

ปฏิญญานិพนธ์

ระบบฐานข้อมูลหน่วยบริการวิชาการของคณะครุศาสตร์อุตสาหกรรม
ACADEMIC SERVICE DATABASE SYSTEM FOR FACULTY OF
INDUSTRIAL EDUCATION



นายยุทธวิช ชูสวน
นางสาวอรไท ก้อนมณี
นายอุทัย สำรวมจิตร

ปฏิญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์
ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เลขหมู่.....
เลขทะเบียน 40168
วัน, เดือน, ปี 17 ส.ค. 2544

b. 11091506
1.....

สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาควิชาครุศาสตร์วิศวกรรม
คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาโท

ชื่อหัวข้อ ระบบฐานข้อมูลหน่วยบริการวิชาการของคณะครุศาสตร์อุตสาหกรรม
Academic Service Database System for Faculty of Industrial Education

ชื่อนักศึกษา 1. นายยุทธวิช ชูสวน รหัสประจำตัว 42035344
2. นางสาวอรไท ก้อนมณี รหัสประจำตัว 42035362
3. นายอุทัย สำรวมจิตต์ รหัสประจำตัว 42035364

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชา อิเล็กทรอนิกส์และคอมพิวเตอร์

อาจารย์ที่ปรึกษา อาจารย์สุระชัย พิมพ์สาดี

อาจารย์ที่ปรึกษาร่วม อาจารย์ปิยะ จิตรธรรมมาภิรมย์

คณะกรรมการสอบปริญญาโท	ลายมือชื่อ
1. อาจารย์สุระชัย พิมพ์สาดี	
2. อาจารย์ปิยะ จิตรธรรมมาภิรมย์	
3. อาจารย์สุชิน อาจหาญ	
4. อาจารย์อำพล ทองระอา	
5. อาจารย์ไพบุลย์ พวงวงศ์ตระกูล	

วันเดือนปีที่สอบ วันเสาร์ที่ 9 ธันวาคม พ.ศ. 2543 เวลา 13.00 น.

สถานที่สอบ ห้อง ค.311 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว
ลงนาม.....
(ผศ.วิสุทธิ์ อธิพรธรรม)
หัวหน้าภาควิชาครุศาสตร์วิศวกรรม
วันที่ 12 เดือน 12 พ.ศ. 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง ระบบฐานข้อมูลหน่วยบริการวิชาการของคณะครุศาสตร์อุตสาหกรรม
Academic Service Database System for Faculty of Industrial Education

วัตถุประสงค์

1. เพื่อศึกษาการใช้งาน โปรแกรม Microsoft SQL Server 7.0 และ Visual Basic 6.0
2. เพื่อศึกษาการเขียนโปรแกรมระบบฐานข้อมูลแบบ Client /Server
3. เพื่อออกแบบระบบฐานข้อมูลสำหรับหน่วยงานบริการวิชาการของคณะครุศาสตร์
อุตสาหกรรม
4. เพื่อสร้างโปรแกรมระบบฐานข้อมูลสำหรับหน่วยงานบริการวิชาการของคณะครุศาสตร์
อุตสาหกรรม
5. เพื่อทดสอบการใช้งาน โปรแกรมระบบฐานข้อมูลสำหรับหน่วยงานบริการวิชาการของ
คณะครุศาสตร์อุตสาหกรรม
6. เพื่อนำโปรแกรมระบบฐานข้อมูลสำหรับหน่วยงานบริการวิชาการของคณะครุศาสตร์
อุตสาหกรรมไปใช้งาน

ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถใช้งาน โปรแกรม Microsoft SQL Server 7.0 และ Visual Basic 6.0 ได้
2. สามารถเขียนโปรแกรมระบบฐานข้อมูลแบบ Client /Server ได้
3. ออกแบบระบบฐานข้อมูลสำหรับหน่วยงานบริการวิชาการของคณะครุศาสตร์
อุตสาหกรรมได้
4. ทดสอบการใช้งาน โปรแกรมระบบฐานข้อมูลสำหรับหน่วยงานบริการวิชาการของคณะ
ครุศาสตร์อุตสาหกรรมได้
5. ได้โปรแกรมระบบฐานข้อมูลสำหรับหน่วยงานบริการวิชาการของคณะครุศาสตร์
อุตสาหกรรม
6. นำโปรแกรมระบบฐานข้อมูลสำหรับหน่วยงานบริการวิชาการของคณะครุศาสตร์
อุตสาหกรรมไปใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	ระบบฐานข้อมูลหน่วยบริการวิชาการของ คณะครุศาสตร์อุตสาหกรรม	
นักศึกษา	นายยุทธวิท	ชูสวน
	นางสาวอรไท	ก้อนมณี
	นายอุทัย	ตำรวจจิตร
อาจารย์ที่ปรึกษา	อาจารย์สุระชัย	พิมพ์สาตี
อาจารย์ที่ปรึกษาร่วม	อาจารย์ปิยะ	จิตรธรรมมาภิรมย์
หลักสูตร	ครุศาสตร์อุตสาหกรรมบัณฑิต	
สาขาวิชา	อิเล็กทรอนิกส์และคอมพิวเตอร์	
ปีการศึกษา	2543	

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอการออกแบบการเขียนโปรแกรมฐานข้อมูลหน่วยบริการวิชาการของคณะครุศาสตร์อุตสาหกรรม ในส่วนของการจัดการตารางสอน สามารถเพิ่มเติม, แก้ไข และค้นหาข้อมูลได้ โดยออกแบบและพัฒนาขึ้นเพื่อใช้เป็นฐานข้อมูลของตารางเรียนตารางสอน ซึ่งช่วยให้อาจารย์หรือนักศึกษาสามารถสืบค้นข้อมูลได้สะดวกยิ่งขึ้น สามารถเลือกแสดงรายการได้หลายรูปแบบ เช่น ตารางสอนของอาจารย์แต่ละท่าน หรือเลือกวิชาที่ต้องการได้ นอกจากนี้ยังสามารถนำฐานข้อมูล ตารางเรียน ตารางสอนนี้ไปใช้พัฒนางานในระบบอื่นๆ เช่น การจัดทำใบเบิกค่าสอน หรือ จัดทำใบภาระงานของอาจารย์ได้

Thesis Title	Academic Service Database System for Faculty of Industrial Education	
Students	Mr. Yuthawit	Chusaun
	Miss Oratai	Konmanee
	Mr. Uthai	Sumruamchit
Advisor	Mr. Surachai	Pimsalee
Co-Advisor	Mr. Piya	Jitthammapirom
Education Level	Bachelor of Science in Industrial Education	
Program in	Electronic and Computer	
Academic Year	2000	

ABSTRACT

This thesis presents the designing of an Academic Database program for Industrial Education. The program is designed to help teachers and students search for information more easily. It includes features for adding, changing, and searching for information. This database can show information in many patterns; for example, students can see the schedule of each teacher or choose the subjects they want. Besides, this database can be used for developing work in other areas.

กิตติกรรมประกาศ

ปริยญาพันธบัตรฉบับนี้ล่วงไปด้วยดี เนื่องมาจากความร่วมมือของสมาชิกภายในกลุ่มทุกท่าน คณะผู้จัดทำขอขอบคุณท่านอาจารย์ที่ปรึกษารวมถึงคณาจารย์ภาควิชาครุศาสตร์วิศวกรรมทุกท่าน ที่ให้ความกรุณาให้คำปรึกษา และข้อเสนอแนะพร้อมทั้งแนวทางแก้ไขปัญหาในการดำเนินงาน นอกจากนั้นต้องขอขอบคุณเพื่อนๆ ทุกคนที่ให้ความช่วยเหลือ และคอยให้กำลังใจตลอดมา และสุดท้ายความดีทั้งหมดของปริยญาพันธบัตรฉบับนี้ขอมอบให้แก่ บุคคลที่สำคัญที่สุด ซึ่งทำให้ได้รับการศึกษามาตั้งแต่อดีตจนถึงปัจจุบัน คือ บิดามารดา ซึ่งในโอกาสนี้คณะผู้จัดทำรู้สึกซาบซึ้งในพระคุณของทุกท่าน คณะผู้จัดทำขอขอบพระคุณเป็นอย่างยิ่งมา ณ โอกาสนี้ด้วย



สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญรูป	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 ชัดความสามารถของโครงการ	1
1.3 เนื้อหาโดยสังเขป	2
บทที่ 2 ทฤษฎี และหลักการ	3
2.1 ระบบเพิ่มข้อมูล	3
2.2 ระบบฐานข้อมูล	4
2.3 องค์ประกอบของระบบฐานข้อมูล	5
2.3.1 ข้อมูล	5
2.3.2 ฮาร์ดแวร์	6
2.3.3 ซอฟต์แวร์	6
2.3.4 ผู้ใช้ระบบฐานข้อมูล	7
2.3.5 Database Management System (DBMS)	7
2.3.6 คีย์	7
2.3.7 ความสัมพันธ์	8
2.4 ขั้นตอนการออกแบบฐานข้อมูล	11
2.5 การทำ Normalization	11
2.6 การออกแบบฐานข้อมูลเชิงสัมพันธ์	13
2.6.1 ฐานข้อมูลเชิงสัมพันธ์	13
2.6.2 ส่วนประกอบของภาษาฐานข้อมูล	14
2.6.3 ส่วนประกอบพื้นฐานของฐานข้อมูลเชิงสัมพันธ์	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.7 Microsoft SQL Server	16
2.7.1 การรวมเข้ากับ Windows NT Server	17
2.8 การใช้งาน MS-SQL	19
2.8.1 การออกแบบฐานข้อมูล	23
2.8.2 การสร้างตาราง	24
2.8.3 การออกแบบโครงสร้างของตาราง	25
2.8.4 ภาษา SQL (Structured Query Language)	25
2.8.5 การเลือกดูข้อมูลด้วยคำสั่ง SELECT	26
2.8.6 การใช้ AND และ OR ในภาษา SQL	26
2.8.7 การใช้เงื่อนไข NOT ในภาษา SQL	27
2.8.8 การใช้เงื่อนไข IN ในภาษา SQL	27
2.8.9 การใช้เงื่อนไข BETWEEN ในภาษา SQL	27
2.8.10 คำสั่งในการค้นหาข้อมูลด้วย LIKE NOT และ LIKE	28
2.9 การทำงานแบบ Client /Server	28
2.9.1 องค์ประกอบของการทำงานแบบ Client / Server	29
2.9.2 การเชื่อมต่อระหว่าง Client กับ Server ฐานข้อมูล	29
2.9.3 ระบบ Client /Server กับฐานข้อมูล	30
2.9.4 การเขียนโปรแกรมกับระบบ Client /Server	31
2.9.5 ระบบ Client /Server ที่สนับสนุนการประมวลผลแบบกระจาย	32
บทที่ 3 การออกแบบ การสร้าง และการทำงาน	35
3.1 กล่าวนำ	35
3.2 กำหนดโครงสร้างของระบบฐานข้อมูล	35
3.2.1 แนะนำเข้าสู่ระบบฐานข้อมูล	35
3.2.2 การจัดการระบบฐานข้อมูล	36
3.2.3 การสร้างฐานข้อมูล	36
3.2.4 การออกแบบฐานข้อมูล	40
3.2.5 การสร้างตาราง	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

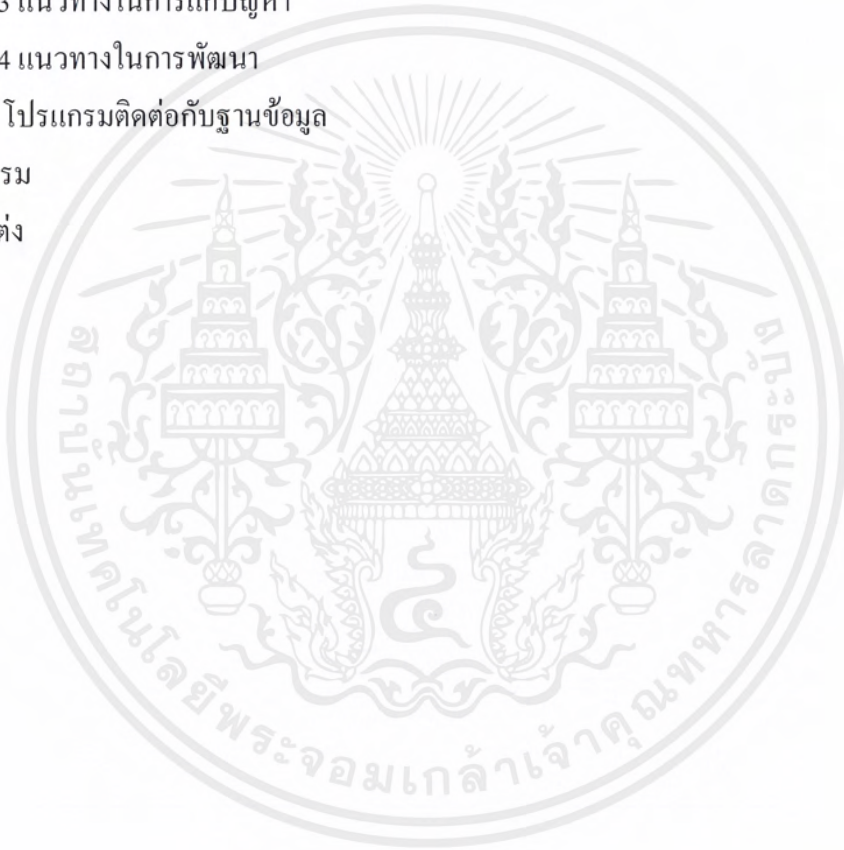
สารบัญ (ต่อ)

เรื่อง	หน้า
3.2.6 การออกแบบ โครงสร้างของตาราง	41
3.3 การกำหนดโครงสร้างและรายการภายในโปรแกรม	43
3.4 ข้อมูลระบบงาน	43
3.4.1 การใช้ SQL Enterprise Manager สร้างตาราง	47
3.5 ความสัมพันธ์ของเพิ่มข้อมูล	49
3.6 การออกแบบโปรแกรมด้วย Visual Basic	50
3.6.1 การทำงานของ Data Environment	50
3.6.2 การสร้างความสัมพันธ์ระหว่าง Object Command	58
3.6.3 การแสดงข้อมูลจาก Data Environment	64
3.7 การจัดการบริหารยูสเซอร์	67
3.7.1 System Administrator หรือ SA	67
3.7.2 การกำหนดความสามารถในการใช้งานแต่ละฐานข้อมูล	68
3.7.3 ขั้นตอนการสร้าง Guest User	69
3.7.4 การกำหนดบทบาทต่างๆ ของ User หรือ Roles	69
3.8 การรักษาความปลอดภัยให้ข้อมูล	71
3.8.1 Password ของ User System Administrator หรือ SA	71
3.8.2 กลไกในการรักษาความปลอดภัยฐานข้อมูล	72
3.8.3 รูปแบบของ Permission	72
บทที่ 4 การทดลอง และผลการทดลอง	74
4.1 การเข้าสู่โปรแกรม	74
4.1.1 เข้าสู่โปรแกรม Microsoft Visual Basic 6.0	74
4.2 การใช้งานโปรแกรม	75
4.2.1 ข้อมูลอาจารย์	75
4.2.2 ข้อมูลรายวิชา	77
4.2.3 ข้อมูลนักศึกษา	79
4.2.4 แก้ไขจากตาราง	82
4.2.5 เพิ่มเติม / แก้ไข	83

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 5 สรุปัญหาแนวทางแก้ไข และพัฒนา	87
5.1 สรุปผลการทดลอง	87
5.2 ปัญหาในการทำงาน	87
5.3 แนวทางในการแก้ปัญหา	87
5.4 แนวทางในการพัฒนา	88
ภาคผนวก โปรแกรมติดต่อกับฐานข้อมูล	89
บรรณานุกรม	123
ประวัติผู้แต่ง	124



สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 แสดงการเปรียบเทียบศัพท์ระหว่างฐานข้อมูล	17
ตารางที่ 2.2 รายชื่อ และนามสกุลของพนักงานที่ได้	20
ตารางที่ 2.3 รายชื่อ และนามสกุลของพนักงานในแผนกคอมพิวเตอร์	20
ตารางที่ 2.4 รายชื่อ และนามสกุลของพนักงานเรียงตามชื่อแผนก	21
ตารางที่ 2.5 รายชื่อ และนามสกุลของพนักงานจากตาราง employee	22
ตารางที่ 2.6 ตัวอย่างโครงสร้างของตาราง Customers	25
ตารางที่ 3.1 พจนานุกรมข้อมูล	44
ตารางที่ 3.2 แสดงชนิดข้อมูลของ SQL Server	45
ตารางที่ 3.2 แสดงชนิดข้อมูลของ SQL Server (ต่อ)	46
ตารางที่ 3.3 ข้อมูลในการเชื่อมต่อเพิ่มเติม	53
ตารางที่ 3.4 การกำหนดรายละเอียดต่างๆ ของ Relate to a parent command object	59

สารบัญรูป

รูป	หน้า
รูปที่ 2.1 ฝ่ายการตลาด, คลังสินค้า, พนักงาน ที่มีการพัฒนา โปรแกรมที่ใช้ข้อมูลจากระบบเพิ่มข้อมูล	3
รูปที่ 2.2 ฐานข้อมูลรวมของบริษัทที่มีข้อมูล พนักงาน, สินค้าคงคลัง, พนักงานขาย และลูกค้า	4
รูปที่ 2.3 ตัวอย่างข้อมูลที่จัดเก็บในระบบฐานข้อมูลของโรงพยาบาล	5
รูปที่ 2.4 ตัวอย่างฐานข้อมูลในการจองห้องพักผู้ป่วย	6
รูปที่ 2.5 การติดต่อข้อมูลภายในฐานข้อมูล	7
รูปที่ 2.6 ความสัมพันธ์แบบ One - to - Many	9
รูปที่ 2.7 ความสัมพันธ์แบบ One - to - One	10
รูปที่ 2.8 ขั้นตอนการทำ Normalization	12
รูปที่ 2.9 (ก) การทำงานทั่วไปของระบบฐานข้อมูลเชิงสัมพันธ์	13
รูปที่ 2.9 (ข) การทำงานทั่วไปของระบบฐานข้อมูลเชิงสัมพันธ์	14
รูปที่ 2.10 ความสัมพันธ์ระหว่างตาราง Personal และ Location	16
รูปที่ 2.11 บริการของ SQL Server	19
รูปที่ 2.12 ตัวอย่าง เอนทิตี	23
รูปที่ 2.13 ตัวอย่างแอททริบิวต์	23
รูปที่ 2.14 ตัวอย่างความสัมพันธ์ระหว่างเอนทิตี ลูกค้า, สินค้า และใบสั่งซื้อ	24
รูปที่ 2.15 สถาปัตยกรรม ANSI / SPARC ของระบบฐานข้อมูล	30
รูปที่ 2.16 การประมวลผลแบบกระจาย	32
รูปที่ 2.17 คอมพิวเตอร์ที่ทำหน้าที่เป็น Client	33
รูปที่ 2.18 Entire Database System	34
รูปที่ 3.1 โครงสร้างระบบฐานข้อมูล SQL Server 7.0	36
รูปที่ 3.2 การเลือกข้อมูลเพื่อสร้างฐานข้อมูล	37
รูปที่ 3.3 การสร้างฐานข้อมูล	37
รูปที่ 3.4 การเปลี่ยน part ที่เก็บฐานข้อมูล	38

สารบัญรูป(ต่อ)

รูป	หน้า
รูปที่ 3.5 การกำหนดชื่อ และตำแหน่งที่มีลักษณะคล้ายตามการกำหนด ฐานข้อมูลอย่างอัตโนมัติ	39
รูปที่ 3.6 ฐานข้อมูลที่เพิ่มสร้างปรากฏขึ้น	40
รูปที่ 3.7 ความสัมพันธ์ระหว่างข้อมูลทั้ง 3 ส่วน	42
รูปที่ 3.8 โครงสร้างและเมนูภายใน โปรแกรม	43
รูปที่ 3.9 การเข้าสู่ SQL Server Enterprise Manage	47
รูปที่ 3.10 หน้าต่างของ Design Table	48
รูปที่ 3.11 ชื่อตาราง Friend ใน Object Table	48
รูปที่ 3.12 ความสัมพันธ์ของเพิ่มข้อมูล	49
รูปที่ 3.13 การเพิ่ม Object Connection	50
รูปที่ 3.14 หน้าต่างของเมนู Properties	51
รูปที่ 3.15 การเลือก Microsoft OLE DB Provider for ODBC Drivers เพื่อติดต่อกับฐานข้อมูล	52
รูปที่ 3.16 การกำหนดตำแหน่งและชื่อไฟล์ฐานข้อมูลพร้อมทั้งชื่อรหัสผ่าน	52
รูปที่ 3.17 การกำหนดคุณสมบัติเพิ่มเติมให้กับ Object Connection	54
รูปที่ 3.18 การสร้าง Object Command	55
รูปที่ 3.19 หน้าต่างของเมนู Properties	56
รูปที่ 3.20 การตั้งชื่อ Object Command ที่สร้าง และเลือก Object Connection	56
รูปที่ 3.21 หน้าต่างของ Query Designer	57
รูปที่ 3.22 ส่วนประกอบต่างๆ ของหน้าต่าง Query Designer	58
รูปที่ 3.23 การสร้าง Object Command 2 ออบเจกต์ที่ติดต่อกับตาราง	59
รูปที่ 3.24 ไดอะล็อกซ์ Properties ของ Object Command	60
รูปที่ 3.25 ความสัมพันธ์ที่เรากำหนดระหว่าง Object ทั้งสอง	60
รูปที่ 3.26 หน้าต่าง Properties ของ Object Command	61
รูปที่ 3.27 การเลือกแท็บ Grouping และเช็คในหัวข้อ Group Command Object	62
รูปที่ 3.28 หน้าต่าง Properties ของ Object Command	63
รูปที่ 3.29 การกำหนดรายละเอียดต่างๆของฟิลด์ Aggregate	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูป	หน้า
รูปที่ 3.30 ผลลัพธ์ที่คำนวณค่าข้อมูลใน Object Command	64
รูปที่ 3.31 การสร้าง Form แสดงข้อมูลในฐานข้อมูล	65
รูปที่ 3.32 การสร้าง Project ใหม่ โดยเลือกชนิดของ Project เป็น Standard.Pxe	66
รูปที่ 3.33 ผลลัพธ์การทำงานเป็น Form ที่แสดงฟิลด์ รหัส, ชื่อ	66
รูปที่ 3.34 เมนู Data Grid	67
รูปที่ 3.35 การกำหนด Interlity Mode	68
รูปที่ 3.36 โปรแกรมตารางของ User	69
รูปที่ 3.37 การตั้ง Password ให้กับ SA	71
รูปที่ 4.1 เมนูหลักของโปรแกรม	74
รูปที่ 4.2 เมนูข้อมูลอาจารย์	75
รูปที่ 4.3 ข้อมูลอาจารย์	76
รูปที่ 4.4 เมนูข้อมูลรายวิชา	77
รูปที่ 4.5 ข้อมูลรายวิชา	78
รูปที่ 4.6 เมนูข้อมูลนักศึกษา	79
รูปที่ 4.7 ข้อมูลนักศึกษา	80
รูปที่ 4.8 หน้าต่างของปุ่ม Report	81
รูปที่ 4.9 เมนูแก้ไข	82
รูปที่ 4.10 เมนูเพิ่มเติม	83
รูปที่ 4.11 ปุ่มข้อมูลอาจารย์	83
รูปที่ 4.12 ปุ่มข้อมูลนักศึกษา	84
รูปที่ 4.13 ปุ่มข้อมูลรายวิชา	85
รูปที่ 4.14 ปุ่มตารางสอน	86

บทที่ 1

บทนำ

1.1 ความเป็นมา และความสำคัญของปัญญาประดิษฐ์

การดำรงชีวิตประจำวันของมนุษย์นั้นต้องมีส่วนเกี่ยวข้องกับข้อมูลข่าวสารต่างๆ มากมาย ไม่ว่าจะเป็นการศึกษาหรือแม้แต่ในการดำเนินธุรกิจก็จำเป็นต้องมีการเก็บบันทึกข้อมูลและรวบรวมข่าวสารต่างๆ ไว้ใช้ ดังนั้นย่อมจะเป็นการยากแน่ถ้าเราจะบันทึกข้อมูลต่างๆ ลงบนกระดาษ ยิ่งถ้าข้อมูลดังกล่าวนั้นมีจำนวนมากก็จะทำให้กระดาษมีปริมาณมากเช่นกัน และเมื่อเราต้องการจะค้นหาข้อมูลก็จะสามารถทำได้ยาก ไม่สะดวกและใช้เวลานาน ดังนั้น การจัดเก็บฐานข้อมูลเพื่อการใช้งานอย่างมีประสิทธิภาพสูงสุดจึงเป็นสิ่งจำเป็นอย่างยิ่ง อย่างไรก็ตาม ฐานข้อมูลที่คิดจะต้องอำนวยความสะดวกให้กับผู้ใช้ ทั้งในเรื่องการใช้งาน การจัดเก็บข้อมูลลงไฟล์ การเรียกใช้ข้อมูลได้โดยสะดวกรวดเร็ว จัดข้อมูลที่ไม่ต้องการได้ง่าย

คอมพิวเตอร์จึงได้เข้ามามีบทบาทต่ออาชีพและสายงานต่างๆ อย่างมากมาย ซึ่งคอมพิวเตอร์เอง ก็เป็นส่วนที่จะช่วยให้บุคลากรที่ทำงานอยู่ในสายอาชีพนั้น ได้รับความความสะดวกสบาย รวมทั้งงานในองค์กรนั้นๆ ก็จะได้รับความสะดวก เพราะงานเป็นไปด้วยความรวดเร็ว มีความถูกต้องสูง แทบจะไม่มีผิดพลาดเกิดขึ้นเลย ดังนั้นทางผู้จัดทำจึงได้จัดทำระบบฐานข้อมูลสำหรับงานบริการวิชาการของคณะครุศาสตร์อุตสาหกรรมขึ้น เพื่อให้สามารถเก็บข้อมูลได้ง่าย และรวดเร็วอีกทั้งยังสามารถค้นหาข้อมูลที่ต้องการได้สะดวก และรวดเร็วยิ่งขึ้น

1.2 ขีดความสามารถของโครงการ

โปรแกรมฐานข้อมูลหน่วยบริการวิชาการของคณะครุศาสตร์อุตสาหกรรมมีขีดความสามารถดังนี้

1. สามารถทำงานในลักษณะ Client /Server ได้
2. สามารถใช้งานฐานข้อมูลพร้อมกันหลาย ๆ คนได้
3. สามารถกำหนดสิทธิ์ของผู้ใช้หลาย ๆ คนได้
4. สามารถเพิ่มเติม/แก้ไขข้อมูลตารางสอนเลือกรูปแบบการแสดงผล และค้นหาข้อมูล

ได้สะดวก

5. สามารถพิมพ์เอกสารตารางสอนออกทางเครื่องพิมพ์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปฏิญญาพันธบัตรฉบับนี้แบ่งออกเป็นบทต่างๆ เพื่อสะดวกต่อการศึกษาและทำความเข้าใจ ในแต่ละบทประกอบด้วยเนื้อหาดังต่อไปนี้

บทที่ 2 ทฤษฎีและหลักการ ประกอบด้วยเนื้อหาดังนี้ คือ ระบบเพิ่มข้อมูล, ระบบฐานข้อมูล, องค์ประกอบของระบบฐานข้อมูล, ขั้นตอนการออกแบบฐานข้อมูล, การทำ Normalization, การออกแบบฐานข้อมูลเชิงสัมพันธ์, Microsoft SQL Server, การใช้งาน MS-SQL, และการทำงานแบบ Client / Server

บทที่ 3 การออกแบบการสร้าง และการทำงาน กล่าวถึงเนื้อหาเกี่ยวกับ การกำหนดโครงสร้างของระบบฐานข้อมูล, การกำหนดโครงสร้างและรายการภายในโปรแกรม, การวิเคราะห์ระบบงาน, ความสัมพันธ์ของเพิ่มข้อมูล, การออกแบบโปรแกรมด้วย Visual Basic, การจัดการบริหารยูสเซอร์, และการรักษาความปลอดภัยให้ข้อมูล

บทที่ 4 การทดลองและผลการทดลอง ประกอบด้วย การเข้าสู่โปรแกรม, การใช้งานโปรแกรม และการทดลองการใช้งาน

บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา ขั้นการสรุปผล ในการจัดทำโครงการปัญหาที่เกิดขึ้น และได้เสนอแนวทางแก้ไขปัญหา รวมทั้งแนวทางการพัฒนาให้มีประสิทธิภาพมากยิ่งขึ้น

ภาคผนวก โปรแกรมติดต่อกับฐานข้อมูล

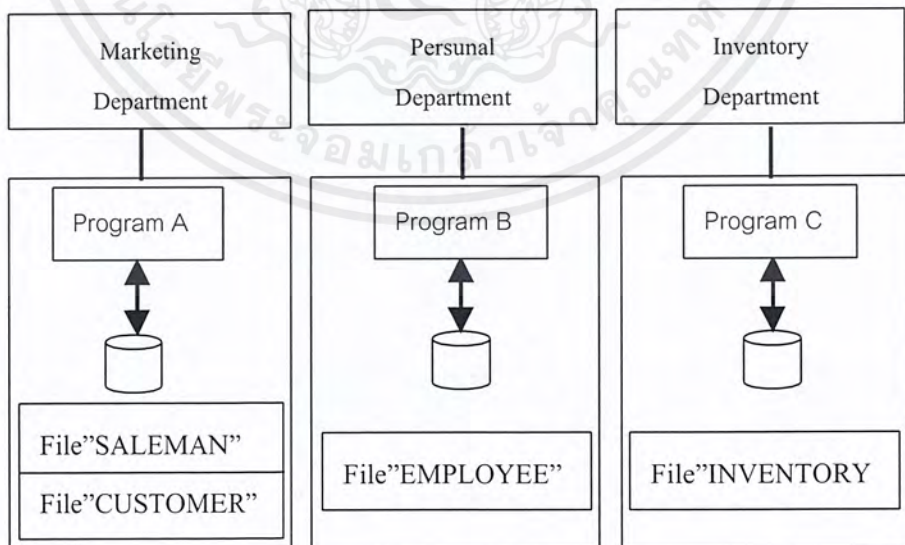
บทที่ 2

ทฤษฎีและหลักการ

2.1 ระบบแฟ้มข้อมูล (File System)

ในอดีต องค์กรต่างๆ มักเก็บข้อมูลต่างไว้ในแฟ้มเอกสาร เมื่อองค์กรมีขนาดใหญ่ขึ้น การเก็บข้อมูลก็ยิ่งมากขึ้นด้วยเช่นกัน การค้นหาข้อมูลจึงต้องใช้เวลาในการค้นหาค่อนข้างมากและเป็น การยากลำบากมากขึ้น การจัดเก็บข้อมูลในคอมพิวเตอร์จึงได้ถูกนำมาใช้แทนการเก็บข้อมูลใน ลักษณะเดิม ในการเก็บข้อมูลเป็นเพียงการเก็บในลักษณะของแฟ้มข้อมูลแทน การเก็บข้อมูลใน ลักษณะนี้ได้แพร่หลายเพิ่มมากขึ้นจึงต้องมีการควบคุมโครงสร้างและการใช้งานอย่างเหมาะสม และรวมแฟ้มข้อมูลนี้เป็นระบบเรียกว่า “ระบบแฟ้มข้อมูล (File System)”

การใช้งานระบบแฟ้มข้อมูลจะต้องอาศัยโปรแกรมเมอร์พัฒนาโปรแกรมเพื่ออ่านข้อมูลจาก แฟ้มข้อมูลต่างๆ ขึ้นมาประมวลผล เพื่อให้ได้ผลลัพธ์ตามที่ผู้ใช้ต้องการ ในยุคเริ่มต้นของการใช้ ระบบฐานข้อมูล แต่ละหน่วยงานในองค์กร จะมีการสร้างระบบแฟ้มข้อมูลขึ้นใช้งานภายในหน่วย งานของตนเอง ซึ่งแต่ละหน่วยงานนั้น จะมีการพัฒนาโปรแกรมที่ใช้กับระบบแฟ้มข้อมูลของตน เองขึ้น เพื่อใช้งานภายในหน่วยงานนั้นๆ เช่น ฝ่ายการพนักงานที่มีการพัฒนาโปรแกรมที่ใช้ข้อมูล จากระบบแฟ้มข้อมูลพนักงานในการคำนวณเงินเดือน ค่าตอบแทนภาษีเงินได้ ฯลฯ ดังรูปที่ 2.1

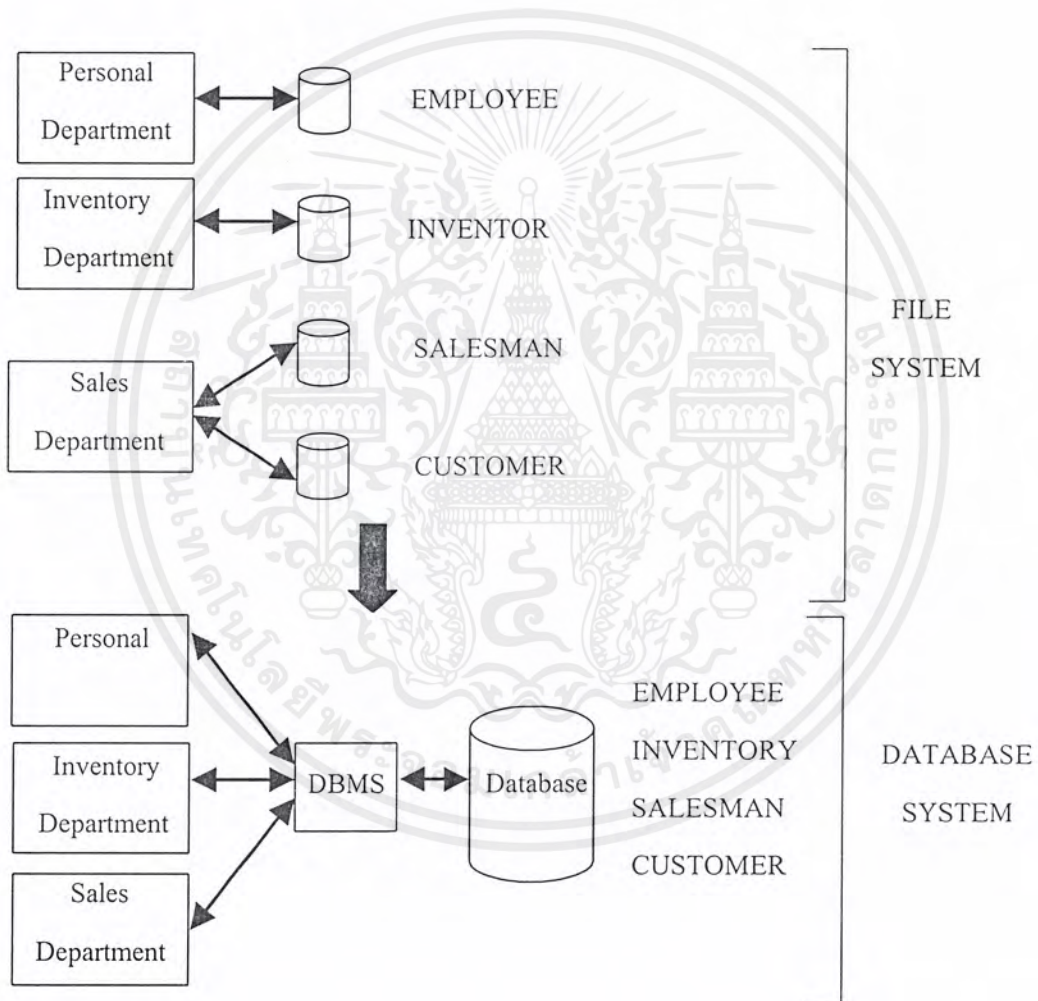


รูปที่ 2.1 ฝ่ายการตลาด, คลังสินค้า, พนักงาน ที่มีการพัฒนาโปรแกรมที่ใช้ข้อมูล จากระบบแฟ้มข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 ระบบฐานข้อมูล (Database System)

ฐานข้อมูล (Database) หมายถึง การนำเอาข้อมูลต่างๆ ที่มีความสัมพันธ์กัน มาจัดเก็บไว้ในที่เดียวกัน เช่น ข้อมูลพนักงาน สินค้าคงคลัง พนักงานขาย และลูกค้า แต่เดิมจะจัดเก็บอยู่ในรูปของแฟ้มข้อมูลของฝ่ายต่างๆ ได้ถูกนำมาจัดเก็บรวมกันภายในฐานข้อมูลเดียว ซึ่งเป็นฐานข้อมูลรวมของบริษัท ส่งผลให้แต่ละฝ่ายสามารถใช้ข้อมูลร่วมกัน และสามารถแก้ปัญหาต่างๆ ที่เกิดขึ้นในระบบแฟ้มข้อมูลได้ ดังรูปที่ 2.2



รูปที่ 2.2 ฐานข้อมูลรวมของบริษัท ที่มีข้อมูลพนักงาน, สินค้าคงคลัง, พนักงานขาย และลูกค้า

ระบบฐานข้อมูล (Database System) หมายถึง ฐานข้อมูลที่จัดทำขึ้นเพื่อสนับสนุนการดำเนินงานอย่างใดอย่างหนึ่ง เช่น ระบบฐานข้อมูลเงินเดือน ซึ่งเป็นฐานข้อมูลที่จัดเก็บข้อมูลต่างๆที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

