สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบบริหารการเงินภาควิชาวิศวกรรมคอมพิวเตอร์

Department's Financial Management System

นายธีระ พันธุ์พาณิชย์

นายวานิจ ศรีรักษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เลขหมู่.....
เลขทะเบียน 37081
วัน, เดือน, ปี 30 ส.ค. 2543

ระบบบริหารการเงินภาควิชาวิศวกรรมคอมพิวเตอร์

Department's Financial Management System

โดย

นายธีระ พันธุ์พาณิชย์

นายวานิจ ศรีรักษ์

อาจารย์ที่ปรึกษา

อาจารย์ธนา หงษ์สุวรรณ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

ปริญญานิพนธ์ปีการศึกษา 2542

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบบริหารการเงินภาควิชาวิศวกรรมคอมพิวเตอร์

Department's Financial Management System

ผู้จัดทำ

1.นายธีระ พันธุ์พาณิชย์ รหัสประจำตัว 40013253

2.นายวานิจ ศรีรักษ์ รหัสประจำตัว 40013267

อาจารย์ที่ปรึกษา

(อาจารย์ธนา หงษ์สุวรรณ)

ระบบบริหารการเงินภาควิชาวิศวกรรมคอมพิวเตอร์

นายธีระ พันธุ์พาณิชย์ 4013253
นายวานิจ ศรีรักษ์ 40013267
อาจารย์ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
ปีการศึกษา 2542

บทคัดย่อ

โครงงานระบบบริหารการเงินภาควิชาวิศวกรรมคอมพิวเตอร์เป็นส่วนหนึ่งของโครงงานระบบสารสนเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระเจ้าเกล้า เจ้าคุณทหารลาดกระบัง ซึ่งเกี่ยวกับการจัดการฐานข้อมูลภายในภาควิชาทำให้การทำงานและบริหารงานของภาควิชา เป็นระบบและมีประสิทธิภาพยิ่งขึ้น

ระบบบริหารงานของภาควิชาสามารถช่วยผู้ใช้ในด้านต่างๆ ต่อไปนี้ จัดทำรายงาน, จัดการงบประมาณของภาควิชา รวมทั้งจัดเก็บและอำนวยความสะดวกในการเรียกดูข้อมูลฯ ต่าง ในภาควิชา เช่น ข้อมูลเกี่ยวกับรายละเอียดรายรับ – รายจ่าย, รายละเอียดเกี่ยวกับยอดเงินคงเหลือ เป็นค้น สำหรับซอฟแวร์ที่ใช้ในการพัฒนาระบบใช้โปรแกรม Delphi ในส่วนของฟรอน์ดเอนท์ (Font End) และใช้แบคเอนท์ (MS SQL Server) ในส่วนของ (Back End) ซึ่งเป็นโปรแกรมจัดการฐานข้อมูลที่มีเครื่องมือที่สามารถช่วยในการพัฒนาระบบได้อย่างมีประสิทธิภาพ

Department 's Financial Management System

Teera Punpanid

Wanit Srirak

Thana Hongsuwan Advisor

ABSTRACT

"Department 's Financial Management System" is one project that comprises of Information System Project in Engineering Faculty of Kingmongkut Institute of Technology Ladkrabang (KMITL). This project is about management in individual department (control management is at faculty). We can use it as tool to manage department information formally and efficiently.

Department 's Financial Management System has many helpful abilities for users. It can manage data that users can retrive result of instructor 's information. Meterial funds and material orders listing. This system was implemented by Delphi (part of font end) and MS SQL Server (part of back end) which has many powerful tools for system developing.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ไม่อาจเสร็จไปได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลายๆฝ่ายด้วยกัน ไม่ว่าจะเป็นอาจารย์ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษาปริญญานิพนธ์ อาจารย์นวพรวรรณวิมลศรี รวมทั้งเจ้าหน้าที่ธุรการของภาควิชาวิชาวิศวกรรมคอมพิวเตอร์ทุกท่าน ที่ให้คำแนะนำ และช่วยเหลือเสมอมา

ทางคณะผู้จัดทำขอถือโอกาสนี้ขอบพระคุณมายังทุกท่านเป็นอย่างมากที่ได้ให้ความอนุเคราะห์ช่วยเหลือในด้านต่างๆ มา ณ ที่นี้ด้วย

ธีระ พันธุ์พาณิชย์

นายวานิจ ศรีรักษ์

สารบัญ

หน้าที่

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VIII

สารบัญภาพ IX

บทที่ 1 บทนำ 1

บทที่ 2 หลักการและทฤษฎี 4

2.1 หลักการของระบบฐานข้อมูล (Database) 4

2.2 ลักษณะของฐานข้อมูลที่ดี 4

2.3 รีเลชั่นนอลโมเดล (Relational Model) 5

2.3.1 ศัพท์เฉพาะของรีเลชั่นนอลโมเดล 6

2.3.2 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง 8

2.3.3 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม 8

2.3.4 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม 8

2.4 ทฤษฎีของระบบฐานข้อมูลแบบรีเลชั่นนอล 9

2.4.1 กฎของความถูกต้อง (Integrity Rule) 9

2.4.2 ฟังก์ชันการขึ้นต่อกัน (Functional Dependency) 10

2.4.3 การนอร์มัลไลซ์ (Normalization) 11

2.5 การออกแบบฐานข้อมูลโดยใช้วิธีไนแอม 13

2.6 หลักการของไคลเอ็นท์เซอร์ฟเวอร์ 16

2.6.1 ความรู้พื้นฐานเกี่ยวกับระบบไคล์เอนท์เซิร์ฟเวอร์ 17

2.6.2 ข้อดีของระบบไคลเอ็นท์เซอร์ฟเวอร์ 20

2.6.3 ข้อเสียของระบบฐานข้อมูลไคลเอ็นท์เซอร์ฟเวอร์ 20

2.7 โอเพนดาต้าเบสคอนเน็คติวิตี (โอดีบีซี) (Open Database Connectivity, ODBC) 21

2.7.1 ข้อดีของการติดต่อโดยใช้โอดีบีซี 23

2.7.2 การติดต่อโดยใช้โอดีบีซี (ODBC Interface) 23

2.7.3 ส่วนประกอบของโอดีบีซี 24

2.8 Borland Delphi 25

2.8.1 คุณสมบัติต่างๆที่มีอยู่ใน Delphi 25

2.8.2 สิ่งสำคัญในการเขียนโปรแกรมเชิงวัตถุ 26

สารบัญ

หน้าที่

2.8.3 คอมไพเลอร์ที่แท้จริง 27
2.8.4 คอมโพเนนต์ไลบรารี 27
2.8.5 OCX และ ActiveX 27
2.8.6 Wizard และ Object Repository 27
2.8.7 การติดต่อฐานข้อมูล 28
2.8.8 แบบต่างๆของ Delphi 28
2.8.9 โปรแกรมและเครื่องมือในเดลไฟ 4 28
2.8.10 สภาพแวดล้อมของ Delphi 29
2.8.11 BDE (Borland Database Engine) 34
2.9 ความรู้เกี่ยวกับเอสคิวแอลเซิร์ฟเวอร์ (SQL Server) 35
2.9.1 โครงสร้างข้อมูลในเอสคิวแอลเซิร์ฟเวอร์ 36
2.9.2 ซิสเต็ม-แอดมินิสเตเตอร์ – เอสเอ 37
2.9.3 ซิสเต็มเทเบิ้ล 37
2.9.4 ชนิดข้อมูล 38
2.9.5 วีว 39
2.9.6 เอวคิวแอลอินเด็กซ์ 39
2.9.7 ทรานสแอคชั่น-เอสคิวแอล 40
2.9.8 สถาปัตยกรรมระบบฐานข้อมูลของ SQL Server 40
2.9.9 ฐานข้อมูลของระบบ SQL Server 41
2.9.10 คุณสมบัติใหม่ๆที่เพิ่มเข้ามาใน SQL Server 7.0 45
บทที่ 3 การออกแบบโปรแกรม 47
3.1 ระบบงานด้านการเงินภาควิชาวิศวกรรมคอมพิวเตอร์ 47
3.1.1 บัญชีรายรับ 47
3.1.2 บัญชีรายจ่าย 49
3.2 การออกแบบฐานข้อมูล 53
3.2.1 เทเบิลเก็บข้อมูลยอดค้างชำระค่าปริ้นเตอร์ 54
3.2.2 เทเบิลเก็บข้อมูลบริษัท 54
3.2.3 เทเบิลเก็บข้อมูลประเภทเงินงบประมาณ 54
3.2.4 เทเบิลเก็บข้อมูลชื่อประเภทเงินงบประมาณ 55
3.2.5 เทเบิลเก็บข้อมูลการใช้เงินงบประมาณในใบตั้งเบิก 55
3.2.6 เทเบิลเก็บข้อมูลใบตั้งเบิก 55

สารบัญ

หน้าที่

3.2.7 เทเบิลเก็บข้อมูลใบเสร็จต่างๆในใบตั้งเบิก 56
3.2.8 เทเบิลเก็บข้อมูลใบเสร็จ 56
3.2.9 เทเบิลเก็บข้อมูลรายจ่ายจากเงินค่าชีต 57
3.2.10 เทเบิลเก็บข้อมูลรายจ่ายจากเงินค่าปริ้นเตอร์ 57
3.2.11 เทเบิลเก็บข้อมูลรายจ่ายจากเงิน 5 เปอร์เซ็นต์ 57
3.2.12 เทเบิลเก็บข้อมูลใบเสร็จของบริษัทต่างๆ (Punch Table) 58
3.2.13 เทเบิลเก็บข้อมูลใบเสร็จของบริษัทต่างๆและรายการต่างๆในใบเสร็จ 58
3.2.14 เทเบิลเก็บข้อมูลรายรับอื่นๆ 58
3.2.15 เทเบิลเก็บข้อมูลรายรับเงินงบประมาณภาควิชา 59
3.2.16 เทเบิลเก็บข้อมูลรายรับเงิน 5 เปอร์เซ็นต์ 59
3.2.17 เทเบิลเก็บข้อมูลรายรับเงินค่าปริ้นเตอร์ 59
3.2.18 เทเบิลเก็บข้อมูลรายรับเงินค่าชีต 60
3.2.19 เทเบิลเก็บข้อมูลอาจารย์ 60
3.2.20 เทเบิลเก็บข้อมูลการโอนย้ายเงินระหว่างงบประมาณ 60
3.3 การหาผลสรุปของข้อมูลจากเทเบิลต่างๆ 61
3.3.1 ยอดรวมของเงินรายได้จากงบประมาณภาควิชา 61
3.3.2 ยอดรวมของเงินรายได้จากการเก็บเงิน 5 เปอร์เซ็นต์ 61
3.3.3 ยอดรวมของเงินรายได้จากการเก็บเงินค่าปริ้นเตอร์ 61
3.3.4 ยอดรวมของเงินรายได้จากการเก็บเงินค่าชีต 62
3.3.5 ยอดรวมของเงินจ่ายที่ใช้โดยเงินค่าชีต 62
3.3.6 ยอดรวมของเงินจ่ายที่ใช้โดยเงินค่าปริ้นเตอร์ 62
3.3.7 ยอดรวมของเงินจ่ายที่ใช้โดยเงิน 5 เปอร์เซ็นต์ 62
3.3.8 ยอดรวมของเงินจ่ายที่ใช้โดยเงินงบประมาณภาควิชา 62
3.4 การออกแบบส่วนติดต่อกับผู้ใช้งานโปรแกรม 63
3.4.1 เมนูหลัก 63
3.4.2 เมนูย่อย 63
3.4.3 หน้าจอรายรับ 64
3.4.4 หน้าจอรายจ่าย 66
3.4.5 หน้าจอแก้ไขข้อมูล 69

สารบัญ

หน้าที่

3.5 การออกแบบรายงานต่างๆที่ใช้ในโปรแกรม 74

3.5.1 รายงานรายรับเงิน 5 เปอร์เซ็น 74

3.5.2 รายงานรายรับเงินค่าปริ้นเตอร์ 74

3.5.3 รายงานรายรับเงินค่าชีต 75

3.5.4 รายงานรายจ่ายจากเงิน 5 เปอร์เซ็น 75

3.5.5 รายงานรายจ่ายเงินค่าปริ้นเตอร์ 76

3.5.6 รายงานรายจ่ายเงินค่าชีต 76

3.5.7 รายงานรายจ่ายจากเงินงบประมาณคณะวิชา 77

3.5.8 รายงานใบบันทึกรายการขอซื้อขอจ้าง 77

3.5.9 รายงานใบเบิกวัสดุ 79

3.5.10 รายงานใบสำคัญเพื่อคืนเงินทดลองจ่าย 79

บทที่ 4 บทสรุปและวิจารณ์ 80

4.1 ข้อได้เปรียบของระบบเดิม 80

4.2 ข้อเสียเปรียบของระบบเดิม 80

4.3 ข้อดีของระบบใหม่ 80

4.4 ข้อเสียของระบบใหม่ 81

สารบัญตาราง

หน้า

ตารางที่ 2.1 ตารางแสดงการเก็บข้อมูลของซิสเต็มเทเบิ้ลต่างๆ 37
ตารางที่ 2.2 ตารางแสดงการเก็บข้อมูลของดาต้าดิคชั่นนารี 38
ตารางที่ 2.3 ตารางแสดงชนิดข้อมูลในเอสคิวแอลเซิร์ฟเวอร์ 38
ตารางที่ 2.4 ตารางแสดง System Tables ของ mdb Database 43
ตารางที่ 3.1 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลยอดค้างชำระค่าปริ้นเตอร์ 54
ตารางที่ 3.2 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลบริษัท 54
ตารางที่ 3.3 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลประเภทเงินงบประมาณ 54
ตารางที่ 3.4 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลชื่อประเภทเงินงบประมาณ 55
ตารางที่ 3.5 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลการใช้เงินงบประมาณในใบตั้งเบิก 55
ตารางที่ 3.6 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลใบตั้งเบิก 55
ตารางที่ 3.7 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลใบเสร็จต่างๆในใบตั้งเบิก 56
ตารางที่ 3.8 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลใบเสร็จ 56
ตารางที่ 3.9 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลรายจ่ายจากเงินค่าชีต 57
ตารางที่ 3.10 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลรายจ่ายจากเงินค่าปริ้นเตอร์ 57
ตารางที่ 3.11 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลรายจ่ายจากเงิน 5 เปอร์เซ็นต์ 57
ตารางที่ 3.12 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลใบเสร็จของบริษัทต่างๆ 58
ตารางที่ 3.13 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลใบเสร็จและรายการต่างๆในใบเสร็จ 58
ตารางที่ 3.14 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลรายรับอื่นๆ 58
ตารางที่ 3.15 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลรายรับเงินงบประมาณภาควิชา 59
ตารางที่ 3.16 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลรายรับเงิน 5 เปอร์เซ็นต์ 59
ตารางที่ 3.17 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลรายรับเงินค่าปริ้นเตอร์ 59
ตารางที่ 3.18 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลรายรับเงินค่าชีต 60
ตารางที่ 3.19 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลอาจารย์ 60
ตารางที่ 3.20 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลการโอนย้ายระหว่างงบประมาณ 60

สารบัญภาพ

หน้า

รูปที่ 2.1 รูปแสดงข้อมูลแบบรีเลชั่นนอล 7
รูปที่ 2.2 รูปความสัมพันธ์แบบหนึ่งต่อหนึ่ง 8
รูปที่ 2.3 รูปแสดงความสัมพันธ์แบบหนึ่งต่อกลุ่ม 8
รูปที่ 2.4 รูปแสดงความสัมพันธ์แบบกลุ่มต่อกลุ่ม 9
รูปที่ 2.5 รูปแสดงกฎความถูกต้องของการอ้างอิง 10
รูปที่ 2.6 รูปแสดง R.X มีฟังก์ชันการขึ้นอยู่กับ R.Y 10
รูปที่ 2.7 รูปแสดง R.Y มีฟังก์ชันการขึ้นอยู่กับ R.X อย่างเต็มที่ 11
รูปที่ 2.8 แสดงสัญลักษณ์ของชนิดเอนติตี้วัสดุ 14
รูปที่ 2.9 แสดงสัญลักษณ์ของชนิดเลเบิและรหัสวัสดุ 14
รูปที่ 2.10 แสดงความสัมพันธ์อ้างอิงแบบ วันทูวัน (one to one) 14
รูปที่ 2.11 แสดงการเขียนความสัมพันธ์อ้างอิงแบบ วันทูวัน (One to one) อย่างย่อ 14
รูปที่ 2.12 แสดงความจริงแบบ แมนนีทูวัน (Many to one) 14
รูปที่ 2.13 แสดงความจริงแบบเมนี่ทูเมนี่ (many to many) 14
รูปที่ 2.14 แสดงการใช้ inter fac type uniqueness constraint 15
รูปที่ 2.15 แสดงการใช้อีควอลิตี้ คอนสเตรน (equality constrint) 15
รูปที่ 2.16 แสดงการใช้เอ็กคลูสชั่น คอนสเตรนท์ (Exclusion constraint) 15
รูปที่ 2.17 แสดงการใช้สับเซตคอนสเตรท (Subse constraint) 16
รูปที่ 2.18 แสดงการใช้สับไทพท์คอนสเตรท (Subtype constrint) 16
รูปที่ 2.19 แสดงการใช้เมนดาทอรี คอนสเตรนท และเล็กซิคอล คอนสเตรท 16
รูปที่ 2.20 รูปแสดงความสัมพันธ์ระหว่างไคล์เอนท์กับเซิร์ฟเวอร์ 17
รูปที่ 2.21 รูปแสดงความสัมพันธ์ระหว่างไคล์เอนท์กับเซิรฟ์เวอร์ 18
รูปที่ 2.22 รูปแสดงสถาปัตยกรรมของเซิร์ฟเวอร์ฐานข้อมูล 18
รูปที่ 2.23 รูปแสดงลักษณะของมิดเดิ้ลแวร์ 19
รูปที่ 2.24 รูปแสดงลักษณะทั่วๆ ไปของโอดีบีซี 22
รูปที่ 2.25 รูปแสดงแผนภาพโอดีบีซีไดร์ฟเวอร์ (ODBC Driver Diagram) 22
รูปที่ 2.26 รูปแสดงส่วนประกอบต่างๆ ใน โอดีบีซี (ODBC Component) 24
รูปที่ 2.27 แสดงกรุ๊ปของโปรแกรมเดลไฟ 28
รูปที่ 2.28 แสดงส่วนต่างๆของโปรแกรมบอแลนด์เดลไฟ 30
รูปที่ 2.29 แสดงวินส์โดว์หลัก 30
รูปที่ 2.30 แสดงวินโดว์ออบเจ็คอินสเป็คเตอร์ 31
รูปที่ 2.31 แสดงวินโดว์ฟอร์ม 32

