

ระบบสารสนเทศของสำนักทะเบียนและประมวลผล

ส่วนการจัตตารางเรียนและตารางการใช้ห้องเรียน ผ่านระบบเครือข่ายอินเทอร์เน็ต



โดย

นายพระยา พุกสวัสดิ์
นายอัศวิน สุรสิทธิ์

เลขประจำตัว 38013279
เลขประจำตัว 38013305

อาจารย์ที่ปรึกษา

อาจารย์วิบูลย์ พร้อมพานิชย์



-ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชา วิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าลาดกระบัง
ปีการศึกษา 2540

เลขหม.....

เลขทะเบียน..... 30490

วัน, เดือน, ปี..... 17 ก.ค. 2541

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบสารสนเทศของสำนักทะเบียนและประมวลผลผ่านระบบเครือข่ายอินเทอร์เน็ต

พระยา พุกสวัสดิ์

อัศวิน สุรสิทธิ์

วิบูลย์ พร้อมพานิชย์ อาจารย์ที่ปรึกษา

ปีการศึกษา 2540

บทคัดย่อ

โครงการนี้ เป็นโครงการที่พัฒนาขึ้นเพื่อใช้ในการทำงาน การให้บริการของสำนักทะเบียน และประมวลผล ในส่วนงานของการจัดการารเรียน และตารางห้องเรียนของนักศึกษา จุดประสงค์เพื่อลดความผิดพลาด , เพื่อเพิ่มประสิทธิภาพในการทำงาน , สามารถให้บริการ ได้อย่างรวดเร็ว และเพียงพอ กับความต้องการของผู้ใช้บริการ โดยนำเอาเทคโนโลยีทางด้านสารสนเทศ และระบบอินเทอร์เน็ต มาช่วยในการพัฒนาเพื่อให้ใช้ได้กว้างขวางมากขึ้น

ในการพัฒนาจะใช้สถาปัตยกรรมแบบ 3 ลำดับชั้น (3 Tiers Architecture) ซึ่งจะแบ่งการทำงาน ออกเป็น 3 ส่วนใหญ่ๆ ที่เป็นอิสระจากกัน คือ

ส่วนที่ 1 : เป็นส่วนที่ใช้ติดต่อกับผู้ใช้บริการ (GUI – Graphic User Interface) ใช้โปรแกรม

Web Browser ที่ชื่อ Microsoft Internet Explorer Version 3

ส่วนที่ 2 : เป็นส่วนที่ให้บริการกฎเกณฑ์ของระบบ ในส่วนนี้จะใช้ โปรแกรม Web Server

ที่ชื่อ Microsoft Internet Information Server Version 3

ส่วนที่ 3 : เป็นส่วนที่ให้บริการฐานข้อมูล โดยจะใช้ระบบการจัดการฐานข้อมูล (DBMS - Data

Base Management System) ที่ชื่อ Microsoft SQL Server 6.5

ปริญญาโทปีการศึกษา 2540

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบสารสนเทศของสำนักทะเบียนและประมวลผล ส่วนการจัดตารางเรียนและตารางการใช้
ห้องเรียน ผ่านเครือข่ายอินเทอร์เน็ต

ผู้จัดทำ

1. นายพระยา หุกสวัสดิ์ เลขประจำตัว 38013279
2. นายอัศวิน สุรสิทธิ์ เลขประจำตัว 38013305



..... อาจารย์ที่ปรึกษา
(อาจารย์วิบูลย์ พร้อมพานิชย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Application of Data Base Management System
for
The Register's Office in University by Internet System

Praya Puksawad
Usawin Surasit
Wiboon Prompanich Advisor
1997

Abstract

This Application is developed for processing , using in Office of Central Register and Processing. In part of manage study-schedule and room - schedule composing. This Application can use for reduce errors , increase efficiency of processing , can take a faster services and sufficient for support of user request . We use Information Technology and Internet System to develop .

In application we use 3 Tier Architecture , It have 3 parts are

Tier 1 : User Service is use Microsoft Internet Explorer 3.0

Tier 2 : Business Service is use Microsoft Internet Information Server 3.0

Tier 3 : Data Service is use Microsoft SQL Server 6.5 for Data Base Management System.

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 ไคลเอนต์ – เซิร์ฟเวอร์ (Client – Server)	2
2.2 สถาปัตยกรรมไคลเอนต์ – เซิร์ฟเวอร์ แบบ 2 ระดับ	3
2.3 สถาปัตยกรรมแบบ 3 ระดับ (Three – tier Architecture)	4
2.4 โมเดลเชิงสัมพันธ์ (Relational Model)	7
2.5 ทฤษฎีของระบบฐานข้อมูลเชิงสัมพันธ์	11
2.6 การออกแบบฐานข้อมูลรวมแบบรีเลชันแนล	13
2.7 ระบบสารสนเทศเพื่อการจัดการ	14
2.8 การออกแบบฐานข้อมูลด้วยวิธี 3NF	16
2.9 ONF อัลกอริทึม (The Optimal Normal Form algorithm)	21
2.10 SQL Server	22
2.11 โอเพนดาต้าเบสคอนเน็คทีวิตี (โอดีบีซี)	25
บทที่ 3 การดำเนินงานและพัฒนาระบบ	29
3.1 การดำเนินงานในปัจจุบัน	29
3.2 วัตถุประสงค์ของโครงการ	31
3.3 ขั้นตอนในการพัฒนาโครงการ	32
3.4 โครงสร้างของระบบ	33
3.5 การทำงานของระบบ	46
บทที่ 4 การใช้งาน และการทดลอง	53
4.1 ฮาร์ดแวร์และซอฟต์แวร์ที่ใช้	53
4.2 การเข้าใช้งานระบบ	54
4.3 สรุปการทดลอง	75
บทที่ 5 สรุป วิเคราะห์ และแนวทางในการพัฒนา	76

สารบัญรูปภาพ

รูป	หน้า
รูปที่ 2.1 ระบบไคลเอนต์-เซิร์ฟเวอร์	2
รูปที่ 2.2 ลักษณะของสถาปัตยกรรมไคลเอนต์ – เซิร์ฟเวอร์แบบ 2 ระดับ	3
รูปที่ 2.3 เซอร์วิสโมเดล	5
รูปที่ 2.4 ตัวอย่างสถาปัตยกรรมแบบ 3 ระดับ	6
รูปที่ 2.5 แสดงสัญลักษณ์ของชนิดเอนติตีภาควิชา	17
รูปที่ 2.6 แสดงสัญลักษณ์ของชนิดเลเวลรหัสภาควิชา	17
รูปที่ 2.7 แสดงความสัมพันธ์อ้างอิงแบบ one to one	17
รูปที่ 2.8 แสดงการเขียนความสัมพันธ์อ้างอิงแบบ one to one อย่างย่อ	17
รูปที่ 2.9 แสดงความจริงแบบ many to one	17
รูปที่ 2.10 แสดงความจริงแบบ Many to one	18
รูปที่ 2.11 แสดงการใช้ Inter fact type uniqueness constraint	18
รูปที่ 2.12 แสดงการใช้ Inter fact type uniqueness constraint	19
รูปที่ 2.13 แสดงการใช้ Equality constraint	19
รูปที่ 2.14 แสดงการใช้ Exclusion constraint	20
รูปที่ 2.15 แสดงการใช้ Subtype constraint	20
รูปที่ 2.16.1 แสดงการใช้ Mandatory constraint, lexical constraint	20
รูปที่ 2.16.2 (หมายความว่า Employee ทุกคนต้องเสียภาษี)	21
รูปที่ 2.17 ตัวอย่างแบบจำลองข้อมูล (Conceptual Schema)	22
รูปที่ 3.1 การเชื่อมต่อของระบบผ่านทางอินเทอร์เน็ต	34
รูปที่ 3.2 แสดงการทำงานของ Internet	35
รูปที่ 3.3 แสดงการทำงานของ Static Page	37
รูปที่ 3.4 แสดงการทำงานของ Dynamic Page	37
รูปที่ 3.5 แสดงการเชื่อมต่อของระบบอินเทอร์เน็ตผ่าน ODBC	38
รูปที่ 3.6 ลักษณะอื่นๆของ ISAPI	41
รูปที่ 3.7 Internet Database Connection ทำงานร่วมกับ ISAPI	42
รูปที่ 3.8 การทำงานของ ISAPI ขณะที่ติดต่อกับฐานข้อมูล	42
รูปที่ 3.9 CGI System	43
รูปที่ 3.10 รูปแบบของการ Setup Internet Database Connector	47

สารบัญรูปภาพ (ต่อ)

รูป	หน้า
รูปที่ 3.11 แสดง Data Flow Diagram ชั้นที่ 1	48
รูปที่ 3.12 แสดง Data Flow Diagram ชั้นที่ 2	49
รูปที่ 4.1 แสดงหน้าจอการรับข้อมูลผู้ใช้	54
รูปที่ 4.2 แสดงหน้าจอการรับข้อมูลผู้ใช้ไม่ครบถ้วน	55
รูปที่ 4.3 แสดงหน้าจอการรับข้อมูลผู้ใช้ไม่ถูกต้อง	56
รูปที่ 4.4 แสดงหน้าจอการรับข้อมูลผู้ใช้ถูกต้อง	57
รูปที่ 4.5 แสดงหน้าจอเมนูหลัก	58
รูปที่ 4.6 แสดงหน้าจอการรับข้อมูล เพื่อการแสดงผลข้อมูลชั้นเรียน	59
รูปที่ 4.7 หน้าจอแสดงผลการเรียกค้นข้อมูลชั้นเรียนในกรณีที่มีข้อมูลหลายแถว	60
รูปที่ 4.8 แสดงหน้าจอผลลัพธ์การเรียกค้นข้อมูลชั้นเรียนในกรณีที่ข้อมูลที่พบมีเพียง 1 แถว	61
รูปที่ 4.9 แสดงหน้าจอการเรียกค้นข้อมูลชั้นเรียน ในกรณีที่ไม่พบข้อมูลที่ต้องการ	62
รูปที่ 4.10 หน้าจอการทำงานที่ 2 การแสดงผลรายละเอียดวิชาเรียนตามรหัสวิชา	63
รูปที่ 4.11 หน้าจอการทำงานที่ 2 การแสดงผลรายละเอียดวิชาเรียนตามชื่อวิชา	64
รูปที่ 4.12 หน้าจอผลลัพธ์ของการเรียกค้นข้อมูลรายละเอียดชั้นเรียนตามชื่อวิชา Compiler	65
รูปที่ 4.13 หน้าจอผลลัพธ์ของการเรียกค้นข้อมูลรายละเอียดชั้นเรียนตามรหัสวิชา 01073110	66
รูปที่ 4.14 หน้าจอผลลัพธ์ของการเรียกค้นข้อมูลรายละเอียดชั้นเรียนในกรณีที่ไม่พบข้อมูล	67
รูปที่ 4.15 แสดงหน้าจอการทำงานที่ 3 การเรียกค้นข้อมูลวิชาเรียน	68
รูปที่ 4.16 แสดงหน้าจอผลลัพธ์การเรียกค้นข้อมูลวิชาเรียนในกรณีที่ผู้ใช้ไม่ได้ป้อนรหัสวิชา	69
รูปที่ 4.17 แสดงหน้าจอผลลัพธ์การเรียกค้นข้อมูลวิชาเรียน ในกรณีที่ผู้ใช้ป้อนข้อมูลเพียงบางส่วน	70
รูปที่ 4.18 แสดงหน้าจอผลลัพธ์การเรียกค้นข้อมูลวิชาเรียน ในกรณีที่ผู้ใช้ป้อนข้อมูลครบถ้วน	71
รูปที่ 4.19 แสดงหน้าจอผลลัพธ์การเรียกค้นข้อมูลวิชาเรียน ในกรณีที่ผู้ใช้ป้อนข้อมูลไม่ถูกต้อง	72
รูปที่ 4.20 แสดงหน้าจอการทำงานที่ 4 การแสดงผลตารางเรียน	73
รูปที่ 4.21 แสดงหน้าจอผลลัพธ์การเรียกค้นข้อมูลตารางเรียน ของนักศึกษา	74
รูปที่ 4.22 แสดงหน้าจอเตือนการป้อนข้อมูลนักศึกษาไม่ถูกต้อง	75

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง	9
ตารางที่ 2.2 ความสัมพันธ์แบบหนึ่ง ต่อ กลุ่ม	10
ตารางที่ 2.3 ความสัมพันธ์แบบกลุ่ม ต่อ กลุ่ม	10
ตารางที่ 2.4a อธิบายกฎความคงสภาพอ้างอิง	12
ตารางที่ 2.4b อธิบายกฎความคงสภาพอ้างอิง	12
ตารางที่ 2.5 อธิบายฟังก์ชันการขึ้นต่อกัน	13
ตารางที่ 2.6 อธิบายฟังก์ชันการขึ้นต่อกันอย่างเต็มที่	13
ตารางที่ 3.1 แสดงตัวอย่างของ URL ที่แตกต่างกัน	36
ตารางที่ 3.2 แสดงตัวอย่างของการร้องขอแบบต่างๆ	36
ตารางที่ 3.3 เป็นตัวอย่างของนามสกุลเทียบกับตัว Interpreter	45
ตารางที่ 3.4 Room Table	50
ตารางที่ 3.5 Building Table	50
ตารางที่ 3.6 Class Table	50
ตารางที่ 3.7 Subject Table	51
ตารางที่ 3.8 Teacher Table	51
ตารางที่ 3.9 TeacherClass Table	52

บทที่ 1

บทนำ

งานทะเบียนและเป็นงานหลักที่สำคัญในด้านการบริการทางการศึกษา การทำงานในส่วนของงานทะเบียนนั้นมีหลายส่วน เช่น การเก็บประวัตินักศึกษา , การจัดตารางเรียน – ตารางสอน การลงทะเบียน , การออกเกรด , การออกใบทรานสคริปต์ เป็นต้น ซึ่งในปัจจุบันสถาบันการศึกษาหลาย ๆ แห่งมักจะประสบปัญหาเกี่ยวกับความล่าช้า ความผิดพลาดของการทำงานในส่วนของงานทะเบียนอยู่เสมอ ดังนั้นการนำเอาเทคโนโลยีสารสนเทศเข้ามาใช้ในส่วนของงานทะเบียนนับว่ามีความจำเป็นอย่างมากเพราะสามารถเพิ่มประสิทธิภาพในการทำงานในส่วนทะเบียนได้ ที่เห็นได้ชัดเจน เช่น ลดจำนวนบุคลากรลงไปได้มาก , ลดความผิดพลาดที่เกิดขึ้น , ให้บริการได้มากในช่วงเวลาที่เท่ากัน และ สามารถลดข้อจำกัดบางอย่างของมนุษย์ได้ แต่ในบางครั้งผู้ขอใช้บริการอาจไม่ได้รับความสะดวกจากการให้บริการ เนื่องจากจำนวนเครื่องคอมพิวเตอร์ที่ให้บริการมีไม่เพียงพอ ดังนั้นโครงการนี้จึงได้นำเอาระบบอินเทอร์เน็ตเข้ามาช่วย ในโครงการนี้จะเป็นส่วน ที่ช่วยในการจัดตารางเรียน ตารางการใช้ห้องเรียน และรายละเอียดของแต่ละรายวิชา ซึ่งเป็นส่วนแยกย่อยมาจากส่วนหนึ่งของงานทะเบียน

เมื่อโครงการนี้เสร็จจะช่วยอำนวยความสะดวกแก่ผู้ขอใช้บริการ โดยผู้ขอใช้บริการสามารถสอบถามข้อมูลต่างๆ ที่ต้องการรู้ผ่าน Web page โดยโปรแกรม WEB BROWSER ที่มีให้เลือกมากมาย อีกทั้งผู้ขอใช้บริการไม่จำเป็นต้องรู้ถึงรายละเอียดหรือยุ่งเกี่ยวกับคำสั่ง SQL ทำให้ผู้ขอใช้บริการมีความสะดวกในการค้นหาข้อมูลที่ต้องการได้ง่ายและมีขอบเขตมากขึ้นกว่าเดิม

ในปฏิญานิพนธ์ฉบับนี้ จะมีหลักการและทฤษฎีต่างๆที่ใช้ในการพัฒนาโครงการ ซึ่งจะอธิบายถึงรายละเอียดในส่วนต่างๆ เพื่อที่จะทำให้มองเห็นวิธีการ และแนวทางในการพัฒนาระบบ จากนั้นจะเป็นส่วนของแนวทางการพัฒนาระบบ จากนั้นจะเป็นการอธิบายวิธีการใช้งาน และผลที่ได้จากการทดลองใช้งาน และสุดท้ายจะเป็นการสรุปผลงาน และแนวทางการพัฒนาต่อไป ทางผู้จัดทำได้พยายามจัดทำปฏิญานิพนธ์ฉบับนี้ ให้สามารถอ่านได้เข้าใจง่าย แต่เนื่องจากปฏิญานิพนธ์ฉบับนี้ได้เกี่ยวกับงานทางด้านเทคโนโลยีที่สมัยใหม่ จึงทำให้บางครั้งจำเป็นต้องมีการใช้ศัพท์ทางเทคนิคประกอบด้วย แต่ทางคณะผู้จัดทำก็พยายามอธิบายความหมายของคำนั้นๆ ไว้เป็นที่เรียบร้อยแล้ว

สุดท้ายนี้ ทางคณะผู้จัดทำหวังว่าปฏิญานิพนธ์ฉบับนี้จะเป็นประโยชน์ต่อผู้อื่นบ้างไม่มากก็น้อย และหากมีความผิดพลาดประการใด ทางคณะผู้จัดทำก็ขออภัยไว้ ณ ที่นี้ด้วย

บทที่ 2

ทฤษฎีและหลักการ

2.1 ไคลเอนต์ – เซิร์ฟเวอร์ (Client – Server)

เป็นเทคนิคในการแบ่งงานๆ หนึ่งออกเป็น 2 กระบวนการ (Process) หรือหลายๆ กระบวนการ ในทางลอจิกคอล (Logical)

ไคลเอนต์ (Client Process) จะมีการร้องขอ เซิร์ฟเวอร์โปรเซส (Server Process) ให้การทำงานในบางส่วนของมัน การกระทำกระบวนการดังกล่าวจะเรียกว่า โคออปอเรทีฟโปรเซสซิง (Cooperative Processing) เนื่องจากโปรเซสที่ถูกแบ่งทั้งสองส่วนจะทำงานร่วมกัน

แต่ละโปรเซสอาจจะวิ่ง (Run) อยู่บน ฟิสิกส์คอลแมคชีน (Physical Machine) เดียวกันหรือคนละ ฟิสิกส์คอลแมคชีนก็ได้ กล่าวคือ โปรเซสของไคลเอนต์ – เซิร์ฟเวอร์ อาจจะวิ่งอยู่บนเครื่องคอมพิวเตอร์ส่วนบุคคลเพียงเครื่องเดียว หรือเครื่องอื่นๆ ที่ห่างออกไปเป็นพันไมล์ก็ได้



รูปที่ 2.1 ระบบไคลเอนต์-เซิร์ฟเวอร์

ตัวอย่างของระบบไคลเอนต์ – เซิร์ฟเวอร์ ดังรูป ซึ่งเป็นระบบพริ้นเซิร์ฟเวอร์ (Print Server) เมื่อเครื่องที่เป็นไคลเอนต์ต้องการพิมพ์งานออกจากเครื่องพิมพ์ที่อยู่ห่างออกไป ซึ่งมันจะส่งเพิ่มข้อมูลที่ต้องการจะพิมพ์ไปยังเครื่องเซิร์ฟเวอร์ และเครื่องเซิร์ฟเวอร์ก็จะส่งงานนั้นต่อไปยังเครื่องพิมพ์ เมื่องานเสร็จเรียบร้อยแล้วเครื่องเซิร์ฟเวอร์ก็จะส่งสัญญาณกลับมาที่เครื่องไคลเอนต์นั้น เพื่อบอกว่างานที่ให้พิมพ์เสร็จสมบูรณ์แล้ว

ข้อดีของระบบไคลเอนต์-เซิร์ฟเวอร์ มีดังนี้

- ช่วงเวลาในการร้องขอของเครื่องไคลเอนต์จะขึ้นอยู่กับประสิทธิภาพของเครื่องเซิร์ฟเวอร์ ซึ่งถ้าเกิดเครื่องเซิร์ฟเวอร์มีประสิทธิภาพและความเร็วสูงก็จะสามารถตอบสนองต่อการร้องขอของเครื่องไคลเอนต์ได้อย่างรวดเร็ว
- เซิร์ฟเวอร์โปรเซสจะส่งเฉพาะผลลัพธ์ไคลเอนต์ต้องการจริงๆ เท่านั้น โดยปกติผลลัพธ์จะมีขนาดเล็กซึ่งจะช่วยลดปริมาณข้อมูลในเครือข่าย
- เซิร์ฟเวอร์โปรเซสจะสามารถจัดการและควบคุมดูแลความถูกต้องของข้อมูลได้ซึ่งมันจะมีสิ่งๆที่ช่วยในการจัดเก็บเกี่ยวกับความถูกต้องของข้อมูลได้ เช่น การใช้ทรานแซกชัน (Transaction) , การทำโรลแบค (Roll – backs), การใช้ล็อกไฟล์ (Log files) เป็นต้น

2.2 สถาปัตยกรรมไคลเอนต์ – เซิร์ฟเวอร์ แบบ 2 ระดับ (Two – Tier Client – Server Architecture)

จะมีการแบ่งออกเป็น 2 ระดับ ในทางลจจกคคคค คือ จะแบ่งเป็น ฟรอนต์เอนด์ (Front – end) และแบคเอนด์ (Back–end)



รูปที่ 2.2 ลักษณะของสถาปัตยกรรมไคลเอนต์ – เซิร์ฟเวอร์แบบ 2 ระดับ

ข้อจำกัดของสถาปัตยกรรมไคลเอนต์ - เซิร์ฟเวอร์ แบบ 2 ระดับ ได้แก่

- ลำบากต่อการบริหารฟรอนต์เอนด์ที่เพิ่มขึ้น (Difficult to Manage Front – End Enhancements) กล่าวคือ ถ้าโปรแกรมประยุกต์ (Applications) เป็นแบบฟรอนต์เอนด์ หลายระดับจะมีผลกระทบ ซึ่งผลกระทบดังกล่าวทำให้การวิเคราะห์และควบคุมการเปลี่ยนแปลงที่เป็นอิสระต่อกันได้ลำบาก
- ลำบากต่อการใช้ร่วมกันของกระบวนการที่ใช้กันบ่อย (Difficult to Share Common Processes) หากเป็นสถาปัตยกรรมไคลเอนต์ – เซิร์ฟเวอร์ แบบ 2 ระดับ โปรเซสที่ฝังอยู่ในโปรแกรมประยุกต์จะยากต่อการนำมาใช้กับโปรแกรมประยุกต์อื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รักษาความปลอดภัยของข้อมูลได้ลำบาก (Difficult to Secure) ในการรักษาความปลอดภัยของข้อมูลจะมีทั้งในส่วนของ ฟรอนต์เอนด์ และ แบคเอนด์
 - ส่วนฟรอนต์เอนด์ ก็จะเป็นระบบรักษาความปลอดภัยในโปรแกรมประยุกต์เอง คือ อาจจะสามารถใช้โปรแกรมประยุกต์ตัวอื่น เช่น Microsoft Excel ในการเข้าถึงฐานข้อมูล (Database) แทนได้
 - ส่วนแบคเอนด์ ก็จะเป็นระบบรักษาความปลอดภัยของฐานข้อมูล คือ ผู้ใช้แต่ละคนจะมีสิทธิ์ในการเข้าถึงข้อมูลในแต่ละส่วนแตกต่างกัน ดังนั้นแทบจะเป็นไปไม่ได้เลยที่จะจัดกลุ่มของผู้ใช้แล้วให้สิทธิ์ที่เหมาะสมกับผู้ใช้แต่ละคน ทั้งนี้ หากให้ส่วนของการจัดการความปลอดภัยอยู่ทั้งส่วนฟรอนต์เอนด์ และแบคเอนด์ก็ จะทำให้การบริหารความปลอดภัยเป็นไปได้อย่างยาก
- การพัฒนาส่วนฟรอนต์เอนด์เป็นไปได้อย่างลำบากเนื่องจากภายในฟรอนต์เอนด์จะมีส่วนของบิสซิเนสรูลส์ (Business Rules) ที่สำคัญๆ เป็นจำนวนมากรวมอยู่ด้วย
- ในการทำงานของระบบฐานข้อมูลในแง่ธุรกิจ ส่วนบิสซิเนสรูลส์จะแยกอยู่ทั้งใน ส่วนของฟรอนต์เอนด์ และ แบคเอนด์ ทำให้การบริหารระบบฐานข้อมูลเพื่อให้ได้ ประสิทธิภาพและผลประโยชน์สูงสุดเป็นไปได้อย่างจำกัด

2.3 สถาปัตยกรรมแบบ 3 ระดับ (Three – tier Architecture)

ระดับที่เพิ่มขึ้นมา จะเป็นส่วนที่รวบรวมบิสซิเนสรูลส์ ซึ่งจะเป็นชั้นตรงกลางอยู่ระหว่าง ฟรอนต์เอนด์ และ แบคเอนด์ และจะมีโค้ด (Code) ที่แยกจากส่วนโปรแกรมประยุกต์และฐานข้อมูลเลย

ในการติดต่อสื่อสารหรือการส่งสัญญาณร้องขอจะสามารถทำได้ในระดับเดียวกัน, ระดับที่สูงกว่า หรือระดับที่ต่ำกว่าได้ จะเห็นได้ว่าส่วนของโปรแกรมประยุกต์นั้นจะ ไม่สามารถติดต่อโดยตรงกับ ส่วนคาล์เซอรัวิสได้

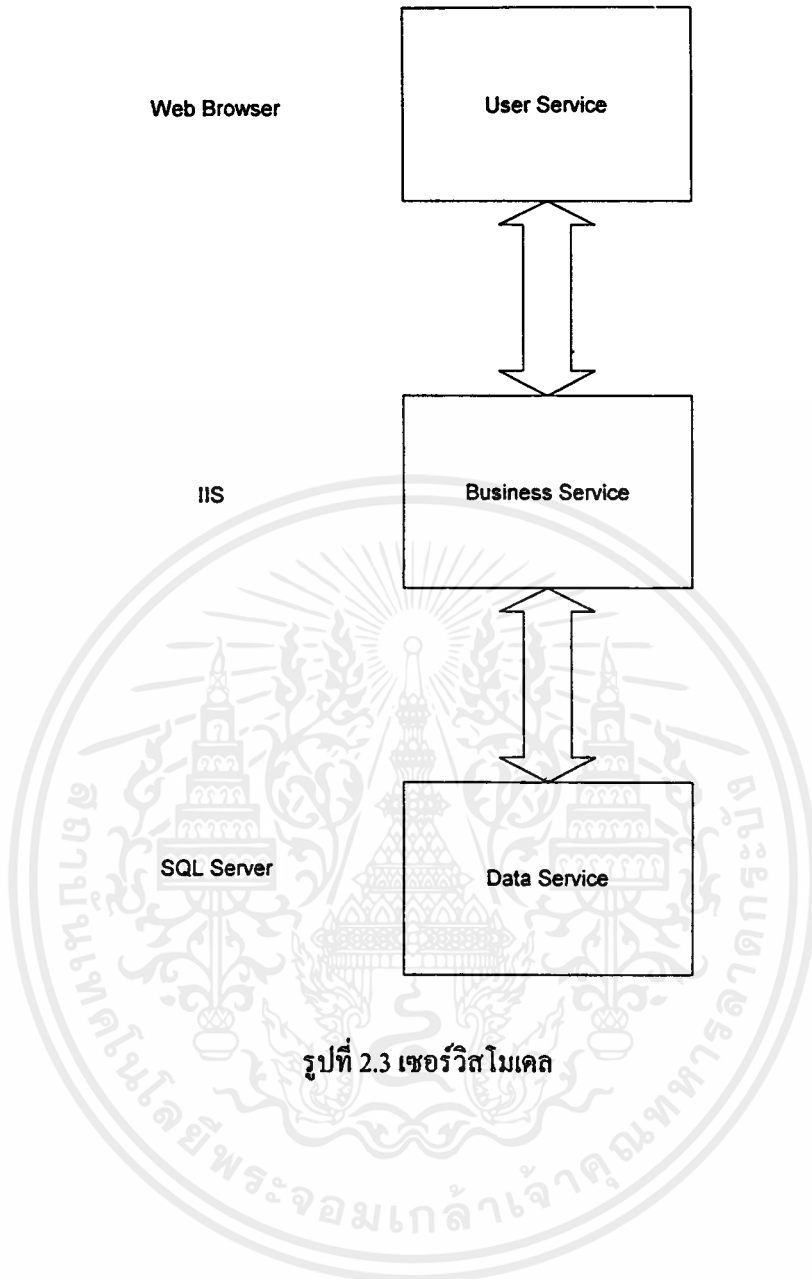
ในส่วนของยูสเซอร์เซอรัวิส ฟังก์ชันในโปรแกรมประยุกต์จะเป็นส่วนที่ติดต่อกับบิสซิเนสเซอรัวิส ซึ่งในการออกแบบโปรแกรมประยุกต์ที่ดีนั้น ควรจะมีลักษณะดังนี้

- สามารถเข้าใจในส่วนของบิสซิเนสเซอรัวิสทั้งหมดได้
- สามารถเรียกใช้บิสซิเนสเซอรัวิสได้อย่างมีประสิทธิภาพ

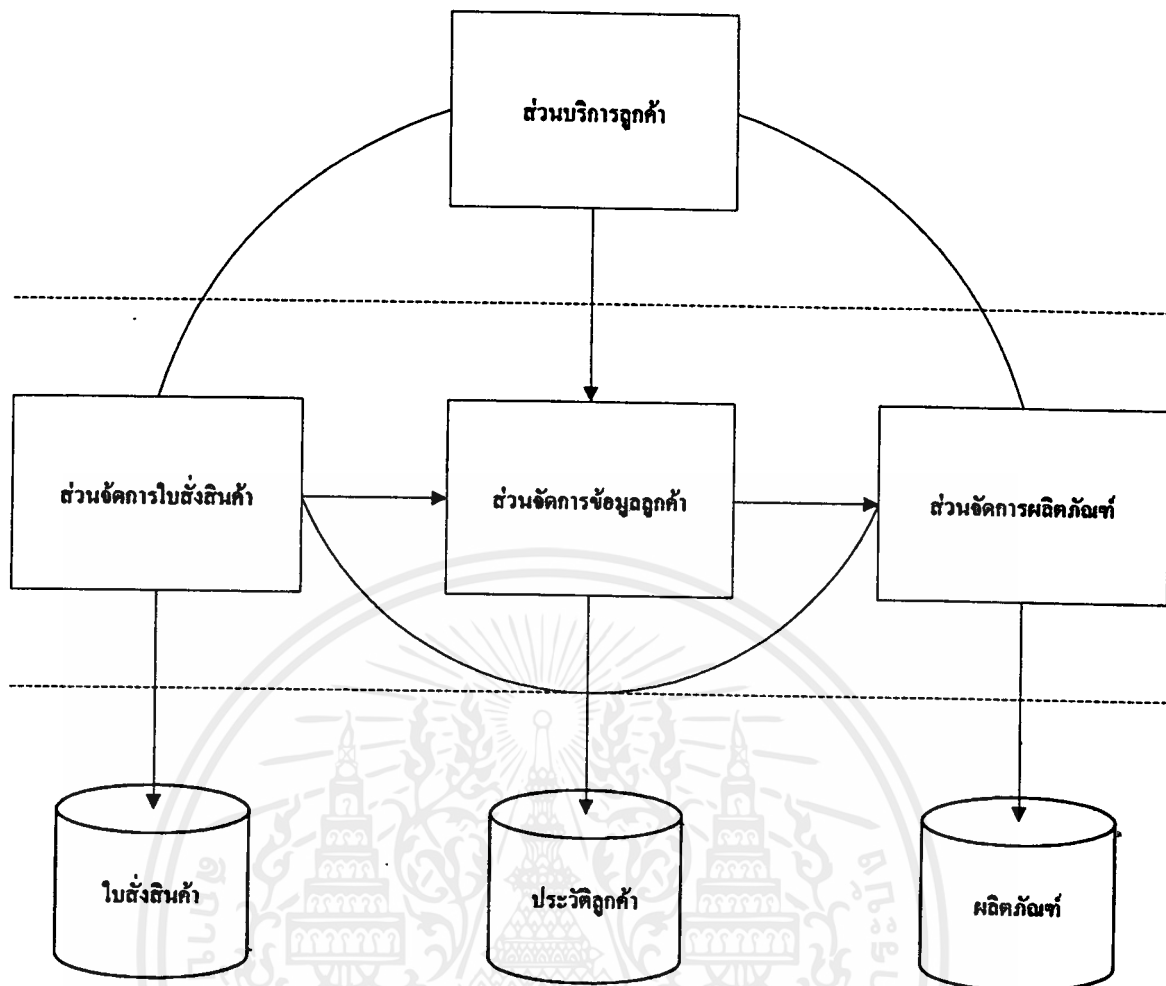
โปรแกรมประยุกต์ส่วนใหญ่จะเขียนด้วยภาษา 4GLs, Visual Basic หรือโปรแกรมประยุกต์ที่มี อยู่แล้ว เช่น Microsoft Excel เป็นต้น

