

ฐานข้อมูลเชิงวัตถุกับการประยุกต์เชิงเวลา 1  
OBJECT-ORIENTED DATABASE IN TEMPORAL APPLICATION 1



เลขหนังสือ.....  
เลขทะเบียน..... 42818  
วัน, เดือน, ปี 10 ส.ย. 2545

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

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ฐานข้อมูลเชิงวัตถุกับการประยุกต์เชิงเวลา

OBJECT-ORIENTED DATABASE IN TEMPORAL APPLICATION I

ผู้จัดทำ

1. นาย ประเมข หทัยรักษัธรรม รหัสประจำตัว 40010425

1. นาย ไพฑูรย์ กุณโฮง รหัสประจำตัว 40010534





อาจารย์ที่ปรึกษา

(รศ.ดร. สุกมิตร จิตตะยโสธร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ฐานข้อมูลเชิงวัตถุกับการประยุกต์เชิงเวลา 1

นายปรเมศ หทัยรักษัธรรม 40010425

นายไพฑูรย์ ภูณ โสภ 40010534

รศ.ดร. ศุภมิตร จิตตะย โสธร อาจารย์ที่ปรึกษา

ปีการศึกษา 2543

## บทคัดย่อ

ในปัจจุบันเทคโนโลยีทางด้านฐานข้อมูลได้พัฒนาไปมาก ทั้งในแง่ของประสิทธิภาพการทำงานที่เร็วขึ้น ทั้งในแง่ของความสามารถที่มากขึ้น รวมทั้งแนวคิดต่างๆ เกี่ยวกับฐานข้อมูล โดยปกติการเก็บข้อมูลจะเก็บเฉพาะความจริง (Fact) ก็คือข้อมูลที่เป็นจริงในเวลาปัจจุบันเท่านั้น โดยข้อมูลเดิมที่ถูกเปลี่ยนแปลงหรือถูกลบก็จะหายไป และจะไม่เก็บข้อมูลที่จะเป็นจริงในอนาคตแต่ยังไม่เป็นจริงในปัจจุบัน ฐานข้อมูลเชิงเวลา (Temporal Database) สามารถจัดการกับสิ่งเหล่านี้ได้โดยการเพิ่มส่วนของเวลาไปกับข้อมูล ทำให้ฐานข้อมูลเชิงเวลาสามารถเก็บข้อมูลทั้งในอดีต ปัจจุบัน และอนาคตได้

อย่างไรก็ตาม ฐานข้อมูลแบบรีเลชันแนลก็ยังมีข้อเสียในแง่ของการแสดงความสัมพันธ์ของข้อมูล เพราะแสดงได้ในแบบของตารางเท่านั้น ไม่สามารถแสดงข้อมูลแบบกลุ่มซ้ำในตารางๆเดียวได้ ทำให้ต้องมีการแยกเป็นตารางย่อยๆ อีกทั้งถ้าระบบมีขนาดใหญ่ การวิเคราะห์และการออกแบบฐานข้อมูล จะมีความซับซ้อนและยากมาก จึงมีความพยายามใช้ฐานข้อมูลเชิงวัตถุแทน เพื่อแก้ไขปัญหาดังกล่าว

ปัญญานิพนธ์ฉบับนี้ได้้นำแนวคิดเรื่องฐานข้อมูลเชิงวัตถุมาประยุกต์ใช้กับฐานข้อมูลเชิงเวลา เพื่อศึกษาถึงข้อดีและข้อเสียต่างๆ โดยพยายามนำเอาคุณสมบัติและข้อได้เปรียบของฐานข้อมูลเชิงวัตถุมาใช้ในการแก้ไขปัญหาแทนการใช้ฐานข้อมูลแบบรีเลชันแนลซึ่งพบว่ายังมีข้อเสียอยู่

## Object-Oriented Database in Temporal Application 1

Poramate Hatairaktham

Paitoon Gunhong

Assoc. Prof. Dr. Suphamit Chitayasothorn

Advisor

### ABSTRACT

Recently, Database Technology was developed rapidly both performance and ability, include the several concept of database. Usually, The keeping of data will especially keep fact which is only fact in the present. At the beginning data is modified or deleted, as a result, It will be lost and it will not keep the data which be fact in the future but not be fact in the present. The Temporal Database can manage with that thing by adding the aspect of time with the data, It will be able to keep the data in the past, in the present and in the future.

However, The relational database still have disadvantage in sense of the relation of data because it only represent table form. It can't represent repeated group in the only table which make it split the table be minor table. And if it have big system, Analysis and design database will have duplicate and several difficult. It will be attempt to use the object-oriented database replace the relational database for solving problem.

This thesis lead the conceptual topic of the object-oriented database which be led to apply with the temporal database for study both the advantage and disadvantage which try lead the property and advantage use in solving problem replace the relational database which have many disadvantage

## กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และความร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นผู้มีส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ ท่านอาจารย์ สุภมิตร จิตตะยโสธร อาจารย์ที่ปรึกษาปริญญาบัตร ที่ได้คอยให้คอยคำปรึกษาที่ดี รวมไปถึงการแนะนำในสิ่งต่างๆ ทั้งเรื่องเรียนและเรื่องต่างๆ โดยเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างสูง รวมไปถึงท่านอาจารย์ทุก ๆ ท่านที่ประสิทธิ์ประสาทวิชาความรู้ต่าง ๆ ที่ได้คอยเฝ้าอบรมมาตลอดสี่ปีที่ผ่านมา

และบุคคลที่สำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูอบรม และอุปการะข้าพเจ้ามาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุกๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอ กราบขอบพระคุณมา ณ ที่นี้

สุดท้ายขอขอบคุณความดีจากปริญญาบัตร ฉบับนี้ให้แก่ คุณ มารดา บิดา และครูอาจารย์ผู้มีพระคุณทุกท่าน

ประเมษ ทวีรัศม์ธรรม  
ไพฑูรย์ กุณโฮง

# สารบัญ

|   | หน้าที่ |
|---|---------|
| บทคัดย่อภาษาไทย   | I       |
| บทคัดย่อภาษาอังกฤษ  | II      |
| กิตติกรรมประกาศ   | III     |
| สารบัญ  | IV      |
| สารบัญตาราง   | X       |
| สารบัญภาพ   | XI      |
| บทที่ 1 บทนำ  | 1       |
| 1.1 ความสำคัญและที่มา                                     | 1       |
| 1.2 วัตถุประสงค์ของงานวิจัย                               | 1       |
| 1.3 ขอบเขตของงานวิจัย                                     | 1       |
| 1.4 วิธีการดำเนินงาน                                      | 2       |
| บทที่ 2 ฐานข้อมูลเชิงเวลา                                 | 3       |
| 2.1 ฐานข้อมูลเชิงเวลา                                     | 3       |
| 2.1.1 วาลิดไทม์   | 3       |
| 2.1.2 ทรานแซกชันไทม์                                      | 3       |
| 2.1.3 ยูสเซอร์สไทม์ไทม์                                   | 4       |
| 2.2 ความซ้ำซ้อนที่เกิดขึ้นในฐานข้อมูลเชิงเวลา             | 4       |
| 2.2.1 นอน-ซีเควนซ์ดัพพลิเคต                               | 5       |
| 2.2.2 แวลูอีควิวไลน์ดัพพลิเคต                             | 5       |
| 2.2.3 เคอเรนตฺดัพพลิเคต                                   | 5       |
| 2.2.4 ซีเควนซ์ดัพพลิเคต                                   | 5       |
| 2.3 ประเภทของฐานข้อมูลเชิงเวลา                            | 6       |
| 2.3.1 ฐานข้อมูลวาลิดไทม์                                  | 6       |
| 2.3.2 ฐานข้อมูลทรานแซกชันไทม์ (Transaction-Time Database) | 6       |
| 2.3.3 ฐานข้อมูลไบเทม โพรอล                                | 7       |
| 2.4 ตัวอย่างแอปพลิเคชันที่เกี่ยวข้องกับข้อมูลเชิงเวลา     | 7       |
| 2.4.1 การเงิน   | 7       |
| 2.4.2 การประกันภัย  | 7       |
| 2.4.3 ระบบการจอง  | 7       |
| 2.4.4 ระบบการจัดการข้อมูลทางการแพทย์                      | 7       |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|  | หน้าที่ |
|--|---------|
| 2.4.5 ระบบการสนับสนุนการตัดสินใจ   | 7       |
| 2.4.6 ระบบการผลิต  | 8       |
| 2.4.7 การวิจัยทางวิทยาศาสตร์   | 8       |
| 2.5 คำถามเชิงเวลาในฐานะข้อมูลเชิงเวลา  | 8       |
| 2.5.1 การตอบคำถามแบบทั่วไป   | 8       |
| 2.5.2 การตอบคำถามแบบปัจจุบัน   | 8       |
| 2.5.3 การตอบคำถามแบบซีเควนซ์   | 9       |
| 2.5.4 การตอบคำถามแบบนอนซีเควนซ์  | 9       |
| 2.6 การตอบคำถามเชิงเวลาที่มีการ JOIN   | 10      |
| 2.6.1 การตอบคำถามแบบทั่วไปโดยใช้การ JOIN   | 10      |
| 2.6.2 การตอบคำถามแบบปัจจุบันโดยใช้การ JOIN   | 10      |
| 2.6.3 การตอบคำถามแบบนอนซีเควนซ์โดยใช้การ JOIN  | 11      |
| 2.6.4 การตอบคำถามแบบซีเควนซ์โดยใช้การ JOIN   | 11      |
| 2.6.4.1 ช่วงเวลาของตาราง L1 อยู่ในช่วงเวลาของตาราง L2<br>แสดงดังรูปที่ 2-12  | 11      |
| 2.6.4.2 เวลาเริ่มต้นของตาราง L1 มากกว่าเวลาเริ่มต้นของตาราง L2<br>แต่น้อยกว่าเวลาสิ้นสุดของตาราง L2 และเวลาสิ้นสุดของตาราง<br>L1 มากกว่าเวลาสิ้นสุดของตาราง L2 แสดงดังรูปที่ 2-2 | 12      |
| 2.6.4.3 เวลาเริ่มต้นของตาราง L2 มากกว่าเวลาเริ่มต้นของตาราง L1<br>แต่น้อยกว่าเวลาสิ้นสุดของตาราง L1 และเวลาสิ้นสุดของตาราง<br>L2 มากกว่าเวลาสิ้นสุดของตาราง L1 แสดงดังรูปที่ 2-3 | 12      |
| 2.6.4.4 ช่วงเวลาของตาราง L2 อยู่ในช่วงเวลาของตาราง L1 แสดงดังรูปที่ 4-4  | 12      |
| 2.7 การเปลี่ยนแปลงข้อมูลในฐานะข้อมูลเชิงเวลา   | 14      |
| 2.7.1 การเปลี่ยนแปลงแบบปัจจุบัน  | 15      |
| 2.7.1.1 การลบแบบปัจจุบัน   | 15      |
| 2.7.1.2 การแก้ไขแบบปัจจุบัน  | 16      |
| 2.7.2 การเปลี่ยนแปลงแบบซีเควนซ์  | 17      |
| 2.7.2.1 การลบแบบซีเควนซ์   | 17      |
| 2.7.2.2 การแก้ไขแบบซีเควนซ์  | 19      |
| 2.7.3 การเปลี่ยนแปลงแบบนอนซีเควนซ์   | 21      |
| บทที่ 3 แนวความคิดเชิงวัตถุ  | 22      |
| 3.1 คุณสมบัติของแนวความคิดแบบเชิงวัตถุ   | 22      |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|   | หน้าที่ |
|---|---------|
| 3.2 คุณสมบัติของ Object-Oriented                                      | 23      |
| 3.2.1 Encapsulation/Information hiding                                | 23      |
| 3.2.2 Inheritance   | 23      |
| 3.2.3 Polymorphism  | 23      |
| 3.2.4 Abstraction   | 24      |
| 3.2.5 Persistence   | 24      |
| 3.2.6 Concurrency   | 24      |
| 3.3 หลักการพื้นฐานของ Object-Oriented                                 | 24      |
| 3.4 ความสัมพันธ์ระหว่าง Object  | 26      |
| 3.4.1 Association   | 26      |
| 3.4.2 Aggregation   | 26      |
| 3.4.3 Generalization  | 26      |
| 3.4.4 Depends On  | 26      |
| 3.5 ข้อดีของภาษาทางด้าน Object-Oriented                               | 27      |
| บทที่ 4 ฐานข้อมูลเชิงวัตถุ  | 28      |
| 4.1 ระบบจัดการฐานข้อมูลเชิงวัตถุ                                      | 28      |
| 4.1.1 Object Identity   | 28      |
| 4.1.2 Complex Value   | 28      |
| 4.1.3 Method  | 29      |
| 4.1.4 Class   | 29      |
| 4.1.5 การสืบทอดคุณสมบัติ  | 29      |
| 4.1.6 Polymorphism  | 30      |
| 4.2 ประโยชน์จากฐานข้อมูลเชิงวัตถุ                                     | 31      |
| บทที่ 5 ฐานข้อมูลเชิงวัตถุกับการประยุกต์เชิงเวลา                      | 34      |
| 5.1 ปัญหาต่างในการนำฐานข้อมูลเชิงสัมพันธ์ประยุกต์กับฐานข้อมูลเชิงเวลา | 34      |
| 5.1.1 ปัญหาเรื่องของความซ้ำซ้อนของข้อมูล                              | 34      |
| 5.1.2 ปัญหาเรื่องของ Primary Key                                      | 35      |
| 5.2 การใช้ฐานข้อมูลเชิงวัตถุกับการประยุกต์เชิงเวลา                    | 35      |
| บทที่ 6 การนำฐานข้อมูลเชิงวัตถุและแนวคิดของฐานข้อมูลเชิงเวลา          | 39      |
| มาประยุกต์ใช้กับระบบงานทะเบียนนักศึกษา                                |         |
| 6.1 ระบบงานทะเบียนนักศึกษา  | 39      |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|   | หน้าที่   |
|---|-----------|
| 6.2 คลาสไลค์แอมของระบบงานทะเบียนนักศึกษา  | 39        |
| 6.3 ปัญหาที่พบในระบบทะเบียนนักศึกษาเดิมและวิธีแก้ปัญหาระบบทะเบียน<br>นักศึกษาเดิม | 44        |
| 6.4 ความสามารถของระบบทะเบียนนักศึกษาที่พัฒนา                                      | 45        |
| 6.5 เทคนิคที่ใช้ในการแก้ไขปัญหาระบบทะเบียนนักศึกษา                                | 45        |
| 6.6 ขั้นตอนการสร้างTempString   | 45        |
| 6.7 ขั้นตอนการนำTempStringไปกำหนดให้กับแอททริบิวต์                                | 49        |
| 6.8 รายละเอียดของเมธอดกลาง  | 50        |
| <b>บทที่ 7 การพัฒนาโปรแกรม</b>  | <b>66</b> |
| 7.1 สภาพแวดล้อมของระบบ  | 66        |
| 7.1.1 วาลิดไทม์   | 66        |
| 7.2 การติดต่อระหว่าง Cache' และ ActiveX   | 67        |
| 7.3 Cache' Object และ Visual Basic  | 68        |
| 7.3.1 การปรับแต่ง Visual Basic เพื่อใช้กับ Cache' Object                          | 68        |
| 7.3.1.1 การเลือก Project Reference ใน Visual Basic                                | 68        |
| 7.4 การจัดการกับ Cache' Objects ใน Visual Basic                                   | 69        |
| 7.4.1 การติดต่อกับ Server   | 69        |
| 7.4.2 การสร้าง Object Instance ใหม่   | 70        |
| 7.4.3 การบันทึกค่าของวัตถุ  | 70        |
| 7.4.4 การเปิดวัตถุที่มีอยู่แล้วขึ้นมา   | 70        |
| 7.4.5 การใช้วัตถุใน Visual Basic  | 71        |
| 7.4.6 การเรียก Query ใน Visual Basic  | 71        |
| 7.4.7 การตรวจสอบข้อผิดพลาดใน Visual Basic   | 72        |
| <b>บทที่ 8 บทสรุป</b>   | <b>73</b> |
| 8.1 ข้อเสนอแนะ ปัญหาที่พบ และ แนวทางการศึกษาต่อ                                   | 73        |
| 8.2 สรุป  | 74        |
| <b>บรรณานุกรม</b>   | <b>76</b> |
| <b>ภาคผนวก ก. UML (Unified Modeling Language)</b>                                 | <b>77</b> |
| ก.1 Unified Modeling Language   | 77        |
| ก.2 ส่วนประกอบของ UML   | 77        |
| ก.2.1 view  | 78        |
| ก.2.1.1 Use-case View   | 78        |
| ก.2.1.2 Logical View  | 78        |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|  | หน้าที่ |
|--|---------|
| ก.2.1.3 Component View   | 78      |
| ก.2.1.4 Deployment View  | 78      |
| ก.2.1.5 Process View   | 78      |
| ก.2.2 Diagram  | 79      |
| ก.2.2.1 Use Case Diagram   | 79      |
| ก.2.2.2 Class Diagram  | 80      |
| ก.2.2.3 Object Diagram   | 81      |
| ก.2.2.4 Collaboration Diagram                                    | 82      |
| ก.2.2.5 Sequence Diagram   | 82      |
| ก.2.2.6 State Diagram  | 83      |
| ก.2.2.7 Activity diagram   | 84      |
| ก.2.2.8 Component diagram  | 84      |
| ก.2.2.9 Deployment diagram                                       | 84      |
| ก.2.3 Model element  | 85      |
| ภาคผนวก ข. ระบบจัดการฐานข้อมูลเชิงวัตถุคาเช่ (Cache')            | 87      |
| ข.1 ขั้นตอนการติดตั้งคาเช่                                       | 87      |
| ข.2 ขั้นตอนการสร้างเนมสเปส (Name Space)                          | 90      |
| ข.3 ขั้นตอนของการนำเนมสเปสที่ได้กำหนดไว้แล้วมาใช้งาน             | 95      |
| ข.4 ขั้นตอนอิมพอร์ตซีดีแอลไฟล์                                   | 96      |
| ข.5 ขั้นตอนของการเอ็กพอร์ตคลาสเป็นซีดีแอลไฟล์                    | 99      |
| ข.6 ขั้นตอนการคอมไพล์คลาสในคาเช่                                 | 100     |
| ข.7 ขั้นตอนการสร้างคลาส  | 102     |
| ข.8 ขั้นตอนการสร้างProperty                                      | 103     |
| ข.9 ขั้นตอนการสร้างเมธอด   | 104     |
| ภาคผนวก ค. ระบบงานทะเบียนนักศึกษา                                | 107     |
| ค.1 ขั้นตอนการใช้งานระบบทะเบียนนักศึกษา                          | 107     |
| ค.2 ขั้นตอนการเพิ่มข้อมูลแก้ไขข้อมูลเกี่ยวกับคณะ ภาควิชาสาขาวิชา | 109     |
| ค.2.1 สำหรับคณะ  | 109     |
| ค.2.2 สำหรับภาควิชา  | 109     |
| ค.2.3 สำหรับสาขาวิชา   | 110     |
| ค.3 ขั้นตอนการเพิ่ม/ลบ/แก้ไขวิชา                                 | 111     |
| ค.4 ขั้นตอนของการเพิ่ม/ลบ/แก้ไขหลักสูตร                          | 113     |
| ค.5 ขั้นตอนของการเพิ่ม/ลบรายวิชาในหลักสูตร                       | 117     |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|  | หน้าที่ |
|--|---------|
| ค.6 ขั้นตอนของการเพิ่ม/แก้ไข/ลบนักศึกษา              | 119     |
| ค.7 ขั้นตอนของการใช้งานข้อมูลทะเบียนนักศึกษาเชิงเวลา | 121     |
| ค.7.1 ขั้นตอนการใช้งานข้อมูลนักศึกษาเชิงเวลา         | 121     |
| ค.7.2 ขั้นตอนการใช้งานข้อมูลวิชาเชิงเวลา             | 129     |
| ค.8 กิจกรรมการลงทะเบียน                              | 137     |
| ค.9 กิจกรรมผลการเรียน                                | 139     |
| ค.10 การพิจารณาจบ                                    | 141     |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

|   | หน้าที่ |
|---|---------|
| ตารางที่ 2-1 แสดงตัวอย่างของการเก็บValid timeลงในฐานข้อมูล            | 3       |
| ตารางที่ 2-2 แสดงตัวอย่างของการเก็บ transaction time ลงในฐานข้อมูล    | 3       |
| ตารางที่ 2-3 ดัดทอนมาจาก ตาราง NICUStatus                             | 4       |
| ตารางที่ 2-4 ความสัมพันธ์ของความซ้ำซ้อนแบบต่างๆ                       | 5       |
| ตารางที่ 2-5 ตัวอย่างของฐานข้อมูลลาคิดใหม่                            | 6       |
| ตารางที่ 2-6 ตัวอย่างของฐานข้อมูลทรานแซคชั่นใหม่                      | 6       |
| ตารางที่ 2-7 ตารางวิว   | 8       |
| ตารางที่ 2-8 คำตอบของตัวอย่างการตอบคำถามแบบปัจจุบัน                   | 9       |
| ตารางที่ 2-9 คำตอบของตัวอย่างการตอบคำถามแบบซีเควนซ์                   | 9       |
| ตารางที่ 2-10 คำตอบของตัวอย่างการตอบคำถามแบบนอนซีเควนซ์               | 10      |
| ตารางที่ 2-11 คำตอบของตัวอย่างการตอบคำถามแบบนอนซีเควนซ์โดยใช้การ JOIN | 11      |
| ตารางที่ 2-12 คำตอบของตัวอย่างการตอบคำถามแบบซีเควนซ์โดยใช้การ JOIN    | 14      |
| ตารางที่ 2-13 ตารางเพศัว  | 15      |
| ตารางที่ 2-14 ผลของตัวอย่างการลบแบบปัจจุบัน                           | 16      |
| ตารางที่ 5-1 แสดงความซ้ำซ้อนที่เกิดจากการใช้ฐานข้อมูลแบบสัมพันธ์      | 34      |
| ตารางที่ 5-2 แสดงความซ้ำซ้อนของข้อมูลอื่นที่ไม่ได้เปลี่ยนแปลงตามเวลา  | 34      |
| ตารางที่ 6-1 แสดงรายละเอียดของคลาสนักศึกษา                            | 41      |
| ตารางที่ 6-2 แสดงรายละเอียดของคลาสวิชา                                | 42      |
| ตารางที่ 6-3 แสดงรายละเอียดของคลาสหลักสูตร                            | 42      |
| ตารางที่ 6-4 แสดงรายละเอียดของคลาสการลงทะเบียน การคิดผลการศึกษา       | 43      |
| ตารางที่ 6-5 แสดงรายละเอียดของคลาสคณะ                                 | 43      |
| ตารางที่ 6-6 แสดงรายละเอียดของคลาสภาควิชา                             | 43      |
| ตารางที่ 6-7 แสดงรายละเอียดของคลาสสาขาวิชา                            | 44      |
| ตารางที่ 6-8 แสดงรายละเอียดของคลาสภาควิชาในหลักสูตร                   | 44      |
| ตารางที่ 6-9 แสดงรายละเอียดของ Property ในคลาสTempString              | 48      |
| ตารางที่ 7-1 ความต้องการของเครื่องที่ใช้ในการพัฒนา                    | 66      |
| ตารางที่ 7-2 แสดงรหัสข้อผิดพลาดที่ของ Cache' ใน ActiveX               | 72      |

# สารบัญรูป

|  | หน้าที่ |
|--|---------|
| รูปที่ 2-1 กรณีที่ 1   | 11      |
| รูปที่ 2-2 กรณีที่ 2   | 12      |
| รูปที่ 2-3 กรณีที่ 3   | 12      |
| รูปที่ 2-4 กรณีที่ 4   | 12      |
| รูปที่ 2-5 การเปลี่ยนแปลงเพศของวัว   | 14      |
| รูปที่ 2-6 กรณีของการแก้ไขแบบปัจจุบัน  | 16      |
| รูปที่ 2-7 กรณีของการลบแบบซีควนซ์  | 17      |
| รูปที่ 2-7 (ต่อ) กรณีของการลบแบบซีควนซ์  | 18      |
| รูปที่ 2-8 กรณีของการแก้ไขแบบซีควนซ์   | 19      |
| รูปที่ 2-8 (ต่อ) กรณีของการแก้ไขแบบซีควนซ์   | 20      |
| รูปที่ 4-1 แสดงโมเดลการเก็บข้อมูลของฐานข้อมูลเชิงวัตถุ   | 30      |
| รูปที่ 5-1 แสดงคลาส Temporal_Salary  | 37      |
| รูปที่ 5-2 แสดงคลาส Department   | 37      |
| รูปที่ 5-3 แสดงคลาส Temporal_employee_VT   | 37      |
| รูปที่ 6-1 รูปคลาสไดอะแกรมแสดงความสัมพันธ์ระหว่างคลาสในระบบงานทะเบียนนักศึกษา                          | 40      |
| รูปที่ 6-2 แสดงการกดปุ่ม New Class ทางด้านบน   | 45      |
| รูปที่ 6-3 แสดงหน้าต่าง Add Class Wizard หน้าแรก   | 46      |
| รูปที่ 6-4 แสดงหน้าต่าง Add Class Wizard หน้าสอง   | 46      |
| รูปที่ 6-5 แสดงคลาส TempString ตามที่ได้ทำการสร้าง   | 47      |
| รูปที่ 6-6 แสดงหน้าต่างการเพิ่ม Property   | 47      |
| รูปที่ 6-7 แสดงหน้าต่างการใส่รายละเอียดเกี่ยวกับ Property  | 48      |
| รูปที่ 6-8 แสดงการเพิ่ม Property เข้าไปในคลาส  | 48      |
| รูปที่ 6-9 แสดง Property ในคลาส TempString   | 49      |
| รูปที่ 6-10 แสดง Property ที่เกี่ยวกับการเปลี่ยนแปลงเชิงเวลา   | 49      |
| รูปที่ 6-11 แสดงการสร้างแอททริบิวต์ที่เกี่ยวกับการเปลี่ยนแปลงเชิงเวลา                                  | 50      |
| รูปที่ 6-12 รูปคลาสไดอะแกรมแสดงความสัมพันธ์ระหว่างคลาสในระบบงานทะเบียน<br>นักศึกษากับคลาสเทมโพรอลเมธอด | 50      |
| รูปที่ 6-13 แสดงการอินเฮอริตคลาสคลาสเทมโพรอลเมธอด  | 65      |
| รูปที่ 6-14 แสดงรายละเอียดของเมธอดที่มีในคลาส TemporalMethod   | 65      |
| รูปที่ 7-1 แสดงการติดต่อระหว่าง Cache' Object Server สำหรับ ActiveX                                    | 67      |
| รูปที่ 7-2 แสดงเมนู pop up ในการเขียน โค้ดใน Visual Basic จากการใช้ ActiveX Binding                    | 68      |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|   | หน้าที่ |
|---|---------|
| รูปที่ 7-3 เมนู pop up ช่วยแสดงไวยากรณ์ของการใช้ Cache' Object                          | 68      |
| รูปที่ ก-1 สถาปัตยกรรมของ View  | 79      |
| รูปที่ ก-2 uses case diagram ของบริษัทขายรถยนต์   | 79      |
| รูปที่ ก-3 สัญลักษณ์ของ Class diagram   | 80      |
| รูปที่ ก-4 ตัวอย่างของ Class diagram  | 81      |
| รูปที่ ก-5 สัญลักษณ์ของ Instance ใน Object diagram                                      | 81      |
| รูปที่ ก-6 Object diagram ของรถยนต์   | 81      |
| รูปที่ ก-7 Collaboration diagram สำหรับการสนทนาทางโทรศัพท์                              | 82      |
| รูปที่ ก-8 object message sequence chart  | 82      |
| รูปที่ ก-9 Sequence diagram ของการสนทนาทางโทรศัพท์                                      | 83      |
| รูปที่ ก-10 State ของ person  | 83      |
| รูปที่ ก-11 State diagram ของ นุคคิด  | 84      |
| รูปที่ ก-12 สัญลักษณ์ของ Component Diagram  | 84      |
| รูปที่ ก-13 Deployment Diagram การเชื่อมต่อระหว่าง Client ขากที่ต่าง ๆ และ Database     | 85      |
| รูปที่ ก-14 model element ที่ใช้ร่วมกันใน diagram                                       | 85      |
| รูปที่ ก-15 ตัวอย่างของความสัมพันธ์แบบต่าง ๆ  | 86      |
| รูปที่ ข-1 แสดงการติดตั้งไฟล์เคอร์ในตำแหน่งที่เราต้องการ                                | 87      |
| รูปที่ ข-2 แสดงการติดตั้งโปรแกรมตามที่เราได้กำหนดตำแหน่งไฟล์เคอร์                       | 87      |
| รูปที่ ข-3 รูปแสดงหน้าต่างคาเซเซตอัพ  | 88      |
| รูปที่ ข-4 แสดงการเลือกภาษาสำหรับการติดตั้งระบบจัดการฐานข้อมูลคาเซ                      | 88      |
| รูปที่ ข-5 แสดงการสอบถามเกี่ยวกับข้อตกลงในลิขสิทธิ์ และแสดงรายละเอียดเกี่ยวกับลิขสิทธิ์ | 88      |
| รูปที่ ข-6 แสดงหน้าต่างการติดตั้งการแสดงผลของระบบจัดการฐานข้อมูลคาเซ                    | 89      |
| รูปที่ ข-7 รูปแสดงการสรุปรายละเอียดเกี่ยวกับข้อมูลที่เราได้ทำการเลือกในการติดตั้งคาเซ   | 89      |
| รูปที่ ข-8 แสดงการเซตอัพระบบของระบบจัดการฐานข้อมูลคาเซ                                  | 90      |
| รูปที่ ข-9 แสดงการเลือก Configuration Manager   | 90      |
| รูปที่ ข-10 แสดงการเลือกแทป Namespace แล้วทำการกดปุ่ม Add                               | 91      |
| รูปที่ ข-11 แสดงการใส่ค่าของเนมสเปส   | 91      |
| รูปที่ ข-12 แสดงการกำหนดค่าดาต้าเบส   | 92      |
| รูปที่ ข-13 แสดงการใส่ชื่อของดาต้าเบส   | 92      |
| รูปที่ ข-14 แสดงการกำหนดตำแหน่งที่ต้องการให้ทำการเก็บดาต้าเบส                           | 93      |
| รูปที่ ข-15 แสดงการขนาดเริ่มต้นของดาต้าเบส  | 93      |
| รูปที่ ข-16 แสดงการการจบขั้นตอนการสร้างเนมสเปส  | 94      |
| รูปที่ ข-17 แสดงการการสอบถามการเปลี่ยนแปลง  | 94      |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|  | หน้าที่ |
|--|---------|
| รูปที่ ข-18 แสดงการเลือก Object Architect  | 95      |
| รูปที่ ข-19 แสดงการหน้าจอของการติดต่อไปยังเนมสเปสโดยผ่านชื่อที่ใช้ในการติดต่อ              | 95      |
| รูปที่ ข-20 แสดงหน้าจอแสดงการใส่ชื่อที่ใช้ในการติดต่อและชื่อเนมสเปสที่เราต้องการติดต่อด้วย | 96      |
| รูปที่ ข-21 แสดงหน้าต่าง Cache_Object Architect  | 96      |
| รูปที่ ข-22 แสดงวิธีการอิมพอร์ตซีดีแอลไฟล์   | 97      |
| รูปที่ ข-23 แสดงหน้าต่างซีดีแอลอิมพอร์ต  | 97      |
| รูปที่ ข-24 แสดงหน้าต่างของการเปิดซีดีแอลไฟล์  | 98      |
| รูปที่ ข-25 แสดงหน้าต่างซีดีแอลอิมพอร์ตหลังจากการเลือกซีดีแอลไฟล์                          | 98      |
| รูปที่ ข-26 แสดงหน้าต่างบอกถึงสถานะการโหลดซีดีแอลไฟล์                                      | 99      |
| รูปที่ ข-27 แสดงวิธีการเอ็กพอร์ตคลาส   | 99      |
| รูปที่ ข-28 แสดงหน้าต่างการเซฟเอ็กพอร์ตคลาส  | 100     |
| รูปที่ ข-29 แสดงหน้าต่างการเลือกคลาสที่ต้องการคอมไพล์                                      | 100     |
| รูปที่ ข-30 แสดงการกดปุ่มCompile ClassในCache_Object Architect                             | 101     |
| รูปที่ ข-31 แสดงหน้าต่างแสดงผลของการคอมไพล์  | 101     |
| รูปที่ ข-32 แสดงการกดปุ่มNew Classเพื่อทำการสร้างคลาส                                      | 102     |
| รูปที่ ข-33 แสดงการการใส่ชื่อคลาสที่เราต้องการ   | 102     |
| รูปที่ ข-34 แสดงการกำหนดชนิดของคลาสที่เราต้องการ   | 103     |
| รูปที่ ข-35 แสดงการสร้างคลาสที่เราได้กำหนด   | 103     |
| รูปที่ ข-36 แสดงรายละเอียดของPropertyที่เราได้กำหนด  | 104     |
| รูปที่ ข-37 แสดงการสร้างPropertyที่เราได้กำหนด   | 104     |
| รูปที่ ข-38 แสดงการสร้างMethodที่เราได้กำหนด   | 105     |
| รูปที่ ข-39 แสดงการกำหนดคัลกษณะของMethod   | 105     |
| รูปที่ ข-40 แสดงการเขียนMethod Code  | 106     |
| รูปที่ ค-1 แสดงการเปิดโปรเจกจากโปรแกรมวิซวลเบสิก   | 107     |
| รูปที่ ค-2 แสดงการ โหลดโปรเจกที่เราได้ทำการเลือกไว้  | 107     |
| รูปที่ ค-3 แสดงการเลือกProject ->References  | 108     |
| รูปที่ ค-4 แสดงการเซต References   | 108     |
| รูปที่ ค-5 แสดงหน้าจอหลักของระบบทะเบียนนักศึกษา  | 109     |
| รูปที่ ค-6 แสดงหน้าต่างคณะ   | 109     |
| รูปที่ ค-7 แสดงหน้าต่างภาควิชา   | 110     |
| รูปที่ ค-8 แสดงหน้าต่างสาขาวิชา  | 110     |
| รูปที่ ค-9 แสดงหน้าต่างเพิ่ม/ลบ/แก้ไขวิชา  | 111     |
| รูปที่ ค-10 แสดงหน้าต่างใส่ข้อมูลวิชา  | 111     |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|  | หน้าที่ |
|--|---------|
| รูปที่ ค-11 แสดงการใส่รหัสวิชาเพื่อทำการโหลดข้อมูลรายวิชา                        | 112     |
| รูปที่ ค-12 แสดงข้อมูลรายวิชาที่ถูกโหลด  | 112     |
| รูปที่ ค-13 แสดงการแก้ไขข้อมูลรายวิชาที่ถูกโหลด                                  | 113     |
| รูปที่ ค-14 แสดงหน้าต่างหลักสูตร   | 113     |
| รูปที่ ค-15 แสดงการกรอกรายละเอียดหลักสูตร  | 114     |
| รูปที่ ค-16 แสดงการเพิ่มหลักสูตร   | 114     |
| รูปที่ ค-17 แสดงการเลือกหลักสูตรที่ต้องการแก้ไข                                  | 115     |
| รูปที่ ค-18 แสดงหน้าต่างเพื่อหาข้อมูลคณะ/ภาควิชา/สาขาวิชา                        | 115     |
| รูปที่ ค-19 แสดงข้อมูลคณะ/ภาควิชา/สาขาวิชาที่ได้จากการคิวรี                      | 116     |
| รูปที่ ค-20 แสดงข้อมูลคณะ/ภาควิชา/สาขาวิชาที่เพิ่มเข้าไปในหลักสูตร               | 116     |
| รูปที่ ค-21 แสดงการเลือกหลักสูตรที่ต้องการเพิ่ม/ลบรายวิชาในหลักสูตร              | 117     |
| รูปที่ ค-22 แสดงหน้าต่างรายละเอียดหลักสูตร                                       | 117     |
| รูปที่ ค-23 แสดงหน้าต่างค้นหาวิชา  | 118     |
| รูปที่ ค-24 แสดงรายละเอียดวิชาที่ค้นหาได้ในฟอร์มรายละเอียดวิชา                   | 118     |
| รูปที่ ค-25 แสดงรายวิชาที่ถูกเพิ่มเข้าไปในหลักสูตร                               | 119     |
| รูปที่ ค-26 แสดงหน้าต่างข้อมูลนักศึกษา   | 119     |
| รูปที่ ค-27 แสดงข้อความยืนยันการเพิ่มข้อมูลของนักศึกษา                           | 120     |
| รูปที่ ค-28 แสดงการโหลดข้อมูลของนักศึกษา   | 120     |
| รูปที่ ค-29 แสดงการเตือนรหัสข้อมูลที่ทำกรแก้ไขนี้มีอยู่ในฐานข้อมูลแล้ว           | 121     |
| รูปที่ ค-30 แสดงการเลือกรายงานข้อมูลนักศึกษาเชิงเวลา                             | 121     |
| รูปที่ ค-31 แสดงหน้าต่างประวัตินักศึกษา  | 122     |
| รูปที่ ค-32 แสดงการใส่รหัสนักศึกษาที่ต้องการ โหลด                                | 122     |
| รูปที่ ค-33 แสดงการโหลดข้อมูลรหัสนักศึกษาที่ต้องการ โหลด                         | 123     |
| รูปที่ ค-34 แสดงประวัติชื่อภาษาไทยของนักศึกษา                                    | 123     |
| รูปที่ ค-35 แสดงประวัตินามสกุลภาษาไทยของนักศึกษา                                 | 124     |
| รูปที่ ค-36 แสดงประวัติชื่อภาษาอังกฤษของนักศึกษา                                 | 124     |
| รูปที่ ค-37 แสดงประวัตินามสกุลภาษาอังกฤษของนักศึกษา                              | 125     |
| รูปที่ ค-38 แสดงการเพิ่มข้อมูลนักศึกษาเชิงเวลาในแบบชีเควินซ์                     | 125     |
| รูปที่ ค-39 แสดงข้อมูลนักศึกษาเชิงเวลาในแบบชีเควินซ์ที่ถูกเพิ่มเข้าไปในฐานข้อมูล | 126     |
| รูปที่ ค-40 แสดงการแก้ไขข้อมูลนักศึกษาเชิงเวลาในแบบชีเควินซ์                     | 127     |
| รูปที่ ค-41 แสดงข้อมูลนักศึกษาเชิงเวลาในแบบชีเควินซ์ที่ถูกแก้ไขเข้าไปในฐานข้อมูล | 127     |
| รูปที่ ค-42 แสดงการรีเซตข้อมูลนักศึกษาเพื่อที่จะให้ทำการ โหลดข้อมูลมาใหม่        | 128     |
| รูปที่ ค-43 แสดงการลบข้อมูลนักศึกษาเชิงเวลาโดยการดับเบิลคลิกเลือกข้อมูลที่จะลบ   | 128     |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|   | หน้าที่ |
|---|---------|
| รูปที่ ค-44 แสดงการลบข้อมูลนักศึกษาเชิงเวลาโดยการใส่เป็นช่วงเวลาที่ลบ                                   | 129     |
| รูปที่ ค-45 แสดงผลของการลบข้อมูลนักศึกษาเชิงเวลา  | 129     |
| รูปที่ ค-46 แสดงการเลือกงานข้อมูลวิชาเชิงเวลา   | 130     |
| รูปที่ ค-47 แสดงหน้าต่างประวัติวิชา   | 130     |
| รูปที่ ค-48 แสดงการใส่รหัสวิชาที่ต้องการโหลด  | 131     |
| รูปที่ ค-49 แสดงการโหลดข้อมูลรหัสวิชาที่ต้องการโหลด   | 131     |
| รูปที่ ค-50 แสดงประวัติชื่อวิชาภาษาไทย  | 132     |
| รูปที่ ค-51 แสดงประวัติชื่อวิชาภาษาอังกฤษ   | 132     |
| รูปที่ ค-52 แสดงเพิ่มข้อมูลวิชาเชิงเวลาในแบบชีเควินซ์   | 133     |
| รูปที่ ค-53 แสดงข้อมูลวิชาเชิงเวลาในแบบชีเควินซ์ที่ถูกเพิ่มเข้าไปในฐานข้อมูล                            | 133     |
| รูปที่ ค-54 แสดงโหลดข้อมูลวิชาโดยการใส่รหัสวิชาที่ไม่ใช่ปัจจุบัน แต่ใส่วันที่ที่มันยังคง<br>เป็นจริงแทน | 134     |
| รูปที่ ค-55 แสดงข้อมูลประวัติวิชาที่ถูกโหลดขึ้นมา   | 134     |
| รูปที่ ค-56 แสดงการแก้ไขข้อมูลวิชาเชิงเวลาในแบบชีเควินซ์  | 135     |
| รูปที่ ค-57 แสดงข้อมูลวิชาเชิงเวลาในแบบชีเควินซ์ที่ถูกแก้ไขเข้าไปในฐานข้อมูล                            | 135     |
| รูปที่ ค-58 แสดงการรีเซตข้อมูลวิชาเพื่อที่จะให้ทำการโหลดข้อมูลมาใหม่                                    | 135     |
| รูปที่ ค-59 แสดงการลบข้อมูลวิชาเชิงเวลาโดยการดับเบิลคลิกเลือกข้อมูลที่จะลบ                              | 136     |
| รูปที่ ค-60 แสดงการลบข้อมูลวิชาเชิงเวลาโดยการใส่เป็นช่วงเวลาที่ลบ                                       | 136     |
| รูปที่ ค-61 แสดงหน้าต่างของการลงทะเบียน   | 137     |
| รูปที่ ค-62 แสดงหน้าต่างของการลงทะเบียนในครั้งแรก   | 137     |
| รูปที่ ค-63 แสดงหน้าต่างบอกรายละเอียดของการลงทะเบียน  | 138     |
| รูปที่ ค-64 แสดงตัวอย่างการเพิ่มรายวิชาในการลงทะเบียน   | 138     |
| รูปที่ ค-65 แสดงตัวอย่างการถอนรายวิชาในการลงทะเบียน   | 139     |
| รูปที่ ค-66 แสดงหน้าต่างของการแสดงผลการเรียน  | 139     |
| รูปที่ ค-67 แสดงรายละเอียดของผลการเรียนที่มีอยู่  | 140     |
| รูปที่ ค-68 แสดงการใส่ผลการศึกษา  | 140     |
| รูปที่ ค-69 แสดงการประมวลผลสำหรับการพิจารณาจบ   | 141     |
| รูปที่ ค-70 แสดงตัวอย่างของรายงานสรุปผลการศึกษา   | 142     |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มา

ในปัจจุบัน โลกเป็นยุคของสารสนเทศและข้อมูลต่างๆ ยิ่งในธุรกิจต่างๆด้วยแล้วนั้น ข้อมูลถือได้ว่ามีความสำคัญเป็นอย่างมาก ในการดำเนินธุรกิจต่างๆ ข้อมูลต่างๆเหล่านี้มีจำนวนมากมาย หากไม่ได้รับการจัดการดูแลที่ดีแล้วนั้น ก็จะทำให้ไม่สามารถนำมาใช้ประโยชน์ได้อย่างมีประสิทธิภาพ จึงมีความจำเป็นอย่างยิ่งในการนำเอา ระบบจัดการฐานข้อมูล มาใช้ในการบริหารและจัดการข้อมูลต่างๆ

อย่างไรก็ดี การจัดการฐานข้อมูลที่ใช้กันอยู่ในปัจจุบันส่วนใหญ่นั้น ถูกออกแบบมาเพื่อเก็บข้อมูลสถานะสุดท้ายเพียงสถานะเดียว เมื่อมีการเปลี่ยนแปลงข้อมูล ไม่ว่าจะเป็นการลบหรือแก้ไข ข้อมูลที่มีอยู่เดิม จะถูกลบทิ้งไปจากฐานข้อมูล แม้ว่าฐานข้อมูลต่างๆไปจะสามารถรองรับบางงานได้ดี แต่บางครั้งก็ไม่เพียงพอ กับงานที่ต้องการข้อมูลในอดีต (Past Data) และ/หรือข้อมูลในอนาคต (Future Data) ด้วย สิ่งที่ต้องการก็คือ ฐานข้อมูลที่สนับสนุนการเก็บและการตอบคำถาม (Query) ข้อมูลที่เปลี่ยนแปลงตามเวลาอย่างเต็มที่ ฐานข้อมูลที่เก็บข้อมูลในอดีต ข้อมูลในปัจจุบัน และข้อมูลในอนาคต เรียกว่า ฐานข้อมูลเชิงเวลา (Temporal Database)

การนำฐานข้อมูลเชิงสัมพันธ์มาประยุกต์ใช้กับฐานข้อมูลเชิงเวลานั้น ซึ่งแต่เดิมใช้กันอยู่นั้น ยังคงมีปัญหาต่างๆอยู่ทั้งเรื่องของความซ้ำซ้อนของข้อมูล ปัญหาเรื่องของคีย์หลัก เนื่องมาจากการนำโมเดลของฐานข้อมูลเชิงสัมพันธ์มาประยุกต์ใช้ยังมีคงมีปัญหาดังกล่าวอยู่ ด้วยข้อเสียต่างๆเหล่านี้ ทำให้มีการนำเอาฐานข้อมูลเชิงวัตถุมาใช้แทนฐานข้อมูลเชิงสัมพันธ์ เพื่อแก้ไขปัญหาต่างๆ โดยอาศัยคุณสมบัติและข้อดีต่างๆ ของฐานข้อมูลเชิงวัตถุ มาประยุกต์ใช้กับฐานข้อมูลเชิงเวลา โดยปริญาณิพนธ์นี้ได้จะนำเสนอการนำเอาฐานข้อมูลเชิงวัตถุมาใช้กับฐานข้อมูลเชิงเวลา เสนอแนวคิดในการออกแบบ ตัวอย่างการนำเอามาประยุกต์ใช้งานเบื้องต้น และบทสรุปของการทบทวน

### 1.2 วัตถุประสงค์ของงานวิจัย

- 1.2.1 เพื่อศึกษาการนำฐานข้อมูลเชิงวัตถุ (Object-Oriented Database) มาประยุกต์ใช้กับงานที่เกี่ยวข้องกับข้อมูลเชิงเวลา (Temporal Data)
- 1.2.2 เพื่อศึกษาการนำเอาฐานข้อมูลเชิงเวลามาประยุกต์ใช้กับงานที่เกี่ยวข้องกับข้อมูลเชิงเวลา
- 1.2.3 เพื่อศึกษาถึงข้อดีและข้อเสียของการนำ ฐานข้อมูลเชิงวัตถุมาประยุกต์ใช้กับ ฐานข้อมูลเชิงเวลา

### 1.3 ขอบเขตของงานวิจัย

โครงการนี้เป็นโครงการที่มีลักษณะเป็นงานวิจัย โดยการนำฐานข้อมูลเชิงวัตถุมาประยุกต์ใช้กับฐานข้อมูลเชิงเวลา ศึกษาถึงคุณสมบัติและข้อดีของฐานข้อมูลเชิงวัตถุ เพื่อหาวิธีที่เหมาะสมในการนำมาใช้กับฐานข้อมูลเชิงเวลา ทั้งในส่วนของ Data Structure และการออกแบบฐานข้อมูล และนำเสนอแนวคิดที่ออกแบบนั้นไปประยุกต์ใช้กับงานตัวอย่าง เพื่อศึกษาถึงข้อดีและข้อเสียของระบบว่ามีความเหมาะสมมากน้อยเพียงใดในการ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น เมื่อผู้ใดเห็นหน้าเบาะประเขียนต้นการไว้ ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำแนวคิดของฐานข้อมูลเชิงวัตถุมาใช้กับฐานข้อมูลเชิงเวลา และมีการเปรียบเทียบข้อดีและข้อเสียระหว่างฐานข้อมูลเชิงวัตถุกับฐานข้อมูลเชิงสัมพันธ์มาใช้งาน สำหรับส่วนของงานแอปพลิเคชัน(Application)ที่ทำในงานวิจัยชิ้นนี้คืองานทะเบียนนักศึกษา โดยในส่วนของงานแอปพลิเคชันนั้นที่นำมาพัฒนานั้นจะใช้ระบบจัดการฐานข้อมูลเชิงวัตถุมาช่วยในการจัดการกับข้อมูลเชิงเวลา เพื่อแสดงว่าเมื่อนำเอาระบบจัดการฐานข้อมูลเชิงวัตถุมาช่วยในการจัดการกับฐานข้อมูลเชิงเวลาแล้วมีความสะดวกและการจัดการได้ดีเพียงใด

#### 1.4 วิธีการดำเนินงาน

โครงการนี้จะเป็นการศึกษาทฤษฎีที่เกี่ยวข้องกับงานวิจัย โดยมีเรื่องต่างๆดังต่อไปนี้คือ

- ฐานข้อมูลเชิงเวลา ศึกษาลักษณะและคุณสมบัติต่างๆของฐานข้อมูลเชิงเวลา
- ความซับซ้อนของข้อมูล ศึกษาความซับซ้อนแบบต่างๆของข้อมูลเชิงเวลา
- ตารางวลิตใหม่และตารางทรานแซกชันใหม่ ศึกษาการใช้ฐานข้อมูลเชิงสัมพันธ์จัดการกับข้อมูลเชิงเวลา
- ภาษา SQL เชิงเวลา ศึกษาการใช้ฐานข้อมูลเชิงเวลาจัดการกับข้อมูลเชิงเวลา
- งานตัวอย่าง ศึกษาเกี่ยวกับงานทะเบียนนักศึกษา รายละเอียดเกี่ยวกับงานทะเบียนนักศึกษา ภายหลังจากที่ได้ศึกษาแล้ว ก็จะมาทำการออกแบบระบบฐานข้อมูล จากนั้นจึงนำฐานข้อมูลที่เราได้ทำการออกแบบ มาลงมือปฏิบัติตามที่ได้ออกแบบ

งานวิจัยในโครงการนี้จะเริ่มด้วยการศึกษาทฤษฎีพื้นฐานต่าง ๆ ที่เกี่ยวข้องกับงานวิจัย ซึ่งก็มีเรื่องหลัก ๆ อยู่ 3 เรื่องด้วยกัน คือ แนวคิดของฐานข้อมูลเชิงวัตถุ แนวคิดเชิงวัตถุ ฐานข้อมูลเชิงวัตถุ และแนวคิดของการนำเอาฐานข้อมูลเชิงวัตถุมาประยุกต์ใช้กับฐานข้อมูลเชิงเวลา ซึ่งมีรายละเอียดอยู่ในบทที่ 2,3,4 และ 5 และกล่าวถึงความเหมาะสมของการนำเอาฐานข้อมูลเชิงวัตถุมาใช้แก้ปัญหาในฐานข้อมูลเชิงเวลา ซึ่งเดิมการใช้ฐานข้อมูลแบบสัมพันธ์นั้นยังพบว่ามีปัญหาอยู่ ซึ่งในงานวิจัยได้จะแสดงให้เห็นถึงเป็นหาค้างกล่าว และเสนอแนะแนวทางในการแก้ไข โดยการใช้ฐานข้อมูลเชิงวัตถุ

จากนั้นก็เริ่มเข้าสู่ขั้นตอนของการพัฒนาโปรแกรม โดยได้ยกเอาระบบทะเบียนนักศึกษามาเป็นกรณีศึกษา โดยบทที่ 6 จะกล่าวถึงการออกแบบฐานข้อมูลของระบบทะเบียนนักศึกษา โดยใช้แนวคิดเชิงวัตถุมาใช้ในการทำการวิเคราะห์และออกแบบ และบทถัดไปจะเป็นแนวทางการออกแบบฐานข้อมูลเชิงเวลาโดยใช้ฐานข้อมูลเชิงวัตถุ กล่าวถึงเมทริกซ์ต่างๆ ที่สร้างขึ้นมานำไปใช้เกี่ยวกับข้อมูลเชิงเวลา และบทสุดท้ายจะกล่าวถึงปัญหาที่พบ เป็นการเสนอแนะแนวทางการนำไปพัฒนาต่อไป แนวทางการนำไปประยุกต์ใช้ และกล่าวสรุปถึงการทำให้โครงการครั้งนี้ผลที่ได้รับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### ฐานข้อมูลเชิงเวลา

#### 2.1 ฐานข้อมูลเชิงเวลา

ข้อมูลที่เก็บอยู่ในฐานข้อมูล ท้ายๆไปส่วนใหญ่แล้วจะเก็บเฉพาะข้อมูลที่เป็นความจริงในปัจจุบันเท่านั้น อดีตและอนาคตของข้อมูลนั้นจะไม่ถูกเก็บไว้ในฐานข้อมูล ข้อมูลในอดีตนั้นหมายถึงข้อมูลที่อาจจะเคยอยู่ใน ฐานข้อมูลมาก่อนแล้วมีการแก้ไขเปลี่ยนแปลงหรือลบออกไปจากฐานข้อมูล แล้ว ณ เวลาปัจจุบัน

แต่ฐานข้อมูลเชิงเวลานั้น ข้อมูลที่ถูกเก็บอยู่ใน ฐานข้อมูลจะมีความแตกต่างจากข้อมูลที่เก็บในฐานข้อมูลทั่วๆไป ตรงที่จะมีการเพิ่มส่วนของเวลาติดไปกับส่วนของข้อมูล เพื่อเก็บช่วงของเวลาที่ข้อมูลนี้เป็นจริง (valid time) หรือช่วงเวลาที่ข้อมูลถูกเก็บอยู่ในฐานข้อมูล (transaction time) เวลาที่ถูกจัดเก็บอยู่ในฐานข้อมูลนั้นแบ่งได้ 3 ชนิดดังนี้

##### 2.1.1 Valid Time หมายถึง เวลาที่ข้อมูลนั้นเป็นจริง เช่น

| SSN   | Ename | Salary | Valid from | Valid_to   |
|-------|-------|--------|------------|------------|
| 00001 | Smith | 25000  | 03/01/1995 | 05/20/1998 |
| 00002 | Jane  | 30000  | 03/15/1995 | 12/31/9999 |

ตารางที่ 2-1 แสดงตัวอย่างของการเก็บ valid time ลงในฐานข้อมูล

จากตารางที่ 2-1 ในแถวแรกนั้นมีความหมายว่า ลูกจ้างชื่อ Smith มีเงินเดือน คือ 25000 ตั้งแต่วันที่ 1 มีนาคม 1995 จนถึงวันที่ 20 พฤษภาคม 1998 และในแถวที่สอง มีความหมายว่า ลูกจ้างชื่อ Jane มีเงินเดือน ตั้งแต่วันที่ 15 มีนาคม 1995 จนถึงปัจจุบัน ก็ยังเป็นจริงอยู่ตลอดไป ยังคงทำงานอยู่ที่นี้ โดยวันที่ “12/31/9999” แทนค่าของอนาคตเพื่อบอกสถานะปัจจุบันของข้อมูลตัวนั้น

##### 2.1.2 Transaction Time หมายถึง เวลาที่ข้อมูลนั้นถูกเก็บอยู่ใน ฐานข้อมูลจริงๆ หรืออาจกล่าวง่ายๆได้ว่า เป็นเวลาเก็บไว้เพื่อติดตามการจัดการ(manipulate)กับฐานข้อมูลซึ่งได้แก่ การทำinsert, delete , update เช่น

| SSN   | Ename | Salary | Tx_start   | Tx_stop    |
|-------|-------|--------|------------|------------|
| 00001 | Smith | 25000  | 03/01/1995 | 05/20/1998 |
| 00002 | Jane  | 30000  | 03/15/1995 | 12/31/9999 |

ตารางที่ 2-2 แสดงตัวอย่างของการเก็บ transaction time ลงในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 2-2 ในแถวแรกมีความหมายว่า ความจริงที่ถูกอ้างอิงชื่อ Smith มีเงินเดือน 25000 ถูกเก็บลงในฐานข้อมูลเมื่อ 1 มีนาคม 1995 จนถึงวันที่ 20 พฤษภาคม 1998 และในแถวที่สอง มีความหมายว่า ความจริงที่ถูกอ้างอิงชื่อ Jane มีเงินเดือน 30000 ถูกเก็บลงในฐานข้อมูลตั้งแต่วันที่ 15 มีนาคม 1995 จนถึง ปัจจุบันก็ยังคงอยู่ในฐานข้อมูล

2.1.3 User Defined Time หมายถึงคุณสมบัติของความจริงที่ไม่ต้องแปลความหมายของวันหรือเวลา อย่างเช่น วันเกิด หรือ วันแรงงาน ซึ่งเป็นคุณสมบัติที่ไม่ต้องการการจัดการทำให้ผู้ใช้จะต้องจัดการเอง ไม่ว่าจะ insert delete หรือ update ซึ่งจะไม่ว่ากล่าวถึงในรายงานฉบับนี้

## 2.2 ความซ้ำซ้อนที่เกิดขึ้นในฐานข้อมูลเชิงเวลา

ปกติการจัดเก็บข้อมูลลงในฐานข้อมูลจะเป็นการจัดเก็บความจริง(Fact) ของข้อมูลลงในฐานข้อมูลซึ่งเป็นสถานะของข้อมูลล่าสุด มิได้จัดเก็บช่วงเวลาที่มีข้อมูลนั้นเป็นจริงลงไปด้วย แต่ถ้ามีการจัดเก็บเวลาต่างลงไปด้วย สิ่งก็ตามมาก็คือ ความซ้ำซ้อนที่เกิดขึ้นกับฐานข้อมูลที่มีเก็บข้อมูลเหล่านั้นเพราะเวลาที่เก็บเข้าไปกับข้อมูลนั้นอาจจะมีการคาบเกี่ยวกันเกิดขึ้นนั่นเอง

