

ภาษาเรียกค้นตามตัวอย่างเชิงเวลาบนระบบฐานข้อมูลเชิงเวลา

TQBE ON TEMPORAL DATABASE



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เลขหน้า.....
เลขทะเบียน..... 46130
วัน, เดือน, ปี 20 ส.ค. 2546

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

๒๓๓๐๙๗๖๕๘

ภาษาเรียกค้นตามตัวอย่างเชิงเวลาบนระบบฐานข้อมูลเชิงเวลา

TQBE ON TEMPORAL DATABASE



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาค้นคว้าตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

ปริญญานิพนธ์ปีการศึกษา 2544

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ภาษาเรียกค้นตามตัวอย่างเชิงเวลาบนระบบฐานข้อมูลเชิงเวลา

TQBE ON TEMPORAL DATABASE

ผู้จัดทำ

1. นายประสิทธิ์ พรหมรี รหัสประจำตัว 42015309

2. นายศิวพงษ์ สีแพน รหัสประจำตัว 42015321



(อาจารย์ บัณฑิต พัสยา)

อาจารย์ที่ปรึกษา

ภาษาเรียกค้นตามตัวอย่างเชิงเวลาบนระบบฐานข้อมูลเชิงเวลา

นายประสิทธิ์ พรหมรี 42015309

นายสิวพงษ์ สีแพน 42015321

อาจารย์ บัณฑิต พัสยา อาจารย์ที่ปรึกษา

ปีการศึกษา 2544

บทคัดย่อ

ระบบการจัดการฐานข้อมูลในปัจจุบันมีความสำคัญมากเพราะกิจกรรมต่างๆ ที่กระทำได้ต้องการเก็บข้อมูลไว้เพื่อเป็นข้อมูลในการอ้างอิงการกระทำ ระบบการจัดการฐานข้อมูลได้ช่วยอำนวยความสะดวกต่อผู้ใช้งานมาทั้งในด้านเก็บข้อมูลและการเรียกค้นข้อมูล

แต่อย่างไรก็ตามในการเรียกค้นข้อมูลยังคงมีปัญหาอยู่เนื่องจากต้องใช้ภาษา SQL ในการเรียกค้นข้อมูล การเรียกค้นข้อมูลต้องเป็นบรรทัดคำสั่ง ทำให้เกิดความยุ่งยาก คือผู้ที่เรียกค้นข้อมูลมาดูได้ต้องมีความรู้เรื่องภาษา SQL ซึ่งยากต่อการเรียนรู้สำหรับผู้ที่ไม่มีความรู้ทางด้านนี้มาก่อน การเรียกค้นข้อมูลโดย QBE จะมีความง่ายในการเรียนรู้ เพราะมีลักษณะเป็นกราฟิก QBE จะเป็นทางเลือกที่ดีอีกทางหนึ่งในการเรียกค้นข้อมูล ในด้านการเก็บข้อมูลก็เป็นอีกเรื่องหนึ่งการเก็บข้อมูลจะเป็นการเก็บข้อมูลล่าสุดเท่านั้นถ้าหากเราต้องการทราบประวัติข้อมูลจะทำได้ ถ้าหากต้องการทราบประวัติข้อมูลจะต้องทำการระบุเวลาเข้ากับข้อมูล

ปริญญานิพนธ์ฉบับนี้ได้เสนอเรื่อง TOBE การเก็บข้อมูลจะเก็บทุกข้อมูลที่เปลี่ยนแปลงทำให้การเรียกค้นประวัติการเปลี่ยนแปลงของข้อมูลได้และการเรียกค้นข้อมูลจะง่ายสำหรับผู้ใช้งานทั่วไป

TQBE ON TEMPORAL DATABASE

Prasit Promree

Siwapong Seepan

Mr. Bandit Passaya Advisor

Abstract

Currently, database management is so many important behavior to keep data to refer perform. Database management help convenience for many querist in side data storage and data query.

However, data query has the SQL language problem to query. Data query must be difficult line command to query. Querist should know SQL language so that hard to learn for querist haven't knowledge. Data query by QBE is easy to learn because it has graphic characteristic. That is a good choice to query data. Data storage is only recently data so it hasn't data history.

This thesis presented about TQBE which data storage will keep all data changing. Data query of data changing history will easy to use for user.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ดี ด้วยคำแนะนำและปรึกษาและตรวจสอบจาก อาจารย์ บัณฑิต พัสยา ซึ่งเป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ ผู้ทำวิทยานิพนธ์รู้สึกซาบซึ้งในความอนุเคราะห์อย่างมาก และขอผู้จัดทำขอบพระคุณเป็นอย่างสูง

ขอบคุณ InterSystem ที่สนับสนุนระบบจัดการฐานข้อมูล Cache และให้การอบรมให้ความรู้ในการทำวิทยานิพนธ์ครั้งนี้

สุดท้ายนี้ขอขอบคุณบิดา มารดา ญาติพี่น้อง อันเป็นที่เคารพยิ่ง ที่คอยเป็นกำลังใจ รวมทั้งเพื่อน ๆ พี่ ๆ น้อง ๆ ในห้อง olala ที่ให้คำแนะนำช่วยเหลือตลอดมา

นายประสิทธิ์ พรหมรี
นายสิวพงษ์ สีแพน



สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูป	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 แนวคิดเกี่ยวกับการทำวิจัย	1
1.4 ขอบเขตของงาน โดยย่อ	2
บทที่ 2 ฐานข้อมูลเชิงเวลา	3
2.1 ประเภทและความหมายของเวลา	3
2.1.1 User-Defined Time	3
2.1.2 Valid Time	3
2.1.3 Transaction Time	3
2.2 ช่วงเวลาและลักษณะเวลา	3
2.3 ฐานข้อมูลเชิงเวลา	4
2.3.1 ฐานข้อมูลอิงประวัติ (Historical Database)	4
2.3.2 ฐานข้อมูลย้อนกลับ(Rollback Database)	5
2.3.3 ฐานข้อมูลแบบสแนปช็อต(Snapshot Database)	5
2.3.4 ฐานข้อมูลแบบอิงเวลา(Bitemporal Database)	5
2.4 การระบุเวลาเข้ากับฐานข้อมูล	7
2.4.1 การระบุเวลาที่กระทำกับทUPLEเปิด(Tuple Time stamping)	8
2.4.2 การระบุเวลาที่กระทำกับแอททริบิวต์(Attribute Time stamping)	9

สารบัญ(ต่อ)

	หน้า
บทที่ 3 ภาษาเรียกค้นตามตัวอย่าง	12
3.1 มาตรฐานของ QBE	12
3.1.1 การ Query พื้นฐาน	13
3.1.2 การใช้ And / Or ในการ Query	13
3.1.3 การจัดการความซ้ำซ้อน(Duplicates)	14
3.1.4 การ Query ใช้หลายตาราง (Join Queries)	15
3.1.5 การใช้ คอลัมน์ที่ไม่มีการระบุชื่อ (Unnamed Column)	16
3.1.6 การใช้ตารางที่เป็นลบ (Negative table)	16
3.1.7 การใช้ ตัวปฏิบัติการรวม (Aggregation)	17
3.1.8 การกำหนดเงื่อนไข (Condition Box)	18
3.1.9 การนำข้อมูลเข้า (Inserting tuples)	18
3.1.10 การลบ และ การแก้ไข (Delete and Update)	19
3.2 ภาษาเรียกค้นตามตัวอย่างใน Microsoft Access	20
3.2.1 การเรียกค้นข้อมูลขั้นพื้นฐาน	20
3.2.2 การเรียกค้นข้อมูลแล้วเรียงลำดับ	21
3.2.3 การใช้เงื่อนไขในการค้นหา	21
3.2.4 การใช้ตัวปฏิบัติการรวม	23
3.2.5 การเรียกค้นข้อมูลหลายตาราง	25
บทที่ 4 ระบบฐานข้อมูล Cache`	27
4.1 สถาปัตยกรรมของ Cache`	27
4.1.1 ส่วนประกอบ ของสถาปัตยกรรม Cache`	27
4.2 การ เข้าถึง ถึง Cache` ในรูปแบบ ต่างๆ	28
4.2.1 การเข้าถึง โดย Cache` Object Script	28
4.2.2 การเข้าถึง โดย Cache` SQL	29

สารบัญ(ต่อ)

	หน้า
บทที่ 5 ภาษาเรียกค้นตามตัวอย่างเชิงเวลา	35
5.1 สถาปัตยกรรม ของ TQBE	35
5.2 ส่วนกำหนดช่วงเวลา (Time Range)	35
5.3 ส่วนกำหนดหน่วยย่อยเวลา (Time Granularity)	36
5.4 การพัฒนาระบบ TQBE	36
5.4.1 เครื่องคอมพิวเตอร์	36
5.4.2 ระบบ TQBE	36
5.4.2.1 ขั้นตอนการทำงานของระบบ TQBE	36
5.4.2.2 รายละเอียดของระบบ TQBE	42
บทที่ 6 สรุปผลการทดลอง	53
6.1 ตัวอย่างการทำงานของระบบ TQBE	53
6.2 สรุปการทำวิจัย	67
6.3 ข้อเสนอแนะในการพัฒนาการวิจัย	67
เอกสารอ้างอิง	68
ภาคผนวก ก การเขียนโปรแกรม Visual Basic ติดต่อกับฐานข้อมูล Cache'	69
ภาคผนวก ข การ Configuration ODBC	77
ภาคผนวก ค ขั้นตอนในการสร้าง Class ใน Cache	80
ภาคผนวก ง ขั้นตอนในการเข้าใช้ Cache' SQL	85
ภาคผนวก จ การติดตั้ง Cache' 4.0	87

สารบัญตาราง

	หน้า
ตารางที่ 6.1 ตาราง Student	56
ตารางที่ 6.2 ตาราง Major	56
ตารางที่ 6.3 ตาราง Course	57
ตารางที่ 6.4 ตาราง Teacher	57
ตารางที่ 6.5 ตาราง RegDate	58



สารบัญรูป

	หน้า
รูปที่ 2.1ฐานข้อมูลอิงประวัติ	5
รูปที่ 2.2 ฐานข้อมูลแบบย้อนกลับ	6
รูปที่ 2.3 ฐานข้อมูลสแนปช็อต	6
รูปที่ 2.4 ฐานข้อมูลแบบอิงเวลา	7
รูปที่ 2.5 การระบุเวลาที่ Tuple ด้วยช่วงเวลา	8
รูปที่ 2.6 การระบุเวลาที่ Tuple ด้วยส่วนประกอบเชิงเวลา	9
รูปที่ 2.7 การระบุเวลาที่ Attribute ด้วยช่วงเวลา	10
รูปที่ 2.8 การระบุเวลาที่ Attribute ด้วยส่วนประกอบเชิงเวลา	11
รูปที่ 3.1 แสดง Example table หรือ Skeleton table	12
รูปที่ 3.2 แสดงการเรียกค้นข้อมูลพื้นฐาน	13
รูปที่ 3.3 โครงสร้างตารางและการเรียกค้นพื้นฐาน	21
รูปที่ 3.4 การเรียกค้นที่มีการเรียงลำดับ	21
รูปที่ 3.5 การใช้เงื่อนไขในการค้นหา	22
รูปที่ 3.6 ผลลัพธ์จากเงื่อนไข ของ President ที่เกิดปี 1865	22
รูปที่ 3.7 การใช้ AND และ OR	22
รูปที่ 3.8 ผลลัพธ์จากการใช้ AND และ OR	23
รูปที่ 3.9 การใช้ตัวปฏิบัติการรวม	24
รูปที่ 3.10 ผลจากการใช้ตัวปฏิบัติการรวม AVG	24
รูปที่ 3.11 การใช้ตัวปฏิบัติการรวม SUM	24
รูปที่ 3.12 ผลลัพธ์จากการใช้ตัวปฏิบัติการรวม SUM	25
รูปที่ 3.13 การใช้ตัวปฏิบัติการรวม COUNT	25
รูปที่ 3.14 ผลลัพธ์จากการใช้ตัวปฏิบัติการรวม COUNT	25
รูปที่ 3.15 การเรียกค้นจากหลายตาราง	26
รูปที่ 3.16 ผลจากการเรียกค้นข้อมูลจากหลายตาราง	26
รูปที่ 4.1 สถาปัตยกรรมของการติดต่อ Cache	27
รูปที่ 4.2 ความแตกต่าง ของ OREFs และ OIDs	29
รูปที่ 4.3 Object และ ตารางที่สร้างจากคลาส	30
รูปที่ 4.4 การสืบทอดและการสร้างตาราง	32

สารบัญรูป(ต่อ)

	หน้า
รูปที่ 4.5 การสร้างตารางจากข้อมูลชนิดต่าง ๆ	33
รูปที่ 5.1 สถาปัตยกรรมพื้นฐาน	35
รูปที่ 5.2 ขั้นตอนการเลือกคลาส	37
รูปที่ 5.3 ขั้นตอนเลือกแอททริบิวต์เพื่อสอบถามข้อมูล	38
รูปที่ 5.4 แสดงขั้นตอนการสร้างเงื่อนไข	39
รูปที่ 5.5 ขั้นตอนการสร้าง Timerange	40
รูปที่ 5.6 ขั้นตอนการสร้างคำสั่ง SQL	41
รูปที่ 5.7 แสดงเริ่มต้นโปรแกรม TQBE	42
รูปที่ 5.8 การเลือก DSN ที่สร้างจาก ODBC	43
รูปที่ 5.9 การเลือกคลาสจะปรากฏคลาสและสมาชิกคลาส	44
รูปที่ 5.10 การสอบถามข้อมูลเชิงเวลา	45
รูปที่ 5.11 หน้าต่าง Condition	46
รูปที่ 5.12 การสร้างเงื่อนไขทางเวลาแบบปัจจุบัน	47
รูปที่ 5.13 การสร้างเงื่อนไขทางเวลาแบบ ประวัติข้อมูล	48
รูปที่ 5.14 การสร้างเงื่อนไขทางเวลาแบบช่วงเวลา	49
รูปที่ 5.15 คำสั่ง SQL ที่ได้ของระบบ TQBE	50
รูปที่ 5.16 ผลจากการสอบถามข้อมูลไม่แสดงเวลา	51
รูปที่ 5.17 ผลการสอบถามข้อมูลที่แสดงเวลาด้วย	52
รูปที่ 6.1 สมาชิกของคลาส Student	53
รูปที่ 6.2 สมาชิกของคลาส Course	54
รูปที่ 6.3 สมาชิกของคลาส Teacher	54
รูปที่ 6.4 สมาชิกคลาส Major	55
รูปที่ 6.5 สมาชิกคลาส RegDate	55
รูปที่ 6.6 การเลือกสมาชิกข้อมูลในตัวอย่าง 6.1	59
รูปที่ 6.7 การระบุเงื่อนไขในตัวอย่าง 6.1	60
รูปที่ 6.8 การระบุเวลาปัจจุบันในตัวอย่างที่ 6.1	60
รูปที่ 6.9 ผลลัพธ์ของตัวอย่าง 6.1	61
รูปที่ 6.10 ผลลัพธ์ของตัวอย่าง 6.1 เมื่อเลือก Showtime	61

สารบัญรูป(ต่อ)

	หน้า
รูปที่ 6.11 กำหนด Time state เป็น History ของตัวอย่างที่ 6.2	61
รูปที่ 6.12 ผลลัพธ์ของการสอบถามข้อมูลตัวอย่างที่ 6.2 เลือกแสดงเวลา	62
รูปที่ 6.13 การเลือกสมาชิกข้อมูลจากตัวอย่างที่ 6.3	63
รูปที่ 6.14 การกำหนดช่วงเวลาจากตัวอย่างที่ 6.3	63
รูปที่ 6.15 แสดงผลจากตัวอย่างที่ 6.3	63
รูปที่ 6.16 การเลือกสมาชิกข้อมูลในตัวอย่างที่ 6.4	64
รูปที่ 6.17 การสร้างเงื่อนไขจาก Timecondition ในตัวอย่างที่ 6.4	65
รูปที่ 6.18 ผลของการสอบถามในตัวอย่างที่ 6.4	65
รูปที่ 6.19 การเลือกสมาชิกข้อมูลในตัวอย่างที่ 6.5	66
รูปที่ 6.20 ผลการสอบถามข้อมูลในตัวอย่างที่ 6.5	66



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ฐานข้อมูลที่มีใช้กันอยู่เป็นฐานข้อมูลแบบไม่อิงเวลา จะเก็บค่าของข้อมูล ณ. ขณะเวลาใดเวลาหนึ่งเท่านั้น ซึ่งปกตินั้นจะเป็นค่าของข้อมูลล่าสุด การเปลี่ยนแปลงข้อมูลนั้นข้อมูลเก่าจะถูกเขียนทับโดยข้อมูลใหม่ การเรียกค้นข้อมูลว่าถูกต้องหรือไม่เป็นการเรียกค้นข้อมูลที่มีจริงขณะที่เรียกค้นเท่านั้นการเรียกค้นเมื่อต้องการประวัติของข้อมูลจะทำได้ ยกตัวอย่างเช่นในปี 1998 นายประสิทธิ์ เลือกเรียนวิชา Calculus รหัสวิชา 04 แต่แล้วเมื่อปี 1999 วิชารหัส 04 ได้เปลี่ยนชื่อเป็น English ดังนั้นในปี 2000 เมื่อนายประสิทธิ์เรียนจบที่ใบแสดงผลการเรียนจะเป็นวิชา English แทนที่จะเป็น Calculus ถ้าหากเป็นฐานข้อมูลเชิงเวลาซึ่งมีการเก็บสถานะข้อมูลเก่า ใบแสดงผลการเรียนของนายประสิทธิ์จะเป็นวิชา Calculus ซึ่งเป็นข้อมูลที่เป็นจริงในขณะนั้น

อีกทั้งปัจจุบันภาษาเรียกค้นข้อมูล SQL เป็นที่ใช้กันทั่วไปแต่การใช้ SQL จะเป็นลักษณะของการใช้บรรทัดคำสั่งในการเรียกค้นข้อมูลทำให้ผู้ใช้ต้องรู้ภาษา SQL จึงจะสามารถเรียกค้นข้อมูลได้และไม่สามารถรู้โครงสร้างของตาราง สำหรับ QBE นั้นการเรียกค้นข้อมูลจะเป็นการแทนค่าตัวอย่างลงใน โครงสร้างของตารางทำให้ผู้ใช้ไม่จำเป็นต้องรู้ภาษา SQL ก็สามารถเรียกค้นข้อมูลได้การเรียกค้นมองเห็นโครงสร้างของตารางด้วยทำให้ง่ายในการกำหนดการแสดงผล

ในปัจจุบันโปรแกรมประยุกต์จำนวนมากมีความต้องการในการเก็บสถานะเดิมของข้อมูลหรือสถานะของข้อมูลในอนาคต ระบบการจัดการฐานข้อมูลในปัจจุบันยังขาดส่วนที่สนับสนุนทางเวลา โปรแกรมประยุกต์ส่วนใหญ่หากต้องการจัดการข้อมูลที่มีเวลาเกี่ยวข้องกับตัว ต้องพัฒนาโปรแกรมประยุกต์ เป็นพิเศษ ซึ่งทำให้สิ้นเปลืองค่าใช้จ่ายในการพัฒนามาก ตัวอย่างโปรแกรมประยุกต์ที่ต้องการส่วนสนับสนุนทางเวลาเช่น โปรแกรมการจัดการเงินของธนาคาร โปรแกรมจองตั๋วเครื่องบิน โปรแกรมจองห้องพักของโรงแรม และโปรแกรมที่จัดการประวัติคนไข้ ในโรงพยาบาลซึ่งการสร้างฐานข้อมูลเชิงเวลาต้องระบุส่วนของการเก็บเวลาที่เกี่ยวข้องกับข้อมูลเพิ่มเข้ามา

1.2 วัตถุประสงค์

1. เพื่อเสนอแนวทางในการจัดการฐานข้อมูลเชิงเวลา
2. เพื่อศึกษาการทำงานของ QBE
3. เพื่อสร้าง TQBE ที่สามารถตอบคำถามเชิงเวลาได้

1.3 แนวคิดเกี่ยวกับการทำวิจัย

ภาษาเรียกค้นตามตัวอย่างเชิงเวลาบนฐานข้อมูลเชิงเวลาแนวคิดที่เข้ามาร่วมกันคือ การใช้ภาษาเรียกค้นตามตัวอย่างมาใช้ในการเรียกค้นข้อมูลร่วมกับฐานข้อมูลเชิงเวลา โดยจะมีการสร้างฟอร์มของ TQBE เพื่อเรียกค้นข้อมูลให้เชื่อมโยงเข้ากับส่วนของการเรียกค้นข้อมูล โดยการระบุส่วนเรียกค้นจะระบุไว้ที่กฎความถูกต้อง

ต้องของข้อมูลโดยเขียนโปรแกรมย่อยไว้กับกฎความถูกต้องของข้อมูล ฐานข้อมูลที่จะนำมาใช้ในการเรียกค้นนั้นจะเป็นฐานข้อมูลที่มีระยะเวลาที่กระทำกับ Attribute แบบช่วงเวลาคือหนึ่ง Attribute จะประกอบด้วย Set ของข้อมูลที่ประกอบด้วย ข้อมูล เวลาเริ่มต้น และเวลาสิ้นสุด

1.4 ขอบเขตของงานโดยย่อ

สำหรับขอบเขตของโครงการได้กำหนดขอบเขตคือสามารถเรียกค้นข้อมูลได้ทั้งที่ไม่เกี่ยวข้องกับเวลา และข้อมูลที่มีช่วงเวลาเข้ามาเกี่ยวข้องการจัดการฐานข้อมูลจะระบุเวลาที่ทำกับ Attribute เฉพาะ Valid Time เท่านั้น ซึ่งทราบเฉพาะประวัติของข้อมูลเท่านั้น



บทที่ 2

ฐานข้อมูลเชิงเวลา

ฐานข้อมูลเชิงเวลาได้ถูกคิดค้นขึ้นเพื่อปรับปรุงระบบฐานข้อมูลแบบเดิมซึ่งฐานข้อมูลแบบเดิมไม่มีเวลา มาเกี่ยวข้องจะเกิดปัญหาขึ้นคือ

1. หากมีการเปลี่ยนแปลงข้อมูลกับฐานข้อมูลแล้วข้อมูลใหม่จะเขียนทับข้อมูลเดิม
2. ไม่สามารถค้นหาประวัติของข้อมูลที่ต้องการได้

2.1 ประเภทและความหมายของเวลา

ในฐานข้อมูลเชิงเวลานั้นจะมีการระบุเวลาเข้ากับฐานข้อมูลซึ่งประเภทของเวลาที่ระบุในฐานข้อมูลมีอยู่ 3 ชนิด [1] ที่ระบุเข้าไปคือ

2.1.1 User-Defined Time

เป็นเวลาที่ผู้ใช้จะเป็นคนตีความหมายเอง ระบบจะไม่ทำการแปลความหมายแต่ว่าระบบจะเห็นเป็นเพียง Attribute หนึ่งเท่านั้นเช่น Attribute วันเกิด วันแต่งงาน ซึ่งในปัจจุบันนี้ฐานข้อมูลทั่วไปก็สนับสนุน User-defined time อยู่แล้ว ซึ่งผู้ใช้สามารถเข้าไปกำหนดค่าเปลี่ยนแปลงแก้ไขข้อมูลได้

2.1.2 Valid Time

เป็นเวลาที่แสดงว่าเมื่อไหร่ที่ Fact เป็นจริง หรือแสดงว่าเมื่อไหร่ที่ข้อมูลเป็นจริงในฐานข้อมูล เช่น ระยะเวลาที่ นายประสิทธิ์ เป็นนักศึกษาของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ถึงแม้ว่านายประสิทธิ์จะจบไปแล้วฐานข้อมูลยังจะแสดงว่าข้อมูลเป็นจริงในขณะที่นายประสิทธิ์กำลังเรียนอยู่หรือการจอตัว ภาพยนตร์ล่วงหน้าซึ่งการบันทึกจะบันทึกว่าช่วงใดที่ข้อมูลเป็นจริง

2.1.3 Transaction Time

เป็นเวลาที่บันทึกเมื่อมีการเปลี่ยนแปลงข้อมูลจริง หรือเมื่อ Fact ถูกใส่เข้าไปในฐานข้อมูล ซึ่ง ทรานแซกชันไทม์ จะเป็นตัวคอยเก็บเวลาในการเปลี่ยนแปลงข้อมูล Transaction Time ระบบจะเป็นตัว Generate ให้เองเช่น ทำการ Insert Update ข้อมูลเมื่อใด เช่น เมื่อเราเปลี่ยนชื่อ Valid time จะถูกบันทึกเป็นประวัติของข้อมูล ส่วน Transaction Time จะตั้งค่าเวลาปัจจุบันเป็นสิ้นสุดข้อมูลเป็นจริงเก่า และจะเริ่มคั้นของข้อมูลเป็นจริงใหม่

2.2 ช่วงเวลาและลักษณะเวลา

ช่วงเวลา [1] (Time Interval) คือเวลาระหว่าง 2 เหตุการณ์ ช่วงเวลาอาจจะแสดงโดย set ของความต่อเนื่องของจุดเวลา

ส่วนประกอบเชิงเวลา[1](Time Element) คือ การ Union ไม่จำกัดของเวลาส่วนประกอบเชิงเวลาจะสนับสนุนการ Intersection, Union และการ Complementation ส่วนประกอบเชิงเวลาจะใช้ในการระบุเวลาเข้ากับข้อมูล

เวลามีลักษณะโมเดลพื้นฐานอยู่ 3 แบบ ในฐานข้อมูลเชิงเวลาคือ แบบต่อเนื่อง(Continuous) แบบพิกิด(Dense) แบบไม่ต่อเนื่อง (Discrete)

- แบบต่อเนื่อง คือบนช่วงเวลา 2 จุด จะประกอบด้วยจุดเวลาจำนวนมากกว่า 2 จุดนี้ ซึ่งเวลาที่ระบุนี้จะ เป็นจำนวนจริงในการระบุเวลา
- แบบพิกิด จะระบุเวลาด้วยพิกิดเวลาเป็นเศษส่วน
- แบบไม่ต่อเนื่อง ระหว่างจุด 2 จุดของช่วงเวลาจะมีจุดเวลาต่างๆ เป็นจุดจำนวนเต็ม

ในการระบุเวลาในฐานข้อมูลจะระบุเวลาเป็นช่วง (Time interval) ซึ่งจะมีจุดเริ่มต้น และ จุดสิ้นสุดของ ช่วงเวลาหรือขอบเขตเวลาซึ่งจะแสดงช่วงเวลาโดยใช้ S เป็นจุดเริ่มต้น E เป็นจุดสิ้นสุด และใช้ตัวปฏิบัติการเปรียบเทียบ < และ ≤ ในการแสดงช่วงเวลาเริ่มต้นและสิ้นสุดโดย t จะแทน เวลาในช่วงเวลานั้น

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

ช่วงเวลาเริ่มต้นที่ t มากกว่าหรือเท่ากับ S และสิ้นสุดที่ t น้อยกว่าหรือเท่ากับ E

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

ช่วงเวลาเริ่มต้นที่ t มากกว่า S และสิ้นสุดที่ t น้อยกว่าหรือเท่ากับ E

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

ช่วงเวลาเริ่มต้นที่ t มากกว่าหรือเท่ากับ S และสิ้นสุดที่ t น้อยกว่า E

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

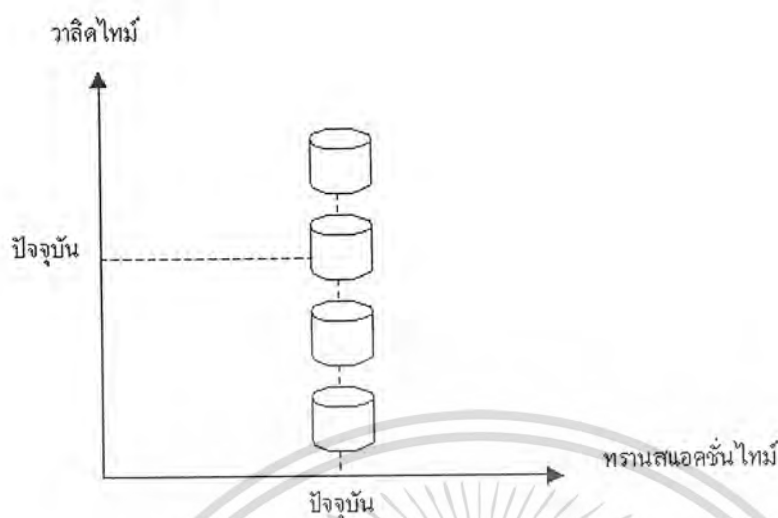
ช่วงเวลาเริ่มต้นที่ t มากกว่า S และสิ้นสุดที่ t น้อยกว่า E

2.3 ฐานข้อมูลเชิงเวลา

ฐานข้อมูลเชิงเวลา [1] (Temporal Database) คือ คือฐานข้อมูลที่สนับสนุนเวลาแบบต่างๆ แต่ไม่รวมถึง User-Define Time ฐานข้อมูล Valid Time และ Transaction Time ทำให้เกิดฐานข้อมูลเชิงเวลาซึ่งเรียกตามชนิดการ เก็บข้อมูลดังนี้คือ

2.3.1 ฐานข้อมูลอิงประวัติ(Historical Database)

เป็นการเก็บประวัติของข้อมูลไว้ ในการเก็บจะเป็นการเก็บเฉพาะ Valid Time ไว้เท่านั้นเอง ซึ่งสามารถ เปลี่ยนแปลงข้อมูลได้ดังรูป 3.1



รูปที่ 2.1 ฐานข้อมูลอิงประวัติ

2.3.2 ฐานข้อมูลย้อนกลับ(Rollback Database)

