

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบฐานข้อมูลมัลติมีเดีย
MULTIMEDIA DATABASE



ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2542

เลขที่.....
เลขทะเบียน..... 37046
วัน, เดือน, ปี..... 30 ธ.ค. 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
และการตลาดได้ ทั้งนี้ขอสงวนสิทธิ์ในการตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2542

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบฐานข้อมูลมัลติมีเดีย

MULTIMEDIA DATABASE

ผู้จัดทำ

1. นาย เกียรติณรงค์ ทองประเสริฐ รหัสประจำตัว 39014043

2. นาย ปรัชญา ศรีเสาวชาติ รหัสประจำตัว 39014309



อาจารย์ที่ปรึกษา

(อ. บัณฑิต พัสยา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบฐานข้อมูลมัลติมีเดีย

นายเกียรติณรงค์ ทองประเสริฐ 39014043

นายปรัชญา ศรีเสาวชาติ 39014309

อ.บัณฑิต พัสยา อาจารย์ที่ปรึกษา

ปีการศึกษา 2542

บทคัดย่อ

โมเดลของระบบฐานข้อมูลที่มีใช้กันอยู่ในปัจจุบันเป็นฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ซึ่งอยู่ในรูปของตาราง โดยจะมีการใช้งานกับชนิดข้อมูลที่มีรูปแบบง่าย ๆ ไม่ซับซ้อน เช่น ข้อมูลจำนวนเต็ม ตัวอักษร และข้อความ เป็นต้น แต่เมื่อมีงานที่จำเป็นต้องใช้ข้อมูลชนิดที่มีความซับซ้อนสูง (Complex Data type) เช่น ข้อมูลประเภทมัลติมีเดีย ข้อมูลประเภทออบเจกต์ ทำให้ระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ไม่สามารถนำมาใช้งานเพื่อตอบสนองความต้องการได้ จึงทำให้ต้องมีการใช้โมเดลอื่นเพื่อมาทดแทนข้อเสียของโมเดลนี้

ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ (Object Relational Database) ก็เป็นโมเดลหนึ่งที่ได้พัฒนาจากระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ซึ่งได้มีการขยายขีดความสามารถในการจัดการกับข้อมูลที่มีความซับซ้อน โดยมีการเพิ่มเติมคุณสมบัติของออบเจกต์ คอลเลกชัน นอกจากนี้ยังสามารถที่จะเขียนฟังก์ชัน เมธอด ต่าง ๆ เพื่อใช้ในการจัดการกับข้อมูลที่ซับซ้อนได้

ปฏิญานพนธ์นี้ได้พัฒนาระบบสารสนเทศของห้องสมุดเสมือน (Virtual Libaray) ซึ่งมีการจัดเก็บข้อมูลประเภทมัลติมีเดีย ซึ่งมีการจัดเก็บเป็นรูปภาพ เสียง และภาพยนตร์ โดยใช้ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ คือ อินฟอร์มิคซ์ ยูนิเวอร์เซล เซิร์ฟเวอร์ เวอร์ชัน 9.14 และในส่วนของแอปพลิเคชันในการพัฒนาคือ วิซวลเบสิก เวอร์ชัน 6.0 โดยการติดต่อผ่านคาล่าไคเร็กซ์เตอร์ เวอร์ชัน 3.5 และระบบการจัดการหนังสือ เป็นแบบของห้องสมุดรัฐสภาอเมริกัน

MULTIMEDIA DATABASE

Kiatnarong Tongprasert

Pratya Srisaowachart

Bundit Patsaya Advisor

ABSTRACT

Model currently used in database system which is in form of table is Relational Database. It is suitable for noncomplex data type e.g. integer, character, string. But in job concerned with complex data type, it doesn't satisfy us. This lead to bring other model to remove its disadvantage.

Object Relational Database is a model developed from Relational Database with more capability. It can manage complex data with high performance by object, collection qualification. In additional, we can write function or methods to use in database management.

This project has develop information system of Virtual Library by Informix Universal Server version 9.14 to manage multimedia data e.g. picture, sound, movie application by Visual Basic 6.0 and interface connection by Data Director 3.5. Classified book system in this project is according to LCC.

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือและร่วมมือจากบุคคลหลาย ๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้ปริญญาานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์ บัณฑิต พัสยา อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

ขอขอบพระคุณพี่ปิยะ รุ่นที่ปริญญาโท ที่ให้การแนะนำในเรื่องของการจัดเก็บข้อมูลชนิดมัลติมีเดีย ฝ่ายเทคนิคของบริษัทอินฟอรั่มิกส์ ประเทศไทย ที่ให้คำแนะนำเกี่ยวกับดาต้าโคเรกเตอร์

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจเอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

เกียรติณรงค์ ทองประเสริฐ
ปรัชญา ศรีเสาวชาติ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	VII
สารบัญตาราง	X
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของงานวิจัย	2
1.4 วิธีดำเนินงาน	2
บทที่ 2 Informix Universal Server	3
2.1 อินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์ (INFORMIX - Universal Server)	3
- สถาปัตยกรรมของอินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์	3
2.2 จุดเด่นของโครงสร้างสถาปัตยกรรมของอินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์	4
- การออกแบบมีประสิทธิภาพสูง (Performance)	4
- การออกแบบให้สามารถขยายได้ (Extensible)	4
- ออกแบบให้เชื่อมต่อกันอย่างดี (Integrated)	4
- ออกแบบที่นวัตกรรมใหม่ (Innovation)	4
- การควิรีข้อมูลของอินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์	5
2.3 ชนิดข้อมูลในอินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์	5
- ข้อมูลชนิดพื้นฐาน (Built-in Data types)	5
- ชนิดข้อมูลที่สร้างใหม่ (User defined type)	6
2.4 การสืบทอดคุณสมบัติในอินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์	8
- การสืบทอดคุณสมบัติสำหรับ named row type	8
- การสืบทอดคุณสมบัติสำหรับตาราง	9
บทที่ 3 หลักการดาต้าไดเรกเตอร์ออบเจกต์ Data Director Objects	10
3.1 ดาต้าไดเรกเตอร์ออบเจกต์คืออะไร	10
3.2 ทำไมจึงใช้ดาต้าไดเรกเตอร์ออบเจกต์	10
3.3 โครงสร้างลำดับชั้นของดาต้าไดเรกเตอร์ออบเจกต์	11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.4 ออบเจกต์และคอลเลกชัน (Objects and Collections)	12
3.5 ชนิดของออบเจกต์	12
- ออบเจกต์เฟรมเวิร์ค (Framework Objects)	12
- ออบเจกต์ข้อมูล (Data Objects)	14
- ออบเจกต์ช่วยเหลือ (Support Object)	17
3.6 การใช้คำสั่งไคลเรคเตอร์ออบเจกต์	17
- การใช้คุณสมบัติ และ เมทอด	18
- การเริ่มต้นการทำงานกับคีย์โอ	19
- การ ล็อคออนไปยังฐานข้อมูล	21
- การทำงานกับข้อมูล	24
บทที่ 4 ระบบการจัดหมวดหมู่หนังสือแบบห้องสมุดรัฐสภาอเมริกัน	30
4.1 สัญลักษณ์ของระบบการจัดหมู่หนังสือแบบห้องสมุดรัฐสภาอเมริกัน	30
4.2 ข้อดีข้อเสียของระบบ LCC	34
4.3 แผนการจัดหมู่หนังสือของระบบห้องสมุดรัฐสภาอเมริกัน	35
4.4 การใช้ตารางเลขคัตเตอร์และการใช้เลขคัตเตอร์ในกรณีพิเศษ	37
- การใช้ตารางกำหนดเลขคัตเตอร์	37
- หลักและวิธีการกำหนดเลขคัตเตอร์และเลขผู้แต่ง	38
- การใช้เลขคัตเตอร์ในกรณีพิเศษ	40
4.5 การกำหนดและวิเคราะห์เลขหมู่หนังสือหมวด A-Z	43
บทที่ 5 การออกแบบ ER Diagram และการออกแบบโปรแกรมประยุกต์	45
5.1 Requirement และ Specification ของ Application	45
5.2 การวิเคราะห์และการออกแบบ ER Diagram	48
- การออกแบบ ER Diagram	48
- การออกแบบตารางที่ใช้ใน โปรแกรมประยุกต์	51
- การออกแบบโปรแกรมประยุกต์	57
- โฟลว์ชาร์ทของโปรแกรมประยุกต์	58
- การออกแบบการจัดการคำสั่งเบสเซิร์ฟเวอร์	61
5.3 Data Director	66
บทที่ 6 Stored Procedure Language	67
6.1 บทนำ	67
6.2 การสร้าง SPL รูทีน	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
6.3 กำหนดชื่อ พารามิเตอร์ และชนิดข้อมูลที่จะทำการส่งค่าคืน	67
6.4 การประกาศค่าตัวแปรและการกำหนดค่าให้ SPL	69
6.5 Foreach และ Cursor	70
6.6 Flow Control	71
6.7 การจัดการกับ Collection Type	73
6.8 การเรียกใช้รoutines	76
บทที่ 7 ตัวอย่างการใช้งานโปรแกรมระบบสารสนเทศห้องสมุด	77
7.1 ตัวอย่างโปรแกรมของส่วนผู้ใช้	77
7.2 ตัวอย่างโปรแกรมของผู้ดูแลระบบ	84
บทที่ 8 สรุปและวิจารณ์	87
8.1 สรุป	87
8.2 แนวทางในการพัฒนาต่อ	87
บรรณานุกรม	88
ภาคผนวก ก การติดตั้ง Informix Universal Server	89
ภาคผนวก ข วิธีการสร้างโอเปคไทป์ลงใน Informix Universal Server	93
ภาคผนวก ค การสร้างการติดต่อระหว่าง Visual Basic กับฐานข้อมูลโดยผ่าน Data Director	108
ภาคผนวก ง Source Code	114

สารบัญรูปภาพ

	หน้า
รูปที่ 2-1 สถาปัตยกรรมของอินฟอร์เม็กซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์	3
รูปที่ 2-2 การควิรีข้อมูลของระบบฐานข้อมูลแบบต่างๆ	5
รูปที่ 2-3 Complex types ที่มีอยู่ใน Informix Universal Server	7
รูปที่ 2-4 ลักษณะการสืบทอดคุณสมบัติระดับ named row type และตาราง	9
รูปที่ 3-1 โครงสร้างลำดับชั้นของดีดีโอ	11
รูปที่ 5-1 ER Diagram ความสัมพันธ์ของหนังสือ	48
รูปที่ 5-2 ER Diagram ความสัมพันธ์ของภาพยนตร์	49
รูปที่ 5-3 ER Diagram ความสัมพันธ์ของเพลง	50
รูปที่ 5-4 สถาปัตยกรรมของโปรแกรมประยุกต์ระบบสารสนเทศห้องสมุด	57
รูปที่ 5-5 โฟลว์ชาร์ทการทำงานของระบบสารสนเทศห้องสมุด	58
รูปที่ 5-6 โฟลว์ชาร์ทการจัดการดูแลระบบ	59
รูปที่ 5-7 โฟลว์ชาร์ทการสืบค้นข้อมูล	60
รูปที่ 5-8 การอิมพอร์ตโมเดล	66
รูปที่ 7-1 หน้าต่างแรกเมื่อเข้าโปรแกรม	77
รูปที่ 7-2 เมนู ในการเลือกค้นข้อมูล	77
รูปที่ 7-3 การค้นหาหนังสือ	78
รูปที่ 7-4 ผลการค้นหาหนังสือ	78
รูปที่ 7-5 รายละเอียดของหนังสือที่ค้นหาพร้อมรูปภาพ	79
รูปที่ 7-6 ตำแหน่งของหนังสือในห้องสมุด	79
รูปที่ 7-7 ตำแหน่งของหนังสือในตู้โดยละเอียด	80
รูปที่ 7-8 เมนูในการค้นหาข้อมูลประเภทเพลง	80
รูปที่ 7-9 ผลการค้นหาเพลง	81
รูปที่ 7-10 รายละเอียดของเพลงพร้อมตัวอย่างเพลง	81
รูปที่ 7-11 ตำแหน่งของเพลงในห้องสมุด	82
รูปที่ 7-12 หน้าจอค้นหาข้อมูลภาพยนตร์	82
รูปที่ 7-13 รายการของการค้นหาข้อมูลภาพยนตร์	83
รูปที่ 7-14 รายละเอียดของภาพยนตร์พร้อมภาพยนตร์ตัวอย่าง	83
รูปที่ 7-15 ตำแหน่งของภาพยนตร์ในห้องสมุด	84
รูปที่ 7-16 หน้าต่างแรกเมื่อเรียกโปรแกรม	84
รูปที่ 7-17 เมนูในการป้อนข้อมูล	85
รูปที่ 7-18 หน้าต่างการ Insert ข้อมูลหนังสือ	85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 7-19 หน้าต่างการ Update ข้อมูลหนังสือ	86
รูปที่ 7-20 หน้าต่างการ Select ของหนังสือ	86
รูปที่ ก-1 Informix Setnet32	92
รูปที่ ข-1 BladeSmith	95
รูปที่ ข-2 New Project Wizard page 1	95
รูปที่ ข-3 New Project Wizard page 2	96
รูปที่ ข-4 เมื่อสิ้นสุด Wizard page	96
รูปที่ ข-5 การ Insert Opaque Data Type	97
รูปที่ ข-6 New Opaque Type Wizard page 1	97
รูปที่ ข-7 New Opaque Type Wizard page 2	98
รูปที่ ข-8 New Opaque Type Wizard page 3	98
รูปที่ ข-9 New Opaque Type Wizard page 4	99
รูปที่ ข-10 การระบุชนิดข้อมูลใน Opaque datatype	99
รูปที่ ข-11 การสร้าง Source code	100
รูปที่ ข-12 การ Open Workspace	100
รูปที่ ข-13 การ Compile เป็น file นามสกุล bld	101
รูปที่ ข-14 การเรียกใช้งาน BladPack	101
รูปที่ ข-15 หน้าต่าง BladPack	102
รูปที่ ข-16 Installation Build Wizard page 1	102
รูปที่ ข-17 Installation Build Wizard page 2	103
รูปที่ ข-18 Installation Build Wizard page 3	103
รูปที่ ข-19 Installation Build Wizard page 4	104
รูปที่ ข-20 Select an installation to build	104
รูปที่ ข-21 Installation Build Output	104
รูปที่ ข-22 การใช้ BladeManager	105
รูปที่ ข-23 BladeManger เมื่อเร าสื่อกรฐานข้อมูลแล้ว	105
รูปที่ ข-24 การ login ใน โปรแกรม Schema Knowledge	106
รูปที่ ข-25 โปรแกรม Schema Knowledge เมื่อเลือกชนิดข้อมูลที่มิพื้นฐานข้อมูล	107
รูปที่ ค-1 ODBC Data Source Administrator	108
รูปที่ ค-2 Create New Data Source	108
รูปที่ ค-3 Informix-CLI DSN setup	109

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ ค-4 Data Director Connect	109
รูปที่ ค-5 Select Data Source	110
รูปที่ ค-6 Informix Connect	110
รูปที่ ค-7 Data Director Connect	111
รูปที่ ค-8 Model Viewer	111
รูปที่ ค-9 Import Model	112
รูปที่ ค-10 Items to Import	112
รูปที่ ค-11 Verify Inferred Relationships	113
รูปที่ ค-12 Model Viewer	113



สารบัญตาราง

	หน้า
ตารางที่ 2-1 Built-in data type ของ Informix Universal Server	6
ตารางที่ 4-1 ตารางพยัญชนะ ใช้สำหรับชื่อที่ขึ้นต้นด้วยพยัญชนะตัวแรกไม่มีรูปสระประกอบ แบ่งออกเป็น 4 กลุ่ม แต่ละกลุ่มมีตัวเลขประจำ ดังต่อไปนี้	41
ตารางที่ 4-2 ตารางสระ ใช้สำหรับชื่อที่ขึ้นต้นด้วยคำแรกหรือพยางค์แรกที่มีรูปสระประกอบแบ่ง ออกเป็น 4 กลุ่ม แต่ละกลุ่มมีตัวเลขประจำ ดังต่อไปนี้	41
ตารางที่ ข-1 ส่วนต่างๆ ที่ถูกสร้างภายใน Project	93



บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

เนื่องจากปัจจุบันนี้มีความต้องการใช้และจัดเก็บข้อมูลที่มีความซับซ้อนมากยิ่งขึ้นอย่างเช่น ข้อมูลเสียง ภาพ วิดีโอ และ ข้อมูลประเภทมัลติมีเดีย (Multimedia) อื่น ๆ เป็นต้น การจัดการกับข้อมูลเหล่านี้ก็ต้องมีความยุ่งยากและซับซ้อนมากยิ่งขึ้นเช่นกัน ระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System : RDBMS) จะมีความยุ่งยากในการจัดการและจัดเก็บข้อมูลเหล่านี้ เราจึงได้ทำการศึกษาระบบฐานข้อมูลแบบเชิงวัตถุสัมพันธ์ (Object Relational Database Management : ORDBMS) เนื่องจากระบบฐานข้อมูลแบบนี้มีความเหมาะสมในการรองรับข้อมูลที่มีความซับซ้อนมาก ๆ ได้ ช่วยให้ง่ายในการจัดการกับข้อมูลที่มีความซับซ้อน และช่วยให้โปรแกรมเมอร์พัฒนาได้ง่ายยิ่งขึ้น โดยระบบฐานข้อมูลแบบเชิงวัตถุสัมพันธ์นี้ได้นำเอาหลักการของ Object Oriented มาประยุกต์ร่วมกับหลักการจัดการข้อมูลของระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System)

ในโครงการนี้ได้ทำการเลือกที่จะศึกษาระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ของบริษัท Informix ซึ่งผลิตภัณฑ์ของ Informix รุ่นที่มีคุณสมบัติเป็นระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ที่เราเลือกใช้ก็คือ Informix Universal Server 9.14 มาศึกษาและจัดเก็บข้อมูลประเภทมัลติมีเดียและจัดการข้อมูลของระบบ

ระบบที่ได้ทำการทดลองพัฒนานี้ เป็นระบบสืบค้นข้อมูลของห้องสมุด โดยจะทำเป็นห้องสมุดเสมือน (Virtual Library) ซึ่งสามารถจะดูข้อมูลของหนังสือนั้น ๆ ได้ และทราบที่อยู่ของหนังสือได้ นอกจากนี้ยังสามารถสืบค้นข้อมูลที่เป็นเพลง และภาพยนตร์ที่มีบริการอยู่ในห้องสมุดได้ พร้อมทั้งแสดงข้อมูลที่เป็นเพลง และภาพยนตร์ผ่านหน้าจอผู้ใช้ได้ ซึ่งในการจะทำให้เกิดระบบนี้ได้จะต้องจะเก็บข้อมูลทางด้านมัลติมีเดียได้แก่ รูปของหนังสือ เพลง และภาพยนตร์

1.2 วัตถุประสงค์ของโครงการ

1. ศึกษาหลักการของโมเดลฐานข้อมูลเชิงวัตถุสัมพันธ์ เพื่อนำมาเป็นแบบจำลองข้อมูลระดับแนวคิด และศึกษาลักษณะการแทนข้อมูลลงในฐานข้อมูล
2. ทดลองการใช้งานระบบจัดการฐานข้อมูล (Database Management System : DBMS) ที่สนับสนุนโมเดลฐานข้อมูลเชิงวัตถุสัมพันธ์ซึ่งในโครงการนี้ได้ใช้อินฟอร์มิคส์ยูนิเวอร์แซลเซิร์ฟเวอร์ (Informix Universal Server) ซึ่งเป็นระบบจัดการฐานข้อมูลที่สามารถจัดเก็บข้อมูลได้ทั้งที่เป็นโมเดลฐานข้อมูลเชิงสัมพันธ์และโมเดลฐานข้อมูลเชิงวัตถุสัมพันธ์
3. ศึกษาการสร้างและการออกแบบระบบฐานข้อมูลเพื่อนำความรู้ที่ได้ไปประยุกต์ใช้งาน
4. ศึกษาและทำการทดลองใช้งานชนิดข้อมูลต่างๆที่สนับสนุนโมเดลฐานข้อมูลเชิงวัตถุสัมพันธ์ที่ระบบจัดการฐานข้อมูลมีให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. สร้างโปรแกรมประยุกต์ (Application) ที่ทำการเก็บข้อมูลที่มีความซับซ้อนซึ่งได้ทำการเลือกกระบวนการสืบค้นข้อมูลของห้องสมุด ซึ่งเป็นการจัดเก็บข้อมูลประเภทมัลติมีเดีย โดยใช้คุณลักษณะของฐานข้อมูลเชิงวัตถุสัมพันธ์ของ Informix Universal Server
6. ศึกษากระบวนการจัดหมวดหมู่หนังสือแบบห้องสมุดรัฐสภาอเมริกัน เพื่อใช้ในการจัดเก็บข้อมูลในโปรแกรมประยุกต์ตามแบบมาตรฐาน

1.3 ขอบเขตของงานวิจัย

งานวิจัยนี้จะทำการสร้างฐานข้อมูลที่ใช้โมเดลฐานข้อมูลเชิงวัตถุสัมพันธ์ โดยจะทำการพัฒนาบนระบบจัดการฐานข้อมูลของอินฟอร์มิคส์ที่ซอร์สโค้ดเปิด เซิร์ฟเวอร์งาน และได้นำเอาความสามารถของระบบฐานข้อมูลเชิงวัตถุสัมพันธ์มาใช้ในการจัดเก็บข้อมูลมัลติมีเดีย ได้แก่ รูปภาพ เสียง และภาพยนตร์ และได้ทำการพัฒนาโปรแกรมประยุกต์ที่ทำการติดต่อกับฐานข้อมูลที่เป็นฐานข้อมูลเชิงวัตถุสัมพันธ์ซึ่งในงานวิจัยนี้เลือกกระบวนการสืบค้นข้อมูลของห้องสมุดเป็นกรณีศึกษา โดยจะทำการพัฒนาโดยใช้วีซวลเบสิก (Visual Basic) ทำการติดต่อกับฐานข้อมูลผ่านดาต้าไดเรกเตอร์ (Data Director) ซึ่งเป็นตัวจัดการการติดต่อกับฐานข้อมูลของระบบจัดการฐานข้อมูลอินฟอร์มิคส์ซอร์สโค้ดเปิดเซิร์ฟเวอร์และทำการพัฒนาระบบนี้บนระบบปฏิบัติการ WindowNT

1.4 วิธีการดำเนินงาน

จะเป็นการศึกษากระบวนการจัดการฐานข้อมูลที่ใช้ การใช้งานระบบ ความสามารถของระบบในการรองรับระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ ชนิดข้อมูลการจัดการกับข้อมูลซึ่งจะมีรายละเอียดอยู่ในบทที่ 4 รวมถึงมีวิธีการจัดตั้งระบบ

ในการดำเนินโครงการนี้ คือ ศึกษาตัวจัดการระบบฐานข้อมูลของ อินฟอร์มิคส์ ซอร์สโค้ดเปิดเซิร์ฟเวอร์ ซึ่งได้ทำการศึกษาในเรื่อง ระบบการจัดการฐานข้อมูล การใช้งานระบบ ความสามารถของระบบในการรองรับระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ ชนิดข้อมูลการจัดการกับข้อมูล ชนิดข้อมูลที่เหมาะสมในการเก็บข้อมูลประเภทมัลติมีเดีย หลังจากนั้นได้ศึกษาหลักการของ Data Director Objects ระบบการจัดหมวดหมู่หนังสือแบบห้องสมุดรัฐสภาอเมริกัน และได้ออกแบบระบบฐานข้อมูลที่ใช้ในการเก็บข้อมูลประเภทมัลติมีเดีย

หลังจากที่ได้ทำการออกแบบแล้ว จึงเริ่มทำการเขียนโปรแกรมโดยใช้ Visual Basic 6.0 ทำการติดต่อกับระบบฐานข้อมูล Informix Universal Server โดยใช้การติดต่อผ่านทาง Data Director ซึ่งเป็นเครื่องมือเฉพาะที่ใช้ในการติดต่อ เพื่อใช้ในการพัฒนาระบบสืบค้นหนังสือห้องสมุด ที่มีการจัดเก็บข้อมูลตามแบบห้องสมุดรัฐสภาอเมริกัน

บทที่ 2

Informix Universal Server

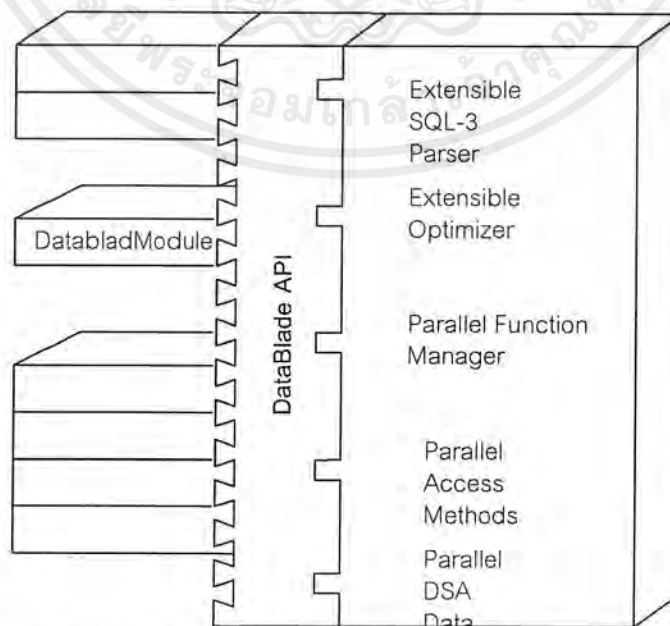
2.1 อินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์ (INFORMIX - Universal Server)

อินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์ เป็นระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ (ORDBMS) ที่มีความสามารถของระบบฐานข้อมูลเชิงวัตถุ (OODBMS) และระบบฐานข้อมูลเชิงสัมพันธ์ (RDBMS) โดยภาษาที่ใช้กับระบบฐานข้อมูลนี้ได้มีการนำภาษา SQL3 มาใช้ ดังนั้นการทำงานของ อินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์ จะต้องอ้างอิงกับมาตรฐานของ SQL3 ที่ทาง ANSI และ ISO ได้กำหนดไว้เช่นชนิดของข้อมูล, การทำคิวรี เป็นต้น

ระบบการจัดการฐานข้อมูลนี้ได้รวมเอาระบบฐานข้อมูลที่มีประสิทธิภาพและสามารถขยายระบบได้อย่างต่อเนื่อง (Scalability) ของสถาปัตยกรรมของอินฟอร์มิคซ์ ในแบบดีเอสเอ (Dynamic Scalable Architecture) การออกแบบเช่นนี้ทำให้อินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์เป็นระบบจัดการฐานข้อมูลที่เหมาะสมสำหรับองค์กรที่ต้องการระบบจัดการฐานข้อมูลที่มีความเชื่อถือได้และมีประสิทธิภาพสูง มีการขยายขีดความสามารถในการจัดการการค้นหาข้อมูลที่ซับซ้อนและข้อมูลในทุกรูปแบบ เช่น ภาพ เสียง เว็บ หรือเอกสารอิเล็กทรอนิกส์หรือข้อมูลแบบ 2D 3D ได้ ซึ่งอินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์จะใช้ค่าตัวเลขในการจัดการข้อมูลนั้นได้อย่างมีประสิทธิภาพ

2.1.1 สถาปัตยกรรมของอินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์

ในอินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์ ข้อมูลเกี่ยวกับชนิดของข้อมูล (Data type), ฟังก์ชันและการเข้าถึงเมทริกซ์จะเก็บไว้ในตารางแคตตาล็อก (Catalog Table) มากกว่าที่จะเก็บฮาร์ดโค้ด (Hard-coded) ไว้ในเซิร์ฟเวอร์



รูปที่ 2-1 สถาปัตยกรรมของอินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบของเซิร์ฟเวอร์เอ็นจิน (Server Engine) และหน้าที่ต่างๆ มีดังนี้

- **พาสเซอร์ (Parser)** ทำหน้าที่อ่านและตรวจสอบไวยากรณ์ ยูสเซอร์ดีฟายด์ฟังก์ชัน (User-defined function) และ ชนิดของข้อมูลจะถูกบันทึกลงในตารางแคตตาล็อกและจะถูกตรวจสอบโดยพาสเซอร์
- **อ็อบติไมซ์เซอร์ (Optimizer)** ทำหน้าที่เลือกทางที่ดีที่สุดเพื่อเข้าถึงข้อมูลที่ถูกคิวรี ซึ่งจะทำได้โดยคาค้าเบลค เดเวลอปเปอร์ (DataBlad Developer)
- **ฟังก์ชันเมเนเจอร์ (Function Manager)** ทำหน้าที่หาและจัดการยูสเซอร์ดีฟายด์ฟังก์ชันเพราะว่ายูสเซอร์ดีฟายด์ฟังก์ชันจะถูกเก็บอยู่ในตารางแคตตาล็อก
- **แอกซสเมททอด (Access Method)** ที่เหมาะสมจะถูกเลือกมาสำหรับข้อมูลที่กำลังถูกแก้ไข ซึ่งจะทำได้โดยคาค้าเบลคเดเวลอปเปอร์
- **คาค้าเมเนเจอร์ (Data Manager)** ทำหน้าที่เคลื่อนย้ายข้อมูลเข้า-ออกจากดิสก์

2.2 จุดเด่นของโครงสร้างสถาปัตยกรรมของอินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์

2.2.1 การออกแบบมีประสิทธิภาพสูง (Performance)

การออกแบบอินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์ ใช้โครงสร้างเดียวกับระบบฐานข้อมูลแบบขนานแบบดีเอสเอ ซึ่งออกแบบให้สามารถใช้ได้ทั้งระบบคอมพิวเตอร์แบบโปรเซสเซอร์เดี่ยวและแบบหลายโปรเซสเซอร์ และเป็นระบบที่สามารถใช้ประสิทธิภาพของฮาร์ดแวร์ได้อย่างมีประสิทธิภาพ และอินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์ยังสามารถปรับแต่งให้การทำงานของระบบที่บางครั้งอาจเป็นโพรเซสที่ซับซ้อนให้สามารถจัดการกับข้อมูลที่เกี่ยวข้องอย่างใกล้ชิด ทำให้การพัฒนาโปรแกรมทำได้ง่ายโดยไม่จำเป็นต้องค้นหาข้อมูลที่ซับซ้อนเอง

2.2.2 การออกแบบให้สามารถขยายได้ (Extensible)

ในระบบการจัดการฐานข้อมูลโดยทั่วไปจะจัดการกับข้อมูลที่เป็นตัวเลข ตัวอักษร แต่อินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์ออกแบบให้สามารถใช้งานอย่างไม่จำกัดโดยสามารถใช้งานกับข้อมูลประเภทใดก็ได้ เช่น ภาพ วิดีโอ เสียง แคนเวลา สองมิติ สามมิติ ภาพ ข้อความ และข้อมูลที่ใช้กับเว็บ

2.2.3 ออกแบบให้เชื่อมต่อกันอย่างดี (Integrated)

การออกแบบที่รวบรวมข้อดีของระบบจัดการฐานข้อมูลแบบขนานและโครงสร้างข้อมูลแบบออบเจกต์โอเรียนเตดที่ทำให้ระบบสามารถขยายขีดความสามารถและนำกลับมาใช้ใหม่ได้อย่างมีประสิทธิภาพ การรวมกันนี้ทำให้เป็นระบบจัดการฐานข้อมูลที่ทั้งสามารถขยายได้และเพิ่มความสามารถของระบบได้เป็นอย่างดี การที่สามารถรวมอยู่ในระบบเดียวกันได้นี้ทำให้ได้ประสิทธิภาพสูงสุด และลดความซ้ำซ้อนของระบบจัดการฐานข้อมูลและระบบงานได้ และยังคงไว้ด้วยความปลอดภัยและความเชื่อถือได้ของระบบ

2.2.4 ออกแบบที่นวัตกรรมใหม่ (Innovation)

การออกแบบอินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์ที่คำนึงถึงความเป็นระบบเปิด (Open System) ระบบที่ขยายได้โดยไม่มีขีดจำกัด (Unlimit Extensibility) และเพิ่มประสิทธิผลให้กับโปรแกรมเมอร์ทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้สามารถสร้างระบบที่เป็นสากลต่างๆ เช่น มาตรฐานเอสคิวแอล 3 การเชื่อมต่อกับเน็ตสเคปหรือ ไมโครซอฟต์บราวซ์เซอร์ หรือการใช้งานในระบบขนาดเล็กจนถึงขนาดใหญ่ได้อย่างมีประสิทธิภาพ

2.2.5 การคิวรีข้อมูลของอินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์

ระบบฐานข้อมูลของอินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์มีข้อมูลที่มีโครงสร้างเป็นแบบข้อมูลที่ซับซ้อน (Complex Data) ซึ่งเราสามารถที่จะคิวรีข้อมูลในรูปแบบของข้อมูลที่ซับซ้อนได้ รวมทั้งสามารถสร้างข้อมูลที่เป็นไทพ์ชนิดใหม่ได้

Query	Relational DBMS	Object Relational DBMS
No Query	File System	Object-Oriented DBMS
	Simple Data	Complex Data

รูปที่ 2-2 การคิวรีข้อมูลของระบบฐานข้อมูลแบบต่างๆ

2.3 ชนิดข้อมูลในอินฟอร์มิคซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์

ในระบบฐานข้อมูลของ Informix Universal Server ได้มีการแบ่งออกเป็น 3 ประเภทหลักๆคือ

- ชนิดข้อมูลพื้นฐาน (Built-in data type) เป็นชนิดข้อมูลที่มีมากับ Informix Universal Server
- ชนิดข้อมูลที่สร้างขึ้นใหม่ (User-defined types) เป็นชนิดข้อมูลที่ระบบฐานข้อมูลอนุญาตให้ผู้ใช้สามารถที่จะสร้างชนิดข้อมูลขึ้นมาใช้เอง ซึ่งแบ่งเป็น 2 ชนิดคือ
 1. Opaque type
 2. Distinct type
- ข้อมูลแบบ Complex เป็นชนิดของข้อมูลที่มีการกำหนดไว้ในมาตรฐาน SQL3 ซึ่งประกอบด้วย
 1. Collection type (set, list, multiset)
 2. Row types

2.3.1 ข้อมูลชนิดพื้นฐาน (Built-in Data types)

เป็นชนิดข้อมูลที่มีลักษณะดังนี้

- เป็นชนิดข้อมูลพื้นฐานซึ่งค่าค่าเบสเซิร์ฟเวอร์จัดให้มีไว้
- ไม่สามารถแบ่งย่อยออกเป็นส่วนเล็กๆ ต่อไปได้อีก (Atomic)
- บิวท์อินค่าค่าไทพ์ของยูนิเวอร์แซล เซิร์ฟเวอร์ มีดังนี้

ตารางที่ 2-1 Built-in data type ของ Informix Universal Server

Data types		Informix Universal Server
Character		CHAR, CHARACTER VARYING (VARCHAR), LVARCHAR
Numeric	Exact numeric	DECIMAL, MONEY, SMALLINT, INTEGER, INT8, SERIAL, SERIAL8
	Approximate numeric	SMALLFLOAT, FLOAT
Large object	Simple large object	TEXT, BYTE
	Smart large object	CLOB, BLOB
Time		DATE, DATETIME, INTERVAL
Miscellaneous		BOOLEAN

2.3.2 ชนิดข้อมูลที่สร้างขึ้นใหม่ (User defined type)

เป็นชนิดข้อมูลที่ผู้ใช้สร้างขึ้นใหม่ นอกเหนือจากที่ทาง Informix Universal Server มีให้เพื่อความยืดหยุ่นในการเก็บข้อมูลและการจัดการ โดยทาง Informix Universal server ได้แบ่งข้อมูลชนิดนี้ออกเป็น 2 ชนิดคือ

2.3.2.1 Opaque type

เป็นชนิดข้อมูลที่มีลักษณะซ่อนเร้นข้อมูลภายในทั้งหมด (Fully Encapsulated) ซึ่งเป็นข้อมูลที่ไม่สามารถจะแบ่งแยกย่อยได้ (Atomic Data type) ที่เราสามารถที่จะกำหนดหรือสร้างไว้ในระบบฐานข้อมูล โดยที่ระบบฐานข้อมูลจะไม่รู้ถึงลักษณะของโครงสร้างภายในของข้อมูลชนิดนี้ว่าเป็นแบบใดเหมือนกับข้อมูลที่มากับระบบฐานข้อมูล (Built-in Data type) ดังนั้นจึงจำเป็นที่จะต้องมีฟังก์ชันต่างๆ ที่คอยจัดการและดำเนินการกับ Opaque type Opaque type ใช้เก็บข้อมูลขนาดใหญ่ เช่น ข้อความ เสียง วิดีโอ เป็นต้น Opaque type จะสนับสนุนออบเจกต์โอเรียนเต็ลของข้อมูลไปรเวท (Private data)

2.3.2.2 Distinct type

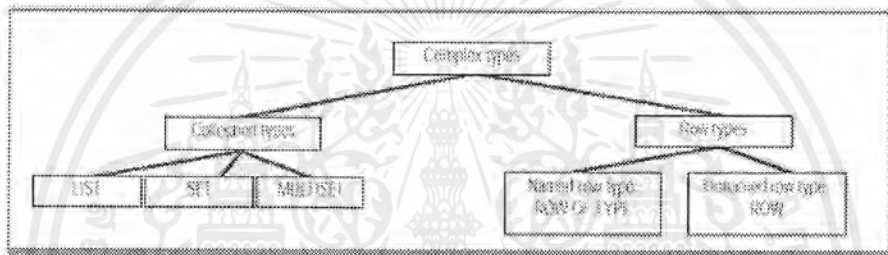
เป็นชนิดของข้อมูลที่โครงสร้างภายในเหมือนกับชนิดข้อมูลที่มีอยู่แล้ว โดยที่จะมีชื่อและฟังก์ชันที่ถูกสร้างมาแตกต่างไปจากข้อมูลที่เป็นต้นแบบ (Source type) ตัวอย่างเช่น เราสร้างข้อมูลชนิด decnum ที่มีลักษณะของโครงสร้างภายในเหมือนกับข้อมูลชนิด real ดังนั้นฟังก์ชันทั้งหมดที่ใช้กับ real จะสามารถที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะทำงานกับ decnum ได้ แต่อย่างไรก็ตามเราไม่สามารถที่จะทำการบวก ลบ หรือเปรียบเทียบข้อมูลระหว่าง decnum กับ real ได้ จำเป็นจะต้องทำผ่านฟังก์ชันที่มีหน้าที่แปลงชนิดข้อมูล(Casting function เป็นฟังก์ชันที่ทำการ Cast (เป็น mechanism ที่ทำการแปลงค่าจากข้อมูลชนิดหนึ่งไปเป็นข้อมูลอีกชนิดหนึ่ง) ซึ่งเราสามารถที่จะสร้างขึ้นมาเองได้นอกเหนือจากที่ระบบฐานข้อมูลมีให้) เพื่อทำการแปลงชนิดของข้อมูลก่อน

2.3.3 ข้อมูลแบบ Complex

เป็นข้อมูลที่ใช้สร้างที่สามารถจะประกอบไปด้วยข้อมูลชนิดต่างๆ ซึ่งคุณสมบัติที่สำคัญของชนิดข้อมูลนี้ก็คือการที่ง่ายที่จะเข้าถึงแต่ละส่วน (component) ของข้อมูลที่ไม่เหมือนกับข้อมูลชนิด Built-in หรือแบบ Opaque ที่มีลักษณะซ่อนเร้นกล่าวคือถ้าเราจะเข้าถึงข้อมูลของ Opaque ก็จะต้องทำผ่านฟังก์ชันที่กำหนดไว้กับ Opaque



รูปที่ 2-3 Complex types ที่มีอยู่ใน Informix Universal Server

จากรูปที่ จะเห็นว่าข้อมูลแบบ Complex ประกอบด้วย

□ **ข้อมูลแบบ Collection:** เป็นกลุ่มของข้อมูลที่มีชนิดข้อมูลเดียวกัน โดยอนุญาตให้เราสามารถที่จะเก็บและจัดการกับ Collection ของข้อมูลภายใน Row เดียว ข้อมูลแบบ Collection ประกอบด้วย 2 ส่วนคือ Type constructor และ Element type โดยที่

1. Type constructor ประกอบด้วย set, multiset และ list

□ SET : เป็นข้อมูลแบบ Collection ที่ Elements ไม่มีลำดับและไม่สามารถที่จะมีข้อมูลซ้ำกันได้ภายใน Set ซึ่งในการกำหนด Set จะต้องใช้ Set constructor

□ LIST : เป็นข้อมูลแบบ Collection ที่ Elements มีลำดับและ Element สามารถที่จะซ้ำกันได้ ในการกำหนด List จะต้องใช้ List constructor

□ MULTISET : เป็นข้อมูลแบบ Collection ที่ไม่มีลำดับของ Element และอนุญาตให้ข้อมูลซ้ำกันได้ ในการกำหนด Multiset จะต้องใช้ Multiset constructor

2. Element type เป็นการระบุชนิดของข้อมูลของ Element ซึ่ง Element สามารถที่จะมีชนิดใดก็ได้ นอกจาก SERIAL และ SERIAL8

□ **Row types** เป็นชนิดข้อมูลที่มีลักษณะคล้ายกับข้อมูลแบบโครงสร้าง (Structure) ในภาษา C โดยภายในจะประกอบด้วยฟิลด์ (Fields) แต่ละฟิลด์จะมีชื่อและชนิดข้อมูลเป็นของตัวเองซึ่งฟิลด์ของ Row

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

type ก็เปรียบได้กับคอลัมน์ของตารางนั่นเองซึ่งจะเห็นว่า Row type ของ Informix Universal Server ก็เป็นชนิดของข้อมูลชนิดเดียวกับ Row type ที่มีอยู่มาตรฐานของ SQL3 โดยที่ Row type ใน Informix Universal Server นั้นมีอยู่ด้วย 2 แบบคือ named row type และ Unnamed row type

1. Named row type : เป็นกลุ่มของฟิลด์ ที่การกำหนดภายใต้ชื่อชื่อหนึ่ง โดยฟิลด์นี้หมายถึงคอมพิวเตอร์ของ Row type โดยในการสร้าง named row type จะต้องมีการตั้งชื่อไม่ซ้ำกับ named row type ที่มีในฐานข้อมูลอยู่ก่อนแล้ว เราจะมีการสร้าง named row type ก็ต่อเมื่อเราต้องการที่จะเข้าถึงคอมพิวเตอร์ของข้อมูลได้เช่นเราอาจจะสร้าง named row type ที่เก็บ address ซึ่งสามารถที่จะเข้าถึงข้อมูลแต่ละคอมพิวเตอร์ของ address ได้เช่น ถนน เมือง จังหวัด เป็นต้น

2. Unnamed row type : เป็นกลุ่มของฟิลด์ที่กำหนดโดยใช้โครงสร้างของตนเองซึ่งไม่เหมือนกับ named row type เราสามารถที่จะกำหนด unnamed row type ให้กับตารางและใช้ในการกำหนดชนิดให้กับคอลัมน์ได้

2.4 การสืบทอดคุณสมบัติในอินฟอร์มิกซ์ ยูนิเวอร์แซล เซิร์ฟเวอร์

การสืบทอดคุณสมบัติใน Informix Universal Server จะทำได้ 2 แบบ คือ

1. การสืบทอดคุณสมบัติสำหรับ named row type
2. การสืบทอดคุณสมบัติสำหรับตาราง

2.4.1 การสืบทอดคุณสมบัติสำหรับ named row type

เมื่อเราสร้างโรว์ไทป์หนึ่งเป็นซูปเปอร์ไทป์ เราสามารถสร้างซับไทป์ของมันได้โดยใช้คำสั่ง

UNDER <ชื่อของ Supertype>

ตัวอย่างเช่น

```
CREATE ROW TYPE EmployeeType
(
    Name char(20),
    Dept integer,
    Salary money
);
CREATE ROW TYPE SalemanType
(
    Quota money,
    Attainment decimal(4,2),
    Commission decimal(4,2),
    Accounts set(varchar(30) not null)
)
UNDER EmployeeType;
```

จากตัวอย่าง เป็นการสร้างให้ EmployeeType เป็นซูปเปอร์ไทป์ของ SalemanType ซับไทป์จะสืบทอดคุณสมบัติทุกๆ ฟิล์ดข้อมูล (Data Field) ของซูปเปอร์ไทป์นั้นคือเราจะมองได้ว่าฟิล์ด name ,dept และ salary ถูกรวมเป็นฟิล์ดข้อมูลของ SalemanType ด้วย

2.4.2 การสืบทอดคุณสมบัติสำหรับตาราง

เมื่อเรา Create Table หนึ่งเป็น SuperTable เราสามารถ Create SubTable ของมัน ได้โดยใช้คำสั่ง

UNDER <ชื่อของ SuperTable>

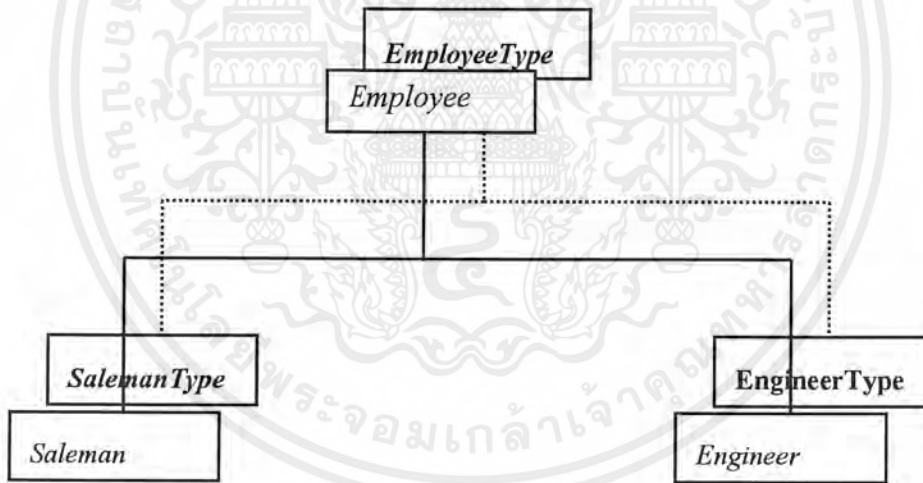
ตัวอย่างเช่น

```
CREATE TABLE Employee OF TYPE EmployeeType;
```

```
CREATE TABLE Engineer OF TYPE EmployeeType UNDER Employee;
```

```
CREATE TABLE Saleman OF TYPE EmployeeType UNDER Employee;
```

SubTable ที่สร้างขึ้นนี้จะสามารถสืบทอดคุณสมบัติต่างๆ ของ SuperTable ได้แก่ คอลัมน์ , ดัชนี (Indices) , ทรริกเกอร์ , การเลือกที่เก็บข้อมูล (Storage Options) และข้อกำหนด (Constraint) ลักษณะความสัมพันธ์ของไทป์และตาราง จะมีลักษณะดังรูป



รูปที่ 2-4 ลักษณะการสืบทอดคุณสมบัติระดับ named row type และตาราง

บทที่ 3

หลักการดาต้าไดเรกเตอร์ออบเจกต์ Data Director Objects

3.1 ดาต้าไดเรกเตอร์ออบเจกต์คืออะไร

ดาต้าไดเรกเตอร์ออบเจกต์เป็นชุดของการโปรแกรมออบเจกต์ต่าง ๆ ที่ซ่อนการทำงานต่าง ๆ ของฐานข้อมูลซึ่งมีความสัมพันธ์กับดาต้าไดเรกเตอร์จียูไอ (Data Director GUI)

3.2 ทำไมจึงใช้ดาต้าไดเรกเตอร์ออบเจกต์

3.2.1 ขยายการทำแอปพลิเคชันต้นแบบ

เราไม่ต้องสร้างออบเจกต์ทั้งหมดในการเขียนโปรแกรม ถ้าเราได้สร้างออบเจกต์เหล่านั้นผ่านทางยูสเซอร์อินเทอร์เฟซ (User interface) แล้ว เราสามารถอ้างถึงออบเจกต์เหล่านั้นอย่างง่าย ๆ ได้ ในการพัฒนาแอปพลิเคชัน โดยส่วนมากจะเริ่มต้นด้วยการใช้ดาต้าไดเรกเตอร์จียูไอ พัฒนาแอปพลิเคชันต้นแบบ และพัฒนาต้นแบบนั้นโดยการ ใช้ดีดีโอ (DDO)

ครั้งแรกที่แอปพลิเคชันทำงานได้และได้ผ่านการทดสอบแล้ว เราสามารถเพิ่มการทำงานที่ซับซ้อนได้โดยการ ใช้ดีดีโอ ฟอรัมการติดต่อกับดีดีโอ (DDO interface form) การทำงานพื้นฐานจะสร้างในจียูไอ (GUI) ดังนั้นการขยายโค้ดจะหมายถึงการเพิ่มประสิทธิภาพมากกว่าการเขียนใหม่ เราจะใช้จียูไอในการสร้างโค้ดออบเจกต์แบบที่มองเห็นได้ (Visual object) ต่าง ๆ ที่จำเป็นลงบนฟอรัม และใช้ดีดีโอในการเขียนโค้ดในส่วนของการติดต่อกับฐานข้อมูล (Database) ทั้งหมด

3.2.2 ประสิทธิภาพ

ในบางกรณีประสิทธิภาพของแอปพลิเคชันสามารถปรับปรุงได้โดยการกำจัดการลิงค์แบบแดร็กและครีอป (Drag-and-drop) ตัวอย่างเช่น สมมุติว่าแอปพลิเคชันของเรามีฟอรัมหนึ่งที่ใช้ต้องการใช้งานเพียงหนึ่งครั้งในหนึ่งเดือน ถ้าเราใช้วิธีแดร็กและครีอปในการสร้างการติดต่อระหว่าง Visual object บนฟอรัมกับ คอลัมน์ต่าง ๆ ในฐานข้อมูล การเชื่อมต่อครั้งแรกจะเกิดเมื่อเราเริ่มใช้แอปพลิเคชันซึ่งอาจจะใช้เวลาไม่มาก ในการให้ข้อมูลที่ใช้ไม่บ่อย ถ้าเราใช้ดีดีโอแทน โค้ดจะถูกประมวลผลเมื่อผู้ใช้ได้ใช้ฟอรัมนั้นซึ่งจะไม่ได้ประมวลผลทุกครั้งเมื่อเราเริ่มใช้ แอปพลิเคชัน

3.2.3 สามารถติดต่อกับซอฟต์แวร์ประเภทอื่น ๆ ได้

การเขียนโปรแกรมด้วยดีดีโอจะอยู่บนพื้นฐานของคอมโพเนนต์ออบเจกต์โมเดล (Component Object Model (COM)) ซึ่งทำให้โค้ดที่เป็นดีดีโอสามารถทำงานกับซอฟต์แวร์ที่สนับสนุนคอม (COM) ได้ ตัวอย่างเช่น เราสามารถเขียนโค้ดที่เป็นดีดีโอในโปรแกรมของเอกเซลล์ (Excel Application) ทำงานกับเอกสารที่ใช้ข้อมูลจากฐานข้อมูลอินฟอर्मิกส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 ความชื่นชอบของผู้พัฒนาโปรแกรม

เหมาะสมกับผู้พัฒนาโปรแกรมที่ชอบเขียนโค้ดเอง เพราะว่าการฟังก์ชันทำงานของ Data Director ทั้งหมดสามารถใช้งาน โดยการเขียนโค้ดผ่านคีโอดี

3.2.5 การจัดการกับข้อมูลชนิดพิเศษสำหรับข้อมูลที่ไม่มีรูปแบบ

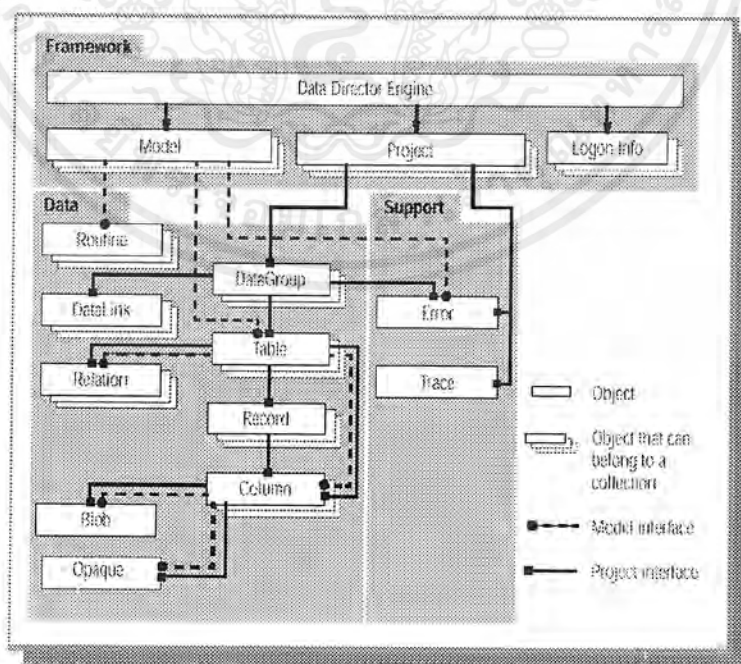
ถ้าข้อมูลที่เก็บในโอเปค (Opaque) นั้นไม่ใช่ข้อมูลที่เป็นตัวอักษร เราต้องใช้คีโอดีในการแสดงค่าของข้อมูลชนิดโอเปคในแอปพลิเคชันของเรา ข้อมูลที่เก็บในโอเปคส่วนมากเป็นข้อมูลที่ไม่มีรูปแบบแน่นอน เช่น เสียง วิดีโอ และซี เราต้องใช้คีโอดีในการทำงานที่แตกต่างกับข้อมูล เช่น การอ่านและการเขียนจากบัฟเฟอร์หรือไฟล์

3.2.6 มีความยืดหยุ่นเป็นอย่างมากในขณะทำงาน

การใช้คีโอดีอนุญาตให้แอปพลิเคชันเปลี่ยนโมเดลขณะรันโปรแกรมได้ แต่เมื่อเราใช้แอปพลิเคชันที่เป็นจียูไอขณะรันโปรแกรม ต้องใช้โมเดลที่สร้างในขณะออกแบบโปรแกรมเท่านั้น

3.3 โครงสร้างลำดับชั้นของคิตโต้เรคเตอร์ออบเจกต์

ไลบรารี (Library) ของคิตโต้ประกอบด้วยออบเจกต์ต่างๆ ที่จะเข้าถึงออบเจกต์อื่น ๆ ในโครงสร้างลำดับชั้นทาง Logical ออบเจกต์ที่เป็นเจ้าของออบเจกต์อื่น ๆ เรียกออบเจกต์นั้นว่า ออบเจกต์พ่อแม่ (Parent object) ส่วนออบเจกต์ที่ถูกควบคุมเรียกว่า ออบเจกต์ลูก (Child object) ออบเจกต์แต่ละออบเจกต์สามารถมีพ่อแม่ได้มากกว่า 1 ออบเจกต์ตัวอย่างเช่น ออบเจกต์โมเดล (Model object), ออบเจกต์โปรเจกต์ (Project object) และออบเจกต์คิตโต้กรุป (DataGroup object) เป็นพ่อแม่ของออบเจกต์ความผิดพลาด (Error object)



รูปที่ 3-1 โครงสร้างลำดับชั้นของคิตโต้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ออบเจกต์และคอลเลกชัน (Objects and Collections)

คอลเลกชันหนึ่ง ๆ ประกอบไปด้วยหลาย ๆ ออบเจกต์เช่น ตาราง, โปรเจก และ โมเดล ตัวอย่างเช่น ออบเจกต์ค่าตัวกรุป สามารถเป็นสมาชิกของค่าตัวกรุปคอลเลกชัน (DataGroup collection) ซึ่งค่าตัวกรุปคอลเลกชันนี้ ประกอบด้วยหลาย ๆ ออบเจกต์ค่าตัวกรุป คอลเลกชันหนึ่ง ๆ จะมีกลุ่มของคุณสมบัติและเมทร็อคเฉพาะและแตกต่างจากสมาชิกต่าง ๆ ใน คอลเลกชัน ดังนั้นคอลเลกชันต่าง ๆ จะไม่มีเมทร็อคและคุณสมบัติที่เหมือนกัน

คุณสมบัติของคอลเลกชัน

- สามารถแก้ไขหรือดึงข้อมูลแต่ละออบเจกต์ในคอลเลกชันได้
- สามารถดึงแต่ละออบเจกต์ใน คอลเลกชันได้
- สามารถนับจำนวนของแต่ละออบเจกต์ในคอลเลกชันได้

ออบเจกต์ที่สามารถเป็น คอลเลกชัน ได้ ได้แก่ เอนจิน (Engine), โปรเจก, โมเดล, ค่าตัวกรุป, ตาราง, ความสัมพันธ์, คำสั่ง (Routines) และเรคคอร์ด คอลเลกชันต่าง ๆ เป็นเจ้าของแต่ละออบเจกต์ที่รวมกันเป็นคอลเลกชัน ตัวอย่างเช่น ตารางคอลเลกชันเป็นเจ้าของออบเจกต์ตาราง และเรคคอร์ดคอลเลกชันเป็นเจ้าของออบเจกต์เรคคอร์ด

3.5 ชนิดของออบเจกต์

ค่าตัวไคเรคเตอร์ออบเจกต์ แบ่งเป็น 3 ประเภทดังนี้

- ออบเจกต์เฟรมเวิร์ค (Framework objects)
- ออบเจกต์ข้อมูล (Data objects)
- ออบเจกต์สนับสนุน (Support objects)

3.5.1 ออบเจกต์เฟรมเวิร์ค (Framework Objects)

แต่ละแอปพลิเคชันที่เราสร้างด้วยค่าตัวไคเรคเตอร์และดีดีไอจะมีการจัดการและโครงสร้างแบบนามธรรม (Logical) ฐานข้อมูลที่เราทำงานด้วยนั้นก็มีการจัดการแบบนามธรรมของมันเอง ออบเจกต์เฟรมเวิร์คจะแสดงลักษณะ โครงสร้างเหล่านี้และมีการจัดการเข้าถึงออบเจกต์อื่น ๆ ที่สร้างขึ้นโดยแอปพลิเคชัน และจัดเตรียมข้อมูลที่แอปพลิเคชันของเราจะเข้าถึง

3.5.1.1 เอนจิน (Engine)

เอนจิน เป็นออบเจกต์ต้น (Root Object) เราจะต้องสร้างออบเจกต์นี้ก่อนที่จะสร้างค่าตัวไคเรคเตอร์ออบเจกต์อื่น ๆ โดยออบเจกต์เอนจิน (Engine object) นี้เราสามารถควบคุมโปรเจก และออบเจกต์โมเดลได้ เราสามารถสร้าง โปรเจก และโมเดลได้หลายอันในออบเจกต์เอนจินเพียงอันเดียว เนื่องจากการทำงานของวิซวลเบสิกหรือค่าตัวไคเรคเตอร์ จะอนุญาตให้มีเอนจินที่รัน ได้เพียงเอนจินเดียว

สิ่งที่เราสามารถกระทำกับ เอนจินได้ มีดังนี้

- สร้างออบเจกต์โปรเจก
- สร้างออบเจกต์โมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

□ สร้างออบเจกต์ลือคอินโฟ

ออบเจกต์ อื่น ๆ ไม่สามารถเป็นเจ้าของ ออบเจกต์เอนจิน ได้ ออบเจกต์เอนจิน เป็นเจ้าของ โปรเจค , โมเดลและออบเจกต์ลือคอินโฟ

3.5.1.1.1 โปรเจค (Project)

ออบเจกต์โปรเจค จะจัดการกลุ่มของคาค้ากรู๊ป ตาราง คอลัมน์ และออบเจกต์ความสัมพันธ์ (Relation object) เหมือนออบเจกต์อื่น ๆ ที่ใช้ในแอปพลิเคชันของเรา ออบเจกต์โปรเจคจะอยู่ในรูปของ ไฟล์โปรเจคของคาค้าไคเรคเตอร์ (.ddx)

สิ่งที่เราสามารถกระทำกับออบเจกต์โปรเจค ได้มีดังนี้

- สร้างออบเจกต์คาค้ากรู๊ป
- เขตให้มีการทำงาน หรือไม่มีการทำงานของการตรวจจับความผิดพลาด (Handling Error) ขณะรันโปรแกรม

ออบเจกต์เอนจิน เป็นเจ้าของออบเจกต์โปรเจค ออบเจกต์โปรเจค เป็นเจ้าของออบเจกต์คาค้ากรู๊ป ออบเจกต์ความสัมพันธ์ และ ออบเจกต์เทรซ (Trace object)

3.5.1.1.2 โมเดล (Model)

ออบเจกต์โมเดล จะทำหน้าที่เข้าถึงและปรับปรุง (Update) ข้อมูลของโมเดลของคาค้าไคเรคเตอร์ (Data Director Model) ซึ่งอยู่ในรูปของไฟล์โมเดล (.mlt หรือ .mlx)

สิ่งที่เราสามารถกระทำกับออบเจกต์โมเดล ได้มีดังนี้

- อิมพอร์ตโครงสร้างของฐานข้อมูลไปยังโมเดล
- คัดลอกและจัดเก็บโมเดล
- สร้างออบเจกต์ตาราง
- สร้าง LogonInfo object

ออบเจกต์เอนจิน เป็นเจ้าของออบเจกต์โมเดล ออบเจกต์โมเดล เป็นเจ้าของ ออบเจกต์ตาราง, ออบเจกต์รูทีน และ ออบเจกต์ความสัมพันธ์

3.5.1.1.3 ลือคออนอินโฟ (LogonInfo)

ลือคออนอินโฟ จะทำการระบุข้อมูลเกี่ยวกับ ชนิดของฐานข้อมูล, ชื่อผู้ใช้, รหัสผ่าน, หมายเลขเซิร์ฟเวอร์ และ ชื่อฐานข้อมูลเมื่อเรากำลึงลือคอิน ไปยังฐานข้อมูล

สิ่งที่เราสามารถกระทำกับออบเจกต์ลือคออนอินโฟ ได้มีดังนี้

- ลือคออน (Log on) ไปยังฐานข้อมูล
- อิมพอร์ตฐานข้อมูลไปยังโมเดล

ออบเจกต์เอนจิน เป็นเจ้าของออบเจกต์ลือคออนอินโฟ ออบเจกต์ลือคออนอินโฟ ไม่สามารถเป็นเจ้าของออบเจกต์อื่น ๆ ได้

3.5.2 ออบเจ็กต์ข้อมูล (Data Objects)

แต่ละฐานข้อมูลที่เราทำงานด้วยจะประกอบด้วยส่วนต่าง ๆ ที่แน่นอน เช่น ตาราง, คอลัมน์ หรือ แถว ออบเจ็กต์ข้อมูล ซึ่งทำให้เราสามารถทำงานกับข้อมูลต่าง ๆ ของฐานข้อมูลเหล่านี้ได้

3.5.2.1 คาดำกรุป (DataGroup)

ออบเจ็กต์คาดำกรุป มีความสอดคล้องกับคาดำไคเรคเตอร์ คาดำกรุป เป็นกลุ่มของความสัมพันธ์ของ ตาราง ในฐานข้อมูลและคาดำลิงค์

สิ่งที่เราสามารถกระทำกับ ออบเจ็กต์คาดำกรุป ได้มีดังนี้

- สร้างออบเจ็กต์ลือคออนอินโฟ
- สร้างออบเจ็กต์คาดำลิงค์
- สร้างออบเจ็กต์รูทีน (Routine object)
- ลือคอินแและลือคออพจากฐานข้อมูล
- ตั้งค่าออบชั่นเกี่ยวกับการรายงานความผิดพลาด
- ประมวลผลคำสั่งเอสคิวแอล (SQL)
- ตั้งค่าออบชั่นของระดับการลือคเรคคอด (Record_Level Locking)

ออบเจ็กต์โปรเจค เป็นเจ้าของออบเจ็กต์คาดำกรุป ออบเจ็กต์คาดำกรุป เป็นเจ้าของออบเจ็กต์คาดำลิงค์ ออบเจ็กต์ตาราง และ ออบเจ็กต์ความผิดพลาด

3.5.2.2 คาดำลิงค์(DataLink)

ออบเจ็กต์คาดำลิงค์ มีความสอดคล้องกับคาดำลิงค์ของคาดำไคเรคเตอร์ คาดำลิงค์ เป็นการติดต่อระหว่างคอลัมน์ของฐานข้อมูลกับคอนโทรลของวิซวลเบสิก คาดำลิงค์ นี้จะเป็นตัวกำหนดว่าข้อมูลใดจะปรากฏอยู่ในคอนโทรลบนฟอร์มของแอปพลิเคชัน

สิ่งที่เราสามารถกระทำกับออบเจ็กต์คาดำลิงค์ ได้มีดังนี้

- ลือคลือกคาดำลิงค์
- ตั้งค่าคุณสมบัติของ คาดำลิงค์
- ระบุว่าคาดำลิงค์เป็นคาดำลิงค์จริง หรือเป็นคาดำลิงค์เสมือน (Virtual DataLink)

ออบเจ็กต์คาดำกรุป เป็นเจ้าของ ออบเจ็กต์คาดำลิงค์ ออบเจ็กต์คาดำลิงค์ ไม่สามารถเป็นเจ้าของออบเจ็กต์อื่น ๆ ได้

3.5.2.3 ตาราง

ออบเจ็กต์ตาราง มีความสอดคล้องกับตารางในฐานข้อมูล ตารางในฐานข้อมูลประกอบไปด้วย คอลัมน์ และแถว

