



ระบบค้นหาสื่อผ่านอินเทอร์เน็ต

Multimedia Searching via Internet System



โดย

นายภูมิศักดิ์ สนามชัยสกุล

นายระพีพันธ์ เจริญกิจกำจร

วัน เดือน ปี.....	14 ค.ค. 2541
เลขทะเบียน.....	038948
เลขเรียกหนังสือ.....	1.20189 อ. 679 จ.

ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์

ภาควิชาวิศวกรรมคอมพิวเตอร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

ระบบค้นหาสื่อผ่านอินเทอร์เน็ต
Multimedia Searching via Internet System



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์
ภาควิชาวิศวกรรมคอมพิวเตอร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท สาขาการศึกษา 2540

ภาควิชา วิศวกรรมศาสตร์คอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบค้นหาสื่อผ่านอินเทอร์เน็ต

(Multimedia Searching via Internet System)

ผู้จัดทำ

1. นายภูมิศักดิ์ สนามชัยสกุล รหัสประจำตัว 37014323
2. นายระพีพันธ์ เจริญกิจกำจร รหัสประจำตัว 37014351



..... อาจารย์ที่ปรึกษา

(.....)

ระบบค้นหาสื่อผ่านอินเทอร์เน็ต

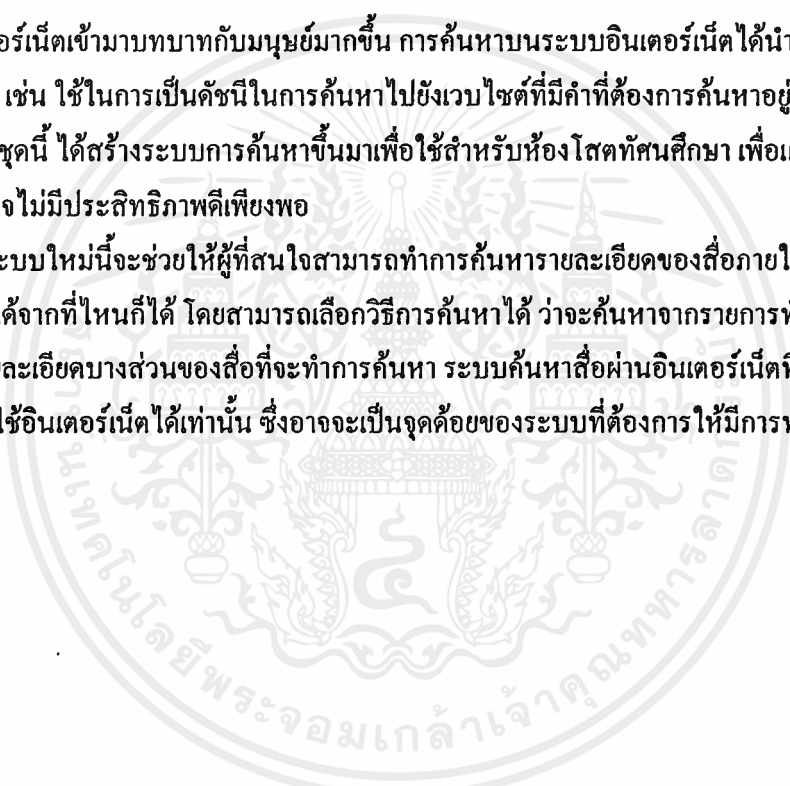
นายภูมิศักดิ์ สนามชัยสกุล
นายระพีพันธ์ เจริญกิจกำจร

อ.เอื้อน ปิ่นเงิน อาจารย์ที่ปรึกษา
ปีการศึกษา 2540

บทคัดย่อ

อินเทอร์เน็ตเข้ามาบทบาทกับมนุษย์มากขึ้น การค้นหาบนระบบอินเทอร์เน็ตได้นำไปประยุกต์ใช้กับงานต่าง ๆ มากมาย เช่น ใช้ในการเป็นดัชนีในการค้นหาไปยังเว็บไซต์ที่มีค่าที่ต้องการค้นหาอยู่ เป็นต้น ซึ่งในปฏิญานิพนธ์ชุดนี้ ได้สร้างระบบการค้นหาขึ้นมาเพื่อใช้สำหรับห้องโสตทัศนศึกษา เพื่อแทนที่ระบบการค้นหาแบบเดิมๆ ที่อาจไม่มีประสิทธิภาพดีเพียงพอ

โดยระบบใหม่นี้จะช่วยให้ผู้ที่สนใจสามารถทำการค้นหารายละเอียดของสื่อภายในห้องโสตทัศนศึกษาที่ต้องการทราบได้จากที่ไหนก็ได้ โดยสามารถเลือกวิธีการค้นหาได้ว่าจะค้นหาจากรายการทั้งหมด หรือค้นหาด้วยการกำหนดรายละเอียดบางส่วนของสื่อที่จะทำการค้นหา ระบบค้นหาสื่อผ่านอินเทอร์เน็ตที่ว่า พัฒนาขึ้นบนระบบคอมพิวเตอร์ที่ใช้อินเทอร์เน็ตได้เท่านั้น ซึ่งอาจจะเป็นจุดค้อยของระบบที่ต้องการให้มีการพัฒนาต่อไป



Multimedia Searching via Internet System

Poomisake Snamchaisakul

Rapeepan Jreonkijjumnorn

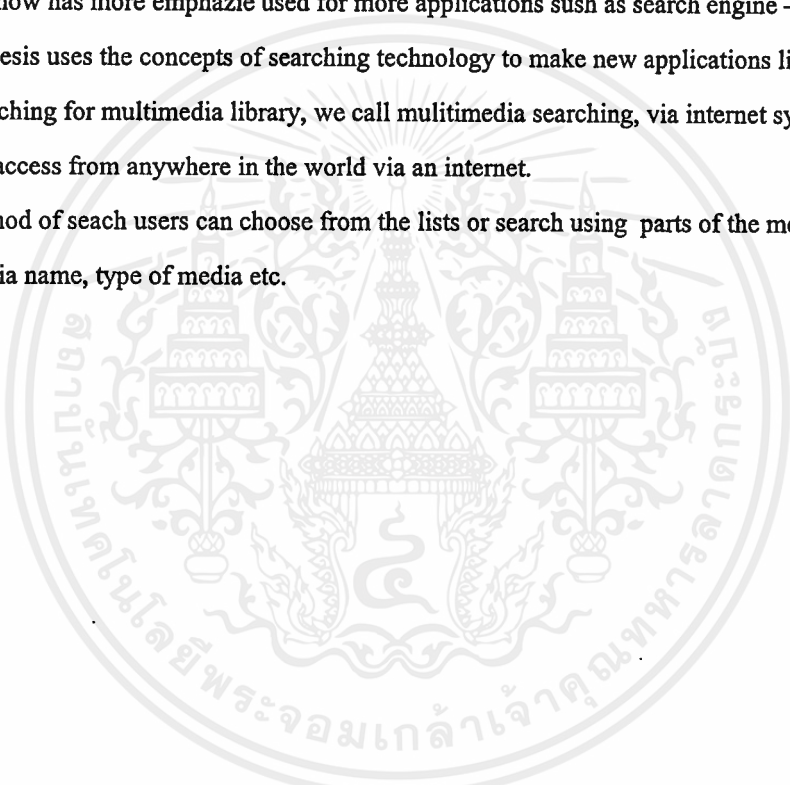
Ouen Pin-Ngern Ph.D. Advisor

1998

Abstract

Internet now has more emphazie used for more applications sush as search engine – search for keywords. This thesis uses the concepts of searching technology to make new applications like search engine. It is a multimedia searching for multimedia library, we call multimedia searching, via internet system. For the new system users can access from anywhere in the world via an internet.

The method of seach users can choose from the lists or search using parts of the media, like author name, media name, type of media etc.



สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
สารบัญ.....	III
สารบัญรูป.....	VIII
บทที่ 1.....	1
บทนำ.....	1
บทที่ 2.....	2
ทฤษฎีฐานข้อมูลเชิงสัมพันธ์.....	2
2.1 หลักการของโมเดลเชิงสัมพันธ์ (Relational model).....	2
2.1.1 ศัพท์เฉพาะของโมเดลเชิงสัมพันธ์	3
2.1.2 โมเดล ER (ER-model).....	4
การแปลงความสัมพันธ์ของข้อมูลจากโมเดล ER ไปสู่ในรูปของรีเลชัน	4
2.1.3 ทฤษฎีของระบบฐานข้อมูลเชิงสัมพันธ์	5
2.1.3.1 กฎของความคงสภาพ (Integrity rule).....	5
2.1.3.2 ฟังก์ชันการขึ้นต่อกัน (Function dependency).....	6
2.1.4 การนอร์มัลไลซ์ (Normalization).....	6
2.1.4.1 รูปแบบนอร์มัลระดับที่ 1.....	6
2.1.4.2 รูปแบบนอร์มัลระดับที่ 2.....	7
2.1.1.3 รูปแบบนอร์มัลระดับที่ 3.....	7
บทที่ 3.....	8
แอสคิวแอล เซิร์ฟเวอร์ และ โอดีบีซี	8
3.1 ไมโครซอฟต์ แอสคิวแอล เซิร์ฟเวอร์ (Microsoft SQL Server).....	8
3.1.1 ชุดเครื่องมือสำหรับแอสคิวแอลเซิร์ฟเวอร์.....	8
3.2 หลักการของโอดีบีซี (ODBC).....	9
3.2.1 ความหมายของคำว่าโอดีบีซี	9
3.2.2 ข้อดีของการติดต่อโดยใช้โอดีบีซี	9
3.2.3 องค์ประกอบของโอดีบีซี.....	10

บทที่ 4	11
อินเทอร์เน็ตอินฟอร์เมชันเซิร์ฟเวอร์, ไอเอสเอพีไอ และ ซีจีไอ	11
4.1 อินเทอร์เน็ต อินฟอร์เมชัน เซิร์ฟเวอร์ (Internet Information Server)	11
4.1.1 การใช้งานอินเทอร์เน็ต อินฟอร์เมชัน เซิร์ฟเวอร์	11
4.1.2 หลักการทำงานของอินเทอร์เน็ต อินฟอร์เมชัน เซิร์ฟเวอร์.....	11
4.1.2.1 การร้องขอด้วย URL	12
4.1.2.2 การตอบสนองของเว็บเซิร์ฟเวอร์.....	12
4.1.3 การนำ อินเทอร์เน็ต อินฟอร์เมชัน เซิร์ฟเวอร์ ไปใช้งาน	12
4.2 ISAPI (Internet Server Application Programming Interface)	14
4.2.1 การทำงานของ ISAPI	14
4.2.2 รูปแบบการใช้งานของ ISAPI.....	15
4.2.2.1 ISAPI Extension	15
4.2.2.2 ISAPI Filter	15
4.3 Common Gateway Interface (CGI)	16
บทที่ 5	17
ไมโครซอฟต์ฟันเดชันคลาส	17
5.1 CRecordset	17
5.1.1 CRecordset Class Member	17
5.1.1.1 Data Member.....	17
5.1.1.2 Construction.....	18
5.1.1.3 Recordset Attributes	19
5.1.1.4 Recordset Update Operations	21
5.1.1.5 Recordset Navigation Operations	21
5.1.1.6 Other Recordset Operations	22
5.1.1.7 Recordset Overridables.....	25
5.2 CDatabase	26
5.2.1 CDatabase Class Member	26
5.2.1.1 Data Member.....	26
5.2.1.2 Construction	26
5.2.1.3 Data Attributes	27
5.2.1.4 Database Operations.....	28

5.2.1.5 Database Overridables.....	29
5.3 CInternetConnection	29
5.3.1 CInternetConnection Class Members	29
5.3.1.1 Construction.....	29
5.3.1.2 Operations	29
5.3.1.3 Operators	30
5.4 CFtpConnection	30
5.4.1 CFtpConnection Class Members.....	31
5.4.1.1. Construction	31
5.4.1.2 Operations	31
5.5.CHttpServer.....	33
5.5.1 CHttpServer Class Members.....	33
5.5.1.1 Constructor	33
5.5.1.2 Overridables	33
5.5.1.3 Attributes	34
5.6 CHttpServerContext.....	35
5.6.1 CHttpServerContext Class Members.....	35
5.6.1.1. Data Members	35
5.6.1.2 Construction	35
5.6.1.3 Operations	35
5.6.1.4 Operators	36
บทที่ 6.....	37
ภาษาไฮเปอร์เท็กซ์ มาร์กอัพ	37
6.1 รู้จักกับ HTML.....	37
6.2 การเขียน HTML เบื้องต้น	37
6.2.1 ส่วนประกอบที่มีต้องมีในทุกเพจ	37
6.2.2 คำสั่งกำหนดหัวข้อ (Header Element)	38
6.2.3 คำสั่งจัดการลำดับ (List Element)	38
6.2.4 การจัดย่อหน้า (Paragraph)	38
6.2.5 Logical Styles และ Physical Styles	38
6.2.5.1 รูปแบบทางความหมาย.....	39

6.2.5.2 รูปแบบทางกายภาพ	39
6.2.6 ระบุสถานที่ (Address).....	39
6.2.7 การขีดเส้นใต้ (Horizontal Rules)	39
6.2.8 การใส่รูปภาพ	40
6.2.9 การเชื่อมโยง (Link)	40
6.2.10 mailto.....	40
6.3 คำสั่งที่ซับซ้อนๆ ของภาษา HTML	40
บทที่ 7.....	43
การศึกษาและความจำเป็นในการจัดทำระบบใหม่	
และ ความต้องการของระบบค้นหาสื่อผ่านอินเทอร์เน็ต	43
7.1 การวิเคราะห์ระบบการค้นหาสื่อเดิม	43
7.1.1 ขั้นตอนการค้นหาสื่อ	43
7.1.2 ปัญหาที่เกิดขึ้น	43
7.2 ความจำเป็นในการจัดทำระบบใหม่	44
7.3 ความต้องการของระบบใหม่.....	44
บทที่ 8.....	46
การออกแบบระบบใหม่.....	
8.1 การออกแบบระบบใหม่.....	46
8.2 การออกแบบและการประยุกต์ใช้งานระบบคอมพิวเตอร์.....	47
8.2.1.ระบบบริการฐานข้อมูล.....	47
8.2.2.ระบบบริการเว็รด์ไวด์เว็บและเอฟทีพี.....	49
8.2.3. โปรแกรมแก้ไขฐานข้อมูล	50
8.3 การออกแบบฐานข้อมูล.....	51
บทที่ 9.....	52
การนำไปใช้งาน.....	
9.1 แนะนำและติดตั้งระบบบริการฐานข้อมูล.....	52
9.2 แนะนำและติดตั้งระบบบริการเว็รด์ไวด์เว็บและเอฟทีพี.....	55
9.2.1 การติดตั้ง WWWQuery.dll	55

9.3 แนะนำและติดตั้งโปรแกรมแก้ไขฐานข้อมูล.....	56
9.3.1 การติดตั้ง โอดีบีซี และดีเอสเอ็น	56
9.3.2 การใช้งานโปรแกรมแก้ไขฐานข้อมูล.....	56
9.3.3 การเพิ่มเติมข้อมูล.....	58
9.3.4 การแก้ไขข้อมูลที่บันทึกไว้เรียบร้อยแล้ว.....	59
9.3.4.1 การลบข้อมูลที่ไม่ต้องการ	59
9.3.4.2 การค้นหาสื่อที่ต้องการ.....	59
บทที่ 10.....	61
สรุปวิจารณ์.....	61



สารบัญภาพ

	หน้า
รูปที่ 2-1 ตัวอย่างตารางรีเลชัน	3
รูปที่ 4-1 การติดต่อระหว่างเว็บเบราว์เซอร์กับเว็บเซิร์ฟเวอร์	12
รูปที่ 4-2 การทำเว็บเซิร์ฟเวอร์ด้วยอินเทอร์เน็ต อินฟอร์มเมชัน เซิร์ฟเวอร์	13
รูปที่ 4-3 การติดต่อระหว่างเครื่องไคลเอนท์และเซิร์ฟเวอร์	14
รูปที่ 4-4 หลักการทำงานของ ISAPI Extension	15
รูปที่ 4-5 หลักการทำงานของ ISAPI Filter	16
รูปที่ 4-6 หลักการทำงานของ CGI	16
รูปที่ 5-1 ออบเจกต์ CRecordset	17
รูปที่ 5-2 ออบเจกต์ Cdatabase	26
รูปที่ 5-3 ออบเจกต์ CInternetConnection	29
รูปที่ 5-4 ออบเจกต์ CFtpConnecion	30
รูปที่ 8-1 ความสัมพันธ์ของระบบทั้งหมด	47
รูปที่ 8-2 ระบบบริการฐานข้อมูล	48
รูปที่ 8-3 ระบบบริการเวิร์ดไวด์เว็บ	49
รูปที่ 8-4 โปรแกรมแก้ไขฐานข้อมูล	50
รูปที่ 9-1 ตัวอย่างการสร้างตารางด้วยไมโครซอฟต์เอสคิวแอลเซิร์ฟเวอร์	53
รูปที่ 9-2 การกำหนด Type_no เป็นไพรมารีคีย์	53
รูปที่ 9-3 กำหนดให้ class_no เป็นไพรมารีคีย์	54
รูปที่ 9-4 ODBC Administrator	56
รูปที่ 9-5 ตัวอย่างการสร้างคาค้าซอส	57
รูปที่ 9-6 โปรแกรมแก้ไขฐานข้อมูล	57
รูปที่ 9-7 การเพิ่มเติมข้อมูลตารางชนิดของสื่อ	59
รูปที่ 9-6 ตัวอย่างการค้นหาเพื่อการแก้ไข	60

บทที่ 1

บทนำ

ในปัจจุบันแหล่งที่มาของการศึกษาไม่ได้ปิดกั้นอยู่แค่ในห้องเรียนหรือแค่หนังสือเท่านั้น ในยุคโลกาภิวัตน์นี้ แหล่งความรู้ได้ถูกจัดเก็บในสื่อรูปแบบต่างๆ อีกมากมาย อาทิ เช่น วีดีโอเทป เทปคาสเซ็ท และ ซีดีรอม เป็นต้น ทั้งนี้ระบบที่รวบรวมสื่อต่าง ๆ เหล่านี้ไว้สำหรับให้บริการผู้ใช้ ส่วนมากจะเป็นห้องโสตทัศนศึกษา เช่น ตามโรงเรียน มหาวิทยาลัย หรือบางบริษัท ซึ่งส่วนมากภายในห้องโสตทัศนศึกษา ยังไม่มีระบบค้นหาสื่อ เพื่ออำนวยความสะดวกแก่ผู้ใช้มากนัก ที่พบเห็นส่วนมากจะให้ผู้ใช้งาน ค้นหาสื่อที่ต้องการด้วยตนเอง โดยอาจจะมียุทธศาสตร์สื่อทั้งหมดเอาไว้ให้ ซึ่งการค้นหาด้วยตนเองนั้นนอกจากจะใช้เวลานานแล้ว บางครั้งสื่อที่ได้อาจจะไม่ตรงกับความต้องการจริงๆ ของผู้ใช้เท่าใด เนื่องจากไม่มีทั้งรายละเอียด ตัวอย่างสื่อ เป็นต้น และด้วยเหตุผลต่าง ๆ เหล่านี้เอง เป็นสาเหตุให้พัฒนาระบบค้นหาสื่อขึ้นมาเพื่ออำนวยความสะดวกให้กับผู้ใช้บริการ ระบบที่ว่าควรจะสามารถในการค้นหาได้มีประสิทธิภาพมากกว่าการค้นหาด้วยตัวเอง

โดยสภาวะแวดล้อมที่จะนำระบบไปติดตั้งนั้น จะทำให้อยู่ในรูปของระบบบนอินเทอร์เน็ต ทั้งนี้เนื่องจากระบบอินเทอร์เน็ตใช้กันอย่างแพร่หลายและเป็นระบบเครือข่าย คอมพิวเตอร์ขนาดใหญ่มาก เพื่อให้ระบบสามารถนำไปใช้ได้อย่างกว้างขวางจึงเลือกขอบเขตของการพัฒนาแอปพลิเคชันบนอินเทอร์เน็ต ผ่านบริการเวิร์ดไวด์เว็บ (Word Wide Web : WWW) โดยให้มีการติดต่อกันระหว่างเว็บเบราว์เซอร์กับเว็บเซิร์ฟเวอร์ ซึ่งเป็นสถาปัตยกรรมของระบบคอมพิวเตอร์ประเภทหนึ่งที่นิยมใช้กันอย่างกว้างขวาง เนื่องจากมีข้อดีหลายอย่าง เช่น เป็นมาตรฐาน ใช้ง่าย ติดตั้งง่าย เป็นต้น

ซึ่งในปฏิญญาพันธบัตรจะอธิบายถึงทฤษฎีต่าง ๆ ที่ใช้ ประกอบไปด้วย ทฤษฎีข้อมูลเชิงสัมพันธ์ (Relational Database) ไมโครซอฟต์เอสคิวแอลเซิร์ฟเวอร์ (Microsoft SQL Server) โอเคบีซี (ODBC) อินเทอร์เน็ต อินฟอร์เมชันเซิร์ฟเวอร์ (Internet Information Server) ไอเอสเอพีไอ (ISAPI) ซีจีไอ (CGI) และทฤษฎีการวิเคราะห์ ออกแบบและพัฒนาระบบตามหลักของซอฟต์แวร์เอ็นจิเนียริง (Software Engineering) โดยรายละเอียดของทฤษฎีต่าง ๆ เหล่านี้จะกล่าวโดยละเอียดต่อไป ในส่วนของการทดลองระบบนั้น จะนำไปติดตั้งที่ห้องโสตทัศนศึกษา สำนักหอสมุดกลาง สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งนำระบบที่พัฒนานี้ไปงานแทนระบบเดิมที่ผู้ใช้งานค้นหาสื่อจากรายการสื่อเอง และยังสามารถค้นหาได้จากภายนอก โดยไม่ต้องเดินทางมาเองยังห้องโสตทัศนศึกษาด้วยตนเอง

บทที่ 2

ทฤษฎีข้อมูลเชิงสัมพันธ์

2.1 หลักการของโมเดลเชิงสัมพันธ์ (Relational model)

โมเดลเชิงสัมพันธ์เป็น โมเดลที่ใช้ในการอธิบายความสัมพันธ์ของข้อมูลที่ถูกเก็บด้วยระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System : RDBMS) ซึ่งเป็นผลงานของดร.คอดด์ (Codd) ที่ได้เสนอผลงานวิจัยให้ชาวโลกรู้จักในปี.ศ.2513 โดยมีบรรดานักวิชาการทางคอมพิวเตอร์ ก็ได้ให้ความสนใจและทุ่มเททำการวิจัยเกี่ยวกับโมเดลนี้มากมาย จนในปัจจุบันนี้ได้แพร่หลายไปมาก มีการนำไปใช้งานกับเครื่องระดับตั้งแต่เมนเฟรมลงไปจนถึงเครื่องระดับไมโครด้วย และก็เป็นที่ยอมรับกันแล้วว่า บรรดาผู้ใช้ระบบฐานข้อมูล (โดยเฉพาะผู้ที่ทำงานด้วยเครื่องระดับมินิและระดับไมโคร) จะมีความคุ้นเคยกับโมเดลเชิงสัมพันธ์นี้มากกว่าอีก 2 โมเดล คือ โมเดลเชิงแตกสาขา และ โมเดลเชิงโครงข่าย ที่มีมาก่อนหน้านี้

นอกเหนือจากความแพร่หลายของ โมเดลเชิงสัมพันธ์นี้แล้ว ข้อดีของโมเดลเชิงสัมพันธ์ที่มีมากกว่าอีก 2 โมเดล ดังนี้

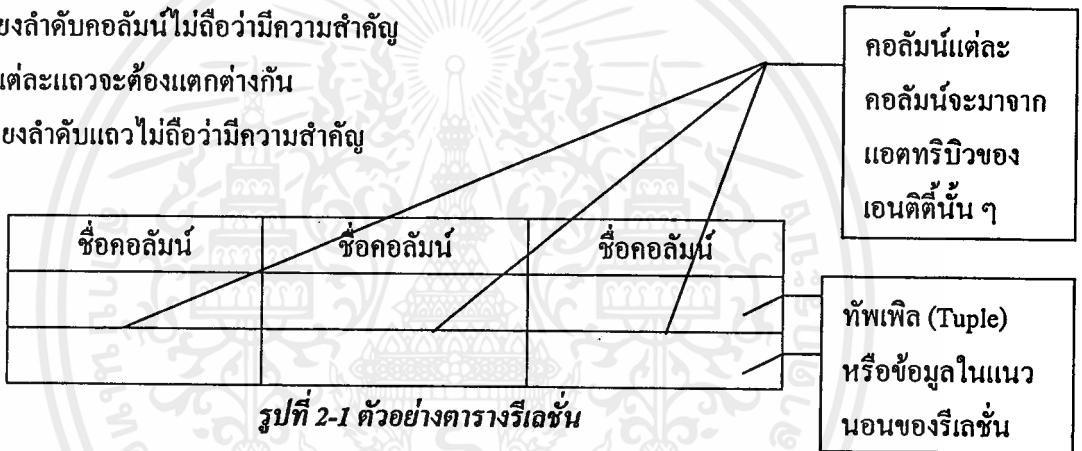
1. โมเดลเชิงสัมพันธ์เป็นโมเดลที่สามารถสร้างความเข้าใจง่ายกว่า เพราะภาพลักษณ์ของข้อมูล ที่เก็บโดยโมเดลเชิงสัมพันธ์จะมาจากมุมมองของผู้ใช้ ซึ่งจะมีความซับซ้อนน้อยกว่า ภาพลักษณ์ของข้อมูลที่เก็บโดยอีก 2 โมเดล
 2. ระบบส่วนใหญ่ที่ใช้โมเดลเชิงสัมพันธ์นี้มักจะมีเครื่องมือช่วยให้ผู้ใช้สามารถจัดการเก็บข้อมูลที่เก็บอยู่ได้ง่ายกว่าข้อมูลที่จัดเก็บด้วยโมเดลแบบอื่น
 3. โมเดลเชิงสัมพันธ์นี้มีเครื่องมือช่วยให้ผู้ใช้สามารถค้นพบปัญหาที่เกิดขึ้นในการออกแบบระบบฐานข้อมูลได้ โดยง่าย และยังง่ายในการแก้ไขการออกแบบที่ผิดพลาดนั้นด้วย
 4. โมเดลเชิงสัมพันธ์เป็น โมเดลที่มีความสอดคล้องกับหลักฐานของฐานข้อมูล ผู้ใช้ไม่ต้องพะวง กับรายละเอียดของการจัดข้อมูลของ โมเดลอื่น
 5. ภาษาที่ใช้ในการจัดการกับข้อมูลที่จัดเก็บด้วยระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (ภาษา SQL:Structure Query Language) เป็นภาษาแบบ Set oriented ซึ่งจะต่างกับภาษาที่ใช้ในการจัดการกับข้อมูลที่จัดเก็บด้วยระบบจัดการฐานข้อมูลของ โมเดลอื่นที่เป็นภาษาแบบ record-at-a-time
- แม้ว่าโมเดลเชิงสัมพันธ์จะมีข้อดีหลายประการดังได้กล่าวไปแล้ว แต่ในปัจจุบันก็ยังมี จุดอ่อนที่มีการอ้างอิงถึงเสมอ คือ ระบบจัดการฐานข้อมูลแบบโมเดลเชิงสัมพันธ์นี้มักจะมีประสิทธิภาพ ในการใช้งานสู้อีก 2 โมเดลไม่ได้ โดยเฉพาะในการประยุกต์ใช้งานขององค์กรขนาดใหญ่ จุดอ่อนนี้ก็ได้มีการแย้งกลับมาในแง่ที่ว่า โมเดลเชิงสัมพันธ์เป็น โมเดลที่มีอายุการพัฒนาน้อยกว่าอีก 2 โมเดล จึงเป็นไปได้ว่า การพัฒนาที่ผ่านมาของโมเดล