ในส่วนของบิสซิเนสเซอรัวิส จะมีบิสซิเนสรูลส์ในการควบคุมและตรวจสอบความถูกต้องของ ข้อมูลในฐานข้อมูลและนอกจากนี้จะช่วยในการลดความซ้ำซ้อนของข้อมูลได้อีกด้วย

บิสซิเนสเซอรัวิสมักจะเขียนด้วยภาษา Visual Basic, COBOL และ C เป็นต้น ส่วนสุดท้ายนั่นก็คือ ส่วนของคาล์เซอรัวิส ซึ่งจะทำหน้าที่ในการจัดการข้อมูลดิบและทำงานตามที่บิสซิเนสเซอรัฟเวอร์ร้องขอ พร้อมทั้งจะควบคุมและตรวจสอบความถูกต้องของข้อมูลที่มีเก็บไว้



รูปที่ 2.3 เซอร์วิส โมเดล



รูปที่ 2.4 ตัวอย่างสถาปัตยกรรมแบบ 3 ระดับ

จากรูป 2.4 จะเป็นตัวอย่างที่แสดงถึงสถาปัตยกรรมแบบ 3 ระดับ โดยมีการทำงานคร่าวๆ ดังนี้

- เมื่อพนักงานได้กรอกการสั่งสินค้าของลูกค้าลงในส่วนของโปรแกรมประยุกต์ระดับผู้ใช้ ต่อมาโปรแกรมประยุกต์ดังกล่าวจะส่งข้อมูลต่างๆ ของใบสั่งสินค้านั้นไปยังส่วนจัดการใบสั่งสินค้าในชั้นบิสซิเนสเซอร์ฟเวอร์
- ส่วนจัดการใบสั่งสินค้าก็จะตรวจสอบข้อมูลใบสั่งสินค้าว่าครบถ้วนถูกต้องหรือไม่ โดยจะทำการติดต่อไปยังส่วนจัดการข้อมูลลูกค้า และส่วนจัดการผลิตภัณฑ์ ซึ่งถ้าทุกอย่างถูกต้องแล้วส่วนจัดการใบสั่งสินค้าก็จะทำการติดต่อไปยังส่วนดาต้าเซิร์ฟเวอร์เพื่อเพิ่มข้อมูลใบสั่งสินค้านี้ลงในฐานข้อมูล
- ส่วนดาต้าเซิร์ฟเวอร์ก็จะทำการเพิ่มข้อมูลใบสั่งสินค้านี้ลงในฐานข้อมูล

ข้อได้เปรียบของสถาปัตยกรรมแบบ 3 ระดับ มีดังนี้

- สามารถจัดการเกี่ยวกับการพัฒนาระบบได้ เนื่องจากเมื่อมีการเปลี่ยนแปลงแก้ไขในระดับชั้นใดชั้นหนึ่งแล้ว จะไม่ส่งผลกระทบต่อระดับชั้นอื่น
- บิตซิเนสเชิร์ฟเวอร์สามารถแบ่งกันใช้ได้ หรือที่เรียกว่า โค้ดรียูส (Code Reuse) เนื่องจากเดิมโปรแกรมจะถูกสร้างขึ้นโดยใช้ทรัพยากรที่มีอยู่ในตัวภาษาที่ใช้สร้างโปรแกรมเอง หรือเรียกใช้ซ้ำภายในโปรแกรมที่กำลังสร้างนั้น ซึ่งการพัฒนาโปรแกรมในลักษณะดังกล่าวจะเป็นการพัฒนาโปรแกรมในระดับจุลภาค (Micro – Level Programming) ต่อมาได้มีการพัฒนาโปรแกรมโดยมีการเขียนโปรแกรมที่ถูกเรียกใช้บ่อยเก็บไว้ ซึ่งในการพัฒนาในลักษณะดังกล่าวเป็นการพัฒนาโปรแกรมในระดับมหภาค (Macro – Level Programming) ดังนั้นทำให้การสร้างโปรแกรมรวดเร็วและมีประสิทธิภาพมากขึ้น
- บิตซิเนสเชิร์ฟเวอร์สามารถจัดการเกี่ยวกับความปลอดภัยของข้อมูลได้ ระบบป้องกันความปลอดภัยจะถูกโยกย้ายมาไว้ในส่วนบิตซิเนสเชิร์ฟเวอร์ ซึ่งแต่เดิมจะถูกแยกอยู่ในส่วนของยูสเซอร์เซอวีส์ และคาค้าเซอวีส์ ดังนั้นเราจึงสามารถสร้างระบบความปลอดภัยในส่วนบิตซิเนสเชิร์ฟเวอร์นี้ให้เหมาะสมตามระดับความต้องการของแต่ละองค์กรได้
- ผู้ใช้สามารถสร้างส่วนโปรแกรมประยุกต์เป็นของตนเองได้ เนื่องจากบิตซิเนสรูลส์ที่เคออยู่ในส่วนของฟรอนต์เอนด์ เมื่อถูกย้ายไปไว้ในบิตซิเนสเชิร์ฟเวอร์แล้ว ดังนั้นผู้ใช้จึงสามารถสร้างโปรแกรมประยุกต์ของตนเองโดยที่ไม่ต้องคำนึงถึงบิตซิเนสรูลส์อีก
- บิตซิเนสเชิร์ฟเวอร์จะมีพร็อพเพอร์ตี้ (Property) และเมทอด (Method) ที่สามารถออกแบบให้บรรลุเป้าหมายของบิตซิเนสรูลส์ต่างๆ ได้
- บิตซิเนสเชิร์ฟเวอร์จะสามารถวิ่งโปรแกรมบางอย่างได้โดยอัตโนมัติ ซึ่งทำให้การทำงานเป็นไปได้อย่างมีประสิทธิภาพมากขึ้น

จะเห็นได้ว่าส่วนที่เพิ่มขึ้นมาที่เรียกว่า บิตซิเนสเชิร์ฟเวอร์มีคุณลักษณะพิเศษต่างๆ ที่เอื้ออำนวยต่อการบริหารงานทางด้านฐานข้อมูลเป็นอย่างดี ดังนั้นจึงเป็นการง่ายที่จะออกแบบงานด้านฐานข้อมูลโดยใช้สถาปัตยกรรมแบบ 3 ระดับ

2.4 โมเดลเชิงสัมพันธ์ (Relational Model)

เป็นโมเดลที่ใช้ในการอธิบายความสัมพันธ์ของข้อมูลที่ถูกจัดเก็บด้วยระบบจัดการข้อมูลเชิงสัมพันธ์ (Relational Database Management System; RDBMS) ซึ่งเป็นผลงานของ ดร.คอดด์ (Codd) มีการนำไปใช้งานกันอย่างกว้างขวางมาก โดยเฉพาะผู้ที่ปฏิบัติงานกับเครื่อง มินิคอมพิวเตอร์ หรือเมนเฟรม จะนิยมใช้โมเดลเชิงสัมพันธ์มากกว่าโมเดลแบบอื่นๆ

ข้อได้เปรียบของโมเดลเชิงสัมพันธ์

- สามารถทำความเข้าใจได้ง่ายกว่า เพราะว่าระบบการจัดเก็บข้อมูลเป็นการออกแบบในมุมมองของมนุษย์ ซึ่งจะมีความซับซ้อนกว่าโมเดลแบบอื่น
- ระบบที่ใช้โมเดลเชิงสัมพันธ์นี้ มักจะมีเครื่องมือช่วยให้ผู้ใช้สามารถจัดการกับข้อมูลได้ง่ายกว่าข้อมูลที่จัดเก็บด้วยโมเดลแบบอื่น
- โมเดลเชิงสัมพันธ์มีเครื่องมือที่ช่วยให้ผู้ใช้สามารถพบปัญหาที่เกิดขึ้น ในการออกแบบฐานข้อมูลได้ง่าย และยังสามารถแก้ไขการออกแบบที่ผิดพลาดได้ง่าย
- เป็นโมเดลที่สอดคล้องกับหลักการของฐานข้อมูล ผู้ใช้ไม่ต้องพะวงกับการจัดการรายละเอียดของการจัดเก็บข้อมูล
- ภาษาที่ใช้จัดการกับฐานข้อมูล (คือ SQL : Structure Query Language) เป็นภาษาแบบเซตโอเรียนต์ (Set Oriented) ซึ่งจะต่างกับภาษาโดยทั่วไปที่เป็นแบบ Record-at-a-time

แม้ว่าโมเดลเชิงสัมพันธ์จะมีข้อได้เปรียบกว่าโมเดลแบบอื่นๆ แต่ก็ยังมีจุดอ่อนอยู่เมื่อนำไปประยุกต์ใช้งานกับฐานข้อมูล หรือองค์กรที่มีขนาดใหญ่แล้วมักจะมีปัญหาเสมอ ที่เป็นเช่นนี้ก็เพราะว่าองค์กรขนาดใหญ่มีฐานข้อมูลที่สลับซับซ้อนมาก ทำให้การออกแบบทำได้ยาก ดังนั้นโอกาสที่จะผิดพลาดจึงเป็นไปได้สูง

2.4.1 คำศัพท์เฉพาะของโมเดลเชิงสัมพันธ์

ในหัวข้อนี้จะกล่าวถึงนิยาม คำจำกัดความ และคำศัพท์ต่างๆที่เกี่ยวข้องกับโมเดลเชิงสัมพันธ์ ที่ควรจะทราบ

ข้อมูลที่จัดเก็บโดยโมเดลเชิงสัมพันธ์ จะถูกเก็บไว้ในตารางที่เรียกว่า “ริเลชัน” โดยที่ทุกริเลชันจะถูกแสดงในรูปของตาราง แต่บางตารางอาจไม่เป็นริเลชันก็ได้

ตารางที่มีคุณสมบัติจะต้องมีลักษณะดังนี้

1. แต่ละช่องของตารางจะบรรจุข้อมูลได้เพียงค่าเดียว
2. ชื่อหัวข้อในแต่ละคอลัมน์มีเพียงค่าเดียว ซึ่งเป็นชื่อของแอตทริบิวของเอนทิตี
3. ค่าของข้อมูลในแต่ละคอลัมน์ คือ ค่าของแอตทริบิวของเอนทิตี
4. การเรียงลำดับของคอลัมน์ไม่มีความสำคัญ
5. ข้อมูลแต่ละแถวจะต้องแตกต่างกัน
6. การเรียงลำดับแถวไม่ถือว่ามีความสำคัญ

ตารางที่มีคุณสมบัติดังกล่าวจะเรียกว่า ริเลชัน ดังนั้นนิยามของฐานข้อมูลเชิงสัมพันธ์คือ ฐานข้อมูลที่เกิดจากการรวบรวมริเลชันต่างๆ ที่มีความสัมพันธ์ (Relationship) ระหว่างกัน

เราจะเรียกข้อมูลแต่ละแถวในแนวนอนว่า “ทัพเพิล” (Tuple) และเรียกข้อมูลในแนวตั้ง หรือแนว คอลัมน์ว่า “แอตทริบิวต์” (Attribute) โดยที่คำว่า “คีย์” (Key) จะหมายถึงข้อมูลที่เกิดจากแอตทริบิวต์หลายตัว หรือตัวเดียวก็ได้

แต่ละรีเลชัน (ตาราง) จะต้องมีสิ่งที่เรียกว่า “คีย์หลัก” (Primary key) คือข้อมูลของแอตทริบิวต์ 1 ตัว หรือหลายตัวก็ได้ ที่สามารถใช้เป็นตัวเจาะจงบอกเราได้ว่า กำลังอ้างถึงทัพเพิลใด

ส่วนคีย์ที่เป็นแอตทริบิวต์ของรีเลชันอื่นซึ่งซ้ำกับแอตทริบิวต์ที่เป็นคีย์หลักของอีกรีเลชันหนึ่ง จะ เรียกว่า “คีย์นอก” (Foreign key)

ในกรณีที่มีรีเลชันมีแอตทริบิวต์ หรือ กลุ่มของแอตทริบิวต์ ที่มีได้ถูกเลือกเป็นคีย์หลัก จะเรียกว่า “คีย์ คู่แข่ง” (Candidate key) หรือ “คีย์สำรอง” (Alternate key) และแอตทริบิวต์ที่เหลือที่ไม่ได้ถูกเลือกเป็นคีย์ หลัก จะถูกเรียกว่า “Nonkey Attribute”

คำว่า โดเมน (Domain) จะหมายถึงกรอบของค่าต่างๆ ที่เป็นไปได้ เช่น โดเมนของแอตทริบิวต์วันที่ ก็จะหมายถึงค่าที่จะเป็นไปได้ซึ่งก็คือ 1 ถึง 31

แต่ในการเก็บข้อมูลลงในรีเลชันนั้น บางกรณีที่เรากำหนดโดเมนให้แอตทริบิวต์แล้ว แต่ข้อมูลอาจจะบรรจุเข้าไปภายหลัง ลักษณะนี้จะทำให้เกิดค่าว่าง (Null value) ขึ้นชั่วคราว ดังนั้นคำว่าค่าว่างจะหมายถึง ค่าที่ยังไม่ทราบแน่ชัดว่าจะเป็นค่าใด โดยมีข้อสังเกตุว่า แอตทริบิวต์ที่เป็นคีย์หลักจะเป็นค่าว่างไม่ได้โดยเด็ดขาด เพราะหากว่างก็จะทำให้การเข้าถึงทัพเพิลนั้นๆกระทำไม่ได้

เมื่อมีการจัดเก็บข้อมูลใดๆแล้ว ข้อมูลจะถูกจัดแบ่งกันไปตามความสัมพันธ์ เป็นกลุ่มๆ ซึ่งจะเรียกกลุ่มข้อมูลนี้ว่า “เอนติตี้” (Entity) ซึ่งจะประกอบไปด้วยแอตทริบิวต์ที่มีความสัมพันธ์กัน

2.4.2 ความสัมพันธ์ระหว่างเอนติตี้

2.4.2.1 ความสัมพันธ์แบบหนึ่ง ต่อ หนึ่ง (One to One)

ชื่อนักเรียน	ชื่อผู้ปกครอง
A	A
B	B
C	C

ตารางที่ 2.1 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง

ความสัมพันธ์แบบหนึ่ง ต่อ หนึ่งระหว่างเอนติตี้หมายความว่า เมื่อเอนติตี้หนึ่งมีข้อมูลคีย์หลักค่าหนึ่ง ค่าของข้อมูลก็จะมีความสัมพันธ์ กับค่าข้อมูลของคีย์หลักของอีกเอนติตี้หนึ่งเพียงค่าเดียว เช่น หากเรากำหนดให้ความสัมพันธ์ระหว่างเอนติตี้ นักเรียน และผู้ปกครอง เป็นแบบหนึ่ง ต่อ หนึ่ง แล้ว ก็จะหมายความว่า การที่เราจะอ้างถึงนักเรียนคนหนึ่ง ก็จะสามารถอ้างถึงผู้ปกครองได้คนเดียวเท่านั้น

2.4.2.2 ความสัมพันธ์แบบหนึ่ง ต่อ กลุ่ม (One to Many)

ชื่อนักเรียน	ชื่อผู้ปกครอง
A	a
B	a
C	c

ตารางที่ 2.2 ความสัมพันธ์แบบหนึ่ง ต่อ กลุ่ม

หมายถึง เมื่อ เอนิตีมีข้อมูลของคีย์หลักค่าหนึ่ง สามารถที่จะอ้างอิงถึงข้อมูลที่อยู่อีกเอนิตีหนึ่งได้หลายค่า เช่น หากเรากำหนดให้ความสัมพันธ์ระหว่างเอนิตีนักเรียนกับเอนิตีผู้ปกครอง เป็นแบบ หนึ่งต่อกลุ่มแล้ว จะหมายความว่า การที่เราอ้างอิงถึงนักเรียนคนใดคนหนึ่ง ก็จะสามารถอ้างอิงถึงผู้ปกครองคนใดคนหนึ่งได้เพียงคนเดียวเท่านั้น และในทางตรงกันข้ามก็มีความหมายว่า เมื่อเราอ้างอิงถึงผู้ปกครองคนใดคนหนึ่งแล้วก็สามารถอ้างอิงถึงนักเรียนได้เพียงคนเดียวเท่านั้น แต่ผู้ปกครองที่เราอ้างอิงเป็นคนละคนกันก็จะอ้างอิงถึงนักเรียนคนเดียวกันก็เป็นได้

2.4.2.3 ความสัมพันธ์แบบกลุ่ม ต่อ กลุ่ม (Many to Many)

ชื่อนักเรียน	ชื่อผู้ปกครอง
A	a
B	a
C	c
C	d

ตารางที่ 2.3 ความสัมพันธ์แบบกลุ่ม ต่อ กลุ่ม

ความสัมพันธ์แบบกลุ่ม ต่อ กลุ่มระหว่างเอนิตีก็หมายความว่า ค่าของข้อมูลคีย์หลักของเอนิตีหนึ่งที่แตกต่างกันอาจอ้างอิงถึงค่าข้อมูลของคีย์หลักของอีกเอนิตีหนึ่งได้ค่าเดียวหรือหลายค่าก็ได้ เช่น หากเรากำหนดให้ความสัมพันธ์ระหว่างเอนิตีนักเรียนกับเอนิตีของผู้ปกครองเป็นแบบกลุ่ม ต่อ กลุ่มแล้วก็จะหมายความว่า การที่เราอ้างอิงถึงนักเรียนคนหนึ่งหรือหลายคนก็จะสามารถอ้างอิงถึงผู้ปกครองคนเดียวกันได้ และในทางสลับกัน การที่เราอ้างอิงถึงนักเรียนคนหนึ่งหรือหลายคนก็จะสามารถอ้างอิงถึงนักเรียนคนเดียวกันได้

นอกจากนี้ เอนิตีบางประเภทที่การอ้างอิงตัวมันได้อย่างสมบูรณ์นั้นจะต้องอ้างอิงเอนิตีอื่นเสมอ เช่น หากระบบฐานข้อมูลของบุคลากรในโรงเรียนมีเอนิตีของลูกนการภารโรงด้วยแล้ว การอ้างอิงสารานเป็นเอกสารทสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใดไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อิงถึงเอนทิตีนี้ได้จะต้องอ้างอิงถึงเอนทิตีนี้การการ ราวด้วยเสมอ เอนทิตีประเภทนี้เราเรียกว่า เอนทิตีชนิดอ่อน (Weak entity)

2.5 ทฤษฎีของระบบฐานข้อมูลเชิงสัมพันธ์

2.5.1 กฎของความคงสภาพ (Integrity rule)

กฎของความคงสภาพของโมเดลเชิงสัมพันธ์ (Relational model) เป็นทฤษฎีที่ช่วยยืนยันความถูกต้องของความสัมพันธ์ระหว่างข้อมูลว่า รีเลขันใดที่เป็นไปตามกฎของความคงสภาพนี้แล้วย่อมจะมีความสัมพันธ์ระหว่างข้อมูลอย่างถูกต้องอยู่ตลอดเวลา ไม่ว่ารีเลขันนั้นจะมีการเปลี่ยนแปลงแก้ไขข้อมูลไปในรูปแบบใดก็ตาม

กฎของความคงสภาพมีความหมายอยู่ 2 ลักษณะ คือ กฎความคงสภาพของเอนทิตี (Entity integrity rule) และกฎความคงสภาพของการอ้างอิง (Referential integrity rule) ดังอธิบายได้ดังนี้

2.5.1.1 กฎความคงสภาพของเอนทิตี กล่าวว่

“แอตทริบิวต์ทุกตัวที่เป็นส่วนของคีย์หลักจะไม่อนุญาตให้มีค่าว่าง” หมายความว่า คีย์หลักของทุกรีเลขันจะไม่สามารถเก็บค่าข้อมูลที่เป็นค่าว่างได้ เหตุผลของข้อกำหนดนี้ก็คือเพื่อให้การเข้าถึงข้อมูลในแถวใดๆ ของรีเลขันมีความเป็นไปได้เสมอเพราะถ้าคีย์หลักของแถวใดมีค่าข้อมูลเป็นค่าว่างแล้ว ก็จะส่งผลให้การเข้าถึงข้อมูลในแถวนั้นไม่สามารถกระทำได้อย่างแน่นอน

2.5.1.2 กฎความคงสภาพของการอ้างอิง กล่าวว่

“ถ้าเรามีรีเลขัน R_2 ซึ่งมี FK เป็นคีย์นอกที่อ้างอิงถึงคีย์หลัก PK ในรีเลขัน R_1 สำหรับทุกค่าของ FK ใน R_2 จะต้อง

- ก. มีค่าเท่ากับค่า PK ในแถวใดแถวหนึ่งในรีเลขัน R_1
- ข. มีค่าของแอตทริบิวต์ทุกตัวใน FK เป็นค่าว่าง”

หมายความว่า แอตทริบิวต์ใดๆที่เป็นคีย์หลักของรีเลขันหนึ่ง เมื่อมีการนำแอตทริบิวต์นั้นไปเป็นคีย์นอกของอีกรีเลขันหนึ่ง การเป็นคีย์นอกของแอตทริบิวต์นั้นจะต้องมีโดเมนเป็นโดเมนเดียวกันกับแอตทริบิวต์ที่เป็นคีย์หลัก ทั้งนี้ ก็เพื่อให้การนำรีเลขันของแอตทริบิวต์มาใช้งานร่วมกัน (การนำรีเลขันมา Join กัน) กระทำได้อย่างถูกต้อง คือ ทุกแอตทริบิวต์ที่เป็นคีย์นอกจะต้องมีข้อมูลซ้ำกับข้อมูลของแอตทริบิวต์อย่างแน่นอน แต่อาจมีบางค่าของข้อมูลแอตทริบิวต์ที่เป็นคีย์หลักเป็นข้อมูลไม่อยู่ในโดเมนของแอตทริบิวต์ที่เป็นคีย์นอกก็ได้ นั่นคือ โดเมนของคีย์นอกจะต้องเล็กกว่าหรือเท่ากับโดเมนของคีย์หลักเสมอ

คีย์หลักของ R1	คีย์อื่นๆของ R1
A	1
B	2
C	3
D	4

รีเลชัน R2

คีย์หลักของ R2	คีย์นอกของ R1
A	A
B	B
C	B
D	C

ตารางที่ 2.4a และ 2.4b อธิบายกฎความคงสภาพอ้างอิง

2.5.2 ฟังก์ชันการขึ้นต่อกัน (Function dependency)

ฟังก์ชันการขึ้นต่อกัน เป็นข้อกำหนดที่ช่วยให้เราเห็นความสัมพันธ์ของแอตทริบิวต์ต่าง ๆ ที่อยู่ในรีเลชัน ทั้งนี้เพราะแอตทริบิวต์ต่างๆ ที่อยู่เอนติตี้เดียวกัน จะมีความสัมพันธ์กันเองโดยที่ความสัมพันธ์นี้อาจเกี่ยวข้องหรือไม่เกี่ยวข้องกับความสัมพันธ์ที่มันมีต่อคีย์หลักของเอนติตี้นั้นก็เป็นที่ได้ ซึ่งการที่แอตทริบิวต์เหล่านี้มีความสัมพันธ์กันเองจะเป็นสิ่งที่เราต้องพิจารณาแยกออกเป็นรีเลชันย่อยๆ เพราะแอตทริบิวต์ของแต่ละรีเลชันก็ควรจะมีสัมพันธ์กันคีย์หลักของรีเลชันของตนเองเท่านั้น

กำหนดรีเลชัน R ถ้ามีแอตทริบิวต์ Y ของ R เป็นฟังก์ชันที่ขึ้นต่อแอตทริบิวต์ X ของรีเลชัน เราสามารถเขียนแทนได้ด้วยสัญลักษณ์

$R.X \rightarrow R.Y$

อ่านว่า R.X มีฟังก์ชันการขึ้นอยู่กับ R.Y

หรือ R.X มีฟังก์ชันการขึ้นอยู่กับ R.Y

หรือ R.Y ขึ้นอยู่กับ R.X

นิยาม R.X มีฟังก์ชันการขึ้นต่อกัน R.Y ก็ต่อเมื่อ ทุกค่าของแอตทริบิวต์ X ใน R จะมีค่าของข้อมูลแอตทริบิวต์ Y ใน R ได้ค่าเดียวเสมอ โดยที่แอตทริบิวต์ X และ Y อาจจะมีคีย์แบบรวม (composite key) ก็ได้

X	Y
a	1
b	2
a	1
b	1

ตารางที่ 2.5 อธิบายฟังก์ชันการขึ้นต่อกัน

นิยาม R.X มีฟังก์ชันการขึ้นต่อกันกับ R.Y อย่างเต็มที่ ก็ต่อเมื่อ R.Y และไม่ขึ้นกับข้อมูลเพียงบางส่วนของ R.X โดยที่แอตทริบิวต์ X และ Y อาจจะเป็นคีย์แบบรวมก็ได้
รีเลชัน R

X	Y
A	1
B	2
A	1
B	3

ตารางที่ 2.6 อธิบายฟังก์ชันการขึ้นต่อกันอย่างเต็มที่

2.6 การออกแบบฐานข้อมูลรวมแบบรีเลชันแนล

การจัดเก็บข้อมูลจำนวนมาก โดยใช้เทคโนโลยีฐานข้อมูลช่วยในการจัดเก็บต้องคำนึงถึงวิธีการออกแบบระบบฐานข้อมูล และระบบจัดการฐานข้อมูล (DBMS) ที่เหมาะสม

ปัจจุบันวิธีการออกแบบฐานข้อมูลที่รู้จักกันอย่างแพร่หลายได้แก่ Normalization ซึ่งเป็นวิธีที่มีแนวคิดในการปรับปรุงคุณสมบัติของรีเลชันเป็นขั้นตอน อย่างมีระบบเพื่อลดโอกาสที่จะเกิดความผิดพลาดเนื่องจากการเปลี่ยนแปลงข้อมูล การออกแบบฐานข้อมูลด้วยวิธีนอร์มัลไลเซชันนี้ เริ่มต้นด้วยการกำหนดยูนิเวอร์แซลรีเลชัน (Universal Relation) พร้อมทั้งกำหนดความสัมพันธ์ระหว่างแอตทริบิวต์ (Attribute) ต่างๆ ของยูนิเวอร์แซลรีเลชันนั้นในรูปของฟังก์ชันแนล ดีเพนเดนซ์ซี (Functional Dependency) มัลติแวลู ดีเพนเดนซ์ซี (Multivalued Dependency) และจอยน์ ดีเพนเดนซ์ซี (Join Dependency) แล้วทำให้ รีเลชันในแต่ละขั้นตอนมีคุณสมบัติตรงตามที่กำหนดไว้ ตั้งแต่ขั้นตอนที่ 1-5 ผลที่ได้ในขั้นสุดท้ายจะได้ Fifth Normal Form (5NF)

ถึงแม้วิธีนอร์มัลไลเซชันนี้จะเป็นวิธีที่มีขั้นตอนที่เป็นระบบ แต่ก็ยังเป็นวิธีที่ค่อนข้างยุ่งยากและซับซ้อน โดยเฉพาะอย่างยิ่งเมื่อระบบงานมีขนาดใหญ่ แอทริบิวต์มีจำนวนมาก การออกแบบด้วยวิธีนอร์มัลไลเซชันจึงเป็นเรื่องลำบาก

ขอเสนอการออกแบบระบบฐานข้อมูลรวมแบบรีเลชันแนล (Integrated Relation Database System) โดยใช้ในแอมเป็นเครื่องมือในการออกแบบ เนื่องจากแนวคิดที่ให้ Conceptual Schema มีพื้นฐานมาจากโครงสร้างภาษาธรรมชาติ ใช้รูปประโยคที่มีประธาน กริยา กรรม วิธีแสดงรูปแบบความสัมพันธ์ของข้อมูล เป็นแบบจำลองที่มีความหมาย และมีเครื่องหมายแสดงความสัมพันธ์ของข้อมูล และข้อจำกัดของข้อมูลได้อย่างชัดเจน นอกจากนี้ ยังสามารถแปลง Conceptual Schema เป็น Relation Database Schema ซึ่งจะอยู่ในรูปของ Fifth Normal Form และเนื่องจากวิธีการนี้ใช้รูปสัญลักษณ์ที่แสดงความสัมพันธ์ของข้อมูลและง่ายต่อการเข้าใจ ดังนั้นจึงสะดวกในการออกแบบระบบฐานข้อมูลของระบบงานใหญ่

เนื่องจากเป็นที่ยอมรับกันโดยทั่วไปแล้วว่า ระบบฐานข้อมูลรีเลชันแนลเป็นเครื่องมือที่เหมาะสมในการช่วยพัฒนาระบบสารสนเทศ โดยเฉพาะอย่างยิ่งระบบสารสนเทศที่มีจุดประสงค์ ในการสนับสนุนการจัดการ (Management Information System: MIS) ดังนั้นการออกแบบฐานข้อมูลรวม สำหรับองค์กรจึงเป็นสิ่งสำคัญยิ่ง เพราะฐานข้อมูลที่ได้จะเป็นโครงสร้างข้อมูลหลัก สำหรับรองรับระบบสารสนเทศ ดังกล่าว

2.7 ระบบสารสนเทศเพื่อการจัดการ

ในปัจจุบันฐานข้อมูลส่วนใหญ่ถูกออกแบบมาเพียงเพื่อช่วยงานในระดับปฏิบัติการให้สำเร็จเป็นระบบงานอิสระเท่านั้น การที่ฐานข้อมูลจะสามารถรองรับงานในระดับบริหารได้นั้น ต้องมีองค์ประกอบหลายอย่าง เนื่องจากผู้บริหารใช้ข้อมูลสนเทศในการบริหาร ทั้งแบบที่เป็นรายงานประจำที่ควรรับทราบ และข้อมูลที่ต้องการ ในขณะที่ขณะหนึ่ง โดยไม่ได้เตรียมรายงานได้ก่อนเพื่อใช้ในการตัดสินใจได้ทันเหตุการณ์ การจะตอบสนองความต้องการด้านข้อมูล แก่ผู้บริหารได้อย่างรวดเร็ว และถูกต้อง ระบบฐานข้อมูลดังกล่าว จะต้องมีคุณสมบัติต่อไปนี้

2.7.1 ข้อมูลที่มีความซ้ำซ้อนน้อยที่สุด

กล่าวคือข้อมูลจะต้องไม่มีการเก็บซ้ำซ้อน ซึ่งอาจจะทำให้เกิดความขัดแย้งของข้อมูล อันเนื่องมาจากการเปลี่ยนแปลงข้อมูลเฉพาะที่ คุณสมบัติในข้อนี้คืออาศัยการออกแบบฐานข้อมูลที่ดี ดังนั้นการเลือกใช้วิธีในแอมในการออกแบบฐานข้อมูล ซึ่งเป็นประโยชน์อย่างมาก เพราะเมื่อแปลงรูปแบบจำลองเป็นรีเลชันก็จะได้รีเลชันอยู่ในรูปของ 5NF ในทันที

2.7.2 มีการจัดระบบคลังข้อมูล

ผู้รับผิดชอบฐานข้อมูลจะต้องทราบว่าข้อมูลใดถูกจัดเก็บไว้ในตารางชื่ออะไร เพื่อที่ว่าเมื่อมีคำถามจากผู้บริหารจะได้หาข้อมูลเพื่อตอบคำถามได้ในทันทีด้วยภาษาฐานข้อมูล เมื่อกำหนดถึงวัตถุประสงค์นี้ร่วมกับเรื่องของการลดความซ้ำซ้อน ตารางข้อมูลที่ถูกสร้างขึ้นจะต้องได้รับการออกแบบเป็นอย่างดี และการสร้างตารางจะทำได้ โดยกลุ่มผู้รับผิดชอบฐานข้อมูลเท่านั้น ผู้ใช้ในระดับปฏิบัติการไม่ควรมีสิทธิแก้ไขข้อมูล และควรมีสิทธิเพียงเรียกใช้ข้อมูล เพื่อที่จะได้ให้สิทธิที่ถูกต้องบนตาราง หรือส่วนของตารางที่สมควรให้ใช้ หลักการนี้เรียกว่า หลักการระบบคลังข้อมูล

ในระบบจัดการฐานข้อมูลปัจจุบัน ส่วนมากจะมีพจนานุกรมข้อมูล (Data Dictionary) หรือที่บางระบบเรียกว่า System Catalog ที่ประกอบด้วยตาราง ซึ่งเก็บรายละเอียดของสิ่งที่มีความสำคัญต่อ ฐานข้อมูล เช่น รายละเอียดเกี่ยวกับผู้ใช้ ตาราง วิว อินเด็กซ์ เป็นต้น

