

การพัฒนาโปรแกรมภายใต้ระบบปฏิบัติการลินุกซ์

Linux Application Development



โดย

นาย ต่อชัย พัฒนพานิชสวัสดิ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

Handwritten signatures and notes in Thai script.

เลขหมู่.....

เลขทะเบียน.....33121

วัน, เดือน, ปี 15.08.2542

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาโปรแกรมภายใต้ระบบปฏิบัติการลินุกซ์

Linux Application Development

โดย

นาย ต่อชัย พัฒนพานิชสวัสดิ์ 38014166

อาจารย์ที่ปรึกษา

รศ.ดร. สุวิพล สิทธีชีวกาศ

ผศ. เกรียงไกร วงศ์โรจนารณ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2541

ภาควิชาวิศวกรรมโทรคมนาคม

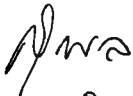
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาโปรแกรมภายใต้ระบบปฏิบัติการลินุกซ์

**Linux Application Development**

ผู้จัดทำ

นาย ต่อชัย พัฒนพานิชสวัสดิ์ 38014166



(รศ.ดร. สุวิมล สิทธิชีวภาค)

อาจารย์ที่ปรึกษา



(ผศ. เกรียงไกร วงศ์โรจนกรณ์)

อาจารย์ที่ปรึกษา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาโปรแกรมภายใต้ระบบปฏิบัติการลินุกซ์  
Linux Application Development

โดย นาย ต่อชัย พัฒนพานิชสวัสดิ์ 38014166

อาจารย์ที่ปรึกษา รศ.ดร. สุวิพล สิริชิวภาค  
ผศ. เกียรติกร วงศ์โรจนกรณ์

**บทคัดย่อ**

โครงการนี้เป็นการพัฒนาระบบฐานข้อมูลสารสนเทศขึ้นเพื่อการใช้งาน ในส่วนของการจัดการฐานข้อมูลภายในภาควิชา เพื่อให้ผู้ใช้สามารถสืบค้นข้อมูลได้อย่างรวดเร็วและมีประสิทธิภาพ โดยนำเทคโนโลยีทางด้านสารสนเทศและระบบอินเทอร์เน็ตมาช่วยในการพัฒนาเพื่อให้ใช้งานได้กว้างขวางมากขึ้น

การพัฒนาระบบฐานข้อมูลดังกล่าวใช้ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ MySQL ซึ่งทำงานบนระบบปฏิบัติการลินุกซ์ (Linux) และใช้ภาษาสคริปต์ PHP ในการติดต่อกับ ฐานข้อมูลและเขียนภาษา HTML ในการติดต่อกับผู้ใช้งาน เพื่อแสดงผลในรูปแบบ Web เพื่อให้ผู้ใช้งานใช้ได้อย่างง่าย และสะดวกจากเครื่องคอมพิวเตอร์ใดๆ ก็ได้ที่ติดต่อกับระบบอินเทอร์เน็ตไว้

**Abstract**

This project is aimed to develop the Information Database System in order to facilitate Information management within the department. Thus, the user, both internal and external personnels, will be able to utilize the information more expediently and effectively. Information Technology and Internet System are brought in here as the assisting tools for further development and innovation.

Such database development, as prescribed earlier, uses relative database management system; MySQL, which is operating on LINUX, and PHP Script language to communicate-bilaterally with the database. The result, moreover, can be displayed on the web by the use of HTML for remote personnels with Internet accessibility.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

เรื่อง	หน้า
บทที่1 บทนำ	1
บทที่2 ทฤษฎี	
2.1 ระบบสารสนเทศ	2
2.2 การจัดการฐานข้อมูล	5
2.3 สถาปัตยกรรมของระบบฐานข้อมูล	32
2.4 การออกแบบฐานข้อมูล	37
2.5 ภาษา SQL	51
2.6 PHP	84
2.7 World Wide Web (WWW)	89
2.8 ภาษา HTML	91
2.9 ลินุกซ์ (Linux)	102
บทที่ 3 การสร้างและการพัฒนาโปรแกรม	
3.1 ศึกษาและวิเคราะห์วางแผนการศึกษา	103
3.2 วัตถุประสงค์ของโครงการ	103
3.3 โครงสร้างของระบบ	103
3.4 อุปกรณ์ที่ใช้ในระบบฐานข้อมูล	104
3.5 ขั้นตอนในการพัฒนาโครงการ	105
3.6 การออกแบบระบบและการสร้างระบบฐานข้อมูล	105
บทที่ 4 ผลการทดลอง	
4.1 การทดลองส่วนหน้าจอหลัก	118
4.2 การทดลองส่วนการค้นหารายชื่อนักศึกษา	118
4.3 การทดลองส่วนการค้นหาวិชาเรียน	123
4.4 การทดลองส่วนการค้นหารายชื่อโครงการนักศึกษา	128
4.5 การทดลองส่วนการค้นหารายชื่ออาจารย์	134
4.6 การทดลองส่วนการค้นหารายชื่อนักศึกษาเก่า	138
บทที่ 5 วิจารณ์และสรุป	140
ภาคผนวก	
หนังสืออ้างอิง	

## สารบัญรูปภาพ

ลำดับภาพ	หน้า
รูปที่ 2.1 ระบบการประมวลผลเพิ่มข้อมูล	6
รูปที่ 2.2 ระบบการประมวลผลฐานข้อมูล	8
รูปที่ 2.3 แสดงตัวอย่างเพิ่มข้อมูลนักศึกษา	12
รูปที่ 2.4 แสดงตัวอย่างของเอนิตี และ แอททริบิวท์	13
รูปที่ 2.5 แสดงความสัมพันธ์ระหว่างเอนิตีนักศึกษาและเอนิตีสาขาวิชาโดยอาศัยรหัสสาขา และรหัสคณะเป็นตัวเชื่อม	13
รูปที่ 2.6 แสดงตัวอย่างความสัมพันธ์แบบหนึ่งต่อกลุ่ม	15
รูปที่ 2.7 แสดงตัวอย่างความสัมพันธ์แบบกลุ่มต่อกลุ่ม	16
รูปที่ 2.8 ก. แสดง โครงสร้างฐานข้อมูลแบบลำดับชั้น	17
รูปที่ 2.8 ข. แสดงตัวอย่างข้อมูลในฐานข้อมูลแบบลำดับชั้น	18
รูปที่ 2.9 แสดง โครงสร้างรีเลชัน	19
รูปที่ 2.10 แสดงองค์ประกอบของภาษา 3 รูปแบบของภาษา SQL	22
รูปที่ 2.11 ตัวอย่างตารางแสดงการกำหนดสิทธิให้กับผู้ใช้งานข้อมูลแต่ละคน	23
รูปที่ 2.12 แสดงส่วนประกอบของระบบ จัดการฐานข้อมูล (DBMS)	24
รูปที่ 2.13 แสดงความสัมพันธ์กันของข้อมูล	29
รูปที่ 2.14 แสดงความสัมพันธ์ระหว่างองค์ประกอบของระบบฐานข้อมูล	31
รูปที่ 2.15 แสดงสถาปัตยกรรม 3 ระดับของระบบฐานข้อมูล	33
รูปที่ 2.16 ตัวอย่างสถาปัตยกรรม 3 ระดับ ของฐานข้อมูลระบบทะเบียนนักศึกษา	35
รูปที่ 2.17 แผนภาพแบบ E-R ของบริษัทสุจริตการก่อสร้าง จำกัด	43
รูปที่ 2.18 แสดงตัวอย่างความสัมพันธ์แบบหนึ่งต่อหนึ่ง	45
รูปที่ 2.19 แสดงตัวอย่างข้อมูลภายในรีเลชันที่มีการแปลงอย่างไม่ถูกต้อง	47
รูปที่ 2.20 แสดงตัวอย่างความสัมพันธ์ของข้อมูลภายในรีเลชันคนงานและการทำงานที่เป็นแบบ 1:N และรีเลชันสถานที่ก่อสร้างกับการทำงานก็เป็นแบบ 1:N	48
รูปที่ 2.21 แสดงความสัมพันธ์รีเคอร์ซีฟแบบ 1:N	49
รูปที่ 2.22 แสดงตัวอย่างข้อมูลภายในรีเลชันคนงานที่มีความสัมพันธ์แบบรีเคอร์ซีฟ	50
รูปที่ 2.23 แสดงผลลัพธ์ในตารางชั่วคราวที่เกิดจากการเชื่อมข้อมูลจากตารางคนงาน และตารางการทำงาน	67
รูปที่ 2.24 แสดงผลลัพธ์ในตารางชั่วคราวที่เกิดจากเงื่อนไขไรหัดสถานที่ก่อสร้าง = 516	67
รูปที่ 3.1 แสดง ไฟล์ชาร์ตแสดงการทำงานของระบบ	106
รูปที่ 3.2 แสดงการออกแบบหน้าจอส่วน Client	107

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ (ต่อ)

ลำดับภาพ	หน้า
รูปที่ 3.3 แสดง โมเดลเชิงสัมพันธ์ของระบบฐานข้อมูลสารสนเทศ	111
รูปที่ 4.1 แสดง หน้าจอเมนูหลัก	118
รูปที่ 4.2 แสดง หน้าจอเพื่อรับข้อมูลเพื่อการแสดงข้อมูลรายชื่อนักศึกษา	119
รูปที่ 4.3 แสดง ผลลัพธ์จากการค้นหารายชื่อนักศึกษา	120
รูปที่ 4.4 แสดง หน้าจอ ในกรณีที่ไม่มีพบข้อมูลที่ต้องการ	121
รูปที่ 4.5 แสดง ผลลัพธ์จากการค้นหาเลขประจำตัวนักศึกษา	122
รูปที่ 4.6 แสดง รายชื่อนักศึกษาปี 4A ทั้งหมด	123
รูปที่ 4.7 แสดง หน้าจอการให้เลือกภาคเรียน	124
รูปที่ 4.8 แสดง หน้าจอการรับข้อมูล เพื่อการแสดงข้อมูลวิชาเรียน	125
รูปที่ 4.9 แสดง ผลลัพธ์จากการค้นหารายชื่อวิชา โดยรหัสวิชา	126
รูปที่ 4.10 แสดง ผลลัพธ์จากการค้นหารายชื่อวิชา	127
รูปที่ 4.11 แสดง รายชื่อวิชาปี 4A เทอม 1 ทั้งหมด	128
รูปที่ 4.12 แสดง หน้าจอการรับข้อมูล เพื่อการแสดงข้อมูลชื่อ โครงการนักศึกษา	129
รูปที่ 4.13 แสดง ผลลัพธ์จากการค้นหาชื่อ โครงการ	130
รูปที่ 4.14 แสดง ผลลัพธ์จากการค้นหา รายชื่อ โครงการ โดยชื่อนักศึกษา	131
รูปที่ 4.15 แสดง ผลลัพธ์จากการค้นหา รายชื่อ โครงการ โดยเลขประจำตัวนักศึกษา	132
รูปที่ 4.16 แสดง ผลลัพธ์จากการค้นหา รายชื่อ โครงการ โดยชื่ออาจารย์ที่ปรึกษา	133
รูปที่ 4.17 แสดง รายชื่อ โครงการ ปีการศึกษา 2541 ทั้งหมด	134
รูปที่ 4.18 แสดง หน้าจอการรับข้อมูล เพื่อการแสดงข้อมูลอาจารย์	135
รูปที่ 4.19 แสดง ผลลัพธ์จากการค้นหาชื่ออาจารย์	136
รูปที่ 4.20 แสดง ผลลัพธ์การค้นหาชื่ออาจารย์ โดยห้องทำงานอาจารย์	137
รูปที่ 4.21 แสดง รายชื่ออาจารย์ของภาควิชา โทศมนาคมทั้งหมด	138

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ลำดับตาราง	หน้า
ตารางที่ 2.5 แสดงกลุ่มคำสั่งที่เป็น DML	57
ตารางที่ 2.6 ตารางแสดงเครื่องหมายแสดงความสัมพันธ์และตัวอย่าง	58
ตารางที่ 2.7 ตารางแสดงค่าตัวอย่างของนิพจน์	59
ตารางที่ 2.8 ตารางแสดงสัญลักษณ์ทางตรรกศาสตร์และความหมาย	59
ตารางที่ 2.9 ตัวอย่างการใช้ AND และ OR	60
ตารางที่ 2.10 ก. ตารางแสดงคนงาน	61
ตารางที่ 2.10 ข. ตารางแสดงการทำงาน	61
ตารางที่ 2.10 ค. ตารางแสดงสถานที่ก่อสร้าง	62
ตารางที่ 2.10 ง. ตารางแสดงความชำนาญ	62
ตารางที่ 2.11 แสดงฟังก์ชันทางสถิติที่มีใช้ใน SQL	72
ตารางที่ 2.12 แสดงวิว ELECT_WORK	81
ตารางที่ 3.1 แสดงตาราง studentXXXX	108
ตารางที่ 3.2 แสดงตาราง projectXXXX	109
ตารางที่ 3.3 แสดงตาราง studyXXXX	109
ตารางที่ 3.4 แสดงตาราง teacherXXXX	109

# บทที่ 1

## บทนำ

การจัดเก็บข้อมูลและข่าวสาร โดยทั่วไป จะอยู่ในรูปของเอกสาร หลักฐาน ซึ่งอาจจะเป็นหนังสือ สมุดบันทึก รายงาน หรือจดหมายเหตุ เป็นต้น เอกสารเหล่านี้มักกระจัดกระจายกันอยู่ และอีกทั้งสภาพทางกายภาพของสื่อเหล่านี้ ส่วนใหญ่จะเป็นกระดาษ มีการเปลี่ยนแปลงผุกร่อนไปตามธรรมชาติ เมื่อเวลาผ่านไป ข้อมูลที่บันทึกอยู่ในสื่อเหล่านี้ ย่อมเกิดการสูญหาย หรือเลอะเลือนจนไร้ค่า

ต่อมาได้มีการพัฒนาจัดเก็บเอกสาร หลักฐานต่างๆ โดยจัดเก็บไว้เป็นหมวดหมู่ มีสถานที่เก็บที่ดีขึ้น ง่ายต่อการค้นหา จวบจนปัจจุบันได้มีการคิดค้นเครื่องคอมพิวเตอร์ขึ้นมา และใช้งานกันอย่างกว้างขวาง เช่น ใช้เครื่องคอมพิวเตอร์ช่วยงานในโรงงานอุตสาหกรรม งานด้านวิทยาศาสตร์ด้านการแพทย์ และที่สำคัญปัจจุบันได้มีการนำเอาคอมพิวเตอร์มาใช้งานในด้านการสื่อสาร มีการจัดเก็บข้อมูลลงในระบบคอมพิวเตอร์ มีการจัดการข้อมูล (Information Management) เพื่อนำข้อมูลมาใช้ให้เกิดประโยชน์ อย่างมีประสิทธิภาพ หน่วยงาน โดยทั่วไปจึงต้องการใช้คอมพิวเตอร์เพื่อการจัดเก็บ และจัดการข้อมูล

โครงการนี้เป็นการพัฒนาระบบฐานข้อมูลสารสนเทศขึ้นเพื่อการใช้งาน ในส่วนของการจัดการฐานข้อมูลภายในภาควิชา เพื่อให้ผู้ใช้สามารถสืบค้นข้อมูลได้อย่างรวดเร็วและมีประสิทธิภาพ โดยนำเทคโนโลยีทางด้านสารสนเทศและระบบอินเทอร์เน็ตมาช่วยในการพัฒนาเพื่อให้ใช้งานได้กว้างขวางมากขึ้น

การพัฒนาระบบฐานข้อมูลดังกล่าวใช้ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ MySQL ซึ่งทำงานบนระบบปฏิบัติการลินุกซ์ (Linux) และใช้ภาษาสคริปต์ PHP ในการติดต่อกับ ฐานข้อมูลและเขียนภาษา HTML ในการติดต่อกับผู้ใช้งาน เพื่อแสดงผลในรูปแบบ Web เพื่อให้ผู้ใช้งานใช้ได้อย่างง่าย และสะดวกจากเครื่องคอมพิวเตอร์ใดๆ ก็ได้ที่ต่อกับระบบอินเทอร์เน็ตไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### ทฤษฎี

#### 2.1 ระบบสารสนเทศ

เนื่องจากในปัจจุบันได้มีการให้ความสำคัญเกี่ยวกับระบบสารสนเทศ (Information System) กันมาก ซึ่งเป็นการรวบรวมข้อมูลที่เรียกว่า ข้อมูลดิบ(raw data) จากที่ต่างๆ มาผ่านกระบวนการเช่นการเรียงลำดับ การคำนวณ การจัดกลุ่ม หรือสรุปผล เพื่อสร้างเป็นรายงานหรือ จัดให้อยู่ในรูปแบบที่เหมาะสมต่อการนำเสนอขององค์กร ซึ่งจะเรียกข้อมูลดิบหลังจากที่ผ่านกระบวนการข้างต้นแล้วนี้ว่าเป็น สารสนเทศ (Information)

โดยทั่วไปสารสนเทศไม่จำเป็นต้องเกิดจากข้อมูลดิบที่ถูกเก็บและถูกประมวลผลโดยคอมพิวเตอร์เท่านั้น ข้อมูลดิบสามารถรวบรวมด้วยวิธีใดก็ได้เช่นอาจอยู่ในรูปของกระดาษ และใช้คนเป็นผู้จัดการข้อมูลเหล่านั้น เพื่อสร้างเป็นสารสนเทศก็ได้ แต่เนื่องจากคอมพิวเตอร์ได้เข้ามามีบทบาทมาก และในปัจจุบันต้องใช้ข้อมูลอย่างรวดเร็วทันต่อเหตุการณ์และมีความถูกต้องเชื่อถือได้ ดังนั้นเมื่อมีการกล่าวถึงระบบสารสนเทศ ส่วนใหญ่จะหมายถึงการเก็บรวบรวมข้อมูลและทำการประมวลผลโดยใช้คอมพิวเตอร์ จึงอาจเรียกได้ว่าเป็น Computer Information System หรือ CIS

##### 2.1.1 องค์ประกอบของระบบสารสนเทศ

ขบวนการหรือขั้นตอนการประมวลผลข้อมูลให้เป็นสารสนเทศ จะต้องประกอบด้วยองค์ประกอบดังต่อไปนี้

- ฮาร์ดแวร์(Hardware)
- ซอฟต์แวร์(Software)
- ข้อมูล(Stored Data)
- บุคลากร(personnel)
- ขั้นตอนการดำเนินงาน(Procedures)

##### 2.1.1.1 ฮาร์ดแวร์ (Hardware)

เป็นองค์ประกอบแรกของระบบสารสนเทศ ซึ่งฮาร์ดแวร์นี้จะหมายถึงอุปกรณ์ทางคอมพิวเตอร์ที่ใช้ในการเก็บข้อมูลและประมวลผลข้อมูลเพื่อสร้างสารสนเทศขึ้น อันได้แก่เครื่องคอมพิวเตอร์ซึ่งอาจเป็นได้ตั้งแต่เครื่องระดับไมโครคอมพิวเตอร์ เมนเฟรมคอมพิวเตอร์ หรือ แม้กระทั่งซูเปอร์คอมพิวเตอร์ซึ่งเป็นเครื่องที่มีขนาดใหญ่สุด นอกจากนี้สารสนเทศยังสามารถถูกเก็บอยู่ในระบบเครือข่าย (Network) ซึ่งเป็นการเชื่อมโยงไมโครคอมพิวเตอร์หลายตัวเข้าด้วยกัน และเชื่อมกับเครื่องขนาดใหญ่เช่น เมนเฟรมคอมพิวเตอร์ อีกได้เช่นกัน

จากตัวอย่างของธุรกิจการออกอัลบั้มเพลง ถ้าต้องการเก็บข้อมูลต่าง ๆ ไม่ว่าจะเป็นข้อมูลการขายหรือข้อมูลลูกค้าเข้าสู่เครื่องคอมพิวเตอร์ จะต้องใช้อุปกรณ์ที่เป็นหน่วยนำข้อมูลเข้าเช่นเทอร์มินัล เอกสารนี้เป็นเอกสารที่ส่งวนเวียนสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Terminal) ซึ่งประกอบด้วยแป้นพิมพ์และจอภาพ จำเป็นต้องมีการใช้หน่วยเก็บข้อมูลสำรองเช่นจานแม่เหล็กหรือดิสก์ (Disk) ในการเก็บข้อมูลที่นำเข้านี้ไว้อย่างถาวร จำเป็นต้องมีหน่วยประมวลผลกลางได้แก่ ซีพียู ทำการประมวลผลเช่นถ้าจะคำนวณกายออคขายโดยรวมของอัลบัลเพลงแต่ละระแเภท และสร้างรายงานว่าอัลบัลประเภทใดที่มียอดขายสูงสุดเป็นต้น และท้ายสุดต้องมีอุปกรณ์นำข้อมูลออก เช่น เครื่องพิมพ์ เพื่อพิมพ์รายงานสรุปรายการขายออกมาบนกระดาษ ซึ่งจะเป็นสารสนเทศที่ส่งให้ผู้บริหารใช้ประกอบการตัดสินใจอีกที

ถ้ามีการใช้ระบบเครือข่ายในการสร้างสารสนเทศ ก็จำเป็นต้องมีอุปกรณ์ทางการสื่อสารอื่นๆ เช่น โมเด็ม(Modem) ฯลฯ เป็นองค์ประกอบทางฮาร์ดแวร์ด้วย

### 2.1.1.2 ซอฟต์แวร์ (Software)

เป็นองค์ประกอบที่สำคัญอีกองค์ประกอบหนึ่งขบวนการสร้างสารสนเทศ ซึ่งซอฟต์แวร์จะหมายถึง โปรแกรมเหนือชุดคำสั่งที่ถูกเขียนขึ้นมา เพื่อใช้สั่งงานคอมพิวเตอร์ให้ทำงาน สามารถแบ่งซอฟต์แวร์ได้เป็น 2 ประเภทใหญ่ ได้แก่

- ซอฟต์แวร์ประยุกต์(Application Software)
- ซอฟต์แวร์ระบบ(system Software)

ซอฟต์แวร์ทั้งสองนี้มีความสำคัญต่อการสร้างระบบสารสนเทศโดยซอฟต์แวร์ประยุกต์โดยทั่วไปจะเป็นโปรแกรมที่ถูกพัฒนาขึ้นมาเพื่อการทำงานในเรื่องใดเรื่องหนึ่งโดยเฉพาะ เช่น โปรแกรมระบบสินค้าคงคลังซึ่งอาจใช้ภาษาขั้นสูง เช่น โคบอล(Cobol) ซี (C) ปาสคาล (Pascal) หรือภาษาในยุคที่ 4 (4GLs:Fourth Generation Language)เป็นตัวพัฒนา เป็นต้น

สำหรับซอฟต์แวร์ระบบโดยทั่วไปจะ ได้แก่ โปรแกรมที่มีหน้าที่ควบคุมเครื่องคอมพิวเตอร์ให้สามารถปฏิบัติงานได้อย่างราบรื่น รวมทั้งควบคุมการทำงานของอุปกรณ์ต่าง ๆ ที่ต่ออยู่กับระบบคอมพิวเตอร์ตัวอย่างเช่น โปรแกรมระบบปฏิบัติการดอส (DOS)หรือ ยูนิกซ์ (UNIX) แต่ส่วนใหญ่แล้วผู้ใช้ระบบมักต้องยุ่งเกี่ยวกับซอฟต์แวร์เป็นหลัก

### 2.1.1.3 ข้อมูล (Stored Data)

เป็นข้อมูลที่เก็บรวบรวมไว้ในระบบคอมพิวเตอร์ จะถูกเรียกใช้เพื่อการประมวลผลโดยโปรแกรมประยุกต์ต่าง ๆ ข้อมูลที่เก็บรวบรวมไว้ในระบบคอมพิวเตอร์นี้ อาจอยู่ในรูปของ

- แฟ้มข้อมูลหรือไฟล์
- ฐานข้อมูล

ข้อมูลที่เก็บอยู่นี้อาจเป็นแฟ้มข้อมูลเพียงแฟ้มเดียว หรือหลายแฟ้ม หรืออยู่ในรูปของฐานข้อมูลซึ่งจะเป็นการรวมแฟ้มข้อมูลตั้งแต่หนึ่งแฟ้มข้อมูลขึ้นไป ที่มีความสัมพันธ์กันเก็บไว้ในที่ที่เดียวกันในหน่วยเก็บข้อมูลสำรองเช่นจานแม่เหล็กหรือดิสก์ เพื่อให้บุคลากรจากหลายหน่วยงานสามารถใช้ข้อมูลในฐานข้อมูลนี้ร่วมกันได้

#### 2.1.1.4 บุคลากร (Personal)

ระบบสารสนเทศจะไม่สามารถปฏิบัติงานต่าง ๆ ได้เอง ถ้าไม่มีคนเป็นผู้จัดการ คนในที่นี่จะหมายถึงบุคลากรประเภทต่างๆ ดังต่อไปนี้

- ผู้ใช้งาน (Users)
- ผู้ปฏิบัติงาน (Operating Personnel)
- ผู้ควบคุมระบบและพัฒนาโปรแกรม (System and Application Programmer)

ผู้ใช้งาน โดยทั่วไปจะเป็นผู้ที่นำสารสนเทศที่เกิดจากระบบคอมพิวเตอร์ไปใช้ ตัวอย่างเช่น รายงานลูกค้าค้างชำระจะเป็นสารสนเทศที่ส่งให้แก่พนักงานฝ่ายสินเชื่อ เพื่อนำไปใช้ในการติดตามเก็บเงินจากลูกค้าหรือรายงานสรุปยอดขายของอัลบัลด ก็จะถูกส่งให้แก่ผู้บริหารระดับสูง เพื่อใช้ในการตัดสินใจจัดสรรงบประมาณว่าจะมุ่งเน้นลงทุนกับอัลบัลดประเภทใด เป็นต้น ดังนั้นพนักงานฝ่ายสินเชื่อและผู้บริหารระดับสูงต่างก็เป็นผู้ใช้งานสารสนเทศทั้งสิ้น ซึ่งผู้ใช้งานนี้อาจเป็นบุคคลที่ไม่มีความรู้เกี่ยวกับคอมพิวเตอร์เท่าไรนักก็ได้ แต่จะรู้ขั้นตอนการเรียกใช้สารสนเทศจากระบบคอมพิวเตอร์

ผู้ปฏิบัติงาน โดยทั่วไปจะเป็นบุคลากรที่มีหน้าที่นำข้อมูลเข้าสู่ระบบคอมพิวเตอร์ ให้สามารถปฏิบัติงานได้อย่างราบรื่น ไม่มีปัญหา หรือคอยแก้ไขปัญหาที่เกิดจากการปฏิบัติงานของคอมพิวเตอร์ สำหรับผู้พัฒนาโปรแกรมจะได้แก่บุคลากรที่มีหน้าที่เกี่ยวกับการพัฒนาโปรแกรมประยุกต์ต่าง ๆ เพื่อสั่งงานให้คอมพิวเตอร์ทำการประมวลผลและสร้างสารสนเทศในระบบงานใด ๆ เป็นต้น

#### 2.1.1.5 ขั้นตอนการดำเนินงาน

องค์ประกอบสุดท้ายของระบบสารสนเทศได้แก่ขั้นตอนการดำเนินงานหรือการปฏิบัติงาน โดยถ้าเปรียบว่าฮาร์ดแวร์จะไม่สามารถทำงานได้ถ้าปราศจากซอฟต์แวร์ คนก็จะไม่รู้ว่าจะต้องทำอะไร ถ้าปราศจากขั้นตอนการดำเนินงาน

ขั้นตอนการดำเนินงานจะเป็นสิ่งที่บอกให้ผู้ใช้งานสารสนเทศจากระบบคอมพิวเตอร์ได้อย่างไร และจะบอกผู้ปฏิบัติงานว่าจะสั่งให้ระบบคอมพิวเตอร์ทำงานได้อย่างไร ซึ่งผู้ใช้และผู้ปฏิบัติงานว่าจะสั่งให้ระบบคอมพิวเตอร์ทำงานได้อย่างไร ซึ่งผู้ใช้และผู้ปฏิบัติงานจะต้องได้รับการอบรมถึงขั้นตอนการป้อนข้อมูลเข้าสู่คอมพิวเตอร์ หรือขั้นตอนการเรียกใช้โปรแกรมประยุกต์ต่าง ๆ หรือขั้นตอนการรับสารสนเทศจากคอมพิวเตอร์ เป็นต้น

## 2.2 การจัดการฐานข้อมูล

### 2.2.1 การประมวลผลในระบบสารสนเทศ

จากที่ได้กล่าวแล้วว่าข้อมูลที่ถูกเก็บอยู่ในระบบคอมพิวเตอร์เพื่อนำมาใช้สร้างเป็นสารสนเทศนั้น อาจอยู่ในรูปของแฟ้มข้อมูล(หรือ ไฟล์)หรืออาจอยู่ในรูปของฐานข้อมูล ซึ่งการประมวลผลข้อมูลเพื่อสร้างระบบสารสนเทศนั้น ถ้าข้อมูลที่เก็บนั้นเป็นระบบแฟ้มข้อมูล เมื่อเปรียบเทียบกับระบบฐานข้อมูล จะพบว่ามีความคล้ายคลึงกัน ดังจะได้กล่าวในหัวข้อต่อไป

### 2.2.2 ระบบการประมวลผลแฟ้มข้อมูล

แรกเริ่มที่มีการใช้ระบบคอมพิวเตอร์เพื่อการประมวลผลข้อมูลทางธุรกิจ หรือเพื่อสร้างสารสนเทศนั้นจะมีการเก็บกลุ่มของระเบียบต่าง ๆ ไว้ในแฟ้มข้อมูลที่แยกจากกันและจะเรียกว่าเป็นระบบการประมวลผลแฟ้มข้อมูล ถึงแม้ว่าระบบการประมวลผลแฟ้มข้อมูลนี้ก็ยังมีข้อจำกัดหลายอย่างด้วยกัน ดังจะแสดงให้เห็นต่อไปนี้

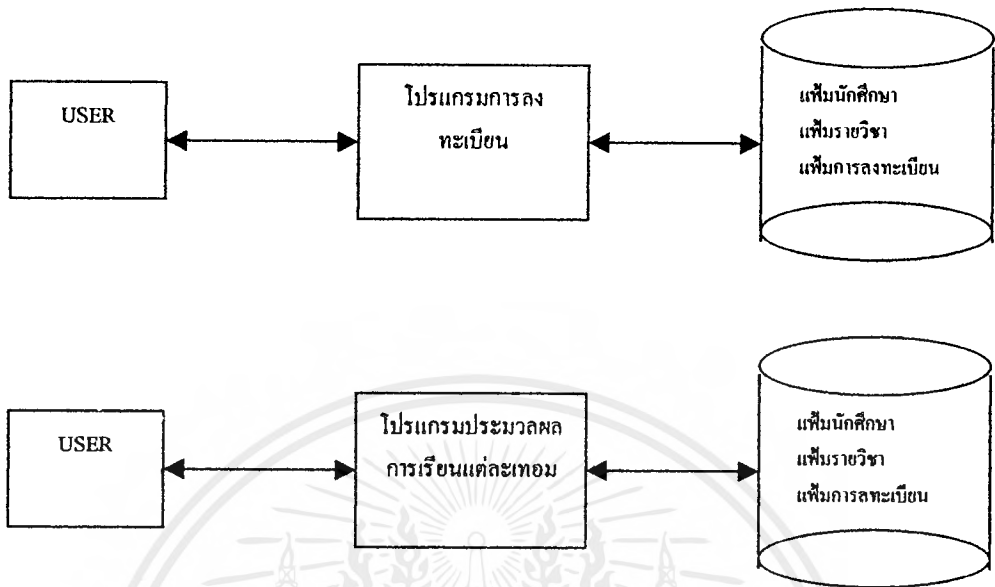
จะขอยกตัวอย่างการสร้างระบบสารสนเทศในระบบการศึกษาของมหาวิทยาลัย ซึ่งสำนักทะเบียนมหาวิทยาลัยจะต้องรับลงทะเบียนนักศึกษาแต่ละคน เพื่อทำรายงานสรุปว่าวิชาแต่ละวิชามีนักศึกษาลงทะเบียนเรียนกี่คน นักศึกษาแต่ละคนมีการลงทะเบียนเรียนวิชาใดบ้าง และนอกจากนี้ยังต้องทำการรวบรวมผลการเรียนของนักศึกษาแต่ละคนจากอาจารย์ในแต่ละวิชา เพื่อนำมาทำรายงานสรุปผลการเรียนของนักศึกษาแต่ละคนในแต่ละปีการศึกษา ซึ่งรายงานต่าง ๆ นั้นจะถือว่าเป็นสารสนเทศเพื่อส่งให้นักศึกษาและอาจารย์ที่ปรึกษา

ในการนี้ถ้าใช้ระบบแฟ้มข้อมูลในการประมวลผลข้อมูล ระบบแฟ้มข้อมูลหลักที่ต้องมีจะประกอบด้วยแฟ้มต่าง ๆ เช่น แฟ้มรายวิชา แฟ้มนักศึกษา แฟ้มการลงทะเบียน และแฟ้มผลการเรียนประจำเทอม แต่ละแฟ้มจะประกอบด้วยเขตข้อมูลหรือฟิลด์(Fields) ต่าง ๆ เช่น แฟ้มรายวิชาประกอบด้วยเขตข้อมูลรหัสวิชา ชื่อวิชาและจำนวนหน่วยกิต แฟ้มนักศึกษาประกอบด้วยข้อมูลรหัสนักศึกษา ชื่อ-สกุล ที่อยู่ รหัสนักศึกษา แฟ้มการลงทะเบียนประกอบด้วยปีการศึกษา เทอม รหัสนักศึกษา และเกรดของวิชาที่ลง แฟ้มผลการเรียนประจำเทอมประกอบด้วยเขตข้อมูลปีการศึกษา เทอม รหัสนักศึกษา จำนวนหน่วยกิตต่อเทอม คะแนนสะสมต่อเทอม

ในระบบการลงทะเบียน ซึ่งจะต้องมีการออกรายงานว่ามีนักศึกษาลงทะเบียนเรียนวิชาใดบ้าง หรือวิชาแต่ละวิชามีนักศึกษาค้นใดบ้างที่ลงทะเบียนเรียน จะต้องมีการสร้าง โปรแกรมประยุกต์ซึ่งอาจเขียนด้วยภาษาระดับสูงเช่น COBOL ,C ,PASCAL ฯลฯ เพื่อจัดการหรือค้นหาข้อมูลในแฟ้มข้อมูลต่าง ๆ ที่เกี่ยวข้องในการออกรายงานดังกล่าว ตัวอย่าง เช่น โปรแกรมการลงทะเบียน เพื่อจัดการข้อมูลในแฟ้มนักศึกษา แฟ้มรายวิชาและแฟ้มการลงทะเบียน

สำหรับในระบบการประมวลผลการเรียนนักศึกษา จะต้องมีการออกรายงานสรุปผลการเรียนของนักศึกษาแต่ละคนในแต่ละปีการศึกษา เพื่อให้กับนักศึกษาและอาจารย์ที่ปรึกษา จะต้องมีการสร้าง

โปรแกรมประมวลผลการเรียนแต่ละเทอมเพื่อจัดการข้อมูลในแฟ้มนักศึกษา แฟ้มรายวิชา แฟ้มการลงทะเบียนและแฟ้มผลการเรียนประจำเทอม ดังรูป 2.1



รูปที่ 2.1 ระบบการประมวลผลเพิ่มข้อมูล

### ข้อดีของการประมวลผลข้อมูลในระบบเพิ่มข้อมูล

1. การประมวลผลข้อมูลทำได้อย่างรวดเร็ว
2. ค่าลงทุนในเบื้องต้นจะต่ำ อาจไม่จำเป็นต้องใช้คอมพิวเตอร์ที่มีความสามารถมาก ก็สามารถทำการประมวลผลข้อมูลได้
3. โปรแกรมประยุกต์แต่ละโปรแกรมสามารถควบคุมการใช้ข้อมูลในแฟ้มข้อมูลของตนเองได้

### ข้อเสียของการประมวลผลข้อมูลในระบบเพิ่มข้อมูล

การประมวลผลข้อมูลในระบบเพิ่มข้อมูล อาจมีข้อเสียที่เกิดขึ้นได้ ดังต่อไปนี้

1. มีความซ้ำซ้อนของข้อมูล (Redundancy)
  - ทำให้เสียเนื้อที่การใช้งานในหน่วยเก็บข้อมูลสำรองเช่น ดิสก์
  - ถ้ามีการเปลี่ยนแปลงข้อมูลในแฟ้มข้อมูลหนึ่ง ก็จะต้องตามไปแก้ไขข้อมูลในแฟ้มข้อมูลอื่นทุกแฟ้มที่มีข้อมูลนั้นอยู่ด้วย จึงอาจเกิดปัญหาที่เกี่ยวกับความขัดแย้งของข้อมูล (data Inconsistency) เนื่องจากข้อมูลในแต่ละแฟ้มเกิดความไม่สอดคล้องกันขึ้น ซึ่งเป็นปัญหาที่พบบ่อยในระบบการประมวลผลแบบเพิ่มข้อมูล ตัวอย่างเช่น นักศึกษาคนใดมีการเปลี่ยนชื่อหรือนามสกุลนักศึกษาคนนั้นในแฟ้มนักศึกษาในระบบการลงทะเบียน ไม่มีการเปลี่ยนในแฟ้มนักศึกษาของระบบการประมวลผลการเรียน ก็จะเกิดปัญหาความขัดแย้งกันของข้อมูลขึ้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาเรื่องความขัดแย้งกันของข้อมูลนี้เป็นปัญหาที่สำคัญมาก เนื่องจากอาจเกิดกรณีที่ระบบลงทะเบียนมีการดึงชื่อ-นามสกุลนักศึกษาจากแฟ้มนักศึกษาของระบบ เพื่อดูออกรายงานการลงทะเบียน และระบบการประมวลผลการเรียนประจำเทอม ก็จะมีการดึงชื่อ-นามสกุลนักศึกษาจากแฟ้มนักศึกษาของระบบเอง มาออกรายงานผลการเรียนประจำเทอม ซึ่งจะพบว่ารายงานทั้ง 2 ฉบับจะมีนักศึกษาที่มีรหัสเดียวกัน แต่มีชื่อหรือนามสกุลแตกต่างกันเกิดขึ้น ทั้ง ๆ ที่เป็นคนเดียวกัน

## 2. ความยากในการประมวลผลข้อมูลในแฟ้มข้อมูลหลายแฟ้มข้อมูล

ในการสร้างรายงานของแต่ละระบบเช่นการสร้างรายงานการลงทะเบียน ว่าแต่ละวิชามีนักศึกษาค้นใดบ้างที่ลงทะเบียนเรียน จะต้องมีการเขียน โปรแกรมประยุกต์เช่น โปรแกรมลงทะเบียน เพื่อทำการดึงข้อมูลรหัสวิชา รหัสนักศึกษา จากแฟ้มข้อมูลการลงทะเบียน และต้องนำรหัสวิชาที่ได้ไปค้นชื่อวิชาและจำนวนหน่วยกิตที่มีรหัสวิชาตรงกันจากแฟ้มรายวิชา ส่วนรหัสนักศึกษาที่ได้ก็ต้องนำไปค้นชื่อนักศึกษาที่มีรหัสตรงกันจากแฟ้มนักศึกษา ซึ่ง โปรแกรมการลงทะเบียนที่เขียนจะต้องมีความซับซ้อนพอสมควร เนื่องจากต้องมีการจัดการกับแฟ้มข้อมูลมากกว่า 1 แฟ้มข้อมูลขึ้นไป

## 3. ไม่มีผู้ควบคุมหรือรับผิดชอบระบบทั้งหมด

เนื่องจากผู้เขียน โปรแกรมด้านใดด้านหนึ่ง ก็จะดูแลเฉพาะข้อมูลที่จะมีการใช้กับงานของตนเท่านั้น

## 4. ความขึ้นต่อกัน (Dependency) ระหว่าง โปรแกรมประยุกต์และ โครงสร้างของแฟ้มข้อมูล

โครงสร้างของแฟ้มข้อมูลมักจะเป็นส่วนหนึ่งของ โปรแกรมประยุกต์ที่สร้างขึ้น ตัวอย่างเช่นถ้ามีการเขียนโปรแกรมประยุกต์ด้วยภาษา COBOL โครงสร้างของแฟ้มข้อมูลที่จะใช้เช่นชื่อเขตข้อมูลต่าง ๆ ขนาดของเขตข้อมูล จะต้องประกาศไว้ในส่วนของ DATA DIVITION ของ โปรแกรมประยุกต์ ปัญหาก็คือถ้ามีการเปลี่ยนแปลงโครงสร้างของแฟ้มข้อมูลเมื่อใด ก็จะต้องไปทำการแก้ไข โปรแกรมประยุกต์ คือต้องไปเปลี่ยนโครงสร้างแฟ้มข้อมูลในส่วน DATA DIVITON นั้นด้วย ตัวอย่างเช่นถ้ามีการเขียนโปรแกรมลงทะเบียนด้วยภาษา COBOL โครงสร้างของแฟ้มนักศึกษา แฟ้มรายวิชา และแฟ้มการลงทะเบียน จะต้องประกาศไว้ในโปรแกรมด้วย ดังนั้นถ้ามีการเปลี่ยน โครงสร้างของแฟ้มใด เช่นแฟ้มนักศึกษาต้องการขยายขนาดของเขตข้อมูล ชื่อ-นามสกุลนักศึกษามาจากเดิมมีขนาด 20 ตัวอักษร เปลี่ยนเป็น 30 ตัวอักษร โปรแกรมการลงทะเบียนหรือ โปรแกรมอื่นๆ ที่มีการเรียกใช้แฟ้มข้อมูลนักศึกษา จะต้องถูกนำมาแก้ไข โปรแกรมย่อมจะต้องใช้เวลานานและอาจมีข้อผิดพลาดเกิดขึ้นได้

จากข้อเสียดังกล่าวของการประมวลผลในระบบแฟ้มข้อมูล จึงเป็นที่มาของการพัฒนาระบบการประมวลผลแบบใหม่ เพื่อแก้ไขข้อเสียของการประมวลผลในระบบแฟ้มข้อมูลนี้ ซึ่งเรียกการประมวลผลแบบใหม่นี้ว่าระบบการประมวลผลฐานข้อมูล

### 2.2.3 ระบบการประมวลผลฐานข้อมูล

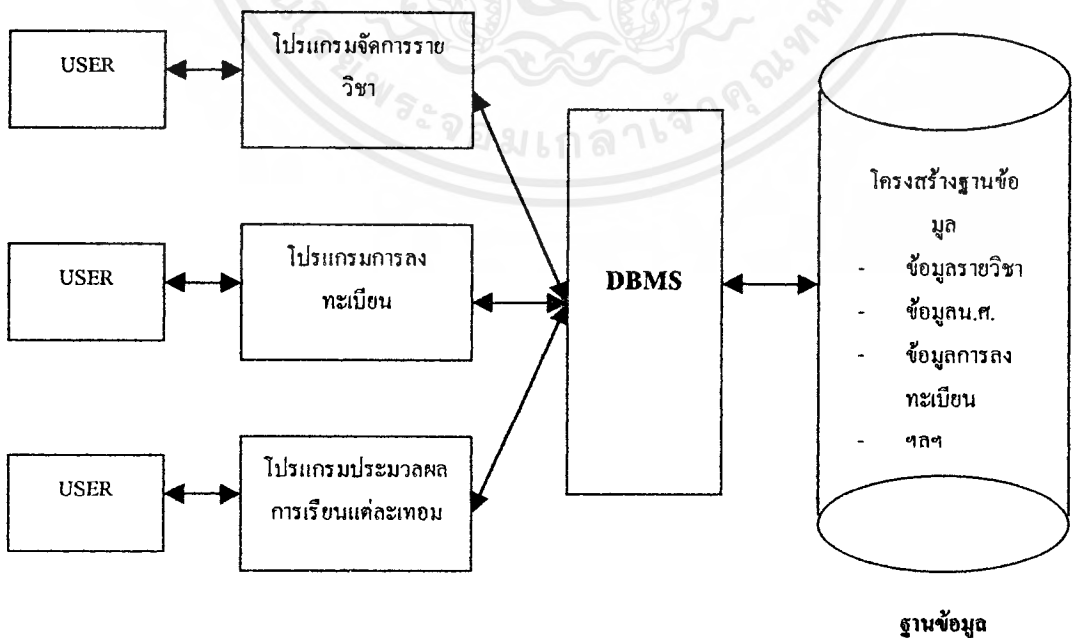
จากข้อจำกัดของระบบการประมวลผลเพิ่มข้อมูลดังกล่าวข้างต้น จึงมีความพยายามคิดหาเทคโนโลยีการประมวลผลแบบใหม่ เพื่อเอาชนะข้อจำกัดดังกล่าว เทคโนโลยีหนึ่งที่ถูกพัฒนาขึ้นมาแทนที่ระบบการประมวลผลเพิ่มข้อมูลได้แก่ ระบบการประมวลผลฐานข้อมูล

คำว่า ฐานข้อมูล โดยทั่วไปจะหมายถึงการเก็บรวบรวมข้อมูลที่มีความสัมพันธ์กันไว้ในที่ที่เดียวกัน

ในระบบการประมวลผลฐานข้อมูล จะมีรูปแบบและวิธีการจัดการข้อมูลที่แตกต่างจากระบบเพิ่มข้อมูลดังรูป 2.2 จะมีองค์ประกอบหนึ่งเพิ่มขึ้นมาจากระบบการประมวลผลเพิ่มข้อมูลได้แก่องค์ประกอบที่เรียกว่า ระบบจัดการฐานข้อมูล หรือ DBMS (Database management System) ซึ่งเป็นโปรแกรมชนิดหนึ่งที่ถูกสร้างขึ้นมาเพื่อแก้ไขข้อบกพร่องของระบบการประมวลผลเพิ่มข้อมูล ในระบบการประมวลผลฐานข้อมูลนี้เพิ่มข้อมูลต่างๆ ที่มีความเกี่ยวข้องกันหรือมีความสัมพันธ์กันจะถูกเก็บอยู่รวมกันในที่ที่เดียว ซึ่งจะช่วยลดความซ้ำซ้อนของข้อมูลทำให้ข้อมูลมีความถูกต้องและทันสมัยอยู่ตลอดเวลา นอกจากนี้โปรแกรมประยุกต์ที่เขียนก็จะไม่ขึ้นกับ โครงสร้างของเพิ่มข้อมูลอีกด้วย

DBMS จะช่วยในการสร้าง เรียกใช้ข้อมูลและปรับปรุงฐานข้อมูล โดยจะทำหน้าที่เสมือนตัวกลางระหว่างผู้ใช้และฐานข้อมูลให้สามารถติดต่อกันได้

ให้สังเกตข้อแตกต่างระหว่างระบบการประมวลผลเพิ่มข้อมูลในรูปที่ 2.2 และระบบการประมวลผลฐานข้อมูลในรูป 2.1 จะเห็นว่าเพิ่มข้อมูลต่าง ๆ ในรูป 2.1 โปรแกรมประยุกต์ที่สร้างขึ้นในระบบการประมวลผลเพิ่มข้อมูล จะมีการอ่านหรือเขียนข้อมูลจากเพิ่มข้อมูลโดยตรง ในขณะที่โปรแกรมประยุกต์ที่สร้างในระบบการประมวลผลฐานข้อมูลนั้นจะต้องออกคำสั่งผ่าน DBMS ก่อน DBMS จึงจะทำหน้าที่อ่านข้อมูลที่ต้องการแล้วส่งต่อไปโปรแกรมประยุกต์อีกทีหนึ่ง



รูปที่ 2.2 ระบบการประมวลผลฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานที่ต้องผ่าน DBMS ทุกครั้งนี้จะทำให้การเขียน โปรแกรมประยุกต์มีความสะดวกยิ่งขึ้น เนื่องจากผู้เขียน โปรแกรมไม่ต้องยุ่งเกี่ยวหรือสนใจว่าในทางกายภาพข้อมูลถูกเก็บอยู่อย่างไรใน ดิสก์ หรือแม้แต่วิธีการ ในการจัดการกับข้อมูลไม่ต้องสนใจว่าใช้วิธีแบบอินเด็กซ์ไฟล์(Indexed file) หรือ แบบอินเด็กซ์ซีควิวเอนซ์ไฟล์(Indexed sequential File)เป็นต้น ผู้ใช้เพียงแต่ออกคำสั่งง่ายๆ ในการเรียกใช้ ข้อมูล เพิ่มข้อมูล ปรับปรุงข้อมูล หรือลบข้อมูลผ่านทาง DBMS แทน

### ข้อดีของการประมวลผลข้อมูลในฐานข้อมูล

ฐานข้อมูลซึ่งจะมีการใช้คำว่า ตาราง(Table) แทนเพิ่มข้อมูลแต่ละเพิ่ม ซึ่งจะได้อีกถึงคุณ ลักษณะของฐานข้อมูลประเภทนี้อีกทีในหัวข้อถัดไป และจะใช้ตัวอย่างฐานข้อมูลเชิงสัมพันธ์ในรูป 2.4 เป็นตัวอย่างประกอบการอธิบายข้อดีของการประมวลผลข้อมูลในฐานข้อมูลที่เหนือกว่าระบบเพิ่มข้อมูล ดังต่อไปนี้

1. ข้อมูลมีการเก็บอยู่ร่วมกันและสามารถใช้ข้อมูลร่วมกัน ได้  
ในระบบฐานข้อมูล ข้อมูลทั้งหมดจะถูกเก็บอยู่ในที่ที่เดียวกันที่เรียกว่า ฐานข้อมูล โปรแกรมประยุกต์ สามารถออกคำสั่งผ่าน DBMS ให้ทำการอ่านข้อมูลจากหลายตารางได้ เช่นจากรูป 2.2 โปรแกรมประมวลผลการเรียนแต่ละเทอม จะสามารถออกคำสั่งง่ายๆ เพื่ออ่านข้อมูลจากทั้งสองตารางออกมา จะเป็นหน้าที่ของ DBMS ที่จะจัดการเชื่อมข้อมูลจากทั้งสองตารางนี้ให้
2. ลดความซ้ำซ้อนของข้อมูล  
ในการประมวลผลฐานข้อมูล ข้อมูลจะมีความซ้ำซ้อนน้อยที่สุด เนื่องจากข้อมูลจะถูกเก็บอยู่เพียงที่เดียวในฐานข้อมูล เช่นข้อมูลรายวิชา ข้อมูลการลงทะเบียน ข้อมูลนักศึกษาและข้อมูลผลการเรียน ประจำเทอม ซึ่งจะเป็นการประหยัดเนื้อที่การใช้งานหน่วยเก็บข้อมูลสำรอง นอกจากนี้ถ้าจะมีการปรับปรุงหรือเปลี่ยนแปลงในตารางนักศึกษาเพียงครั้งเดียวเท่านั้น ดังนั้นจึงเป็นการลดความซ้ำซ้อนของข้อมูลลง ข้อมูลจะมีความถูกต้องไม่มีความขัดแย้งของข้อมูลเกิดขึ้น
3. สามารถหลีกเลี่ยงความขัดแย้งกันของข้อมูลที่อาจเกิดขึ้น ได้  
สืบเนื่องจากผลของข้อ 2 คือการลดความซ้ำซ้อนของข้อมูล จะทำให้สามารถลดความขัดแย้งของข้อมูลที่อาจเกิดขึ้น ได้ด้วย ตัวอย่างเช่นฐานข้อมูลระบบทะเบียนนักศึกษาในรูป 2.4 ชื่อนักศึกษา จะเก็บอยู่ในตารางนักศึกษาเพียงแห่งเดียว ดังนั้นถ้าจะมีการเปลี่ยนชื่อหรือนามสกุลนักศึกษา ก็สามารถแก้ในตารางนักศึกษาเพียงแห่งเดียว

#### 4. การควบคุมความคงสภาพของข้อมูล

ความคงสภาพ(Integrity) จะหมายถึงความถูกต้อง ความคล่องจง ความสมเหตุสมผลหรือความเชื่อถือได้ของข้อมูล ซึ่งนอกจากลักษณะของข้อมูลที่ต้องมีความซ้ำซ้อนกันน้อยที่สุดแล้วความคงสภาพของข้อมูลก็เป็นความสำคัญไม่ยิ่งหย่อนไปกว่ากัน กล่าวคือข้อมูลภายในฐานข้อมูลนั้นควรจะต้องมีความถูกต้องสมเหตุสมผล เช่นข้อมูลเกรดนักศึกษาจะต้องมีค่าไม่เกิน 4.0 หรือการเก็บข้อมูลจำนวน ชม. การทำงานของพนักงานในแต่ละวัน ข้อมูลจำนวนชม.ที่จะเก็บในฐานข้อมูล ไม่เกิน 24 ชม. ดังนั้นถ้ามีการบันทึกจำนวน ชม.เป็น 30 ชม. ในหนึ่งวัน ระบบฐานข้อมูลที่ดีควรจะมีการป้องกันการบันทึกข้อมูลที่ไม่สมเหตุสมผลนี้ โดยจะมี DBMS เป็นตัวควบคุมไม่ให้มีการบันทึกข้อมูลที่ไม่ถูกต้องลง ไปเก็บในฐานข้อมูลได้

อีกตัวอย่างหนึ่งของความคงสภาพเช่น ถ้ามีนักศึกษาลาออกไป ในตารางนักศึกษาจะต้องลบนักศึกษาคนนั้นออก และจำเป็นจะต้องลบนักศึกษาคนนี้ออกไปจากตารางลงทะเบียนและตารางผลการเรียนประจำเทอมด้วย เพื่อให้ฐานข้อมูลมีความคงสภาพของข้อมูลเกิดขึ้น

#### 5. การจัดการข้อมูลในฐานข้อมูลจะทำได้ง่าย

การจัดการกับข้อมูลไม่ว่าจะเป็นการเรียกใช้ข้อมูล การเพิ่มข้อมูล การแก้ไขข้อมูลหรือการลบข้อมูลของตารางใดภายในฐานข้อมูล จะสามารถทำได้ง่าย โดยการออกคำสั่งผ่านไปยัง DBMS ซึ่ง DBMS จะเป็นตัวจัดการข้อมูลภายในฐานข้อมูลให้เอง

#### 6. ความเป็นอิสระระหว่าง โปรแกรมประยุกต์และข้อมูล

โปรแกรมประยุกต์ที่เขียนขึ้นจะไม่ขึ้นกับ โครงสร้างของตารางที่มีการเปลี่ยนแปลงไป เนื่องจากโครงสร้างของตารางต่าง ๆ และตัวข้อมูลในแต่ละตารางจะถูกเก็บอยู่ในฐานข้อมูลทั้งหมด โปรแกรมประยุกต์ไม่จำเป็นต้องเก็บ โครงสร้างของตาราง เช่น การเปลี่ยนแปลงขนาดของเขตข้อมูลในตารางใดภายในฐานข้อมูล ก็ไม่จำเป็นต้องไปทำการแก้ไข โปรแกรมประยุกต์ที่มีการเรียกใช้เขตข้อมูลนั้นซึ่งทำหน้าที่เป็น DBMS ที่จะไปปรับปรุงข้อมูลภายในฐานข้อมูลให้มีโครงสร้างตามที่มีการเปลี่ยนแปลง ยกเว้นว่าถ้ามีการเพิ่มเขตข้อมูล เปลี่ยนชื่อเขตข้อมูลหรือลบเขตข้อมูลใดออกไปจากตาราง โปรแกรมประยุกต์ที่มีการใช้งานเขตข้อมูลนั้นจึงจะถูกแก้ไข

#### 7. การมีผู้ควบคุมระบบเพียงคนเดียว

ผู้ควบคุมระบบฐานข้อมูลจะเรียกว่า DBA (Database Administrator)ซึ่งจะเป็นผู้ควบคุมและบริหารจัดการระบบฐานข้อมูลทั้งหมด โดยจะสามารถจัดการกับ โครงสร้างฐานข้อมูลได้ เช่นการสร้างฐานข้อมูลหรือการเปลี่ยน โครงสร้างข้อมูล รวมถึงจะเป็นผู้กำหนดสิทธิเข้าไปใช้งานฐานข้อมูลว่าจะให้ผู้ใช้ใดเข้าไปใช้ฐานข้อมูลได้บ้าง และสามารถใช้งานฐานข้อมูลระดับใด เช่น ดูข้อมูลได้อย่างเดียว หรือจะสามารถปรับปรุงข้อมูลในฐานข้อมูลไม่ให้สามารถเข้าไปก่อความเสียหายให้กับระบบฐานข้อมูลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ข้อเสียของการประมวลผลข้อมูลในฐานข้อมูล

แม้ว่าการประมวลผลข้อมูลในฐานข้อมูล จะให้ข้อดีหลายประการแต่ก็จะมีข้อเสียอยู่บ้างเช่นกัน ในเรื่องต่าง ๆ ต่อไปนี้

### 1. การใช้งานฐานข้อมูลจะเสียค่าใช้จ่ายค่อนข้างสูง

เนื่องจากราคา DBMS จะมีราคาค่อนข้างแพง นอกจากนี้การใช้งานฐานข้อมูล จะต้องใช้กับเครื่องคอมพิวเตอร์ที่มีประสิทธิภาพค่อนข้างสูง เช่นต้องมีความเร็วสูง มีขนาดหน่วยความจำ และ หน่วยเก็บข้อมูลสำรองความจุสูง เป็นต้น

### 2. การสูญเสียข้อมูลที่อาจเกิดขึ้นได้

เนื่องจากข้อมูลต่าง ๆ ภายในฐานข้อมูลจะถูกเก็บอยู่ในที่ที่เดียวกัน ดังนั้นถ้าดิสก์ที่เก็บฐานข้อมูลนั้นเกิดมีปัญหา อาจทำให้ต้องสูญเสียข้อมูลทั้งหมดในฐานข้อมูลได้ ในขณะที่ระบบเพิ่มข้อมูล จะสามารถเก็บเพิ่มข้อมูลต่าง ๆ แยกกันอยู่ในดิสก์หลายตัวได้ ดังนั้นถ้าดิสก์ตัวใดมีปัญหา เพิ่มข้อมูลในดิสก์ตัวอื่นจะยังคงอยู่ ไม่ต้องสูญเสียเพิ่มข้อมูลทั้งหมด ดังนั้นในระบบฐานข้อมูลที่ดียิ่งจึงต้องมีการป้องกันปัญหาที่อาจเกิดขึ้นนี้ โดยจะต้องมีการสำรองข้อมูลทั้งหมดจากดิสก์ขึ้นเก็บไว้ในเทปแม่เหล็กทุกสิ้นวัน หรือทุกสิ้นเดือน และเก็บไว้ในที่ที่ปลอดภัย เพื่อเป็นการป้องกันข้อมูลจากกรณีที่เกิดปัญหาเกิดขึ้น

#### 2.2.4 ฐานข้อมูล

จากสิ่งที่ได้กล่าวมาข้างต้นจะเห็นว่าระบบการประมวลผลฐานข้อมูลเป็นระบบที่สามารถแก้ไขข้อบกพร่องของระบบการประมวลผลแบบเพิ่มข้อมูลได้ ดังนั้นในปัจจุบันจึงมีหลายหน่วยงานหันมาให้ความสนใจกับระบบฐานข้อมูลกันมาก บางหน่วยงานถึงกับมีการเปลี่ยนแปลงการทำงานจากระบบเดิมคือระบบการประมวลผลเพิ่มข้อมูลมาเป็นระบบการประมวลผลฐานข้อมูลเลยทีเดียว

จากนิยามของฐานข้อมูลที่ว่า ฐานข้อมูลหมายถึงการเก็บรวบรวมข้อมูลที่มีความสัมพันธ์กันไว้ในที่ที่เดียวกัน ซึ่งจะได้อธิบายถึงลักษณะความสัมพันธ์ของข้อมูลภายในหัวข้อถัดไป แต่ก่อนอื่นจะขอกล่าวถึงลำดับขั้นของโครงสร้างข้อมูลพื้นฐานที่มีอยู่ในระบบเพิ่มข้อมูลและระบบฐานข้อมูลก่อนดังนี้

ฐานข้อมูลพื้นฐานที่เล็กที่สุดภายในเพิ่มข้อมูลคือ บิต (Bit : Binary Digit) ซึ่งเป็นหน่วยข้อมูลพื้นฐานที่เก็บอยู่ในหน่วยความจำภายในคอมพิวเตอร์ บิตนี้จะแทนด้วยตัวเลข 1 ตัว ได้แก่ 1 หรือ 0 ใดอย่างหนึ่ง เรียกตัวเลข 1 หรือ 0 นี้ว่าเป็นบิต 1 บิต ข้อมูลซึ่งได้แก่ตัวอักษร (Character) แต่ละตัวเช่น A , B , ..., Z , 0,1,2,...,9 และสัญลักษณ์พิเศษอื่น ๆ เช่น \$ , & , + , - , \* , / ฯลฯ เมื่อจะถูกนำไปเก็บไว้ในคอมพิวเตอร์ จะต้องถูกแปลงให้อยู่ในรูปของบิตหลายบิตที่มาประกอบกัน โดยตัวอักษร 1 ตัว จะแทนด้วยบิต 7 หรือ 8 บิต ตัวอักษรแต่ละตัวจะเรียกได้อีกอย่างว่า ไบท์ (Byte) ตัวอย่างเช่น ตัวอักษร A เมื่อเก็บอยู่ในคอมพิวเตอร์จะเก็บเป็น 1000001 ตัวอักษร B จะเก็บเป็น 1000010 เป็นต้น

ตัวอักษรแต่ละตัวจะถูกนำมาประกอบกันเป็นกลุ่มคำที่มีความหมายมากขึ้น เช่นกลุ่มตัวอักษรที่

ประกอบกันเป็นชื่อหรือนามสกุล กลุ่มตัวเลขที่ประกอบกันเป็นรหัสประจำตัว จะเรียกกลุ่มของตัวอักษร

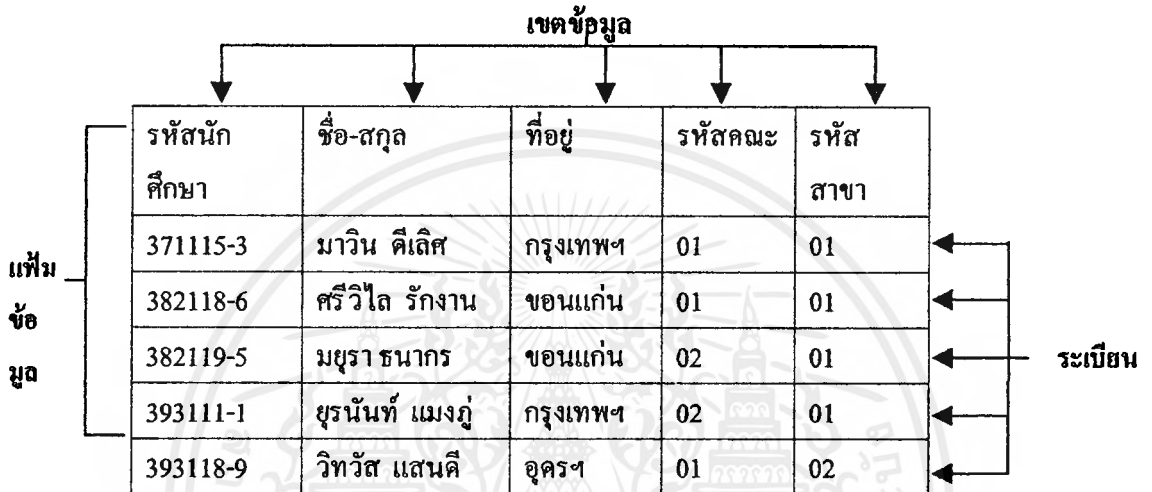
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่รวมกันขึ้นมาถือว่าเป็น **เขตข้อมูลหรือฟิลด์ (Field)** เช่น เขตข้อมูลรหัสนักศึกษา เขตข้อมูลชื่อ เขตข้อมูลที่อยู่เป็นต้น