เชิงสัมพันธ์ก็ยังมีจำนวนระดับขั้นที่ได้พัฒนาไปแล้วน้อยกว่าอีก 2 โมเดล ดังนั้นหากต้องการเปรียบเทียบการทำงาน ระหว่างโมเดลเชิงสัมพันธ์กับโมเดลอื่น ก็ควรที่จะทำ การเปรียบเทียบที่ระดับจำนวนขั้นการพัฒนาที่เท่ากัน จึงจะสมเหตุสมผล

2.1.1 ศัพท์เฉพาะของโมเดลเชิงสัมพันธ์

ในหัวข้อนี้จะกล่าวถึงโมเดลเชิงสัมพันธ์ โดยกำหนดนิยามและกล่าวถึงคำศัพท์ต่าง ๆ ที่เกี่ยวข้องกับโมเดลนี้ จากการที่ข้อมูลที่เกี่ยวข้องกับโมเดลเชิงสัมพันธ์จะถูกเก็บไว้ในตารางที่จะถูกเรียกว่า "รีเลชัน" โดยที่รีเลชันทุกรีเลชันจะอยู่ในรูปของตาราง แต่ตารางบางตารางอาจไม่เป็นรีเลชันก็ได้ ดังนั้นตารางที่มีลักษณะเป็นรีเลชันจะต้องมีคุณลักษณะดังนี้

1. แต่ละช่องของตารางจะบรรจุข้อมูลได้เพียงค่าเดียว
2. ชื่อหัวข้อในแต่ละคอลัมน์มีความแตกต่างกัน อันเป็นชื่อของแอตทริบิวของเอนิตีนั้นๆ
3. ค่าข้อมูลที่อยู่ในแต่ละคอลัมน์ คือ ค่าของแอตทริบิวตามที่ระบุหัวข้อไว้ที่หัวของคอลัมน์นั้นๆ
4. การเรียงลำดับคอลัมน์ไม่ถือว่ามีความสำคัญ
5. ข้อมูลแต่ละแถวจะต้องแตกต่างกัน
6. การเรียงลำดับแถวไม่ถือว่ามีความสำคัญ



ตารางที่มีคุณลักษณะดังกล่าวจะเรียกว่า รีเลชัน

ดังนั้น เราจะได้นิยามของฐานข้อมูลเชิงสัมพันธ์ คือ ฐานข้อมูลที่เกิดจากการรวบรวมรีเลชันต่าง ๆ ที่มี ความสัมพันธ์ (Relationship) ระหว่างกัน

เราจะเรียกข้อมูลแต่ละแถวในแนวนอนของรีเลชันว่า ทัพเพิล (tuple) และเรียกข้อมูลแต่ละแถวในแนวตั้ง หรือแนวคอลัมน์ว่า แอตทริบิว โดยที่คำว่าคีย์ จะหมายถึงข้อมูล ที่เกิดจากแอตทริบิว 1 ตัว หรือหลายตัวก็ได้ ที่สามารถใช้เป็นตัวเจาะจงบอกเราได้ว่ากำลังอ้างอิงถึงข้อมูลทัพเพิลใด ส่วนคีย์ที่เป็นแอตทริบิวของรีเลชันอื่นที่ซ้ำกับแอตทริบิวที่เป็นคีย์หลักของรีเลชันหนึ่งจะเรียกว่า คีย์นอก (Foreign key) ในกรณีที่มีรีเลชันมีแอตทริบิวหรือกลุ่มแอตทริบิวที่มีคุณสมบัติเป็นคีย์หลัก ได้อยู่หลายแอตทริบิว เราจะเรียกแอตทริบิวหรือกลุ่มแอตทริบิวที่มีได้ถูกเลือกให้เป็นคีย์หลักว่า คีย์คู่แข่ง (Candidate key) หรือคีย์สำรอง (alternate key) และแอตทริบิวอื่น ๆ ที่เหลือที่มีได้เป็นคีย์หลักและไม่ได้เป็นส่วนใดส่วนหนึ่งของคีย์หลักก็จะถูกเรียกว่า นันคีย์ (Nonkey attribute)

คำว่า โดเมน จะหมายถึงกรอบของค่าต่าง ๆ ที่เป็นไปได้ เช่น โดเมนของแอตทริบิววันที่ ก็จะหมายถึงค่าของวันที่ที่เป็นไปได้ คือมีค่าเท่ากับ 1 ถึง 31 ในเดือนที่ลงท้ายด้วยคำว่า "คม" มีค่าเท่ากับ 1 ถึง 30 ในเดือนที่ลงท้ายด้วยคำว่า "ยน" และในเดือนกุมภาพันธ์อาจมีค่าเท่ากับ 1 ถึง 28 หรือ 29 วันก็ได้ แต่ในการเก็บข้อมูลลงในรีเลชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั้น บางกรณีที่เรามีการกำหนดโดเมนให้กับแอตทริบิวต์แล้ว แต่ข้อมูลที่จะถูกเก็บเข้าไปอาจถูกบรรจุเข้าไปในภายหลัง ลักษณะนี้จะทำให้เกิด คำว่าง ขึ้นชั่วคราวก่อนที่มี การบรรจุค่าข้อมูลที่อยู่ในโดเมนที่กำหนดไว้เข้าไป ดังนั้นคำว่า "คำว่าง" จึงหมายถึงค่าที่ยังมีทราบชัดว่าแอตทริบิวต์ นั้นจะมีค่าเป็นค่าใด หรือค่าของข้อมูลที่ไม่อยู่ในโดเมนที่กำหนด โดยมีข้อบังคับไว้ว่า แอตทริบิวต์ ที่ทำหน้าที่เป็นคีย์หลัก ของรีเลชันจะมีค่าข้อมูลเป็นคำว่างไม่ได้เสมอ เพราะจะทำให้การเข้าถึงข้อมูลในทัพเพิล นั้นกระทำไม่ได้

เมื่อมีการจัดเก็บข้อมูลในฐานข้อมูลใด ๆ แล้ว ข้อมูลจะถูกแยกออกเป็นกลุ่มของข้อมูลเป็นชุดที่ ประกอบด้วยแอตทริบิวต์ต่าง ๆ ที่มีความสัมพันธ์กัน เช่น การเก็บข้อมูลของบุคลากรในโรงเรียนก็อาจแยกเก็บ เป็นกลุ่มข้อมูลของนักเรียน กลุ่มข้อมูลของครูอาจารย์ และกลุ่มข้อมูลของนักการภารโรง เป็นต้น กลุ่มข้อมูลแต่ละกลุ่มนี้จะเรียกว่า เอนติตี้ ซึ่งแต่ละเอนติตี้จะประกอบไปด้วย แอตทริบิวต์ต่างๆ ที่มีความสัมพันธ์กัน เช่น เอนติตี้ของนักเรียนก็จะประกอบไปด้วยชื่อ นามสกุล ที่อยู่ ชั้นเรียน เป็นต้น

จากการแยกจัดเก็บข้อมูลออกเป็นเอนติตี้ แต่ละเอนติตี้ก็จะมี ความสัมพันธ์กัน ความสัมพันธ์ระหว่างเอนติตี้ สามารถแบ่งออกเป็น 3 ชนิด คือ

- ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (one to one)
- ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (one to many)
- ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (many to many)

2.1.2 โมเดล ER (ER-model)

โมเดล ER (ER-model : Entity Relationship model) เป็นแผนภาพที่ถูกรวบรวมมาเพื่อแสดง ความสัมพันธ์ ระหว่างเอนติตี้ต่าง ๆ ในรูปแบบที่เป็นรูปธรรมมากขึ้น เมื่อมีการใช้โมเดล ER แสดงความสัมพันธ์ระหว่างเอนติตี้แล้วก็ไม่จำเป็นต้องมีคำอธิบายความสัมพันธ์ใด ๆ อีก เพราะ โมเดล ER ประกอบไปด้วยสัญลักษณ์ต่าง ๆ ที่แสดงถึงคุณลักษณะของเอนติตี้และแอตทริบิวต์ได้ในตัวเองแล้ว

หลังจากเราสามารถหาความสัมพันธ์ระหว่างข้อมูลต่าง ๆ โดยแสดงความสัมพันธ์ของข้อมูลเหล่านั้นด้วยโมเดล ER แล้ว ขั้นตอนต่อไปนี้คือการเปลี่ยนความสัมพันธ์ ของข้อมูลที่อยู่บนโมเดล ER ให้อยู่ในรูปของรีเลชัน นั้นจะต้องมีขั้นตอนในการเปลี่ยนอยู่ 7 ขั้นตอน หลังจากนั้นก็ต้องทำการปรับปรุงรีเลชันที่ได้ให้มีความซ้ำซ้อนของความสัมพันธระหว่างข้อมูลหมดไป ซึ่งการปรับปรุงดังกล่าวก็จะมีทฤษฎีที่จะต้องอ้างอิงคือ กฎของความคงสภาพทฤษฎีการนอร์มัลไลซ์ ดังจะได้กล่าวต่อไป

การแปลงความสัมพันธ์ของข้อมูลจากโมเดล ER ไปสู่ในรูปของรีเลชัน

การแปลงความสัมพันธ์ของข้อมูลจากโมเดล ER ไปสู่รูปของรีเลชันมีขั้นตอนดังนี้

1. สำหรับแต่ละเอนติตี้ที่ไม่ใช่เอนติตี้แบบอ่อนของโมเดล ER เราจะสร้างเป็นรีเลชัน โดยมีทุกแอตทริบิวต์ที่เป็นแอตทริบิวต์ธรรมดา (Simple attribute) มาประกอบกัน แล้วทำการเลือกแอตทริบิวต์ใด แอตทริบิวต์หนึ่งหรือกลุ่มของแอตทริบิวต์มาทำหน้าที่เป็นคีย์หลักของรีเลชัน

2. สำหรับแต่ละเอนทิตีแบบอ่อน เราจะสร้างรีเลชันที่เกิดจากการรวมกันของแอตทริบิวธรรมดาของเอนทิตีนั้น โดยที่รีเลชันนี้จะมีคีย์หลักคือคีย์รวม (Combine key) ที่เกิดจากการรวมกันของคีย์หลักของเอนทิตีแบบอ่อน (Patial key) กับคีย์หลักของเอนทิตีที่มันต้องอ้างอิง (ในกรณีนี้คีย์นี้จะเรียกว่าเป็นคีย์นอก)

3. สำหรับความสัมพันธ์ระหว่างเอนทิตีแบบหนึ่งต่อหนึ่ง เราจะสร้างรีเลชันจากความสัมพันธ์ดังกล่าวได้ 2 ลักษณะ คือ เลือกคีย์หลักของเอนทิตีใดเอนทิตีหนึ่งมาเป็นคีย์หลักของรีเลชันนี้ แล้วให้คีย์หลักของอีกเอนทิตีหนึ่งมาเป็นคีย์นอกของรีเลชันนี้ โดยถ้าความสัมพันธ์นี้มีแอตทริบิวท์ก็ให้นำแอตทริบิวท์เหล่านั้นมารวมอยู่ในรีเลชันนี้ด้วย

4. สำหรับความสัมพันธ์ระหว่างเอนทิตีแบบหนึ่งต่อกลุ่ม เราจะสร้างรีเลชันจากความสัมพันธ์ดังกล่าวโดยนำเอาคีย์หลักของเอนทิตีฝั่งที่มีความสัมพันธ์แบบกลุ่มมาเป็นคีย์หลักของรีเลชันนี้ แล้วให้นำเอาคีย์หลักของเอนทิตีฝั่งที่มีความสัมพันธ์แบบหนึ่งมาเป็นคีย์นอกของรีเลชันนี้ โดยถ้าความสัมพันธ์นี้มีแอตทริบิวท์ก็ให้นำเอาแอตทริบิวท์เหล่านั้นมารวมอยู่ในรีเลชันนี้ด้วย

5. สำหรับความสัมพันธ์ระหว่างเอนทิตีแบบกลุ่มต่อกลุ่ม เราจะสร้างรีเลชันจากความสัมพันธ์นี้โดยนำเอาคีย์หลักของทั้งสองเอนทิตีมาประกอบกันเป็นคีย์หลักของรีเลชันนี้ โดยถ้าความสัมพันธ์นี้มีแอตทริบิวท์ก็ให้นำเอาแอตทริบิวท์เหล่านั้นมารวมอยู่ในรีเลชันนี้ด้วย

6. สำหรับเอนทิตีใดที่มีแอตทริบิวท์ที่มีค่าข้อมูลแบบหลายค่า (Multivalued attribute หรือ Repeating group) ก็ให้สร้างรีเลชันใหม่โดยมีคีย์หลักของเอนทิตีร่วมกับแอตทริบิวท์ดังกล่าวเป็นคีย์หลักของรีเลชันนี้

7. สำหรับความสัมพันธ์ระหว่างเอนทิตีที่เกิดจากเอนทิตีมากกว่า 2 เอนทิตี ให้สร้างรีเลชันของความสัมพันธ์นี้โดยนำคีย์หลักของทุกเอนทิตีมาประกอบกันเป็นคีย์หลักของเอนทิตีนี้ โดยถ้าความสัมพันธ์นี้มีแอตทริบิวท์ก็ให้นำแอตทริบิวท์เหล่านั้นมารวมกันอยู่ในรีเลชันนี้ด้วย

เมื่อทำการเปลี่ยนความสัมพันธ์ระหว่างข้อมูลที่อยู่ในรูปของโมเดล ER มาอยู่ในรูปของรีเลชันแล้ว เราจะได้รีเลชันที่อยู่ในรูปแบบนอร์มัลระดับที่ 1 ซึ่งจะต้องมีการทำนอร์มัลไลซ์ต่อไปเสียก่อนจึงจะสามารถนำรีเลชันเหล่านี้ไปใช้งานในระบบจัดการฐานข้อมูลเชิงสัมพันธ์ได้

2.1.3 ทฤษฎีของระบบฐานข้อมูลเชิงสัมพันธ์

2.1.3.1 กฎของความคงสภาพ (Integrity rule)

กฎของความคงสภาพของโมเดลเชิงสัมพันธ์ เป็นทฤษฎีที่ช่วยยืนยันความถูกต้องของความสัมพันธ์ระหว่างข้อมูลว่า รีเลชันใดที่เป็นไปตามกฎของความคงสภาพนี้แล้วจะมีความสัมพันธ์ระหว่างข้อมูลถูกต้องอยู่ตลอดเวลา ไม่ว่ารีเลชันนั้นจะมีการเปลี่ยนแปลงแก้ไขข้อมูลไปในรูปแบบใดก็ตาม

กฎของความคงสภาพมีความหมายอยู่ 2 ลักษณะ คือ กฎความคงสภาพของเอนทิตี และกฎความคงสภาพของการอ้างอิง ดังอธิบายได้ดังนี้

1. กฎความคงสภาพของเอนติตี้ กล่าวว่

"แอตทริบิวต์ทุกตัวที่เป็นส่วนของคีย์หลักจะไม่อนุญาตให้มีค่าว่าง"

หมายความว่า คีย์หลักของทุกรีเลชันจะไม่สามารถเก็บค่าข้อมูลที่เป็นค่าว่างได้ เหตุผลของข้อกำหนดนี้คือ เพื่อให้การเข้าถึงข้อมูลเป็นค่าว่างแล้ว ก็จะส่งผลให้การเข้าถึงข้อมูลในทัพเฟิลนั้นไม่สามารถกระทำได้อย่างแน่นอน

2. กฎความคงสภาพของการอ้างอิง กล่าวว่

"ถ้าเรามีรีเลชัน R2 ซึ่งมี FK เป็นคีย์นอกที่อ้างอิงถึงคีย์หลัก PK ใน รีเลชัน R1 สำหรับทุกค่าของ FK ใน R2 จะต้อง

ก. มีค่าเท่ากับค่า PK ในทัพเฟิลใดทัพเฟิลหนึ่งในรีเลชัน R1

หรือ ข. มีค่าของแอตทริบิวต์ทุกตัวใน FK เป็นค่าว่าง

หมายความว่า แอตทริบิวต์ใด ๆ ที่เป็นคีย์หลักของรีเลชันหนึ่ง เมื่อมีการนำแอตทริบิวต์นั้นไปเป็น คีย์นอกของอีกรีเลชันหนึ่ง การเป็นคีย์นอกของของแอตทริบิวต์นั้นจะต้องมีโดเมนเป็นโดเมนเดียวกัน กับแอตทริบิวต์ที่เป็นคีย์หลักอย่างแน่นอน แต่อาจมีบางค่าข้อมูลของแอตทริบิวต์ที่เป็นคีย์หลักเป็นข้อมูล ไม่อยู่ในโดเมนของแอตทริบิวต์เป็นคีย์นอกก็ได้ นั่นคือ โดเมนของคีย์นอกจะต้องเล็กกว่าหรือเท่ากับโดเมนของคีย์หลักเสมอ

2.1.3.2 ฟังก์ชันการขึ้นต่อกัน (Function dependency)

ฟังก์ชันการขึ้นต่อกันเป็นข้อกำหนดที่ช่วยให้เราเห็นถึงความสัมพันธ์ของแอตทริบิวต์ต่าง ๆ ที่อยู่ในรีเลชัน ทั้งเพราะแอตทริบิวต์ต่าง ๆ ที่อยู่ในเอนติตี้เดียวกันก็เป็นไปได้ที่แอตทริบิวต์เหล่านั้น จะมีความสัมพันธ์กันเองโดยที่ความสัมพันธ์นี้อาจเกี่ยวข้องหรือไม่เกี่ยวข้องกับความสัมพันธ์ที่มีต่อคีย์หลักนั้นก็เป็นที่ได้ ซึ่งการที่แอตทริบิวต์เหล่านั้นมีความสัมพันธ์กันเองจะเป็นสิ่งที่เราต้องพิจารณาแยกออกเป็นรีเลชันย่อย ๆ เพราะแอตทริบิวต์ของแต่ละรีเลชันก็ควรจะมีความสัมพันธ์กับคีย์หลักของรีเลชันของตนเองเท่านั้น

2.1.4 การนอร์มัลไลซ์ (Normalization)

การนอร์มัลไลซ์เป็นการออกแบบฐานข้อมูลแบบที่เป็นมาตรฐานที่สุด ออกแบบโดย ดร.คอด โดยมีจุดประสงค์ของการออกแบบก็เพื่อลดความซ้ำซ้อนของความสัมพันธ์ของข้อมูล ให้เหลือน้อยที่สุด ซึ่งตามมาตรฐานปกติมีอยู่ 3 ระดับคือ

1NF (First Normal Form)

2NF (Second Normal Form)

3NF (Third Normal Form)

โดยที่รีเลชันใดที่ยังไม่สอดคล้องตามรูปแบบนอร์มัล ทั้งสามจะต้องมีการแยกกรีเลชันนั้น ๆ ออกเป็นรีเลชันย่อยๆ ต่อไปอีก

ต่อมาได้มีการออกแบบเพิ่มเติมขึ้นอีก 2 ระดับ คือ

4NF (Forth Normal Form)

5NF (Fifth Normal Form)

หากรีเลชันใดมีมาตรฐานถึงรูปแบบนอร์มัลระดับที่ 5 แล้วก็จะมั่นใจได้ว่า รีเลชันนั้นจะไม่มี ความซ้ำซ้อนของความสัมพันธ์ของข้อมูลอย่างแน่นอน

2.1.4.1 รูปแบบนอร์มัลระดับที่ 1

การปรับรีเลชันให้อยู่ในรูปแบบนอร์มัลระดับที่ 1 คือ การปรับจากรีเลชันที่ไม่นอร์มัล เช่น รีเลชันที่มีข้อมูลขอแอตทริบิวบางช่องมีมากกว่า 1 ค่า

นิยาม รีเลชันอยู่ในรูปแบบนอร์มัลระดับที่ 1 ก็ต่อเมื่อ โดเมนของแต่ละแอตทริบิวประกอบด้วย ข้อมูลที่เป็นหน่วยย่อยที่สุด สิ่งที่ได้จากการที่รีเลชันอยู่ในรูปแบบนอร์มัลระดับที่ 1 ก็คือ รีเลชันยังคงมีความซ้ำซ้อนของความสัมพันธ์ ระหว่างข้อมูลอยู่มากมาย เพราะนิยามของรูปแบบนอร์มัลระดับที่ 1 นี้กำหนดเพียงเฉพาะว่า แต่ละแอตทริบิวของรีเลชันจะมีโดเมนที่มีสมาชิกเป็นหน่วยที่เล็กที่สุดเท่านั้น มิได้เป็นการลดความซ้ำซ้อนของความสัมพันธ์ระหว่างข้อมูลแต่ประการใด

2.1.4.2 รูปแบบนอร์มัลระดับที่ 2

นิยาม รีเลชันจะอยู่ในรูปแบบนอร์มัลระดับที่ 2 ก็ต่อเมื่อรีเลชันนั้นอยู่ในรูปแบบนอร์มัลระดับที่ 1 แล้ว และทุกแอตทริบิวที่ไม่เป็นส่วนใดส่วนหนึ่งของคีย์หลักจะต้องขึ้นกับคีย์หลักของรีเลชันนั้นอย่างเต็มที่

สิ่งที่ได้จากการที่รีเลชันอยู่ในรูปแบบนอร์มัลระดับที่ 2 คือข้อมูลของบางแอตทริบิวที่ไม่ใช่คีย์หลักอาจมีความสัมพันธ์กันเองถือว่าเป็นความซ้ำซ้อนประการหนึ่งของรีเลชันนั้นๆ ที่จะต้องทำการลดด้วยรูปแบบนอร์มัลในระดับต่อไป

2.1.1.3 รูปแบบนอร์มัลระดับที่ 3

นิยาม รีเลชันจะอยู่ในรูปแบบนอร์มัลระดับที่ 3 ก็ต่อเมื่อรีเลชันนั้นอยู่ในรูปแบบนอร์มัลระดับที่ 2 แล้ว และทุกแอตทริบิวที่ไม่เป็นส่วนใดส่วนหนึ่งของคีย์หลักจะต้องไม่เป็นฟังก์ชันที่ขึ้นต่อกันเอง

โดยปกติแล้ว สิ่งที่ได้จากรีเลชันอยู่ในรูปแบบนอร์มัลระดับที่ 3 คือ รีเลชันจะไม่มี ความซ้ำซ้อนอีกต่อไป โดยที่จะสอดคล้องกับรูปแบบนอร์มัลระดับที่ 4 และ 5 ด้วย แต่ก็มีบางรีเลชันที่ต้องทำการนอร์มัลในระดับที่ 4 และ 5 ต่อเอาเอง

บทที่ 3

เอสคิวแอล เซิร์ฟเวอร์ และ โอดีบีซี

3.1 ไมโครซอฟต์ เอสคิวแอล เซิร์ฟเวอร์ (Microsoft SQL Server)

ไมโครซอฟต์ เอสคิวแอล เซิร์ฟเวอร์ เป็นโปรแกรมสำหรับจัดการระบบฐานข้อมูลที่ใช้กับเครื่องคอมพิวเตอร์ส่วนบุคคล และทำงานอยู่ภายใต้ระบบปฏิบัติการวินโดวส์เอ็นที สำหรับตัวเอสคิวแอลเซิร์ฟเวอร์นี้ได้รับการพัฒนามาจากโปรแกรมจัดการระบบฐานข้อมูลไซเบส เอสคิวแอลเซิร์ฟเวอร์ (Sybase SQL Server) โดยบริษัทไมโครซอฟต์ คอปอเรชั่น จำกัด ในส่วนของการจัดการระบบฐานข้อมูลทางกายภาพ โดยมีวิธีการทำงานที่ละขั้นตอนเป็นงาน ๆ ไป โดยจะควบคุมการนำข้อมูลเข้า และแสดงผลจากการประมวลผลในวงจรต่าง ๆ พร้อม ๆ กัน ไปด้วยกับการปฏิบัติการด้านอื่น ๆ กล่าวโดยสรุปแล้วก็คือ ระบบฐานข้อมูลของตัวเอง โดยไม่ต้องพึ่งพาระบบปฏิบัติการของเครื่องคอมพิวเตอร์ทั้งหมด อีกทั้งยังสามารถรับสัญญาณการติดต่อในระบบเครือข่าย รวมถึงการเชื่อมโยงในรูปแบบต่าง ๆ ได้แก่ ไอพีเอกซ์/เอสพีเอกซ์ (IPX/SPX) เนมด์ไปปี้ (Named Pipes) เนตบีดียูไอ (NetBEUI) ซอคเกต (Sockets) และ ทีซีพี/ไอพี (TCP/IP) ระบบการติดต่อกับผู้ใช้ของเอสคิวแอลเซิร์ฟเวอร์จะเป็นไปตามรูปแบบของวินโดวส์เอ็นที เพอร์ฟอร์แมนซ์ มอนิเตอร์ (Windows NT Performance Monitor) ซึ่งเป็นระบบตรวจสอบรูปแบบการทำงานต่าง ๆ ที่โปรแกรมกำลังทำอยู่ โดยแสดงเป็นรูปสัญลักษณ์ การสั่งงานไม่ว่าจะเป็นด้านงานฐานข้อมูล ระบบเครือข่าย ระบบปฏิบัติการ หรือแม้กระทั่งการทำงานในส่วนของฮาร์ดแวร์ เช่นการใช้ประโยชน์จากซีพียู การนำเข้า และการส่งออกของข้อมูลต่าง ๆ (I/O Activity) และความสามารถในการจับหน่วยความจำในแคช (Cache hits and misses) เป็นต้น ทั้งหมดนี้กระทำผ่านสัญลักษณ์รูปที่ระบบมีไว้ให้แล้วทั้งสิ้น จึงลดภาระการจดจำค่าต่าง ๆ ในการจัดแต่งเครื่องไม้เครื่องมือในฮาร์ดแวร์หรือในระบบเครือข่ายไปได้มากทีเดียว อีกทั้งยังทำให้สามารถกำหนดเงื่อนไขการใช้คำสั่งทริกเกอร์ (Trigger) กับระบบปฏิบัติการได้อีกด้วย

