos - start_pos

cutdotpost = Temp.SelText

End Function

'CutSpacepost will return prestring with no space

Private Function CutSpacepost(X As String) As String

Dim start_pos, end_pos As Long

Dim Y As String

CutSpacepost = X

If x <> "" Then

Temp.Text = X

end_pos = Temp.Find(" ")

If end_pos <> -1 Then

start_pos = 0

Temp.SelStart = start_pos

Temp.SelLength = end_pos - start_pos

Y = Temp.SelText

CutSpacepost = Temp.SelText

End If

'End If

End Function

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Function FromTable() As String
    Dim i As Integer
    Dim j As Integer
    Dim string1, string2 As String

    Dim Is_Add As Boolean
    TempList.Clear
    For i = 0 To List1.ListCount - 1
        If TempList.ListCount > 0 Then
            ' string1 = List1.List(i)
            Is_Add = True
            For j = 0 To TempList.ListCount - 1
                If (TempList.List(j) = cutdotpost(List1.List(i))) Then
                    Is_Add = False
                End If
            Next j
            If Is_Add Then
                TempList.AddItem (cutdotpost(List1.List(i)))
            End If
        Else:
            TempList.AddItem (cutdotpost(List1.List(i)))
        End If
    Next i
    string2 = " "
    For i = 0 To TempList.ListCount - 1
        If i < TempList.ListCount - 1 Then
            string2 = string2 & TempList.List(i) & ", "
        Else: string2 = string2 & TempList.List(i)
        End If
    Next i
    FromTable = string2
End Function

Private Function add_order_by(X As String) As String

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim i, end_pos, start_pos As Long
```

```
Dim is_orderby As Boolean
```

```
Dim SortStr As String
```

```
temp2.Text = ""
```

```
is_orderby = False
```

```
For i = 0 To List1.ListCount - 1
```

```
temp2.Text = List1.List(i)
```

```
start_pos = temp2.Find(" ")
```

```
If start_pos <> -1 Then
```

```
start_pos = start_pos + 1
```

```
temp2.SelStart = start_pos
```

```
temp2.SelLength = 4
```

```
If ((temp2.SelText = " Asc") Or (temp2.SelText = " Des")) Then
```

```
SortStr = temp2.SelText
```

```
If is_orderby = False Then
```

```
is_orderby = True
```

```
X = X & "ORDER BY" & " " & _
```

```
CutSpacepost(List1.List(i)) & " " & SortStr
```

```
Else:
```

```
X = X & ", " & CutSpacepost(List1.List(i)) & " " & SortStr
```

```
End If
```

```
End If
```

```
End If
```

```
Next i
```

```
X = X
```

```
add_order_by = X
```

```
End Function
```

```
Private Sub AddTree1()
```

```
Dim i As Integer
```

```
Dim j As Integer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TreeView1.Nodes.Clear
Set mNode = TreeView1.Nodes.Add()
mNode.Text = "DATABASE"
mNode.Key = "Root1"
mNode.Image = "key1"
Dim nodX As Node
Dim nodY As Node
Dim keyj As String
Dim FieldName As String
For j = 0 To Data1.Database.TableDefs.Count - 1
If (Data1.Database.TableDefs(j).Attributes And _
(dbSystemObject + dbHiddenObject)) = 0 Then
If (Check1.Value = 0) Or (j Mod 2 = 1) Then
Set nodX = TreeView1.Nodes.Add("Root1", tvwChild)
keyj = j
nodX.Text = CutDot(Data1.Database.TableDefs(j).Name)
nodX.Key = keyj & " Table $%@#" & nodX.Text
nodX.Image = "key" & 1
For i = 0 To Data1.Database.TableDefs(j).Fields.Count - 1
FieldName = Data1.Database.TableDefs(j).Fields(i).Name
If (FieldName <> "valid_from") And (FieldName <> "valid_to") And _
(FieldName <> "txstart") And (FieldName <> "txstop") Then
Set nodY = TreeView1.Nodes.Add(nodX.Key, tvwChild)
keyi = i
nodY.Key = "child" & keyi & keyj & nodX.Text
nodY.Text = Data1.Database.TableDefs(j).Fields(i).Name
nodY.Image = "key" & 2
End If
Next i
End If
Next j

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
```

```
Private Sub AddTree2()
```

```
Dim i As Integer
```

```
Dim j As Integer
```

```
TreeView2.Nodes.Clear
```

```
Set mNode = TreeView2.Nodes.Add()
```

```
mNode.Text = "DATABASE"
```

```
mNode.Key = "Root1"
```

```
mNode.Image = "key1"
```

```
Dim nodX As Node
```

```
Dim nodY As Node
```

```
Dim keyj As String
```

```
For j = 0 To Data1.Database.TableDefs.Count - 1
```

```
If (Data1.Database.TableDefs(j).Attributes And _  
    (dbSystemObject + dbHiddenObject)) = 0 Then
```

```
If (Check1.Value = 0) Or (j Mod 2 = 1) Then
```

```
Set nodX = TreeView2.Nodes.Add("Root1", tvwChild)
```

```
keyj = j
```

```
nodX.Text = CutDot(Data1.Database.TableDefs(j).Name)
```

```
nodX.Key = keyj & " Table $%@#" & nodX.Text
```

```
nodX.Image = "key" & 1
```

```
For i = 0 To Data1.Database.TableDefs(j).Fields.Count - 1
```

```
Set nodY = TreeView2.Nodes.Add(nodX.Key, tvwChild)
```

```
keyi = i
```

```
nodY.Key = "child $%@#" & keyi & keyj & nodX.Text
```

```
nodY.Text = Data1.Database.TableDefs(j).Fields(i).Name
```

```
nodY.Image = "key" & 2
```

```
Next i
```

```
End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End If
Next j

End Sub

Private Sub Add_Click()
On Error Resume Next
Dim i As Integer
Dim have_list As Boolean
Dim last_node As Node
Dim temp_node As Node
Dim temp_node2 As Node
have_data = False
Temp.Text = TreeView1.SelectedItem.Key