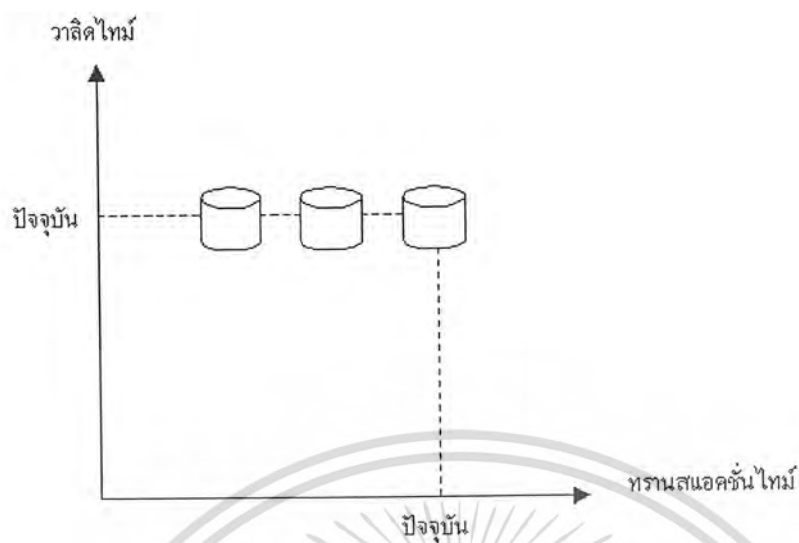
จะเก็บเฉพาะ Transaction Time ในการเก็บจะไม่เก็บประวัติของข้อมูลโดยตรงแต่จะเก็บบันทึกเฉพาะเวลาของการเปลี่ยนแปลงข้อมูล Tuple จะไม่เปลี่ยนสถานะแต่จะมีข้อมูลใหม่ซ้ำสถานะจะไม่ สามารถ เปลี่ยนแปลงข้อมูลได้แต่จะเพิ่มข้อมูลได้ หรือจะเป็นการเก็บพฤติกรรมข้อมูล ดังรูป 2.2

2.3.3 ฐานข้อมูลแบบสแนปช็อต(Snapshot Database)

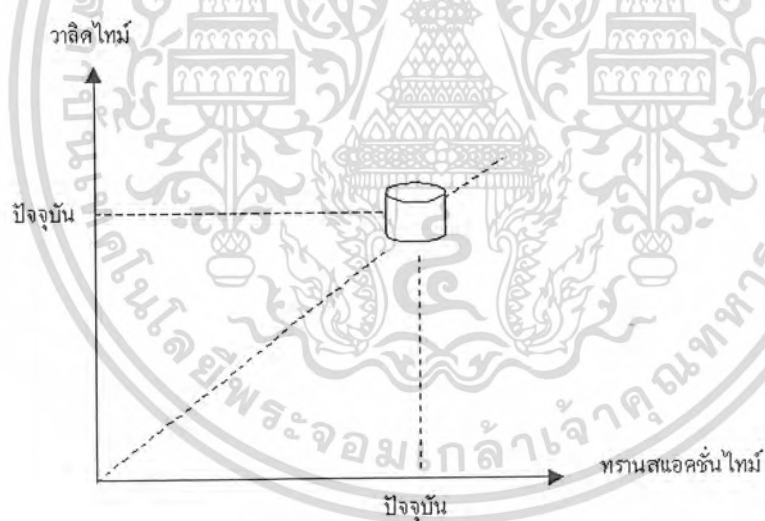
จะเป็นการเก็บทั้ง Valid Time และ Transaction time ในการเก็บจะเป็นการเก็บข้อมูลปัจจุบันซึ่งการแก้ไขข้อมูลจะมีผลหลัง commit แล้วหลังจากเสร็จสิ้นแล้วจะเก็บฐานข้อมูลใหม่และไม่สามารถรู้สถานะข้อมูลเดิม ดังรูป 2.3

2.3.4 ฐานข้อมูลแบบอิงเวลา(Bitemporal Database)

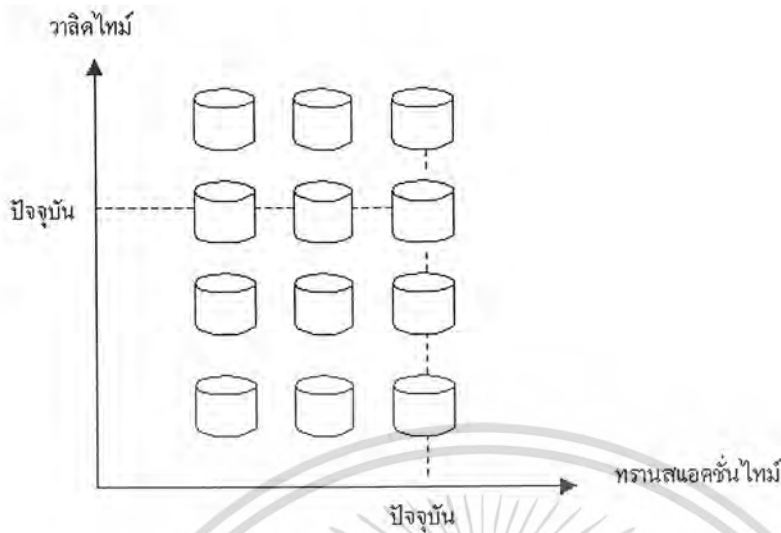
จะทำการเก็บทั้ง Transaction Time และ Valid Time ซึ่งจะทำให้ได้ทั้งพฤติกรรมข้อมูลและประวัติของข้อมูลซึ่งจะเป็นฐานข้อมูลที่ชัดเจนที่สุดในฐานข้อมูลเชิงเวลา ดังรูป 2.4



รูปที่ 2.2ฐานข้อมูลแบบย้อนกลับ



รูปที่ 2.3 ฐานข้อมูลสแนปช็อต



รูปที่ 2.4 ฐานข้อมูลแบบอิงเวลา

2.4 การระบุเวลาเข้ากับฐานข้อมูล

การระบุเวลากับข้อมูล[1] คือการระบุเวลาให้สัมพันธ์กับค่าของข้อมูลเพื่ออ้างเวลาในการเข้าหาวัตถุนั้นซึ่งเวลาที่ระบุจะเป็น ช่วงเวลา จุดเวลา หรือส่วนประกอบเชิงเวลา

ในโมเดลของฐานข้อมูลเชิงเวลา Facts จะถูกแทนด้วยข้อมูลที่มีการระบุเวลาเพื่อเป็นการแสดงว่าระหว่างคาบเวลาใดที่ข้อมูลเป็นจริงและหรือเก็บอยู่ในฐานข้อมูล ซึ่งหน่วยย่อยข้อมูลจะเป็นข้อมูลแบบค่าเดียวหรือเป็นกลุ่มของข้อมูลหรือเป็น set ของกลุ่มข้อมูลในฐานข้อมูล การระบุเวลากับข้อมูลต้องกำหนดว่าจะระบุเวลาเข้าที่ส่วนไหนของข้อมูลและชนิดของเวลาที่ระบุนั้นเป็นแบบใด

ระบบจัดการฐานข้อมูลเชิงเวลาจะเก็บ facts ซึ่งระบุคาบเวลาที่แตกต่างกัน การระบุเวลาเข้ากับฐานข้อมูลนั้นขึ้นอยู่กับว่าเราจะระบุเวลาเข้ากับส่วนไหนของข้อมูลซึ่งจะเป็นการระบุที่ Tuple หรือ ที่กลุ่ม Attribute (set of attribute) ของข้อมูล การระบุเวลาที่ Tuple จะสนับสนุนฐานข้อมูลประเภท Relational และการระบุที่ กลุ่ม Attribute จะสนับสนุนจากฐานข้อมูลประเภท Object-Oriented

การระบุเวลาจะประกอบด้วย การระบุเวลาด้วยจุดเวลา(Time stamping with Time instant) เป็นการระบุว่าข้อมูลเป็นจริงที่เวลา ณ จุดใด การระบุเวลาด้วยช่วงเวลา(Time stamping with Time Interval) เป็นการระบุว่าข้อมูลเป็นจริงในช่วงเวลาใดและการระบุเวลาด้วยส่วนประกอบเชิงเวลา(Time stamping with Time Element) เป็นการระบุว่าข้อมูลเป็นจริงในช่วงเวลาที่คาบเวลาไม่เกี่ยวข้องกัน

2.4.1 การระบุเวลาที่กระทำกับทับเปิล(Tuple Time stamping)

การระบุเวลาที่กระทำกับทับเปิลนี้ปกติจะใช้กับฐานข้อมูลเชิงสัมพันธ์ ซึ่งการระบุเวลาแบบนี้จะสนับสนุนความสัมพันธ์แบบ INF การระบุเวลาจะระบุเวลาเข้าไปในทุกๆทับเปิล ในแต่ละทับเปิลจะถูกระบุด้วยคาบเวลา Valid Time ของข้อมูล หรือ Transaction Time ของข้อมูล หรือทั้ง transaction time และ Valid Time ของข้อมูล

ข้อเสียของวิธีนี้คือจะเกิดความซ้ำซ้อนของข้อมูลใน Attribute ที่ไม่เกี่ยวกับเวลาทำให้ตารางมีขนาดใหญ่ขึ้นหรือจำนวนทับเปิลมากขึ้นนั่นเอง

2.4.1.1 วิธีการระบุเวลาเข้ากับทับเปิล

2.4.1.1.1 การระบุเวลาด้วยช่วงเวลา(Time stamping with Time Interval)

จะเป็นการระบุว่าข้อมูลเป็นจริง ณ ช่วงเวลาใดโดยจะมีการระบุจุดเริ่มต้นและจุดสิ้นสุดของข้อมูลที่เป็นจริง ซึ่งแอตทริบิวต์ที่เพิ่มเข้ามาในทับเปิลจะมี 2 ค่าคือ StartDate และ EndDate ซึ่งจะเป็นจุดเริ่มต้นและจุดสิ้นสุดของข้อมูลที่เป็นจริงดังที่กล่าวมาแล้วว่าการระบุเวลาเข้าเวลาที่ทับเปิลจะทำให้เกิดความซ้ำซ้อนขึ้นระหว่างทับเปิลทำให้จำนวนทับเปิลเพิ่มขึ้นหากมีการแก้ไขข้อมูลใหม่ เช่น

SUBID	SUBNAME	TEACHER	CREDIT	SDATE	EDATE
100	Math	Surachai	3	1994	1996
100	Calculus	Surachai	3	1996	9999
101	Digital	Somchai	3	1994	9999
102	Programing	Somsri	3	1994	9999
103	English	Prateep	2	1994	1996
103	English	Prateep	2	1999	9999

รูปที่ 2.5 การระบุเวลาที่ Tuple ด้วยช่วงเวลา

จากรูปที่ 2.5 จะเห็นว่าวิชา Math เมื่อเปลี่ยนชื่อเป็น Calculus ในปี 1996 จะมี Tuple เพิ่มขึ้นอีก 1 บรรทัด และวิชา English ถูกตัดออกในปี 1996 และถูกเพิ่มเข้ามาใหม่ในปี 1999 ก็มี Tuple เพิ่มขึ้นอีก 1 Tuple เหมือนกัน จะเห็นว่าข้อมูลเกิดความซ้ำซ้อนแต่ข้อดีคือการเรียกค้นข้อมูลที่กระทำง่ายกว่าการระบุเวลา ที่ Attribute

2.4.1.1.2 การระบุเวลาด้วยส่วนประกอบเชิงเวลา(Time stamping with Temporal Element)

การระบุเวลาด้วยส่วนประกอบเชิงเวลาข้อมูลหรือ Fact ที่เป็นจริงในฐานข้อมูลจะถูกระบุด้วยส่วนประกอบเชิงเวลาซึ่งมีคุณสมบัติปิดภายใต้ทฤษฎีของเซตของการ Intersect Difference และ Union

การระบุเวลาแบบนี้ Attribute ที่เพิ่มเข้ามาจะเป็นเซตดังนั้นสามารถเพิ่ม Attribute เดียวในการระบุเวลาและระบุเป็นเซตของข้อมูลในการเก็บ StartDate และ EndDate ไว้ในแอตทริบิวต์เดียวกันได้ถ้าหากเกิดการเปลี่ยนแปลงแอตทริบิวต์ที่ไม่เกี่ยวกับเวลาจะทำให้เกิดความซ้ำซ้อนกับข้อมูลหรือจำนวนทับเปิลมากขึ้นนั่นเอง แต่ถ้าหากเกิดการเปลี่ยนแปลงเฉพาะช่วงเวลาจะไม่เกิดการซ้ำซ้อนของข้อมูลทับเปิลจะไม่เพิ่มขึ้น เช่น

SUBID	SUBNAME	TEACHER	CREDIT	VALID
100	Math	Surachai	3	{<1994,1996>}
100	Calculus	Surachai	3	{<1996,9999>}
101	Digital	Somchai	3	{<1994,9999>}
102	Programing	Somsri	3	{<1994,9999>}
103	English	Prateep	2	{<1994,1996>,<1999,9999>}

รูปที่ 2.6 การระบุเวลาที่ Tuple ด้วยส่วนประกอบเชิงเวลา

จากรูปที่ 2.6 จะเห็นว่ายังเกิดความซ้ำซ้อนของข้อมูลอยู่หากข้อมูลมีการเปลี่ยนแปลงในส่วนที่ไม่เกี่ยวข้องกับเวลา จากการเปลี่ยนชื่อจาก math เป็น calculus ในปี 1996 ส่วนในวิชา English จะไม่เกิดความซ้ำซ้อน

2.4.2 การระบุเวลาที่กระทำกับแอตทริบิวต์(Attribute Time stamping)

เป็นการระบุเวลาเข้าไปกับทุกๆ แอตทริบิวต์ ซึ่งจะขจัดความซ้ำซ้อนของข้อมูลการเก็บประวัติข้อมูลจะเป็นการเก็บประวัติของแต่ละแอตทริบิวต์ ซึ่งการที่จะสามารถระบุเวลาเข้าไปในแอตทริบิวต์ได้นั้นฐานข้อมูลต้องมีลักษณะเป็น Object Database หรือ NFNF ซึ่งจะสนับสนุนการระบุเวลาแบบนี้

2.4.2.1 วิธีการระบุเวลาที่กระทำกับแอตทริบิวต์

2.4.2.1.1 การระบุเวลาด้วยช่วงเวลา(Time stamping with Time Interval)

การระบุช่วงเวลาในแอตทริบิวต์ในแต่ละแอตทริบิวต์จะถูกระบุเวลา StartDate และ EndDate ซึ่งจะเป็นการเก็บประวัติของแต่ละแอตทริบิวต์ทำให้ลดความซ้ำซ้อนของข้อมูลลงได้ และจำนวนทับเปิลไม่เพิ่มแต่จะเกิดความซ้ำซ้อนขึ้นถ้าหากข้อมูลไม่เปลี่ยนแปลง แต่ช่วงเวลาเปลี่ยนแปลงไม่เกี่ยวข้องกัน เช่น

จากรูปที่ 2.7 จะเห็นว่าส่วนที่เพิ่มเข้ามาคือ OID ซึ่งเป็นตัวที่ใช้อ้างอิง Object เดียวกันเมื่อมีการเปลี่ยนแปลงซึ่งระบบจะเป็นตัวสร้างให้เองจะเห็นว่าในแต่ละ Tuple จะไม่มีการซ้ำซ้อนกันของ Tuple แต่ในแต่ละ Attribute ที่คาบเวลาไม่เกี่ยวข้องต่อเนื่องกันจะต้องเก็บแยกเป็นแต่ละกลุ่มข้อมูล

OID	SUBID	SUBNAME	TEACHER	CREDIT
1	{<100,1994,9999>}	{<Math,1994,1996>, <calculus,1996,9999>}	{<Surachai,1994,9999>}	{<3,1994,9999>}
2	{<101,1994,9999>}	{<Digital,1994,9999>}	{<Somchai,1994,9999>}	{<3,1994,9999>}
3	{<102,1994,9999>}	{<Programing ,1994,9999>}	{<Somsri,1994,9999>}	{<3,1994,9999>}
4	{<103,1994,1996>, <103,1999,9999>}	{<English,1994,1996>, <English,1999,9999>}	{<Prateep,1994,1996>, <Prateep,1999,9999>}	{<2,1994,1996>, <2,1999,9999>}

รูปที่ 2.7 การระบุเวลาที่ Attribute ด้วยช่วงเวลา

2.4.2.1.2 การระบุเวลาด้วยส่วนประกอบเชิงเวลา(Time stamping with Time Element)

การระบุแบบนี้จะกำหนดแต่ละแอตทริบิวต์เป็น Set ข้อมูลและภายใน Set ข้อมูลจะประกอบไปด้วย Set ของส่วนประกอบของเวลา ซึ่งจะช่วยลดความซ้ำซ้อนในกรณีของการระบุด้วยช่วงเวลาแอตทริบิวต์กับการระบุเวลาด้วยส่วนประกอบเชิงเวลาที่ทับซ้อนแต่การเรียกค้นข้อมูลจะซับซ้อนกว่าทั้ง 2 อย่าง เช่น

จากรูปที่ 2.8 เรากำหนดส่วนของเวลาให้เป็น set ของข้อมูลอีกทีหนึ่งจะทำให้กลุ่มของข้อมูลไม่ซ้ำซ้อนหากคาบเวลาไม่ต่อเนื่องกัน

จากการอธิบายข้างต้น TQBE จะใช้ได้กับการระบุเวลาที่กระทำกับ Attribute ของข้อมูลที่เป็นแบบระบุด้วยช่วงเวลาช่วงเวลา (Time stamping with time Interval) ใช้ในการระบุเวลาเข้ากับฐานข้อมูล

OID	SUBID	SUBNAME	TEACHER	CREDIT
1	{<100,{<1994,9999>}>}	{<Math,{<1994,1996>},<Calculus,<1996,9999>}>}	{<Surachai,<1994,9999>}>}	{<3,{<1994,9999>}>}
2	{<101,{<1994,9999>}>}	{<Digital,<1994,9999>}>}	{<Surachai,<1994,9999>}>}	{<3,{<1994,9999>}>}
3	{<102,{<1994,9999>}>}	{<Programing,<1994,9999>}>}	{<Surachai,<1994,9999>}>}	{<3,{<1994,9999>}>}
4	{<103,{<1994,1996>,<1999,9999>}>}	{<English,{<1994,1996>,<1999,9999>}>}	{<Prateep,{<1994,1996>,<1996,9999>}>}	{<2,{<1994,1996>,<1996,9999>}>}

รูปที่ 2.8 การระบุเวลาที่ Attribute ด้วยส่วนประกอบเชิงเวลา



บทที่ 3

ภาษาเรียกค้นตามตัวอย่าง

3.1 มาตรฐานของ QBE

การเรียกค้นโดยใช้ตัวอย่าง[2] (Query-by-Example หรือ QBE) เป็นชื่อ ของทั้ง data-manipulation และ database system QBE เป็นพื้นฐานบน Domain Relation Calculus ระบบฐานข้อมูล QBE ถูกพัฒนาขึ้นโดย บริษัท IBM ที่ T.J. Watson Research Center ในต้นปี ค.ศ.1970 ที่ผ่านมานั้น QBE เป็นภาษาที่ใช้จัดการกับข้อมูล ใช้ใน IBM's Query Management Facility (QMF) ทุกวันนี้ ระบบฐานข้อมูล บางระบบ สำหรับ PC สามารถรองรับการทำงาน ของ ภาษา QBE เราจะพิจารณา ภาษาที่ใช้จัดการกับข้อมูล เพียงอย่างเดียว ซึ่งมันมีคุณสมบัติเด่นๆ อยู่ 2 ลักษณะ

- 1 . QBE มีไวยากรณ์ เป็น 2 มิติ เหมือนการมองเห็นตาราง ซึ่งถ้าเป็นภาษาที่เป็นมิติเดียว จะเป็น บรรทัด คำสั่ง เช่น SQL ภาษาที่เป็น 2 มิติ ต้องการ ประโยคที่เป็น 2 ชุดคำสั่ง
2. การเรียกค้นของ QBE นั้นเป็น จะแสดงโดยใช้ตัวอย่าง “ by example” แทนการให้ รวมคำตอบที่ต้องการ ในโปรแกรมย่อย ผู้ใช้เพียงให้ ตัวอย่าง ของความต้องการ ระบบก็จะทำการคำนวณ จากตัวอย่างที่ให้ ก็จะ ได้คำตอบจากการเรียกค้นนั้นๆ

เราจะทำการเรียกค้นโดยการใส่ค่าตัวแปรลงไปในตารางตัวอย่าง (Example table) หรือ ตารางโครงร่าง (Skeleton table) สำหรับ การเรียกค้นในตารางเดียวนั้น (Single relation query) เรานำทั้ง ค่าคงที่ หรือ ส่วนของตัวอย่าง (Example element) ในคอลัมน์ ของตาราง นั้น ส่วนของตัวอย่าง(Example element) ซึ่งตั้งอยู่เป็น ตัวแปรโดเมน และ เป็นค่าเฉพาะเหมือน ค่าตัวอย่าง (Example value)ซึ่งจะมี เครื่องหมาย _ (underscore) รวมทั้งยังมี ตัวนำหน้าคือ P. (เรียกว่า P dot operator) เป็นตัวแปรที่บอกถึงว่าเราต้องการ พิมพ์ค่าคำตอบของ คอลัมน์นั้นออกมา หรือ แสดงค่าในคอลัมน์ออกมา ซึ่งค่าคงที่นั้นจะต้องสอดคล้องกับคอลัมน์นั้น

โดยที่ ฟิลด์แรกจะเป็นชื่อ ของตาราง (Table Name) ต่อไปก็เป็นชื่อฟิวส์ต่างๆ

Reserves	Sid	Bid	Day

Boats	Bid	Bname	Color

รูปที่ 3.1 แสดง Example table หรือ Skeleton table

Sailors	Sid	Sname	Rating	Age

รูปที่ 3.1 (ต่อ)แสดง Example table หรือ Skeleton table

3.1.1 การ Query พื้นฐาน

ตัวอย่าง 3.1 แสดง รายชื่อ และ อายุ ของตาราง ทหารเรือ (Sailors) ทั้งหมด

Sailors	Sid	Sname	Rating	Age
		P. N		P. A

ตัวอย่าง 3.2 แสดงทุกฟิวส์ในตาราง สำหรับทหารเรือ (Sailors) ที่มี Rating > 8 เรียงลำดับ จากน้อยไปหามาก order by (Rating, Age)

Sailors	Sid	Sname	Rating	Age
P.			AO(1).>8	AO(2).

รูปที่ 3.2 แสดงการเรียกค้นข้อมูลพื้นฐาน

โดยใน QBE สามารถกำหนดการจัดการเรียงลำดับคำตอบซ้อนกันได้ โดยที่แสดงค่าจากน้อยไปหามากของ Rating ก่อนแล้วค่อยแสดง Age จากน้อยไปหามาก จากตัวอย่างที่ 2 เราสามารถเขียน P. เพื่อแสดงของข้อมูลทั้งหมดได้ เขียนในรูป Domain relation calculus ได้ดังนี้ $\{(I,N,R,A) \mid (I,N,R,A) \in \text{Sailors} \wedge T > 8\}$ ซึ่งเขียนในรูป DRC จะไม่มีการเรียงลำดับ

3.1.2 การใช้ And / Or ในการ Query

สำหรับ การ And หรือ Or นั้น หลักการในการพิจารณาคือ ตัวแปรโดเมน (Domain variable) และ แถว(Tuple) คือถ้าหากเราต้องการทำการ Or นั้น 2 แถว จะต้องไม่มีตัวแปรโดเมน ที่ซ้ำกัน ดังตัวอย่าง 3.3 สำหรับการ And ตัวแปรโดเมนซ้ำกัน ในคอลัมน์เดียวกัน โดยใน คอลัมน์ Sid มี 2 แถวที่มีตัวแปรโดเมนที่ซ้ำกัน ดังตัวอย่าง 3.4 ถ้าแถวเดียวกัน ค่า Default จะเป็นการ And กัน ดังตัวอย่าง 3.5

ตัวอย่าง 3.3 แสดงรายชื่อ ของทหารเรือ (Sailors) ที่มีอายุมากกว่า 20 ปี หรือ น้อยกว่า 30 ปี

Sailors	Sid	Sname	Rating	Age
		P.		< 30
		P.		> 20

ตัวอย่าง 3.4 แสดงชื่อของ ทหารเรือ (Sailors) ที่มีอายุมากกว่า 20 ปี และ น้อยกว่า 30 ปี

Sailors	Sid	Sname	Rating	Age
	_Id	P.		< 30
	_Id	P.		> 20

ตัวอย่าง 3.5 แสดงชื่อของทหารเรือ (Sailors) ที่มีอายุน้อยกว่า 30 ปี และ ระดับ 4 ขึ้นไป

Sailors	Sid	Sname	Rating	Age
	_Id	P.	> 4	< 30

3.1.3 การจัดการความซ้ำซ้อน(Duplicates)

3.1.3.1 แสดงแถวเดียว เราไม่สามารถกำจัดความซ้ำซ้อนโดย Default โดยเราสามารถใส่ NQ. (Unique) ในการกำจัดความซ้ำซ้อน โดยจะแสดงรายชื่อที่ ซ้ำกันเพียงแถวเดียว ดังตัวอย่าง 3.6 ตัวอย่าง 3.6 แสดงชื่อ ของทหารเรือ(Sailors)ทุกคน ที่อายุน้อยกว่า 30 ลงมา

Sailors	Sid	Sname	Rating	Age
UNQ.		P.		< 30

3.1.3.2 แสดงหลายแถว กำจัดความซ้ำซ้อนโดย Default ซึ่งเราสามารถหลีกเลี่ยงโดยใช้ ALL. ซึ่งจะแสดงทุก แถวของ ตาราง ดังตัวอย่าง 3.7

ตัวอย่าง 3.7 แสดง ชื่อของทหารเรือ(Sailors) ที่อายุมากกว่า 20 และน้อยกว่า 30 ปี

Sailors	Sid	Sname	Rating	Age
ALL.	_Id	P.		< 30
	_Id	P.		> 20

3.1.4 การ Query ใช้หลายตาราง (Join Queries)

การ Join นั้นเป็นการที่เราต้องการเรียกค้น โดยต้องอาศัยมากกว่า 1 ตารางจึงต้องมีการ Join เกิด ขึ้น ซึ่งชื่อของฟิลด์ ที่ซ้ำกันจะเหมือนกับอีกตารางหนึ่งเมื่อมีการอ้างอิงต้องใช้ ตัวแปร โดเมนในคอลัมน์ที่มีชื่อฟิลด์ที่เหมือนกัน ดังตัวอย่าง 3.8 แสดงการ Join กัน ของ ตาราง Sailors กับ ตาราง Reserves

ตัวอย่าง 3.8 แสดงรายชื่อของ ทหารเรือ(Sailors) คนใดที่ได้ประจำเรือในวันที่ 8/4/01 และมีอายุมากกว่า 25 ปีขึ้นไป

Sailors	Sid	Sname	Rating	Age
	_Id	P._S		> 25

Reserves	Sid	Bid	Day
	_Id		'8/4/01'

#หมายเหตุ * วันที่ , สตรีกับ ที่ว่างหรือ ตัวอักษรพิเศษต้องใส่เครื่องหมายคำพูด (' ')

การเพิ่มเงื่อนไข (Condition) ในการ Join ของหลายตาราง

ตัวอย่าง 3.9 แสดงชื่อ และอายุ ของทหารเรือ(Sailors) ที่ประจำเรือเดียวกับ ทหารเรือที่มี

Sid = 22

Sailors	Sid	Sname	Rating	Age
	_Id	P.		P.

Reserves	Sid	Bid	Day
	22	_B	
	_Id	_B	

ปี

ตัวอย่าง 3.10 แสดงชื่อของเรือที่ถูกจองโดยทหารเรือ(Sailors) คนที่จองในวันที่ 8/4/01 และมีอายุมากกว่า 25

Sailors	Sid	Sname	Rating	Age
	_Id	_S		> 25

Reserves	Sid	Bid	Day
	22	_B	'8/4/01'

Boats	Bid	Bname	Color
	22	'Interlake'	P.