พิจารณาตาราง NICUStatus ซึ่งเป็นตารางที่เก็บสถานะของผู้ป่วยที่รักษาตัวอยู่ที่ Neonatal Intensive care Unit Moines ซึ่งแสดงดังตารางที่ 2-3

| Name           | Status   | From_date  | To_date    |
|----------------|----------|------------|------------|
| Kenneth Robert | Serious  | 1997-11-19 | 1997-11-21 |
| Alexis May     | Serious  | 1997-11-19 | 1997-11-27 |
| Natalie Sue    | Serious  | 1997-11-19 | 1997-11-25 |
| Kelsey Ann     | Serious  | 1997-11-19 | 1997-11-26 |
| Brandon James  | Serious  | 1997-11-19 | 1997-11-26 |
| Nathan Roy     | Serious  | 1997-11-19 | 1997-11-28 |
| Joel Steven    | Critical | 1997-11-19 | 1997-11-20 |
| Joel Steven    | Serious  | 1997-11-20 | 1997-11-26 |
| Kenneth Robert | Fair     | 1997-11-21 | 1998-01-03 |
| Alexis May     | Fair     | 1997-11-27 | 1998-01-11 |
| Alexis May     | Fair     | 1997-12-02 | 1999-12-31 |
| Alexis May     | Fair     | 1997-12-02 | 1999-12-31 |

ตารางที่ 2-3 ตัดทอนมาจาก ตาราง NICUStatus

From\_date แสดงว่าผู้ป่วยมีอาการนั้นๆตั้งแต่วันที่ ไหน

To\_date แสดงว่าผู้ป่วยมีอาการเปลี่ยนไปในวันที่ ไหน

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาวิจัยเท่านั้น ไม่สามารถนำข้อมูลไปใช้ในการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

“Close-Open” กล่าวคือ From\_date จะรวมอยู่ในช่วง แต่ To\_date จะไม่รวมอยู่ช่วง เช่น ในแถวสุดท้ายนั้น วันที่ 2 ธันวาคม 1997 จะรวมอยู่ในช่วงเวลาที่ Alexis มีอาการ Fair แต่ในวันที่ 31 ธันวาคม 1999 จะไม่รวมอยู่ในช่วงนั้น

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

ความซ้ำซ้อนในฐานข้อมูลเชิงเวลานั้นอาจแบ่งออกได้เป็น 4 ประเภท ได้แก่

- 2.2.1 Non-sequence Duplicate เป็นความซ้ำซ้อนที่เกิดขึ้นใน row ใดๆที่เหมือนกัน ในทุกๆ column เช่นใน 2 row สุดท้ายในตารางที่ 3
- 2.2.2 Value-equivalent Duplicate เป็นความซ้ำซ้อนที่เกิดขึ้นเมื่อตัด column ที่เกี่ยวข้องกับเวลาออกไปแล้ว เช่นใน 3 row สุดท้ายในตารางที่ 3
- 2.2.3 Current Duplicate เป็นความซ้ำซ้อนที่เกิดขึ้นในระยะเวลาใดเวลาหนึ่งในปัจจุบัน เช่น ถ้ากำหนดให้ปัจจุบันเป็นวันที่ 6 มกราคม 1998 จะได้ว่าเกิดความซ้ำซ้อนใน 3 row สุดท้าย
- 2.2.4 Sequence Duplicate เป็นความซ้ำซ้อนที่เกิดขึ้นบนช่วงเวลาใดๆ เช่น Alexis มีอาการอยู่ในขั้น Fair ใน 3 row สุดท้าย เมื่ออยู่ในช่วงเวลาตั้งแต่ เดือนธันวาคม 1997 ถึง 11 วันแรกของปี 1998

ตารางที่ 2-4 แสดงให้เห็นถึงความสัมพันธ์ของความซ้ำซ้อนแบบต่างๆ ยกตัวอย่างเช่น ถ้าเป็นความซ้ำซ้อนแบบนอนซีเควนซ์ก็จะเป็นความซ้ำซ้อนแบบซีเควนซ์ด้วย แต่ถ้าเป็นความซ้ำซ้อนแบบซีเควนซ์ก็ไม่จำเป็นต้องเป็นความสัมพันธ์แบบนอนซีเควนซ์ เป็นต้น

| ความซ้ำซ้อน | ซีเควนซ์ | ปัจจุบัน | ค่าเท่ากัน | นอนซีเควนซ์ |
|-------------|----------|----------|------------|-------------|
| ซีเควนซ์    | ✓        |          | ✓          |             |
| ปัจจุบัน    | ✓        | ✓        | ✓          |             |
| ค่าเท่ากัน  |          |          | ✓          |             |
| นอนซีเควนซ์ | ✓        |          | ✓          | ✓           |

ตารางที่ 2-4 ความสัมพันธ์ของความซ้ำซ้อนแบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3 ประเภทของฐานข้อมูลเชิงเวลา

วาลิดไทม์และทรานแซกชันไทม์ก่อให้เกิดฐานข้อมูลเชิงเวลาหลายลักษณะดังต่อไปนี้

### 2.3.1 ฐานข้อมูลวาลิดไทม์ (Valid-Time Database) หรือฐานข้อมูลประวัติ (Historical Database) เก็บเฉพาะวาลิดไทม์

| NAME  | SSN   | SALARY | VALID_START | VALID_END  |
|-------|-------|--------|-------------|------------|
| Smith | 00001 | 25000  | 1997-06-15  | 1998-05-31 |
| Smith | 00001 | 30000  | 1998-06-01  | 9999-12-31 |
| Brown | 00002 | 40000  | 1996-05-01  | 1997-08-10 |
| Wong  | 00003 | 38000  | 1998-08-01  | 9999-31-12 |

ตารางที่ 2-5 ตัวอย่างของฐานข้อมูลวาลิดไทม์

ฐานข้อมูลวาลิดไทม์จะเก็บช่วงเวลาที่เป็นจริงเท่านั้น จากตัวอย่างเป็นฐานข้อมูลเชิงเวลาที่เก็บประวัติ เงินเดือนของลูกจ้างในบริษัทแห่งหนึ่ง จากตาราง 2-5 จะพบว่า 2 แถวแรกของตาราง จะบอกถึงประวัติเงินเดือนของนาย Smith ซึ่งเคยได้รับเงินเดือน 25000 ระหว่างวันที่ 15 มิถุนายน 1997 ถึง วันที่ 31 เมษายน 1998 และปัจจุบัน (ปัจจุบันและอนาคตแทนเวลาด้วยค่า 9999-31-12) ได้รับเงินเดือน 30000 นับตั้งแต่วันที่ 1 มิถุนายน 1998 และจากตารางจะพบว่าข้อมูลของพนักงานที่ไม่ได้ถูกว่าจ้างแล้วก็ยังคงมีอยู่ เช่น พนักงานที่ชื่อ Brown ซึ่งเคยทำงานและได้รับเงินเดือน 40000 ระหว่างวันที่ 1 พฤษภาคม 1996 ถึง วันที่ 10 สิงหาคม 1997 เป็นต้น

### 2.3.2 ฐานข้อมูลทรานแซกชันไทม์ (Transaction-Time Database) หรือฐานข้อมูลย้อนกลับ (Rollback Database) เก็บเฉพาะทรานแซกชันไทม์

| ชั่วโมง | นาทีก่อน | วินาที | มุมเอียง | นาทีก่อนเอียง | ผู้ค้นพบ   | ความสว่าง | เวลาเริ่มต้น | เวลาสิ้นสุด |
|---------|----------|--------|----------|---------------|------------|-----------|--------------|-------------|
| 00      | 00       | 00     | 75       | 30            | 'A 1248'   | 12.0      | 03/12/1989   | 11/14/1992  |
| 00      | 00       | 09     | 75       | 30            | 'A 1248'   | 12.0      | 11/15/1992   | 05/17/1994  |
| 00      | 00       | 09     | 75       | 30            | 'A 1248'   | 10.5      | 05/18/1994   | 07/22/1995  |
| 00      | 00       | 08     | 75       | 30            | 'A 1248'   | 10.5      | 07/23/1995   | 12/31/9999  |
| 05      | 57       | 40     | 00       | 02            | 'BU 1190'  | 6.5       | 11/08/1988   | 12/31/9999  |
| 04      | 13       | 20     | 50       | 32            | 'CHR 15'   | 15.5      | 02/09/1990   | 12/31/9999  |
| 01      | 23       | 70     | -09      | 55            | 'HJ 3433'  | 10.5      | 03/25/1991   | 12/31/9999  |
| 02      | 33       | 10     | -09      | 25            | 'LDS 3402' | 10.6      | 12/19/1993   | 07/08/1996  |

ตารางที่ 2-6 ตัวอย่างของฐานข้อมูลทรานแซกชันไทม์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลทรานแซกชันใหม่ จะเก็บค่าเวลาที่ ความจริงนั้นถูกเก็บอยู่ใน ฐานข้อมูลจริงๆ หรือ อาจกล่าวง่ายๆ ได้ว่า เป็นเวลาเก็บไว้เพื่อติดตามการจัดการ(manipulate)กับฐานข้อมูล จากตัวอย่างเป็นฐานข้อมูลที่เก็บประวัติการค้นพบของดวงดาวต่างๆ ซึ่งจะมีการค้นพบแล้วมีการบันทึกลงในฐานข้อมูล จากตัวอย่าง ตารางดวงดาว เป็นตารางทรานแซกชันใหม่ของตารางที่ 2-6 โดย 5 คอลัมน์แรกของตาราง เป็นตำแหน่งของดวงดาว คอลัมน์ผู้ค้นพบประกอบด้วยตัวอักษรตั้งแต่ 1 ถึง 3 ตัวอักษรตามด้วยหมายเลขผู้ค้นพบ คอลัมน์ผู้ค้นพบนี้จะ เป็น Primary Key ของตารางนี้ด้วย ส่วนคอลัมน์เวลาเริ่มต้นจะเก็บเวลาที่ข้อมูลถูกใส่เข้าไปในตารางหรือเวลาที่ข้อมูลถูกแก้ไข (ข้อมูลใหม่จะถูกเก็บไว้ที่นี้) และคอลัมน์เวลาสิ้นสุดจะเก็บเวลาที่ข้อมูลถูกลบออกจากตารางหรือถูกแก้ไข (ข้อมูลเก่าจะถูกเก็บไว้ที่นี้)

จะเห็นว่าข้อมูลของดวงดาว LDS 3402 ถูกใส่เข้าไปในวันที่ 19 เดือนธันวาคม ปี ค.ศ. 1993 แล้วถูกลบออกในวันที่ 8 เดือนกรกฎาคม ปี ค.ศ. 1996 เนื่องจากพบข้อผิดพลาดของข้อมูล และข้อมูลของดวงดาว A 1248 ถูกใส่เข้าไปครั้งแรกเมื่อวันที่ 12 เดือนมีนาคม ปี ค.ศ. 1989 และได้ทำการแก้ไขข้อมูลตำแหน่งวินาทีของดวงดาวนี้ในวันที่ 15 เดือนพฤศจิกายน ปี ค.ศ. 1992 แก้ไขข้อมูลความสว่างในวันที่ 18 เดือนพฤษภาคม ปี ค.ศ. 1994 และแก้ไขข้อมูลตำแหน่งวินาทีในวันที่ 23 เดือนกรกฎาคม ปี ค.ศ. 1995 สังเกตว่าการเปลี่ยนแปลงเหล่านี้ไม่ได้หมายความว่าดวงดาวเกิดการเปลี่ยนแปลง แต่หมายความว่าเกิดความผิดพลาดขึ้นในการวัดครั้งก่อน จึงทำการแก้ไขข้อมูลให้ถูกต้อง ดังนั้นแถวที่มีคอลัมน์เวลาสิ้นสุดเป็นอดีตจะเป็นข้อมูลที่ไม่ถูกต้อง

สังเกตว่าข้อมูลในตารางทรานแซกชันใหม่จะเป็นข้อมูลตั้งแต่อดีตจนถึงปัจจุบันเท่านั้น เนื่องจากไม่มีทางทราบได้เลยว่าในอนาคตข้อมูลใดจะเกิดข้อผิดพลาด

### 2.3.3 ฐานข้อมูลไบเทมโพรอล (Bitemporal Database) เก็บทั้งเวลาอดีตและทรานแซกชันใหม่

ฐานข้อมูลไบเทมโพรอลจะเก็บประวัติของข้อมูลทั้งเวลาอดีต และทรานแซกชันใหม่ สังเกตว่าประวัติที่ว่าข้อมูลถูกเก็บไว้ในฐานข้อมูลเมื่อใดหรือทรานแซกชันใหม่ ถูกจำกัดที่เวลาในอดีตจนถึงปัจจุบัน เพราะว่าทรานแซกชันใหม่ถูกจัดการ โดยระบบโดยตรง

การพิจารณาว่าจะใช้ฐานข้อมูลชนิดใดนั้น ขึ้นอยู่กับลักษณะของงานที่เราจะนำไปใช้ แต่โดยส่วนมากแล้ว ฐานข้อมูลเชิงเวลาที่ถูกนำไปใช้มักจะเป็นฐานข้อมูลเชิงเวลาแบบเวลาอดีตเสียเป็นส่วนใหญ่ ทั้งนี้เนื่องจากโดยส่วนมากงานต่างๆ จะให้ความสำคัญกับช่วงเวลาที่มีข้อมูลมากกว่า เวลาในการ manipulation ข้อมูลนั้น

## 2.4 ตัวอย่างแอปพลิเคชันที่เกี่ยวข้องกับข้อมูลเชิงเวลา

- 2.4.1 การเงิน เช่น การวิเคราะห์แนวโน้มของหุ้นในแต่ละวันทำการ
- 2.4.2 การประกันภัย เช่น กรรมธรรม์ประกันภัยจะคุ้มครองในช่วงเวลาใด
- 2.4.3 ระบบการจอง เช่น ห้องพักในโรงแรมห้องใดถูกจองในช่วงเวลาใด
- 2.4.4 ระบบการจัดการข้อมูลทางการแพทย์ เช่น ประวัติของคนไข้
- 2.4.5 ระบบการสนับสนุนการตัดสินใจ (Decision Support Systems) เช่น การวางแผนใน

อนาคค  
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2.4.6 ระบบการผลิต เช่น การวิเคราะห์คุณภาพและปริมาณของสินค้าในแต่ละชั่วโมง
- 2.4.7 การวิจัยทางวิทยาศาสตร์ เช่น การวิเคราะห์คุณภาพของอากาศในแต่ละวัน
- ในความเป็นจริงเกือบทุกแอปพลิเคชันจะมีความเกี่ยวข้องกับข้อมูลเชิงเวลา

## 2.5 คำถามเชิงเวลาในฐานะข้อมูลเชิงเวลา

เพื่อแสดงให้เห็นถึงการตอบคำถามเชิงเวลา ซึ่งเป็นคำถามที่เกี่ยวข้องโดยตรงกับฐานข้อมูลเชิงเวลา ซึ่งลักษณะของคำถามจะมีอยู่ด้วยกันหลายลักษณะ หัวข้อต่อไปนี้จะแสดงให้เห็นตัวอย่างของคำถามเชิงเวลา ต่างๆ โดยการใช้ตัวอย่างภาษา SQL กับตารางวลิดใหม่ ขอยกตัวอย่างตารางวิวดังตารางที่ 7

| หมายเลขฟาร์ม | หมายเลขลือท | หมายเลขคอก | จำนวนวัว | เวลาเริ่มต้น | เวลาสิ้นสุด |
|--------------|-------------|------------|----------|--------------|-------------|
| 1            | 317         | 1          | 17       | 02/07/1998   | 02/17/1998  |
| 1            | 219         | 1          | 43       | 02/25/1998   | 02/28/1998  |
| 1            | 219         | 1          | 20       | 03/01/1998   | 03/13/1998  |
| 1            | 219         | 2          | 23       | 03/01/1998   | 03/13/1998  |
| 1            | 219         | 2          | 43       | 03/14/1998   | 12/31/9999  |
| 1            | 374         | 1          | 14       | 02/20/1998   | 12/31/9999  |

ตารางที่ 2-7 ตารางวิว

### 2.5.1 การตอบคำถามแบบทั่วไป (Conventional Query)

พิจารณาคำถามที่ว่า “ในแต่ละคอกมีวัวกี่ตัวที่มาจากลือทที่ 219 ในฟาร์มที่ 1” จะสามารถตอบได้โดยใช้ SQL ดังต่อไปนี้

```
SELECT หมายเลขคอก, จำนวนวัว
FROM วิว
WHERE หมายเลขฟาร์ม = 1 AND หมายเลขลือท = 219
```

### 2.5.2 การตอบคำถามแบบปัจจุบัน (Current Query)

พิจารณาคำถามที่ว่า “ปัจจุบันในแต่ละคอกมีวัวกี่ตัวที่มาจากลือทที่ 219 ในฟาร์มที่ 1” จะสามารถตอบได้โดยใช้ SQL ดังต่อไปนี้

```
SELECT หมายเลขคอก, จำนวนวัว
FROM วิว
WHERE หมายเลขฟาร์ม = 1 AND หมายเลขลือท = 219
```

AND เวลาสิ้นสุด = '12/31/9999'

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้คำตอบดังตารางที่ 4-2

| หมายเลขคอก | จำนวนวัว |
|------------|----------|
| 2          | 43       |

ตารางที่ 2-8 คำตอบของตัวอย่างการตอบคำถามแบบปัจจุบัน

### 2.5.3 การตอบคำถามแบบซีควเอนซ์ (Sequenced Query)

พิจารณาคำถามที่ว่า “ตั้งแต่ในอดีตจนถึงปัจจุบันในแต่ละคอกมีวัวกี่ตัวที่มาจากลือทที่ 219 ในฟาร์มที่ 1” จะสามารถตอบได้โดยใช้ SQL ดังต่อไปนี้

```
SELECT หมายเลขคอก, จำนวนวัว, เวลาเริ่มต้น, เวลาสิ้นสุด
FROM วัว
WHERE หมายเลขฟาร์ม = 1 AND หมายเลขลือท = 219
```

จะได้คำตอบดังตารางที่ 2-9

| หมายเลขคอก | จำนวนวัว | เวลาเริ่มต้น | เวลาสิ้นสุด |
|------------|----------|--------------|-------------|
| 1          | 43       | 02/25/1998   | 02/28/1998  |
| 1          | 20       | 03/01/1998   | 03/13/1998  |
| 2          | 23       | 03/01/1998   | 03/13/1998  |
| 2          | 43       | 03/14/1998   | 12/31/9999  |

ตารางที่ 2-9 คำตอบของตัวอย่างการตอบคำถามแบบซีควเอนซ์

### 2.5.4 การตอบคำถามแบบนอนซีควเอนซ์ (Nonsequenced Query)

พิจารณาคำถามที่ว่า “ในเวลาใดๆในแต่ละคอกมีวัวกี่ตัวที่มาจากลือทที่ 219 ในฟาร์มที่ 1” จะสามารถตอบได้โดยใช้ SQL ดังต่อไปนี้

```
SELECT หมายเลขคอก, จำนวนวัว
FROM วัว
WHERE หมายเลขฟาร์ม = 1 AND หมายเลขลือท = 219
```

จะได้คำตอบดังตารางที่ 2-10

| หมายเลขคอก | จำนวนวัว |
|------------|----------|
| 1          | 43       |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|   |    |
|---|----|
| 1 | 20 |
| 2 | 23 |
| 2 | 43 |

ตารางที่ 2-10 คำตอบของตัวอย่างการตอบคำถามแบบนอนซีเควนซ์

## 2.6 การตอบคำถามเชิงเวลาที่มีการ JOIN

การตอบคำถามโดยส่วนมากนั้น ไม่ได้ข้อมูลครบจากตารางเพียงตารางเดียว โดยส่วนมากจะหาคำตอบจากหลายๆตาราง ซึ่งการทำเช่นนั้น จำเป็นต้องมีการ join กันของ ตารางเพื่อให้ได้คำตอบที่สมบูรณ์ ต่อไปนี้เป็นตัวอย่างการตอบคำถามเชิงเวลา ที่ต้องอาศัยการ join กันของข้อมูลจากตารางต่างๆ

### 2.6.1 การตอบคำถามแบบทั่วไปโดยใช้การ JOIN (Conventional Join Query)

พิจารณาคำถามที่ว่า “มีวัวลือทใดที่อยู่ด้วยกันในแต่ละคอก” จะสามารถตอบได้โดยใช้ SQL ดังต่อไปนี้

```
SELECT L1.หมายเลขลือท, L2.หมายเลขลือท, L1.หมายเลขคอก
FROM วัว AS L1, วัว AS L2
WHERE L1.หมายเลขลือท < L2.หมายเลขลือท
AND L1.หมายเลขฟาร์ม = L2.หมายเลขฟาร์ม
AND L1.หมายเลขคอก = L2.หมายเลขคอก
```

### 2.6.2 การตอบคำถามแบบปัจจุบันโดยใช้การ JOIN (Current Join Query)

พิจารณาคำถามที่ว่า “ปัจจุบันมีวัว Lot ใดที่อยู่ด้วยกันในแต่ละคอก” จะสามารถตอบได้โดยใช้ SQL ดังต่อไปนี้

```
SELECT L1.หมายเลขลือท, L2.หมายเลขลือท, L1.หมายเลขคอก
FROM วัว AS L1, วัว AS L2
WHERE L1.หมายเลขลือท < L2.หมายเลขลือท
AND L1.หมายเลขฟาร์ม = L2.หมายเลขฟาร์ม
AND L1.หมายเลขคอก = L2.หมายเลขคอก
AND L1.เวลาสิ้นสุด = '12/31/9999'
AND L2.เวลาสิ้นสุด = '12/31/9999'
```

### 2.6.3 การตอบคำถามแบบนอนซีควเอนซ์โดยใช้การ JOIN (Nonsequenced Join Query)

พิจารณาคำถามที่ว่า “มีวัวลือทใดที่อยู่ด้วยกันในแต่ละคอก อาจจะในคนละเวลา” จะสามารถตอบได้โดยใช้ SQL ดังต่อไปนี้

```
SELECT L1.หมายเลขลือท, L2.หมายเลขลือท, L1.หมายเลขคอก
FROM วัว AS L1, วัว AS L2
WHERE L1.หมายเลขลือท < L2.หมายเลขลือท
      AND L1.หมายเลขฟาร์ม = L2.หมายเลขฟาร์ม
      AND L1.หมายเลขคอก = L2.หมายเลขคอก
```

จะได้คำตอบดังตารางที่ 4-11

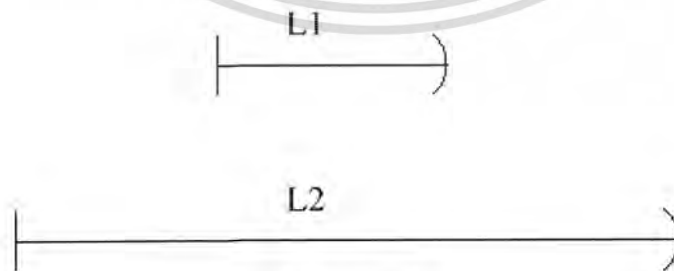
| L1  | L2  | หมายเลขคอก |
|-----|-----|------------|
| 137 | 219 | 1          |
| 137 | 219 | 1          |
| 137 | 374 | 1          |
| 219 | 374 | 1          |
| 219 | 374 | 1          |

ตารางที่ 2-11 คำตอบของตัวอย่างการตอบคำถามแบบนอนซีควเอนซ์โดยใช้การ JOIN

### 2.6.4 การตอบคำถามแบบซีควเอนซ์โดยใช้การ JOIN (Sequenced Join Query)

พิจารณาคำถามที่ว่า “ตั้งแต่ในอดีตจนถึงปัจจุบันมีวัว Lot ใดบ้างที่อยู่ด้วยกันในแต่ละคอก” คำถามแบบนี้จะต้องแบ่งออกเป็น 4 กรณีดังนี้

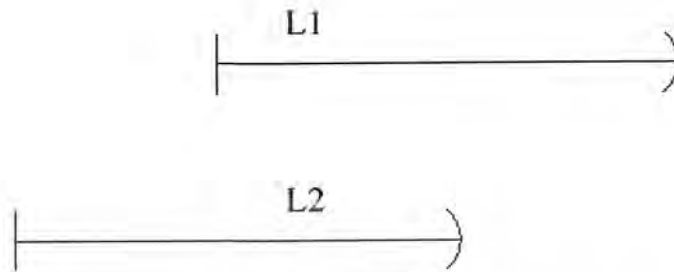
2.6.4.1 ช่วงเวลาของตาราง L1 อยู่ในช่วงเวลาของตาราง L2 แสดงดังรูปที่ 2-12



รูปที่ 2-1 กรณีที่ 1

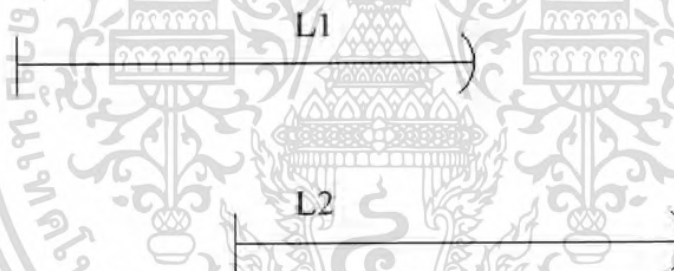
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2.6.4.2 เวลาเริ่มต้นของตาราง L1 มากกว่าเวลาเริ่มต้นของตาราง L2 แต่น้อยกว่าเวลาสิ้นสุดของตาราง L2 และเวลาสิ้นสุดของตาราง L1 มากกว่าเวลาสิ้นสุดของตาราง L2 แสดงดังรูปที่ 2-2



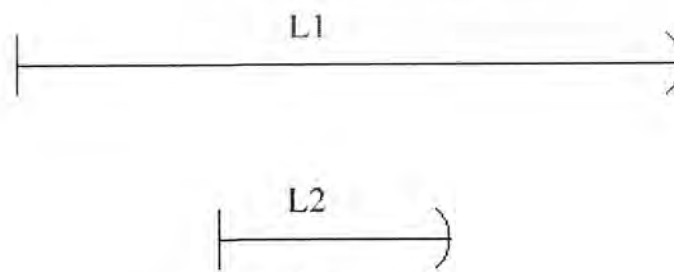
รูปที่ 2-2 กรณีที่ 2

- 2.6.4.3 เวลาเริ่มต้นของตาราง L2 มากกว่าเวลาเริ่มต้นของตาราง L1 แต่น้อยกว่าเวลาสิ้นสุดของตาราง L1 และเวลาสิ้นสุดของตาราง L2 มากกว่าเวลาสิ้นสุดของตาราง L1 แสดงดังรูปที่ 2-3



รูปที่ 2-3 กรณีที่ 3

- 2.6.4.4 ช่วงเวลาของตาราง L2 อยู่ในช่วงเวลาของตาราง L1 แสดงดังรูปที่ 4-4



รูปที่ 2-4 กรณีที่ 4

ดังนั้นจะสามารถตอบได้โดยใช้ SQL ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SELECT L1.หมายเลขลือท, L2.หมายเลขลือท, L1.หมายเลขคอก, L1.เวลาเริ่มต้น, L1.เวลาสิ้นสุด
FROM วั AS L1, วั AS L2
```

```
WHERE L1.หมายเลขลือท < L2.หมายเลขลือท
      AND L1.หมายเลขฟาร์ม = L2.หมายเลขฟาร์ม
      AND L1.หมายเลขคอก = L2.หมายเลขคอก
      AND L2.เวลาเริ่มต้น <= L1.เวลาเริ่มต้น
      AND L1.เวลาสิ้นสุด <= L2.เวลาสิ้นสุด
```

UNION

```
SELECT L1.หมายเลขลือท, L2.หมายเลขลือท, L1.หมายเลขคอก, L1.เวลาเริ่มต้น, L2.เวลาสิ้นสุด
FROM วั AS L1, วั AS L2
```

```
WHERE L1.หมายเลขลือท < L2.หมายเลขลือท
      AND L1.หมายเลขฟาร์ม = L2.หมายเลขฟาร์ม
      AND L1.หมายเลขคอก = L2.หมายเลขคอก
      AND L1.เวลาเริ่มต้น > L2.เวลาเริ่มต้น
      AND L2.เวลาสิ้นสุด < L1.เวลาสิ้นสุด
      AND L1.เวลาเริ่มต้น < L2.เวลาสิ้นสุด
```

UNION

```
SELECT L1.หมายเลขลือท, L2.หมายเลขลือท, L1.หมายเลขคอก, L2.เวลาเริ่มต้น, L1.เวลาสิ้นสุด
FROM วั AS L1, วั AS L2
```

```
WHERE L1.หมายเลขลือท < L2.หมายเลขลือท
      AND L1.หมายเลขฟาร์ม = L2.หมายเลขฟาร์ม
      AND L1.หมายเลขคอก = L2.หมายเลขคอก
      AND L2.เวลาเริ่มต้น > L1.เวลาเริ่มต้น
      AND L1.เวลาสิ้นสุด < L2.เวลาสิ้นสุด
      AND L2.เวลาเริ่มต้น < L1.เวลาสิ้นสุด
```

UNION

```
SELECT L1.หมายเลขลือท, L2.หมายเลขลือท, L1.หมายเลขคอก, L2.เวลาเริ่มต้น, L2.เวลาสิ้นสุด
FROM วั AS L1, วั AS L2
```

```
WHERE L1.หมายเลขลือท < L2.หมายเลขลือท
      AND L1.หมายเลขฟาร์ม = L2.หมายเลขฟาร์ม
      AND L1.หมายเลขคอก = L2.หมายเลขคอก
      AND L2.เวลาเริ่มต้น >= L1.เวลาเริ่มต้น
      AND L2.เวลาสิ้นสุด <= L1.เวลาสิ้นสุด
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้คำตอบดังตารางที่ 2-12

| หมายเลขลือท | หมายเลขลือท | หมายเลขคอก | เวลาเริ่มต้น | เวลาสิ้นสุด |
|-------------|-------------|------------|--------------|-------------|
| 219         | 374         | 1          | 02/25/1998   | 02/28/1998  |
| 219         | 374         | 1          | 03/01/1998   | 03/13/1998  |

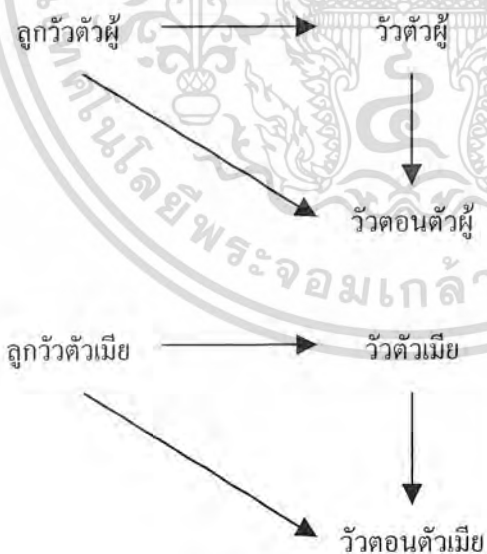
ตารางที่ 2-12 คำตอบของตัวอย่างการตอบคำถามแบบซีควนซ์โดยใช้การ JOIN

## 2.7 การเปลี่ยนแปลงข้อมูลในฐานข้อมูลเชิงเวลา

ในฐานข้อมูลโดยทั่วไปข้อมูล จะต้องมีการเปลี่ยนแปลงแก้ไขเกิดขึ้น อันเนื่องมาจากการ เพิ่มข้อมูล การลบข้อมูล และการแก้ไขข้อมูล ซึ่งในฐานข้อมูลเชิงเวลาการทำสิ่งต่างๆเหล่านี้ จะต้องมียุทธวิธีเฉพาะ ไม่เหมือนกับที่ทำอยู่ในฐานข้อมูลแบบธรรมดาต่างๆ ไปเนื่องจากเวลาเป็นส่วนหนึ่งของข้อมูล

ต่อไปจะเป็นตัวอย่างการเปลี่ยนแปลงของข้อมูลในฐานข้อมูลเชิงเวลา ทั้งการ เพิ่มข้อมูล การลบข้อมูล และการแก้ไขข้อมูล ซึ่งในการเปลี่ยนแปลงข้อมูลอย่างหนึ่ง อาจจะต้องทำการเปลี่ยนแปลงข้อมูลอันอื่นด้วย เช่น ในการลบข้อมูลอาจจะต้องมีการ แก้ไขข้อมูลเก่าด้วย หรือ การเพิ่มข้อมูลเข้าไปใหม่ อาจจะต้องมีการแก้ไขข้อมูลด้วยเช่นเดียวกัน เป็นต้น

รูปที่ 2-6 ต่อไปนี้แสดงถึงการเปลี่ยนแปลงเพศของวัว



รูปที่ 2-5 การเปลี่ยนแปลงเพศของวัว

เพื่อแสดงให้เห็นถึงการเปลี่ยนแปลงตารางวาลิดใหม่โดยใช้ภาษา SQL ของยกตัวอย่างตารางเพศวัวดังตารางที่ 2-13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| หมายเลขลือท | รหัสเพศ | เวลาเริ่มต้น | เวลาสิ้นสุด |
|-------------|---------|--------------|-------------|
| 101         | C       | 01/01/1998   | 03/22/1998  |
| 101         | S       | 03/23/1998   | 12/31/9999  |
| 234         | C       | 02/17/1998   | 10/16/1998  |
| 234         | S       | 10/17/1998   | 12/31/9999  |
| 799         | C       | 03/12/1998   | 12/31/9999  |
| 800         | B       | 01/01/1998   | 12/31/9999  |
| 801         | H       | 01/01/1998   | 12/31/9999  |
| 802         | W       | 01/01/1998   | 12/31/9999  |
| 803         | N       | 01/01/1998   | 12/31/9999  |

ตารางที่ 2-13 ตารางเพศัว

โดยรหัสเพศ C หมายถึงลูกัวตัวผู้ B หมายถึงัวตัวผู้ S หมายถึงัวตอนตัวผู้ H หมายถึงลูกัวตัวเมีย W หมายถึงัวตัวเมีย และ N หมายถึงัวตอนตัวเมีย

#### 2.7.1 การเปลี่ยนแปลงแบบปัจจุบัน (Current Modification)

##### 2.7.1.1 การลบแบบปัจจุบัน (Current Deletion)

พิจารณาการเปลี่ยนแปลงที่ว่า “ัวลือท 234 ย้ายออกจากฟาร์ม” จะสามารถทำได้โดยใช้ SQL ดังต่อไปนี้

```
UPDATE เพศัว
SET เวลาสิ้นสุด = CURRENT_DATE
WHERE หมายเลขลือท = 234
AND เวลาสิ้นสุด >= CURRENT_DATE
AND เวลาเริ่มต้น < CURRENT_DATE

DELETE FROM เพศัว
WHERE หมายเลขลือท = 234
AND เวลาเริ่มต้น > CURRENT_DATE
```

สมมติให้วันนี้เป็นวันที่ 29 เดือนกรกฎาคม ปี ค.ศ. 1998 ดังนั้นจะได้ผลดังตารางที่ 2-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| หมายเลขลือท | รหัสเพศ | เวลาเริ่มต้น | เวลาสิ้นสุด |
|-------------|---------|--------------|-------------|
| 101         | C       | 01/01/1998   | 03/22/1998  |
| 101         | S       | 03/23/1998   | 12/31/9999  |
| 234         | C       | 02/17/1998   | 07/28/1998  |
| 799         | C       | 03/12/1998   | 12/31/9999  |
| 800         | B       | 01/01/1998   | 12/31/9999  |
| 801         | H       | 01/01/1998   | 12/31/9999  |
| 802         | W       | 01/01/1998   | 12/31/9999  |
| 803         | N       | 01/01/1998   | 12/31/9999  |

ตารางที่ 2-14 ผลของตัวอย่างการลบแบบปัจจุบัน

### 2.7.1.2 การแก้ไขแบบปัจจุบัน (Current Update)

พิจารณาการเปลี่ยนแปลงที่ว่า “เปลี่ยนววลือท 799 เป็นววลือนตัวผู้” การเปลี่ยนแปลงแบบนี้จะต้องแบ่งออกเป็น 3 กรณี แสดงดังรูปที่ 2-6 ดังต่อไปนี้

กรณีที่ 1 :

ผล : ไม่เปลี่ยนแปลง

กรณีที่ 2 :

ผล : เปลี่ยนเวลาสิ้นสุด

เป็นวันปัจจุบันและ

เพิ่มรหัสเพศใหม่

เข้าไป

กรณีที่ 3 :

ผล : เปลี่ยนรหัสเพศใหม่

ปัจจุบัน

รูปที่ 2-6 กรณีของการแก้ไขแบบปัจจุบัน

ดังนั้นจะสามารถทำได้โดยใช้ SQL ดังต่อไปนี้

INSERT INTO เพศว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

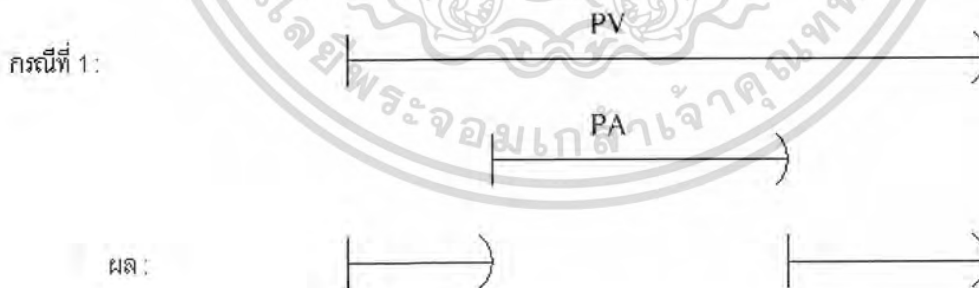
```

SELECT หมายเลขล๊อต, 'S', CURRENT_DATE, เวลาสิ้นสุด
FROM เพศวัว
WHERE หมายเลขล๊อต = 799
      AND เวลาเริ่มต้น <= CURRENT_DATE
      AND เวลาสิ้นสุด > CURRENT_DATE
UPDATE เพศวัว
SET เวลาสิ้นสุด = CURRENT_DATE
WHERE หมายเลขล๊อต = 799
      AND รหัสเพศ <> 'S'
      AND เวลาเริ่มต้น < CURRENT_DATE
      AND เวลาสิ้นสุด > CURRENT_DATE
UPDATE เพศวัว
SET รหัสเพศ = 'S'
WHERE หมายเลขล๊อต = 799 AND เวลาเริ่มต้น >= CURRENT_DATE
    
```

2.7.2 การเปลี่ยนแปลงแบบซีควนซ์ (Sequenced Modification)

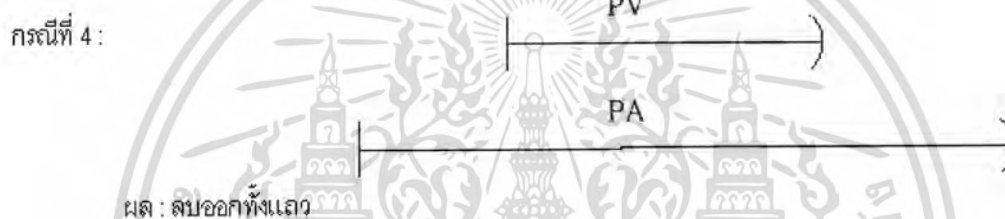
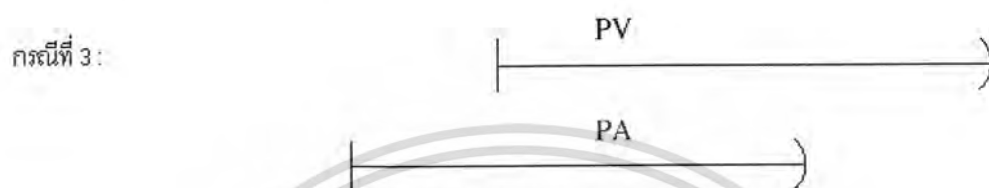
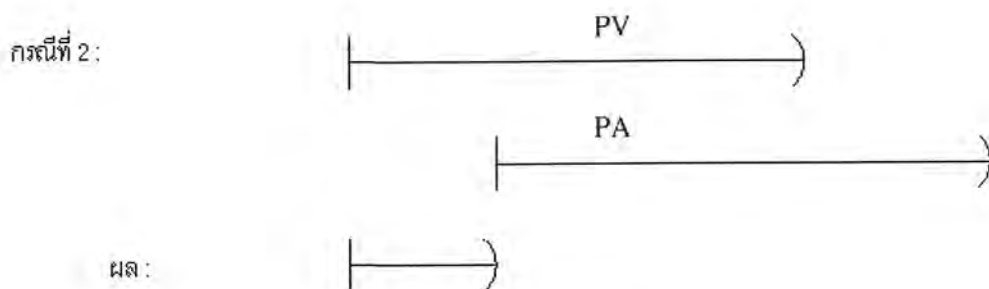
2.7.2.1 การลบแบบซีควนซ์ (Sequenced Deletion)

พิจารณาการเปลี่ยนแปลงที่ว่า “วัวล๊อต 234 ย้ายออกจากฟาร์มใน 3 สัปดาห์แรกของเดือนตุลาคม ซึ่งในช่วงเวลานั้นได้มีการวางแผนที่จะเปลี่ยนวัวล๊อต 234 นี้เป็นวัวตอนตัวผู้” การเปลี่ยนแปลงแบบนี้จะต้องแบ่งออกเป็น 4 กรณี แสดงดังรูปที่ 2-7 ดังต่อไปนี้



รูปที่ 2-7 กรณีของการลบแบบซีควนซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### รูปที่ 2-7 (ต่อ) กรณีของการลบแบบซีเควนซ์

โดย PV คือ ช่วงเวลาของข้อมูลเดิม (Period Of Validity) และ PA คือ ช่วงเวลาที่ต้องการเปลี่ยนแปลง (Period Of Applicability) ดังนั้นจะสามารถทำได้โดยใช้ SQL ดังต่อไปนี้

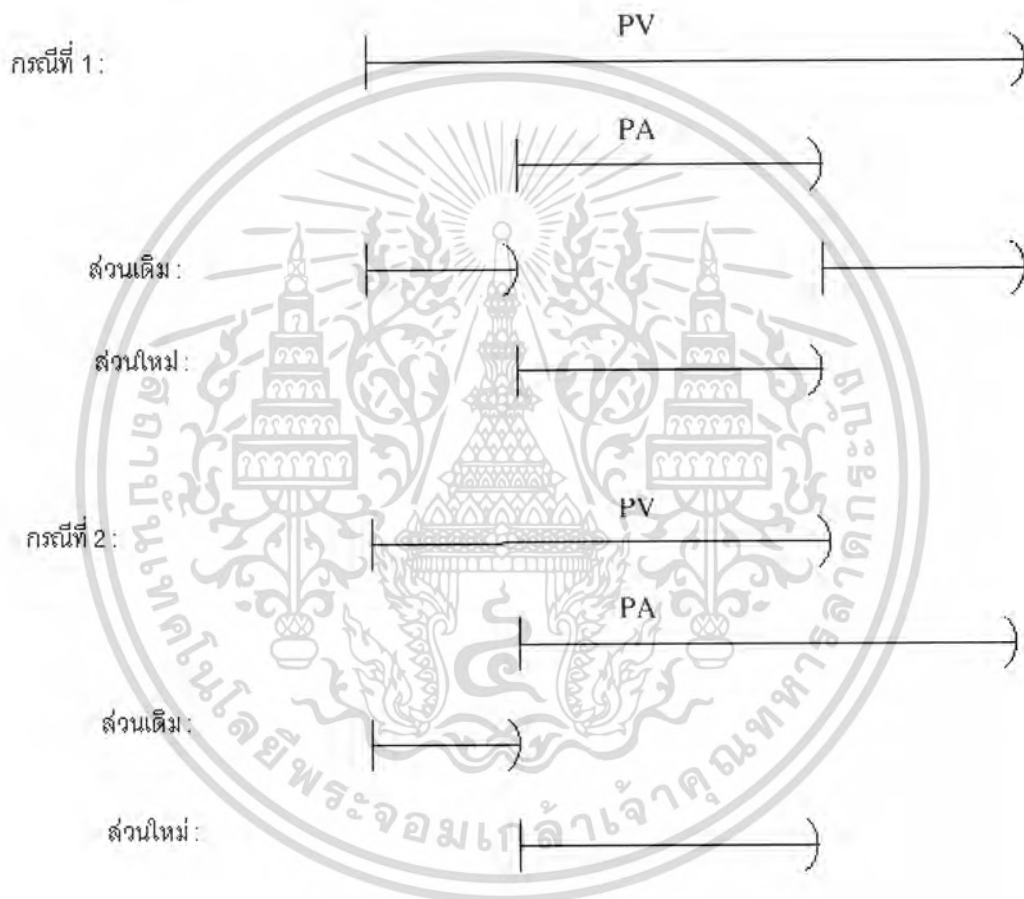
```
INSERT INTO เพศวัว
SELECT หมายเลขลือท, รหัสเพศ, '10/22/1998', เวลาสิ้นสุด
FROM เพศวัว
WHERE หมายเลขลือท = 234
      AND เวลาเริ่มต้น <= '10/01/1998' AND เวลาสิ้นสุด > '10/22/1998'
UPDATE เพศวัว
SET เวลาสิ้นสุด = '10/01/1998'
WHERE หมายเลขลือท = 234
      AND เวลาเริ่มต้น < '10/01/1998'
      AND เวลาสิ้นสุด >= '10/01/1998'
UPDATE เพศวัว
SET เวลาเริ่มต้น = '10/22/1998'
WHERE หมายเลขลือท = 234
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AND เวลาเริ่มต้น < '10/22/1998' AND เวลาสิ้นสุด >= '10/22/1998'  
 DELETE FROM เพศัว  
 WHERE หมายเลขลือท = 234  
 AND เวลาเริ่มต้น >= '10/01/1998' AND เวลาสิ้นสุด <= '10/22/1998'

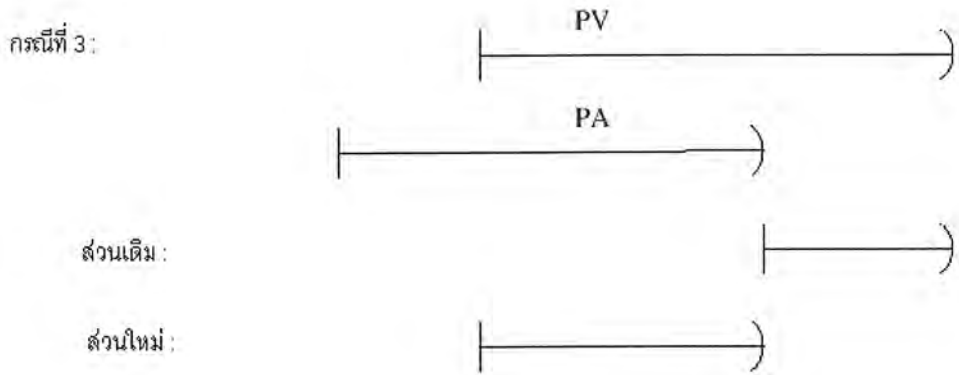
### 2.7.2.2 การแก้ไขแบบซีเควนซ์ (Sequenced Update)

พิจารณาการเปลี่ยนแปลงที่ว่า “เปลี่ยนววลือท 799 เป็นววลือทตัวผู้เฉพาะในเดือนมีนาคม” การเปลี่ยนแปลงแบบนี้จะต้องแบ่งออกเป็น 4 กรณี แสดงดังรูปที่ 2-8 ดังต่อไปนี้



รูปที่ 2-8 กรณีของการแก้ไขแบบซีเควนซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ผล : แก้ไขทั้งหมด

รูปที่ 2-8 (ต่อ) กรณีของการแก้ไขแบบซีเควนซ์

ดังนั้นจะสามารถทำได้โดยใช้ SQL ดังต่อไปนี้

```

INSERT INTO เพศวัว
SELECT หมายเลขลือท, รหัสเพศ, เวลาเริ่มต้น, '03/01/1998'
FROM เพศวัว
WHERE หมายเลขลือท = 799
      AND เวลาเริ่มต้น < '03/01/1998' AND เวลาสิ้นสุด > '03/01/1998'
INSERT INTO เพศวัว
SELECT หมายเลขลือท, รหัสเพศ, '04/01/1998', เวลาสิ้นสุด
FROM เพศวัว
WHERE หมายเลขลือท = 799
      AND เวลาเริ่มต้น < '04/01/1998' AND เวลาสิ้นสุด > '04/01/1998'
UPDATE เพศวัว
SET รหัสเพศ = 'S'
WHERE หมายเลขลือท = 799
      AND เวลาเริ่มต้น < '04/01/1998' AND เวลาสิ้นสุด > '03/01/1998'
UPDATE เพศวัว
SET เวลาเริ่มต้น = '03/01/1998'
WHERE หมายเลขลือท = 799
      AND เวลาเริ่มต้น < '03/01/1998' AND เวลาสิ้นสุด > '03/01/1998'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UPDATE เพศวีว

SET เวลาสิ้นสุด = '04/01/1998'

WHERE หมายเลขลือท = 799

AND เวลาเริ่มต้น < '04/01/1998' AND เวลาสิ้นสุด > '04/01/1998'

### 2.7.3 การเปลี่ยนแปลงแบบอนซีควเอนซ์ (Nonsequenced Modification)

พิจารณาการเปลี่ยนแปลงที่ว่า “ลบข้อมูลของวีวลือท 234 ที่มีช่วงเวลามากกว่า 3 เดือน” จะสามารถทำได้โดยใช้ SQL ดังต่อไปนี้

DELETE FROM เพศวีว

WHERE หมายเลขลือท = 234

AND (เวลาสิ้นสุด - เวลาเริ่มต้น MONTH) > INTERVAL '3' MONTH



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### แนวความคิดเชิงวัตถุ

#### 3.1 คุณสมบัติของแนวความคิดแบบเชิงวัตถุ

ในอดีตที่ผ่านมาการเขียนโปรแกรม จะเป็นแบบสั่งให้ คอมพิวเตอร์ทำงานเป็น ลำดับของคำสั่ง ( instruction ) ซึ่งในสมัยนั้นเป็นที่นิยมอย่างมาก เนื่องจากเขียนโปรแกรมได้ง่ายและรวดเร็ว และใช้วิธีการเก็บรวบรวมข้อมูลต่างๆ เป็นแฟ้มข้อมูล ( File ) หรือ ฐานข้อมูล ( Database ) แยกกันกับโปรแกรมในระบบงาน ถ้าโปรแกรมมีขนาดใหญ่จำเป็นต้องมีการแบ่งงานให้กับโปรแกรมเมอร์หลายคนทำการเขียนโปรแกรม และจะต้องมีสร้าง function ที่ซ้ำกันเกิดขึ้น ทำให้ขาดประสิทธิภาพ และเมื่อได้ทำการใช้งานไปนานๆ จะพบว่า การเขียนโปรแกรมในลักษณะนี้มีข้อจำกัดอยู่มาก เช่น ในเรื่องของการนำโปรแกรมมาพัฒนาให้ทันสมัยนั้นทำได้ยาก เพราะต้องทำการศึกษาโปรแกรมเก่าให้เข้าใจอย่างแท้จริงก่อนลงมือแก้ไข และไม่มีความปลอดภัยของข้อมูล เพราะทุกโพรเซส ( Process ) สามารถนำข้อมูลนั้น ไปใช้งานได้ตามต้องการ

ทำให้เกิดแนวความคิดการเขียนโปรแกรมเชิงวัตถุ ( Object - Oriented Programming ) ขึ้นมา โดยเป็นวิธีใหม่ในการพัฒนาซอฟต์แวร์ ( Software ) เพื่อเพิ่มประสิทธิภาพในการผลิตซอฟต์แวร์ วิธีนี้ทำให้การออกแบบและการเขียนซอฟต์แวร์มีความใกล้ชิดกันมากขึ้น และจะกำหนดสิ่งต่างๆ ที่อยู่ภายในขอบเขตของระบบงานเป็นออบเจกต์ ( Object ) ซึ่งในออบเจกต์จะประกอบด้วยข้อมูล ( Data ) และเมธอด ( Method ) และการเข้าถึงข้อมูลนั้นได้จะต้องผ่านเมธอดที่กำหนดไว้ ( Encapsulation ) เมธอดเปรียบเทียบกับทำงานเองเดียวกันกับโพรซีเจอร์ ( Procedure ) ในโปรแกรมระบบงานแบบเดิมนั้นเอง การเรียกใช้เมธอดหรือการสั่งให้เกิดการกระทำตามที่กำหนดไว้ในเมธอดทำได้โดยการส่งเมสเสจ ( Message ) ไปยังออบเจกต์นั้นๆ ดังนั้นหากมีการเปลี่ยนแปลงหน้าที่การทำงาน ( Function ) ใดๆ ของออบเจกต์นั้น จะไม่มีผลกระทบต่อเมสเสจที่ใช้ติดต่อมายังออบเจกต์นั้น ซึ่งนับว่าเป็นผลดีอย่างมากกับระบบงานเพราะ โดยปกติแล้วพบว่าระบบงานมักจะมีการเปลี่ยนแปลงวิธีการปฏิบัติอยู่เสมอๆ เมธอดสามารถใช้อธิบายถึงพฤติกรรมของออบเจกต์แต่ละออบเจกต์ได้ ออบเจกต์ที่มีพฤติกรรมแบบเดียวกันสามารถจัดรวมกลุ่มเป็นกลุ่มเดียวกันได้เรียกว่า คลาส ( Class ) ในระหว่างคลาสสามารถที่จะถ่ายทอดคุณสมบัติ ( Inheritance ) จากคลาสหนึ่งไปยังอีกคลาสหนึ่งได้ โดยเรียกคลาสที่ถ่ายทอดคุณสมบัติว่า ซุปเปอร์คลาส ( Superclass ) และเรียกคลาสที่ได้รับการถ่ายทอดคุณสมบัติว่า ซับคลาส ( Subclass ) ซึ่งทำให้การบำรุงรักษาระบบงานทำได้ง่าย และสามารถขจัดปัญหาต่างๆ ในการเขียนโปรแกรมแบบโพรซีเจอร์อลได้อีกด้วย โดย Object - Oriented Language โปรแกรมแรกซึ่งออกมาในปี ค.ศ. 1967 คือโปรแกรม Simula ภายในโปรแกรมมีการกำหนด Object , Class , inheritance และ อื่นๆ ต่อมาได้มีโปรแกรม SmallTalk ซึ่งเริ่มมีการใช้ graphic เข้ามาช่วยในโปรแกรมมากขึ้น จนถึงปัจจุบันได้มีโปรแกรมที่เข้ามาสนับสนุนภาษาทางด้าน Object Oriented มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.2 คุณสมบัติของ Object-Oriented

### 3.2.1 Encapsulation/Information hiding

ทฤษฎีนี้ถูกคิดค้นโดย James Rumbaugh ซึ่งหมายถึง การแยกลักษณะภายนอก (Interface) ซึ่งสามารถติดต่อกับ Object อื่นๆ ออกจากส่วนที่ implement ภายในของ Object ซึ่งจะถูกใช้ได้เฉพาะตัว Object นั้นๆ มีข้อดีคือ เพิ่มความสามารถในการปกป้องข้อมูลโดยไม่ให้ Object อื่นเข้ามาทำการเปลี่ยนแปลงข้อมูลก่อนได้รับอนุญาต ทั้งนี้ตัว Object ควรรู้และทำการเปลี่ยนค่าด้วยตนเองซึ่งเป็นผลให้ maintain object ง่ายขึ้น

### 3.2.2 Inheritance

คือหลักการในการที่ Object ใดๆตัวใน generalized collection ได้มีการใช้ข้อมูล และพฤติกรรม (behavior) ร่วมกันซึ่งจะมีความสัมพันธ์แบบ is-a relationship โดยเรียก class ที่อยู่เหนือกว่าว่า super class ซึ่งจะถ่ายทอดคุณสมบัติทั้ง Attribute และ method มายัง class ที่ต่ำกว่าเรียกว่า subclass ซึ่งจะมี Attribute และ method เพิ่มเติมจาก super class มีข้อดีคือ

- เพิ่ม consistency เพียงแค่เปลี่ยน Method หรือ Attribute ที่ Super class ซึ่งเป็นผลให้ค่าที่ Subclass เปลี่ยนไปด้วย
- เป็นการส่งเสริมการนำ Object กลับมาใช้ใหม่

จากคุณสมบัติที่มีมาจากการสืบทอดนี้ทำให้เราได้ประโยชน์จากการที่เราสามารถเพิ่มศักยภาพทางด้านการนำกลับมาใช้ใหม่ (Reusability) และการขยายเพิ่มเติมคุณสมบัติ (Extensibility) ได้

### 3.2.3 Polymorphism

สำหรับความหมายของ Polymorphism โดยรวมแล้วก็คือ รูปแบบหนึ่งของการสืบทอดคุณสมบัติ โดยจะแบ่งออกได้เป็นสองลักษณะใหญ่ก็คือ ลักษณะแบบ Overloading จะหมายถึงการที่ Method สองตัวใดๆ ใน Class เดียวกันที่มีชื่อเหมือนกัน แต่รูปแบบการใช้งานที่แตกต่างกัน และแบบ Overriding ซึ่งหมายถึงความสามารถของโปรแกรมที่อ้างอิงถึง Attribute หรือ Method ที่มีชื่อเหมือนกัน แต่มีการเปลี่ยนแปลงความหมายไปเมื่อมีการสืบทอดค่าคุณสมบัตินั้นมาจาก Class ต้น

โพลิมอร์ฟิซึม แบ่งออกได้เป็น 2 ลักษณะ คือ

1. ช่วงการคอมไพล์โปรแกรม ( Compile time ) การตรวจสอบชนิดออบเจกต์ของฟังก์ชันที่ถูกเรียกมาทำงาน รวมทั้งเตรียมข้อมูลบางอย่างที่จำเป็นจะทำขณะคอมไพล์โปรแกรม การตัดสินใจใช้ฟังก์ชันใดขึ้นอยู่กับคุณสมบัติของพารามิเตอร์ของฟังก์ชันนั้น
2. ช่วงการรันโปรแกรม ( Run time ) การตรวจสอบนั้นจะทำขณะที่โปรแกรมกำลังทำงาน จะทำการตัดสินใจเลือกฟังก์ชันใดฟังก์ชันหนึ่งที่มีชื่อซ้ำกันในแต่ละคลาสมาใช้งาน