เมื่อนำเขตข้อมูลหลายเขตข้อมูลมารวมกันจะเรียกว่าเป็น **ระเบียนหรือเรคอร์ด (Record)** เช่น ระเบียนนักศึกษาประกอบด้วยเขตข้อมูลรหัสนักศึกษา ชื่อนักศึกษา รหัสคณะ และรหัสสาขา เป็นต้น

ระเบียนแต่ละระเบียนของข้อมูลชนิดเดียวกันจะสามารถนำมารวมกันเป็น **แฟ้มข้อมูลหรือไฟล์ (File)**



รูปที่ 2.3 แสดงตัวอย่างแฟ้มข้อมูลนักศึกษา

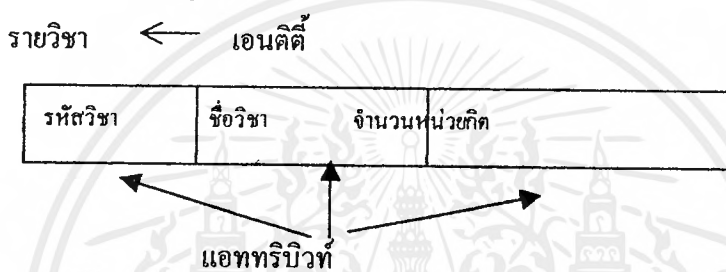
จากคำนิยามของฐานข้อมูลหลายคนอาจจะคิดว่าถ้านำแฟ้มข้อมูลหลายแฟ้มข้อมูลมารวมกันไว้ในที่ที่เดียวกันก็จะกลายเป็นฐานข้อมูลได้ ซึ่งข้อความนี้จะยังไม่ถูกต้องเสียทีเดียว เนื่องจากฐานข้อมูลจะเป็นยิ่งกว่าการเก็บรวบรวมแฟ้มข้อมูลหลายแฟ้มเข้าด้วยกันเท่านั้น แต่ฐานข้อมูลยังจะต้องมีการเก็บคำอธิบายเกี่ยวกับโครงสร้างของฐานข้อมูลที่เรียกว่า **พจนานุกรมข้อมูล (Data Dictionary)** หรืออาจเรียกอีกอย่างว่า **เมตาดेटา (MetaData)**

ถ้าเปรียบเทียบฐานข้อมูล เหมือนกับห้องสมุดของมหาวิทยาลัย ที่มีการเก็บหนังสือเล่มต่าง ๆ หนังสือเหล่านั้นจะเปรียบเสมือนกับข้อมูลที่เก็บอยู่ภายในฐานข้อมูล และในห้องสมุดจะต้องมีการทำบัญชีรายชื่อหนังสือต่าง ๆ เก็บไว้ เพื่อให้บอกรายละเอียดเกี่ยวกับหนังสือแต่ละเล่มว่าใครเป็นผู้แต่ง เก็บอยู่ที่ใดในห้องสมุด บัญชีรายชื่อหนังสือนี้จะเปรียบได้กับพจนานุกรมข้อมูล ซึ่งจะมีหน้าที่อธิบายลักษณะของข้อมูลที่เก็บอยู่ในฐานข้อมูลรวมทั้งความสัมพันธ์ของข้อมูลเช่นระหว่างระเบียนของแฟ้มข้อมูลหนึ่งและแฟ้มข้อมูลอื่นด้วย ซึ่งพจนานุกรมข้อมูลนี้จะถูกเก็บและถูกเรียกใช้งานในระหว่างที่มีการประมวลผลฐานข้อมูล

### 2.2.4.1 เอนทิตี แอททริบิวต์และความสัมพันธ์

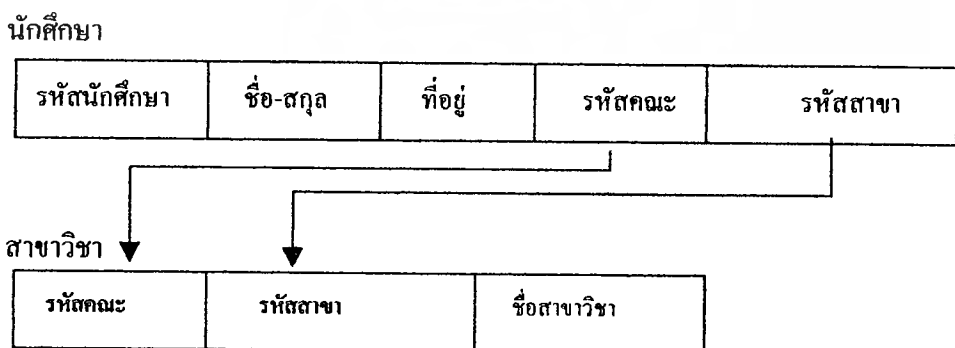
เอนทิตี (Entity) เปรียบเสมือนกับเป็นคำนาม ซึ่งหมายถึงสิ่งต่าง ๆ ที่ผู้ใช้งานฐานข้อมูลจะต้องยุ่งเกี่ยวกับเมื่อมีการออกแบบระบบฐานข้อมูลขึ้น ซึ่งอาจเป็นสิ่งที่ป็นรูปธรรมคือสามารถมองเห็นได้ด้วยตา เช่นเอนทิตีที่เป็นบุคคล สถานที่ หรือสิ่งของ หรืออยู่ในรูปของนายธรรมคือไม่สามารถมองเห็นได้ด้วยตา เช่นเอนทิตีการลงทะเบียน การทำงาน การสั่งซื้อ เป็นต้น ตัวอย่างในระบบการลงทะเบียน จะประกอบด้วยเอนทิตีรายวิชา นักศึกษา การลงทะเบียน ผลการเรียนประจำเทอม สาขาวิชา และคณะ เป็นต้น

แอททริบิวต์ (Attribute) จะเป็นข้อมูลที่แสดงถึงคุณสมบัติของเอนทิตีนั้น ๆ เช่น เอนทิตีรายวิชา จะประกอบด้วยแอททริบิวต์รหัสวิชา ชื่อวิชา และจำนวนหน่วยกิต เป็นต้น



รูปที่ 2.4 แสดงตัวอย่างของเอนทิตี และ แอททริบิวต์

ดังนั้นถ้าจะเปรียบไปแล้วจะเห็นว่าเอนทิตีก็เหมือนกับเพิ่มข้อมูล แอททริบิวต์ก็เหมือนกับเขตข้อมูลนั่นเอง ส่วน ความสัมพันธ์ นั้นจะหมายถึงความสัมพันธ์ระหว่างเอนทิตีสองเอนทิตี เช่นความสัมพันธ์ระหว่างเอนทิตีนักศึกษา และเอนทิตีสาขาวิชา เป็นความสัมพันธ์ในลักษณะที่ว่านักศึกษาคนนี้เรียนอยู่ในสาขาวิชาใดของคณะใด การสร้างความสัมพันธ์นี้จะทำได้โดยการใส่รหัสสาขา และรหัสคณะ ซึ่งรวมกันเป็น คีย์หลัก (Primary Key) ในเอนทิตีสาขาวิชา ให้เป็นแอททริบิวต์ที่อยู่ในเอนทิตีนักศึกษาด้วย



รูปที่ 2.5 แสดงความสัมพันธ์ระหว่างเอนทิตีนักศึกษาและเอนทิตีสาขาวิชาโดยอาศัยรหัสสาขาและรหัสคณะเป็นตัวเชื่อม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นค่าของรหัสคณะและรหัสสาขาในระเบียบนักศึกษาแต่ละคนจะเป็นตัวกำหนดว่า นักศึกษาคนไหนเรียนอยู่คณะใด สาขาวิชาใด หรือในทางกลับกันถ้าอยากทราบว่ารหัสคณะ 01 สาขาวิชา 01 มีนักศึกษาคณะใดเรียนอยู่บ้าง ก็ทำได้โดยดูจากเอนติตี้นักศึกษา เลือกเฉพาะระเบียบนักศึกษาที่มีรหัสคณะเป็น 01 และรหัสสาขาเป็น 01 เป็นต้น ซึ่งค่ารหัสคณะและรหัสสาขาในเอนติตี้สาขาวิชานี้จะร่วมกันเป็นคีย์หลักในการเข้าถึงข้อมูลที่ต้องการ ในเอนติตี้สาขาวิชา เช่น การอ้างอิงรหัสคณะ 01 และรหัสสาขา 01 จะหมายถึงการอ้างอิงระเบียบสาขาวิทยาการคอมพิวเตอร์นั่นเอง

### 2.2.5 ความสัมพันธ์ระหว่างเอนติตี้

ดังได้กล่าวแล้วว่าเอนติตี้ภายในฐานข้อมูลอาจมีความสัมพันธ์กัน โดยทั่วไปความสัมพันธ์ระหว่างเอนติตี้ จะมีอยู่ 3 ประเภทใหญ่ ๆ ดังนี้

#### 2.2.5.1 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One-to-One Relationship)

เป็นลักษณะความสัมพันธ์ที่มีระเบียบเพียง 1 ระเบียบในเอนติตี้ A ที่มีความสัมพันธ์กับระเบียบเพียง 1 ระเบียบในเอนติตี้ B และในทางกลับกันระเบียบเพียง 1 ระเบียบในเอนติตี้ B ก็จะมีความสัมพันธ์กับระเบียบเพียง 1 ระเบียบในเอนติตี้ A

ตัวอย่างของความสัมพันธ์แบบนี้เช่น ถ้าบริษัทขายรถยนต์แห่งหนึ่ง ได้ตั้งข้อกำหนดว่า ลูกค้าที่มาซื้อรถแต่ละคนถูกกำหนดให้สามารถซื้อรถได้เพียงคนละ 1 คัน เท่านั้น และรถยนต์แต่ละคันจะมีลูกค้าเป็นเจ้าของเพียง 1 คนเท่านั้น บริษัทอาจมีการแยกเก็บข้อมูลไว้เป็น 2 ตาราง โดยตารางแรกจะเป็นตารางลูกค้าที่เก็บประวัติลูกค้าแต่ละคน ซึ่งประกอบด้วยรหัสลูกค้า ชื่อลูกค้า ที่อยู่ ฯลฯ ลูกค้าแต่ละคนจะมีระเบียบเพียงระเบียบเดียวในตารางลูกค้า สำหรับตารางที่สองจะเป็นตารางการขายรถยนต์ ซึ่งจะประกอบด้วยรหัสรถยนต์แต่ละคัน รหัสลูกค้าที่ซื้อรถคันนั้น ฯลฯ (ซึ่งรหัสลูกค้านี้จะเป็นคีย์หลักที่อยู่ในตารางลูกค้า) และรถแต่ละคันก็จะมีเพียงระเบียบเดียวเท่านั้นในตารางการขายรถยนต์ ดังจะเห็นจากตัวอย่างว่าลูกค้ารหัส 233-1111 จะซื้อรถยนต์ที่มีรหัส C 111-1256 ได้เพียง 1 คันเท่านั้น และในทางกลับกันรถยนต์ที่มีรหัส C111-1256 ก็จะสามารถขายให้กับลูกค้าเพียงคนเดียวเท่านั้นคือลูกค้าที่มีรหัสเป็น 233-1111 เป็นต้น (ตัวอย่างข้อมูลที่แสดงอาจจะมีเขตข้อมูลอื่นอีกก็ได้ แต่จะยกตัวอย่างให้เห็นเฉพาะเขตข้อมูลที่จำเป็นต้องใช้เท่านั้น)

### 2.2.5.2 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One-to-Many Relationship)

หมายถึงความสัมพันธ์ที่ระเบียบหนึ่งระเบียบในเอนทิตี A มีความสัมพันธ์กับระเบียบหลายระเบียบในเอนทิตี B และในทางกลับกันหลายระเบียบในเอนทิตี B จะมีความสัมพันธ์กับระเบียบเพียงหนึ่ง ระเบียบในเอนทิตี A

ตัวอย่างที่แสดงความสัมพันธ์แบบหนึ่งต่อกลุ่มเช่น ลูกค้านั่งหนึ่ง ๆ สามารถเลือกซื้อสินค้าได้หลายอย่าง แต่สินค้าแต่ละอย่างจะต้องเป็นของลูกค้าเพียงคนเดียวเท่านั้น หรือนักศึกษาแต่ละคนสามารถยืมหนังสือจากห้องสมุดได้หลายเล่ม แต่หนังสือแต่ละเล่มจะมีนักศึกษาเพียงคนเดียวเท่านั้นที่ยืมได้ หรืออาจารย์แต่ละคนอาจเป็นที่ปรึกษาให้กับนักศึกษาหลายคน แต่นักศึกษาแต่ละคนจะต้องมีอาจารย์ที่ปรึกษาได้เพียงคนเดียวเป็นต้น

จากตัวอย่างการซื้อสินค้าของลูกค้าในธุรกิจการขายส่ง จะพบว่าลูกค้าแต่ละรายจะสามารถซื้อสินค้าได้หลายชนิด แต่สินค้าแต่ละชนิดจะต้องเป็นของลูกค้าเพียงคนเดียว จะสามารถสร้างตารางได้หลายตารางดังต่อไปนี้ ตารางแรกเป็นตารางลูกค้าที่เก็บรายละเอียดเกี่ยวกับรหัสลูกค้า ชื่อลูกค้า ที่อยู่ ฯลฯ ตารางที่สองจะเป็นตารางการขายสินค้าซึ่งเก็บข้อมูลเกี่ยวกับรหัสลูกค้าที่สั่ง รหัสสินค้า วันเดือนปีที่สั่งสินค้า จำนวนสินค้าที่สั่ง ฯลฯ และอาจมีตารางอื่น ๆ เช่นตารางสินค้าซึ่งจะเก็บรายละเอียดเกี่ยวกับสินค้าได้แก่ รหัสสินค้า ชื่อสินค้า จำนวนที่มีอยู่ ฯลฯ

ลูกค้า		การขายสินค้า		
รหัสลูกค้า	ชื่อลูกค้า	รหัสลูกค้า	รหัสสินค้า	วันเดือนปีที่ตั้ง
123336	สมจิตรคลังหนังสือ	433225	2560	1/2/96
322322	ร้านหนังสือดอกฝ้าย	433225	3450	1/2/96
433225	ร้านค้าเพ็ญจันทร์	322322	4550	12/3/96
678355	ร้านค้าวิสตินเตอร์	123336	1203	12/3/96
		123336	2540	12/3/96
		678355	5560	12/3/96

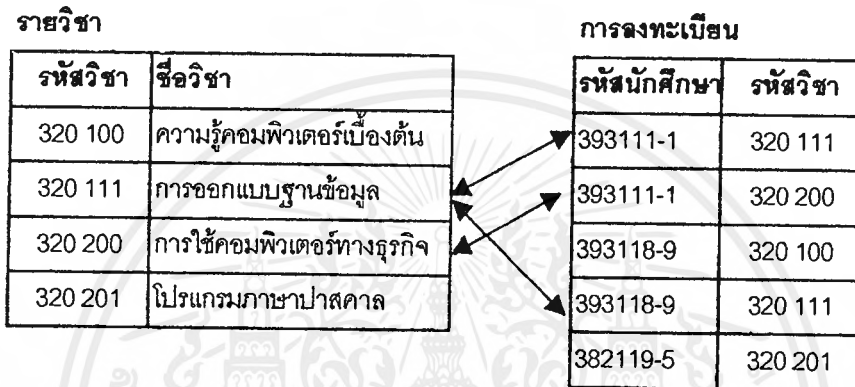
รูปที่ 2.6 แสดงตัวอย่างความสัมพันธ์แบบหนึ่งต่อกลุ่ม

จากตัวอย่างดังรูป 2.6 จะพบว่าในตารางการขายสินค้าลูกค้าที่มีรหัส 123336 จะซื้อสินค้า 2 ชนิดได้แก่ สินค้าที่มีรหัสเป็น 1203 และ 2540 แต่สินค้าแต่ละชนิดจะต้องถูกซื้อจากลูกค้าเพียงคนเดียว ซึ่งเป็นตัวอย่างของความสัมพันธ์แบบหนึ่งต่อกลุ่มนั่นเอง

### 2.2.5.3 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many-to-Many Relationships)

เป็นความสัมพันธ์ที่แต่ละระเบียนในเอนทิตี A จะมีความสัมพันธ์กับระเบียนหลายระเบียนในเอนทิตี B และในทางกลับกันแต่ละระเบียนในเอนทิตี B จะมีความสัมพันธ์กับระเบียนหลายระเบียนที่อยู่ในเอนทิตี A

ตัวอย่างของความสัมพัธ์แบบนี้เช่น การเลือกวิชาของนักศึกษา จะพบว่า นักศึกษาแต่ละคนจะสามารถเลือกเรียนวิชาได้หลายวิชา และในทางกลับกันแต่ละวิชาที่สามารถมีนักศึกษาเรียนได้หลายคน ดังรูป 2.7



รูปที่ 2.7 แสดงตัวอย่างความสัมพันธ์แบบกลุ่มต่อกลุ่ม

จากตัวอย่างในรูป 2.7 จะพบว่าวิชา 320 111 จะสามารถมีนักศึกษาเลือกเรียนได้มากกว่า 1 คนได้แก่ นักศึกษาที่มีรหัสเป็น 393111-1 และ 393118-9 และในทางกลับกันนักศึกษาที่มีรหัส 393111-1 จะสามารถเลือกเรียนวิชาได้มากกว่าหนึ่งวิชาได้แก่วิชาที่มีรหัสเป็น 320 111 และ 320 200 เป็นต้น

ในตัวอย่างการซื้อสินค้าของลูกค้า ลูกค้าแต่ละคนสามารถเลือกซื้อสินค้าได้มากกว่าหนึ่งอย่าง แต่ถ้าพิจารณาสินค้าแต่ละชิ้น จะพบว่าจะสามารถขายให้ลูกค้าได้เพียงหนึ่งคนเท่านั้น ซึ่งเป็นลักษณะของความสัมพัธ์แบบหนึ่งต่อกลุ่ม แต่ถ้าพิจารณาว่าสินค้าแต่ละชนิด อาจมีหลายชิ้นที่เป็นชนิดเดียวกันคือเหมือนกัน ดังนั้นจึงอาจถือว่าสินค้าแต่ละชนิด จะสามารถขายให้ลูกค้าได้หลายคนด้วย ดังนั้นถ้าพิจารณาจากชนิดของสินค้าก็จะเป็นความสัมพันธ์แบบกลุ่มต่อกลุ่มได้

### 2.2.6 ประเภทของฐานข้อมูล

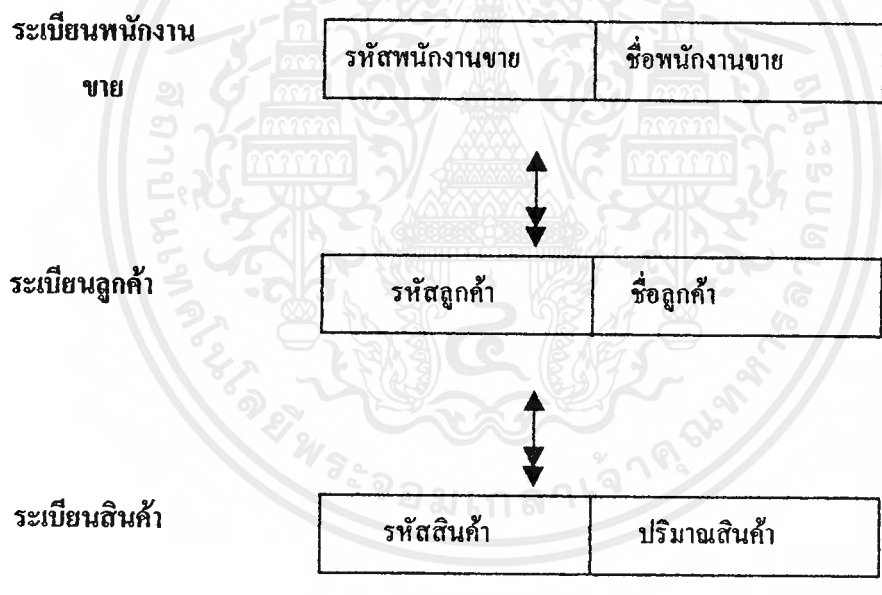
ข้อมูลในฐานข้อมูล โดยทั่วไปจะถูกสร้างให้มีโครงสร้างที่ง่ายต่อความเข้าใจและการใช้งานของผู้ใช้โดยทั่วไปแล้วฐานข้อมูลที่มีใช้อยู่ในปัจจุบันจะมีโครงสร้าง 3 แบบด้วยกัน คือ ฐานข้อมูลแบบลำดับขั้น (Hierarchical Database) ฐานข้อมูลแบบเครือข่าย (Network Database) และฐานข้อมูลแบบเชิงสัมพันธ์ (Relational Database)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

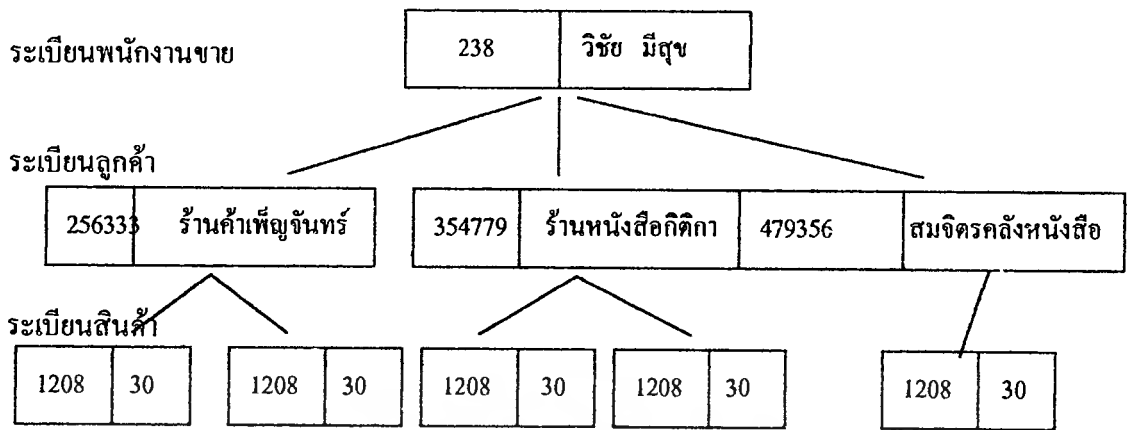
2.6.6.1 ฐานข้อมูลแบบลำดับชั้น (Hierarchical Database)

เป็นลักษณะของฐานข้อมูลที่มีความสัมพันธ์ของข้อมูลเป็นแบบหนึ่งต่อหนึ่ง หรือ หนึ่งต่อกลุ่ม แต่จะ ไม่มีความสัมพันธ์แบบกลุ่มต่อกลุ่มในฐานข้อมูลแบบนี้

ลักษณะโครงสร้างของฐานข้อมูลแบบลำดับชั้นนี้ จะมีลักษณะคล้ายต้นไม้ที่คว่ำหัวลง จึงอาจเรียกโครงสร้างฐานข้อมูลแบบนี้ได้อีกแบบว่าเป็น โครงสร้างแบบต้นไม้ (Tree Structure) โดยจะมีระเบียบที่อยู่แถวบนซึ่งจะเรียกว่าเป็น ระเบียบพ่อแม่ (Parent record) ระเบียบในแถวถัดลงมาจะเรียกว่า ระเบียบลูก (Child record) ซึ่งระเบียบพ่อแม่จะสามารถมีระเบียบลูกได้มากกว่าหนึ่งระเบียบ แต่ระเบียบลูกแต่ละระเบียบจะมีระเบียบพ่อแม่เพียงหนึ่งระเบียบเท่านั้น ตัวอย่างของฐานข้อมูลแบบนี้เช่นการขายสินค้าของพนักงานให้แก่ลูกค้าแต่ละคน จะพบว่าพนักงานขายแต่ละคน จะมีลูกค้าได้หลายคน และลูกค้าแต่ละคนก็อาจซื้อสินค้าได้มากกว่า 1 อย่างขึ้นไป จะสามารถแสดงความสัมพันธ์ของระเบียบพนักงานขาย และระเบียบลูกค้า และความสัมพันธ์ระหว่างระเบียบลูกค้าและระเบียบสินค้า ของฐานข้อมูลแบบลำดับชั้นนี้ได้ โดยใช้ลูกศรดังรูป 2.8 ก. โดยหัวลูกศรคู่จะแสดงความสัมพันธ์ด้านกลุ่ม และสำหรับการค้นหาข้อมูลในระเบียบใดก็ค้นหาตามแนวลูกศรที่เชื่อม โยงความสัมพันธ์ดังกล่าวดังรูป 2.8 ข.



รูปที่ 2.8 ก. แสดงโครงสร้างฐานข้อมูลแบบลำดับชั้น



รูปที่ 2.8 ข. แสดงตัวอย่างข้อมูลในฐานข้อมูลแบบลำดับชั้น

จากตัวอย่างข้างต้นจะเห็นว่า ลูกค้าแต่ละคนจะไม่สามารถได้รับบริการจากพนักงานขายมากกว่าหนึ่งคนได้ เนื่องจากลูกค้าแต่ละคนถือว่าเป็นระเบียบลูก และพนักงานขายจะถือว่าเป็นระเบียบพ่อแม่ของลูกค้า สินค้าแต่ละชนิดก็จะถูกซื้อ โดยลูกค้าเพียงคนเดียวเท่านั้น เนื่องจากสินค้าแต่ละชนิดจะเป็นระเบียบลูกของระเบียบลูกค้าเป็นต้น

#### 2.2.6.2 ฐานข้อมูลแบบเชิงสัมพันธ์ (Relational Database)

ฐานข้อมูลเชิงสัมพันธ์เป็นฐานข้อมูลที่มีความนิยมใช้กันมากในปัจจุบัน ซึ่งจะสามารถใช้งานได้กับเครื่องคอมพิวเตอร์ทุกระดับตั้งแต่ไมโครคอมพิวเตอร์ จนกระทั่งถึงเมนเฟรมคอมพิวเตอร์ ฐานข้อมูลแบบนี้จะมีโครงสร้างข้อมูลต่างจากฐานข้อมูลสองแบบแรก กล่าวคือ ข้อมูลจะถูกเก็บอยู่ในรูปแบบของตาราง (table) ซึ่งภายในตารางก็จะแบ่งออกเป็น แถว (row) และ คอลัมน์ (column) แต่ละตารางจะมีจำนวนแถว ได้หลายแถวและจำนวน คอลัมน์ได้หลายคอลัมน์ แถวแต่ละแถวจะสามารถเรียกได้อีกชื่อว่า ระเบียบหรือเรคอร์ด (record) คอลัมน์แต่ละคอลัมน์เรียกได้อีกชื่อว่าเขตข้อมูลหรือฟิลด์ (field)

นอกจากนี้ตารางแต่ละตารางยังสามารถเรียกได้อีกชื่อว่า รีเลชัน (Relation) แถวแต่ละแถวภายในตารางยังอาจเรียกว่า ทัพเพิล (tuple) และคอลัมน์แต่ละคอลัมน์อาจถูกเรียกว่า แอททริบิวท์ (attribute)

แอมทริบิวท์ (คอลัมน์ , เขตข้อมูลหรือฟิลด์)

รหัสสินค้า	ชื่อสินค้า	ราคา/หน่วย	จำนวนสินค้าที่มีอยู่
1208	สมุดปกอ่อน	10.00	200
2344	สมุดปกแข็ง	20.00	200
2590	ยางลบดินสอ	6.00	100
2900	ยางลบปากกา	8.00	100
3010	ไม้บรรทัด	3.00	100
3201	น้ำยาลบคำผิด	45.00	250

ริเลชัน (ตาราง , เพิ่มข้อมูล หรือไฟล์)

ทัฟเพิล (แถว ระเบียบ หรือ เรกอร์ด)

รูปที่ 2.9 แสดงโครงสร้างริเลชัน

## 2.2.7 ระบบฐานข้อมูล (Database Systems)

หลังจากที่ได้รู้จักความหมายของฐานข้อมูล และข้อดีของการประมวลผลข้อมูลในฐานข้อมูลแล้วต่อไปจะกล่าวถึงระบบฐานข้อมูล ซึ่งระบบฐานข้อมูลที่สมบูรณ์จะต้องประกอบด้วยองค์ประกอบ 4 องค์ประกอบหลักดังต่อไปนี้

- ฮาร์ดแวร์ (Hardware)
- ซอฟต์แวร์ (Software)
- ข้อมูล (Data)
- บุคลากร (People)

### 2.2.7.1 องค์ประกอบทางด้านฮาร์ดแวร์

ได้แก่อุปกรณ์ต่าง ๆ ทางคอมพิวเตอร์ ซึ่งเป็นองค์ประกอบที่สำคัญขององค์ประกอบหนึ่งในระบบฐานข้อมูล เนื่องจากฐานข้อมูลจะต้องใช้อุปกรณ์ทางคอมพิวเตอร์เพื่อเก็บข้อมูลและการประมวลผลข้อมูล อุปกรณ์ทางคอมพิวเตอร์อาจประกอบด้วยเครื่องคอมพิวเตอร์ตั้งแต่หนึ่งตัวขึ้นไป หน่วยเก็บข้อมูลสำรองเช่นจานแม่เหล็กหรือดิสก์ เทปแม่เหล็ก อุปกรณ์ขั้วดิสก์ อุปกรณ์ขั้วเทปแม่เหล็ก หน่วยนำข้อมูลเข้า เช่น เทอร์มินัลซึ่งประกอบด้วยแป้นพิมพ์และจอภาพ หน่วยนำข้อมูลออกเช่นเครื่องพิมพ์ นอกจากนี้ยังต้องมีอุปกรณ์การสื่อสารเพื่อเชื่อมโยงอุปกรณ์ทางคอมพิวเตอร์หลาย ๆ เครื่องให้สามารถแลกเปลี่ยนข้อมูลกันได้ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องคอมพิวเตอร์ที่จะใช้เป็นอุปกรณ์สำหรับประมวลผลข้อมูลในฐานะข้อมูลนั้น สามารถเป็นได้ตั้งแต่เครื่องเมนเฟรมคอมพิวเตอร์ มินิคอมพิวเตอร์ หรือ ไมโครคอมพิวเตอร์ ซึ่งเป็นเป็นเครื่องระดับเมนเฟรมคอมพิวเตอร์หรือมินิคอมพิวเตอร์ จะสามารถใช้ต่อกับเทอร์มินัลหลายตัว เพื่อให้ผู้ใช้งานฐานข้อมูลหลายคน ทำการดึงข้อมูลหรือปรับปรุงข้อมูลภายในฐานข้อมูลเดียวกันพร้อมกันได้ ซึ่งเป็นลักษณะของการทำงานแบบมัลติยูสเซอร์ (Multiusers)

สำหรับการประมวลผลฐานข้อมูลในเครื่องระดับไมโครคอมพิวเตอร์ สามารถทำการประมวลผลได้ 2 แบบ แบบแรกจะเป็นการประมวลผลฐานข้อมูลในเครื่องไมโครคอมพิวเตอร์เพียงเครื่องเดียว โดยมีผู้ใช้งานได้เพียงคนเดียวเท่านั้น (Single user) สามารถดึงข้อมูลหรือปรับปรุงข้อมูลภายในฐานข้อมูลได้

สำหรับการประมวลผลฐานข้อมูลในเครื่องระดับไมโครคอมพิวเตอร์แบบที่สอง จะเป็นการนำไมโครคอมพิวเตอร์หลายตัวมาเชื่อมต่อกันในลักษณะของเครือข่ายแบบใกล้ (Local Area Network หรือ LAN) ซึ่งเป็นรูปแบบของระบบเครือข่ายแบบมีลูกข่าย/แม่ข่าย (Client/Server Network) โดยจะมีการเก็บฐานข้อมูลอยู่ที่เครื่องแม่ข่าย การประมวลผลต่าง ๆ จะกระทำที่เครื่องแม่ข่าย (Server) สำหรับเครื่องลูกข่าย (Client) จะมีหน้าที่ดึงข้อมูลหรือส่งข้อมูลเข้ามาปรับปรุงในเครื่องแม่ข่าย หรือคอยรับผลลัพธ์จากการประมวลผลจากเครื่องแม่ข่าย ดังนั้นการประมวลผลแบบนี้จึงเป็นการเปิดโอกาสให้ผู้ใช้งานหลายคนสามารถใช้งานฐานข้อมูลร่วมกันได้ ซึ่งเป็นลักษณะของมัลติยูสเซอร์ (Multiusers) นั่นเอง

โดยส่วนใหญ่แล้วเครื่องแม่ข่าย ควรเป็นเครื่องไมโครคอมพิวเตอร์ที่มีความสามารถสูง หรือเป็นเครื่องระดับมินิคอมพิวเตอร์หรือเมนเฟรม หรืออาจเป็นเครื่องคอมพิวเตอร์ใด ๆ ที่มีตัวประมวลผลมากกว่าหนึ่งตัวก็ได้

การเก็บข้อมูลภายในฐานข้อมูลส่วนใหญ่จะเก็บอยู่ในดิสก์ เนื่องจากดิสก์มีคุณสมบัติการเข้าถึงข้อมูลเป็นแบบสุ่ม (Random Access) ซึ่งจะสามารถอ่านข้อมูลได้อย่างรวดเร็ว สำหรับเทปแม่เหล็กมักใช้เป็นอุปกรณ์สำรอง (Backup) ข้อมูลจากดิสก์ขึ้นมาเก็บไว้ เพื่อเป็นการป้องกันข้อมูลจากกรณีที่เกิดปัญหา ซึ่งอาจทำให้ต้องสูญเสียข้อมูลทั้งหมดได้

ระบบฐานข้อมูลที่มีประสิทธิภาพดีนั้น เครื่องคอมพิวเตอร์ที่จะใช้เก็บและประมวลผลฐานข้อมูลควรเป็นเครื่องที่มีประสิทธิภาพสูง เช่นควรมีหน่วยความจำหลักและหน่วยเก็บข้อมูลสำรองที่มีความจุสูงเพื่อใช้เก็บข้อมูลที่อาจมีปริมาณมาก และนอกจากนี้ควรมีหน่วยประมวลผลกลางหรือ ซีพียู (CPU) ที่มีความเร็วในการทำงานสูง เพื่อรองรับการทำงานจากผู้ใช้หลายคน ที่อาจมีการอ่านข้อมูลหรือปรับปรุงข้อมูลพร้อมกันในเวลาเดียวกันได้อย่างรวดเร็ว

### 2.2.7.2 องค์ประกอบทางด้านซอฟต์แวร์

นอกจากองค์ประกอบทางด้านฮาร์ดแวร์แล้ว องค์ประกอบถัดไปของระบบฐานข้อมูลได้แก่องค์ประกอบด้านซอฟต์แวร์ ซึ่งจะแบ่งซอฟต์แวร์ที่ใช้ในระบบฐานข้อมูลได้เป็น 2 แบบ

- ซอฟต์แวร์ประยุกต์ (Application Software)
- ระบบจัดการฐานข้อมูล (Database Management System หรือ DBMS)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## - ซอฟต์แวร์ประยุกต์

เป็นซอฟต์แวร์หรือโปรแกรมที่ถูกเขียนขึ้นโดยใช้ภาษาระดับสูงเช่นภาษา COBOL หรือ C หรือใช้ภาษาชั้นสูงมากที่เรียกว่าภาษาในยุคที่ 4 (Fourth-Generation Language) โดยนักเขียนโปรแกรม (Programmers) เพื่อใช้ทำงานในเรื่องใดเรื่องหนึ่งโดยเฉพาะ เช่น โปรแกรมระบบสินค้าคงคลัง โปรแกรมการสั่งซื้อ เป็นต้น

ซอฟต์แวร์ประยุกต์นี้จะสารณใช้คำสั่งที่มีอยู่ DBMS ในการดึงข้อมูลหรือจัดการกับข้อมูลภายในฐานข้อมูล เพื่อประมวลผลหรือนำข้อมูลมาออกรายงานตามความต้องการได้

## - ระบบจัดการฐานข้อมูล

สามารถเรียกได้อีกอย่างว่า DBMS เป็นซอฟต์แวร์ระบบชนิดหนึ่งคล้ายกับซอฟต์แวร์ระบบปฏิบัติการทั่วไป มีหน้าที่ให้บริการแก่ผู้ใช้งานฐานข้อมูล และผู้เขียนโปรแกรม ในการจัดการกับข้อมูลใดๆ ภายในฐานข้อมูล

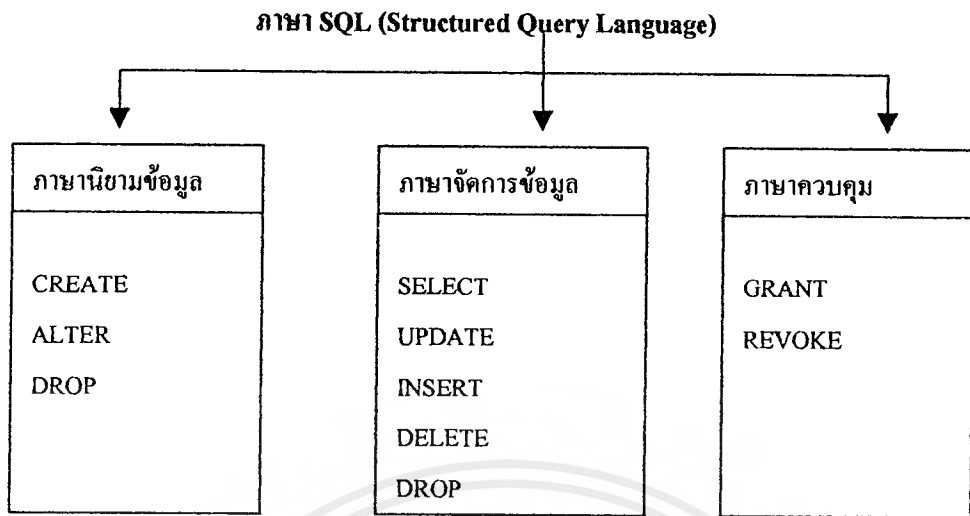
ในปัจจุบันได้มีการพัฒนา DBMS ออกมามากมายเพื่อใช้งานได้กับเครื่องตั้งแต่ไมโครคอมพิวเตอร์ จนถึงระดับเมนเฟรม ซึ่ง DBMS แต่ละตัวอาจมีคุณสมบัติของการทำงานที่แตกต่าง ๆ กัน ดังนั้นการจะพิจารณาว่าจะเลือกใช้ DBMS ตัวใดจึงต้องพิจารณาจากคุณสมบัติของ DBMS แต่ละตัวว่ามีความสามารถทำงานในสิ่งที่เราต้องการได้หรือไม่ อีกทั้งเรื่องราคาก็เป็นเรื่องสำคัญ เนื่องจากราคาของ DBMS แต่ละตัวจะไม่เท่ากัน DBMS ที่มีความสามารถมากก็จะมีราคาแพงมากขึ้น นอกจากนี้ก็ยังคงต้องพิจารณาถึงความเข้ากันได้กับฮาร์ดแวร์ และซอฟต์แวร์ระบบปฏิบัติการ (OS) ที่เรามีอยู่ด้วย

ก่อนที่จะกล่าวถึงความสามารถโดยทั่วไปของ DBMS จะกล่าวถึงส่วนประกอบของ DBMS ซึ่งเป็นส่วนประกอบหลักที่จะพบใน DBMS หลายตัว ดังนี้

## ส่วนประกอบของ DBMS

### 1. ภาษา SQL (Structured Query Language)

เป็นภาษาที่มีรูปแบบเป็นภาษาอังกฤษ ง่ายต่อการเรียนรู้และการเขียนโปรแกรม ซึ่งเป็นภาษาที่มีอยู่ใน DBMS หลายตัว มีความสามารถใช้นิยามโครงสร้างตารางภายในฐานข้อมูล การจัดการข้อมูล รวมไปถึงการควบคุมสิทธิการใช้งานฐานข้อมูล SQL จะประกอบด้วยภาษา 3 รูปแบบด้วยกัน แต่ละแบบก็จะมีหน้าที่เฉพาะแตกต่างกันไปดังต่อไปนี้



รูปที่ 2.10 แสดงองค์ประกอบของภาษา 3 รูปแบบของภาษา SQL

- **ภาษาสำหรับนิยามข้อมูล (Data Definition Language – DDL)**

เป็นภาษาที่ใช้นิยามโครงสร้างของฐานข้อมูล เพื่อทำการสร้าง เปลี่ยนแปลงหรือยกเลิกโครงสร้างของฐานข้อมูลตามที่ได้ออกแบบไว้ ซึ่งโครงสร้างของฐานข้อมูลนี้จะสามารถเรียกได้อีกชื่อว่า สคีมา (Schema) ดังนั้น DDL จึงเป็นภาษาที่ใช้ในการสร้างสกีมานั่นเอง ตัวอย่างเช่นจะเป็นการกำหนดว่าฐานข้อมูลที่สร้างมีชื่อว่าอะไร มีโครงสร้างประกอบด้วยตารางที่ชื่ออะไรบ้าง แต่ละตารางประกอบด้วยเขตข้อมูลใดบ้าง เขตข้อมูลแต่ละตัวมีประเภทของข้อมูลเป็นอะไร มีความกว้างของข้อมูลเท่าใด แต่ละตารางจะมีการอินเด็กซ์ (Index) ช่วยในการค้นหาข้อมูลหรือไม่ ถ้ามีจะใช้เขตข้อมูลใดบ้างที่เป็นคีย์ เป็นต้น

ตัวอย่างของภาษา DDL เช่น

- ◆ คำสั่งการสร้าง (CREATE) ได้แก่การสร้างตารางและการสร้างดัชนี
- ◆ คำสั่งเปลี่ยนโครงสร้างตาราง (ALTER)
- ◆ คำสั่งยกเลิก (DROP) ได้แก่การยกเลิกโครงสร้างตาราง และการยกเลิกโครงสร้างดัชนี
- ◆

- **ภาษาสำหรับการจัดการฐานข้อมูล (Data Manipulation Language-DML)**

องค์ประกอบของภาษาในรูปแบบที่ 2 ของภาษา SQL ซึ่ง DML เป็นภาษาที่ใช้จัดการข้อมูลภายในตารางของฐานข้อมูล ตัวอย่างของภาษา DML เช่น

- ◆ คำสั่งการเรียกค้นระเบียบข้อมูล (SELECT)
- ◆ คำสั่งการเพิ่มระเบียบข้อมูล (INSERT)
- ◆ คำสั่งการปรับปรุงระเบียบข้อมูล (UPDATE)
- ◆ คำสั่งลบระเบียบข้อมูล (DELETE)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบของภาษาที่เป็น DML นี้จะถูกจัดอยู่ในกลุ่มของภาษาในยุคที่ 4 ที่มีความง่ายต่อการเขียนและการทำความเข้าใจมากกว่าภาษาในยุคที่ 3 เช่น COBOL หรือ C

ดังนั้นจะเห็นว่า DDL จะเป็นภาษาที่ใช้กำหนดโครงสร้างของฐานข้อมูล แต่ DML จะเป็นภาษาที่ใช้จัดการกับข้อมูลที่อยู่ภายในโครงสร้างนั้น

DBMS บางตัวจะมีภาษาจัดการข้อมูลเป็นของตนเองเช่น dBASE IV FoxPro R:BASE ฯลฯ แต่ก็มีที่ปรับปรุงเพิ่มเติม SQL เข้าไปด้วยในเวอร์ชันปัจจุบัน และมี DBMS หลายตัวจะมีภาษาที่เรียกว่า QBE (Query by Example) และ QUEL (Query Language) รวมอยู่ด้วยซึ่งเป็นรูปแบบของภาษาการเรียกค้นข้อมูลได้อย่างง่ายอีกแบบหนึ่ง นอกจากนี้ DBMS หลายตัวอนุญาตให้ซอฟต์แวร์ประยุกต์บางตัวที่เขียนด้วยภาษาในยุคที่ 3 เช่น COBOL สามารถใช้คำสั่งการจัดการข้อมูลที่มีอยู่ใน SQL เขียนร่วมอยู่ในภาษา ยุคที่ 3 นั้นได้ (Embedded SQL) เพื่อให้การเขียนโปรแกรมที่เกี่ยวกับการจัดการข้อมูลทำได้ง่ายขึ้น

- ภาษาควบคุม (Control Language –CU)

เป็นภาษาที่ใช้ควบคุมระบบรักษาความปลอดภัยของฐานข้อมูล ซึ่งประกอบไปด้วยคำสั่ง 2 คำสั่ง ได้แก่

- ◆ คำสั่ง GRANT
- ◆ คำสั่ง REVOKE

คำสั่ง GRANT เป็นคำสั่งที่ใช้กำหนดสิทธิให้กับผู้ใช้แต่ละคน ให้มีสิทธิกระทำการใดกับข้อมูล เช่นการเพิ่มข้อมูล การแก้ไขข้อมูลหรือการลบข้อมูลในตารางใดได้บ้าง หรือการกำหนดให้มีสิทธิดูข้อมูลได้เพียงอย่างเดียว เป็นต้น

คำสั่ง REVOKE จะเป็นคำสั่งให้มีการยกเลิกสิทธินั้นหลังจากที่ได้ GRANT แล้ว

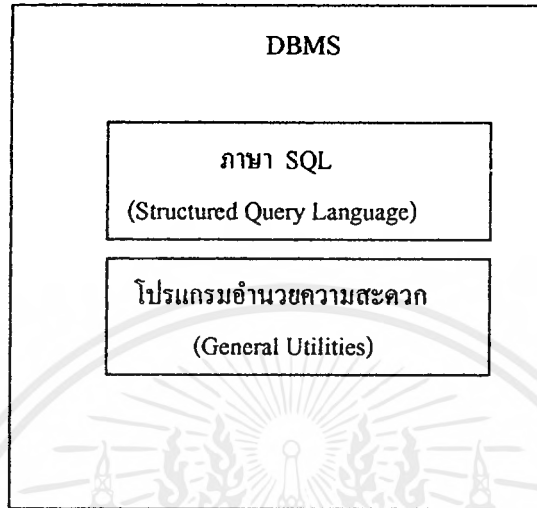
ผู้ที่ใช้คำสั่ง GRANT หรือ REVOKE ได้นั้นจะต้องเป็นผู้ที่มีหน้าที่รับผิดชอบในการออกแบบและควบคุมฐานข้อมูลเช่น DBA เพื่อเป็นการป้องกันกับผู้ใช้งานคนอื่นที่ไม่เกี่ยวข้องมีโอกาสเข้ามาดูข้อมูลของคนอื่นหรือมาทำลายข้อมูลได้

ผู้ใช้ทั่วไป	UPDATE	INSERT	DELETE	SELECT
สมศักดิ์	X	X	X	X
วิไล	X			X
พิศมัย		X	X	X
พิชัย				X

### รูปที่ 2.11 ตัวอย่างตารางแสดงการกำหนดสิทธิให้กับผู้ใช้งานฐานข้อมูลแต่ละคน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ : เครื่องหมายกากบาท (X) หมายถึงการอนุญาตให้ผู้ใช้คนนั้นสามารถกระทำการใดกับฐานข้อมูลได้ เช่น พิกมัยมีการกำหนดสิทธิให้สามารถเพิ่ม(INSERT) ลบ(DELETE) และ เรียกค้น(SELECT) ข้อมูลจากฐานข้อมูลได้ แต่ไม่มีสิทธิปรับปรุง (UPDATE) ข้อมูลได้ เป็นต้น



รูปที่ 2.12 แสดงส่วนประกอบของระบบ จัดการฐานข้อมูล (DBMS)

## 2. โปรแกรมอำนวยความสะดวก (General Utilities)

เป็นโปรแกรมส่วนหนึ่งที่อยู่ใน DBMS ซึ่งจะช่วยให้คุณจัดการฐานข้อมูลเช่นการสร้างฐานข้อมูลและตาราง การค้นหา การเพิ่ม การลบ หรือการปรับปรุงระเบียบข้อมูลจากตารางภายในฐานข้อมูล โดยสามารถเรียกผ่านจากเมนูของโปรแกรมอำนวยความสะดวกนี้ได้ เป็นต้น

## 3. โปรแกรมช่วยสร้างโปรแกรมประยุกต์และรายงาน (Application and Report Generators)

DBMS บางตัวจะมีภาษาโปรแกรมที่สร้างขึ้นมาสำหรับการทำงานบางอย่างที่ซับซ้อน ซึ่งผู้ใช้งานฐานข้อมูลจะต้องเขียนขึ้น เพื่อใช้จัดการกับข้อมูลภายในฐานข้อมูล ที่อาจจะต้องมีการใช้ข้อมูลจากหลายแฟ้มข้อมูลเพื่อสร้างแบบฟอร์มการบันทึกข้อมูล หรือรายงานที่มีรูปแบบที่ซับซ้อน เมื่อทำการแปลหรือ COMPILE โปรแกรมที่เขียนขึ้นนี้ โปรแกรมช่วยสร้างโปรแกรมประยุกต์ (Application Generators) จะทำการสร้างโปรแกรมอื่นอีกที่อยู่ในรูปแบบภาษาในยุคที่ 3 เช่น COBOL ,C หรือ PASCAL เพื่ออำนวยความสะดวกในการนำโปรแกรม ดังกล่าวไปปรับปรุงเพิ่มเติมเพื่อทำงานอย่างอื่นที่ซับซ้อนยิ่งขึ้นต่อไป

สำหรับโปรแกรมช่วยสร้างรายงาน (Report Generators)จะเป็นโปรแกรมที่ช่วยในการสร้างรายงาน โดยผู้ใช้งานฐานข้อมูลไม่จำเป็นต้องเขียนรายละเอียดของโปรแกรมการสร้างรายงานมากนัก เพียงเอกสกรีนเป็นเอกสารพิมพ์ หรือใช้เพื่อใช้ในการนำเสนอข้อมูล โดยผู้ดูแลระบบจะเก็บข้อมูลเข้ามารวบรวมเข้าด้วยกัน แล้วจึงนำข้อมูลเหล่านั้นมาประมวลผลและนำเสนอออกมาเป็นรายงานในรูปแบบต่างๆ ซึ่งการนำข้อมูลมาประมวลผลและนำเสนอออกมาเป็นรายงานนั้น จะขึ้นอยู่กับความต้องการของผู้ใช้งานนั่นเอง ไม่ว่าจะเป็นกรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่บอกรูแบบของรายงานที่ต้องการ เช่น ข้อความในหัวรายงาน ข้อมูลที่จะทำรายงานนำมาจากเพิ่มข้อมูลใด ต้องการแสดงข้อมูลใดบ้างออกมาในรายงาน ให้มีการแสดงผลรวมของข้อมูลในเขตข้อมูลใด เป็นต้น ซึ่งโปรแกรมช่วยสร้างรายงานนี้จะทำการสร้างรายงานออกมาให้ตามข้อกำหนดที่ผู้ใช้กำหนดขึ้น

รูปแบบของ SQL โปรแกรมช่วยสร้างโปรแกรมประยุกต์ (Application Generators) หรือ โปรแกรมช่วยสร้างรายงาน (Report Generators) นี้ จะอยู่ในรูปแบบของภาษาในยุคที่ 4 เนื่องจากมีความง่ายต่อการเขียน และทำความเข้าใจ ผู้ที่ไม่เคยเขียนโปรแกรมทางคอมพิวเตอร์มาก่อนก็สามารถเรียนรู้ และใช้งานฐานข้อมูลได้อย่างง่ายดาย

#### 4. พจนานุกรมข้อมูล (Data Dictionary)

DBMS หลายตัวจะมีการรวมพจนานุกรมข้อมูลเป็นส่วนหนึ่งของ DBMS ด้วย ซึ่งพจนานุกรมข้อมูลนี้จะป็นองค์ประกอบทางซอฟต์แวร์ ทำหน้าที่เก็บรายละเอียดเกี่ยวกับข้อมูลภายในฐานข้อมูล เช่น โครงสร้างของแต่ละตาราง ใครเป็นผู้สร้าง และสร้างเมื่อใด และแต่ละตารางประกอบด้วยเขตข้อมูลใดบ้าง คุณลักษณะของแต่ละเขตข้อมูลเป็นอย่างไร มีการเรียกใช้อุปกรณ์โปรแกรมประยุกต์ใดบ้าง และมีตารางใดที่มีความสัมพันธ์กันบ้าง มีเขตข้อมูลใดเป็นคีย์บ้าง เป็นต้น

พจนานุกรมข้อมูลยังมีส่วนที่ทำหน้าที่สำคัญดังต่อไปนี้

- **ควบคุมระบบรักษาความปลอดภัยของข้อมูล และความคงสภาพของข้อมูล (Data Security and Data Integrity)**

เนื่องจากในระบบฐานข้อมูลที่มีผู้ใช้หลายคนที่มีการใช้งานฐานข้อมูลร่วมกัน ข้อมูลบางอย่างที่ถูกเก็บอยู่ในฐานข้อมูล ควรมีการป้องกันไม่ให้ผู้ที่ไม่มีส่วนเกี่ยวข้องกับข้อมูลนั้นเข้ามาดูหรือแก้ไขข้อมูลนั้นได้ ซึ่ง DBMS ได้มีส่วนในการจัดการเกี่ยวกับเรื่องนี้โดยเฉพาะ โดยจะมีการเก็บรายละเอียดเกี่ยวกับผู้ใช้งานแต่ละคนไว้ในพจนานุกรมข้อมูล ว่ามีใครบ้างที่สามารถเข้ามาใช้งานฐานข้อมูลได้ และมีรหัสผ่านเป็นอย่างไรสามารถใช้งานได้ในระดับใด เช่นดูข้อมูลได้อย่างเดียว หรือสามารถแก้ไขข้อมูลได้ด้วย เป็นต้น

นอกจากนี้พจนานุกรมข้อมูลยังมีส่วนในการควบคุมความคงสภาพของข้อมูล กล่าวคือรายละเอียดเกี่ยวกับข้อมูลภายในเขตข้อมูลแต่ละตัวจะถูกเก็บอยู่ในพจนานุกรมข้อมูลทั้งหมด ดังนั้นการบันทึกหรือแก้ไขข้อมูลในเขตข้อมูลใดแต่ละครั้ง พจนานุกรมข้อมูลจะต้องทำการตรวจสอบว่าข้อมูลนั้นอยู่ในขอบเขตที่กำหนดหรือไม่ และจะยอมรับให้มีการบันทึกหรือแก้ไขเมื่อข้อมูลนั้นอยู่ในขอบเขตที่กำหนดเท่านั้น ตัวอย่างเช่น ข้อมูลเกี่ยวกับเงินเดือน ที่เก็บอยู่ในเขตข้อมูลเงินเดือนในตารางพนักงาน ถ้ามีการกำหนดไว้ตั้งแต่ตอนสร้างตารางพนักงานว่าข้อมูลในเขตข้อมูลเงินเดือนจะต้องเป็นตัวเลขเท่านั้น ดังนั้นเมื่อมีการเพิ่มหรือแก้ไขข้อมูลเงินเดือนในตารางพนักงานแต่ละคน พจนานุกรมข้อมูลจะทำการตรวจสอบว่าข้อมูลบันทึกนั้นเป็นตัวเลขหรือไม่ ถ้าไม่ใช่จะไม่ยอมให้มีการบันทึกหรือแก้ไขข้อมูลนั้นได้ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **ควบคุมเกี่ยวกับการใช้งานฐานข้อมูลพร้อมกัน(Concurrency Control)**

ในกรณีที่มีผู้ใช้งานฐานข้อมูลพร้อมกันหลายคน ผู้ใช้งานแต่ละคนจะสามารถอ่านข้อมูลได้พร้อมกัน แต่ถ้าจะทำการเพิ่มข้อมูล แก้ไขข้อมูลหรือลบข้อมูลใด จะมีเพียงผู้เดียวเท่านั้นที่สามารถทำได้ โดยจะมีการใช้หลักการล็อก (Locking) ข้อมูลนั้นเพื่อป้องกันการแก้ไขข้อมูลอันเดียวกันซึ่งหึ่งถูกแก้ไขไปแล้วโดยผู้อื่น ดังนั้นขณะใดขณะหนึ่งจะมีเพียงผู้เดียวเท่านั้นที่จะสามารถแก้ไขข้อมูลนั้นได้ รายละเอียดเหล่านี้จะถูกเก็บอยู่ในพจนานุกรมข้อมูลทั้งหมด

### **ตัวอย่างของ DBMS ที่มีใช้อยู่ในปัจจุบัน**

ตารางในตารางที่ 2.1 แสดงตัวอย่างของ DBMS บางตัวที่มีใช้อยู่ในปัจจุบันและเป็นที่ยอมรับกัน โดยจะแสดงให้เห็นถึงชื่อของ DBMS ชนิดของเครื่องคอมพิวเตอร์ที่จะใช้งานกับ DBMS แต่ละประเภท โครงสร้างของฐานข้อมูลที่ใช้ ภาษาที่ใช้จัดการข้อมูล (DML) สำหรับ DBMS แต่ละประเภท

### **หลักเกณฑ์การเลือกใช้ DBMS**

เนื่องจากในปัจจุบันได้มีการพัฒนา DBMS ออกมาเป็นจำนวนมาก ซึ่ง DBMS แต่ละตัวจะมีคุณสมบัติที่แตกต่างกัน เห็นได้จากตัวอย่างในตารางที่ 2.1 DBMS บางตัวจะมีโครงสร้างฐานข้อมูลแบบเชิงสัมพันธ์ (Relational Database) บางตัวจะเป็นแบบลำดับชั้น (Hierarchical Database) และบางตัวจะเป็นแบบเครือข่าย (Network Database) และนอกจากนี้ DBMS แต่ละตัวสามารถใช้กับเครื่องคอมพิวเตอร์ที่แตกต่างกันไปรวมทั้งภาษาจัดการข้อมูลก็อาจจะแตกต่างกัน ดังนั้น ในการที่องค์กรจะตัดสินใจเลือกใช้ DBMS ตัวใดนั้นจึงต้องมีการกำหนดหลักเกณฑ์ต่าง ๆ ขึ้นมาเพื่อพิจารณา เพื่อจะได้สามารถตัดสินใจเลือก DBMS ที่มีความเหมาะสมต่อการใช้งานขององค์กรนั้น ๆ

ชื่อ DBMS	ประเภทของคอมพิวเตอร์	โครงสร้างฐานข้อมูล	ภาษาจัดการข้อมูล
DB2	เมนเฟรมคอมพิวเตอร์	ฐานข้อมูลเชิงสัมพันธ์	SQL , QBE
DBASE IV	ไมโครคอมพิวเตอร์	ฐานข้อมูลเชิงสัมพันธ์	มีของตนเอง , SQL
FoxBASE+	ไมโครคอมพิวเตอร์	ฐานข้อมูลเชิงสัมพันธ์	มีเป็นของตนเอง
FoxPro	ไมโครคอมพิวเตอร์	ฐานข้อมูลเชิงสัมพันธ์	มีของตนเอง , SQL
IDMS	เมนเฟรมคอมพิวเตอร์	ฐานข้อมูลแบบเครือข่าย	มีเป็นของตนเอง
IMS/VS	เมนเฟรมคอมพิวเตอร์	ฐานข้อมูลแบบลำดับขั้น	CICS
Ingres	มินิคอมพิวเตอร์	ฐานข้อมูลเชิงสัมพันธ์	SQL , QUEL
Informix	ไมโครคอมพิวเตอร์	ฐานข้อมูลเชิงสัมพันธ์	SQL , QBE
	ซูเปอร์มินิคอมพิวเตอร์		
Oracle	ไมโครคอมพิวเตอร์	ฐานข้อมูลเชิงสัมพันธ์	SQL
	เมนเฟรมคอมพิวเตอร์		
	มินิคอมพิวเตอร์		
PARADOX	ไมโครคอมพิวเตอร์	ฐานข้อมูลเชิงสัมพันธ์	มีเป็นของตนเอง

ตารางที่ 2.1 แสดงตัวอย่างของ DBMS ที่เป็นที่ยอมรับและมีใช้อยู่ในปัจจุบัน

ตัวอย่างของหลักเกณฑ์ที่จะสามารถนำมาใช้ในการพิจารณา DBMS มีดังต่อไปนี้

### 1. ความเข้ากันได้กับฮาร์ดแวร์ที่มีอยู่

จากที่กล่าวมาแล้วว่า DBMS แต่ละตัวจะขึ้นอยู่กับประเภทของเครื่องคอมพิวเตอร์ เช่น DBMS บางตัวจะสามารถใช้กับเครื่องระดับไมโครคอมพิวเตอร์เท่านั้น ไม่สามารถใช้กับเครื่องระดับอื่นได้ แต่บางตัวก็สามารถใช้กับเครื่องคอมพิวเตอร์ได้หลายระดับเช่นตั้งแต่ไมโครคอมพิวเตอร์จนถึงเมนเฟรมคอมพิวเตอร์ ดังนั้นการจะตัดสินใจเลือกใช้ DBMS ตัวใดจึงต้องดูว่าองค์กรของเราใช้ระบบคอมพิวเตอร์เป็นแบบใด เช่นเป็นไมโครคอมพิวเตอร์ มินิคอมพิวเตอร์ หรือเมนเฟรมคอมพิวเตอร์ เพื่อจะได้เลือก DBMS ที่สามารถใช้ได้กับระบบเครื่องคอมพิวเตอร์ที่มีอยู่

### 2. ความเร็วในการประมวลผลข้อมูล

DBMS แต่ละตัวอาจมีความเร็วในการประมวลผลข้อมูล การค้นหาข้อมูล หรือการเรียงลำดับข้อมูล ที่แตกต่างกันเมื่อนำมาใช้กับระบบเครื่องคอมพิวเตอร์ที่มีอยู่ ดังนั้นจึงต้องมีการทดสอบความสามารถของ แต่ละตัวว่าจะมีความเร็วในการทำงานมากน้อยแค่ไหน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. จำนวนผู้ใช้งานระบบเครื่องคอมพิวเตอร์

DBMS แต่ละตัวจะอนุญาตให้มีจำนวนผู้ใช้งานคอมพิวเตอร์พร้อมกันได้ไม่เท่ากัน ดังนั้นจึงต้องดูจากลักษณะงานของเราว่าต้องการให้มีผู้ใช้งานคอมพิวเตอร์พร้อมกันได้มากที่สุดกี่คน แล้วเลือกใช้ DBMS ให้ตรงกับความต้องการนั้น

### 4. จำนวนเพิ่มข้อมูลและขนาดของระเบียบ

จากลักษณะงานของเราควรต้องพิจารณาว่าต้องมีการเปิดใช้เพิ่มข้อมูลมากที่สุดกี่เพิ่ม แต่ละเพิ่มมีขนาดระเบียบใหญ่สุดประมาณเท่าใด (กี่ไบต์) แล้วทำการเลือก DBMS ที่มีความสามารถในการเปิดเพิ่มข้อมูลได้มากตามที่ต้องการ

### 5. ภาษาที่ใช้จัดการข้อมูล

เนื่องจาก DBMS แต่ละตัว จะมีภาษาที่ใช้จัดการข้อมูลต่างกัน เช่นบางตัวใช้ภาษาที่เรียกว่า SQL บางตัวก็จะมีภาษาการจัดการข้อมูลเป็นของตนเอง เป็นต้น

### 6. การใช้งานโปรแกรมช่วยสร้างโปรแกรมประยุกต์

DBMS บางตัวจะมีโปรแกรมช่วยสร้าง โปรแกรมประยุกต์หรือที่เรียกว่า Application Generator การพิจารณาอาจพิจารณาจากความสามารถในการทำงาน ความง่ายในการใช้งาน และภาษาหรือโปรแกรมประยุกต์ที่สร้างขึ้นเป็นภาษาใด ตรงกับความต้องการขององค์กรหรือไม่ เป็นต้น

### 7. การใช้งานโปรแกรมช่วยสร้างงาน

เป็นการพิจารณาความสามารถและความง่ายของการใช้งาน โปรแกรมช่วยสร้างรายงาน หรือที่เรียกว่า Report Generator ที่มีอยู่ใน DBMS แต่ละตัว