สิ่งที่เราสามารถกระทำกับออบเจ็กต์ตาราง ได้มีดังนี้

- สร้างออบเจ็กต์ความสัมพันธ์
- สร้างออบเจ็กต์คอลัมน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สร้างออบเจกต์เรคคอด
- สามารถเดินทางโดยผ่านเรคคอดในเซตของผลลัพธ์ได้
- สามารถเดินทางโดยเรคคอดที่ได้ทำการบู๊คมาร์ก ได้
- กำหนดและประมวลผลฟังก์ชันของสตอร์โปรซีเจอร์ (Stored procedure Function) และ Routine ที่ผู้ใช้สร้างขึ้นใหม่ (User-define routines) ได้
- ตั้งค่าตัวแปรในคำสั่งเอสคิวแอล

ดาต้ากรุป และ ออบเจกต์โมเดล เป็นเจ้าของ ออบเจกต์ตาราง ออบเจกต์ตาราง เป็นเจ้าของ คอลัมน์ และ Relation ได้หลาย ๆ อัน ถ้า ตาราง เป็นส่วนหนึ่งของ ดาต้ากรุป มันจะเป็นเจ้าของ เรคคอดคอลเลกชัน ได้หลาย ๆ อันด้วย

3.5.2.4 คอลัมน์

ออบเจกต์คอลัมน์ มีความสอดคล้องกับ คอลัมน์ ในฐานข้อมูล คอลัมน์ ในฐานข้อมูลเป็นหน่วยในการจัดการขั้นพื้นฐานของฐานข้อมูล คอลัมน์ จะประกอบด้วยข้อมูลปกติทั่ว ๆ ไป เช่น นามสกุล ชื่อ ที่อยู่ และ หมายเลขโทรศัพท์

สิ่งที่เราสามารถกระทำกับ ออบเจกต์คอลัมน์ ได้มีดังนี้

- เข้าถึงออบเจกต์ที่เป็นข้อมูลแบบไบนารีขนาดใหญ่ (Binary large object (BLOB)) จากฐานข้อมูล ได้
- เข้าถึงชนิดข้อมูลที่ผู้ใช้สร้างขึ้นใหม่ (โอเปค) จากฐานข้อมูลได้
- กำหนดการดึงและแก้ไขข้อมูลและแคชชิ่ง (Caching) สำหรับข้อมูลแบบบล็อบ (BLOB) ได้
- กำหนดชนิดข้อมูลในคอลัมน์, ขนาด และ เมทโธดการเรียงลำดับ (Sorting method) ได้
- กำหนดคีย์หลัก (Primary key) ให้ คอลัมน์ ได้

ตาราง และ ออบเจกต์เรคคอด เป็นเจ้าของ ออบเจกต์คอลัมน์ ออบเจกต์คอลัมน์ เป็นเจ้าของ ออบเจกต์บล็อบ และ ออบเจกต์โอเปค

3.5.2.5 เรคคอด (Record)

ออบเจกต์เรคคอด มีความสอดคล้องกับเรคคอด ในฐานข้อมูล เรคคอด ในฐานข้อมูลจะบรรจุข้อมูลเฉพาะซึ่งถูกกำหนด โดย คอลัมน์ ให้แยกกันเป็นส่วน ๆ

สิ่งที่เราสามารถกระทำกับ ออบเจกต์เรคคอด ได้มีดังนี้

- คัดลอกเรคคอด
- ล้างค่าแฟลก (Flag) ต่าง ๆ จากเรคคอด
- รีเซตค่าของ เรคคอด ให้กลับไปอย่างเดิม
- การล๊อคเรคคอด ขณะที่กำลังถูกเปลี่ยนแปลง

ออบเจกต์ตาราง เป็นเจ้าของ ออบเจกต์เรคคอด

ออบเจกต์เรคคอด เป็นเจ้าของ ออบเจกต์คอลัมน์

3.5.2.6 ความสัมพันธ์ (Relation)

ออบเจกต์ความสัมพันธ์ ใช้อธิบายความสัมพันธ์ระหว่างตาราง 2 ตาราง

สิ่งที่เราสามารถกระทำกับ ออบเจกต์ความสัมพันธ์ ได้มีดังนี้

- ซึ่ไปที่จุดเริ่มต้นและจุดหมายใน ตาราง ที่เราต้องการสร้างความสัมพันธ์ (Relationship)
- ซึ่ไปที่จุดเริ่มต้นและจุดหมายใน คอลัมน์ ที่เราต้องการสร้างความสัมพันธ์
- เพิ่มจุดเริ่มต้นและจุดหมายของ คอลัมน์ ให้เป็นความสัมพันธ์
- ซึ่ชนิดของความสัมพันธ์

ออบเจกต์ตาราง เป็นเจ้าของ ออบเจกต์ความสัมพันธ์ ออบเจกต์ความสัมพันธ์ ไม่สามารถเป็น เจ้าของออบเจกต์อื่น ๆ ได้

3.5.2.7 บลอบ (Blob)

ออบเจกต์บลอบ ใช้แสดงออบเจกต์ที่เป็นข้อมูลแบบไบนารีขนาดใหญ่ โดยปกติแล้วค่าที่เก็บใน บลอบ จะใช้เก็บข้อมูลที่มีขนาดใหญ่ เช่น รูปภาพ ซึ่งเราต้องการที่จะเข้าถึงข้อมูลนั้นเป็นหน่วย ๆ เดียว บลอบ มีความสัมพันธ์กับคอลัมน์และเรคคอด ขณะที่เราเคลื่อนย้ายระหว่างเรคคอดกับเรคคอด ออบเจกต์บลอบ จะซึ่ไปยังค่าบลอบสำหรับ เรคคอด นั้น ๆ

สิ่งที่เราสามารถกระทำกับ ออบเจกต์บลอบ ได้มีดังนี้

- ดึงข้อมูลบลอบได้
- กำหนดค่าข้อมูลไปยังออบเจกต์บลอบ ได้
- บันทึกลงและโหลดออบเจกต์บลอบ ไปมาระหว่าง ไฟล์
- นำข้อมูลแบบบลอบออกมา

ออบเจกต์คอลัมน์ เป็นเจ้าของออบเจกต์บลอบ ออบเจกต์บลอบ ไม่สามารถเป็นเจ้าของ ออบเจกต์อื่น ๆ ได้

3.5.2.8 โอเปก (Opaque)

ออบเจกต์โอเปก แสดงชนิดข้อมูลที่ผู้ใช้สร้างขึ้นมาในอินฟอร์มิทซ์ที่เป็นออปชันของข้อมูลของ อินฟอร์มิทซ์ ชนิดข้อมูลที่ผู้ใช้สร้างขึ้นมา เป็นชนิดข้อมูลที่สร้างโดยผู้เขียน โปรแกรมเพื่อใช้แสดงข้อมูล ชนิดพิเศษ เช่น Spatial หรือ ข้อมูลเกี่ยวกับเวลา (Time-series)

ออบเจกต์โอเปก มีความสัมพันธ์กับคอลัมน์ และเรคคอด ถ้าเราเคลื่อนย้ายข้อมูลระหว่างเรคคอด กับเรคคอด ออบเจกต์โอเปก จะมีความสัมพันธ์กับค่าข้อมูลสำหรับแต่ละเรคคอดที่เกิดขึ้น

สิ่งที่เราสามารถกระทำกับออบเจกต์โอเปก ได้มีดังนี้

- แก้ไขข้อมูลที่ผู้ใช้กำหนดขึ้นมา
- มีการบันทึกและดึงข้อมูลออบเจกต์โอเปกไปมาระหว่างบัฟเฟอร์ของแอปพลิเคชัน
- มีการบันทึกและดึงข้อมูลออบเจกต์โอเปกไปมาระหว่าง ไฟล์ต่าง ๆ
- ทดสอบว่าออบเจกต์โอเปกมีความยาวจำกัดหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

□ ทดสอบว่าออบเจกต์โอเปกไม่มีค่าหรือไม่

□ เปรียบเทียบออบเจกต์ 2 ออบเจกต์ที่เห็น ถ้าเรามีข้อมูลแบบโอเปกที่เหมือนกัน

ออบเจกต์คอลลัมน์ เป็นเจ้าของ ออบเจกต์โอเปก ออบเจกต์โอเปก ไม่สามารถเป็นเจ้าของออบเจกต์อื่น ๆ ได้

สิ่งสำคัญ : ถ้า ออบเจกต์โอเปก ถูกสร้างจากผลของการเรียกใช้รูทีนของเซิร์ฟเวอร์ผ่านทางออบเจกต์รูทีน ทำให้ ออบเจกต์คอลลัมน์ เป็นเจ้าของ ออบเจกต์โอเปก แทนที่ ออบเจกต์คอลลัมน์

3.5.2.9 รูทีน (Routine)

ออบเจกต์รูทีน ใช้แสดงฟังก์ชันทุก ๆ ชนิดของผู้ใช้หรือฟังก์ชันที่ระบบกำหนดหรือสครอโปรซีเจอร์ในฐานข้อมูล การใช้ ออบเจกต์รูทีน เราสามารถเพิ่มสครอโปรซีเจอร์และรูทีนที่ผู้ใช้สร้างขึ้น จากฐานข้อมูลไปยังออบเจกต์โมเดล ของเราได้ ออบเจกต์โมเดล เป็นเจ้าของ ออบเจกต์รูทีน

สิ่งสำคัญ : เราจะต้องได้รับออบเจกต์รูทีน จากออบเจกต์คอลลัมน์ ไม่ใช่ได้มาจากออบเจกต์โมเดล ในการเรียกใช้รูทีน ออบเจกต์รูทีน ไม่สามารถเป็นเจ้าของออบเจกต์อื่น ๆ ได้

3.5.3 ออบเจกต์ช่วยเหลือ (Support Object)

แต่ละโปรเจกที่เราทำงานด้วยโดยใช้ คาด้าไคเรคเตอร์ ต้องการการแสดงผลฟังก์ชันเข้าคิบป์ (Housekeeping) ที่แน่นอน เช่น การเก็บเส้นทางของการเกิดความผิดพลาดและการเก็บที่อยู่ของไฟล์ของออบเจกต์ช่วยเหลือ ช่วยในการทำงานฟังก์ชันเหล่านี้

3.5.3.1 ความผิดพลาด

□ ออบเจกต์ความผิดพลาด ใช้ออก ชนิด, เวลา และข้อความของความผิดพลาด

□ โปรเจก, โมเดลและออบเจกต์คอลลัมน์เป็นเจ้าของ ออบเจกต์ความผิดพลาด

□ ออบเจกต์ความผิดพลาด ไม่สามารถเป็นเจ้าของ Object อื่น ๆ ได้

3.5.3.2 การติดตาม (Trace)

ออบเจกต์ติดตาม (Trace object) ตรงกับไฟล์ติดตามของคาด้าไคเรคเตอร์ ไฟล์ติดตาม เป็นสื่อคือไฟล์ ที่ใช้เก็บคำสั่งที่คาด้าไคเรคเตอร์ทำการประมวลผล

สิ่งที่เราสามารถกระทำกับ ออบเจกต์ติดตาม ได้มีดังนี้

□ เปิดหรือปิดการติดตาม

□ เพิ่มข้อมูลไปยังไฟล์ติดตาม

ออบเจกต์โปรเจก เป็นเจ้าของออบเจกต์ติดตาม ออบเจกต์ติดตาม ไม่สามารถเป็นเจ้าของออบเจกต์อื่น ๆ ได้

3.6 การใช้ คาด้าไคเรคเตอร์ออบเจกต์

ในส่วนนี้จะอธิบายวิธีการใช้การติดต่อกับคิไอ (DDO interface) ในการโปรแกรมในหัวข้อที่สำคัญดังนี้

- การใช้คุณสมบัติ และเมธอด
- การเริ่มต้นการทำงานคิโอดี
- การลึคออนไปยังฐานข้อมูล
- การทำงานกับข้อมูล

ตามแบบการตั้งชื่อของอินชแทนซ์ของออบเจ็กต์ในวิหวลเบสิก ในโค้ดตัวอย่างจะมีอักษร o อยู่ด้านหน้าชื่อตัวแปรเหล่านั้นในส่วนนี้ ตัวอย่าง ออบเจ็กต์ค่าตัวกรุป ถูกอ้างถึงด้วยชื่อ oDataGroup ในโค้ดตัวอย่าง

3.6.1 การใช้ คุณสมบัติ และ เมธอด

ข้อดีของคิโอดี คือการช่อนการเข้าถึงข้อมูลในเมธอดและคุณสมบัติของออบเจ็กต์เราสามารถ ใช้เมธอดและ คุณสมบัติ เหล่านี้ได้ง่าย ๆ โดยการอ้างถึงสิ่งเหล่านั้น

ครั้งหนึ่งที่เราได้สร้างออบเจ็กต์พื้นฐานโดยใช้เมธอดของคิโอดีและคุณสมบัติของคิโอดี จะเป็นส่วนที่ติดตามไปในโค้ดของเรา แต่ละค่าไคเรคเตอร์ออบเจ็กต์มีเมธอดและคุณสมบัติ ที่เราสามารถจัดการหรือเรียกขึ้นมาใช้งานได้

3.6.1.1 การใช้ คุณสมบัติ

การใช้คุณสมบัติของออบเจ็กต์เราจะใช้ชื่อออบเจ็กต์และชื่อคุณสมบัติ เป็น โครงสร้างของรูปแบบคำสั่ง โดยมีรูปแบบทั่ว ๆ ไปดังนี้ คุณสมบัติ จะไม่มีตัวแปรแต่สามารถตั้งค่าที่เหมาะสมโดยใช้ตามรูปแบบคำสั่งดังนี้

```
ObjectName.PropertyName = Value
```

3.6.1.2 การใช้เมธอด

การใช้เมธอดของออบเจ็กต์เราจะใช้ชื่อออบเจ็กต์ ชื่อเมธอดและตัวแปรของเมธอดเป็น โครงสร้างของรูปแบบคำสั่ง โดยมีรูปแบบทั่ว ๆ ไป ดังนี้

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

ถ้าเมธอดไม่มีตัวแปร เราสามารถตัดตัวแปรออกได้ดังนี้

```
ObjectName.MethodName
```

3.6.1.3 ประโยชน์ของคุณสมบัติและเมธอด

มีคุณสมบัติและเมธอดมากมายที่หาได้ในแต่ละออบเจ็กต์แต่มีเมธอดและคุณสมบัติ บางอันที่เราใช้ประโยชน์เป็นพิเศษได้แก่

- คุณสมบัติของพ่อแม่ (Parent property) แต่ละค่าไคเรคเตอร์ออบเจ็กต์มี คุณสมบัติของพ่อแม่ ทำให้เราเคลื่อนย้ายลำดับชั้นของคิโอดี โดยไม่ต้องมีการอ้างถึงที่แน่นอน ตัวอย่าง

```
ObjectSource.[[parentObjectName]...].ObjectDestination.PropertyMethodName
```

คุณสมบัติของพ่อแม่ สามารถเรียกไปได้จนกระทั่งถึงส่วนบนสุดของคิโอดี(ออบเจ็กต์เอนจิน) สำหรับแอปพลิเคชันปัจจุบัน

- คุณสมบัติของแอปพลิเคชันของแต่ละค่าไคเรคเตอร์ออบเจ็กต์มีคุณสมบัติของแอปพลิเคชัน ซึ่งใช้ในการส่งค่าการอ้างอิงไปยัง ออบเจ็กต์เอนจิน สำหรับแอปพลิเคชันปัจจุบัน การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อ้างอิงแบบนี้ทำให้เราเขียนโปรแกรมโดยติดต่อโดยตรงกับ ออบเจกต์เอนจิน โดยไม่ต้องเรียกหลาย ๆ ครั้งผ่านทาง คุณสมบัติของพ่อแม่

- คำนวณจำนวนตารางหรือคอลัมน์เพื่อใช้นับจำนวนของตารางในดาต้ากรุป โดยใช้โค้ด ดังนี้


```
oDataGroup.Tables.Count
```

เพื่อใช้นับจำนวนของ คอลัมน์ ในตาราง ใช้โค้ดต่อไปนี้

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

- คำนวณชื่อของตาราง ใช้ในการบอกชื่อของตารางจากคอลัมน์ของดาต้ากรุป โดยใช้โค้ดต่อไปนี้

```
oDataGroup.Tables(1).Name
```

เมทอด **Clone** ใช้ในการสร้างออบเจกต์ที่เหมือนกับออบเจกต์ปัจจุบันรวมถึงส่วนประกอบทุกอย่างของออบเจกต์นั้น ๆ เมทอด **โคลน** มีอยู่ในโมเดล และออบเจกต์ลือคออนอินโฟ

3.6.2 การเริ่มต้นการทำงานกับดีดีโอ

ก่อนที่เราสามารถจะใช้ ดาต้าไดเรกเตอร์ออบเจกต์ได้ เราจะต้องสร้างอินชแทนซ์ของออบเจกต์ก่อน มีขั้นตอน 2 ขั้นตอนในการเริ่มต้นใช้ดีดีโอ

1. สร้าง ออบเจกต์เอนจิน
2. สร้าง ออบเจกต์โปรเจก

3.6.2.1 ขั้นตอนที่ 1 สร้าง ออบเจกต์เอนจิน

ขั้นแรกในการใช้งานดีดีโอคือการสร้าง ออบเจกต์เอนจิน ขั้นตอนนี้เป็นขั้นตอนที่จำเป็นเพราะว่าออบเจกต์เอนจิน เป็นออบเจกต์ระดับบนสุดในโครงสร้างลำดับชั้นของดีดีโอ ดาต้าไดเรกเตอร์ออบเจกต์อื่น ๆ ทั้งหมดจะต้องถูกติดต่อผ่านออบเจกต์เอนจิน

ในการสร้าง ออบเจกต์เอนจิน ใช้รูปแบบคำสั่งดังนี้

```
Dim oEngine As New ddoEngine
```

```
Set oEngine = CreateObject("DataDirector.Engine")
```

```
Set oEngine = New ddoEngine
```

โค้ดนี้ได้ประกาศตัวแปร oEngine และกำหนดให้เป็นอินชแทนซ์ของดีดีโอออบเจกต์เอนจิน

3.6.2.2 ขั้นตอนที่ 2 สร้าง ออบเจกต์โปรเจก

หลังจากเราสร้างออบเจกต์เอนจินแล้ว ต่อมาเราก็สร้างออบเจกต์โปรเจก ขั้นตอนนี้เป็นขั้นตอนที่จำเป็นเพราะว่าทุก ๆ ออบเจกต์ในโครงสร้างลำดับชั้นของดีดีโอจะต้องติดต่อผ่านออบเจกต์โปรเจก ยกเว้นออบเจกต์โมเดล และออบเจกต์ลือคออนอินโฟ

ใช้ เมทอด **CreateProject** ของออบเจกต์เอนจิน ในการสร้างออบเจกต์โปรเจก ดังนี้

```
Dim oProject As ddoProject
```

```
Set oProject = oEngine.CreateProject
```

oProject.Name = "Virtual Library"

โค้ดนี้มีการสร้าง ออบเจกต์โปรเจกต์ ที่มีชื่อว่า Virtual Library

สิ่งสำคัญ: แต่ละครั้งที่แอปพลิเคชันเรียก **CreateProject** คำสั่งไคเรคเตอร์ จะสร้างออบเจกต์โปรเจกต์ใหม่ โดยจะไม่อ้างอิงถึงโปรเจกต์แรกที่ได้สร้างมาก่อน คำสั่งไคเรคเตอร์จะเริ่มต้นใช้งานแต่ละโปรเจกต์ใหม่ในเทรด (Thread) ของมันเองเพื่อความคล่องตัว

3.6.2.3 ขั้นตอนที่ 3 การสร้างแม่เจกต์อื่น ๆ

หลังจากเราสร้างคำสั่งไคเรคเตอร์ออบเจกต์ขั้นพื้นฐานมาแล้ว เราสามารถสร้างออบเจกต์ต่าง ๆ ในลำดับขั้นของดีดีไอได้ แต่ละออบเจกต์ที่ถูกสร้างจะใช้เมททอดจากออบเจกต์พ่อแม่ของมัน ตัวอย่างเช่น เราสร้าง ออบเจกต์คำคำกรุป โดยใช้เมททอด **CreateDataGroup** ของออบเจกต์โปรเจกต์ ซึ่งเป็นพ่อแม่ของ ออบเจกต์คำคำกรุป ทำนองเดียวกันเราสร้างคำสั่งลิงค์ โดยใช้เมททอด **CreateLink** ของ ออบเจกต์คำคำกรุป

3.6.2.3.1 การสร้างออบเจกต์โมเดล

หลังจากเราสร้างออบเจกต์เอนจินแล้ว เราสามารถสร้างออบเจกต์โมเดล เพื่อใช้เข้าถึงตารางและคอลัมน์ในฐานข้อมูลของเราได้ ตัวอย่างเช่น เราใช้ข้อมูลที่เก็บในโมเดลเพื่อสร้างคำสั่งลิงค์ระหว่างคอลัมน์ ในฐานข้อมูลของเรากับคอนโทรลในวิซวลเบสิก

เราใช้เมททอด **CreateModel** ของออบเจกต์เอนจิน ในการสร้างออบเจกต์โมเดล ดังแสดงในตัวอย่างต่อไปนี้

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

โค้ดนี้ใช้สร้างออบเจกต์โมเดล บนไฟล์โมเดลที่มีอยู่แล้วที่ชื่อ library.mlt ไฟล์โมเดล นี้ถูกสร้างโดยการอิมพอร์ตโมเดล (Model Import Wizard) โดยผ่านทางกริดติดต่อผ่านผู้ใช้ของคำสั่งไคเรคเตอร์

3.6.2.3.2 การสร้างออบเจกต์คำคำกรุป

หลังจากเราสร้างออบเจกต์โปรเจกต์ เราสามารถสร้าง ออบเจกต์คำคำกรุป เพื่อเข้าถึงและจัดการกับข้อมูลในฐานข้อมูลของเราขณะ Runtime ได้

เราใช้เมททอด **CreateDataGroup** ของออบเจกต์โปรเจกต์ ในการสร้างออบเจกต์คำคำกรุป ดังแสดงในตัวอย่างด้านล่างนี้

```
Dim oDataGroup As ddoDataGroup
Set oDataGroup = oProject.CreateDataGroup ("book", "DGBook", "c:\Program
Files\Microsoft Visual Basic\library.mlt")
```

ตัวอย่างโค้ดนี้จะเป็นการสร้าง ออบเจกต์ค่าตัวกรุป ที่ชื่อ DGBook ซึ่งอยู่บนไฟล์โมเดล ชื่อ library.mlt มันเป็นรายละเอียดเกี่ยวกับตาราง Book ขณะเป็น ตารางหลัก สำหรับ ค่าตัวกรุป นี้

3.6.3 การ ล็อคออนไปยังฐานข้อมูล

ในส่วนนี้จะแสดง โค้ดที่สาธิตการใช้ ออบเจกต์ ล็อคออนอินโฟ และการ ล็อคออนไปยังฐานข้อมูล โดยผ่านทาง โมเดลหรือออบเจกต์ค่าตัวกรุป สำหรับออบเจกต์โมเดล เราจะใช้ข้อมูลชนิดพิเศษในการ ล็อคออนเพื่อนำโครงสร้างของฐานข้อมูลไปเก็บในโมเดลขณะรันโปรแกรม เราจะใช้ข้อมูลชนิดพิเศษในการ ล็อคออนสำหรับ ออบเจกต์ค่าตัวกรุป เพื่อเข้าถึงและจัดการกับข้อมูลในฐานข้อมูลของเราผ่านทางค่าตัวกรุป

วิธีต่าง ๆ ในการ ล็อคออน ไปยังฐานข้อมูล

- ตั้งค่าตัวแปรล็อคออน โดยการสร้างออบเจกต์ล็อคออนอินโฟ
- ใช้วิธีอิมพอร์ตของ ออบเจกต์โมเดล
- ใช้วิธีล็อคออนของ ออบเจกต์ค่าตัวกรุป หรือโมเดล โดยการใช้ออบเจกต์ล็อคออนอินโฟ หรือ โดยการส่งค่าตัวแปรเฉพาะ โดยตรงใน ล็อคออน

3.6.3.1 การสร้างออบเจกต์ล็อคออนอินโฟ

ในการอธิบายข้อมูลในการ ล็อคออน สำหรับฐานข้อมูลทำได้โดยใช้เมธอด **CreateLogonInfo** ของ โมเดล หรือ ออบเจกต์ค่าตัวกรุป เพื่อตั้งค่าของออบเจกต์ล็อคออนอินโฟ

สิ่งสำคัญเล็ก ๆ น้อยที่เราต้องเก็บเอาไว้ในใจเพื่อพิจารณาเมื่อเราใช้ออบเจกต์ล็อคออนอินโฟ

- ถ้าเราล็อคออนไปยังฐานข้อมูลโดยกช ใช้ออบเจกต์ค่าตัวกรุป และไม่ได้ส่งข้อมูลเกี่ยวกับการ ล็อคออนหรือไม่ได้ส่งค่าตัวแปรทั้งหมดสำหรับออบเจกต์ล็อคออนอินโฟ ทำให้ค่าตัวใดเรคเตอร์ ใช้ตัวแปรที่เหมือนกันสำหรับค่าตัวกรุป ที่เราเคยใช้ในออบเจกต์ล็อคออนอินโฟ ตอนนำโมเดลเข้ามา
- ถ้าเราใช้ค่าข้อมูลที่ดึงมาจากโมเดลออบเจกต์ล็อคออนอินโฟ เมื่อทำการล็อคออนไปยังค่าตัวกรุป ทำให้ ค่าตัวใดเรคเตอร์ไม่ได้จัดเตรียมรหัสผ่านใช้เป็นค่าตั้งต้น
- เพียงครั้งเดียวที่เราได้ส่งข้อมูลในการล็อคออนให้ค่าตัวกรุป ค่าตัวใดเรคเตอร์จะใช้ค่านี้ทุกครั้งเป็นค่าตั้งต้นสำหรับการล็อคออนครั้งต่อไป เราสามารถตั้งค่าตั้งต้นได้ใหม่โดยการส่งตัวแปรค่าใหม่ไปยังล็อคออนของค่าตัวกรุปเมื่อมันต้องการ

ในการสร้างออบเจกต์ล็อคออนอินโฟ ทำได้โดยใช้เมธอด **CreateLogonInfo** ของออบเจกต์โมเดล หรือโมเดลของ ออบเจกต์ค่าตัวกรุป

ตัวแปรของออบเจกต์ล็อคออนอินโฟ มีดังต่อไปนี้

- varUser
- varPassword
- varServer
- varDatabase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- varDBMSName (ในส่วนนี้จะอ้างถึงตัวไครเวอร์ของฐานข้อมูลและทุกครั้งจะมีค่าเป็น ODBC)
- varDatasource
- varName

ถ้าเราใช้ตัวแปรทั้งหมดของออบเจกต์ลือคออนอินโฟ เราจะได้รูปแบบของคำสั่งดังต่อไปนี้

```
Dim oLogonInfo As ddoLogonInfo
```

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

3.6.3.1.1 การตั้งค่าออบเจกต์ลือคออนอินโฟ สำหรับ คาด้ากรู๊ป

ตัวอย่างต่อไปนี้แสดงว่าจะสร้างออบเจกต์ลือคออนอินโฟ อย่างไรสำหรับคาด้ากรู๊ป ตัวอย่างนี้สมมติว่าเราได้สร้างออบเจกต์ต่าง ๆ ต่อไปนี้แล้วคือ

- ออบเจกต์เอนจิน (DataDirector.Engine)
- ออบเจกต์โปรเจก (sports.ddx)
- ออบเจกต์คาด้ากรู๊ป (DGCustomer)

สำหรับจุดประสงค์ของตัวอย่างนี้คือ การส่งค่าชื่อของยูสเซอร์ (admin), รหัสผ่าน (admin), ฐานข้อมูล (sports), ชื่อของดีบีเอ็มเอส (DBMS) (ODBC) และคาด้าซอร์ส (Data source) (sports) เมื่อเราลือคออนไปยังฐานข้อมูลขณะรันโปรแกรมโดยคาด้าไครเรเตอร์ จะลือคออนโดยไม่มีการแสดงข้อมูลการ ลือคออนให้ผู้ใช้เห็น

```
' Create a LogonInfo object for the DataGroup
```

```
Dim oLogonInfo As ddoLogonInfo
```

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

3.6.3.1.2 การใช้ฟังก์ชัน (Public) สำหรับชื่อของยูสเซอร์ และรหัสผ่าน

เราสามารถอธิบายฟังก์ชัน ในการตั้งค่าชื่อยูสเซอร์ และรหัสผ่าน ในออบเจกต์ลือคออนอินโฟ ได้ดังนี้

```
Public Function SetLogonInfo(szName As String, szPassword As String, oLogonInfo  
As Object) As Boolean
```

ตัวอย่างต่อไปนี้ใช้ฟังก์ชันฟังก์ชันใน **SetLogonInfo** มันจะเช็คว่าได้มีการสร้างออบเจกต์ลือคออนอินโฟไว้หรือยัง ถ้ายังมันจะสร้างขึ้นมาอันหนึ่ง ถ้าออบเจกต์ลือคออนอินโฟมีอยู่แล้วโค้ดนี้จะตั้งค่าตัวแปร ชื่อยูสเซอร์ และรหัสผ่าน ไปยังออบเจกต์ที่มีอยู่แล้ว

```
Public Function SetLogonInfo(szName As String, szPassword As String, oLogonInfo  
As Object) As Boolean
```

```
On Error GoTo Error
```

If oLogonInfo Is Nothing Then

```
Set oLogonInfo = oDataGroup.CreateLogonInfo(szName, szPassword,
"sports", "ODBC", "sports")
```

Else

```
oLogonInfo.User = szName
```

```
oLogonInfo.Password = szPassword
```

End If

```
SetLogonInfo = True
```

Exit Function

```
Error:
```

```
SetLogonInfo = False
```

End Function

3.6.3.2 การ ล็อกออนจริง (Actual)

เพื่อแสดงการล็อกออนจริงด้วยการใช้เมทรูดอิมพอร์ต สำหรับออบเจกต์โมเดลหรือเมทรูดล็อกออน สำหรับออบเจกต์คาค้ากรุป ดังที่ได้อธิบายไว้แล้วในส่วนนี้

3.6.3.2.1 การเข้าถึงโครงสร้างฐานข้อมูล

หลังจากเราได้สร้างออบเจกต์ล็อกออนอินโฟ แล้วเราใช้เมทรูดอิมพอร์ตของออบเจกต์โมเดล เพื่อนำโครงสร้างของฐานข้อมูลเข้ามาในไฟล์โมเดล เมื่อนำเข้าโครงสร้างฐานข้อมูลไปที่โมเดล ตัวคาค้าไคเรคเตอร์ จะแสดงการล็อกออนเข้าฐานข้อมูล ,อิมพอร์ตฐานข้อมูลและล็อกออฟโดยอัตโนมัติ

```
oModel.Import
```

โค้ดนี้เอาตัวแปรที่อยู่ในออบเจกต์ล็อกออนอินโฟ และใช้ข้อมูลเหล่านั้นในการล็อกออนไปยังฐานข้อมูลโดยการใช้เมทรูด **Import** จากโมเดล ซึ่งก็คือ oModel

3.6.3.2.2 การ ล็อกออน ไปยังคาค้ากรุปโดยตรง

หลังจากเราสร้างออบเจกต์ล็อกออนอินโฟ เราใช้เมทรูด **Logon** ของ ออบเจกต์คาค้ากรุป ในการ ล็อกออน ไปยังฐานข้อมูล

```
oDataGroup.Logon
```

โค้ดนี้เอาตัวแปรที่อยู่ในออบเจกต์ล็อกออนอินโฟ และใช้ข้อมูลเหล่านั้นในการล็อกออนไปยังฐานข้อมูลโดยการใช้เมทรูด **Logon** จากคาค้ากรุปซึ่งก็คือ oModel

3.6.3.2.3 การ ล็อกออน โดยไม่ใช้ออบเจกต์ล็อกออนอินโฟ

เราสามารถล็อกออน ไปยังฐานข้อมูลได้โดยไม่ใช้ออบเจกต์ล็อกออนอินโฟ ได้โดยส่งตัวแปรเฉพาะในการล็อกออน โดยการใช้ล็อกออน ของออบเจกต์คาค้ากรุป ดังตัวอย่างด้านล่างนี้

```
oDataGroup.Logon "MyName", "MyPassword"
```

การทำงานนี้จะ ล็อกคอน ไปยังฐานข้อมูลด้วย *MyName* และ *MyPassword* เพื่อตั้งค่า ล็อกคอนของผู้ใช้และ รหัสผ่าน เราสามารถตั้งค่าเซิร์ฟเวอร์, ฐานข้อมูล, ชื่อดีบีเอ็มเอส และ คาด้าซอร์สถ้าเราต้องการ สังเกตได้ว่าวิธีการนี้วิธีการนี้จะไม่ใช่ตัวแปร (()) เป็นส่วนหนึ่งของรูปแบบคำสั่ง

3.6.3.2.4 การ ล็อกคอน โดยไม่ใช่ข้อมูลใด ๆ ของ โมเดล

บางครั้งวิธีนี้ก็จำเป็นในการใช้ล็อกคอนคาด้าไดเรกเตอร์ โดยไม่ใช่ข้อมูลของโมเดล ตัวอย่างเช่น เราจะทำสิ่งนี้ถ้าเราพัฒนาแอปพลิเคชันโดยใช้ฐานข้อมูลอันหนึ่งอยู่ แต่เปลี่ยนแอปพลิเคชันไปยังฐานข้อมูลอื่น ๆ ในการจะทำสิ่งนี้ให้ตั้งค่าคุณสมบัติของ ล็อกคอนโดยใช้โมเดลของคาด้ากรุป (DataGroup LogonUsingModel) ให้เป็น False

3.6.4 การทำงานกับข้อมูล

ในส่วนนี้จะแสดงตัวอย่างโค้ดที่แสดงว่าจะทำงานกับข้อมูลในการเขียนโปรแกรมอย่างไร รวมถึงการสร้างคอสต์ลิงค์, คิวรีข้อมูล, แสดงข้อมูล, และการใช้ข้อมูลโอเปค

3.6.4.1 การสร้างคาด้าลิงค์โดยใช้ดีดีโอ

คาด้าลิงค์ เป็นการติดต่อที่เราสร้างขึ้นระหว่าง คอลัมน์ ในฐานข้อมูลกับส่วนที่ใช้คอนโทรลของวิซวลเบสิก เราสามารถสร้างลิงค์เสมือน ซึ่งใช้ชื่อเฉพาะ คอลัมน์ ของฐานข้อมูลแต่ไม่ได้ชี้ไปที่คอนโทรลของวิซวลเบสิก ลิงค์เสมือนทำให้เราเรียกค้นข้อมูลจากโปรแกรมโดยไม่ต้องมีการแสดงมันในแอปพลิเคชันขณะทำงาน

เมื่อเราสร้างคาด้าลิงค์ เราจะต้องตั้งเส้นทางของข้อมูลของแต่ละคาด้าลิงค์ไปยังตารางและคอลัมน์ เฉพาะที่ซึ่งคาด้าลิงค์ ทำการเชื่อมต่อไปหา เส้นทางของข้อมูลจะต้องมีลักษณะเฉพาะกับตารางหลัก ขณะที่ป็นรูทคอด้านลิงค์ที่ไปยัง คอลัมน์ แล้วสามารถถูกหาที่อยู่ได้โดยคาด้าไดเรกเตอร์ ตามด้วยเส้นทางที่เริ่มต้นที่ ตารางหลัก

อย่างไรก็ตามเพื่อสร้างคาด้าลิงค์ ไปยังคอลัมน์ ที่ไม่สามารถไปถึงจากตารางหลักได้ คอลัมน์เหล่านั้นจะถูกเรียกค้นเมื่อ คาด้ากรุป ถูกคิวรีแต่ไม่มี มาสเตอร์-ดีเทล (Master-detail) เกิดขึ้นพร้อมกัน

ใช้เมทอด **CreateLink** ของออบเจกต์คาด้ากรุป เพื่อสร้างคาด้าลิงค์ สำหรับรูปแบบคำสั่งของคาด้ากรุป ของคาด้าลิงค์ เป็นดังนี้

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

ตัวแปร *varControl* เป็นออปชันถ้าเราต้องการสร้างลิงค์เสมือนที่ไม่เป็นลิงค์จริง ๆ ไปยังคอนโทรลของวิซวลเบสิก สำหรับตัวแปร *varControl* เราสามารถส่งค่าในรูปของชื่อเฉพาะของคอนโทรลของวิซวลเบสิก (*form1.text1*) ถ้าเราต้องการติดต่อ คอลัมน์ ของฐานข้อมูลไปยังตัวคอนโทรล

เราสามารถสร้าง คาด้าลิงค์ เพื่อออบเจกต์ที่ไม่ใช่คอม (Non-visual COM object) ซึ่งคาด้าลิงค์ชนิดนี้มีประสิทธิภาพมากกว่าคาด้าลิงค์เสมือน เพราะว่าทำให้แอปพลิเคชันของเราไม่นานเกินไปเมื่อมีการร้องขอข้อมูลทุก ๆ ครั้งที่มีการเปลี่ยนแปลง

3.6.4.2 การระบุเส้นทางข้อมูล

มีทาง 3 ที่เราสามารถระบุเส้นทางข้อมูลสำหรับค่าลิงค์

- รายละเอียดของตารางเริ่มต้นและคอลัมน์
- รายละเอียดเส้นทางบางส่วน (Partial path) และให้ ค่าค่าไคเรคเตอร์ แสดงความสัมพันธ์ที่แน่นอนซึ่งอยู่บนพื้นฐานของความสัมพันธ์ในโมเดล
- รายละเอียดเส้นทางทั้งหมดเริ่มต้นที่ ตารางหลัก

คำแนะนำ : ถ้าโค้ดของเรามีเส้นทางข้อมูลที่ไม่ครบหรือมีหลายเส้นทางสำหรับค่าลิงค์ จะทำให้ ค่าค่าไคเรคเตอร์ ฟ้องความผิดพลาดขึ้นมา

โค้ดตัวอย่างในส่วนนี้สาธิตว่าจะสร้างค่าลิงค์และค่าลิงค์เสมือนที่สมบูรณ์ได้อย่างไร บางตัวอย่างอาจแสดงวิธีการที่ต่างกันที่เราสามารถตั้งเส้นทางของข้อมูลที่ต่างกันของค่าลิงค์ ได้ ตัวอย่างโค้ดนี้สมมติว่า ตารางหลัก สำหรับ oDataGroup คือตารางลูกค้าและ คอนโทรลของวิซวลเบสิกที่เรากำลังจะสร้าง ค่าลิงค์ ไปชื่อว่า form1.text1

3.6.4.2.1 การระบุตารางเริ่มต้นและ คอลัมน์

โค้ดด้านล่างนี้สร้างค่าลิงค์โดยตรงไปยังคอลัมน์ของ customer_num ในตารางลูกค้า

```
Set oInkCustID = oDataGroup.CreateLink
("customer.customer_num", form1.text1)
```

3.6.4.2.2 การระบุ Partial Path

โค้ดด้านล่างนี้ระบุเส้นทางของข้อมูลและให้ค่าค่าไคเรคเตอร์อ้างถึงความสัมพันธ์

```
Set oInkOrderID =
oDataGroup.CreateLink("customer\order.order_num", form1.text1)
```

3.6.4.2.3 การระบุเส้นทางทั้งหมด

โค้ดด้านล่างนี้อธิบายเส้นทางทั้งหมดอย่างชัดเจน

```
Set oInkCustID = oDataGroup.CreateLink
("customer.customer_num/orders.customer_num", form1.text1)
```

3.6.4.2.4 การสร้าง ลิงค์เสมือน

โค้ดด้านล่างนี้ไม่ได้ตั้งค่าตัวแปร varControl และสร้างฮาร์ฟลิงค์ไปยังคอลัมน์ customer_num ในตารางลูกค้า

```
Set oInkCustID=
oDataGroup.CreateLink("customer.customer_num")
```

โค้ดด้านล่างนี้ไม่ได้ตั้งค่าตัวแปร varControl และสร้างฮาร์ฟลิงค์ (Half-link) ไปยังคอลัมน์ order_num ในตาราง order โดยใช้การแสดงผลเส้นทางแบบสมบูรณ์ดังนี้

```
Set oInkOrderID =
oDataGroup.CreateLink("customer\orders.order_num")
```

3.6.4.3 การ เรียกค้นข้อมูล

การ เรียกค้น ข้อมูลนำไปสู่การไปเอาข้อมูลจากฐานข้อมูลมาโดยปกติเพื่อแสดงในแอปพลิเคชัน บนวินโดวส์ด้วยดีไอโอข้อมูลที่ถูกระบุเรียกค้น จะถูกกระทำผ่านตาราง และออบเจกต์ค่าตัวกรุป ซึ่งมีระยะที่ กว้างเพื่อเรียกค้นข้อมูลโดยใช้ดีไอโอรวมถึงเมื่อแอปพลิเคชันเป็นการทำงานในโหมด Query by example (QBE) และเมื่อการ เรียกค้น โดยใช้คำสั่งในการทำงาน (Executable statement)

3.6.4.3.1 การ เรียกค้น ตาราง

เราสามารถแก้ไขข้อมูลที่ถูกระบุเรียกค้น สำหรับ คอลัมน์ จาก ตาราง, เรคคอร์ด หรือออบเจกต์ค่าตัว กรุป

ตัวอย่างต่อไปนี้เป็นวิธีการ เรียกค้น ข้อมูลว่าทำงานอย่างไรจากตารางถูกค่าในออบเจกต์ oDataGroup

```
Set QueryTable = oDataGroup.Tables("customer")
```

```
QueryTable.QueryData
```

3.6.4.3.2 การ เรียกค้น ข้อมูลในโหมดคิวบีอี (QBE)

ในการ เรียกค้น แบบคิวบีอีต้องทำตามขั้นตอนดังนี้

1. บอกให้ค่าตัวใดเรคเตอร์ รู้ว่าเราต้องการจัดการคิวบีอีโดยใช้การตั้งค่าต่อไปนี้

```
DataGroup.StartQBE = True
```

สถานะปัจจุบันของค่าตัวกรุป จะถูกรักษาไว้และออบเจกต์ตาราง จะถูกเตรียมสำหรับการ ทำงานในโหมดคิวบีอี

2. ตั้งค่าเงื่อนไขของคิวบีอี ในคอลัมน์ที่เกี่ยวข้องกับคิวบีอี เราสามารถทำสิ่งนี้ได้โดยการตั้งค่า ข้อมูลในคอลัมน์ ดังนี้

```
oDataGroup.StartQBE = True
```

```
oCustomers.Columns("Name").Value='A%'
```

3. จัดการกับ DataGroup.QueryData หรือ DataGroup.Tables(index).QueryData

ผลของคิวบีอีจะมีในค่าตัวกรุปของเรา สังเกตว่าหลังจากจัดการกับคำสั่ง คิวบีอี ทำให้คุณสมบัติ ของโหมดคิวบีอีบน ค่าตัวกรุป ถูกตั้งค่าให้เป็นฟอลส์โดยอัตโนมัติ

ตัวอย่างต่อไปนี้เป็นวิธีที่จะทำอย่างไรในการใช้โหมดคิวบีอีในการเรียกค้นข้อมูล ตัวอย่างนี้ใช้ พับบลิคฟังก์ชันที่ชื่อ QueryCustomer ชื่อของตารางที่ใช้ในการเรียกค้นถูกส่งเป็นค่าตัวแปรชื่อ szTableName

```
* Query a customer whose name is passed in as szName
```

```
Public Function QueryCustomer(szTableName as string) as
```

```
Object
```

```
Dim oTable as object
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

On error resume next

' Let Data Director know that you are starting QBE query

oDataGroup.StartQBE = True

' Set the customer surname into the LAST NAME column

oDataGroup.Tables("customer").Columns("lname").Value =

szCustName

' Execute the query

Set QueryCustomer = oDataGroup.Tables("customer")

QueryCustomer.QueryData

End Function

3.6.4.3.3 การแทนที่ค่าของ คอลัมน์ ด้วยผลการคำนวณ

ตัวอย่างโค้ดต่อไปนี้จะแทนที่ค่าในคอลัมน์ที่มีอยู่แล้วชื่อ price ด้วยการคำนวณค่าจากคอลัมน์ที่

ชื่อ calc

Set price = oDataGroup.CreateLink("items.price")

Set calc = oDataGroup.CreateLink("items.calc")

calc.expression = calc.value * 1.05

oDataGroup.QueryData

price.value = calc.value

3.6.4.3.4 การ เรียกค้นด้วยคำสั่งในการเอ็กซ์ซีคิว (Executable Statements)

ตารางเสมือนอนุญาตให้เราส่งค่าคำสั่งเอสคิวแอลหรือคำสั่งที่เรากำหนดขึ้นหรือคำสั่งในการเอ็กซ์ซีคิว ได้โดยตรงโดยผ่านควาต้าไครเคเตอร์เพื่อเรียกค้น ฐานข้อมูลของเราใช้ ตารางเสมือน เพื่อจัดการกับคำสั่งเอสคิวแอล เช่น Ad hoc query, Subquery, Group-by clause และสโตรโปรเซเจอร์

ใช้คุณสมบัติ **QueryCommand** ของออบเจกต์ตารางเพื่อรวบรวมคำสั่งในการเรียกค้นที่สัมพันธ์กับตารางเสมือนและใช้เมทอด **ExecuteCommand** ของออบเจกต์ตาราง ในการจัดการกับคำสั่งในการเรียกค้น หลังจากเราสร้าง ตารางเสมือน เราสามารถเพิ่มคุณสมบัติที่ทำได้ไปยังออบเจกต์ตาราง ได้เพื่อสร้างและจัดการคำสั่งอินเซอร์ท (INSERT), อัปเดต (UPDATE) และดิลิต (DELETE) ถ้าต้องการข้อมูลมากขึ้นเกี่ยวกับ ตารางเสมือน ให้ไปดูที่คำด้ากรู๊ป ตัวอย่างข้างล่างนี้จะเป็นการสร้างตารางเสมือน ที่มีชื่อว่า **oVTable** มีการกำหนดคำสั่งในการรันในคุณสมบัติ **QueryCommand** ส่งคำสั่งผ่านเมทอด **SetParam** และหลังจากนั้นก็ทำการคิวรี

' Query a customer whose name is passed in as szName

Function QueryCustomer(szName as String) as Object

Dim oVTable as object

' Instantiate virtual table

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Set oVTable = oDataGroup.CreateVirtualTable()
' Define SQL query statement
oVTable.QueryCommand = "SELECT * from customers where name = ?"
' Set name parameter
oVTable.SetParam QUERY_CMD, 1, szName
' Execute the query
oVTable.ExecuteQueryCommand
End Function

```

3.6.4.3.5 การคิวรีคอลัมน์ที่เป็นโรว์ไทป์ (Row Type)

โรว์ไทป์ จะเก็บข้อมูลจากหลาย ๆ คอลัมน์เป็น 1 คอลัมน์ ตัวอย่างเช่น ข้อมูลชนิดที่อยู่จะประกอบไปด้วยถนน,เมือง ,รัฐ และอื่น ๆ ตัวอย่างข้างล่างนี้จะเป็นเลือกข้อมูลจากตาราง **Employee** zip code จะถูกส่งผ่านเป็นพารามิเตอร์ของฟังก์ชัน ในฐานะข้อมูล zip code จะเก็บในคอลัมน์ชื่อ **code** ภายในโรว์ไทป์ ที่ชื่อ **zip** ซึ่งตัวมันเองยังจะเป็นคอลัมน์หนึ่งของโรว์ไทป์ที่ชื่อ **address**

```

Set oVTable = oDataGroup.CreateVirtualTable()
oVTable.QueryCommand = "SELECT * from employee where address.zip.code = ?"
oVTable.SetParam QUERY_CMD, 1, code
oVTable.ExecuteQueryCommand
' Print the value stored in the first column of the virtual table:
Debug.print ovTable.columns(1).value

```

หมายเหตุ ตัวอย่างนี้สมมติว่าไดร์เวอร์ฐานข้อมูลโอดีบีซี (ODBC database driver) รองรับการส่งผ่านพารามิเตอร์

3.6.4.3.6 การตรวจสอบผลลัพธ์

หลังจากที่ได้คิวรีตารางหรือตารางเสมือน สามารถที่จะดูผลลัพธ์ได้ซึ่งตัวอย่างข้างล่างนี้ใช้ตารางเสมือน ที่ชื่อ **ovTable** ประการแรก อาจจะแสดงข้อความ ถ้าไม่มีข้อมูลของการคิวรี เช่น

```

If ovTable is Nothing Then
MsgBox "No results were returned."
End If

```

ประการที่สอง อาจจะอ่านผลลัพธ์ขึ้นมาและแสดงผลเรคคอร์ดนั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If Not ovTable Is Nothing Then
  For i = 0 To ovTable.Columns.Count - 1
    Debug.Print ovTable.Columns(i + 1).Value
  ovTable.NextRecord
  Next
End If

```

3.6.4.4 การแสดงผลข้อมูล

ตัวอย่างข้างล่างนี้แสดงผลของการคิวรีข้อมูลของลูกค้าในแอปพลิเคชัน