การแก้ไขพจนานุกรมข้อมูลนี้ระบบจัดการฐานข้อมูล จะจัดการเองโดยอัตโนมัติ เมื่อมีการเปลี่ยนแปลงใดๆ ของตารางและผู้ใช้ขึ้นนับว่าเป็นเครื่องมือที่มีประโยชน์มากต่อผู้บริหาร

2.7.3 สามารถตอบคำถามเฉพาะกิจ

ปัจจุบันระบบจัดการฐานข้อมูลจะใช้ภาษาฐานข้อมูลในการจัดการข้อมูล ภาษาฐานข้อมูลดังกล่าวมีหลายภาษา เช่น SQL, QUEL ซึ่งเป็นภาษาที่เข้าใจง่าย ใช้เขียนโปรแกรมในการเรียกดูข้อมูลในตารางหนึ่ง หรือหลายตารางได้อย่างรวดเร็ว ผู้ใช้ที่เชี่ยวชาญสามารถตอบคำถามที่เกิดขึ้นในขณะใดขณะหนึ่งได้ โดยไม่จำเป็นต้องเขียนโปรแกรมภาษาอื่น ซึ่งจะเสียเวลานานมากจนไม่ทันต่อความต้องการใช้ข้อมูล นอกจากนี้ยังประหยัดงบประมาณ และเวลาในการจัดทำโปรแกรมในภาษาอื่นที่จะใช้ในการตอบคำถาม

2.7.4 ความปลอดภัยของข้อมูลสูง

ข้อนี้มีความสำคัญมาก เพราะหากทุกคนสามารถเรียกดู และเปลี่ยนแปลงข้อมูลได้ อาจเกิดความเสียหายต่อข้อมูล ข้อมูลบางส่วนของไม่อาจเปิดเผยได้ หรือข้อมูลเฉพาะผู้บริหาร หากไม่มีการจัดการด้านความปลอดภัยของข้อมูล ฐานข้อมูลก็ไม่สามารถใช้เก็บข้อมูลบางส่วนได้

- ดังนั้นในการจัดการฐานข้อมูลส่วนใหญ่จะมีการรักษาความปลอดภัยโดยใช้รหัสผ่านและยังมีการแบ่งระดับผู้ใช้ฐานข้อมูลออกเป็นหลายระดับอีกด้วย
- RESOURCE ผู้ที่สามารถสร้างและจัดการตารางของข้อมูลทั้งหมดในตารางของตน และสามารถอนุญาตให้ผู้อื่นใช้ตารางของตนในทางใดๆ
- CONNECT ผู้ที่สามารถเรียกดู เพิ่มเติม แก้ไข ลบข้อมูลบางส่วนที่ได้รับอนุญาตให้ใช้ผู้ออกแบบฐานข้อมูลสามารถใช้วิว (VIEW) เพื่อประโยชน์ในการรักษาความปลอดภัยได้เป็นอย่างดี โดยการสร้างวิวที่ทำตัวเสมือนเป็นตารางของผู้ใช้จริงๆ และข้อมูลที่เกี่ยวข้องกับงานของผู้ใช้เท่านั้น

2.7.5 สามารถจัดการการใช้ข้อมูลได้ในเวลาเดียวกัน (Concurrency Control)

ระบบจัดการฐานข้อมูลของระบบขนาดใหญ่มักจะเป็นระบบแบบ Multi-user ซึ่งบางหน่วยงานที่มีการใช้ข้อมูลร่วมกัน บางครั้งอาจมีความต้องการเข้าถึงข้อมูลในตารางเดียวกันหรือข้อมูลเดียวกันพร้อมๆ กัน

ดังนั้นจึงต้องมีระบบจัดการฐานข้อมูลสำหรับทำงานด้านนี้คือ Concurrency Control Mechanism โดยจะทำการล็อกข้อมูลในฐานของข้อมูล เพื่อวัตถุประสงค์ต่างกัน โดยจัดลำดับตามวัน เวลา ความสำคัญของงานดังนี้

- Shares Lock (read lock) คือ สามารถอ่านข้อมูลได้พร้อมกัน แต่ไม่สามารถเปลี่ยนแปลงข้อมูล หรือจัดรูปแบบตารางได้
- Exclusive Lock (write lock) คือ ขณะที่ผู้ใช้กำลังเปลี่ยนแปลงข้อมูล ผู้ใช้คนอื่นจะไม่สามารถใช้ข้อมูลในตารางหรือข้อมูลในแถวนั้นได้

กิจกรรมทั้งหมดนี้ระบบจัดการฐานข้อมูลเป็นผู้จัดการโดยผู้ใช้ไม่จำเป็นต้องรับรู้ ถ้าไม่ใช้ระบบจัดการฐานข้อมูล ผู้พัฒนาโปรแกรมจะต้องจัดการสิ่งเหล่านี้เองทั้งหมด

2.8 การออกแบบฐานข้อมูลด้วยวิธีในแอม

Step1: กำหนดขอบเขตของงาน (Universe of Discourse: UOD) และความจริงที่เกิดขึ้นในขอบเขตของงานที่กำหนดไว้

Step2: วาด Conceptual Schema Diagram โดยวาดคร่าวๆ จากความจริงในขอบเขตของงาน

Step3: จัดรูปของ Schema ให้เป็นระเบียบและหาชนิดความจริงที่ได้รับข้อมูลมาจากชนิดความจริงอื่น

Step4: เติมสัญลักษณ์แสดง Uniqueness constraints

Step5: ตรวจสอบความถูกต้องของชนิดความจริง

Step6: เติมสัญลักษณ์แสดง Lexical, Mandatory Role, and Subtype constraints

Step7: ตรวจสอบ Unique Identifier ของแต่ละชนิดเอนติตี้

Step8: เติมสัญลักษณ์แสดง Equality, Exclusion, and Subset constraints

Step9: ตรวจสอบความสมบูรณ์ของ Conceptual Schema ว่าสอดคล้องกับตัวอย่างข้อมูล และไม่มีความซ้ำซ้อนของข้อมูล

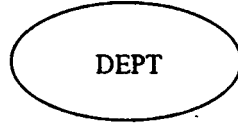
2.8.1 ส่วนประกอบพื้นฐานของในแอม

- ชนิดเอนติตี้ (Entity Type)
- ชนิดเลเบล (Label Type)
- ชนิดความจริง (Fact Type)
- ชนิดอ้างอิง (Reference Type)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ข้อจำกัดเพื่อความถูกต้องของข้อมูล (Integrity Coonstraints)

สัญลักษณ์และตัวอย่างการใช้ส่วนประกอบพื้นฐานของแบบจำลองในแอมแสดงไว้ดังรูป



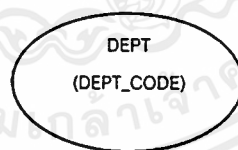
รูปที่2.5 แสดงสัญลักษณ์ของชนิดเอนติตี้ภาควิชา



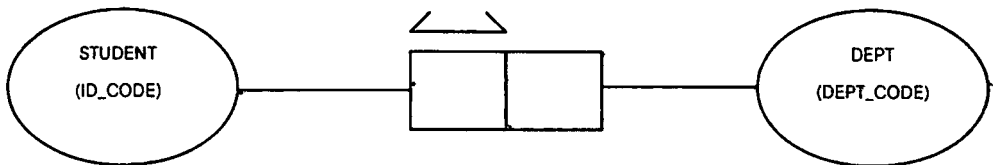
รูปที่2.6 แสดงสัญลักษณ์ของชนิดเลเบลรหัสภาควิชา



รูปที่2.7 แสดงความสัมพันธ์อ้างอิงแบบ one to one (ภาควิชาใดๆจะมีรหัสภาควิชาเพียงรหัสเดียวเท่านั้นและไม่ซ้ำกับภาควิชาอื่น)



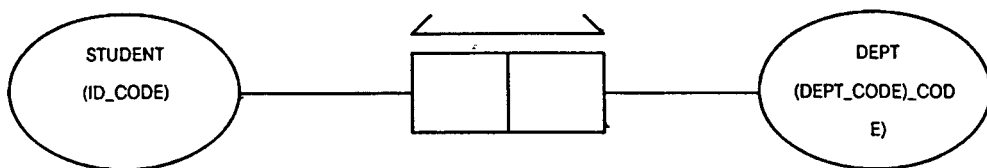
รูปที่2.8 แสดงการเขียนความสัมพันธ์อ้างอิงแบบ one to one อย่างย่อ



รูปที่2.9 แสดงความจริงแบบ many to one

(นักศึกษาคณะหนึ่งจะสังกัดภาควิชาได้เพียงภาควิชาเดียว แต่ภาควิชาใดๆ สามารถมีนักศึกษาได้มากกว่าหนึ่งคน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 แสดงความจริงแบบ Many to one

(นักศึกษาคนหนึ่งสามารถลงทะเบียนเรียนได้หลายวิชา และแต่ละวิชาที่เปิดสอนสามารถรับจำนวนนักศึกษาได้มากกว่าหนึ่งคน แต่นักศึกษาที่ลงทะเบียนเรียนวิชาใดๆ แล้วจะลงทะเบียนซ้ำวิชาเดิมไม่ได้)

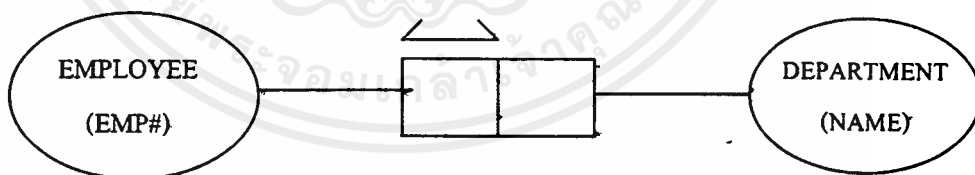
ชนิดเอนทิตีเป็นเซต (Set) ซึ่งมีสมาชิกเป็นตัวอย่างเอนทิตี (Entity Instance) เช่น ภาควิชา A (ภาควิชาวิศวกรรมโทรคมนาคม) ภาควิชา B (ภาควิชาวิศวกรรมคอมพิวเตอร์) เป็นตัวอย่างเอนทิตีของชนิดเอนทิตีภาควิชา

เครื่องหมายความสัมพันธ์ที่เป็นส่วนเชื่อมโยงระหว่างเอนทิตี และชนิดเอนทิตี หรือชนิดเลเบล นั้นเรียกว่า บทบาท (Role) จะเขียนความหมายของบทบาทนั้นไว้ภายในเหนือข้างๆ สัญลักษณ์ของมัน

การแปลงข้อมูลที่วิเคราะห์มาให้อยู่ในรูปแบบจำลอง ก่อนอื่นต้องนำข้อมูลมากำหนด เป็นชนิดเอนทิตีและเลเบลให้เรียบร้อยเสียก่อน จึงนำเอนทิตีที่ได้มาเขียนเป็นประโยค ความจริงมูลฐาน (Elementary Fact)

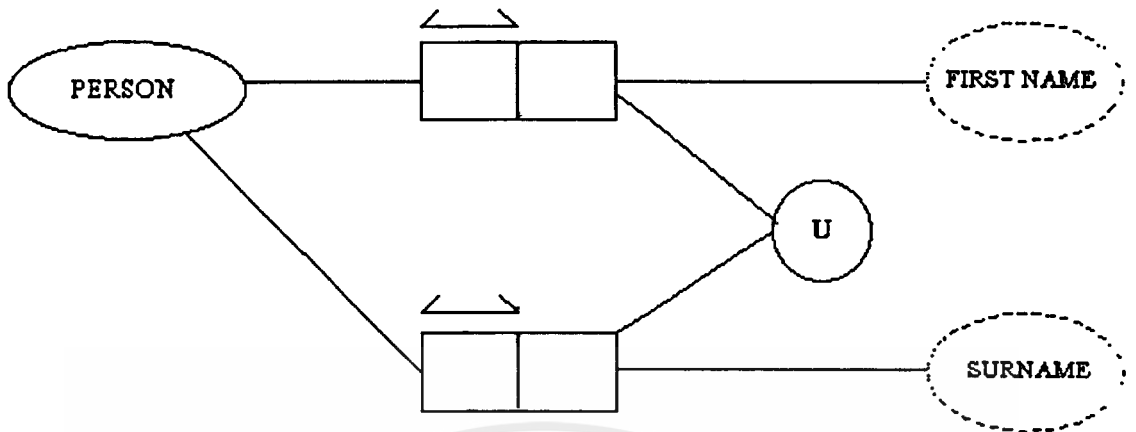
เอาความจริงที่ได้มาทั้งหมดเขียนเป็นแบบจำลอง และเติมข้อจำกัดต่างๆลงไปตามความเป็นจริงในขอบเขตของงาน

ตัวอย่างการใช้ข้อจำกัดต่างๆ แสดงไว้ดังนี้

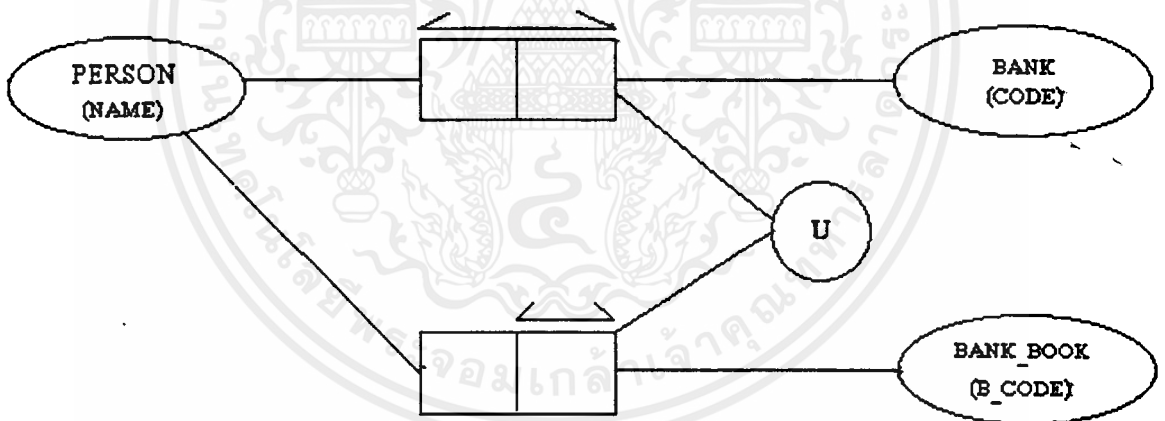


รูปที่ 2.11 แสดงการใช้ Inter fact type uniqueness constraint

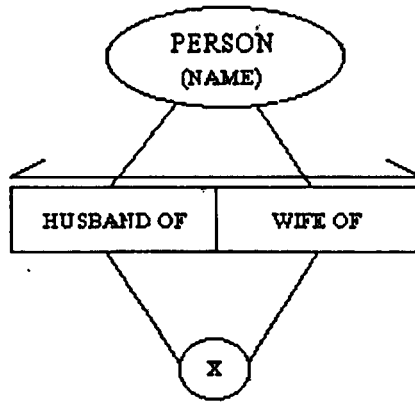
(Employee หนึ่งคนจะมีที่ทำงานได้ทีเดียวเท่านั้น)



รูปที่ 2.12 แสดงการใช้ Inter fact type uniqueness constraint
(บุคคลหนึ่งจะมีชื่อ 1 ชื่อ นามสกุล 1 นามสกุล ชื่อของบางคนอาจจะซ้ำกันและนามสกุลอาจซ้ำกัน แต่ชื่อรวมกับนามสกุลต้องไม่ซ้ำกัน)

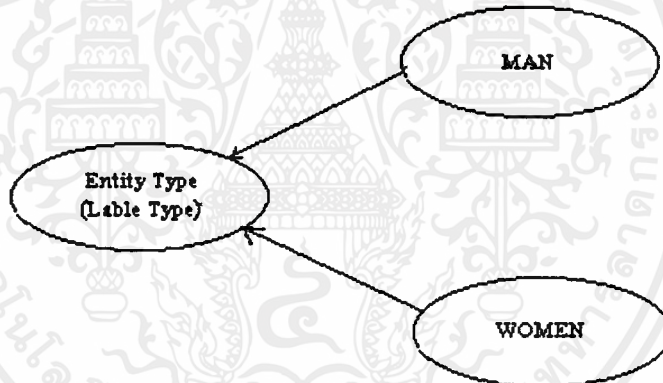


รูปที่ 2.13 แสดงการใช้ Equality constraint
(ถ้าบุคคลหนึ่งเป็นลูกค้าของธนาคารใดแล้ว บุคคลนั้นต้องมีสมุดบัญชีของธนาคารนั้นด้วย หรือในทางกลับกัน ถ้าบุคคลใดมีสมุดบัญชีของธนาคารใดแล้วก็ต้องเป็นลูกค้าของธนาคารนั้นด้วย)



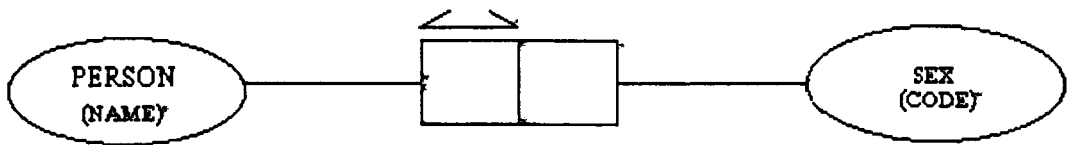
รูปที่ 2.14 แสดงการใช้ Exclusion constraint

(บุคคลใดเป็นภรรยาของอีกบุคคลหนึ่งแล้ว บุคคลนั้นต้องไม่เป็นสามีของบุคคลใดๆ ในทางกลับกัน บุคคลที่เป็นสามีของบุคคลหนึ่งแล้ว ต้องไม่เป็นภรรยาของบุคคลใดด้วย)



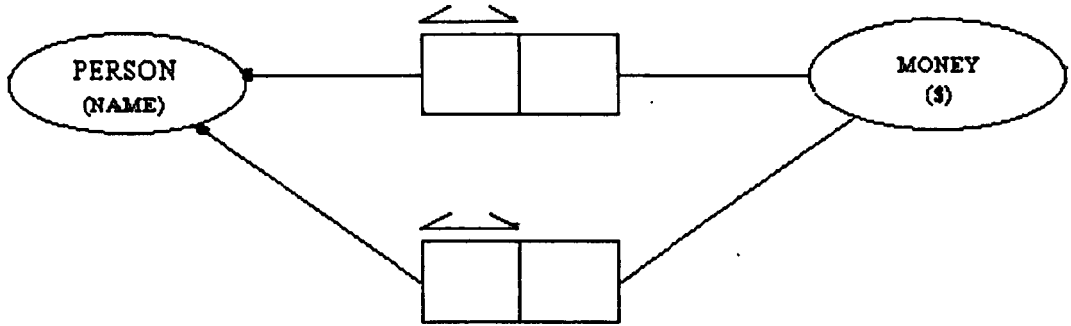
รูปที่ 2.15 แสดงการใช้ Subtype constraint

(เอนทิตีทุกตัวของชนิดเอนทิตีผู้ชาย และเอนทิตีผู้หญิง ต่างก็เป็นสมาชิกของเอนทิตีบุคคล)



รูปที่ 2.16.1 แสดงการใช้ Mandatory constraint, lexical constraint

(บุคคลทุกคนต้องมีเพศ และสมาชิกต้องมีเพียง Male, female)

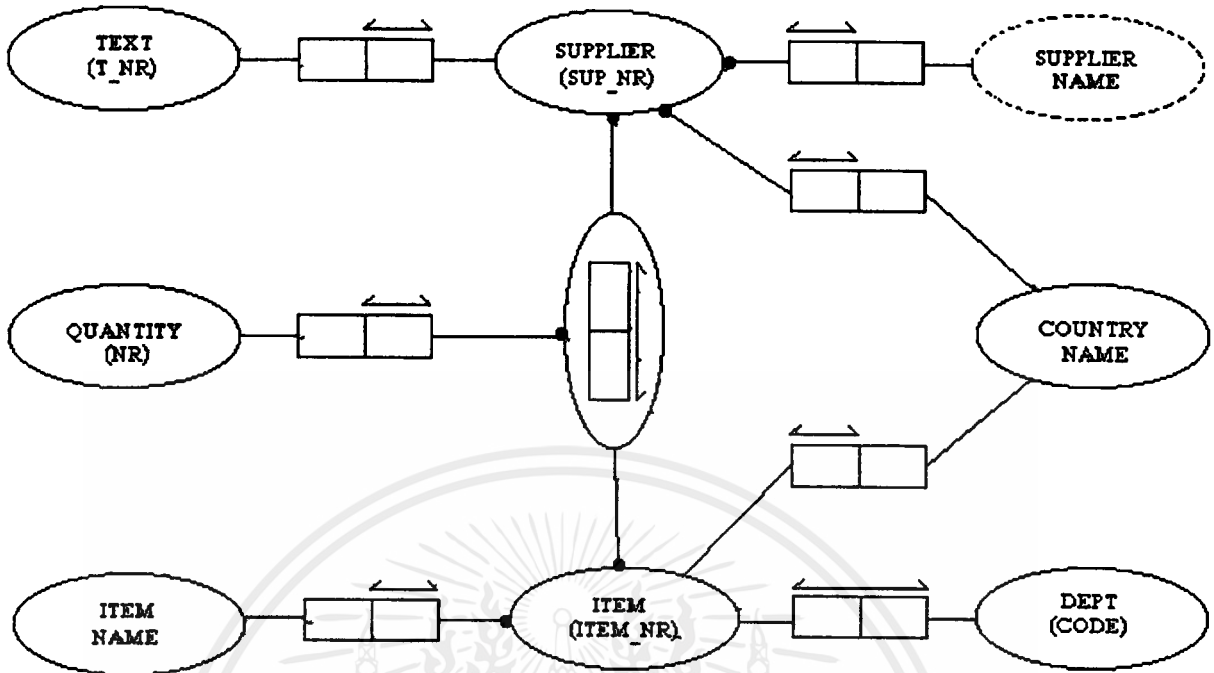


รูปที่ 2.16.2 (หมายความว่า Employee ทุกคนต้องเสียภาษี)

2.9 ONF อัลกอริทึม (The Optimal Normal Form algorithm)

เป็นวิธีการจัดกลุ่มความจริงใน Conceptual Schema ให้เป็น Relational Database Schema โดยมีหลักการโดยสรุปดังต่อไปนี้

1. สร้าง 1 รีเลชัน สำหรับชนิดความจริงแบบไบนารีซึ่งมีความสัมพันธ์แบบ many to many โดยที่ Unique Identifier ของชนิดเอนิตีที่เกี่ยวข้องทั้งหมดเป็น Primary Key
2. สร้าง 1 รีเลชัน สำหรับแต่ละชนิดความจริงแบบ n-ary โดยที่ Unique Identifier ของชนิดเอนิตีซึ่งมี role ถูกบังคับด้วย Uniqueness Constraint เดียวกันเป็น Candidate Key
3. พิจารณาชนิดเอนิตีที่เกี่ยวข้องกับความจริงแบบไบนารีที่มีความสัมพันธ์เป็นแบบ one to one หรือ many to one โดยที่ role ของชนิดเอนิตีเหล่านี้ถูกบังคับด้วย Unique Identifier ของชนิดเอนิตีเหล่านี้เป็น Candidate Key



รูปที่ 2.17 ตัวอย่างแบบจำลองข้อมูล (Conceptual Schema)

2.10 SQL Server

SQL Server เป็นระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS) ซึ่งมีคุณสมบัติต่าง ๆ ดังต่อไปนี้

2.10.1 ด้าน Architecture แบ่งลักษณะการทำงานออกเป็น 2 ชนิดคือ

2.10.1.1 User/Client Process เป็นกระบวนการทำงานของผู้ใช้

2.10.1.2 Server process ทำหน้าที่จัดการติดต่อสื่อสารระหว่าง Server process และ User process เพื่อทำตามที่ User process ต้องการ โดยมีข้อกำหนดเกี่ยวกับรูปแบบหลายรูปแบบโดยจะแบ่งตามจำนวนของผู้ใช้ต่อ Server process. แบ่งเป็น

1. Dedicated Server Configuration : server process จะรองรับ user process เพียง 1 process

2 Multi-Threaded Server Configuration : user process หลาย Process ใช้ server process ร่วมกันเพื่อลดจำนวน server process สามารถทำงานอยู่บนอุปกรณ์ที่แตกต่างกันได้ ซึ่ง user process และ server process ต้องแยกออกจากกัน เป็นการช่วยให้

- ลด Overhead จาก context switching เนื่องจาก server process ลดลงและยังเป็น การขจัดปัญหาเรื่อง bottleneck ของ CPU, I/O, Memory และ Operating System
- เพิ่มจำนวนผู้ใช้ต่อ Node
- จำนวนของ server process จะเป็นแบบ dynamic กล่าวคือ สามารถเปลี่ยนแปลงได้ตามขนาดของงาน โดยไม่มีผลกระทบต่อการใช้งาน

2.10.2 ด้าน Semantic Data Control

2.10.2.1 View Management

ระบบจะยอมให้ผู้ใช้สามารถที่จะทำการแทรก แก้ไข หรือลบข้อมูลผ่าน View ได้โดยมีกฎ บังคับคือ

- ถ้า view ที่ถูกสร้างขึ้นจาก join operation, distinct operation หรือ group by clause จะไม่สามารถแทรก แก้ไข หรือ ลบข้อมูลผ่าน view ได้
- ถ้า view ถูกระบุโดยมี with check option row จะไม่สามารถแทรก แก้ไข ลบ ข้อมูลผ่าน view ได้
- ถ้า view ถูกระบุเป็น not null column ที่ไม่มี default clause ใน base table จะไม่สามารถแทรก row ผ่านทาง view ได้
- ถ้า view ถูกสร้างโดย expression จะไม่สามารถแทรก แก้ไข ลบข้อมูล ผ่าน view ได้

กล่าวโดยรวมก็คือระบบสามารถควบคุมการแก้ไขผ่านทาง View ให้ถูกต้องได้ โดยจะไม่ อนุญาตให้มีการแก้ไขตามกรณีดังกล่าวมาแล้วข้างต้น

2.10.2.2 Data Integrity

1. Entity Integrity ซึ่งแบ่งเป็น

- DEFAULT Integrity Constraint : สำหรับคอลัมน์ที่ไม่ได้กำหนดระหว่างที่ insert
- UNIQUE Integrity Constraint : เป็น Integrity Constraint ซึ่งใช้กำหนดให้ คอลัมน์ที่ต้องการให้ rows 2 rows ใดๆ ในตารางมีค่าเดียวกันในคอลัมน์
- NOT NULL Integrity Constraint : เป็น Integrity Constraint ซึ่งใช้กำหนดให้ คอลัมน์ที่ต้องการในตาราง จะต้องไม่มีค่าเดียวกันในคอลัมน์
- CHECK Integrity Constraint : เป็น Integrity Constraint ซึ่งใช้กำหนดคอลัมน์ ที่ต้องการให้เป็นไปตามเงื่อนไขที่กำหนด โดยตรวจสอบทุกๆ rows ในตาราง

แล้วพบว่าเงื่อนไขนั้นเป็นเท็จ จะระงับ statement นั้นและจะหยุดการเปลี่ยนแปลงข้อมูล

2. Referential Integrity ซึ่งแบ่งเป็น

- PRIMARY KEY Integrity Constraint: แต่ละตารางในฐานะข้อมูลจะมี Constraint นี้ได้เพียง Constraint เดียว Rows ในตารางจะถูกกำหนดระบุและอ้างอิงโดย Primary Key ซึ่งต้อง มีความถูกต้องตาม Constraint นี้
- FOREIGN KEY Integrity Constraint เป็น Integrity Constraint ที่ทำหน้าที่รักษาความสัมพันธ์ระหว่างค่าของ Primary Key ใดๆ จะต้องมีค่าของ Foreign Key ในอีกตารางหนึ่งซึ่งการ match ก็สามารแบ่งได้เป็น
 - match full
 - match partial
 - match null

Action ซึ่งกำหนดโดย Referential Integrity Constraint แบ่งเป็น

- Update และ Delete Restrict : ค่าของ Key ที่เป็นตัวอ้างอิงให้กับ key อื่น จะถูกเปลี่ยนแปลงหรือลบไม่ได้ เช่น Primary Key จะถูกเปลี่ยนแปลงหรือลบไม่ได้ เพราะยังมีค่านั้นปรากฏอยู่เป็น Foreign Key ในตารางอื่นๆ
- Delete cascade : ถ้า row ที่มี key ที่เป็นตัวอ้างอิงให้กับ Key อื่นๆ ถูกลบแล้วทุก rows ที่มี Foreign Key ที่อ้างอิงจาก key นั้นจะต้องถูกลบไปด้วย

2.10.2.3 Database Trigger

ระบบจะยอมให้ผู้บริหารฐานข้อมูล และผู้ใช้สามารถควบคุม Integrity ได้โดยไม่จำเป็นต้องประกาศแบบที่เรียกว่า “ non-declarative “ โดยใช้ database trigger

Database Trigger หมายถึง Database Procedure ที่ใช้ควบคุม Integrity ซึ่งใช้เก็บไว้และสามารถเรียกใช้ในการ insert, delete หรือ update operations

เหตุการณ์ที่ควรใช้ Database trigger

1. เมื่อใช้ referential integrity โดยไม่ใช่ Integrity constraint
2. ใช้ referential integrity เมื่อใช้ child และ parent ตารางที่อยู่ต่าง node กันในระบบ distributed database
3. ใช้ business rule ที่ยุ่งยากซับซ้อนและไม่สามารถใช้ Integrity constraint ได้

2.10.3 DISTRIBUTED DATABASE MANAGEMENT

เป็นการเชื่อมต่อระหว่างโปรแกรมประยุกต์กับฐานข้อมูลโดยอนุญาตให้มีการเชื่อมต่อระหว่างโปรแกรมประยุกต์ Client และ server ของฐานข้อมูลซึ่งแยกจากการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ที่กำลังทำงานอยู่

Replicater ตัวกระจายข้อมูลจากเครื่องต้นทาง ไปยังเครื่องปลายทางตัวอื่นๆ สามารถทำได้หลายแบบไม่ว่าจะเป็นทั้งฐานข้อมูล, ตาราง, คอลัมน์ รวมทั้งการจัด Configuration ต่างๆ เช่น peer to peer , Master-Slave , Cascade และเวลาในการ replicate นั้น สามารถกำหนดแบบทันทีทันใด ตามตารางเวลา หรือ ทำเมื่อสั่งก็ได้

2.11 โอเพินดาต้าเบสคอนเนคตีวิตี (โอดีบีซี) (Open Database Connectivity, ODBC)

เป็นมาตรฐานการอินเทอร์เฟซซึ่งจะทำให้โปรแกรมประยุกต์สามารถเข้าถึงข้อมูลที่เก็บอยู่ในระบบฐานข้อมูล ซึ่งถูกดูแลโดย DBMS โดยใช้คำสั่งของ SQL เป็นมาตรฐานในการติดต่อ โดยก่อนที่จะมีการพัฒนา ODBC ขึ้นมานั้น การพัฒนาแอปพลิเคชันจะขึ้นกับ DBMS ซึ่งมีลักษณะเป็นเอมเบสเคดเอสคิวแอล (Embedded SQL) ซึ่งประสิทธิภาพของเอมเบสเคดเอสคิวแอลนี้จะขึ้นอยู่กับฮาร์ดแวร์และสภาพแวดล้อมของระบบปฏิบัติการ ซึ่งจะทำให้ซอฟต์แวร์จะต้องถูกคอมไพล์ใหม่สำหรับแต่ละสภาพแวดล้อม

การที่แอปพลิเคชัน ใช้โอดีบีซีเป็นตัวติดต่อเพื่อการเข้าถึงข้อมูลนี้จะทำให้แอปพลิเคชันสามารถเข้าถึงข้อมูลคิวบีเอ็มเอสที่แตกต่างกันได้ ซึ่งทำให้ผู้พัฒนาแอปพลิเคชันสามารถนำแอปพลิเคชันที่พัฒนาแล้วไปใช้กับคิวบีเอ็มเอสตัวใดก็ได้ โดยที่ผู้ใช้สามารถเพิ่มเติมโมดูล (Database Driver) ที่จะลิงก์แอปพลิเคชันให้ใช้กับคิวบีเอ็มเอสที่ต้องการได้

2.11.1 ส่วนประกอบในการอินเทอร์เฟซกับ ODBC

- ไลบรารีของฟังก์ชัน ซึ่งจะช่วยให้แอปพลิเคชันติดต่อกับคิวบีเอ็มเอสเพื่อที่จะเอ็กรหัสคิวทีคำสั่งแล้วดึงข้อมูลออกมา
- ไลบรารีของเอสคิวแอลซึ่งยึดถือตามมาตรฐานของ X/Open และ SQL Access Group(SAG)SQL CAE Specification (1992)
- มาตรฐานรหัสข้อผิดพลาด (Error code)
- วิธีมาตรฐานในการติดต่อและการล็อกเข้าคิวบีเอ็มเอส
- มาตรฐานประเภทของข้อมูล

