

ระบบการจัดการขนส่งสินค้าทางเรือ

Containered Shipping Line System



จักรพล ศิริจันทร์
ทิมนท์ ลำคงคา
สมลักษณ์ พงษ์วิไลย์

รฟท.
จ 224 ร

เลขหน้ 2544
เลขทะเบียน 43030
วัน, เดือน, ปี 26 ส.ย. 2545

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

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2544

Containered Shipping Line System



A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT OR THE DEGREE OF BACHELOR OF SCIENCE
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2001




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ ระบบการจัดการขนส่งสินค้าทางเรือ
CONTAINERED SHIPPING LINE SYSTEM

ชื่อนักศึกษา นายจักรพล ศิริจันทร์ 41056010
นางสาวทิมณฑิ์ ลำคงคา 41056032
นางสาวสมลักษณ์ พงษ์วิสัย 41056114

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขาวิชา วิทยาการคอมพิวเตอร์
ปีการศึกษา 2544
อาจารย์ที่ปรึกษา อาจารย์ วีระชัย ตันยะสิทธิ์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2544

	คณะกรรมการสอบ	ลายมือชื่อ
ประธานกรรมการ	อาจารย์ วิสันต์ ตั้งวงษ์เจริญ	
กรรมการ	อาจารย์ สิริลักษณ์ อนันต์สถิตย์สิน	
กรรมการและอาจารย์ที่ปรึกษา	อาจารย์ วีระชัย ตันยะสิทธิ์	



(ผู้ช่วยศาสตราจารย์ไพโรจน์ พันธ์รักษ์พงษ์)

หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	ระบบระบบการจัดการขนส่งสินค้าทางเรือ	
ชื่อนักศึกษา	นายจักรพล ศิริจันทร์	41056010
	นางสาวทิมณฑิ์ ลำคงคา	41056032
	นางสาวสมลักษณ์ พงษ์วิสัย	41056114
ปริญญา	วิทยาศาสตรบัณฑิต	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2544	
อาจารย์ที่ปรึกษา	อาจารย์ วีระชัย ต้นยะสิทธิ์	

บทคัดย่อ

ในปัจจุบันการประกอบธุรกิจไม่ได้จำกัดอยู่แค่ภายในประเทศ มีการติดต่อธุรกิจไปทั่วทุกมุมโลก มีการขนส่งสินค้าระหว่างประเทศทั้งธุรกิจขนาดใหญ่และขนาดเล็ก ซึ่งในการขนส่งสินค้าที่มีน้ำหนักและมีปริมาณมาก การขนส่งทางเรื่อนั้นเป็นทางเลือกที่ดีที่สุด การขนส่งทางเรือมีการแข่งขันในเรื่องความเร็วและความปลอดภัยในการขนส่ง ถ้าหากส่งสินค้าไม่ทันตามกำหนดเวลาก็จะต้องเสียค่าชดเชยให้กับลูกค้า ในการเดินเรือแต่ละครั้งนั้นเสียค่าใช้จ่ายสูง จึงจำเป็นต้องจัดการบริหารการขนส่งสินค้าให้เหมาะสมและเป็นระเบียบแบบแผน เพื่อลดค่าใช้จ่ายให้มากที่สุด

ดังนั้นระบบการจัดการขนส่งสินค้าทางเรือ (Containered Shipping Line System) จึงได้ถูกพัฒนาขึ้นเพื่อช่วยอำนวยความสะดวกในการเก็บรวบรวมข้อมูล ประมวลผลข้อมูลตามขั้นตอนการทำงานต่างๆ อีกทั้งการแสดงผลข้อมูลการขนส่งสินค้าทางจอภาพ เพื่อประโยชน์ในด้านการจัดการขนส่งสินค้าของผู้ใช้

Special Project Title	Containered Shipping Line System		
Students	Mr. Jakrapol	Sirichantr	41056010
	Miss. Thimon	Lomkongka	41056032
	Miss. Somluk	Phongwilai	41056114
Degree	Bachelor's Degree of Science		
Department	Mathematics and Computer Science, Faculty of Science		
Programme	Computer Science		
Academic Year	2001		
Special Project Advisor	Lecturer Weerachai Tanyasit		

ABSTRACT

According to the business system now a day does not only limited in a country but there are many contract around the world. International import and export include large and small business has been developing. One of the best choice to due it was shipping that because of the lower cost and more comfortable. Causes there are many companies so the customer can choose any one. In the business competition time and security are very important. If some product did not delivery on time then the customer will gone. By the way the investment of shipping was highly so they need the suitable management to decrease these number and make a lot of profit.

Containered Shipping Line System is the software that is developed for manage the collection of manipulate data and processing systematically. Further more to represent the hiding data for the vision of the user.

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่องระบบการขนส่งสินค้าทางเรือสามารถสำเร็จลุล่วงไปได้ด้วยดี คณะผู้จัดทำต้องขอขอบพระคุณ อาจารย์ วีระชัย ตัญยะสิทธิ์ อาจารย์ผู้รับผิดชอบปัญหาพิเศษฉบับนี้ที่กรุณาให้คำแนะนำและเป็นพี่ปรึกษาในการแก้ปัญหาต่างๆรวมทั้งเป็นผู้ตรวจสอบความถูกต้องของปัญหาพิเศษฉบับนี้

นอกจากนี้คณะผู้จัดทำต้องขอขอบพระคุณ บิดา มารดา ที่ได้ให้ความสนับสนุนทางด้านกำลังใจและทุนทรัพย์ จนการทำปัญหาพิเศษครั้งนี้สำเร็จลุล่วงด้วยดี รวมทั้งเพื่อนๆทุกคนที่ให้ความช่วยเหลือในด้านต่างๆเกี่ยวกับปัญหาพิเศษ ไว้ ณ ที่นี้

คณะผู้จัดทำ
มีนาคม 2545



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VIII
สารบัญภาพ.....	IX
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 สมมติฐานของการศึกษา.....	1
1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการศึกษา.....	2
1.5 ขอบเขตของการศึกษา.....	2
1.6 ขั้นตอนในการดำเนินงาน.....	2
บทที่ 2 ทฤษฎีและหลักเกณฑ์ที่เกี่ยวข้อง.....	3
2.1 ความรู้พื้นฐานเกี่ยวกับฐานข้อมูล.....	3
2.2 ความสามารถของ VB6 กับการจัดการฐานข้อมูล.....	4
2.3 รูปแบบการติดต่อกับฐานข้อมูลด้วย Visual Basic.....	5
2.3.1 การติดต่อฐานข้อมูลใน Visual Basic.....	5
2.3.2 การใช้งานคอนโทรล ADO Data ร่วมกับคอนโทรล TextBox	8
2.4 ADO Data Control.....	12
2.4.1 พื้นฐานการเขียนโปรแกรมด้วยเทคโนโลยี ActiveX Data Object.....	12
2.4.2 เทคโนโลยี ActiveX Data Object.....	14
2.4.3 ชุดออบเจกต์ใน ADO 2.5.....	15
2.5 ออบเจกต์ Connection.....	17
2.5.1 วิธีการเชื่อมต่อกับฐานข้อมูลโดยใช้คุณสมบัติ ConnectionString.....	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.6 ออบเจกต์ Command.....	18
2.6.1 วิธีการเชื่อมต่อกับฐานข้อมูลด้วยออบเจกต์ Command.....	18
2.6.2 วิธีการสร้างออบเจกต์ Recordset ด้วยออบเจกต์ Command.....	19
2.7 ออบเจกต์ Recordset.....	20
2.7.1 วิธีการจัดการจำนวนเรคคอร์ด.....	22
2.7.2 วิธีการอ้างค่าที่อยู่ในฟิลด์.....	23
2.8 การใช้งานภาษา SQL.....	24
2.8.1 โครงสร้างของภาษา SQL.....	24
2.8.2 ตัวดำเนินการ (Operator).....	25
2.8.3 ตัวดำเนินการด้านตรรกะ (Logical Operator).....	25
2.8.4 กลุ่มฟังก์ชัน Aggregate.....	25
2.9 ต้นไม้ (Trees)	26
2.9.1 การแทนที่ต้นไม้แบบไดนามิก.....	27
2.9.2 การท่องต้นไม้ไบนารีโดยใช้สแตค.....	29
2.10 การสร้างและการจัดการระบบฐานข้อมูล SQL Server 7.0.....	31
2.10.1 การจัดการระบบฐานข้อมูล.....	32
2.10.2 การสร้างฐานข้อมูล.....	32
2.10.3 การสร้างตารางและกำหนดประเภทของข้อมูล.....	37
บทที่ 3 วิธีการดำเนินงานวิจัยและออกแบบระบบ.....	42
3.1 Functional Decomposition Diagram.....	42
3.2 Context Diagram.....	43
3.3 Data Flow Diagram ระดับ 0.....	44
3.4 Data Flow Diagram ระดับ 1 กระบวนการที่ 1 SET UP.....	45
3.5 Data Flow Diagram ระดับ 1 กระบวนการที่ 2 จัดการข้อมูลสินค้า.....	46
3.6 Data Flow Diagram ระดับ 1 กระบวนการที่ 3 จัดการตารางการขนถ่ายสินค้า.....	47
3.7 Data Flow Diagram ระดับ 1 กระบวนการที่ 4 แสดงข้อมูลการขนถ่ายสินค้า.....	48
3.8 Data Flow Diagram ระดับ 2 กระบวนการที่ 1.1 จัดการข้อมูลแผนการเดินทาง.....	49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
3.9 Data Flow Diagram ระดับ 2 กระบวนการที่ 1.2 จัดการข้อมูลเส้นทางเดินเรือ.....	50
3.10 Data Flow Diagram ระดับ 2 กระบวนการที่ 1.3 จัดการข้อมูลเรือ.....	51
3.11 Data Flow Diagram ระดับ 2 กระบวนการที่ 1.4 จัดการข้อมูลท่าเรือ.....	52
3.12 Pseudocode for Kruskal's algorithm.....	53
3.13 Pseudocode for depth-first search.....	54
3.14 ER Diagram.....	55
3.15 ตารางแสดงความสัมพันธ์ของข้อมูล.....	56
บทที่ 4 การใช้งานโปรแกรม.....	60
4.1 หน้าจอหลักของโปรแกรมสำหรับเครื่อง Server.....	60
4.2 การเพิ่ม และแก้ไขข้อมูลท่าเรือ.....	61
4.2.1 การเพิ่ม และแก้ไขข้อมูลท่าเรือสำหรับท่าเรือ Master.....	61
4.2.2 การเพิ่ม และแก้ไขข้อมูลท่าเรือสำหรับท่าเรือ Feeder.....	62
4.3 การเพิ่ม และแก้ไขข้อมูลเรือ.....	63
4.3.1 การเพิ่ม และแก้ไขข้อมูลเรือสำหรับเรือประเภท Master	63
4.3.2 การเพิ่ม และแก้ไขข้อมูลเรือสำหรับเรือประเภท Feeder.....	64
4.4 แสดงเส้นทางเดินเรือ และเลือกเส้นทางเดินเรือสำหรับเรือต่างๆ.....	65
4.5 การเพิ่มข้อมูลเส้นทางเดินเรือใหม่.....	66
4.6 การกำหนดแผนกำหนดการเดินทาง.....	67
4.7 หน้าจอหลักของโปรแกรมสำหรับเครื่อง Client.....	68
4.8 สินค้า (Item).....	69
4.8.1 การเพิ่มข้อมูลสินค้า.....	69
4.8.2 การค้นหาสินค้า.....	71
4.8.3 การค้นหาแผนกำหนดการเดินทาง.....	72
4.9 กำหนดการขนถ่ายสินค้า.....	73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
บทที่ 5 สรุป วิจัยรณ และแนวทงในการแก้ปัญหา.....	74
5.1 ความสามารถของโปรแกรม.....	74
5.2 ข้อจำกัดของโปรแกรม.....	74
5.3 แนวทงในการพัฒนาต่อไป.....	74
ภาคผนวก คู่มือการติดตั้งโปรแกรม.....	75
1. เครื่องคอมพิวเตอร์ที่เหมาะสมสำหรับโปรแกรม.....	75
2. ขั้นตอนการสร้างฐานข้อมูล.....	75
3. การติดตั้งโปรแกรมสำหรับเครื่อง Server.....	76
4. การเพิ่มข้อมูลในโปรแกรมเริ่มต้น.....	79
5. การติดตั้งโปรแกรมสำหรับเครื่อง Client.....	80
บรรณานุกรม.....	83

สารบัญตาราง

ตารางที่	หน้า
1 ตารางการติดต่อกับฐานข้อมูล.....	11
2 ตารางตัวดำเนินการ (Operator).....	25
3 ตารางกลุ่มฟังก์ชัน Aggregate.....	25
4 ตารางแสดงตัวอย่างการจัดเก็บข้อมูลประเภทสตริง.....	39
5 ตารางแสดงตัวอย่างการจัดเก็บข้อมูลประเภทไบนารี.....	39
6 ตารางแสดงประเภทของ Integer.....	40
7 ตารางแสดงความสัมพันธ์ของประเภทข้อมูลแบบ float.....	40
8 ตารางแสดงตารางข้อมูลเรือ.....	56
9 ตารางแสดงข้อมูลท่าเรือ.....	56
10 ตารางแสดงข้อมูลสินค้า.....	57
11 ตารางแสดงข้อมูลเส้นทางเดินเรือ.....	58
12 ตารางแสดงข้อมูลเส้นทางสินค้า.....	58
13 ตารางแสดงข้อมูลสินค้าในท่าเรือ.....	59
14 ตารางแสดงข้อมูลสินค้าในเรือ.....	59
15 ตารางแสดงข้อมูลระยะเวลาระหว่างท่าเรือ.....	59
16 ตารางแสดงข้อมูลตารางเดินเรือ.....	59

สารบัญรูป

รูปที่	หน้า
2.1 ตัวอย่างฐานข้อมูล.....	3
2.2 Control	8
2.3 ตัวอย่างการกำหนดคุณสมบัติ Database Name.....	9
2.4 OLEDB Provider.....	10
2.5 ADO Data Control.....	12
2.6 ชุดออบเจกต์ใน ADO 2.5.....	15
2.7 กลุ่มออบเจกต์ในโมเดล DAO.....	16
2.8 แบบจำลองออบเจกต์ Command.....	18
2.9 ออบเจกต์ Recordset.....	21
2.10 แสดงรูปร่างต้นไม้.....	26
2.11 แสดงตัวอย่างต้นไม้โดยใช้ลิงค์.....	28
2.12 แสดงตัวอย่างต้นไม้โดยใช้ลิงค์.....	28
2.13 แสดงตัวอย่างต้นไม้โดยใช้ลิงค์.....	29
2.14 แสดงค่าที่อยู่ในสแตกเมื่อใช้อัลกอริทึม prestack (T) กับต้นไม้รูป 2.13.....	30
2.15 แสดงค่าที่อยู่ในสแตก ตามอัลกอริทึม instack (T) โดยใช้ต้นไม้รูป 2.13.....	31
2.16 แสดงการสร้างฐานข้อมูลโดยใช้ Enterprise Manager.....	32
2.17 ไดอะแกรมสำหรับการสร้างฐานข้อมูล.....	33
2.18 แสดงการเปลี่ยนตำแหน่งที่เก็บไฟล์ของฐานข้อมูล.....	33
2.19 แสดงการกำหนดชื่อ และตำแหน่งมีลักษณะคล้ายตามการกำหนดฐานข้อมูล.....	34
2.20 แสดงถึงฐานข้อมูลที่สร้างขึ้น.....	35
2.21 แสดงการกำหนดออบชันให้ฐานข้อมูล.....	36
3.1 Functional Decomposition Diagram.....	42
3.2 Context Diagram.....	43
3.3 Data Flow Diagram ระดับ 0.....	44
3.4 Data Flow Diagram ระดับ 1 กระบวนการที่ 1 SET UP.....	45
3.5 Data Flow Diagram ระดับ 1 กระบวนการที่ 2 จัดการข้อมูลสินค้า.....	46
3.6 Data Flow Diagram ระดับ 1 กระบวนการที่ 3 จัดการข้อมูลตารางขนถ่ายสินค้า.....	47
3.7 Data Flow Diagram ระดับ 1 กระบวนการที่ 4 แสดงข้อมูลการขนถ่ายสินค้า.....	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.8 Data Flow Diagram ระดับ 2 กระบวนการที่ 1.1 จัดการข้อมูลแผนการเดินทางเรือ.....	49
3.9 Data Flow Diagram ระดับ 2 กระบวนการที่ 1.2 จัดการข้อมูลเส้นทางเดินทางเรือ.....	50
3.10 Data Flow Diagram ระดับ 2 กระบวนการที่ 1.3 จัดการข้อมูลเรือ.....	51
3.11 Data Flow Diagram ระดับ 2 กระบวนการที่ 1.4 จัดการข้อมูลท่าเรือ.....	52
3.12 Pseudocode for Kruskal's algorithm.....	53
3.13 Pseudocode for depth – first search.....	54
3.14 ER Diagram.....	55
4.1 แสดงหน้าจอหลักของโปรแกรม (สำหรับเครื่อง Server).....	60
4.2.1 แสดงหน้าจอการเพิ่มและแก้ไขข้อมูลท่าเรือสำหรับเรือประเภท Master.....	61
4.2.2 แสดงหน้าจอการเพิ่มและแก้ไขข้อมูลท่าเรือสำหรับเรือประเภท Feeder.....	62
4.3.1 การเพิ่มและแก้ไขข้อมูลเรือสำหรับเรือประเภท Master.....	63
4.3.2 การเพิ่มและแก้ไขข้อมูลเรือสำหรับเรือประเภท Feeder.....	64
4.4 หน้าจอแสดงเส้นทางเดินทางเรือ และเลือกเส้นทางเดินทางเรือสำหรับเรือต่างๆ.....	65
4.5 การเพิ่มข้อมูลเส้นทางเดินทางเรือใหม่.....	66
4.6 การกำหนดแผนกำหนดการเดินทางเรือ.....	67
4.7 แสดงหน้าจอหลักของโปรแกรม (สำหรับเครื่อง Client).....	68
4.8.1 แสดงหน้าจอการเพิ่มข้อมูลสินค้า.....	69
4.8.2 แสดงระยะเวลาโดยประมาณในการขนส่งสินค้า.....	70
4.8.3 แสดงหน้าจอการค้นหาสินค้า.....	71
4.8.4 แสดงหน้าจอการค้นหาแผนกำหนดการเดินทางเรือสำหรับแต่ละท่าเรือ.....	72
4.9 แสดงกำหนดการขนส่งสินค้า.....	73
5.1 Shipping Setup.....	76
5.2 การเลือก Directory ที่ต้องการติดตั้งโปรแกรม.....	76
5.3 การเลือก Program Group.....	77
5.4 หน้าต่าง Setup.....	77
5.5 หน้าต่าง Setup ที่เสร็จสมบูรณ์.....	78
5.6 Client Setup.....	80
5.7 การเลือก Directory ที่ต้องการติดตั้งโปรแกรม.....	80

สารบัญญรูป(ต่อ)

รูปที่	หน้า
5.8 การเลือก Program Group.....	81
5.9 หน้าต่าง Setup.....	81
5.10 หน้าต่าง Setup ที่เสร็จสมบูรณ์.....	82



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันการประกอบธุรกิจไม่ได้จำกัดอยู่แค่ภายในประเทศ มีการติดต่อธุรกิจไปทั่วทุกมุมโลก มีการขนส่งสินค้าระหว่างประเทศทั้งธุรกิจขนาดใหญ่และขนาดเล็ก ซึ่งในการขนส่งสินค้าที่มีน้ำหนักและมีปริมาณมาก การขนส่งทางเรื่อนั้นเป็นทางเลือกที่ดีที่สุด

การขนส่งทางเรือมีการแข่งขันในเรื่องความเร็วและความปลอดภัยในการขนส่ง ในระบบการขนส่งทางเรื่อนั้นเริ่มจากการจัดรถบรรทุกไปรับสินค้ามาบรรจุใส่ตู้ container ตามประเภทของสินค้า เพื่อนำมาจัดลงเรือและขนส่งไปยังท่าเรือปลายทางต่าง ๆ ที่ลูกค้าต้องการ โดยที่แต่ละท่าเรือมีการขนถ่ายสินค้าขึ้น – ลง หรือ ขนถ่ายสินค้าจากเรือลำหนึ่งไปยังเรืออีกลำหนึ่ง เนื่องจากมีสินค้าในแต่ละท่าเรือไม่คงที่แน่นอนและไม่สามารถคาดการณ์ล่วงหน้าได้ ถ้าหากส่งสินค้าไม่ทันตามกำหนดเวลาก็จะต้องเสียค่าชดเชยให้กับลูกค้า ในการเดินเรือแต่ละครั้งนั้นเสียค่าใช้จ่ายสูง เช่น ค่าน้ำมัน ค่าใช้จ่ายในการขนถ่ายสินค้า และมีข้อจำกัดในด้านระวางขนถ่ายของเรือ เส้นทางเดินเรือ จึงจำเป็นต้องจัดการบริหารการขนถ่ายสินค้าให้เหมาะสมและเป็นระเบียบแบบแผนเพื่อลดค่าใช้จ่ายให้มากที่สุด

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1. พัฒนา Software เพื่อใช้ในการจัดการขนถ่ายสินค้า
2. เพื่อให้การขนถ่ายสินค้าเป็นไปอย่างมีระเบียบแบบแผน
3. บริหารการขนถ่ายสินค้าเพื่อให้สินค้าไปถึงที่หมายทันเวลาที่กำหนด
4. เพื่อจัดการตู้ container ลงเรืออย่างมีลำดับก่อนหลังตามระยะทาง

1.3 สมมติฐานของการศึกษา

1. จัดการบริหารสินค้าได้อย่างเหมาะสมและเป็นระเบียบแบบแผน
2. เพื่อให้เสียค่าใช้จ่ายในการขนถ่ายสินค้าน้อยที่สุด
3. ทำให้การจัดการบริหารการขนถ่ายสินค้ามีความน่าเชื่อถือมากขึ้น
4. ลดความผิดพลาดจากการบริหารโดยคน
5. ทราบรายละเอียดของสินค้าที่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการศึกษา

- 1.4.1 ระบบการจัดการฐานข้อมูล
- 1.4.2 รูปแบบการติดต่อกับฐานข้อมูลด้วย Visual Basic 6.0
- 1.4.3 Trees Algorithm
- 1.4.4 ER – Diagram
- 1.4.5 Data Flow Diagram

1.5 ขอบเขตของปัญหา

1. จัดสินค้าในการขนถ่ายลงเรือในแต่ละลำสำหรับแต่ละท่าเรือ
2. มีการกำหนดเส้นทางการเดินทางเรือทั้งหมดไว้อย่างแน่นอน
3. รับข้อมูลเข้าเป็นรายการตู้ container ที่มีสินค้าบรรจุอยู่เรียบร้อยแล้ว
4. จัดลำดับตู้ container ลงเรือโดยเรียงลำดับตามระยะทาง
5. มีการกำหนดตารางเดินเรือล่วงหน้า 1 เดือน
6. ข้อมูลออกของระบบ คือ รายการขนถ่ายสินค้าขึ้น – ลงในแต่ละท่าเรือ
7. การเชื่อมต่อระหว่าง Client และ Server ใช้ระบบ LAN

1.6 ขั้นตอนในการดำเนินงาน

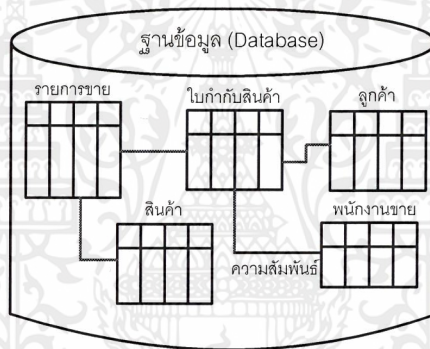
1. ศึกษาระบบงานของการจัดการขนถ่ายสินค้าทางเรือ และลักษณะของข้อมูล
2. เก็บรวบรวมเอกสารและข้อมูลต่าง ๆ
3. วิเคราะห์และออกแบบระบบ
4. พัฒนาระบบงาน
5. ทดสอบโปรแกรมกับข้อมูลจริง
6. แก้ไขข้อผิดพลาดที่เกิดขึ้น
7. ปรับปรุงระบบ

บทที่ 2

ทฤษฎีและหลักเกณฑ์ที่เกี่ยวข้อง

2.1 ความรู้พื้นฐานเกี่ยวกับฐานข้อมูล (Database)

ฐานข้อมูล (Database) คือ ที่อยู่ของข้อมูลที่มีความสัมพันธ์กัน หรืออาจจะเปรียบเทียบเป็นคลังของข้อมูลก็ได้ ข้อมูลเหล่านี้จะถูกจัดเก็บรวมกันอย่างเป็นระบบและมีรูปแบบ ทำให้ง่ายต่อการประมวลผลและจัดการ โดยปกติการใช้งานจะต้องมีโปรแกรมเพื่อจัดการฐานข้อมูลที่มีอยู่ซึ่งเรียกว่า DBMS (Database Management System) สำหรับฐานข้อมูลที่ได้รับนิยามมากที่สุดในปัจจุบันจะเป็นแบบ Relational Database ซึ่งจะจัดเก็บข้อมูลอยู่ในรูปของตาราง (Table) โดยที่ข้อมูลในแต่ละตารางจะมีความสัมพันธ์ซึ่งกันและกัน



รูปที่ 2.1 แสดงตัวอย่างฐานข้อมูล

ตัวอย่างในรูปแสดงข้อมูลการขายสินค้า ซึ่งประกอบไปด้วยตารางใบกำกับสินค้า รายการขาย ลูกค้า สินค้า และพนักงานขาย จะสังเกตได้ว่าในแต่ละตารางจะมีความสัมพันธ์ซึ่งกันและกัน และตารางที่ถูกจัดเก็บในฐาน ข้อมูลจะเป็นข้อมูลที่มีความเกี่ยวพันกันเท่านั้น ข้อมูลใดที่ไม่เกี่ยวข้องมักจะถูกแยกไปในฐานข้อมูลอื่นที่ไม่เกี่ยวข้องกัน

2.2 ความสามารถของ VB6 กับการจัดการฐานข้อมูล

VB6 เป็นคอมพิวเตอร์ที่มีความสามารถและเหมาะสมเป็นอย่างมากในการพัฒนาระบบงานฐานข้อมูล ทั้งแบบที่ใช้งานคนเดียว แบบใช้งานหลายคนพร้อมกัน หรือการสร้างโปรแกรมเป็น FrontEnd ของ Server ซึ่งได้รับความนิยมอย่างกว้างขวางทั้งในและต่างประเทศ ทำให้ปัจจุบันมีระบบงานฐานข้อมูลที่พัฒนาด้วย VB6 เป็นจำนวนมากเนื่องจาก

2.2.1 VB6 สามารถติดต่อและจัดการฐานข้อมูลได้หลากหลายชนิด เช่น Microsoft Access, dBase, Paradox, Foxpro และอื่น ๆ ซึ่ง VB6 มีส่วนของโปรแกรมที่ติดต่อกับฐานข้อมูลได้โดยตรง (Database Engine) ช่วยให้เราสามารถสร้างโปรแกรมติดต่อกับฐานข้อมูลและนำไปติดตั้งได้อย่างเบ็ดเสร็จ โดยเครื่องที่จะติดตั้งโปรแกรมไม่จำเป็นต้องมีระบบจัดการฐานข้อมูล (Database Management System หรือ DBMS) อยู่ก่อนเลย

2.2.2 นอกจากความสามารถในการติดต่อกับฐานข้อมูลที่มีผู้ใช้งานคนเดียว หรือหลายคนพร้อมกันบนเครื่อง PC แล้ว VB6 ยังสามารถติดต่อกับฐานข้อมูลขนาดใหญ่ หรือ Database Server ได้เป็นอย่างดีอีกด้วย

2.2.3 สามารถจัดการฐานข้อมูลได้อย่างง่ายดายเนื่องจาก VB6 มีเครื่องมือที่เรียกว่า Data Control ทำให้ลดเวลาในการเขียนโปรแกรมเพื่อติดต่อและจัดการกับข้อมูลอีกด้วย

2.2.4 VB6 มีเครื่องมือที่เรียกว่า Application Wizard ทำให้เราสามารถสร้างโปรแกรมได้โดยไม่ต้องมีประสบการณ์มาก่อน เพียงตอบคำถามบางอย่างกับวิชาตเท่านั้น เราก็สามารถจะสร้างระบบงานที่ใช้งานได้จริง และใช้เวลาในการเขียนโปรแกรมน้อยมาก

2.2.5 มีเครื่องมือในการสร้างรายงาน กราฟ และการแสดงรูปภาพจากฐานข้อมูลได้โดยตรง

2.2.6 สามารถสร้างระบบงานที่ใช้งานได้จริงเพราะ VB6 มีเครื่องมือในการตรวจสอบความผิดพลาดของข้อมูลนำเข้า (Input) ก่อนการบันทึกเข้าไปในฐานข้อมูล เช่น การใช้งาน MaskedEdit เป็นต้น การยกเลิกการบันทึกที่บันทึกไปแล้ว รวมถึงการป้องกันความผิดพลาดที่อาจเกิดขึ้น ด้วยการใช้คำสั่ง On Error..... ทำให้โปรแกรมที่พัฒนาขึ้นมีความเชื่อถือได้สูง

2.2.7 เราสามารถสร้างระบบงานฐานข้อมูลเพื่อใช้งานบน Internet ได้โดยอาศัย ActiveX คอนโทรล

2.2.8 มีวิชาตเพื่อช่วยในการสร้างแผ่นติดตั้งโปรแกรม (Setup Disk) ทำให้โปรแกรมเมอร์ไม่ต้องยุ่งยากในการเรียนรู้โปรแกรมเพื่อสร้างแผ่นติดตั้งโปรแกรมอื่น ๆ ที่ค่อนข้างยุ่งยากและซับซ้อน เช่น โปรแกรม InstallShield เป็นต้น โดยเราสามารถสร้างแผ่นเพื่อติดตั้งโปรแกรมได้อย่างง่ายดายและรวดเร็ว