```

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

```

บทที่ 4

ระบบการจัดหมวดหมู่หนังสือแบบห้องสมุดรัฐสภาอเมริกัน

4.1 สัญลักษณ์ของระบบการจัดหมวดหมู่หนังสือแบบห้องสมุดรัฐสภาอเมริกัน

สัญลักษณ์ หรือเลขเรียกหนังสือของระบบห้องสมุดรัฐสภาอเมริกันประกอบด้วยส่วนสำคัญ 2 ส่วน คือส่วนที่เป็นเลขหมู่หนังสือ (Class No.) ส่วนหนึ่ง กับส่วนที่เป็นเลขผู้แต่ง (Author No.) อีกส่วนหนึ่ง นอกจากนี้ก็อาจมีอักษรชื่อเรื่อง (Workmark) และ/หรือ ปีที่พิมพ์ประกอบเป็นบางกรณี

4.1.1 เลขหมู่หนังสือ (Class No.)

สัญลักษณ์แสดงเนื้อหาเรื่องราวของหนังสือระบบห้องสมุดรัฐสภาอเมริกันเป็นแบบผสม (Mixed notation) คือ มีทั้งอักษร (โรมัน) และตัวเลข (อารบิก) ผสมกัน

อักษรที่เป็นสัญลักษณ์อาจมีหนึ่งหรือสองตัวดังนี้

- Q สัญลักษณ์ของหนังสือหมวดวิทยาศาสตร์ (หมวดใหญ่)
- QA สัญลักษณ์ของหนังสือคณิตศาสตร์ (หมวดย่อย)
- QB สัญลักษณ์ของหนังสือดาราศาสตร์ (หมวดย่อย)
- QC สัญลักษณ์ของหนังสือฟิสิกส์ (หมวดย่อย)

สำหรับหมวด E-F และหมวด Z นั้นอักษรที่เป็นสัญลักษณ์แสดงหมวดหมู่จะมีเพียงตัวเดียวเสมอไป เนื่องจากยังไม่มีการใช้อักษรตัวที่สองซ้อนเลย

ต่อจากอักษรจะเป็นเลขอารบิก ซึ่งเป็น ได้ตั้งแต่เลขตัวเดียว คือ 1, 2, 3, 4, 5 เรื่อยไปจนถึงเลข 4 ตัว คือ 9999 ซึ่งเลขแต่ละจำนวนนี้อาจมีทศนิยมต่อท้ายได้ตามแต่กรณี ขึ้นอยู่กับตารางจำแนกเลขจะกำหนด เลขหมู่หนังสือของระบบห้องสมุดรัฐสภาอเมริกันจึงมีความสั้นยาวต่าง ๆ กัน ได้ดังนี้

อักษรตัวเดียว + เลขตัวเดียว (จำนวนเต็ม)

อักษรตัวเดียว + เลขหลายตัว (จำนวนเต็ม)

อักษรสองตัว + เลขตัวเดียว (จำนวนเต็ม)

อักษรสองตัว + เลขหลายตัว (จำนวนเต็ม)

อักษรตัวเดียว + เลขตัวเดียว + จุดทศนิยมและตัวเลข

อักษรตัวเดียว + เลขหลายตัว + จุดทศนิยมและตัวเลข

อักษรสองตัว + เลขตัวเดียว + จุดทศนิยมและตัวเลข

อักษรสองตัว + เลขหลายตัว + จุดทศนิยมและตัวเลข

ดังตัวอย่าง

Q 9

D 13.2

HC 462.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

QA 76.575

บางครั้งเลขหมู่หนังสือระบบห้องสมุดรัฐสภาอเมริกันจะต่างไปจากที่กล่าวมาแล้วคือ มีตัวอักษรเพิ่มอยู่หลังจุดทศนิยมก่อนจะถึงตัวเลขด้วย จุดทศนิยม + ตัวอักษร + ตัวเลข นี้รวมเรียกว่าเลขคัดเตอร์ เลขคัดเตอร์ที่ประกอบเป็นส่วนหนึ่งของเลขหมู่หนังสือเป็นครั้งคราวนี้มักเป็นเครื่องบ่งชี้เนื้อหาหนังสือที่มีการระบุเรื่องราวทางภูมิศาสตร์ เช่น ประเทศ ภูมิภาค รัฐ เมือง หรือมีเช่นนั้น ก็ใช้เพื่อระบุยุค สมัย ภาษา หรือ หัวข้อวิชาต่าง ๆ ตัวอย่างเช่น

L 961.T6 เป็นเลขหมู่หนังสือ นามานุกรมมหาวิทยาลัย และวิทยาลัยในประเทศไทย

.T6 คือเลขคัดเตอร์ที่เป็นส่วนหนึ่งของเลขหมู่หนังสือ ซึ่งกำหนดขึ้นเพื่อระบุชื่อประเทศ (Thailand)

AM 63.O8 เป็นเลขหมู่หนังสือ Directory of museums in Oslo

.O8 คือเลขคัดเตอร์ที่เป็นส่วนหนึ่งของเลขหมู่หนังสือ ซึ่งกำหนดขึ้นเพื่อระบุชื่อเมือง (Oslo)

4.1.2 เลขผู้แต่ง (Author No.)

เลขผู้แต่ง (Author No.) หนังสือภาษาต่างประเทศประกอบด้วยอักษรโรมันที่เป็นตัวพิมพ์ใหญ่ที่ได้มาจากอักษรแรกของชื่อผู้แต่ง กับเลขอารบิกที่กำหนดจากตาราง (Table) ซึ่งห้องสมุดรัฐสภาอเมริกันกำหนดขึ้นไว้

ตัวอย่างเช่น หนังสือ Analysis for action nursing care of elderly โดย Shirley R. Good และ Susan S. Rodgers เลขเรียกหนังสือ คือ RC964.G6

.G6 คือเลขผู้แต่งที่ได้จาก Good, Shirley, R.

หรือ หนังสือ Annual report จัดทำโดย Commercial and Industries Security Corporation เลขเรียกหนังสือ คือ HV 8290.C6

.C6 คือเลขผู้แต่งที่ได้จากชื่อองค์กรผู้จัดทำ “Commercial...”

ในบางครั้งที่รายการหลักของบัตรมิได้ลงรายการด้วยชื่อผู้แต่ง เลขผู้แต่งก็จะกำหนดจากชื่อที่ใช้ลงรายการหลัก เช่น

Museums Directory of the U.S. เลขเรียกหนังสือ AM11.M8

.M8 เป็นเลขผู้แต่งที่ได้จากรายการหลัก ซึ่งในกรณีนี้ใช้ชื่อเรื่อง “Museums...” ในรายการผู้แต่ง สำหรับหนังสือภาษาไทยเลขผู้แต่งจะใช้เขียนเป็นอักษร ก-ฮ ซึ่งได้มาจากอักษรแรกของชื่อผู้แต่ง (หรือชื่อเรื่อง) กับตัวเลขที่กำหนดจากตารางซึ่งมีอยู่หลายแบบแผนสุดแต่ห้องสมุดใดจะเลือกใช้แบบแผนใด

4.1.3 ปีที่พิมพ์ (Year of edition)

โดยปกติห้องสมุดรัฐสภาอเมริกันจะลงปีที่พิมพ์ในเลขเรียกหนังสือ หากงานนั้นมีใช้การพิมพ์ครั้งแรก (First edition) การเขียนสัญลักษณ์ใช้ปีที่พิมพ์ (Date of Publication) ลงต่อจากเลขผู้แต่ง หากมีการพิมพ์เกินหนึ่งครั้งในปีเดียวกันก็ใส่อักษร a พิมพ์ไว้ข้างล่างตอนท้ายของปีที่พิมพ์ สำหรับการพิมพ์ครั้งที่ 2 หรือใช้ b ในกรณีที่มีการพิมพ์ครั้งที่ 3 ในปีเดียวกัน

ดังตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Macmillan, Harold, 1894-

Winds of change, 1914-1939. London, Melbourne
(etc.) Macmillan, 1966.

Viii, 664p. 16 plates (incl. ports) 22 ½ cm.
55/-

(B66-16645)

Tables on endpapers.

Bibliographical footnotes.

I. Gt. Brit.-Pol. & govt.-20th cent. 2. Gt.

Brit.-For.rel.-20th cent. I. Title.

DA566.9.M33A3 1966 942.0820924 (B) 66-74717

Library of Congress (3)

พิมพ์ปี 1966

Macmillan, Harold, 1894-

Winds of change, 1914-1939. (1st U.S. ed.)

New York, Harper & Row (1966)

vi, 584p. illus., ports. 24 cm.

First vol. Of the author's memoirs.

Bibliographical footnotes.

I. Gt. Brit.-Pol. & govt.-20th cent. 2. Gt.

Brit.-For. rel.-20th cent. I. Title.

DA566.9.M33A3 1966a 66-21710

942.0820924 (B)

Library of Congress (67w7)

พิมพ์ปี 1966a

แต่ห้องสมุดรัฐสภาอเมริกันก็มีได้ถือเคร่งครัดในการกำหนดเช่นนี้นัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4 อักษรชื่อเรื่อง (Workmarks)

อักษรชื่อเรื่อง หรือ Workmarks คือ ส่วนหนึ่งของเลขเรียกหนังสือ ซึ่งใช้เฉพาะในบางกรณีเท่านั้น ส่วนมากมีใช้ในหมวด PZ ซึ่งเป็นหมวดวรรณกรรมสำหรับวัยรุ่น Workmarks ประกอบด้วยอักษรแรกของชื่อหนังสือ (เขียนด้วยตัวพิมพ์ใหญ่) เป็นสำคัญ แต่ในบางครั้งอักษรที่เป็นตัวพิมพ์ใหญ่นี้จะตามด้วยอักษรที่เป็นตัวพิมพ์เล็ก ซึ่งได้มาจากอักษรตัวถัดไปของชื่อหนังสือ หรือตัวเลขที่กำหนดขึ้นเพื่อให้การเรียงงานของนักเขียนคนหนึ่งเป็นไปอย่างเหมาะสมก็มี

ตัวอย่าง

PZ	Huxley, Aldous Leonard, 1894-
3	Crome yellow, by Aldous Huxley. New York,
.H 981	George H. Doran company (c1922)
Cr	307 p. 19 1/2 com
	I. Title

PZ	นวนิยายหรือวรรณกรรมวัยรุ่นภาษาอังกฤษ
3	นวนิยายภาษาอังกฤษ โดยผู้แต่งที่เป็นบุคคลเดียว
.H 981	เลขคัดเตอร์สำหรับผู้แต่ง
Cr	Workmark ของหนังสือชื่อ Crome yellow

4.1.5 การเขียนสัญลักษณ์

การเขียนสัญลักษณ์ของระบบห้องสมุดรัฐสภาอเมริกัน อาจเขียนได้หลายวิธี สำหรับห้องสมุดรัฐสภาอเมริกันเขียนสัญลักษณ์ดังนี้คือ

1. เลขเรียกหนังสือที่เขียนในบัตรรายการ ห้องสมุดรัฐสภาอเมริกัน เขียนอักษรกับเลขที่เป็นจำนวนเต็มไว้ในบรรทัดแรก แล้วเขียนเลขที่เป็นจุดทศนิยม (ถ้ามี) ไว้ในบรรทัดถัดมา บรรทัดถัดไปเป็นเลขคัดเตอร์ ซึ่งเขียนอยู่หลังจุดทศนิยม (แม้จะมีเลขคัดเตอร์ถึงสองครั้งก็จะเขียนไว้ในบรรทัดเดียวกัน) ปีที่พิมพ์ (ถ้าจำเป็นต้องมี) จะอยู่บรรทัดล่างสุด ดังนี้

QA9
E19
1994

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

QA76

.8

.I2594

A2i

1995

2. เลขเรียกหนังสือที่เขียนที่สันหนังสือ

เลขเรียกหนังสือที่เขียนที่สันหนังสือก็จะเป็นไปในทำนองเดียวกับเลขเรียกหนังสือที่เขียนบนบัตรรายการเว้นแต่แยกตัวเลขที่เป็นจำนวนเต็มมาไว้เป็นบรรทัดที่สอง ดังนี้

QA	QA76
9	.8
E19	.I2594
1994	A2i
	1995

อย่างไรก็ตาม ห้องสมุดอื่น ๆ ก็มิได้ใช้วิธีนี้เสมอไป ห้องสมุดหลายแห่งนิยมที่จะไม่ใส่จุดทศนิยมหน้าเลขคัตเตอร์เลย

4.2 ข้อดีข้อเสียของระบบ LCC

ปัจจุบันการจัดหมู่หนังสือระบบห้องสมุดรัฐสภาอเมริกันเป็นระบบกษ จัดหมู่หนังสือที่นิยมใช้แพร่หลาย โดยเฉพาะอย่างยิ่งในห้องสมุดระดับอุดมศึกษา นับตั้งแต่ประมาณปี พ.ศ. 1961 เป็นต้นมาห้องสมุดมหาวิทยาลัยต่าง ๆ ในสหรัฐอเมริกาได้เริ่มเปลี่ยนระบบการจัดหมู่หนังสือจากระบบทศนิยมคิวอี้ที่ใช้อยู่เดิมมาใช้ระบบห้องสมุดรัฐสภาอเมริกัน ทั้งนี้บัตรรายการสำเร็จรูปที่ฝ่ายจัดหมู่และทำบัตรรายการของห้องสมุดรัฐสภาอเมริกันจัดพิมพ์จำหน่ายอาจเป็นข้อจูงใจอย่างหนึ่งที่ทำให้ห้องสมุดหลายแห่งหันมานิยมจัดหมู่หนังสือด้วยระบบนี้ สำหรับในประเทศไทยห้องสมุดมหาวิทยาลัยหลายแห่งก็ได้เลือกใช้หรือเปลี่ยนมาใช้การจัดหมู่หนังสือระบบนี้กันมากขึ้น

4.2.1 ข้อดีของระบบการจัดหมู่หนังสือแบบห้องสมุดรัฐสภาอเมริกัน

1. ระบบการจัดหมู่หนังสือแบบห้องสมุดรัฐสภาอเมริกัน เป็นระบบการจัดหมู่หนังสือที่มีขอบเขตกว้างขวางครอบคลุมวิชาการต่าง ๆ ได้ทุกแขนงวิชา และแบ่งย่อยต่อไปได้ละเอียดจนสามารถชี้เฉพาะเนื้อหาของหนังสือแต่ละเล่ม จึงเป็นระบบการจัดหมู่หนังสือที่เหมาะสมสำหรับห้องสมุดขนาดใหญ่ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีหนังสือ สิ่งพิมพ์ หรือ เอกสารวัสดุเป็นจำนวนมาก ทั้งในหลาย ๆ สาขาวิชาและที่เป็นจำนวนมากเฉพาะในสาขาวิชาใดวิชาหนึ่ง

2. สัญลักษณ์ที่ใช้แทนหมวดหมู่ของหนังสือเป็นสัญลักษณ์แบบผสมที่เข้าใจได้ง่ายไม่เป็นตัวเลข ยืดยาวจนเกินไป ทำให้สะดวกในการค้นหา และในการจัดเรียงชั้นชั้น

3. ระบบการจัดหมู่หนังสือแบบห้องสมุดรัฐสภาอเมริกันเป็นระบบที่ยืดหยุ่น คือ สามารถขยายหรือเพิ่มเติมได้

- ด้วยการนำอักษรที่เว้นข้ามไว้ (I, O, W, X และ Y) มาใช้เป็นสัญลักษณ์
- ด้วยการเพิ่มอักษรตัวที่ 2 หรือตัวที่ 3, 4 ตามความเหมาะสม
- ด้วยการเพิ่มเลขกลางจำนวนที่ยังเว้นข้ามไว้
- ด้วยการขยายเลขที่มีอยู่แล้วออกเป็นทศนิยม
- ด้วยการใช้เลขคัตเตอร์

ด้วยวิธีเหล่านี้ จะทำให้ระบบการจัดหมู่หนังสือแบบห้องสมุดรัฐสภาอเมริกัน ยังคงใช้ได้ต่อไปอย่างมี

ประสิทธิภาพอีกนานเท่านาน แม้จะมีวิทยาการแขนงใหม่ ๆ เกิดขึ้นในเวลาข้างหน้า

4. ตารางต่าง ๆ (Tables) ในตารางจำแนกเลขตลอดจนเลขคัตเตอร์ที่ใช้ในกรณีพิเศษทำให้ระบบการจัดหมู่หนังสือแบบห้องสมุดรัฐสภาอเมริกัน สามารถกำหนดเลขเรียกหนังสือของหนังสือแต่ละเล่มได้ ซึ่งเฉพาะกว่าระบบอื่นๆ

5. ระบบการจัดหมู่หนังสือแบบห้องสมุดรัฐสภาอเมริกันมีการเพิ่มเติมแก้ไขอยู่ตลอดเวลา จึงเป็นระบบการจัดหมู่หนังสือที่นับได้ว่าทันสมัยที่สุด

4.2.2 ข้อเสียของระบบการจัดหมู่หนังสือแบบห้องสมุดรัฐสภาอเมริกัน

สำหรับข้อบกพร่องของระบบนี้ก็คือ ไม่มีตารางจำแนกเลขเล่มใดที่สมบูรณ์จริง ๆ กับทั้งไม่มีคู่มือ (Manual) อธิบายการใช้ที่ละเอียดและชัดเจน ผู้กำหนดเลขหมาย (Classifier) จะต้องเรียนรู้และทำความเข้าใจวิธีการใช้ด้วยตนเอง เพราะในตารางจำแนกเลขไม่มีคำอธิบายที่ละเอียดชัดเจนพอ และหรือในบางครั้งก็ละไว้ให้ผู้เข้าใจเอง นอกจากนี้ยังขาดดัชนีรวมที่ครอบคลุมตารางจำแนกเลขทั้งหมดด้วย

4.3 แผนการจัดหมู่หนังสือของระบบห้องสมุดรัฐสภาอเมริกัน

วัตถุประสงค์แรกเริ่มของระบบการจัดหมู่หนังสือแบบห้องสมุดรัฐสภาอเมริกันนั้น กำหนดขึ้นเพื่อใช้เฉพาะในห้องสมุดดังกล่าว การจำแนกหมวดหมู่วิชาก็มีได้อิงหลักปรัชญาใด ๆ และมีได้ถือการเรียงลำดับของวิทยาการเป็นข้อสำคัญ แต่พิจารณาความสะดวกในทางปฏิบัติเป็นเกณฑ์ ระบบการจัดหมู่หนังสือแบบห้องสมุดรัฐสภาอเมริกัน กำหนดวิทยาการต่าง ๆ เป็นหมวดหมู่ดังนี้ คือ

- | | |
|-------|------------------------|
| A | งานทั่ว ๆ ไป, ตารางเลข |
| B-BJ | ปรัชญา |
| BL-BX | ศาสนา |
| C | ประวัติศาสตร์เบ็ดเตล็ด |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D	ประวัติศาสตร์ทั่วไป และประวัติศาสตร์โลกเก่า
E-F	ประวัติศาสตร์โลกใหม่
G	ภูมิศาสตร์, มานุษยวิทยา, นิทานพื้นบ้าน, มรรยาทและประเพณี สันตนาการ
H	สังคมศาสตร์
J	รัฐศาสตร์
K	กฎหมาย
L	การศึกษา
M	จิตกรรม
N	วิจิตรศิลป์
P-PA	นิรุกติศาสตร์, ภาษาศาสตร์, วรรณกรรมคลาสสิก
PB-PH	ภาษายุโรปสมัยปัจจุบัน
PG	วรรณกรรมรัสเซีย
PJ-PM	ภาษาและ วรรณกรรมของเอเชีย, อัฟริกา, ดินแดนต่าง ๆ ในมหาสมุทร, ในทวีปอเมริกา, ภาษาประสม, และภาษาเทียม
PQ part 1	วรรณกรรมฝรั่งเศส
PQ part 2	วรรณกรรมอิตาลี, สเปน, และ ปอตุเกส
PT part 1	วรรณกรรมเยอรมัน
PT part 2	วรรณกรรมดัตช์ และ สแกนดิเนเวีย
Q	วิทยาศาสตร์
R	แพทยศาสตร์
S	เกษตรกรรม, อุตสาหกรรมที่เกี่ยวข้องด้วยพืชและสัตว์ การเพาะเลี้ยงปลาและการประมง, กีฬาล่าสัตว์
T	เทคนิควิทยา
U	ยุทธศาสตร์
V	นาวิกศาสตร์
Z	บรรณานุกรม และ บรรณารักษศาสตร์

จะเห็นได้ว่าหมวด B-P เป็นเรื่องทางมนุษยศาสตร์และสังคมศาสตร์ หมวด Q-V ว่าด้วยวิทยาศาสตร์และเทคโนโลยี หมวด Z สำหรับงานทางที่ไม่สามารถจัดเข้าไว้ในหมวดอื่นใดได้

ในขณะที่ตารางจำแนกเลขทุกเล่มเป็นอันหนึ่งอันเดียวกันด้วยหลักการพื้นฐาน แต่ละเล่มแต่ละหมวดก็มีโครงสร้างภายในแตกต่างกันไปตามธรรมชาติของวิชา ทำให้ตารางจำแนกเลขแต่ละเล่มมีวิธีการแบ่งย่อยแตกต่างกันไป และต่างก็มีตารางเฉพาะแบบส่วนตน ซึ่งไม่อาจใช้ได้กับทั้งระบบ เช่น การแบ่งทางภูมิศาสตร์ ในหมวดสังคมศาสตร์ (H) การแบ่งตามเรื่อง (subject) ในหมวดรัฐศาสตร์ (J) หรือการแบ่งในหมวดภาษาและวรรณกรรม (P) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องสมุดรัฐสภาอเมริกันได้กำหนดขอบข่ายของตารางจำแนกเลขแต่ละหมวดไว้ดังนี้ ดังจะยกตัวอย่างต่อไปนี้

4.4 การใช้ตารางเลขคัดเตอร์และการใช้เลขคัดเตอร์ในกรณีพิเศษ

การกำหนดสัญลักษณ์เพิ่มเติมให้กับหนังสือบางเล่มบางประเภท เพื่อบ่งบอกเนื้อหาหรือเรื่องราวโดยจำเพาะของหนังสือเล่มนั้น ๆ สัญลักษณ์ดังกล่าวประกอบด้วยจุดทศนิยม อักษรโรมัน และเลขอารบิก มีชื่อเรียกรวมกันว่า “เลขคัดเตอร์ (Cutter number) ในส่วนนี้จะกล่าวถึงการกำหนดเลขคัดเตอร์ เนื่องจากกาหนดเลขคัดเตอร์และการกำหนดเลขผู้แต่งของห้องสมุดรัฐสภาอเมริกัน เป็นการกำหนดเลขจากตารางเลขตารางเดียวกัน มีหลักและวิธีการกำหนดโดยทั่วไปเป็นแบบเดียวกัน จึงจะกล่าวถึงการกำหนดเลขคัดเตอร์และเลขผู้แต่งควบคู่กันไป

ตารางที่ห้องสมุด รัฐสภาอเมริกันจัดทำขึ้นเพื่อใช้เป็นเกณฑ์ในการกำหนดเลขผู้แต่งและเลขคัดเตอร์มีลักษณะเป็นตารางสั้น ๆ ง่าย ๆ ประกอบด้วยตารางเลข 4 ตารางดังนี้

ตารางเลขผู้แต่งของห้องสมุดรัฐสภาอเมริกัน

1. ตารางที่ 1 อักษรแรกเป็น S

อักษรตัวที่ 2	a	ch	e	hi	mop	t	u
เลขที่กำหนด	2	3	4	5	6	7-8	9

2. ตารางที่ 2 อักษรสองตัวแรกเป็น Qu

อักษรตัวที่ 3	a	e	i	o	r	y
เลขที่กำหนด	3	4	5	6	7	9

3. ตารางที่ 3 อักษรแรกที่เป็นพยัญชนะอื่น นอกจาก S, Qu

อักษรตัวที่ 2	a	e	i	o	r	u
เลขที่กำหนด	3	4	5	6	7	8

4. ตารางที่ 4 อักษรแรกที่เป็นสระ (Vowels)

อักษรตัวที่ 2	b	d	l m	n	p	r	st
เลขที่กำหนด	2	3	4	5	6	7	8

ตัวเลขในตารางทั้งหมด มีค่าเป็นเลขทศนิยม ผู้กำหนดเลขหมายอาจเพิ่มหรือลดค่าของตัวเลขได้ตามความเหมาะสม คืออาจให้เลขมีค่าสูงหรือต่ำกว่าตัวเลขที่กำหนดไว้ในตารางได้ไม่มากหรือน้อยกว่าหนึ่ง

4.4.1 การใช้ตารางกำหนดเลขคัดเตอร์

ก่อนอื่นผู้กำหนดเลขหมาย (เลขผู้แต่ง หรือเลขคัดเตอร์) จะต้องพิจารณาว่าชื่อผู้แต่ง หรือชื่ออื่นใดก็ตามที่ตารางจำแนกเลข (Schedule) ระบุว่าต้องกำหนดเลขคัดเตอร์นั้นขึ้นต้นด้วยอักษรอะไร เพื่อที่จะ

ทราบได้ว่าจะต้องใช้ตารางใดเป็นตัวกำหนด (ในกรณีที่เป็นการกำหนดเลขผู้แต่ง ต้องพิจารณาจากชื่อที่ใช้ลงรายการหลักตาม Anglo-American Cataloguing Rules)

ต่อจากนั้นจึงพิจารณาอักษรตัวที่สองของชื่อนั้น ๆ ตรวจสอบจากตารางว่าตารางได้กำหนดให้อักษรนั้นใช้เลขใด

ตัวอย่างเช่น ชื่อ Saint มีอักษรตัวแรกเป็น S ต้องใช้ตารางที่ 1 ในการกำหนด อักษรตัวที่สองของ Saint คือ a ตารางกำหนดให้เป็นเลข 2 (ดูตารางประกอบ) ชื่อ Saint จึงได้เลขผู้แต่ง หรือเลขคัดเตอร์เป็น .S2

หรือ เมื่อตารางจำแนกเลขระบุว่าเลขหมู่หนังสือที่สมบูรณ์จะตั้ง กำหนดเลขคัดเตอร์ประกอบโดยใช้วลี "By country, A-Z" หรือ "By state, A-Z" หรือ "By cities, A-Z" หรือ "By topic, A-Z" ฯลฯ บ่งบอกผู้กำหนดเลขหมายก็ต้องพิจารณาว่าหนังสือชิ้นนั้น ๆ เป็นเรื่องของประเทศ รัฐ เมือง หรือ เรื่องใด แล้วพิจารณาใช้ตารางกำหนดตามตัวอักษรตัวแรกของชื่อนั้น เช่น Germany อักษรแรกเป็น G ต้องใช้ตารางที่ 3 ซึ่งเป็นตารางสำหรับกำหนดเลขคัดเตอร์ของชื่อที่ขึ้นต้นด้วยอักษรอื่นที่ไม่ใช่ S และ Qu อักษรตัวที่ 2 เป็น e ตารางกำหนดให้ใช้เลข 4 เลขคัดเตอร์ของหนังสือที่ว่าด้วยเยอรมันนี่เล่มนั้นจึงต้องใช้ .G4 ประกอบดังนี้ เป็นต้น

ดังตัวอย่างต่อไปนี้

ชื่อที่ขึ้นต้นด้วยพยัญชนะ S

การกำหนดเลขผู้แต่ง หรือเลขคัดเตอร์ชื่อที่ขึ้นต้นด้วยพยัญชนะ S ใช้ตารางที่ 1 กำหนดตัวตัวอย่างต่อไปนี้

Schaffer เลขผู้แต่งหรือเลขคัดเตอร์คือ .S3

(อักษรที่อยู่ต่อจาก S คือ ch ซึ่งตารางกำหนดเลขให้ = 3)

Seaton เลขผู้แต่งหรือเลขคัดเตอร์คือ .S4

(อักษรที่อยู่ต่อจาก S คือ e ซึ่งตารางกำหนดเลขให้ = 4)

Shippely เลขผู้แต่งหรือเลขคัดเตอร์คือ .S5

(อักษรที่อยู่ต่อจาก S คือ h ซึ่งตารางกำหนดให้ = 5)

4.4.2 หลักและวิธีการกำหนดเลขคัดเตอร์และเลขผู้แต่ง

ตารางทั้ง 4 ที่กล่าวถึงข้างต้นมิใช่ตารางสำเร็จรูป หากเป็นตารางเลขอย่างกว้าง ๆ ซึ่งในการปฏิบัติงานจริง ๆ แล้ว ผู้กำหนดเลขหมายจะพบปัญหาและข้อยุ่งยากบ้าง เป็นต้นว่า ชื่อบางชื่อที่จะต้องกำหนดเลขผู้แต่ง หรือเลขคัดเตอร์ ประสมด้วยอักษรที่มีได้กำหนดไว้ในตาราง หรือบางครั้งชื่อผู้แต่งก็ขึ้นต้นและตามด้วยอักษรที่ซ้ำ ๆ กัน ซึ่งมีหลักและวิธีการกำหนดเลขผู้แต่ง หรือเลขคัดเตอร์ของห้องสมุดรัฐสภาอเมริกันดังนี้

1). ผู้กำหนดเลขหมายอาจเพิ่มหรือลดค่าของตัวเลขจากที่ตารางกำหนดไว้ได้ เพื่อหลีกเลี่ยงมิให้หนังสือมีสัญลักษณ์ซ้ำกัน

2). ตัวเลขที่กำหนดไว้ในแต่ละตารางนั้น มีค่าเป็นเลขทศนิยม เมื่อต้องการหลีกเลี่ยงมิให้หนังสือมีสัญลักษณ์ซ้ำกัน จึงเพิ่มหรือขยายเลขได้ตามความเหมาะสม โดยพิจารณาให้เลขตามลำดับของอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(อักษรที่อยู่ในลำดับต้น ๆ เช่น a,b,c,... ควรให้ตัวเลขที่มีค่าต่ำ ส่วนอักษรที่อยู่ในลำดับหลัง ๆ เช่น u,x,y,z ควรให้ตัวเลขที่มีค่าสูง)

3). การกำหนดเลขคัตเตอร์ หรือเลขผู้แต่งสำหรับชื่อที่มีได้สะกดด้วยตัวอักษรที่มีกำหนดไว้ในตาราง ผู้กำหนดเลขหมายสามารถกำหนดตัวเลขให้ได้โดยให้เลขมีค่าอยู่ในระหว่างอักษรตัวที่อยู่ก่อน กับอักษรตัวที่อยู่หลัง เช่น ในตารางที่ 3 กำหนดไว้ดังนี้

อักษรที่เป็นพยัญชนะอื่น นอก S และ Qu

อักษรตัวถัดไป	a	e	i	o	r	u
เลขที่ให้	3	4	5	6	7	8

ชื่อ Clapp ซึ่งต้องใช้ตารางนี้กำหนด มีอักษรตัวที่ 2 เป็น l ซึ่งมีได้มีกำหนดไว้ในตารางเลขที่ กำหนดให้จึงควรอยู่ระหว่างเลขของ i-o คือมีค่าระหว่าง 5-6 ซึ่งอาจเป็น .C51-.C59 ได้ (ในที่นี้ควรเป็น .C55 เพราะ l เป็นอักษรตัวกลาง ๆ ระหว่าง i-o)

อักษรตัวอื่น ๆ ที่มีได้มีกำหนดไว้ในตารางก็พิจารณาให้เลขในทำนองเดียวกันนี้

4). การกำหนดเลขผู้แต่งสำหรับหนังสือในหมวดเดียวกัน ที่ชื่อผู้แต่งขึ้นต้นด้วยอักษรเดียวกัน อักษรถัดไปซ้ำกัน ควรให้เลขต่างกัน โดยพิจารณาจากอักษรที่อยู่ลำดับต่อไป เช่น

ผู้แต่งชื่อ Sabine Sacramento Saint Sake เขียนหนังสือที่มีเนื้อเรื่อง (Subject) เดียวกันจึงมีเลขหมู่เหมือนกัน ผู้กำหนดเลขหมาย ควรกำหนดเลขผู้แต่งให้ต่างกัน แม้ว่าในตารางจะกำหนดไว้ว่า ชื่อที่ขึ้นต้นด้วย S อักษรตัวที่ 2 เป็น a เลขที่ให้ = .S2

เพื่อให้หนังสือแต่ละเล่มมีสัญลักษณ์ซ้ำกัน ผู้กำหนดเลขหมาย อาจกำหนดเลขผู้แต่งของชื่อทั้ง 4 นี้เป็นดังนี้

Sabine	=	.S12
Sacramento	=	.S13
Saint	=	.S15
Sake	=	.S16

แต่หากหนังสือของผู้แต่งทั้ง 4 คนนี้มีเลขหมู่หนังสือต่างกัน ก็อาจกำหนดเลขผู้แต่งของชื่อทั้ง 4 นี้ เป็น .S2 เหมือนกันทั้งหมดได้

สำหรับชื่อ Steel Storch Sturges ซึ่งอักษรตัวที่ 2 เป็น t และตารางกำหนดเลขให้ 7-8 นั้น หากเป็นชื่อผู้แต่งหนังสือที่อยู่ในหมวดเดียวกัน มีเลขหมู่เหมือนกันแล้ว เลขผู้แต่งที่กำหนดให้ชื่อเหล่านี้ จะต้องแตกต่างกันโดยพิจารณาจากอักษรตัวที่ 3 เช่น กำหนดให้

Steel	=	.S7
Storch	=	.S75
Sturges	=	.S8

แต่หากงานเขียนของบุคคลทั้งสามนี้ มีเนื้อหาต่างกัน เลขหมู่หนังสือต่างกันแล้ว เลขผู้แต่งของชื่อทั้ง 3 นี้ก็อาจให้เป็น .S7 เหมือนกันได้

สำหรับผู้แต่งที่เป็นบุคคลเดียวกัน เขียนหนังสือหมวดเดียวกัน แต่คนละเรื่อง ห้องสมุดรัฐสภาอเมริกันมีวิธีกำหนดสัญลักษณ์หนังสือให้ต่างกัน โดยเพิ่มเลขซึ่งพิจารณาจากชื่อเรื่องดังตัวอย่างต่อไปนี้

หนังสือ Managers for small industry ของ Stanford Research Institute

สัญลักษณ์ที่กำหนด HD 31.S68

หนังสือ Planning, action, profits; highlights of a Long Range Planning Service Management seminar ของ Stanford Research Institute

สัญลักษณ์ที่กำหนด HD 31.S683

จะเห็นว่าหนังสือทั้งสองเล่มนี้เนื้อหาอยู่ในหมวดเดียวกัน คือ HD 31 ผู้จัดทำหนังสือเป็นสถาบันเดียวกัน คือ Stanford Research Institute ห้องสมุดรัฐสภาอเมริกันกำหนดให้เลขผู้แต่งของหนังสือทั้งสองเล่มมีค่าแตกต่างกัน แต่ในขณะที่เดียวกันก็ใกล้เคียงกัน คือ .S68 และ .S683 ดังนี้เป็นต้น

4.4.3 การใช้เลขคัตเตอร์ในกรณีพิเศษ

นอกจากจะใช้เลขคัตเตอร์ดังที่ได้กล่าวถึงมาข้างต้นนี้แล้ว ห้องสมุดรัฐสภาอเมริกันยังมีการนำเลขคัตเตอร์มาใช้เป็นกรณีพิเศษอีก 3 ลักษณะ ดังนี้คือ

4.4.3.1 เลขคัตเตอร์ที่ใช้เป็นเลขสงวน (Official or reserved Cutter numbers)

เลขคัตเตอร์ที่ใช้เป็นเลขสงวน คือ เลขคัตเตอร์ที่ตารางจำแนกเลขสงวนไว้ให้เป็นเลขหมายเลขแสดงลักษณะ หรือขอบข่ายเนื้อหาของหนังสือบางเล่ม บางกรณี (ซึ่งส่วนใหญ่มักกำหนดให้เป็นเลขสำหรับเรื่องราวที่เป็นทางการ จึงมีชื่อเรียกว่า “Official Cutter numbers” เมื่อจะต้องมีการกำหนดเลขคัตเตอร์ในกรณีปกติทั่วไปประกอบเลขหลักตัวเดียวกันด้วย ผู้ทำหน้าที่กำหนดเลขหมู่ จึงต้องระวังมิให้นำเลขคัตเตอร์ที่ถูกกำหนดเป็นเลขสงวนไปใช้เป็นเลขคัตเตอร์ทั่วไป เพราะจะทำให้ความหมายผิดไปจากที่ตารางกำหนด

โดยปกติเลขคัตเตอร์ที่เป็นเลขสงวน จะเป็นเลขคัตเตอร์ลำดับต้นสุด ได้แก่ A1, A2 หรือ A1-3, A1-4, A1-5, A1-6 เป็นต้น

4.4.3.2 เลขคัตเตอร์ที่ใช้ซ้อน (Double Cutter numbers)

เลขคัตเตอร์ที่ใช้ซ้อน คือการนำเอาเลขคัตเตอร์มาใช้ในการกำหนดเลขหมายซ้อนกัน 2 ครั้ง กล่าวคือ ใช้ครั้งแรกเพื่อแสดงเนื้อหา และใช้ครั้งที่สอง สำหรับเป็นเลขผู้แต่ง แต่ในบางครั้งก็อาจใช้เลขคัตเตอร์ทั้งสองในการจำแนกเนื้อหาและขอบข่ายลักษณะเนื้อหาที่เฉพาะ

4.4.3.3 เลขคัตเตอร์ที่ใช้ตาม (Successive Cutter numbers)

เลขคัตเตอร์ที่ใช้ตาม เป็นการนำเลขหลังจากที่มีการกำหนดเลขคัตเตอร์ไปครั้งหนึ่งแล้ว เลขคัตเตอร์ที่ใช้ตาม ต่างจากเลขคัตเตอร์ที่ใช้ซ้อน กล่าวคือ เลขคัตเตอร์ที่ใช้ตามประกอบด้วยเลขคัตเตอร์ (จุดทศนิยมอักษร โรมัน เลขอารบิก) และเลขอารบิก

เลขคัตเตอร์ที่ใช้ตามอีกประเภทหนึ่ง เป็นเลขคัตเตอร์ที่มีเลขตามได้เป็นชุด แต่ละชุดอาจมี 2 จำนวน 3 จำนวน หรือมากกว่านั้น เลขในชุดจะไม่มีกำหนดค่าให้ตายตัว เหมือนเลขคัตเตอร์ที่ใช้ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเภทแรก เลขเหล่านี้อาจเป็นส่วนหนึ่งที่ประกอบอยู่ในตารางช่วย หรือ อาจเป็นตารางเลขง่าย ๆ สำหรับใช้เป็นเลขคัตเตอร์ที่ใช้ตามโดยเฉพาะ

เลขคัตเตอร์ที่ใช้ตามประเภทนี้จะมีใช้มากกว่าประเภทแรก โดยเฉพาะในหมวดวรรณกรรม และอาจมีใช้ในหมวดหมู่อื่น ๆ บ้าง ในกรณีที่ต้องการจะบ่งบอกลักษณะการประพันธ์

4.4.3.4 เลขคัตเตอร์สำหรับหนังสือภาษาไทย

เนื่องจากการจัดหมู่ระบบหอสมุดรัฐสภาอเมริกัน เกิดจากการปฏิบัติงานจริงตามหนังสือที่มีอยู่ในหอสมุดรัฐสภาอเมริกันโดยเฉพาะ ดังนั้น เพื่อนำมาใช้กับหนังสือภาษาไทยจึงต้องมีการคิดตารางเลขคัตเตอร์ขึ้น เพื่อใช้กำหนดเลขเรียกหนังสือภาษาไทย ดังต่อไปนี้คือ ตารางเลขคัตเตอร์สำหรับผู้แต่งคนไทย และตารางภูมิศาสตร์ที่ลำดับด้วยตัวอักษร ตามแบบของคัตเตอร์สำหรับชื่อจังหวัดในประเทศไทย

4.4.3.4.1 ตารางเลขคัตเตอร์สำหรับผู้แต่งคนไทยของ สำนักหอสมุดมหาวิทยาลัยธรรมศาสตร์

ซึ่ง น.ส. ทองหยด ประทุมวงษ์ ได้จัดทำขึ้นโดยอาศัยหลักตามตารางเลขคัตเตอร์ของการจัดหมู่ระบบหอสมุดรัฐสภาอเมริกัน แต่ได้ดัดแปลงให้เหมาะสมกับลักษณะของการเรียงลำดับอักษรและรูปสระตามแบบของพจนานุกรมฉบับราชบัณฑิตยสถาน คือ

ตารางที่ 4-1 ตารางพยัญชนะ ใช้สำหรับชื่อที่ขึ้นต้นด้วยพยัญชนะตัวแรกไม่มีรูปสระประกอบแบ่งออกเป็น 4 กลุ่ม แต่ละกลุ่มมีตัวเลขประจำ ดังต่อไปนี้

ก ข ฃ ค ฅ ฉ ง จ ฉ ช ฌ ญ ฎ ฏ	2
ฐ ฑ ฒ ณ ค ต ถ ท ธ น บ ป ผ ฝ พ	3
ฟ ภ ม ย ร ฤ ฤ ล ก ฎ ฎ	4
ว ศ ษ ส ห พ อ ย	5

ตัวอย่าง

ปชา = ป2	เพราะ = ช=2
ปทุม = ป3	เพราะ = ท=3

ตารางที่ 4-2 ตารางสระ ใช้สำหรับชื่อที่ขึ้นต้นด้วยคำแรกหรือพยางค์แรกที่มีรูปสระประกอบแบ่งออกเป็น 4 กลุ่ม แต่ละกลุ่มมีตัวเลขประจำ ดังต่อไปนี้

ะ อี อี๋ ำ อ่า อี อี๋ อี	6
อี อู อู เ เอะ เอา เอาะ เอ็	7
เอ็ย เอ็ยะ เอ็อ เอ็อะ แอ แอ๊ะ	8
โ โอะ ำ ำ	9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง

เกษรา = ก7 เพราะ ะ = 7
 เรืองยศ = ร8 เพราะ ื่อ = 8

ใช้กับชื่อที่เป็นตัวอักษรย่อ ตัวเลขที่ใช้แทนคือ 1 แต่ไม่ควรใช้ 1 ควรใช้ 11-19, 111-199, 1111-1999 ฯลฯ

ตัวอย่าง ก. ก. = ก12
 ก. แก้วประเสริฐ = ก13
 ก. วรณโกวิท = ก15

4.4.3.4.2 ตารางภูมิศาสตร์ที่ลำดับด้วยอักษร A การระบุเขตภูมิศาสตร์

เช่น ชื่อจังหวัดหรือภาค เป็นสิ่งจำเป็นในการระบุรายละเอียดของเนื้อเรื่องในหนังสือภาษาไทย จึงได้มีห้องสมุดมหาวิทยาลัย 2 แห่ง ที่ใช้การจัดหมู่ระบบห้องสมุดรัฐสภาอเมริกัน จัดทำตารางชื่อจังหวัดตามอักษร A-Z ตามแบบตารางภูมิศาสตร์ที่ลำดับด้วยอักษรขึ้น คือ สำนักบรรณสารการพัฒนา สถาบันบัณฑิตพัฒนบริหารศาสตร์ และ หอสมุดกลาง มหาวิทยาลัยเกษตรศาสตร์

ตัวอย่างตารางภูมิศาสตร์ที่ลำดับด้วยอักษร

1. แบบของสำนักบรรณสารการพัฒนา สถาบันบัณฑิตพัฒนบริหารศาสตร์

DS	Thailand.		
589	Local history and description.		
	By city, town, etc., A-Z		
.A5	Ang Thong	อ่างทอง	อท
.A9	Ayutthaya	พระนครศรีอยุธยา	อย
.B3	Bangkok	พระนคร	กท
.B8	Buri Ram	บุรีรัมย์	บร

2. แบบของหอสมุดกลาง มหาวิทยาลัยเกษตรศาสตร์

จังหวัด/ภาค (เขียนเป็นภาษาไทย)	จังหวัด/ภาค (เขียนเป็นภาษาอังกฤษ)	L.C. Cutter	หมายเหตุ
กระบี่	Krabi	K7	
กรุงเทพฯ	Bangkok	B2	
กาญจนบุรี	Kanchanaburi	K32	
กาฬสินธุ์	Kalasin	K3	
กำแพงเพชร	Kamphaeng Phet	K31	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับอักษรย่อบอก ลำดับเล่มและอักษรย่อบอกจำนวนฉบับในเลขเรียกหนังสือสำหรับหนังสือภาษาไทย ใช้ตัวอย่างนี้

- ล เล่ม (Volume)
ฉ ฉบับ (Copy)

4.5 การกำหนดและวิเคราะห์เลขหมู่หนังสือหมวด A-Z

4.5.1 ขั้นตอนการปฏิบัติในการกำหนดเลขหมู่หนังสือระบบห้องสมุดรัฐสภาอเมริกัน

การกำหนดเลขหมู่หนังสือตามระบบห้องสมุดรัฐสภาอเมริกัน มีขั้นตอนในการปฏิบัติดังนี้

1. พิจารณาเนื้อหาของหนังสือว่าหนังสือที่ว่าด้วยเรื่องอะไร ควรใช้ตารางจำแนกเลขหมวดใด เช่น หนังสือประวัติศาสตร์ เนื้อเรื่องหลักคือการศึกษา จึงควรพิจารณาตารางจำแนกเลขหมวด L การศึกษา หนังสือ Applied numerical analysis เนื้อเรื่องหลักคือวิทยาศาสตร์ จึงควรพิจารณาตารางจำแนกเลขหมวด Q วิทยาศาสตร์ เป็นต้น

2. เมื่อพิจารณาได้ว่าจะต้องใช้ตารางจำแนกเลขหมวดใดแล้ว ก็เปิด Synopsis และ/หรือ Outline ของตารางจำแนกเลขหมวดนั้น ๆ ว่าครอบคลุมเรื่องราวดังกล่าวหรือไม่ ถ้าเป็นเรื่องที่ตารางจำแนกเลขนั้นครอบคลุม ก็จะต้องตรวจสอบดูกลุ่มเลขที่ Synopsis และ/หรือ Outline ระบุไว้ในตารางจำแนกเลขต่อไป หากไม่ใช่เรื่องที่มีอยู่ในตารางจำแนกเลขหมวดนั้น ๆ ให้พิจารณาทบทวนใหม่ว่าเรื่องนั้น ๆ อาจอยู่ในหมวดอื่นใดได้บ้าง แล้วเปิด Synopsis และ/หรือ Outline ของหมวดนั้น ๆ คูณกว่าจะพบ

3. ตรวจสอบดัชนีท้ายเล่มของตารางจำแนกเลข (ถ้ามี) ว่าได้กำหนดเลขหมู่เรื่องนั้น ๆ หรือเรื่องใกล้เคียงเรื่องนั้น ๆ ไว้อย่างไร แล้วเปิดดูเลขที่ได้จากตารางจำแนกเลขในเล่มเพื่อตรวจสอบความถูกต้องและความสมบูรณ์

4. หากไม่มีดัชนี หรือ ดัชนี ไม่มีเรื่องที่ต้องการ เปิดตารางจำแนกเลขพิจารณากลุ่มเลขที่ Synopsis หรือ Outline ให้ไว้อย่างกว้าง ๆ เพื่อหาเลขที่ตรงกับเนื้อหาของหนังสือมากที่สุด

5. อ่านคำอธิบาย ข้อแนะนำ ข้อพึงปฏิบัติ ข้อพึงสังเกต คำสั่ง ฯลฯ (ซึ่งมักพิมพ์ด้วยอักษรขนาดจิ๋ว) ปฏิบัติตามคำสั่งและข้อแนะนำที่มีอยู่ในตารางจำแนกเลขนั้น ๆ ทุกขั้นตอน

หมายเหตุ

การกำหนดและวิเคราะห์เลขหมู่หนังสือแต่ละหมวดของระบบห้องสมุดรัฐสภาอเมริกัน มีหลักใหญ่ ๆ เป็นหลักการเดียวกัน ผู้ศึกษาสามารถนำวิธีการกำหนดและวิเคราะห์ ที่ได้แสดงไว้ในหมวดหนึ่งไปใช้ในการกำหนดและวิเคราะห์ เลขหมู่หนังสือ ในหมวดอื่น ๆ ได้ หากตารางจำแนกเลขได้แสดงไว้ในลักษณะเดียวกัน

4.5.2 ตัวอย่างการวิเคราะห์ที่ยกมามีเฉพาะหมวด Q

หมวด Q วิทยาศาสตร์ (Science)

ขอบข่ายและลักษณะของตารางจำแนกเลข

หมวดนี้ครอบคลุมเรื่องราวที่เป็นวิทยาศาสตร์บริสุทธิ์ ได้แก่วิทยาศาสตร์ทั่วไป คณิตศาสตร์ ดาราศาสตร์ ฟิสิกส์ เคมี ธรณีวิทยา ธรรมชาติวิทยา ชีววิทยา พฤกษศาสตร์ สัตววิทยา มนุษยกายวิภาคศาสตร์ สรีรวิทยา จุลชีววิทยา

ตารางจำแนกเลขหมวดนี้ไม่มีการใช้ตารางประกอบ มีเพียงการใช้ตารางภายในง่าย ๆ และที่แตกต่างไปจากหมวดอื่น ๆ คือ มีการใช้ “Book numbers” Book numbers ในหลาย ๆ แห่งใช้ในความหมายเดียวกับเลขผู้แต่ง (Author number) แต่ในที่นี้ “Book nos.” เป็นส่วนหนึ่งของเลขหมู่หนังสือ

ตัวอย่างการกำหนดและวิเคราะห์หนังสือในหมวด Q

หนังสือ *Applied numerical analysis*

เลขหมู่หนังสือที่กำหนด QA 297

กำหนดและวิเคราะห์ได้ดังนี้

- | | |
|-----|--|
| Q | สัญลักษณ์ของหมวดวิทยาศาสตร์ |
| QA | อักษรสองตัวแสดงหมวดย่อย-คณิตศาสตร์ |
| 297 | ตัวเลขที่กำหนดไว้สำหรับเรื่องการวิเคราะห์ตัวเลขที่เป็นเรื่องทั่วไป งานนิพนธ์ หรือตำราเรียน |

หนังสือ *Organic chemistry*

เลขหมู่หนังสือที่กำหนด QD 251

กำหนดและวิเคราะห์ได้ดังนี้

- | | |
|-----|---|
| Q | สัญลักษณ์ของหมวดวิทยาศาสตร์ |
| QD | อักษรสองตัวแสดงหมวดย่อย-เคมี |
| 251 | ตัวเลขที่กำหนดไว้สำหรับ Organic chemistry ที่เป็นเรื่องทั่วไป งานนิพนธ์ หรือตำราชั้นสูง ตั้งแต่ต้นจนถึงปี ค.ศ. 1970 |

บทที่ 5

การออกแบบ ER Diagram และการออกแบบโปรแกรมประยุกต์

ในบทนี้จะกล่าวถึงการนำสิ่งต่างๆที่ได้ศึกษามาทำการสร้าง โปรแกรมประยุกต์ซึ่งได้เลือกระบบสารสนเทศห้องสมุดมาเป็นกรณีศึกษา

5.1 Requirement และ Specification ของ Application

เป็น โปรแกรมประยุกต์ที่ทำหน้าที่ในการสืบค้นข้อมูลในห้องสมุดโดยสนับสนุนการทำงานแบบไคลเอนท์/เซิร์ฟเวอร์ โดยฝั่งไคลเอนท์จะพัฒนาโดยใช้วิซวลเบสิก (Visual Basic) และทางฝั่งคาค้าเบสเซิร์ฟเวอร์ ใช้อินฟอร์มิคซ์ยูนิเวอร์แซลเซิร์ฟเวอร์ซึ่งการจัดการกับฐานข้อมูลจะพัฒนาโดยใช้โมเดลฐานข้อมูลเชิงวัตถุสัมพันธ์ แบ่งการทำงานออกเป็น 4 ส่วน

5.1.1 เซิร์ฟเวอร์

โปรแกรมประยุกต์นี้จะทำการอ่านข้อมูลเพื่อจัดการทุกอย่างจากฝั่งเซิร์ฟเวอร์ โดยเราจะทำการสร้างเอสพีแอลรูทีน (SPL Routine) ซึ่งเขียนโดยใช้สตอร์โพรซีเจอร์ของอินฟอร์มิคซ์ (Informix Stored Procedure Language) ซึ่งเป็นภาษาที่มีความสามารถของเอสคิวแอล (SQL) และเพิ่มความสามารถในการทำการคอนโทรลการทำงานลงไป (Flow Control) เช่น การทำลูป (Looping) หรือ การทำทางเลือก (Branching) เป็นตัวการในการตรวจสอบและจัดการกับข้อมูลให้กับไคลเอนท์

5.1.2 Data Director

การติดต่อระหว่าง โปรแกรมประยุกต์กับฐานข้อมูลจะติดต่อผ่านคาค้าไคเรคเตอร์ (Data Director) ซึ่งจะเป็นตัวจัดการการติดต่อให้ คาค้าไคเรคเตอร์เป็นทูลที่ทางอินฟอร์มิคซ์ยูนิเวอร์แซลเซิร์ฟเวอร์มีมาให้สำหรับการติดต่อระหว่างฐานข้อมูลกับวิซวลเบสิกโดยเฉพาะ คาค้าไคเรคเตอร์จะช่วยให้การติดต่อกับฐานข้อมูลทำได้โดยง่าย ลดการเขียนโค้ดลงไปได้มาก อีกทั้งยังใช้งานง่ายเพียงลากแล้วปล่อยเมาส์ลงบนคอนโทรลของวิซวลเบสิก (Drag-Drop) ก็สามารถนำข้อมูลจากฐานข้อมูลขึ้นมาแสดงผลได้

5.1.3 ส่วนของผู้ใช้

ส่วนของผู้ใช้เป็นส่วนของผู้ใช้ทั่ว ๆ ไป ซึ่งสามารถค้นหาข้อมูลจากระบบได้อย่างเดียวโดยที่ไม่สามารถแก้ไขข้อมูลต่าง ๆ ในฐานข้อมูลได้ซึ่งมีฟังก์ชันการทำงานหลัก ๆ ดังต่อไปนี้

1. ฟังก์ชันการค้นหาข้อมูลหนังสือที่ต้องการ ซึ่งสามารถค้นหาได้จากฟิลด์ต่าง ๆ ดังนี้

- 1.1 เลขหมู่หนังสือ
- 1.2 ชื่อหนังสือ
- 1.3 ชื่อผู้แต่ง
- 1.4 ประเภทของหนังสือ
- 1.5 สำนักพิมพ์
- 1.6 ปีที่พิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ฟังก์ชันการค้นหาข้อมูลเพลงที่ต้องการ ซึ่งสามารถค้นหาได้จากฟิลด์ต่าง ๆ ดังนี้

- 2.1 เลขหมู่เพลง
- 2.2 ชื่อเพลง
- 2.3 ชื่อศิลปิน
- 2.4 ชื่ออัลบั้ม
- 2.5 ปีที่ออกอัลบั้ม
- 2.6 แนวเพลง
- 2.7 ค่ายเพลง

3. ฟังก์ชันการค้นหาข้อมูลภาพยนตร์ที่ต้องการ ซึ่งสามารถค้นหาได้จากฟิลด์ต่าง ๆ ดังนี้

- 3.1 เลขหมู่ภาพยนตร์
- 3.2 ชื่อภาพยนตร์
- 3.3 นักแสดง
- 3.4 ประเภทของภาพยนตร์
- 3.5 ปีที่สร้างภาพยนตร์
- 3.6 ชื่อของบริษัทผู้สร้าง

4. ฟังก์ชันในการแสดงข้อมูลโดยละเอียดของหนังสือซึ่งจะแสดงรูปภาพหน้าปกของหนังสือเล่มนั้น ในส่วนของรายละเอียดของเพลงก็จะแสดงเสียงของเพลง ๆ นั้น ในส่วนของรายละเอียดของภาพยนตร์จะแสดงภาพยนตร์ของภาพยนตร์นั้น ๆ

5. ฟังก์ชันในการแสดงตำแหน่งของหนังสือ เพลง และภาพยนตร์ ที่อยู่ในหนังสือสมุด

5.1.4 ส่วนของผู้ดูแลระบบ

ในส่วนของผู้ดูแลระบบเป็นส่วนของผู้ที่ทำหน้าที่ดูแลระบบนี้เอง ซึ่งจะทำให้ผู้ดูแลระบบสะดวกในการจัดการกับข้อมูลที่อยู่ในระบบ เช่นการเพิ่ม (Insert) ข้อมูลใหม่ การแก้ไขข้อมูลที่มีอยู่แล้ว (Update) ซึ่งในส่วนนี้ประกอบไปด้วยฟังก์ชันต่าง ๆ ดังนี้

1. ฟังก์ชันในการแก้ไขข้อมูลในฐานข้อมูล ใช้ในการแก้ไขข้อมูลที่มีอยู่แล้วในฐานข้อมูล
 - 1.1 ฟังก์ชันในการแก้ไขข้อมูลประเภทหนังสือ
 - 1.2 ฟังก์ชันในการแก้ไขข้อมูลประเภทเพลง
 - 1.3 ฟังก์ชันในการแก้ไขข้อมูลประเภทภาพยนตร์
 - 1.4 ฟังก์ชันในการแก้ไขข้อมูลประเภทตู้เก็บหนังสือ เพลง ภาพยนตร์ และห้องสมุด
2. ฟังก์ชันในการเพิ่มข้อมูลเข้าไปในฐานข้อมูล ใช้ในการป้อนข้อมูลใหม่ ๆ เข้าไปในระบบ
 - 2.1 ฟังก์ชันในการเพิ่มข้อมูลประเภทหนังสือ
 - 2.2 ฟังก์ชันในการเพิ่มข้อมูลประเภทเพลง
 - 2.3 ฟังก์ชันในการเพิ่มข้อมูลประเภทภาพยนตร์
 - 2.4 ฟังก์ชันในการเพิ่มข้อมูลประเภทตู้เก็บหนังสือ เพลง ภาพยนตร์ และห้องสมุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

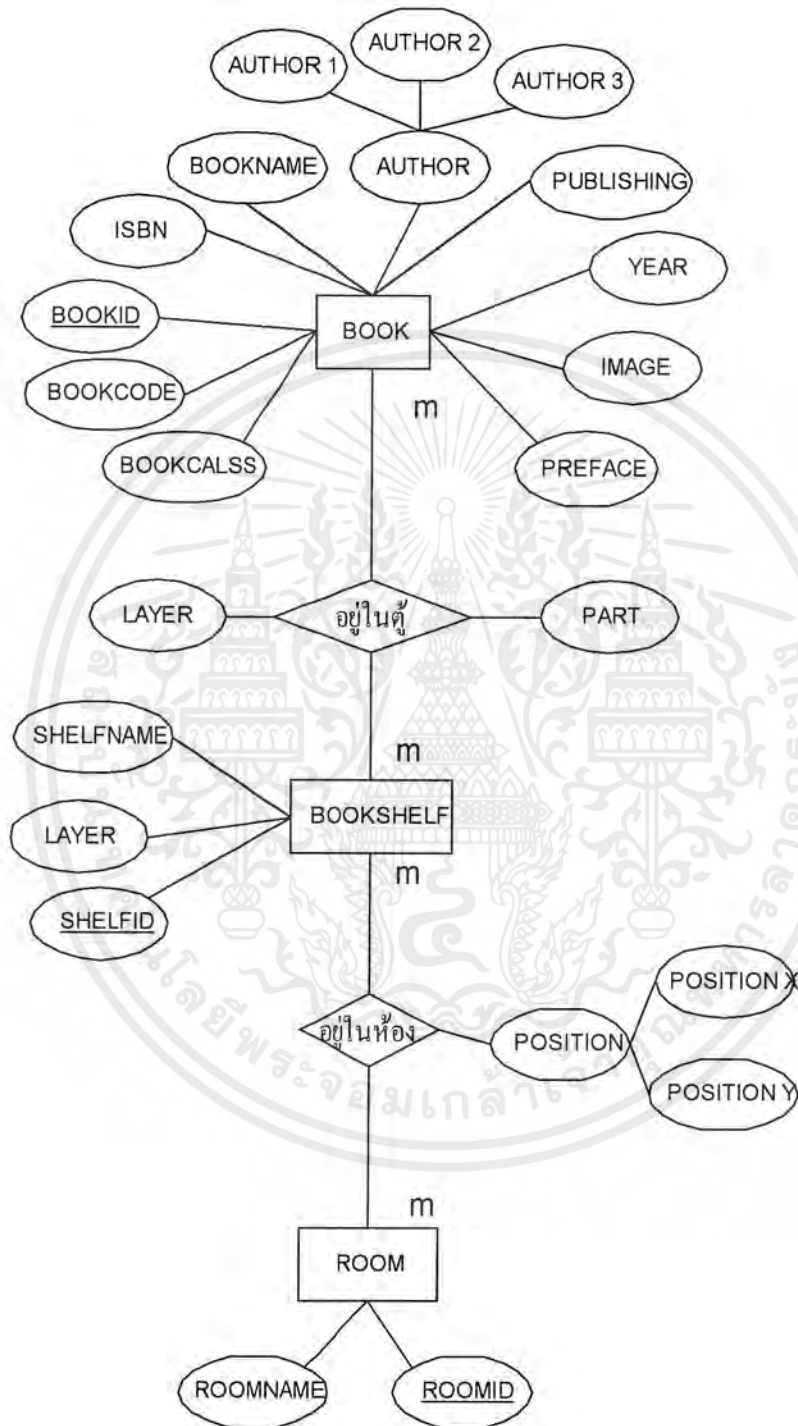
3. ฟังก์ชันในการค้นหาข้อมูลในฐานข้อมูล ใช้ในการตรวจสอบข้อมูลที่ได้เพิ่ม หรือแก้ไขไปแล้วว่าถูกต้องครบถ้วนไหม
 - 3.1 ฟังก์ชันในการค้นหา ข้อมูลประเภทหนังสือ
 - 3.2 ฟังก์ชันในการค้นหาข้อมูลประเภทเพลง
 - 3.3 ฟังก์ชันในการค้นหาข้อมูลประเภทภาพยนตร์
 - 3.4 ฟังก์ชันในการค้นหาข้อมูลประเภทตู้เก็บหนังสือ เพลง ภาพยนตร์ และห้องสมุด
4. ฟังก์ชันในการลบข้อมูลในฐานข้อมูล
 - 4.1 ฟังก์ชันในการลบข้อมูลประเภทหนังสือ
 - 4.2 ฟังก์ชันในการลบข้อมูลประเภทเพลง
 - 4.3 ฟังก์ชันในการลบข้อมูลประเภทภาพยนตร์
 - 4.4 ฟังก์ชันในการลบข้อมูลประเภทตู้เก็บหนังสือ เพลง ภาพยนตร์ และห้องสมุด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

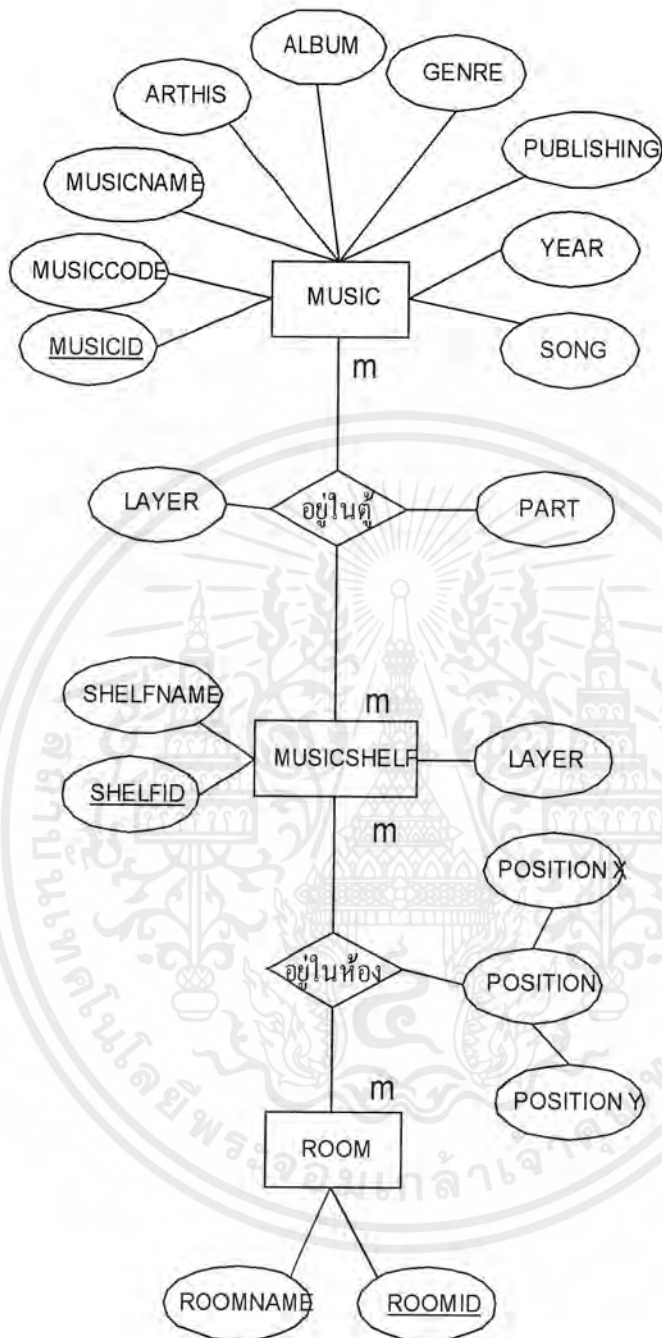
5.2 การวิเคราะห์และการออกแบบ ER Diagram

5.2.1 การออกแบบ ER Diagram



รูปที่ 5-1 ER Diagram ความสัมพันธ์ของหนังสือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-3 ER Diagram ความสัมพันธ์ของเพลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 การออกแบบตารางที่ใช้ในโปรแกรมประยุกต์

ตารางต่าง ๆ ที่ใช้ในระบบฐานข้อมูล

ตาราง BOOK เป็นตารางที่ใช้ในการเก็บรายละเอียดต่าง ๆ ของหนังสือแต่ละเล่ม

BOOKID	BOOKCODE	ISBN	BOOKNAME	AUTHOR		
				AUTHOR1	AUTHOR2	AUTHOR3

BOOKCLASS	PUBLISHING	YEAR	IMAGE	PREFACE
-----------	------------	------	-------	---------

ซึ่งสามารถสร้างด้วยภาษา SQL ดังนี้

```
CREATE ROW TYPE author_t
```

```
(
AUTHOR1 varchar(20),
AUTHOR2 varchar(20),
AUTHOR3 varchar(20)
);
```

```
CREATE ROW TYPE position_t
```

```
(
X integer,
Y integer
);
```

```
CREATE TABLE BOOK
```

```
(
BOOKID serial primary key,
BOOKCODE varchar(15) not null unique,
ISBN varchar(13),
BOOKNAME varchar(40) not null,
AUTHOR author_t,
BOOKCLASS varchar(15),
PUBLISHING varchar(20),
```

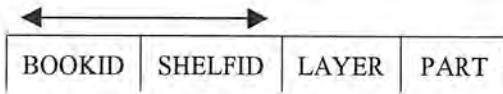
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

YEAR integer,
IMAGE media_t เป็นข้อมูลชนิด Opaque
PREFACE media_t เป็นข้อมูลชนิด Opaque
);

```

ตาราง ADDRESSBOOK เป็นตารางที่ใช้ในการเก็บที่อยู่ของหนังสือในตู้หนังสือ



ซึ่งสามารถสร้างด้วยภาษา SQL ดังนี้

```

CREATE TABLE ADDRESSBOOK
(
BOOKID integer REFERENCES book (bookid) ON DELETE CASCADE,
SHELFID integer REFERENCES bookshelf (shelfid) ON DELETE CASCADE,
LAYER integer not null,
PART char(1) not null,
primary key (BOOKID,SHELFID)
);

```

ตาราง BOOKSHELF เป็นตารางที่ใช้ในการเก็บรายละเอียดต่างๆ ของตู้หนังสือ



ซึ่งสามารถสร้างด้วยภาษา SQL ดังนี้

```

CREATE TABLE BOOKSHELF
(
SHELFID serial primary key,
SHELFNAME varchar(10) not null,
LAYER integer
);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง ADDRESSBOOKSHELF เป็นตารางที่ใช้ในการเก็บที่อยู่ของตู้หนังสือในห้องสมุด

ADDRESSBOOKSHELF		
SHELFID	ROOMID	POSITION
		X Y

ซึ่งสามารถสร้างด้วยภาษา SQL ดังนี้

```
CREATE TABLE ADDRESSBOOKSHELF
(
SHELFID integer REFERENCES bookshelf (shelfid) ON DELETE CASCADE,
ROOMID integer REFERENCES room (roomid) ON DELETE CASCADE,
POSITION position_t not null,
PRIMARY KEY (SHELFID,ROOMID)
);
```

ตาราง ROOM เป็นตารางที่ใช้ในการเก็บรายละเอียดของห้องสมุดแต่ละห้อง

ROOM	
ROOMID	ROOMNAME

ซึ่งสามารถสร้างด้วยภาษา SQL ดังนี้

```
CREATE TABLE ROOM
(
ROOMID serial primary key,
ROOMNAME varchar(10) not null
);
```

ตาราง MUSIC เป็นตารางที่ใช้ในการเก็บรายละเอียดต่าง ๆ ของเพลงแต่ละเพลง

MUSIC						
MUSICID	MUSICCODE	MUSICNAME	ARTHIST	ALBUM	GENRE	PUBLISHING

YEAR	SONG

ซึ่งสามารถสร้างด้วยภาษา SQL ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CREATE TABLE music
(
MUSICID serial primary key,
MUSICCODE varchar(15) not null unique,
MUSICNAME varchar(30) not null,
ARTHIST varchar(25),
ALBUM varchar(20),
GENRE varchar(10),
PUBLISHING varchar(20),
YEAR integer,
SONG media_t เป็นข้อมูลชนิด Opaque
);
```

ตาราง ADDRESSMUSIC เป็นตารางที่ใช้ในการเก็บที่อยู่ของเพลงในตู้เก็บเพลง



MUSICID	SHELFID	LAYER	PART
---------	---------	-------	------

ซึ่งสามารถสร้างด้วยภาษา SQL ดังนี้

```
CREATE TABLE ADDRESSMUSIC
(
MUSICID integer REFERENCES music (musicid) ON DELETE CASCADE,
SHELFID integer REFERENCES musicshelf (shelfid) ON DELETE CASCADE,
LAYER integer not null,
PART char(1) not null,
primary key (MUSICID,SHELFID)
);
```

ตาราง MUSICSHELF เป็นตารางที่ใช้ในการเก็บรายละเอียดต่าง ๆ ของตู้เก็บเพลง



SHELFID	SHELFNAME	LAYER
---------	-----------	-------

ซึ่งสามารถสร้างด้วยภาษา SQL ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CREATE TABLE MUSICSHELF
(
SHELFID serial primary key,
SHELFNAME varchar(10) not null,
LAYER integer
);
```

ตาราง ADDRESSMUSICSHELF เป็นตารางที่ใช้ในการเก็บที่อยู่ของตู้เก็บเพลงในห้องสมุด

SHELFID	ROOMID	POSITION	
		X	Y

ซึ่งสามารถสร้างด้วยภาษา SQL ดังนี้

```
CREATE TABLE ADDRESSMUSICSHELF
(
SHELFID integer REFERENCES musicshelf (shelfid) ON DELETE CASCADE,
ROOMID integer REFERENCES room (roomid) ON DELETE CASCADE,
POSITION position_t not null,
PRIMARY KEY (SHELFID,ROOMID)
);
```

ตาราง MOVIE เป็นตารางที่ใช้ในการเก็บรายละเอียดต่างๆ ของภาพยนตร์แต่ละเรื่อง

MOVIEID	MOVIECODE	MOVIEName	ACTOR		MOVIECLASS
			ACTOR 1	ACTOR 2	

PUBLISHING	YEAR	FILM

ซึ่งสามารถสร้างด้วยภาษา SQL ดังนี้

```
CREATE ROW TYPE actor_t
(
ACTOR1 varchar(20),
ACTOR2 varchar(20)
);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CREATE TABLE movie
(
MOVIEID serial primary key,
MOVIECODE varchar(15) not null unique,
MOVIEName varchar(35) not null,
ACTOR actor_t,
MOVIECLASS varchar(20),
PUBLISHING varchar(20),
YEAR integer,
FILM media_t เป็นข้อมูลชนิด Opaque
);
```

ตาราง ADDRESSMOVIE เป็นตารางที่ใช้ในการเก็บที่อยู่ของภาพยนตร์ในตู้เก็บภาพยนตร์



MOVIEID	SHELFID	LAYER	PART
---------	---------	-------	------

ซึ่งสามารถสร้างด้วยภาษา SQL ดังนี้

```
CREATE TABLE ADDRESSMOVIE
(
MOVIEID integer REFERENCES movie (movieid) ON DELETE CASCADE,
SHELFID integer REFERENCES movieshelf (shelfid) ON DELETE CASCADE,
LAYER integer not null,
PART char(1) not null,
primary key (MOVIEID,SHELFID)
);
```

ตาราง MOVIESHELF เป็นตารางที่ใช้ในการเก็บรายละเอียดต่าง ๆ ของตู้เก็บภาพยนตร์



SHELFID	SHELFNAME	LAYER
---------	-----------	-------

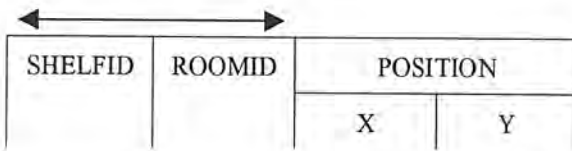
ซึ่งสามารถสร้างด้วยภาษา SQL ดังนี้

```
CREATE TABLE MOVIESHELF
(
SHELFID serial primary key,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SHELFNAME varchar(10) not null,
 LAYER integer
);

ตาราง ADDRESSMOVIESHELF เป็นตารางที่ใช้ในการเก็บที่อยู่ของผู้เก็บภาพยนตร์ในห้องสมุด

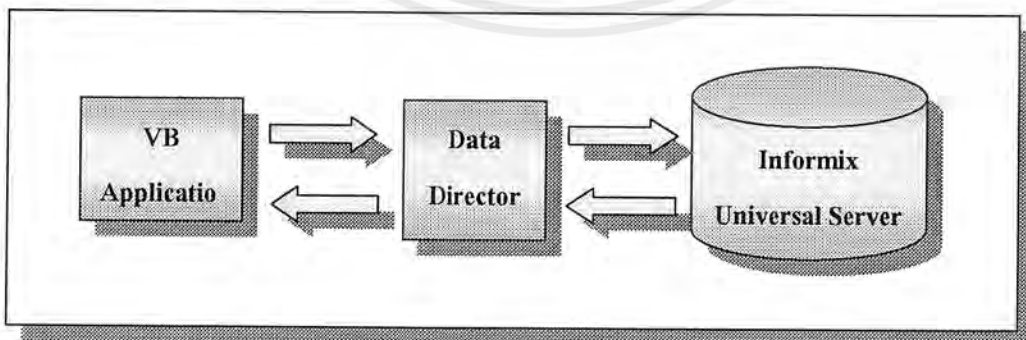


ซึ่งสามารถสร้างด้วยภาษา SQL ดังนี้

```
CREATE TABLE ADDRESSMOVIESHELF
(
  SHELFID integer REFERENCES movieshelf (shelfid) ON DELETE CASCADE,
  ROOMID integer REFERENCES room (roomid) ON DELETE CASCADE,
  POSITION position_t not null,
  PRIMARY KEY (SHELFID,ROOMID)
);
```

5.2.3 การออกแบบโปรแกรมประยุกต์

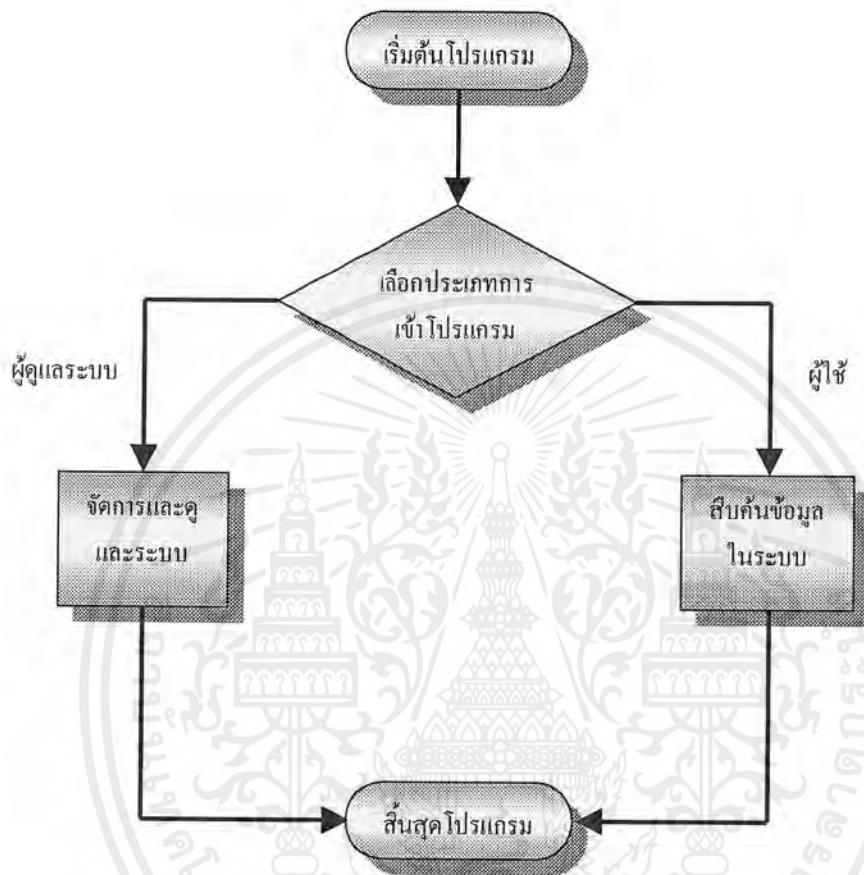
ในการพัฒนาโปรแกรมประยุกต์ระบบสารสนเทศหือ สมุดมีโครงสร้างดังแสดงในรูปที่ 5-4 จากรูปโปรแกรมประยุกต์จะใช้วิซวลเบสิกในการพัฒนาและทำการติดต่อกับฐานข้อมูลโดยใช้ดาต้าไคเร็กเตอร์ซึ่งใช้หลักการของดาต้าไคเร็กเตอร์ออบเจกต์ (DDO) ในการติดต่อ ซึ่งดาต้าไคเร็กเตอร์นี้เป็นเครื่องมือที่ทางอินฟอร์มิซ ได้จัดเตรียมไว้ให้ใช้ในการพัฒนาโปรแกรมประยุกต์ที่จะต้องทำงานติดต่อกับฐานข้อมูลอินฟอร์มิซ และใช้ระบบการจัดการฐานข้อมูลของอินฟอร์มิซในการจัดเก็บข้อมูลมัลติมีเดีย



รูปที่ 5-4 สถาปัตยกรรมของโปรแกรมประยุกต์ระบบสารสนเทศห้องสมุด

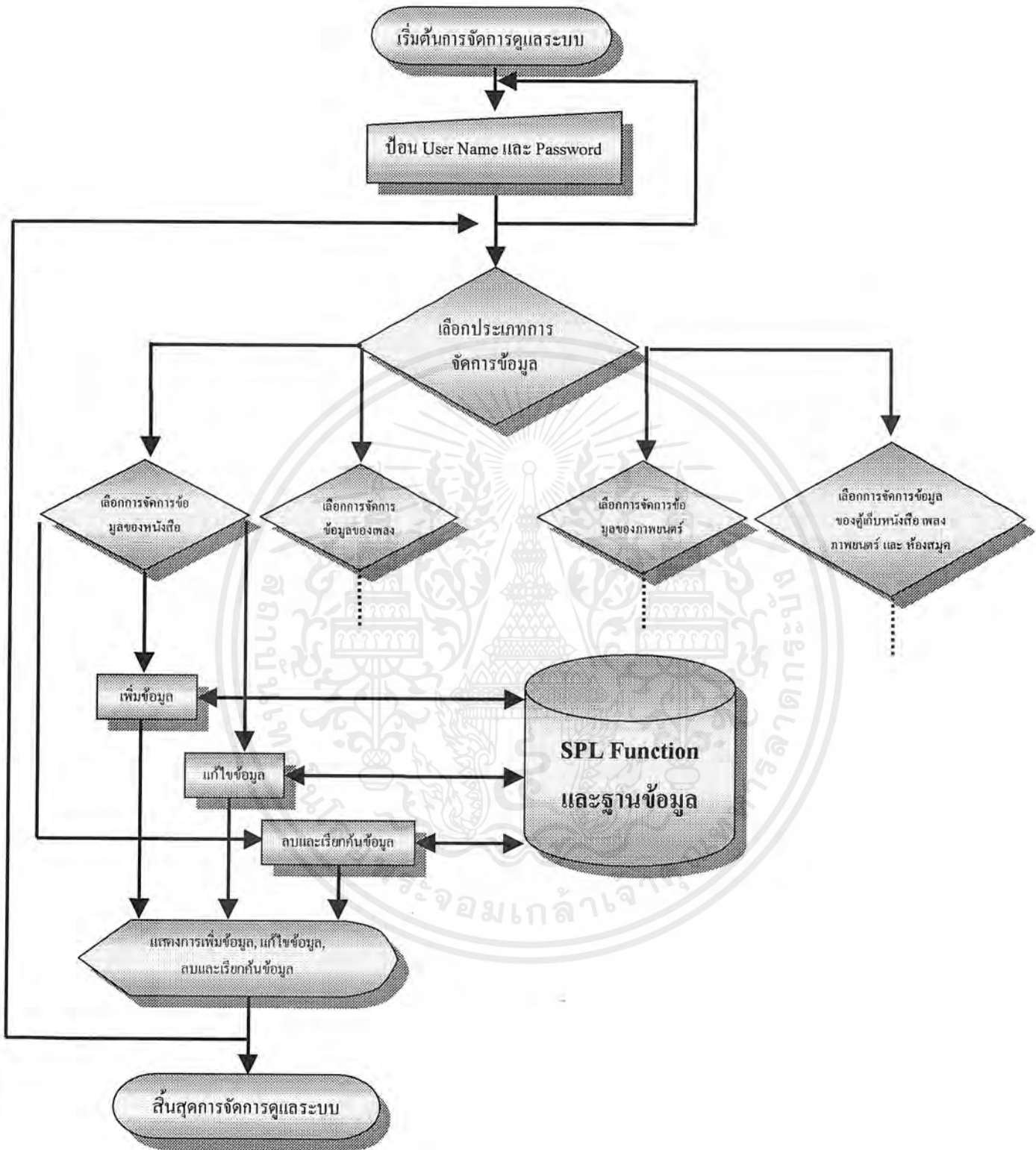
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.4 โฟลว์ชาร์ทของโปรแกรมประยุกต์