2.11.2 ข้อดีของอินเทอร์เฟซกับโอคิบีซี

- สามารถกำหนดคำสั่งเอสคิวแอลได้ทั้งในซอร์สโค้ด (Source code) ของแอปพลิเคชันนั้นและขณะกำลังทำงาน (Run time)
- ซอร์สโค้ดเดียวกันสามารถเข้าถึงข้อมูลที่เกี่ยวข้องบนคิบีเอ็มเอสคนละตัวกันได้
- โพรโทคอลในการสื่อสารข้อมูล (Data Communication Protocol) ระหว่างแอปพลิเคชันแต่ละตัวไม่มีความสำคัญ
- สามารถปรับแต่งรูปแบบของข้อมูลที่รับและส่งระหว่างแอปพลิเคชันกับตัวโอคิบีซีได้ตามต้องการ

2.11.3 ส่วนประกอบของโอคิบีซี (ODBC Components)

- แอปพลิเคชัน (Application) มีหน้าที่หลักในการเข้าถึงข้อมูลภายในฐานข้อมูล คือ การเรียกใช้ฟังก์ชันของโอคิบีซี และการส่งคำสั่งเอสคิวแอล และการรับข้อมูลผลลัพธ์
- ไดรฟ์เวอร์ แมเนเจอร์ (Driver Manager) จะทำการโหลดไดรฟ์เวอร์ในนามของแอปพลิเคชัน
- ไดรฟ์เวอร์ (Driver) ปฏิบัติตามคำสั่งที่เรียกมาจากแอปพลิเคชัน ส่งคำสั่งเอสคิวแอลไปยังดาต้าซอร์ส (Datasource) และรอรับผลตอบสนองเพื่อส่งต่อไปยังแอปพลิเคชัน นอกจากนี้ยังสามารถแก้ไขปรับปรุงรูปแบบการร้องขอจากแอปพลิเคชันเพื่อให้สอดคล้องกับคิบีเอ็มเอส
- ดาต้าซอร์ส (Data source) ประกอบด้วยข้อมูลที่ผู้ใช้ต้องการจะเข้าถึง รวมไปถึงระบบปฏิบัติการที่เกี่ยวข้อง คิบีเอ็มเอส และเน็ตเวิร์คแพลตฟอร์ม (Network Platform)

2.11.4 แอปพลิเคชัน

แอปพลิเคชันจะทำงานต่างๆ ผ่านทางโอคิบีซีอินเทอร์เฟซดังต่อไปนี้

- ทำการร้องขอการติดต่อไปยังดาต้าซอร์ส
- ส่งเอสคิวแอลไปยังดาต้าซอร์ส
- กำหนดเงื่อนไขและรูปแบบสำหรับเก็บผลลัพธ์จากการประมวลผลคำสั่งเอสคิวแอล
- ร้องขอผลลัพธ์
- ประมวลผลผลลัพธ์กลับไปสู่ผู้ใช้
- รายงานผลลัพธ์กลับไปสู่ผู้ใช้
- ร้องขอการคอมมิต (Commit) และการโรลแบค (Rollback) สำหรับการควบคุมทรานเซกชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้แก่กิจการติดต่อจากคิบีเอ็มเอสเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11.5 ไคร์ฟเวอร์แมนเนเจอร์

ไคร์ฟเวอร์แมนเนเจอร์ เป็นไดนามิกลิงค์ไลบรารี (Dynamic-link library, DLL) จุดประสงค์หลักของไคร์ฟเวอร์แมนเนเจอร์ คือการโหลดไคร์ฟเวอร์ โดยจะทำตามลำดับดังต่อไปนี้

- ใช้ ODBC.INI ไฟล์เพื่อแมปชื่อของคาคำขอเพื่อค้นหาไคร์ฟเวอร์ดีแอลแอล
- ทำการประมวลผลโอดีบีซีฟังก์ชันสำหรับแต่ละไคร์ฟเวอร์
- จัดการโอดีบีซีฟังก์ชันสำหรับแต่ละไคร์ฟเวอร์
- จัดการพารามิเตอร์ให้ถูกต้องสำหรับการเรียกใช้โอดีบีซี

2.11.6 ไคร์ฟเวอร์

ไคร์ฟเวอร์เป็นดีแอลแอลที่อิมพลิเมนต์โอดีบีซีฟังก์ชันคอล และทำการติดต่อกับข้อมูลที่คาคำขอ โดยไคร์ฟเวอร์แมนเนเจอร์จะทำการโหลดไคร์ฟเวอร์เมื่อแอปพลิเคชันใดเรียกใช้ฟังก์ชัน SQLBrowseConnect, SQLDriverConnect โดยใช้ไคร์ฟเวอร์จะทำตามขั้นตอนดังต่อไปนี้เพื่อตอบรับการเรียกโอดีบีซีฟังก์ชันของแอปพลิเคชัน

- สร้างการติดต่อไปยังคาคำขอ
- ส่งการร้องขอไปยังคาคำขอ
- แปลงข้อมูลให้เป็นรูปแบบอื่นถ้าเป็นการเรียกโดยแอปพลิเคชัน
- ส่งผลลัพธ์กลับไปยังแอปพลิเคชัน
- ถ้าเกิดข้อผิดพลาดขึ้นจะมีการส่งคัดข้อผิดพลาดกลับไปยังแอปพลิเคชันที่เรียกมา
- ทำการประกาศและจัดการเกี่ยวกับเคอร์เซอร์ (Cursors) ถ้าจำเป็น
- อินนิเชียลทรานเซกชันถ้าคาคำขอสั้น ๆ ต้องการ

2.11.7 ชนิดของไคร์ฟเวอร์

โอดีบีซีกำหนดชนิดของไคร์ฟเวอร์ไว้ 2 รูปแบบคือ

- ซิงเกิลเทียร์ (Single-tier) ไคร์ฟเวอร์ชนิดนี้จะมีหน้าที่เอ็กซีคิวท์ฟังก์ชันที่ส่งมาจากโอดีบีซี รวมทั้งคำสั่งเอสคิวแอลด้วย ดังนั้นไคร์ฟเวอร์ชนิดนี้จึงมีหน้าที่บางส่วนของคาคำขออยู่ด้วย
- มัลติเพิลเทียร์ (Multiple- tier) ไคร์ฟเวอร์ชนิดนี้จะมีหน้าที่เอ็กซีคิวท์เฉพาะฟังก์ชันที่ส่งมาจากโอดีบีซีเท่านั้น ส่วนคำสั่งเอสคิวแอล จะส่งไปที่กับตัวคาคำขอซึ่งระบบ ๆ หนึ่งอาจจะประกอบด้วยไคร์ฟเวอร์ทั้งสองชนิดเลขก็ได้

2.11.8 ระดับข้อตกลงของโอดีบีซี (ODBC Conformance Levies)

ในการใช้มาตรฐานโอดีบีซีในการเชื่อมต่อนั้นจะต้องมีข้อตกลงมาตรฐานเพื่อที่จะใช้คนพัฒนา ส่วนของไครฟเวอร์สามารถครอบคลุมฟังก์ชันต่างๆ ที่จำเป็นได้ครบ และทางด้านแอฟลิเคชันสามารถเลือกใช้ไครฟเวอร์ตามฟังก์ชันมาตรฐานได้โดยง่าย

โดยระดับข้อตกลงของไครฟเวอร์แบ่งเป็น

- ODBC API ซึ่งแบ่งย่อยออกได้เป็น
 - CORE API อ้างอิงตามมาตรฐานของ X/OPEN & SQL Access Group Call Level Interface
 - Level1 API
 - Level2 API

ซึ่งสองระดับท้ายสุดถือว่าเป็นระดับเอ็กเทนเดดฟังก์ชันนอลลิตี (Extended Functionality)

- ODBC SQL Grammar และ ODBC SQL datatypes แบ่งออกเป็น
 - Minimum SQL Grammar ซึ่งเป็นระดับพื้นฐาน
 - Core SQL Grammar ยึดตามมาตรฐานที่ระบุใน X/OPEN & SQL Access Group SQL CAE Specification (1992)
 - Extended SQL Grammar Support Extension SQL ของดีบีเอ็มเอสทั่วไป

2.11.9 คอนเนกชันและทรานแซกชัน (Connection & Transaction)

ก่อนที่โปรแกรมประยุกต์จะสามารถติดต่อกับโอดีบีซีได้จะต้องทำการร้องขอต่อโอดีบีซีเพื่อจอง เอ็นไวรอนเมนต์แฮนเดิล (Environment Handle, henv) และก่อนที่จะติดต่อกับค้ำดาซอสได้นั้นแอฟลิเคชัน จะต้องร้องขอเพื่อจองคอนเนกชันแฮนเดิล (Connection Handle, hdbc) และยังคงใช้ henv, hdbc ไว้สำหรับการอ้างอิงภายหลังด้วย

แอฟลิเคชันหนึ่งๆ สามารถจะจอง Henv ได้หลายๆ ตัวสำหรับค้ำดาซอสตัวหนึ่งหรือ หลายๆ ตัวก็ได้ ซึ่งแต่ละคอนเนกชันจะถือว่าการทำทรานแซกชันนั้นแยกออกจากกัน และในเวลาหนึ่งๆจะมีเพียง หนึ่งคอนเนกชันเท่านั้นที่จัดเป็นแอคทีฟคอนเนกชัน (Active Connection) ไครฟเวอร์จะเป็นตัวจัดการ ทำทรานแซกชันของแอคทีฟเซกชัน ซึ่งแอฟลิเคชันสามารถร้องขอได้ว่าจะให้ค้ำสแตทัสเอนั้นคอมมิต โดยทันทีหรือไม่ (รวมไปถึงการ โรลแบคด้วย)

ส่วนทางด้านไครฟเวอร์มีแมเนเจอร์จะคอยดูแลและจัดการ สวิตช์ไปมาระหว่างคอนเนกชันต่างๆ ในขณะที่ทรานแซกชันกำลังดำเนินอยู่

บทที่ 3

การดำเนินงานและพัฒนาระบบ

3.1 การดำเนินงานในปัจจุบัน

ในปัจจุบันสำนักงานทะเบียนประมวลผล ประสบกับปัญหาในการดำเนินงานเนื่องจากไม่มีความสะดวกในการเข้าถึงข้อมูลบางระดับที่จัดเก็บอยู่ที่สำนักบริการคอมพิวเตอร์ เพื่อทำการแก้ไขการลงทะเบียนเรียนของนิสิตให้ทันเหตุการณ์ หรือการเรียกใช้ข้อมูลเพื่อตรวจสอบความถูกต้องก่อนออกหลักฐานทางการศึกษาต่างๆ นอกจากนั้นข้อมูลบางอย่างซึ่งเป็นที่ต้องการของหน่วยงานต่างๆ สำนักทะเบียนและประมวลผลไม่สามารถออกได้รวดเร็ว ทันความต้องการของผู้ใช้ในระดับต่างๆที่เกี่ยวข้อง

นั่นคือภาพรวมของทั้งหมดของระบบสารสนเทศของสำนักทะเบียนและประมวลผล แต่เนื่องจากได้มีการแบ่งระบบรวมทั้งหมดออกเป็นระบบย่อยๆ 4 ระบบ คือ

- ระบบทะเบียนประวัตินิสิต และหลักสูตร
- ระบบการจัดชั้นเรียน และห้องเรียน
- ระบบการลงทะเบียน และคิดค่าธรรมเนียม
- ระบบการคิดคะแนน – เกรด และออกเอกสารทางการศึกษา

โดยแต่ละส่วน จะถูกแบ่งออกให้แต่ละกลุ่ม ซึ่งในโครงการนี้จะเป็นส่วนของระบบการจัดชั้นเรียนและห้องเรียน

3.1.1 ลักษณะของการทำงานโดยรวมของระบบทั้งหมด

ระดับคณะวิชา

- **อาจารย์ประจำวิชา** ต้องการเรียกใช้ข้อมูลจากระบบเพื่อทราบถึงจำนวนนิสิตที่ลงทะเบียนในวิชาที่ตนรับผิดชอบ จำแนกตามคณะ สาขาวิชาเอก ชั้นปี ประเภทการลงทะเบียนเรียนในแต่ละภาคการศึกษา ค่าตอบแทนในการสอนภาคฤดูร้อน การส่งคะแนนผ่านระบบ on-line การติดต่อสื่อสารข้อมูลกับอาจารย์ที่ปรึกษาของนิสิตที่ลงทะเบียนเรียนในวิชาของตน โดยผ่านระบบการลงทะเบียนเรียน เป็นต้น
- **อาจารย์ที่ปรึกษาของนิสิต** ต้องการเรียกใช้ข้อมูลจากระบบเพื่อทราบถึงจำนวนนิสิตที่อยู่ในความรับผิดชอบของตน เพื่อตรวจสอบสถานภาพ และประวัติส่วนตัว ประวัติการเรียนและลงทะเบียนรวมถึงการติดต่อสื่อสารข้อมูลกับนิสิตในความดูแลรับผิดชอบโดยผ่านระบบการลงทะเบียนเรียน

- **นายทะเบียนคณะ** ต้องการเรียกใช้ข้อมูลจากระบบเพื่อทราบถึงประวัติการลงทะเบียนสถานภาพนิสิตในคณะของตน ข้อมูลภาระการสอนของอาจารย์ในสาขาวิชาต่างๆ ภายในคณะของตน จำนวนนิสิตในคณะ และต่างคณะที่ลงทะเบียนเรียนรายวิชาของคณะ จำแนกตามคณะ สาขาวิชา ชั้นปี ประเภทการลงทะเบียนเรียน การสื่อสารข้อมูลกับอาจารย์ประจำวิชาในคณะและต่างคณะ โดยผ่านระบบการลงทะเบียนเรียน เพื่อติดตามผลคะแนนของนิสิตที่ตนรับผิดชอบ
- **นิสิต** ต้องการเรียกใช้ข้อมูลจากระบบ เพื่อทราบถึงสถานภาพนิสิต ข้อมูลการลงทะเบียนเรียนของตน ระดับคะแนนรายวิชาที่ลงทะเบียนเรียน การติดต่อสื่อสารข้อมูลกับอาจารย์ที่ปรึกษาของตน อาจารย์ประจำวิชาของตน โดยผ่านระบบการลงทะเบียนเรียน การทราบข่าวสารข้อมูลการลงทะเบียนเรียนในปัจจุบัน

ระดับหน่วยงาน

- **กองคลัง** ต้องการเรียกใช้ข้อมูลจากระบบ เพื่อทราบถึงจำนวนเงินค่าธรรมเนียมการศึกษา
- **กองแผนงาน** ต้องการเรียกใช้ข้อมูลจากระบบ เพื่อทราบถึงภาระงานสอนของอาจารย์ จำนวนนิสิตที่ลงทะเบียนเรียนในรายวิชาต่างๆ ในคณะวิชาต่างๆ จำนวนทรัพยากรที่ใช้ในการเรียนการสอน

ระดับสำนัก

- **สำนักทะเบียนและประมวลผล** ต้องการเรียกใช้ข้อมูลจากระบบ เพื่อทราบถึงการลงทะเบียนของนิสิต สถานภาพนิสิต จำนวนนิสิตที่จะลงทะเบียนเรียนในแต่ละวิชา และการเรียกใช้ข้อมูลที่เกี่ยวข้องกับการลงทะเบียนเรียนโดยผ่านระบบ online .

ระดับผู้บริหาร

- ต้องการเรียกใช้ข้อมูลจากระบบ เพื่อศึกษาแนวโน้ม และทิศทางของการจัดการศึกษาในมหาวิทยาลัย เพื่อสนับสนุนหรือตัดสินใจในการกำหนดนโยบายที่เกี่ยวข้องกับการจัดการศึกษาให้สอดคล้องกับทรัพยากรที่มีอยู่ หรือเพื่อแสวงหาทรัพยากรเพิ่มเติมเพื่อรองรับการขยายตัวของความต้องการการศึกษาในสาขาวิชาต่างๆ

นอกจากนี้ปัญหาที่สำคัญและพบอยู่ในระบบปัจจุบันซึ่งสมควรจะต้องได้รับการ

ปรับปรุงก็คือ

3.1.2 ปัญหาในการทำงาน

ปัญหาในการลงทะเบียน เมื่อนักศึกษาทำการลงทะเบียนและขอแก้ไขการลงทะเบียนแล้วทางสำนักทะเบียนและประมวลผลกลางจะทำการตรวจสอบวิชาที่ต้องผ่านมาก่อนในการลงทะเบียนรายวิชาต่างๆ (การตรวจสอบ Prerequisite ของรายวิชาที่นักศึกษาแต่ละคนลงทะเบียนเรียน) ซึ่งโดยปกติแล้วการเอกสารนี้เป็นเอกสารที่ส่งวนเวียนสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรวจสอบจะกระทำหลังจากที่นักศึกษาได้ลงทะเบียนเสร็จเรียบร้อยแล้ว เจ้าหน้าที่ของสำนักทะเบียนและประมวลผลกลางจะเป็นผู้ตรวจสอบความถูกต้องต่างๆ จากหนังสือหลักสูตรการเรียน – การสอนของแต่ละคณะ หากพบความผิดพลาดก็จะแจ้งให้นักศึกษามาทำการถอนรายวิชานั้นๆ ต่อไป

ซึ่งจะเห็นว่าทั้งนักศึกษาและเจ้าหน้าที่ต่างก็ต้องคอยติดตามการตรวจสอบวิชาที่ต้องผ่านมาก่อนในการลงทะเบียนในรายวิชาต่างๆ ในภายหลังเอง ซึ่งจะทำให้ล่าช้า

ปัญหาในการออกเอกสารทางการศึกษา ในการยื่นคำร้องขอเอกสารทางการศึกษาแต่ละครั้งของนิสิต พบว่าเกิดความล่าช้าในการดำเนินงาน เนื่องจากการจัดเก็บข้อมูลและเอกสารต่างๆ ถูกจัดเก็บแยกตามหน่วยงาน และแผนกต่างๆ ทำให้การดำเนินการจัดทำเกิดความล่าช้า นอกจากนี้ยังเป็นผลสืบเนื่องมาจากการดำเนินงานที่ซ้ำซ้อนอีกด้วย

จากที่กล่าวมาคือปัญหาสำคัญที่เกิดขึ้นในการดำเนินงานในปัจจุบัน ซึ่งระบบที่สร้างขึ้นใหม่จะต้องสามารถแก้ปัญหาเหล่านี้ได้ และปรับปรุงระบบย่อยส่วนอื่นๆ ให้มีประสิทธิภาพมากขึ้นกว่าเดิมได้ตามความต้องการของผู้ใช้ระบบ ดังจะกล่าวในหัวข้อถัดไป

3.2 วัตถุประสงค์ของโครงการ

จะจำแนกออกเป็น 2 วัตถุประสงค์หลักๆ ก็คือ

1. ในด้านการปรับปรุงและเพิ่มประสิทธิภาพในการดำเนินงานของสำนักทะเบียน และประมวลผล
2. ปรับปรุงและเพิ่มประสิทธิภาพของตัวซอฟต์แวร์ของรุ่นที่เคยได้ทำไว้ แก้ไขข้อจำกัดต่างๆ ของซอฟต์แวร์เดิม ให้สามารถรองรับปริมาณงานที่อาจจะเพิ่มขึ้นในอนาคตได้

3.2.1 ปรับปรุงและเพิ่มประสิทธิภาพในการดำเนินงานของสำนักทะเบียนและประมวลผล

เนื่องจากสำนักทะเบียนประมวลผลมีความต้องการใช้ระบบคอมพิวเตอร์ เพื่อนำมาช่วยงานในระบบต่างๆ ที่เกี่ยวข้องกับทะเบียนประวัติ และการลงทะเบียนของนิสิต โดยที่ทุกระบบจะต้องมีการติดต่อเป็นแบบ GUI (Graphical User Interface)

ฉะนั้นจุดมุ่งหมายของโครงการนี้ นอกจากแก้ปัญหาที่พบได้ในการดำเนินงานในปัจจุบัน และเพิ่มประสิทธิภาพของระบบย่อยๆ อื่นแล้ว ระบบฐานข้อมูลที่เหมาะสมของสำนักทะเบียนประมวลผลมีความจำเป็นต้องได้รับการพัฒนาขึ้นมาเป็นระบบสารสนเทศเพื่อการจัดการหรือระบบทะเบียนนิสิตเพื่อการจัดการการศึกษาให้สอดคล้องกับงานของฝ่ายต่างๆ ภายในสำนักงานที่ต้องเกี่ยวข้องกับงานทะเบียนประวัติ นิสิต และการลงทะเบียนประวัติ นิสิต และการลงทะเบียนเรียนของนิสิต โดยมีวัตถุประสงค์ที่สำคัญ คือ

- เพื่อช่วยผู้บริหารและผู้ปฏิบัติงานเข้าใจกระบวนการปฏิบัติงานได้ชัดเจนมากขึ้น
- และสามารถเพิ่มประสิทธิภาพให้กระบวนการปฏิบัติงานที่รับผิดชอบให้มีความกระชับและคล่องตัวมากขึ้น
- เพื่อลดขั้นตอนการบวนวิธีการปฏิบัติงานประจำวัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เพื่อจัดเตรียมองค์กรให้มีขีดความสามารถในการสร้างงานและการรองรับงาน
- ทะเบียนประวัตินิติศาสตร์ และการลงทะเบียนเรียนของนิติศาสตร์ที่จะขยายตัวเพิ่มขึ้นในอนาคต

3.2.2 ปรับปรุงและเพิ่มประสิทธิภาพของตัวซอฟต์แวร์

ซอฟต์แวร์ที่ได้ทำการพัฒนาก่อนหน้านี้ไม่สามารถที่จะนำมาปรับปรุงให้สามารถติดต่อผ่านระบบเครือข่ายอินเทอร์เน็ตได้ หรือเป็นไปได้ด้วยความยากลำบากมาก การใช้งานก็ไม่สะดวกเพราะที่เครื่องลูกข่ายที่ต้องการจะใช้งานจะต้องทำการติดตั้งซอฟต์แวร์เอาไว้ก่อน จึงเป็นการไม่สะดวกในการใช้งาน ดังนั้นในโครงการนี้จึงทำการพัฒนาระบบโดยการเปลี่ยนมาใช้ระบบ WWW แทน ซึ่งจะมีข้อได้เปรียบหลายประการด้วยกัน อาทิ

- เครื่องลูกข่ายติดตั้งเฉพาะ Browser ก็สามารถใช้งานได้
- สามารถใช้งานได้ไม่ว่าจะอยู่ที่ใด เพราะระบบติดต่อผ่านระบบอินเทอร์เน็ต
- ขจัดปัญหาเครื่องลูกข่ายไม่เพียงพอต่อการใช้งาน เพราะสามารถใช้เครื่องคอมพิวเตอร์ที่สามารถติดต่ออินเทอร์เน็ตเครื่องใดก็ได้

3.3 ขั้นตอนในการพัฒนาโครงการ

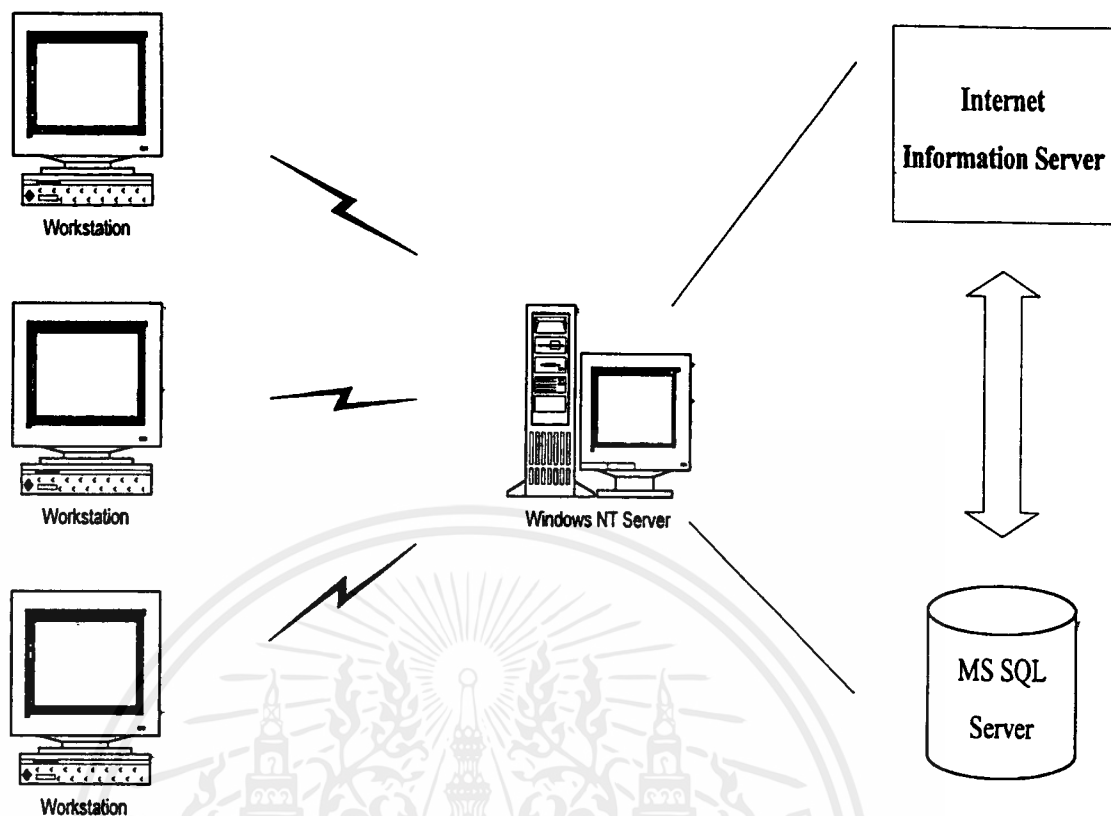
1. สรุบบทบาทของงานที่จะทำ โดยพิจารณาถึงลักษณะการนำไปใช้ในงานจริง
2. ศึกษาขั้นตอนต่างๆที่ควรจะเป็นของระบบแล้วนำมาเขียนเป็น Data Flow Diagram และ Diagram แสดงขั้นตอนการทำงานของระบบ
3. ศึกษาฐานข้อมูลของระบบการลงทะเบียนของโครงการเก่า (โครงการระบบสารสนเทศของสำนักทะเบียนและประมวลผล ซึ่งเป็นโปรแกรมประยุกต์บนระบบเครือข่ายแบบไคลเอนต์/เซิร์ฟเวอร์) และนำ Data Flow Diagram มาใช้ประกอบในการพิจารณาการออกแบบฐานข้อมูลเพิ่มเติมจากฐานข้อมูลเดิม โดยใช้ NIAM แสดงความสัมพันธ์ของข้อมูลแล้วทำการแปลงออกมาเป็นตาราง จากนั้นจึงทำการระบุชนิดของข้อมูล และลักษณะของข้อมูลที่จะเก็บลงตาราง
4. ทำการติดตั้งระบบปฏิบัติการ Window NT 4.0 Thai Enable, ติดตั้ง IIS WebServer และติดตั้ง Microsoft SQL Sever
5. ทำการนำเข้าข้อมูลที่จำเป็นต่อการทดลอง ของโครงการนี้
6. ทำการออกแบบลักษณะของหน้าจอ (User Interface) และแบบฟอร์มต่างๆ ที่จะนำมาแสดงบนบราวเซอร์ โดยยึดหลักการใช้งานง่ายและถูกต้อง
7. สำหรับการสร้างแบบฟอร์มนั้นจะสร้างโดยใช้ภาษา HTML และ Java Script ในการพัฒนา จะเขียนแล้วเก็บลงในรูปแบบของไฟล์ HTX

8. สำหรับส่วน Server จะต้องมีกรเขียน Script IDC ขึ้นมาเพื่อทำหน้าที่เป็นตัวกลางในการที่ เว็บเบราว์เซอร์ จะทำงานกับฐานข้อมูลที่อยู่บน Server นี้ นอกจากนั้นยังต้องสร้างสตอร์โพซิเจอร์ต่างๆที่จำเป็นต่อระบบเพื่ออำนวยความสะดวกในการพัฒนาระบบให้มีประสิทธิภาพยิ่งขึ้น
9. ในการพัฒนาต้องมีการตรวจสอบความถูกต้องของระบบในทุกๆขั้นตอน ถ้าพบข้อผิดพลาดก็มีความจำเป็นที่จะต้องแก้ไขข้อผิดพลาดนั้นเพราะถ้าไม่ทำการแก้ไขแล้วก็จะไม่สามารถทำการพัฒนาระบบต่อไปได้ทุกครั้งที่มีการแก้ไขจะมีการบันทึกปัญหาต่างๆและวิธีการแก้ไขลง Log Book ไว้ด้วย
10. จัดทำเอกสารอธิบายถึงรายละเอียดทั้งหมดของระบบทั้งในด้านทฤษฎีต่างๆที่นำมาใช้และบอกถึงผลงานที่ได้ทำไปพร้อมแนวทางการพัฒนาต่อ
11. จัดทำคู่มือการใช้งาน (User Manual)

3.4 โครงสร้างของระบบ

โครงสร้างหลักของระบบการลงทะเบียนเรียนผ่านอินเทอร์เน็ต นั้นประกอบไปด้วยส่วนสำคัญๆดังนี้

1. ผู้ขอใช้บริการ (Client) จะขอใช้บริการได้โดยใช้เว็บเบราว์เซอร์ ซึ่งโครงงานนี้ต้องการเว็บเบราว์เซอร์ของ Internet Explorer เวอร์ชัน 3.0 ขึ้นไปและ Netscape Navigator เวอร์ชัน 3.0 ขึ้นไป
2. ระบบเครือข่ายซึ่งเชื่อมต่อกับระบบเครือข่าย Internet ก็ต้องสนับสนุนการสื่อสารด้วยโปรโตคอล TCP/IP
3. ผู้ให้บริการ (Web&Database Server) เป็นส่วนที่ให้บริการ WWW และบริการฐานข้อมูล



รูปที่ 3.1 การเชื่อมต่อของระบบผ่านทางอินเทอร์เน็ต

3.4.1 สถาปัตยกรรมแบบ 3 ระดับ (Tree – Tiers Architecture)

การออกแบบโดยใช้สถาปัตยกรรมแบบ 3 ระดับ (Tree – Tiers Architecture) จะแบ่งการทำงานเป็น 3 ส่วนแยกการทำงานเป็นอิสระจากกัน ทำให้เมื่อเกิดการเปลี่ยนแปลงในระดับใดแล้วจะไม่ส่งผลกระทบต่อระดับอื่นๆ โดยจะแบ่งเป็น 3 ระดับดังนี้

3.4.1.1. ระดับให้บริการแก่ผู้ใช้บริการ (User Service)

เป็นส่วนของโปรแกรมประยุกต์ ที่สำหรับใช้ติดต่อกับผู้ใช้บริการ โดยส่วนนี้ในปัจจุบันมักจะใช้การเชื่อมต่อแบบรูปภาพ (GUI – Graphic User Interface) เพราะผู้ใช้บริการจะใช้ได้ง่ายโดยส่วนนี้เราใช้โปรแกรม Web Browser ต่างที่มีอยู่ทั่วไป ทำให้ผู้ใช้บริการคุ้นเคยและสะดวก

3.4.1.2. ระดับให้บริการกฎเกณฑ์ของระบบ (Business Service)

เป็นส่วนที่ให้บริการ และ เก็บกฎเกณฑ์ต่างๆ ของระบบ เป็นส่วนที่ทำหน้าที่เชื่อมโยงกับ User Service กับ Data Service ส่วนนี้จะตรวจสอบ Input ที่เข้ามาจาก User Service ว่ามีความถูกต้องตามกฎเกณฑ์ที่ได้กำหนดไว้หรือไม่ เพื่อเป็นการรักษาความปลอดภัยให้แก่ระบบโดยส่วนนี้จะใช้โปรแกรม Internet Information Server โดยโปรแกรมนี้จะทำหน้าที่เป็น Web Server คือดูแลเกี่ยวกับระบบอินเทอร์เน็ต

3.4.1.3. ระดับให้บริการฐานข้อมูล (Data Service)

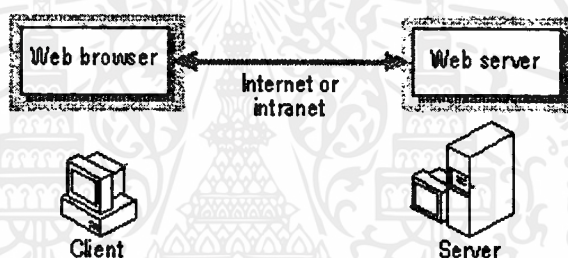
เป็นส่วนที่ให้บริการเกี่ยวกับการจัดการฐานข้อมูล ซึ่งส่วนนี้จะรองรับคำสั่งขอข้อมูลที่ถูกต้องมาจาก Business Service และนำไปทำการค้นหาจาก Data Base โดยส่วนนี้จะใช้โปรแกรม Microsoft SQL Server 6.5 ซึ่ง run ระบบปฏิบัติการ Windows NT 4.0