สารบัญภาพ

หน้า

รูปที่ 2.32 แสดงวินโดว์เอดิเตอร์ 33
รูปที่ 2.33 แสดง BDE Administrator 35
รูปที่ 2.34 แสดงโครงสร้างข้อมูลของเอสคิวแอลเซิร์ฟเวอร์ 36
รูปที่ 2.35 รูปแสดงโครงสร้างฐานข้อมูลของ SQL Server 40
รูปที่ 2.36 รูปแสดงโครงสร้างของ master database 42
รูปที่ 2.37 รูปแสดงลักษณะของ System Tables 42
รูปที่ 2.38 รูปแสดงโครงสร้างของ mdb 42
รูปที่ 2.39 แสดงโครงสร้างของ model Database 44
รูปที่ 2.40 รูปแสดงโครงสร้างของ tempdb Database 45
รูปที่ 3.1 ดาต้าโฟลไดอะแกรมของระบบงานทั้งหมด 47
รูปที่ 3.2 ดาต้าโฟลไดอะแกรมแสดงรายได้เงิน 5 เปอร์เซ็นต์ 48
รูปที่ 3.3 ดาต้าโฟลไดอะแกรมแสดงรายได้จากการขายเอกสารประกอบการเรียน 48
รูปที่ 3.4 ดาต้าโฟลไดอะแกรมแสดงรายได้เงินค่าปริ้นเตอร์ 49
รูปที่ 3.5 ดาต้าโฟลไดอะแกรมแสดงรายได้จากเงินงบประมาณคณะวิศวกรรมศาสตร์ 49
รูปที่ 3.6 ดาต้าโฟลไดอะแกรมแสดงรายจ่ายที่เกี่ยวกับงานจัดซื้อ/จัดจ้าง 50
รูปที่ 3.7 ดาต้าโฟลไดอะแกรมแสดงรายจ่ายเกี่ยวกับสวัสดิการภายในภาควิชา 51
รูปที่ 3.8 ดาต้าโฟลไดอะแกรมแสดงรายจ่ายซ่อมบำรุงปริ้นเตอร์ 51
รูปที่ 3.9 ดาต้าโฟลไดอะแกรมแสดงรายจ่ายเกี่ยวกับงานเอกสาร 51
รูปที่ 3.10 แสดงแผนภาพไนแอมของระบบงาน 53
รูปที่ 3.11 รูปแสดงเมนูหลัก 63
รูปที่ 3.12 รูปแสดงเมนูย่อย 63
รูปที่ 3.13 รูปแสดงหน้าจอรายรับเงินงบประมาณคณะ 64
รูปที่ 3.14 รูปแสดงหน้าจอรายรับเงิน 5 เปอร์เซ็นต์ 64
รูปที่ 3.15 รูปแสดงหน้าจอรายรับเงินค่าปริ้นเตอร์ 65
รูปที่ 3.16 รูปแสดงหน้าจอรายรับเงินค่าขายเอกสาร 65
รูปที่ 3.17 รูปแสดงหน้าจอรายรับเงินจากรายได้อื่นๆ 66
รูปที่ 3.18 รูปแสดงหน้าจอรายจ่ายเงินสวัสดิการภาควิชา 66
รูปที่ 3.19 รูปแสดงหน้าจอรายจ่ายจากเงินค่าขายเอกสาร 67
รูปที่ 3.20 รูปแสดงหน้าจอใช้จ่ายในภาควิชา 67
รูปที่ 3.21 รูปแสดงหน้าจอใบตั้งเบิก 68
รูปที่ 3.22 รูปแสดงหน้าจอใบรายการ 68

สารบัญภาพ

หน้า

รูปที่ 3.23 เมนูแก้ไขข้อมูล 69
รูปที่ 3.24 รูปแสดงหน้าจอแก้ไขข้อมูลอาจารย์ 69
รูปที่ 3.25 รูปแสดงหน้าจอแก้ไขข้อมูลบริษัท 69
รูปที่ 3.26 รูปแสดงหน้าจอแก้ไขข้อมูลรายรับเงิน 5 เปอร์เซ็นต์ 70
รูปที่ 3.27 รูปแสดงหน้าจอแก้ไขข้อมูลรายรับเงินค่าขายเอกสาร 70
รูปที่ 3.28 รูปแสดงหน้าจอแก้ไขข้อมูลรายรับเงินงบประมาณคณะ 71
รูปที่ 3.29 รูปแสดงหน้าจอแก้ไขข้อมูลรายรับทั่วไป 71
รูปที่ 3.30 รูปแสดงหน้าจอแก้ไขข้อมูลรายจ่ายเงิน 5 เปอร์เซ็นต์ 72
รูปที่ 3.31 รูปแสดงหน้าจอแก้ไขข้อมูลรายจ่ายเงินค่าขายเอกสาร 72
รูปที่ 3.32 รูปแสดงหน้าจอแก้ไขข้อมูลใบตั้งเบิก 73
รูปที่ 3.33 รูปแสดงหน้าจอแก้ไขข้อมูลใบรายการ 73
รูปที่ 3.34 แสดงรายงานรายรับเงิน 5 เปอร์เซ็นต์ 74
รูปที่ 3.35 แสดงรายงานรายรับเงินค่าปริ้นเตอร์ 74
รูปที่ 3.36 แสดงรายรับเงินค่าชีต 75
รูปที่ 3.37 แสดงรายงานรายจ่ายเงิน 5 เปอร์เซ็นต์ 75
รูปที่ 3.38 แสดงรายงานรายจ่ายเงินค่าปริ้นเตอร์ 76
รูปที่ 3.39 แสดงรายงานรายจ่ายค่าชีต 76
รูปที่ 3.40 รูปแสดงรายงานใบส่งของ 77
รูปที่ 3.41 รูปแสดงรายงานใบบันทึกรายการขอซื้อขอจ้าง 78
รูปที่ 3.42 รูปแสดงรายงานใบเบิกวัสดุ 79
รูปที่ 3.43 รูปแสดงรายงานใบสำคัญเพื่อคืนเงินทดลองจ่าย 79

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันได้มีการนำเทคโนโลยีทางด้านคอมพิวเตอร์มาช่วยในงานต่างๆ อาทิเช่น งานด้านการทำบัญชีการเงิน,งานด้านการจัดเก็บข้อมูลสินค้าคงคลังงานด้านการจัดกระบวนการการทำงานงานประจำ (Transcation) เป็นต้น ซึ่งระบบต่างๆ ล้วนแล้วแต่ต้องการความถูกต้องและการจัดการกับข้อมูลได้อย่างมีประสิทธิภาพ และตอบสนองความต้องการของงานในด้านการจัดการระบบสารสนเทศ (Management Information System) ซึ่งต้องการความรวดเร็วในการนำเสนอรายงานได้อย่างทันถ่วงที สามารถเรียกดูสารสนเทศที่ผู้บริหารต้องการได้อย่างมีประสิทธิภาพ

ภาควิชาวิศวกรรมคอมพิวเตอร์ก็เป็นอีกหน่วยงานหนึ่งที่ได้นำเอาระบบสารสนเทศมาใช้ในกระบวนการทำงานประจำของภาควิชา อาทิเช่น การจัดการทางด้านการเงิน, การจัดการในด้านวัสดุและครุภัณฑ์ ซึ่งได้พัฒนาระบบสารสนเทศมาใช้เพื่ออำนวยความสะดวกแก่เจ้าหน้าที่ฝ่ายธุรการภาควิชาในการทำบัญชีรายรับ – รายจ่าย, อีกทั้งยังมีประโยชน์ในด้านระบบงานสารสนเทศทำให้สามารถติดตามงบประมาณที่นำมาใช้จ่าย และสามารถทำรายงานนำเสนอได้อย่างมีประสิทธิภาพ ซึ่งแต่เดิมนั้นอาศัยเจ้าหน้าที่ฝ่ายธุรการภาควิชาเป็นผู้ทำการ ลงบัญชีและจัดเก็บข้อมูลทั้งหมด ซึ่งมีความยุ่งยากและอาจเกิดข้อผิดพลาดขึ้นมาได้ง่าย

ดังนั้นโครงงานระบบ สารสนเทศบริหารการเงินภาควิชานี้จึงถูกพัฒนาขึ้นมาเพื่อลองรับความต้องการทางด้านการทำงานประจำของเจ้าหน้าที่ฝ่ายธุรการภาควิชา และเพื่ออำนวยความสะดวกในการทำบัญชีและติดตามยอดเงินงบประมาณที่ภาควิชานำมาใช้จ่าย เพื่อให้มีการจัดการที่มีประสิทธิภาพโดยนำระบบคอมพิวเตอร์มาช่วยพัฒนาใช้รูปบบของ Database Client/Server ซึ่งทำให้การทำงานมีประสิทธิภาพ สามารถพัฒนาโปรแกรมแอพพลิเคชันมาเพื่อสร้างโปรแกรมในส่วนของฟรอนด์เอนต์ (Front End) เพื่อใช้ในการติดต่อกับผู้ใช้ และระบบที่เป็นแบคเอนต์ (Back End) เพื่อใช้ในการจัดการกับดาต้าเบส

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 เพื่อพัฒนาระบบงานบัญชีของภาควิชาวิศวกรรมคอมพิวเตอร์ จากเดิมที่ทำด้วยมือเพียงอย่างเดียวมาเป็นระบบคอมพิวเตอร์ทั้งหมด

1.2.2 เพื่อศึกษาการนำระบบฐานข้อมูลมาใช้งานในการสร้างแอปพลิเคชั่นที่สามารถใช้งานได้จริง

1.2.3 เพื่อศึกษาระบบงานฐานข้อมูลผ่านเครือข่ายในระบบไคลเอ็นเซิร์ฟเวอร์

1.3 ขอบเขตของงานวิจัย

โครงงานนี้จะสร้างระบบฐานข้อมูลขึ้นมาบนเครือข่ายแบบไคลเอ็นเซิร์ฟเวอร์ โดยใช้แบบจำลองข้อมูลไนแอมในการสร้างฐานข้อมูล สำหรับออกแบบส่วนติดต่อกับผู้ใช้จะสร้างให้ผู้ใช้สามารถใช้งานได้ง่ายไม่ซ้ำซ้อน ซึ่งในงานวิจัยนี้จะออกแบบส่วนต่างๆระบบทั้งหมด

สำหรับในส่วนของการนำไปใช้งาน โครงงานนี้สามารถสร้างรายงานเพื่อสรุปผลข้อมูลที่ผู้ใช้ได้ทำการป้อนเข้าไปออกมาได้ โดยรูปแบบรายงานถูกออกแบบใหม่ในบางส่วน และบางส่วนก็จะนำมาจากรูปแบบรายงานที่ใช้กันอยู่แล้ว

นอกจากนั้นในโครงงานนี้ยังถือว่าเป็นโครงการที่ทดลองสร้าง เพื่อศึกษาความเป็นไปได้ในการใช้งาน ดังนั้นจึงอาจมีข้อผิดพลาดในการใช้งานเกิดขึ้นมาได้บ้าง ซึ่งต้องมีการพัฒนาต่อไปในอนาคต

1.4 วิธีการดำเนินงาน

ขบวนการในการพัฒนาระบบโครงงานสารสนเทศบริหารการเงินภาควิชานี้ ได้ทำการพัฒนาระบบโดยอาศัยทฤษฎีและหลักการความรู้ทางด้านซอฟร์แวร์เอ็นจิเนียริ่ง (Software Engineering) โดยขั้นตอนในการพัฒนาระบบทำการพัฒนาระบบเป็นช่วงๆ ประกอบไปด้วยเฟสการทำงานต่างๆ เริ่มตั้งแต่การทำความเข้าใจเกี่ยวกับระบบงานที่จะทำการพัฒนาโปรแกรม (Domain) และทำการศึกษาหาความต้องการของผู้ใช้ วิเคราะห์ปัญหาและการทำงานโดยอาศัยการสัมภาษณ์จากเจ้าหน้าที่ผู้รับผิดชอบในงานด้านการเงิน พิจารณาจากเอกสารแบบฟอร์มต่างๆ ที่เจ้าหน้าที่ทำงานประจำได้ทำอยู่เพื่อรวบรวมหาความต้องการของระบบ (requirement) และนำมาใช้เป็นแนวทางในการกำหนดขอบเขตความต้องการในการพัฒนาระบบ (specification) ทำการทดลองสร้างต้นแบบของระบบเพื่อทดสอบหาความต้องการของระบบที่แท้จริงเพื่อให้ได้ความต้องการของระบบที่ชัดเจนยิ่งขึ้น

ทำการพัฒนาระบบ (Implementation) โดยออกแบบระบบฐานข้อมูลโดยนำผลที่ได้จากการวิเคราะห์ระบบงาน ทำการเขียนโครงสร้างข้อมูลโดยใช้วิธีการวิเคราะห์ข้อมูลในรูปแบบไนแอม (NIAM Conceptual Model) เพื่อนำมาสร้างฐานข้อมูลแบบสัมพันธ์ (Relational Database Management System : RDBMS) พัฒนาโปรแกรมระบบในส่วนของฟรอนต์เอนด์ (Front End) เพื่อใช้งานในด้านการเก็บข้อมูลด้านการเงินของงานประจำและทำการทดสอบความถูกต้องในการออก แบบควบคู่ไปกับการพัฒนาระบบ ซึ่งเป็นการทดสอบและฝึกหัดให้ผู้ใช้ระบบสามารถเรียนรู้การออกแบบระบบควบคู่ไปด้วยในระหว่างการพัฒนา ทำให้ได้ความต้องการของระบบที่แท้จริงตอบสนองความต้องการใช้งานของผู้ใช้อย่างถูกต้อง ทดสอบการใช้งานระบบที่ได้ออกแบบควบคู่ไปกับงานประจำเดิมจนกระทั่งผู้ใช้มีความคุ้นเคยและพอใจกับระบบการทำงานใหม่ที่ใช้คอมพิวเตอร์มาร่วมด้วย ซึ่งช่วยอำนวยความสะดวกและเริ่มนำระบบที่พัฒนามาใช้งานแทนที่ระบบเดิม

หลังจากที่ได้ส่วนของฟรอนต์เอนด์ตรงตามความต้องการของผู้ใช้แล้ว ก็ทำการพัฒนาในส่วนติดต่อกับผู้ใช้ให้มีการใช้งานที่ง่ายขึ้น และพัฒนาในส่วนของรายงานให้ครอบคลุมต่อการนำไปใช้งานทั้งหมดของระบบบัญชีของภาควิชา

ซึ่งผลที่ได้จากการพัฒนาระบบโครงงานนี้ทำให้สามารถนำความรู้ความสามารถในด้านทฤษฎีและหลักการทางซอฟร์แวร์เอ็นจิเนียริ่งมาใช้ในการพัฒนาประบบโปรแกรมประยุกต์ที่สามารถใช้งานได้จริง อีกทั้งสามารถนำระบบที่พัฒนามาใช้เพื่อเป็นประโยชน์แก่ทางภาควิชาได้เป็นอย่างดี

บทที่ 2
หลักการและทฤษฎี

2.1 หลักการของระบบฐานข้อมูล (Database)

ฐานข้อมูล หมายถึง กลุ่มของข้อมูลและความสัมพันธ์ (A collection of data and relationships) ความสัมพันธ์ของฐานข้อมูลแสดงในรูปของ โมเดลของข้อมูลซึ่งโมเดลที่เป็นที่นิยม 3 โมเดลมีดังนี้

1. ไฮเออร์ราร์คีคอล ดาต้า โมเดล (Hierarchical Data Model) โดยบริษัทไอบีเอ็ม มีโครงสร้างข้อมูลระดับลอจิคอล (logical) เป็นโครงสร้างต้นไม้ (Tree Structure) ฐานข้อมูลจะต้องมีความสามารถในการค้นหาข้อมูลในโครงสร้างต้นไม้ได้ (Search Tree) เหมาะกับความสัมพันธ์ระหว่างข้อมูลหนึ่งต่อกลุ่ม (one to many) แต่ไม่เหมาะกับความสัมพันธ์แบบกลุ่มต่อกลุ่ม (many to many) เมื่อเขียนโปรแกรมจะเห็นเป็นโครงสร้างต้นไม้
2. เน็ทเวอร์ค ดาต้า โมเดล (Network Data Model) โดยกลุ่มบริษัทที่ไม่ใช่บริษัทไอบีเอ็มเพื่อแก้จุดเสียของไฮเออร์ราร์คีคอลดาต้าโมเดลให้เหมาะกับข้อมูลทั้งแบบ หนึ่งต่อกลุ่มและแบบกลุ่มต่อกลุ่มจึงทำให้เพิ่มความซับซ้อนมากขึ้น โครงสร้างข้อมูลระดับลอจิคอลเป็นมัลติลิสต์ (Multilist) เมื่อเขียนโปรแกรมเห็นเป็น ลิงค์ลิสต์ (Link List) เพื่อเลี่ยงการเข้าใจผิดเกี่ยวกับชื่อของโมเดลที่มีคำว่าเน็ทเวอร์อยู่ด้วยแต่มิได้มีความเกี่ยวข้องกับโครงข่ายสื่อสารแต่อย่างใด เน็ทเวอร์คดาต้าโมเดล จึงมักถูกเรียกว่า คอดาซิลโมเดล (CODASYL Model : Conference On DAta System Language) หรือ ดีบีทีจี (DBTG : Database Task Group)
3. รีเลชั่นนอลดาต้าโมเดล (Relational Data Model) คิดค้นโดย Edgar E.Codd การทำงานโดยอาศัยหลักการทางคณิตศาสตร์และแนวคิดที่ว่าผลของการออกแบบทุกครั้งควรได้ผลเหมือนกัน การออกแบบแต่ละครั้งตามโมเดล 2 โมเดลแรกที่กล่าวมาอาจได้ผลไม่เหมือนกันได้ เนื่องจากอาศัยหลักทางศิลป์มากกว่าทางคณิตศาสตร์

2.2 ลักษณะของฐานข้อมูลที่ดี

2.2.1 ความซ้ำซ้อนน้อย (Minimum Redundancy) ควรให้มีความซ้ำซ้อนน้อยที่สุดความซ้ำซ้อน (Redundancy) หมายถึง ความสัมพันธ์ระหว่างข้อมูล (Fact) มีมากกว่า 1 ครั้ง

2.2.1.1 ข้อเสียของการมีความซ้ำ ซ้อน

- เปลืองเนื้อที่ในการเก็บข้อมูล
- ทำให้เกิด Multiple Update ซึ่งอาจทำให้เกิดการ Update ไม่ครบได้ง่าย

2.2.1.2 ข้อดีของการมีความซ้ำซ้อน

- หากเป็นความซ้ำซ้อนที่มีการควบคุมที่ดี อาจถือได้ว่าเป็นการสำรองข้อมูล (Backup) ได้
- ความซ้ำซ้อนอาจช่วยให้ทำคิวรี่ (Query) ได้เร็วขึ้น

2.2.2 ความถูกต้อง (Maximum Integrity) ให้มีความถูกต้องของข้อมูลมากที่สุด แบ่งเป็น