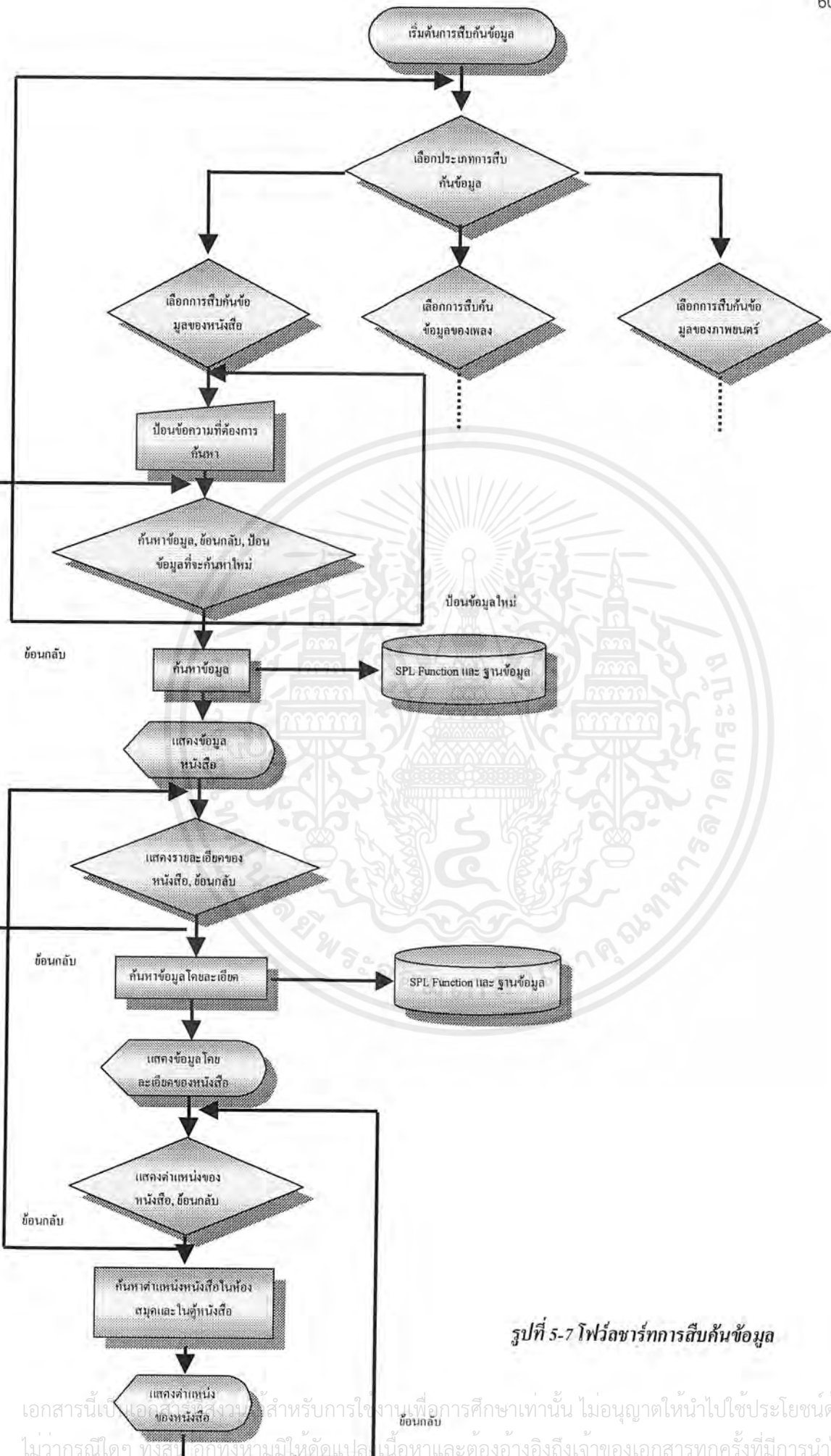
รูปที่ 5-5 โฟลว์ชาร์ทการทำงานของระบบสารสนเทศห้องสมุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-6 โฟลว์ชาร์ทการจัดการดูแลระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-7 โฟลว์ชาร์ทการสืบค้นข้อมูล

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น เอกสารสงวนลิขสิทธิ์ให้เด็ดขาดเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของโปรแกรมประยุกต์สามารถอธิบายได้ดังรูปที่ 5-5 5-6 และ รูปที่ 5-7 ซึ่งในการทำงานได้แบ่งการทำงานเป็น 2 ส่วนใหญ่ ๆ คือ ส่วนของผู้ดูแลระบบ และ ส่วนของผู้ใช้ระบบ โดยจะทำการเลือกในการเริ่มต้นโปรแกรมประยุกต์

ในส่วนของผู้ดูแลระบบดังรูปที่ 5-6 จะเริ่มด้วยการใช้ผู้ดูแลระบบเลือกว่าจะต้องการจัดการกับข้อมูลประเภทใดโดยมีให้เลือกดังต่อไปนี้ ประเภทหนังสือ ประเภทเพลง ประเภทภาพยนตร์ และผู้เก็บหนังสือ เพลง ภาพยนตร์ และห้องสมุด เมื่อได้ทำการเลือกว่าต้องการจัดการกับข้อมูลประเภทใดแล้วก็จะ มีฟังก์ชันให้เลือกทำงานดังต่อไปนี้ คือ ฟังก์ชันในการเพิ่มข้อมูล ฟังก์ชันในการแก้ไขข้อมูล ฟังก์ชันในการลบและค้นหาข้อมูล โดยในการทำงานของฟังก์ชันต่าง ๆ ดังที่ได้กล่าวมานี้จะมีการดึงข้อมูลจากฐานข้อมูลโดยผ่าน SPL Function และผลการทำงานของจะแสดงให้ผู้ดูแลระบบได้ทราบทางจอภาพ

ในส่วนของผู้ใช้ระบบดังรูปที่ 5-7 จะเริ่มด้วยการให้ผู้ใช้ระบบเลือกว่าจะทำการค้นหาข้อมูลประเภทใดเมื่อได้ทำการเลือกสิ่งที่ค้นหาแล้ว เช่น ต้องการค้นหาหนังสือ จะต้องทำการป้อนรายละเอียดที่ต้องการค้นหาลงในหัวข้อที่จัดไว้เช่น รหัสหนังสือ ชื่อหนังสือ หรือผู้แต่ง เมื่อเราสั่งให้โปรแกรมประยุกต์ค้นหาข้อมูลตัวโปรแกรมจะทำการติดต่อกับฐานข้อมูลเพื่อดึงข้อมูลบางส่วนขึ้นมาแสดง โดยทำการติดต่อกับ SPL Function และนำข้อมูลนั้นมาแสดงผลซึ่งข้อมูลที่นำมาแสดงจะไม่ละเอียดนัก เช่น แสดงชื่อหนังสือ รหัสหนังสือ ประเภทหนังสือ เมื่อต้องการทราบรายละเอียดของหนังสือเล่มนั้น ๆ โปรแกรมจะแสดงรายละเอียดของหนังสืออีกครั้งหนึ่ง โดยจะแสดงรายละเอียดที่มากขึ้นเช่นแสดงว่าอยู่ที่ ชั้นใด ห้องใดในห้องสมุด แสดงเลข ไอเอสบีเอ็นของหนังสือ และสามารถแสดงตำแหน่งเฉพาะเจาะจงลงไปในห้องสมุดได้ด้วย ซึ่งในการค้นหาข้อมูลประเภท เพลงและภาพยนตร์จะมีลักษณะการทำงานของโปรแกรมเหมือนกัน

5.2.5 การออกแบบการจัดการดาต้าเบสเซอร์ฟเวอร์

การทำงานที่ฝั่งดาต้าเบสเซอร์ฟเวอร์นอกจากการ จัดเก็บข้อมูลแล้วที่ฝั่งดาต้าเบสเซอร์ฟเวอร์จะมีการฝังรูทีน เพื่อให้ฝั่งไคลเอนท์มาเรียกใช้ซึ่งรูทีนดังกล่าวเขียนโดยใช้ภาษา Stored Procedure ของ อินฟอร์มิคส์ซึ่งโดยที่ฝั่งเซอร์ฟเวอร์จะประกอบไปด้วยรูทีนต่างๆดังนี้

5.2.5.1 รูทีนที่ทำงานให้กับผู้ใช้ทั่วไป

ชื่อรูทีน GETINFOBOOK

หน้าที่ ทำการค้นหาข้อมูลเบื้องต้นของหนังสือในห้องสมุด เพื่อให้ผู้ใช้ห้องสมุดได้ทราบว่าหนังสือที่ต้องการค้นหา มีรายละเอียดเบื้องต้นอะไรบ้าง

ถูกเรียกใช้เมื่อ ต้องการค้นหาหนังสือในห้องสมุด

รายละเอียด จะทำการค้นหาหนังสือในห้องสมุด เพื่อแสดงข้อมูลให้ผู้ใช้งานห้องสมุดได้ทราบข้อมูลเบื้องต้นของหนังสือที่ค้นหา (ถ้ามีหนังสือเล่มนั้นในฐานข้อมูล) โดยในการค้นหานี้ สามารถจำกัดลักษณะของหนังสือในการค้นหาได้ เช่น ต้องการหนังสือประเภทคอมพิวเตอร์ เป็นต้น เพื่อความสะดวกรวดเร็วในการค้นหา

ชื่อรูทีน GETINFOMUSIC

หน้าที่ ทำการค้นหาข้อมูลเบื้องต้นของเพลงในห้องสมุด เพื่อให้ผู้ใช้ห้องสมุดได้ทราบว่าเพลงที่ต้องการค้นหา มีรายละเอียดเบื้องต้นอะไรบ้าง

ถูกเรียกใช้เมื่อ ต้องการค้นหาเพลงในห้องสมุด

รายละเอียด จะทำการค้นหาเพลงในห้องสมุด เพื่อแสดงข้อมูลให้ผู้ใช้ห้องสมุดได้ทราบข้อมูลเบื้องต้นของเพลงที่ค้นหา (ถ้ามีเพลงเพลงนั้นในฐานข้อมูล) โดยในการค้นหานี้ สามารถจำกัดลักษณะของเพลงในการค้นหาได้ เช่น ต้องการเพลงประเภทเพลงร็อค เป็นต้น เพื่อความสะดวกรวดเร็วในการค้นหา

ชื่อรูทีน GETINFOMOVIE

หน้าที่ ทำการค้นหาข้อมูลเบื้องต้นของภาพยนตร์ในห้องสมุด เพื่อให้ผู้ใช้ห้องสมุดได้ทราบว่าภาพยนตร์ที่ต้องการค้นหา มีรายละเอียดเบื้องต้นอะไรบ้าง

ถูกเรียกใช้เมื่อ ต้องการค้นหาภาพยนตร์ในห้องสมุด

รายละเอียด จะทำการค้นหาภาพยนตร์ในห้องสมุด เพื่อแสดงข้อมูลให้ผู้ใช้ห้องสมุดได้ทราบข้อมูลเบื้องต้นของภาพยนตร์ที่ค้นหา (ถ้ามีภาพยนตร์เรื่องนั้นในฐานข้อมูล) โดยในการค้นหานี้ สามารถจำกัดลักษณะของภาพยนตร์ในการค้นหาได้ เช่น ต้องการภาพยนตร์ประเภทแอคชั่น เป็นต้น เพื่อความสะดวกรวดเร็วในการค้นหา

ชื่อรูทีน GETDETAILBOOK

หน้าที่ ทำการแสดงผลข้อมูลรายละเอียดของหนังสือ ที่ผู้ใช้ได้ระบุหนังสือเล่มที่ต้องการจะทราบรายละเอียด

ถูกเรียกใช้เมื่อ ต้องการที่จะทราบรายละเอียดของหนังสือที่ระบุ

รายละเอียด จะทำการแสดงผลข้อมูลรายละเอียดของหนังสือ เช่น ชื่อหนังสือ ชื่อผู้แต่ง ภาพหน้าปก เป็นต้น ที่ผู้ใช้ได้ระบุหนังสือเล่มที่ต้องการจะทราบรายละเอียด โดยผู้ใช้จะระบุหนังสือที่ต้องการได้จากผลที่ได้จากการค้นหาแบบคร่าว ๆ

ชื่อรูทีน GETDETAILMUSIC

หน้าที่ ทำการแสดงผลข้อมูลรายละเอียดของเพลง ที่ผู้ใช้ได้ระบุเพลงที่ต้องการจะทราบรายละเอียด

ถูกเรียกใช้เมื่อ ต้องการที่จะทราบรายละเอียดของเพลงที่ระบุ

รายละเอียด จะทำการแสดงผลข้อมูลรายละเอียดของเพลง เช่น ชื่อเพลง ศิลปิน ตัวอย่างเพลง เป็นต้น ที่ผู้ใช้ได้ระบุเพลงที่ต้องการจะทราบรายละเอียด โดยผู้ใช้จะระบุเพลงที่ต้องการได้จากผลที่ได้จากการค้นหาแบบคร่าว ๆ

ชื่อรูทีน GETDETAILMOVIE

หน้าที่ ทำการแสดงผลข้อมูลรายละเอียดของภาพยนตร์ ที่ผู้ใช้ได้ระบุภาพยนตร์เรื่องที่ต้องการจะทราบรายละเอียด

ถูกเรียกใช้เมื่อ ต้องการที่จะทราบรายละเอียดของภาพยนตร์ที่ระบุ

รายละเอียด จะทำการแสดงผลข้อมูลรายละเอียดของภาพยนตร์ เช่น ชื่อภาพยนตร์ นักแสดง ตัวอย่างภาพยนตร์ เป็นต้น ที่ผู้ใช้ได้ระบุภาพยนตร์ที่ต้องการจะทราบรายละเอียด โดยผู้ใช้จะระบุภาพยนตร์ที่ต้องการได้จากผลที่ได้จากการค้นหาแบบคร่าว ๆ

5.2.5.2 รูทีนที่ทำงานให้กับผู้ดูแลห้องสมุด

ชื่อรูทีน INSERTBOOK

หน้าที่ ทำการเพิ่มข้อมูลของหนังสือที่จะจัดเก็บในห้องสมุดลงไปในฐานข้อมูล

ถูกเรียกใช้เมื่อ ต้องการที่จะเพิ่มหนังสือเล่มใหม่เข้าไปในห้องสมุด

รายละเอียด จะทำการเพิ่มข้อมูลต่าง ๆ ของหนังสือที่จะจัดเก็บในห้องสมุดโดยมีการระบุถึงรายละเอียดต่าง ๆ ของหนังสือแต่ละเล่ม เช่น ชื่อผู้แต่ง รหัสหนังสือ และทำการระบุที่อยู่ของหนังสือด้วยว่าอยู่ในตู้หนังสือตู้ใด ชั้นใดและในส่วนของใดของตู้ เช่น ซ้าย ขวาหรือกลางของตู้ ในการเพิ่มข้อมูลเข้าไปจะมีตรวจสอบว่าหนังสือที่เพิ่มเข้าไปมีรหัสของหนังสือซ้ำกับที่มีอยู่แล้วหรือไม่อีกด้วยโดยจะมีการคืนค่า ถ้าให้เพิ่มเข้าไปก็จะคืนค่า “Y” และถ้าไม่ให้เพิ่มเข้าไปจะคืนค่า “N”

ชื่อรูทีน INSERTMUSIC

หน้าที่ ทำการเพิ่มข้อมูลของเพลงที่จะจัดเก็บในห้องสมุดลงไปในฐานข้อมูล

ถูกเรียกใช้เมื่อ ต้องการที่จะเพิ่มเพลงเข้าไปในห้องสมุด

รายละเอียด จะทำการเพิ่มข้อมูลต่าง ๆ ของเพลงที่จะจัดเก็บในห้องสมุดโดยมีการระบุถึงรายละเอียดต่าง ๆ ของเพลงแต่ละเพลง เช่น ชื่อเพลง ศิลปิน รหัสเพลง และทำการระบุที่อยู่ของเพลงด้วยว่าอยู่ในตู้เก็บเพลงตู้ใด ชั้นใดและในส่วนของใดของตู้ เช่น ซ้าย ขวาหรือกลางของตู้ ในการเพิ่มข้อมูลเข้าไปจะมีตรวจสอบว่าเพลงที่เพิ่มเข้าไปมีรหัสของเพลงซ้ำกับที่มีอยู่แล้วหรือไม่อีกด้วยโดยจะมีการคืนค่า ถ้าให้เพิ่มเข้าไปก็จะคืนค่า “Y” และถ้าไม่ให้เพิ่มเข้าไปจะคืนค่า “N”

ชื่อรูทีน INSERTMOVIE

หน้าที่ ทำการเพิ่มข้อมูลของภาพยนตร์ที่จะจัดเก็บในห้องสมุดลงไปในฐานข้อมูล

ถูกเรียกใช้เมื่อ ต้องการที่จะเพิ่มภาพยนตร์เรื่องใหม่เข้าไปในห้องสมุด

รายละเอียด จะทำการเพิ่มข้อมูลต่าง ๆ ของภาพยนตร์ที่จะจัดเก็บในห้องสมุดโดยมีการระบุถึงรายละเอียดต่าง ๆ ของภาพยนตร์แต่ละเรื่อง เช่น ชื่อภาพยนตร์ ชื่อนักแสดง รหัสภาพยนตร์ และทำการระบุที่อยู่ของภาพยนตร์ด้วยว่าอยู่ในตู้เก็บภาพยนตร์ตู้ใด ชั้นใดและในส่วนของใดของตู้ เช่น ซ้าย ขวาหรือกลางของตู้ ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเพิ่มข้อมูลเข้าไปจะมีตรวจสอบว่าภาพยนตร์ที่เพิ่มเข้าไปมีรหัสของภาพยนตร์ซ้ำกับที่มีอยู่แล้วหรือไม่ อีกด้วยโดยจะมีการคืนค่า ถ้าให้เพิ่มเข้าไปก็จะคืนค่า “Y” และถ้าไม่ให้เพิ่มเข้าไปจะคืนค่า “N”

ชื่อรุ่นที่ SEARCHUPDATEBOOK

หน้าที่ ทำการค้นหาข้อมูลของหนังสือในห้องสมุดเพื่อที่จะปรับปรุงเปลี่ยนแปลง

ถูกเรียกใช้เมื่อ ต้องการที่จะปรับปรุงเปลี่ยนแปลงข้อมูลของหนังสือในห้องสมุด

รายละเอียด จะทำการค้นหาข้อมูลรายละเอียดต่าง ๆ ของหนังสือในห้องสมุด เพื่อแสดงข้อมูลที่มีอยู่แล้วของหนังสือให้กับผู้ดูแลห้องสมุด ได้แก้ไขปรับปรุงหรือเปลี่ยนแปลง ในการค้นหาที่เพียงแต่ส่งค่าของ BOOKID เข้ามา เพื่อทำการค้นหารายละเอียดของหนังสือ

ชื่อรุ่นที่ SEARCHUPATEMUSIC

หน้าที่ ทำการค้นหาข้อมูลของเพลงในห้องสมุดเพื่อที่จะปรับปรุงเปลี่ยนแปลง

ถูกเรียกใช้เมื่อ ต้องการที่จะปรับปรุงเปลี่ยนแปลงข้อมูลของเพลงในห้องสมุด

รายละเอียด จะทำการค้นหาข้อมูลรายละเอียดต่าง ๆ ของเพลงในห้องสมุด เพื่อแสดงข้อมูลที่มีอยู่แล้วของเพลงให้กับผู้ดูแลห้องสมุด ได้แก้ไขปรับปรุงหรือเปลี่ยนแปลง ในการค้นหาที่เพียงแต่ส่งค่าของ MUSICID เข้ามา เพื่อทำการค้นหารายละเอียดของเพลง

ชื่อรุ่นที่ SEARCHUPATEMOVIE

หน้าที่ ทำการค้นหาข้อมูลของภาพยนตร์ในห้องสมุดเพื่อที่จะปรับปรุงเปลี่ยนแปลง

ถูกเรียกใช้เมื่อ ต้องการที่จะปรับปรุงเปลี่ยนแปลงข้อมูลของภาพยนตร์ในห้องสมุด

รายละเอียด จะทำการค้นหาข้อมูลรายละเอียดต่าง ๆ ของภาพยนตร์ในห้องสมุด เพื่อแสดงข้อมูลที่มีอยู่แล้วของภาพยนตร์ให้กับผู้ดูแลห้องสมุด ได้แก้ไขปรับปรุงหรือเปลี่ยนแปลง ในการค้นหาที่เพียงแต่ส่งค่าของ MOVIEID เข้ามา เพื่อทำการค้นหารายละเอียดของภาพยนตร์

ชื่อรุ่นที่ SEARCHDETAILBOOK

หน้าที่ ทำการค้นหาข้อมูลของหนังสือในห้องสมุด เพื่อให้ผู้ดูแลห้องสมุดได้ทราบว่าหนังสืออะไรบ้างอยู่ในห้องสมุด

ถูกเรียกใช้เมื่อ ต้องการที่จะทราบว่าหนังสืออะไรบ้างอยู่ในห้องสมุด

รายละเอียด จะทำการค้นหาข้อมูลรายละเอียดต่าง ๆ ของหนังสือในห้องสมุด เพื่อแสดงข้อมูลให้ผู้ดูแลห้องสมุดได้ทราบว่าหนังสือเล่มใดบ้างอยู่ในห้องสมุดและอยู่ในตำแหน่งใดในตู้หนังสือ โดยในการค้นหา นั้น สามารถจำกัดลักษณะของหนังสือในการค้นหาได้ เช่น ต้องการหนังสือประเภทคอมพิวเตอร์ เป็นต้น เพื่อความสะดวกรวดเร็วในการค้นหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อรุ่น SEARCHDETAILMUSIC

หน้าที่ ทำการค้นหาข้อมูลของเพลงในห้องสมุด เพื่อให้ผู้ดูแลห้องสมุดได้ทราบว่ามิเพลงอะไรบ้างอยู่ในห้องสมุด

ถูกเรียกใช้เมื่อ ต้องการที่จะทราบว่ามิเพลงอะไรบ้างอยู่ในห้องสมุด

รายละเอียด จะทำการค้นหาข้อมูลรายละเอียดต่าง ๆ ของเพลงในห้องสมุด เพื่อแสดงข้อมูลให้ผู้ดูแลห้องสมุดได้ทราบว่ามิเพลงใดบ้างอยู่ในห้องสมุดและอยู่ในตำแหน่งใดในตู้เก็บเพลง โดยในการค้นหานี้สามารถจำกัดลักษณะของเพลงในการค้นหาได้ เช่น ต้องการเพลงประเภทเพลงร็อก เป็นต้น เพื่อความสะดวกรวดเร็วในการค้นหา

ชื่อรุ่น SEARCHDETAILMOVIE

หน้าที่ ทำการค้นหาข้อมูลของภาพยนตร์ในห้องสมุด เพื่อให้ผู้ดูแลห้องสมุดได้ทราบว่ามิภาพยนตร์เรื่องใดบ้างอยู่ในห้องสมุด

ถูกเรียกใช้เมื่อ ต้องการที่จะทราบว่ามิภาพยนตร์เรื่องใดบ้างอยู่ในห้องสมุด

รายละเอียด จะทำการค้นหาข้อมูลรายละเอียดต่าง ๆ ของภาพยนตร์ในห้องสมุด เพื่อแสดงข้อมูลให้ผู้ดูแลห้องสมุดได้ทราบว่า มิภาพยนตร์เรื่องใดบ้างอยู่ในห้องสมุดและอยู่ในตำแหน่งใดในตู้เก็บภาพยนตร์ โดยในการค้นหานี้ สามารถจำกัดลักษณะของภาพยนตร์ในการค้นหาได้ เช่น ต้องการภาพยนตร์ประเภทแอคชั่น เป็นต้น เพื่อความสะดวกรวดเร็วในการค้นหา

ชื่อรุ่น UPDATEBOOK

หน้าที่ ทำการปรับปรุงเปลี่ยนแปลงข้อมูลรายละเอียดของหนังสือในห้องสมุด

ถูกเรียกใช้เมื่อ ต้องการที่จะปรับปรุงเปลี่ยนแปลงข้อมูลรายละเอียดของหนังสือในห้องสมุด

รายละเอียด จะทำการปรับปรุงเปลี่ยนแปลงข้อมูลรายละเอียดของหนังสือ และตำแหน่งที่อยู่ของหนังสือในตู้หนังสือที่อยู่ในห้องสมุด

ชื่อรุ่น UPDATEMUSIC

หน้าที่ ทำการปรับปรุงเปลี่ยนแปลงข้อมูลรายละเอียดของเพลงในห้องสมุด

ถูกเรียกใช้เมื่อ ต้องการที่จะปรับปรุงเปลี่ยนแปลงข้อมูลรายละเอียดของเพลงในห้องสมุด

รายละเอียด จะทำการปรับปรุงเปลี่ยนแปลงข้อมูลรายละเอียดของเพลง และตำแหน่งที่อยู่ของเพลงในตู้เก็บเพลงที่อยู่ในห้องสมุด

ชื่อรุ่น UPDATEMOVIE

หน้าที่ ทำการปรับปรุงเปลี่ยนแปลงข้อมูลรายละเอียดของภาพยนตร์ในห้องสมุด

ถูกเรียกใช้เมื่อ ต้องการที่จะปรับปรุงเปลี่ยนแปลงข้อมูลรายละเอียดของภาพยนตร์ในห้องสมุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

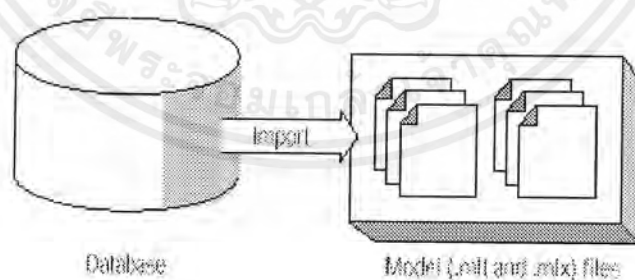
รายละเอียด จะทำการปรับปรุงเปลี่ยนแปลงข้อมูลรายละเอียดของภาพยนตร์ และตำแหน่งที่อยู่ของภาพยนตร์ในตู้เก็บภาพยนตร์ที่อยู่ในห้องสมุด

5.3 Data Director

การติดต่อระหว่างโปรแกรมประยุกต์กับฐานข้อมูลจะติดต่อผ่านค่าตัวไดเรคเตอร์ (Data Director) ค่าตัวไดเรคเตอร์เป็นทูลที่ทางอินฟอร์มิทิกส์ยูนิเวอร์แซลเซิร์ฟเวอร์ มีมาให้สำหรับการติดต่อระหว่างฐานข้อมูลกับวิซวลเบสิกโดยเฉพาะ การใช้งานค่าตัวไดเรคเตอร์จะมี 2 วิธี คือ การใช้งานแบบกราฟิกซึ่งเมื่อทำการติดตั้งค่าตัวไดเรคเตอร์ ตัวค่าตัวไดเรคเตอร์ คอมโพเนนท์ต่างๆจะถูกนำเข้าไว้ในวิซวลเบสิกทุกครั้งที่เราเรียกใช้จะปรากฏค่าตัวไดเรคเตอร์ขึ้นมาเสมอซึ่งจะช่วยให้เราสามารถนำข้อมูลขึ้นมาแสดงบนฟอร์มได้ง่ายโดยไม่ต้องมีการเขียนโค้ดใด ๆ เลย อีกวิธีเป็นการเรียกใช้ API ของค่าตัวไดเรคเตอร์ในการติดต่อกับฐานข้อมูล วิธีนี้จะต้องใช้กับการโปรแกรมมิ่งเท่านั้น

การทำงานของค่าตัวไดเรคเตอร์ จะทำการอิมพอร์ตโมเดลของฐานข้อมูลเข้ามาซึ่งจะทำให้โปรแกรมประยุกต์รู้โครงสร้างภายในฐานข้อมูล โดยในการสร้างโปรแกรมประยุกต์เพื่อทำการติดต่อฐานข้อมูลหนึ่งจะต้องใช้งานค่าตัวไดเรคเตอร์ออบเจกต์ (Data Director Objects) หรือ ดีดีโอ (DDO) ซึ่งเราจะต้องทำการสร้างออบเจกต์ oEngine ให้กับทุก ๆ โปรแกรมประยุกต์ oEngine จะทำหน้าที่เป็นรูท (Root) ให้กับทุก ๆ ดีดีโอ แล้วทำการสร้างค่าตัวกรุป (Datagroup) ซึ่งเป็นตัวที่จัดการรวมข้อมูลไว้ด้วยกัน โดยทำการสร้างออบเจกต์ oDatagroup ขึ้นมา

การใช้งานข้อมูลในฐานข้อมูลหากเราต้องการทำการคิวรีข้อมูลต้องทำการสร้างเวอร์ชวลเทเบิล (Virtual Table) เพื่อทำการเก็บผลที่ได้จากการคิวรีเราสามารถใส่คำสั่ง SQL ได้โดยตรง โดยการสร้างเวอร์ชวลเทเบิลจะต้องทำการสร้างออบเจกต์ oTable



รูปที่ 5-8 การอิมพอร์ตโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

Stored Procedure Language

ในบทนี้จะกล่าวถึง Stored Procedure Language ว่าคืออะไร มีประโยชน์อย่างไร การสร้างและการเรียกใช้งาน

6.1 บทนำ

Stored Procedure Language หรือ SPL เป็นภาษาที่มีความสามารถของภาษา SQL และได้เพิ่มความสามารถในส่วนการควบคุมการทำงาน เช่น การทำลูป การทำทางเลือก เข้าไป โดย SPL เป็นรูทีนที่ฝังอยู่กับระบบจัดการฐานข้อมูล (Database Management System : DBMS) ข้อดีของการมีรูทีนคือสามารถรวบรวมคำสั่งภาษา SQL หลายๆ คำสั่งไว้ด้วยการเรียกใช้รูทีนเพียงครั้งเดียว (Encapsulate Multiple SQL Statement) และขยายขีดความสามารถของบิวท์อินฟังก์ชันที่มีอยู่เดิม (Extend Function Built-In Data Type)

คุณสมบัติที่สำคัญอีกประการของ SPL คือความสามารถในการเข้าไปจัดการแต่ละสมาชิกของคอลเลกชันไทป์

6.2 การสร้าง SPL รูทีน

สามารถสร้าง SPL รูทีนได้ 2 ประเภท คือ โพรซีเจอร์ซึ่งไม่มีการส่งคืนค่าข้อมูลและฟังก์ชันซึ่งจะต้องมีการกำหนดการส่งคืนค่าข้อมูล การสร้าง SPL รูทีนนั้นสามารถสร้างให้มีชื่อซ้ำกันได้เนื่องจากอินฟอรมิกซ์สนับสนุนความสามารถในการทำรูทีนโอเวอร์โหลดดิ้ง (Routine Overloading) แต่ๆละรูทีนต้องมีซิกเนเจอร์ต่างกัน โดยเราสามารถกำหนดความแตกต่างของรูทีนได้หลายวิธีดังนี้

- ชนิดของรูทีนต่างกัน
- ชื่อของรูทีนต่างกัน
- จำนวนพารามิเตอร์
- ชนิดของพารามิเตอร์ของรูทีน
- ลำดับของพารามิเตอร์

6.3 กำหนดชื่อ พารามิเตอร์ และชนิดข้อมูลที่จะทำการส่งค่าคืน

เป็นการกำหนดชนิดของรูทีนที่จะทำการสร้างโดยใช้คีย์เวิร์ดเป็นตัวระบุชนิด ถ้ามีการส่งพารามิเตอร์จะต้องทำการกำหนดชื่อและชนิดของพารามิเตอร์ และในกรณีของฟังก์ชันจะต้องมีการกำหนดชนิดของข้อมูลที่จะส่งคืนด้วย ดังตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CREATE PROCEDURE New_price(percent REAL)
.
.
END PROCEDURE
หรือ
CREATE FUNCTION find_group(id INT)
RETURNING INT,REAL;
.
.
END FUNCTION

```

สำหรับการกำหนดพารามิเตอร์เราไม่สามารถกำหนดพารามิเตอร์เป็น Text หรือ Byte โดยตรงได้ ต้องใช้คีย์เวิร์ด REFERENCES ในการอ้างอิงพารามิเตอร์ที่เป็นข้อมูลชนิด Text หรือ Byte ดังกล่าว ตามตัวอย่าง

```
CREATE PROCEDURE Proc1(lo_text REFERENCES TEXT)
```

ใน SPL จะมีการสร้างบล็อก(Block) การทำงานภายในโดยใช้คีย์เวิร์ด BEGIN และ END ส่วนการกำหนดคอมเมนต์ (Comment) ทำได้โดยการใช้ { } หรือ -- ดังตัวอย่าง

```

CREATE FUNCTION block_demo()           -- SPL Function
RETURNING INT;
DEFINE distance INT;
LET distance = 37;
{ Show how to create implicit statement block}
BEGIN                                   -- Begin implicitstatement block
DEFINE distance INT;
LET distance = 2;
END                                     -- End implicitstatement block
RETURN distance;
END FUNCTION;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 การประกาศค่าตัวแปรและการกำหนดค่าให้ SPL

การประกาศค่าตัวแปรทำได้โดยใช้คีย์เวิร์ด DEFINE ใน SPL รูทีมนั้น สามารถกำหนดตัวแปรขึ้นมาใช้งานได้ 2 ลักษณะ คือ ตัวแปรโกลบอล (Global Variable) และตัวแปรโลคอล (Local Variable) โดยตัวแปรโกลบอลจะสามารถใช้งานร่วมกันได้ในหลายรูทีนเพราะจะทำการจองเนื้อที่บนหน่วยความจำไว้และจะต้องทำการกำหนดค่าดีฟอลท์(Default)ไว้ แต่ตัวแปรโลคอลจะสามารถใช้งานได้เพียงบน SPL นั้นๆ จะถูกเคลียร์ค่าทุกครั้งที่มีการใช้งานรูทีมนั้นสิ้นสุดลง และไม่สามารถกำหนดค่าของข้อมูลเป็นดีฟอลท์ (Default) ได้ ตัวอย่างการประกาศตัวแปร โลคอลชนิดต่างๆ

```

DEFINE today DATETIME YEAR TO DAY;    -- Built in type
DEFINE b REFERENCES BYTE;              -- Simple large object
DEFINE a COLLECTION;                  -- Collection type
DEFINE c SET                            -- Collection Type
CREATE ROW TYPE zip_1                  -- Named row type
(z_code CHAR(5),                       -- Unnamed row type
 z_suffix CHAR(4));
DEFINE manager ROW ( nameVARCHAR(30),
                    department VARCHAR(30),
                    salaryINTEGER

```

การประกาศตัวแปรโกลบอลจะต้องมีคีย์เวิร์ด GLOBAL และการกำหนดค่า DEFAULT เพิ่มเข้ามาดังตัวอย่าง

```
DEFINE GLOBAL gvar INT DEFAULT 2;
```

การกำหนดค่าให้กับตัวแปร 4 ลักษณะดังนี้

I. LET เป็นการกำหนดตัวแปรให้กับค่าใดๆซึ่งจะอยู่ในรูปของ Expression

```

LET a = 5;
LET b = 6; LET c = 10;
LET a,b = 10,c+d;
LET a,b = (SELECT cola,colb FROM tab1 WHERE cola=10);
LET d = func1(x,y);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LET a = ROW ('A Street', 'Nowhere', 'AA',
ROW(NULL, NULL))::address_t
```

2. SELECT...INTO.. เป็นการกำหนดค่าตัวแปรโดยทำการ Select มาจากฐานข้อมูล แล้วทำการเก็บค่าลงไปในตัวแปร
3. EXECUTE FUNCTION...INTO... เป็นการกำหนดตัวแปรโดยใช้ค่าที่ได้จากฟังก์ชัน เมื่อทำการเอ็กซีคิวต์ฟังก์ชันค่าที่รีเทิร์นกลับมาจะเก็บไว้ในตัวแปร
4. CALL...RETURNING... เป็นการกำหนดตัวแปรโดยการคืนค่าจากโพรซีเจอร์

ตัวอย่างการใช้การกำหนดค่าวิธีอื่นๆ

```
SELECT fname, lname INTO a, b FROM customer
WHERE customer_num = 101

EXECUTE FUNCTION read_address('Smith')
INTO p_fname, p_lname, p_add, p_city, p_state, p_zip;

CALL read_address('Smith')
RETURNING p_fname, p_lname, p_add, p_city, p_state, p_zip;
```

6.5 Foreach และ Cursor

ใน SPL มีการใช้เคอร์เซอร์ (Cursor) เนื่องมาจากการกำหนดค่าในตัวแปรตัวหนึ่ง เมื่อมีการคืนค่ากลับขึ้นมาจะมีค่ามากกว่าหนึ่งค่าถูกส่งมา จึงมีคำสั่ง Foreach มาจัดการกับแต่ละ Row ที่ถูกส่งคืนมา เช่น ในคำสั่ง Select, Execute และ Call ซึ่งในการใช้งาน Foreach และเคอร์เซอร์นี้ สามารถย่อยใช้ได้อีกภายใน Foreach นั้น SPL routine ซึ่งรีเทิร์นกลุ่มของ Row จะเรียกว่า เคอร์เซอร์รูทีน (Cursor Routine)

ตัวอย่างการใช้งาน Foreach

```
CREATE PROCEDURE increase_by_pct( pct INTEGER )
DEFINE s INTEGER;
FOREACH sal_cursor FOR
SELECT salary INTO s FROM employee
WHERE salary > 35000
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LET s = s + s * ( pct/100 );
UPDATE employee SET salary = s
WHERE CURRENT OF sal_cursor;

END FOREACH
END PROCEDURE;

```

การใช้ Foreach ในการใช้งานคอลเลกชันไทม์ทำได้โดย Select เอาคอลัมน์ที่เป็นคอลเลกชันไทม์ขึ้นมาเก็บในตัวแปรซึ่งตัวแปรนั้นจะเปรียบเสมือนเทเบิลที่เป็นสมาชิกของคอลเลกชันไทม์แล้วใช้ Foreach เข้าไปจัดการกับแต่ละสมาชิกได้

6.6 Flow Control

Flow Control ใน SPL มี 2 ชนิดคือ

- IF-ELIF-ELSE ใช้ในการตรวจเงื่อนไขแล้วแยกทำแต่ละส่วนตามเงื่อนไข ตัวอย่างการใช้ IF-ELIF-ELSE

```

CREATE FUNCTION str_compare( str1 CHAR(20), str2 CHAR(20))
RETURNING INTEGER;
DEFINE result INTEGER;
IF str1 > str2 THEN
    Result = 1;
ELIF str2 > str1 THEN
    result = -1;
ELSE
    result = 0;
END IF
RETURN result;
END FUNCTION;

```

- WHILE และ FOR ใช้ในการทำลูป ทำบล็อก (Block) ของสแตตเมนต์ที่โดยจะมีเงื่อนไขในการตรวจสอบเพื่อออกจากลูป ตัวอย่างการใช้งานลูป WHILE และ FOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CREATE PROCEDURE test_rows( num INT )
    DEFINE i INTEGER;
    LET i = 1;
    WHILE i < num
        INSERT INTO table1 (numbers) VALUES (i);
        LET i = i + 1;
    END WHILE;
END PROCEDURE;

```

```

FOR i = 1 TO 10
    IF i = 5 THEN
        CONTINUE FOR;
    .
    .
    ELIF i = 8 THEN
        EXIT FOR;
    END IF;
END FOR;

```

การรีเทิร์นค่าของ SPL มีได้ทั้งคืนมาค่าเดียวและคืนมาหลายค่าดังตัวอย่าง

```

CREATE FUNCTION increase_by_pct(amt DECIMAL, pct DECIMAL)
    RETURNING DECIMAL;
    DEFINE result DECIMAL;
    LET result = amt + amt * (pct/100);
    RETURN result;
END FUNCTION;

```

```

CREATE FUNCTION b_date_2( num INTEGER )
    RETURNING VARCHAR(30), DATE;
    DEFINE n VARCHAR(30);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DEFINE b DATE;
FOREACH cursor1 FOR
    SELECT name, bdate INTO n, b FROM person
        WHERE emp_no > num;
    RETURN n, b WITH RESUME;
END FOREACH
END FUNCTION;

```

ในกรณีที่การคืนค่ามีมากกว่าหนึ่งเช่นดังตัวอย่างต้องทำการประกาศตัวแปรไว้ และถ้าไม่มี Row Return ค่าของตัวแปรที่ถูกส่งคืนกลับมาจะเป็น Null

6.7 การจัดการกับ Collection Type

การจัดการกับข้อมูลที่เป็นคอลเลกชันมีขั้นตอนดังนี้ ประกาศตัวแปรคอลเลกชันประกาศตัวแปรที่จะทำการเก็บสมาชิกแต่ละตัวในคอลเลกชันแล้วทำการ Select คอลเลกชันจากฐานข้อมูล ดังตัวอย่าง

```

DEFINE P_coll SET( INTEGER NOT NULL );
DEFINE P INTEGER;
SELECT primes INTO P_coll FROM numbers
    WHERE id = 220;

```

เมื่อเราทำการสร้างตัวแปรที่เป็นตัวเก็บค่าที่รีเทิร์นจากคอลเลกชัน ตัวแปรนั้นจะเหมือนกับเป็นเทเบิลหนึ่งการจะ Insert ค่าให้กับตัวแปรคอลเลกชันนั้นทำได้ดังนี้

```
INSERT INTO TABLE (<Collection Variable>) VALUES (<Value>)
```

```
INSERT INTO TABLE (P_coll) VALUES (3);
```

การ Insert ข้อมูลที่เป็นคอลเลกชันนั้นทำได้ 2 วิธีคือใช้เคอร์เซอร์และไม่ใช้เคอร์เซอร์ การ Insert แบบไม่ใช้เคอร์เซอร์จะทำได้เมื่อคอลเลกชันที่เราใช้ต้องเป็นเซตหรือมัลติเซตเท่านั้น เพราะลำดับของสมาชิกไม่มีความสำคัญ ดังตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CREATE PROCEDURE new_emp( emp VARCHAR(30), mgr VARCHAR(30) )
  DEFINE r SET(VARCHAR(30) NOT NULL);
  SELECT direct_reports INTO r FROM manager
    WHERE mgr_name = mgr;
  INSERT INTO TABLE (r) VALUES(emp);
  UPDATE manager SET direct_reports = r
    WHERE mgr_name = mgr;
END PROCEDURE;

```

เมื่อเราทำการแก้ไขข้อมูลในตัวแปรคอลเลกชันต้องมีการแก้ไขค่าดังกล่าวลงไปบนฐานข้อมูล ดังตัวอย่าง

```

CREATE PROCEDURE shapes()
  DEFINE vertexes SET(point NOT NULL);
  DEFINE pnt point;
  SELECT definition INTO vertexes FROM polygons
    WHERE id = 207;
  FOREACH cursor1 FOR
    SELECT * INTO pnt FROM TABLE(vertexes)
    IF pnt = '(3,4)' THEN
      DELETE FROM TABLE(vertexes)
        WHERE CURRENT OF cursor1;
      EXIT FOREACH;
    ELSE
      CONTINUE FOREACH;
    END IF;
  END FOREACH
  UPDATE polygons SET definition = vertexes
    WHERE id = 207;
END PROCEDURE;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจะเข้าไป Select ข้อมูลที่เป็นคอลเลกชันต้องเริ่มด้วยการเคอร์เซอร์โดยใช้คีย์เวิร์ด FOREACH

```
FOREACH cursor1 FOR
    SELECT * INTO pnt FROM TABLE(vertexes)
    .
    .
END FOREACH
```

การอัปเดตสมาชิกในคอลเลกชันทำได้โดยการดึงข้อมูลมาเก็บในตัวแปรคอลเลกชัน แล้วทำการประกาศเคอร์เซอร์ขึ้นมา ดังตัวอย่าง

```
DEFINE s SET(INTEGER NOT NULL);
DEFINE n INTEGER;
SELECT numbers INTO s FROM orders
    WHERE order_num = 10;
FOREACH cursor1 FOR
    SELECT * INTO n FROM TABLE(s)
    IF ( n = 500 ) THEN
        UPDATE TABLE(s)(x)
            SET x = 400 WHERE CURRENT OF cursor1;
        EXIT FOREACH;
    ELSE
        CONTINUE FOREACH;
    END IF;
END FOREACH
```

เมื่อเราต้องการ จะอัปเดตข้อมูลคอลเลกชันในฐานข้อมูลโดยใช้ตัวแปรคอลเลกชันสามารถทำได้ดังตัวอย่าง

```
CREATE PROCEDURE new_report(mgr VARCHAR(30),
    old VARCHAR(30), new VARCHAR(30) )
    DEFINE s SET (VARCHAR(30) NOT NULL);
    DEFINE n VARCHAR(30);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SELECT direct_reports INTO s FROM manager
    WHERE mgr_name = mgr;
FOREACH cursor1 FOR
    SELECT * INTO n FROM TABLE(s)
    IF ( n = old ) THEN
        UPDATE TABLE(s)(x)
            SET x = new WHERE CURRENT OF cursor1;
    EXIT FOREACH;
    ELSE
        CONTINUE FOREACH;
    END IF;
END FOREACH
UPDATE manager SET mgr_name = s
    WHERE mgr_name = mgr;
END PROCEDURE;

```

6.8 การเรียกใช้รoutines

สามารถเรียกใช้ได้หลายวิธีดังนี้

- คำสั่ง EXECUTE PROCEDURE หรือ EXECUTE FUNCTION ใน DB-Access
- เรียกจาก SPL routines อื่นหรือจากรoutines ภายนอก ด้วยคำสั่ง CALL
- เรียกผ่าน SQL statement

ตัวอย่างการเรียกใช้รoutines แบบต่างๆ