### 3.2.4 Abstraction

การแยกแยะเอกลักษณ์เป็นวิธีการหนึ่งที่มนุษย์ใช้ในการมองสิ่งใดๆ เพื่อลดความซับซ้อนของสิ่งนั้นๆ โดยแยกเอาเฉพาะสิ่งที่ผู้มองสนใจออกมา ซึ่งแน่นอนว่าถ้าผู้มองระบบเป็นคนละคนกันก็จะแยกแยะออกมาได้ต่างกัน การแยกแยะเอกลักษณ์เป็นการแสดงคุณลักษณะที่สำคัญของออบเจกต์ ซึ่งเป็นคุณลักษณะที่ทำให้ออบเจกต์นั้นๆ แตกต่างจากออบเจกต์อื่นๆ เช่น ออบเจกต์รถยนต์ ถ้าเป็นการแยกแยะเอกลักษณ์ของนักเดินทางก็จะมองว่ารถยนต์นั้นต้องพาผู้โดยสารไปยังปลายทางได้สะดวกรวดเร็ว แต่ถ้าเป็นการแยกแยะเอกลักษณ์ของช่างเครื่องก็จะมองว่ารถยนต์ประกอบด้วยเครื่องยนต์ต่างๆ มากมายที่ทำงานร่วมกัน

### 3.2.5 Persistence

การคงสถานะ (Persistence) เป็นคุณสมบัติของวัตถุคือ วัตถุจะยังคงสามารถอยู่ได้ถึงแม้ว่าจะเวลาล่วงเลยไปแล้วก็ตาม ตัวอย่างเช่น วัตถุยังสามารถอยู่ต่อไปได้ ถึงแม้ผู้สร้างวัตถุขึ้นมาจะหยุดและปิดการทำงานไปแล้วก็ตาม และรวมทั้งพื้นที่ของวัตถุที่อยู่ด้วยเช่น ถ้าวัตถุย้ายที่จากหน่วยความจำที่เคยอยู่ เมื่อตอนถูกสร้างขึ้นมา วัตถุในซอฟต์แวร์นั้นจะอยู่ในพื้นที่ของหน่วยความจำและช่วงเวลาที่กำหนดเท่านั้น

### 3.2.6 Concurrency

การทำงานพร้อมกันนั้น เกี่ยวข้องกับการแยกแยะเอกลักษณ์ของโพรเซส และการชิงโครโนเซชัน สามารถแบ่งวัตถุได้เป็น 2 ชนิดคือวัตถุที่สามารถทำงานด้วยตัวเอง (active object) และวัตถุที่ไม่สามารถทำงานด้วยตัวเอง (passive object) ต้องใช้บริการจากวัตถุเอกทิฟ ในระบบที่ออกแบบด้วยวิธีเชิงวัตถุสามารถกำหนดให้วัตถุที่สามารถทำงานแบบเอกทิฟทำงานพร้อมๆ กันได้ โดยวัตถุที่ทำงานในระบบอาจจะเป็นอิสระต่อกัน หรืออาจจะต้องมีการสื่อสารระหว่างกันก็ได้

## 3.3 หลักการพื้นฐานของ Object-Oriented

- **Object** : Object เป็นการเปรียบเสมือนกล่องๆ หนึ่ง ซึ่งข้างในกล่องมี Data ( instance variable ) และ Method รวมกันอยู่ภายใน และ Object นี้จะทำการรับและการส่ง message ระหว่าง Object เพื่อทำการใช้งาน Data และ Method ที่อยู่ภายใน โดยการรับและการส่ง message นี้จึงเปรียบเสมือน interface ของ Object นั้นๆ ด้วย โดย Object นี้ถือได้ว่าเป็นคุณสมบัติสำคัญในการทำความเข้าใจการเขียนโปรแกรมแบบ Object Oriented
- **Class** : คลาสเป็นการจัดกลุ่มของ Object ตามคุณสมบัติที่เหมือนกัน นำมารวมกันเป็นคลาส และผู้ใช้สามารถติดต่อกับคลาสได้โดยผ่านทางเมธอดภายในคลาส หรือเมธอดในซูเปอร์คลาส เท่านั้น ( superclass คือ super class ) โดยปกติคลาสจะถูกออกแบบให้มีความสัมพันธ์กับคลาสอื่นๆ ซึ่งคลาสหนึ่งคลาสสามารถเป็นต้นแบบให้แก่คลาสอื่นๆ ได้หลายคลาส ( Inheritance )
- **Method** : เป็นการทำงาน ( Operation ) หรือเป็นวิธีการกระทำที่สามารถทำงานกับออบเจกต์ได้ แต่ละคลาสจะมีเมธอดของตัวเอง โดยการทำงาน จะเริ่มจากการส่ง message ไปยัง Object ที่ต้องการ ( Receiver Object ) เพื่อเรียกใช้เมธอดที่อยู่ภายใน Object นั้น หรือรอรับ message ที่ส่งมาจาก Object อื่นๆ เพื่อมาเรียกใช้เมธอดของตัวเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Message** : เป็นตัวกลางในการติดต่อสื่อสารระหว่างแต่ละออบเจ็กต์ โดยข้อมูลที่จะถูกส่งไปกับ message ด้วยคือ
    - ออบเจ็กต์ที่ message ที่ระบุ
    - ชื่อของเมธอดที่จะทำการติดต่อ
    - พารามิเตอร์ต่างๆที่เมธอดต้องการ
  - **Encapsulation** : คือ การรวมกันของโครงสร้างข้อมูล ( Data Structure ) กับฟังก์ชัน ( Method , Action ) เกิดเป็นวัตถุใหม่ที่มีความสามารถในการซ่อนข้อมูลจากระบบภายนอกได้ ( มีลักษณะเหมือนเป็นยาเม็ดแคปซูล ซึ่งจะมองไม่เห็นตัวยากภายในรู้แต่เพียงว่ายามีตัวยานี้ใช้รักษาโรคอะไรเท่านั้น ) ทำให้ข้อมูลมีความมั่นคงขึ้น ซึ่งวิธีการป้องกันนั้นจะอยู่ที่ชนิดของการประกาศข้อมูลภายในคลาส ซึ่งจะมียู่ 3 ระดับคือ
    - การประกาศข้อมูลแบบที่ใช้ภายในตัวเอง จะมีผลทำให้ไม่มีคำสั่งใดที่จะอ้างถึงข้อมูลภายในคลาสนั้นได้เลย นอกจากเมธอด ภายในคลาสนั้นๆเอง ซึ่งตามปกติแล้วจะมีอย่างน้อยหนึ่งเมธอดที่ไม่ได้ประกาศให้เป็นข้อมูลชนิดนี้ เพื่อที่จะได้ใช้ในการเข้าถึงข้อมูลภายในคลาสได้
    - การประกาศข้อมูลแบบสาธารณะ คือจะมีผลทำให้คลาสอื่นๆ หรือเมธอดอื่นๆ สามารถอ้างถึงข้อมูลภายในคลาสนี้ได้อย่างอิสระ
    - การประกาศข้อมูลแบบสงวนไว้ให้คลาสลูก จะมีผลทำให้มีเฉพาะเมธอดภายในของคลาสนั้นและคลาสที่สืบทอด ( Inheritance ) เท่านั้นที่สามารถอ้างถึงข้อมูลภายในคลาสได้
  - **Inheritance** : คือ การสร้างคลาสใหม่ขึ้นมา โดยมีการสืบทอดคุณสมบัติพื้นฐานที่มาจากคลาสเดิม แต่จะมี ข้อมูล หรือ เมธอด เพิ่มขึ้นมาจากคลาสเดิมด้วยก็สามารถทำได้ ถ้าคลาส B เป็นซับคลาสของคลาส A ออบเจ็กต์ที่เป็นอินสแตนซ์ของคลาส B จะมีคุณสมบัติพิเศษกว่าอินสแตนซ์ของคลาส A แต่อย่างน้อยที่สุด จะต้องมียุทธศาสตร์เหมือนอินสแตนซ์ของคลาส A การถ่ายทอดคุณสมบัติ นี้จะรวมถึง data และ method ของคลาส A และคลาส B จะถูกเรียกว่าเป็น ซุปเปอร์คลาสของคลาส B การสืบทอดแบบนี้ นับได้ว่าเป็นคุณสมบัติเด่นของการเขียนโปรแกรมแบบ Object Oriented เพราะทำให้เราสามารถนำโค้ดเดิมมาใช้ได้อีกโดยไม่ต้องมีการเขียนโปรแกรมที่เขียนไว้แล้วขึ้นมาใหม่อีก
  - **Polymorphism** : คือ การที่เราสร้างเมธอดที่เหมือนกัน ไปในออบเจ็กต์ที่ต่างกัน แต่ละออบเจ็กต์จะตอบสนองออกมาไม่เหมือนกัน ตามแต่ชนิดและหน้าที่ของออบเจ็กต์ ความสามารถใช้เมธอดเหมือนกัน สำหรับการกระทำที่เหมือนกัน ไปยังออบเจ็กต์ต่างชนิดกัน มีลักษณะเหมือนกับการที่มนุษย์คิดในการแก้ปัญหาหนึ่ง ๆ
- ตัวอย่าง การ polymorphism
- AB + CD ได้ผลลัพธ์ ABCD ( concat )
  - 5 + 3 ได้ผลลัพธ์ 8 ( add )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้ง 2 แบบ เป็นการส่งเมสเสจ '+' เข้าไปยังออบเจ็กต์ ภายในคลาส String และ Integer ตามลำดับ

- **Dynamic Binding** : คือ การนำโปรแกรมย่อยๆ มาประกอบให้ใช้งานได้ ในขณะที่ run time โดยในขณะที่ Compile time นั้นจะเก็บโปรแกรมในรูปแบบของ Class ต่างๆ ไว้ เพื่อไม่ให้เกิดความยุ่งยาก ซับซ้อน ต่อจากนั้น เมื่อนำมาใช้ในขณะที่ run time เมื่อมีการเรียกใช้ Class นั้นๆ จะทำการนำ Class นั้นๆ มาไว้ใน ส่วนของโปรแกรม และเมื่อใช้งานเสร็จแล้วจะถูกลบออกจากหน่วยความจำ
- **Override** : คือ การที่เราสร้าง Subclass ขึ้นมาใหม่ และมีการสร้าง Method ที่ซ้ำกับ Method ที่เป็นของ Superclass
- **Overload** : คือ การที่เราสร้าง Method ชื่อเหมือนกัน แต่รับ Parameter ต่างกัน ภายใน Class เดียวกัน

### 3.4 ความสัมพันธ์ระหว่าง Object

#### 3.4.1 Association

เป็นการแสดงให้เห็นถึงความสัมพันธ์ระหว่างโครงสร้างของ Object Oriented ว่ามีความสัมพันธ์กันหรือไม่ โดยจะมีการแสดงการเชื่อมต่อระหว่าง Object Oriented ที่มีความสัมพันธ์แบบถาวร โดยมีโครงสร้าง ดังนี้

- **constraint** คือการแสดงข้อมูลที่เกี่ยวกับความสัมพันธ์ระหว่าง Objects
- **Link** คือ การเชื่อมต่อระหว่าง Objects ที่มีความสัมพันธ์แบบ ถาวร ซึ่งเป็น instance ของ Association

#### 3.4.2 Aggregation

ในบางครั้งเราอาจถือได้ว่า Aggregation เป็นความสัมพันธ์แบบพิเศษของ Association โดยจะมีความสัมพันธ์ในรูปแบบ Whole/Part หรือแบบ a-part-of หรืออาจเรียกอีกแบบหนึ่งว่ามีความสัมพันธ์แบบ has-a-relationship คือ class A จำเป็นต้องมีการเรียกใช้ class B ในการทำงาน ดังนั้น class A จะขาด class B ไม่ได้ แต่ class B สามารถขาด class A ได้จึงเรียกได้ว่า class B มีความสัมพันธ์เป็นแบบ Aggregation กับ class A

#### 3.4.3 Generalization

เป็นความสัมพันธ์ในการจัดกลุ่มสิ่งๆ ที่เหมือนกันระหว่าง Object หรือ Class ขึ้นมาเป็นอีก Object หนึ่งโดยพิจารณาจากข้อมูล (structure) หรือ พฤติกรรม (behavior) ในกลุ่มของ Object นั้น จะมีลักษณะเป็นลักษณะทั่วไป ไม่ใช่เฉพาะเจาะจง

#### 3.4.4 Depends On

เป็นความสัมพันธ์ที่ชี้ให้เห็นถึง source หรือ client ขึ้นอยู่กับ destination หรือ supplier เช่นใน Class หนึ่งสามารถมี Method ที่ขึ้นอยู่กับ Class อื่น เวลาทำงานจะมีการอ้างถึง Class นั้นด้วย

### 3.5 ข้อดีของภาษาทางด้าน Object – Oriented

- **Security** : ในภาษาทาง Object Oriented มีความสามารถในการทำ ( Encapsulation / Information hiding ) เนื่องจาก Object Oriented จะเก็บข้อมูลไว้ในออบเจกต์ซึ่งจะถูกจัดการโดยเมธอดที่เหมาะสมเท่านั้น ดังนั้นจึงทำให้ข้อมูลมีความปลอดภัยสูงกว่าภาษาแบบ procedural
- **Reusability** : Object Oriented สามารถนำคอมโพเนนท์ที่สร้างไว้แล้วกลับมาใช้งานได้อีก โดยไม่จำเป็นต้องทำการเขียนใหม่เลย
- **Portability** : ระบบที่ถูกออกแบบมาให้ใช้ได้บนสภาพแวดล้อมหนึ่งๆ สามารถที่จะปรับปรุงแก้ไขเพียงเล็กน้อยหรืออาจไม่ต้องปรับปรุงเลย ในการนำไปใช้กับอีกสภาพแวดล้อมอื่นๆ
- **Integrity** : เนื่องจาก Object Oriented มีคุณสมบัติในการทำ Encapsulation จึงทำให้เกิดความผิดพลาดของข้อมูลได้ยาก ทำให้มีความน่าเชื่อถือสูง
- **Flexibility** : ด้วยคุณสมบัติในด้านการสืบทอดความสัมพันธ์ ( Inheritance ) ของ Object Oriented จึงทำให้มีความยืดหยุ่นในการแก้ไขได้ง่าย คือ สามารถเพิ่มหรือลด ความสามารถบางส่วนให้แก่แต่ละคลาสย่อยได้ โดยอาจเพิ่ม หรือลดที่คลาสแม่ ( Superclass ) เท่านั้น ในทุกคลาสย่อยก็จะถูกเพิ่มเติม หรือ ลดความสามารถนั้นๆออกไปด้วย
- **Understandability** : ด้วยคุณลักษณะ Abstract ของ Object Oriented ทำให้ภาพรวมของตัวโปรแกรมตรงกับความเข้าใจของผู้ใช้ จึงทำให้ผู้ใช้สามารถเข้าใจการทำงานของระบบได้โดยง่าย
- **Maintainability** : เนื่องด้วยความสามารถทางด้านการสืบทอดความสัมพันธ์ ( Inheritance ) ของ Object Oriented จึงทำให้ง่ายต่อการปรับปรุง เมื่อมีความต้องการเพิ่มเติม specification ของระบบและด้วยข้อดีในด้าน Flexibility จึงทำให้เป็นการง่ายต่อการบำรุงรักษา
- **Easy to Development** : ในการพัฒนาทางด้าน Object Oriented นั้นเราไม่จำเป็นต้องสนใจโค้ดของโปรแกรมมากนัก เราเพียงแค่ทราบรูปแบบในการสื่อสาร ( message ) ของแต่ละออบเจกต์ และเอาทันที ที่ได้ในแต่ละแบบของการสื่อสารนั้นๆ ก็เพียงพอที่จะทำให้เราสามารถพัฒนาโปรแกรมในแบบขนานได้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### ฐานข้อมูลเชิงวัตถุ

#### 4.1 ระบบจัดการฐานข้อมูลเชิงวัตถุ

มีลักษณะการเก็บข้อมูลและการเรียกใช้ข้อมูลที่มีรูปแบบแตกต่างจากระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database) คือจะเป็นการมองข้อมูลที่ถูกจัดเก็บในรูปของวัตถุ(Object)ซึ่งวัตถุจะประกอบด้วยแอตทริบิวต์ และเมธอด

สำหรับลักษณะโดยรวมที่สำคัญๆ ของระบบฐานข้อมูลเชิงวัตถุสามารถอธิบายโดยสังเขปได้โดยแบ่งเป็นหัวข้อย่อยๆ ได้ดังนี้

##### 4.1.1 Object Identity

จะสนับสนุนไอดีเนนทีตี(Identity) ของ Object Instance โดยไม่สนใจว่าค่าที่เป็นลักษณะเฉพาะ (Attribute)ของแต่ละ Object จะเป็นเช่นไร และแต่ละ Object ต้องมีค่าของ Object Identity เพื่อไว้ใช้อ้างอิงไปยัง Object ตัวอื่นในแอปพลิเคชัน นอกจากนี้ Object Identity ยังต้องระบุไว้ให้ชัดเจนในแอปพลิเคชันด้วย

สำหรับในระบบฐานข้อมูลที่เป็นแบบเน้นการแสดงค่า (Value-Oriented) จะใช้ค่าๆ หนึ่งที่อยู่ในฐานข้อมูลเป็นคีย์หลัก (Key) ในการแทนถึงวัตถุใดวัตถุหนึ่ง ปัญหาที่พบเมื่อมีการแทนถึงวัตถุในลักษณะนี้ก็คือ ในกรณีที่คีย์หลักที่ใช้ในการแทนถึงวัตถุหรือสิ่งของใดๆ นั้นเป็นค่าที่มีการเปลี่ยนแปลงด้วย จะทำให้เกิดปัญหาของการแทนค่าถึงวัตถุตัวใดตัวหนึ่งเนื่องจากค่าที่ใช้ในการอ้างอิงนั้นเกิดการเปลี่ยนแปลงไปจากเดิม และต้องแก้ปัญหาคือการแก้ไขคีย์หลักที่ใช้ทำการอ้างอิงนี้ทุกตัวเมื่อมีการเปลี่ยนแปลงเกิดขึ้น เพื่อให้การอ้างอิงถึงค่าของสิ่งใดสิ่งหนึ่งในฐานข้อมูลเป็นไปได้เหมือนเดิม

ในกรณีของระบบฐานข้อมูลเชิงวัตถุ จะแก้ปัญหานี้ด้วยการแทนค่าที่ใช้ในการอ้างอิงถึงวัตถุตัวนั้นเป็นค่าต่างหากค่าหนึ่ง ซึ่งจะเรียกค่านี้อีกว่า Object Identifier ลักษณะของค่านี้นี้จะแตกต่างไปจากค่าที่เก็บอยู่ในฐานข้อมูลค่าอื่นๆ ก็คือ จะมีค่าที่ไม่ซ้ำกันเพื่อที่ระบบจัดการฐานข้อมูลใช้ค่าเหล่านี้ในการอ้างอิงถึง Object ตัวใดตัวหนึ่งที่เก็บอยู่ในฐานข้อมูลเป็นหลัก ซึ่งในอ้างอิงนี้เราอาจพบว่า จะมี Object สองตัวใดๆ ที่มีค่าเหมือนกัน แต่มีการอ้างอิงโดยค่า Object Identifier คนละตัวก็ได้ ซึ่งค่านี้นี้ ระบบฐานข้อมูลจะใช้ในการอ้างอิงถึง Object ตัวนั้นเป็นแบบถาวร คือ ค่าที่ใช้ในการอ้างอิงเหล่านี้จะถูกสร้างและทำลายโดยตัวระบบจัดการฐานข้อมูลเอง และจะไม่สามารถทำการเปลี่ยนแปลงค่าต่างๆ เหล่านี้ได้ ซึ่งจะแตกต่างจากค่าอื่นๆ ที่อยู่ในฐานข้อมูล ส่วนความสามารถที่ผู้ใช้ระบบฐานข้อมูลจะนำค่าเหล่านี้ไปใช้ในการอ้างอิงหรือใช้งานได้หรือไม่ นั้น จะขึ้นอยู่กับระบบจัดการฐานข้อมูลที่ใช้ว่าจะเปิดโอกาสให้ผู้ใช้ได้เข้าถึงข้อมูลของค่าเหล่านี้หรือไม่ ซึ่งก็แตกต่างกันไปในแต่ละระบบ

##### 4.1.2 Complex Value

สำหรับ Object แต่ละตัวจะมีค่าที่เกี่ยวข้องด้วยอยู่ค่าหนึ่ง ซึ่งอาจเป็นค่าที่มีลักษณะซับซ้อนก็ได้ ซึ่งค่านี้นี้ก็เป็นค่าของ Object Identifier ก็ได้ เป็นผลให้สามารถมี Object สองตัวใดๆ ที่อ้างอิงถึง Object อื่น

เอกลักษณะเป็นเอกลักษณ์ของระบบจัดการฐานข้อมูลเชิงวัตถุก็คือ การที่ผู้ใช้สามารถเข้าถึงข้อมูลของค่าเหล่านี้ได้โดยไม่ต้องผ่านการดำเนินการใดๆ ทั้งสิ้น อีกทั้งยังมีให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวหนึ่งซึ่งเป็นตัวเดียวกันได้ โดยใช้ Object Identifier ที่มีค่าเหมือนกัน และเป็นผลให้เมื่อมีการเปลี่ยนแปลงข้อมูลในตัว Object นั้นๆ ก็จะไม่ส่งผลกระทบต่อไปยัง Object ตัวอื่นๆ ที่มีการอ้างอิงถึง ซึ่งค่าที่เก็บอยู่ใน Object นี้สามารถเรียกได้อีกถือว่าเป็นสถานะ (State) ของ Object ตัวนั้นได้ด้วย

Complex Datatype ข้อมูลอย่างง่าย ๆ (Simple datatype) โดยทั่วไปจะประกอบไปด้วย Integer , Floating Point, และ String ส่วนข้อมูลที่ซับซ้อนจะประกอบด้วยการนำเอา Simple datatype มารวมเป็นโครงสร้างตามลำดับของ String และ Array ที่มีอยู่ในแอปพลิเคชันโปรแกรม การรวมกันของสิ่งเหล่านี้จะต้องการฟังก์ชันการทำงานที่นอกเหนือจากที่มีอยู่ใน Storage Engine จากจุดนี้เองที่ทำให้มีการนำเอา ODBMS มาช่วยให้เกิดการรวมกันของข้อมูลโดยผ่านทาง OOP เพื่อให้ง่ายต่อการสร้างหรือการรวมกันของข้อมูล เมื่อทำการสร้างและเก็บไว้เป็นฐานข้อมูลโดยผ่านทาง ODBMS ที่เป็นตัวจัดการก็จะได้ฐานข้อมูลขนาดหนึ่งบล็อก ที่ไม่สามารถแตกย่อยหรือกระจายให้เป็นส่วนหนึ่งของข้อมูลที่เล็กลงได้อีก

User – Defined type ODBMS จะสนับสนุนการนิยามและการกำหนดชนิดของข้อมูลได้หลากหลาย รวมทั้ง Compound Type และ Variable – Length Type ด้วย โดย ODBMS จะทำการเก็บค่าของข้อมูลที่ผู้ใช้กำหนดไว้เป็นหนึ่งส่วนของข้อมูล ต่อจากนั้นก็แปลงข้อมูลเหล่านั้นไปเป็นฐานข้อมูล

#### 4.1.3 Method

Object ที่เก็บอยู่ในระบบฐานข้อมูลเชิงวัตถุ นั้นจะมีวิธีการใช้งานโดยผ่านทาง Method นี้ โดย Method จะประกอบด้วย ชื่อ, รูปแบบในการใช้ (Signature), และส่วนที่เป็นตัวเนื้อหาของ Method

ในการใช้งานนี้จะต้องอ้างอิง Method ตามรูปแบบชื่อและรูปแบบในการใช้ให้ถูกต้องตามที่มีอยู่ ส่วนที่เป็นตัวเนื้อหาของ Method ซึ่งแสดงถึงรูปแบบการทำงานของ Method นั้นๆ โดยปกติจะถูกเขียนด้วยภาษาที่เป็นส่วนขยายเพิ่มเติมมาจากภาษาโปรแกรม

#### 4.1.4 Class

จะหมายถึง กลุ่มของ Object ที่มีโครงสร้างภายในที่เหมือนกัน นั่นคือ จะมีค่ารายละเอียดของค่า Attribute และรายละเอียดของ Method ที่เหมือนกัน สำหรับตัว Object ใดๆ ที่เป็นของ Class ใด Class หนึ่ง ที่อยู่เป็นฐานข้อมูลจะเรียก Class นี้ว่าเป็น Class หลัก (Primary Class, Immediate Class หรือ Proper Class) ของ Object นั้นๆ ซึ่งจะเป็น Class ที่ Object ตัวนั้นมีความสัมพันธ์ในรูปแบบที่เป็น ตัวแทน (Instance-of) ของ Class นั้น และใน Class นี้จะเก็บลักษณะต่างๆ ของตัว Object ที่อยู่ภายใน Class ได้แก่ Attribute และ Method ต่างๆ ในบางกรณีอาจพบว่าเราสามารถให้ Object ตัวหนึ่ง ๆ เป็นตัวแทนของ Class มากกว่า 1 Class ได้ โดยการสืบทอดคุณสมบัติ (Inheritance) ซึ่งจะอธิบายต่อไป แต่อย่างไรก็ตาม Object ตัวนั้นจะต้องมี Class หลักได้เพียงตัวเดียว

#### 4.1.5 การสืบทอดคุณสมบัติ

ในระบบฐานข้อมูลเชิงวัตถุ แนวคิดของการสืบทอดคุณสมบัติทำให้ Object มี Attribute และ Method ที่เจาะจง (Specialized) ลงไปจาก Class ที่เป็น Class ดั้ง (Superclass หรือ Parent Class) ได้ โดยการสืบทอด Attribute และ Method จาก Class ดังกล่าวมายัง Class ของตัวจะเรียกว่า Subclass สำหรับ Object ที่อยู่ภายใน Class ใหม่ นี้ก็ยิ่งถือว่าเป็นตัวแทนของ Class ดั้งที่สืบทอดมาจากคุณสมบัติที่มีมาจากการ

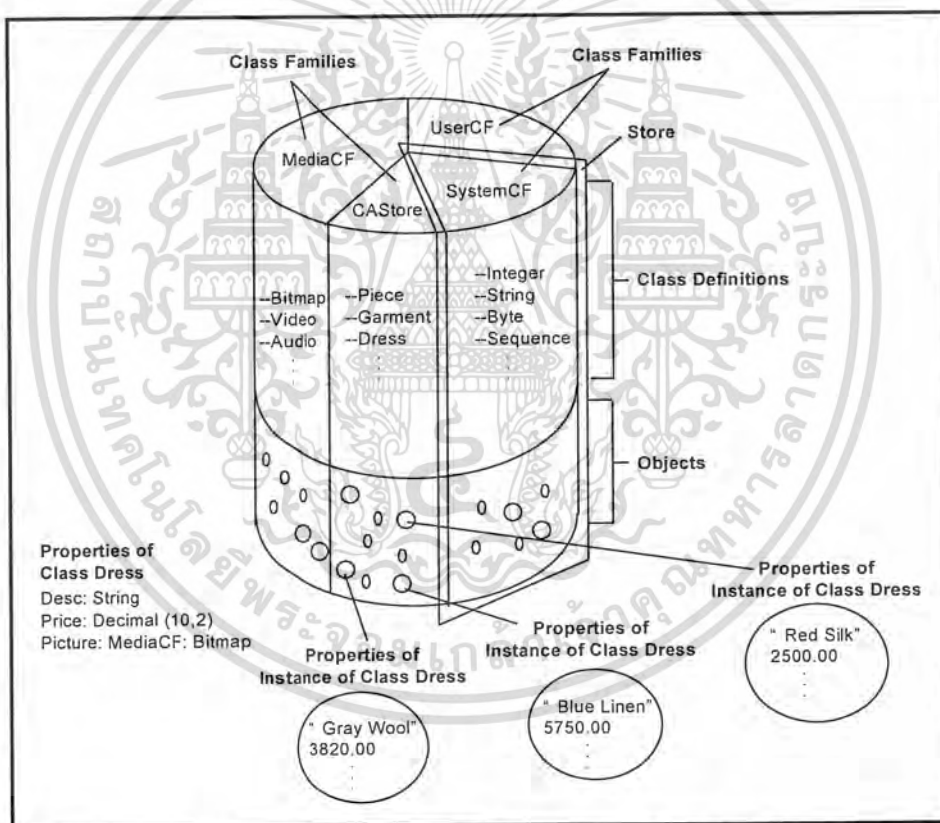
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สืบทอดนี้ทำให้เราได้ประโยชน์จากการที่เราสามารถเพิ่มศักยภาพทางด้านการนำกลับมาใช้ใหม่ (Reusability) และการขยายเพิ่มเติมคุณสมบัติ (Extensibility) ได้

ในส่วนต่อไปจะเป็นการแสดงตัวอย่างของลำดับชั้นของ Class (Class Hierarchies) ที่ได้มาจากการสืบทอดคุณสมบัติ โดยการสืบทอดจะเป็นในลักษณะแบบ Class ที่อยู่ในระดับที่ต่ำกว่าจะสืบทอดค่าคุณสมบัติต่างๆ จาก Class ที่อยู่สูงขึ้นไป

4.1.6 Polymorphism

สำหรับความหมายของ Polymorphism โดยรวมแล้วก็คือ รูปแบบหนึ่งของการสืบทอดคุณสมบัติ โดยจะแบ่งออกได้เป็นสองลักษณะใหญ่ก็คือ ลักษณะแบบ Overloading จะหมายถึงการที่ Method สองตัวใดๆ ใน Class เดียวกันที่มีชื่อเหมือนกัน แต่รูปแบบการใช้งานที่แตกต่างกัน และแบบ Overriding ซึ่งหมายถึงความสามารถของโปรแกรมที่อ้างอิงถึง Attribute หรือ Method ที่มีชื่อเหมือนกัน แต่มีการเปลี่ยนแปลงความหมายไปเมื่อมีการสืบทอดค่าคุณสมบัตินั้นมาจาก Class ดั้ง



รูปที่ 4-1 แสดงโมเดลการเก็บข้อมูลของฐานข้อมูลเชิงวัตถุ

อธิบายรูป

- Class families คือ collection ของ class ที่มีลักษณะเดียวกันซึ่ง classes ใน class family จะอยู่ในรูปของ tree หรือ set ของ tree
- Classes จะหมายถึง collection ของ object ที่มี feature พื้นฐานเดียวกัน (common feature) ตัวอย่างเช่น class ของ dress จะสามารถประกอบด้วย dress objects + common feature (Description, ราคา, รูปของชุด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Objects/Instances คือ ส่วนของ data ใน class เช่น แต่ละ Dress ใน Dress class คือ Object(หรือ instance) โดยแต่ละ object จะประกอบด้วย information(Properties) เช่น ราคา,description,picture

- Method เป็น functions ที่จะยอมให้เราสามารถ

- calculate derived data(เช่น ราคา+tax)

- Maintain redundant data

- Access Information ที่อยู่นอกระบบ database

- ซ่อนรายละเอียดของ Information ภายใน

- Subclass คือ การ inherit properties และ methods ของ class (superclass) เช่น Class Manager นั้น inherits properties และ method ของ Class Employee นั่นก็หมายความว่า Manager จะมี properties และ method ที่มีรูปแบบเดียวกับ Employee (เช่น Name,DOB,Phone#) และเราสามารถเพิ่ม properties(หรือ method) ใหม่ ๆ ให้กับ class Manager ก็ได้

- ถ้า class นั้นๆ มี Superclass มากกว่า 1 superclass เราจะเรียกว่า multiple inheritance

- เมื่อเราพิจารณาที่ class families ที่ CAStore จะพบว่าประกอบด้วย class definition เช่น piece(ผ้าเป็นชิ้น) , garment(เสื้อผ้า) , Dress(กระโปรง) เห็นได้ว่าจะประกอบด้วย Object พื้นฐานเดียวกันคือเป็นผ้า และนอกจากนี้ properties ของ dress คือ desc(String),Price(Decimal [10,2]),Picture(MediaCF:Bitmap) ถูกแบ่งข้อมูลเป็นแต่ละส่วนชัดเจน

## 4.2 ประโยชน์จากฐานข้อมูลเชิงวัตถุ

4.2.1 ฐานข้อมูลเชิงวัตถุได้ผสมความสามารถในด้านความเร็วของฐานข้อมูลบนระบบเครือข่ายเข้ากับความเร็วของฐานข้อมูลเชิงสัมพันธ์ และเพิ่มความสามารถในการทำหน่วยการทำงาน (transaction) ละเอียด โดยการใช้ persistent lock และการทำ version แบบอัตโนมัติเป็นซึ่งพื้นฐานให้ฐานข้อมูลเชิงวัตถุสามารถทำการประมวลผลแบบกระจายได้ นอกจากนี้การทำงานด้วยวิธีการส่งผ่านข้อมูลที่ใช้อยู่ในการโปรแกรมเชิงวัตถุก็ยังช่วยอธิบายการทำงานแบบกระจายได้อีกด้วย

4.2.2 ฐานข้อมูลเชิงวัตถุให้ประสิทธิภาพในการทำงานกับวัตถุ และความสัมพันธ์ระหว่างวัตถุ ที่ซับซ้อนที่ดีกว่า ทั้งนี้เนื่องมาจากการทำงานของฐานข้อมูลเชิงวัตถุไม่จำเป็นต้องมีการรวม(join)ตาราง ซึ่งเป็นการทำงานที่ช้าที่สุดในการทำงานของระบบเชิงสัมพันธ์ เพราะฐานข้อมูลเชิงวัตถุจะเก็บวัตถุไว้ด้วยกัน การค้นหาวัตถุจึงทำได้ในครั้งเดียว แม้ว่าจะเป็นวัตถุขนาดใหญ่ การทำงานของ โปรแกรมประยุกต์ในระบบเชิงสัมพันธ์นั้น โปรแกรมประยุกต์จะได้ข้อมูลจากร่างที่ไม่ได้เชื่อมต่อกันในขณะที่ทำงาน (run time) ในขณะที่โปรแกรมประยุกต์ในระบบฐานข้อมูลเชิงวัตถุ ข้อมูลจะถูกเก็บโดยตรงดังนั้นแทนที่จะต้องมีการรวมตารางกัน โปรแกรมประยุกต์สามารถค้นหาวัตถุได้โดยใช้ตัวชี้ที่อยู่ในวัตถุทำ นอกจากนี้การแยกเก็บข้อมูลของวัตถุในรูปของคุณลักษณะ(attribute) หรือตารางของฐานข้อมูลเชิงสัมพันธ์นั้น ยังเป็นการเพิ่มการจราจร(traffic) ระหว่างโปรแกรมประยุกต์กับฐานข้อมูลในการรวบรวมเพื่อให้ได้มาซึ่งวัตถุ และในทางกลับกันเพื่อเก็บวัตถุลงฐานข้อมูลด้วย การรวมตารางเพื่อให้ได้มาซึ่งวัตถุ จะยังมีปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มากขึ้นเมื่อฐานข้อมูลต้องทำการเก็บข้อมูลที่มีความซับซ้อนสูง เช่น รูปภาพ เอกสาร บัญชี และอื่น ๆ นอกจากนี้การเก็บวัตถุของฐานข้อมูลเชิงวัตถุยังประหยัดพื้นที่มากกว่าของระบบเชิงสัมพันธ์ที่ต้องมีการสร้าง index file จำนวนมากกว่าด้วย

4.2.3 การอธิบายด้วยวัตถุสามารถรองรับ โปรแกรมประยุกต์ทางด้านสื่อผสม(multimedia) ซึ่งประกอบด้วยวัตถุที่มีคุณสมบัติเป็นรูปร่าง และมีการเปลี่ยนแปลงตามเวลา เช่น วิดีโอ เสียง ข้อความ หรือสื่อชนิดอื่น ๆ ด้วยการสร้างแบบจำลอง(model) และจัดเก็บได้ภายใน conceptual schema เดียวกัน

4.2.4 การนำกลับมาใช้ได้ใหม่ของวัตถุ และความสามารถในการเพิ่มเติมคุณสมบัติให้กับวัตถุที่มีอยู่เดิม เป็นประโยชน์อีกส่วนหนึ่งที่ได้จากโครงสร้างแบบ classification ของฐานข้อมูลเชิงวัตถุ

4.2.5 ฐานข้อมูลเชิงวัตถุมีการควบคุมการค้นหาที่ดีกว่า เพราะตัวชี้ที่ชี้ไปยังวัตถุที่มีความสัมพันธ์กันนั้นถูกเก็บอยู่ในตัววัตถุเอง

4.2.6 ฐานข้อมูลเชิงวัตถุมีการตรวจสอบความถูกต้องในการอ้างอิงอยู่ในตัว เนื่องจาก การอ้างอิงถึงวัตถุอื่นนั้น ไม่ใช่การอ้างอิงถึงเท่านั้นแต่เป็นการบอกที่อยู่ของวัตถุนั้น โดยตรง จึงมีความถูกต้อง อยู่ในตัวมันอยู่แล้ว นอกจากนี้ฐานข้อมูลเชิงวัตถุยังแสดงถึงความสัมพันธ์ของวัตถุได้มากกว่าระบบอื่น ด้วยความสามารถในการสร้างชนิดข้อมูล และรวมเอาข้อบังคับต่าง ๆ เข้าไว้ในวัตถุ อีกทั้งยังมีความสามารถในการจัดการข้อยกเว้น และการข้ามผ่านข้อบังคับด้วย

4.2.7 ในระบบเชิงสัมพันธ์นั้นการกระจายกันเก็บของวัตถุ เป็นปัญหาที่สำคัญอย่างหนึ่งของผู้ออกแบบ เพราะผู้ออกแบบจะต้องทำให้แน่ใจได้ว่าตารางที่มีการรวมกันบ่อย ๆ จะถูกเก็บที่เก็บเดียวกัน และอยู่ในพื้นที่ที่ใกล้เคียงกันเพื่อลดปริมาณการเข้าถึงดิสก์(disk access) แต่ในฐานข้อมูลเชิงวัตถุนั้น วัตถุที่เป็นกลุ่มเดียวกันจะถูกเก็บไว้ด้วยกันเหมือนกับการประกอบกันของวัตถุ โครงสร้างของคลาส จะถูกนำมาเป็นหลักในการพิจารณาการแยกวัตถุออกเป็นกลุ่มในการเก็บ

4.2.8 ดังที่ได้กล่าวไปแล้วว่าโปรแกรมประยุกต์ที่อยู่ในระบบฐานข้อมูลเชิงสัมพันธ์นั้น จะใช้ข้อมูลที่มีโครงสร้างไม่เหมือนกัน คือ โปรแกรมประยุกต์ใช้ตัวแปร ในขณะที่ฐานข้อมูลใช้ตาราง ดังนั้นการทำงานกับฐานข้อมูล จะต้องมีภาษาอีกภาษาหนึ่งมาช่วยในการทำงานนั้นก็คือ Data Manipulation Language (DML) ซึ่งทำให้ผู้เขียนโปรแกรมต้องเขียนภาษาถึง 2 ภาษาก็คือ ภาษาของตัวโปรแกรมประยุกต์นั้น กับ DML ในขณะที่ในระบบฐานข้อมูลเชิงวัตถุนั้น โปรแกรมประยุกต์และฐานข้อมูล จะใช้ตัวแปรในรูปแบบเดียวกัน จึงไม่จำเป็นต้องมี DML ซึ่งเป็นการลดภาระให้กับผู้พัฒนาระบบ

4.2.9 ความสามารถในการสร้าง library ของคลาสของโปรแกรมประยุกต์และฐานข้อมูลที่มีลักษณะคล้าย ๆ กันช่วยให้การพัฒนากระบวนการเป็นไปได้อย่างรวดเร็วขึ้น เพราะไม่ต้องเสียเวลาในการสร้าง class เดิมทีเคยทำมาก่อนแล้ว

4.2.10 ฐานข้อมูลเชิงวัตถุสามารถแปลงสถาปัตยกรรมไปเป็นแบบเครื่องให้บริการ/เครื่องขอ บริการ(client/server) และ การทำงานแบบกระจายได้โดยง่าย เพราะการส่งผ่านข้อมูลไปยัง method ที่ถูกห่อหุ้ม(encapsulate)อยู่ในตัววัตถุนั้นแสดงให้เห็นถึงสถาปัตยกรรมแบบเครื่องให้บริการ/เครื่องขอ บริการ (client/server) อยู่แล้ว จึงไม่จำเป็นต้องทำการศึกษาการทำงานของโปรแกรมประยุกต์ต่าง ๆ อีก ตัวอย่าง เช่น วัตถุชื่อ Employee มี method คำนวณอายุอยู่ในวัตถุ เป็นศูนย์กลางให้เรียกใช้อยู่แล้วโปรแกรม เอกสารนี้เป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประยุกต์เพียงแต่ทำการเรียกใช้ method เท่านั้น ไม่จำเป็นต้องคัดลอก function นี้ไปไว้ในโปรแกรมประยุกต์เหมือนกับการโปรแกรมที่ไม่ใช่เชิงวัตถุ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### ฐานข้อมูลเชิงวัตถุกับการประยุกต์เชิงเวลา

#### 5.1 ปัญหาต่างในการนำฐานข้อมูลเชิงสัมพันธ์ประยุกต์กับฐานข้อมูลเชิงเวลา

##### 5.1.1 ปัญหาเรื่องของความซ้ำซ้อนของข้อมูล

ในช่วงบทที่ 2 ได้มีการนำเสนอตัวอย่าง การนำฐานข้อมูลแบบรีเลชัน มาประยุกต์ใช้กับฐานข้อมูลเชิงเวลาบางแล้ว ซึ่งเรียกวิธีการดังกล่าวว่า tuple versioning เป็นวิธีการในการแก้ปัญหาของฐานข้อมูลเชิงเวลา ซึ่งวิธีดังกล่าว เป็นการสร้าง tuple ของข้อมูลขึ้นมาใหม่ เพื่อข้อมูลใน tuple เดิมมีการเปลี่ยนแปลงแก้ไขขึ้น ซึ่งวิธีดังกล่าว ยังมีปัญหาในเรื่องของความซ้ำซ้อนของข้อมูลอยู่ เนื่องจากค่าในคอลัมน์อื่นที่ไม่มีการเปลี่ยนแปลงนั้นก็จะถูกเขียนซ้ำลงไป ใน tuple อันใหม่ ตัวอย่างเช่น

| NAME  | SSN   | SALARY | VALID_START | VALID_END  |
|-------|-------|--------|-------------|------------|
| Smith | 00001 | 25000  | 1997-06-15  | 1998-05-31 |
| Smith | 00001 | 30000  | 1998-06-01  | 9999-12-31 |
| Brown | 00002 | 40000  | 1996-05-01  | 1997-08-10 |
| Wong  | 00003 | 38000  | 1998-08-01  | 9999-31-12 |

ตารางที่ 5-1 แสดงความซ้ำซ้อนที่เกิดจากการใช้ฐานข้อมูลแบบสัมพันธ์

จากตารางลิสต์ใหม่เก็บประวัติเงินเดือนของพนักงาน 2 tuple แรกของตาราง ซึ่งบอกประวัติเงินเดือนของนาย Smith ครั้งแรกนั้นพนักงานคนดังกล่าว มีเงินเดือน 25000 แต่ต่อมาเงินเดือนเปลี่ยนเป็น 3000 ทำให้ต้องมีการแก้ไข ข้อมูลเสียใหม่ ซึ่งการทำดังกล่าวคือ การเขียนเวลาสิ้นสุดของช่วงเวลาที่นาย Smith มีเงินเดือน 25000 และหลังจากนั้นก็ทำการ insert ข้อมูลที่บอกว่า Smith มีเงินเดือน 30000 เข้าไปในฐานข้อมูล

|       |       |       |            |            |
|-------|-------|-------|------------|------------|
| Smith | 00001 | 25000 | 1997-06-15 | 1998-05-31 |
| Smith | 00001 | 30000 | 1998-06-01 | 9999-12-31 |

ตารางที่ 5-2 แสดงความซ้ำซ้อนของข้อมูลอื่นที่ไม่ได้เปลี่ยนแปลงตามเวลา

ซึ่งการทำดังกล่าวทำให้ข้อมูลอื่นที่ไม่เกี่ยวข้อง เช่น ชื่อ และหมายเลขประจำตัว เกิดปรากฏขึ้นทั้ง 2 tuple ซึ่งถ้าหากมีการเปลี่ยนแปลงเงินเดือนอีก ก็จะเกิดความซ้ำซ้อนเช่นนี้อีกต่อไปเรื่อย ซึ่งในความเป็นจริงจะมีหลายคอลัมน์ที่ ข้อมูลไม่ได้มีการเปลี่ยนแปลง เป็นผลให้เกิดความซ้ำซ้อนขึ้นเป็นจำนวนมาก ซึ่งทำให้ฐานข้อมูลไม่มีประสิทธิภาพ เพราะความซ้ำซ้อนทำให้การ insert , delete , update นั้นมีปัญหาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และความซ้ำซ้อนยังทำให้สิ้นเปลืองพื้นที่ในการจัดเก็บข้อมูลด้วย ซึ่งในความเป็นจริงแล้วนั้นในตารางหนึ่งๆ มักจะประกอบด้วยข้อมูลต่างๆ มากมาย ซึ่งถ้าเป็นเช่นนั้นแล้วก็ยิ่งเพิ่มความซ้ำซ้อนให้มากยิ่งขึ้น

### 5.1.2 ปัญหาเรื่องของ Primary Key

นอกจากปัญหาเรื่องของความซ้ำซ้อนแล้ว การนำฐานข้อมูลเชิงสัมพันธ์มาใช้กับฐานข้อมูลเชิงเวลา ยังเกิดปัญหาเรื่องของ Primary Key ด้วย เพราะในฐานข้อมูลเชิงเวลา ไม่สามารถใช้ Primary Key เหมือนในฐานข้อมูลแบบธรรมดาได้ ทั้งนี้เพราะ Primary Key จะเกิดการซ้ำกันขึ้น ยกตัวอย่างเช่น ฐานข้อมูลเก็บประวัติเงินเดือนเดิม ใช้ หมายเลข SSN เป็น Primary Key แต่พอเป็นฐานข้อมูลเชิงวัตถุแล้ว ไม่สามารถใช้ SSN เป็น Primary Key เหมือนเดิมได้ ต้องมีการสร้าง Primary Key ใหม่โดยการนำไปรวมกับ VALID\_START เพื่อไม่ให้ Primary Key เกิดการซ้ำกันเกิดขึ้น

ซึ่งจากปัญหาดังกล่าว ทำให้การนำฐานข้อมูลเชิงสัมพันธ์มาประยุกต์ใช้กับฐานข้อมูลเชิงเวลานั้น ไม่ค่อยมีประสิทธิภาพเท่าที่ควรนัก

## 5.2 การใช้ฐานข้อมูลเชิงวัตถุกับการประยุกต์เชิงเวลา

การนำเอาฐานข้อมูลเชิงวัตถุมาประยุกต์ใช้กับฐานข้อมูลเชิงเวลานั้น โดยอาศัยคุณสมบัติของวัตถุเข้ามาช่วยแก้ไขปัญหา วัตถุสามารถมีโครงสร้างข้อมูลที่ซับซ้อนได้เช่น ลิสต์, เซต, อาร์เรย์ เป็นต้น และยังสามารถกำหนดชนิดของโครงสร้างข้อมูลได้เอง ทำให้มีความยืดหยุ่นมากกว่า การใช้ฐานข้อมูลเชิงสัมพันธ์ จากข้อดีที่ว่าทำให้วัตถุหนึ่งๆ สามารถเก็บข้อมูลต่างๆ ไว้ในแอททริบิวต์ชื่อเดียวกันได้ โดยการใช้ลักษณะของลิสต์ หรือ อาร์เรย์ เป็นต้น

วิธีการที่นำฐานข้อมูลเชิงวัตถุมาประยุกต์ใช้กับฐานข้อมูลเชิงเวลา เรียกว่า แอททริบิวต์เวอร์ชันนิ่ง (Attribute version) แอททริบิวต์ที่สามารถมีการเปลี่ยนแปลงตามเวลา(time-varying) ได้นั้นค่าของข้อมูลในเวลาที่ต่างๆ จะถูกเก็บรวมกันไว้ในแอททริบิวต์นั้น โดยมีลักษณะเป็นลิสต์ โดยในแต่ละตัวของลิสต์นอกจากจะเก็บข้อมูลแล้ว ยังมีข้อมูลของเวลาคือวาลิดไทม์ (valid time), ทรานแซคชันไทม์(transaction time) หรือ ทั้ง 2 ตัวกำกับช่วงเวลาของข้อมูลตัวนั้นไว้ว่ามีค่าความเป็นจริงอยู่ในช่วงใดหรือ มีการเปลี่ยนแปลงในช่วงเวลาใด ซึ่งเวลาจะเรียงจากอดีตมาล่าสุด โดยสมาชิกของลิสต์ตัวสุดท้ายจะเป็นสถานะล่าสุดของวัตถุนั้นและสมาชิกตัวแรกของลิสต์จะเป็นสถานะของวัตถุที่เก่าที่สุดในอดีต ส่วนแอททริบิวต์อื่นที่ไม่เกี่ยวข้องหรือไม่มีการเปลี่ยนแปลงตามเวลานั้นก็เก็บตามปกติ ซึ่งทำให้ปัญหา เรื่องของความซ้ำซ้อนของแอททริบิวต์ที่ไม่เปลี่ยนแปลงตามเวลา ไม่เกิดขึ้นเหมือนในฐานข้อมูลเชิงสัมพันธ์

ตัวอย่างการประยุกต์ใช้ฐานข้อมูลเชิงวัตถุกับการประยุกต์เชิงเวลา สมมุติเราต้องการเก็บประวัติเงินเดือนของลูกจ้างในบริษัท เราสามารถออกแบบฐานข้อมูลแบบง่ายๆ โดยใช้คุณสมบัติของฐานข้อมูลเชิงวัตถุมาช่วยในการเก็บประวัติเงินเดือนของลูกจ้างแต่ละคน

```

class Temporal_salary
{
    attribute Date          valid_start_time;
    attribute Date          valid_end_time;
    attribute Float         Salary;
};

```

คลาส Temporal\_salary เป็นคลาสต้นแบบของข้อมูลเงินเดือนที่จะใช้เก็บประวัติ โดยในคลาสนี้จะประกอบด้วยแอททริบิวต์ 3 ตัว ได้แก่แอททริบิวต์ชื่อ Salary มีค่าไทม์คือจำนวนจริงเก็บค่าของเงินเดือน และ valid\_start\_time,valid\_end\_time มีค่าไทม์เป็นวันที่เก็บค่าของช่วงเวลาที่ค่าเงินเดือนนั้นๆ มีค่าเป็นจริงในตัวอย่างใช้ฐานข้อมูลเชิงเวลาแบบวาลิดใหม่

```

class Department
{
    attribute String         Dname;
    attribute Number        Dno;
    attribute Temporal_Employee_VT Manager;
};

```

คลาส Department เป็นคลาสที่แทนวัตถุแผนกที่ลูกจ้างคนนั้นทำงานอยู่ ประกอบไปด้วยแอททริบิวต์ ได้แก่ Dname คือชื่อของแผนก Dno เป็นหมายเลขของแผนกนั้น และ Manager แทนหัวหน้าของแผนกนั้น ซึ่งมีค่าไทม์เป็นคลาส Temporal\_Employee\_VT ซึ่งจะกล่าวถึงต่อไป โดยการกำหนดค่าไทม์ในลักษณะแบบนี้จะใช้การเลพเฟอเรนกันของวัตถุ โดยอาศัยตัวบ่งชี้วัตถุ (object identifier)

```

class Temporal_Employee_VT
{
    attribute String         Name;
    attribute String         SSN;
    attribute Department     Department;
    attribute Temporal_Employee_VT Supervisor;
    attribute list<Temporal_Salary> Salary_history;
};

```

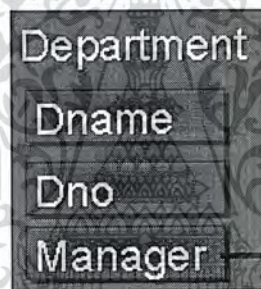
คลาส Temporal\_Employee\_VT เป็นคลาสของลูกจ้างที่เราต้องการเก็บประวัติเงินเดือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งประกอบด้วยแอททริบิวต์ต่างๆ ดังนี้ name คือชื่อของลูกจ้าง ssn คือหมายเลขประกันสังคมของลูกจ้างคนนั้น แอททริบิวต์ชื่อ Department ซึ่งมีค่าไทม์เป็นคลาส Department จะอ้างอิงกับวัตถุแผนก แอททริบิวต์ Supervisor มีค่าไทม์เป็น Temporal\_Employee\_VT จะอ้างอิงกับวัตถุของลูกจ้างนั่นเอง และสุดท้ายคือ แอททริบิวต์ Salary\_history มีค่าไทม์เป็น Temporal\_Salary ที่เราได้ออกแบบไว้ในตอนต้น ซึ่งแอททริบิวต์นี้มีลักษณะเป็นลิสต์ของวัตถุ Temporal\_Salary ด้วย

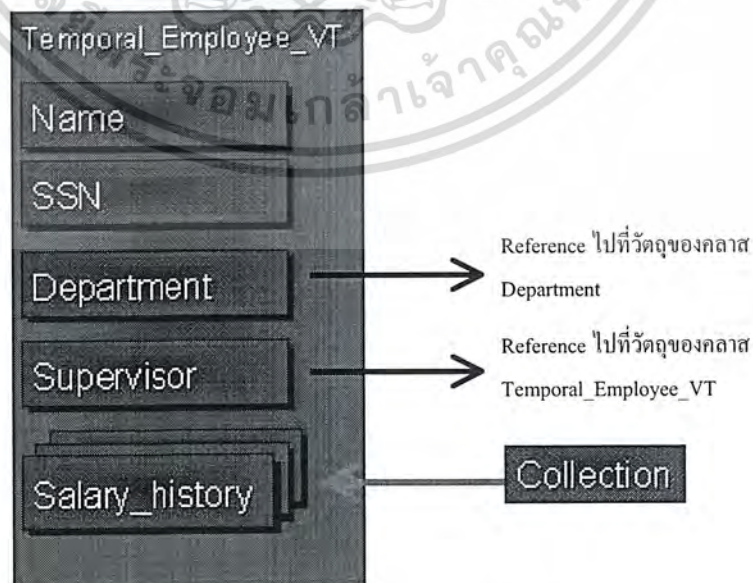


รูปที่ 5-1 แสดงคลาส Temporal\_Salary



Reference ไปที่วัตถุของคลาส  
Temporal\_Employee\_VT

รูปที่ 5-2 แสดงคลาส Department



Reference ไปที่วัตถุของคลาส  
Department

Reference ไปที่วัตถุของคลาส  
Temporal\_Employee\_VT

Collection

รูปที่ 5-3 แสดงคลาส Temporal\_employee\_VT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งการทำแบบนี้ทำให้เราสามารถเก็บประวัติเฉพาะเงินเดือนของลูกจ้างได้ โดยที่แอททริบิวต์อื่นๆ ที่ไม่ได้เปลี่ยนแปลงตามเวลาจะไม่เกิดความซ้ำซ้อนขึ้น แล้วยังทำให้การจัดการง่ายขึ้นด้วย เพราะเราสามารถหาประวัติเงินเดือนของลูกจ้างต่างๆ ได้โดยการดูจากวัตถุของลูกจ้างนั้นๆ ได้โดยไม่ต้องเสียเวลาอีกทั้งประวัติเงินเดือนก็มีการเรียงลำดับตามเวลาแล้ว โดยสมาชิกของลิสต์ตัวสุดท้ายก็คือสถานะปัจจุบันของข้อมูลนั้นๆ นั่นเอง เพราะถ้าหากเวลาไม่เรียงลำดับแล้วนั้น การนำไปใช้จำเป็นต้องค้นหาก่อน ทำให้ยุ่งยากเสียเวลา

ส่วนปัญหาเรื่องของคีย์หลักนั้น ในฐานข้อมูลเชิงวัตถุจะมี ตัวบ่งชี้วัตถุ ซึ่งใช้ในการอ้างอิงวัตถุแต่ละตัว ทำให้ปัญหาเรื่องของคีย์หลัก ที่เกิดกับฐานข้อมูลเชิงสัมพันธ์ไม่เกิดขึ้น เพราะถึงแม้ว่าในวัตถุจะมีค่าของคีย์หลักเหมือนกัน แต่เราก็สามารถแยกความแตกต่างของวัตถุ 2 ตัวนั้นได้ จากตัวบ่งชี้วัตถุ และไม่ต้องยุ่งยากในการทำคีย์หลักใหม่เหมือนในฐานข้อมูลแบบสัมพันธ์ด้วย

สำหรับการใช้โครงสร้างข้อมูลแบบลิสต์ในการเก็บข้อมูลเชิงเวลานั้น เนื่องจากคุณลักษณะของลิสต์ที่มีตำแหน่งของแต่ละอีลีเมนต์ ทำให้การจัดการต่างๆ สะดวก โดยสมาชิกของลิสต์ตัวแรกก็จะเป็นสถานะของข้อมูลที่เก่าที่สุด และตัวสุดท้ายก็จะมีสถานะของข้อมูลล่าสุดนั่นเอง ซึ่งโดยปกติเราจะยุ่งกับข้อมูลในสถานะสุดท้ายเป็นส่วนมาก ทำให้การเข้าถึงสะดวก เนื่องจากเราสามารถอ้างอิงถึงสมาชิกตัวสุดท้ายของลิสต์นั้นได้เลย แต่ลิสต์เองก็มีข้อเสียคือไม่รับประกันว่าสมาชิกนั้นจะไม่มีค่าซ้ำกันเกิดขึ้น ซึ่งสำหรับข้อมูลเชิงเวลาเราต้องคำนึงถึงเรื่องของความซ้ำซ้อน แต่หากเป็นโครงสร้างข้อมูลแบบอื่นที่คำนึงถึงความซ้ำกันของสมาชิกเช่น เซต ก็จะทำให้สามารถรับประกันความซ้ำซ้อนได้ในระดับโครงสร้างของข้อมูลโดยปริยาย ในการแก้ปัญหาความซ้ำซ้อนของลิสต์นั้นในโครงการเราจะใช้ข้อดีของฐานข้อมูลเชิงวัตถุมาช่วย โดยมีเมธอดมาตรวจสอบข้อมูลก่อนที่จะมีการเพิ่มหรือแก้ไขเปลี่ยนแปลง ข้อมูลสำหรับสมาชิกของลิสต์นั้นๆ