2.3 รูปแบบการติดต่อกับฐานข้อมูลด้วย Visual Basic

ในการติดต่อกับฐานข้อมูล โดยปกติแล้ว VB จะเชื่อมโยงผ่านทาง Database Engine ที่เรียกว่า JET Engine จึงอาจกล่าวได้อีกนัยหนึ่งว่า JET Engine คือ ไดรเวอร์ชนิดหนึ่ง ซึ่งทำหน้าที่เป็นตัวเชื่อมโยงให้ VB สามารถติดต่อกับฐานข้อมูลได้นั่นเอง โดยที่ฐานข้อมูลหลัก (Default) ที่ VB รู้จักเป็นอย่างดีก็คือ MS Access แต่ Visual Basic สามารถติดต่อกับฐานข้อมูลได้ทุกชนิดเช่นกัน โดยอาศัยเทคโนโลยีหลาย ๆ อย่าง

สำหรับฐานข้อมูลที่ผู้เขียนจะนำเสนอ จะใช้ฐานข้อมูลของ MS Access 2000 Thai Edition เป็นหลัก ดังนั้นถ้าคุณสามารถใช้งาน MS Access 2000 ได้บ้าง ก็จะทำให้การสร้างแอปพลิเคชันด้านฐานข้อมูลด้วย Visual Basic รวดเร็วขึ้น โดยที่ผู้เขียนจะกล่าวถึงวิธีการใช้งาน MS Access 2000 เท่านั้นที่เกี่ยวข้อง

2.3.1 การติดต่อกับฐานข้อมูลใน Visual Basic จะแยกออกเป็น 4 ประเภท

2.3.1.1 ติดต่อกับคอนโทรลมาตรฐานด้านฐานข้อมูล

มีศัพท์เรียกคอนโทรลกลุ่มนี้โดยเฉพาะว่า Bound Controls ส่วนใหญ่แล้วก็คือกลุ่มของคอนโทรลมาตรฐานที่ใช้กันโดยทั่วไปนั่นเอง เช่น คอนโทรล TextBox , PictureBox , Image , ListBox , ComboBox เป็นต้น โดยใช้คอนโทรล Data (Data Control) เป็นตัวเชื่อมระหว่างฐานข้อมูลกับกลุ่ม Bound Controls คุณสามารถตรวจสอบได้ว่า คอนโทรลตัวใดบ้างถูกจัดอยู่ในกลุ่มของ Bound Controls โดยการตรวจสอบว่าคอนโทรลตัวนั้น มีคุณสมบัติที่ขึ้นต้นด้วยคำว่า Data... เช่น DataField , DataFormat , DataSource หรือไม่ ถ้ามีหมายถึง คอนโทรลตัวดังกล่าวถูกจัดอยู่ในกลุ่ม Bound Controls เช่นกัน

ยังมีคอนโทรลอีกกลุ่มที่เรียกว่า ActiveX Bound Controls หมายถึง กลุ่มของคอนโทรลที่มีคุณสมบัติที่ขึ้นต้นด้วยคำว่า Data ... เช่นกัน แต่มีข้อแตกต่างจากคอนโทรลในกลุ่ม Bound Controls ก็คือ กลุ่ม ActiveX Bound Controls อาจจะมาจากผู้ผลิตรายอื่น ๆ (Thirds Party) ซึ่งเป็นคอนโทรลที่ไม่ได้อยู่ในแถบเครื่องมือมาตรฐานของ Visual Basic โดยที่คุณต้องเพิ่มเติมคอนโทรลกลุ่มนี้เข้ามาในแถบเครื่องมือเอง ดังนั้นจึงเรียกคอนโทรลกลุ่มนี้ว่า ActiveX Bound Controls เช่น คอนโทรล DBGrid , คอนโทรล DbList หรือคอนโทรล DBCombo ฯลฯ เป็นต้น

2. 3.1.2 ติดต่อโดยใช้ออบเจกต์ Data Access Object (DAO)

ถือว่าเป็นวิธีที่ล้าสมัยแล้ว โดยมีแนวคิดในการติดต่อหรือเข้าถึงข้อมูลในฐานข้อมูลผ่านทางองค์ประกอบต่าง ๆ ในฐานข้อมูล เช่น ฟیلด์ (Field) , เร็คคอร์ด (Record) , ความสัมพันธ์ระหว่างตาราง (Relation) เป็นต้น โดยจะแต่ละองค์ประกอบเหล่านั้นด้วยออบเจกต์ (Object) และควบคุมออบเจกต์ต่าง ๆ เหล่านี้โดยการเขียนโค้ด แม้จะทำงานได้ดีกว่า อีสระกว่า แต่มีความยุ่งยากในการเขียนโปรแกรมด้วยเช่นกัน อีกทั้งเป็นเทคโนโลยีที่เก่าแก่มากคือ เน้นเฉพาะระบบฐานข้อมูลที่เป็นตาราง (โดยเฉพาะ Access รุ่นเก่า ๆ) แต่ว่าการเก็บข้อมูลในปัจจุบัน ถูกจัดเก็บอยู่ในสภาพแวดล้อมแตกต่างกันมากมาย เช่น รูปภาพ (Image) , ข้อความ (Text) และรูปแบบอื่น ๆ อีกมากมายทำให้ต้องสร้างออบเจกต์ใหม่ ๆ ขึ้นมาเรื่อย ๆ แต่นั่นไม่ใช่สิ่งที่ทำได้ง่าย และกลายเป็นข้อจำกัดที่สำคัญของ DAO ตัวอย่างของออบเจกต์ในกลุ่มนี้ เช่น ออบเจกต์ RecordSet , ออบเจกต์ TableDef , คอลเล็กชัน Fields เป็นต้น

2.3.1.3 ติดต่อผ่าน ODBC โดยตรง (ODBC Direct)

เป็นการติดต่อกับฐานข้อมูลแบบ 32 บิตที่สนับสนุนมาตรฐาน ODBC (Open DataBase Connectivity) ที่ JET Engine (กลไกการติดต่อฐานข้อมูลของไมโครซอฟท์ซึ่งเน้นที่ Access) ไม่สามารถจัดการได้ เช่น ฐานข้อมูลของ Oracle , ฐานข้อมูลของ Microsoft SQL Server เป็นต้น ซึ่งเป็นการติดต่อเฉพาะฐานข้อมูลที่มีการเก็บข้อมูลในรูปแบบตารางเท่านั้น

ไมโครซอฟท์เองได้สร้างออบเจกต์ขึ้นมาอีกชุดหนึ่งที่ชื่อว่า Remote Data Object หรือเรียกสั้น ๆ ว่า RDO เพื่อใช้สำหรับติดต่อกับฐานข้อมูลแบบเครือข่าย (เช่น Oracle , SQL Server , DB2 , ฯลฯ) ซึ่งเป็นไปตามสถาปัตยกรรม n – Tier (n เทียร์) เช่น Client / Server (2-Tier ทุเทียร์) หรือ Middle Teir (3-Tier ตรีเทียร์) โดยอาศัยมาตรฐาน ODBC ในการเชื่อมโยงที่เกี่ยวข้องกับ Visual Basic คือ

- คอนโทรล Remote Data (Remote Data Control – RDC)
- ออบเจกต์ Remote Data (Remote Data Object – RDO)

2.3.1.4 เข้าถึงข้อมูลโดยอาศัยเทคโนโลยี OLEDB

เห็นรูปแบบการติดต่อกับฐานข้อมูลผ่านทางกลุ่มออบเจกต์ในโมเดล ADO ซึ่งใช้ OLEDB Provider เป็นกลไกในการเข้าถึงข้อมูลในฐานข้อมูลแทน JET Engine โดย

เปลี่ยนจากมุมมองการติดต่อฐานข้อมูลแบบเดิม ๆ ที่ต้องกำหนดชนิดของฐานข้อมูลมาเป็นมองที่รูปแบบของการเชื่อมต่อ (Connection) เข้ากับฐานข้อมูล OLEDB เป็นเทคโนโลยีล่าสุดที่ไม่โครซอฟท์วางตำแหน่งให้ยุคต่อไป ในการเข้าถึงข้อมูลในฐานข้อมูลที่มีรูปแบบการเก็บข้อมูลสารพัดรูปแบบ ซึ่งไม่ได้จำกัดตัวเองเฉพาะตาราง โดย OLEDB เป็นส่วนหนึ่งของสถาปัตยกรรม Universal Data Access (UDA) ของไมโครซอฟท์

ใน Visual Basic สามารถใช้ OLEDB ได้ 2 แบบคือ โดยการให้

- คอนโทรล ADO Data (ADO Data Control) ร่วมกับกลุ่มของ ActiveX Bound Controls ซึ่งสังเกตจากชื่อของคอนโทรลที่มีคำว่า OLEDB ต่อท้าย เช่น คอนโทรล DataGrid , DataList , Hierarchical FlexGrid เป็นต้น
- ชุดออบเจกต์ ActiveX Data (ActiveX Data Object – ADO) โดยการเขียนโปรแกรมควบคุม

Relational Database

Relational Database หมายถึง กลุ่มข้อมูลที่จัดเก็บอยู่ในฐานข้อมูล โดยอาศัยรูปแบบของตาราง (Table) เป็นตัวสร้างความสัมพันธ์ระหว่างข้อมูล โดยที่ถ้ามองข้อมูลในลักษณะแนวนอน (Row) จะถูกเรียกว่า เร็คคอร์ด (Record) แต่ถ้ามองในลักษณะ แนวตั้ง (Column) จะถูกเรียกว่าฟิลด์ (Field) หรือแอตทริบิวต์ (Attribute)

ตารางดังกล่าว จะมีความสัมพันธ์กัน (Relation) หรือเกี่ยวข้องกันในลักษณะที่ข้อมูลที่อยู่ในเร็คคอร์ดของตารางหนึ่งมีความสัมพันธ์ หรือเชื่อมโยงกับเร็คคอร์ดของอีกตารางหนึ่ง ส่งผลให้ตารางทั้ง 2 ตาราง มีความสัมพันธ์ และอยู่ร่วมกันเป็นฐานข้อมูลเชิงสัมพันธ์ (Relational Database) โดยมี RDBMS ทำหน้าที่จัดการข้อมูลที่อยู่ในฐานข้อมูลให้มีประสิทธิภาพ

ชนิดของความสัมพันธ์ระหว่างตาราง

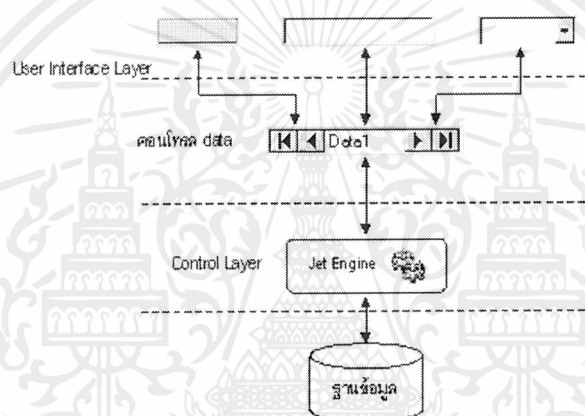
ความสัมพันธ์ของข้อมูลระหว่างตาราง สามารถแยกออกได้ 3 ชนิด คือ

1. One-to-One หมายถึงข้อมูล 1 เร็คคอร์ดที่อยู่ในตารางหนึ่ง มีความสัมพันธ์หรือเชื่อมโยงกับอีกตารางหนึ่ง โดยที่สามารถอ้างอิงได้เพียงหนึ่งเร็คคอร์ดเท่านั้น เช่น นักศึกษา 1 คน มีรหัสประจำตัวได้ 1 หมายเลข ในทางกลับกันรหัสนักศึกษา 1 หมายเลข สามารถอ้างอิงถึงนักศึกษาได้เพียง 1 คนเท่านั้น

2. One-to-Many หมายถึงข้อมูล 1 เร็คคอร์ดที่อยู่ในตารางหนึ่ง มีความสัมพันธ์หรือเชื่อมโยงกับอีกตารางหนึ่ง โดยที่สามารถอ้างอิงได้มากกว่า 1 เร็คคอร์ด เช่น มหาวิทยาลัยมีนักศึกษาได้หลายคน ในทางกลับกันมีนักศึกษาหลายคนศึกษาอยู่ในมหาวิทยาลัยเดียวกัน
3. Many-to-Many หมายถึง ข้อมูลหลายเร็คคอร์ดที่อยู่ในตารางหนึ่ง มีความสัมพันธ์หรือเชื่อมโยงกับอีกตารางหนึ่ง โดยที่สามารถอ้างอิงได้มากกว่า 1 เร็คคอร์ด เช่น นักศึกษา 1 คน ลงทะเบียนได้หลายวิชา และแต่ละวิชามีนักศึกษาลงทะเบียนได้หลายคนเช่นกัน

2.3.2 การใช้งานคอนโทรล ADO Data ร่วมกับคอนโทรล TextBox

คอนโทรล Data (Data Control) เป็นคอนโทรลที่ถือได้ว่าเป็นตัวกลางในการเชื่อมต่อระหว่างกลุ่ม Bound Controls กับฐานข้อมูล

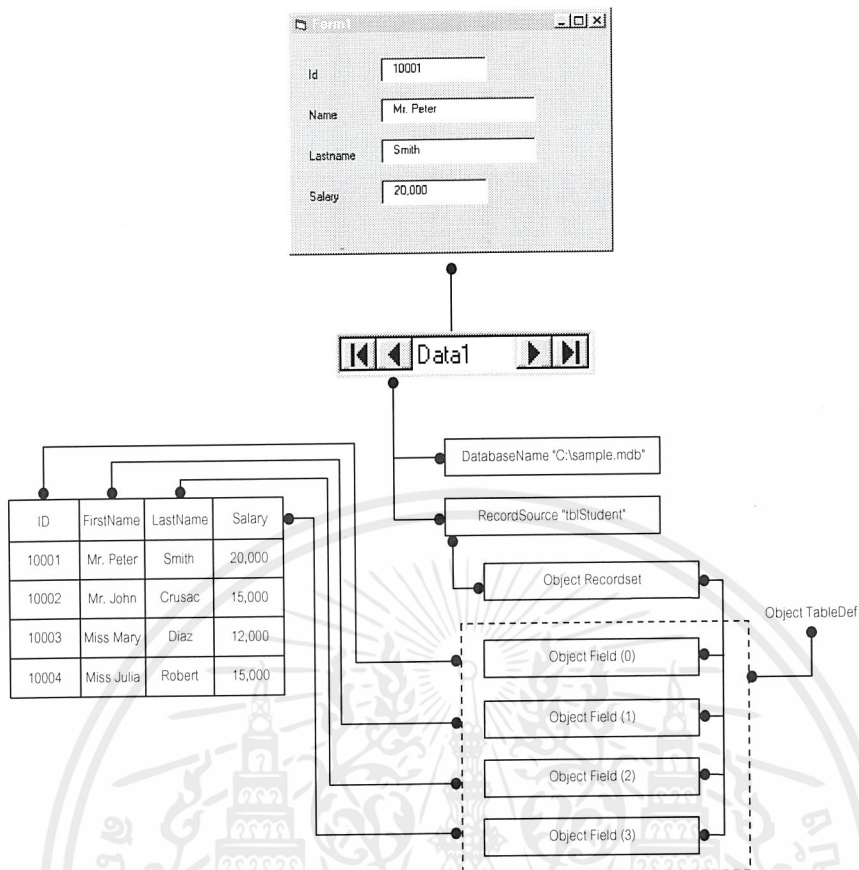


รูปที่ 2.2 Control Data

คอนโทรล Data จะอาศัยเทคโนโลยี Data Access Object (DAO) ในการเข้าถึงข้อมูลในฐานข้อมูล โดยที่ต้องกำหนดชนิดของฐานข้อมูล , ชื่อฐานข้อมูลที่ต้องการติดต่อ ฯลฯ เพื่อกำหนดให้คอนโทรล Data ไปสั่งให้ JET Engine ทำงาน

แต่เมื่อข้อมูลมีความหลากหลายมากขึ้น ไม่ว่าจะเป็นชนิดข้อมูลที่ต้องการเก็บ , รูปแบบข้อมูล , ข้อมูลถูกจัดเก็บใน RDBMS ที่แตกต่างกัน จากผู้ผลิตหลายรายทำให้เป็นข้อจำกัดของ JET Engine ไปโดยปริยายที่ไม่รู้จักโครงสร้างของฐานข้อมูลชนิดนั้น เนื่องจากออบเจกต์ต่าง ๆ ในโมเดลของ DAO ไม่ได้ถูกออกแบบมาให้รู้จักฐานข้อมูลทุกประเภท ส่งผลให้ JET Engine ไม่สามารถเข้าไปจัดการฐานข้อมูลประเภทอื่น ๆ ได้นั่นเอง แม้ว่าไมโครซอฟท์จะใช้มาตรฐาน ODBC มาเป็นเทคโนโลยีช่วยให้ JET Engine รู้จักฐานข้อมูลประเภทอื่นมากขึ้น แต่ก็คงไม่ใช่เป็นการแก้ปัญหาที่ดีนัก เพราะว่ามีมาตรฐาน ODBC ก็มีข้อจำกัดตัวมันเองเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 ตัวอย่างการกำหนดคุณสมบัติ Database Name

จากรูปที่ 2.3 จะเห็นว่า เมื่อกำหนดคุณสมบัติ DatabaseName และคุณสมบัติ RecordSource ให้กับคอนโทรล Data แล้ว กลไกที่ซ่อนอยู่ภายในคอนโทรล Data จะอาศัย

ออบเจกต์ต่าง ๆ ของโมเดล DAO เข้าไปจัดการข้อมูลในฐานข้อมูลนั่นเอง

ผลที่ได้จากฐานข้อมูลสิ่งแรกก็คือ

- ออบเจกต์ Recordset ซึ่งเก็บเรคคอร์ดของตารางที่คุณใช้งาน
- ออบเจกต์ TableDef ใช้แทนตารางในฐานข้อมูล
- ออบเจกต์ Field ใช้แทนฟิลด์ต่างๆ ที่อยู่ในตาราง
- ออบเจกต์อื่น ๆ อีกมากมาย

จากขั้นตอนดังกล่าว ทำให้ทุกๆ ครั้งที่คุณใช้งานคอนโทรล Data คุณจึงต้องทำผ่านทางออบเจกต์ Recordset เท่านั้น ดังนั้นถ้าจะเรียกคอนโทรล Data เป็น DAO Data Control ก็ไม่ผิดนัก

จากเหตุผลข้างต้น ใคร่ขอพที่ไม่ต้องการแก้ปัญหาดังกล่าว ด้วยการเพิ่มเติมออบเจกต์ให้กับโมเดล DAO มากขึ้น เพื่อให้ DAO รู้จักกับฐานข้อมูลประเภทอื่น ๆ ซึ่งจะกลายเป็นการ

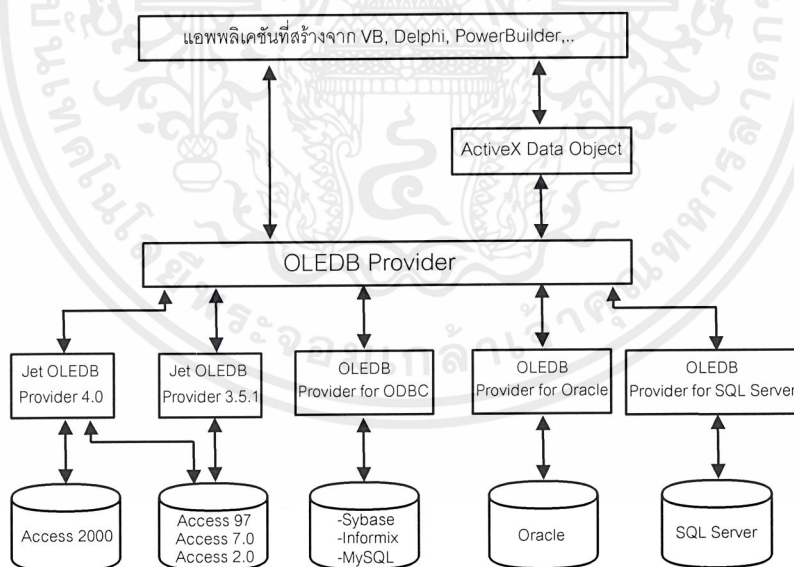
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แก้ปัญหาไม่รู้จกจบสิ้น ไมโครซอฟท์เห็นว่าเทคโนโลยี DAO มาถึงขอบเขตของมันแล้ว แม้ว่าในปัจจุบันไมโครซอฟท์จะออก DAO เวอร์ชัน 3.6 มาพร้อมกับ Service Pack 3 ของ Visual Basic แต่เป็นการออกมาเพื่อให้โปรแกรมเมอร์ที่ต้องการใช้งาน DAO ร่วมกับ Access 2000 เท่านั้น

สาเหตุหลักที่ DAO ถึงขอบเขตของมันก็คือ ความหลากหลายของข้อมูลที่เกิดขึ้นในปัจจุบันนั่นเอง ทำให้ไมโครซอฟท์ต้องการสร้างเทคโนโลยีในการเข้าถึงข้อมูลในฐานข้อมูลในรูปแบบใหม่ขึ้นมา และจะต้องทำลายข้อจำกัดดังกล่าวให้ได้ รวมถึงจะต้องสามารถเข้าถึงข้อมูลในฐานข้อมูลผ่านทางอินเทอร์เน็ตที่กลายเป็นเครือข่าย Network ที่ใหญ่ที่สุดได้ด้วย

สถาปัตยกรรม Universal Data Access เรียกว่า UDA เป็นเทคโนโลยีที่ไมโครซอฟท์สร้างขึ้นมาจากด้วยเหตุผลข้างต้น เป็นเทคโนโลยีที่สร้างขึ้นมาจากใช้ OLEDB Provider ทำหน้าที่เป็น Engine ในการเข้าถึงข้อมูลในฐานข้อมูลแทน JET Engine ทำให้ไม่เกิดข้อจำกัดเหมือนอย่างเช่นที่เคยเกิดขึ้นกับ JET Engine มาแล้ว

UDA เป็นเทคโนโลยีที่ถูกสร้างขึ้นใหม่โดยไม่ได้ใช้สถาปัตยกรรมเก่าใด ๆ เป็นแกน ทำให้ไมโครซอฟท์ต้องสร้าง OLEDB Provider ให้กับเทคโนโลยีเก่าที่ออกมาก่อนหน้านี้ทั้งหมด เพื่อให้ UDB สามารถใช้ได้กับฐานข้อมูลได้ทุกชนิดที่มีอยู่ในปัจจุบัน



รูปที่ 2.4 OLEDB Provider

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1 ตารางการติดต่อกับฐานข้อมูล

ODBC Provider	การใช้งาน
Microsoft Jet 4.0 OLE DB Provider	ใช้สำหรับ JET Engine 4.0 ซึ่งจะใช้กับ Access 2000 และ Access 97
Microsoft Jet 3.51 OLE DB Provider	ใช้สำหรับ JET Engine 3.51 ซึ่งใช้กับ Access 97 และเวอร์ชันก่อนหน้า
Microsoft OLE DB Provider for ODBC Drivers	ใช้กับฐานข้อมูลที่ติดต่อโดยอาศัยมาตรฐาน ODBC ในการติดต่อ
Microsoft OLE DB Provider for Oracle	ใช้กับฐานข้อมูล Oracle
Microsoft OLE DB Provider for SQL Server	ใช้กับฐานข้อมูลชนิด MS SQL Server
Microsoft OLE DB Simple Provider	ใช้กับฐานข้อมูลชนิดอื่น ๆ

จากรูป 2.4 จะสังเกตได้ว่า ในฐานข้อมูลของ MS SQL Server และ Oracle สามารถติดต่อได้ 2 แบบคือ ผ่านทางมาตรฐาน ODBC และผ่านทาง OLE DB Provider เฉพาะตัว เนื่องจากว่า ทั้ง MS SQL Server และ Oracle เป็น RDBMS ที่อยู่ในระดับที่เรียกว่า Database Server ทำให้ไม่ใคร่ซอฟต์แวร์ต้องออก OLE DB Provider แยกออกมาต่างหาก เพื่อประสิทธิภาพสูงสุดในการทำงาน

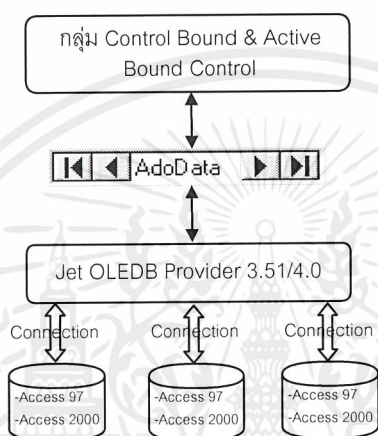
ใน Visual Basic เราสามารถเรียกใช้เทคโนโลยี ADO ได้ 2 แบบคือ

1. การใช้งานกลุ่ม ActiveX Bound Controls และกลุ่ม Bound Controls ที่มีคำว่า OLEDB ต่อท้าย
2. การเรียกใช้งานกลุ่มออบเจกต์ในโมเดล ActiveX Data Object (ADO) โดยตรง โมเดล ADO ถูกออกแบบมาให้สามารถเข้าถึงข้อมูลในฐานข้อมูลผ่านทางอินเทอร์เน็ตได้ อีกด้วย โดยอาศัยภาษาสคริปต์ Active Server Pages (ASP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ADO Data Control

คอนโทรล ADO Data เป็นคอนโทรลที่ถูกออกแบบมาแทนคอนโทรล Data โดยที่คอนโทรล ADO Data จะใช้เทคโนโลยี OLEDB เป็นกลไกการเข้าถึงข้อมูลในฐานข้อมูล รูปแบบการติดต่อของคอนโทรล ADO Data กับฐานข้อมูล จะเป็นลักษณะการเชื่อมต่อ (Connection) เข้ากับฐานข้อมูลแทนการกำหนดชื่อฐานข้อมูล คำว่า connection คุณจะได้พบเจอบ่อยครั้งมากเมื่อใช้ ADO



รูปที่ 2.5 ADO Data Control

จากรูป 2.5 Microsoft Jet 3.51 OLEDB Provider จะใช้กับ Access 97 และเวอร์ชันก่อนหน้านี ส่วน Microsoft Jet 4.0 OLEDB Provider จะใช้กับ Access 97 และ Access 2000 วิธีการจัดการข้อมูลในฐานข้อมูลของคอนโทรล ADO Data จะทำผ่านทางออบเจกต์ Recordset เช่นเดียวกับคอนโทรล Data ซึ่งเป็นออบเจกต์เพียงตัวเดียวที่เหลืออยู่ ที่นำมาใช้กับ ADO และเป็นออบเจกต์ที่มีความสำคัญเป็นอย่างยิ่งของโมเดล ADO อีกด้วย

2.4.1 พื้นฐานการเขียนโปรแกรมด้านฐานข้อมูลด้วยเทคโนโลยี ActiveX Data Object

เทคโนโลยี Data Access Object หรือเรียกว่า DAO ใช้หลักการในการนิยามองค์ประกอบต่าง ๆ ที่อยู่ในฐานข้อมูลออบเจกต์ทั้งหมด จะประกอบไปด้วยกลุ่มของออบเจกต์ที่เป็นตัวแทนส่วนต่าง ๆ ที่อยู่ในฐานข้อมูล เช่น ฟิลด์ต่าง ๆ ที่อยู่ในตาราง จะถูกนิยามเป็นออบเจกต์ตัวหนึ่ง หรือแม้ว่าตัวตารางเองก็จะมีออบเจกต์ที่เป็นตัวแทนของตารางด้วยเช่นกัน และ error ที่เกิดขึ้น ก็คือออบเจกต์ตัวหนึ่งเช่นกัน

จะเห็นได้ว่าองค์ประกอบต่าง ๆ ในฐานข้อมูล จะถูก DAO มองเป็นออบเจกต์ทั้งหมด ด้วยเหตุนี้เองจึงสามารถจัดการส่วนปลีกย่อยต่าง ๆ ในฐานข้อมูลได้เป็นอย่างดี แต่ก็มีความยุ่งยากในการเขียนโค้ดเช่นกัน

องค์ประกอบของ DAO

ลักษณะของออบเจกต์ ในโมเดลของ DAO เบื้องต้นมีดังนี้

1. ออบเจกต์ DBEngine (DBEngine Object) เป็นออบเจกต์ระดับบนสุด เป็นตัวแทนของ JET Engine ที่ใช้สำหรับควบคุม และ / หรือจัดการโครงสร้างของฐานข้อมูลที่เราสร้างขึ้นมา ซึ่งจะมีได้เพียงตัวเดียวเท่านั้นใน 1 โปรเจกต์
2. ออบเจกต์ Workspace (Workspace Object) เป็นออบเจกต์ที่ใช้สำหรับกำหนดพื้นที่สมมติขึ้นมาเพื่อเก็บฐานข้อมูลที่เราสร้างขึ้นมา
3. ออบเจกต์ Database (Database Object) เป็นออบเจกต์ที่เป็นตัวแทนของฐานข้อมูลนั้น ๆ ซึ่งจะประกอบไปด้วยออบเจกต์ที่เป็นองค์ประกอบของฐานข้อมูล จะแบ่งออกเป็น 2 กลุ่มใหญ่ คือ

กลุ่มที่ 1 กลุ่มของออบเจกต์ที่ใช้สำหรับกำหนดลักษณะฐานข้อมูลมีอยู่ 3 ตัว คือ

1. ออบเจกต์ TableDef (TableDef Object) เป็นออบเจกต์ที่ใช้สำหรับนิยามหรือแทนตารางที่อยู่ในฐานข้อมูลใด ๆ
2. ออบเจกต์ Field (Field Object) เป็นออบเจกต์ที่ใช้แทนฟิลด์ที่อยู่ในตารางใด ๆ
3. ออบเจกต์ Index (Index Object) เป็นออบเจกต์ที่ใช้แทนค่าดัชนีของตารางนั้น ๆ

กลุ่มที่ 2 กลุ่มของออบเจกต์ที่ใช้สำหรับกำหนดรายละเอียดอื่น ๆ ของฐานข้อมูลที่สำคัญมีอยู่ 2 ตัว

1. ออบเจกต์ Relation (Relation Object) ใช้กำหนดหรือแทนความสัมพันธ์ระหว่างตาราง ในฐานข้อมูลใด ๆ
2. ออบเจกต์ QueryDef (QueryDef) ใช้สำหรับสร้างชุดสตริงที่เป็นคำสั่ง SQL ที่ใช้ในฐานข้อมูลนั้น