3.4.2 Internet Information Server

Microsoft Internet Information Server คือ web server ซึ่งจะส่งข้อมูลโดยใช้ Hypertext Transfer Protocol (HTTP) และ Internet Information Server โดยทั่วไปจะให้บริการ File Transfer Protocol(FTP) และ Gopher บริการ FTP จะช่วยผู้ใช้ในการส่งไฟล์ระหว่างแต่ละ Web Site ส่วน Gopher เป็นส่วนที่ใหญ่มากที่สุดที่กันไว้ค้นหาโดย HTTP protocol

3.4.2.1 การทำงานของ Internet Information Server

Web โดยพื้นฐาน คือ ระบบในการร้องขอและตอบสนอง โดย Web Browser ร้องขอข้อมูลโดยการส่ง URL ไปยัง Web Server แล้ว Web Server ก็จะตอบสนองโดยการคืน HTML page กลับมา



รูปที่ 3.2 แสดงการทำงานของ Internet

HTML page อาจจะเป็น static page ที่ได้ทำการจัดรูปแบบและเก็บไว้เรียบร้อยแล้วบน Web Site นั้น หรือ อาจจะเป็น page ที่ Server สร้างขึ้นมาจากการตอบสนองข้อมูลจากผู้ใช้ หรือ อาจจะเป็น page ที่แสดงรายชื่อไฟล์ หรือ folder บน Web Site นั้น

3.4.3 Web Browser URL Request

URL มีไว้เพื่อระบุถึงเพจแต่ละเพจ Web Browser ร้องขอเพจโดยการส่ง URL ไปยัง Web Server และ Server จะใช้ URL นี้ในการระบุถึงที่อยู่และแสดงเพจนี้

URL มีไวยากรณ์ คือ protocol, ชื่อของโดเมน, พาร์ทของข้อมูล โดยที่ Protocol คือ วิธีการติดต่อสื่อสาร เช่น Hypertext Transport Protocol (HTTP) และ Internet Information Server สนับสนุนการทำงานของ HTTP Protocol, FTP และ Gopher

ชื่อโดเมน คือ ชื่อ Domain Name System(DNS) ของเครื่องคอมพิวเตอร์

พาร์ท คือ พาร์ทของข้อมูล

ต่อไปนี้เป็นตารางแสดงถึงตัวอย่างของ URL ที่แตกต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Protocol	Domain Name	Path to Information
http://	www.microsoft.com	/backoffice
Https:// (secure HTTP)	www.company.com	/catalog/orders.htm
gopher://	gopher.college.edu	/research/astronomy/index.htm
ftp://	orion.bureau.gov	/stars/alpha quadrant/starlist.txt

ตารางที่ 3.1 แสดงตัวอย่างของ URL ที่แตกต่างกัน

URL สามารถเก็บข้อมูลที่ Web Server จะใช้ก่อนกลับไปยังเพจนั้น โดยข้อมูลจะอยู่ที่ส่วนท้ายสุดของพาร์ท Web Server จะส่งข้อมูลให้แก่โปรแกรมเพื่อประมวลผล และ คืนค่ากลับมาเป็น Web Page

ตารางนี้ คือ ตัวอย่างของการร้องขอแบบต่างๆ

Request Type	URL
Static HTML page	http://www.microsoft.com/backoffice/home.htm
ISAPI application	http://www.msn.com/custom/page1.dll?CUST=on
Internet Database Connector	http://www.microsoft.com/feedback/input.idc
Common Gateway Interface (CGI) script	http://www.company.com/calculator/add.pl?2.2

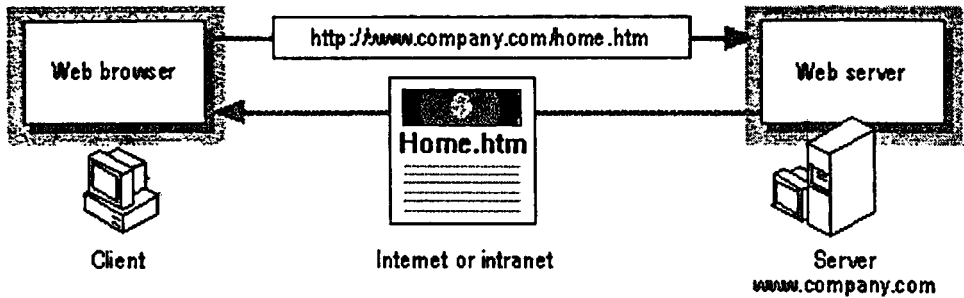
ตารางที่ 3.2 แสดงตัวอย่างของการร้องขอแบบต่างๆ

3.4.4 Web Server Response

Web server ตอบสนองต่อการร้องขอจาก Web Browser โดยการคืน HTML page ซึ่ง page นี้มี 3 ชนิดดังต่อไปนี้ คือ

3.4.4.1 Static page

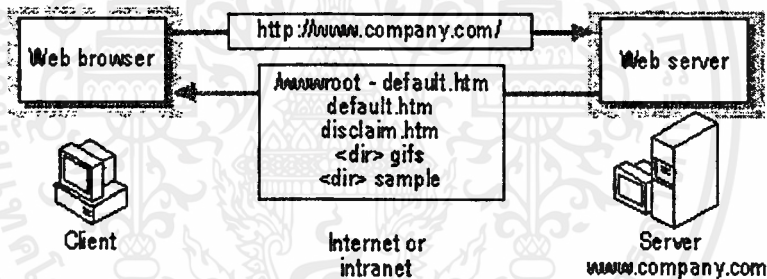
Static page คือ HTML page ที่ได้จัดเตรียมไว้รอการร้องขอ Web Server จะคืน HTML page ให้แก่ผู้ใช้โดยไม่มีกรกระทำพิเศษใดๆ ผู้ใช้สามารถร้องขอ static page นี้ได้โดยการพิมพ์ URL (ในรูปคือ <http://www.company.com/home.htm>) หรือการคลิกที่ตัวลิงค์ที่เชื่อมโยงไปยัง URL นี้ การร้องขอจะถูกส่งไปยัง Server และ Server จะตอบสนองโดยการคืน Static HTML page



รูปที่ 3.3 แสดงการทำงานของ Static Page

3.4.4.2 Dynamic page

Dynamic page ถูกสร้างเพื่อตอบสนองการร้องขอจากผู้ใช้ Web Browser จะเก็บค่าข้อมูลโดยการสร้าง page ที่ประกอบด้วย text box,เมนู,และ check box ให้ผู้ใช้กรอก เมื่อผู้ใช้คลิกปุ่มแล้วข้อมูลที่ผู้ใช้กรอกก็จะถูกส่งไปยัง Web Server โดยที่ Server อาจส่งข้อมูลนั้นให้แก่แอปพลิเคชันเพื่อประมวลผลหรือ ค้นหา หรือ เก็บข้อมูลในฐานข้อมูล Server จะคืนค่าเป็น HTML page ให้แก่ผู้ใช้



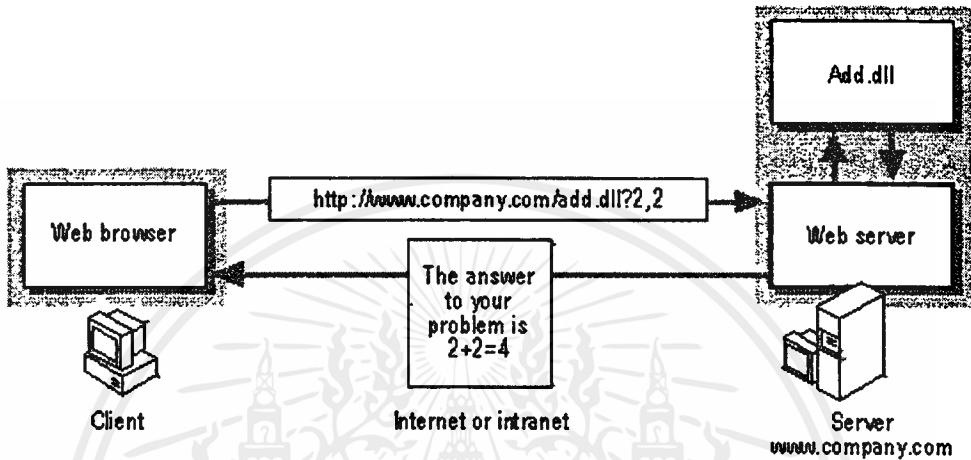
รูปที่ 3.4 แสดงการทำงานของ Dynamic Page

3.4.4.3 Directory listing

เมื่อผู้ใช้ร้องขอข้อมูล(ค้นหาข้อมูล)โดยไม่ได้ระบุชื่อไฟล์ในการค้นหา Server ก็จะคืนค่ารายชื่อไคเร็กทอรีทั้งหมดในรูปแบบของ HTML page เพื่อที่ผู้ใช้จะได้สามารถไปยังไฟล์ที่ต้องการได้โดยการคลิกที่ชื่อไฟล์นั้นๆ

3.4.5 ไฟล์ HTML Rxtension (htx)

เพื่อคืนค่าข้อมูลให้แก่ Client ดังนั้นต้องทำการรวม ไฟล์ htx ,ไฟล์ idc เข้ากับข้อมูลของ ODBC โดยข้อมูลที่รวมกันนี้จะถูก ส่งไปพร้อมกับ HTTP header มาตรฐาน (สถานะ 200 OK , Content-Type, และอื่นๆ) และส่งไปยัง WWW ซึ่งต่อไปจะส่งกลับคืนมายัง Client



รูปที่ 3.5 แสดงการเชื่อมต่อของระบบอินเทอร์เน็ตผ่าน ODBC

ไฟล์ htx เป็นเอกสาร HTML ประเภทหนึ่งที่มี tag อยู่ในเครื่องหมาย <%> หรือ <!--%--> ซึ่งเครื่องหมายนี้ในไฟล์ idc ใช้ในการเพิ่มข้อมูลแบบไดนามิกให้แก่เอกสาร รูปแบบของไฟล์ htx นั้นจะใช้สำหรับการคืนข้อมูล คำสงวนที่ใช้มีทั้งหมด 6 ตัว (begindetail, enddetail, if, else, endif, and "%z") ซึ่งใช้ควบคุมวิธีการรวมข้อมูลจากฐานข้อมูลเข้ากับไฟล์ htx ชื่อคอลัมน์ในฐานข้อมูลจะใช้ในการระบุว่าข้อมูลตัวใดที่จะถูกคืนกลับไปยังเอกสาร HTML ตัวอย่าง เช่น บรรทัดต่อไปนี้เป็นตัวอย่างของไฟล์ htx ที่รวมเข้ากับข้อมูลจากคอลัมน์ Emailname ของทุกแถว <%begindetail%><%Emailname%><%enddetail%>

Sample.htx เป็นตัวอย่างของเอกสาร HTML ที่เก็บค่า tag ของ Internet Database Connector สำหรับข้อมูลที่รับคืนมาจากฐานข้อมูล(tag จะเป็นตัวทึบ)

```
<HTML>
<BODY>
<HEAD><TITLE>Authors and YTD Sales</TITLE></HEAD>
<%if idc.sales eq ""%>
<H2>Authors with sales greater than <I>5000</I></H2>
<%else%>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<H2>Authors with sales greater than <I><%idc.sales%></I></H2>
```

```
<%endif%>
```

```
<P>
```

```
<%begindetail%>
```

```
<%if CurrentRecord EQ 0 %>
```

```
Query results:
```

```
<B>Author YTD Sales<BR></B>
```

```
<%endif%>
```

```
<%au_lname%><%ytd_sales%>
```

```
<%enddetail%>
```

```
<P>
```

```
<%if CurrentRecord EQ 0 %>
```

```
<I><B>Sorry, no authors had YTD sales greater than <I><%idc.sales%>.</B>
```

```
<P>
```

```
<%else%>
```

```
<HR>
```

```
<I>
```

The Web page you see here was created by merging the results of the SQL query with the template file Sample.htx.

```
<P>
```

The merge was done by the Microsoft Internet Database Connector and the results were returned to this Web browser by the Microsoft Internet Information Server.

```
</I>
```

```
<%endif%>
```

```
</BODY>
```

```
</HTML>
```

The <%begindetail%> and <%enddetail%> sections delimit where rows returned from the database will appear in the document. Columns returned from the query are surrounded by <%%>, such as

<%au_lname%> and <%ytd_sales%> in this example.

ส่วนที่อยู่ระหว่าง tag <%begindetail%> และ tag <%enddetail%> ใช้ในการกำหนดขอบเขตของแถวข้อมูลจากฐานข้อมูลที่จะแสดงในเอกสาร กอถันที่ฐานข้อมูลคืนมาจะอยู่ในเครื่องหมาย <%%> เช่น จากตัวอย่าง <%au_lname%> และ <%ytd_sales%>

3.4.6 HTML

ภาษาHTML(Hypertext Markup Language) เป็นภาษาไฮเปอร์บายอิงข้อมูลที่แสดงบนเว็บ รวมถึงข้อมูลประเภทมัลติมีเดีย และ การใช้ไฮเปอร์ลิงค์ HTMLเป็นภาษาที่มีลักษณะของโค้ดเป็นไฟล์ตัวอักษร ในมาตรฐานของรหัสแอสกี

3.4.6.1 เครื่องมือในการสร้าง HTML

ไฟล์ HTML สามารถสร้างโดยใช้ text editor ตัวใดก็ได้ เช่น Notepad เป็นต้น แต่ในปัจจุบันก็ได้มีการพัฒนา เครื่องมือขึ้นมาสำหรับสร้างไฟล์ html โดยเฉพาะ เช่น Microsoft FrontPage , Internet Assistant for Microsoft Word , Hotdog , Web Edit ต่างๆ เป็นต้น ทั้งนี้เพื่อให้ง่ายต่อการพัฒนา แอปพลิเคชัน

3.4.6.2 Publishing html files and others files format

ข้อมูลที่แสดงบน Web สามารถเป็นได้ทั้งภาพ และ เสียง รวมทั้งการใช้ไฮเปอร์ลิงค์ เชื่อมโยง Web ไปไฟล์อื่นๆ ผู้ใช้ที่อยู่ห่างไกลจำเป็นต้องมีแอปพลิเคชันที่เหมาะสมในการแสดงข้อมูลเหล่านั้น เช่น ถ้ามีการเชื่อมโยงข้อมูลไปยังไฟล์เอกสารบน Microsoft Word

เครื่องคอมพิวเตอร์ที่สามารถแสดงเอกสารนี้ได้จำเป็นต้องมีโปรแกรม Microsoft Word บนเครื่องคอมพิวเตอร์ และ เมื่อผู้ใช้ทำการคลิกที่ตัวลิงค์ เอกสารเวิร์คก็่จะปรากฏบนเครื่องคอมพิวเตอร์

เมื่อทำการสร้างไฟล์ Html แล้วอาจจะทำการคัดลอกไปเก็บไว้ยังไครเรกทอรี InetPub\Wwwroot หรือ เปลี่ยนค่าโฮมไดเรกทอรี(Home Directory)ให้เป็นไครเรกทอรีที่เก็บไฟล์นี้ก็ได้

3.4.7 MIME

ถ้าในweb site นั้นประกอบด้วยไฟล์หลากหลายรูปแบบ ก็จำเป็นต้องใช้ Multipurpose Internet Mail Extension (MIME) เพื่อจับคู่ไฟล์แต่ละแบบ เพราะ ถ้าไม่มีการติดตั้งค่า MIME สำหรับไฟล์แต่ละแบบแล้ว browser จะไม่สามารถเรียกไฟล์เหล่านั้นขึ้นมาดูได้ ซึ่งสามารถตั้งค่านี้ได้ ใน Registry Editor Publishing Dynamic Applications

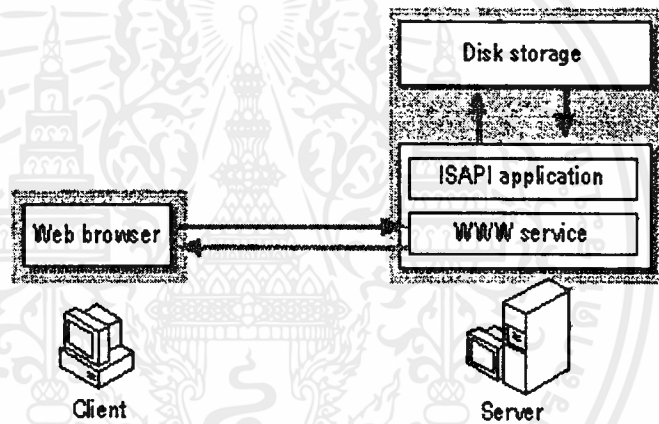
ลักษณะที่น่าสนใจในการใช้ Microsoft Internet Information Server คือ ความสามารถในการพัฒนาแอปพลิเคชันที่จะเริ่มดำเนินการทำงานเมื่อผู้ใช้เพียงแค่คลิกที่ตัวลิงค์ หรือ กรอกและส่งแบบฟอร์ม แอปพลิเคชันเหล่านี้สามารถเขียนได้โดยภาษาต่างๆ เช่น C หรือ Perl การสร้างแอปพลิเคชัน หรือ สคริปต์

แอปพลิเคชันที่สามารถโต้ตอบกับผู้ใช้ได้ส่วนใหญ่จะถูกไค้คบนภาษาที่เป็น32 bits เช่น C , Perl หรือ Windows NT batch files(มีนามสกุลเป็น .bat หรือ .cmd) เมื่อจะไค้คแอปพลิเคชัน จำเป็นต้องใช้ Interface 1 ใน 2 ตัวต่อไปนี้ คือ , Microsoft Internet Server Application Programming Interface (ISAPI) หรือ Common Gateway Interface (CGI)

แอปพลิเคชันที่ใช้ ISAPI จะถูกคอมไพล์ให้เป็น dynamic-link libraries (DLLs) ซึ่งจะถูกลoad โดย WWW ขณะเปิดเครื่อง เหตุผลนี้ทำให้ โปรแกรมแบบ ISAPI มักจะเร็วกว่าแอปพลิเคชันที่เขียนโดย CGI

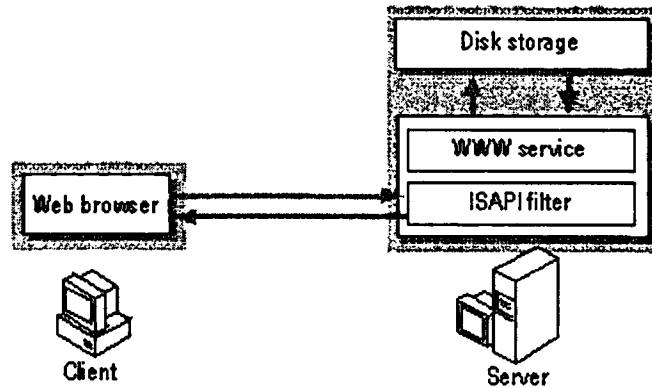
ISAPI สำหรับ Windows NT สามารถใช้สร้างแอปพลิเคชันที่จะเริ่มการทำงานเมื่อผู้ใช้เพียงแค่คลิกที่คัวลิ้งค์ หรือกรอกและส่งแบบฟอร์ม แอปพลิเคชันเหล่านี้จะรับข้อมูลจากผู้ใช้ , ทำงานตามโปรแกรม และ คืนค่ากลับมายังผู้ใช้ในรูปแบบของ Html Page หรือ อาจจะส่งค่าไปเก็บไว้ที่ฐานข้อมูล

ISAPI สามารถใช้ในการสร้างแอปพลิเคชันที่ทำงานเหมือนเป็น DLL บน Web Server ได้ ถ้าเปรียบเทียบระหว่างโปรแกรมที่เขียนโดยใช้ CGI กับ ISAPI ก็จะพบว่า แอปพลิเคชันที่เป็น ISAPI นั้นมีประสิทธิภาพที่ดีกว่า เพราะว่า แอปพลิเคชันจะถูกโหลดมาเก็บไว้ที่หน่วยความจำในขณะที่ Server ทำงาน

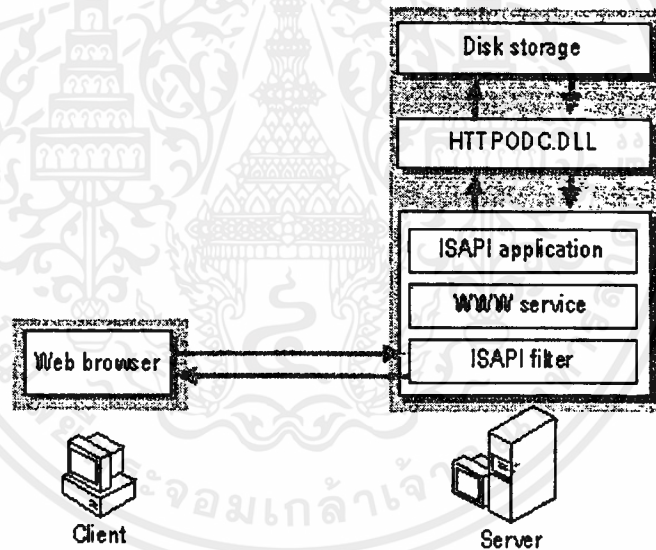


รูปที่ 3.6 ลักษณะอื่นๆของ ISAPI

อนุญาตให้มีการรับความต้องการเข้ามาก่อน และ ส่งผลลัพธ์คืนที่หลังการทำงานของแอปพลิเคชันได้ และ ยังอนุญาตให้มีการกำหนดการเรียกใช้งาน HTTP เฉพาะสำหรับแต่ละ site



รูปที่ 3.7 Internet Database Connection ทำงานร่วมกับ ISAPI filters และ ISAPI Application เพื่อสร้าง Web ที่สลับซับซ้อนขึ้นได้



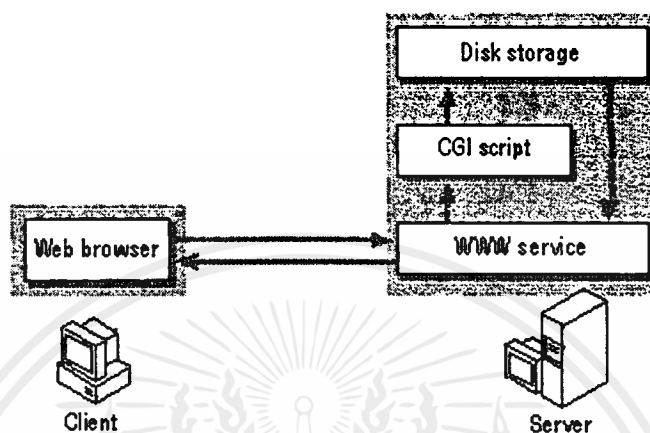
รูปที่ 3.8 การทำงานของ ISAPI ขณะที่ติดต่อกับฐานข้อมูล

3.4.8 Common Gateway Interface

Common Gateway Interface (CGI) คือ ชุดของรายละเอียดในการส่งข้อมูลต่างๆระหว่าง client web browser , web server และ แอปพลิเคชันที่เป็น CGI client web browser จะเป็นผู้เริ่มต้นการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของแอปพลิเคชัน โดยการกรอกแบบฟอร์ม หรือ การคลิกที่ตัวลิงค์ แอปพลิเคชันที่เป็น CGI นี้ก็เหมือนกับแอปพลิเคชันที่เป็น ISAPI คือ สามารถรับข้อมูลจากผู้ใช้ ทำงานตามโปรแกรม และ ส่งค่าคืนในรูปแบบของ HTML page หรือ อาจจะส่งค่าข้อมูลนั้น ไปเก็บไว้ที่ฐานข้อมูลก็ได้



รูปที่ 3.9 CGI System

3.4.8.1 การทำงานระหว่าง Client, Server และ แอปพลิเคชัน มีทั้งหมด 5 ขั้นตอน ดังนี้

1. Client ส่งการร้องขอไปยัง Server Client browser สามารถสร้างการร้องขอส่งไปยัง Server ได้ 2 วิธี ดังนี้

- GET Client จะส่งข้อมูลไปยัง server โดยต่อท้าย URL
- POST client จะส่งข้อมูลผ่านฟิลด์ข้อมูลของ HTTP

Client จะกำหนดค่าเริ่มต้นในการทำงานโดยคลิกที่ต่างๆ ต่อไปนี้

- ไฮเปอร์ลิงค์ที่สามารถเริ่มต้นการทำงานของแอปพลิเคชันได้โดยตรง
- ปุ่ม submit บนฟอร์ม
- สิ่งที่เรียกขึ้นมาโดยการใช้วิธี GET

2. Server รับการร้องขอจาก Client URL ที่ Client ส่งไปยัง Server นั้นจะเก็บชื่อของแอปพลิเคชันที่เป็น CGI เพื่อที่จะให้ทราบว่าจะต้องทำงานกับแอปพลิเคชันใด Server จะเปรียบเทียบค่านามสกุลของไฟล์นั้นกับค่าที่จับคู่ไว้ใน Registry key ที่เครื่อง Server ทั้งนี้เพื่อเริ่มต้นการทำงานของแอปพลิเคชันที่ไฟล์นั้นต้องใช้ เช่น ถ้า Script Map มีข้อมูลเข้าเป็นไฟล์ .cmd หรือ .bat จะเริ่มต้นการทำงานของ Cmd.exe และสำหรับข้อมูลที่เป็นไฟล์ .idc จะเริ่มต้นการทำงานของ Internet Database Connection

3. Server ส่งการร้องขอไปยังแอปพลิเคชัน Server จะส่งข้อมูลไปให้แก่แอปพลิเคชันที่เป็น CGI ในลักษณะของตัวแปรสภาพแวดล้อม (environment variables) ตัวแปรเหล่านี้บางตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั้นขึ้นกับ Server ตัวแปรส่วนใหญ่ที่ได้มาจาก Client และมีความสัมพันธ์กับทั้ง client browser และ การร้องขอที่ส่งไป

4.แอปพลิเคชัน คืนค่าข้อมูลให้แก่ Server แอปพลิเคชันจะทำงานตามที่โปรแกรมไว้ และ จะคืนค่าข้อมูลให้แก่ Server ผ่านทาง สายข้อมูลออกมาตรฐาน (Standard Output Stream , STDOUT)

5.Server คืนค่าข้อมูลให้แก่ Client Server รับค่าข้อมูลจาก STDOUT แล้วทำการเพิ่ม ส่วนหัวของHTTP และส่งHttp Message กลับไปยัง Client

3.4.8.2 CGI และ Internet Information Server

ถึงแม้ว่า WWW สนับสนุนการทำงานของ CGI มาตรฐาน แต่ก็มีข้อพึงระวังในการใช้งาน CGI บน Internet Information Server ดังต่อไปนี้

- แอปพลิเคชันที่เป็น CGI 32 bits เท่านั้นที่สามารถทำงานร่วมกับ WWW ได้ใน เวอร์ชันนี้
- ตัวแปร REMOTE_USER ค่าจะไม่เป็นปัจจุบันเมื่อผู้ใช้ล็อกเข้าระบบในฐานะของผู้ใช้นิรนาม
- ตัวแปรทุกตัวที่สร้างขึ้นสำหรับแอปพลิเคชันที่เป็น ISAPI จะถูกส่งไปยัง แอปพลิเคชันที่เป็น CGI ในรูปของตัวแปรสิ่งแวดล้อม

สังเกตว่า แอปพลิเคชันที่เป็น CGI ส่วนใหญ่จะเป็นแอปพลิเคชันที่ทำงานอิสระ ซึ่งเมื่อทำการเปรียบเทียบกับแอปพลิเคชันที่เป็น ISAPI

3.4.8.3 ระบบรักษาความปลอดภัย

กฎในการรักษาความปลอดภัย คือ อนุญาตให้มีการเอ็กซิกิวข้อมูลที่ไคเรกทอรีเสมือนที่ซึ่งเก็บ แอปพลิเคชันทั้งที่เป็น CGI หรือ Internet Server API(ISAPI) ได้เพียงครั้งละ1เท่านั้น

ข้อแนะนำที่ดีอีกอย่างหนึ่ง คือ ควรจัดค่า Script mapping เพื่อให้แน่ใจว่า interpreter ที่ใช้นั้นถูกต้องเหมาะสม (ตัวอย่าง เช่น Cmd.exe) เมื่อ Client ทำการร้องขอไฟล์ที่เอ็กซิกิวได้

ไคเรกทอรี WWW ควรอนุญาตให้มีการเข้าถึงแบบอ่านได้อย่างเดียวเท่านั้น ไฟล์ที่สามารถเอ็กซิกิวได้ที่ดาวน์โหลดมาจากไคร์ฟ NTFS ก็เช่นเดียวกันก็ควรมีการเข้าถึงแบบอ่านได้อย่างเดียวเท่านั้น

Batch file สามารถทำงานได้เสมือนเป็น ไฟล์ CGI ที่สามารถเอ็กซิกิวได้ แต่ควรทำด้วยความระมัดระวัง

3.4.8.4 การติดตั้งแอปพลิเคชันบน Internet Information Server

เมื่อทำการเขียนแอปพลิเคชันเรียบร้อยแล้ว ก็ต้องนำแอปพลิเคชันนั้นไปเก็บไว้ที่ ไคเรกทอรีสคริปต์ และต้องมั่นใจว่าทุกขั้นตอนทำงานภายใต้ชื่อที่ได้รับอนุญาตให้สามารถทำงานได้ และถ้าแอปพลิเคชันมีการโต้ตอบกับไฟล์อื่นๆแล้วชื่อที่ใช้ต้องได้รับอนุญาตให้สามารถทำงานกับไฟล์นั้นๆได้ โดยทั่วไปแอปพลิเคชันทำงานบนชื่อ IUSR_COMPUTERNAME ซึ่งได้รับอนุญาตให้เอ็กซิกิวได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าแอปพลิเคชันสามารถทำงานได้โดยไม่ต้องการข้อมูลจากผู้ใช้ ก็จะง่ายต่อการสร้าง ไฟล์ HTML เพื่อเชื่อมโยงไปยังแอปพลิเคชันนั้น แต่ถ้าหากแอปพลิเคชันนี้ต้องการข้อมูลจากผู้ใช้ ก็สามารถทำได้โดยใช้ฟอร์ม HTML หรือ อาจจะส่งผ่านไปทาง URL

เนื่องจากมีภาษามากมายในการสร้างแอปพลิเคชัน ดังนั้น Internet Information Server ก็จะใช้นามสกุลเป็นตัวแยกความแตกต่าง เพื่อตัดสินใจว่าต้องใช้ Interpreter ตัวใด ตารางนี้เป็นตัวอย่างของนามสกุลเทียบกับตัว Interpreter และนอกจาก3ตัวนี้ ยังสามารถเพิ่มตัว Interpreter ใหม่ได้โดยใช้ Registry Editor

Extension	Default Interpreter
.bat, .cmd	Cmd.exe
.idc	Httpodbc.dll
.exe, .com	System

ตารางที่ 3.3 เป็นตัวอย่างของนามสกุลเทียบกับตัว Interpreter

3.4.9 Security Implications

เมื่อได้อนุญาตให้ผู้ใช้สามารถใช้อัปพลิเคชัน ก็จะเกิดความเสี่ยงตามมาจากกาที่ผู้โจกรรมข้อมูล พยายามเข้าถึงระบบ Microsoft Internet Information Server ได้มีวิธีการลดความเสี่ยงนี้ 2 วิธี คือ

3.4.9.1. ไคเร็กทอรีสคริปต์(Scripts)

เป็นที่เก็บแอปพลิเคชัน administrator เท่านั้นที่สามารถเพิ่มโปรแกรมในไคเร็กทอรีซึ่งได้กำหนดแล้วว่าให้สามารถเอ็กซคิวติวได้อย่างเดียวเท่านั้น ดังนั้นผู้ใช้ทั่วไปไม่สามารถทำสำเนาโปรแกรมนี้ได้ มีข้อเสนอว่าควรจะอนุญาตให้ผู้ใช้ IUSR_COMPUTERNAME สามารถทำการอ่าน และเอ็กซคิวติวไคเร็กทอรีที่เกี่ยวข้องกับโฟ้วเดอร์เสมือนเท่านั้น และ administrator เท่านั้นที่จะควบคุมการทำงานทั้งหมด สคริปต์ในภาษา Perl (นามสกุล .pl) และ ไฟล์ idc (นามสกุล .idc หรือ .htx) จะต้องการทั้งการเข้าถึงแบบอ่านและเอ็กซคิวติว แต่อย่างไรก็ตามเพื่อป้องกันการคิดตั้งไฟล์ที่ไม่ปลอดภัยบนเครื่องคอมพิวเตอร์ สามารถทำได้โดยการไม่อนุญาตให้เข้าถึงแบบเขียน

3.4.9.2. ถ้ามีการตั้งค่าให้ผู้ใช้ WWW ทั้งหมดเป็นผู้ใช้นิรนาม(Anonymous)