3.1.5 การใช้ คอลัมน์ที่ไม่มีการระบุชื่อ (Unnamed Column)

ในการใช้ คอลัมน์ที่ไม่ได้ระบุชื่อนั้น สำหรับ QBE นั้นยอมให้แสดงผล(P.)ได้มากที่สุด 1 ตาราง ถ้าเราต้องการแสดงผลของ 2 ฟิวส์ หรือ หลายตารางจะต้องระบุในคอลัมน์ที่ไม่ระบุชื่อ ดังตัวอย่าง 3.11 นั้นมีการ Join กันระหว่าง ตาราง Sailors กับ ตาราง Reserves เนื่องจาก QBE ยอมให้แสดงผลได้มากที่สุด 1 ตาราง ดังนั้นเมื่อเราต้องการแสดงผล วันที่ของ ตาราง Reserves ดังนั้นเราจึงนำ คอลัมน์ที่ไม่ระบุชื่อมาใช้ในตาราง Sailors

ตัวอย่าง 3.11 แสดงชื่อของทหารเรือ(Sailors)

Sailors	Sid	Sname	Rating	Age		
	_Id	P.	_R	_A	P._D	P.(_R / _A)

Reserves	Sid	Bid	Day
	_Id		_D

#หมายเหตุ miniQBE ยอมให้มี P. ในหลายตาราง

3.1.6 การใช้ตารางที่เป็นลบ (Negative table)

เราสามารถใส่ สัญลักษณ์ $\neg$ แทน ในคอลัมน์ที่มีความสัมพันธ์ ซึ่งมีความหมายว่าไม่ (NOT) ซึ่งตัวแปรจะปรากฏที่ตารางที่เป็นลบ (Negative table) และจะปรากฏที่ตารางเป็นบวก (Positive table) จากตัวอย่าง 3.12 เป็นการแสดงให้เห็นการใช้ $\neg$ (Not)

ตัวอย่าง 3.12 แสดงรายชื่อ ของทหารเรือ(Sailors) ที่มีอายุไม่ต่ำกว่า 25 ปี

Sailors	Sid	Sname	Rating	Age
		P. S		— 25

ตัวอย่าง 3.13 เป็นการแสดงให้เห็นถึงการ ใช้ — (Not) ที่คอลัมน์เดียว โดยไม่ได้ยู่หน้าตัวแปรโดเมนฯ
และเป็นการ Join กันของ 2 ตารางด้วยระหว่าง ตาราง ทหารเรือ (Sailors)กับตาราง การจองเรือ(Reserves)

ตัวอย่าง 3.13 แสดงรายชื่อของ ทหารเรือ(Sailors) ที่ไม่ได้ประจำเรือ

Sailors	Sid	Sname	Rating	Age
	_Id	P. S		

Reserves	Sid	Bid	Day
—	_Id	_B	

#หมายเหตุminiQBE users สามารถใช้ NOT หรือ ~

3.1.7 การใช้ ตัวปฏิบัติการรวม (Aggregation)

ใน QBE สามารถใช้ ตัวปฏิบัติการรวม (Aggregate Operator) ดังนี้ ค่าเฉลี่ย (AVG) , นับ (COUNT) , ค่าต่ำสุด(MIN), ค่าสูงสุด (MAX), การรวม(SUM) ซึ่งที่กล่าวมาไม่สามารถที่จะกำจัดความซ้ำซ้อน(Duplicates) ได้ แต่จะกำจัดการ ซ้ำซ้อน(Duplicates)ได้ โดยใช้ UNQ (Unique)

ส่วนใน คอลัมน์ ที่มี G. นั้นหมายถึง Group by คือ ทุกแถว(tuple) จะมีค่าเหมือนกันในฟิวส์นี้
ในการตอบคำถามนั้น เมื่อเราใช้ .AO(Ascending Order)คือ การเรียงลำดับจากน้อยไปหามาก และถ้าเป็น .DO(Descending Order) เป็นการเรียงลำดับจากมากไปหาน้อย

ในการใช้ Aggregation นั้นทุกคอลัมน์ที่มี P. จะต้องมี G. หรือตัวปฏิบัติการรวม (Aggregate Operator)

ตัวอย่าง 3.14 แสดงค่าเฉลี่ยของ อายุ และ ชั้น ของทหารเรือ(Sailors) โดยเรียงลำดับจากน้อยไปหามากที่มีชื่อและ ชั้นที่เท่ากัน

Sailors	Sid	Sname	Rating	Age	
	_Id	G.	G.P.AO	_A	P.AVG. _A

3.1.8 การกำหนดเงื่อนไข (Condition Box)

การที่มีการกำหนดเงื่อนไข (Conditions Box) เพื่อให้ง่ายในการกำหนดเงื่อนไขในการ Query อาจจะไม่สะดวกนัก หรือไม่ก็ ไม่สามารถทำได้ ที่จะแสดงเงื่อนไขทั้งหมดบน ตัวแปรโดเมน ในตารางตัวอย่าง (Example table) ดังนั้นจึงกลายเป็นการยากที่ในการแสดงเงื่อนไขต่างๆ ดังนั้น QBE จึงได้มีการรวม คุณสมบัติของ การกำหนดเงื่อนไข(Condition Box) ที่ยอมให้มีการแสดงเงื่อนไขต่างๆบน ตัวแปรโดเมน เหมือนทั่วๆไปใน การกำหนดเงื่อนไข ได้และสามารถที่แสดงเงื่อนไขที่มีการใช้กรุป เหมือนกับ Having ใน SQL ได้ ดูได้จากตัวอย่าง ที่แสดงเงื่อนไข ของ ตัวแปรโดเมน _X โดยเป็นการหาค่าเฉลี่ย ของ อายุของทหารเรือ

ตัวอย่าง 3.15 แสดงชั้น ของทหารเรือที่มีค่าเฉลี่ยอายุมากกว่า 30 ปี

Sailors	Sid	Sname	Rating	Age
			G.P	_A

Condition
AVG. _A > 30

นอกจากนั้นเราสามารถใช่ And และ OR ใน condition box ได้เหมือนกัน ดังดูได้จากตัวอย่าง 3.16

ตัวอย่าง 3.16 แสดงรายชื่อของทหารเรือที่ อายุมากกว่า 20 ปี แต่น้อยกว่า 30 ปี

Sailors	Sid	Sname	Rating	Age
		P.		_A

Condition
$20 < _A$ And $_A < 30$

3.1.9 การนำข้อมูลเข้า (Inserting tuples)

3.1.9.1 การนำเข้า แถวเดียว (single – insert tuple) เป็นการนำข้อมูลเข้า แถวเดียวในตาราง โดยจะมีโอเปอเรเตอร์ I. เป็นตัวบอกถึงการ นำข้อมูลเข้า(Insert)

ตัวอย่าง 3.17 นำข้อมูลเข้า Sid = 74 , Sname = Janice , Rating =7 , Age =14

Sailors	Sid	Sname	Rating	Age
I.	74	Janice	7	14

3.1.9.2 การนำเข้าหลายตาราง (multiple insert tuples) สำหรับการนำเข้าข้อมูลเข้ามากกว่า 1 ตาราง นั้น จะมีเงื่อนไขในการกำหนด (Condition Box)การนำเข้าข้อมูลเข้า(Insert)ดังเห็นได้จากตัวอย่าง 3.18

ตัวอย่าง 3.18 นำ _Id , _N, _A เข้า ตาราง Sailors และ Students โดยมีเงื่อนไขว่า _A มากกว่า 18 ปี หรือ _N มีตัว C ขึ้นหน้าขึ้นก่อน

Sailors	Sid	Sname	Rating	Age
I.	_Id	_N		_A

Students	Sid	Name	Login	Age
	_Id	_N		_A

Conditions
_A > 18 OR _N like ' C% '

3.1.10 การลบ และ การแก้ไข (Delete and Update)

ในการลบ (Delete) ข้อมูลจากตารางนั้นมีหลายแบบเหมือนกับการเรียกค้นข้อมูล การใช้โอเปอเรเตอร์ D. นั้นสามารถใช้ได้กับ ตารางเดียวเท่านั้นส่วนถ้าเราต้องการที่จะลบทิ้งเปิดออกจากหลายตารางนั้นเราจะต้องใช้ โอเปอเรเตอร์ D. สำหรับ แต่ละตาราง ดังดูได้จากตัวอย่างต่างๆ ในการ แก้ไข(Update) เราต้องการที่จะทำการแก้ไขค่าใดค่าหนึ่งในตาราง เราใช้ โอเปอเรเตอร์ U.

ข้อจำกัดของคำสั่งในการ Update

- ไม่สามารถรวม I. , D. และ U. ในตารางตัวอย่างเดียวได้ หรือ รวมทั้งหมดด้วย P. หรือ G.
- เราไม่ยอมให้ผู้แก้ไข คีย์หลัก (Primary key)
- ไม่สามารถ Insert , Update หรือ modify tuples โดยการใช้ จาก field ของ tuples อื่นๆ ในตารางเดียวกัน

ตัวอย่างในการ Delete

ตัวอย่าง 3.19 ลบการจองทั้งหมด ของทหารเรือที่มี ชั้นน้อยกว่า 4

Sailors	Sid	Sname	Rating	Age
	_Id		< 4	

Reserves	Sid	Bid	Day
D.	_Id		

ตัวอย่างในการ Update

ตัวอย่าง 3.20 บวกอายุเข้าไปอีก 1 ของทหารเรือที่ Sid = 74

Sailors	Sid	Sname	Rating	Age
	74			U._A+1

ตัวอย่างต่อไปนี้เป็นการที่หาผิดหลักการของการ Update เราไม่สามารถที่จะทราบว่าจะทำการ Update ตัวไหนแน่ โดยคำถามที่เราจะถาม

เราควรจะทำการ Update อายุของ Joe ทุกครั้งไหม?

อายุของ John อันไหนที่เราควรจะใช้?

Sailors	Sid	Sname	Rating	Age
		John		_A
		Joe		U._A+1

3.2 ภาษาเรียกค้นตามตัวอย่างใน Microsoft Access

การเรียกค้นข้อมูลโดยใช้ตัวอย่าง (QBE) คือวิธีการเลือกรูปแบบของการเรียกค้นโดยจะเลือกตาราง ฟิลด์ ข้อมูลจากรายการเลือกเงื่อนไข และการเรียงลำดับข้อมูลซึ่งจะง่ายในการเรียกค้น [3]

3.2.1 การเรียกค้นข้อมูลขั้นพื้นฐาน

จากรูปที่ 3.21 การเรียกค้นข้อมูลโดยใช้ตัวอย่างใน Ms Access จะมีอยู่ 2 ส่วน คือส่วนที่หนึ่งจะแสดงตารางและฟิลด์ของตารางหรือโครงสร้างของตารางนั่นเอง ส่วนที่ 2 คือส่วนที่ใช้ในการเรียกค้นข้อมูลซึ่งจะเป็นการกำหนดสิ่งที่ต้องการจะรู้ การเรียกค้นข้อมูลโดยใช้ตัวอย่างนี้จะกำหนดง่าย โดย การเลือกกำหนดโครงสร้างที่ต้องการได้เนื่องจากตัวช่วยในการกำหนด

ในการเรียกค้นจากรูปที่ 3.3 จะเป็นการแสดงรายชื่อของประธานาธิบดี วันเกิด และระยะเวลาที่ดำรงตำแหน่งจากตาราง PRESIDENT หากเลือกสตาร์ (*) ที่ Colum ที่ 1 จะเป็นการแสดงทุกฟิลด์ในตารางเหมือนกับ (*) ใน SQL

Query1 : แบบสอบถามแบบใช้เลือกข้อมูล

PRESIDENT

* PRES_NAME
BIRTH_YR
YRS_SERV
DEATH_AGE

เขตข้อมูล:	PRES_NAME	BIRTH_YR	YRS_SERV	
ตาราง:	PRESIDENT	PRESIDENT	PRESIDENT	
เรียงลำดับ:				
แสดง:	☒	☒	☒	☐
เงื่อนไข:				
หรือ:				

รูปที่ 3.3 โครงสร้างตารางและการเรียกค้นพื้นฐาน

3.2.2 การเรียกค้นข้อมูลแล้วเรียงลำดับ

จากรูปที่ 3.4 จะเป็นการแสดงรายชื่อของประธานาธิบดี วันเกิด และระยะเวลาที่ดำรงตำแหน่งจากตาราง PRESIDENT แล้วทำการเรียงลำดับตามวันเกิดจากน้อยไปหามาก (กรณีที่เป็นตัวอักษรจะเรียงลำดับตามตัวอักษร)

Query1 : แบบสอบถามแบบใช้เลือกข้อมูล

PRESIDENT

* PRES_NAME
BIRTH_YR
YRS_SERV
DEATH_AGE

เขตข้อมูล:	PRES_NAME	BIRTH_YR	YRS_SERV	
ตาราง:	PRESIDENT	PRESIDENT	PRESIDENT	
เรียงลำดับ:		Ascending		
แสดง:	☒	☒	☒	☐
เงื่อนไข:				
หรือ:				

รูปที่ 3.4 การเรียกค้นที่มีการเรียงลำดับ

3.2.3 การใช้เงื่อนไขในการค้นหา

การใส่เงื่อนไขในการค้นหาจะใส่ที่ช่องเงื่อนไขโดยในช่องเดียวสามารถใส่เงื่อนไข And Or ในช่องเดียวกันได้ และการใส่เงื่อนไขในการค้นหาในแถวเดียวกันจะเป็นการใส่เงื่อนไขเป็น And หากการใช้เงื่อนไขในการค้นหาคนละบรรทัดจะเป็นการใส่เงื่อนไขการค้นหาเป็น or เช่นจากรูปที่ 3.5 แสดงรายชื่อชื่อประธานาธิบดี ปีเกิด และระยะเวลาดำรงตำแหน่ง ที่เกิดปี ค.ศ. 1865

Query1 : แบบสอบถามแบบใช้เลือกข้อมูล

PRESIDENT			
*			
PRES_NAME			
BIRTH_YR			
YRS_SERV			
DEATH_AGE			

เขตข้อมูล:	PRES_NAME	BIRTH_YR	YRS_SERV	
ตาราง:	PRESIDENT	PRESIDENT	PRESIDENT	
เรียงลำดับ:				
แสดง:	☒	☒	☒	
เงื่อนไข:		"1865"		
หรือ:				

รูปที่ 3.5 การใช้เงื่อนไขในการค้นหา

Query1 : แบบสอบถามแบบใช้เลือกข้อมูล

	PRES_NAME	BIRTH_YR	YRS_SERV
▶	Harding W G	1865	2
*			

รูปที่ 3.6 ผลลัพธ์จากเงื่อนไข ของ President ที่เกิดปี 1865

และอีกตัวอย่างในรูปที่ 3.7 เป็นการเรียกค้นการใช้เงื่อนไข and และ or ในการแสดงชื่อประธานาธิบดี ปีเกิด และระยะเวลาดำรงตำแหน่ง ที่เกิดปี ค.ศ. 1865 และระยะดำรงตำแหน่ง 2 ปี หรือ ประธานาธิบดีที่เกิดในปี ค.ศ. 1913 และผลลัพธ์ของการเรียกค้นข้อมูลจะได้ดังรูปที่ 3.8

Query1 : แบบสอบถามแบบใช้เลือกข้อมูล

PRESIDENT			
*			
PRES_NAME			
BIRTH_YR			
YRS_SERV			
DEATH_AGE			

เขตข้อมูล:	PRES_NAME	BIRTH_YR	YRS_SERV	
ตาราง:	PRESIDENT	PRESIDENT	PRESIDENT	
เรียงลำดับ:				
แสดง:	☒	☒	☒	☐
เงื่อนไข:		"1865"	2	
หรือ:		"1913"		

รูปที่ 3.7 การใช้ AND และ OR

Query1 : แบบสอบถามแบบใช้เลือกข้อมูล

	PRES_NAME	BIRTH_YR	YRS_SERV	
	Harding W G	1865	2	
	Nixon R M	1913	5	
	Ford G R	1913	2	

รูปที่ 3.8 ผลลัพธ์จากการใช้ AND และ OR

ในช่องเงื่อนไข จะสามารถใส่เงื่อนไขในการเรียกค้นข้อมูล ซึ่งจะคล้ายส่วน Where ในภาษาเรียกค้นข้อมูล SQL ตัวอย่างการเปรียบเทียบที่จะใช้ในการเรียกค้น

ตัวเปรียบเทียบ	ตัวอย่าง
Operators	<, >, =, <=, >=, Between
Numbers	YRS_SERV > 2
Text	
Common	Name ="Jones"
Match All	Name Like "J*"
Match One	Name Like "?m*"
Date	Date Between #8/15/99# And #8/31/99#
Missing Data	City is Null
Negation	Name is not Null

3.2.4 การใช้ตัวปฏิบัติการรวม

ในไมโครซอฟต์ Access ได้มี ตัวปฏิบัติการภายในมาให้ที่สำคัญให้มาซึ่งเมื่อเราต้องการใช้ตัวปฏิบัติการเหล่านี้จะเลือกการเรียกค้นแบบ รวม ตัวปฏิบัติการที่สำคัญ เช่น

SUM = ผลรวม

AVG = ค่าเฉลี่ย

MIN = ค่าต่ำสุด

MAX = ค่าสูงสุด

COUNT = นับจำนวน

ซึ่งตัวปฏิบัติการ Sum , Avg, Max , Min จะใช้กระทำกับ Column ส่วน Count จะใช้กับแถวในตาราง ตัวอย่างการใช้ตัวปฏิบัติการรวมเช่น

จากรูปที่ 3.9 การหาค่าเฉลี่ยของการดำรงตำแหน่งของประธานาธิบดีจากตาราง president จะได้ผลลัพธ์ดังรูปที่ 3.10

Query1 : แบบสอบถามแบบใช้เลือกข้อมูล

PRESIDENT

* PRES_NAME
BIRTH_YR
YRS_SERV
DEATH_AGE

เขตข้อมูล: YRS_SERV
ตาราง: PRESIDENT
ผลรวม: Avg
เรียงลำดับ:
แสดง: ☒
เงื่อนไข:
หรือ:

รูปที่ 3.9 การใช้ตัวปฏิบัติการรวม

Query1 : แบบสอบถามแบบใช้เลือกข้อมูล

AvgOfYRS_SERV

4.7435897435897435897435897436

รูปที่ 3.10 ผลจากการใช้ตัวปฏิบัติการรวม AVG

อีกตัวอย่างในรูปที่ 3.11 แสดงจำนวนทั้งหมดที่พรรค Democratic เป็นประธานาธิบดีผลลัพธ์ดังรูปที่ 3.30

Query1 : แบบสอบถามแบบใช้เลือกข้อมูล

PRESIDENT

* PRES_NAME
BIRTH_YR
YRS_SERV
DEATH_AGE

เขตข้อมูล: PARTY YRS_SERV
ตาราง: PRESIDENT PRESIDENT
ผลรวม: Where Sum
เรียงลำดับ:
แสดง: ☐
เงื่อนไข: "Democratic" ☒
หรือ:

รูปที่ 3.11 การใช้ตัวปฏิบัติการรวม SUM

Query1 : แบบสอบถามแบบใช้เลือกข้อมูล

SumOfYRS_SERV
73

รูปที่ 3.12 ผลลัพธ์จากการใช้ตัวปฏิบัติการรวม SUM

ในรูปที่ 3.13 เป็นอีกตัวอย่างเป็นการใช้การรวมกลุ่ม ซึ่งเป็นการแสดงชื่อพรรคและจำนวนครั้งที่แต่ละพรรคได้บริหารประเทศ จะได้ผลลัพธ์ดังรูปที่ 3.14

Query2 : แบบสอบถามแบบใช้เลือกข้อมูล

PRESIDENT

*
PRES_NAME
BIRTH_YR
YRS_SERV
DEATH_AGE

เขตข้อมูล: PARTY PARTY
ตาราง: PRESIDENT PRESIDENT
ผลรวม: Group By Count
เรียงลำดับ:
แสดง: ☒ ☒ ☐ ☐
เงื่อนไข:
หรือ:

รูปที่ 3.13 การใช้ตัวปฏิบัติการรวม COUNT

Query2 : แบบสอบถามแบบใช้เลือกข้อมูล

PARTY	CountOfPARTY
Democratic	13
Demo-Rep	4
Federalist	2
Republican	16
Whig	4

รูปที่ 3.14 ผลลัพธ์จากการใช้ตัวปฏิบัติการรวม COUNT

3.2.5 การเรียกค้นข้อมูลหลายตาราง

การเรียกค้นข้อมูลหลายตารางหรือการ join กันของตารางซึ่ง QBE ของ Ms Access นั้นจะเรียกตารางขึ้นมาและกำหนดความสัมพันธ์ของตารางจากนั้นก็ทำการเรียกค้นตามปกติ

จากตัวอย่างในรูปที่ 3.15 จะเป็นการเรียกค้นเกี่ยวกับการแต่งงานของ President แต่ละท่าน ชื่อภรรยา อายุของ President แต่ละท่านและอายุของภรรยาจากสองตารางคือตาราง President และ ตาราง Pres_marriage จากรูปเส้นที่ขีดต่อกันคือแสดงความสัมพันธ์กันของตาราง จะได้ผลลัพธ์ดังรูปที่ 3.16

The screenshot shows a database query interface. At the top, there are two table definitions: **PRESIDENT** with attributes PRES_NAME, BIRTH_YR, YRS_SERV, and DEATH_AGE; and **PRES_MARRIAGE** with attributes PRES_NAME, SPOUSE_NAME, PR_AGE, and SP_AGE. Below these, a query result table is displayed with the following data:

เขตข้อมูล:	PRES_NAME	SPOUSE_NAME	PR_AGE	SP_AGE
ตาราง:	PRESIDENT	PRES_MARRIAGE	PRES_MARRIAGE	PRES_MARRIAGE
เงื่อนไข:	☒	☒	☒	☒

รูปที่ 3.15 การเรียกค้นจากหลายตาราง

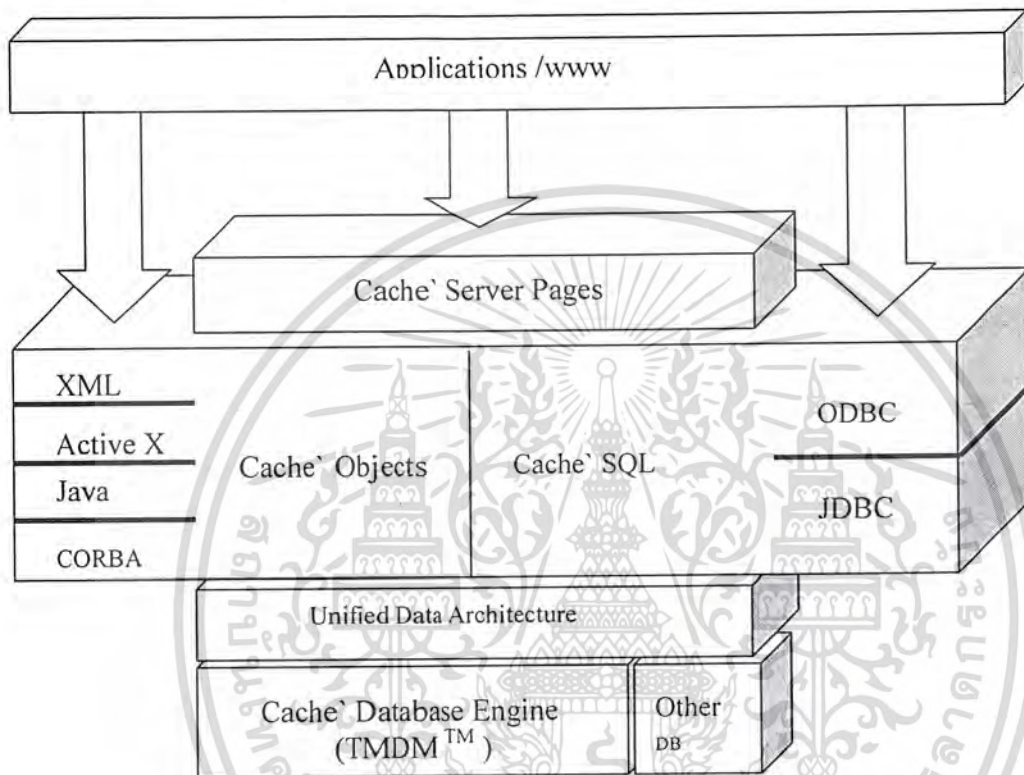
	PRES_NAME	SPOUSE_NAM	PR_AGE	SP_AGE
▶	Adams J	Smith A	28	19
	Adams J Q	Johnson L C	30	22
	Arthur C A	Herndon E L	29	22
	Carter J E	Smith R	21	18
	Cleveland G	Folson F	49	21
	Coolidge C	Goodhue G A	33	26
	Eisenhower D D	Doud G	25	19
	Fillmore M	Powers A	26	27
	Fillmore M	McIntosh C C	58	44
	Ford G R	Warren E B	35	30
	Garfield J A	Rudolph L	26	26
	Grant U S	Dent J B	26	22
	Harding W G	De Wolfe F K	25	30

รูปที่ 3.16 ผลจากการเรียกค้นข้อมูลจากหลายตาราง

บทที่ 4

ระบบฐานข้อมูล Cache'

4.1 สถาปัตยกรรมของ Cache'



รูปที่ 4.1 สถาปัตยกรรมของการติดต่อ Cache'

4.1.1 ส่วนประกอบ ของสถาปัตยกรรม Cache'

1. Application / WWW เป็นส่วน ของ Application ที่ทำงานกับ Cache' รวมถึงการทำงาน ในส่วนของระบบอินเทอร์เน็ตด้วย
2. Cache' Server Pages ในส่วนนี้ Cache' ได้จัดทำภาษาที่เขียน เพื่อการติดต่อกับ Database โดยทำงานบน WWW ไร่ ซึ่ง Brower สามารถรองรับ Cache' Server Pages ซึ่งจะเป็น ไฟล์นามสกุล .CSP โดยทำงานเหมือนกับ .ASP ของ Microsoft และ .JSP ของ Sun ซึ่งแต่ละชนิดจะสนับสนุน ผลิตภัณฑ์ ของตนเอง
3. ส่วนเข้าถึง Cache' ซึ่งได้มีการ เข้าถึงอยู่ 2 รูปแบบ
 - Cache' Objects ซึ่งส่วนนี้จะเป็นการเข้าถึงในส่วนของ Object ซึ่ง ได้จัดทำในส่วนของ Object ไร่ตามคุณสมบัติของ Object-Oriented Programming เช่นการ Inheritance ,Polymorphism

,Persistency โดยทาง Cache' ได้จัดทำ ที่ให้ในการเข้าถึง มีด้วยกันหลายวิธีและ หลาย มาตรฐาน ต่างๆ ที่สามารถทำงานกับ Object ได้เช่น XML , ActiveX , Java , CORBA ซึ่ง สามารถเข้าถึง Object ได้

- Cache' SQL ซึ่งส่วนนี้จะการเป็นเข้าถึงในส่วนของ Relation โดย ในส่วนของ Cache' ได้จัดทำ ในส่วนนี้ไว้ โดยการเข้าถึง Database ได้โดย ผ่าน ทาง ODBC หรือ JDBC ซึ่งขึ้นอยู่กับ Application ที่เราใช้ ถ้าเรา ใช้ Application ที่พัฒนาด้วย Tool ของ Microsoft ก็ใช้ ODBC หรือ ถ้าพัฒนาโดย ใช้ Java ของ Sun ก็ใช้การติดต่อโดยใช้ มาตรฐาน JDBC ซึ่งมีเตรียมพร้อมให้อยู่ แล้ว
- Unified Data Archetecture ซึ่งในส่วนนี้ Cache' ได้จัดทำมาเพื่อให้เป็น สถาปัตยกรรม ที่เป็น กลางเพื่อที่จะรองรับการทำงาน ของการ Objects และ Tables เพื่อให้การทำ งานได้หลาย ทาง โดยจะรองรับในส่วนการแสดงผล ของข้อมูลและ การเชื่อมต่อของทั้ง Object และ Relational Model , UDA(Unified Data Architecture) ได้จัดการเข้าถึงของมูล ใหม่ และ Object ที่มีอยู่แล้ว หรือว่า Application ที่เป็น Relational
- Cache' Database Engine เป็นส่วนของการเก็บของมูล ใน Database ของ Cache' หรือสามารถ เชื่อมต่อกับ ฐานข้อมูล อื่นๆ มากกว่า 1 ระบบ

4.2 การ เข้าถึง ถึง Cache' ในรูปแบบ ต่างๆ

ในส่วนนี้เราจะกล่าว ถึงการเข้าถึง Database รวมๆ ของใน Cache' โดยจะมีการเข้าถึงอยู่ 2 รูปแบบ ดัง ของคนได้กล่าวมา คือ แบบแรก จะเป็นการเข้าถึง โดยใช้ Object และการใช้ SQL ในการเข้าถึง ซึ่งจะมีรายละเอียด ไปดังนี้

4.2.1 การเข้าถึง โดย Cache' Object Script

การอ้างถึง Object

ในการจัดการ Object ของ Cache' ซึ่ง Object จะอยู่ในดิสก์ หรือ ในหน่วยความจำโดยที่Object ที่อยู่ใน ดิสก์จะเป็น Object ที่เก็บในลงใน ฐานข้อมูลส่วน Object ที่อยู่ในหน่วยความจำจะถูกโหลดมาใช้ และมีการเปลี่ยนแปลงอยู่ตลอด เราสามารถอ้างถึง Object เหล่านั้นได้

การอ้างถึงมี 2 รูปแบบ

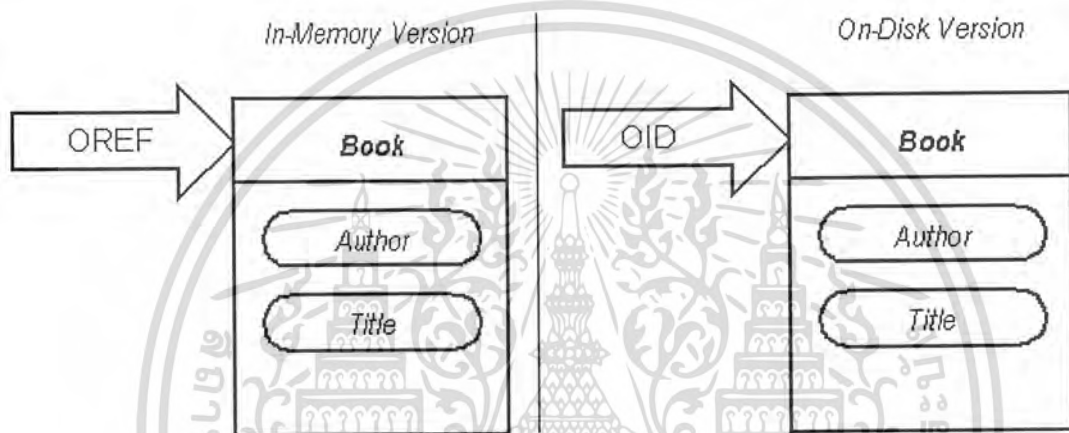
- OREF (Object Reference)

เป็นการอ้างถึง Object ในหน่วยความจำ ในแต่ละครั้งที่ โหลดขึ้นมาที่ หน่วยความจำจะมีค่า OREF ที่ต่าง กัน

- OID (Object Identifier)

เป็นการอ้างอิงถึง Object ที่อยู่ในดิสก์ ซึ่งค่า OID ของ Object จะค่าที่ไม่ซ้ำกัน(Unique)ซึ่งการสร้าง Object เป็นแบบ Persistent ซึ่งจัดเก็บอยู่ในดิสก์ เมื่อเราสร้าง Object จะค่า OID ระบุประจำ Object ในแต่ละ Object โดยค่า OID จะไม่มีเปลี่ยนแปลง

เมื่อเราใช้ OID ของ Object ที่เป็น Persistent โหลดมาไว้ที่หน่วยความจำ จากนั้นระบบจะทำการกำหนด OREF ที่หน่วยความจำเพื่อให้ แอปพลิเคชัน สามารถนำค่า ต่างๆ ของ Object นั้นไปใช้