### 8. ระบบรักษาความปลอดภัยของ DBMS แต่ละตัว

เป็นการพิจารณาความสามารถในการรักษาความปลอดภัยของข้อมูลของ DBMS ว่ามีระบบรักษาความปลอดภัยที่สามารถเชื่อมั่น ได้มากน้อยเพียงใด

### 9. โครงสร้างฐานข้อมูลของ DBMS แต่ละประเภท

เป็นการพิจารณาโครงสร้างฐานข้อมูลของ DBMS ว่าเป็นแบบเชิงสัมพันธ์ หรือแบบลำดับขั้น หรือแบบเครือข่าย เพื่อจะได้เลือกใช้ได้ตรงกับความต้องการ

### 10. ความเหมาะสมของ DBMS ต่อลักษณะของงานที่ต้องทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แม้ว่า DBMS จะมีความสามารถมากมายในเรื่องต่าง ๆ แต่ทั้งนี้ทั้งนั้นก็ต้องพิจารณาด้วยว่า DBMSตัวนั้นมีความสามารถทำงานในสิ่งที่เราต้องการได้หรือไม่ ถ้าทำได้จะมีความยากง่ายอย่างไรในการทำงานสิ่งนั้น เหมาะสมและคุ้มค่ากับการลงทุนหรือไม่

## 11. ราคาของ DBMS แต่ละประเภท

เนื่องจาก DBMS แต่ละตัวจะมีความสามารถต่างกัน ดังนั้นยัง DBMS ตัวใดมีความสามารถมากก็จะมีราคาที่แพงมาก จึงเป็นสิ่งที่องค์กรควรต้องพิจารณาประกอบการตัดสินใจด้วย

### องค์ประกอบทางด้านข้อมูล

เป็นองค์ประกอบที่สำคัญอีกองค์ประกอบหนึ่งที่ต้องมีอยู่ในระบบฐานข้อมูล ตัวอย่างของข้อมูล เช่น ระเบียบของนักศึกษาแต่ละคนที่เก็บอยู่ในตารางนักศึกษา ซึ่งประกอบด้วยเขตข้อมูลรหัสนักศึกษาที่อยู่ รหัสคณะ รหัสสาขาวิชา เป็นต้น ข้อมูลที่เก็บอยู่ในฐานข้อมูลควรมีคุณสมบัติดังต่อไปนี้

#### 1. มีความถูกต้องทันสมัย สมเหตุสมผล

ตัวอย่างเช่น ถ้าระเบียบนักศึกษามีการเก็บข้อมูลเพศอยู่ในเขตข้อมูลเพศด้วย ดังนั้นถ้ามีการบันทึกข้อมูลนักศึกษาที่มีชื่อขึ้นต้นด้วยนาย ก็ควรมีข้อมูลเพศ เป็น M (หรือ Male) แต่ถ้าเป็น น.ส. ก็ควรมีข้อมูลเพศเป็น F (หรือ Female) เป็นต้น

#### 2. มีความซ้ำซ้อนของข้อมูลน้อยที่สุด

เนื่องจากข้อมูลในฐานข้อมูลสามารถประกอบด้วยตารางตั้งแต่หนึ่งตารางขึ้นไป ที่มีความสัมพันธ์กัน ข้อมูลในแต่ละตารางจะต้องมีความซ้ำซ้อนของข้อมูลน้อยที่สุดหรือไม่มีเลย ตัวอย่างเช่นถ้าฐานข้อมูลของระบบบุคลากร ประกอบด้วยตารางพนักงาน ซึ่งประกอบด้วยเขตข้อมูลดังต่อไปนี้

#### พนักงาน

รหัสพนักงาน	ชื่อ	อายุ	การศึกษา	ตำแหน่ง
-------------	------	------	----------	---------

และถ้ามีตารางการฝึกอบรมของพนักงานแต่ละคน ซึ่งประกอบด้วยเขตข้อมูลดังต่อไปนี้

#### การฝึกอบรม

รหัสพนักงาน	วิชาที่อบรม	สถานที่อบรม	วันเดือนปีที่เริ่มอบรม	วันเดือนปีที่จบหลักสูตร
-------------	-------------	-------------	------------------------	-------------------------

รูปที่ 2.13 แสดงความสัมพันธ์กันของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่าตารางการฝึกอบรมจะเก็บรายละเอียดข้อมูลเฉพาะรหัสพนักงานเท่านั้น ไม่ต้องเก็บชื่อ อายุ การศึกษา หรือตำแหน่ง ไว้ในตารางการฝึกอบรมอีก เนื่องจากถูกเก็บอยู่แล้วในตารางพนักงาน ซึ่งเป็นการลดความซ้ำซ้อนที่อาจเกิดขึ้นได้ระหว่างตารางนั่นเอง

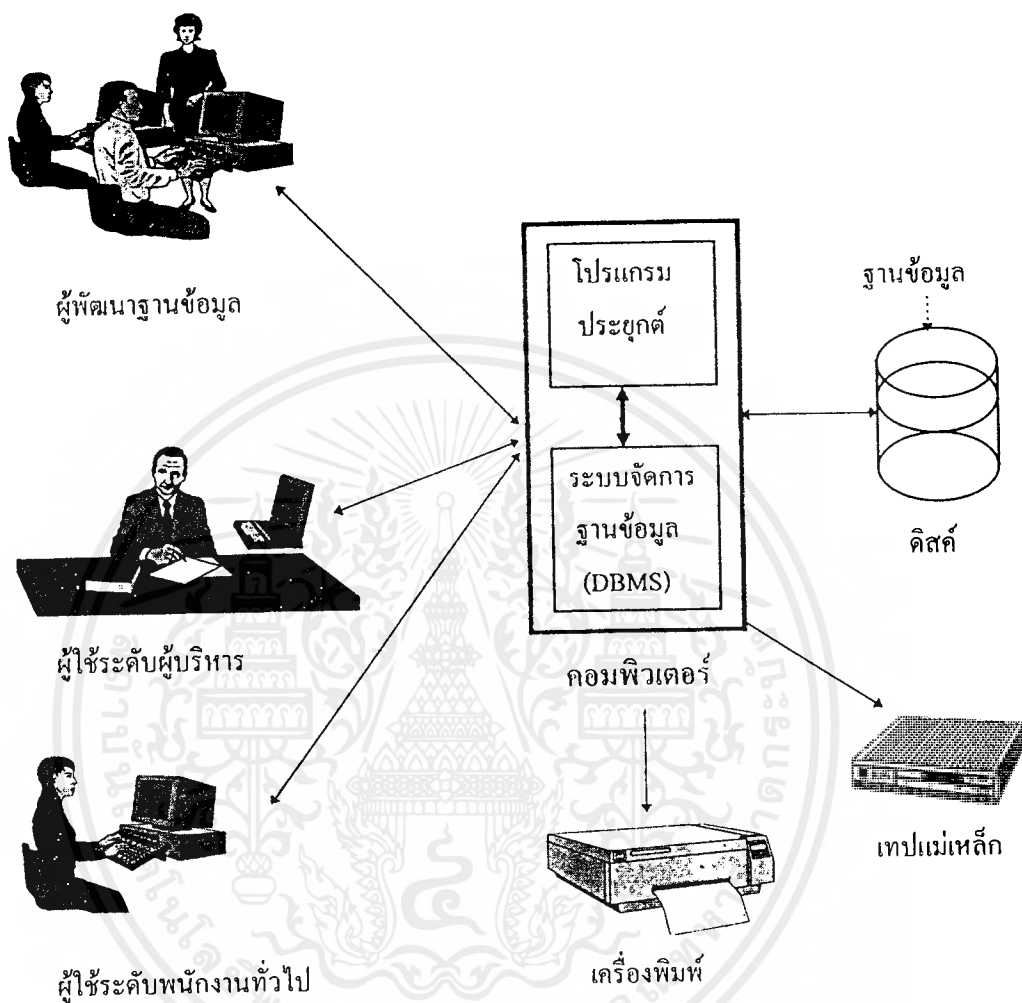
### 3. มีการแบ่งงานกันใช้ข้อมูล

ข้อมูลภายในฐานข้อมูลควรมีลักษณะที่สามารถให้ผู้ใช้งานหลายคนใช้ข้อมูลนั้นร่วมกันได้ เป็นการแบ่งกันใช้ข้อมูล (Sharing) กล่าวคือผู้ใช้งานฐานข้อมูลแต่ละคน จะสามารถดึงข้อมูลขึ้นเดียวกันขึ้นมาดูได้พร้อมกัน แต่ถ้าทำการแก้ไขข้อมูล จะมีเพียงคนเดียวเท่านั้นที่จะสามารถทำได้ เป็นต้น

#### 2.2.8 ความสัมพันธ์ระหว่างองค์ประกอบในระบบฐานข้อมูล

จากที่กล่าวมาแล้วถึงองค์ประกอบต่าง ๆ ที่มีในระบบฐานข้อมูล แต่ละองค์ประกอบจะมีความสำคัญจะขาดองค์ประกอบใดองค์ประกอบหนึ่งไม่ได้ และแต่ละองค์ประกอบนี้มีความสัมพันธ์กัน

จากรูปจะสามารถอธิบายความสัมพันธ์ขององค์ประกอบต่าง ๆ ได้ดังนี้ ผู้พัฒนาฐานข้อมูลซึ่งได้แก่ DBA และนักเขียนโปรแกรม (Programmers) จะเป็นผู้ที่ร่วมกันสร้างและพัฒนาฐานข้อมูลพร้อมกับโปรแกรมประยุกต์สำหรับใช้งานในระบบงานใด ๆ ขึ้นมา ตามความต้องการของผู้ใช้ (Users) ซึ่งผู้ใช้งานอาจจะเป็นผู้บริหารขององค์กรที่ต้องการสารสนเทศที่สร้างขึ้นจากระบบฐานข้อมูล หรืออาจเป็นพนักงานทั่วไปขององค์กร ที่สามารถบันทึกข้อมูลหรือค้นหาข้อมูลจากฐานข้อมูลในส่วนหนึ่งที่ตนเองรับผิดชอบ เพื่อดูหรือปรับปรุงข้อมูลในบางส่วนได้ ในการเรียกใช้ข้อมูลอาจทำได้โดยใช้คำสั่งผ่านทาง DBMS โดยตรง หรือผ่านทางโปรแกรมประยุกต์ซึ่งภายในโปรแกรมประยุกต์ก็จะมีการใช้คำสั่งการเรียกใช้ข้อมูลผ่านทาง DBMS อีกที่ โปรแกรมประยุกต์และ DBMS จะเป็นซอฟต์แวร์ที่มีการปฏิบัติการอยู่บนเครื่องคอมพิวเตอร์ซึ่งอาจเป็นเครื่องเมนเฟรม หรือมินิคอมพิวเตอร์ หรือเครื่องแม่ข่าย (Server) ในระบบเครือข่ายแบบโกลบอลก็ได้ สำหรับฐานข้อมูลที่เก็บรวบรวมข้อมูลต่าง ๆ ในระบบนั้น ก็จะเก็บอยู่ในสื่อบันทึกข้อมูลหรือดิสก์ เนื่องจากเป็นสื่อบันทึกข้อมูลที่มีความเร็วในการเข้าถึงข้อมูลสูง สำหรับเทปมักใช้เป็นสื่อสำหรับการสำรองข้อมูลจากดิสก์ขึ้นมาเก็บไว้ เพื่อเป็นการป้องกันความเสียหายที่อาจเกิดขึ้นกับข้อมูล



รูปที่ 2.14 แสดงความสัมพันธ์ระหว่างองค์ประกอบของระบบฐานข้อมูล

ดังนั้นจะเห็นว่าทุกองค์ประกอบจะมีความสัมพันธ์กันหมด การสร้างระบบฐานข้อมูลขึ้นมาใช้ในองค์กร ใดจะต้องประกอบด้วยองค์ประกอบทั้งสี่นี้จึงทำให้องค์กรประสบความสำเร็จและบรรลุเป้าหมายขององค์กรได้

## 2.3 สถาปัตยกรรมของระบบฐานข้อมูล

### 2.3.1 แนะนำ

คงได้กล่าวในบทก่อนแล้วว่า ฐานข้อมูลจะมีสิ่งพิเศษอยู่ข้อหนึ่งที่เหนือกว่าระบบเพิ่มข้อมูลทั่วไป ได้แก่เรื่องความเป็นอิสระกันของข้อมูล กล่าวคือ โปรแกรมประยุกต์ที่ผู้ใช้เขียนขึ้นจะไม่ขึ้นกับโครงสร้างของข้อมูลที่มีการเปลี่ยนแปลงไป ทั้งนี้เนื่องจากการจัดโครงสร้างของระบบฐานข้อมูล จะมีการแบ่งระดับของข้อมูลเป็น 3 ระดับ ซึ่งเป็นโครงสร้างที่กลุ่ม ANSI/SPARC ได้บัญญัติขึ้น ระดับของข้อมูลทั้ง 3 ระดับนี้จะเรียกว่า **สถาปัตยกรรมของระบบฐานข้อมูล**

### 2.3.2 สถาปัตยกรรม 3 ระดับ

สถาปัตยกรรมฐานข้อมูลนี้จะถูกแบ่งออกเป็น 3 ระดับด้วยกัน ได้แก่ ระดับความคิด (Conceptual Level) ระดับภายนอก (External Level) และระดับภายใน (Internal Level) แต่ละระดับจะมีลักษณะดังต่อไปนี้

#### ระดับความคิด(Conceptual Level)

เป็นระดับของการออกแบบฐานข้อมูล ผู้ที่รับผิดชอบการจัดการข้อมูลในระดับนี้ได้แก่ ดีบีเอ (DBA) ซึ่งจะเริ่มตั้งแต่การวิเคราะห์ความต้องการสารสนเทศของผู้ใช้เพื่อจะนำข้อมูลที่ได้ออกแบบฐานข้อมูล ว่าควรจะมีข้อมูลอะไรบ้าง และความสัมพันธ์ระหว่างข้อมูลเป็นอย่างไร ผลลัพธ์ที่ได้จากการออกแบบฐานข้อมูล จะทำให้เกิดสิ่งที่เรียกว่า สคีมา (Schema) ในระดับนี้จะเรียกว่า Conceptual Schema ซึ่งเป็นสิ่งที่ใช้อธิบายว่าฐานข้อมูลที่สร้างขึ้นประกอบด้วยเอนติตี้อะไรบ้าง แต่ละเอนติตี้ประกอบด้วยเขตข้อมูลใดบ้าง มีลักษณะเป็นอย่างไร ขนาดเท่าไร และความสัมพันธ์ระหว่างเอนติตี้เป็นอย่างไร เป็นต้น

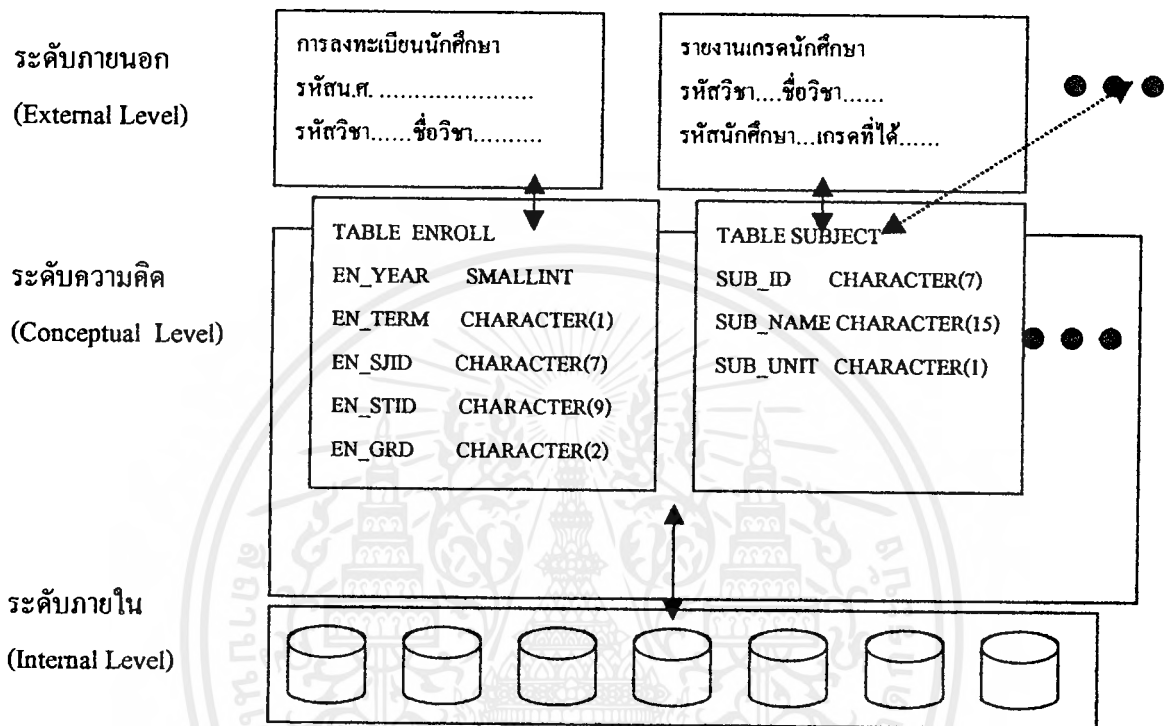
#### ระดับภายนอก(External Level)

เป็นระดับการมองหรือวิว (view) ของข้อมูลภายในฐานข้อมูลสำหรับผู้ใช้งานแต่ละคน ผู้ใช้ในระดับภายนอกนี้สามารถเป็นได้ตั้งแต่ นักเขียนโปรแกรมประยุกต์ (Application Programmer) หรืออาจเป็นผู้ปฏิบัติการทั่วไปที่เรียกว่า end-user ซึ่งระดับนี้จะป็นระดับที่มีการนำข้อมูลจากฐานข้อมูลไปใช้งาน โดยผู้ใช้แต่ละคนสามารถเลือกอ่านข้อมูลเฉพาะที่ตนสนใจหรือต้องการใช้เท่านั้น คำนึงผู้ใช้แต่ละคนจะมีวิวของข้อมูลในฐานข้อมูลที่แตกต่างกันได้ ซึ่งวิวของข้อมูลนี้จะถูกดึงมาจาก Conceptual Schema และสิ่งที่จะใช้อธิบายวิวข้อมูลที่ถูกดึงมาจากฐานข้อมูลที่อยู่ในระดับ Conceptual นี้จะเรียกว่า External Schema หรือ Subschema ซึ่งในระดับภายนอกนี้จะสามารถมีได้หลาย subschema ตามจำนวนผู้ใช้ที่มีการสร้างวิวของตนเองขึ้นมา

#### ระดับภายใน (Internal Level)

เป็นระดับของการจัดเก็บฐานข้อมูลในหน่วยเก็บข้อมูลสำรองจริง ๆ เช่น ข้อมูลถูกเก็บอยู่ที่ตำแหน่งใดในดิสก์ รวมทั้งข้อมูลที่เกี่ยวข้อง และ ก็จะถูกเก็บอยู่ในระดับนี้ทั้งหมด ระดับของข้อมูลเอกสารนี้เป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระดับนี้จะอยู่ในความรับผิดชอบของผู้ออกแบบฐานข้อมูลในระดับกายภาพ ซึ่งจะเป็นคนตัดสินใจว่าจะใช้อุปกรณ์ใดเป็นตัวเก็บข้อมูล วิธีการเข้าถึงข้อมูลเพื่อค้นหาหรือปรับปรุงข้อมูลจะใช้วิธีการใด รวมถึงวิธีการบำรุงรักษา และการเพิ่มประสิทธิภาพของฐานข้อมูล ผู้ใช้งานฐานข้อมูลทั่วไปไม่ต้องยุ่งเกี่ยวกับการจัดการข้อมูลในระดับภายในนี้เลย



รูปที่ 2.15 แสดงสถาปัตยกรรม 3 ระดับของระบบฐานข้อมูล

รูป 2.15 แสดงตัวอย่างของสถาปัตยกรรม 3 ระดับ ในระดับความคิดเป็นการนิยามโครงสร้างฐานข้อมูลระบบทะเบียนนักศึกษา ซึ่งประกอบด้วยตารางต่าง ๆ เช่น ตารางการลงทะเบียน (ENROLL) ตารางรายวิชา (SUBJECT) ฯลฯ ตารางการลงทะเบียนจะมีโครงสร้างที่ประกอบไปด้วยเขตข้อมูล ปีการศึกษา (EN\_YEAR) เทอม (EN\_TERM) รหัสวิชา (EN\_SJID) รหัสนักศึกษา (EN\_STID) เกรดที่ได้ (EN\_GRD) สำหรับตารางรายวิชาจะประกอบด้วยเขตข้อมูลรหัสวิชา (SUB\_ID) ชื่อวิชา (SUB\_NAME) จำนวนหน่วยกิต (SUB\_UNIT) ซึ่งมีการกำหนดประเภทข้อมูลและขนาดตามตัวอย่างในรูป

ในระดับภายนอกจะเป็นการนำเข้าสู่ข้อมูลในระดับความคิดไปใช้งาน จากรูป 2.15 จะแสดงตัวอย่างหน้าจอของผู้ใช้แต่ละคน โดยผู้ใช้คนที่ 1 จะต้องการดูข้อมูลการลงทะเบียนของนักศึกษาแต่ละคน โดยต้องการทราบเฉพาะรหัสนักศึกษา รหัสวิชา และชื่อวิชาที่นักศึกษาแต่ละคนลงทะเบียนเท่านั้น ซึ่งเป็นหน้าที่ของผู้เขียน โปรแกรมที่จะทำการสร้างวิวข้อมูลให้กับผู้ใช้คนที่ 1 โดยดึงเขตข้อมูลบางเขตที่ต้องการจากตารางในระดับความคิดออกมาแสดง สำหรับผู้ใช้คนที่ 2 ต้องการดูเกรดในวิชาที่นักศึกษาลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทะเบียน ผู้เขียน โปรแกรมก็ต้องสร้างวิวประกอบด้วยเขตข้อมูลรหัสวิชา ชื่อวิชา รหัสนักศึกษา และ เกรด ที่ได้ ที่นำมาจากตารางลงทะเบียน และ ตารางรายวิชาเป็นต้น

### 2.3.3 ประโยชน์ของสถาปัตยกรรม 3 ระดับ

จากโครงสร้างสถาปัตยกรรมของฐานข้อมูลทั้ง 3 ระดับนี้ แต่ละระดับจะมี DBMS ทำหน้าที่แมป (map) หรือแปลระดับข้อมูลจากระดับหนึ่งเป็นอีกระดับหนึ่ง ซึ่งได้แก่ระหว่างระดับภายนอกกับระดับความคิด และระหว่างระดับความคิดกับระดับภายใน การแมปหรือการแปลระดับข้อมูลนี้จะทำให้เกิดประโยชน์ในด้านต่าง ๆ ดังต่อไปนี้

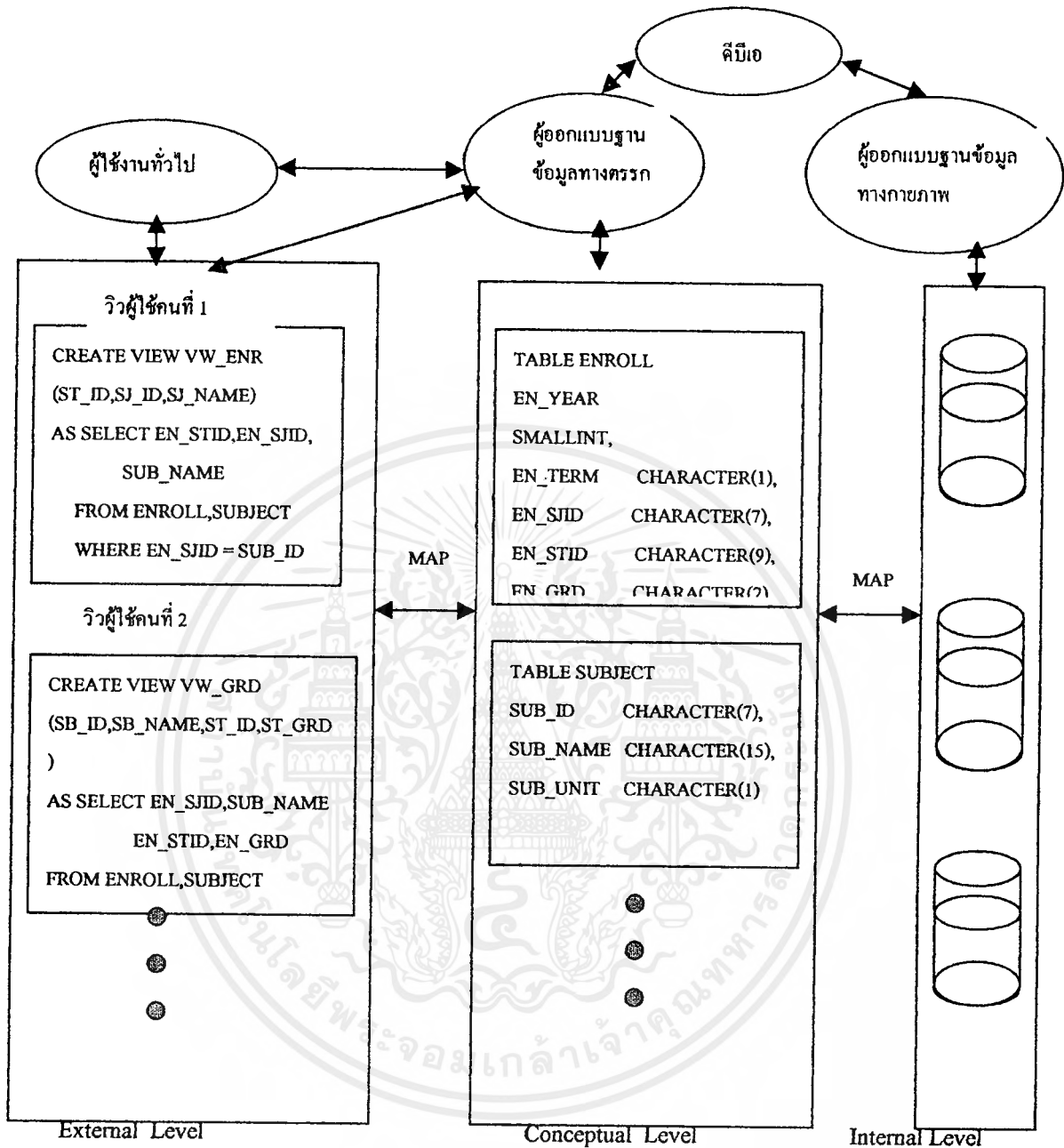
#### 1. มุมมองข้อมูลของผู้ใช้งาน (view of each user)

##### ● ระดับความคิดและระดับภายนอก

การแมปข้อมูลระหว่างระดับความคิดและระดับภายนอก จะทำให้ผู้ใช้ฐานข้อมูลสามารถมีมุมมองข้อมูลที่แตกต่างกันได้ จากตัวอย่างในรูป 2.15 ซึ่งมีการออกแบบฐานข้อมูลในระดับความคิดของระบบทะเบียนนักศึกษา ในรูป 2.16 จะแสดงคำสั่งการสร้างวิวให้แก่ผู้ใช้แต่ละคน เพื่อนำเขตข้อมูลบางเขตที่ผู้ใช้สนใจออกมาแสดงในระดับภายนอก (คำสั่งการสร้างวิว ได้อธิบายรายละเอียดไว้ในบทที่ 8)

ในระดับภายนอก ผู้ใช้คนที่ 1 มีความต้องการทราบว่านักศึกษาแต่ละคนลงเรียนวิชาใดบ้าง โดยต้องการใช้ข้อมูล 3 เขตข้อมูล ได้แก่ รหัสนักศึกษา (ST\_ID) รหัสวิชา (SJ\_ID) จากตารางลงทะเบียน และชื่อวิชา (SJ\_NAME) จากตารางรายวิชา

ในขณะที่ผู้ใช้คนที่ 2 ต้องการทราบว่านักศึกษาที่ลงวิชาเรียนแต่ละวิชาได้เกรดในแต่ละวิชาเป็นเท่าใด โดยต้องการใช้ 4 เขตข้อมูล ได้แก่ รหัสวิชา (SB\_ID) รหัสนักศึกษา (ST\_ID) และเกรด (ST\_GRD) จากตารางลงทะเบียนเรียน และชื่อวิชา (SB\_NAME) จากตารางรายวิชา



รูปที่ 2.16 ตัวอย่างสถาปัตยกรรม 3 ระดับ ของฐานข้อมูลระบบทะเบียนนักศึกษา

คำสั่ง CREATE VIEW แล้วตามด้วยชื่อวิวที่ตั้งขึ้น จะเปรียบเสมือนกับเป็นการสร้างตารางสมมติขึ้นมาโดยนำข้อมูลมาจากตารางจริงที่อยู่ในฐานข้อมูลระดับความคิด โดยคำสั่งนี้จะต้องใช้ร่วมกับคำสั่ง SELECT เพื่อนำข้อมูลจากตารางจริงขึ้นมายังตารางสมมติ ดังนั้นจากคำสั่งต่อไปนี้

```
CREATE VIEW VW_ENR(ST_ID,SJ_ID,SJ_NAME)
AS SELECT EN_STID,EN_SJID,SUB_NAME)
FROM ENROLL,SUBJECT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WHERE EN\_SJID = SUB\_ID

จะเป็นคำสั่งให้มีการสร้างวิวที่ชื่อ VW\_ENR โดยคำสั่ง SELECT จะทำการนำข้อมูลในเขตข้อมูลรหัสนักศึกษา (EN\_STID) รหัสวิชา (EN\_SJID) และ ชื่อวิชา (SUB\_NAME) จากตารางการลงทะเบียนและตารางรายวิชา (อยู่หลัง FROM) โดยมีเงื่อนไข (อยู่หลัง WHERE) ว่า ให้ดึงข้อมูลเฉพาะระเบียบที่มีรหัสวิชา (SUB\_ID) ในตารางการลงทะเบียน เท่ากับรหัสวิชา ในตารางรายวิชาเท่านั้น ข้อมูลจากเขตข้อมูลต่าง ๆ ที่ดึงได้ จะถูกนำไปเก็บไว้ในชื่อเขตข้อมูลใหม่ของวิว VW\_ENR อันได้แก่ เขตข้อมูลรหัสนักศึกษา (ST\_ID) รหัสวิชา (SJ\_ID) และชื่อวิชา (SJ\_NAME) เป็นต้น

จากคำสั่งการสร้างวิวที่ชื่อ VW\_ENR และวิว VW\_GRD จะเห็นว่าผู้ใช้ทั้งสองคนจะมีการมองข้อมูลภายในฐานข้อมูล (ระดับความคิด) ที่แตกต่างกันไป หรือพูดได้ว่ามีวิวของข้อมูลที่แตกต่างกัน นอกจากนี้วิวของผู้ใช้ทั้งสองมีการอ้างถึงชื่อเขตข้อมูลที่ไม่เหมือนกัน ทั้ง ๆ ที่เป็นเขตข้อมูลที่ดึงมาจากตารางเดียวกัน เช่นวิว VW\_ENR ข้อมูลรหัสวิชา (EN\_SJID) ที่ถูกดึงมาจากตารางการลงทะเบียน จะถูกเก็บอยู่ในชื่อ SJ\_ID ในขณะที่วิว VW\_GRD มีการใช้ข้อมูลรหัสวิชา (EN\_SJID) จากตารางรายวิชาเช่นกัน แต่จะถูกเก็บอยู่ในชื่อ SB\_ID แทน ทั้งนี้เนื่องจากเป็นหน้าที่ของ DBMS ที่จะทำการแปล (Map) ชื่อข้อมูลให้ระหว่างระดับภายนอกและระดับความคิด ให้สามารถเข้าใจว่าคือข้อมูลตัวเดียวกันนั่นเอง

#### ● ระดับความคิดและระดับภายใน

การเม็บบข้อมูลระหว่างระดับความคิดและระดับภายใน จะทำให้ผู้ใช้งานฐานข้อมูลไม่ว่าในระดับความคิดหรือระดับภายนอก ไม่จำเป็นต้องทราบว่าคุณสมบัติที่ตนต้องการใช้นั้นถูกจัดเก็บอยู่อย่างไรในดิสก์ เมื่อต้องการใช้ข้อมูลใดไม่ว่าจะเป็นตารางหรือเขตข้อมูล จะสามารถอ้างถึงชื่อเขตข้อมูลนั้นได้โดยตรง ซึ่งเป็นหน้าที่ของ DBMS ที่จะดูว่าข้อมูลที่ผู้ใช้ต้องการนั้นถูกเก็บอยู่ ณ ตำแหน่งใด (แตร็กที่เท่าใด ไชลินเดอร์ที่เท่าใด) ในดิสก์ แล้วทำการดึงข้อมูลนั้นจากดิสก์ มาให้แก่ผู้ใช้ได้

## 2. ความเป็นอิสระกันของข้อมูล

#### ● ระดับความคิดและระดับภายนอก

การเปลี่ยนแปลงฐานข้อมูลในระดับความคิด ไม่ว่าจะเป็นการเพิ่มตาราง การเพิ่มเขตข้อมูล หรือการเปลี่ยนแปลงขนาดของเขตข้อมูล จะไม่มีผลกระทบต่อโปรแกรมประยุกต์ที่ผู้ใช้เขียนขึ้นในระดับภายนอก ตัวอย่างเช่น ถ้ามีการเปลี่ยนแปลงขนาดของเขตข้อมูลรหัสนักศึกษาจากเดิม คือ 9 เปลี่ยนไปเป็น 10 ในตารางการลงทะเบียน (ENROLL) ของฐานข้อมูลในระดับความคิด ก็สามารถทำได้เลยโดยไม่ต้องไปเปลี่ยนโปรแกรมประยุกต์ตามไปด้วย ยกเว้นว่าถ้ามีการลบตารางหรือเขตข้อมูลใดจากฐานข้อมูลในระดับความคิด แล้วโปรแกรมประยุกต์มีการเรียกใช้งานตารางหรือเขตข้อมูลนั้นอยู่ ก็อาจจะต้องไปแก้ไขโปรแกรมประยุกต์นั้นด้วย แต่โดยปกติแล้วการเปลี่ยนแปลงโครงสร้างของฐานข้อมูลเป็นสิ่งที่ไม่ควรทำหรือถ้าทำก็ไม่ควรจะทำบ่อยนัก

#### ● ระดับความคิดและระดับภายใน

การเม็บบข้อมูลระดับความคิดและระดับภายใน ยังทำให้เกิดความเป็นอิสระกันข้อมูลเกิดขึ้นด้วย เนื่องจากถ้าระดับภายในมีการเปลี่ยนแปลงวิธีการจัดเก็บข้อมูลในดิสก์เช่นจากเดิมใช้วิธีการจัดเก็บแบบลำดับเชิงดัชนี (Index Sequential) เปลี่ยนเป็นแบบสุ่ม (Direct Access) ในระดับความคิดเมื่อมีการเอกสารนี้เป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สร้างตารางใดจะไม่มีผลกระทบหรือไม่ต้องรับรู้ถึงการเปลี่ยนแปลงดังกล่าว หรือ โปรแกรมประยุกต์ที่เขียนในระดับภายนอกก็ไม่จำเป็นต้องแก้ไข โปรแกรมตามวิธีการจัดเก็บที่เปลี่ยนแปลงไป

## 2.4 การออกแบบฐานข้อมูล

### 2.4.1 แนะนำ

ในองค์กรที่มีการใช้งานฐานข้อมูลโดยทั่วไป การจะใช้งานฐานข้อมูลได้อย่างมีประสิทธิภาพและตรงกับความต้องการของผู้ใช้นั้น ย่อมจะต้องมาจากการออกแบบฐานข้อมูลที่มีการวางแผนมาเป็นอย่างดี โดยจะต้องมีการพัฒนาระบบฐานข้อมูลตามขั้นตอนที่ถูกต้อง ในบทนี้จะได้ศึกษาถึงขั้นตอนในการพัฒนาระบบฐานข้อมูล และการออกแบบฐาน ซึ่งเราได้เคยศึกษาถึงหลักการในการออกแบบฐานข้อมูลโดยใช้โมเดล E-R ให้อยู่ในรูปแบบของโมเดลเชิงสัมพันธ์ หรือ ในรูปของรีเลชันต่าง ๆ ที่นอร์มัลไลซ์ และนอกจากนี้ก็จะแสดงขั้นตอนการออกแบบฐานข้อมูล โดยไม่ต้องผ่านโมเดล E-R ซึ่งจะเป็นการออกแบบฐานข้อมูลเชิงสัมพันธ์ที่อยู่ในรูปแบบของรีเลชันโดยตรงหลังจากการวิเคราะห์ความต้องการของผู้ใช้แล้ว

### 2.4.2 ขั้นตอนการพัฒนาระบบฐานข้อมูล

ในการพัฒนาฐานข้อมูลเพื่อสร้างระบบสารสนเทศ โดยทั่วไปนั้นจะมีวงจรในการพัฒนา ซึ่งเป็นขั้นตอนที่มีการทำงานเป็นลำดับขั้นตั้งแต่ต้นจนกระทั่งสามารถสร้างระบบสารสนเทศออกมาได้ และเป็นขั้นตอนที่ผู้พัฒนาระบบซึ่งอาจประกอบด้วยผู้จัดการโครงการ นักวิเคราะห์ระบบ (System Analyst) และผู้ออกแบบฐานข้อมูล (DBA) จะต้องร่วมกันศึกษาและทำความเข้าใจในแต่ละขั้นตอน ซึ่งโดยทั่วไปแล้วขั้นตอนการพัฒนาระบบจะมีอยู่ด้วยกัน 7 ขั้นตอนดังต่อไปนี้

#### 1. การวิเคราะห์ปัญหา (Problem Analysis)

เป็นขั้นตอนการวิเคราะห์ปัญหาของระบบงานเดิม เมื่อผู้บริหารขององค์กรมีความต้องการที่จะสร้างระบบสารสนเทศขึ้น เนื่องจากความล้มเหลวของระบบงานเดิม หรือการไม่มีประสิทธิภาพเพียงพอของระบบงานเดิม ที่จะตอบสนองความต้องการในปัจจุบันได้ ตัวอย่างเช่นระบบงานเดิมไม่สามารถให้ข้อมูลข่าวสารว่าสินค้าใดของบริษัทที่มียอดขายสูงสุด หรือการจัดกลุ่มลูกค้าเพื่อตัดสินใจว่าสินค้าชนิดหนึ่งเป็นที่สนใจลูกค้ากลุ่มใด เพื่อนำสารสนเทศที่ได้มาใช้ในการวางแผนและการตัดสินใจ

#### 2. การศึกษาความเป็นไปได้ (Feasibility Study)

หลังจากที่ทราบปัญหาของระบบงานเดิมแล้ว ขั้นตอนต่อไปคือการศึกษาความเป็นไปได้ว่าการสร้างระบบสารสนเทศ หรือการแก้ไขระบบสารสนเทศเดิมมีความเป็นไปได้หรือไม่ ซึ่งจะมีการศึกษาความเป็นไปได้ในด้านต่าง ๆ ดังนี้

- **ความเป็นไปได้ของเทคโนโลยี (Technological Feasibility)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นการศึกษาระบบงานเดิมมีอุปกรณ์ทางด้านฮาร์ดแวร์และซอฟต์แวร์เพียงพอที่จะรองรับสารสนเทศที่จะเกิดขึ้นได้หรือไม่ ถ้าไม่เพียงพอหรือยังไม่มีการวิเคราะห์ได้ว่าควรมีการจัดซื้อฮาร์ดแวร์และซอฟต์แวร์ประเภทใดเพิ่มเติม หรือถ้ามีอยู่แล้วก็จะต้องวิเคราะห์ถึงความสามารถของฮาร์ดแวร์และซอฟต์แวร์ดังกล่าว ว่ามีความสามารถอยู่ในระดับใด เพียงพอที่จะใช้สร้างระบบสารสนเทศได้หรือไม่ เป็นต้น

- **ความเป็นไปได้ทางการปฏิบัติการ (Operational Feasibility)**

เป็นการวิเคราะห์ว่าระบบงานเดิมมีบุคลากรที่มีความสามารถหรือประสบการณ์ในการพัฒนาและติดตั้งระบบหรือไม่ ถ้าไม่มีจะหาได้หรือไม่ นอกจากนี้ยังต้องพิจารณาด้วยว่าผู้ใช้ระบบมีความคิดเห็นอย่างไรกับการเปลี่ยนแปลงของระบบที่จะเกิดขึ้น

- **ความเป็นไปได้ทางด้านเศรษฐศาสตร์ (Economic Feasibility)**

เป็นการศึกษาค่าใช้จ่ายต่าง ๆ ที่จะเกิดขึ้นตั้งแต่เริ่มต้นพัฒนาระบบจนกระทั่งมีการติดตั้งและใช้งานระบบจริง รวมไปถึงค่าใช้จ่ายประจำวันที่จะเกิดขึ้นด้วย นอกจากนี้ยังต้องทำการคาดการณ์ถึงผลประโยชน์ที่จะได้รับ รวมทั้งเวลาที่จะต้องใช้ในการพัฒนาระบบ เพื่อนำข้อมูลที่ได้มาสรุปว่าคุ้มค่าหรือไม่ที่จะมีการเปลี่ยนแปลงระบบเกิดขึ้น ซึ่งในการนี้ผู้บริหารจะเป็นผู้ตัดสินใจว่าสมควรจะให้ดำเนินการพัฒนาต่อไปหรือจะยกเลิกโครงการพัฒนาดังกล่าว

### 3. การวิเคราะห์ความต้องการของผู้ใช้ (Users Requirement Analysis)

หลังจากศึกษาความเป็นไปได้ของระบบ และผู้บริหารเห็นสมควรที่จะให้ดำเนินการพัฒนาต่อขั้นตอนนี้ที่นักวิเคราะห์ระบบจะต้องทำคือ การวิเคราะห์ความต้องการของผู้ใช้ ความต้องการในที่นี้จะหมายถึงความต้องการข้อมูลของผู้ปฏิบัติงาน (end user) และความต้องการสารสนเทศของผู้บริหารซึ่งเป็นเจ้าของหน่วยงาน ซึ่งเป็นขั้นตอนสำคัญ เพื่อให้สามารถออกแบบระบบใหม่ได้ตรงกับความต้องการนั้นมากที่สุด ในขั้นตอนนี้จะเริ่มตั้งแต่การศึกษาระบบการทำงานขององค์กรซึ่งเป็นระบบงานเดิมให้เข้าใจก่อน ว่ามีลักษณะการทำงานอย่างไร และจะมีการเก็บรวบรวมข้อมูลต่าง ๆ จากผู้ใช้ รวมไปถึงกฎเกณฑ์และข้อบังคับต่าง ๆ ด้วย สำหรับวิธีการเก็บรวบรวมข้อมูลนั้น จะสามารถทำได้หลายวิธีเช่นการใช้แบบสอบถาม การสัมภาษณ์ผู้ใช้ในระดับบริหาร และระดับพนักงานทั่วไป หรือจากรายงานต่าง ๆ ขององค์กรนั้น ๆ หลังจากที่ได้ข้อมูลมาพอสมควรก็จะนำข้อมูลเหล่านั้นมาวิเคราะห์เพื่อสรุปให้ได้รายละเอียดต่อไปนี้

- **ขอบเขตของฐานข้อมูลที่จะสร้าง**

จากการวิเคราะห์ความต้องการข้อมูลสารสนเทศขององค์กร ผู้ออกแบบระบบควรจะต้องทราบว่าจะมีฐานข้อมูลที่จะสร้างขึ้นนั้น จะนำมาใช้ช่วยงานทางด้านใดขององค์กร และมีความสามารถทำงาน

เกี่ยวกับอะไรบ้าง ฐานข้อมูลนั้นจะเป็นแบบฐานข้อมูลรวม (Centralized Database) หรือเป็นฐานข้อมูลแบบกระจาย (Distributed Database)

- **ความสามารถของโปรแกรมประยุกต์ที่สร้างขึ้น**

จะต้องทราบว่าโปรแกรมประยุกต์ที่จะสร้างขึ้นจะมีความสามารถในการทำงานด้านใดบ้างเช่น

- รูปแบบการคำนวณหรือการประมวลผลข้อมูลมีขั้นตอนวิธีการอย่างไร
- กฎเกณฑ์ข้อบังคับต่าง ๆ เช่นการบันทึกข้อมูลคนงาน กำหนดว่าคนงานแต่ละคนห้ามมีความชำนาญมากกว่าหนึ่งด้าน เป็นต้น
- นอกจากนี้ยังต้องคำนึงถึงการควบคุมความคงสภาพของข้อมูลเมื่อมีการจัดการกับข้อมูลใด ๆ รวมทั้งการรักษาความปลอดภัยของระบบอีกด้วย

- **อุปกรณ์ทางด้านฮาร์ดแวร์และซอฟต์แวร์ที่จะมีการใช้**

การพิจารณาว่าควรใช้อุปกรณ์ฮาร์ดแวร์หรือซอฟต์แวร์ที่มีความสามารถมากน้อยเพียงใดนั้น จะต้องพิจารณาจากองค์ประกอบหลายอย่างเช่น ในองค์กรนั้นมีผู้ที่ต้องการใช้งานฐานข้อมูลมากน้อยเพียงใด จำนวนรายการเปลี่ยนแปลง (transactions) ที่จะต้องประมวลผลในแต่ละวัน และจำนวนความต้องการในการพิมพ์ข้อมูล เป็นต้น ข้อมูลเหล่านี้จะนำมาช่วยใช้ตัดสินใจว่าถึงขนาดและประเภทของคอมพิวเตอร์จำนวนเนื้อที่ของดิสก์รวม ไปถึงประเภทของ DBMS ที่ต้องการ ซึ่งฮาร์ดแวร์และซอฟต์แวร์นี้ก็จะมีราคาที่แตกต่างกัน ไปขึ้นกับความสามารถของฮาร์ดแวร์และซอฟต์แวร์นั้น ๆ

- **การวางแผนระยะเวลาในการทำงาน**

การวางแผนจะเริ่มตั้งแต่การออกแบบฐานข้อมูล การลงมือเขียนโปรแกรม การแก้ไขข้อผิดพลาดของโปรแกรม การทำเอกสารประกอบการทำงานและการติดตั้งระบบ ซึ่งต้องมีการกะระยะเวลาที่ต้องใช้ในแต่ละขั้นตอนเพื่อจะให้การดำเนินงานสามารถบรรลุตามระยะเวลาที่ตั้งไว้

#### 4. การออกแบบฐานข้อมูล (Database Design)

หลังจากที่ได้เป้าหมายของงานที่ชัดเจนแล้วว่าในระบบใหม่จะต้องทำอะไร มีการออกรายงานอะไรและใช้ข้อมูลใดบ้าง ก็จะมาเริ่มทำการออกแบบฐานข้อมูลซึ่งได้แก่การวิเคราะห์หาเอนทิตีหรือรีเลชัน การวิเคราะห์หาแอททริบิวต์และคีย์ของเอนทิตีหรือรีเลชัน รวมไปถึงการกำหนดความสัมพันธ์ระหว่างเอนทิตีหรือรีเลชัน โดยทั่วไปการออกแบบฐานข้อมูลจะมีอยู่ 3 แบบดังต่อไปนี้

- **การออกแบบฐานข้อมูลในระดับความคิด (Conceptual Database Design)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นการนำเสนอระบบฐานข้อมูลในลักษณะของแผนภาพโดยอาจใช้โมเดลแบบ E-R ซึ่งจะมีการแสดงเอนทิตีทั้งหมดที่มี แอททริบิวต์ของแต่ละเอนทิตีนั้น และความสัมพันธ์ระหว่างเอนทิตีออกมาในรูปแบบของแผนภาพ ข้อดีของโมเดล E-R คือจะสามารถทำความเข้าใจได้ง่าย ทำให้เห็นภาพรวมของฐานข้อมูลทั้งระบบและนอกจากนี้โมเดลที่ได้จะมีความเป็นอิสระจากระบบจัดการฐานข้อมูลหรือ DBMS ที่ใช้ โดยไม่สนใจว่าระบบจัดการฐานข้อมูลที่ใช้จะขึ้นกับโมเดลของฐานข้อมูลรูปแบบใด (เชิงสัมพันธ์ เครื่องข่ายหรือลำดับขั้น) และก็ยังไม่ขึ้นกับฮาร์ดแวร์ใด ๆ อีกด้วย (การออกแบบ โมเดล E-R ได้กล่าวในบทที่ 4)

หลังจากที่สร้าง โมเดล E-R ออกมาได้แล้ว ก็จะต้องมีการแปลงโมเดล E-R ที่ได้ให้อยู่ในรูปแบบของโมเดลแบบอื่นที่สอดคล้องกับระบบจัดการฐานข้อมูลที่ใช้ เช่นถ้าระบบจัดการฐานข้อมูลที่ใช้ขึ้นกับโมเดลเชิงสัมพันธ์ ก็จะต้องแปลง โมเดล E-R นั้นให้อยู่ในรูปแบบของรีเลชันต่าง ๆ ที่อยู่ในรูปแบบที่นอร์มัลไลซ์ ซึ่งก็จะมีขั้นตอนในการแปลงอันเป็นเรื่องที่เราจะได้ศึกษากันในหัวข้อ 6.3 ต่อไป

- **การออกแบบฐานข้อมูลในระดับตรรก (Logical Database Design)**

หลังจากขั้นตอนที่ 3 คือการวิเคราะห์ความต้องการของผู้ใช้ และรวบรวมกฎเกณฑ์ต่าง ๆ อันพึงมีได้แล้ว เราอาจทำการออกแบบฐานข้อมูลในระดับตรรกนี้ได้เลย โดยการใช้โมเดลฐานข้อมูลที่สอดคล้องกับระบบจัดการฐานข้อมูลที่ใช้ เช่นทำการออกแบบฐานข้อมูลโดยใช้โมเดลเชิงสัมพันธ์ ถ้าระบบจัดการฐานข้อมูลอิงกับโมเดลเชิงสัมพันธ์ ซึ่งการออกแบบในระดับตรรกนี้ไม่ต้องมีการออกแบบในแนวความคิด นั่นคือไม่ต้องมีการสร้างแผนภาพ E-R มาก่อน ซึ่งก็เป็นวิธีที่มีผู้นิยมกันพอสมควร แต่ทั้งนี้ก็ต้องทราบถึงกระบวนการในการออกแบบ อันเป็นหัวข้อที่เราจะได้กล่าวถึงในหัวข้อที่ 6.4 ต่อไป

- **การออกแบบฐานข้อมูลในระดับกายภาพ (Physical Database Design)**

เป็นขั้นตอนการออกแบบในระดับล่างสุด ซึ่งจะยุ่งเกี่ยวกับการจัดเก็บข้อมูลจริง ๆ ภายในหน่วยเก็บข้อมูล เช่นดิสก์ เพื่อให้สามารถเพิ่มประสิทธิภาพในการเข้าถึงหรือการค้นหาข้อมูล ในขั้นตอนนี้อาจเป็นการสร้างอินเด็กซ์ (index) การจัดคลัสเตอร์ (Clustering) ซึ่งเป็นการจัดเก็บข้อมูลที่มีการใช้งานบ่อย ๆ ไว้ในหน่วยเก็บข้อมูลเดียวกัน หรือการใช้เทคนิคแฮชชิง (Hashing Technique) ในการจัดตำแหน่งที่อยู่ของข้อมูลภายในหน่วยเก็บ เป็นต้น

จากการออกแบบฐานข้อมูลทั้ง 3 แบบข้างต้น ผู้ออกแบบฐานข้อมูลจะสามารถเลือกได้ว่า จะทำการออกแบบฐานข้อมูลในระดับตรรก โดยข้ามการออกแบบในระดับความคิดไปก็ได้ แต่โดยทั่วไปแล้วการออกแบบฐานข้อมูลที่มีโครงสร้างที่ใหญ่และซับซ้อน การเลือกใช้การออกแบบในระดับความคิดก็จะเหมาะสม เนื่องจากจะสามารถมองเห็นภาพฐานข้อมูลทั้งระบบและความสัมพันธ์ระหว่างเอนทิตีได้อย่างเด่นชัด

## 5. การออกแบบและพัฒนาโปรแกรม (Implementation)

ในขั้นตอนนี้จะมีการเลือกระบบจัดการฐานข้อมูลขึ้นมาใช้ และผู้ออกแบบระบบซึ่งอาจเป็นนักเอกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับว่าเหมาะไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์ระบบหรือผู้ออกแบบฐานข้อมูล จะทำการออกแบบโปรแกรมว่าระบบจะต้องประกอบด้วยโปรแกรมใดบ้าง แต่ละโปรแกรมมีหน้าที่อะไร และมีความสัมพันธ์กันอย่างไร การเชื่อมโยงระหว่างโปรแกรมจะอย่างไร นอกจากนี้ยังต้องมีการออกแบบหน้าจอการนำข้อมูลเข้า รูปแบบรายงาน และการควบคุมความคงสภาพของฐานข้อมูล ซึ่งจะนำมาสร้างเป็นเอกสารที่เรียกว่าข้อมูลการออกแบบโปรแกรม (Program Specification) เพื่อเตรียมส่งให้กับนักเขียนโปรแกรมหรือโปรแกรมเมอร์ใช้เป็นแบบในการเขียนโปรแกรมต่อไป

ในขั้นตอนการพัฒนาโปรแกรม โปรแกรมเมอร์จะทำการเขียนและทดสอบโปรแกรมว่าทำงานได้ถูกต้องหรือไม่ โดยจะมีการทดสอบกับข้อมูลจริงที่มีอยู่ ถ้าเป็นระบบใหญ่ที่ต้องอาศัยโปรแกรมเมอร์หลายคนช่วยกันเขียนโปรแกรม หลังจากที่แต่ละคนทำการทดสอบโปรแกรม (Program Testing) ของตนเองเสร็จแล้ว ก็จะนำโปรแกรมเหล่านั้นรวมกันให้เป็นระบบเดียว แล้วทำการทดสอบอีกที ซึ่งจะเรียกว่าการทดสอบระบบ (System Testing) โดยทั่วไปแล้วการแยกทดสอบเฉพาะโปรแกรมมักจะผ่าน แต่เมื่อมีการทดสอบระบบมักจะไม่ผ่านเนื่องจากโปรแกรมเมอร์แต่ละคน อาจมีความเข้าใจในงานไม่ตรงกัน จึงทำงานไม่ประสานกัน ดังนั้นการทดสอบระบบจึงเป็นเรื่องที่สำคัญ จะต้องทำการแก้ไขจนกว่าจะผ่านได้ และต้องมีการทดสอบข้อมูลนำเข้า เพื่อทดสอบการทำงานของระบบว่าถูกต้องตามต้องการหรือไม่ด้วย

## 6. การทำเอกสารประกอบโปรแกรม (Documentation)

การทำเอกสารประกอบโปรแกรม คือการอธิบายในรายละเอียดของโปรแกรมว่า จุดประสงค์ของโปรแกรมคืออะไร ใช้งานในด้านไหน ฯลฯ ซึ่งอาจจะเป็นการสรุปรายละเอียดของโปรแกรม และแสดงเป็นผังงาน (Flowchart) หรือรหัสจำลอง (Pseudo\_code) ก็ได้

โปรแกรมเมอร์ที่ดีควร จะมีการทำเอกสารประกอบโปรแกรมทุกขั้นตอนของการพัฒนาโปรแกรม ไม่ว่าจะเป็นขั้นตอนการออกแบบ การเขียนโปรแกรม หรือขั้นตอนการทดสอบโปรแกรม ซึ่งการทำเอกสารนี้จะมีประโยชน์อย่างมากต่อหน่วยงาน เนื่องจากบางครั้งอาจต้องกำรเปลี่ยนแปลงแก้ไขโปรแกรมที่ได้มีการทำเสร็จไปนานแล้ว เพื่อให้ตรงกับความต้องการที่เปลี่ยนไป จะทำให้เข้าใจโปรแกรมได้ง่ายขึ้นและจะเป็นการสะดวกต่อผู้ที่ต้องเข้ามารับช่วงงานต่อทีหลัง

### ● เอกสารประกอบโปรแกรมสำหรับผู้ใช้งาน (User Documentation)

จะเหมาะสำหรับผู้ใช้งานที่ไม่ต้องเกี่ยวข้องกับการพัฒนาโปรแกรม แต่เป็นผู้ที่ใช้งานโปรแกรมอย่างเดียว จะอธิบายเกี่ยวกับการใช้โปรแกรม ตัวอย่างเช่น

- ◆ โปรแกรมนี้ทำอะไร ใช้งานในด้านไหน
- ◆ ข้อมูลเข้ามีลักษณะอย่างไร
- ◆ ข้อมูลออกหรือผลลัพธ์มีลักษณะอย่างไร
- ◆ การเรียกใช้โปรแกรมทำอย่างไร
- ◆ คำสั่ง หรือข้อมูล ที่จำเป็นให้โปรแกรมเริ่มทำงาน มีอะไรบ้าง

### ◆ อธิบายเกี่ยวกับประสิทธิภาพ และความสามารถของโปรแกรม

เอกสารนี้เป็นเอกสารที่ส่งมอบให้สำหรับผู้ใช้งานเพื่อใช้ในการศึกษาหาข้อมูลเพิ่มเติม อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เอกสารประกอบโปรแกรมสำหรับผู้เขียนโปรแกรม (Technical Documentation)

จะแบ่งออกได้เป็น 2 ส่วน

- ◆ ส่วนที่เป็นคำอธิบายหรือหมายเหตุในโปรแกรม หรือเรียกอีกอย่างหนึ่งว่าคอมเมนต์
- ◆ (Comment) ซึ่งส่วนใหญ่ผู้เขียนมักจะเขียนแทรกอยู่ในโปรแกรม อธิบายการทำงานของโปรแกรมเป็น ส่วน ๆ
- ◆ ส่วนอธิบายด้านเทคนิค มักจะทำเป็นเอกสารแยกต่างหากจากโปรแกรม ซึ่งจะอธิบายในรายละเอียดที่มากขึ้น เช่นชื่อโปรแกรมย่อยต่าง ๆ มีอะไรบ้าง แต่ละโปรแกรมย่อยทำหน้าที่อะไร และคำอธิบายย่อ ๆ เกี่ยวกับวัตถุประสงค์ของโปรแกรม เป็นต้น

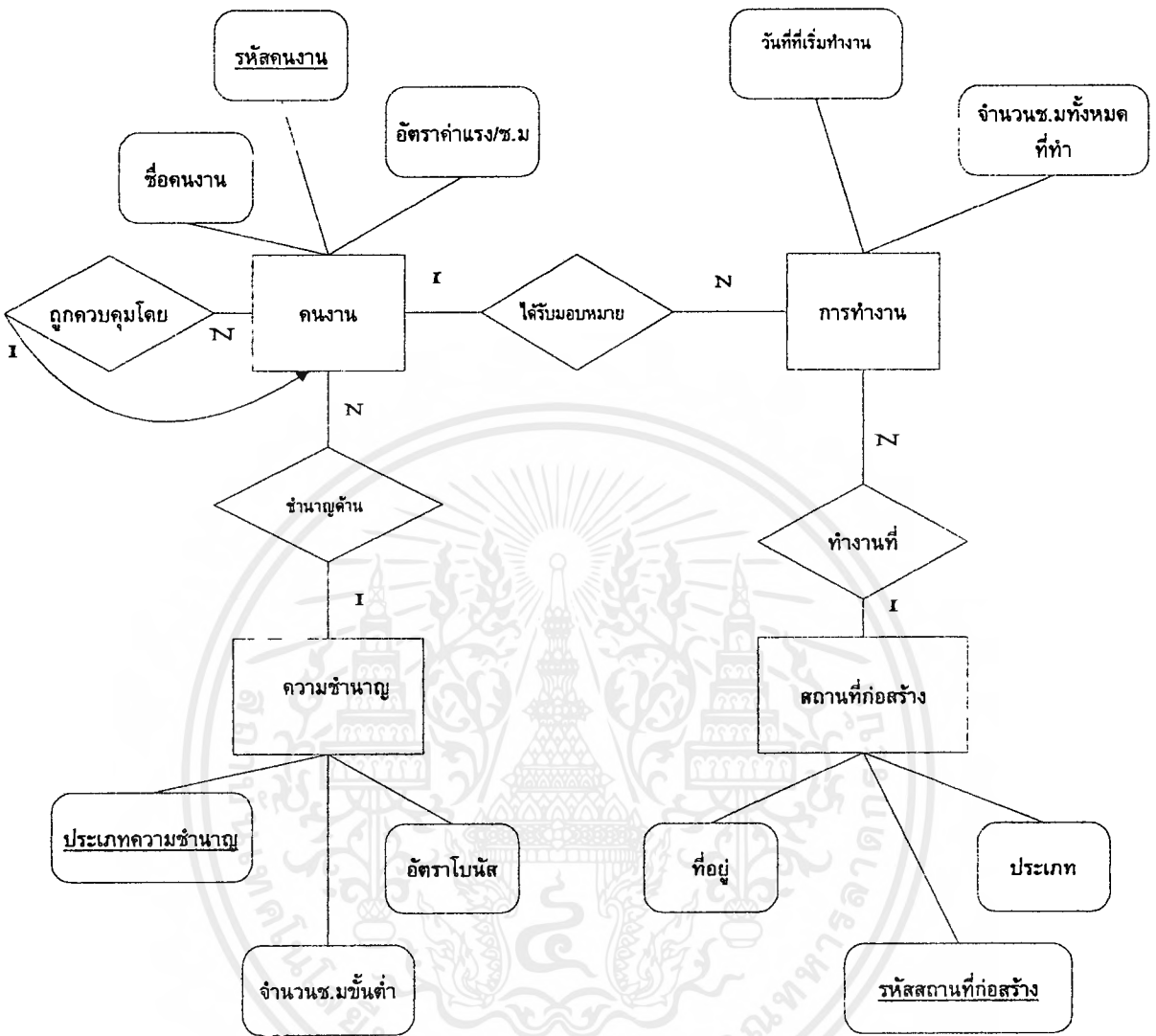
## 7. การติดตั้งและการบำรุงรักษาโปรแกรม (Program Maintenance)

เมื่อโปรแกรมผ่านการตรวจสอบตามขั้นตอนเรียบร้อยแล้ว และถูกนำมาติดตั้งให้ผู้ใช้ได้ใช้งาน ในขั้นตอนนี้จะรวมไปถึงการฝึกอบรมให้แก่ผู้ใช้ซึ่งอาจเป็นพนักงานที่ต้องใช้งานจริง เพื่อให้เข้าใจการทำงานและทำงานได้โดยไม่มีปัญหา ซึ่งในช่วงแรกผู้ใช้อาจจะไม่คุ้นเคย ก็จะทำให้เกิดปัญหาขึ้นมาบ้าง ดังนั้นจึงต้องมีผู้คอยควบคุมดูแลและตรวจสอบการทำงาน และเมื่อมีการใช้งานไปนาน ๆ ก็อาจจะต้องการปรับปรุงแก้ไขโปรแกรมให้เหมาะกับเหตุการณ์ และความต้องการของผู้ใช้ที่เปลี่ยนแปลงไปได้

### 2.4.3 การแปลงโมเดลแบบ E-R เป็นรูปแบบโมเดลเชิงสัมพันธ์

การออกแบบฐานข้อมูลในระดับความคิดในรูปแบบของโมเดล E-R นั้น ได้กล่าวไว้แล้วในบทที่ 4 เพื่อให้สามารถใช้งานฐานข้อมูลนั้นต่อไปได้ จะต้องทำการแปลงโมเดลนี้ให้อยู่ในรูปแบบของโมเดลอื่นที่สอดคล้องกับระบบจัดการฐานข้อมูลที่ใช้ และเนื่องจากเราจะมุ่งเน้นให้ความสนใจกับ โมเดลเชิงสัมพันธ์ ถ้าโมเดล E-R นั้นเป็นโมเดลที่ถูกออกแบบมาในลักษณะที่ผ่านการนอร์มัลไลซ์มาเรียบร้อยแล้ว คือวิเคราะห์ได้อย่างถูกต้องว่าควรมีเอนติตี้ใดบ้าง แต่ละเอนติตี้มีแอททริบิวต์อะไร มีความสัมพันธ์ระหว่างเอนติตี้เป็นอย่างไร ก็สามารถรับประกันได้ว่ารีเลชันที่ถูกแหล่งออกมานั้น เป็นรีเลชันที่อยู่ในรูปแบบที่นอร์มัลไลซ์แล้ว แต่ถ้าโมเดลแบบ E-R ถูกออกแบบในลักษณะที่ยังไม่ได้นอร์มัลไลซ์รีเลชันที่แปลงได้ก็จำเป็นที่จะต้องถูกนำมาผ่านการนอร์มัลไลซ์อีกที

จะยกตัวอย่างแผนภาพ E-R ของบริษัทสุจริตการก่อสร้างมาใช้เป็น ตัวอย่างในการแปลงจากแผนภาพดังกล่าวให้อยู่ในรูปแบบของโมเดลเชิงสัมพันธ์ ดังรูป 2.17



รูปที่ 2.17 แผนภาพแบบ E-R ของบริษัทสุจริตการก่อสร้าง จำกัด

ขั้นตอนการแปลงจะมีดังต่อไปนี้

**1.แปลงเอนทิตีปกติกและแอททริบิวท์ของเอนทิตี**

เป็นขั้นตอนการแปลงโดยตรงไปตรงมา โดยจะทำการแปลงเอนทิตีปกติกที่ไม่ใช่เอนทิตีแบบอ่อนให้อยู่ในรูปรีเลชัน ซึ่งชื่อของเอนทิตี ก็คือชื่อของรีเลชันและแอททริบิวท์ของเอนทิตีก็จะเป็นแอททริบิวท์ของรีเลชัน

จากแผนภาพรูป 6.1 จะสามารถแปลงเอนทิตีปกติกแต่ละเอนทิตีให้อยู่ในรูปของรีเลชันได้ดังนี้

**คนงาน**(รหัสคนงาน,ชื่อคนงาน,อัตราค่าแรง/ชม.)

**ความชำนาญ**(ประเภทความชำนาญ,อัตราโบนัส,จำนวนชม.ขั้นต่ำ)

**สถานที่ก่อสร้าง**(รหัสสถานที่ก่อสร้าง,ที่อยู่,ประเภทสถานที่ก่อสร้าง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อแปลงเอนทิตีปกติให้อยู่ในรูปของรีเลชันแล้ว ให้ทำการพิจารณาลักษณะข้อมูลของแต่ละเรลชันนั้นว่าจะมีแอททริบิวต์ใดที่สามารถใช้เป็นคีย์หลักของรีเลชันได้ และทำการขีดเส้นใต้ที่แอททริบิวต์ที่เป็นคีย์หลักนั้น

## 2. แปลงเอนทิตีแบบอ่อน

ดังได้เคยกล่าวไว้ในบทที่ 5 ว่าเอนทิตีแบบอ่อน (Weak Entity) จะเป็นเอนทิตีที่สามารถปรากฏอยู่ในรูปฐานข้อมูลได้ ก็ต่อเมื่อต้องมีเอนทิตีปรากฏอยู่ในฐานข้อมูลด้วย ซึ่งจากตัวอย่างแผนภาพ E-R ในรูปที่ 6.1 นี้เอนทิตีการทำงานจะเป็นเอนทิตีแบบอ่อน เนื่องจากการจะมีเอนทิตีการทำงานอยู่ในฐานข้อมูลนี้ได้ จะต้องต้องมีเอนทิตีคนงานและเอนทิตีสถานที่ก่อสร้างปรากฏอยู่ด้วย และมีลักษณะการขึ้นต่อกันทั้งแบบเชิงปรากฏและแบบเชิงระบุ เนื่องจากถ้าสร้างรีเลชันการทำงานที่ประกอบด้วยแอททริบิวต์วันที่เริ่มทำงาน และจำนวนชม.ทั้งหมดที่ทำเท่านั้น ดังนี้

การทำงาน (วันที่เริ่มทำงาน, จำนวนชม.ทั้งหมดที่ทำ)

รีเลชันข้างต้นนี้เราจะไม่สามารถหาคีย์ของรีเลชันได้ เนื่องจากอาจมีแถวข้อมูลที่ซ้ำกัน เช่นมีแถวข้อมูลที่มีวันเริ่มทำงาน และจำนวนชม.ทั้งหมดที่ทำ ซ้ำกันมากกว่าหนึ่งแถวขึ้นไป

ดังนั้นเอนทิตีใดที่มีคุณสมบัติเป็นเอนทิตีแบบอ่อนและมีการขึ้นต่อกันเชิงระบุอยู่ จำเป็นที่จะต้องนำแอททริบิวต์ที่เป็นคีย์ของเอนทิตีที่มันขึ้นอยู่กับนั้น มาร่วมกันเป็นคีย์ของรีเลชันแทน ดังนั้นรีเลชันการทำงานข้างต้น ควรจะต้องนำรหัสคนงานจากเอนทิตีคนงาน และรหัสสถานที่ก่อสร้างจากเอนทิตีสถานที่ก่อสร้าง มาใช้ร่วมกันเป็นคีย์ของรีเลชันการทำงานนี้

การทำงาน (รหัสคนงาน, รหัสสถานที่ก่อสร้าง, วันที่เริ่มทำงาน, จำนวนชม.ทั้งหมดที่ทำ)

รหัสคนงานและรหัสสถานที่ก่อสร้าง เพียงสองแอททริบิวต์นี้ก็สามารถใช้อ้างอิงถึงข้อมูลในแถวใดแถวหนึ่งของรีเลชันได้ ดังนั้นจึงถูกเลือกให้เป็นคีย์หลักของรีเลชันการทำงาน

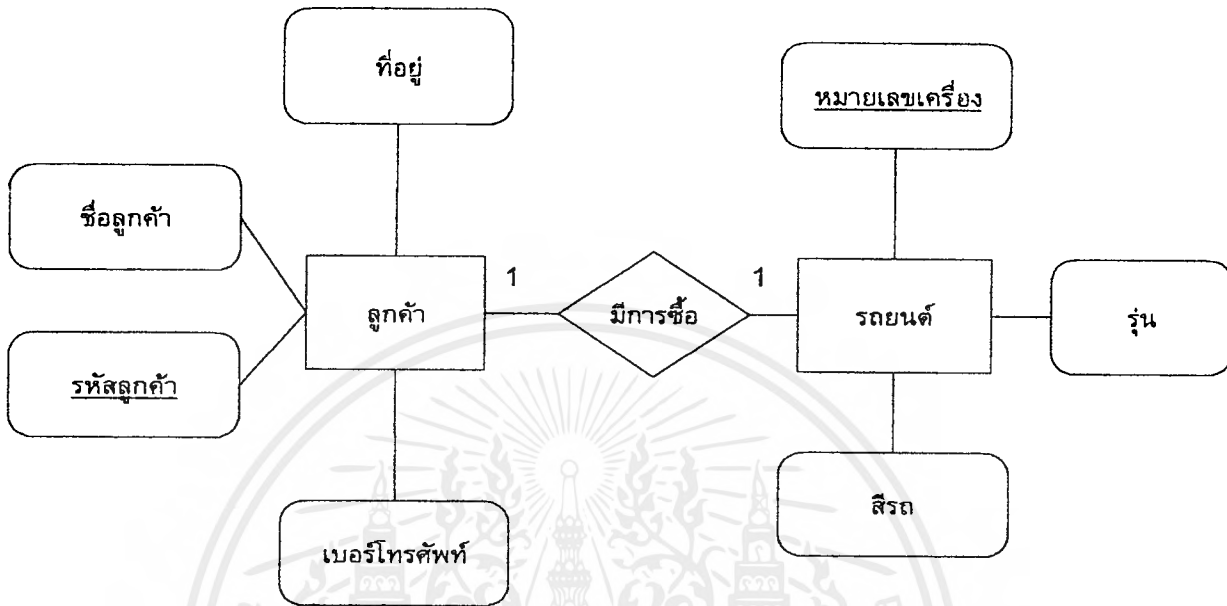
## 3. แปลงความสัมพันธ์ระหว่างเอนทิตี

ดังได้กล่าวแล้วว่าลักษณะความสัมพันธ์ระหว่างเอนทิตีนี้ จะมีอยู่กันสามแบบอันได้แก่แบบหนึ่งต่อหนึ่ง (1:1) แบบหนึ่งต่อกลุ่ม (1:N) และ กลุ่มต่อกลุ่ม (N:M) ซึ่งความสัมพันธ์แต่ละแบบนี้จะมีลักษณะการแปลงให้อยู่ในรูปของรีเลชันที่แตกต่างกัน ดังรายละเอียดต่อไปนี้

- การแทนความสัมพันธ์แบบหนึ่งต่อหนึ่ง

จะขอยกตัวอย่างเดิมที่ได้กล่าวไว้ในบทที่ 5 ถึงรูปแบบความสัมพันธ์แบบหนึ่งต่อหนึ่งของการ

ซื้อรถยนต์ โดยกำหนดว่าลูกค้าหนึ่งคน จะซื้อรถยนต์ได้เพียงหนึ่งคัน และรถยนต์หนึ่งคัน ก็จะเป็นของ  
ลูกค้าได้เพียงคนเดียว ดังรูป 2.18



รูปที่ 2.18 แสดงตัวอย่างความสัมพันธ์แบบหนึ่งต่อหนึ่ง

ถ้าเอนทิตีลูกค้ามีแอททริบิวท์รหัสลูกค้าเป็นคีย์หลัก และเอนทิตีรถยนต์มีหมายเลขเครื่องเป็นคีย์หลัก การแทนความสัมพันธ์แบบหนึ่งต่อหนึ่งให้อยู่ในรูปของรีเลชันจะทำโดย

1. แปลงเอนทิตีแต่ละเอนทิตีให้เป็นรีเลชัน โดยชื่อเอนทิตี ก็คือชื่อรีเลชันและกำหนดคีย์หลักของแต่ละรีเลชัน ดังนี้

ลูกค้า(รหัสลูกค้า, ชื่อลูกค้า, ที่อยู่, เบอร์โทรศัพท์)

รถยนต์(หมายเลขเครื่อง, รุ่น, สีรถ)

2. เพิ่มคีย์ของรีเลชันหนึ่งลงไปเป็นแอททริบิวท์หนึ่งของอีกรีเลชัน

ดังนั้นในขั้นตอนนี้จะสามารถเลือกได้ว่า จะนำแอททริบิวท์รหัสลูกค้าเพิ่มลงในรีเลชันรถยนต์ หรือนำแอททริบิวท์หมายเลขเครื่องเพิ่มลงไปในรีเลชันลูกค้า ดังนี้

รูปแบบที่ 1 ลูกค้า(รหัสลูกค้า, ชื่อลูกค้า, ที่อยู่, เบอร์โทรศัพท์)

รถยนต์(หมายเลขเครื่อง, รุ่น, สีรถ, รหัสลูกค้า)

รูปแบบที่ 2

ลูกค้า(รหัสลูกค้า, ชื่อลูกค้า, ที่อยู่, เบอร์โทรศัพท์, หมายเลขเครื่อง)

รถยนต์(หมายเลขเครื่อง, รุ่น, สีรถ, รหัสลูกค้า)

การนำคีย์ของรีเลชันหนึ่งไปเก็บอยู่ในอีกรีเลชันหนึ่งที่มีความสัมพันธ์กันนั้น คีย์นั้นจะถูกเรียกว่า เป็นคีย์นอก (Foreign Key) ดังนั้นในรูปแบบที่ 1 ข้างต้น แอททริบิวท์รหัสลูกค้าจะเป็นคีย์นอกในรีเลชันรถยนต์ ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายนต์ และในรูปแบบที่ 2 แอททริบิวต์หมายเลขเครื่องจะเป็นคีย์นอกในรีเลชันของลูกค้า (ในที่นี้จะแสดงด้วยการขีดเส้นได้ด้วยเส้นประ)

- การแทนความสัมพันธ์แบบหนึ่งต่อกลุ่ม

จากตัวอย่างแผนภาพแบบ E-R ในรูป 6.1 พิจารณาความสัมพันธ์แบบหนึ่งต่อกลุ่ม (1:N) ที่ไม่ใช่ความสัมพันธ์ที่เชื่อมกับเอนติตีแบบอ่อน จะเห็นว่ามีความสัมพันธ์ที่ชื่อ "ชานาญด้าน" อันเป็นความสัมพันธ์ที่เชื่อมระหว่างเอนติตีคนงานและความชานาญ จะมีขั้นตอนการแปลงความสัมพันธ์นี้ให้อยู่ในรูปแบบของรีเลชันได้ดังนี้

1. แปลงเอนติตีปกติแต่ละเอนติตีให้เป็นรีเลชัน โดยชื่อเอนติตีก็คือชื่อรีเลชัน และกำหนดคีย์หลักของแต่ละรีเลชัน ซึ่งขั้นตอนนี้ได้กระทำไปแล้วในขั้นตอนก่อนหน้า และจะมีรีเลชันดังนี้

คนงาน(รหัสคนงาน, ชื่อคนงาน, อัตราค่าแรง/ชม.)

ความชานาญ(ประเภทความชานาญ, อัตรา โบนัส, จำนวนชม.ขั้นต่ำ)

2. นำคีย์หลักของเอนติตีที่อยู่ด้านความสัมพันธ์เป็น 1 ไปเก็บเป็นแอททริบิวต์ของรีเลชันด้านที่มีความสัมพันธ์เป็น N

ดังนั้นจากรีเลชันที่ได้ข้างต้น แอททริบิวต์ประเภทความชานาญในเอนติตีความชานาญ จะถูกนำไปเก็บเป็นแอททริบิวต์หนึ่งของรีเลชันคนงาน รีเลชันที่แปลงใหม่จะเป็นดังนี้

รูปแบบรีเลชันข้างต้น จะมีการนำรหัสคนงานซึ่งเป็นคีย์หลักในรีเลชันคนงาน ไปใส่ไว้ในรีเลชันสถานที่ก่อสร้าง ซึ่งการแปลงความสัมพันธ์แบบ N:M โดยวิธีข้างต้น จะเป็นวิธีการแปลงที่ไม่ถูกต้อง เนื่องจากทำให้เกิดปัญหาเกี่ยวกับความผิดปกติที่เกี่ยวกับการจัดการข้อมูล ไม่ว่าจะเป็นการเพิ่ม ลบ หรือปรับปรุงข้อมูล ดังตัวอย่างที่จะแสดงให้เห็นดังต่อไปนี้

**คนงาน**

รหัส คนงาน	แอททริบิวต์อื่น ๆ
1245	.....
2521	.....
1411	.....

**สถานที่ก่อสร้าง**

รหัสสถานที่ ก่อสร้าง	ที่อยู่สถานที่ ก่อสร้าง	ประเภทของ สถานที่ก่อสร้าง	รหัสคนงาน
516	128 ลาดพร้าว 78	บ้านพักอาศัย	1245
516	128 ลาดพร้าว 78	บ้านพักอาศัย	2521
311	25 สีลม	สำนักงาน	2521
450	18 พหลโยธิน 55	ร้านค้า	1411

รูปที่ 2.19 แสดงตัวอย่างข้อมูลภายในรีเลชันที่มีการแปลงอย่างไม่ถูกต้อง

จากรูป 2.19 จะเห็นว่ารหัสสถานที่ก่อสร้าง 516 จะมีคนงานทำงานได้มากกว่าหนึ่งคนขึ้นไป คือ มีคนงานที่มีรหัส 1245 และ 2521 ทำงานอยู่ และในทางกลับกัน คนงานรหัส 2521 ก็สามารถทำงานยังสถานที่ก่อสร้างได้หลายแห่ง ได้แก่สถานที่ก่อสร้างรหัส 516 และ 311 เป็นต้น ซึ่งเป็นลักษณะความสัมพันธ์แบบกลุ่มต่อกลุ่ม

ปัญหาที่จะเกิดขึ้นจากการแปลงรีเลชันดังรูปแบบที่ 2.19 นี้ ก็คือปัญหาทางด้านการจัดการข้อมูล ตัวอย่างเช่น ถ้าคนงานที่มีรหัส 1411 ได้ลาออกไป การลบข้อมูลในรีเลชันสถานที่ก่อสร้างในแถวที่มีคนงานรหัส 1411 นั้นอยู่ จะทำให้เราต้องสูญเสียข้อมูลเกี่ยวกับสถานที่ก่อสร้างที่มีรหัส 450 ไปด้วย หรือถ้าบริษัทรับที่จะก่อสร้างยังสถานที่แห่งใหม่แห่งหนึ่ง แต่ยังไม่ได้กำหนดว่าจะให้ใครไปทำการก่อสร้างที่นั่นบ้าง จะทำให้เราไม่สามารถเพิ่มข้อมูลเกี่ยวกับสถานที่ก่อสร้างแห่งใหม่เข้าไปยังรีเลชันสถานที่ก่อสร้างได้ (ถ้ามีการกำหนดไว้ว่ารหัสคนงานจะต้องไม่เป็นค่าว่าง) เป็นต้น

แนวทางแก้ไขสำหรับการแปลงเอนติตี้ที่มีความสัมพันธ์แบบ N:M ให้เป็นรีเลชันนั้น จะทำได้ดังนี้

1. แปลงเอนติตี้ที่มีความสัมพันธ์กันนั้นให้เป็นรีเลชัน โดยชื่อเอนติตี้ก็คือชื่อรีเลชันและ

แอททริบิวต์ของเอนติตี้ก็คือแอททริบิวต์ของรีเลชัน ดังนั้นจะมีรีเลชันดังนี้

คนงาน(รหัสคนงาน,ชื่อคนงาน,ประเภทความชำนาญ,อัตราค่าแรง/ชม)

สถานที่ก่อสร้าง(รหัสสถานที่ก่อสร้าง,ที่อยู่,ประเภทสถานที่ก่อสร้าง)

2. สร้างรีเลชันขึ้นมาใหม่หนึ่งรีเลชัน โดยรีเลชันที่สร้างขึ้นมานี้จะประกอบด้วยแอททริบิวต์ของความสัมพันธ์แบบ N:M และจะมีการนำคีย์หลักของรีเลชันที่สัมพันธ์กันแบบ N:M นั้น มาสร้างเป็นคีย์หลักอยู่ในรีเลชันใหม่ด้วย

ดังนั้นถ้ามีการสร้างรีเลชันใหม่ โดยตั้งชื่อรีเลชันใหม่นี้ว่ารีเลชันการทำงาน ซึ่งประกอบด้วยแอททริบิวท์ วันที่เริ่มทำงาน และจำนวนชม.ทั้งหมดที่ทำ แล้วคีย์หลักจากรีเลชันคนงาน อันได้แก่รหัสคนงาน และคีย์หลักจากรีเลชันสถานที่ก่อสร้าง อันได้แก่รหัสสถานที่ก่อสร้าง มาไว้รวมกันในรีเลชันการทำงานด้วย ดังนี้

การทำงาน(รหัสคนงาน,รหัสสถานที่ก่อสร้าง,วันที่เริ่มทำงาน,จำนวนชม.ทั้งหมดที่ทำ)

3.ความสัมพันธ์ระหว่างรีเลชันเดิม กับรีเลชันใหม่นั้นจะเปลี่ยนไปเป็นแบบหนึ่งต่อกลุ่ม (1:N)

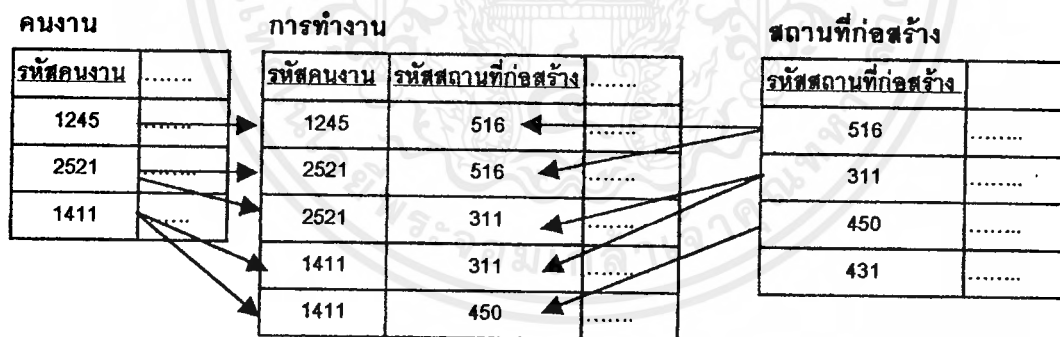
ดังนั้นจากรูปแบบแผนภาพ E-R ในรูป 6.3 เมื่อแปลงให้อยู่ในรูปของรีเลชัน จะทำให้เกิดรีเลชันขึ้นมา 3 รีเลชันดังนี้

คนงาน(รหัสคนงาน,ชื่อคนงาน,ประเภทความชำนาญ,อัตราค่าแรง/ชม.)

สถานที่ก่อสร้าง(รหัสสถานที่ก่อสร้าง,ที่อยู่,ประเภทสถานที่ก่อสร้าง)

การทำงาน(รหัสคนงาน,รหัสสถานที่ก่อสร้าง,วันที่เริ่มทำงาน,จำนวนชม.ทั้งหมดที่ทำ)

ความสัมพันธ์ระหว่างรีเลชันคนงานและการทำงาน จะเป็นแบบ 1:N และความสัมพันธ์ระหว่างรีเลชันสถานที่ก่อสร้างและการทำงาน ก็จะเป็น แบบ 1:N ด้วยเช่นกัน

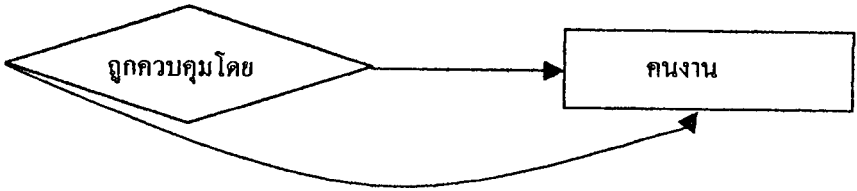


รูปที่ 2.20 แสดงตัวอย่างความสัมพันธ์ของข้อมูลภายในรีเลชันคนงานและการทำงานที่เป็นแบบ 1:N และรีเลชันสถานที่ก่อสร้างกับการทำงานก็เป็นแบบ 1:N

• การแทนความสัมพันธ์แบบรีเคอร์ซีฟ

ความสัมพันธ์แบบรีเคอร์ซีฟ เป็นความสัมพันธ์ระหว่างเอนติตี้ที่เป็นชนิดเดียวกันหรือสมาชิกภายในเอนติตตินั้นอยู่ในโดเมนเดียวกัน ดังตัวอย่างจากแผนภาพในรูป 6.1 พิจารณาเอนติตี้คนงาน ที่มีความสัมพันธ์ชื่อ “ถูกควบคุมโดย”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 แสดงความสัมพันธ์รีเคอร์ซีฟแบบ 1:N

จากรูป 2.21 หมายความว่าคนงานบางคนอาจถูกเลือกให้เป็นผู้ควบคุมคนงานด้วยตัวเองได้ โดยผู้ควบคุมหนึ่งคนจะสามารถควบคุมคนงานได้หลายคน แต่คนงานแต่ละคนจะต้องมีผู้ควบคุมเพียงคนเดียวเท่านั้น ซึ่งเป็นความสัมพันธ์แบบ 1:N

การแปลงความสัมพันธ์แบบรีเคอร์ซีฟให้เป็นรีเลชัน จะมีขั้นตอนดังนี้  
1 แปลงเอนติตีนั้นให้เป็นรีเลชัน โดยมีชื่อเอนติตีคือชื่อรีเลชัน และแอททริบิวต์ของเอนติตีก็คือแอททริบิวต์ของรีเลชัน ดังนี้

คนงาน(รหัสคนงาน,ชื่อคนงาน,ประเภทความชำนาญ,อัตราค่าแรง/ชม)

2. ใช้หลักการแปลงความสัมพันธ์แบบ 1:N โดยการนำคีย์ของรีเลชันฝั่งที่มีความสัมพันธ์เป็น 1 ไปใส่ไว้เป็นคีย์นอกในรีเลชันฝั่งที่มีความสัมพันธ์เป็น N แต่ให้ทำการเปลี่ยนชื่อคีย์นอกนั้นใหม่ แต่เนื่องจากความสัมพันธ์แบบรีเคอร์ซีฟนี้ รีเลชันที่แปลงได้จะมีเพียงรีเลชันเดียว ซึ่งเป็นรีเลชันที่มีความสัมพันธ์ภายในตัวมันเอง ดังนั้นจากขั้นตอนในขั้นที่ 2 นี้จะได้รีเลชันดังนี้

คนงาน(รหัสคนงาน,ชื่อคนงาน,ประเภทความชำนาญ,อัตราค่าแรง/ชม,รหัสผู้ควบคุม).....

หมายเหตุ : การที่ต้องมีการเปลี่ยนชื่อคีย์นอกของรีเลชันคนงานเป็นผู้ควบคุม แทนที่จะเป็นรหัสคนงานที่เหมือนกับคีย์หลักนั้น ก็เนื่องจากคุณสมบัติของรีเลชัน ที่ไม่อนุญาตให้มีแอททริบิวต์ใดที่มีชื่อเหมือนกันมากกว่าหนึ่งตัว อยู่ในรีเลชันได้

รหัสผู้ควบคุมนี้จะป็นคีย์นอกของรีเลชันคนงานที่ใช้อ้างอิงถึงคีย์หลักคือรหัสคนงานซึ่งอยู่ภายในรีเลชันเดียวกัน ซึ่งเป็นลักษณะของรีเลชันที่มีความสัมพันธ์แบบรีเคอร์ซีฟ

คนงาน(WORKER)				
รหัสคนงาน WORKER_ID	ชื่อคนงาน WORKER_NAM	อัตราค่าแรงต่อชม. HOUR_RATE	ประเภทความชำนาญ SKILL_TYPE	รหัสผู้ควบคุม คุม SUPV_ID
	E			
1245	สุดใจ ดีพร้อม	16.00	ไฟฟ้า	1411
2521	พิเชฏ มวกมี	16.00	ปูน	2522
1411	พิชัย ใจซื่อ	16.50	ไฟฟ้า	
2511	พิเชฏ มวกมี	15.00	ประปา	
2522	ด้วง เกื้อกุน	16.50	ปูน	

รูปที่ 2.22 แสดงตัวอย่างข้อมูลภายในรีเลชันคนงานที่มีความสัมพันธ์แบบรีเคอร์ซีฟ

จากขั้นตอนการแปลงแผนภาพ E-R ทั้งหมดข้างต้น จะสามารถสรุปรีเลชันที่จะเกิดขึ้นจากการแปลงแผนภาพ E-R ในรูป 2.17 ได้ดังนี้

คนงาน(รหัสคนงาน,ชื่อคนงาน,ประเภทความชำนาญ,อัตราค่าแรง/ชม,รหัสผู้ควบคุม)

คีย์นอก : ประเภทความชำนาญ ที่อ้างอิงรีเลชันความชำนาญ

รหัสผู้ควบคุม ที่อ้างอิงรีเลชันคนงาน

ความชำนาญ (ประเภทความชำนาญ,อัตราโบนัส,จำนวนชม.ขั้นต่ำ)

สถานที่ก่อสร้าง(รหัสสถานที่ก่อสร้าง,ที่อยู่,ประเภทสถานที่ก่อสร้าง)

การทำงาน (รหัสคนงาน,รหัสสถานที่ก่อสร้าง,วันที่เริ่มทำงาน,จำนวนชม.ทั้งหมดที่ทำ)

คีย์นอก : รหัสคนงาน ที่อ้างอิงรีเลชันคนงาน

รหัสสถานที่ก่อสร้าง ที่อ้างอิงรีเลชันสถานที่ก่อสร้าง

ของรีเลชันเท่านั้น จึงสามารถรับประกันได้ว่ารีเลชันทั้งหมดนี้เป็นรีเลชันถูกนอร์มัลไลซ์มาแล้ว

#### 2.4.4 การออกแบบฐานข้อมูลในระดับตรรก

การออกแบบฐานข้อมูลในระดับตรรก เป็นการออกแบบฐานข้อมูลเชิงสัมพันธ์โดยตรงหลังจากการเก็บรวบรวมข้อมูลและวิเคราะห์ความต้องการของผู้ใช้ได้แล้ว รวมทั้งกฎเกณฑ์ต่าง ๆ ที่พึงมี ก็จะเริ่มทำการแปลงความต้องการนั้นให้อยู่ในรูปของรีเลชัน โดยตรง แล้วจึงทำการนอร์มัลไลซ์รีเลชันเหล่านั้น ซึ่งจะมีกระบวนการหลัก ๆ ในการออกแบบฐานข้อมูลในระดับตรรกดังต่อไปนี้

1. การเปลี่ยนรูปแบบความต้องการให้อยู่ในรูปแบบของรีเลชันนั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ทำการนอร์มัลไลซ์รีเลชันที่ได้
3. กำหนดแอททริบิวท์ที่จะใช้เป็นคีย์ต่าง ๆ ของรีเลชัน

## 2.5 ภาษา SQL

### 2.5.1 แนะนำ

ส่วนประกอบของ DBMS ซึ่งเป็นซอฟต์แวร์ที่ใช้ในการจัดการฐานข้อมูล และมี DBMS หลายตัวที่ถูกสร้างขึ้นมาสำหรับฐานข้อมูลที่มีโมเดลเชิงสัมพันธ์ และส่วนประกอบหนึ่งคือภาษาที่เรียกว่า SQL (Structured Query Language) เป็นภาษาที่พบได้ใน DBMS เชิงสัมพันธ์หลายตัว และมีความนิยมในการใช้งานเป็นอย่างมากในปัจจุบัน เนื่องจากมีรูปแบบที่เป็นภาษาอังกฤษ ง่ายต่อการเข้าใจและเรียนรู้

เนื่องจาก SQL เป็นภาษามาตรฐานที่เป็นที่ยอมรับและมีการใช้งานกันอย่างกว้างขวาง จึงมีการขยายการใช้งาน SQL ให้สามารถใช้ได้กับเครื่องคอมพิวเตอร์ตั้งแต่ระดับพีซีไปจนถึงระดับเมนเฟรม ในเครื่องระดับใหญ่จะมี DBMS หลายตัวที่มี SQL รวมอยู่ด้วยเช่น DB , SQL/DS , Oracle , R:Base , Informix , Ingres ฯลฯ สำหรับเครื่องระดับพีซี แต่เดิม dBase หรือ FoxBase บนดอสรุ่นเก่าเป็น DBMS ที่ไม่มี SQL รวมอยู่ด้วย แต่ปัจจุบันได้มีการพัฒนาสร้าง dBase รุ่น dBase IV และ FoxPro รุ่นใหม่ที่มีพัฒนาบนวินโดวส์ที่มีความสามารถใส่คำสั่ง SQL บางคำสั่งได้

ในบทนี้จะพูดถึงรูปแบบของคำสั่ง SQL ในส่วนของภาษาที่ใช้สำหรับการนิยามข้อมูล (Data Definition Language : DDL) และภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language : DML) สำหรับภาษาควบคุม (Control Language)

ภาษา DDL และ DML ที่จะพูดถึงในบทนี้เป็น SQL ที่มีอยู่ใน DBMS สำหรับเครื่องระดับใหญ่ ส่วนรูปแบบของภาษาทั้งสองที่มีใน DBMS สำหรับเครื่องระดับเล็ก เช่น FoxPro สำหรับวินโดวส์

### 2.5.2 ภาษาสำหรับการนิยามข้อมูล (Data Definition Language -DDL)

หลังจากที่ได้มีการออกแบบฐานข้อมูลเรียบร้อยแล้ว ทำให้ทราบว่าฐานข้อมูลที่จะสร้างจะประกอบด้วยรีเลชันใดบ้าง แต่ละรีเลชันสัมพันธ์กันอย่างไร คีย์หลักของรีเลชันคืออะไร ขั้นตอนต่อไปคือการสร้างฐานข้อมูลนั้นลงบนเครื่องคอมพิวเตอร์นั้นให้สามารถทำงานได้ โดยการแปลงรีเลชันค่า ๆ ให้อยู่ในรูปของภาษาสำหรับการนิยามข้อมูล หรือ DDL

จะสามารถสรุปคำสั่งต่าง ๆ ที่เป็น DDL ที่มีอยู่ใน DBMS สำหรับเครื่องระดับใหญ่ ได้ดังนี้

คำสั่ง	การทำงาน
CREATE TABLE	นิยามโครงสร้างข้อมูลในตารางที่สร้างขึ้น
DROP TABLE	ลบโครงสร้างตารางข้อมูลออกจากระบบ
ALTER TABLE	แก้ไขปรับปรุงโครงสร้างตาราง
CREATE INDEX	สร้างดัชนีของตาราง โดยอาจใช้ 1 คอลัมน์หรือมากกว่า
DROP INDEX	ลบดัชนีของตารางออกจากระบบ
CREATE VIEW	กำหนดโครงสร้างวิวของผู้ใช้
DROP VIEW	ลบโครงสร้างวิวออกจากระบบ

รูปที่ 2.2 ตารางสรุปคำสั่งประเภท DDL

เฉพาะคำสั่ง CREATE VIEW และ DROP VIEW จำเป็นต้องมีความรู้เกี่ยวกับคำสั่ง SELECT ซึ่งเป็นคำสั่งจัดการข้อมูลในภาษา DDL มาก่อน ดังนั้นจึงขอกล่าวถึง 2 คำสั่งนี้ในตอนท้ายของบท ส่วนคำสั่งอื่น ๆ นั้นจะได้กล่าวถึงในหัวข้อต่อไป

### คำสั่งนิยามโครงสร้างตาราง

การสร้างตารางใน สำหรับเครื่องขนาดใหญ่ จะทำได้ด้วยการออกคำสั่ง แล้วตามด้วยชื่อตาราง พร้อมทั้งชื่อคอลัมน์ต่าง ๆ ที่ตั้งขึ้นในตาราง รวมไปถึงประเภทของข้อมูลของแต่ละคอลัมน์นั้น รูปแบบทั่วไปมีดังนี้