## บทที่ 6

# การนำฐานข้อมูลเชิงวัตถุและแนวคิดของฐานข้อมูลเชิงเวลา มาประยุกต์ใช้กับระบบงานทะเบียนนักศึกษา

ในตอนต้นนั้น ได้กล่าวถึงแนวคิดของฐานข้อมูลเชิงเวลา ประเภทของฐานข้อมูลเชิงเวลาแบบต่างๆ และประโยชน์ของฐานข้อมูลเชิงเวลาที่มีความสามารถในการเก็บสถานะของข้อมูลในอดีตได้ ทำให้ข้อมูลต่างๆ ที่จำเป็นต้องอ้างอิงข้อมูลในอดีตหรือเกี่ยวข้องกับเวลา สามารถทำได้โดยมีประสิทธิภาพมากกว่าฐานข้อมูลแบบปกติ ที่เก็บเฉพาะข้อมูลที่สถานะปัจจุบันเท่านั้น และจากแนวคิดของฐานข้อมูลเชิงวัตถุ ทำให้การนำฐานข้อมูลเชิงเวลามาปฏิบัติ นั้น ทำให้สอดคล้องมากขึ้นกว่าการนำเอาฐานข้อมูลแบบสัมพันธ์มาใช้ เนื่องจากฐานข้อมูลเชิงสัมพันธ์ดังกล่าว ก่อให้เกิดปัญหาบางอย่างขึ้น จากการนำเอาฐานข้อมูลเชิงเวลาไปปฏิบัติ เช่น การซ้ำซ้อนของข้อมูล การซ้ำซ้อนของ primary key เดิม และความยุ่งยากของการทำ primary key ใหม่ ดังที่ได้กล่าวมาในบทก่อนนี้

ต่อไปนี้จะเป็นอย่างของการนำเอาระบบฐานข้อมูลเชิงวัตถุและแนวคิดของฐานข้อมูลเชิงเวลา มาประยุกต์ใช้กับระบบงาน เพื่อทำการทดสอบแนวคิด ดังที่ได้ศึกษามาในช่วงต้น และประเมินผลการใช้งาน ถึงข้อดี และข้อด้อยของการนำแนวคิดดังกล่าวไปใช้กับระบบงาน โครงการนี้จะขอ นำตัวอย่างระบบงานทะเบียนนักศึกษา มาเป็นกรณีศึกษา

### 6.1 ระบบงานทะเบียนนักศึกษา

ตัวอย่างลักษณะงานของระบบทะเบียนนักศึกษา

1. งานข้อมูลและประวัตินักศึกษา การเพิ่ม แก้ไข และลบประวัติ
2. การเรียน เช่น การลงทะเบียนเรียน การประเมินผลการศึกษา การพิจารณาจบ
3. การแก้ไขและบันทึกข้อมูลต่างๆ วิชาเรียน หลักสูตร

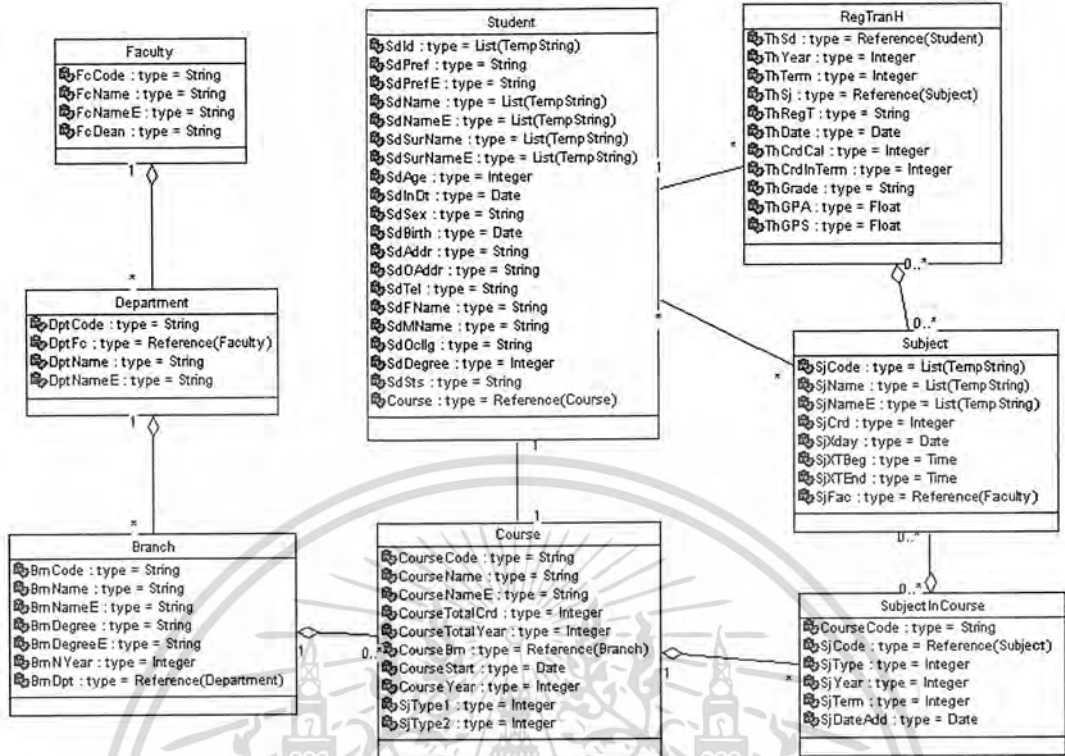
ซึ่งจากลักษณะของงานดังกล่าว เราจะนำเอาแนวคิดของฐานข้อมูลเชิงเวลาไปประยุกต์ใช้ ข้อมูลประวัตินักศึกษา บางแอททริบิวต์อาจจะมีการเปลี่ยนแปลงได้ซึ่งเราจะแสดงให้เห็นว่าสามารถนำเอาแนวคิดของฐานข้อมูลเชิงเวลามาใช้เก็บประวัติของข้อมูลต่างๆ ได้

ในการพัฒนาระบบงานทะเบียนศึกษานั้น เราได้ทำการพัฒนาเมธอดกลางต่างๆ เพื่อให้ผู้พัฒนาสามารถเรียกใช้ได้โดยเราไม่ต้องทำการเขียนโค้ดใหม่ทุกครั้งเมื่อทำการเพิ่ม แก้ไขหรือลบแอททริบิวต์ที่เปลี่ยนแปลงตามเวลา โดยเราสามารถเรียกใช้เมธอดกลางได้เลย

### 6.2 คลาสไลอะแกรมของระบบงานทะเบียนนักศึกษา

หลังจากที่เราได้ทำการศึกษาระบบงานทะเบียนนักศึกษาแล้วเราก็จะทำการออกแบบระบบงานทะเบียนนักศึกษาที่เราได้ทำการศึกษา โดยในการออกแบบงานทะเบียนศึกษานั้นเราจะใช้คลาสไลอะแกรมในการแสดงความสัมพันธ์ระหว่างคลาสต่างๆ จากนั้นเราก็จะนำคลาสไลอะแกรมมาใช้ในการทำฐานข้อมูลระบบงานทะเบียนนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-1 รูปคลาสไดอะแกรมแสดงความสัมพันธ์ระหว่างคลาสในระบบงานทะเบียนนักศึกษา

สำหรับรายละเอียดในคลาสต่างๆเป็นดังต่อไปนี้

- คลาส Student ความหมาย เป็นคลาสนักศึกษา
- คลาส Subject ความหมาย เป็นคลาสวิชา
- คลาส Course ความหมาย เป็นคลาสที่เกี่ยวข้องกับหลักสูตร
- คลาส RegTranH ความหมาย เป็นคลาสที่เกี่ยวข้องกับการลงทะเบียน การคิดผลการศึกษา
- คลาสFaculty ความหมาย เป็นคลาสที่เกี่ยวข้องกับคณะ
- คลาสDepartment ความหมาย เป็นคลาสที่เกี่ยวข้องกับภาควิชา
- คลาสBranch ความหมาย เป็นคลาสที่เกี่ยวข้องกับสาขาวิชา
- คลาสSubjectInCourse ความหมาย เป็นคลาสที่แสดงว่าเมื่อแต่ละวิชาไปอยู่ในแต่ละCourseแล้วนั้นมีสถานะเป็นเช่นไรเช่น วิชาเลือก , วิชาบังคับ

จากแผนภาพคลาสไดอะแกรม เราสามารถอธิบายคลาสต่างๆอย่างละเอียดได้ดังต่อไปนี้

**คลาส Student**

ความหมาย เป็นคลาสนักศึกษา

| ลำดับที่ | ชื่อแอททริบิวต์ | ประเภท     | ขนาด | คำอธิบาย     | หมายเหตุ |
|----------|-----------------|------------|------|--------------|----------|
| 1.       | SdId            | TempString | 15   | รหัสนักศึกษา | List     |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|     |            |            |     |                        |   |
|-----|------------|------------|-----|------------------------|---|
| 2.  | SdPref     | String     | 1   | คำนำหน้าชื่อภาษาไทย    |   |
| 3.  | SdPrefE    | String     | 1   | คำนำหน้าชื่อภาษาอังกฤษ |   |
| 4.  | SdName     | TempString | 55  | ชื่อภาษาไทย            | List  |
| 5.  | SdNameE    | TempString | 55  | ชื่อภาษาอังกฤษ         | List  |
| 6.  | SdSurName  | String     | 55  | นามสกุลภาษาไทย         |   |
| 7.  | SdSurNameE | String     | 55  | นามสกุลภาษาอังกฤษ      |   |
| 8.  | SdAge      | Integer    | 200 | อายุ                   |   |
| 9.  | SdInDt     | Date       |     | วันที่เข้ารับการศึกษา  |   |
| 10. | SdOutDt    | Date       |     | วันที่จบการศึกษา       |   |
| 11. | SdSex      | Integer    | 1   | เพศ                    | 1-ชาย<br>2-หญิง   |
| 12. | SdBirth    | Date       |     | วันเดือนปีเกิด         |   |
| 13. | SdAddr     | String     | 100 | ที่อยู่ปัจจุบัน        |   |
| 14. | SdOAddr    | String     | 100 | ภูมิลำเนาเดิม          |   |
| 15. | SdTel      | String     | 14  | เบอร์โทรศัพท์          |   |
| 16. | SdFName    | String     | 85  | ชื่อ-สกุลบิดา          |   |
| 17. | SdMName    | String     | 85  | ชื่อ-สกุลมารดา         |   |
| 18. | SdOcollg   | String     | 55  | สถานศึกษาเดิม          |   |
| 19. | SdDegree   | String     | 55  | วุฒิการศึกษาเดิม       |   |
| 20. | SdSts      | String     | 1   | รหัสสถานะของนักศึกษา   | 1-ผ่าน<br>2-ไม่ผ่าน<br>3-พักชั้น<br>4-พ้นสภาพ<br>5-ลาพักการศึกษา<br>6-รักษาสภาพ |
| 21. | Course     | Course     |     | หลักสูตรของนักศึกษา    | Reference   |

ตารางที่ 6-1 แสดงรายละเอียดของคลาสนักเรียน

### คลาส Subject

ความหมาย เป็นคลาสวิชา

| ลำดับที่ | ชื่อแอททริบิวต์ | ประเภท     | ขนาด | คำอธิบาย | หมายเหตุ |
|----------|-----------------|------------|------|----------|----------|
| 1.       | SjCode          | TempString |      | รหัสวิชา | List     |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|    |         |            |    |                    |           |
|----|---------|------------|----|--------------------|-----------|
| 2. | SjName  | TempString |    | ชื่อวิชาภาษาไทย    | List      |
| 3. | SjNameE | TempString |    | ชื่อวิชาภาษาอังกฤษ | List      |
| 4. | SjCrd : | Integer    | 10 | หน่วยกิต           |           |
| 5. | SjXday  | Date       |    | วันที่สอบ          |           |
| 6. | SjXTBeg | Time       |    | เวลาเริ่มสอบ       |           |
| 7. | SjXTEnd | Time       |    | เวลาเริ่มสอบ       |           |
| 8. | SjFac   | Faculty    |    | วิชาของคณะ         | Reference |

ตารางที่ 6-2 แสดงรายละเอียดของคลาสวิชา

### คลาส Course

ความหมาย เป็นคลาสที่เกี่ยวกับหลักสูตร

| ลำดับที่ | ชื่อแอททริบิวต์ | ประเภท  | ขนาด | คำอธิบาย                       | หมายเหตุ  |
|----------|-----------------|---------|------|--------------------------------|-----------|
| 1.       | CourseCode      | String  | 10   | รหัสหลักสูตร                   |           |
| 2.       | CourseName      | String  | 75   | ชื่อหลักสูตรภาษาไทย            |           |
| 3.       | CourseNameE     | String  | 75   | ชื่อหลักสูตรภาษาอังกฤษ         |           |
| 4.       | CourseTotalCrd  | Integer | 250  | จำนวนหน่วยกิตทั้งหมดในหลักสูตร |           |
| 5.       | CourseTotalYear | Integer | 10   | จำนวนปีทั้งหมดในหลักสูตร       |           |
| 6.       | CourseStart     | Date    |      | วันที่เริ่มใช้หลักสูตร         |           |
| 7.       | CourseYear      | Integer | 9999 | หลักสูตรปี                     |           |
| 8.       | CourseBrn       | Branch  |      | สาขาวิชา                       | Reference |
| 9.       | SjType1         | Integer | 50   | วิชาบังคับเลือกที่ต้องผ่าน     |           |
| 10.      | SjType2         | Integer | 50   | วิชาเลือกเสรีที่ต้องผ่าน       |           |

ตารางที่ 6-3 แสดงรายละเอียดของคลาสหลักสูตร

### คลาส RegTranH

ความหมาย เป็นคลาสที่เกี่ยวกับการลงทะเบียน

| ลำดับที่ | ชื่อแอททริบิวต์ | ประเภท  | ขนาด | คำอธิบาย            | หมายเหตุ            |
|----------|-----------------|---------|------|---------------------|---------------------|
| 1.       | ThSd            | Student |      | นักศึกษา            | Reference           |
| 2.       | ThYear          | Integer | 9999 | ปีการศึกษา          |                     |
| 3.       | ThTerm          | Integer | 5    | ภาคการศึกษา         |                     |
| 4.       | ThRegT          | String  | 1    | ชนิดของการลงทะเบียน | 1-ลงทะเบียนครั้งแรก |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|    |             |         |     |                       |  |
|----|-------------|---------|-----|-----------------------|--|
|    |             |         |     |                       | 2-ลงทะเบียนเพิ่ม<br>3-ลงทะเบียนเปลี่ยน<br>4-ลงทะเบียนถอน |
| 5. | ThDate      | Date    |     | วันที่ลงทะเบียน       |  |
| 6  | ThCrdCal    | Integer | 250 | หน่วยกิตสะสม          |  |
| 7  | ThCrdInTerm | Integer | 50  | หน่วยกิตที่ลงในเทอม   |  |
| 8  | ThGrade     | String  | 2   | เกรดของวิชา           |  |
| 9  | ThSj        | Subject |     | วิชาที่ลงทะเบียน      | Reference  |
| 10 | ThGpa       | Float   | 4   | ผลการศึกษาร้อยละ      |  |
| 11 | ThGps       | Float   | 4   | ผลการศึกษาในแต่ละเทอม |  |

ตารางที่ 6-4 แสดงรายละเอียดของคลาสการลงทะเบียน การคิดผลการศึกษา

#### คลาสFaculty

ความหมาย เป็นคลาสที่เกี่ยวกับคณะ

| ลำดับที่ | ชื่อแอททริบิวต์ | ประเภท | ขนาด | คำอธิบาย          | หมายเหตุ |
|----------|-----------------|--------|------|-------------------|----------|
| 1.       | FcCode          | String | 10   | รหัสคณะ           |          |
| 2.       | FcName          | String | 55   | ชื่อคณะภาษาไทย    |          |
| 3.       | FcNameE         | String | 55   | ชื่อคณะภาษาอังกฤษ |          |
| 4.       | FcDean          | String | 55   | ชื่อคณบดี         |          |

ตารางที่ 6-5 แสดงรายละเอียดของคลาสคณะ

#### คลาสDepartment

ความหมาย เป็นคลาสที่เกี่ยวกับภาควิชา

| ลำดับที่ | ชื่อแอททริบิวต์ | ประเภท  | ขนาด | คำอธิบาย          | หมายเหตุ  |
|----------|-----------------|---------|------|-------------------|-----------|
| 1.       | DptCode         | String  | 10   | รหัสภาควิชา       |           |
| 2.       | DptFc           | Faculty |      | คณะ               | Reference |
| 3.       | DptName         | String  | 55   | ชื่อคณะภาษาไทย    |           |
| 4.       | DptNameE        | String  | 55   | ชื่อคณะภาษาอังกฤษ |           |

ตารางที่ 6-6 แสดงรายละเอียดของคลาสภาควิชา

#### คลาสBranch

ความหมาย เป็นคลาสที่เกี่ยวกับสาขาวิชา

| ลำดับที่ | ชื่อแอททริบิวต์ | ประเภท     | ขนาด | คำอธิบาย            | หมายเหตุ  |
|----------|-----------------|------------|------|---------------------|-----------|
| 1        | BmDpt           | Department |      | ภาควิชา             | Reference |
| 2.       | BmCode          | String     | 10   | รหัสหลักสูตร        |           |
| 3.       | BmName          | String     | 60   | ชื่อสาขาวิชาภาษาไทย |           |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|    |            |         |    |                        |  |
|----|------------|---------|----|------------------------|--|
| 4. | BrmNameE   | String  | 60 | ชื่อสาขาวิชาภาษาอังกฤษ |  |
| 5. | BrmDegree  | String  | 40 | ชื่อหลักสูตรภาษาไทย    |  |
| 6. | BrmDegreeE | String  | 40 | ชื่อหลักสูตรภาษาอังกฤษ |  |
| 7. | BrmNYear   | Integer | 10 | จำนวนปีหลักสูตร        |  |

ตารางที่6-7 แสดงรายละเอียดของคลาสสาขาวิชา

#### คลาสSubjectInCourse

ความหมาย เป็นคลาสที่แสดงว่าเมื่อแต่ละวิชา ไปอยู่ในแต่ละหลักสูตรแล้วนั้นมีสถานะเป็นเช่นไรเช่น วิชาเลือก , วิชาบังคับ

| ลำดับที่ | ชื่อแอททริบิวต์ | ประเภท  | ขนาด | คำอธิบาย                | หมายเหตุ     |
|----------|-----------------|---------|------|-------------------------|--------------|
| 1.       | Course          | Course  |      | หลักสูตร                | Reference    |
| 2.       | Subject         | Subject |      | วิชา                    | Reference    |
| 3.       | SjType          | Integer | 10   | ชนิดของวิชา             |              |
| 4.       | SjYear          | Integer | 10   | ปีของนักศึกษาที่เรียน   | เช่น ปี1,ปี2 |
| 5.       | SjTerm          | Integer | 4    | เทอมของนักศึกษาที่เรียน |              |
| 6.       | SjDateAdd       | Date    |      | วันที่ทำการเพิ่มวิชา    |              |

ตารางที่6-8 แสดงรายละเอียดของคลาสวิชาในหลักสูตร

#### 6.3 ปัญหาที่พบในระบบทะเบียนนักศึกษาเดิมและวิธีแก้ปัญหาระบบทะเบียนนักศึกษาเดิม

##### ปัญหาที่พบในระบบทะเบียนนักศึกษาเดิม

- 1 ปัญหาในเรื่องของการที่รหัสวิชา,ชื่อวิชาสามารถเปลี่ยนแปลงได้ ซึ่งเมื่อมีการเปลี่ยนรหัสวิชาหรือชื่อวิชาแล้วไม่สามารถทำการเก็บรหัสวิชาและชื่อวิชาเก่าได้
- 2 ปัญหาในเรื่องของการตรวจสอบผลการศึกษา เช่น กรณีเมื่อนักศึกษาได้มีการลงทะเบียนวิชาในหลักสูตรเก่า ซึ่งวิชานั้นได้มีการเปลี่ยนแปลงข้อมูลไปแล้ว ทำให้พิจารณาการจบไม่ถูกต้อง
- 3 ปัญหาในเรื่องของการเปลี่ยนข้อมูลของนักศึกษา เช่น กรณีนักศึกษาที่จบไปแล้วได้มีการเปลี่ยนชื่อ ต้องสามารถตรวจสอบได้ว่าชื่อใหม่กับชื่อเก่าเป็นคนเดียวกัน

##### วิธีแก้ปัญหาระบบทะเบียนนักศึกษาเดิม

- 1 ปัญหาในเรื่องของรหัสวิชากับชื่อวิชาสามารถเปลี่ยนแปลงได้  
วิธีแก้ปัญหา โดยการทำเป็นฐานข้อมูลเชิงเวลา
- 2 ปัญหาในเรื่องของการตรวจสอบผลการศึกษา  
วิธีแก้ปัญหา โดยการใช้Object-Oriented Databaseมาแก้ปัญหา โดยใช้อ้างอิง Oid ของวิชา แทนการใช้รหัสวิชา
- 3 ปัญหาในเรื่องของการเปลี่ยนแปลงข้อมูลของนักศึกษา  
วิธีแก้ปัญหา เช่นเดียวกับข้อ2 อ้างอิงผ่าน Oid แทนรหัสนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 6.4 ความสามารถของระบบทะเบียนนักศึกษาที่พัฒนา

- 1 สามารถตรวจสอบการสำเร็จการศึกษาของนักศึกษาได้ เมื่อนักศึกษามีการลงวิชาในหลักสูตรใหม่
- 2 สามารถเก็บข้อมูลของวิชาที่มีการเปลี่ยนแปลง
- 3 สามารถออกรหัสวิชา, ชื่อวิชาเองได้รวมไปถึงสามารถเปลี่ยนรหัสนักศึกษาได้
- 4 มีความสามารถของระบบทะเบียนนักศึกษาเดิม

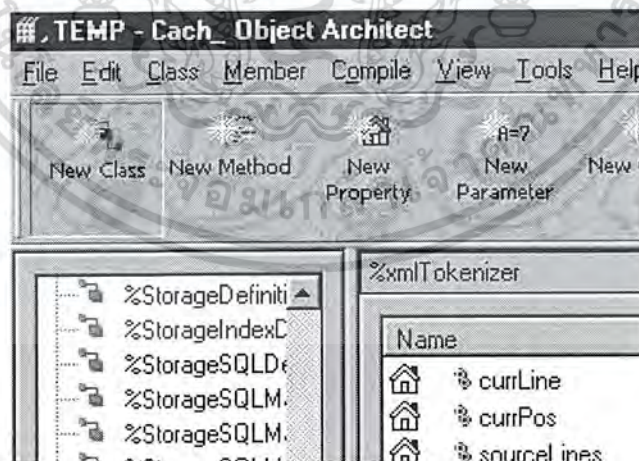
#### 6.5 เทคนิคที่ใช้ในการแก้ไขปัญหของระบบทะเบียนนักศึกษา

สำหรับในการพัฒนาระบบทะเบียนนักศึกษานี้จะใช้เทคนิคต่างๆดังต่อไปนี้

- 1 เราจะใช้การอ้างอิงผ่านOidแทนของออบเจกต์ในกรณีที่ออบเจกต์นั้นมีแอททริบิวต์ที่สามารถเปลี่ยนแปลงตามเวลาได้
- 2 สำหรับแอททริบิวต์ที่มีการเปลี่ยนแปลงตามเวลา เราจะทำให้การเปลี่ยนแปลงเกิดขึ้นกับแอททริบิวต์นั้น แทนที่เราจะต้องทำการสร้างแถวขึ้นมาใหม่
- 3 สำหรับโครงสร้างของแอททริบิวต์ที่ต้องการเปลี่ยนแปลงตามเวลานั้นเราจะให้ชนิดของแอททริบิวต์นั้นเป็นคลาสของเอ็มเบดออบเจกต์ที่เราได้สร้างมารองรับกับข้อมูลเชิงเวลา โดยมีการเก็บแต่ละเอ็มเบดออบเจกต์เป็นแบบลิสต์ ในแอททริบิวต์นั้น
- 4 สำหรับคลาสของเอ็มเบดออบเจกต์นี้เราจะตั้งชื่อเป็นTempString ซึ่งรองรับการเก็บข้อมูลชนิดสตริงเชิงเวลา

#### 6.6 ขั้นตอนการสร้างTempString

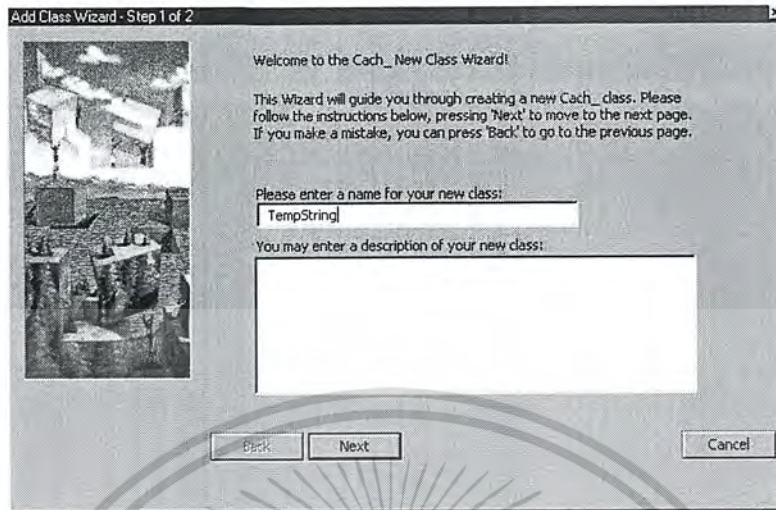
- 1 ให้ทำการกดปุ่มNew Class ทางด้านบน



รูปที่ 6-2 แสดงการกดปุ่มNew Class ทางด้านบน

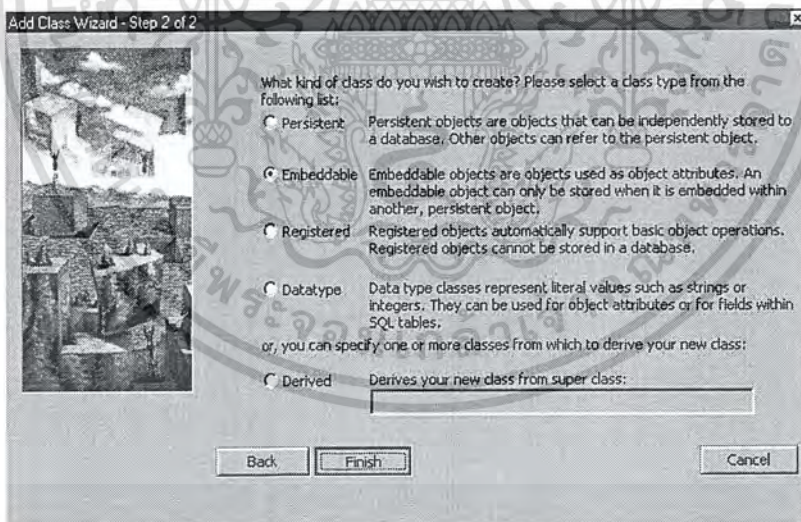
- 2 จะปรากฏหน้าต่างAdd Class Wizard จากนั้นให้ทำการใส่ชื่อคลาสเป็นTempString จากนั้นให้ทำการกดปุ่ม Next

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-3 แสดงหน้าต่าง Add Class Wizard หน้าแรก

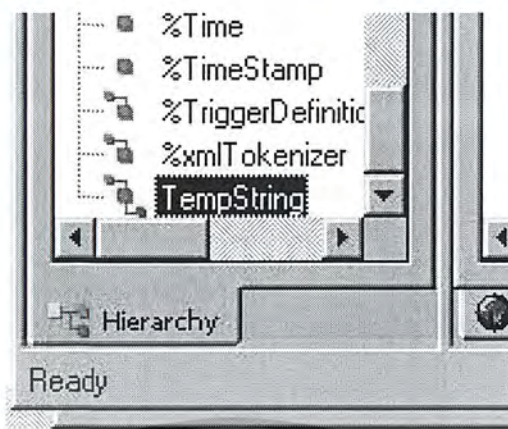
- 3 ในขั้นตอนต่อมาโปรแกรมจะถามถึงชนิดของคลาสที่ต้องการสร้างให้ทำการเลือก Embeddable จากนั้นก็ให้คลิกปุ่ม Finish



รูปที่ 6-4 แสดงหน้าต่าง Add Class Wizard หน้าสอง

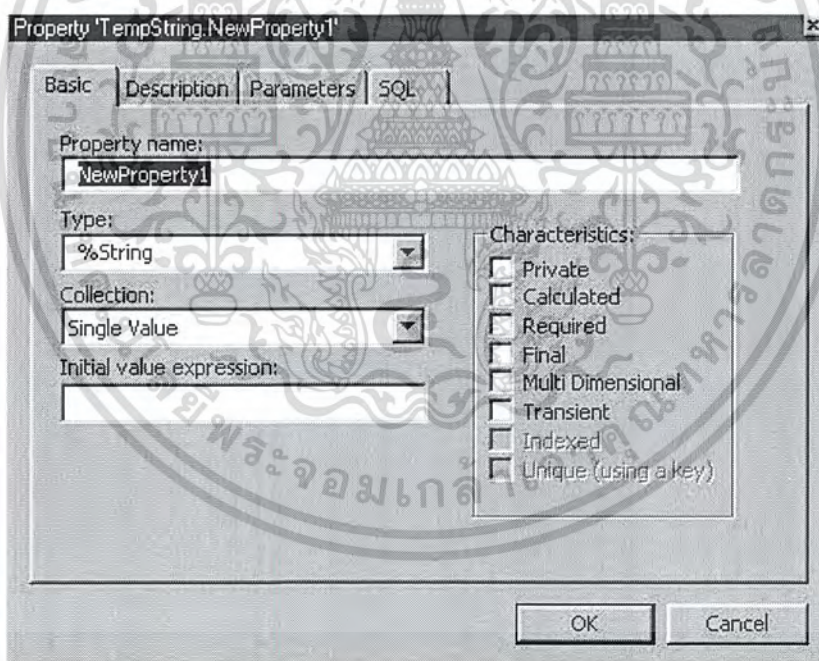
- 4 จากนั้น โปรแกรมจะทำการสร้างคลาส TempString

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-5 แสดงคลาสTempStringตามที่ได้ทำการสร้าง

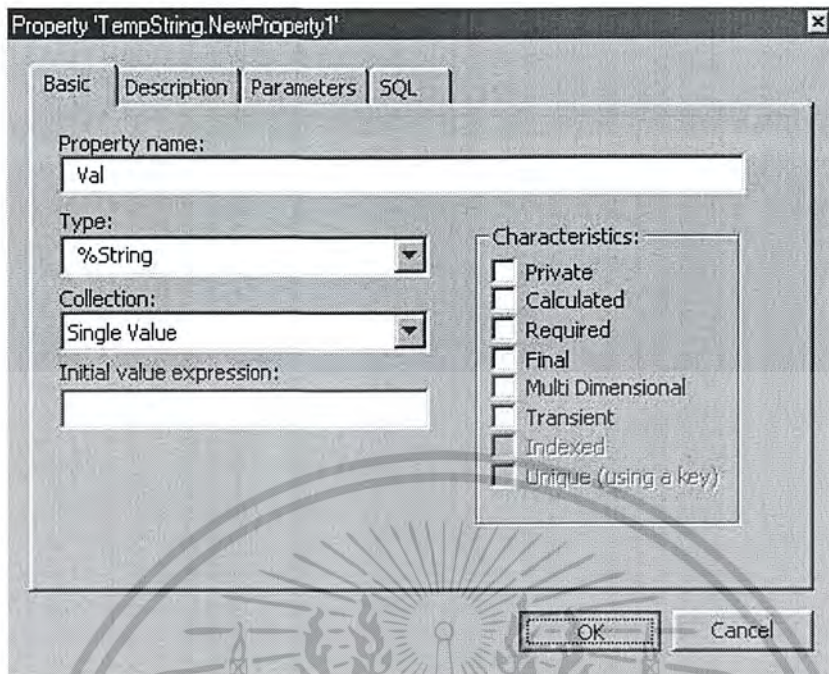
- 5 ให้ทำการกำหนดProperty โดยให้ทำการกดปุ่ม New Propertyทางด้านบน



รูปที่ 6-6 แสดงหน้าต่างการเพิ่มProperty

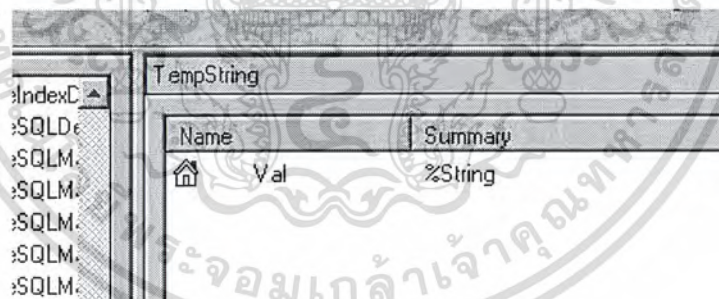
- 6 จากนั้นให้ทำการใส่Property เลือก Name Type เลือก Collection แล้วทำการกดปุ่ม OK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-7 แสดงหน้าต่างการใส่รายละเอียดเกี่ยวกับProperty

- 7 โปรแกรมจะทำการสร้างPropertyตามที่เราได้กำหนด



รูปที่ 6-8 แสดงการเพิ่มProperty เข้าไปในคลาส

- 8 ให้ทำการสร้างPropertyของคลาสดังต่อไปนี้

| Property Name | Type   | Collection   |
|---------------|--------|--------------|
| Val           | String | Single Value |
| FromDate      | Date   | Single Value |
| ToDate        | Date   | Single Value |

ตารางที่ 6-9 แสดงรายละเอียดของPropertyในคลาสTempString

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| Name     | Summary |
|----------|---------|
| FromDate | %Date   |
| ToDate   | %Date   |
| Val      | %String |

รูปที่ 6-9 แสดงPropertyในคลาสTempString

- 9 จะได้คลาสTempStringซึ่งเราจะสามารถนำไปใช้กำหนดTypeให้กับแอททริบิวต์ที่ต้องการให้เก็บการเปลี่ยนแปลงเชิงเวลาได้
- 10 สำหรับสตริงในค่านั้นสามารถที่นำมาเก็บข้อมูลที่เป็นInteger,Float,Char ฯลฯ ได้ และสามารถนำมาคำนวณได้เหมือนปกติ

#### 6.7 ขั้นตอนการนำTempStringไปกำหนดให้กับแอททริบิวต์

- 1 เราจะนำTempString ไปกำหนดให้กับแอททริบิวต์ที่ต้องการเปลี่ยนแปลงตามเวลา
- 2 จากคลาสที่เราต้องการให้มีแอททริบิวต์ที่ต้องการให้เปลี่ยนแปลงตามเวลาให้เราทำการสร้างProperty นั้นก่อน
- 3 จากนั้นตรงช่องของTypeให้เรากำหนดเป็น TempString
- 4 จากนั้นตรงช่องของCollectionให้เรากำหนดเป็น List

รูปที่ 6-10 แสดงPropertyที่เก็บการเปลี่ยนแปลงเชิงเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

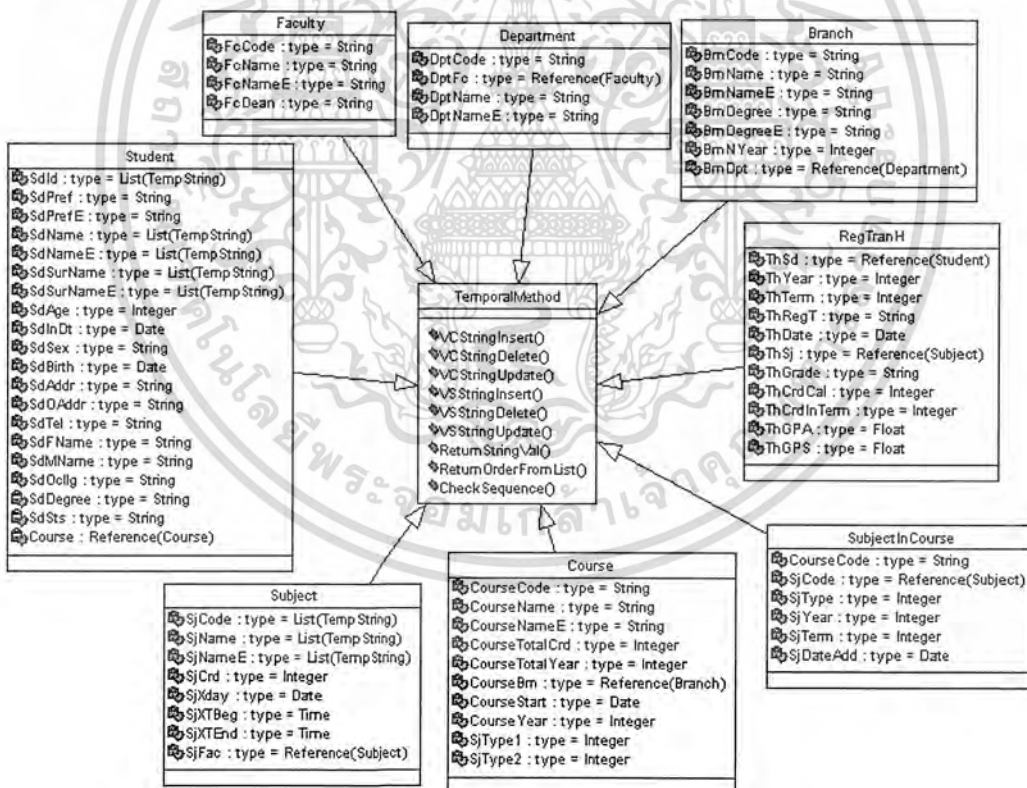
5 ให้ทำการคดปุ่มOK โปรแกรมจะทำการสร้างแอททริบิวต์ตามที่เราได้ทำการกำหนด



รูปที่ 6-11 แสดงการสร้างแอททริบิวต์ที่เก็บการเปลี่ยนแปลงเชิงเวลา

## 6.8 รายละเอียดของเมธอดกลาง

สำหรับในการพัฒนาระบบทะเบียนนักศึกษา นั้นเราจะทำการพัฒนาโดยเราจะทำการสร้างเมธอดกลางขึ้นมาเพื่อให้ผู้พัฒนาสามารถเรียกใช้เมธอดกลางได้เลย โดยไม่ต้องทำการสร้างใหม่ในแต่ละคลาส โดยในการพัฒนานี้เราจะกำหนดคลาส TemporalMethod เป็นที่ทำการเก็บเมธอดกลาง



รูปที่ 6-12 รูปคลาสไดอะแกรมแสดงความสัมพันธ์ระหว่างคลาสในระบบงานทะเบียน

นักศึกษากับคลาสเทมโพรอลเมธอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาสเทมโพรอลเมธอด(TemporalMethod) เป็นคลาสที่เป็นMethodกลาง มีไว้สำหรับให้ผู้พัฒนาสามารถเรียกใช้ได้โดยไม่ต้องเขียนใหม่ โดยตัวแปรAttrที่เป็นอาร์กิวเมนต์ในทุกเมธอดนั้นเป็นตัวแปรที่จะรับลิสต์แอททริบิวต์มาจัดการตามแต่ละเมธอด โดยในคลาสเทมโพรอลเมธอด นี้จะมี Method ที่สำคัญๆ ในการจัดการเกี่ยวกับเทมโพรอล แอปพลิเคชัน โดยเมธอดที่ได้ยกมาเป็นเมธอดที่เกี่ยวกับสตริง ซึ่งมีดังต่อไปนี้

เมธอดเช็ควิธีซ้อน(CheckSequence) เป็นเมธอดกลางที่ให้ผู้พัฒนาใช้สำหรับการตรวจสอบการซ้ำซ้อนเชิงเวลา จากเงื่อนไขที่ว่า ค่าความจริงเดียวกัน จะมีปรากฏอยู่แค่เพียงช่วงเวลาเดียว โดยเมธอดนี้จะทำการค้นหาว่ามีการซ้ำซ้อนเกิดขึ้นหรือไม่ภายในลิสต์ของวัตถุเดียวกัน กับค่าที่ต้องการอินเซิร์ตหรือว่าการแก้ไข โดยถ้าตรวจพบการซ้ำซ้อนก็จะส่งค่ากลับมาเป็น 1 หากไม่ซ้ำซ้อนก็จะส่งค่า 0 กลับคืนมา มีอาร์กิวเมนต์ที่ใช้ได้แก่ Attr, Val , fromDate, toDate

```
new SDup,count,TempFromDate,TempToDate,i
set count=Attr.Count()
set SDup=0
if count>0 quit SDup
for i=1:1:count do
. set TempFromDate=Attr.GetAt(i).FromDate
. set TempToDate=Attr.GetAt(i).ToDate
. if Attr.GetAt(i).Val=SdId do
. . if fromDate<TempToDate do
. . . if TempFromDate<toDate do
. . . . set SDup=1
quit SDup
```

เมธอดวิธีสตริงอินเซิร์ต(VCStringInsert) เป็นเมธอดกลางที่ให้ผู้พัฒนาใช้สำหรับการอินเซิร์ตแบบเคอร์เนลพหุคูณ โดยเมธอดวิธีสตริงอินเซิร์ตสามารถเขียนเป็นโค้ดด้วยภาษาคาเชออบเจ็กต์สคริปต์ได้ดังต่อไปนี้

สำหรับอาร์กิวเมนต์ที่ใช้ Attr,ValInsert

```
new Temp,check,message
set message="Current Insert Successful"
set check=$$CheckCurrentDuplicate(Attr)
if check=0 do
. set Temp=##class(TempString).%New()
. set Temp.Val=ValInsert
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

. set Temp.FromDate=$h
. set Temp.ToDate=2980013
. do Attr.Insert(Temp)
else do
. set message="Have CurrentDuplicate"
. quit
quit message

```

**CheckCurrentDuplicate** เป็นฟังก์ชันสำหรับการเช็คเคอเรนดูพลีเค็ทที่เกิดขึ้น ว่า ณ ปัจจุบันค่าที่จะอินเซิร์ตใส่ในนั้นมีค่าซ้ำหรือไม่ ถ้ามีก็จะไม่มีการอินเซิร์ต แต่ถ้าไม่มีเมธอดวีซีสตริงอินเซิร์ตสก็จะเซตค่าในวันเริ่มต้นเป็นวันปัจจุบันแต่ในวันสิ้นสุดเป็นฟอร์ออฟเวอร์

```

CheckCurrentDuplicate(Attr)
new CDup,count,TempFromDate,TempToDate,i
set count=Attr.Count()
set CDup=0
for i=1:1:count do
. set TempFromDate=Attr.GetAt(i).FromDate
. set TempToDate=Attr.GetAt(i).ToDate
. if '(TempFromDate>$h) do
.. if $h<TempToDate do
... set CDup=1
... quit
.. quit
. quit
quit Cdup

```

**เมธอดวีซีสตริงดีลิต(VCStringDelete)** เป็นเมธอดกลางที่ให้ผู้พัฒนาใช้สำหรับการดีลิตแบบเคอเรนดูพลีเค็ท โดยเมธอดวีซีสตริงดีลิต สามารถเขียนเป็นโค้ดด้วยภาษาคาเชออบเจ็กต์สคริปได้ดังต่อไปนี้

สำหรับอากิวเมนต์ที่ใช้ Attr

```

new count,i,TempFromDate,TempToDate,pos,message
set count=Attr.Count()

```

```

set message="Current Delete Successful"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;update ToDate -> CurrentDate
for i=1:1:count do
. set TempToDate=Attr.GetAt(i).ToDate
. if '(TempToDate<$h) do
.. set TempFromDate=Attr.GetAt(i).FromDate
.. if TempFromDate<$h do
... set Attr.GetAt(i).ToDate=$h
... quit
.. quit
. quit

```

```

;Delete List that FromDate

```

```

for do quit:pos=0

```

```

. set pos=$$posDel(Attr)

```

```

. do Attr.RemoveAt(pos)

```

```

. quit

```

```

quit message

```

**posDel** เป็นฟังก์ชันสำหรับการหาตำแหน่งที่จะทำการลบในลิสต์

```

posDel(TempAttr)

```

```

new result,TempCount,j,TempFD

```

```

set TempCount=TempAttr.Count()

```

```

set result=0

```

```

for j=1:1:TempCount do

```

```

. if result=0 do

```

```

...set TempFD=TempAttr.GetAt(j).FromDate

```

```

..if TempFD>$h do

```

```

...set result=j

```

```

...quit

```

```

..quit

```

```

. quit

```

```

quit result

```

โดยเมธอดวีซีสตริงคือลิสต์นั้นจะทำการอัปเดตลิสต์ของช่วงเวลาในวันปัจจุบันอยู่ในช่วงเวลาของลิสต์ เมธอดจะทำการเซตวันสิ้นสุดเป็นวันปัจจุบันจากนั้นจะทำการลบลิสต์ที่ตำแหน่งที่ posDel ี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมธอดวีซีสตริงอัปเดต(VCStringUpdate) เป็นเมธอดกลางที่ให้ผู้พัฒนาใช้สำหรับการอัปเดตแบบเคเร็นคูปพลิเค็ท โดยเมธอดวีซีสตริงอัปเดต สามารถเขียนเป็นโค้ดด้วยภาษาคาเชออบเจ็กต์สคริปได้ดังต่อไปนี้

สำหรับอากิวเมนต์ที่ใช้ Attr,NewVal

```

;Insert
new Temp,count,i,TempFromDate,TempToDate,message,tempcount
set message="Current Update Successful"
set count=Attr.Count()
set tempcount=0
for i=1:1:count do
. set TempFromDate=Attr.GetAt(i).FromDate
. set TempToDate=Attr.GetAt(i).ToDate
. if '(TempFromDate>$h) do
.. if TempToDate>$h do
... set Temp=##class(TempString).%New()
... set Temp.Val=NewVal
... set Temp.FromDate=$h
... set Temp.ToDate=TempToDate
... set tempcount=i+1
... quit
.. quit
. quit
;arrange
if tempcount'=0 do
. do Attr.InsertAt(Temp,tempcount)
. quit
;Update ToDate
set count=Attr.Count()
for i=1:1:count do
. set TempToDate=Attr.GetAt(i).ToDate
. if TempToDate>$h do
.. set TempFromDate=Attr.GetAt(i).FromDate

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

... set Attr.GetAt(i).ToDate=$h
... quit
.. quit
. quit
;Update Attr
set count=Attr.Count()
for i=1:1:count do
. set TempFromDate=Attr.GetAt(i).FromDate
. if '(TempFromDate<$h) do
.. set Attr.GetAt(i).Val=NewVal
.. quit
. quit
quit message

```

เมธอดวีเอสสตริงอินเซิร์ต(VSStringInsert) เป็นเมธอดกลางที่ให้ผู้พัฒนาใช้สำหรับการอินเซิร์ตแบบซีเคิร์นซ์คู่พหุสิทธิ์ โดยเมธอดวีเอสสตริงอินเซิร์ตสามารถเขียนเป็นโค้ดด้วยภาษาคาเชออบเจ็กต์สคริปได้ดังต่อไปนี้

สำหรับอากิวเมนต์ที่ใช้ Attr,Val,FromDate,ToDate

```

new Temp,check,message,TempFromDate,tempcount,count,i
set message="Sequenced Insert Successful"
set tempcount=0
set count=Attr.Count()
set check=$$CheckFromDateToDate(FromDate,ToDate)
if check=0 do
. set check=$$CheckSequencedDuplicate(Attr,FromDate,ToDate)
. if check=0 do
.. set Temp=##class(TempString).%New()
.. set Temp.Val=Val
.. set Temp.FromDate=FromDate
.. set Temp.ToDate=ToDate
.. if count=0 do
... do Attr.Insert(Temp)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

... quit
.. else do
... for i=1:1:count do
.... set TempFromDate=Attr.GetAt(i).FromDate
.... set TempToDate=Attr.GetAt(i).ToDate
.... if ToDate<TempFromDate do
..... if tempcount=0 do
..... set tempcount=i
..... quit
..... quit
.... if i=count do
..... if FromDate>TempToDate do
..... do Attr.Insert(Temp)
..... quit
..... quit
.... quit
.. if tempcount'=0 do
... do Attr.InsertAt(Temp,tempcount)
... quit
.. quit
. else do
.. set message="Have Sequenced Duplicate"
.. quit
. quit
else do
. set message="FromDate>ToDate"
quit message

```

**CheckFromDateToDate** เป็นฟังก์ชันที่ใช้ในการตรวจสอบว่าวันเริ่มต้นมากกว่าวันสิ้นสุดหรือไม่ ถ้าวันเริ่มต้นมากกว่าวันสิ้นสุด แล้วเมธอดวีเอสสตรีงอินเชิร์ทจะออกจากการทำงาน โดยถือว่าผู้ใช้เมธอดให้ค่าวันเริ่มต้นและวันสิ้นสุดผิด โดยวันเริ่มต้นจะต้องไม่มากกว่าวันสิ้นสุด

```
CheckFromDateToDate(FromDate, ToDate)
```

```
new result
```

```
if FromDate>ToDate do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

.set result=1
.quit
else do
.set result=0
.quit
quit result

```

**CheckSequencedDuplicate** เป็นฟังก์ชันที่จะใช้ในการตรวจสอบว่าค่าวันเริ่มต้นและวันสิ้นสุดนั้นเป็นค่าที่เกิดซ้ำซ้อนหรือไม่

CheckSequencedDuplicate(Attr,FromDate,ToDate)

```

new SDup,count,TempFromDate,TempToDate,i
set count=Attr.Count()
set SDup=0
for i=1:1:count do
.set TempFromDate=Attr.GetAt(i).FromDate
.set TempToDate=Attr.GetAt(i).ToDate
.if FromDate<TempToDate do
.. if TempFromDate<ToDate do
... set SDup=1
... quit
.. quit
.quit
quit SDup

```

**เมธอดวีเอสสตริงดีลิต (VSStringDelete)** เป็นเมธอดกลางที่ให้ผู้พัฒนาใช้สำหรับการดีลิตแบบซีเควินซ์ดูพลิต์ โดยเมธอดวีเอสสตริงดีลิตสามารถเขียนเป็นโค้ดด้วยภาษาคาเชออบเจ็กต์สคริปได้ดังต่อไปนี้

สำหรับอากิวเมนต์ที่ใช้ Attr,paFromDate,paToDate

```

new Temp,count,i,TempFromDate,TempToDate,Val,message,tempcount
set message="Sequenced Delete Successful"
set tempcount=0
set count=Attr.Count()
;Insert Case1
for i=1:1:count do

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

. set TempFromDate=Attr.GetAt(i).FromDate
. set TempToDate=Attr.GetAt(i).ToDate
. if '(TempFromDate>paFromDate) do
.. if TempToDate>paToDate do
... set Val=Attr.GetAt(i).Val
... set Temp=##class(TempString).%New()
... set Temp.Val=Val
... set Temp.FromDate=paToDate
... set Temp.ToDate=TempToDate
... set tempcount=i+1
... quit
.. quit
. quit
if tempcount!=0 do
. do Attr.InsertAt(Temp,tempcount)
. quit
;Update Case1 or Case2
set count=Attr.Count()
for i=1:1:count do
. set TempFromDate=Attr.GetAt(i).FromDate
. set TempToDate=Attr.GetAt(i).ToDate
.. if TempFromDate<paFromDate do
.. if '(TempToDate<paFromDate) do
... set Attr.GetAt(i).ToDate=paFromDate
... quit
.. quit
. quit
;Update Case3
set count=Attr.Count()
for i=1:1:count do
. set TempFromDate=Attr.GetAt(i).FromDate
. set TempToDate=Attr.GetAt(i).ToDate
. if TempFromDate<paToDate do
.. if '(TempToDate<paToDate) do

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

... set Attr.GetAt(i).FromDate=paToDate
... quit
.. quit
. quit
;Delete Case4
for do quit:pos=0
.set pos=$$SposDel(Attr,paFromDate,paToDate)
.do Attr.RemoveAt(pos)
.quit
quit message

```

SposDel เป็นฟังก์ชันที่ใช้หาตำแหน่งที่จะใช้ลบ โดยฟังก์ชันนี้จะให้ตำแหน่งของลิสต์ที่จะลบ

```

SposDel(TempAttr,paFromDate,paToDate)
new result,TempCount,j,TempFromDate,TempToDate
set TempCount=TempAttr.Count()
set result=0
for j=1:1:TempCount do
. if result=0 do
.. set TempFromDate=TempAttr.GetAt(j).FromDate
.. set TempToDate=TempAttr.GetAt(j).ToDate
.. if '(TempFromDate<paFromDate) do
... if '(TempToDate>paToDate) do
.... set result=j
.... quit
... quit
..quit
. quit
quit result

```

เมธอดวีเอสสตริงอัปเดต (VStringUpdate) เป็นเมธอดกลางที่ให้ผู้พัฒนาใช้สำหรับการอัปเดตแบบซีเคิร์นซ์ดิวพลีเค็ท โดยเมธอดวีเอสสตริงอัปเดต สามารถเขียนเป็นโค้ดด้วยภาษาคาเชออบเจ็กต์สคริปได้ดังต่อไปนี้

สำหรับอากิวเมนต์ที่ใช้ Attr,NewVal,paFromDate,paToDate

;Insert the old value from the FromDate to the begining pa  
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

new Temp,count,i,TempFromDate,TempToDate,Val,message,tempcount
set message="Sequenced Update Successful"
set count=Attr.Count()
set tempcount=0
for i=1:1:count do
. set TempFromDate=Attr.GetAt(i).FromDate
. set TempToDate=Attr.GetAt(i).ToDate
. if tempcount=0 do
.. if TempFromDate<paFromDate do
... if TempToDate>paFromDate do
.... set Val=Attr.GetAt(i).Val
.... set Temp=##class(TempString).%New()
.... set Temp=$$FirstInsert(Temp,Val,TempFromDate,paFromDate)
.... set tempcount=i
.... do Attr.InsertAt(Temp,tempcount)
.... quit
... quit
.. quit
. quit
;Insert the old values from the end of the pa to ToDate
set count=Attr.Count()
set tempcount=0
for i=1:1:count do
. set TempFromDate=Attr.GetAt(i).FromDate
. set TempToDate=Attr.GetAt(i).ToDate
. if tempcount=0 do
.. if TempFromDate<paToDate do
... if TempToDate>paToDate do
.... set Val=Attr.GetAt(i).Val
.... set Temp=##class(TempString).%New()
.... set Temp=$$SecondInsert(Temp,Val,paToDate,TempToDate)
.... set tempcount=i+1
.... do Attr.InsertAt(Temp,tempcount)
.... quit

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

... quit
.. quit
. quit
;Update the explicit column of rows that overlap the pa
set count=Attr.Count()
for i=1:1:count do
. set TempFromDate=Attr.GetAt(i).FromDate
. set TempToDate=Attr.GetAt(i).ToDate
. if TempFromDate<paToDate do
.. if TempToDate>paFromDate do
... set Attr.GetAt(i).Val=NewVal
... quit
.. quit
. quit
;Update the FromDate to begin at the beginning of the end of the pa
set count=Attr.Count()
for i=1:1:count do
. set TempFromDate=Attr.GetAt(i).FromDate
. set TempToDate=Attr.GetAt(i).ToDate
. if TempFromDate<paFromDate do
.. if TempToDate>paFromDate do
... set Attr.GetAt(i).FromDate=paFromDate
... quit
.. quit
. quit
;Update the ToDate to end at the end of the pa of rows that overlap the pa
set count=Attr.Count()
for i=1:1:count do
. set TempFromDate=Attr.GetAt(i).FromDate
. set TempToDate=Attr.GetAt(i).ToDate
. if TempFromDate<paToDate do
.. if TempToDate>paToDate do
... set Attr.GetAt(i).ToDate=paToDate

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
.. quit
. quit
quit message
```

**FirstInsert** เป็นการอินเซิร์ตค่าในกรณีที่ค่าวันเริ่มต้นในลิสต์น้อยกว่าค่าวันเริ่มต้นที่เราต้องการอัปเดต

```
FirstInsert(instant,Val,TempFromDate,paFromDate)
```

```
set instant.Val=Val
set instant.FromDate=TempFromDate
set instant.ToDate=paFromDate
quit instant
```

**SecondInsert** เป็นการอินเซิร์ตค่าในกรณีที่ค่าวันสิ้นสุดในลิสต์มากกว่าค่าวันสิ้นสุดที่เราต้องการอัปเดต

```
SecondInsert(instant,Val,paToDate,TempToDate)
```

```
set instant.Val=Val
set instant.FromDate=paToDate
set instant.ToDate=TempToDate
quit instant
```

**NonSeqDel** เป็นเมธอดจะทำการลบล้างในลักษณะของเนียนซีควีนซ์ดัพพลิเกทสำหรับอาทิวเมนต์ที่ใช้ Attr,period

```
new pos
for do quit:pos=0
.set pos=$$NonSposDel(Attr,period)
.do Attr.RemoveAt(pos)
.quit
quit
```

**NonSposDel** หาดำเนินที่จะลบในลิสต์

```
NonSposDel(TempAttr,period)
```

```
new result,TempCount,j,TempFromDate,TempToDate,time
set TempCount=TempAttr.Count()
```

```
set result=0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for j=1:1:TempCount do
. if result=0 do
.. set TempFromDate=TempAttr.GetAt(j).FromDate
.. set TempToDate=TempAttr.GetAt(j).ToDate
.. set time=TempToDate-TempFromDate
.. if time>period do
... set result=j
... quit
.. quit
. quit
quit result

```

ในคลาสเทมโพรอลเมธอดนี้ยังมีเมธอดอื่นให้สามารถเลือกใช้ได้ โดยมีรายละเอียดดังต่อไปนี้  
 เมธอดรีเทิร์นสตริงวาล(ReturnStringVal) เป็นเมธอดที่จะให้ค่าในลิสต์ของวันที่เราต้องการ  
 สำหรับอาทิวเมนต์ที่ใช้ Attr,DateAtFind