รูปที่ 4.2 ความแตกต่าง ของ OREFs และ OIDs

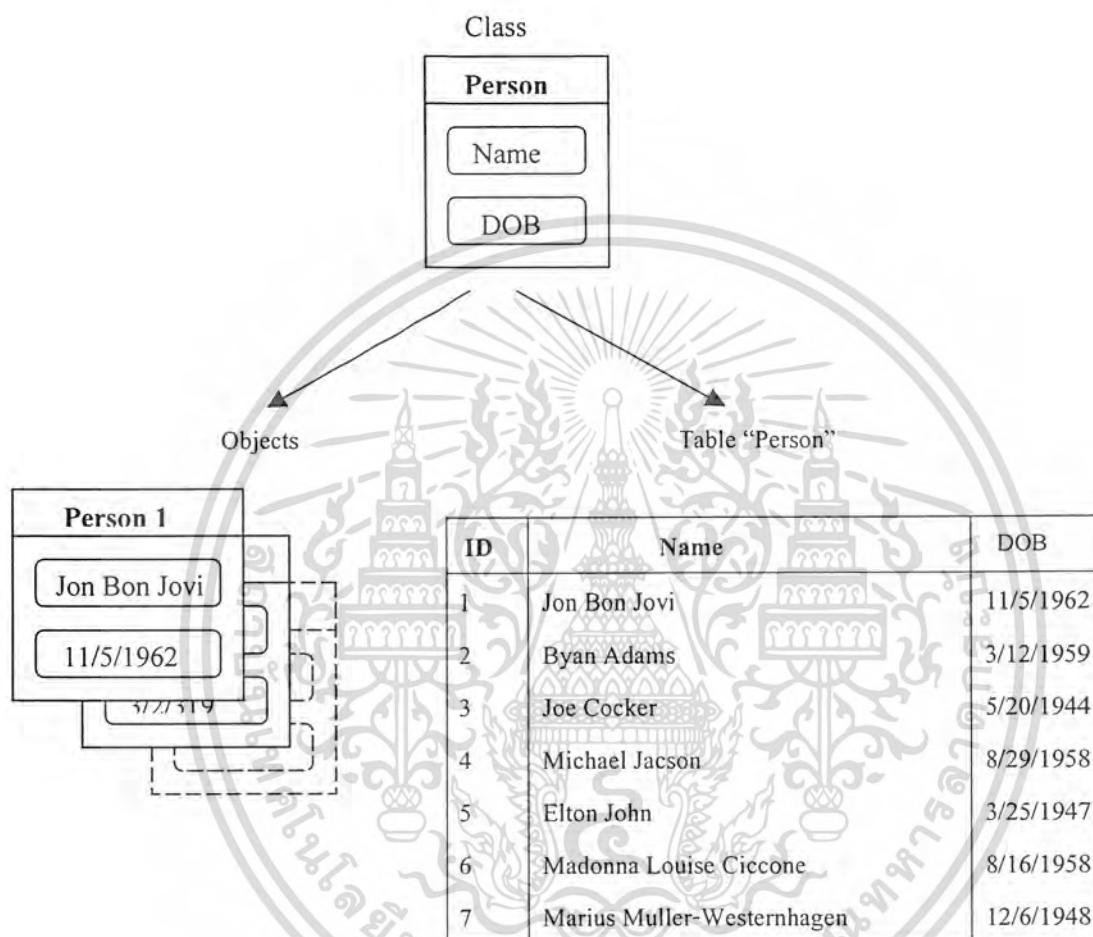
4.2.2 การเข้าถึงโดย Cache` SQL

ซึ่งในส่วนของ Cache` นั้นได้มีการออกแบบให้มีการรองรับการทำงานของ SQL ไว้อยู่แล้วซึ่ง Object-oriented database จะต้องสามารถรองรับการทำงานของ SQL ได้ โดยส่วนของ Cache` นั้นได้จัด Unified Data Architecture (UDA) ให้ติดต่อระหว่าง Cache` Database Engine กับ Application ได้ทั้งในโลกของแบบเก่า(Relational Table) และแบบใหม่ (Object-oriented instance)

4.2.2.1 แสดงถึงความสัมพันธ์ของ Class เป็น Object และ Table

ภายในสถาปัตยกรรม ของ Unified Data Architecture (UDA) ทั้ง Cache` Object และ Cache` SQL จะแสดงเป็น hierachy model .ใน Cache` Class ที่แตกต่างกัน ด้วยเหตุที่ Cache` Object ทำงานกับ Cache` SQL โดยการใช้ตารางในการอ้างอิง อย่างไรก็ตามจะต้องเป็น Single Data และ Unique Data

เมื่อเราพิจารณา viewpoint จาก Object-oriented เราเห็น คำ instance ของ Class จาก viewpont ของ Relation ซึ่งข้อมูลของ Class กลายมาเป็นข้อมูลของ Table ส่วนของแถวของ ตารางนั้นก็จะเป็นข้อมูลของแต่ละ instance ของแต่ละ Class ตามลำดับ



รูปที่ 4.3 Object และ ตารางที่สร้างจากคลาส

อย่างไรก็ตาม คุณสมบัติทั้งหมดของ Hierachy class ไม่สามารถที่จะแทนใน Relation ได้ทั้งหมด ได้ นี้ เป็นเหตุผลหนึ่งคุณสมบัติที่ Object และ Method มี ด้วยเหตุผลที่ ตารางประกอบไปด้วยข้อมูลที่ มี มี แม้ว่า จะแสดงให้เห็น ถึงความถูกต้องและมีความซับซ้อน ของ โครงสร้างของข้อมูล ของ Hierachy class ใน ตาราง ซึ่งทั้งหมดแสดงให้เห็นพฤติกรรม และขีดความสามารถที่เป็นไปได้

4.2.2.2 การเปลี่ยนจาก Class เป็น ตาราง

เมื่อเราต้องการที่จะทำการปรับเปลี่ยน Class ไปเป็น ตารางซึ่งเราสามารถนำ ทั้งสองรูปแบบมาทำการเปลี่ยนจากโลกของ Object เป็นโลกของ Relational เหมือนกับจะสามารถที่จะทำการเปลี่ยนจาก Object ไปเป็น Relation model ได้แต่ไม่มีความเป็น Equivalent สำหรับ Parameter class คุณสมบัติ multidimensional และ Instance method การเปลี่ยนไม่เป็นแสดงให้เป็นถึงคุณสมบัตินั้นของ Object ที่ทำได้ เพราะว่า เนื่องจากเรื่องของ การคำนวณ ซึ่งค่าไม่สามารถอยู่ภายในตาราง อันเนื่องมาจากเรื่องของ การ กำหนดในส่วนของ Class ซึ่งกำหนดให้เป็นค่าที่ต้องเป็นในส่วนของ Function มาแล้ว

From the Object Concept

Package

Class

Instance

Object identifier(OID)

Literal property

Reference to persistent object

Embedded object

List collection

Array collection

Stream

Index

Query

Class method

To the relation Concept

SQL schema

Table

Row

ID-column as primary key

Column

Foreign key

Individual column

Column with list field

Child table

BLOB

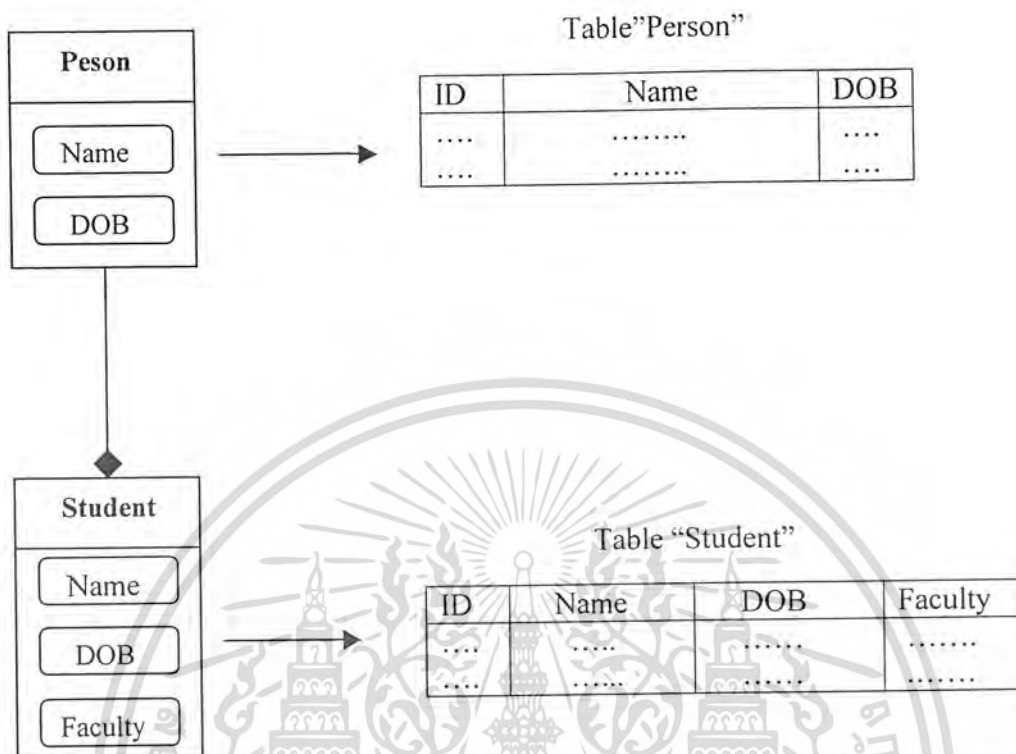
Index

Optional: stored procedure or view

Optional: stored procedure

4.2.2.3 การสืบทอดและ ตาราง

ในแนวความคิดในเรื่องการสืบทอดไม่ได้มีแนวความคิดนี้ใน Relational Model ซึ่งการที่เรา ทำการเปลี่ยนนั้นเราจะทำในส่วนของ Class แม่ มากกว่าจะเป็นลำดับที่สืบทอดมา โดยที่คุณสมบัติของการเปลี่ยนมาเป็น ตารางนั้นจะรวมเอาคุณสมบัติของ Class มาด้วยรวมถึงการสืบทอดมาจาก Class แม่ ทำให้ได้เห็นถึงความสำคัญของเรื่อง ข้อมูลที่อยู่ใน เร็คคอร์ด ในตารางหนึ่งๆ ที่จะเป็น Class แม่ที่รวมการ Instance ทั้งหมดของ Class ย่อยๆ



รูปที่ 4.4 การสืบทอดและการสร้างตาราง

จะเห็นได้ว่าตามตัวอย่างของ ตาราง Person จะมี Person ทั้งหมดอยู่ ส่วนตาราง Student นั้นก็จะเป็นส่วน Student เพียงอย่างเดียวแต่จะเพิ่มในส่วน Column ของ Faculty เข้าไปเป็น Subclass ของ Student

4.2.2.4 การอ้างอิงถึง Persistent Objects

การอ้างอิงถึง Persistent Objects นั้น นำมาใช้ใน Cache SQL ซึ่งจะเหมือนกับเป็น Foreign key ของอีกตารางหนึ่งส่วนพืวส์ ID จะรวมเป็นส่วนของ ID ที่เป็นส่วนหนึ่งของ OID ซึ่งเป็น Key จริงๆ การอ้างอิงในส่วนของ VB Object ซึ่งจะอยู่ใน Class ของ VB Object โดยจะแสดงในส่วนของ พืวส์ ของ VB ด้วย เมื่อมีการ Query ใน SQL ก็จะเป็นการ Join กันของ 2 ตาราง และ ใช้ Foreign key ในการ Join ด้วย

4.2.2.5 Embedded Object

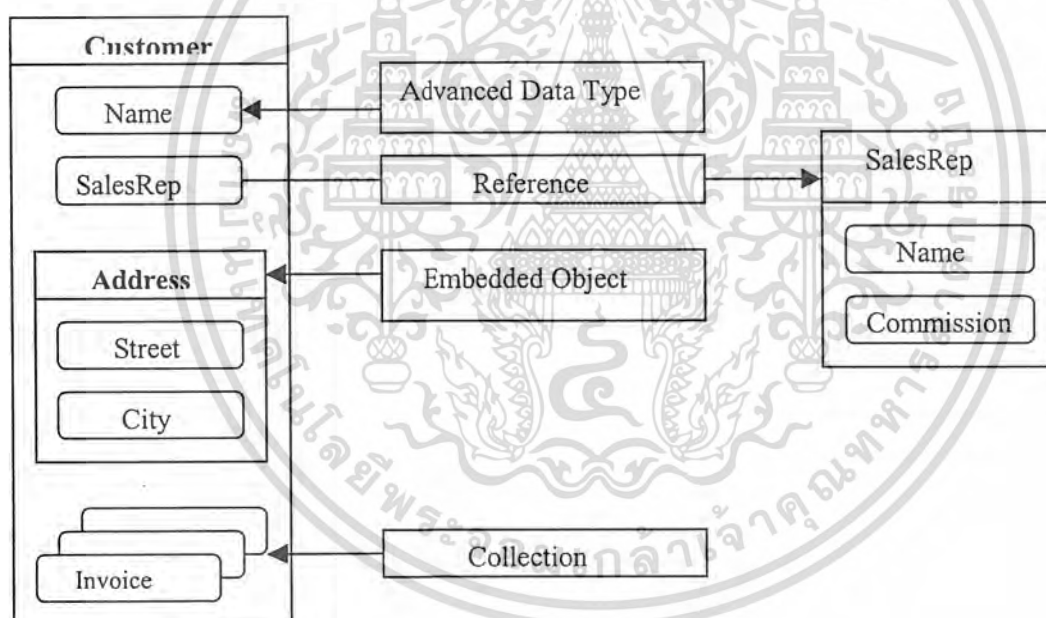
คุณสมบัติของ Embedded object เมื่อเปลี่ยนไปเป็น ตารางก็จะไปเป็นพืวส์ แต่ละพืวส์ ในตาราง ชื่อที่ทำการประกาศไว้จาก Embedded class รวมถึงชื่อคุณสมบัติต่างๆที่ Associate กันดูได้จากตัวอย่าง Street เป็น Embedded class ของ Address จะกลายเป็น Column ของตาราง Address_Street

4.2.2.6 การเก็บข้อมูลแบบ Array

กรเก็บข้อมูลแบบ Array ถูกแปลงไปเป็น Child table โดยการติดต่อกับ ตารางหลัก ด้วย Foreign key ชื่อของ Child table จะอยู่ในรูปแบบ ที่ชื่อมาจาก ตารางหลัก ตัวอย่าง ประกาศ Invoice โดยที่ Collection เป็นแบบ Array ของ Class Customer แล้วจะทำการสร้าง Child table เป็นชื่อ Customer_Invoice ซึ่ง Child table จะประกอบด้วยส่วนต่างๆดังนี้

- Foreign key ที่อยู่ตารางหลัก จะกลายเป็นชื่อของ ตารางหลักอีกตารางหนึ่ง
- Key เดียวที่อยู่ภายใน ตารางลูก จะมีชื่อเป็น element_key
- ส่วนประกอบต่างๆใน Class ก็จะถูกแปลงไปเป็นส่วนต่างๆที่ได้ระบุไว้ข้างต้นแล้ว

จะขอยกตัวอย่าง ที่มีความซับซ้อน ในการแปลงจาก Class ให้กลายเป็น ตารางโดยตัวอย่างจะมีความซับซ้อนมาก เมื่อผู้อ่านได้เห็นรูปประกอบซึ่งก็จะทำให้เกิดความเข้าใจมากยิ่งขึ้น



รูปที่ 4.5 การสร้างตารางจากชนิดข้อมูลแบบต่างๆ

Table"Customer"

ID	Name	Sales Rep	Address Street	Address City
1	Siwapong	3		
2	Prasit	2		
3	Adulsit	3		
4	Wisanu	1		
5	Mattana	1		

Table"SaleRep"

ID	Name	Commission
1	Ackachan	8%
2	Jessada	4%
3	Tosapol	2%

Table "Customer_Invoice"

Customer	Element_key	Invoice
4	1	
4	2	
2	1	

รูปที่ 4.5(ต่อ) การสร้างตารางจากข้อมูลชนิดต่าง ๆ

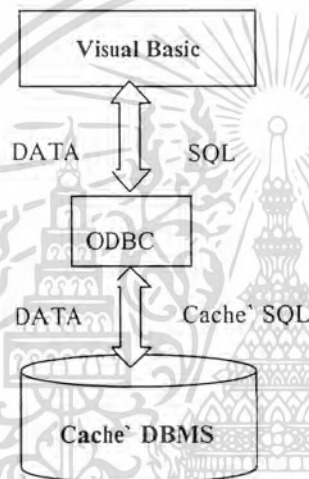
จากรูปข้างบนแสดงให้เห็นถึงการเปลี่ยนจากรูปแบบของ Class ที่มีความซับซ้อนให้อยู่ในรูปตาราง ซึ่งใน Cache' สนับสนุนในเรื่องนี้ ต่อไปจะเป็นตัวอย่างจาก Project ซึ่งจะแสดงให้เห็นถึงรูปแบบ และวิธีการต่าง ซึ่งใน Cache' จะมีการทำ Mapping นี้อยู่แล้วเมื่อใดที่เราทำการ สร้าง Class ใน Cache' ก็จะทำให้การ จะมีส่วนที่ให้อ้างอิง อยู่ 2 รูปแบบที่กล่าวไว้ข้างต้นแล้วคือ สามารถ เข้าถึง ด้วย Object และการเข้าถึงด้วย Cache' SQL (ขั้นตอนต่างๆ ในการสร้าง Class จากข้างบน และ การ ใช้ SQL Maneger ในการ ดูค่าต่างๆ ดูได้จาก ภาคผนวก ก และ ภาคผนวก ง)

บทที่ 5

ภาษาเรียกค้นตามตัวอย่างเชิงเวลา

เราทำการเพิ่มส่วนของเวลา (Temporal) เข้าไปในการเรียกค้นโดยใช้ตัวอย่าง ก็จะกลายเป็นส่วนของ Temporal QBE ซึ่งเราจะยึดหลักการของ QBE ไว้แต่จะเพิ่มส่วนของเวลาเป็น ช่วงของเวลา (Time range) ทำให้ได้ การเรียกค้นที่มีประสิทธิภาพมากขึ้นและจะได้ข้อมูลจริงในช่วงเวลานั้นๆ

5.1 สถาปัตยกรรม ของ TQBE



รูปที่ 5.1 สถาปัตยกรรมพื้นฐาน

สถาปัตยกรรม การติดต่อระหว่าง แอปพลิเคชัน กับ ระบบการจัดการฐานข้อมูล(DBMS) ซึ่งเราใช้ Visual Basic ในการ สร้างแอปพลิเคชันในการสร้าง SQL แล้วส่งเข้าไปสอบถามข้อมูลโดยผ่าน ODBC เข้าไปติดต่อกับ Cache DBMS โดยใน DBMS ของ Cache จะมี SQL Manager ที่ทำหน้าที่ในการจัดการ Object ไปเป็น ตาราง เมื่อได้ข้อมูลที่ต้องการแล้ว DBMS จะส่งค่ากลับไปให้ แอปพลิเคชัน ผ่าน ODBC กลับไปยังระบบ TQBE ซึ่ง ใน TQBE ของเราจะมองเห็น Schema ใน Cache DBMS

5.2 ส่วนกำหนดช่วงเวลา (Time Range)

การกำหนดช่วงเวลา ถูกใช้เพื่อการกำหนดความสัมพันธ์ของช่วงเวลาของตัวแปร โดยใช้ตัวปฏิบัติการทางคณิตศาสตร์ หรือตัวปฏิบัติการทางคณิตศาสตร์ หรือตัวปฏิบัติการทางเวลา[4] 13 ตัว ของ Allen ได้แก่ before, after, meets, met-by, overlaps, overlapped-by, during, include, starts, start-by, finishes, finished-by และ equal

ตัวอย่าง 4.1 ต้องการหาชื่อคนเกิดในช่วงวันที่ 1/3/1978 ถึง 1/8/1978

Time Range
1/3/1978 < Birthday < 1/8/1978

5.3 ส่วนกำหนดหน่วยย่อยเวลา (Time Granularity)

ส่วนกำหนดหน่วยย่อยเวลา ใช้ในการ ระบุเวลา (Time stamping) ได้แก่ day , month ,year

ตัวอย่าง 4.2 ต้องการระบุเวลา เป็นเดือน

Time Granularity
month

5.4 การพัฒนาระบบ TQBE

5.4.1 เครื่องคอมพิวเตอร์

ในการพัฒนาระบบ TQBE ได้ใช้อุปกรณ์ในการพัฒนาระบบระบบคือ

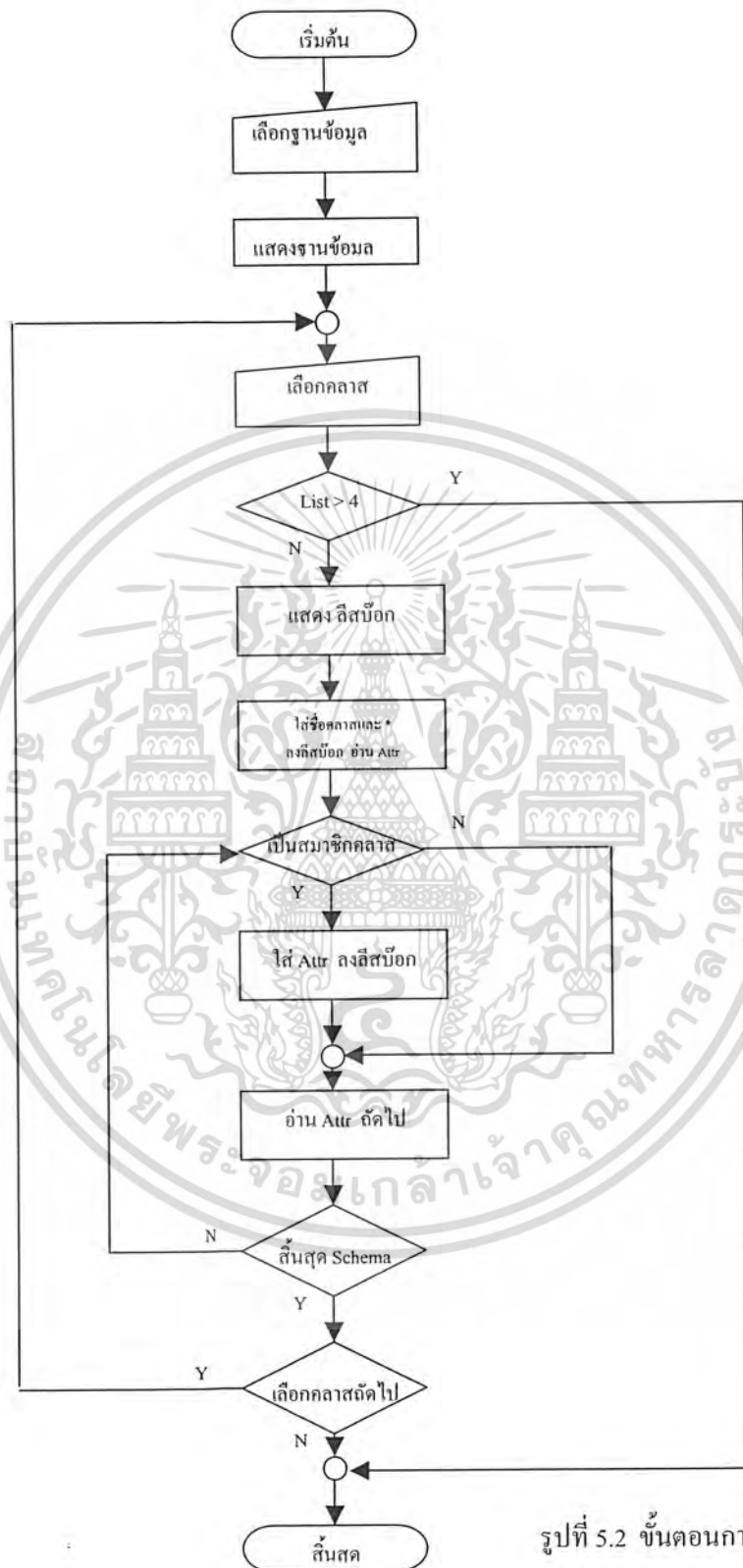
- CPU AMD K6-2 400 เมกะเฮิร์ต
- ความจุของฮาร์ดดิส 4.3 กิกะไบท์
- หน่วยความจำ 256 เมกะไบท์
- ระบบปฏิบัติการ วินโดวส์ 2000 เซิร์ฟเวอร์
- โปรแกรมไมโครซอฟท์ Visual Basic 6.0
- ระบบจัดการฐานข้อมูล Cache เวอร์ชัน 4.0

5.4.2 ระบบ TQBE

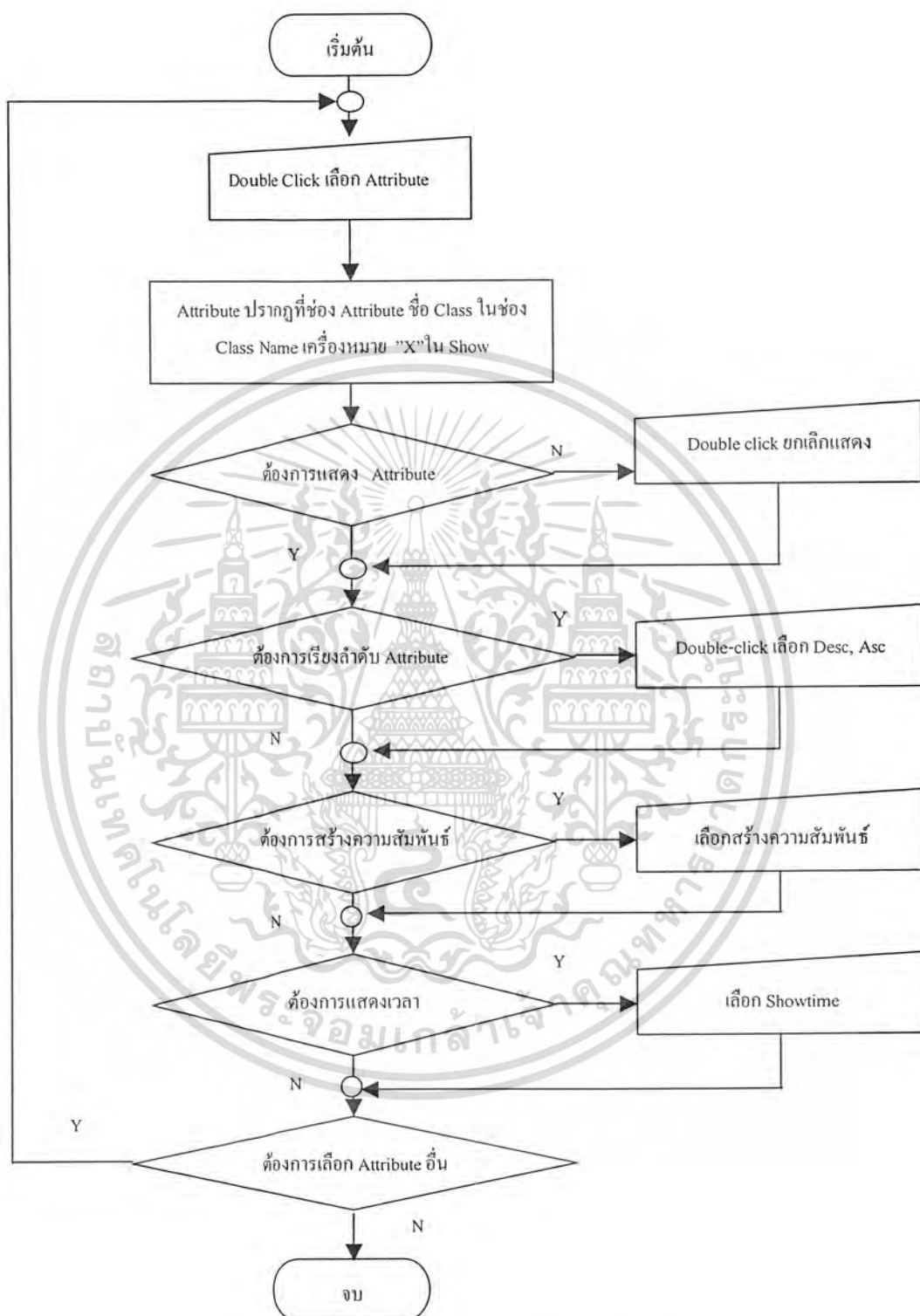
ระบบการสอบถามข้อมูลเชิงเวลา TQBE ได้พัฒนาขึ้นเพื่อช่วยให้ผู้ใช้สามารถสอบถามข้อมูลที่เกี่ยวข้องกับเวลา รายละเอียดการใช้งานระบบมีดังนี้

5.4.2.1 ขั้นตอนการทำงานของระบบ TQBE

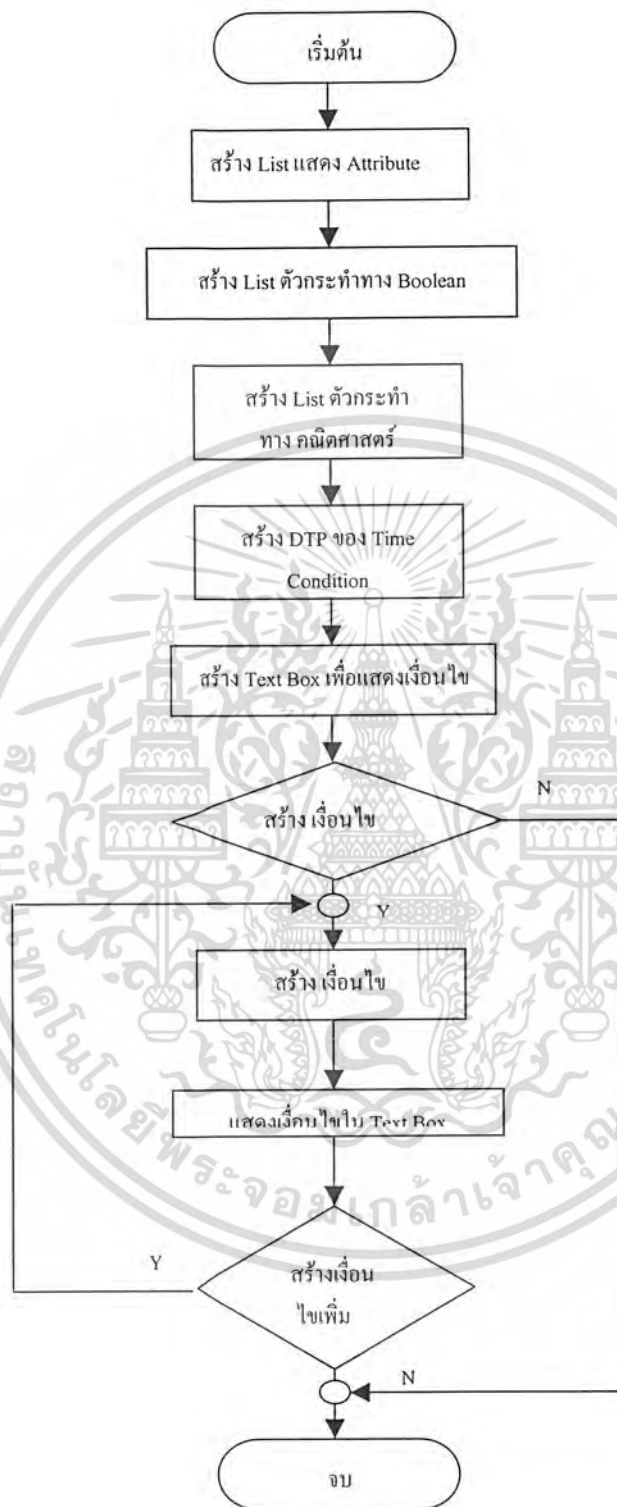
ขั้นตอนการทำงานของระบบได้อธิบายตาม Flow chat ต่างในการสร้างคำสั่ง SQL เพื่อไปสอบถามข้อมูลจากระบบฐานข้อมูลดังนี้