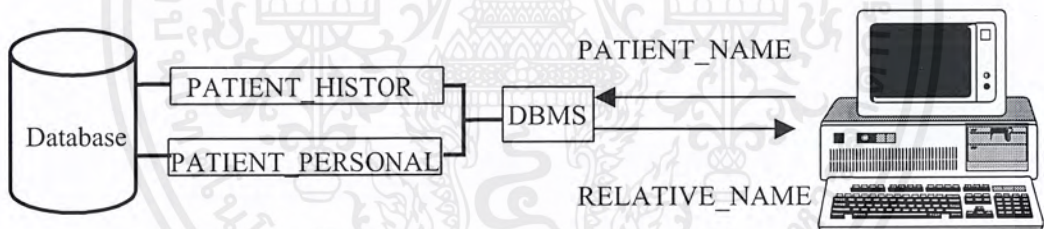
สนับสนุนการคำนวณเงินเดือน หรือระบบฐานข้อมูลประชากร ซึ่งเป็นฐานข้อมูลที่จัดเก็บข้อมูลต่างๆ ที่สนับสนุนการจัดทำสำมะโนประชากร เป็นต้น

2.3 องค์ประกอบของระบบฐานข้อมูล

ระบบฐานข้อมูลโดยทั่วไป แบ่งเป็น 4 ส่วนหลักๆ ดังนี้

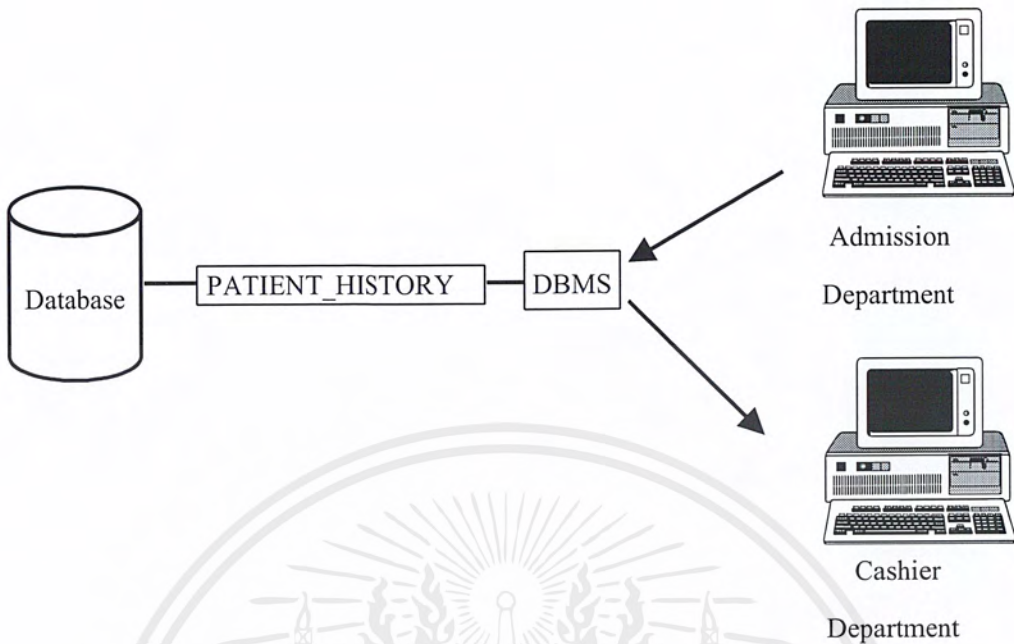
2.3.1 ข้อมูล

ข้อมูลที่จัดเก็บอยู่ในระบบฐานข้อมูล ข้อมูลส่วนใหญ่ในแต่ละส่วนต้องนำมาประกอบกันได้ (Data Integrated) เช่นเมื่อแพทย์รักษาผู้ป่วย แพทย์จะอาศัยข้อมูลจากประวัติการรักษาพยาบาลของผู้ป่วย (PATIENT_HISTORY) มาประกอบการรักษา แต่ในกรณีฉุกเฉินที่ต้องการติดต่อญาติของผู้ป่วย ซึ่งข้อมูลในส่วนนี้ไม่ปรากฏอยู่ในประวัติการรักษาพยาบาลทางโรงพยาบาลสามารถนำชื่อผู้ป่วย (Field "PATIENT_NAME") ไปค้นหาชื่อญาติ(Field "RELATIVE_NAME")ในทะเบียนผู้ป่วย (PATIENT_PERSONAL)ได้ โดยไม่จำเป็นต้องเก็บชื่อญาติของผู้ป่วยไว้ในประวัติการรักษาพยาบาลแต่อย่างใด ดังรูปที่ 2.3



รูปที่ 2.3 ตัวอย่างข้อมูลที่จัดเก็บในระบบฐานข้อมูลของโรงพยาบาล

ในระบบคอมพิวเตอร์ขนาดใหญ่ที่มีผู้ใช้บริการจำนวนมาก ข้อมูลในฐานข้อมูลต้องสามารถใช้ร่วมกัน(DataSharing)จากผู้ใช้หลายคนได้เช่นข้อมูลในการจองห้องพักของผู้ป่วย (PATIENT_ADMIT) จะต้องสามารถนำไปใช้ในการออกแบบใบเสร็จรับเงินเพื่อเก็บค่ารักษาพยาบาลโดยฝ่ายการเงินได้ในขณะเดียวกัน ดังรูปที่ 2.4



รูปที่ 2.4 ตัวอย่างฐานข้อมูลในการจองห้องพักของผู้ป่วย

2.3.2 ฮาร์ดแวร์

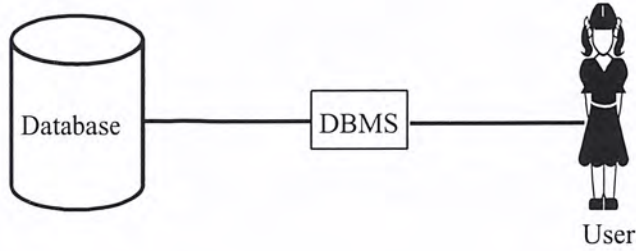
อุปกรณ์คอมพิวเตอร์ที่มีส่วนเกี่ยวข้องกับระบบฐานข้อมูล ประกอบด้วย 2 ส่วนหลักๆ ดังนี้

- 1) หน่วยความจำสำรอง (Secondary Storage) ต้องคำนึงถึงความจุของหน่วยความจำสำรองที่นำมาใช้เก็บข้อมูลของฐานข้อมูล

- 2) หน่วยประมวลผล และหน่วยความจำหลัก เป็นอุปกรณ์ที่ทำงานร่วมกัน เพื่อนำข้อมูลจากฐานข้อมูลมาประมวลผลตามคำสั่งที่กำหนดต้องคำนึงถึงความเร็วของหน่วยประมวลผล และขนาดของหน่วยความจำหลัก

2.3.3 ซอฟต์แวร์

การติดต่อกับข้อมูลภายในฐานข้อมูลของผู้ใช้จะต้องกระทำผ่าน โปรแกรมที่มีชื่อว่า โปรแกรม Database Management System (DBMS) ดังรูปที่ 2.5



รูปที่ 2.5 การติดต่อข้อมูลภายในฐานข้อมูล

หน้าที่หลักของโปรแกรม DBMS ได้แก่ การทำให้การเรียกใช้ข้อมูลจากฐานข้อมูล เป็นอิสระจากส่วนของ Hardware มีหน้าที่ในการจัดการและควบคุมความถูกต้อง ความซ้ำซ้อน และความสัมพันธ์ระหว่างข้อมูลต่างๆ ภายในฐานข้อมูลแทนโปรแกรมเมอร์

2.3.4 ผู้ใช้ระบบฐานข้อมูล

สามารถแบ่งออกเป็น 3 กลุ่มดังนี้

- 1) Application Programmer ได้แก่ ผู้ทำหน้าที่พัฒนาโปรแกรม (Application Program) เพื่อเรียกใช้ข้อมูลจากระบบฐานข้อมูลมาประมวลผล
- 2) End User ได้แก่ ผู้ที่นำข้อมูลจากฐานข้อมูลไปใช้งาน
- 3) Database Administrator (DBA) ได้แก่ ผู้บริหารที่ทำหน้าที่ควบคุมและตัดสินใจในการกำหนดโครงสร้างของฐานข้อมูล ชนิดข้อมูล ชนิดของข้อมูล วิธีการจัดเก็บข้อมูล รูปแบบในการเรียกใช้ข้อมูล ความปลอดภัยของข้อมูลและระเบียบที่ใช้ควบคุมความถูกต้องของข้อมูลภายในฐานข้อมูล

2.3.5 Database Management System (DBMS)

เป็นโปรแกรมที่ทำหน้าที่เป็นตัวกลางการติดต่อระหว่างผู้ใช้กับฐานข้อมูล เพื่อจัดการและควบคุมความถูกต้อง ความซ้ำซ้อน และความสัมพันธ์ระหว่างข้อมูลต่างๆ ภายในฐานข้อมูลแทนโปรแกรมเมอร์ ในการติดต่อกับข้อมูลคำสั่งทุกคำสั่งจะถูกนำมาแปล (Compile) ด้วยโปรแกรม DBMS เป็นการกระทำ (Operation) ต่างๆ ภายใต้คำสั่งเพื่อนำไปกระทำกับตัวข้อมูลภายในฐานข้อมูล

2.3.6 คีย์

คีย์ (key) คือข้อมูลที่จะเป็นกุญแจสำหรับเข้าถึงหรือบ่งชี้ว่าเรคอร์ดใดเป็นเรคอร์ดใด หรือต่างจากเรคอร์ดอื่นๆ อย่างไร คีย์ที่ใช้ในระบบฐานข้อมูลมีดังนี้

1) **Primary key** เป็นคีย์หลักที่ใช้ในการจัดเรียงและแยกแยะข้อมูลแต่ละเรคอร์ดใน Table ออกจากกัน ต้องมีค่าในทุกเรคอร์ด จะปล่อยให้ไม่มีค่า (หรือเรียกว่าเป็น NULL) ไม่ได้ และจะต้องไม่ซ้ำกัน

2) **Candidate key** ในกรณีที่มีหลายฟิลด์ซึ่งสามารถเป็น Primary key ได้ จะเรียกฟิลด์เก่านั้นแต่ละตัวว่าเป็น Candidate key คือสามารถเลือกมาเป็น Primary key ได้เหมือนกัน

3) **Composite key (หรือ Compound key)** เป็นการนำเอาหลายๆ ฟิลด์มารวมกันเป็นคีย์ เพื่อให้มีคุณสมบัติครบถ้วน คือไม่ซ้ำและไม่เป็น Null ใช้ในกรณีที่หากเลือกเพียงฟิลด์ใดฟิลด์หนึ่งมาเป็นคีย์แล้วยังมีโอกาสซ้ำกันหรือเป็น Null ได้

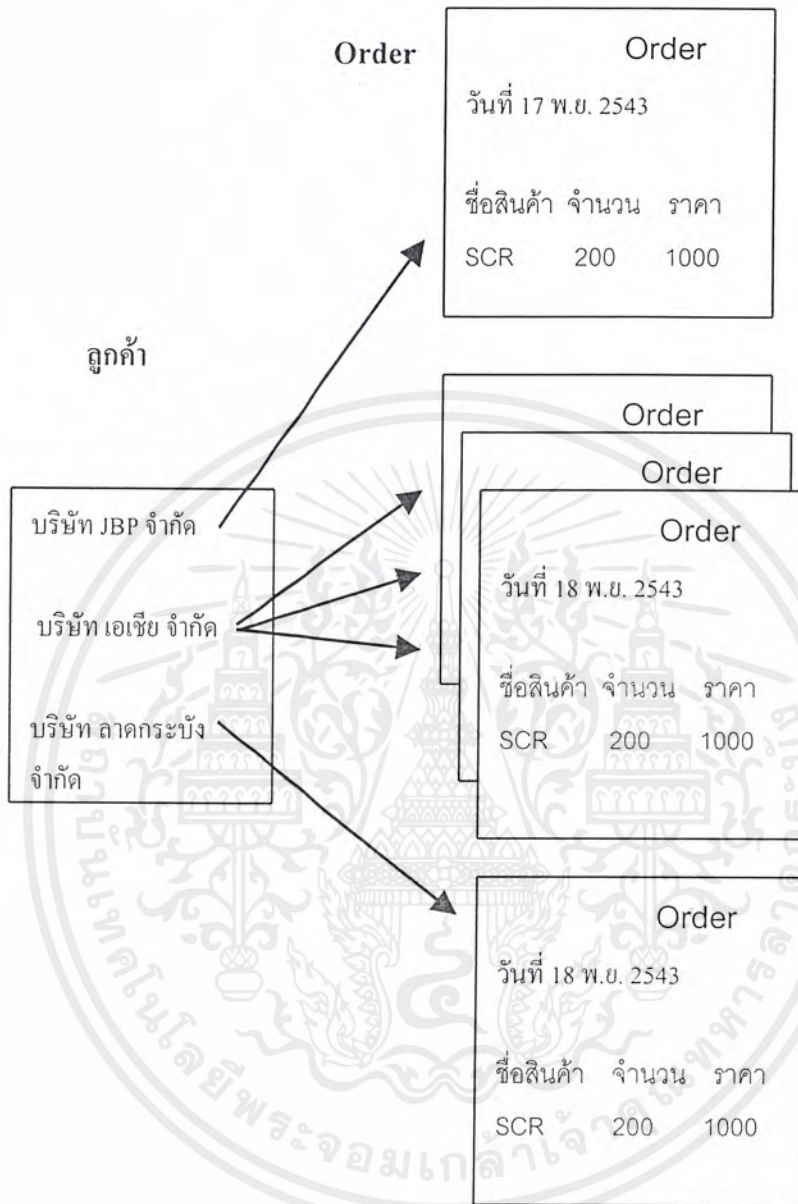
4) **Foreign key** เป็นฟิลด์ที่มีเก็บอยู่ในหลาย Table จึงสามารถใช้เป็นคีย์ในการเชื่อมโยงข้อมูลระหว่างหลาย Table เข้าด้วยกันได้

5) **Secondary key หรือดัชนี (Index)** เป็นคีย์ที่ใช้แยกแยะข้อมูล เนื่องจากมี Primary key ที่ไม่ซ้ำกันอยู่แล้ว Secondary key จึงซ้ำกันได้ ในการเรียกข้อมูลโดยทั่วไปมักจะต้องการจัดเรียงตามลำดับเพื่อความสะดวก

2.3.7 ความสัมพันธ์

เป็นสิ่งที่เชื่อมโยงระหว่างข้อมูลใน Table ต่างๆ เข้าด้วยกัน โดยอาศัยฟิลด์ที่มีค่าตรงกันในระหว่าง Table เป็นตัวเชื่อม แบ่งได้เป็น 3 ลักษณะคือ

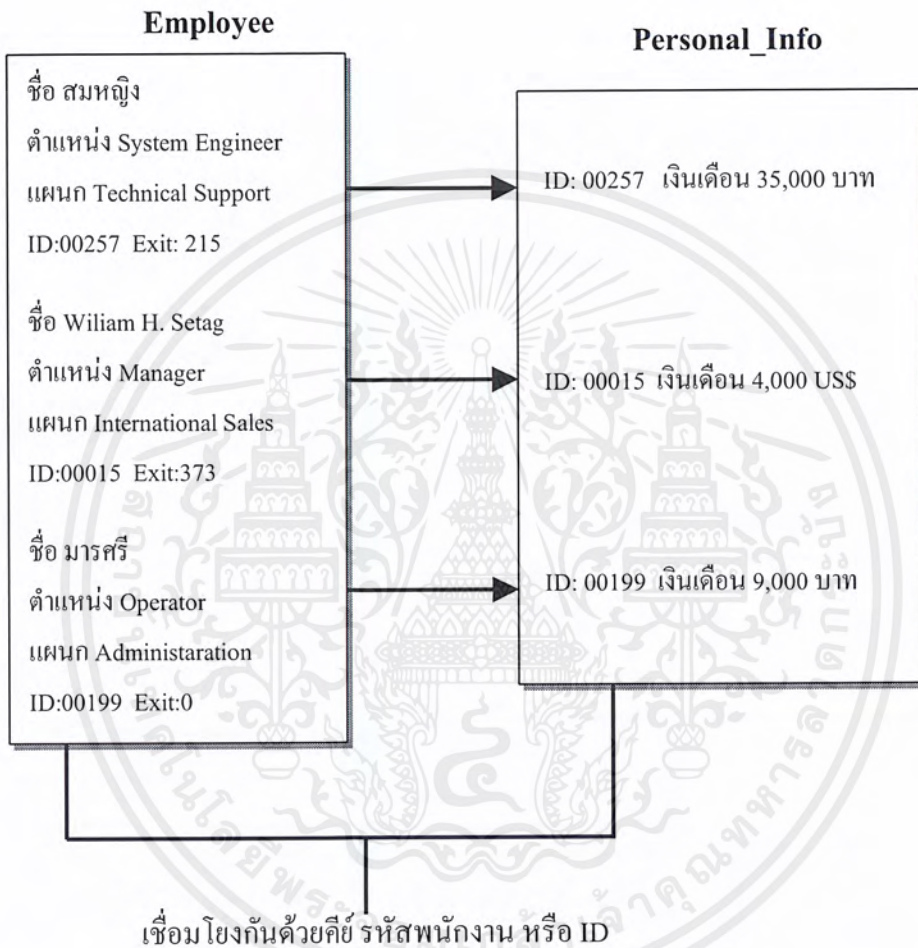
1) **One-To-Many** เป็นความสัมพันธ์แบบที่ฟิลด์ในเรคอร์ดใดๆ ของ Table หนึ่งมีค่าตรงกับฟิลด์ของหลายๆ เรคอร์ดใน Table อื่น เช่น ใน Table Customer มีรหัสลูกค้าซึ่งตรงกับใน Table Order หลายๆ ใบ (เพราะลูกค้าคนหนึ่งสั่งของได้หลายครั้ง) แต่ในทางกลับกัน ในแต่ละ Order จะเป็นของลูกค้าเพียงรายเดียวเท่านั้น คือรหัสลูกค้าในแต่ละ Order จะตรงกับเรคอร์ดเดียวเท่านั้นใน Customer (เพราะ Order 1 ใบย่อมเป็นของลูกค้าเพียงรายเดียว) ดังรูปที่ 2.6



รูปที่ 2.6 ความสัมพันธ์แบบ One-To-Many

2) **One-To-One** เป็นความสัมพันธ์แบบที่ฟิลด์ในเรคอร์ดใดๆ ของ Table หนึ่งมีค่าตรงกับฟิลด์ของเรคอร์ดเพียงเรคอร์ดเดียวใน Table อื่น เช่น ใน Table Employee มีรหัสพนักงานซึ่งตรงกับใน Table Personal_Info ซึ่งอาจรวมเป็น Table เดียวได้ แต่บางครั้งต้องแยกเป็นคนละ Table กัน เพราะ ใน Personal_Info อาจมีข้อมูลขนาดใหญ่ เช่น รูปถ่ายและแผนที่บ้านของพนักงานแต่ละคน ซึ่งบางคนก็ยังไม่ครบหรือไม่อยากให้ ข้อมูลเหล่านี้หากเก็บไว้ใน Table Employee ด้วยก็จะทำให้ Table มีขนาดใหญ่ แต่หากแยกเป็นคนละ Table ก็อาจเรียกใช้ข้อมูลได้เร็วกว่า หรืออาจเป็นเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะเพื่อการรักษาความลับ เช่น Personal_Info มีข้อมูลส่วนตัว เช่น เงินเดือน จึงต้องแยกเป็น
คนละส่วน และควบคุมให้มีเฉพาะบางคนเท่านั้นที่จะเข้าใช้ข้อมูลใน Table นี้ได้ ในขณะที่ทุกคน
สามารถเข้าถึงข้อมูลข้อมูลทั่วๆ ไปใน Employee เช่น ตำแหน่ง หมายเลขโทรศัพท์ภายในได้ ดังรูป
ที่ 2.7



รูปที่ 2.7 ความสัมพันธ์แบบ One-To-One

3) **Many-To-Many** เป็นความสัมพันธ์แบบที่ฟิลด์ในเรคอร์ดใดๆ ของ Table หนึ่งมีค่าตรงกับฟิลด์ของหลายๆ เรคอร์ดใน Table อื่น เช่น ในการเก็บข้อมูลรายชื่อพนักงานขายกับลูกค้า พนักงานขายคนหนึ่งอาจดูแลลูกค้าหลายราย ลูกค้าคนหนึ่งหากสั่งซื้อสินค้าจากหลายแผนกในบริษัท ก็อาจมีการติดต่อกับพนักงานขายหลายคนได้ ซึ่งความสัมพันธ์ในลักษณะนี้สามารถแปลงให้เป็นแบบ One-To-Many ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ขั้นตอนการออกแบบฐานข้อมูล

ขั้นที่ 1 เก็บรวบรวมรายละเอียด

ขั้นตอนนี้เป็นการรวบรวมรายละเอียดของงานที่จะเก็บและเรียกใช้ข้อมูล เช่น

- 1.1) เป็นข้อมูลเกี่ยวกับ subject อะไรบ้าง
- 1.2) มีลักษณะของข้อมูลแต่ละตัวเป็นอย่างไร เป็นเลขที่หลัก
- 1.3) วิธีการหรือความต้องการใช้งานข้อมูลเป็นแบบใด
- 1.4) ฯลฯ

ขั้นที่ 2 กำหนดโครงสร้าง Table

เป็นการลงสร้าง Table ขึ้นมาเป็นต้นแบบก่อน เพื่อจะนำไปใช้ในขั้นต่อไป ผลที่ได้จากขั้นนี้จะเป็นชุดของเอนทิตีและแอตทริบิวต์ทั้งหมดที่มีในฐานข้อมูล

ขั้นที่ 3 กำหนดคีย์

เป็นการกำหนดคีย์แบบต่างๆ ให้กับแต่ละ Table ที่ได้จากขั้นที่ 2 ตามความเหมาะสม เพื่อนำไปสร้างความสัมพันธ์กับ Table อื่นๆ ต่อไป

ขั้นที่ 4 กำหนดความสัมพันธ์

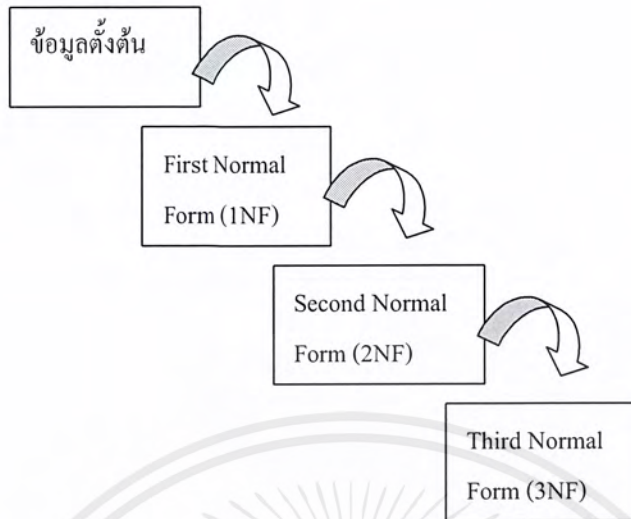
เป็นการกำหนดความสัมพันธ์ระหว่าง Table ในแบบต่างๆ โดยอาศัยคีย์ที่ตั้งไว้เป็นตัวเชื่อม

ขั้นที่ 5 ปรับรูปแบบข้อมูล (Normalization)

เป็นการนำเอาเอนทิตีและแอตทริบิวต์ที่ได้มาปรับรูปแบบและลดความซ้ำซ้อนเพื่อให้ถูกต้องตามหลักการ

2.5 การทำ Normalization

การทำ Normalization เป็นกระบวนการที่ใช้ปรับรูปแบบข้อมูลที่ออกแบบมาให้อยู่ในรูปแบบของ Table ที่เหมาะสม โดยผลที่ได้จะเรียกว่า “รูปแบบปกติ” (Normal Form) กระบวนการนี้มีได้สูงสุด 5 ขั้นตอนหรือเรียกว่า Normal Form แต่ในทางปฏิบัติมักจะทำกันไม่เกิน 3 ขั้น คือถึง 3NF เท่านั้น



รูปที่ 2.8 ขั้นตอนในการทำ Normalization

1) **First Normal Form (1NF)** ขั้นตอนนี้เป็นการกำจัดความซ้ำซ้อนกันที่ทำให้เกิดข้อมูลขึ้นหลายชุดอยู่ในเรคอร์ดเดียวกัน โดยจะแยกข้อมูลเหล่านั้นออกมาเป็นเรคอร์ดใน Table ใหม่ โดยใน Table เดิมจะเชื่อมโยงกับเรคอร์ดใน Table ใหม่แบบ One-To-Many

2) **Second Normal Form (2NF)** ใน Table ที่เป็น First Normal Forms แล้ว ให้พิจารณาว่าฟิลด์ใดที่ไม่ใช่คีย์ และเป็นข้อมูลที่ไม่ได้ขึ้นกับ Primary key1 ทั้งตัวหากขึ้นกับคีย์เหล่านั้นเพียงบางส่วน ข้อมูลเหล่านี้ถือว่าซ้ำซ้อนและสามารถกำจัดไปได้ อาจจับแยกเป็น Table ใหม่ก็ได้ ในตัวอย่างนี้ข้อมูลในใบสั่งซื้อ

3) **Third Normal Form (3NF)** จาก Table ที่เป็น Second Normal Form ให้พิจารณาต่อว่าฟิลด์ใดเป็นข้อมูลที่ไม่ได้ขึ้นกับ primary key เลย แต่ไปขึ้นกับฟิลด์อื่น หรือคำนวณๆ ได้จากฟิลด์อื่นๆ ในเรคอร์ดเดียวกันเท่านั้นเอง

-ในกรณีแรกที่ขึ้นฟิลด์อื่นก็อาจแยกเป็น Table ใหม่แยกก็ได้ โดยใช้ฟิลด์อื่นนั้นเป็นคีย์ของเรคอร์ดใน Table ใหม่

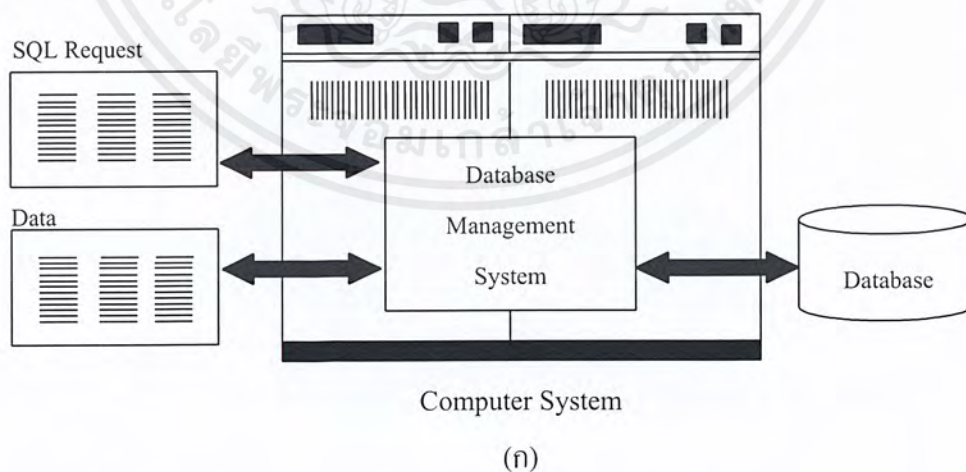
-ในกรณีหลังที่สามารถคำนวณได้จากฟิลด์อื่นก็อาจทิ้งไป โดยเมื่อจะใช้ค่อยทำการคำนวณใหม่แต่ละครั้ง

2.6 การออกแบบฐานข้อมูลเชิงสัมพันธ์

ฐานข้อมูลที่มีโครงสร้างข้อมูลในแบบ Relational ได้รับการพัฒนาขึ้นจากแบบจำลอง ข้อมูลที่จัดเก็บอยู่ในฐานข้อมูลที่มีโครงสร้างข้อมูลในแบบ Relational จะถูกแยกจัดเก็บออกเป็นหน่วยย่อยๆ ที่เรียกว่า Relation หรือเรียกว่า Table ที่อยู่ในรูปของตารางที่ประกอบด้วยชุดของแถวและชุดของสดมภ์ ข้อมูลที่จัดเก็บอยู่ในแต่ละ Relation จะเป็นข้อมูลที่แยกเป็นเอกเทศ แต่สามารถนำมาสร้างความสัมพันธ์ร่วมกันได้ โดยความสัมพันธ์ที่สร้างขึ้นนี้ จะอยู่ในรูปแนวคิดมากกว่าโครงสร้าง

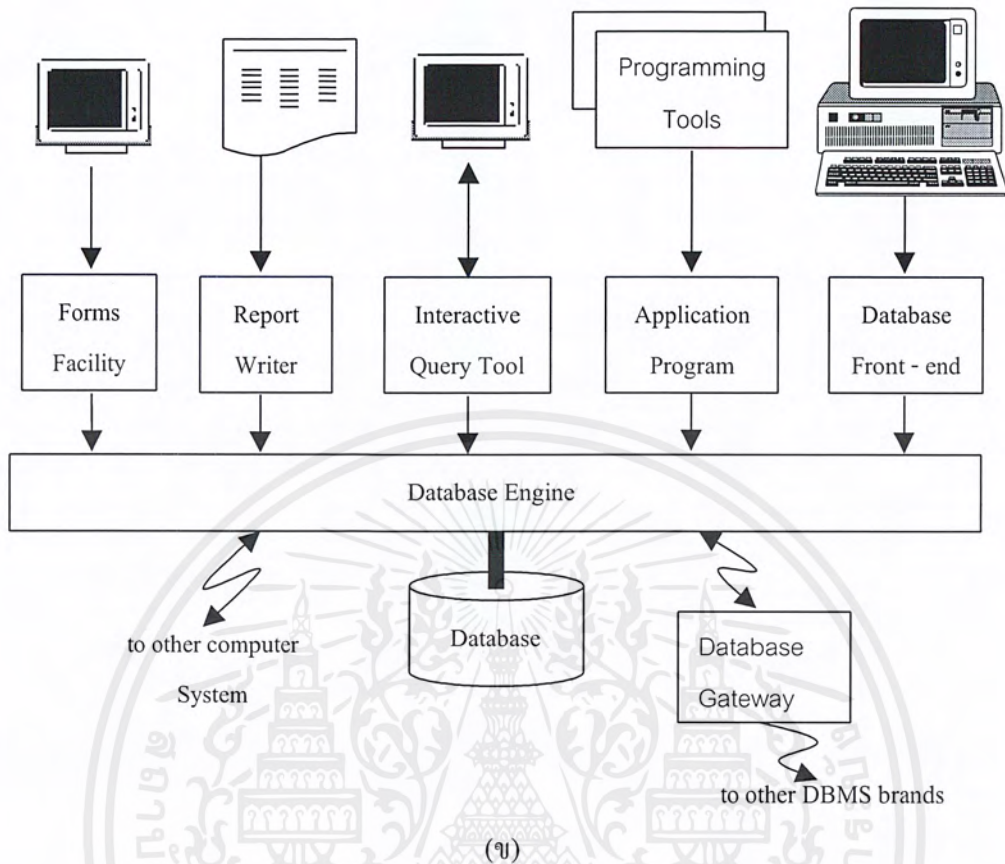
2.6.1 ฐานข้อมูลเชิงสัมพันธ์

ฐานข้อมูลเชิงสัมพันธ์เป็นระบบฐานข้อมูลที่ได้ออกแบบและพัฒนาแบบ Hierarchical Database (มีโครงสร้างข้อมูลแบบต้นไม้-Tree) และโมเดลแบบ Network Database ผู้ใช้งานจะเห็นข้อมูลถูกเก็บในรูปของตาราง (Table) สองมิติ ประกอบด้วย แถว (Row) และคอลัมน์ (Column) โดยที่ข้อมูลในแต่ละแถวจะประกอบด้วยหลายคอลัมน์ที่สัมพันธ์กัน เรียกว่า 1 เรคอร์ด ภาษา SQL เป็นภาษาฐานข้อมูลที่ใช้ในการติดต่อกับระบบจัดการฐานข้อมูล หรือ DBMS (Database Management System) เพื่อจัดการฐานข้อมูลค้นหาข้อมูล ฯลฯ ดังรูปที่ 2.9 ก. จะเห็นว่า SQL ไม่ใช่ DBMS แต่เป็นภาษาฐานข้อมูลที่ทำงานร่วมกับ DBMS บนฐานข้อมูลเชิงสัมพันธ์ จากรูปที่ 2.9 ข. จะเห็นว่า SQL จะไม่จัดการกับฐานข้อมูลในระดับกายภาพโดยตรง แต่จะสั่งให้ DBMS ซึ่งมี Database Engine เป็นตัวจัดการ ค้นหา จัดเก็บข้อมูลจากฐานข้อมูล



รูปที่ 2.9 (ก) การทำงานทั่วไปของระบบฐานข้อมูลเชิงสัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 (ข) การทำงานทั่วไปของระบบฐานข้อมูลเชิงสัมพันธ์

2.6.2 ส่วนประกอบของภาษาฐานข้อมูล

ภาษาฐานข้อมูล SQL จะทำการกำหนดโครงสร้างของข้อมูล จัดการฐานข้อมูล ค้นหาข้อมูล โดยการติดต่อและควบคุมระบบจัดการฐานข้อมูล (DBMS) อีกทีหนึ่ง มีส่วนประกอบสำคัญอยู่ 3 ส่วนคือ

1) ส่วนภาษานิยามข้อมูล (Data Definition Language : DDL)

เป็นส่วนของคำสั่งที่ใช้กำหนดโครงสร้างของข้อมูล สร้างฐานข้อมูล การสร้างตาราง (Base Tables) การกำหนดดัชนี (Indexes) การสร้างวิว (View) เพื่อผลลัพธ์ นอกจากนี้ยังใช้ในการกำหนดกฎเกณฑ์ต่างๆ ให้ DBMS ใช้ในการตรวจสอบความถูกต้องของข้อมูลด้วย

2) ส่วนภาษาจัดการข้อมูล (Data Manipulation Language : DML)

เป็นส่วนของคำสั่งในการค้นหาข้อมูล (Retrive) เพิ่มเติมข้อมูล (Insert) เปลี่ยนแปลงแก้ไขข้อมูล (Update) การลบข้อมูล (Delete)

3) ส่วนภาษาในการควบคุมข้อมูล (Data Control Language : DCL)

เป็นส่วนของคำสั่งที่ใช้ควบคุมการเข้าถึงข้อมูล (Access control) ของผู้ใช้ โดยจะกำหนดว่าผู้ใช้คนใดสามารถเพิ่ม ลบ แก้ไข-เปลี่ยนแปลงข้อมูล ผู้ใช้คนใด ทำได้เพียงเรียกดู ค้นหาข้อมูล เป็นการกำหนดสิทธิ์ (Privilege) ให้แก่ผู้ใช้งาน เพื่อความปลอดภัยให้กับข้อมูล

2.6.3 ส่วนประกอบพื้นฐานของฐานข้อมูลเชิงสัมพันธ์

1) ตาราง (Table)

ฐานข้อมูลเชิงสัมพันธ์จะบรรจุตารางไว้จำนวนมาก และในฐานข้อมูลจะเก็บอินฟอร์มเมชันที่สัมพันธ์กันทุกตาราง ในการเก็บข้อมูลทั้งในคอลัมน์และโลว์จะมีหลักการดังนี้

- 1.1) ตารางจะต้องมีชื่อไม่ซ้ำกัน และประกอบด้วยโลว์และคอลัมน์
- 1.2) ทุกคอลัมน์จะบรรจุข้อมูลได้เพียงชนิดเดียว เช่น char, Integer
- 1.3) ข้อมูลในแต่ละโลว์ต้องไม่ซ้ำกัน

2) หลัก (Column)

เป็นที่เก็บรายละเอียดต่างๆ ของข้อมูล และกำหนดประเภทของข้อมูลที่จะใช้ใช้งาน ลงไป เช่น คอลัมน์ Name ให้ประเภทของข้อมูลเป็น Character (ตัวอักษร) และคอลัมน์ Salary ให้ประเภทของข้อมูลเป็น Integer/Numeric ในตารางจะประกอบไปด้วยหลายคอลัมน์ เพื่อบรรจุนรายละเอียดของชื่อ ที่อยู่ ตำแหน่ง เงินเดือน ประวัติการทำงาน ฯลฯ

3) แถว (Row)

โลว์หรือแถวประกอบด้วยคอลัมน์หลายคอลัมน์ ใน 1 โลว์จะมีข้อมูลต่างๆ ที่อยู่ในแต่ละคอลัมน์ เช่น รหัสพนักงาน ชื่อ-นามสกุล ตำแหน่ง เงินเดือน วันลาป่วย ฯลฯ และในตารางข้อมูลจะประกอบด้วยหลายโลว์เช่นกัน

4) คีย์หลัก (Primary Key)

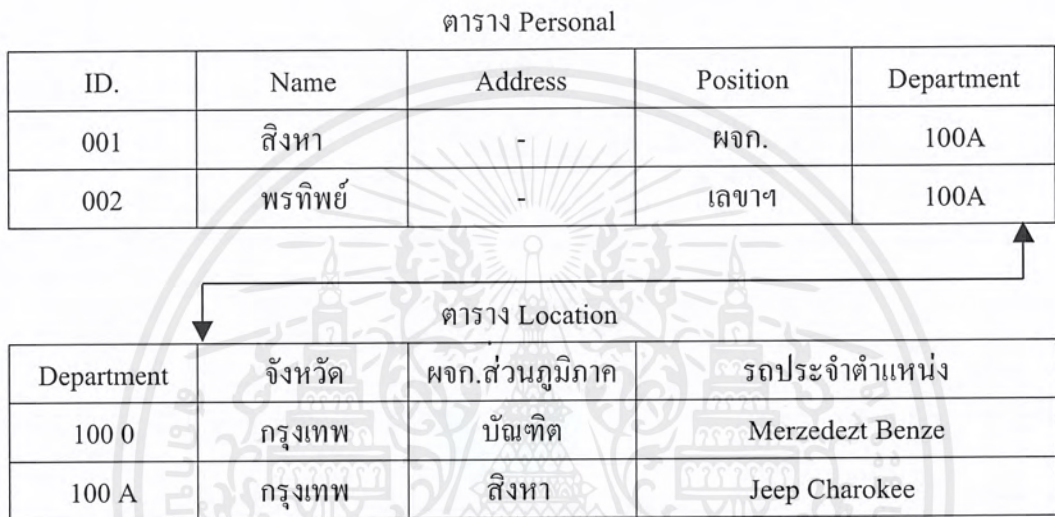
คีย์หลัก หมายถึง คอลัมน์ที่มีข้อมูลไม่ซ้ำกัน เป็นเรื่องสำคัญในระบบฐานข้อมูล คอลัมน์ที่มีคุณสมบัติเป็นคีย์หลักได้จะต้องมีค่าของข้อมูลไม่ซ้ำกันเลย เช่น คอลัมน์รหัสพนักงาน ซึ่งตามปกติแล้วพนักงานแต่ละคนจะมีรหัสประจำตัวต่างกันแน่นอน ข้อกำหนดคีย์ดังนี้