If Temp.Find(" Table $%#@#" ) = -1 Then

For i = 0 To List1.ListCount
If TreeView1.SelectedItem.Parent.Text & "." & TreeView1.SelectedItem.Text = _
CutSpacepost(List1.List(i)) Then
have_list = True
End If
Next i
If Not have_list Then

List1.AddItem TreeView1.SelectedItem.Parent.Text & "." & TreeView1.SelectedItem.Text
End If

Else:
Set last_node = TreeView1.SelectedItem.Child.LastSibling
Set temp_node = TreeView1.SelectedItem.Child.FirstSibling
'Label6.Caption = last_node.Text
Do While temp_node.Key <> last_node.Key
have_list = False
For i = 0 To List1.ListCount

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If TreeView1.SelectedItem.Text & "." & temp_node.Text = _
CutSpacepost(List1.List(i)) Then
    have_list = True
End If
Next i
If Not have_list Then
    List1.AddItem TreeView1.SelectedItem.Text & "." & temp_node.Text
End If

Set temp_node2 = temp_node.Next
Set temp_node = temp_node2
Loop
End If

'Label6.Caption = TreeView1.SelectedItem.Child.FirstSibling.Text
End Sub
Private Sub Addcondition(strin As String)
    Dim str As String
    On Error Resume Next
    Temp = TreeView2.SelectedItem.Key
    If Temp.Find("child $%@#") <> -1 Then
        str = Combo4.List(Combo4.ListIndex) & " "
        If str = "Same " Then str = ""
        If RichTextBox3.Text <> "" Then
            RichTextBox3.Text = RichTextBox3.Text & " " & strin & " ("
        Else:
            RichTextBox3.Text = "(" & RichTextBox3.Text
        End If
        RichTextBox3.Text = RichTextBox3.Text & str & TreeView2.SelectedItem.Parent.Text & "." _
        & TreeView2.SelectedItem.Text & " " & Combo3.List(Combo3.ListIndex) _
        & " )"
    End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Addcon_Click()
```

```
  Addcondition ("AND")
```

```
End Sub
```

```
Private Sub Buildcmd_Click()
```

```
  Dim Space, Comma As String
```

```
  Dim i As Integer
```

```
  Dim Xstr As String
```

```
  Space = " "
```

```
  RichTextBox2.Text = "SELECT" & Space
```

```
  For i = 0 To List1.ListCount - 1
```

```
    Comma = ""
```

```
    If (i < (List1.ListCount - 1)) Then
```

```
      Comma = ","
```

```
    End If
```

```
    RichTextBox2.Text = RichTextBox2.Text & Space & CutSpacepost(List1.List(i)) & Comma
```

```
  Next i
```

```
  If Combo1.Enabled Then
```

```
    'If Combo1.List(Combo1.ListIndex) <> "None" Then
```

```
    ' RichTextBox2.Text = Combo1.List(Combo1.ListIndex) & NewLine & RichTextBox2.Text
```

```
    'End If
```

```
  If Combo1.List(Combo1.ListIndex) = "ValidTime" Then
```

```
    If Combo2.List(Combo2.ListIndex) = "Current" Then
```

```
    ElseIf Combo2.List(Combo2.ListIndex) = "Sequenced" Then
```

```
      RichTextBox2.Text = RichTextBox2.Text & ", " & cutdotpost(List1.List(0)) & ". " _
```

```
      & "valid_from" & ", " & cutdotpost(List1.List(0)) & ". " & "valid_to"
```

```
    ElseIf Combo2.List(Combo2.ListIndex) = "NonSequenced" Then
```

```
    End If
```

```
  ElseIf Combo1.List(Combo1.ListIndex) = "TransactionTime" Then
```

```
    If Combo2.List(Combo2.ListIndex) = "Current" Then
```

```
    ElseIf Combo2.List(Combo2.ListIndex) = "Sequenced" Then
```

```
      RichTextBox2.Text = RichTextBox2.Text & ", " & cutdotpost(List1.List(0)) & ". " _
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    & "txstart" & ", " & cutdotpost(List1.List(0)) & "." & "txstop"
  ElseIf Combo2.List(Combo2.ListIndex) = "NonSequenced" Then
    End If
  End If
End If

RichTextBox2.Text = RichTextBox2.Text & Newline & _
"FROM" & Space & FromTable() & Newline
If RichTextBox3.Text <> "" Then
RichTextBox2.Text = RichTextBox2.Text & "WHERE " & RichTextBox3.Text

```

```

If Combo1.Enabled Then
If Combo1.List(Combo1.ListIndex) = "ValidTime" Then
  If Combo2.List(Combo2.ListIndex) = "Current" Then
    RichTextBox2.Text = RichTextBox2.Text & "AND (valid_to = #12/31/9456#)"
  ElseIf Combo2.List(Combo2.ListIndex) = "Sequenced" Then
  ElseIf Combo2.List(Combo2.ListIndex) = "NonSequenced" Then
  End If
ElseIf Combo1.List(Combo1.ListIndex) = "TransactionTime" Then
  If Combo2.List(Combo2.ListIndex) = "Current" Then
    RichTextBox2.Text = RichTextBox2.Text & "AND (txstop = #12/31/9456#)"
  ElseIf Combo2.List(Combo2.ListIndex) = "Sequenced" Then
  ElseIf Combo2.List(Combo2.ListIndex) = "NonSequenced" Then
  End If
End If
End If
Else:

```