เอสคิวแอลเซิร์ฟเวอร์ ได้รับการปรับเปลี่ยนให้ระบบการจัดการล็อก (Lock Manager) ให้ทำการปลดกฏข้อบังคับ (Constraints) ณ จุดที่มีการปรับเปลี่ยนข้อมูล (Update-in-place) และตัวออปติไมเซอร์ (Optimizer) สามารถเข้าถึงข้อมูลแบบไม่ผ่านการทำดัชนีจัดกลุ่ม (Nonclustered index) ด้วยคำสั่งออร์เดอร์บาย (Order by Clause) และมีจุดตรวจสอบระบบ (Checkpoint) ในลักษณะที่ไม่สัมพันธ์กัน ซึ่งจะมีข้อดีคือ ทำให้งานต่าง ๆ สามารถดำเนินต่อไปได้อย่างต่อเนื่อง ในขณะที่อีกจุดหนึ่งกำลังถูกตรวจรับความถูกต้องอยู่

3.1.1 ชุดเครื่องมือสำหรับเอสคิวแอลเซิร์ฟเวอร์

ชุดเครื่องมือสำหรับเอสคิวแอลเซิร์ฟเวอร์จะประกอบไปด้วย

1. เอสคิวแอลออบเจกต์แมนเนเจอร์ (SQL Object Manager) ซึ่งจะเป็นตัวจัดการสร้างสโตร์โปรซีเคอร์ ทริกเกอร์ ตาราง คณิต การดูแลข้อมูล (View) และงานอื่น ๆ ในระบบฐานข้อมูล โดยจะมีบัลค์คอปี้โปรแกรม (Bulk-copy program) เพื่อรายงานความผิดพลาดต่าง ๆ ในระหว่างปฏิบัติงาน ตัวเอสคิวแอลออบเจกต์

แมนเนเจอร์นี้ สามารถสร้างภาษานิยามข้อมูลเอสคิวแอล ซึ่งเป็นภาษาที่สร้างจากตัวโปรแกรมฐานข้อมูลที่กำลังใช้ อยู่ โดยเรียกผ่านทางตัวเซิร์ฟเวอร์ตัวอื่น ๆ หรืออาจจะสร้างเอกสาร โครงสร้างฐานข้อมูลที่กำลังเรียกใช้ได้โดยตรง

2. ชุดเครื่องมือดูแลเอสคิวแอล (SQL Administrator Tool) ทำงานเกี่ยวกับเครื่องต่อพ่วงต่าง ๆ การจัดการ ฐานข้อมูล และการสร้างฐานข้อมูล ชุดต่อเชื่อมต่าง ๆ กำหนดตัวผู้ใช้ และสร้างระบบรักษาความปลอดภัยให้กับ ข้อมูล

3. โปรแกรมไอเอสคิวแอลสำหรับวินโดวส์ (ICSQL/Windows หรือ ISQL/W) มีลักษณะที่คล้ายคลึงกับ ระบบติดต่อกับผู้ใช้ในวินโดวส์ ที่เข้าใจง่าย ซึ่งจะช่วยให้สามารถสร้างผังเรียกหาข้อมูล (Show plan) ที่แสดงเป็น สัญลักษณ์กลุ่มรูป และคำสั่งจัดการเรียกใช้ข้อมูลต่าง ๆ

3.2 หลักการของโอดีบีซี (ODBC)

3.2.1 ความหมายของคำว่าโอดีบีซี

โอเพน เดต้าเบส คอนเน็กติวิตี (Open Database Connectivity) คือวิธีการติดต่อและเข้าถึงจากแอปพลิเคชัน ทุกระบบบริหารฐานข้อมูล โดยใช้ภาษา เอสคิวแอล เป็นมาตรฐานการเข้าถึงข้อมูล ความสามารถในการเชื่อมต่อ แบบนี้ทำให้แอปพลิเคชันสามารถเข้าถึงฐานข้อมูลได้หลายรูปแบบ ซึ่งทำให้ผู้พัฒนาโปรแกรมสามารถพัฒนา โปรแกรมไปได้โดยไม่ต้องทำการระบุชนิดของ ระบบบริหารฐานข้อมูล

แต่เดิมนั้นการพัฒนาโปรแกรมประยุกต์ใช้งานเกี่ยวกับฐานข้อมูล การเข้าใช้ฐานข้อมูล โปรแกรมเหล่านี้จะ ทำผ่านการเรียกใช้เอ็มเบดเดด เอสคิวแอล (Embedded SQL) ซึ่งในขณะนั้นวิถีทางแบบนี้ก็ดูจะไปได้ดีทีเดียว เพราะว่า ตัวโปรแกรมสามารถทำการเปลี่ยนรูปแบบของระบบไม่ว่าจะเป็นทางด้านฮาร์ดแวร์ หรือ ซอฟต์แวร์ได้หลายรูปแบบ รวมทั้งระบบปฏิบัติการด้วย (โดยการคอมไพล์ใหม่ทุกครั้งที่มีการย้ายระบบ)

อย่างไรก็ตามในการพัฒนาโปรแกรมในระบบที่มีความแตกต่างกัน เช่น การเรียกใช้ข้อมูลของออราเคิล จาก ไมโครซอฟท์ เอ็กเซล (Microsoft Excel) วิธีการเข้าถึงข้อมูลแบบเดิมนั้นจะต้องทำการพรีคอมไพล์โค้ดของ เอ็กเซล และออราเคิลโดยใช้ โอบีเอ็ม ปรีคอมไพเลอร์ (IBM Precompiler) และออราเคิล ปรีคอมไพเลอร์ (Oracle Precompiler) ตามลำดับ ซึ่งจะเห็นว่าเป็นการยุ่งยากมากทีเดียว

วิธีการต่อเชื่อมแบบ โอดีบีซี จะให้ความสะดวกในการติดต่อข้อมูลมากกว่าวิธีการดั้งเดิม โดยการกำหนด มาตรฐานการต่อเชื่อมของข้อมูล (Data protocol, DBMS capability) และแนวทางนี้ได้ทำให้เกิดความคิดที่จะสร้าง ไคร์เวอร์การติดต่อของงานข้อมูลขึ้นมา

3.2.2 ข้อดีของการติดต่อโดยใช้โอดีบีซี

1. ฟังก์ชันของโอดีบีซี อนุญาตให้แอปพลิเคชันติดต่อกับดีบีเอ็มเอส ได้โดยสะดวก (การทำคำสั่งเอสคิว แอลและการรับผลลัพธ์)

2. ใช้ภาษาเอสคิวแอลตามมาตรฐาน เอสคิวแอล ซีเออี (SQL CAE) เอ็กซ์/โอเพน (X/Open) และ เอสเอจี (SAG : SQL Access Group)

3. มีการกำหนด การส่งกลับรหัสความผิดพลาด (Error Code) เป็นมาตรฐานเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เป็นวิธีการมาตรฐานในการติดต่อกับคีย์เอ็มเอส
5. มีการกำหนดชนิดของข้อมูล (Data Type) เป็นมาตรฐาน
6. มีชุดคำสั่งเอสคิวแอล สามารถกำหนดได้แม้แต่ในขณะทำงาน
7. สามารถเขียนโปรแกรมชุดเดียวแต่สามารถเข้าใช้คีย์เอ็มเอสได้หลายตัว
8. ตัวโปรแกรมไม่ต้องรับผิดชอบในการดูแลการติดต่อกับข้อมูลกับคีย์เอ็มเอส
9. ค่าข้อมูลสามารถถูกส่งหรือรับได้ในรูปแบบที่สะดวกขึ้น

3.2.3 องค์ประกอบของโอดีบีซี

สถาปัตยกรรมของโอดีบีซีประกอบด้วย 4 ส่วนสำคัญ

1. แอปพลิเคชัน ทำหน้าที่ประมวลผลและเรียกใช้ฟังก์ชันของ โอดีบีซี ตามคำสั่งภาษาเอสคิวแอล พร้อมทั้งทำการรับผลลัพธ์ด้วย
2. ตัวจัดการไดรเวอร์ (Driver Manager) ทำหน้าที่ เรียกโปรแกรมควบคุม เชื่อมต่อกับแหล่งข้อมูล
3. ไดรเวอร์ (Driver) ทำหน้าที่ประมวลผลการเรียกใช้ฟังก์ชันของ โอดีบีซี ส่งคำสั่ง เอสคิวแอล ไปสู่แหล่งข้อมูลที่ถูกต้องและทำการส่งผลลัพธ์กลับให้แอปพลิเคชันและในบางครั้งไดรเวอร์ จะทำหน้าที่แปลงคำสั่งที่ส่งมาให้อยู่ในรูปแบบที่สนับสนุนโดย ระบบจัดการฐานข้อมูล แต่ละชนิดอีกด้วย
4. แหล่งข้อมูล (Data Source) เป็นแหล่งข้อมูลที่ใช้ต้องการเข้าถึง

ตัวโปรแกรมจะเรียกใช้การเชื่อมโอดีบีซี ในการทำงานต่อไปนี้

1. ร้องขอการต่อเชื่อมกับแหล่งข้อมูล
2. ส่งคำสั่งเอสคิวแอลสู่แหล่งข้อมูล
3. กำหนดพื้นที่การจัดเก็บและรูปแบบของข้อมูล ที่เป็นผลลัพธ์จากเอสคิวแอล รีเควสท์ (SQL Request)
4. ร้องขอผลลัพธ์
5. ประมวลผลและจัดการกับข้อผิดพลาด
6. รายงานผลให้กับผู้ใช้ (ถ้าจำเป็น)
7. ร้องขอการคอมมิต (Commit) หรือ โรลแบ็ค (Roll Back) สำหรับควบคุมการประมวลผลทรานแซกชัน (Transaction)
8. ยกเลิกการติดต่อกับแหล่งข้อมูล

บทที่ 4

อินเทอร์เน็ตอินฟอร์เมชันเซิร์ฟเวอร์ ไอเอสเอพีไอ และ ซีจีไอ

4.1 อินเทอร์เน็ต อินฟอร์เมชัน เซิร์ฟเวอร์ (Internet Information Server)

อินเทอร์เน็ตอินฟอร์เมชันเซิร์ฟเวอร์ เป็นเว็บเซิร์ฟเวอร์(Web Server) ที่ให้บริการข้อมูล บนอินเทอร์เน็ต รวมไปถึงการประยุกต์การใช้งานต่าง ๆ ได้ อินเทอร์เน็ตอินฟอร์เมชันเซิร์ฟเวอร์จะส่งข้อมูลต่าง ๆ ผ่านทางโปรโตคอลที่มีชื่อว่า "ไฮเปอร์เท็กซ์ ทรานเฟอร์ โปรโตคอล (Hypertext Transfer Protocol : HTTP)" นอกจากนี้ อินเทอร์เน็ต อินฟอร์เมชัน เซิร์ฟเวอร์ ยังมีความสามารถในการให้บริการแบบ ไฟล์ ทรานเฟอร์ โปรโตคอล (File Transfer Protocol : FTP) และ บริการโกเฟอร์ (Gopher services) ได้

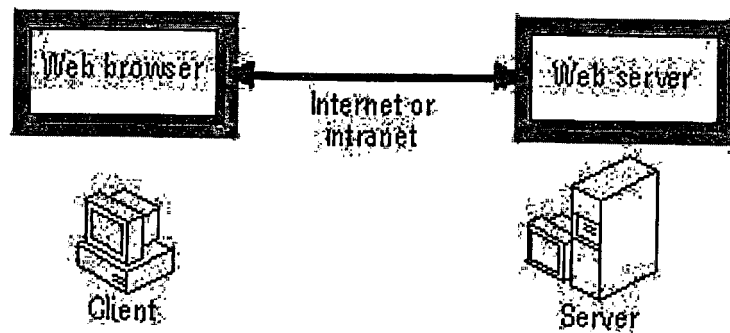
4.1.1 การใช้งานอินเทอร์เน็ต อินฟอร์เมชัน เซิร์ฟเวอร์

อินเทอร์เน็ต อินฟอร์เมชัน เซิร์ฟเวอร์ สามารถนำมาประยุกต์ใช้กับงานด้านต่าง ๆ ได้ตามความต้องการ อาทิ

- ให้บริการเวิร์ดไวด์เว็บ (World Wide Web:WWW) บนระบบอินเทอร์เน็ต หรืออินทราเน็ต โดยอาจอยู่ในลักษณะของการค้าขาย การศึกษา หรือแม้แต่บันเทิง เป็นต้น
- ให้บริการที่เป็นโปรแกรมโต้ตอบ (Interactive Program) ระหว่างเซิร์ฟเวอร์กับผู้ใช้
- ให้บริการข้อมูลเกี่ยวกับสินค้า และรับการสั่งซื้อสินค้าจากลูกค้า
- ใช้สำหรับการดำเนินการและการติดตามฐานข้อมูล

4.1.2 หลักการทำงานของอินเทอร์เน็ต อินฟอร์เมชัน เซิร์ฟเวอร์

บริการเวิร์ดไวด์เว็บ เป็นระบบพื้นฐานหนึ่งของการร้องขอ และการตอบสนองด้วยเว็บเบราว์เซอร์ (Web browsers) จะร้องขอข้อมูลต่างๆ ไปยังเว็บเซิร์ฟเวอร์ เมื่อเว็บเซิร์ฟเวอร์ ได้รับการร้องขอแล้ว ก็จะมีการตอบสนองกลับด้วยการส่ง ภาษาไฮเปอร์เท็กซ์ (Hypertext Markup Language : HTML) กลับไปให้กับ เว็บเบราว์เซอร์เพื่อแสดงผลต่อไป



รูปที่ 4-1 การติดต่อระหว่างเว็บเบราว์เซอร์กับเว็บเซิร์ฟเวอร์

การแสดงผลของโฮมเพจ (Home page) ในปัจจุบันสามารถนำเสนอได้สองรูปแบบใหญ่ๆ คือ

1. Static Page คือรูปแบบคงที่ของภาษา HTML ที่สร้างขึ้น โดยจะเก็บไว้ที่เว็บไซต์ (Web site) ที่พร้อมใช้งาน

2. Dynamic Page คือรูปแบบไม่คงที่ของภาษา HTML ที่สร้างขึ้น โดยตัวเซิร์ฟเวอร์ ต้องหน้าที่ในการสร้างขึ้นเองจากข้อมูลต่างๆ เพื่อทำการตอบสนองให้กับผู้ใช้ หรือ อาจจะเป็นรายการไฟล์ ที่มีอยู่หลาย ๆ ไฟล์สำหรับการเลือกใช้งาน

● 4.1.2.1 การร้องขอด้วย URL

บนอินเทอร์เน็ตทุกโฮมเพจ จะมีการระบุ URL ของแต่ละเครื่องไม่ซ้ำกัน เมื่อเว็บเบราว์เซอร์ มีการร้องขอไปยังเว็บเซิร์ฟเวอร์ เว็บเบราว์เซอร์ก็จะทำการส่ง URL ไปยังเว็บเซิร์ฟเวอร์นั้น หลังจากฝั่งเซิร์ฟเวอร์รู้ว่ามีกรร้องขอ ก็จะส่งข้อมูลที่เกี่ยวข้องกับการร้องขอนั้นกลับมาให้กับเว็บเบราว์เซอร์ เพื่อแสดงผลต่อไป

● 4.1.2.2 การตอบสนองของเว็บเซิร์ฟเวอร์

การตอบสนองของเว็บเซิร์ฟเวอร์ ผู้เว็บเบราว์เซอร์ที่มีการร้องขอ จะทำการส่ง ภาษา HTML ที่สร้างไว้กลับไป รูปแบบของภาษา HTML ที่จะส่งกลับส่วนมากจะมี 3 ชนิดคือ static HTML page, dynamic HTML page หรือ directory-listing page

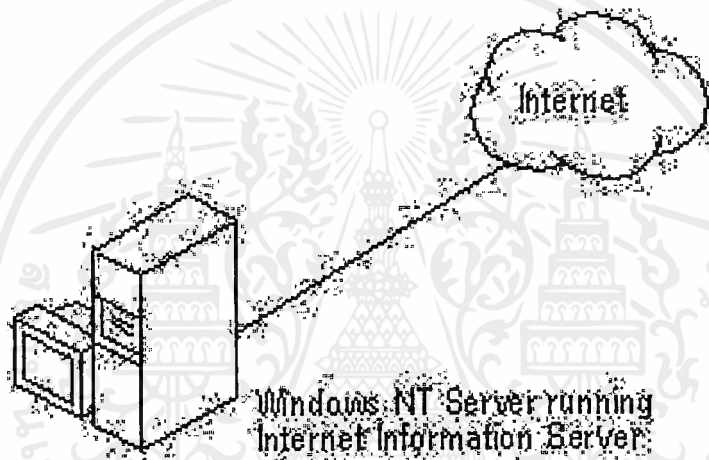
4.1.3 การนำ อินเทอร์เน็ต อินฟอรมะชัน เซิร์ฟเวอร์ ไปใช้งาน

อินเทอร์เน็ตอินฟอรมะชันเซิร์ฟเวอร์ มีจุดเด่นคือความยืดหยุ่นในการใช้งาน กล่าวคือ ในการนำอินเทอร์เน็ตอินฟอรมะชัน เซิร์ฟเวอร์ไปใช้นั้นจะเหมาะกับการทำงานต่าง ๆ ขององค์กร คือ สามารถเพิ่มขอบเขตของการติดตั้ง (scalable) จากระบบเซิร์ฟเวอร์ที่สนับสนุนการทำงานบนไซต์เดียว ไปเป็นเซิร์ฟเวอร์ที่สนับสนุนการทำงานแบบหลายไซต์ได้ ซึ่งปัจจุบันโฮมเพจของบริษัทใหญ่ๆ ที่มีผู้ใช้งานจำนวนมาก นิยมทำกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

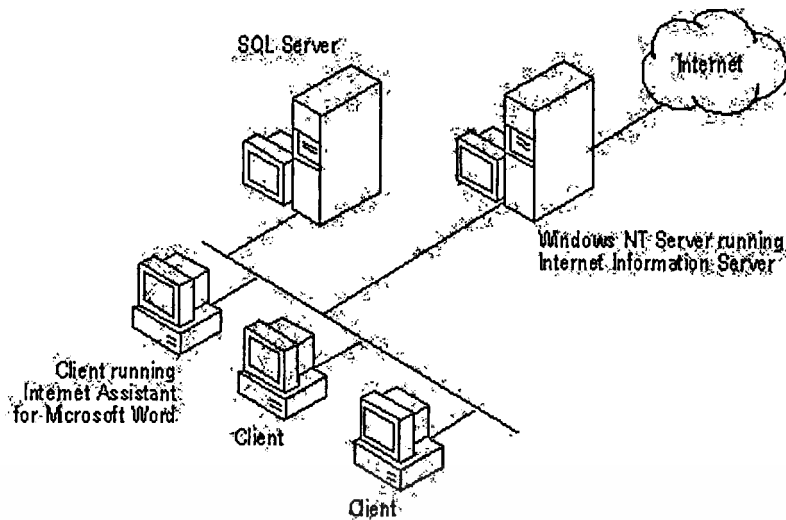
นอกจากนั้น อินเทอร์เน็ตอินฟอรมะชันเซิร์ฟเวอร์ จะทำการรวมสถานะแวดล้อมที่มีเข้าเป็นระบบเดียว เนื่องจากว่า อินเทอร์เน็ต อินฟอรมะชัน เซิร์ฟเวอร์ เองมีการรวมระบบความปลอดภัย (security) ของวินโดวส์เอ็นที และระบบเครือข่าย เข้าไว้ด้วยกัน จึงสามารถที่จะเพิ่มซอฟต์แวร์เข้าสู่คอมพิวเตอร์ที่มีอยู่ และรหัสของผู้ใช้ได้ ก็ไม่จำเป็นที่ต้องกำหนดว่าต้องนำคอมพิวเตอร์เครื่องใดมาใช้เป็น อินเทอร์เน็ต อินฟอรมะชัน เซิร์ฟเวอร์ ในระบบธุรกิจขนาดใหญ่ที่มีหลาย ๆ แผนกหรือหลาย ๆ กลุ่มงาน ซึ่งแต่ละแผนกต้องมีการใช้งาน อินเทอร์เน็ต อินฟอรมะชัน เซิร์ฟเวอร์ บนแฟ้มข้อมูลที่มีอยู่บนไฟล์ เซิร์ฟเวอร์ ของกลุ่มงานเฉพาะนั้น ดังนั้นเซิร์ฟเวอร์ ที่เป็นศูนย์กลางข้อมูลอาจจะต้องใช้เป็นข้อมูลเครือข่ายของบริษัท เช่นใช้เป็นคู่มือการทำงานของพนักงาน หรือเป็นไดเรกทอรี ของบริษัท

แต่ส่วนสำคัญที่จะกล่าวถึงการนำ อินเทอร์เน็ต อินฟอรมะชัน เซิร์ฟเวอร์ สามารถทำเป็นเว็บเซิร์ฟเวอร์ ซึ่งสามารถทำได้ ดังรูป



รูปที่ 4-2 การทำเว็บเซิร์ฟเวอร์ด้วยอินเทอร์เน็ต อินฟอรมะชัน เซิร์ฟเวอร์

ในองค์กรขนาดใหญ่สามารถที่จะใช้งาน อินเทอร์เน็ต อินฟอรมะชัน เซิร์ฟเวอร์ โดยใช้บนเครือข่ายภายในขององค์กรนั้นๆ โดยจะอนุญาตให้มีการติดต่อจากเครื่องไคลเอนท์ (Client) เข้าสู่เครื่องเซิร์ฟเวอร์ ซึ่งเครื่องไคลเอนท์สามารถเข้าค้นหาข้อมูลต่างๆ ที่อยู่บนเซิร์ฟเวอร์ หรือ เพื่อใช้โปรแกรมต่าง ๆ ที่ติดตั้งไว้บนตัวเซิร์ฟเวอร์ได้



รูปที่ 4-3 การติดต่อระหว่างเครื่องไคลเอนท์และเซิร์ฟเวอร์

อินเทอร์เน็ตอินฟอร์เมชันเซิร์ฟเวอร์ เป็นบริการหนึ่งที่มีมาให้กับวินโดวส์เอ็นทีเซิร์ฟเวอร์ (Windows NT Server) โดยที่ตัววินโดวส์เอ็นที มีความสามารถที่จะสร้างงานให้กับเซิร์ฟเวอร์ให้มีหลายๆ หน้าที่การทำงาน

4.2 ISAPI (Internet Server Application Programming Interface)

ISAPI คือ การให้บริการของ ไฮเปอร์เท็กซ์ ทรานมิตชัน โพรโตคอล (Hypertext Transmission Protocol : HTTP) ที่รันบน HTTP Server หรือ โปรแกรมเพื่อจุดประสงค์ต่างๆ ที่ไม่คงที่ (Dynamic) เช่น การเรียกค้นรายการจากฐานข้อมูล การสั่งซื้อสินค้า เป็นต้น ซึ่งจะแตกต่างจากการให้บริการแบบคงที่ (Static) เช่นการให้บริการเฉพาะเอกสารด้วยภาษา HTML แบบธรรมดาทั่วไป

ISAPI ทำงานในลักษณะเดียวกับ CGI (Common Gateway Interface) แต่ได้ปรับปรุงข้อด้อยต่างๆที่พบใน CGI และเพิ่มความสามารถใหม่ๆที่ CGI ไม่เคยมีมาก่อน

4.2.1 การทำงานของ ISAPI

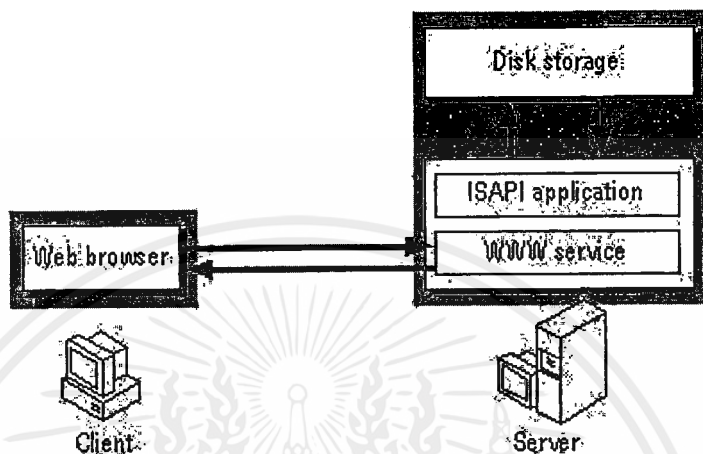
โดยพื้นฐานแล้ว ISAPI ทำงานเหมือนกับ CGI เกือบทุกประการ เช่น การกรอกแบบฟอร์ม ซึ่งปรากฏอยู่บนหน้าเว็บเพจของเว็บเบราว์เซอร์ ข้อมูลที่ผู้ใช้กรอกเข้าไปในแบบฟอร์มจะถูกส่งจากฝั่ง ไคลเอนท์ ผ่านเครือข่ายอินเทอร์เน็ต ไปยังฝั่งเซิร์ฟเวอร์ที่เป็นเว็บเซิร์ฟเวอร์ โดยจะมีโปรแกรมที่เป็น CGI หรือ ISAPI คอยประมวลผลข้อมูลเหล่านั้นอยู่ เช่น การนำข้อมูลที่ได้จากแบบฟอร์มมาเป็น คีย์ในการค้นหาข้อมูล บางอย่างบนระบบฐานข้อมูล (Database Server) เมื่อได้ผลลัพธ์จากฐานข้อมูลเรียบร้อยแล้ว โปรแกรม CGI หรือ ISAPI ก็จะแปลงข้อมูลที่ได้มานั้นให้อยู่ในรูปแบบของเอกสารภาษา HTML เพื่อส่งกลับไปแสดงผล บนเว็บเบราว์เซอร์ของผู้ใช้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 รูปแบบการใช้งานของ ISAPI