```

new count,TempFromDate,TempToDate,Val,Temp,found,i
set count=Attr.Count()
set found=0
for i=1:1:count do
. set TempFromDate=Attr.GetAt(i).FromDate
. set TempToDate=Attr.GetAt(i).ToDate
. if DateAtFind<TempFromDate do
.. if DateAtFind>TempToDate do
... set Temp=Attr.GetAt(i).Val
... set found=1
... quit
.. quit
. quit
if found=0 do
. set Temp="not found"
. quit
quit Temp

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมธอดรีเทิร์นออเดอร์ฟอร์มลิสต์(ReturnOrderFromList) ให้ลำดับที่ในListที่อยู่ในDate โดยเราให้Dateที่ต้องการหา

```

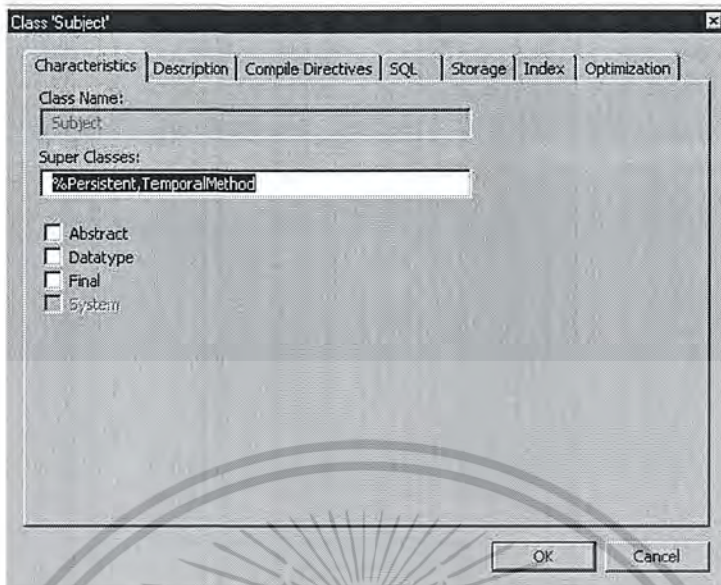
new count,TempFromDate,TempToDate,Val,Temp,found,i
set count=Attr.Count()
set found=0
for i=1:1:count do
. set TempFromDate=Attr.GetAt(i).FromDate
. set TempToDate=Attr.GetAt(i).ToDate
. if DateAtFind<TempFromDate do
.. if DateAtFind>TempToDate do
... set Temp=i
... set found=1
... quit
.. quit
. quit
if found=0 do
. set Temp="not found"
. quit
quit Temp

```

ในส่วนของแอททริบิวต์ที่เป็นลิสต์นั้น ก่อนที่เราจะให้ค่ากับลิสต์นั้นเราจะต้องทำการสร้างเอ็มเบดออบเจกต์ก่อน โดยเอ็มเบดออบเจกต์ที่เราได้ทำนี้จะรวมเอาค่าแอททริบิวต์ไว้ในเอ็มเบดออบเจกต์ โดยแอททริบิวต์ที่อยู่ในเอ็มเบดออบเจกต์มีดังนี้ Val,FromDate,ToDate โดยเราจะทำเป็นเหมือนกับชนิดข้อมูลกลาง เช่นTempString จะเป็นเสมือนชนิดข้อมูลที่เป็นเทมโพลกลางที่เป็นแบบสตริงให้เราเรียกใช้เพียงแต่เราอาจจะต้องกำหนดเป็นลิสต์เพื่อให้มันสามารถเก็บการเปลี่ยนแปลงตามเวลาในแบบที่เป็นลิสต์ได้

การใช้คลาสเทมโพลเมธอด สำหรับการใช้คลาสเทมโพลเมธอด นั้นเราจะต้องทำดังนี้  
 1 ต้องทำการอินเฮอริตจากคลาสเทมโพลเมธอด โดยทำดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-13 แสดงการอินเฮอริตคลาสคลาสเทมโพรอลเมธอด

จากรูปคลาสซบเจตต้องการใช้คลาสเทมโพรอลเมธอด จึงต้องทำการอินเฮอริตมาก่อน 2 จากนั้นหากแอททริบิวต์ใดต้องการใช้ต้องสร้างโอเรฟของแอททริบิวต์นั้นก่อน เช่น

```
new SubjectOBJ,SubAttr,message,ok
```

```
set SubjectOBJ=##class(Subject).%OpenId(Oid)
```

set SubAttr=SubjectOBJ.SjCode ทำการกำหนดโอเรฟมายังแอททริบิวต์ที่ต้องการ โดยค่านี้จะส่งให้กับอากิวเมนต์ที่ชื่อAttrในเมธอดของคลาสเทมโพรอลเมธอด

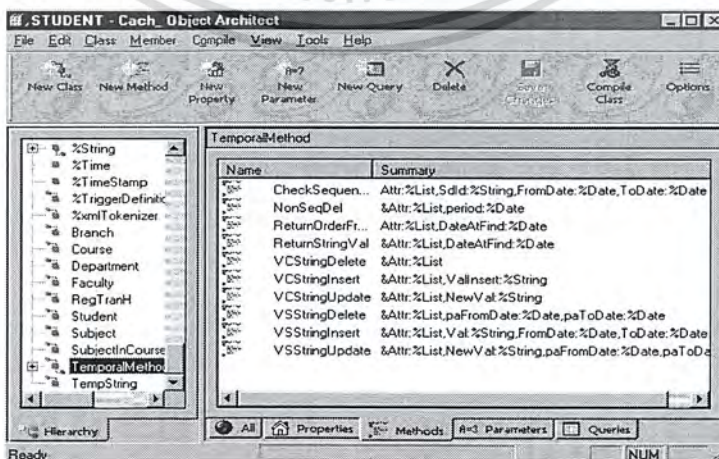
```
set message=SubjectOBJ.VSStringInsert(SubAttr,SubCode,FromDate,ToDate)
```

```
write !,message
```

```
set ok=SubjectOBJ.%Save()
```

```
do SubjectOBJ.%Close()
```

```
quit message
```



รูปที่ 6-14 แสดงรายละเอียดของเมธอดที่มีในคลาสTemporalMethod

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 7

## การพัฒนาโปรแกรม

### 7.1 สภาพแวดล้อมของระบบ

#### 7.1.1 โปรแกรมที่ใช้ในการติดตั้งเพื่อทำงานกับฐานข้อมูล Cache'

##### Server

- Cache PC version 3.2

##### Client

- ActiveX โดยใช้ Visual Basic 6.0

โดยที่เครื่อง Client จะมีความต้องการของระบบ (System Requirements) ดังนี้

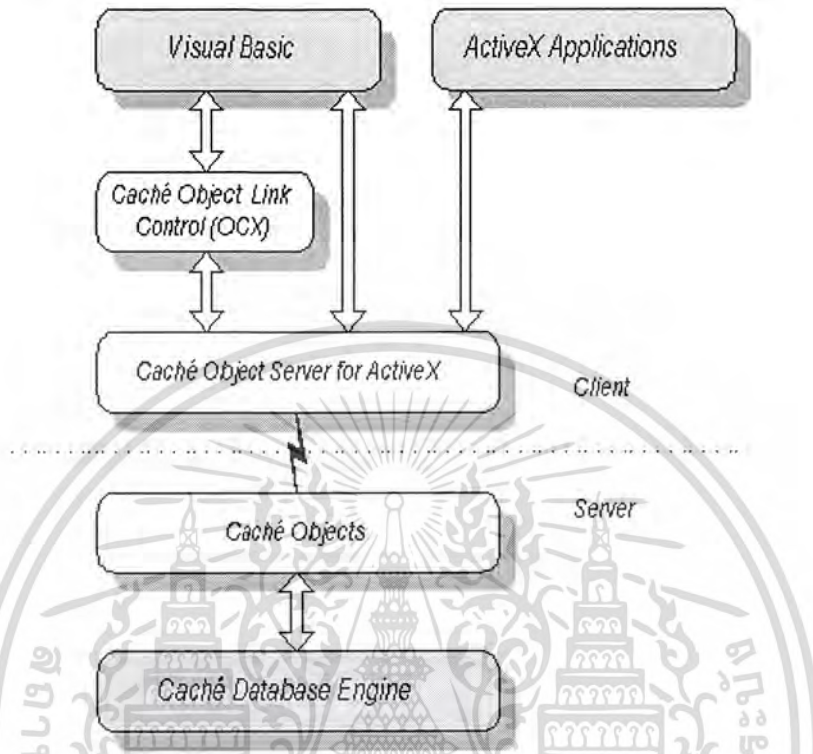
| Requirement     | Client   | Server   |
|-----------------|--|--|
| Hardware        | ไมโครโปรเซสเซอร์ 80486<br>หรือสูงกว่า                            | ไมโครโปรเซสเซอร์ 80486<br>หรือสูงกว่า                            |
| Windows Version | Windows 98 SE  | Windows 98 SE  |
| RAM             | 64 MB  | 64 MB  |
| Hard-disk       | 2 GB (หรือมากกว่านี้ ขึ้นอยู่กับ Component ที่เลือกทำการติดตั้ง) | 2 GB (หรือมากกว่านี้ ขึ้นอยู่กับ Component ที่เลือกทำการติดตั้ง) |

ตารางที่ 7-1 ความต้องการของเครื่องที่ใช้ในการพัฒนา

การพัฒนาโปรแกรมเพื่อใช้ในการติดต่อกับฐานข้อมูลในครั้งนี้ใช้ Visual Basic เขียนขึ้น โดยอาศัยข้อดีของการที่ ฐานข้อมูลเป็นฐานข้อมูลเชิงวัตถุแล้วใช้ภาษาโปรแกรมมิ่งที่ลักษณะเป็นภาษาเชิงวัตถุ นอกจากนี้ ฐานข้อมูลที่ใช้ยังมีการผนวกเอาคุณสมบัติของ ActiveX เข้าไปไว้ด้วย ทำให้การพัฒนา ด้วยการ ใช้ Visual Basic นั้นสะดวก ผู้จัดทำโครงการจึงได้ใช้ Visual Basic ในการพัฒนา นอกจากจะสามารถพัฒนาด้วย Visual Basic แล้ว ยังสามารถพัฒนาด้วย Delphi , C++ , JAVA และยังสามารถพัฒนาโดยใช้เทคโนโลยีของอินเทอร์เน็ตในการติดต่อกับฐานข้อมูลนี้ได้ด้วย อนึ่งการถ้าต้องการความเร็วในการทำงานควรพัฒนาด้วย C++ และหากเครื่องไคลเอนท์ที่ใช้มีความหลากหลายของฮาร์ดแวร์และระบบปฏิบัติการก็ควรจะใช้ JAVA หรือถ้าต้องการ ให้ผู้ใช้สามารถเข้ามาใช้งานจากที่ไกลๆ ได้การใช้ อินเทอร์เน็ตก็เป็นอีกทางเลือกหนึ่ง ที่เหมาะสม

## 7.2 การติดต่อระหว่าง Cache' และ ActiveX

การเขียนโปรแกรมติดต่อกับฐานข้อมูล Cache' โดยการใช้ ActiveX นั้น มีเครื่องมือให้ใช้ในการพัฒนามากมาย เช่น Visual Basic, Delphi , C++ เป็นต้น



รูปที่ 7-1 แสดงการติดต่อระหว่าง Cache' Object Server สำหรับ ActiveX

Cache' Objects จะประกอบไปด้วยคอมโพเนนต์ต่างๆ ได้แก่

- Cache' Object Server สำหรับ ActiveX เป็นตัวที่คอยรอรับการติดต่อจาก ActiveX แล้วทำการส่ง objects ที่ฝั่ง Server ไปให้เป็น ActiveX objects ที่ทางฝั่ง Client
- Visual Cache' เป็นชุดของเครื่องมือที่ติดมาให้สำหรับอำนวยความสะดวกในการพัฒนาด้วย Visual Basic ซึ่งเครื่องมือดังกล่าวจะประกอบไปด้วย Cache' Object Link Control, Cache' List Control, Cache' Query Control และ Cache' Form Wizard
  - Cache' Object Link Control เป็น ActiveX Control ที่จำเป็นสำหรับการเขียนโปรแกรมด้วย Visual Basic ทำให้คุณสมบัติต่างๆของ Cache' Object นั้น ใน Visual Basic เข้าใจ
  - Cache' List Control เป็น ActiveX control เขียนขึ้นสำหรับ Visual Basic เอาไว้ช่วยในการแสดงผลของการ Query
  - Cache' Query Control เป็น ActiveX control ที่เขียนขึ้นมาสำหรับ Visual Basic ทำให้สามารถพัฒนาส่วนติดต่อกับผู้ใช้ในการ query และแสดงผลของผลลัพธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Cache' Form Wizard เป็น Visual Basic add-in ที่เพิ่มขึ้นมา อนุญาตให้ผู้พัฒนาสามารถพัฒนาโปรแกรมอย่างง่าย ขึ้นมาที่สามารถติดต่อและจัดการกับฐานข้อมูลที่ออกแบบได้

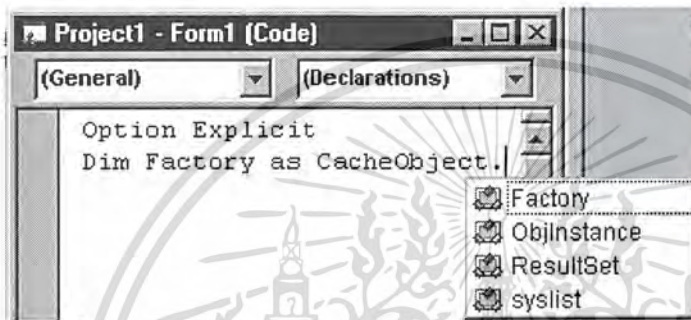
### 7.3 Cache' Object และ Visual Basic

#### 7.3.1 การปรับแต่ง Visual Basic เพื่อใช้กับ Cache' Object

การผนวกเอา ActiveX รวมเข้าไปกับ Visual Basic ทำให้การใช้งาน Cache' Object ใน Visual Basic นั้นสะดวกและง่ายขึ้น ยกตัวอย่างเช่น การประกาศตัวแปร สามารถกำหนดได้ดังนี้

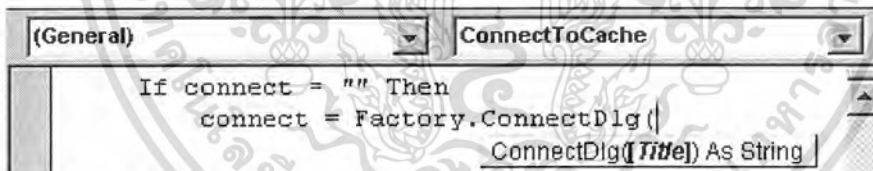
```
Dim List as CacheObject.SysList
```

คุณสมบัติจากการผนวกเอา ActiveX เข้าไป ทำให้มี pop up เมนู ไว้แสดงอีลิเมนต์ที่เป็นไปได้ที่ผู้ใช้จะใช้งาน และให้ผู้ใช้งานเลือกจากลิสต์ที่เป็นไปได้



รูปที่ 7-2 แสดงเมนู pop up ในการเขียนโค้ดใน Visual Basic จากการใช้ ActiveX Binding

นอกจากนี้แล้วยังสามารถแสดงไวยากรณ์ของการใช้คำสั่งต่างๆ ให้เห็นสำหรับการใช้ Cache' Object ด้วย ทำให้ผู้ใช้เขียนคำสั่งได้ถูกต้อง



รูปที่ 7-3 เมนู pop up ช่วยแสดงไวยากรณ์ของการใช้ Cache' Object

#### 7.3.1.1 การเลือก Project Reference ใน Visual Basic

ในการใช้งาน Visual Basic เพื่อให้สามารถใช้งานกับ Cache' object ได้นั้น ก่อนอื่น ต้องเลือก Project Reference เสียก่อนโดยการ cacheobject.dll เอาไว้ในโปรเจกต์ที่จำใช้เสียก่อนซึ่งสามารถทำได้ดังนี้

1. เปิดหน้าต่าง Reference จากเมนู Project|Reference
2. คลิกที่ ปุ่ม FIND
3. แล้วหา cacheobject.dll ใน <Cache' root>\bin ไดรคทอรี เจอแล้วทำการดับเบิลคลิก
4. ทำการเลือก project reference ต่อด้วยการทำ
5. เปิดหน้าต่าง Reference จากเมนู Project|Reference
6. เลือก check box ของ CacheObject ที่แสดงแล้วกดปุ่ม OK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 7.4 การจัดการกับ Cache' Objects ใน Visual Basic

### 7.4.1 การติดต่อกับ Server

สิ่งแรกที่ต้องทำ คือในส่วนของ Client โปรแกรมต้องทำการสร้าง Cache' Factory Object ขึ้นมาเสียก่อน และทำการติดต่อไปที่เครื่อง Server ที่กำลังรันนิ่ง Cache' Objects อยู่ ใน Visual Basic โค้ดแสดงการติดต่อดังนี้

```
` Visual Basic Code
option Explicit
Dim factory As CacheObject.Factory
```

```
Private Sub Form_Load()
```

```
` ทำการสร้าง instance ของ factory object
```

```
` ต้องใช้คำสั่ง "Set" ในการกำหนดค่าให้กับ object
```

```
Set factory = CreateObject("CacheObject.Factory")
```

```
` ทำการร้องขอติดต่อไปที่ Server ถ้ายังไม่ได้ทำการติดต่อ
```

```
If Not factory.IsConnected() Then
```

```
    Dim connectstring As String
```

```
` สามารถกำหนดค่า connection string เพื่อใช้สำหรับการติดต่อที่เจาะจงname space ได้
```

```
    connectstring = "cn_localhost:127.0.0.1[1972]:USER"
```

```
` หรืออีกทางเลือกให้เลือกรหัสผ่าน
```

```
` การใช้งานเมทอด จะส่งค่าของการติดต่อเก็บไว้ที่ connection string
```

```
    connectstring = factory.ConnectDlg()
```

```
    Dim success As Boolean
```

```
    success = factory.Connect(connectstring)
```

```
End If
```

ครั้งแรกต้องสร้าง instance ของ Cache' Factory object ขึ้นมาเสียก่อน ด้วยการ ใช้ฟังก์ชัน CreateObject

```
Set factory = CreateObject("CacheObject.Factory")
```

CreateObject จะล้มเหลวถ้าหาก ไม่สามารถหาหรือทำการโหลด cacheobject.dll ไม่ได้

เราสามารถทดสอบว่า Cache' Factory object ว่าได้ติดต่อไปแล้วหรือไม่ด้วยการ ใช้เมทอด factory.IsConnect

() จะส่งค่ากลับมาเป็นค่าความจริงหากได้ติดต่อแล้ว

การใช้งาน Cache' Factory object สามารถทำการติดต่อกับ Server ได้โดยการใช้ เมทอด Connect()

ของ Factory object

```
success = factory.Connect(connectstring)
```

Connect() จะถูกทำการส่งค่าของ connection string ซึ่งประกอบด้วยรูปแบบของ โปรโตคอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ใช้ติดต่อ , IP address ของเครื่อง Server และ หมายเลข port รวมทั้งชื่อของ name space ที่จะใช้ค้นด้วยเครื่องหมายโคลน Connect() จะส่งค่ากลับมาเป็นจริงหากติดต่อได้สำเร็จ และส่งค่าเท็จหากทำการติดต่อไปไม่ได้

#### 7.4.2 การสร้าง Object Instance ใหม่

ในการสร้าง instance ของ Cache' Object ใน Visual Basic เราสามารถทำได้โดยใช้ Cache' Factory object โดยใช้เมทอด New()

```
Dim Patient As Object
Set Patient = factory.New("Patient")
```

ซึ่งการใช้เมทอด New() นี้จะเกิดการกระทำต่างๆ ดังนี้

- สร้าง object instance ที่ฝั่ง Server แล้วส่งค่า OREF ของทางฝั่ง Server มาให้ Cache' Object Server เพื่อส่งให้ทางฝั่ง ActiveX
- สร้าง instance ของ ActiveX object ทางฝั่ง Client ซึ่งจะไปทาง object ทางฝั่ง Server
- ถ้าเมทอด New() ไม่สามารถสร้าง object ได้สำเร็จก็จะแสดง Error บอกให้ทราบ

#### 7.4.3 การบันทึกค่าของ Object

เราสามารถบันทึก Persistent Object เก็บลงไปที่ฐานข้อมูลได้โดยใช้ เมทอดชื่อ %Save() หรือถ้าใช้ใน Visual Basic ก็ sys\_Save

```
Dim status As String
patient.sys_Save      * บันทึก object ลงฐานข้อมูล
patient.sys_Close    * ปิด object ทางฝั่ง Server ลง
Set patient = Nothing * เช็ดค่า Object ใน Visual Basic ให้เป็น Nothing
```

สังเกตว่าเราต้องเรียกเมทอด sys\_Close เมื่อเราใช้ object นั้นเสร็จแล้ว ซึ่งการดำเนินงานนี้ทำให้ object ทางฝั่ง Server ถูกปิดลงด้วย และต้องเช็ดค่า object ใน Visual Basic ให้เป็น Nothing

#### 7.4.4 การเปิด Object ที่มีอยู่แล้วขึ้นมา

การโหลดวัตถุที่มีอยู่ในฐานข้อมูลขึ้นมาทำได้โดยการใช้เมทอด Open() จาก Cache Factory object

```
Dim Patient As Object
Set Patient = factory.Open("Patient", oid)
```

เมทอด Open() มี 2 อาร์กิวเมนต์ คือ ชื่อคลาส ที่จะเปิด และ OID ซึ่งมีค่าเป็นสตริงของวัตถุตัวที่ต้องการเปิดที่เก็บอยู่ในฐานข้อมูล เมทอด Open() จะทำสิ่งต่างๆ เหล่านี้ให้

- จะทำการโหลดวัตถุที่ต้องการมาไว้ที่ในหน่วยความจำทางฝั่ง Server และส่งค่าของ OREF ของวัตถุในหน่วยความจำไปให้กับวัตถุที่อยู่ทางฝั่ง Client
- สร้าง instance ของวัตถุ ActiceX ที่ทางฝั่ง Client ซึ่งจะติดต่อกับวัตถุที่อยู่ในหน่วยความจำทางฝั่ง Server ถ้าต้องการเปิดวัตถุโดยการระบุค่าของ OID ไปโดยตรงต้องใช้เมทอดของ Cache Factory object ที่ชื่อ OpenId() ดังตัวอย่าง

```
Dim Patient As Object
Set Patient = factory.OpenId("Patient", "22")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 7.4.5 การใช้วัตถุใน Visual Basic

เมื่อเปิดหรือสร้างวัตถุขึ้นมาโดยการเรียกเมธอดของ Cache' Factory Object โดยใช้เมธอดชื่อ New() หรือ Open() เราก็สามารถใช้วัตถุดังกล่าวเสมือนวัตถุใน Visual Basic ทั่วไป ยกตัวอย่างเช่นการกำหนดค่าให้กับพอร์ต์

```
Dim name As String
name = patient.Name
patient.Name = name
```

และสามารถเรียกใช้เมธอดของวัตถุดังกล่าวได้ ซึ่งเมธอดที่เรียกจะถูกปฏิบัติที่ทางฝั่ง Server เช่น

```
patient.Admit()
```

มีข้อแตกต่างกันระหว่างการเรียกใช้เมธอดใน Visual Basic และ ใน Cache' ObjectScript คือใน Cache ObjectScript นั้นเมธอดจะขึ้นต้นด้วย "%" ส่วนใน Visual Basic จะขึ้นต้นด้วย "sys\_" ยกตัวอย่างเช่น ใน ObjectScript เรียกใช้ pat.%Save() แต่ใน Visual Basic จะเรียกโดยใช้ pat.sys\_Save()

#### 7.4.6 การเรียก Query ใน Visual Basic

ใน Visual Basic นั้นการ Query ที่สร้างขึ้นจากคลาสในฐานข้อมูลสามารถเรียกใช้ได้โดยใช้วัตถุของคลาส ResultSet ซึ่ง วัตถุของคลาส ResultSet นั้นต้องกำหนดชื่อของคลาสที่เก็บ Query และชื่อของ Query ที่สร้างไว้ ในการจะเรียกใช้งาน Query ดังกล่าว ยกตัวอย่างการใช้งาน Query ที่อยู่ในคลาส Person ที่ชื่อ ByName

```
Dim rset As CacheObject.ResultSet
Dim columns As Integer
Dim counter As Integer
Set rset = factory.ResultSet("Person", "ByName")
' หาว่ามีกี่คอลัมน์
datacolumns = rset.GetColumnCount()
' ทำการรัน Query
' Query นี้จะส่งผลลัพธ์ของ Person ที่ขึ้นต้นด้วยอักษรมาให้
rset.Execute("A")
' วนลูปจนกว่าจะไม่มีผลลัพธ์อีกต่อไป

While rset.Next()
  For counter = 1 To columns
    Print rset.GetData(counter)
  Next counter
Wend
' ทำการปิดวัตถุ ResultSet ที่ได้สร้างขึ้นมา

rset.Close()
```

จากตัวอย่างการใช้งานข้างบน สามารถเรียกใช้งาน Query ชื่อ ByName ซึ่งถูกกำหนดขึ้นไว้ในคลาส Person ซึ่ง Query นี้จะส่งค่าของวัตถุ Person ทั้งหมดที่มีชื่อขึ้นต้นด้วยตัวอักษรที่กำหนดลงไป ("A") หลังจากทำแล้วก็สามารถเรียกดูค่าวัตถุทีละตัวได้ โดยใช้เมธอด rset.Next() และสามารถหาได้ว่าผลลัพธ์จากการทำ Query นั้นให้ค่ามากี่คอลัมน์ในแต่ละแถวของผลลัพธ์โดยใช้เมธอด rset.GetColumnCount() แล้วใช้ลูป FOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วนอ่านค่าของแต่ละคอลัมน์พร้อมกัน วนไปอ่านผลลัพธ์แถวถัดไปมาอ่าน เมื่อจัดการเสร็จแล้วก็ปิดวัตถุ ResultSet ด้วยเมทอด rset.Close()

#### 7.4.7 การตรวจสอบข้อผิดพลาดใน Visual Basic

Cache' Factory object จะแสดงข้อผิดพลาดให้ Visual Basicทราบหากเจอกับข้อผิดพลาดเกิดขึ้น ซึ่งผู้ใช้สามารถนำข้อมูลความผิดพลาดนั้นมาใช้ได้โดยการใช้วัตถุใน Visual Basic คือ err ดังตัวอย่าง

```
Private Sub OpenObject(oid As String)
```

```
    ' ทำการเชื่อมต่อและดูแลข้อผิดพลาด
```

```
    On Error GoTo error_trap:
```

```
    ' ทดสอบโดยการพยายามเปิดวัตถุที่ไม่มีอยู่
```

```
    Dim Patient As Object
```

```
    Set Patient = factory.Open("Patient", oid)
```

```
    '.....
    Exit Sub
```

```
error_trap:
```

```
    ' ส่วนที่เอาไว้แสดงข้อผิดพลาดจะแสดง 'โดยะลือกของ ข้อผิดพลาดให้เห็น
```

```
    MsgBox (Err.Description)
```

```
End Sub
```

อีกวิธีหนึ่งในการตรวจสอบข้อผิดพลาดคือการใช้เมทอด GetErrorText() ของคลาสFactory ซึ่งเมทอดนี้จะรับค่าอธิบายการเกิดข้อผิดพลาดที่สร้างจากวัตถุ err มาแล้วส่งค่าเป็นข้อความแสดงการเกิดข้อผิดพลาดไปให้ ยกตัวอย่างเช่น

- Cache ไม่สามารถบันทึกวัตถุในคลาสได้

- ส่งค่าของข้อผิดพลาดไปยังวัตถุ err