2.2.2.1 ความถูกต้องทางสแตติค (Static Integrity) คือ ความถูกต้องของสถานะของข้อมูลเช่น พระสงฆ์ทุกรูปต้องมีอายุตั้งแต่ 20 ปีขึ้นไป ดังนั้นหากมีพระสงฆ์อายุ 15 ปีก็จะผิดจากความถูกต้องทางสแตติคหรือ จำนวนวันใน 1 สัปดาห์ต้องเป็น 7 วัน หากมีที่ใดที่เก็บข้อมูลจำนวนวันใน 1 สัปดาห์เป็นค่าที่ไม่ใช่ 7 ก็จะผิดตามความถูกต้องทางสแตติกเช่นกัน

2.2.2.2 ความถูกต้องทางไดมามิค (Dynamic Integrity) คือ ความถูกต้องของลำดับการเปลี่ยนแปลงข้อมูล (Update) เช่น การเปลี่ยนแปลงสถานะของฝ่ายหญิง จากหย่าจะต้องเป็นนางมิใช่นางสาวเป็นต้น

2.2.3 ข้อมูลมีความเป็นอิสระสูง (High Degree of Data Independence) โปรแกรมเป็นอิสระจากการเปลี่ยนแปลงโครงสร้างข้อมูล หมายความถึง หากเปลี่ยนโครงสร้างทาง Physical แล้วไม่ส่งผลทาง Logical หรือหากเปลี่ยนโครงสร้างทาง Logical แล้วโปรแกรมไม่ต้องเปลี่ยนแปลง เป็นต้น

2.2.4 ความปลอดภัยสูง (High Security) ต้องมีระบบรักษาความปลอดภัยของข้อมูลที่ดี เช่น

- มีรหัสผ่าน (Login Password) ทั้งในระดับ โอเอส (OS) และระดับดีบีเอ็มเอส (DBMS)
- การมองเห็น (View) กำหนดได้ว่าจะให้ผู้ใช้มีสิทธิเห็นข้อมูลได้แค่ไหน
- การอนุญาตให้เข้าถึงข้อมูล (Access Permission) กำหนดได้ว่าจะให้ผู้ใช้ใช้งานอะไรได้บ้าง เช่น การแทรกข้อมูล (Insert), การลบข้อมูล (Delete), การเปลี่ยนแปลงข้อมูล (Modify)
- การเข้าถึงข้อมูลต้องผ่านโอดีบีซีเสมอ (Non Subversion Rule Logical/ Physical mapping) คือ ระบบฐานข้อมูลใดๆ ต้องไม่ยอมให้โปรแกรมเข้าถึงข้อมูลได้โดยไม่ผ่าน DBMS

2.2.5 การควบคุมจากศูนย์กลางทางลอจิคอล (Logically Centralized Control) คือ ระบบฐานข้อมูลนั้นต้องมีการควบคุมจากศูนย์กลาง ทำโดยบุคคลในตำแหน่ง ดีบีเอ หรือผู้ดูแลระบบฐานข้อมูล (DBA : Database system Administrator)

2.3 รีเลชั่นนอลโมเดล (Relational Model)

รีเลชั่นนอลโมเดลเป็นโมเดลที่ใช้ในการอธิบายความสัมพันธ์ของข้อมูลที่ถูกเก็บด้วยระบบจัดการฐานข้อมูลแบบรีเลชั่นนอล (Relational Database Management System : RDBMS) ในปัจจุบันนี้มีการใช้งานกันอย่างแพร่หลายไปมาก มีการนำไปใช้งานกับเครื่องระดับตั้งแต่เมนเฟรมลงไปจนถึงเครื่องระดับไมโคร และเป็นที่ยอมรับกันว่าผู้ใช้ระบบฐานข้อมูลส่วนใหญ่ (โดยเฉพาะผู้ที่ทำงานด้วยเครื่องมินิ และระดับไมโคร) จะมีความคุ้นเคยกับรีเลชั่นนอลโมเดลมากกว่าอีก 2 โมเดลคือ ไฮเออร์ราร์คีเคิลโมเดล (Hierarchical Model) และ เน็ตเวอร์คโมเดล (Network Model)

ข้อดีของรีเลชั่นอลโมเดลที่มีเหนือกว่าอีก 2 โมเดล ดังนี้

- ข้อมูลที่เก็บโดยรีเลชั่นนอลโมเดลจะมาจากมุมมองของผู้ใช้ ซึ่งจะมีความซับซ้อนน้อยกว่าข้มูลที่เก็บโดยอีก 2 โมเดล ดังนั้นรีเลชั่นนอลโมเดล จึงเป็นโมเดลที่เข้าใจได้ง่ายกว่า

- ระบบส่วนใหญ่ที่ใช้รีเลชั่นนอลโมเดลนี้มักจะมีเครื่องมือที่ช่วยให้ผู้ใช้สามารถจัดการกับข้อมูลที่เก็บอยู่ได้ง่ายกว่าข้อมูลที่จัดเก็บด้วยโมเดลแบบอื่นๆ
- รีเลชั่นนอลโมเดลนี้มีเครื่องมือที่ช่วยให้ผู้ใช้สามารถค้นพบปัญหาที่เกิดขึ้นในการออกแบบระบบฐานข้อมูลได้โดยง่าย และยังง่ายในการแก้ไขการออกแบบที่ผิดพลาดด้วย
- รีเลชั่นนอลโมเดลเป็นโมเดลที่มีความสอดคล้องกับหลักการของฐานข้อมูล ผู้ใช้ไม่ต้องพะวงกับรายละเอียดของการจัดเก็บข้อมูลเหมือนกับการจัดข้อมูลของโมเดลอื่น
- ภาษาที่ใช้ในการจัดการกับข้อมูลที่จัดเก็บด้วยระบบจัดการฐานข้อมูลแบบรีเลชั่นนอล (ภาษา SQL : Structure Query Language) เป็นภาษาแบบเซต (set oriented) ซึ่งจะต่างกับภาษาที่ใช้ในการจัดการกับข้อมูลที่จัดเก็บด้วยระบบจัดการฐานข้อมูลของโมเดลอื่นที่เป็นภาษาแบบเรคคอร์ดแอทอะไทม์ (record-at-a-time)

แม้ว่ารีเลชั่นนอลโมเดลจะมีข้อดีหลาสยประการดังได้กล่าวไปแล้ว แต่ในปัจจุบันก็ยังมีจุดอ่อนที่มีการอ้างอิงถึงเสมอ คือ ระบบฐานข้อมูลแบบรีเลชั่นนอลโมเดลนี้มักมีประสิทธิภาพในการใช้งานสู้อีก 2 โมเดลไม่ได้โดยเฉพาะในการประยุกต์ใช้งานขององค์กรขนาดใหญ่

2.3.1 ศัพท์เฉพาะของรีเลชั่นนอลโมเดล

ตารางที่เก็บข้อมูลที่เป็นรีเลชั่นนอลโมเดล เรียกว่า "รีเลชั่น" โดยที่ตารางบางตารางอาจไม่เป็นรีเลชั่นก็ได้ ตารางที่มีลักษณะเป็นรีเลชั่นจะต้องมีคุณลักษณะดังนี้

- แต่ละช่องของตารางจะบรรจุข้อมูลได้เพียงค่าเดียว
- ชื่อหัวข้อในแต่ละคอลัมน์มีความแตกต่างกัน อันเป็นชื่อของแอตทริบิวของเอนติตี้
- ค่าของข้อมูลที่อยู่ในแต่ละคอลัมน์คือค่าของแอตทริบิวตามที่ระบุหัวข้อไว้ที่หัวของคอลัมน์นั้นๆ
- การเรียงลำดับคอลัมน์ไม่ถือว่ามีความสำคัญ
- ข้อมูลแต่ละแถวจะต้องแตกต่างกัน
- การเรียงลำดับแถวไม่ถือว่ามีความสำคัญ

ตารางที่มีคุณลักษณะดังกล่าวจะเรียกว่า รีเลชั่น

ดังนั้น เราจะได้นิยามของ ฐานข้อมูลแบบรีเลชั่นนอล คือ ฐานข้อมูลที่เกิดจากการรวบรวมรีเลชั่นต่างๆ ที่มีความสัมพันธ์ (relationship) ระหว่างกัน

เราจะเรียกข้อมูลแต่ละแถวในแนวนอนของรีเลชั่นว่า ทัพเพิล (tuple) และเรียกข้อมูลแต่ละแถวในแนวตั้งหรือแนวคอลัมน์ว่า แอตทริบิว (attribute) โดยที่คำว่า คีย์ (key) จะหมายถึงข้อมูลที่เกิดจากแอตทริบิว 1 ตัวหรือหลายตัวก็ได้

ชื่อแอตทริบิว	ชื่อแอตทริบิว	ชื่อแอตทริบิว
ค่าของแอตทริบิว	ค่าของแอตทริบิว	ค่าของแอตทริบิว
ค่าของแอตทริบิว	ทัพเพิล (tuple)	ค่าของแอตทริบิว

รูปที่ 2.1 รูปตารางแสดงข้อมูลแบบรีเชชั่นนอล

แต่ละรีเลชั่นจะต้องมีสิ่งที่เรียกว่า คีย์หลัก (primary key) คือ ข้อมูลของแอตทริบิว 1 ตัวหรือมากกว่า 1 ตัวก็ได้ ที่สามารถใช้เป็นตัวระบุให้เราได้ว่ากำลังอ้างอิงถึงข้อมูลทัพเพิลใด ส่วนแอตทริบิวที่เป็นคีย์หลักของรีเลชั่น จะเรียกว่า คีย์นอก (foreign key) (เช่น รีเลชั่น A มีแอตทริบิวรหัสสินค้าเป็นคีย์หลัก แล้วในรีเลชั่น B มีแอตทริบิวรหัสสินค้าเช่นเดียวกับ A เราจะเรียก แอตทริบิวดังกล่าวของรีเลชั่น B ว่าเป็นคีย์นอกของคีย์หลักของรีเลชั่น A) ในกรณีที่รีเลชั่นมีแอตทริบิวหรือกลุ่มแอตทริบิวที่มีคุณสมบัติเป็นคีย์หลักได้อยู่หลายแอตทริบิวเราจะเรียกแอตทริบิวหรือกลุ่มแอตทริบิวที่มิได้ถูกเลือกให้เป็นคีย์หลักว่า คีย์คู่แข่ง (candidate key) หรือคีย์สำรอง (alternate key) และแอตทริบิวอื่นๆ ที่เหลือที่มิได้เป็นคีย์หลักและไม่ได้เป็นส่วนใดส่วนหนึ่งของคีย์หลักก็จะถูกเรียกว่า นอนคีย์ (nonkey attribute)

คำว่า โดเมน (domain) หมายถึง กรอบของค่าต่างๆ ที่เป็นไปได้เช่น โดเมนของแอตทริบิววันในหนึ่งสัปดาห์ หมายถึง ค่าของวันที่เป็นไปได้ คือ จันทร์ อังคาร พุธ พฤหัสบดี ศุกร์ เสาร์ อาทิตย์ แต่ในการเก็บค่าข้อมูลในรีเลชั่นนอลนั้น บางกรณีที่เรามีการกำหนดโดเมนให้กับแอตทริบิวแล้ว แต่ข้อมูลที่จุถูกเก็บเข้าไปอาจถูกบรรจุเข้าไปในภายหลัง ลักษณะนี้จะทำให้เกิด ค่าว่าง (Null value) ขึ้นชั่วขณะก่อนที่จะมีการบรรจุค่าข้อมูลที่อยู่ในโดเมนที่กำหนดไว้เข้าไป ดังนั้น “ค่าว่าง” จึงหมายถึงค่าที่ยังไม่ทราบแน่ชัดว่าแอตทริบิวนั้นจะมีค่าเป็นค่าใด หรือค่าของข้อมูลที่ไม่อยู่ในโดเมนที่กำหนด โดยมีข้อบังคับไว้ว่า แอตทริบิวที่ทำหน้าที่เป็นคีย์หลักของรีเลชั่นจะมีค่าข้อมูลห้ามเป็นค่าว่างเสมอ เพราะจะทำให้การเข้าถึงข้อมูลในทัพเพิล (tuple) นั้นกระทำไม่ได้

เมื่อมีการจัดเก็บข้อมูลในฐานข้อมูลใดๆ แล้ว ข้อมูลจะถูกแยกออกเป็นกลุ่มของข้อมูลเป็นชุดที่ประกอบด้วยแอตทริบิวต่างๆ ที่มีความสัมพันธ์กัน เช่น การเก็บข้อมูลของบุคลากรในโรงเรียนก็อาจแยกเก็บเป็นกลุ่มข้อมูลของนักเรียน, กลุ่มข้อมูลอาจารย์, และกลุ่มข้อมูลของนักการภารโรง เป็นต้น กลุ่มข้อมูลแต่ละกลุ่มนี้จะเรียกว่า เอนติตี้ (entity) ซึ่งแต่ละเอนทริตี้จะประกอบไปด้วยแอตทริบิวต่างๆ ที่มีความสัมพันธ์กันเช่น เอนทริตี้ของนักเรียนก็จะประกอบไปด้วยชื่อ, นามสกุล, ที่อยู่, ชั้นเรียน เป็นต้น

จากการแยกจัดเก็บข้อมูลออกเป็นเอนติตี้นี้ แต่ละเอนติตี้ก็จะมีความสัมพันธ์กัน ความสัมพันธ์ระหว่างเอนติตี้สามารถแบ่งออกเป็น 3 ชนิด คือ

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

2.3.2 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง

ความสัมพันธ์แบบหนึ่งต่อหนึ่งระหว่างเอนติตี้ คือ เมื่อเอนติตี้หนึ่งมีข้อมูลของคีย์หลักค่าหนึ่ง ค่าข้อมูลดังกล่าวจะมีความสัมพันธ์กับค่าข้อมูลของคีย์หลักของอีกเอนติตีหนึ่งเพียงค่าเดียวเท่านั้น เช่น กำหนดให้ความสัมพันธ์ระหว่างเอนติตี้นักเรียน กันเอนติตี้ผู้ปกครองเป็นแบบหนึ่งต่อหนึ่ง หมายความว่า การที่เราอ้างถึงนักเรียนคนใดคนหนึ่งจะสามารถอ้างถึงผู้ปกครองได้เพียงคนเพียวเท่านั้น และในทางตรงกันข้ามจะต้องเป็นจริงด้วยคือ เมื่อเราอ้างถึงผู้ปกครองคนใดคนหนึ่งแล้วจะสามารถอ้างอิงถึงนักเรียนได้เพียงคนเดียวเท่านั้น

ชื่อนักเรียน	ผู้ปกครอง
ธีระ	วิศิษฐ์
วีระชัย	วารศีร
ชูชาติ	กิติ

รูปที่ 2.2 รูปความสัมพันธ์แบบหนึ่งต่อหนึ่ง

2.3.3 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม

ความสัมพันธ์แบบหนึ่งต่อกลุ่มระหว่างเอนติตี้ คือ เมื่อเอนติตี้หนึ่งมีข้อมูลของคีย์หลักค่าหนึ่ง ค่าข้อมูลดังกล่าวจะมีความสัมพันธ์กับค่าข้อมูลของคีย์หลักของอีกเอนติตี้หนึ่งได้หลายค่า เช่น กำหนดให้ความสัมพันธ์ระหว่างเอนติตี้ภาควิชา กับเอนติตี้ของนักเรียน เป็นแบบหนึ่งต่อกลุ่มหมายความว่า การที่เราอ้างถึงภาควิชาใดภาควิชาหนึ่งจะสามารถอ้างถึงนักเรียนได้หลายคน และในทางตรงข้ามจะมีความหมายว่า เมื่อเราอ้างถึงนักเรียนคนใดคนหนึ่งแล้ว จะสามารถอ้างอิงถึงภาควิชาได้เพียงภาควิชาเดียวเท่านั้น แต่นักเรียนที่เราอ้างถึงเป็นคนละคนกันอาจจะอ้างอิงถึงภาคิชาเดียวกันก็เป็นได้

ชื่อนักเรียน	ภาควิชา
ธีระ	คอมพิวเตอร์
วานิจ	โทรคมนาคม
มานิตย์	คอมพิวเตอร์

รูปที่ 2.3 รูปแสดงความสัมพันธ์แบบหนึ่งต่อกลุ่ม

2.3.4 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม

ความสัมพันธ์แบบกลุ่มต่อกลุ่มระหว่างเอนติตี้ คือ ค่าข้อมูลของคีย์หลักของเอนติตี้หนึ่งที่ต่างกันอาจอ้างอิงถึงค่าข้อมูลของคีย์หลักของอีกเอนติตี้หนึ่งได้ค่าเดียวหรือหลายค่าก็ได้ เช่น กำหนดให้ ความ

สัมพันธ์ระหว่างเอนติตี้ของนักเรียนกับเอนติตี้ของอาจารย์เป็นแบบกลุ่มต่อกลุ่มแล้วหมายความว่า การที่เราอ้างถึงนักเรียนคนหนึ่งหรือหลายคน จะสามารถอ้างอิงถึงอาจารย์คนเดียวกันก็ได้ และในทางกลับกัน การที่เราอ้างอิงถึงอาจารย์คนหนึ่งหรือหลายคนจะสามารถอ้างอิงถึงนักเรียนคนเดียวกันก็ได้

ชื่อนักเรียน	ชื่ออาจารย์
ธีระ	ธนา
วานิจ	ธนา
มานิตย์	นวพร
ปกรณ์	สมศักดิ์

รูปที่ 2.4 รูปแสดงความสัมพันธ์แบบกลุ่มต่อกลุ่ม

2.4 ทฤษฎีของระบบฐานข้อมูลแบบรีเลชั่นนอล

2.4.1 กฎของความถูกต้อง (Integrity Rule)

กฎของความถูกต้อง (Integrity Rule) ของรีเลชั่นนอลโมเดล (Relational Model) เป็นทฤษฎีที่ช่วยยืนยันความถูกต้องของความสัมพันธ์ระหว่างข้อมูลว่า รีเลชั่นใดที่เป็นไปตามกฎของความถูกต้องนี้แล้วย่อมจะมีความสัมพันธ์ระหว่างข้อมูลอย่างถูกต้องอยู่ตลอดเวลา ไม่ว่ารีเลชั่นนั้นจะมีการเปลี่ยนแปลงแก้ไขข้อมูลไปในรูปแบบใดก็ตาม

กฎของความถูกต้องมีความหมายอยู่ 2 ลักษณะคือ กฎความถูกต้องของเอนติตี้ (entity integrity rule) และ กฎความถูกต้องของการอ้างอิง (referential integrity rule) ดังอธิบายได้ดังนี้

กฎความถูกต้องของเอนติตี้ กล่าวว่า

"แอตทริบิวทุกตัวที่เป็นส่วนของคีย์หลักไม่อนุญาติให้มีค่าว่า"

หมายความว่า คีย์หลักของทุกรีเลชั่นห้ามเก็บค่าข้อมูลที่เป็นค่าว่าง เหตุผลของข้อกำหนดนี้คือ เพื่อให้การเข้าถึงข้อมูลในทัพเพิลใดๆ ของรีเลชั่นมีความเป็นไปได้เสมอ เพราะถ้าคีย์หลักของทัพเพิลใดมค่าข้อมูลเป็นค่าว่างแล้ว จะส่งผลให้ไม่สามารถเข้าถึงข้อมูลในทัพเพิลนั้นได้