● 4.2.2.1 ISAPI Extension

จะเป็นโปรแกรมที่ใช้ในการโต้ตอบกันระหว่างเว็บไคลเอนท์ (ใช้เว็บเบราว์เซอร์) กับเว็บเซิร์ฟเวอร์ มีการหลักการทำงานดังรูป



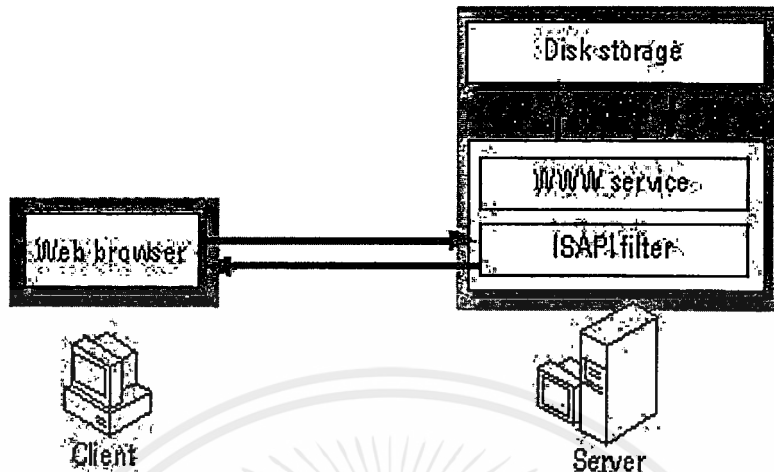
รูปที่ 4-4 หลักการทำงานของ ISAPI Extension

ส่วนของ ISAPI Extension จะทำงานอยู่บนเว็บเซิร์ฟเวอร์ ในรูปแบบของ DLL (Dynamic Link Library) การทำงานของ DLL มีวิธีการดังนี้คือ ในอดีตนั้นฟังก์ชันการทำงานต่าง ๆ ภายในโปรแกรมจะต้องถูกคอมไพล์ (Compile) และลิงค์ (Link) กับออบเจกต์ (Object) ที่สำคัญ ๆ ให้กลายเป็นภาษาเครื่อง (Machine Code) ก่อนที่จะใช้งาน ซึ่งฟังก์ชันทั้งหลายเหล่านั้นจะอยู่ตายตัวภายในโปรแกรม เพราะฉะนั้นเราจึงสามารถ ใช้งานฟังก์ชันได้จำนวนจำกัดคือได้เฉพาะฟังก์ชันที่อยู่ภายในโปรแกรมเท่านั้น ซึ่งเป็นการทำงานที่ไม่ยืดหยุ่นเลย ดังนั้นจึงได้มีการพัฒนา DLL ขึ้นมาเพื่อแก้ไขปัญหานี้ โดยจะเก็บฟังก์ชันมาตรฐานต่างๆ ไว้ในรูปแบบของไลบรารี เมื่อโปรแกรมหลักทำการโหลด DLL ที่ต้องการมาเก็บไว้ในหน่วยความจำ จะทำให้สามารถลิงค์ หรือเชื่อมต่อกับ ฟังก์ชันที่ต้องการได้ในขณะปฏิบัติงานจริง ในเมื่อ ISAPI ทำงานอยู่ในเนื้อที่หน่วยความจำของเว็บเซิร์ฟเวอร์ อยู่ตลอดเวลา ทำให้การทำงานเป็นไปด้วยความรวดเร็ว เมื่อมีการร้องขอจากเครื่องไคลเอนท์

● 4.2.2.2 ISAPI Filter

ISAPI Filter ทำหน้าที่เป็นตัวกรองและควบคุมการทำงานของเว็บเซิร์ฟเวอร์ สำหรับการติดต่อในรูปแบบของไฮเปอร์เท็กซ์โปรโตคอล โดยที่ก่อนและหลังการให้บริการของเว็บเซิร์ฟเวอร์ การกรองข้อมูลการของ ISAPI Filter นี้จะส่งผลโดยตรงต่อการรักษาความปลอดภัยของระบบ ISAPI Filter เข้ามาเพิ่มความสามารถ ให้กับไฮเปอร์เท็กซ์โปรโตคอล ส่งผลให้ผู้พัฒนาระบบสามารถเพิ่มคุณสมบัติใหม่ ๆ ให้กับไฮเปอร์เท็กซ์โปรโตคอลได้

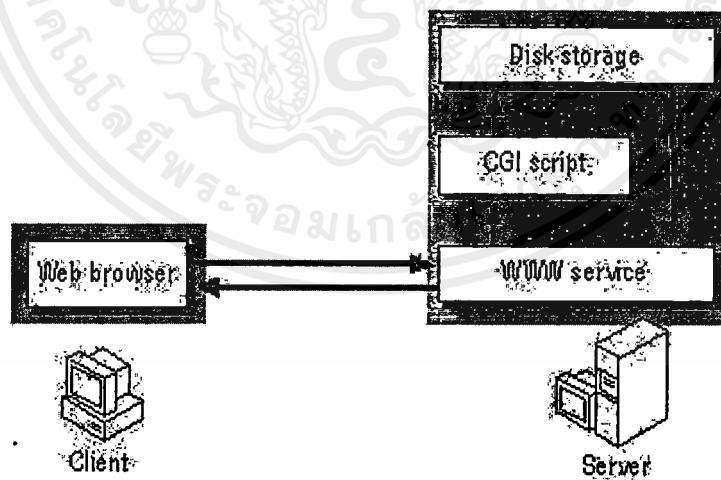
เมื่อเว็บเซิร์ฟเวอร์เริ่มทำงาน ISAPI Filter ก็จะถูกโหลดให้ปฏิบัติงานด้วยพร้อมโดยทันที เพื่อคอยตรวจสอบ การติดต่อจากเว็บไคลเอนท์



รูปที่ 4-5 หลักการทำงานของ ISAPI Filter

4.3 Common Gateway Interface (CGI)

CGI เป็นชุดของข้อกำหนดสำหรับการผ่านข้อมูลต่าง ๆ ระหว่างเว็บไคลเอนท์ กับ เว็บเซิร์ฟเวอร์ และ ตัวแอปพลิเคชันที่พัฒนาด้วย CGI ในส่วนของตัวเว็บไคลเอนท์สามารถเรียกใช้ CGI ได้โดยการใส่ลงในฟอร์มที่เขียนด้วยภาษา HTML หรือ การใส่ลิงค์เข้าไปในเพจของภาษา HTML บน เว็บเซิร์ฟเวอร์ ซึ่งมีหลักการทำงานดังรูป



รูปที่ 4-6 หลักการทำงานของ CGI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ไมโครซอฟต์ฟิวนเดชันคลาส

ไมโครซอฟต์ ฟิวนเดชัน คลาส (Microsoft Foundation Class) หรือเรียกสั้น ๆ ว่า เอ็มเอฟซี (MFC) เป็นไลบรารีที่บริษัท ไมโครซอฟต์ พัฒนาขึ้นมาสำหรับผู้ใช้ ไมโครซอฟต์วิซวลซีพลัสพลัส (Microsoft Visual C++) เพื่อใช้งานร่วมกับออบเจกต์ที่ผู้ใช้สร้างขึ้นเอง โดยออบเจกต์ต่าง ๆ สามารถนำมาใช้งานร่วมกัน ในที่นี้จะกล่าวถึงออบเจกต์เพียงบางตัวเท่านั้น

5.1 CRecordset



รูปที่ 5-1 CRecordset

CRecordset เป็นออบเจกต์ที่ใช้แสดงถึงกลุ่มเซตของเรคคอร์ด (Record) ที่เลือกมาจากค้ำาซอส (Data Source) โดย ออบเจกต์ CRecordset ใช้อยู่ 2 รูปแบบ คือ แบบไดนามิก (Dynamic) และ สแตติก (Static) โดยจะมีศัพท์ที่ใช้เรียกเฉพาะคือ Dynaset และ Snapshot ตามลำดับ โดยแบบ ไดนามิก เมื่อมีการเปลี่ยนแปลงที่ฐานข้อมูลที่เป็นค้ำาซอส ค่าของออบเจกต์ CRecordset ก็จะเปลี่ยนตามไปด้วย แต่สำหรับแบบ สแตติก หากมีการเปลี่ยนแปลงฐานข้อมูล ค่าของออบเจกต์ CRecordset จะไม่มีการเปลี่ยนแปลง

การใช้งาน CRecordset ต้องทำการติดต่อกับฐานข้อมูลและสร้างออบเจกต์เรคคอร์ดเซต ซึ่งจะส่งผ่านพอยน์เตอร์ (Pointer) ไปยังออบเจกต์ CDatabase โดยใช้เมมเบอร์ (Member) ฟังก์ชัน Open ซึ่งสามารถกำหนดได้ว่าจะให้เป็นแบบสแตติก หรือ ไดนามิก เมื่อเปิดฐานข้อมูลได้แล้ว เราสามารถทำการรีเฟรช (Refresh) เรคคอร์ดที่เพิ่มหรือเปลี่ยนแปลงได้ด้วยเมมเบอร์ฟังก์ชัน Requery และเมมเบอร์ฟังก์ชัน Close สำหรับการยกเลิกออบเจกต์ที่เลิกใช้งาน

5.1.1 CRecordset Class Member

5.1.1.1 Data Member

m_hstmt	ใช้สำหรับเรคคอร์ดเซตที่เกี่ยวข้องกับ โอดีบีซี CRecordset::m_hstmt
m_nFields	ตัวเลขฟิลด์ของค้ำาเมมเบอร์ภายในเรคคอร์ดเซต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	<code>CRecordset::m_nFields</code>	
<code>m_nParams</code>	<code>CRecordset::m_nParams</code>	ตัวเลขพารามิเตอร์ของคำสั่งในเรคคอร์ดเซต
	<code>CRecordset::m_nParams</code>	
<code>m_pDatabase</code>	<code>CRecordset::m_pDatabase</code>	พอยน์เตอร์ที่ชี้ไปยัง ออกเจกต์ <code>CDatabase</code> ผ่านเรคคอร์ดเซตที่ติดต่อกับคำสั่งขอ
	<code>CRecordset::m_pDatabase</code>	
<code>m_strFilter</code>	<code>CRecordset::m_strFilter</code>	ใช้ร่วมกับออกเจกต์ <code>CString</code> เพื่อใช้คำสั่ง <code>Where</code> ของภาษา SQL ซึ่งสามารถใช้ฟิลเตอร์เลือกเฉพาะเรคคอร์ดที่ต้องการได้
	<code>CRecordset::m_strFilter</code>	
<code>m_strSort</code>	<code>CRecordset::m_strSort</code>	ใช้ร่วมกับออกเจกต์ <code>CString</code> เพื่อใช้คำสั่ง <code>Order By</code> ของภาษา SQL ซึ่งสามารถควบคุมการจัดเรียงของเรคคอร์ด
	<code>CRecordset::m_strSort</code>	

5.1.1.2 Construction

<code>CRecordset</code>	<code>CRecordset::CRecordset</code>	สร้างออบเจกต์ <code>CRecordset</code> ซึ่งทำงานร่วมกับ
	<code>CRecordset::CRecordset</code>	
	<code>CRecordset(CDatabase* pDatabase = NULL);</code>	
<code>Open</code>	<code>CRecordset::Open</code>	เปิดเรคคอร์ดเซตด้วยการดึงข้อมูลมาจากตารางหรือโดยการคิวรี (Query)
	<code>virtual BOOL Open(UINT nOpenType = AFX_DB_USE_DEFAULT_TYPE, LPCTSTR lpszSQL = NULL, DWORD dwOptions = none); throw(CDBException, CMemoryException);</code>	
	<code>Return Value</code>	โดยจะส่งค่าที่ไม่เป็นศูนย์กลับถ้าออบเจกต์สามารถเปิดได้สำเร็จ
<code>Close</code>	<code>CRecordset::Close</code>	ปิดเรคคอร์ดเซตและ ODBC
	<code>virtual void Close();</code>	
	ตัวอย่างเช่น	
	<code>// Example for CRecordset::Close</code>	
	<code>// Construct a snapshot object</code>	
	<code>CCustSet rsCustSet(NULL);</code>	
	<code>if(!rsCustSet.Open())</code>	
	<code>return FALSE;</code>	
	<code>// Use the snapshot ...</code>	
	<code>// Close the snapshot</code>	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```
rsCustSet.Close( );
// Destructor is called when the function exits
```

5.1.1.3 Recordset Attributes

```
CanAppend      ส่งค่าที่ไม่ใช่ศูนย์กลับมาถ้าเรคคอร์ดใหม่สามารถเพิ่มเข้าไปยังเรคคอร์ดเซตได้ โดย
                ผ่านเมมเบอร์ฟังก์ชัน AddNew
                CRecordset::CanAppend
                BOOL CanAppend( ) const;
                Return Value

Can Bookmark   ส่งค่าที่ไม่ใช่ศูนย์กลับมาถ้าเรคคอร์ดเซตนั้นซัพพอร์ต (Support) บัญชีมาร์ค (Bookmark)
                CRecordset::CanBookmark
                BOOL CanBookmark( ) const;
                Return Value

CanRestart     ส่งค่าที่ไม่ใช่ศูนย์กลับมาถ้าเมมเบอร์ฟังก์ชัน Requery สามารถส่งค่าจากเรคคอร์ดเซตใหม่
                ไปได้
                CRecordset::CanRestart
                BOOL CanRestart( ) const;
                Return Value

CanScroll      ส่งค่าที่ไม่ใช่ศูนย์กลับมาถ้าสามารถสครู (Scroll) ผ่านเรคคอร์ดได้
                CRecordset::CanScroll
                BOOL CanScroll( ) const;
                Return Value

CanTransact    ส่งค่าที่ไม่ใช่ศูนย์กลับมาถ้าดาต้าซอส ซัพพอร์ต ทรานแซคชัน (Transaction)
                CRecordset::CanTransact
                BOOL CanTransact( ) const;
                Return Value

CanUpdate      ส่งค่าที่ไม่ใช่ศูนย์ถ้าเรคคอร์ดเซตสามารถอัปเดตได้ (เพิ่ม ลบ หรือ อัปเดต เรคคอร์ด)
                CRecordset::CanUpdate
                BOOL CanUpdate( ) const;
                Return Value

GetOdbcFieldCount  ส่งค่าตัวเลขของฟิลด์ในเรคคอร์ดเซตกลับ
                CRecordset::GetOdbcFieldCount
                short GetOdbcFieldCount( ) const;
                Return Value
```

GetRecordCount	ส่งค่าตัวเลขของเรคคอร์ดในเรคคอร์ดเซตกลับ CRecordset::GetRecordCount long GetRecordCount() const; Return Value โดยค่าที่ส่งกลับถ้าเป็นศูนย์แสดงว่าไม่มีเรคคอร์ด
GetStatus	เก็บค่าสถานะของเรคคอร์ดเซต CRecordset::GetStatus void GetStatus(CRecordsetStatus& rStatus) const;
GetTableName	เก็บชื่อของตารางที่สนใจเรคคอร์ดเซตนั้นอยู่ CRecordset::GetTableName const CString& GetTableName() const; Return Value
GetSQL	เก็บสตริงที่ใช้ในการเลือกเรคคอร์ดเซตที่เป็นคำสั่งเอสคิวเอล CRecordset::GetSQL const CString& GetSQL() const; Return Value
IsOpen	ส่งค่าที่ไม่เป็นศูนย์กลับ ถ้าก่อนหน้านี้ทำการเปิดได้สำเร็จ CRecordset::IsOpen BOOL IsOpen() const; Return Value
IsBOF	ส่งค่าที่ไม่เป็นศูนย์กลับ ถ้าเรคคอร์ดเซต ณ ตำแหน่งนั้น อยู่ก่อนเรคคอร์ดแรก CRecordset::IsBOF BOOL IsBOF() const; Return Value
IsEOF	ส่งค่าที่ไม่เป็นศูนย์กลับ ถ้าเรคคอร์ดเซต ณ ตำแหน่งนั้น อยู่ก่อนเรคคอร์ดสุดท้าย CRecordset::IsEOF BOOL IsEOF() const; Return Value
IsDeleted	ส่งค่าที่ไม่เป็นศูนย์กลับตำแหน่งของเรคคอร์ดเซตอยู่บนเรคคอร์ดที่ลบไปแล้ว CRecordset::IsDeleted BOOL IsDeleted() const; Return Value

5.1.1.4 Recordset Update Operations

AddNew	เตรียมพร้อมสำหรับการเพิ่มเรคคอร์ดใหม่ โดยการเรียก Update เพื่อเพิ่มความสมบูรณ์ CRecordset::AddNew virtual void AddNew(); throw(CDBException);
CancelUpdate	ยกเลิกการอัปเดตทั้งหมด ที่กระทำด้วย AddNew และ Edit CRecordset::CancelUpdate void CancelUpdate();
Delete	ลบตำแหน่งปัจจุบันออกจากเรคคอร์ดเซต และจะทำการเลื่อนตำแหน่งของเรคคอร์ดไปยังตำแหน่งอื่นได้ หลังจากลบ CRecordset::Delete virtual void Delete(); throw(CDBException);
Edit	เตรียมพร้อมสำหรับการแก้ไข ณ ตำแหน่งที่ปัจจุบันของเรคคอร์ด CRecordset::Edit virtual void Edit(); throw(CDBException, CMemoryException);
Update	ทำให้ AddNew หรือ Edit เพื่อให้การเพิ่มหรือแก้ไขข้อมูลสมบูรณ์ CRecordset::Update virtual BOOL Update(); throw(CDBException); Return Value โดยจะส่งค่าที่ไม่เป็นศูนย์ถ้าอย่างน้อยหนึ่งเรคคอร์ดสามารถอัปเดตได้อย่างสมบูรณ์

5.1.1.5 Recordset Navigation Operations

GetBookmark	กำหนดค่าบุ๊กมาร์ก (Bookmark) ของเรคคอร์ด เพื่อส่งเป็นพารามิเตอร์ให้กับออบเจกต์ CRecordset::GetBookmark void GetBookmark(CDBVariant& varBookmark); throw(CDBException, CMemoryException);
Move	
MoveFirst	เลื่อนตำแหน่งจากเรคคอร์ดปัจจุบัน ไปยังตำแหน่งแรกของเรคคอร์ดเซต CRecordset::MoveFirst void MoveFirst(); throw(CDBException, CMemoryException);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MoveLast	<p>เลื่อนตำแหน่งจากเรคคอร์ดปัจจุบันไปยังเรคคอร์ดสุดท้าย</p> <p>CRecordset::MoveLast</p> <pre>void MoveLast(); throw(CDBException, CMemoryException);</pre>
MoveNext	<p>เลื่อนตำแหน่งจากเรคคอร์ดปัจจุบันไปยังเรคคอร์ดที่อยู่ถัดไป ภายในแถวเดียวกัน</p> <p>CRecordset::MoveNext</p> <pre>void MoveNext(); throw(CDBException, CMemoryException);</pre>
MovePrev	<p>เลื่อนตำแหน่งจากเรคคอร์ดปัจจุบันไปยังเรคคอร์ดก่อนหน้านั้น ภายในแถวเดียวกัน</p> <p>CRecordset::MovePrev</p> <pre>void MovePrev(); throw(CDBException, CMemoryException);</pre>
SetAbsolutePosition	<p>เลื่อนตำแหน่งไปยังเรคคอร์ดเซ็คบนเรคคอร์ด ที่ตรงกับเลขเรคคอร์ดที่ต้องการ</p> <p>CRecordset::SetAbsolutePosition</p> <pre>void SetAbsolutePosition(long nRows); throw(CDBException, CMemoryException);</pre>
SetBookmark	<p>เลื่อนตำแหน่งไปยังเรคคอร์ดเซ็คบนเรคคอร์ดที่กำหนดโดยผู้ควมาร์ค</p> <p>CRecordset::SetBookmark</p> <pre>void SetBookmark(const CDBVariant& varBookmark); throw(CDBException, CMemoryException);</pre>
<p>5.1.1.6 · Other Recordset Operations</p>	
Cancel	<p>ยกเลิกการกระทำทั้งหมด</p> <p>CRecordset::Cancel</p> <pre>void Cancel();</pre>
FlushResultSet	<p>ส่งค่าที่ไม่เป็นศูนย์ ถ้ามีการรับข้อมูล</p> <p>CRecordset::FlushResultSet</p> <pre>BOOL FlushResultSet() const; throw(CDBException);</pre>
GetFieldValue	<p>ส่งค่าประจำฟิลด์ของเรคคอร์ดเสร็จกลับ</p> <p>CRecordset::GetFieldValue</p> <pre>void GetFieldValue(LPCTSTR lpszName, CDBVariant& varValue, short nFieldType = DEFAULT_FIELD_TYPE); throw(CDBException, CMemoryException);</pre>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void GetFieldValue( short nIndex, CDBVariant& varValue, short nFieldType =
DEFAULT_FIELD_TYPE );
throw( CDBException, CMemoryException );
```

```
void GetFieldValue( LPCTSTR lpszName, CString& strValue );
throw( CDBException, CMemoryException );
```

```
void GetFieldValue( short nIndex, CString& strValue );
throw( CDBException, CMemoryException );
```

GetODBCFieldInfo

ส่งค่าชนิดของข้อมูลที่เกี่ยวกับฟิลด์ของเรคคอร์ดเซต

CRecordset::GetODBCFieldInfo

```
void GetODBCFieldInfo( LPCTSTR lpszName, CODBCFieldInfo& fieldinfo );
throw( CDBException );
```

```
void GetODBCFieldInfo( short nIndex, CODBCFieldInfo& fieldinfo );
throw( CDBException );
```

GetRowsetSize

ส่งค่าตัวเลขของเรคคอร์ดที่ต้องการดึงข้อมูลในขณะที่ทำการเฟ็ท (Fetch) 1 ครั้ง

CRecordset::GetRowsetSize

```
DWORD GetRowsetSize( ) const;
```

Return Value

GetRowsFetched

ส่งค่าตัวเลขแท้จริงของแถวที่มีการดึงข้อมูลขณะทำการเฟ็ท

CRecordset::GetRowsFetched

```
DWORD GetRowsFetched( ) const;
```

Return Value

GetRowStatus

ส่งค่าสถานะของแถวที่สิ้นสุดการเฟ็ท

CRecordset::GetRowStatus

```
WORD GetRowStatus( WORD wRow ) const;
```

Return Value

IsFieldDirty

ส่งค่าไม่เป็นศูนย์กลับถ้าฟิลด์ที่พิจารณาในเรคคอร์ดปัจจุบันมีการเปลี่ยนแปลง

CRecordset::IsFieldDirty

```
BOOL IsFieldDirty( void* pv );
```

```
throw( CMemoryException );
```

Return Value

IsFieldNull

ส่งค่าไม่ใช่ศูนย์กลับถ้าฟิลด์ที่พิจารณาในเรคคอร์ดปัจจุบันมีค่าเป็นนูล (Null)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	<pre> CRecordset::IsFieldNull BOOL IsFieldNull(void* pv); throw(CMemoryException); Return Value </pre>
IsFieldNullable	<p>ส่งค่าที่ไม่เป็นศูนย์กลับถ้าฟิลด์ที่พิจารณาสามารถกำหนดค่าให้เป็น null ได้</p> <pre> CRecordset::IsFieldNullable BOOL IsFieldNullable(void* pv); throw(CDBException) </pre>
RefreshRowset	<p>รีเฟรชข้อมูลและสถานะของแถวที่ต้องการ</p> <pre> CRecordset::RefreshRowset void RefreshRowset(WORD wRow, WORD wLockType =SQL_LOCK_NO_CHANGE); </pre>
Requery	<p>รันคิวรีของเรคคอร์ดเซตอีกครั้ง เพื่อทำการรีเฟรชเรคคอร์ดที่เลือก</p> <pre> CRecordset::Requery virtual BOOL Requery(); throw(CDBException, CMemoryException); Return Value </pre>
SetFieldDirty	<p>มาร์กฟิลด์ที่มีการเปลี่ยนแปลงค่าในเรคคอร์ด</p> <pre> CRecordset::SetFieldDirty void SetFieldDirty(void* pv, BOOL bDirty = TRUE); </pre>
SetFieldNull	<p>กำหนดค่าฟิลด์ที่พิจารณาในเรคคอร์ดปัจจุบันให้เป็น null</p> <pre> CRecordset::SetFieldNull void SetFieldNull(void* pv, BOOL bNull = TRUE); </pre>
SetLockingMode	<p>เซ็ทให้อยู่ในโหมดล็อก (Locking Mode)</p> <pre> CRecordset::SetLockingMode void SetLockingMode(UINT nMode); </pre>
SetParamNull	<p>กำหนดค่าพารามิเตอร์เป็น null</p> <pre> CRecordset::SetParamNull void SetParamNull(int nIndex, BOOL bNull = TRUE); </pre>
SetRowsetCursorPosition	<p>เลื่อนตำแหน่งเคอร์เซอร์ไปยังเรคคอร์ดเซตที่ต้องการภายในแถวเดียวกัน</p> <pre> CRecordset::SetRowsetCursorPosition void SetRowsetCursorPosition(WORD wRow, WORD wLockType = SQL_LOCK_NO_CHANGE); </pre>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.1.7 Recordset Overridables