ข้อเสียของ DAO

จะเห็นได้ว่า ถ้าต้องการจัดการฐานข้อมูล อาจจะต้องสร้างออบเจกต์ทุกตัวที่กล่าวมา โดยที่ออบเจกต์บางตัวไม่ได้ใช้งานแต่ก็ต้องสร้างขึ้นมา เพราะว่าในโมเดลของ DAO มีข้อบังคับอยู่อย่างหนึ่งก็คือ เมื่อต้องการใช้งานออบเจกต์ในระดับล่าง จะต้องสร้างออบเจกต์ในระดับบนขึ้นมาก่อน ซึ่งการกำหนดไว้เช่นนี้เป็นกลไกการเปลี่ยนแปลงหน่วยความจำเป็นอย่างยิ่ง เช่น ต้องการใช้งานออบเจกต์ Field ต้องสร้างออบเจกต์ DBEngine ตามด้วยออบเจกต์ Workspace แล้วจึงออบเจกต์ Database เป็นต้น

จากหลักการทำงาน DAO จะเห็นได้ว่า โครงสร้างของโมเดล DAO ค่อนข้างซับซ้อนและไม่ยืดหยุ่นเท่าที่ควร ที่กล่าวมาเป็นโมเดลของ DAO เฉพาะที่เกี่ยวข้องกับ JET Engine เท่านั้น ยังไม่รวมถึง DAO ที่เกี่ยวข้องกับ RDBMS ตัวอื่น ๆ อีก ด้วยเหตุนี้จึงทำให้เทคโนโลยีนี้ไม่ได้รับการพัฒนาต่อ

2.4.2 เทคโนโลยี ActiveX Data Object

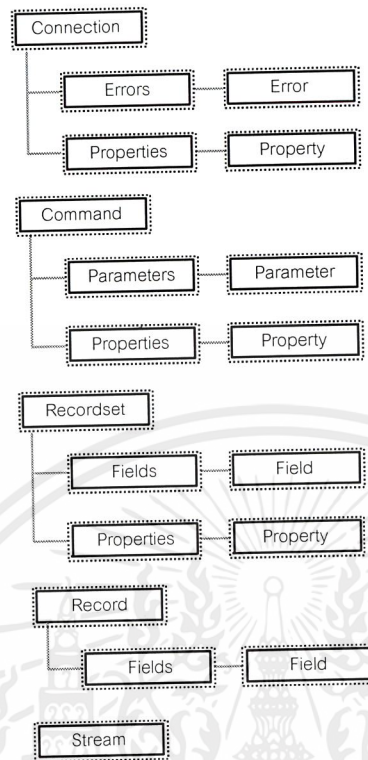
ADO เป็นเทคโนโลยีที่มีแนวคิดมาจาก DAO บางส่วน กล่าวคือจะมองฐานข้อมูลเป็นออบเจกต์เช่นกันแต่จะใช้ OLEDB Provider เป็นตัวจัดการ ข้อมูลและโครงสร้างของฐานข้อมูลแทน โดยที่จะไม่นิยามออบเจกต์ขึ้นมาเพื่อแทนโครงสร้างของฐานข้อมูลแต่ละส่วนเหมือนกับ DAO แต่จะใช้วิธีการสร้าง OLEDB Provider ให้กับ RDBMS แต่ละชนิดแทน เช่น เมื่อฐานข้อมูลเป็นชนิด JET ก็จะใช้ Microsoft Jet OLEDB Provider ถ้าเป็น Oracle ก็จะใช้ Microsoft OLEDB Provider for Oracle เป็นต้น ทำให้ออบเจกต์ในโมเดล ADO ไม่ซับซ้อนและยุ่งยากเท่ากับ DAO

จากหลักการทำงานดังกล่าว จึงทำให้ OLEDB Provider เป็นกลไกสำคัญที่สุดที่ทำให้ติดต่อกับฐานข้อมูลได้ทุกชนิด ไม่โครซอฟท์ได้ออกแบบ OLEDB Provider ไว้ให้กับโปรแกรมเมอร์ที่มาจากเกือบทุกเทคโนโลยีที่เกี่ยวข้องด้านฐานข้อมูล

ใน Visual Basic สามารถเรียกใช้ OLEDB Provider ได้ 2 วิธีคือ

1. อาศัยคอนโทรล ADO Data (ADO Data Control) ร่วมกับกลุ่มคอนโทรล Bound & ActiveX Bound Controls ที่มีคำว่า OLEDB ต่อท้าย
2. โดยการเรียกใช้วามกลุ่มออบเจกต์ ADO โดยตรง

2.4.3 ชุดออบเจ็กต์ใน ADO 2.5



รูปที่ 2.6 ชุดออบเจ็กต์ใน ADO 2.5

สำหรับในโมเดลของ ADO 2.5 จะประกอบไปด้วยออบเจ็กต์ 9 ตัวคือ

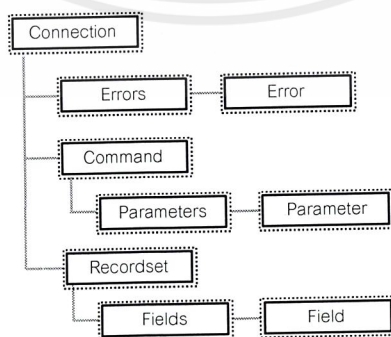
- ออบเจ็กต์ Command เป็นออบเจ็กต์ตัวกลางที่ใช้สำหรับส่งคำสั่ง (Execute) ไปยังฐานข้อมูลเป็นเป้าหมายที่ต้องการติดต่อ มีศัพท์เรียกฐานข้อมูลดังกล่าวว่า Data Sources
- ออบเจ็กต์ Connection ใช้สำหรับเริ่มต้นเปิดการเชื่อมต่อกับฐานข้อมูล
- ออบเจ็กต์ Error เป็นออบเจ็กต์ที่เก็บข้อมูลเกี่ยวกับความผิดพลาดของการเข้าถึงลงข้อมูลในฐานข้อมูลของ OLEDB Provider
- ออบเจ็กต์ Field ใช้สำหรับเก็บฟิลด์ในฟิลด์ใดฟิลด์หนึ่งที่อยู่ในฐานข้อมูล
- ออบเจ็กต์ Parameter ใช้สำหรับรับ - ส่งตัวแปร หรืออาร์กิวเมนต์ระหว่างการทำคิวรีหรือ Stored Procedure ของออบเจ็กต์ Command
- ออบเจ็กต์ Property ใช้สำหรับกำหนดคุณสมบัติต่าง ๆ ของออบเจ็กต์ ADO
- ออบเจ็กต์ Record ใช้สำหรับเก็บเรคคอร์ดใดเรคคอร์ดหนึ่ง ที่อยู่ในออบเจ็กต์ Recordset

Recordset

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ออบเจกต์ Recordset เป็นผลลัพธ์ที่ได้มาจากฐานข้อมูล ประกอบไปด้วยกลุ่มของเรคคอร์ดและฟิลด์ที่มาจากตารางที่เชื่อมต่อ เป็นออบเจกต์ที่มีความสำคัญมากที่สุด และมีรายละเอียดมากที่สุดเช่นกัน
 - ออบเจกต์ Stream เป็นออบเจกต์ที่ทำหน้าที่เป็นตัวกลาง สำหรับรับ - ส่งข้อมูลชนิดข้อความ (Stream of Text) ให้กับระบบไฟล์ข้อความ File System Object (FSO) เช่น การสร้างไฟล์ *.txt ด้วย ASP จะอาศัยออบเจกต์ตัวนี้ ทำหน้าที่ลำเลียงข้อมูลนั่นเอง
- สำหรับออบเจกต์ Stream นั้นยังประกอบไปด้วยคอลเล็คชัน 4 ตัวคือ
- คอลเล็คชัน Error เป็นที่เก็บออบเจกต์ Error ซึ่งเกิดขึ้นระหว่างการเข้าถึงข้อมูลในฐานข้อมูลของ OLEDB Provider
 - คอลเล็คชัน Fields ใช้สำหรับเก็บออบเจกต์ Field ที่มาจากตาราง แต่เนื่องจากว่าผลลัพธ์ของการติดต่อกับฐานข้อมูล จะได้กลับมาในรูปแบบของออบเจกต์ Recordset ดังนั้นแท้ที่จริงแล้วคอลเล็คชัน Fields ก็คือรายชื่อฟิลด์ที่อยู่ในออบเจกต์ Recordset นั่นเอง
 - คอลเล็คชัน Parameter ใช้สำหรับเก็บออบเจกต์ Parameter ที่เกิดขึ้น
 - คอลเล็คชัน Properties ใช้สำหรับเก็บออบเจกต์ Property ที่เกิดขึ้น

จากออบเจกต์ทั้ง 9 ตัวของ ADO 2.5 จะมีออบเจกต์ที่เป็นแกนหลักอยู่ 3 ตัวคือ ออบเจกต์ Connection , Command , Recordset ลักษณะที่สำคัญที่สุดของออบเจกต์ทั้ง 3 ตัวคือ ออบเจกต์แต่ละตัวจะมีอิสระต่อกัน คุณไม่จำเป็นต้องสร้างออบเจกต์ตัวใดตัวหนึ่งขึ้นมา ก่อน แล้วจึงสร้างออบเจกต์อีกตัวหนึ่งให้เหมือนกับกลุ่มออบเจกต์ในโมเดล DAO ดังรูป 2.7



รูปที่ 2.7 กลุ่มออบเจกต์ในโมเดล DAO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 การใช้งานออบเจกต์ Connection

ออบเจกต์ Connection มีหน้าที่สำหรับสร้างการเชื่อมต่อกับฐานข้อมูล การใช้ออบเจกต์ Connection เพื่อเชื่อมต่อกับฐานข้อมูลสามารถทำได้ 2 วิธีคือ

วิธีที่ 1 - โดยใช้เมธอด Open

วิธีที่ 2 - โดยใช้คุณสมบัติ ConnectionString

ในที่นี้จะกล่าวในกรณีที่ใช้คุณสมบัติ ConnectionString

2.5.1 วิธีการเชื่อมต่อกับฐานข้อมูลโดยใช้คุณสมบัติ ConnectionString

สำหรับการเชื่อมต่อโดยใช้ ConnectionString มีขั้นตอนดังนี้

1. กำหนดคุณสมบัติ ConnectionString เพื่อระบุชนิดของ OLEDB Provider และ พารามิเตอร์ของฐานข้อมูลที่ต้องการติดต่อ

ตัวอย่าง

```
Dim rs As ADODB.Connection { สร้างตัวแปรชนิด ADODB.Connection }
```

```
Set Conn = New ADODB.Connection { กำหนดให้สร้างการเชื่อมต่อ }
```

2. กำหนดเมธอด Open เพื่อเริ่มต้นเปิดฐานข้อมูล

ตัวอย่าง

```
With Conn
```

```
.ConnectionString = StudentPath { ติดต่อกับฐานข้อมูลตามพารามิเตอร์
```

```
StudentPath เก็บไว้ }
```

```
.Open { เปิดฐานข้อมูลด้วยเมธอด Open }
```

```
End With
```

3. กำหนดเมธอด Close เพื่อปิดการเชื่อมต่อกับฐานข้อมูล

ตัวอย่าง

```
Conn.Close { จบการเชื่อมโยง }
```

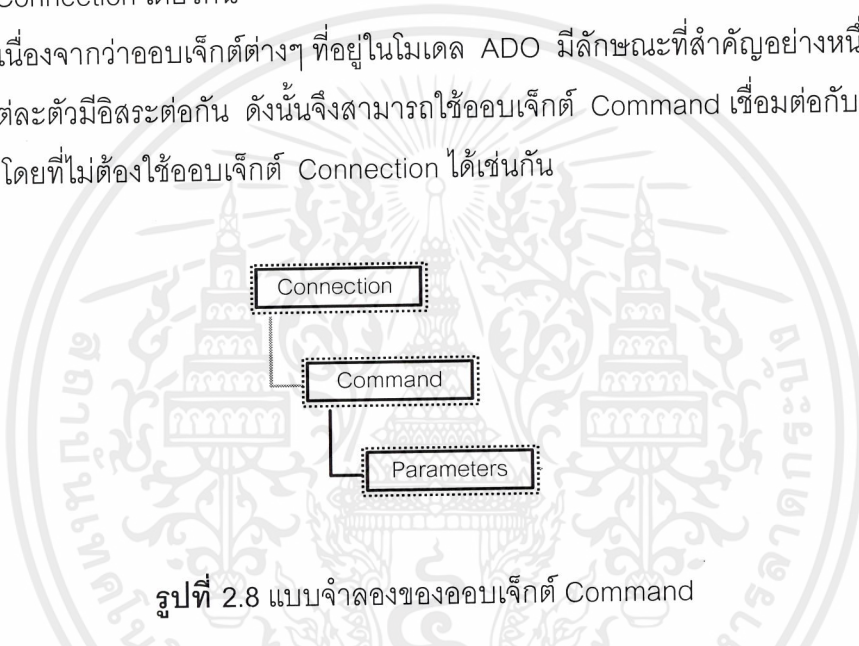
```
Set Conn = Nothing { คืนหน่วยความจำของตัวแปร Conn }
```

2.6 ออบเจกต์ Command

เมื่อเราต้องการเข้าถึงข้อมูลในฐานข้อมูล จะต้องสร้างการเชื่อมต่อกับออบเจกต์ Connection หลังจากที่เราเชื่อมต่อกับฐานข้อมูลแล้ว เราจะใช้ออบเจกต์ Command ในการเข้าถึงข้อมูลในฐานข้อมูล

อาจกล่าวได้ว่า ออบเจกต์ Command เป็นตัวกลางที่ทำหน้าที่รับคำสั่ง เช่น ชุดคำสั่ง SQL จากออบเจกต์ Connection หรือจากการใช้ method execute ของตัวมันเองเข้าไปยังฐานข้อมูล เพื่อที่จะได้ข้อมูลกลับมาในรูปแบบของออบเจกต์ Recordset ภายในการเชื่อมต่อของออบเจกต์ Connection เดียวกัน

แต่เนื่องจากว่าออบเจกต์ต่างๆ ที่อยู่ในโมเดล ADO มีลักษณะที่สำคัญอย่างหนึ่งก็คือ ออบเจกต์แต่ละตัวมีอิสระต่อกัน ดังนั้นจึงสามารถใช้ออบเจกต์ Command เชื่อมต่อกับฐานข้อมูล โดยที่ไม่ต้องใช้ออบเจกต์ Connection ได้เช่นกัน



รูปที่ 2.8 แบบจำลองของออบเจกต์ Command

2.6.1 วิธีการเชื่อมต่อกับฐานข้อมูลด้วยออบเจกต์ Command

เราสามารถใช้ออบเจกต์ Command เชื่อมต่อกับฐานข้อมูล และส่งคำสั่งไปยังฐานข้อมูลได้เช่นกัน โดยที่การเชื่อมต่อเข้ากับฐานข้อมูล อาจใช้การเชื่อมต่อเดิมที่มาจากออบเจกต์ Connection หรือสร้างการเชื่อมต่อใหม่ด้วยออบเจกต์ Command เองก็ได้ โดยการกำหนดข้อความเชื่อมต่อให้กับคุณสมบัติ ActiveConnection ก็ได้ ดังตัวอย่าง Code Project ที่ 1

```
Private Sub Form_Load()
```

```
Dim Cmd As ADODB.Command
```

```
Set Cmd = New ADODB.Command
```

```
With Cmd
```

```
.CommandText = "Select * From student"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

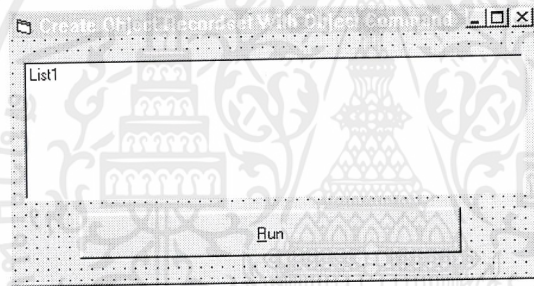
```
.CommandType = adCmdText
.ActiveConnection = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=D:\student.mdb;Persist Security Info=False"
End With
End Sub
```

Code Project ที่ 1 การเชื่อมต่อเข้ากับฐานข้อมูลด้วยออบเจกต์ Command

2.6.2 วิธีการสร้างออบเจกต์ Recordset ด้วยออบเจกต์ Command

เป็นวิธีการส่งชุดคำสั่ง SQL โดยการใช้เมธอด execute ของออบเจกต์ Command ไปยังตารางที่ต้องการเพื่อที่จะให้ได้ออบเจกต์ Recordset กลับมาก่อน แล้วแสดงข้อมูลที่อยู่ในออบเจกต์ Recordset ดังกล่าวออกมา ดังตัวอย่างใน Code Project ที่ 3

1. ออกแบบฟอร์มดังรูป



2. ให้แก้ไขคุณสมบัติคอนโทรลต่างๆ ดังต่อไปนี้

คอนโทรล	คุณสมบัติ	ค่าที่กำหนด
Command1	Name	CmdRun
	Caption	&Run

3. จากนั้นเขียน code เพื่อสร้างออบเจกต์ Recordset ดังนี้

```
Private Sub cmdRun_Click()
Dim Conn As New ADODB.Connection
Dim Cmd = New ADODB.Command
Dim rs As New ADODB.Recordset
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
With Conn
```

```
.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" & _  
"Data Source=D:\student.mdb;Persist Security Info=False"  
.Open
```

เชื่อมต่อฐานข้อมูลด้วย
ออบเจกต์ Connection

```
End With
```

```
With Cmd
```

```
.ActiveConnection = Conn  
.CommandText = "Select * From student"  
.CommandType = adCmdText
```

สร้างชุดคำสั่ง SQL
ให้กับออบเจกต์

```
End With
```

```
Set rs = Cmd.Execute
```

```
List1.Clear
```

```
While Not rs.EOF
```

```
List1.AddItem rs.Fields("FirstName").Value
```

```
Rs.MoveNext
```

```
Wend
```

```
Set rs = Nothing
```

```
Set Cmd = Nothing
```

```
Set Conn = Nothing
```

```
End Sub
```

แสดงข้อมูลจากฟิลด์ชื่อ FirstName
ที่อยู่ในออบเจกต์ Recordset เข้าไป
ในคอนโทรล

2.7 ออบเจกต์ Recordset

เมื่อติดต่อกับฐานข้อมูล ผลลัพธ์ที่ได้กลับมามีคือ ข้อมูลของตารางที่อยู่ในฐานข้อมูลดังกล่าว ซึ่งจะถูกนิยามแทนด้วยออบเจกต์ Recordset ดังนั้นอาจกล่าวได้ว่า ออบเจกต์ Recordset เก็บข้อมูลของแต่ละเรคคอร์ดที่อยู่ในตารางเชื่อมโยงนั่นเอง

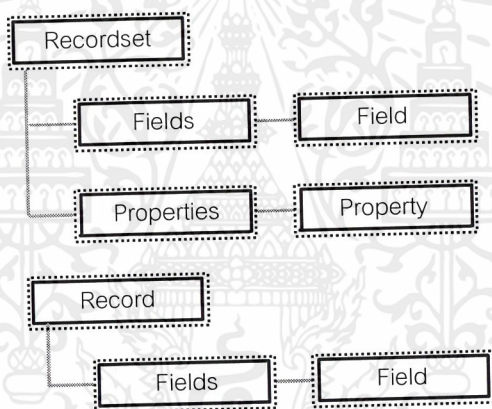
ในการเข้าถึงข้อมูลในฐานข้อมูล ถ้าคุณไม่ได้ระบุเงื่อนไขใดๆ ไว้ในขณะที่เชื่อมต่อออบเจกต์ Recordset ก็จะมีทุกเรคคอร์ดที่มีอยู่ในตาราง แต่ถ้าระบุเงื่อนไขกำกับไว้ด้วยออบเจกต์ Recordset ก็จะมีเฉพาะเรคคอร์ดที่อยู่ในเงื่อนไขที่ได้ระบุไว้เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเชื่อมต่อกับฐานข้อมูลสามารถทำได้หลายวิธี ซึ่งเป็นลักษณะที่สำคัญของ ADO ทำให้เราสามารถสร้างออบเจกต์ Recordset ได้หลายทางเช่นกัน ดังนี้

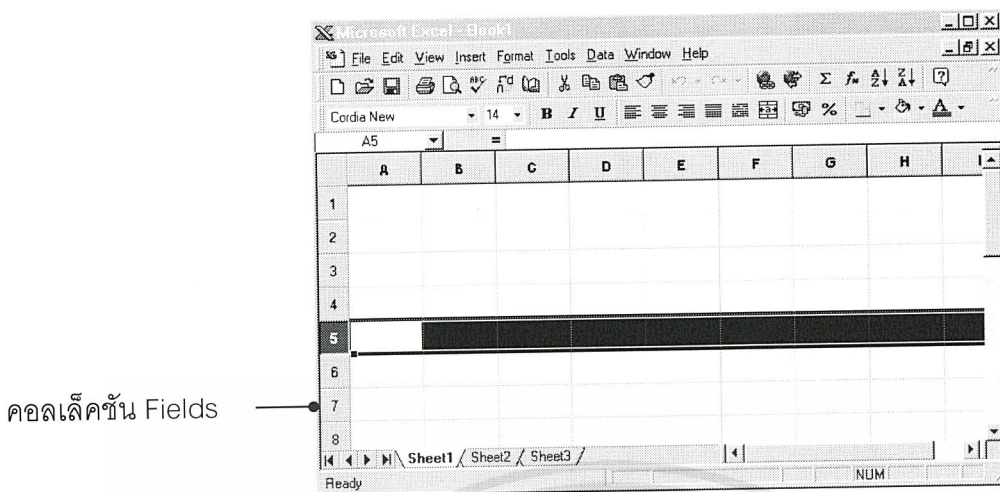
- วิธีที่ 1 - โดยใช้คอนโทรล ADO Data
- วิธีที่ 2 - โดยใช้ออบเจกต์ Connection
- วิธีที่ 3 - โดยใช้ออบเจกต์ Command
- วิธีที่ 4 - โดยใช้ออบเจกต์ Recordset เอง
- วิธีที่ 5 - โดยใช้ DataEnvironment

ใน ADO 2.5 ออบเจกต์ Recordset ถูกปรับปรุงให้มีความยืดหยุ่นมากขึ้น ฟิลด์ และเรคคอร์ดต่างๆ ที่อยู่ภายในออบเจกต์ Recordset จะถูกแบ่งออกเป็นออบเจกต์ย่อยลงไปอีก คือ ออบเจกต์ Field และออบเจกต์ Record ดังรูป



รูปที่ 2.9 ออบเจกต์ Recordset

จากรูปที่ 2.9 ออบเจกต์ Field จะถูกเก็บอยู่รวมกันเป็นคอลเล็คชัน Field ถ้าคุณนึกไม่ออกว่า ออบเจกต์ Recordset, ออบเจกต์ Record และออบเจกต์ Field เกี่ยวกันอย่างไร อยู่ด้วยกันในฐานอะไร ให้คุณนึกถึง Worksheet ของ Excel เป็นภาพที่แสดงให้เห็นความสัมพันธ์ของออบเจกต์ทั้ง 3 ตัว ได้เป็นอย่างดี



แผ่น Sheet1 ทั้งหมด หมายถึง ออบเจ็กต์ Recordset ข้อมูลที่อยู่ในแต่ละเซลล์มองได้ 2 แบบ คือ อยู่ในออบเจ็กต์ Record หรืออยู่ในออบเจ็กต์ Field โดยที่ถ้ากล่าวถึงคอลัมน์ทั้งหมด จะหมายถึง คอลเล็คชัน Fields

และเหตุที่ผู้เขียนกล่าวว่า มีความยืดหยุ่นสูงขึ้น เปรียบเทียบได้ว่า ในแต่ละเซลล์คุณสามารถใช้ฟอนต์ต่างกันได้, ใช้ตัวเอน, ตัวหนา ในหลายๆ เซลล์ได้ เพราะว่า แต่ละเซลล์ถูกมองเป็นวัตถุอันหนึ่ง ที่สามารถคุณสมบัติและเมธอดได้ ซึ่งเป็นผลมาจากที่ไม่ใครซอฟต์แวร์เพิ่มเติมให้มีออบเจ็กต์ Record และออบเจ็กต์ Field ในออบเจ็กต์ Recordset นั้นเอง

2.7.1 วิธีการจัดการจำนวนเร็คคอร์ด

ในการติดต่อกับตารางใดๆ ออบเจ็กต์ Recordset จะเก็บเร็คคอร์ดไว้ตามเงื่อนไขของคุณ แต่เมื่อคุณต้องการแสดงเร็คคอร์ดดังกล่าวออกมา คุณสามารถกำหนดให้ออบเจ็กต์ Recordset แสดงเร็คคอร์ดต่างๆ ออกมาได้ทั้งหมด 4 วิธี คือ

วิธีที่ 1 - แสดงครั้งละ 1 เร็คคอร์ด

ซึ่งเป็นวิธีการที่ใช้กันโดยทั่วไป โดยการใช้งานร่วมกับกลุ่ม Bound Controls

วิธีที่ 2 - แสดงหมดทุกเร็คคอร์ดในคราวเดียว

โดยการใช้งานร่วมกับกลุ่ม Data-Aware Controls นั้นเอง เช่น คอนโทรล DataList, คอนโทรล DataGrid เป็นต้น

วิธีที่ 3 - จำกัดจำนวนเรคคอร์ดที่จะแสดงในแต่ละครั้ง

เป็นวิธีการกำหนดหน้า ใน ADO เรียกว่า Page ให้กับเรคคอร์ด เพื่อแบ่งเรคคอร์ดทั้งหมด ออกเป็น Page ก่อน แล้วจึงแสดงออกมา เช่น สมมติว่ามีเรคคอร์ดทั้งสิ้น 105 เรคคอร์ด กำหนด แสดงครั้งละ 10 เรคคอร์ด ก็จะมีทั้งสิ้น 11 หน้า โดยที่ใน 10 หน้าแรกจะมีหน้าละ 10 เรคคอร์ด ส่วนในหน้าสุดท้าย จะมี 5 เรคคอร์ด

วิธีที่ 4 - จำกัดจำนวนเรคคอร์ด

เป็นการกำหนดว่า ให้ออบเจกต์ Recordset แสดงกี่เรคคอร์ด เช่น สมมติว่ามี 50 เรคคอร์ด แต่กำหนดให้แสดงแค่ 15 เรคคอร์ด หรือกำหนดว่าเรคคอร์ดที่จะแสดงต้องมีข้อมูลที่คุณ ต้องการ โดยการระบุที่คุณสมบัติ Filter เป็นต้น

2.7.2 วิธีการอ้างค่าที่อยู่ในฟิลด์

วิธีการสั่งให้เคอร์เซอร์อ่านค่าที่อยู่ในแต่ละฟิลด์ สามารถทำได้ทั้งหมด 5 แบบ ดังนี้

แบบที่ 1 : อาศัยคอลเล็คชัน Fields กับชื่อของฟิลด์

ตัวอย่างเช่น

```
rs.Fields("StudentID") หรือ
rs.Fields("FirstName").Value
```

แบบที่ 2 : อาศัยคอลเล็คชัน Fields และอ้างลำดับฟิลด์แบบอาร์เรย์ โดยที่ฟิลด์แรกมี

ค่าเท่ากับ 0

ตัวอย่างเช่น

```
rs.Fields(0) หรือ
rs.Fields(1).Value
```

แบบที่ 3 : อาศัยเครื่องหมาย !

ตัวอย่างเช่น

```
rs!StudentID หรือ
rs!FirstName.Value
```

แบบที่ 4 : ใช้ชื่อฟิลด์โดยตรง

ตัวอย่างเช่น