```
CREATE TABLE <ชื่อตารางที่ตั้งขึ้น>
(<ชื่อคอลัมน์ ประเภทข้อมูล>[, <ชื่อคอลัมน์ ประเภทข้อมูล>]...);
```

ตัวอย่างที่ 1 จะขอยกตัวอย่างฐานข้อมูลของบริษัทสุจริตการก่อสร้าง จำกัด ที่ได้มีการออกแบบฐานข้อมูลไว้แล้วในบทก่อน เพื่อศึกษารูปแบบของคำสั่งต่าง ๆ ที่ใช้จัดการฐานข้อมูลนี้คือ ไป

#### CREATE TABLE WORKER

```
(WK_ID          SMALLINT NOT NULL,
WK_NAME        CHAR(15),
HOUR_RATE      DECIMAL(5,2),
SKILL_TYPE     CHAR(7));
```

คำสั่งนี้เป็นคำสั่งให้มีการสร้างโครงสร้างตาราง 1 ตารางชื่อตาราง WORKER ที่ยังไม่มีข้อมูลใด

ๆ เก็บอยู่ โครงสร้างตารางนี้จะประกอบด้วยคอลัมน์ 5 คอลัมน์คือ รหัสคนงาน (WK\_ID) ชื่อคนงาน เอกสารนี้เป็นเอกสารที่ส่งงานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(WK\_NAME) อัตราค่าจ้างต่อชม. (HOUR\_RATE) และ ประเภทของความชำนาญ (SKILL\_TYPE) แต่ละคอลัมน์จะมีประเภทข้อมูลหลายแบบ ดังตัวอย่างต่อไปนี้

หมายเหตุ : มีการใช้คำว่า NOT NULL ใ้ค่อท้ายประเภทข้อมูลของคอลัมน์ WK\_ID จะหมายความว่าข้อมูลรหัสคนงานที่ถูกเก็บอยู่ในคอลัมน์นี้จะต้องไม่มีข้อมูลที่เป็นค่าว่าง (NOT NULL) แต่คอลัมน์อื่นที่ไม่ได้กำหนด NOT NULL จะมีข้อมูลเป็นค่าว่างได้

### ประเภทข้อมูล (Data Type)

ประเภทข้อมูลของคอลัมน์ในตารางจะขึ้นกับลักษณะของข้อมูลที่มีการใส่ลงในคอลัมน์แต่ละคอลัมน์ ซึ่ง DBMS แต่ละตัวจะมีประเภทของข้อมูลของคอลัมน์ที่แตกต่างกันได้ จะแสดงตัวอย่างประเภทของข้อมูลบางประเภทที่มีอยู่ใน Informix ดังนี้

ประเภทข้อมูล	ความหมาย
CHAR (n)	เป็นข้อมูลที่เป็นตัวอักขระใด ๆ โดย n คือจำนวนตัวอักขระที่มากที่สุด
DECIMAL (m,n)	เป็นประเภทข้อมูลที่เป็นตัวเลขจุดทศนิยม โดย m คือจำนวนตัวเลขทั้งหมด (รวมจุดทศนิยม) และ n คือจำนวนตัวเลขหลังจุดทศนิยม
INTEGER	เป็นประเภทข้อมูลที่เป็นเลขจำนวนเต็มบวกหรือลบขนาดเล็ก โดยต้องเป็นตัวเลขที่มีค่าตั้งแต่ -2,147,483,647 ถึง +2,147,483,647
SMALLINT	เป็นตัวเลขจำนวนเต็มบวกหรือลบขนาดเล็ก โดยต้องเป็นตัวเลขที่มีค่าตั้งแต่ -32,767 ถึง +32,767 ตัวเลขจำนวนเต็มประเภทนี้จะมีการจองเนื้อที่น้อยกว่าแบบ INTEGER
DATE	เป็นข้อมูลวันที่ซึ่งจะมีหลายรูปแบบให้เลือกใช้เช่น year/month/day หรือ month/year/day ฯลฯ

ตารางที่ 2.3 แสดงประเภทของข้อมูลบางประเภทที่มีอยู่ใน Informix ซึ่งเป็น DBMS สำหรับเครื่องระดับใหญ่

หมายเหตุ : คำสั่ง CREATE TABLE ใน DBMS บางตัวสามารถเพิ่มเติมคำสั่งควบคุมฐานข้อมูลให้เป็นการควบคุมความคงสภาพของข้อมูล 2 ข้อ โดยมีการกำหนดคีย์หลักและคีย์นอกไว้ได้

### คำสั่งการลบโครงสร้างตาราง

ใน Informix เมื่อต้องการลบ โครงสร้างตารางที่ถูกนิยามขึ้นนี้ จะสามารถทำได้ด้วยคำสั่ง DROP TABLE แล้วตามด้วยชื่อตารางที่ต้องการลบ รูปแบบทั่วไปมีดังนี้

```
DROP TABLE <ชื่อตารางที่ต้องการลบ>
```

หมายเหตุ : index ทุกตัวและวิวที่สร้างขึ้นสำหรับตารางนี้ ก็จะถูกลบไปพร้อม ๆ กันด้วย

### คำสั่งการเปลี่ยนแปลงโครงสร้างตาราง

ในบางครั้งเราอาจมีความจำเป็นที่จะต้องเปลี่ยนแปลง โครงสร้างตารางที่มีการนิยามไว้ข้างต้น เช่นอาจต้องการเพิ่มคอลัมน์บางคอลัมน์ลงไป ในตาราง หรือต้องการเปลี่ยนประเภทข้อมูลของคอลัมน์ จะสามารถทำได้ด้วยคำสั่ง และตามด้วยชื่อตารางที่ต้องการเปลี่ยน พร้อมทั้งระบุคอลัมน์ที่จะเพิ่มหรือเปลี่ยนรูปแบบทั่วไปมีดังนี้

```
ALTER TABLE <ชื่อตารางที่จะเปลี่ยนแปลง>  
<คำสั่งการเปลี่ยนแปลง> (<ชื่อคอลัมน์ ประเภทข้อมูล>);
```

#### ตัวอย่างที่ 2

```
ALTER TABLE WORKER  
ADD (SUPV_ID SMALLINT);
```

คำสั่งข้างต้นนี้เป็นคำสั่งให้ทำการเพิ่มคอลัมน์รหัสผู้ควบคุม (SUPV\_ID) ที่มีประเภทข้อมูลเป็น SMALLINT ลงไปในตาราง WORKER

#### ตัวอย่างที่ 3

```
ALTER TABLE WORKER  
MODIFY (SKILL_TYPE CHAR(8));
```

เป็นคำสั่งการเปลี่ยนแปลงคอลัมน์ชื่อ SKILL\_TYPE โดยจะทำการเปลี่ยนขนาดของคอลัมน์ให้เพิ่มขึ้น จากขนาด 7 ตัวอักษร เป็น 8 ตัวอักษร

หมายเหตุ : ถ้าการเปลี่ยนขนาดคอลัมน์ให้มีขนาดเล็กกว่าเดิม จะทำได้ก็ต่อเมื่อไม่มีข้อมูลใด ๆ เก็บอยู่ในตารางนั้น มิฉะนั้น DBMS จะไม่อนุญาตให้ทำการเปลี่ยนได้

### คำสั่งสร้างดัชนี

ดัชนี (Index) มีความสำคัญมากต่อฐานข้อมูลเชิงสัมพันธ์ เนื่องจาก DBMS ของฐานข้อมูลประเภทนี้ จะใช้ดัชนีในการค้นหระเบียนที่ต้องการได้อย่างรวดเร็ว โดยดัชนีที่ถูกสร้างขึ้นนี้ จะถูกเก็บเป็นตารางอยู่ในเนื้อที่อีกแห่งในดิสก์ที่แยกจากตารางข้อมูล ซึ่งถ้าตารางใดไม่มีการสร้างดัชนีไว้การค้นหข้อมูลในตารางนั้นจะต้องทำแบบเรียงลำดับจากระเบียนแรกจนถึงระเบียนสุดท้าย

ถ้าสมมติว่ามีการบันทึกข้อมูลลงไป ในตารางคนงาน ไว้แล้วด้วยคำสั่ง INSERT ที่จะได้กล่าวในหัวข้อการจัดการข้อมูลต่อไป ดังตารางที่ 2.4

แถวที่	รหัสคนงาน WK_ID	ชื่อคนงาน WK_NAME	อัตราค่าแรงต่อช. ม. HOUR_RATE	ประเภทความชำนาญ SKILL_TYPE	รหัสผู้ควบคุม SUPV_ID
1	1245	สุดใจ ดีพร้อม	16.00	ไฟฟ้า	1411
2	2521	พิเชฏ มากมี	16.00	ปูน	2522
3	1411	พิชัย ใจซื่อ	16.50	ไฟฟ้า	
4	2511	พิเชฏ มากมี	15.00	ประปา	

ก. ตารางข้อมูลคนงาน (WORKER)

ชื่อคนงาน WK_NAME	แถวที่
พิเชฏ มากมี	2
พิเชฏ มากมี	4
พิชัย ใจซื่อ	3
สุดใจ ดีพร้อม	1

ข. การใช้ชื่อคนงานเป็นดัชนี

รหัสคนงาน WK_OD	แถวที่
1245	1
1411	3
2511	4
2521	2

ค. การใช้รหัสคนงานเป็นดัชนี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ดูแลให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างดัชนีสำหรับตารางใด ๆ จะทำได้โดยการเลือกคอลัมน์ใดคอลัมน์หนึ่ง (หรือหลายคอลัมน์) จาก ตารางขึ้นมาเป็นดัชนี และตารางหนึ่ง ๆ ก็สามารถมีดัชนีได้หลายดัชนีตัวอย่างเช่นต้องการสร้างดัชนีจากตาราง WORKER โดยสมมติว่ามีการบันทึกข้อมูลไว้ดังรูป ตารางที่ 2.4 ก ถ้าลักษณะงานของผู้ใช้มีความต้องการค้นหาชื่อคนงานบ่อยจากตาราง WORKER ก็สามารถกำหนดให้คอลัมน์ชื่อคนงาน (WK\_NAME) เป็นดัชนีของตารางได้ระบบก็จะทำการสร้างตารางดัชนี (ตามตารางที่ 2.4 ข.) โดยใช้ชื่อคนงานเป็นคีย์ที่มีการเรียงลำดับชื่อคนงานจากน้อยไปมาก (ตามรหัส ASCII) และเก็บตำแหน่งที่อยู่ระเบียนของคนงานชื่อนั้นไว้ด้วย (ในทางปฏิบัติจะเป็นตำแหน่งทางกายภาพเช่นอยู่ที่แทรกที่เท่าใด ไซลินเดอร์ที่เท่าใด ฯลฯ ในดิสก์) ดังนั้นจากที่สร้างดัชนีไว้แล้ว ถ้าเราต้องการดึงข้อมูลของคนงานที่มีชื่อว่า พิชัย ใจชื่อ ระบบก็จะไปค้นในตารางดัชนี (ตารางที่ 2.4 ข.) เพื่อดูว่าระเบียนของ พิชัย ใจชื่อ อยู่ในแถวใด (ในทางปฏิบัติจะหมายถึงตำแหน่งที่อยู่จริง ๆ ของระเบียนนี้ในดิสก์) เมื่อทราบว่าอยู่ในแถวที่ 3 ก็จะไปดึงระเบียนของ พิชัย ใจชื่อ นั้นขึ้นมา ซึ่งการดึงระเบียนของเราต้องการค้นหารหัสคนงานบ่อย ก็สามารถกำหนดให้รหัสคนงาน (WK\_ID) เป็นดัชนีของตารางได้เช่นกัน ดังตารางที่ 2.4 ค.

ดังนั้น ในการค้นหาข้อมูลใด ๆ ระบบจะต้องมีการ ไปค้นที่ตารางดัชนีก่อน แล้วจึงจะไปดึงข้อมูลจริงจากตารางจริง ซึ่งย่อมต้องกินเวลาพอสมควร นอกจากนี้การเพิ่มหรือลบข้อมูลจากตารางจริง ก็ย่อมต้องมีการปรับปรุงตารางดัชนีตามไปด้วย ก็จะกินเวลาอีกเช่นกัน ดังนั้นการสร้างดัชนีจึงเหมาะกับตารางที่มีจำนวนระเบียนเก็บอยู่เป็นจำนวนมาก ๆ เนื่องจากจะเพิ่มประสิทธิภาพในการค้นหาและจัดการข้อมูล มากกว่าตารางที่มีจำนวนระเบียนเก็บอยู่น้อย

การสร้างดัชนีจะใช้คำสั่ง CREATE INDEX แล้วตามด้วยชื่อดัชนีที่เราตั้งขึ้น พร้อมทั้งระบุชื่อตารางและชื่อคอลัมน์ที่จะนำมาสร้างเป็นดัชนี รูปแบบทั่วไปมีดังนี้

```
CREATE [UNIQUE] INDEX <ชื่อดัชนีที่ตั้งขึ้น>
ON <ชื่อตารางที่จะสร้างดัชนี> (<ชื่อคอลัมน์-1>[,<ชื่อคอลัมน์-2>]...);
```

#### ตัวอย่าง4

```
CREATE UNIQUE INDEX CODE_IDX
ON WORKER (WK_ID);
```

ประโยชน์ของการสร้างดัชนีนอกจากจะเพิ่มความเร็วในการดึงข้อมูลแล้ว ยังสามารถนำมาใช้ในการควบคุมคอลัมน์ที่นำมาสร้างเป็นดัชนีให้มีการเก็บข้อมูลที่ไม่ซ้ำกัน Unique ได้อีกด้วย

ตัวอย่างที่ 4 เป็นคำสั่งให้มีการสร้างดัชนีเพื่อ CODE\_IDX จากตารางชื่อ WORKER โดยใช้คอลัมน์รหัสคนงาน (WK\_ID) ซึ่งมีการบังคับว่าคอลัมน์รหัสคนงานที่นำมาสร้างเป็นดัชนีนี้จะต้องไม่มีข้อมูลรหัสคนงานคนใดที่ซ้ำกันเกิดขึ้น

### ตัวอย่าง 5

```
CREATE INDEX NAME_IDX
ON WORKER (WK_NAME);
```

ในตัวอย่างที่ 5 นี้ไม่มีการใช้คำสั่ง UNIQUE ดังนั้นในคอลัมน์ชื่อคนงานที่นำมาสร้างเป็นดัชนีจึงสามารถมีข้อมูลชื่อคนงานที่ซ้ำกัน ได้ เช่นจากตัวอย่างคือ พิเชฐ มากมี เป็นต้น

ดัชนีที่ถูกสร้างด้วยคำสั่งประเภท SQL สำหรับ DBMS เช่น Informix นี้ เมื่อผู้ใช้ต้องการเรียกใช้งานตารางข้อมูลใด ๆ ที่มีการสร้างดัชนีไว้ ผู้ใช้สามารถเรียกใช้งานตารางนี้ได้โดยไม่ต้องระบุชื่อดัชนีที่ถูกสร้างขึ้นของตารางนี้ด้วย เนื่องจาก DBMS จะเป็นตัวจัดการเปิดดัชนีของตารางนั้นขึ้นมาให้อัตโนมัติ

### คำสั่งลบดัชนี

เมื่อต้องการลบดัชนีที่สร้างขึ้น ก็สามารถทำได้ด้วยคำสั่ง DROP INDEX แล้วตามด้วยชื่อดัชนีที่ต้องการลบ รูปแบบทั่วไปเป็นดังนี้

```
DROP INDEX <ชื่อดัชนีที่ต้องการลบ>
```

### 2.5.3 ภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language-DML)

ในเบื้องต้นถ้าเราออกคำสั่งประเภท DDL เช่น CREATE TABLE เราจะได้โครงสร้างตารางว่าง ๆ ที่ไม่มีข้อมูลใด ๆ เก็บอยู่ ถ้าตอนนี้เราต้องการที่จะบันทึกข้อมูลเข้าไปเก็บไว้ในตารางแต่ละตารางที่ได้สร้างไว้แล้วจะสามารถทำได้โดยคำสั่ง INSERT ซึ่งเป็นคำสั่งหนึ่งที่ถูกจัดว่าเป็นภาษาสำหรับการจัดการข้อมูล หรือเรียกย่อ ๆ ว่า DML นั่นเอง

SQL ที่เป็น DML ที่มีอยู่ใน DBMS สำหรับเครื่องขนาดใหญ่จะประกอบด้วย 4 คำสั่งดังตารางที่ 2.5

คำสั่ง	ความหมาย
SELECT	เรียกค้นข้อมูลในตาราง
INSERT	เพิ่มแถวข้อมูลลงไปในตาราง
UPDATE	ปรับปรุงแถวข้อมูลในตาราง
DELETE	ลบแถวข้อมูลในตาราง

ตารางที่ 2.5 แสดงกลุ่มคำสั่งที่เป็น DML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนที่จะกล่าวถึงคำสั่ง INSERT, UPDATE และ DELETE จะขอกล่าวถึงคำสั่ง SELECT ก่อน ซึ่งในที่นี้จะสมมติว่ามีการบันทึกข้อมูลลงไปในตารางว่างต่าง ๆ ที่สร้างขึ้นแล้ว และเราต้องการที่จะเรียกค้นข้อมูลเหล่านั้นขึ้นมาดู จะทำได้ด้วยการใช้คำสั่ง SELECT

### คำสั่งเรียกค้นข้อมูล

คำสั่ง SELECT ซึ่งเป็นคำสั่งการเรียกดูข้อมูล หรือ ค้นหาข้อมูล ตามเงื่อนไขที่เราระบุ เนื่องจากคำสั่ง SELECT เป็นคำสั่งที่มีรูปแบบการใช้งานที่ง่าย DBMS ส่วนใหญ่จึงมีการรวมคำสั่งนี้ไว้เป็นส่วนหนึ่งด้วย เพื่อช่วยในการเรียกค้นข้อมูล ซึ่งรวมทั้ง FoxPro สำหรับวินโดวส์นี้ด้วย โดยจะมีรูปแบบของคำสั่งเกือบทั้งหมดใกล้เคียงกับคำสั่ง SELECT ที่มีอยู่ใน DBMS สำหรับเครื่องใหญ่ ดังนั้นเราจะได้ศึกษารูปแบบของคำสั่ง SELECT ในหัวข้อนี้ก่อน

เนื่องจากคำสั่ง SELECT ที่จะได้กล่าวถึงนี้ บางคำสั่งจะมีการใช้เงื่อนไขทางตรรกศาสตร์ในคำสั่ง WHERE ดังนั้นก่อนที่จะได้กล่าวถึงรูปแบบของคำสั่ง SELECT จะขอกล่าวถึงเงื่อนไขทางตรรกศาสตร์เพิ่มเติมสักนิดเพื่อเป็นการปูพื้นความรู้เกี่ยวกับเรื่องดังกล่าว ซึ่งจะได้มีโอกาสนำไปใช้งานในการระบุเงื่อนไขในคำสั่ง SELECT ต่อไป

### ข้อมูลชนิดตรรกศาสตร์ (Boolean-Type Data)

เป็นข้อมูลที่แสดงค่าความจริงอย่างใดอย่างหนึ่ง จะมีค่าความจริงของข้อมูล 2 แบบ ได้แก่ ข้อมูลจริง (true) และ ข้อมูลเท็จ (false) โดยทั่วไปจะใช้ในการเปรียบเทียบนิพจน์ทางตรรกศาสตร์ (Boolean Expressions)

สามารถแบบนิพจน์ทางตรรกศาสตร์ออกได้เป็น 2 แบบ

- นิพจน์ทางตรรกศาสตร์ที่ใช้สัญลักษณ์แสดงความสัมพันธ์เป็นตัวเชื่อม

เป็นนิพจน์ที่ประกอบด้วยส่วนของ Operands ซึ่งเชื่อมกันด้วย Operator ซึ่งเป็นเครื่องหมายที่แสดงความสัมพันธ์ (Relation Operators) หรือการเปรียบเทียบ

ความหมาย	สัญลักษณ์	ตัวอย่าง
เท่ากับ	=	X = Y
มากกว่า	>	X > Y
น้อยกว่า	<	X < Y
มากกว่าหรือเท่ากับ	>=	X >= Y
น้อยกว่าหรือเท่ากับ	<=	X <= Y
ไม่เท่ากับ	<>	X <> Y

ตารางที่ 8.5 ตารางแสดงเครื่องหมายแสดงความสัมพันธ์และตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ : Operand ได้แก่ X และ Y จะใช้แทนตัวเลข ตัวอักษร ตัวแปร หรือ นิพจน์ ทางเลขคณิตใด

ตัวอย่าง

ถ้า Operand X และ Y เป็นตัวเลขใด ๆ ที่มีประเภทข้อมูลประเภทเดียวกัน จะแสดงค่าของนิพจน์ที่เชื่อมกันด้วยสัญลักษณ์แสดงความสัมพันธ์ได้ดังต่อไปนี้

นิพจน์ทางตรรกศาสตร์	ค่าของนิพจน์
$2 = 3$	False
$2 < 3$	True
$0.6 \geq 1.5$	False
$0.6 \geq -1.5$	True
$-4 < 4$	True
$1.7 \leq -2.2$	false

ตารางที่ 2.7 ตารางแสดงค่าตัวอย่างของนิพจน์

- นิพจน์ทางตรรกศาสตร์ที่ใช้สัญลักษณ์ทางตรรกศาสตร์เป็นตัวเชื่อม เป็นนิพจน์ที่ประกอบด้วยส่วนของ Operands ตรรกศาสตร์ และ Operator หรือ สัญลักษณ์ทางตรรกศาสตร์ (Logical Operator)

สัญลักษณ์	ความหมาย
AND	ใช้เชื่อม Operand ตั้งแต่ 2 Operand ขึ้นไป ซึ่งนิพจน์จะเป็นจริงก็ต่อเมื่อทุก Operand จะต้องเป็นจริง
OR	ใช้เชื่อม Operand ตั้งแต่ 2 Operand ขึ้นไป ซึ่งนิพจน์จะเป็นจริงก็ต่อเมื่อ Operand ใดตัวหนึ่งหรือทุกตัวเป็นจริง
NOT	ใช้นำหน้า Operand ถ้านำหน้า Operand ที่เป็นจริง จะทำให้นิพจน์เป็นเท็จ แต่ถ้านำหน้า Operand ที่เป็นเท็จ จะทำให้นิพจน์เป็นจริง

ตารางที่ 2.8 ตารางแสดงสัญลักษณ์ทางตรรกศาสตร์และความหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตัวอย่างการใช้ AND และ OR

ถ้ากำหนดให้ตัวแปร  $n$  เป็นตัวแปรชนิด integer และมีการกำหนดค่าข้อมูลเป็น 10 จะแสดงตัวอย่างการหาค่าของนิพจน์ดังต่อไปนี้

นิพจน์	ค่าของนิพจน์
$(n > 0) \text{ AND } (n < 20)$	True
$(n > 0) \text{ AND } (n < 5)$	False
$(n > 0) \text{ OR } (n < 5)$	True
$(n < 0) \text{ OR } (n > 20)$	False
$(n = 10) \text{ AND } (n < 5)$	True

### ตารางที่ 2.9 ตัวอย่างการใช้ AND และ OR

หมายเหตุ : เพื่อให้การเขียนนิพจน์ไม่กำกวม ควรเขียน Operand อยู่ภายในเครื่องหมายวงเล็บ

#### ตัวอย่าง การใช้ NOT

สมมติถ้า  $j$  เป็นตัวแปรชนิด integer ที่มีค่า 10 เก็บอยู่ และมีนิพจน์ทางตรรกศาสตร์คือ  $j > 6$  เป็นจริง (true) ดังนั้น

$\text{NOT } (j > 6)$  จะได้ผลลัพธ์ที่เป็นเท็จ (false) เพราะ  $j > 6$  เป็นจริง (true)

$\text{NOT } (j \leq 6)$  จะได้ผลลัพธ์เป็นจริง (true) เพราะ  $j \leq 6$  เป็นเท็จ (false)

#### • คำสั่งการเรียกดูข้อมูลอย่างง่าย (Simple Queries)

เป็นรูปแบบของคำสั่ง ที่สามารถเขียนได้ง่ายที่สุด โดยจะประกอบด้วยส่วนประกอบหลัก ๆ ของคำสั่ง 3 ส่วน ดังรูปแบบข้างล่างนี้ รูปแบบทั่วไปมีดังนี้

```
SELECT <ชื่อคอลัมน์ที่ต้องการดูข้อมูล>
FROM <ชื่อตาราง>
WHERE <เงื่อนไขตามทีระบุ>
```

**SELECT** –เป็นคำสั่งให้ทำการเรียกดูข้อมูลในคอลัมน์ที่ระบุอยู่หลังคำว่า SELECT ซึ่งอาจมีได้มากกว่า 1 คอลัมน์ขึ้นไป และถ้ามีมากกว่า 1 คอลัมน์ แต่ละคอลัมน์จะต้องคั่นด้วยเครื่องหมายคอมม่า (,) และนอกจากการระบุด้วยชื่อคอลัมน์แล้ว เรายังสามารถใช้เครื่องหมาย "\*" หลัง SELECT แสดงถึงความต้องการเรียกดูข้อมูลของทุกคอลัมน์ในตาราง

**FROM** –เป็นส่วนประกอบของคำสั่งที่บอกถึงตารางต่าง ๆ ซึ่งอาจมีมากกว่า 1 ตาราง ที่จะถูกเรียกใช้ข้อมูลจากคำสั่ง SELECT คอลัมน์ทุกคอลัมน์ที่มีอยู่ในคำสั่ง SELECT และ WHERE จะต้องเป็นคอลัมน์ที่มี

อยู่ในตารางใดตารางหนึ่งใน FROM นี้ใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**WHERE** –เป็นส่วนประกอบของคำสั่งที่บอกถึงเงื่อนไขที่จะใช้ในการค้นหาข้อมูลขึ้นมาจากตารางใด ๆ ที่อยู่หลัง **FROM** ซึ่งเงื่อนไขนี้อาจเป็นเงื่อนไขทางตรรกศาสตร์ที่มีค่าความจริงเป็นจริงหรือเท็จก็ได้ หรืออาจอยู่ในรูปแบบอื่น อีกตามที่จะได้กล่าวถึงต่อไป

### ตัวอย่างการเรียกดูข้อมูลอย่างง่าย

จากฐานข้อมูลบริษัทสุจริตการก่อสร้าง ถ้ามีการบันทึกข้อมูลในตารางต่าง ๆ ตามตารางที่ 2.10

#### คนงาน (WORKER)

รหัสคนงาน WK_ID	ชื่อคนงาน WK_NAME	อัตราค่าแรงต่อชม. HOOR_RATE	ประเภทความชำนาญ SKILL_TYPE	รหัสผู้ควบคุม SUPV_ID
1245	สุดใจ ศีพร้อม	16.00	ไฟฟ้า	1411
2521	พิเชฏ มวกมี	16.00	ปูน	2522
1411	พิชัย ใจซื่อ	16.50	ไฟฟ้า	
2511	พิเชฏ มวกมี	15.00	ประปา	
2522	ด้วง เกื้อกุน	16.50	ปูน	

ตารางที่ 2.10 ก. ตารางแสดงคนงาน

#### การทำงาน (ASSIGN)

รหัสคนงาน WK_ID	รหัสสถานที่ก่อสร้าง BLDG_ID	วันที่เริ่มทำงาน START_DATE	จำนวนชม.ทั้งหมดที่ทำ TOT_HOUR
1245	516	01/01/40	50
2521	516	01/01/40	56
1245	311	08/01/40	40
2521	450	15/01/40	45
2521	431	23/01/40	36
1411	311	01/02/40	20
2511	431	10/01/40	53

ตารางที่ 2.10 ข. ตารางแสดงการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### สถานที่ก่อสร้าง (BUILDING)

รหัสสถานที่ก่อสร้าง BLDG_ID	ที่อยู่สถานที่ก่อสร้าง BLDG_NAME	ประเภทของสถานที่ก่อสร้าง TYPE
516	128 ลาดพร้าว 78	บ้านพักอาศัย
311	25 สีลม	สำนักงาน
450	18 พหลโยธิน 55	ร้านค้า
431	50 อินทามระ 27	บ้านพักอาศัย
211	141 วิภาวดี	อะพาร์ตเมนต์

ตารางที่ 2.10 ค. ตารางแสดงสถานที่ก่อสร้าง

### ความชำนาญ(SKILL)

ประเภทความชำนาญ SKILL_TYPE	อัตราโบนัส BONUS_RATE	จำนวนช.ม. ขั้นต่ำ MIN_HOUR
ประปา	3.0	45
ไฟฟ้า	3.5	40
ปูน	2.5	50

ตารางที่ 2.10 ง. ตารางแสดงความชำนาญ

ตารางที่ 2.10 ตัวอย่างรีเลชันต่าง ๆ ที่มีในฐานะข้อมูลของบริษัทรับเหมาก่อสร้าง

จะแสดงตัวอย่างการเรียกดูข้อมูลด้วยคำสั่ง SELECT ในรูปแบบต่าง ๆ ดังต่อไปนี้  
ตัวอย่างที่ 6 ต้องการแสดงรายชื่อคนงานทุกคนที่มีอยู่

```
SELECT WK_NAME
FROM WORKER
```

ผลลัพธ์คือ

```
WK_NAME
สุคใจ คีพร้อม
พิเชฐ มากมี
พิชัย ใจซื่อ
พิเชฐ มากมี
ค้วง เกื้อกุน
```

**คำอธิบาย :** เป็นการเลือกแสดงเฉพาะรายชื่อคนงานที่อยู่ในตารางคนงาน โดยไม่มีเงื่อนไขใด ๆ  
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 7 ต้องการทราบว่า “มีคนงานคนใดบ้างที่ทำงานเป็นช่างไฟฟ้า?”