- 1.1) ในทุกตารางจะต้องมีคีย์หลัก
- 1.2) ใน 1 ตารางจะมีคีย์หลักได้เพียง 1 คีย์เท่านั้น
- 1.3) ในคีย์หลักค่าของข้อมูลในคอลัมน์จะต้องไม่เป็นค่าว่าง หรือ Null

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) คีย์นอก (Foreign Key)

คีย์นอกจะเป็นคอลลัมน์ หรือคอมบินชัน (Combination) ของตารางหนึ่ง ซึ่งความสัมพันธ์กับคอลลัมน์ในตารางอื่น และเชื่อมโยงข้อมูลระหว่างกันได้ดังรูปที่ 2.10 จะเห็นว่าตาราง Personal มีคอลลัมน์ชื่อ Department เป็นคีย์นอกซึ่งจะใช้ในการเชื่อมโยงข้อมูลกับคอลลัมน์ Department ในตาราง Location คีย์นอกสามารถจะมีค่าว่าง (Null) ได้



รูปที่ 2.10 ความสัมพันธ์ระหว่างตาราง Personal และ Location

6) โดเมน (Domain)

โดเมน คือ การกำหนดขอบเขต-ประเภทของข้อมูลหรือค่าของข้อมูล ที่ใช้ในคอลลัมน์หรือในฟิลด์ เช่นในคอลลัมน์ SALARY มีการกำหนดขอบเขต-ประเภทของข้อมูลให้เป็น Integer และคอลลัมน์อื่นๆ เช่น

- Name : กำหนดให้เฉพาะตัวอักษรใหญ่-เล็ก เท่านั้น
 Salary : กำหนดให้เฉพาะเลขจำนวนเต็มบวก มีค่าระหว่าง 0-999
 Address : กำหนดให้มีทั้งตัวอักษร และตัวเลข

2.7 Microsoft SQL Server

ฐานข้อมูลเชิงสัมพันธ์ SQL Server (Structure Query Language) เป็นแอปพลิเคชันฐานข้อมูลในระดับองค์กร หรือ Enterprise Database ที่มีประสิทธิภาพสูง สามารถรองรับข้อมูลขนาดใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทอร์ราไบต์ มีความยืดหยุ่นทางแพลตฟอร์ม ตั้งแต่ระดับเดสก์ท็อป (Windows 95/98) จนถึง เซิร์ฟเวอร์ (Windows NT Server) และ คลัสเตอร์เซิร์ฟเวอร์ (Windows NT Server Enterprise Edition) โดยใช้คำสั่งในการบริการและจัดการกลุ่มเดียวกัน Microsoft SQL Server สนับสนุนการทำ “Two phased Commit”(Tight Consistency) เพื่อรักษาเสถียรภาพของข้อมูลระหว่างเซิร์ฟเวอร์หลายๆ ตัวในระบบ โดยมี Commit เพื่อเก็บการเปลี่ยนแปลงข้อมูลที่รับ-ส่ง ระหว่างเซิร์ฟเวอร์ สามารถจะทำการ เรพลิตข้อมูลไปสู่ระบบฐานข้อมูลอื่น เช่น ORACLE, ACCESS โดยผ่านไดรฟ์เวอร์ ODBC(Open Database Connectivity) อีกด้วย

ตารางที่ 2.1 แสดงการเปรียบเทียบศัพท์ระหว่างฐานข้อมูล

ระบบข้อมูลแบบไฟล์	ฐานข้อมูลเชิงสัมพันธ์	SQL
ไฟล์ (File)	รีเลชัน (Relation)	ตาราง (Table)
ฟิลด์ (Field)	แอททริบิวต์ (Attribute)	คอลัมน์ (Column)
เรคอร์ด (Record)	ทัพเพิล (Tuple)	แถว (Row)

2.7.1 การรวมเข้ากับ Windows NT Server

1) ความปลอดภัย (Security) ฐานข้อมูล SQL Server ได้รวมระบบความปลอดภัยเข้ากับ Windows NT ทำให้การจัดการหรืออนุญาตในการเข้าถึงทรัพยากรเป็นไปอย่างง่ายดาย เช่น การใช้รหัสผ่านเดียวกัน สามารถจะเข้าถึงทรัพยากรของ SQL Server และ Windows NT ได้ นอกจากนี้ SQL Server ยังใช้การเข้ารหัสของ Windows NT สำหรับความปลอดภัยบนเน็ตเวิร์ก ได้อีกด้วย

2) การตรวจดูเหตุการณ์ (Event View) ระบบปฏิบัติการ Windows NT มีการเก็บบันทึกการทำงานต่างๆ ที่เกิดขึ้นไว้ใน Application, Security, System ช่วยให้ผู้บริหารระบบสามารถจะตรวจดู ล็อกแล้วแก้ปัญหาได้ SQL Server จะทำการเขียนเมสเสจการทำงานของฐานข้อมูลลงไปในล็อก ทั้ง 3 เพื่อใช้ในการติดตามการทำงาน และรายงานปัญหาที่เกิดขึ้นให้ทราบ

3) สนับสนุนมัลติโพรเซสเซอร์ (Multiprocessor Support) ระบบปฏิบัติการ Windows NT ได้ถูกออกแบบให้สนับสนุน มัลติโพรเซสเซอร์แบบ SMP (Symmetric Multiprocessing) ทำให้เครื่องเซิร์ฟเวอร์ที่มีซีพียู 2 ตัวขึ้นไป สามารถกระจายงานได้รวดเร็วกว่า SQL Server 7.0 ก็ สนับสนุน มัลติโพรเซสเซอร์ 4 ตัว ส่วน SQL Server 7.0 Enterprise สนับสนุนการใช้ โพรเซสเซอร์ สูงถึง 32 ตัว

4) เซอร์วิสบน NT (Windows NT Service) ผู้บริหารระบบคอมพิวเตอร์สามารถจะควบคุม Windows NT ได้ เพราะ มีเซอร์วิสให้จัดการ SQL Server จะทำงานเป็นเซอร์วิส บน Windows NT สามารถจะสตาร์ทหรือสตั๊อปเซอร์วิส ของ SQL ได้เช่นกัน

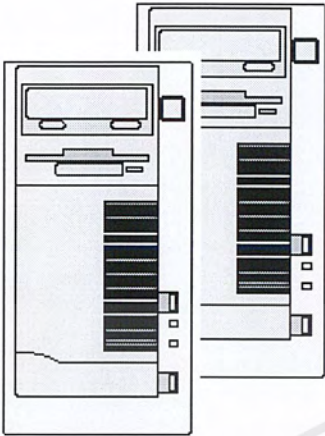
5) การเฝ้าดูประสิทธิภาพ (Performance Monitor) งานหลักในการบริหารระบบเน็ตเวิร์กให้มีความเร็วตามต้องการ สามารถจะใช้ Performance Monitor ในการเฝ้าดูประสิทธิภาพการทำงานของเซิร์ฟเวอร์, ฐานข้อมูล, เน็ตเวิร์ก, หน่วยความจำ ฯลฯ และเฝ้าดูประสิทธิภาพการทำงานของ SQL Server ได้ด้วย Performance Monitor

6) เซอร์วิสของ SQL Server ฐานข้อมูล SQL Server ทำงานร่วมกับระบบปฏิบัติการ Windows NT Server โดยมีเซอร์วิส 3 ตัว คือ MSSQLServer, SQL Server Agent และ MS DTC เซอร์วิส ดังรูปที่ 2.11 ซึ่งแต่ละตัวมีหน้าที่ดังนี้

- MSSQL Server Service เป็นเซอร์วิสของ RDBMS ที่จะใช้จัดการกับสแตทเมนต์ Transact SQL จัดการทุกไฟล์ฐานข้อมูลบนเซิร์ฟเวอร์ จัดสรรทรัพยากรของระบบให้ผู้ใช้กันอย่างเหมาะสม ในกรณีเกิดสภาวะใช้งานพร้อมกัน ป้องกันปัญหาจากการ Update ที่เหมือนกันในเวลาเดียวกัน

- SQL Server Agent Service เป็นเซอร์วิสของ NT ที่จะรันไปพร้อมๆ กับ SQL Server ในการสร้างและจัดการกับ Jobs, Alerts และ Operators ของโลกอลเซิร์ฟเวอร์ หรือมัลติเซิร์ฟเวอร์ เมื่อ SQL Server Agent Service เริ่มสตาร์ทจะทำการรีจิสเตอร์กับ EventLog Server และเชื่อมต่อเข้า SQL Server ทำให้ SQLServerAgent สามารถจะรับการแจ้งเตือนจาก SQL Server Event ทันทีเมื่อมีการเขียนลงล็อกของ NT (Application log) SQLServerAgent Service จะทำการสื่อสารกับ MSSQL Server Service ในการแสดงเหตุการณ์ที่เกิดขึ้น

- Microsoft Distributed Transaction Coordinator (MS DTC) Service เป็นส่วนหนึ่งของ Microsoft Transaction Server ทรานส์แอ็คชั่นจะจัดการให้ไคลเอนต์สามารถรับข้อมูลจากภายนอกได้หลายรูปแบบ MS DTC จะใช้การทำ Two phase commit เพื่อประสานการกระจายทรานส์แอ็คชั่น ไปยังทุกเซิร์ฟเวอร์บนระบบอย่างสมบูรณ์และถูกต้อง



SQL Server

MSSQL Server - Data Management

- Transaction and Query Processing

SQL Server Agent - Jobs

- Alerts

- Operators

MS DTC - Distributed Transactions Management

รูปที่ 2.11 แสดงเซอรัวิสของ SQL Server

2.8 การใช้งาน MS-SQL

เนื่องจาก SQL เป็นภาษาที่ใช้ในการเขียนโปรแกรมเพื่อจัดการฐานข้อมูลโดยเฉพาะ แบ่งการทำงานได้ 4 ประเภทดังนี้

- 1) Select query ใช้สำหรับดึงข้อมูลที่ต้องการ (retrieve)
- 2) Update query ใช้สำหรับแก้ไขข้อมูล (edit)
- 3) Insert query ใช้สำหรับเพิ่มข้อมูล (append)
- 4) Delete query ใช้สำหรับลบข้อมูลออกไป (delete)

1) Select query

ใช้ในการดึงข้อมูลที่ต้องการจากฐานข้อมูล โดยมีรูปแบบการใช้งานดังนี้

```
Select column list From table Where condition
```

ตัวอย่างเช่น

```
Select Name,Surname From employee
```

เมื่อโปรแกรมเริ่มทำงาน จะใช้คำสั่งที่เขียนด้วยภาษา SQL ไปดึงข้อมูลที่ต้องการคือ ชื่อ และนามสกุลจากฐานข้อมูลพนักงานจากตารางข้อมูล employee แล้วสร้างตารางผลลัพธ์ดังตารางที่ 2.2

ตารางที่ 2.2 รายชื่อและนามสกุลของพนักงานที่ได้

Name	Surname
กิตติภูมิ	วรรณิตร
คณกรณ์	หอศิริธรรม
ทนงศักดิ์	ยี่ละ

หากต้องการเฉพาะรายชื่อและนามสกุลของพนักงานในแผนกคอมพิวเตอร์ ก็จะต้องระบุเงื่อนไขเพิ่มเติมเข้าไปด้วยดังนี้

```
Select Name,Surname From employee Where
Department="คอมพิวเตอร์"
```

จะได้ผลลัพธ์ดังตารางที่ 2.3

ตารางที่ 2.3 รายชื่อและนามสกุลของพนักงานในแผนกคอมพิวเตอร์

Name	Surname
กิตติภูมิ	วรรณิตร

และหากต้องการให้แสดงข้อมูล โดยเรียงรายการตามแผนกจากน้อยไปมาก จะใช้คำสั่งดังนี้

```
Select Name,Surname,Department From employee Order
by department
```

จะได้ผลลัพธ์ดังตารางที่ 2.4

ตารางที่ 2.4 รายชื่อและนามสกุลของพนักงานเรียงตามชื่อแผนก

Name	Surname	Department
คณกรณ์	หอศิริธรรม	การตลาด
ทงศักดิ์	ยี่ละ	การตลาด
กิตติภูมิ	วรรณตร	คอมพิวเตอร์

2) Update query

ใช้สำหรับแก้ไข-ปรับปรุงข้อมูล โดยมีวิธีการใช้ดังนี้

Update table Set = new value Where condition

จากตัวอย่างฐานข้อมูลพนักงาน กรณีที่พนักงานชื่อ ทงศักดิ์ ยี่ละ ได้ย้ายแผนกจากการตลาด ไปอยู่ คอมพิวเตอร์ จะใช้คำสั่งดังนี้

Update employee Set department = 'คอมพิวเตอร์' Where Name='ทงศักดิ์'
and Surname='ยี่ละ'

จะได้ผลลัพธ์เป็นการเปลี่ยนแปลงข้อมูลของพนักงานที่มีชื่อเท่ากับ ทงศักดิ์ และนามสกุลเท่ากับ ยี่ละ ตามเงื่อนไขไปเป็นแผนก คอมพิวเตอร์ เมื่อใช้ select query เรียกข้อมูลเฉพาะแผนกคอมพิวเตอร์ดังนี้

Select Name, Surname From employee

จะได้ผลลัพธ์ดังตารางที่ 2.5

ตารางที่ 2.5 รายชื่อและนามสกุลของพนักงานจากตาราง employee

Name	Surname	Department
คุณกรณ์	ห่อศิริธรรม	การตลาด
ทองศักดิ์	ยี่ละ	คอมพิวเตอร์
กิตติภูมิ	วรฉัตร	คอมพิวเตอร์

โดยที่ข้อมูลของพนักงานที่ชื่อทองศักดิ์ ยี่ละ ได้ถูกเปลี่ยนแปลงข้อมูลแผนกจากการตลาดไปเป็น คอมพิวเตอร์ ตามคำสั่ง update query

3) Insert query

ใช้ในการเพิ่มเติมข้อมูลใหม่เข้าไปในฐานข้อมูล มีรูปแบบการใช้งานดังนี้

```
Insert Into table (column list) Values (values list)
```

เมื่อมีพนักงานใหม่เพิ่มขึ้น โดยที่พนักงานใหม่มีรหัสพนักงาน เป็น 4 มีชื่อว่า สันติ นามสกุล เสียมใหม่ จะอยู่แผนก การตลาด จะเขียนได้ดังนี้

```
Insert Into employee (code,Name,Surname,Department) Values ('4', 'สันติ', 'เสียมใหม่', 'การตลาด')
```

การเพิ่มข้อมูลด้วยคำสั่ง Insert Into เป็นการเพิ่มรายการข้อมูลเข้าไปในตารางข้อมูลและบันทึกค่าข้อมูลในแต่ละคอลัมน์ ถ้าคอลัมน์ใดต้องการบันทึกข้อมูล จะต้องระบุค่าของข้อมูล (value) ให้มีตำแหน่งตรงกับคอลัมน์ คือ column list และ values list จะต้องสัมพันธ์กัน

4) Delete query

ใช้ในการลบข้อมูลออกจากฐานข้อมูล มีรูปแบบการใช้งานดังนี้

```
Delete From table Where condition
```

ตัวอย่างการลบข้อมูลออกจากตารางพนักงาน ถ้าพนักงานรหัสที่ 3 ลาออก ก็จะต้องลบข้อมูลออกจากตารางด้วยคำสั่งดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Delete From employee Where code= '3'

2.8.1 การออกแบบฐานข้อมูล

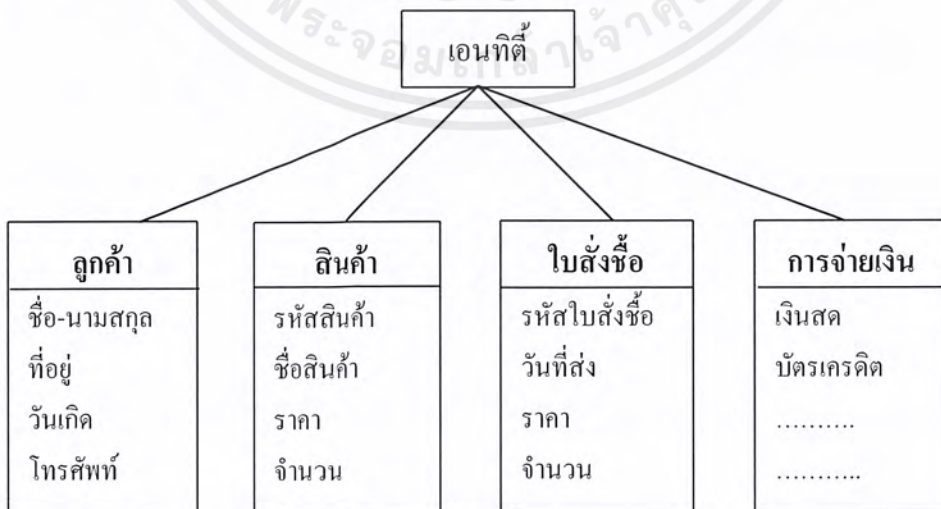
ก่อนที่จะสร้างตาราง ควรจะทำการออกแบบฐานข้อมูลให้เหมาะสมกับงาน เพื่อเพิ่มประสิทธิภาพให้กับข้อมูลที่จะใช้งานและง่ายในการปรับแต่ง การจัดระเบียบให้ข้อมูลอยู่เป็นหมวดหมู่ จะช่วยในการดูแลรักษาฐานข้อมูล โดยจะใช้ Normalization ช่วยลดความซ้ำซ้อนของข้อมูล

เอนทิตี (Entity) หมายถึง คน สถานที่ วัตถุ หรือสิ่งต่างๆ เช่นพนักงาน ลูกค้า สินค้า ราคา ใบสั่งซื้อ ฯลฯ การออกแบบสร้างฐานข้อมูลเกี่ยวกับประวัติศาสตร์ซื้อสินค้าของลูกค้า เอนทิตีในส่วนนี้จะมีลูกค้า สินค้า ใบสั่งซื้อ การจ่ายเงิน เป็นต้น ดังรูปที่ 2.12



รูปที่ 2.12 ตัวอย่างเอนทิตี

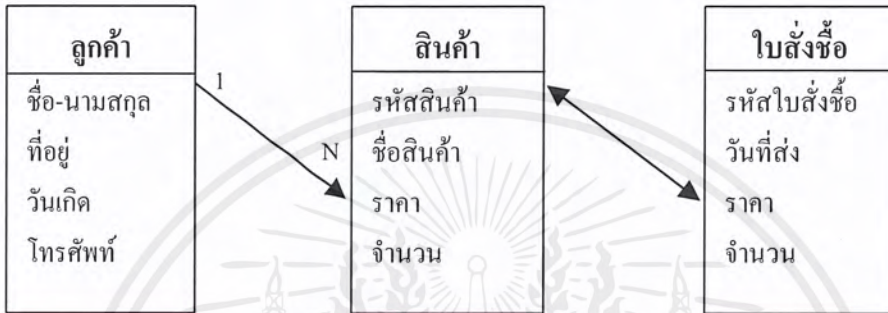
แอททริบิวต์ (Attribute) หมายถึง ข้อมูลแต่ละเอนทิตีที่ใช้แสดงรายละเอียด เกี่ยวกับเอนทิตีนั้นๆ เช่น แอททริบิวต์เอนทิตีของลูกค้า จะประกอบไปด้วย ชื่อ ที่อยู่ วันเกิด แอททริบิวต์เอนทิตีของสินค้าจะประกอบด้วย รหัส ชื่อสินค้า ราคา ดังรูปที่ 2.13



รูปที่ 2.13 ตัวอย่างแอททริบิวต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์ (Relationships) หมายถึง ความสัมพันธ์ของเอนทิตีที่มีต่อกันบนระบบ เช่น ประวัติการซื้อสินค้าของลูกค้าประกอบด้วยเอนทิตีลูกค้า เอนทิตีสินค้าและเอนทิตีใบสั่งซื้อ ดังรูปที่ 2.14 มีความสัมพันธ์ระหว่าง 3 เอนทิตี โดยเอนทิตีลูกค้าเชื่อมความสัมพันธ์กับสินค้า และสินค้านี้มีความสัมพันธ์กับใบสั่งซื้อฐานข้อมูลเชิงสัมพันธ์จึงเป็นการรวมเอนทิตีที่มีความสัมพันธ์ระหว่างกันบนระบบเอาไว้ในฐานข้อมูล



รูปที่ 2.14 ตัวอย่างความสัมพันธ์ระหว่างเอนทิตีลูกค้า, สินค้า และใบสั่งซื้อ

ในการออกแบบฐานข้อมูลควรจะมีการวางแผน มีการกำหนดความสัมพันธ์ระหว่างเอนทิตีของแต่ละตารางให้เหมาะสม และจัดแบ่งประเภทของข้อมูลให้ถูกต้อง หลักการพื้นฐานสำหรับการออกแบบฐานข้อมูล มีดังนี้

1. รวบรวมข้อมูลที่เกี่ยวข้องกับตารางนั้นๆ ให้มากที่สุด เช่น ตาราง Customers ควรจะมีข้อมูลที่เกี่ยวข้องกับชื่อ-นามสกุล ที่อยู่ วัน/เดือน/ปี เบอร์โทรศัพท์

2. ทำการ Normalization เพื่อให้เกิดความซ้ำซ้อนน้อยที่สุด เนื่องจากในตารางจะประกอบไปด้วยฟิลด์ข้อมูลหลายฟิลด์

3. กำหนดค่าคีย์หลัก (Primary Key) และคีย์นอก (Foreign Key) สำหรับตารางที่เกี่ยวข้องหรือมีความสัมพันธ์กัน

4. ให้กำหนดประเภทของข้อมูลอย่างเหมาะสมกับแต่ละฟิลด์ เพราะประเภทข้อมูลจะมีผลต่อการทำงานโดยตรง เช่น ฟิลด์ Name ควรเป็น Character ฟิลด์ Salary ควรเป็น Numeric

2.8.2 การสร้างตาราง

ฐานข้อมูลเชิงสัมพันธ์จะเก็บข้อมูลเอาไว้ในรูปของตาราง (Table) ซึ่งทุกตารางสามารถที่จะเชื่อมโยงข้อมูลระหว่างกันได้ ในการสร้างตารางด้วยภาษา SQL จะใช้ส่วนภาษานิยามข้อมูล (Data Definition Language : DDL) เพื่อกำหนดโครงสร้างของข้อมูล รูปแบบคอลัมน์ ชนิดของข้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มูด ก่อนที่จะเริ่มสร้างตารางจะออกแบบตารางคร่าวๆ บนแผ่นกระดาษ โดยเขียน ชื่อ คอลัมน์ ชนิดข้อมูล ความกว้าง กำหนดคีย์หลัก และความสัมพันธ์ระหว่างตารางให้ถูกต้อง

2.8.3 การออกแบบโครงสร้างของตาราง

ในการออกแบบโครงสร้างของตารางจะต้องใช้ทฤษฎีและความเป็นจริงร่วมกัน เพื่อให้ได้ ตารางข้อมูลที่มีประสิทธิภาพ สามารถเปลี่ยนแปลงแก้ไขได้ในอนาคตและไม่กระทบกระเทือนกับ ส่วนอื่น ที่สำคัญจะต้องทราบวัตถุประสงค์ในการทำงานว่า ตารางนี้จะใช้เก็บข้อมูลเกี่ยวกับอะไร เช่น ประวัตินักศึกษา ผลสอบแต่ละวิชา ประวัติพนักงาน

ตารางที่ 2.6 ตัวอย่างโครงสร้างของตาราง Customers

ชื่อคอลัมน์	ชนิดข้อมูล	ขนาดความกว้าง	ฟิลด์ว่าง	ความหมาย
Cust_ID	Nchar	4		รหัสของลูกค้า
Name	Nvarchar	20		ชื่อ-นามสกุล
Sex	Char	1		เพศ
Address	Nvarchar	30		ที่อยู่
City	Nvarchar	30		จังหวัด
Postalcode	Nvarchar	5		รหัสไปรษณีย์
Region	Nvarchar	10		ภาค

2.8.4 ภาษา SQL (Structured Query Language)

การใช้ภาษา SQL นั้น ผู้เขียนโปรแกรมไม่จำเป็นต้องรู้ภาษาคอมพิวเตอร์ลึกซึ้งซึ่งมากนัก เพียงแต่จะต้องศึกษาหลักภาษา SQL และ ทฤษฎีเกี่ยวกับเซต (SET) ที่สัมพันธ์กับภาษานี้อย่างลึกซึ้ง จึงจะสามารถเขียนโปรแกรมได้อย่างคล่องแคล่ว หลักการของการใช้ภาษา SQL อยู่กับหลักการเดียวกับฐานข้อมูลแบบสัมพันธ์ กล่าวคือ ข้อมูลจะเก็บในตาราง (Table) และในตารางจะประกอบด้วย ฟิลด์ข้อมูลต่างๆ ตามทฤษฎีของฐานข้อมูล ตารางแนวคอลัมน์ เรียกว่า Attributes สำหรับแถวแนวนอน Row เรียกว่า Tuples

จากการตั้งชื่อของตารางในภาษา SQL นั้น มีหลักการง่ายๆ กล่าวคือ จะต้องนำหน้าชื่อ ตารางด้วยพยัญชนะใดๆ หรือ \$, # หรือ @ ก็ได้ และไม่ควรตั้งชื่อย่อยยาวกว่า 18 ตัวอักษร และ ในชื่อสามารถผสมคำจากอักษรใดๆ และอักษรโรมันตัวเล็ก เลข 0..9, \$, #, @ และ _ (Underscores อย่างไรก็ตาม ชื่อไม่ควรสั้นเกินไป (m หรือ mm) หรือ ไม่ควรยาวเกินไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.5 การเลือกข้อมูลด้วยคำสั่ง SELECT

คำสั่งพื้นฐานในการเลือกข้อมูล คือ SELECT...FROM...[WHERE...]

1) หลักการของ SELECT มี Syntax ดังนี้ คือ

```
SELECT      [predicate] { * | table.* | [table.] field1 [AS alias1] [, [table.]
            field2 [AS alias2] [, ...] }

FROM table expression [, ...] [IN externaldatabase]

[WHERE...]

[GROUP BY...]

[HAVING...]

[ORDER BY...]
```

ข้อความในเครื่องหมายวงเล็บใหญ่ [] หมายความว่า ไม่จำเป็นที่จะใช้ก็ได้ คำสั่ง SELECT เป็นคำสั่งที่ใช้ในการเลือกฟิลด์ข้อมูลที่ต้องการจากตารางที่กำหนด หรือถ้าต้องการดูทั้งหมด จะใส่สัญลักษณ์ * (SELECT *) โดยต้องใส่ชื่อของตารางลงในช่อง FROM ซึ่งจะเป็นชื่อตาราง หรือ View ก็ได้ นอกจากนี้ยังสามารถใส่เงื่อนไขของการเลือกลงไป หลังจากคำสั่ง WHERE ได้อีกด้วย

2) หลักการเปรียบเทียบตามเงื่อนไข

การเปรียบเทียบตามเงื่อนไขของภาษา SQL จะต้องอยู่หลังคำสั่ง WHERE ซึ่งสามารถเปรียบเทียบตามสัญลักษณ์ LOGICAL BOOLEAN นอกจากนี้ยังสามารถใช้เงื่อนไขพิเศษ เช่น NOT, OR และ AND ได้อีกด้วย

2.1) คำสั่ง ORDER BY

เป็นคำสั่งที่ต้องการเรียง หรือ Sorting ข้อมูลที่อยู่ภายในฟิลด์ที่กำหนดสามารถกำหนดการเรียงจากมากมาหาน้อย หรือ จากน้อยมามากได้ (Descending order, Ascending order)

2.2) คำสั่ง SELECT DISTINCT

คำสั่งนี้ใช้สำหรับในการเลือกข้อมูลที่บางครั้งมีหลายๆ ฟิลด์ข้อมูลที่ซ้ำกัน

2.8.6 การใช้ AND และ OR ในภาษา SQL

AND และ OR สามารถนำมาใช้ใน WHERE ได้ในกรณีที่มีเงื่อนไขที่มีความซับซ้อน เช่น ต้องการทราบชื่อของเจ้าหน้าที่ ที่มีเงินเดือนมากกว่า 20,000 และ ทำงานในแผนก MIS จะเขียนได้เป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SELECT NAME FROM EMPLOYEE WHERE SALARY > 20000 AND DEPT = 'MIS' ;
```

2.8.7 การใช้เงื่อนไข NOT ในภาษา SQL

NOT สามารถนำไปหา SET ที่ไม่ใช่สิ่งที่อยู่ในเงื่อนไขได้ เช่น ถ้าต้องการค้นหาชื่อพนักงานทุกคนที่ไม่ได้ทำงานในแผนก MIS แทนที่จะทำการเช็คทีละแผนก สามารถใส่คำว่า NOT DEPT = 'MIS' ได้ทันที ตัวอย่างเช่น ต้องการทราบชื่อของพนักงานทุกคน และ แผนกของพนักงานที่ไม่ได้ทำงานในแผนก MIS จะเขียนได้ดังนี้

```
SELECT NAME, DEPT FROM EMPLOYEE WHERE NOT DEPT = 'MIS'
```

2.8.8 การใช้เงื่อนไข IN ในภาษา SQL

หลายๆ ครั้งที่การค้นหาข้อมูลจะต้องเขียนเงื่อนไขที่ยาว เช่น ถ้าต้องการทราบชื่อพนักงานที่ทำงานในแผนก MIS หรือ ACCOUNTING หรือ MARKETING จะต้องเขียนว่า

```
SELECT NAME FROM EMPLOYEE WHERE
DEPT = 'MIS'
OR DEPT = 'ACCOUNTING'
OR DEPT = 'MARKETING'
```

2.8.9 การใช้เงื่อนไข BETWEEN ในภาษา SQL

เงื่อนไขการเปรียบเทียบข้อมูลระหว่าง เช่น ต้องการทราบข้อมูลพนักงานที่มีรายได้ระหว่าง 12000 ถึง 20000 ว่ามีใครบ้าง โปรแกรมเมอร์สามารถเขียนได้ 2 แบบ คือ

```
SELECT NAME FROM EMPLOYEE WHERE
SALARY >= 12000 AND SALARY <= 20000
```

หรือ ใช้คำว่า BETWEEN

```
SELECT NAME FROM EMPLOYEE WHERE
SALARY BETWEEN 1200 AND 20000
```

2.8.10 คำสั่งในการค้นหาข้อมูลด้วย LIKE NOT และ LIKE

LIKE เป็นคำสั่งที่ใช้ในการเปรียบเทียบข้อมูลอักษร โดยสามารถใส่อักษรตัวต้น หรือ ตัว ถัดๆ ไปได้ใน การตรวจสอบ เช่น ต้องการทราบรายชื่อพนักงานที่มีอักษรนำหน้าด้วย 'R' จะเขียน ด้วยภาษา SQL ได้ดังนี้