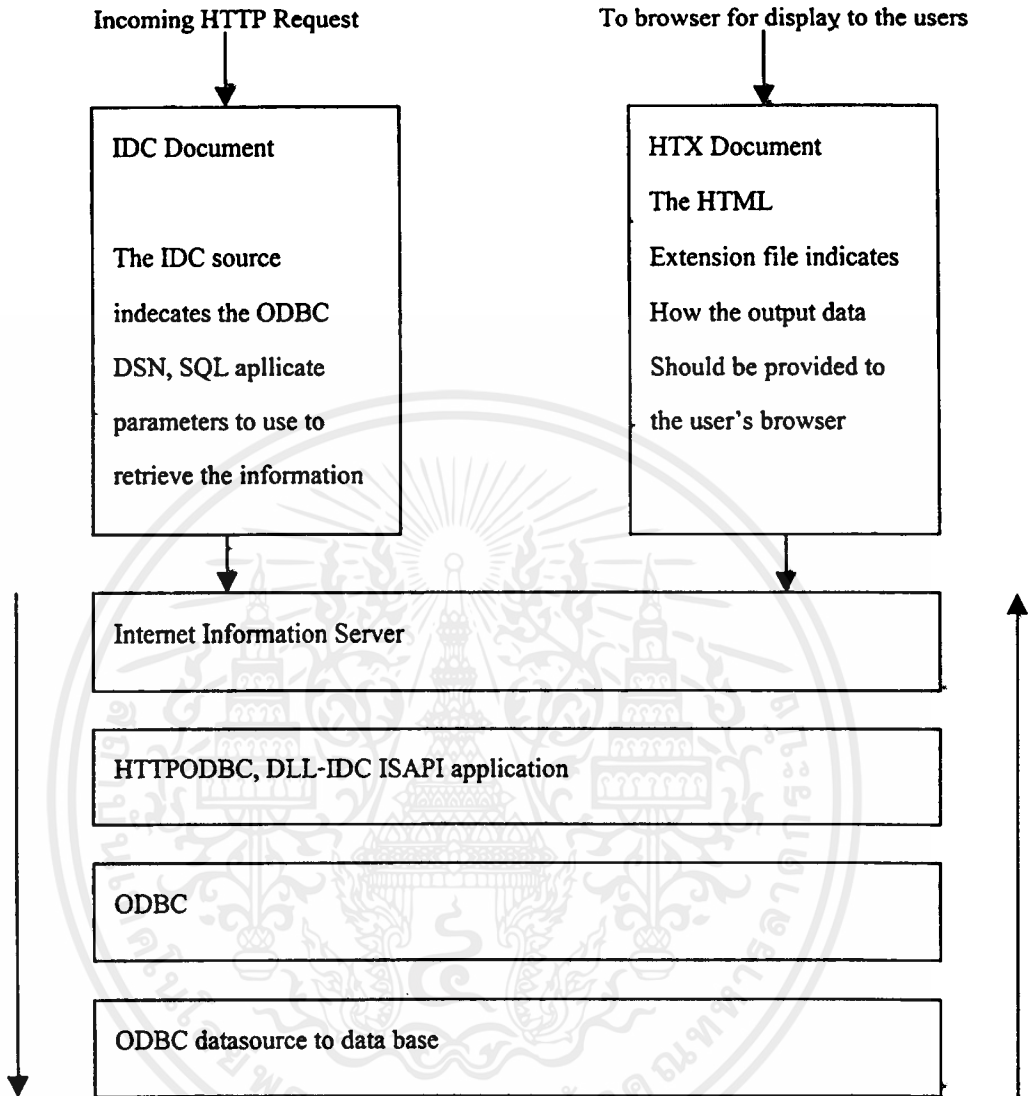
การร้องขอทั้งหมดจะให้อยู่ในชื่อ IUSR_COMPUTERNAME ซึ่งโดยทั่วไปแล้ว ชื่อนี้จะไม่สามารถทำการลบหรือเปลี่ยนแปลงไฟล์โดยใช้ Windows NT File System(NTFS) ได้ถ้า administrator ไม่อนุญาต

3.5 การทำงานของระบบ

ส่วนที่ 1 User Interface เนื่องมาจากโครงงานนี้เราใช้โปรแกรม Web Browser เป็น User Service ดังนั้นจึงใช้ภาษา HTML , Java Script และ Microsoft FrontPage ซึ่งเป็นโปรแกรม Web Editor มาทำ Homepage ซึ่งเราจะนำค่า Key ที่ได้มาจากการร้องขอของ User นั้นในรูปแบบ HTML file มาเก็บไว้ที่ Web Browser เพื่อที่จะตอบสนองความต้องการของ User เหตุผลที่เลือกใช้โปรแกรม Internet Explorer ในการติดต่อกับผู้ใช้บริการเพราะว่าโปรแกรมนี้มีรูปแบบการใช้งานโปรแกรมที่สะดวก ซึ่งสามารถตอบสนอง HTML page ทั้งแบบ static page และ dynamic page ได้ดี (static page คือ รูปแบบคงที่ของไฟล์ HTML ที่ทำการเก็บไว้ที่ Web site ที่พร้อมใช้งาน ส่วน dynamic page คือ รูปแบบไม่คงที่ของไฟล์ HTML ที่ Server ต้องสร้างขึ้นด้วย Information ที่จะทำการตอบสนองแก่ผู้ใช้งาน หรืออาจจะเป็นรายการไฟล์ที่มีอยู่หลายๆไฟล์สำหรับการเลือกใช้ folder บน Web site

ส่วนที่ 2 เป็นส่วนของ *Web Server* โดยเราจะใช้ Application ของ Internet Information Server ทำหน้าที่เป็น Web Server โดยจะติดต่อกับ SQL ผ่าน ODBC เพื่อตอบสนอง SQL request ที่เข้ามาจากผู้ให้บริการ

ส่วนที่ 3 เป็นส่วนของ *SQL Server* โดยจะทำหน้าที่รับเอา SQL request ที่ได้มาทำการ query Data Base โดยจะตรวจสอบสิทธิต่างๆ ของผู้ใช้บริการที่เข้ามาให้ตามสิทธินั้นๆ โดยที่การ query database ของ SQL Server นั้นเราจะทำการเขียนฟังก์ชันต่างๆ ซึ่งโดยทั่วไปก็จะเป็นการ Insert , Delete , Update , Search , Sort ผู้ใช้แต่ละระดับก็มีสิทธิในการเข้าถึงข้อมูลที่แตกต่างกันออกไป ทั้งนี้ขึ้นอยู่กับ การจัดการจาก Data Base Administrator จะเป็นผู้มีหน้าที่ในการจัดการกำหนดสิทธิต่าง ๆ เพื่อป้องกัน ความผิดพลาด และยังเป็นการรักษาความปลอดภัยของระบบอีกด้วย

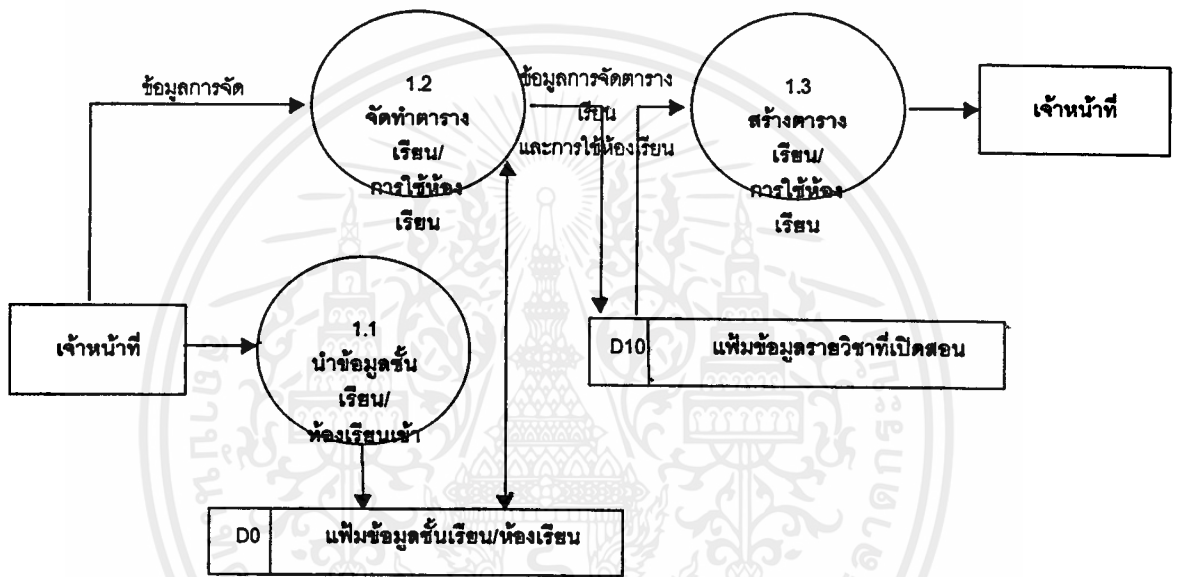


ODBC Access using HTML and HTML Extensions with the Internet Database Connector

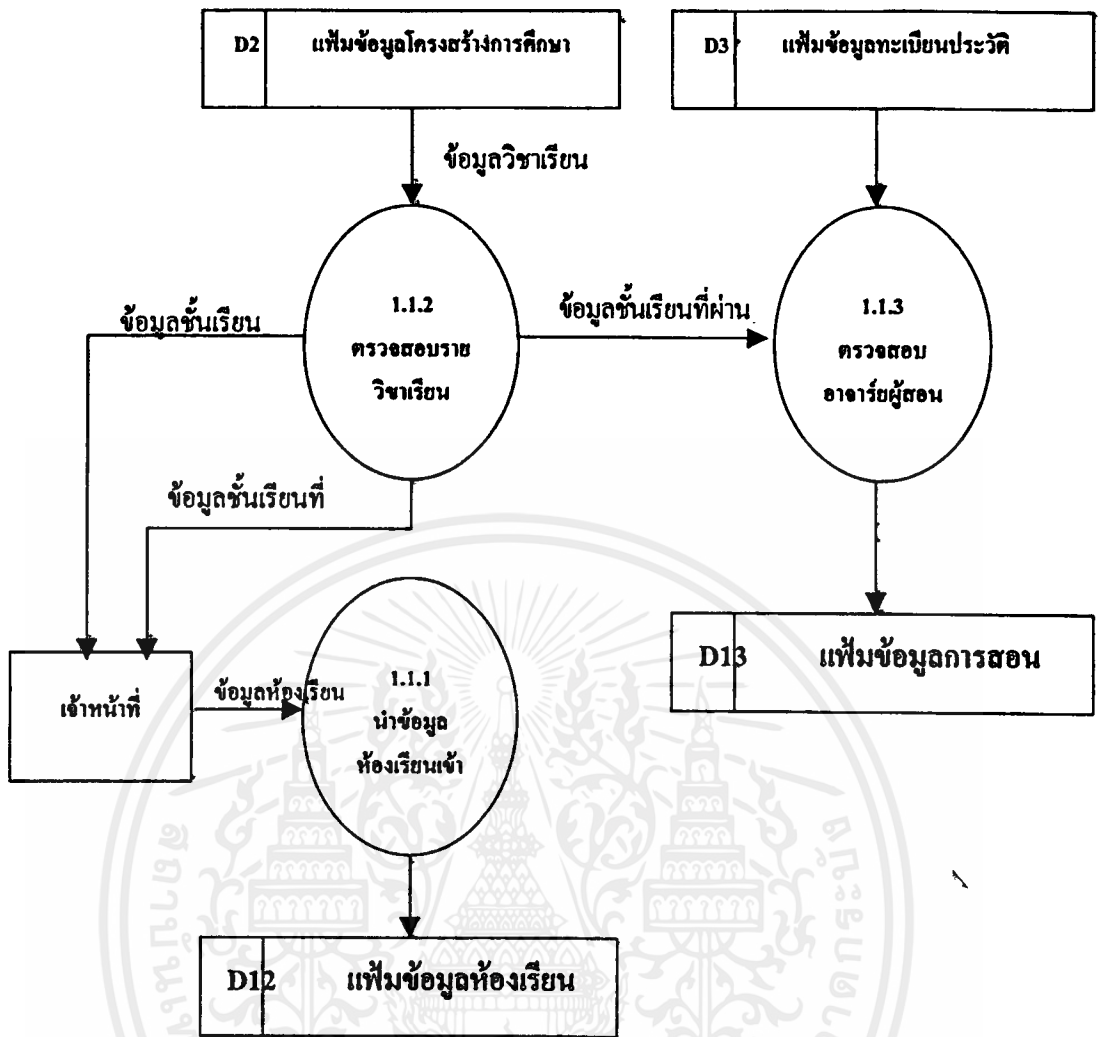
รูปที่ 3.10 รูปแบบของการ Setup Internet Database Connector

หลังจากที่ได้ศึกษาถึงระบบการจัดการใช้ห้องเรียนและตารางเรียนแล้ว ก็จะนำมาออกแบบโดยเขียนเป็นโมเดลที่มาทำการอธิบายระบบ โดยจะใช้ Data Flow Diagram มาอธิบายถึงลักษณะการไหลของข้อมูลตั้งแต่เริ่มกระบวนการจนเสร็จสิ้นกระบวนการ ซึ่งจะมีประโยชน์อย่างมากเมื่อเวลาทำการพัฒนาโครงการให้เป็นที่ไปตามลำดับขั้นที่ถูกต้องและเหมาะสม เพราะการอธิบายด้วยรูปจะสามารถทำให้เห็นภาพโดยรวมได้มากกว่า

ส่วนฐานข้อมูลนั้น โครงการนี้ ออกแบบฐานข้อมูลโดยใช้ NIAM model มาแสดงความสัมพันธ์ของข้อมูลซึ่งเมื่อทำการ map ออกมาแล้วจะได้ตารางฐานข้อมูลที่เป็น Fifth Normal Form (5NF) เลย โดยไม่ต้องทำการ Normalized อีก



รูปที่ 3.11 แสดง Data Flow Diagram ชั้นที่ 1 ของระบบการจัดการเรียนและตารางการใช้ห้องเรียน



รูปที่ 3.12 แสดง Data Flow Diagram ชั้นที่ 2 ของระบบการจัดการรายเรียนและตารางการใช้ห้องเรียน

หลังจากที่ได้วิเคราะห์ระบบโดยละเอียด แล้วก็ทำการออกแบบเค้าโครงจากนั้นเมื่อก่อร่างเป็นตารางได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่ได้วิเคราะห์ระบบโดยละเอียด แล้วก็ทำการออกแบบค่าเบส จากนั้นแมปออกมาเป็นตารางได้ดังนี้

ตารางที่ 3.4 Room Table

Field Name	Data Type	Data Length	KEY
RoomCode	Char	20	PK
BuildingCode	Char	7	
RoomFloorNo	Int	4	

ตารางที่ 3.5 Building Table

Field Name	Data Type	Data Length	KEY
BuildingCode	Char	7	PK
BuildingName	Char	30	
BuildingPlace	Char	150	

ตารางที่ 3.6 Class Table

Field Name	Data Type	Data Length	KEY
SubjId	Char	5	PK
AcademicYear	Char	4	PK
AcademicTerm	Char	1	PK
LecGroupNo	Char	2	PK
LabGroupNo	Char	2	PK
ClassDay	Char	4	PK
ClassTimeBegin	Char	5	PK
ClassTimeEnd	Char	5	PK
RoomCode	Char	20	PK
ClassMaxStd	Int	4	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.7 Subject Table

Field Name	Data Type	Data Length	KEY
SubjId	Char	8	PK
SubjNameThai	Char	50	
SubjNameEng	Char	50	
SubjDescripThai	Char	255	
SubjDescripEng	Char	255	
SubjDeptId	Char	3	
SubjGroupTypeId	Int	4	
SubjLecCredit	SmallInt	2	
SubjLabCredit	SmallInt	2	
SubjYearBegin	Char	4	
SubjTermBegin	Char	1	
SubjYearEnd	Char	4	
SubjTermEnd	Char	1	
YearStudy	Char	1	

ตารางที่ 3.8 Teacher Table

Field Name	Data Type	Data Length	KEY
TeacherId	Char	8	PK
TeacherTitleThai	Char	6	
TeacherNameThai	Char	15	
TeacherSnameThai	Char	25	
TeacherPositionThai	Char	18	
DeptId	Char	3	FK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.9 TeacherClass Table

Field Name	Data Type	Data Length	KEY
SubjId	Char	6	PK
LecGroupNo	Char	2	PK
LabGroupNo	Char	2	PK
AcademicYear	Char	4	PK
AcademicTerm	Char	1	PK
TeacherId	Char	5	PK

ตารางที่แนบมาได้นี้จะใช้เก็บข้อมูลของระบบจัดชั้นเรียน และห้องเรียน ในส่วนของคณาจารย์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การใช้งาน และการทดลอง ระบบทะเบียน ในส่วนการจัดตารางเรียนและตารางสอน

4.1 ฮาร์ดแวร์และซอฟต์แวร์ที่ใช้

4.1.1 เครื่องที่ทำหน้าที่ควบคุมเครือข่าย (Server)

- คอมพิวเตอร์เพนเทียมโปร 180 เม็กกะเฮิร์ต
- หน่วยความจำ 64 เม็กกะไบต์
- Cache Memory L1 256 กิโลไบต์
- งานเก็บข้อมูลชนิดแข็งชนิด IDE ขนาด 2 กิกะไบต์
- ซีพียูความเร็ว 18 เท่า
- Floppy Disk Drive ขนาด 3.5 นิ้ว
- Monitor SVGA 14 นิ้ว
- Keyboard ไทย-อังกฤษ 104 คีย์
- Mouse ชนิด 2 ปุ่ม
- ติดตั้งระบบปฏิบัติการ Windows NT 4.0 Thai Edition
- ติดตั้ง Microsoft SQL Server 6.5
- Internet Information Server

4.1.2 เครื่องที่ทำหน้าที่เป็นลูกข่าย

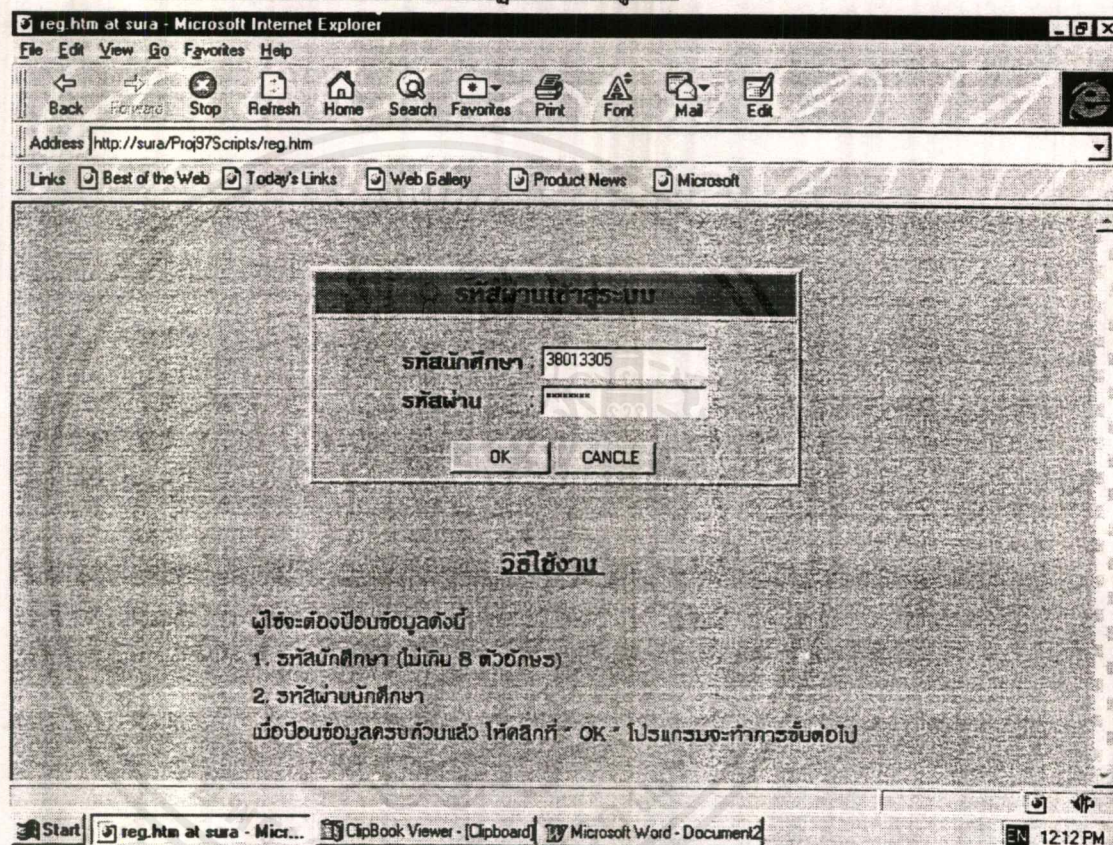
- คอมพิวเตอร์เพนเทียม 133 เม็กกะเฮิร์ต MMX
- หน่วยความจำ 32 เม็กกะไบต์
- Cache Memory L1 512 กิโลไบต์
- งานเก็บข้อมูลชนิดแข็งชนิด IDE ขนาด 1.6 กิกะไบต์
- ซีพียูความเร็ว 24 เท่า
- Floppy Disk Drive ขนาด 3.5 นิ้ว
- Monitor SVGA 15 นิ้ว
- Keyboard ไทย-อังกฤษ 104 คีย์ , Mouse ชนิด 2 ปุ่ม
- ติดตั้งระบบปฏิบัติการ Windows 95 Thai Edition
- ติดตั้ง Microsoft Internet Explorer 4.0

4.2 การเข้าใช้งานระบบ

ผู้ใช้งานจำเป็นต้องป้อนข้อมูลผู้ใช้งานก่อนการเข้าใช้งานระบบลงทะเบียน ข้อมูลที่ป้อน คือ รหัสนักศึกษา และ รหัสผ่าน

4.2.1 การเข้าใช้งาน

- ป้อน URL (<http://sura/Proj97Scripts/reg.htm>) เพื่อเข้าสู่หน้าจอกำหนดข้อมูลผู้ใช้งานก่อนการเข้าใช้งานระบบ ซึ่งจะปรากฏหน้าจอดังรูปที่ 1

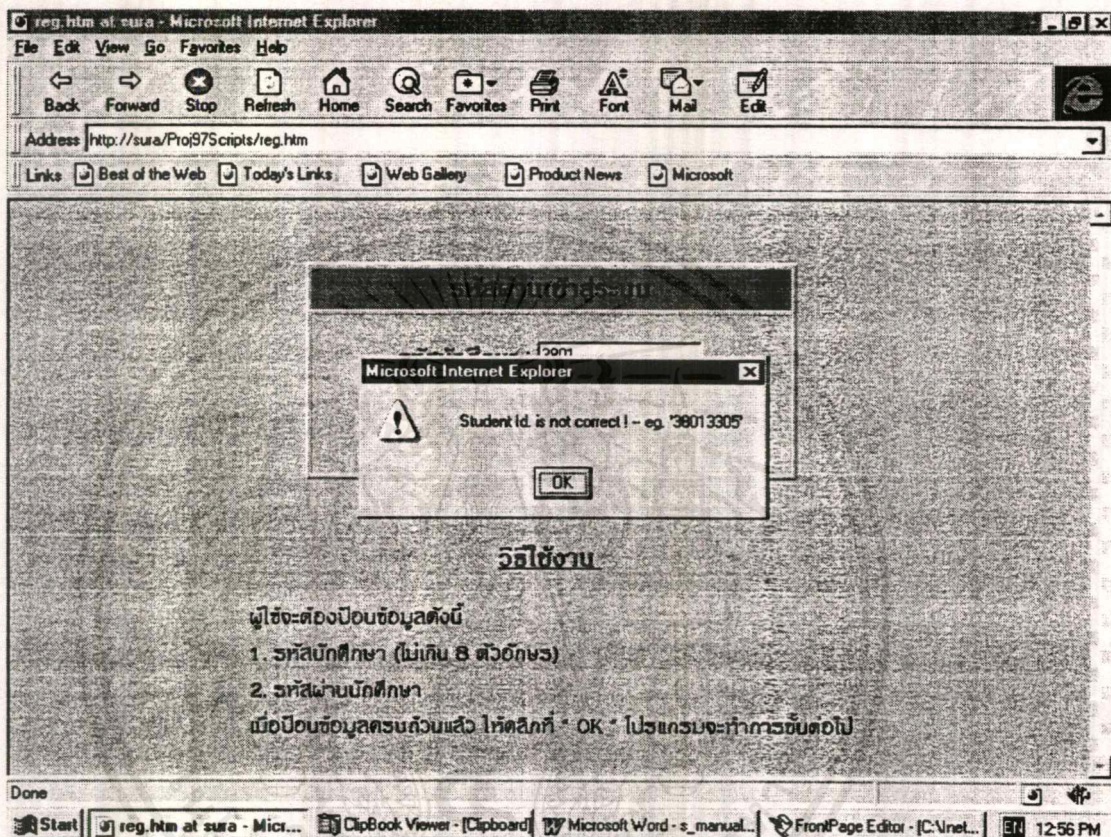


รูปที่ 4.1 แสดงหน้าจอกำหนดข้อมูลผู้ใช้งาน

- ป้อนข้อมูลผู้ใช้งาน เช่น รหัสนักศึกษา คือ 38013305 และ รหัสผ่าน คือ 38013305
- คลิกปุ่ม OK เพื่อตกลง หรือ คลิกปุ่ม CANCEL เมื่อไม่ต้องการ
- ผลลัพธ์มีได้ 3 อย่าง คือ ข้อมูลถูกต้อง , ข้อมูลไม่ถูกต้อง หรือ ข้อมูลไม่ครบถ้วน

A. ในกรณีที่ผู้ใช้ป้อนข้อมูลไม่ครบถ้วน

- รหัสนักศึกษาไม่ครบ 8 ตัวอักษร
- รหัสนักศึกษาไม่ใช่ตัวเลขทั้งหมด

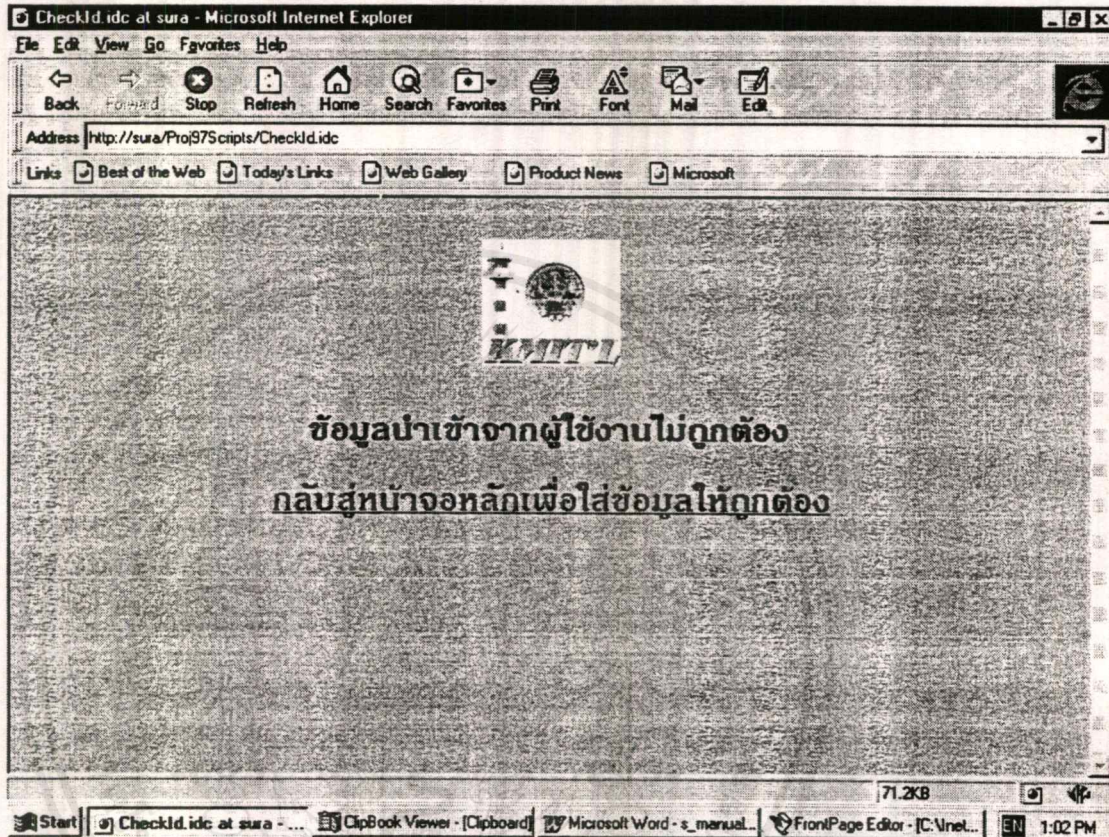


รูปที่ 4.2 แสดงหน้าจอการรับข้อมูลผู้ใช้ไม่ครบถ้วน

B. ในกรณีที่ผู้ใช้ป้อนข้อมูลไม่ถูกต้อง

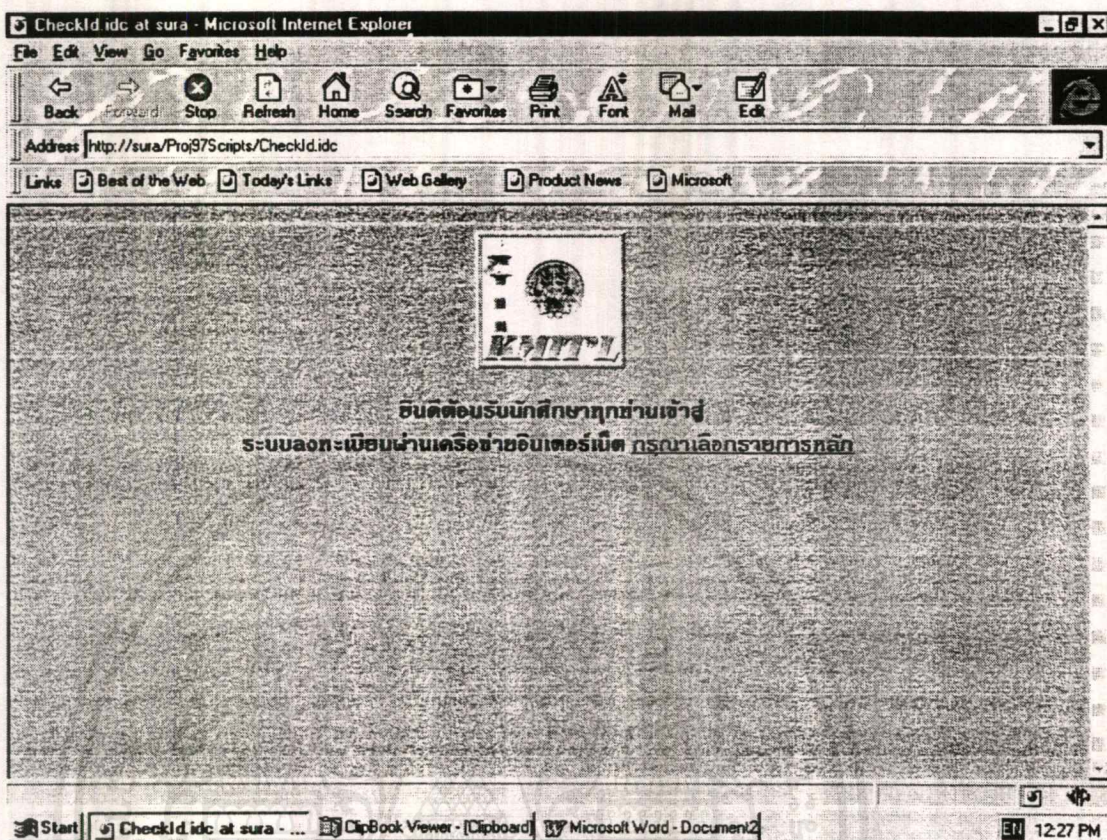
- รหัสนักศึกษาไม่ถูกต้อง เช่น 3709610145
- รหัสผ่านไม่ถูกต้อง

จะปรากฏหน้าจอดังรูป



รูปที่ 4.3 แสดงหน้าจอการรับข้อมูลผู้ใช้ไม่ถูกต้อง

- C. ในกรณีที่ผู้ใช้ป้อนข้อมูลถูกต้องและครบถ้วน เช่น รหัสนักศึกษา คือ 38013305 และ รหัสผ่าน คือ 38013305 จะปรากฏหน้าจอการทำงานต่อไปดังรูป