Check	เรียกค่าที่ส่งกลับจากฟังก์ชัน โอดีบีซีเอทีไอ (ODBC API Function) เพื่อทำการทดสอบ CRecordset::Check virtual BOOL Check(RETCODE nRetCode) const; Return Value โดยส่งค่าไม่เป็นศูนย์กลับถ้าค่าของ <i>nRetCode</i> คือ SQL_SUCCESS, SQL_SUCCESS_WITH_INFO, SQL_NO_DATA_FOUND หรือ SQL_NEED_DATA; นอกจากนั้นเป็นศูนย์
CheckRowsetError	หาความผิดพลาดที่เกิดขึ้นในขณะที่ทำการเพ็ช CRecordset::CheckRowsetError virtual void CheckRowsetError(RETCODE nRetCode); throw(CDBException);
DoBulkFieldExchange	ทำการแลกเปลี่ยนแถวของข้อมูลกับคำคำขอส่งเรคคอร์ดเซต CRecordset::DoBulkFieldExchange virtual void DoBulkFieldExchange(CFieldExchange* pFX); throw(CDBException);
DoFieldExchange	ทำการแลกเปลี่ยนข้อมูลระหว่างฟิลด์ของคำคำที่เป็นสมาชิกของเรคคอร์ดเซตกับเรคคอร์ดที่อยู่บนคำคำขอ CRecordset::DoFieldExchange virtual void DoFieldExchange(CFieldExchange* pFX); throw(CDBException);
GetDefaultConnect	เรียกค่าคีย์พอลต์ (Default) ของการติดต่อ CRecordset::GetDefaultConnect virtual CString GetDefaultConnect(); Return Value โดยจะทำการส่งค่าสตริงกับออกมา
GetDefaultSQL	เรียกค่าคีย์พอลต์ของคำสั่งภาษาเอสคิวแอล CRecordset::GetDefaultSQL virtual CString GetDefaultSQL(); Return Value
OnSetOptions	กำหนดค่าตัวเลือกสำหรับ โอดีบีซี CRecordset::OnSetOptions virtual void OnSetOptions(HSTMT hstmt);
SetRowsetSize	หาตำแหน่งของเรคคอร์ดที่ต้องการจะดึงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CRecordset::SetRowsetSize

virtual void SetRowsetSize(DWORD dwNewRowsetSize);

5.2 CDatabase



รูปที่ 5-2 CDatabase

CDatabase เป็นออบเจกต์ที่แสดงการติดต่อกับค้ำข้อมูล โดยสามารถทำการ โอเปอเรต (Operate) ผ่านค้ำข้อมูลได้ โดยค้ำข้อมูลอยู่บน โยสต์ที่เป็น คีบีเอ็มเอส (DBMS) เช่น ไมโครซอฟต์เอสคิวแอลเซิร์ฟเวอร์ ไมโครซอฟต์แอคเซส บอร์เลนดคีเบส เป็นต้น การใช้งานออบเจกต์ CDatabase มากกว่าหนึ่งตัว ภายใต้แอปพลิเคชันเดียวกัน

การใช้งาน CDatabase เริ่มจากการสร้างออบเจกต์ CDatabase โดยการใช้เมมเบอร์ฟังก์ชัน OpenEx ซึ่งการใช้งานออบเจกต์ CDatabase นั้นต้องใช้งานร่วมกับออบเจกต์ CRecordset เนื่องจาก CRecordset จะเป็นตัวส่งผ่านพอยต์เตอร์ของเรคคอร์ดเซตมายังออบเจกต์ CDatabase เพื่อยกเลิกการติดต่อและสิ้นสุดการใช้งานออบเจกต์ให้ใช้ฟังก์ชันเมมเบอร์ Close ซึ่งจะทำการปิดการติดต่อทั้งหมดของเรคคอร์ดเซต

5.2.1 CDatabase Class Member

5.2.1.1 Data Member

m_hdbc ทำการติดต่อกับค้ำข้อมูลโดยโอดีบีซี
CDatabase::m_hdbc

5.2.1.2 Construction

CDatabase สร้างออบเจกต์ CDatabase โดยการเรียกฟังก์ชัน OpenEx หรือ Open
CDatabase::CDatabase
CDatabase();
Open กำหนดการเชื่อมต่อกับค้ำข้อมูลผ่านทางโอดีบีซี
CDatabase::Open
virtual BOOL Open(LPCTSTR lpszDSN, BOOL bExclusive = FALSE, BOOL
bReadOnly = FALSE, LPCTSTR lpszConnect = "ODBC;", BOOL bUseCursorLib =
TRUE);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

throw( CDBException, CMemoryException );
OpenEx      กำหนดการเชื่อมต่อกับดาต้าซอสผ่านทางโอดีบีซี
            CDatabase::OpenEx
            virtual BOOL OpenEx( LPCTSTR lpszConnectionString, DWORD dwOptions = 0 );
            throw( CDBException, CMemoryException );
            Return Value
Close      ปิดการเชื่อมต่อระหว่างดาต้าซอส
            CDatabase::Close
            virtual void Close( );

```

5.2.1.3 Data Attributes

```

GetConnect  ส่งค่าสตริงที่ได้จากการเชื่อมต่อกับ โอดีบีซี ให้กับออบเจกต์ CDatabase
            CDatabase::GetConnect
            const CString& GetConnect( ) const;
            Return Value
IsOpen     ส่งค่าที่ไม่เป็นศูนย์กลับ ถ้าออบเจกต์ CDatabase ทำการเชื่อมต่อเข้ากับดาต้าซอสได้
            สำเร็จ
            CDatabase::IsOpen
            BOOL IsOpen( ) const;
            Return Value
GetDatabaseName  ส่งชื่อดาต้าเบสที่ใช้ในขณะนั้นกลับ
            CDatabase::GetDatabaseName
            CString GetDatabaseName( ) const;
            Return Value
CanUpdate  ส่งค่าที่ไม่เป็นศูนย์กลับถ้าออบเจกต์ CDatabase อัปเดตแล้ว
            CDatabase::CanUpdate
            BOOL CanUpdate( ) const;
            Return Value
CanTransact  ส่งค่าที่ไม่เป็นศูนย์กลับถ้าดาต้าซอสซัพพอร์ตการส่งข้อมูลแบบทรานแซคชัน
            CDatabase::CanTransact
            BOOL CanTransact( ) const;
            Return Value
SetLoginTimeout  กำหนดเวลาในหน่วยวินาทีเพื่อกำหนดสิ้นสุดเวลาการติดต่อกับดาต้าซอส
            CDatabase::SetLoginTimeout

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	<code>void SetLoginTimeout(DWORD <i>dwSeconds</i>);</code>
<code>SetQueryTimeout</code>	กำหนดเวลาในหน่วยวินาทีเพื่อกำหนดการสิ้นสุดเวลาการกระทำของคิวรี
	<code>CDatabase::SetQueryTimeout</code>
	<code>void SetQueryTimeout(DWORD <i>dwSeconds</i>);</code>
<code>GetBookmarkPersistence</code>	ระบุชนิดของการกระทำบนเรคคอร์ดที่เชื่อมผ่านบุ๊กมาร์ก
	<code>CDatabase::GetBookmarkPersistence</code>
	<code>DWORD GetBookmarkPersistence() const;</code>
	Return Value
<code>GetCursorCommitBehavior</code>	ระบุผลที่เกิดขึ้นหลังจากการส่งข้อมูลของเรคคอร์ดสำเร็จ
	<code>CDatabase::GetCursorCommitBehavior</code>
	<code>int GetCursorCommitBehavior() const;</code>
	Return Value
<code>GetCursorRollbackBehavior</code>	ระบุผลที่เกิดขึ้นหลังจากการ โรลแบค (Rollback) ข้อมูลของเรคคอร์ดเซต
	<code>CDatabase::GetCursorRollbackBehavior</code>
	<code>int GetCursorRollbackBehavior() const;</code>
	Return Value

5.2.1.4 Database Operations

<code>BeginTrans</code>	เริ่มต้นทรานแซคชัน
	<code>CDatabase::BeginTrans</code>
	<code>BOOL BeginTrans();</code>
	Return Value
<code>CommitTrans</code>	การทรานแซคชันสำเร็จ
	<code>CDatabase::CommitTrans</code>
	<code>BOOL CommitTrans();</code>
	Return Value
<code>Rollback</code>	ส่งคำสั่งทำขของทรานแซคชันกลับ
	<code>CDatabase::Rollback</code>
	<code>BOOL Rollback();</code>
	Return Value
<code>Cancel</code>	ยกเลิกการกระทำทั้งหมด
	<code>CDatabase::Cancel</code>
	<code>void Cancel();</code>
<code>ExecuteSQL</code>	ปฏิบัติงานตามคำสั่งภาษาเอสคิวเอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CDatabase::ExecuteSQL
ExecuteSQL( LPCSTR lpszSQL );
throw( CDBException );

```

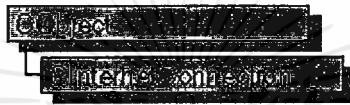
5.2.1.5 Database Overridables

```

OnSetOptions      กำหนดค่าทั้งหมดเป็นมาตรฐาน
                  CDatabase::OnSetOptions
                  virtual void OnSetOptions( HSTMT hstmt );

```

5.3 CInternetConnection



รูปที่ 5-3 CInternetConnection

CInternetConnection เป็นเอ็มเอฟซีคลาสที่ทำหน้าที่จัดการการติดต่อกับอินเทอร์เน็ตเซิร์ฟเวอร์ ซึ่งเป็นเอ็มเอฟซีคลาสพื้นฐานการใช้งานทั่วไป โดยที่จะมีคลาสที่ทำหน้าที่ใกล้เคียงกันเช่น CFtpConnetion, CHttpConection, CGopherConnection แต่ละคลาสต่าง ๆ กันนี้จะมีฟังก์ชันการติดต่อกับเซิร์ฟเวอร์ต่าง ๆ กัน FTP, HTTP หรือ Gopher

การใช้งาน การจะติดต่อกับอินเทอร์เน็ตเซิร์ฟเวอร์โดยตรง จำเป็นต้องใช้ออบเจกต์สองตัวต่อไปนี้คือ CInternetSession กับ CInternetConnection

5.3.1 CInternetConnection Class Members

5.3.1.1 Construction

```

CInternetConnection  สร้างออบเจกต์ CInternetConnection
                    CInternetConnection::CInternetConnection
                    CInternetConnection( CInternetSession* pSession, LPCTSTR pstrServer,
                    INTERNET_PORT nPort = INTERNET_INVALID_PORT_NUMBER, DWORD
                    dwContext = 1 );

```

5.3.1.2 Operations

```

GetContext          เก็บค่าหลายเลขประจำตัวคอนเท็กซ์ (Context) สำหรับการติดต่อกับออบเจกต์
                    CInternetConnection::GetContext

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DWORD GetContext() const;

Return Value

GetSession

เก็บค่าพอยต์เตอร์ให้กับออบเจกต์ CInternetSession สำหรับการติดต่อ

CInternetConnection::GetSession

CInternetSession* GetSession() const;

Return Value

GetServerName

เก็บชื่อเซิร์ฟเวอร์ที่ทำการติดต่อ

CInternetConnection::GetServerName

CString GetServerName() const;

Return Value

5.3.1.3 Operators

operator HINTERNET

จับของการติดต่อกับอินเทอร์เน็ต

CInternetConnection::operator HINTERNET

operator HINTERNET() const;

5.4 CFtpConnection



รูปที่ 5-4 CFtpConnection

CFtpConnection เป็นออบเจกต์ที่ทำหน้าที่ทั้งจัดการการติดต่อบริการเอฟทีพี (FTP) เข้ากับอินเทอร์เน็ต เซิร์ฟเวอร์ และอนุญาตให้มีการติดต่อกับไฟล์ ไคเรกทอรี ได้โดยตรงบนตัวเอฟทีพีเซิร์ฟเวอร์ เอฟทีพีเป็นหนึ่งในสามของบริการบนอินเทอร์เน็ตที่มีผู้ใช้บริการมากที่สุด และยังเป็น 1 ใน 3 ของบริการด้านอินเทอร์เน็ตที่ครอบคลุมการทำงานด้วย WinInet Class

การใช้งานก่อนที่จะทำการเชื่อมต่อเข้ากับตัวเอฟทีพีเซิร์ฟเวอร์ต้องทำการสร้างออบเจกต์ CInternetSession CFtpConnection เราไม่สามารถทำการสร้าง CFtpConnection โดยตรงได้ ถ้าจะทำการสร้างต้องเรียกใช้ดังนี้ CInternetSession::GetFtpConnection สำหรับการสร้างออบเจกต์ CFtpConnection และทำการส่งค่าพอยต์เตอร์กลับมายังตัวมัน

5.4.1 CFtpConnection Class Members

5.4.1.1. Construction

CftpConnection สร้างออบเจกต์ CFtpConnection ด้วยคำสั่ง CInternetSession::GetFtpConnection
 CFtpConnection::CftpConnection
 CFtpConnection();

5.4.1.2 Operations

SetCurrentDirectory กำหนดไดเรกทอรีปัจจุบันสำหรับบริการเอฟทีพี
 CFtpConnection::SetCurrentDirectory
 BOOL SetCurrentDirectory(LPCTSTR *pstrDirName*);
 Return Value
 โดยค่าที่ส่งกลับคือถ้าสำเร็จจะส่งค่าที่ไม่ใช่ศูนย์กลับมา ถ้าไม่สำเร็จจะส่งค่าเป็นศูนย์กลับมา และฟังก์ชันของ Win 32 ที่ชื่อ GetLastError จะถูกเรียกขึ้นมาบอกว่าเกิดปัญหาอะไรขึ้น

GetCurrentDirectory เก็บค่าไดเรกทอรีปัจจุบันสำหรับการติดต่อ
 CFtpConnection::GetCurrentDirectory
 BOOL GetCurrentDirectory(CString& *strDirName*) const;
 BOOL GetCurrentDirectory(LPTSTR *pstrDirName*, LPDWORD *lpdwLen*) const;
 Return Value
 ส่งค่าไม่เป็นศูนย์กลับถ้าสำเร็จ ถ้าไม่สำเร็จฟังก์ชัน GetLastError ของ Win 32 จะถูกเรียกขึ้นมาบอกถึงสาเหตุของความผิดพลาด

GetCurrentDirectoryAsURL เก็บค่าไดเรกทอรีปัจจุบันสำหรับการติดต่อ ในรูปของ URL
 CFtpConnection::GetCurrentDirectoryAsURL
 BOOL GetCurrentDirectoryAsURL(CString& *strDirName*) const;
 BOOL GetCurrentDirectoryAsURL(LPTSTR *pstrDirName*, LPDWORD
lpdwLen) const;
 Return Value

RemoveDirectory ลบไดเรกทอรีที่กำหนดออกจากเซิร์ฟเวอร์
 CFtpConnection::RemoveDirectory
 BOOL RemoveDirectory(LPCTSTR *pstrDirName*);
 Return Value

CreateDirectory สร้างไดเรกทอรีที่ต้องการบนเซิร์ฟเวอร์
 CFtpConnection::CreateDirectory

	<code>BOOL CreateDirectory(LPCTSTR <i>pstrDirName</i>);</code>
	Return Value
Rename	เปลี่ยนชื่อไฟล์บนเซิร์ฟเวอร์ <code>CFtpConnection::Rename</code> <code>BOOL Rename(LPCTSTR <i>pstrExisting</i>, LPCTSTR <i>pstrNew</i>);</code>
	Return Value
Remove	ลบไฟล์ออกจากเซิร์ฟเวอร์ <code>CFtpConnection::Remove</code> <code>BOOL Remove(LPCTSTR <i>pstrFileName</i>);</code>
	Return Value
PutFile	วางไฟล์บนเซิร์ฟเวอร์ <code>CFtpConnection::PutFile</code> <code>BOOL PutFile(LPCTSTR <i>pstrLocalFile</i>, LPCTSTR <i>pstrRemoteFile</i>, DWORD <i>dwFlags</i> = FTP_TRANSFER_TYPE_BINARY, DWORD <i>dwContext</i> = 1);</code>
	Return Value
GetFile	คัดลอกไฟล์มาจากการเชื่อมต่อ <code>CFtpConnection::GetFile</code> <code>BOOL GetFile(LPCTSTR <i>pstrRemoteFile</i>, LPCTSTR <i>pstrLocalFile</i>, BOOL <i>bFailIfExists</i> = TRUE, DWORD <i>dwAttributes</i>=FILE_ATTRIBUTE_NORMAL, DWORD <i>dwFlags</i> = FTP_TRANSFER_TYPE_BINARY, DWORD <i>dwContext</i> = 1);</code>
	Return Value
OpenFile	เปิดไฟล์ผ่านการเชื่อมต่อ <code>CFtpConnection::OpenFile</code> <code>CInternetFile* OpenFile(LPCTSTR <i>pstrFileName</i>, DWORD <i>dwAccess</i> = GENERIC_READ, DWORD <i>dwFlags</i> = FTP_TRANSFER_TYPE_BINARY, DWORD <i>dwContext</i> = 1);</code>
	Return Value
Close	ปิดการเชื่อมต่อระหว่างเซิร์ฟเวอร์ทั้งหมด <code>CFtpConnection::Close</code> <code>virtual void Close();</code>

5.5.CHtTpServer

CHtTpServer เป็นคลาสที่ต้องทำงานร่วมกับคลาส CHtTpServerContext โดยจะมีหน้าที่ในการจัดการเกี่ยวกับฟังก์ชันของไอเอสเอทีไอ โดยที่คลาส CHtTpServer นี้รองรับการทำงานฟังก์ชัน ไอเอสเอทีไอทุกตัว รวมทั้งซีจีไอด้วย

หลักการเมื่อ ISAPI HTTP Server ได้รับความร้องขอจากไคลเอนท์ผ่านบราวเซอร์ ออบเจกต์ CHtTpServer จะถูกสร้างและกำหนดค่าขึ้นมา และจะทำการสร้าง ออบเจกต์ CHtTpServerContext ขึ้นด้วย โดยหนึ่งโมดูลสามารถที่จะมีออบเจกต์ CHtTpServer ได้เพียงออบเจกต์เดียว

5.5.1 CHtTpServer Class Members

5.5.1.1 Constructor

CHtTpServer สร้างออบเจกต์ CHtTpServer
 CHtTpServer::CHtTpServer
 CHtTpServer(TCHAR cDelimiter);

5.5.1.2 Overridables

CallFunction ค้นหาและปฏิบัติตามคำสั่งที่ส่งมาจาก URL
 CHtTpServer::CallFunction
 virtual int CallFunction(CHtTpServerContext* pCtxt, LPTSTR pszQuery, LPTSTR pszCommand);
 Return Value โดยค่าที่ส่งกลับจะมีความหมายดังนี้

callOK	เรียกใช้งานฟังก์ชันสมบูรณ์
callParamRequired	พารามิเตอร์บางตัวหายไป
callBadParamCount	พารามิเตอร์มากหรือน้อยเกินไป
callBadCommand	หาชื่อของคำสั่งไม่ครบ
callNoStackSpace	สแตกไม่พอ
callNoStream	CHtmlStream ไม่ว่าง
callMissingQuote	รูปแบบของพารามิเตอร์ไม่ถูก
callMissingParams	ไม่มีพารามิเตอร์
callBadParam	พารามิเตอร์ไม่ถูกตามรูปแบบ

OnParseError ตรวจจับความผิดพลาดเพื่อส่งกลับ
 CHtTpServer::OnParseError
 virtual BOOL OnParseError(CHtTpServerContext* pCtxt, int nCause);
 Return Value

HttpExtensionProc ใช้ฟังก์ชัน Callback เพื่ออ่านข้อมูลจากไคลเอนท์ เพื่อตัดสินใจการกระทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	CHttpServer::HttpExtensionProc
	virtual DWORD HttpExtensionProc(EXTENSION_CONTROL_BLOCK *pECB
);
	Return Value
GetExtensionVersion	เก็บค่าเวอร์ชันของ DLL
	CHttpServer::GetExtensionVersion
	virtual BOOL GetExtensionVersion(HSE_VERSION_INFO *pVer);
	Return Value
ConstructStream	สร้างออบเจกต์ CHtmlStream
	CHttpServer::ConstructStream
	virtual CHtmlStream* ConstructStream();
	Return Value
5.5.1.3 Attributes	
StartContent	ใส่แท็กเริ่มต้นของภาษา HTML ในออบเจกต์ CHtmlStream เพื่อส่งกลับ ไปให้ เครื่องไคลเอนท์
	CHttpServer::StartContent
	virtual void StartContent(CHttpRequestContext* pCtxt) const;
EndContent	ใส่แท็กสิ้นสุดของภาษา HTML ให้กับออบเจกต์ CHtmlStream เพื่อส่งกลับ ไปให้ เครื่องไคลเอนท์
	CHttpServer::EndContent
	virtual void EndContent(CHttpRequestContext* pCtxt) const;
WriteTitle	ใส่ไตเติ้ลแท็กของภาษา HTML ให้กับ ออบเจกต์ CHtmlStream เพื่อส่งกลับ ไปให้ เครื่องไคลเอนท์
	CHttpServer::WriteTitle
	virtual void WriteTitle(CHttpRequestContext* pCtxt) const;
GetTitle	เก็บค่าไตเติ้ลของเอกสาร HTML ส่งกลับ ให้กับเครื่องไคลเอนท์
	CHttpServer::GetTitle
	virtual LPCTSTR GetTitle() const;
	Return Value
AddHeader	เพิ่มส่วนหัวก่อนที่จะส่งกลับ ไปยังเซิร์ฟเวอร์
	CHttpServer::AddHeader
	void AddHeader(CHttpRequestContext* pCtxt, LPCTSTR pszString) const;
InitInstance	กำหนดค่าเริ่มต้นให้กับออบเจกต์ CHttpRequest
	CHttpServer::InitInstance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
virtual BOOL InitInstance(CHttpServerContext* pCtxt );
```

```
Return Value
```

5.6 CHttpServerContext

CHttpServerContext มีเครื่องมือสำหรับออบเจกต์ CHttpServer เพื่อช่วยในการประมวลผลข้อมูลที่ไคลเอนท์ทำการส่งมายังเซิร์ฟเวอร์ โดยที่ขณะที่ไมโครซอฟต์ อินเทอร์เน็ต อินฟอร์เมชัน เซิร์ฟเวอร์ ได้รับการร้องขอจากไคลเอนท์ผ่านบราวเซอร์

5.6.1 CHttpServerContext Class Members

5.6.1.1. Data Members

m_pECB ส่งพอยต์เตอร์ชี้ไปยัง EXTENSION_CONTROL_BLOCK

```
CHttpServerContext::m_pECB
```

m_pStream ส่งพอยต์เตอร์ชี้ไปยัง CHtmlStream.