```
rs("StudentID") หรือ
rs("FirstName").Value
```

แบบที่ 5 : อ้างลำดับฟิลด์แบบอาร์เรย์

ตัวอย่างเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

rs(0) หรือ

rs(1).Value

โดย rs หมายถึง ตัวแปรออบเจกต์ Recordset

2.8 การใช้งานภาษา SQL

SQL ย่อมาจากคำว่า Structured Query Language หมายถึง ภาษามาตรฐานกลางที่ใช้สำหรับจัดการข้อมูลในฐานข้อมูล โดยเฉพาะอย่างยิ่งฐานข้อมูลประเภท RDBMS (Relational Database Management System) จะรู้จักภาษา SQL อย่างดี เราจะใช้ SQL เพื่อทำควิรี่ (Query) ข้อมูลที่อยู่ในตาราง ในวัตถุประสงค์ที่ต่าง ๆ กัน เช่น อาจจะต้องการข้อมูลที่มาจกตารางเดียว หรือหลายตารางมาแสดงด้วยกันในเวลาเดียวกัน ดังนั้นการทำควิรี่จึงเป็นการสร้างตารางเสมือนขึ้นมา ประกอบด้วยข้อมูลที่มาจกตารางเดียวหรือหลายตารางก็ได้ เป็นตารางที่ไม่มีอยู่จริงในฐานข้อมูล เป็นมุมมองของข้อมูลในฐานข้อมูลตามที่ต้องการ

2.8.1 โครงสร้างของภาษา SQL

ภาษา SQL ประกอบไปด้วย 3 ส่วนใหญ่ ๆ คือ

1. Data Definition Language (DDL) เป็นกลุ่มคำสั่งในภาษา SQL ที่ใช้สำหรับจัดการโครงสร้างของฐานข้อมูล เช่น การสร้างฐานข้อมูล, ปรับปรุงโครงสร้างของฐานข้อมูล เป็นต้น ตัวอย่างการใช้งานกลุ่มคำสั่ง DDL นี้ก็คือ การสร้างฐานข้อมูลด้วย MS SQL Server 7.0 ก็จะมีการใช้งานคำสั่งกลุ่ม DDL เป็นหลัก

2. Data Manipulation Language (DML) เป็นกลุ่มคำสั่งในภาษา SQL ที่ใช้สำหรับจัดการข้อมูลในฐานข้อมูล เช่น การแสดงข้อมูลแบบมีเงื่อนไข, การลบข้อมูล, การเพิ่มข้อมูล และการแสดงข้อมูลที่มาจกตารางหลายตาราง เป็นต้น

3. กลุ่มฟังก์ชัน Aggregate Function เป็นฟังก์ชันพิเศษของภาษา SQL ที่ทำหน้าที่เฉพาะอย่าง เช่น หาผลรวมเร็คคอร์ด, ค่าสูงสุด เป็นต้น เป็นกลุ่มฟังก์ชันที่มีประโยชน์มาก เพราะช่วยลดภาระให้ไม่ต้องเขียนโค้ดจัดการเอง

สำหรับการใช้งานภาษา SQL ร่วมกับ Visual Basic เพื่อตัดการข้อมูลในฐานข้อมูลจะใช้งานกลุ่มคำสั่ง DML เป็นหลัก

คำสั่งในกลุ่มของ DML จะมีคำสั่งพื้นฐานอยู่ 4 คำสั่งคือ

1. DELETE เป็นคำสั่งที่ใช้สำหรับลบข้อมูลหรือลบเร็คคอร์ดใด ๆ ในตาราง
2. INSERT เป็นคำสั่งที่ใช้สำหรับเพิ่มข้อมูลหรือเพิ่มเร็คคอร์ดใด ๆ เข้าไปในตาราง

3. SELECT เป็นคำสั่งที่ใช้สำหรับเลือกข้อมูลหรือแสดงเรคคอร์ดใด ๆ ที่ต้องการจากตาราง อาจจะมี มาจากตารางเดียว หรือหลายตารางก็ได้
4. UPDATE ใช้สำหรับแก้ไขข้อมูลหรือแก้ไขเรคคอร์ดใด ๆ ในตาราง

2.8.2 ตัวดำเนินการ (Operator)

ตัวดำเนินการเปรียบเทียบที่น่าสนใจ ได้แก่

ตารางที่ 2 ตัวดำเนินการ (Operator)

ตัวดำเนินการ	ความหมาย
=	เท่ากับ (Equal)
<>	ไม่เท่ากับ (Not Equal)
<	น้อยกว่า (Less Than)
>	มากกว่า (Greater Than)
<=	น้อยกว่าหรือเท่ากับ (Less Than or Equal To)
>=	มากกว่าหรือเท่ากับ (Greater Than or Equal To)
Like	เป็นการเปรียบเทียบโดยใช้ตัวอักษรพิเศษ (Wild Card Character) เข้ามาร่วมด้วย

2.8.3 ตัวดำเนินการด้านตรรกะ (Logical Operator)

ตัวดำเนินการด้านตรรกะที่นิยมใช้จะมีอยู่ 3 ชนิดคือ And , Or , และ Not

2.8.4 กลุ่มฟังก์ชัน Aggregate

กลุ่มฟังก์ชัน Aggregate เป็นฟังก์ชันที่ใช้คำนวณทางคณิตศาสตร์ โดยช่วยให้การนำเสนอผลการค้นหาเป็นไปอย่างมีประสิทธิภาพ ซึ่งมีฟังก์ชันพื้นฐานต่อไปนี้

ตารางที่ 3 กลุ่มฟังก์ชัน Aggregate

ชื่อฟังก์ชัน	หน้าที่
AVG ()	หาค่าเฉลี่ยของฟิลด์ จากเรคคอร์ดทั้งหมด
COUNT()	นับจำนวนเรคคอร์ด
FIRST()	หาค่าแรกในฟิลด์
LAST()	หาค่าสุดท้ายในฟิลด์
MAX()	หาค่ามากที่สุด หรือค่าสูงสุด
MIN()	หาค่าน้อยที่สุด หรือค่าต่ำสุด
SUM()	หาผลรวมทั้งหมดของฟิลด์

2.9 ต้นไม้ (TREES)

โครงสร้างต้นไม้เป็นโครงสร้างที่มีระดับ (Hierarchical Structure) อาจกล่าวได้เลยว่า ผู้อ่านทุกคนเคยเห็นโครงสร้างต้นไม้มาแล้วเช่น โครงสร้างแสดงองค์ประกอบของหน่วยงานต่าง ๆ เป็นโครงสร้างต้นไม้ ในการเรียนเรื่องความน่าจะเป็นอาศัยโครงสร้างต้นไม้แสดงหนทางทั้งหมดที่เป็นไปได้ที่เกิดเหตุการณ์

นิยามของต้นไม้

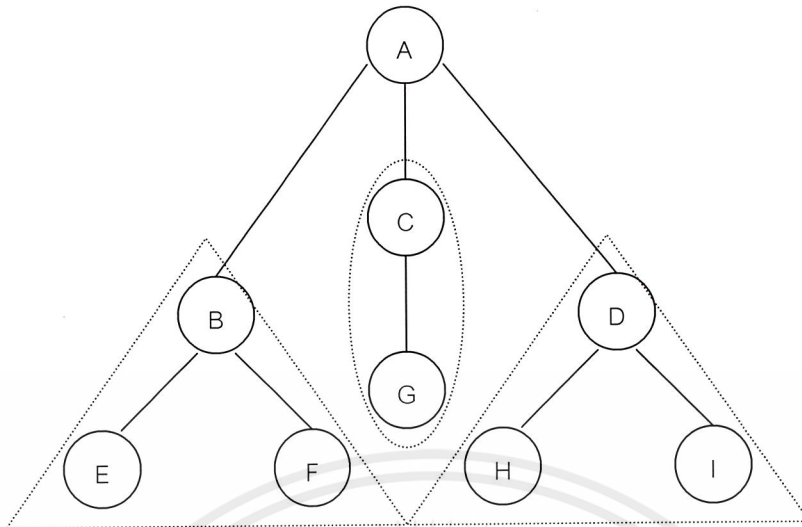
นิยามของต้นไม้เป็นนิยามแบบรีเคอร์ซีฟ (Recursive Definition) ว่า ต้นไม้ประกอบด้วยสมาชิกที่เรียกว่า โหนด (Node) ที่

1. ว่าง (ไม่มีโหนดในต้นไม้) หรือ
2. มีโหนดหนึ่งถือเป็นรากโหนด ส่วนโหนดที่เหลือแบ่งเป็นต้นไม้ย่อย T_1, T_2, \dots, T_k ($k \geq 0$) โดยต้นไม้ย่อยมีคุณสมบัติเป็นต้นไม้เช่นกัน

ความสัมพันธ์ระหว่างรากโหนดและรากโหนดของต้นไม้ย่อยเป็นไปในลักษณะพ่อกับลูกคือราก

โหนด

เป็นโหนดพ่อ และรากโหนดของต้นไม้ย่อยเป็นโหนดลูก



รูปที่ 2.10 แสดงรูปร่างต้นไม้

ตัวอักษรหรือตัวเลขในโหนดเป็นชื่อของโหนดหรืออาจเป็นค่าที่อยู่ในโหนด ต้นไม้นี้มีโหนด A เป็นรากของต้นไม้ ต้นไม้ย่อยของ A มี 3 ต้นคือ ต้นไม้ T1, T2 และ T3 ที่มี B, C และ D เป็นรากต้นไม้ B มีต้นไม้ E และ F เป็นต้นไม้ย่อย ต้นไม้ C มีต้นไม้ G เป็นต้นไม้ย่อย ต้นไม้ D มี H และ I เป็นต้นไม้ย่อย หรือพูดอีกอย่างว่า โหนด A เป็นโหนดพ่อของ B, C และ D โหนด B เป็นโหนดพ่อของ E และ F โหนด C เป็นโหนดพ่อของ G และโหนด D เป็นโหนดพ่อของ H และ I หรือในทางกลับกัน โหนด B, C และ D เป็นลูกของโหนด A โหนด E และ F เป็นโหนดลูกของ B

โหนดพี่น้อง (Brother Node) คือโหนดที่มีพ่อเดียวกัน จากต้นไม้รูป 1 โหนด B, C และ D เป็นโหนดพี่น้องกัน โหนด E และ F เป็นโหนดพี่น้องกัน H และ I เป็นโหนดพี่น้องกัน

กิ่ง (Branch) หรือเอดจ์ (Edge) คือเส้นที่เชื่อมต่อระหว่างพ่อกับลูก โหนดที่ไม่มีลูก เช่นโหนด E, F, G, H และ I เรียกว่า ลีฟโหนด (Leaf Node) ส่วนโหนดที่ไม่ใช่ราก และไม่ใช่อีฟโหนดเช่น โหนด B, C และ D เรียกว่าโหนดกิ่ง (Branch Node)

ดีกรี (Degree) ของโหนด X ใด ๆ คือจำนวนลูกของโหนดนั้น ๆ เช่น ดีกรีของลีฟโหนดเท่ากับ 0

โหนดดีเซนแดนซ์โหนดหรือโหนดที่มาจากที่หลังทันที (Direct Decendent Node) ของโหนด X ใด ๆ คือโหนดที่เป็นลูกของโหนด X นั้น เช่น B, C และ D เป็นโหนดดีเซนแดนซ์ของโหนด A

ดีเซนแดนซ์โหนดคือโหนดลูกของ X และโหนดที่เป็นดีเซนแดนซ์ของลูกของ X เช่น ดีเซนแดนซ์ของโหนด A คือทุกโหนดที่เหลือในต้นไม้

โหนดแอนเซสเตอร์โหนดคือโหนดที่มาก่อนทันที (Direct Ancestor Node) ของโหนด X ใด ๆ (ยกเว้นรูตโหนด) คือโหนดพ่อของ X นั้นเช่น โหนด E และ F มีโหนด B เป็นโหนดแอนเซสเตอร์ และ โหนด B มี A เป็นโหนดแอนเซสเตอร์

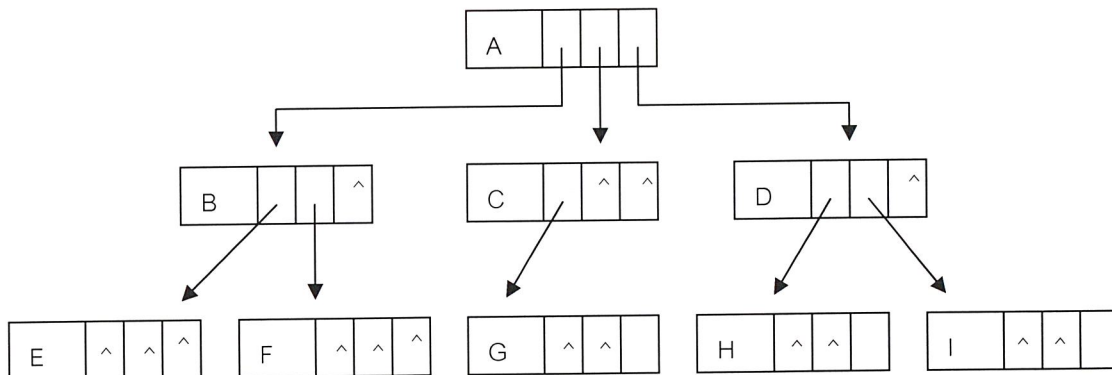
แอนเซสเตอร์ของโหนด X ใด ๆ คือโหนดพ่อและโหนดที่เป็นแอนเซสเตอร์ของโหนดพ่อของ X เช่น A, B เป็นแอนเซสเตอร์ของ E และ F

เลเวล (Level) เป็นหมายเลขแสดงระดับของโหนดในต้นไม้รูตโหนดถูกกำหนดให้อยู่ในระดับ 0 ส่วนโหนดดีเซนแดนซ์อยู่ในระดับถัดไป คือ 1 คือถ้าโหนด X (ยกเว้นลีฟโหนด) ใด ๆ อยู่ในระดับ l ลูกของโหนด X จะอยู่ในระดับ $l + 1$

ความสูงหรือความลึก (Height หรือ Depth) ของต้นไม้คือ ระดับสูงสุดของต้นไม้ นั้นจากต้นไม้รูป 1 โหนด A อยู่ระดับ 0 โหนด B, C และ D อยู่ในระดับ 1 โหนด E, F, G, H และ I อยู่ระดับ 2 และมีความสูงเท่ากับ 2

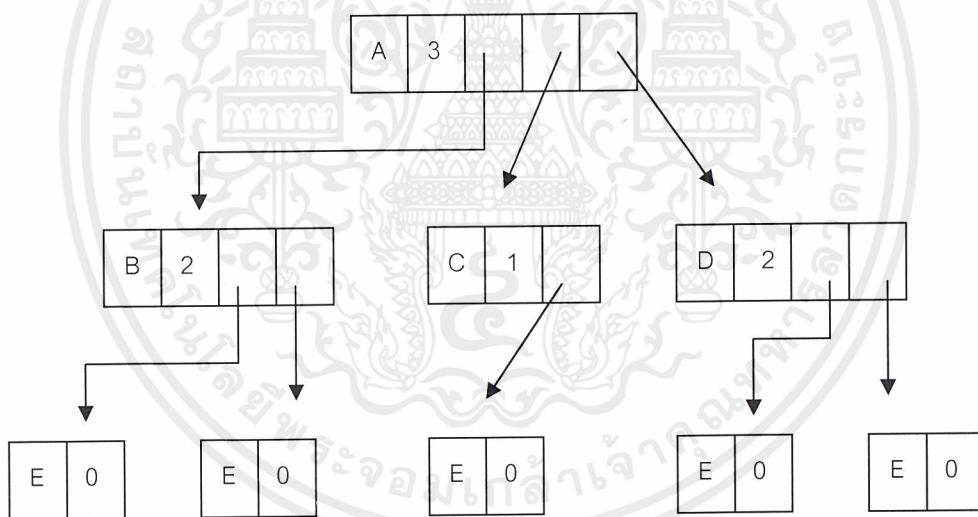
2.9.1 การแทนที่ต้นไม้แบบไดนามิก

เนื่องจากโหนดในแต่ละต้นไม้มีจำนวนลูกไม่เท่ากันการแทนที่ข้อมูลต้นไม้แบบสแตติกจึงไม่เหมาะสม การแทนที่ข้อมูลแบบไดนามิกที่แสดงต่อไปมี 2 วิธีการคือหนึ่งแต่ละโหนดมีพอยเตอร์เพื่อชี้ยังลูกเท่ากับจำนวนลูกหรือต้นไม้ย่อยสูงสุดเช่นจากต้นไม้รูป 2.10 จำนวนต้นไม้ย่อยสูงสุดเป็น 3 ดังนั้นต้นไม้ที่แทนที่โดยใช้ลิงค์เป็นดังรูป



รูปที่ 2.11 แสดงตัวอย่างต้นไม้โดยใช้ลิงค์

ข้อเสียของวิธีการนี้คือ ถ้าแต่ละโหนดมีต้นไม้ย่อยที่ต่างกันมาก เช่นถ้าจำนวนต้นไม้ย่อยสูงสุดเป็น 15 ส่วนโหนดอื่นมีต้นไม้ย่อยเพียง 1 หรือ 2 เท่านั้นจะเสียเนื้อที่ส่วนพอยเตอร์ไปมาก วิธีการที่สองอาจระบุจำนวนต้นไม้ย่อยไปด้วย และพอยเตอร์มีเท่ากับตัวเลขที่ระบุ ดังรูป 2.12



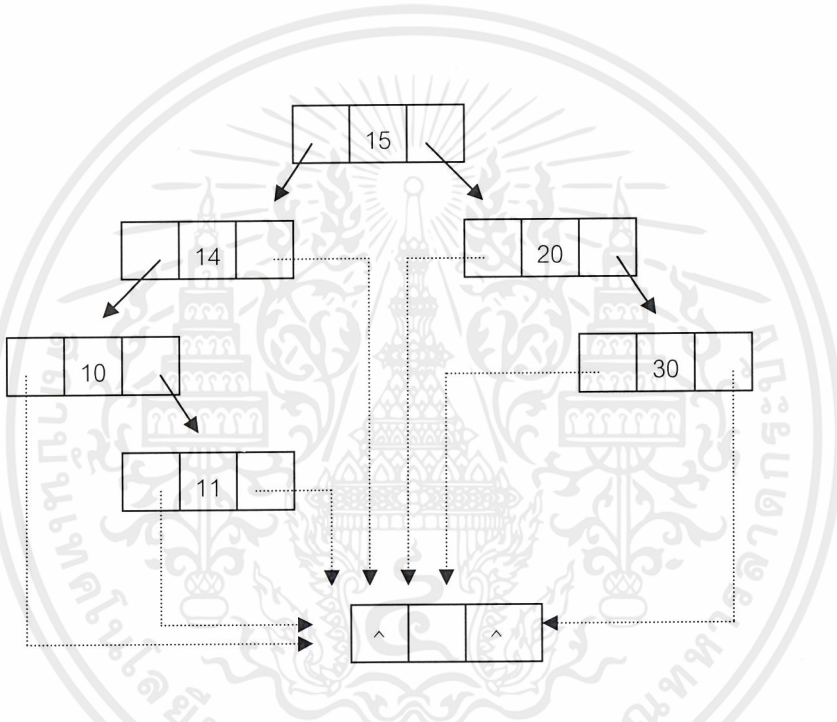
รูปที่ 2.12 แสดงตัวอย่างต้นไม้โดยใช้ลิงค์

2.9.2 การท่องต้นไม้ไบนารีโดยใช้สแตก

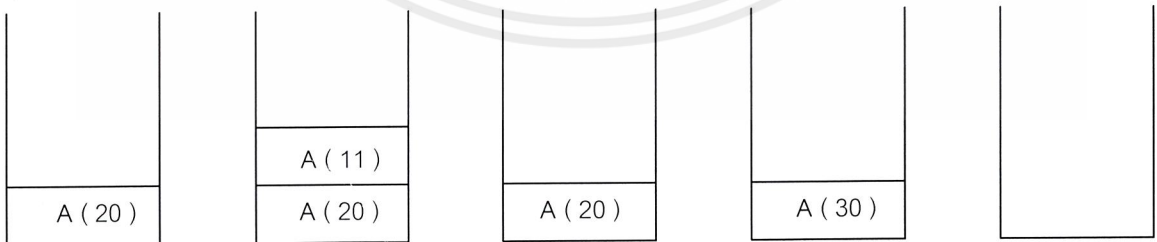
เราสามารถหลีกเลี่ยงการท่องไปในต้นไม้ในลักษณะรีเคอร์ซีฟ โดยการใช้อัดแดกเก็บแอดเดรสของโหนดที่พบก่อนแต่ควรถูกท่องทีหลัง เช่นการท่องแยะพรีออเดอร์ พิมพ์ค่าในรู้ตทางต้นไม้ย่อยซ้ายและเก็บค่าพอยเตอร์ของต้นไม้ย่อยขวา อัลกอริทึม $prestack(T)$ ท่องต้นไม้ไบนารีอย่างพรีออเดอร์โดยใช้สแตก เป็นดังนี้โดยให้ L, R เป็นพอยเตอร์ชี้ต้นไม้ย่อยซ้ายและขวาของรู้ตที่ชี้ด้วยพอยเตอร์ P

1. $P \leftarrow T$ (T เป็นพอยเตอร์ชี้วัตถุของต้นไม้)
2. ทำจนกระทั่ง สแตกว่าง
 - 2.1 พิมพ์ค่าในโหนด P
 - 2.2 ถ้า $R \llcorner \wedge$, push ในสแตก
 - 2.3 ถ้า $R \llcorner \wedge$, $P \leftarrow L$
 มิฉะนั้น $P \leftarrow$ พอยเตอร์ที่ pop จากสแตก

รูปที่ 2.14 a) ถึง e) แสดงค่าที่อยู่ในสแตกโดยให้ A (ค่าในโหนด) แทนแอดเดรสของโหนด เมื่อใช้ อัลกอริทึม prestack (T) กับต้นไม้รูป 2.13



รูปที่ 2.13 แสดงตัวอย่างต้นไม้โดยใช้ลิงค์



- พิมพ์ 15 พิมพ์ 14, 10 พิมพ์ 11 พิมพ์ 20 พิมพ์ 30
- a) b) c) d) e)

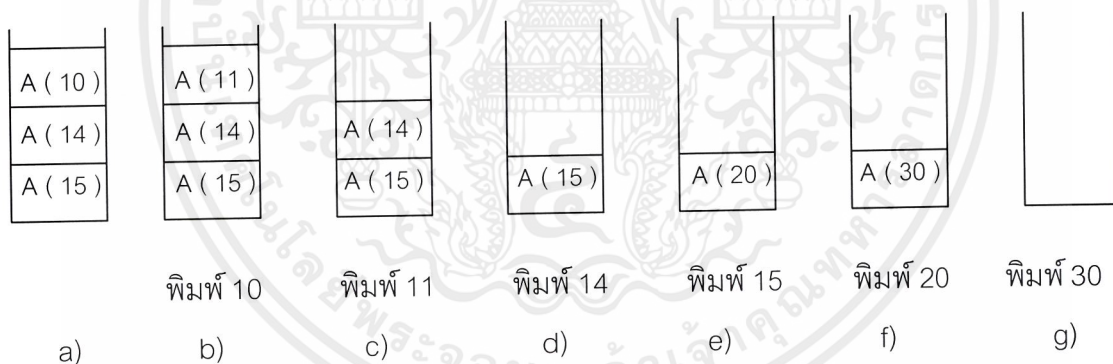
รูปที่ 2.14 แสดงค่าที่อยู่ในสแตกเมื่อใช้อัลกอริทึม prestack (T) กับต้นไม้รูป 2.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การท่อดันไม้ย่อกอินออกเคอร์ ใช้สแตกแทนการรีเคอร์ซีฟ อัลกอริทึม instack (T) นี้

1. $P \leftarrow T$
2. ในระหว่างที่ $P \neq NIL$ ทำ
 - 2.1 ในระหว่างที่ $P \neq NIL$ ทำ
 - 2.1.1 push พอยเคอร์ซีโหนด P ลงสแตก