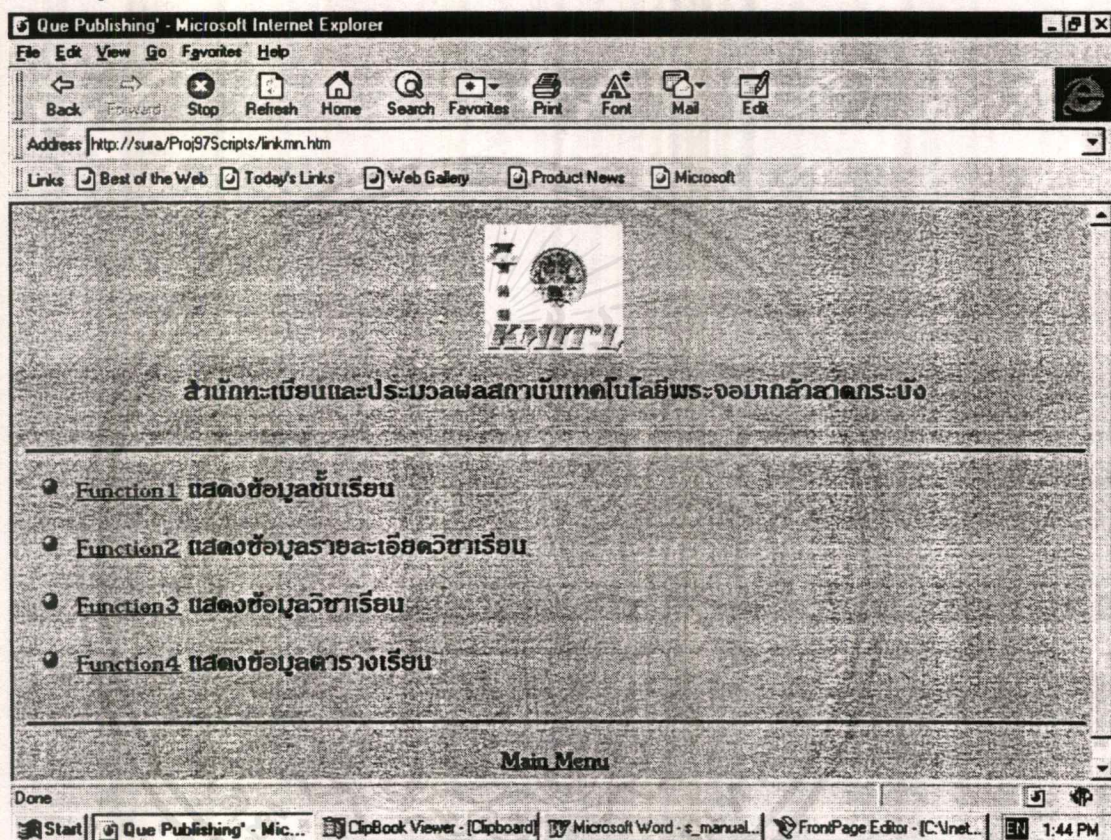
รูปที่ 4.4 แสดงหน้าจอการรับข้อมูลผู้ใช้ถูกต้อง

4.2.2 เมนูหลัก

เป็นส่วนการทำงานหลัก ซึ่งแบ่งการทำงานออกเป็น 4 ส่วน ดังนี้

- การแสดงข้อมูลชั้นเรียน
- การแสดงข้อมูลรายละเอียดวิชาเรียน
- การแสดงข้อมูลวิชาเรียน
- การแสดงข้อมูลตารางเรียน

ดังรูป



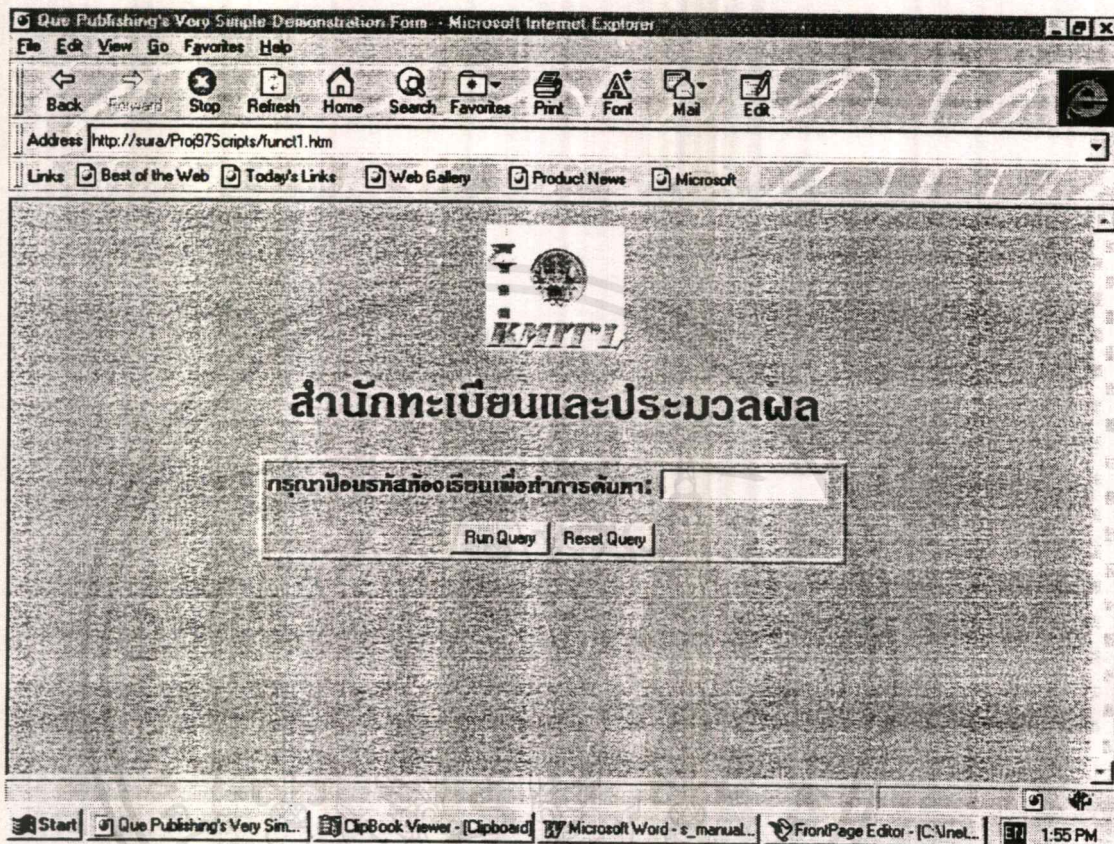
รูปที่ 4.5 แสดงหน้าจอเมนูหลัก

4.2.2.1 การแสดงข้อมูลชั้นเรียน

- เลือกการทำงานที่ 1 จากหน้าจอเมนูหลัก (รูปที่ 5) จะปรากฏหน้าจอการทำงานดังรูปที่ 6
- ป้อนรหัสห้องเรียนทั้งหมด หรือ บางส่วน เช่น A หรือ A-201
- คลิกปุ่ม Run Query เพื่อตกลง หรือ คลิกปุ่ม Reset Query เพื่อลบข้อมูลที่ป้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ผลลัพธ์ มีได้ 3 อย่าง คือ ค้นพบข้อมูลหลายแถว , 1แถว หรือ ไม่พบข้อมูลที่สอดคล้อง กับข้อมูลที่ป้อน ซึ่งข้อมูลที่แสดง จะประกอบด้วย รหัสชั้นเรียน , ชั้นที่ , ชนิดของห้องเรียน , ความจุ , อาคาร , ชื่ออาคาร และ สถานที่
- เมื่อเสร็จสิ้นการทำงานผู้ใช้สามารถกลับสู่หน้าจอเมนูหลัก ได้โดยการคลิกที่ **กลับสู่เมนูหลัก** ซึ่งจะปรากฏหน้าจอดังรูปที่ 4.5



รูปที่ 4.6 แสดงหน้าจอการรับข้อมูล เพื่อการแสดงผลข้อมูลชั้นเรียน

A. ในกรณีที่พบข้อมูลมากกว่า 1 แถว

เมื่อผู้ใช้ป้อนทำการข้อมูลเพียงบางส่วน เช่น A หรือ B เป็นต้น ซึ่งจะทำให้ได้ผลลัพธ์ เป็นข้อมูลหลายแถว ดังรูป

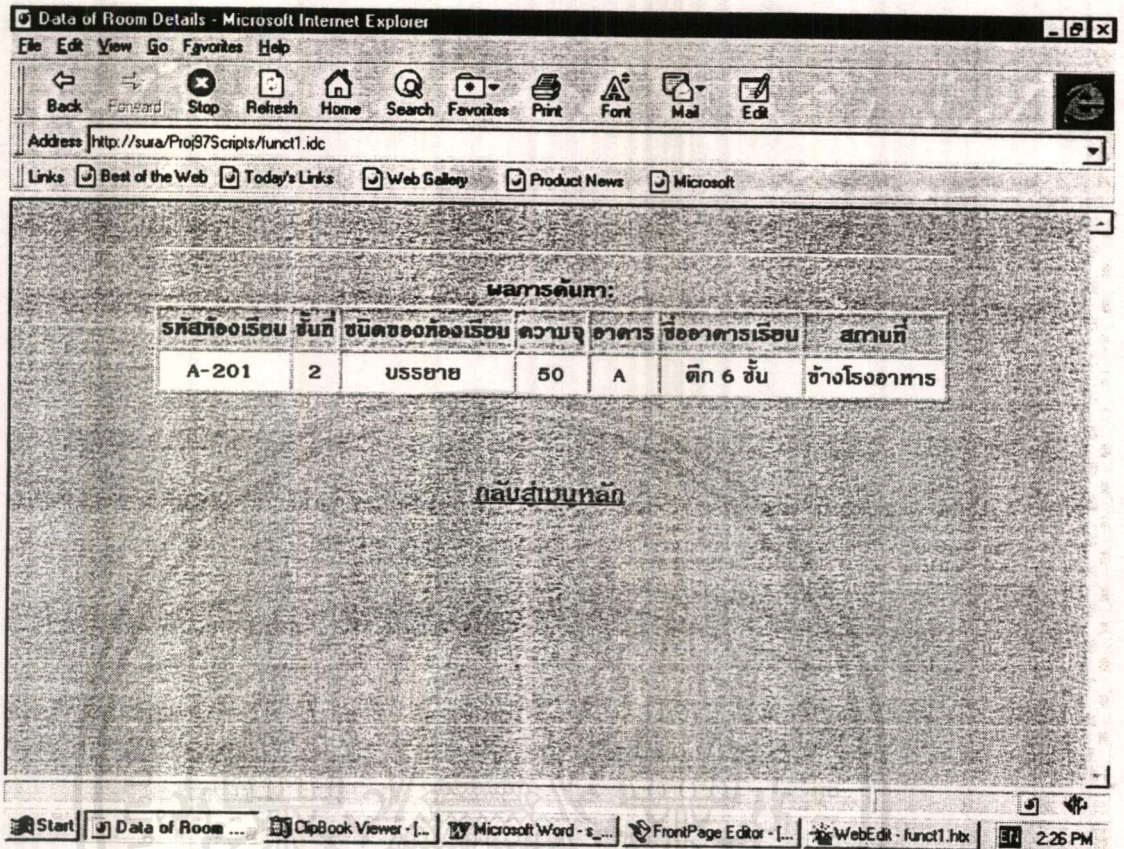
ผลการค้นหา:

รหัสห้องเรียน	ชั้นที่	ชนิดของห้องเรียน	ความจุ	อาคาร	ชื่ออาคารเรียน	สถานที่
A-201	2	บรรยาย	50	A	ตึก 6 ชั้น	ข้างโรงอาหาร
A-202	2	บรรยาย	50	A	ตึก 6 ชั้น	ข้างโรงอาหาร
A-203	2	บรรยาย	50	A	ตึก 6 ชั้น	ข้างโรงอาหาร
A-204	2	บรรยาย	50	A	ตึก 6 ชั้น	ข้างโรงอาหาร
A-205	2	บรรยาย	80	A	ตึก 6 ชั้น	ข้างโรงอาหาร
A-401	4	บรรยาย	50	A	ตึก 6 ชั้น	ข้างโรงอาหาร
A-402	4	บรรยาย	50	A	ตึก 6 ชั้น	ข้างโรงอาหาร
A-403	4	บรรยาย	50	A	ตึก 6 ชั้น	ข้างโรงอาหาร
A-404	4	บรรยาย	50	A	ตึก 6 ชั้น	ข้างโรงอาหาร
A-405	4	บรรยาย	50	A	ตึก 6 ชั้น	ข้างโรงอาหาร

รูปที่ 4.7 หน้าจอแสดงผลการเรียกค้นข้อมูลชั้นเรียนในกรณีที่มีข้อมูลหลายแถว

B. ในกรณีที่พบข้อมูลเพียง 1 แถว

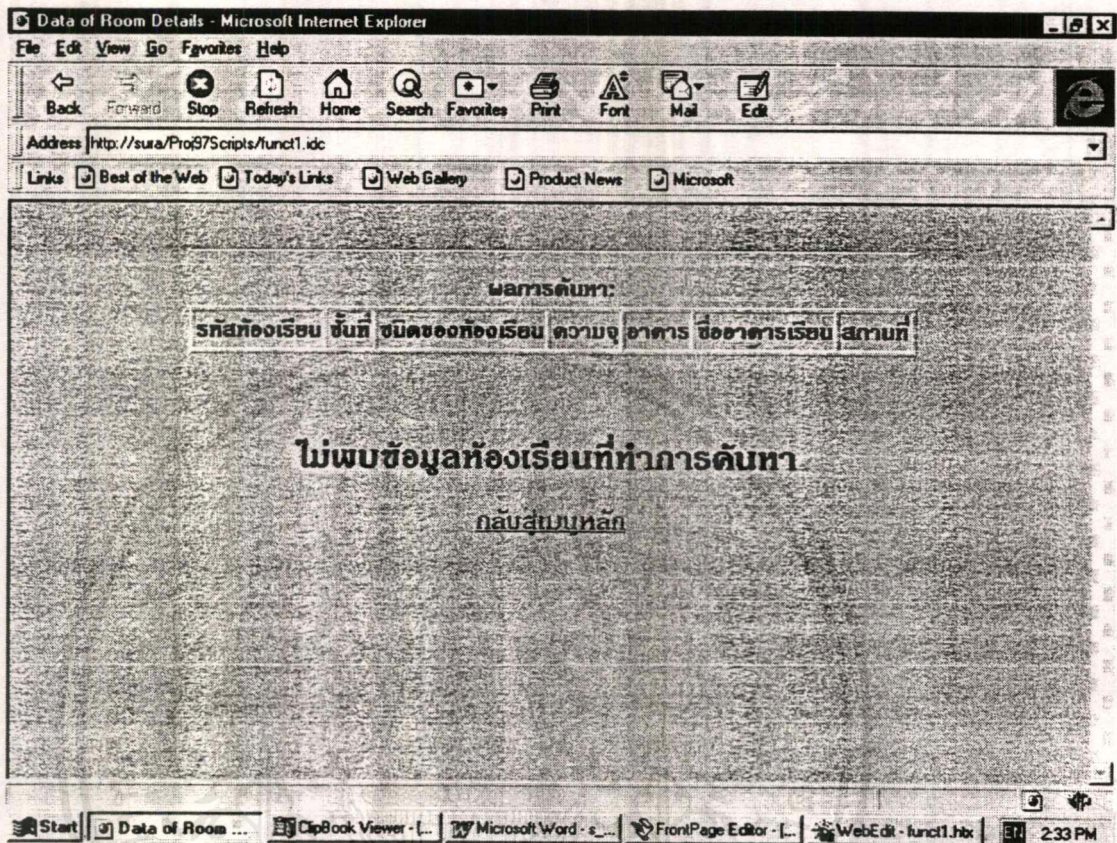
เมื่อผู้ใช้ป้อนทำการข้อมูลครบ เช่น A-201 เป็นต้น ซึ่งจะทำให้ได้ผลลัพธ์เป็นข้อมูลเพียง 1 แถว ดังรูป



รูปที่ 4.8 แสดงหน้าจอผลลัพธ์การเรียกค้นข้อมูลชั้นเรียนในกรณีที่ข้อมูลที่พบมีเพียง 1 แถว

C. ในกรณีที่ไม่พบข้อมูลที่สอดคล้องกับข้อมูลที่ผู้ใช้ป้อน

เมื่อผู้ใช้ป้อนรหัสที่ไม่ถูกต้อง เช่น Z-909 , P-506 เป็นต้น จะปรากฏหน้าจอเตือน การใส่ข้อมูลไม่ถูกต้อง ดังรูป



รูปที่ 4.9 แสดงหน้าจอการเรียกค้นข้อมูลชั้นเรียน ในกรณีที่ไม่มีข้อมูลที่ต้องการ

4.2.2.2 การแสดงข้อมูลรายละเอียดวิชาเรียน

ผู้ใช้สามารถสอบถามข้อมูลรายละเอียดวิชาเรียนได้ใน 2 ลักษณะ คือ

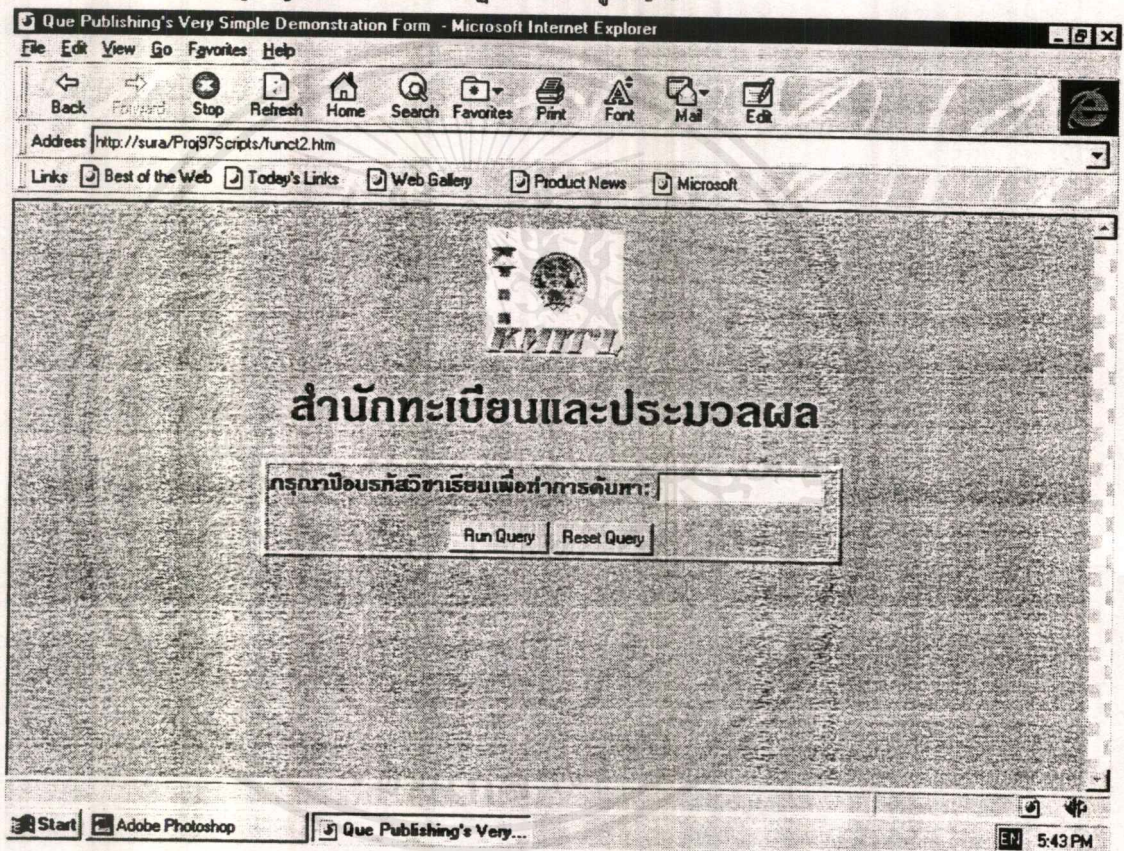
- การสอบถามโดยใช้รหัสวิชาเรียน
- การสอบถามโดยใช้ชื่อวิชาเรียน

ซึ่งข้อมูลที่แสดงจะประกอบด้วย

A. การสอบถามโดยใช้รหัสวิชาเรียน ขั้นตอนการใช้งาน

- เลือกการทำงานที่ 2 จากหน้าจอเมนูหลัก(รูปที่ 4.5) จะปรากฏหน้าจอการทำงานดังรูปที่ 4.10

- ป้อนรหัสวิชาเรียนทั้งหมด หรือ บางส่วน เช่น 0107 หรือ 01073110 โดยเมื่อผู้ใช้ทำการป้อนข้อมูลเพียงบางส่วนผลลัพธ์ที่ได้จะเป็นข้อมูลแถวแรกที่สอดคล้องกับข้อมูลที่ผู้ใช้ป้อน และเมื่อผู้ใช้ป้อนข้อมูลครบถ้วนผลลัพธ์ คือ ข้อมูลที่สอดคล้องกับข้อมูลที่ผู้ใช้ป้อน
- คลิกปุ่ม Run Query เพื่อตกลง หรือ คลิกปุ่ม Reset Query เพื่อลบข้อมูลที่ป้อน
- ผลลัพธ์มีได้ 2 อย่าง คือ พบ หรือ ไม่พบข้อมูลที่สอดคล้อง ดังรูปที่ 13 และ 14 ตามลำดับ
- เมื่อเสร็จสิ้นการทำงาน ผู้ใช้สามารถกลับสู่หน้าจอเมนูหลัก ได้โดยการคลิกที่ กลับสู่เมนูหลัก ซึ่งจะปรากฏหน้าจอ ดังรูปที่ 4.5



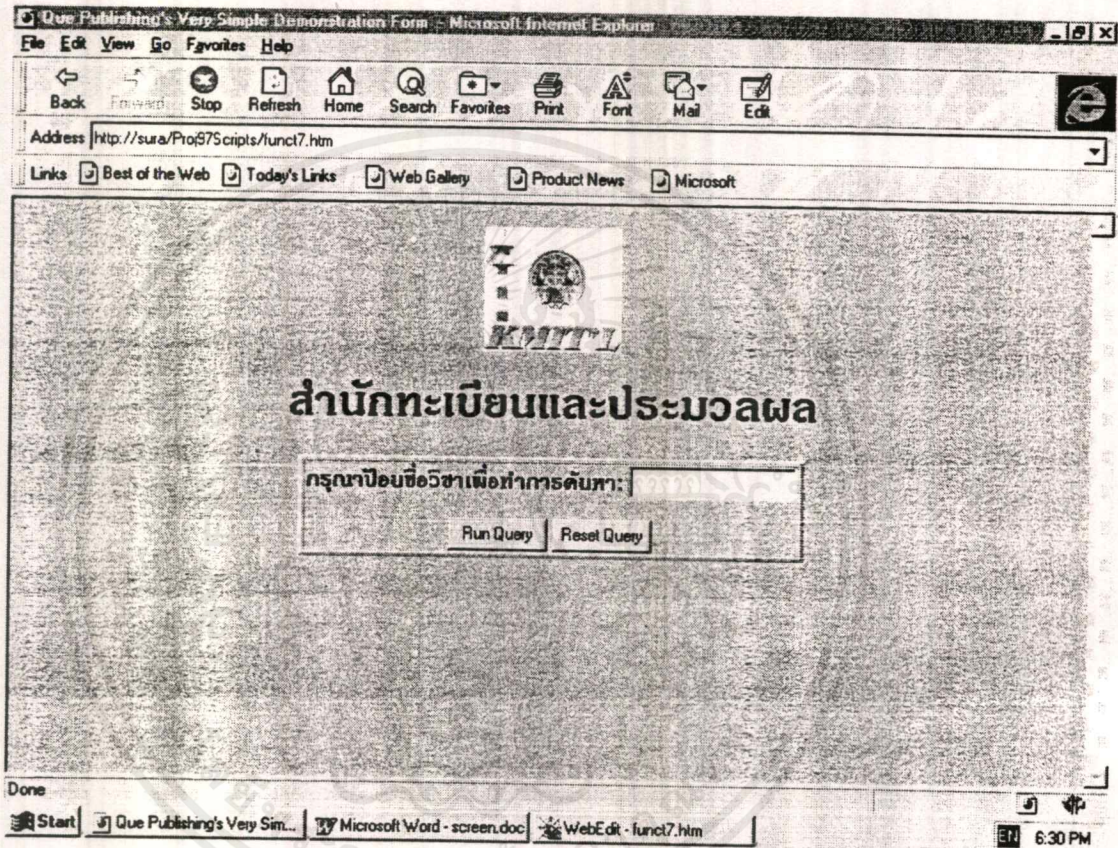
รูปที่ 4.10 หน้าจอการทำงานที่ 2 การแสดงข้อมูลรายละเอียดวิชาเรียนตามรหัสวิชา

- B. การสอบถามโดยใช้ชื่อวิชาเรียน ขั้นตอนการใช้งาน
- เลือกการทำงานที่ 7 จากหน้าจอเมนูหลัก(รูปที่ 5) จะปรากฏหน้าจอการทำงานดังรูปที่ 11
 - ป้อนชื่อวิชาเรียนทั้งหมด หรือ บางส่วน เช่น "Compiler" หรือ "Compiler Construction" ซึ่งตัวอักษรที่พิมพ์ทั้งตัวพิมพ์ใหญ่ และ ตัวพิมพ์เล็ก สามารถเรียกค้นข้อมูลได้เช่นเดียวกัน โดยเมื่อผู้ใช้ทำการป้อนข้อมูลเพียงบางส่วนผลลัพธ์ที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเป็นข้อมูลแถวแรกที่สอดคล้องกับข้อมูลที่ผู้ใช้ป้อน และเมื่อผู้ใช้ป้อนข้อมูลครบถ้วนผลลัพธ์ คือ ข้อมูลที่สอดคล้อง กับข้อมูลที่ผู้ใช้ป้อน

- คลิกปุ่ม Run Query เพื่อตกลง หรือ คลิกปุ่ม Reset Query เพื่อลบข้อมูลที่ป้อน
- ผลลัพธ์มีได้ 2 อย่าง คือ พบ หรือ ไม่พบข้อมูลที่สอดคล้อง ดังรูปที่ 12 และ 14 ตามลำดับ
- เมื่อเสร็จสิ้นการทำงาน ผู้ใช้สามารถกลับสู่หน้าจอเมนูหลัก ได้โดยการคลิกที่ กลับสู่เมนูหลัก ซึ่งจะปรากฏหน้าจอดังรูปที่ 4.5



รูปที่ 4.11 หน้าจอการทำงานที่ 2 การแสดงข้อมูลรายละเอียดวิชาเรียนตามชื่อวิชา

Authors Details - Microsoft Internet Explorer

File Edit View Go Favorites Help

Address <http://sura/Prof97Scripts/funct7.idc>

Links Best of the Web Today's Links Web Gallery Product News Microsoft

รหัสวิชา	01073110	ปีการศึกษา	2540	ภาคเรียนที่	2
ชื่อวิชา	Compiler Construction	หน่วยกิตบรรยาย	3	ปฏิบัติ	0
อาจารย์ผู้สอน					
รหัสอาจารย์	00000001	ชื่ออาจารย์ผู้สอน	วิบูลย์	นามสกุล	พร้อมพามิชัย
วัน-เวลาเรียน					
วันเรียน	WED	เวลาเริ่ม	09:00	เวลาเลิก	12:00
ห้องเรียน					
รหัสห้องเรียน	A-501	ชนิด	บรรยาย	ความจุ	50
ชั้น	5	ชื่อตึกเรียน	A	ชื่ออาคารเรียน	ตึก 6 ชั้น

รายละเอียดของวิชาเรียนรหัส compiler

กลับสู่เมนูหลัก

Start | Authors Details - Micr... | Microsoft Word - screen.doc | WebEdit - funct7.htm | 6:46 PM

รูปที่ 4.12 หน้าจอผลลัพธ์ของการเรียกค้นข้อมูลรายละเอียดชั้นเรียนตามชื่อวิชา เช่น Compiler

Authors Details - Microsoft Internet Explorer

File Edit View Go Favorites Help

Address http://sura/Proj97Scripts/lunct2.idc

Links Best of the Web Today's Links Web Gallery Product News Microsoft

รหัสวิชา	01073110	ปีการศึกษา	2540	ภาคเรียนที่	2
ชื่อวิชา	Compiler Construction	หน่วยกิตบรรณราช	3	ปฏิบัติ	0
อาจารย์ผู้สอน					
รหัสอาจารย์	00000001	ชื่ออาจารย์ผู้สอน	วิบูลย์	นามสกุล	พร้อมพานิชย์
วัน-เวลาเรียน					
วันเรียน	WED	เวลาเริ่ม	09:00	เวลาเลิก	12:00
ห้องเรียน					
รหัสห้องเรียน	A-501	ชนิด	บรรยาย	ความจุ	50
ชั้น	5	ชื่อตึกเรียน	A	ชื่ออาคารเรียน	ตึก 6 ชั้น

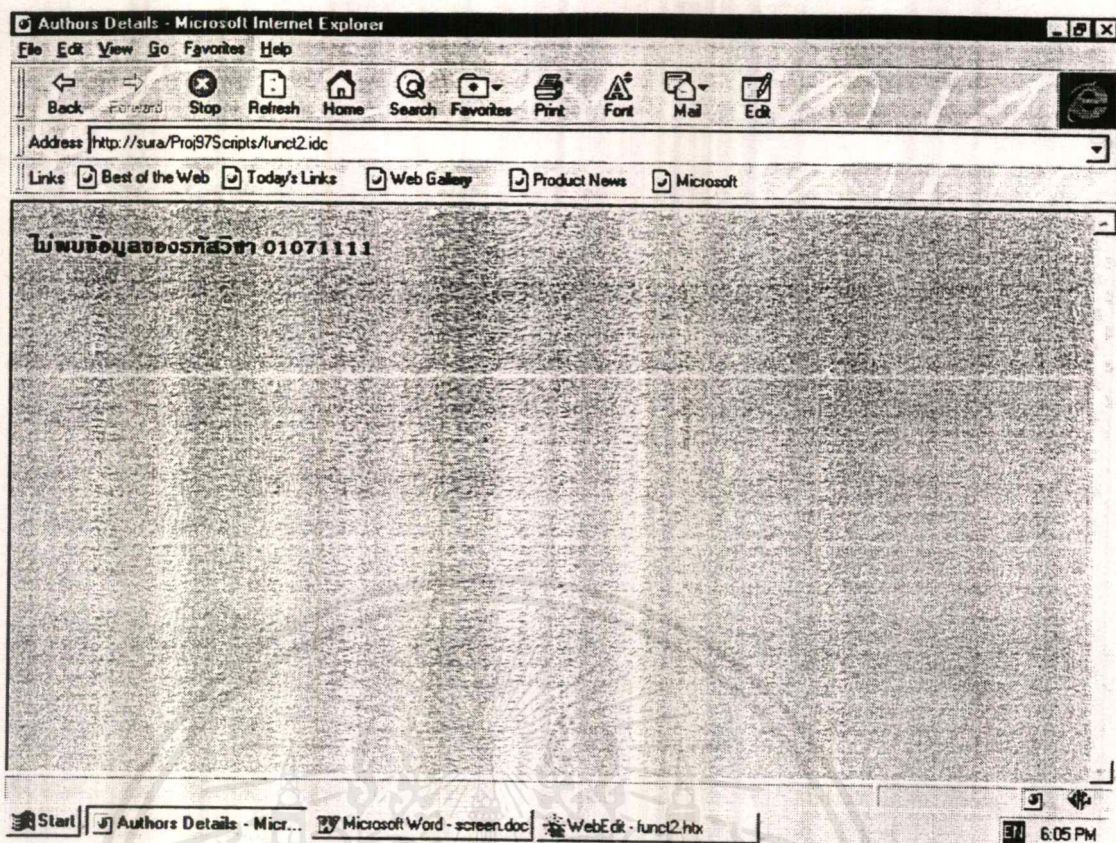
รายละเอียดของวิชาเรียนรหัส 01073110

กลับสู่เมนูหลัก

Start Authors Details - Micr... ClipBook Viewer - [Clipboard] Microsoft Word - Document1 5:46 PM

รูปที่ 4.13 หน้าจอผลลัพธ์ของการเรียกค้นข้อมูลรายละเอียดชั้นเรียนตามรหัสวิชา เช่น 01073110