กฎความถูกต้องของการอ้างอิง กล่าวว่า

"ถ้าเรามีรีเลชั่น R2 ซึ่งมี FK เป็นคีย์นอกที่อ้างอิงถึงคีย์หลัก PK ในรีเลชั่น R1 สำหรับทุกค่าของ FK ใน R2 จะต้องมีค่าเท่ากับค่า PK ในทัพเพิลใดทัพเพิลหนึ่งในรีเลชั่น R1 มีค่าของแอตทริบิวทุกตัวใน FK เป็นค่าว่าง"

หมายความว่าแอตทริบิวใดๆ ที่เป็นคีย์หลักของรีเลชั่นหนึ่ง เมื่อมีการนำแอตทริบิวนั้นไปเป็นคีย์นอกของอีกรีเลชั่นหนึ่ง การเป็นคีย์นอกของแอตทริบิวนั้นจะต้องมีโดเมนเป็นโดเมนเดียวกันกับแอตทริบิวที่เป็นคีย์หลัก ทั้งนี้เพื่อให้การนำรีเลชั่นมาใช้งานร่วมกัน (การนำรีเลชั่นมา join กัน) กระทำได้อย่างถูกต้อง คือ ทุกแอตทริบิวที่เป็นคีย์นอกจะต้องมีข้อมูลซ้ำกับข้อมูลของแอตทริบิตที่เป็นคีย์หลักอย่างแน่นอน

แต่อาจมีบางค่าข้อมูลของแอตทริบิวที่เป็นคีย์หลักเป็นข้อมูลไม่อยู่ในโดเมนของแอตทริบิวที่เป็นคีย์นอกก็ได้ นั่นคือ โดเมนของคีย์นอกจะต้องเล็กกว่าหรือเท่ากับโดเมนของคีย์หลักเสมอ

PK ของ R1	FK ของ R1
001	Red
002	Blue
003	Black
004	Green

PK ของ R2	FK ของ R1
S1	001
S2	002
S1	003
S3	004

รูปที่ 2.5 รูปแสดงกฎความถูกต้องของการอ้างอิง

2.4.2 ฟังก์ชันการขึ้นต่อกัน (Functional Dependency)

ฟังก์ชันการขึ้นต่อกัน เป็นข้อกำหนดที่ช่วยให้เห็นความสัมพันธ์ของแอตทริบิวต่างๆ ที่อยู่ในรีเลชั่นเพราะเป็นไปได้ที่แอตทริบิวต่างๆ ที่อยู่ในเอนติตี้เดียวกันเหล่านี้มีความสัมพันธ์กันเองโดยที่ความสัมพันธ์นี้อาจเกี่ยวข้องหรือไม่เกี่ยวกับความสัมพันธ์ที่มันมีต่อคีย์หลักของเอนติตี้นั้นก็เป็นได้ ซึ่งการที่แอตทริบิวเหล่านั้นมีความสัมพันธ์กันเองจะเป็นสิ่งที่เราต้องพิจารณาแยกออกเป็นรีเลชั่นย่อยๆ เพราะแอตทริบิวแต่ละรีเลชั่นก็ควรจะมีความสัมพันธ์กับคีย์หลักของรีเลชั่นของตนเองเท่านั้น

กำหนดรีเลชั่น R ถ้ามีแอตทริบิว Y ของ R เป็นฟังก์ชันที่ขึ้นต่อแอตทริบิว X ของรีเลชั่น เราสามารถเขียนแทนได้ด้วยสัญลักษณ์

$$R.X \longrightarrow R.Y$$

อ่านว่า R.X มีฟังก์ชันการขึ้นอยู่กับ R.Y

หรือ R.X มีฟังก์ชันในการเลือ R.Y

หรือ R.Y ขึ้นอยู่กับ R.X

นิยาม R.X มีฟังก์ชันการขึ้นอยู่กับ R.Y ก็ต่อเมื่อ ทุกค่าข้อมูลของแอตทริบิว X ใน R มีค่าข้อมูลของแอตทริบิว Y ใน R ได้เพียงค่าเดียวเสมอ โดยที่แอตทริบิว X และ Y อาจจะเป็นคีย์ประกอบ (Composite key) ก็ได้

รีเลชั่น R

X	Y
S1	P1
S1	P2

รูปที่ 2.6 รูปแสดง R.X มีฟังก์ชันการขึ้นอยู่กับ R.Y

นิยาม R.Y มีฟังก์ชันการขึ้นอยู่กับ R.X อย่างเต็มที่ก็ต่อเมื่อ R.X มีฟังก์ชันการขึ้นอยู่กับ R.X และไม่ขึ้นอยู่กับข้อมูลเพียงบางส่วนของ R.X โดยที่แอตทริบิว X และ Y อาจจะเป็นคีย์ประกอบก็ได้

รีเลชั่น R

X

X1	**X2**	**Y**
S1	P1	Red
S1	P2	Blue
S2	P2	Black
S3	P4	Green

รูปที่ 2.7 รูปแสดง R.Y มีฟังก์ชันการขึ้นอยู่กับ R.X อย่างเต็มที่

2.4.3 การนอร์มัลไลซ์ (Normalization)

การนอร์มัลไลซ์เป็นการออกแบบฐานข้อมูลแบบที่เป็นมาตราฐานที่สุด โดยมีจุดประสงค์ของการออกแบบเพื่อลดความซ้ำซ้อนของความสัมพันธ์ของข้อมูลให้เหลือน้อยที่สุด (minimum redundancy) ซึ่งตามมาตราฐานปกติจะมีอยู่ 3 ระดับคือ

- 1 NF (First Normal Form)
- 2 NF (Second Normal Form)
- 3 NF (Third Normal form)

โดยที่รีเลชั่นใดที่ยังไม่สอดคล้องตามรูปแบบนอร์มัล (Normal Form) ทั้งสามก็จะต้องมีการแยกรีเลชั่นนั่นๆ ออกเป็นรีเลชั่นย่อยๆ ต่อไปอีก (Decomposition Method)

- 4 NF (Forth Normal Form)
- 5 NF (Fifth Normal Form)

หากรีเลชั่นใดมีมาตราฐานถึงรูปแบบนอร์มัลระดับที่ 5 (5 NF) แล้วก็จะมั่นใจได้ว่า รีเลชั่นนั้นจะไม่มีความซ้ำซ้อนของความสัมพันธ์ของข้อมูลอย่างแน่นอน

นอกจากนี้ยังมีการออกแบบรูปแบบนอร์มัลเพิ่มเตมระหว่างรูปแบบนอร์มัลที่ 3 (3 NF) และรูปแบบนอร์มัลที่ 4 (4 NF) โดย Boyce และ Codd ซึ่งมีชื่อว่า Boyce Code Normal Form (BCNF) อีกด้วย

2.4.3.1 รูปแบบนอร์มัลระดับที่ 1

การปรับรีเลชั่นให้อยู่ในรูปแบบนอร์มัลระดับที่ 1 คือ การปรับจากรีเลชั่นที่ไม่นอร์มัล (unnormalize relation) เช่น รีเลชั่นที่มีข้อมูลของแอตทริบิวบางช่องมีมากกว่า 1 ค่า (มีแอตทริบิตที่มีข้อมูลเป็น repeating group)

นิยาม รีเลชั่นจะอยู่ในรูปแบบนอร์มัลระดับที่ 1 (1 NF) ก็ต่อเมื่อโดเมนของแต่ละแอตทริบิวประกอบด้วยข้อมูลที่เป็นหน่อยย่อยที่สุด

สิ่งที่ได้จากการที่รีเลชั่นอยู่ในรูปแบบนอร์มัลระดับที่ 1 ก็คอ รีเลชั่นยังคงมีความซ้ำซ้อนของความสัมพันธ์ระหว่างข้อมูลอยู่มากมาย เพราะนิยามของรูปแบบนอร์มัลระดับที่ 1 นี้กำหนดเพียงเฉพาะว่า แต่ละแอตทริบิวของรีเลชั่นจะมีโดเมนที่มีสมาชิกเป็นหน่วยที่เล็กที่สุดเท่านั้น มิได้เป็นการลดความซ้ำซ้อนของความสัมพันธ์ระหว่างข้อมูลแต่ประการใด

2.4.3.2 รูปแบบนอร์มัลระดับที่ 2

นิยาม รีเลชั่นจะอยู่ในรูปแบบนอร์มับระดับที่ 2 (2 NF) ก็ต่อเมื่อรีเลชั่นนั้นอยู่ในรูปแบบนอร์มัลระดับที่ 1 แล้วและทุกแอตทริบิวที่ไม่เป็นส่วนหนึ่งของคีย์หลัก (non-key attribute) จะต้องขึ้นอยู่กับคีย์หลักของรีเลชั่นนั้นอย่างเต็มที่

สิ่งที่ได้จากการที่รีเลชั่นอยู่ในรูปแบบนอร์มัลระดับที่ 2 คือ ข้อมูลของบางแอตทริบิวที่ไม่ใช่คีย์หลักอาจมีความสัมพันธ์กันเองที่ไม่มีความหมายกับคีย์หลักเลย ซึ่งความสัมพันธ์ดังกล่าวนี้ถือว่าเป็นความซ้ำซ้อนประการหนึ่งของรีเลชั่นนั้นๆ ที่จะต้องทำการลดด้วยรูปแบบนอร์มัลในระดับต่อไป

2.4.3.3 รูปแบบนอร์มัลระดับที่ 3

นิยาม รีเลชั่นจะอยู่ในรูปแบบนอร์มัลระดับที่ 3 (3 NF) ก็ต่อเมื่อรีเลชั่นนั้นอยู่ในรูปแบบนอร์มัลระดับที่ 2 แล้วและทุกแอตทริบิวที่ไม่เป็นส่วนใดส่วนหนึ่งของคีย์หลักจะต้องไม่เป็นฟังก์ชันที่ขึ้นต่อกันเอง

โดยปกติแล้ว สิ่งที่ได้จากการที่รีเลชั่นอยู่ในรูปแบบนอร์มัลระดับที่ 3 คือ รีเลชั่นจะไม่มีความซ้ำซ้อนอีกต่อไปโดยที่จะสอดคล้องกับรูปแบบนอร์มัลระดับที่ 4 และ 5 ด้วย แต่ก็มีรีเลชั่นบางลักษณะที่จะต้องทำให้อยู่ในรูปแบบนอร์มัลระดับที่ 4 และ 5 ต่อไปอีก รีเลชั่นดังกล่าวจะมีลักษณะดังต่อไปนี้

- เป็นรีเลชั่นที่มีหลายคีย์คู่แข่ง และ
- เป็นคีย์คู่แข่งที่เกิดจากการรวมกันของคีย์ย่อยๆ (candidate key เป็น combine key) และ
- เป็นคีย์คู่แข่งที่มีการเหลื่อมซ้อนกัน (overlap กัน)

2.4.4.4 รูปแบบนอร์มัลระดับที่ 4

นิยาม รีเลชั่นจะอยู่ในรูปแบบ 4 NF ก็ต่อเมื่อ รีเลชั่นนั้นอยู่ในรูปแบบนอร์มัลระดับที่ 3 (3 NF หมายถึง BCMF) และเป็นรีเลชั่นไม่มีการขึ้นต่อกันเชิงกลุ่ม

2.4.4.5 รูปแบบนอร์มัลระดับที่ 5

นิยาม รีเลชั่นจะอยู่ในรูปแบบ 5 NF ก็ต่อเมื่อ ทำการแตกรีเลชั่นแล้วนำกลับมารวมต้องได้รีเลชั่นเดิมไม่สามารถแบ่งย่อยออกไปได้อีก

2.5 การออกแบบฐานข้อมูลโดยใช้วิธีไนแอม

(NIAM : Nijssen 's Information Analysis Methodology)

การออกแบบฐานข้อมูลโดยใช้วิธีไนแอม (NIAM) เป็นวิธีในการออกแบบฐานข้อมูลโดยการแสดงความหมาย, ความสัมพันธ์และข้อจำกัดต่างๆ ของข้อมูล ด้วยแบบจำลองข้อมูลที่ประกอบไปด้วยสัญลักษณ์ต่างๆและเป็นวิธีที่มีอัลกอริทึมที่สามารถออกแบบฐานข้อมูลแบบรีเลชั่นนอลที่อยู่ในรูปของ 5 NF ได้โดยตรง

2.5.1 ขั้นตอนการออกแบบ

ขั้นตอนการออกแบบฐานข้อมูลโดยใช้ NIAM มีขั้นตอนการออกแบบอยู่ 9 ขั้นตอน

1. กำหนดขอบเขตของงาน (Universal of Discourse : UoD) และความจริงที่เกิดขึ้นภายในขอบเขตของงานที่กำหนดไว้
2. วาดคอนเซ็ปชวลสคีมาไดอะแกรม (Conceptual Schema Diagram) คร่าวๆ จากความจริงในขอบเขตของงาน
3. จัดรูปของ Schema ให้เป็นระเบียบและหาชนิดความจริงที่ได้รับข้อมูลมาตามชนิดความจริงอื่น
4. เติมสัญลักษณ์แสดงยูนิคเสสคอนสเตนส (Uniqueness Constrints)
5. ตรวจสอบความถูกต้องของชนิดความจริง
6. เติมสัญลักษณ์แสดงเล็กซิคอล (Lexical) , แมนดาทอรี่โรล (Mandatory Role), สับไทพท์คอนสเตรนท์ (Subtype constrints)
7. ตรวจสอบยูนิคไอเดนติไฟเออร์ (Unique Identifier) ของแต่ละเอนติตี้
8. เติมสัญลักษณ์แสดงอีกควอลิตี (Equality), เอ็กซ์คลูชั่น (Exclusion), สับเซตคอนสเตรนท์ (Subset constrints)
9. ตรวจสอบความสมบูรณ์ของคอนเซ็ปชวลสกีมา (Conceptual Schema) ว่าสอดคล้องกับตัวอย่างข้อมูลและไม่มีความซ้ำซ้อนของข้อมูล

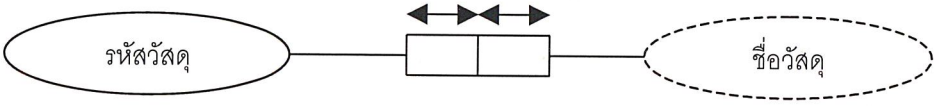
2.5.2 ส่วนประกอบของฐานข้อมูลใน NIAM

- ชนิดเอนติตี้ (Entity Type)
- ชนิดเลเบิล (Label Type)
- ชนิดความจริง (Fact Type)
- ชนิดอ้างอิง (Reference Type)
- ข้อจำกัดเพื่อความถูกต้องของข้อมูล (Integrity Constrints)

2.5.3 สัญลักษณ์และตัวอย่างการใช้ส่วนประกอบพื้นฐานของแบบจำลองไนแอม

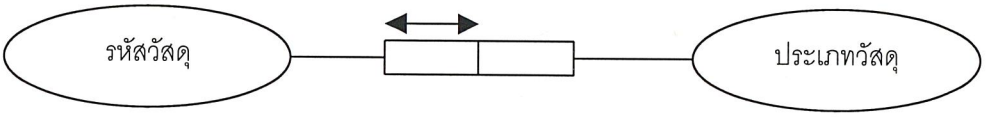
รูปที่ 2.8 แสดงสัญลักษณ์ของชนิดเอนติตี้วัสดุ

รูปที่ 2.9 แสดงสัญลักษณ์ของชนิดเลเบิและรหัสวัสดุ

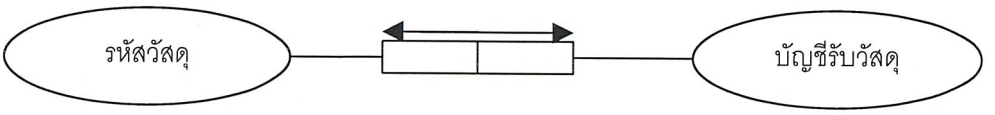


รูปที่ 2.10 แสดงความสัมพันธ์อ้างอิงแบบ วันทูวัน (one to one) หมายความว่า วัสดุใดๆ จะมีรหัสวัสดุเพียงรหัสเดียวเท่านั้นไม่ซ้ำกับวัสดุอื่น

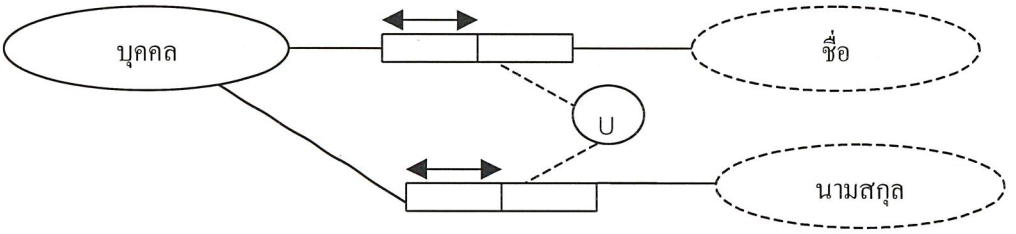
รูปที่ 2.11 แสดงการเขียนความสัมพันธ์อ้างอิงแบบ วันทูวัน (One to one) อย่างย่อ



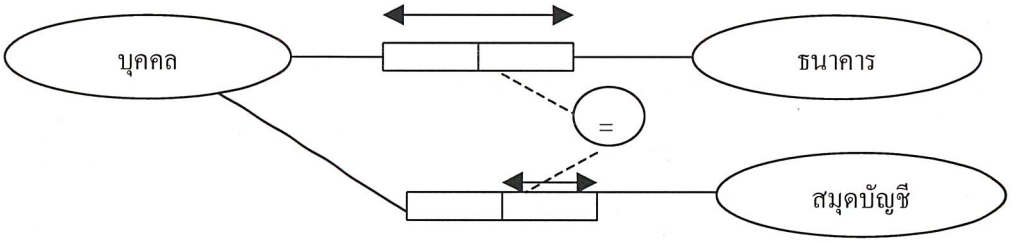
รูปที่ 2.12 แสดงความจริงแบบ แมนนีทูวัน (Many to one) หมายความว่า รหัสวัสดุประเภทใดๆ สามารถมีวัสดุในสังกัดได้มากกว่า 1 ชนิด



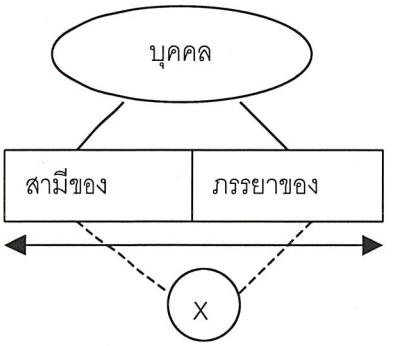
รูปที่ 2.13 แสดงความจริงแบบเมนี่ทูเมนี่ (many to many) หมายความว่า รหัสวัสดุหนึ่งๆสามารถอยู่ได้ในหลายๆ บัญชี และในบัญชีวัสดุหนึ่งๆ ก็สามารถมีรหัสวัสดุได้หลายๆ อย่าง