```
CHttpServerContext::m_pStream
```

5.6.1.2 Construction

CHttpServerContext สร้างออบเจกต์ CHttpServerContext

```
CHttpServerContext::CHttpServerContext
```

```
CHttpServerContext( EXTENSION_CONTROL_BLOCK* pECB );
```

5.6.1.3 Operations

GetServerVariable ก็อปปี้ข้อมูลที่ส่งผ่านเซิร์ฟเวอร์ใส่ในบัฟเฟอร์

```
CHttpServerContext::GetServerVariable
```

```
BOOL GetServerVariable( LPTSTR lpszVariableName, LPVOID lpvBuffer,
LPDWORD lpdwSize );
```

WriteClient ส่งข้อมูลกับยังไคลเอนท์

```
CHttpServerContext::WriteClient
```

```
BOOL WriteClient( LPVOID lpvBuffer, LPDWORD lpdwBytes, DWORD
dwReserved = 0 );
```

```
Return Value
```

ReadClient อ่านข้อมูลส่วนบอดีเครื่องไคลเอนท์ที่ส่งมาเข้าบัฟเฟอร์เพื่อเตรียมนำไปใช้ต่อ

```
CHttpServerContext::ReadClient
```

```
BOOL ReadClient( LPVOID lpvBuffer, LPDWORD lpdwSize );
```

```
Return Value
```

ServerSupportFunction CHttpServerContext::ServerSupportFunction

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BOOL ServerSupportFunction(DWORD *dwHSERRequest*, LPVOID *lpvBuffer*,
LPDWORD *lpdwSize*, LPDWORD *lpdwDataType*);

Return Value

5.6.1.4 Operators

operator <<

เขียนข้อมูลแบบสตรีม (Stream)

CHttpServerContext::operator <<

void operator<<(LPCTSTR *psz*);

void operator<<(long int *dw*);

void operator<<(short int *w*);

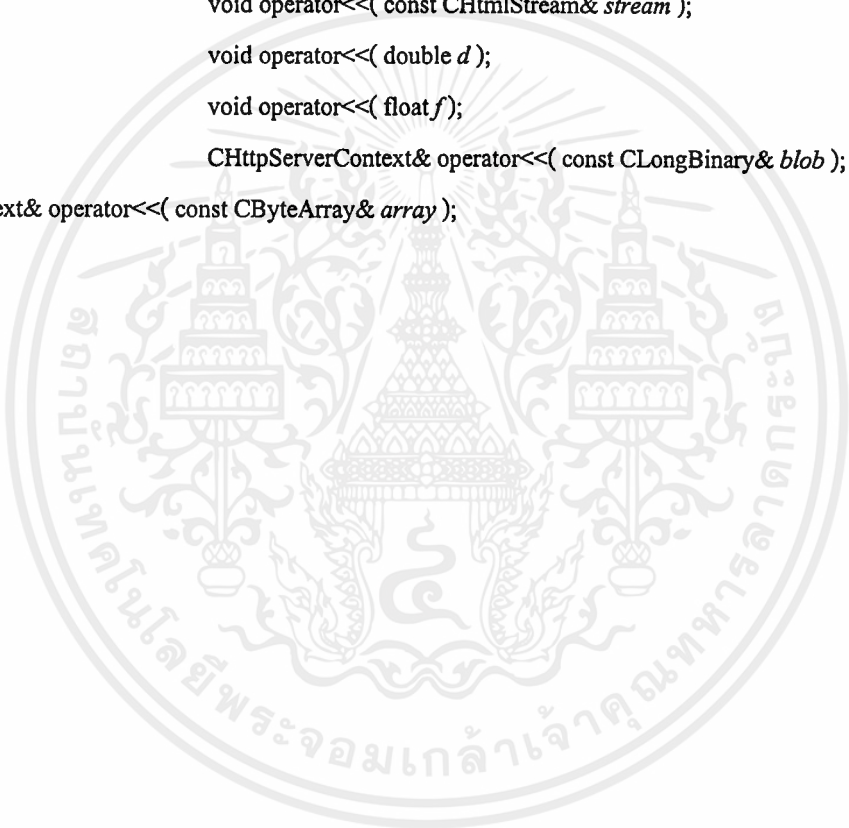
void operator<<(const CHtmlStream& *stream*);

void operator<<(double *d*);

void operator<<(float *f*);

CHttpServerContext& operator<<(const CLongBinary& *blob*);

CHttpServerContext& operator<<(const CByteArray& *array*);



บทที่ 6

ภาษาไฮเปอร์เท็กซ์ มาร์กอัพ

6.1 รู้จักกับ HTML

HTML (Hyper Text Markup Language) เป็นภาษาที่พัฒนาขึ้นมาเพื่อใช้ในการสร้าง เอกสารที่ใช้ บน อินเทอร์เน็ต ซึ่งเป็นภาษาที่เขียนได้ง่าย มีรูปแบบที่เป็นมาตรฐานที่แน่นอน โดยที่ปัจจุบันได้พัฒนาถึงเวอร์ชัน 4.0 แต่ที่ใช้กันอยู่โดยมากคือ HTML 3.2 ซึ่งหน่วยงานที่รับผิดชอบในการกำหนดมาตรฐานคือ W3C (<http://www.w3c.org>) โดยหลักการเว็บเบราว์เซอร์จะตีความแปลจากภาษา HTML มาแสดงผลให้กับผู้ใช้

โดยรูปแบบของภาษา HTML จะมีส่วนที่เรียกว่า Markup tag (ส่วนขยายลักษณะหรือเรียกย่อๆว่า แทก) ที่จะคอยบอกให้โปรแกรมเว็บเบราว์เซอร์ ทราบว่าหน้าจอหรือข้อความนั้น ๆ จะถูกแสดงอยู่ในลักษณะอย่างไร ส่วน ขยายลักษณะหรือ Markup tag นี้ จะเป็นข้อความหรือคำสั่งที่มีเครื่องหมาย <> ครอบเอาไว้ และใช้เครื่องหมาย “/” เป็นตัวบอการสิ้นสุดของคำสั่ง เช่น ขึ้นต้นด้วย <HTML> จบด้วย </HTML> ภาษาสั่งงาน HTML นี้ มีการพัฒนาปรับปรุงให้มีความสามารถในการแสดงผลในรูปแบบแปลกๆ ใหม่ๆ ขึ้นเรื่อยๆ จาก HTML 1.0 มาจนถึง HTML 3.2 ในปัจจุบันที่มีลูกเล่นมากมาย แต่ผู้ใช้บริการต้องระวังในกรณีที่ใช้ โปรแกรมเว็บเบราว์เซอร์เก่าๆ ที่ไม่สามารถรับคำสั่ง HTML 3.0 หรือ 3.2 ได้

6.2 การเขียน HTML เบื้องต้น

ภาษา HTML กำหนดส่วนประกอบหลักๆ (Element) ขึ้นมาเพื่อใช้ในการเขียนเอกสาร โดยที่ส่วน ประกอบหนึ่งๆ อาจประกอบด้วย แทก (เริ่มต้นด้วยอักษร <และลงท้ายด้วยตัว อักษร>) แอททริบิวท์ (Attribute) และ ตัวอักษร(Text) ซึ่งอยู่ในรูปแบบดังนี้

<tag-name>text</tag-name> <tag-name attribute-name=arg>text</tag-name> และ <tag-name>

6.2.1 ส่วนประกอบที่มีต้องมีในทุกเพจ

ทุกเพจที่เขียนด้วยภาษา HTML จะมีตัวกำหนดขอบเขตของเอกสาร ซึ่งประกอบด้วย ส่วนหัว เอกสาร ส่วนเนื้อเรื่องของเอกสารซึ่งจะอยู่ในรูปแบบดังนี้

```
<HTML>
<HEAD>
<TITLE>ชื่อหัวเรื่อง</TITLE>
</HEAD>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<BODY Attribute ... >

รายละเอียดของเพจ

</BODY>

</HTML>

- <HTML> ... </HTML> เป็นส่วนที่ใช้กำหนดขอบเขตของเอกสาร
- <HEAD> ... </HEAD> เป็นส่วนที่ใช้กำหนดขอบเขตของหัวเรื่อง ซึ่งโดยส่วนใหญ่แล้วหัวเรื่อง
- <BODY> ... </BODY> ใช้กำหนดขอบเขตของเนื้อหาในเอกสาร

6.2.2 คำสั่งกำหนดหัวข้อ (Header Element)

รูปแบบคำสั่งจะอยู่ในรูป <h#> ... </h#> โดยที่ # แทนตัวเลขตั้งแต่ 1-6 ใช้ในการแสดง ส่วนของเอกสารที่ต้องการจะให้เป็นหัวข้อ

6.2.3 คำสั่งจัดการลำดับ (List Element)

โดยคำสั่งที่ใช้ในการจัดการลำดับ คือ คือการจัดการลำดับแบบไม่จำกัด และ จะจำกัดการจัดการลำดับ

6.2.4 การจัดย่อหน้า (Paragraph)

ย่อหน้า เป็นกลุ่มประโยค ที่อยู่ใน คำสั่ง <p align="center" > ... </p> คำแอททริบิวต์ align= "center" เป็นตัวบอกให้ทราบว่าเราต้องการให้กลุ่มของประโยค อยู่ตรงกลางของเพจ

6.2.5 Logical Styles และ Physical Styles

มีคำสั่งบางคำสั่งที่ใช้กำหนดรูปแบบตัวอักษร ให้เป็นรูปแบบต่าง ๆ เช่น ตัวเอียง หรือตัวเข้ม เราสามารถแบ่งชุดคำสั่งนี้เป็น 2 กลุ่ม คือ ทางความหมาย (Logical) และทางกายภาพ (Physical) Logical จะเปลี่ยนตัวอักษร โดยใช้ความหมายในการใช้งานเป็นหลัก เช่น ต้องการเน้น ต้องการแสดงว่าเป็นตัวแปร ตัวอักษรที่ปรากฏจริงเป็นอย่างไรขึ้นอยู่กับบราวเซอร์ ส่วนกายภาพจะเปลี่ยนตัวอักษร ตามที่จะปรากฏจริง การเลือกใช้กลุ่มใดขึ้นอยู่กับดุลยพินิจและควรใช้กลุ่มเดียวกันทั้งเอกสาร แต่อย่างไรก็ตาม ควรตระหนักว่ารูปแบบทางความหมาย อาจจะไม่สนับสนุนโดย HTML เวอร์ชันใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.5.1 รูปแบบทางความหมาย

คำสั่งดังต่อไปนี้จะใช้เป็นหัวเรื่องของหนังสือหรือภาพยนตร์

` ... `

(emphasize) เน้น

` ... `

(strong emphasize) เน้นหนัก

`<cite> ... </cite>`

คำสั่งดังต่อไปนี้จะใช้ในงานคอมพิวเตอร์

`<code> ... </code>`

แสดงว่าเป็นโค้ด (code)

`<samp> ... </samp>`

แสดงว่าเป็นเอาต์พุต (sample output)

`<kbd> ... </kbd>`

แสดงว่าเป็น อินพุตจาก คีย์บอร์ด (keyboard)

`<var> ... </var>`

แสดงว่าเป็นตัวแปร (variable)

6.2.5.2 รูปแบบทางกายภาพ

` ... `

ตัวเข้ม (bold)

`<i> ... </i>`

ตัวเอียง (italic)

`<tt> ... </tt>`

ตัวพิมพ์ (Typewriter font)

6.2.6 ระบุสถานที่ (Address)

เป็น คำสั่ง ที่มักใช้ใช้การแสดงผล Email Address เช่น

`<address> If you have problem mail to webmaster@mls.dynip.com</address>`

6.2.7 การขีดเส้นใต้ (Horizontal Rules)

`<hr>` เป็นแทกที่ใช้ในการสร้างเส้นแนวนอน ส่วนมากใช้ในการแยกข้อความที่ไม่ขึ้นต่อกัน เป็นส่วนๆ และเราสามารถเปลี่ยนขนาดของเส้นได้

6.2.8 การใส่รูปภาพ

เว็บเบราว์เซอร์ สามารถแสดงภาพ (ซึ่งจะแสดงติดกับอักษร) ที่มีรูปแบบมาตรฐานของภาพ เช่น XBM, TIF, JPG, GIF และ PNG เราสามารถใส่ภาพในโฮมเพจ โดยใช้ คำสั่ง

```

```

6.2.9 การเชื่อมโยง (Link)

ดังที่ได้กล่าวมาแล้วว่าคุณสมบัติหนึ่งของภาษา HTML คือความสามารถในการเชื่อมโยงระหว่างเอกสาร ซึ่งไม่เกี่ยวข้องกัน โดยเบราว์เซอร์แสดงส่วนนี้ให้เด่นกว่าส่วนอื่น เรียกว่า anchor มีรูปแบบดังนี้

```
<a href="ชื่อไฟล์">ข้อความที่จะแสดงเป็นลิงค์</a>
```

6.2.10 mailto

เป็นคำสั่งที่ใช้ในการสร้างลิงค์ให้กับผู้ใช้ สามารถส่งอีเมลล์ให้กับผู้สร้างเพจนั้น (หรือบุคคลที่ผู้สร้างเพจต้องการ) โดยรูปแบบคือ

```
<a href="mailto:// ชื่ออีเมลล์แอดเดรสที่ต้องการ">ชื่อที่ต้องการแสดงบนหน้าจอ</a>
```

6.3 คำสั่งที่ใช้บ่อยๆ ของภาษา HTML

<!-->	สร้างคำอธิบายขัง html ของคุณที่จะไม่สามารถมองเห็นบนเพจ
<a href>	สร้าง link ไปยังเพจอื่นๆ หรือ link ไปยังบริเวณอื่นบนเพจเดียวกัน
<a name>	สร้างจุด anchor กระโดด ไปยังส่วนบางส่วนของเพจ
<address>	ระบุชื่อหรือที่อยู่บนเว็บเพจ
<applet>	เริ่มต้นคำสั่งสำหรับ java application
<area>	กำหนดพื้นที่การคลิกของ image map
<audioscope>	กำหนดการปรากฏของการแสดงกราฟิกสำหรับเสียงเมื่อมีการ view ด้วย Web TV
	การทำภายในเท็กซ์ของ tags เพื่ออักษร bold
<base>	การระบุ URL ของภาในเพจของ โคลด์ HTML
<basefont>	การกำหนดค่าเริ่มต้น ขนาด สี และฟอนต์ สำหรับทำเพจทั้งหมด
<bgsound>	การเพิ่มเสียงให้กับแบคกราวบนเว็บเพจ โดยการ ใช้ไฟล์ midi หรือ wav
<big>	การทำเท็กซ์ให้มีขนาดใหญ่ระดับหนึ่งมากกว่า base Font
<blink>	การทำเท็กซ์ที่มีอยู่ใน tag กระพริบในระหว่างเวลาปกติ
<blackface>	การทำฟอนต์ขนาดสองเท่า &bold สำหรับผู้ใช้ Web TV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<body>	กำหนดรูปภาพแบคราวนด์ เท็กซ์ link & สีของ link ที่ไปเยี่ยมชม
<bg>	ชุดของเท็กซ์ที่ย่อหน้า สำหรับผู้ที่ใช่ Web TV
 	ทำ line break หลังจากชุดของเท็กซ์หรือกราฟิก
<caption>	สร้างชื่อของตาราง
<center>	จัดกึ่งกลางชุดมูลเท็กซ์และกราฟิก
<cite>	ตัวพิมพ์เข้ม (Emphasized) ในตอนหนึ่งของข้อเขียนของเท็กซ์ในรูปแบบตัวเอียง (italics)
<code>	รูปแบบเท็กซ์ในฟอนต์ monospace
<comment>	วางคำอธิบายใน HTML และซ่อนคำอธิบายจากเพจจริง
<dd>	ย่อหน้าเท็กซ์หนึ่งระดับเมื่อใช้ภายใน <dl> tag
<dfn>	กำหนดเงื่อนไข
<dir>	ทำรายชื่อสั้นๆ ของ items ในหนึ่งแนวตั้ง
<dl>	แสดงรายชื่อของเงื่อนไข & คำจำกัดความ
<dt>	การกำหนด term ในรายชื่อ <dl>
	การทำตัวเข้ม (Emphasized) ตอนหนึ่งของข้อเขียนของเท็กซ์ในตัวเอียง (italics)
<embed>	การฝังไฟล์เสียงหรือ avi movie ลงยัง html
<fn>	การสร้าง anchor point ที่คล้ายกับ <a name> tags
	การเปลี่ยนขนาดสี & ฟอนต์ของเท็กซ์
<form>	กำหนดเพจหรือพื้นที่ของเท็กซ์ไปยัง input form
<frame>	กำหนด frame ภายใน <framest> tag
<framset>	กำหนด attributes ของเพจที่จะมีการใช้ frames
<h1> ... <h6>	แสดงเท็กซ์ในรูปแบบ header ให้มีขนาดใหญ่ขึ้น
<head>	กำหนดส่วนบนของเอกสาร
<hr>	สร้างเส้นในแนวนอน
<html>	การระบุที่เอกสารว่าเขียนใน html
<I>	แสดงเท็กซ์ในลักษณะตัวเอียง (italics)
	แทรกรูปภาพ imagemap หรือแอนิเมชันไปยังเพจ
<input>	กำหนดค่าอินพุทฟิลด์ใน form
<isindex>	สร้าง prompt ไปยังการจัดเก็บความต้องการสำหรับ index
<kbd>	แสดงเท็กซ์ในฟอนต์ monospaced
	สร้าง bullet point หรือบรรทัดใหม่ ในรายชื่อเมื่อใช้ภายใน <dir>, <menu>, <lo> และ tag
<limittext>	กำหนดค่าความกว้างสูงสุดสำหรับเท็กซ์เพื่อทำการแสดงสำหรับการใช้ด้วย Web TV
<link>	สร้างการจัดระเบียบสำหรับชุด navigating ของเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<listing>	ชุด formatted (รูปแบบเดิมๆ) ของเท็กซ์ใน list (รายการ) ที่แสดงในฟอนต์ monospaced
<map>	จำกัดความ client-side map
<marquee>	สร้างเท็กซ์ scroll เมื่อ view ด้วย MSIE
<menu>	การสร้างรายชื่อ
<nobr>	การทำ line break เพื่อหยุดในเท็กซ์หรือชุดของรูปภาพ
<noframes>	การจำกัดการเห็น frame ถ้าการ view นั้นเกิดจากบราวเซอร์ที่ไม่มีความสามารถ
	การกำหนดการเริ่มต้นและจบของ list
<option>	การกำหนด list ของ items ที่ view จาก choose (ตัวเลือก)
<p>	การกำหนดการเริ่มต้นและสิ้นสุดของวรรค ตอน
<plaintext>	แสดงเท็กซ์ในฟอนต์ monospace
<pre>	ชุด Preformats ของเท็กซ์ในฟอนต์ monospace
<s>	การวาด line ผ่านจุดหรือเท็กซ์ถ้ามีการ cross กันภายนอก
<samp>	แสดงเท็กซ์ในฟอนต์ monospace
<script>	การกำหนด ค่าเริ่มต้น และสิ้นสุดของ java script
<select>	แสดง list ของ items ในแต่ละ view จากฟอร์มตัวเลือก (choose)
<server>	เส้นทางต่างๆ ของการทำแอนิเมชันจากเซิร์ฟเวอร์
<small>	แสดงเท็กซ์ขนาดเล็กหนึ่งขนาดนอกจากคำตีพิมพ์เท็กซ์
<strike>	การวาดเส้นผ่านเท็กซ์ถ้ามีการลากตัดกันไปยังภายนอก
	แสดงเท็กซ์ในฟอนต์ boldface (ตัวพิมพ์หนา)
<sub>	แสดงเท็กซ์เหมือน subscript (เล็กและต่ำลง)
<sup>	แสดงเท็กซ์เหมือน subscript (เล็กและโผล่ขึ้น)
<table>	สร้างตาราง
<td>	การกำหนดเซลล์ภายในของตาราง
<textarea>	การระบุขนาดของพื้นที่เท็กซ์

บทที่ 7

การศึกษาและความจำเป็นในการจัดทำระบบใหม่

และ ความต้องการของระบบค้นหาสื่อผ่านอินเทอร์เน็ต

จากหลักการของวิชาซอฟต์แวร์เอ็นจิเนียริง (Software Engineering) การพัฒนาระบบหรือสร้างแอปพลิเคชันขึ้นมาใหม่นั้นต้องทำตามขั้นตอนต่าง ๆ ดังต่อไปนี้

1. วิเคราะห์ปัญหาที่เกิดขึ้นของระบบเดิม
2. หาความจำเป็นในการสร้างระบบใหม่
3. กำหนดความต้องการของระบบใหม่ (Requirement Specification)
4. ออกแบบระบบใหม่

7.1 การวิเคราะห์ระบบการค้นหาสื่อเดิม

7.1.1 ขั้นตอนการค้นหาสื่อ

การค้นหา ผู้ใช้เริ่มด้วยการเลือกชนิดของรายการว่าเป็นรายการของสื่อประเภทใด เช่น วีดีโอ หรือ คาสเซ็ท เทป เป็นต้น จากนั้นผู้ใช้งานต้องทำการค้นหารายชื่อจากรายการสื่อที่เลือกมา ซึ่งผู้ใช้งานต้องทำการค้นหาเองด้วยสายตา ซึ่งการค้นหาดังกล่าวอาจไม่มีประสิทธิภาพดีพอนักเมื่อเทียบกับการค้นหาด้วยระบบคอมพิวเตอร์ กล่าวคือการค้นหาด้วยสายตาต้องไล่หาชื่อสื่อที่ต้องการ โดยการค้นหาตั้งแต่ชื่อแรกจนกว่าจะเจอชื่อสื่อที่ต้องการ ทำให้เสียเวลา และบางครั้งการค้นหาที่ใช้เวลานานๆ เมื่อค้นหาแล้วไม่เจอ อาจทำให้ผู้ใช้รู้สึกไม่พอใจได้

7.1.2 ปัญหาที่เกิดขึ้น

จากการวิเคราะห์ระบบห้องสมุดทัศนศึกษา พบว่ามีปัญหาดังต่อไปนี้

1. ในการให้ผู้ใช้เลือกสื่อต้องการไม่มีตัวอย่างที่เป็นเสียง ภาพ ภาพเคลื่อนไหว ซึ่งเป็นสิ่งที่ผู้ใช้จะได้จากสื่อจริง ๆ จึงเป็นผลให้ผู้ใช้ได้สื่อไม่ตรงตามความต้องการ
2. การค้นหาของผู้ใช้โดยสายตาอาจไม่มีประสิทธิภาพและไม่สะดวกเพียงพอ เมื่อเทียบกับระบบคอมพิวเตอร์
3. การค้นหาสื่อจำกัดแค่ในห้องทัศนศึกษาเท่านั้น ทำให้ไม่สะดวกต่อผู้ใช้เมื่อต้องการค้นหาสื่อ เพราะไม่สามารถค้นหาสื่อจากภายนอกได้
4. ผู้ดูแลระบบต้องทำการเพิ่มข้อมูลด้วยตัวเองทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 ความจำเป็นในการจัดทำระบบใหม่

เพื่อเพิ่มความสะดวกและรวดเร็วในการค้นหาสื่อที่ต้องการ อีกทั้งยังไม่จำเป็นต้องเดินทางมายังที่ห้องสมุดทัศนศึกษาโดยตรง ผู้ใช้ก็สามารถทำการค้นหา รายชื่อสื่อต่าง ๆ ที่สนใจ ผ่านทางอินเทอร์เน็ต ซึ่งระบบพัฒนาขึ้นสามารถนำไปใช้งานได้ทั้งภายในห้องสมุดทัศนศึกษาและภายนอก โดยทำการติดตั้งระบบลงบนอินเทอร์เน็ตเซิร์ฟเวอร์ เพื่อให้ผู้ใช้จากภายนอกสามารถเข้ามาค้นหา ตรวจสอบ ข้อมูลได้ และระบบใหม่นี้ยังช่วยอำนวยความสะดวกให้แก่ผู้ดูแลระบบ โดยที่ผู้ดูแลระบบสามารถที่จะแก้ไข เปลี่ยนแปลง หรือเพิ่มเติมข้อมูล ได้โดยผ่านทางอินเทอร์เน็ตได้เลยโดยตรง

7.3 ความต้องการของระบบใหม่

จากการวิเคราะห์ระบบห้องสมุดทัศนศึกษา พบปัญหาดังที่กล่าวมาแล้ว จึงกำหนดเป็นความต้องการ ของระบบใหม่ ซึ่งควรมีคุณสมบัติดังนี้

1. ผู้ใช้สามารถค้นหาสื่อได้

1.1) การค้นหาสามารถทำได้ดังนี้

- 1.1.1 ระบุชื่อหรือส่วนหนึ่งของชื่อสื่อเพื่อค้นหาได้
- 1.1.2 ระบุชื่อหรือส่วนหนึ่งของชื่อผู้แต่งหรือผู้ผลิตเพื่อค้นหาได้
- 1.1.3 สามารถระบุประเภทของเรื่องสำหรับค้นหาได้ เช่น สารคดี บันเทิง การศึกษา
- 1.1.4 สามารถระบุชนิดของสื่อสำหรับค้นหาได้ เช่น วีดีโอ เทป ซีดีรอม
- 1.1.5 สามารถระบุรายละเอียดตั้งแต่ข้อ 1.1.1-1.1.4 บางส่วน หรือทั้งหมดเพื่อการค้นหาได้ หมายเหตุ ประเภทของเรื่องและชนิดของสื่อสามารถแก้ไขได้ตามต้องการ

1.2) ผลที่ได้จากการค้นหาจะเป็นรายชื่อเรื่องของสื่อ ผู้ผลิต ชนิด และประเภท เรียงตามลำดับดังต่อไปนี้

- 1.2.1 ตามลำดับชื่อเรื่อง
- 1.2.2 ตามลำดับผู้แต่ง / ผู้ผลิต

2. ผู้ใช้สามารถอ่านรายละเอียดของสื่อผ่านทางระบบอินเทอร์เน็ตได้โดย

2.1) เลือกสื่อได้จาก

- เลือกสื่อจากรายชื่อเรื่องทั้งหมดที่มี
- เลือกสื่อจากการค้นหาสื่อตามข้อ 1.2

2.2) รายละเอียดของสื่อที่ต้องแสดงมีดังนี้

- ชื่อเรื่อง
- ผู้แต่ง / ผู้ผลิต
- เนื้อเรื่องย่อหรือคำอธิบายอย่างละเอียด
- ความยาว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ประเภทของเรื่อง

- ชนิดของสื่อ

3. สามารถใช้บริการค้นหาและหารายละเอียดของสื่อในห้องโสตทัศนศึกษาได้เช่นเดียวกับใช้บนอินเทอร์เน็ต

4. ระบบสามารถให้เจ้าหน้าที่เพิ่มเติม เปลี่ยนแปลง หรือลบข้อมูลเกี่ยวกับสื่อผ่านทางระบบ ที่พัฒนาผ่านทางอินเทอร์เน็ต ได้

4.1) ข้อมูลเกี่ยวกับสื่อมีดังนี้

4.1.1. หมายเลขประจำสื่อ

4.1.2. หมายเลขประจำเรื่อง

4.1.3. ชื่อเรื่อง

4.1.4. ผู้แต่ง / ผู้ผลิต

4.1.5. ประเภทของเรื่อง

4.1.6. ชนิดของสื่อ



บทที่ 8

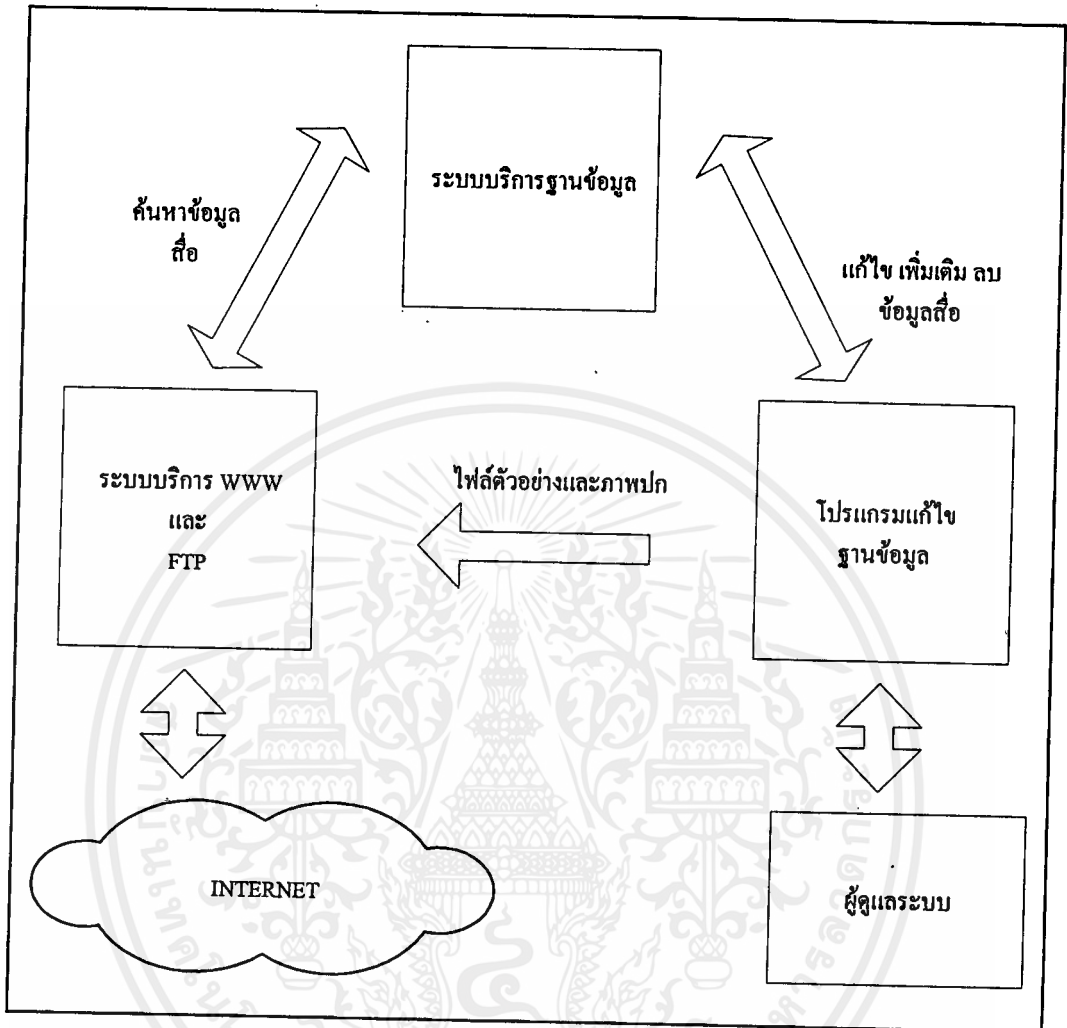
การออกแบบระบบใหม่

8.1 การออกแบบระบบใหม่

จากการวิเคราะห์ระบบเดิม และหาความต้องการระบบใหม่ จึงได้ออกแบบระบบคอมพิวเตอร์เพื่อรองรับการให้บริการใหม่ดังนี้