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



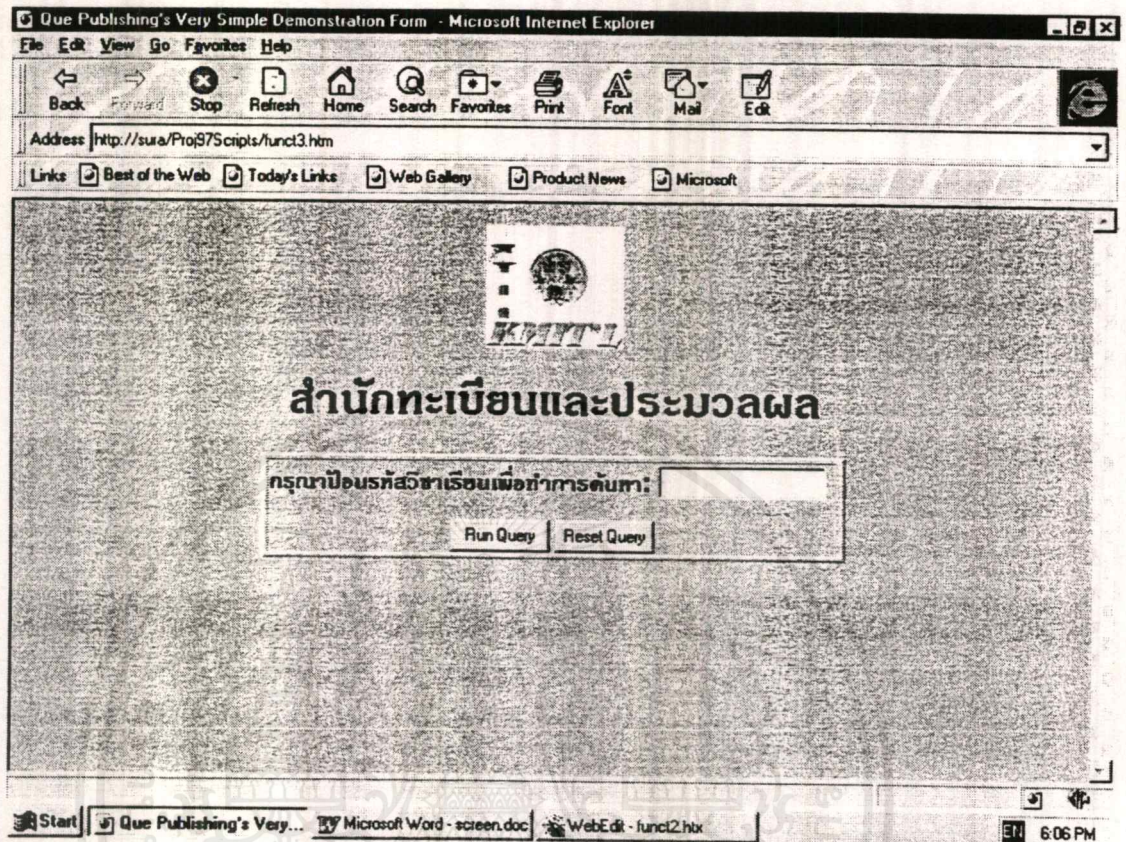
รูปที่ 4.14 หน้าจอผลลัพธ์ของการเรียกค้นข้อมูลรายละเอียดชั้นเรียนในกรณีที่ไม่พบข้อมูลที่สอดคล้องกับข้อมูลที่ผู้ใช้ป้อน เช่น 01071111

4.2.2.3 การแสดงข้อมูลวิชาเรียน

ข้อมูลผลลัพธ์ที่แสดง จะประกอบด้วย รหัสวิชาเรียน และ ชื่อวิชาเรียน
ขั้นตอนการใช้งาน

- เลือกการทำงานที่ 3 จากหน้าจอเมนูหลัก(รูปที่ 4.5) จะปรากฏหน้าจอการทำงานดังรูปที่ 4.15
- ป้อนรหัสวิชาเรียนทั้งหมด หรือ บางส่วน เช่น “0107” หรือ “01073110” โดยเมื่อผู้ใช้ทำการป้อนข้อมูล เพียงบางส่วนผลลัพธ์ที่ได้จะเป็นข้อมูลแถวแรก ที่สอดคล้องกับข้อมูลที่ผู้ใช้ป้อน และเมื่อผู้ใช้ป้อนข้อมูลครบถ้วนผลลัพธ์ คือ ข้อมูลที่สอดคล้องกับข้อมูลที่ผู้ใช้ป้อน ในกรณีที่ผู้ใช้ไม่ป้อนค่าข้อมูลรหัสวิชาเรียน ผลลัพธ์ที่ได้ คือ ข้อมูลวิชาเรียนทุกวิชา
- คลิกปุ่ม Run Query เพื่อคกกลง หรือ คลิกปุ่ม Reset Query เพื่อลบข้อมูลที่ป้อน
- ผลลัพธ์มีได้ 2 อย่าง คือ พบ หรือ ไม่พบข้อมูลที่สอดคล้อง

- เมื่อเสร็จสิ้นการทำงาน ผู้ใช้สามารถกลับสู่หน้าจอเมนูหลัก ได้โดยการคลิกที่ **กลับสู่เมนูหลัก** ซึ่งจะปรากฏหน้าจอดังรูปที่ 4.5



รูปที่ 4.15 แสดงหน้าจอการทำงานที่ 3 การเรียกค้นข้อมูลวิชาเรียน

A. ในกรณีที่ผู้ใช้ไม่ป้อนข้อมูลรหัสวิชาเรียน

เมื่อผู้ใช้ไม่ทราบรหัสวิชาเรียน ผู้ใช้สามารถเรียกค้นข้อมูลวิชาเรียนได้ทันทีโดยไม่ต้องป้อนรหัสวิชาเรียน ระบบจะทำการเรียกค้นข้อมูลทั้งหมด และจะปรากฏหน้าจอ ดังรูป

The screenshot shows a web browser window titled "Data of Room Details - Microsoft Internet Explorer". The address bar shows "http://sura/Pro97Scripts/funct3.idc". The main content area displays a table with the following data:

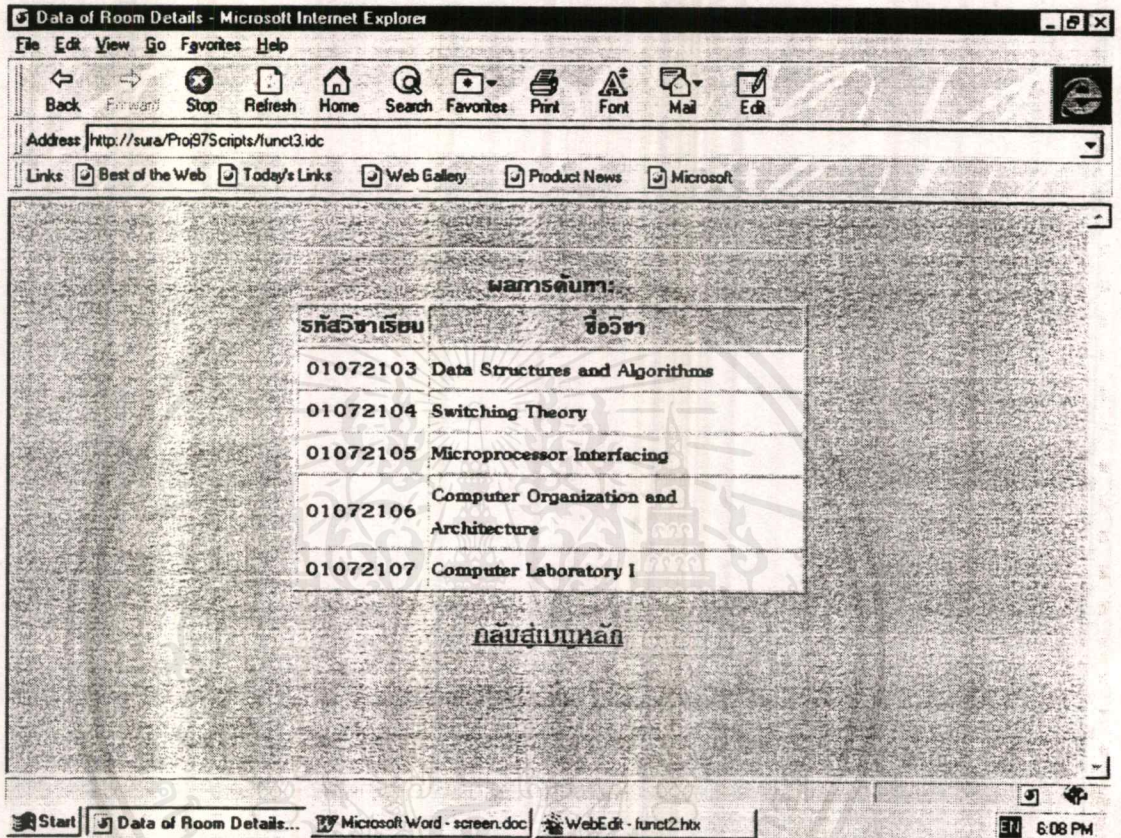
รหัสวิชาเรียน	ชื่อวิชา
01001004	Engineering Mathematics I
01001005	Engineering Mathematics II
01010001	Electromagnetics
01070001	Mathematics for Computer Engineering
01072103	Data Structures and Algorithms
01072104	Switching Theory
01072105	Microprocessor Interfacing
01072106	Computer Organization and Architecture
01072107	Computer Laboratory I
01073100	Training Job

The browser's taskbar at the bottom shows the Start button, several open applications including "Data of Room Details...", "Microsoft Word - screen.doc", and "WebEdit - funct2.htm", and the system clock showing 6:07 PM.

รูปที่ 4.16 แสดงหน้าจอผลลัพธ์การเรียกค้นข้อมูลวิชาเรียนในกรณีที่ผู้ใช้ไม่ได้ป้อนรหัสวิชาเรียน

B. ในกรณีที่ผู้ใช้ป้อนข้อมูลรหัสวิชาเรียนเพียงบางส่วน

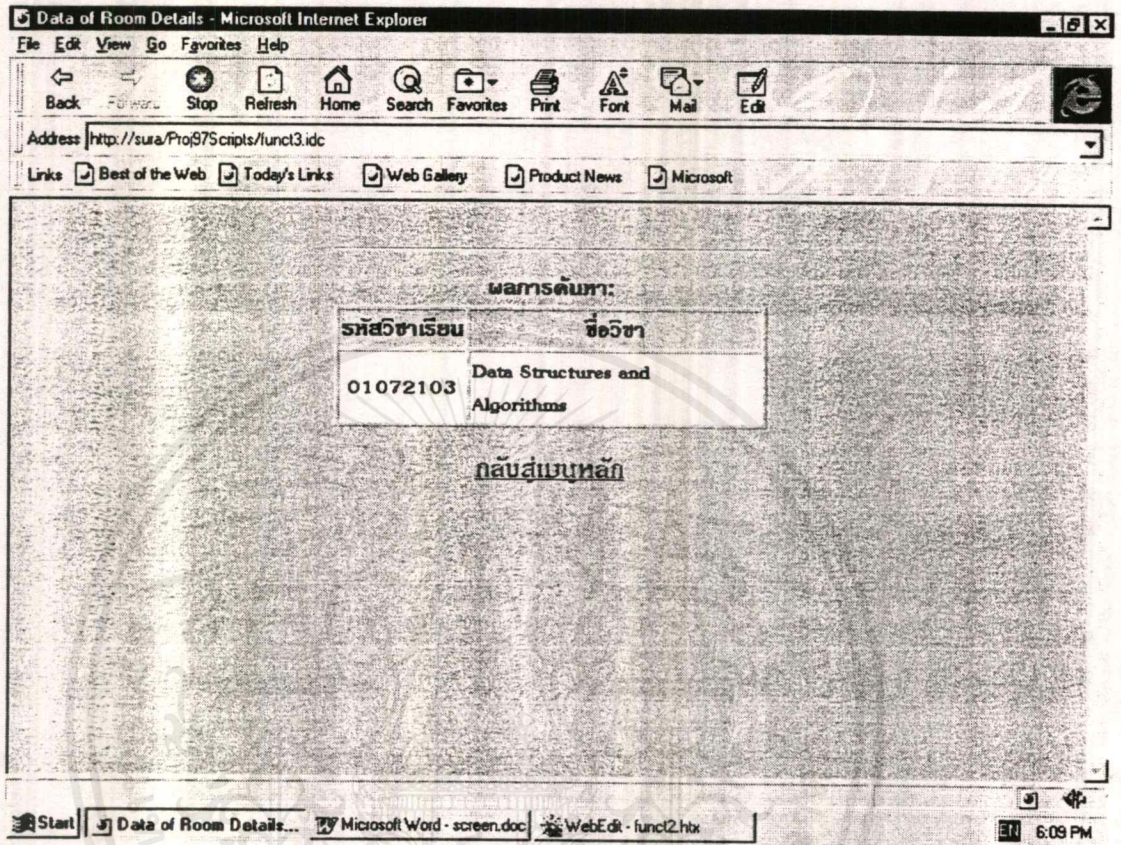
เมื่อผู้ใช้ทราบรหัสวิชาเรียนเพียงบางส่วน เช่น “0107” , “01073” หรือ “010731” เป็นต้น ผู้ใช้สามารถป้อนข้อมูลเท่าที่ทราบ แล้ว ระบบจะทำการเรียกค้นข้อมูลที่มีค่าสอดคล้องกับค่าที่ผู้ใช้ป้อน และจะปรากฏหน้าจอดังรูป



รูปที่ 4.17 แสดงหน้าจอผลลัพธ์การเรียกค้นข้อมูลวิชาเรียน ในกรณีที่ผู้ใช้ป้อนข้อมูลเพียงบางส่วน เช่น 0107210

C. ในกรณีที่ผู้ใช้ป้อนข้อมูลครบถ้วน

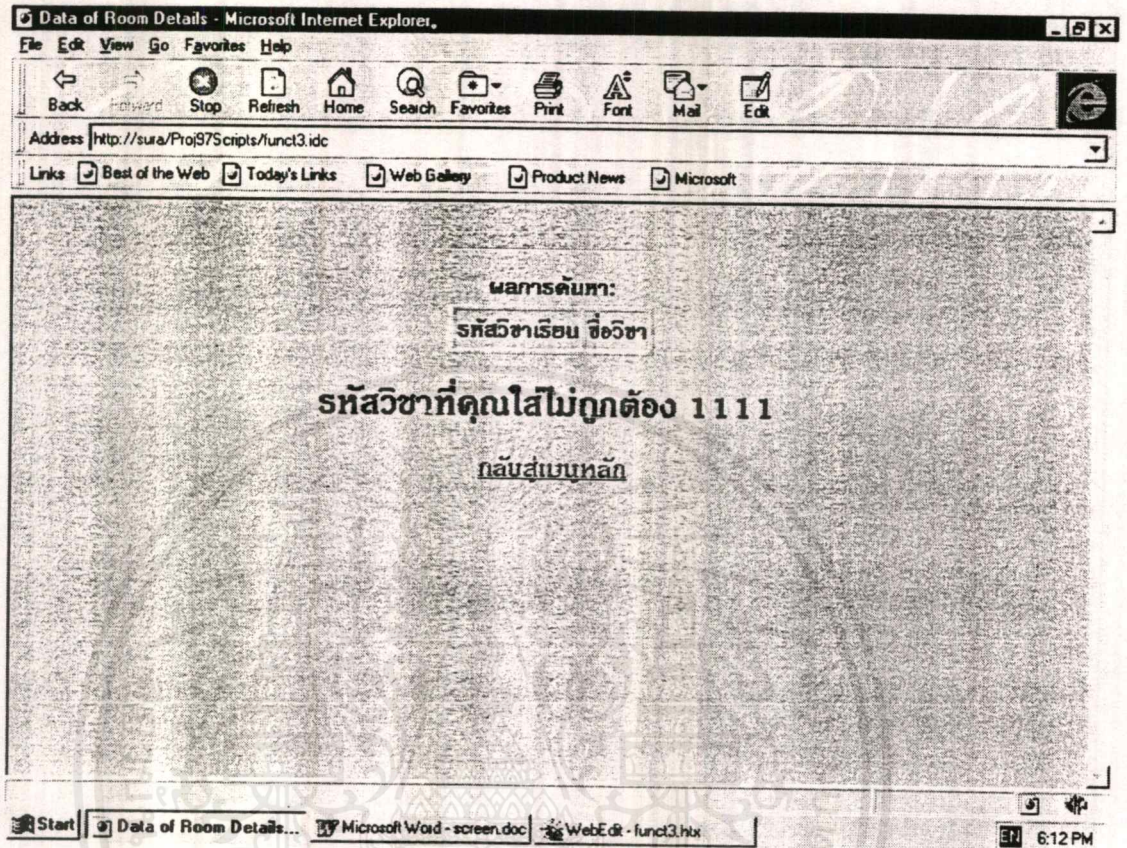
เมื่อผู้ใช้งานกรอกข้อมูลรหัสวิชาเรียนครบถ้วน ผู้ใช้สามารถเรียกค้นข้อมูลได้ทันที โดยผู้ใช้ทำการป้อนข้อมูลรหัสวิชาเรียน เช่น “01073110” เป็นต้น แล้วระบบจะทำการเรียกค้นข้อมูล และจะปรากฏหน้าจอผลลัพธ์ ดังรูป



รูปที่ 4.18 แสดงหน้าจอผลลัพธ์การเรียกค้นข้อมูลวิชาเรียน ในกรณีที่ผู้ใช้ป้อนข้อมูลครบถ้วน เช่น 01073110

D. ในกรณีที่ผู้ใช้ป้อนข้อมูลไม่ถูกต้อง

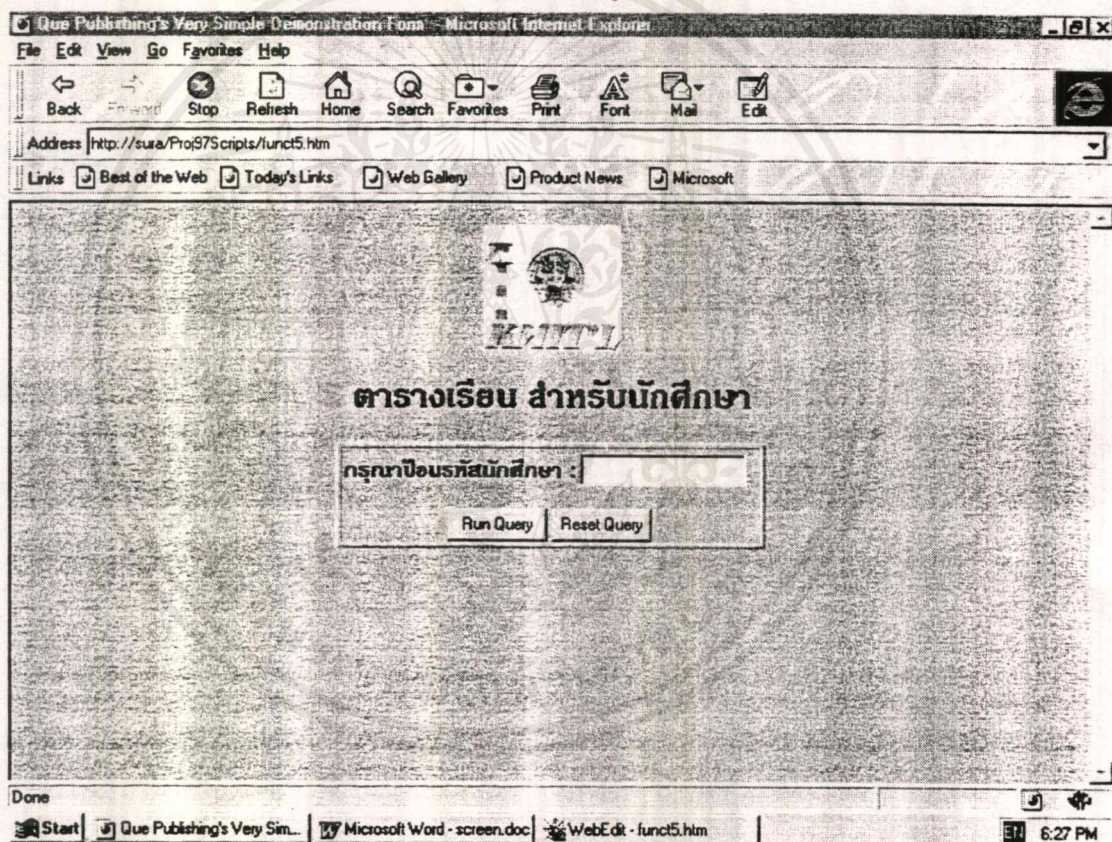
หากผู้ใช้ป้อนข้อมูลไม่ถูกต้อง เช่น 1111 ซึ่งไม่สอดคล้องกับข้อมูลรหัสวิชาเรียน ระบบจะทำการแจ้งข้อความเตือน ดังรูป



รูปที่ 4.19 แสดงหน้าจอผลลัพธ์การเรียกค้นข้อมูลวิชาเรียน ในกรณีที่ผู้ใช้ป้อนข้อมูลไม่ถูกต้อง เช่น 1111

4.2.2.4 การแสดงข้อมูลตารางเรียน ขั้นตอนการใช้งาน

- เลือกการทำงานที่ 3 จากหน้าจอเมนูหลัก(รูปที่ 5) จะปรากฏหน้าจอการทำงานดังรูปที่ 20
- ป้อนรหัสนักศึกษา เช่น “38013305” ซึ่งถ้าหากกรหัสนักศึกษา ที่ป้อนไม่ถูกต้อง จะปรากฏข้อความเตือน ดังรูปที่ 22 แล้วจะให้ป้อนรหัสนักศึกษาให้ถูกต้องอีกครั้ง
- คลิกปุ่ม Run Query เพื่อตกลง หรือ คลิกปุ่ม Reset Query เพื่อลบข้อมูลที่ป้อน
- ผลลัพธ์ที่ได้จะแสดงประกอบด้วย รหัสนักศึกษา , ชื่อนักศึกษา และตารางเวลา ซึ่งจะระบุวิชาที่เรียนในแต่ละช่วงเวลาของนักศึกษาผู้นั้น ดังรูปที่ 21 ในกรณีที่นักศึกษาผู้นั้น ไม่ได้ลงทะเบียนไว้ ก็จะปรากฏเป็นตารางเวลาว่าง
- เมื่อเสร็จสิ้นการทำงาน ผู้ใช้สามารถกลับสู่หน้าจอเมนูหลัก ได้โดยการคลิกที่ กลับสู่เมนูหลัก ซึ่งจะปรากฏหน้าจอดังรูปที่4.5



รูปที่ 4.20 แสดงหน้าจอการทำงานที่ 4 การแสดงข้อมูลตารางเรียน

func5.idc at sura - Microsoft Internet Explorer

File Edit View Go Favorites Help

Back Forward Stop Refresh Home Search Favorites Print Font Mail Edit

Address <http://sura/Proj97Scripts/func5.idc>

Links Best of the Web Today's Links Web Gallery Product News Microsoft

Student ID = "38013305"

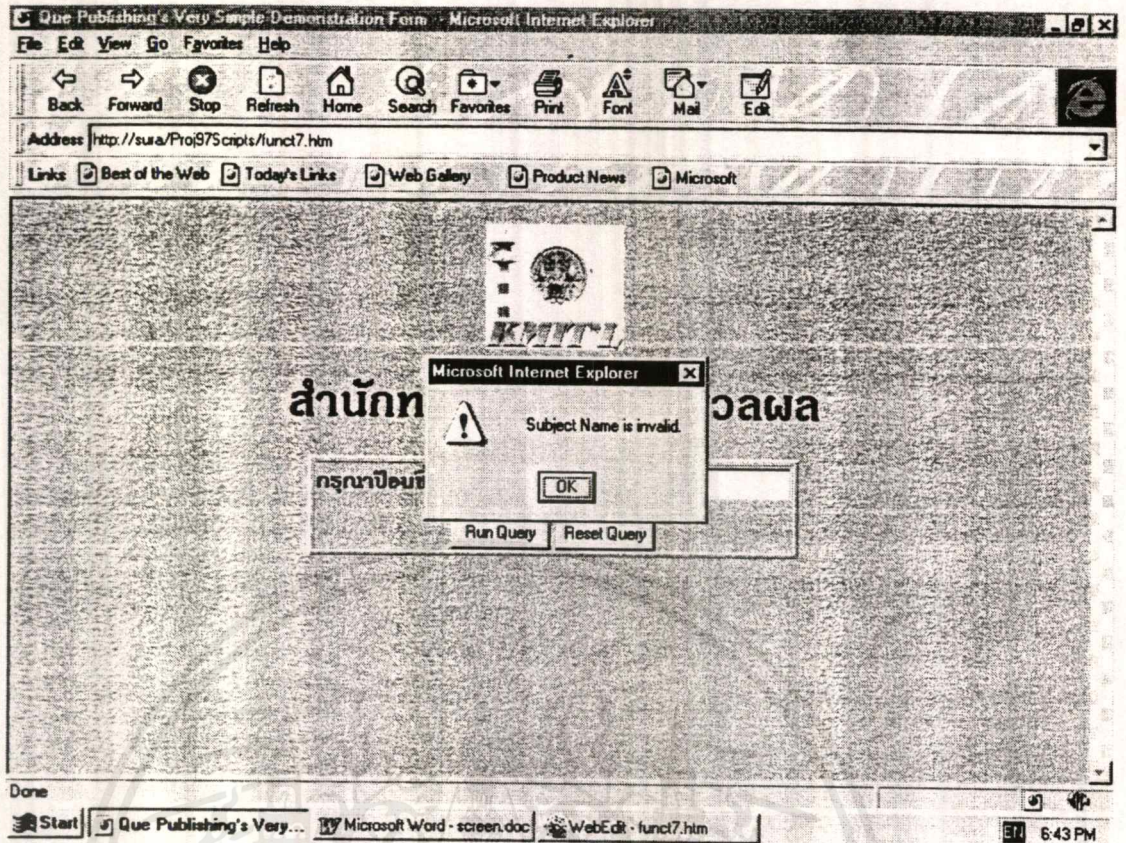
Student Name = "Uawin Surasit"

Time	P01	P02	P03	Noon	P07
Day	9:00-10:00	10:00-11:00	11:00-12:00		17:30-18:30
MON					
TUE	DataStruct	DataStruct	DataStruct		
WED	Compiler	Compiler	Compiler		
THU					
FRI					
SAT					

Start | [func5.idc at sura - M...](#) | [Microsoft Word - screen.doc](#) | [WebEdit - func3.htm](#) | 6:14 PM

รูปที่ 4.21 แสดงหน้าจอผลลัพธ์การเรียกค้นข้อมูลตารางเรียน ของนักศึกษารหัส 38013305

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 แสดงหน้าจอเตือนการป้อนข้อมูลนักศึกษาไม่ถูกต้อง

4.3 สรุปการทดลอง

จากการทดลองใช้งานพบว่าโปรแกรมให้ผลเป็นที่น่าพอใจ สามารถเรียกดูรายละเอียดของแต่ละชั้นเรียนได้อย่างสะดวกพอสมควร สามารถค้นหารายวิชาต่างๆ ได้ ส่วนในเรื่องของความเร็วนั้น ในการทดลองได้ใช้เครื่องลูกข่าย 3 เครื่องทำงานในฟังก์ชันเดียวกัน ปรากฏว่าความเร็วลดลงไปเพียงเล็กน้อยเท่านั้น แต่ทางผู้จัดทำคิดว่า ความเร็วในการทำงานเมื่อนำไปใช้งานจริงน่าจะขึ้นกับตัวแปรอื่นๆ ด้วย

บทที่ 5

สรุป วิจารณ์ และแนวทางในการพัฒนา

5.1 ความสามารถของโปรแกรม

- สามารถเรียกดูข้อมูลต่างๆ ในการจัดชั้นเรียน และห้องเรียน ผ่านเครือข่ายอินเทอร์เน็ตได้
- ช่วยอำนวยความสะดวกให้แก่ผู้ที่ต้องทราบรายละเอียดของรายวิชานั้น
- สามารถแสดงตารางเรียนของนักศึกษาแต่ละคนในรูปของตารางได้เมื่อใช้ประกอบกับระบบการลงทะเบียน
- สามารถใช้งานร่วมกับระบบเก่าที่ใช้ Visual Basic เป็น Front End ได้
- สามารถตรวจสอบการใช้ห้องเรียนไม่ให้มีการซ้ำซ้อน หรือคาบเกี่ยวกันได้อย่างมีประสิทธิภาพเพราะว่า ระบบจะแจ้งให้ทราบทันที เมื่อเกิดการซ้ำซ้อนขึ้น

5.2 ข้อจำกัดของโปรแกรม

- ระบบฐานข้อมูลที่ใช้เป็นฐานข้อมูลที่ได้ออกแบบไว้ก่อนแล้ว บางครั้งไม่สามารถรองรับการทำงานที่ซับซ้อนมากๆ ได้
- โปรแกรมใช้สถาปัตยกรรมแบบ IDC และ HTX ซึ่งค่อนข้างเก่าและล้าสมัยทำให้การทำงานบางฟังก์ชันที่ต้องการความสามารถในการแสดงผลสูงๆ เช่นการแสดงผลในรูปตาราง หรือโปรแกรมที่ต้องตรวจสอบเงื่อนไขเป็นจำนวนมาก ทำได้ยากมากหรือนางครั้งทำไม่ได้เลย
- User Interface อาจจะยังไม่ดีพอ ทั้งนี้เพราะทางผู้พัฒนายังมีประสบการณ์ในการออกแบบโอเอสน้อย

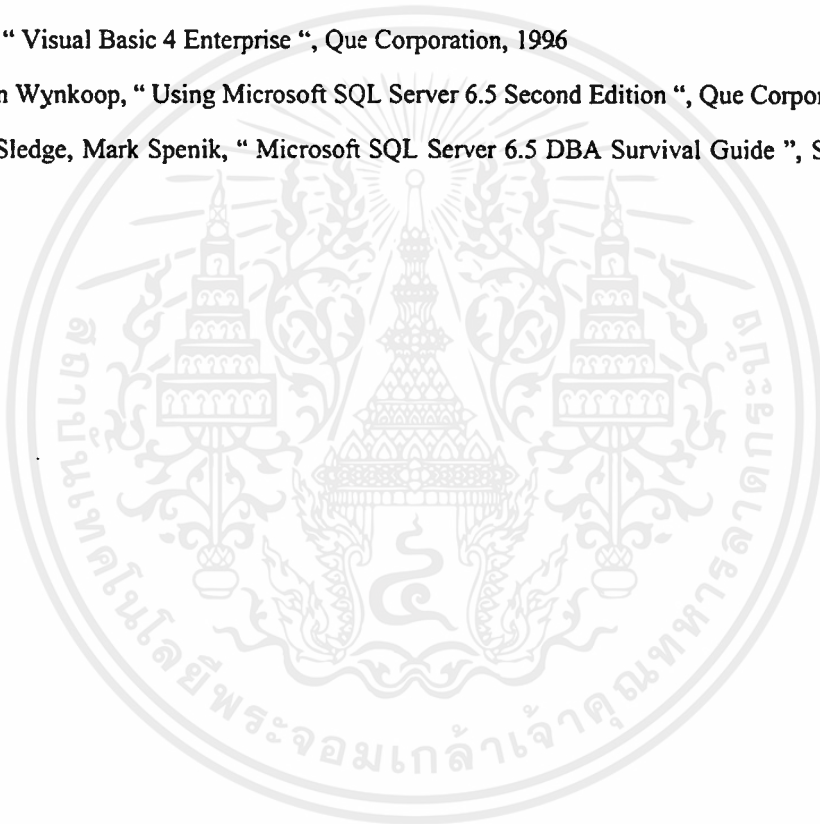
5.3 แนวทางการพัฒนา

- ควรที่จะต้องแก้ไขค่าเซตเพื่อให้สามารถรองรับการทำงานที่สลับซับซ้อนมากๆ ได้
- HTML มีข้อจำกัดในการแสดงผล ดังนั้นจึงควรจะนำภาษาจาวาเข้ามาใช้เพื่อให้ User Interface ใช้งานได้ง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. Microsoft Corporation, “ Product Document of Microsoft Internet Information Server 3.0 “, Microsoft Corporation, 1997
2. Stephen Feather, “ JavaScript By Example”, Que Corporation, 1996
3. จิตเกษม พัฒนาศิริ, “ สร้าง Homepage ด้วย HTML “, บริษัท ธนาเพรส แอนด์ กราฟฟิค จำกัด, พ.ย. 2539
4. ดร.ศุภมิตร จิตตะขุโสธร, “ Relational Database Design “ , สถาบันไทยสารสนเทศเทคโนโลยี (ITIT), 1994
5. Goren, “ Visual Basic 4 Enterprise “, Que Corporation, 1996
6. Stephen Wynkoop, “ Using Microsoft SQL Server 6.5 Second Edition “, Que Corporation, 1997
7. Orryn Sledge, Mark Spenik, “ Microsoft SQL Server 6.5 DBA Survival Guide ”, Sams Publishing, 1996



กิตติกรรมประกาศ

สุดท้ายนี้โครงการระบบสารสนเทศของสำนักทะเบียนและประมวลผล ผ่านระบบเครือข่าย อินเทอร์เน็ต จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดีถ้าปราศจากผู้ที่ให้การสนับสนุนดังต่อไปนี้

1. บุพการี ที่ให้กำเนิด ดูแลอบรมเลี้ยงดู คอยให้กำลังใจ ส่งเสียให้เล่าเรียนจนมีวันนี้
2. ญาติพี่น้อง ที่คอยให้กำลังใจ ใต้ถ่มทุกข์สุข และให้ความช่วยเหลืออยู่เสมอ
3. บรรดาคณาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้มาตั้งแต่เยาว์วัย
4. อ.วิบูลย์ พร้อมพานิชย์ ที่ให้คำปรึกษา รวมถึงความสะดวกด้านสถานที่รวมทั้งอุปการะในการดำเนินงาน
5. Mr. Paul Iu ให้คำปรึกษาปัญหาทางด้านเน็ตเวิร์ค
6. เพื่อนฝูงทุกคนที่คอยให้คำแนะนำ และความช่วยเหลือ ในระหว่างการดำเนินงาน
7. เหล่าผู้มีพระคุณทุกท่านที่มีได้กล่าวถึง

พระยา พุกสวัสดิ์

อัศวิน สุรสิทธิ์

29 พฤษภาคม 2541