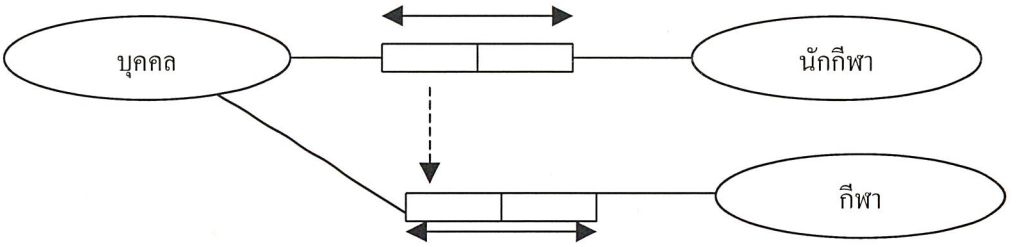
รูปที่ 2.14 แสดงการใช้ inter fac type uniqueness constraint หมายความว่า บุคคลหนึ่งจะมีชื่อ 1 นามสกุล 1 ซึ่งบางคนอาจซ้ำกันและนามสกุลของบางคนอาจจะซ้ำกันแต่ชื่อรวมกับนามสกุลจะต้องไม่ซ้ำกัน



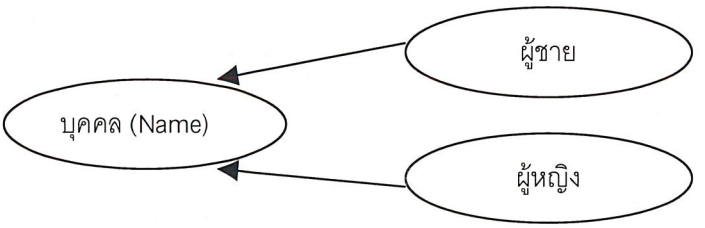
รูปที่ 2.15 แสดงการใช้อีควอลิตี้ คอนสเตรน (equality constrint) หมายความว่า ถ้าบุคคลหนึ่งเป็นลูกค้าของธนาคารใดแล้ว บุคคลนั้นต้องมีสมุดบัญชีของธนาคารนั้นด้วย หรือในทางกลับกันถ้าบุคคลใดมีสมุดบัญชีของธนาคารใดแล้วก็ต้องเป็นลูกค้าของธนาคารนั้นด้วย



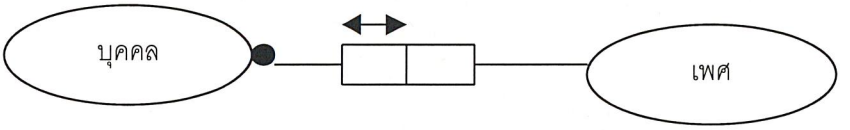
รูปที่ 2.16 แสดงการใช้เอ็กคลูสชั่น คอนสเตรนท์ (Exclusion constraint) หมายความว่าบุคคลใดเป็นภรรยาของอีกบุคคลหนึ่งแล้วบุคคลนั้นต้องไม่เป็นสามีของบุคคใดๆ ในทางกลับกันบุคคลที่เป็นสามีของบุคคลหนึ่งแล้วจะต้องไม่เป็นภรรยาของบุคคลใดด้วย



รูปที่ 2.17 แสดงการใช้สับเซตคอนสเตรท (Subse constraint) หมายความว่า บุคคลที่ชนะการแข่งขันกีฬาทุกคนจะตองเป็นบุคคลที่เล่นกีฬา แต่บุคคลที่เล่นกีฬาไม่จำเป็นต้องนะการแข่งขันกีฬาทุกคน



รูปที่ 2.18 แสดงการใช้สับไทพท์คอนสเตรท (Subtype constrint) หมายความว่า ตัวอย่างเอนติตี้ทุกตัวของชนิดเอนติตี้ผู้ชาย และชนิดเอนติตี้ผู้หญิงต่างก็เป็นสมาชิกของเอนติตี้บุคคล



{ M, F }

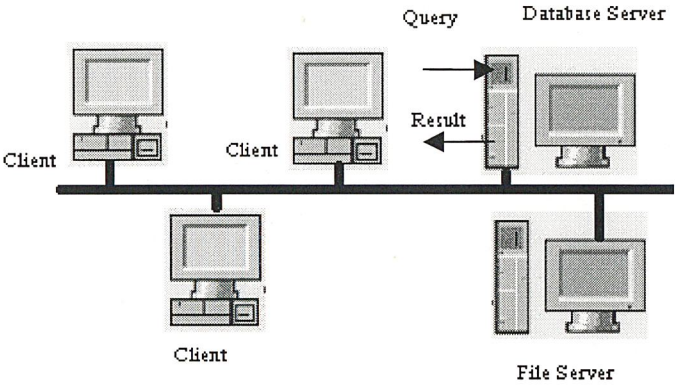
รูปที่ 2.19 แสดงการใช้เมนดาทอรี คอนสเตรนท และเล็กซิคอล คอนสเตรท หมายความว่า บุคคลทุกคนต้องมีเพศ และเป็นสมาชิกของเอนติตี้เพศมีเพียง M (Male) และ F (Female) เท่านั้น

2.6 หลักการของไคลเอ็นท์เซอร์ฟเวอร์

ไคลเอ็นท์เซอร์ฟเวอร์ เป็นโครงสร้างของระบบคอมพิวเตอร์รูปแบบหนึ่งที่แบ่งแยกการประมวลผลข้อมูลออกเป็น 2 ระบบ โดยฝั่งไคลเอ็นท์ (ผู้ขอใช้บริการ) จะมีระบบฟรอนท์เอ็นด์ (Front End System) หรือส่วนดาต้าเบสแอพพลิเคชั่นทำงานอยู่ และฝั่งดาต้าเบสเซอร์ฟเวอร์ (ผู้ให้บริการ) จะมีระบบแบคเอ็นด์ (Back End System) หรือส่วนที่ทำหน้าที่เป็น ดีบีเอ็มเอสจริงๆ ทำงานอยู่ ซึ่งระบบฟรอนท์เอ็นด์นี้จะจัดการการประมวลผลสำหรับหน้าจอ และอินพุต เอาท์พุทของผู้ใช้ และระบบแบคเอ็นด์จะจัดการการประมวลผลข้อมูลและทำการดิสต์แอคเซส (Disk Access) เช่น เมื่อผู้ใช้ระบบฟรอนท์เอ็นท์สร้าง

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

คิวรี่ (Query) เพื่อสอบถามจากดาต้าเบสเซอร์เวอร์โดยผ่านระบบเครือข่าย ส่วนเซอร์ฟเวอร์ก็จะทำการค้นหาข้อมูลที่ผู้ใช้ต้องการแล้วส่งข้อมูลกลับไปให้ดังรูปที่ 2.20



รูปที่ 2.20 รูปแสดงความสัมพันธ์ระหว่างไคล์เอนท์กับเซิร์ฟเวอร์

วัตถุประสงค์หลักของระบบไคลเอ็นท์เซอร์ฟเวอร์ คือการอนุญาตให้แอพพลิเคชันของผู้ขอใช้บริการเข้ามาเรียกใช้ข้อมูลที่ถูกจัดการโดยผู้ให้บริการได้ โดยผู้ให้บริการสามารถรันอยู่ในเครื่องที่ตั้งอยู่ในที่ห่างไกลกับเครื่องที่ผู้ขอใช้บริการรันอยู่

โดยทั่วไปแล้วระบบไคลเอ็นท์จะถูกใช้งานกับพีซี และส่วนดาต้าเบสเซอร์ผเวอร์สามารถทำงานบนเครื่องใดก็ได้ตั้งแต่พีซีไปจนถึงเมนเฟรม ในโครงงานนี้อาศัยระบบจัดการฐานข้อมูลโดยใช้ไมโครซอฟท์เอสคิวแอลเซิร์ฟเวอร์ เวอรชัน 7 (Microsoft SQL Server Version 7.0) เป็นแบคเอ็นท์ของระบบ ทำหน้าที่ในการประมวลผลข้อมูล โดยอาศัยเดลไฟล เวอร์ชัน 4 (Delphi Versoion 4) เป็นฟรอน์ทเอนด์ สำหรับสร้างแอพพลิเคชั่นเพื่อทำงานกับข้อมูลในฐานข้อมูลผ่านโอดีบีซี (Open Database Connectivity : ODBC)

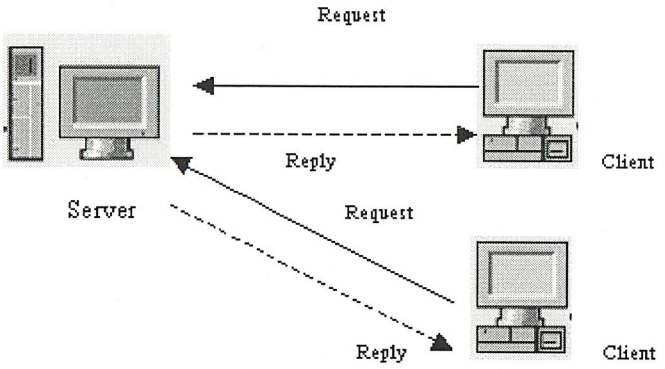
2.6.1 ความรู้พื้นฐานเกี่ยวกับระบบไคล์เอนท์เซิร์ฟเวอร์

โดยทั่วไปความหมายของไคล์เอนท์เซิรฟ์เวอร์ หมายถึง ความสัมพันธ์ระหว่าง 2 ระบบ หรือกระบวนการโดยไคล์เอนท์ หมายถึง ระบบที่ทำการร้องขอการบริการจากส่วนเซิร์ฟเวอร์ โดยเปรียบไคล์เอนท์เป็นๆผู้ร้องขอ (Requester) ทำการขอการบริการ (Service) จากส่วนเซิร์ฟเวอร์

เซิร์ฟเวอร์เป็นผู้ให้บริการตามที่ไคล์เอนท์ร้องขอ โดยในเซิรฟ์เวอร์ 1 เซิร์ฟเวอร์สามารถให้บริการกับไคล์เอนท์ได้มากกว่า 1 การบริการ

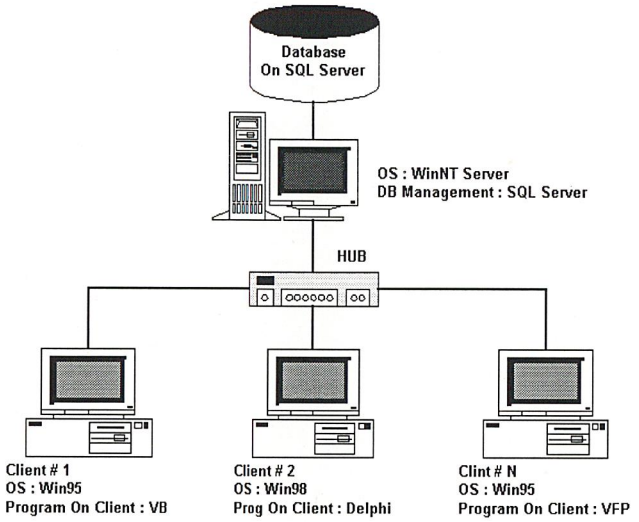
ลักษณะความสัมพันธ์ระหว่างไคล์เอนท์และเซิรฟ์เวอร์ แสดงดังรูป

37081



รูปที่ 2.21 รูปแสดงความสัมพันธ์ระหว่างไคล์เอนท์กับเซิรฟ์เวอร์

ในการทำโครงงานการเชื่อมต่อระบบไคล์เอนท์เซิรฟ์เวอร์ มุ่งเน้นการเชื่อมต่อในรูปแบบของเซิร์ฟเวอร์ฐานข้อมูล (Database Server) ซึ่งมีรูปแบบลักษณะการเชื่อมต่อแสดงดังรูป



รูปที่ 2.22 รูปแสดงสถาปัตยกรรมของเซิร์ฟเวอร์ฐานข้อมูล

จากรูปแสดงความ สัมพันธ์ของระบบไคล์เอนท์เซิร์ฟเวอร์ในลักษณะของเซิร์ฟเวอร์ฐานข้อมูล และแสดงถึงการเคลื่อนที่ของข้อมูล และการควบคุมในระบบไคล์เอนท์เซิร์ฟเวอร์แบบฐานข้อมูล (Database Clien/Server)

ในส่วนแอพพลิเคชันโปรแกรมประกอบด้วย การร้องขอไปยังฐานข้อมูล (Database Access Request) เช่น คำสั่งภาษาเอสคิวแอล (SQL Statement) ในระหว่งการคอมไพล์ (Compile) และการเชื่อมโยง (Link) ส่วนประกอบต่างๆ ของโปรแกรมนั้นรีโมทดาต้าแอคเซสสตับ (Remote Database Acess Stub) จากไลบรารีของฐานข้อมูล (Database Application Library) จะทำการเชื่อมต่อกันเป็นโปรแกรมที่สมบูร์ จากนั้นรีโมทดาต้าเบสอินเตอร์เฟล (Remote Database Interface) จะทำการจัดรูปแบบของการร้อง

ขอทำงานกับฐานข้อมูล (Database Access Request) เพื่อส่งผ่านไปตามเครือข่าย (Network) โดยผ่านในส่วนของการเชื่อมต่อระบบเครือข่าย (Network Interface) ไปยังส่วนเซิร์ฟเวอร์

ในส่วนของเซิร์ฟเวอร์นั้น ส่วนของการเชื่อมต่อระบบเครือข่ายจะทำหน้าที่รับข่าวสาร (Message) ที่ส่งมาจากส่วนไคล์เอนท์ และนำส่งต่อไปให้ยังเซิร์ฟเวอร์ฐานข้อมูลเพื่อทำการประมวลผลตามการร้องขอที่ได้รับและส่งผลลัพธ์กลับคืนสู่แอพพลิเคชันโปรแกรมที่ร้องขอในส่วนไคล์เอนท์โดยผ่านระบบเครือข่าย

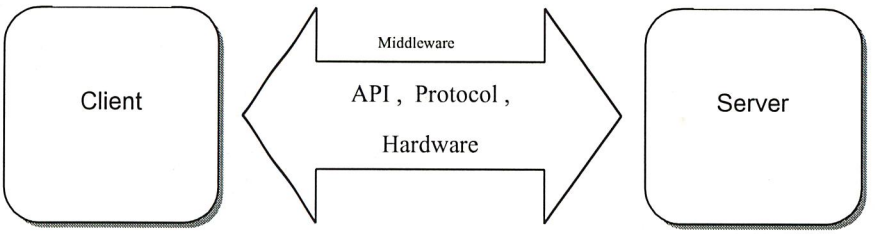
ดังนั้นจากข้อมูลข้างต้นเราสามารถแบ่งส่วนประกอบต่าง ๆ ภายในระบบไคล์เอนท์เซิร์ฟเวอร์ ออกได้เป็น 3 ส่วนประกอบหลักได้แก่

1. ส่วนไคล์เอนท์ ทำงานบนอุปกร์ไคลเอนท์ (Client Hardware) เช่น แสดงรายงาน และทำการคำนวณเบื้องต้น เป็นต้น โดยจะทำการส่งงานที่มีความซับซ้อนไปยังเซิร์ฟเวอร์ เช่น ไคล์เอนท์ทำการเรียกข้อมูลจากเซิรฟ์เวอร์ แล้วทำการเก็บข้อมูลที่เรียกไว้ให้กับผู้ใช้ และอนุญาตให้ผู้ใช้ทำงานกับข้อมูลเหล่านั้นได้ เช่น Search, Graph, Import และอื่นๆ ในการแก้ไขข้อมูลที่เรียกมานั้น ในบางครั้งเป็นการแก้ไขแบบชั่วคราว หรือบางครั้งจะส่งข้อมูลที่แก้ไขแล้วกลับไปที่เซิร์ฟเวอร์ ซึ่งจะเป็นการแก้ไขข้อมูลอย่างถาวร

ไคล์เอนท์โปรแกรม (Client Program) จะมีส่วนที่ทำการติดต่อกับมิดเดิ้ลแวร์โดยปกติ คือ เอพีไอ (Application Program Interface) ซึ่งจะเป็นส่วนที่อนุญาตให้โปรแกรมสามารถทำการติดต่อสื่อสารกับมิดเดิ้ลแวร์ได้ โดยเอพีไอที่นิยมใช้ได้แก่ โอดีบีซี (Open Database Connectivity : ODBC)

2. ส่วนมิดเดิ้ลแวร์ ทำงานอยู่ระหว่างไคล์เอนท์ และเซิร์ฟเวอร์ ซึ่งนับได้ว่าเป็นส่วนที่ซับซ้อนมากที่สุด โดยทั่วไปมิดเดิ้ลแวร์ หมายถึง ซอฟท์แวร์ที่ทำให้ไคล์เอนท์และเซิร์ฟเวอร์สามารถสื่อสารกันได้ (เรียกว่า Glue) มิดเดิ้ลแวร์โดยปกติมีหลายชั้น เช่น ถ้าใน Delphi ไคล์เอนท์ที่ต้องการเชื่อมต่อกับเอสคิวแอลเซิร์ฟเวอร์จะต้องอาศัยมิดเดิ้ลแวร์ที่ประกอบด้วย Winsock, TCP/IP, ODBC และ ODBC Driver

โดยปกติแล้ว ผู้ใช้จะไม่สามารถมองเห็นมิดเดิ้ลแวร์ ซึ่งในมิดเดิ้ลแวร์จะประกอบไปด้วยกลุ่มของไดร์ฟเวอร์ และโปรแกรม



รูปที่ 2.23 รูปแสดงลักษณะของมิดเดิ้ลแวร์

API เช่น Messageing API, ODBC API

Protocol เช่น TCP/IP, Winsock

Hardware เช่น Ethernet, Token Ring

เอพีไอจะถูกเรียกโดยไคล์เอนท์และจะทำการติดต่อกับเซิร์ฟเวอร์ โดยอาศัยโปรโตคอล (Protocol) ผ่านฮาร์แวร์ (Hardware)

3. ส่วนเซิร์ฟเวอร์ จะทำงานที่ซับซ้อน และทำการคำนวณเป็นหลัก โดยทั่วไปแล้วเซิร์ฟเวอร์หมายถึงฮาร์ดแวร์ที่ทำการทำงานโปรแกรมเซิร์เวอร์ (Run Server Program) ซึ่งตอบสนองคำร้องขอผ่านระบบเครือข่าย เช่น เซิร์ฟเวอร์ฐานข้อมูลรับคำขอข้อมูลจากไคลเอนท์ และทำการส่งผลลัพธ์ที่ได้จากคำขอมายังไคล์เอนท์ เป็นต้น

ในส่วนของเซิร์ฟเวอร์มักจะใช้งานยาก รูปแบบที่แสดงไม่สื่อความหมายเนื่องจากไม่มีการติดต่อกับผู้ใช้โดยตรง แต่มีเพียงการติดต่อกับผ่านมิดเดิ้ลแวร์ และระบบเครือข่ายเท่านั้น