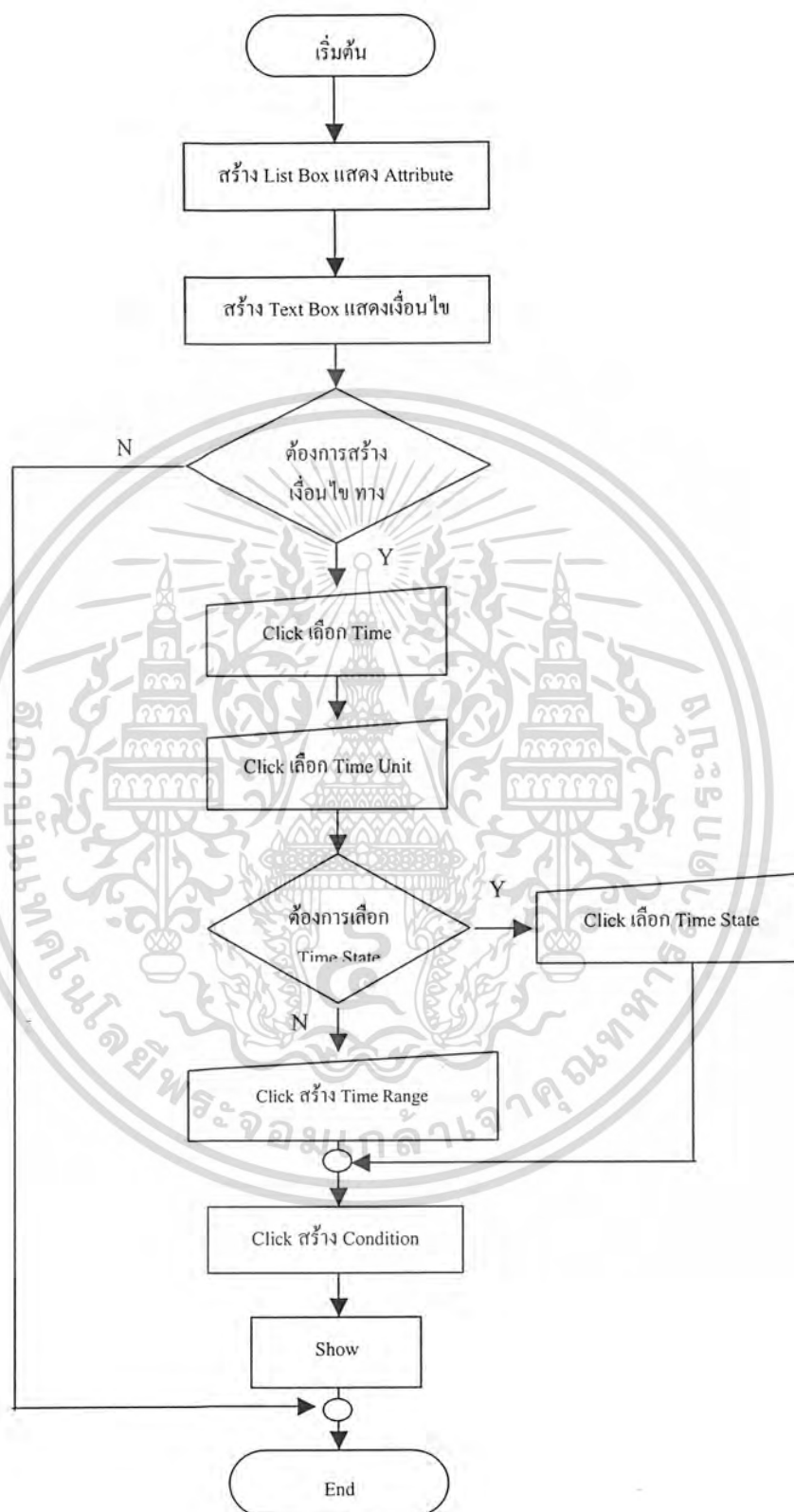
รูปที่ 5.2 ขั้นตอนการเลือกคลาส



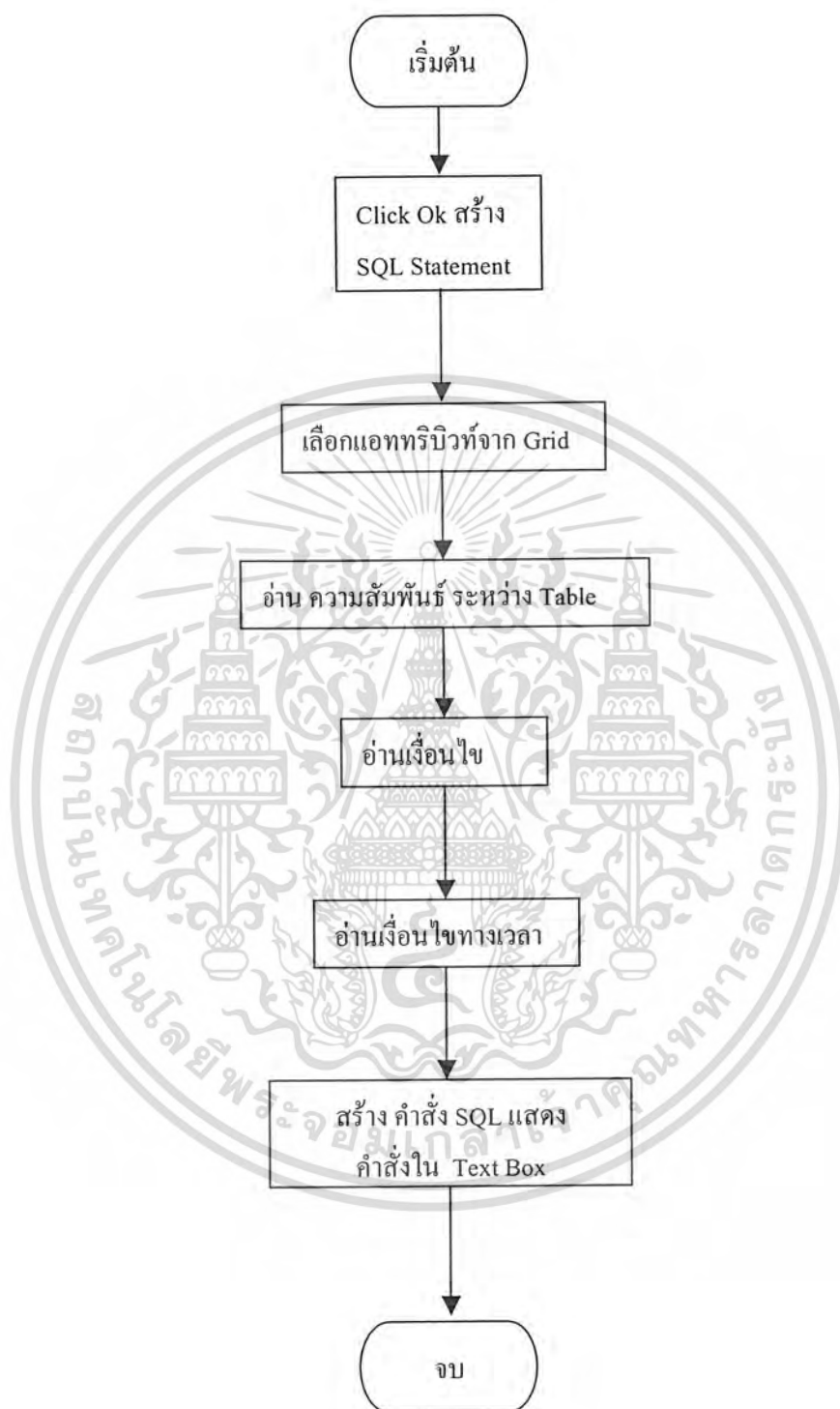
รูปที่ 5.3 ขั้นตอนเลือกแอททริบิวต์เพื่อสอบถามข้อมูล



รูปที่ 5.4 แสดงขั้นตอนการสร้างเงื่อนไข



รูปที่ 5.5 ขั้นตอนการสร้าง Timerange



รูปที่ 5.6 ขั้นตอนการสร้างคำสั่ง SQL

5.4.2.2 รายละเอียดของระบบ TQBE

เมื่อเปิดโปรแกรมเข้าสู่ โปรแกรม TQBE จะปรากฏดังรูปที่ 1 ซึ่งจะประกอบไปด้วยแท็บ 5 ส่วนด้วยกัน คือ ส่วนของ Query ส่วนของ Condition ส่วนของ Timerange ส่วนของ SQL และส่วนของ Result ของโปรแกรม ที่แท็บ Query ประกอบด้วยส่วนประกอบต่าง ๆ ดังนี้

ที่เฟรม Database Available จะเป็นส่วนสำหรับเลือก DSN ของฐานข้อมูลที่เราได้สร้างไว้ก่อนหน้านี้และ ส่วนสำหรับเลือก Classname ของฐานข้อมูลที่ได้สร้างไว้ในฐานข้อมูล(ส่วนการสร้าง DSN ดูตัวอย่างที่ภาคผนวก) โดยเมื่อเลือก Classname แล้วจะปรากฏโครงสร้างของคลาสที่เลือกว่ามีสมาชิกอะไรที่เราสามารถมาใช้ในการสอบถามข้อมูลหากเลือก (*) จะเป็นการเลือกสมาชิกทุกตัวของคลาสรวมถึง ID ของข้อมูลด้วย

รูปที่ 5.7 เริ่มต้นโปรแกรม TQBE

ที่เฟรม Relationship จะเป็นกำหนดความสัมพันธ์ระหว่างตาราง ถ้าเราเลือกตัวเลือก join คือการสร้างความสัมพันธ์ ระหว่างคลาส หากคลาสที่เลือกมาไม่มีสมาชิกคลาสที่สัมพันธ์กัน แต่ถ้าหากเลือกตัวเลือก unjoin จะเป็นการยกเลิกการสร้างความสัมพันธ์ของคลาสที่เลือกมาแม้จะมีสมาชิกของคลาสที่สัมพันธ์กัน ซึ่งมีค่าเท่ากับไม่ได้เลือกทั้ง 2 ตัวเลือก ทำให้ผลของการสอบถามข้อมูลจะเป็นการแสดงทุกข้อมูลที่เป็นไปได้ระหว่างสองคลาส

Showtime สำหรับกำหนดว่าจะให้แสดงเวลาของแต่ละสมาชิกของคลาสด้วยหรือไม่ ถ้าเลือกคือให้แสดงเวลาของสมาชิกคลาสด้วยในการแสดงผลของแบบสอบถามข้อมูล หากไม่เลือกก็ไม่ต้องแสดงเวลาในการแสดงผลของแบบสอบถามข้อมูล

ส่วน Grid สำหรับแสดงสมาชิกของคลาสที่ต้องการสอบถามข้อมูล โดยการสอบถามข้อมูลข้อมูลเราจะต้อง double click ที่สมาชิกคลาสที่ต้องการสอบถามข้อมูลจะปรากฏที่ ชื่อคลาส สมาชิกคลาส และเครื่องหมายบอกว่าเลือกมาเพื่อแสดงข้อมูลหรือเลือกมาใช้เป็นข้อมูลในการกำหนดเงื่อนไขในการสอบถาม

The screenshot shows a dialog box with the following components:

- Database Available:** A section containing a 'DSN' dropdown menu (currently showing '(-none-)') and a 'Class Name' dropdown menu. Below these is a list of available databases: Visual FoxPro Database, Visual FoxPro Tables, dBase Files - Word, FoxPro Files - Word, DeluxeCD, CACHE Samples, CACHE User, and **PROJECT** (which is highlighted).
- RelationShip:** A section with three radio buttons: 'Join', 'Unjoin', and 'Showtime'.
- Grid:** A table-like structure with columns for 'Attribute', 'ClassName', 'Sort', 'Show', 'Criteria', and 'Or'. It is currently empty.
- Buttons:** At the bottom, there are four buttons: 'OK', 'Cancel', 'Query', and 'Exit'.

รูปที่ 5.8 การเลือก DSN ที่สร้างจาก ODBC

Query Condition Time Range SQL Result

Database Available

DSN: PROJECT Class Name: Student

Student

- ID
- Major
- SFirstName
- SLastName
- StdID

Relationship

☐ Join ☐ Unjoin ☐ Showtime

Attribute

ClassName

Sort

Show

Criteria

Dir

OK Cancel Query Exit

รูปที่ 5.9 การเลือกคลาสจะปรากฏคลาสและสมาชิกคลาส

ปุ่ม OK สำหรับสร้างคำสั่ง SQL เพื่อที่จะส่งเข้าไปเรียกค้นข้อมูลที่ต้องการสอบถาม

ปุ่ม Cancel สำหรับล้างข้อมูลที่จะเรียกค้นทั้งหมดของ TQBE

ปุ่ม Query สำหรับส่งคำสั่ง SQL ลงไปเรียกค้นข้อมูลในฐานะข้อมูลที่ต้องการตาม DSN ที่เลือกไว้

ปุ่ม Exit สำหรับจบและออกจากหน้าต่าง TQBE ไปยังเริ่มโปรแกรม

เริ่มต้นการสอบถามข้อมูลเราจะต้องเลือกรฐานข้อมูล จากนั้นเลือกชื่อคลาสที่ต้องการสอบถามข้อมูลจะปรากฏ ลิสต์บ็อกพร้อมทั้งโครงสร้างในส่วน แอททริบิวท์ของข้อมูลในคลาส

ในการเลือกแอททริบิวท์ที่ต้องการสอบถามข้อมูลก็ให้ double-click ที่แอททริบิวท์นั้นทำให้แอททริบิวท์ดังกล่าวมาปรากฏที่กริดบ็อก

ภายในกริดบ็อกซ์จะมีส่วนที่บอกว่า Attribute ,Classname, Sort , Show หลังจากเลือกแอททริบิวท์ที่ต้องการแล้วจะมาปรากฏชื่อคลาสที่ช่อง Classname ชื่อแอททริบิวท์ที่ Attribute และแสดงเครื่องหมาย X ที่ช่อง Show คือเป็นการเลือกแสดงข้อมูล เราสามารถเลือกที่จะแสดงข้อมูลได้โดยการ double-click ที่ช่องที่มีเครื่องหมาย X ที่ช่อง Sort จะเป็นช่องที่บอกว่าเลือกใช้แอททริบิวท์ใดในการเรียงลำดับข้อมูลโดยการ double-click ที่ช่องที่ต้องการในการเลือกที่จะเรียงลำดับแบบ มากไปน้อยหรือน้อยไปมาก กลับ(Delete) ที่เป็นพิมพ์

Query Condition Time Range SQL Result

Database Available

DSN: PROJECT Class Name: Major

Student

- ID
- Major
- SFirstName
- SLastName
- StdID

Major

- ID
- MajorDesc
- MajID

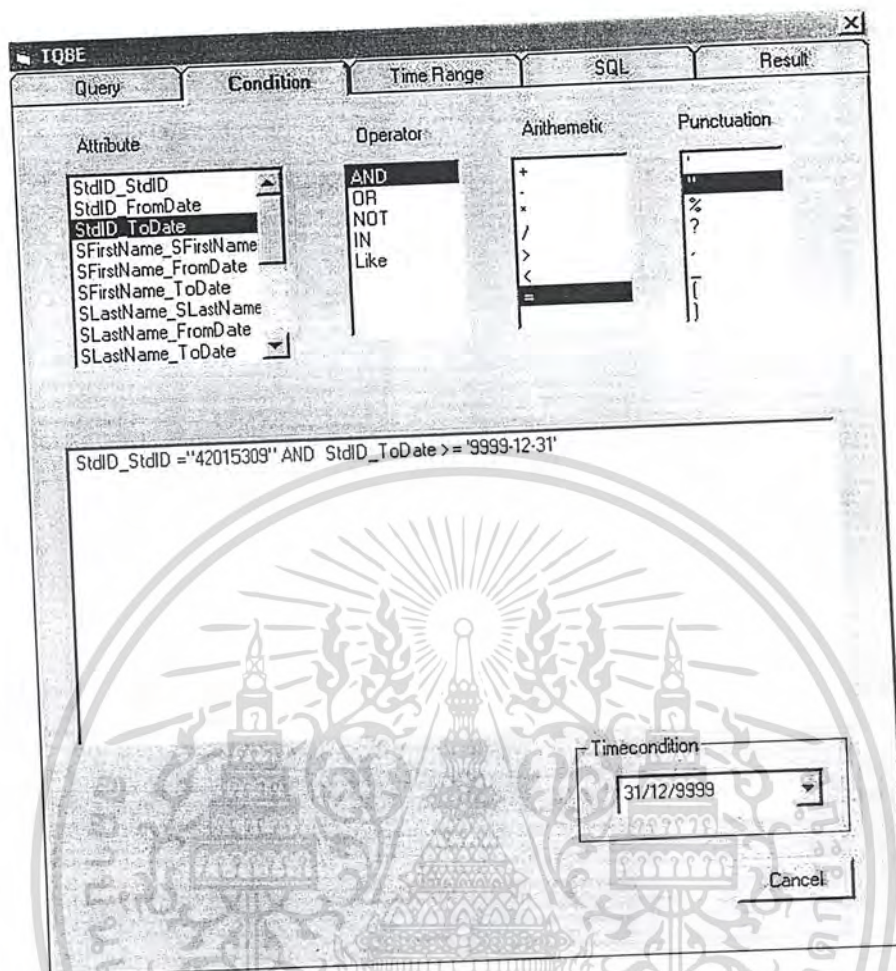
Relationship

☒ Join ☐ Union ☒ Showtime

Attribute	StdID	SFirstName	SLastName	MajID	MajorDesc
ClassName	Student	Student	Student	Major	Major
Sort					
Show	x	x	x	x	x
Criteria					
Or					

OK Cancel Query Exit

รูปที่ 5.10 การสอบถามข้อมูลเชิงเวลา



รูปที่ 5.11 หน้าต่าง Condition

ที่หน้าต่าง Condition ประกอบด้วยส่วนประกอบต่าง ๆ ดังนี้

- ที่ Attribute สำหรับเลือกสมาชิกคลาสที่จะใช้สร้างเป็นเงื่อนไขในการสอบถามข้อมูล โดยการ double click ที่สมาชิกข้อมูลจะปรากฏที่กล่องข้อความ
- ที่ Operator สำหรับเลือกตัวกระทำระหว่างสมาชิกข้อมูลที่เป็นเงื่อนไขทางลอจิก ในการสอบถามข้อมูล โดยการ double click ที่สมาชิกข้อมูลจะปรากฏที่กล่องข้อความ
- ที่ Arithmetic สำหรับเลือกตัวกระทำกันระหว่างสมาชิกข้อมูลที่เป็นเงื่อนไขทางคณิตศาสตร์ ในการสอบถามข้อมูล โดยการ double click ที่สมาชิกข้อมูลจะปรากฏที่กล่องข้อความ
- ที่ Punctuation สำหรับเลือกเครื่องหมายในการกำหนดชนิดข้อมูล และเครื่องหมายวรรคตอน ของเงื่อนไข ที่มาใช้ในการสอบถามข้อมูล โดยการ double click ที่สมาชิกข้อมูลจะปรากฏที่กล่องข้อความ
- ที่ Timecondition สำหรับระบุเงื่อนไขทางเวลาซึ่งสามารถกำหนดเวลาของแต่ละ สมาชิกคลาสได้ตามต้องการ โดยการ double click ที่สมาชิกข้อมูลจะปรากฏที่กล่องข้อความ

Timeattribute:

- StdID_FromDate
- StdID_ToDate
- SFirstName_Fro
- SFirstName_Tol
- SLastName_Frc
- SLastName_To
- MajID_FromDa
- MajID_ToDate
- MajorDesc_Fror

StdID_ToDate >= '2002-01-01' and SFirstName_ToDate >= '2002-01-01' and SLastName_ToDate >= '2002-01-01' and MajID_ToDate >= '2002-01-01' and MajorDesc_ToDate >= '2002-01-01'

Time

☒ Time

TimeUnit

☐ Day

☐ Month

☐ Year

Time state

☐ Current

☐ History

TimeRange

From: 12/12/2001 To: 31/12/9999

OK Cancel

รูปที่ 5.12 การสร้างเงื่อนไขทางเวลาแบบปัจจุบัน

ที่หน้าต่าง TimeRange ประกอบด้วยส่วนประกอบต่าง ๆ ดังนี้

ที่ Timeattribute จะแสดงสมาชิกของคลาสที่เกี่ยวกับเวลา

ที่ เฟรม Time จะเป็นส่วนระบุเงื่อนไขทางเวลาแบบเป็นช่วงเวลา ซึ่งประกอบด้วย TimeUnit Timestate TimeRange โดยที่ TimeUnit จะเป็นการระบุว่าจะดูเงื่อนไขทางเวลาเป็นวันที่ เดือน หรือ ปี กับ Timestate จะเป็นการระบุว่าการสอบถามข้อมูลปัจจุบันหรืออดีตของข้อมูล ทั้งสองต้องกำหนดร่วมกัน โดยถ้ากำหนด Timestate เป็น current จะเป็นการเรียกค้นข้อมูลที่เวลาสิ้นสุดของข้อมูลมากกว่าหรือเท่ากับปัจจุบัน ถ้ากำหนด Timestate เป็น History จะเป็นการสอบถามข้อมูลที่ เวลาสิ้นสุดข้อมูลน้อยกว่าค่าเวลาที่กำหนดในช่อง From ของส่วนกำหนดช่วงเวลา แต่ถ้ากำหนดเฉพาะหน่วยของเวลาแต่ไม่กำหนด Timestate จะเป็นสอบถามข้อมูลที่มีช่วงเวลาอยู่ระหว่าง From กับ To ของส่วนกำหนดช่วงเวลา การกด OK จะเป็นการสร้างเงื่อนไขทางเวลาที่เลือกไว้หากต้องการยกเลิกก็สามารถกด Cancel ได้ ซึ่งถ้าหากไม่เข้ามำหนดส่วน Timerange จะเป็นการสอบถามทุกข้อมูลที่เป็นไปได้

TQBE

Query Condition Time Range SQL Result

TimeAttribute

StdID_FromDate
StdID_ToDate
SLastName_FromDate
SLastName_ToDate
MajrID_FromDate
MajrID_ToDate
MajorDesc_FromDate
MajorDesc_ToDate

StdID_FromDate <= '2001-12-31' and SLastName_FromDate <= '2001-12-31' and MajrID_FromDate <= '2001-12-31' and MajorDesc_FromDate <= '2001-12-31'

Time

TimeUnit
☐ Day
☐ Month
☒ Year

☒ Time

Time state
☐ Current
☒ History

TimeRange
 From 3/3/2001 To 31/12/9999

OK Cancel

รูปที่ 5.13 การสร้างเงื่อนไขทางเวลาแบบ ประวัติข้อมูล

การเลือก Time state ว่าเป็น Current ระบบจะสร้างช่วงเวลาที่เป็นปัจจุบัน หรือ History ระบบจะสร้างช่วงเวลาที่เป็นประวัติข้อมูล ถ้าหากไม่เลือก Time state จะต้องระบุช่วงเวลาเอง

TQBE

Query	Condition	Time Range	SQL	Result
-------	-----------	------------	-----	--------

Timeattribute

StdID_FromDate
StdID_ToDate
SFirstName_Fro
SFirstName_Tol
SLastName_Frc
SLastName_To
MajID_FromDa
MajID_ToDate
MajorDesc_Fror

StdID_FromDate >= '2001-01-01' and StdID_ToDate <= '9999-12-31' and SFirstName_FromDate >= '2001-01-01' and SFirstName_ToDate <= '9999-12-31' and SLastName_FromDate >= '2001-01-01' and SLastName_ToDate <= '9999-12-31' and MajID_FromDate >= '2001-01-01' and MajID_ToDate <= '9999-12-31' and MajorDesc_FromDate >= '2001-01-01' and MajorDesc_ToDate <= '9999-12-31'

Time

TimeUnit
☐ Day
☐ Month
☒ Year

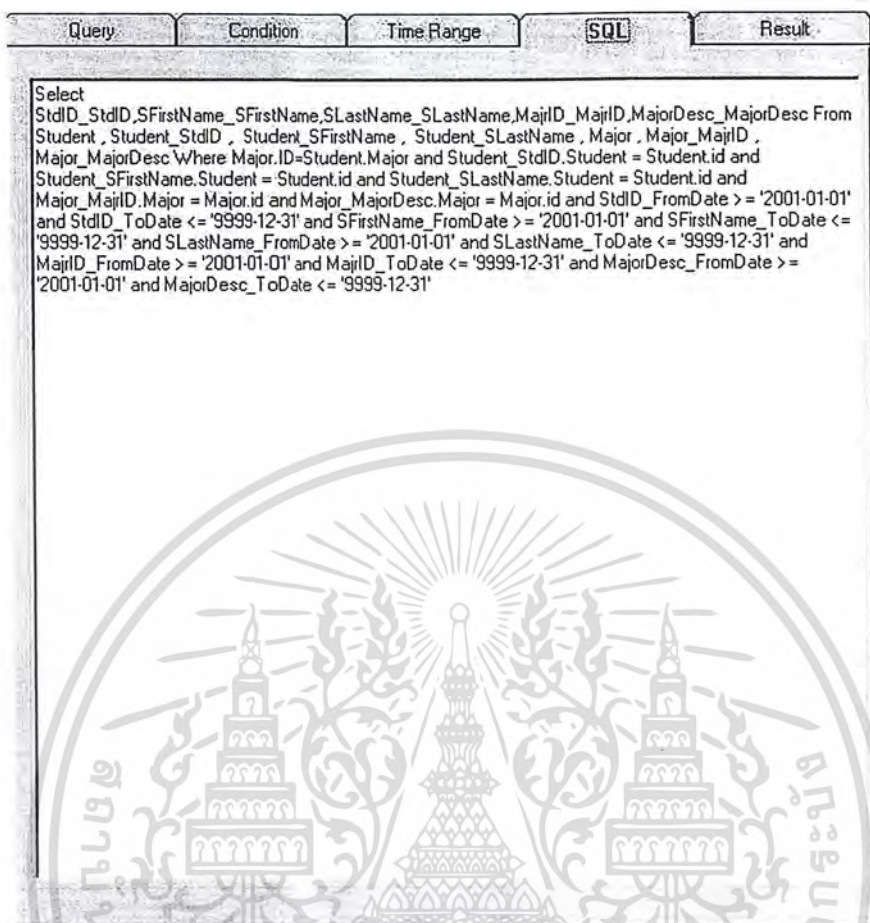
☒ Time

Time state
☐ Current
☐ History

TimeRange
 From 3/12/2001 To 31/12/9999

OK Cancel

รูปที่ 5.14 การสร้างเงื่อนไขทางเวลาแบบช่วงเวลา



รูปที่ 5.15 คำสั่ง SQL ที่ได้ของระบบ TQBE

ที่หน้าต่าง SQL จะประกอบไปด้วยส่วนองค์ประกอบต่าง ๆ ดังนี้

จะมีกล่องข้อความสำหรับแสดงคำสั่ง SQL ที่ได้จากการสร้างแบบสอบถามทั้งหมดโดยจะเป็นตัวคำสั่ง SQL ที่จะส่งเข้าไปเรียกค้นข้อมูลจากแบบสอบถามที่สร้างขึ้น

Query	Condition	Time Range	SQL	Result
StdID : StdID	SFirstName : SFirstN...	SLastName : SLastN...	MajrID : MajrID	MajorD
40215309	Surasit	Prommaree	01	Compu
40215321	Jirapong	Seerapan	01	Compu
42015323	Akachan	Pitisakornwit	01	Compu
42015324	AdulSit	Pengpa	02	Teleco
40215325	Wichitchai	Peerapa	02	Teleco
42015326	Mattana	Wongpisarn	03	Electro
42015327	Jiradech	Apawuttkoon	03	Electro
42015328	Tossapon	Tummala	04	Control
42015329	Jessada	Kamput	04	Control

รูปที่ 5.16 ผลจากการสอบถามข้อมูลไม่แสดงเวลา

ที่หน้าต่าง Result จะประกอบไปด้วยส่วนองค์ประกอบต่าง ๆ ดังนี้

จะมี กล่องแสดงผลการทำงานของ การสอบถามข้อมูลซึ่งสามารถแสดงได้ทั้งแบบที่แสดงเวลาของสมาชิกข้อมูลแต่ละสมาชิกที่เปลี่ยนไป และไม่แสดงเวลาแล้วแต่ความต้องการของผู้สอบถามข้อมูล

Query	Condition	Time Range	SQL	Result			
StdID	StdID	StdID	FromDate	ToDate	SFirstName	SFirstN...	SFirstN
40215309		31/12/2001		31/12/9999	Surasit		31/12/
40215321		31/12/2001		31/12/9999	Jirapong		31/12/
42015323		31/12/2001		31/12/9999	Akachan		31/12/
42015324		31/12/2001		31/12/9999	AdulSit		31/12/
40215325		31/12/2001		31/12/9999	Wichitchai		31/12/
42015326		31/12/2001		31/12/9999	Mattana		31/12/
42015327		31/12/2001		31/12/9999	Jiradech		31/12/
42015328		31/12/2001		31/12/9999	Tossapon		31/12/
42015329		31/12/2001		31/12/9999	Jessada		31/12/