```

If Combo1.Enabled Then
If Combo1.List(Combo1.ListIndex) = "ValidTime" Then
  If Combo2.List(Combo2.ListIndex) = "Current" Then
    RichTextBox2.Text = RichTextBox2.Text & "WHERE " & _
    "(valid_to = #12/31/9456#)"
  ElseIf Combo2.List(Combo2.ListIndex) = "Sequenced" Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ElseIf Combo2.List(Combo2.ListIndex) = "NonSequenced" Then
End If
ElseIf Combo1.List(Combo1.ListIndex) = "TransactionTime" Then
If Combo2.List(Combo2.ListIndex) = "Current" Then
RichTextBox2.Text = RichTextBox2.Text & "WHERE " & _
"(txstop = #12/31/9456#)"
ElseIf Combo2.List(Combo2.ListIndex) = "Sequenced" Then
ElseIf Combo2.List(Combo2.ListIndex) = "NonSequenced" Then
End If
End If
End If
End If
Xstr = RichTextBox2.Text
RichTextBox2.Text = add_order_by(Xstr)

TabStrip1.Tabs(3).Selected = True
Frame1(2).ZOrder 0
End Sub

Private Sub CancelCmd_Click()
Wizard.Hide
End Sub

Private Sub Check1_Click()
AddTree1
AddTree2
..

If Check1.Value = 1 Then
Combo1.Enabled = True
Combo2.Enabled = True
Else
Combo1.Enabled = False
Combo2.Enabled = False

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End If
End Sub

Private Sub Combo5_Click()
Dim start_pos, end_pos As Integer
If List1.List(List1.ListIndex) <> "" Then
Temp = List1.List(List1.ListIndex)
If (Temp.Find("Unsorted") = -1) And (Temp.Find("Ascending") = -1) And _
(Temp.Find("Descending") = -1) Then
If Combo5.List(Combo5.ListIndex) <> "Unsorted" Then
List1.List(List1.ListIndex) = List1.List(List1.ListIndex) & " " & Combo5.List(Combo5.ListIndex)
End If
Else:
If Combo5.List(Combo5.ListIndex) <> "Unsorted" Then
List1.List(List1.ListIndex) = CutSpacepost(List1.List(List1.ListIndex)) & " " & Combo5.List
(Combo5.ListIndex)
Else:
List1.List(List1.ListIndex) = CutSpacepost(List1.List(List1.ListIndex))
End If
End If
End If
End Sub

```

```
Private Sub DBGrid1_Click()
```

```
End Sub
```

```
Private Sub Deletecon_Click()
```

```
Addcondition ("OR")
```

```
End Sub
```

```
Private Sub Form_Activate()
```

```
On Error GoTo Err_Handler
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If IsAddTree <> False Then
```

```
Check1.Value = 1
```

```
'AddTree1
```

```
'AddTree2
```

```
End If
```

```
IsAddTree = False
```

```
Err_Handler:
```

```
End Sub
```

```
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
```

```
KeyIn = Shift
```

```
End Sub
```

```
Private Sub Form_KeyUp(KeyCode As Integer, Shift As Integer)
```

```
KeyIn = 0
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Dim i As Integer
```

```
Newline = Chr(10)
```

```
help1 = " -HighLight the Treeview and Click Add Button to Display the Column" & Newline _
```

```
& " -Select the Listbox and Choose Sorting method" & Newline _
```

```
& " -Enable Temporal method and Choose Time Variance to "
```

```
help2 = " -HighLight the Treeview ,Select condition and match case then Click AND or OR to add  
condition to textbox"
```

```
help3 = " -"
```

```
For i = 0 To Frame1.Count - 1
```

```
With Frame1(i)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

.Move TabStrip1.ClientLeft, _
TabStrip1.ClientTop, _
TabStrip1.ClientWidth, _
TabStrip1.ClientHeight
End With
Next i
Frame1(0).ZOrder 0

Data1.Connect = "ODBC;DSN=student_data;" & _
"UID=" & LoginForm.User.Text & ";PWD=" & LoginForm.Passwprd.Text & ";"

Combo1.ListIndex = 0
Combo2.ListIndex = 0
Combo3.ListIndex = 0
Combo4.ListIndex = 0
Combo5.ListIndex = 0

If TabStrip1.SelectedItem.Index = 1 Then
    RichTextBox1.Text = help1
End If
keylist = 0
IsAddTree = True

End Sub

Private Sub RemAll_Click()
    List1.Clear
End Sub

Private Sub Remove_Click()
    On Error Resume Next
    List1.RemoveItem (List1.ListIndex)
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub RichTextBox3_DragDrop(Source As Control, X As Single, Y As Single)
    If Source.Name = "TreeView2" Then
        RichTextBox3.Text = RichTextBox3.Text & " " & TreeView2.SelectedItem.Text
        TreeView2.Drag vbCancel
        TreeView2.SelectedItem = TreeView2.SelectedItem.Parent
    End If
End Sub

```

```

Private Sub SendtoEdit_Click()
    EditForm.TabStrip1.Tabs(2).Selected = True
    EditForm.Frame1(1).ZOrder 0

    EditForm.OutputSql.Text = RichTextBox2.Text
    Wizard.Hide
End Sub

```

```

Private Sub TabStrip1_Click()

    If TabStrip1.SelectedItem.Index = 1 Then
        RichTextBox1.Text = help1
    ElseIf TabStrip1.SelectedItem.Index = 2 Then
        RichTextBox1.Text = help2
    ElseIf TabStrip1.SelectedItem.Index = 3 Then
        RichTextBox1.Text = help3
    End If
    Frame1(TabStrip1.SelectedItem.Index - 1).ZOrder 0
End Sub

```

```

Private Sub TreeView1_Collapse(ByVal Node As MSComctlLib.Node)
    Node.Image = "key" & 1
End Sub

```

```

Private Sub TreeView1_Expand(ByVal Node As MSComctlLib.Node)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Node.Image = "key" & 3
```

```
End Sub
```

```
Private Sub TreeView2_DblClick()
```

```
' RichTextBox3.Text = RichTextBox3.Text & " " & TreeView2.SelectedItem.Text
```