```
SELECT WK_NAME
FROM WORKER
WHERE SKILL_TYPE = “ไฟฟ้า”
```

ผลลัพธ์คือ

```
WK_NAME
สุดใจ ดีพร้อม
พิชัย ใจชื้อ
```

คำอธิบาย: เป็นคำสั่งการค้นหาข้อมูลจากตารางคนงาน โดยพิจารณาเงื่อนไขที่อยู่หลัง WHERE ซึ่งเป็นเงื่อนไขทางตรรกศาสตร์ที่มีเครื่องหมายแสดงความสัมพันธ์(=) เป็นตัวเชื่อม หมายความว่าแถวข้อมูลใดที่มีประเภทความชำนาญ (SKILL\_TYPE) เท่ากับ “ไฟฟ้า” จริง จะมีการนำข้อมูลชื่อคนงาน (WK\_NAME) ของแถวข้อมูลนั้นออกมาแสดง

ตัวอย่างที่ 8 ให้แสดงข้อมูลทั้งหมดเกี่ยวกับสถานที่ก่อสร้างที่เป็นบ้านพักอาศัย

```
SELECT * FROM BUILDING
WHERE TYPE = “บ้านพักอาศัย”
```

ผลลัพธ์คือ

BLDG_ID	BLDG_NAME	TYPE
516	128 ลาดพร้าว 78	บ้านพักอาศัย
431	50 อินทามระ 27	บ้านพักอาศัย

คำอธิบาย: เครื่องหมาย “\*” จะหมายถึงให้แสดงข้อมูลจากทุกคอลัมน์ในตารางสถานที่ก่อสร้างเฉพาะที่มีประเภทสถานที่ก่อสร้างเป็นบ้านพักอาศัยออกมาแสดง

ตัวอย่างที่ 9 “มีคนงานคนใดบ้างที่มีอัตราค่าแรงต่อ ชม. อยู่ระหว่าง 15.00 ถึง 16.00 บาท/ชม.”

```
SELECT * FROM WORKER
WHERE HOUR_RATE >= 15.00 AND HOUR_RATE <=16.00
```

ผลลัพธ์คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1245	สุดใจ คีพร้อม	16.00	ไฟฟ้า	1411
2521	พิเชฏ มากมี	16.00	ปูน	2522
2511	พิเชฏ มากมี	15.00	ประปา	

คำอธิบาย: เงื่อนไขที่อยู่หลัง WHERE ประกอบด้วยนิพจน์ทางตรรกศาสตร์ 2 นิพจน์ นิพจน์แรกคือ HOUR\_RATE >=15.00 นิพจน์ที่สอง คือ HOUR\_RATE <= 16.00 และแต่ละนิพจน์เชื่อมกันด้วยสัญลักษณ์ AND ซึ่งหมายความว่านิพจน์หลัง จะเป็นจริงก็ต่อเมื่อนิพจน์แรกมีค่าความจริงเป็นจริงคือ HOUR\_RATE >= 15.00 จริง และ นิพจน์สองก็ต้องมีค่าความจริงเป็นจริงด้วยคือ HOUR\_RATE <= 16.00 ด้วยเช่นกัน ดังนั้นแถวข้อมูลใดที่มีค่าข้อมูล HOUR\_RATE >= 15.00 และ HOUR\_RATE <=16.00 แถวข้อมูลนั้นก็จะถูกดึงออกมาแสดง

ในตัวอย่างที่ 9 จะสามารถเขียนได้อีกรูปแบบดังต่อไปนี้ ซึ่งจะ ได้ผลลัพธ์เหมือนกัน

```
SELECT * FROM WORKER
WHERE HOUR_RATE BETWEEN 15.00 AND 16.00
```

คำอธิบาย: สามารถใช้คำว่า BETWEEN แทนรูปแบบของเงื่อนไขในตัวอย่างที่ 8.9 ได้

ตัวอย่างที่ 10 ต้องการทราบข้อมูลของคณงานที่มีความชำนาญด้าน “ประปา” และ “ไฟฟ้า”

```
SELECT * FROM WORKER
WHERE SKILL_TYPE IN (“ประปา”, “ไฟฟ้า”)
```

ผลลัพธ์คือ

WK_ID	WK_NAME	HOUR_RATE	SKILL_TYPE	SUPV_ID
1245	สุดใจ คีพร้อม	16.00	ไฟฟ้า	1411
1411	พิชัย ใจซื่อ	16.50	ไฟฟ้า	
2511	พิเชฏ มากมี	15.00	ประปา	

คำอธิบาย: เงื่อนไขหลัง WHERE มีการใช้คำว่า IN จะมีความหมายเหมือนกับเขียนว่า

```
WHERE SKILL_TYPE = “ประปา” OR
SKILL_TYPE = “ไฟฟ้า”
```

ตัวอย่างที่ 11 ต้องการทราบรหัสคณงานและชื่อคณงานที่ไม่มีผู้ควบคุม

```
SELECT WK_ID, WK_NAME FROM WORKER
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และจะขึ้นชื่อเพื่อการค้าภายใต้ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ขอสงวนสิทธิ์ในนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## WHERE SUPV\_ID IS NULL

## ผลลัพธ์คือ

WK_ID	WK_NAME
1411	พิชัย ใจชื่อ
2511	พิเชฐ มากมี
2522	ค้วง เกื้อกุน

คำอธิบาย: -คำว่า IS NULL เป็นเงื่อนไขหนึ่งที่สามารถใช้ใน WHERE ได้ เป็นการตรวจสอบว่ามีแถวข้อมูลใดที่มีคอลัมน์รหัสผู้ควบคุมเป็นค่าว่าง (NULL) บ้าง ให้แสดงเฉพาะชื่อคนงานที่ระบุใน SELECT ออกมา

-: ถ้าใช้คำว่า IS NOT NULL จะหมายความว่าแถวข้อมูลใดมีรหัสผู้ควบคุมไม่เป็นค่าว่างให้แสดงชื่อคนงานในแถวข้อมูลนั้นออกมา

- การเรียกดูข้อมูลจากหลายตาราง (Multiple-Table Queries)

ในตัวอย่างก่อนได้แสดงตัวอย่างการเรียกดูข้อมูลอย่างง่ายจากตารางเพียง 1 ตาราง ถ้าเราต้องการที่จะเรียกดูข้อมูลจากตารางมากกว่า 1 ตารางก็สามารถทำได้เช่นกัน ในลักษณะของการเชื่อมโยงข้อมูลระหว่างตารางดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 12 ต้องการทราบว่า “คนงานที่ทำงานอยู่ที่สถานที่ก่อสร้างที่มีรหัส 516 มีความชำนาญทางด้านใดบ้าง”

จะเห็นว่าข้อมูลที่เราต้องการ จะมีการเชื่อมข้อมูลจากตาราง 2 ตาราง อันได้แก่ตารางคนงานและตารางการทำงาน กล่าวคือในตารางการทำงานต้องค้นดูก่อนว่ารหัสสถานที่ก่อสร้าง 516 มีคนงานที่มีรหัสใดทำงานอยู่หลังจากนั้นก็จะนำรหัสคนงานที่ได้ไปค้นดูในตารางคนงานว่า รหัสคนงานนี้มีประเภทความชำนาญเป็นอะไร ดังคำสั่งต่อไปนี้

```
SELECT SKILL_TYPE
FROM WORKER,ASSIGN
WHERE WORKER.WK_ID = ASSIGN,WK_ID
AND BLDG_ID = 516
```

คำอธิบาย: จากคำสั่ง SELECT ข้างต้นระบบจะมีขั้นตอนการประมวลผล 3 ขั้นตอนดังนี้

1) พิจารณาตารางที่อยู่หลัง FROM จะประกอบด้วยตาราง 2 ตารางคือ ตารางคนงานและตารางการทำงานระบบ จะสร้างตารางชั่วคราวขึ้นมาตารางหนึ่งทีเก็บคอลัมน์ทุกคอลัมน์จากตารางทั้งสองไว้

ด้วยกัน ดังนั้นตารางชั่วคราวนี้จะมีทั้งหมด 9 คอลัมน์ด้วยกัน (5 คอลัมน์จากตารางคนงานและ 4 คอลัมน์จากตารางการทำงาน) หลังจากนั้นจะนำข้อมูลจากตารางชั่วคราวนี้ไปเชื่อมข้อมูลกับตารางคนงานและตารางการทำงานอีกครั้งหนึ่งเพื่อที่จะได้ข้อมูลที่ถูกต้องและสมบูรณ์ที่สุด

คอลัมน์จากตารางการทำงาน) สำหรับจำนวนแถวข้อมูลทั้งหมดในตารางชั่วคราวจะมีทั้งหมด 35 แถว (จำนวนแถวข้อมูลในตารางคนงานมี 5 แถว จำนวนแถวข้อมูลในตารางการทำงานมี 7 แถว ดังนั้นผลลัพธ์ของการเชื่อมข้อมูลทั้งสองตารางที่เป็นไปได้จะเป็น  $5 \times 7 = 35$  แถวข้อมูล)

กล่าวคือ แถวที่ 1 ในตารางคนงาน จะเชื่อมกับแถวข้อมูลทั้ง 7 แถวในตารางการทำงาน แถวที่ 2 ในตารางคนงาน ก็จะเชื่อมกับแถวข้อมูลทั้ง 7 แถวในตารางการทำงาน เช่นเดียวกับแถวที่ 3, 4 และ 5 ในตารางคนงาน ต่างก็จะเชื่อมกับข้อมูลทั้ง 7 แถวในตารางการทำงาน ดังนั้นแถวข้อมูลทั้งหมดในตารางชั่วคราวจะมี 35 แถว

2) หลังจากทีระบบ ได้สร้างผลลัพธ์ของการเชื่อมข้อมูลจากทั้งสองตารางเก็บไว้ในตารางชั่วคราวที่ประกอบด้วย 9 คอลัมน์ 35 แถว แล้ว ระบบจะพิจารณาเงื่อนไขที่อยู่หลัง WHERE เป็นลำดับถัดไป จะเห็นว่าประกอบด้วย 2 เงื่อนไขคือ

1. WORKER,WK\_ID = ASSIGN.WK\_ID
2. BLDG\_ID = 516

พิจารณาเงื่อนไขแรก เนื่องจากว่าทั้งในตารางคนงานและตารางการทำงานชื่อคอลัมน์รหัสคนงาน (WK\_ID) จะเขียนเหมือนกัน และผลลัพธ์ในตารางชั่วคราวจะมีคอลัมน์รหัสคนงานเก็บอยู่ 2 คอลัมน์ที่เหมือนกัน ดังนั้นเพื่อให้เกิดความแตกต่างระหว่างชื่อคอลัมน์รหัสคนงานของตารางคนงานและชื่อคอลัมน์รหัสคนงานของตารางการทำงาน จึงมีการเติมชื่อตารางลงไปข้างหน้าชื่อคอลัมน์รหัส คนงานของแต่ละคอลัมน์ เพื่อให้ทราบว่ารหัสคนงานนี้เป็นคอลัมน์ที่อยู่ในตารางใด เช่น WORKER.WK\_ID จะหมายถึงชื่อคอลัมน์รหัสคนงานที่อยู่ในตารางคนงาน ASSIGN,WK\_ID จะหมายถึงชื่อคอลัมน์รหัสคนงานที่อยู่ในตารางการทำงาน เป็นต้น

ในเงื่อนไขแรกนี้ เป็นเงื่อนไขที่บอกว่าต้องการแถวข้อมูลที่มีรหัสคนงานจากตารางคนงานเท่ากับ รหัสคนงานจากตารางการทำงาน ดังนั้นในตารางชั่วคราวแถวข้อมูลแถวใดที่มีค่าข้อมูลจากสองคอลัมน์นี้ไม่เท่ากันก็ถูกตัดออกไป เหลืออยู่เฉพาะแถวข้อมูลที่มีค่าจากสองคอลัมน์ที่เท่ากันเท่านั้น ดังรูปที่ 2.23

WORKER.WK_ID	WK_NAME	HR	SKILL_TYPE	SUPV_ID
1245	สุดใจ ศีพร้อม	16.00	ไฟฟ้า	1411
2521	พิเชฏ มากมี	16.00	ปูน	2522
1245	สุดใจ ศีพร้อม	16.00	ไฟฟ้า	1411
2521	พิเชฏ มากมี	16.00	ปูน	2522
2521	พิเชฏ มากมี	16.00	ปูน	2522
1411	พิชัย ใจซื่อ	16.50	ไฟฟ้า	
2511	พิเชฏ มากมี	15.00	ประปา	
ASSIGN.WK_ID	BLDG_ID	START_DATE	TOT_HOUR	
1245	516	01/01/40	50	
2521	516	01/01/40	56	
1245	311	08/01/40	40	
2521	450	15/01/40	45	
2521	431	23/01/40	36	
1411	311	01/02/40	20	
2511	431	10/01/40	53	

รูปที่ 2.23 แสดงผลลัพธ์ในตารางชั่วคราวที่เกิดจากการเชื่อมข้อมูลจากตารางคนงานและตารางการทำงาน

พิจารณาเงื่อนไขที่สอง คือ BLDG\_ID = 516 จากเงื่อนไขนี้แถวข้อมูลในตารางชั่วคราวในรูป 2.23 จะถูกตัดออกอีก เหลือเฉพาะแถวข้อมูลที่มีคอลัมน์รหัสสถานที่ก่อสร้างเท่ากับ 516 เท่านั้นดังรูป 2.24

WORKER.WK_ID	WK_NAME	HR	SKILL_TYPE	SUPV_ID
1245	สุดใจ ศีพร้อม	16.00	ไฟฟ้า	1411
2521	พิเชฏ มากมี	16.00	ปูน	2522
ASSIGN.WK_ID	BLDG_ID	START_DATE	TOT_HOUR	
1245	516	01/01/40	50	
2521	516	01/01/40	56	

รูปที่ 2.24 แสดงผลลัพธ์ในตารางชั่วคราวที่เกิดจากเงื่อนไขรหัสสถานที่ก่อสร้าง = 516

- 3) พิจารณาชื่อคอลัมน์ที่อยู่หลัง SELECT จะพบว่า มีเฉพาะคอลัมน์ประเภทความชำนาญเท่านั้นที่จะถูกดึงข้อมูลนั้นขึ้นมาแสดง ดังผลลัพธ์ต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ผลลัพธ์คือ

SKILL\_TYPE

ไฟฟ้า

ปูน

ตัวอย่างที่ 13 ต้องการทราบรายชื่อคนงานทุกคนที่ได้รับมอบหมายให้ไปทำการก่อสร้างที่อาคารสำนักงาน

จะเห็นว่าจากคำถามนี้จะต้องทำการเชื่อมข้อมูลจากตาราง 3 ตาราง คือ ตารางประเภทสถานที่ก่อสร้าง เพื่อดึงรหัสสถานที่ก่อสร้างที่เป็น “สำนักงาน” ตารางทำงานเพื่อดึงรหัสคนงานที่ทำงานยังสถานที่ก่อสร้างห้สนั้นและตารางคนงานเพื่อดึงชื่อคนงานที่มีรหัสตามที่ดึง ได้จากตารางการทำงาน ดังประโยคคำสั่งนี้

```
SELECT WK_NAME
FROM WORKER,ASSIGN,BUILDING
WHERE WORKER.WK_ID = ASSIGN.WK_ID
AND ASSIGN.BLDG_ID = BUILDING.BLDG_ID
AND TYPE = “สำนักงาน”
```

## ผลลัพธ์คือ

WK\_NAME

สุทธิ ติพร้อม

พิชัย ใจชื่อ

หมายเหตุ : สังเกตว่าถ้าชื่อคอลัมน์เช่นรหัสคนงานหรือ รหัสสถานที่ก่อสร้าง มีปรากฏอยู่ในตารางมากกว่า 1 ตารางขึ้นไป เราจะต้องใส่ชื่อตารางกำกับไว้ข้างหน้าชื่อคอลัมน์นั้นด้วย เพื่อเป็นการบอกว่าคอลัมน์นั้นเป็นคอลัมน์ที่อยู่ในตารางใด และเป็นการกำจัดความกำกวมที่อาจเกิดขึ้น แต่ถ้าคอลัมน์มีอยู่ในตารางเพียงตารางเดียวเช่นประเภทสถานที่ก่อสร้าง (TYPE) ก็ไม่จำเป็นต้องใส่ชื่อตารางกำกับอยู่หน้าชื่อคอลัมน์ดังกล่าว เนื่องจากไม่มีความกำกวมเกิดขึ้น

### การเรียกดูแบบซ้อนกัน (Subqueries)

การเรียกดูข้อมูลแบบซ้อนกัน จะหมายถึงการนำประโยคการเรียกดูข้อมูลใส่อยู่ภายใต้ประโยคการเรียกดูข้อมูลอีกประโยค มีรูปแบบทั่วไปดังนี้

```
SELECT <ชื่อคอลัมน์>
FROM <ชื่อตาราง>
WHERE <ชื่อคอลัมน์> IN
    (SELECT <ชื่อคอลัมน์>
     FROM <ชื่อตาราง>
     WHERE <ชื่อคอลัมน์>)
```

ตัวอย่างที่ 14 จากตัวอย่างที่ 12 คือต้องการทราบประเภทความชำนาญของคนงานที่ได้รับมอบหมายให้ไปทำงานก่อสร้างยังสถานที่รหัส 516 จงเขียนคำสั่งในรูปแบบการเรียกดูข้อมูลแบบซ้อนกัน

```
SELECT SKILL_TYPE
FROM WORKER
WHERE WK_ID IN
    (SELECT WK_ID
     FROM ASSIGN
     WHERE BLDG_ID = 516)
```

คำอธิบาย : วิธีการทำงานของประโยคคำสั่งการเรียกดูแบบซ้อนกันนี้ คำสั่งการเรียกดูข้อมูลคำสั่งในสุด (คำสั่ง SELECT ที่อยู่ในวงเล็บ) จะมีการทำงานก่อน ผลลัพธ์ของคำสั่งการเรียกดูข้อมูลด้านในสุดนี้จะส่งต่อไปให้กับคำสั่งการเรียกดูข้อมูลส่วนนอก ดังนั้นจากประโยคคำสั่งข้างต้นคำสั่งแรกที่จะทำงานคือ

```
SELECT WK_ID
FROM ASSIGN
WHERE BLDG_ID = 516
```

ผลลัพธ์คือ

```
WK_ID
1245
2521
```

ผลลัพธ์ที่เกิดขึ้นนี้ จะถูกส่งให้กับคำสั่ง SELECT ที่อยู่ด้านนอกสุด ซึ่งจะมีค่าเท่ากับคำสั่งว่า

```

SELECT SKILL_TYPE
FROM WORKER
WHERE WK_ID IN (1245,2521)

```

หรือมีความหมายเหมือนกับคำสั่ง

```

SELECT SKILL_TYPE
FROM WORKER
WHERE WK_ID = 1245 OR WK_ID =2521

```

ผลลัพธ์คือ

SKILL\_TYPE

ไฟฟ้า

ปูน

ตัวอย่างที่ 15 จากตัวอย่างที่ 13 คือให้แสดงรายชื่อคนงานที่ได้รับมอบหมายให้ไปก่อสร้าง ณ อาคาร  
สำนักงาน โดยเขียนคำสั่งในรูปแบบของการเรียกข้อมูลดูแบบซ้อน

เนื่องจากว่าจะต้องมีการดึงข้อมูลมาจาก 3 ตาราง ดังนั้นจึงต้องใช้คำสั่งการเรียกดูข้อมูล 3 ชั้นดังนี้

นี้

```

SELECT WK_NAME
FROM WORKER
WHERE WK_ID IN
    (SELECT WK_ID
     FROM ASSIGN
     WHERE BLDG_ID IN
        (SELECT BLDG_ID
         FROM BUILDING
         WHERE TYPE = "สำนักงาน"))

```

ผลลัพธ์คือ

WK\_NAME

สุคใจ ศีพร้อม

พิชัย ใจชื่อ

คำอธิบาย : -จะมีการค้นหาหารหัสสถานที่ก่อสร้างจากตารางสถานที่ก่อสร้างที่มีประเภทเป็นสำนักงานก่อน  
ซึ่งผลลัพธ์ที่ได้คือรหัสที่ได้ ก็จะส่งไปยังคำสั่ง SELECT ในลำดับถัดไปข้างบนนี้คือการค้นหาหารหัสคนงานการค่า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานจากตารางการทำงานที่มีการทำงานยังสถานที่ก่อสร้างรหัส 311 ซึ่งจะได้รหัสคนงาน 2 คน คือ 1245 และ 1411 ผลลัพธ์นี้จะส่งต่อไปยังคำสั่ง SELECT บนสุด ได้แก่การค้นหาชื่อคนงานจากตารางคนงานที่มีรหัสเป็น 1245 และ 1411 ก็จะได้ผลลัพธ์ดังข้างต้น

- สังเกตว่าเราไม่มีการใช้ชื่อตารางไว้หน้าชื่อคอลัมน์เช่นรหัสคนงานหรือรหัสสถานที่ก่อสร้าง ทั้งนี้เพราะคำสั่งการเรียกดูย่อยแต่ละคำสั่งจะมีตารางที่ต้องดึงข้อมูลเพียงตารางเดียว ดังนั้นจึงทราบว่า รหัสคนงานหรือรหัสสถานที่ก่อสร้างเป็นคอลัมน์ที่ดึงมาจากตารางใด จึงไม่เกิดความกำกวมกับชื่อคอลัมน์ขึ้น

ตัวอย่างที่ 16 มีคนงานคนใดบ้างที่ยังไม่ได้ถูกกำหนดให้ไปทำงานยังสถานที่ก่อสร้างแห่งใดเลย

```
SELECT * FROM WORKER
WHERE WK_ID NOT IN
(SELECT WK_ID FROM ASSIGN)
```

ผลลัพธ์คือ

WK_ID	WK_NAME	HRUR_RATE	SKILL_TYPE	SUPV_ID
2522	ด้วง เกื้อกูน	16.50		ปูน

ตัวอย่างที่ 17 ต้องการทราบว่า มีสถานที่ก่อสร้างแห่งใดที่มีคนงานที่มีรหัส 2521 และ 2511 ทำงานอยู่ด้วยกัน

```
SELECT ASSIGN.BLDG_ID,BLDG_NAME,TYPE
FROM ASSIGN,BUILDING
WHERE ASSIGN.BLDG_ID = BUILDING.BLDG_ID
AND WK_ID = 2521
AND ASSIGN.BLDG_ID IN
(SELECT BLDG_ID FROM ASSIGN WHERE WK_ID = 2511)
```

ผลลัพธ์คือ

BLDG_ID	BLDG_NAME	TYPE
431	50 อินทามระ 27	บ้านพักอาศัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบาย - จะมีการค้นหารหัสสถานที่ก่อสร้างของคอนกรีต 2511 จากตารางการทำงานก่อน ซึ่งจะได้รหัสสถานที่ก่อสร้างคือรหัส 431 หลังจากนั้นจึงจะไปค้นหารหัสสถานที่ก่อสร้าง ชื่อ และ ประเภทสถานที่ก่อสร้างของคอนกรีต 2521 ที่ทำงานอยู่ในสถานที่ก่อสร้างรหัส 431 เช่นกัน

### การใช้ฟังก์ชัน (Built-In Functions)

SQL ได้จัดหาฟังก์ชันทางสถิติเพื่อช่วยในการคำนวณกลุ่มแถวข้อมูลของคอลัมน์ที่ระบุในตาราง ซึ่งจะประกอบด้วยฟังก์ชันต่าง ๆ ดังต่อไปนี้

ฟังก์ชัน	ความหมาย
COUNT	นับจำนวนแถวข้อมูลของคอลัมน์
SUM	หาผลรวมข้อมูลที่เป็นตัวเลขของคอลัมน์
AVG	หาค่าเฉลี่ยของข้อมูลที่เป็นตัวเลขของคอลัมน์
MAX	หาค่าข้อมูลตัวเลขที่มากที่สุดของคอลัมน์
MIN	หาค่าข้อมูลตัวเลขที่น้อยที่สุดของคอลัมน์

ตารางที่ 2.11 แสดงฟังก์ชันทางสถิติที่มีใช้ใน SQL

ตัวอย่างที่ 18 ต้องการทราบอัตราค่าแรงต่อ ช.ม. ที่มากที่สุดและน้อยที่สุดเป็นเท่าไร

```
SELECT MAX (HOUR_RATE), MIN(HOUR_RATE)
FROM WORKER
```

ผลลัพธ์คือ

16.50 15.00

คำอธิบาย :

MAX และ MIN เป็นฟังก์ชันที่ใช้หาค่ามากที่สุดและน้อยที่สุดของข้อมูลในคอลัมน์หนึ่ง ๆ ในตาราง

ตัวอย่างที่ 19 จงหาค่าเฉลี่ยของจำนวนช.ม. ทั้งหมดที่ทำ ของคอนกรีตที่ได้รับมอบหมายให้ไปทำงานยังสถานที่ก่อสร้างรหัส 516

```
SELECT AVG(TOT_HOUR)
FROM ASSIGN
WHERE BLDG_ID = 516
```

## ผลลัพธ์คือ

53

คำอธิบาย: จากตัวอย่างข้างต้นจะมีการคำนวณค่าเฉลี่ยของจำนวน ชม. ทั้งหมดที่ทำ เฉพาะแถวในตารางการทำงานที่มีรหัสสถานที่ก่อสร้างเป็น 516 เท่านั้น ซึ่งจะมีข้อมูลอยู่ 2 แถว คือ 50 และ 60 ตัวอย่างที่ 8.20 ให้หาผลรวมของจำนวน ชม. ทั้งหมดที่ทำ ของคนงานที่มีความชำนาญด้านปูน

```
SELECT SUM(TOT_HOUR)
FROM WORKER,ASSIGN
WHERE WORKER.WK_ID = ASSIGN.WK_ID
AND SKILL_TYPE = "ปูน"
```

## ผลลัพธ์คือ

137.00

คำอธิบาย: เนื่องจากมีการกล่าวถึงคนงานที่มีความชำนาญด้านปูน ซึ่งได้แก่คนงานรหัส 2521 และ 2522 จึงต้องข่งเกี่ยวกับตารางคนงาน และเนื่องจากต้องการหาผลรวมจำนวน ชม. ทั้งหมดที่คนงานที่มีความชำนาญด้านนี้ทำ ซึ่งจำนวน ชม. ทั้งหมดนี้เป็นคอลัมน์ที่มีอยู่ในตารางการทำงาน ดังนั้นจึงต้องมีการเชื่อมข้อมูลจากทั้งสองตาราง ส่วนฟังก์ชัน SUM จะใช้หาผลรวมดังกล่าว (ในที่นี้มีเฉพาะคนงานรหัส 2521 เท่านั้นที่ถูกกำหนดให้ทำงาน ดังนั้นจึงเป็นผลรวมจำนวน ชม. ของคนงานรหัส 2521)

ตัวอย่างที่ 21 จงหาผลรวมค่าแรงของคนงานรหัส 2521 จากผลคูณระหว่างจำนวน ชม. ทั้งหมดที่ทำกับอัตราค่าแรงปกติ พร้อมทั้งให้แสดงผลรวมจำนวน ชม. ทั้งหมดของคนงานดังกล่าวด้วย

```
SELECT SUM (TOT_HOUR),SUM(HOUR_RATE*TOT_HOUR)
FROM WORKER, ASSIGN
WHERE ASSIGN.WK_ID = WORKER.WK_ID
AND ASSIGN.WK_ID = 2521
```

## ผลลัพธ์คือ

137.0 2192.00

คำอธิบาย - คำสั่ง SUM สามารถใช้หาผลรวมของเขตข้อมูลที่อยู่ในรูปของนิพจน์ ในที่นี้คือ HOUR\_RATE \* TOT\_HOUR โดย HOUR\_RATE เป็นเขตข้อมูลที่อยู่ในตารางคนงาน

ตัวอย่างที่ 22 ต้องการทราบว่า มีประเภทความชำนาญที่แตกต่างกันกี่ด้าน

```
SELECT COUNT(DISTINCT SKILL_TYPE)
FROM WORKER
```

ผลลัพธ์คือ

3

คำอธิบาย เนื่องจากแต่ละแถวจะมีประเภทความชำนาญที่ซ้ำกันได้ จึงต้องใช้คำว่า "DISTINCT" เพื่อบอกระบบว่าถ้าพบแถวข้อมูลใดที่ซ้ำกันจะไม่สนใจแถวข้อมูลที่ซ้ำนั้น และเมื่อมีการใช้ "DISTINCT" ร่วมกับ "COUNT" จะหมายความว่าถ้าประเภทความชำนาญใดมีมากกว่า 1 แถว ให้นับเพียง 1 เท่านั้น นอกจากนี้ "DISTINCT" ยังสามารถใช้ร่วมกับฟังก์ชันอื่น ได้อีกเช่นกัน

ตัวอย่างที่ 23 ต้องการทราบว่า มีสถานที่ก่อสร้างกี่แห่งที่เป็นบ้านพักอาศัย

```
SELECT COUNT (*)
FROM BUILDING
WHERE TYPE = "บ้านพักอาศัย"
```

ผลลัพธ์คือ

2

คำอธิบาย : COUNT (\*) จะหมายถึงจำนวนแถวทั้งหมดที่นับได้ แม้ข้อมูลในคอลัมน์นั้นจะเป็นค่าว่างก็ตาม

### GROUP BY , ORDER BY และ HAVING

ในระดับบริหารแล้วส่วนใหญ่จะสนใจข้อมูลทางสถิติที่มีการแบ่งออกเป็นกลุ่ม ๆ ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 24 ต้องการทราบว่าคนงานแต่ละคนมีจำนวนชม.ทั้งหมดที่ทำงานมากที่สุดคนละกี่ชม.

```
SELECT WK_ID,MAX(TOT_HOUR)
FROM ASSIGN
GROUP BY WK_ID
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ผลลัพธ์คือ

WK_ID	MAX(TOT_HOUR)
1245	50
2521	56
1411	20
2511	53
2512	

คำอธิบาย : **GROUP BY** <ชื่อคอลัมน์ > จะเป็นคำสั่งให้มีการจัดกลุ่มแถวข้อมูลตามคอลัมน์ที่ระบุหลัง **GROUP BY** โดยข้อมูลที่เหมือนกันจะถูกจัดให้อยู่ในกลุ่มเดียวกัน ดังนั้นจากตัวอย่างข้างต้น แถวข้อมูลที่มีรหัสคนงานเหมือนกันจะถูกจัดกลุ่มให้อยู่กลุ่มเดียวกัน และจากคำสั่ง **SELECT** ให้ทำการดึงข้อมูลรหัสคนงาน และจำนวนชม. ทั้งหมดที่ทำมาสุดของคณงานแต่ละคน โดยให้แสดงผลลัพธ์แยกตามกลุ่มของคณงานแต่ละกลุ่ม

ตัวอย่างที่ 25 จากตัวอย่างที่ 24 ถ้าให้แสดงผลลัพธ์โดยเรียงลำดับตามรหัสคนงานจากมากไปน้อย

```
SELECT WK_ID,MAX(TOT_HOUR)
FROM ASSIGN
GROUP BY WK_ID
ORDER BY WK_ID DESC
```

## ผลลัพธ์คือ

WK_ID	MAX(TOT_HOUR)
2521	56
2511	53
1411	20
1245	50
1246	

คำอธิบาย : คำสั่ง **ORDER BY** <ชื่อคอลัมน์> [*DESC*] ใช้ในกรณีที่ต้องการเรียงลำดับข้อมูลในคอลัมน์ที่ระบุ ถ้าไม่มีการใช้คำว่า *DESC* ใน **ORDER BY** จะทำการเรียงลำดับข้อมูลจากน้อยไปมาก แต่ถ้ามีการใช้คำว่า *DESC* จะทำการเรียงลำดับจากมากไปน้อยแทน

ตัวอย่างที่ 26 ถ้าต้องการเรียงลำดับตาม จากมากไปน้อยจะทำได้ดังนี้

```
SELECT WK_ID,MAX(TOT_HOUR)
FROM ASSIGN
GROUP BY WK_ID
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ORDER BY 2 DESC

WK_ID	ผลลัพธ์คือ MAX(TOT_HOUR)
2521	56
2511	53
1245	50
1411	20

คำอธิบาย: ตัวเลข 2 ที่อยู่หลัง ORDER BY จะเป็นการบอกระบบให้ทำการเรียงลำดับข้อมูลตามคอลัมน์ที่ 2 ของคำสั่ง SELECT ซึ่งก็คือ MAX(TOT\_HOUR) นั่นเอง

ตัวอย่างที่ 27 ให้แสดงข้อมูลคนงานที่ได้รับมอบหมายให้ทำงาน โดยเรียงลำดับตามรหัสคนงานและจำนวน ชม. ทั้งหมดที่ทำ จากน้อยไปมาก

```
SELECT *
FROM ASSIGN
ORDER BY WK_ID,TOT_HOUR
```

ผลลัพธ์คือ

WK_ID	BLDG_ID	START_DATE	TOT_HOUR
1245	311	08/01/40	40
1245	516	01/01/40	50
1411	311	01/02/40	20
2511	431	10/01/40	53
2521	431	23/01/40	36
2521	450	10/01/40	45
2521	516	01/01/40	56

คำอธิบาย: คำสั่ง ORDER BY สามารถใช้ในการเรียงลำดับคอลัมน์มากกว่าหนึ่งคอลัมน์ได้ด้วย โดยระบบจะทำการเรียงตามคอลัมน์ที่หนึ่งคือรหัสคนงานก่อน และภายในคอลัมน์รหัสคนงานถ้ามีรหัสคนงานซ้ำกัน ก็จะทำการเรียงลำดับข้อมูลตามคอลัมน์ที่สองคือจำนวน ชม. ทั้งหมดที่ทำอีกที

ตัวอย่างที่ 28 ต้องการให้แสดงรหัสสถานที่ก่อสร้างที่มีการสร้างโดยคนงานมากกว่า 1 คน

SELECT BLDG\_ID FROM ASSIGN WHERE BLDG\_ID IN (SELECT BLDG\_ID FROM ASSIGN GROUP BY BLDG\_ID HAVING COUNT(\*) > 1)

เอกสารนี้เป็นเอกสารตัวอย่างเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FROM ASSIGN  
 GROUP BY BLDG\_ID  
 HAVING COUNT (\*) > 1

### ผลลัพธ์คือ

BLDG\_ID

516

311

431

คำอธิบาย: คำสั่ง HAVING จะเหมือนกับคำสั่ง WHERE ที่ต้องตามด้วยเงื่อนไข แต่ HAVING จะใช้ในกรณีที่มีการจัดกลุ่มหรือการใช้กับ GROUP BY เท่านั้น ข้อแตกต่างระหว่าง HAVING กับ WHERE คือ HAVING จะใช้ในการกำจัดกลุ่มที่ไม่ต้องการออก แต่ WHERE จะใช้ในการกำจัดแถวที่ไม่ต้องการออก

### คำสั่งเพิ่มแถวข้อมูล

คำสั่ง SQL ที่ใช้ในการเพิ่มแถวข้อมูลลงในตาราง จะใช้คำสั่ง INSERT ซึ่งจะมีอยู่ 2 รูปแบบคือแบบแรกจะเป็นการเพิ่มข้อมูลที่ละแถว โดยระบุค่าข้อมูลของแต่ละคอลัมน์ลงในคำสั่ง INSERT โดยตรง แบบที่สองเป็นการใช้คำสั่งค้นหาข้อมูล (SELECT) เพื่อดึงกลุ่มข้อมูลซึ่งอาจมีหลายแถว ส่งให้กับคำสั่ง INSERT ทำการเพิ่มข้อมูลลงไปในตาราง

รูปแบบทั่วไป แบบที่ 1 : การเพิ่มข้อมูลที่ละแถวลงไปในตาราง

```
INSERT INTO <ชื่อตารางที่จะเพิ่มข้อมูล>
VALUES (<ค่าข้อมูลของแต่ละคอลัมน์>)
```

ตัวอย่างที่ 29

```
INSERT INTO WORKER
VALUES(1245,"สุดใจ ดีพร้อม",16.00,"ไฟฟ้า", 1411)
```

คำอธิบาย - คำสั่งข้างต้นเป็นคำสั่งการเพิ่มข้อมูล 1 แถวลงไปในตารางคนงาน(WORKER) ข้อมูลคอลัมน์แต่ละคอลัมน์ที่อยู่ในวงเล็บนั้นจะต้องมีประเภทข้อมูลตรงกับประเภทข้อมูลของคอลัมน์ในตารางที่สร้างไว้ เช่น คอลัมน์รหัสคนงาน มีประเภทข้อมูลเป็น SMALLINT แต่ถ้าเราใส่ข้อมูลเป็น "1245" หรือข้อความนี้เป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความใด ๆ ที่อยู่ภายในเครื่องหมาย "" จะทำให้ระบบเข้าใจว่าเป็นกลุ่มตัวอักษรหรือสตริง (String) ซึ่ง จะไม่สามารถเพิ่มข้อมูลลงไปในตารางได้ และนอกจากนี้ถ้าจำนวนข้อมูลที่ใส่ในวงเล็บไม่ตรงกับจำนวน คอลัมน์ที่มีในตาราง ก็จะทำให้ไม่สามารถเพิ่มข้อมูลลงไปได้เช่นกัน

### รูปแบบทั่วไป แบบที่ 2

```
INSERT INTO <ชื่อตารางที่จะเพิ่มข้อมูล>
<ประโยคคำสั่ง SELECT>
```

ตัวอย่างที่ 30 สมมติว่ามีการสร้างตารางชื่อ WORKER\_2 อยู่ ที่ประกอบด้วยคอลัมน์ต่าง ๆ เหมือนกับตาราง และสมมติต่ออีกว่าในตาราง WORKER เราได้ทำการเพิ่มข้อมูลลงไปแล้วเป็นจำนวนหลายแถว ต่อไปเราต้องการนำข้อมูลทั้งหมดที่เก็บอยู่ในตาราง WORKER เพิ่มเข้าไปในตาราง WORKER\_2 เราจะทำ ได้ด้วยคำสั่งต่อไปนี้

```
INSERT INTO WORKER_2
SELECT * FROM WORKER
```

หมายเหตุ : นอกจากคำสั่ง INSERT แล้ว DBMS บางตัวเช่น Informix ได้จัดหาเครื่องมือที่ช่วยในการเพิ่ม ข้อมูลลงในตารางอีกในกรณีที่มีการเก็บแถวข้อมูลซึ่งมีเป็นจำนวนมาก ๆ ไว้ในรูปของแฟ้มอยู่แล้ว ตัวอย่างเช่น ถ้ามีการเก็บข้อมูลคนงานไว้ในแฟ้มชื่อ "WORKER.UNL" ซึ่งมีตัวอย่างข้อมูลดังนี้

แฟ้ม "WORKER.UNL" ตัวต้นคอลัมน์คือ"1"

1245Iสุคนใจ ศีพร้อมI16.00Iไฟฟ้าI141I

252Iพิเชฏ มากมีI16.00IปูนI2522I

141Iพิชัย ใจชื่อI16.50Iไฟฟ้าI

251Iพิเชฏ มากมีI15.00IประปาI

๑๑๑

สามารถใช้โปรแกรมอำนวยความสะดวกชื่อ "LOAD" นำข้อมูลจากแฟ้ม "worker.unl" นี้ เพิ่มลงไปใน ตาราง WORKER ดังตัวอย่างต่อไปนี้

```
LOAD FROM "WORKER.UNL"
INSERT INTO WORKER
```

แต่ถ้าต้องการนำข้อมูลจากตาราง WORKER ออกมาเก็บไว้ในรูปของแฟ้มที่มีรูปแบบเหมือนกับแฟ้ม “WORKER.UNL” ข้างต้น ก็สามารถทำได้ด้วยคำสั่ง “UNLOAD” ดังตัวอย่างต่อไปนี้

```
UNLOAD INTO "WORKER.UNL"
SELECT * FROM WORKER
```

### คำสั่งปรับปรุงแถวข้อมูล

การปรับปรุงแถวข้อมูลจะทำได้ด้วยคำสั่ง UPDATE เป็นการปรับปรุงหรือแก้ไขค่าคอลัมน์ซึ่งอาจมีมากกว่า 1 คอลัมน์ ในแถวทุกแถวที่มีเงื่อนไขสอดคล้องกับที่ระบุไว้หลังคำว่า WHERE ในคำสั่ง UPDATE รูปแบบทั่วไปมีดังนี้

```
UPDATE <ชื่อตารางที่ต้องการปรับปรุง>
SET <ชื่อคอลัมน์> =<ค่าข้อมูล>
WHERE <เงื่อนไขตามที่ระบุ>
```

ตัวอย่างที่ 31 ถ้าต้องการปรับปรุงคอลัมน์อัตราค่าแรงต่อช.ม.(HOUR\_RATE) ในตารางคนงานของคนที่ มีรหัส 2511 จากค่าแรงเดิมคือ 15.00 เปลี่ยนเป็น 15.50 บาท/ช.ม. ทำได้ด้วยคำสั่งดังนี้

```
UPDATE WORKER
SET HOUR_RATE = 15.50
WHERE WK_ID = 2511
```

คำอธิบาย -จากคำสั่งข้างต้นจะมีแถวข้อมูลคนงานเพียงแถวเดียวเท่านั้นที่ถูกปรับปรุง เนื่องจากคนงานที่มีรหัส 2511 จะมีเพียงคนเดียวเท่านั้น

หมายเหตุ : ถ้าไม่มีการใช้ประโยคเงื่อนไข WHERE ในคำสั่ง UPDATE จะทำให้แถวข้อมูลคนงานจะถูกปรับปรุงให้มีอัตราค่าแรงต่อช.ม.เป็น 15.50 หมดทุกแถว

### คำสั่งลบแถวข้อมูล

คำสั่งที่ใช้ในการลบแถวข้อมูล จะได้แก่คำสั่ง DELETE ซึ่งจะเป็คำสั่งให้มีการลบแถวข้อมูล

ทุกแถวที่มีเงื่อนไขสอดคล้องกับที่ระบุไว้หลังคำว่า WHERE ในคำสั่ง DELETE เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับนักเรียนเพื่อการศึกษาเท่านั้น เมื่อนุญาดให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบทั่วไปมีดังนี้

```
DELETE <ชื่อตารางที่จะลบ>
WHERE <เงื่อนไขตามที่ระบุ>
```

ตัวอย่างที่ 32 ถ้าคนงานที่มีรหัส 1411 ได้ลาออกไป และเราต้องลบแถวข้อมูลคนงานที่มีรหัส 1411 ออกไป จากตารางคนงาน จะทำได้โดยออกคำสั่ง

```
DELETE WORKER
WHERE WK_ID = 1411
```

หมายเหตุ : ถ้าไม่มีประโยคเงื่อนไข WHERE จะทำให้มีการลบแถวข้อมูลทุกแถวออกไปจากตารางคนงาน

#### 2.5.4 วิว (View)

ในหัวข้อ 2.5.4 เราได้ตั้งเรื่องการสร้างวิวสำหรับภาษานิยามข้อมูลไว้ เนื่องจากคำสั่งการสร้างวิวจะยุ่งเกี่ยวกับคำสั่งที่ใช้เรียกดูข้อมูล ดังนั้นจึงต้องอธิบายรูปแบบคำสั่งการเรียกดูข้อมูลให้เข้าใจก่อน จึงจะสามารถสร้างวิวด้วยตนเองได้ ซึ่งจะได้อธิบายไว้ในหัวข้อนี้

การป้องกันข้อมูลด้วยการสร้างวิวไปบ้างแล้ว ซึ่งวิว (VIEW) ก็คือตารางสมมติ (Virtual table) ที่ถูกสร้างขึ้นมาโดยการดึงข้อมูลบางส่วนมาจากตารางจริง (base table) ตั้งแต่ 1 ตารางขึ้นไปในฐานข้อมูล ประโยชน์เพื่อเป็นการป้องกันข้อมูล ให้ผู้ใช้สามารถใช้ข้อมูลเฉพาะในส่วนที่ตนเกี่ยวข้องเท่านั้น ตามวิวที่ถูกสร้างขึ้นของแต่ละคน

วิวหรือตารางสมมติจะไม่ถูกเก็บจริงในฐานข้อมูล เมื่อผู้ใช้ต้องการดูข้อมูลจากวิวหรือต้องการจัดการข้อมูลภายในวิวเช่นการเพิ่ม การปรับปรุงหรือการลบข้อมูลจากวิว ผู้ใช้ก็สามารถใช้คำสั่ง SQL จัดการกับวิวได้โดยตรง โดยผู้ใช้เองไม่ทราบว่าตารางที่ตนใช้นั้นเป็นวิวที่ถูกสร้างขึ้นก็ได้ DBMS จะทำการแปลงคำสั่ง SQL ที่ผู้ใช้คำสั่งที่เป็นการกระทำต่อตารางจริงแทน

#### คำสั่งการสร้างวิว

จะมีการสร้างวิวโดยใช้คำสั่ง CREATE VIEW แล้วตามด้วยชื่อวิวที่ตั้งขึ้น และคำสั่ง SELECT ที่เป็นคำสั่งดึงข้อมูลบางส่วนจากตารางจริงนำมาสร้างเป็นวิว รูปแบบทั่วไปมีดังนี้

```
CREATE VIEW <ชื่อวิวที่ตั้งขึ้น>
[( <ชื่อคอลัมน์_1> , <ชื่อคอลัมน์_2> , ... )]
AS <ประโยคคำสั่ง SELECT>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 33 ต้องการสร้างวิวคอนงานที่ได้รับมอบหมายให้ไปทำงานก่อสร้างที่ตึกรหัส 311 เฉพาะคนที่มี ความชำนาญทางด้านไฟฟ้า

```
CREATE VIEW ELEC_WORK (NAME,BG_ID,ST_DT,TOT_HR)
AS SELECT WK_NAME,BLDG_ID,START_DATE,TOT_HOUR
FROM WORKER,ASSIGN
WHERE WORKER,WK_ID = ASSIGN,WK_ID
AND SKILL_TYPE = "ไฟฟ้า"
AND BLDG_ID = 311
```

คำอธิบาย - คำสั่งข้างต้นนี้เป็นคำสั่งให้มีการสร้างวิวชื่อ ELEC\_WORK โดยจะทำการดึงข้อมูลที่เป็นชื่อคนงาน (WK\_NAME) รหัสสถานที่ก่อสร้าง (BLDG\_ID) วันที่เริ่มทำงาน(START\_DATE) และจำนวนช.ม.ทั้งหมดที่ทำ (TOT\_HOUR) จากตารางคนงาน(WORKER) และ การทำงาน (ASSIGN) โดยเงื่อนไขหลัง WHERE ว่ารหัสคนงาน ในตารางคนงานและตารางการทำงานต้องเหมือนกัน (เพื่อเป็นการเชื่อมข้อมูลจากตารางทั้งสอง) และนอกจากนี้จะต้องเป็นคนงานที่ทำงานที่ตึกรหัส 311 และเป็นคนงานที่มีความชำนาญทางด้านไฟฟ้าอีกด้วย

- คำสั่งนี้มีการกำหนดชื่อคอลัมน์ของวิวใหม่ให้แตกต่างจากชื่อคอลัมน์ของตารางจริงที่ทำการดึงข้อมูลมานั้นคือ ข้อมูลจาก WK\_NAME จะถูกนำไปเก็บอยู่ในคอลัมน์ NAME ข้อมูลจาก BLDG\_ID จะถูกเก็บใน BG\_ID ข้อมูลจาก START\_DATE จะถูกเก็บอยู่ใน ST\_DT ข้อมูลจาก TOT\_HOUR จะถูกเก็บใน TOT\_HOUR ตามลำดับ

วิว ELEC\_WORK

ชื่อคนงาน NAME	รหัสสถานที่ก่อสร้าง BG_ID	วันที่เริ่มทำงาน ST_DT	จำนวนช.ม.ทั้งหมดที่ทำ TOT_HR
สุดใจ ศิพร้อม	311	08/01/40	40
พิชัย ใจซื่อ	311	01/02/40	20

ตารางที่ 2.12 แสดงวิว ELEC\_WORK

ดังนั้นผู้ใช้จะสามารถมองเห็นข้อมูลเฉพาะชื่อคนงาน รหัสสถานที่ก่อสร้าง วันที่เริ่มทำงาน และจำนวนช.ม. ทั้งหมด ที่ทำจากตาราง (วิว) ที่ชื่อ ELEC\_WORK เพื่อเป็นการป้องกันข้อมูลในส่วนอื่น ๆ ที่ไม่ต้องการให้ผู้ทราบได้ คือ ไปลองดูว่าถ้าผู้ใช้ทำการออกคำสั่งต่อไปนี้

```
SELECT * FROM ELEC_WORK
```

```
WHERE TOT_HR > 30
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับครูอาจารย์ในการเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งข้างต้นนี้ DBMS จะทำการแปลงให้เป็นคำสั่งที่ใช้จัดการกับตารางจริงดังนี้

```
SELECT NAME,BLDG_ID,START_DATE,TOT_HOUR
FROM WORKER,ASSIGN
WHERE WORKER.WK_ID = ASSIGN.WK_ID
AND SKILL_TYPE = "ไฟฟ้า"
AND TOT_HOUR > 30
```

ผลลัพธ์คือ

NAME	BG_ID	ST_DT	TOT_HR
สุจิต ธิพร้อม	311	08/01/40	40

ซึ่งเป็นรูปแบบคำสั่งที่ซับซ้อนพอสมควร ดังนั้นจะเห็นว่าการใช้วินนอกจากจะเป็นการป้องกันข้อมูลจากผู้ที่ไม่มีส่วนเกี่ยวข้องกับข้อมูลแล้ว ยังทำให้ผู้ใช้วินนั้นสามารถออกคำสั่งค้นหาข้อมูลจากวิวได้ง่ายกว่าการออกคำสั่งค้นหาข้อมูลจากตารางจริงโดยตรง

### ปัญหาที่เกิดจากการปรับปรุงและการเพิ่มข้อมูลลงในวิว

ดังได้กล่าวแล้วว่าวิวที่สร้างขึ้น ผู้ใช้สามารถใช้คำสั่ง SQL ไม่ว่าจะเป็นการเรียกดูข้อมูล การปรับปรุงหรือการเพิ่มข้อมูลลงไป ในวิว โดย DBMS จะแปลงคำสั่งให้เป็นคำสั่งที่กระทำกับตารางจริงแทน ซึ่งในการปรับปรุงและการเพิ่มข้อมูลลงในวิวบางครั้งอาจเกิดปัญหาขึ้นได้

### ตัวอย่างการปรับปรุงข้อมูลในวิว

ถ้าผู้ใช้วิวที่สร้างขึ้นในตัวอย่าง 30 ต้องการปรับปรุงข้อมูลในวิว โดยต้องการเปลี่ยนรหัสสถานที่ก่อสร้างจากรหัส 311 เป็น 531 ดังนี้

```
UPDATE ELECT_WORK
SET BG_ID = 531
WHERE NAME = "พิชัย ใจซื่อ"
```

คำสั่งนี้จะถูกแปลงให้เป็นคำสั่ง UPDATE ที่ทำกับตารางการทำงาน แต่จุดประสงค์ของการสร้างวิว ELECT\_WORK นี้ ก็เพื่อให้เก็บข้อมูลคนงานที่ทำงานอยู่ที่ตึกรหัส 311 และมีความชำนาญด้านไฟฟ้าเท่านั้น ถ้าไปทำการเปลี่ยนข้อมูลรหัสตึกเป็นอย่างอื่นก็จะทำให้จุดประสงค์ของการสร้างวิวไม่ถูกต้องได้

### ตัวอย่างการเพิ่มข้อมูลในวิว

ในทำนองเดียวกันถ้ามีการเพิ่มข้อมูลดังนี้

```
INSERT INTO ELECT_WORK
VALUES ("สมพงษ์ ใจกล้า"531,20/01/40,20)
```

จะเห็นว่าคอลัมน์รหัสสถานที่ก่อสร้างที่เพิ่มเข้าไปเป็น 531 ไม่ใช่ 311 ดังนั้นจึงเป็นอีกตัวอย่างหนึ่งที่เกิดปัญหาทำให้ไม่\*\*\*\*\*

นอกจากนี้คำสั่ง INSERT ข้างต้นต้องแน่ใจว่ามีคนงานชื่อ สมพงษ์ ใจกล้าเก็บอยู่ในตารางคนงานแล้วและ DBMS ต้องมีความสามารถในการรับรู้รหัสสำหรับคนงานคนนั้นจากตารางคนงานได้ ซึ่งก็เป็นเรื่องยากพอสมควร แต่ถ้ายังไม่มีรหัสคนงานนี้อยู่ในตารางคนงาน การเพิ่มคนงานชื่อนี้ลงไปในการทำงาน อาจมีการใส่รหัสคนงานให้เป็นค่าว่าง (NULL) ซึ่งจะเป็นปัญหาตามมาอีกได้ เนื่องจากส่วนประกอบของคีย์ต้องไม่เป็นค่าว่าง

ใน DBMS บางตัวได้แก้ปัญหานี้โดยการหาคำสั่งที่ช่วยป้องกันการเกิดเหตุการณ์ดังกล่าวด้วยคำสั่ง WITH CHECK OPTION โดยใส่ไว้ในคำสั่ง CREATE VIEW

```
CREATE VIEW ELEC_WORK(NAME,BG_ID,ST_DT,TOT_HR)
AS SELECT WK_NAME,BLDG_ID
      START_DATE,TOT_HOUR
FROM WORKER,ASSIGN
WHERE WORKER.WK_ID = ASSIGN.WK_ID
AND SKILL_TYPE = "ไฟฟ้า"
AND BLDG_ID = 311
```

**WITH CHECK OPTION**

ดังนั้น ถ้ามีการสร้างวิวและใช้คำสั่ง WITH CHECK OPTION กำกับไว้ด้วย จะมีผลให้การปรับปรุงหรือการเพิ่มขึ้นมูลกับวิวทุกครั้ง DBMS จะมีการตรวจสอบว่าขัดกับเงื่อนไขของวิวที่สร้างหรือไม่ ถ้าขัดก็จะไม่อนุญาตให้กระทำการปรับปรุงหรือเพิ่มข้อมูลนั้นได้

## 2.6 PHP

PHP เวอร์ชัน 3.0 เป็นหนึ่งในภาษาสคริปต์ (scripting language) ที่ใช้กันแพร่หลายที่สุด มักจะพบในอินเทอร์เน็ต ซึ่งมีความยืดหยุ่นสูงสำหรับนักเขียนโปรแกรมบนเว็บ งานบางอย่างอาจจะยากเมื่อใช้ภาษาอื่น แต่จะเขียนด้วย โค้ด PHP เวอร์ชัน 3.0 เพียงไม่กี่บรรทัด ซึ่งความเป็นจริงแล้ว โค้ด PHP เวอร์ชัน 3.0 สามารถใช้ได้เทียบเท่า HTML ซึ่งทำให้การเขียนภาษาสะดวกขึ้น

ซึ่งสร้างขึ้นโดย Rasmus Lerdorf และเริ่มแพร่หลายในวงกว้าง หลังจากนั้นก็มีผู้นำมาปรับปรุงจนได้เป็น PHP เวอร์ชัน 3.0 ซึ่งในที่สุดภาษานี้ก็ได้รับความนิยมเชื่อถือและพัฒนาระบบ จนกระทั่งมีการให้ควอน์โทลด์ฟรีในเว็บ

PHP เวอร์ชัน 3.0 ส่วนใหญ่จะเป็นการร่วมเอาคอนเซ็ปต์ (concept) ของ Perl, Java และ ภาษา C โครงสร้าง syntax ที่ส่วนใหญ่จะคล้ายกับภาษา C ทำให้ง่ายต่อการเรียนรู้แม้สำหรับนักโปรแกรมเมอร์เริ่มต้น PHP เวอร์ชัน 3.0 จะมีฟังก์ชันทางการคำนวณคณิตศาสตร์ การให้ข้อมูลทางด้าน network และยังให้ความยืดหยุ่นทางด้าน mail และ ฟังก์ชันพื้นฐาน และอื่น ๆ คุณสมบัติที่เด่นที่สุดของ PHP เวอร์ชัน 3.0 คือ ความสามารถ ด้าน database interfacing ซึ่งปกติการเชื่อมต่อ database กับอินเทอร์เน็ตเป็นเรื่องที่ทำให้ยาก นอกจากนี้มันยังรองรับ การทำงานของ database server เกี่ยวกับการตลาด

ปัจจัยที่ทำให้ PHP เวอร์ชัน 3.0 มีความสามารถมากเนื่องจากมันเป็นภาษา oriented มันถูกเขียนเพื่อให้งานนั้นสำเร็จได้อย่างรวดเร็ว และเป็นระเบียบ ก่อนที่จะได้ศึกษาต่อไปจำเป็นต้องรู้จุดประสงค์สำคัญดังต่อไปนี้คือ

1. PHP เวอร์ชัน 3.0 เป็นภาษาสคริปต์ แบบ server-side ( server-side scripting language ) ดังนั้นจึง ต้องมีการติดตั้ง language interpreter บน server ก่อนจะมีการใช้งานคำสั่ง ซึ่งสามารถติดต่อผู้ให้บริการอินเทอร์เน็ตเพื่อขอข้อมูลเพิ่มเติมได้ หรือถ้าหากว่าเป็น server administrator สามารถติดต่อได้ที่ <http://www.php.net> เพื่อควอน์โทลด์โปรแกรม
2. PHP เวอร์ชัน 3.0 ได้ถูกเขียนใหม่แทนเวอร์ชันเก่า ดังนั้น ควรสอบถามผู้ให้บริการอินเทอร์เน็ตถึงเวอร์ชันที่ติดตั้ง หากเป็นเวอร์ชัน 2.0 ควรอัปเดต เนื่องจากเวอร์ชัน 3.0 จะมีความเร็วและความสามารถสูงกว่า สำหรับ syntax ในโครงการนี้จะให้ ของเวอร์ชัน 3.0

### 2.6.1 แนะนำ PHP เวอร์ชัน 3.0

PHP เวอร์ชัน 3.0 สามารถใช้แทน HTML ได้ ซึ่งจะทำให้การรวม dynamic information กับ static document ได้ง่ายขึ้น เช่นถ้าเราต้องการให้มีการแสดงวันเดือนปีลงใน HTML document ยกตัวอย่างดังต่อไปนี้

```

<HTML>
  <HEAD>
    <TITLE>Our first PHP3.0 script</TITLE>
  </HEAD>
  <BODY>
    <CENTER>Our first PHP3.0 script</CENTER>
    <?
      /* the above "<?" signals that the PHP script has begun */
      $today = date("Y-m-d");
      PRINT "<CENTER>Today is: $today.</CENTER>";
    # the following ">" closes the script
    ?>
  </BODY>
</HTML>

```

ดังนั้นหากวันนี้เป็นวันที่ 27 สิงหาคม 1998 จะได้ output ดังต่อไปนี้

```

Our first PHP3.0 script
Today is: 1998-08-27.

```

ข้อควรสังเกต

1. การเขียนคำสั่งใน PHP เวอร์ชัน 3.0 จะต้องอยู่ภายใต้เครื่องหมาย "<?" และ ">" ส่วนอีกวิธีหนึ่งคือ ให้เขียนคำสั่งภายใต้เครื่องหมาย "<?php.....?>"
2. การเขียนคอมเมนต์ (comments) สามารถเขียนกับ script ได้ แต่อยู่ภายใต้เครื่องหมาย "/\*" และ "\*/" หรือเขียน เครื่องหมาย # คั่นบรรทัดที่ต้องการเขียน comment
3. สำหรับ statement ที่ต้องการแสดงออกทางหน้าจอต้องอยู่ภายใต้เครื่องหมาย " และนำด้วยคำสั่ง PRINT
4. คำสั่งส่วนใหญ่ของ PHP เวอร์ชัน 3.0 จะลงท้ายด้วย เครื่องหมาย ;
5. คำสั่ง HTML ที่ถูกแทนที่ด้วยคำสั่ง PRINT จะต้องถูกแปลงคำสั่งด้วย browser และแสดงสถานะการทำงานปกติ
6. สำหรับเอกสารที่มี statement ของ PHP เวอร์ชัน 3.0 อยู่จะต้องบันทึกด้วยรูปแบบของ \*.php3 เช่น myphpfile.php3 ซึ่งจะเป็นตัวบอกให้ interpreter ของ PHP เวอร์ชัน 3.0 จัดการกับคำสั่งที่พบใน script
7. เราสามารถเขียนการจัดการคำสั่งวันเดือนปีดังนี้

**Syntax: string date(string format, int timestamp);**

ฟังก์ชันนี้สามารถมีตัวแปรให้เลือกได้ 2 ตัวแปร (timestamp) ถ้าใช้กับ timestamp ฟังก์ชันจะ

ทำการคืนค่า string ซึ่งจะบรรจุรูปแบบของวันเดือนปีไว้ ดังตัวอย่างข้างต้น Y-m-d

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะให้ค่าของปี เดือน และวัน ซึ่งจะอยู่ในรูปแบบตัวเลขทั้งหมด สำหรับรูปแบบอื่นของตัวอักษรที่จะแทนที่ได้ในรูปแบบ string ก็จะมีอีกมากมาย

### 2.6.2 statement ซึ่ครวม

ความสามารถในการใช้งาน PHP เวอร์ชัน 3.0 อีกอย่างหนึ่งคือการสร้าง HTML templates ซึ่งจะมีประโยชน์มากเมื่อผู้ใช้งานกำลังพัฒนาการรวม page หลายๆ อันเข้าเป็น ไซค์รวม ยกตัวอย่างเช่นผู้ใช้ต้องการจะใส่ footer ที่ท้ายหน้ากระดาษแต่ละหน้าเช่น

Copyright © 1997-99 ngenuity. All rights reserved.

ซึ่งจะเกิดอะไรขึ้นหากต้องพิมพ์เช่นนี้ไปทุกหน้าหากเอกสารนั้นมีถึง 50 หน้ากระดาษ หมายความว่าต้องใช้เวลามากและใช้ความพยายามอย่างสูงทั้งยังอาจเกิดความคิดพลาดกับ statement อื่นในหน้านั้นด้วย

อย่างไรก็ตาม PHP เวอร์ชัน 3.0 ได้ตอบสนองการใช้งานด้านนี้โดยให้ผู้ใช้สามารถนำไฟล์ต่างหากมารวมอยู่ในเอกสารของ HTML ด้วยการแทรกเอาข้อความ copyright ข้างต้นไว้ในไฟล์แยกต่างหากและบันทึกเป็น footer.txt หลังจากนั้นผู้ใช้ก็เพียงแค่แทรกลงไปในส่วนท้ายของไฟล์ HTML

```
<? include("footer.txt")?>
```

และ footer จะปรากฏหลังจากโหลดเข้าสู่ browser ถ้าหากว่าไซค์ทั้ง 50หน้ามีการเปลี่ยนแปลงส่วน copyright เราสามารถเปลี่ยนแปลงไฟล์ footer ได้เพียงแค่เปิดไฟล์ footer และเปลี่ยนแปลงตามที่ต้องการ หลังจากนั้น ทั้ง 50หน้าจะเปลี่ยนแปลงเองโดยอัตโนมัติ

### 2.6.3 รูปแบบและตัวแปรของ HTML

ความสามารถในการทำงานของ PHP เวอร์ชัน 3.0 อีกอย่างคือ ความสามารถในการเปลี่ยนแปลงตัวแปรผ่านรูปแบบของ HTML ด้วยตัวแปรเหล่านี้ ผู้ใช้สามารถทำงานต่างๆ ได้ง่าย เช่น ส่งอีเมลล์ผ่านเว็บ ให้เอาท์พุทออกทางหน้าจอ รับและส่งข้อมูลจากฐานข้อมูล แสดงคังต่อไปนี่

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Request for more information</TITLE>
```

```
<BODY>
```

```
<CENTER>Would you like more information about our company?
```

```
<P>
```

```
<TABLE WIDTH = 400><TR><TD align = right>
```

```
<FORM ACTION="email.php3" METHOD="POST">
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Your name:<BR>

```
<INPUT TYPE="text" NAME="name" SIZE="20" MAXLENGTH="30">
```

```
<P>
```

Your email address:<BR>

```
<INPUT TYPE="text" NAME="email" SIZE="20" MAXLENGTH="30">
```

```
<P>
```

I prefer:

```
<SELECT NAME="preference">
```

```
<OPTION value = Apples>Apples
```

```
<OPTION value = Oranges>Oranges
```

```
</SELECT>
```

```
<P>
```

```
<INPUT TYPE="submit" VALUE="Send it!">
```

```
</FORM>
```

```
</TD></TR></TABLE></CENTER>
```

```
</BODY>
```

```
</HTML>
```

บันทึก script ข้างต้น เป็น *moreinfo.html*

ซึ่งคำสั่ง ACTION จะเรียกใช้ ไฟล์ *email.php3* ซึ่งจะมีคำสั่งเหล่านี้ปรากฏอยู่

*The email.php3 file:*

```
<?
```

```
/* this script will handle the variables passed from the moreinfo.html file */
```

```
PRINT "<CENTER>";
```

```
PRINT "Hello, $name.";
```

```
PRINT "<BR><BR>";
```

```
PRINT "Thank you for your interest.<BR><BR>";
```

```
PRINT "We will send information to $email, and have noted that you like $preference.";
```

```
PRINT "</CENTER>";
```

```
?>
```

อย่าลืมบันทึก ไฟล์ข้างต้นนี้เป็นชื่อ *emai.php3*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่ ผู้ใช้พิมพ์ชื่อและอีเมลลงในแบบฟอร์ม HTML และกดปุ่ม Send it! แบบฟอร์มจะเรียกไฟล์ email.php3 และจะแสดงผลที่ออกมา(เช่น ชื่อว่า Bill, มีอีเมลแอดเดรสเป็น [bgates@devshed.com](mailto:bgates@devshed.com) และชอบ Apples)

Hello, Bill.