```

EXECUTE PROCEDURE update_orders();
EXECUTE FUNCTION scale_rectangles( 1 07, 1.9 )
    INTO new;
CALL area(rectv.length, rectv.width) RETURNING a;
SELECT increase_by_pct(price, 20) INTO p
    FROM inventory WHERE prod_num = num;

```

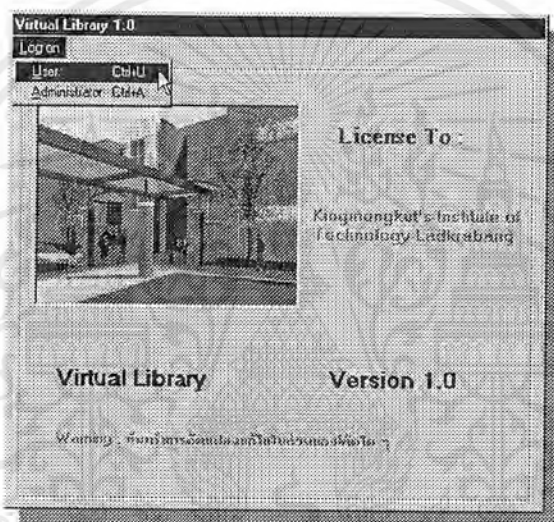
บทที่ 7

ตัวอย่างการใช้งานโปรแกรมระบบสารสนเทศห้องสมุด

โปรแกรมประยุกต์นี้แบ่งออกเป็น 2 ส่วนคือ ส่วนของผู้ใช้และ ส่วนของผู้ดูแลระบบ

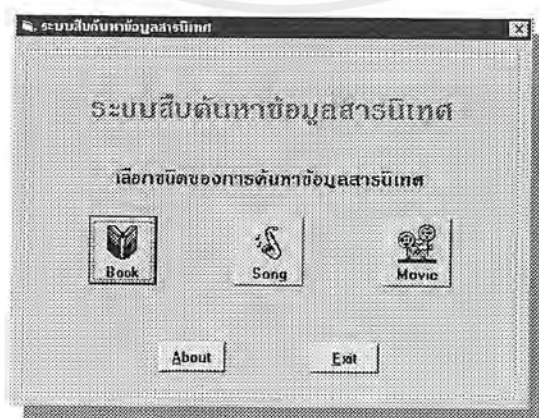
7.1 ตัวอย่างโปรแกรมของส่วนผู้ใช้

เมื่อเปิดโปรแกรมจะปรากฏดังรูปที่ 7-1



รูปที่ 7-1 หน้าต่างแรกเมื่อเข้าโปรแกรม

โดยโปรแกรมจะมีการ Login เข้าสู่ระบบ 2 แบบคือ แบบ User และแบบ Administrator โดยเลือกแบบ User จะปรากฏดังรูปที่ 7-2



รูปที่ 7-2 เมนู ในการเลือกค้นข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเราคปุม Book จะปรากฏดังรูปที่ 7-3

รูปที่ 7-3 การค้นหาหนังสือ

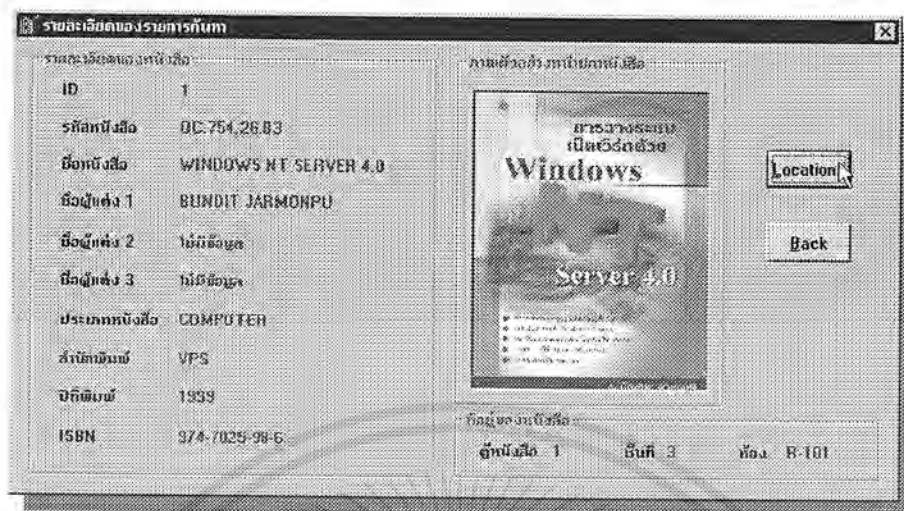
เมื่อเราป้อนข้อมูลในช่องประเภทหนังสือด้วยคำ Computer แล้วกดปุ่ม Search จะปรากฏดังรูปที่ 7-4

BOOKCODE	BOOKNAME	BOOKCATALOG	PUBLISHING	YEAR
QC.754.26.B3	WINDOWS NT SERVER	COMPUTER	VPS	1995
QC.754.87.F6	DEVELOPING TIME ORI	COMPUTER	MCGRAWHILL	1990
QC.789.56.J2	OBJECT ORIENTED MD	COMPUTER	MCGRAWHILL	1998
QC.854.25.C6	AN INTRODUCTION TO	COMPUTER	ADDISON WESLEY	1998
QC.756.84.C5	HANDBOOK OF RELATI	COMPUTER	ADDISON WESLEY	1997
QC.710.84.M4	OBJECT DATABASES TI	COMPUTER	MCGRAWHILL	1997
QC.454.63.A5	OBJECT ORIENTED DA	COMPUTER	MCGRAWHILL	1997
QC.454.26.T2	MULTIMEDIA MAKING I	COMPUTER	OSBARNE	1993
QC.789.56.F3	E-BUSINESS ROADMAP	COMPUTER	MCGRAWHILL	1998
QC.854.25.58	DIGITAL ELECTORNICS	COMPUTER	PROMPT PUBLICAT	1998
QC.756.84.52	JASMINE OBJECT DATA	COMPUTER	JASMINE	1998

รูปที่ 7-4 ผลการค้นหาหนังสือ

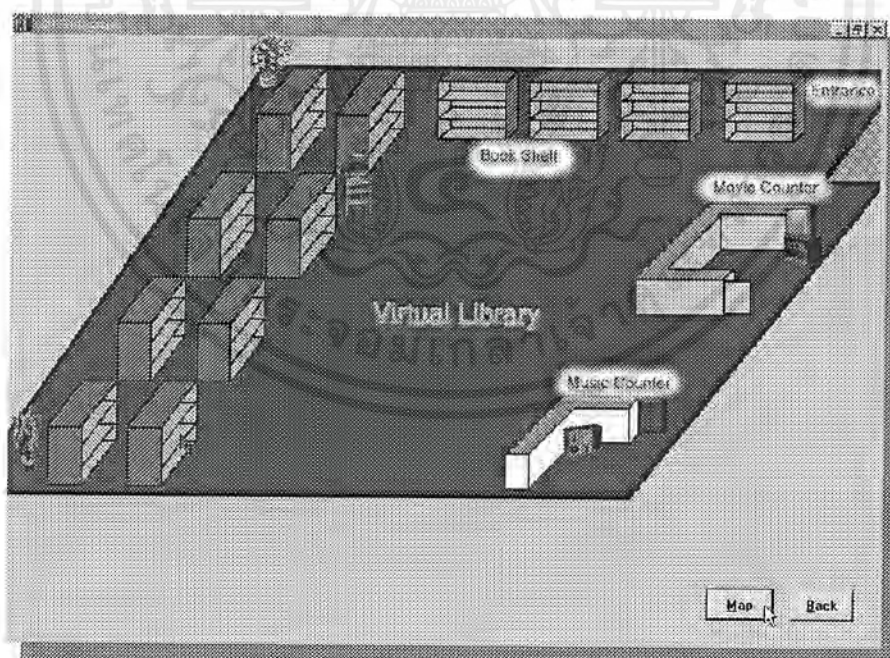
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเรากดปุ่ม Detail จะให้ข้อมูลที่ละเอียดของหนังสือเล่มนั้นดังรูปที่ 7-5



รูปที่ 7-5 รายละเอียดของหนังสือที่ค้นหาพร้อมรูปภาพ

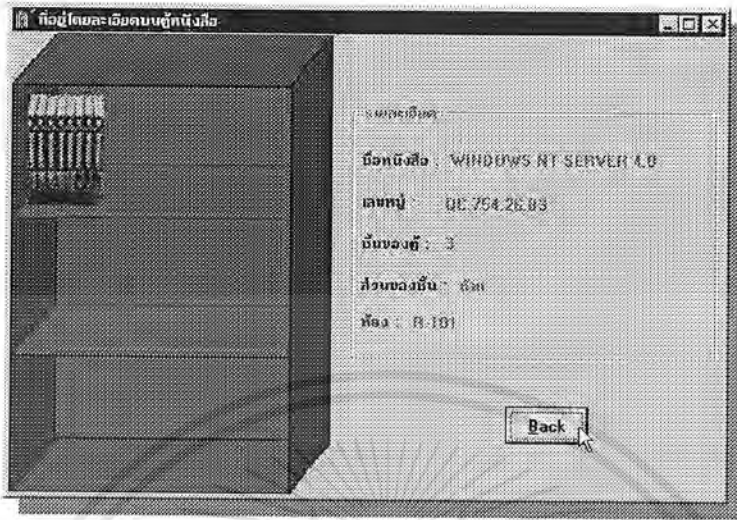
เมื่อกดปุ่ม Location จะแสดงแผนที่ห้องสมุด และเมื่อกดปุ่ม Map โปรแกรมจะเคลื่อนลูกศรแสดงที่อยู่ของหนังสือในห้องสมุดดังรูปที่ 7-6



รูปที่ 7-6 ตำแหน่งของหนังสือในห้องสมุด

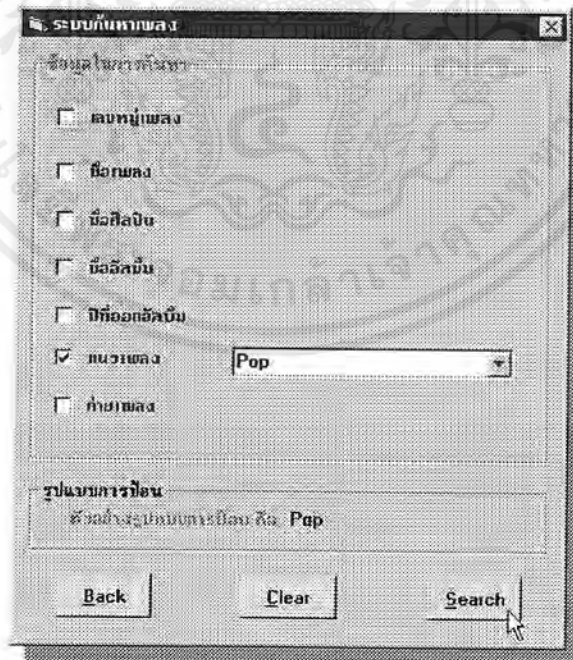
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และจะแสดงตำแหน่งของหนังสือโดยละเอียดบนตู้หนังสือดังรูปที่ 7-7



รูปที่ 7-7 ตำแหน่งของหนังสือในตู้โดยละเอียด

เราสามารถกดปุ่ม Back เพื่อกลับไปข้างหน้าที่ผ่านมาแล้วได้ และกดปุ่ม Exit เพื่อออกจากโปรแกรม ถ้าเรากดปุ่ม Music ที่เมนูเริ่มต้นจะปรากฏดังรูปที่ 7-8

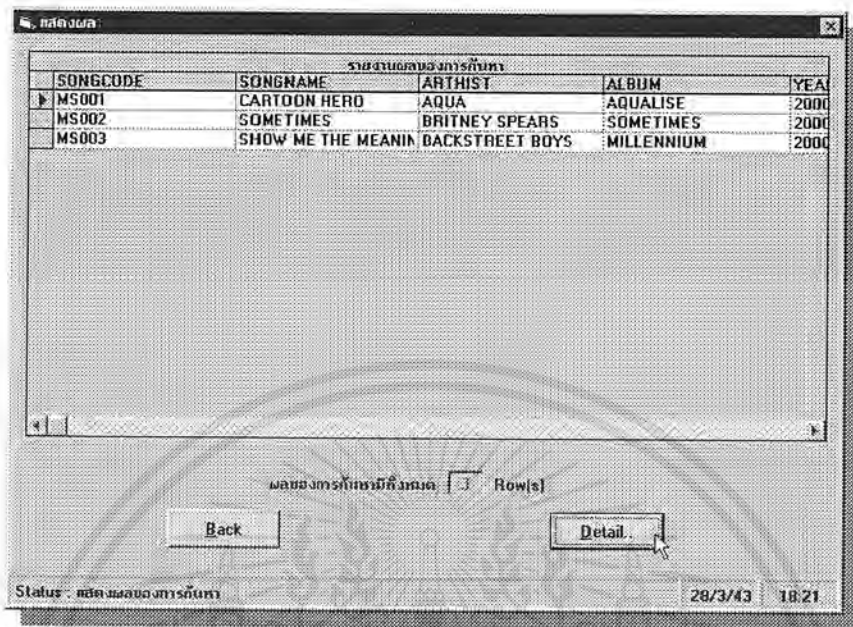


รูปที่ 7-8 เมนูในการค้นหาข้อมูลประเภทเพลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

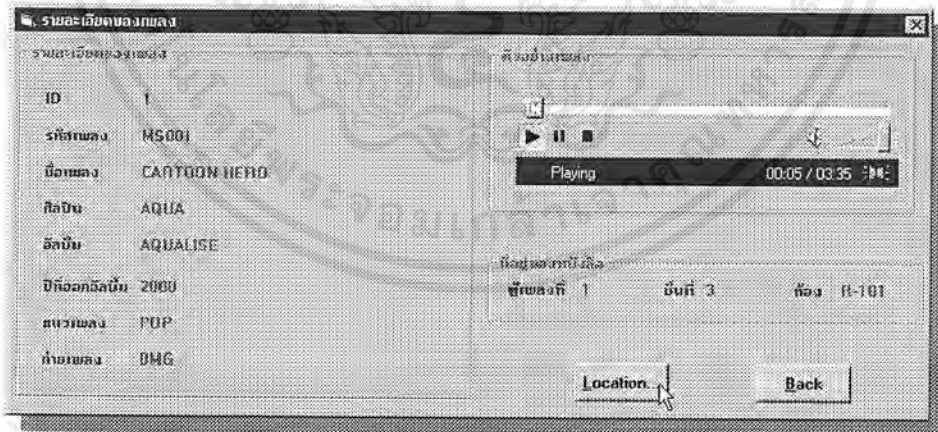
เมื่อเราป้อนข้อมูลที่ต้องการค้นหาเช่น ที่ช่อง แนวเพลง เลือก pop แล้วกดปุ่ม Search จะปรากฏดังรูปที่

7-9



รูปที่ 7-9 ผลการค้นหาเพลง

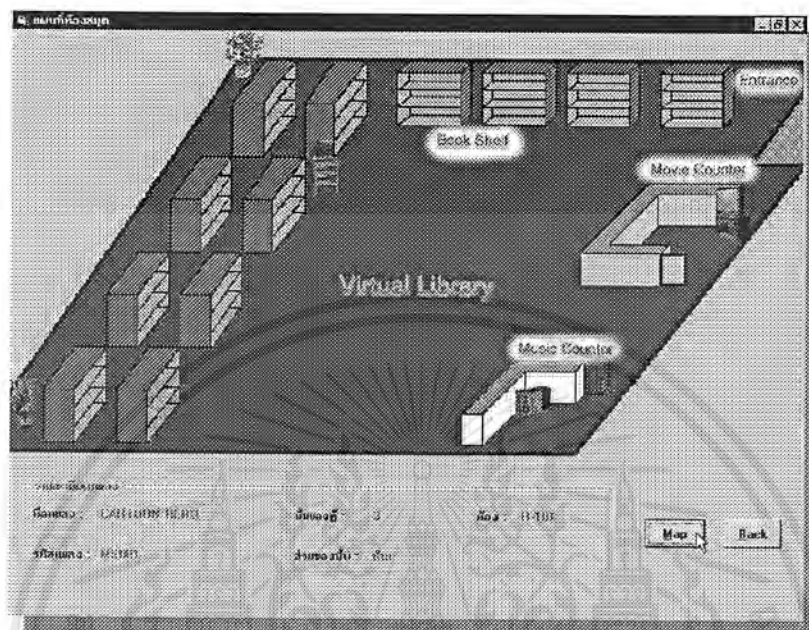
เมื่อเรากดปุ่ม Detail จะปรากฏดังรูปที่ 7-10



รูปที่ 7-10 รายละเอียดของเพลงพร้อมตัวอย่างเพลง

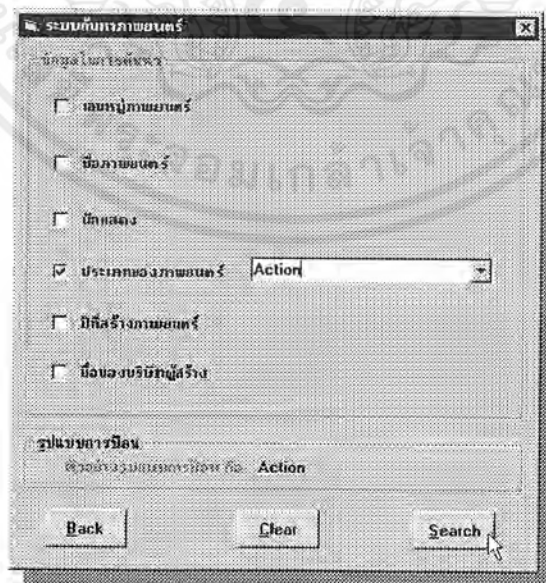
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 7-10 เราสามารถกดปุ่มเล่นเพลงที่อยู่ในส่วนของตัวอย่างเพลงเพื่อฟังเพลงที่ต้องการได้ และเมื่อเรากดปุ่ม Location โปรแกรมจะแสดงแผนที่ของห้องสมุดและเมื่อกดปุ่ม Map ลูกศร จะเคลื่อนที่ไปยังที่อยู่ที่ของเพลง ๆ นั้นดังรูปที่ 7-11



รูปที่ 7-11 ตำแหน่งของเพลงในห้องสมุด

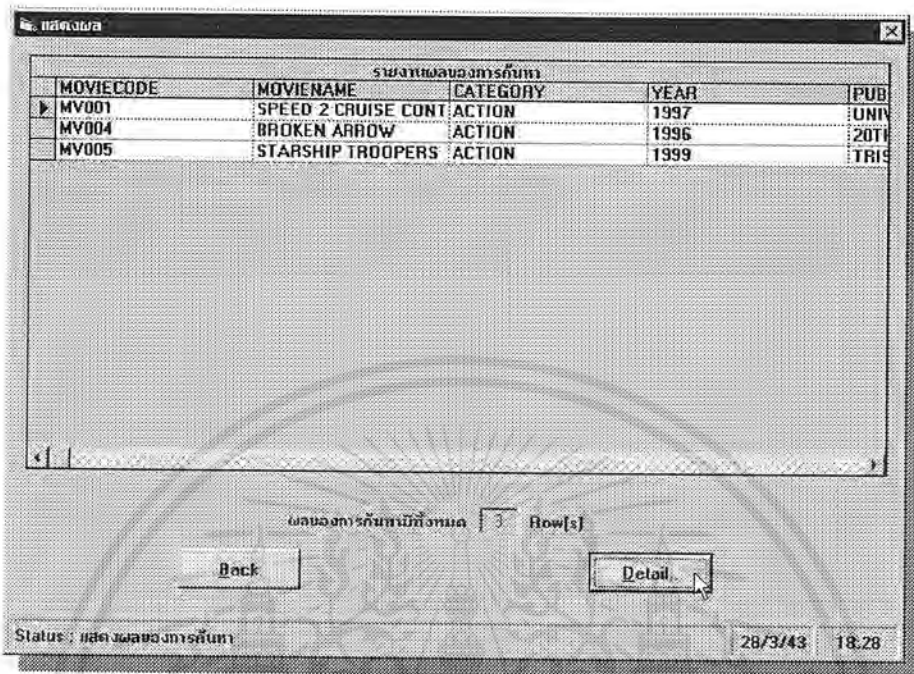
เมื่อเราเลือกเมนูค้นหาภาพยนตร์จะปรากฏดังรูปที่ 7-12



รูปที่ 7-12 หน้าจอค้นหาข้อมูลภาพยนตร์

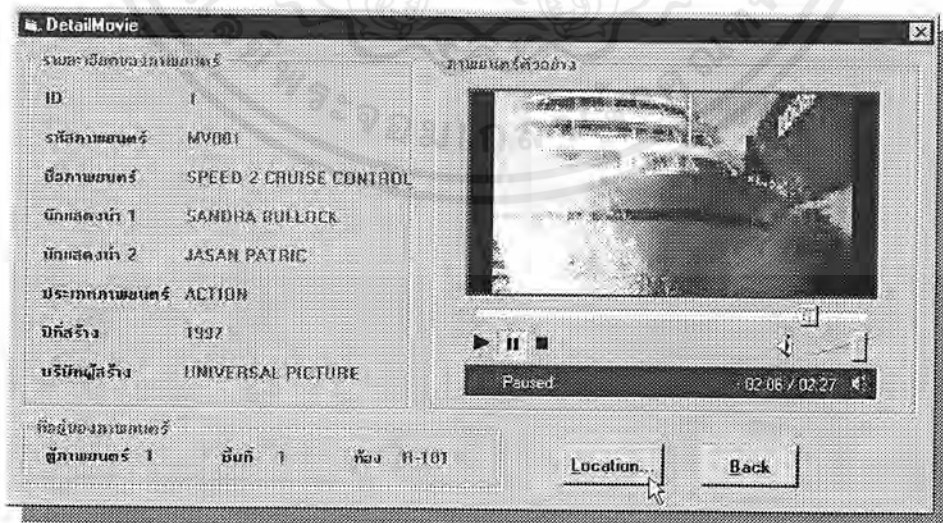
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเราป้อนข้อมูลเช่น ที่ช่อง ประเภทของภาพยนตร์ เราป้อนค่า Action แล้วกดปุ่ม Search จะปรากฏดังรูปที่ 7-13



รูปที่ 7-13 รายการของการค้นหาข้อมูลภาพยนตร์

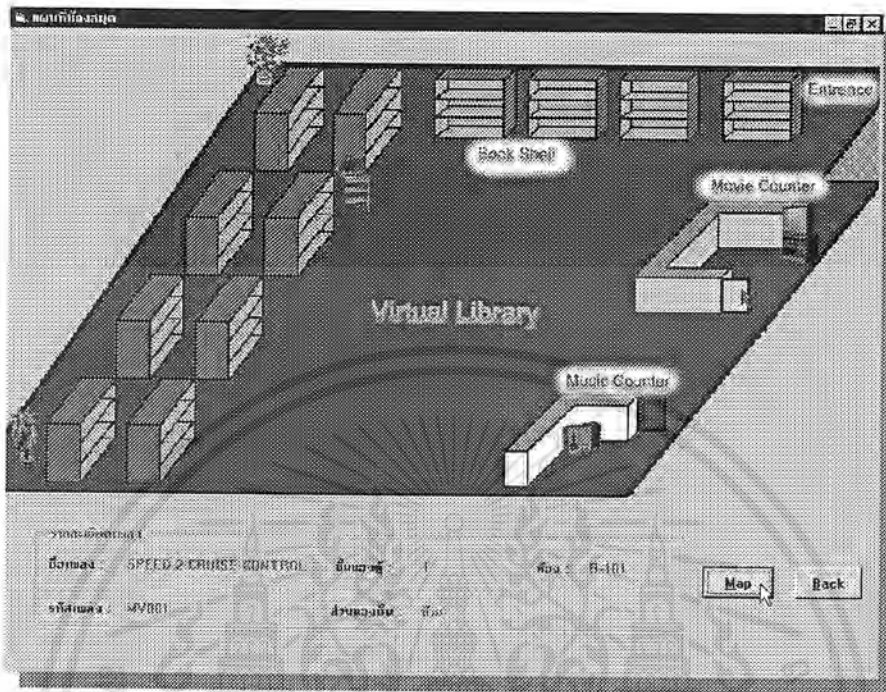
เมื่อเรากดปุ่ม Detail จะแสดงข้อมูลโดยละเอียดของภาพยนตร์ที่ค้นหา ซึ่งเราสามารถกดปุ่มเพื่อให้เห็นภาพยนตร์ที่เราค้นหานั้น ได้ดังแสดงในรูปที่ 7-14



รูปที่ 7-14 รายละเอียดของภาพยนตร์พร้อมภาพยนตร์ตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

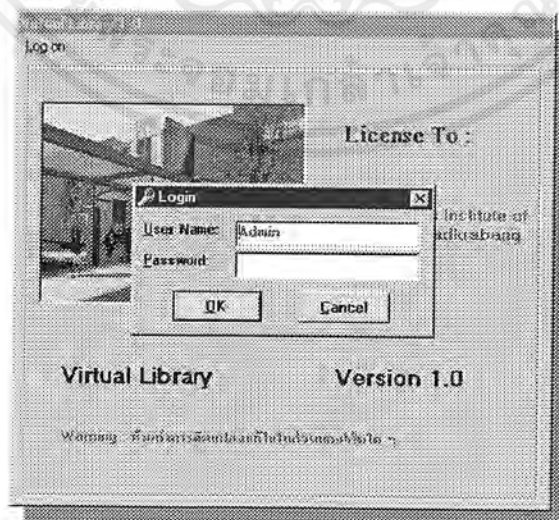
เมื่อเรากดปุ่ม Location จะแสดงแผนที่ของห้องสมุด และเมื่อกดปุ่ม Map ลูกศรจะเคลื่อนที่ไปที่ตำแหน่งของที่อยู่ของภาพยนตร์เรื่องนั้น ๆ ดังแสดงในรูปที่ 7-15



รูปที่ 7-15 ตำแหน่งของภาพยนตร์ในห้องสมุด

7.2 ตัวอย่างโปรแกรมของผู้ดูแลระบบ

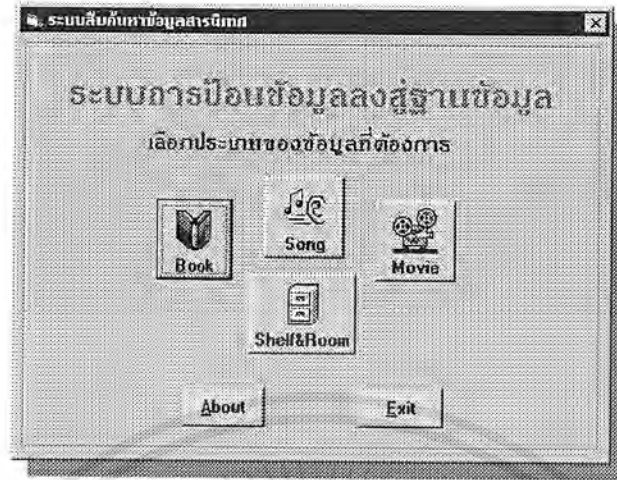
เมื่อเปิดโปรแกรมจะต้องทำการ log on แบบ Administrator ซึ่งจะปรากฏให้ใส่ User Name และ Password ดังรูปที่ 7-16



รูปที่ 7-16 หน้าต่างแรกเมื่อเรียกโปรแกรม

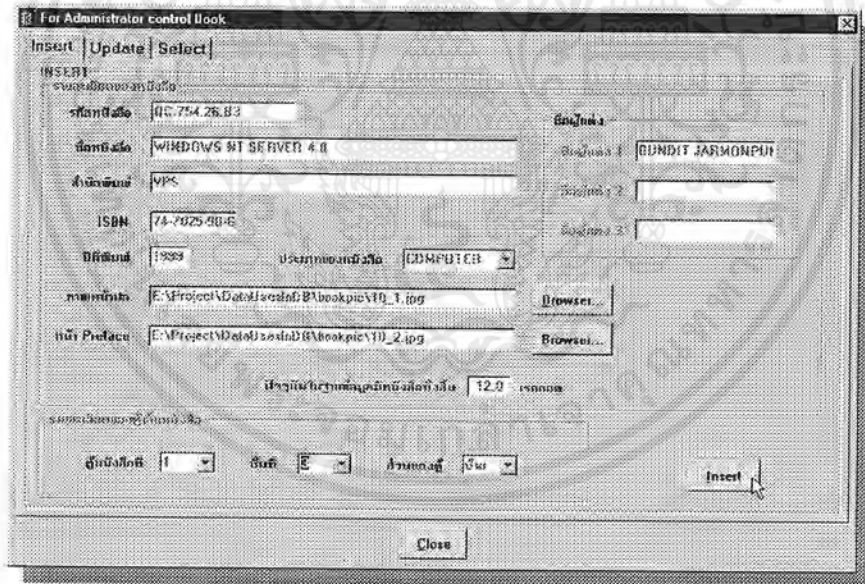
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเราป้อน User Name และ Password ที่ถูกต้องแล้ว จะปรากฏดังรูปที่ 7-17



รูปที่ 7-17 เมนูในการป้อนข้อมูล

เมื่อเราเลือก Book จะปรากฏหน้าจอ ดังรูปที่ 7-18 ซึ่งยังไม่มีข้อมูลอยู่



รูปที่ 7-18 หน้าต่างการ Insert ข้อมูลหนังสือ

จากรูปที่ 7-18 เมื่อเราป้อนข้อมูลต่าง ๆ ลงไปเสร็จแล้วเรากดปุ่ม Insert ข้อมูลที่เราป้อนก็จะถูกจัดเก็บลงในฐานข้อมูล รูปที่ 7-19 แสดงหน้าจอของการ Update โดยเราสามารถค้นหาข้อมูลที่จะแก้ไขได้โดยป้อนข้อมูลที่ต้องการแล้วกดปุ่มค้นหา หลังจากนั้นเราก็สามารถแก้ไขข้อมูลต่าง ๆ ที่มีได้ทุกฟิลด์ เมื่อแก้ไขเสร็จแล้วก็กดปุ่ม Update ข้อมูลใหม่ก็จะถูกเก็บเอาไว้ ซึ่งแสดงในรูปที่ 7-19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

For Administrators control Book

Insert | Update | Select

UPDATE

ID: 1

Author: [Field]

Title: [Field]

ISBN: [Field]

Update

รูปที่ 7-19 หน้าต่างการ Update ข้อมูลหนังสือ

เมื่อต้องการ Select ข้อมูลที่ได้ทำการป้อนไปแล้วเราสามารถใช้เมนู Select ได้ซึ่งมีหน้าจอดังรูปที่ 7-20 โดยเราสามารถค้นหาข้อมูลได้ตามฟิลด์ต่าง ๆ และสามารถลบ หรือ แก้ไขข้อมูลที่ได้จากการ Select ได้เลย

For Administrators control Book

Insert | Update | Select

SELECT

BOOKID	BOOKCODE	BOOKNAME
1	QC.754.26.B3	WINDOWS I
2	QC.754.07.H6	DEVELOPIM
3	QC.789.06.J2	OBJECT-OR
4	QC.854.26.C6	AN INTROD
5	QC.756.04.C5	HANDBOOK
6	QC.710.84.M4	OBJECT DA
7	QC.854.63.A5	OBJECT-OR
8	QC.454.26.12	MULTIMEDI
9	QC.654.07.H5	TO ENGINE
10	QC.789.56.H3	E-BUSINES
11	QC.854.25.59	DIGITAL ELI
12	QC.756.04.S2	JASMINE DR

Search

Delete

Update

รูปที่ 7-20 หน้าต่างการ Select ของหนังสือ

ซึ่งในส่วนของ Music และ Movie ก็จะมีทั้งส่วน Insert, Delete และ Update เช่นเดียวกันและมีการใช้งานเหมือนกันกับส่วนของหนังสือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

สรุปและวิจารณ์

8.1 สรุป

ในโครงการนี้ได้ทำการศึกษาหลักการของโมเดลฐานข้อมูลเชิงวัตถุสัมพันธ์ ศึกษาการออกแบบและการใช้งานฐานข้อมูลเชิงวัตถุสัมพันธ์เพื่อประยุกต์ใช้ในงานต่างๆ หลังจากมีความรู้และความเข้าใจโมเดลฐานข้อมูลแล้วได้ทำการศึกษาระบบจัดการฐานข้อมูล (Database Management System : DBMS) ที่สนับสนุนโมเดลฐานข้อมูลเชิงวัตถุสัมพันธ์

โครงการนี้ใช้อินฟอร์มิคซ์ยูนิเวอร์แซลเซิร์ฟเวอร์ (Informix Universal Server) เวอร์ชัน 9.14 เป็นระบบจัดการฐานข้อมูลที่สามารถจัดเก็บข้อมูลได้ทั้งที่เป็น โมเดลฐานข้อมูลเชิงสัมพันธ์และโมเดลฐานข้อมูลเชิงวัตถุสัมพันธ์ ศึกษาและทดลองใช้งานชนิดข้อมูลต่างๆที่สนับสนุนโมเดลฐานข้อมูลเชิงวัตถุสัมพันธ์ที่ระบบจัดการฐานข้อมูลมีให้

หลังจากมีความคุ้นเคยกับการใช้งานระบบจัดการฐานข้อมูลก็ได้ทำการทดลอง สร้างโปรแกรมประยุกต์ (Application) ที่ทำการเก็บข้อมูลประเภทมัลติมีเดีย โดยจัดทำเป็นโปรแกรมระบบสารสนเทศห้องสมุด (Virtual Library) โดยใช้คุณลักษณะของฐานข้อมูลเชิงวัตถุสัมพันธ์ในการจัดเก็บข้อมูล โดยในการพัฒนาโปรแกรมประยุกต์เริ่มจากการออกแบบฐานข้อมูลโดยยึดแนวคิดการออกแบบตามคุณลักษณะของฐานข้อมูลเชิงวัตถุสัมพันธ์ แล้วทำการสร้างรูนที่เขียนด้วย Stored Procedure Language หรือ SPL ซึ่งเป็นภาษาที่มีความสามารถของภาษา SQL โดยรูนที่ดังกล่าวจะมีไว้ให้โคลเอนที่เรียกเพื่อจัดการกับฐานข้อมูล

การติดต่อระหว่างโคลเอนกับดาต้าเบสเซิร์ฟเวอร์ไม่ได้ติดต่อกันโดยตรงแต่ทำการติดต่อกันผ่านทางดาต้าไคลเร็กเตอร์ซึ่งจะเป็นตัวที่ทำหน้าที่อิมพอร์ตโมเดลในฐานข้อมูล ให้กับโปรแกรมประยุกต์เพื่อทำให้การติดต่อกับฐานข้อมูลเป็นไปได้โดยง่าย

โปรแกรมประยุกต์ซึ่งพัฒนาโดยใช้วีซวลเบสิกมีความสามารถในการทำงานในการเรียกค้นข้อมูลประเภทมัลติมีเดียในฐานข้อมูลเพื่อนำมาแสดงผลได้ แต่มีข้อจำกัดที่ดาต้าไคลเร็กเตอร์ คือ ไม่สามารถเรียกค้นข้อมูลที่เป็นภาษาไทยได้

8.2 แนวทางการพัฒนาต่อ

ในการพัฒนาโปรแกรมโดยใช้ Visual Basic ติดต่อกับฐานข้อมูลที่ใช้ดาต้าไคลเร็กเตอร์เป็นตัวเชื่อมต่อควรจะมีการใช้ดาต้าไคลเร็กเตอร์ที่สนับสนุนภาษาไทยเพื่อความสมบูรณ์ในการพัฒนาโปรแกรมสำหรับคนไทย เนื่องจากในปัจจุบันนี้การพัฒนาโปรแกรมที่ใช้งานกันบนอินเทอร์เน็ตมีความนิยมและแพร่หลายเป็นอย่างมาก ดังนั้นแนวทางการพัฒนาต่อไปควรจะทำให้แอปพลิเคชันสามารถใช้งานบนอินเทอร์เน็ตได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] กิตติ ภัคดีวัฒนะกุล และ จำลอง ทรูอดุสาหะ : “Visual Basic 6 ฉบับโปรแกรมเมอร์”, พิมพ์ครั้งที่ 2, เลทีพี คอมพิวเตอร์ แอนด์ คอนซัลท์, กรุงเทพฯ 1999
- [2] ชาริน สิริธรรมชารี และ สุรสิทธิ์ คิวประสพศักดิ์ : “การเขียนโปรแกรมด้วย Advanced Visual Basic 6 ”, ส.เอเซียเพรส(1989), กรุงเทพฯ
- [3] บุญถาวร หงสกุล : “การจัดหมวดหมู่หนังสือระบบห้องสมุดรัฐสภาอเมริกัน”, พิมพ์ครั้งที่ 6, สำนักพิมพ์มหาวิทยาลัยรามคำแหง, กรุงเทพฯ 1987.
- [4] สัจจะ จรัสรุ่งรวีร์ : “คู่มือการสร้างแอปพลิเคชันด้วย Visual Basic 6 Basic & Advanced”, พิมพ์ครั้งที่ 2, อินโฟเพรส, กรุงเทพฯ 1999
- [5] อัมพร ทิษระ : “คู่มือการจัดหมู่ระบบหอสมุดรัฐสภาอเมริกัน”, ภาควิชาบรรณารักษศาสตร์ คณะอักษรศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, กรุงเทพฯ 1989
- [6] Excalibur Image DataBlade Module User’s Guide
- [7] Informix Client Products Installation Guide for Microsoft Windows Environments, Version 2.0
- [8] Informix -Universal Server Informix Guide to SQL: Reference, Version 9.1
- [9] Informix -Universal Server Informix Guide to SQL: Syntax, Version 9.1
- [10] Informix -Universal Server Informix Guide to SQL: Tutorial, Version 9.1
- [11] Informix -Universal Server SQL Quick Syntax Guide, Version 9.1
- [12] Informix-Data Director for Visual Basic Tutorial
- [13] Informix-Data Director for Visual Basic User Guide
- [14] Informix-Universal Server Quick Reference Guide, Version 9.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

การติดตั้ง Informix Universal Server

1. การติดตั้ง

ยูนิเวอร์แซลเซิร์ฟเวอร์ (Universal Server) เป็นผลิตภัณฑ์แบบไคลเอนท์/เซิร์ฟเวอร์ ดังนั้นยูนิเวอร์แซลเซิร์ฟเวอร์จึงแบ่งเป็น 2 ส่วน คือ ส่วนของเซิร์ฟเวอร์ และส่วนของไคลเอนท์

- ส่วนของเซิร์ฟเวอร์จะประกอบไปด้วยยูนิเวอร์แซลเซิร์ฟเวอร์ (Informix Universal Server) ที่สนับสนุน ความสามารถฐานข้อมูลเชิงวัตถุสัมพันธ์และยูทิลิตี้ (Utilities) ต่างๆ เช่น Dbaccess, Onmode, Oninit และ Onspaces เป็นต้น
- ส่วนของไคลเอนท์มีชุด (Tool) เพื่อใช้งานต่างๆ ดังนี้เช่น
 - Schema Knowledge ทำการแสดงผลข้อมูลทั้งหมดของยูนิเวอร์แซลเซิร์ฟเวอร์ เช่น รายละเอียดของ Type Table , Routines , Operator Cast , Aggregate และ Access Method
 - SQLeditor เป็น ISQL ของยูนิเวอร์แซลเซิร์ฟเวอร์ที่อยู่ฝั่งไคลเอนท์
 - DBDK DataBlade Developer Kit เป็นชุดที่ใช้พัฒนาตัวแปลโมดูล (DataBlade Module)
 - Iconnect เป็น ODBC ของยูนิเวอร์แซลเซิร์ฟเวอร์

2. การเตรียมการติดตั้ง

การติดตั้งเซิร์ฟเวอร์ของอินฟอร์มีกซ์ยูนิเวอร์แซลเซิร์ฟเวอร์ มีอยู่ 2 แพลตฟอร์ม (Platform) คือ บนยูนิกซ์ (UNIX) และวินโดวส์เอ็นที (Window NT) ได้ทำการทดลองติดตั้งบนพีซีซึ่งเลือกใช้เวอร์ชัน (Version) บนวินโดวส์เอ็นทีซึ่งมีความต้องการของระบบ ดังนี้

- วินโดวส์เอ็นที เวอร์ชัน 4.0
- โพรโตคอล TCP/IP
- หน่วยความจำแรม (RAM) 16 เมกกะไบต์
- เนื้อที่บนหน่วยความจำ (Disk Space) 40 เมกกะไบต์

3. ขั้นตอนการติดตั้ง

1. ผู้ติดตั้งต้องอยู่ในกลุ่มของ Administration Group
2. Run Setup.exe เพื่อทำการติดตั้ง
3. ใส่ Serial Number และ Cd Key
4. ใส่ Product Registration เช่นชื่อ ตำแหน่ง บริษัท ที่อยู่ และข้อมูลในการติดต่อ
5. กำหนดไดเรกทอรีที่จะทำการติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. เลือกการกำหนด บทบาท ของ DB-Admin ที่จะแยกหน้าที่กันในแต่ละงาน หรือ รวมกันใช้เพียง Account เดียว หากเลือกที่จะแยกบทบาทในการดูแลระบบจะต้องปฏิบัติตามขั้นตอนต่อไปนี้เพิ่มเติม

6.1 จัดประเภทงานของผู้ดูแลระบบ เช่น งานตรวจสอบระบบ, งานรักษาความปลอดภัยให้
กับฐานข้อมูล

6.2 ระบุชื่อผู้ตรวจสอบระบบ

6.3 ระบุชื่อผู้คอยรักษาความปลอดภัยให้กับฐานข้อมูล

7. กำหนดชื่อของ Database Server

8. กำหนด Root Dbspace ที่จะใช้สำหรับเก็บฐานข้อมูลและตารางข้อมูลต่างๆ

9. กำหนด Sbspace ซึ่งเอาไว้เก็บข้อมูลประเภท Smart large object

10. ระบุข้อมูลของอุปกรณ์ที่จะใช้เก็บข้อมูลสำรอง ได้แก่ ความจุของอุปกรณ์, ที่อยู่ของอุปกรณ์ เพื่อ
ใช้ในการกู้คืนข้อมูลหากระบบพัง

11. การติดตั้ง จะให้กำหนดรหัสผ่าน (Password) สำหรับ Account Informix เนื่องจาก Informix จะ
สร้าง Account Informix มีหน้าที่ดูแลนิเวศเวิร์กเซิร์ฟเวอร์ (DBAdmin)

12. กำหนดที่เก็บ Sqlhost File

13. เมื่อทุกอย่างเสร็จสิ้น ทำการกำหนดค่าเริ่มต้นให้กับระบบ (Initialize)

4. เขตให้ระบบฐานข้อมูลเริ่มทำงาน

4.1 ขั้นตอนการเขตให้ database server ทำงาน

4.1.1 คลิกที่ไอคอน Control Panel ใน Start menu ของ Windows NT

4.1.2 ดับเบิลคลิกที่ไอคอน Services

4.1.3 เลือก INFORMIX-Universal Server จากกล่องรายการ service

4.1.4 คลิก Start

4.2 หากเป็นการเขตให้ server ทำงานเป็นครั้งแรกต้องทำตามขั้นตอนดังนี้

4.2.1 คลิกที่ไอคอน Control Panel ใน Start menu ของ Windows NT

4.2.2 ดับเบิลคลิกที่ไอคอน Services

4.2.2 เลือก INFORMIX-Universal Server จากกล่องรายการ service

4.2.3 ใส่อักษร -iy ลงในกล่อง Startup Parameters

4.2.4 คลิก Start

4.3 ขั้นตอนการเขตให้ server ทำงานโดยอัตโนมัติ

4.3.1 คลิกที่ไอคอน Control Panel ใน Start menu ของ Windows NT

4.3.2 ดับเบิลคลิกที่ไอคอน Services

4.3.3 เลือก INFORMIX-Universal Server จากกล่องรายการ service

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4.3.4 คลิก Start และ Automatic ใน service dialog box
- 4.3.5 จากนั้นคลิก OK
- 4.3.6 ตรวจสอบว่าไม่มีข้อความใดๆ ในกล่อง Startup Parameters
- 4.3.7 คลิก Start

4.4 ขั้นตอนการเซตให้ระบบฐานข้อมูลหยุดทำงาน

- 4.4.1 คลิกที่ไอคอน Control Panel ใน Start menu ของ Windows NT
- 4.4.2 ดับเบิลคลิกที่ไอคอน Services
- 4.4.3 เลือก INFORMIX-Universal Server จากกล่องรายการ service
- 4.4.4 คลิก Stop

5. ขั้นตอนในการเซตให้เครื่องไคลเอนต์ติดต่อกับเครื่องเซิร์ฟเวอร์ได้

5.1 ให้ไปแก้ไขไฟล์ชื่อ Hosts ของเครื่องไคลเอนต์ ซึ่งอยู่ในไดเรกทอรี C:\WINNT\System32\driver\etc โดยให้เพิ่ม IP address ของเครื่องเซิร์ฟเวอร์ และตามด้วยชื่อเครื่องเซิร์ฟเวอร์ (โดยชื่อนี้ตัวอักษรตัวเล็กตัวใหญ่มีผลด้วย) ลงไปที่ท้ายสุดของไฟล์นี้ ดังเช่นตัวอย่าง 161.246.6.153 MultimediaDB

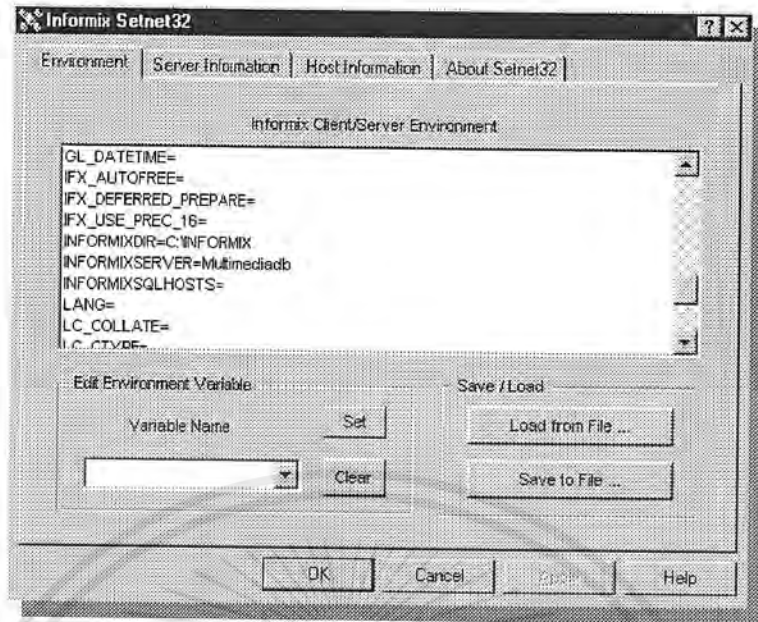
5.2 ให้ไปแก้ไขไฟล์ชื่อ Services ของทั้งเครื่องเซิร์ฟเวอร์และเครื่องไคลเอนต์ ซึ่งอยู่ในไดเรกทอรี C:\WINNT\System32\driver\etc โดยให้เพิ่มข้อความดังนี้

turbo 1526/tcp #INFORMIX-Universal Server

ซึ่งคำว่า Turbo จะเป็นชื่อบริการของตัวเซิร์ฟเวอร์

6. การเซต Informix Setnet32

เมื่อเข้าไปในโปรแกรมจะปรากฏดังรูป



รูปที่ ก-1 Informix Setnet32

- ในส่วนของ Environment
 - ที่ช่อง INFORMIXSERVER = ซึ่งให้ใส่ชื่อของ Database Server ลงไป ไม่ใช่ชื่อเครื่องเซิร์ฟเวอร์
- ในส่วนของ Server Information
 - ในช่อง Host name ให้ใส่ชื่อของเครื่องเซิร์ฟเวอร์ลงไป
 - ในช่อง Protocolname ให้เลือกค่าonsoctcp
 - ในช่อง Service Name ให้ใส่ชื่อของ Service ที่ได้ตั้งไว้ เช่น turbo
- ในส่วนของ Host Information
 - ในช่อง User Name ให้ใส่ชื่อผู้ที่จะใช้ฐานข้อมูลลงไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

วิธีการสร้างโอเปกไทป์ลงใน Informix Universal Server

ในการสร้างชนิดข้อมูลใหม่เข้าไปในตัว Server นั้นมีเครื่องมือช่วยดังต่อไปนี้

1. เครื่องช่วย

1.1 BladeSmith

เราใช้โปรแกรม BladeSmith เพื่อเริ่มต้นการสร้าง DataBlade Module อย่างเช่น ใช้ระบุว่าจะภายใน DataBlade Module จะประกอบด้วยอะไรบ้าง และใช้ในการสร้างไฟล์ต่างๆ รวมทั้ง Source code ด้วย โปรแกรม BladeSmith จะแนะนำแนวทางเราเกี่ยวกับการกำหนดรายละเอียดให้กับ DataBlade Module ด้วย Wizard pages โปรแกรมสามารถที่จะทำงานได้หลายอย่างแบบอัตโนมัติในขั้นตอนการสร้างออบเจกต์ เช่น การเขียนคำสั่งของ SQL ที่จำเป็นพื้นฐานสำหรับใช้กำหนดออบเจกต์ใน ฐานข้อมูล

การใช้ BladeSmith นั้น ให้สร้าง Project ขึ้นมาหนึ่งอันแล้วทำการเพิ่มเอาส่วนต่างๆ ที่ต้องการเข้ามา ซึ่งสิ่งเหล่านั้น ได้แก่

- User-defined objects, aggregates, casts, errors, interfaces, routines และ data types
- Files ที่เป็นคำสั่งของ SQL หรือไฟล์ที่จำเป็นสำหรับ Client
- Imported objects เช่น built-in data types และ interfaces จาก DataBlade Module อื่น

หลังจากทำการกำหนด User-defined objects, imported object และไฟล์ต่างๆที่เราต้องการรวมไว้ใน DataBlade Module เสร็จแล้ว ใช้ BladeSmith ทำการสร้างไฟล์ต่างๆ ที่จะใช้ในการคอมไพล์ เป็น Shared object file หรือ Dynamic link library และไฟล์ต่างๆที่ใช้ในการจัดการกับ DataBlade Module ใน อินฟอร์มิก ยูนิเวอร์แซล เซิร์ฟเวอร์ ฟังก์ชันที่ใช้ในการทดสอบสิ่งที่เราได้เพิ่มเข้าไปได้แก่ฟังก์ชันชนิดข้อมูล เป็นต้น นอกจากนี้ยังมีไฟล์ที่ใช้สำหรับทำ Packaging file และส่วนต่างๆ ดังตารางนี้

ตารางที่ ข-1 ส่วนต่างๆ ที่ถูกสร้างภายใน Project

Type of Generated File	Description
Source code	เราจะใช้ไฟล์เหล่านี้เพื่อสร้าง Shared object หรือ Dynamic link library files ต่างๆเหล่านี้ประกอบด้วย Source code files, Header files และ Makefiles ที่เราคอมไพล์เป็น Shared object file หรือ ynamic link library
SQL script	ไฟล์เหล่านี้จะบรรจุชุดคำสั่งของภาษา SQL ที่สนับสนุน DataBlade Modules ในระบบตารางของฐานข้อมูลไฟล์เหล่านี้ประกอบด้วย 2 Scripts คือ Prepare script ที่ใช้อธิบายตัว DataBlade Module และ Object script ที่ใช้อธิบาย DataBlade objects

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Type of Generated File	Description
Test	ไฟล์เหล่านี้ใช้เพื่อตรวจสอบความถูกต้องของ User-defined routines, Opaque data type, Support routines และการ Casts
Packaging	เราจะใช้ไฟล์เหล่านี้ร่วมกับโปรแกรม BladePack เพื่อสร้าง Installation files

1.2 BladePack

เป็นโปรแกรมที่ใช้สำหรับสร้าง Package ที่ใช้สำหรับติดตั้ง DataBlade Module โปรแกรมนี้จะต้องการใช้ Packaging file ที่ถูกสร้างจากโปรแกรม BladeSmith สำหรับ Packaging file จะอ้างถึง SQL script shared object file และไฟล์อื่นๆที่จำเป็นต่อ DataBlade Module

เราสามารถใส่โปรแกรม BladePack ทำงานเหล่านี้ได้

- ทำการเพิ่มเอาไฟล์ต่างๆให้กับ DataBlade Module ยกตัวอย่างเช่น เราสามารถรวมเอา Documentation, on-line help และ ไฟล์ตัวอย่าง เข้าไปกับ DataBlade Module ได้
 - รวมเอาหลายๆ BladePack projects ไว้ใน Installation package ยกตัวอย่างเช่น เราอาจรวมเอาหลายๆ DataBlade Module ที่คล้ายๆกันเข้าไว้ใน Installation package อันเดียวได้
 - แบ่งไฟล์ออกเป็น Separate components, Sub components และ Shared components ยกตัวอย่างเช่นเราสามารถจะทำการตั้งชื่อแต่ละคอมโพเนนท์ย่อยต่างๆ แล้วให้คนที่ทำการติดตั้งสามารถที่จะเลือกได้ว่าต้องการติดตั้งคอมโพเนนท์ย่อยตัวใดบ้าง
 - รวมเอา Custom installation routines ไว้
- นอกจากที่กล่าวมาแล้วก็ยังสามารถทำงานอื่นๆได้อีก

1.3 BladeManager

โปรแกรมนี้จะใช้เพื่อทำการ Register และ Unregister ตัว DataBlade Module ในฐานข้อมูลและเพื่อติดตั้งและไม่ติดตั้ง DataBlade module client files

หลังจากได้ทำการติดตั้ง DataBlade Module ลงบนเซิร์ฟเวอร์แล้วเราจะต้องทำการรีจิสเตอร์ซึ่งเป็นขั้นตอนที่ทำให้เกิดการเพิ่ม DataBlade คู่ระบบฐานข้อมูล และ เกิดการสร้าง Shared object หรือ Dynamic link library ของ DataBlade ให้มีขึ้นในเซิร์ฟเวอร์

โปรแกรมจะทำการตรวจสอบความสัมพันธ์ระหว่าง DataBlade Module ถ้าในกรณีที่มีการใช้การติดต่อที่มาจาก DataBlade Module ตัวอื่น โปรแกรมนี้จะทำการรีจิสเตอร์ให้กับ DataBlade Module ของเราก่อนเมื่อมั่นใจแล้วว่าการติดต่อที่ DataBlade Module ต้องการนั้น ได้ถูกรีจิสเตอร์ให้กับฐานข้อมูลไว้ก่อนแล้ว

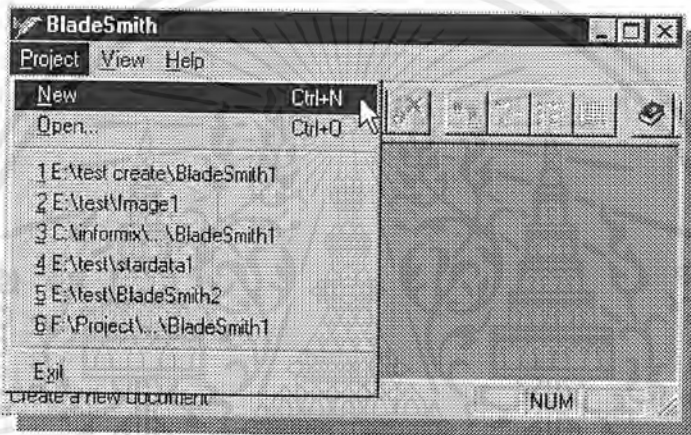
สำหรับในกรณีที่เราจะทำการเพิ่มเติมตัว DataBlade Module ของโปรแกรมนี้ก็จะทำการ Unregister ตัว DataBlade Module ตัวเดิมออกเองโดยอัตโนมัติ

เราสามารถใช้อุปกรณ์เพื่อทำการ Unregister ตัว DataBlade Module ได้ โดยที่โปรแกรมจะตรวจดูว่า DataBlade Module ที่เราต้องการจะ Unregister นั้นมีส่วนที่ DataBlade Module อื่นต้องการใช้งานหรือไม่ ถ้าหากว่ามีโปรแกรมจะไม่ทำการ Unregister เพราะจะเป็นการทำให้มีผลกระทบไปยัง DataBlade Module ตัวอื่นด้วย

2. การใช้งานเครื่องมือต่าง ๆ

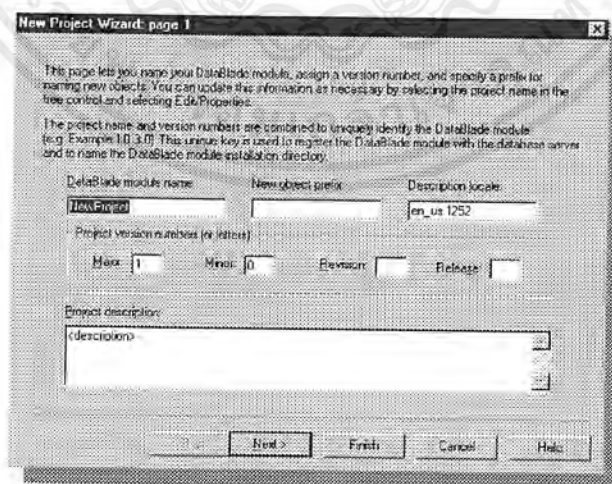
2.1 การใช้ Bladesmith

Bladesmith ใช้ในการสร้างชนิดของข้อมูลใหม่ที่ต้องการจะสร้าง เริ่มต้นด้วยการเรียกใช้งานโปรแกรม BladeSmith แล้วเลือกเมนู New ที่เป็นเมนูย่อยในเมนู Project ดังรูปที่ ข-1



รูปที่ ข-1 BladeSmith

เมื่อเลือก New แล้วโปรแกรมก็จะมีแบบฟอร์มมาให้กรอกดังรูปที่ ข-2

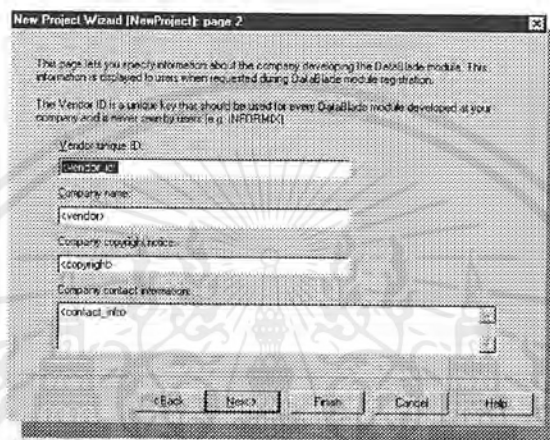


รูปที่ ข-2 New Project Wizard page 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

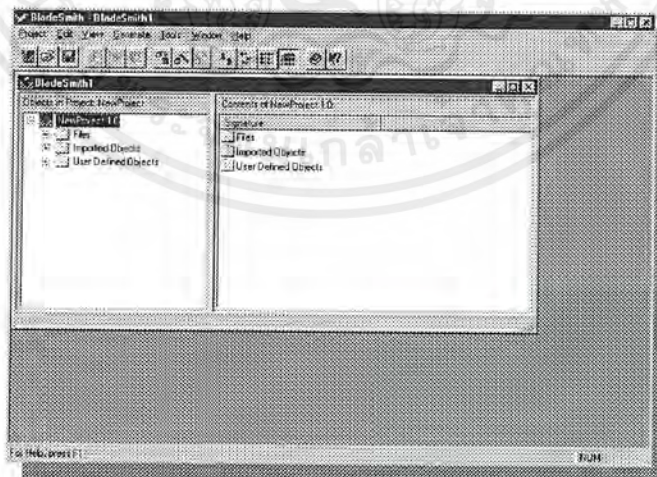
ช่อง DataBlade module name เป็นช่องที่มีไว้ให้ตั้งชื่อของ DataBlade Module ที่กำลังจะสร้าง ในกรณีที่ไม่ได้กรอกข้อมูลโปรแกรมก็จะมีค่าเริ่มต้นว่า NewProject ช่องที่ชื่อ New object prefix เป็นช่องที่โปรแกรมอนุญาตให้เราตั้งชื่อใดๆไว้แล้วในภายหลังถ้าเราสร้างสิ่งต่างๆ เช่นชนิดข้อมูลหรืออาจจะเป็น routine ให้กับ DataBlade Module ของเรา โปรแกรมก็จะกำหนดให้มันชื่อขึ้นต้นตรงกับที่เราตั้งไว้เอง ส่วนช่อง Project description เป็นช่องที่ใช้อธิบายถึงตัว DataBlade Module ที่เรากำลังสร้างอยู่

ถ้าหับช่อง New object prefix และช่อง Project description นั้น จะไม่กรอกก็ได้ เมื่อทำการกรอกข้อมูลเสร็จแล้วก็กดปุ่ม Next เพื่อทำงานต่อไป



รูปที่ ข-3 New Project Wizard page 2

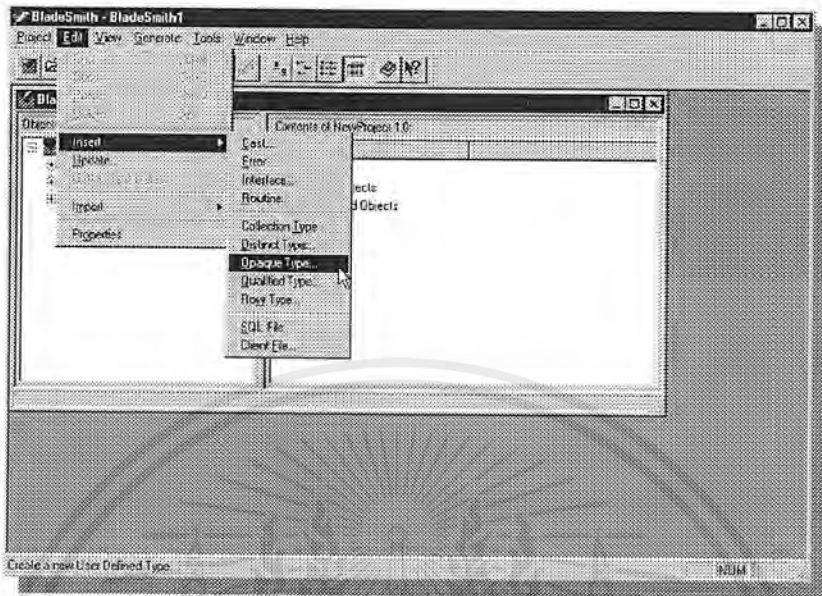
จากนั้น โปรแกรมก็จะให้กรอกข้อมูลเกี่ยวกับผู้สร้าง DataBlade Module ถ้าไม่ต้องการกรอกอะไรก็ได้ ให้กด Next ข้ามไปเลย แล้วโปรแกรมก็จะให้เราดู Script ภาษา SQL แล้วเราก็กดปุ่ม Finish ได้เลยแล้วก็จะเห็นตัว Project ใหม่ที่เราสร้างขึ้นดังรูปที่ ข-4



รูปที่ ข-4 เมื่อสิ้นสุด Wizard page

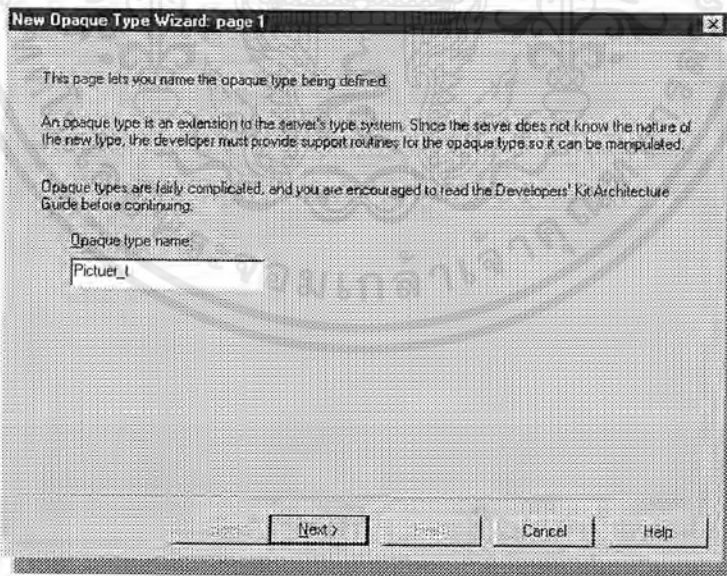
ที่นี่เราก็สามารถที่จะทำการสร้าง Objects ต่างๆที่เราต้องการได้ เช่นยกตัวอย่างว่าถ้าหากเราต้องการจะสร้าง Object ที่เป็น Opaque type ก็ทำได้โดยกดที่เมนู Edit แล้วก็ไปเลือกเมนูย่อย Insert แล้วจะเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เห็นสิ่งต่างๆที่เราสามารถจะสร้างได้ ถ้าต้องการสร้าง Opaque type ให้เลือกที่ Opaque type ดังตัวอย่างในรูปที่ ข-5



รูปที่ ข-5 การ Insert Opaque Data Type

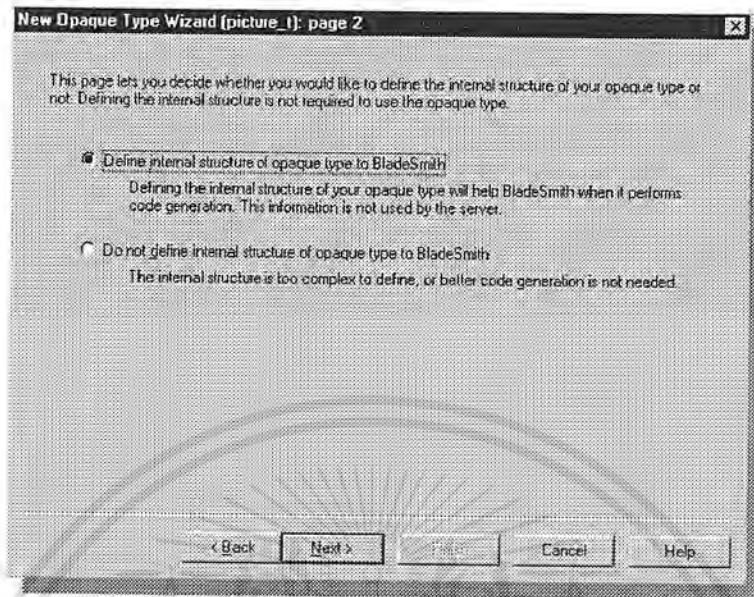
เมื่อเลือกแล้ว โปรแกรมจะถามเพื่อให้ทำการตั้งชื่อของ Opaque type ที่ต้องการ จะสร้าง เมื่อกรอกชื่อแล้วก็กดปุ่ม Next เพื่อทำงานต่อ



รูปที่ ข-6 New Opaque Type Wizard page 1

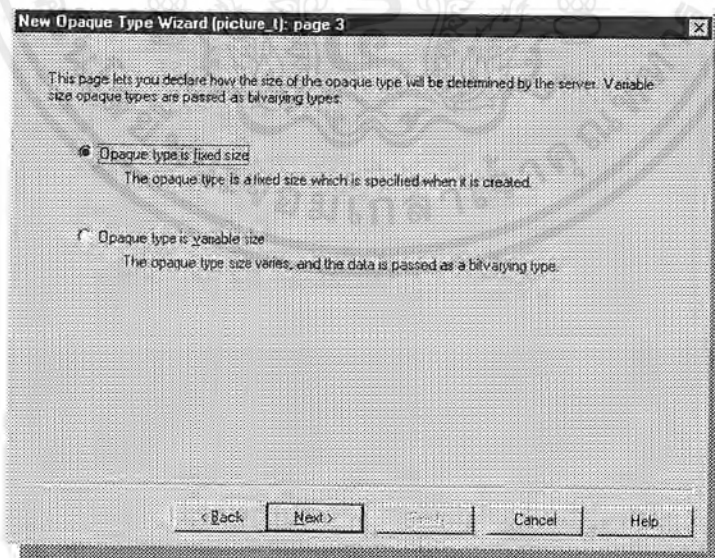
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กดปุ่ม Next จะปรากฏดังรูปที่ ข-7



รูปที่ ข-7 New Opaque Type Wizard page 2

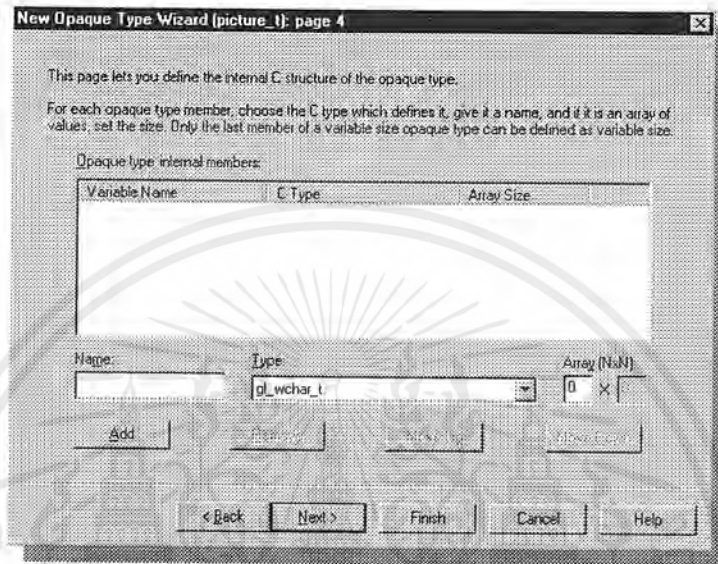
โปรแกรมจะให้เราเลือกว่าต้องการที่จะกำหนด Internal structure โดยใช้โปรแกรมนี้หรือไม่ ถ้าหากต้องการก็เลือกด้านบน แต่หาก Internal structure ที่ต้องการ จะกำหนดนั้นมีความซับซ้อนเกินกว่าจะใช้โปรแกรมนี้กำหนดได้ ก็สามารถเลือกด้านล่าง และเราก็จะต้องไปทำการกำหนด Internal structure เอง ภายหลัง สำหรับตัวอย่างนี้จะใช้ โปรแกรม BladeSmith เป็นตัวกำหนดเลย โดยการกด Next เพื่อทำต่อไป



รูปที่ ข-8 New Opaque Type Wizard page 3

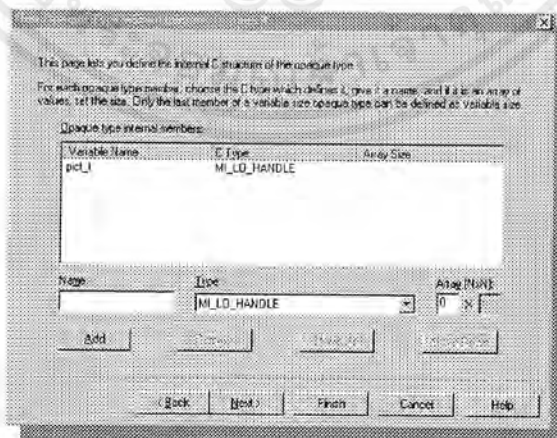
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจาก Opaque type มีแบบที่มีขนาดแน่นอนกับแบบที่มีขนาดไม่แน่นอน จากกรอบด้านบน โปรแกรมจะให้เราเลือกว่าเราต้องการที่จะสร้าง Opaque type ประเภทใด ถ้าเลือกแบบบนก็จะเป็น Opaque type แบบที่มีขนาดแน่นอน แต่แบบล่างจะเป็น Opaque type แบบที่มีขนาดไม่แน่นอน เมื่อเลือกแบบที่ต้องการแล้วก็ให้กดปุ่ม Next ต่อไป แล้วก็จะเข้าสู่กรอบที่ใช้ในการกำหนด Internal structure ดังรูปที่ ข-9



รูปที่ ข-9 New Opaque Type Wizard page 4

จากรูปที่ ข-9 เวลาที่จะเพิ่ม Field ใหม่ให้กับ Opaque type ทำได้โดยการกดปุ่ม Add แต่ก่อนหน้าที่จะ Add ก็ต้องทำการตั้งชื่อของ Field นั้นก่อนตรงช่อง Name แล้วเลือกชนิดของข้อมูลสำหรับ Field นั้น โดยกดลูกศรที่ตรง Type แล้วจะมีรายการให้เลือก สำหรับช่องที่เขียนว่า Array นั้นใช้สำหรับกำหนดชนิดข้อมูลให้เป็นแบบ Array

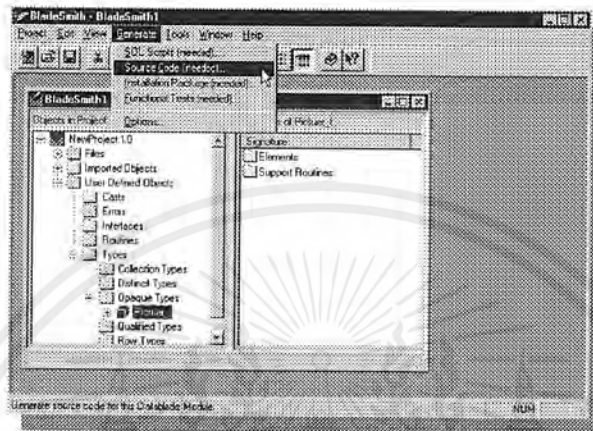


รูปที่ ข-10 การระบุชนิดข้อมูลใน Opaque datatype

จากรูปที่ ข-10 เป็นตัวอย่างการกำหนด Internal structure ให้มี 1 Field คือ pict_t มีชนิดข้อมูลเป็น MI_LO_HANDLE สำหรับ Field ที่ Add ไปแล้ว แล้วต้องการจะยกเลิก Field นั้น ทำได้โดยการเลือกชื่อเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Field นั้นแล้วกดปุ่ม Remove หรือถ้าต้องการเลื่อนขึ้นก็กดปุ่ม Move Up ถ้าต้องการเลื่อนลงกดปุ่ม Move Down เมื่อเพิ่มเติมแก้ไขจนพอใจแล้วก็กดปุ่ม Finish แล้วก็จะได้ Opaque type ใหม่ขึ้น

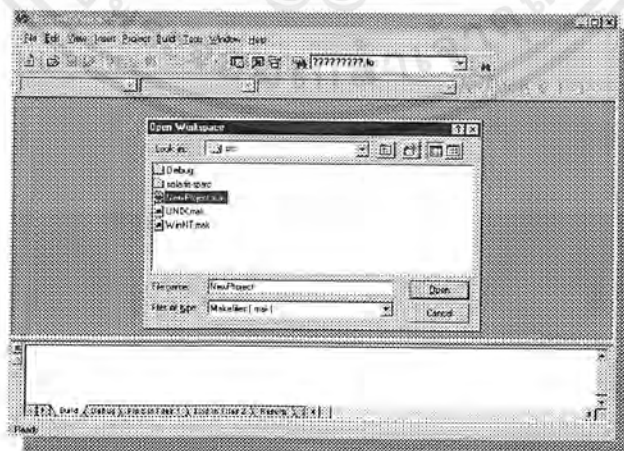
ต่อมาเราจะต้องมาสร้าง File ที่จำเป็นในการนำ Opaque ที่ออกแบบไว้ไป Register โดยให้โปรแกรมสร้าง Source files ต่างๆ โปรแกรมจะสร้างเป็น Source code ภาษา C ให้ สำหรับวิธีการที่จะสร้างไฟล์ต่างๆนั้นทำได้โดยเลือกที่เมนู Generate แล้วจะเห็นดังรูปที่ ข-11



รูปที่ ข-11 การสร้าง Source code

ก็ให้เลือกที่ละหัวข้องานครบ ในตอนที่เลือกอันแรกโปรแกรมจะถามให้ทำการบันทึกข้อมูลเสียก่อน ก็ทำการบันทึกแล้วก็จะได้ไฟล์ต่างๆที่โปรแกรมสร้างให้ก็จะอยู่ใน Subdirectory ที่ชื่อ Scripts, Src, Install และ Functest ซึ่งจะอยู่ใน Directory ที่เราทำการบันทึก Project ไว้อีกที

หลังจากนี้เราก็ให้นำเอา Source code ภาษา C++ ไปทำการ Compile โดยใช้โปรแกรม Visual C++ เวอร์ชัน 6 โดยทำการ Compile ให้เป็น Library file ซึ่งมีวิธีการโดยการ Open Workspace ด้วย file นามสกุล mak ที่อยู่ใน Directory Scr ที่ได้จากการ Generated ของ Bladesmith ดังรูปที่ ข-12

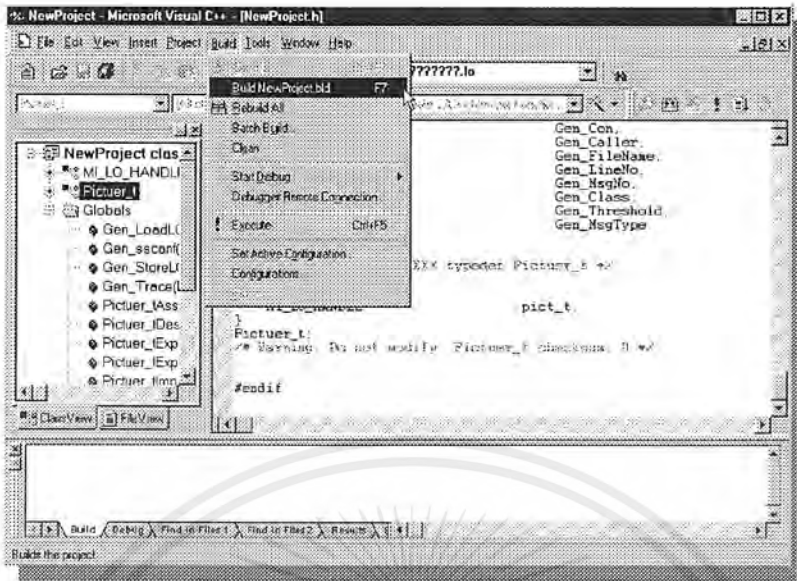


รูปที่ ข-12 การ Open Workspace

หลังจากนั้นใช้คำสั่ง Build เพื่อสร้าง file นามสกุล bld เพื่อนำไป Copy ลงใน Directory ที่ได้จาก

การสร้างของ Bladepacking อีกที่ดังรูปที่ ข-13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

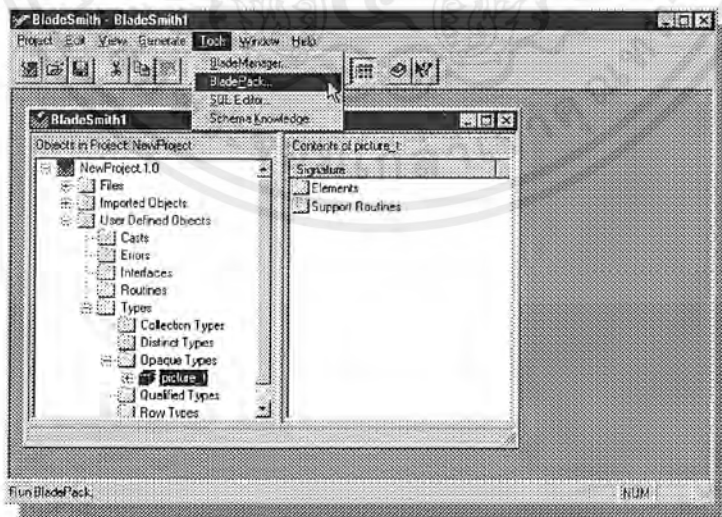


รูปที่ ข-13 การ Compile เป็น file นามสกุล bld

ทำการ Registry ชนิดข้อมูลโอเพนไทป์ที่ได้มาจาก DBDK BladePack ลงในคีย์ระบบโดยใช้ DBDK Bladesmanager ซึ่งมีวิธีใช้ดังนี้

2.2 การใช้ BladePack

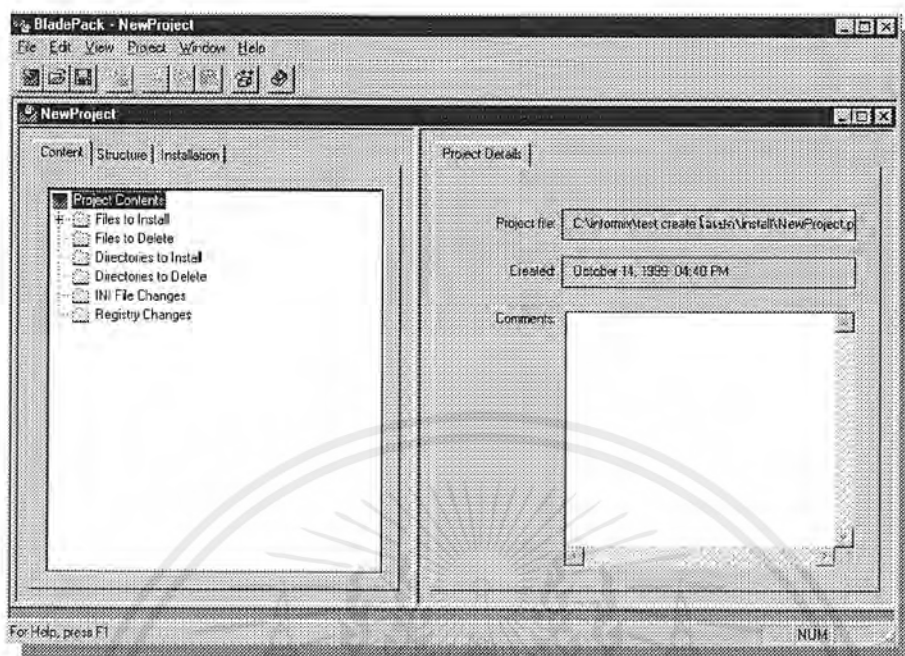
หลังจากสามารถสร้าง DataBlade Module ได้แล้ว ต่อไปก็จะสร้าง Package installation โดยใช้โปรแกรม BladePack ที่สามารถเรียกใช้ได้โดยเลือกที่เมนู tools ที่อยู่ในโปรแกรม BladeSmith เมื่อเปิดโปรแกรมเสร็จแล้วจะได้ project ใหม่ดังนี้



รูปที่ ข-14 การเรียกใช้งาน BladPack

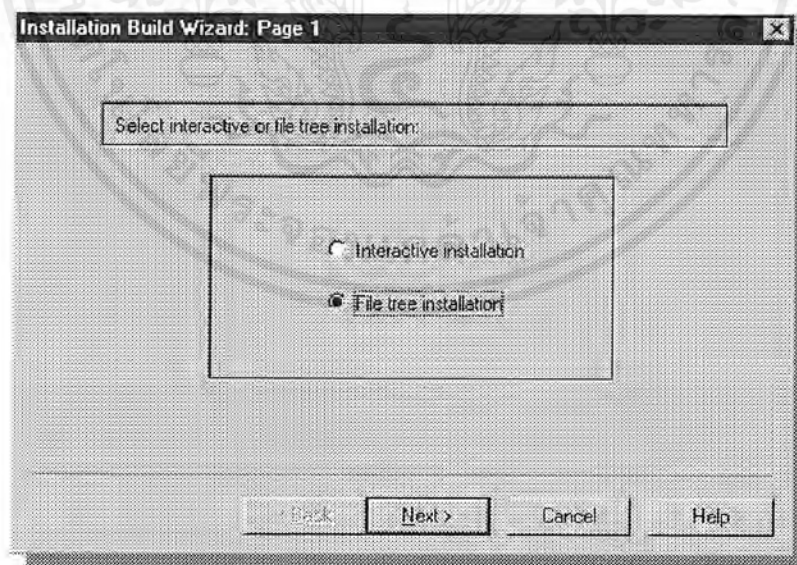
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ Click จะปรากฏดังรูปที่ ข-15



รูปที่ ข-15 หน้าต่าง *BladPack*

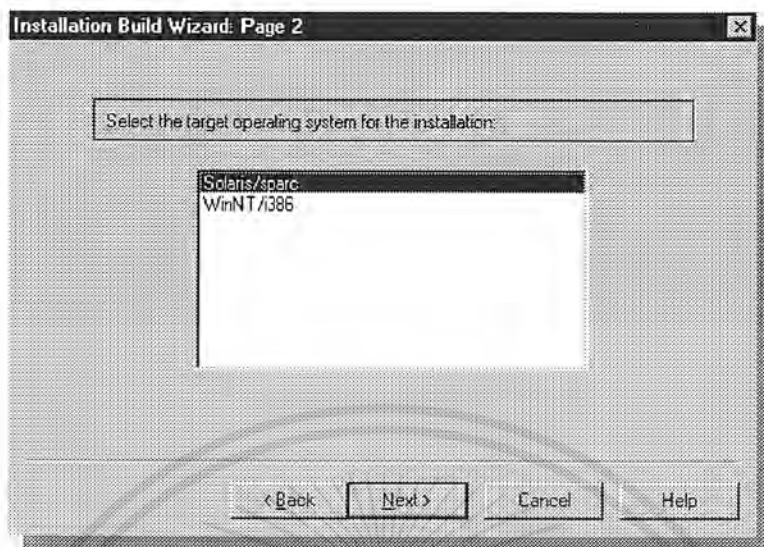
ต่อไปก็เลือกเมนู Project แล้วเลือก Build installation แล้วโปรแกรมก็จะถามเพื่อให้เราบันทึก Project ที่เราสร้างใหม่ใน BladePack ก็ทำการ Save ไปแล้วโปรแกรมจะให้เลือกดังรูปที่ ข-16



รูปที่ ข-16 *Installation Build Wizard page 1*

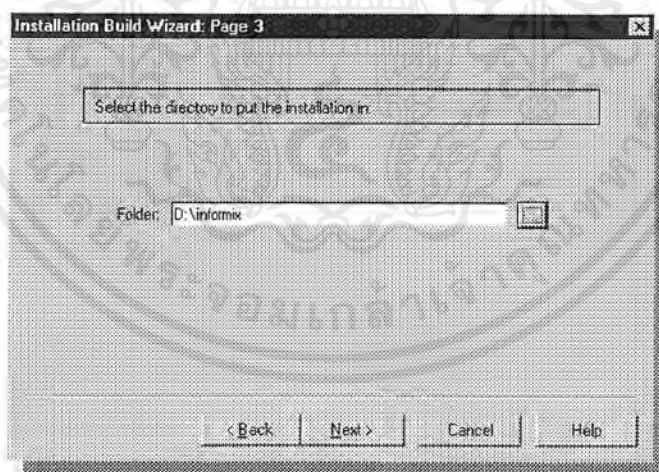
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้เลือก File tree installation แล้วกด Next แล้วโปรแกรมก็จะให้เลือกต่อค่า ดังรูปที่ ข-17



รูปที่ ข-17 Installation Build Wizard page 2

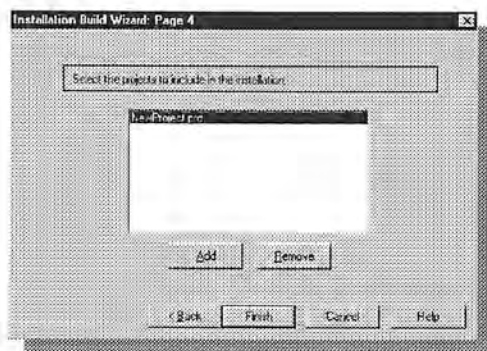
ก็ให้เลือก WinNT/i386 แล้วกด Next ต่อ แล้วโปรแกรมก็จะให้ป้อนชื่อของ Folder ที่ต้องการจะให้ตัวติดตั้ง. ให้ป้อนที่อยู่ที่ตั้ง Informix Universal Server (เมื่อ Bladepack ทำการสร้างตัวติดตั้งเสร็จมันจะเอาไปได้ที่ Directory :/informix/extend) ดังรูปที่ ข-18 เมื่อป้อนเสร็จก็กดปุ่ม Next



รูปที่ ข-18 Installation Build Wizard page 3

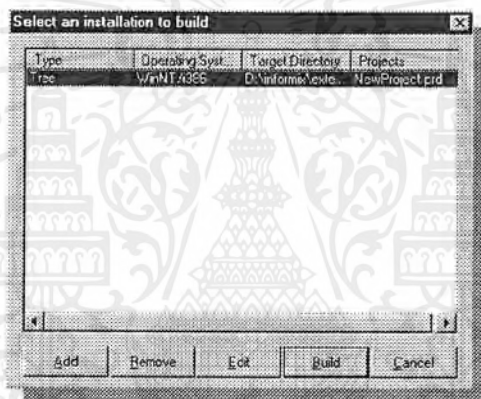
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้กดปุ่ม Next จะปรากฏดังรูปที่ ข-19



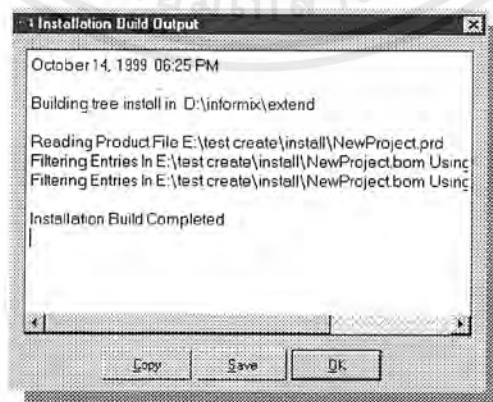
รูปที่ ข-19 Installation Build Wizard page 4

คลิก Finish จะปรากฏดังรูปที่ ข-20



รูปที่ ข-20 Select an installation to build

กดปุ่ม Build จะปรากฏดังรูปที่ ข-21



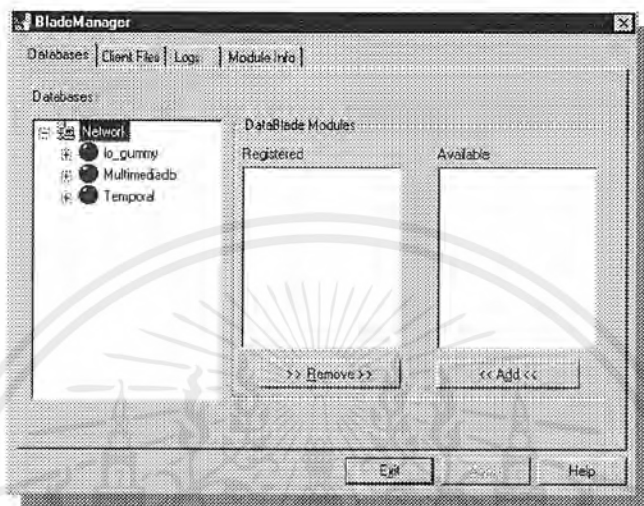
รูปที่ ข-21 Installation Build Output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก็เป็นอันเสร็จสิ้นการสร้างตัวติดตั้ง DataBlade Module ต่อไปก็ออกจากโปรแกรม BladePack ได้

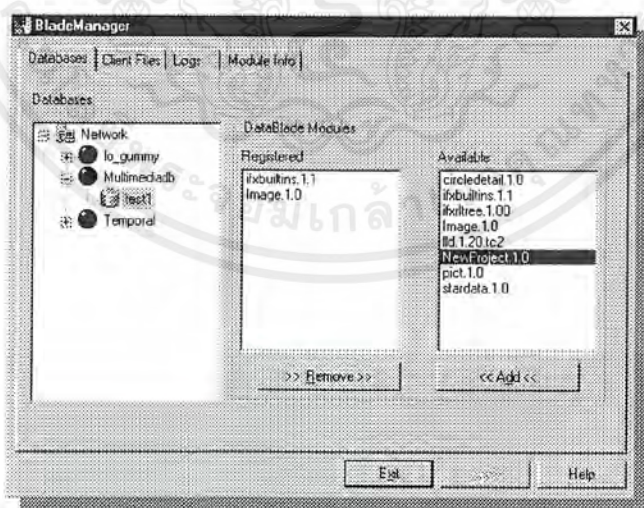
2.3 BladManager

การริจิสเตอร์ตัว DataBlade Module ให้กับ database ที่ต้องการ สามารถทำได้โดยใช้โปรแกรม BladeManager เมื่อทำการเรียกใช้งานโปรแกรมแล้ว จะมีหน้าจอเป็นดังรูปที่ ข-22



รูปที่ ข-22 การใช้ BladeManager

ให้ทำการเลือก Database server ที่มี database ที่ต้องการจะทำการริจิสเตอร์เช่นถ้าเลือก Multimedia โปรแกรมก็จะแสดงชื่อของ Database ที่มีอยู่ใน Server ที่ชื่อ Multimedia ดังรูปที่ ข-23



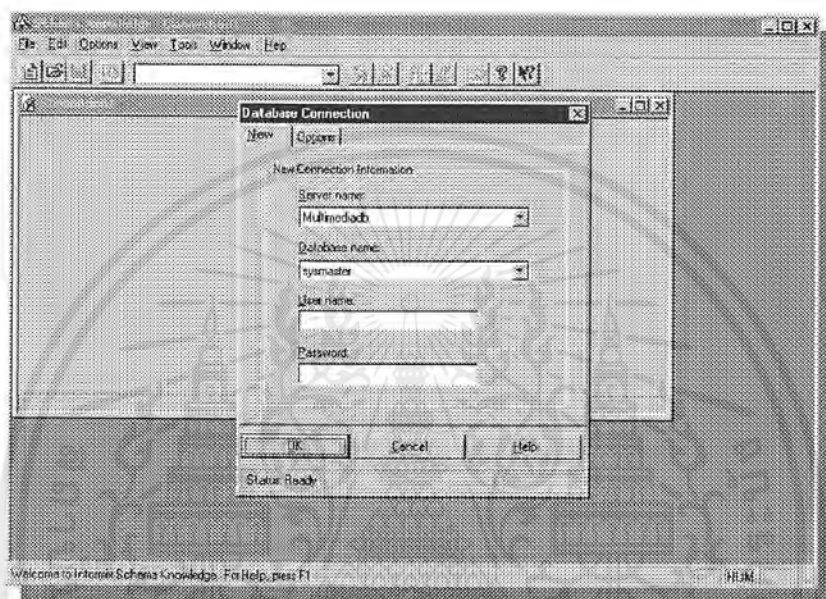
รูปที่ ข-23 BladeManger เมื่อเราเลือกฐานข้อมูลแล้ว

เมื่อเราได้เลือก Database ที่ต้องการจะ Register แล้ว ก็จะมีข้อมูลมาในกรอบสองกรอบด้านขวาคือกรอบ Registered และกรอบ Available สำหรับกรอบแรกนั้นคือกรอบที่แสดงรายชื่อของ DataBlade Module ที่ได้ทำการริจิสเตอร์ไว้ใน Database นี้แล้ว ส่วนอีกกรอบจะเป็นรายชื่อของ DataBlade Module ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มืออยู่ใน database server ที่สามารถทำการรีจิสเตอร์ได้ เมื่อต้องการรีจิสเตอร์ก็ทำได้โดยการเลือก DataBlade Module ที่ต้องการจากกรอบ Available แล้วก็กดปุ่ม Add ที่อยู่ด้านล่างของกรอบ และเมื่อต้องการจะทำการ Unregister ก็ทำได้โดยการเลือก DataBlade Module ที่ได้รับรีจิสเตอร์ไว้แล้วจากกรอบ Registered แล้วกดปุ่ม Remove หลังทำการรีจิสเตอร์และ Unregister เรียบร้อยแล้ว ก็กดปุ่ม Apply แล้วก็กดปุ่ม Exit ก็เป็นการเสร็จสิ้นขั้นตอนการทำ Register และ Unregister

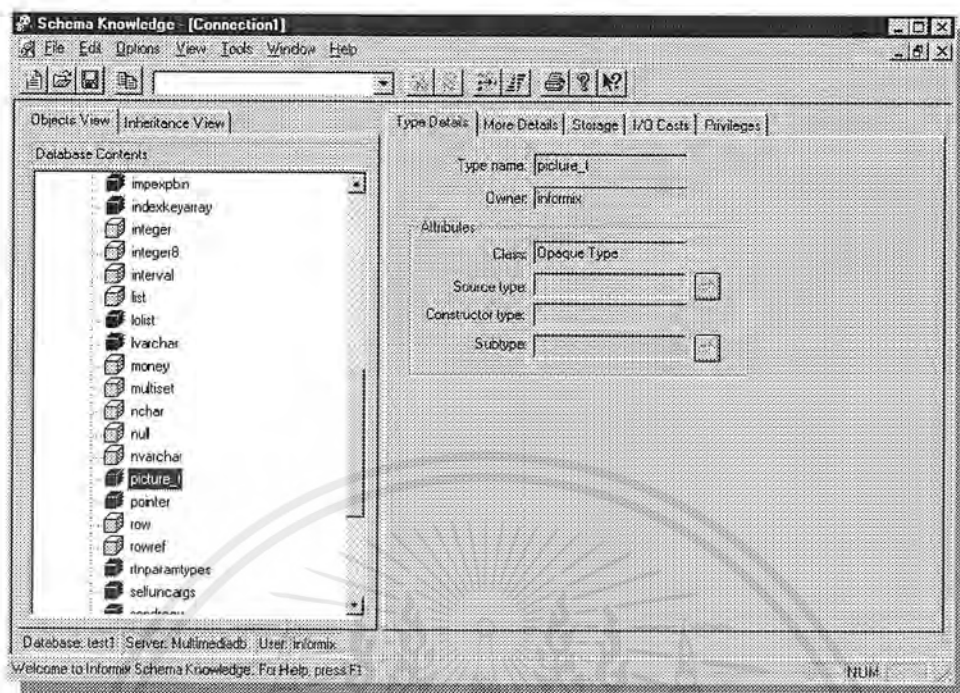
เราสามารถดูชนิดข้อมูลที่เรารสร้างได้โดยใช้โปรแกรม Schema Knowledge โดยเราจะต้องสร้าง Connection ขึ้นมาดังรูปที่ ข-24



รูปที่ ข-24 การ login ใน โปรแกรม Schema Knowledge

ทำการ Login user name และ จะปรากฏดังรูปเลือกที่ Types จะปรากฏชนิดข้อมูลที่เรทำได้ทำการ กำหนดขึ้นมาใหม่ดังรูปที่ ข-25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-25 โปรแกรม Schema Knowledge เมื่อเลือกชนิดข้อมูลที่มีในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

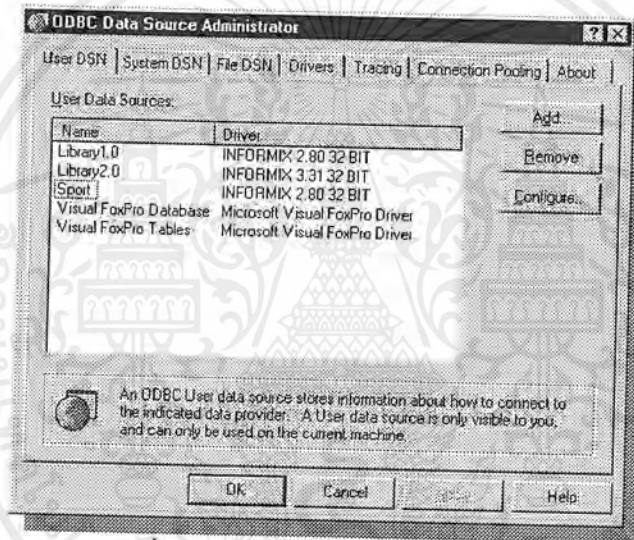
ภาคผนวก ค

การสร้างการติดต่อระหว่าง Visual Basic กับฐานข้อมูลโดยผ่าน Data Director

การสร้างการติดต่อระหว่าง Visual Basic กับฐานข้อมูลโดยผ่าน Data Director นั้นมีขั้นตอนดังนี้

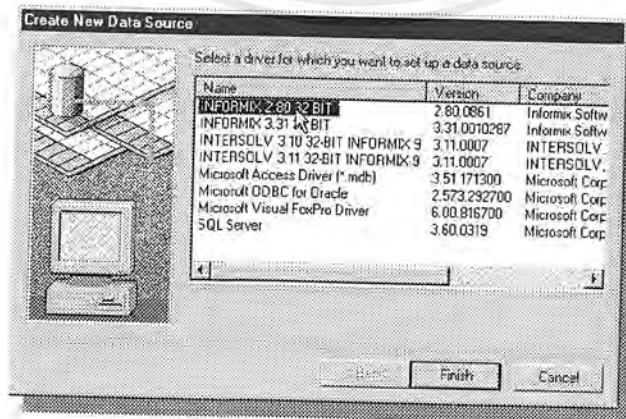
1. สร้าง Data Source ใน ODBC Driver มีรายละเอียดดังนี้

1.1 ไปที่ StartMenu -> Setting -> Control Panel เลือก ODBC จะปรากฏดังรูปที่ ค-1



รูปที่ ค-1 ODBC Data Source Administrator

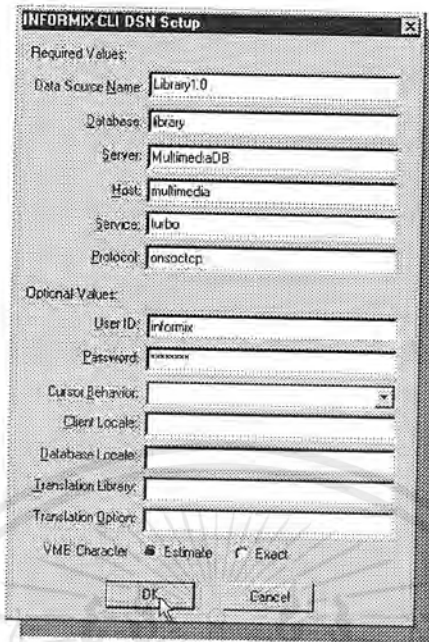
1.2 สร้าง Data Source โดยการ คลิก Add... จะได้ดังรูปที่ ค-2



รูปที่ ค-2 Create New Data Source

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 คลิกเลือก Informix Driver ที่ต้องการ แล้วคลิก Finish จะได้ดังรูปที่ ค-3

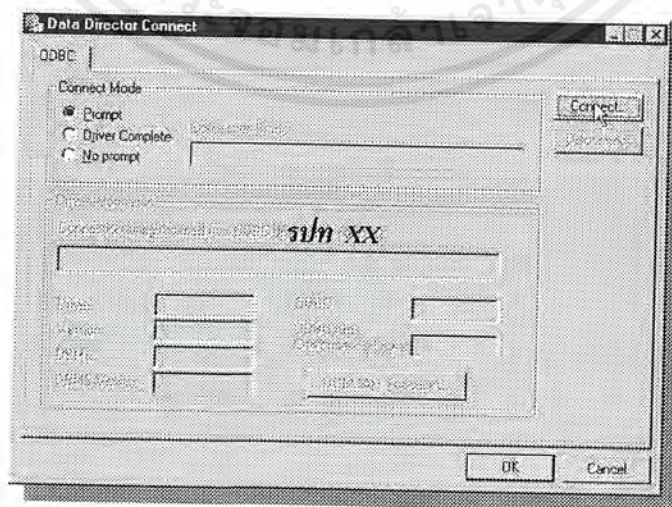


รูปที่ ค-3 Informix-CLI DSN setup

1.4 ป้อนชื่อ Data Source , Database, Server,Host,Server,Host,Service,Protocol,User ID และ Password

2. ทดสอบการเชื่อมต่อระหว่าง Data Director กับ ฐานข้อมูลโดยใช้ Data Director Connect (อยู่ในชุด Software ของ Data Director)

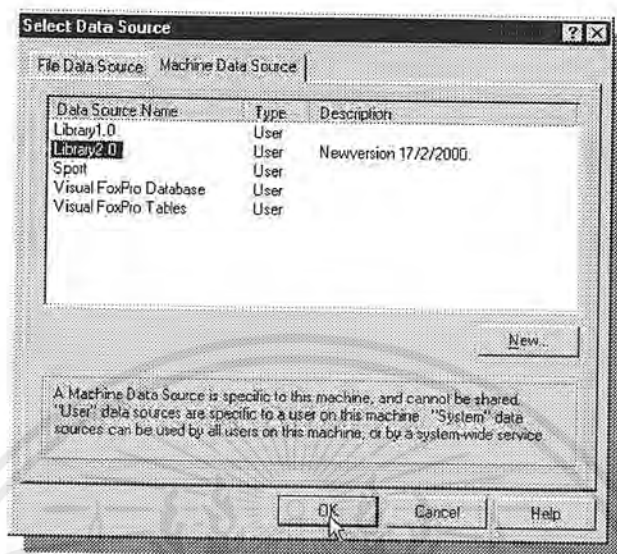
2.1 ไปที่ StartMenu -> Program -> Data Director 3.5 -> Data Director Connect จะปรากฏดังรูปที่ ค-4



รูปที่ ค-4 Data Director Connect

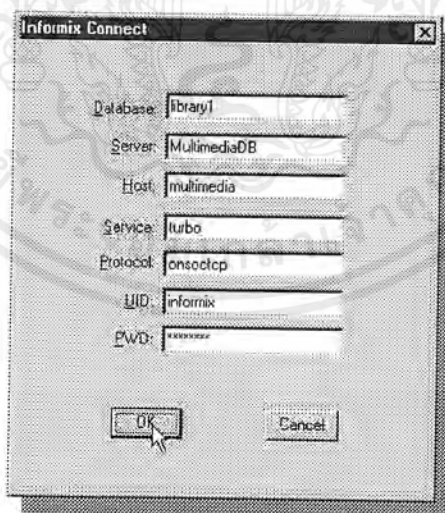
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 คลิก Connect เพื่อลองติดต่อกับฐานข้อมูลจะได้ดังรูปที่ ก-5



รูปที่ ก-5 Select Data Source

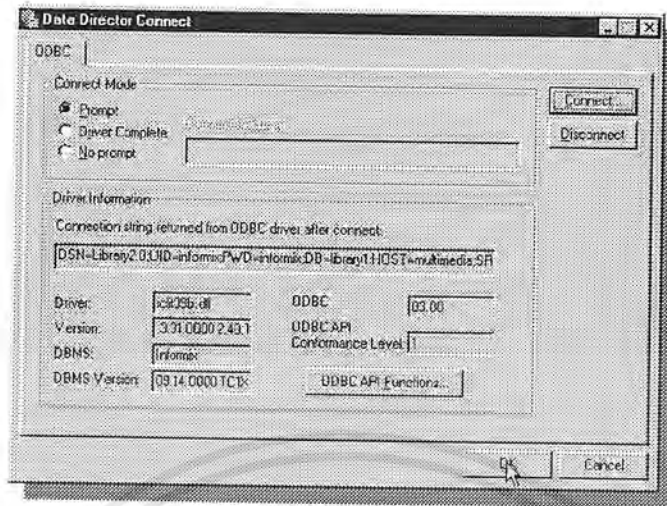
2.3 คลิกเลือก Data Source ที่ต้องการติดต่อ ในที่นี้เลือก Library2.0 แล้วคลิก OK จะได้ดังรูปที่ ก-6 แล้วคลิก OK เพื่อยืนยันค่าต่างๆ ที่ป้อน



รูปที่ ก-6 Informix Connect

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

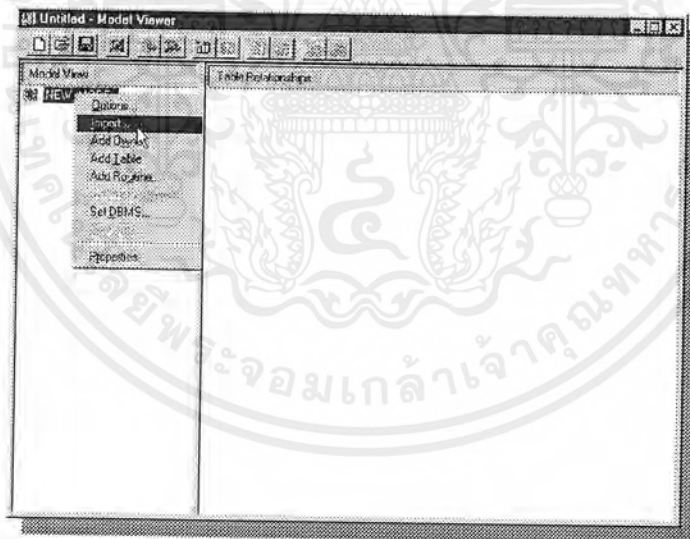
2.4 ถ้าการติดต่อสำเร็จจะปรากฏดังรูปที่ ค-7



รูปที่ ค-7 Data Director Connect

3. สร้าง Model ของฐานข้อมูล โดยใช้ Data Director

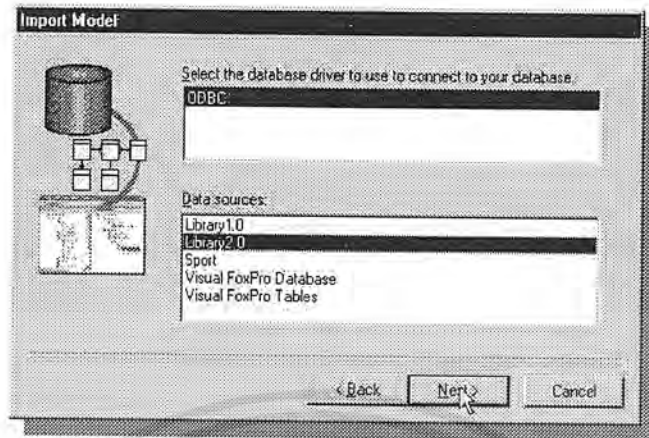
3.1 เปิดโปรแกรม Visual Basic -> View -> Model Viewer แล้วคลิกเมาส์ขวาจะได้ดังรูปที่ ค-8



รูปที่ ค-8 Model Viewer

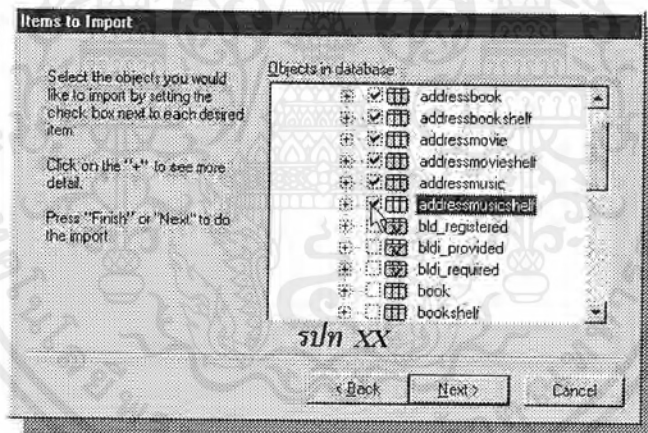
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 เลือก Import เพื่อดึงเอาตารางในฐานข้อมูลเข้ามาใน Visual Basic จะปรากฏดังรูปที่ ก-9



รูปที่ ก-9 Import Model

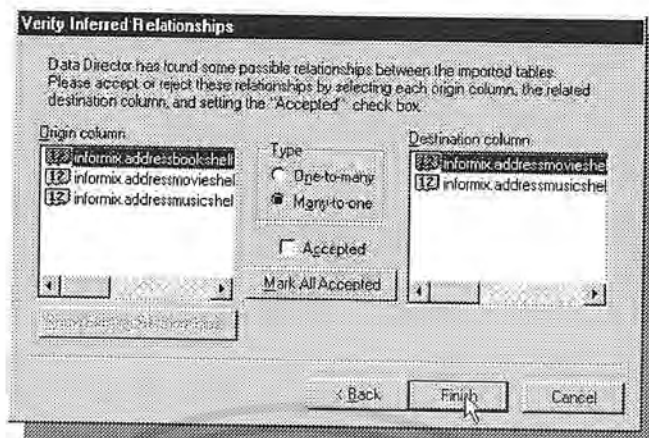
3.3 เลือก Data Source ที่ต้องการจะใช้ แล้วคลิก Next จะได้ดังรูปที่ ก-10



รูปที่ ก-10 Items to Import

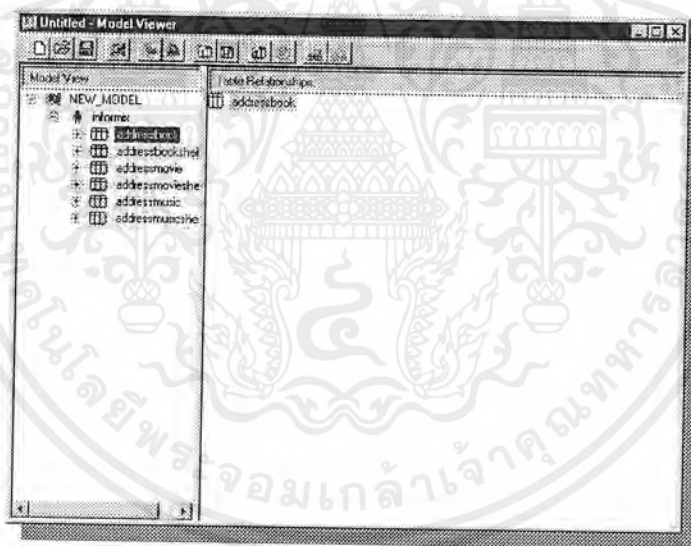
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3.4 เช็คล็อกหน้าตารางที่ต้องการจะติดต่อกับฐานข้อมูล เมื่อเลือกแล้วคลิก Next จะได้ดังรูปที่ ค-11 แล้วคลิกเลือกความสัมพันธ์ระหว่างคอลัมน์



รูปที่ ค-11 Verify Inferred Relationships

- 3.5 เมื่อกำหนดความสัมพันธ์ระหว่างคอลัมน์ของแต่ละตารางแล้วคลิก Finish จะปรากฏดังรูปที่ ค-12



รูปที่ ค-12 Model Viewer

- 3.6 ขณะนี้เราได้นำรูปแบบของตารางต่างๆ ในฐานข้อมูลเข้ามาใน Visual Basic แล้ว ในการติดต่อกับฐานข้อมูลเราจะติดต่อผ่าน Model นี้โดยที่เราไม่ต้องติดต่อกับฐานข้อมูลจริง Model จะติดต่อกับฐานข้อมูลให้เราเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

Source Code

1. Stored Procedure Routine