```
msg = factory.GetErrorText(err.description)
```

- ตอนนีตัวแปร msg เก็บค่าสตรงของข้อผิดพลาดไว้

- คือค่า "Unable to save class: MyClass"

ใน ActiveX จะมีรูปแบบของการเกิดข้อผิดพลาดของ Cache' 3 แบบคือ

| Error Code | Description   |
|------------|---|
| 9990       | เกิดขึ้นเมื่อขณะพยายามสร้างวัตถุขึ้นมา ถ้าเกิดข้อผิดพลาดชนิดนี้ให้ตรวจสอบว่ามีคลาสดังกล่าวอยู่หรือไม่ ใน namespace และ Server ที่ติดต่อชื่อถูกต้องหรือไม่ |
| 9991       | เป็นข้อผิดพลาดที่เกิดจากภายในระบบ Cache' System   |
| 9992       | เป็นข้อผิดพลาดที่เกิดกับเมทอดที่มีการส่งค่ากลับมาเป็นชนิด %Status ซึ่งเมทอดดังกล่าวส่งค่า False กลับมาให้   |

ตารางที่ 7-2 แสดงรหัสข้อผิดพลาดที่ของ Cache' ใน ActiveX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 8

## บทสรุป

### 8.1 ข้อเสนอแนะ ปัญหาที่พบ และ แนวทางการศึกษาต่อ

จากการศึกษาและทำวิจัยในโครงการงาน Object-Oriented Database in Temporal Application ในส่วนของฐานข้อมูลเชิงวัตถุ และแนวคิดของฐานข้อมูลเชิงเวลานั้นทำให้ทราบถึงปัญหาและแนวทางการแก้ไข ซึ่งสามารถรวบรวมเป็นข้อเสนอแนะได้ดังนี้

1. ฐานข้อมูลเชิงวัตถุได้พัฒนาจากแนวคิดเชิงวัตถุ คุณสมบัติต่างๆ เชิงวัตถุ เช่น การสืบทอด โพลิมอร์ฟิซึม , การซ่อนรายละเอียด , เมธอด ซึ่งสิ่งต่างๆเหล่านี้ทำให้ฐานข้อมูลเชิงวัตถุ มีข้อได้เปรียบกว่าฐานเชิงสัมพันธ์ หากในการพัฒนาฐานข้อมูลดังกล่าว ใช้ฐานข้อมูลเชิงวัตถุมาช่วย โดยใช้คุณสมบัติต่างๆเชิงวัตถุ อย่างเต็มที่แล้วนั้น ย่อมทำให้ได้ประสิทธิภาพในการทำงานที่ดี

2. ในการออกแบบฐานข้อมูลเชิงวัตถุนั้นจำเป็นต้องใช้การวิเคราะห์เชิงวัตถุ หากผู้พัฒนานั้นมีความเข้าใจเกี่ยวกับแนวคิดเชิงวัตถุไม่ดี อาจจะทำให้การออกแบบที่ได้มาไม่ถูกต้อง ซึ่งอาจก่อให้เกิดปัญหาต่างๆตามมามากมายได้ หรืออาจทำให้งานล้มเหลวได้ หากการออกแบบนั้นขาดการวิเคราะห์เชิงวัตถุที่ดี ซึ่งในปัจจุบันมีเครื่องมือที่ช่วยในการวิเคราะห์ และออกแบบระบบ ด้วยวิธีการเชิงวัตถุช่วย เช่น UML (Unified Modeling Language) ทำให้การวิเคราะห์และออกแบบ มีความผิดพลาดน้อยลง

3. การนำแนวคิดของฐานข้อมูลเชิงเวลามาใช้ ทำให้การออกแบบและพัฒนา มีความยุ่งยากมากขึ้นกว่าแบบธรรมดา เพราะต้องคำนึงถึงสถานะของข้อมูลในเวลา ต่างๆ เพื่อไม่ให้เกิดความซ้ำซ้อนทางเวลาเกิดขึ้น การออกแบบและพัฒนาฐานข้อมูลเชิงวัตถุจึงต้องให้ความสำคัญกับสถานะของข้อมูล และนอกจากนี้ระบบควรมีลักษณะที่อำนวยความสะดวกต่อผู้ใช้ให้มากที่สุด เช่น การเขียนเมธอดต่างๆ เกี่ยวกับทางด้านเวลาไว้ให้ผู้พัฒนาโปรแกรม และผู้ใช้ไว้ใช้งานต่อไป หรือมีลักษณะเป็นคอมมอด เพื่อให้สามารถเอาไปใช้ในการพัฒนาในงานอื่นๆต่อไปได้ โดยการใช้คุณสมบัติเชิงวัตถุ เช่นการมีเมธอดมาตรฐาน สามารถนำไปใช้งานอื่นได้อีก

4. ในการทำโครงการที่ผ่านมา นั้น มีปัญหาเกิดขึ้น เช่นปัญหาจากทางด้านฐานข้อมูลเกี่ยวกับ เรื่องของเวลาที่ ระบบจัดการฐานข้อมูลที่ใช้ นั้น เก็บค่าของเวลาอีกรูปแบบหนึ่ง ซึ่งในแอปพลิเคชันเก็บในอีกรูปแบบหนึ่ง ทำให้การส่งค่าไปเก็บจากแอปพลิเคชันไปที่ฐานข้อมูล บางครั้งค่าของเวลาที่มีค่ามากๆ เกิดการเปลี่ยนแปลงค่าได้ เพราะฉะนั้นหากผู้พัฒนาไม่ศึกษาให้ดีจะทำให้เกิดข้อผิดพลาดได้

5. ระบบจัดการฐานข้อมูลที่ใช้ นั้น ก็มีความสำคัญ ในโครงการนี้ใช้ฐานข้อมูลเชิงวัตถุ ซึ่งยังไม่แพร่หลายนัก เอกสารการใช้งานต่างๆ ยังมีไม่มากพอ ซึ่งทำให้ต้องค้นคว้าหาข้อมูลต่างๆ ด้วยตนเองเป็นสำคัญ ซึ่งหากศึกษาไม่เข้าใจดีแล้วนั้น ในการพัฒนาภายหลังอาจจะทำให้ไม่ได้ใช้คุณสมบัติต่างๆ ที่มีอยู่อย่างเต็มที่

6. ระบบจัดการฐานข้อมูลที่ใช้ ไม่มีภาษาที่ใช้จัดการกับข้อมูลที่มีความสลับซับซ้อน ทำให้การพัฒนาจำเป็นต้องเขียนเมธอด จัดการกับข้อมูลที่มีความสลับซับซ้อนซึ่งไม่ค่อยสะดวกนัก ทำให้ไม่สามารถตอบคำถามแบบทันทีทันใดได้ หากเกี่ยวข้องกับข้อมูลเหล่านั้น

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวทางในการนำเอาระบบฐานข้อมูลเชิงเวลาไปพัฒนาต่อในระบบทะเบียนนักศึกษา ระบบทะเบียนนักศึกษาในสถาบันนั้น เป็นระบบที่มีความสลับซับซ้อนและมีขนาดใหญ่ มีปัญหาและความต้องการต่างๆ อีกที่ทางผู้จัดทำโครงการเล็งเห็นและขอเสนอแนวทางในการนำไปพัฒนาต่อไปดังนี้

- ระบบทะเบียนนักศึกษา เกี่ยวข้องกับงานย่อยๆ อีกหลายงาน ซึ่งทางผู้จัดทำโครงการได้ยกกรณีศึกษามาบางระบบเท่านั้น
- การเรียนยังมีความซับซ้อนในความเป็นจริง เช่น นักศึกษาสามารถเรียนและทำการโอนย้ายไปเรียน ต่างหลักสูตรได้ ซึ่งทำให้หลักสูตรที่เคยเรียนเปลี่ยน ซึ่งอาจเป็นไปได้ที่ต้องย้ายคณะ , ภาควิชา และสาขาวิชาที่เรียนด้วย ซึ่งในโครงการนี้นำเสนอเพียงแค่การเรียนปกติที่นักศึกษาผู้นั้นเรียนเพียงหลักสูตรเดียว
- วิชาบางครั้งสามารถ มีรหัสวิชา และชื่อวิชาซ้ำกันได้ เนื่องจากการออกหลักสูตรนั้น ทำต่างคณะกัน ทำให้ระยะเวลาเดียวกันอาจจะเป็นไปได้ที่มีวิชาที่มีรหัสวิชา และชื่อเหมือนกัน เกิดขึ้นซึ่งแท้ที่จริงแล้วเป็นคนละวิชา ซึ่งแนวทางในการแก้ไขคือการใช้ Object identifier
- หลักสูตรบางหลักสูตรบางครั้ง บางวิชาไม่ได้มีอยู่ในหลักสูตรในตอนแรก แต่มาเปิดสอนทีหลังทำให้ในหลักสูตรตอนแรกที่ใส่ลงในฐานข้อมูลไม่มี ข้อมูลวิชาดังกล่าวอยู่เช่น วิชาเลือก
- โปรแกรมทางฝั่งไคลเอนท์นั้น ใช้ Visual Basic ในการพัฒนา การติดต่อระหว่าง Client และ Server ทำให้ไม่สามารถให้บริการจากผู้ใช้ที่อยู่ในระยะไกลได้ หากนำเทคโนโลยีทางด้านอินเทอร์เน็ตมาใช้ ก็จะทำให้สามารถเพิ่มประสิทธิภาพในการให้บริการมากขึ้น เช่น ระบบงานลงทะเบียน , สอบถามผลการเรียน เป็นต้น

อนึ่งสำหรับระบบที่มีขนาดใหญ่และมีความซับซ้อนสูงนั้น การใช้แนวคิดเชิงวัตถุมาวิเคราะห์จะทำให้ความซับซ้อนต่างๆ น้อยลง เนื่องจากคุณสมบัติเชิงวัตถุที่อาศัยการแยกแยะเอกลักษณ์ต่างๆ แล้วสร้างความสัมพันธ์ระหว่างวัตถุออกมา แต่สิ่งที่ยากคงเป็นเรื่องของการวิเคราะห์ระบบออกมาให้มีความถูกต้อง สมบูรณ์ครบถ้วนนั้น จำเป็นต้องอาศัยความรู้ความชำนาญในเรื่องของแนวคิดเชิงวัตถุที่ดี และเรื่องของระบบงานที่จะทำการวิเคราะห์เป็นสำคัญ

## 8.2 สรุป

จากการทดลองและทำการวิจัย โครงการในหัวข้อฐานข้อมูลเชิงเวลา โดยใช้ฐานข้อมูลเชิงวัตถุมาช่วยสามารถสรุปเป็น 2 ส่วนได้ดังนี้

- แนวคิดและทฤษฎีในการออกแบบ

แนวคิดในการออกแบบฐานข้อมูลเชิงวัตถุกับการประยุกต์เชิงเวลานั้น จากข้อเสียของการใช้ฐานข้อมูลเชิงสัมพันธ์ ในฐานข้อมูลเชิงเวลา ยังพบว่ายังมีปัญหาดังกล่าว อยู่ เช่น เรื่องของความซ้ำซ้อนของข้อมูลอื่นที่ไม่ได้เปลี่ยนแปลงตามเวลาด้วย ปัญหาอีกหลักที่ไม่สามารถใช้งานได้ แต่ในฐานข้อมูลเชิงวัตถุ การพัฒนาจะอาศัยคุณสมบัติของวัตถุ เช่น การมีค่าตัวโทที่สลับซับซ้อนได้ เช่น ลิสต์ อาร์เรย์ นอกจากนี้เรายังสามารถกำหนดค่าตัวโทเองได้อีกด้วย และจากแนวคิดเชิงวัตถุ วัตถุจะมีตัวบ่งชี้วัตถุเป็นค่าบอกความแตกต่างของแต่ละวัตถุ ซึ่งสามารถแยกแยะวัตถุต่างๆ ได้อย่างสะดวก ไม่ต้องคำนึงถึงเรื่องของ คีย์หลัก เหมือนในฐานข้อมูลเชิงสัมพันธ์ ในการเก็บค่าของการเปลี่ยนแปลงของข้อมูลก็จะอาศัยคุณสมบัติของค่าตัวโทที่ได้กล่าวมาแล้ว ในการเก็บประวัติการเปลี่ยนแปลง ซึ่งสามารถเก็บเฉพาะเหตุการณ์ที่มีการเปลี่ยนแปลงได้ โดยไม่กระทบกับเอกสารเป็นเอกสารที่ส่งวนเวียนไปเพื่อการศึกษาค้นคว้า เมื่ออนุญาตให้เข้าใช้ระบบงานด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอททริบิวต์อื่น ทำให้ข้อมูลไม่เกิดความซับซ้อนขึ้น นอกจากนี้เรายังสามารถเขียนเมธอดเพื่อเอาไว้จัดการกับข้อมูลเชิงเวลาดังกล่าวได้ ทั้งการเพิ่ม แก้ไข และลบ ข้อมูลเหล่านั้น ช่วยอำนวยความสะดวกในการพัฒนามากขึ้น โดยพยายามเขียนเมธอดต่างๆ ให้มีลักษณะเป็นกลางๆ ให้สามารถเรียกใช้จากวัตถุใดๆ ได้เพื่อจัดการกับข้อมูลเชิงเวลาเหล่านั้น และนอกจากนี้ ยังสามารถนำไปใช้ในการพัฒนางานอื่นได้อีกด้วย โดยอาศัยคุณสมบัติที่มีอยู่ในฐานข้อมูลเชิงวัตถุ

- ในส่วนของการพัฒนาและการนำไปใช้งาน

ปัจจุบันแนวคิดเชิงวัตถุ ในการเขียนโปรแกรมนั้น มีความนิยมเป็นอย่างมาก ดังจะเห็นได้

จากภาษาโปรแกรมมิ่งในปัจจุบันล้วนแล้วแต่เป็น OOP (Object-Oriented Programming) แทบทั้งสิ้น เช่น JAVA , Visual Basic , Delphi , C++ เป็นต้น ซึ่งระบบจัดการฐานข้อมูลได้ผนวกเอาคุณสมบัติดัง เช่น ActiveX ทำให้การเขียนโปรแกรมติดต่อกันระหว่างฐานข้อมูล และภาษาโปรแกรมมิ่งที่เป็นภาษาเชิงวัตถุทำได้สะดวก ซึ่งในโครงการนี้ใช้ Visual Basic เป็นตัวอย่างในการพัฒนา แต่ระบบจัดการฐานข้อมูลที่ใช้นั้นไม่มีภาษาจัดการกับฐานข้อมูลเชิงเวลา ที่มีคุณสมบัติเชิงวัตถุอย่างเต็มตัว เพียงแต่เป็นภาษาที่มีพื้นฐานมาจากภาษา SQL-92 แล้วเพิ่มเติมบางส่วนลงไปเพื่อให้สามารถจัดการกับฐานข้อมูลเชิงวัตถุได้บางลักษณะได้เท่านั้น เช่น การ reference แต่ไม่สามารถสนับสนุนคุณสมบัติทั้งหมดได้ เช่น การสืบค้นข้อมูลภายในดาต้าไทป์ที่มีความสลับซับซ้อนได้ ซึ่งทำให้การใช้งานไม่สะดวกเท่าที่ควร การแก้ไขคือการเขียนเมธอดจัดเตรียมไว้ล่วงหน้าไว้ใช้งาน

ระบบทะเบียนนักศึกษาที่ได้นำมาเป็นกรณีศึกษานั้น ได้ใช้แนวคิดของฐานข้อมูลเวลาไปประยุกต์ใช้ในการเก็บประวัติทั้งนักศึกษา หลักสูตร วิชาต่างๆ และพยายามใช้ข้อดีของฐานข้อมูลเชิงวัตถุมาช่วยในการพัฒนา

ในการนำฐานข้อมูลเชิงวัตถุไปใช้นั้น ก่อให้เกิดประโยชน์ เพราะทำให้สามารถสืบค้นประวัติต่างๆ ของข้อมูลต่างๆ ได้ ติดตามการเปลี่ยนแปลงข้อมูลต่างๆ ได้ ยิ่งถ้าเป็นข้อมูลที่มีความสำคัญด้วยแล้ว การเก็บประวัติของข้อมูลไว้ หากข้อมูลมีความผิดพลาด ก็สามารถทำการกู้คืนกลับมาได้ นอกจากนั้นแล้วในการนำข้อมูลต่างๆ ที่เป็นข้อมูลเชิงเวลาไปใช้งาน ยังเอื้อประโยชน์ต่างๆ เช่น ใช้ในการตัดสินใจต่าง การมีข้อมูลในอดีตย่อมทำให้การตัดสินใจนั้น มีความแม่นยำมากขึ้น ซึ่งในโลกของความจริงข้อมูลต่างๆ ส่วนมากล้วนแล้วมีความเกี่ยวข้องกับเรื่องของเวลา ดังนั้นการนำแนวคิดของฐานข้อมูลเชิงเวลาไปใช้นั้น ย่อมก่อให้เกิดประโยชน์ซึ่งฐานข้อมูลที่จะนำมาสนับสนุนการพัฒนาฐานข้อมูลเชิงเวลานั้น ฐานข้อมูลเชิงวัตถุถือได้ว่าเป็นความเหมาะสม

## บรรณานุกรม

- [1] Abdullah Uz Tansel, James Clifford, Shashi Gadia, Sushil Jajodia, Arie Segev, Richard Snodgrass (1993) : “*Temporal Databases : Theory, Design, and Implementation*”, “The Benjamin/Cummings Publishing Company, Inc.”, 1993.
- [2] Richard T. Snodgrass (1998) : “*Managing Temporal Data : A Five-Part Series*”, “A TimeCenter Technical Report”, 1998.
- [3] S. Jensen and R. T. Snodgrass (1997) : “*Temporal Data Management*” , “A TimeCenter Technical Report”, 1997.
- [4] Steiner and M. C. Norrie (1997) : “*Implementing Temporal Databases in Object-Oriented Systems*”, “A TimeCenter Technical Report”, 1997.
- [5] C.J. Date (2000) : “*An Introduction to DATABASE SYSTEMS*” ,”Addison Wesley Longman, Inc.”,2000.
- [6] Richard T. Snodgrass (1996) : “*A Road Map of Additions to SQL/Temporal*”, 1996.
- [7] Richard T. Snodgrass, Michael H. Bohlen, Christian S. Jensen, Andreas Steiner (1996) : “*Adding Valid Time to SQL/Temporal*”, “ANSI Expert's Contribution”, 1996.
- [8] Richard T. Snodgrass, Michael H. Bohlen, Christian S. Jensen, Andreas Steiner (1996) : “*Adding Transaction Time to SQL/Temporal*”, “ANSI Expert's Contribution”, 1996.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก.

### UML (Unified Modeling Language)

#### ก.1 Unified Modeling Language

Unified Modeling Language เป็นการสร้างโมเดลจำลอง (Abstract Model) ที่มีสัญลักษณ์ (notation) ที่มีประสิทธิภาพซึ่งจะช่วยในส่วนของการวิเคราะห์ (Analysis) และออกแบบ (Design) ซึ่งมีประโยชน์ในหลายด้าน สำหรับใช้ในการออกแบบระบบต่างๆ โดยจะเป็นรูปแบบของออบเจกต์ โอเรียนเท็ด ซึ่งการออกแบบจะทำให้เราสามารถวิเคราะห์ส่วนประกอบโดยรวมของระบบและยังสามารถที่จะรองรับภาษาที่จะใช้ในการ พัฒนา ได้หลากหลายภาษามากและยังหลีกเลี่ยงความซ้ำซ้อนและเฉพะเจาะจงเกินไป ซึ่งการออกแบบโดยวิธี UML นี้สามารถที่จะแสดงออกมาได้ในรูปแบบของไดอะแกรม (diagram) ต่างๆ

การพัฒนาซอฟต์แวร์ในปัจจุบันนั้นเกิดปัญหาขึ้นมากมายเช่น ระบบที่พัฒนามาไม่ตรงตามความต้องการของลูกค้าหรืออาจใช้เวลามากเกินไป ซึ่งก็ได้มีการพัฒนาแก้ไขปัญหาคือใช้เทคนิคใหม่ ๆ เข้ามาช่วยแต่ก็ยังคงเกิดปัญหาขึ้นอยู่ดี ทางแก้ของปัญหาต่าง ๆ นี้คือต้องมีการสร้างโมเดลจำลอง ของระบบขึ้นมาก่อน ซึ่งจะช่วยให้มีข้อดีต่างๆ ดังนี้

1. ความถูกต้องตามความต้องการจริงของระบบ
2. ทุก ๆ ส่วนของระบบมีความสอดคล้องกัน
3. ง่ายในการสื่อสารระหว่างผู้พัฒนาและผู้ใช้ระบบ
4. สามารถเปลี่ยนแปลงได้ง่าย
5. สามารถทำความเข้าใจได้ง่ายทั้งผู้พัฒนาและผู้ใช้ระบบ

#### ก.2 ส่วนประกอบของ UML

UML ประกอบด้วยหลาย ๆ ส่วน ดังนี้

**View:** แสดงมุมมองต่าง ๆ ของระบบที่ถูกออกแบบขึ้นมาโดยจะใช้ไดอะแกรมต่าง ๆ ในการอธิบาย

**Diagram:** เป็นไดอะแกรมที่ใช้ในการอธิบายส่วนต่าง ๆ ของ view ซึ่งใน UML มีไดอะแกรมทั้งหมด 9 ไดอะแกรม

**Model element:** เป็นสัญลักษณ์ที่ใช้ในไดอะแกรมเพื่อแสดงหรือเป็นตัวแทนของสิ่งต่างๆ เช่น คลาส (class), ออบเจกต์, แมสเสจ (message) และความสัมพันธ์ (relationship) เป็นต้น

**General mechanism:** เป็นส่วนที่แสดงคอมเม้นเพิ่มเติม (extra comment), ข้อมูลอื่น ๆ ที่จำเป็นหรือความหมายของโมเดลเอลิเมนต์ (model element)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ก.2.1 view

ในการออกแบบระบบที่มีขนาดใหญ่และมีความซับซ้อนมาก ๆ นั้นจะทำให้ผู้ออกแบบระบบไม่สามารถที่จะออกแบบระบบได้ครบถ้วน ดังนั้นจึงต้องมีมุมมองระบบเป็นมุมมองต่างๆ เพื่อให้ง่ายในการออกแบบ เช่น มุมมองด้านฟังก์ชันนอล , นอนฟังก์ชันนอล (Nonfunctional), มุมมองขององค์กร เป็นต้น ดังนั้นระบบจึงมี view ที่ต่างๆ กันซึ่งแต่ละ view จะแสดงมุมมองเฉพาะของระบบซึ่งอธิบายรวมกันเป็นระบบที่สมบูรณ์ ซึ่งจะประกอบด้วย View ต่าง ๆ ดังนี้

### ก.2.1.1 Use-case View

อธิบายการทำงานต่าง ๆ ของระบบที่ถูกมองจากภายนอกหรือผู้ใช้งาน Use-case view ซึ่งอธิบายโดย ยูสเคสไดอะแกรม (use-case diagram) เป็นมุมมองสำหรับลูกค้า , ผู้ออกแบบ , ผู้พัฒนาระบบ และผู้ทดสอบระบบ

### ก.2.1.2 Logical View

อธิบายการทำงานต่าง ๆ ที่ถูกออกแบบไว้ภายในระบบ ว่าระบบจะมีบริการอะไรให้กับผู้ใช้งาน โดยจะแสดงโครงสร้างแบบสถิต (Static) เช่น คลาส, ออบเจกต์, ความสัมพันธ์และการทำงานร่วมกันแบบไดนามิก (dynamic collaboration) ซึ่งจะเกิดขึ้นเมื่อออบเจกต์ ส่งเมสเสจ ระหว่างกันในการทำงาน โครงสร้างแบบสถิตจะอธิบายโดยใช้ คลาสไดอะแกรม (Class diagram) และออบเจกต์ไดอะแกรม (object diagram) ส่วนการทำงานร่วมกันแบบไดนามิกจะอธิบายโดยใช้ สเตตไดอะแกรม (state diagram), ซีควีนซ์ไดอะแกรม (sequence diagram), คีอแลโบลีชั่น ไดอะแกรม (collaboration diagram) และ แอคติวิตีไดอะแกรม (activity diagram)

### ก.2.1.3 Component View

อธิบายการสร้างและความขึ้นต่อกันของโมดูล (Module) ที่ใช้ในการพัฒนาระบบ โดยใช้คอมโพเนนต์ไดอะแกรม (Component diagram) ในการอธิบาย

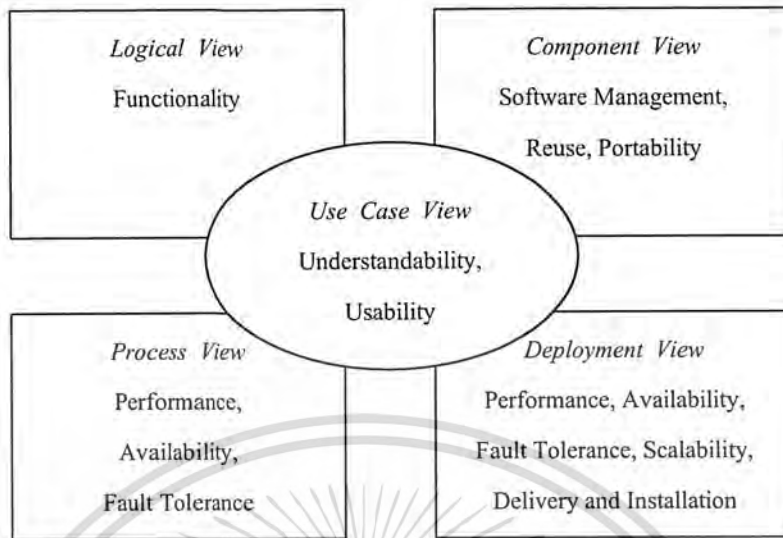
### ก.2.1.4 Deployment View

อธิบายการจัดวางระบบให้เหมาะในด้านกายภาพ (Physical) แสดงด้วยคอมพิวเตอร์และโหนด (nodes) ต่าง ๆ เพื่อให้ระบบมีเสถียรภาพมากขึ้น โดยใช้ ดีพลอยเม้นต์ไดอะแกรม (deployment diagram) ในการอธิบาย

### ก.2.1.5 Process View

แสดงการทำงานร่วมกันและการติดต่อกันของส่วนต่าง ๆ ในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



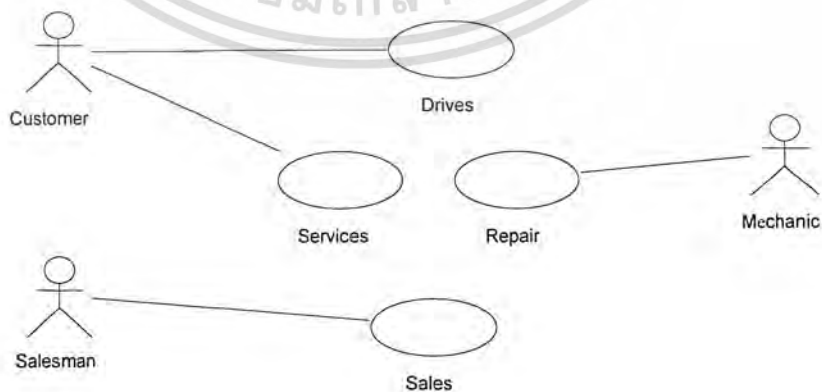
รูปที่ ก-1 สถาปัตยกรรมของ View

## ก.2.2 Diagram

เป็นไดอะแกรมซึ่งแสดงสัญลักษณ์ที่ถูกจัดเรียงเพื่อ อธิบายระบบในมุมมองต่าง ๆ ซึ่งในระบบหนึ่ง ๆ จะประกอบด้วยหลาย ๆ ไดอะแกรม และในแต่ละไดอะแกรมยังสามารถมองในหลาย ๆ มุมมองได้ UML มีไดอะแกรมที่ต่างกันอยู่ 9 ไดอะแกรม ดังนี้

### ก.2.2.1 Use Case Diagram

ยูสเคสไดอะแกรมจะแสดงความสัมพันธ์ระหว่างผู้ใช้งาน (User) กับระบบ โดยใช้แอ็กเตอร์ (actors) แทนผู้ใช้งานและแอ็กเตอร์จะต้องติดต่อกับระบบโดยผ่าน ยูสเคส (use case) ต่าง ๆ ซึ่งยูสเคสใน use-case diagram ก็คือการทำงานต่าง ๆ ของระบบที่ผู้ใช้ต้องการ



รูปที่ ก-2 uses case diagram ของบริษัทเช่ารถยนต์

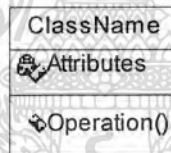
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอดมินไม่ใช่ส่วนประกอบของระบบ แต่จะเป็นส่วนที่ใช้ติดต่อกับระบบ ซึ่งอาจจะเป็นเพียง การป้อนข้อมูลเข้าไปในระบบหรือเป็นเพียงการรับข้อมูลจากระบบ หรืออาจจะเป็นทั้งสองอย่างก็ได้ ใน ระบบหนึ่งๆ จะต้องมี แอดมินซึ่งจะต้องทำการกำหนดคุณลักษณะของแอดมิน (identify actor) ให้ได้ ก่อน ดังนี้

1. ใครเป็นผู้ใช้งานระบบ ?
2. ใครที่มีความเหมาะสมที่จะใช้งานระบบ ?
3. ใครที่มีส่วนสนับสนุนข้อมูลในระบบ ใช้ข้อมูลในระบบและแก้ไขข้อมูลในระบบ ?
4. ใครที่สนับสนุนการบำรุงรักษาระบบ(Maintenance)?
5. ระบบมีการใช้งานกับภายนอกหรือเปล่า ?

### ก.2.2.2 Class Diagram

คลาสไดอะแกรมจะแสดง โครงสร้างที่ไม่เปลี่ยนแปลงของระบบในเทอมของคลาส และแสดง ความสัมพันธ์ระหว่างคลาสต่าง ๆ เหล่านั้น ซึ่งในระบบหนึ่ง ๆ จะประกอบด้วยหลาย ๆ คลาสไดอะแกรม คลาส คือ กลุ่มของออบเจกต์ที่มีคุณสมบัติ(attributes) และพฤติกรรม(behavior) ร่วมกัน สัญลักษณ์ของคลาส แสดงดังรูปที่ ก-3 ส่วนแรกเก็บ ชื่อคลาส ส่วนที่สองเก็บคุณสมบัติ ส่วนที่ สามเก็บ การทำงาน(operations) โดยที่คลาส 1 คลาส จะต้องมีอย่างน้อย 1 ส่วน คือ ชื่อคลาส



รูปที่ ก-3 สัญลักษณ์ของ Class diagram

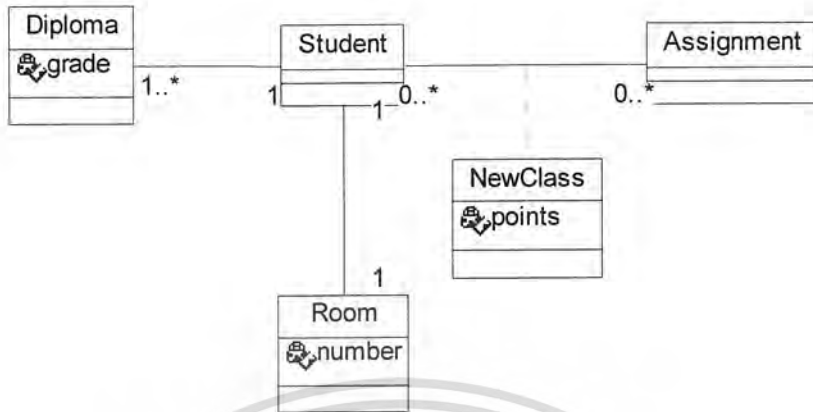
สัญลักษณ์ความสัมพันธ์ต่าง ๆ ของคลาสไดอะแกรม มีดังนี้

- 1 หมายถึงจะมีออบเจกต์ในคลาสไดอะแกรมได้หนึ่งออบเจกต์เท่านั้น
- 0..1 หมายถึงจะมีออบเจกต์ในคลาสไดอะแกรมได้แค่หนึ่งหรืออาจจะไม่มีก็ได้
- M..N หมายถึงจะมีออบเจกต์ในคลาสไดอะแกรมได้ตั้งแต่ M ถึง N ( เมื่อ M และ N เป็น จำนวนเต็มบวก)
- \* หมายถึงจะมีออบเจกต์ในคลาสไดอะแกรมได้ตั้งแต่ศูนย์ขึ้นไป
- 0.\* หมายถึงจะมีออบเจกต์ใน คลาสไดอะแกรมได้ตั้งแต่ศูนย์ขึ้นไป
- 1..\* หมายถึงจะมีออบเจกต์ใน คลาสไดอะแกรมได้ตั้งแต่หนึ่งขึ้นไป

ตัวอย่าง คลาสไดอะแกรมแสดงความสัมพันธ์ของนักเรียน(student)ที่จะต้องทำรายงาน

(Perform assignment) โดยผลที่ได้จากการทำรายงานก็คือคะแนน (points) เก็บเอาไว้ ซึ่งความสัมพันธ์ของ เอกสารนี้เป็นเอกสารที่ส่งวนเวียนสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติไหนไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คะแนนเรียกว่าความสัมพันธ์ระหว่างแอตทริบิวต์ (attribute association) โดยที่นักเรียนแต่ละคนจะต้องมีห้องเรียนคนละหนึ่งห้องเท่านั้น และอาจจะมีหรือยังไม่มีใบเกรด (Diploma) ก็ได้



รูปที่ ก-4 ตัวอย่างของ Class diagram

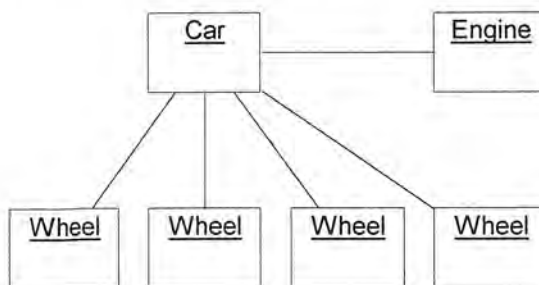
### ก.2.2.3 Object Diagram

ออบเจกต์ไดอะแกรมจะได้อมาจากคลาสไดอะแกรม ซึ่งจะเป็นการแสดงความเชื่อมต่อระหว่างอินชแทนซ์ (instances) ต่าง ๆ ที่อยู่ภายในคลาสไดอะแกรมอย่างละเอียด เพื่อใช้เป็นตัวอย่างสำหรับคลาสไดอะแกรมที่มีความซับซ้อน โดยจะนำมาแทนเป็น ออบเจกต์ จริงๆ และแสดงความสัมพันธ์ต่างๆ ให้เห็น โดยสัญลักษณ์ของอินชแทนซ์ในออบเจกต์ไดอะแกรม จะใช้การขีดเส้นใต้เป็นสัญลักษณ์ ภายในกรอบสี่เหลี่ยมดังรูปที่ ก-5



รูปที่ ก-5 สัญลักษณ์ของ Instance ใน Object diagram

ตัวอย่าง ออบเจกต์ไดอะแกรมอธิบายโครงสร้างของรถยนต์ (Car) ซึ่งรถแต่ละคันจะมีเครื่องยนต์ (engine) หนึ่งเครื่องและมีล้อ 4 ล้อ

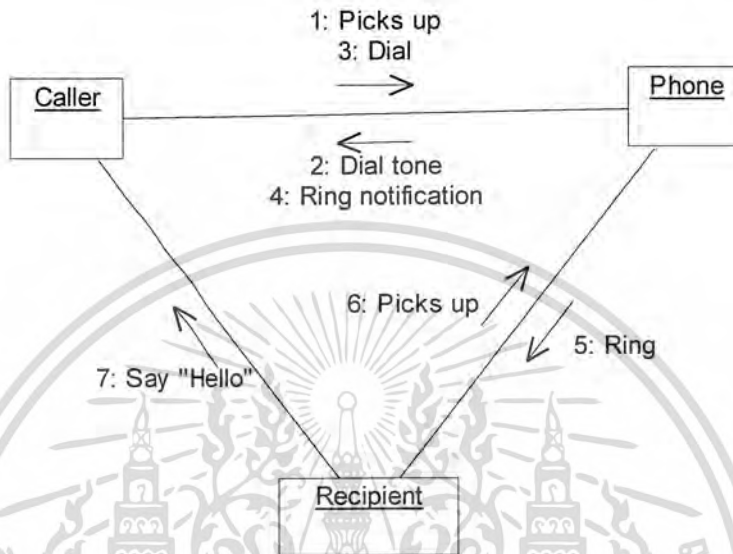


รูปที่ ก-6 Object diagram ของรถยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ก.2.2.4 Collaboration Diagram

คือลเลโบเรชั่นไดอะแกรมใช้อธิบายการทำงานร่วมกันของออบเจ็กต์ที่มีส่งแมสเสจ ระหว่างกัน เพื่อบอกถึงลำดับขั้นตอนการทำงานของระบบ โดยใช้สัญลักษณ์ลูกศรเป็นตัวกำหนดทิศทางซึ่ง คือลเลโบเรชั่นไดอะแกรมจะเป็นส่วนอธิบายการทำงานของออบเจ็กต์ไดอะแกรม



รูปที่ ก-7 Collaboration diagram สำหรับการสนทนาทางโทรศัพท์

จากรูปที่ ก-7 จะมีการติดต่อกันระหว่าง 3 ออบเจ็กต์ คือ ผู้เรียก (Caller) , ผู้รับ (Recipient) และ โทรศัพท์ (Phone) โดยมีลำดับก่อนหลังในการส่งแมสเสจ ให้แต่ละออบเจ็กต์ทำงาน

### ก.2.2.5 Sequence Diagram

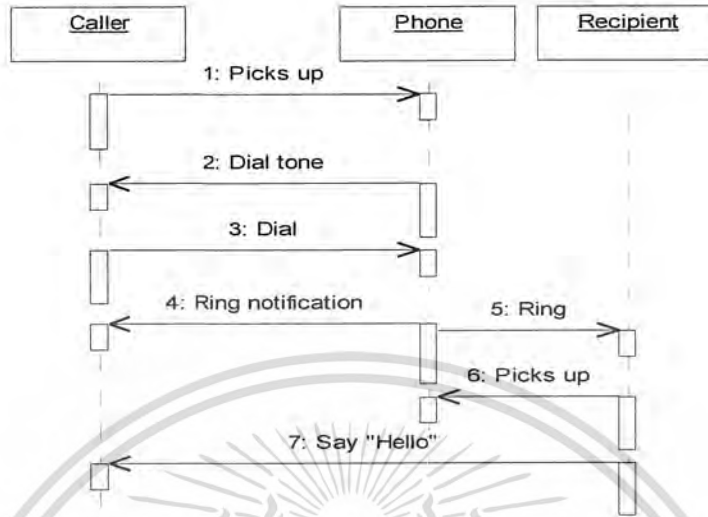
ซีเควินซ์ไดอะแกรมแสดงการติดต่อโต้ตอบระหว่างออบเจ็กต์ ที่เน้นไปที่ลำดับก่อนหลังและเวลาที่มีความสำคัญในการเปลี่ยนแปลง จะมีการส่งแมสเสจ เพื่อให้ออบเจ็กต์ ต่าง ๆ ที่อยู่ในระบบทำงาน โดยใช้สัญลักษณ์ที่เรียกว่า ออบเจ็กต์แมสเสจ ซีเควินซ์ชาท (Object message sequence chart) ในการแสดงดังรูปที่ ก-8

Example

รูปที่ ก-8 object message sequence chart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

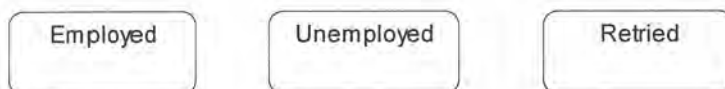
ตัวอย่าง ซีเคิร์ฟซีโคอะแกรมอธิบายการสนทนาทางโทรศัพท์ จะมีการทำงานตามขั้นตอนโดยจะเริ่มต้นจากลำดับที่ 1 เป็นต้นไปจนถึงลำดับสุดท้าย



รูปที่ ก-9 Sequence diagram ของการสนทนาทางโทรศัพท์

#### ก.2.2.6 State Diagram

สเตทโคอะแกรมมีข้ออธิบายคลาสต่างๆ ในระบบโดยจะแสดงทุก ๆ สถานะที่เป็นไปได้และเหตุการณ์ที่ทำให้ออบเจกต์เหล่านั้นเกิดการเปลี่ยนแปลง โดยเหตุการณ์ที่ทำให้เกิดการเปลี่ยนแปลงอาจเกิดจากออบเจกต์อื่นส่งแอสเซสสมา การเปลี่ยนแปลงสถานะเรียกว่า ทรานซิชัน (Transitions) จากรูปที่ ก-10 แสดงสเตทของบุคคล (person) อาจจะเป็นได้ทั้ง 3 สเตท คือ ไม่ถูกจ้าง (Unemployed), ถูกจ้าง(Employed) และ ถูกปลด (Retired)

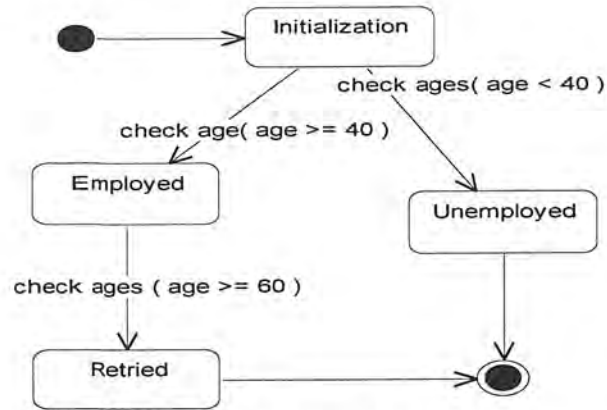


รูปที่ ก-10 State ของ person

สเตทของบุคคลจะขึ้นอยู่กับเงื่อนไขต่าง ๆ ที่บริษัทกำหนด เช่น บริษัทจะรับลูกจ้างอายุ 40 ปีขึ้นไปและจะถูกปลดเมื่ออายุ 60 ปี

เราสามารถนำลักษณะของสถานะต่างๆ ที่เรากำหนดมาแสดงเป็นสเตทโคอะแกรมของบุคคล (person) ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



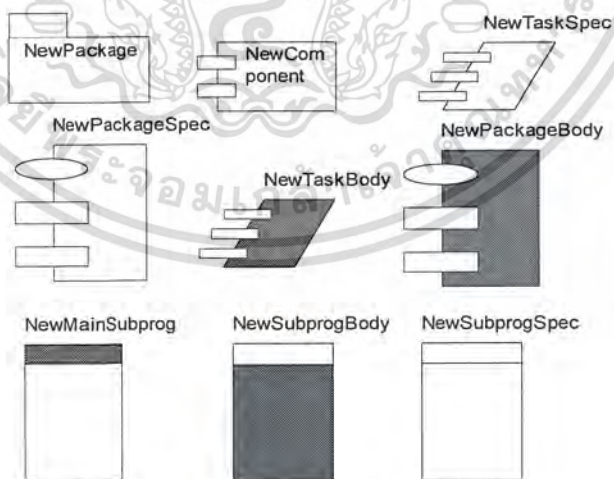
รูปที่ ก-11 State diagram ของ บุคคล

### ก.2.2.7 Activity diagram

แอกติวิตี้ไดอะแกรมแสดงลำดับการไหลของกิจกรรม(Activity) ต่าง ๆ โดยจะอธิบายกิจกรรมต่าง ๆ ในลักษณะของการกระทำในไดอะแกรม จะแสดงเป็นสถานะการกระทำ(Action State) ซึ่งสถานะจะเปลี่ยนไปเมื่อเกิดการกระทำอย่างใดอย่างหนึ่งเกิดขึ้นตามเงื่อนไขหรือการตัดสินใจที่กำหนดไว้เพื่อควบคุมการไหลของกิจกรรมรวมถึงสามารถมีแมสเชสที่รับ - ส่งระหว่างแต่ละกิจกรรม

### ก.2.2.8 Component diagram

คอมโพเนนต์ไดอะแกรมแสดงโครงสร้างทางกายภาพของโค้ดในรูปคอมโพเนนต์ของโค้ด (Code) อาจเป็นส่วนประกอบของซอร์สโค้ด (source code) คอมโพเนนต์ประกอบด้วยข้อมูลเกี่ยวกับ ลอจิกัล คลาส (logical class) ของคอมโพเนนต์ในไดอะแกรม มีการแสดงความสัมพันธ์หรือความพึ่งพากันของคอมโพเนนต์ที่ช่วยในการวิเคราะห์ว่าเมื่อเกิดการเปลี่ยนแปลงของ คอมโพเนนต์หนึ่งจะมีผลต่อคอมโพเนนต์อื่น ๆ อย่างไร ซึ่งช่วยในการโปรแกรม โดยมีสัญลักษณ์ของคอมโพเนนต์ไดอะแกรม ดังนี้

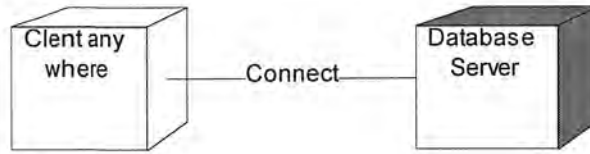


รูปที่ ก-12 สัญลักษณ์ของ Component Diagram

### ก.2.2.9 Deployment diagram

ดีพลอยเม้นต์ไดอะแกรมแสดงสถาปัตยกรรมทางกายภาพของส่วนประกอบต่างๆของฮาร์ดแวร์ (hardware) ซึ่งถูกเรียกว่า โหนด (Node) ในระบบซึ่งสามารถแสดงเป็นโหนดและการเชื่อมต่อระหว่างกัน เอกสารนี้เป็นเอกสารที่ส่งมอบให้วิศวกรใช้งานเพื่อการรักษาเท่านั้น เมื่อผู้ดูแลเห็นใบใช้ประโยชน์ท่านกรุณาไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดของการเชื่อมต่อ นอกจากนี้ภายในโหนด ยังสามารถมีคอมโพเน้น หรือออบเจกต์ที่สามารถปฏิบัติกับโหนดเพื่อแสดงว่าโปรแกรมส่วนใดถูกปฏิบัติบนโหนดใดและความสัมพันธ์ระหว่างคือมโพเน้นที่อยู่บนโหนด ซึ่งโหนดจะถูกแทนด้วยสัญลักษณ์ของสี่เหลี่ยมลูกบาศก์ดังรูปที่ ก-13



รูปที่ ก-13 Deployment Diagram การเชื่อมต่อระหว่าง Client ขากที่ต่าง ๆ และ Database

### ก.2.3 Model element

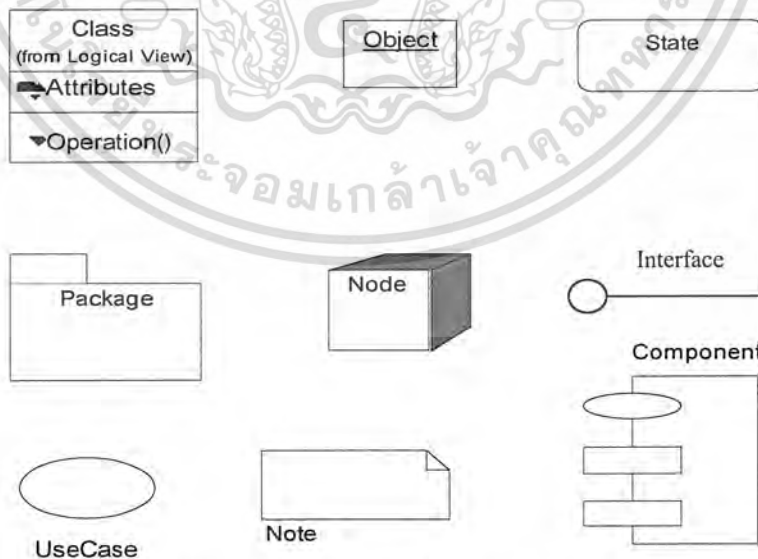
เป็นสัญลักษณ์ที่ใช้ใน ไดอะแกรมเพื่อแสดงหรือเป็นตัวแทนของสิ่งต่าง ๆ เช่น คลาส ,ออบเจกต์ ,แมสเชส และความสัมพันธ์ ต่าง ๆ ที่มี เช่น

Association ใช้เชื่อมต่อระหว่างอุปกรณ์ที่มีความสัมพันธ์กัน

Generalization บางครั้งเรียกว่า อินเฮริเท้น (inheritance) ซึ่งเป็นความสัมพันธ์แบบพิเศษระหว่างอุปกรณ์ หนึ่งที่มีต่ออีกอุปกรณ์หนึ่ง

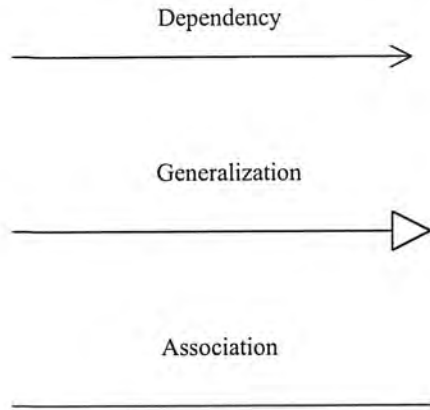
Dependency การพึ่งพากันระหว่างอุปกรณ์ในทางใดทางหนึ่ง

Aggregation เป็นรูปแบบการแสดงอุปกรณ์ที่ประกอบจากอุปกรณ์อื่นหลายอุปกรณ์



รูปที่ ก-14 model element ที่ใช้ร่วมกันใน diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Aggregation(a from of association)



รูปที่ ก-15 ตัวอย่างของความสัมพันธ์แบบต่างๆ



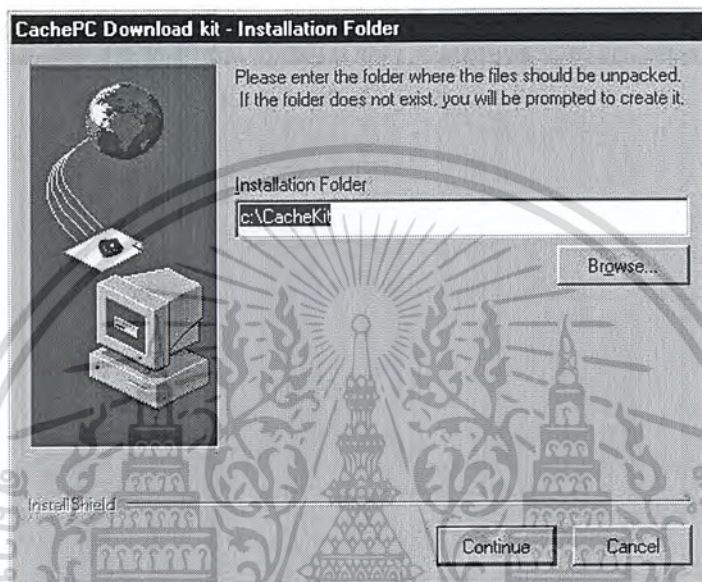
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ข.

### ระบบจัดการฐานข้อมูลเชิงวัตถุคาเช่ (Cache')

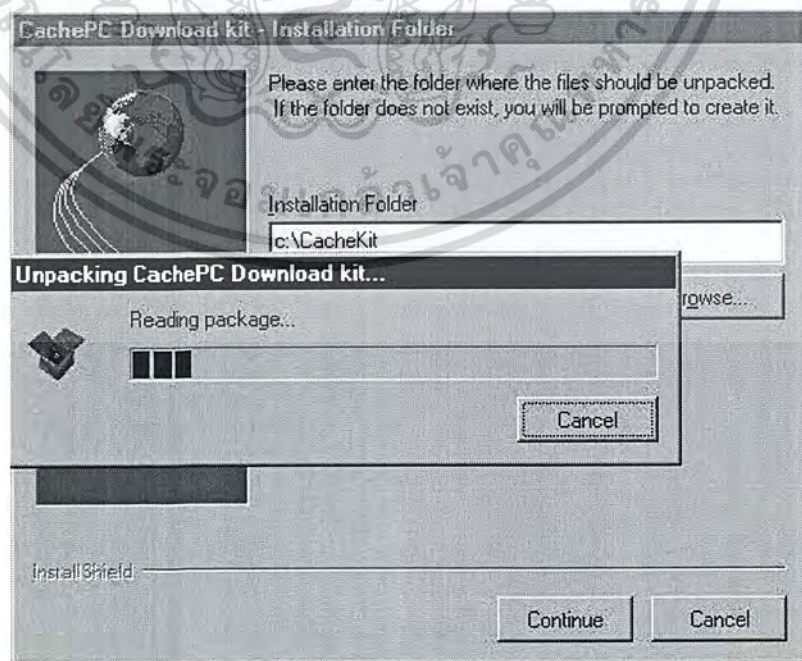
#### ข.1 ขั้นตอนการติดตั้งคาเช่

1. ใส่แผ่นซีดีรอมคาเช่ลงในไดรฟ์ซีดีรอม จากนั้นระบบก็จะทำการสอบถามถึงตำแหน่งโฟลเดอร์ที่เราต้องการติดตั้ง ดังรูปที่ ข-1



รูปที่ ข-1 แสดงการติดตั้งโฟลเดอร์ในตำแหน่งที่เราต้องการ

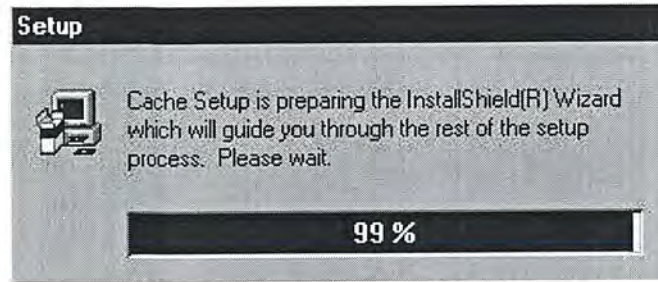
2. จากนั้นระบบจะทำการติดตั้งโปรแกรมตามที่เราได้กำหนดตำแหน่งโฟลเดอร์ ดังรูปที่ ข-2



รูปที่ ข-2 แสดงการติดตั้งโปรแกรมตามที่เราได้กำหนดตำแหน่งโฟลเดอร์

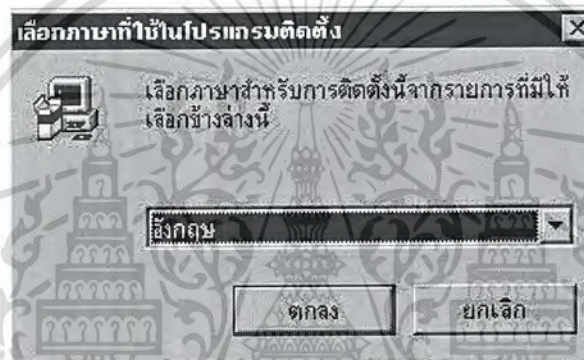
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ขั้นตอนถัดไประบบก็จะทำการเซตอัพ



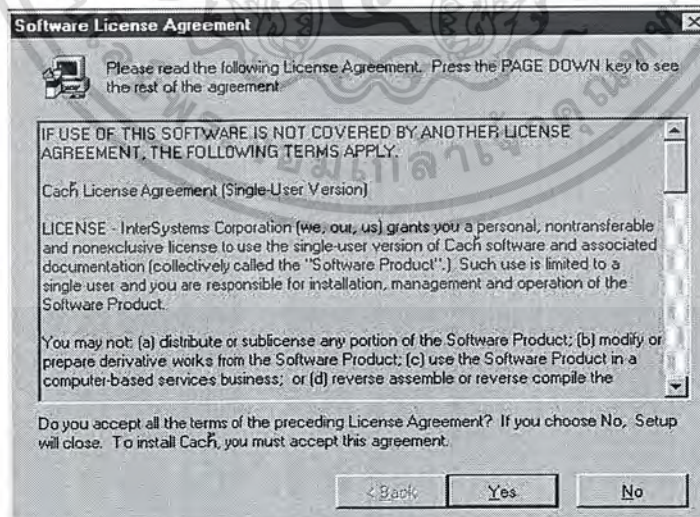
รูปที่ ข-3 รูปแสดงหน้าต่างคาเซเซตอัพ

4. จากนั้น โปรแกรมคาเซก็จะให้ทำการเลือกภาษาสำหรับการติดตั้ง



รูปที่ ข-4 แสดงการเลือกภาษาสำหรับการติดตั้งระบบจัดการฐานข้อมูลคาเซ

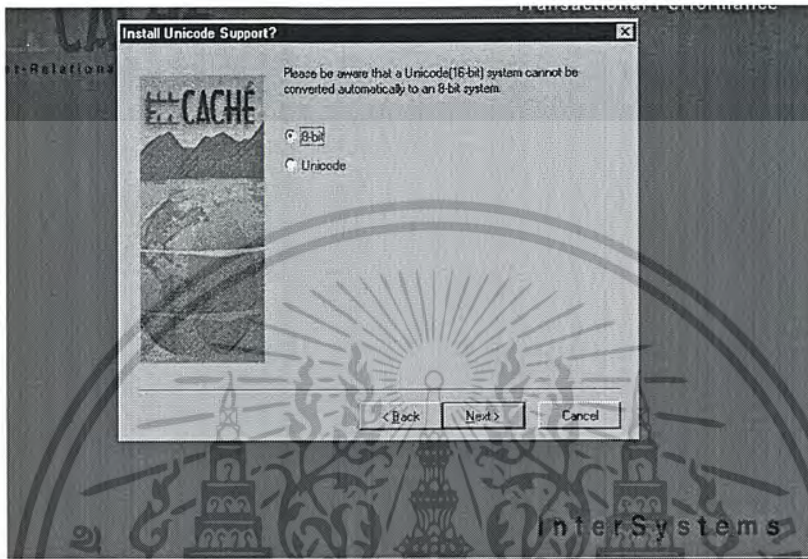
5. ขั้นตอนต่อไประบบสอบถามเกี่ยวกับข้อตกลงในลิขสิทธิ์ และแสดงรายละเอียดเกี่ยวกับลิขสิทธิ์ ให้ทำการตอบใช่



รูปที่ ข-5 แสดงการสอบถามเกี่ยวกับข้อตกลงในลิขสิทธิ์ และแสดงรายละเอียดเกี่ยวกับลิขสิทธิ์

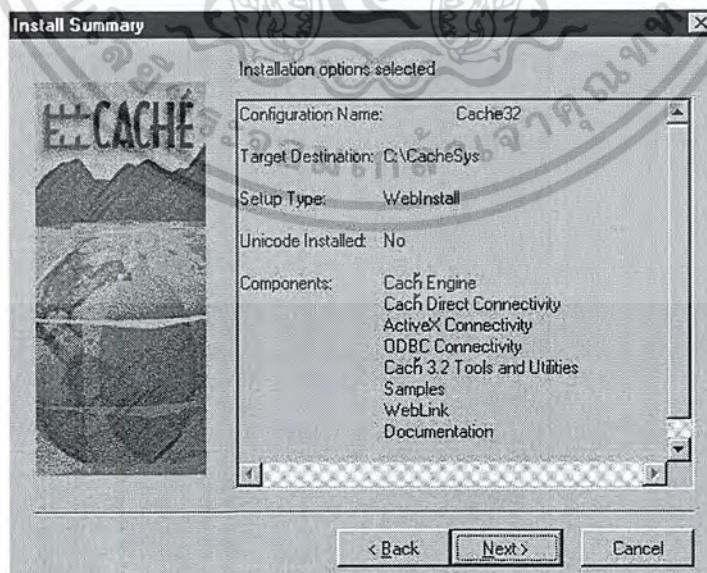
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ภายหลังจากที่เราได้ทำการตอบไข้ไปแล้ว ระบบก็จะทำการสอบถามถึงการแสดงผลว่าต้องการแสดงเป็นแบบ 8บิตหรือแบบยูนิโค้ด(Unicode) ให้เราทำการเลือกรูปแบบการแสดงผล จากนั้นให้ทำการกดปุ่มถัดไป ดังรูปที่ ข-6



รูปที่ ข-6 แสดงหน้าต่างการติดตั้งการแสดงผลของระบบจัดการฐานข้อมูลคาเซ่

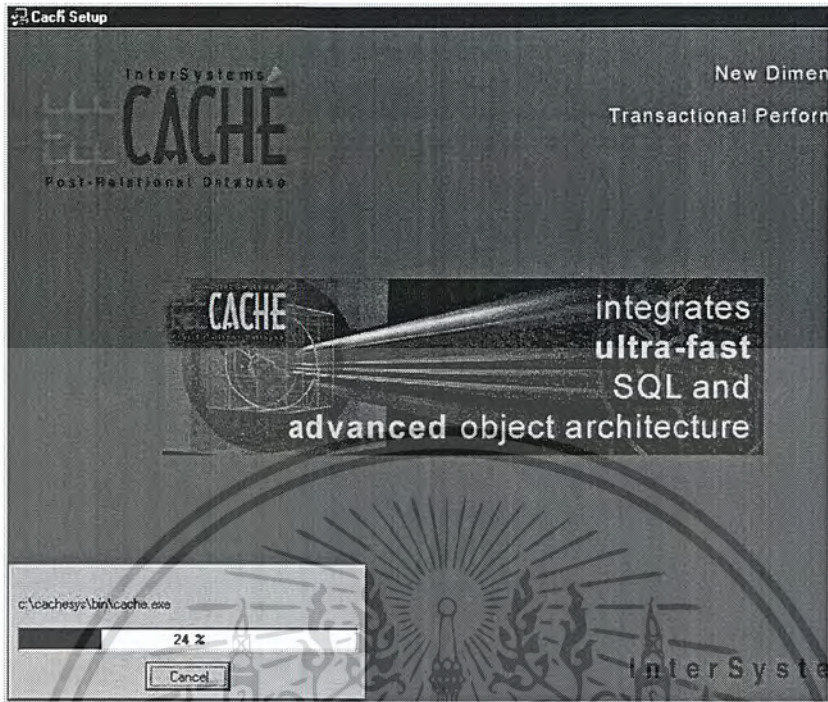
7. แสดงรายละเอียดเกี่ยวกับข้อมูลที่เราได้ทำการเลือกในการติดตั้งระบบจัดการฐานข้อมูลคาเซ่



รูปที่ ข-7 รูปแสดงการสรุปรายละเอียดเกี่ยวกับข้อมูลที่เราได้ทำการเลือกในการติดตั้งระบบจัดการฐานข้อมูลคาเซ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 8. จากนั้นระบบจัดการฐานข้อมูลก็จะทำการเซตอัพระบบ



รูปที่ ข-8 แสดงการเซตอัพระบบของระบบจัดการฐานข้อมูลแคช

### ข.2 ขั้นตอนการสร้างเนมสเปส(Name Space)

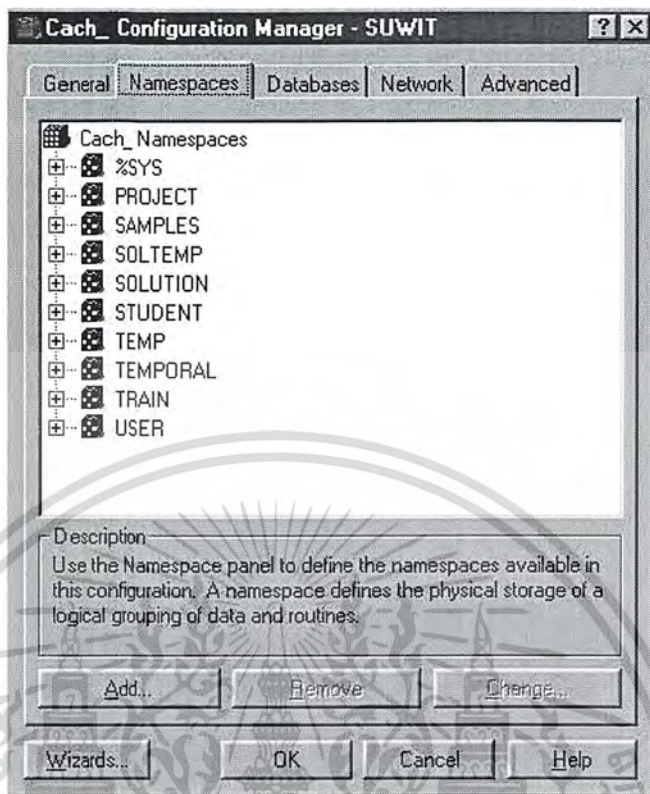
#### 1 ให้เริ่มต้นด้วยการเลือก Configuration Manager



รูปที่ ข-9 แสดงการเลือก Configuration Manager

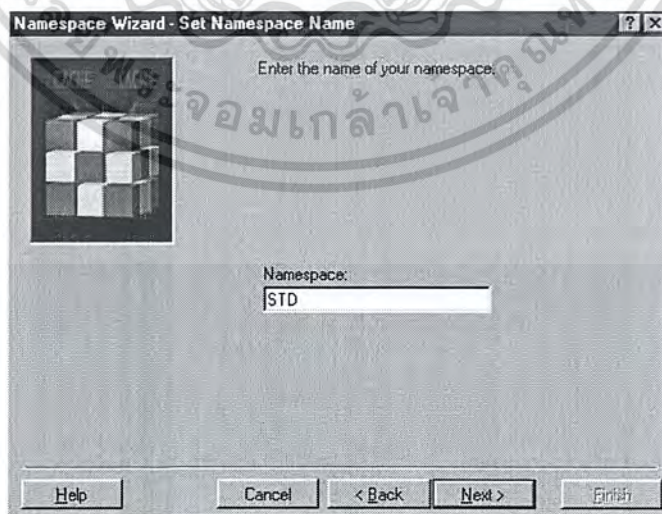
#### 2 จากนั้นให้เลือกแท็บ Namespace แล้วทำการกดปุ่ม Add

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-10 แสดงการเลือกแท็บ Namespace แล้วทำการกดปุ่ม Add

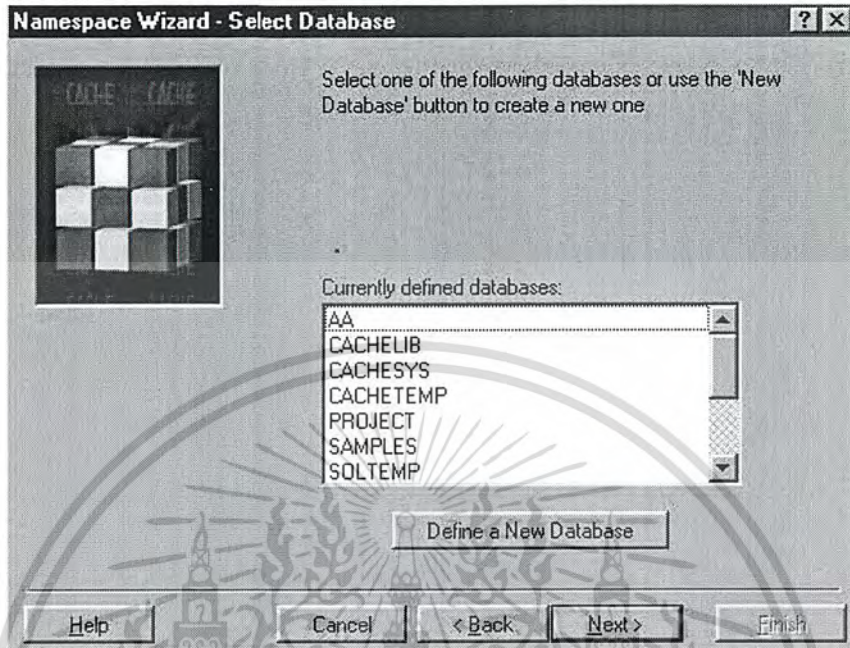
- 3 จากนั้น โปรแกรมจะให้คุณใส่ชื่อเนมสเปส ให้ทำการตั้งชื่อเนมสเปสที่คุณต้องการ ในที่นี้ใส่ตั้งชื่อเนมสเปสคือSTD



รูปที่ ข-11 แสดงการใส่ค่าของเนมสเปส

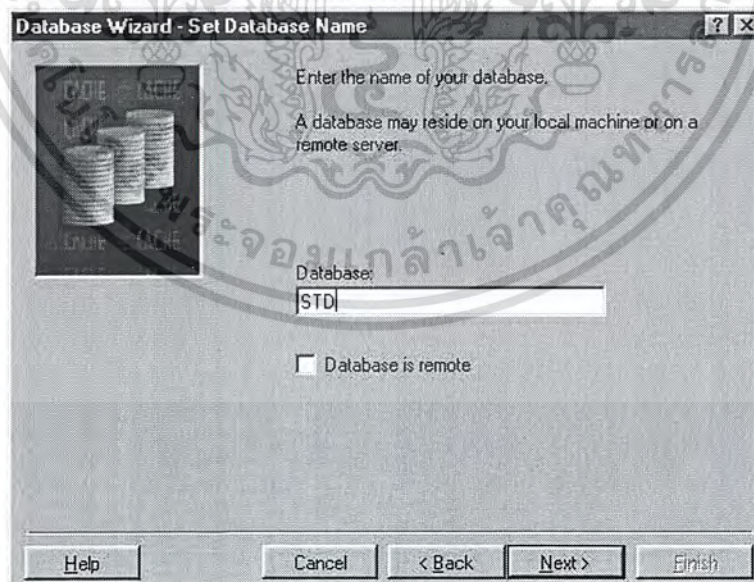
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4 ขั้นตอนต่อไปเป็นการกำหนดค่าเบส เราสามารถเลือกค่าเบสที่มีอยู่แล้ว หรือสามารถทำการกำหนดค่าเบสใหม่ก็ได้ ในที่นี้จะทำการเลือกค่าเบสใหม่ ให้คลิกปุ่ม Define a New Database



รูปที่ ข-12 แสดงการกำหนดค่าเบส

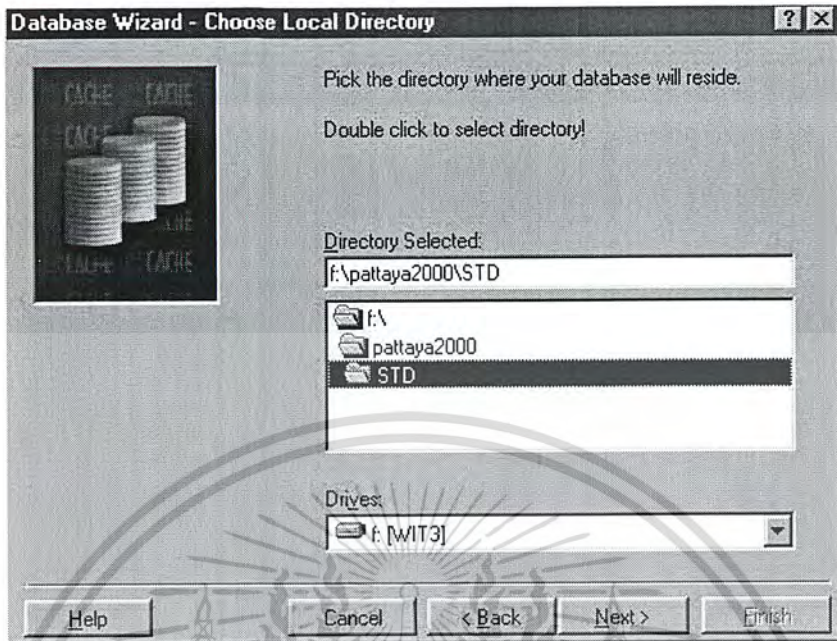
5 จากนั้นให้ทำการใส่ชื่อของค่าเบส ในที่นี้ใส่ชื่อSTD



รูปที่ ข-13 แสดงการใส่ชื่อของค่าเบส

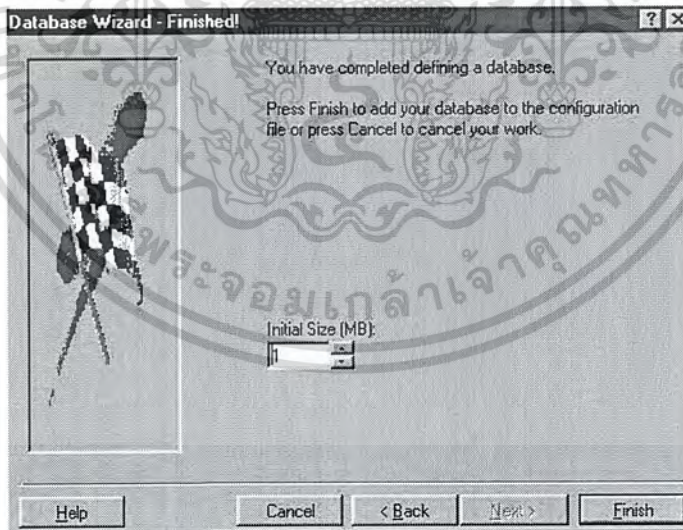
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6 กำหนดตำแหน่งที่ต้องการให้ทำการเก็บดาต้าเบส



รูปที่ ข-14 แสดงการกำหนดตำแหน่งที่ต้องการให้ทำการเก็บดาต้าเบส

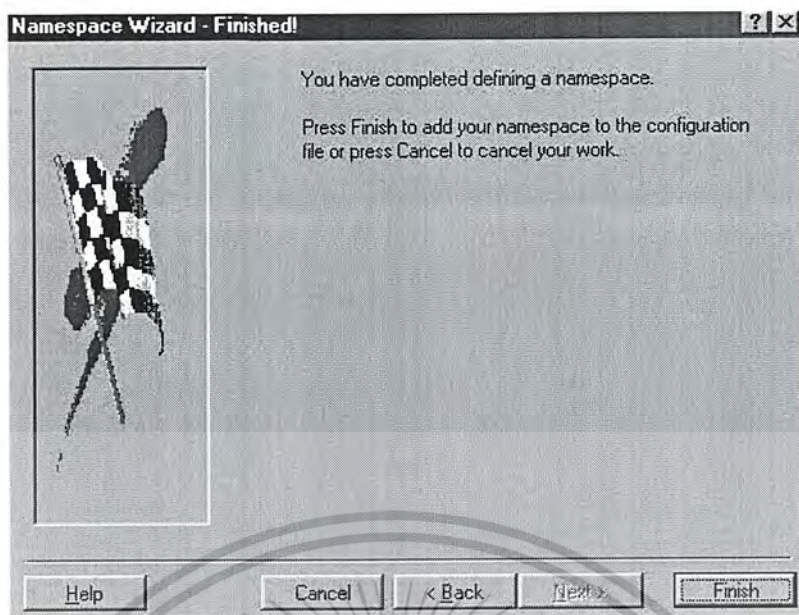
## 7 กำหนดขนาดเริ่มต้นของดาต้าเบส



รูปที่ ข-15 แสดงการขนาดเริ่มต้นของดาต้าเบส

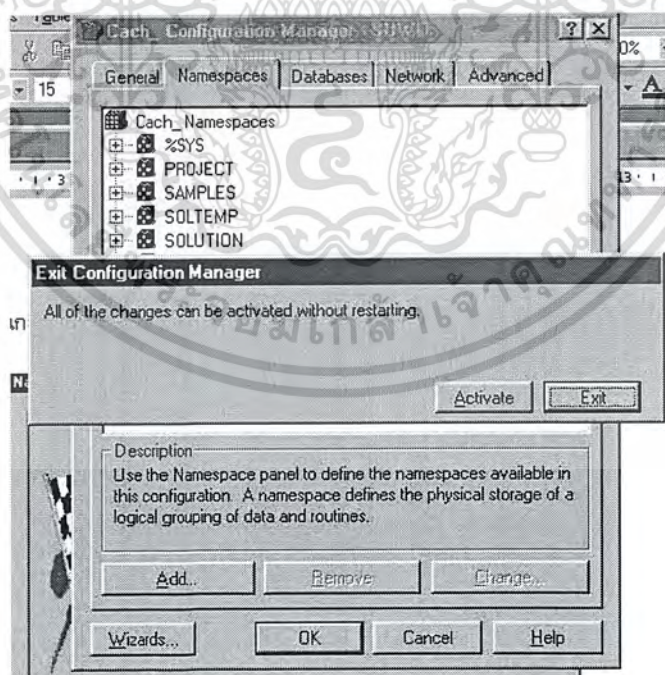
## 8 จากนั้น โปรแกรมจะทำการบอกถึงการจบขั้นตอนการสร้างเนมสเปส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-16 แสดงการจบขั้นตอนการสร้างเนมสเปส

- 9 หลังจากนั้นจะกลับมาหน้าจอดังรูปที่ ข-10 ให้คลิกปุ่ม OK
- 10 ต่อมาโปรแกรมก็จะถามว่าให้เราเปลี่ยนแปลงตามที่เราได้ระบุ โดยไม่ต้องมีการรีสตาร์ทใหม่หรือไม่ ให้ทำการกด Activate

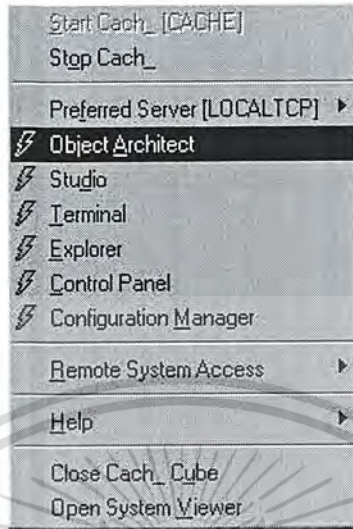


รูปที่ ข-17 แสดงการการสอบถามการเปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

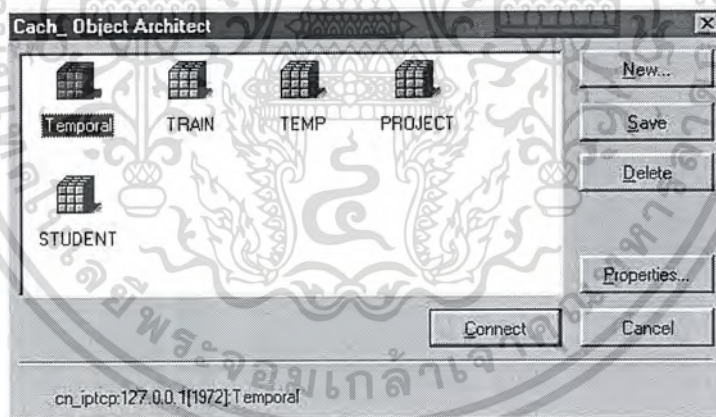
### ข.3 ขั้นตอนของการนำเนมสเปสที่ได้กำหนดไว้แล้วมาใช้งาน

1 ให้เริ่มต้นด้วยการเลือก Object Architect



รูปที่ ข-18 แสดงการเลือก Object Architect

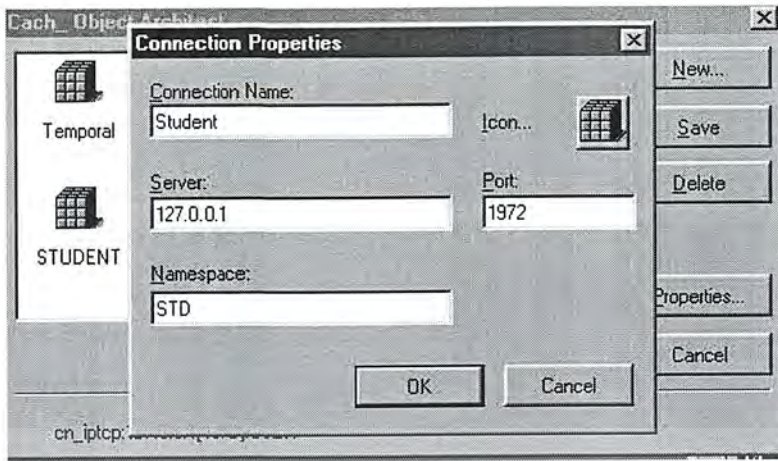
2 จากนั้นจะปรากฏหน้าจอของการติดต่อไปยังเนมสเปสโดยผ่านชื่อที่ใช้ในการติดต่อ



รูปที่ ข-19 แสดงการหน้าจอของการติดต่อไปยังเนมสเปสโดยผ่านชื่อที่ใช้ในการติดต่อ

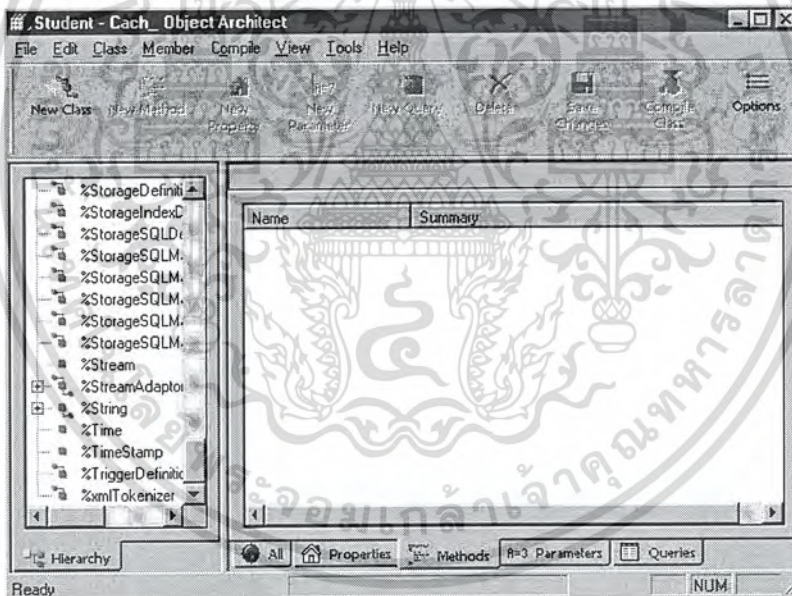
- 3 เนื่องจากเราต้องการเรียกใช้เนมสเปสที่เราต้องการสร้าง ให้เราทำการกดปุ่ม New
- 4 จะปรากฏหน้าต่าง Connection Properties จากนั้นให้เราทำการใส่ชื่อที่ใช้ในการติดต่อและชื่อเนมสเปสที่เราต้องการติดต่อด้วย จากนั้นให้กด OK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-20 แสดงหน้าจอแสดงการใส่ชื่อที่ใช้ในการติดต่อและชื่อเนมสเปสที่เราต้องการติดต่อด้วย

- 5 จากนั้นให้ทำการเลือกชื่อที่เราเพิ่งได้ทำการ New แล้วทำการกด Connect จากนั้นก็จะเข้าสู่หน้าต่างของ Cache\_Object Architect โดยจะเป็นมีการแสดงคลาสตามที่มีในดาต้าเบส

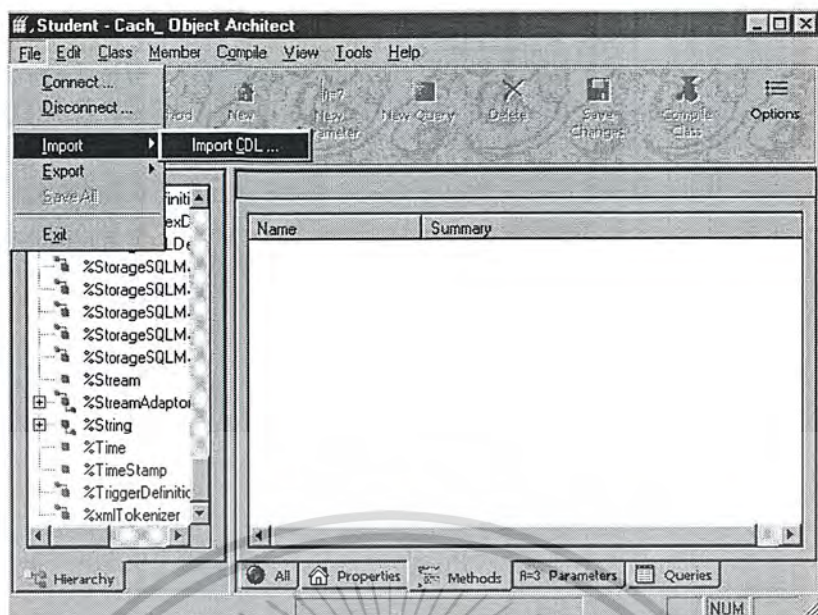


รูปที่ ข-21 แสดงหน้าต่าง Cache\_Object Architect

#### ข.4 ขั้นตอนอิมพอร์ตซีดีแอลไฟล์

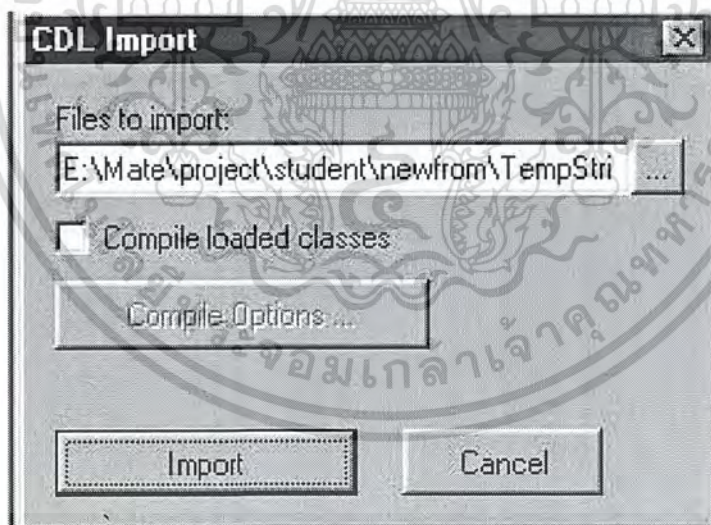
- 1 ให้คลิกที่เมนู File -> Import -> Import CDL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-22 แสดงวิธีการอิมพอร์ตซีดีแอลไฟล์

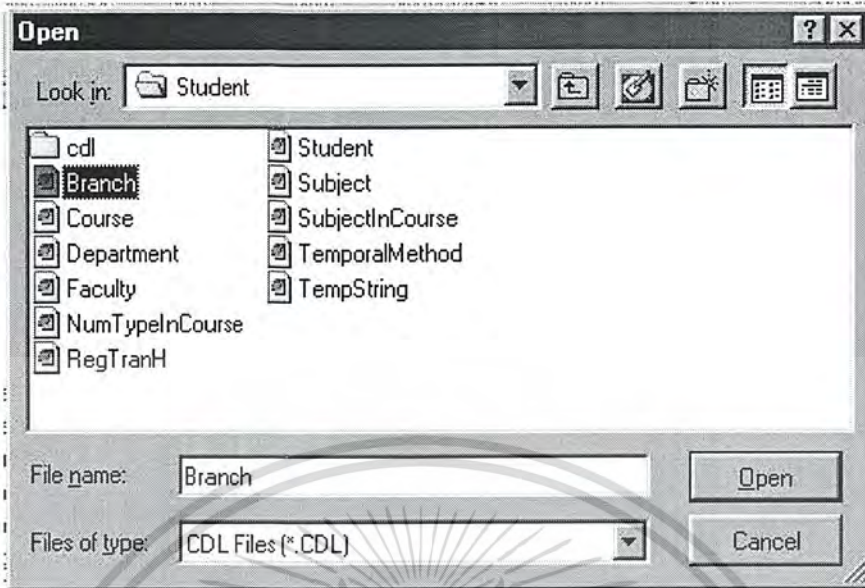
- 2 จากนั้นจะปรากฏหน้าต่างดังรูปที่ ข-22 เราสามารถทำการเลือกซีดีแอลไฟล์ที่ต้องการอิมพอร์ตจากปุ่มด้านข้าง



รูปที่ ข-23 แสดงหน้าต่างซีดีแอลอิมพอร์ต

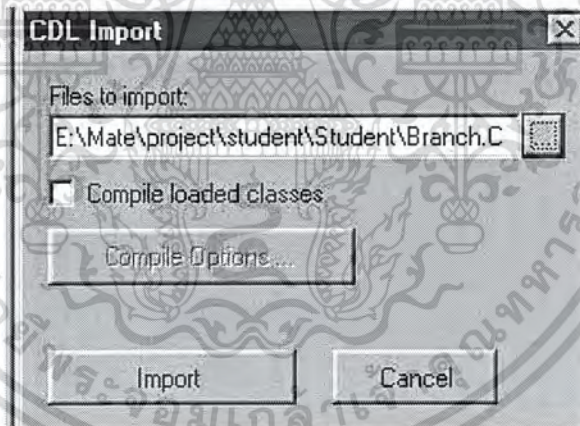
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3 จากนั้นให้ทำการเลือกไฟล์ที่ต้องการอิมพอร์ตในที่นี้ทำการเลือกไฟล์Branch.cdl จากนั้นกดปุ่มOpen



รูปที่ ข-24 แสดงหน้าต่างของการเปิดซิติแอลไฟล์

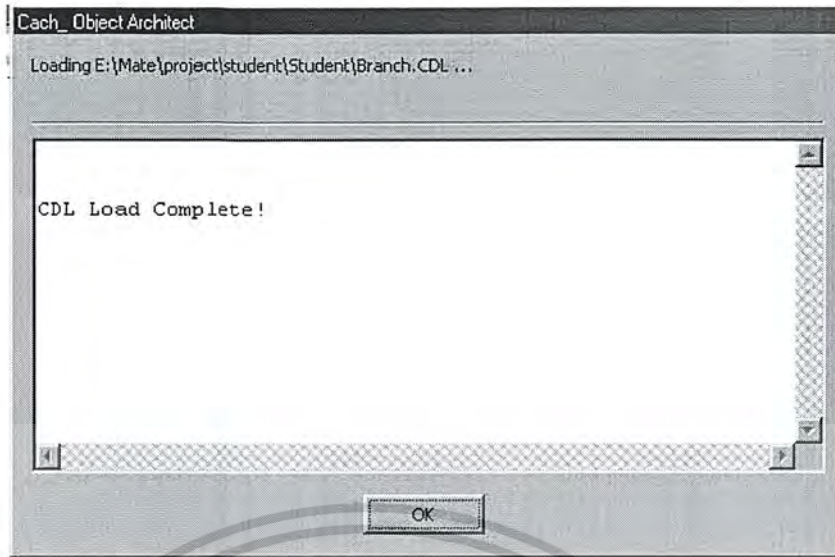
- 4 หลังจากนั้นก็จะกลับมายังหน้าต่างซิติแอลอิมพอร์ต จากนั้นให้ทำการกดปุ่มอิมพอร์ต



รูปที่ ข-25 แสดงหน้าต่างซิติแอลอิมพอร์ตภายหลังจากการเลือกซิติแอลไฟล์

- 5 หลังจากนั้นโปรแกรมก็จะทำการอิมพอร์ตซิติแอลไฟล์ลงดาต้าเบส พร้อมกับแสดงหน้าต่างบอกถึงสถานะการโหลดซิติแอลไฟล์

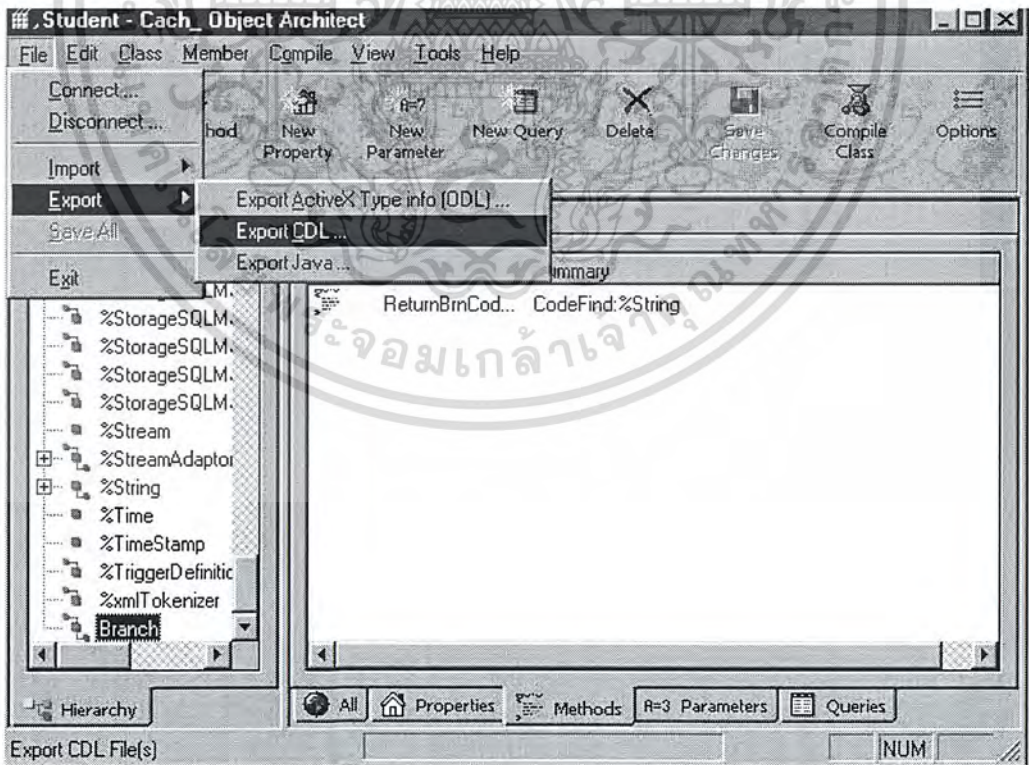
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-26 แสดงหน้าต่างบอกถึงสถานะการโหลดซิติแอลไฟล์

#### ข.5 ขั้นตอนของการเอ็กพอร์ตคลาสเป็นซิติแอลไฟล์

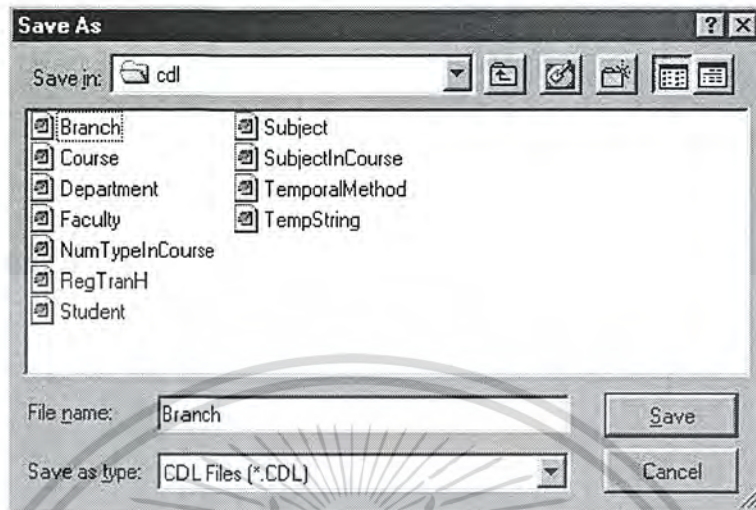
- 1 ให้คลิกที่เลือกคลาสที่ต้องการเอ็กพอร์ต ในที่นี้เลือกคลาสBranch จากนั้นให้คลิกเมนู File -> Export -> Export CDL



รูปที่ ข-27 แสดงวิธีการเอ็กพอร์ตคลาส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 จากนั้นจะปรากฏหน้าต่างดังรูปที่ ข-28 เราต้องทำการเลือกตำแหน่งที่จะให้เก็บซิดีแอลไฟล์ที่ต้องการเอ็กพอร์ตจากคลาส จากนั้นให้กดปุ่มเซฟ



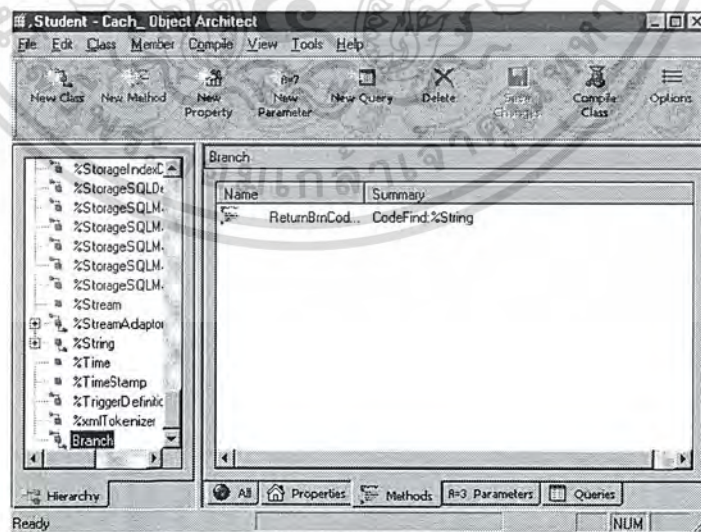
รูปที่ ข-28 แสดงหน้าต่างการเซฟเอ็กพอร์ตคลาส

3 จากนั้น โปรแกรมจะทำการเซฟเป็นซิดีแอลไฟล์ในตำแหน่งที่ได้ระบุไว้

#### ข.6 ขั้นตอนการคอมไพล์คลาสในคาเซ่

ภายหลังจากที่เราได้ทำการเอ็กพอร์ตซิดีแอลไฟล์แล้ว เราจะต้องทำการคอมไพล์คลาสก่อนจึงจะทำงานได้ ซึ่งมีวิธีการดังต่อไปนี้

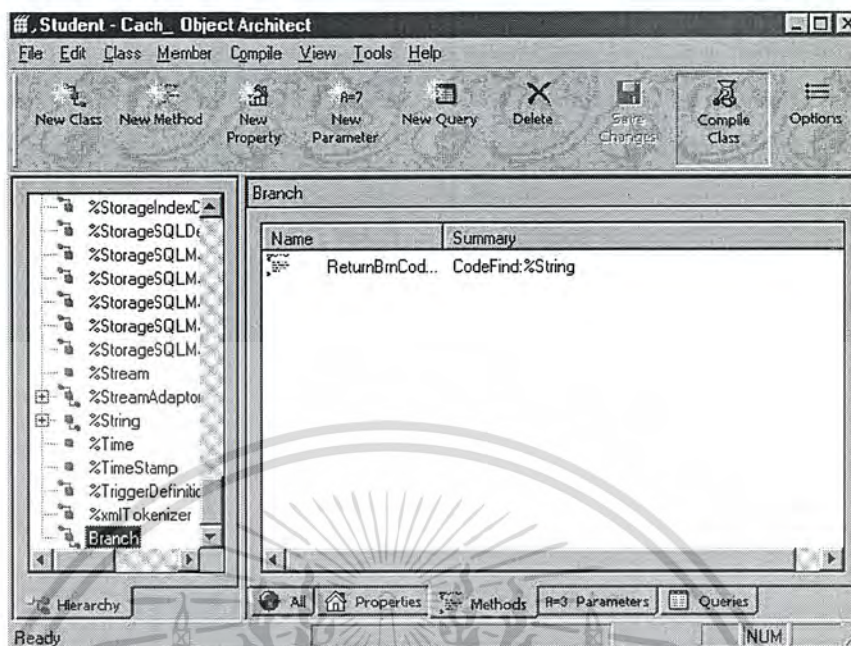
1 ให้ทำการเลือกคลาสที่ต้องการคอมไพล์



รูปที่ ข-29 แสดงหน้าต่างการเลือกคลาสที่ต้องการคอมไพล์

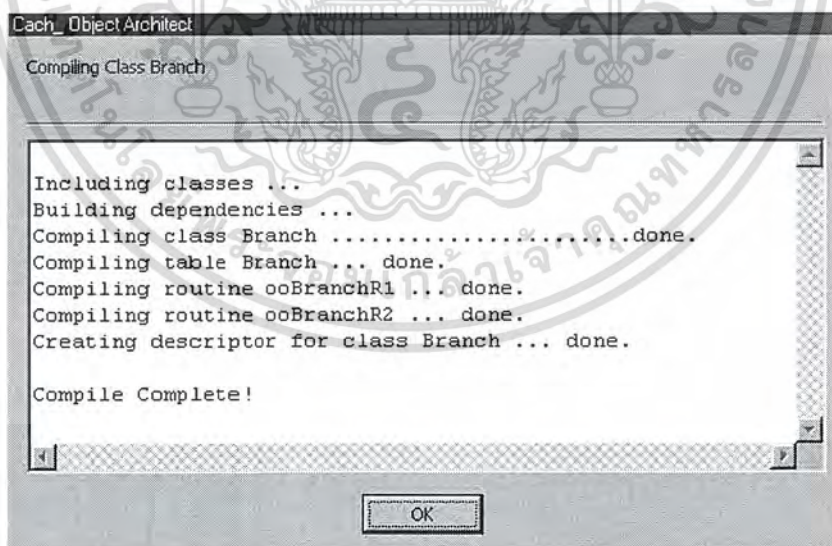
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2 จากนั้นกดปุ่ม Compile Class ใน Cache\_Object Architect



รูปที่ ข-30 แสดงการกดปุ่ม Compile Class ใน Cache\_Object Architect

## 3 จากนั้นโปรแกรมจะแสดงหน้าต่างแสดงผลของการคอมไพล์

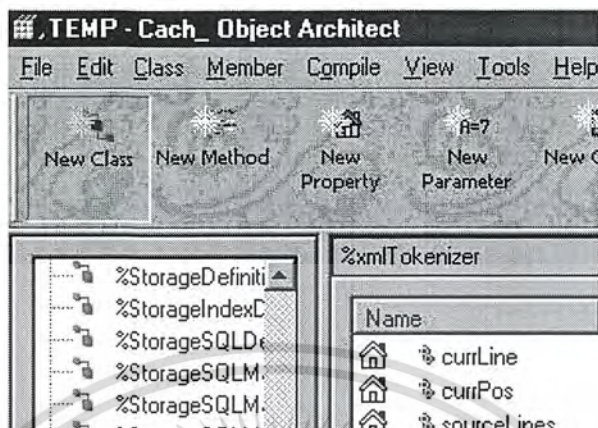


รูปที่ ข-31 แสดงหน้าต่างแสดงผลของการคอมไพล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

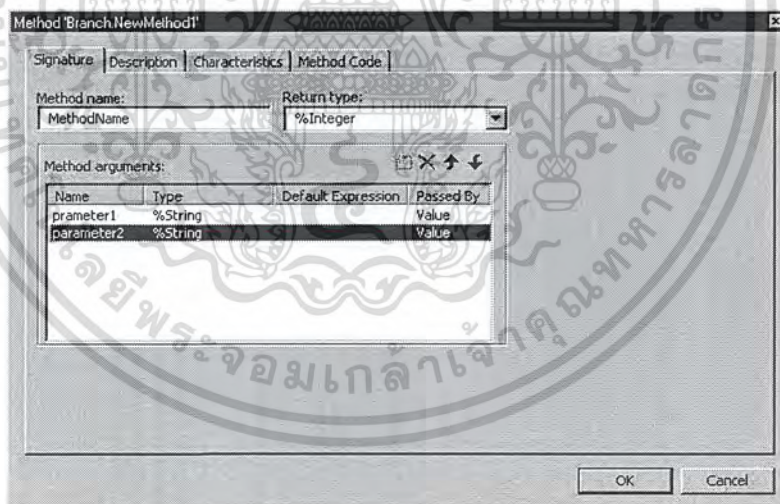
## ข.7 ขั้นตอนการสร้างคลาส

- 1 ให้ทำการกดปุ่ม New Class ทางด้านบน



รูปที่ ข-32 แสดงการกดปุ่ม New Class เพื่อทำการสร้างคลาส

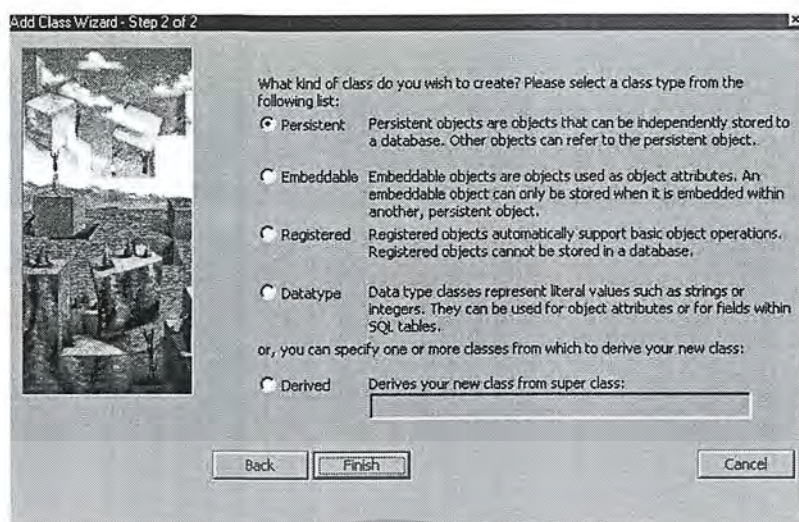
- 2 จะปรากฏหน้าต่าง Add Class Wizard จากนั้นให้ทำการใส่ชื่อคลาสที่ต้องการ จากนั้นให้ทำการกดปุ่ม Next



รูปที่ ข-33 แสดงการใส่ชื่อคลาสที่เราต้องการ

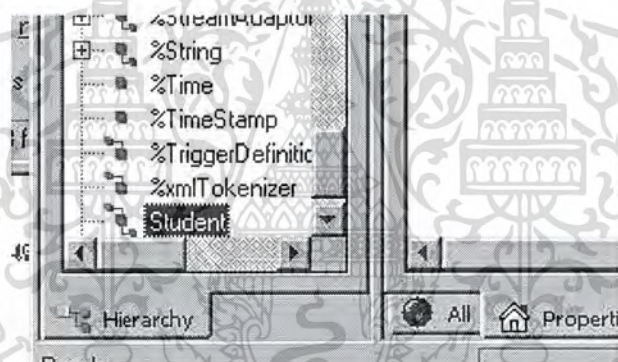
- 3 ในขั้นตอนต่อมาโปรแกรมจะถามถึงชนิดของคลาสที่ต้องการสร้างให้ทำการเลือกชนิดของคลาส จากนั้นก็ให้กดปุ่ม Finish

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-34 แสดงการกำหนดชนิดของคลาสที่เราต้องการ

#### 4 จากนั้นโปรแกรมจะทำการสร้างคลาสที่เราได้กำหนด

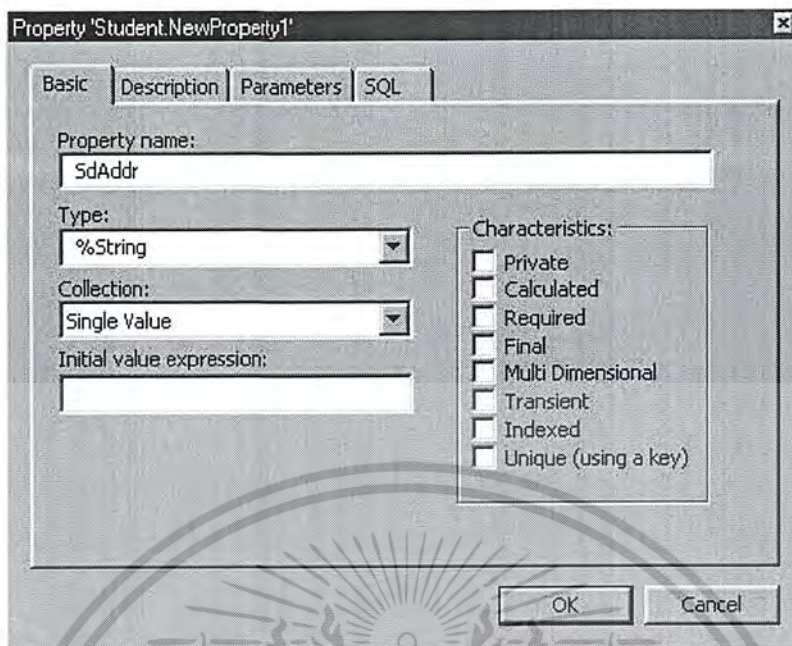


รูปที่ ข-35 แสดงการสร้างคลาสที่เราได้กำหนด

#### ข.8 ขั้นตอนการสร้าง Property

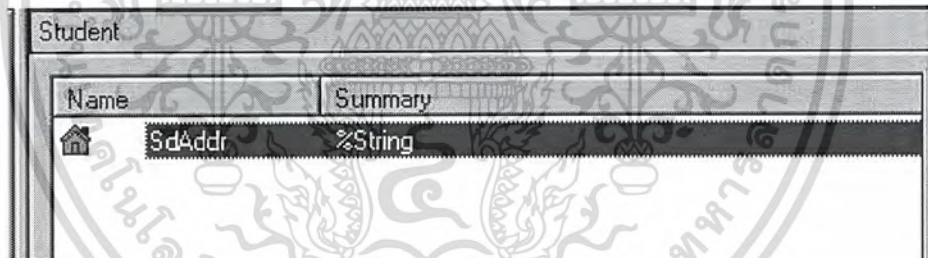
- 1 ให้ทำการเลือกคลาสที่ต้องการสร้างProperty
- 2 ให้ทำการกำหนดProperty โดยให้ทำการกดปุ่ม New Propertyทางด้านบนก่อน จากนั้นจึงค่อยกำหนดรายละเอียดของPropertyนั้นตามที่เรต้องการ แล้วทำการกดปุ่มOK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-36 แสดงรายละเอียดของ Property ที่เราได้กำหนด

- 3 จากนั้นโปรแกรมจะทำการสร้างProperty นั้นขึ้นมา



รูปที่ ข-37 แสดงการสร้าง Property ที่เราได้กำหนด

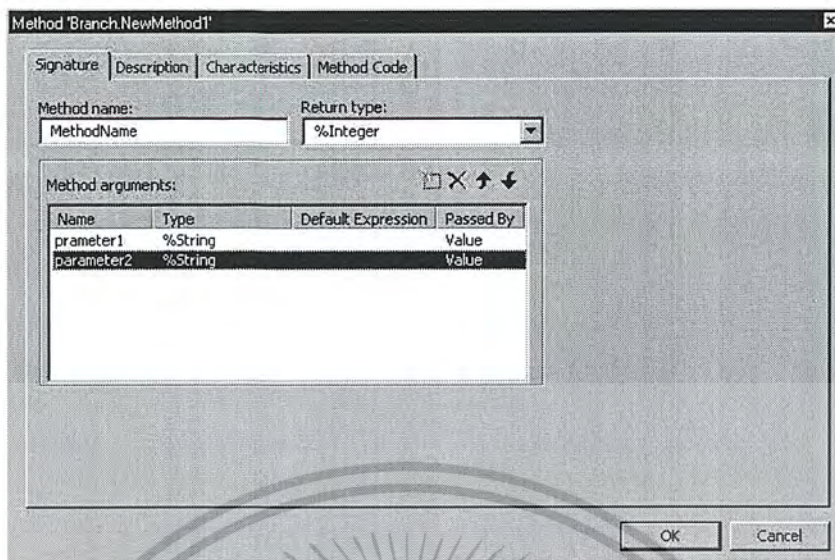
### ข.9 ขั้นตอนการสร้างเมธอด

1 ให้ทำการเลือกคลาสที่ต้องการสร้างเมธอด

2 ให้ทำการกำหนดเมธอดโดยให้ทำการกดปุ่ม New Method

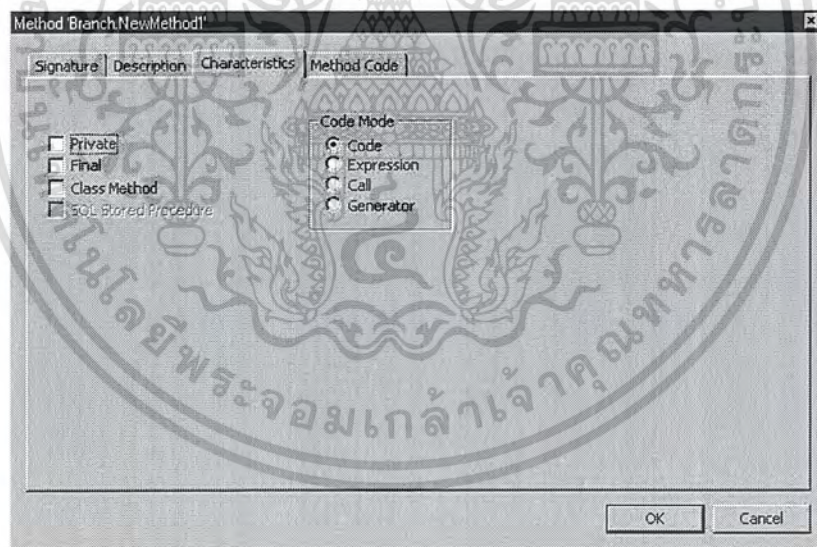
กำหนดชื่อเมธอด หากเมธอดมีการรับค่าอาร์กิวเมนต์ ก็ต้องกำหนดชื่อและชนิดของอาร์กิวเมนต์นั้นๆ และหากเมธอดมีการส่งค่ากลับก็ต้องกำหนดชนิดของค่าที่จะส่งกลับด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-38 แสดงการสร้าง Method ที่เราได้กำหนด

กำหนดชนิดเมธอดว่าจะเป็นเมธอดของคลาส หรือ อินสแตนซ์ เมธอด และลักษณะของโค้ดที่

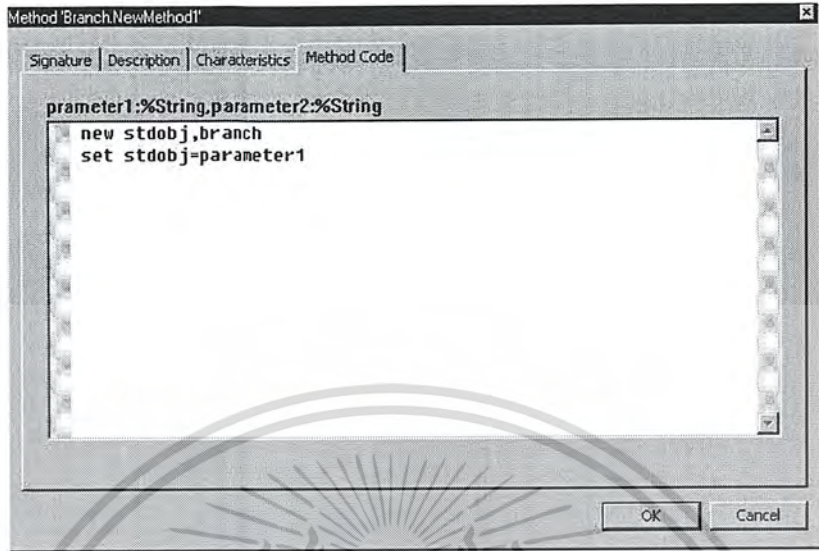


รูปที่ ข-39 แสดงการกำหนดลักษณะของ Method

จะ ใช้ว่ามีลักษณะเป็นแบบใด โดยปกติก็จะใช้เป็นแบบ Code คือเขียนอธิบายการทำงานต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เขียนโค้ดอธิบายการทำงานของเมธอดที่สร้างขึ้นมา



รูปที่ ข-40 แสดงการเขียน Method Code

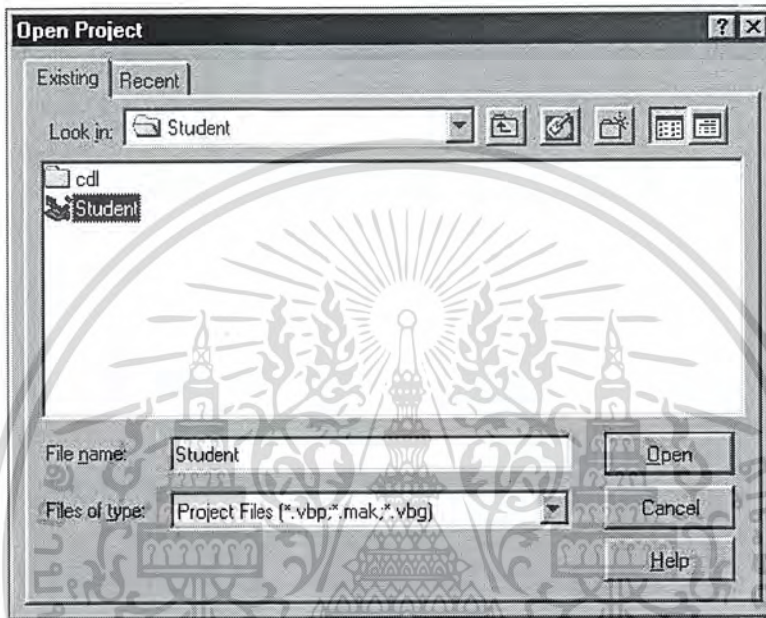
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ค.

### ระบบงานทะเบียนนักศึกษา

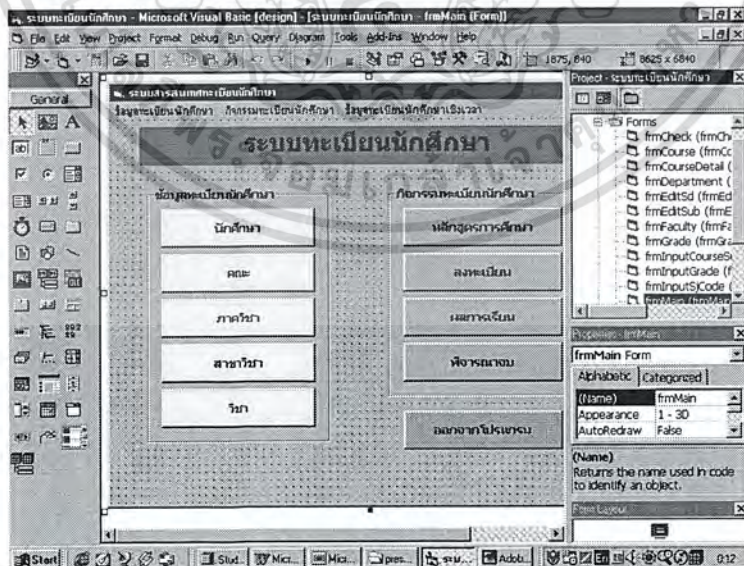
ค.1 ขั้นตอนการใช้งานระบบทะเบียนนักศึกษา

1 เปิด โปรเจกจาก โปรแกรมวิซวลเบสิก



รูปที่ ค-1 แสดงการเปิดโปรเจกจากโปรแกรมวิซวลเบสิก

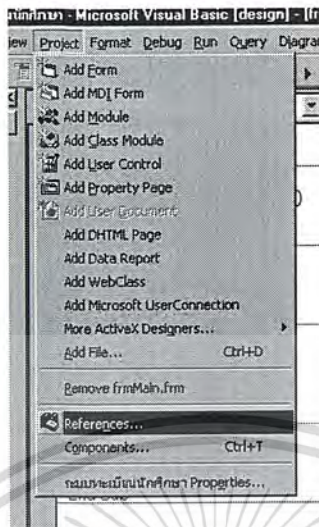
2 จากนั้น โปรแกรมจะทำการ โหลด โปรเจกที่เราได้ทำการเลือกไว้



รูปที่ ค-2 แสดงการโหลดโปรเจกที่เราได้ทำการเลือกไว้

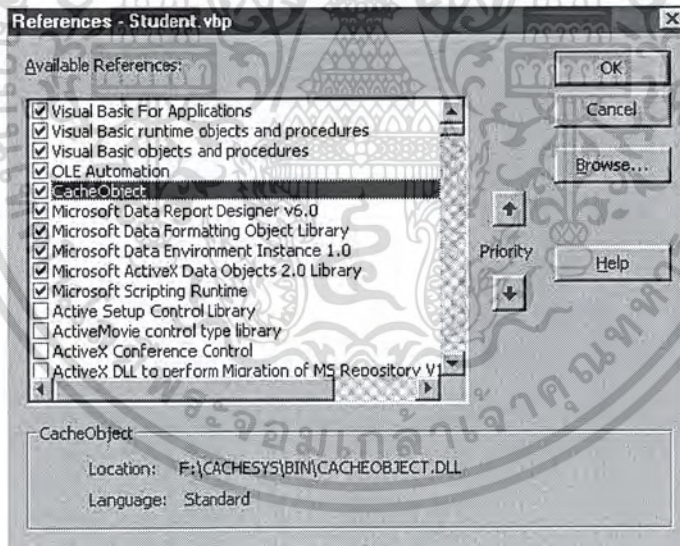
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3..ให้ลองดูที่Project ->References



รูปที่ ค-3 แสดงการเลือก Project ->References

4 จากนั้นให้ดูที่CacheObjectถ้ายังไม่มีเครื่องหมายถูกก็ให้ทำการคลิกเลือก



รูปที่ ค-4 แสดงการเซตReferences

5 จากนั้นให้ทำการดูโค้ดของฟอร์มfrmMain โดยให้ดูการเชื่อมต่อว่าเชื่อมต่อผ่านชื่อให้ถูกต้อง

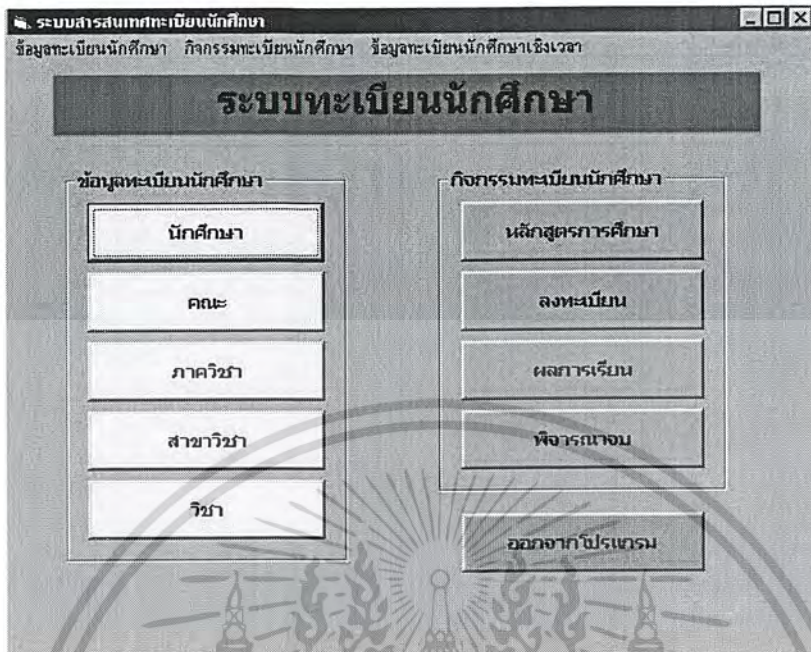
```
Private Sub Form_Load()
```