รูปที่ 5.17 ผลการสอบถามข้อมูลที่แสดงเวลาด้วย

บทที่ 6

สรุปผลการทดลอง

6.1 ตัวอย่างการทำงานของระบบ TQBE

จากการพัฒนาระบบ TQBE นี้เราจะใช้ SQL เข้าไปสอบถามข้อมูลเชิงเวลาซึ่งในฐานข้อมูล Cache' นั้นมีความสามารถในการแปลงข้อมูลแบบ Object Oriented Database ให้เป็น Relational Database จึงทำให้เราสามารถ
ใช้ SQL เข้าไปสอบถามข้อมูลได้ TQBE ที่ได้สร้างขึ้นจะสนับสนุนการเก็บข้อมูลแบบ 1 สมาชิกข้อมูลคลาสจะ
ประกอบไปด้วย ตัวข้อมูล ระยะเวลาเริ่มต้น ระยะเวลาสิ้นสุด และมีชนิดข้อมูลเป็น Collection แบบ อาร์เรย์ คือทุก
สมาชิกคลาสสามารถเปลี่ยนแปลงได้ ยกเว้นสมาชิกคลาสที่เก็บ OID ของความสัมพันธ์

พิจารณาคลาสที่ใช้ในการทดสอบจะประกอบด้วย คลาส Student, Course, Major, Teacher และ คลาส
Register ซึ่ง

คลาส Student จะประกอบด้วยสมาชิกคลาสดังตาราง

สมาชิกคลาส	ส่วนประกอบข้อมูล	คำอธิบาย
StdID	StdID FromDate ToDate	เก็บ รหัสนักศึกษา
SFirstName	SFirstName FromDate ToDate	เก็บ ชื่อนักศึกษา
SLastName	SLastName FromDate ToDate	เก็บ นามสกุลนักศึกษา
Major	Major	เก็บ OID ของสาขาที่ศึกษา

รูปที่ 6.1 สมาชิกของคลาส Student

คลาส Course จะประกอบด้วยสมาชิกคลาสดังตาราง

สมาชิกคลาส	ส่วนประกอบข้อมูล	คำอธิบาย
CourseID	CourseID FromDate ToDate	เก็บรหัสรายวิชา
CourseDesc	CourseDesc FromDate ToDate	เก็บรายชื่อวิชา
Credit	Credit FromDate ToDate	เก็บหน่วยกิตรายวิชา
Teacher	Teacher	เก็บ OID ของอาจารย์ผู้สอน

รูปที่ 6.2 สมาชิกของคลาส Course

คลาส Teacher จะประกอบด้วยสมาชิกดังตาราง

สมาชิกคลาส	ส่วนประกอบข้อมูล	คำอธิบาย
Tid	Tid FromDate ToDate	เก็บรหัสอาจารย์
TFirstName	TFirstName FromDate ToDate	เก็บชื่ออาจารย์
TLastName	TLastName FromDate ToDate	เก็บนามสกุล

รูปที่ 6.3 สมาชิกของคลาส Teacher

คลาส Major จะประกอบด้วยสมาชิกดังตาราง

สมาชิกคลาส	ส่วนประกอบข้อมูล	คำอธิบาย
MajrID	MajrID FromDate ToDate	เก็บรหัสสาขา
MajorDesc	MajorDesc FromDate ToDate	เก็บรายชื่อสาขา

รูปที่ 6.4 สมาชิกคลาส Major

คลาส Register ประกอบด้วย

สมาชิกคลาส	ส่วนประกอบข้อมูล	คำอธิบาย
Student	Student	เก็บ OID ของนักศึกษา
Course	Course	เก็บ OID ของรายวิชา
RegDate	RegDate FromDate ToDate	เก็บวันที่ลงทะเบียน
Grade	Grade FromDate ToDate	เก็บเกรด
Semester	Semester FromDate ToDate	เก็บเทอมที่ศึกษา
EdYear	EdYear FromDate ToDate	เก็บปีการศึกษา

รูปที่ 6.5 สมาชิกคลาส RegDate

ID	StdID	SFirstName	SLastName	Major
1	{<42015309,1/1/1998,31/12/2001>,<40215309,31/12/2001,31/21/9999>}	{<Prasit,1/1/1998,31/12/2001>,<Surasit,31/12/2001,31/21/9999>}	{<Promree,1/1/1998,31/12/2001>,<Prommaree,31/12/2001,31/21/9999>}	1
2	{<42015321,1/1/1998,31/12/2001>,<40215321,31/12/2001,31/21/9999>}	{<Siwapong,1/1/1998,31/12/2001>,<Jirapong,31/12/2001,31/21/9999>}	{<Seepan,1/1/1998,31/12/2001>,<Seerapan,31/12/2001,31/21/9999>}	1
3	{<42015323,1/1/1998,31/12/2/9999>}	{<Akachan,1/1/1998,31/12/9999>}	{<Pitisakornwit,1/1/1998,31/12/9999>}	1
4	{<42015324,1/1/1998,31/12/2/9999>}	{<Adulsit,1/1/1998,31/12/9999>}	{<Pengpa,1/1/1998,31/12/9999>}	2
5	{<42015325,1/1/1998,31/12/2/2001>,<40215309,31/12/2001,31/12/9999>}	{<Wisanu,1/1/1998,31/12/2001>,<Wichitchai,31/12/2001,31/21/9999>}	{<Tookeaw,1/1/1998,31/12/2001>,<Peerapa,31/12/2001,31/21/9999>}	2
6	{<420153026,1/1/1998,31/12/9999>}	{<Mattana,1/1/1998,31/12/9999>}	{<Wongpisarn,1/1/1998,31/12/9999>}	3
7	{<42015327,1/1/1998,31/12/2/9999>}	{<Jiradech,1/1/1998,31/12/9999>}	{<Apawuttikoon,1/1/1998,31/12/9999>}	3
8	{<42015328,1/1/1998,31/12/2/9999>}	{<Tossapon,1/1/1998,31/12/9999>}	{<Tummala,1/1/1998,31/12/9999>}	4
9	{<42015329,1/1/1998,31/12/2/9999>}	{<Jessada,1/1/1998,31/12/9999>}	{<Kamput,1/1/1998,31/12/9999>}	4

ตารางที่ 6.1 ตาราง Student

OID	MajrID	MajorDesc
1	{<01,1/1/1998,31/12/9999>}	{<Computer Engineering,1/1/1998,31/12/9999>}
2	{<02,1/1/1998,31/12/9999>}	{<Telacommunication,1/1/1998,31/12/9999>}
3	{<03,1/1/1998,31/12/9999>}	{<Electronics Engineering,1/1/1998,31/12/9999>}
4	{<04,1/1/1998,31/12/9999>}	{<Control Engineering,1/1/1998,31/12/9999>}

ตารางที่ 6.2 ตาราง Major

OID	CourseID	CourseDesc	Credit
1	{<01074003,1/1/1998,31/12/2001>,<00000001,31/12/2001,31/12/9999>}	{<Advance Database System,1/1/1998,31/12/2001>,<Database System,31/12/2001,31/12/9999>}	{<3,1/1/1998,31/12/9999>}
2	{<01074006,1/1/1998,31/12/2001>,<00000002,31/12/2001,31/12/9999>}	{<Network Programming,1/1/1998,31/12/2001>,<Computer Network,31/12/2001,31/12/9999>}	{<3,1/1/1998,31/12/9999>}
3	{<01074007,1/1/1998,31/12/2001>,<00000003,31/12/2001,31/12/9999>}	{<Computer Systems Security,1/1/1998,31/12/2001>,<Security system,31/12/2001,31/12/9999>}	{<3,1/1/1998,31/12/9999>}
4	{<01074012,1/1/1998,31/12/2001>,<00000004,31/12/2001,31/12/9999>}	{<Compilers System,1/1/1998,31/12/2001>,<Compilers Design,31/12/2001,31/12/9999>}	{<3,1/1/1998,31/12/9999>}
5	{<01074022,1/1/1998,31/12/2001>,<00000005,31/12/2001,31/12/9999>}	{<Computer Graphics,1/1/1998,31/12/2001>,<Computer 3D,31/12/2001,31/12/9999>}	{<3,1/1/1998,31/12/9999>}
6	{<01074002,1/1/1998,31/12/2001>,<00000006,31/12/2001,31/12/9999>}	{<Digital System,1/1/1998,31/12/2001>,<Digital Design,31/12/2001,31/12/9999>}	{<3,1/1/1998,31/12/9999>}

ตารางที่ 6.3 ตาราง Course

OID	Tid	TFirstName	TLastName
1	{<0001,01/01/1998,31/12/9999>}	{<Bandit,01/01/1998,31/12/9999>}	{<Passaya,01/01/1998,31/12/9999>}
2	{<0002,01/01/1998,31/12/2001>,<0005,31/12/2001,31/12/9999>}	{<Surat,01/01/1998,31/12/2001>,<Surin,31/12/2001,31/12/9999>}	{<SapraKun,01/01/1998,31/12/2001>,<SuksuKoon,31/12/2001,31/12/9999>}
3	{<0003,01/01/1998,31/12/2001>,<0006,31/12/2001,31/12/9999>}	{<Surachai,01/01/1998,31/12/2001>,<Supachai,31/12/2001,31/12/9999>}	{<Nakronchai, 01/01/1998,31/12/2001>,<Sangnakorn,31/12/2001,31/12/9999>}
4	{<0004,01/01/1998,31/12/9999>}	{<Warawan,01/01/1998,31/12/9999>}	{<Chanthong,01/01/1998,31/12/9999>}

ตารางที่ 6.4 ตาราง Teacher

OID	Course	Student	RegDate	Grade	Semester	EdYear
1	1	1	{<1998-01-01,01/01/1998,31/12/9999>}	{<A,01/01/1998,31/12/9999>}	{<2,01/01/1998,31/12/9999>}	{<1998,01/01/1998,31/12/9999>}
2	2	1	{<1998-01-01,01/01/1998,31/12/9999>}	{<B+,01/01/1998,31/12/9999>}	{<2,01/01/1998,31/12/9999>}	{<1998,01/01/1998,31/12/9999>}
3	3	1	{<1998-01-01,01/01/1998,31/12/9999>}	{<B,01/01/1998,31/12/9999>}	{<2,01/01/1998,31/12/9999>}	{<1998,01/01/1998,31/12/9999>}
4	1	2	{<1998-01-01,01/01/1998,31/12/9999>}	{<B,01/01/1998,31/12/9999>}	{<2,01/01/1998,31/12/9999>}	{<1998,01/01/1998,31/12/9999>}
5	2	2	{<1998-01-01,01/01/1998,31/12/9999>}	{<B+,01/01/1998,31/12/9999>}	{<2,01/01/1998,31/12/9999>}	{<1998,01/01/1998,31/12/9999>}
6	3	2	{<1998-01-01,01/01/1998,31/12/9999>}	{<B,01/01/1998,31/12/9999>}	{<2,01/01/1998,31/12/9999>}	{<1998,01/01/1998,31/12/9999>}
7	4	3	{<1998-01-01,01/01/1998,31/12/9999>}	{<C+,01/01/1998,31/12/9999>}	{<2,01/01/1998,31/12/9999>}	{<1998,01/01/1998,31/12/9999>}
8	5	3	{<1998-01-01,01/01/1998,31/12/9999>}	{<C+,01/01/1998,31/12/9999>}	{<2,01/01/1998,31/12/9999>}	{<1998,01/01/1998,31/12/9999>}
9	6	3	{<1998-01-01,01/01/1998,31/12/9999>}	{<B,01/01/1998,31/12/9999>}	{<2,01/01/1998,31/12/9999>}	{<1998,01/01/1998,31/12/9999>}
10	6	4	{<1998-01-01,01/01/1998,31/12/9999>}	{<B,01/01/1998,31/12/9999>}	{<2,01/01/1998,31/12/9999>}	{<1998,01/01/1998,31/12/9999>}
11	5	4	{<1998-01-01,01/01/1998,31/12/9999>}	{<B,01/01/1998,31/12/9999>}	{<2,01/01/1998,31/12/9999>}	{<1998,01/01/1998,31/12/9999>}
12	4	4	{<1998-01-01,01/01/1998,31/12/9999>}	{<B+,01/01/1998,31/12/9999>}	{<2,01/01/1998,31/12/9999>}	{<1998,01/01/1998,31/12/9999>}
13	1	5	{<1998-01-01,01/01/1998,31/12/9999>}	{<A,01/01/1998,31/12/9999>}	{<2,01/01/1998,31/12/9999>}	{<1998,01/01/1998,31/12/9999>}
14	3	5	{<1998-01-01,01/01/1998,31/12/9999>}	{<C+,01/01/1998,31/12/9999>}	{<2,01/01/1998,31/12/9999>}	{<1998,01/01/1998,31/12/9999>}

OID	Course	Student	RegDate	Grade	Semester	EdYear
15	4	5	{<1998-01-01,01/01/1998,31/12/9999>}	{<C+,01/01/1998,31/12/9999>}	{<2,01/01/1998,31/12/9999>}	{<1998,01/01/1998,31/12/9999>}
16	1	6	{<1998-01-01,01/01/1998,31/12/9999>}	{<B+,01/01/1998,31/12/9999>}	{<2,01/01/1998,31/12/9999>}	{<1998,01/01/1998,31/12/9999>}
17	4	6	{<1998-01-01,01/01/1998,31/12/9999>}	{<B+,01/01/1998,31/12/9999>}	{<2,01/01/1998,31/12/9999>}	{<1998,01/01/1998,31/12/9999>}
18	6	6	{<1998-01-01,01/01/1998,31/12/9999>}	{<B,01/01/1998,31/12/9999>}	{<2,01/01/1998,31/12/9999>}	{<1998,01/01/1998,31/12/9999>}

ตารางที่ 6.5(ต่อ) ตาราง RegDate

ตัวอย่างที่ 6.1 ให้หา รายชื่อ นามสกุล นักศึกษา และสาขาที่เรียน ที่รหัส 01 ณ ปัจจุบัน?

The screenshot shows a database query tool interface. At the top, there are tabs for 'Query', 'Condition', 'Time Range', 'SQL', and 'Result'. Below these, the 'Database Available' section shows 'DSN: PROJECT' and 'Class Name: Major'. Two tables, 'Student' and 'Major', are listed with their attributes: 'Student' has ID, Major, SFirstName, SLastName, and StdID; 'Major' has ID, MajorDesc, and MajID. The 'RelationShip' section has radio buttons for 'Join' (selected), 'Union', and a checkbox for 'Showtime'. Below this is a table with columns 'Attribute', 'SFirstName', 'SLastName', 'MajorDesc', and 'MajID'. The 'Show' row has 'x' marks under 'SFirstName', 'SLastName', and 'MajorDesc'. At the bottom are buttons for 'OK', 'Cancel', 'Query', and 'Exit'.

รูปที่ 6.6 เลือกสมาชิกข้อมูลในตัวอย่าง 6.1

Query	Condition	Time Range	SQL	Result
Attribute	Operator	Arithmetic	Punctuation	
SLastName_SLastName SLastName_FromDate SLastName_ToDate MajrID_MajrID MajrID_FromDate MajrID_ToDate MajorDesc_MajorDesc MajorDesc_FromDate MajorDesc_ToDate	AND OR NOT IN Like	+ - * / > < =	" % ? , { }	
MajrID_MajrID = '01'				

รูปที่ 6.7 การระบุเงื่อนไขในตัวอย่าง 6.1

Time	
☒ Time	TimeUnit ☐ Day ☐ Month ☒ Year Time state ☒ Current ☐ History
TimeRange From 1/1/2002 To 30/12/9999	
OK Cancel	

รูปที่ 6.8 การระบุเวลาปัจจุบันในตัวอย่างที่ 6.1

กำหนด Time state เป็น Current แล้วสร้าง SQL จะได้คำสั่ง SQL ดังนี้

```

Select SFirstName_SFirstName,SLastName_SLastName,MajorDesc_MajorDesc
From Student , Student_SFirstName , Student_SLastName , Major , Major_MajorDesc , Major_MajrID
Where Major.ID=Student.Major and Student_SFirstName.Student = Student.id and
Student_SLastName.Student = Student.id and Major_MajorDesc.Major = Major.id and
Major_MajrID.Major = Major.id and MajrID_MajrID = '01' and SFirstName_ToDate >= '2002-01-01'
and SLastName_ToDate >= '2002-01-01' and MajorDesc_ToDate >= '2002-01-01' and
MajrID_ToDate >= '2002-01-01'

```

Query	Condition	Time Range	SQL	Result
SFirstName_SFirstN...	SLastName_SLastN...	MajorDesc_MajorD...		
Surasit	Prommaree	Computer Engineerri...		
Jirapong	Seerapan	Computer Engineerri...		
Akachan	Pitisakornwit	Computer Engineerri...		

รูปที่ 6.9 ผลลัพธ์ของตัวอย่าง 6.1

เมื่อกด Query จะได้ผลการสอบถามข้อมูลข้อมูลดังรูป 6.8 ถ้าหากต้องการแสดงเวลาด้วยก็เลือก Show Time ด้วยและจะได้ผลดังรูปที่ 6.9

Query	Condition	Time Range	SQL	Result
SFirstName_SFirstN...	SFirstName_FromD...	SFirstName_ToDate	SLastName_SLastN...	SLastN
Surasit	31/12/2001	31/12/9999	Prommaree	31/12/
Jirapong	31/12/2001	31/12/9999	Seerapan	31/12/
Akachan	1/1/1998	31/12/9999	Pitisakornwit	1/1/19

รูปที่ 6.10 ผลลัพธ์ของตัวอย่าง 6.1 เมื่อเลือก Showtime

ตัวอย่างที่ 6.2 ให้หา รายชื่อ นามสกุล นักศึกษา และสาขาที่เรียน ที่รหัส 01 ที่เป็นประวัติข้อมูลที่เวลาเริ่มต้นน้อยกว่าปัจจุบัน

Time

☒ Time

TimeUnit

☐ Day
☐ Month
☒ Year

Time state

☐ Current
☒ History

TimeRange

From To

OK

Cancel

รูปที่ 6.11 กำหนด Time state เป็น History ของตัวอย่างที่ 6.2

ทำการเลือกสมาชิกข้อมูลที่ต้องการสอบถามเหมือนตัวอย่างที่ 6.1 แต่กำหนด Time state เป็น History ทำการสร้าง SQL จะได้คำสั่งดังข้างล่าง

```
Select SFirstName_SFirstName,SFirstName_FromDate,SFirstName_ToDate,SLastName_SLastName,
      SLastName_FromDate,SLastName_ToDate,MajorDesc_MajorDesc,MajorDesc_FromDate,
      MajorDesc_ToDate
From Student , Student_SFirstName , Student_SLastName , Major , Major_MajorDesc , Major_MajrID
Where Major.ID=Student.Major and Student_SFirstName.Student = Student.id and
      Student_SLastName.Student = Student.id and Major_MajorDesc.Major = Major.id and
      Major_MajrID.Major = Major.id and MajrID_MajrID ='01' and
      SFirstName_FromDate <= '2002-12-31' and SLastName_FromDate <= '2002-12-31' and
      MajorDesc_FromDate <= '2002-12-31' and MajrID_FromDate <= '2002-12-31'
```

Query	Condition	Time Range	SQL	Result
SFirstName_SFirstN...	SFirstName_FromD...	SFirstName_ToDate	SLastName_SLastN...	SLastN
Prasit	1/1/1998	31/12/2001	Promree	1/1/19:
Prasit	1/1/1998	31/12/2001	Prommaree	31/12/
Surasit	31/12/2001	31/12/9999	Promree	1/1/19:
Surasit	31/12/2001	31/12/9999	Prommaree	31/12/
Siwapong	1/1/1998	31/12/2001	Seepan	1/1/19:
Siwapong	1/1/1998	31/12/2001	Seerapan	31/12/
Jirapong	31/12/2001	31/12/9999	Seepan	1/1/19:
Jirapong	31/12/2001	31/12/9999	Seerapan	31/12/
Akachan	1/1/1998	31/12/9999	Pitisakornwit	1/1/19:

รูปที่ 6.12 ผลลัพธ์ของการสอบถามข้อมูลตัวอย่างที่ 6.2 เลือกแสดงเวลา

ตัวอย่างที่ 6.3 ให้หา รายชื่อ นามสกุล นักศึกษา และสาขาที่เรียน ที่ช่วงเวลาระหว่าง 01/01/1998 ถึง 30/12/9999

ทำการเลือกสมาชิกของข้อมูลส่วนกำหนดทำการเลือกหน่วยของเวลาว่าเป็น (วันที่)Day (เดือน)Mounth หรือ ปี(Year) ส่วนกำหนด ช่วงเวลา(Time state) ไม่ต้องใส่แต่ใส่เวลาในเฟรมช่วงเวลาเมื่อสร้าง SQL จะได้คำสั่ง SQL ดังข้างนี้

```
Select SFirstName_SFirstName,SLastName_SLastName,MajorDesc_MajorDesc
From Student , Student_SFirstName , Student_SLastName , Major , Major_MajorDesc
Where Major.ID=Student.Major and Student_SFirstName.Student = Student.id and
      Student_SLastName.Student = Student.id and Major_MajorDesc.Major = Major.id and
      SFirstName_FromDate <= '2001-12-30' and SFirstName_ToDate >= '1998-1-1' and
```

SLastName_FromDate <= '2001-12-30' and SLastName_ToDate >= '1998-1-1' and

MajorDesc_FromDate <= '2001-12-30' and MajorDesc_ToDate >= '1998-1-1'

Attribute	SFirstName	SLastName	MajorDesc
ClassName	Student	Student	Major
Sort			
Show	x	x	x
Criteria			
Or			

OK Cancel Query Exit

รูปที่ 6.13 การเลือกสมาชิกข้อมูลจากตัวอย่างที่ 6.3

SFirstName_FromDate <= '2001-12-30' and SFirstName_ToDate >= '1998-1-1' and
 SLastName_FromDate <= '2001-12-30' and SLastName_ToDate >= '1998-1-1' and
 MajorDesc_FromDate <= '2001-12-30' and MajorDesc_ToDate >= '1998-1-1'

Time

☒ Time

TimeUnit

☒ Day

☐ Month

☐ Year

Time state

☐ Current

☐ History

TimeRange

From 1/1/1998 To 30/12/2001

OK Cancel

รูปที่ 6.14 การกำหนดช่วงเวลาจากตัวอย่างที่ 6.3

Query	Condition	Time Range	SQL	Result
SFirstName_SFirstN...	SLastName_SLastN...	MajorDesc_MajorD...		
Prasit	Promree	Computer Engineeri...		
Siwapong	Seepan	Computer Engineeri...		
Akachan	Pitisakornwit	Computer Engineeri...		
AdulSit	Pengpa	Telecommunication ...		
Wisanu	Tookeaw	Telecommunication ...		
Mattana	Wongpisarn	Electronic Engineerr...		
Jiradech	Apawuttikoon	Electronic Engineerr...		
Tossapon	Tummala	Control Engineering		
Jessada	Kamput	Control Engineering		

รูปที่ 6.15 แสดงผลจากตัวอย่างที่ 6.3

ตัวอย่างที่ 4 ให้หา ชื่อ นามสกุลนักศึกษาในปัจจุบัน และชื่อรายวิชาที่เรียนวันที่ 31/12/2001 และเกรดที่ได้

Attribute	SFirstName	SLastName	CourseDesc	Grade
ClassName	Student	Student	Course	Register
Sort				
Show	☒	☒	☒	☒
Criteria				
Or				

OK Cancel Query Exit

รูปที่ 6.16 การเลือกสมาชิกข้อมูลในตัวอย่างที่ 6.4

สร้างเงื่อนไขทางเวลาโดยผู้ใช้งานและสร้าง SQL ได้ดังนี้

```
Select SFirstName_SFirstName,SLastName_SLastName,CourseDesc_CourseDesc,Grade_Grade
From Student , Student_SFirstName , Student_SLastName , Course , Course_CourseDesc , Register ,
Register_Grade
Where Student.ID=Register.Student and Course.ID=Register.Course and
Student_SFirstName.Student = Student.id and Student_SLastName.Student = Student.id and
Course_CourseDesc.Course = Course.id and Register_Grade.Register = Register.id and
```

SFirstName_ToDate >= '9999-12-31' AND SLastName_ToDate >= '9999-12-31' AND
CourseDesc_FromDate <= '2001-12-30'

Query	Condition	Time Range	SQL	Result
Attribute	Operator	Arithmetic	Punctuation	
SFirstName_ToDate SLastName_SLastName SLastName_FromDate SLastName_ToDate CourseID_CourseID CourseID_FromDate CourseID_ToDate CourseDesc_CourseDesc CourseDesc_FromDate	AND OR NOT IN Like	+ - * / > < =	" % ? , (	
SFirstName_ToDate >= '9999-12-31' AND SLastName_ToDate >= '9999-12-31' AND CourseID_FromDate <= '2001-12-30'				

รูปที่ 6.17 การสร้างเงื่อนไขจาก Timecondition ในตัวอย่างที่ 6.4

Query	Condition	Time Range	SQL	Result
SFirstName_SFirstN...	SLastName_SLastN...	CourseDesc_Cours...	Grade_Grade	
Surasit	Prommaree	Advanced Databas...	A	
Surasit	Prommaree	Network Programmi...	B+	
Surasit	Prommaree	Computer Systems ...	B	
Jirapong	Seerapan	Advanced Databas...	B	
Jirapong	Seerapan	Network Programmi...	B+	
Jirapong	Seerapan	Computer Systems ...	B	
Akachan	Pitisakornwit	Compilers System	C+	
Akachan	Pitisakornwit	Computer Graphics	C+	
Akachan	Pitisakornwit	Digital System	B	
AdulSiti	Pengpa	Digital System	B	
AdulSiti	Pengpa	Computer Graphics	B	
AdulSiti	Pengpa	Compilers System	B+	
Wichitchai	Peerapa	Advanced Databas...	A	
Wichitchai	Peerapa	Computer Systems ...	C+	
Wichitchai	Peerapa	Compilers System	C+	
Mattana	Wongpisarn	Advanced Databas...	B+	
Mattana	Wongpisarn	Compilers System	B+	
Mattana	Wongpisarn	Digital System	B	

รูปที่ 6.18 ผลของการสอบถามในตัวอย่างที่ 6.4

ตัวอย่างที่ 5 ให้หา รหัส ชื่อ นามสกุลนักศึกษา รหัสสาขา และสาขาที่เรียน ที่มีรหัสนักศึกษา เป็น 42015309

Attribute	StdID	SFirstName	SLastName	MajrID	MajorD
ClassName	Student	Student	Student	Major	Major
Sort					
Show	x	x	x	x	x
Criteria					
Or					

OK Cancel Query Exit

รูปที่ 6.19 การเลือกสมาชิกข้อมูลในตัวอย่างที่ 6.5

จะสร้างคำสั่ง SQL ได้ดังนี้

```
Select StdID_StdID,SFirstName_SFirstName,SLastName_SLastName,MajrID_MajrID,MajorDesc_MajorDesc
From Student , Student_StdID , Student_SFirstName , Student_SLastName , Major , Major_MajrID ,
Major_MajorDesc
Where Major.ID=Student.Major and Student_StdID.Student = Student.id and
Student_SFirstName.Student = Student.id and Student_SLastName.Student = Student.id and
Major_MajrID.Major = Major.id and Major_MajorDesc.Major = Major.id and
StdID_StdID ='42015309'
```

Query	Condition	Time Range	SQL	Result
StdID_StdID	SFirstName_SFirstN...	SLastName_SLastN...	MajrID_MajrID	MajorD
42015309	Prasit	Promree	01	Compu
42015309	Prasit	Prommaree	01	Compu
42015309	Surasit	Promree	01	Compu
42015309	Surasit	Prommaree	01	Compu

รูปที่ 6.20 ผลการสอบถามข้อมูลในตัวอย่างที่ 6.5

6.2 สรุปการทำวิจัย

ระบบการจัดการฐานข้อมูล TQBE ที่ได้พัฒนาขึ้นนั้นเพื่อสนับสนุนข้อมูลเชิงเวลา รูปแบบจะเป็นแบบกราฟฟิคตามแนวคิด QBE เพื่อให้ผู้ใช้สามารถได้ง่าย

ระบบ TQBE ที่พัฒนาขึ้นนี้สนับสนุนการระบุเวลาที่แอททริบิวต์ของข้อมูลโดยสมมุติว่าทุกแอททริบิวต์สามารถเปลี่ยนแปลงตามเวลาได้ ดังนั้นในหนึ่งแอททริบิวต์ข้อมูลจะประกอบไปด้วยเซตของข้อมูล ระยะเวลาเริ่มต้น และระยะเวลาสิ้นสุดข้อมูล การสอบถามข้อมูลจะสามารถสอบถามได้ทั้งข้อมูลปัจจุบัน ประวัติของข้อมูล ข้อมูลในช่วงเวลาที่ต้องการ และสามารถระบุได้ด้วยว่าต้องการแสดงช่วงเวลาที่เป็นจริงของข้อมูลด้วยหรือไม่ไม่แสดงได้แต่จะไม่สนับสนุนการเปลี่ยนแปลงระหว่างความสัมพันธ์ที่เกิดขึ้น

การสอบถามข้อมูลจะสร้างการสอบถามที่เป็น QBE ก่อนแล้วแปลงเป็น SQL เพื่อเข้าไปสอบถามข้อมูลโดยผ่าน ODBC เข้าไปและส่งข้อมูลมาแสดงผลที่ระบบเนื่องจากการสร้างคำสั่ง SQL ที่ด้านบนไม่เข้าไปมอง Schema ของฐานข้อมูลทำให้ระบบ TQBE ที่สร้างขึ้นใช้ได้กับฐานข้อมูลที่มีรูปแบบการเก็บแบบฐานข้อมูลตัวอย่าง

ระบบ TQBE ที่พัฒนาขึ้นขณะนี้มีความสามารถเพียงสอบถามข้อมูลเท่านั้นไม่สามารถจัดการกับข้อมูลเช่น ใส่ข้อมูล เปลี่ยนแปลงข้อมูลและลบข้อมูลได้ และการสอบถามข้อมูลจะเป็นการสอบถามแบบง่าย ๆ ไม่ซับซ้อน ไม่สารสร้างตัวปฏิบัติการรวมจาก หน้าต่างสร้างแบบสอบถามได้

ส่วนการกำหนดเงื่อนไขจะสามารถกำหนดได้ทั้งเงื่อนไขทั่วไปและเงื่อนไขทางเวลาโดยสร้างจากส่วน Timecondition

ส่วนการระบุช่วงเวลาจะเป็นการกำหนดเงื่อนไขทางเวลาที่ข้อมูลทุกตัวที่ต้องการสอบถามจะมีช่วงเวลาตรงตามช่วงที่กำหนดทุกแอททริบิวต์