```
End Sub
```

```
Private Sub TreeView2_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
  If KeyIn = 2 Then
```

```
    Temp.Text = TreeView2.SelectedItem.Key
```

```
    If Temp.Find("child $%@#") <> -1 Then
```

```
      TreeView2.Drag vbBeginDrag
```

```
    Else:
```

```
      TreeView2.Drag vbCancel
```

```
  End If
```

```
End If
```

```
End Sub
```

```
Private Sub TreeView2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
  If KeyIn = 2 Then
```

```
    TreeView2.SelectedItem = TreeView2.HitTest(X, Y)
```

```
  End If
```

```
End Sub
```

```
Private Sub TreeView2_NodeClick(ByVal Node As MSCComctlLib.Node)
```

```
' If KeyIn = 2 Then
```

```
' Temp.Text = TreeView2.SelectedItem.Key
```

```
' If Temp.Find("child $%@#") <> -1 Then
```

```
'   TreeView2.Drag vbBeginDrag
```

```
' Else:
```

```
'   TreeView2.Drag vbCancel
```

```
' End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
' End If
```

```
End Sub
```

3. โปรแกรม QueryBuilder

Source Code ในส่วนของ Login.frm

```
Private Sub Command1_Click()
```

```
'On Error Resume Next
```

```
'EditForm.Data1.Connect =
```

```
"ODBC;DSN=student_dsn;DATABASE=database_student;UID=informix;PWD=informix;HOST=islant;SERVER=ol_island;SERVICE=turbo;PROTOCOL=olsoctcp;"
```

```
EditForm.Data1.Connect = "ODBC;DSN=student_data;" & _
```

```
"UID=" & User.Text & ";PWD=" & Passwprd.Text & ";"
```

```
EditForm.Show
```

```
LoginForm.Hide
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
End
```

```
End Sub
```

Source Code ในส่วนของ Qform1.frm

```
Option Explicit
```

```
Private RstStudent_tt As Recordset
```

```
Private RstSubject_tt As Recordset
```

```
Dim FieldStd As String
```

```
Dim FieldSub As String
```

```
Private Sub AddStudent(Column As String)
```

```
Dim i As Integer
```

```
    Select Case Column
```

```
        Case "ÃĖÑĖ'ÑĵĖÖĵĖÖ"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        FieldStd = "stdid"
    Case "□Ó'ÓË'éÒ'×éÍ(ÀÒÉÒä·Ä)"
        FieldStd = "tpren"
    Case "□Ó'ÓË'éÒ'×éÍ(ÀÒÉÒÍÑ§;ÄÉ)"
        FieldStd = "epren"
    Case "ª×éÍ(ÀÒÉÒä·Ä)"
        FieldStd = "tname"
    Case "ª×éÍ(ÀÒÉÒÍÑ§;ÄÉ)"
        FieldStd = "ename"
    Case "ÈÄÑ;ÈÛμÄ"
        FieldStd = "degree"
    Case "ª'Ó'çÍ'Ñ;ÈÖ;ÉÒ"
        FieldStd = "stdtype"
End Select

Set RstStudent_tt = Form1.cnnTemporal.OpenRecordset _
("select distinct(" & FieldStd & ") from student_tt", _
dbOpenSnapshot)', dbRunAsync)

RowStudent.Clear
RstStudent_tt.MoveFirst
Do While Not RstStudent_tt.EOF
    RowStudent.AddItem RstStudent_tt.Fields_
    (FieldStd).Value
    RstStudent_tt.MoveNext
Loop
RstStudent_tt.Close

End Sub

```

```
Private Sub AddSubject(Column As String)
```

```
Dim i As Integer
```

```
Select Case Column
```

```
Case "ÄËÑÈÇÓ"Ò"
```

```
FieldSub = "subid"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Case "×èíÇÓ"Ò(ÀÒÉÒã·Á)"
    FieldSub = "tname"
Case "×èíÇÓ"Ò(ÀÒÉÍÑ§;ÃÉ)"
    FieldSub = "ename"
Case "È'èÇ·Á;Óμ"
    FieldSub = "credit"
Case "·Ó'èÍ§ÇÓ"Ò"
    FieldSub = "type"
End Select

Set RstSubject_tt = Form1.cnnTemporal.OpenRecordset _
("select distinct(" & FieldSub & ") from subject_tt", _
dbOpenSnapshot)', dbRunAsync)

RowSub.Clear
RstSubject_tt.MoveFirst
Do While Not RstSubject_tt.EOF
    RowSub.AddItem RstSubject_tt.Fields _
(FieldSub).Value
    RstSubject_tt.MoveNext
Loop
RstSubject_tt.Close

End Sub

Private Sub ColumnStudent_Click()
    AddStudent (ColumnStudent.Text)
End Sub

Private Sub ColumnSub_Click()
    AddSubject (ColumnSub.Text)
End Sub

Private Sub exitcmd_Click()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
QForm1.Hide
```

```
Form1.Show
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
ColumnStudent.AddItem "ÃĒÑĒ'Ñ;ĒÖ;ĒÖ"
```