```
    ConnectObjects "Student"    ในที่นี้ใช้ชื่อStudent
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6 จากนั้นให้ทำการรัน โปรแกรม

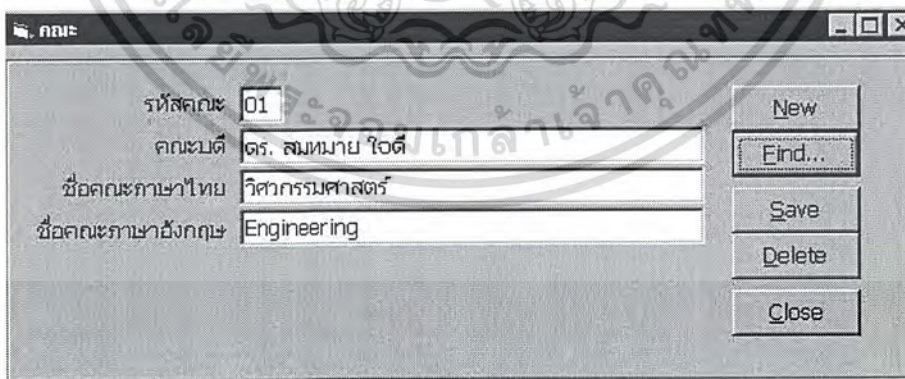


รูปที่ ค-5 แสดงหน้าจอหลักของระบบทะเบียนนักศึกษา

### ค.2 ขั้นตอนการเพิ่มข้อมูลแก้ไขข้อมูลเกี่ยวกับคณะ ภาควิชาสาขาวิชา

#### ค.2.1 สำหรับคณะ

- 1 จากหน้าจอหลักให้คลิกเลือกที่คณะ
- 2 จะปรากฏหน้าต่างของคณะขึ้นมา  
เราสามารถทำการเพิ่มข้อมูลของคณะ แก้ไขข้อมูลเดิม หรือลบข้อมูลที่ไม่ต้องการได้ และสามารถค้นหาคณะที่เราต้องการได้



รูปที่ ค-6 แสดงหน้าต่างคณะ

#### ค.2.2 สำหรับภาควิชา

- 1 จากหน้าจอหลักให้คลิกเลือกที่ภาควิชา
- 2 จะปรากฏหน้าต่างของภาควิชาขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถทำการเพิ่มข้อมูลของภาควิชา แก้ไขข้อมูลเดิม หรือลบข้อมูลที่ไม่ต้องการได้ และสามารถค้นหาภาควิชาที่เราต้องการได้

รูปที่ ค-7 แสดงหน้าต่างภาควิชา

### ค.2.3 สำหรับสาขาวิชา

- 1 จากหน้าจอหลักให้คลิกเลือกที่สาขาวิชา
- 2 จะปรากฏหน้าต่างของสาขาวิชาขึ้นมา

เราสามารถทำการเพิ่มข้อมูลของสาขาวิชา แก้ไขข้อมูลเดิม หรือลบข้อมูลที่ไม่ต้องการได้ และสามารถค้นหาสาขาวิชาที่เราต้องการได้

รูปที่ ค-8 แสดงหน้าต่างสาขาวิชา

ข้อมูลของคณะ ภาควิชา สาขาวิชานั้น เป็นข้อมูลทั่วไปของระบบทะเบียนนักศึกษาที่จำเป็นสำหรับกิจกรรมของระบบทะเบียนนักศึกษา อันได้แก่การจัดทำหลักสูตรการศึกษา ซึ่งกิจกรรมเหล่านั้นจะปฏิบัติได้นั้น จำเป็นต้องมีข้อมูลเหล่านี้เกิดขึ้นมาก่อน ซึ่งข้อมูลเหล่านี้เป็นข้อมูลธรรมดาที่ทุกๆ ไป ไม่ได้จัดทำเป็นข้อมูลเชิงเวลา แต่หากเราให้ความสำคัญกับข้อมูลเหล่านี้สำหรับสถานะของข้อมูลในช่วงเวลาต่างๆ แล้วก็สามารถทำเป็นข้อมูลเชิงเวลาได้ เช่น การเก็บประวัติของคณบดีทำให้ทราบว่าใครเคยเป็นคณบดีมาแล้วบ้าง เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ค.3 ขั้นตอนการเพิ่ม/ลบ/แก้ไขวิชา

- 1 จากหน้าจอหลักให้คลิกเลือกที่วิชา
- 2 จะปรากฏหน้าต่างเพิ่ม/ลบ/แก้ไขวิชา

รูปที่ ค-9 แสดงหน้าต่างเพิ่ม/ลบ/แก้ไขวิชา

- 3 กรณีการเพิ่มรายวิชา ให้ทำการใส่ข้อมูลของวิชาที่ต้องการเพิ่มจากนั้นให้กดปุ่มเพิ่มวิชา โปรแกรมจะทำการเพิ่มวิชาเข้าไปยังฐานข้อมูล

รูปที่ ค-10 แสดงหน้าต่างใส่ข้อมูลวิชา

- 4 กรณีของการแก้ไขข้อมูลของรายวิชาให้เราทำการใส่รหัสวิชาที่ต้องการแก้ไข จากนั้นให้เราทำการกดแป้นโปรแกรมจะทำการค้นหาข้อมูลของรายวิชานั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพิ่ม/ลบ/แก้ไขวิชา

รายวิชา

รหัสวิชา 01074101 หน่วยกิต

ชื่อวิชาภาษาไทย

ชื่อวิชาภาษาอังกฤษ

วันสอบ ตัวอย่าง: (mm/dd/yyyy)

เวลาเริ่มสอบ 0:00:00

เวลาสิ้นสุดการสอบ 0:00:00

คณะ

เพิ่มวิชา ล้างหน้าจอ ลบวิชา ปิดหน้าต่าง

รูปที่ ค-11 แสดงการใส่รหัสวิชาเพื่อทำการโหลดข้อมูลรายวิชา

เพิ่ม/ลบ/แก้ไขวิชา

รายวิชา

รหัสวิชา 01074101 หน่วยกิต 3

ชื่อวิชาภาษาไทย โครงการ 1

ชื่อวิชาภาษาอังกฤษ Project 1

วันสอบ 9/24/01 ตัวอย่าง: (mm/dd/yyyy)

เวลาเริ่มสอบ 9:00:00

เวลาสิ้นสุดการสอบ 12:00:00

คณะ วิศวกรรมศาสตร์

แก้ไขวิชา ล้างหน้าจอ ลบวิชา ปิดหน้าต่าง

รูปที่ ค-12 แสดงข้อมูลรายวิชาที่ถูกโหลด

5 ให้ทำการแก้ไขข้อมูลของวิชา จากนั้นให้ทำการกดปุ่มแก้ไขวิชาเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพิ่ม/ลบ/แก้ไขวิชา

รายวิชา

รหัสวิชา 01074101 หน่วยกิต 3

ชื่อวิชาภาษาไทย โครงการวิศวกรรม1

ชื่อวิชาภาษาอังกฤษ Project Engineering I

วันสอบ 9/24/01 ตัวอย่าง: (mm/dd/yyyy)

เวลาเริ่มสอบ 9 :00:00

เวลาสิ้นสุดการสอบ 12:00:00

คณะ วิศวกรรมศาสตร์

แก้ไขวิชา ล้างหน้าจอ ลบวิชา ปิดหน้าต่าง

รูปที่ ค-13 แสดงแก้ไขข้อมูลรายวิชาที่ถูกโหลด

- 6 โปรแกรมจะทำการแก้ไขข้อมูลวิชาเรียน
- 7 กรณีของการลบวิชาเรียนก็ให้ทำในลักษณะเดียวกับการแก้ไขกล่าวคือให้ทำการ โหลดรายวิชาขึ้นมา ก่อนจากนั้นให้ทำการกดปุ่มลบวิชา วิชาที่เราได้ทำการ โหลดขึ้นมา ก็จะถูกลบ
- 8 ส่วนของปุ่มล้างหน้าจอ นั้น เป็นปุ่มสำหรับในการรีเซตข้อมูลวิชาที่เราได้ทำการ โหลดมา

#### ค.4 ขั้นตอนของการเพิ่ม/ลบ/แก้ไขหลักสูตร

- 1 จากหน้าจอหลักให้คลิกเลือกที่หลักสูตรการศึกษา
- 2 จากนั้นจะปรากฏหน้าต่างหลักสูตร โดยในหน้าต่างจะมีการแสดงรายชื่อหลักสูตรที่มีอยู่ในดาต้าเบส

หลักสูตร

รหัสหลักสูตร หลักสูตรปี วันเริ่มปีหลักสูตร

ชื่อหลักสูตรภาษาไทย คณะ

ชื่อหลักสูตรภาษาอังกฤษ สาขาวิชา

จำนวนหน่วยกิตทั้งหมดในหลักสูตร จำนวนปีในหลักสูตร สาขาวิชา

ผ่านวิชาบังคับคือก หน่วยกิต

ผ่านวิชาเลือกเสรี หน่วยกิต

รายละเอียดหลักสูตร

ค้นหาข้อมูลคณะ ...

รายชื่อหลักสูตร

| รหัสหลักสูตร | ชื่อหลักสูตรภาษาไทย     | ชื่อหลักสูตรภาษาอังกฤษ       | หลักสูตรปี |
|--------------|-------------------------|------------------------------|------------|
| 01           | ปริญญาตรีวิศวกรรมศาสตร์ | Bachelor Degree Engineering  |            |
| 02           | ปริญญาตรีวิศวกรรมศาสตร์ | Bachelor Degree Engineering  |            |
| 03           | ปริญญาตรีสถาปัตยกรรม    | Bachelor Degree Architecture |            |

เพิ่มหลักสูตร ล้างหลักสูตร ล้างหน้าจอ ออกจากหน้าจอ

รูปที่ ค-14 แสดงหน้าต่างหลักสูตร

3 กรณีการเพิ่มหลักสูตร ให้เรากรอกรายละเอียดของหลักสูตรใหม่ แล้วทำการกดปุ่มเพิ่มหลักสูตร เอกสารนี้เป็นเอกสารที่ส่งงานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ค-15 แสดงการกรอกรายละเอียดหลักสูตร

4 หลักสูตรจะถูกทำการเพิ่มลงในดาต้าเบส ในที่นี้คือหลักสูตรที่มีรหัส04 นั่นเอง

รูปที่ ค-16 แสดงการเพิ่มหลักสูตร

5 หากเราต้องการทำการแก้ไขหลัก ให้ดับเบิลคลิกเลือกหลักสูตรที่ต้องการแก้ไข ภายหลังจากแก้ไขก็ให้กดปุ่มแก้ไขหลักสูตร ข้อมูลก็ถูกบันทึกลงดาต้าเบส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสหลักสูตร 04    หลักสูตรปี 2544    วันเริ่มใช้หลักสูตร 04/07/2002

ชื่อหลักสูตรภาษาไทย ปริญญาตรีวิศวกรรมศาสตร์    คณะ

ชื่อหลักสูตรภาษาอังกฤษ Bachelor Degree Engineering    ภาควิชา

จำนวนหน่วยกิตทั้งหมดในหลักสูตร 141    จำนวนปีในหลักสูตร 4    สาขาวิชา

ผ่านวิชาบังคับเลือก 8    หน่วยกิต

ผ่านวิชาเลือกเสรี 8    หน่วยกิต

รายละเอียดหลักสูตร    ค้นหาข้อมูลคณะ ...

รายชื่อหลักสูตร

| รหัสหลักสูตร | ชื่อหลักสูตรภาษาไทย        | ชื่อหลักสูตรภาษาอังกฤษ       | หลักสูตรปี |
|--------------|----------------------------|------------------------------|------------|
| 01           | ปริญญาตรีวิศวกรรมศาสตร์    | Bachelor Degree Engineering  |            |
| 02           | ปริญญาตรีวิศวกรรมศาสตร์    | Bachelor Degree Engineering  |            |
| 03           | ปริญญาตรีสถาปัตยกรรมศาสตร์ | Bachelor Degree Architecture |            |
| 04           | ปริญญาตรีวิศวกรรมศาสตร์    | Bachelor Degree Engineering  |            |

แก้ไขหลักสูตร    ลบหลักสูตร    สบหน้าจอ    ออกจากหน้าจอ

รูปที่ ค-17 แสดงการเลือกหลักสูตรที่ต้องการแก้ไข

- 6 กรณีนี้จะแสดงการแก้ไขโดยจะทำการแก้ไขข้อมูลของคณะ/ภาควิชา/สาขาวิชา ให้ทำการกดปุ่มค้นหาข้อมูลคณะ จะปรากฏหน้าต่างคิวรี่ เพื่อหาข้อมูลของคณะ/ภาควิชา/สาขาวิชา

Query Name: BranInfo

Query    Cancel    Select

| ID | FcName | DptName | BrnName |
|----|--------|---------|---------|
|----|--------|---------|---------|

รูปที่ ค-18 แสดงหน้าต่างเพื่อหาข้อมูลคณะ/ภาควิชา/สาขาวิชา

- 7 ให้ทำการกดปุ่ม Query โดยตรง Query Name ให้ใช้ชื่อ BranchInfo จะปรากฏข้อมูลของคณะ/ภาควิชา/สาขาวิชาที่มีในขณะนั้น จากรูปจะพบว่ามีการแสดงรายละเอียดข้อมูลของคณะ/ภาควิชา/สาขาวิชา

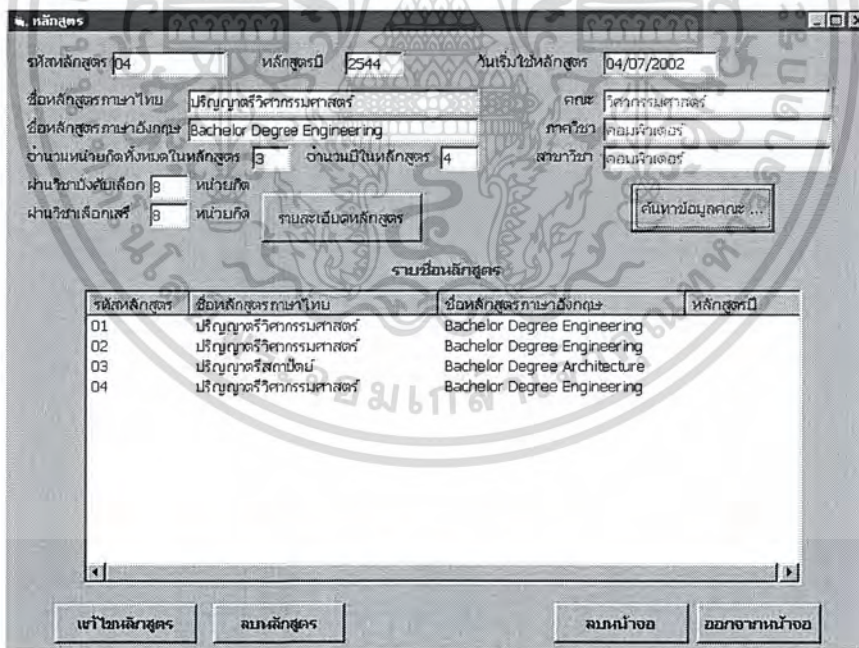
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่มีในขณะนั้น



รูปที่ ค-19 แสดงข้อมูลคณะ/ภาควิชา/สาขาวิชาที่ได้จากการคิวรี

- 8 ให้ทำการดับเบิลคลิกคณะ/ภาควิชา/สาขาวิชาที่ต้องการ จะสังเกตว่าตรงข้อมูลของคณะ/ภาควิชา/สาขาวิชาจะถูกเพิ่มเข้ามา จากนั้นให้ทำการกดปุ่มแก้ไขหลักสูตร ข้อมูลการเปลี่ยนแปลงจะถูกบันทึกลงฐานข้อมูล



รูปที่ ค-20 แสดงข้อมูลคณะ/ภาควิชา/สาขาวิชาที่เพิ่มเข้าไปในหลักสูตร

- 9 กรณีของการลบหลักสูตรก็ทำคล้ายกัน โดยดับเบิลคลิกเลือกหลักสูตรที่ต้องการลบจากนั้นข้อมูลของหลักสูตรก็จะปรากฏ จากนั้นให้ทำการกดปุ่มลบหลักสูตร หลักสูตรนั้นก็จะถูกลบออกจากคาด้าเบส
- 10 ในส่วนของปุ่มลบหน้าจอก็จะทำการรีเซตข้อมูลของหลักสูตรที่ได้ทำการโหลดมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11 ส่วนปุ่มของรายละเอียดหลักสูตร เราจะใช้ในกรณีของการเพิ่ม/ลบวิชาในหลักสูตร

### ค.5 ขั้นตอนของการเพิ่ม/ลบรายวิชาในหลักสูตร

1 ให้ทำการเลือกหลักสูตรที่ต้องการเพิ่ม/ลบรายวิชาในหลักสูตร

หลักสูตร

รหัสหลักสูตร: 04    หลักสูตรที่: 2544    วันที่เพิ่มหลักสูตร: 04/07/2002

ชื่อหลักสูตรภาษาไทย: วิศวกรรมศาสตรบัณฑิต    คณะ: วิศวกรรมศาสตร์  
ชื่อหลักสูตรภาษาอังกฤษ: Bachelor Degree Engineering    ภาควิชา: วิศวกรรมเครื่องกล  
จำนวนหน่วยกิตทั้งหมดในหลักสูตร: 8    จำนวนปีในหลักสูตร: 4    สาขาวิชา: วิศวกรรมเครื่องกล  
นำวิชาบังคับเลือก: 0    หน่วยกิต:    รวมเป็นยอดหลักสูตร:    ค้นหาข้อมูล...