 - 2.1.2 $P \leftarrow L$ (* ไปยังต้นไม้ย่อยซ้ายของ P *)
 - 2.2 (* เมื่อ $P = NIL$ *)
 - ทำ (2.2.1 ถึง 2.2.3) จนกระทั่งสแตกว่าง หรือ $P = NIL$
 - 2.2.1 $P \leftarrow$ พอยเคอร์ซีโหนด pop จากสแตก
 - 2.2.2 พิมพ์ข้อมูลในโหนด P
 - 2.2.3 $P \leftarrow R$ (* ไปยังต้นไม้ย่อยขวาของ P *)
3. จบงาน

รูป 2.15 a) ถึง g) แสดงค่าที่อยู่ในสแตก ขณะที่ทำตามอัลกอริทึม instack (T) โดยใช้ต้นไม้รูป 2.13 เช่นเดิม



รูปที่ 2.15 แสดงค่าที่อยู่ในสแตก ตามอัลกอริทึม instack (T) โดยใช้ต้นไม้รูป 2.13

สแตกที่ใช้ช่วยท่อดันไม้เป็นสแตกของพอยเคอร์ซีโหนด บรรยายปาสคาล ดังนี้

CONTS Smax = 10;

TYPE pointer = node;

Stack = ARRAY[1..Smax] OF pointer ;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 การสร้าง และการจัดการระบบฐานข้อมูล SQL Server 7.0

แนะนำเข้าสู่ระบบฐานข้อมูล

ก่อนจะเริ่มใช้งานเพื่อเก็บข้อมูลได้นั้น จะต้องมีการสร้าง และจัดการกับระบบฐานข้อมูล ให้เรียบร้อยเสียก่อน สำหรับใน SQL Server 7.0 ได้มีการปรับปรุงรูปแบบ ตลอดจนโครงสร้างของระบบฐานข้อมูลได้ดีขึ้นจากในเวอร์ชันเดิมเป็นอย่างมาก

ในการสร้างฐานข้อมูล (Database) หนึ่ง ๆ โดยปกติแล้วจะประกอบไปด้วยไฟล์อย่างน้อย 2 ไฟล์ คือ

1. ไฟล์ข้อมูล (Data File) เป็นไฟล์ที่เก็บข้อมูลจริง
2. ไฟล์ทรานแซคชัน (Transaction Log File) เป็นไฟล์สำหรับเก็บโอเปอเรชัน หรือ ข้อมูลล็อก (Log) ซึ่งเกิดจากการทำงานกับข้อมูลจริง

สำหรับ SQL Server 7.0 แล้ว สามารถสร้างให้ไฟล์ข้อมูล หรือไฟล์ทรานแซคชันล็อก มีมากกว่า 1 ไฟล์ได้ ทั้งนี้เพื่อแยกการทำงานที่ชัดเจน และประโยชน์ในแง่ของการบำรุงรักษาอีกด้วย

2.10.1 การจัดการระบบฐานข้อมูล

SQL Server 7.0 มีวิธีการจัดการระบบฐานข้อมูล ทั้งการสร้าง, เปลี่ยนแปลง หรือ ทำลาย รวมถึงการกำหนดขอบเขตต่าง ๆ ของฐานข้อมูล โดยใช้เครื่องมือสำหรับบริหารงานอย่าง Enterprise Manager ซึ่งง่ายต่อการใช้งานมากกว่า การใช้คำสั่งของ Transact – SQL จึงทำให้ผู้ดูแลระบบจัดการกับฐานข้อมูลได้อย่างไม่ยากเย็นนัก

ถึงแม้ Transact – SQL จะใช้งานยากกว่า เพราะต้องคอยจดจำรูปแบบ ไวยากรณ์ของคำสั่งต่างๆ แต่แท้ที่จริงแล้วเบื้องหลังการใช้งานจาก Enterprise Manager คือการแปลงข้อมูลที่ถูกกำหนดไว้ให้กลายเป็นคำสั่งของ Transact – SQL ก่อน แล้วจึงทำตามคำสั่งเหล่านั้นต่อไป

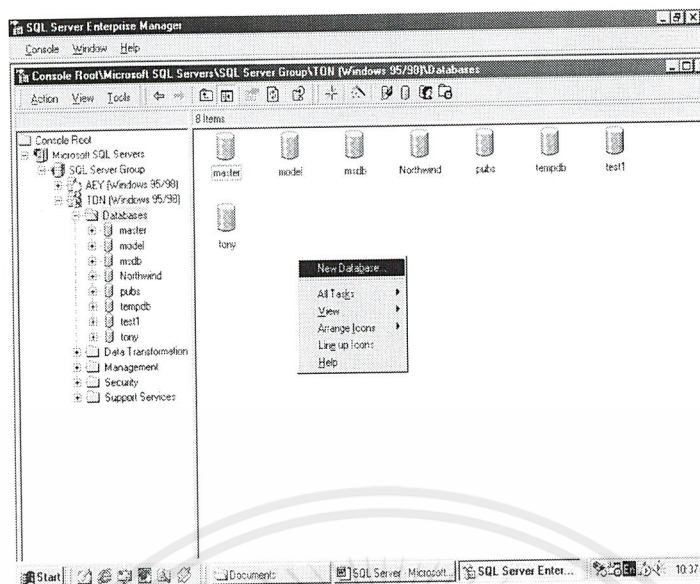
ในบางเหตุการณ์ที่ Enterprise Manager ไม่สามารถทำงานได้ ผู้บริหารระบบก็ต้องใช้ Transact – SQL เข้ามาช่วย ดังนั้นจึงจำเป็นอย่างยิ่งที่นักบริหารระดับมืออาชีพต้องเรียนรู้ และเข้าใจการใช้งานของ Transact – SQL เป็นอย่างดี

2.10.2 การสร้างฐานข้อมูล

สำหรับการสร้างฐานข้อมูลโดยใช้ Enterprise Manager มีขั้นตอนดังนี้

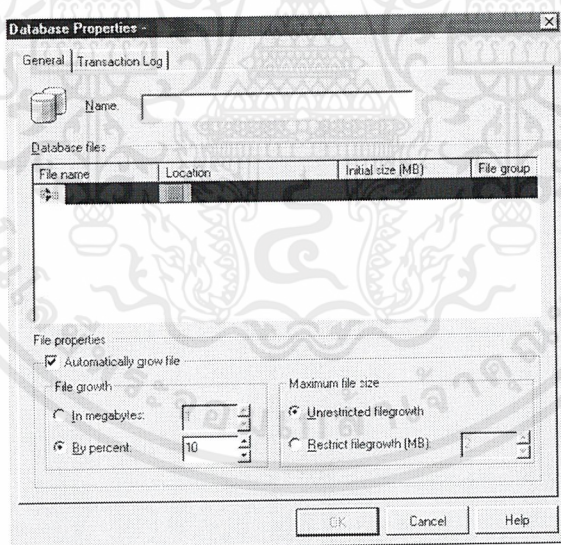
1. ติดต่อ SQL Server ด้วย Enterprise Manager
2. ขยายโพลเดอริไปยัง Database
3. คลิกขวายังพื้นที่ว่างของเฟรมขวา พร้อมทั้งเลือก New Database... ดังรูปที่ 2.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 แสดงการสร้างฐานข้อมูลโดยใช้ Enterprise Manager

ซึ่งทำให้ได้อะด็อกสำหรับการสร้างฐานข้อมูลปรากฏขึ้น ดังรูปที่ 2.17

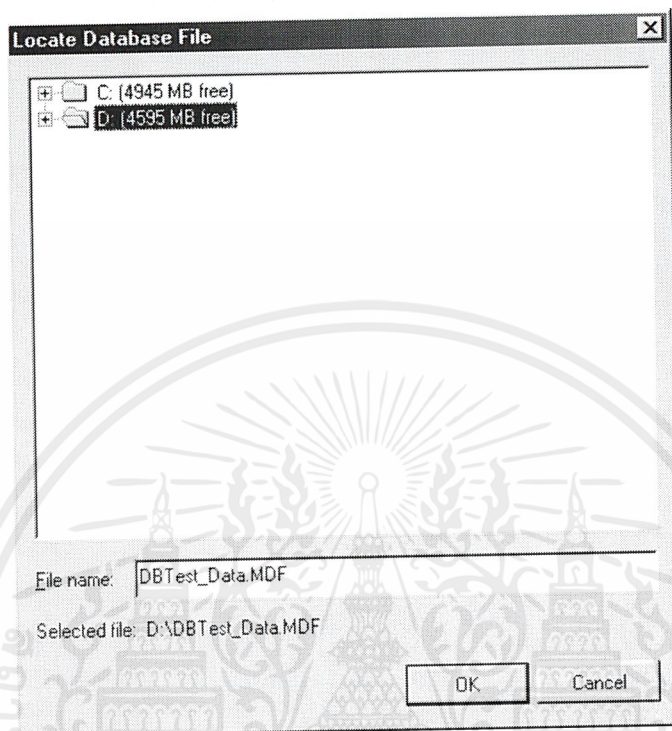


รูปที่ 2.17 ได้อะด็อกสำหรับการสร้างฐานข้อมูล

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. นอกจากนี้ในส่วนของ Database files ก็สามารเปลี่ยนตำแหน่งที่เก็บไฟล์ของฐานข้อมูลได้ ปกติจะอยู่ในไดเรกทอรี \MSSQL7\Data โดยการคลิกที่ปุ่ม ... ซึ่งจะได้ไดอะล็อกบ็อกซ์ สำหรับเปลี่ยนพาทที่ต้องการ ดังรูปที่ 2.18

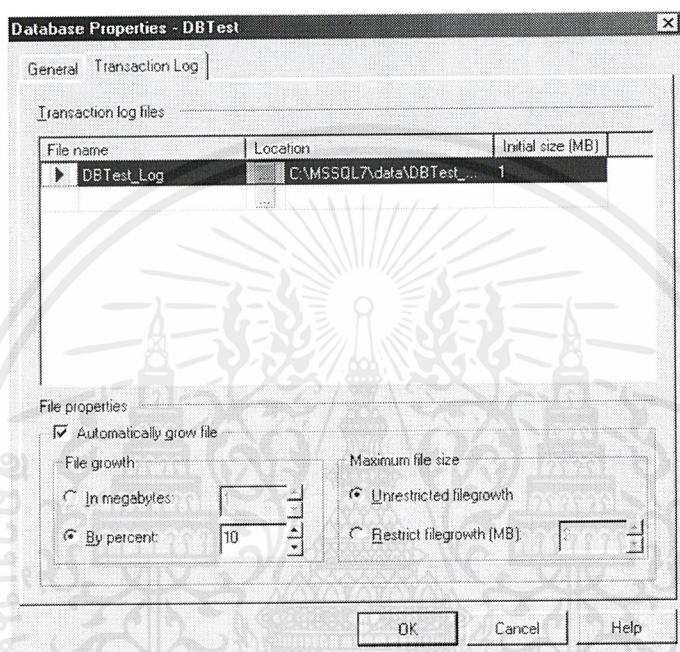


รูปที่ 2.18 แสดงการเปลี่ยนตำแหน่งที่เก็บไฟล์ของฐานข้อมูล

3. กำหนดขนาดเริ่มต้นของฐานข้อมูลที่ช่อง Initial size (MB) ซึ่งค่าปกติจะเป็น 1 MB ส่วนช่อง File group ใช้สำหรับสร้างฐานข้อมูลให้เป็นแบบไฟล์กรุป (Filegroups) ตามปกติจะกำหนดค่าให้เป็น PRIMARY
4. ในส่วนของ File properties เช็คบ็อกซ์ที่ Automatically grow file ได้ถูกเลือกไว้อยู่แล้วซึ่งทำให้สามารถกำหนดลักษณะการเพิ่มขนาดของฐานข้อมูล เมื่อฐานข้อมูลเกิดเต็มขึ้นมาได้ ซึ่งแบ่งออกเป็น
- File growth สำหรับระบุการเพิ่มในแต่ละครั้ง ซึ่งสามารถเลือกได้เป็น In megabytes: คือเพิ่มตามจำนวนที่กำหนดไว้ หน่วยเป็น MB และ By percent เพิ่มขึ้นเป็นจำนวนเปอร์เซ็นต์จากขนาดของฐานข้อมูลเดิม
 - Maximum file size สำหรับกำหนดขนาดของฐานข้อมูลสูงสุดที่สามารถเป็นไปได้ ทั้งสิ้น ซึ่งสามารถเลือกได้เป็น Unrestricted filegrowth คือไม่จำกัดขนาดของ

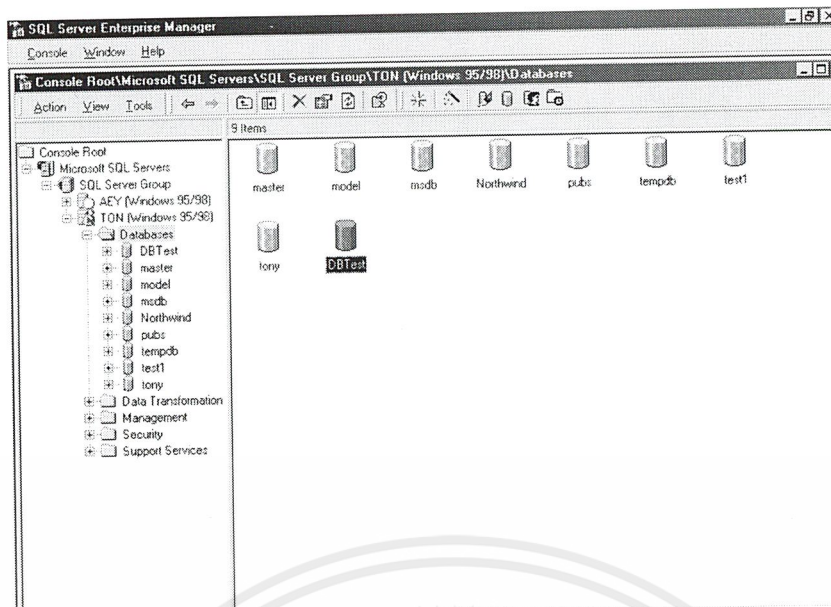
หรือจนกว่าเนื้อที่ว่างบนดิสก์จะเต็ม และ Restricted filegrowth (MB): เพื่อจำกัดขนาดของฐานข้อมูลที่สามารถเป็นไปได้สูงสุด

- เลือกที่แท็บ Transaction Log จะพบว่าลักษณะการกำหนดชื่อ และตำแหน่งมีลักษณะคล้ายตามการกำหนดฐานข้อมูลอย่างอัตโนมัติ ดังแสดงในรูปที่ 2.19 โดยเริ่มต้นด้วยชื่อของฐานข้อมูล ตามด้วย _Log และมีนามสกุลเป็น .LDF หากไม่เหมาะสมสามารถเปลี่ยนชื่อรวมทั้งตำแหน่งของพาร์ทิเก็บบนได้อีกตามต้องการ



รูปที่ 2.19 แสดงการกำหนดชื่อ และตำแหน่งมีลักษณะคล้ายตามการกำหนดฐานข้อมูล

- เมื่อกำหนดข้อมูลเสร็จให้คลิกปุ่ม เป็นอันเสร็จสิ้นขั้นตอนของการสร้างฐานข้อมูล ซึ่งจะได้สัญลักษณ์รูปทรงประบอกสี่เหลี่ยม แสดงถึงฐานข้อมูลที่เพิ่งสร้างปรากฏขึ้น ดังรูปที่ 2.20

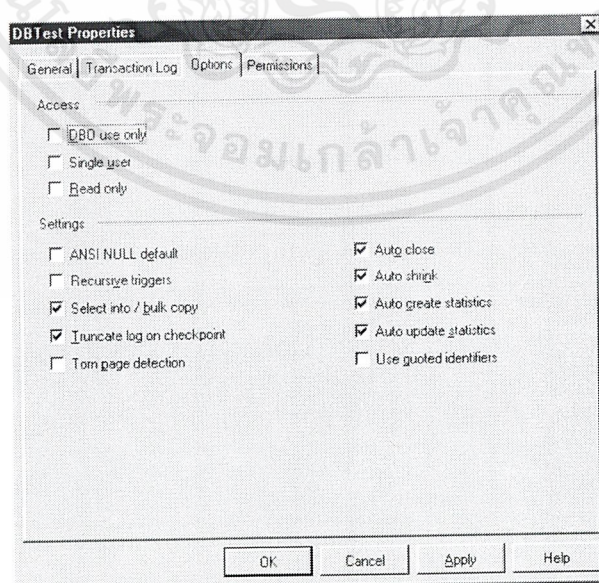


รูปที่ 2.20 แสดงถึงฐานข้อมูลที่สร้างขึ้น

การกำหนดขอบชั้นของฐานข้อมูล

หลังจากที่สร้างฐานข้อมูลเรียบร้อยแล้ว เราสามารถกำหนดขอบชั้นให้ฐานข้อมูลมีลักษณะต่างๆ ดังนี้

1. จากรูป 2.20 ให้ดับเบิลคลิกยังฐานข้อมูลที่ต้องการกำหนดขอบชั้น หรือคลิกขวาที่ฐานข้อมูลนั้น แล้วเลือก Properties หรือจากเมนูเลือก Action > Properties ในขณะที่ฐานข้อมูลได้ถูกไฮไลต์จะได้ผลลัพธ์ดังรูปที่ 2.21



รูปที่ 2.21 แสดงการกำหนดขอบชั้นให้ฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เลือกที่แท็บ Options มีการกำหนดเป็น 2 ส่วนคือ

2.1 Access ประกอบไปด้วย

- DBO use only เป็นการระบุให้ผู้ที่สามารถใช้งานฐานข้อมูลนี้ได้คือ สมาชิกของ dbo_owner เท่านั้น ปกติมักจะถูกกำหนดเมื่อต้องการทำ Performance Maintenance ฐานข้อมูล ซึ่งจะไม่ยอมให้มีการใช้งานจากผู้ใช้เกิดขึ้นนั่นเอง
- Single user เป็นการกำหนดให้ผู้ใช้สามารถติดต่อเพื่อขอใช้งานฐานข้อมูลได้เพียงครั้งละคนเท่านั้น
- Read only เป็นการกำหนดห้ามไม่ให้ฐานข้อมูลมีการเปลี่ยนแปลงเกิดขึ้น ผู้ใช้งานสามารถติดต่อเพื่อเข้ามาอ่านข้อมูลอย่างเดียวเท่านั้น ไม่สามารถเขียนข้อมูลลงไปเ็นฐานข้อมูลนี้ได้

2.2 Settings ประกอบไปด้วย

- ANSI NULL default เป็นการกำหนดให้ฟิลต์ในแต่ละตารางของฐานข้อมูลที่ถูกสร้างขึ้นใหม่ มีค่าดีฟอลต์เป็น NULL เสมอ
- Recursive triggers เป็นเหตุการณ์ที่เกิดขึ้นเมื่อมีการเปลี่ยนแปลงข้อมูลจากตารางหนึ่ง ทำให้เกิดการทริกเกอร์ (trigger) ไปยังอีตารางหนึ่ง ผลคือตารางที่สองซึ่งถูกทริกเกอร์ จะทำการทริกเกอร์กลับไปยังตารางแรก
- Select into / bulk copy เป็นการยินยอมให้การกระทำใดๆ กับฐานข้อมูลสามารถทำได้โดยตรงกับตัวข้อมูลไม่จำเป็นต้องผ่านทรานแซ็คชันล็อกก่อน
- Truncate log on checkpoint เป็นการระบุให้มีการตัดรายการของทรานแซ็คชันล็อก หลังจากข้อมูลได้ถูกเขียนลงดิสก์เรียบร้อยแล้วออกไป ซึ่งระบบจะใช้จุดเช็คพอยต์ (Checkpoint) เป็นเกณฑ์
- Torn page detection เป็นการตรวจสอบเมื่อมีเหตุการณ์เขียนข้อมูลที่มีลักษณะเสียหายไม่ครบถ้วนเกิดขึ้น หรือเกิดการคอร์รัปต์ (corrupt) ของข้อมูลนั่นเอง
- Auto close เป็นการกำหนดให้มีการปิดฐานข้อมูลหลังจากไม่มีใครใช้งาน มีประโยชน์คือ ช่วยประหยัดทรัพยากร (resource) ของระบบ
- Auto create statistics เป็นการกำหนดให้มีการทำสถิติในแต่ละฟิลด์ข้อมูล ซึ่งถูกใช้ในตอนทำ Query Optimizer

- Auto update statistics เป็นการกำหนดให้มีการอัปเดตการทำสถิติของฟิลด์ ข้อมูลได้อัตโนมัติ
- Use quoted identifiers เป็นการกำหนดให้มีการใช้เครื่องหมายคำพูด (“...”) (Double Quotation Mark) ในการอ้างชื่อของออบเจกต์ต่างๆ เช่น ชื่อฟิลด์ ชื่อตาราง เป็นต้น นอกจากนี้ยังรวมถึงการอ้างถึงคำสงวน (Reserved Word) อีกด้วย

2.10.3 การสร้างตารางและกำหนดประเภทของข้อมูล

ส่วนประกอบของตาราง (Table)

ตารางจะประกอบไปด้วย 2 ส่วนหลักคือ คอลัมน์ (Column) (มองในแนวตั้ง) สำหรับระบุประเภทของข้อมูลและแถว (Row) (มองในแนวนอน) ซึ่งประกอบไปด้วยข้อมูลของแต่ละคอลัมน์ในแถวนั้นๆ ตัวอย่างเช่น ตารางพนักงาน ในแต่ละแถวเป็นตัวแทนของพนักงานแต่ละคน โดยในแต่ละคอลัมน์แทนข้อมูลต่างๆ ของพนักงาน เช่น ชื่อ, ที่อยู่, อายุ และเพศ เป็นต้น

1. คอลัมน์ (Column) หรือ ฟิลด์ (Field) จะถูกกำหนดเป็นชื่อที่ไม่ซ้ำกันสำหรับในตารางหนึ่งๆ โดยมีการระบุประเภทของข้อมูลที่จะถูกนำมาเก็บยังฟิลด์นี้ รวมถึงการกำหนดขนาดความยาวของข้อมูลที่สามารถรองรับในแต่ละฟิลด์เหล่านี้ได้ หรือความสามารถในการยอมรับข้อมูลที่ไม่ทราบค่า (Unknown) ซึ่งรู้จักกันดีในนามของ Null Value

2. แถว (Row) หรือ เร็คคอร์ด (Record) แทนลักษณะของสิ่งหนึ่งสิ่งใดที่อยู่ภายในบรรทัดเดียวกัน อธิบายง่าย ๆ ก็คือ ข้อมูลที่เกิดขึ้นในแต่ละคอลัมน์โดยพิจารณาที่บรรทัดเดียวกันเท่านั้น สำหรับความสามารถของ SQL Server 7.0 ภายใน 1 ฐานข้อมูลสามารถบรรจุตารางไว้ได้มากถึง 2 พันล้านตาราง และภายใน 1 ตารางสามารถมีจำนวนคอลัมน์ได้มากถึง 1,024 คอลัมน์ โดยแต่ละคอลัมน์สามารถเก็บข้อมูลที่มีขนาดใหญ่ถึง 8,000 ไบต์ ทั้งนี้ไม่นับรวมข้อมูลประเภท Text และ Binary ซึ่งเก็บได้ถึง 2 GB แต่ไม่ได้ถูกเก็บไว้ในตารางจริง มีเพียงพอยเตอร์ (Pointer) ที่สามารถลิงค์ไปยังข้อมูล เก็บไว้แทนเท่านั้น