```
ColumnStudent.AddItem "Ó'ÓĒ'ĒÖ×ĒÍ(ÀÖĒÖ.Á)"
```

```
ColumnStudent.AddItem "Ó'ÓĒ'ĒÖ×ĒÍ(ÀÖĒÖÍÑ§;ÃĒ)"
```

```
ColumnStudent.AddItem "×ĒÍ(ÀÖĒÖ.Á)"
```

```
ColumnStudent.AddItem "×ĒÍ(ÀÖĒÖÍÑ§;ÃĒ)"
```

```
ColumnStudent.AddItem "ĒÃÑ;ĒÜ"
```

```
ColumnStudent.AddItem "'ĒÍÑ;Ē;ĒÖ;ĒÖ"
```

```
ColumnSub.AddItem "ÃĒÑĒÇÖ"Ö"
```

```
ColumnSub.AddItem "×ĒÍÇÖ"Ö(ÀÖĒÖ.Á)"
```

```
ColumnSub.AddItem "×ĒÍÇÖ"Ö(ÀÖĒÖÍÑ§;ÃĒ)"
```

```
ColumnSub.AddItem "Ē'ĒÇÁ;Ö"
```

```
ColumnSub.AddItem "'ĒÍÇÖ"Ö"
```

```
ColumnStudent.ListIndex = 0
```

```
ColumnSub.ListIndex = 0
```

```
AddStudent (ColumnStudent.Text)
```

```
AddSubject (ColumnSub.Text)
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
Form1.Show
```

```
End Sub
```

```
Private Sub querycmd_Click()
```

```
Dim SelectSql, Temp_Rowsub As String
```

```
Temp_Rowsub = Form1.cut(RowSub.Text)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If FieldSub <> "credit" Then Temp_Rowsub = "" + Temp_Rowsub + ""
SelectSql = "select t1.stdid, t1.tpren, t1.tname, t1.epren, t1.ename, t1.degree, t1.stdtype, t2.subid,
t2.tname, t2.ename, t2.presub1, t2.presub2, t2.credit, t2.type, "
SelectSql = SelectSql + "t3.yr, t3.term, t3.grade, t3.section, t3.valid_from, t3.valid_to from student_tt as
t1, subject_tt as t2, registration_tt as t3"
SelectSql = SelectSql & " where t3.std_int in (select distinct(std_int) From student_tt where "
SelectSql = SelectSql + FieldStd + " = "" + Form1.cut(RowStudent.Text) + "" )"
'SelectSql = SelectSql & " tpren = ""'Ö'"" )"
SelectSql = SelectSql & " and t3.sub_int in (select distinct(sub_int) From subject_tt where "
SelectSql = SelectSql + FieldSub + " = " + Temp_Rowsub + " )"
'SelectSql = SelectSql & " credit = 3 )"
SelectSql = SelectSql & " and (t1.std_int = t3.std_int) and (t2.sub_int = t3.sub_int) and (t3.valid_from
>= t1.valid_from) and (t3.valid_from < t1.valid_to) and (t3.valid_from >= t2.valid_from) and
(t3.valid_from < t2.valid_to)"

With Output.Data1
.RecordSource = SelectSql
.Refresh
End With

Output.Show
End Sub

```

Source Code ในส่วนของ Qform2.frm

```

Private RstStudent_tt As Recordset
Dim FieldStd As String

Private Sub AddStudent(Column As String)
Dim i As Integer
Select Case Column
Case "ÃËÑË'Ñ;ÉÖ;ÉÖ"
FieldStd = "stdid"
Case "Ó'ÓË'éÖ"×éÍ(ÀÒÉÖä·Á)"
FieldStd = "tpren"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Case "ᐁÓ'ÓÈ'éÒ³×èÍ(ÀÒÉÒÍÑ§;ÄÉ)"
```

```
FieldStd = "epren"
```

```
Case "³×èÍ(ÀÒÉÒä·Ä)"
```

```
FieldStd = "tname"
```

```
Case "³×èÍ(ÀÒÉÒÍÑ§;ÄÉ)"
```

```
FieldStd = "ename"
```

```
Case "ÈÄÑ;ÈÙμÄ"
```

```
FieldStd = "degree"
```

```
Case "³Ó'èÍÑ;ÈÖ;ÉÒ"
```

```
FieldStd = "stdtype"
```

```
End Select
```

```
Set RstStudent_tt = Form1.cnnTemporal.OpenRecordset _
```

```
("select distinct(" & FieldStd & ") from student_tt", _  
dbOpenSnapshot)', dbRunAsync)
```

```
RowStudent.Clear
```

```
RstStudent_tt.MoveFirst
```

```
Do While Not RstStudent_tt.EOF
```

```
RowStudent.AddItem RstStudent_tt.Fields _
```

```
(FieldStd).Value
```

```
RstStudent_tt.MoveNext
```

```
Loop
```

```
RstStudent_tt.Close
```

```
End Sub
```

```
Private Sub ColumnStudent_Click()
```

```
AddStudent (ColumnStudent.Text)
```

```
End Sub
```

```
Private Sub exitcmd_Click()
```

```
QForm2.Hide
```

```
Form1.Show
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

Private Sub Form_Load()

ColumnStudent.AddItem "แ่ÑÑ'Ñ;ÈÖ;ÉÒ"

ColumnStudent.AddItem "Ò'ÓÈ'éÒ×èÍ(ÀÒÉÒä·Ä)"

ColumnStudent.AddItem "Ò'ÓÈ'éÒ×èÍ(ÀÒÉÒÍÑ§;ÄÉ)"

ColumnStudent.AddItem "×èÍ(ÀÒÉÒä·Ä)"

ColumnStudent.AddItem "×èÍ(ÀÒÉÒÍÑ§;ÄÉ)"

ColumnStudent.AddItem "ÈÄÑ;ÈÛμÄ"

ColumnStudent.AddItem "Ò'èÍ'Ñ;ÈÖ;ÉÒ"

ColumnStudent.ListIndex = 0

AddStudent (ColumnStudent.Text)

End Sub

Private Sub Form_Unload(Cancel As Integer)

Form1.Show

End Sub

Private Sub querycmd_Click()

Dim SelectSql As String

SelectSql = "select * from student_tt where std_int in (select distinct(std_int) from student_tt where "

SelectSql = SelectSql + FieldStd + " = "" + RowStudent.Text + "")"