2.6.2 ข้อดีของระบบไคลเอ็นท์เซอร์ฟเวอร์

ข้อดีอย่างแรกของระบบไคลเอ็นท์เซอร์เวอร์เกิดจากการการแบ่งการประมวลผลระหว่างไคลเอ็นท์และเซอร์เวอร์ เพราะการประมวลผลฐานข้อมูลถูกกระทำที่ฝั่งแบคเอ็นต์ ดังนั้นทำให้ความเร็วของดีบีเอ็มเอสไม่ถูกยึดติดกับความเร็วของเวอร์คสเตชั่น

นอกจากนี้เวอร์สเตชั่นยังต้องการเพียงให้สามารถวิ่งฟรอนท์เอ็นด์ซอฟแวร์ให้ได้เท่านั้นจึงช่วยยืดอายุของเครื่องพีซีรุ่นเก่า เพราะไม่จำเป็นต้องใช้เครื่องที่มีความสามารถสูงๆ

การแบ่งดังกล่าวยังช่วยลดทราฟฟิกของระบบเครือข่าย เนื่องจากระบบไม่ต้องส่งไฟล์ฐานข้อมูลทั้งหมดกลับไปกลับมา จึงทำให้การติดต่อกันระหว่างไคลเอ็นท์และเซอร์เวอร์เป็นไปอย่างรวดเร็วด้วย

นอกเหนือจากเรื่องประสิทธิภาพแล้ว การแบ่งดังกล่าวทำให้ผู้ใช้ไม่ต้องยึดติดกับระบบหรือแพลิ์ทฟอร์มใดแบบหนึ่ง ในระบบไคลเอ็นท์เซอร์ฟเวอร์ เวอร์สเตชั่นอาจเป็นคอมแพททิเบิลพีซี (Compatible PC), แมคอินทอช (Machintoshe), ยูนิกส์ (Unix) หรือหลายๆ แบบรวมกันก็ได้ นอกจากนี้อาจวิ่งโปรแกรมปฏิบัติการที่ไม่เหมือนกันก็ได้

ข้อดีอีกอย่างหนึ่งของไคลเอ็นท์เซอร์ฟเวอร์ ก็คือการสามารถรักษาความถูกต้องของข้อมูลได้เนื่องจากปัจจุบันนี้ ดาต้าเบสเซอร์ฟเวอร์จะวิ่งดีบีเอ็มเอสที่ใช้ดาต้าโมเดลแบบรีเลชั่นนอล และผู้ใช้จะถูกกั้นไม่ให้สามารถเข้าถึงข้อมูลได้โดยตรงโดยไม่ผ่านดีบีเอ็มเอส และดีบีเอ็มเอสก็สามารถใช้บริการอื่นๆ ที่ช่วยรักษาความปลอดภัยของข้อมูลได้ดีด้วย ไม่ว่าจะเป็นการเข้ารหัสข้อมูล การทำเรียลไทม์แบ็กค์อัๆ (Realtime backup) ข้อมูลลงสู่เทป เป็นต้น

2.6.3 ข้อเสียของระบบฐานข้อมูลไคลเอ็นท์เซอร์ฟเวอร์

ข้อเสียหลักๆ ของระบบไคลเอ็นท์เซอร์ฟเวอร์ ก็อยู่ที่ค่าใช้จ่ายที่เพิ่มขึ้นในการจัดการระบบ และการหาบุคคลากรที่มาคอยดูแลดาต้าเบสเซอร์ฟเวอร์ บนระบบเครือข่ายขนาดเล็ก ผู้ควบคุมเครือข่าย (Network Administrator) อาจเป็นผู้ดูแลดาต้าเบสเซอร์ฟเวอร์เองเลยก็ได้ แต่อย่างไรก็ตามเมื่อมีจำนวนผู้ใช้มากขึ้นหรือตัวดาต้าเบสเองโตขึ้นเรื่อยๆ ก็ควรจะต้องมีผู้มาดูแลเฉพาะ

นอกจากนี้ยังต้องมีค่าใช้จ่ายเพิ่มเติมในด้านฮาร์ดแวร์ด้วย คือ ถึงแม้ว่าผู้ขายดีบีเอ็มเอสแบบไคลเอ็นท์เซอร์ฟเวอร์จะอ้างว่ามันสามารถวิ่งอยู่บนฮาร์ดแวร์ตัวเดียวกันกับไฟล์เซอร์ฟเวอร์ได้เลยก็ตาม ดาต้าเบสเซอร์ฟเวอร์ก็ควรจะถูกวิ่งบนเครื่องที่อุทิศเพื่องานนี้โดยเฉพาะเพื่องานนี้โดยเฉพาะเพื่อเพิ่มความมั่นใจในเรื่องประสิทธิภาพและความถูกต้องและปลอดภัยของข้อมูล

ราคาของซอฟท์แวร์ไคลเอ็นท์เซอร์ฟเวอร์ก็มักจะมีราคาสูงกว่าโปรแกรมโดยทั่วไป และจะต้องมีค่าใช้จ่ายในด้านการฝึกอบรมโปรแกรมเมอร์ (Programmer) เพื่อทำงานกับระบบใหม่อีกด้วย นอกจากเรื่องต่างๆ ที่กล่าวมาแล้ว ความซับซ้อนของระบบไคลเอ็นท์เซอร์ฟเวอร์ก็เป็นอีกเรื่องหนึ่งที่ต้องคำนึงด้วย เมื่อระบบมีองค์ประกอบมาก จึงทำให้เป็นการยากมากขึ้นในการตรวจหาข้อผิดพลาดที่อาจเกิดขึ้น และการติดตั้งระบบแบบไคลเอ็นท์เซอร์ฟเวอร์ก็ยากกว่าด้วยเหตุผลในทำนองเดียวกัน

2.7 โอเพนดาต้าเบสคอนเน็คติวิตี (โอดีบีซี) (Open Database Connectivity, ODBC)

โอดีบีซี คือ วิธีการติดต่อและเข้าถึงจากแอพพลิเคชั่นสู่ระบบจัดการฐานข้อมูล (DBMS) โดยใช้ภาษาแอสคิวแอล เป็นมาตราฐานการเข้าถึงข้อมูลความสามารถในการเชื่อมต่อแบบนี้ทำให้แอพพลิเคชั่นสามารถเข้าถึงฐานข้อมูลได้หลายอยู่แบบ ซึ่งทำให้ผู้พัฒนาโปรแกรมสามารถพัฒนาโปรแกรมไปได้โดยไม่ต้องทำการระบุชนิดของระบบจัดการฐานข้อมูล

แต่เดิมนั้นการพัฒนาโปรแกรมประยุกต์ที่ใช้งานเกี่ยวกับฐานข้อมูล การเข้าใช้ฐานข้อมูลโปรแกรมเหล่านี้จะทำผ่านการเรียกใช้ เอ็มเบทเด็ตเอสคิวแอล (Embedded SQL) ซึ่งในขณะนั้นวิถีทางแบบนี้ก็ดูจะไปได้ทีเดียว เพราะว่าตัวแปรสามารถทำการเปลี่ยนรูปแบบของระบบไม่ว่าจะเป็นทางด้านฮาร์แวร์หรือซอฟท์แวร์ได้หลายแบบรวมทั้งระบบปฏิบัติการด้วย (โดยการคอมไพล์ใหม่ทุกครั้งที่มีการย้ายระบบ)

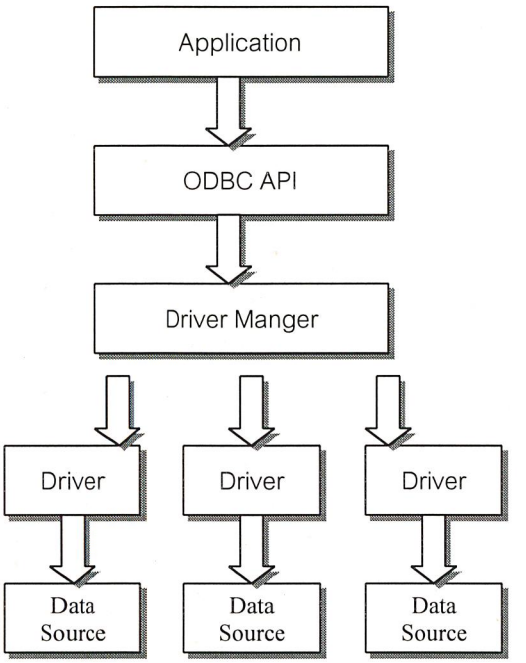
อย่างไรก็ตามในการพัฒนาโปรแกรมในระบบที่มีความแตกต่างกัน เช่น การเรียกใช้ข้อมูลของออรา -เคิลจากไมโครซอฟท์เอ็กเซล (Microsoft excel) วิธีการเข้าถึงข้อมูลแบบเดิมนั้นจะต้องทำการพรีคอมไพล์โค้ดของเอ็กซ์เซลและออราเคิลโดยใช้ ไอบีเอ็มพรีคอมไพเลอร์ (IBM Precompiler) และออราเคิลพรีคอมไพเลอร์ (Oracle Precompiler) ตามลำดับ ซึ่งจะเห็นว่าเป็นการยุ่งยากมากทีเดียว

วิธีการต่อเชื่อมแบบโอดีบีซีจะให้ความสะดวกในการติดต่อข้อมูลมากกว่าวิธีการดั้งเดิม โดยการกำหนดมาตราฐานการต่อเชื่อมของข้อมูล (Data Protocol, DBMS capability) และแนวทางนี้ทำให้เกิดความคิดที่จะสร้างไดร์ฟเวอร์การติดต่อกับงานข้อมูลขึ้นมา (DLL)

หมายเหตุ DLL (Dynamically Linked Libraries) เป็นไดรฟ์เวอร์ที่ทำหน้าที่แปลงชุดคำสั่งโอดีบีซีเอพีไอไปเป็นฟังก์ชันที่ดาต้าซอสสามารถเรียกใช้งานได้ โดยมากมักจะแปลงจากรูปแบบของเอสคิวแอลในโอดีบีซีไปเป็นรูปแบบที่ดาต้าซอสเข้าใจ

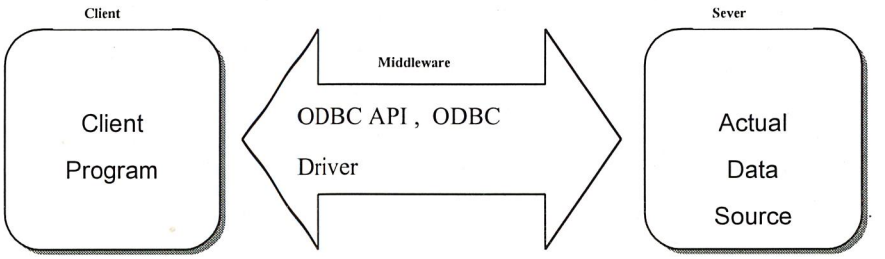
โอดีบีซีเป็นมาตราฐานสำหรับการสร้างดาต้าเบสไดร์ฟเวอร์ (Database Driver) ตามรูป โดยแอพพลิเคชั่นโปรแกรมจะใช้งานโอดีบีซีผ่านคำสั่งมาตราฐานคือ โอดีบีซีเอพีไอ (Open Database Connectivity Application Program Interface : ODBCAPI) โอดีบีซีเอพีไอไดร์ฟเวอร์ใช้ตามดาต้าซอสที่จะใช้แอพพลิเค

ชั่นโปรแกรม โดยใช้คำสั่งโอดีบีซี และสามารถใช้งานดาต้าซอสอื่นๆ ที่มีโอดีบีซีไดร์ฟเวอร์ได้ โดยไม่จำเป็นต้องเปลี่ยนแปลงโปรแกรม



รูปที่ 2.24 รูปแสดงลักษณะทั่วๆ ไปของโอดีบีซี

มาตราฐานของโอดีบีซีสร้างด้วยไมโครซอฟท์ (Microsoft) โดยโอดีบีซีจะทำงานเป็นมิดเดิ้ลแวร์ระหว่างแอพพลิเคชั่นและดาต้าซอส โดยผ่านโอดีบีซีไดร์ฟเวอร์ ดังรูป



รูปที่ 2.25 รูปแสดงแผนภาพโอดีบีซีไดร์ฟเวอร์ (ODBC Driver Diagram)

โอดีบีซีจะมีเอพีไอมาตราฐานเพื่อช่วยในการเข้าถึงดาต้าซอสหลายๆ ชนิดได้พร้อมๆ กัน โดยโอดีบีซีมีความสามารถที่เรียกว่า System Table Transparency (System Table ถูกแก้ไขโดยอัตโนมัติโดย Driver) Schrollable Cursor, Transactions, Asynchronous Calling, Array Fetch และ Update, Dynamic Driver Information และ Stored Procedure สำหรับ SQL Database

โอดีบีซีเป็นมาตราฐานเอสคิวแอล (SQL Based) ซึ่งยอมรับการคิวรี (Query) ทั้งหมดผ่านคำสั่งภาษาเอสคิวแอล ซึ่งใช้ได้กับ Scalar Function, Outer Joins, Stored Procedures, Complex Data Type (เช่น Date, Time, Time Stamp และ Binary) และ SQL Pass – Through เพื่อใช้งาน System Function

2.7.1 ข้อดีของการติดต่อโดยใช้โอดีบีซี

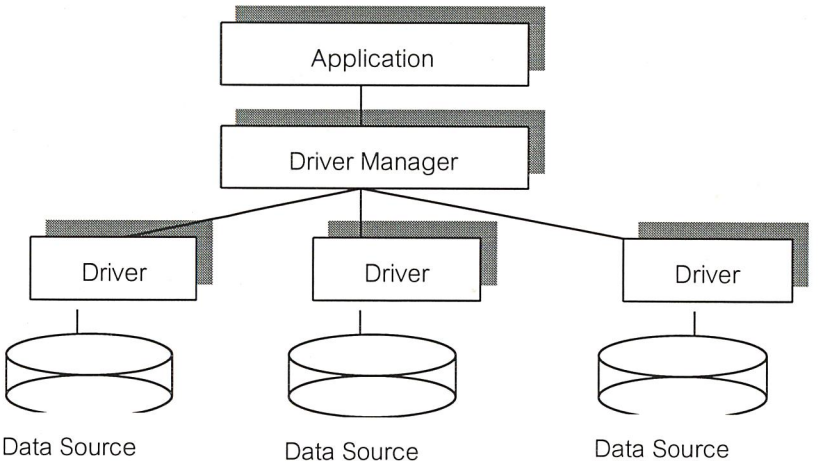
1. ฟังก์ชั่นของโอดีบีซีอนุญาตให้แอพพลิเคชั่นติดต่อกับดีบีเอ็มเอสได้โดยสะดวก (การทำคำสั่งเอสคิวแอลและการรับผลลัพท์)
2. ใช้ภาษาเอสคิวแอลตามมาตราฐานเอสคิวแอล ซีเออี (SQL CAE), เอ็กซ์/ โอเพน (X/Open) และเอสคิวแอลแอกเซสกรุ๊ป (SQL Access Group, SAG)
3. มีการกำหนดการส่งกลับรหัสความผิดพลาด (Error Code) เป็นมาตราฐานเดียวกัน
4. เป็นวิธีการมาตราฐานในการติดต่อกับดีบีเอ็มเอส
5. มีการกำหนดชนิดของข้อมูล (Data Type) เป็นมาตราฐาน
6. ชุดคำสั่งเอสคิวแอล สามารถกำหนดได้แม้ในขณะรัน
7. สามารถเขียนโปรแกรมชุดเดียวแต่สามารถเข้าใช้ดีบีเอ็มเอสได้หลายตัว
8. ตัวโปรแกรมไม่ต้องรับผิดชอบในการดูแลการติดต่อข้อมูลกับดีบีเอ็มเอส
9. ค่าข้อมูลสามารถถูกส่งหรือรับได้ในรูปแบบที่สะดวกขึ้น

2.7.2 การติดต่อโดยใช้โอดีบีซี (ODBC Interface) การใช้โอดีบีซีในการติดต่อนั้นจำเป็นต้องมี

1. ไลบรารี่ของโอดีบีซีฟังก์ชันคอล ซึ่งจะทำให้แอพพลิเคชันติดต่อกับดีบีเอ็มเอสเพื่อที่จะเอ็กซิคิวท์คำสั่งแล้วดึงข้อมูลขึ้นมา
2. เซตมาตราฐานของโค้ดที่แสดงข้อผิดพลาด
3. กฎเอสคิวแอล (SQL syntax) บน X/Open และ ข้อกำหนดเอสคิวแอลแอคเซสกรุ๊ป (SQL Acess Group, SAG) เอสคิวแอลซีเออี (SQL CAE)
4. วิธีมาตราฐานในการติดต่อและการล็อคเข้าดีบีเอ็มเอส

2.7.3 ส่วนประกอบของโอดีบีซี

โครงสร้างของโอดีบีซีจะประกอบด้วยส่วนประกอบ 4 ส่วนดังนี้



รูปที่ 2.26 รูปแสดงส่วนประกอบต่างๆ ใน โอดีบีซี (ODBC Component)

1. แอพพลิเคชั่นทำหน้าที่ประมวลผลและเรียกใช้ฟังก์ชันของ โอดีบีซีตามคำสั่งภาษาแอสคิวแอล พร้อมทั้งทำการรับผลลัพธ์ด้วย
2. ไดรเวอร์เมเนเจอร์ทำหน้าที่โหลด (load) ไดร์ฟเวอร์ เชื่อมต่อกับแหล่งข้อมูล
3. ไดร์เวอร์ (Driver) ทำหน้าที่ประมวลผลการเรียกใช้ฟังก์ชันของโอดีบีซีส่งคำสั่งแอสคิวแอลไปสู่แหล่งข้อมูลที่ถูกต้องและทำการส่งผลลัพธ์กลับให้แอพพลิเคชั่นและในบางครั้งไดร์เวอร์ จะทำหน้าที่แปลงคำสั่งที่ส่งมาให้อยู่ในรูปที่สนับสนุนโดยระบบฐานข้อมูล แต่ละชนิดอีกด้วย
4. ดาต้าซอส (Data Source) ประกอบด้วยข้อมูลที่ผู้ใช้ต้องการจะเข้าถึงระบบปฏิบัติการที่เกี่ยวข้อง ดีบีเอ็มเอสและเน็ตเวิร์คแพลตฟอร์ม (Network Platform) (ถ้ามี)

2.7.3.1 แอพพลิเคชัน

แอพพลเคชันจะใช้การติดต่อกับโอดีบีซี ทำได้ดังนี้

1. ทำการร้องขอ (Request) ไปยังดาต้าซอส (Data source)
2. ส่งเอสคิวแอลไปยังดาต้าซอส
3. กำหนดเนื้อที่และรูปแบบข้อมูลสำหรับเก็บผลลัพธ์จากเอสคิวแอลรีเควสซ์ (SQL Request)
4. ร้องขอผลลัพธ์
5. ประมวลผลข้อผิดพลาดที่อาจจะเกิดขึ้น
6. รายงานผลลัพธ์กลับไปยังผู้ใช้ (ถ้าจำเป็น)
7. ร้องขอการคอมมิท (Commit) และโรวแบค (Roll Back) สำหรับการควบคุมทรานแสคชัน (Transaction)
8. ยกเลิกการติดต่อกับแหล่งข้อมูล