3. ประเภทของข้อมูล (Data Type) ขั้นตอนหนึ่งของการสร้างตารางคือ การกำหนดลักษณะหรือประเภทของข้อมูลที่จะจัดเก็บลงยังตารางของฐานข้อมูล สำหรับประเภทของข้อมูลใน SQL Server มีอยู่หลายลักษณะทั้ง ตัวเลข ตัวอักษร ซึ่งแยกย่อยได้อีกในหลายรูปแบบ ดังนี้

3.1 Strings (สตริง) เป็นประเภทของข้อมูลที่สามารถใช้เก็บอักขระต่างๆ ร่วมกันทั้งตัวอักษร ตัวเลข หรือแม้แต่สัญลักษณ์ โดยสามารถเก็บข้อมูลในแต่ละคอลัมน์สูงสุดได้ไม่เกิน 8,000 อักขระ นอกจากนี้สตริงยังแบ่งการเก็บข้อมูลออกเป็น 2 ลักษณะ คือ

- 3.1.1 char (n) เป็นการเก็บข้อมูลของสตริง โดยกำหนดจำนวนสูงสุดเท่ากับ n อักขระ หากข้อมูลที่ถูกจัดเก็บมีจำนวนมากกว่าที่กำหนดไว้ ข้อมูลจะถูกตัดทอนให้เหลือเพียง n อักขระตัวแรกเท่านั้น นอกจากนี้ยังมีข้อที่น่าสนใจคือการกำหนดจำนวนอักขระเป็น n อักขระจะเป็นการจองขนาดข้อมูลที่เท่ากับจำนวนที่กำหนดไว้ หากใช้ไม่ครบ ข้อมูลที่ถูกจัดเก็บจริงจะต้องจัดเก็บช่องว่าง (Space) เพิ่มเติมให้ครบจำนวนเข้าไปด้วย
- 3.1.2 varchar (n) มีลักษณะเช่นเดียวกับ char แต่ต่างกันตรงที่การกำหนดขนาดของข้อมูล ไม่ได้ถูกจองเท่ากับจำนวนที่ได้รับไว้ นั่นหมายความว่าหากใช้ไม่ครบ ข้อมูลที่ถูกจัดเก็บจริงจะเก็บตามจำนวนอักขระที่รับเข้ามาเท่านั้น ตัวอย่างของทั้ง 2 แบบ แสดงในตารางที่ 4

ตารางที่ 4 แสดงตัวอย่างการจัดเก็บข้อมูลประเภทสตริง

ข้อมูลที่ต้องการจัดเก็บ	ข้อมูลที่ถูกจัดเก็บจริง char(6)	ข้อมูลที่ถูกจัดเก็บจริง varchar(6)
Microsoft	Micros	Micros
SQL	SQL***	SQL
Server	Server	Server

3.2 Binary (ไบนารี) เป็นประเภทข้อมูลที่จัดเก็บค่าแบบไบนารี คือ เรียงกันเป็นชุดของตัวเลข 0 และ 1 แต่ในการนำค่าข้อมูลเข้า และออกจากระบบจะแสดงให้เห็นเป็นค่าของตัวเลขฐาน 16 แทน โดยในแต่ละคอลัมน์สามารถเก็บข้อมูลสูงสุดได้ไม่เกิน 8,000 ไบต์ นอกจากนี้ไบนารียังแบ่งการเก็บข้อมูลออกเป็น 2 ลักษณะ คือ

3.2.1 binary (n) เป็นการเก็บค่าของข้อมูลให้เป็นแบบไบนารี โดยมีความหมายคล้ายกันกับ char (n) เช่น สามารถกำหนดจำนวนของข้อมูลที่ต้องการเก็บสูงสุดได้เท่ากับ n ไบต์ และเช่นเดียวกัน หากข้อมูลที่ถูกจัดเก็บมีจำนวนมากกว่าที่กำหนดไว้ ข้อมูลจะถูกตัดทอนให้เหลือเพียง n ไบต์แรกเท่านั้น นอกจากนี้ยังมีคุณสมบัติที่ต้องทำการจัดเก็บช่องว่าง (Space) เพิ่มเติมให้ครบจำนวนเมื่อข้อมูลที่ถูกจัดเก็บมีขนาดน้อยกว่าที่กำหนดไว้อีกด้วย

3.2.2 varbinary (n) มีลักษณะเช่นเดียวกับ binary และมีความหมายคล้ายกันกับ varchar (n) กล่าวคือข้อมูลที่ถูกจัดเก็บจริงจะจัดเก็บตามจำนวนไบต์ที่รับเข้ามา

เท่านั้น หากมีขนาดน้อยกว่าจำนวนที่กำหนดไว้ จะไม่มีการเพิ่มช่องว่างให้ครบตามจำนวนแต่อย่างใด ตัวอย่างของทั้ง 2 แบบเป็นไปดังตารางที่ 5

ตารางที่ 5 แสดงตัวอย่างการจัดเก็บข้อมูลประเภทไบนารี

ข้อมูลที่ต้องการจัดเก็บ	ข้อมูลที่ถูกจัดเก็บจริง binary(2)	ข้อมูลที่ถูกจัดเก็บจริง varbinary(2)
0x0	0x0000	0x00
0x1	0x1100	0x10
0x111	0x0111	0x0111

3.3 Integer เป็นประเภทข้อมูลที่ใช้เก็บเพียงค่าตัวเลข โดยแบ่งออกเป็น 3 ประเภท ตามขนาดการเก็บข้อมูล ดังแสดงในตารางที่ 6

ตารางที่ 6 แสดงประเภทของ Integer

ประเภท ขนาดข้อมูล(ไบต์)	จำนวนค่าทั้งหมด	ค่าที่นำมาใช้
Tinyint 1	$2^8 = 256$	0 ถึง 255
Smallint 2	-2^{15} ถึง $2^{15} - 1$	-32,766 ถึง 32,767
Int 4	-2^{31} ถึง $2^{31} - 1$	-2,147,483,648 ถึง 2,147,483,647

3.4 Approximate Numeric เป็นประเภทของข้อมูลที่เก็บค่าตัวเลขรวมถึงทศนิยม โดยมีการกำหนดจำนวนหลักของตัวเลขทั้งหมดไว้ก่อน ซึ่งแบ่งออกเป็น 2 ประเภทดังนี้

3.4.1 real หากกำหนดประเภทของข้อมูลเป็น real ระบบจะทราบทันทีว่าจำนวนตัวเลขทั้งหมดเป็น 7 หลัก และใช้ขนาดข้อมูลในการจัดเก็บ 4 ไบต์ สำหรับช่วงของค่าที่ยอมรับคือ $-3.40E + 38$ ถึง $3.40E + 38$

3.4.2 float มีช่วงของค่าที่ยอมรับได้คือ $-1.79E + 308$ ถึง $1.79E + 308$ โดย n คือจำนวนบิตที่ใช้เก็บค่า mantissa ซึ่งมีค่าอยู่ในช่วงระหว่าง 1 - 53 และมี

ความสัมพันธ์ตามตารางที่ 7 หากไม่ระบุค่า n จะได้ค่าดีฟอลต์สำหรับจำนวนหลัก เป็น 15

ตารางที่ 7 แสดงความสัมพันธ์ของประเภทข้อมูลแบบ float

ค่า n	จำนวนหลัก	ขนาดข้อมูล (ไบต์)
1 – 24	7	4
25 – 53	15	8

3.5 Date and Time เป็นประเภทข้อมูลที่เก็บค่าของวันที่ (Date) หรือเวลา (Time) หรือทั้งสองอย่างรวมกัน สำหรับการใส่ค่าข้อมูลของวันที่สามารถระบุได้หลายลักษณะ คือ

“Jun 28 1999” “28 Jun 1999” “1999 28 Jun” “28 Jun 1999”
 “Jun 28 99” “28 Jun 99” “99 28 Jun” “28 Jun 99”

หรือสามารถใช้เครื่องหมายคั่นช่วย ในกรณีที่การระบุเดือนกำหนดเป็นลำดับที่ของตัวเลขในปี เช่น

6/28/99 (m/d/y) 6-28-99 (m-d-y) 6.28.99 (m.d.y)

6/99/28 (m/y/d) 99-6-28 (y-m-d) 28.99.6 (d.y.m)

28/06/99 (mm/dd/yy) 06-28-1999 (mm-dd-yyyy)

ส่วนในลักษณะของเวลาก็เช่นเดียวกัน ตัวอย่างเช่น

8:30 8:30 PM 20:30

20:30:15:850 (20 ชั่วโมง 30 นาที 15 วินาที 850 มิลลิวินาที)

20.30.15.8 (20 ชั่วโมง 30 นาที 15 วินาที 8/10 วินาที)

3.6 Special มีประเภทของข้อมูลบางส่วนที่ไม่สามารถจัดกลุ่มได้ จึงนำมาไว้รวมกัน มีรายละเอียดดังนี้

3.6.1 bit เป็นการเก็บข้อมูลแบบบูลีน (Boolean) โดยค่าที่เป็นไปได้มีเพียง 2 ค่าเท่านั้น คือ จริง หรือ เท็จ ซึ่งอาจแทนค่าได้เป็น True หรือ False , Yes หรือ No , On หรือ Off เป็นต้น สำหรับการเก็บข้อมูลจริงของระบบจะเก็บเป็นค่าตัวเลข 0 หมายถึงเท็จ และ 1 หมายถึงจริง แต่ไม่ได้หมายความว่าฟิลด์ที่กำหนดประเภทของข้อมูลให้เป็น bit จะสามารถเก็บค่าได้เพียง 2 ค่านี้เท่านั้น เพราะเป็นไปได้ที่จะระบุให้ยอมรับค่าที่เป็น Null ได้อีกด้วย สำหรับขนาดข้อมูลที่ใช้จะถือรวมภายใน 1 ตาราง หากมีฟิลด์ประเภท

นี้น้อยกว่า หรือเท่ากับ 8 จะจัดเก็บข้อมูลด้วยขนาด 1 ไบต์ ถ้าอยู่ระหว่าง 9 – 16 ก็จัดเก็บเป็น 2 ไบต์ ไปเรื่อย ๆ เมื่อนับรวมครบถึง 8 bit นั้นเอง

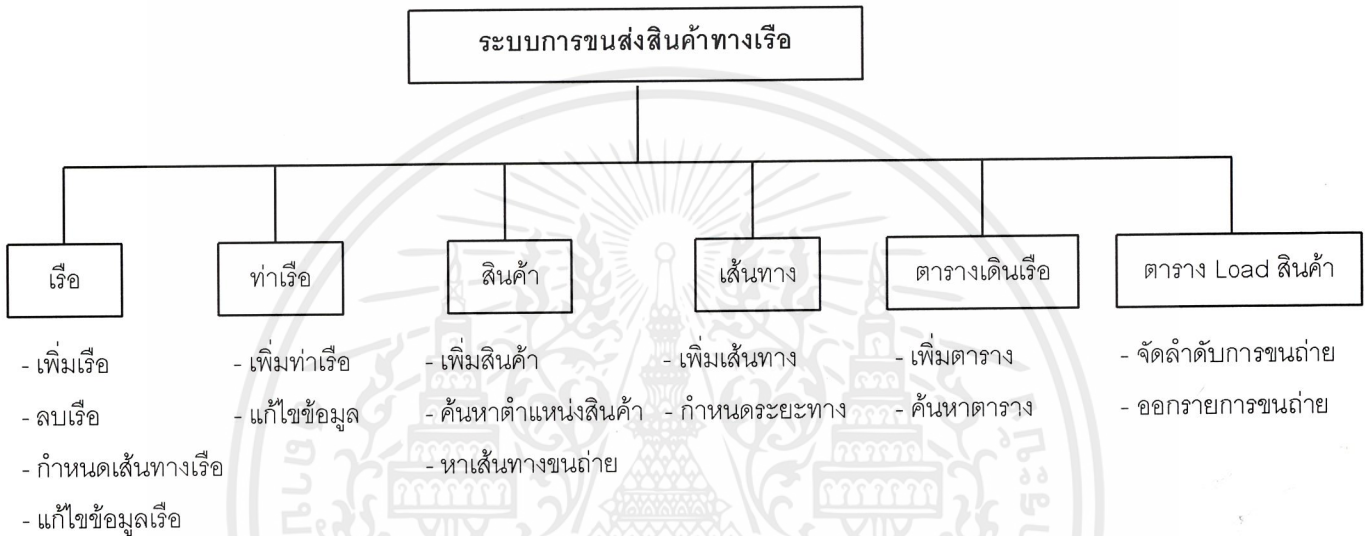
- 3.6.2 text เป็นประเภทของข้อมูลที่มีขึ้นเพื่อทำลายข้อจำกัดของข้อมูลให้สามารถเก็บอักขระได้มากกว่า 8,000 ตัว โดยสามารถเก็บได้สูงสุดถึง 2 GB เลยทีเดียว ข้อมูลเหล่านี้ไม่ได้ถูกจัดเก็บรวมอยู่ในตารางเหมือนประเภทของข้อมูลชนิดอื่นๆ แต่จะเก็บเป็นพอยเตอร์ ขนาด 16 ไบต์แทน โดยทำหน้าที่ชี้ไปยังข้อมูลซึ่งได้ถูกแบ่งออกเป็นแต่ละดาต้าเพจ (Data Page) ขนาด 8 KB อีกทีหนึ่ง
- 3.6.3 image เช่นเดียวกับกับ text แต่แทนที่จะเก็บข้อมูลที่เป็นตัวอักขระ เช่น ข้อความยาวๆ เป็นการเก็บข้อมูลไบนารีขนาดใหญ่ เช่น ไฟล์รูปภาพ เสียง เป็นต้น



บทที่ 3

วิธีการดำเนินงานวิจัยและการออกแบบระบบ

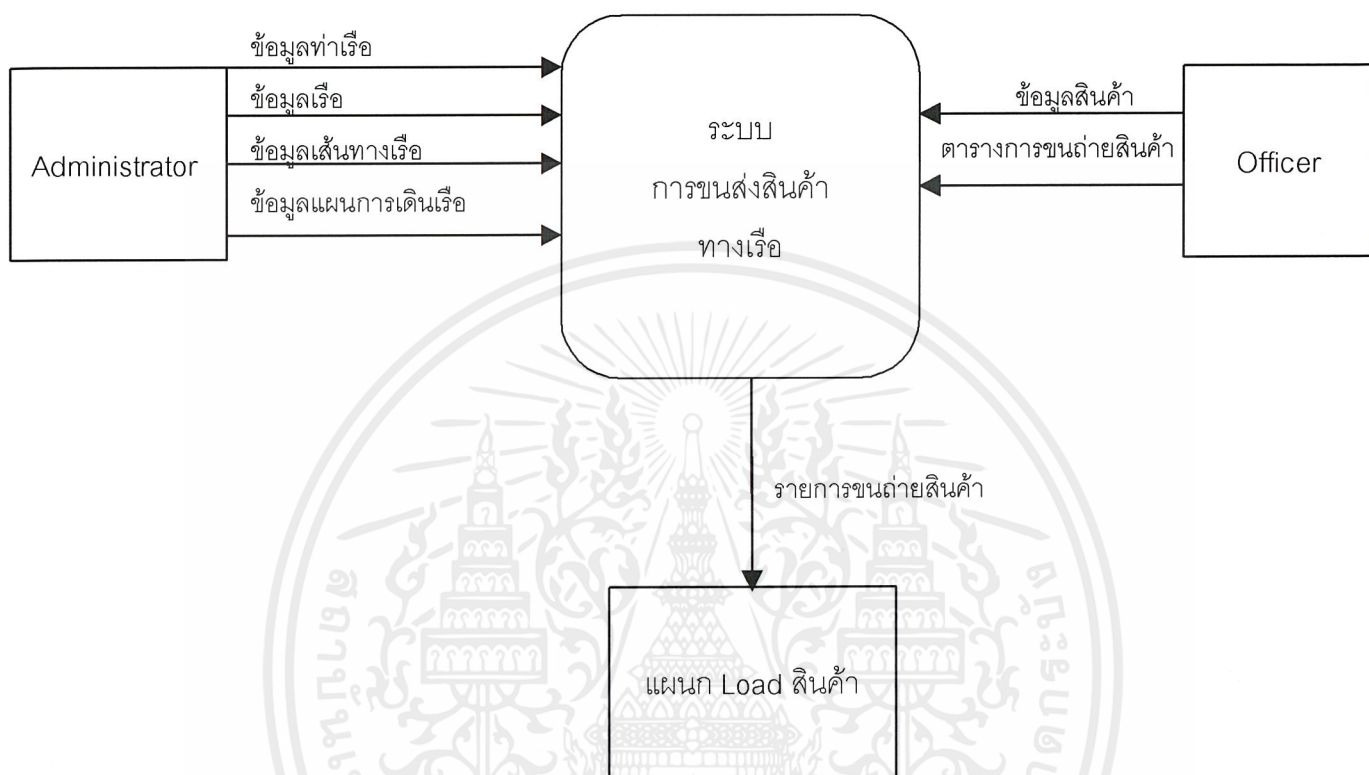
3.1 Functional Decomposition Diagram



รูปที่ 3.1 Functional Decomposition Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

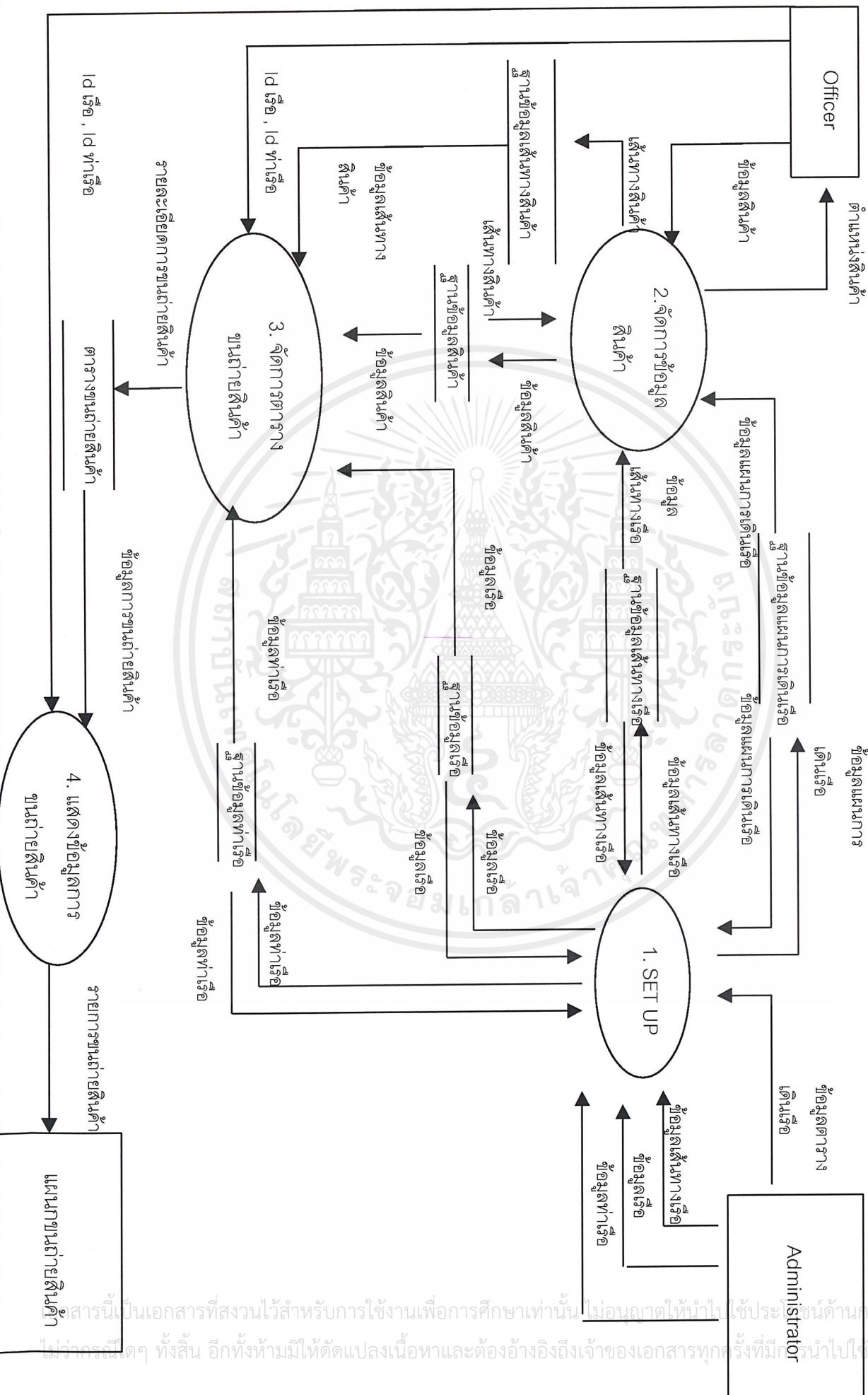
3.2 Context Diagram



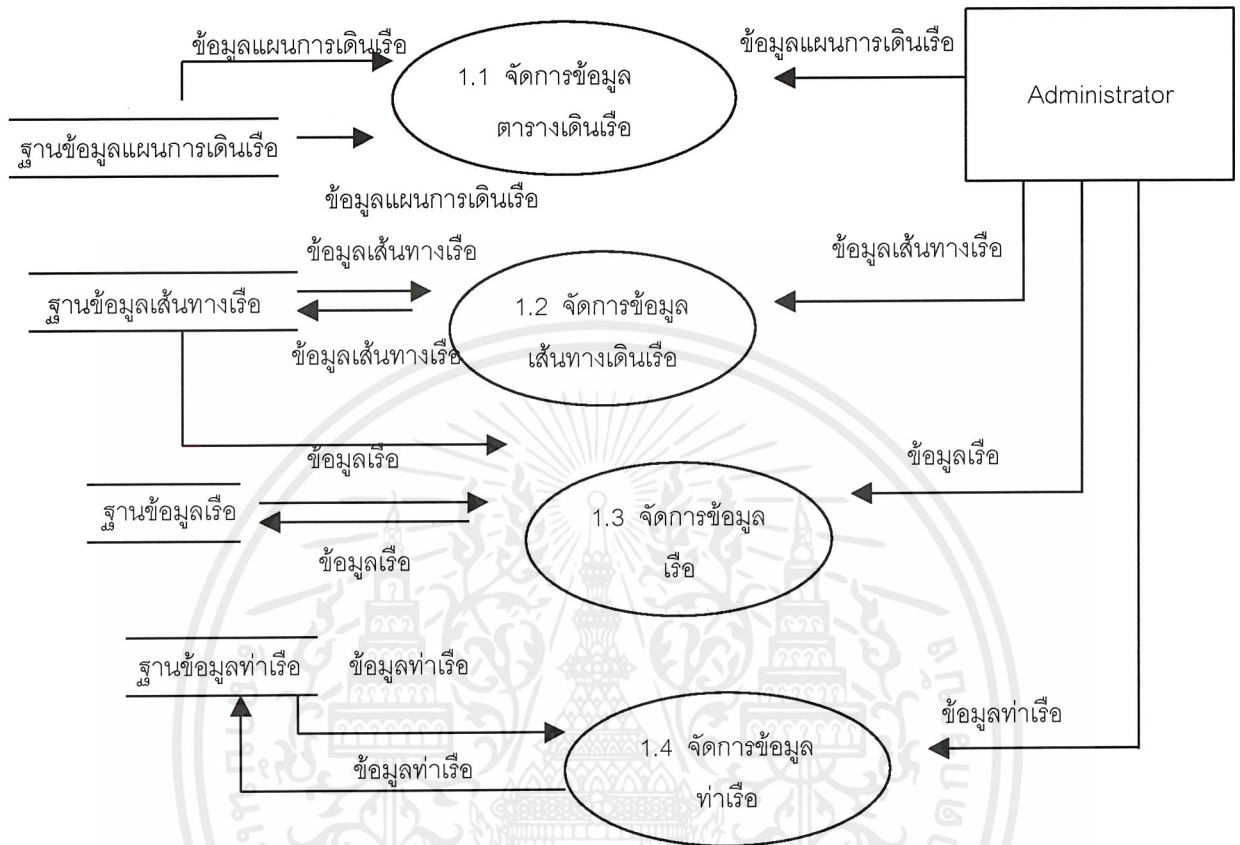
รูปที่ 3.2 Context Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 Data Flow Diagram ระดับ 0



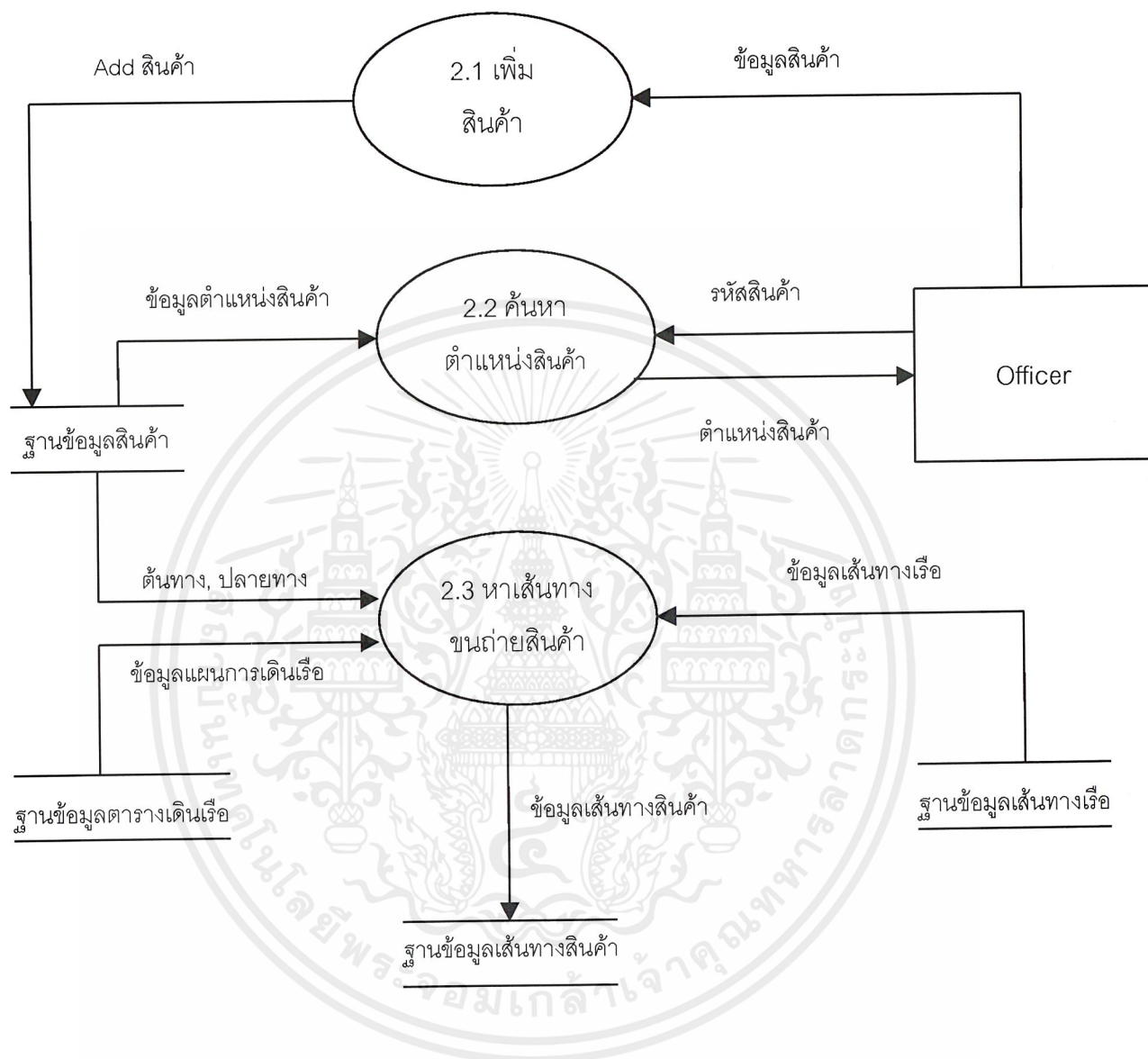
3.4 Data Flow Diagram ระดับ 1 ของกระบวนการงานที่1 SET UP



รูปที่ 3.4 Data Flow Diagram ระดับ 1 ของกระบวนการงานที่1 SET UP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

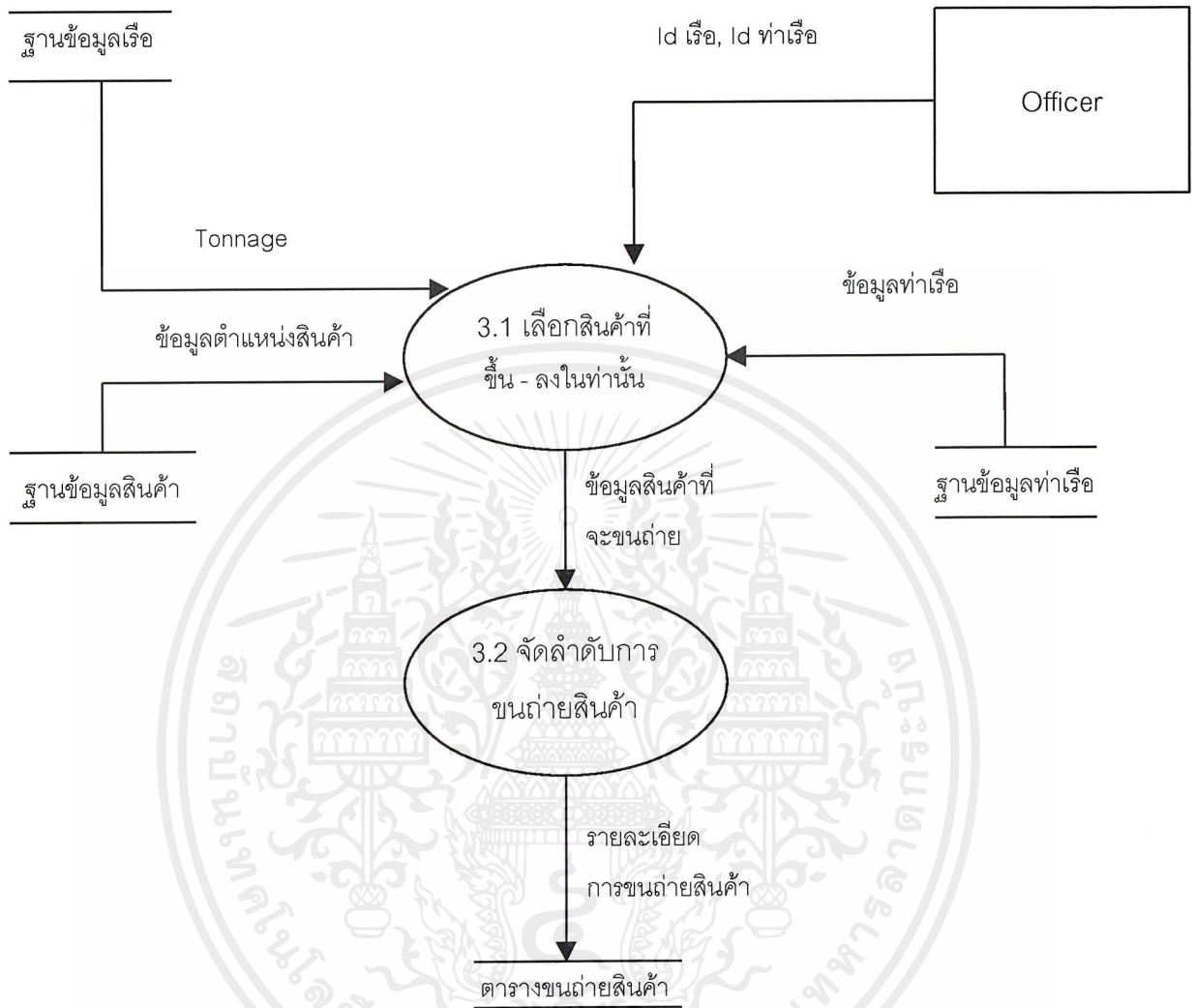
3.5 Data Flow Diagram ระดับ 1 ของกระบวนการงานที่ 2 จัดการข้อมูลสินค้า



รูปที่ 3.5 Data Flow Diagram ระดับ 1 ของกระบวนการงานที่ 2 จัดการข้อมูลสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

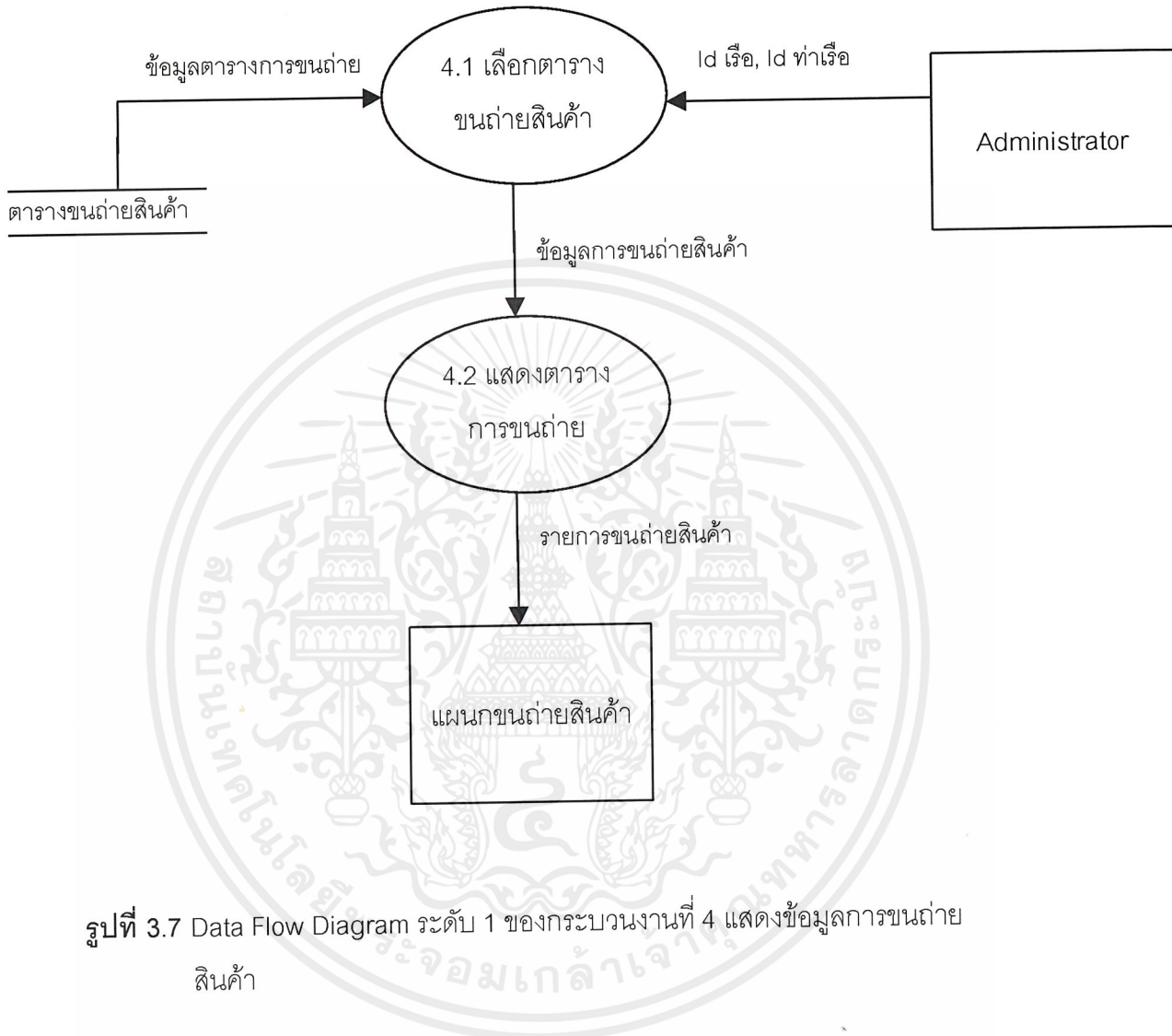
3.6 Data Flow Diagram ระดับ 1 ของกระบวนการงานที่ 3 จัดการตารางขนถ่ายสินค้า



รูปที่ 3.6 Data Flow Diagram ระดับ 1 ของกระบวนการงานที่ 3 จัดการตารางขนถ่ายสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

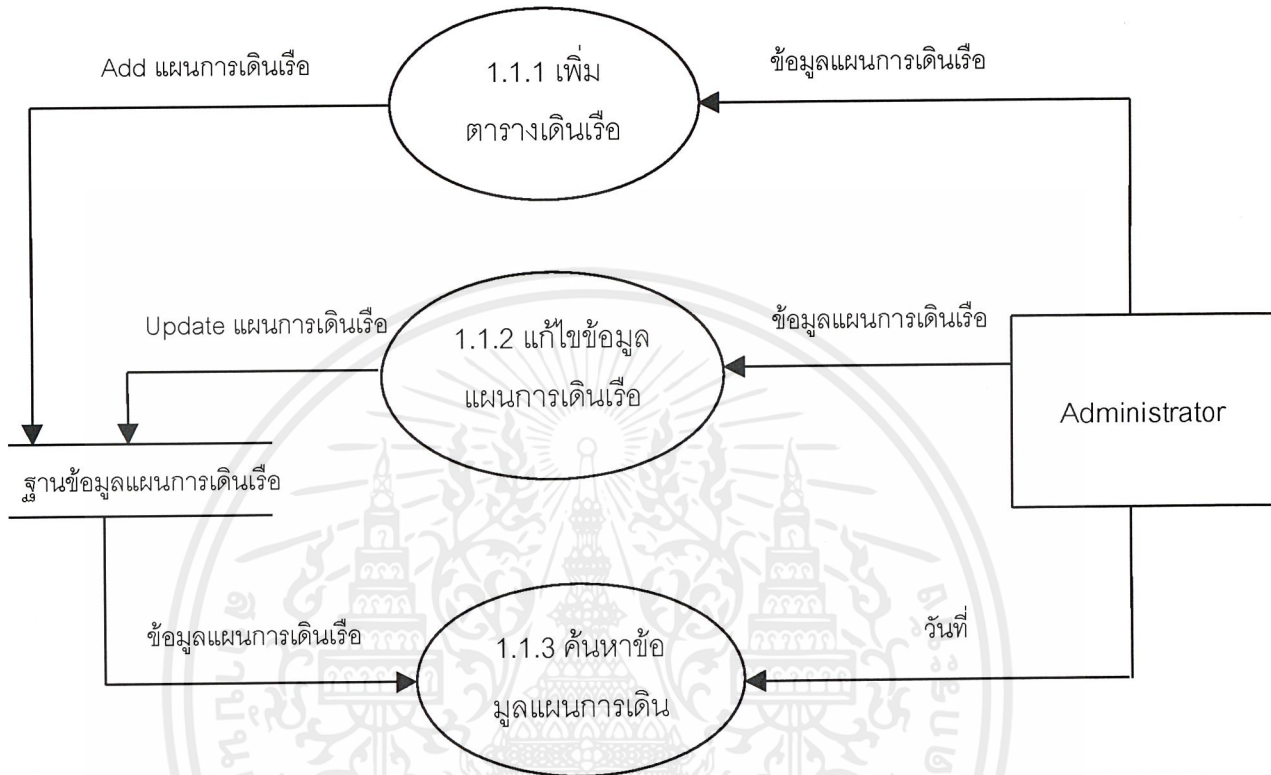
3.7 Data Flow Diagram ระดับ 1 ของกระบวนการงานที่ 4 แสดงข้อมูลการขนถ่ายสินค้า



รูปที่ 3.7 Data Flow Diagram ระดับ 1 ของกระบวนการงานที่ 4 แสดงข้อมูลการขนถ่ายสินค้า

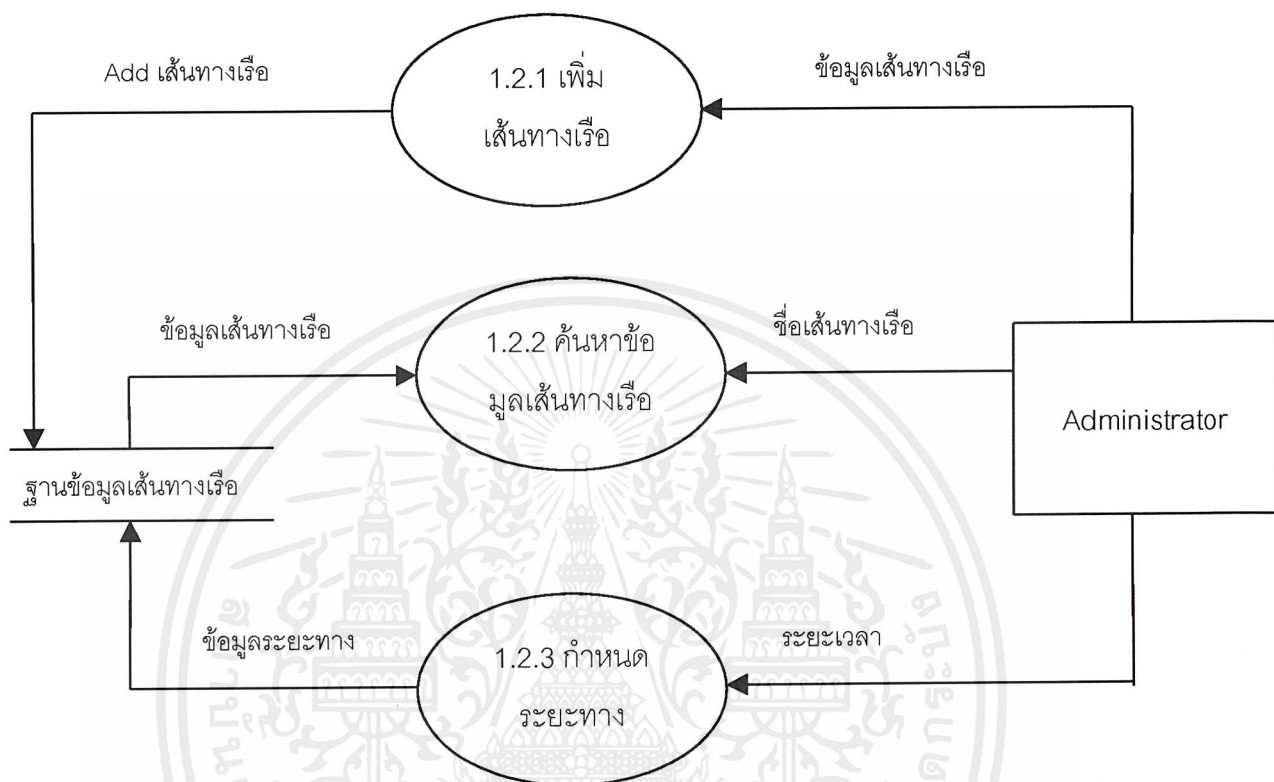
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 Data Flow Diagram ระดับ 2 ของกระบวนการงานที่ 1.1 จัดการข้อมูลแผนการเดินทาง



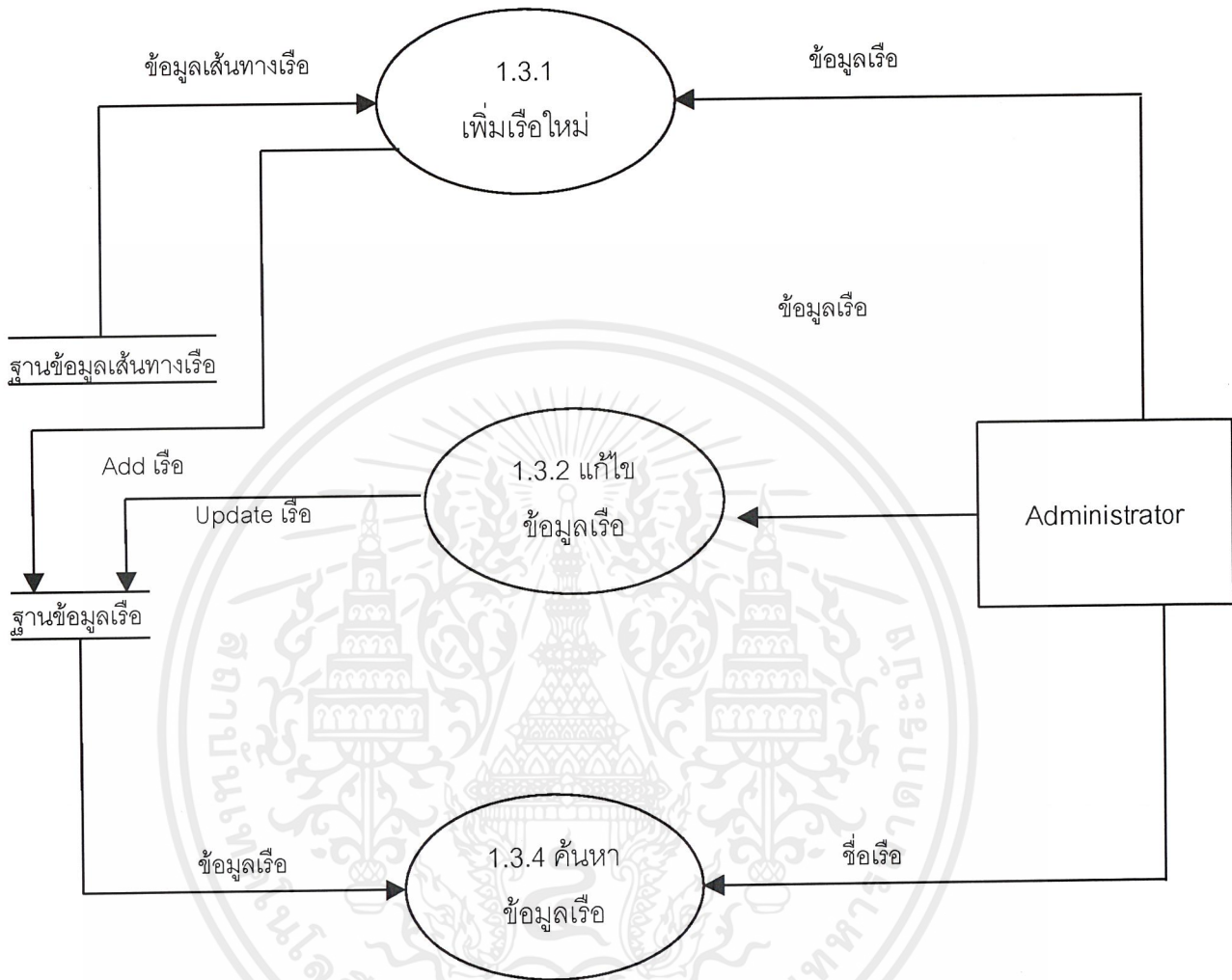
รูปที่ 3.8 Data Flow Diagram ระดับ 2 ของกระบวนการงานที่ 1.1 จัดการข้อมูลแผนการเดินทาง

3.9 Data Flow Diagram ระดับ 2 ของกระบวนการงานที่ 1.2 จัดการข้อมูลเส้นทาง เรือ



รูปที่ 3.9 Data Flow Diagram ระดับ 2 ของกระบวนการงานที่ 1.2 จัดการข้อมูลเส้นทาง
เรือ

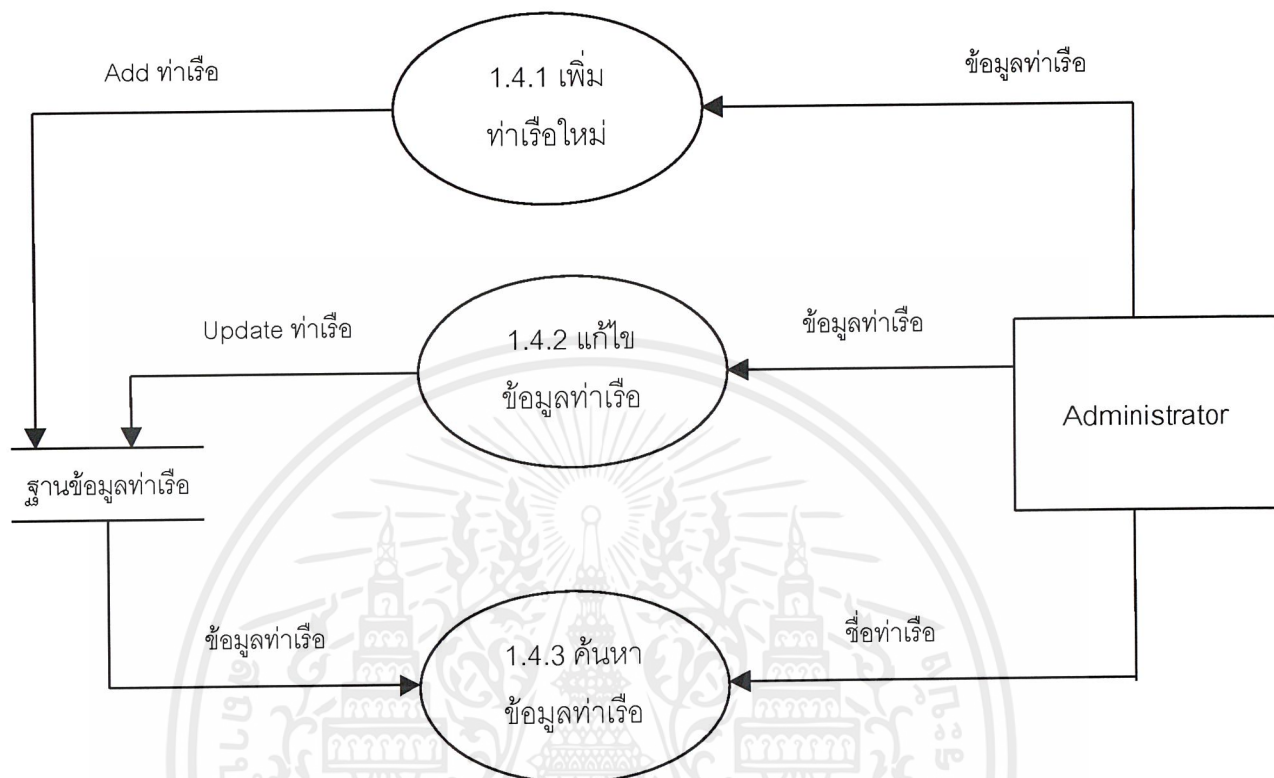
3.10 Data Flow Diagram ระดับ 2 ของกระบวนการงานที่ 1.3 จัดการข้อมูลเรือ



รูปที่ 3.10 Data Flow Diagram ระดับ 2 ของกระบวนการงานที่ 1.3 จัดการข้อมูลเรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.11 Data Flow Diagram ระดับ 2 ของกระบวนการงานที่ 1.4 จัดการข้อมูลท่าเรือ



รูปที่ 3.11 Data Flow Diagram ระดับ 2 ของกระบวนการงานที่ 1.4 จัดการข้อมูลท่าเรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.12 Pseudocode for Kruskal's algorithm