Thank you for your interest.

We will send information to [bgates@devshed.com](mailto:bgates@devshed.com), and have noted that you like Apples.

อย่างไรก็ตาม โปรแกรมของเรายังไม่สมบูรณ์แบบทีเดียว เพราะ โปรแกรมของเราจะทำการโต้ตอบกับผู้ใช้ แต่ไม่มีการเก็บข้อมูลของผู้ใช้ไว้ ดังนั้นเราจึงไม่สามารถส่งอีเมลกลับไปยัง Bill ได้

ทางเดียวที่จะทำเช่นนั้นได้ก็คือ เพิ่มคำสั่ง MAIL ของ PHP3.0 เข้าไป

**Syntax:** void mail(string to, string subject, string message, string add\_headers);

To - ผู้ที่ต้องการส่งเมลล์ไปให้

Subject - ข้อความที่ต้องการให้แสดงที่ต้นเรื่อง

Message - ข้อความที่ต้องการส่ง

Add\_headers - สำหรับแทรก string ไว้ที่ header

ถ้าเราแทรกคำสั่งข้างต้นนี้ไว้ท้ายคำสั่ง PRINT ใน script ข้างต้น เราจะสามารถส่ง อีเมลไปยังผู้ที่ขอข้อมูล และ ไปยัง administrator ได้ ซึ่งเราจะรู้ได้โดยการเขียน script

```
mail("$email", "Your request for information", "$name\n
```

```
Thank you for your interest!\n
```

```
We sell fresh corn daily over the Internet!
```

```
Place your order at http://www.buycom.com,
```

```
and receive a free package of $preference!");
```

```
mail("administration@buycom.com",
```

```
"Visitor request for info.",
```

```
"$name requested for information.\n
```

```
The email address is $email.\n
```

```
The visitor prefers $preference.");
```

ฟังก์ชันเมลล์ที่จะทำงานเมื่อเราได้ติดตั้ง SENDMAIL บน server แล้ว ซึ่งปกติทั่วไปจะมีอยู่แล้ว แต่ควรตรวจสอบกับผู้ให้บริการอินเทอร์เน็ตก่อนที่จะมีการใช้ฟังก์ชันนี้

ถ้าเกิดมีผู้เข้ามาใช้ข้อมูลมากเกินไป เราจะไม่สามารถเก็บอีเมลล์ได้ทั้งหมด หรือถ้าเราต้องการถามระบบว่าใครชอบแอปเปิ้ลมากกว่าส้ม เราสามารถใช้ส่วนดีของ ระบบฐานข้อมูล(database) มาช่วยได้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างมาก หนึ่งในระบบฐานข้อมูลที่เร็วที่สุดในด้านการตลาด คือ MySQL ซึ่งมีความสามารถในด้านความเร็วและความง่ายต่อการใช้งาน รวมทั้งความยืดหยุ่นในการใช้งานร่วมกับ PHP3.0 ด้วย

## 2.7 World Wide Web (WWW)

จากการที่เครือข่ายอินเทอร์เน็ตมีพัฒนาการที่รวดเร็ว มีการประยุกต์อย่างกว้างขวางบนเครือข่าย จนทำให้อัตราการขยายตัวของผู้ใช้บนเครือข่ายเพิ่มสูงขึ้นเป็นทวีคูณ เครือข่ายที่มีอยู่จึงสร้างหนทางเชื่อมโยงเข้าสู่อินเทอร์เน็ตเพื่อส่งข้อมูลข่าวสารระหว่างกัน

ความต้องการใช้ข้อมูลข่าวสารของทุกคนที่อยู่บนเครือข่ายได้ถึงจุดหนึ่งที่สำคัญจากเดิมใช้ส่งผ่านข้อความ เช่น จดหมายอิเล็กทรอนิกส์ การพูดคุย การอ่านข่าว การใช้ทรัพยากร คอมพิวเตอร์ร่วมกัน (Telnet) การโอนย้ายข้อมูล (FTP) ข้อมูลที่ส่งผ่านยังอยู่ในกรอบของโปรแกรมประยุกต์นั้น ๆ

เพื่อให้การทำงานแบบออนไลน์ได้ผลและเกิดประโยชน์ จึงเกิดแนวความคิดที่สร้างฐานข้อมูลแบบกระจาย เสมือนให้แต่ละหน่วยงานกองข้อมูลข่าวสารที่ต้องการให้บริการไว้บนเครือข่ายกองข้อมูลเหล่านี้เปิดให้ใครก็ได้เรียกเข้ามาเปิดอ่าน การบริการลักษณะนี้จึงเสมือนมีหนังสือข้อมูลข่าวสารไว้ให้จำนวนมาก กระจัดกระจายเชื่อมโยงกันอยู่ทั่วโลก และหนังสือเหล่านี้เป็นหนังสือที่เรียกว่า ไฮเปอร์เท็กซ์ (Hyper Text)

World Wide Web หรือ WWW หรือ W3 หรือ จะเรียกกันสั้น ๆ ว่า เว็บ (Web) เป็นรูปแบบหนึ่งของระบบการเชื่อมโยงเครือข่ายข่าวสาร ใช้ในการค้นหาข้อมูลข่าวสารบนอินเทอร์เน็ตจากแหล่งข้อมูลหนึ่งไปยังแหล่งข้อมูลที่อยู่ห่างไกลออกไปให้มีความง่ายต่อการใช้งานมากที่สุด

WWW ครั้งแรกเกิดขึ้นที่ห้องปฏิบัติการวิจัยเซิร์น แห่งประเทศสวีเดนแลนด์ ซึ่งทำการวิจัยทางด้านฟิสิกส์ กลุ่มนักวิจัยเน้นให้เป็นระบบสนับสนุนการส่งข้อมูลข่าวสารการเรียกดูข้อมูลข่าวสารที่อยู่ในรูปของไฮเปอร์เท็กซ์ ต่อมา มีสถาบันและมหาวิทยาลัยอีกหลายแห่งได้ร่วมกันและพัฒนาอย่างต่อเนื่อง จนทำให้ WWW เป็นมาตรฐานและสามารถเชื่อมโยงถึงกันอย่างเป็นระบบ นอกจากนี้ผู้พัฒนายังเน้นรูปแบบการใช้งานง่าย มีประสิทธิภาพ ฐานข้อมูลแบบ WWW จึงเป็นที่รู้จักและมีความนิยมใช้ในขณะนี้

WWW จะแสดงผลอยู่ในรูปแบบของเอกสารไฮเปอร์เท็กซ์ ซึ่งเป็นฐานข้อมูลชนิดหนึ่งทำหน้าที่รวบรวมข่าวสารข้อมูลที่อยู่คล้ายกับเส้นใยแมงมุมที่ถักทอเส้นสายเชื่อมโยงกันไปมา แม้ว่าจะมีเส้นใยจำนวนมาก แต่ละเส้นจะถูกจัดวางทับกันมีจุดเชื่อมต่อที่ทำให้ตัวแมงมุมสามารถที่จะเดินทางไปยังจุดใด ๆ บนเส้นใยเหล่านี้ได้

การให้บริการของอินเทอร์เน็ตแบบ WWW เป็นระบบงานที่ทรงพลังมากในยุคปัจจุบันทำให้ระบบเครือข่ายอินเทอร์เน็ตเป็นที่นิยมใช้กันอย่างแพร่หลายทั่วโลกโดยแท้จริง สาเหตุที่บริการแบบ WWW ได้รับความนิยมมากที่สุดก็เนื่องจาก WWW สามารถให้บริการข้อมูลได้ทั้งในแบบข้อความ เสียง ภาพนิ่ง และภาพเคลื่อนไหว ทำให้ข้อมูลที่นำมาแสดงมีความน่าสนใจยิ่งขึ้นเป็นอันมาก

ในการค้นหาข้อมูลแบบ WWW จะมีการเชื่อมโยงข้อมูลตามเส้นทางที่กำหนดไว้เรียกว่า LINKS โดยในแต่ละจุดเชื่อมโยงจะมีเส้นทางเดินไปยังจุดต่อไปติดตั้งอยู่เป็นระยะในรูปแบบของข้อความหรือรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพ ทำให้ผู้ใช้สามารถเลือกที่จะเดินทางต่อไปยังจุดมุ่งหมายใดก็ได้ และที่นั่นผู้ใช้จะพบว่ายังมีทางเดินไปถึงจุดหมายอื่นได้อีก ทำให้สามารถค้นหาข้อมูลที่ได้รับ รายละเอียดมากยิ่งขึ้น

### 2.7.1 การเชื่อมโยงข้อมูลใช้โพรโตคอล http

เพื่อให้การรวมข้อมูลเป็นกลุ่มก่อนเข้าด้วยกัน เก็บไว้เพื่อให้เรียกใช้ได้จากที่ห่างไกล ข้อมูลที่เก็บไว้จึงเป็นการง่ายต่อการเรียกดู ข้อมูลที่นำมารวมอาจเป็นข้อมูลเท็กซ์ หรือเป็นตัวอักษร รูปภาพ ภาพเคลื่อนไหว เสียง ข้อมูลเหล่านี้จึงต้องมีมาตรฐาน

ไฮเปอร์เท็กซ์เป็นข้อมูลที่เก็บรวบรวมไว้เป็นหมวดหมู่ การจับเก็บนี้เก็บตามมาตรฐานรูปแบบที่ชื่อว่า html - hypertext markup language การจัดสร้างข้อมูลเพื่อให้เรียกเปิดดูได้โดยใช้โปรแกรมที่พัฒนาสำหรับการอ่านไฮเปอร์เท็กซ์ เช่น lynx, mosaic, netscape ส่วนผู้ใช้ต้องการเชื่อมโยงเข้าไปเปิดอ่านหรือที่เรียกว่า บราวเซอร์ (browser) ก็ใช้โพรโตคอล http (Hypertext Transfer Protocol) โดยโครงสร้างของไฮเปอร์เท็กซ์ตามฟอร์แมต html กำหนดรูปแบบที่ง่ายต่อการสร้าง ทำให้แต่ละหน่วยงานสามารถพัฒนาฐานข้อมูล WWW ของตัวเองขึ้นมาและเชื่อมโยงเข้าหากันได้โดยง่าย

### 2.7.2 Home Page

โฮมเพจเป็นผลผลิตของไฮเปอร์เท็กซ์ที่ผู้ใช้ไปใช้บริการบนอินเทอร์เน็ตพบเห็นกันได้มากที่สุด และเป็นจุดเด่นให้ผู้ใช้บริการเกิดความสนใจที่จะใช้อินเทอร์เน็ตมากขึ้นนอกเหนือจากบริการพื้นฐานทั่วไป

ผู้ให้บริการอินเทอร์เน็ต (ISP) และศูนย์บริการเว็บ (Web Site) ทุกแห่งต่างมีโฮมเพจเป็นของตนเองเพื่อเป็นข้อมูลเบื้องต้นที่แสดงให้ผู้ใช้บริการทราบว่าในสถานีนี้นั้น ๆ ให้บริการสิ่งใดบ้าง และเมื่อเข้าโฮมเพจนั้น ๆ แล้วเราจะเดินทางไปยังแห่งใดก็ได้ ทำหน้าที่เป็นจุดรวมของการเดินทางเข้าไปสู่ดินแดนแห่งใหม่ นับได้ว่าโฮมเพจเป็นหน้าตา เป็นการประชาสัมพันธ์ของสถานีหรือองค์กรนั้น และเราเรียกข้อมูลอื่น ๆ ซึ่งอาจเชื่อมโยงต่อจากโฮมเพจว่า “เว็บเพจ” (Web Page)

รูปร่างหน้าตาของโฮมเพจแต่ละแห่งจะแตกต่างกันออกไป ขึ้นอยู่กับประเภทและรูปแบบขององค์กรนั้น โฮมเพจที่ดีไม่ควรที่จะมีรูปภาพมากเกินไป เนื่องจากจะทำให้การทำงานช้าลง ไม่ควรมีหลายหน้าและควรจะมีส่วนแสดงให้ผู้ใช้ทราบว่าในสถานีนี้นั้นมีบริการอะไรอยู่บ้าง

ด้วยการพัฒนาของไฮเปอร์เท็กซ์ ได้เกิดภาษาใหม่ที่กลายมาเป็นมาตรฐานบนอินเทอร์เน็ตภาษาหนึ่ง นิยมนำมาใช้สร้างโฮมเพจและเว็บเพจบนอินเทอร์เน็ตกันนั้นคือ ภาษา HTML ที่กำลังจะกล่าวถึงต่อไป

## 2.8 ภาษา HTML

HTML ย่อมาจากคำว่า Hyper Text Markup Language เป็นรูปแบบหนึ่งของภาษา SGML (Standard Generalized Markup Language) นิยมใช้กันทั่วไปบนอินเทอร์เน็ต เหมือนกับที่เราใช้โปรแกรมระบบปฏิบัติการ DOS ซึ่งถูกคัดแยกออกมาจากโปรแกรมระบบปฏิบัติการ UNIX เช่นเดียวกับ HTML ซึ่งเป็นภาษาหลักสำหรับการสร้างโฮมเพจ เพิ่มเอกสาร HTML ที่สร้างขึ้นจะนำไปแสดงผลได้ด้วยโปรแกรมเว็บเบราว์เซอร์ เช่น โปรแกรม Netscape Navigator หรือ Mosaic

### 2.8.1 HTML ทำงานอย่างไร

การให้บริการบนอินเทอร์เน็ตไม่ว่าจะเป็น E-mail , FTP , Gopher , Telnet หรือบริการอื่น ๆ ต้องใช้อุปกรณ์เชื่อมต่อภายในอันซับซ้อนของฮาร์ดแวร์ที่สามารถทำงานได้ด้วยโปรแกรมเฉพาะที่ทำงานบนอินเทอร์เน็ตเท่านั้น

WWW แบ่งออกเป็น 2 ส่วนคือ ส่วนที่เป็น Client และส่วนที่เป็น Server เหมือนกับที่มีแบ่งกันในระบบเครือข่ายทั่วไป ทั้งสองส่วนจะถูกเชื่อมโยงถึงกันผ่านทางอินเทอร์เน็ต โดยมี HTML เป็นส่วนฐานข้อมูลสำคัญ เมื่อเว็บเบราว์เซอร์ส่งข้อความร้องขอข้อมูลที่อยู่ในรูปแบบของไฟล์ HTML จากเครื่องคอมพิวเตอร์ที่เราใช้งานอยู่ผ่านโมเด็มหรืออุปกรณ์สื่อสารข้อมูลอื่นไปยังศูนย์บริการอินเทอร์เน็ต (ISP) ตามโพรโตคอล (Protocol ที่กำหนดไว้ผ่านทาง URLs (Uniform Resource Locators) และเมื่อข้อมูลเดินทางมาถึงเว็บเซิร์ฟเวอร์ (Web Server) ศูนย์บริการปลายทางที่ผู้ใช้ต้องการ ณ ที่นี้เครื่องเว็บเซิร์ฟเวอร์ของศูนย์ฯ จะทำการอ่านข้อมูลที่ถูกส่งมาและจะทำงานตามคำสั่งที่กำหนด โดยอาจมีการเชื่อมโยงไปยังเว็บเซิร์ฟเวอร์อื่นอีก หลังจากจบสิ้นกระบวนการแล้วจะทำการจัดส่งข้อมูลคำตอบย้อนกลับมายังเครื่องคอมพิวเตอร์ที่เราใช้งานอยู่ โปรแกรมเว็บเบราว์เซอร์ที่เครื่องคอมพิวเตอร์ของเราจะแปลงสัญญาณคำสั่งและแสดงผลเป็นข้อความ รูปภาพ เสียง ให้เราได้ใช้งานกันต่อไป

จุดเด่นของไฮเปอร์เท็กซ์ที่ใช้ตามมาตรฐาน html คือการเชื่อมโยงเข้าหากันเอง ออบเจกต์ ทั้งเอกสาร รูปภาพ เสียง หรือภาพเคลื่อนไหว โดยให้ออบเจกต์เหล่านี้ เชื่อมโยงต่อระหว่างไฟล์เข้าด้วยกัน โครงสร้างการเชื่อมโยงเป็นการจัดการ ให้เป็นไปตามความต้องการ เพื่อให้เข้าใจได้ง่าย ลองพิจารณาจากโครงสร้างการเชื่อมโยง ซึ่งเป็นการเชื่อมโยงระหว่างกัน สังเกตว่าการเชื่อมโยงระหว่างไฟล์ ที่เป็นไฮเปอร์เท็กซ์ด้วยกันอาจจะอยู่ในเครื่องเดียวกันหรือต่างเครื่องกันก็ได้ ดังนั้นจึงฐานข้อมูลแบบนี้มีการโยงกันไปได้มาก ผู้พัฒนาผู้หนึ่งสร้างตัวเชื่อมโยงให้เรียกของอีกที่หนึ่งที่ห่างไกลก็ได้

HTML นอกจากใช้ในการสร้างฐานข้อมูลบนอินเทอร์เน็ตแล้ว ยังมีความสามารถทางด้านการเชื่อมโยงข้อมูลจากแหล่งข้อมูลอื่นบนอินเทอร์เน็ต ไม่ว่าจะเป็น E-mail , FTP , Gopher , Telnet หรือ News (ขึ้นอยู่กับชนิดของเว็บเบราว์เซอร์แต่ละชนิดว่ามีความสามารถหรือไม่) ทำให้เราสามารถเรียกใช้บริการเหล่านี้ได้ทันที ต่างจากเมื่อขณะเริ่มแรกที่มีการเปิดให้ใช้บริการ จะใช้บริการใดก็ต้องไปหาโปรแกรมที่ทำงานเฉพาะมาทำงาน

มีคำศัพท์ 2 คำที่ควรทราบเกี่ยวกับการสร้าง Web Page ด้วย HTML ได้แก่

-Text หมายถึง ข้อความต่างๆ ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-Tag หมายถึง คำสั่งที่ใช้ในการกำหนดรูปแบบของ Text โดยคำสั่งของ HTML จะอยู่ภายในเครื่องหมาย < และ > เช่น

This is a <B> Bold </B> word

ตัวอักษร a, b, c, ... คือ text

<B> และ </B> คือ Tag โดยทั่วไปจะมี Tag เริ่มต้นและ Tag จบท้ายโดยจะใช้ตัวเล็กหรือตัวใหญ่ก็มีผลเหมือนกัน จากตัวอย่างหมายถึงการทำข้อความ Bold ให้เป็นตัวเข้ม ผลลัพธ์ที่ได้คือ

This is a **bold** word

## 2.8.2 กฎในการใช้ HTML

-แต่ละ Tag จะต้องอยู่ในเครื่องหมาย <> เท่านั้น

-จะใช้ตัวพิมพ์เล็กหรือใหญ่ก็ได้ มีความหมายเดียวกัน เช่น <title>, <Title> หรือ <TITLE> ต่างก็ให้ความหมายเดียวกัน

-แทบทุก Tag จะมี Tag เปิด และ Tag ปิด โดย Tag ปิด จะมี / รวมอยู่ด้วย แต่ก็มีบาง Tag ที่ ไม่มีตัวปิด เช่น <BR>, <P> เป็นต้น

-ในการพิมพ์เอกสาร HTML นั้นจะพิมพ์ติดต่อกันทั้งข้อความและคำสั่งหรือพิมพ์แยกบรรทัดหรือเลาะช่องว่างก็แล้วแต่ผู้พิมพ์ แต่การพิมพ์ให้ดูสวยงามจะทำให้ง่ายต่อการอ่านและแก้ไขในภายหลัง

## 2.8.3 การทำงานของ HTML กับ Browser

ปกติถ้าเรานำข้อมูลที่สร้างจากโปรแกรมทั่ว ๆ ไปที่ไม่ใช่ HTML มาแสดงใน Web Browser ผลลัพธ์ที่ได้จะไม่ถูกต้องตามเอกสารต้นฉบับ เนื่องจาก Browser ไม่เข้าใจในโครงสร้างของข้อมูล แต่ก็แสดงผลให้ โดยนำข้อความทั้งหมดต่อกัน

Browser เป็นตัวแปลภาษา HTML ที่แปลได้อย่างตรงไปตรงมา ถ้าโครงสร้างผิด ๆ ก็จะแสดงผลแบบผิด ๆ ถ้าถูกก็จะแสดงผลให้ตรงตามที่ต้องการ ไม่มีกรบอกว่าผิดพลาด Error ที่ตำแหน่งใด เหมือนกับตัวแปลภาษาอื่น ๆ แต่จะแสดงผลลัพธ์ให้เห็นเลยไม่ว่าจะผิดหรือถูก ดังนั้นจึงจำเป็นต้องค้นหาส่วนที่ผิดให้ได้เองดังตัวอย่าง

Dve's Desk

West Lafayette, Indiana

3 November, 1996

Dear Reader,

Thank you for connecting to my web sever, but I regret to tell you that thing aren't up and running yet! They will be\_soon\_}but they aren't today.

Sincerely ,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dave Taylor

จะได้

Dve's Desk West Lafayette, Indiana 3 November, 1996 Dear Reader, Thank you for connecting to my web sever, but I regret to tell you that thing aren't up and running yet! They will be\_soon\_}but they aren't today. Sincerely ,Dave Taylor

#### 2.8.4 การจัดรูปแบบให้เป็น Paragraph และเป็นบรรทัด

จะใช้ <HTML> วางไว้ที่ต้นเอกสารและ </HTML> ไว้ที่ท้ายเอกสารเพื่อให้ Browser ทราบว่าเป็นเอกสาร HTML

จะใช้ <P> เป็นตัวจัดให้ข้อความที่ตามหลังมาขึ้นย่อหน้า (Paragraph) ใหม่ เพื่อความเข้าใจ ไม่มีคำสั่งปิดท้าย

จะใช้ <BR> เพื่อให้ข้อความขึ้นบรรทัดใหม่  
คำสั่งตัวอย่าง

```
<HTML> Dve's Desk
West Lafayette, Indiana
3 November, 1996
```

```
<P>
```

Thank you for connecting to my web sever, but I regret to tell you that thing aren't up and running yet!  
They will be\_soon\_,but they aren't today.

```
<P>
```

Sincerely ,

```
<P>
```

Dave Taylor

```
<HTML>
```

จะได้

Dve's Desk West Lafayette, Indiana 3 November, 1996

Dear Reader,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thank you for connecting to my web sever, but I regret to tell you that thing aren't up and running yet!  
They will be\_soon\_,but they aren't today.

Sincerely ,

Dave Taylor

### 2.8.5 การจัดรูปแบบข้อความให้ตรงตามเอกสารเดิม

ในบางครั้งเราจะได้ข้อมูลทีโอนมาจากโปรแกรมอื่น เช่น Excel ซึ่งจัดรูปแบบไว้เรียบร้อยแล้ว  
จึงเป็นการเสียเวลา ถ้าจะต้องมาใช้คำสั่ง HTML จัดซ้ำใหม่ หรือบางครั้งเราอาจต้องการจัดให้เสร็จเร็ว  
ร็อยใน Editor เลยก็ได้ ลักษณะนี้จะใช้คำสั่ง <PRE> และ </PRE> ไว้ท้ายข้อมูลดังตัวอย่าง

```
<PRE> Once upon a time I was a young man,
      But then I
          ate some walnuts
              Sold some pretzels
                  and juggled kittens. Or not.(/PRE)
```

จะได้

```
Once upon a time I was a young man,
      But then I
          ate some walnuts
              Sold some pretzels
                  and juggled kittens. Or not.
```

### 2.8.6 การแบ่งเอกสารออกเป็น ส่วน ๆ (Section)

กรณีทีเอกสาร HTML ที่สร้างมามีขนาดใหญ่ เราอาจแบ่งเอกสารออกเป็น ส่วน ๆ เพื่อให้การค้นหาหรือแก้ไขทำได้ง่ายขึ้น การใส่ชื่อเรื่อง และขอบเขตของเนื้อหาลงไป

<HEAD>,</HEAD> ใช้กำหนดชื่อเรื่องให้กับส่วนเอกสาร แต่จะไม่เห็นใน Browser

<BODY>,</BODY> ใช้กำหนดขอบเขตของเนื้อหาในเอกสารส่วนนั้น (เนื้อหาที่จะแสดงใน Browser)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังตัวอย่าง

```
<HTML>
```

```
<HEAD></HEAD>
```

```
<BODY>
```

```
Dve's Desk<BR>
```

```
West Lafayette, Indiana<BR>
```

```
3 November, 1996<BR>
```

```
<P>
```

```
Dear Reader,
```

```
<P>
```

Thank you for connecting to my web sever, but I regret to tell you that thing aren't up and running yet! They will Benson\_,but they aren't today.

```
<P>
```

Sincerely ,

```
<P>
```

Dave Taylor<BR>

```
</BODY>
```

```
</HTML>
```

เป็นการกำหนดว่าเอกสาร HTML นี้ชื่ออะไร (Head) และมีขอบเขตของเนื้อหาที่จะแสดงใน Browser อยู่ระหว่างช่วงใด (Body)

### 2.8.7 การกำหนดชื่อเอกสารให้แสดงบน Browser

สามารถแสดงเนื้อหาเรื่องไว้ใน Browser ได้ด้วยคำสั่ง <TITLE> และ </TITLE> โดยจะต้องวางไว้ระหว่างคำสั่ง <HEAD> และ </HEAD> เสมอ

### 2.8.8 การกำหนดขนาดของตัวอักษรในแต่ละ Section (Headline)

กรณีทีเอกสารถูกแบ่งออกเป็นส่วน ๆ และเราต้องการกำหนดขนาดของตัวอักษรในแต่ละ Section ให้แตกต่างกันออกไป ก็ทำได้ง่าย ๆ โดยใช้คำสั่ง <Hn> และ </Hn> โดย n แทนตัวเลข 1 ถึง 6 1 คือขนาดเล็กสุด และ 6 คือขนาดใหญ่สุด แก้ไขขนาดเองไม่ได้ เนื่องจากถูกกำหนดมาไว้แล้ว โดยผู้ออกแบบ HTML ข้อความที่กำหนดด้วย Headline จะเริ่มต้นที่บรรทัดใหม่ทุกครั้ง

ดังตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<H1> Monday </H1>

<H2> Tuesday </H2>

<H3> Wednesday </H3>

<H4> Thursday </H4>

<H5> Friday </H5>

<H1> Saturday </H1>

จะได้

**Monday**

Tuesday

Wednesday

Thursday

Friday

Saturday

### 2.8.9 การกำหนด Footer

เป็นข้อความที่นิยมใส่ไว้ด้านล่างเอกสาร เช่น ที่อยู่ที่เราจะติดต่อได้ ใช้คำสั่ง <ADDRESS>และ </ADDRESS> ข้อความที่อยู่ระหว่าง Tags นี้ถูกกำหนดให้เป็นตัวเอน (Italic) โดยอัตโนมัติ

### 2.8.10 รูปภาพ

ไฟล์รูปภาพหรือที่เรียกว่า Graphics Files (รูปภาพที่แสดงบน Web จะเรียกว่า “Image”) นั้นมีหลายชนิด โดยสังเกตจากนามสกุลของ File Web Browsers โดยทั่วไปจะใช้ Graphics File มาตรฐาน 2 ชนิดคือ GIF และ PEG ซึ่งโปรแกรม Graphics โดยทั่วไปสามารถ บันทึกให้รูปภาพที่เกิดจากการสร้างของตัวเองให้เป็น GIF หรือ JPEG หรือทั้ง 2 ชนิดก็ได้ หรือมีโปรแกรมที่เราใช้งานอยู่เกิดไม่สามารถบันทึกเป็นชนิดทั้ง 2 ชนิดก็ได้เพราะมีโปรแกรมที่สามารถเปลี่ยนแปลง File นามสกุลอื่น ๆ ให้เป็น JPEG หรือ GIF ได้ เช่น LVIEW Pro, Graphics Workshop ซึ่งโปรแกรมที่สามารถ Load มาใช้งานได้จาก Internet

#### 2.8.10.1 GIF Format

เพิ่มข้อมูลที่อยู่ในรูปแบบของ GIF (Graphics Image Format) จะใช้เทคนิคที่เรียกว่า “Lossless Compression” คือเมื่อผ่านการ Compress แล้ว จะมีการสูญหายของข้อมูลน้อย นั่นคือ ได้ภาพที่เหมือนต้นเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฉบับมาก แต่ข้อเสียก็คือ Software ที่จะสนับสนุนข้อมูล ของ GIF มีน้อย เพราะถ้าใครจะพัฒนา Software ใหม่ๆ จะต้องเสียค่าลิขสิทธิ์ให้บริษัทผู้คิด GIF ขึ้นมาด้วยทำให้ Software ที่จะสนับสนุน GIF มีประสิทธิภาพไม่คึกและภาพที่ได้ก็ใช้สีได้สูงสุดเพียง 256 สีเท่านั้น ซึ่งไม่เพียงพอในปัจจุบัน

ปัจจุบันรูปแบบ Files ของ GIF จะเป็น GIF89a ซึ่งแต่เดิมจะเป็น GIF87a

### 2.8.10.2 JPEG Format

File ที่อยู่ในรูปของ JPEG (joint Photographic Expert Group) ใช้เทคนิคที่เรียกว่า “Lossy Compression “ คือ มีโอกาสมากที่ทำให้ได้ไฟล์ที่ไม่เหมือนต้นฉบับ แต่สิ่งที่มีประโยชน์มากก็คือ ใช้สีได้สูงสุดถึง 16.7 ล้านสี ทำให้ JPEG เป็นรูปแบบที่นิยมกันมาก นามสกุลของไฟล์ที่เป็น JPEG อาจมีทั้ง .JPG, .JPEG หรือ .JPE software ที่สนับสนุน JPEG มีมาก และพัฒนากันไปอย่างไม่หยุดยั้ง เพราะไม่เสียค่าลิขสิทธิ์

### 2.8.11 การใส่รูปภาพ

ทำได้โดยใช้คำสั่ง <IMG> ร่วมกับตัวเลือก src ใส่งไปใน Source Code ของ HTML เช่น

```
<IMG SRC="KLOGO.GIF">
```

### 2.8.12 การกำหนดความสูงและความกว้างของรูปภาพ

มี 2 Attribute ที่สามารถนำมาใช้กำหนดขนาดของรูปภาพได้คือ HEIGHT และ WIDTH โดยกำหนดขนาดมาตรฐานของ Pixel โดยการกำหนดขนาดมีข้อควรระวังคือ ขนาดของรูปภาพ ถ้ามีการเปลี่ยนแปลงหรือแก้ไขในภายหลัง เช่น ทำให้เสีกลง Browser จะไม่จัดรูปแบบของเอกสารให้ใหม่ แต่จะยังคงให้ส่วนอื่น ๆ อยู่ในตำแหน่งเดิม ทั้งนี้เพื่อไม่ให้กระทบกระเทือนถึงกันนั่นเอง

ตัวอย่าง <IMG SRC="klogo.GIF" WIDTH=200 HEIGHT=20>

การแทนที่รูปภาพด้วยข้อความ (Alternative)

ALT เป็น Parameter ตัวหนึ่งที่ใช้กับ <IMG> ใช้กำหนดให้แสดงคำหรือข้อความแทนที่ตรงส่วนที่เป็นรูปภาพ มีประโยชน์กับ Browser ที่เป็นแบบ Text ไม่สามารถแสดงรูปภาพได้ เช่น Lynx เพื่อหลีกเลี่ยงปัญหาที่อาจเกิดขึ้นได้ หรือ ประโยชน์อีกอย่างหนึ่งก็คือ ให้แสดงคำหรือข้อความให้ผู้ใช้เห็นก่อนที่จะ Load รูปภาพขึ้นมา ตัวอย่าง <IMG SRC="KLOGO.GIF" ALT=" This is a wonderful picture">

### 2.8.13 การวางตำแหน่งของรูปภาพ (Align)

เป็น Parameter ที่ใช้ในการกำหนดให้คำหรือข้อความในรูปภาพ หรือตัวรูปภาพเองวางอยู่ในตำแหน่งใดของหน้าใน NETSCAPE สามารถระบุการวางตำแหน่งได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ALIGN=LEFT วางไว้ที่ด้านซ้ายของหน้าเอกสารส่วน text ที่ตามมาจะถูกจัดให้อยู่รอบๆ ทางด้านซ้าย บน และล่างของรูปภาพ

ALIGN=RIGHT วางไว้ทางด้านขวาของหน้าเอกสาร ส่วน text ที่ตามมาจะถูกจัดให้อยู่รอบๆ ทางด้านซ้าย บน และล่างของรูปภาพ

ALIGN=ABSMIDDLE วางส่วนกลางของรูปภาพ ไว้ที่กลางบรรทัดของ Text

ALIGN=ABSBOTTOM วางส่วนล่างของรูปภาพ ไว้ที่เส้นบรรทัด

ALIGN=TOP วางด้านบนของรูปภาพ ไว้ที่บรรทัดบนสุด

ALIGN=MIDDLE วางส่วนกลางของรูปภาพ ไว้ในตำแหน่งกลางของ text

ALIGN=BOTTOM วางส่วนกลางของรูปภาพ ไว้ที่บรรทัดล่างของ text

#### 2.8.14 การกำหนดระยะในแนวดิ่งและแนวนอน (VSPACE, HSPACE)

กำหนดระยะห่างของรูปภาพในแนวดิ่งและแนวนอนได้ โดยการใช้ตัวเลือก HSPACE และ VSPACE โดย HSPACE เป็นระยะห่างในแนวนอน (left margin และ right margin จะห่างจากรูปต่างกันจะมีหน่วยเป็น Pixel เช่นเดียวกัน

#### 2.8.15 การกำหนดรูปภาพให้เป็น Links (A HREF)

นอกจากการกำหนด Text ให้ Link แล้วเรายังสามารถกำหนดรูปภาพให้ Link ได้ด้วย โดยใช้คำสั่ง <A HREF > คู่กับคำสั่ง <IMG SRC >

#### 2.8.16 การควบคุมข้อความและรูปภาพ

ปกติข้อความกับรูปภาพจะถูกจัดให้สวยงาม โดยข้อความจะอยู่รอบๆ รูป แต่ในกรณีที่มีรูปมีขนาดใหญ่ แต่มีข้อความสำหรับรูปนั้นเพียงเล็กน้อย การจะกระโดดไปยังส่วนอื่น ๆ ของเอกสารจะทำได้ลำบาก เนื่องจาก <BR> เพื่อเพิ่มบรรทัดว่างไปจนกว่าจะพ้นบริเวณของรูปภาพ

ในลักษณะเช่นนี้จะใช้ตัวเลือก CLEAR มาใส่เพิ่มใน <BR> ได้แก่

<BR CLEAR=LEFT> ให้ข้อความที่ตามหลังมาไปเริ่มต้นที่ LEFT Margin

<BR CLEAR=RIGHT> ให้ข้อความที่ตามหลังมาไปเริ่มต้นที่ Right Margin เลย

#### 2.8.17 การสร้าง LINK

##### 2.8.17.1 ความหมายและการทำงานของ Links

Link เป็นการเชื่อมโยงจากจุดหนึ่งของเอกสารในปัจจุบัน ไปยังอีกจุดหนึ่งของเอกสารเดียวกัน หรือคนละเอกสารก็ได้ ซึ่งการ Link อาจจะเป็นการ Link กันระหว่าง Text กับรูปภาพก็ได้ จุดที่ Link กันนี้ เราเรียกว่า “anchor” ใช้คำสั่ง <A> โดยจะมีตัวเลือก “HREF= “ อยู่ร่วมด้วย เช่น

This <A HREF=“ docex1.htm”>text</a> is linked

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้

This text is linked

### 2.8.17.2 การ Link ไปยัง File อื่น

การระบุการ Link ไปยังสถานที่อื่น ๆ ที่ไม่ใช่ในระบบของเรา จะต้องระบุรูปแบบของ URL ให้ถูกต้อง แต่ถ้า Link ไปยังเอกสารที่อยู่ภายในระบบของเราเอง ให้ทำในรูปแบบดังนี้

- อยู่ใน Directory เดียวกัน ให้ใส่เฉพาะชื่อ ไฟล์
- อยู่ใน Subdirectory ของ directory ปัจจุบัน เช่น ถ้าปัจจุบันอยู่ที่ me/homepage/rcipes.htm และต้องการ Link ไปยัง me/homepage/cook/cookie.htm ให้กำหนดเป็น

```
<A HREF="cook/cookie.htm">
```

- อยู่ใน Directory ที่สูงกว่า Directory ปัจจุบัน เช่น ปัจจุบันอยู่ที่ me/homepage/recits.htm ต้องการ Link ไปที่ tetra

```
<A HREF="../tetra.htm">
```

### 2.8.17.3 การ Link ไปยังส่วนอื่นภายในเอกสารเดียวกัน

บางครั้งเราอาจต้องการ LINK ไปยังตำแหน่งอื่น ๆ แต่ยังคงอยู่ในเอกสารเดียวกัน ให้กำหนด Label หรือชื่อให้กับตำแหน่งที่ต้องการจะ Link โดยนำตัวเลือก Name มาใช้ร่วมกับคำสั่ง <A>

### 2.8.17.4 การ Link ไปยังส่วนใดๆ ใน File อื่น

ถ้าเป็นการ Link ภายในเอกสารเดียวกัน ก็เพียงแต่ระบุชื่อ (Name) ของส่วนนั้นลงไป แต่ถ้าเป็นเอกสารอื่น เราจะต้องระบุชื่อเอกสารลงไปด้วย พร้อมทั้งอย่าลืมตั้งชื่อ Label ให้กับส่วนของเอกสารนั้น ๆ ด้วย

### 2.8.17.5 การ Link ไปยังบริการ อื่น ๆ บนอินเทอร์เน็ต

การ Link ไปยัง FTP สามารถ Link ไปยัง ftp เพื่อขอใช้บริการการ โอนข้อมูลได้ โดยการระบุ Address ของ ftp ที่ต้องการลงไป เช่นต้องการ Link ไปที่

```
ftp://ftp4.netscape.com/pub/smart/SM10R2.EXE ให้กำหนดรูปแบบดังนี้
```

```
<A HREF=ftp://ftp4.netscape.com/pub/smart/SM10R2.EXE>
```

```
Netscape's Smartmarks File </A>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์ก็คือ จะแสดงข้อความ Netscape's Smartmarks File เป็นแถบสว่างให้ ถ้าเราแสดงที่ข้อความ โปรแกรมก็จะ Link ไปที่ site ตามที่ระบุทันที ถ้าเรามีสิทธิ์เข้าไปใช้

- การ Link ไปยัง Gophers

สามารถ Link เพื่อไปขอใช้บริการ Gophers ได้เช่นกัน โดยจะต้องระบุ URL ให้ถูกต้องพร้อมทั้งการ กำหนด Port number's Gopher เช่น

<A HREF="gopher://gopher.stolaf.edu:70/00/Internet%20Resource/US-State-Department-

## 2.8.18 การตกแต่งเอกสาร

### 2.8.18.1 ตัวอักษรพิเศษ

มีตัวอักษรพิเศษที่ไม่มีบน Keyboard หรือใช้เป็น tag ในคำสั่ง HTML ไปแล้ว เช่น <> แต่ถ้า ต้องการพิมพ์ใส่ไว้เอกสาร HTML ก็ยังสามารถทำได้ โดยกำหนดรูปแบบต่อไปนี่

### 2.8.18.2 การใส่หมายเหตุ (Comment)

ใช้ในการอธิบายบางส่วนของ HTML ที่เราเขียนมา เพื่อจะได้สะดวกในการติดตามแก้ไขในภายหลัง Comment นี้จะไม่ถูกแสดงบน Browser รูปแบบคำสั่งจะแตกต่างจาก HTML ทั่วไปคือ <!-- และปิด ด้วย -->

### 2.8.18.3 การวางตำแหน่งข้อความ

สามารถจัดวางข้อความให้ชิดด้านใดด้านหนึ่งของบรรทัด โดยใช้คำสั่ง (Alignment) <ALIGN ตัวเลือก> โดยตัวเลือกอาจเป็น Left, Right หรือ center ก็ได้

### 2.8.18.4 ขึ้นบรรทัดใหม่ (Line Break)

ใช้คำสั่ง <BR> โดยสามารถใช้คำสั่ง ALIGN มาช่วยได้

### 2.4.18.5 ขึ้นย่อหน้าใหม่ (Paragraph break)

ใช้คำสั่ง <P> และใช้คู่กับ ALIGN ได้ด้วย เช่น <P ALIGN CENTER>

### 2.8.18.6 อยู่กลางบรรทัด (Center)

การจัดข้อความให้อยู่กลางบรรทัด นอกจากจะใช้คำสั่ง <ALIGN CENTER> แล้วอาจใช้คำสั่ง < CENTER> และ </CENTER> ก็ได้ แต่คำสั่งนี้จะใช้ไม่ได้กับทุก Browser ดังนั้นควรตรวจสอบให้แน่ใจด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.8.18.7 สร้างเส้นตรงแนวนอน (Horizontal Rule)

ใช้คำสั่ง <HR>ร่วมกับตัวเลือกต่อไปนี้

- NOSHADE กำหนดให้เป็นเส้นทึบดำ เช่น
- SIZE กำหนดความหนาให้กับเส้น โดยมีหน่วยเป็น Pixel เช่น  
<HR SIZE=10>
- WIDTH กำหนดความยาวของเส้น
- ALIGN กำหนดตำแหน่งของเส้นในแนวนอนในกรณีที่เส้นตรงมีความยาวไม่เต็มจอภาพ กำหนดเป็น Left, Right หรือ center เช่น  
<HR WIDTH=50% ALIGN=CENTER>หมายถึง เป็นเส้นตรงที่มีความยาวเป็นครึ่งหนึ่งของจอภาพ และวางที่กลางบรรทัดพอดี

### 2.8.19 การจัดรูปแบบของตัวอักษร (Styles)

รูปแบบของตัวอักษรมี 2 แบบ คือ

- Physical Styles เป็นรูปแบบที่ Browser บางตัวอาจไม่รู้จัก ทำให้แสดงออกไม่ได้
- Logical Styles จะถูกแปลงให้ไปใช้ตามรูปแบบของ Browser ที่เรียกมาใช้งานทำให้ใช้งานได้ทุก Browser

### 2.8.20 การกำหนดขนาดและสีของตัวอักษร

กำหนดขนาดของตัวอักษร โดยใช้คำสั่ง <FONT SIZE=n> โดย n แทนตัวเลขตั้งแต่ 1 ถึง 7  
1 คือ ขนาดเล็กที่สุด

7 คือ ขนาดใหญ่ที่สุด ปิดท้ายคำสั่งด้วย </FONT> เพื่อให้กลับ ไปใช้ FONT ขนาดปกติ เช่น  
<FONT SIZE=7>some important text</FONT>

กำหนดขนาดคำตอบปกติด้วยคำสั่ง <BASEFONT SIZE=3>

อีกคำสั่งหนึ่งที่ใช้กำหนดขนาดของตัวอักษร ได้คือ <FONT SIZE+n> โดย n แทนตัวเลข 1 ถึง 7 เป็นการปรับขนาดของตัวอักษร ให้เพิ่มขึ้น + หรือลดลง - เป็นจำนวนเท่าใด เมื่อเทียบกับขนาดตัวอักษรปกติที่ใช้ในขณะนั้น แต่ทั้งนี้และทั้งนี้ จะเพิ่มหรือลดได้ไม่เกินจากขนาดต่ำสุดและสูงสุดของ Font ปกติ คือ 1 ถึง 7 เช่น

This is a <Font size =+1>very</FONT>important issue to US.

กำหนดสีของตัวอักษร ได้ด้วยคำสั่ง <FONT COLOR = สี>ปิดท้ายด้วยคำสั่ง </FONT> โดยขณะนี้มีตัวเลือกคือ `ague`, `back`, `fuchsia`, `gray`, `green`, `line`, `maroon`, `navy`, `olive`, `purple`, `red`, `silver`, `teal`, `white` และ `yellow`

## 2.9 ลินุกซ์ (Linux)

ลินุกซ์ระบบปฏิบัติการแบบ 32 บิต ที่เป็นยูนิกซ์โคลน สำหรับเครื่องพีซี และแจกจ่ายให้ใช้ฟรี สนับสนุนการใช้งานแบบหลากหลายงาน หลายผู้ใช้ (MultiUser-MultiTasking) มีระบบ X วินโดวส์ ซึ่งเป็นระบบการติดต่อผู้ใช้แบบกราฟฟิก ที่ไม่ขึ้นกับโอเอสหรือฮาร์ดแวร์ใดๆ (มักใช้กันมากในระบบยูนิกซ์) และมาตรฐานการสื่อสาร TCP/IP ที่ใช้เป็นมาตรฐานการสื่อสารในอินเทอร์เน็ตมาให้ในตัว

ลินุกซ์มีความเข้ากันได้ (compatible) กับ มาตรฐาน POSIX ซึ่งเป็นมาตรฐานอินเทอร์เน็ตที่ระบบยูนิกซ์ส่วนใหญ่จะต้องมีและมีรูปแบบบางส่วนที่คล้ายกับระบบปฏิบัติการยูนิกซ์จากค่าย Berkeley และ System V

โดยความหมายทางเทคนิคแล้วลินุกซ์ เป็นเพียงเคอร์เนล (kernel) ของระบบปฏิบัติการ ซึ่งจะทำหน้าที่ในด้านของการจัดสรรและบริหาร โพรเซสงาน การจัดการไฟล์และอุปกรณ์ I/O ต่างๆ แต่ผู้ใช้ทั่วไปจะรู้จักลินุกซ์ผ่านทางแอฟพลิเคชันและระบบอินเทอร์เน็ตที่เขาเหล่านั้นเห็น (เช่น Shell หรือ X วินโดวส์)

ถ้ารันลินุกซ์บนเครื่อง 386 หรือ 486 มันจะเปลี่ยนพีซีให้กลายเป็นยูนิกซ์เวิร์กสเตชันที่มีความสามารถสูง เคยมีผู้เทียบประสิทธิภาพระหว่างลินุกซ์บนเครื่องเพนเทียม และเครื่องเวิร์กสเตชันของซันในระดับกลาง และได้ผลออกมาว่าให้ประสิทธิภาพที่ใกล้เคียงกัน และนอกจากแพลตฟอร์มอินเทลแล้ว ปัจจุบันลินุกซ์ยังได้ทำการพัฒนาระบบเพื่อให้สามารถใช้งานได้บนแพลตฟอร์มอื่นๆ ด้วย เช่น DEC Alpha , Motorola Power-PC , MIPS เมื่อสร้างแอฟพลิเคชันขึ้นมาบนแพลตฟอร์มใดแพลตฟอร์มหนึ่งแล้วสามารถย้ายแอฟพลิเคชันของไปวิ่งบนแพลตฟอร์มอื่นได้ไม่ยาก

เนื่องจากเป็นระบบปฏิบัติการที่ฟรี และมีผู้นิยมใช้มาก ทำให้มีผู้นำลินุกซ์ไปแก้ไขให้สามารถใช้งานได้บนตัวประมวลผลกลางหลากหลายตั้งแต่อินเทล, โมโตโรลา, ดิจิตอลอัลฟา, พาวเวอร์พีซี, ไปจนถึงสปาร์คของซัน นอกจากนี้ยังมีผู้พัฒนาโปรแกรมประยุกต์ออกมาอีกมากมาย และมีประสิทธิภาพและมีคุณภาพสูง ลินุกซ์เป็นระบบปฏิบัติการ 32 บิตเต็มรูปแบบ ซึ่งสามารถจะดึงเอาพลังของเครื่องคอมพิวเตอร์ออกมาได้อย่างเต็มกำลัง มีคุณลักษณะของระบบ UNIX เต็มรูปแบบ และเป็นระบบหลายผู้ใช้ หลายงานอย่าง แท้จริง ลินุกซ์มีระบบอินเทอร์เน็ตแบบกราฟฟิกที่เรียกกันว่า X Windows ซึ่งเป็น มาตรฐานของระบบยูนิกซ์ทั่วไป และสามารถ ใช้ window manager ได้หลายชนิด ตามความต้องการ นอกจากนี้ยังสนับสนุน โพรโตคอลแบบ TCP/IP , SLIP, PPP, UUCP และอื่นๆ สามารถหาข้อมูลเพิ่มเติมได้ง่าย

### บทที่ 3

#### การสร้างและการพัฒนาโปรแกรม

##### 3.1 ศึกษาและวิเคราะห์วางแผนการศึกษา

เริ่มจากศึกษาการทำงานของระบบฐานข้อมูลแบบรีเลชันเพื่อสามารถนำไปใช้ในการออกแบบตารางฐานข้อมูล และออกแบบความสัมพันธ์ระหว่างข้อมูล ในเรื่องต่างๆ ต่อไปนี้

- หลักการในการจัดเก็บข้อมูลที่มีความสัมพันธ์กันระหว่างข้อมูลในลักษณะต่างๆ
- หลักการในการออกแบบฐานข้อมูลให้มีความถูกต้องและซ้ำซ้อนน้อยที่สุด
- ศึกษาหลักการของระบบไคลต์เซิร์ฟเวอร์ (Client Server) ในเรื่องต่างๆ คือ ความสามารถของระบบ ข้อดีและข้อเสียของระบบ แพลตฟอร์ม (Platform) ของระบบและการติดต่อสื่อสารของระบบ
- ศึกษาและทดลองใช้เครื่องมือในการพัฒนาระบบ (Application Development Tools) โดยทำการเปรียบเทียบและเลือกเครื่องมือที่เหมาะสม เพื่อที่จะนำมาประยุกต์ใช้ในโครงการซึ่งในโครงการได้เลือกใช้ My SQL จัดการฐานข้อมูล (DBMS) และใช้ PHP เป็นส่วนเชื่อมต่อกับฐานข้อมูล และเลือกการเว็บเพจ (Web Page) ในส่วนติดต่อกับผู้ใช้งาน

##### 3.2 วัตถุประสงค์ของโครงการ

1. เพื่อเพิ่มประสิทธิภาพของการดำเนินงานของภาควิชา ฯ ให้มีความรวดเร็วและถูกต้องมากยิ่งขึ้น
2. เพื่อให้ผู้ใช้ที่ต้องการทราบข้อมูล สามารถตรวจสอบและรับข้อมูลได้อย่างง่ายดายและรวดเร็ว โดยผ่านทางระบบอินเทอร์เน็ต
3. เพื่อส่งเสริมการพัฒนา Application ในระบบปฏิบัติการ Linux

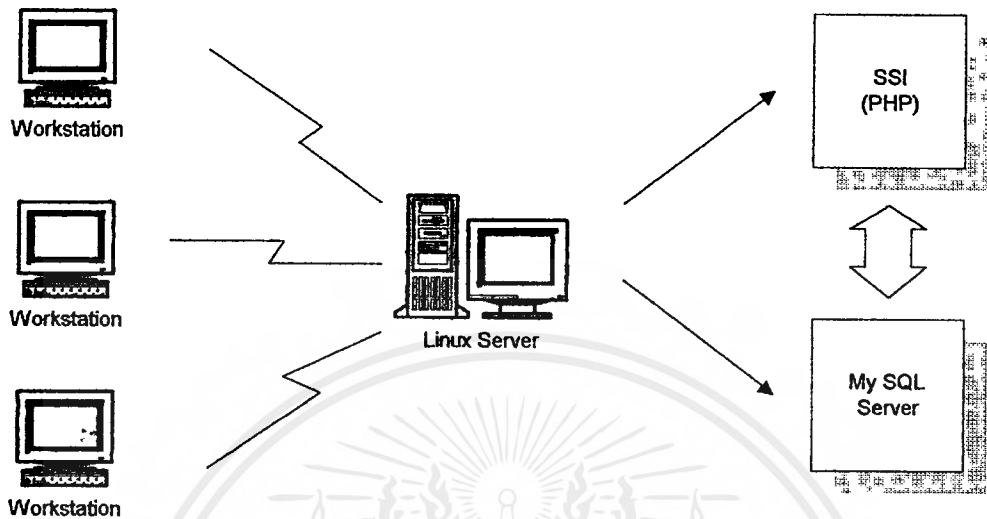
##### 3.3 โครงสร้างของระบบ

โครงสร้างของระบบฐานข้อมูลระบบสารสนเทศของภาควิชาโทรคมนาคมนั้นประกอบไปด้วยส่วนสำคัญดังนี้

1. ผู้ให้บริการฐานข้อมูลและเว็บ (Web & Database Server) เป็นส่วนที่ให้บริการ www และบริการฐานข้อมูล ซึ่งในโครงการนี้ใช้ระบบปฏิบัติการ ลินุกซ์ เรดแฮท (Redhat-Linux) , Web Server ที่ใช้ก็คือ Apache และส่วน Database Server ก็คือ My SQL ส่วน Server Side Include (SSI) ก็คือ PHP
2. ระบบเครือข่ายซึ่งเชื่อมต่อกับระบบเครือข่ายอินเทอร์เน็ต คือต้องสนับสนุนการสื่อสารด้วยโปรโตคอล (Protocol) TCP/IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ผู้ใช้บริการ (Client) จะสามารถสืบค้นฐานข้อมูลโดยใช้เว็บเบราว์เซอร์ (Web Browser) ซึ่งโครงงานนี้ต้องการ



### 3.4 อุปกรณ์ที่ใช้ในระบบฐานข้อมูล

ในโครงงานนี้ประกอบไปด้วยส่วน Server และ Client ดังต่อไปนี้

#### ส่วน Server

- CPU "Intel" Pentium 120 MHz<sub>2</sub>
- RAM 40 MB
- Hard Disk 1 GB
- Ethernet Card NE-2000 Compatible

#### ส่วน Client

- CPU "Intel" Pentium 120 MHz<sub>2</sub>
- RAM 40 MB
- Hard Disk 1 GB
- Ethernet Card NE-2000 Compatible

#### Software

##### ส่วน Server

- Linux-Redhat 5.2
- My SQL
- PHP 3

##### ส่วน Client

เอกสารนี้เป็นเอกสารของ Microsoft Windows 98 ที่ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

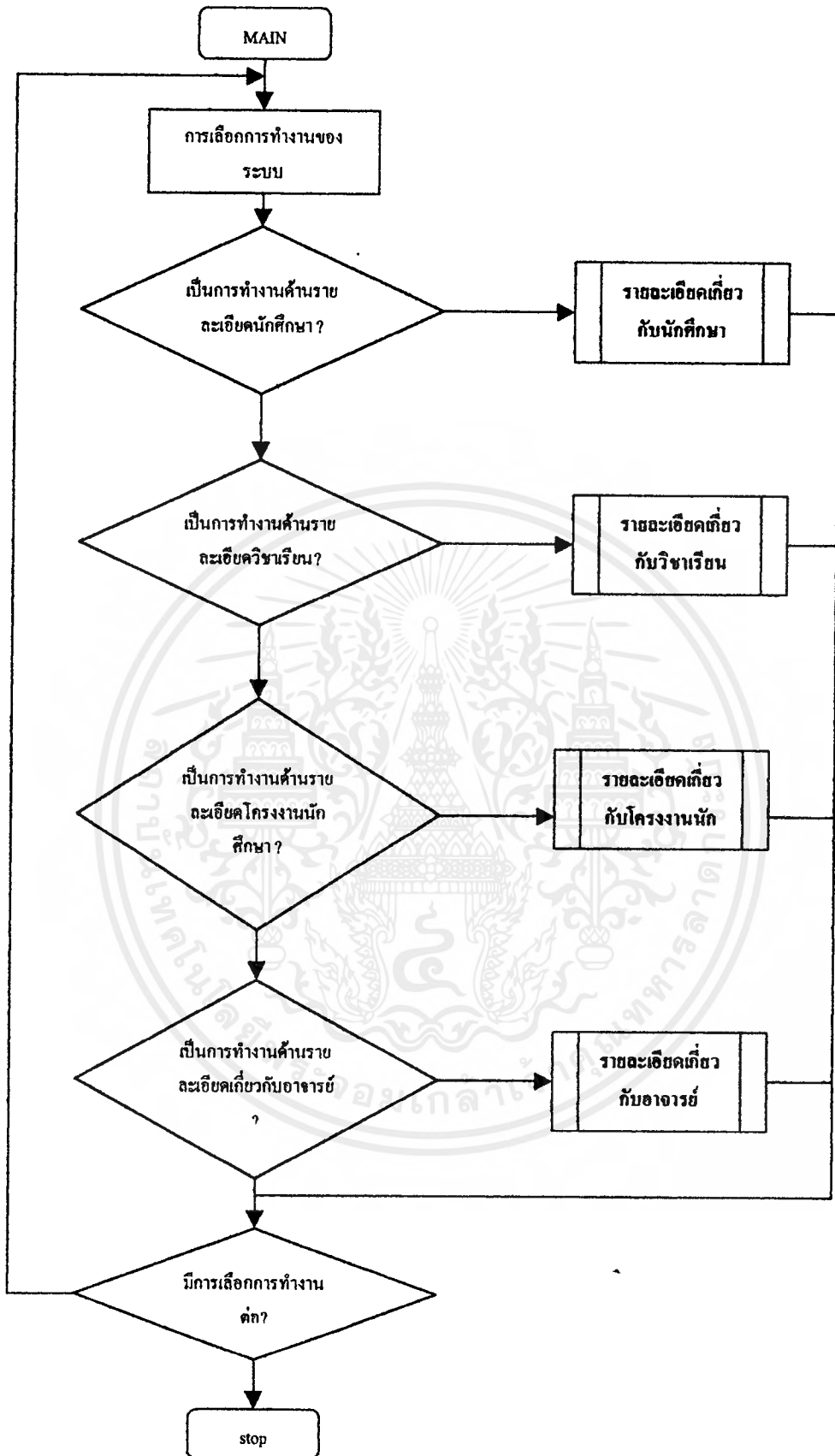
- Microsoft Internet Explorer 4

### 3.5 ขั้นตอนในการพัฒนาโปรแกรม

1. สรุปขอบเขตของงานที่จะทำ
2. ทำการติดตั้งระบบปฏิบัติการ Linux Redhat , ติดตั้ง SSI-PHP และติดตั้ง My SQL (Database Server)
3. ทำการนำเข้าข้อมูลที่เป็นต่อการทดลองของโครงการนี้
4. ทำการออกแบบลักษณะของหน้าจอ (User Interface) และแบบฟอร์มต่างๆ ที่จะนำมาแสดงบนบราวเซอร์ (Browser) โดยยึดหลักการใช้งานง่ายและถูกต้อง
5. สำหรับการสร้างแบบฟอร์มนั้นจะสร้างโดยใช้ภาษา HTML
6. สำหรับส่วน Server จะมีการเขียน PHP Script ขึ้นมาเพื่อทำหน้าที่เป็นตัวกลางในการที่เว็บเบราว์เซอร์ จะทำงานกับฐานข้อมูลที่อยู่บน Server นี้ เพื่ออำนวยความสะดวกในการพัฒนาระบบให้มีประสิทธิภาพยิ่งขึ้น
7. ในการพัฒนาต้องมีการตรวจสอบความถูกต้องของระบบในทุกๆขั้นตอนถ้าพบข้อผิดพลาดก็มีความจำเป็นที่จะต้องแก้ไขข้อผิดพลาดนั้นๆ

### 3.6 การออกแบบระบบและการสร้างระบบฐานข้อมูล

ในโครงการระบบฐานข้อมูลสารสนเทศ มีการออกแบบโครงสร้างผังไถ่ลวชาร์ต (Flow Chart) ของโปรแกรมของระบบดังต่อไปนี้

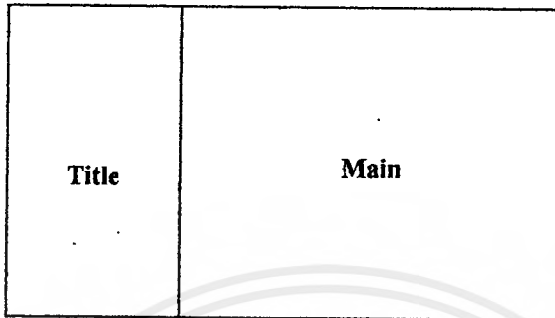


รูปที่ 3.1 แสดงโฟลว์ชาร์ตแสดงการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.6.1 ส่วนติดต่อกับผู้ใช้ (User Interface Part)

ในส่วน Client โครงการนี้ได้ออกแบบการปรับปรุงใช้งานกับ Graphic User Interface ในรูปของ Web page ดังแสดงดังรูปที่ 3.2



รูปที่ 3.2 แสดงการออกแบบหน้าจอส่วน Client

โดยที่ทำการสร้าง Web Page โดยใช้ภาษา HTML โดยที่ภาษา HTML มีการเรียกใช้ PHP Script โดยจะทำการส่งผ่านตัวแปรไปให้ PHP เพื่อใช้ประมวลผลต่อไปดังต่อไปนี้