```
SELECT NAME FROM EMPLOYEE WHERE NAME LIKE 'R*';
```

2.9 การทำงานแบบ Client / Server

Client/Server เป็นสถาปัตยกรรมทางด้านระบบคอมพิวเตอร์แบบ Distributed แบบหนึ่ง เป็นการประมวลผลแบบกระจาย โดยจะแบ่งส่วนการทำงานของโปรแกรมออก เพื่อให้เครื่องที่เป็น ลูกข่ายทำงาน ส่วนหนึ่งแทนที่จะให้คอมพิวเตอร์ส่วนกลางทำทั้งหมด ความสามารถของเครื่องลูก ข่ายในการประมวลผลแล้วเครื่องลูกข่ายและเครื่องส่วนกลาง จะเชื่อมต่อกันด้วยเครือข่าย ส่วนมาก จะเป็นเครือข่ายท้องถิ่น (Local Area Network หรือ แลน) โดยที่เครื่องส่วนกลางจะถูกเรียกว่า Server ส่วนเครื่องไมโครคอมพิวเตอร์ จะถูกเรียกว่า Client แบ่งโปรแกรมออกเป็น 3 ส่วน คือ

- 1) ส่วนติดต่อกับผู้ใช้ (User Interface หรือ UI)
- 2) ส่วนที่ประมวลผลงาน (Business Logic หรือ BL)
- 3) ส่วนที่ปรับปรุงฐานข้อมูล (Data Access หรือ DA)

ภายในระบบคอมพิวเตอร์แบบ Client/Server โดยทั่วไป มักจะประกอบด้วย เครื่อง คอมพิวเตอร์ที่ทำหน้าที่เป็น Server 1 เครื่อง ที่เชื่อมต่อกับเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Client มากกว่า 1 เครื่อง ซึ่งจะมีจำนวนเท่าใดขึ้นอยู่กับปริมาณงานภายในระบบ เครื่องคอมพิวเตอร์ที่ทำ หน้าที่เป็น Server จะใช้จัดเก็บฐานข้อมูล และโปรแกรม DBMS ที่มีหน้าที่ในการรับคำสั่งจาก Client มาประมวลผล และนำข้อมูลข่าวสารที่ Client ต้องการ ส่งกลับไปยัง Client ดังนั้น จึงมักเรียก ส่วน Server นี้ว่า ส่วนของโปรแกรม Backend ส่วนเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Client แต่ละ เครื่อง จะใช้สำหรับ Run โปรแกรมที่ทำหน้าที่ในการคำสั่งสำหรับเรียกใช้ข้อมูลจากฐานข้อมูลไป ยัง Server เพื่อให้ Server ส่งข้อมูลข่าวสารที่ต้องการกลับมา และนำไปแสดงผลให้กับผู้ใช้ต่อไป ดัง นั้น จึงมักเรียกส่วน Client นี้ว่า ส่วนของโปรแกรม Frontend

ในบางระบบ อาจมีการใช้ Server มากกว่า 1 ตัว ซึ่งอาจเนื่องมาจาก มีการใช้ฐานข้อมูล หลายฐานข้อมูลและจำเป็นที่จะต้องแยกเก็บใน Server ที่ต่างเครื่องกัน ในกรณีนี้ Client จะต้อง สามารถที่จะเชื่อมต่อกับ Server แต่ละ Server ได้ แต่ในการเชื่อมต่อของ Client ไปยัง Server ในแต่ละเครื่อง จะเกิดขึ้นได้เพียง Server ตัวเดียว ดังนั้น ผู้ใช้ของระบบในลักษณะนี้ จึงต้องทราบถึง Server ที่จัดเก็บข้อมูลที่ตนต้องการ แต่ก็มีบางกรณีที่ Client สามารถใช้งานหลาย ๆ Server ในแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อเนื่องได้ ซึ่งมักเกิดกับฐานข้อมูล ที่มีการแยกข้อมูลในฐานข้อมูลออกเป็น ส่วน ๆ และจัดเก็บลงในแต่ละ Server แต่การใช้งานในกรณีนี้ จะไม่จัดเป็นระบบคอมพิวเตอร์ในแบบ Client/Server เนื่องจาก ระบบมีการทำงานที่ตรงข้ามกับระบบคอมพิวเตอร์ในแบบ Client/Server กล่าวคือ แต่ละ Server จะมอง Client เป็น Server เพียง Server เดียวแทน เนื่องจาก จะต้องทยอยส่งข้อมูลไปให้กับ Client แทน

2.9.1 องค์ประกอบของการทำงานแบบ Client/Server

ส่วนประกอบของการทำงานในระบบ ประกอบด้วย 3 ส่วน คือ

1) เครื่องลูกข่ายหรือ Client

เครื่องที่เป็นลูกข่ายจะทำงานในส่วนของ UI และ BL ซึ่งงานที่ต้องทำได้แก่ การแสดงผล ลัพท์, การนำข้อมูลเข้า, การตรวจสอบความถูกต้องของข้อมูลและการคำนวณต่างๆ

2) ระบบเครือข่าย

เป็นส่วนที่ใช้สำหรับสื่อสารความต้องการและผลลัพธ์ระหว่างเครื่อง Client กับ Server

3) เครื่องแม่ข่ายหรือ Server

เป็นเครื่องที่ทำหน้าที่ให้บริการด้านต่างๆ แก่เครื่อง Client เช่น เพิ่มข้อมูล, งานพิมพ์, ฐานข้อมูล, โทรสาร, การเชื่อมต่อระยะไกล เป็นต้น

2.9.2 การเชื่อมต่อระหว่าง Client กับ Server ฐานข้อมูล

มาตรฐานการเชื่อมต่อระหว่างโปรแกรม แบ่งออกเป็น 2 แบบ คือ ระบบปิด (Closed System) และระบบเปิด (Opened System)

1) ระบบปิด

เป็นระบบที่ผู้พัฒนาฐานข้อมูลหรือผู้พัฒนาโปรแกรมด้าน Client สร้างโปรแกรม สำหรับการเชื่อมต่อขึ้นมาเอง

2) ระบบเปิด

เป็นระบบที่มีมาตรฐานกลางในการเชื่อมต่อซึ่งอยู่กึ่งกลางระหว่าง PC Client กับ Server ฐานข้อมูล ทำให้ผู้ใช้สามารถเขียนโปรแกรมโดยไม่จำเป็นต้องกำหนดว่าจะใช้กับระบบฐานข้อมูลใด โดยที่การติดต่อจะทำผ่านภาษา SQL (Structured Query Language) ซึ่งถูกแปลความหมายโดย Server ฐานข้อมูลแล้วส่งผลลัพธ์มาให้ PC Client มาตรฐานของระบบนี้ได้แก่

2.1) ODBC (Open Database Connectivity)

ODBC ได้กำหนดวิธีการที่จะติดต่อระหว่าง Client กับ Server และได้แบ่งระดับความสามารถในการทำงานออกเป็น ระดับ 3 ระดับคือ Core, Level 1, Level 2 ซึ่งระดับที่จำเป็นในการ

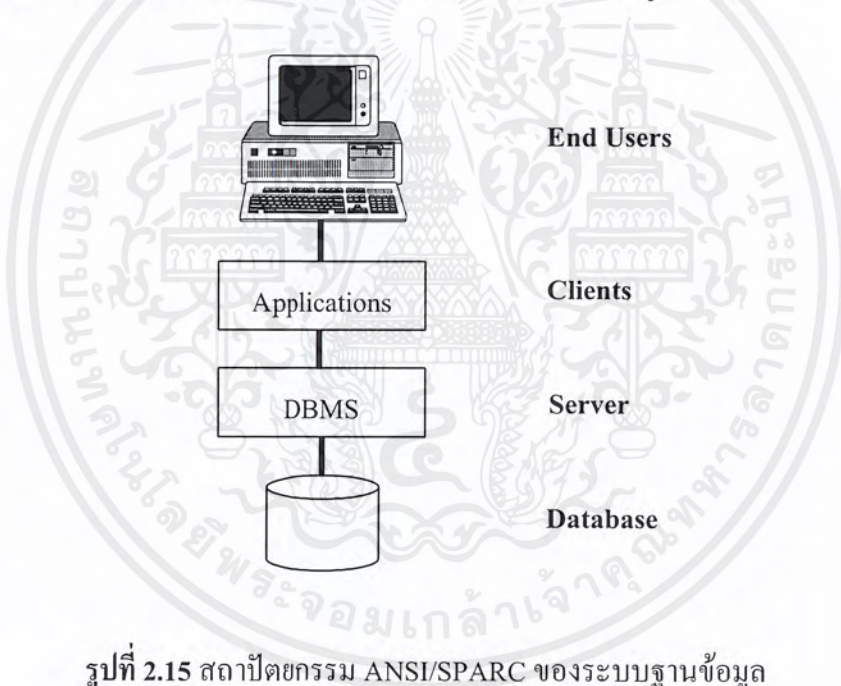
ทำงานคือระดับ Core ข้อดีของ ODBC คือโปรแกรมที่เขียนขึ้นจะไม่ผูกติดกับระบบฐานข้อมูล ทำให้ผู้ใช้สามารถเปลี่ยนระบบฐานข้อมูลได้ง่าย

2.2) IDAPI (Integrated Database Application Programming Interface)

IDAPI ได้กำหนดวิธีการติดต่อระหว่าง Client กับ Server ไว้เพื่อให้โปรแกรมที่เขียนขึ้นสามารถใช้งานกับฐานข้อมูลหลายๆ ประเภทได้ การเชื่อมต่อระหว่าง Client กับ Server จะทำผ่านโปรแกรม BDE (Borland Database Engine)

2.9.3 ระบบ Client/Server กับฐานข้อมูล

ด้วยมุมมองตามสถาปัตยกรรม ANSI/SPARC ของระบบฐานข้อมูล อาจสามารถแบ่งระบบฐานข้อมูลออกเป็น 2 ส่วนคือ ส่วนของ Server ซึ่งโดยทั่วไปมักเรียกว่า โปรแกรม Backend และส่วนของ Client ซึ่งโดยทั่วไปมักเรียกว่า โปรแกรม Front-end ดังรูปที่ 2.15



รูปที่ 2.15 สถาปัตยกรรม ANSI/SPARC ของระบบฐานข้อมูล

ส่วน Server ในแง่ของฐานข้อมูล ได้แก่ โปรแกรม DBMS ที่ทำหน้าที่สนับสนุนการทำงานต่าง ๆ ของฐานข้อมูล เช่น การสร้างและกำหนดโครงสร้างของฐานข้อมูล การจัดการข้อมูลภายในฐานข้อมูล การรักษาความปลอดภัยให้กับข้อมูล การรักษาความถูกต้องของข้อมูล ฯลฯ เป็นต้น

ส่วน Client ในแง่ของฐานข้อมูล ได้แก่ โปรแกรมต่าง ๆ ที่ทำงานร่วมกับโปรแกรม DBMS เพื่อใช้งานฐานข้อมูล โปรแกรมเหล่านี้ อาจพัฒนาขึ้นด้วยภาษาคอมพิวเตอร์ภาษาใดภาษาหนึ่ง เช่น ภาษา C, COBOL, FOCUS ฯลฯ เป็นต้น หรืออาจพัฒนาขึ้นด้วย Tool ของตัว DBMS เองเช่น Tool ที่ใช้ในการสร้างรายงานต่าง ๆ ภาษา Query Language ตารางที่ใช้แสดงข้อมูลในรูปแบบ Spreadsheet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาษาที่ใช้ในการสร้าง Form สำหรับจัดการกับข้อมูลหรือที่เรียกว่า 4GL รูปภาพกราฟฟิกที่ใช้แสดงข้อมูลทางสถิติ หรือในบาง Tool อาจมีส่วนที่ใช้ในการสร้างโปรแกรมขึ้นใช้งาน หรือที่เรียกว่า CASE (Computer-aided Software Engineering) ฯลฯ เป็นต้น

ทั้ง 2 ส่วนของระบบฐานข้อมูลนี้ จะถูกนำไปใช้บนเครื่องคอมพิวเตอร์ที่ต่างกัน เพื่อสนับสนุนการประมวลผลในแบบกระจาย (Distributed Processing) เช่น ระบบคอมพิวเตอร์ในแบบ Client/Server ที่ซึ่งเครื่องคอมพิวเตอร์แต่ละเครื่องถูกเชื่อมต่อกันด้วยระบบเครือข่าย ซึ่งส่งผลให้ข้อมูลชุดเดียวกันสามารถส่งถ่ายไปมาระหว่างเครื่องคอมพิวเตอร์เครื่องต่าง ๆ ที่เชื่อมต่อกันอยู่ภายในเครือข่ายนั้นได้

2.9.4 การเขียนโปรแกรมกับระบบ Client/Server

สิ่งที่โปรแกรมเมอร์ จะต้องคำนึงถึงในการพัฒนาโปรแกรมขึ้นใช้งาน ในระบบคอมพิวเตอร์แบบ Client/Server ได้แก่ การมอง Server เป็นเครื่องมือในการเข้าถึงข้อมูล และการมองข้อมูลในระดับ Record เนื่องจาก Request ที่ Client ส่งไปให้กับ Server เป็นคำสั่งที่ต้องการได้ผลลัพธ์ในระดับ Record และจะต้องอยู่ในรูปของคำสั่งในแบบ Non-procedural เช่น คำสั่ง SQL ดังนั้น จึงไม่มีการใช้คำสั่งที่ก่อให้เกิดการทำซ้ำ (Loop) หรือเป็นคำสั่งที่ใช้ในการแก้ไข หรือลบ ข้อมูล แต่เมื่อต้องการลดจำนวนของการติดต่อระหว่าง Client และ Server ให้นำเอา Stored Procedure มาใช้แทนคำสั่ง SQL (Stored Procedure เป็นคำสั่งที่เรียกใช้ข้อมูลจากฐานข้อมูลเช่นเดียวกับ SQL แต่สามารถนำมาเอาคำสั่งมารวมกันเป็นชุดคำสั่งได้ รวมทั้งจะถูกจัดเก็บ และทำงานอยู่บน Server และจะถูกเรียกใช้โดย Client)

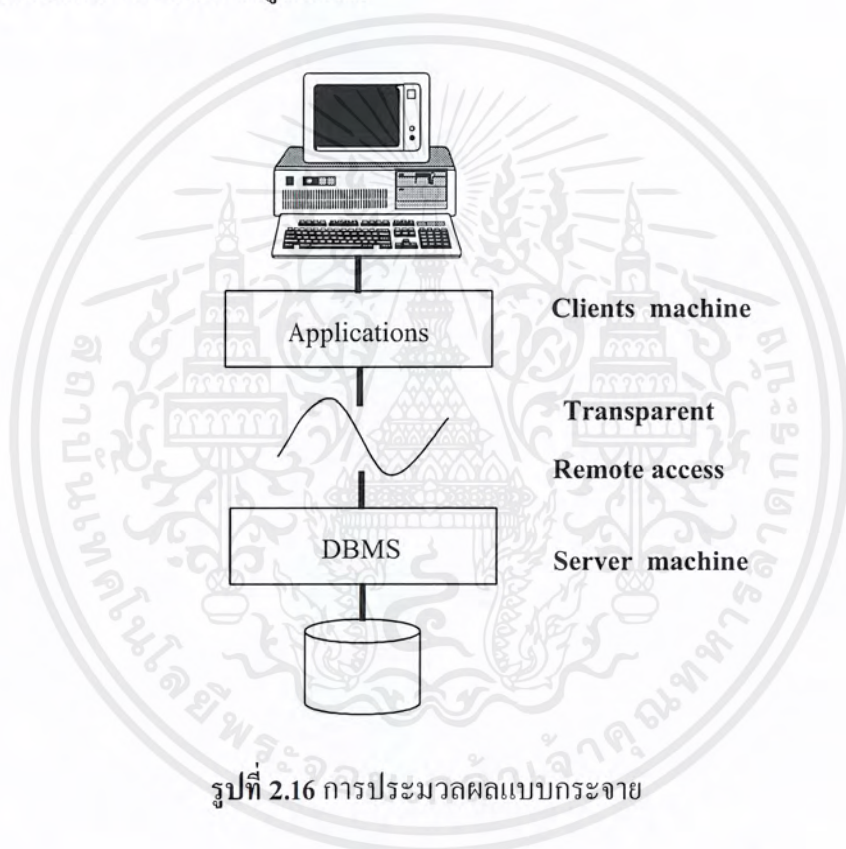
สำหรับข้อดีของการใช้ Stored procedure ประกอบด้วย

1. Stored Procedure สามารถปิดบังความซับซ้อนทางด้านการใช้งาน ส่งผลให้ผู้ใช้สามารถเรียกใช้ข้อมูลภายในฐานข้อมูลได้ง่าย ซึ่งเป็นไปตามคุณสมบัติทางด้าน Data Independence
2. แต่ละ Stored Procedure สามารถถูกเรียกใช้ร่วมกัน โดยหลาย ๆ Client
3. ผู้ใช้สามารถตรวจสอบความถูกต้องของ Stored Procedure ก่อนนำไปใช้งาน ซึ่งต่างจากการใช้คำสั่ง SQL ที่จะไม่สามารถตรวจสอบความถูกต้องก่อนนำไปใช้งานได้ เนื่องจาก เป็นคำสั่งที่อยู่ในรูปของ Request ที่ Client ส่งไปยัง Server ให้ทำงาน ขณะนั้น
4. Stored Procedure สามารถกำหนดความปลอดภัยให้กับข้อมูลได้ดีกว่าการใช้คำสั่ง SQL เนื่องจาก สามารถกำหนดผู้ที่มีสิทธิ์ในการใช้ Stored Procedure นั้นได้โดยตรง

แต่อย่างไรก็ตาม ข้อเสียของการใช้ Stored Procedure ได้แก่ ความแตกต่างในการใช้งาน Stored Procedure ของแต่ละผลิตภัณฑ์ เนื่องจาก ยังไม่มีการกำหนดมาตรฐานของ Stored Procedure

2.9.5 ระบบ Client/Server ที่สนับสนุนการประมวลผลแบบกระจาย

เนื่องจากการประมวลผลแบบกระจาย เป็นการนำเอาคอมพิวเตอร์หลาย ๆ เครื่อง มาเชื่อมต่อกันด้วยระบบเครือข่าย ด้วยจุดประสงค์ ที่ต้องการใช้ข้อมูลร่วมกัน ดังนั้นระบบฐานข้อมูลที่นำมาใช้กับการประมวลผลแบบกระจาย จึงต้องแบ่งออกเป็น ส่วน Server (โปรแกรม Backend) และ ส่วน Client (โปรแกรม Front-end) ไว้ในคอมพิวเตอร์ที่ต่างเครื่องกัน เพื่อที่จะให้คอมพิวเตอร์แต่ละเครื่องที่ทำหน้าที่เป็น Client สามารถเรียกใช้ข้อมูลจากฐานข้อมูลที่เกิดขึ้นในเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Server ได้ร่วมกัน ดังรูปที่ 2.16



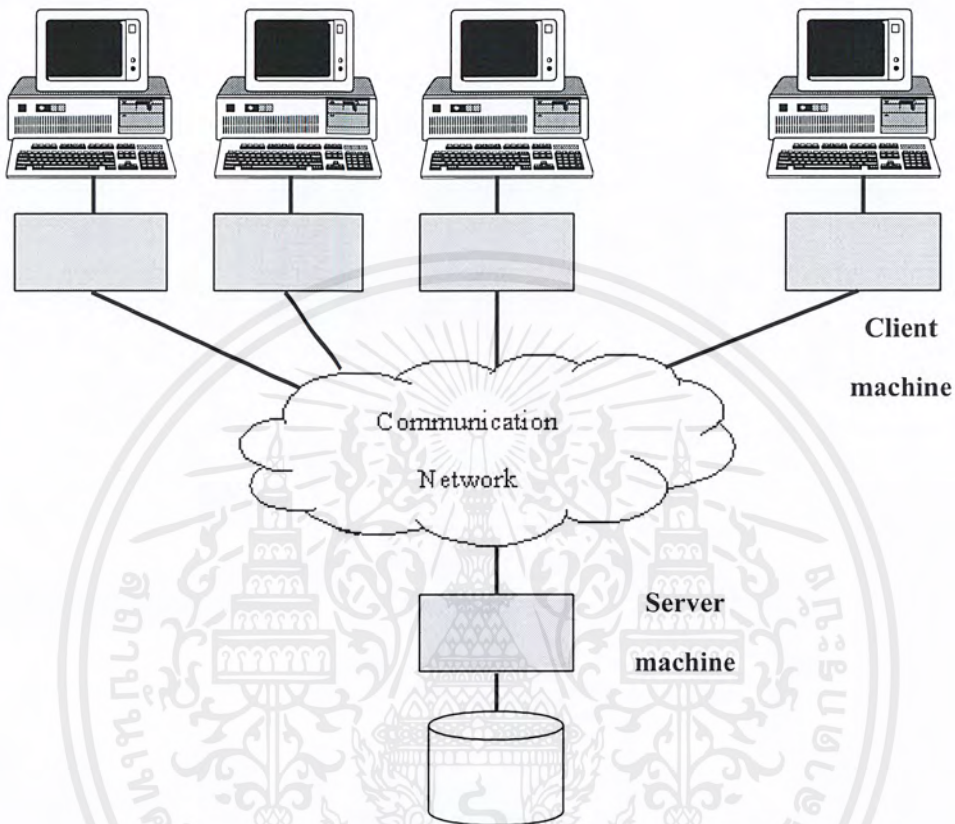
รูปที่ 2.16 การประมวลผลแบบกระจาย

การนำเอาระบบ Client/Server มาใช้ในการประมวลผลแบบกระจายนี้ จะส่งผลให้

1. การประมวลผลของ Client และ Server จะอยู่ในรูปแบบขนาน ดังนั้น เวลาในการตอบสนอง (Response Time) ของระบบที่มีต่อการทำงานจะใช้เวลาน้อยลง รวมทั้งปริมาณผลลัพธ์ที่ได้จากการประมวลผล (Throughput) จะได้ปริมาณที่มากขึ้นด้วย
2. ประสิทธิภาพการทำงานของ DBMS จะดีขึ้น โดยเฉพาะอย่างยิ่ง ในกรณีที่คอมพิวเตอร์ที่ใช้เป็น Server มีความสามารถที่สนับสนุนฟังก์ชันการทำงานต่าง ๆ ของ DBMS
3. การติดต่อกับผู้ใช้ (User Interface) จะมีความรวดเร็ว และมีประสิทธิภาพมากขึ้น เนื่องจากหน้าที่หลักของคอมพิวเตอร์ที่ทำหน้าที่เป็น Client ได้แก่ การแสดงผล

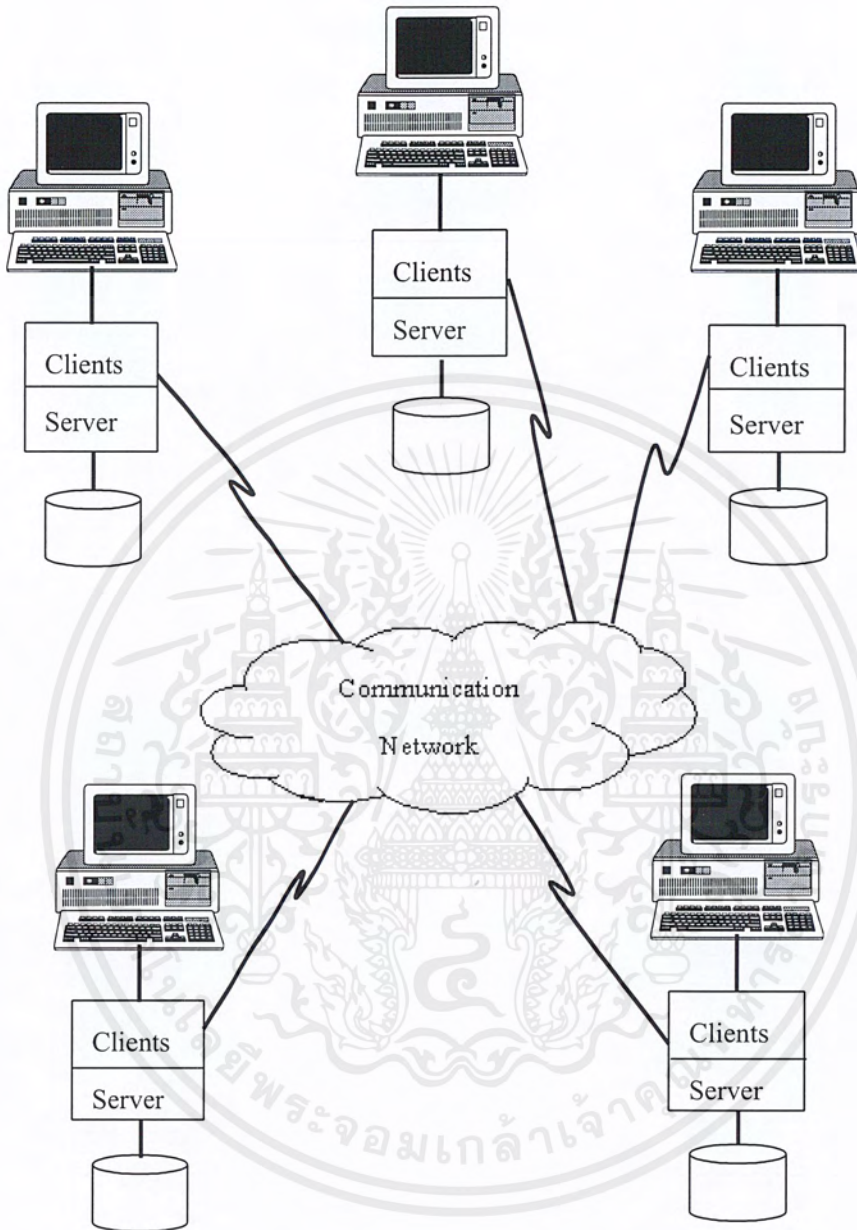
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. คอมพิวเตอร์ที่ทำหน้าที่เป็น Client สามารถใช้ข้อมูลร่วมกันได้ซึ่งสามารถแสดงด้วยแผนภาพได้ดังรูปที่ 2.17



รูปที่ 2.17 คอมพิวเตอร์ที่ทำหน้าที่เป็น Client

ในองค์กรที่มีขนาดใหญ่ ที่มีการจัดเก็บข้อมูลไว้ในหลาย ๆ ที่ เช่น ธนาคาร ที่มีการแยกจัดเก็บข้อมูลตามสาขาต่าง ๆ อาจมีการกำหนดให้คอมพิวเตอร์ที่ทำหน้าที่เป็น Client มีการจัดเก็บข้อมูลของตนเองไว้ และในขณะเดียวกันคอมพิวเตอร์ที่ทำหน้าที่เป็น Server ก็อาจมีโปรแกรมที่เป็นของตนเองเช่นเดียวกัน และด้วยรูปแบบในลักษณะนี้จะส่งผลให้ แต่ละเครื่องคอมพิวเตอร์ ทำหน้าที่ได้ทั้ง Server และ Client หรือกล่าวอีกนัยหนึ่ง คอมพิวเตอร์เครื่องหนึ่งสามารถทำหน้าที่เป็น Server เพื่อคอยส่งข้อมูลให้กับคอมพิวเตอร์เครื่องอื่น และในขณะเดียวกัน ก็สามารถทำหน้าที่เป็น Client เพื่อเรียกใช้ข้อมูลจากฐานข้อมูลในคอมพิวเตอร์เครื่องอื่นได้เช่นเดียวกัน สำหรับรูปแบบของระบบในลักษณะนี้ เรียกว่า Entire Database System ดังรูปที่ 2.18



รูปที่ 2.18 Entire Database System

ในกรณีที่มีการจัดเก็บฐานข้อมูลใน Server หลาย ๆ ตัว Client จะต้องมีความสามารถเพิ่มเติมในการเชื่อมต่อไปยัง Server แต่ละตัว เพื่อที่จะเรียกใช้ข้อมูลในฐานข้อมูลของแต่ละ Server ได้ แต่การเรียกใช้ข้อมูลในฐานข้อมูลของแต่ละ Server นี้ จะเกิดขึ้นได้เพียง Server เดียว ในแต่ละครั้งของการติดต่อระหว่าง Client และ Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบ การสร้าง และการทำงาน

3.1 กล่าวนำ

ในการทำธุรกิจหรือกิจการค้าต่างๆ จำเป็นที่จะต้องมีการวางแผนหรือการออกแบบเพื่อใช้เป็นแนวทางและจะได้ทราบถึงปัญหาที่เกิดขึ้น ก่อนที่จะมีการปฏิบัติจริง ทำให้เราสามารถหาแนวทางแก้ไขในการแก้ปัญหานั้นได้ทันท่วงที และมีประสิทธิภาพ

การสร้างฐานข้อมูลก็เช่นกัน เรามีความจำเป็นอย่างยิ่งที่จะต้องมีการออกแบบฐานข้อมูลเสียก่อน เพื่อให้ง่ายต่อการเขียนโปรแกรม ถ้าเราไม่มีการวางแผนหรือออกแบบก่อนก็อาจจะทำให้มีข้อผิดพลาดเกิดขึ้นได้ ดังนั้นเราจึงต้องมีการวิเคราะห์ระบบตารางเรียนเสียก่อนว่ามีขอบเขต และรายละเอียดอย่างไร

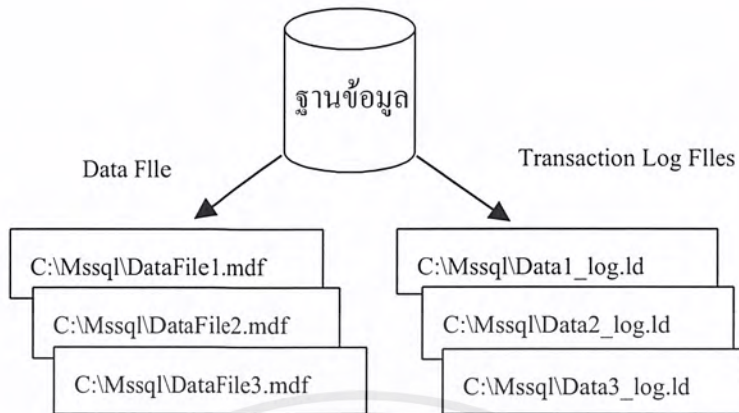
3.2 กำหนดโครงสร้างของระบบฐานข้อมูล

3.2.1 แนะนำเข้าสู่ระบบฐานข้อมูล

ก่อนที่จะเริ่มใช้งานเพื่อเก็บข้อมูลได้นั้น จะต้องมีการสร้าง และจัดการกับระบบฐานข้อมูลให้เรียบร้อยเสียก่อน ในการสร้างฐานข้อมูล (Database) หนึ่ง ๆ โดยปกติแล้วจะประกอบไปด้วยไฟล์อย่างน้อย 2 ไฟล์ คือ

- 1) ไฟล์ข้อมูล (Data File) เป็นไฟล์ที่เก็บข้อมูลจริง
- 2) ไฟล์ทรานแซคชันล็อก (Transaction Log File) เป็นไฟล์สำหรับเก็บ Operation log หรือข้อมูลล็อก (log) ซึ่งเกิดขึ้นจากการทำงานกับข้อมูลจริง

สำหรับ SQL Server 7.0 แล้ว สามารถสร้างไฟล์ข้อมูล หรือไฟล์ Transaction Log มีมากกว่า 1 ไฟล์ได้ ทั้งนี้เพื่อแยกการทำงานที่ชัดเจน และประโยชน์ในแง่ของการบำรุงรักษาอีกด้วย



รูปที่ 3.1 โครงสร้างระบบฐานข้อมูลของ SQL Server 7.0

จากรูปที่ 3.1 แสดงให้เห็นถึงโครงสร้างของระบบฐานข้อมูล ซึ่งประกอบด้วยไฟล์หลาย ๆ ไฟล์ ทั้งในส่วนที่เป็นไฟล์ข้อมูล และไฟล์ Transaction Log โดยในแต่ละไฟล์สามารถเก็บอยู่ต่าง พาร์ท หรือแม้แต่ต่าง Drive กันได้ ซึ่งช่วยให้การเก็บข้อมูลสามารถทำได้หลายที่ และไม่ต้องกังวล กับขนาดเนื้อที่ว่างของ Drive ที่มีอยู่อย่างจำกัด รวมทั้งช่วยเพิ่มประสิทธิภาพในการเข้าถึงข้อมูลให้ดีขึ้นอีกด้วย

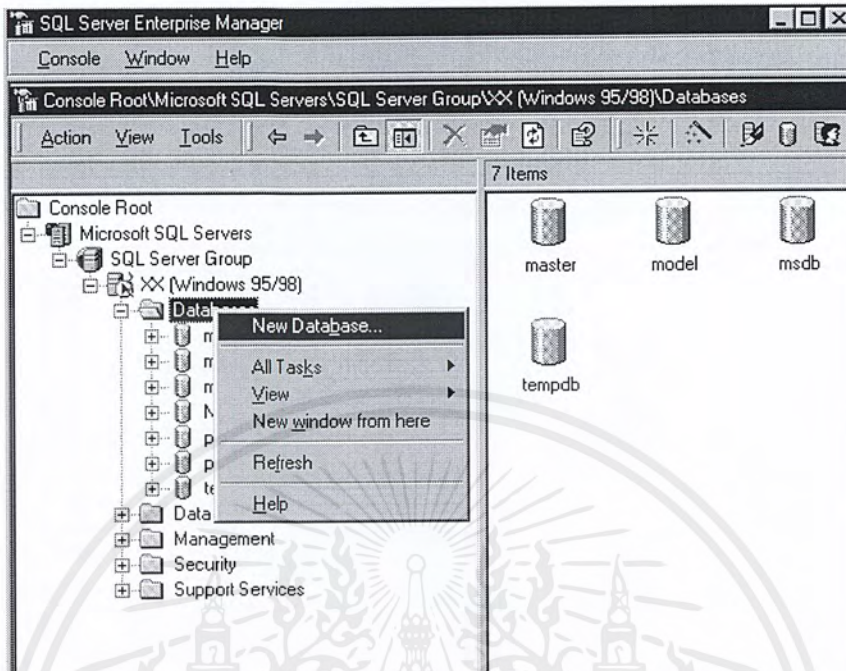
3.2.2 การจัดการระบบฐานข้อมูล

SQL Server 7.0 มีวิธีการจัดการระบบฐานข้อมูล ทั้งการสร้าง, เปลี่ยนแปลง หรือทำลาย รวมถึงการกำหนด Option ต่าง ๆ ของฐานข้อมูล โดยใช้เครื่องมือสำหรับบริหารงานอย่าง Enterprise Manager ซึ่งง่ายต่อการใช้งานมากกว่าการใช้คำสั่งของ Transact – SQL

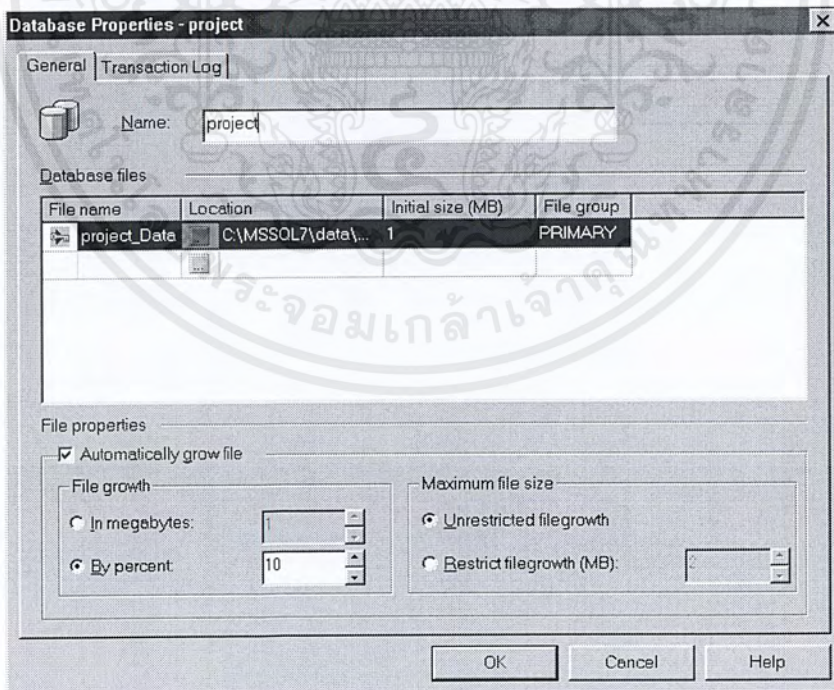
3.2.3 การสร้างฐานข้อมูล

สำหรับการสร้างฐานข้อมูลโดยใช้ Enterprise Manager มีขั้นตอน ดังนี้

- 1) ทิศต่อ SQL Server ด้วย Enterprise Manager
- 2) ขยายโพลเดอร์ไปยัง Database
- 3) คลิกขวายังพื้นที่ว่างของ Fram ขวา พร้อมทั้งเลือก New Database ดังรูปที่ 3.2 หรือเลือกที่เมนู Action > New Database ซึ่งทำให้ได้อะด็อกสำหรับการสร้างข้อมูลปรากฏขึ้น ดังรูปที่ 3.3

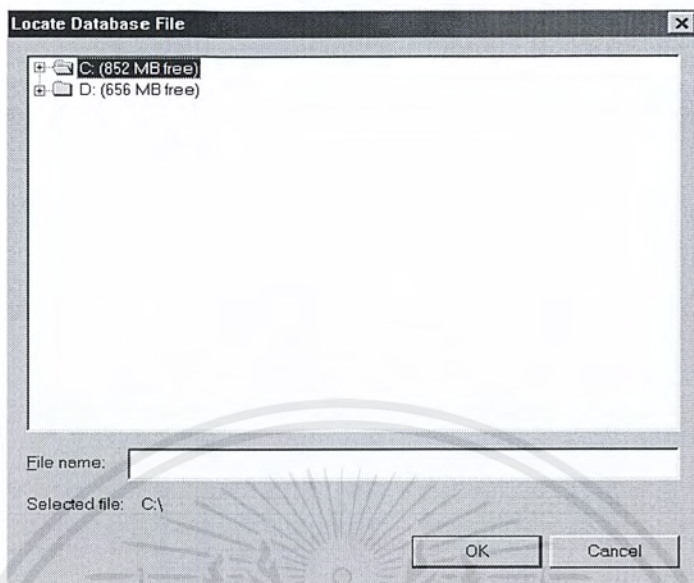


รูปที่ 3.2 การเลือกเมนูเพื่อสร้างฐานข้อมูล



รูปที่ 3.3 การสร้างฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 การเปลี่ยน part ที่เก็บฐานข้อมูล

4) ที่แท็บ General ให้ใส่ชื่อของฐานข้อมูลที่ต้องการสร้างยังช่อง Name: ซึ่งจะมีผลทำให้การตั้งชื่อไฟล์ของฐานข้อมูล ขึ้นต้นด้วยชื่อที่ระบุนี้แล้วตามด้วย _Data และมีนามสกุลเป็น .MDF โดยอัตโนมัติ หากไม่เหมาะสมก็สามารถเปลี่ยนได้ตามต้องการ

5) นอกจากนี้ในส่วนของ Database files สามารถเปลี่ยนตำแหน่งที่เก็บไฟล์ของฐานข้อมูลได้ ปกติจะอยู่ภายใต้ไดเรกทอรี \MSSQL7\Data โดยการคลิกที่ปุ่ม [...] ซึ่งจะได้ไดอะล็อกบ็อกซ์สำหรับเลือกเปลี่ยนพาทที่ต้องการ ดังรูปที่ 3.4

6) กำหนดขนาดเริ่มต้นของฐานข้อมูลที่ช่อง Initial size (MB) ซึ่งมีค่าดีฟอลต์เป็น 1 MB ส่วนช่อง File group สำหรับสร้างฐานข้อมูลให้เป็นแบบไฟล์กรุป (Filegroups)

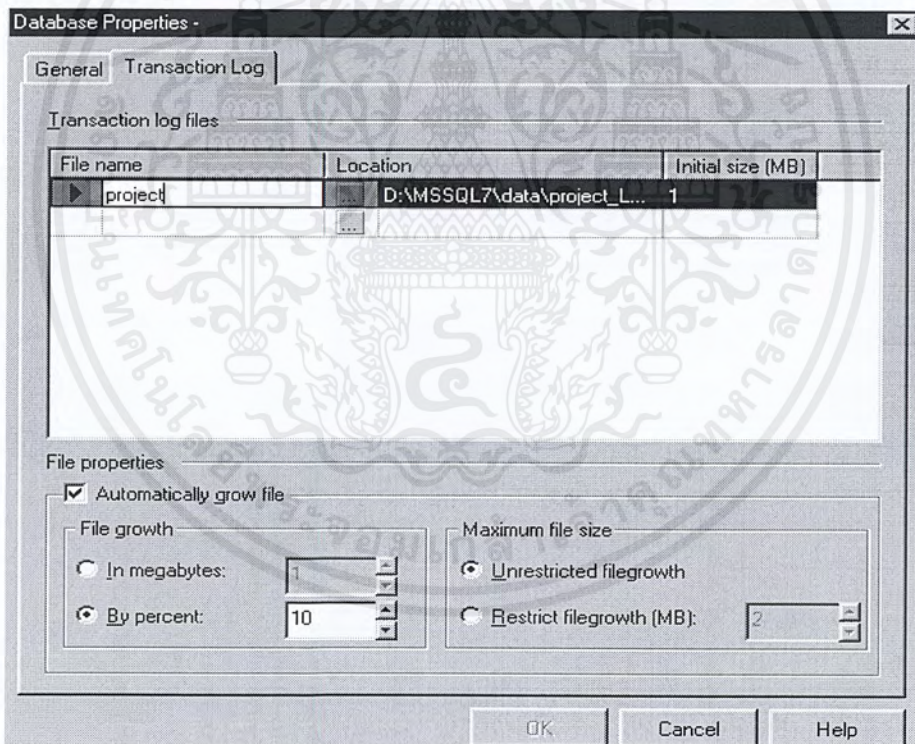
7) ในส่วนของ File properties เช็คบ็อกซ์ที่ Automatically grow file ได้ถูกเลือกไว้โดยดีฟอลต์ ซึ่งทำให้สามารถกำหนดลักษณะการเพิ่มขนาดของฐานข้อมูล เมื่อฐานข้อมูลเกิดเต็มขึ้นมาได้ ซึ่งแบ่งออกเป็น

File growth สำหรับระบุการเพิ่มในแต่ละครั้ง ซึ่งสามารถเลือกได้เป็น In megabytes: คือเพิ่มตามจำนวนที่กำหนดไว้ หน่วยเป็น MB และ By percent เพิ่มขึ้นเป็นจำนวนเปอร์เซ็นต์จากขนาดของฐานข้อมูลเดิม

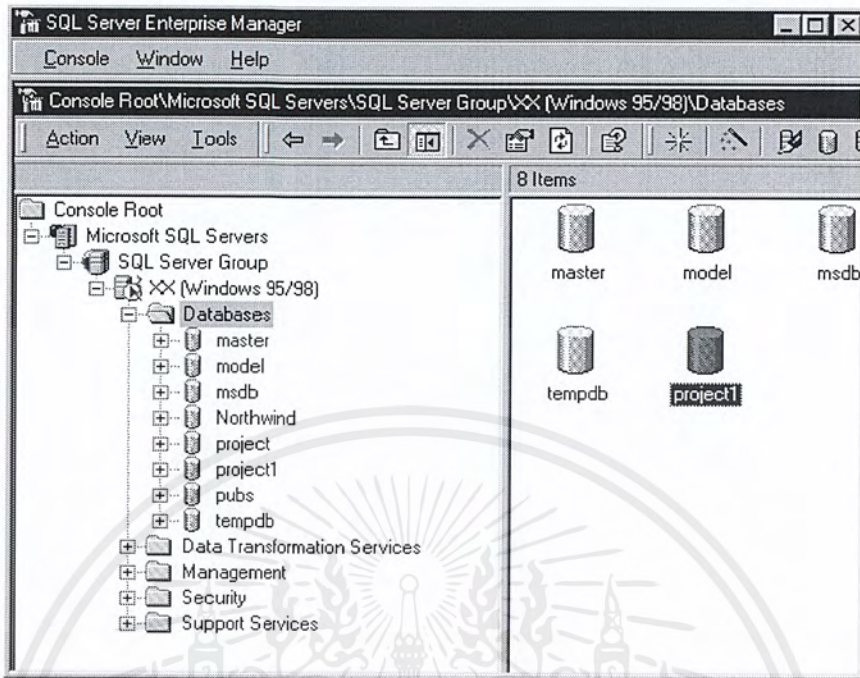
Maximum file size สำหรับกำหนดขนาดของฐานข้อมูลสูงสุดที่สามารถเป็นไปได้ทั้งสิ้น ซึ่งสามารถเลือกได้เป็น Unrestricted filegrowth คือไม่จำกัดขนาด หรือจนกว่าเนื้อที่ว่างบนดิสก์จะเต็ม และRestricted filegrowth (MB) : เพื่อจำกัดขนาดของฐานข้อมูลที่สามารถเป็นไปได้สูงสุด

8) เลือกที่แท็บ Transaction Log จะพบว่าลักษณะการกำหนดชื่อ และตำแหน่งมีลักษณะคล้ายตามการกำหนดฐานข้อมูลอย่างอัตโนมัติ แสดงดังรูปที่ 3.5 โดยขึ้นต้นด้วยชื่อของฐานข้อมูลตามด้วย _Log และมีนามสกุลเป็น .LDF หากไม่เหมาะสมสามารถเปลี่ยนชื่อรวมทั้งตำแหน่งของพาธที่เก็บได้ตามต้องการ

9) เมื่อกำหนดข้อมูลเสร็จให้คลิกปุ่ม OK เป็นอันเสร็จสิ้นขั้นตอนของการสร้างฐานข้อมูล ซึ่งจะได้สัญลักษณ์รูปทรงกระบอกสี่เหลี่ยม แสดงถึงฐานข้อมูลที่เพิ่มสร้างปรากฏขึ้น ดังแสดงในรูปที่ 3.6



รูปที่ 3.5 การกำหนดชื่อ และตำแหน่งที่มีลักษณะคล้ายตามการกำหนดฐานข้อมูลอย่างอัตโนมัติ



รูปที่ 3.6 ฐานข้อมูลที่สร้างปรากฏขึ้น

3.2.4 การออกแบบฐานข้อมูล

ในการออกแบบฐานข้อมูลควรจะมีการวางแผน และใช้กฎการ Normalization เข้ามาช่วย เพื่อลดความซ้ำซ้อนของข้อมูล มีการกำหนดความสัมพันธ์ระหว่างเอนทิตีของแต่ละตารางให้เหมาะสม และจัดแบ่งประเภทของข้อมูลให้ถูกต้อง หลักการพื้นฐานสำหรับออกแบบฐานข้อมูลมีดังนี้

- 1) รวบรวมข้อมูลที่เกี่ยวข้องกับตารางนั้น ๆ ให้มากที่สุด เช่น ตาราง Customers ควรจะมีข้อมูลที่เกี่ยวข้องกับชื่อ-นามสกุล ที่อยู่ วัน/เดือน/ปี เบอร์โทรศัพท์ ฯลฯ ตาราง Products มีข้อมูลเกี่ยวกับรหัสสินค้า จำนวน ราคา ฯลฯ
- 2) ทำการ Normalization เพื่อให้เกิดความซ้ำซ้อนน้อยที่สุด เนื่องจากในตารางจะประกอบไปด้วยฟิลด์ข้อมูลหลายฟิลด์
- 3) กำหนดคีย์หลัก (Primary Key) และคีย์นอก (Foreign Key) สำหรับตารางที่เกี่ยวข้อง หรือมีความสัมพันธ์กัน

4) ให้กำหนดประเภทของข้อมูลอย่างเหมาะสมกับแต่ละฟิลด์ เพราะประเภทข้อมูลจะมีผลต่อการทำงานโดยตรง เช่น ฟิลด์ Name ควรเป็น Character ฟิลด์ Salary ควรเป็น Numeric ฟิลด์ Order_date ควรเป็น Datetime

3.2.5 การสร้างตาราง

ฐานข้อมูลเชิงสัมพันธ์จะเก็บข้อมูลเอาไว้ในรูปของตาราง (Table) ซึ่งทุกตารางสามารถที่จะเชื่อมโยงข้อมูลระหว่างกันได้ ในการสร้างตารางด้วยภาษา SQL จะใช้ส่วนภาษานิยามข้อมูล (Data Definition Language : DDL) เพื่อกำหนดโครงสร้างของข้อมูล รูปแบบคอลัมน์ชนิดของข้อมูล ก่อนที่จะเริ่มสร้างตารางควรจะออกแบบตารางคร่าว ๆ บนแผ่นกระดาษ โดยเขียนชื่อคอลัมน์ชนิดข้อมูล ความกว้าง กำหนดคีย์หลัก และความสัมพันธ์ระหว่างตารางให้ถูกต้อง

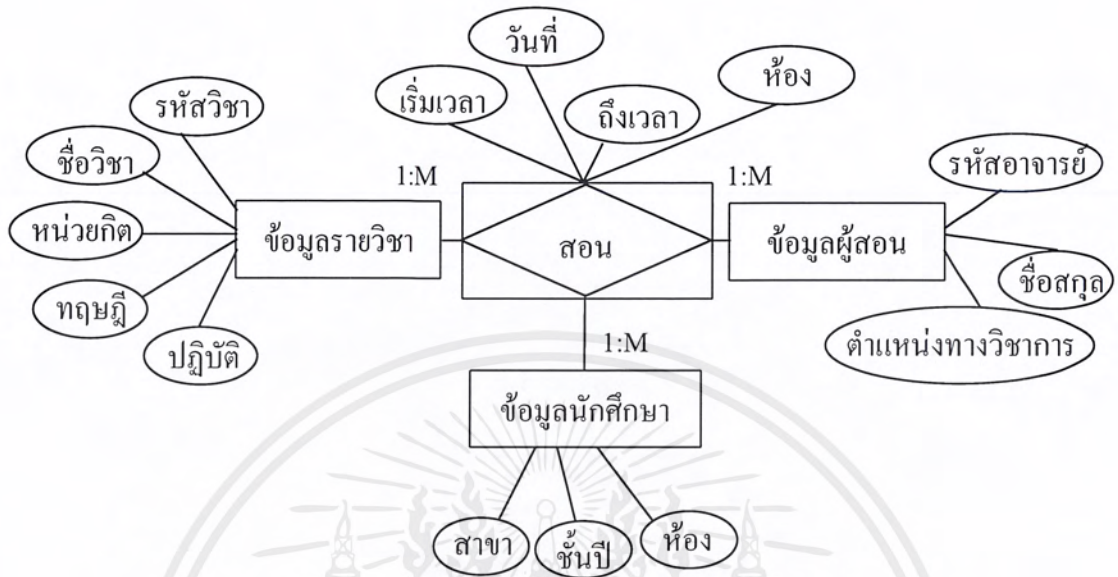
3.2.6 การออกแบบโครงสร้างของตาราง

ในการออกแบบโครงสร้างของตารางจะต้องใช้ทฤษฎีและความเป็นจริงร่วมกัน เพื่อให้ได้ตารางข้อมูลที่มีประสิทธิภาพ สามารถเปลี่ยนแปลงแก้ไขได้ในอนาคตและไม่กระทบกระเทือนกับส่วนอื่น ที่สำคัญจะต้องทราบวัตถุประสงค์ในการทำงานว่า ตารางนี้จะใช้เก็บข้อมูลเกี่ยวกับอะไร? เช่น ประวัตินักศึกษา ผลสอบแต่ละวิชา ประวัติพนักงาน ข้อมูลการขายสินค้า เป็นต้น ฯลฯ

จากการศึกษาถึงขอบเขต และหน้าที่ของระบบงาน ทำให้เราได้ทราบถึงโครงสร้างของระบบฐานข้อมูล ดังนั้น โปรแกรมจะมีองค์ประกอบอยู่ 4 ส่วน ได้แก่

- 1) ข้อมูลรายวิชา
- 2) ข้อมูลผู้สอน
- 3) ข้อมูลนักศึกษา
- 4) ข้อมูลการสอน

ซึ่งทั้ง 4 ส่วนมีความสัมพันธ์กันดังรูปที่ 3.7



รูปที่ 3.7 ความสัมพันธ์ระหว่างข้อมูลทั้ง 3 ส่วน

รายละเอียดของข้อมูลทั้ง 4 ส่วนมีดังนี้

1) ข้อมูลรายวิชา

ระบบฐานข้อมูลรายวิชาประกอบด้วยข้อมูลต่าง ๆ ดังนี้

1. รหัสวิชา
2. ชื่อวิชา
3. หน่วยกิต
4. ทฤษฎี
5. ปฏิบัติ

2) ข้อมูลผู้สอน

ระบบฐานข้อมูลผู้สอนประกอบด้วยข้อมูลต่าง ๆ ดังนี้

1. รหัสอาจารย์
2. ชื่อ
3. นามสกุล
4. ตำแหน่งทางวิชาการ

3) ข้อมูลนักศึกษา

ระบบฐานข้อมูลนักศึกษาประกอบด้วยข้อมูลต่าง ๆ ดังนี้

- 1.1) รหัสสาขา
- 1.2) ชั้นปี
- 1.3) ห้องเรียน

4) ข้อมูลการสอน

ระบบฐานข้อมูลการสอนประกอบด้วยข้อมูลต่าง ๆ ดังนี้

- 1.1) วันที่
- 1.2) ห้อง
- 1.3) เริ่มเวลา
- 1.4) เวลาสิ้นสุด

3.3 กำหนดโครงสร้าง และรายการภายในโปรแกรม

โปรแกรมระบบฐานข้อมูล จะมีโครงสร้าง และรายการภายใน โปรแกรมดังรูปที่ 3.8