การระบุเวลากับข้อมูลจะระบุเป็น คอลเลกชันแบบอาร์เรย์ แทนแบบลิสต์เพื่อจะโปรเจกต์ข้อมูลไปเป็นตารางได้และสามารถส่งคำสั่ง SQL เข้าไปสอบถามข้อมูลได้

6.3 ข้อเสนอแนะในการวิจัย

สำหรับระบบ TQBE ที่พัฒนาขึ้นนี้มีความสามารถเพียงสอบถามข้อมูลได้เท่านั้นไม่สามารถจัดการกับข้อมูลได้ดังนั้นในการพัฒนาต่อไปควรเพิ่มส่วนจัดการข้อมูล ส่วนการสอบถามข้อมูลที่ซับซ้อน ส่วนของการสร้างฟอร์ม และส่วนสร้างรายงานเข้าไปเพื่อให้มีความสามารถครบถ้วน

สำหรับการพัฒนาครั้งนี้ใช้คำสั่ง SQL ในการเข้าถึงในการพัฒนาระบบ ยังมีอีกวิธีหนึ่งที่สามารถเข้าถึงฐานข้อมูลได้คือใช้ Object Script ของตัวจัดการฐานข้อมูลซึ่งอาจจะดีกว่าใช้ SQL เข้าไปสอบถามข้อมูลก็ได้

เอกสารอ้างอิง

- [1] Abdulla Uz Tansel and Group “Temporal Database” The Benjamin/Cummings Publishing Company, Inc. 1993
- [2] Abraham Silberschatz , Henry F. Korth , S. Sudrashan. “Database System Concept” Third Edition. The McGraw-Hill Companies, Inc 1997
- [3] Gerald V. Post “Database management System” Internation Edition , Western Kentucky University , 1999
- [4] Tanat Aramrusmevanich “การพัฒนาระบบจัดการฐานข้อมูลเชิงสัมพันธ์แบบคิวบ์อิงเวลา สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังม2000
- [5] W. Kirsten and Group “Object-Oriented Aplication Development Using the Cache’ Postrelational Database” ACM Computing Classification,1998
- [6] ชัชวาล ศุภเกษม “Microsoft Visual Basic 6.0 ภาคปฏิบัติ” ซีเอ็ดดูเคชั่น จำกัด(มหาชน),1999

ภาคผนวก ก

การเขียนโปรแกรม Visual Basic ติดต่อกับฐานข้อมูล Cache'

เทคนิคการเขียนโปรแกรมติดต่อกับ Cache' นั้น มีด้วยกัน 2 วิธี

วิธีที่ 1 การติดต่อ โดยใช้ Cache' Objects

การสร้าง Visual Basic Project

โดยทั่วไปแล้ว Cache' Object สามารถเข้าถึงจาก VB โดยไม่ต้องมีการตั้งค่า Configuration แต่อย่างไรก็ตามก็จำเป็นต้อง Set ค่าอะไรบ้างอย่างด้วยเล็กน้อย ซึ่งใน VB มีการกำหนดคุณสมบัติบางอย่างเช่น การกำหนด Object's type ตัวอย่างการ ประกาศ และการสร้าง New Object list สำหรับ Type CacheObject.SysList

```
Dim List as CacheObject.SysList
```

```
Set List= new CacheObject.SysList
```

ซึ่งเป็นการยอมให้ Vb สามารถ Pop – Up window ได้ สำหรับตัวอย่างเป็นการบอกถึง Type สำหรับ Instance ใหม่สามารถเลือกจาก Object จาก List ที่มี Type เป็นไปได้

การ Select Project Component

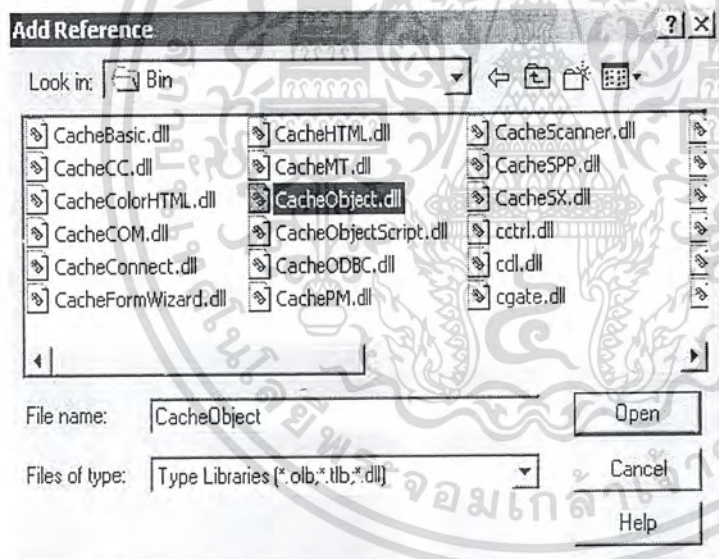
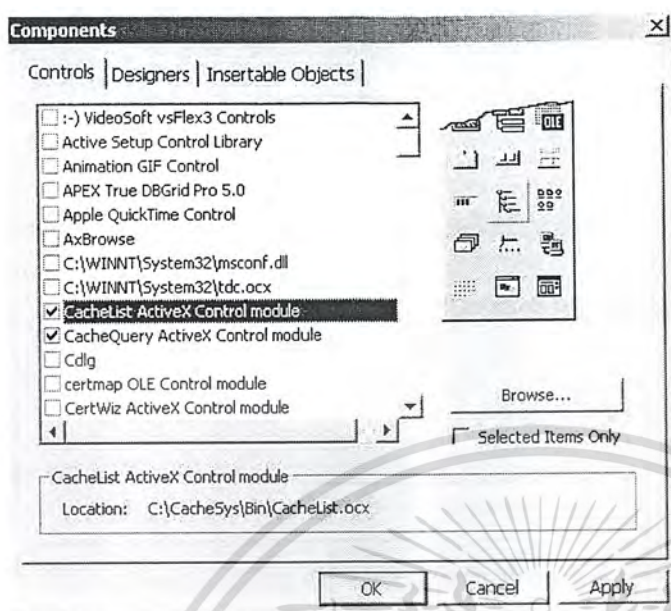
Step แรกใน Configure Project ใน VB เราจำเป็นต้อง add Component ใน VB มีให้ก่อน วิธีการเลือก Component ดังต่อไปนี้

1. เปิด Component ใน VB โดยไปที่ menu Project / Component
2. Select box ที่ซ้ายมือ ของ CacheList ActiveX Control
3. Select box ที่ซ้ายมือ ของ CacheQuery ActiveX Control

การเลือก Project References

Step ที่ 2 เป็นการ Configure Project ใน VB โดยการกำหนด Project References เมื่อ Cache' Object ถูกใช้ใน Step แรก เราจะต้องเรียก File CacheObject.dll

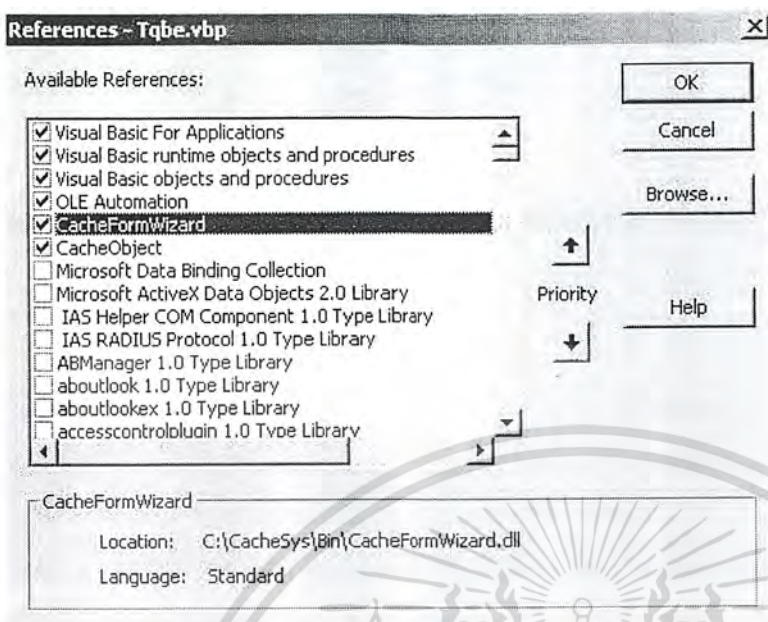
1. เปิด Window ไปที่ References อยู่ที่ Menu Option ที่ Project\Reference
2. Click ที่ Search..... button
3. เลือก ไฟล์ CacheObject.dll ใน Subdirectory ที่ ใช้ ติดตั้ง โดยปกติแล้วจะอยู่ที่ \Cachesys\Bin และ Click ที่ Button Open



รูปแสดงการเลือก Component Cache

4. เลือก box ทางซ้ายมือ

- CacheObject
- CacheFormWizard



รูปแสดงการเลือก Component Cache'

การติดต่อกับ Server

โดยปกติแล้ว Application ที่ทำหน้าที่เป็น Client จะต้องมีการสร้าง Cache' Factory และสร้างการเชื่อมต่อ กับ Server เราสามารถเขียนตาม Code ดังต่อไปนี้

' Visual Basic Code

Option Explicit

Dim Factory As CacheObject.Factory

Dim Connectstring As String

Dim Success As Boolean

Private Sub Form_Load()

' Create an instance of the Factory object

' "Set" must be used to assign an object value

Set Factory = CreateObject("CacheObject.Factory")

' If no connection exists to a server

' Create one

If Not Factory.IsConnected() then

' The string with the connection parameters

```

        ' can be specified explicitly;
Connectstring ="cn_ip tcp"127.0.0.1(1972) :User "
Connectstring =Factory.ConnectDlg()
Success = Factory.Connect(Connectstring)
End If
End Sub

```

การสร้าง Object ใหม่

Method New() เป็น Method ที่อยู่ใน Cache' Factory โดยการสร้าง Object ใหม่ของ Cache' Object นั้นดังนี้

```

Dim Aircraft As Object
Set Aircraft = Factory.New("Aviation.Aircraft")

```

ชื่อของ Class Cache' เราต้องการให้รับ argument เดียว สำหรับ Method New() ซึ่งจะทำตาม ลำดับต่อไปนี้

- จะทำการสร้าง Instance เฉพาะ Object บน Server จะเหมือนกับ ##Class(Aircraft) call.%New() ใน Cache' โดยจะทำการ Return ค่า Cache' Object ทาง Server สำหรับ ActiveX โดยค่านั้นๆจะเป็น OREF สำหรับ Object บน Server นี้
- มันจะทำการสร้าง Instance ใหม่ สำหรับ ActiveX object บน Client โดยจะติดต่อผ่านทาง Server Object โดยจะ Return ค่า handle (LPDISPATCH) ไปที่ ActiveX object
- จะรายงาน ความผิดพลาดเมื่อเราใช้ New()แล้วไม่สามารถสร้าง Object ได้

การบันทึก Object

การที่จะบันทึก Instance ของ Pesistent object จะใช้ Method %Save() (sys_Save ใน Visual Basic)
Method sys_Save เป็น Method ของ Cache' Object ไม่ใช่ของ Cache' Factory : ซึ่งจะประกาศดังนี้

```

Dim rc As String
Rc = aircraft.sys_save
Aircraft.sys_Close
Set aircraft = Nothing

```

เมื่อเรบันทึกเรียบร้อยแล้วทำการเรียก sys_Close สำหรับปิดObject ซึ่ง Object บน Server จะต้องปิดด้วย

การเปิด Object ที่มีอยู่แล้ว

Method Open() เป็น Method ของ Cache' Factory ซึ่งจะ load มาจาก ฐานข้อมูล

```
Dim aircraft As CacheObject.objectinstance
```

```
Set aircraft = factory.Open( "Aviationt.Aircraft", oid )
```

การเปิดนั้นจะต้องการ argumennt 2 ค่าด้วยกัน คือ

- ชื่อที่ใช้ในการเปิดของ Cache' Class
- Oid ที่ถูก Save ในฐานข้อมูล ซึ่งค่า ของ Oid นั้นจะถูกส่งไปยัง Cache' List ตามรูปแบบที่ได้สร้างไว้

Method Open() จะกระทำดังนี้คือ

- Load Object มาไว้ใน Memory ของ Serverซึ่งจะเหมือนกันการเรียก
##Class(Aviation.Aircraft).%Open(Oid) ใน Cache' โดยค่าของ OREF จะถูก Return กลับมาที่ Cache'
Object Server ของ Object ActiveX
- สร้าง Instant ใหม่สำหรับ Object ActiveX บน Computer ฝั่ง Client จะทำการ ติดต่อผ่าน Object Server
โดยจะ Return ค่า (LPDISPATCH) มาที่ Object ActiveX
- จะ Return ค่าผิดพลาดเมื่อไม่สามารถ ใช้ Method Open ได้

วิธีที่ 2 การติดต่อ โดยใช้ Cache' SQL

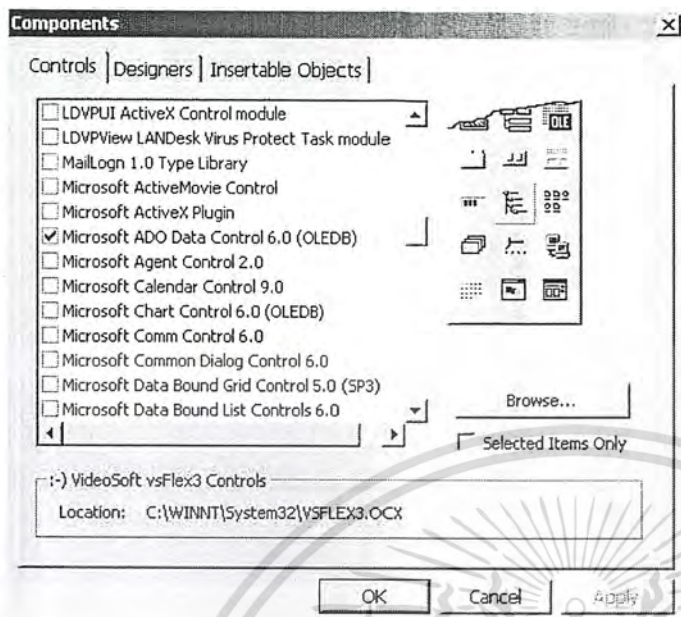
การที่เราจะติดต่อเราจะใช้ ODBC ในการติดต่อ Cache' Database โดยใน Visual Basic เราจะทำการส่ง Cache' SQL ลงไปใน Database จะส่งโดย ADO ในการส่งผ่าน ODBC

การส่ง SQL โดย ADO

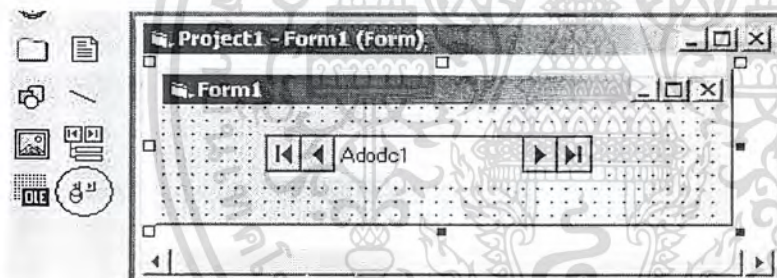
โดยจะแบ่งออกเป็น 2 รูปแบบ

รูปแบบที่ 1 ผ่านทาง Control ADO Data

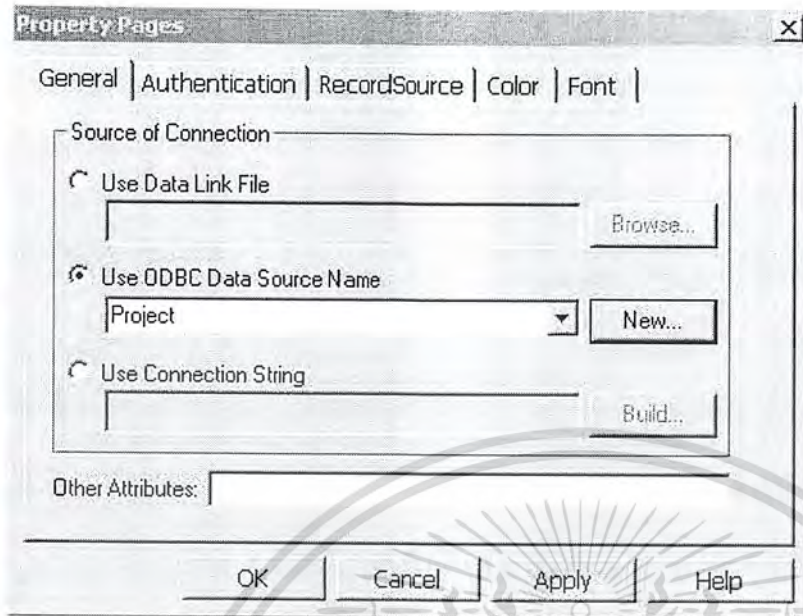
1. ไปที่ Menu bar / Project แล้วเลือก Component จะเห็น Component หลาย เราเลือก ที่ Microsoft ADO Data Control ดังรูป แล้วกด OK



2. จะได้ Component เพิ่ง ตรงที่วงกลม ไว้ แล้วทำการ Drag and Drop ที่ form จะได้ดังรูป



3. แล้ว Click ขวา ที่ ตัว Component ที่เราเลือก ทำการ เลือก Use ODBC Data Source Name เป็น Project ซึ่งเราได้ทำมาแล้ว ใน ภาคผนวก ข ถ้าไม่เข้าใจกลับไปดูใหม่ได้ แล้วก็เสร็จ



รูปแบบที่2 ผ่านทาง Command ADO

การใช้งานในส่วนของ Visual Basic

เมื่อเราได้ทำการสร้างการเชื่อมต่อโดยใช้ ODBC กันมาแล้วต่อไปเป็นการเรียกมาใช้งาน โดยใช้ VB

กำหนดตัวแปร ชื่อ Conn เป็น New ADODB แล้วเรียกการ Function Connection ดังข้างล่าง

```
conn As New ADODB.Connection
```

ในส่วนของการเรียกใช้ในตัว Function การทำงานต่างๆ โดยจะเป็นการเปิด การ Connection ชื่อ myDSN เป็นชื่อที่เราตั้งไว้เมื่อครั้งทำการ สร้างการเชื่อมต่อด้วย ODBC ส่วน "system" เป็นชื่อในการ login ส่วน "sys" เป็น Password ที่เราตั้งไว้

```
conn.Open myDSN, "_system", "sys"
```

เมื่อได้ทำการเชื่อมต่อเสร็จเรียบร้อยแล้วต่อไปก็เป็นเรื่องของ การเข้าถึงข้อมูลภายใน Cache' ซึ่งจะเป็นในด้านของการเข้าถึงโดยใช้ SQL ในการเข้าถึง โดยใน Visual Basic ได้มี Function ที่ใช้ให้เราเข้าถึง Database คือ Recordset รูปแบบการใช้งาน เราประกาศ ตัวแปร Rssc ดังนี้

Rssc As ADODB.Recordset

การเข้าถึง Schema ของ ตารางนั้นๆ โดยจะเลือกในส่วนของ Columns นี้เป็นเพียงตัวอย่างหนึ่งเท่านั้น

Set Rssc = conn.OpenSchema(adSchemaColumns)

สำหรับส่วนนี้เป็นส่วนของการ ส่งคำสั่ง SQL ส่งผ่าน ADO ซึ่งเราจะเขียนเองหรือใช้ Tool ที่ VB มีให้ก็ได้แล้วแต่ความต้องการของ Programmer แต่ในส่วนข้างล่างจะเป็นการส่งคำสั่ง SQL ผ่าน ADO ในการส่งผ่าน และ Execute ต่างๆ

Dim objcmd As New ADODB.Command

objcmd.ActiveConnection = conn

objcmd.CommandText = SQLText



ภาคผนวก ข

การ Configuration ODBC

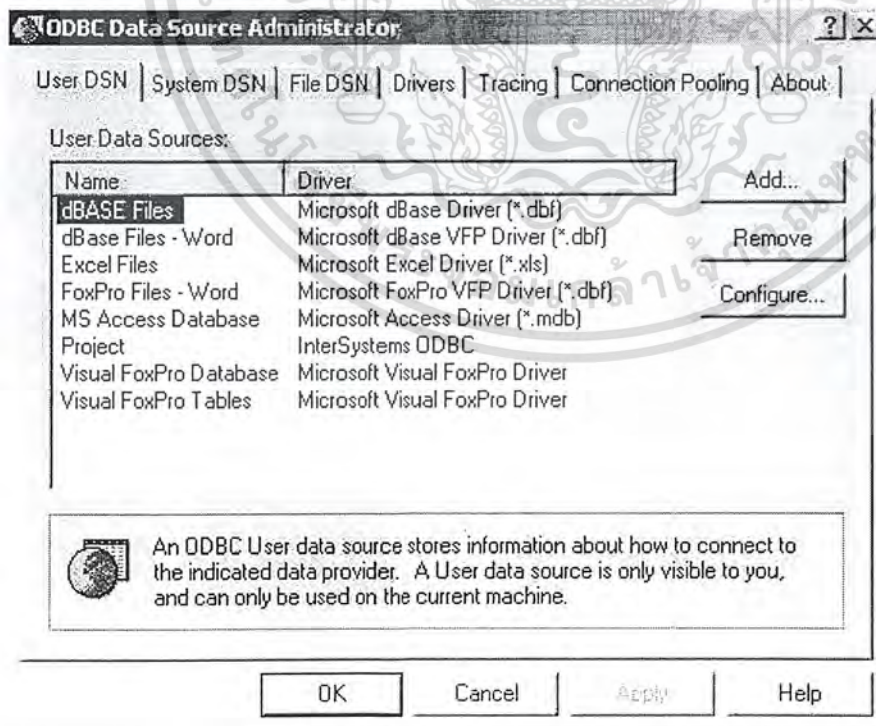
การติดต่อด้วย ODBC

สำหรับ Application ที่ใช้ SQL การเข้าถึง Database จากภายนอกนั้น เช่น Tool ที่ใช้สำหรับรายงานผลต่างๆ Application ที่ใช้ มีการเข้าถึง Database ที่เป็น Relation สำหรับ Cache' SQL Server นั้นจัดการเรื่องการเข้าถึงแบบ Relational จากภายนอกไว้อยู่แล้วโดยจะจัดการในเรื่องของ APIs ของ ODBC

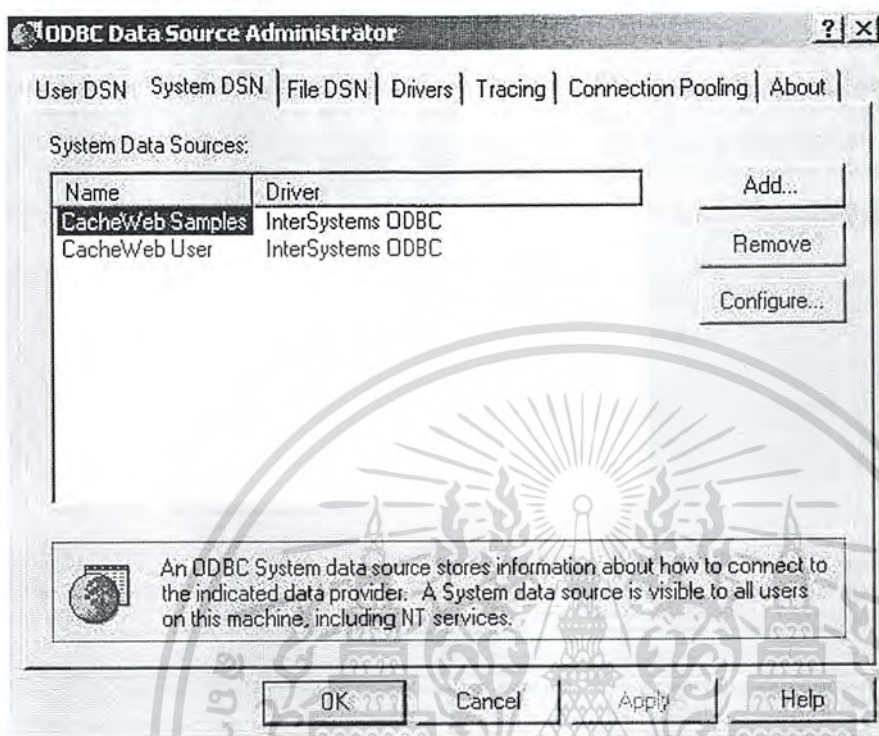
สำหรับ Database ที่ต้องการติดต่อ ผ่าน ODBC จำเป็นจะต้องมีเรื่องของ ระบบ Client ที่เรียกว่า ODBC driver ซึ่งจะมีจำเพาะในแต่ละผู้ผลิตจะมีให้ ซึ่งจะเป็น มาตรฐานของ ODBC สำหรับระบบ ปฏิบัติการ Window นั้นจัดการในเรื่องของ Driver สำหรับ Cache' ไว้อยู่แล้วเพียงแต่เราทำการติดตั้งเพิ่งเข้าไปเท่านั้น โดยมีขั้นตอนดังนี้

ขั้นตอนการติดตั้ง Driver ของ Cache' ผ่าน ODBC

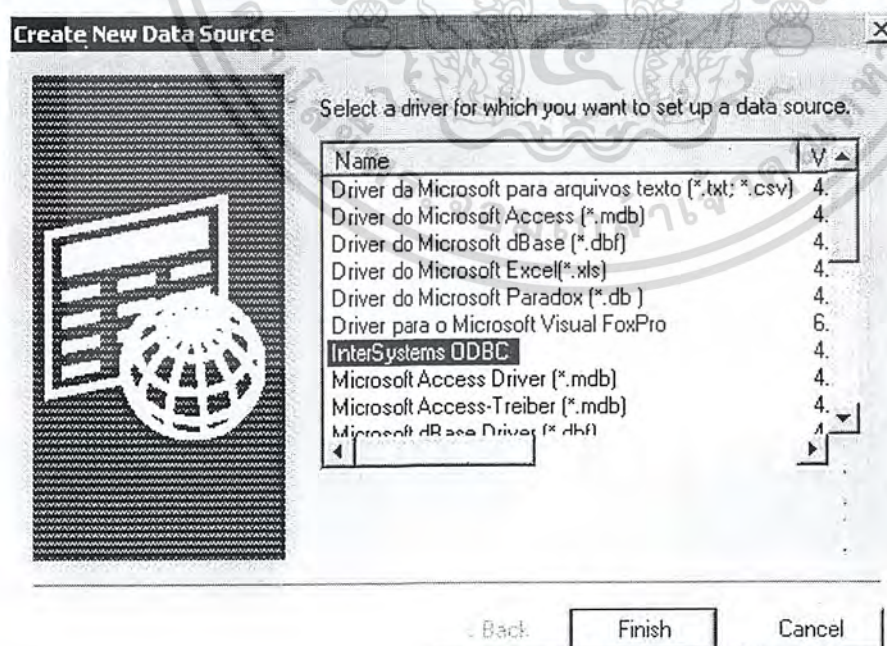
1. เข้าไปที่ **Control Panel** และ Double Click ที่ Icon ODBC ก็จะได้เห็น ODBC Data Source Administrator ปรากฏ ขึ้นมาบน Window



2. จากนั้นเลือก Tab ไปที่ System DSN กด button Add ทำการ เลือก Driver ของ InterSys ดังรูปแบบ



3. ทำการเลือก Driver ที่ InterSystem ODBC ดังรูปข้างล่าง แล้วกด Finish



1. จากนั้นทำการเลือก ในเฟรม Data Source ฟิลด์ Name โดยจะเป็นชื่อ ของ Connection ที่ใช้ในการติดต่อในส่วนเฟรม Connection นั้นเราใช้ เป็น loopback เป็น 127.0.0.1 ซึ่งจะให้ตัวเองเป็น Server เองส่วนถ้าจะให้ เครื่องอื่นที่ทำหน้าที่เป็น Database Server ก็ทำการใส่ IP เป็น IP ของ Server นั้นๆ ส่วน Port เป็น Port 1972 เลือก NameSpace เป็น NameSpace ที่เราต้องการจะติดต่อ ส่วน เฟรม Login จะเป็นชื่อที่ใช้ในการ Login จะตั้ง User Name เป็น _system และ Password เป็น sys จากนั้นทำการ Test Connection ดูว่าการ Connection นั้นสมบูรณ์หรือไม่

InterSystems Cach? ODBC Data Source Setup

Data Source

Name: Description:

Connection

☒ TCP/IP Host Name (IP Address): Port:

☐ Shared Memory

Cach? Configuration

Cach? Namespace:

☐ ODBC Log ☐ Static Cursors

☒ Disable Query Timeout

Login

User Name: Password:

OK Cancel Test Connection Help

5. เมื่อการ Connection Compleet แล้วจะขึ้นข้อความดังข้างล่างนี้

InterSystems Cach? ODBC Data Source Setup

Cach? ODBC DSN Configuration Summary:

Driver = C:\CacheSys\bin\CacheODBC.dll
Driver Version = 4.0.818.0

Data Source Name = Project
Protocol = TCP/IP
Host (IP Address) = 127.0.0.1
Port = 1972
Cach? Namespace = PROJECT

Connectivity test completed successfully!

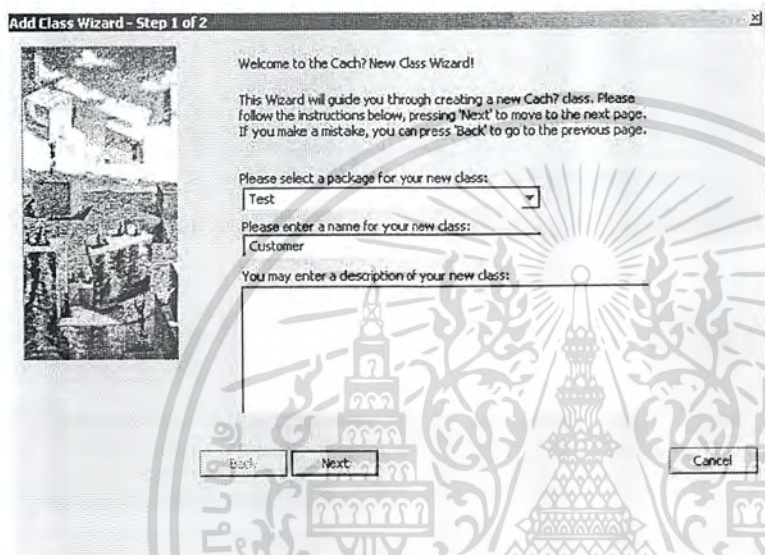
OK

ภาคผนวก ค

ขั้นตอนในการสร้าง Class ใน Cache'

ซึ่งเราได้ออกแบบไว้จากข้างบน มีขั้นตอนดังนี้

1. สร้าง Class Customer เป็น Class หลักก่อนดังที่ได้ออกแบบไว้ แล้ว กด Next



Add Class Wizard - Step 1 of 2

Welcome to the Cach7 New Class Wizard!