```
<FORM ACTION= script.php3 method="post">
```

ส่วนภาษา HTML ที่ใช้ในโครงการนี้ได้แสดงในภาคผนวก โดยไฟล์ต่างๆ ที่ใช้งานในระบบฐานข้อมูลมีดังนี้ (หมายเหตุ XXXXX คือ ชื่อชั้นของนักศึกษา เช่น 3T\_R1)

- index.html เป็นเว็บเพจ (Web Page) หลักของระบบ ซึ่งทำการแบ่งหน้าจอออกเป็น 2 เฟรม คือ เฟรม title และเฟรม main
- title.html เป็นส่วนแสดงหัวข้อให้เลือกว่าจะใช้บริการฐานข้อมูลแบบใด
- stdXXXXX\_main.html เป็นส่วนให้ป้อนข้อมูลรายชื่อนักศึกษาในรูปแบบฟอร์ม แล้วส่งค่าที่รับมาไปให้กับ searchXXXXX\_name.php3 และ searchXXXXX\_id.php3 เพื่อค้นหารายชื่อนักศึกษา
- studyXXXXX\_main.html เป็นส่วนเลือกภาคเรียนที่ต้องการของนักศึกษานี้ๆ เพื่อดูวิชาเรียน ซึ่งจะทำการลิงค์ไปยัง searchXXXXX\_studytermY.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- searchXXXXX\_studytermY.html เป็นส่วนให้ป้อนข้อมูลรายชื่อนักศึกษาในรูปแบบฟอร์ม แล้วส่งค่าที่รับมาไปให้ กับ searchXXXXX\_studyXXXXX\_termY.php3 และ searchXXXXX\_studyXXXXX\_id.php3 เพื่อค้นหาวิชาเรียนของนักศึกษา
- projZZZZ\_main.html เป็นส่วนให้ป้อนข้อมูลรายชื่อโครงการของนักศึกษาในรูปแบบฟอร์ม แล้วส่งค่าที่รับมาไปให้กับ searchZZZZ\_proj\_projname.php3 , searchZZZZ\_proj\_stdname.php3 , searchZZZZ\_proj\_id.php3 , searchZZZZ\_proj\_tcname.php3 เพื่อค้นหารายชื่อโครงการของนักศึกษา (ZZZZ = ปีการศึกษา, Y = ภาคเรียน)
- teacher\_main.html เป็นส่วนให้ป้อนข้อมูลรายชื่ออาจารย์ในรูปแบบฟอร์ม แล้วส่งค่าที่รับมาไปให้กับ searchtcr\_name.php3 และ searchtcr\_room.php3 เพื่อค้นหารายชื่ออาจารย์
- stdZZZZ\_main.html เป็นส่วนให้ป้อนข้อมูลรายชื่อนักศึกษาในรูปแบบฟอร์ม แล้วส่งค่าที่รับมาไปให้กับ searchZZZZ\_name.php3 และ searchZZZZ\_id.php3 เพื่อค้นหาชื่อ นักศึกษา

### 3.6.2 ส่วนการจัดการข้อมูล (Database Part)

จากการวิเคราะห์และออกแบบระบบเพื่อหาความสัมพันธ์ของข้อมูลในด้านต่างๆ ซึ่งความสามารถเขียนออกมาเป็นตารางได้ดังนี้

id	studen_name	class_year
int (8)	char(30)	char(2)

ตารางที่ 3.1 แสดงตาราง StudentXXXX

id	รหัสนักศึกษา
student_name	ชื่อนามสกุล ของนักศึกษา
class_year	ชั้นปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Id	Project_name	teacher_name	Project_year
Int(8)	Char(50)	char(30)	int(4)

ตารางที่ 3.2 แสดงตาราง projectXXXX

id                      รหัสนักศึกษา  
project\_name        ชื่อโครงการ  
teacher\_name        ชื่ออาจารย์ที่ปรึกษา  
project\_year        ปีการศึกษา

study_code	study_name	study_credit	study_daytime	syudy_room	study_teacher
int(8)	char(50)	int(1)	char(35)	char(20)	char(30)

ตารางที่ 3.3 แสดงตาราง studyXXXX

study\_code        รหัสวิชา  
study\_name        ชื่อวิชา  
study\_credit       หน่วยกิต  
study\_daytime    วัน-เวลาที่เรียน  
study\_room        ห้องเรียน  
study\_teacher     ชื่ออาจารย์ที่สอน

teacher_name	teacher_room
char(30)	char(5)

ตารางที่ 3.4 แสดงตาราง teacherXXXX

teacher\_name      ชื่ออาจารย์  
teacher\_room        ห้องทำงานอาจารย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทำการสร้างตารางในฐานข้อมูลโดยใช้คำสั่งจากภาษา SQL ดังต่อไปนี้

### คำสั่งสร้าง studentXXXX

```
#
# Table structure for table 'studentXXXX'
#
```

```
CREATE TABLE studentXXXX (
  id int(8) DEFAULT '0' NOT NULL,
  student_name char(30) NOT NULL,
  class_year char(2) NOT NULL,
  PRIMARY KEY (id),
  KEY id (id),
  UNIQUE id_2 (id)
);
```

### คำสั่งสร้างตาราง projectZZZZ

```
#
# Table structure for table 'projectZZZZ'
#
```

```
CREATE TABLE project2541 (
  id int(8) DEFAULT '0' NOT NULL,
  project_name char(100) NOT NULL,
  teacher_name char(30),
  project_year int(4) DEFAULT '0' NOT NULL,
  PRIMARY KEY (id),
  KEY id (id, project_name),
  UNIQUE id_2 (id)
);
```

### คำสั่งสร้างตาราง studyXXXX

```
#
# Table structure for table 'studyXXXX'
#
```

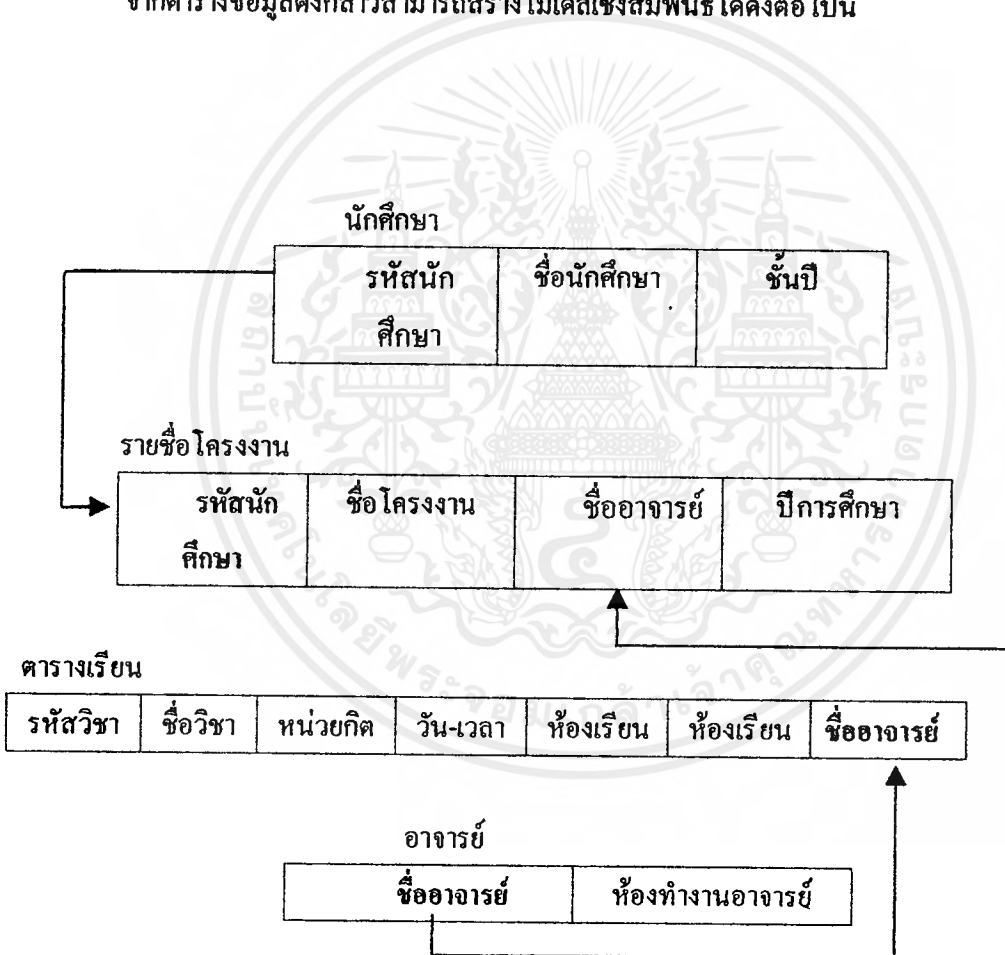
```
CREATE TABLE studyXXXX (
  study_code int(8) DEFAULT '0' NOT NULL,
  study_name char(50) NOT NULL,
  study_credit int(1),
  study_daytime char(35),
  study_room char(20),
  study_teacher char(30),
  PRIMARY KEY (study_code),
  KEY study_code (study_code),
  UNIQUE study_code_2 (study_code)
);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### คำสั่งสร้างตาราง teacher

```
#
# Table structure for table 'teacher'
#
CREATE TABLE teacher (
  teacher_name char(30) NOT NULL,
  teacher_room char(5),
  PRIMARY KEY (teacher_name),
  KEY teacher_name (teacher_name),
  UNIQUE teacher_name_2 (teacher_name)
);
```

จากตารางข้อมูลดังกล่าวสามารถสร้างโมเดลเชิงสัมพันธ์ได้ดังต่อไปนี้



รูปที่ 3.3 แสดง โมเดลเชิงสัมพันธ์ของระบบฐานข้อมูลสารสนเทศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.6.2.1 ส่วนการเชื่อมโยงกันของ PHP และ My SQL Database

เมื่อเราได้ทำการสร้างฐานข้อมูล ซึ่งในโครงการนี้ให้ชื่อว่า project และได้ทำการป้อนข้อมูลลงไปเรียบร้อยแล้ว ต่อไปจะเป็นการเขียนโปรแกรมเพื่อให้ PHP ทำการติดต่อกับ MySQL Database ได้

#### - Interface ของ MySQL database

ในการที่จะแสดงฟังก์ชันเหล่านี้ MySQL จะต้องถูกติดตั้งใน server ก่อน

ในระดับพื้นฐาน PHP จะทำหน้าที่เสมือนตัวกลางระหว่าง database กับ เว็บเบราว์เซอร์ เพื่อติดต่อสื่อสารคำสั่งจากที่หนึ่งไปยังอีกที่หนึ่ง การสื่อสารโดยวิธีนี้จะทำให้การติดต่อเป็นแบบ interactive มากขึ้น เช่น การตอบแบบสอบถาม ระบบตอบรับแบบ online หรือ งานอื่นๆ ที่ต้องให้ผู้ใช้ใส่ข้อมูลเข้าไป การติดต่อแบบสองทางนี้จะเริ่มด้วยการติดต่อไปยัง MySQL database โดยการติดต่อนี้จะใช้คำสั่งต่อไปนี้

Syntax: int mysql\_connect(string hostname, string username, string password);

Hostname : เป็น hostname ของ database (เช่น ของ suvepon1 ก็จะเป็น [suvepon1.telecom.eng.kmitl.ac.th](http://suvepon1.telecom.eng.kmitl.ac.th))

Username : เป็น username ที่จะเข้าสู่ระบบ MySQL database

Password : เป็น password ที่จะเข้าสู่ระบบ MySQL database

ถ้าตอบกลับมาเป็นบวกแสดงว่าการติดต่อ สำเร็จเสร็จสมบูรณ์ แต่ถ้าเป็นลบ แสดงว่าไม่สำเร็จ คำสั่งทุกคำสั่งต้องอยู่ภายใต้ "<" และ ">"

ลองกลับไปยังงานของผู้ใช้ สมมุติว่าจะสร้าง ตารางภายใต้ MySQL database

```
mysql> CREATE TABLE information (
```

```
> name VARCHAR (25),
```

```
> email VARCHAR (25),
```

```
> choice VARCHAR (8) );
```

ขณะนี้ สมมุติว่าเราต้องการที่จะใส่ข้อมูลของผู้ใช้ลงใน MySQL database ซึ่งสามารถทำได้โดยเปลี่ยนไฟล์ email.php3 เป็นดังต่อไปนี้

```
<?
```

```
/* this script will handle the variables passed from the moreinfo.html file */
```

```
/* declare some relevant variables */
```

```
$hostname = "devshed";
```

```
$username = "myusername";
```

```
$password = "mypassword";
```

```
$dbName = "mydbName";
```

```
/* MySQL table created to store the data */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

$userstable = "information";

/* the site administrator's email address */
$adminaddress = "administration@buycorn.com";

/* make connection to database */

MYSQL_CONNECT($hostname,$username,$password) OR DIE("Unable to connect to my
database");

@mysql_select_db("$dbName") or die("Unable to select database");

PRINT "<CENTER>";

PRINT "Hello, $name.";

PRINT "<BR><BR>";

PRINT "Thank you for your interest.<BR><BR>";

PRINT "We will send information to $email, and have noted that you like $preference";

PRINT "</CENTER><BR><BR>";

/* Send relevant emails */

mail("$email", "Your request for information",
"$name\nThank you for your interest!\n
We sell fresh corn daily over the Internet!
Place your order at http://www.buycorn.com,
and receive a free package of $preference!");

mail("$adminaddress",
"Visitor request for info.",
"$name requested for information.\n
The email address is $email. \n The visitor prefers $preference.");

/* Insert information into table */

$query = "INSERT INTO $userstable VALUES('$name','$email', '$preference')";
$result = MYSQL_QUERY($query);

PRINT "Your information has also been inserted into our database, for future reference.";

/* Close the database connection */

MYSQL_CLOSE();

?>

```

### หมายเหตุ

1. ตัวแปรที่ประกาศไว้ตอนต้นของ script ไว้สำหรับฟังก์ชัน MYSQL\_CONNECT เราสามารถแทนค่าต่างๆ ในฟังก์ชันได้ อย่างไรก็ตาม ถ้างานของเรามีขนาดใหญ่ขึ้น ค่าต่างๆ เหล่านี้สามารถแทนให้เอกสารนี้เป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อยู่ในไฟล์ต่างหากได้ และรวมเข้าไปอยู่ในไฟล์(ใช้คำสั่ง #include) จะทำให้ง่ายต่อการเปลี่ยนแปลงข้อมูลที่ต้องการจะแก้ไข

2. ฟังก์ชัน @mysql\_select\_db จะเลือก database เพื่อทำการบันทึกลงใน script และยอมให้ผู้ใช้สามารถเข้าถึง statement ได้โดยไม่ต้องใส่ชื่อ database

Syntax: int mysql\_select\_db(string database\_name, int link\_identifier);

\*database\_name จะต้องเป็นชื่อของ database ซึ่งปรากฏอยู่ใน server เท่านั้น

\*link\_identifier (optional) หมายถึง การลิงค์ไปยัง database เป็นเหตุผลในการเชื่อมต่อ

ถ้าผู้ใช้ไม่ได้ระบุตำแหน่ง ลิงค์ที่เปิดไว้เป็นครั้งสุดท้ายจะถูกใช้

\*การให้ผลลัพธ์ true/false ขึ้นอยู่กับว่าการส่งข้อมูลสำเร็จหรือไม่

3. ฟังก์ชัน MYSQL\_QUERY ใช้สำหรับส่งคำถามไปยัง MySQL database

Syntax: int mysql\_query(string query, int link\_identifier);

\*query – คือคำถามที่เราต้องการส่งไปยัง database

\*link\_identifier – เป็นชื่อของ database (ให้เลือก แต่ถ้าไม่มี จะใช้ database ที่เปิดบ้อยที่สุด) หรือถ้าผู้ใช้เลือกว่าจะไม่ใช่ฟังก์ชัน @mysql\_select\_db ผู้ใช้จะต้องเป็นผู้ใส่ชื่อของ database เอง

\*ผลลัพธ์จะออกมาเป็นจำนวนเต็ม ถ้าเป็นบวกแสดงว่าสำเร็จ ถ้าเป็นลบแสดงว่าไม่สำเร็จ

4. สำหรับคำสั่ง MYSQL\_CLOSE จะเป็นการปิดการติดต่อกับ MySQL database ด้วย syntax ต่อไปนี้ โดย link\_identifier หมายถึงชื่อของ database ถ้าไม่กำหนดมันจะปิด database ที่เปิดไว้ล่าสุด

Syntax: int mysql\_close(int link\_identifier);

\*link\_identifier – เหมือนข้างต้น

\*ผลลัพธ์เป็นจำนวนเต็ม ถ้าเป็นบวกแสดงว่าสำเร็จ ถ้าเป็นลบแสดงว่าไม่สำเร็จ

### การแสดงผล output ของ MySQL

หลังจากที่เราได้เข้าถึงข้อมูลของผู้ใช้แล้ว และขณะนี้ข้อมูลได้ถูกเก็บไว้ใน database แล้วนั้น ก็มาถึงขั้นตอนที่เราจะต้องรู้ว่าเราจะหาข้อมูลนั้นได้อย่างไรและแสดงผลออกมาได้อย่างไรซึ่งมี 3 ขั้นตอนดังนี้

1. ต้องการข้อมูลที่เหมาะสมกับข้อจำกัดของคำถาม
2. ต้องมีการตรวจสอบข้อจำกัดที่เหมาะสม
3. เอาที่พูดของข้อจำกัดที่เหมาะสม

เราจะได้เอาที่พูดไปยังบราวเซอร์ถึงผู้ใช้ทั้งหมดที่ชอบแอปเปิ้ลมากกว่าส้ม

<?

```
/* script to output to screen all users preferring Apples to Oranges */
```

```
/* declare some relevant variables */
```

```
$hostname = "devshed";
```

```
$username = "myusername";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

$password = "mypassword";
$userstable = "information";
$dbName = "mydbname";
/* make connection to database */
MYSQL_CONNECT($hostname, $username, $password) OR DIE("Unable to connect to n
database");

@mysql_select_db( "$dbName") or die( "Unable to select database");
/* Select all users with the preference Apples */
$query = "SELECT * FROM $userstable WHERE choice = 'Apples'";
$result = MYSQL_QUERY($query);
/* How many of these users are there? */
$number = MYSQL_NUMROWS($result);
/* Print these results to the screen in a nice format */
$i = 0;
IF ($number == 0) :
    PRINT "<CENTER><P>Nobody in the database prefers Apples!</CENTER>";
ELSEIF ($number > 0) :
    PRINT "<CENTER><P>Users preferring Apples: $number<BR><BR>";
    WHILE ($i < $number):
        $name = mysql_result($result,$i,"name");
        $email = mysql_result($result,$i,"email");
        PRINT "Visitor $name likes Apples.<BR>";
        PRINT "Email address: $email.";
        PRINT "<BR><BR>";
        $i++;
    ENDWHILE;
    PRINT "</CENTER>";
ENDIF;
?>

```

บันทึกข้อมูลข้างต้นเป็น *apples.php3*

คำอธิบายฟังก์ชันใหม่:

```
$number = MYSQL_NUMROWS($result);
```

**Syntax: int mysql\_num\_rows(string result);**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

\*result – เป็น array of records ซึ่งเป็นผลลัพธ์ของฟังก์ชัน MYSQL\_QUERY

\*จะคืนค่าตัวเลขซึ่งเป็นค่าของแถวที่ถูกเก็บไว้ใน \$result

ในที่สุด เราจะได้เอาท์พุทของงานวิจัยชิ้นนี้ ถ้าไม่มีผู้ใดใน database ขอบแอปเปิ้ลเลย เอาท์พุทก็จะปรากฏผลออกมา แต่ถ้ามีผู้ขอบแอปเปิ้ล ก็จะปรากฏผลออกมาเป็นชื่อและอีเมลล์ ซึ่งเอาท์พุทจะเกิดขึ้นผ่านคำสั่ง WHILE

```
$name = MYSQL_RESULT($result,$i,"name");
```

**Syntax: int mysql\_result(int result, int i, column);**

ฟังก์ชัน mysql\_result จะทำการแยกแต่ละ record และจะทำการแทนค่าแต่ละอันลงในตัวแปร

\* \$result หมายถึง array ของคำถาม

\* \$i หมายถึง การนับพิเศษที่เราได้เข้าไป

\* column จะเป็นชื่อของคอลัมน์จริง ๆ ที่อยู่ใน MySQL table ซึ่งอาจจะเป็นค่า offset หรือ เป็นค่าของ filename.fieldtable

ดังนั้น การใช้ WHILE loop เราก็จะสามารถให้ค่าเอาท์พุทของข้อมูลออกมาที่บราวเซอร์ได้ง่ายพอสมควร

### ฟังก์ชันอื่น ๆ

สำหรับการใช้ฟังก์ชัน MYSQL\_QUERY ทางหนึ่งที่จะเข้าถึงหมายเลขของฟังก์ชัน SQL ก็คือการใช้ฟังก์ชัน DELETE และ UPDATE

Delete

สมมติว่าเราต้องการลบ record ทุกอันที่มีคำว่า “Bunny”

```
$query = "DELETE FROM $userstable WHERE name = "Bunny";  
MYSQL_QUERY($query);
```

Update

หรือว่าเราต้องการที่จะเปลี่ยนแปลง record ทุกอันที่มีคำว่า “Bunny”

```
$query = "UPDATE $userstable SET name = "Bugs Bunny" WHERE name = "Bunny";  
MYSQL_QUERY($query);
```

ซึ่งโค้ด (code) ของ PHP ที่ใช้การเชื่อมโยงกับ MySQL Database ที่ใช้ในโครงการนี้ ได้แสดงไว้ในการผนวกแล้ว

### 3.6.3 การโอนย้ายไฟล์ไปยัง Linux Server

เมื่อสร้าง Web page ด้วยภาษา HTML และส่วนตัวเชื่อมโยงกับ MySQL Database ด้วย PHP สำเร็จแล้ว ก็จะทำการ โอนย้าย (transfer) ไฟล์เหล่านี้ไปยัง Linux Server ซึ่งมี domain name คือ surepon1.telecom.eng.kmitl.ac.th และ IP address คือ 161.246.18.206



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### ผลการทดลอง

#### 4.1 การทดลองส่วนหน้าจอหลัก

จากการสร้างและพัฒนาฐานข้อมูลสารสนเทศของภาควิชาฯ ก็ได้มีการทดลองคุณสมบัติในการค้นหาและเข้าถึงฐานข้อมูลที่อยู่ใน Linux Server โดยใช้ Web Browser ในการเข้าถึงข้อมูลผ่านทางเครือข่ายอินเทอร์เน็ต (TCP/IP) graphic mode ป้อน URL ไปยัง <http://suevpon1.kmitl.ac.th/~project/> ก็จะได้ผล



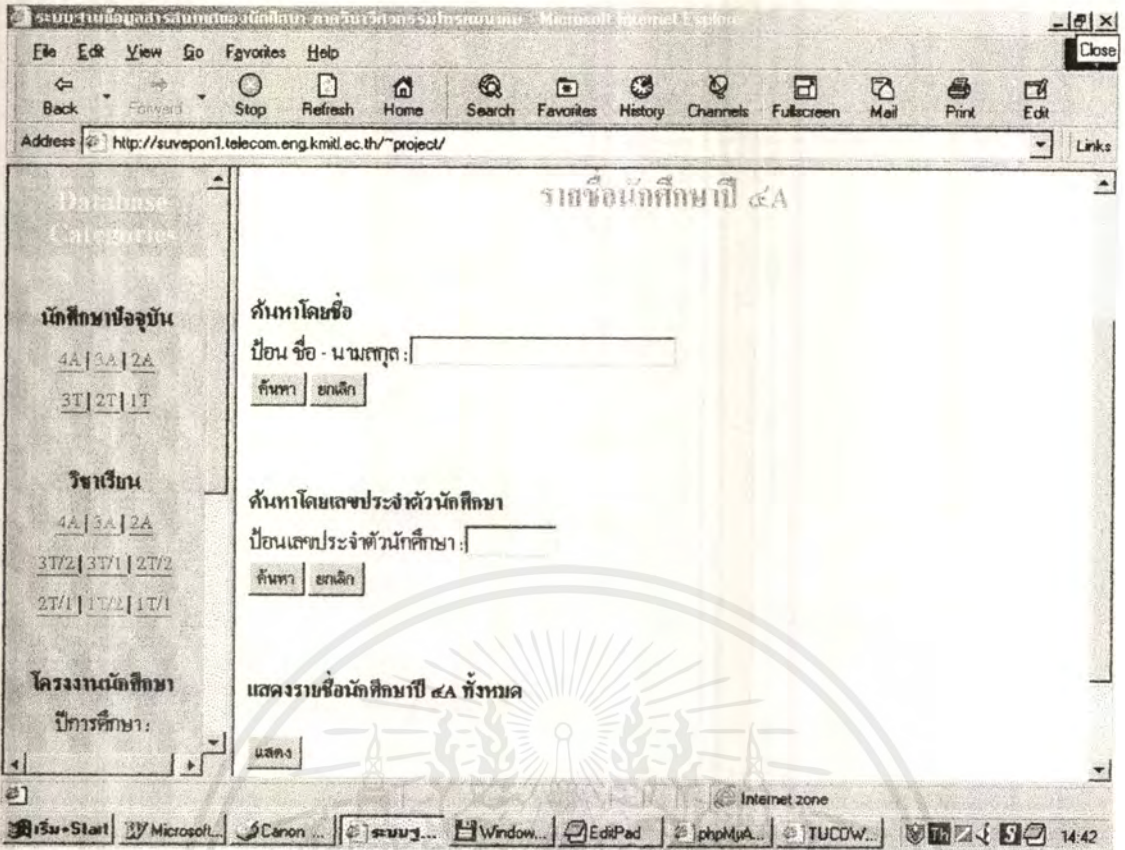
ดังนี้

รูปที่ 4.1 แสดงหน้าจอเมนูหลัก

#### 4.2 การทดลองส่วนการค้นหารายชื่อนักศึกษา

เมื่อเลือกการทำงานส่วนของนักศึกษาปัจจุบันในชั้นปีต่างๆ ตัวโปรแกรมจะแสดงแบบฟอร์มให้ป้อนเพื่อค้นหารายชื่อนักศึกษา โดยสามารถค้นหาได้จากทั้งชื่อ-นามสกุล และค้นหาจากเลขประจำตัวนักศึกษา หรือถ้าหากต้องการแสดงรายชื่อนักศึกษาในชั้นปีนั้นๆ ทั้งหมดก็สามารถเลือกให้แสดงออกมาทั้งหมดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

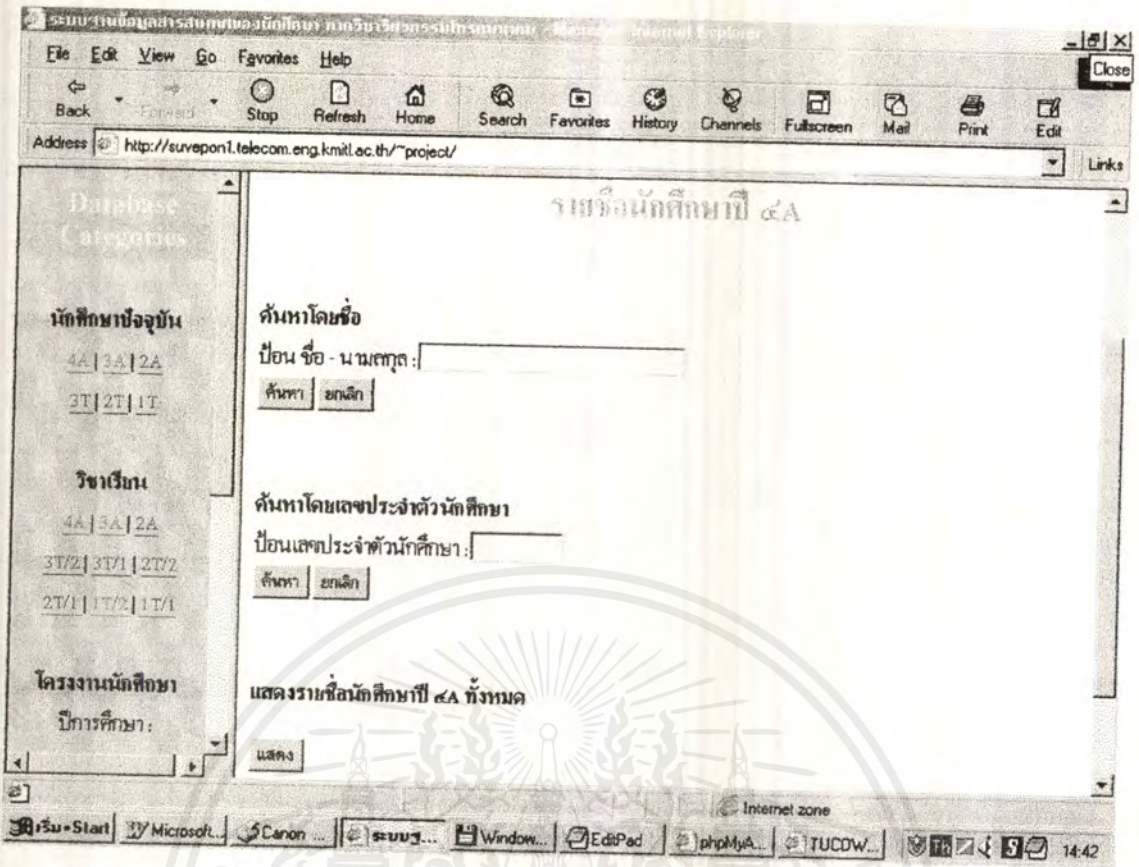


รูปที่ 4.2 แสดงหน้าจอเพื่อรับข้อมูลเพื่อการแสดงข้อมูลรายชื่อนักศึกษา

#### 4.2.1 ค้นหาโดยชื่อ

ในที่นี้จะทำการป้อนชื่อที่ต้องการค้นหาเข้าไปในแบบฟอร์ม และเมื่อส่งค้นหา ก็จะทำให้การค้นหาชื่อที่มีตัวอักษรตรงกับที่ได้รับไป เมื่อค้นหาเสร็จสิ้นแล้วก็จะแสดงผลที่ออกมาในรูปแบบของตาราง โดยจะแสดงทั้งชื่อนักศึกษาและเลขประจำตัวนักศึกษา ในที่นี้เมื่อป้อนคำว่า “ณัฐ” ไปในส่วนชื่อนักศึกษาปี 4A แล้ว โปรแกรมก็จะแสดงผลที่ออกมา 3 ชื่อที่ตรงคือ “ณัฐพงศ์”, “ณัฐพงษ์” และ “ณัฐวุฒิ” ดังแสดงในรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงหน้าจอเพื่อรับข้อมูลเพื่อการแสดงข้อมูลรายชื่อนักศึกษา

#### 4.2.1 ค้นหาโดยชื่อ

ในที่นี้จะทำการป้อนชื่อที่ต้องการค้นหาเข้าไปในแบบฟอร์ม และเมื่อส่งค้นหา ก็จะทำให้การค้นหาชื่อที่มีตัวอักษรตรงกับที่ได้รับไป เมื่อค้นหาเสร็จสิ้นแล้วก็จะแสดงผลลัพธ์ออกมาในรูปแบบของตาราง โดยจะแสดงทั้งชื่อนักศึกษาและเลขประจำตัวนักศึกษา ในที่นี้เมื่อป้อนคำว่า “ณัฐ” ไปในส่วนของนักศึกษาปี 4A แล้ว โปรแกรมก็จะแสดงผลลัพธ์ออกมา 3 ชื่อที่ตรงคือ “ณัฐพงศ์”, “ณัฐพงษ์” และ “ณัฐวุฒิ” ดังแสดงในรูปที่ 4.3

Database Categories

นักศึกษาปัจจุบัน

4A | 3A | 2A

3T | 2T | 1T

วิชาเรียน

4A | 3A | 2A

3T/2 | 3T/1 | 2T/2

2T/1 | 1T/2 | 1T/1

โครงการนักศึกษา

ปีการศึกษา:

| ชื่อนักศึกษา      | เลขประจำตัวนักศึกษา |
|-------------------|---------------------|
| ณัฐพงศ์ ศรีรัตน   | 38014142            |
| ณัฐพงษ์ พัฒนพงษ์  | 38014143            |
| ณัฐวุฒิ มหเมธากิจ | 38014152            |

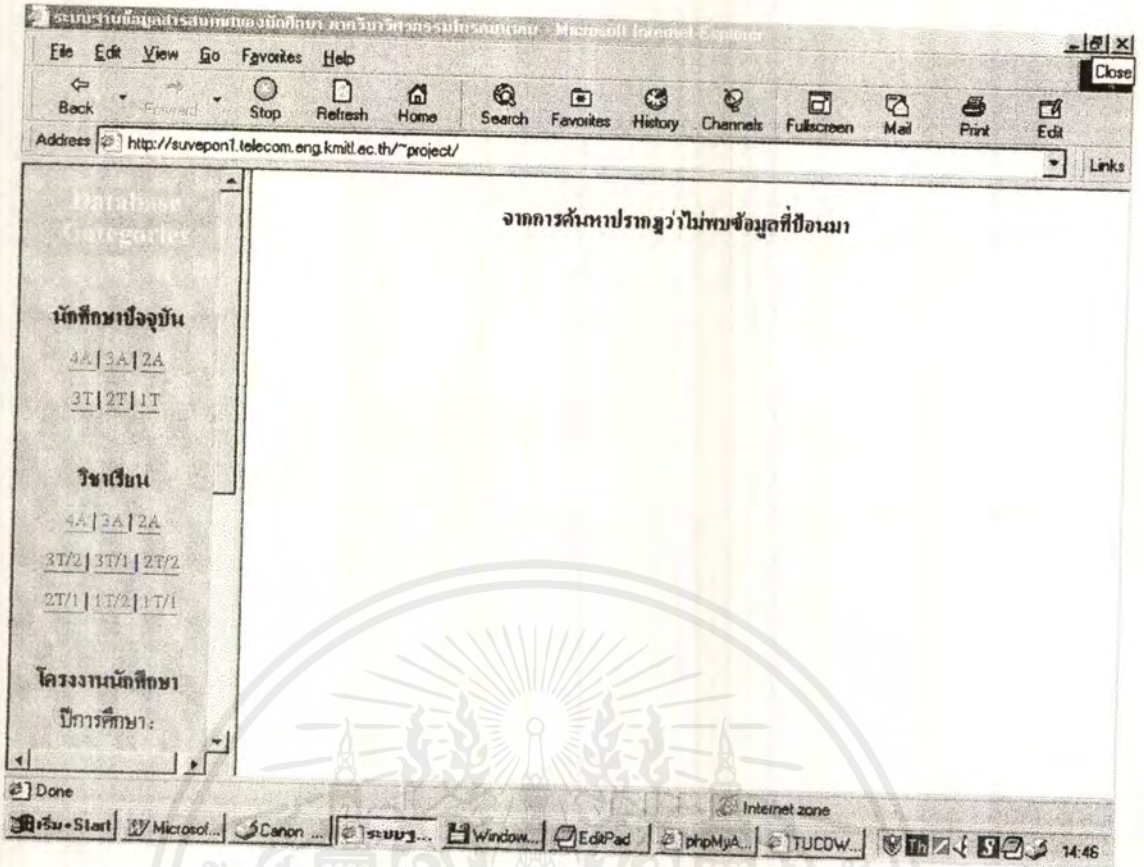
Done Internet zone

เริ่ม Start Microsoft Canon ... ระบบ Window EditPad phpMyA TUCOW 14:44

รูปที่ 4.3 แสดงผลลัพธ์จากการค้นหารายชื่อนักศึกษา

แต่ถ้าในกรณีที่ไม่มีพบข้อมูลที่สอดคล้องกับข้อมูลที่ใส่ป้อนให้ ก็จะปรากฏหน้าจอเตือนว่าไม่พบข้อมูลที่ป้อนมา ในที่นี้ทดลองป้อนชื่อว่า “จาดิ” เข้าไปในส่วนของนักศึกษาปี 4A ซึ่งไม่มีชื่อนี้อยู่ในฐานข้อมูล ก็จะปรากฏหน้าจอดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงหน้าจอ ในกรณีที่ ไม่พบข้อมูลที่ต้องการ

#### 4.2.2 ค้นหาโดยเลขประจำตัวนักศึกษา

ในที่นี้จะทำการป้อนเลขประจำตัวที่ต้องการค้นหาเข้าไปในแบบฟอร์ม และเมื่อส่งค้นหา ก็จะทำการค้นหาเลขประจำตัวที่มีเลขตรงกับที่ได้รับไป เมื่อค้นหาเสร็จสิ้นแล้วก็จะแสดงผลลัพธ์ออกมาในรูปแบบของตาราง โดยจะแสดงทั้งชื่อนักศึกษาและเลขประจำตัวนักศึกษา ในที่นี้เมื่อป้อนรหัสว่า "3801416" เข้าไปในส่วนของนักศึกษาปี 4A แล้ว โปรแกรมก็จะแสดงผลลัพธ์ออกมา 3 รหัสคือ "38014160", "38014166" และ "38014167" ดังแสดงในรูปที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Database Categories

นักศึกษาปัจจุบัน

4A | 3A | 2A

3T | 2T | 1T

วิชาเรียน

4A | 3A | 2A

3T/2 | 3T/1 | 2T/2

2T/1 | 1T/2 | 1T/1

โครงการนักศึกษา

ปีการศึกษา:

| ชื่อนักศึกษา            | เลขประจำตัวนักศึกษา |
|-------------------------|---------------------|
| คาริน หิรัญวิทย์        | 38014160            |
| ต่อชัย พัฒนพานิชสวัสดิ์ | 38014166            |
| ต่อพงศ์ สุภาคุสัย       | 38014167            |

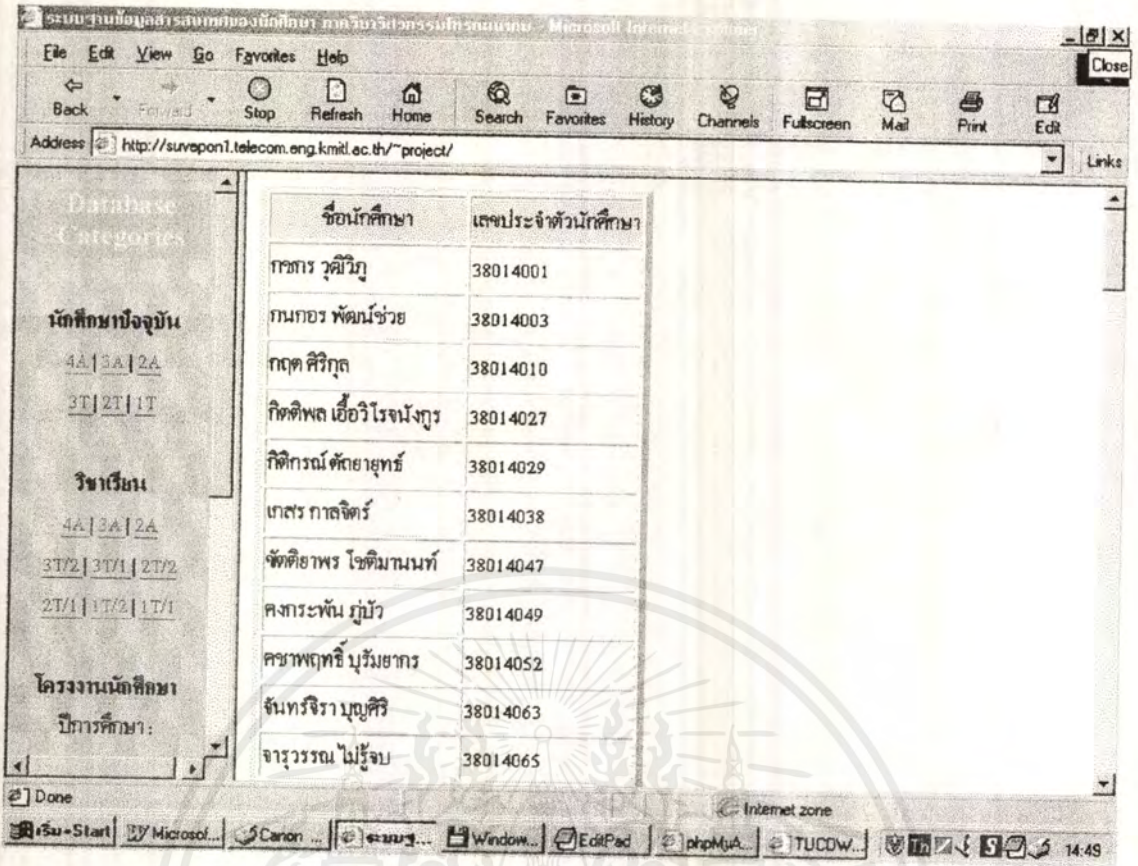
Done Internet zone

เริ่ม Start Microsoft... Canon ... ระบบว... Window... EditPad phpMjA... TUCOW... 14:48

รูปที่ 4.5 แสดงผลลัพธ์จากการค้นหาเลขประจำตัวนักศึกษา

#### 4.2.3 แสดงรายชื่อนักศึกษาทั้งหมด

เมื่อเลือกให้แสดงรายชื่อนักศึกษาทั้งหมด โปรแกรมก็จะไปสั่งให้ฐานข้อมูลแสดงรายชื่อนักศึกษาของชั้นปีนั้นๆ ออกมาทั้งหมด เพื่อเป็นประโยชน์ต่อการใช้ข้อมูลต่อไป ในที่นี้เลือกแสดงรายชื่อนักศึกษาของปี 4A ทั้งหมด ดังแสดงในรูปที่ 4.6



รูปที่ 4.6 แสดงรายชื่อนักศึกษาปี 4A ทั้งหมด

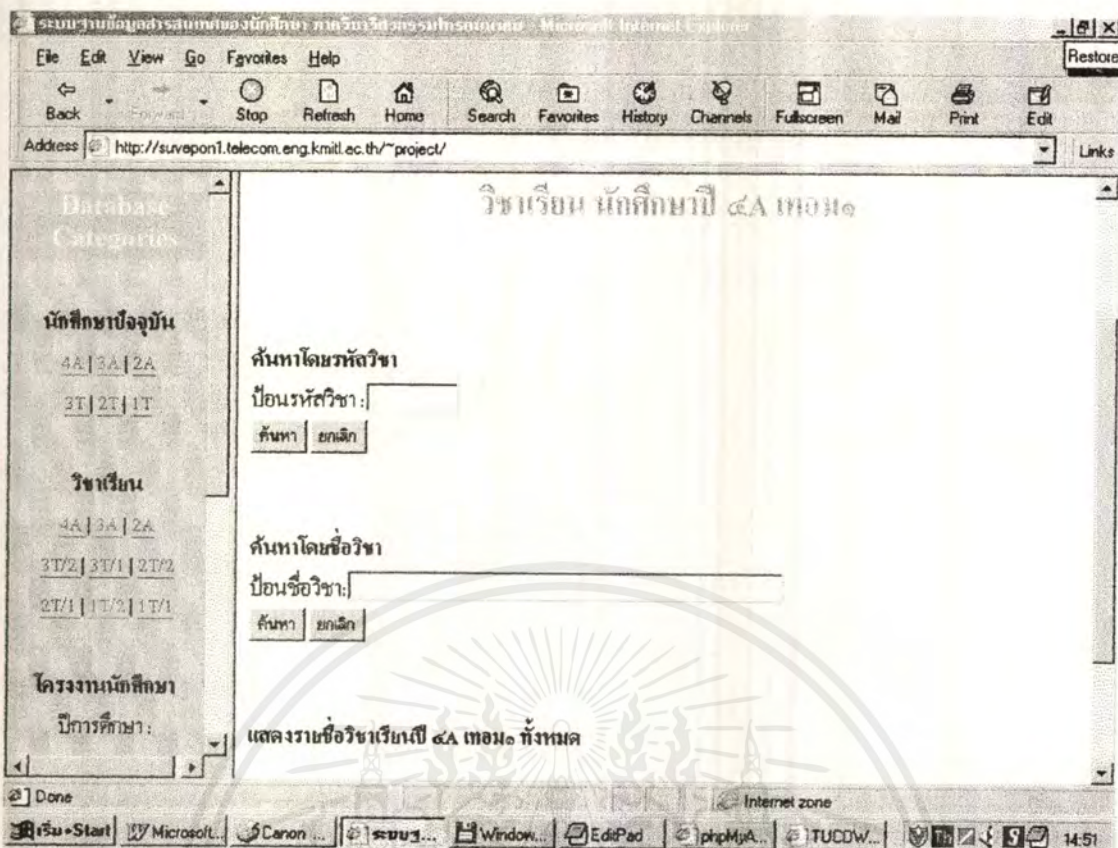
### 4.3 การทดลองส่วนการค้นหาวิชาเรียน

เมื่อเลือกการทำงานส่วนนี้ของนักศึกษาในชั้นปีต่างๆ ตัวโปรแกรมจะแสดงให้เลือกภาคเรียนที่ต้องการ ดังแสดงในรูปที่ 4.7

รูปที่ 4.7 แสดงหน้าจอการให้เลือกภาคเรียน

เมื่อทำการเลือกภาคเรียนที่ต้องการแล้ว ก็จะแสดงหน้าจอที่มีแบบฟอร์มให้ป้อน เพื่อค้นหาวิชาเรียน โดยสามารถค้นหาได้จากทั้งรหัสวิชา และชื่อวิชา หรือถ้าหากต้องการแสดงรายวิชาที่เรียนทั้งหมดในภาคเรียนนั้นๆ ก็สามารถเลือกให้แสดงออกมาทั้งหมดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 แสดงหน้าจอการรับข้อมูล เพื่อการแสดงข้อมูลวิชาเรียน

#### 4.3.1 ค้นหากรหัสวิชา

ในที่นี้จะทำการป้อนรหัสวิชาที่ต้องการค้นหาเข้าไปในแบบฟอร์ม และเมื่อส่งค้นหา ก็จะทำการค้นหาชื่อวิชาที่มีรหัสตรงกับที่ได้รับไป เมื่อค้นหาเสร็จสิ้นแล้วก็จะแสดงผลลัพธ์ออกมา โดยจะแสดงทั้งรหัสวิชา, ชื่อวิชา, หน่วยกิต, วันเวลา, ห้องเรียน และอาจารย์ผู้สอน ในที่นี้เมื่อป้อนรหัสวิชาว่า "101412" ในส่วนของนักศึกษาปี 4A เทอม1 ก็จะมีการแสดงผลลัพธ์ออก 5 วิชา คือ "MODERN LOCAL COMMUNICATION ARCHITECTURE", "MULTIMEDIA COMMUNICATION", "ERROR CONTROL CODING FOR DIGITAL COMMUNICATION SYSTEM" และ "INTERNET SYSTEM" ดังแสดงรูปที่ 4.9

| รหัสวิชา | ชื่อวิชา  | หน่วย<br>กิต | วัน - เวลา          | ห้อง<br>เรียน | อาจารย์ผู้<br>สอน |
|----------|---|--------------|---------------------|---------------|-------------------|
| 1014121  | MODERN LOCAL COMMUNICATION<br>ARCHITECTURE            | 3            | พ.ย. ๕.๐๐-<br>12.00 | A-503         | ศ.ดร.สุวิ<br>พล   |
| 1014124  | OPTICAL PROCESSING IN<br>COMMUNICATION                | 3            | อ. ๕.๐๐-<br>12.00   | A-501         | ดร.ไกรสิน         |
| 1014126  | MULTIMEDIA COMMUNICATION                              | 3            | ศ. ๕.๐๐-<br>12.00   | A-406         | ดร.อุทก<br>พงษ์   |
| 1014127  | ERROR CONTROL CODING FOR DIGITAL<br>COMMUNICATION SYS | 3            | ศ.๕.๐๐-<br>12.00    | A-405         | ดร.ปริญญ์         |
| 1014128  | INTERNET SYSTEM                                       | 3            | พ.ย.๑๑.๐๐-<br>16.00 | A-501         | อ.นภัทร           |

รูปที่ 4.9 แสดงผลลัพธ์จากการค้นหารายชื่อวิชาโดยรหัสวิชา

#### 4.3.2 ค้นหาจากชื่อวิชา

ในที่นี้เมื่อทำการป้อนชื่อวิชาที่ต้องการค้นหาแล้ว เมื่อสืบค้นหาที่จะมีการค้นหารายชื่อวิชาที่มีตัวอักษรตรงกับที่ได้รับไป เมื่อค้นหาเสร็จสิ้นแล้ว ก็จะแสดงผลที่ออกมาในรูปแบบของตาราง โดยจะแสดงครบทุกคอลัมน์ ในที่นี้เมื่อป้อนคำว่า “OPTICAL” เข้าไปในส่วนของชั้นปี 4A เทอม1 แล้ว ก็จะมีผลลัพธ์ออกมา 2 ชื่อวิชา คือ “OPTICAL COMMUNICATION” และ “OPTICAL PROCESSING IN COMMUNICATION” ดังแสดงในรูปที่ 4.10

Database Categories

นักศึกษาปัจจุบัน  
[4A](#) | [3A](#) | [2A](#)  
[3T](#) | [2T](#) | [1T](#)

วิชาเลือก  
[4A](#) | [3A](#) | [2A](#)  
[3T/2](#) | [3T/1](#) | [2T/2](#)  
[2T/1](#) | [1T/2](#) | [1T/1](#)

โครงการนักศึกษา  
 ปีการศึกษา:

| รหัสวิชา | ชื่อวิชา                               | หน่วย<br>กิต | วัน - เวลา          | ห้อง<br>เรียน | อาจารย์ผู้<br>สอน |
|----------|--|--------------|---------------------|---------------|-------------------|
| 1014104  | OPTICAL COMMUNICATION                  | 3            | พท. ๑๓.๐๐-<br>16.00 | A-502         | ศศ.อภิวัฒน์       |
| 1014124  | OPTICAL PROCESSING IN<br>COMMUNICATION | 3            | อ. ๕.๐๐-๑๒.๐๐       | A-501         | ดร.โกกรทิน        |

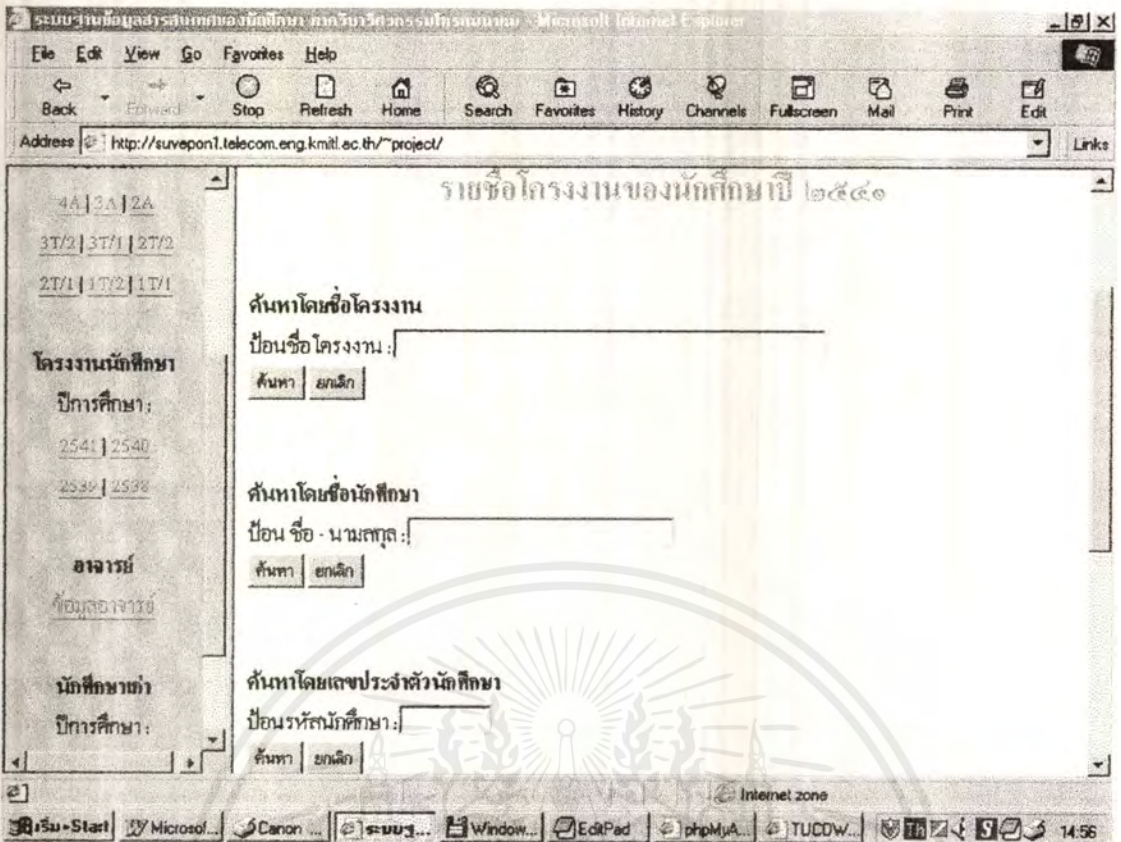
Done Internet zone

เริ่ม Start Microsoft... Canon ... ระบบ... Window... EditPad... phpMyA... TUCOW... 14:54

รูปที่ 4.11 แสดงรายชื่อวิชาปี 4A เทอม 1 ทั้งหมด

#### 4.4 การทดลองส่วนการค้นหารายชื่อโครงการนักศึกษา

เมื่อเลือกการทำงานส่วนนี้ของนักศึกษาในชั้นปีการศึกษาต่างๆ ตัวโปรแกรมจะแสดงแบบฟอร์มให้ป้อนเพื่อค้นหารายชื่อโครงการของนักศึกษา โดยสามารถค้นหาจาก ชื่อโครงการ, ชื่อนักศึกษา, รหัสประจำตัวนักศึกษา และอาจารย์ที่ปรึกษา หรือถ้าหากต้องการให้แสดงรายชื่อโครงการในปีการศึกษานั้นๆ ออกมาทั้งหมด ก็สามารถเลือกให้แสดงออกมาทั้งหมดได้ ดังแสดงในรูปที่ 4.12



รูปที่ 4.12 แสดงหน้าจอการรับข้อมูล เพื่อการแสดงผลข้อมูลชื่อโครงการนักศึกษา

#### 4.4.1 ค้นหาโดยชื่อโครงการ

ในที่นี้จะทำการป้อนชื่อโครงการที่ต้องการค้นหาเข้าไปในแบบฟอร์ม และเมื่อส่งค้นหาที่จะทำการค้นหาชื่อโครงการจากฐานข้อมูล เมื่อค้นหาเสร็จสิ้นแล้วก็จะแสดงผลลัพธ์ออกมา โดยจะแสดงทั้งชื่อนักศึกษา, รหัสประจำตัวนักศึกษา, ชื่อโครงการ และชื่ออาจารย์ที่ปรึกษา ในที่นี้เมื่อป้อนชื่อโครงการว่า “การสื่อสาร” ในส่วนของโครงการปีการศึกษา 2541 ก็จะมีการแสดงผลลัพธ์ออกมา 3 ชื่อโครงการ คือ “การสื่อสารข้อมูลผ่านอินฟราเรด”, “การสื่อสารข้อมูลเสียงผ่านเครือข่ายอินเทอร์เน็ตโดยใช้โครงข่ายโทรศัพท์” และ “การสื่อสารสัญญาณคลื่นไฟฟ้าหัวใจผ่านโครงข่ายโทรศัพท์” ดังแสดงในรูปที่ 4.13

| ชื่อนักศึกษา         | เลขประจำตัวนักศึกษา | ชื่อโครงการ   | ชื่ออาจารย์ที่ปรึกษา |
|----------------------|---------------------|---|----------------------|
| กฤต ศิริกุล          | 38014010            | การสื่อสารข้อมูลผ่านอินเทอร์เน็ต                                      | สุวิมล สิทธิวิชา     |
| บุรินทร์ ปิยะพรพันธ์ | 38014264            | การสื่อสารข้อมูลผ่านอินเทอร์เน็ต                                      | สุวิมล สิทธิวิชา     |
| ศรพทธี บุริมษากร     | 38014052            | การสื่อสารข้อมูลเสียงผ่านเครือข่ายอินเทอร์เน็ต โดยใช้โครงข่ายโทรศัพท์ | นภัทร สระเอี่ยม      |
| ณัฐพงษ์ พัฒนพงษ์     | 38014143            | การสื่อสารข้อมูลเสียงผ่านเครือข่ายอินเทอร์เน็ต โดยใช้โครงข่ายโทรศัพท์ | นภัทร สระเอี่ยม      |
| ปาลธรรม เกษมทรัพย์   | 38014291            | การสื่อสารข้อมูลเสียงผ่านเครือข่ายอินเทอร์เน็ต โดยใช้โครงข่ายโทรศัพท์ | นภัทร สระเอี่ยม      |
|                      |                     | การสื่อสารสัญญาณคลื่นไฟฟ้าหัวใจผ่านโครงข่าย                           | สุวิมล สิทธิวิชา     |

รูปที่ 4.13 แสดงผลลัพธ์จากการค้นหาชื่อโครงการ

#### 4.4.2 ค้นหาโดยชื่อนักศึกษา

ในที่นี้เมื่อทำการป้อนชื่อนักศึกษาที่ต้องการค้นหาเข้าไปในแบบฟอร์ม และเมื่อส่งค้นหา ก็จะทำให้การค้นหาชื่อโครงการที่มีชื่อนักศึกษาตรงกับที่ได้รับไป เมื่อค้นหาเสร็จสิ้นแล้ว ก็จะแสดงผลลัพธ์ออกมา ในที่นี้เมื่อชื่อนักศึกษาว่า “ต่อชัย” ในส่วนของโครงการนักศึกษาปี 2541 ก็จะมีการแสดงผลลัพธ์ออกมา 1 ชื่อโครงการคือ “การพัฒนาโปรแกรมภายใต้ระบบปฏิบัติการลินุกซ์” ดังแสดงในรูปที่ 4.14

The screenshot shows a web browser window with the following content:

- Address bar: <http://suvepon1.telecom.eng.kmitl.ac.th/~project/>
- Table of search results:
 

| ชื่อนักศึกษา     | เลขประจำตัวนักศึกษา | ชื่อโครงการ                                | ชื่ออาจารย์ที่ปรึกษา |
|------------------|---------------------|--|----------------------|
| ต่อชัย พัฒนพานิช | 38014166            | การพัฒนาโปรแกรมภายใต้ระบบปฏิบัติการลินุกซ์ | สุวิมล ฤทธิ์วิภา     |
- Sidebar (Left):
  - โครงการนักศึกษา
  - ปีการศึกษา:
    - 2541 | 2540
    - 2539 | 2538
  - อาจารย์
  - ชื่อนักศึกษา
  - นักศึกษาค้นหา
  - ปีการศึกษา:

รูปที่ 4.14 แสดงผลลัพธ์จากการค้นหารายชื่อโครงการ โดยชื่อนักศึกษา

#### 4.4.3 ค้นหาโดยเลขประจำตัวนักศึกษา

ในที่นี้เมื่อทำการป้อนเลขประจำตัวนักศึกษาลงไปในแบบฟอร์ม แล้วส่งค้นหา ก็จะทำการค้นหารายชื่อโครงการที่มีเลขประจำตัวนักศึกษาตรงกับที่ได้รับไป เมื่อค้นหาเสร็จสิ้นแล้ว ก็จะแสดงผลลัพธ์ออกมา ในที่นี้เมื่อป้อนเลขประจำตัวนักศึกษาคือ “3801416” ในส่วนของปีการศึกษา 2541 ก็จะมีการแสดงผลลัพธ์ออกมา 2 ชื่อโครงการ คือ “การพัฒนาโปรแกรมภายใต้ระบบปฏิบัติการลินุกซ์” และ “เครื่องวัดระยะทางด้วยแสง” ดังแสดงในรูปที่ 4.15

| ชื่อนักศึกษา     | เลขประจำตัวนักศึกษา | ชื่อโครงการ                                 | ชื่ออาจารย์ที่ปรึกษา |
|------------------|---------------------|---|----------------------|
| ต่อชัย พัฒนพานิช | 38014166            | การพัฒนาโปรแกรมภายใต้ระบบปฏิบัติการวินโดวส์ | ศิวพล สิทธิรัช       |
| คาริน หิรัญวิทย์ | 38014160            | เครื่องวิเคราะห์ทางด้วยแสง                  | สมเกียรติ ฤกษ์       |

รูปที่ 4.15 แสดงผลลัพธ์จากการค้นหารายชื่อโครงการโดยเลขประจำตัวนักศึกษา

#### 4.4.4 ค้นหาโดยอาจารย์ที่ปรึกษา

ในที่นี้เมื่อทำการป้อนชื่ออาจารย์ที่ปรึกษาที่ต้องการค้นหาเข้าไปในแบบฟอร์ม และเมื่อส่งค้นหา ก็จะทำการค้นหาชื่อโครงการที่มีชื่ออาจารย์ที่ปรึกษาตรงกับที่ได้รับเข้าไป เมื่อค้นหาเสร็จสิ้นแล้ว ก็จะแสดงผลที่ออกมา ในที่นี้เมื่อชื่ออาจารย์ที่ปรึกษาคือ “สมเกียรติ” ในส่วนของโครงการนักศึกษาปี 2541 ก็จะมีการแสดงผลที่ออกมา 1 ชื่อโครงการคือ “เครื่องวิเคราะห์ทางด้วยแสง” ดังแสดงในรูปที่ 4.16

| ชื่อนักศึกษา     | เลขประจำตัวนักศึกษา | ชื่อโครงการ              | ชื่ออาจารย์ที่ปรึกษา |
|------------------|---------------------|--------------------------|----------------------|
| คาริน หิรัญวิทย์ | 38014160            | เครื่องวัดระยะทางด้วยแสง | สมเกียรติ ฤกษ์วิญญู  |
| ภัทริน วิวารัตน์ | 38014368            | เครื่องวัดระยะทางด้วยแสง | สมเกียรติ ฤกษ์วิญญู  |

รูปที่ 4.16 แสดงผลลัพธ์จากการค้นหารายชื่อโครงการ โดยชื่ออาจารย์ที่ปรึกษา

#### 4.4.5 แสดงรายชื่อโครงการทั้งหมด

เมื่อเลือกให้แสดงรายชื่อโครงการทั้งหมด โปรแกรมก็จะไปส่งในฐานข้อมูล แสดงรายชื่อโครงการของปีการศึกษานั้นๆ ออกมาทั้งหมด เพื่อเป็นประโยชน์ต่อการนำไปใช้ต่อไป ในที่นี้เลือกแสดงรายชื่อโครงการของปีการศึกษา 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The screenshot shows a web browser window with the address <http://suvsapon1.telecom.eng.kmitl.ac.th/~project/>. The main content is a table with the following data:

| ชื่อนักศึกษา               | เลขประจำตัวนักศึกษา | ชื่อ ครงงาน  | ชื่ออาจารย์ที่ปรึกษา |
|----------------------------|---------------------|--|----------------------|
| ต่อชัย พัฒนพานิชาทวีศักดิ์ | 38014166            | การพัฒนาโปรแกรมภายใต้ระบบปฏิบัติการลินุกซ์                           | สุวิพล ลิทธิธิภาค    |
| กฤต ศิริกุล                | 38014010            | การสื่อสารข้อมูลผ่านอินฟราเรด  | สุวิพล ลิทธิธิภาค    |
| บุรินทร์ ปิยะพรพันธ์       | 38014264            | การสื่อสารข้อมูลผ่านอินฟราเรด  | สุวิพล ลิทธิธิภาค    |
| ศรภาพุทธิ์ บูรัมย์กร       | 38014052            | การสื่อสารข้อมูลเสียงผ่านเครือข่ายอินเทอร์เน็ตโดยใช้โครงข่ายโทรศัพท์ | นภัทร สระเยี่ยม      |
| ณัฐพงษ์ วัฒนพงษ์           | 38014143            | การสื่อสารข้อมูลเสียงผ่านเครือข่ายอินเทอร์เน็ตโดยใช้โครงข่ายโทรศัพท์ | นภัทร สระเยี่ยม      |
| ปาลชรรณ เกษม               | 38014052            | การสื่อสารข้อมูลเสียงผ่านเครือข่ายอินเทอร์เน็ต                       |                      |

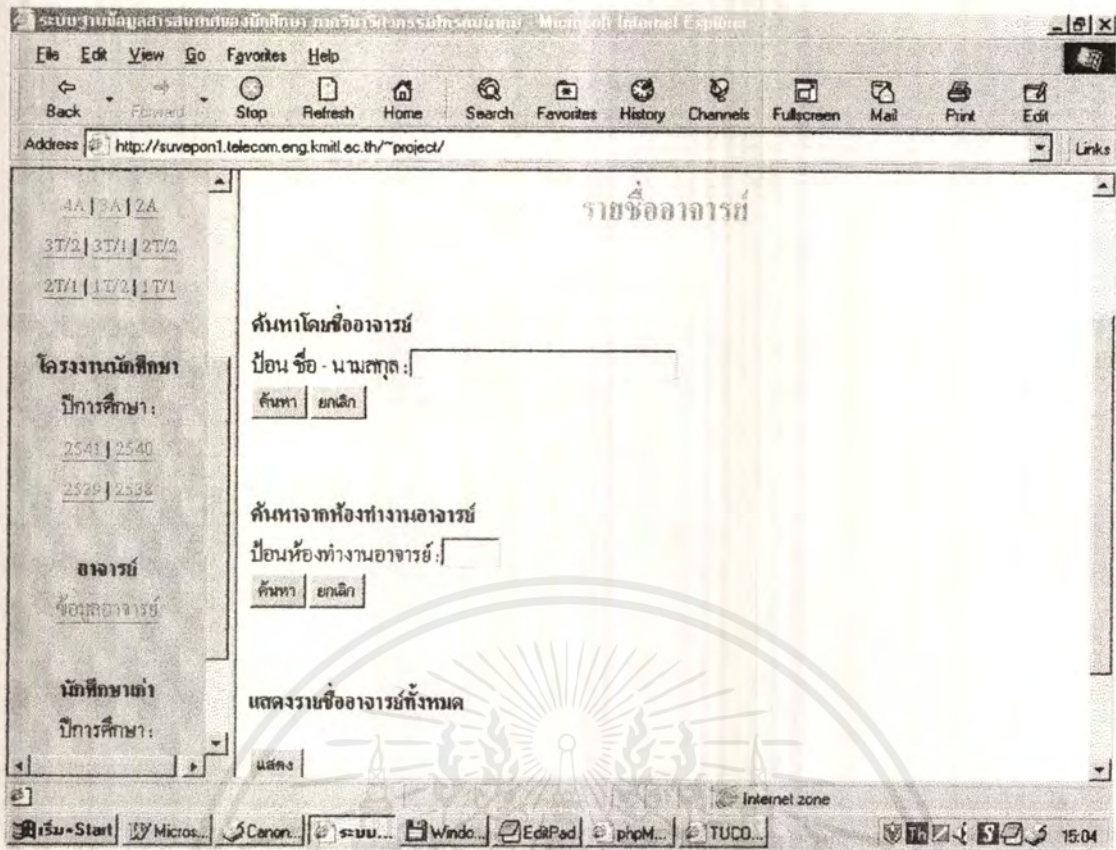
The sidebar on the left contains navigation links for semesters (4A, 3A, 2A, 3T/2, 3T/1, 2T/2, 2T/1, 1T/2, 1T/1) and sections for 'โครงการนนักศึกษ' (Student Project), 'อาจารย์' (Faculty), and 'นักศึกษาเก่า' (Former Students).