```

CREATE PROCEDURE UPDATEMUSIC(Input_MusicID integer, Input_MusicCode varchar(15),
Input_MusicName varchar(30), Input_Arthist varchar(25), Input_Album varchar(20), Input_Publishing
varchar(20), Input_year integer, Input_Song varchar(100), Input_Genre varchar(10), Input_Shelfid
integer, Input_Layer integer, Input_Part char(1))
DEFINE Result varchar(15);
LET Result = ( SELECT MUSICCODE FROM MUSIC WHERE MUSICCODE = Input_MusicCode );
IF ( Result IS NULL ) THEN
    IF Input_Song != "NOT" THEN
        UPDATE MUSIC
        SET musiccode = Input_MusicCode, musicname = Input_MusicName, arthist =
Input_Arthist, genre = Input_Genre, album = Input_Album, publishing = Input_Publishing, year =
Input_Year, Song = Input_Song
        WHERE musicid = Input_MusicID;
    ELSE UPDATE MUSIC
        SET musiccode = Input_MusicCode, musicname = Input_MusicName, arthist =
Input_Arthist, genre = Input_Genre, album = Input_Album, publishing = Input_Publishing, year =
Input_Year
        WHERE musicid = Input_MusicID;
    END IF;
ELSE
    IF Input_Song != "NOT" THEN
        UPDATE MUSIC
        SET musicname = Input_MusicName, arthist = Input_Arthist, genre = Input_Genre, album =
Input_Album, publishing = Input_Publishing, year = Input_Year, Song = Input_Song
        WHERE musicid = Input_MusicID;
    ELSE UPDATE MUSIC

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        SET musicname = Input_MusicName, arthist = Input_Arthist, genre = Input_Genre, album =
Input_Album, publishing = Input_Publishing, year = Input_Year

```

```

        WHERE musicid = Input_MusicID;

```

```

    END IF;

```

```

END IF;

```

```

UPDATE ADDRESSMUSIC

```

```

SET shelfid = Input_shelfid, layer = Input_Layer, part = Input_Part

```

```

WHERE musicid = Input_MusicID;

```

```

END PROCEDURE;

```

```

:-----:

```

```

CREATE PROCEDURE DELETESHelf_ROOM(Input_shelf_roomid integer, Input_Type integer)

```

```

    IF Input_Type = 0 THEN

```

```

        DELETE FROM bookshelf WHERE shelfid = Input_shelf_roomid;

```

```

    ELIF Input_Type = 1 THEN

```

```

        DELETE FROM musicshelf WHERE shelfid = Input_shelf_roomid;

```

```

    ELIF Input_Type = 2 THEN

```

```

        DELETE FROM movieshelf WHERE shelfid = Input_shelf_roomid;

```

```

    ELSE

```

```

        DELETE FROM room WHERE roomid = Input_shelf_roomid;

```

```

    END IF;

```

```

END PROCEDURE;

```

```

:-----:

```

```

CREATE PROCEDURE UPDATEBOOK(Input_BookID integer, Input_BookCode varchar(15),
Input_BookName varchar(40), Input_Author1 varchar(20), Input_Author2 varchar(20), Input_Author3
varchar(20), Input_ISBN varchar(13), Input_Publishing varchar(20), Input_year integer, Input_Title
varchar(100), Input_Preface varchar(100), Input_BookClass varchar(15), Input_Shelfid integer,
Input_Layer integer, Input_Part char(1))

```

```

    DEFINE Result varchar(15);

```

```

    LET Result = ( SELECT BOOKCODE FROM BOOK WHERE BOOKCODE = Input_BookCode );

```

```

    IF ( Result IS NULL ) THEN

```

```

        IF (Input_Title != "NOT") and (Input_Preface != "NOT") THEN

```

```

            UPDATE BOOK

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SET bookcode = Input_BookCode, bookname = Input_BookName, bookclass =
Input_BookClass, isbn = Input_ISBN, publishing = Input_Publishing, year = Input_Year, Image =
Input_Title, Preface = Input_Preface, author = row ( Input_Author1, Input_Author2, Input_Author3 ) ::
Author_t

WHERE bookid = Input_BookID;

ELIF (Input_Title != "NOT") and (Input_Preface = "NOT") THEN

UPDATE BOOK

SET bookcode = Input_BookCode, bookname = Input_BookName, bookclass =
Input_BookClass, isbn = Input_ISBN, publishing = Input_Publishing, year = Input_Year, Image =
Input_Title, author = row ( Input_Author1, Input_Author2, Input_Author3)::Author_t

WHERE bookid = Input_BookID;

ELIF (Input_Title = "NOT") and (Input_Preface != "NOT") THEN

UPDATE BOOK

SET bookcode = Input_BookCode, bookname = Input_BookName, bookclass =
Input_BookClass, isbn = Input_ISBN, publishing = Input_Publishing, year = Input_Year, Preface =
Input_Preface, author = row(Input_Author1,Input_Author2,Input_Author3)::Author_t

WHERE bookid = Input_BookID;

ELSE UPDATE BOOK

SET bookcode = Input_BookCode, bookname = Input_BookName, bookclass =
Input_BookClass, isbn = Input_ISBN, publishing = Input_Publishing, year = Input_Year, author = row(
Input_Author1, Input_Author2, Input_Author3)::Author_t

WHERE bookid = Input_BookID;

END IF;

ELSE

IF (Input_Title != "NOT") and (Input_Preface != "NOT") THEN

UPDATE BOOK

SET bookcode = Input_BookCode, bookname = Input_BookName, bookclass =
Input_BookClass, isbn = Input_ISBN, publishing = Input_Publishing, year = Input_Year, Image =
Input_Title, Preface = Input_Preface, author = row (Input_Author1, Input_Author2, Input_Author3) ::
Author_t

WHERE bookid = Input_BookID;

ELIF (Input_Title != "NOT") and (Input_Preface = "NOT") THEN

UPDATE BOOK

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SET bookcode = Input_BookCode, bookname = Input_BookName, bookclass =
Input_BookClass, isbn = Input_ISBN, publishing = Input_Publishing, year = Input_Year, Image =
Input_Title, author = row(Input_Author1, Input_Author2, Input_Author3)::Author_t
WHERE bookid = Input_BookID;
ELIF (Input_Title = "NOT") and (Input_Preface != "NOT") THEN
UPDATE BOOK
SET bookcode = Input_BookCode, bookname = Input_BookName, bookclass =
Input_BookClass, isbn = Input_ISBN, publishing = Input_Publishing, year = Input_Year, Preface =
Input_Preface, author = row(Input_Author1, Input_Author2, Input_Author3)::Author_t
WHERE bookid = Input_BookID;
ELSE UPDATE BOOK
SET bookcode = Input_BookCode, bookname = Input_BookName, bookclass =
Input_BookClass, isbn = Input_ISBN, publishing = Input_Publishing, year = Input_Year, author = row(
Input_Author1, Input_Author2, Input_Author3)::Author_t
WHERE bookid = Input_BookID;
END IF;
END IF;
UPDATE ADDRESSBOOK
SET shelfid = Input_shelfid, layer = Input_Layer, part = Input_Part
WHERE bookid = Input_BookID;
END PROCEDURE;
:-----:

```

```

CREATE FUNCTION searchdetailmusic(Input_MusicID integer, Input_MusicCode varchar(15),
Input_MusicName varchar(30), Input_Arthist varchar(25), Input_Album varchar(20), Input_Publishing
varchar(20), Input_year integer, Input_Genre varchar(10)) returning integer, varchar(15), varchar(30),
varchar(25), varchar(20), varchar(20), integer, varchar(10), integer, integer, char(1);
DEFINE Output_MusicCode varchar(15);
DEFINE Output_Genre varchar(10);
DEFINE Output_Publishing, Output_Album varchar(20);
DEFINE Output_MusicName varchar(30);
DEFINE Output_Arthist varchar(25);
DEFINE Output_Part char(1);
DEFINE Output_MusicId, Output_ShelfId, Output_Year, Output_Layer integer;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LET Input_MusicCode = concat("%",concat(Input_MusicCode,"%"));
LET Input_Genre = concat("%",concat(Input_Genre,"%"));
LET Input_MusicName = concat("%",concat(Input_MusicName,"%"));
LET Input_Arthist = concat("%",concat(Input_Arthist,"%"));
LET Input_Album = concat("%",concat(Input_Album,"%"));
LET Input_Publishing = concat("%",concat(Input_Publishing,"%"));
IF (Input_MusicID != 0) and (Input_Year = 0) THEN
    FOREACH Cursor1 FOR
        SELECT t1.musicid, musiccode, musicname, arthist, album, publishing, year, genre,
t2.shelfid,layer,part
        INTO Output_MusicId, Output_MusicCode, Output_MusicName, Output_Arthist,
Output_Album, Output_Publishing, Output_Year, Output_Genre, Output_ShelfId, Output_Layer,
Output_Part
        FROM music t1,addressmusic t2
        WHERE t1.musicid = Input_MusicID and musiccode like Input_MusicCode and
musicname like Input_MusicName and arthist like Input_Arthist and genre like Input_Genre and album
like Input_Album and publishing like Input_Publishing and t2.musicid = t1.musicid
        RETURN Output_MusicId, Output_MusicCode, Output_Musicname, Output_Arthist,
Output_Album, Output_Publishing, Output_Year, Output_Genre, Output_ShelfId, Output_Layer,
Output_Part with resume;
    END FOREACH;
ELIF (Input_MusicID = 0) and (Input_Year != 0) THEN
    FOREACH Cursor1 FOR
        SELECT t1.musicid, musiccode, musicname, arthist, album, publishing, year, genre,
t2.shelfid,layer,part
        INTO Output_MusicId, Output_MusicCode, Output_MusicName, Output_Arthist,
Output_Album, Output_Publishing, Output_Year, Output_Genre, Output_ShelfId, Output_Layer,
Output_Part
        FROM music t1,addressmusic t2
        WHERE musiccode like Input_MusicCode and musicname like Input_MusicName and
arthist like Input_Arthist and genre like Input_Genre and album like Input_Album and publishing like
Input_Publishing and year = Input_Year and t2.musicid = t1.musicid

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RETURN Output_MusicId, Output_MusicCode, Output_Musicname, Output_Arthist,
Output_Album, Output_Publishing, Output_Year, Output_Genre, Output_ShelfId, Output_Layer,
Output_Part with resume;
END FOREACH;
ELIF (Input_MusicID != 0) and (Input_Year != 0) THEN
FOREACH Cursor1 FOR
SELECT t1.musicid, musiccode, musicname, arthist, album, publishing, year, genre,
t2.shelfid, layer, part
INTO Output_MusicId, Output_MusicCode, Output_MusicName, Output_Arthist,
Output_Album, Output_Publishing, Output_Year, Output_Genre, Output_ShelfId, Output_Layer,
Output_Part
FROM music t1,addressmusic t2
WHERE t1.musicid = Input_MusicID and musiccode like Input_MusicCode and
musicname like Input_MusicName and arthist like Input_Arthist and genre like Input_Genre and album
like Input_Album and publishing like Input_Publishing and year = Input_Year and t2.musicid =
t1.musicid
RETURN Output_MusicId, Output_MusicCode, Output_Musicname, Output_Arthist,
Output_Album, Output_Publishing, Output_Year, Output_Genre, Output_ShelfId, Output_Layer,
Output_Part with resume;
END FOREACH;
ELSE FOREACH Cursor1 FOR
SELECT t1.musicid, musiccode, musicname, arthist, album, publishing, year, genre,
t2.shelfid, layer,part
INTO Output_MusicId, Output_MusicCode, Output_MusicName, Output_Arthist,
Output_Album, Output_Publishing, Output_Year, Output_Genre, Output_ShelfId, Output_Layer,
Output_Part
FROM music t1,addressmusic t2
WHERE musiccode like Input_MusicCode and musicname like Input_MusicName and
arthist like Input_Arthist and genre like Input_Genre and album like Input_Album and publishing like
Input_Publishing and t2.musicid = t1.musicid
RETURN Output_MusicId, Output_MusicCode, Output_Musicname, Output_Arthist,
Output_Album, Output_Publishing, Output_Year, Output_Genre, Output_ShelfId, Output_Layer,
Output_Part with resume;
END FOREACH;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    END IF;
END FUNCTION;
:-----:

```

```

CREATE FUNCTION searchdetailmovie(Input_MovieID integer,Input_MovieCode varchar
(15),Input_MovieName varchar(35),Input_Actor varchar(20),Input_Publishing varchar(20),Input_year
integer,Input_MovieClass varchar(20)) returning integer,varchar(15),varchar(35),varchar(20),varchar
(20),varchar(20),integer,varchar(20),integer,integer,char(1);

    DEFINE Output_MovieCode varchar(15);
    DEFINE Output_Publishing,Output_MovieClass,Output_Actor1,Output_Actor2 varchar(20);
    DEFINE Output_MovieName varchar(35);
    DEFINE Output_Part char(1);
    DEFINE Output_MovieId,Output_ShelfId,Output_Year,Output_Layer integer;
    LET Input_MovieCode = concat("%",concat(Input_MovieCode,"%"));
    LET Input_MovieClass = concat("%",concat(Input_MovieClass,"%"));
    LET Input_MovieName = concat("%",concat(Input_MovieName,"%"));
    LET Input_Actor = concat("%",concat(Input_Actor,"%"));
    LET Input_Publishing = concat("%",concat(Input_Publishing,"%"));
    IF (Input_MovieID != 0) and (Input_Year = 0) THEN
        FOREACH Cursor1 FOR
            SELECT t1.movieid, moviecode, moviename, actor.actor1, actor.actor2, publishing, year,
movieclass, t2.shelfid,layer,part
            INTO Output_MovieId, Output_MovieCode, Output_MovieName, Output_Actor1,
Output_Actor2, Output_Publishing, Output_Year, Output_MovieClass, Output_ShelfId, Output_Layer,
Output_Part
            FROM movie t1,addressmovie t2
            WHERE t1.movieid = Input_MovieID and moviecode like Input_MovieCode and
moviename like Input_MovieName and (actor.actor1 like Input_Actor or actor.actor2 like Input_Actor)
and movieclass like Input_MovieClass and publishing like Input_Publishing and t2.movieid =
t1.movieid
            RETURN Output_MovieId, Output_MovieCode, Output_Moviename, Output_Actor1,
Output_Actor2, Output_Publishing, Output_Year, Output_MovieClass, Output_ShelfId, Output_Layer,
Output_Part with resume;
        END FOREACH;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ELIF (Input_MovieID = 0) and (Input_Year != 0) THEN
    FOREACH Cursor1 FOR
        SELECT t1.movieid, moviecode, moviename, actor.actor1, actor.actor2, publishing, year,
        movieclass, t2.shelfid, layer,part
        INTO Output_MovieId, Output_MovieCode, Output_MovieName, Output_Actor1,
        Output_Actor2, Output_Publishing, Output_Year, Output_MovieClass, Output_ShelfId, Output_Layer,
        Output_Part
        FROM movie t1,addressmovie t2
        WHERE moviecode like Input_MovieCode and moviename like Input_MovieName and
        (actor.actor1 like Input_Actor or actor.actor2 like Input_Actor) and movieclass like Input_MovieClass
        and publishing like Input_Publishing and t2.movieid = t1.movieid and year = Input_Year
        RETURN Output_MovieId, Output_MovieCode, Output_Moviename, Output_Actor1,
        Output_Actor2, Output_Publishing, Output_Year, Output_MovieClass, Output_ShelfId, Output_Layer,
        Output_Part with resume;
    END FOREACH;
ELIF (Input_MovieID != 0) and (Input_Year != 0) THEN
    FOREACH Cursor1 FOR
        SELECT t1.movieid, moviecode, moviename, actor.actor1, actor.actor2, publishing, year,
        movieclass, t2.shelfid, layer,part
        INTO Output_MovieId, Output_MovieCode, Output_MovieName, Output_Actor1,
        Output_Actor2, Output_Publishing, Output_Year, Output_MovieClass, Output_ShelfId, Output_Layer,
        Output_Part
        FROM movie t1,addressmovie t2
        WHERE t1.movieid = Input_MovieID and moviecode like Input_MovieCode and
        moviename like Input_MovieName and (actor.actor1 like Input_Actor or actor.actor2 like Input_Actor)
        and movieclass like Input_MovieClass and publishing like Input_Publishing and t2.movieid =
        t1.movieid and year = Input_Year
        RETURN Output_MovieId, Output_MovieCode, Output_Moviename, Output_Actor1,
        Output_Actor2, Output_Publishing, Output_Year, Output_MovieClass, Output_ShelfId, Output_Layer,
        Output_Part with resume;
    END FOREACH;
ELSE FOREACH Cursor1 FOR
    SELECT t1.movieid, moviecode, moviename, actor.actor1, actor.actor2, publishing,
    year, movieclass, t2.shelfid, layer,part

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        INTO Output_MovieId, Output_MovieCode, Output_MovieName, Output_Actor1,
        Output_Actor2, Output_Publishing, Output_Year, Output_MovieClass, Output_ShelfId, Output_Layer,
        Output_Part

```

```

        FROM movie t1,addressmovie t2

```

```

        WHERE moviecode like Input_MovieCode and moviename like Input_MovieName
        and (actor.actor1 like Input_Actor or actor.actor2 like Input_Actor) and movieclass like
        Input_MovieClass and publishing like Input_Publishing and t2.movieid = t1.movieid

```

```

        RETURN Output_MovieId, Output_MovieCode, Output_Moviename, Output_Actor1,
        Output_Actor2, Output_Publishing, Output_Year, Output_MovieClass, Output_ShelfId, Output_Layer,
        Output_Part with resume;

```

```

    END FOREACH;

```

```

END IF;

```

```

END FUNCTION;

```

```

:-----:

```

```

CREATE FUNCTION searchdetailbook (Input_BookId integer, Input_BookCode varchar(15),
Input_ISBN varchar(13), Input_BookName varchar(40), Input_Author varchar(20), Input_Publishing
varchar(20), Input_year integer, Input_BookClass varchar(15)) returning integer, varchar(15), varchar
(40), varchar(20), varchar(20), varchar(20), varchar(13), varchar(20), integer, varchar(15), integer,
integer, char(1);

```

```

    DEFINE Output_BookCode,Output_BookClass varchar(15);

```

```

    DEFINE Output_ISBN varchar(13);

```

```

    DEFINE Output_Publishing,Output_Author1,Output_Author2,Output_Author3 varchar(20);

```

```

    DEFINE Output_BookName varchar(40);

```

```

    DEFINE Output_Part char(1);

```

```

    DEFINE Output_BookId,Output_ShelfId,Output_Year,Output_Layer integer;

```

```

    LET Input_BookCode = concat("%",concat(Input_BookCode,"%"));

```

```

    LET Input_ISBN = concat("%",concat(Input_ISBN,"%"));

```

```

    LET Input_BookName = concat("%",concat(Input_BookName,"%"));

```

```

    LET Input_Author = concat("%",concat(Input_Author,"%"));

```

```

    LET Input_BookClass = concat("%",concat(Input_BookClass,"%"));

```

```

    LET Input_Publishing = concat("%",concat(Input_Publishing,"%"));

```

```

    IF (Input_BookID != 0) and (Input_Year = 0) THEN

```

```

        FOREACH Cursor1 FOR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SELECT t1.bookid, bookcode, bookname, author.author1, author.author2, author.author3, isbn,
publishing, year,bookclass,t2.shelfid,layer,part
    INTO Output_BookId, Output_BookCode, Output_BookName, Output_Author1,
Output_Author2, Output_Author3, Output_ISBN, Output_Publishing, Output_Year, Output_BookClass,
Output_ShelfId, Output_Layer, Output_Part
    FROM book t1,addressbook t2
    WHERE t1.bookid = Input_BookID and bookcode like Input_BookCode and bookname
like Input_BookName and (author.author1 like Input_Author or author.author2 like Input_Author or
author.author3 like Input_Author) and bookclass like Input_Bookclass and isbn like Input_ISBN and
publishing like Input_Publishing and t2.bookid = t1.bookid
    RETURN Output_BookId, Output_BookCode, Output_Bookname, Output_Author1,
Output_Author2, Output_Author3, Output_ISBN, Output_Publishing, Output_Year, Output_BookClass,
Output_ShelfId, Output_Layer, Output_Part with resume;
END FOREACH;
ELIF (Input_BookID = 0) and (Input_Year != 0) THEN
    FOREACH Cursor1 FOR
        SELECT t1.bookid, bookcode, bookname, author.author1, author.author2, author.author3, isbn,
publishing, year, bookclass, t2.shelfid, layer,part
        INTO Output_BookId, Output_BookCode, Output_BookName, Output_Author1,
Output_Author2, Output_Author3, Output_ISBN, Output_Publishing, Output_Year, Output_BookClass,
Output_ShelfId, Output_Layer,Output_Part
        FROM book t1,addressbook t2
        WHERE bookcode like Input_BookCode and bookname like Input_BookName and
(author.author1 like Input_Author or author.author2 like Input_Author or author.author3 like
Input_Author) and bookclass like Input_Bookclass and isbn like Input_ISBN and publishing like
Input_Publishing and t2.bookid = t1.bookid and year = Input_Year
        RETURN Output_BookId, Output_BookCode, Output_Bookname, Output_Author1,
Output_Author2, Output_Author3, Output_ISBN, Output_Publishing, Output_Year, Output_BookClass,
Output_ShelfId, Output_Layer, Output_Part with resume;
    END FOREACH;
    ELIF (Input_BookID != 0) and (Input_Year != 0) THEN
        FOREACH Cursor1 FOR
            SELECT t1.bookid, bookcode, bookname, author.author1, author.author2, author.author3, isbn,
publishing, year, bookclass, t2.shelfid, layer,part

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    INTO Output_BookId, Output_BookCode, Output_BookName, Output_Author1,
    Output_Author2, Output_Author3, Output_ISBN, Output_Publishing, Output_Year, Output_BookClass,
    Output_ShelfId, Output_Layer, Output_Part

```

```

    FROM book t1,addressbook t2

```

```

    WHERE t1.bookid = Input_BookID and bookcode like Input_BookCode and bookname
    like Input_BookName and (author.author1 like Input_Author or author.author2 like Input_Author or
    author.author3 like Input_Author) and bookclass like Input_Bookclass and isbn like Input_ISBN and
    publishing like Input_Publishing and t2.bookid = t1.bookid and year = Input_Year

```

```

    RETURN Output_BookId, Output_BookCode, Output_Bookname, Output_Author1,
    Output_Author2, Output_Author3, Output_ISBN, Output_Publishing, Output_Year, Output_BookClass,
    Output_ShelfId, Output_Layer, Output_Part with resume;

```

```

    END FOREACH;

```

```

    ELSE FOREACH Cursor1 FOR

```

```

    SELECT t1.bookid, bookcode, bookname, author.author1, author.author2, author.author3, isbn,
    publishing, year, bookclass, t2.shelfid, layer,part

```

```

    INTO Output_BookId, Output_BookCode, Output_BookName, Output_Author1,
    Output_Author2, Output_Author3, Output_ISBN, Output_Publishing, Output_Year, Output_BookClass,
    Output_ShelfId, Output_Layer, Output_Part

```

```

    FROM book t1,addressbook t2

```

```

    WHERE bookcode like Input_BookCode and bookname like Input_BookName and
    (author.author1 like Input_Author or author.author2 like Input_Author or author.author3 like
    Input_Author) and bookclass like Input_Bookclass and isbn like Input_ISBN and publishing like
    Input_Publishing and t2.bookid = t1.bookid

```

```

    RETURN Output_BookId, Output_BookCode, Output_Bookname, Output_Author1,
    Output_Author2, Output_Author3, Output_ISBN, Output_Publishing, Output_Year, Output_BookClass,
    Output_ShelfId, Output_Layer, Output_Part with resume;

```

```

    END FOREACH;

```

```

    END IF;

```

```

    END FUNCTION;

```

```

    :-----:

```

```

    CREATE FUNCTION searchupdatemusic(Input_MusicId integer) returning varchar(15),varchar
    (30),varchar(25),varchar(20),varchar(20),integer,varchar(10),integer,integer,char(1),integer

```

```

    DEFINE Output_MusicCode varchar(15);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DEFINE Output_Genre varchar(10);
DEFINE Output_Arthist varchar(25);
DEFINE Output_Publishing,Output_Album varchar(20);
DEFINE Output_MusicName varchar(30);
DEFINE Output_Part char(1);
DEFINE Output_CountLayer,Output_ShelfId,Output_Year,Output_Layer integer;
    FOREACH Cursor1 FOR
        SELECT musiccode, musicname, arthist, album, publishing, year, genre, t2.shelfid, t2.layer,
part, t3.layer
        INTO Output_MusicCode, Output_MusicName, Output_Arthist, Output_Album,
Output_Publishing, Output_Year, Output_Genre, Output_ShelfId, Output_Layer, Output_Part,
Output_CountLayer
        FROM music t1,addressmusic t2,musicshelf t3
        WHERE t1.musicid = Input_MusicId and t2.musicid = t1.musicid and t3.shelfid = t2.shelfid
        RETURN Output_MusicCode, Output_Musicname, Output_Arthist, Output_Album,
Output_Publishing, Output_Year, Output_Genre, Output_ShelfId, Output_Layer, Output_Part,
Output_CountLayer with resume;
    END FOREACH;
END FUNCTION;
:-----:

```

```

CREATE FUNCTION searchupdatemovie(Input_MovieId integer) returning varchar(15),varchar
(35),varchar(20),varchar(20),varchar(20),integer,varchar(20),integer,integer,char(1),integer
    DEFINE Output_MovieCode varchar(15);
    DEFINE Output_Publishing,Output_Actor1,Output_Actor2,Output_MovieClass varchar(20);
    DEFINE Output_MovieName varchar(35);
    DEFINE Output_Part char(1);
    DEFINE Output_CountLayer,Output_ShelfId,Output_Year,Output_Layer integer;
        FOREACH Cursor1 FOR
            SELECT moviecode, moviename, actor.actor1, actor.actor2, publishing, year, movieclass,
t2.shelfid, t2.layer, part, t3.layer
            INTO Output_MovieCode, Output_MovieName, Output_Actor1, Output_Actor2,
Output_Publishing, Output_Year, Output_MovieClass, Output_ShelfId, Output_Layer, Output_Part,
Output_CountLayer

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

FROM movie t1,addressmovie t2,movieshelf t3

WHERE t1.movieid = Input_MovieId and t2.movieid = t1.movieid and t3.shelfid =
t2.shelfid

RETURN Output_MovieCode, Output_Moviename, Output_Actor1, Output_Actor2,
Output_Publishing, Output_Year, Output_MovieClass, Output_ShelfId, Output_Layer, Output_Part,
Output_CountLayer with resume;

END FOREACH;
END FUNCTION;
:-----:

```

```

CREATE FUNCTION searchupdatebook(Input_BookId integer) returning varchar(15),varchar
(40),varchar(20),varchar(20),varchar(20),varchar(13),varchar(20),integer,varchar(15),integer,integer,
char(1),integer

DEFINE Output_BookCode,Output_BookClass varchar(15);
DEFINE Output_ISBN varchar(13);
DEFINE Output_Publishing,Output_Author1,Output_Author2,Output_Author3 varchar(20);
DEFINE Output_BookName varchar(40);
DEFINE Output_Part char(1);
DEFINE Output_CountLayer,Output_ShelfId,Output_Year,Output_Layer integer;

FOREACH Cursor1 FOR

SELECT bookcode, bookname, author.author1, author.author2, author.author3, isbn,
publishing, year, bookclass, t2.shelfid, t2.layer, part,t3.layer

INTO Output_BookCode, Output_BookName, Output_Author1, Output_Author2,
Output_Author3, Output_ISBN, Output_Publishing, Output_Year, Output_BookClass, Output_ShelfId,
Output_Layer, Output_Part, Output_CountLayer

FROM book t1,addressbook t2,bookshelf t3

WHERE t1.bookid = Input_BookId and t2.bookid = t1.bookid and t3.shelfid = t2.shelfid

RETURN Output_BookCode, Output_Bookname, Output_Author1, Output_Author2,
Output_Author3, Output_ISBN, Output_Publishing, Output_Year, Output_BookClass, Output_ShelfId,
Output_Layer, Output_Part, Output_CountLayer with resume;

END FOREACH;
END FUNCTION;
:-----:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CREATE FUNCTION INSERTMUSIC(Input_MusicCode varchar(15),Input_MusicName varchar(30),
Input_Arthist varchar(25),Input_Album varchar(20),Input_Genre varchar(10),Input_Publishing varchar
(20), Input_year integer,Input_Song varchar(100),Input_Shelfid integer,Input_Layer integer,Input_Part
char(1)) returning integer,char(1);
    DEFINE Output_Count integer;
    DEFINE Result varchar(15);
    DEFINE Flag_Insert char(1);
    LET Result = ( SELECT musiccode FROM music WHERE musiccode = Input_MusicCode );
    IF ( Result IS NULL ) THEN
        INSERT INTO Music
VALUES(0,Input_MusicCode,Input_MusicName,Input_Arthist,Input_Album,Input_Genre,Input_Publis
hing,Input_Year,Input_Song);
        LET Output_Count = (SELECT musicid FROM music WHERE musiccode = Input_MusicCode);
        INSERT INTO ADDRESSMUSIC
VALUES(Output_Count,Input_Shelfid,Input_Layer,Input_Part);
        LET Flag_Insert = "Y";
        ELSE LET Flag_Insert = "N";
    END IF;
    LET Output_Count = ( SELECT COUNT(*) FROM music );
    RETURN Output_Count,Flag_Insert;
END FUNCTION;
:-----:

```

```

CREATE FUNCTION INSERTMOVIE(Input_MovieCode varchar(15),Input_MovieName varchar(35),
Input_Actor1 varchar(20), Input_Actor2 varchar(20), Input_MovieClass varchar(20), Input_Publishing
varchar(20), Input_year integer,Input_Film varchar(100), Input_Shelfid integer, Input_Layer integer,
Input_Part char(1)) returning integer,char(1);
    DEFINE Output_Count integer;
    DEFINE Result varchar(15);
    DEFINE Flag_Insert char(1);
    LET Result = ( SELECT moviecode FROM movie WHERE moviecode = Input_MovieCode );
    IF ( Result IS NULL ) THEN
        INSERT INTO Movie

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

VALUES(0, Input_MovieCode, Input_MovieName, ROW(Input_Actor1,Input_Actor2)::Actor_t,
Input_MovieClass,Input_Publishing,Input_Year,Input_Film);
LET Output_Count = (SELECT movieid FROM movie WHERE moviecode = Input_MovieCode);
INSERT INTO ADDRESSMOVIE
VALUES(Output_Count,Input_Shelfid,Input_Layer,Input_Part);
LET Flag_Insert = "Y";
ELSE LET Flag_Insert = "N";
END IF;
LET Output_Count = ( SELECT COUNT(*) FROM movie );
RETURN Output_Count,Flag_Insert;
END FUNCTION;
:-----:

```

```

CREATE FUNCTION INSERTBOOK(Input_BookCode varchar(15),Input_ISBN varchar(13),
Input_BookName varchar(40),Input_Author1 varchar(20),Input_Author2 varchar(20),Input_Author3
varchar(20),Input_BookClass varchar(15),Input_Publishing varchar(20),Input_year integer,Input_Title
varchar(100),Input_Preface varchar(100),Input_Shelfid integer,Input_Layer integer,Input_Part char(1))
returning integer,char(1);
DEFINE Output_Count integer;
DEFINE Result varchar(15);
DEFINE Flag_Insert char(1);
LET Result = ( SELECT BOOKCODE FROM BOOK WHERE BOOKCODE = Input_BookCode );
IF ( Result IS NULL ) THEN
INSERT INTO BOOK
VALUES(0, Input_BookCode, Input_ISBN, Input_BookName, ROW(Input_Author1, Input_Author2,
Input_Author3)::Author_t, Input_BookClass,Input_Publishing,Input_year,Input_Title,Input_Preface);
LET Output_Count = (SELECT BOOKID FROM BOOK WHERE BOOKCODE =
Input_Bookcode);
INSERT INTO ADDRESSBOOK
VALUES(Output_Count,Input_Shelfid,Input_Layer,Input_Part);
LET Flag_Insert = "Y";
ELSE LET Flag_Insert = "N";
END IF;
LET Output_Count = ( SELECT COUNT(*) FROM BOOK );

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RETURN Output_Count,Flag_Insert;
END FUNCTION;
:-----:

CREATE FUNCTION getinfomusic(input_musiccode varchar(15), input_musicname varchar
(30),input_arthist varchar(25),input_album varchar(20), input_year integer, input_genre varchar(10),
input_publishing varchar(20)) returning varchar(15),varchar(30),varchar(25),varchar(20),integer,varchar
(10),varchar(20);

DEFINE output_musiccode varchar(15);
DEFINE output_musicname varchar(30);
DEFINE output_arthist varchar(25);
DEFINE output_album,output_publishing varchar(20);
DEFINE output_year integer;
DEFINE output_genre varchar(10);
LET input_musiccode = concat("%",concat(input_musiccode,"%"));
LET input_musicname = concat("%",concat(input_musicname,"%"));
LET input_arthist = concat("%",concat(input_arthist,"%"));
LET input_album = concat("%",concat(input_album,"%"));
LET input_genre = concat("%",concat(input_genre,"%"));
LET input_publishing = concat("%",concat(input_publishing,"%"));
IF input_year != 0 THEN
    FOREACH Cursor1 FOR
        select musiccode,musicname,arthist,album,year,genre,publishing
        INTO output_musiccode, output_musicname, output_arthist, output_album, output_year,
output_genre, output_publishing
        from music where musiccode like input_musiccode and musicname like input_musicname
and arthist like input_arthist and album like input_album and year = input_year and genre like
input_genre and publishing like input_publishing
        RETURN output_musiccode, output_musicname, output_arthist, output_album,
output_year, output_genre, output_publishing with resume;
    END FOREACH;
ELSE FOREACH Cursor1 FOR
    select musiccode,musicname,arthist,album,year,genre,publishing

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        into output_musiccode, output_musicname, output_artist, output_album, output_year,
        output_genre, output_publishing

```

```

        from music where musiccode like input_musiccode and musicname like input_musicname
        and artist like input_artist and album like input_album and genre like input_genre and publishing like
        input_publishing

```

```

        RETURN output_musiccode, output_musicname, output_artist, output_album,
        output_year, output_genre, output_publishing with resume;

```

```

    END FOREACH;

```

```

END IF;

```

```

END FUNCTION;

```

```

:-----:

```

```

CREATE FUNCTION getinfomovie(input_moviecode varchar(15),input_moviename varchar
(35),input_actor varchar(20),input_movieclass varchar(20),input_year integer,input_publishing varchar
(20)) returning varchar(15),varchar(35),varchar(20),integer,varchar(20);

```

```

    DEFINE output_moviecode varchar(15);

```

```

    DEFINE output_moviename varchar(35);

```

```

    DEFINE output_movieclass varchar(20);

```

```

    DEFINE output_year integer;

```

```

    DEFINE output_publishing varchar(20);

```

```

    LET input_moviecode = concat("%",concat(input_moviecode,"%"));

```

```

    LET input_moviename = concat("%",concat(input_moviename,"%"));

```

```

    LET input_actor = concat("%",concat(input_actor,"%"));

```

```

    LET input_movieclass = concat("%",concat(input_movieclass,"%"));

```

```

    LET input_publishing = concat("%",concat(input_publishing,"%"));

```

```

    IF input_year != 0 THEN

```

```

        FOREACH Cursor1 FOR

```

```

            select moviecode,moviename,movieclass,year,publishing

```

```

            into output_moviecode,output_moviename,output_movieclass,output_year,output_publishing

```

```

            from movie where moviecode like input_moviecode and moviename like input_moviename

```

```

            and (actor.actor1 like input_actor or actor.actor2 like input_actor) and movieclass like input_movieclass
            and year = input_year and publishing like input_publishing

```

```

            RETURN output_moviecode, output_moviename, output_movieclass, output_year,
            output_publishing with resume;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

END FOREACH;

ELSE FOREACH Cursor1 FOR
    select moviecode,moviename,movieclass,year,publishing
    into output_moviecode,output_moviename,output_movieclass,output_year,output_publishing
    from movie where moviecode like input_moviecode and moviename like input_moviename
    and (actor.actor1 like input_actor or actor.actor2 like input_actor) and movieclass like input_movieclass
    and publishing like input_publishing

    RETURN output_moviecode, output_moviename, output_movieclass, output_year,
output_publishing with resume;

END FOREACH;

END IF;

END FUNCTION;

```

```

:-----:

```

```

CREATE FUNCTION getinfobook(input_bookcode varchar(15),input_bookname varchar
(40),input_author varchar(20),input_bookclass varchar(15),input_publishing varchar(20),input_year
integer) returning varchar(15),varchar(40),varchar(15),varchar(15),integer;

    DEFINE output_bookcode,output_bookclass varchar(15);
    DEFINE output_author,output_publishing varchar(15);
    DEFINE output_bookname varchar(40);
    DEFINE output_year integer;

    LET input_bookcode = concat("%",concat(input_bookcode,"%"));
    LET input_bookname = concat("%",concat(input_bookname,"%"));
    LET input_author = concat("%",concat(input_author,"%"));
    LET input_bookclass = concat("%",concat(input_bookclass,"%"));
    LET input_publishing = concat("%",concat(input_publishing,"%"));

    IF input_year != 0 THEN
        FOREACH Cursor1 FOR
            select bookcode,bookname,bookclass,publishing,year
            into output_bookcode,output_bookname,output_bookclass,output_publishing,output_year
            from book where bookcode like input_bookcode and bookname like input_bookname and
            (author.author1 like input_author or author.author2 like input_author or author.author3 like
            input_author) and bookclass like input_bookclass and publishing like input_publishing and year =
            input_year

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RETURN output_bookcode, output_bookname, output_bookclass, output_publishing,
output_year with resume;

```