รูปที่ 3.8 โครงสร้าง และรายการภายในโปรแกรม

3.4 ข้อมูลระบบงาน

จากการศึกษาขอบเขตของตารางเรียน คุณสมบัติของผลลัพธ์ที่ต้องการ และจากการรวบรวมข้อมูลทั้งหมด ได้จัดทำอยู่ในรูปพจนานุกรมข้อมูลดังนี้

ตารางที่ 3.1 พจนานุกรมข้อมูล

เพิ่มข้อมูล		ข้อมูล				
ชื่อ	ความหมาย	ชื่อ	ความหมาย	ชนิด	ยาว	ทศนิยม
student	ข้อมูลนักศึกษา	BranchID	รหัสสาขา	C	10	
		Year	ชั้นปี	C	2	
		Classroom	ห้องเรียน	C	2	
teacher	ข้อมูลอาจารย์	TeacherID	รหัสอาจารย์	N	4	
		Name	ชื่อ	C	50	
		Surname	นามสกุล	C	50	
		PositionID	ตำแหน่งทางวิชาการ	C	30	
subject	วิชา	SubjectID	รหัสวิชา	N	10	
		SubjectName	ชื่อวิชา	C	50	
		Credit	หน่วยกิต	C	1	
		TheoryHour	ทฤษฎี	C	1	
		PracticeHour	ปฏิบัติ	C	1	
teach	ตารางสอน	TeacherID	รหัสอาจารย์	N	4	
		Branch	สาขา	C	10	
		Date	วัน	N	10	
		Time	เวลา	N	15	
		Room	ห้อง	N	20	
RefStudent	สาขานักเรียน	BranchID	รหัสสาขา	C	10	
		BranchName	ชื่อสาขา	C	30	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 แสดงชนิดข้อมูลของ SQL Server

ชนิดของข้อมูล	ความหมาย
Binary	เป็นข้อมูลแบบไบนารี fixed-length มีค่าสูงสุด 8,000 ไบต์
Varbinary	เป็นข้อมูลแบบไบนารี variable-length มีค่าสูงสุด 8,000 ไบต์
char	เป็นข้อมูลประเภทตัวอักษร นำมาคำนวณไม่ได้ และเป็น fixed-length มีค่าสูงสุด 8,000 ไบต์
varchar	เป็นข้อมูลประเภทตัวอักษรแบบ variable-length มีค่าสูงสุด 8,000 ไบต์
nchar	เป็นข้อมูลประเภทตัวอักษรที่ใช้เก็บ Unicode Characters ขนาด 2 ไบต์ ต่อ 1 character nchar เป็นข้อมูลแบบ fixed-length มีค่าสูงสุด 4,000 ไบต์
nvarchar	เป็นข้อมูลประเภทตัวอักษรที่ใช้เก็บ Unicode Characters ขนาด 2 ไบต์ ต่อ 1 character nvarchar เป็นข้อมูลแบบ variable-length มีค่าสูงสุด 4,000 ไบต์
datetime	เป็นข้อมูลเกี่ยวกับ วัน เดือน ปี จะมีค่าเริ่มตั้งแต่ 1 มกราคม 1753 ถึง 31 ธันวาคม 9999 มีขนาด 8 ไบต์
smalldatetime	เป็นข้อมูลเกี่ยวกับ วัน เดือน ปี จะมีค่าเล็กกว่าโดยจะเริ่มตั้งแต่ 1 มกราคม 1900 ถึง 6 มิถุนายน 2079 มีขนาด 4 ไบต์
decimal	เป็นข้อมูลประเภทตัวเลข นำมาคำนวณได้ มีค่าตั้งแต่ $-10^{38}-1$ ถึง $10^{38}-1$
numeric	เป็นข้อมูลประเภทตัวเลข นำมาคำนวณได้ มีค่าตั้งแต่ $-10^{38}-1$ ถึง $+10^{38}-1$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

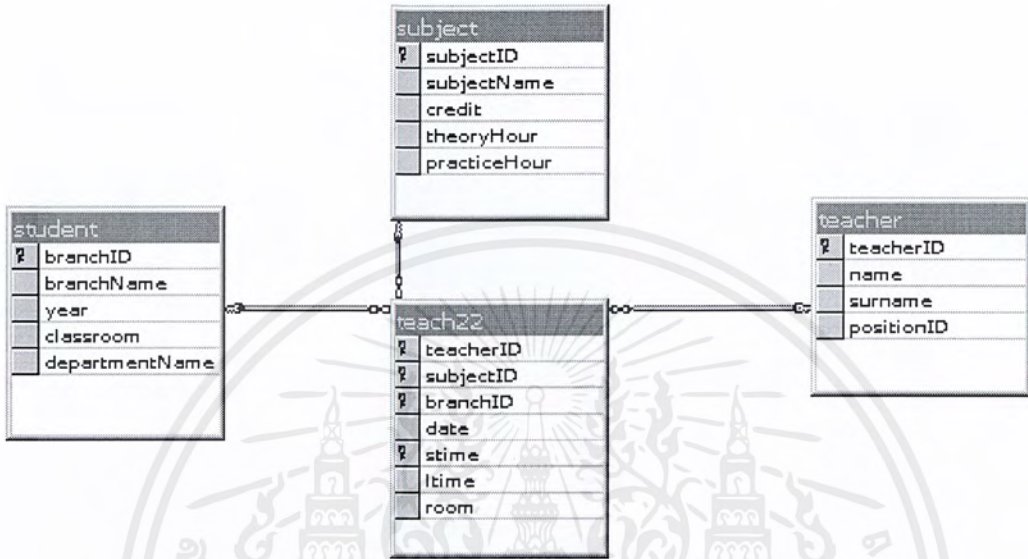
ตารางที่ 3.2 (ต่อ) แสดงชนิดข้อมูลของ SQL Server

ชนิดของข้อมูล	ความหมาย
int	เป็นข้อมูลประเภทเลขจำนวนเต็มสามารถนำมาคำนวณได้มีค่าตั้งแต่ -2^{31} (-2, 147, 483, 648) ถึง $2^{31}-1$ (2, 147, 483, 647)
smallint	เป็นข้อมูลประเภทเลขจำนวนเต็ม แต่มีขนาดเล็กกว่า int 0 มีค่าตั้งแต่ -2^{15} (-32, 768) ถึง $2^{15}-1$ (32, 767)
tinyint	เป็นข้อมูลประเภทเลขจำนวนเต็มเช่นกัน มีขนาดตั้งแต่ 0 ถึง 255
float	เป็นข้อมูลประเภททศกกำลัง ใช้เก็บตัวเลขทศกกำลัง ตัวเลขวิทยาศาสตร์ มีค่าบวกตั้งแต่ $2.23E-308$ ถึง $-1.79E+308$ และค่าลบตั้งแต่ $-2.23E-308$ ถึง $-1.79E+308$
real	เป็นข้อมูลประเภทเดียวกับ float แต่มีขนาดเพียง 7 ดิจิต มีค่าบวกตั้งแต่ $1.18E-38$ ถึง $3.40E+38$ และมีค่าลบตั้งแต่ $-1.18E$ ถึง $-1.79E+38$
money	เป็นข้อมูลเกี่ยวกับการเงิน ประเภทเงินตรา จะมีค่าตั้งแต่ -922, 337, 203, 685, 477.5707 ถึง 922, 337, 203, 685, 477.5807
smallmoney	เป็นข้อมูลเกี่ยวกับเงินเหมือน money แต่จะมีขนาดเล็กกว่า คือ -214, 748.3648 ถึง +214, 748.3647
text	เป็นข้อมูลประเภทเท็กซ์ สามารถจะเก็บ character ได้ $2^{31}-1$ หรือ 2, 147, 483, 647 ตัว
ntext	เป็นข้อมูลประเภทเท็กซ์ สามารถจะเก็บ character ได้ $2^{30}-1$ หรือ 1, 073, 741, 823
Image	เป็นข้อมูลแบบอิมเมจ เก็บข้อมูลได้ $2^{31}-1$ หรือ 2, 147, 483, 647

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ความสัมพันธ์ของเพิ่มข้อมูล

ความสัมพันธ์ของเพิ่มข้อมูลภายในระบบงานจะมีความสัมพันธ์กัน ดังรูปที่ 3.12



รูปที่ 3.12 ความสัมพันธ์ของเพิ่มข้อมูล

จากรูปที่ 3.12 ตาราง refStudent จะมีความสัมพันธ์กับตาราง student ในลักษณะ **One-To-One** คือ เป็นความสัมพันธ์แบบที่ฟิลด์ในเรคอร์ดใดๆ ของ Table หนึ่งมีค่าตรงกับฟิลด์ของเรคอร์ดเพียงเรคอร์ดเดียวใน Table อื่น โดยที่มี branchID เป็น primary key ของทั้งสองตารางจึงมีความสัมพันธ์ ตามลักษณะดังกล่าว

ตาราง student ยังมีความสัมพันธ์กับตาราง teach ในลักษณะของ **One-To-Many** เนื่องจากตาราง student มี branchID เป็น primary key และในตารางของ teach มี branch เป็น forien key

ตาราง teach มี subjectID เป็น forien key จึงมีความสัมพันธ์กับตาราง subject ในลักษณะของ **One-To-Many** โดยมี subjectID เป็น primary key

ตาราง teacher มี teacherID เป็น primary key มีความสัมพันธ์กับตาราง teach มี teacherID เป็น forien key เป็นความสัมพันธ์แบบ **One-To-Many** คือที่ฟิลด์ในเรคอร์ดใดๆ ของ Table หนึ่งมีค่าตรงกับฟิลด์ของหลายๆ เรคอร์ดใน Table อื่น

3.6 การออกแบบโปรแกรมด้วย Visual Basic

3.6.1 การทำงานของ Data Environment

การทำงานของ Data Environment นั้น จะเป็นตามขั้นตอนต่อไปนี้

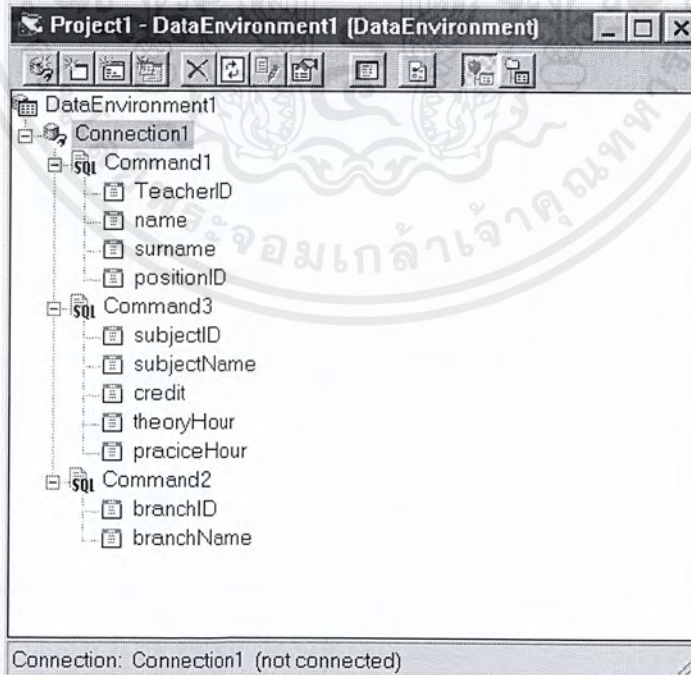
1. สร้าง Object Connection ขึ้นมาก่อน เพื่อเชื่อมต่อกับฐานข้อมูลที่เราจะทำงานด้วย ซึ่งเราจะต้องกำหนดคุณสมบัติต่างๆ ที่จำเป็น เช่น ชื่อฐานข้อมูล ชื่อผู้ใช้ และรหัสผ่าน เป็นต้น

2. สร้าง Object Command เพื่อดึงข้อมูลจาก Object Connection ที่เชื่อมต่อกับฐานข้อมูลที่เราต้องการผลลัพธ์

1) การเชื่อมต่อกับฐานข้อมูลด้วย Object Connection

ก่อนที่เราจะเข้าถึงฐานข้อมูลด้วย Data Environment นั้น เราต้องสร้างและกำหนดคุณสมบัติให้กับ Object Connection ในการเชื่อมต่อกับฐานข้อมูล ซึ่ง Object Connection 1 ตัวสามารถเชื่อมต่อกับฐานข้อมูลได้เพียงตัวเดียวเท่านั้น ถ้าเราต้องการเชื่อมต่อกับฐานข้อมูลตัวอื่น ๆ อีก เราต้องสร้าง Object Connection เพิ่มเพื่อทำการเชื่อมต่อ ซึ่งทำให้ Data Environment แต่ละตัวจะต้องมี Object Connection อยู่อย่างน้อย 1 Object เพื่อเชื่อมต่อกับฐานข้อมูลอย่างน้อย 1 ฐานข้อมูล

ในการเพิ่ม Object Connection นั้น เราจะเพิ่มได้โดยการ Click mouse ปุ่ม Add Connection ดังรูปที่ 3.13



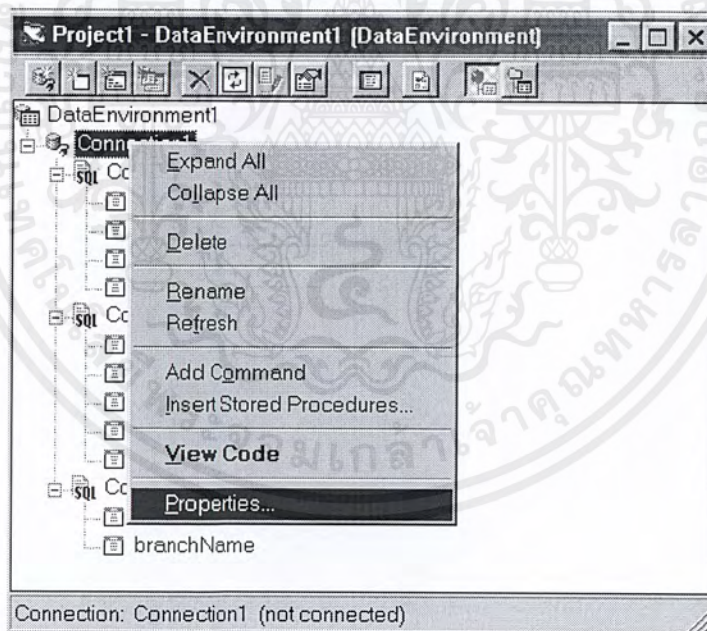
รูปที่ 3.13 การเพิ่ม Object Connection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

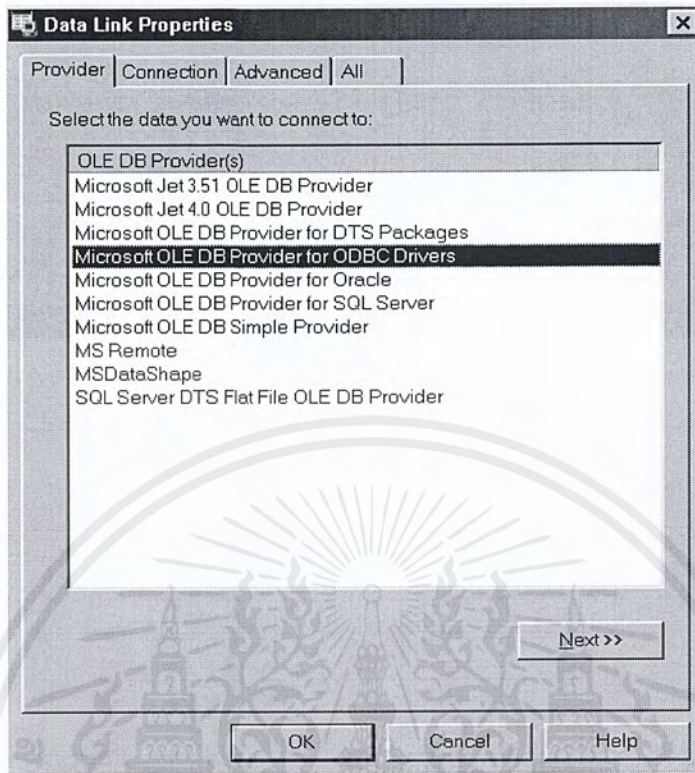
2) การกำหนดคุณสมบัติการเชื่อมต่อให้กับ Object Connection

เมื่อเราเพิ่ม Object Connection แล้ว เราต้องกำหนดให้ Object Connection เชื่อมต่อไปยังฐานข้อมูลก่อนเป็นขั้นแรก ด้วยกำหนดคุณสมบัติของ Object Connection ให้เชื่อมไปยังฐานข้อมูล Order.MDB ให้เราทำตามขั้นตอนต่อไปนี้

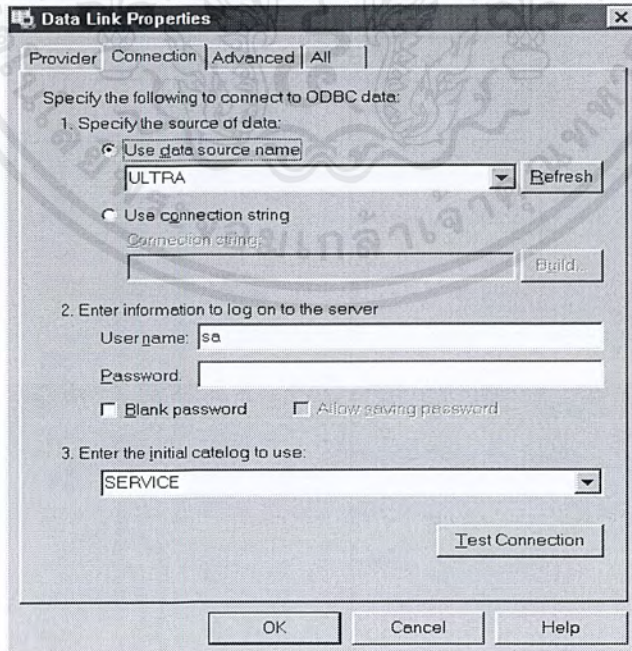
1. Click mouse ปุ่มขวาที่ Object Connection แล้วเลือกเมนู Properties ดังรูปที่ 3.14
2. จะปรากฏไดอะล็อกซ์ Data Link Properties ให้เราเลือกแท็บ Provider แล้วเลือก Microsoft OLE DB Provider for ODBC Drivers เพื่อเชื่อมต่อกับไฟล์ฐานข้อมูลชนิด Access ดังรูปที่ 3.15
3. เลือกแท็บ Connection แล้วให้เรากำหนดตำแหน่ง และชื่อไฟล์ฐานข้อมูลที่จะติดต่อด้วยพร้อมทั้งชื่อผู้ใช้ และรหัสผ่าน ดังรูปที่ 3.16
4. เสร็จแล้ว Click mouse ปุ่ม OK ซึ่งตอนนี้โปรแกรมเราก็สามารถติดต่อกับฐานข้อมูลได้แล้ว



รูปที่ 3.14 หน้าต่างของเมนู Properties



รูปที่ 3.15 การเลือก Microsoft OLE DB Provider for ODBC Drivers เพื่อติดต่อกับฐานข้อมูล



รูปที่ 3.16 การกำหนดตำแหน่งและชื่อไฟล์ฐานข้อมูลพร้อมทั้งชื่อรหัสผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

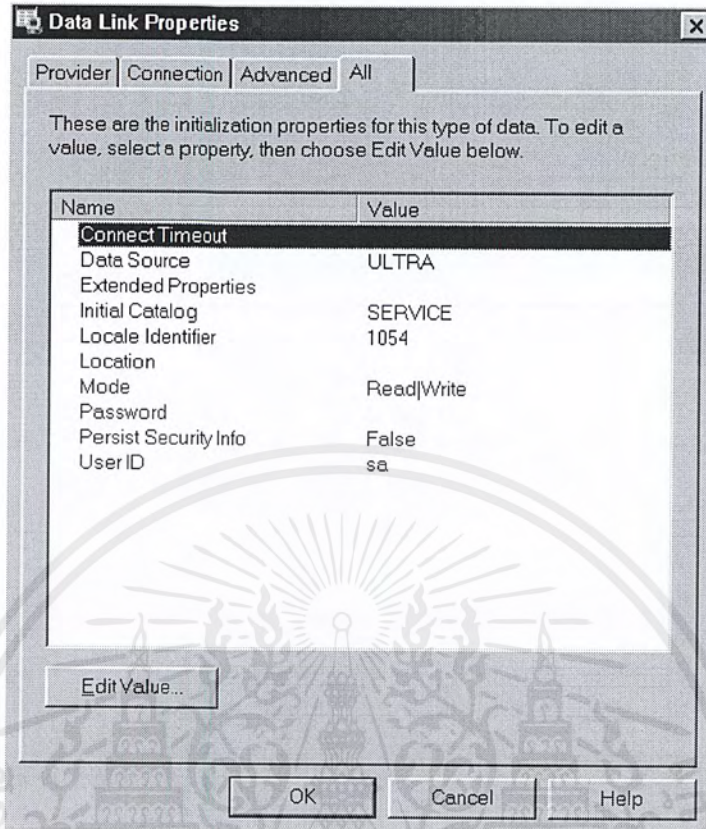
3) การกำหนดข้อมูลในการเชื่อมต่อเพิ่มเติม

นอกจากการกำหนดชื่อฐานข้อมูล ชื่อผู้ใช้ และรหัสผ่านแล้ว เรายังสามารถกำหนดคุณสมบัติที่สำคัญเพิ่มเติมได้ ซึ่งคุณสมบัติต่าง ๆ ที่เพิ่มเติมขึ้นมาจะมีความหมายดังต่อไปนี้

ตารางที่ 3.3 ข้อมูลในการเชื่อมต่อเพิ่มเติม

ชื่อคุณสมบัติ	คำอธิบาย
DesignUserName, DesignPassword	เป็นชื่อ และรหัสผ่านเพื่อเข้าใช้งานฐานข้อมูล ในขณะที่ ออกแบบโปรแกรม
RunUserName, RunPassword	เป็นชื่อ และรหัสผ่านเพื่อเข้าใช้งานฐานข้อมูลในขณะที่รัน โปรแกรม
DesignSaveAuthentication	ถ้าค่าคุณสมบัตินี้เป็น True จะมีการบันทึกค่าในคุณ สมบัติ DesignUserName กับ DesignPassword เก็บไว้ใน Project ด้วย
RunSaveAuthentication	ถ้าค่าคุณสมบัตินี้เป็น True จะมีการบันทึกค่าในคุณ สมบัติ RunUserName กับ RunPassword เก็บไว้ใน Project ด้วย
CursorLocation	ใช้กำหนดว่า จะให้เราใช้เครื่อง Client หรือ Server ใน การเก็บข้อมูลผลลัพธ์ของคิวรี
ConnectionTimeout	กำหนดเวลาการเชื่อมต่อกับฐานข้อมูลของ Object Connection ถ้าเกินจากนี้ แสดงว่าเกิดข้อผิดพลาดขึ้น
CommandTimeout	กำหนดเวลาในการหาผลลัพธ์ของ Object Command ถ้าเกินจากนี้ แสดงว่าเกิดข้อผิดพลาดขึ้น

นอกจากนี้ เรายังสามารถกำหนดคุณสมบัติเพิ่มเติมให้กับ Object Connection ได้ในแท็บ
Advanced กับ All ดังรูปที่ 3.17



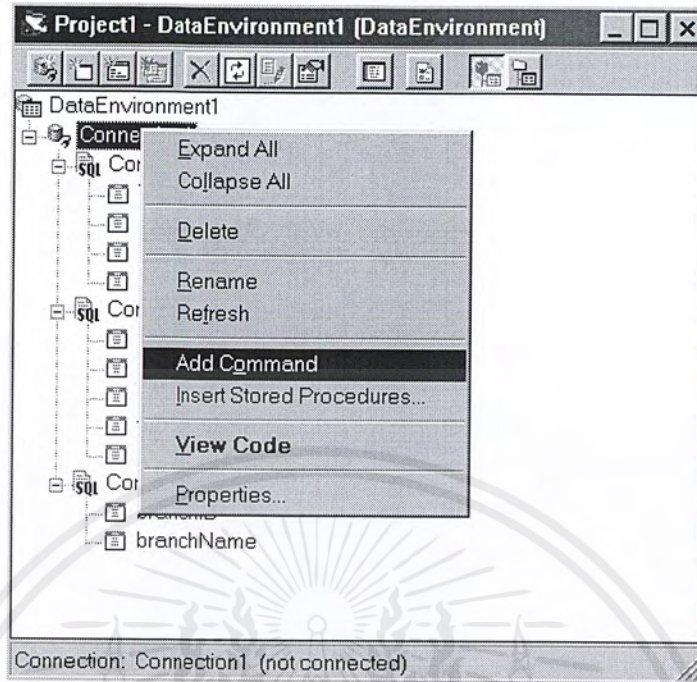
รูปที่ 3.17 การกำหนดคุณสมบัติเพิ่มเติมให้กับ Object Connection

4) การทำงานกับข้อมูลด้วย Object Command

ในการทำงานกับข้อมูลในฐานข้อมูลที่เชื่อมต่อกับ Data Environment เช่น การดึงข้อมูล การแก้ไขข้อมูล เราจะต้องสร้าง Object Command ขึ้น ซึ่ง Object Command นี้สร้างได้จาก Object ในฐานข้อมูล

5) การสร้าง Object Command

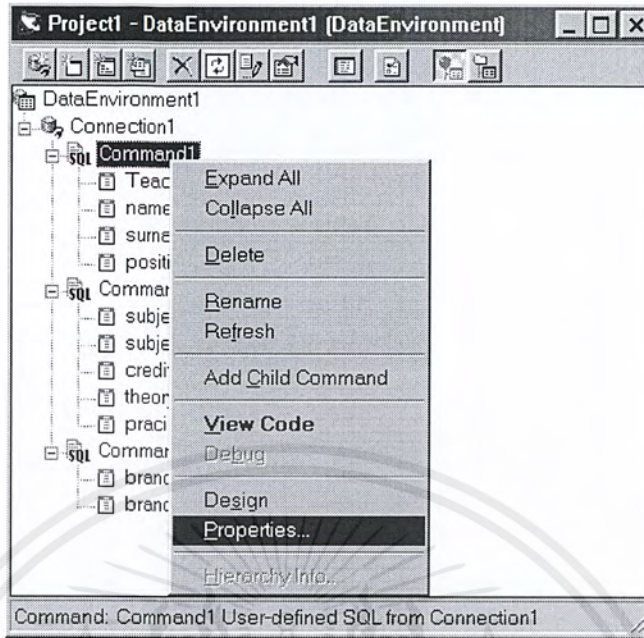
เราสามารถสร้าง Object Command ได้ด้วยการ Click mouse ปุ่มขวา Object Connection ที่เชื่อมต่อกับฐานข้อมูลที่เราจะทำงานด้วย และให้เราเลือกเมนู Add Command จะมี Object Command เพิ่มเข้ามาได้ Object Connection ดังรูปที่ 3.18



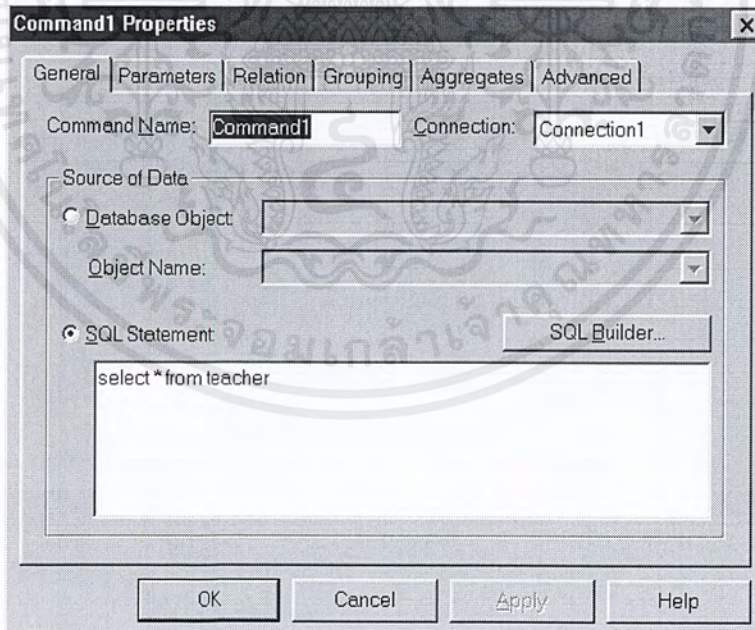
รูปที่ 3.18 การสร้าง Object Command

เมื่อเราเพิ่ม Object Command แล้ว ขั้นตอนต่อไปเราจะต้องกำหนด Object Command ให้เป็นผลลัพธ์ที่เราต้องการ เพื่อจะได้นำไปใช้สร้างโปรแกรม โดยให้ทำตามขั้นตอนต่อไปนี้

- 1) ให้เรา Click mouse ปุ่มขวาที่ Object Command แล้วเลือกเมนู Properties ดังรูปที่ 3.19
- 2) จะปรากฏไดอะล็อกซ์ General ให้เราตั้งชื่อ Object Command ที่สร้าง และเลือก Object Connection ที่ Object Command ดังรูปที่ 3.20
- 3) ให้เรากำหนดผลลัพธ์ของ Object Command ซึ่งทำได้ 2 วิธี คือ
 - 3.1 สร้างจากตารางหรือคิวรีที่มีอยู่ โดยให้เราเลือกชนิด Object (Table ,Query หรือ Store Procedure) จากช่อง Data Object และเลือกชื่อ Object จาก Object Name
 - 3.2 สร้างขึ้นมาใหม่โดยการเลือก SQL Statement แล้วใส่คำสั่ง SQL ลงไป หรือ Click mouse ปุ่ม SQL Builder.. ซึ่งจะปรากฏ Query Designer Window ที่ช่วยในการสร้างคำสั่ง SQL



รูปที่ 3.19 หน้าต่างของเมนู Properties

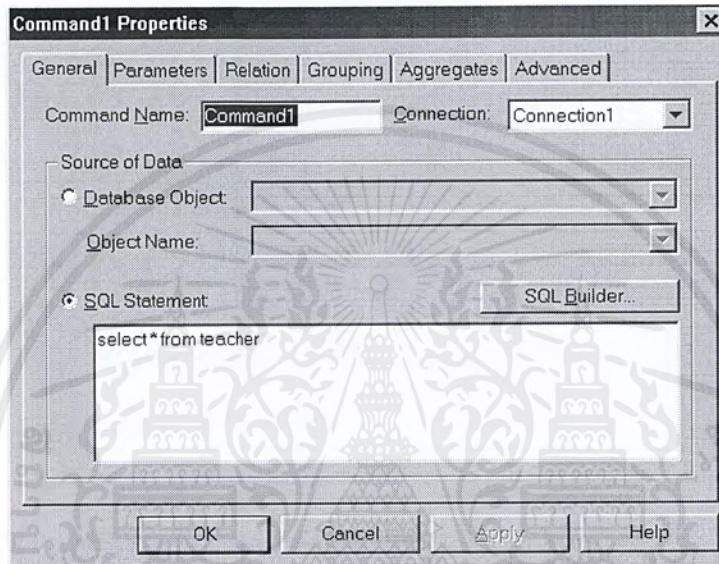


รูปที่ 3.20 การตั้งชื่อ Object Command ที่สร้าง และเลือก Object Connection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) การสร้างคิวรีด้วย Query Designer

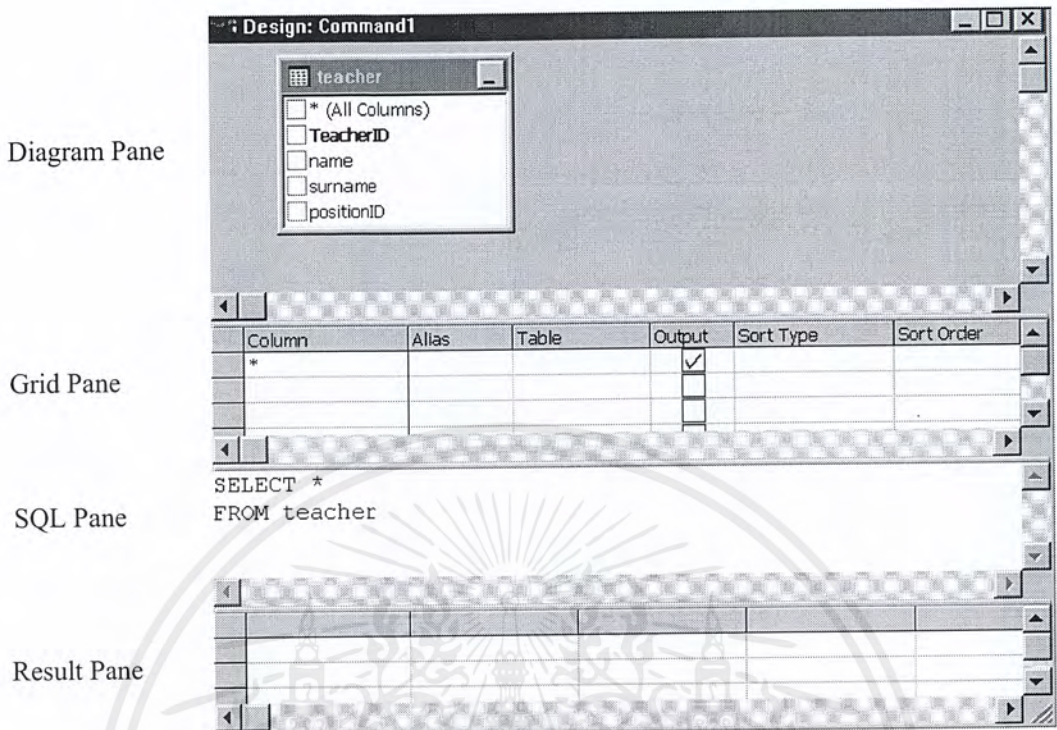
นอกจากเราจะสามารถสร้าง Object Command จากตาราง และคิวรีได้แล้ว เรายังสามารถสร้าง Object Command ด้วยการใช่ Query Designer Window เพื่อช่วยให้เราสามารถสร้างผลลัพธ์ที่เฉพาะเจาะจงมากยิ่งขึ้นเราสามารถเรียกหน้าต่าง Query Designer ขึ้นมาได้โดยการ Click mouse ปุ่ม SQL Builder... ดังรูปที่ 3.21



รูปที่ 3.21 หน้าต่างของ Query Designer

จากรูปที่ 3.22 หน้าต่าง Query Designer จะมีส่วนประกอบต่าง ๆ ที่มีรายละเอียดดังต่อไปนี้

1. Diagram Pane เป็นส่วนที่เก็บตาราง หรือคิวรี(View) ซึ่งจะถูกนำมาสร้างผลลัพธ์
2. Grid Pane เป็นส่วนที่ใช้เลือกฟิลด์ในตาราง หรือคิวรีที่อยู่ใน Diagram Pane ที่จะนำไปเป็นผลลัพธ์ ซึ่งสามารถจัดเรียง รวมทั้งจัดกลุ่มผลลัพธ์ตามฟิลด์ได้ด้วย
3. SQL Pane เป็นส่วนที่แสดงคำสั่ง SQL ที่สอดคล้องกับที่เราเลือกใน Grid Pane และเรายังสามารถใส่คำสั่ง SQL ลงไปได้โดยตรงอีกด้วย
4. Result Pane เป็นส่วนที่แสดงผลลัพธ์ของคิวรีที่สร้าง โดยเราต้อง Click mouse เลือกเมนู Query>Run เพื่อรันคิวรีที่สร้าง



รูปที่ 3.22 ส่วนประกอบต่างๆ ของหน้าต่าง Query Designer

3.6.2 การสร้างความสัมพันธ์ระหว่าง Object Command

เราสามารถสร้างความสัมพันธ์ระหว่าง Object Command ได้ โดยการนำฟิลด์จากทั้งสอง Object มาเชื่อมต่อกัน ซึ่งคล้ายกับการเชื่อมตารางหลาย ๆ ตาราง เพื่อนำมาเป็นผลลัพธ์ในคิวรี ซึ่งผลที่ได้จะเป็น Recordset แบบเป็นลำดับขั้นที่สัมพันธ์กัน และ Object Command หนึ่งจะถูกเรียกว่า Parent Command ส่วน Object Command อีกตัวหนึ่งจะถูกเรียกว่า Child Command และ Parent Command นั้นจะสัมพันธ์แบบ One-to-Many กับ Child Command

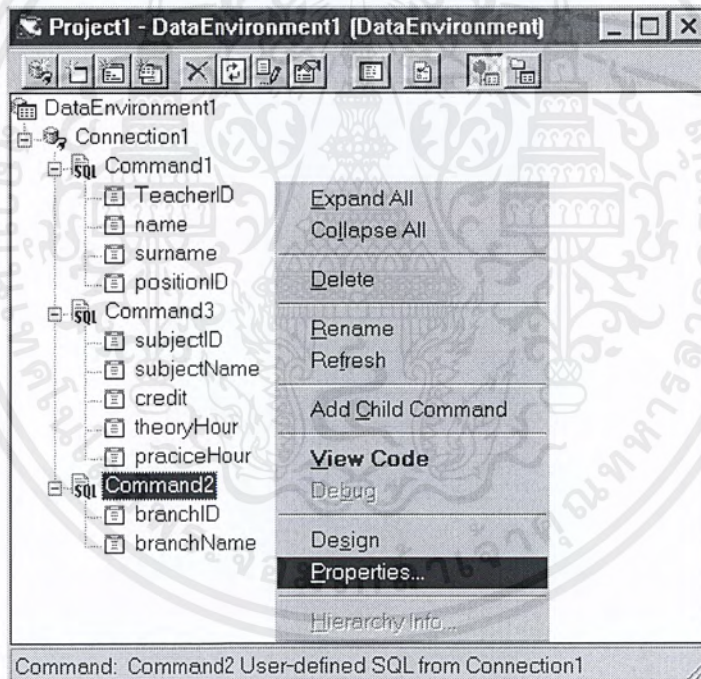
การสร้างความสัมพันธ์ระหว่าง Object Command สามารถทำตามขั้นตอนต่อไปนี้

1. ให้เราสร้าง Object Command 2 ออบเจกต์ ที่ติดต่อกับตาราง TblCategories และ TblProducts และให้เรา Click mouse ปุ่มขวาที่ Object Command ที่เชื่อมต่อกับตาราง TblProducts ที่จะเป็น Child Command และเลือก Properties ดังรูปที่ 3.23
2. จะปรากฏไอคอนเลือก Properties ของ Object Command ให้เราเลือกแท็บ Relation ดังรูปที่ 3.24 ซึ่งถ้าเราต้องการสร้างความสัมพันธ์ก็ให้เราเช็กที่หัวข้อ Relate to a Parent Command Object และกำหนดรายละเอียดต่าง ๆ ดังตารางที่ 3.4

ตารางที่ 3.4 การกำหนดรายละเอียดต่างๆ ของ Relate to a Parent Command Object

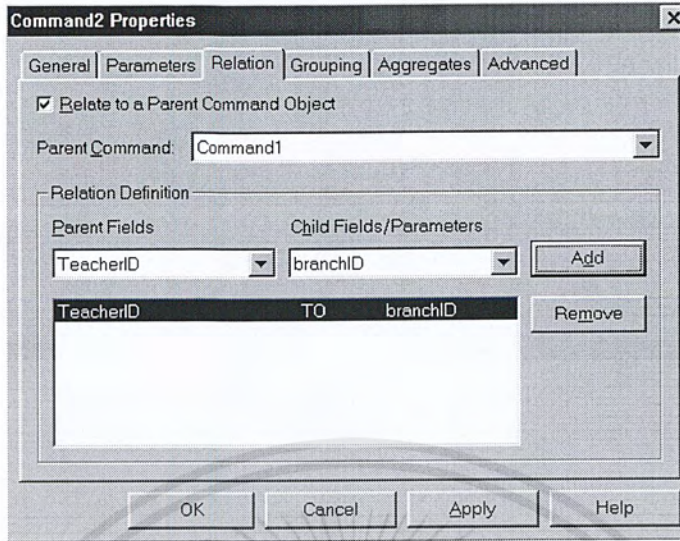
ข้อความ	คำอธิบาย
Parent Command Relation Definition	ให้เราเลือก Object Command ที่จะเป็น Parent เป็นส่วนที่ให้เราเลือกฟิลด์ที่สร้างสัมพันธ์ระหว่าง Object Command ทั้งสองเมื่อเลือกเสร็จแล้วให้เรา Click mouse ปุ่ม Add เพื่อสร้างความสัมพันธ์

3. เสร็จแล้ว ให้เรา Click mouse ปุ่ม OK เพื่อให้การกำหนดค่ามีผล
4. ความสัมพันธ์ที่เรากำหนดระหว่าง Object ทั้งสองจะถูกสร้างขึ้นมา ดังรูปที่ 3.25

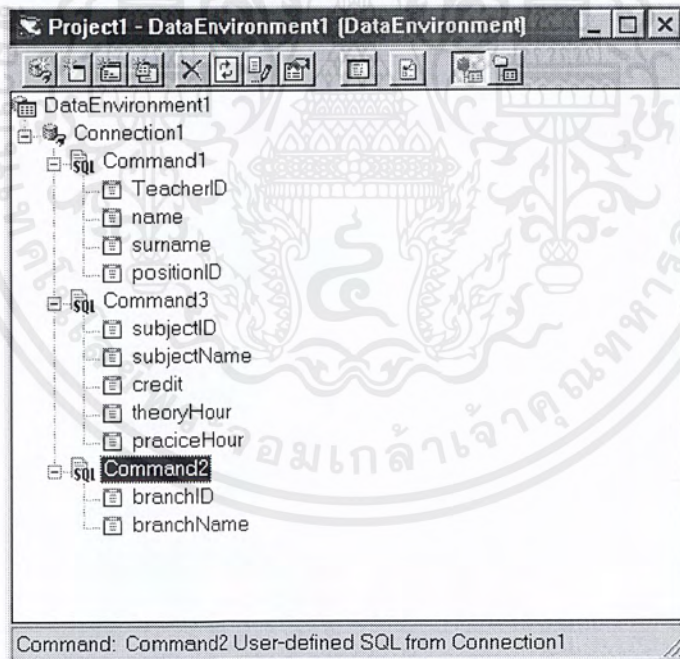


รูปที่ 3.23 การสร้าง Object Command 2 ออบเจ็กต์ ที่ติดต่อกับตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.24 ไดอะล็อกซ์ Properties ของ Object Command



รูปที่ 3.25 ความสัมพันธ์ที่กำหนดระหว่าง Object ทั้งสอง

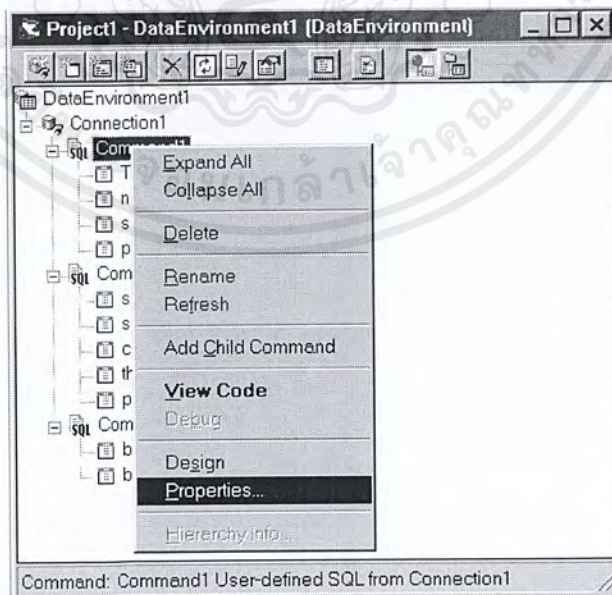
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) การจัดกลุ่มข้อมูลใน Object

เราสามารถจัดกลุ่มข้อมูลใน Object Command ได้ ซึ่งจะช่วยให้เราหาผลลัพธ์บางอย่าง เราสามารถจัดกลุ่มได้โดยการเลือกฟิลด์ที่ใช้จัดกลุ่มใน Object Command ซึ่งเมื่อเราจัดกลุ่มแล้ว ฟิลด์ที่ใช้ในการจัดกลุ่มที่เลือกไว้จะถูกสร้างเป็น Parent Command ของ Object Command เดิมโดยที่ Object Parent Command นี้จะเก็บฟิลด์ที่เชื่อมโยงไปยังแต่ละ Recordset ของ Object Command เดิมที่มีค่าฟิลด์สัมพันธ์กัน

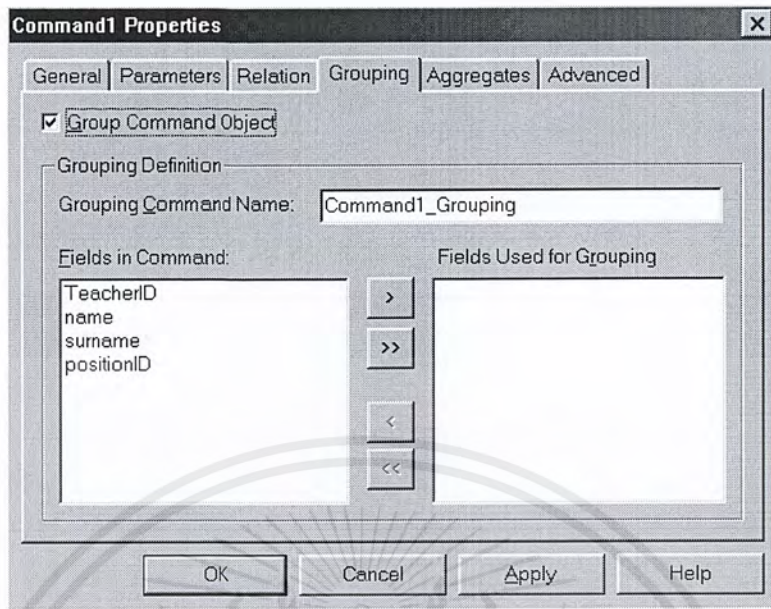
2) การจัดกลุ่มข้อมูลใน Object Command

1. ให้เราเปิดหน้าต่าง Properties ของ Object Command ที่จะจัดกลุ่ม โดยให้เรา Click mouse ปุ่มขวาที่ Object Command ที่จะจัดกลุ่ม และเลือก Properties ดังรูปที่ 2.26
2. จะปรากฏหน้าต่าง Properties ขึ้นมา ให้เราเลือกแท็บ Grouping และให้เราเช็กในหัวข้อ Group Command Object ดังรูปที่ 2.27
3. ให้เราใส่ชื่อ Object Command ใหม่ที่ได้จากการจัดกลุ่มใน Grouping Command Name ซึ่ง Object ใหม่นี้จะประกอบด้วยฟิลด์ที่ใช้จัดกลุ่ม และตัวเชื่อมโยงไปยังฟิลด์ใน Object Command
4. ให้เราเลือกฟิลด์ที่จะใช้ในการจัดกลุ่มจากฝั่ง Fields in Command ไปยัง Fields Used for Grouping เสร็จแล้ว ให้เรา Click mouse ปุ่ม OK
5. จะได้ผลลัพธ์ที่เป็น Object Command 2 ออบเจกต์ ซึ่ง Object Command อันหนึ่งประกอบด้วยฟิลด์ที่ใช้ในการจัดกลุ่ม และ Object Command อันเดิม



รูปที่ 3.26 หน้าต่าง Properties ของ Object Command

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.27 การเลือกแท็บ Grouping และเช็กในหัวข้อ Group Command Object

3) การคำนวณค่าข้อมูลใน Object Command

เราสามารถคำนวณค่าบางอย่างใน Object Command ที่เราต้องการทราบ เช่น จำนวนเรคอร์ดทั้งหมด ใน Object Command ซึ่งเราสามารถผลลัพธ์ในลักษณะนี้ได้โดยการสร้างฟิลด์ Aggregate ขึ้นมาใน Object Command

4) การสร้าง Aggregate ใน Object Command

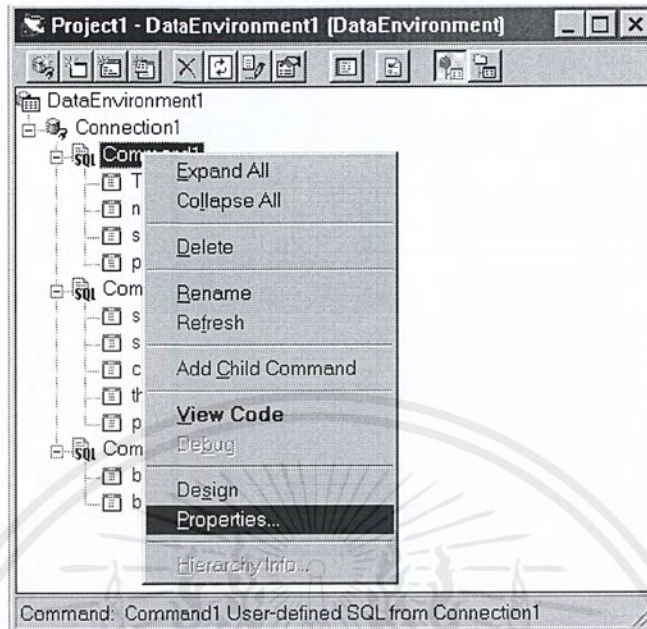
1. ให้เราสร้าง Object Command ที่ติดต่อกับตาราง TblProducts แล้วให้เราเปิดหน้าต่าง Properties ของ Object Command ที่จะสร้างฟิลด์ Aggregate โดยให้เราคลิกเมาส์ปุ่มขวาที่ Object Command เลือก Properties ดังรูปที่ 3.28

2. จะปรากฏหน้าต่าง Properties ขึ้นมา ให้เราเลือกแท็บ Aggregate และให้เราคลิกเมาส์ปุ่ม Add เพื่อสร้างฟิลด์ Aggregate จากนั้นกำหนดรายละเอียดต่างๆ ของฟิลด์ Aggregate ในฝั่ง Aggregate Setting ดังรูปที่ 3.29

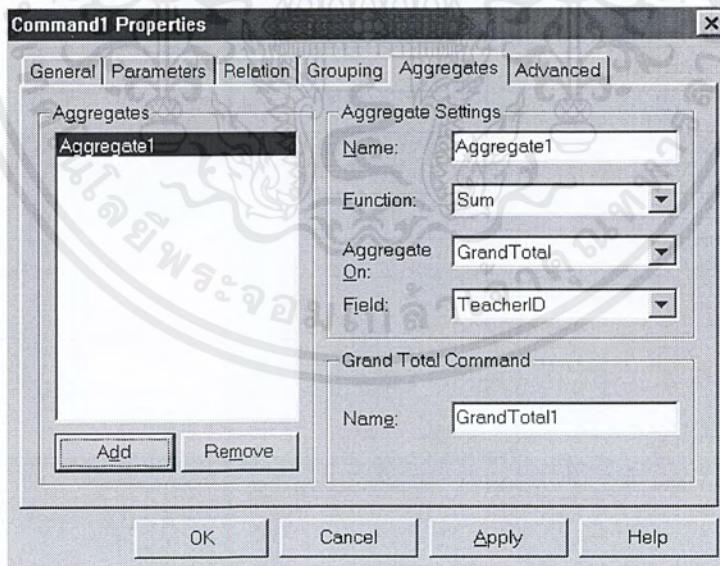
3. คลิกเมาส์ปุ่ม OK เพื่อให้การกำหนดค่ามีผล

4. จะได้ผลลัพธ์เป็นฟิลด์ใหม่ที่เก็บฟิลด์ที่คำนวณค่าข้อมูลใน Object Command นั้น

ดังรูปที่ 3.30

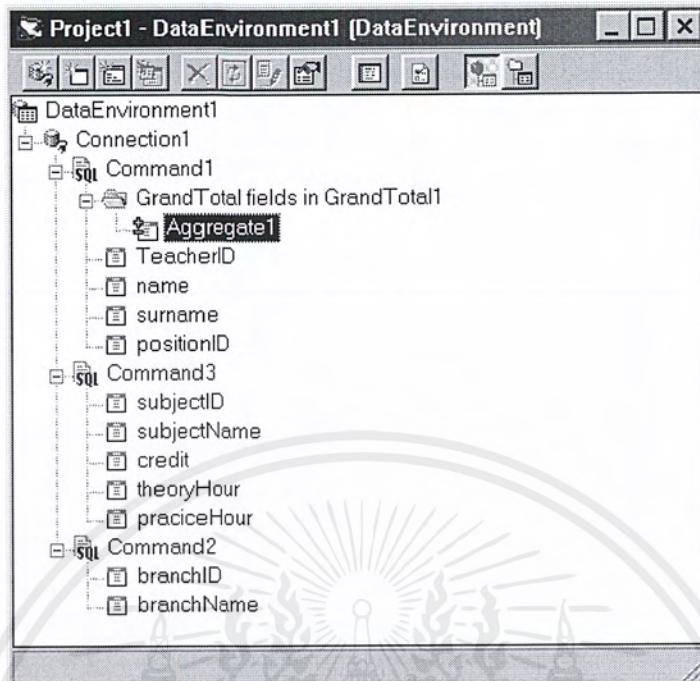


รูปที่ 3.28 หน้าต่าง Properties ของ Object Command



รูปที่ 3.29 การกำหนดรายละเอียดต่างๆ ของฟิลด์ Aggregate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.30 ผลลัพธ์ที่คำนวณค่าข้อมูลใน Object Command

3.6.3 การแสดงข้อมูลจาก Data Environment

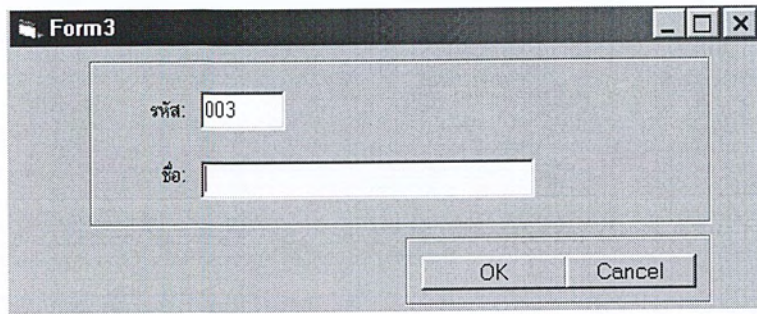
เมื่อเราสร้าง Object Command ได้แล้วต่อไปเราจะแสดงข้อมูลจาก Object นี้ ซึ่งเราสามารถนำข้อมูลใน Data Environment มาแสดงบนฟอร์มได้ 2 วิธี ซึ่งมีดังต่อไปนี้

1. ลากฟิลด์ที่ต้องการแสดงจาก Data Environment ไปยังฟอร์ม
2. กำหนดคุณสมบัติต่างๆ ของ Control ต่างๆ เอง

1) การแสดงข้อมูลจาก Data Environment ด้วยวิธี Drag-and-Drop

เราสามารถสร้างฟอร์มแสดงข้อมูลในฐานข้อมูลด้วยการ Drag ฟิลด์ที่เราจะแสดง จาก Object Command ในหน้าต่าง Data Environment ไปยังฟอร์มที่ใช้แสดงข้อมูล ได้ทันที ดังรูปที่

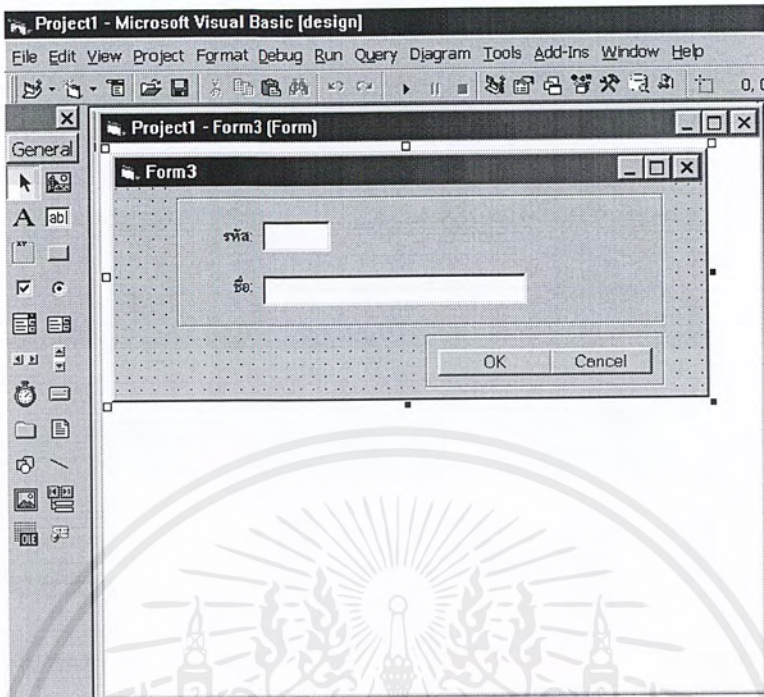
3.31



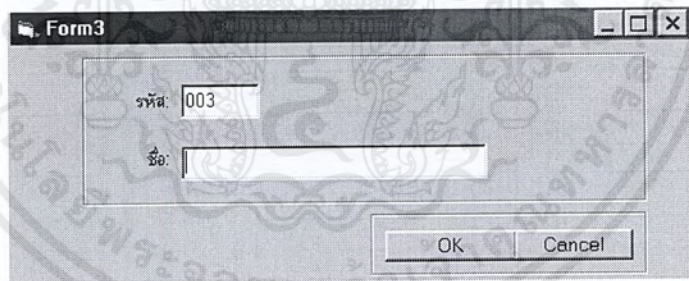
รูปที่ 3.31 การสร้าง Form แสดงข้อมูลในฐานข้อมูล

เราสามารถสร้าง Form ข้างต้นได้ดังนี้

1. ให้สร้าง Project ใหม่ โดยเลือกชนิดของ Project เป็น Standard.exe และเพิ่ม Data Environment ลงไป ดังรูปที่ 3.32
2. ให้เราสร้าง Object Connection เชื่อมกับฐานข้อมูล Order.mdb และสร้าง Object Command ที่เชื่อมไปยังตาราง TblProduces
3. ให้เรา Drag เม้าส์ ลากฟิลด์ รหัส, ชื่อ ไปยังฟอร์ม
4. ให้ Run โปรแกรม เราจะได้ผลลัพธ์การทำงานเป็น Form ที่แสดงฟิลด์ รหัส, ชื่อ ดังรูปที่ 3.33

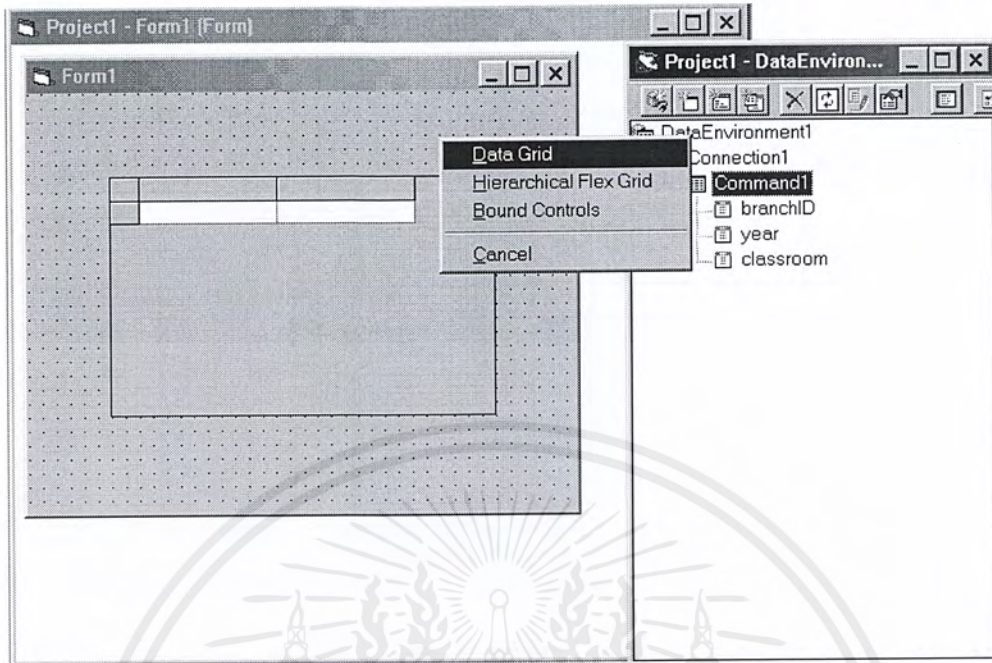


รูปที่ 3.32 การสร้าง Project ใหม่ โดยเลือกชนิดของ Project เป็น Standard.exe



รูปที่ 3.33 ผลลัพธ์การทำงานเป็น Form ที่แสดงฟิลด์ รหัส, ชื่อ

นอกจากนี้แล้ว เรายังสามารถสร้าง Control DataGridView ที่แสดงข้อมูลใน Object Command ด้วยการ Click mouse ปุ่มขวา และ Drag Mouse ลาก Object Command ที่จะสร้าง DataGridView ไปวางบนฟอร์ม ก็จะปรากฏเมนูแล้วให้เราเลือก Data Grid ตามรูปที่ 3.34



รูปที่ 3.34 เมนู Data Grid

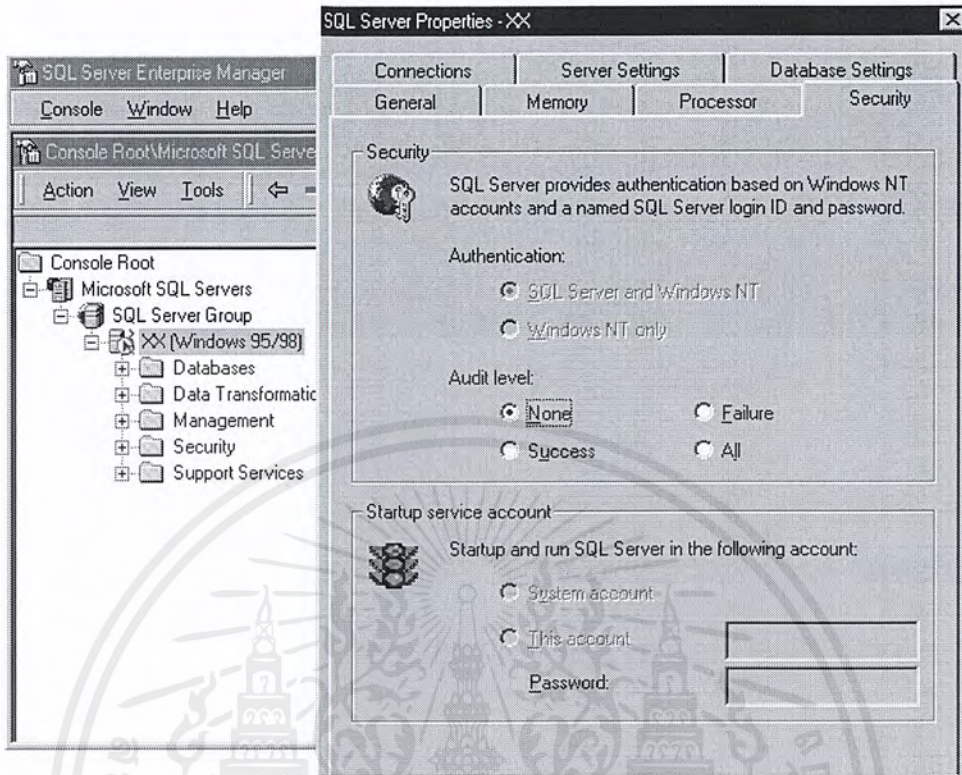
3.7 การจัดการบริหารยูสเซอร์

3.7.1 System Administrator หรือ SA

Sa จะเป็นผู้บริหารระบบหรือ Super User ที่สามารถทำงานหรือควบคุมการทำงานบน SQL Server ได้ทุกอย่าง ไม่มีข้อจำกัดในการทำงาน ซึ่งเป็น User แรกหลังจากการติดตั้ง Microsoft SQL 7.0 และผู้บริหารระบบหรือ System Administrator จำเป็นต้องบริหาร User โดยกำหนดให้ User คนใดสามารถทำอะไรได้บ้างรวมทั้งการสร้าง User เพื่อให้ User นั้นสามารถเข้ามาใช้ระบบได้

การเข้าถึงระบบ SQL Server สามารถทำได้ 2 วิธีดังนี้

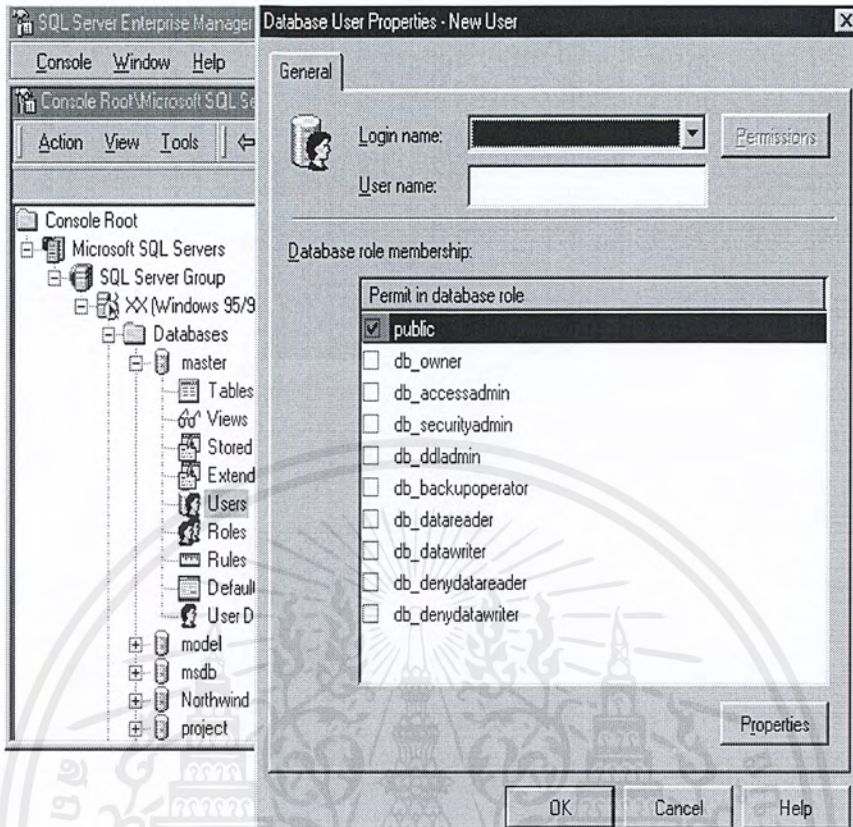
1. การเข้าถึงระบบ SQL Server โดยใช้ Login ของ SQL Server ซึ่งเรียกว่า SQL Server Authentication Mode
2. การเข้าถึงระบบ SQL Server โดยใช้ Login ของ Windows NT ซึ่งเรียกว่า Windows NT Authentication Mode ดังรูปที่ 3.35