With Output.Data1

.RecordSource = SelectSql

.Refresh

End With

Output.Show

End Sub

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source Code ในส่วนของ Qform3.frm

```
Private RstSubject_tt As Recordset
```

```
Dim FieldSub As String
```

```
Private Sub AddSubject(Column As String)
```

```
Dim i As Integer
```

```
    Select Case Column
```

```
        Case "ÃĒÑĒÇÓ"Ò"
```

```
            FieldSub = "subid"
```

```
        Case "×èÍÇÓ"Ò(ÀÒÉÒä·Á)"
```

```
            FieldSub = "tname"
```

```
        Case "×èÍÇÓ"Ò(ÀÒÉÍÑ§;ĂÉ)"
```

```
            FieldSub = "ename"
```

```
        Case "È'èÇĀ;Ôµ"
```

```
            FieldSub = "credit"
```

```
        Case "×'Ó'çÍ§ÇÓ"Ò"
```

```
            FieldSub = "type"
```

```
    End Select
```

```
    Set RstSubject_tt = Form1.cnnTemporal.OpenRecordset _
```

```
    ("select distinct(" & FieldSub & ") from subject_tt", _
```

```
    dbOpenSnapshot)', dbRunAsync)
```

```
    RowSub.Clear
```

```
    RstSubject_tt.MoveFirst
```

```
    Do While Not RstSubject_tt.EOF
```

```
        RowSub.AddItem RstSubject_tt.Fields _
```

```
        (FieldSub).Value
```

```
        RstSubject_tt.MoveNext
```

```
    Loop
```

```
    RstSubject_tt.Close
```

```
End Sub
```

```
Private Sub ColumnSub_Click()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End With

Output.Show

End Sub

Source Code ในส่วนของ Qform4.frm

Private RstStudent As Recordset

Private RstSubject As Recordset

Dim FieldStd As String

Dim FieldSub As String

Private Sub AddStudent(Column As String)

Dim i As Integer

Select Case Column

Case "ÃĖÑĖ'ÑĵĚÖĵĚÖ"

FieldStd = "stdid"

Case "ġÓ'ÓĖ'ėÓ*ĚĪ(ÀÒÉÒă·Ā)"

FieldStd = "tpren"

Case "ġÓ'ÓĖ'ėÓ*ĚĪ(ÀÒÉÓĪÑĵ;ĀĖ)"

FieldStd = "cpren"

Case "ġ*ĚĪ(ÀÒÉÒă·Ā)"

FieldStd = "tname"

Case "ġ*ĚĪ(ÀÒÉÓĪÑĵ;ĀĖ)"

FieldStd = "cname"

Case "ĖĀÑĵĚÜġĀ"

FieldStd = "degree"

Case "ġ'ĚĪÑĵĚÖĵĚÖ"

FieldStd = "stdtype"

End Select

Set RstStudent = Form1.cnnTemporal.OpenRecordset _

("select distinct(" & FieldStd & ") from student", _

dbOpenSnapshot)', dbRunAsync)

RowStudent.Clear

RstStudent.MoveFirst

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Do While Not RstStudent.EOF
    RowStudent.AddItem RstStudent.Fields _
        (FieldStd).Value
    RstStudent.MoveNext
Loop
RstStudent.Close

```

End Sub

```
Private Sub AddSubject(Column As String)
```

```
Dim i As Integer
```

```
Select Case Column
```

```
Case "ÃÑÊÇÓ"Ò"
```

```
FieldSub = "subid"
```

```
Case "ª×éÍÇÓ"Ò(ÀÒÉÒä·Ä)"
```

```
FieldSub = "tname"
```

```
Case "ª×éÍÇÓ"Ò(ÀÒÉÍÑ§;ÄÉ)"
```

```
FieldSub = "ename"
```

```
Case "È'èÇÄ;Ôµ"
```

```
FieldSub = "credit"
```

```
Case "ª'Ó'éÍÇÓ"Ò"
```

```
FieldSub = "type"
```

```
End Select
```

```
Set RstSubject = Form1.cnnTemporal.OpenRecordset _
```

```
("select distinct(" & FieldSub & ") from subject", _
```

```
dbOpenSnapshot)', dbRunAsync)
```

```
RowSub.Clear
```

```
RstSubject.MoveFirst
```

```
Do While Not RstSubject.EOF
```

```
RowSub.AddItem RstSubject.Fields _
```

```
(FieldSub).Value
```

```
RstSubject.MoveNext
```

```
Loop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
RstSubject.Close
```

```
End Sub
```

```
Private Sub ColumnStudent_Click()
```

```
    AddStudent (ColumnStudent.Text)
```

```
End Sub
```

```
Private Sub ColumnSub_Click()
```

```
    AddSubject (ColumnSub.Text)
```

```
End Sub
```

```
Private Sub exitcmd_Click()
```

```
    QForm1.Hide
```

```
    Form1.Show
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    ColumnStudent.AddItem "ÃĒÑĒ'ÑĵĒÖĵĒÖ"
```

```
    ColumnStudent.AddItem "ĲÓ'ÓĒ'ĒÓ'×ĒÍ(ÀÓĒÒă·Â)"
```

```
    ColumnStudent.AddItem "ĲÓ'ÓĒ'ĒÓ'×ĒÍ(ÀÓĒÒĪÑĵ;ĂĒ)"
```

```
    ColumnStudent.AddItem "×ĒÍ(ÀÓĒÒă·Â)"
```

```
    ColumnStudent.AddItem "×ĒÍ(ÀÓĒÒĪÑĵ;ĂĒ)"
```

```
    ColumnStudent.AddItem "ĒĂÑĵĒÛĒ"
```

```
    ColumnStudent.AddItem "×ĒÓ'ĒĪÑĵĒÖĵĒÖ"
```

```
    ColumnSub.AddItem "ÃĒÑĒĲÓ'Ò"
```

```
    ColumnSub.AddItem "×ĒÍĲÓ'Ò(ÀÓĒÒă·Â)"
```

```
    ColumnSub.AddItem "×ĒÍĲÓ'Ò(ÀÓĒĪÑĵ;ĂĒ)"
```

```
    ColumnSub.AddItem "Ē'ĒĲĂĵĒ"
```

```
    ColumnSub.AddItem "×ĒÓ'ĒĪÑĲÓ'Ò"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ColumnStudent.ListIndex = 0
ColumnSub.ListIndex = 0
AddStudent (ColumnStudent.Text)
AddSubject (ColumnSub.Text)
End Sub