```

END FOREACH;

```

```

ELSE FOREACH Cursor1 FOR

```

```

select bookcode,bookname,bookclass,publishing,year

```

```

into output_bookcode,output_bookname,output_bookclass,output_publishing,output_year

```

```

from book where bookcode like input_bookcode and bookname like input_bookname and
(author.author1 like input_author or author.author2 like input_author or author.author3 like
input_author) and bookclass like input_bookclass and publishing like input_publishing

```

```

RETURN output_bookcode, output_bookname, output_bookclass, output_publishing,
output_year with resume;

```

```

END FOREACH;

```

```

END IF;

```

```

END FUNCTION;

```

```

:-----:

```

```

CREATE FUNCTION getdetailmusic(input_musiccode varchar(15)) returning integer,varchar
(15),varchar(30),varchar(25),varchar(20),integer,varchar(10),varchar(20),integer,integer,varchar
(8),media_t,char(1),integer,integer;

```

```

DEFINE output_musiccode varchar(15);

```

```

DEFINE output_musicname varchar(30);

```

```

DEFINE output_year integer;

```

```

DEFINE output_genre varchar(10);

```

```

DEFINE output_song media_t;

```

```

DEFINE output_arhist varchar(25);

```

```

DEFINE output_album,output_publishing varchar(20);

```

```

DEFINE output_musicid,output_shelfid,output_layer,output_position_x,output_position_y integer;

```

```

DEFINE output_part char(1);

```

```

DEFINE output_room varchar(8);

```

```

FOREACH Cursor1 FOR

```

```

select t1.musicid, musiccode, musicname, arhist, album, year, genre, publishing, t2.shelfid,
t2.layer, t5.roomname, song, part, position.x, position.y

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

into output_musicid, output_musiccode, output_musicname, output_artist, output_album,
output_year, output_genre, output_publishing, output_shelfid, output_layer, output_room, output_song,
output_part, output_position_x, output_position_y

```

```

from music t1,addressmusic t2,musicshelf t3,addressmusicshelf t4,room t5

```

```

where t1.musiccode = input_musiccode and t2.musicid = t1.musicid and t3.shelfid = t2.shelfid
and t4.roomid = t5.roomid and t3.shelfid = t4.shelfid

```

```

RETURN output_musicid, output_musiccode, output_musicname, output_artist, output_album,
output_year, output_genre, output_publishing, output_shelfid, output_layer, output_room, output_song,
output_part, output_position_x, output_position_y with resume;

```

```

END FOREACH

```

```

END FUNCTION

```

```

:-----:

```

```

CREATE FUNCTION getdetailmovie(input_moviecode varchar(15)) returning integer,varchar
(15),varchar(35),varchar(20),varchar(20),varchar(20),integer,varchar(20),integer,integer,varchar
(8),media_t,char(1),integer,integer;

```

```

DEFINE output_moviecode varchar(15);

```

```

DEFINE output_moviename varchar(35);

```

```

DEFINE output_year integer;

```

```

DEFINE output_film media_t;

```

```

DEFINE output_movieclass,output_publishing,output_actor1,output_actor2 varchar(20);

```

```

DEFINE output_movieid,output_shelfid,output_layer,output_position_x,output_position_y integer;

```

```

DEFINE output_part char(1);

```

```

DEFINE output_room varchar(8);

```

```

FOREACH Cursor1 FOR

```

```

select t1.movieid, moviecode, moviename, actor.actor1, actor.actor2, movieclass, year, publishing,
t2.shelfid, t2.layer, t5.roomname, film, part, position.x, position.y

```

```

into output_movieid, output_moviecode, output_moviename, output_actor1, output_actor2,
output_movieclass, output_year, output_publishing, output_shelfid, output_layer, output_room,
output_film, output_part, output_position_x, output_position_y

```

```

from movie t1,addressmovie t2,movieshelf t3 ,addressmovieshelf t4 ,room t5

```

```

where t1.moviecode = input_moviecode and t2.movieid = t1.movieid and t3.shelfid = t2.shelfid
and t4.roomid = t5.roomid and t3.shelfid = t4.shelfid

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RETURN output_movieid,output_moviecode,output_moviename,output_actor1,output_actor2,
output_movieclass, output_year,output_publishing,output_shelfid,output_layer,output_room,
output_film, output_part, output_position_x, output_position_y with resume;

```

```

END FOREACH;

```

```

END FUNCTION;

```

```

:-----:

```

```

CREATE FUNCTION getdetailbook(input_bookcode varchar(15)) returning integer,varchar
(15),varchar(40),varchar(15),varchar(15),varchar(15),varchar(15),varchar(15),integer,varchar
(15),integer,integer,varchar(8),media_t,char(1),integer,integer,media_t;

```

```

DEFINE output_bookcode, output_bookclass, output_isbn, output_author1, output_author2,
output_author3, output_publishing varchar(15);

```

```

DEFINE output_bookname varchar(40);

```

```

DEFINE output_year integer;

```

```

DEFINE output_image media_t;

```

```

DEFINE output_bookid,output_shelfid,output_layer,output_position_x,output_position_y integer;

```

```

DEFINE output_part char(1);

```

```

DEFINE output_room varchar(8);

```

```

DEFINE output_preface media_t;

```

```

FOREACH Cursor1 FOR

```

```

select t1.bookid, bookcode, bookname, author.author1, author.author2, author.author3, bookclass,
publishing, year,isbn, t2.shelfid,t2.layer,t5.roomname,image,part,position.x,position.y,preface

```

```

into output_bookid, output_bookcode, output_bookname, output_author1, output_author2,
output_author3, output_bookclass, output_publishing, output_year, output_isbn, output_shelfid,
output_layer, output_room, output_image, output_part, output_position_x, output_position_y,
output_preface

```

```

from book t1,addressbook t2,bookshelf t3,addressbookshelf t4,room t5

```

```

where t1.bookcode = input_bookcode and t2.bookid = t1.bookid and t3.shelfid = t2.shelfid and
t4.roomid = t5.roomid and t3.shelfid = t4.shelfid

```

```

RETURN output_bookid, output_bookcode, output_bookname, output_author1, output_author2,
output_author3, output_bookclass, output_publishing, output_year, output_isbn, output_shelfid,
output_layer, output_room, output_image, output_part, output_position_x, output_position_y,
output_preface with resume;

```

```

END FOREACH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
END FUNCTION;
```

```
;-----;
```

2. Visual Basic

2.1 About.frm

```
Option Explicit
```

```
Const READ_CONTROL = &H20000
```

```
Const KEY_QUERY_VALUE = &H1
```

```
Const KEY_SET_VALUE = &H2
```

```
Const KEY_CREATE_SUB_KEY = &H4
```

```
Const KEY_ENUMERATE_SUB_KEYS = &H8
```

```
Const KEY_NOTIFY = &H10
```

```
Const KEY_CREATE_LINK = &H20
```

```
Const KEY_ALL_ACCESS = KEY_QUERY_VALUE + KEY_SET_VALUE + _  
KEY_CREATE_SUB_KEY + KEY_ENUMERATE_SUB_KEYS + _  
KEY_NOTIFY + KEY_CREATE_LINK + READ_CONTROL
```

```
Const HKEY_LOCAL_MACHINE = &H80000002
```

```
Const ERROR_SUCCESS = 0
```

```
Const REG_SZ = 1 'Unicode nul terminated string
```

```
Const REG_DWORD = 4 '32-bit number
```

```
Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"
```

```
Const gREGVALSYSINFOLOC = "MSINFO"
```

```
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
```

```
Const gREGVALSYSINFO = "PATH"
```

```
Private Declare Function RegOpenKeyEx Lib "advapi32" Alias "RegOpenKeyExA" (ByVal hKey As  
Long, ByVal lpSubKey As String, ByVal ulOptions As Long, ByVal samDesired As Long, ByRef  
phkResult As Long) As Long
```

```
Private Declare Function RegQueryValueEx Lib "advapi32" Alias "RegQueryValueExA" (ByVal hKey  
As Long, ByVal lpValueName As String, ByVal lpReserved As Long, ByRef lpType As Long, ByVal  
lpData As String, ByRef lpcbData As Long) As Long
```

```
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long) As Long
```

```
Private Sub cmdSysInfo_Click()
```

```
Call StartSysInfo
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub cmdOK_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Public Sub StartSysInfo()
```

```
    On Error GoTo SysInfoErr
```

```
    Dim rc As Long
```

```
    Dim SysInfoPath As String
```

```
    If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO, gREGVALSYSINFO, SysInfoPath) Then
```

```
        ElseIf GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFOLOC, gREGVALSYSINFOLOC, SysInfoPath) Then
```

```
            If (Dir(SysInfoPath & "MSINFO32.EXE") <> "") Then
```

```
                SysInfoPath = SysInfoPath & "MSINFO32.EXE"
```

```
            Else
```

```
                GoTo SysInfoErr
```

```
            End If
```

```
        Else
```

```
            GoTo SysInfoErr
```

```
        End If
```

```
        Call Shell(SysInfoPath, vbNormalFocus)
```

```
        Exit Sub
```

```
SysInfoErr:
```

```
    MsgBox "System Information Is Unavailable At This Time", vbOKOnly
```

```
End Sub
```

```
Public Function GetKeyValue(KeyRoot As Long, KeyName As String, SubKeyRef As String, ByRef KeyVal As String) As Boolean
```

```
    Dim i As Long                                ' Loop Counter
```

```
    Dim rc As Long                                ' Return Code
```

```
    Dim hKey As Long                              ' Handle To An Open Registry Key
```

```
    Dim hDepth As Long
```

```
    Dim KeyValType As Long                        ' Data Type Of A Registry Key
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Dim tmpVal As String          ' Temporary Storage For A Registry Key Value
Dim KeyValSize As Long       ' Size Of Registry Key Variable
rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey) ' Open Registry Key
If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError    ' Handle Error...
tmpVal = String$(1024, 0)      ' Allocate Variable Space
KeyValSize = 1024             ' Mark Variable Size
rc = RegQueryValueEx(hKey, SubKeyRef, 0, _
    KeyValType, tmpVal, KeyValSize) ' Get/Create Key Value
If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError    ' Handle Errors
If (Asc(Mid(tmpVal, KeyValSize, 1)) = 0) Then    ' Win95 Adds Null Terminated String...
    tmpVal = Left(tmpVal, KeyValSize - 1)      ' Null Found, Extract From String
Else
    ' WinNT Does NOT Null Terminate String...
    tmpVal = Left(tmpVal, KeyValSize)          ' Null Not Found, Extract String Only
End If
Select Case KeyValType
    ' Search Data Types...
Case REG_SZ
    ' String Registry Key Data Type
    KeyVal = tmpVal                          ' Copy String Value
Case REG_DWORD
    ' Double Word Registry Key Data Type
    For i = Len(tmpVal) To 1 Step -1
        ' Convert Each Bit
        KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1))) ' Build Value Char. By Char.
    Next
    KeyVal = Format$("&h" + KeyVal)           ' Convert Double Word To String
End Select
GetKeyValue = True                        ' Return Success
rc = RegCloseKey(hKey)                   ' Close Registry Key
Exit Function                             ' Exit
GetKeyError: ' Cleanup After An Error Has Occured...
KeyVal = ""                               ' Set Return Val To Empty String
GetKeyValue = False                       ' Return Failure
rc = RegCloseKey(hKey)                   ' Close Registry Key
End Function
:-----:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Book.frm

Option Explicit

Dim oVTableResult As ddoTable

Dim checkInsert As Boolean

Dim CountRowFirst As ddoTable

Private Function PrepareParameter() As String

Dim P_SQL As String

Dim Part As String

If cmbIPart.ListIndex = 0 Then

Part = "L"

ElseIf cmbIPart.ListIndex = 1 Then

Part = "M"

ElseIf cmbIPart.ListIndex = 2 Then

Part = "R"

Else: Part = "L"

End If

P_SQL = "" + UCase(txtIcode.Text) + """,""" + UCase(txtIISBN.Text) + """,""" + UCase
(txtIName.Text) + """,""" + _UCase(txtIAuthor1.Text) + """,""" + UCase(txtIAuthor2.Text) + """,""" +
UCase(txtIAuthor3.Text) + """,""" + _UCase(cmbIClass.Text) + """,""" + UCase(txtIPublishing.Text) +
"""," + (txtIYear.Text) + "," + _"""" + txtITitle.Text + """,""

If txtIPreface.Text = "" Then

txtIPreface.Text = "NULL"

P_SQL = P_SQL + txtIPreface + ","

Else: P_SQL = P_SQL + """" + txtIPreface.Text + """,""

End If

P_SQL = P_SQL + cmbIShelf.Text + "," + cmbILayer + """," + Part + """"

PrepareParameter = P_SQL

End Function

Private Sub Check1_Click(Index As Integer)

If Check1(Index).Value = 1 Then

If Index < 8 Then

Text3(Index + 1).Enabled = True

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Text3(Index + 1).SetFocus
```

```
Else: Select Case Index
```

```
Case 8: cmdUBrowserTitle.Enabled = True
```

```
Case 9: cmdUBrowserPreface.Enabled = True
```

```
Case 10: cmbUClass.Enabled = True
```

```
Case 11: cmbUShelf.Enabled = True
```

```
Case 12: cmbULayer.Enabled = True
```

```
Case 13: cmbUPart.Enabled = True
```

```
End Select
```

```
End If
```

```
Else:
```

```
If Index < 8 Then
```

```
Text3(Index + 1).Enabled = False
```

```
Else: Select Case Index
```

```
Case 8: cmdUBrowserTitle.Enabled = False
```

```
Case 9: cmdUBrowserPreface.Enabled = False
```

```
Case 10: cmbUClass.Enabled = False
```

```
Case 11: cmbUShelf.Enabled = False
```

```
Case 12: cmbULayer.Enabled = False
```

```
Case 13: cmbUPart.Enabled = False
```

```
End Select
```

```
End If
```

```
End If
```

```
End Sub
```

```
Private Sub Check2_Click(Index As Integer)
```

```
If Check2(Index).Value = 1 Then
```

```
If Index <> 7 Then
```

```
With Text1(Index)
```

```
.Text = ""
```

```
.Visible = True
```

```
.SetFocus
```

```
End With
```

```
Else: cmbSClass.Visible = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End If
```

```
Else:
```

```
  If Index <> 7 Then
```

```
    Text1(Index).Visible = False
```

```
    Text1(Index).Text = ""
```

```
  Else: cmbSClass.Visible = False
```

```
  End If
```

```
End If
```

```
End Sub
```

```
Private Sub cmbIShelf_Click()
```

```
Dim i As Integer
```

```
Dim CountLayer As ddoTable
```

```
MainScreen.oDataGroupBook.BeginTransaction
```

```
Set CountLayer = MainScreen.oDataGroupBook.ExecuteSqlCommand("Select layer from bookshelf  
where shelfid = " + cmbIShelf)
```

```
MainScreen.oDataGroupBook.CommitTransaction
```

```
cmbILayer.Clear
```

```
cmbILayer.Text = 1
```

```
For i = 1 To CountLayer.Columns(1).Value
```

```
  cmbILayer.AddItem (i)
```

```
Next i
```

```
End Sub
```

```
Private Sub cmbUShelf_Click()
```

```
Dim i As Integer
```

```
Dim CountLayer As ddoTable
```

```
MainScreen.oDataGroupBook.BeginTransaction
```

```
Set CountLayer = MainScreen.oDataGroupBook.ExecuteSqlCommand("Select layer from bookshelf  
where shelfid = " + cmbUShelf)
```

```
MainScreen.oDataGroupBook.CommitTransaction
```

```
cmbULayer.Clear
```

```
cmbULayer.Text = 1
```

```
For i = 1 To CountLayer.Columns(1).Value
```

```
  cmbULayer.AddItem (i)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Next i
```

```
End Sub
```

```
Private Sub cmdIBrowserPreface_Click()
```

```
DialogInsert.ShowOpen
```

```
DialogInsert.Filter = "All Picture"
```

```
txtIPreface.Text = DialogInsert.FileName
```

```
If checkInsert Then
```

```
    cmdInsert.Enabled = True
```

```
End If
```

```
End Sub
```

```
Private Sub cmdIBrowserTitle_Click()
```

```
DialogInsert.ShowOpen
```

```
DialogInsert.Filter = "All Picture"
```

```
txtITitle.Text = DialogInsert.FileName
```

```
If checkInsert Then
```

```
    cmdInsert.Enabled = True
```

```
End If
```

```
End Sub
```

```
Private Sub cmdInsert_Click()
```

```
Dim Parameter As String
```

```
Dim CountRow As ddoTable
```

```
Parameter = PrepareParameter
```

```
Parameter = "{CALL insertbook(" + Parameter + ")}"
```

```
MainScreen.oDataGroupBook.BeginTransaction
```

```
Set CountRow = MainScreen.oDataGroupBook.ExecuteSQLCommand(Parameter)
```

```
MainScreen.oDataGroupBook.CommitTransaction
```

```
If CountRow.Columns(2).Value = "N" Then
```

```
    MsgBox "รหัสหนังสือที่คุณ Insert เข้ากับข้อมูลที่มีอยู่แล้ว", , "Insert"
```

```
Else: MsgBox "Insert ข้อมูลเรียบร้อยแล้ว", , "Insert"
```

```
End If
```

```
txtICount.Text = CountRow.Columns(1).Value
```

```
MainScreen.oDataGroupBook.DeleteVirtualTable ("CountRow")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
```

```
Private Sub cmdSClear_Click()
```

```
Dim i As Byte
```

```
For i = 0 To 6
```

```
    Check2(i).Value = 0
```

```
    Text1(i).Text = ""
```

```
    Text1(i).Visible = False
```

```
Next i
```

```
    Check2(7).Value = 0
```

```
    cmbSClass.Visible = False
```

```
End Sub
```

```
Private Sub cmdSDelete_Click()
```

```
Dim DeleteSQL As String
```

```
oVTableResult.GotoRecord (TDBGridS1.Row + 1)
```

```
DeleteSQL = "delete from book where bookid = " + oVTableResult.Columns(1).Value
```

```
MainScreen.oDataGroupBook.BeginTransaction
```

```
MainScreen.oDataGroupBook.ExecuteSQLCommand (DeleteSQL)
```

```
Set CountRowFirst = MainScreen.oDataGroupBook.ExecuteSQLCommand("Select count(*) from  
book")
```

```
MainScreen.oDataGroupBook.CommitTransaction
```

```
txtICount.Text = CountRowFirst.Columns(1).Value
```

```
TDBGridS1.Refresh
```

```
Call PassSQL
```

```
End Sub
```

```
Private Sub cmdSSearch_Click()
```

```
Dim check As Boolean
```

```
Dim i As Byte
```

```
check = True
```

```
For i = 0 To 6
```

```
    If (Check2(i) = 1) And (Text1(i).Text = "") Then
```

```
        check = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End If
Next i
If check Then
Call PassSQL
Else: MsgBox "โปรดใส่ข้อความที่คุณต้องการค้นหา หัวข้อที่คุณเช็ค", "Search"
End If
End Sub

```

```

Private Sub PassSQL()
Dim SQLString As String
SQLString = PrepareParameterSearch
SQLString = "{CALL searchdetailbook(" + SQLString + ")}"
MainScreen.oDataGroupBook.BeginTransaction
Set oVTableResult = MainScreen.oDataGroupBook.ExecuteSQLCommand(SQLString)
MainScreen.oDataGroupBook.CommitTransaction
Call ClearGrid
End Sub

```

```

Private Function PrepareParameterSearch() As String

```

```

Dim i As Byte

```

```

Dim SQL As String

```

```

If Check2(0).Value = 0 Then

```

```

    Text1(0).Text = "0"

```

```

End If

```

```

If Check2(6).Value = 0 Then

```

```

    Text1(6).Text = "0"

```

```

End If

```

```

For i = 0 To 5

```

```

    SQL = SQL + UCase(Text1(i).Text)

```

```

    If i = 0 Then

```

```

        SQL = Text1(i).Text + ","

```

```

    ElseIf i <> 5 Then

```

```

        SQL = SQL + ""','

```

```

End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Next i
SQL = SQL + """," + Text1(6).Text + ","
If Check2(7).Value = 0 Then
    SQL = SQL + """"""
Else: SQL = SQL + """" + cmbSClass + """"
End If
PrepareParameterSearch = SQL
End Function

```

```

Private Sub cmdSUpdate_Click()
Dim Search_Update As String
TabBook.Tabs(2).Selected = True
Frame1(1).ZOrder 0
oVTableResult.GotoRecord (TDBGridS1.Row + 1)
Search_Update = "{Call SearchUpdateBook(" + oVTableResult.Columns(1).Value + ")}"
Text3(0).Text = oVTableResult.Columns(1).Value
MainScreen.oDataGroupBook.BeginTransaction
Set oVTableResult = MainScreen.oDataGroupBook.ExecuteSQLCommand(Search_Update)
MainScreen.oDataGroupBook.CommitTransaction
Call Input_to_txt
End Sub

```

```

Private Sub cmdUBrowserPreface_Click()
DialogUpdate.ShowOpen
DialogUpdate.Filter = "All Picture"
Text3(10).Text = DialogUpdate.FileName
End Sub

```

```

Private Sub cmdUBrowserTitle_Click()
DialogUpdate.ShowOpen
DialogUpdate.Filter = "All Picture"
Text3(9).Text = DialogUpdate.FileName
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

cmdUUpdate.Enabled = True
Else: MsgBox "No Data in this Multimedia database", , "No Data"
    Text3(0).Text = ""
cmdUUpdate.Enabled = False
End If
MainScreen.oDataGroupBook.DeleteVirtualTable ("oVTableResult")
End Sub

```

```

Private Sub cmdUUpdate_Click()
Dim ParameterUpdate As String
Dim CheckUpdate As ddoTable
ParameterUpdate = PrepareparameterUpdate
ParameterUpdate = "execute procedure UPDATEBOOK(" + ParameterUpdate + ")"
MsgBox ParameterUpdate
MainScreen.oDataGroupBook.BeginTransaction
Set CheckUpdate = MainScreen.oDataGroupBook.ExecuteSQLCommand(ParameterUpdate)
MainScreen.oDataGroupBook.CommitTransaction
MsgBox " Update ข้อมูลเรียบร้อยแล้ว", , "Update"
MainScreen.oDataGroupBook.DeleteVirtualTable ("CheckUpdate")
End Sub

```

```

Function PrepareparameterUpdate() As String

```

```

Dim U_SQL As String

```

```

Dim Part As String

```

```

Dim i As Byte

```

```

If cmbUPart.ListIndex = 0 Then

```

```

    Part = "L"

```

```

ElseIf cmbUPart.ListIndex = 1 Then

```

```

    Part = "M"

```

```

ElseIf cmbUPart.ListIndex = 2 Then

```

```

    Part = "R"

```

```

Else: Part = "L"

```

```

End If

```

```

U_SQL = Text3(0).Text + ","""

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

For i = 1 To 7
    U_SQL = U_SQL + UCase(Text3(i).Text)
    If i <> 7 Then
        U_SQL = U_SQL + "',''"
    End If
Next i
If Text3(9).Text = "" Then
    Text3(9).Text = "NOT"
End If
If Text3(10).Text = "" Then
    Text3(10).Text = "NOT"
End If
U_SQL = U_SQL + "',''" + Text3(8).Text + "',''" + Text3(9).Text + "',''" + Text3(10).Text +
"'',''" + _cmbUClass + "',''" + cmbUShelf + "','" + cmbULayer + "','" + Part + "',''"
PrepareparameterUpdate = U_SQL
End Function

Private Sub Form_Load()
    Dim i As Byte
    Dim CountShelf As ddoTable
    Dim CountLayer As Byte
    For i = 0 To Frame1.Count - 1
        With Frame1(i)
            .Move TabBook.ClientLeft + 40, _
                TabBook.ClientTop + 40, _
                TabBook.ClientWidth - 100, _
                TabBook.ClientHeight - 100
        End With
    Next i
    Frame1(0).ZOrder 0
    MainScreen.oDataGroupBook.BeginTransaction
    Set CountRowFirst = MainScreen.oDataGroupBook.ExecuteSQLCommand("Select count(*) from
book")

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Set CountShelf = mainScreen.oDataGroupBook.ExecuteSqlCommand("Select shelfid,layer from
bookshelf")
MainScreen.oDataGroupBook.CommitTransaction
txtfCount.Text = CountRowFirst.Columns(1).Value
cmbIShelf.Text = CountShelf.Columns(1).Value
cmbUShelf.Text = CountShelf.Columns(1).Value
CountLayer = CountShelf.Columns(2).Value
For i = 1 To CountLayer
    cmbILayer.AddItem (i)
    cmbULayer.AddItem (i)
Next i
Do While CountShelf.EOT = False 'à¼ÒèÁçéíÁÛÁàçéÖä»ä' cmbIShelf áÁD cmbILayer
    cmbIShelf.AddItem (CountShelf.Columns(1).Value)
    cmbUShelf.AddItem (CountShelf.Columns(1).Value)
    CountShelf.NextRecord
Loop
MainScreen.oDataGroupBook.DeleteVirtual Table ("CountRowFirst")
MainScreen.oDataGroupBook.DeleteVirtual Table ("CountShelf")
End Sub

Private Sub TabBook_Click()
Frame1(TabBook.SelectedItem.Index - 1).ZOrder 0
If TabBook.Tabs(3).Selected Then
    cmdSDelete.Enabled = False
    cmdSUpdate.Enabled = False
    cmdUUpdate.Enabled = False
End If
End Sub

Private Sub TDBGridS1_UnboundReadData(EyVal RowBuf As TrueDBGridStd.RowBuffer,
StartLocation As Variant, ByVal ReadPriorRows As Boolean)
    Dim i, j, r As Integer
    Dim BOT As Boolean, EOT As Boolean
    If oVTableResult Is Nothing Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RowBuf.RowCount = 0
Exit Sub
End If
If IsNull(StartLocation) Then
  If ReadPriorRows Then
    oVTableResult.LastRecord
    BOT = False
    EOT = False
  Else
    oVTableResult.FirstRecord
    BOT = False
    EOT = False
  End If
Else
  oVTableResult.GotoRecord (StartLocation)
  If ReadPriorRows Then
    If oVTableResult.PreviousRecord Then
      BOT = False
      EOT = False
    Else
      BOT = True
      EOT = False
    End If
  Else
    If oVTableResult.NextRecord Then
      EOT = False
      BOT = False
    Else
      EOT = True
      BOT = False
    End If
  End If
End If
End If
For i = 0 To RowBuf.RowCount - 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If BOT Or EOT Then
    Exit For
End If
For j = 0 To oVTableResult.Columns.Count - 1
    RowBuf.Value(i, j) = oVTableResult.Columns(j + 1).Value
Next j
RowBuf.Bookmark(i) = oVTableResult.CurrentRecordNumber
If ReadPriorRows Then
    If oVTableResult.PreviousRecord Then
        BOT = False
    Else
        BOT = True
    End If
Else
    If oVTableResult.NextRecord Then
        EOT = False
    Else
        EOT = True
    End If
End If
r = r + 1
Next i
oVTableResult.FirstRecord
RowBuf.RowCount = r
End Sub

```

```

Private Sub ClearGrid()
    Dim i As Integer
    Dim rc As Integer
    rc = 0
    For i = TDBGridS1.Columns.Count To 0 Step -1
        If i = 0 Then Exit For
        TDBGridS1.Columns.Remove (i - 1)
    Next i

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If oVTableResult Is Nothing Then      'No data return
    MsgBox "No Data in this Multimedia database", , "No Data"
    mainScreen.oDataGroupBook.DeleteVirtualTable ("oVTableResult")
    cmdSDelete.Enabled = False
    cmdSUpdate.Enabled = False
    cmdUUpdate.Enabled = False
Else: Call SetNameColumn
    cmdSDelete.Enabled = True
    cmdSUpdate.Enabled = True
    cmdUUpdate.Enabled = True
End If
End Sub

```

```

Private Sub SetNameColumn()
    Dim i As Integer
    Dim c As Object
    Dim ColumnName(12) As String
    ColumnName(0) = "BOOKID"
    ColumnName(1) = "BOOKCODE"
    ColumnName(2) = "BOOKNAME"
    ColumnName(3) = "AUTHOR1"
    ColumnName(4) = "AUTHOR2"
    ColumnName(5) = "AUTHOR3"
    ColumnName(6) = "ISBN"
    ColumnName(7) = "PUBLISHING"
    ColumnName(8) = "YEAR"
    ColumnName(9) = "BOOKCLASS"
    ColumnName(10) = "SHELFID"
    ColumnName(11) = "LAYER"
    ColumnName(12) = "PART"
    For i = 0 To oVTableResult.Columns.Count - 1
        Set c = TDBGridS1.Columns.Add(i)
        With c
            .Visible = True

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        .Caption = UCase(ColumnNme(i))
        .Locked = True
    End With
Next i
End Sub

```

```
Private Sub Text1_Change(Index As Integer)
```

```
Dim CheckInteger As String
```

```
If Index = 0 Then
```

```
    CheckInteger = Val(Text1(0).Text)
```

```
    If CheckInteger = Text1(0).Text Then
```

```
        cmdSSearch.Enabled = True
```

```
    Else: cmdSSearch.Enabled = False
```

```
    End If
```

```
End If
```

```
End Sub
```

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

```
Dim CheckInteger As String
```

```
If Index = 0 Then
```

```
    CheckInteger = Val(Text3(0).Text)
```

```
    If CheckInteger = Text3(0).Text Then
```

```
        cmdUSearch.Enabled = True
```

```
    Else: cmdUSearch.Enabled = False
```

```
    End If
```

```
ElseIf Index = 8 Then
```

```
    CheckInteger = Val(Text3(8).Text)
```

```
    If CheckInteger = Text3(8).Text Then
```

```
        cmdUUpdate.Enabled = True
```

```
    Else: cmdUUpdate.Enabled = False
```

```
    End If
```

```
End If
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub txtITitle_Change()
```

```
  If checkInsert Then
```

```
    cmdInsert.Enabled = True
```

```
  End If
```

```
End Sub
```

```
Private Sub txtIYear_Change()
```

```
  Dim CheckInteger As String
```

```
  CheckInteger = Val(txtIYear.Text)
```

```
  If CheckInteger = txtIYear.Text Then
```

```
    checkInsert = True
```

```
  Else: checkInsert = False
```

```
  End If
```

```
End Sub
```

```
-----:
```

2.3 DetailBook.frm

```
Dim FileImage As String
```

```
Public Filepreface As String
```

```
Private Sub Command1_Click(Index As Integer)
```

```
  If Index = 0 Then
```

```
    mainScreen.oDataGroupBook.DeleteVirtualTable ("oVTableDetail")
```

```
  End If
```

```
  DetailBook.Hide
```

```
  Load ShowLibrary
```

```
  ShowLibrary.Show
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
  Unload Me
```

```
  Result.Show
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
  For a% = 0 To 12
```

```
    If (Result.oVTableDetail.Columns(a% + 1).Value = "") Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Label2(a%).Caption = "ไม่มีข้อมูล"
Else:
Label2(a%).Caption = Result.oVTableDetail.Columns(a% + 1).Value
End If
Next a%
FileImage = Mid(Result.oVTableDetail.Columns(a% + 1).Value, 2, 11)
If Result.oVTableDetail.Columns(18).Value <> "" Then
Filepreface = Mid(Result.oVTableDetail.Columns(18).Value, 2, 11)
End If
Image1.Picture = LoadPicture(FileImage)
End Sub

```

```

Private Sub Form_Unload(Cancel As Integer)
Result.StatusBar1.SimpleText = "กำลังรายงานผล"
FileSystem.Kill (FileImage)
If Result.oVTableDetail.Columns(18).Value <> "" Then
FileSystem.Kill (Filepreface)
End If
End Sub

```

```

Private Sub Image1_Click()
If Result.oVTableDetail.Columns(18).Value <> "" Then
Me.Hide
Load Showpreface
Showpreface.Show
Else: MsgBox "ไม่มีข้อมูลหน้า Preface", , "ไม่มีข้อมูล"
End If
End Sub

```

```

:-----:

```

2.4 DetailMovie.frm

```

Dim FileMovie As String
Private Sub Command1_Click(Index As Integer)
If Index = 0 Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MainScreen.oDataGroupMusic.DeleteVirtualTable ("oVTableDetail")
```

```
End If
```

```
DetailMovie.Hide
```

```
Load ShowMovie
```

```
ShowMovie.Show
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
Unload Me
```

```
Result.Show
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
For a% = 0 To 10
```

```
If (Result.oVTableDetail.Columns(a% + 1).Value = "") Then
```

```
Label2(a%).Caption = "ไม่มีข้อมูล"
```

```
Else: Label2(a%).Caption = Result.oVTableDetail.Columns(a% + 1).Value
```

```
End If
```

```
Next a%
```

```
FileMovie = Mid(Result.oVTableDetail.Columns(a% + 1).Value, 2, 11)
```

```
MediaPlayer1.FileName = FileMovie
```

```
End Sub
```

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

```
MediaPlayer1.FileName = ""
```

```
FileSystem.Kill (FileMovie)
```

```
Result.StatusBar1.SimpleText = "กำลังรายงานผล"
```

```
End Sub
```

```
:-----:
```

2.5 DetailSong.frm

```
Dim FileMusic As String
```

```
Private Sub Command1_Click(Index As Integer)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If Index = 0 Then
MainScreen.oDataGroupMusic.DeleteVirtualTable ("oVTableDetail")
End If
DetailSong.Hide
Load ShowMusic
ShowMusic.Show
End Sub

```

```

Private Sub Command2_Click()
Unload Me
Result.Show
End Sub

```

```

Private Sub Form_Load()
For a% = 0 To 10
If (Result.oVTableDetail.Columns(a% + 1).Value = "") Then
Label2(a%).Caption = "ไม่มีข้อมูล"
Else: Label2(a%).Caption = Result.oVTableDetail.Columns(a% + 1).Value
End If
Next a%
FileMusic = Mid(Result.oVTableDetail.Columns(a% + 1).Value, 2, 11)
MediaPlayer1.FileName = FileMusic
End Sub

```

```

Private Sub Form_Unload(Cancel As Integer)
Result.StatusBar1.SimpleText = "กำลังรายงานผล"
MediaPlayer1.FileName = ""
FileSystem.Kill (FileMusic)
End Sub

```

```

:-----:

```

2.6 Login.frm

```
Option Explicit
```

```
Public LoginSucceeded As Boolean
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub cmdCancel_Click()
```

```
    LoginSucceeded = False
```

```
    Me.Hide
```

```
End Sub
```

```
Private Sub cmdOK_Click()
```

```
    If txtUserName = "Admin" And txtPassword = "informix" Then
```

```
        LoginSucceeded = True
```

```
        Unload Me
```

```
        Unload MainScreen
```

```
        MenuAdmin.Show
```

```
    Else
```

```
        MsgBox "Invalid Password, try again!", , "Login"
```

```
        txtPassword.SetFocus
```

```
        SendKeys "{Home}+{End}"
```

```
    End If
```

```
End Sub
```

```
-----:
```

2.7 MainScreen.frm

```
Dim oEngine1 As ddoEngine
```

```
Dim oProject1 As ddoProject
```

```
Dim oModel1 As ddoModel
```

```
Public oDataGroupBook As ddoDataGroup
```

```
Public oDataGroupMusic As ddoDataGroup
```

```
Public oDataGroupMovie As ddoDataGroup
```

```
Const PathModel As String = "E:\Project\ProjectVB\VirtualLibrary\Model\VirtualLibraryN.mlt"
```

```
Private Sub Administrator_Click()
```

```
    Login.Show vbModal
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Form_Load()
Set oEngine1 = CreateObject("DataDirector.Engine")
Set oEngine1 = New ddoEngine
Set oProject1 = oEngine1.CreateProject
oProject1.Name = "Virtual Library2.0"
Set oModel1 = oEngine1.CreateModel(PathModel)
Set oDataGroupBook = oProject1.CreateDataGroup("BOOK", "BOOK1", PathModel)
Set oDataGroupMusic = oProject1.CreateDataGroup("MUSIC", "MUSIC1", PathModel)
Set oDataGroupMovie = oProject1.CreateDataGroup("MOVIE", "MOVIE1", PathModel)
End Sub

```

```

Private Sub User_Click()
Unload MainScreen
MenuClient.Show
End Sub

```

```

:-----:

```

2.8 MenuAdmin.frm

```

Public Type Search As Byte ' 0 = SearchBook 1 = SearchSong 2 = SearchMovie

```

```

Private Sub cmdAbout_Click()

```

```

Load About

```

```

About.Show vbModal

```

```

End Sub

```

```

Private Sub Command1_Click(Index As Integer)

```

```

Command1(Index).BackColor = &H80FF80

```

```

Select Case Index

```

```

Case 0

```

```

If Not (MainScreen.oDataGroupBook.LoggedOn) Then

```

```

MainScreen.oDataGroupBook.Logon "informix", "informix"

```

```

End If

```

```

Book.Show

```

```

Me.Hide

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Case 1

```

If Not (MainScreen.oDataGroupMusic.LoggedOn) Then
    MainScreen.oDataGroupMusic.Logon "informix", "informix"
End If
Music.Show
Me.Hide

```

Case 2

```

If Not (MainScreen.oDataGroupMovie.LoggedOn) Then
    MainScreen.oDataGroupMovie.Logon "informix", "informix"
End If
Movie.Show
Me.Hide

```

Case 3

```

If Not (MainScreen.oDataGroupBook.LoggedOn) Then
    MainScreen.oDataGroupBook.Logon "informix", "informix"
End If
Shelf_Room.Show
Me.Hide
End Select
End Sub

Private Sub Command1_MouseMove(Index As Integer, Button As Integer, Shift As Integer, X As
Single, Y As Single)
If Command1(Index).BackColor <> &HC000& Then
Command1(Index).BackColor = &HC000&
End If
End Sub

Private Sub Command2_Click()
If MsgBox("ท่านแน่ใจที่จะออกจากโปรแกรมหรือไม่?", 52, "โปรดยืนยัน") = vbYes Then
If (MainScreen.oDataGroupBook.LoggedOn) Then
MainScreen.oDataGroupBook.Logoff
End If
If (MainScreen.oDataGroupMusic.LoggedOn) Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    MainScreen.oDataGroupMusic.Logoff
End If
If (MainScreen.oDataGroupMovie.LoggedOn) Then
    MainScreen.oDataGroupMovie.Logoff
End If
End
End If
End Sub

Private Sub Frame1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
For i% = 0 To 3
    If Command1(i%).BackColor <> &H80FF80 Then
        Command1(i%).BackColor = &H80FF80
        Exit Sub
    End If
Next i%
End Sub

```

2.9 MenuClient.frm

```
Public TypeSearch As Byte ' 0 = SearchBook 1 = SearchSong 2 = SearchMovie
```

```
Private Sub Command1_Click(Index As Integer)
```

```
Command1(Index).BackColor = &H80FF80
```

```
Select Case Index
```

```
Case 0
```

```
TypeSearch = 0 'SearchBook
```

```
If Not (MainScreen.oDataGroupBook.LoggedOn) Then
```

```
    MainScreen.oDataGroupBook.Logon "informix", "informix"
```

```
End If
```

```
SearchBook.Show
```

```
Me.Hide
```

```
Case 1
```

```
TypeSearch = 1 'SearchSong
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If Not (MainScreen.oDataGroupMusic.LoggedOn) Then
    MainScreen.oDataGroupMusic.Logon "informix", "informix"
End If

SearchSong.Show

Me.Hide

Case 2

TypeSearch = 2      'SearchMovie

If Not (MainScreen.oDataGroupMovie.LoggedOn) Then
    MainScreen.oDataGroupMovie.Logon "informix", "informix"
End If

Searchmovie.Show

Me.Hide

End Select

End Sub

Private Sub Command1_MouseMove(Index As Integer, Button As Integer, Shift As Integer, X As
Single, Y As Single)
If Command1(Index).BackColor <> &HC000& Then
Command1(Index).BackColor = &HC000&
End If
End Sub

Private Sub Command2_Click()
If MsgBox("ท่านแน่ใจที่จะออกจากโปรแกรมหรือไม่?", 52, "โปรดยืนยัน") = vbYes Then
    If (MainScreen.oDataGroupBook.LoggedOn) Then
        MainScreen.oDataGroupBook.Logoff
    End If
    If (MainScreen.oDataGroupMusic.LoggedOn) Then
        MainScreen.oDataGroupMusic.Logoff
    End If
    If (MainScreen.oDataGroupMovie.LoggedOn) Then
        MainScreen.oDataGroupMovie.Logoff
    End If
End

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End If
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
    Load About
```

```
    About.Show
```

```
End Sub
```

```
Private Sub Frame1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
For i% = 0 To 2
```

```
    If Command1(i%).BackColor <> &H80FF80 Then
```

```
        Command1(i%).BackColor = &H80FF80
```

```
    Exit Sub
```

```
End If
```

```
Next i%
```

```
End Sub
```

```
-----
```

2.10 Movie.frm

```
Option Explicit
```

```
Dim oVTableResult As ddoTable
```

```
Dim checkInsert As Boolean
```

```
Dim CountRowFirst As ddoTable
```

```
Private Sub Check1_Click(Index As Integer)
```

```
    If Check1(Index).Value = 1 Then
```

```
        If Index < 6 Then
```

```
            Text3(Index + 1).Enabled = True
```

```
            Text3(Index + 1).SetFocus
```

```
        Else: Select Case Index
```

```
            Case 6: cmdUBrowser.Enabled = True
```

```
            Case 7: cmbUClass.Enabled = True
```

```
            Case 8: cmbUShelf.Enabled = True
```

```
            Case 9: cmbULayer.Enabled = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Case 10: cmbUPart.Enabled = True
    End Select
End If
Else:
    If Index < 6 Then
        Text3(Index + 1).Enabled = False
    Else: Select Case Index
        Case 6: cmdUBrowser.Enabled = False
        Case 7: cmbUClass.Enabled = False
        Case 8: cmbUShelf.Enabled = False
        Case 9: cmbULayer.Enabled = False
        Case 10: cmbUPart.Enabled = False
    End Select
End If
End If
End Sub

Private Sub Check2_Click(Index As Integer)
If Check2(Index).Value = 1 Then
    If Index <> 6 Then
        With Text1(Index)
            .Text = ""
            .Visible = True
            .SetFocus
        End With
    Else: cmbSClass.Visible = True
End If
Else:
    If Index <> 6 Then
        Text1(Index).Visible = False
        Text1(Index).Text = ""
    Else: cmbSClass.Visible = False
End If
End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

```
Private Sub cmbIShelf_Click()
```

```
Dim i As Integer
```

```
Dim CountLayer As ddoTable
```

```
MainScreen.oDataGroupMovie.BeginTransaction
```

```
Set CountLayer = MainScreen.oDataGroupMovie.ExecuteSQLCommand("Select layer from movieshelf  
where shelfid = " + cmbIShelf)
```

```
MainScreen.oDataGroupMovie.CommitTransaction
```

```
cmbILayer.Clear
```

```
cmbILayer.Text = 1
```

```
For i = 1 To CountLayer.Columns(1).Value
```

```
    cmbILayer.AddItem (i)
```

```
Next i
```

```
End Sub
```

```
Private Sub cmbUShelf_Click()
```

```
Dim i As Integer
```

```
Dim CountLayer As ddoTable
```

```
MainScreen.oDataGroupMovie.BeginTransaction
```

```
Set CountLayer = MainScreen.oDataGroupMovie.ExecuteSQLCommand("Select layer from movieshelf  
where shelfid = " + cmbIShelf)
```

```
MainScreen.oDataGroupMovie.CommitTransaction
```

```
cmbULayer.Clear
```

```
cmbULayer.Text = 1
```

```
For i = 1 To CountLayer.Columns(1).Value
```

```
    cmbULayer.AddItem (i)
```

```
Next i
```

```
End Sub
```

```
Private Sub cmdIBrower_Click()
```

```
DialogInsert.ShowOpen
```

```
DialogInsert.Filter = "All type of movie"
```

```
txIFilm.Text = DialogInsert.FileName
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End Sub

Private Sub cmdInsert_Click()
Dim Parameter As String
Dim CountRow As ddoTable
Parameter = PrepareParameter "Set String ๔Í§ Parameter ๔Í§๓ÓÉÑè§ Insert Book
Parameter = "{CALL insertmovie(" + Parameter + ")}"
MainScreen.oDataGroupMovie.BeginTransaction
Set CountRow = MainScreen.oDataGroupMovie.ExecuteSqlCommand(Parameter)
MainScreen.oDataGroupMovie.CommitTransaction
If CountRow.Columns(2).Value = "N" Then
    MsgBox ("รหัสภาพยนตร์ที่คุณ Insert ซ้ำกับข้อมูลที่มีอยู่แล้ว")
Else: MsgBox (" Insert ข้อมูลเรียบร้อยแล้ว")
End If
txtICount.Text = CountRow.Columns(1).Value
MainScreen.oDataGroupBook.DeleteVirtualTable ("CountRow")
End Sub

Private Function PrepareParameter() As String
Dim P_SQL As String
Dim Part As String
If cmbIPart.ListIndex = 0 Then
    Part = "L"
ElseIf cmbIPart.ListIndex = 1 Then
    Part = "M"
ElseIf cmbIPart.ListIndex = 2 Then
    Part = "R"
Else: Part = "L"
End If
P_SQL = """" + UCase(txtICode.Text) + """, """" + UCase(txtIName.Text) + """, """" + UCase
(txtIActor1.Text) + _ """, """" + UCase(txtIActor2.Text) + """, """" + UCase(cmbIClass) + """, """" + UCase
(txtIPublishing.Text) + _ """, "" + (txtIYear.Text) + ", """" + txtIFilm.Text + """, "" + cmbIShelf.Text + ", " +
cmbILayer + ", """" + Part + """"
PrepareParameter = P_SQL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Function
```

```
Private Sub cmdSClear_Click()
```

```
Dim i As Byte
```

```
For i = 0 To 5
```

```
    Check2(i).Value = 0
```

```
    Text1(i).Text = ""
```

```
    Text1(i).Visible = False
```

```
Next i
```

```
    Check2(6).Value = 0
```

```
    cmbSClass.Visible = False
```

```
End Sub
```

```
Private Sub cmdSDelete_Click()
```

```
Dim DeleteSQL As String
```

```
oVTableResult.GotoRecord (TDBGridS1.Row + 1)
```

```
DeleteSQL = "delete from movie where movieid = " + oVTableResult.Columns(1).Value
```

```
MainScreen.oDataGroupMovie.BeginTransaction
```

```
MainScreen.oDataGroupMovie.ExecuteSQLCommand (DeleteSQL)
```

```
Set CountRowFirst = MainScreen.oDataGroupMovie.ExecuteSQLCommand("Select count(*) from  
movie")
```

```
MainScreen.oDataGroupMovie.CommitTransaction
```

```
txtCount.Text = CountRowFirst.Columns(1).Value
```

```
TDBGridS1.Refresh
```

```
Call PassSQL
```

```
End Sub
```

```
Private Sub cmdSSearch_Click()
```

```
Dim check As Boolean
```

```
Dim i As Byte
```

```
check = True
```

```
For i = 0 To 4
```

```
    If (Check2(i) = 1) And (Text1(i).Text = "") Then
```

```
        check = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    End If
Next i
If check Then
Call PassSQL
Else: MsgBox ("โปรดใส่ข้อความที่คุณต้องการค้นหา หัวข้อที่คุณเช็ค")
End If
End Sub

```

```

Private Sub PassSQL()
Dim SQLString As String
SQLString = PrepareParameterSearch
SQLString = "{CALL searchdetailmovie(" + SQLString + ")}"
MainScreen.oDataGroupMovie.BeginTransaction
Set oVTableResult = MainScreen.oDataGroupMovie.ExecuteSQLCommand(SQLString)
MainScreen.oDataGroupMovie.CommitTransaction
Call ClearGrid
End Sub

```

```

Private Function PrepareParameterSearch() As String

```

```

Dim i As Byte

```

```

Dim SQL As String

```

```

If Check2(0).Value = 0 Then

```

```

    Text1(0).Text = "0"

```

```

End If

```

```

If Check2(5).Value = 0 Then

```

```

    Text1(5).Text = "0"

```

```

End If

```

```

For i = 0 To 4

```

```

    SQL = SQL + UCase(Text1(i).Text)

```

```

    If i = 0 Then

```

```

        SQL = Text1(i).Text + ","

```

```

    ElseIf i <> 4 Then

```

```

        SQL = SQL + "" & Text1(i).Text & ","

```

```

    End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Next i
SQL = SQL + """," + Text1(5).Text + ","
If Check2(6).Value = 0 Then
    SQL = SQL + """"""
Else: SQL = SQL + """" + cmbSClass + """"
End If
PrepareParameterSearch = SQL
End Function

```

```

Private Sub ClearGrid()
    Dim i As Integer
    Dim rc As Integer
    rc = 0
    For i = TDBGridS1.Columns.Count To 0 Step -1
        If i = 0 Then Exit For
        TDBGridS1.Columns.Remove (i - 1)
    Next i
    If oVTableResult Is Nothing Then
        'No data return
        MsgBox ("No Data in this Multimedia database")
        mainScreen.oDataGroupMovie.DeleteVirtualTable ("oVTableResult")
        cmdSDelete.Enabled = False
        cmdSUpdate.Enabled = False
        cmdUUpdate.Enabled = False
    Else: Call SetNameColumn
        cmdSDelete.Enabled = True
        cmdSUpdate.Enabled = True
        cmdUUpdate.Enabled = True
    End If
End Sub

```

```

Private Sub SetNameColumn()
    Dim i As Integer
    Dim c As Object
    Dim ColumnName(10) As String

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ColumnName(0) = "MOVIEID"
ColumnName(1) = "MOVIECODE"
ColumnName(2) = "MOVIEName"
ColumnName(3) = "ACTOR 1"
ColumnName(4) = "ACTOR 2"
ColumnName(5) = "PUBLISHING"
ColumnName(6) = "YEAR"
ColumnName(7) = "MOVIECLASS"
ColumnName(8) = "SHELFID"
ColumnName(9) = "LAYER"
ColumnName(10) = "PART"
For i = 0 To oVTableResult.Columns.Count - 1
    Set c = TDBGridS1.Columns.Add(i)
    With c
        .Visible = True
        .Caption = UCase(ColumnName(i))
        .Locked = True
    End With
Next i
End Sub

Private Sub cmdSUpdate_Click()
Dim Search_Update As String
TabMovie.Tabs(2).Selected = True
Frame1(1).ZOrder 0
oVTableResult.GotoRecord (TDBGridS1.Row + 1)
Search_Update = "{Call SearchUpdateMovie(" + oVTableResult.Columns(1).Value + ")}"
Text3(0).Text = oVTableResult.Columns(1).Value
MainScreen.oDataGroupMovie.BeginTransaction
Set oVTableResult = MainScreen.oDataGroupMovie.ExecuteSQLCommand(Search_Update)
MainScreen.oDataGroupMovie.CommitTransaction
Call Input_to_txt
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub cmdUBrowser_Click()
DialogUpdate.ShowOpen
DialogUpdate.Filter = "All type of movie"
Text3(7).Text = DialogUpdate.FileName
End Sub

```

```

Private Sub cmdUSearch_Click()
Dim Search_Update As String
Search_Update = "{Call SearchUpdateMovie(" + Text3(0).Text + ")}"
MainScreen.oDataGroupMovie.BeginTransaction
Set oVTableResult = MainScreen.oDataGroupMovie.ExecuteSQLCommand(Search_Update)
MainScreen.oDataGroupMovie.CommitTransaction
Call Input_to_txt
End Sub

```

```

Private Sub Input_to_txt()
Dim i As Integer
If Not (oVTableResult Is Nothing) Then
For i = 1 To 6
Text3(i).Text = oVTableResult.Columns(i).Value
Next i
cmbUClass.Text = oVTableResult.Columns(7).Value
cmbUShelf.Text = oVTableResult.Columns(8).Value
cmbULayer.Clear
cmbULayer.Text = oVTableResult.Columns(9).Value
If oVTableResult.Columns(10).Value = "L" Then
cmbUPart.Text = "ซ้าย"
ElseIf oVTableResult.Columns(10).Value = "M" Then
cmbUPart.Text = "กลาง"
ElseIf oVTableResult.Columns(10).Value = "R" Then
cmbUPart.Text = "ขวา"
End If
For i = 1 To oVTableResult.Columns(11).Value
cmbULayer.AddItem (i)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Next i
cmdUUpdate.Enabled = True
Else: MsgBox ("No Data in this Multimedia database")
    Text3(0).Text = ""
cmdUUpdate.Enabled = False
End If
MainScreen.oDataGroupMovie.DeleteVirtualTable ("oVTableResult")
End Sub

Function PrepareparameterUpdate() As String
Dim U_SQL As String
Dim Part As String
Dim i As Byte
If cmbUPart.ListIndex = 0 Then
    Part = "L"
ElseIf cmbUPart.ListIndex = 1 Then
    Part = "M"
ElseIf cmbUPart.ListIndex = 2 Then
    Part = "R"
Else: Part = "L"
End If
U_SQL = Text3(0).Text + ","
For i = 1 To 5
    U_SQL = U_SQL + UCase(Text3(i).Text)
    If i <> 5 Then
        U_SQL = U_SQL + " , "
    End If
Next i
If Text3(7).Text = "" Then
    Text3(7).Text = "NOT"
End If
U_SQL = U_SQL + "" , " + Text3(6).Text + " , "" + Text3(7).Text + "" , "" + _
    cmbUClass + "" , " + cmbUShelf + " , " + cmbULayer + " , "" + Part + "" ""
PrepareparameterUpdate = U_SQL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Function

```
Private Sub cmdUUpdate_Click()
```

```
Dim ParameterUpdate As String
```

```
Dim CheckUpdate As ddoTable
```

```
ParameterUpdate = PrepareparameterUpdate
```

```
ParameterUpdate = "execute procedure UPDATEMOVIE(" + ParameterUpdate + ")"
```

```
MainScreen.oDataGroupMovie.BeginTransaction
```

```
Set CheckUpdate = MainScreen.oDataGroupMovie.ExecuteSqlCommand(ParameterUpdate)
```

```
MainScreen.oDataGroupMovie.CommitTransaction
```

```
MsgBox (" Update ข้อมูลเรียบร้อยแล้ว")
```

```
MainScreen.oDataGroupMovie.DeleteVirtualTable ("CheckUpdate")
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
Unload Me
```

```
MenuAdmin.Show
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Dim i As Byte
```

```
Dim CountShelf As ddoTable
```

```
Dim CountLayer As Byte
```

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

```
With Frame1(i)
```

```
.Move TabMovie.ClientLeft + 40, _
```

```
TabMovie.ClientTop + 40, _
```

```
TabMovie.ClientWidth - 100, _
```

```
TabMovie.ClientHeight - 100
```

```
End With
```

```
Next i
```

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

```
MainScreen.oDataGroupMovie.BeginTransaction
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Set CountRowFirst = mainScreen.oDataGroupMovie.ExecuteSQLCommand("Select count(*) from
movie")
Set CountShelf = mainScreen.oDataGroupMovie.ExecuteSQLCommand("Select shelfid,layer from
moviesshelf")
MainScreen.oDataGroupMovie.CommitTransaction
txtICount.Text = CountRowFirst.Columns(1).Value
cmbIShelf.Text = CountShelf.Columns(1).Value
cmbUShelf.Text = CountShelf.Columns(1).Value
CountLayer = CountShelf.Columns(2).Value
For i = 1 To CountLayer
    cmbILayer.AddItem (i)
    cmbULayer.AddItem (i)
Next i
Do While CountShelf.EOT = False
    cmbIShelf.AddItem (CountShelf.Columns(1).Value)
    cmbUShelf.AddItem (CountShelf.Columns(1).Value)
    CountShelf.NextRecord
Loop
MainScreen.oDataGroupMovie.DeleteVirtualTable ("CountRowFirst")
MainScreen.oDataGroupMovie.DeleteVirtualTable ("CountShelf")
End Sub

Private Sub TabMovie_Click()
Frame1(TabMovie.SelectedItem.Index - 1).ZOrder 0
If TabMovie.Tabs(3).Selected Then
    cmdSDelete.Enabled = False
    cmdSUpdate.Enabled = False
    cmdUUpdate.Enabled = False
End If
End Sub

Private Sub TDBGridS1_UnboundReadData(PyVal RowBuf As TrueDBGridStd.RowBuffer,
StartLocation As Variant, ByVal ReadPriorRows As Boolean)
    Dim i, j, r As Integer

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dim BOT As Boolean, EOT As Boolean

If oVTableResult Is Nothing Then

 RowBuf.RowCount = 0

 Exit Sub

End If

r = 0

If IsNull(StartLocation) Then

 If ReadPriorRows Then

 oVTableResult.LastRecord

 BOT = False

 EOT = False

 Else

 oVTableResult.FirstRecord

 BOT = False

 EOT = False

 End If

Else

 oVTableResult.GotoRecord (StartLocation)

 If ReadPriorRows Then

 If oVTableResult.PreviousRecord Then

 BOT = False

 EOT = False

 Else

 BOT = True

 EOT = False

 End If

 Else

 If oVTableResult.NextRecord Then

 EOT = False

 BOT = False

 Else

 EOT = True

 BOT = False

 End If

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    End If
End If
For i = 0 To RowBuf.RowCount - 1
    If BOT Or EOT Then
        Exit For
    End If
    For j = 0 To oVTableResult.Columns.Count - 1
        RowBuf.Value(i, j) = oVTableResult.Columns(j + 1).Value
    Next j
    RowBuf.Bookmark(i) = oVTableResult.CurrentRecordNumber
    If ReadPriorRows Then
        If oVTableResult.PreviousRecord Then
            BOT = False
        Else
            BOT = True
        End If
    Else
        If oVTableResult.NextRecord Then
            EOT = False
        Else
            EOT = True
        End If
    End If
    r = r + 1
Next i
oVTableResult.FirstRecord
RowBuf.RowCount = r
End Sub

```

```

Private Sub Text1_Change(Index As Integer)
    Dim CheckInteger As String
    If Index = 0 Then
        CheckInteger = Val(Text1(0).Text)
        If CheckInteger = Text1(0).Text Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        cmdSSearch.Enabled = True
    Else: cmdSSearch.Enabled = False
    End If
End If
End Sub

```

```

Private Sub Text3_Change(Index As Integer)
    Dim CheckInteger As String
    If Index = 0 Then
        CheckInteger = Val(Text3(0).Text)
        If CheckInteger = Text3(0).Text Then
            cmdUSearch.Enabled = True
        Else: cmdUSearch.Enabled = False
        End If
    ElseIf Index = 6 Then
        CheckInteger = Val(Text3(6).Text)
        If CheckInteger = Text3(6).Text Then
            cmdUUpdate.Enabled = True
        Else: cmdUUpdate.Enabled = False
        End If
    End If
End If
End Sub

```

```

Private Sub txtIFilm_Change()
    If checkInsert Then
        cmdInsert.Enabled = True
    Else: cmdInsert.Enabled = False
    End If
End Sub

```

```

Private Sub txtIYear_Change()
    Dim CheckInteger As String
    CheckInteger = Val(txtIYear.Text)
    If CheckInteger = txtIYear.Text Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    checkInsert = True
Else: checkInsert = False
End If
End Sub
:-----;

```

2.11 Music.frm

Option Explicit

Dim oVTableResult As ddoTable

Dim checkInsert As Boolean

Dim CountRowFirst As ddoTable

Private Sub Check1_Click(Index As Integer)

If Check1(Index).Value = 1 Then

 If Index < 6 Then

 Text3(Index + 1).Enabled = True

 Text3(Index + 1).SetFocus

 Else: Select Case Index

 Case 6: cmdUBrowser.Enabled = True

 Case 7: cmbUGenre.Enabled = True

 Case 8: cmbUShelf.Enabled = True

 Case 9: cmbULayer.Enabled = True

 Case 10: cmbUPart.Enabled = True

 End Select

End If

Else:

 If Index < 6 Then

 Text3(Index + 1).Enabled = False

 Else: Select Case Index

 Case 6: cmdUBrowser.Enabled = True

 Case 7: cmbUGenre.Enabled = False

 Case 8: cmbUShelf.Enabled = False

 Case 9: cmbULayer.Enabled = False

 Case 10: cmbUPart.Enabled = False

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    End Select
  End If
End If
End Sub

```

```

Private Sub Check2_Click(Index As Integer)

```

```

  If Check2(Index).Value = 1 Then

```

```

    If Index <> 7 Then

```

```

      With Text1(Index)

```

```

        .Text = ""

```

```

        .Visible = True

```

```

        .SetFocus

```

```

      End With

```

```

    Else: cmbSGenre.Visible = True

```

```

  End If

```

```

Else:

```

```

  If Index <> 7 Then

```

```

    Text1(Index).Visible = False

```

```

    Text1(Index).Text = ""

```

```

  Else: cmbSGenre.Visible = False

```

```

  End If

```

```

End If

```

```

End Sub

```

```

Private Sub cmbIShelf_Click()

```

```

  Dim i As Integer

```

```

  Dim CountLayer As ddoTable

```

```

  MainScreen.oDataGroupMusic.BeginTransaction

```

```

  Set CountLayer = MainScreen.oDataGroupMusic.ExecuteSQLCommand("Select layer from musicshelf
  where shelfid = " + cmbIShelf)

```

```

  MainScreen.oDataGroupMusic.CommitTransaction

```

```

  cmbILayer.Clear

```

```

  cmbILayer.Text = 1

```

```

  For i = 1 To CountLayer.Columns(1).Value

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    cmbILayer.AddItem (i)
Next i
End Sub

Private Sub cmbUShelf_Click()
Dim i As Integer
Dim CountLayer As ddoTable
MainScreen.oDataGroupMusic.BeginTransaction
Set CountLayer = MainScreen.oDataGroupMusic.ExecuteSqlCommand("Select layer from musicshelf
where shelfid = " + cmbIShelf)
MainScreen.oDataGroupMusic.CommitTransaction
cmbULayer.Clear
cmbULayer.Text = 1
For i = 1 To CountLayer.Columns(1).Value
    cmbULayer.AddItem (i)
Next i
End Sub

Private Sub cmdIBrower_Click()
DialogInsert.ShowOpen
DialogInsert.Filter = "All type of song"
txtISong.Text = DialogInsert.FileName
End Sub

Private Sub cmdInsert_Click()
Dim Parameter As String
Dim CountRow As ddoTable
Parameter = PrepareParameter
Parameter = "{CALL insertmusic(" + Parameter + ")}"
MainScreen.oDataGroupMusic.BeginTransaction
Set CountRow = MainScreen.oDataGroupMusic.ExecuteSqlCommand(Parameter)
MainScreen.oDataGroupMusic.CommitTransaction
If CountRow.Columns(2).Value = "N" Then
    MsgBox "รหัสเพลงที่คุณ Insert ซ้ำกับข้อมูลที่มีอยู่แล้ว", , "Insert"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Else: MsgBox (" Insert ข้อมูลเรียบร้อยแล้ว")
End If
txtICount.Text = CountRow.Columns(1).Value
MainScreen.oDataGroupBook.DeleteVirtualTable ("CountRow")
End Sub

```

```

Private Function PrepareParameter() As String

```

```

Dim P_SQL As String

```

```

Dim Part As String

```

```

If cmbIPart.ListIndex = 0 Then

```

```

    Part = "L"

```

```

Elseif cmbIPart.ListIndex = 1 Then

```

```

    Part = "M"

```

```

Elseif cmbIPart.ListIndex = 2 Then

```

```

    Part = "R"

```

```

Else: Part = "L"

```

```

End If

```

```

P_SQL = "" + UCase(txtICode.Text) + "," + UCase(txtIName.Text) + "," + UCase
(txtIArthist.Text) + _
"" + UCase(txtIAlbum.Text) + "," + UCase(cmbIGenre) + "," +
UCase(txtIPublishing.Text) + _
"" + (txtIYear.Text) + "," + txtISong.Text + "," + cmbIShelf.Text
+ "," + cmbILayer + "," + Part + ""

```

```

PrepareParameter = P_SQL

```

```

End Function

```

```

Private Sub cmdSClear_Click()

```

```

Dim i As Byte

```

```

For i = 0 To 6

```

```

    Check2(i).Value = 0

```

```

    Text1(i).Text = ""

```

```

    Text1(i).Visible = False

```

```

Next i

```

```

    Check2(7).Value = 0

```

```

    cmbSGenre.Visible = False

```

```

End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub cmdSDelete_Click()
Dim DeleteSQL As String
oVTableResult.GotoRecord (TDBGridS1.Row + 1)
DeleteSQL = "delete from music WHERE musicid = " + oVTableResult.Columns(1).Value
MainScreen.oDataGroupMusic.BeginTransaction
MainScreen.oDataGroupMusic.ExecuteSQLCommand (DeleteSQL)
Set CountRowFirst = MainScreen.oDataGroupMusic.ExecuteSQLCommand("Select count(*) from
music")
MainScreen.oDataGroupMusic.CommitTransaction
txtIFCount.Text = CountRowFirst.Columns(1).Value
TDBGridS1.Refresh
Call PassSQL
End Sub

```

```

Private Sub cmdSSearch_Click()
Dim check As Boolean
Dim i As Byte
check = True
For i = 0 To 6
If (Check2(i) = 1) And (Text1(i).Text = "") Then
check = False
End If
Next i
If check Then
Call PassSQL
Else: MsgBox "โปรดใส่ข้อความที่คุณต้องการค้นหา", , "ค้นหา"
End If
End Sub

```

```

Private Sub PassSQL()
Dim SQLString As String
SQLString = PrepareParameterSearch
SQLString = "{CALL searchdetailmusic(" + SQLString + ")}"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MainScreen.oDataGroupMusic.BeginTransaction
Set oVTableResult = MainScreen.oDataGroupMusic.ExecuteSQLCommand(SQLString)
MainScreen.oDataGroupMusic.CommitTransaction
Call ClearGrid
End Sub

```