ระบบนี้ประกอบด้วยระบบใหญ่ๆ 3 ระบบคือ

1. ระบบบริการฐานข้อมูล ทำหน้าที่ดูแลฐานข้อมูลของสื่อ และค้นหาข้อมูลของสื่อที่เก็บไว้
2. ระบบบริการเวิร์คไวด์เว็บ และเอฟทีพี จะเป็นระบบที่ติดต่อระหว่างผู้ใช้ทางอินเทอร์เน็ตกับระบบบริการฐานข้อมูลของสื่อ โดยจะมีโปรแกรมซีจีไอ เพื่อติดต่อกับระบบบริการฐานข้อมูล และเป็นระบบที่เก็บไฟล์ตัวอย่าง และไฟล์ปกของสื่อด้วย
3. โปรแกรมแก้ไขฐานข้อมูล จะติดต่อกับระบบบริการฐานข้อมูลเพื่อ แก้ไข เพิ่มเติม หรือลบข้อมูลเกี่ยวกับสื่อ



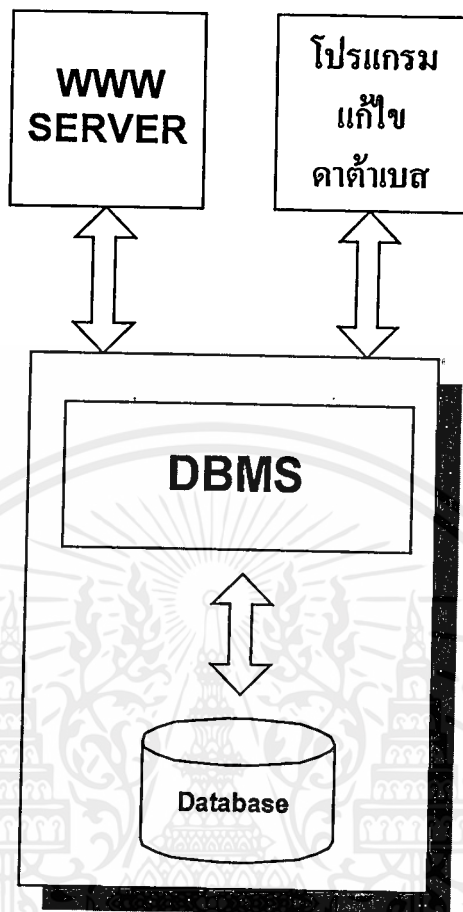
รูปที่ 8-1 ความสัมพันธ์ของระบบทั้งหมด

8.2 การออกแบบและการประยุกต์ใช้งานระบบคอมพิวเตอร์

จากหัวข้อที่แล้วได้กล่าวถึงระบบที่ทำการออกแบบไว้แล้ว ในหัวข้อนี้จะกล่าวถึงระบบคอมพิวเตอร์ที่สร้างขึ้น

8.2.1 ระบบบริการฐานข้อมูล

ระบบนี้เป็นระบบที่ทำหน้าที่ให้บริการการจัดเก็บข้อมูลและค้นหาข้อมูลให้แก่ระบบบริการเวิร์ลด์ไวด์เว็บ และ โปรแกรมแก้ไขฐานข้อมูล โดยระบบนี้ใช้ไมโครซอฟต์เอสคิวแอลเซิร์ฟเวอร์เป็นที่เก็บฐานข้อมูลทั้งหมด



รูปที่ 8-2 ระบบบริการฐานข้อมูล

ประกอบด้วย 2 ส่วนคือ

1. ดาต้าเบส เมนเนจเม้นท์ : Database Management System(DBMS)

เป็นตัวจัดการการกระทำทั้งหมดที่เกี่ยวข้องกับฐานข้อมูล โดยด้านหนึ่งจะรับคำสั่งเป็นภาษา SQL จากระบบอื่น แล้วนำมากระทำกับดาตาเบสโดยมีการกระทำดังนี้

- ค้นหาข้อมูล
- เพิ่มเติมข้อมูล
- แก้ไขเปลี่ยนแปลงข้อมูล
- ลบข้อมูล

แล้วจะมีการตอบกลับผลลัพธ์การกระทำไปให้แก่ระบบที่ส่ง SQL มา

2. ดาต้าเบส(Database) เป็น ฟิสิคอลลดาตาของข้อมูลที่ DBMS จัดการ

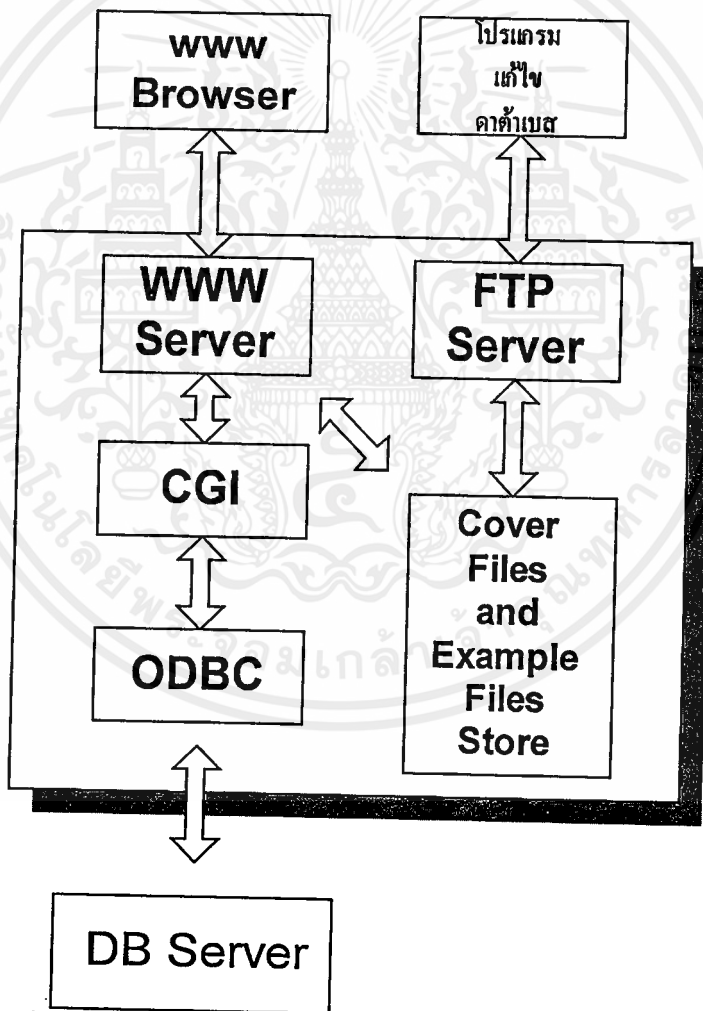
8.2.2. ระบบบริการเวิร์ดไวด์เว็บและเอฟทีพี

เป็นระบบที่ติดต่อกันระหว่างผู้ใช้ทางอินเทอร์เน็ตกับระบบบริการฐานข้อมูลของสื่อ โดยจะมีโปรแกรมซีจีไอ เพื่อติดต่อกับระบบบริการฐานข้อมูลด้วยโอดีบีซี และเป็นที่เก็บไฟล์ตัวอย่าง และไฟล์ปกของสื่อด้วย โดยระบบนี้ใช้อินเทอร์เน็ต อินฟอรมะชัน เซิร์ฟเวอร์ ทำหน้าที่ให้บริการเวิร์ดไวด์เว็บ และ เอฟทีพี

ประกอบด้วย 2 ส่วนคือ

1. ระบบบริการเวิร์ดไวด์เว็บ ทำหน้าที่เป็นตัวติดต่อกันระหว่างผู้ใช้กับระบบบริการฐานข้อมูล ทำการเปลี่ยนรูปแบบของข้อมูลเพื่อให้สวยงามและง่ายในการใช้งาน โดยในส่วนนี้อาศัยหลักการของไอเอสเอพีไอ สำหรับการสร้างเว็บเพจเพื่อติดต่อกับผู้ใช้

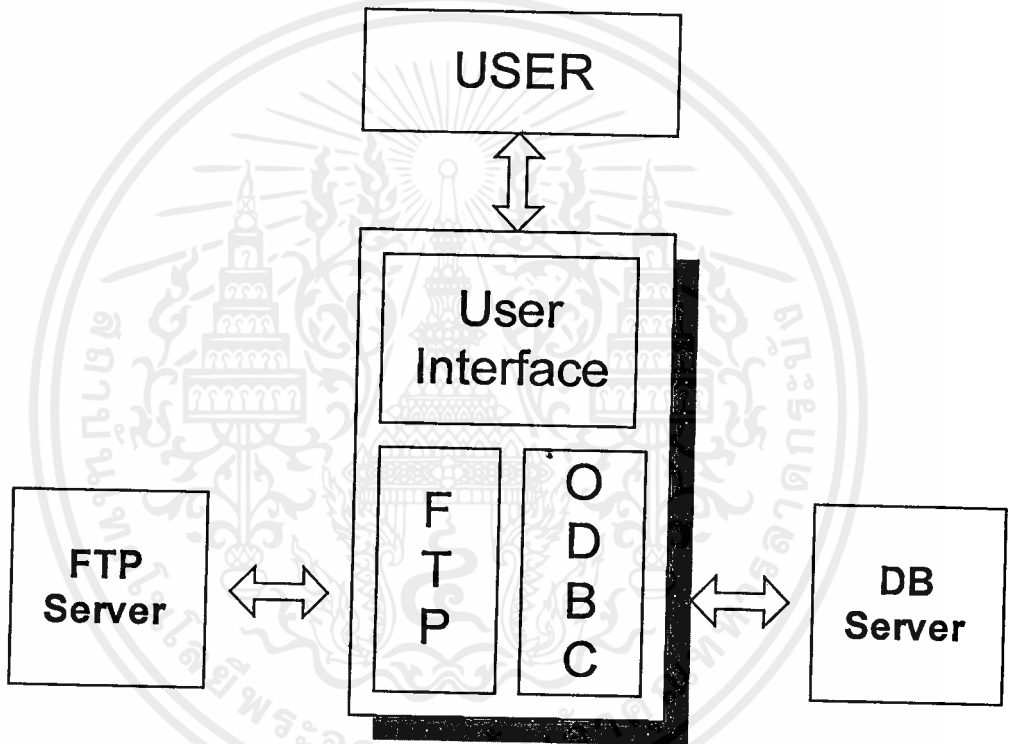
2. ระบบบริการเอฟทีพี ระบบนี้จะรับไฟล์ตัวอย่างและไฟล์ปกจากโปรแกรมแก้ไขค่าเบส มาเก็บไว้ในที่เก็บ เพื่อให้บริการเวิร์ดไวด์เว็บทำการแสดงต่อไป



รูปที่ 8-3 ระบบบริการเวิร์ดไวด์เว็บ และเอฟทีพี

8.2.3. โปรแกรมแก้ไขฐานข้อมูล

จะติดต่อกับระบบบริการฐานข้อมูลเพื่อ แก้ไข เพิ่มเติม หรือลบข้อมูลเกี่ยวกับสื่อ โดยผู้ดูแลระบบสามารถทำการเพิ่มแก้ไข ฐานข้อมูลได้โดย รายละเอียดจะกล่าวในบทถัดไปเรื่องการนำไปใช้งาน ซึ่งโปรแกรมแก้ไขฐานข้อมูลนี้พัฒนาด้วยไมโครซอฟต์วิซวลซีพลัสพลัส โดยอาศัยหลักการของโอดีบีซี เป็นไคร์เวอร์สำหรับการส่งผ่านค่าของข้อมูลต่างลงไปเก็บยังระบบบริการฐานข้อมูล



รูปที่ 8-4 โปรแกรมแก้ไขฐานข้อมูล

8.3 การออกแบบฐานข้อมูล

ฐานข้อมูลของระบบนี้มี 3 ตารางดังนี้

1. ตารางเรื่อง เป็นตารางที่เก็บข้อมูลเกี่ยวกับสื่อ มีคาดำมีคาดำเบส สก็มาดังนี้

หัวข้อเรื่อง	ชื่อเรื่อง	ชื่อผู้แต่ง	ความยาว	ชื่อไฟล์ ตัวอย่าง	ชื่อ ไฟล์ ภาพปก	หมายเลข ประเภทสื่อ	หมายเลข ประเภทเรื่อง	คำอธิบาย
--------------	------------	-------------	---------	----------------------	--------------------	-----------------------	-------------------------	----------

2. ตารางชนิดของสื่อ เป็นตารางที่เก็บประเภทสื่อ เช่น วิดีโอเทป ซีดีรอม

หมายเลขประเภทสื่อ	ชื่อประเภทสื่อ
-------------------	----------------

3. ตารางประเภทของเรื่อง เป็นตารางที่เก็บประเภทเรื่องเช่น สารคดี ยันเทิง

หมายเลขประเภทเรื่อง	ชื่อประเภทเรื่อง
---------------------	------------------

บทที่ 9

การนำไปใช้งาน

9.1 แนะนำและติดตั้งระบบบริการฐานข้อมูล

ในระบบประกอบไปด้วยไฟล์ที่จำเป็นต้องใช้ดังต่อไปนี้

1. WWWQuery.dll เป็นโปรแกรมซีจีไอของระบบให้บริการเวิร์ดไวด์เว็บ เป็นโปรแกรมที่ติดต่อกับ ผู้ใช้บริการเพื่อค้นหาข้อมูลสื่อ การติดตั้งอยู่ในระบบบริการเวิร์ดไวด์เว็บ
2. MIsAdmin.exe เป็นโปรแกรมที่ใช้แก้ไขฐานข้อมูลเกี่ยวกับสื่อ และส่งไฟล์ตัวอย่าง ไฟล์ภาพปก ไปเก็บยังระบบให้บริการเวิร์ดไวด์เว็บ
3. DBMSSOCN.dll เป็นไลบรารีของโอคิปซี

ระบบนี้จะใช้โปรแกรมให้บริการฐานข้อมูลของไมโครซอฟต์ที่ชื่อว่า เอสคิวแอลเซิร์ฟเวอร์ โดยเวอร์ชันที่ใช้คือ 6.5 หรือ อาจใช้โปรแกรมจัดการฐานข้อมูลอื่นก็ได้โดยมีคุณสมบัติดังนี้

1. สามารถเข้าถึงฐานข้อมูลด้วยโอคิปซีได้
2. มีโอคิปซีไครเวอร์บนวินโดวส์เก้าห้า

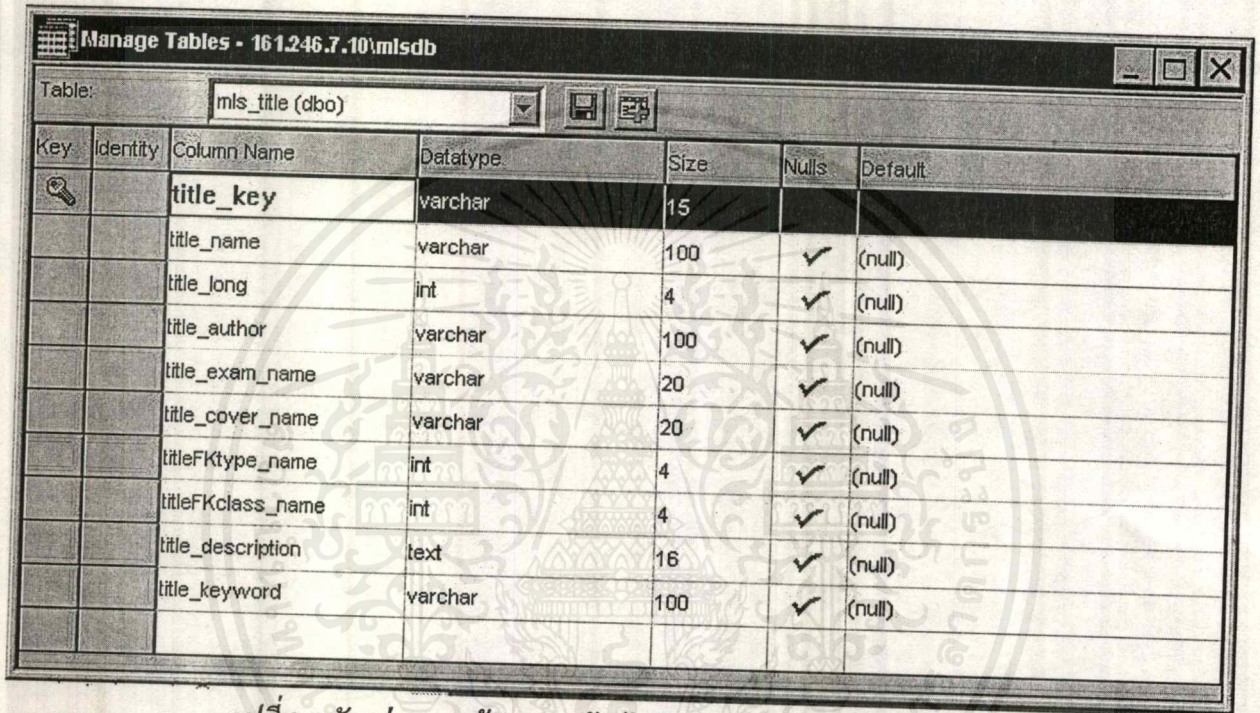
การติดตั้งโปรแกรมให้ศึกษาจากคู่มือการติดตั้ง โปรแกรมต่างๆ การติดตั้งนี้ต้องการผู้ที่มีความรู้ในด้านฐานข้อมูลเป็นอย่างดี

เมื่อติดตั้งเสร็จให้สร้างฐานข้อมูลที่ชื่อว่า mlsdb โดยประกอบด้วยตารางดังต่อไปนี้

1. ตาราง mls_title มีรายละเอียดของแต่ละฟิลด์ดังนี้
 - 1) title_key มีชนิดของข้อมูลเป็นตัวอักษรขนาด 15 ตัวอักษร สำหรับเก็บรหัสของสื่อ
 - 2) title_name มีชนิดของข้อมูลเป็นตัวอักษรขนาด 100 ตัวอักษร สำหรับเก็บชื่อสื่อ
 - 3) title_long มีชนิดของข้อมูลเป็นเลขจำนวนเต็มขนาด 4 ไบท์ สำหรับเก็บความยาวสื่อ
 - 4) title_author มีชนิดของข้อมูลเป็นตัวอักษรขนาด 100 ตัวอักษร สำหรับเก็บชื่อผู้แต่งผู้ผลิต
 - 5) title_keyword มีชนิดของข้อมูลเป็นตัวอักษรขนาด 100 ตัวอักษร สำหรับเก็บคีย์คำของสื่อ
นั้นๆ.
 - 6) title_exam_name มีชนิดของข้อมูลเป็นตัวอักษรขนาด 20 ตัวอักษร ใช้สำหรับเก็บชื่อไฟล์ตัวอย่างของสื่อ
 - 7) title_cover_name มีชนิดของข้อมูลเป็นตัวอักษรขนาด 20 ตัวอักษร ใช้สำหรับเก็บชื่อไฟล์ที่ภาพปกของสื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

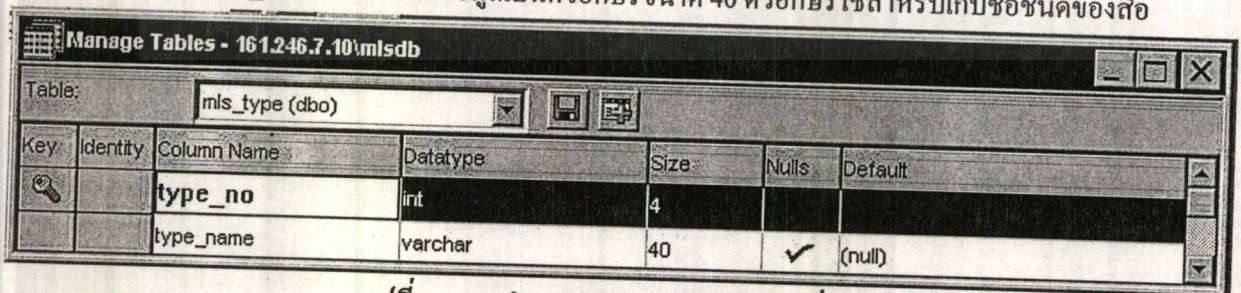
- 8) titleFKtype_name มีชนิดของข้อมูลเป็นเลขจำนวนเต็มขนาด 4 ไบท์ ใช้สำหรับเก็บเลขรหัสชนิดของสื่อ
 - 9) titleFKclass_name มีชนิดของข้อมูลเป็นเลขจำนวนเต็มขนาด 4 ไบท์ ใช้สำหรับเก็บเลขรหัสประเภทของเรื่อง
 - 10) title_description มีชนิดของข้อมูลเป็นตัวอักษรขนาด 300 ตัวอักษรใช้สำหรับเก็บคำบรรยายของสื่อ
- ตารางนี้ให้ title_key เป็นไพรมารีคีย์ (Primary Key) ถ้าใช้โปรแกรมไมโครซอฟต์เอสคิวแอล



Key	Identity	Column Name	Datatype	Size	Nulls	Default
Key		title_key	varchar	15		
		title_name	varchar	100	✓	(null)
		title_long	int	4	✓	(null)
		title_author	varchar	100	✓	(null)
		title_exam_name	varchar	20	✓	(null)
		title_cover_name	varchar	20	✓	(null)
		titleFKtype_name	int	4	✓	(null)
		titleFKclass_name	int	4	✓	(null)
		title_description	text	16	✓	(null)
		title_keyword	varchar	100	✓	(null)

รูปที่ 9-1 ตัวอย่างการสร้างตารางด้วยไมโครซอฟต์เอสคิวแอลเซิร์ฟเวอร์

2. ตาราง mls_type มีรายละเอียดของแต่ละฟิลด์ดังนี้
 - 1) type_no มีชนิดของข้อมูลเป็นเลขจำนวนเต็มขนาด 4 ไบท์ ใช้สำหรับเก็บเลขรหัสชนิดของสื่อ
 - 2) type_name มีชนิดของข้อมูลเป็นตัวอักษรขนาด 40 ตัวอักษรใช้สำหรับเก็บชื่อชนิดของสื่อ



Key	Identity	Column Name	Datatype	Size	Nulls	Default
Key		type_no	int	4		
		type_name	varchar	40	✓	(null)

รูปที่ 9-2 การกำหนด Type_no เป็นไพรมารีคีย์

3. ตาราง mls_class มีรายละเอียดของแต่ละฟิลด์ดังนี้

- 1) class_no มีชนิดของข้อมูลเป็นเลขจำนวนเต็มขนาด 4 ไบต์ ใช้สำหรับเก็บเลขรหัสประเภทเรื่อง
- 2) class_name มีชนิดของข้อมูลเป็นตัวอักษรขนาด 40 ตัวอักษร ใช้เก็บชื่อประเภทเรื่อง

Key	Identity	Column Name	Datatype	Size	Nulls	Default
		class_no	int	4		
		class_name	varchar	40	✓	(null)

รูปที่ 9-3 กำหนดให้ class_no เป็น primary key

และ สร้างวิว ที่ชื่อว่า mls_title_view จากคำสั่งภาษา SQL ดังนี้

```
CREATE VIEW mls_title_view AS
SELECT title_key,title_name,title_long,title_author,title_keyword,
       title_exam_name,title_cover_name,type_name,class_name,
       title_description
FROM mls_title,mls_type,mls_class
WHERE type_no=titleFKtype_name,class_no=titleFKclass_name
```

จากคำสั่งนี้จะ ได้วิวที่มีรายละเอียดฟิลด์ดังนี้

1. title_key มีชนิดของข้อมูลเป็นตัวอักษรขนาด 15 ตัวอักษร สำหรับเก็บรหัสของสื่อ
2. title_name มีชนิดของข้อมูลเป็นตัวอักษรขนาด 100 ตัวอักษร สำหรับเก็บชื่อสื่อ
3. title_long มีชนิดของข้อมูลเป็นเลขจำนวนเต็มขนาด 4 ไบต์ สำหรับเก็บความยาวสื่อ
4. title_author มีชนิดของข้อมูลเป็นตัวอักษรขนาด 100 ตัวอักษร สำหรับเก็บชื่อผู้แต่งผู้ผลิต
5. title_keyword มีชนิดของข้อมูลเป็นตัวอักษรขนาด 100 ตัวอักษร สำหรับเก็บกุญแจคำของสื่อ นั้นๆ
6. title_exam_name มีชนิดของข้อมูลเป็นตัวอักษรขนาด 20 ตัวอักษร ใช้สำหรับเก็บชื่อไฟล์ตัวอย่างของสื่อ
7. title_cover_name มีชนิดของข้อมูลเป็นตัวอักษรขนาด 20 ตัวอักษร ใช้สำหรับเก็บชื่อไฟล์ที่ภาพปกของสื่อ

8. type_name มีชนิดของข้อมูลเป็นตัวอักษรขนาด 40 ตัวอักษรใช้สำหรับเก็บชื่อชนิดของสื่อ
9. class_name มีชนิดของข้อมูลเป็นตัวอักษรขนาด 40 ตัวอักษรใช้เก็บชื่อประเภทเรื่อง
10. title_description มีชนิดของข้อมูลเป็นตัวอักษรขนาด 300 ตัวอักษรใช้สำหรับเก็บคำบรรยายของสื่อ

ในทุกตาราง และทุกวิว ทุกฟิลด์ ต้องตั้งชื่อให้ตรงกับที่กำหนดมาด้วย มิฉะนั้น โปรแกรมค้นหา จะไม่สามารถใช้งานได้