รูปที่ 4.17 แสดงรายชื่อ ครงงานปีการศึกษา 2541 ทั้งหมด

#### 4.5 การทดลองส่วนการค้นหารายชื่ออาจารย์

เมื่อเลือกการทำงานส่วนนี้ ตัวโปรแกรมจะแสดงแบบฟอร์มให้ป้อนเพื่อค้นหารายชื่ออาจารย์ของภาควิชาโทรคมนาคมพร้อมทั้งห้องทำงานของอาจารย์ โดยสามารถค้นหาได้จากชื่ออาจารย์ และห้องทำงานของอาจารย์ หรือถ้าหากต้องการให้แสดงรายชื่ออาจารย์ออกมาทั้งหมด ก็สามารถเลือกให้แสดงออกมาทั้งหมดได้ ดังแสดงในรูปที่ 4.18

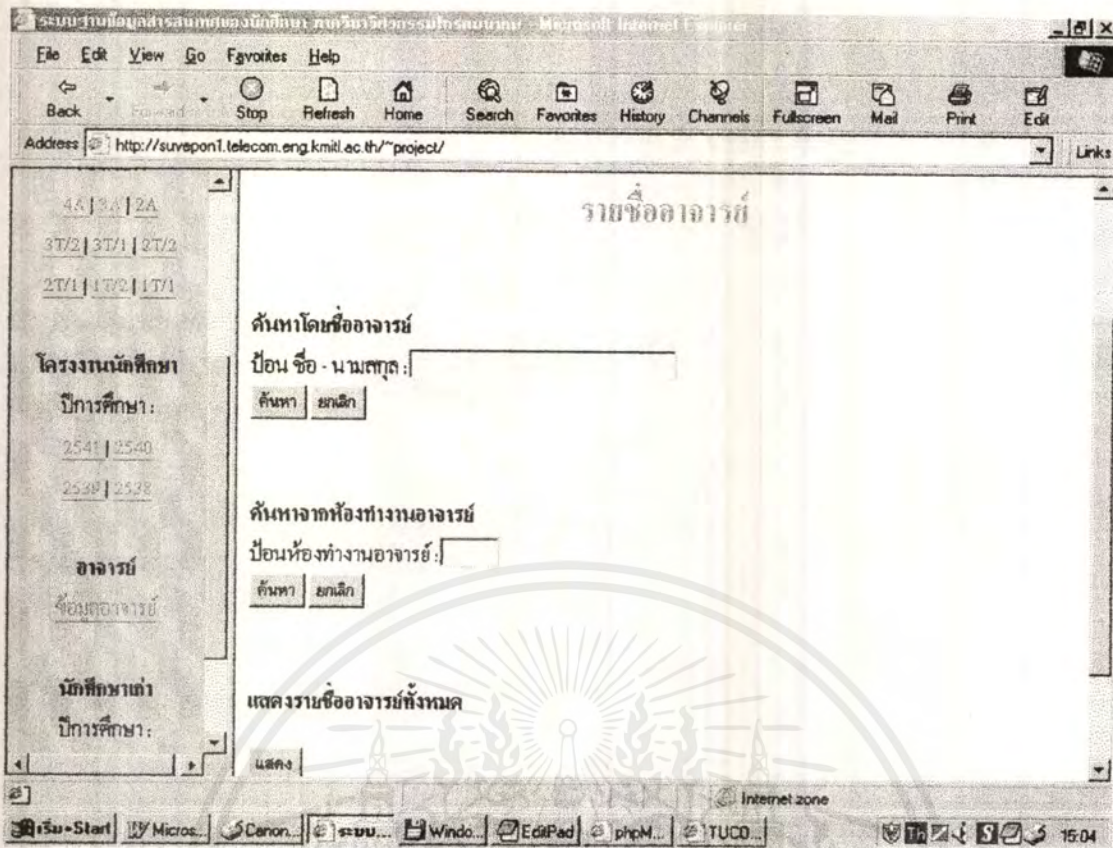
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 แสดงหน้าจอการรับข้อมูลเพื่อการแสดงข้อมูลอาจารย์

#### 4.5.1 ค้นหาโดยชื่ออาจารย์

ในที่นี้จะทำการป้อนชื่ออาจารย์ที่ต้องการค้นหาเข้าไปในแบบฟอร์ม และเมื่อส่งค้นหาที่จะทำการค้นหาชื่ออาจารย์จากฐานข้อมูล เมื่อค้นหาเสร็จสิ้นแล้ว ก็จะแสดงผลลัพธ์ออกมา โดยจะแสดงออกมาทั้งชื่ออาจารย์และห้องทำงานอาจารย์ ในที่นี้เมื่อป้อนชื่ออาจารย์ว่า “สุ” ก็จะมีการแสดงผลลัพธ์ออกมา 3 ชื่อคือ “สุทธิชัย นพนาดิพงษ์”, “สุวิพล ลิทธิชีวะภาค” และ “สุรพล บุญจันทร์” ดังแสดงในรูปที่ 4.19



รูปที่ 4.18 แสดงหน้าจอการรับข้อมูลเพื่อการแสดงข้อมูลอาจารย์

4.5.1 ค้นหาโดยชื่ออาจารย์

ในที่นี้จะทำการป้อนชื่ออาจารย์ที่ต้องการค้นหาเข้าไปในแบบฟอร์ม และเมื่อส่งค้นหาก็จะทำการค้นหาชื่ออาจารย์จากฐานข้อมูล เมื่อค้นหาเสร็จสิ้นแล้ว ก็จะแสดงผลลัพธ์ออกมา โดยจะแสดงออกมาทั้งชื่ออาจารย์และห้องทำงานอาจารย์ ในที่นี้เมื่อป้อนชื่ออาจารย์ว่า “สุ” ก็จะมีการแสดงผลลัพธ์ออกมา 3 ชื่อคือ “สุทธิชัย นพนาศิพงษ์”, “สุวิพล ลิทธิชีวภาค” และ “สุรพล บุญจันทร์” ดังแสดงในรูปที่ 4.19

ระบบงานข้อมูลสารสนเทศของมหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี - Microsoft Internet Explorer

Address: http://suevon1.telecom.eng.kmitl.ac.th/~project/

| ชื่ออาจารย์         | ห้องทำงานอาจารย์ |
|---------------------|------------------|
| มนุญ สุขเกษม        | T201             |
| ศุทธิชัย นพนาสีพงษ์ | T302             |
| ศุวิพล ศิทธิชีวภาค  | T304             |
| ศุวิพล บุญจันทร์    | T308             |

โครงการนักศึกษา  
ปีการศึกษา:  
2541 | 2540  
2536 | 2538

อาจารย์  
ข้อมูลอาจารย์

นักศึกษาเก่า  
ปีการศึกษา:

Done Internet zone

เริ่ม Start Microsoft Canon ระบบ Window EditPad phpMyA TUCOW 15:05

รูปที่ 4.19 แสดงผลลัพธ์จากการค้นหาชื่ออาจารย์

#### 4.5.2 ค้นหาโดยห้องทำงานอาจารย์

ในที่นี้เมื่อทำการป้อนห้องทำงานอาจารย์ที่ต้องการค้นหาเข้าไปในแบบฟอร์ม เมื่อส่งค้นหาก็จะทำการค้นหาชื่ออาจารย์ที่มีห้องทำงานอาจารย์ตรงกับที่รับเข้าไป เมื่อค้นหาเสร็จสิ้นแล้ว ก็จะแสดงผลลัพธ์ออกมา ในที่นี้เมื่อป้อนชื่อทำงานว่า "T31" ก็จะมีการแสดงผลลัพธ์ออกมา 3 ชื่ออาจารย์คือ "ไกรสิน ส่วงวัฒนา", "อัศรพล ศรีรัตน์" และ "สมเกียรติ ฤกษ์วีรญาญ" ดังแสดงในรูปที่ 4.20

ระบบฐานข้อมูลครูของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง - Microsoft Internet Explorer

File Edit View Go Favorites Help

Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit

Address <http://suvepon1.telecom.eng.kmitl.ac.th/~project/> Links

| ชื่ออาจารย์           | ห้องทำงานอาจารย์ |
|-----------------------|------------------|
| ไกรสิน ตังวัฒนา       | T310             |
| ยัศวพล ตวีรัตน์       | T314             |
| ศมเกียรติ ฤกษ์วีระบุญ | T316             |

โครงการนักศึกษา  
ปีการศึกษา:  
2541 | 2540  
2539 | 2538

อาจารย์  
ข้อมูลอาจารย์

นักศึกษาเก่า  
ปีการศึกษา:

Done Internet zone

เริ่ม Start Microsoft Canon ... ธรรม... Window... EditPad phpMyA... TUCOW... 15:07

รูปที่ 4.20 แสดงผลลัพธ์การค้นหาชื่ออาจารย์โดยห้องทำงานอาจารย์

#### 4.5.3 แสดงรายชื่ออาจารย์ทั้งหมด

เมื่อเลือกให้แสดงรายชื่ออาจารย์ทั้งหมด โปรแกรมก็จะไปส่งให้ฐานข้อมูลแสดงรายชื่ออาจารย์ของภาควิชาฯ ออกมาทั้งหมด เพื่อเป็นประโยชน์ต่อการนำไปใช้ต่อไป ดังแสดงในรูปที่ 4.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address: <http://suepon1.telecom.eng.kmitl.ac.th/~project/>

4/ 3A | 2A  
3T/2 | 3T/1 | 2T/2  
2T/1 | 1T/2 | 1T/1

**โครงการนักศึกษา**  
ปีการศึกษา:  
2541 | 2540  
2539 | 2538

**อาจารย์**  
ข้อมูลอาจารย์

**นักศึกษาเก่า**  
ปีการศึกษา:

| ชื่ออาจารย์            | ห้องทำงานอาจารย์ |
|------------------------|------------------|
| ปราโมทย์ วัฒนเขิน      | T101             |
| เกรียงไกร วงศ์โรจนภรณ์ | T101             |
| สมยศ จุณณะปิยะ         | T102             |
| ไมโนย ไกรฤกษ์          | T104             |
| สมผล โกศลวิทย์         | T105             |
| ทองทศ วานิชศิริ        | T107             |
| นภัทร สระเอี่ยม        | T110             |
| ถวิศ พิ้งมา            | T201             |
| มนูญ สุขเกษม           | T201             |
| พิเชษฐ ม่วงนวล         | T201             |
| กอมรัช เดชหาญ          | T203             |

Done Internet zone

เริ่ม \*Starl | Microsoft... | Canon ... | าระบบ... | Window... | EditPad | phpMyA... | TUCDW... | 15:07

รูปที่ 4.21 แสดงรายชื่ออาจารย์ของภาควิชาโทรคมนาคมทั้งหมด

#### 4.6 การทดลองส่วนการค้นหารายชื่อนักศึกษาเก่า

การทำงานส่วนนี้มีหน้าที่และคุณสมบัติเหมือนกับการค้นหารายชื่อนักศึกษาทุกประการ และที่ฐานข้อมูลจะเก็บข้อมูลของนักศึกษาเก่า เพื่อประโยชน์ในการนำไปใช้ต่อไป ในที่นี้เมื่อเลือกให้แสดงรายชื่อนักศึกษาเก่าที่จบการศึกษาในปี 2538 จะแสดงผลดังรูปที่ 4.22

ระบบงานบัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง - Microsoft Internet Explorer

File Edit View Go Favorites Help

Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit

Address http://suvepon1.telecom.eng.kmitl.ac.th/~project/ Links

**โครงการบัณฑิตศึกษา**

**ปีการศึกษา:**  
2541 | 2540  
2539 | 2538

**อาจารย์**  
ข้อมูลอาจารย์

**นักศึกษาเก่า**  
**ปีการศึกษา:**  
2540 | 2539 | 2538

เกี่ยวกับระบบ

| ชื่อนักศึกษา           | เลขประจำตัวนักศึกษา |
|------------------------|---------------------|
| บุญเลิศ สุทธิสุภากร    | 37014220            |
| สวพษ์ภักดิ์ ณะจุฑา     | 37014403            |
| วิวัฒน์ เลิศทนต์ศักดิ์ | 37014443            |
| คุญพงศ์ ออมสิน         | 37014126            |
| สุพิทักษ์ สงศิริ       | 37014513            |
| วรลักษณ์ คงเด่นฟ้า     | 37014376            |
| วาสนา กล้าการนา        | 37014393            |
| พงศธร จินดาพงศ์        | 37014062            |
| ชัยชรา ถาวัง           | 37014043            |
| แก้ว แนวน้อยงาม        | 37014049            |
| สุรณรงค์ คมตะศิลา      | 37014084            |

Internet zone

เริ่ม Start Microsoft... Canon ... ระบบ... Window... EditPad phpMyA... TUCOW... 15:08

รูปที่ 4.22 แสดงรายชื่อนักศึกษาเก่าที่จบการศึกษาในปี 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### วิจารณ์และสรุป

การออกแบบระบบงานสารสนเทศเพื่อการจัดการ เป็นการศึกษารวบรวมข้อมูลจากเอกสารที่เกี่ยวข้อง และการสอบถามความต้องการของผู้ใช้ระบบ ระบบงานสารสนเทศมีวัตถุประสงค์ในการออกแบบเพื่อเน้นในการแบ่งเบาภาระ และอำนวยความสะดวกในการทำงานของผู้ใช้เป็นหลัก เพื่อการจัดการข้อมูล หรือเอกสารที่เกี่ยวข้องกับนักศึกษา และเพื่อเพิ่มประสิทธิภาพในการทำงาน การจัดเก็บข้อมูล การแก้ไขข้อมูล การจัดพิมพ์เอกสาร ให้เป็นไปอย่างรวดเร็วและถูกต้อง อีกทั้งยังช่วยในการลดขั้นตอนต่างๆ อาทิ ในการคำนวณคะแนนเฉลี่ยสะสม ในการค้นหาวิชาเรียนที่นักศึกษาลงทะเบียน เพื่อนำมาใส่เอกสารทางการศึกษา เป็นต้น ทำให้งานทะเบียนสามารถตอบสนองความต้องการในด้านการบริการแก่นักศึกษาได้เป็นอย่างดี

#### ประโยชน์ของโครงการ

1. ช่วยอำนวยความสะดวกให้แก่ผู้ที่ต้องการทราบรายละเอียดต่างๆ ของภาควิชาโทรคมนาคม
2. ทำให้สะดวกต่อการเรียกดู เพราะสามารถเรียกดูจากคอมพิวเตอร์เครื่องใดก็ได้ที่ต่อกับเครือข่ายอินเทอร์เน็ตไว้
3. ทำให้ประหยัดทรัพยากรต่างๆ ทั้งบุคลากรและอุปกรณ์ต่างๆ
4. การทำงานจะมีความถูกต้องแม่นยำมากยิ่งขึ้น

#### แนวทางการพัฒนา

- HTML มีข้อจำกัดในการแสดงผล ดังนั้นจึงควรนำภาษาจาวามาใช้เพื่อให้ User Interface ใช้งานได้ง่ายขึ้น
- เพิ่มเติมข้อมูลของภาควิชาฯ ให้สมบูรณ์ยิ่งขึ้นไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก. แสดงโปรแกรมที่ใช้ในระบบฐานข้อมูลสารสนเทศ

แสดงโปรแกรม index.html

```
<html>
```

```
<head>
```

```
<title>ระบบฐานข้อมูลสารสนเทศของนักศึกษา ภาควิชาวิศวกรรม โทรคมนาคม</title>
```

```
</head>
```

```
<frameset cols="160,*">
```

```
  <frame src="title.html" noresize>
```

```
  <frame src="main.html" name=main>
```

```
</frameset>
```

```
</html>
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## แสดงโปรแกรม title.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Title</TITLE>
```

```
</HEAD>
```

```
<BODY background="qube.gif">
```

```
<center></center><br>
```

```
<center>
```

```
<b>นักศึกษาปัจจุบัน</b><br>
```

```
<A HREF="std4A_main.html" TARGET="main">4A </A>|
```

```
<A HREF="std3A_main.html" TARGET="main">3A </A>|
```

```
<A HREF="std2A_main.html" TARGET="main">2A </A>
```

```
<br>
```

```
<A HREF="std3T_main.html" TARGET="main">3T </A>|
```

```
<A HREF="std2T_main.html" TARGET="main">2T </A>|
```

```
<A HREF="std1T_main.html" TARGET="main">1T </A>
```

```
</center>
```

```
<br>
```

```
<center>
```

```
<b>วิชาเรียน</b><br>
```

```
<A HREF="study4A_main.html" TARGET="main">4A </A>|
```

```
<A HREF="study3A_main.html" TARGET="main">3A </A>|
```

```
<A HREF="study2A_main.html" TARGET="main">2A </A>
```

```
<br>
```

```
<A HREF="study3T_R2_main.html" TARGET="main">3T/2 </A>|
```

```
<A HREF="study3T_R1_main.html" TARGET="main">3T/1 </A>|
```

```
<A HREF="study2T_R2_main.html" TARGET="main">2T/2 </A>
```

```
<br>
```

```
<A HREF="study2T_R1_main.html" TARGET="main">2T/1 </A>|
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<A HREF="study1T\_R2\_main.html" TARGET="main">IT/2 </A>|

<A HREF="study1T\_R1\_main.html" TARGET="main">IT/1 </A>

</center>

<br>

<center>

<b>โครงการนักศึกษา</b><br>

ปีการศึกษา :<br>

<A HREF="proj2541\_main.html" TARGET="main">2541 </A>|

<A HREF="proj2540\_main.html" TARGET="main">2540 </A><br>

<A HREF="proj2539\_main.html" TARGET="main">2539 </A>|

<A HREF="proj2538\_main.html" TARGET="main">2538 </A>

</center>

<br>

<center>

<b>อาจารย์</b><br>

<A HREF="teacher\_main.html" TARGET="main">ข้อมูลอาจารย์</A>

</center>

<br>

<center>

<b>นักศึกษาเก่า</b><br>

ปีการศึกษา :<br>

<A HREF="std2540\_main.html" TARGET="main">2540 </A>|

<A HREF="std2539\_main.html" TARGET="main">2539 </A>|

<A HREF="std2538\_main.html" TARGET="main">2538 </A>

</center>

<br>

<center>

<A HREF="main.html" TARGET="main">

<b>เกี่ยวกับระบบ</b>

</A>

</center>

</BODY>เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

</HTML> ไม่มีการแก้ไขทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## แสดงโปรแกรม main.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Title</TITLE>
```

```
</HEAD>
```

```
<BODY bgcolor="white">
```

```
<center>
```

```
<br>
```

```
<h2><font color="Olive">ระบบฐานข้อมูลสารสนเทศสำหรับภาควิชาวิศวกรรมโทรคมนาคม</font></h2>
```

```
</center>
```

```
<br>
```

โครงการนี้เป็นการพัฒนาฐานข้อมูลสารสนเทศขึ้นเพื่อการใช้งาน ในส่วนของการจัดการฐานข้อมูลภายในภาควิชา เพื่อให้ผู้ใช้สามารถสืบค้นข้อมูล ได้อย่างรวดเร็วและมีประสิทธิภาพ โดยนำเทคโนโลยีทางด้านสารสนเทศและระบบอินเทอร์เน็ตมาช่วยในการพัฒนาเพื่อให้ใช้งานได้กว้างขวางมากขึ้น

```
<br><br>
```

การพัฒนาฐานข้อมูลดังกล่าวใช้ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ MySQL ซึ่งทำงานบนระบบปฏิบัติการลินุกซ์ (Linux) และใช้ภาษาสคริปต์ PHP ในการติดต่อกับ ฐานข้อมูลและเขียนภาษา HTML ในการติดต่อกับผู้ใช้งาน เพื่อแสดงผลในรูปแบบ Web เพื่อให้ผู้ใช้งานใช้ได้อย่างง่าย และสะดวกจากเครื่องคอมพิวเตอร์ใดๆ ก็ได้ที่ต่อกับระบบอินเทอร์เน็ตไว้

```
<center>
```

```
<h2><font color="Olive">INFORMATION DATABASE SYSTEM FOR COMMUNICATION  
ENGINEERING DEPARTMENT</font></h2>
```

```
</center>
```

This project is aimed to develop the Information Database System in order to facilitate Information management within the department. Thus, the user, both internal and external personnels, will be able to utilize the information more expediently and effectively. Information Technology and Internet System are brought in here as the assisting tools for further development and innovation.

```
<br><br>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Such database development, as prescribed earlier, uses relative database management system; MySQL, which is operating on LINUX, and PHP Script language to communicate bilaterally with the database. The result, moreover, can be displayed on the web by the use of HTML for remote personnels with Internet assessibility.

<br><br>

<center><b>จัดทำโดย นาย ต่อชัย พัฒนพานิชสวัสดิ์ 38014166</b></center>

</BODY>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## แสดงโปรแกรม std4A\_main.html

```
<html>
<head>
<title>ระบบฐานข้อมูลสารสนเทศของนักศึกษา ภาควิชาวิศวกรรมโทรคมนาคม ปี 4A</title>
</head>

<body text="Black" links="Blue" vlinks="Purple">

<!--Logo-->
<center>
<br>
</center>
<!--End Logo-->

<!--Student Search-->

<center><H2><font color="Green">รายชื่อนักศึกษาปี 4A</font></H2><br></center>

<form action=search4A_name.php3 method="post">
<b>ค้นหาโดยชื่อ</b><br>
ป้อน ชื่อ - นามสกุล :<input type="text" name="name4A" size="30"><br>
<input type="submit" value="ค้นหา">
<input type="reset" value="ยกเลิก">
</form>

<br>

<form action=search4A_id.php3 method="post">
<b>ค้นหาโดยเลขประจำตัวนักศึกษา</b><br>
ป้อนเลขประจำตัวนักศึกษา :<input type="text" name="id4A" size="9" maxlength="8"><br>
<input type="submit" value="ค้นหา">
<input type="reset" value="ยกเลิก">
</form>

<br>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>แสดงรายชื่อนักศึกษาปี 4A ทั้งหมด</b><br>

<form action=search4A\_stdall.php3 method="post">

<input type="submit" value="แสดง">

</form>

<br>

<i>\*\*\* ถ้าไม่ป้อนข้อมูล โปรแกรมจะทำการแสดงผลทั้งหมด</i>

<!--End Student Search-->

</body>

</html>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### แสดงโปรแกรม searchd4A\_name.php3

```
<html>
<body>

<?php

mysql_connect("localhost", "project", "project");
mysql_select_db("project");

$result = mysql_query("SELECT student_name, id
    from student4A where student_name like '%$name4A%' order by id");

if ($myrow = mysql_fetch_array($result))
    {
        echo "<table border=5>\n";
        echo "<tr bgcolor=#f1b6c1> <td><center>ชื่อนักศึกษา</center></td> <td><center>เลขประจำตัวนัก
ศึกษา</center></td> </tr>\n";
        do    {
            printf("<tr bgcolor=#ffffe0> <td>%s</td> <td>%s</td></tr>\n", $myrow
["student_name"], $myrow["id"]);
        }
        while ($myrow = mysql_fetch_array($result));

        echo "</table>\n";
    }
    else
    {
        echo "<center><b>จากการค้นหาปรากฏว่าไม่พบข้อมูลที่ป้อนมา</b></center>";
    }
mysql_close();
?>

</body>
</html>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### แสดงโปรแกรม searchd4A\_id.php3

```
<html>
<body>

</php

mysql_connect("localhost", "project", "project");
mysql_select_db("project");

$result = mysql_query("SELECT student_name,id
    from student4A where id like '%Sid4A%' order by id");

if ($myrow = mysql_fetch_array($result))
    {

    echo "<table border=5>\n";
    echo "<tr bgcolor=#ffb6c1> <td><center>ชื่อนักศึกษา</center></td> <td><center>เลขประจำตัวนักศึกษา</center>
</td> </tr>\n";

    do    {
        printf("<tr bgcolor=#ffffe0> <td>%s</td> <td>%s</td></tr>\n", $myrow["student_name"], $myrow
["id"]);
    }
    while ($myrow = mysql_fetch_array($result));

    echo "</table>\n";
    }
    else
    {
    echo "<center><b>จากการค้นหาปรากฏว่าไม่พบข้อมูลที่ป้อนมา</b></center>";
    }

mysql_close();
?>

</body>
</html>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## แสดงโปรแกรม study4A\_main.html

```
<html>
<head>
<title>ระบบฐานข้อมูลสารสนเทศของนักศึกษา ภาควิชาวิศวกรรมโทรคมนาคม ปี 4A</title>
</head>

<body text="Black" links="Blue" vlinks="Purple">

<center>
<br>
</center>

<!--Select Term-->
<center><H2><font color="Green">วิชาเรียน นักศึกษาปี 4A</font></H2><br></center><br>
<center>โปรดเลือกภาควิชาเรียน :<br>
<a href="search4A_studyterm1.html">ภาควิชาเรียนที่ 1 </a><br>
<a href="search4A_studyterm2.html">ภาควิชาเรียนที่ 2 </a><br>
</center>
<!--End Select Term-->

</body>
</html>
```



## แสดงโปรแกรม search4A\_studyterm1.html

```
<html>
<head>
<title>ระบบฐานข้อมูลสารสนเทศของนักศึกษา ภาควิชาวิศวกรรมโทรคมนาคม ปี 4A</title>
</head>

<body text="Black" links="Blue" vlinks="Purple">

<!--Logo-->
<center>
<br>
</center>
<!--End Logo-->

<!--Term1-->
<center><H2><font color="Green">วิชาเรียน นักศึกษาปี 4A เทอม1</font></H2><br></center><br>

<form action=search4A_study4At1_code.php3 method="post">
<b>ค้นหาโดยรหัสวิชา</b><br>
ป้อนรหัสวิชา :<input type="text" name="studycode4At1" size="9" maxlength="8"><br>
<input type="submit" value="ค้นหา">
<input type="reset" value="ยกเลิก">
</form>

<br>

<form action=search4A_study4At1_name.php3 method="post">
<b>ค้นหาโดยชื่อวิชา</b><br>
ป้อนชื่อวิชา:<input type="text" name="studyname4At1" size="50"><br>
<input type="submit" value="ค้นหา">
<input type="reset" value="ยกเลิก">
</form>

<br>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>แสดงรายชื่อวิชาเรียนปี 4A เทอม1ทั้งหมด</b><br>

<form action=search4A\_study4At1\_all.php3 method="post">

<input type="submit" value="แสดง">

</form>

<br>

<i>\*\*\* ถ้าไม่ป้อนข้อมูล โปรแกรมจะทำการแสดงผลทั้งหมด</i>

<!--End Term1-->

</body>

</html>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### แสดงโปรแกรม search4A\_study41\_code.php3

```
<html>
<body>

<?php

mysql_connect("localhost", "project" ,"project");
mysql_select_db("project");

$result = mysql_query("SELECT study_code, study_name, study_credit, study_daytime, study_room,
study_teacher
from study4A_T1 where study_code like '%$studycode4A1%' order by study_code");

if ($myrow = mysql_fetch_array($result))
{

echo "<table border=5>\n";
echo "<tr bgcolor=#f1b6c1> <td><center>รหัสวิชา</center></td> <td><center>ชื่อวิชา</center>
</td>
<td><center>หน่วยกิต</center></td> <td><center>วัน - เวลา</center></td>
<td><center>ห้องเรียน</center></td> <td><center>อาจารย์ผู้สอน</center></td>
</tr>\n";

do {
printf("<tr bgcolor=#ffffe0> <td>%s</td> <td>%s</td> <td>%s</td> <td>%s</td> <td>%s</td>
<td>%s</td> </tr>\n"
, $myrow["study_code"], $myrow["study_name"], $myrow["study_credit"], $myrow
["study_daytime"],
$myrow["study_room"], $myrow["study_teacher"]);
}
while ($myrow = mysql_fetch_array($result));

echo "</table>\n";
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else
{
echo "<center><b>จากการค้นหาปรากฏว่าไม่พบข้อมูลที่ป้อนมา</b></center>";
}
```

```
mysql_close();
```

```
?>
```

```
</body>
```

```
</html>
```

```
<b>แสดงรายชื่อวิชาเรียนปี 4A เทอม1 ทั้งหมด</b><br>
```

```
<form action=search4A_study4A1_all.php3 method="post">
```

```
<input type="submit" value="แสดง">
```

```
</form>
```

```
<br>
```

```
<i>*** ถ้าไม่ป้อนข้อมูล โปรแกรมจะทำการแสดงผลทั้งหมด</i>
```

```
<!--End Term1-->
```

```
</body>
```

```
</html>
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## แสดงโปรแกรม proj2541\_main.html

```
<html>
<head>
<title>ระบบฐานข้อมูลสารสนเทศของนักศึกษา ภาควิชาวิศวกรรมโทรคมนาคม ปี 4A</title>
</head>

<body text="Black" links="Blue" vlinks="Purple">

<!--Logo-->
<center>
<br>
</center>
<!--End Logo-->

<!--Project Search-->
<center><H2><font color="Green">รายชื่อ โครงการงานของนักศึกษาปี 2541</font></H2></center><br>

<form action=search4A_proj_projname.php3 method="post">
<b>ค้นหาโดยชื่อ โครงการงาน</b><br>
ป้อนชื่อ โครงการงาน :<input type="text" name="projname4A" size="50"><br>
<input type="submit" value="ค้นหา">
<input type="reset" value="ยกเลิก">
</form>

<br>

<form action=search4A_proj_stdname.php3 method="post">
<b>ค้นหาโดยชื่อนักศึกษา</b><br>
ป้อน ชื่อ - นามสกุล :<input type="text" name="stdname4A" size="30"><br>
<input type="submit" value="ค้นหา">
<input type="reset" value="ยกเลิก">
</form>

<br>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<form action=search4A_proj_id.php3 method="post">
```

```
<b>ค้นหา โดยเลขประจำตัวนักศึกษา</b><br>
```

```
ป้อนรหัสนักศึกษา :<input type="text" name="projid4A" size="9" maxlength="8"><br>
```

```
<input type="submit" value="ค้นหา">
```

```
<input type="reset" value="ยกเลิก">
```

```
</form>
```

```
<br>
```

```
<form action=search4A_proj_tcrname.php3 method="post">
```

```
<b>ค้นหาจากอาจารย์ที่ปรึกษา</b><br>
```

```
ป้อน ชื่ออาจารย์ที่ปรึกษา :<input type="text" name="projtcr4A" size="30"><br>
```

```
<input type="submit" value="ค้นหา">
```

```
<input type="reset" value="ยกเลิก">
```

```
</form>
```

```
<br>
```

```
<b>แสดงรายชื่อรายชื่อ โครงการปี 2541 ทั้งหมด</b><br>
```

```
<form action=search4A_projall.php3 method="post">
```

```
<input type="submit" value="แสดง">
```

```
</form>
```

```
<i>*** ถ้าไม่ป้อนข้อมูล โปรแกรมจะทำการแสดงผลทั้งหมด</i>
```

```
<!--End Project Search-->
```

```
</body>
```

```
</html>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### แสดงโปรแกรม search4A\_proj\_projname.php3

```
<html>
```

```
<body>
```

```
<?php
```

```
mysql_connect("localhost", "project", "project");
```

```
mysql_select_db("project");
```

```
$result = mysql_query("select student_name,student4A.id,project_name,teacher_name  
from student4A,project2541  
where (student4A.id=project2541.id) && (project2541.project_name like '%$projname4A%') order  
by project_name desc ");
```

```
if ($myrow = mysql_fetch_array($result))
```

```
{
```

```
echo "<table border=5>\n";
```

```
echo "<tr bgcolor=#f1b6c1> <td><center>ชื่อนักศึกษา</center></td>
```

```
<td><center>เลขประจำตัวนักศึกษา</center></td>
```

```
<td><center>ชื่อ  โครงการงาน</center></td>
```

```
<td><center>ชื่ออาจารย์ที่ปรึกษา</center></td></tr>\n";
```

```
do {
```

```
print("<tr bgcolor=#ffffe0> <td>%s</td> <td>%s</td> <td>%s</td> <td>%s</td>
```

```
</tr>\n",
```

```
$myrow["student_name"], $myrow["id"], $myrow["project_name"],
```

```
$myrow["teacher_name"]);
```

```
}
```

```
while ($myrow = mysql_fetch_array($result));
```

```
echo "</table>\n";
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else
```

```
{
```

```
echo "<center><b>จากการค้นหาปรากฏว่าไม่พบข้อมูลที่ป้อนมา</b></center>";
```

```
}
```

```
mysql_close();
```

```
?>
```

```
</body>
```

```
</html>
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### แสดงโปรแกรม search4A\_proj\_stdname.php3

```
<html>
<body>

<?php

mysql_connect("localhost", "project", "project");
mysql_select_db("project");

echo $proname4A;

$result = mysql_query("select student_name, student4A.id, project_name, teacher_name
    from student4A, project2541
    where (student4A.id=project2541.id) && (student_name like '%$stdname4A%') order by
project_name desc");

if ($myrow = mysql_fetch_array($result))
    {

    echo "<table border=5>\n";
    echo "<tr bgcolor=#f1b6c1> <td><center>ชื่อนักศึกษา</center></td>
        <td><center>เลขประจำตัวนักศึกษา</center></td>
        <td><center>ชื่อโครงการ</center></td>
        <td><center>ชื่ออาจารย์ที่ปรึกษา</center></td></tr>\n";

    do    {

        printf("<tr bgcolor=#ffffe0> <td>%s</td> <td>%s</td> <td>%s</td> <td>%s</td>
</tr>\n",

            $myrow["student_name"], $myrow["id"], $myrow["project_name"],
            $myrow["teacher_name"]);

    }

    while ($myrow = mysql_fetch_array($result));

    echo "</table>\n";
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else
{
echo "<center><b>จากการค้นหาปรากฏว่าไม่พบข้อมูลที่ป้อนมา</b></center>";
}
```

```
mysql_close();
```

```
?>
```

```
</body>
```

```
</html>
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### แสดงโปรแกรม search4A\_proj\_id.php3

```
<html>
<body>

<?php

mysql_connect("localhost", "project", "project");
mysql_select_db("project");

echo $projname4A;

$result = mysql_query("select student_name,student4A.id,project_name,teacher_name
    from student4A,project2541
    where (student4A.id=project2541.id) && (student4A.id like '%$projid4A%') order by project_name
    desc");

if ($myrow = mysql_fetch_array($result))
    {

    echo "<table border=5>\n";
    echo "<tr bgcolor=#ffb6c1> <td><center>ชื่อนักศึกษา</center></td>
        <td><center>เลขประจำตัวนักศึกษา</center></td>
        <td><center>ชื่อ โครงการ</center></td>
        <td><center>ชื่ออาจารย์ที่ปรึกษา</center></td></tr>\n";

    do    {

        printf("<tr bgcolor=#ffffe0> <td>%s</td> <td>%s</td> <td>%s</td> <td>%s</td>
</tr>\n",

            Smyrow["student_name"], Smyrow["id"], Smyrow["project_name"],
            Smyrow["teacher_name"]);

    }

    while ($myrow = mysql_fetch_array($result));

    echo "</table>\n";
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    else
    {
    echo "<center><b>จากการค้นหาปรากฏว่า ไม่พบข้อมูลที่ป้อนมา</b></center>";
    }

mysql_close();
?>

</body>
</html>
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### แสดงโปรแกรม search4A\_proj\_tcrname.php3

```
<html>
<body>

<?php

mysql_connect("localhost", "project", "project");
mysql_select_db("project");

echo $projname4A;

$result = mysql_query("select student_name, student4A.id, project_name, teacher_name
    from student4A, project2541
    where (student4A.id=project2541.id) && (teacher_name like '%$projtr4A%') order by project_name
    desc ");

if ($myrow = mysql_fetch_array($result))
    {

    echo "<table border=5>\n";
    echo "<tr bgcolor=#ffb6c1> <td><center>ชื่อนักศึกษา</center></td>
        <td><center>เลขประจำตัวนักศึกษา</center></td>
        <td><center>ชื่อโครงการ</center></td>
        <td><center>ชื่ออาจารย์ที่ปรึกษา</center></td></tr>\n";

    do    {

        printf("<tr bgcolor=#ffffe0> <td>%s</td> <td>%s</td> <td>%s</td> <td>%s</td>
</tr>\n",

            $myrow["student_name"], $myrow["id"], $myrow["project_name"],
            $myrow["teacher_name"]);

    }

    while ($myrow = mysql_fetch_array($result));

    echo "</table>\n";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}  
else  
{  
echo "<center><b>จากการค้นหาปรากฏว่าไม่พบข้อมูลที่ป้อนมา</b></center>";  
}
```

```
mysql_close();
```

```
?>
```

```
</body>
```

```
</html>
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## แสดงโปรแกรม teacher\_main.html

```
<html>
<head>
<title>ระบบฐานข้อมูลสารสนเทศของนักศึกษา ภาควิชาวิศวกรรมโทรคมนาคม ปี 4A</title>
</head>

<body text="Black" links="Blue" vlinks="Purple">

<!--Logo-->
<center>
<br>
</center>
<!--End Logo-->

<!--Teacher Search-->

<center><H2><font color="Green">รายชื่ออาจารย์</font></H2><br></center>

<form action=searchtcr_name.php3 method="post">
<b>ค้นหาโดยชื่ออาจารย์</b><br>
ป้อน ชื่อ - นามสกุล :<input type="text" name="tcname" size="30"><br>
<input type="submit" value="ค้นหา">
<input type="reset" value="ยกเลิก">
</form>

<br>

<form action=searchtcr_room.php3 method="post">
<b>ค้นหาจากห้องทำงานอาจารย์</b><br>
ป้อนห้องทำงานอาจารย์ :<input type="text" name="tcrroom" size="5" maxlength="5"><br>
<input type="submit" value="ค้นหา">
<input type="reset" value="ยกเลิก">
</form>

<br>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>แสดงรายชื่ออาจารย์ทั้งหมด</b><br>

<form action=searchtcr\_all.php3 method="post">

<input type="submit" value="แสดง">

</form>

<br>

<i>\*\*\* ถ้าไม่ป้อนข้อมูล โปรแกรมจะทำการแสดงผลทั้งหมด</i>

<!--End Teacher Search-->

</body>

</html>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### แสดงโปรแกรม searchter\_name.php3

```
<html>
<body>

<?php
mysql_connect("localhost", "project", "project");
mysql_select_db("project");

$result = mysql_query("SELECT teacher_name, teacher_room
    from teacher where teacher_name like '%$stername%' order by teacher_room");

if ($myrow = mysql_fetch_array($result))
    {
        echo "<table border=5>\n";
        echo "<tr bgcolor=#ffb6c1> <td><center>ชื่ออาจารย์</center></td> <td><center>ห้องทำงานอาจารย์</center></td> </tr>\n";

        do {
            printf("<tr bgcolor=#ffffe0> <td>%s</td> <td>%s</td></tr>\n", $myrow
["teacher_name"], $myrow["teacher_room"]);
        }
        while ($myrow = mysql_fetch_array($result));

        echo "</table>\n";
    }
else
    {
        echo "<center><b>จากการค้นหาปรากฏว่าไม่พบข้อมูลที่ป้อนมา</b></center>";
    }

mysql_close();
?>

</body>
</html>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### แสดงโปรแกรม searchctr\_room.php3

```
<html>
<body>

<?php
mysql_connect("localhost", "project", "project");
mysql_select_db("project");

$result = mysql_query("SELECT teacher_name,teacher_room
    from teacher where teacher_room like '%$stcrroom%' order by teacher_room");
if ($myrow = mysql_fetch_array($result))
    {

    echo "<table border=5>\n";
    echo "<tr bgcolor=#ffb6c1> <td><center>ชื่ออาจารย์</center></td> <td><center>ห้องทำงานอาจารย์
</center></td> </tr>\n";

    do    {
        printf("<tr bgcolor=#ffffe0> <td>%s</td> <td>%s</td></tr>\n", $myrow
["teacher_name"], $myrow["teacher_room"]);
    }
    while ($myrow = mysql_fetch_array($result));

    echo "</table>\n";
    }
    else
    {
    • echo "<center><b>จากการค้นหาปรากฏว่าไม่พบข้อมูลที่ป้อนมา</b></center>";
    }

mysql_close();
?>
</body>
</html>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงโปรแกรม std2540\_main.html

```
<html>
<head>
<title>ระบบฐานข้อมูลสารสนเทศของนักศึกษา ภาควิชาวิศวกรรมโทรคมนาคม ปี 2538</title>
</head>

<body text="Black" links="Blue" vlinks="Purple">

<!--Logo-->
<center>
<br>
</center>
<!--End Logo-->

<!--Project Search-->
<center><H2><font color="Green">รายชื่อ โครงการงานของนักศึกษาปี 2538</font></H2></center><br>

<form action=search2538_proj_projname.php3 method="post">
<b>ค้นหาโดยชื่อ โครงการงาน</b><br>
ป้อนชื่อ โครงการงาน :<input type="text" name="projname2538" size="50"><br>
<input type="submit" value="ค้นหา">
<input type="reset" value="ยกเลิก">
</form>

<br>

<form action=search2538_proj_stdname.php3 method="post">
<b>ค้นหาโดยชื่อนักศึกษา</b><br>
ป้อน ชื่อ - นามสกุล :<input type="text" name="stdname2538" size="30"><br>
<input type="submit" value="ค้นหา">
<input type="reset" value="ยกเลิก">
</form>

<br>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<form action=search2538_proj_id.php3 method="post">
<b>ค้นหาโดยเลขประจำตัวนักศึกษา</b><br>
ป้อนรหัสนักศึกษา :<input type="text" name="projid2538" size="9" maxlength="8"><br>
<input type="submit" value="ค้นหา">
<input type="reset" value="ยกเลิก">
</form>
```

<br>

```
<form action=search2538_proj_tcname.php3 method="post">
<b>ค้นหาจากอาจารย์ที่ปรึกษา</b><br>
ป้อนชื่ออาจารย์ที่ปรึกษา:<input type="text" name="projtcr2538" size="30"><br>
<input type="submit" value="ค้นหา">
<input type="reset" value="ยกเลิก">
</form>
```

<br>

```
<b>แสดงรายชื่อรายชื่อ โครงการปี 2538 ทั้งหมด</b><br>
<form action=search2538_projall.php3 method="post">
<input type="submit" value="แสดง">
</form>
```

<i>\*\*\* ถ้าไม่ป้อนข้อมูล โปรแกรมจะทำการแสดงผลทั้งหมด</i>

<!--End Project Search-->

</body>

</html>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### แสดงโปรแกรม search2538\_name.php3

```
<html>
<body>
<?php
mysql_connect("localhost", "project", "project");
mysql_select_db("project");

$result = mysql_query("SELECT student_name, id
    from student2538 where student_name like '%$name2538%' order by id");

if ($myrow = mysql_fetch_array($result))
    {
        echo "<table border=5>\n";
        echo "<tr bgcolor=#ffb6c1> <td><center>ชื่อนักศึกษา</center></td> <td><center>เลขประจำตัวนัก
ศึกษา</center></td> </tr>\n";

        do    {
            printf("<tr bgcolor=#ffffe0> <td>%s</td> <td>%s</td></tr>\n", $myrow
["student_name"], $myrow["id"]);
        }
        while ($myrow = mysql_fetch_array($result));

        echo "</table>\n";
    }
    else
    {
        echo "<center><b>จากการค้นหาปรากฏว่าไม่พบข้อมูลที่ป้อนมา</b></center>";
    }

mysql_close();
?>
</body>
</html>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### แสดงโปรแกรม search2538\_id.php3

```
<html>
<body>

<?php

mysql_connect("localhost", "project", "project");
mysql_select_db("project");

$result = mysql_query("SELECT student_name,id
    from student2538 where id like '%" . $id2538 . "%' order by id");

if ($myrow = mysql_fetch_array($result))
    {
        echo "<table border=5>\n";
        echo "<tr bgcolor=#ffb6c1> <td><center>ชื่อนักศึกษา</center></td> <td><center>เลขประจำตัวนัก
ศึกษา</center></td> </tr>\n";
        do
            {
                printf("<tr bgcolor=#ffffe0> <td>%s</td> <td>%s</td></tr>\n", $myrow
["student_name"], $myrow["id"]);
            }
        while ($myrow = mysql_fetch_array($result));
        echo "</table>\n";
    }
    else
    {
        echo "<center><b>จากการค้นหาปรากฏว่า ไม่พบข้อมูลที่ป้อนมา</b></center>";
    }

mysql_close();
?>
</body>
</html>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## mysql\_affected\_rows

### Name

mysql\_affected\_rows — get number of affected rows in last query

### Description

```
int mysql_affected_rows(int [link_identifier] );
```

Returns: The number of affected rows by the last query.

mysql\_affected\_rows returns the number of rows affected by the last INSERT, UPDATE or DELETE query on the server associated with the specified link identifier. If the link identifier isn't specified, the last opened link is assumed.

This command is not effective for SELECT statements, only on statements which modify records. To retrieve the number of rows returned from a SELECT, use mysql\_num\_rows.

## mysql\_close

### Name

mysql\_close — close MySQL connection

### Description

```
int mysql_close(int [link_identifier] );
```

Returns: true on success, false on error

mysql\_close closes the link to a MySQL database that's associated with the specified link identifier. If the link identifier isn't specified, the last opened link is assumed.

Note that this isn't usually necessary, as non-persistent open links are automatically closed at the end of the script's execution.

mysql\_close will not close persistent links generated by mysql\_pconnect().

See also: mysql\_connect, and mysql\_pconnect.

## mysql\_connect

### Name

`mysql_connect` — open MySQL server connection

### Description

```
int mysql_connect(string [hostname] , string [username] , string [password]
);
```

Returns: A positive MySQL link identifier on success, or false on error.

`mysql_connect` establishes a connection to a MySQL server. All of the arguments are optional, and if they're missing, defaults are assumed ('localhost', user name of the user that owns the server process, empty password). The hostname string can also include a port number. eg. "hostname:port"

In case a second call is made to `mysql_connect` with the same arguments, no new link will be established, but instead, the link identifier of the already opened link will be returned.

The link to the server will be closed as soon as the execution of the script ends, unless it's closed earlier by explicitly calling `mysql_close`.

See also `mysql_pconnect`, and `mysql_close`.

## mysql\_create\_db

### Name

`mysql_create_db` — create MySQL database

### Description

```
int mysql_create_db(string database name, int [link_identifier] );
```

`mysql_create_db` attempts to create a new database on the server associated with the specified link identifier.

See also: `mysql_drop_db`. For downwards compatibility `mysql_createdb` can also be used.

## mysql\_data\_seek

### Name

`mysql_data_seek` — move internal row pointer

### Description

```
int mysql_data_seek(int result_identifier, int row_number);
```

Returns: true on success, false on failure

`mysql_data_seek` moves the internal row pointer of the MySQL result associated with the specified result identifier to point to the specified row number. The next call to `mysql_fetch_row` would return that row.

See also: `mysql_data_seek`.

## mysql\_dbname

### Name

`mysql_dbname` — get current MySQL database name

### Description

```
string mysql_dbname(string result, int i);
```

`mysql_dbname` returns the database name stored in position `i` of the result pointer returned from the `mysql_list_dbs` function. The `mysql_num_rows` function can be used to determine how many database names are available.

## mysql\_db\_query

### Name

`mysql_db_query` — send MySQL query

### Description

```
int mysql_db_query(string database, string query, int link_identifier);
```

Returns: A positive MySQL result identifier to the query result, or false on error.

`mysql_db_query` selects a database and executes a query on it. If the optional link identifier isn't specified, the function will try to find an open link to the MySQL server and if no such link is found it'll try to create one as if `mysql_connect` was called with no arguments

See also `mysql_connect`. For downwards compatibility `mysql` can also be used.

## mysql\_drop\_db

### Name

`mysql_drop_db` — drop (delete) MySQL database

### Description

```
int mysql_drop_db(string database_name, int [link_identifier] );
```

Returns: true on success, false on failure.

`mysql_drop_db` attempts to drop (remove) an entire database from the server associated with the specified link identifier.

See also: `mysql_create_db`. For downward compatibility `mysql_dropdb` can also be used.

## mysql\_errno

### Name

`mysql_errno` — returns error number of last mysql call

### Description

```
int mysql_errno();
```

Errors coming back from the mySQL database backend no longer issue warnings. Instead, use these functions to retrieve the error number.

```
<?php
mysql_connect("marliesle");
echo mysql_errno().": ".mysql_error()."<BR>";
mysql_select_db("nonexistentdb");
echo mysql_errno().": ".mysql_error()."<BR>";
$conn = mysql_query("SELECT * FROM nonexistenttable");
echo mysql_errno().": ".mysql_error()."<BR>";
?>
```

See also: `mysql_error`

## mysql\_error

### Name

`mysql_error` — returns error message of last mysql call

### Description

```
string mysql_error();
```

Errors coming back from the MySQL database backend no longer issue warnings. Instead, use these functions to retrieve the error string.

```
<?php
mysql_connect("marliesle");
echo mysql_errno().": ".mysql_error()."<BR>";
mysql_select_db("nonexistentdb");
echo mysql_errno().": ".mysql_error()."<BR>";
$conn = mysql_query("SELECT * FROM nonexistenttable");
echo mysql_errno().": ".mysql_error()."<BR>";
?>
```

See also: `mysql_errno`

## mysql\_fetch\_array

### Name

`mysql_fetch_array` — fetch row as array

### Description

```
array mysql_fetch_array(int result);
```

Returns: An array that corresponds to the fetched row, or false if there are no more rows.

`mysql_fetch_array` is an extended version of `mysql_fetch_row`. In addition to storing the data in the numeric indices of the result array, it also stores the data in associative indices, using the field names as keys.

An important thing to note is that using `mysql_fetch_array` is NOT significantly slower than using `mysql_fetch_row`, while it provides a significant added value.

For further details, also see `mysql_fetch_row`

#### Example 1. mysql fetch array

```
<?php
mysql_connect($host,$user,$password);
$result = mysql_db_query("database","select * from table");
while($row = mysql_fetch_array($result)) {
```

```

    echo $row["user_id"];
    echo $row["fullname"];
}
mysql_free_result($result);
?>

```

## mysql\_fetch\_field

### Name

`mysql_fetch_field` — get field information

### Description

object `mysql_fetch_field(int result, int [field_offset] );`

Returns an object containing field information.

`mysql_fetch_field` can be used in order to obtain information about fields in a certain query result. If the field offset isn't specified, the next field that wasn't yet retrieved by `mysql_fetch_field` is retrieved.

The properties of the object are:

- `name` - column name
- `table` - name of the table the column belongs to
- `max_length` - maximum length of the column
- `not_null` - 1 if the column cannot be null
- `primary_key` - 1 if the column is a primary key
- `unique_key` - 1 if the column is a unique key
- `multiple_key` - 1 if the column is a non-unique key
- `numeric` - 1 if the column is numeric
- `blob` - 1 if the column is a BLOB
- `type` - the type of the column
- `unsigned` - 1 if the column is unsigned
- `zerofill` - 1 if the column is zero-filled

See also `mysql_field_seek`

## mysql\_fetch\_lengths

### Name

`mysql_fetch_lengths` — get max data size of output columns

### Description

```
int mysql_fetch_lengths(int result);
```

Returns: An array that corresponds to the lengths of each field in the last row fetched by `mysql_fetch_row`, or false on error.

`mysql_fetch_lengths` stores the lengths of each result column in the last row returned by `mysql_fetch_row` in an array, starting at offset 0.

See also: `mysql_fetch_row`.

## mysql\_fetch\_object

### Name

`mysql_fetch_object` — fetch row as object

### Description

```
int mysql_fetch_object(int result);
```

Returns: An object with properties that correspond to the fetched row, or false if there are no more rows.

`mysql_fetch_object` is similar to `mysql_fetch_array`, with one difference - an object is returned, instead of an array. Indirectly, that means that you can only access the data by the field names, and not by their offsets (numbers are illegal property names).

Speed-wise, the function is identical to `mysql_fetch_array`, and almost as quick as `mysql_fetch_row` (the difference is insignificant).

See also: `mysql_fetch_array` and `mysql_fetch_row`.

## mysql\_fetch\_row

### Name

`mysql_fetch_row` — get row as enumerated array

### Description

```
array mysql_fetch_row(int result);
```

Returns: An array that corresponds to the fetched row, or false if there are no more rows.

`mysql_fetch_row` fetches one row of data from the result associated with the specified result identifier. The row is returned as an array. Each result column is stored in an array offset, starting at offset 0.

Subsequent call to `mysql_fetch_row` would return the next row in the result set, or false if there are no more rows.

See also: `mysql_fetch_array`, `mysql_fetch_object`, `mysql_data_seek`, `mysql_fetch_lengths`, and `mysql_result`.

## mysql\_field\_name

### Name

`mysql_field_name` — get field name

### Description

```
string mysql_field_name(string result, int i);
```

`mysql_field_name` returns the name of the specified field. Arguments to the function is the result identifier and the field index, ie. `mysql_field_name($result, 2)`;

Will return the name of the second field in the result associated with the result identifier.

For downwards compatibility `mysql_fieldname` can also be used.

## mysql\_field\_seek

### Name

`mysql_field_seek` — set field offset

### Description

```
int mysql_field_seek(int result, int field_offset);
```

Seeks to the specified field offset. If the next call to `mysql_fetch_field` won't include a field offset, this field would be returned.

See also: `mysql_fetch_field`.

## mysql\_field\_table

### Name

`mysql_field_table` — get table name for field

### Description

```
string mysql_field_table(int result, int field_offset);
```

Get the table name for field. For downward compatibility `mysql_fieldtable` can also be used.

## mysql\_field\_type

### Name

`mysql_field_type` — get field type

### Description

```
string mysql_field_type(string result, int field_offset);
```

`mysql_field_type` is similar to the `mysql_field_name` function. The arguments are identical, but the field type is returned. This will be one of "int", "real", "string", or "blob".

#### Example 1. mysql field types

```
<?php
mysql_connect("localhost:3306");
mysql_select_db("wisconsin");
$result = mysql_query("SELECT * FROM onek");
$fields = mysql_num_fields($result);
$rows   = mysql_num_rows($result);
$i = 0;
```

```

$stable = mysql_field_table($result, $i);
echo "Your '". $stable.'" table has ".$fields." fields and ".$rows." records
<BR>";
echo "The table has the following fields <BR>";
while ($i < $fields) {
    $stype = mysql_field_type ($result, $i);
    $name = mysql_field_name ($result, $i);
    $len = mysql_field_len ($result, $i);
    $flags = mysql_field_flags ($result, $i);
    echo $stype." ". $name." ". $len." ". $flags."<BR>";
    $i++;
}
mysql_close();
?>

```

For downward compatibility `mysql_fieldtype` can also be used.

## mysql\_field\_flags

### Name

`mysql_field_flags` — get field flags

### Description

```
string mysql_field_flags(string result, int field_offset);
```

`mysql_field_flags` returns the field flags of the specified field. The flags are reported as a single word per flag separated by a single space, so that you can split the returned value using `explode`.

The following flags are reported, if your version of MySQL is current enough to support them: "not\_null", "primary\_key", "unique\_key", "multiple\_key", "blob", "unsigned", "zerofill", "binary", "enum", "auto\_increment", "timestamp".

For downward compatibility `mysql_fieldflags` can also be used.

## mysql\_field\_len

### Name

`mysql_field_len` — get field length

### Description

```
int mysql_field_len(string result, int field_offset);
```

`mysql_field_len` returns the length of the specified field. For downward compatibility `mysql_fieldlen` can also be used.

## mysql\_free\_result

### Name

`mysql_free_result` — free result memory

### Description

```
int mysql_free_result(int result);
```

`mysql_free_result` only needs to be called if you are worried about using too much memory while your script is running. All associated result memory for the specified result identifier will automatically be freed.

For downward compatibility `mysql_freeresult` can also be used.

## mysql\_insert\_id

### Name

`mysql_insert_id` — get generated id from last INSERT

### Description

```
int mysql_insert_id(void);
```

`mysql_insert_id` returns the ID generated for an `AUTO_INCREMENTED` field. This function takes no arguments. It will return the auto-generated ID returned by the last `INSERT` query performed.

## mysql\_list\_fields

### Name

`mysql_list_fields` — list result fields

### Description

```
int mysql_list_fields(string database, string tablename);
```

`mysql_list_fields` retrieves information about the given `tablename`. Arguments are the database name and the table name. A result pointer is returned which can be used with `mysql_field_flags`, `mysql_field_len`, `mysql_field_name`, and `mysql_field_type`.

A result identifier is a positive integer. The function returns -1 if a error occurs. A string describing the error will be placed in `$phperrmsg`, and unless the function was called as `@mysql()` then this error string will also be printed out.

For downward compatibility `mysql_listfields` can also be used.

## mysql\_list\_dbs

### Name

`mysql_list_dbs` — list MySQL databases on server

### Description

```
int mysql_listdbs(void);
```

`mysql_listdbs` will return a result pointer containing the databases available from the current `mysql` daemon. Use the `mysql_dbname` function to traverse this result pointer.

For downward compatibility `mysql_listdbs` can also be used.

## mysql\_list\_tables

### Name

`mysql_list_tables` — list tables in a MySQL database

### Description

```
int mysql_list_tables(string database);
```

`mysql_list_tables` takes a database name and result pointer much like the `mysql_db_query` function. The `mysql_tablename` function should be used to extract the actual table names from the result pointer.

For downward compatibility `mysql_listtables` can also be used.

## mysql\_num\_fields

### Name

`mysql_num_fields` — get number of fields in result

### Description

```
int mysql_num_fields(int result);
```

`mysql_num_fields` returns the number of fields in a result set.

See also: `mysql_db_query`, `mysql_query`, `mysql_fetch_field`, `mysql_num_rows`.

For downward compatibility `mysql_numfields` can also be used.

## mysql\_num\_rows

### Name

`mysql_num_rows` — get number of rows in result

### Description

```
int mysql_num_rows(string result);
```

`mysql_num_rows` returns the number of rows in a result set.

See also: `mysql_db_query`, `mysql_query` and `mysql_fetch_row`.

For downward compatibility `mysql_numrows` can also be used.

## mysql\_pconnect

### Name

`mysql_pconnect` — open persistent MySQL connection

### Description

```
int mysql_pconnect(string [hostname] , string [username] , string [password]
);
```

Returns: A positive MySQL persistent link identifier on success, or false on error

`mysql_pconnect` acts very much like `mysql_connect` with two major differences.

First, when connecting, the function would first try to find a (persistent) link that's already open with the same host, username and password. If one is found, an identifier for it will be returned instead of opening a new connection.

Second, the connection to the SQL server will not be closed when the execution of the script ends. Instead, the link will remain open for future use (`mysql_close` will not close links established by `mysql_pconnect()`).

This type of links is therefore called 'persistent'.

## mysql\_query

### Name

`mysql_query` — send MySQL query

### Description

```
int mysql_query(string query, int [link_identifier] );
```

Returns: A positive MySQL result identifier on success, or false on error.

`mysql_query` sends a query to the currently active database on the server that's associated with the specified link identifier. If the link identifier isn't specified, the last opened link is assumed. If no link is open, the function tries to establish a link as if `mysql_connect` was called, and use it.

See also: `mysql_db_query`, `mysql_select_db`, and `mysql_connect`.

## mysql\_result

### Name

`mysql_result` — get result data

### Description

```
int mysql_result(int result, int row, mixed field);
```

Returns: The contents of the cell at the row and offset in the specified MySQL result set.

`mysql_result` returns the contents of one cell from a MySQL result set. The field argument can be the field's offset, or the field's name, or the field's table dot field's name (fieldname.tablename). If the column name has been aliased ('select foo as bar from...'), use the alias instead of the column name.

When working on large result sets, you should consider using one of the functions that fetch an entire row (specified below). As these functions return the contents of multiple cells in one function call, they're MUCH quicker than `mysql_result`. Also, note that specifying a numeric offset for the field argument is much quicker than specifying a fieldname or tablename.fieldname argument.

Recommended high-performance alternatives: `mysql_fetch_row`, `mysql_fetch_array`, and `mysql_fetch_object`.

## mysql\_select\_db

### Name

`mysql_select_db` — select MySQL database

### Description

```
int mysql_select_db(string database_name, int [link_identifier] );
```

Returns: true on success, false on error

`mysql_select_db` sets the current active database on the server that's associated with the specified link identifier. If no link identifier is specified, the last opened link is assumed. If no link is open, the function will try to establish a link as if `mysql_connect` was called, and use it.

Every subsequent call to `mysql_query` will be made on the active database.

See also: `mysql_connect`, `mysql_pconnect`, and `mysql_query`

For downward compatibility `mysql_selectdb` can also be used.

## mysql\_tablename

### Name

`mysql_tablename` — get table name of field

### Description

```
string mysql_tablename(int result, int i);
```

`mysql_tablename` takes a result pointer returned by the `mysql_list_tables` function as well as an integer index and returns the name of a table. The `mysql_num_rows` function may be used to determine the number of tables in the result pointer.

#### Example 1. mysql\_tablename() example

```
<?php
mysql_connect ("localhost:3306");
$result = mysql_listtables ("wisconsin");
$i = 0;
while ($i < mysql_num_rows ($result)) {
    $tb_names[$i] = mysql_tablename ($result, $i);
    echo $tb_names[$i] . "<BR>";
    $i++;
}
?>
```

## กิตติกรรมประกาศ

ขอขอบพระคุณผู้ให้การช่วยเหลือและสนับสนุนดังต่อไปนี้

อมริศา มณีรุ่ง

เสกสรรค์ แร่ทอง

ยุทธชัย อินปา

นพพร มะหิงสิป

ผู้มีพระคุณทุกท่านที่มีได้กล่าวถึง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## หนังสืออ้างอิง

1. J.Harvey Trimble Jr, "A VISUAL INTRODUCTON TO SQL", John Wiley & Sons, Ind.1989
2. James R.Groff and Paul N. Weiberg, " LAN TIMES GUIDE TO SQL", Osborne McGraw-Hill 1994
3. Kamran Husain and Tim Parker, "LINUX UNLEASHED", Sams Publishing 1995
4. Jessisa Perry Hekman, "LINUX IN A NUTSHELL", O'Reilly & Associates, Inc.1997
5. Bruce Morris, "HTML IN ACTION", Microsoft Press 1996
6. สมจิตร อางอินทร์, "ระบบฐานข้อมูล", ขอนแก่นการพิมพ์,2541
7. กิตติ ภักดีวิวัฒนะกุล, "สร้าง Web Page แบบมืออาชีพด้วย HTML", ไทยเจริญการพิมพ์,2540
8. วีระ บุญจริง, "หลักการเขียนโปรแกรม", ดวงกมลสมัย ,2539



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้