รายชื่อหลักสูตร:

| รหัสหลักสูตร | ชื่อหลักสูตรภาษาไทย | ชื่อหลักสูตรภาษาอังกฤษ       | หลักสูตรที่ |
|--------------|---------------------|------------------------------|-------------|
| 01           | วิศวกรรมศาสตรบัณฑิต | Bachelor Degree Engineering  |             |
| 02           | วิศวกรรมศาสตรบัณฑิต | Bachelor Degree Engineering  |             |
| 03           | วิศวกรรมศาสตรบัณฑิต | Bachelor Degree Architecture |             |
| 04           | วิศวกรรมศาสตรบัณฑิต | Bachelor Degree Engineering  |             |

แก้ไขหลักสูตร    ลบหลักสูตร    เพิ่มหลักสูตร    ลบจากหลักสูตร

รูปที่ ค-21 แสดงการเลือกหลักสูตรที่ต้องการเพิ่ม/ลบรายวิชาในหลักสูตร

2 จะปรากฏหน้าต่างฟอร์มรายละเอียดหลักสูตร ซึ่งจะแสดงรายวิชาที่มีอยู่แล้วในหลักสูตร

เพิ่ม/ลบรายวิชาเลือกหลักสูตร

รหัสหลักสูตร: 04    ชื่อหลักสูตรภาษาไทย: วิศวกรรมศาสตรบัณฑิต    หลักสูตรที่: 2544

รหัสวิชา:    สาขาวิชา:    มติของวิชา:    วิชาของปี:    วิชาของแผน:    หมายเหตุ:

ชื่อวิชาภาษาไทย:    ชื่อวิชาภาษาอังกฤษ:    วิชาในหลักสูตร:

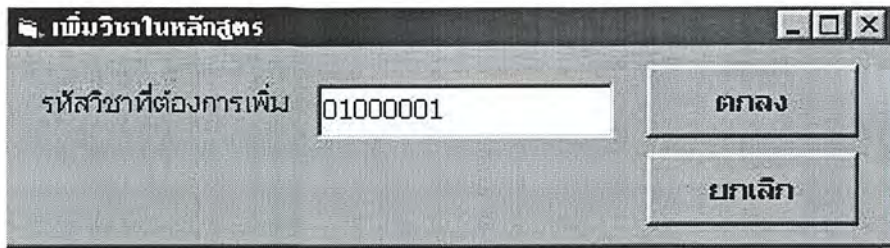
| รหัสวิชา | ชื่อวิชาภาษาไทย | ประเภทวิชา | ปี | เทอม | หน่วยกิต |
|----------|-----------------|------------|----|------|----------|
|          |                 |            |    |      |          |

เพิ่มวิชา    ลบวิชา    ส่งหน้าจอ    ปิดหน้าต่าง

รูปที่ ค-22 แสดงหน้าต่างรายละเอียดหลักสูตร

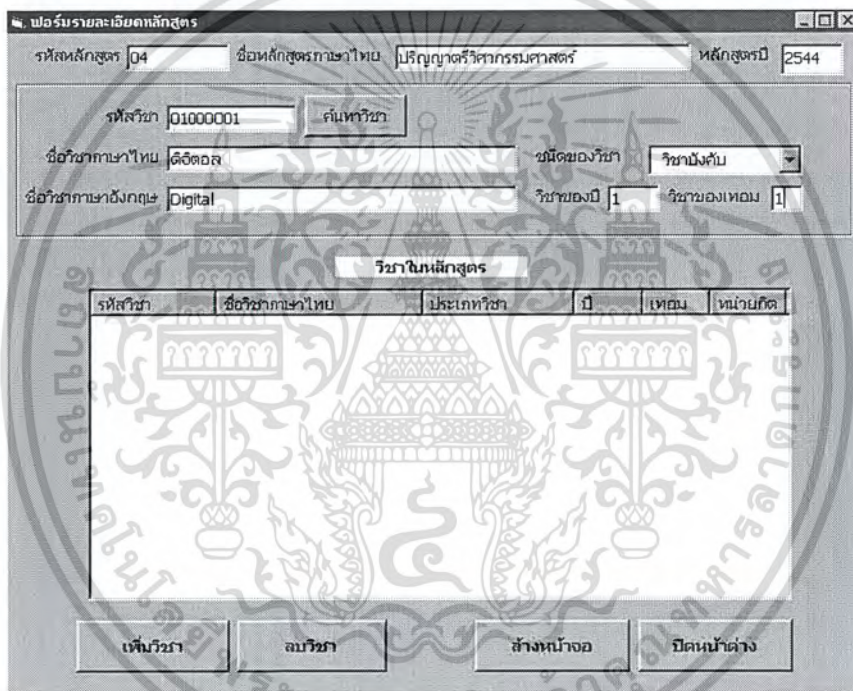
3 กรณีของการเพิ่มวิชาในหลักสูตร ให้เราทำการกดเลือกค้นหาววิชา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค-23 แสดงหน้าต่างค้นหาวิชา

- 4 ระบบจะทำการค้นหาวิชาและแสดงรายละเอียดของวิชา จากนั้นให้เราทำการกำหนดชนิดของวิชา เป็นวิชาของปีอะไร เทอมอะไร



รูปที่ ค-24 แสดงรายละเอียดวิชาที่ค้นหาได้ในฟอร์มรายละเอียดวิชา

- 5 ให้ทำการกดปุ่มเพิ่มวิชา จะสังเกตว่าวิชาที่เราได้ทำการค้นหาจะถูกเพิ่มเข้าไปในหลักสูตรของวิชานั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟอร์มรายละเอียดหลักสูตร

รหัสหลักสูตร 04 ชื่อหลักสูตรภาษาไทย ปริญญาตรีวิศวกรรมศาสตร์ หลักสูตรมี 2544

รหัสวิชา  ค้นหาวิชา

ชื่อวิชาภาษาไทย  ชนิดของวิชา

ชื่อวิชาภาษาอังกฤษ  วิชาของมี  วิชาของไม่มี

วิชาในหลักสูตร

| รหัสวิชา | ชื่อวิชาภาษาไทย | ประเภทวิชา | ปี | เทอม | หน่วยกิต |
|----------|-----------------|------------|----|------|----------|
| 01000001 | ดีอีตอล         | วิชาบังคับ | 1  | 1    | 3        |

เพิ่มวิชา ลบวิชา ล้างหน้าจอ ปิดหน้าต่าง

รูปที่ ค-25 แสดงรายวิชาที่ถูกเพิ่มเข้าไปในหลักสูตร

#### ค.6 ขั้นตอนของการเพิ่ม/แก้ไข/ลบนักศึกษา

- 1 จากหน้าจอหลักให้คลิกเลือกที่นักศึกษา
- 2 จะปรากฏหน้าต่างของข้อมูลนักศึกษา

ข้อมูลส่วนตัวนักศึกษา

รหัสนักศึกษา

ชื่อ-สกุล ภาษาไทย  ชื่อ-สกุล ภาษาอังกฤษ  เพศ

วันที่เกิด  11 มีนาคม 2544 อายุ  เบอร์โทรศัพท์

ชื่อประจำตัว  ชื่อมารดา

ชื่อผู้ประกัน  ชื่อผู้เก็บ

สถานศึกษาเดิม  ภูมิลำเนา

คณะ  สถานะภาพ

ภาควิชา  วันเข้าเรียนการศึกษา  1 มกราคม 2540

สาขาวิชา  วันที่สำเร็จการศึกษา  31 ธันวาคม 10542

เพิ่ม/แก้ไข ประวัติ ล้างหน้าจอ ค้นหาประวัติ ออกจกหน้าจอ

รูปที่ ค-26 แสดงหน้าต่างข้อมูลนักศึกษา

- 3 กรณีการเพิ่มนักศึกษาให้ทำการกรอกข้อมูลของนักศึกษาใหม่
- 4 ภายหลังจากที่ได้ทำการกรอกข้อมูลนักศึกษาแล้วให้ทำการกดปุ่มเพิ่มประวัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5 จากนั้นโปรแกรมจะแสดงข้อความยืนยันการเพิ่มข้อมูลของนักศึกษา

The screenshot shows a web form for student registration. At the top, there is a confirmation dialog box titled "บันทึกข้อมูลสำเร็จ" (Data saved successfully) with the message "ทำการเพิ่มข้อมูล รหัส 40010535" (Added data, ID 40010535) and an "OK" button. The background form contains the following information:

|                      |   |                      |                   |
|----------------------|---|----------------------|-------------------|
| รหัสนักศึกษา         | 40010535  |                      |                   |
| ชื่อ-สกุล ภาษาไทย    | นาย ไพฑูย์                                      | สกุล                 | ไพฑูย์            |
| ชื่อ-สกุล ภาษาอังกฤษ | Mr Pailoon                                      | สกุล                 | Popum             |
| เพศ                  | ชาย   | วันเกิด              | 12 เมษายน 2521    |
| อายุ                 | 22  | เบอร์โทรศัพท์        | 02-7127000        |
| ชื่อบิดา             | นาย สมชาย รัชชาติ                               | ชื่อมารดา            | นาง สมศรี รัชชาติ |
| ที่อยู่ปัจจุบัน      | 1/473 หมู่ 1 แขวงคลองกุ่ม เขตบึงกุ่ม กทม. 10230 |                      |                   |
| สถานที่ศึกษาเดิม     | ร.ร. เตรียมอุดมศึกษา                            |                      |                   |
| คณะ                  | วิศวกรรมศาสตร์                                  | สถานะภาพ             | ผ่าน              |
| ภาควิชา              | วิศวกรรมโยธา                                    | วันที่เข้ารับการศึกษ | 1 มกราคม 2540     |
| สาขาวิชา             | วิศวกรรมโยธา                                    | วันที่สำเร็จการศึกษา | 31 ธันวาคม 10542  |

รูปที่ ค-27 แสดงข้อความยืนยันการเพิ่มข้อมูลของนักศึกษา

6 กรณีการแก้ไขข้อมูลนักศึกษาให้ทำการพิมพ์รหัสนักศึกษา จากนั้นกดแท็บถัดจากนั้นใน ปัจจุบันระบบจะทำการโหลดข้อมูลนักศึกษานั้น จากนั้นให้ทำการแก้ไขข้อมูลนักศึกษา จากนั้นให้กดปุ่มแก้ไขประวัติ

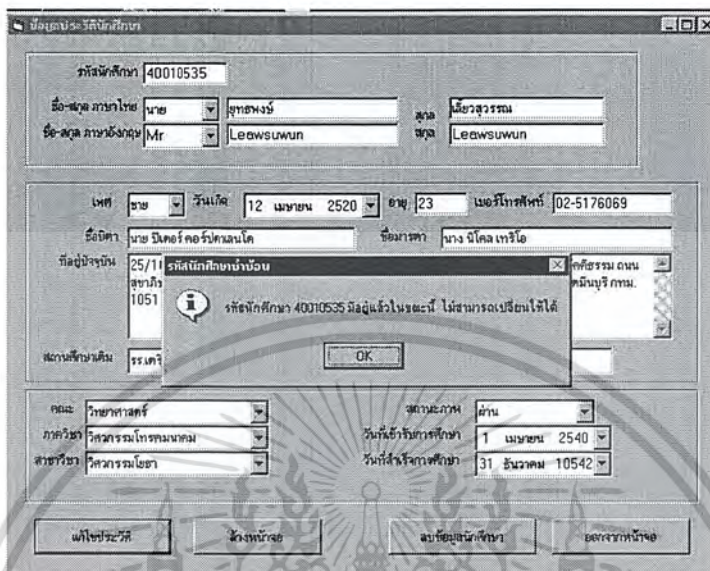
The screenshot shows the same student registration form, but with the "แก้ไขประวัติ" (Edit History) button highlighted. The form contains the following information:

|                      |   |                      |                  |
|----------------------|---|----------------------|------------------|
| รหัสนักศึกษา         | 40010620  |                      |                  |
| ชื่อ-สกุล ภาษาไทย    | นาย ปุทธพงษ์  | สกุล                 | เลี้ยวสุวรรณ     |
| ชื่อ-สกุล ภาษาอังกฤษ | Mr Leawsuwun  | สกุล                 | Leawsuwun        |
| เพศ                  | ชาย   | วันเกิด              | 12 เมษายน 2520   |
| อายุ                 | 23  | เบอร์โทรศัพท์        | 02-5176069       |
| ชื่อบิดา             | นาย นิโคร์ คอบดาเลนโต   | ชื่อมารดา            | นาง นิโคล เทริโอ |
| ที่อยู่ปัจจุบัน      | 25/103 หมู่ 10 ซอย สามีศรีธรรม ถนน สุขุมวิท 3 แขวงมีนบุรี เขตมีนบุรี กทม. 10510 |                      |                  |
| สถานที่ศึกษาเดิม     | ร.ร.เตรียมอุดมศึกษาน้อมเกล้า  |                      |                  |
| วุฒิการศึกษา         | มัธยมศึกษาปีที่ 6   |                      |                  |
| คณะ                  | วิทยาศาสตร์   | สถานะภาพ             | ผ่าน             |
| ภาควิชา              | วิศวกรรมโทรคมนาคม   | วันที่เข้ารับการศึกษ | 1 เมษายน 2540    |
| สาขาวิชา             | วิศวกรรมโยธา  | วันที่สำเร็จการศึกษา | 31 ธันวาคม 10542 |

รูปที่ ค-28 แสดงการโหลดข้อมูลของนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7 กรณีที่การแก้ไขข้อมูลนักศึกษาในส่วนของรหัสนักศึกษา แล้วเกิดมีรหัสศึกษานั้นอยู่ในฐานข้อมูล ระบบจะทำการบอกว่ารหัสข้อมูลนี้มีอยู่ในฐานข้อมูลแล้ว



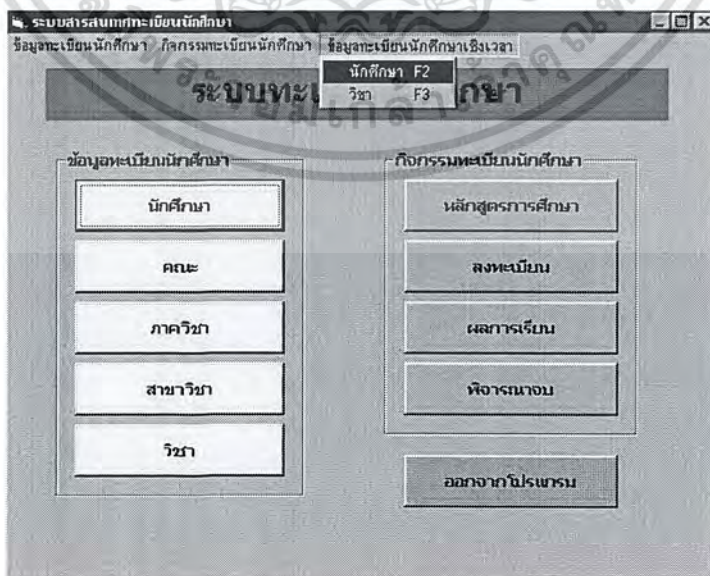
รูปที่ ค-29 แสดงการเตือนรหัสข้อมูลที่ทำกรแก้ไขนี้มีอยู่ในฐานข้อมูลแล้ว

ค.7 ขั้นตอนของการใช้งานข้อมูลทะเบียนนักศึกษาเชิงเวลา

สำหรับงานข้อมูลทะเบียนนักศึกษาเชิงเวลา จะแบ่งเป็น2ส่วนคือส่วนที่เป็นนักศึกษา กับส่วนที่เป็นวิชา โดยเราสามารถที่จะทำการเพิ่ม/แก้ไข/ลบข้อมูลเชิงเวลาได้ ซึ่งมีรายละเอียดดังต่อไปนี้

ค.7.1 ขั้นตอนการใช้งานข้อมูลนักศึกษาเชิงเวลา

1 ให้ทำการคลิกที่เมนูข้อมูลทะเบียนนักศึกษาเชิงเวลา แล้วทำการเลือกนักศึกษาหรือคูปุ่มF2



รูปที่ ค-30 แสดงการเลือกงานข้อมูลนักศึกษาเชิงเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2 จะปรากฏหน้าต่างประวัตินักศึกษา

รูปที่ ก-31 แสดงหน้าต่างประวัตินักศึกษา

- 3 จากนั้นให้ทำการโหลดข้อมูลนักศึกษาที่ต้องการเช่น ต้องการรหัส40010425 ก็ให้ทำการพิมพ์รหัสนักศึกษาในเท็กบ็อกซ์ของรหัสนักศึกษา จากนั้นให้ทำการกดแทป โดยโปรแกรมจะทำการหาข้อมูลที่เป็นปัจจุบันในกรณีที่เราไม่ได้ทำการกำหนดเวลา

รูปที่ ก-32 แสดงการใส่รหัสนักศึกษาที่ต้องการโหลด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4 จากนั้นถ้าปัจจุบันในฐานะข้อมูลมีข้อมูลของรหัสนักศึกษาที่ได้ทำการใส่ค่าเข้าไป โปรแกรมก็จะทำการโหลดข้อมูลของนักศึกษาคนนั้น

| รหัสนักศึกษา | อายุวันที่ | ถึงวันที่  |
|--------------|------------|------------|
| 40010425     | 03/26/2001 | 12/31/9999 |

รูปที่ ค-33 แสดงการโหลดข้อมูลรหัสนักศึกษาที่ต้องการโหลด

- 5 เราสามารถที่จะดูประวัติข้อมูลของนักศึกษาได้โดยการเลือกประวัติที่เราต้องการ

| ชื่อ   | อายุวันที่ | ถึงวันที่  |
|--------|------------|------------|
| ประเมษ | 03/26/2001 | 12/31/9999 |

รูปที่ ค-34 แสดงประวัติชื่อภาษาไทยของนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัตินักศึกษา

ประวัติชื่อภาษาอังกฤษ | ประวัตินามสกุลภาษาอังกฤษ |  
ประวัติรหัสนักศึกษา | ประวัติชื่อภาษาไทย | ประวัตินามสกุลภาษาไทย |

| นามสกุล    | จากวันที่  | ถึงวันที่  |
|------------|------------|------------|
| ทศวรรษธรรม | 03/26/2001 | 12/31/9999 |

นามสกุล      จากวันที่      ถึงวันที่

รูปที่ ค-35 แสดงประวัตินามสกุลภาษาไทยของนักศึกษา

ประวัตินักศึกษา

ประวัติรหัสนักศึกษา | ประวัติชื่อภาษาไทย | ประวัตินามสกุลภาษาไทย |  
ประวัติชื่อภาษาอังกฤษ | ประวัตินามสกุลภาษาอังกฤษ |

| ชื่อ     | จากวันที่  | ถึงวันที่  |
|----------|------------|------------|
| Poramate | 03/26/2001 | 12/31/9999 |

ชื่อ      จากวันที่      ถึงวันที่

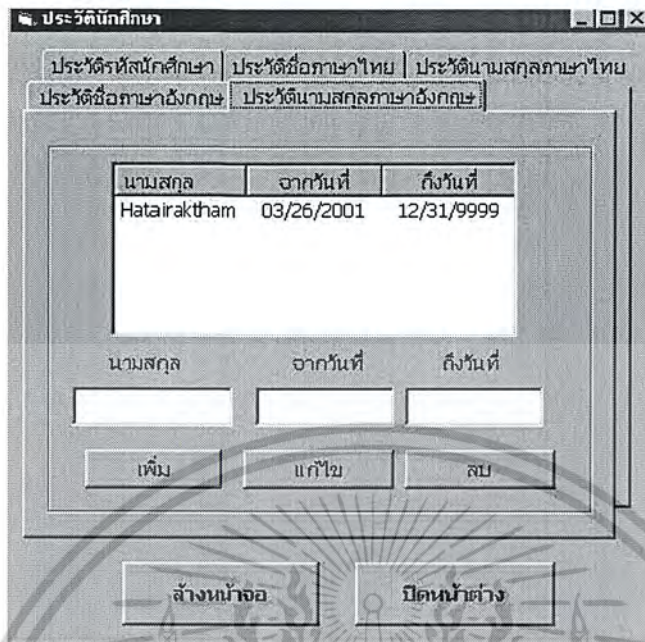
          

รูปที่ ค-36 แสดงประวัติชื่อภาษาอังกฤษของนักศึกษา

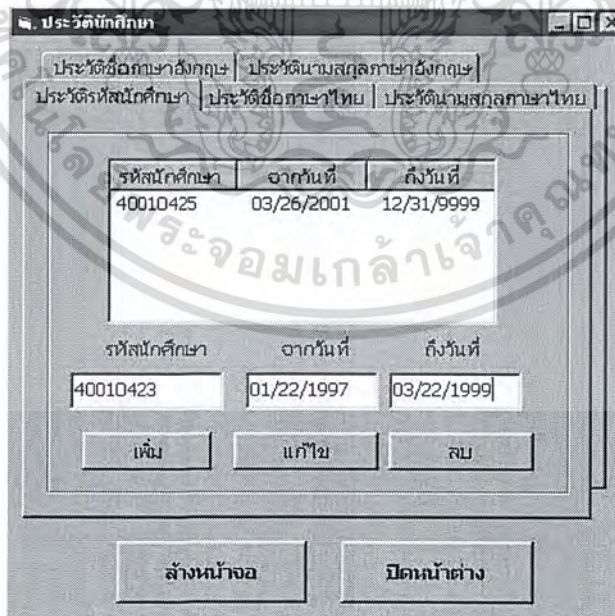
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



| นามสกุล      | จากวันที่  | ถึงวันที่  |
|--------------|------------|------------|
| Hatairaktham | 03/26/2001 | 12/31/9999 |

รูปที่ ค-37 แสดงประวัตินามสกุลภาษาอังกฤษของนักศึกษา

- 6 กรณีที่เราต้องการเพิ่มข้อมูลเชิงเวลาในแบบซีเควันซ์ สามารถทำได้โดยให้เราทำการใส่ข้อมูลรวมไปถึงช่วงเวลาที่ข้อมูลเป็นจริง



| รหัสนักศึกษา | จากวันที่  | ถึงวันที่  |
|--------------|------------|------------|
| 40010425     | 03/26/2001 | 12/31/9999 |

รูปที่ ค-38 แสดงการเพิ่มข้อมูลนักศึกษาเชิงเวลาในแบบซีเควันซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 7 จากนั้นให้เราทำการกดปุ่มเพิ่ม โปรแกรมก็จะทำการเพิ่มประวัติข้อมูลตามช่วงเวลาที่เราได้ทำการกำหนด กรณีที่เราได้ทำการเพิ่มข้อมูลที่มีความซ้ำซ้อนในช่วงเวลาโปรแกรมก็จะไม่ยอมให้ทำการเพิ่มข้อมูล

| รหัสนักศึกษา | จากวันที่  | ถึงวันที่  |
|--------------|------------|------------|
| 40010423     | 01/22/1997 | 03/22/1999 |
| 40010425     | 03/26/2001 | 12/31/9999 |

รูปที่ ค-39 แสดงข้อมูลนักศึกษาเชิงเวลาในแบบซีเคิร์ฟที่ถูกเพิ่มเข้าไปในฐานข้อมูล

- 8 กรณีที่เราต้องการแก้ไขข้อมูลก็สามารถทำได้โดยให้เราใส่ข้อมูลที่เราจะแก้ไขพร้อมทั้งช่วงเวลาที่เราจะทำการแก้ไข จากนั้นให้เราทำการกดปุ่มแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



| รหัสนักศึกษา | จากวันที่  | ถึงวันที่  |
|--------------|------------|------------|
| 40010423     | 01/22/1997 | 03/22/1999 |
| 40010425     | 03/26/2001 | 12/31/9999 |

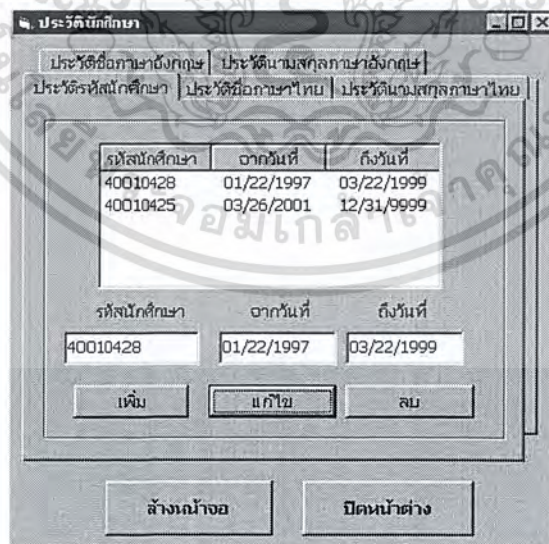
รหัสนักศึกษา: 40010428 | จากวันที่: 01/22/1997 | ถึงวันที่: 03/22/1999

[เพิ่ม] [แก้ไข] [ลบ]

[ล้างหน้าจอ] [ปิดหน้าต่าง]

รูปที่ ค-40 แสดงการแก้ไขข้อมูลนักศึกษาเชิงเวลาในแบบซีเควันซ์

- 9 โปรแกรมก็จะทำการแก้ไขข้อมูลตามที่เราได้กำหนดช่วงเวลา สังกัดจากข้อมูลจะพบว่า โปรแกรมได้ทำการแก้ไขข้อมูลให้เป็นรหัสนักศึกษา40010428 ตามช่วงเวลาที่เราได้ทำการกำหนด



| รหัสนักศึกษา | จากวันที่  | ถึงวันที่  |
|--------------|------------|------------|
| 40010428     | 01/22/1997 | 03/22/1999 |
| 40010425     | 03/26/2001 | 12/31/9999 |
| 40010428     | 01/22/1997 | 03/22/1999 |

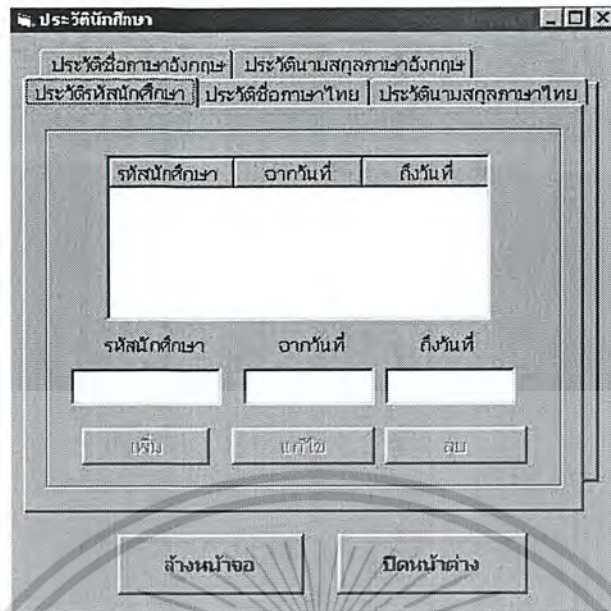
รหัสนักศึกษา: 40010428 | จากวันที่: 01/22/1997 | ถึงวันที่: 03/22/1999

[เพิ่ม] [แก้ไข] [ลบ]

[ล้างหน้าจอ] [ปิดหน้าต่าง]

รูปที่ ค-41 แสดงข้อมูลนักศึกษาเชิงเวลาในแบบซีเควันซ์ที่ถูกแก้ไขเข้าไปในฐานข้อมูล

- 10 เราสามารถทำการรีเซตข้อมูลเพื่อที่จะให้ทำการโหลดข้อมูลมาใหม่ได้ด้วยการกดปุ่มล้างหน้าจอ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



| รหัสนักศึกษา | จากวันที่ | ถึงวันที่ |
|--------------|-----------|-----------|
|              |           |           |

รหัสนักศึกษา      จากวันที่      ถึงวันที่

เพิ่ม      แก้ไข      ลบ

ล้างหน้าจอ      ปิดหน้าต่าง

รูปที่ ค-42 แสดงการรีเซตข้อมูลนักศึกษาเพื่อที่จะให้ทำการโหลดข้อมูลมาใหม่

- 11 จากนั้นเราก็สามารถทำการ โหลดข้อมูลข้อมูลนักศึกษาใหม่ได้
- 12 กรณีการลบข้อมูลเชิงเวลาก็สามารถทำได้โดยให้เราทำการ โหลดข้อมูลนักศึกษามาก่อน จากนั้นให้เราทำการเลือกโดยให้การดับเบิลคลิกเลือกข้อมูลที่จะลบหรือสามารถที่จะใส่เป็นช่วงเวลาที่จะลบก็ได้จากนั้นให้ทำการกดปุ่มลบ



| รหัสนักศึกษา | จากวันที่  | ถึงวันที่  |
|--------------|------------|------------|
| 40010428     | 01/22/1997 | 03/22/1999 |
| 40010425     | 03/26/2001 | 12/31/9999 |

รหัสนักศึกษา      จากวันที่      ถึงวันที่

40010428      01/22/1997      03/22/1999

เพิ่ม      แก้ไข      ลบ

รูปที่ ค-43 แสดงการลบข้อมูลนักศึกษาเชิงเวลาโดยการดับเบิลคลิกเลือกข้อมูลที่จะลบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| รหัสนักศึกษา | จากวันที่  | ถึงวันที่  |
|--------------|------------|------------|
| 40010428     | 01/22/1997 | 03/22/1999 |
| 40010425     | 03/26/2001 | 12/31/9999 |

รูปที่ ค-44 แสดงการลบข้อมูลนักศึกษาเชิงเวลาโดยการใส่เป็นช่วงเวลาที่จะลบ

| รหัสนักศึกษา | จากวันที่  | ถึงวันที่  |
|--------------|------------|------------|
| 40010425     | 03/26/2001 | 12/31/9999 |

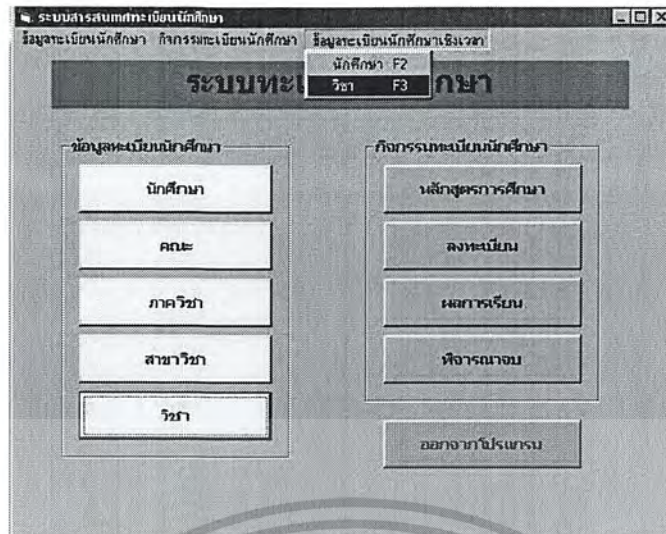
รูปที่ ค-45 แสดงผลของการลบข้อมูลนักศึกษาเชิงเวลา

#### ค.7.2 ขั้นตอนการใช้งานข้อมูลวิชาเชิงเวลา

สำหรับขั้นตอนการเพิ่ม/ลบ/แก้ไขข้อมูลวิชาก็สามารถทำได้คล้ายๆกับการเพิ่ม/ลบ/แก้ไขข้อมูลนักศึกษา โดยมีรายละเอียดดังต่อไปนี้

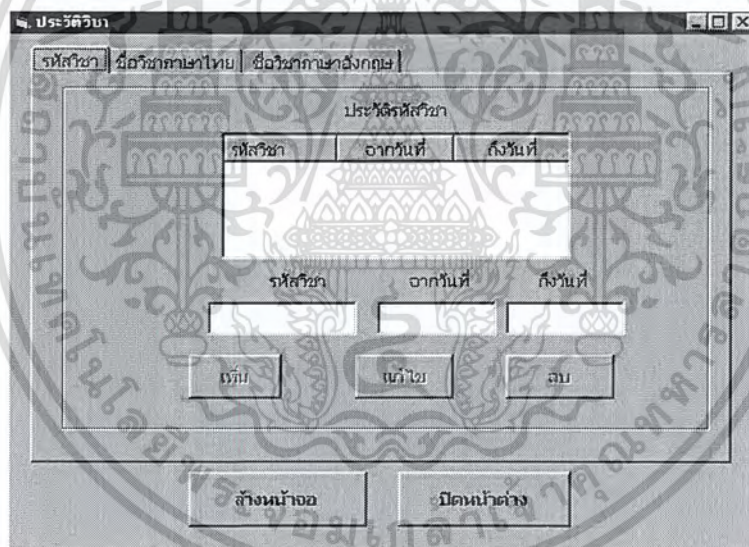
- 1 ให้ทำการคลิกที่เมนูข้อมูลทะเบียนนักศึกษาเชิงเวลาแล้วเลือกวิชาหรือกลุ่มF3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค-46 แสดงการเลือกงานข้อมูลวิชาเชิงเวลา

## 2 จะปรากฏหน้าต่างประวัติวิชา



รูปที่ ค-47 แสดงหน้าต่างประวัติวิชา

- จากนั้นให้ทำการโหลดข้อมูลวิชาที่ต้องการเช่น ต้องการรหัสวิชา01074101 ก็ให้ทำการพิมพ์รหัสวิชาในเท็กบ็อกซ์ของรหัสวิชา จากนั้นให้ทำการกดแทป โดยโปรแกรมจะทำการหาข้อมูลที่เป็นปัจจุบันในกรณีที่เราไม่ได้ทำการกำหนดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ค-48 แสดงการใส่รหัสวิชาวที่ต้องการโหลด

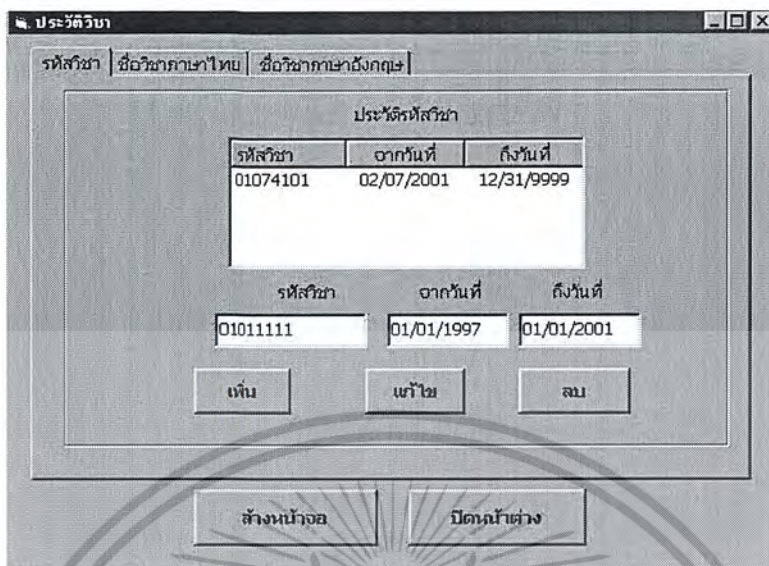
- 4 จากนั้นถ้าปัจจุบันในฐานข้อมูลมีข้อมูลของรหัสวิชาวที่ได้ทำการใส่ค่าเข้าไป โปรแกรมก็จะทำการโหลดข้อมูลของวิชานั้น

รูปที่ ค-49 แสดงการโหลดข้อมูลรหัสวิชาวที่ต้องการโหลด

- 5 เราสามารถที่จะดูประวัติข้อมูลของวิชาได้โดยการเลือกประวัติที่เราต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





| รหัสวิชา | จากวันที่  | ถึงวันที่  |
|----------|------------|------------|
| 01074101 | 02/07/2001 | 12/31/9999 |

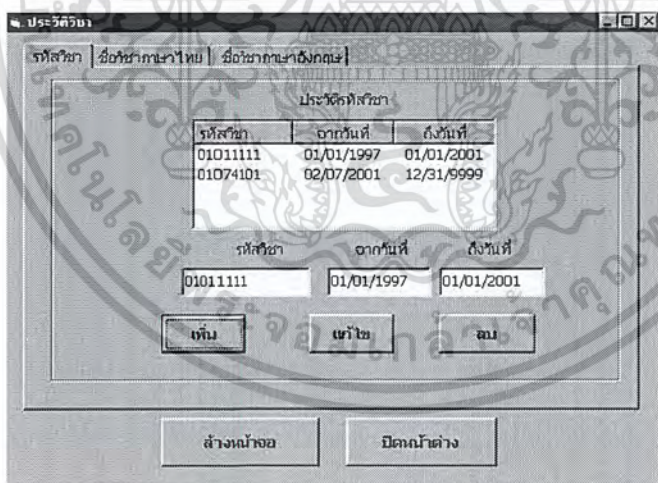
รหัสวิชา: 01011111    จากวันที่: 01/01/1997    ถึงวันที่: 01/01/2001

เพิ่ม    แก้ไข    ลบ

สร้างหน้าจอ    ปิดหน้าจอ

รูปที่ ค-52 แสดงการเพิ่มข้อมูลวิชาเชิงเวลาในแบบซีเควันซ์

- 7 จากนั้นให้เราทำการกดปุ่มเพิ่ม โปรแกรมก็จะทำการเพิ่มประวัติข้อมูลตามช่วงเวลาที่เราได้ทำการกำหนด กรณีที่เราได้ทำการเพิ่มข้อมูลที่มีความซ้ำซ้อนในช่วงเวลาโปรแกรมก็จะไม่ยอมให้ทำการเพิ่มข้อมูล



| รหัสวิชา | จากวันที่  | ถึงวันที่  |
|----------|------------|------------|
| 01011111 | 01/01/1997 | 01/01/2001 |
| 01074101 | 02/07/2001 | 12/31/9999 |

รหัสวิชา: 01011111    จากวันที่: 01/01/1997    ถึงวันที่: 01/01/2001

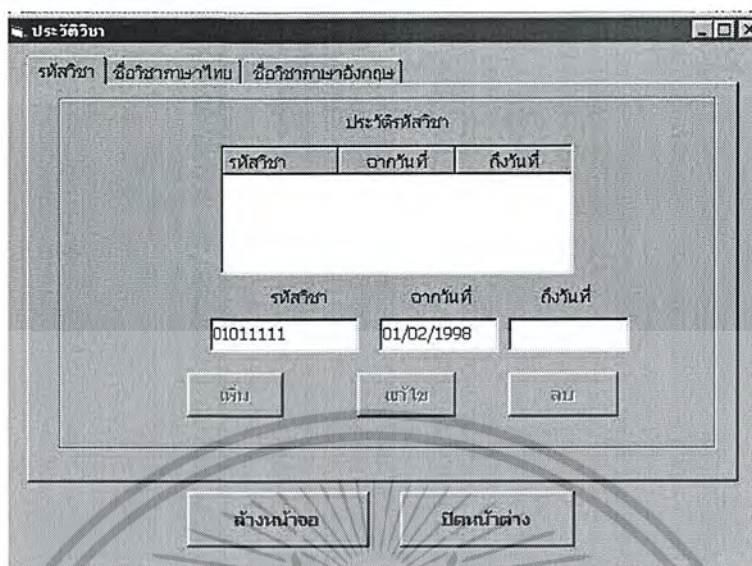
เพิ่ม    แก้ไข    ลบ

สร้างหน้าจอ    ปิดหน้าจอ

รูปที่ ค-53 แสดงข้อมูลวิชาเชิงเวลาในแบบซีเควันซ์ที่ถูกเพิ่มเข้าไปในฐานข้อมูล

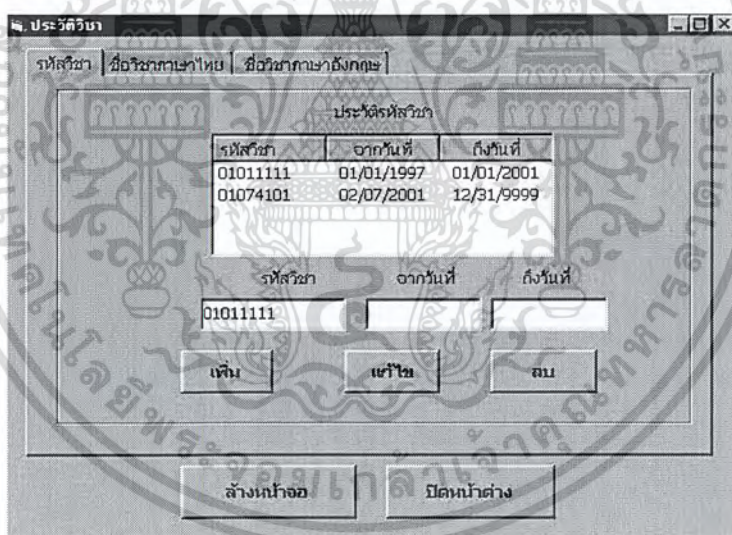
- 8 ในกรณีที่เราต้องการโหลดข้อมูลวิชา โดยการใส่รหัสวิชาที่ไม่ใช่ปัจจุบัน ก็สามารถทำได้ โดยการให้เราใส่ค่าของรหัสวิชา จากนั้นให้ทำการกดปุ่มแท็บ และตรงที่กบ็อกซ์จากวันที่ให้เราทำการใส่วันที่ที่อยู่ในช่วงเวลาที่เราใส่ห้วิชานั้นยังคงจริงอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



| รหัสวิชา | จากวันที่  | ถึงวันที่ |
|----------|------------|-----------|
| 01011111 | 01/02/1998 |           |

รูปที่ ค-54 แสดงโหลดข้อมูลวิชาโดยการใส่รหัสวิชาที่ไม่ใช่ปัจจุบัน แต่ใส่วันที่ที่มันยังคงเป็นจริงแทน



| รหัสวิชา | จากวันที่  | ถึงวันที่  |
|----------|------------|------------|
| 01011111 | 01/01/1997 | 01/01/2001 |
| 01074101 | 02/07/2001 | 12/31/9999 |

รูปที่ ค-55 แสดงข้อมูลประวัติวิชาที่ถูกโหลดขึ้นมา

- 9 กรณีที่เราต้องการแก้ไขข้อมูลก็สามารถทำได้โดยให้เราใส่ข้อมูลที่เราจะแก้ไขพร้อมทั้งช่วงเวลาที่เราจะทำการแก้ไข จากนั้นให้เราทำการกดปุ่มแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติด่วน

รหัสวิชา | ชื่อวิชาภาษาไทย | ชื่อวิชาภาษาอังกฤษ

ประวัติรหัสวิชา

| รหัสวิชา | จากวันที่  | ถึงวันที่  |
|----------|------------|------------|
| 01011111 | 01/01/1997 | 01/01/2001 |
| 01074101 | 02/07/2001 | 12/31/9999 |

รหัสวิชา      จากวันที่      ถึงวันที่

01011175      01/01/1997      01/01/2001

เพิ่ม      แก้ไข      ลบ

ล้างหน้าจอ      ปิดหน้าต่าง

รูปที่ ค-56 แสดงการแก้ไขข้อมูลวิชาเชิงเวลาในแบบชีเควินซ์

ประวัติด่วน

รหัสวิชา | ชื่อวิชาภาษาไทย | ชื่อวิชาภาษาอังกฤษ

ประวัติรหัสวิชา

| รหัสวิชา | จากวันที่  | ถึงวันที่  |
|----------|------------|------------|
| 01011175 | 01/01/1997 | 01/01/2001 |
| 01074101 | 02/07/2001 | 12/31/9999 |

รหัสวิชา      จากวันที่      ถึงวันที่

01011175      01/01/1997      01/01/2001

เพิ่ม      แก้ไข      ลบ

ล้างหน้าจอ      ปิดหน้าต่าง

รูปที่ ค-57 แสดงข้อมูลวิชาเชิงเวลาในแบบชีเควินซ์ที่ถูกแก้ไขเข้าไปในฐานข้อมูล

10 เราสามารถทำการรีเซตข้อมูลเพื่อที่จะให้ทำการโหลดข้อมูลมาใหม่ได้ด้วยการกดปุ่มล้างหน้าจอ

ประวัติด่วน

รหัสวิชา | ชื่อวิชาภาษาไทย | ชื่อวิชาภาษาอังกฤษ

ประวัติรหัสวิชา

| รหัสวิชา | จากวันที่ | ถึงวันที่ |
|----------|-----------|-----------|
|          |           |           |

รหัสวิชา      จากวันที่      ถึงวันที่

01074101               

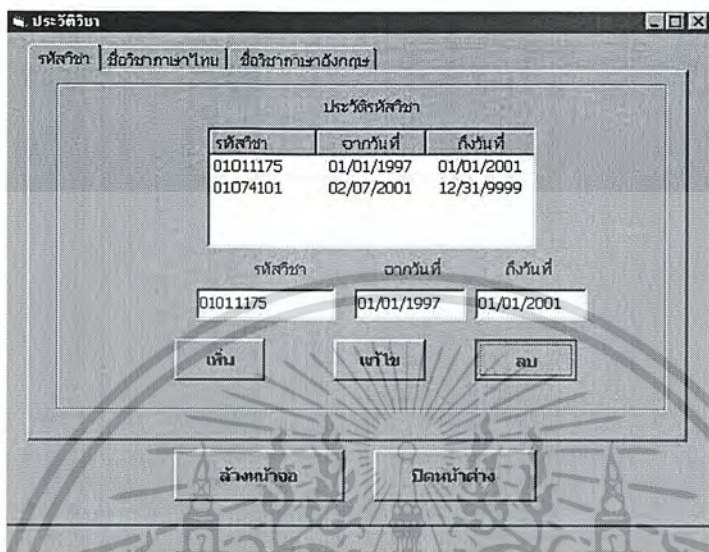
เพิ่ม      แก้ไข      ลบ

ล้างหน้าจอ      ปิดหน้าต่าง

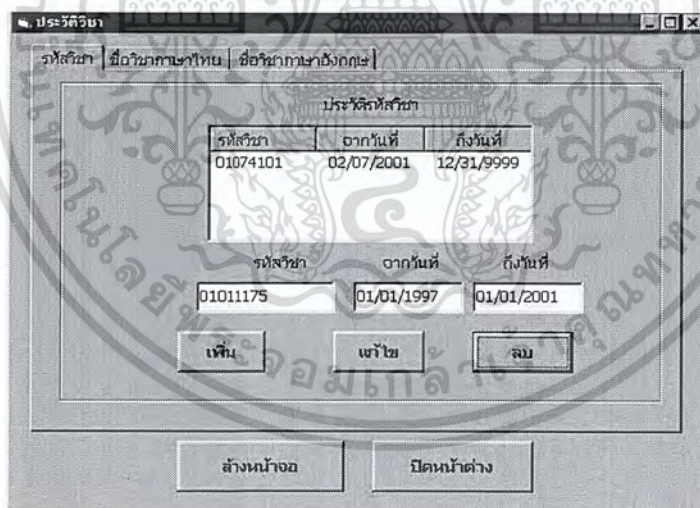
รูปที่ ค-58 แสดงการรีเซตข้อมูลวิชาเพื่อที่จะให้ทำการโหลดข้อมูลมาใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 11 จากนั้นเราก็สามารถทำการ โหลดข้อมูลข้อมูลนักศึกษาใหม่ได้
- 12 กรณีการลบข้อมูลเชิงเวลาที่สามารถทำได้โดยให้เราทำการ โหลดข้อมูลวิชามาก่อน จากนั้นให้เราทำการเลือก โดยให้การดับเบิลคลิกเลือกข้อมูลที่จะลบหรือสามารถที่จะใส่เป็นช่วงเวลาที่ลบก็ได้จากนั้นให้ทำการกดปุ่มลบ



รูปที่ ค-59 แสดงการลบข้อมูลวิชาเชิงเวลาโดยการดับเบิลคลิกเลือกข้อมูลที่จะลบ



รูปที่ ค-60 แสดงการลบข้อมูลวิชาเชิงเวลาโดยการใส่เป็นช่วงเวลาที่ลบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ค.8 กิจกรรมการลงทะเบียน

### 1 จากหน้าจอหลักให้คลิกเลือกที่ ลงทะเบียน

รูปที่ ค-61 แสดงหน้าต่างของการลงทะเบียน

2 ป้อนรหัสนักศึกษาเข้าไป หากในระบบไม่มีข้อมูลของนักศึกษารหัสดังกล่าวก็จะแจ้งให้ทราบว่าไม่พบข้อมูล

3 จากนั้นป้อนปีการศึกษา และ ภาคการศึกษาที่ต้องการลงทะเบียนลงไป หากเป็นการลงทะเบียนครั้งแรกระบบจะแจ้งให้ผู้ใช้ทราบว่า เป็นการลงทะเบียนครั้งแรกของปีการศึกษา และภาคการศึกษานั้น

รูปที่ ค-62 แสดงหน้าต่างของการลงทะเบียนในครั้งแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากมีข้อมูลการลงทะเบียนของนักศึกษาในปีการศึกษาและภาคการศึกษาดังกล่าวอยู่แล้วระบบก็จะแสดงข้อมูลของการลงทะเบียนดังกล่าวออกมาให้

| ลำดับที่ | รหัสวิชา | รายชื่อวิชา                  | หน่วยกิต |
|----------|----------|------------------------------|----------|
| 1        | 01074102 | โครงงาน 2                    | 3        |
| 2        | 01074108 | คอมพิวเตอร์กราฟิก            | 3        |
| 3        | 01074114 | ระบบการจัดการสารสนเทศ        | 3        |
| 4        | 01074119 | การเขียนโปรแกรมระบบเครือข่าย | 3        |
| 5        | 03100004 | การเงินและการธนาคาร          | 2        |
| 6        | 03100054 | การจัดการธุรกิจระหว่างประเทศ | 2        |
| 7        | 01001112 | คณิตศาสตร์ 1                 | 6        |

รูปที่ ค-63 แสดงหน้าต่างบอกรายละเอียดของการลงทะเบียน

สำหรับการเพิ่มรายวิชาที่ทำได้โดยการกดที่ปุ่ม เพิ่มวิชา จะมีหน้าต่างขึ้นมาให้ใส่รหัสวิชาที่ต้องการลงทะเบียน หากรหัสวิชาที่ใส่ไม่มีข้อมูลอยู่ก็จะแสดงข้อความบอกให้ทราบ

| ลำดับที่ | รหัสวิชา | รายชื่อวิชา                         | หน่วยกิต |
|----------|----------|-------------------------------------|----------|
| 1        | 01002001 | คณิตศาสตร์ 3                        | 3        |
| 2        | 01022102 | เครื่องจักรไฟฟ้ากำลังพื้นฐาน        | 3        |
| 3        | 01042101 | อิเล็กทรอนิกส์พื้นฐาน               | 3        |
| 4        | 01050003 | กลศาสตร์ของแข็งและของไหล            | 3        |
| 5        | 01072102 | หลักการเขียนโปรแกรมคอมพิวเตอร์      | 2        |
| 6        | 01072101 | ไมโครโปรเซสเซอร์และไมโครคอมพิวเตอร์ | 3        |
| 7        | 01050004 | วัสดุศาสตร์                         | 3        |
| 8        | 01022101 | การทดลองทางวิศวกรรมสมัยใหม่ 3       | 1        |

รูปที่ ค-64 แสดงตัวอย่างการเพิ่มรายวิชาในการลงทะเบียน

สำหรับการถอนรายวิชาที่ลงทะเบียนไปแล้วออกจากรายการลงทะเบียนก็ให้ทำการดับเบิลคลิกที่รายวิชาที่ต้องการถอนในลิสต์ที่แสดงรายวิชา ก่อน แล้วกดที่ปุ่มถอนรายวิชา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลงทะเบียน

รหัสนักศึกษา 40010425

ชื่อ นาย  ประเมษ สกุล ทิพย์รักษาธรรม

คณะ วิศวกรรมศาสตร์ ภาควิชา วิศวกรรมคอมพิวเตอร์ ปีการศึกษา 2541 เทอม 1

รายการลงทะเบียน

| ลำดับที่ | รหัสวิชา | รายชื่оวิชา                         | หน่วยกิต |
|----------|----------|-------------------------------------|----------|
| 1        | 01002001 | คณิตศาสตร์ 3                        | 3        |
| 2        | 01022102 | เครื่องจักรไฟฟ้ากำลังพื้นฐาน        | 3        |
| 3        | 01042101 | อิเล็กทรอนิกส์พื้นฐาน               | 3        |
| 4        | 01050003 | กลศาสตร์ของแข็งและของไหล            | 3        |
| 5        | 01072102 | หลักการเขียนโปรแกรมคอมพิวเตอร์      | 2        |
| 6        | 01072101 | ไมโครโปรเซสเซอร์และไมโครคอมพิวเตอร์ | 3        |
| 7        | 01050004 | วัสดุศาสตร์                         | 3        |
| 8        | 01022101 | การทดลองทางวิศวกรรมสมัยใหม่ 3       | 1        |

หน่วยกิต 21 หน่วย

เห็นวิชา ถอนวิชา ล้างหน้าจอ ปิดหน้าต่าง

รูปที่ ค-65 แสดงตัวอย่างการถอนรายวิชาในการลงทะเบียน

#### ค.9 กิจกรรมผลการเรียน

- 1 จากหน้าจอหลักให้คลิกเลือกที่ ผลการเรียน
- 2 ป้อนรหัสรหัสนักศึกษาเข้าไป หากในระบบไม่มีข้อมูลของนักศึกษารหัสดังกล่าวก็จะแจ้งให้ทราบว่าจะไม่พบข้อมูล

ผลการเรียน

รหัสนักศึกษา 40010425

ชื่อ นาย  ประเมษ สกุล ทิพย์รักษาธรรม

คณะ วิศวกรรมศาสตร์ ภาควิชา วิศวกรรมคอมพิวเตอร์ ปีการศึกษา 2541 เทอม 1

ผลการเรียน

| ลำดับที่ | รหัสวิชา | รายชื่оวิชา | หน่วยกิต | เกรด |
|----------|----------|-------------|----------|------|
|----------|----------|-------------|----------|------|

หน่วยกิต 00 หน่วย GPS 00.00 หน่วยกิตสะสม 00 หน่วย GPA 00.00

ล้างหน้าจอ ปิดหน้าต่าง

รูปที่ ค-66 แสดงหน้าต่างของการแสดงผลการเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2 จากนั้นป้อนปีการศึกษา และ ภาคการศึกษาที่ต้องการจะใส่ผลการเรียนลงไป หากเป็นหากไม่พบข้อมูลการลงทะเบียนของนักศึกษาในปีการศึกษา และภาคการศึกษาดังกล่าว ระบบจะแจ้งให้ผู้ใช้ทราบว่าไม่พบข้อมูลการลงทะเบียน แต่หากพบก็จะโหลดข้อมูลการลงทะเบียนมาให้

ผลการศึกษา

รหัสนักศึกษา 40010425

ชื่อ นาย ประเมษ สกุล หทัยรักษธรรม

คณะ วิศวกรรมศาสตร์ ภาควิชา วิศวกรรมคอมพิวเตอร์ ปีการศึกษา 2540 เทอม 1

| ลำดับที่ | รหัสวิชา | รายชื้อวิชา             | หน่วยกิต | เกรด |
|----------|----------|-------------------------|----------|------|
| 1        | 01041101 | การวิเคราะห์วงจรไฟฟ้า * | 3        | B+   |
| 2        | 01050001 | กลศาสตร์ *              | 3        | B    |
| 3        | 03150030 | พลศึกษา                 | 2        | C+   |
| 4        | 03150066 | การศึกษาในมหาวิทยาลัย   | 2        | C    |

หน่วยกิต 10 หน่วย GPS 2.85 หน่วยกิตสะสม 10 หน่วย GPA 2.85

ล้างหน้าจอ ปิดหน้าต่าง

รูปที่ ค-67 แสดงรายละเอียดของผลการเรียนที่มีอยู่

ถ้าต้องการใส่ผลการเรียนในรายวิชาใด ก็ดับเบิลคลิกที่รายวิชานั้น ในรายการ จะมีหน้าต่างให้ใส่

ผลการเรียน

ผลการศึกษา

รหัสนักศึกษา 40010425

ชื่อ นาย ประเมษ

คณะ วิศวกรรมศาสตร์

ป้อนเกรดวิชา วิศวกรรม 2

ป้อนเกรด

ตกลง ยกเลิก

เทอม 2

| ลำดับที่ | รหัสวิชา | รายชื้อวิชา                  | หน่วยกิต | เกรด |
|----------|----------|------------------------------|----------|------|
| 1        | 01074102 | โรงงาน 2                     | 3        |      |
| 2        | 01074108 | คอมพิวเตอร์กราฟฟิค           | 3        | A    |
| 3        | 01074114 | ระบบจัดการสารสนเทศ           | 3        | B+   |
| 4        | 01074119 | การเขียนโปรแกรมเชิงเครือข่าย | 3        |      |
| 5        | 03100004 | การเงินและการธนาคาร          | 2        | B+   |
| 6        | 03100054 | การจัดการธุรกิจระหว่างประเทศ | 2        | A    |

หน่วยกิต 10 หน่วย GPS 3.75 หน่วยกิตสะสม 144 หน่วย GPA 3.11

ล้างหน้าจอ ปิดหน้าต่าง

รูปที่ ค-68 แสดงการใส่ผลการศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการลงทะเบียน และการแสดงผลการเรียนนั้น หากรายวิชาที่เคยลงทะเบียนไปแล้วนั้น เกิดการเปลี่ยนแปลง ไม่ว่าจะเป็นรหัสหรือชื่อวิชา ระบบก็จะแสดงชื่อในตอนที่ลงทะเบียนให้ แต่จะแสดง \* เพื่อบอกให้ทราบว่ารายวิชาดังกล่าวได้มีการเปลี่ยนแปลงไปแล้ว ซึ่งทำให้เราทราบว่ารายวิชาดังกล่าว มีการเปลี่ยนแปลง โดยกรที่สามารถทำดังกล่าวได้นั้น เนื่องจากในการลงทะเบียนนั้นจะมีการเก็บเวลา เอาไว้ เพื่อเอาไว้ตรวจสอบกับ ช่วงเวลาของรายวิชาเพื่อจะได้หาข้อมูลของรายวิชาที่ตรงกันสำหรับช่วงเวลาเดียวกัน

## ค.10 การพิจารณาจบ

- 1 จากหน้าจอหลักให้คลิกเลือกที่ พิจารณาจบ
- 2 ป้อนรหัสรหัสนักศึกษาเข้าไป หากในระบบไม่มีข้อมูลของนักศึกษารหัสดังกล่าวก็จะแจ้งให้ทราบว่าไม่พบข้อมูล กดปุ่มประมวลผล ระบบจะทำการประมวลผลการศึกษาให้

รหัสนักศึกษา 40010425

ชื่อ นาย ประเมษ สกุล หทัยกระธรรม ตัวอย่างผลการศึกษา

คณะ วิศวกรรมศาสตร์ ภาควิชา วิศวกรรมคอมพิวเตอร์

สาขาวิชา วิศวกรรมคอมพิวเตอร์ หลักสูตร ปริญญาตรี วิศวกรรมศาสตรบัณฑิต 4 ปี

ประมวลผล

วิชาบังคับเลือกที่เรียน  หน่วยกิต วิชาบังคับเลือกที่ผ่าน  หน่วยกิต วิชาบังคับเลือกที่เหลือ  หน่วยกิต

วิชาเลือกเสรีที่เรียน  หน่วยกิต วิชาเลือกเสรีที่ผ่าน  หน่วยกิต วิชาเลือกเสรีที่เหลือ  หน่วยกิต

รายชื่อวิชาที่ไม่ผ่าน/เกรดไม่สมบูรณ์

| ปีการศึกษา | เทอม | รหัสวิชา | ชื่อวิชา                     | หน่วยกิต | เกรด |
|------------|------|----------|------------------------------|----------|------|
| 2543       | 2    | 01074102 | โครงการ 2                    | 3        |      |
| 2543       | 2    | 01074119 | การเขียนโปรแกรมเชิงเครือข่าย | 3        |      |

สถาน-การ-การ-ศึกษา

สมบูรณ์

ไม่สมบูรณ์

รายชื่อวิชาบังคับที่เรียนไม่ครบ

| รหัสวิชา | ชื่อวิชา | หน่วยกิต | ชั้นปีที่ | เทอม |
|----------|----------|----------|-----------|------|
|          |          |          |           |      |

เปอร์เซ็นต์การประมวลผล 0%  100%

สร้างหน้าจอ

ปิดหน้าต่าง

รูปที่ ค-69 แสดงการประมวลผลสำหรับการพิจารณาจบ

ระบบจะทำการตรวจสอบว่าวิชาต่างๆ ที่นักศึกษาเรียนนั้น มีวิชาใดบ้างที่ผลการเรียนไม่ผ่านหรือไม่สมบูรณ์เช่น ผลการเรียนได้ F หรือผลการเรียนได้ U หรือว่าผลการเรียนยังไม่สมบูรณ์ หากพบก็จะแสดงรายวิชาดังกล่าวแจ้งให้ทราบ และสำหรับวิชาบังคับก็จะทำการตรวจสอบกับหลักสูตรที่นักศึกษาเรียนว่า เรียนครบถ้วนหรือเปล่า หากเรียนไม่ครบ ก็จะแสดงรายชื่อของวิชาดังกล่าวให้ทราบ ส่วนสำหรับวิชาเลือกก็จะตรวจสอบกับหน่วยกิจของหลักสูตรว่า ต้องผ่านวิชาเลือกกี่หน่วยกิตเปรียบเทียบกับที่นักศึกษานี้เป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ศึกษาเรียนว่า ครบหรือไม่ หากตรวจสอบแล้วผลการเรียนสมบูรณ์ ระบบจะแสดงผลการตรวจสอบโดยรวมให้ทราบโดยสังเขตได้จากปุ่มออพชั่น

สำหรับรายวิชาที่เรียนไม่ครบนั้น จะแสดงรายละเอียดของวิชาดังกล่าวในสถานะปัจจุบัน เพื่อบอกให้ทราบว่าปัจจุบันวิชาดังกล่าว มีรหัสและชื่อวิชาปัจจุบันที่ใช้อยู่คืออะไร ซึ่งอาจจะแตกต่างกับชื่อในอดีตที่นักศึกษาต้องลง ทำให้นักศึกษาสามารถลงทะเบียนเรียนวิชาดังกล่าวได้ในปัจจุบัน

3 หากต้องการดูตัวอย่างผลการศึกษานักศึกษาก็กดที่ปุ่ม ตัวอย่างผลการศึกษามีตัวอย่างรายงานสรุปการศึกษาทั้งหมดของนักศึกษาคนนั้น

**ตัวอย่างผลการศึกษา**

Record No. 40010425  
 Name Poramate Hathairaktum  
 Date of Birth 11/03/1979  
 Date of Admission 01/01/1997  
 Degree Bachelor Of Engineering in Computer Engineering  
 Date of Graduation  
 Major Computer Engineering

| CODE                             | CREDIT | GRADE      | COURSE TITLE                       |
|----------------------------------|--------|------------|------------------------------------|
| ปีการศึกษา 2540 ภาคการศึกษาที่ 1 |        |            |                                    |
| 01001002                         | 6      | A          | Mathematics I                      |
| 01021101*                        | 1      | B+         | Engineering Laboratory I           |
| 01041101                         | 3      | B+         | Electrical Circuit Analysis*       |
| 03010020                         | 3      | C+         | English for Engineering I          |
| 03150030*                        | 2      | C+         | Introduction to Physical Education |
| 03150066                         | 2      | C          | University Study                   |
| 01050001                         | 3      | B          | Mechanics*                         |
| GPS : 3.17                       |        | GPA : 3.17 |                                    |
| ปีการศึกษา 2540 ภาคการศึกษาที่ 2 |        |            |                                    |
| 01001003                         | 3      | A          | Mathematic II                      |
| 01010001                         | 3      | B+         | Electromagnetics                   |
| 01021102                         | 1      | B          | Engineering Laboratory II          |
| 01040001*                        | 2      | B+         | Quantum Physics*                   |
| 01050002                         | 3      | B+         | Thermodynamics                     |
| 01051101                         | 1      | C+         | Engineering Drawing                |
| 01071101*                        | 2      | C+         | Introduction to Computer Science*  |
| 01071102                         | 3      | C+         | Digital Circuit and Logic Design   |
| 03010021                         | 3      | D+         | English for Engineering            |
| GPS : 2.97                       |        | GPA : 3.07 |                                    |

รูปที่ ค-70 แสดงตัวอย่างของรายงานสรุปผลการศึกษา

ตัวอย่างของใบแสดงผลการเรียนจะแสดงรายละเอียดของวิชาต่างๆ ที่นักศึกษาคนนั้นเคยเรียน โดยแสดงข้อมูลของวิชาต่างๆ รหัสวิชา ชื่อวิชา ที่ตรงกับที่นักศึกษาคนนั้นเคยลงทะเบียนเรียน ถึงแม้ว่ารายวิชาดังกล่าวจะมีการเปลี่ยนแปลงรหัส ชื่อวิชาดังกล่าวไปแล้วก็ได้ โดยวิชาใดที่มีการเปลี่ยนแปลงข้อมูล ก็จะมีการแสดงให้ทราบโดยเครื่องหมาย \* นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้