2.7.3.2 ไดรเวอร์เมเนเจอร์

ไดรเวอร์เมเนเจอร์เป็นไดนามิกลิงส์ไลบรารี่ (ดีแอลแอล) (Dynamic Link Library, DLL) จุดประสงค์หลักของไดรเวอร์เมเนเจอร์ คือ การโหลดไดรเวอร์ โดยจะทำตามลำดับดังนี้

1. ใช้ ODBC.INI ไฟล์เพื่อแมพชื่อของดาต้าซอสเพื่อค้นหาไดรเวอร์ดีแอลแอล
2. ทำการประมวลผลโอดีบีซีอินนิเชียลไลเซชันคอล (ODBC Initialization Call)
3. จัดการโอดีบีซีฟังก์ชันสำหรับแต่ละไดรเวอร์
4. จัดการพารามิเตอร์ให้ถูกต้องสำหรับการเรียกใช้โอดีบีซี

2.7.3.3 ไดร์เวอร์

ไดร์เวอร์เป็นดีแอลแอลที่อิมพลีเมนต์โอดีบีซีฟังก์ชันคอลและทำการติดต่อกับข้อมูลที่ดาต้าซอส โดยไดร์เวอร์เมเนเจอร์จะทำการโหลดไดรเวอร์ เมื่อแอพพลิเคชั่นได้เรียกฟังก์ชัน เอสคิวแอลบราวส์คอนเนีคท์ (SQLBrowseConnect) เอสคิวแองคอนเน็คท์ (SQLConnect) หรือ เอสคิวแอลไดรฟ์เวอร์คอนเน็คท์ (SQLDriverConnect) โดยไดรเวอร์จะกระทำตามขั้นตอนดังต่อไปนี้เพื่อตอบรับการเรียกโอดีบีซีฟังก์ชันของแอพพลิเคชั่น

1. สร้างการติดต่อไปยังดาต้าซอส
2. ส่งการร้องขอไปยังดาต้าซอส
3. แปลงข้อมูลให้เป็นรูปแบบอื่นถ้าเป็นการเรียกโดยแอพพลิเคชั่น
4. ส่งผลลัพธ์กลับไปยังแอพพลิเคชั่น ถ้าเกิดข้อผิดพลาดขึ้นจะส่งโค้ดข้อผิดพลาดกลับไปยังแอพพลิเคชั่นที่เรียกมา

2.8 Borland Delphi

การสร้างแอปพลิเคชั่นบน Windows ในยุคปัจจุบัน สามารถทำได้ง่ายและสะดวกกว่าในยุคแรกๆที่จะต้องเขียนโค้ดใส่ไว้ในไฟล์เป็นจำนวนมาก แล้วจึงค่อยสั่งให้แปลภาษา และยุ่งยากในการแก้ไขข้อผิดพลาดของโปรแกรม อีกทั้งยากที่จะเห็นผลลัพธ์ที่ได้ว่าจะเป็นไปตามที่ต้องการหรือไม่ แต่ในยุคนี้จะมีเครื่องมือในการสร้างแอปพลิเคชั่นมากมาย ที่สามารถเห็นผลได้ตั้งแต่ในขณะที่กำลังสร้าง มีทั้งที่ใช้ภาษา C ภาษา Basic และภาษา Pascal นั่นก็คือ Delphi นั้นเอง

Delphi สามารถใช้พัฒนาแอปพลิเคชั่นได้ในทุกระดับทุกประเภทครบถ้วนในตัวเดียว โดยไม่ต้องใช้ภาษาอื่นเพิ่มเติมอีกเลย ไม่ว่าจะเป็นการสร้างโปรแกรมง่ายๆสำหรับมือสมัครเล่น หรือใช้ทำงานที่สลับซับซ้อนสำหรับมืออาชีพ อีกทั้งยังรองรับระบบปฏิบัติการทั้ง Windows 95/98 และแม้แต่ Windows NT

2.8.1 คุณสมบัติต่างๆที่มีอยู่ใน Delphi

Delphi เป็นเครื่องมือสำหรับการพัฒนาโปรแกรม ที่มีสิ่งอำนวยความสะดวกไว้อย่างครบถ้วน โดยมีสภาพแวดล้อมในการทำงานที่ช่วยให้สามารถทำทุกอย่างได้จากใน Delphi เอง มีเครื่องมือทุกชนิดที่จำ

เป็นสำหรับการสร้างแอปพลิเคชั่นบน Windows ทั้งในส่วนติดต่อกับผู้ใช้ การแสดงผลกราฟิก การติดต่อกับฐานข้อมูล การจัดการระบบ ตลอดจนการพัฒนาแอปพลิเคชั่นเพื่อทำงานบนอินเตอร์เน็ต และด้วยคุณสมบัติที่มอยู่มากมายของ Delphi จึงพอจะแยกออกเป็นข้อๆดังนี้

2.8.1.1 วิชวลโปรแกรมมิ่ง

การพัฒนาโปรแกรมแบบวิชวล คือการพัฒนาโดยให้เห็นผลที่จะเกิดขึ้นเมือรันโปรแกรมได้ตั้งแต่ในขณะที่กำลังสร้าง โดยการนำชิ้นส่วนต่างๆที่ต้องการได้แก่ ปุ่ม (Button), ข้อความ (Label), รูปภาพ (Image) ฯลฯ ซึ่งเหล่านี้เรียกรวมๆว่า คอมโพเนนต์ (Component) นำมาวางบนวินโดว์ที่เรียกว่า ฟอร์ม (Form) ปรับขนาดและตำแหน่งรวมทั้งคุณสมบัติต่างๆของคอมโพเนนต์ และแม้แต่ฟอร์มเองให้ได้ตามที่ต้องการ และการเปลี่ยนคุณสมบัติเหล่านี้จะมีผลตั้งแต่ในขณะที่กำลังออกแบบ และเมื่อรันโปรแกรมก็จะได้ผลลัพธ์เหมือนกับที่เห็นในขณะออกแบบ

2.8.1.2 การโปรแกรมเชิงวัตถุ (Object Oriented Programming – OOP)

การโปรแกรมเชิงวัตถุเป็นการพัฒนาโปรแกรมโดยการสร้าง วัตถุ หรือ ออบเจ็ค (Object) ที่ต้องการ ในมุมมองของตัววัตถุเองว่าต้องการให้มีลักษณะเป็นอย่างไรและสามารถทำอะไรได้บ้างแทนที่จะมองที่การสร้างรูทีน (Routine) หรือ (Procedure) เป็นหลักเช่นดังก่อนๆนี้ประโยชน์ที่ได้ก็คือ เราสามารถสร้างวัตถุโดยเริ่มจากวัตถุง่ายๆไม่ซับซ้อนเป็นพื้นฐานขึ้นมาเสียก่อน จากนั้นจึงนำวัตถุเหล่านั้นมาตกแต่งปรับปรุงให้มีความสามารถมากขึ้น หรือทำงานได้หลากหลายขึ้น โดยนำสิ่งที่เหมือนกันหรือใช้ร่วมกันมาไว้ในวัตถุ ซึ่งจะเรียกว่า Base Object หรือ Base Class จากนั้นจึงแต่งเติม Base Object นี้ให้กลายเป็นออบเจ็คอื่นตามต้องการ เราสามารถนำวัตุที่ได้นี้กลับมาใช้ใหม่ได้เรื่อยๆในแอพพลิเคชั่นต่างๆและแต่งเติมต่อไปได้เรื่อยๆเช่นกัน

2.8.2 สิ่งสำคัญในการเขียนโปรแกรมเชิงวัตถุ คือจะต้องมีความเข้าใจเกี่ยวกับโครงสร้างของออบเจ็คและหลักการของโอโอพี ดังนี้

2.8.2.1 โครงสร้างของออบเจ็ค ออบเจ็คทุกๆออบเจ็คจะต้องมีโครงสร้างดังต่อไปนี้

2.8.2.1.1 ชนิดของออบเจ็ค ออบเจ็คแต่ล่ะออบเจ็คจะถือว่าเป็นคนล่ะชนิด (Type) กัน เมื่อนำออบเจ็คไปสร้างต่อให้เป็นออบเจ็คใหม่ ก็จะต้องเป็นชนิดใหม่เสมอ หรือจะเรียกได้ว่าเป็นคนล่ะคลาส (Class) กันนั่นเอง ชนิดของออบเจ็คได้แก่ ออบเจ็คชนิดปุ่ม และออบเจ็คชนิดข้อความเป็นต้น

2.8.2.1.2 คุณสมบัติ เรียกอีกอย่างหนึ่งว่า “พร็อพเพอร์ตี้” (Property) หมายถึงคุณลักษณะของออบเจ็คแต่ล่ะตัวที่สามารถกำหนดให้แตกต่างกันไปตามความต้องการที่ต่างกันเช่น ขนาดและสีของปุ่ม หรือข้อความของออบเจ็คที่แสดงอยู่บนปุ่มเป็นต้น

2.8.2.2 หลักการของโอโอพี

คุณสมบัติของการโปรแกรมที่จะถือว่าเป็นเชิงวัตถุได้จะต้องมีคุณสมบัติในการซ่อนเร้นการทำงานนี้ไว้ภายใน สามารถนำคุณสมบัติที่มีไปดัดแปลงใช้งานต่อๆไปได้ และอาจจะแปลงเป็นออบเจ็คใหม่ได้ดังรายละเอียดต่อไปนี้

2.8.2.2.1 Encapsulation เป็นการซ่อนเร้นส่วนการทำงานภายในออบเจ็ค ที่ไม่เกี่ยวข้องกับภายนอกไว้ไม่ให้เห็นและไม่ให้แก้ไขเปลี่ยนแปลงส่วนที่ซ่อนไว้นี้ ซึ่งเราจะนำออบเจ็คไปใช้หรือไปดัดแปลงได้เฉพาะส่วนที่ออบเจ็คนั้นยอมให้เท่านั้น

2.8.2.2.2 Inheritance เป็นการสืบทอดคุณสมบัติของออบเจ็ค เมื่อนำออบเจ็คใดๆไปสร้างเป็นออบเจ็คใหม่ คุณสมบัติของออบเจ็คเดิมจะยังคงมีอยู่ และสามารถเรียกใช้และทำงานได้อย่างครบถ้วน

2.8.2.2.3 Polymorphism คือลักษณะการทำงานที่แตกต่างกันของคุณสมบัติหรือพฤติกรรมเดียวกันแต่เป็นของออบเจ็คคนล่ะชนิดกัน ตัวอย่างเช่น เมธอด Save To File ของเมโมจะได้ Text file ที่เก็บข้อความนั้น ในขณะที่ Save To File ของอิมเมจจะได้ไฟล์รูปภาพ ซึ่งแม้จะเป็นการบันทึกข้อมูลไว้เป็นไฟล์เหมือนกัน แต่วิธีการบันทึกก็ต่างกัน

2.8.3 คอมไพเลอร์ที่แท้จริง

Delphi เป็นคอมไพเลอร์ที่ใช้แปลภาษาโปรแกรมเป็นภาษาเครื่อง ซึ่งหลังจากคอมไพล์โปรแกรมเราจะได้ไฟล์นามสกุล exe ซึ่งเก็บภาษาของเครื่องนั้นๆ และสามารถทำงานได้เลยโดยไม่ต้องผ่านขั้นตอนการแปลภาษาในระหว่างการทำงานอีกทำให้สามารถทำงานได้เร็วและไม่มีข้อจำกัด

2.8.4 คอมโพเนนต์ไลบรารี (Component Library)

คอมโพเนนต์เป็นส่วนประกอบย่อยๆที่เราจะนำมาใช้ในการสร้างแอพพลิเคชั่น ซึ่งใน Delphi มีคอมโพเนนต์ให้เลือกใช้เป็นจำนวนมาก โดยจะเก็บอยู่ใน “ห้องสมุด” หรือ คอมโพเนนต์ไลบรารี (Component Library) และจัดแยกเป็นกลุ่มๆตามประเภทของการใช้งาน

2.8.5 OCX และ ActiveX

OCX เป็นออบเจ็คที่ใช้ใน Visual Basic ของบริษัทไมโครซอฟท์ ซึ่งมีลักษณะคล้ายกับคอมโพเนนต์ของ Delphi ส่วน ActiveX ก็เป็นออบเจ็คที่ถูกพัฒนามาในรูปแบบเดียวกับ OCX แต่มีวัตถุประสงค์เพื่อใช้งานบนอินเตอร์เน็ต ซึ่งทั้ง OCX และ ActiveX ต่างก็เป็นออบเจ็คที่นำมาใช้บน Delphi ได้

2.8.6 Wizard และ Object Repository

Delphi มี Wizard ที่ใช้ในการสร้างฟอร์มและออบเจ็คที่ต้องใช้บ่อยๆหรือสร้างได้ยาก ไว้ให้ใช้เป็นจำนวนมาก ตัวอย่างเช่น Wizard ในการสร้างรายงานแบบต่างๆหรือฟอร์มเช่น About Box และ Dialog Box ซึ่งจะเรียกอีกอย่างหนึ่งว่า เทมเพลต (Template) ก็ได้ เหล่านี้ล้วนช่วยในการสร้างแอพพลิเคชั่นเป็นไปอย่างรวดเร็วและมีความกลมกลืมมากขึ้น นอกจากนี้ยังสามารถเก็บฟอร์มหรือออบเจ็คที่คิดว่าต้องใช้บ่อยๆเพิ่มเข้าไปไว้ให้เรียกใช้ในแอพพลิเคชันอื่นๆได้อีกด้วย โดยการบันทึกไว้ใน Object Repository จากนั้นเมื่อต้องการใช้งานก็เรียกใช้ได้เหมือนกับการใช้ Wizard ทั่วๆไป

2.8.7 การติดต่อกับฐานข้อมูล

การติดต่อกับฐานข้อมูลนั้น Delphi มีคอมโพเนนต์ที่สามารถเชื่อมต่อ เพื่อจัดการกับข้อมูลที่เก็บอยู่ในฐานข้อมูลทุกประเภท ไม่ว่าจะเป็นการเพิ่ม ลบ แก้ไข หรือการเรียกดูข้อมูล โดยผู้ใช้ไม่ต้องเขียนชุดคำสั่งใดๆในโปรแกรมเลยก็สามารถสร้างแอพพลิเคชั่นอย่างง่ายๆที่ทำงานกับฐานข้อมูลขึ้นมาได้แล้ว

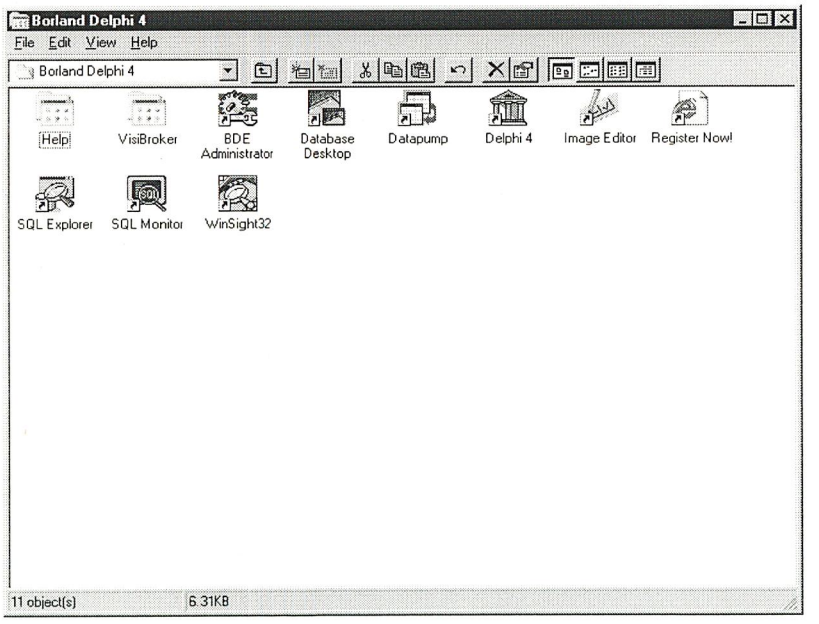
เราสามารถใช้ Delphi จัดการฐานข้อมูลที่เป็นแบบง่ายๆซึ่งได้แก่ dBase หรือ Paradox และ MS Access ซึ่งพวกนี้จะเรียกว่าเป็น Local Database คือ Database ที่ทำงานในเครื่องนั้นเอง หรือเรียกอีกอย่างหนึ่งว่าเป็นแบบ File_oriented Database ก็ได้ เพราะเป็นการเก็บข้อมูลไว้ในโครงสร้างของไฟล์ นอกจากนี้ยังสามารถใช้งานกับระบบฐานข้อมูลที่เป็น Database Server ต่างๆเช่น SQL Server หรือ InterBase ซึ่งอาจจะทำงานอยู่ในเครื่องเดียวกันกับแอปพลิเคชั่นในกรณีของ Local InterBase หรืออาจจะทำงานอยู่บนเครื่อง Server เครื่องอื่น ซึ่งเรียกว่าเป็นการใช้ฐานข้อมูลในแบบ Client/Server ก็ยังได้

2.8.8 แบบต่างๆของ Delphi

Delphi ตั้งแต่เวอร์ชั่นแรกเป็นต้นมา จะมีให้เลือกใช้ได้ 3 แบบคือ Standard, Professional และ Client/Server ซึ่งจะมีส่วนประกอบให้ใช้งานมากน้อยไม่เท่ากัน และมีราคาที่แตกต่างกันด้วย โดยแบบ Standard จะเหมาะสำหรับพัฒนาโปรแกรมทั่วๆไป ส่วนแบบ Professional จะสามารถจัดการฐานข้อมูลได้ และแบบ Client/Server จะใช้สำหรับองค์กรค์ขนาดใหญ่ๆที่ต้องจัดการกับฐานข้อมูลซึ่งมีข้อมูลจำนวนมากโดยการทำงานในแบบ Client/Server

2.8.9 โปรแกรมและเครื่องมือใน Delphi 4

หลังจากติดตั้ง Delphi 4 จะได้กรุ๊ปชื่อ Borland Delphi 4 อยู่ในเมนู Programs ของปุ่ม Start บนทาสก์บาร์ โดยมีตัวอย่างดังรูป



รูปที่ 2.27 แสดงกรุ๊ปของโปรแกรมเดลไฟ

ภายในกรุ๊ปของ Delphi ประกอบด้วยโปรแกรมและเครื่องมือหลักๆดังต่อไปนี้

2.8.9.1 Delphi 4 เป็นตัวโปรแกรม Delphi ที่ใช้ในการพัฒนาแอปพลิเคชั่น

2.8.9.2 BDE Administrator เป็นเครื่องมือที่ใช้สร้างและกำหนดคุณสมบัติต่างๆในการติดต่อกับฐานข้อมูลซึ่งการเชื่อมต่อระหว่างแอปปลิเคชั่นและฐานข้อมูลจะต้องอาศัย BDE (Borland Database Engine) ดังนั้นตัว BDE Administrator จึงเป็นตัวควบคุมการทำงานของ BDE เพื่อให้สามารถใช้งานกับฐานข้อมูลได้ตามที่ต้องการ