รูปที่ 3.35 การกำหนด Interlity Mode

3.7.2 การกำหนดความสามารถในการใช้งานแต่ละฐานข้อมูล

หลังจากเราสร้าง Login ที่สามารถเข้าสู่ระบบได้แล้ว เราก็ต้องกำหนดว่าแต่ละ login จะมีความสามารถใช้งานฐานข้อมูลอะไรได้บ้างโดยเมื่อ Login นั้นจะติดต่อกับฐานข้อมูลเพื่อใช้งานโปรแกรมจะตรวจสอบว่าฐานข้อมูลนั้นๆ มี User ที่เป็นตัวแทนของ Login นั้นๆ ที่สามารถใช้งานฐานข้อมูลนั้นๆ อยู่หรือไม่ ถ้าหากโปรแกรมตรวจสอบไม่พบ User ตัวแทนของ Login นั้นๆ ระบบจะตรวจสอบว่ามี User พิเศษ คือ Guest User เป็น User ที่อนุญาตให้สามารถติดต่อกับฐานข้อมูลได้แต่มีสิทธิ์เพียงอ่านข้อมูลอย่างเดียวไม่สามารถเปลี่ยนแปลงฐานข้อมูลได้อยู่หรือไม่ แต่ถ้าโปรแกรมตรวจสอบแล้วไม่พบ User ทั้งสองลักษณะในฐานข้อมูลก็แสดงค่าผิดพลาดในการติดต่อฐานข้อมูลขึ้นมา ดังรูปที่ 3.36



รูปที่ 3.36 โปรแกรมตรวจสอบ User

3.7.3 ขั้นตอนการสร้าง Guest User

ขั้นตอนการสร้างนั้นทำเหมือนการสร้าง Database User โดยใช้ชื่อ guest ในช่อง User Name: และไม่ต้องใส่ชื่อ Login ในช่อง Login Name กด OK เพื่อให้ระบบสร้าง Guest User หรือ กด Cancel เพื่อยกเลิกการสร้าง

3.7.4 การกำหนดบทบาทต่างๆ ของ User หรือ Roles

แบ่งออกได้เป็น 3 ประเภท

1) Roles

Roles คือกลุ่มมาตรฐานที่ SQL Server กำหนดไว้ให้ เพื่อใช้ในการจัดการกับตัว SQL Serv คล้ายกับ Administrator แบ่งออกได้ 7 กลุ่ม

กลุ่มที่ 1 sysadmin or System Administrators เหมือนเป็น admin ของระบบ บน SQL Server

กลุ่มที่ 2 serveradmin or Server Administrators สมาชิกของกลุ่มนี้สามารถกำหนดเปลี่ยนแปลงแก้ไข Config ของระบบได้ และ shut down

กลุ่มที่ 3 setupadmin or Setup Administrators สามารถทำการ Application , การทำสำเนาฐานข้อมูล เพื่อกระจายข้อมูลไป Server อื่น กำหนด Procedure คือการเขียนคำสั่งต่างๆ ไว้เพื่อเรียกใช้งาน

กลุ่มที่ 4 securityadmin or Security Administrators ทำหน้าที่เกี่ยวข้องกับระบบความปลอดภัยในด้านการสร้าง ควบคุม login การให้สิทธิ์ในการสร้างฐานข้อมูล, ตรวจสอบ SQL Error Log

กลุ่มที่ 5 processadmin or Process Administrators ทำหน้าที่ควบคุมการทำงานของ SQL Server ลบ Query process ที่ไม่ต้องการ

กลุ่มที่ 6 dbcreator or Database Creators ทำหน้าที่จัดการฐานข้อมูล , สร้าง , แก้ไข , ลบ

กลุ่มที่ 7 diskadmin or Disk Administrators ทำหน้าที่กำหนดและจัดการไฟล์ฐานข้อมูล

2) Database Roles

Pre-Defined Database or Database-Specific Fixed Roles แบ่งออกเป็น 9 กลุ่ม

1. db_owner ทำหน้าที่ทุกอย่างภายในฐานข้อมูล
2. db_accessadmin สามารถทำหน้าที่กำหนดให้ User ได้รู้ว่าใครเข้าถึงฐานข้อมูลได้บ้าง แก้ไขการเข้าใช้งานของแต่ละ User
3. db_securityadmin ทำหน้าที่จัดการระบบความปลอดภัยในฐานข้อมูล
4. db_ddladmin ทำหน้าที่จัดการการสร้าง, แก้ไข , ลบ object ต่างๆ ได้
5. db_backupoperator ทำหน้าที่จัดการเกี่ยวกับการ backup database ได้
6. db_datareader ทำหน้าที่อ่านข้อมูลในตารางหรือ view ของฐานข้อมูล
7. db_datawriter ทำหน้าที่ Insert , update, delete ข้อมูลในตารางและ view ได้
8. db_denydatareader ไม่สามารถอ่านข้อมูลในตารางและ view ได้
9. db_denydatawriter ไม่สามารถทำการเปลี่ยนแปลง ข้อมูลจากตารางหรือ view ได้

3) Public Roles

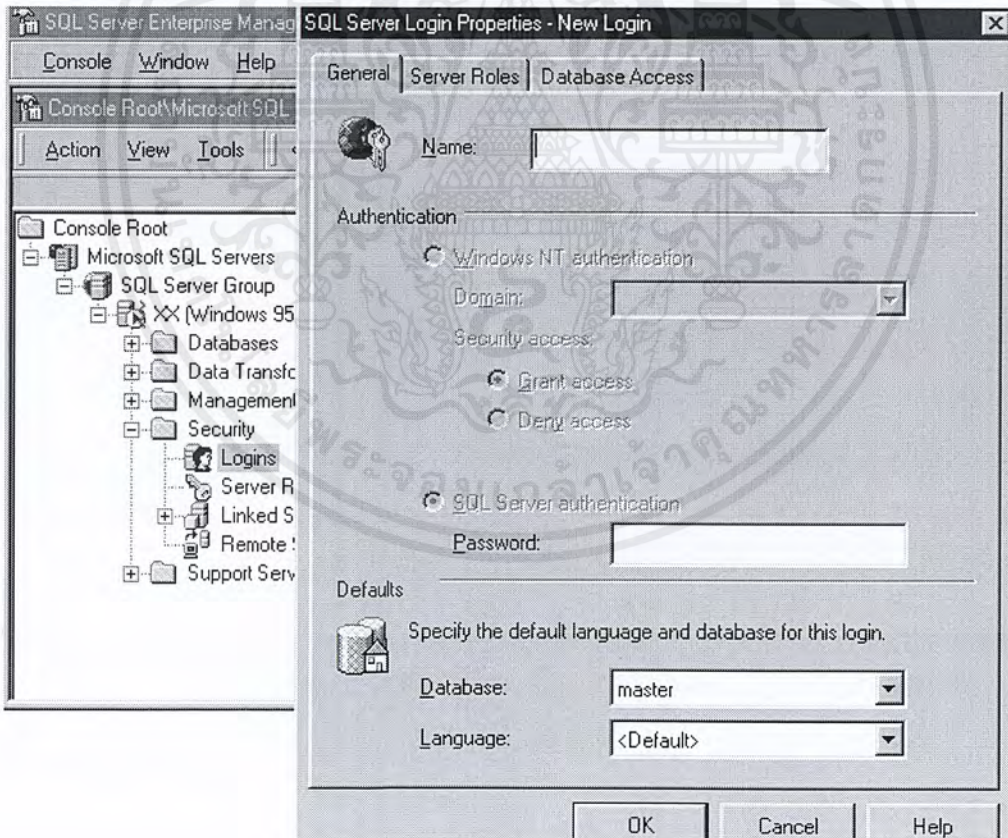
เป็นค่าเริ่มต้นสำหรับทุก User ที่สามารถติดต่อใช้งานของแต่ละฐานข้อมูลได้ ซึ่งเราไม่สามารถลบ Roles นี้ได้ จะต้องมีไว้เพื่อความสะดวกในการอ้างถึง User ต่างๆ ในฐานข้อมูล

3.8 การรักษาความปลอดภัยให้ข้อมูล

วิธีในการรักษาความปลอดภัยให้กับฐานข้อมูล คือการตั้งสิทธิ์ของผู้ใช้งาน โดยสร้าง Role และสิทธิ์การใช้งาน เพื่อป้องกันไม่ให้ผู้ที่ไม่เกี่ยวข้องเข้ามาใช้งานภายในฐานข้อมูล

3.8.1 Password ของ User System Administrator หรือ SA

Sa จะเป็นเหมือนผู้บริหารระบบหรือ Super User ที่สามารถทำงานหรือควบคุมการทำงานบน SQL Server ได้ทุกอย่างไม่มีข้อจำกัดในการทำงาน ซึ่งเป็น User แรกหลังจากการติดตั้ง Microsoft SQL Server 7.0 ซึ่งเราสามารถเข้าระบบในฐานะ sa ได้ ด้วยการใส่ Login Name ว่า sa ส่วน Password เป็นค่าว่าง เพื่อความปลอดภัยของระบบเราควรจะให้ Password กับ Sa แทนที่จะปล่อยให้ค่าว่างไม่เช่นนั้น User อื่น ๆ จะสามารถเข้าใช้งานระบบได้อย่างง่ายดายแค่ Login ด้วยการใส่ Login Name sa แล้ว Password เป็นค่าว่าง และ User ที่เข้ามาในฐานะ sa นั้น จะสามารถทำงานเกี่ยวกับระบบได้ทั้งหมด รวมทั้งการเปลี่ยน Password ของ sa ดังรูปที่ 3.37



รูปที่ 3.37 การตั้ง Password ให้กับ sa

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.2 กลไกในการรักษาความปลอดภัยฐานข้อมูล

ระบบจัดการฐานข้อมูล (DBMS) จะรับคำสั่งจากผู้บริหารระบบฐานข้อมูล (SA) ในการควบคุมการใช้ข้อมูลของ User ซึ่งส่วนประกอบของกลไกในการรักษาความปลอดภัยฐานข้อมูลจะมีอยู่ 3 ส่วนด้วยกัน คือ

1) User (ผู้ใช้)

เป็นผู้ใช้งานระบบฐานข้อมูล ซึ่งในองค์กรหรือบริษัทต่าง ๆ จะมีหลายแผนก ในการทำงานการกำหนด User ID ในการเข้าถึงฐานข้อมูลจะต้องพิจารณาถึงมาตรฐานของแต่ละระบบฐานข้อมูล ซึ่ง ANSI/ISO มีมาตรฐานในการกำหนดชื่อ User ID ความยาว 18 ตัวอักษร แต่สำหรับ MS.SQL Server สามารถกำหนดได้ถึง 30 ตัวอักษร ในการกำหนด User ID สามารถจะกำหนดแบบแต่ละคน หรือเป็นกลุ่มเช่นในส่วนของฝ่ายขาย มีพนักงานที่ต้องการใช้ข้อมูลอยู่ 5 คน ก็กำหนดให้ 5 คนนี้มี User ID เดียวกันในกลุ่มของ SALE ซึ่งจะแบ่งเบาภาระการดูแลให้ SA อีกด้วย

2) Database Object (อ็อบเจกต์ฐานข้อมูล)

ในส่วนนี้จะเกี่ยวข้องกับตารางและการ View การรักษาความปลอดภัย ในการเรียกดูข้อมูลจากตาราง ทำได้โดยผ่าน View ข้อมูลซึ่งเป็นตารางเสมือน (Virtual Table) ผู้ที่สร้างตารางขึ้นมาจะมีสิทธิ์ในการจัดการทุกอย่าง (Select , Update , insert ,Delete) และสามารถจะมอบสิทธิ์บางอย่างหรือทั้งหมดให้กับผู้ใช้คนอื่นได้ การสร้าง View ข้อมูลขึ้นมา จะเป็นการมอบสิทธิ์ให้ผู้ใช้สามารถเรียกดูข้อมูลใน View ด้วยคำสั่ง SELECT แบบอ่านอย่างเดียว (Read Only) เท่านั้น ไม่สามารถจะทำการ INSERT DELETE และ UPDATE ผ่านทาง View ข้อมูลได้ จะเห็นว่าการสร้าง View ข้อมูลนั้นเพิ่มประสิทธิภาพในการรักษาความปลอดภัยให้กับข้อมูลอย่างมาก

3) Privilege (สิทธิ์ในการใช้งาน)

เป็นสิทธิ์ที่ผู้ใช้งานฐานข้อมูลได้รับในการทำงานกับตารางและ Viewซึ่งจะมีอยู่ 4 แบบด้วยกัน คือ

SELECT เป็นสิทธิ์ในการเรียกดูข้อมูลจากตาราง หรือ View โดยสามารถจะเลือกดูทั้งหมดหรือกำหนดเพียงบางคอลัมน์

INSERT เป็นสิทธิ์ในการเพิ่มเรคอร์ดเข้าไปในตารางหรือ View

DELETE เป็นสิทธิ์ในการลบเรคอร์ดของข้อมูลออกจากตาราง หรือ View

UPDATE เป็นสิทธิ์ในการปรับปรุง – เปลี่ยนแปลงข้อมูลในตาราง หรือ View

3.8.3 รูปแบบของ Permission

ระบบฐานข้อมูล MicroSoft SQL Server จะใช้ Permission ในการควบคุมรักษาความปลอดภัยของฐานข้อมูล โดยจะอนุญาตเฉพาะรายการซึ่งผู้ที่มีสิทธิ์ที่จะใช้ Object ฐานข้อมูล ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประกอบด้วย Table , Defaults, Rule, Index, View, Trigger และ Stored Procedure ถ้าปราศจาก Permission ที่ถูกต้อง ผู้ใช้ยังสามารถจะเชื่อมต่อกับ SQL Server ได้ แต่ไม่สามารถจะใช้ Object ฐานข้อมูลได้ Permission ของ SQL Server จะมีอยู่ 3 แบบด้วยกัน คือ

1) Statement Permission

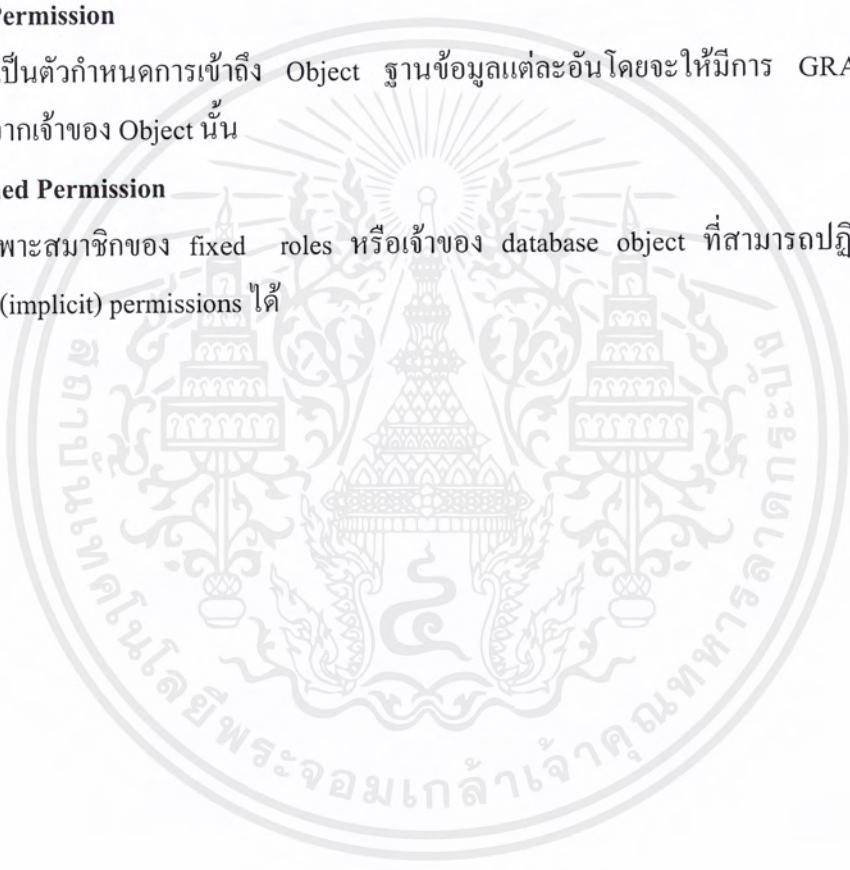
จะอนุญาตให้แต่ละผู้ใช้ หรือ Group ในการปฏิบัติงานบนฐานข้อมูลได้หลายอย่าง (เฉพาะสมาชิกของ Sysadmin, db_owner หรือ bd_Securityadmin role เท่านั้นที่สามารถใช้ statement grant ได้)

2) Object Permission

จะเป็นตัวกำหนดการเข้าถึง Object ฐานข้อมูลแต่ละอันโดยจะให้มีการ GRANT และ REVOKE จากเจ้าของ Object นั้น

3) Predefined Permission

เฉพาะสมาชิกของ fixed roles หรือเจ้าของ database object ที่สามารถปฏิบัติงานกับ Predefined (implicit) permissions ได้



บทที่ 4

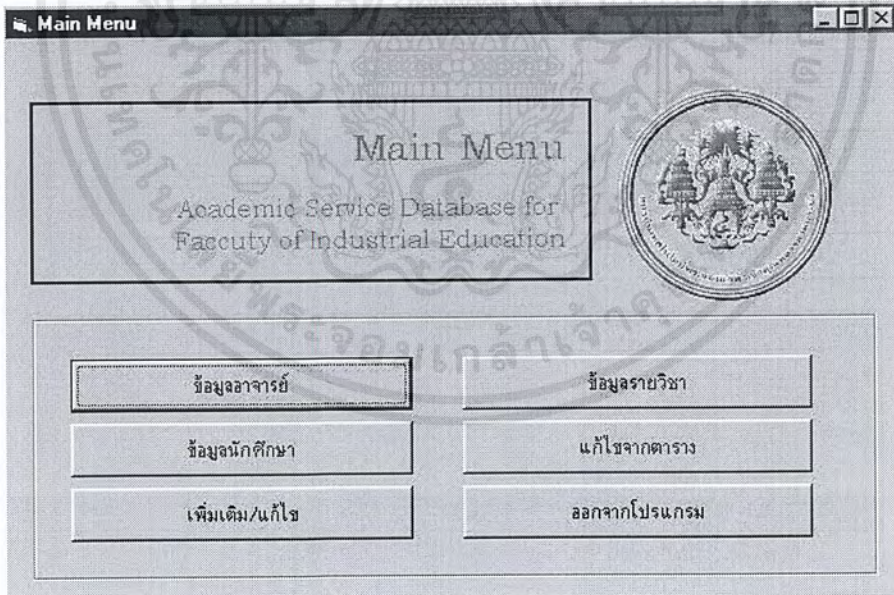
การทดลอง และผลการทดลอง

4.1 การเข้าสู่โปรแกรม

โปรแกรมระบบฐานข้อมูลหน่วยบริการวิชาการของคณะครุศาสตร์อุตสาหกรรม เป็นโปรแกรมที่ทำงานบนระบบปฏิบัติการ Windows NT เนื่องจากตัวโปรแกรมสร้างจากโปรแกรม Microsoft Visual Basic 6 โดยจะเน้นในส่วนของ ตารางเรียน ตารางสอน ซึ่งเราสามารถเข้าสู่โปรแกรมได้โดย ขั้นตอนต่อไปนี

4.1.1 เข้าสู่โปรแกรม Microsoft Visual Basic 6

เมื่อเข้าสู่โปรแกรม Microsoft Visual Basic 6 จากนั้นให้รันโปรแกรมที่ชื่อ Project 1 ก็จะเข้าสู่เมนูหลักของโปรแกรมระบบฐานข้อมูลหน่วยบริการวิชาการของคณะครุศาสตร์อุตสาหกรรม ดังรูปที่ 4.1



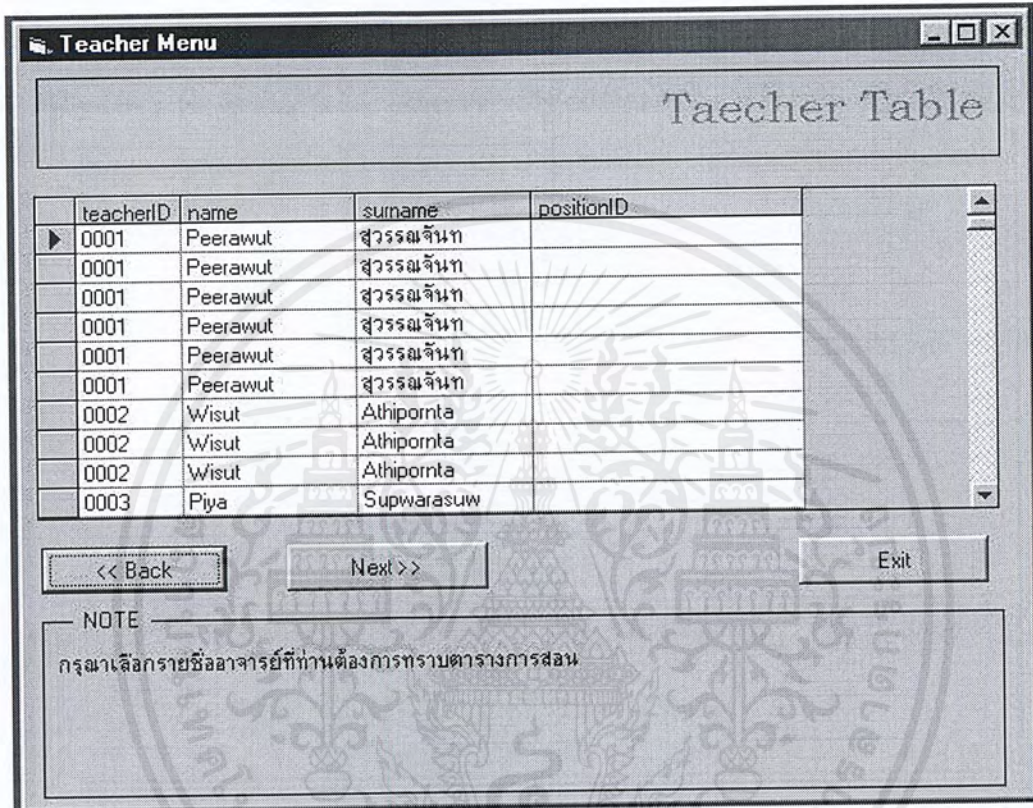
รูปที่ 4.1 เมนูหลักของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การใช้งานโปรแกรม

4.2.1 ข้อมูลอาจารย์

จากรูปที่ 4.1 คลิกปุ่มข้อมูลอาจารย์ จะปรากฏดังรูปที่ 4.2



รูปที่ 4.2 เมนูข้อมูลอาจารย์

จากรูปที่ 4.2 สามารถอธิบายได้ดังนี้

ปุ่ม << Back ใช้ในการกลับไปยังเมนูหลัก

ปุ่ม Next >> ไปเมนูถัดไป ดังรูปที่ 4.3

ปุ่ม Exit ออกจากโปรแกรม

Teacher

Teacher

teacherID: 0001

name: Peerawut

surname: สุวรรณจินท

positionID:

teacherID	subjectName	credit	theoryHou	practiceHou	branchName	year
0001	Principles of Communication	3	3	0	Telecommunication	1
0001	Special Teaching	3	2	3	Telecommunication	1
0001	Principles of Communication	3	3	0	Telecommunication	1
0001	Special Teaching	3	2	3	Telecommunication	1
0001	Digital Communication System	3	3	0	Telecommunication	2
0001	Digital Communication System	3	3	0	Telecommunication	2

<< Back < > Report

รูปที่ 4.3 ข้อมูลอาจารย์

จากรูปที่ 4.3 สามารถอธิบายได้ดังนี้

ปุ่ม <<Back กลับไปสู่เมนูข้อมูลอาจารย์

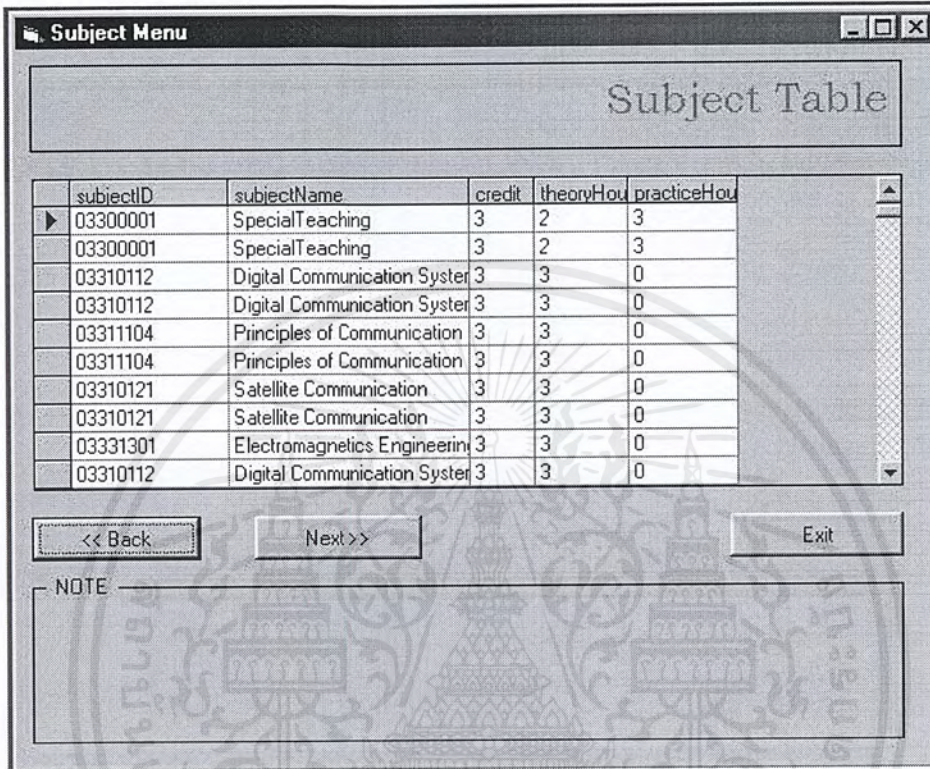
ปุ่ม < เลื่อนลง

ปุ่ม > เลื่อนขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 ข้อมูลรายวิชา

จากรูปที่ 4.1 คลิกปุ่มข้อมูลรายวิชา จะปรากฏดังรูปที่ 4.4



รูปที่ 4.4 เมนูข้อมูลรายวิชา

จากรูปที่ 4.4 สามารถอธิบายได้ดังนี้

ปุ่ม << Back ใช้ในการกลับไปยังเมนูหลัก

ปุ่ม Next >> ไปเมนูถัดไป ดังรูปที่ 4.5

ปุ่ม Exit ออกจากโปรแกรม

Subject

subjectID: 03300001

subjectName: SpecialTeaching

credit: 3

theoryHour: 2

practiceHour: 3

subjectID	subjectName	name	surname	positionID	bl
03300001	SpecialTeaching	Peerawut	สุวรรณจินต		T
03300001	SpecialTeaching	Peerawut	สุวรรณจินต		T

<< Back < > Report

รูปที่ 4.5 ข้อมูลรายวิชา

จากรูปที่ 4.5 สามารถอธิบายได้ดังนี้

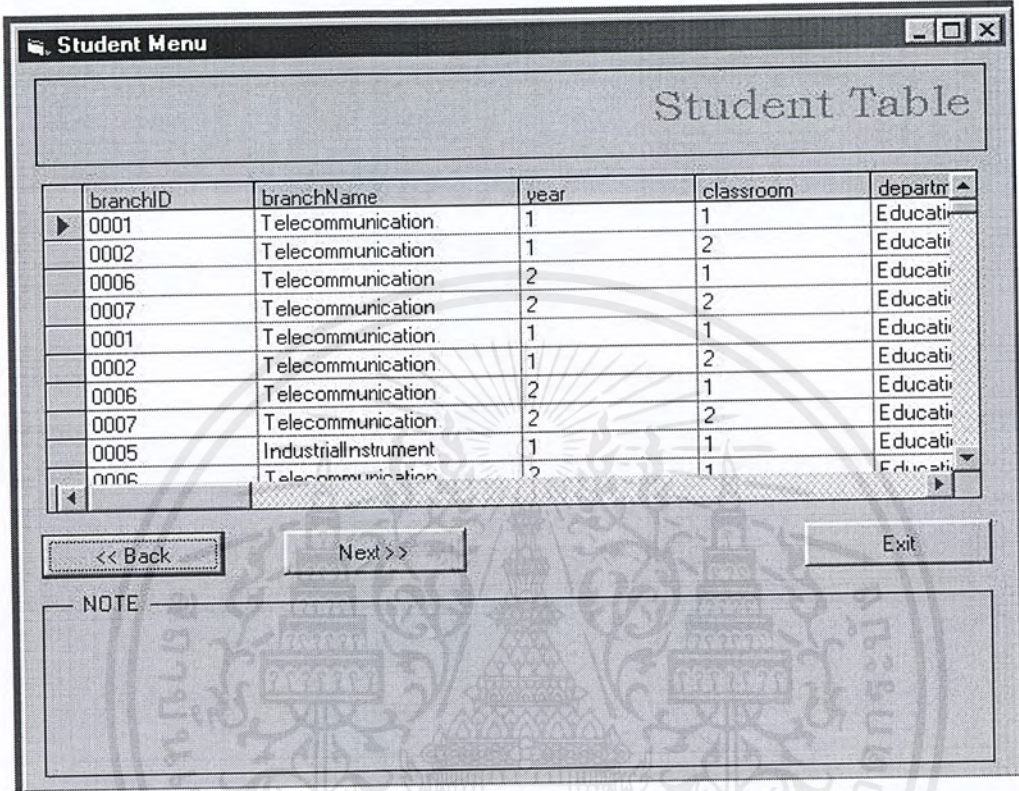
ปุ่ม <<Back กลับไปสู่เมนูข้อมูลรายวิชา

ปุ่ม < เลื่อนลง

ปุ่ม > เลื่อนขึ้น

4.2.3 ข้อมูลนักศึกษา

จากรูปที่ 4.1 คลิกปุ่มข้อมูลนักศึกษา จะปรากฏดังรูปที่ 4.6



รูปที่ 4.6 เมนูข้อมูลนักศึกษา

จากรูปที่ 4.6 สามารถอธิบายได้ดังนี้

ปุ่ม << Back ใช้ในการกลับไปยังเมนูหลัก

ปุ่ม Next >> ไปเมนูถัดไป ดังรูปที่ 4.7

ปุ่ม Exit ออกจากโปรแกรม

Student

Student

branchID: 0001

branchName: Telecommunication

year: 1

classroom: 1

departmentName: EducationEngineer

branchID	branchName	subjectName	credit	theoryHour	practiceHour	name	surname	positio
0001	Telecommun	Principles of I	3	3	0	Peerawut	สุวรรณจินทา	
0001	Telecommun	SpecialTeacl	3	2	3	Peerawut	สุวรรณจินทา	
0001	Telecommun	Electric Circu	3	3	0	Piya	Supwarasuw	
0001	Telecommun	Telephone E	3	3	0	Amornchai	Chaichana	

<< Back < > Report

รูปที่ 4.7 ข้อมูลนักศึกษา

จากรูปที่ 4.7 สามารถอธิบายได้ดังนี้

ปุ่ม <<Back กลับไปสู่เมนูข้อมูลนักศึกษา

ปุ่ม < เลื่อนลง

ปุ่ม > เลื่อนขึ้น

จากรูปที่ 4.3, 4.5 และ 4.7 เมื่อคลิกที่ปุ่ม Report สามารถที่จะ Print และ Save ข้อมูลที่ต้องการได้ ดังรูปที่ 4.8

Teacher Report

Zoom 75%

Table Report
Faculty of Industrial Education
Kingmongkut Institute of Technology Ladkrabang

รหัสอาจารย์: 0001 ชื่อ: Peersrut นามสกุล: สุวรรณจิตต์ ตำแหน่งทางวิชาการ:

รหัสวิชา	ชื่อวิชา	หน่วยกิต	ภาคปฏิบัติ	สาขาวิชา	ชั้นปี	ห้อง	ภาควิชา	วัน	เวลา	ห้อง	
03311104	Principles of	3	3	0	Telecommunication	1	1	Education Engineer	Tuesday	9.00 - 12.00	K.321
03300001	Special Teaching	3	2	3	Telecommunication	1	1	Education Engineer	Thursday	13.00 - 18.00	K.310
03311104	Principles of	3	3	0	Telecommunication	1	2	Education Engineer	Tuesday	13.00 - 16.00	K.310
03300001	Special Teaching	3	2	3	Telecommunication	1	2	Education Engineer	Thursday	8.00 - 13.00	K.310
03310112	Digital	3	3	0	Telecommunication	2	1	Education Engineer	Wednesday	8.00 - 11.00	K.310

Pages: 1

รูปที่ 4.8 หน้าต่างของโปรแกรมรายงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 แก้ไขจากตาราง

จากรูปที่ 4.1 คลิกปุ่มแก้ไขจากตาราง จะปรากฏดังรูปที่ 4.9 เมื่อต้องการแก้ไขข้อมูลให้ทำการระบายข้อมูลที่ต้องการแล้วทำการแก้ไขข้อมูลดังกล่าว

name	surname	positionID	subjectID	subjectName	credit	theoryHo
Peerawut	สุวรรณจินต		03311104	Principles of Communication	3	3
Peerawut	สุวรรณจินต		03300001	SpecialTeaching	3	2
Piya	Supwarasuw		03311102	Electric Circuit Analysis	3	3
Amornchai	Chaichana		03311105	Telephone Engineering	3	3
Peerawut	สุวรรณจินต		03311104	Principles of Communication	3	3
Peerawut	สุวรรณจินต		03300001	SpecialTeaching	3	2
Amornchai	Chaichana		03311105	Telephone Engineering	3	3
Piya	Supwarasuw		03311102	Electric Circuit Analysis	3	3
Peerawut	สุวรรณจินต		03310112	Digital Communication System	3	3
Piya	Supwarasuw		03310112	Digital Communication System	3	3
Worawit	Somha		03312108	Digital Signal Processing	3	3
Pongkieat	Chetpitaks		03310120	Optical Communication	3	3
Wisut	Athipornita		03310121	Satellite Communication	3	3
Peerawut	สุวรรณจินต		03310112	Digital Communication System	3	3
Piya	Supwarasuw		03310112	Digital Communication System	3	3
Worawit	Somha		03312108	Digital Signal Processing	3	3
Wisut	Athipornita		03310121	Satellite Communication	3	3
Pongkieat	Chetpitaks		03310120	Optical Communication	3	3
Suchin	Arijhan		03321206	Computer Engineering	3	3

รูปที่ 4.9 ปุ่มแก้ไขจากตาราง

จากรูปที่ 4.9 สามารถอธิบายได้ดังนี้

ปุ่ม <<Back กลับไปสู่เมนูหลัก

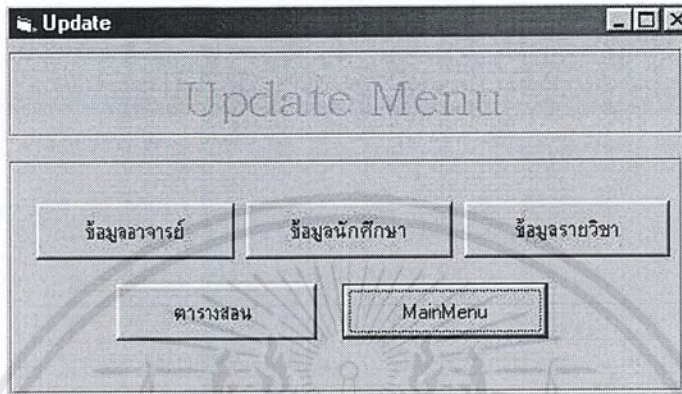
ปุ่ม < เลื่อนลง

ปุ่ม > เลื่อนลง

ปุ่ม Exit ออกจากโปรแกรม

2.2.5 เพิ่มเติม / แก้ไข

เมื่อต้องการเพิ่มเติม / แก้ไขข้อมูลสามารถทำได้โดยคลิกปุ่มเพิ่มเติม / แก้ไขในรูปที่ 4.1 ก็
จะปรากฏหน้าต่างเมนูเพิ่มเติมขึ้น ดังรูปที่ 4.10



รูปที่ 4.10 เมนูเพิ่มเติม / แก้ไข

จากรูปที่ 4.10 สามารถอธิบายได้ดังนี้

1) ปุ่มข้อมูลอาจารย์

จากรูปที่ 4.1 คลิกปุ่มข้อมูลอาจารย์ จะปรากฏดังรูปที่ 4.11

รูปที่ 4.11 ปุ่มข้อมูลอาจารย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.11 สามารถอธิบายได้ดังนี้

- ปุ่ม <<Back กลับไปสู่เมนูเพิ่มเติม / แก้ไข
- ปุ่ม Update แก้ไขข้อมูล
- ปุ่ม AddNew เพิ่มเติมข้อมูล
- ปุ่ม Delete ลบข้อมูล
- ปุ่ม MainMenu กลับสู่เมนูหลัก
- ปุ่ม Exit ออกจากโปรแกรม

2) ปุ่มข้อมูลนักศึกษา

จากรูปที่ 4.12 คลิกปุ่มข้อมูลนักศึกษา จะปรากฏดังรูปที่ 4.12

รูปที่ 4.12 ปุ่มข้อมูลนักศึกษา

จากรูปที่ 4.12 สามารถอธิบายได้ดังนี้

- ปุ่ม <<Back กลับ ไปสู่เมนูเพิ่มเติม / แก้ไข
- ปุ่ม Update แก้ไขข้อมูล
- ปุ่ม AddNew เพิ่มเติมข้อมูล
- ปุ่ม Delete ลบข้อมูล
- ปุ่ม MainMenu กลับสู่เมนูหลัก
- ปุ่ม Exit ออกจากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ปุ่มข้อมูลรายวิชา

จากรูปที่ 4.13 คลิกปุ่มข้อมูลรายวิชา จะปรากฏดังรูปที่ 4.13

The screenshot shows a window titled "Add/Update" with a header "Update/Addnew Subject". Below the header are several input fields: "รหัสวิชา" (Course Code) with a dropdown menu showing "03300002", "ชื่อวิชา" (Course Name) with a text box containing "Classroom Teaching Practice", "หน่วยกิต" (Credits) with a spinner box showing "3", "ทฤษฎี" (Theory) with a spinner box showing "0", and "ปฏิบัติ" (Practical) with a spinner box showing "6". At the bottom of the window is a row of buttons: "<< Back", "Update", "Addnew", "Delete", "MainMenu", and "Exit".

รูปที่ 4.13 ปุ่มข้อมูลรายวิชา

จากรูปที่ 4.13 สามารถอธิบายได้ดังนี้

- ปุ่ม <<Back กลับไปสู่เมนูเพิ่มเติม / แก้ไข
- ปุ่ม Update แก้ไขข้อมูล
- ปุ่ม AddNew เพิ่มเติมข้อมูล
- ปุ่ม Delete ลบข้อมูล
- ปุ่ม MainManu กลับสู่เมนูหลัก
- ปุ่ม Exit ออกจากโปรแกรม

4) ปุ่มตารางสอน

จากรูปที่ 4.14 คลิกปุ่มตารางสอน จะปรากฏดังรูปที่ 4.14

รูปที่ 4.14 ปุ่มตารางสอน

จากรูปที่ 4.14 สามารถอธิบายได้ดังนี้

- ปุ่ม <<Back กลับไปสู่เมนูเพิ่มเติม / แก้ไข
- ปุ่ม Update แก้ไขข้อมูล
- ปุ่ม AddNew เพิ่มเติมข้อมูล
- ปุ่ม Delete ลบข้อมูล
- ปุ่ม MainManu กลับสู่เมนูหลัก
- ปุ่ม Exit ออกจากโปรแกรม

5) ปุ่ม MainMenu

จากรูปที่ 4.10 คลิกปุ่ม MainMenu ก็ จะกลับ ไปสู่เมนูหลัก ดังรูปที่ 4.1

บทที่ 5

บทสรุป ปัญหาแนวทางการแก้ไข และพัฒนา

5.1 สรุปผลการทดลอง

จากการทดลองโปรแกรมฐานระบบฐานข้อมูลหน่วยบริการวิชาการของคณะครุศาสตร์อุตสาหกรรม ในส่วนของตารางเรียน ตารางสอน ปรากฏว่าโปรแกรมสามารถทำงานได้ตรงกับเป้าหมายที่วางไว้ คือ สามารถเพิ่มเติม, แก้ไข และค้นหาข้อมูล ซึ่งช่วยให้อาจารย์และนักศึกษาสามารถสืบค้นข้อมูลได้สะดวกยิ่งขึ้น สามารถเลือกแสดงรายการได้หลายรูปแบบ เช่น คูตารางสอนของอาจารย์แต่ละท่าน หรือเลือกรายวิชาที่ต้องการได้ พร้อมทั้งอธิบายขั้นตอนการใช้งานอย่างละเอียด

5.2 ปัญหาในการทำงาน

1. ระบบปฏิบัติการ Windows NT ไม่สามารถรองรับการใช้งาน Microsoft SQL Server 7.0 ในรูปแบบภาษาไทยได้
2. ความซ้ำซ้อนของข้อมูล
3. การเชื่อมความสัมพันธ์ระหว่างตาราง
4. โปรแกรมที่ออกแบบใช้งานสำหรับผู้ใช้โดย Visual Basic 6.0 มีความยุ่งยากในการติดต่อกับฐานข้อมูล SQL Server 7.0

5.3 แนวทางในการแก้ปัญหา

1. ป้อนข้อมูลภาษาไทยผ่านโปรแกรมตารางเรียนตารางสอน บนระบบปฏิบัติการ Window 98
2. ออกแบบตารางโดยทำให้ข้อมูลมีความกระชับรัดกุมยิ่งขึ้น
3. ทำการลดความซ้ำซ้อนของข้อมูล

5.4 แนวทางในการพัฒนา

1. สามารถขยายขอบเขตของโปรแกรมให้สามารถทำงานอื่นๆ ของภาควิชาได้
2. สามารถที่จะขยายขอบเขตของโปรแกรมให้ใช้ได้ในระดับคณะหรือสถาบันได้
3. พัฒนาให้โปรแกรมให้สามารถสำรองข้อมูล (Backup) ได้ในแต่ละปีการศึกษา
4. สามารถพัฒนาให้โปรแกรมทำงานในระบบเครือข่าย (Network) ได้
5. สามารถ Import ข้อมูลจากภายนอกในรูปของ File ข้อมูลแบบ Text ได้
6. สามารถ Export ข้อมูลออกไปยังฐานข้อมูลอื่นได้





ภาคผนวก
โปรแกรมติดต่อกับฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
///// form 1 Mainmenu /////
```

```
Private Sub Command1_Click()
```

```
Form1.Hide
```

```
Form2.Show
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
Form1.Hide
```

```
Form3.Show
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
Form1.Hide
```

```
Form4.Show
```

```
End Sub
```

```
Private Sub Command4_Click()
```

```
End
```

```
End Sub
```

```
Private Sub Command5_Click()
```

```
Form9.Show
```

```
Form1.Hide
```

```
End Sub
```

```
Private Sub Command6_Click()
```

```
Form8.Show
```

```
Form1.Hide
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Form_Load()
End Sub

////////////////////////////////////

/// form2 ///

Private Sub Command1_Click()
Form2.Hide
Form1.Show
End Sub

Private Sub Command2_Click()
Form2.Hide
Form5.Show
End Sub

Private Sub Command3_Click()
End
End Sub

Private Sub Form_Load()
End Sub

////////////////////////////////////

//// form 10 addnew teacher //////////

Private Sub Combo1_Click()
With DataEnvironment3.rsCommand1
Dim i As Integer
i = 0
.MoveFirst
Do