This Wizard will guide you through creating a new Cach7 class. Please follow the instructions below, pressing 'Next' to move to the next page. If you make a mistake, you can press 'Back' to go to the previous page.

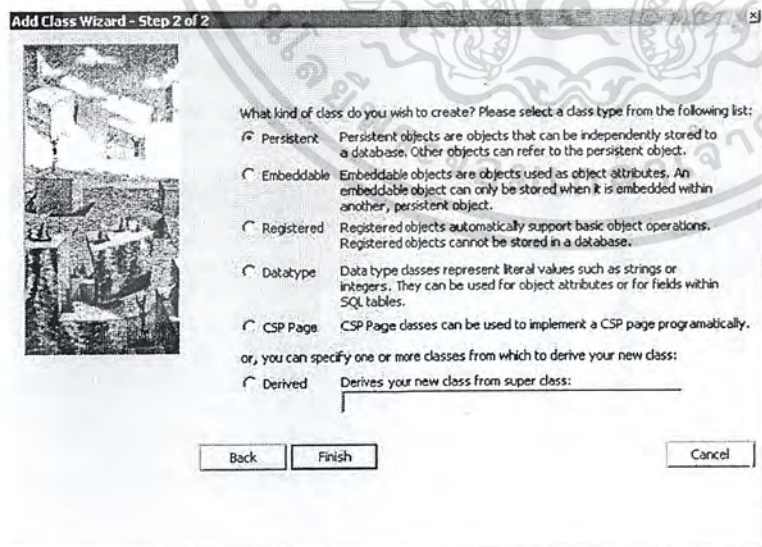
Please select a package for your new class:
Test

Please enter a name for your new class:
Customer

You may enter a description of your new class:

Back Next Cancel

2. หลังจาก กด Next เลือก Class Type เป็น Persistent แล้ว กด Finish



Add Class Wizard - Step 2 of 2

What kind of class do you wish to create? Please select a class type from the following list:

- ☒ Persistent Persistent objects are objects that can be independently stored to a database. Other objects can refer to the persistent object.
- ☐ Embeddable Embeddable objects are objects used as object attributes. An embeddable object can only be stored when it is embedded within another, persistent object.
- ☐ Registered Registered objects automatically support basic object operations. Registered objects cannot be stored in a database.
- ☐ Datatype Data type classes represent literal values such as strings or integers. They can be used for object attributes or for fields within SQL tables.
- ☐ CSP Page CSP Page classes can be used to implement a CSP page programmatically.
- ☐ Derived Derives your new class from super class:

or, you can specify one or more classes from which to derive your new class:

Back Finish Cancel

3. หลังจาก ที่สร้าง Class เสร็จเรียบร้อยแล้ว ก็ทำการกำหนด Property name ให้ แก่ Class Customer ซึ่งจะมี Property name ชื่อ Name มี Type เป็น String

Property: Test.Customer.NewProperty1

Basic | Description | Parameters | SQL

Property Name:
Name

Type:
%Library.String

Collection:
Single Value

Initial value expression:

Characteristics:

- ☐ Private
- ☐ Calculated
- ☐ Required
- ☐ Final
- ☐ Multi Dimensional
- ☐ Transient
- ☐ Indexed
- ☐ Unique (using a key)

Relationship

☐ Relationship

Cardinality:
Children

Inverse Property Name:
Parent

Inverse Property Cardinality:
Parent

OK Cancel

4. สร้าง Property Name ชื่อ SaleRep ซึ่งมี Type เป็น SalesRep โดยที่เราได้ทำการ สร้าง เป็น Class ไว้ ก่อนหน้านี้ แล้ว ซึ่งการสร้างเหมือน การสร้าง Class Customer ที่มี Type Class เป็น Persistent

Property: Test.Customer.NewProperty1

Basic | Description | Parameters | SQL

Property Name:
SalesRep

Type:
Test.SalesRep

Collection:
Single Value

Initial value expression:

Characteristics:

- ☐ Private
- ☐ Calculated
- ☐ Required
- ☐ Final
- ☐ Multi Dimensional
- ☐ Transient
- ☐ Indexed
- ☐ Unique (using a key)

Relationship

☐ Relationship

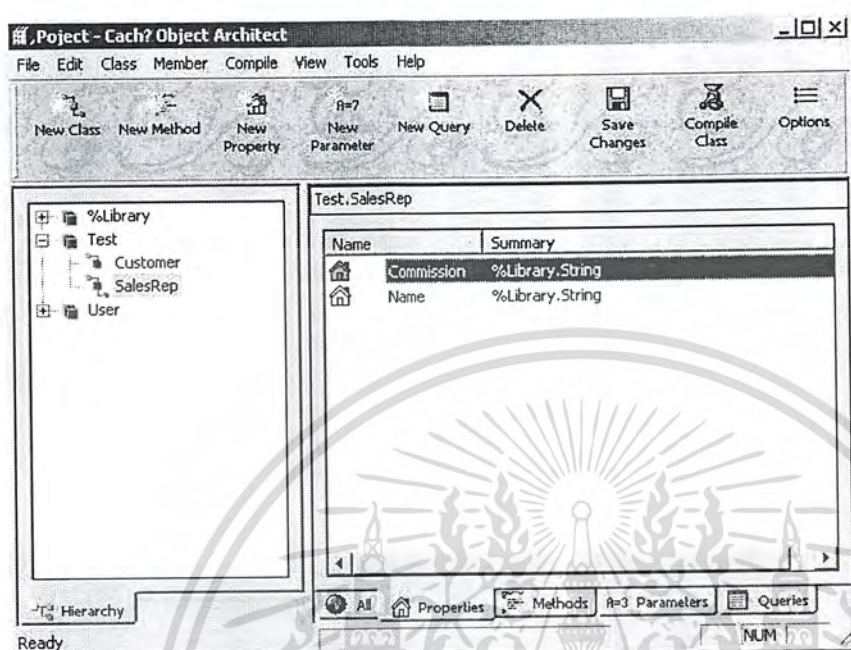
Cardinality:
Children

Inverse Property Name:
Parent

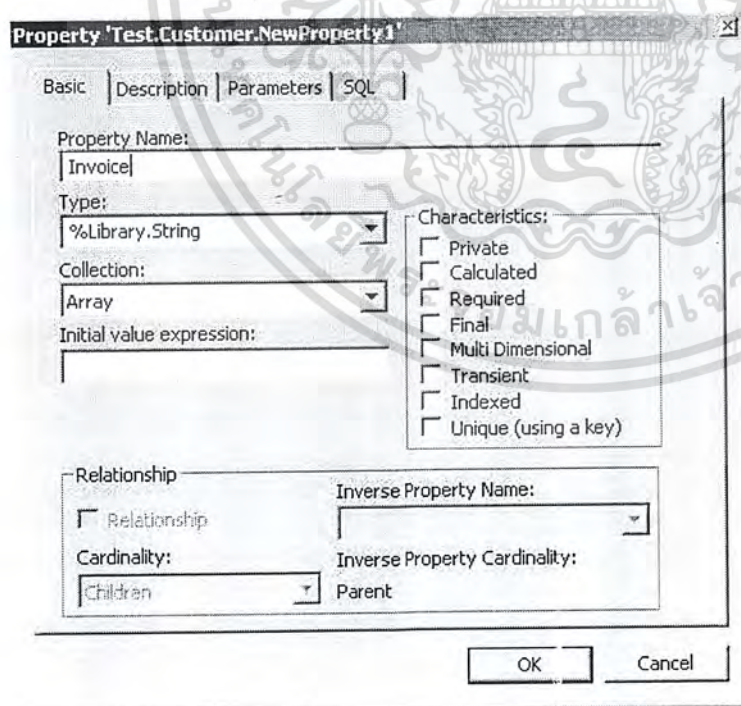
Inverse Property Cardinality:
Parent

OK Cancel

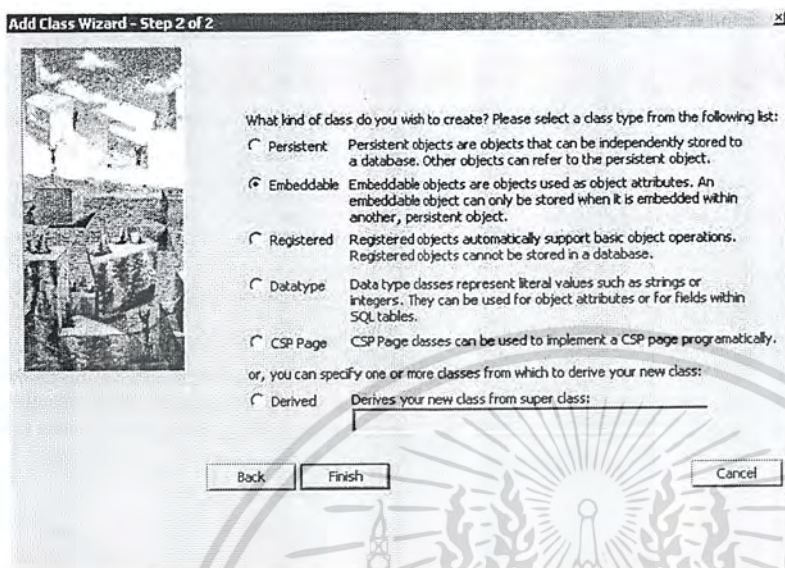
5. กำหนด Property Name มี Nam และ Commission ซึ่งเหมือนกัน การสร้างใน Class Customer



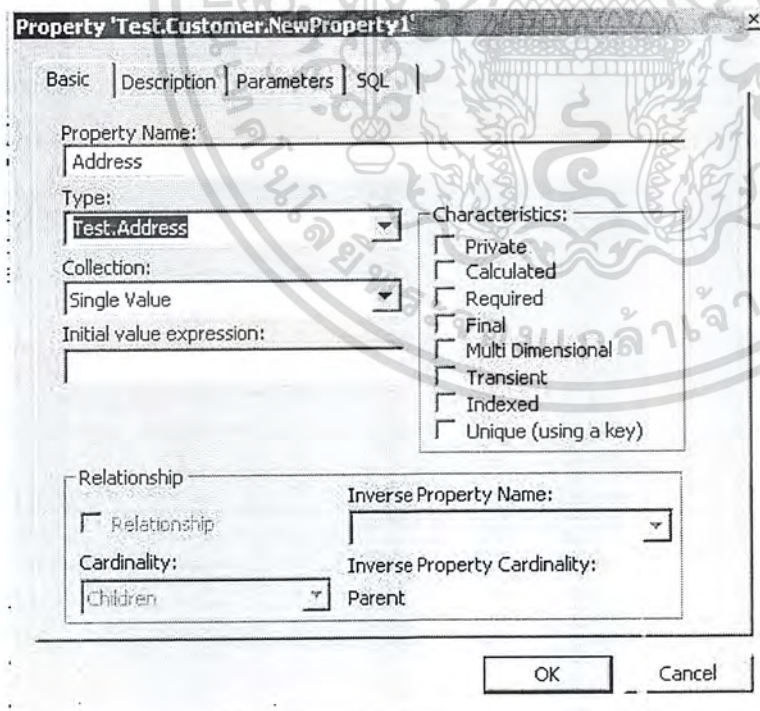
6. ต่อไปเราก็ Add Property Name ของ Class Customer ซึ่งให้มี Collection เป็น Array และ Type เป็น String



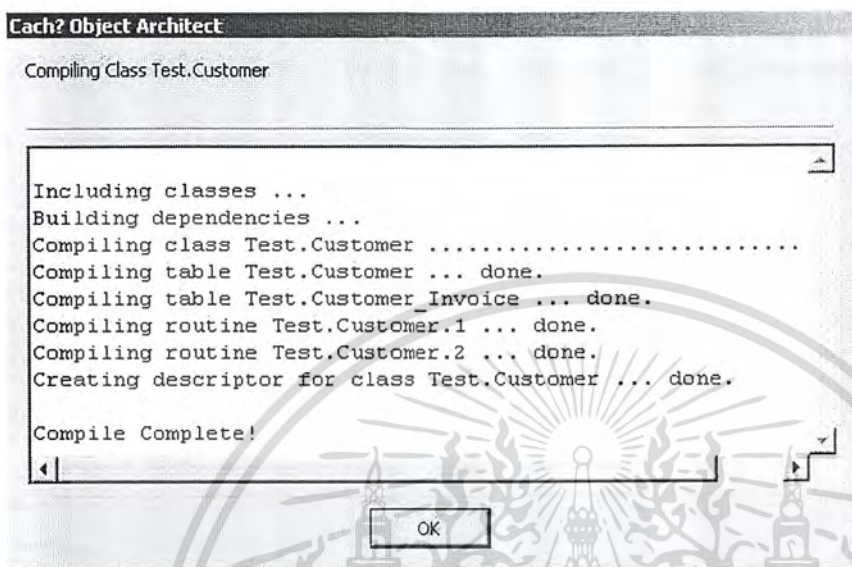
7. สร้าง Class Address โดยให้ Class Type เป็น Embeddable ดังรูปข้างล่าง



8. ภายใน Class กำหนด Type เป็น Address และกำหนด Property Name ให้มี Street และ City เหมือน ใน Class SalesRep



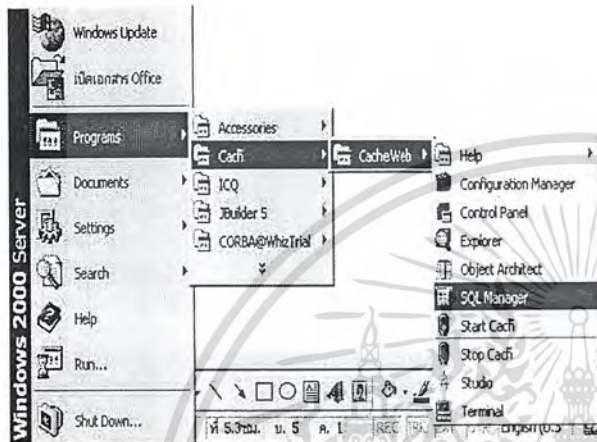
9. เมื่อได้ Class ทั้งหมดเรียบร้อยแล้วทำการ Compile Class ทั้งหมดจะได้ผลดังนี้ หรือถ้าไม่ได้ผลดังนี้ก็ให้ทำการ
ตรวจสอบ และ ทำการ Compile ใหม่อีกครั้ง



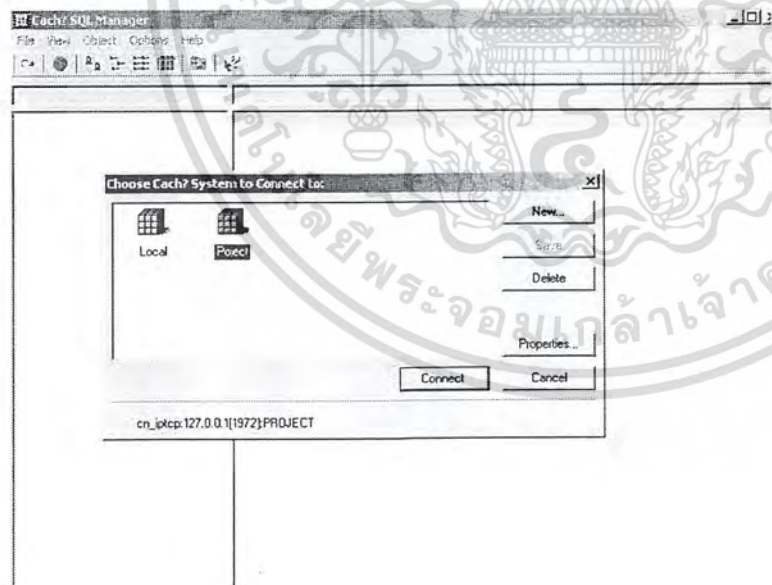
ภาคผนวก ง

ขั้นตอนในการเข้าใช้ Cache' SQL

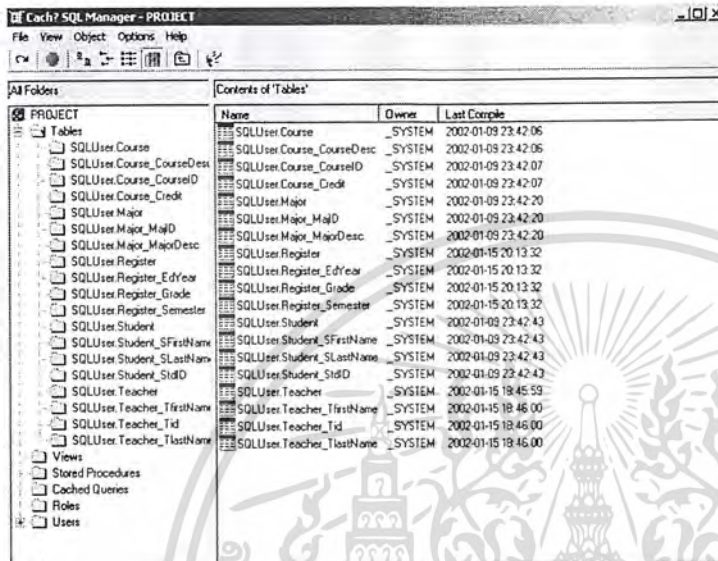
1. เราต้องเข้าไปเลือกที่ Program ที่ menu ดังรูปข้างล่าง โดยที่เลือกไปที่ SQL Manager



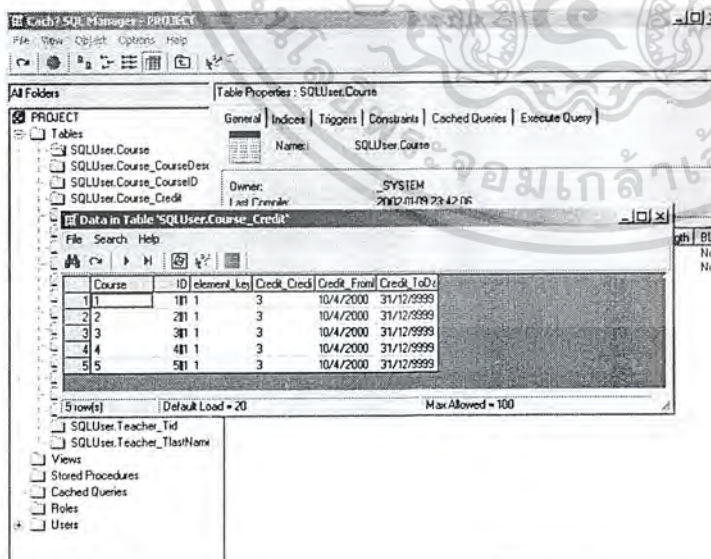
2. เลือก NameSpace ที่เราต้องการจะใช้ SQL ในการจัดการเรื่องต่างๆ ที่ Cache' ได้ติดตั้งไว้



3. เมื่อเลือก Namespace ที่เราต้องการแล้วก็จะได้ รายละเอียด ต่างๆ แสดงตารางที่เราได้สร้างไว้จาก การที่ เรา สร้าง Class ในตอนแรก ซึ่งตารางที่แสดงให้เห็นจะเป็น ในลักษณะที่มี Class ที่เป็น Persistent Class ที่ภายในเรา บรรจุไว้ด้วย Embedded Class โดยจะระบุ Type เป็น Array ในการจัดเก็บ



4. เมื่อเราต้องการดูข้อมูลภายในตารางที่เราเลือกขึ้นมาเราก็ Click ขวาที่ Folder ที่เราต้องการ ก็จะ ได้ข้อมูลที่ แสดงในตาราง นั้นๆ ซึ่งข้อมูลที่จัดเก็บในรูปแบบ ของ Array จะเป็น Folder ที่มี UnderScore ที่ตรงกลาง ดังเห็น ได้จากรูปข้างล่าง ส่วน Folder ที่ไม่มี UnderScore



ภาคผนวก จ

การติดตั้ง Cache` 4.0

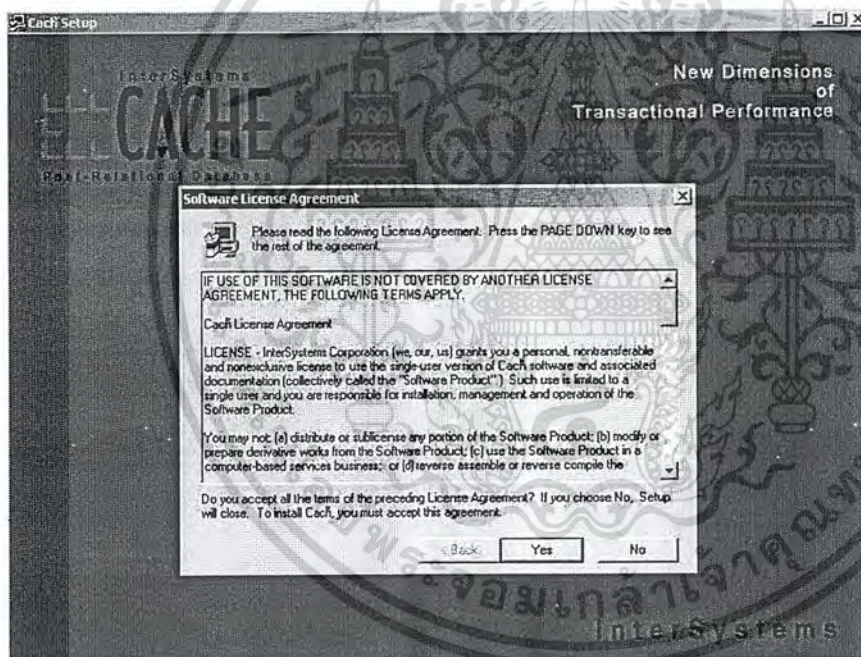
Minimum Hardware ที่ต้องการ

- CPU 486 50mhz Cache` จะทำงานได้อย่างมีประสิทธิภาพถ้าให้ Pentium Processor
- Memory 32Mb

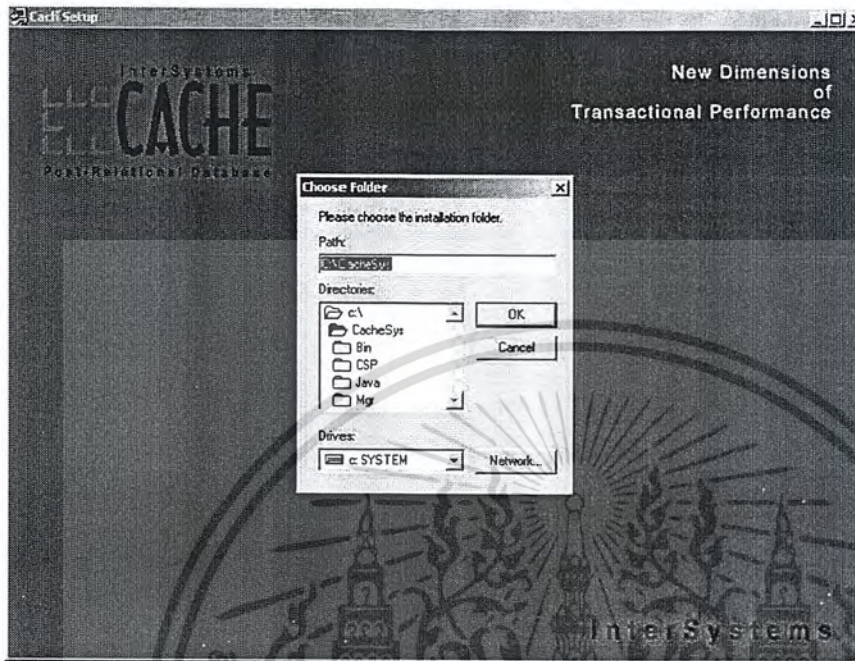
Operating System

- Windows 9X/NT/2000 หรือสามารถ Run บน Unix ,OpenVMS,Red Hat Linux 6.2

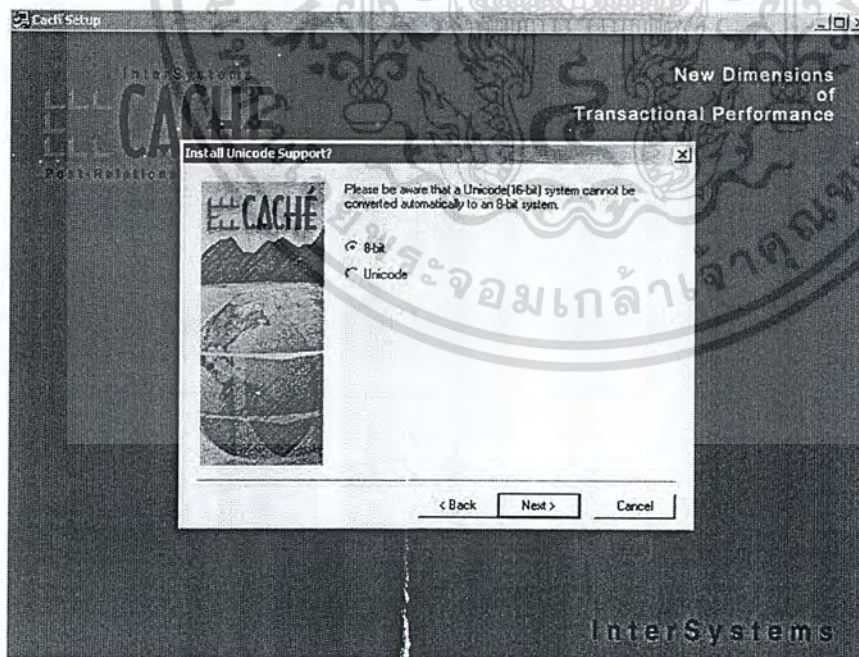
1. เมื่อเราเอาแผ่น CD เข้าแล้วเข้าไป ที่ Directory window แล้วเรียก File Setup.exe จะได้ดังรูป แล้วกด Yes

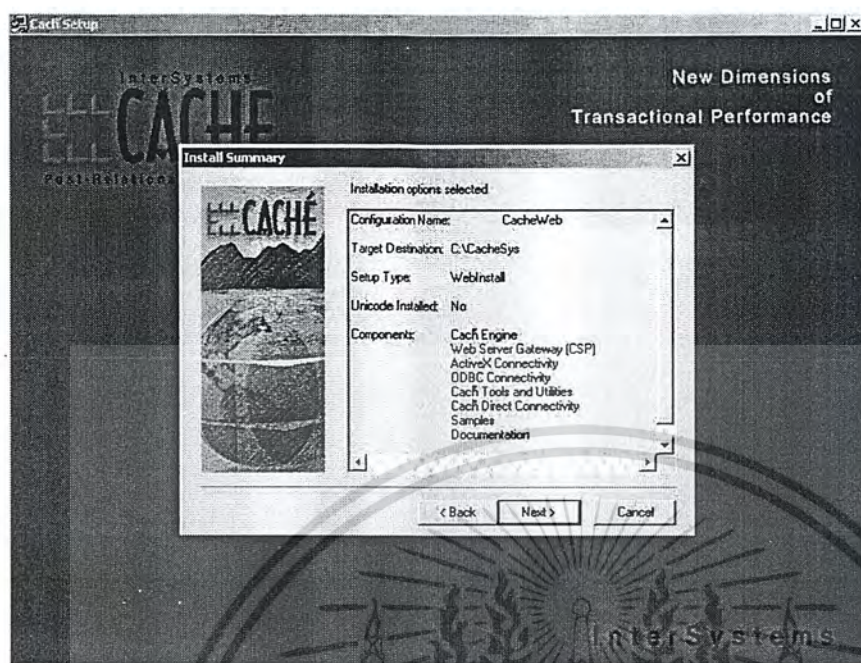


2. ให้เรา เลือก Directory ที่ต้องการเก็บ Directory Cache ไว้ เราสามารถ Set ตาม Default ก็ได้

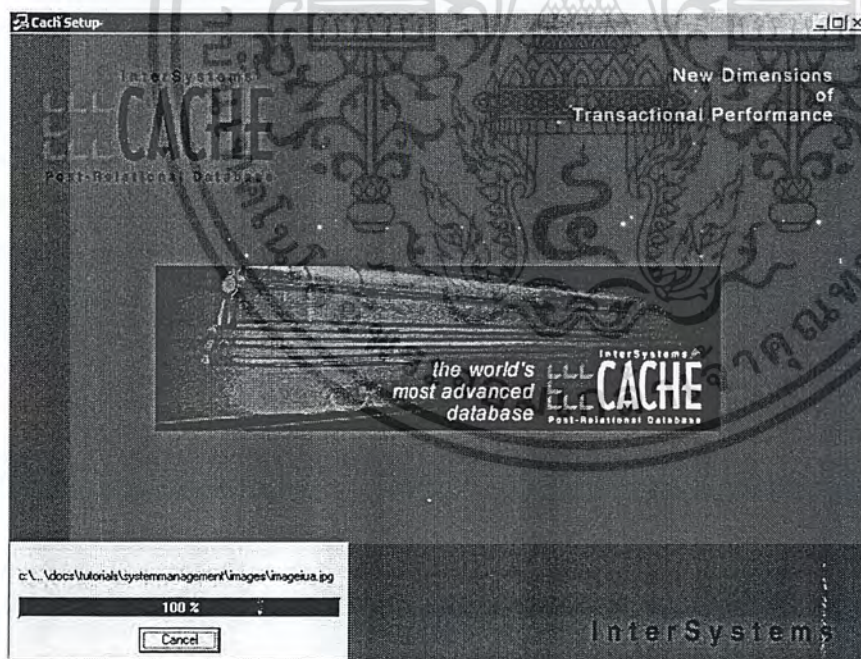


3. เลือก Unicode ให้เราเลือก ตาม Default แล้ว กด Next





4. จากนั้นก็รอจนกว่าจะครบ 100 %



5. กด Finish เป็นอันจบการ ติดตั้ง