```

Void
Kruskal ( Graph G )
{
    int EdgesAccepted;
    DisjSet S;
    PriorityQueue H;
    Vertex U, V;
    SetType Uset, Vset;
    Edge E;

/* 1 */    Initialize ( S );
/* 2 */    ReadGraphIntoHeapArray ( G, H );
/* 3 */    BuildHeap( H );

/* 4 */    EdgesAccepted = 0;
/* 5 */    While ( EdgeAccepted < NumVertex - 1 )
    {
/* 6 */        E = DeleteMin ( H ); /* E = ( U , V ) */
/* 7 */        Uset = Find ( U , S );
/* 8 */        Vset = Find ( V , S );
/* 9 */        If ( Uset != Vset )
        {
            /* Accept the edge */

/* 10 */            EdgesAccepted++;
/* 11 */            SetUnion ( S , Uset , Vset );
        }
    }
}

```

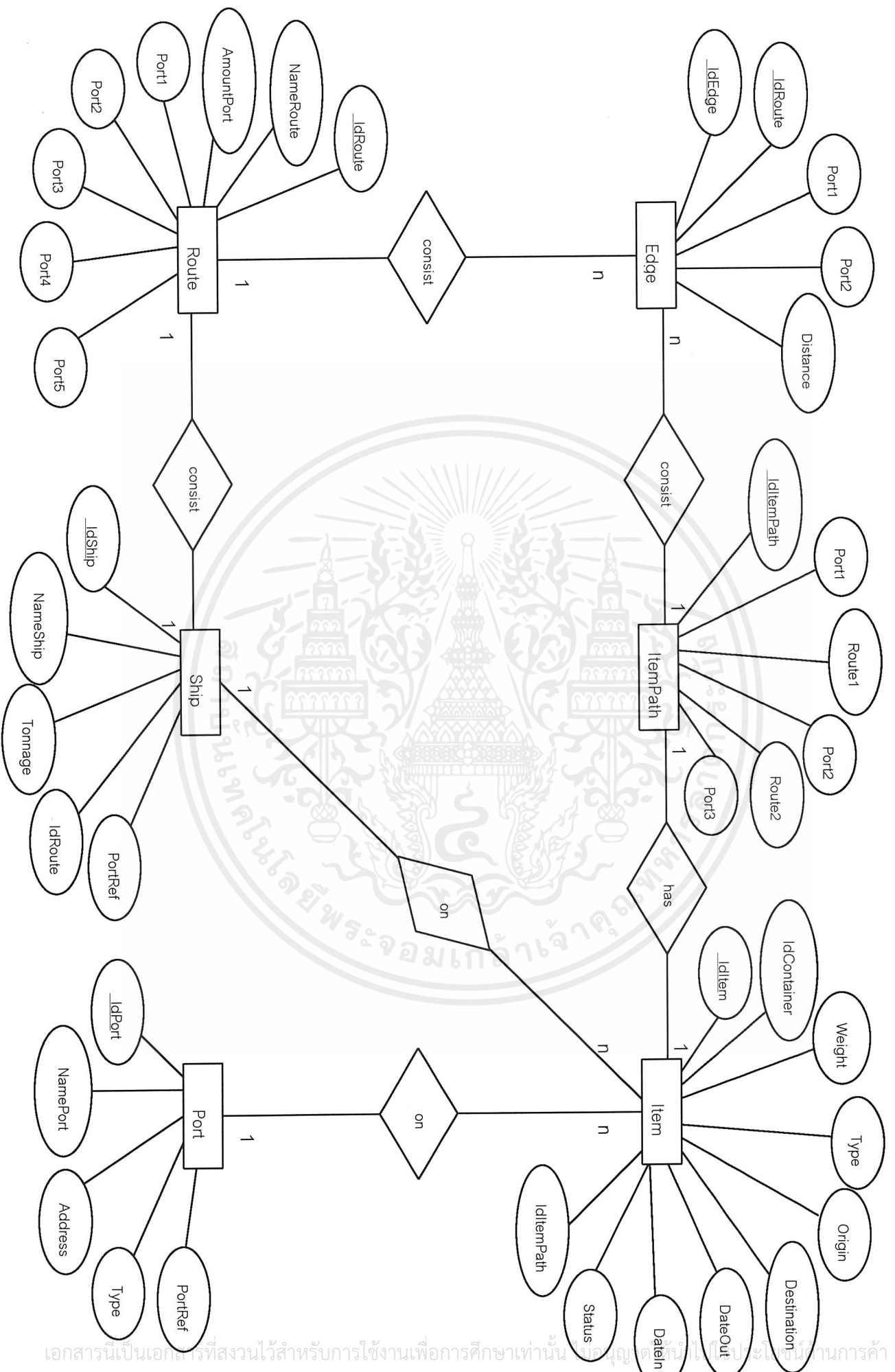
3.13 Pseudocode for depth – first search