```

Private Function PrepareParameterSearch() As String
Dim i As Byte
Dim SQL As String
If Check2(0).Value = 0 Then
    Text1(0).Text = "0"
End If
If Check2(6).Value = 0 Then
    Text1(6).Text = "0"
End If
For i = 0 To 5
    SQL = SQL + UCase(Text1(i).Text)
    If i = 0 Then
        SQL = Text1(i).Text + ","
    ElseIf i <> 5 Then
        SQL = SQL + ","
    End If
Next i
SQL = SQL + """," + Text1(6).Text + ","
If Check2(7).Value = 0 Then
    SQL = SQL + """"
Else: SQL = SQL + """" + cmbSGenre + """"
End If
PrepareParameterSearch = SQL
End Function

```

```

Private Sub ClearGrid()
Dim i As Integer
Dim rc As Integer

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

rc = 0
For i = TDBGridS1.Columns.Count To 0 Step -1
    If i = 0 Then Exit For
    TDBGridS1.Columns.Remove (i - 1)
Next i
If oVTableResult Is Nothing Then      'No data return
    MsgBox ("No Data in this Multimedia database")
    MainScreen.oDataGroupMusic.DeleteVirtualTable ("oVTableResult")
    cmdSDelete.Enabled = False
    cmdSUpdate.Enabled = False
    cmdUUpdate.Enabled = False
Else: Call SetNameColumn
    cmdSDelete.Enabled = True
    cmdSUpdate.Enabled = True
    cmdUUpdate.Enabled = True
End If
End Sub

Private Sub SetNameColumn()
    Dim i As Integer
    Dim c As Object
    Dim ColumnName(10) As String
    ColumnName(0) = "MUSICID"
    ColumnName(1) = "MUSICCODE"
    ColumnName(2) = "MUSICNAME"
    ColumnName(3) = "ARTHIST"
    ColumnName(4) = "ALBUM"
    ColumnName(5) = "PUBLISHING"
    ColumnName(6) = "YEAR"
    ColumnName(7) = "GENRE"
    ColumnName(8) = "SHELFID"
    ColumnName(9) = "LAYER"
    ColumnName(10) = "PART"
    For i = 0 To oVTableResult.Columns.Count - 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Set c = TDBGridS1.Columns.Add(i)
    With c
        .Visible = True
        .Caption = UCase(ColumnName(i))
        .Locked = True
    End With
Next i
End Sub

Private Sub cmdSUpdate_Click()
Dim Search_Update As String
TabMusic.Tabs(2).Selected = True
Frame1(1).ZOrder 0
oVTableResult.GotoRecord (TDBGridS1.Row + 1)
Search_Update = "{Call SearchUpdateMusic(" + oVTableResult.Columns(1).Value + ")}"
Text3(0).Text = oVTableResult.Columns(1).Value
MainScreen.oDataGroupMusic.BeginTransaction
Set oVTableResult = MainScreen.oDataGroupMusic.ExecuteSQLCommand(Search_Update)
MainScreen.oDataGroupMusic.CommitTransaction
Call Input_to_txt
End Sub

Private Sub Input_to_txt()
Dim i As Integer
If Not (oVTableResult Is Nothing) Then
    For i = 1 To 6
        Text3(i).Text = oVTableResult.Columns(i).Value
    Next i
    cmbUGenre.Text = oVTableResult.Columns(7).Value
    cmbUShelf.Text = oVTableResult.Columns(8).Value
    cmbULayer.Clear 'Ã©Ò§¸éÁÛÃâ' ComboBox
    cmbULayer.Text = oVTableResult.Columns(9).Value
    If oVTableResult.Columns(10).Value = "L" Then
        cmbUPart.Text = "«éÒÃ"
    End If
End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ElseIf oVTableResult.Columns(10).Value = "M" Then
    cmbUPart.Text = "เ้าÒ§"
ElseIf oVTableResult.Columns(10).Value = "R" Then
    cmbUPart.Text = "เ้าÒ"
End If
For i = 1 To oVTableResult.Columns(11).Value
    cmbULayer.AddItem (i)
Next i
cmdUUpdate.Enabled = True
Else: MsgBox ("No Data in this Multimedia database")
    Text3(0).Text = ""
    cmdUUpdate.Enabled = False
End If
MainScreen.oDataGroupMusic.DeleteVirtualTable ("oVTableResult")
End Sub

Function PrepareparameterUpdate() As String
Dim U_SQL As String
Dim Part As String
Dim i As Byte
If cmbUPart.ListIndex = 0 Then
    Part = "L"
ElseIf cmbUPart.ListIndex = 1 Then
    Part = "M"
ElseIf cmbUPart.ListIndex = 2 Then
    Part = "R"
Else: Part = "L"
End If
U_SQL = Text3(0).Text + ", """"
For i = 1 To 5
    U_SQL = U_SQL + UCase(Text3(i).Text)
    If i <> 5 Then
        U_SQL = U_SQL + """, """"
    End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Next i
If Text3(7).Text = "" Then
    Text3(7).Text = "NOT"
End If
U_SQL = U_SQL + """," + Text3(6).Text + """," + Text3(7).Text + """," + _
        cmbUGenre + """," + cmbUShelf + """," + cmbULayer + """," + Part + """"
PrepareparameterUpdate = U_SQL
End Function

```

```
Private Sub cmdUBrowser_Click()
```

```
DialogUpdate.ShowOpen
```

```
DialogUpdate.Filter = "All type of music"
```

```
Text3(7).Text = DialogUpdate.FileName
```

```
End Sub
```

```
Private Sub cmdUSearch_Click()
```

```
Dim Search_Update As String
```

```
Text3(0).Text = Val(Text3(0).Text)
```

```
Search_Update = "{Call SearchUpdateMusic(" + Text3(0).Text + ")}"
```

```
MainScreen.oDataGroupMusic.BeginTransaction
```

```
Set oVTableResult = MainScreen.oDataGroupMusic.ExecuteSQLCommand(Search_Update)
```

```
MainScreen.oDataGroupMusic.CommitTransaction
```

```
Call Input_to_txt
```

```
End Sub
```

```
Private Sub cmdUUpdate_Click()
```

```
Dim ParameterUpdate As String
```

```
Dim CheckUpdate As ddoTable
```

```
ParameterUpdate = PrepareparameterUpdate
```

```
ParameterUpdate = "execute procedure UPDITEMUSIC(" + ParameterUpdate + ")"
```

```
MainScreen.oDataGroupMusic.BeginTransaction
```

```
Set CheckUpdate = MainScreen.oDataGroupMusic.ExecuteSQLCommand(ParameterUpdate)
```

```
MainScreen.oDataGroupMusic.CommitTransaction
```

```
MsgBox " Update ข้อมูลเรียบร้อยแล้ว", , "Update"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MainScreen.oDataGroupMusic.DeleteVirtualTable ("CheckUpdate")
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
Unload Me
```

```
MenuAdmin.Show
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Dim i As Byte
```

```
Dim CountShelf As ddoTable
```

```
Dim CountLayer As Byte
```

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

```
With Frame1(i)
```

```
.Move TabMusic.ClientLeft + 40, _
```

```
TabMusic.ClientTop + 40, _
```

```
TabMusic.ClientWidth - 100, _
```

```
TabMusic.ClientHeight - 100
```

```
End With
```

```
Next i
```

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

```
MainScreen.oDataGroupMusic.BeginTransaction
```

```
Set CountRowFirst = MainScreen.oDataGroupMusic.ExecuteSQLCommand("Select count(*) from music")
```

```
Set CountShelf = MainScreen.oDataGroupMusic.ExecuteSQLCommand("Select shelfid,layer from musicshelf")
```

```
MainScreen.oDataGroupMusic.CommitTransaction
```

```
txtICount.Text = CountRowFirst.Columns(1).Value
```

```
cmbIShelf.Text = CountShelf.Columns(1).Value
```

```
cmbUShelf.Text = CountShelf.Columns(1).Value
```

```
CountLayer = CountShelf.Columns(2).Value
```

```
For i = 1 To CountLayer
```

```
cmbILayer.AddItem (i)
```

```
cmbULayer.AddItem (i)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Next i
Do While CountShelf.EOT = False
    cmbIShelf.AddItem (CountShelf.Columns(1).Value)
    cmbUShelf.AddItem (CountShelf.Columns(1).Value)
    CountShelf.NextRecord
Loop
MainScreen.oDataGroupMusic.DeleteVirtualTable ("CountRowFirst")
MainScreen.oDataGroupMusic.DeleteVirtualTable ("CountShelf")
End Sub

Private Sub TabMusic_Click()
Frame1(TabMusic.SelectedItem.Index - 1).ZOrder 0
If TabMusic.Tabs(3).Selected Then
    cmdSDelete.Enabled = False
    cmdSUpdate.Enabled = False
    cmdUUpdate.Enabled = False
End If
End Sub

Private Sub TDBGridS1_UnboundReadData(ByVal RowBuf As TrueDBGridStd.RowBuffer,
StartLocation As Variant, ByVal ReadPriorRows As Boolean)
    Dim i, j, r As Integer
    Dim BOT As Boolean, EOT As Boolean
    If oVTableResult Is Nothing Then
        RowBuf.RowCount = 0
        Exit Sub
    End If
    r = 0
    If IsNull(StartLocation) Then
        If ReadPriorRows Then
            oVTableResult.LastRecord
            BOT = False
            EOT = False
        Else

```

```

oVTableResult.FirstRecord
BOT = False
EOT = False
End If
Else
oVTableResult.GotoRecord (StartLocation)
If ReadPriorRows Then
If oVTableResult.PreviousRecord Then
BOT = False
EOT = False
Else
BOT = True
EOT = False
End If
Else
If oVTableResult.NextRecord Then
EOT = False
BOT = False
Else
EOT = True
BOT = False
End If
End If
End If
For i = 0 To RowBuf.RowCount - 1
If BOT Or EOT Then
Exit For
End If
For j = 0 To oVTableResult.Columns.Count - 1
RowBuf.Value(i, j) = oVTableResult.Columns(j + 1).Value
Next j
RowBuf.Bookmark(i) = oVTableResult.CurrentRecordNumber
If ReadPriorRows Then
If oVTableResult.PreviousRecord Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        BOT = False
    Else
        BOT = True
    End If
Else
    If oVTableResult.NextRecord Then
        EOT = False
    Else
        EOT = True
    End If
End If
    r = r + 1
Next i
oVTableResult.FirstRecord
RowBuf.RowCount = r
End Sub

```

```

Private Sub Text1_Change(Index As Integer)
    Dim CheckInteger As String
    If Index = 0 Then
        CheckInteger = Val(Text1(0).Text)
        If CheckInteger = Text1(0).Text Then
            cmdSSearch.Enabled = True
        Else: cmdSSearch.Enabled = False
        End If
    End If
End Sub

```

```

Private Sub Text3_Change(Index As Integer)
    Dim CheckInteger As String
    If Index = 0 Then
        CheckInteger = Val(Text3(0).Text)
        If CheckInteger = Text3(0).Text Then
            cmdUSearch.Enabled = True
        End If
    End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Else: cmdUSearch.Enabled = False
End If
ElseIf Index = 6 Then
    CheckInteger = Val(Text3(6).Text)
    If CheckInteger = Text3(6).Text Then
        cmdUUpdate.Enabled = True
    Else: cmdUUpdate.Enabled = False
    End If
End If
End Sub

```

```

Private Sub txtISong_Change()
    If checkInsert Then
        cmdInsert.Enabled = True
    End If
End Sub

```

```

Private Sub txtIYear_Change()
    Dim CheckInteger As String
    CheckInteger = Val(txtIYear.Text)
    If CheckInteger = txtIYear.Text Then
        checkInsert = True
    Else: checkInsert = False
    End If
End Sub

```

```

:-----:

```

2.12 Result.frm

```

Public oVTableDetail As ddoTable
Dim oVTableResult As ddoTable 'This Virtual Table for Result Form only
Dim SQLDetail As String
Dim ColumnName() As String
Const StatusStart As String = "Status : "

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Command1_Click()
Me.Hide
oVTableResult.GotoRecord (TDBGridS1.Row + 1)
StatusBar1.Panels(1).Text = StatusStart + "กำลังค้นหา... โปรดรอสักครู"
Command1.MousePointer = 11
Select Case MenuClient.TypeSearch
    Case 0
        SQLDetail = "execute function getdetailbook('"' + oVTableResult.Columns(1).Value + "'")
        MainScreen.oDataGroupBook.BeginTransaction
        Set oVTableDetail = MainScreen.oDataGroupBook.ExecuteSQLCommand(SQLDetail)
        MainScreen.oDataGroupBook.CommitTransaction
        Load DetailBook
        DetailBook.Show vbModal
    Case 1
        SQLDetail = "execute function getdetailmusic('"' + oVTableResult.Columns(1).Value + "'")
        MainScreen.oDataGroupMusic.BeginTransaction
        Set oVTableDetail = MainScreen.oDataGroupMusic.ExecuteSQLCommand(SQLDetail)
        MainScreen.oDataGroupMusic.CommitTransaction
        Load DetailSong
        DetailSong.Show vbModal
    Case 2
        SQLDetail = "execute function getdetailmovie('"' + oVTableResult.Columns(1).Value + "'")
        MainScreen.oDataGroupMovie.BeginTransaction
        Set oVTableDetail = MainScreen.oDataGroupMovie.ExecuteSQLCommand(SQLDetail)
        MainScreen.oDataGroupMovie.CommitTransaction
        Load DetailMovie
        DetailMovie.Show vbModal
End Select
End Sub

```

```

Private Sub Command2_Click()
Unload Me
Select Case MenuClient.TypeSearch
    Case 0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SearchBook.Show
```

```
MainScreen.oDataGroupBook.DeleteVirtualTable ("oVTableBook")
```

```
Case 1
```

```
SearchSong.Show
```

```
MainScreen.oDataGroupMusic.DeleteVirtualTable ("oVTableMusic")
```

```
Case 2
```

```
Searchmovie.Show
```

```
MainScreen.oDataGroupMovie.DeleteVirtualTable ("oVTableMovie")
```

```
End Select
```

```
End Sub
```

```
Private Sub Form_Activate()
```

```
StatusBar1.Panels(1).Text = StatusStart + "แสดงผลการค้นหา"
```

```
Command1.MousePointer = 0
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
StatusBar1.Panels(1).Text = StatusStart
```

```
Select Case MenuClient.TypeSearch
```

```
Case 0
```

```
Set oVTableResult = SearchBook.oVTableBook
```

```
ReDim ColumnName(4)
```

```
ColumnName(0) = "BookCode"
```

```
ColumnName(1) = "BookName"
```

```
ColumnName(2) = "BookCatalog"
```

```
ColumnName(3) = "Publishing"
```

```
ColumnName(4) = "Year"
```

```
Case 1
```

```
Set oVTableResult = SearchSong.oVTableMusic
```

```
ReDim ColumnName(6)
```

```
ColumnName(0) = "SongCode"
```

```
ColumnName(1) = "SongName"
```

```
ColumnName(2) = "Arthist"
```

```
ColumnName(3) = "Album"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ColumnName(4) = "Year"
ColumnName(5) = "Genre"
ColumnName(6) = "Publishing"

```

Case 2

```

Set oVTableResult = Searchmovie.oVTableMovie
ReDim ColumnName(4)
ColumnName(0) = "MovieCode"
ColumnName(1) = "MovieName"
ColumnName(2) = "Category"
ColumnName(3) = "Year"
ColumnName(4) = "Publishing"

```

End Select

Call ClearGrid

Label2.Caption = oVTableResult.ResultSetCount

Label2.Caption = oVTableResult.TotalRecordCount

End Sub

```

Private Sub TDBGridS1_UnboundReadData(ByVal RowBuf As TrueDBGridStd.RowBuffer,
StartLocation As Variant, ByVal ReadPriorRows As Boolean)

```

```

    Dim i, j, r As Integer

```

```

    Dim BOT As Boolean, EOT As Boolean

```

```

    If oVTableResult Is Nothing Then

```

```

        RowBuf.RowCount = 0

```

```

        Exit Sub

```

```

    End If

```

```

    r = 0

```

```

    If IsNull(StartLocation) Then

```

```

        If ReadPriorRows Then

```

```

            oVTableResult.LastRecord

```

```

            BOT = False

```

```

            EOT = False

```

```

        Else

```

```

            oVTableResult.FirstRecord

```

```

            BOT = False

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    EOT = False
End If
Else
    oVTableResult.GotoRecord (StartLocation)
If ReadPriorRows Then
    If oVTableResult.PreviousRecord Then
        BOT = False
        EOT = False
    Else
        BOT = True
        EOT = False
    End If
Else
    If oVTableResult.NextRecord Then
        EOT = False
        BOT = False
    Else
        EOT = True
        BOT = False
    End If
End If
End If
For i = 0 To RowBuf.RowCount - 1
    If BOT Or EOT Then
        Exit For
    End If
    For j = 0 To oVTableResult.Columns.Count - 1
        RowBuf.Value(i, j) = oVTableResult.Columns(j + 1).Value
    Next j
    RowBuf.Bookmark(i) = oVTableResult.CurrentRecordNumber
If ReadPriorRows Then
    If oVTableResult.PreviousRecord Then
        BOT = False
    Else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        BOT = True
    End If
Else
    If oVTableResult.NextRecord Then
        EOT = False
    Else
        EOT = True
    End If
End If
    r = r + 1
Next i
oVTableResult.FirstRecord
RowBuf.RowCount = r
End Sub

Private Sub ClearGrid()
    Dim c As Object
    Dim i As Integer
    Dim rc As Integer
    rc = 0
    For i = TDBGridS1.Columns.Count To 0 Step -1
        If i = 0 Then Exit For
        TDBGridS1.Columns.Remove(i - 1)
    Next i
    Call SetNameColumn
End Sub

Private Sub SetNameColumn()
    For i = 0 To oVTableResult.Columns.Count - 1
        Set c = TDBGridS1.Columns.Add(i)
        With c
            .Visible = True
            .Caption = UCase(ColumnName(i))
            .Locked = True
        End With
    Next i
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Next i
```

```
End Sub
```

```
-----:
```

2.13 SearchBook.frm

```
Public oVTableBook As ddoTable
```

```
Dim SQLString As String 'SQL Current
```

```
Dim Countcheck As Byte 'Count All CheckBox is Checked
```

```
Private Sub Check1_Click(Index As Integer)
```

```
If Index = 3 Then
```

```
    If Check1(3).Value = 1 Then
```

```
        Combo1.Visible = True
```

```
        Countcheck = Countcheck + 1
```

```
        Combo1.SetFocus
```

```
    Else
```

```
        Combo1.Visible = False
```

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

```
        Countchec = Countcheck - 1
```

```
    End If
```

```
Else:
```

```
If Check1(Index).Value = 1 Then
```

```
    With Text1(Index)
```

```
        .Text = ""
```

```
        .Visible = True
```

```
        .SetFocus
```

```
    End With
```

```
    Countcheck = Countcheck + 1
```

```
Else: Text1(Index).Visible = False
```

```
If Index = 5 Then
```

```
    Text1(5).Text = 0
```

```
Else: Text1(Index).Text = ""
```

```
End If
```

```
Label1.Visible = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Countcheck = Countcheck - 1
    End If
End If
If Countcheck > 0 Then
    Command2.Enabled = True
    Label2.Visible = True
Else: Command2.Enabled = False
    Label2.Visible = False
End If
End Sub

Private Sub Combo1_GotFocus()
    Label1.Visible = True
    Label1.Caption = "Computer"
End Sub

Private Sub Command1_Click()
    Unload Me
    MenuClient.Show
End Sub

Private Sub Command2_Click()
    Dim check As Boolean
    check = True
    For i% = 0 To 5
        If i% = 3 Then
            If (Check1(3) = 1) And (Combo1.Text = "") Then
                check = False
            End If
        Else:
            If (Check1(i%) = 1) And (Text1(i%).Text = "") Then
                check = False
            End If
        End If
    End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Next i%
If check Then
PassSQL
Else: MsgBox ("โปรดใส่ข้อความที่คุณต้องการค้นหา หัวข้อที่คุณเช็ค")
End If
End Sub

Private Function PrepareSQL() As String
SQL$ = ""
For i% = 0 To 4
    If i% = 3 Then
        SQL$ = SQL$ + UCase(Combo1.Text)
    Else:
        SQL$ = SQL$ + UCase(Text1(i%).Text)
    End If
    If i% <> 4 Then
        SQL$ = SQL$ + "',''"
    End If
Next i%
SQL$ = SQL$ + "',''" + Text1(5).Text
PrepareSQL = SQL$
End Function

Private Sub PassSQL()
SQLString = PrepareSQL
SQLString = "execute function getinfobook('"' + SQLString + "'")
MainScreen.oDataGroupBook.BeginTransaction
Set oVTableBook = MainScreen.oDataGroupBook.ExecuteSQLCommand(SQLString)
MainScreen.oDataGroupBook.CommitTransaction
If oVTableBook Is Nothing Then
    'No data return
MsgBox ("No Data in this Multimedia database")
MainScreen.oDataGroupBook.DeleteVirtualTable ("oVTableBook")
Else: SearchBook.Hide
Load Result
Result.Show
End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End Sub
Private Sub Command3_Click()
For c% = 0 To 5
Check1(c%).Value = 0
Next c%
End Sub

Private Sub Text1_GotFocus(Index As Integer)
Label1.Visible = True
Select Case Index
Case 0
Label1.Caption = "QA.456"
Case 1
Label1.Caption = "411 Digital"
Case 2
Label1.Caption = "Pratya"
Case 4
Label1.Caption = "Amazon"
Case 5
Label1.Caption = "1998"
End Select
End Sub
:-----:

```

2.14 SearchMovie.frm

```

Public oVTableMovie As ddoTable
Dim SQLString As String 'SQL Current
Dim Countcheck As Byte 'Count All CheckBox is Checked

Private Sub Check1_Click(Index As Integer)
If Index = 3 Then
If Check1(3).Value = 1 Then
Combo1.Visible = True
Countcheck = Countcheck + 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Combo1.SetFocus
Else
Combo1.Visible = False
Combo1.Text = ""
Countchec = Countcheck - 1
End If
Else:
If Check1(Index).Value = 1 Then
With Text1(Index)
.Text = ""
.Visible = True
.SetFocus
End With
Countcheck = Countcheck + 1
Else: Text1(Index).Visible = False
Text1(Index).Text = ""
Label2.Visible = False
Countcheck = Countcheck - 1
End If
End If
If Countcheck > 0 Then
Label1.Visible = True
Command3.Enabled = True
Else: Command3.Enabled = False
Label1.Visible = False
End If
End Sub
Private Sub Combo1_GotFocus()
Label2.Visible = True
Label2.Caption = "Action"
End Sub
Private Sub Command1_Click()
Unload Me
MenuClient.Show

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

```
Private Sub Command2_Click()
```

```
For c% = 0 To 5
```

```
Check1(c%).Value = 0
```

```
Next c%
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
Dim check As Boolean
```

```
check = True
```

```
For i% = 0 To 5
```

```
    If i% = 3 Then
```

```
        If (Check1(3) = 1) And (Combo1.Text = "") Then
```

```
            check = False
```

```
        End If
```

```
    Else:
```

```
        If (Check1(i%) = 1) And (Text1(i%).Text = "") Then
```

```
            check = False
```

```
        End If
```

```
    End If
```

```
    If i% = 4 And (Text1(4).Text = "") Then
```

```
        Text1(4).Text = 0
```

```
    End If
```

```
Next i%
```

```
If check Then
```

```
    PassSQL
```

```
Else: MsgBox ("โปรดใส่ข้อความที่คุณต้องการค้นหา ณ หัวข้อที่คุณเช็ค")
```

```
End If
```

```
End Sub
```

```
Private Sub PassSQL()
```

```
SQLString = PrepareSQL
```

```
SQLString = "execute function getinfomovie('"' + SQLString + "'")"
```

```
MainScreen.oDataGroupMovie.BeginTransaction
```

```
Set oVTableMovie = MainScreen.oDataGroupMovie.ExecuteSQLCommand(SQLString)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MainScreen.oDataGroupMovie.CommitTransaction
If oVTableMovie Is Nothing Then      'No data return
MsgBox ("No Data in this Multimedia database")
MainScreen.oDataGroupMovie.DeleteVirtualTable ("oVTableMovie")
Else: Searchmovie.Hide
Load Result
Result.Show
End If
End Sub

Private Function PrepareSQL() As String
SQL$ = ""
For i% = 0 To 5
    If i% = 3 Then
        SQL$ = SQL$ + UCase(Combo1.Text)
    Else:
        SQL$ = SQL$ + UCase(Text1(i%).Text)
    End If
    If i% <> 5 Then
        SQL$ = SQL$ + "',''"
    End If
Next i%
PrepareSQL = SQL$
End Function

Private Sub Text1_GotFocus(Index As Integer)
Label2.Visible = True
Select Case Index
    Case 0
        Label2.Caption = "MV001"
    Case 1
        Label2.Caption = "Toy Story"
    Case 2
        Label2.Caption = "Merfy"
    Case 4

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Label2.Caption = "1999"
Case 5
Label2.Caption = "Walt Disney"
End Select
End Sub
:-----:

```

2.15 SearchSong.frm

```

Public oVTableMusic As ddoTable
Dim SQLString As String 'SQL Current
Dim Countcheck As Byte 'Count All CheckBox is Checked
Private Sub Check1_Click(Index As Integer)
If Index = 5 Then
    If Check1(5).Value = 1 Then
        Combo1.Visible = True
        Countcheck = Countcheck + 1
        Combo1.SetFocus
    Else
        Combo1.Visible = False
        Combo1.Text = ""
        Countcheck = Countcheck - 1
    End If
Else:
    If Check1(Index).Value = 1 Then
        With Text1(Index)
            .Text = ""
            .Visible = True
            .SetFocus
        End With
        Countcheck = Countcheck + 1
    Else: Text1(Index).Visible = False
        Text1(Index).Text = ""
        Label2.Visible = False
        Countcheck = Countcheck - 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    End If

End If

If Countcheck > 0 Then
    Command3.Enabled = True
    Label1.Visible = True
Else: Command3.Enabled = False
    Label1.Visible = False
End If

End Sub

Private Sub Combo1_GotFocus()
    Label2.Visible = True
    Label2.Caption = "Pop"
End Sub

Private Sub Command1_Click()
    Unload Me
    MenuClient.Show
End Sub

Private Sub Command2_Click()
    For c% = 0 To 6
        Check1(c%).Value = 0
    Next c%
End Sub

Private Sub Command3_Click()
    Dim check As Boolean
    check = True
    For i% = 0 To 6
        If i% = 5 Then
            If (Check1(5) = 1) And (Combo1.Text = "") Then
                check = False
            End If
        End If
    Next i%
Else:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    If (Check1(i%) = 1) And (Text1(i%).Text = "") Then
        check = False
    End If
End If
If (i%) = 4 And (Text1(4).Text = "") Then
    Text1(4).Text = 0
End If
Next i%
If check Then
    PassSQL
Else: MsgBox ("โปรดใส่ข้อความที่คุณต้องการค้นหา น ห้วข้อที่คุณเช็ค")
End If
End Sub

Private Function PrepareSQL() As String
    SQL$ = ""
    For i% = 0 To 6
        If i% = 5 Then
            SQL$ = SQL$ + UCase(Combo1.Text)
        Else:
            SQL$ = SQL$ + UCase(Text1(i%).Text)
        End If
    If i% <> 6 Then
        SQL$ = SQL$ + " " , ""
    End If
    Next i%
    PrepareSQL = SQL$
End Function

Private Sub PassSQL()
    SQLString = PrepareSQL
    SQLString = "execute function getinformusic("" + SQLString + "")"
    MainScreen.oDataGroupMusic.BeginTransaction
    Set oVTableMusic = MainScreen.oDataGroupMusic.ExecuteSQLCommand(SQLString)
    MainScreen.oDataGroupMusic.CommitTransaction

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If oVTableMusic Is Nothing Then      'No data return
MsgBox ("No Data in this Multimedia database")
MainScreen.oDataGroupMusic.DeleteVirtualTable ("oVTableMusic")
Else: SearchSong.Hide
Load Result
Result.Show
End If
End Sub

```

```

Private Sub Text1_GotFocus(Index As Integer)

```

```

Label2.Visible = True

```

```

Select Case Index

```

```

    Case 0

```

```

        Label2.Caption = "MS001"

```

```

    Case 1

```

```

        Label2.Caption = "I knew I loved you"

```

```

    Case 2

```

```

        Label2.Caption = "Savage Garden"

```

```

    Case 4

```

```

        Label2.Caption = "1999"

```

```

    Case 6

```

```

        Label2.Caption = "RS Promotion"

```

```

End Select

```

```

End Sub

```

```

:-----:

```

2.16 Shelf_Room.frm

```

Option Explicit

```

```

Dim CountRoom As ddoTable

```

```

Dim oVTableResult As ddoTable

```

```

Dim ColumnName() As String

```

```

Dim SQLSearchAll As String

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub cmdIBook_Click()
Dim SQLInsertShelf As String
SQLInsertShelf = "execute procedure insertbookshelf(" + """" + UCase(txtINBook) + """, " + _
                txtILBook + ", " + cmbIRBook + ", " + txtIPXBook + ", " + txtIPYBook + ")"
MainScreen.oDataGroupBook.BeginTransaction
MainScreen.oDataGroupBook.ExecuteSqlCommand (SQLInsertShelf)
MainScreen.oDataGroupBook.CommitTransaction
MsgBox "Insert ตู้เก็บหนังสือเรียบร้อยแล้ว", , "Insert ตู้เก็บหนังสือ"
End Sub

```

```

Private Sub cmdIMovie_Click()
Dim SQLInsertShelf As String
SQLInsertShelf = "execute procedure insertmovieshelf(" + """" + UCase(txtINMovie) + """, " + _
                txtILMovie + ", " + cmbIRMovie + ", " + txtIPXMovie + ", " + txtIPYMovie + ")"
MainScreen.oDataGroupBook.BeginTransaction
MainScreen.oDataGroupBook.ExecuteSqlCommand (SQLInsertShelf)
MainScreen.oDataGroupBook.CommitTransaction
MsgBox "Insert ตู้เก็บภาพยนตร์เรียบร้อยแล้ว", , "Insert ตู้เก็บภาพยนตร์"
End Sub

```

```

Private Sub cmdIMusic_Click()
Dim SQLInsertShelf As String
SQLInsertShelf = "execute procedure insertmusicshelf(" + """" + UCase(txtINMusic) + """, " + _
                txtILMusic + ", " + cmbIRMusic + ", " + txtIPXMusic + ", " + txtIPYMusic + ")"
MainScreen.oDataGroupBook.BeginTransaction
MainScreen.oDataGroupBook.ExecuteSqlCommand (SQLInsertShelf)
MainScreen.oDataGroupBook.CommitTransaction
MsgBox "Insert ตู้เก็บเพลงเรียบร้อยแล้ว", , "Insert ตู้เก็บเพลง"
End Sub

```

```

Private Sub cmdIRoom_Click()
Dim SQLInsertRoom As String
SQLInsertRoom = "Insert Into room Values(0, """" + UCase(txtINRoom) + """)"
MainScreen.oDataGroupBook.BeginTransaction

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MainScreen.oDataGroupBook.ExecuteSqlCommand (SQLInsertRoom)
MainScreen.oDataGroupBook.CommitTransaction
MsgBox "Insert ห้องสมุดเรียบร้อยแล้ว", "Insert Room"
Call SetcmbCountRoom
End Sub

```

```

Private Sub cmdSDelete_Click()
Dim DeleteSQL As String
Dim Typedelete As String
oVTableResult.GotoRecord (TDBGridS1.Row + 1)
If (Option1(0).Value) Then
Typedelete = "0"
ElseIf (Option1(1).Value) Then
Typedelete = "1"
ElseIf (Option1(2).Value) Then
Typedelete = "2"
Else: Typedelete = "3"
End If
DeleteSQL = "execute procedure DELETESHELF_ROOM(" + oVTableResult.Columns(1).Value + ","
+ Typedelete + ")"
MainScreen.oDataGroupBook.BeginTransaction
MainScreen.oDataGroupBook.ExecuteSqlCommand (DeleteSQL)
Set oVTableResult = MainScreen.oDataGroupBook.ExecuteSqlCommand(SQLSearchAll)
MainScreen.oDataGroupBook.CommitTransaction
TDBGridS1.Refresh
Call SetcmbCountRoom
Call ClearGrid
End Sub

```

```

Private Sub cmdSSearch_Click()
MainScreen.oDataGroupBook.BeginTransaction
Set oVTableResult = MainScreen.oDataGroupBook.ExecuteSqlCommand(SQLSearchAll)
MainScreen.oDataGroupBook.CommitTransaction
Call ClearGrid

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End Sub

Private Sub ClearGrid()
    Dim i As Integer
    Dim rc As Integer
    rc = 0
    For i = TDBGridS1.Columns.Count To 0 Step -1
        If i = 0 Then Exit For
        TDBGridS1.Columns.Remove (i - 1)
    Next i

    If Not (oVTableResult Is Nothing) Then 'Have data return
        Call SetColumnname
        cmdSDelete.Enabled = True
        cmdSUpdate.Enabled = True
    Else: MsgBox "No Data in this Multimedia database", , "No Data"
        MainScreen.oDataGroupBook.DeleteVirtualTable ("oVTableResult")
        cmdSDelete.Enabled = False
        cmdSUpdate.Enabled = False
    End If
End Sub

Private Sub SetColumnname()
    Dim c As Object
    Dim i As Byte
    If Option1(0) Or Option1(1) Or Option1(2) Then
        ReDim ColumnName(6)
        ColumnName(0) = "SHELFID"
        ColumnName(1) = "SHELFNAME"
        ColumnName(2) = "LAYER"
        ColumnName(3) = "ROOMID"
        ColumnName(4) = "POSITION_X"
        ColumnName(5) = "POSITION_Y"
    Else
        ReDim ColumnName(2)
        ColumnName(0) = "ROOMID"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        ColumnName(1) = "ROOMNAME"
    End If
    For i = 0 To oVTableResult.Columns.Count - 1
        Set c = TDBGridS1.Columns.Add(i)
        With c
            .Visible = True
            .Caption = UCase(ColumnName(i))
            .Locked = True
        End With
    Next i
End Sub

```

```

Private Sub cmdSUpdate_Click()
    Dim Search_Update As String
    TabShelf_Room.Tabs(2).Selected = True
    Frame1(1).ZOrder 0
    oVTableResult.GotoRecord (TDBGridS1.Row + 1)
    If (Option1(0).Value) Then
        txtUIDBook.Text = oVTableResult.Columns(1).Value
        txtUNBook.Text = oVTableResult.Columns(2).Value
        txtULBook.Text = oVTableResult.Columns(3).Value
        cmbURBook.Text = oVTableResult.Columns(4).Value
        txtUPXBook = oVTableResult.Columns(5).Value
        txtUPYBook = oVTableResult.Columns(6).Value
        cmdUUpdateBook.Enabled = True
    ElseIf (Option1(1).Value) Then
        txtUIDMusic.Text = oVTableResult.Columns(1).Value
        txtUNMusic.Text = oVTableResult.Columns(2).Value
        txtULMusic.Text = oVTableResult.Columns(3).Value
        cmbURMusic.Text = oVTableResult.Columns(4).Value
        txtUPXMusic = oVTableResult.Columns(5).Value
        txtUPYMusic = oVTableResult.Columns(6).Value
        cmdUUpdateMusic.Enabled = True
    ElseIf (Option1(2).Value) Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

txtUIDMovie.Text = oVTableResult.Columns(1).Value
txtUNMovie.Text = oVTableResult.Columns(2).Value
txtULMovie.Text = oVTableResult.Columns(3).Value
cmbURMovie.Text = oVTableResult.Columns(4).Value
txtUPXMovie.Text = oVTableResult.Columns(5).Value
txtUPYMovie.Text = oVTableResult.Columns(6).Value
cmdUUpdateMovie.Enabled = True

Else: txtUIDRoom.Text = oVTableResult.Columns(1).Value
      txtUNRoom.Text = oVTableResult.Columns(2).Value
      cmdUUpdateRoom.Enabled = True

End If
End Sub

Private Sub cmdUSearchBook_Click()
Dim SearchDetail As ddoTable
Dim SQLSearch As String
SQLSearch = "Select shelfname,layer,roomid,position.x,position.y from bookshelf t1,addressbookshelf
t2 where t1.shelfid = " + txtUIDBook
MainScreen.oDataGroupBook.BeginTransaction
Set SearchDetail = MainScreen.oDataGroupBook.ExecuteSQLCommand(SQLSearch)
MainScreen.oDataGroupBook.CommitTransaction
If Not (SearchDetail Is Nothing) Then
    txtUNBook.Text = SearchDetail.Columns(1).Value
    txtULBook.Text = SearchDetail.Columns(2).Value
    cmbURBook.Text = SearchDetail.Columns(3).Value
    txtUPXBook.Text = SearchDetail.Columns(4).Value
    txtUPYBook.Text = SearchDetail.Columns(5).Value
    cmdUUpdateBook.Enabled = True

Else: MsgBox "No Data in this Multimedia database", , "No Data"
      txtUIDBook.Text = ""
      cmdUUpdateBook.Enabled = False

End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub cmdUSearchMovie_Click()
Dim SearchDetail As ddoTable
Dim SQLSearch As String
SQLSearch = "Select shelfname, layer, roomid, position.x, position.y from movieshelf t1,
addressmovieshelf t2 where t1.shelfid = " + txtUIDMovie
MainScreen.oDataGroupBook.BeginTransaction
Set SearchDetail = MainScreen.oDataGroupBook.ExecuteSQLCommand(SQLSearch)
MainScreen.oDataGroupBook.CommitTransaction
If Not (SearchDetail Is Nothing) Then
    txtUNMovie.Text = SearchDetail.Columns(1).Value
    txtULMovie.Text = SearchDetail.Columns(2).Value
    cmbURMovie.Text = SearchDetail.Columns(3).Value
    txtUPXMovie.Text = SearchDetail.Columns(4).Value
    txtUPYMovie.Text = SearchDetail.Columns(5).Value
    cmdUUpdateMovie.Enabled = True
Else: MsgBox "No Data in this Multimedia database", , "No Data"
    txtUIDMovie.Text = ""
    cmdUUpdateMovie.Enabled = False
End If
End Sub

```

```

Private Sub cmdUSearchMusic_Click()
Dim SearchDetail As ddoTable
Dim SQLSearch As String
SQLSearch = "Select shelfname, layer, roomid, position.x, position.y from musicshelf t1,
addressmusicshelf t2 where t1.shelfid = " + txtUIDMusic
MainScreen.oDataGroupBook.BeginTransaction
Set SearchDetail = MainScreen.oDataGroupBook.ExecuteSQLCommand(SQLSearch)
MainScreen.oDataGroupBook.CommitTransaction
If Not (SearchDetail Is Nothing) Then
    txtUNMusic.Text = SearchDetail.Columns(1).Value
    txtULMusic.Text = SearchDetail.Columns(2).Value
    cmbURMusic.Text = SearchDetail.Columns(3).Value
    txtUPXMusic.Text = SearchDetail.Columns(4).Value

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

txtUPYMusic.Text = SearchDetail.Columns(5).Value
cmdUUpdateMusic.Enabled = True
Else: MsgBox "No Data in this Multimedia database", , "No Data"
txtUIDMusic.Text = ""
cmdUUpdateMusic.Enabled = False
End If
End Sub

```

```

Private Sub cmdUSearchRoom_Click()
Dim SearchDetail As ddoTable
Dim SQLSearch As String
SQLSearch = "Select roomname from room where roomid = " + txtUIDRoom
MainScreen.oDataGroupBook.BeginTransaction
Set SearchDetail = MainScreen.oDataGroupBook.ExecuteSQLCommand(SQLSearch)
MainScreen.oDataGroupBook.CommitTransaction
If Not (SearchDetail Is Nothing) Then
txtUNRoom.Text = SearchDetail.Columns(1).Value
cmdUUpdateRoom.Enabled = True
Else: MsgBox "No Data in this Multimedia database", , "No Data"
txtUIDRoom.Text = ""
cmdUUpdateRoom.Enabled = False
End If
End Sub

```

```

Private Sub cmdUUpdateBook_Click()
Dim SQLUpdate1, SQLUpdate2 As String
SQLUpdate1 = "Update bookshelf Set shelfname = "" + UCase(txtUNBook) + "" + _
",layer = " + txtULBook + " where shelfid = " + txtUIDBook
SQLUpdate2 = "Update addressbookshelf Set roomid = " + cmbURBook + _
",position = Row(" + txtUPXBook + " , " + txtUPYBook + ")::position_t" + _
" where shelfid = " + txtUIDBook
MainScreen.oDataGroupBook.BeginTransaction
MainScreen.oDataGroupBook.ExecuteSQLCommand (SQLUpdate1)
MainScreen.oDataGroupBook.ExecuteSQLCommand (SQLUpdate2)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MainScreen.oDataGroupBook.CommitTransaction
MsgBox "Update ข้อมูลของตู้เก็บหนังสือเรียบร้อยแล้ว", , "Update ตู้เก็บหนังสือ"
End Sub

```

```

Private Sub cmdUUpdateMovie_Click()
Dim SQLUpdate1, SQLUpdate2 As String
SQLUpdate1 = "Update movieshelf Set shelfname = "" + UCase(txtUNMovie) + """" + _
    ",layer = " + txtULMovie + " where shelfid = " + txtUIDMovie
SQLUpdate2 = "Update addressmovieshelf Set roomid = " + cmbURMovie + _
    ",position = Row(" + txtUPXMovie + " , " + txtUPYMovie + ")::position_1" + _
    " where shelfid = " + txtUIDMovie

```

```

MainScreen.oDataGroupBook.BeginTransaction
MainScreen.oDataGroupBook.ExecuteSqlCommand (SQLUpdate1)
MainScreen.oDataGroupBook.ExecuteSqlCommand (SQLUpdate2)
MainScreen.oDataGroupBook.CommitTransaction
MsgBox "Update ข้อมูลของตู้เก็บภาพยนตร์เรียบร้อยแล้ว", , "Update ตู้เก็บภาพยนตร์"
End Sub

```

```

Private Sub cmdUUpdateMusic_Click()
Dim SQLUpdate1, SQLUpdate2 As String
SQLUpdate1 = "Update musicshelf Set shelfname = "" + UCase(txtUNMusic) + """" + _
    ",layer = " + txtULMusic + " where shelfid = " + txtUIDMusic
SQLUpdate2 = "Update addressmusicshelf Set roomid = " + cmbURMusic + _
    ",position = Row(" + txtUPXMusic + " , " + txtUPYMusic + ")::position_1" + _
    " where shelfid = " + txtUIDMusic

```

```

MainScreen.oDataGroupBook.BeginTransaction
MainScreen.oDataGroupBook.ExecuteSqlCommand (SQLUpdate1)
MainScreen.oDataGroupBook.ExecuteSqlCommand (SQLUpdate2)
MainScreen.oDataGroupBook.CommitTransaction
MsgBox "Update ข้อมูลของตู้เก็บเพลงเรียบร้อยแล้ว", , "Update ตู้เก็บเพลง"
End Sub

```

```

Private Sub cmdUUpdateRoom_Click()
Dim SQLUpdate1, SQLUpdate2 As String

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SQLUpdate1 = "Update room Set roomname = "" + UCase(txtUNRoom) + "" + _
                " where roomid = " + txtUIDRoom
MainScreen.oDataGroupBook.BeginTransaction
MainScreen.oDataGroupBook.ExecuteSqlCommand (SQLUpdate1)
MainScreen.oDataGroupBook.CommitTransaction
MsgBox "Update ข้อมูลของห้องสมุดเรียบร้อยแล้ว", "Update ห้องสมุด"
End Sub

Private Sub Command1_Click()
MenuAdmin.Show
Unload Me
End Sub

Private Sub Form_Load()
Dim i As Byte
For i = 0 To Frame1.Count - 1
    With Frame1(i)
        .Move TabShelf_Room.ClientLeft + 40, _
            TabShelf_Room.ClientTop + 40, _
            TabShelf_Room.ClientWidth - 100, _
            TabShelf_Room.ClientHeight - 100
    End With
Next i
Frame1(0).ZOrder 0
Call SetcmbCountRoom
End Sub

Private Sub SetcmbCountRoom()
MainScreen.oDataGroupBook.BeginTransaction
Set CountRoom = MainScreen.oDataGroupBook.ExecuteSqlCommand("Select roomid from room")
MainScreen.oDataGroupBook.CommitTransaction
If Not (CountRoom Is Nothing) Then
    cmbIRBook.Clear
    cmbIRMusic.Clear

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

cmbIRMovie.Clear
cmbIRBook.Text = CountRoom.Columns(1).Value
cmbIRMusic.Text = CountRoom.Columns(1).Value
cmbIRMovie.Text = CountRoom.Columns(1).Value
cmbURBook.Clear
cmbURMusic.Clear
cmbURMovie.Clear
cmbURBook.Text = CountRoom.Columns(1).Value
cmbURMusic.Text = CountRoom.Columns(1).Value
cmbURMovie.Text = CountRoom.Columns(1).Value
Do While CountRoom.EOF = False
    cmbIRBook.AddItem (CountRoom.Columns(1).Value)
    cmbIRMusic.AddItem (CountRoom.Columns(1).Value)
    cmbIRMovie.AddItem (CountRoom.Columns(1).Value)
    cmbURBook.AddItem (CountRoom.Columns(1).Value)
    cmbURMusic.AddItem (CountRoom.Columns(1).Value)
    cmbURMovie.AddItem (CountRoom.Columns(1).Value)
    CountRoom.NextRecord
Loop
End If
End Sub

Private Sub Option1_Click(Index As Integer)
If Index = 0 Then
    SQLSearchAll = "select t1.shelfid,shelfname,layer,t2.roomid,position.x,position.y" + _
        " from bookshelf t1,addressbookshelf t2,room t3" + _
        " where t2.shelfid = t1.shelfid and t3.roomid = t2.roomid"
ElseIf Index = 1 Then
    SQLSearchAll = "select t1.shelfid,shelfname,layer,t2.roomid,position.x,position.y" + _
        " from musicshelf t1,addressmusicshelf t2,room t3" + _
        " where t2.shelfid = t1.shelfid and t3.roomid = t2.roomid"
ElseIf Index = 2 Then
    SQLSearchAll = "select t1.shelfid,shelfname,layer,t2.roomid,position.x,position.y" + _
        " from movieshelf t1,addressmovieshelf t2,room t3" + _

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

" where t2.shelfid = t1.shelfid and t3.roomid = t2.roomid"
Else
    SQLSearchAll = "select roomid,roomname from room"
End If
End Sub

Private Sub TabShelf_Room_Click()
Frame1(TabShelf_Room.SelectedItem.Index - 1).ZOrder 0
If TabShelf_Room.Tabs(3).Selected Then
    cmdSDelete.Enabled = False
    cmdSUpdate.Enabled = False
    cmdUUpdateBook.Enabled = False
    cmdUUpdateMusic.Enabled = False
    cmdUUpdateMovie.Enabled = False
    cmdUUpdateRoom.Enabled = False
End If
End Sub

Private Sub TDBGridSI_UnboundReadData(ByVal RowBuf As TrueDBGridStd.RowBuffer,
StartLocation As Variant, ByVal ReadPriorRows As Boolean)
    Dim i, j, r As Integer
    Dim BOT As Boolean, EOT As Boolean
    If oVTableResult Is Nothing Then
        RowBuf.RowCount = 0
        Exit Sub
    End If
    r = 0
    If IsNull(StartLocation) Then
        If ReadPriorRows Then
            oVTableResult.LastRecord
            BOT = False
            EOT = False
        Else
            oVTableResult.FirstRecord

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    BOT = False
    EOT = False
End If
Else
    oVTableResult.GotoRecord (StartLocation)
    If ReadPriorRows Then
        If oVTableResult.PreviousRecord Then
            BOT = False
            EOT = False
        Else
            BOT = True
            EOT = False
        End If
    Else
        If oVTableResult.NextRecord Then
            EOT = False
            BOT = False
        Else
            EOT = True
            BOT = False
        End If
    End If
End If
For i = 0 To RowBuf.RowCount - 1
    If BOT Or EOT Then
        Exit For
    End If
    For j = 0 To oVTableResult.Columns.Count - 1
        RowBuf.Value(i, j) = oVTableResult.Columns(j + 1).Value
    Next j
    RowBuf.Bookmark(i) = oVTableResult.CurrentRecordNumber
    If ReadPriorRows Then
        If oVTableResult.PreviousRecord Then
            BOT = False

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Else
    BOT = True
End If

Else
    If oVTableResult.NextRecord Then
        EOT = False
    Else
        EOT = True
    End If
End If

r = r + 1

Next i
oVTableResult.FirstRecord
RowBuf.RowCount = r
End Sub

Private Sub txtUIDBook_Change()
Dim CheckInteger As String
CheckInteger = Val(txtUIDBook.Text)
If CheckInteger = txtUIDBook.Text Then
    cmdUSearchBook.Enabled = True
Else: cmdUSearchBook.Enabled = False
End If
End Sub

Private Sub txtUIDMovie_Change()
Dim CheckInteger As String
CheckInteger = Val(txtUIDMovie.Text)
If CheckInteger = txtUIDMovie.Text Then
    cmdUSearchMovie.Enabled = True
Else: cmdUSearchMovie.Enabled = False
End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub txtUIDMusic_Change()
Dim CheckInteger As String
CheckInteger = Val(txtUIDMusic.Text)
If CheckInteger = txtUIDMusic.Text Then
    cmdUSearchMusic.Enabled = True
Else: cmdUSearchMusic.Enabled = False
End If
End Sub

```

```

Private Sub txtUIDRoom_Change()
Dim CheckInteger As String
CheckInteger = Val(txtUIDRoom.Text)
If CheckInteger = txtUIDRoom.Text Then
    cmdUSearchRoom.Enabled = True
Else: cmdUSearchRoom.Enabled = False
End If
End Sub

```

```

:-----:

```

2.17 ShowLibrary.frm

```

Dim X1, Y1 As Integer
Dim Distancex, Distancey As Single
Private Sub Command2_Click()
X1 = Cint(Result.oVTableDetail.Columns(16).Value)
Y1 = Cint(Result.oVTableDetail.Columns(17).Value)
If (Image2.Left = X1) And (Image2.Top = Y1) Then
    Image2.Left = 11600
    Image2.Top = 1600
    If X1 - Image2.Left > 0 Then
        Distancex = 200
    Else: Distancex = -200
    End If
    If Y1 - Image2.Top > 0 Then
        Distancey = 200

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Else: Distancey = -200
End If
Timer1.Enabled = True
Else:
If X1 - Image2.Left > 0 Then
Distancex = 200
Else: Distancex = -200
End If
If Y1 - Image2.Top > 0 Then
Distancey = 200
Else: Distancey = -200
End If
Timer1.Enabled = True
End If
End Sub
Private Sub Command3_Click()
Unload Me
DetailBook.Show
End Sub
Private Sub Form_Load()
Load Me
Image1.Picture = LoadPicture("e:\Project\ProjectVB\VirtualLibrary\picture\Librapic2_2.bmp")
Image2.Picture = LoadPicture("e:\Project\ProjectVB\VirtualLibrary\picture\pointer3.gif")
Image2.Left = 11600
Image2.Top = 1600
End Sub

Private Sub Timer1_Timer()
If Image2.Top <> Y1 Then
If Distancey > 0 Then
If Image2.Top + Distancey <= Y1 Then
Image2.Top = Image2.Top + Distancey
Else: Image2.Top = Y1
End If
End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Else
    If Image2.Top + Distancecy >= Y1 Then
        Image2.Top = Image2.Top + Distancecy
    Else: Image2.Top = Y1
    End If
End If
Elseif Image2.Left <> X1 Then
    If Distancex > 0 Then
        If Image2.Left + Distancex <= X1 Then
            Image2.Left = Image2.Left + Distancex
        Else: Image2.Left = X1
        End If
    Else
        If Image2.Left + Distancex >= X1 Then
            Image2.Left = Image2.Left + Distancex
        Else: Image2.Left = X1
        End If
    End If
Elseif (Image2.Left = X1) And (Image2.Top = Y1) Then
    Timer1.Enabled = False
    Timer2.Enabled = True
End If
End Sub

```

```

Private Sub Timer2_Timer()
    Load ShowShelf
    ShowShelf.Show vbModal
    Timer2.Enabled = False
End Sub

```

```

:-----:

```

2.18 ShowMovie.frm

```

Dim X1, Y1 As Integer
Dim Distancex, Distancecy As Single

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Command1_Click()
X1 = CInt(Result.oVTableDetail.Columns(14).Value)
Y1 = CInt(Result.oVTableDetail.Columns(15).Value)
    If X1 - Image2.Left > 0 Then
        Distancex = 200
    Else: Distancex = -200
    End If
    If Y1 - Image2.Top > 0 Then
        Distancey = 200
    Else: Distancey = -200
    End If
    Timer1.Enabled = True
End Sub

Private Sub Command2_Click()
Unload Me
DetailMovie.Show
End Sub

Private Sub Form_Load()
Load Me
Image1.Picture = LoadPicture("e:\Project\ProjectVB\VirtualLibrary\picture\Librapic2_2.bmp")
Image2.Picture = LoadPicture("e:\Project\ProjectVB\VirtualLibrary\picture\Pointer3.gif")
Image2.Left = 11600
Image2.Top = 1600
Label6.Caption = Result.oVTableDetail.Columns(3).Value
Label7.Caption = Result.oVTableDetail.Columns(2).Value
Label8.Caption = Result.oVTableDetail.Columns(10).Value
Select Case Result.oVTableDetail.Columns(13).Value
    Case "L"
        Label9.Caption = "ซ้าย"
    Case "M"
        Label9.Caption = "กลาง"
    Case "R"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Label9.Caption = "ขวา"
End Select
Label10.Caption = Result.oVTableDetail.Columns(11).Value
End Sub

```

```

Private Sub Timer1_Timer()
If Image2.Top <> Y1 Then
    If Distancey > 0 Then
        If Image2.Top + Distancey <= Y1 Then
            Image2.Top = Image2.Top + Distancey
        Else: Image2.Top = Y1
        End If
    Else
        If Image2.Top + Distancey >= Y1 Then
            Image2.Top = Image2.Top + Distancey
        Else: Image2.Top = Y1
        End If
    End If
ElseIf Image2.Left <> X1 Then
    If Distancex > 0 Then
        If Image2.Left + Distancex <= X1 Then
            Image2.Left = Image2.Left + Distancex
        Else: Image2.Left = X1
        End If
    Else
        If Image2.Left + Distancex >= X1 Then
            Image2.Left = Image2.Left + Distancex
        Else: Image2.Left = X1
        End If
    End If
ElseIf (Image2.Left = X1) And (Image2.Top = Y1) Then
    Timer1.Enabled = False
End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.19 ShowMusic.frm

Dim X1, Y1 As Integer

Dim Distancex, Distancey As Single

Private Sub Command1_Click()

X1 = CInt(Result.oVTableDetail.Columns(14).Value)

Y1 = CInt(Result.oVTableDetail.Columns(15).Value)

If X1 - Image2.Left > 0 Then

Distancex = 200

Else: Distancex = -200

End If

If Y1 - Image2.Top > 0 Then

Distancey = 200

Else: Distancey = -200

End If

Timer1.Enabled = True

End Sub

Private Sub Command2_Click()

Unload Me

DetailSong.Show

End Sub

Private Sub Form_Load()

Load Me

Image1.Picture = LoadPicture("e:\Project\ProjectVB\VirtualLibrary\picture\Librapic2_2.bmp")

Image2.Picture = LoadPicture("e:\Project\ProjectVB\VirtualLibrary\picture\Pointer3.gif")

Image2.Left = 11600

Image2.Top = 1600

Label6.Caption = Result.oVTableDetail.Columns(3).Value

Label7.Caption = Result.oVTableDetail.Columns(2).Value

Label8.Caption = Result.oVTableDetail.Columns(10).Value

Select Case Result.oVTableDetail.Columns(13).Value

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Case "L"

Label9.Caption = "ซ้าย"

Case "M"

Label9.Caption = "กลาง"

Case "R"

Label9.Caption = "ขวา"

End Select

Label10.Caption = Result.oVTableDetail.Columns(11).Value

End Sub

Private Sub Timer1_Timer()

If Image2.Top <> Y1 Then

If Distancey > 0 Then

If Image2.Top + Distancey <= Y1 Then

Image2.Top = Image2.Top + Distancey

Else: Image2.Top = Y1

End If

Else

If Image2.Top + Distancey >= Y1 Then

Image2.Top = Image2.Top + Distancey

Else: Image2.Top = Y1

End If

End If

ElseIf Image2.Left <> X1 Then

If Distancex > 0 Then

If Image2.Left + Distancex <= X1 Then

Image2.Left = Image2.Left + Distancex

Else: Image2.Left = X1

End If

Else

If Image2.Left + Distancex >= X1 Then

Image2.Left = Image2.Left + Distancex

Else: Image2.Left = X1

End If

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End If
Elseif (Image2.Left = X1) And (Image2.Top = Y1) Then
    Timer1.Enabled = False
End If
End Sub
:-----:

```

2.20 ShowPreface.frm

```

Private Sub Form_Load()
Image1.Picture = LoadPicture(DetailBook.Filepreface)
End Sub
Private Sub Image1_Click()
Unload Me
DetailBook.Show
End Sub
:-----:

```

2.21 ShowShelf.frm

```

Dim ShelfLevel As Integer
Dim PartLevel As String

Private Sub Command1_Click()
    Unload Me
    Load ShowLibrary
    ShowLibrary.Show
End Sub

Private Sub Form_Load()
Label6.Caption = Result.oVTableDetail.Columns(3).Value
Label7.Caption = Result.oVTableDetail.Columns(2).Value
Label8.Caption = Result.oVTableDetail.Columns(12).Value
Select Case Result.oVTableDetail.Columns(15).Value
    Case "L"
        Label9.Caption = "ซ้าย"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Case "M"
    Label9.Caption = "กลาง"
Case "R"
    Label9.Caption = "ขวา"
End Select

Label10.Caption = Result.oVTableDetail.Columns(13).Value
ShelfLevel = Result.oVTableDetail.Columns(12).Value
PartLevel = Result.oVTableDetail.Columns(15).Value
Image3.Picture = LoadPicture("e:\Project\ProjectVB\VirtualLibrary\picture\bookshelf1.gif")
Image3.Visible = False
Image4.Picture = LoadPicture("e:\Project\ProjectVB\VirtualLibrary\picture\Bookpoin3.gif")
Image4.Visible = False
If ShelfLevel = 3 And PartLevel = "L" Then
    Image4.Left = 180
    Image4.Top = 700
    Image3.Visible = True
    Image4.Visible = True
ElseIf ShelfLevel = 3 And PartLevel = "M" Then
    Image4.Top = 700
    Image4.Left = 1080
    Image3.Visible = True
    Image4.Visible = True
ElseIf ShelfLevel = 3 And PartLevel = "R" Then
    Image4.Top = 700
    Image4.Left = 2250
    Image3.Visible = True
    Image4.Visible = True
ElseIf ShelfLevel = 2 And PartLevel = "L" Then
    Image4.Top = 2160
    Image4.Left = 180
    Image3.Visible = True
    Image4.Visible = True
ElseIf ShelfLevel = 2 And PartLevel = "M" Then
    Image4.Top = 2160

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Image4.Left = 1080
Image3.Visible = True
Image4.Visible = True
ElseIf ShelfLevel = 2 And PartLevel = "R" Then
Image4.Top = 2200
Image4.Left = 2250
Image3.Visible = True
Image4.Visible = True
ElseIf ShelfLevel = 1 And PartLevel = "L" Then
Image4.Top = 3860
Image4.Left = 180
Image3.Visible = True
Image4.Visible = True
ElseIf ShelfLevel = 1 And PartLevel = "M" Then
Image4.Top = 3860
Image4.Left = 1080
Image3.Visible = True
Image4.Visible = True
ElseIf ShelfLevel = 1 And PartLevel = "R" Then
Image4.Top = 3860
Image4.Left = 2250
Image3.Visible = True
Image4.Visible = True
End If

```

End Sub

-----;

3. Visual C++

```

/*
** Title:      Media_Datatype.h
** SCCSid:     %W% %E% %U%
** CCid:      %W% %E% %U%
** Author:
** Created:    Jan 23, 2000 08:09:37 PM

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

** Description:      Generated header file.
** Comments:        Generated for project Media_Datatype.1.0.
*/
/*
**      Special Note: This file should not be modified.
**      No merging is performed on this header file. It
**      is regenerated each time the project is written.
*/

#ifndef HDR_Media_Datatype_H
#define HDR_Media_Datatype_H
/*
** Configure tracing by setting TRACE_DEBUG_Media_Datatype
** to 0 to completely disable tracing or 1 to enable
** tracing. This define may be set from the compiler
** command line by using the -DTRACE_DEBUG_Media_Datatype=0 flag.
*/
#ifndef TRACE_DEBUG_Media_Datatype
#define TRACE_DEBUG_Media_Datatype 1
#endif
#ifndef DBDK_LOHSIZE
#define DBDK_LOHSIZE      sizeof(MI_LO_HANDLE)
#define DBDK_LOBINFNSIZE DBDK_LOHSIZE
#endif
/* This data structure returned by LOhandles. */
typedef struct
{
    mi_integer      nlos; /* Number of large object handles. */
    MI_LO_HANDLE    los[1]; /* Valid large object handles. */
} MI_LO_HANDLES;
/*
**      Large object file name mask. '?' is a wild-card that
**      is filled in when a large object is written to disk.
*/
#define LO_FN_MASK    "????????.lo"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*      Error messages
**
**      English versions of these error messages are automatically
**      added to the syserrors table as part of your DataBlade module
**      registration. If you do not like the default messages, you
**      can create new errors and change these defines to use your
**      new codes. You can not, however, change the text of the default
**      messages because they are shared by other DataBlade modules.
*/

#define ERRORMESG1  "UGEN1"
#define ERRORMESG2  "UGEN2"
#define ERRORMESG3  "UGEN3"
#define ERRORMESG4  "UGEN4"
#define ERRORMESG5  "UGEN5"
#define ERRORMESG6  "UGEN6"
#define ERRORMESG7  "UGEN7"
#define ERRORMESG8  "UGEN8"
#define ERRORMESG9  "UGEN9"
#define ERRORMESG10 "UGENA"
#define ERRORMESG11 "UGENB"
#define ERRORMESG12 "UGENC"
#define ERRORMESG13 "UGEND"
#define ERRORMESG14 "UGENE"
#define ERRORMESG15 "UGENF"
#define ERRORMESG16 "UGENG"
#define ERRORMESG17 "UGENH"
#define ERRORMESG18 "UGENI"
#define ERRORMESG19 "UGENJ"

/* Use DBDK_TRACE to direct trace messages to the trace file. */
#if TRACE_DEBUG_Media_Datatype
#define DBDK_TRACE      (1 << 16)
#else
#define DBDK_TRACE      0
#endif

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*
** Print a message to the trace file and for the user.
** N.B.: This macro uses Gen_Con. Your function must
** declare Gen_Con as MI_CONNECTION * and either
** open the connection or set it to NULL.
*/

#define DBDK_TRACE_ERROR( Caller, ErrNo, ErrLevel ) \
    Gen_Trace \
    ( \
        Gen_Con, \
        Caller, \
        __FILE__, \
        __LINE__, \
        ErrNo, \
        "Media_Datatype", \
        ErrLevel, \
        MI_SQL | DBDK_TRACE \
    );

/* Print a message to the trace file. */
#if TRACE_DEBUG_Media_Datatype
/*
** Print a message to the trace file.
** N.B.: This macro uses Gen_Con. Your function must
** declare Gen_Con as MI_CONNECTION * and either
** open the connection or set it to NULL.
*/

#define DBDK_TRACE_MSG( Caller, ErrNo, ErrLevel ) \
    Gen_Trace \
    ( \
        Gen_Con, \
        Caller, \
        __FILE__, \
        __LINE__, \
        ErrNo, \

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        "Media_Datatype",
        ErrLevel,
        DBDK_TRACE
    );

#else
#define DBDK_TRACE_MSG( Caller, ErrNo, ErrLevel )
#endif

/* These macros are used on entry to, and on exit from, a function. */
#define DBDK_TRACE_ENTER( Caller ) DBDK_TRACE_MSG( Caller, ERRORMSG13, 20 )
#define DBDK_TRACE_EXIT( Caller ) DBDK_TRACE_MSG( Caller, ERRORMSG14, 20 )

/*
**      Interval types.
*/

#define YEAR_TO_MONTH 1
#define DAY_TO_SECOND 2

/* Function prototypes. */
mi_integer Gen_nstrwords( gl_mchar_t *, mi_integer );
gl_mchar_t * Gen_sscanf
(
    MI_CONNECTION *      Gen_Con,
    char *               Gen_Caller,
    gl_mchar_t *        Gen_InData,
    mi_integer          Gen_InDataLen,
    mi_integer          Gen_Width,
    char *              Gen_Format,
    char *              Gen_Result
);

void Gen_LoadLOFromFile
(
    MI_CONNECTION *      Gen_Con,
    char *               Gen_Caller,
    char *               Gen_LOFile,
    MI_LO_HANDLE *      Gen_pLOh

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

);
void Gen_StoreLOToFile
(
    MI_CONNECTION *      Gen_Con,
    char *               Gen_Caller,
    char *               Gen_LOFile,
    MI_LO_HANDLE *      Gen_pLOh
);
void Gen_Trace
(
    MI_CONNECTION *      Gen_Con,
    char *               Gen_Caller,
    char *               Gen_FileName,
    mi_integer          Gen_LineNo,
    char *               Gen_MsgNo,
    char *               Gen_Class,
    mi_integer          Gen_Threshold,
    mi_integer          Gen_MsgType
);
/* BladeSmith 3.40.TXXXXX typedef media_t */
typedef struct
{
    MI_LO_HANDLE        media;
}
media_t;
/* Warning: Do not modify. media_t checksum: 0 */
#endif

```