```

```

If (Combo1.Text = txtname.Text) Then
With DataEnvironment3.rsCommand1
i = 0
.MoveFirst
Do
    Combo2.List(i) = txtsurname.Text
    .MoveNext
    i = i + 1
Loop Until .EOF
Combo2.Text = txtsurname.Text
End With
Exit Sub
Else
.MoveNext
End If
Loop Until .EOF
MsgBox ("ไม่พบข้อมูลที่ต้องการหา"), (vbOKOnly), ("Message")
End With
End Sub

Private Sub Combo1_GotFocus()
Combo1.Text = Trim(Combo1.Text)
End Sub

Private Sub Combo2_Click()
With DataEnvironment3.rsCommand1
Dim i As Integer
i = 0
.MoveFirst
Do

```

```

If (Combo2.Text = txtsurname.Text) And (Combo1.Text = txtname.Text) Then
Exit Sub
Else
.MoveNext
End If

Loop Until .EOF
MsgBox ("ไม่พบข้อมูลที่ต้องการหา"), (vbOKOnly), ("Message")

End With
End Sub

Private Sub Combo2_GotFocus()
Combo2.Text = Trim(Combo2.Text)
End Sub

Private Sub Command1_Click()
Form10.Hide
Form8.Show
End Sub

Private Sub Command2_Click()
With DataEnvironment3.rsCommand1
MsgBox ("ท่านต้องการเพิ่มเติมข้อมูลใช่หรือไม่"), (vbOKCancel), ("เพิ่มเติม")
Dim i As Integer
.AddNew
'teacherID.Text = Val(Text1.Text)
Combo1.Text = Trim("")
Combo2.Text = Trim("")
End With
End Sub