ต่อมาให้ทำการเพิ่มชื่อผู้ใช้ สำหรับเข้าถึงฐานข้อมูล mlsdb โดยให้ตั้งชื่อว่า mlsadmin และมีรหัสผ่านว่า admin โดยต้องให้สิทธิในการ อ่าน เขียนข้อมูลในฐานข้อมูล mlsdb ทุกอย่าง

9.2 แนะนำและติดตั้งระบบบริการเวิร์ดไวด์เว็บและเอฟทีพี

ระบบบริการเวิร์ดไวด์เว็บและเอฟทีพี ที่ใช้สำหรับระบบนี้ต้องเป็นไมโครซอฟต์อินเทอร์เน็ตฟอรัมเซิร์ฟเวอร์เท่านั้น เพราะในส่วนโปรแกรมค้นหาเขียนสำหรับใช้กับอินเทอร์เน็ตฟอรัมเซิร์ฟเวอร์ไม่สามารถใช้กับโปรแกรมให้บริการด้านเว็บเซิร์ฟเวอร์ตัวอื่นได้

ในระบบบริการเวิร์ดไวด์เว็บและเอฟทีพีนี้เราต้องสร้าง ไคลเรกทอรีเพื่อใช้เก็บโปรแกรมและตัวอย่างของสื่อ โดยต้องสร้างไคลเรกทอรีชื่อ mls และสร้างไคลเรกทอรีย่อยของ mls ดังนี้

- 1) example เป็นที่เก็บตัวอย่างของสื่อ
- 2) cover เป็นที่เก็บภาพปกของสื่อ
- 3) mls-cgi เป็นที่เก็บโปรแกรม WWWQuery.dll ที่เป็นตัวค้นหาสื่อ
- 4) image เป็นที่เก็บภาพต่างๆที่ใช้สำหรับตกแต่ง

ตามตัวอย่างนี้จะสร้างไคลเรกทอรีชื่อ e:\mls และไคลเรกทอรีย่อยดังกล่าว ต่อมาเราต้องทำการสร้าง ไคลเรกทอรีเสมือน(Virtual Directory) ของบริการเวิร์ดไวด์เว็บและเอฟทีพี เมื่อสร้างไคลเรกทอรีของทั้งสองบริการเรียบร้อยแล้ว เมื่อเราอ้างอิงถึง <http://www.host.com/mls> หรือ <ftp://www.host.com/mls> (เมื่อสมมติให้ www.host.com เป็นชื่อของเวิร์ดไวด์เว็บเซิร์ฟเวอร์และเอฟทีพี) ก็จะหมายถึง e:\mls หรือไคลเรกทอรีใดๆ ที่เราสร้างขึ้น

9.2.1 การติดตั้ง WWWQuery.dll

1. ให้ทำการคัดลอกไฟล์ WWWQuery.dll ลงไปใน e:\mls\mls-cgi เพื่อติดตั้งโปรแกรมค้นหาสื่อผ่านทางอินเทอร์เน็ต
2. ให้ติดตั้งไอดีบีซีและเพิ่มเติม ดีเอสเอ็น : DSN (Data Source Name) เช่นเดียวกับการติดตั้งโปรแกรมแก้ไขฐานข้อมูล ในส่วนนี้ให้อ้างอิงกับหัวข้อการติดตั้งไอดีบีซี และดีเอสเอ็น ของการติดตั้งโปรแกรมแก้ไขฐานข้อมูล

หลังจากทำการติดตั้งเสร็จเรียบร้อยแล้วต้องทำการติดตั้งให้ทดสอบ โดยใช้โปรแกรมเว็บเบราว์เซอร์เช่น Netscape เมื่ออ้างอิงถึง <http://www.host.com/mls/mls-cgi/WWWQuery.dll>? ก็จะพบหน้าจอหลักของบริการค้นหาข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.3 แนะนำและติดตั้งโปรแกรมแก้ไขฐานข้อมูล

ไฟล์ที่เกี่ยวข้องกับโปรแกรมนี้คือ

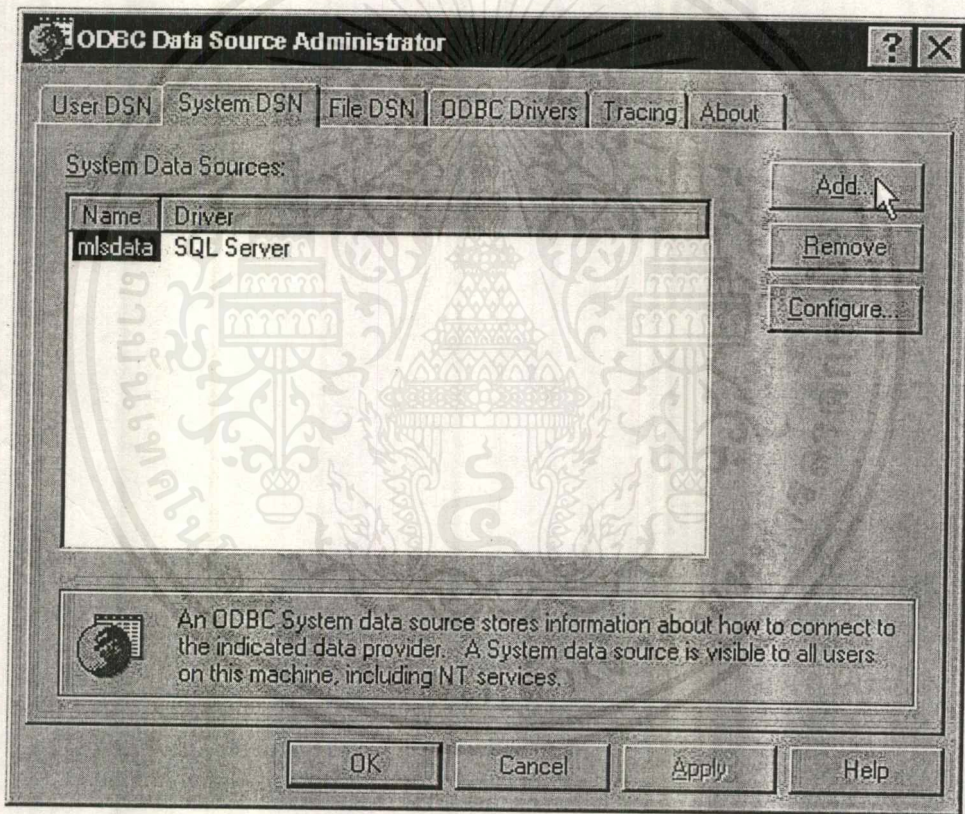
- 1) MIsAdmin.exe เป็น โปรแกรมแก้ไขฐานข้อมูล
- 2) DBMSSOCN.dll เป็น Network Library ของ ODBC

ให้คัดลอกไฟล์ทั้งสองลงบนฮาร์ดดิสก์ในไดเรกทอรี c:\mls หรือไดเรกทอรีใดก็ได้ตามต้องการ

9.3.1 การติดตั้งโอดีบีซี และดีเอสเอ็น

โปรแกรมนี้ติดต่อกับระบบบริการฐานข้อมูลโดยใช้ โอเพนคาต้าคอนเนคทีวิตี ฉะนั้นจึงต้องติดตั้งโอดีบีซี ก่อน ให้ศึกษาการติดตั้งจากคู่มือของโปรแกรม หลังจากติดตั้งเสร็จแล้วเราต้องทำการเพิ่ม ดีเอสเอ็น (Data Source Name) โดยให้ชื่อว่า mlsdata วิธีการมีดังนี้

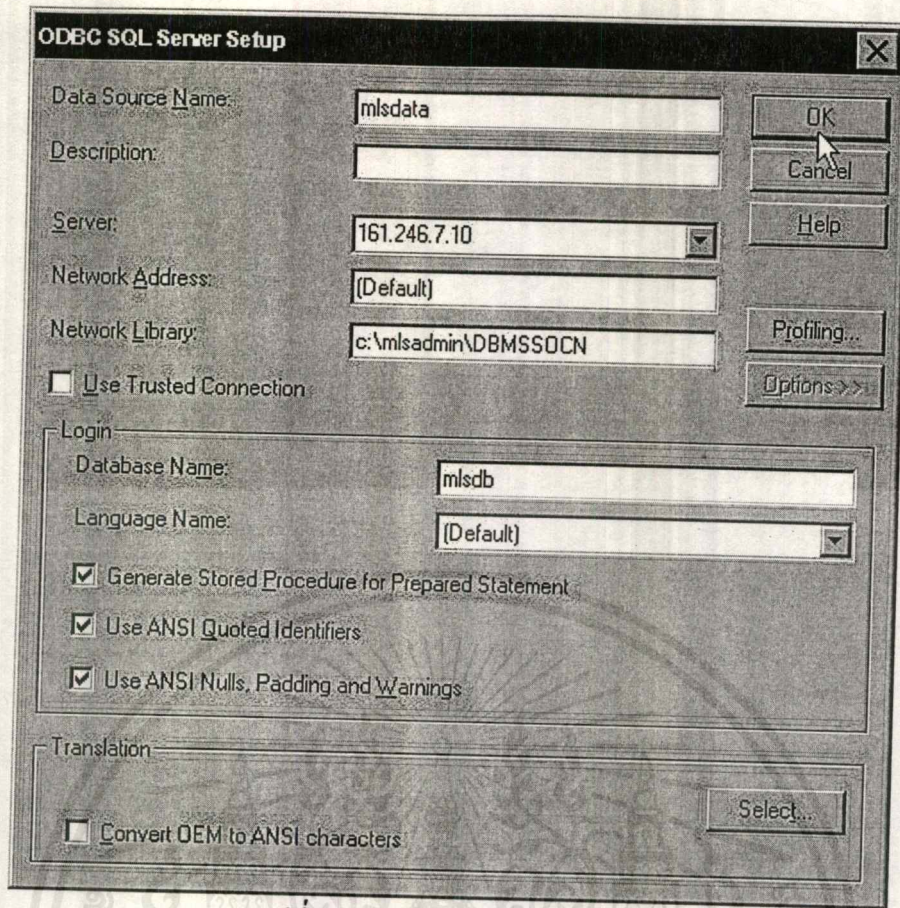
1. เรียกโปรแกรม ODBC Administrator จาก Control Panel



รูปที่ 9-4 ODBC Administrator

2. เลือก System DSN แล้วกดปุ่ม ADD เพื่อเพิ่มเติมรายชื่อ
3. เลือก ODBC driver ตาม โปรแกรมระบบบริการฐานข้อมูลที่ใช้ ในที่นี้เลือกใช้ SQL Server แล้วกดปุ่ม Finish

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

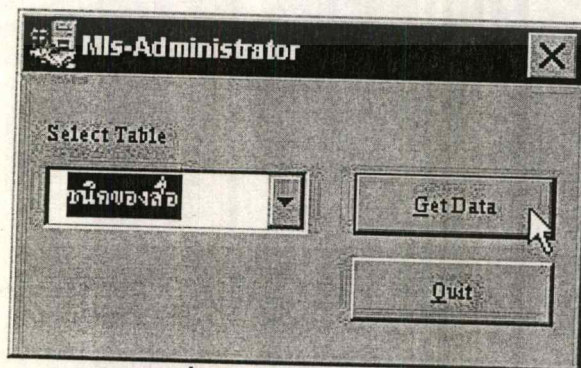


รูปที่ 9-5 ตัวอย่างการสร้างดาต้าซอร์ส

4. ให้กรอก OPTION เพื่อขยายหน้า หลังจากนั้นให้ใส่ข้อมูลดังต่อไปนี้

- 1) ช่อง Data Source Name ให้ใส่ mldata
- 2) ช่อง Server ให้ใส่ชื่อของระบบบริการฐานข้อมูล
- 3) ช่อง Network Library ให้ใส่ที่อยู่ของไฟล์ DBMSSOCN.dll ในที่นี้คือ c:\mldata\DBMSSOCN.dll
- 4) ช่อง Database name ให้ใส่ mldb

เมื่อทำการติดตั้ง ODBC เรียบร้อยแล้วให้ทดลองเรียก โปรแกรม MLSAdmin.exe ถ้าไม่มีปัญหาใดๆเกิดขึ้นจะได้หน้าต่างดังรูป



รูปที่ 9-6 โปรแกรมฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.3.2.การใช้งานโปรแกรมแก้ไขฐานข้อมูล

โปรแกรมนี้เป็นโปรแกรมแก้ไขฐานข้อมูลของสื่อในระบบฐานข้อมูล ซึ่งมีทั้งหมด 3 ตารางคือ

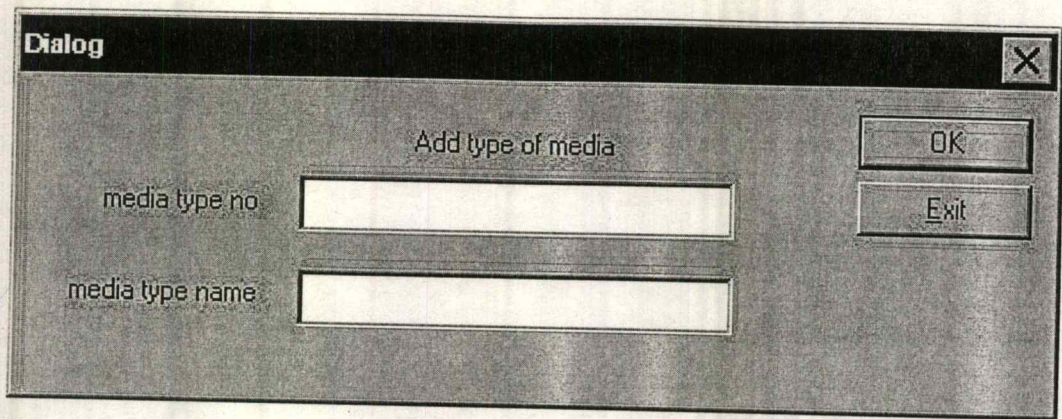
- 1) ตารางชนิดของสื่อ ประกอบด้วย 2 필ด์คือ
 1. หมายเลขชนิดของสื่อ
 2. ชื่อชนิดของสื่อ เช่น วีดีโอเทป คาสเซ็ทเทป
- 2) ตารางประเภทของเรื่อง ประกอบด้วย 2 필ด์คือ
 1. หมายเลขประเภทของเรื่อง
 2. ชื่อชนิดของเรื่อง เช่น บ้านเทิง สารคดี
- 3) ตารางเรื่อง ประกอบด้วย 10 필ด์คือ
 1. หมายเลขเรื่อง
 2. ชื่อเรื่อง
 3. ชื่อผู้แต่ง/ผู้ผลิต
 4. ความยาว
 5. คีย์เวิร์ด
 6. ไฟล์ตัวอย่าง
 7. ไฟล์ภาพปก
 8. ชนิดของสื่อ
 9. ประเภทของเรื่อง
 10. คำอธิบาย

เราสามารถเลือกที่จะดูข้อมูลและแก้ไขแต่ละตารางจากหน้าจอหลักของโปรแกรม โดยเลือกจากรายชื่อแล้วกดปุ่ม GetData จะได้นหน้าต่าง View data ในการดูข้อมูลนั้นจะดูได้ชุดละ 25 บรรทัด ในการดูข้อมูลชุดอื่นนั้นหน้าต่างจะมี

- 1) ปุ่ม First เพื่อดูข้อมูล 25 ชุดแรก
- 2) ปุ่ม Last เพื่อดูข้อมูล 25 ชุดสุดท้าย
- 3) ปุ่ม Next เพื่อดูข้อมูล 25 ชุดต่อไป
- 4) ปุ่ม Prev เพื่อใช้ดูข้อมูล 25 ชุดก่อนหน้า

9.3.3 การเพิ่มเติมข้อมูล

เมื่อกดปุ่ม Add จะปรากฏหน้าต่างให้เพิ่มเติมข้อมูล หน้าต่างที่ปรากฏจะขึ้นอยู่กับตาราง



รูปที่ 9-7 การเพิ่มเติมข้อมูลตารางชนิดของสื่อ

เมื่อเติมข้อมูลตามต้องการแล้ว ให้กดปุ่ม OK ข้อมูลก็จะถูกส่งไปบันทึกบนระบบบริการฐานข้อมูล

9.3.4 การแก้ไขข้อมูลที่บันทึกไว้เรียบร้อยแล้ว

เมื่อต้องการแก้ไขข้อมูล ให้ใช้เมาส์พอยน์เตอร์เลือกคอลัมน์แรกของข้อมูลที่ต้องการในหน้าต่าง View Data แล้วกดปุ่ม Modify ก็จะปรากฏหน้าต่างดังรูปที่ 14 ซึ่งจะมีข้อมูลเดิมเติมอยู่ในช่องต่างๆ ให้ทำการแก้ไขแล้วกดปุ่ม OK เพื่อบันทึกการแก้ไข

9.3.4.1 การลบข้อมูลที่ไม่ต้องการ

ให้นำเมาส์พอยน์เตอร์ไปเลือกคอลัมน์แรกของข้อมูลที่ต้องการแล้ว แล้วกดปุ่ม Delete โปรแกรมจะถามว่าต้องการลบจริงหรือไม่ ถ้าต้องการลบจริงๆ ก็ตอบ OK

9.3.4.2 การค้นหาสื่อที่ต้องการ

สำหรับในหน้าต่าง View Data ของตารางเรื่อง เราสามารถกำหนดให้แสดงข้อมูลสื่อตามต้องการได้ เช่น ข้อมูลสื่อที่เป็นวีดีโอเทปเท่านั้น โดยกดปุ่ม Query จะปรากฏหน้าต่างเหมือนกับการ Add และ Modify ให้ใส่ข้อมูลที่ต้องการลงในช่องต่าง

Title [X]

Query New

Title Key

Title Name

Author Name

Long

Keyword

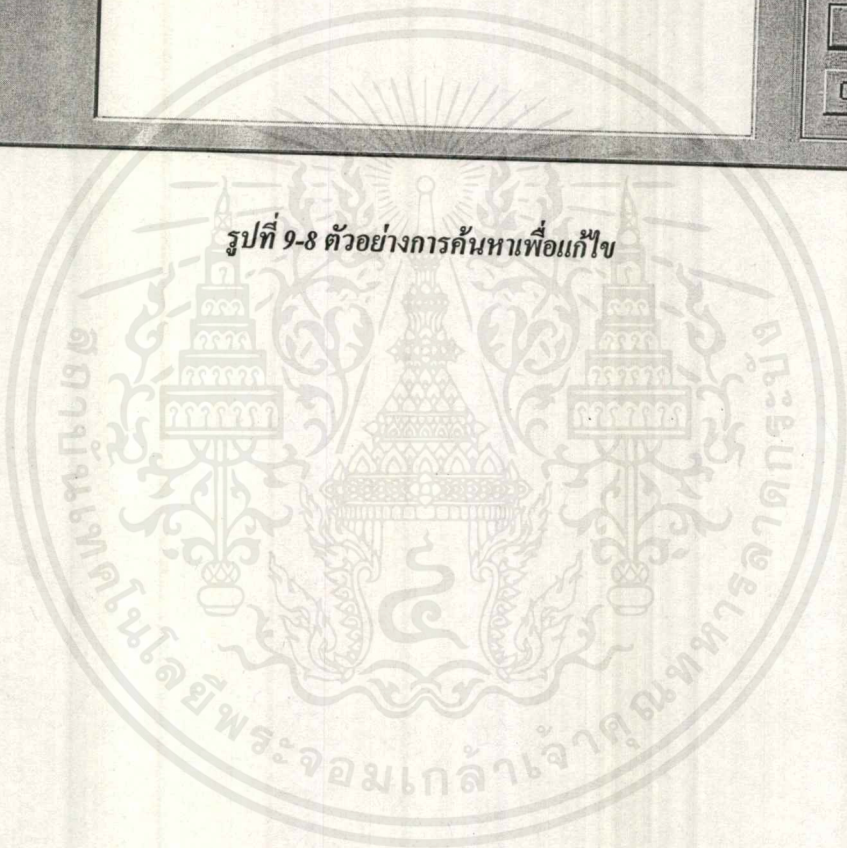
Description

Example File

Cover Image File

Media Type

Title Type



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 10

สรุปวิจารณ์

จากการนำระบบค้นหาข้อมูลทางอินเทอร์เน็ตที่พัฒนาใหม่ไปใช้แทน ทำให้ผู้ใช้ค้นหาสื่อได้รวดเร็ว และมีประสิทธิภาพมากขึ้น อีกทั้งผู้ใช้สนใจและสนุกในการค้นหา มากกว่าเดิมเนื่องจากมีภาพปก รายละเอียด และตัวอย่างของสื่อที่เป็นภาพเคลื่อนไหวหรือเสียงให้กับผู้ใช้ อีกทั้งระบบนี้ ผู้ใช้สามารถใช้ได้ทั้งภายในและจากภายนอกห้องโสตทัศนศึกษา ผ่านทางระบบอินเทอร์เน็ต เพียงแค่มีเว็บเบราว์เซอร์เช่น Netscape Navigator เท่านั้น และระบบนี้ยังสามารถพัฒนาต่อไปเพื่อให้เกิดระบบสารสนเทศของห้องโสตทัศนศึกษาที่สมบูรณ์ต่อไปได้อีกด้วย

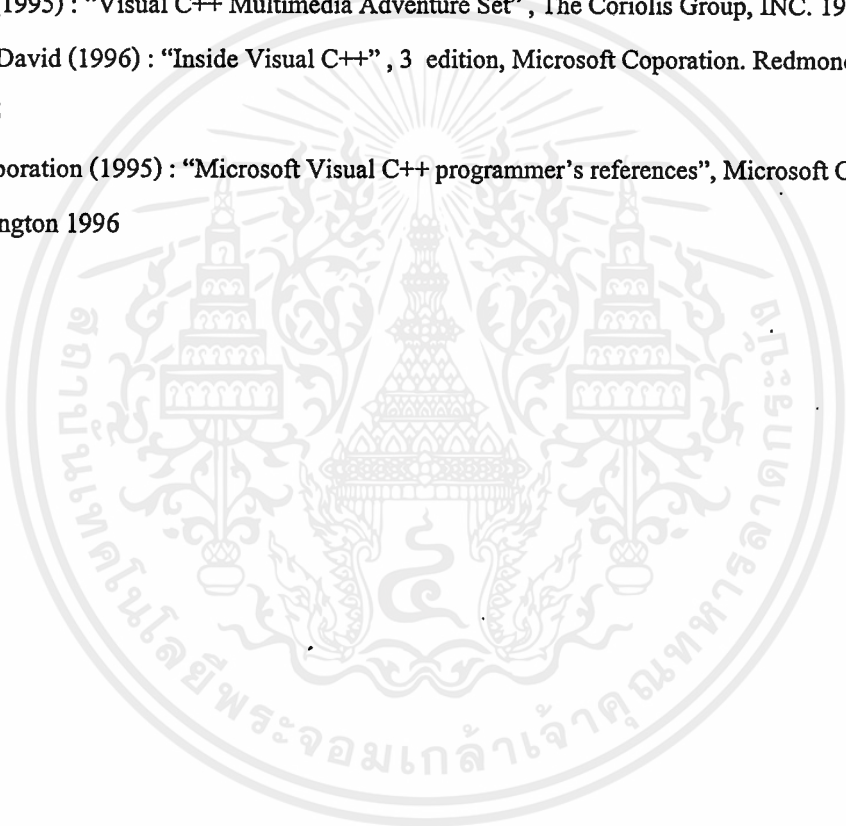
การใช้ระบบที่พัฒนาใหม่นี้มีข้อสรุปได้ดังนี้

1. สามารถทำให้การค้นหาของผู้ใช้มีประสิทธิภาพมากขึ้น เนื่องจากใช้คอมพิวเตอร์ในการค้นหา ผู้ใช้เพียงแต่ป้อนคำที่ต้องการเท่านั้น
2. ข้อมูลเกี่ยวกับสื่อที่ได้จากระบบใหม่มีความครบถ้วนมากกว่าเดิมเนื่องจากมีภาพปก และตัวอย่างของสื่อที่เป็นภาพเคลื่อนไหวมากยิ่งขึ้น ทำให้ผู้ใช้ได้สื่อตามที่ต้องการ
3. ผู้ใช้มีความสนใจและสนุกกับการค้นหาเพิ่มขึ้น เนื่องจากมีการแสดงภาพเคลื่อนไหวและเสียงประกอบ
4. การแก้ไขฐานข้อมูลของผู้ดูแลระบบเป็นไปอย่างรวดเร็วและง่ายต่อการใช้งาน

จากการทดลองทำให้เห็นว่าสมควรอย่างยิ่งที่จะนำระบบนี้ไปใช้งานจริง ตามห้องโสตทัศนศึกษาต่าง ๆ

บรรณานุกรม

- [1] Wynkoop Stephen (1997) : *"Special Edition Using Microsoft SQL Server 6.5"*, 2nd edition, Inner Media., Hollis, NH. 1997
- [2] C.J. Date (1986) : *"An Introduction of Database System"*, 6th edition, Addison Weley, Massachusetts, p.2-74, 1986
- [3]Victor Wolters (1996) : *"Special Edition Using Microsoft Internet Information Server"*, Inner Media, Hollis, NH. 1996
- [4] Aitken Peter (1995) : *"Visual C++ Multimedia Adventure Set"*, The Coriolis Group, INC. 1995
- [5]J.Krunglinski David (1996) : *"Inside Visual C++"*, 3 edition, Microsoft Coporation. Redmond, Washington 1996
- [6] Microsoft Coporation (1995) : *"Microsoft Visual C++ programmer's references"*, Microsoft Coporation. Redmond, Washington 1996



กิตติกรรมประกาศ



โครงการฉบับนี้จะไม่สำเร็จลุล่วงไปได้ หากไม่ได้ความช่วยเหลือจากอาจารย์ที่ปรึกษา
อ.เอื้อน ปิ่นเงิน ที่ให้คำแนะนำในเรื่องต่างๆ รวมทั้งพี่ ๆ ที่ห้องโสตทัศนศึกษา สำนักหอสมุดกลางทุกคน
นอกจากนี้ยังขอขอบคุณเพื่อน ๆ น้อง ๆ ทุกคนที่เป็นกำลังใจให้งานเสร็จสิ้นสมบูรณ์ ขอขอบคุณทุกคน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้