```

```

Private Sub Form_Unload(Cancel As Integer)
    Form1.Show
End Sub

```

```

Private Sub querycmd_Click()
Dim SelectSql, Temp_Rowsub As String
    Temp_Rowsub = Form1.cut(RowSub.Text)
    If FieldSub <> "credit" Then Temp_Rowsub = " " + Temp_Rowsub + " "
SelectSql = "select t1.stdid, t1.tpren, t1.tname, t1.epren, t1.ename, t1.degree, t1.stdtype, t2.subid,
t2.tname, t2.ename, t2.presub1, t2.presub2, t2.credit, t2.type, "
SelectSql = SelectSql + "t3.yr, t3.term, t3.grade, t3.section from student as t1, subject as t2, registration
as t3"
SelectSql = SelectSql & " where ( t1.stdid = t3.stdid ) and ( t2.subid = t3.subid ) and "
SelectSql = SelectSql + "( t1." + FieldStd + " = " + Form1.cut(RowStudent.Text) + " )"
SelectSql = SelectSql + " and ( t2." + FieldSub + " = " + Temp_Rowsub + " )"
Text1 = SelectSql
    With Output.Data1
        .RecordSource = SelectSql
        .Refresh
    End With

```

```

Output.Show
End Sub

```

Source Code ในส่วนของ Qform5.frm

```

Private RstStudent As Recordset
Dim FieldStd As String

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub AddStudent(Column As String)
```

```
Dim i As Integer
```

```
    Select Case Column
```

```
        Case "ÃĚÑĚ'ÑĵĚÖĵĚÖ"
```

```
            FieldStd = "stdid"
```

```
        Case "ġÓ'ÓĚ'ĚÖ*ĚĪ(ÀÒĚÖă-Ā)"
```

```
            FieldStd = "tpren"
```

```
        Case "ġÓ'ÓĚ'ĚÖ*ĚĪ(ÀÒĚÖĪÑĵ;ĂĚ)"
```

```
            FieldStd = "epren"
```

```
        Case "ġ*ĚĪ(ÀÒĚÖă-Ā)"
```

```
            FieldStd = "tname"
```

```
        Case "ġ*ĚĪ(ÀÒĚÖĪÑĵ;ĂĚ)"
```

```
            FieldStd = "ename"
```

```
        Case "ĚĀÑĵĚÛġĀ"
```

```
            FieldStd = "degree"
```

```
        Case "ġĪ'ĚĪÑĵĚÖĵĚÖ"
```

```
            FieldStd = "stdtype"
```

```
    End Select
```

```
    Set RstStudent = Form1.cnnTemporal.OpenRecordset _
```

```
    ("select distinct(" & FieldStd & ") from student", _
```

```
    dbOpenSnapshot)', dbRunAsync)
```

```
    RowStudent.Clear
```

```
    RstStudent.MoveFirst
```

```
    Do While Not RstStudent.EOF
```

```
        RowStudent.AddItem RstStudent.Fields _
```

```
        (FieldStd).Value
```

```
        RstStudent.MoveNext
```

```
    Loop
```

```
    RstStudent.Close
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub ColumnStudent_Click()
    AddStudent (ColumnStudent.Text)
End Sub
```

```
Private Sub exitcmd_Click()
    QForm2.Hide
    Form1.Show
End Sub
```

```
Private Sub Form_Load()
    ColumnStudent.AddItem "ÀÈÑÈ'Ñ;ÈÖ;ÉÒ"
    ColumnStudent.AddItem "Ò'ÓÈ'éÒ*×éÍ(ÀÒÉÒä-À)"
    ColumnStudent.AddItem "Ò'ÓÈ'éÒ*×éÍ(ÀÒÉÒÍÑ;ÄÉ)"
    ColumnStudent.AddItem "×éÍ(ÀÒÉÒä-À)"
    ColumnStudent.AddItem "×éÍ(ÀÒÉÒÍÑ;ÄÉ)"
    ColumnStudent.AddItem "ÈÄÑ;ÈÛμÄ"
    ColumnStudent.AddItem "Ò'çÍÑ;ÈÖ;ÉÒ"

    ColumnStudent.ListIndex = 0
    AddStudent (ColumnStudent.Text)
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
    Form1.Show
End Sub
```

```
Private Sub querycmd_Click()
    Dim SelectSql As String
    SelectSql = "select * from student where "
    SelectSql = SelectSql + FieldStd + " = "" + Form1.cut(RowStudent.Text) + "" "
```

With Output.Data1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

.RecordSource = SelectSql
.Refresh
End With
Output.Show
End Sub

```

Source Code ในส่วนของ Qform6.frm