```

Void
Dfs ( Vertex V )
{
    Visited [ V ] = True;
    For each W adjacent to V
        If ( ! Visited [ W ] )
            Dfs ( W );
}

```





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปเผยแพร่ในที่สาธารณะ
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.15 ตารางแสดงความสัมพันธ์ของข้อมูล

เป็นการแสดงความสัมพันธ์ของข้อมูลที่อยู่ภายในฐานข้อมูลในรูปแบบของตาราง ซึ่งประกอบด้วยตารางต่างๆดังต่อไปนี้

ตารางที่ 8 แสดงตารางข้อมูลเรือ

คำอธิบาย	ชื่อฟิลด์	ประเภท
รหัสเรือ	IdShip	Integer
ชื่อเรือ	NameShip	Text
ระวางขับน้ำ	Tonnage	Integer
จำนวนตู้ Container	ConLimit	Integer
จำนวนตู้ Refer	RefLimit	Integer
รหัสเส้นทางเรือ	IdRoute	Integer
รหัสท่าใหญ่	PortRef	Integer

ตารางที่ 9 แสดงตารางข้อมูลท่าเรือ

คำอธิบาย	ชื่อฟิลด์	ประเภท
รหัสท่าเรือ	IdPort	Integer
ชื่อท่าเรือ	NamePort	Text
ประเภท	Type	Char
ที่อยู่	Address	Text
ประเทศ	Country	Text
ทวีป	Continent	Text
เบอร์โทรศัพท์	Telephone	Text
IP Address	IPAddress	Text
รหัสท่าใหญ่	PortRef	Integer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 10 แสดงตารางข้อมูลสินค้า

คำอธิบาย	ชื่อฟิลด์	ประเภท
รหัสสินค้า	IdItem	Integer
รหัสตู้ Container	IdContainer	Integer
น้ำหนัก	Weight	Integer
ประเภท	Type	Char
ต้นทาง	Origin	Integer
ปลายทาง	Destination	Integer
วันที่รับสินค้า	DateIn	Date
วันที่ส่งสินค้า	DateOut	Date
ระยะเวลาประกัน	Warranty	Integer
สถานะการส่ง	Status	Integer
รหัสเส้นทางสินค้า	IdItemPath	Integer

ตารางที่ 11 แสดงตารางข้อมูลเส้นทางเดินเรือ

คำอธิบาย	ชื่อฟิลด์	ประเภท
รหัสเส้นทาง	IdRoute	Integer
ชื่อเส้นทาง	NameRoute	Text
จำนวนท่าเรือ	AmountPort	Integer
ท่าเรือที่ 1	Port1	Integer
ท่าเรือที่ 2	Port2	Integer
ท่าเรือที่ 3	Port3	Integer
ท่าเรือที่ 4	Port4	Integer
ท่าเรือที่ 5	Port5	Integer
ท่าเรือที่ 6	Port6	Integer
ท่าเรือที่ 7	Port7	Integer
ท่าเรือที่ 8	Port8	Integer
ท่าเรือที่ 9	Port9	Integer
ท่าเรือที่ 10	Port10	Integer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 12 แสดงตารางข้อมูลเส้นทางเดินเรือ

คำอธิบาย	ชื่อฟิลด์	ประเภท
รหัสเส้นทางสินค้า	IdItemPath	Integer
ท่าเรือที่ 1	Port1	Integer
เส้นทางเดินเรือที่ 1	Route1	Integer
ท่าเรือที่ 3	Port2	Integer
เส้นทางเดินเรือที่ 4	Route2	Integer
ท่าเรือที่ 5	Port3	Integer
เส้นทางเดินเรือที่ 6	Route3	Integer
ท่าเรือที่ 7	Port4	Integer
เส้นทางเดินเรือที่ 8	Route4	Integer
ท่าเรือที่ 9	Port5	Integer
เส้นทางเดินเรือที่ 10	Route5	Integer
ท่าเรือที่ 10	Port6	Integer

ตารางที่ 13 แสดงตารางข้อมูลสินค้าในท่าเรือ

คำอธิบาย	ชื่อฟิลด์	ประเภท
รหัสท่าเรือ	IdPort	Integer
รหัสสินค้า	IdItem	Integer
รหัสเรือที่รอ	WaitFor	Integer
ระดับการรอเรือ	Priority	Integer

ตารางที่ 14 แสดงตารางข้อมูลสินค้าบนเรือ

คำอธิบาย	ชื่อฟิลด์	ประเภท
รหัสเรือ	<u>IdShip</u>	Integer
รหัสสินค้า	<u>IdItem</u>	Integer
รหัสท่าเรือที่ขึ้นมา	FromPort	Integer
รหัสท่าเรือที่จะลง	ToPort	Integer
ระดับของระยะทาง	Priority	Integer

ตารางที่ 15 แสดงตารางข้อมูลระยะเวลาระหว่างท่าเรือ

คำอธิบาย	ชื่อฟิลด์	ประเภท
รหัส	<u>IdEdge</u>	Integer
รหัสเส้นทางเดินเรือ	<u>IdRoute</u>	Integer
ท่าเรือที่ 1	Port1	Integer
ท่าเรือที่ 2	Port2	Integer
ระยะเวลา	Distance	Integer

ตารางที่ 16 แสดงตารางข้อมูลแผนการเดินเรือ

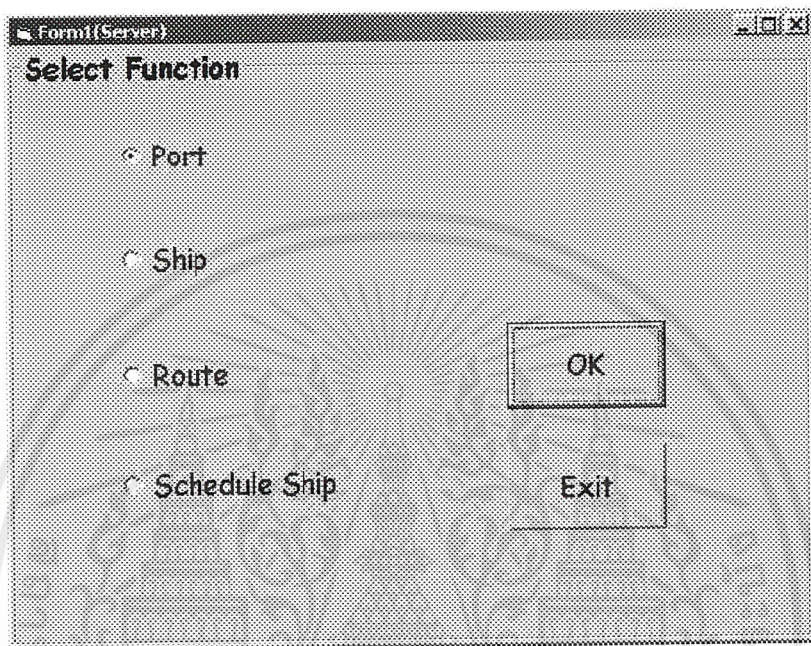
คำอธิบาย	ชื่อฟิลด์	ประเภท
รหัส	<u>IdSchedule</u>	Integer
รหัสเรือ	<u>IdShip</u>	Integer
รหัสท่าเรือ	<u>IdPort</u>	Integer
วันที่เรือออก	DateOut	Date

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

โปรแกรมส่วนที่ติดต่อกับผู้ใช้งาน(User Interface)

4.1 หน้าจอหลักโปรแกรม (สำหรับเครื่อง Server)



รูปที่ 4.1 แสดงหน้าจอหลักของโปรแกรม (สำหรับเครื่อง Server)

ในการทำงานของโปรแกรมระบบการขนถ่ายสินค้าทางเรือ มีหน้าจอการทำงานหลักสำหรับให้ผู้ใช้ เลือกการใช้งานส่วนต่างๆ ของโปรแกรม ประกอบด้วยหน้าจอเรือ (Ship), เส้นทางเดินเรือ (Route), ท่าเรือ (Port) และแผนกำหนดการเดินทางเรือ (Schedule Ship) ซึ่งรายละเอียดในการทำงานสามารถอธิบายได้ดังนี้

4.2 การเพิ่ม และแก้ไขข้อมูลท่าเรือ

4.2.1 การเพิ่ม และแก้ไขข้อมูลท่าเรือสำหรับท่าเรือ Master

The screenshot shows a software window titled "Form7(port)" with a menu bar containing "Menu" and "File". The main area is divided into two sections. On the left, there is a "Select Port" list box containing the following items: Yokohama, Port Kelang, Lamchabang, Qingdao, Hong Kong (highlighted), Yantian, Penang, Jakarta, Surabaya, Belawan, Ningbo, and Fuqing. Below the list box is a "Select Port Reference" button. On the right, there is a "Type" section with two radio buttons: "Big" (selected) and "Small". Below this are several input fields: "Port Id" with the value "1288", "Port Name" with "Hong Kong", "Address" with "235 Nathan Road, Kooloon", "Country" with "Hong Kong", "Continent" with "asia", "Phone No." with "4215322", "IP Address" with "202.14.0.122", and "Port Reference" which is currently empty.

รูปที่ 4.2.1 การเพิ่ม และแก้ไขข้อมูลท่าเรือสำหรับท่าเรือ Master

เมื่อผู้ใช้งานต้องการเพิ่มข้อมูลท่าเรือ Master ผู้ใช้จะต้องเลือก New จากเมนู File > New แล้วป้อนรายละเอียดของข้อมูลเรือต่างๆให้ครบดังนี้

- PortName ชื่อท่าเรือ
- Address ที่ตั้งของท่าเรือ
- Country ประเทศ
- Continent ทวีป
- Phone No. เบอร์โทรศัพท์
- IP Address
- Type ประเภทของท่าเรือ

สำหรับ Port Id โปรแกรมจะทำการ Generate ให้เองโดยอัตโนมัติ แล้วเลือก Save จากเมนู File > Save โปรแกรมจะทำการเพิ่มข้อมูลท่าเรือลงฐานข้อมูล

เมื่อผู้ใช้งานต้องการแก้ไขข้อมูลท่าเรือต้องเลือกท่าเรือจาก List Box แล้วทำการแก้ไข จากนั้นเลือก Save จากเมนู File > Save โปรแกรมจะทำการแก้ไขข้อมูลต่างๆลงบนฐานข้อมูลท่าเรือ

4.2.2 การเพิ่ม และแก้ไขข้อมูลท่าเรือสำหรับท่าเรือ Feeder

The screenshot shows a software window titled 'Furn7(port)'. It contains a 'Select Part' list box on the left with the following items: Yokohama, Port Kelang, Lamchabang (highlighted), Qingdao, Hong Kong, Yantian, Penang, Jakarta, Surabaya, Belawan, Ningbo, and Fuqing. Below the list is a 'Select Port Reference' dropdown menu showing '(Please Select)'. To the right of the list box is a 'Type' section with radio buttons for 'Big' and 'Small', where 'Small' is selected. Further right are several input fields: 'Port Id' (3883), 'Part Name' (Lamchabang), 'Address' (45/2 lam chabang district. c), 'Country' (Thailand), 'Continent' (asia), 'Phone No' (038-546894), 'IP Address' (202.14.0.8), and 'Port Reference' (Port Kelang).

รูปที่ 4.2.2 การเพิ่ม และแก้ไขข้อมูลท่าเรือสำหรับท่าเรือ Feeder

เมื่อผู้ใช้ต้องการเพิ่มข้อมูลท่าเรือเอง ผู้ใช้จะต้องเลือก New จากเมนู File > New แล้วป้อนรายละเอียดของข้อมูลเรือต่างๆให้ครบดังนี้

- PortName ชื่อท่าเรือ
- Address ที่ตั้งของท่าเรือ
- Country ประเทศ
- Continent ทวีป
- Phone No. เบอร์โทรศัพท์
- IP Address
- Port Reference ท่าเรือ Master สำหรับท่าเรือ Feeder
- Type ประเภทของท่าเรือ

สำหรับ Port Id โปรแกรมจะทำการ Generate ให้เองโดยอัตโนมัติ แล้วเลือก Save จากเมนู File > Save โปรแกรมจะทำการเพิ่มข้อมูลท่าเรือลงฐานข้อมูล

เมื่อผู้ใช้ต้องการแก้ไขข้อมูลท่าเรือต้องเลือกท่าเรือจาก List Box แล้วทำการแก้ไข จากนั้นเลือก Save จากเมนู File > Save โปรแกรมจะทำการแก้ไขข้อมูลต่างๆลงบนฐานข้อมูลท่าเรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การเพิ่ม และแก้ไขข้อมูลเรือ

4.3.1 การเพิ่ม และแก้ไขข้อมูลเรือสำหรับเรือประเภท Master

รูปที่ 4.3.1 การเพิ่ม และแก้ไขข้อมูลเรือสำหรับเรือประเภท Master

เมื่อผู้ใช้งานต้องการเพิ่มข้อมูลเรือประเภท Master ผู้ใช้ต้องเลือก New จากเมนู File > New แล้วป้อนรายละเอียดของข้อมูลเรือต่างๆให้ครบดังนี้

- NameShip ชื่อเรือ
- Tonnage ระวางน้ำหนักเรือ
- Container Limited จำนวนจำกัดของตู้ Container ประเภทธรรมดา
- Refer Limited จำนวนจำกัดของตู้ Container ประเภท Refer
- Type ชนิดของเรือ

สำหรับ Id Ship โปรแกรมจะทำการ Generate ให้เองโดยอัตโนมัติ แล้วเลือก Save จากเมนู File > Save โปรแกรมจะทำการเพิ่มข้อมูลเรือลงฐานข้อมูล และโปรแกรมจะเปลี่ยนหน้าจอไปยังหน้าจอเส้นทางเดินเรือ (Route) สำหรับกำหนดเส้นทางเดินเรือให้เรือ

เมื่อผู้ใช้งานต้องการแก้ไขข้อมูลเรือ ผู้ใช้จะต้องเลือกเรือจาก List Box แล้วทำการแก้ไข จากนั้นเลือก Save จากเมนู File > Save โปรแกรมจะทำการแก้ไขข้อมูลเรือต่างๆลงบนฐานข้อมูลเรือ

4.3.2 การเพิ่ม และแก้ไขข้อมูลเรือสำหรับเรือประเภท Feeder

รูปที่ 4.3.2 การเพิ่ม และแก้ไขข้อมูลเรือสำหรับเรือประเภท Feeder

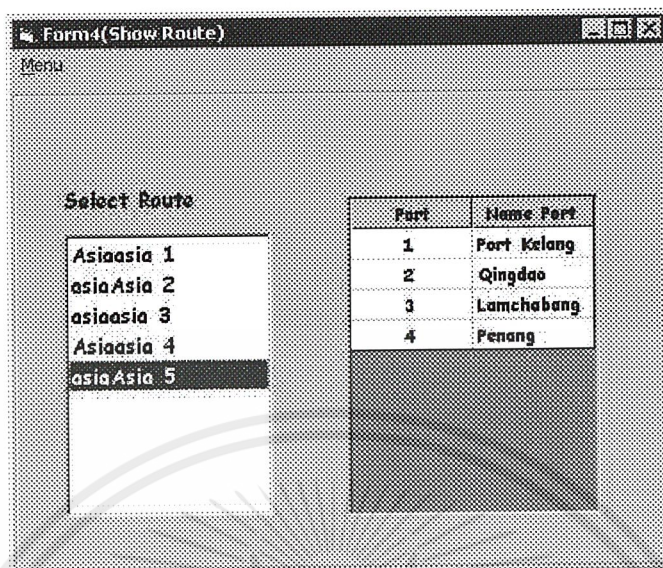
เมื่อผู้ใช้ต้องการเพิ่มข้อมูลเรือประเภท Feeder ผู้ใช้จะต้องเลือก New จากเมนู File > New แล้วป้อนรายละเอียดของข้อมูลเรือต่างๆให้ครบดังนี้

- NameShip ชื่อเรือ
- Tonnage ระวางน้ำหนักเรือ
- Container Limited จำนวนจำกัดของตู้ Container ประเภทธรรมดา
- Refer Limited จำนวนจำกัดของตู้ Container ประเภท Refer
- Port Reference ท่าเรือ Master สำหรับเรือ Feeder
- Type ชนิดของเรือ

สำหรับ Id Ship โปรแกรมจะทำการ Generate ให้เองโดยอัตโนมัติ แล้วเลือก Save จากเมนู File > Save โปรแกรมจะทำการเพิ่มข้อมูลเรือลงฐานข้อมูล และโปรแกรมจะเปลี่ยนหน้าจอไปยังหน้าจอเส้นทางเดินเรือ (Route) สำหรับกำหนดเส้นทางเดินเรือให้เรือ

เมื่อผู้ใช้ต้องการแก้ไขข้อมูลเรือ ผู้ใช้จะต้องเลือกเรือจาก List Box แล้วทำการแก้ไข จากนั้นเลือก Save จากเมนู File > Save โปรแกรมจะทำการแก้ไขข้อมูลเรือต่างๆลงบนฐานข้อมูลเรือ

4.4 แสดงเส้นทางเดินเรือ และเลือกเส้นทางเดินเรือสำหรับเรือต่างๆ



รูปที่ 4.4 หน้าจอแสดงเส้นทางเดินเรือ และเลือกเส้นทางเดินเรือสำหรับเรือต่างๆ

การเข้าสู่หน้าจอนี้จะกระทำได้ 2 ทางคือ

1. เมื่อผู้ใช้ต้องการดูเส้นทางเดินเรือต่างๆ จะทำได้โดยเลือก Route จากหน้าจอหลักของโปรแกรม โดยจะปรากฏชื่อเส้นทางเดินเรือที่ List Box ทางซ้าย เมื่อผู้ใช้ทำการเลือกชื่อเส้นทางเดินเรือที่ List Box โปรแกรมจะแสดงท่าเรือทั้งหมดที่ประจำเส้นทางเดินเรื่อนั้นๆ ขึ้นมา

2. เมื่อผู้ใช้ทำการเพิ่มข้อมูลเรือแล้ว โปรแกรมจะปรากฏหน้าจอนี้ขึ้นมาเพื่อให้ผู้ใช้กำหนดเส้นทางเดินเรือสำหรับเรื่อนั้นๆ ซึ่งทำได้ 2 วิธีคือ

2.1 โดยจะเลือกจากเส้นทางที่มีอยู่แล้วใน List Box แล้วเลือก Save จากเมนู File > Save

2.2 เลือกเพิ่มเส้นทางใหม่ โดยเลือก New Route จากเมนู File > New Route โปรแกรมจะทำการเปลี่ยนไปยังหน้าจอถัดไป (หน้าจอ 4.5 New Route)

4.5 การเพิ่มข้อมูลเส้นทางเดินเรือใหม่

รูปที่ 4.5 การเพิ่มข้อมูลเส้นทางเดินเรือใหม่

จากหน้าจอแสดงเส้นทางเดินเรือ (4.4 Show Route) เมื่อผู้ใช้ต้องการป้อนข้อมูลเส้นทางเดินเรือใหม่ให้กับเรือ จะปรากฏหน้าจอ New Route นี้ขึ้นมาเพื่อเพิ่มเส้นทางเดินเรือ โดยผู้ใช้จะต้องป้อนจำนวนท่าเรือทั้งหมดที่ต้องการให้อยู่บนเส้นทางเดินเรือที่ต้องการ ส่วน Id Route โปรแกรมจะทำการ Generate ขึ้นมาให้อัตโนมัติ แล้วกดปุ่ม Ok จากนั้นโปรแกรมจะแสดงท่าเรือทั้งหมดขึ้นมาที่ Combo Box ให้ผู้ใช้ทำการเลือกท่าเรือที่ต้องการตามจำนวนท่าที่ผู้ใช้ได้ป้อนไว้ จากนั้นทำการเพิ่มข้อมูลเส้นทางเดินเรือโดยการเลือก Save จากเมนู File > Save โปรแกรมจะทำการเพิ่มข้อมูลเส้นทางเดินเรือลงบนฐานข้อมูล

4.6 การกำหนดแผนกำหนดการเดินทางเรือ

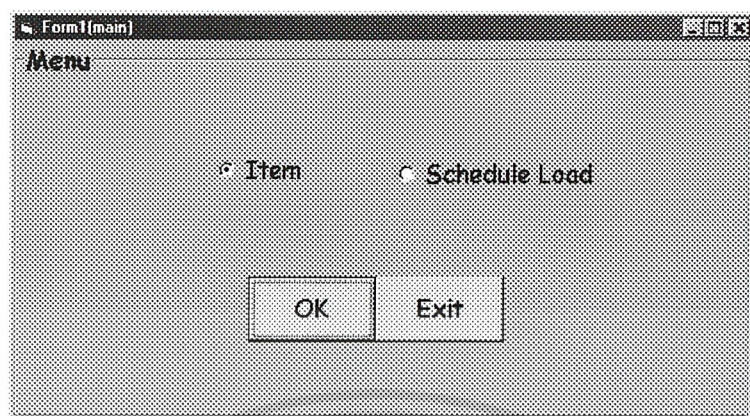
The screenshot shows a software window titled "Form6(Schedule)". It contains a form with the following fields and controls:

- Menu Type**: A label at the top left.
- Arrival Date**: A section header.
- Select Ship**: A dropdown menu showing "MarineS900" and an "OK" button.
- Select Port**: A dropdown menu showing "Leam Chabang" and an "OK" button.
- Date**: Three dropdown menus showing "2", "Mar", and "2002".
- Jakarta**: A text box containing "20/02/2002".
- Yantian**: A text box containing "23/02/2002".
- Qingdao**: A text box containing "25/02/2002".
- Save**: A button at the bottom center.

รูปที่ 4.6 การกำหนดแผนกำหนดการเดินทางเรือ

เมื่อผู้ใช้ต้องการกำหนดแผนกำหนดการเดินทางเรือ ผู้ใช้จะต้องทำการเลือกเรือ แล้วกดปุ่ม Ok เพื่อทำการเลือกท่าเรือ และวันที่ที่เรือจะเข้าเทียบท่านั้นๆ จากนั้นผู้ใช้ต้องกดปุ่ม Save เพื่อเพิ่มและเปลี่ยนแปลงแผนกำหนดการเดินทางเรือ ซึ่งถือว่าการออกแผนกำหนดการเดินทางเรือใหม่

4.7 หน้าจอหลักโปรแกรม (สำหรับเครื่อง Client)



รูปที่ 4.7 แสดงหน้าจอหลักของโปรแกรม (สำหรับเครื่อง Client)

ในการทำงานของโปรแกรมระบบการขนถ่ายสินค้าทางเรือ มีหน้าจอการทำงานหลัก สำหรับให้ผู้ใช้ เลือกการใช้งานส่วนต่างๆ ของโปรแกรม ประกอบด้วยหน้าจอสินค้า (Item) และ กำหนดการขนถ่ายสินค้า (Schedule Load) ซึ่งรายละเอียดในการทำงานสามารถอธิบายได้ดังนี้

4.8 สินค้า (Item)

สำหรับการจัดการข้อมูลของสินค้านี้

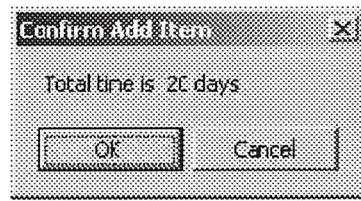
4.8.1 การเพิ่มข้อมูลสินค้า

รูปที่ 4.8.1 แสดงหน้าจอการเพิ่มข้อมูลสินค้า

เมื่อผู้ใช้ต้องการเพิ่มข้อมูลสินค้า (Container) ผู้ใช้จะต้องป้อนข้อมูลของสินค้าให้ครบ

- Weight น้ำหนักของตู้ Container
- Original ท่าเรือต้นทาง
- Destination ท่าเรือปลายทาง
- Cost ในการขนส่งสินค้าซึ่งมี 2 ประเภทให้เลือก คือ
 - Economic ค่าใช้จ่ายในการขนส่งถูก แต่ระยะเวลาการขนส่งนาน
 - Executive ค่าใช้จ่ายในการขนส่งแพงกว่า แต่ระยะเวลาการขนส่งจะเร็วกว่า
- Type ประเภทตู้ Container

ส่วนรหัสสินค้า (Container Id)โปรแกรมจะทำการ generate ให้โดยอัตโนมัติ แล้วกดปุ่ม Ok เพื่อทำการเพิ่มรายการสินค้าเข้าไป จากนั้นโปรแกรมจะกระทำการประมวลผลหาเส้นทางของสินค้าและจะแสดงระยะเวลาโดยประมาณในการขนส่งสินค้าขึ้นมามีดังรูปที่ 4.8.1.1



รูปที่ 4.8.2 แสดงระยะเวลาโดยประมาณในการขนส่งสินค้า

เมื่อผู้ใช้กดปุ่ม Ok โปรแกรมจะทำการนำข้อมูลสินค้าไปเก็บไว้ในฐานข้อมูลสินค้า และถ้าไม่ต้องการเลือกเส้นทางดังกล่าว หรือต้องการจะยกเลิกการเพิ่มข้อมูลสินค้าก็สามารถกระทำได้โดยกดปุ่ม Cancel



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8.2 การค้นหาสินค้า

Form2[Item]	
Menu	
Order	Search
Schedule	
Container Id	1100801
	OK
Port	>> Hong Kong
Route	1002
Port	Singapore
Route	1007
Port	Colombo
Status	Port Hong Kong

รูปที่ 4.8.3 แสดงหน้าจอการค้นหาสินค้า

ผู้ใช้สามารถค้นหาสินค้า (Container) ได้โดยการป้อนรหัสตู้ Container ลงในช่อง Container Id. แล้วกดปุ่ม Ok เพื่อทำการค้นหา โปรแกรมจะทำการค้นหาสินค้าแล้วทำการแสดงเส้นทางของสินค้า และตำแหน่งของสินค้า โดยถ้าสินค้าอยู่บนท่าเรือโปรแกรมจะแสดงชื่อท่าเรือที่สินค้านั้นอยู่

4.8.3 การค้นหาแผนกำหนดการเดินทางเรือ

NameShip	Date
ma01	12/03/2002
ma03	15/03/2002
ma11	20/03/2002
fo101	18/03/2002
ma12	24/03/2002

รูปที่ 4.8.4 แสดงการค้นหาแผนกำหนดการเดินทางเรือสำหรับแต่ละท่าเรือ

เมื่อผู้ใช้งานต้องการค้นหาแผนกำหนดการเดินทางเรือสำหรับแต่ละท่าเรือ ผู้ใช้ต้องเลือกท่าเรือที่ต้องการค้นหาแผนกำหนดการเดินทางเรือจาก Combo box และเลือกเดือนที่ต้องการจะค้นหา จากนั้นกดปุ่ม Ok โปรแกรมจะทำการค้นหาแผนกำหนดการเดินทางเรือและแสดงผลเป็นรายชื่อเรือทั้งหมดสำหรับท่าเรือนั้นๆ รวมทั้งวันที่ที่เรือจะเดินทางภายในเดือนนั้น

4.9 กำหนดการขนถ่ายสินค้า

The screenshot shows a software interface for creating a load table. It includes a 'Port' dropdown menu with 'Hong Kong' selected, a 'Ship' dropdown menu with 'ma02' selected, and a 'Process' button. Below these are two tables: 'Ship to Port' and 'Port to Ship'.

Ship to Port		Port to Ship	
Container Id	Status	Container Id	Priority
1120801	wait	1060402	2
1060801	wait	2090501	1
1120201	completed	1100703	3
1080203	completed	2020501	2
1091004	wait	2091202	3
2061002	wait	1010804	2
		2020802	1
		1120602	2

รูปที่ 4.9 การแสดงกำหนดการขนถ่ายสินค้า

เมื่อผู้ใช้ต้องการออกกำหนดการขนถ่ายสินค้า ผู้ใช้จะต้องทำการเลือกท่าเรือที่ต้องการออกตารางการขนถ่ายสินค้า และเลือกเรือที่จะเทียบท่า แล้วกดปุ่ม Process โปรแกรมจะทำการประมวลผลแล้วแสดงตารางการขนถ่ายสินค้าสำหรับเรือที่จะเทียบท่า โดยสำหรับสินค้าที่จะทำการขนถ่ายลงจากเรือเรือจะแสดงเป็นรหัสของตู้ Container (Container Id) และ Status ซึ่ง Complete คือสินค้านั้นถึงท่าเรือปลายทางเรียบร้อยแล้ว ส่วน Wait คือสินค้านั้นจะต้องรอเรือลำต่อไปเพื่อไปยังท่าเรือปลายทาง และสำหรับสินค้าที่จะทำการขนถ่ายขึ้นเรือจะแสดงเป็นรหัสของตู้ Container (Container Id) และ Priority เพื่อให้พนักงานขนถ่ายสินค้าสามารถจัดลำดับการวางของ Container ได้อย่างสะดวก

บทที่ 5

สรุป วิจารณ์ และแนวทางในการพัฒนา

5.1 ความสามารถของโปรแกรม

- 5.1.1 หาเส้นทางในการต่อเรือของสินค้าได้
- 5.1.2 จัดการการขนส่งโดยเป็นไปอย่างมีลำดับ
- 5.1.3 หาดตารางการขนถ่ายสินค้าในแต่ละท่าเรือได้
- 5.1.3 จัดลำดับการขนถ่ายสินค้าขึ้น - ลงเรือ

5.2 ข้อจำกัดของโปรแกรม

- 5.2.1 ก่อนเริ่มระบบจะต้องใส่ข้อมูลเรือ , ท่าเรือ และเส้นทางเดินเรือก่อน
- 5.2.2 ไม่สามารถเปลี่ยนแปลงเส้นทางเดินเรือได้
- 5.2.3 ท่าเรือไม่สามารถลบได้ แต่ถ้าไม่ใช้ท่าเรือนั้นแล้วสามารถทำได้โดยสร้างเส้นทางเดินเรือใหม่ไม่ให้ผ่านท่าเรือนั้น
- 5.2.4 เส้นทางเดินเรือหนึ่งเส้นทาง จะผ่านท่าเรือได้ไม่เกิน 10 ท่า และสำหรับเรือมาสเตอร์จะต้องผ่านท่าเรืออย่างน้อย 3 ท่า เรือฟีดเดอร์จะต้องผ่านท่าเรืออย่างน้อย 2 ท่า
- 5.2.5 เรือหนึ่งลำมีเส้นทางเดินเรือได้เพียงหนึ่งเส้นทางเท่านั้น
- 5.2.6 เรือมาสเตอร์จะต้องมีการกำหนดแผนการเดินทางเรือล่วงหน้า 1 เดือน เรือฟีดเดอร์ต้องมีการกำหนดแผนการเดินทางเรือล่วงหน้า 1 อาทิตย์
- 5.2.7 จะมีการออกกำหนดการขนถ่ายสินค้าสำหรับเรือแต่ละลำและท่าเรือแต่ละท่า ทุกครั้งก่อนเรือเข้าท่าเรือ 1 วัน

5.3 แนวทางในการพัฒนาต่อไป

- 5.3.1 ในอนาคตการพัฒนาโปรแกรมระบบการจัดการขนส่งสินค้าทางเรือที่อาจพัฒนาอัลกอริทึมในการหาเส้นทางขนส่งสินค้าให้มีความเหมาะสมมากขึ้น
- 5.3.2 อาจมีการพัฒนาให้โปรแกรมสามารถเก็บข้อมูลตำแหน่งของเรือได้ เพื่อใช้ในการพิจารณาเลือกเส้นทางให้สินค้าได้เหมาะสมยิ่งขึ้น

ภาคผนวก

คู่มือการติดตั้งโปรแกรม

1. เครื่องคอมพิวเตอร์ที่เหมาะสมสำหรับโปรแกรม

1.1 คอมพิวเตอร์ที่มีซีพียูรุ่น Pentium 500 MHz ขึ้นไป

1.2 ระบบปฏิบัติการ Windows 2000 Server สำหรับเครื่อง Server และ Window 95/98

สำหรับเครื่อง Client

1.3 ฮาร์ดดิสก์อย่างน้อย 2 GB

1.4 หน่วยความจำอย่างน้อย 128 MB

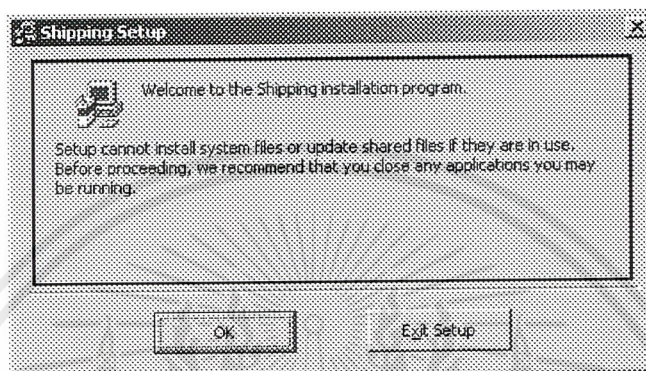
1.5 การ์ดจอภาพที่มีหน่วยความจำ 16 MB ขึ้นไป

2. ขั้นตอนการสร้างฐานข้อมูล

ก่อนทำการติดตั้งโปรแกรมผู้ใช้งานต้องทำการสร้างฐานข้อมูลขึ้นก่อน โดยสร้างฐานข้อมูลชื่อ Shipping ใน SQL Server 7.0 ซึ่งทำงานอยู่บน Windows 2000 Server บนเครื่อง Server และสร้างตารางฐานข้อมูล 9 ตาราง ประกอบด้วย Ship, Port, Item, Route, ItemPath, PortItem, ShiplItem, Edge, Schedule ซึ่ง Fields ต่างๆ ดังที่ได้กล่าวไว้ในบทที่ 3

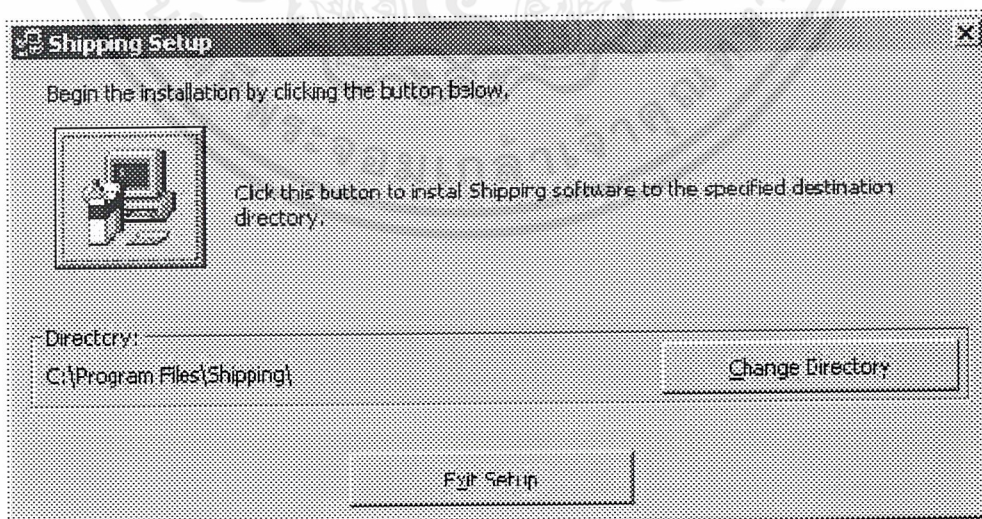
3. การติดตั้งโปรแกรมสำหรับเครื่อง Server

3.1 สำหรับการติดตั้งโปรแกรม ก่อนที่จะมีการติดตั้งโปรแกรมจะมีข้อความแจ้งผู้ใช้ให้ปิดโปรแกรมต่างๆ ก่อนที่จะทำการติดตั้ง เนื่องจากโปรแกรมนั้นจะไม่สามารถติดตั้งได้ถ้า System File และ Shared File ต่างๆถูกใช้อยู่ เมื่อทำการปิดโปรแกรมต่างๆเรียบร้อยแล้วให้กดปุ่ม Ok เพื่อเข้าสู่การติดตั้งโปรแกรม



รูปที่ 5.1 Shipping Setup

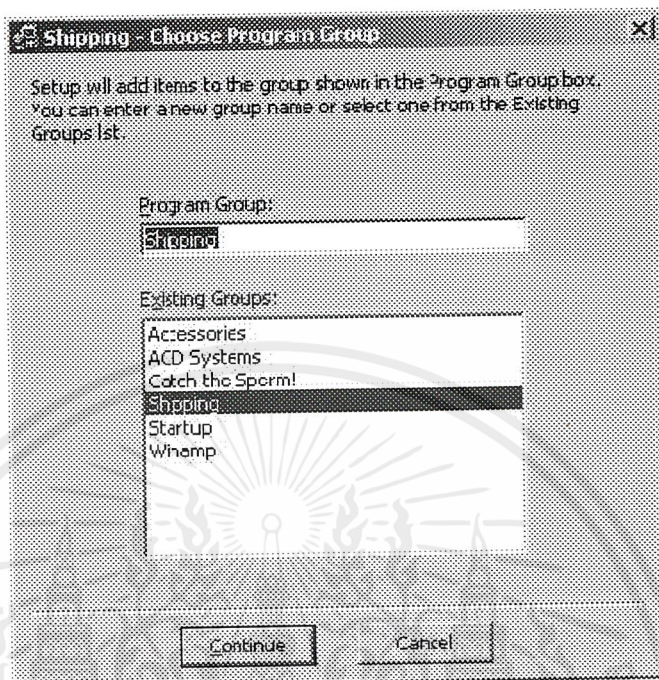
3.2 ผู้ใช้สามารถเลือก Directory ที่ต้องการติดตั้งโปรแกรมใหม่ได้ โดยกดปุ่ม Change Directory แล้วทำการเลือก Directory ที่ต้องการ จากนั้นกดปุ่มไอคอน Installation เพื่อทำการติดตั้งโปรแกรมใน Directory ที่ทำการเลือกไว้



รูปที่ 5.2 การเลือก Directory ที่ต้องการติดตั้งโปรแกรม

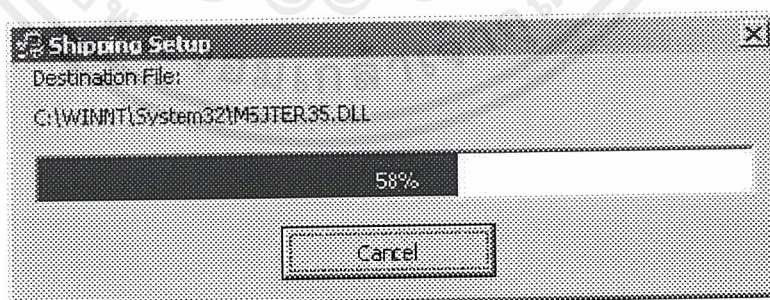
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 จากนั้นผู้ใช้จะต้องทำการเลือก Program Group โดยผู้ใช้สามารถใส่ชื่อ Program Group ใหม่ หรือเลือก Program Group ที่มีอยู่แล้วก็ได้ จากนั้นกดปุ่ม Continue



รูปที่ 5.3 การเลือก Program Group

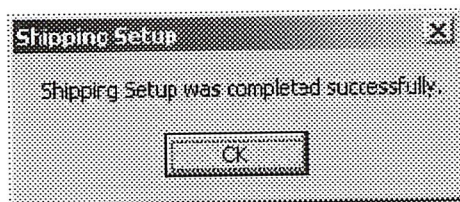
3.4 จะพบหน้าต่าง Setup ปรากฏขึ้นมาโดยจะเริ่มติดตั้งโปรแกรมตั้งแต่ 0% จนถึง 100% ซึ่งถ้าต้องการยกเลิกการติดตั้งโปรแกรมสามารถทำได้โดยกดปุ่ม Cancel



รูปที่ 5.4 หน้าต่าง Setup

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 จากนั้นกดปุ่ม Ok เพื่อให้การติดตั้งเสร็จสมบูรณ์



รูปที่ 5.5 หน้าต่าง Setup ที่เสร็จสมบูรณ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การเพิ่มข้อมูลในโปรแกรมเริ่มต้น

ก่อนทำการติดตั้งเครื่อง Client หรือการประมวลผลในระบบเพื่อหาเส้นทางของสินค้า และกำหนดการขนถ่ายสินค้า Administrator จะต้องทำการใส่ข้อมูลเบื้องต้นต่างๆก่อน ได้แก่

4.1 ข้อมูลท่าเรือ (Port) อย่างน้อย 3 ท่าเรือ เนื่องจากเส้นทางเดินเรือ (Route) แต่ละเส้นทางจะถูกสร้างจากท่าเรือตั้งแต่ 3 ท่าเรือขึ้นไปสำหรับเรือประเภท Master และสำหรับเรือประเภท Feeder เส้นทางจะถูกสร้างจากท่าเรือตั้งแต่ 2 ท่าเรือขึ้นไป

4.2 ข้อมูลเรือ (Ship) เมื่อกรอกข้อมูลเรือครบแล้วทำการเลือก Save จากเมนู File > Save โปรแกรมจะเปลี่ยนหน้าจอไปยังหน้าจอเส้นทางเดินเรือ (Route) สำหรับกำหนดเส้นทางเดินเรือให้เรื่อนั้นๆ ซึ่งทำได้ 2 วิธีคือ

4.2.1 กรณีที่มีเส้นทางเดินเรืออยู่แล้ว โดยจะเลือกจากเส้นทางที่มีอยู่แล้วใน List Box แล้วเลือก Save จากเมนู File > Save

4.2.2 กรณีที่ยังไม่มีเส้นทางเดินเรือหรือต้องการเลือกเพิ่มเส้นทางเดินเรือใหม่ โดยเลือก New Route จากเมนู File > New Route โปรแกรมจะทำการเปลี่ยนไปยังหน้าจอเพิ่มเส้นทางเดินเรือใหม่ (New Route)

สำหรับเรือประเภท Master เส้นทางจะถูกสร้างจากท่าเรือตั้งแต่ 3 ท่าเรือขึ้นไป และสำหรับเรือประเภท Feeder เส้นทางจะถูกสร้างจากท่าเรือตั้งแต่ 2 ท่าเรือขึ้นไป

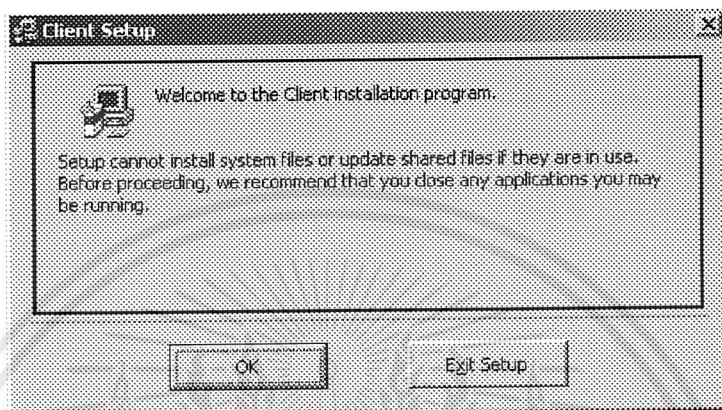
4.3 ข้อมูลเส้นทางเดินเรือใหม่ (New Route) โดยผู้ใช้จะต้องป้อนจำนวนท่าเรือที่ต้องการให้อยู่บนเส้นทางเดินเรือที่ต้องการ แล้วกดปุ่ม Ok จากนั้นโปรแกรมจะแสดงท่าเรือทั้งหมดขึ้นมาที่ Combo Box ให้ผู้ใช้ทำการเลือกท่าเรือตามเส้นทางเดินเรือที่ต้องการตามจำนวนท่าเรือที่ได้ป้อนไว้

เมื่อทำการใส่ข้อมูลข้างต้นเรียบร้อยแล้ว จะต้องมีกำหนดแผนกำหนดการเดินเรือ (Schedule Ship) ก่อน จึงจะเริ่มระบบได้

4.4 การกำหนดแผนกำหนดการเดินเรือ (Schedule Ship) เมื่อผู้ใช้ต้องการกำหนดแผนกำหนดการเดินเรือ โดยสำหรับเรือ Master จะต้องออกแผนกำหนดการเดินเรือล่วงหน้า 1 เดือน และสำหรับเรือ Feeder จะต้องออกแผนกำหนดการเดินเรือล่วงหน้า 1 อาทิตย์ ผู้ใช้จะต้องทำการเลือกเรือ แล้วกดปุ่ม Ok เพื่อทำการเลือกท่าเรือ และวันที่ที่เรือจะเข้าเทียบท่านั้นๆ จากนั้นผู้ใช้ต้องกดปุ่ม Save เพื่อเพิ่มและเปลี่ยนแปลงแผนกำหนดการเดินเรือ ซึ่งถือว่าเป็นการออกแผนกำหนดการเดินเรือใหม่

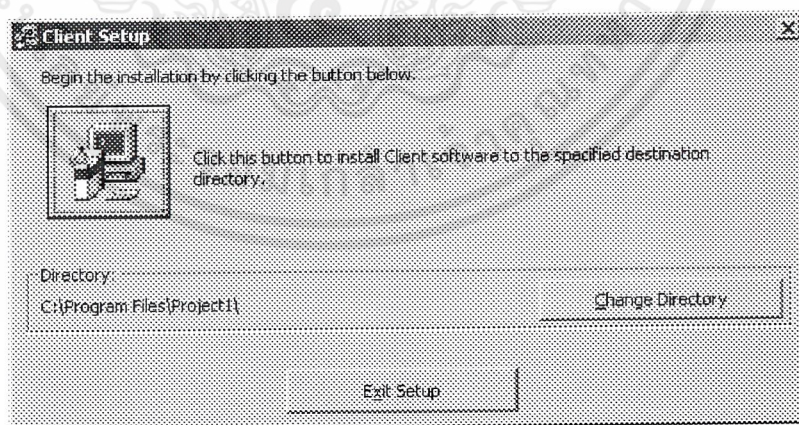
5. การติดตั้งโปรแกรมสำหรับเครื่อง Client

5.1 สำหรับการติดตั้งโปรแกรม ก่อนที่จะมีการติดตั้งโปรแกรมจะมีข้อความแจ้งผู้ใช้ให้ปิดโปรแกรมต่างๆ ก่อนที่จะทำการติดตั้ง เนื่องจากโปรแกรมนั้นจะไม่สามารถติดตั้งได้ถ้า System File และ Shared File ต่างๆถูกใช้อยู่ เมื่อทำการปิดโปรแกรมต่างๆเรียบร้อยแล้วให้กดปุ่ม Ok เพื่อเข้าสู่การติดตั้งโปรแกรม



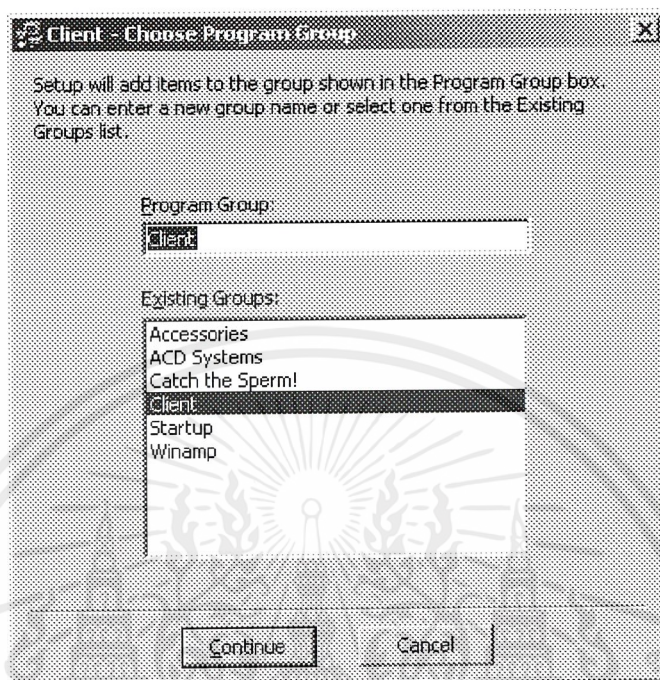
รูปที่ 5.6 Client Setup

5.2 ผู้ใช้สามารถเลือก Directory ที่ต้องการติดตั้งโปรแกรมใหม่ได้ โดยกดปุ่ม Change Directory แล้วทำการเลือก Directory ที่ต้องการ จากนั้นกดปุ่มไอคอน Installation เพื่อทำการติดตั้งโปรแกรมใน Directory ที่ทำการเลือกไว้



รูปที่ 5.7 การเลือก Directory ที่ต้องการติดตั้งโปรแกรม

5.3 จากนั้นผู้ใช้จะต้องทำการเลือก Program Group โดยผู้ใช้สามารถใส่ชื่อ Program Group ใหม่ หรือเลือก Program Group ที่มีอยู่แล้วก็ได้ จากนั้นกดปุ่ม Continue



รูปที่ 5.8 การเลือก Program Group

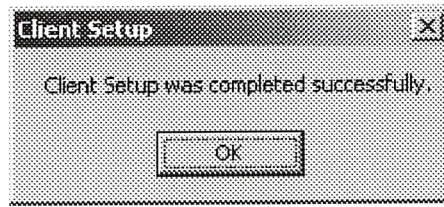
5.4 จะพบหน้าต่าง Setup ปรากฏขึ้นมาโดยจะเริ่มติดตั้งโปรแกรมตั้งแต่ 0% จนถึง 100% ซึ่งถ้าต้องการยกเลิกการติดตั้งโปรแกรมสามารถทำได้โดยกดปุ่ม Cancel



รูปที่ 5.9 หน้าต่าง Setup

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 จากนั้นกดปุ่ม Ok เพื่อให้การติดตั้งเสร็จสมบูรณ์



รูปที่ 5.10 หน้าต่าง Setup ที่เสร็จสมบูรณ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ศุภชัย สมพานิช. Database Programming กับ Visual Basic ฉบับมืออาชีพ. กรุงเทพฯ : อินโฟเพรส, 2543.

สัจจะ จรัสรุ่งรวิธร. คู่มือการเขียนโปรแกรมและใช้งาน Visual Basic 6.0 . กรุงเทพฯ : อินโฟเพรส, 2544.

สมพร ตีวรรณกุล. คู่มือการติดตั้งและใช้งาน Microsoft SQL Server 7.0 ฉบับสมบูรณ์. กรุงเทพฯ : อินโฟเพรส, 2543.

กิตติ ภัคดีวัฒนะกุล, จำลอง ครูอุตสาหะ. Visual Basic 6 ฉบับฐานข้อมูล. กรุงเทพฯ : บริษัท เคทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด, 2544.

กิตติ ภัคดีวัฒนะกุล, จำลอง ครูอุตสาหะ. Visual Basic 6 ฉบับโปรแกรมเมอร์. กรุงเทพฯ : บริษัท เคทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด, 2544.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้