```

```

Private Sub Command3_Click()
    With DataEnvironment3.rsCommand1
        txtname.Text = Combo1.Text
        txtsurname.Text = Combo2.Text
        .Fields("teacherID").Value = teacherID
        .Fields("name").Value = txtname
        .Fields("surname").Value = txtsurname
        .Fields("positionID").Value = txtpositionID
        .Update
    End With
    With DataEnvironment3.rsCommand1
        i = 0
        .MoveFirst
        Do
            Combo1.List(i) = txtname.Text
            .MoveNext
            i = i + 1
            Text1.Text = Val(i + 1)
        Loop Until .EOF
        Combo1.Text = txtname.Text
        Combo2.Text = ""
    End With
    .Update
    .MoveNext
    If .EOF Then
        .MoveLast
    End If
    MsgBox ("เสร็จสิ้นการแก้ไข"), (vbOKOnly), ("แก้ไข")
    End With
End Sub

Private Sub Command4_Click()

```

```

Form1.Show
Form10.Hide
End Sub

Private Sub Command5_Click()
End
End Sub

Private Sub Command6_Click()
With DataEnvironment3.rsCommand1
MsgBox ("ท่านได้ลบข้อมูลที่มีรหัสวิชานี้อยู่หมดแล้วใช่หรือไม่"), (vbOKCancel), ("Delete")
MsgBox ("ท่านต้องการลบข้อมูลใช่หรือไม่"), (vbOKCancel), ("Delete")
.Delete
.MoveNext
If .EOF Then
.MoveLast
End If
With DataEnvironment3.rsCommand1
i = 0
.MoveFirst
Do
Combo1.List(i) = txtname.Text
.MoveNext
i = i + 1
Text1.Text = Val(i + 1)
Loop Until .EOF
Combo1.Text = txtname.Text
End With
End With
End Sub

```

```

Private Sub Form_Load()
With DataEnvironment3.rsCommand1
Dim i, n As Integer
Dim listtext As String
Dim CheckInData As Boolean

i = 0
n = 0
.MoveFirst
Do
CheckInData = False
For n = 0 To i
If Trim(txtname.Text) = Trim(Combo1.List(n)) Then
n = i
CheckInData = True
End If
Next n
If CheckInData = True Then
i = i - 1
Else
Combo1.List(i) = txtname.Text
End If
.MoveNext
i = i + 1
Text1.Text = Val(i + 1)
Loop Until .EOF
End With
End Sub

Private Sub teacherID_GotFocus()
teacherID.Text = Trim(teacherID.Text)

```

```
End Sub
```

```
Private Sub txtpositionID_GotFocus()
```

```
txtpositionID.Text = Trim(txtpositionID.Text)
```

```
End Sub
```

```
Private Sub txtsurname_GotFocus()
```

```
txtsurname.Text = Trim(txtsurname.Text)
```

```
End Sub
```

```
////////////////////////////////////
```

```
//// form 11 Addnew student //////////////////////////////////////
```

```
Private Sub Combo1_Click()
```

```
With DataEnvironment3.rsCommand2
```

```
On Error GoTo handleerror
```

```
.MoveFirst
```

```
Do
```

```
If (Combo1.Text = txtbranchName.Text) Then
```

```
Exit Sub
```

```
Else
```

```
.MoveNext
```

```
End If
```

```
Loop Until .EOF
```

```
MsgBox ("กรุณาใส่ข้อมูลให้ครบ"), (vbOKOnly), ("Message")
```

```
handleerror:
```

```
If MsgBox(Error(ErrorNumber), [vbOKCancel]) = vbOK Then
```

```
With DataEnvironment3.rsCommand2
```

```
Exit Sub
```

```
End With
```

```

End If
End With
Combo1.Text = Combo1.List(0)
End Sub

Private Sub Combo1_GotFocus()
Combo1.Text = Trim(Combo1.Text)
End Sub

Private Sub Combo2_Click()
With DataEnvironment3.rsCommand2
On Error GoTo handleerror
.MoveFirst
Do
If (Combo1.Text = txtbranchName.Text) And (Val(Combo2.Text) = Val(txtyear.Text))
Then
Exit Sub
Else
.MoveNext
End If
Loop Until .EOF
MsgBox ("กรุณาใส่ข้อมูลให้ครบ"), (vbOKOnly), ("Message")
handleerror:
If MsgBox("ข้อมูลถูกต้องใช่ไหม", [vbOKCancel], (Message)) = vbOK Then
With DataEnvironment3.rsCommand2
Exit Sub
End With
End If
End With
Combo2.Text = Combo2.List(0)

```

```
End Sub
```

```
Private Sub Combo2_GotFocus()
```

```
Combo2.Text = Trim(Combo2.Text)
```

```
End Sub
```

```
Private Sub Combo3_Click()
```

```
With DataEnvironment3.rsCommand2
```

```
On Error GoTo handleerror
```

```
.MoveFirst
```

```
Do
```

```
If (Combo1.Text = txtbranchName.Text) And (Val(Combo2.Text) = Val(txtyear.Text))
```

```
And (Val(Combo3.Text) = Val(txtclassroom.Text)) Then
```

```
Exit Sub
```

```
Else
```

```
.MoveNext
```

```
End If
```

```
Loop Until .EOF
```

```
MsgBox ("ไม่พบข้อมูลที่ต้องการหา"), (vbOKOnly), ("Message")
```

```
handleerror:
```

```
If MsgBox("ข้อมูลถูกต้องใช่หรือไม่", [vbOKCancel], (Message)) = vbOK Then
```

```
With DataEnvironment3.rsCommand2
```

```
Exit Sub
```

```
End With
```

```
End If
```

```
End With
```

```
End Sub
```

```
Private Sub Combo3_GotFocus()
```

```
Combo3.Text = Trim(Combo3.Text)
```

```

End Sub

Private Sub Command1_Click()
With DataEnvironment3.rsCommand2

On Error GoTo handleerror

txtbranchName = Combo1.Text
txtyear = Combo2.Text
txtclassroom = Combo3.Text
.Fields("branchID").Value = Trim(txtbranchID)
.Fields("branchName").Value = Trim(txtbranchName)
.Fields("year").Value = Trim(txtyear)
.Fields("classroom").Value = Trim(txtclassroom)
.Fields("departmentName").Value = Trim(txtdepartmentName)

With DataEnvironment3.rsCommand2

Dim n, i As Integer
Dim listtext As String
Dim CheckInData As Boolean

i = 0
n = 0
.MoveFirst

Do
CheckInData = False

For n = 0 To i

If Trim(txtbranchName.Text) = Trim(Combo1.List(n)) Then

n = i

CheckInData = True

End If

Next n

If CheckInData = True Then

```

```

i = i - 1

Else

Combo1.List(i) = txtbranchName.Text

End If

.MoveNext

        i = i + 1

        Text1.Text = Val(i + 1)

Loop Until .EOF

End With

Combo1.Text = ""

Combo2.Text = ""

Combo3.Text = ""

.Update

If MsgBox("การแก้ไข/เพิ่มเติม ข้อมูลเรียบร้อยแล้วท่านต้องการเพิ่มข้อมูลอีกหรือไม่",
[vbOKCancel]) = vbOK Then

    With DataEnvironment3.rsCommand2

        .AddNew

            Combo1.Text = ""

            Combo2.Text = ""

            Combo3.Text = ""

        End With

    End If

    Exit Sub

handleerror:

    If MsgBox("ข้อมูลถูกต้องไหมหรือไม่", [vbOKCancel], (Message)) = vbOK Then

        With DataEnvironment3.rsCommand2

            .Close

        End With

    End If

Exit Sub

```

```
End With
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
On Error GoTo handleerror
```

```
With DataEnvironment3.rsCommand2
```

```
If MsgBox("ท่านต้องการเพิ่มข้อมูลใช่หรือไม่", [vbOKCancel]) = vbOK Then
```

```
With DataEnvironment3.rsCommand2
```

```
.AddNew
```

```
Combo1.Text = ""
```

```
Combo2.Text = ""
```

```
Combo3.Text = ""
```

```
End With
```

```
End If
```

```
Exit Sub
```

```
End With
```

```
handleerror:
```

```
If MsgBox("ข้อมูลถูกต้องใช่หรือไม่", [vbOKCancel], (Message)) = vbOK Then
```

```
With DataEnvironment3.rsCommand2
```

```
Exit Sub
```

```
End With
```

```
End If
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
With DataEnvironment3.rsCommand2
```

```
On Error GoTo handleerror
```

```
MsgBox ("ท่านได้ลบข้อมูลที่มีรหัสวิชานี้อยู่หมดแล้วใช่หรือไม่"), (vbOKCancel), ("Delete")
```

```
MsgBox ("ท่านต้องการลบข้อมูลใช่หรือไม่"), (vbOKCancel), ("Delete")
```

```
.Delete
```

```

.MoveNext

    If .EOF Then
        .MoveLast
    End If

With DataEnvironment3.rsCommand2

Dim n As Integer
Dim i As Integer
Dim listtext As String
Dim CheckInData As Boolean

i = 0
n = 0
.MoveNext
Do
    CheckInData = False
    For n = 0 To i
        If Trim(txtbranchName.Text) = Trim(Combo1.List(n)) Then
            n = i
            CheckInData = True
        End If
    Next n
    If CheckInData = True Then
        i = i - 1
    Else
        Combo1.List(i) = txtbranchName.Text
    End If
.MoveNext
    i = i + 1
Text1.Text = Val(i + 1)
Loop Until .EOF

```

```

End With

Combo1.Text = ""
Combo2.Text = ""
Combo3.Text = ""
End With

handleerror:
If MsgBox("ข้อมูลถูกต้องใช่ไหม", [vbOKCancel], (Message)) = vbOK Then
With DataEnvironment3.rsCommand2
Exit Sub
End With
End If
End Sub

Private Sub Command4_Click()
End
End Sub

Private Sub Command5_Click()
Form1.Hide
Form1.Show
End Sub

Private Sub Command6_Click()
Form1.Hide
Form8.Show
End Sub

Private Sub Form_Load()
With DataEnvironment3.rsCommand2
Dim n, i As Integer

```

```

Dim listtext As String
Dim CheckInData As Boolean

i = 0
n = 0
.MoveFirst

Do
CheckInData = False
For n = 0 To i
If Trim(txtbranchName.Text) = Trim(Combo1.List(n)) Then
n = i
CheckInData = True
End If
Next n
If CheckInData = True Then
i = i - 1
Else
Combo1.List(i) = txtbranchName.Text
End If
.MoveNext
    i = i + 1
    Text1.Text = Val(i + 1)
Loop Until .EOF

End With

End Sub

Private Sub txtbranchID_GotFocus()
txtbranchID.Text = Trim(txtbranchID.Text)
End Sub

Private Sub txtdepartmentName_GotFocus()

```

```

txtdepartmentName.Text = Trim(txtdepartmentName.Text)

End Sub

////////////////////////////////////

//// form 12 Addnew-Delete-update subject //////////////////////////////////

Private Sub Combo1_Click()
With DataEnvironment3.rsCommand3
.MoveFirst
n = 0
Do
If Combo1.Text = txtsubjectID.Text Then
Exit Sub
Else
.MoveNext
End If
Loop Until .EOF
MsgBox ("ไม่พบข้อมูลที่ต้องการหา"), (vbOKOnly), ("Message")
End With
End Sub

Private Sub Combo1_GotFocus()
Combo1.Text = Trim(Combo1.Text)
End Sub

Private Sub Command1_Click()
With DataEnvironment3.rsCommand3
txtsubjectID.Text = Combo1.Text
.Fields("subjectID").Value = txtsubjectID
.Fields("subjectName").Value = txtsubjectName
.Fields("credit").Value = txtcredit

```

```

.Fields("theoryHour").Value = txttheoryHour

.Fields("practiceHour").Value = txtpracticeHour

With DataEnvironment3.rsCommand3

    i = 0

    .MoveFirst

    Do

        Combo1.List(i) = txtsubjectID.Text

        .MoveNext

        i = i + 1

    Loop Until .EOF

End With

If MsgBox("การแก้ไข/เพิ่มเติม ข้อมูลเรียบร้อยแล้วท่านต้องการเพิ่มข้อมูลอีกหรือไม่",
[vbOKCancel]) = vbOK Then

    With DataEnvironment3.rsCommand3

        .AddNew

        Combo1.Text = ""

        Combo2.Text = ""

        Combo3.Text = ""

    End With

End If

End With

End Sub

Private Sub Command2_Click()

With DataEnvironment3.rsCommand3

Combo1.Text = ""

.AddNew

Dim n, i As Integer

Dim listtext As String

Dim CheckInData As Boolean

```

```

i = 0

n = 0

.MoveFirst

Do

CheckInData = False

For n = 0 To i

If Trim(txtsubjectID.Text) = Trim(Combo1.List(n)) Then

n = i

CheckInData = True

End If

Next n

If CheckInData = True Then

i = i - 1

Else

Combo1.List(i) = txtsubjectID.Text

End If

.MoveNext

i = i + 1

Text1.Text = Val(i + 1)

Loop Until .EOF

MsgBox ("ท่านต้องการเพิ่มเติมข้อมูลนี้ใช่หรือไม่"), (vbOKCancel), ("เพิ่มเติม")

End With

End Sub

Private Sub Command3_Click()

Form12.Hide

Form8.Show

End Sub

Private Sub Command4_Click()

```

```
End
```

```
End Sub
```

```
Private Sub Command5_Click()
```

```
Form1.Show
```

```
Form12.Hide
```

```
End Sub
```

```
Private Sub Command6_Click()
```

```
With DataEnvironment3.rsCommand3
```

```
MsgBox ("ท่านได้ลบข้อมูลที่มีรหัสวิชานี้อยู่หมดแล้วใช่หรือไม่"), (vbOKCancel), ("Delete")
```

```
MsgBox ("ท่านต้องการลบข้อมูลใช่หรือไม่"), (vbOKCancel), ("Delete")
```

```
.Delete
```

```
.MoveNext
```

```
If .EOF Then
```

```
.MoveLast
```

```
End If
```

```
Dim i As Integer
```

```
Dim n As Integer
```

```
Dim listtext As String
```

```
Dim CheckInData As Boolean
```

```
With DataEnvironment3.rsCommand3
```

```
i = 0
```

```
n = 0
```

```
.MoveFirst
```

```
Do
```

```
CheckInData = False
```

```
For n = 0 To i
```

```
If Trim(txtsubjectID.Text) = Trim(Combo1.List(n)) Then
```

```
n = i
```

```

CheckInData = True

End If

Next n

If CheckInData = True Then

    i = i - 1

Else

    Combo1.List(i) = txtsubjectID.Text

End If

.MoveNext

    i = i + 1

    Text1.Text = Val(i + 1)

Loop Until .EOF

Combo1.Text = txtsubjectID.Text

End With

End With

End Sub

Private Sub Form_Load()

With DataEnvironment3.rsCommand3

i = 0

.MoveNext

Do

Combo1.List(i) = txtsubjectID.Text

.MoveNext

i = i + 1

Loop Until .EOF

End With

End Sub

```

```

Private Sub txtsubjectName_GotFos()
End Sub

Private Sub txtpracticeHour_GotFocus()
txtpracticeHour.Text = Trim(txtpracticeHour.Text)
End Sub

Private Sub txtsubjectName_GotFocus()
txtsubjectName.Text = Trim(txtsubjectName.Text)
End Sub

////////////////////////////////////

//// form14 Addnew-Delete-update teach22 //////////////////////////////////

Private Sub addressroom_GotFocus()
addressroom.Text = Trim("")
End Sub

Private Sub Combo10_Click()
With DataEnvironment3.rsstudent
On Error GoTo handleerror
.MoveFirst
Do
If (Combo8.Text = txtbranchName.Text) And (Val(Combo9.Text) = Val(txtyear.Text))
And (Val(Combo10.Text) = Val(txtclassroom.Text)) Then
branchID.Text = branchID1.Text
Exit Sub
Else
.MoveNext
End If
Loop Until .EOF

```

```

MsgBox ("ไม่พบข้อมูลที่ต้องการหา"), (vbOKOnly), ("Message")

handleerror:
If MsgBox("ข้อมูลถูกต้องไหมหรือไม่", [vbOKCancel], (Message)) = vbOK Then
    With DataEnvironment3.rsstudent
    Exit Sub
    End With
End If
End With
End Sub

Private Sub Combo11_Click()
With DataEnvironment3.rssubject
.MoveFirst
n = 0
Do
If Combo11.Text = subjectID1.Text Then
    subjectID.Text = subjectID1.Text
    Exit Sub
    Else
    .MoveNext
    End If
    Loop Until .EOF
MsgBox ("ไม่พบข้อมูลที่ต้องการหา"), (vbOKOnly), ("Message")
End With
End Sub

Private Sub Combo11_GotFocus()
Combo11.Text = Trim("")
End Sub

```

```
Private Sub Combo12_Click()
txtdate.Text = Combo12.Text
End Sub
```

```
Private Sub Combo3_Click()
txtstime.Text = Combo3
End Sub
```

```
Private Sub Combo4_Click()
txtltime.Text = Combo4
End Sub
```

```
Private Sub Combo5_Click()
With DataEnvironment3.rsssubject
.MoveFirst
n = 0
Do
If Combo5.Text = subjectID1.Text Then
subjectID.Text = subjectID1.Text
Exit Sub
Else
.MoveNext
End If
Loop Until .EOF
MsgBox ("ไม่พบข้อมูลที่ต้องการหา"), (vbOKOnly), ("Message")
End With
End Sub
```

```
Private Sub Combo5_GotFocus()
Combo5.Text = Trim(Combo5.Text = "")
```

```

End Sub

Private Sub Combo6_Click()
With DataEnvironment3.rsteacher
Dim i As Integer

i = 0
.MoveFirst

Do
If (Combo6.Text = txtname.Text) Then
With DataEnvironment3.rsteacher
i = 0
.MoveFirst
Do
Combo7.List(i) = txtsurname.Text
.MoveNext

i = i + 1
Loop Until .EOF
Combo7.Text = txtsurname.Text
End With
TeacherID.Text = teacherID1.Text

Exit Sub
Else
.MoveNext
End If

Loop Until .EOF
MsgBox ("ไม่พบข้อมูลที่ต้องการหา"), (vbOKOnly), ("Message")
End With
End Sub

Private Sub Combo7_Click()

```

```

With DataEnvironment3.rsteacher

Dim i As Integer

i = 0

.MoveFirst

Do

If (Combo7.Text = txtsurname.Text) And (Combo6.Text = txtname.Text) Then

teacherID.Text = teacherID1.Text

Exit Sub

Else

.MoveNext

End If

Loop Until .EOF

MsgBox ("ไม่พบข้อมูลที่ต้องการหา"), (vbOKOnly), ("Message")

End With

End Sub

Private Sub Combo8_Click()

With DataEnvironment3.rsstudent

On Error GoTo handleerror

.MoveFirst

Do

If (Combo8.Text = txtbranchName.Text) Then

branchID.Text = branchID1.Text

Exit Sub

Else

.MoveNext

End If

Loop Until .EOF

MsgBox ("กรุณาใส่ข้อมูลให้ครบ"), (vbOKOnly), ("Message")

handleerror:

```

```

If MsgBox(Error(ErrorNumber), [vbOKCancel]) = vbOK Then
With DataEnvironment3.rsstudent
Exit Sub
End With
End If

End With

Combo1.Text = Combo1.List(0)
End Sub

Private Sub Combo9_Click()
With DataEnvironment3.rsstudent
On Error GoTo handleerror
.MoveFirst
Do
If (Combo8.Text = txtbranchName.Text) And (Val(Combo9.Text) = Val(txtyear.Text))
Then
branchID.Text = branchID1.Text
Exit Sub
Else
.MoveNext
End If
Loop Until .EOF
MsgBox ("กรุณาใส่ข้อมูลให้ครบ"), (vbOKOnly), ("Message")
handleerror:
If MsgBox("ข้อมูลถูกต้องใช่หรือไม่", [vbOKCancel], (Message)) = vbOK Then
With DataEnvironment3.rsstudent
Exit Sub
End With
End If
End With

```

```
End Sub
```

```
Private Sub Command4_Click()
```

```
With DataEnvironment3.rsteach22
```

```
    .MovePrevious
```

```
If .BOF Then
```

```
    .MoveFirst
```

```
End If
```

```
End With
```

```
End Sub
```

```
Private Sub Command5_Click()
```

```
With DataEnvironment3.rsteach22
```

```
    .MoveNext
```

```
If .EOF Then
```

```
    .MoveLast
```

```
End If
```

```
End With
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
If MsgBox("ท่านต้องการลบข้อมูลใช่หรือไม่", [vbOKCancel]) = vbOK Then
```

```
With DataEnvironment3.rsteach22
```

```
    .Delete
```

```
    .MoveNext
```

```
If .EOF Then
```

```
    .MoveLast
```

```
End If
```

```
End With
```

```
End If
```

```
Exit Sub
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
With DataEnvironment3.rsteach22
```

```
.Fields("teacherID").Value = teacherID
```

```
.Fields("subjectID").Value = subjectID
```

```
.Fields("branchID").Value = branchID
```

```
.Fields("date").Value = txtdate
```

```
.Fields("stime").Value = txtstime
```

```
.Fields("ltime").Value = txtltime
```

```
.Fields("room").Value = txtroom
```

```
.Update
```

```
MsgBox ("การแก้ไขข้อมูลเรียบร้อยแล้ว"), (vbOKOnly), (Message)
```

```
End With
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
With DataEnvironment3.rsteach22
```

```
MsgBox ("ท่านต้องการเพิ่มข้อมูลใช่หรือไม่"), (vbOKOnly)
```

```
addroom.Text = ""
```

```
.MoveNext
```

```
If .EOF Then
```

```
.MoveLast
```

```
End If
```

```
.AddNew
```

```
End With
```

```
End Sub
```

```
Private Sub Command6_Click()
```

```

With DataEnvironment3.rsteach22
MsgBox ("เสร็จสิ้นการเพิ่มข้อมูล"), (vbOKOnly)

.Fields("teacherID").Value = teacherID
.Fields("subjectID").Value = subjectID
.Fields("branchID").Value = branchID

.Fields("date").Value = txtdate
.Fields("stime").Value = txtstime
.Fields("ltime").Value = txtltime
.Fields("room").Value = addroom

.Update
addroom = Trim("")
End With
End Sub

Private Sub Command7_Click()
Form14.Hide
Form1.Show
End Sub

Private Sub Command8_Click()
End
End Sub

Private Sub Command9_Click()
Form14.Hide
Form8.Show
End Sub

Private Sub Form_Load()
Dim i As Integer

```

```

With DataEnvironment3.rsteach22
i = 0
.MoveFirst
Do
    Combo1.List(i) = txtroom.Text
    .MoveNext
    i = i + 1
Loop Until .EOF
End With
With DataEnvironment3.rssubject
i = 0
.MoveFirst
Do
    Combo11.List(i) = subjectID1.Text
    .MoveNext
    i = i + 1
Loop Until .EOF
End With
With DataEnvironment3.rsteacher
Dim n As Integer
Dim listtext As String
Dim CheckInData As Boolean
i = 0
n = 0
.MoveFirst

Do
CheckInData = False
For n = 0 To i
    If Trim(txtname.Text) = Trim(Combo6.List(n)) Then

```

```

n = i
CheckInData = True
End If
Next n
If CheckInData = True Then
    i = i - 1
Else
    Combo6.List(i) = txtname.Text
End If
.MoveNext
    i = i + 1
    Text1.Text = Val(i + 1)
Loop Until .EOF
End With
With DataEnvironment3.rsstudent
Dim n, i As Integer
Dim listtext As String
Dim CheckInData As Boolean
i = 0
n = 0
.MoveNext
Do
    CheckInData = False
    For n = 0 To i
        If Trim(txtbranchName.Text) = Trim(Combo8.List(n)) Then
            n = i
            CheckInData = True
        End If
    Next n
    If CheckInData = True Then

```

```
i = i - 1  
Else  
    Combo8.List(i) = txtbranchName.Text  
End If  
.MoveNext  
  
i = i + 1  
Loop Until .EOF  
End With  
End Sub  
////////////////////////////////////
```



บรรณานุกรม

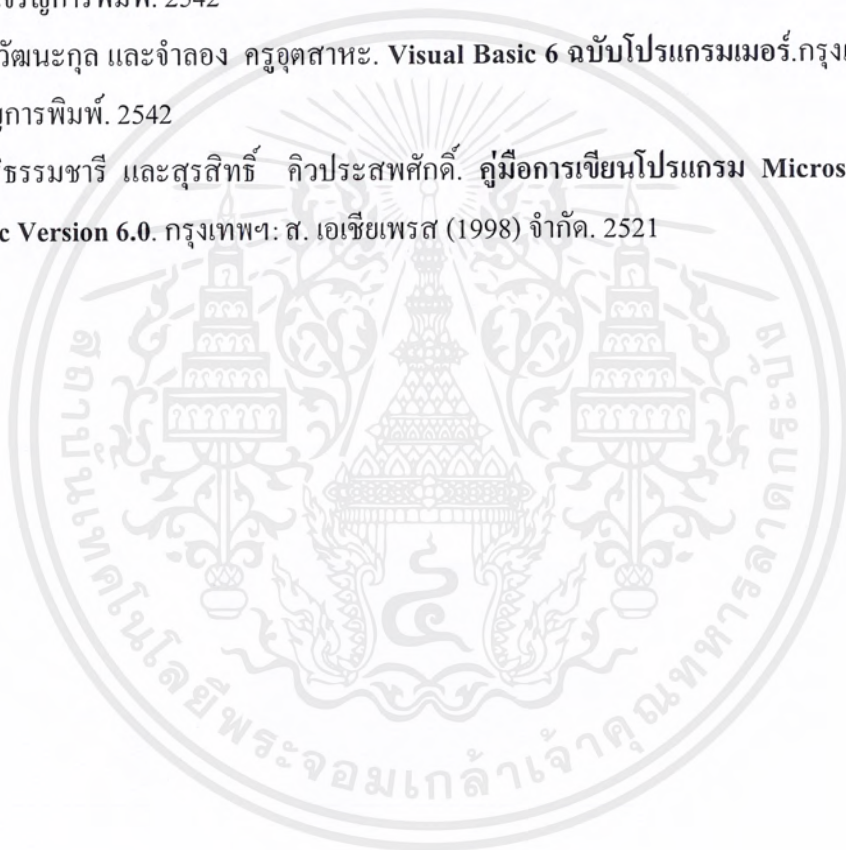
บัณฑิต จามรภูมิ. การใช้งานฐานข้อมูลเชิงสัมพันธ์ Microsoft SQL Server. กรุงเทพฯ:ว.พีเชร
สกล. 2541

บัณฑิต จามรภูมิ. ฐานข้อมูล Microsoft SQL Server 7.0. กรุงเทพฯ:บัณฑิตเพรส. 2543

กิตติ ภัคดีวิวัฒน์กุล และจำลอง ครูอุตสาหะ. การออกแบบฐานข้อมูล Database Design.กรุงเทพฯ:
ไทยเจริญการพิมพ์. 2542

กิตติ ภัคดีวิวัฒน์กุล และจำลอง ครูอุตสาหะ. Visual Basic 6 ฉบับโปรแกรมเมอร์.กรุงเทพฯ: ไทย
เจริญการพิมพ์. 2542

ธาริน สิทธิธรรมชารี และสุรสิทธิ์ กิวประสพศักดิ์. คู่มือการเขียนโปรแกรม Microsoft Visual
Basic Version 6.0. กรุงเทพฯ: ส. เอเชียเพรส (1998) จำกัด. 2521



ประวัติผู้แต่ง



ชื่อผู้ทำปฏิญญาพันธ	นางสาวอรไท ก้อนมณี
วันเดือนปีเกิด	8 พฤษภาคม 2522
สถานที่เกิด	ลพบุรี
ภูมิลำเนาเดิม	จังหวัดลพบุรี
ที่อยู่ปัจจุบัน	103 ม.2 ต. พรหมมาستر อ.เมือง จ.ลพบุรี 15000
โทรศัพท์	(036) 617591
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนบรรจงรัตน์
มัธยมศึกษาตอนต้น	โรงเรียนวินิตศึกษา
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคลพบุรี
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคลพบุรี
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับ	-
คติพจน์	อย่าพึ่งท้อแท้เมื่อยังไม่ลงมือทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปฏิญานិพนธ์	นายยุทธวิท ชูสวน
วันเดือนปีเกิด	28 มกราคม 2522
สถานที่เกิด	กระบี่
ภูมิลำเนาเดิม	จังหวัดกระบี่
ที่อยู่ปัจจุบัน	141 ม.2 ต. เหนือคลอง อ.เหนือคลอง จ.กระบี่ 81130
โทรศัพท์	(075) 636441
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนราชประชานุเคราะห์ 1
มัธยมศึกษาตอนต้น	โรงเรียนเหนือคลองประชานำรุง
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคกระบี่
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคกระบี่
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับ	-
คติพจน์	พยายามทำทุกสิ่งให้ดีที่สุดเท่าที่จะทำได้

ประวัติผู้แต่ง



ชื่อผู้ทำปฏิญานិพนธ์	นายอุทัย ตำรวมจิตร
วันเดือนปีเกิด	31 มีนาคม 2519
สถานที่เกิด	สุพรรณบุรี
ภูมิลำเนาเดิม	จังหวัดสุพรรณบุรี
ที่อยู่ปัจจุบัน	98 ถ.ราษฎร์นิยม ต.สองพี่น้อง อ.สองพี่น้อง จ.สุพรรณบุรี 72110
โทรศัพท์	(035) 531846
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนเทศบาล1 (พานิชอุทิศ)
มัธยมศึกษาตอนต้น	โรงเรียนบางลี่วิทยา
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคสุพรรณบุรี
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคสุพรรณบุรี
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับ	-
คติพจน์	อดทน และพยายาม