```

Private RstSubject As Recordset
Dim FieldSub As String

```

```

Private Sub AddSubject(Column As String)

```

```

Dim i As Integer

```

```

Select Case Column

```

```

Case "ÃĒÑĒÇÓ"Ò"

```

```

FieldSub = "subid"

```

```

Case ""×éÍÇÓ"Ò(ÀÒÉÓă-Â)"

```

```

FieldSub = "tname"

```

```

Case ""×éÍÇÓ"Ò(ÀÒÉÍÑ§;ĂÉ)"

```

```

FieldSub = "ename"

```

```

Case "È'èÇĀ;Óμ"

```

```

FieldSub = "credit"

```

```

Case ""Ó'çÍÇÓ"Ò"

```

```

FieldSub = "type"

```

```

End Select

```

```

Set RstSubject = Form1.cnnTemporal.OpenRecordset _

```

```

("select distinct(" & FieldSub & ") from subject", _

```

```

dbOpenSnapshot)', dbRunAsync)

```

```

RowSub.Clear

```

```

RstSubject.MoveFirst

```

```

Do While Not RstSubject.EOF

```

```

RowSub.AddItem RstSubject.Fields _

```

```

(FieldSub).Value

```

```

RstSubject.MoveNext

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Loop
RstSubject.Close

End Sub

Private Sub ColumnSub_Click()
    AddSubject (ColumnSub.Text)
End Sub

Private Sub exitcmd_Click()
    QForm3.Hide
    Form1.Show
End Sub

Private Sub Form_Load()
    ColumnSub.AddItem "ÃĒÑĒÇÔ"
    ColumnSub.AddItem "ª×èÍÇÔ(ÀÓÉÒä·Á)"
    ColumnSub.AddItem "ª×èÍÇÔ(ÀÓÉÍÑ§;ÃÉ)"
    ColumnSub.AddItem "Ë'èÇÁ;Ôµ"
    ColumnSub.AddItem "ª'Ô'çÍÇÔ"

    ColumnSub.ListIndex = 0
    AddSubject (ColumnSub.Text)
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Form1.Show
End Sub

Private Sub querycmd_Click()
Dim SelectSql, Temp_Rowsub As String
    Temp_Rowsub = Form1.cut(RowSub.Text)
    If FieldSub <> "credit" Then Temp_Rowsub = "" + Temp_Rowsub + ""

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SelectSql = "select * from subject where "
SelectSql = SelectSql + FieldSub + " = " + Temp_Rowsub
```

```
With Output.Data1
.RecordSource = SelectSql
.Refresh
End With
```

```
Output.Show
```

```
End Sub
```

Source Code ในส่วนของ QueryBuilder.frm

```
Public wksODBCDirect As Workspace
Public cnnTemporal As Connection
Public Function cut(Command As String)
Dim start_pos, end_pos As Long
cut = Command
temp.Text = Command
end_pos = temp.Find(" ")
If end_pos < -1 Then
start_pos = 0
temp.SelStart = start_pos
temp.SelLength = end_pos - start_pos
cut = temp.SelText
End If
End Function
```

```
Private Sub disconnect()
cnnTemporal.Close
wksODBCDirect.Close
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Set wksODBCDirect = DBEngine.CreateWorkspace("ODBCDirect", "", "", dbUseODBC)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Set cnnTemporal = wksODBCDirect.OpenConnection("TempOnServer", False, False, _
"ODBC;DSN=student_data;UID=informix;pwd=informix;")
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
    disconnect
```

```
End
```

```
End Sub
```

```
Private Sub mnuAbout_Click()
```

```
    AboutForm.Show
```

```
End Sub
```

```
Private Sub mnuExit_Click()
```

```
    disconnect
```

```
End
```

```
End Sub
```

```
Private Sub q1_Click()
```

```
    Form1.Hide
```

```
    QForm1.Show
```

```
End Sub
```

```
Private Sub q2_Click()
```

```
    Form1.Hide
```

```
    QForm2.Show
```

```
End Sub
```

```
Private Sub q3_Click()
```

```
    Form1.Hide
```

```
    QForm3.Show
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub q4_Click()
```

```
    Form1.Hide
```

```
    QForm4.Show
```

```
End Sub
```

```
Private Sub q5_Click()
```

```
    Form1.Hide
```

```
    QForm5.Show
```

```
End Sub
```

```
Private Sub q6_Click()
```

```
    Form1.Hide
```

```
    QForm6.Show
```

```
End Sub
```

Source Code ในส่วนของ Output.frm

```
Option Explicit
```

```
Private RstStudent As Recordset
```

```
Private Sub Form_Load()
```

```
    Data1.Connect = "ODBC;DSN=student_data;" & _  
    "UID=informix" & ";PWD=informix" & ";"
```

```
End Sub
```

```
Private Sub Form_Resize()
```

```
    OutputGrid.Left = 75
```

```
    OutputGrid.Width = Output.Width - 275
```

```
    OutputGrid.Height = Output.Height - 1500
```

```
    Data1.Top = Output.Height - 1450
```

```
    Data1.Left = 75
```

```
    Data1.Width = Output.Width - 275
```

```
    okcmd.Top = Output.Height - 950
```

```
    printcmd.Top = okcmd.Top
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    okcmd.Left = 1000
    printcmd.Left = Output.Width - 3000
End Sub

Private Sub ok_Click()
    Output.Hide
End Sub

Private Sub print_Click()
'On Error GoTo Err_Enter
Dim ename As String
Dim SelectSql As String
    ename = """"Saran""""
'SelectSql = "select * from student_ft"
'SelectSql = "select * from student_tt where std_int in (select std_serial from student where
student.ename=""Saran"")"

    With Data1
        .RecordSource = SelectSql
        .Refresh
        .Caption = "student"
    End With
Err_Enter:
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- C.S.Jensen And R.T. Snodgrass , Temporal Data Management , Time center cr-17 June , 1997 , www.cs.arizona.edu/people/rts/timercenter/timecenter.html
- R.T. Snodgrass , Managing Temporal Data : A five part series , Time center cr-18 September, 1997 , www.cs.arizona.edu/people/rts/timercenter/timecenter.html
- รศ.ดร. สุชาย ธนเวสเถียร , Fundamental of Visual Basic 6 Structure Programming : Sum Publishing
- ชาริน สัทธรรมชารี , คู่มือการเขียนโปรแกรม Microsoft Visual Basic version 6.0 : บริษัท ซัคเซส มีเดีย จำกัด
- วิสาร กำจรเวทย์ , Fundamental of Visual Basic Database Programming : Sum Publishing



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้