2.8.9.3 Database Desktop เป็นเครื่องมือที่ใช้ในการทำงานกับฐานข้อมูลเช่นการสร้างฐานข้อมูล การสร้างเทเบิล (Table) ในฐานข้อมูล ตลอดจนการทำงานกับข้อมูลในเทเบิล ไม่ว่าจะเป็นการเพิ่มข้อมูล แก้ไขข้อมูล ลบข้อมูลหรือเรียกดูข้อมูล

2.8.9.4 Datapump เป็นเครื่องมือที่ใช้ย้ายข้อมูลจากฐานข้อมูลหนึ่ง ไปยังอีกฐานข้อมูลหนึ่ง

2.8.9.5 Database Explorer เป็นเครื่องมือที่ใช้เรียกดูรายละเอียดในฐานข้อมูล ทั้งโครงสร้างเทเบิลและข้อมูลที่อยู่ในนั้น

2.8.9.6 SQL Monitor มีเฉพาะใน Delphi ชุด Client/Server เท่านั้น โดยเป็นเครื่องมือที่ใช้ในการติดตามดูการติดต่อกันระหว่าง BDE และ Database Server ว่ามีการส่งคำสั่ง SQL อะไรไปบ้าง เช่นเมื่อมีการส่งคำสั่งไปขอข้อมูลจากฐานข้อมูล ก็สามารถใช้ SQL Monitor ตรวจดูว่าในการทำงานนั้น DBE มีการติดต่อกับฐานข้อมูลอย่างไร

2.8.9.7 Image Editor เป็นเครื่องมือที่ใช้ในการตกแต่งภาพ

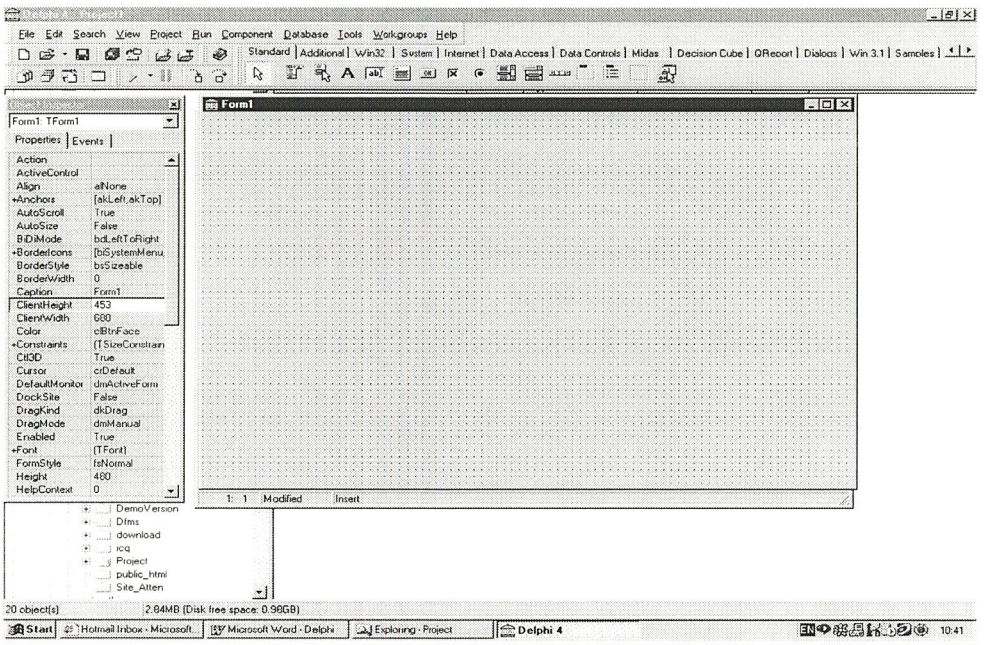
2.8.9.8 WinSight32 เป็นเครื่องมือที่ใช้ในการตรวจดูการทำงานของโปรแกรมที่เราเขียนขึ้นว่ามีการติดต่ออะไรกับ Windows บ้าง

2.8.9.9 Help เป็นส่วนที่แสดงรายละเอียดในการใช้งาน Delphi ทั้งหมด

2.8.10 สภาพแวดล้อมของ Delphi

การสร้างโปรแกรมใน Delphi จะมีรูปแบบที่เรียกว่า Integrated Development Environment (IDE) คือเป็นการสร้างโปรแกรมโดยการทำงานทุกอย่างสามารถทำได้จากในที่แห่งเดียวโดยไม่จำเป็นต้องใช้ Editor ตัวหนึ่งในการเขียนโปรแกรม และคอมไพล์ด้วยคำสั่งที่เป็น Command Line

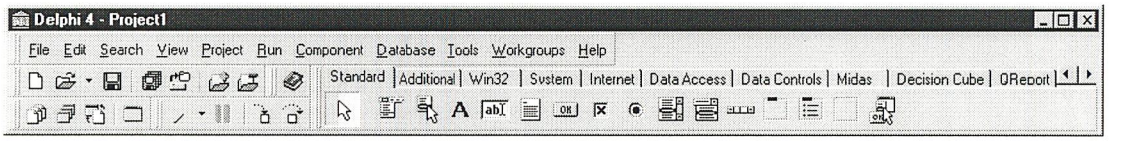
หลังจากที่รันโปรแกรม Delphi เราจะเห็นหน้าตาของ Delphi ซึ่งภายในประกอบด้วยวินโดว์ต่างๆ ดังนี้



รูปที่ 2.28 แสดงส่วนต่างๆของโปรแกรมบอแลนด์เดลไฟ

2.8.10.1 วินโดว์หลัก (Main Window)

วินโดว์หลักทำหน้าที่เป็นตัวกลางสำหรับการควบคุมกระบวนการในการพัฒนาโปรแกรม ไม่ว่าจะเป็นการจัดการเกี่ยวกับไฟล์ภายในแอปพลิเคชั่น การคอมไพล์ และการตรวจสอบข้อผิดพลาด เมื่อรันโปรแกรม Delphi จะเห็นวินโดว์ที่อยู่บนสุด โดยภายในวินโดว์หลักจะแบ่งออกเป็น 4 ส่วนคือ ไตเติลบาร์ เมนูบาร์ สปีดบาร์ และคอมโพเนนต์พาเล็ต ดังนี้



รูปที่ 2.29 แสดงวินส์โดว์หลัก

2.8.10.2 ไตเติลบาร์ (Title Bar)

เป็นส่วนที่อยู่บนสุด แสดงชื่อโปรแกรม Delphi และชื่อโปรเจ็คที่ทำงานอยู่ในขณะนั้น

2.8.10.3 เมนูบาร์ (Menu Bar)

เป็นส่วนประกอบที่จะพบได้ในแอปพลิเคชั่นบน Windows ทั่วไป โดยจะอยู่ใต้ไตเติลบาร์ ภายในเมนูบาร์จะเป็นส่วนที่แสดงเมนูคำสั่งของ Delphi ซึ่งใช้ในการทำงานต่างๆเกือบทั้งหมด เช่นเมนู File ใช้ทำงานกับแฟ้มข้อมูล ไม่ว่าจะเป็นการเปิด การปิด หรือการบันทึกแฟ้มข้อมูล เป็นต้น

2.8.10.4 สปีดบาร์ (Speed Bar)

คือกลุ่มของคำสั่งที่ใช้บ่อยซึ่งจะอยู่ใต้เมนูบาร์ทางด้านซ้ายของวินโดว์หลัก ภายในสปีดบาร์ประกอบด้วยปุ่มแทนรายการต่างๆของเมนู เมื่อคลิกเมาร์ที่ปุ่มใดก็จะเหมือนกับการเลือกคำสั่งจากเมนู

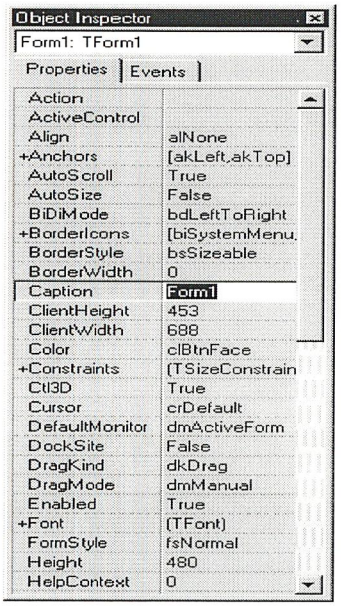
และเมื่อเลื่อนเมาส์ไปหยุดบนปุ่มใดก็จะเห็นกรอบข้อความ (tooltips) แสดงชื่อเมนูและฟังก์ชั่นคีย์ซึ่งเป็นคีย์ลัดสำหรับใช้แทนปุ่มนั้น

2.8.10.5 คอมโพเนนต์พาเล็ต (Component Palette)

คอมโพเนนต์พาเล็ตคือส่วนที่อยู่ถัดจากสปีดบาร์ไปทางด้านขวา ประกอบด้วยคอมโพเนนต์สำหรับสร้างแอปพลิเคชั่น โดยจัดกลุ่มไว้ในแท็ปต่างๆ คลิกที่แท็ปเพื่อแสดงคอมโพเนนต์ในแต่ละกลุ่ม และเมื่อเลื่อนเมาส์ไปหยุดบนคอมโพเนนต์ใดก็จะปรากฏกรอบข้อความแสดงชื่อคอมโพเนนต์นั้นขึ้น

2.8.10.6 วินโดว์ออบเจ็คอินสเป็คเตอร์ (Object Inspector Window)

วินโดว์ออบเจ็คอินสเป็คเตอร์เป็นวินโดว์ซึ่งอยู่ใต้วินโดว์หลักทางด้านซ้าย ใช้สำหรับแสดงและปรับแต่งค่าพร็อพเพอร์ตี้ (property) และอีเวนต์ (evevt) ของคอมโพเนนต์ จารูปจะเห็นได้ว่าใต้ไตเติลบาร์ของวินโดว์ออบเจ็คอินสเป็คเตอร์ จะเป็นคอมโบบ็อกซ์สำหรับแสดงชื่อและชนิดของคอมโพเนนต์ที่ทำงานอยู่ในขณะนั้น ในรูปเป็น "Form1:Tform1" แสดงว่าเรากำลังทำงานกับคอมโพเนนต์ชื่อ From1 และมีชนิดของคอมโพเนนต์เป็น Tform1 ถัดลงมาจะมีแท็บ 2 แท็บคือ แท็บ Properties และแท็บ Events



รูปที่ 2.30 แสดงวินโดว์ออบเจ็คอินสเป็คเตอร์

2.8.10.7 แท็บ Properties

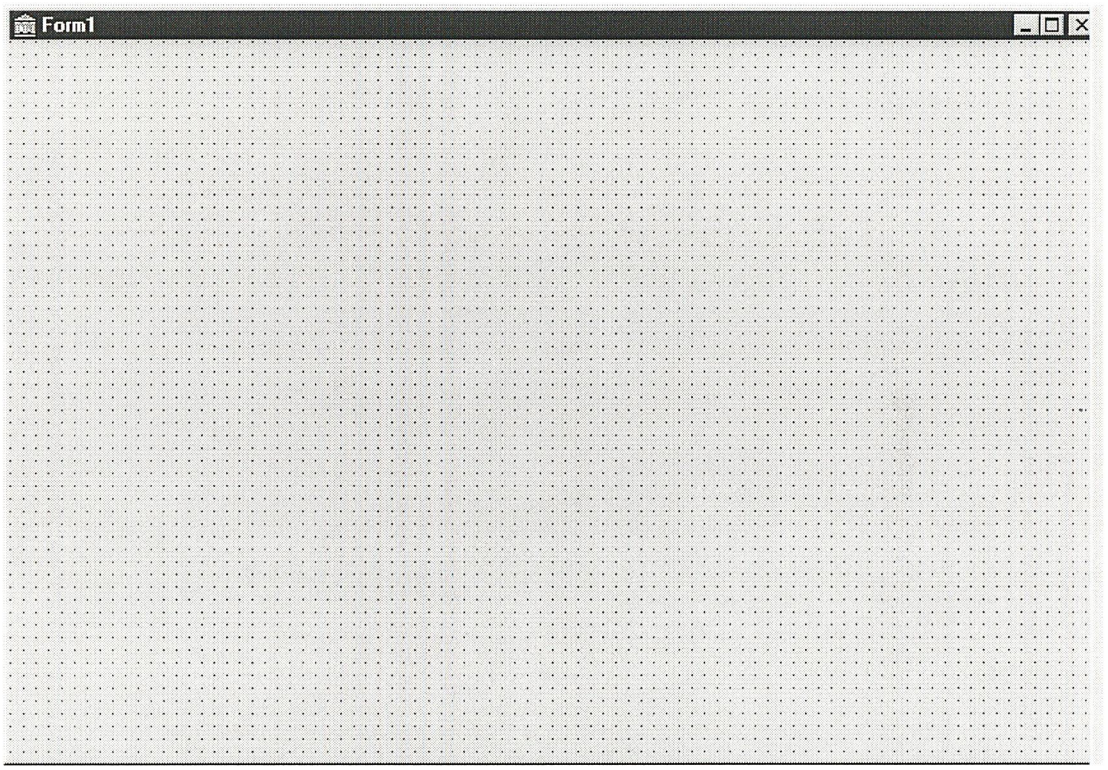
ใช้สำหรับแสดงและกำหนดค่าคุณสมบัติหรือพร็อพเพอร์ตี้ (property) ของคอมโพเนนต์ ภายในแท็บแบ่งออกเป็น 2 คอลัมน์ ดดยคอลัมน์ทางซ้ายแสดงชื่อพร็อพเพอร์ตี้ ส่วนคอลัมน์ทางขวามีไว้สำหรับแสดงและแก้ไขค่าของพร็อพเพอร์ตี้

2.8.10.8 แท็บ Events

ใช้สำหรับกำหนดการทำงานเมื่อมีเหตุการณ์หรือการกระทำใดๆเกิดขึ้นกับคอมโพเนนต์ ซึ่งอาจเกิดจากการกระทำของผู้ใช้หรือตัวโปรแกรมเอง ตัวอย่างเช่น การใช้เมาส์คลิกที่ปุ่มจะเกิด อีเวนต์ OnClick ขึ้นกับปุ่มนั้น หรือการกด Enter บน EditBox ก็จะเกิดอีเวนต์ OnKeyDown, OnKeyUp และ ONKeyPress ขึ้นมา

2.8.10.9 วินโดว์ฟอร์ม (Form Window)

ฟอร์มเป็นวินโดว์ที่ใช้สำหรับออกแบบส่วนที่ติดต่อกับผู้ใช้ โดยการนำคอมโพเนนต์ต่างๆจากคอมโพเนนต์พาเล็ตมาวางลงบนฟอร์ม

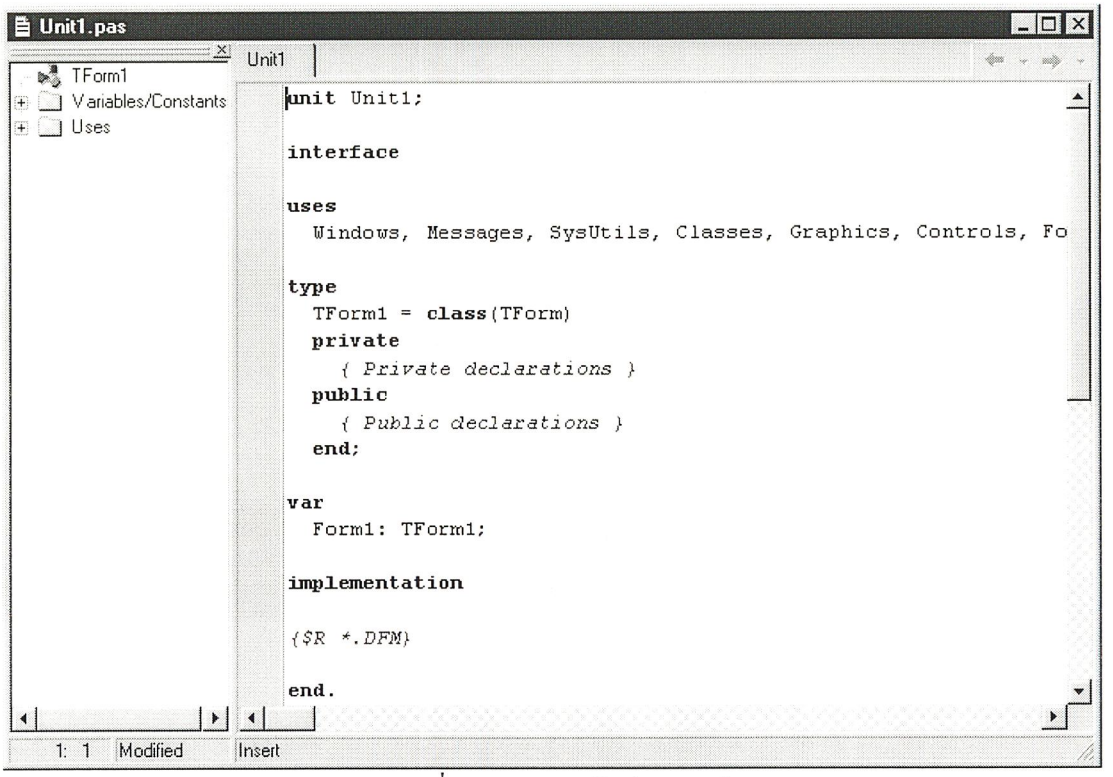


รูปที่ 2.31 แสดงวินโดว์ฟอร์ม

เมื่อเปิดโปรแกรม Delphi ขึ้นมาหรือสร้างแอปพลิเคชั่นใหม่ Delphi จะสร้างฟอร์มให้หนึ่งฟอร์มเสมอ ในแอปพลิเคชั่นหนึ่งๆจะประกอบด้วยฟอร์มอย่างน้อยหนึ่งฟอร์มเสมอ

2.8.10.10 วินโดว์โค้ดเอดิเตอร์ (Code Editor Window)

วินโดว์โค้ดเอดิเตอร์มีไว้สำหรับเขียนโปรแกรม โดยอาจจะถูกบังอยู่ใต้ฟอร์ม เมื่อเลื่อนวินโดว์ฟอร์มออกไปหรือคลิกที่ปุ่ม minimize ของฟอร์ม จะเห็นวินโดว์โค้ดเอดิเตอร์ปรากฏขึ้นดังรูป



รูปที่ 2.32 แสดงวินโดว์เอดิเตอร์

ภายในวินโดว์โค้ดเอดิเตอร์ จะประกอบด้วยส่วนต่างๆดังต่อไปนี้

2.8.10.10.1 Unit Explorer เป็นกรอบที่อยู่ทางด้านซ้าย ซึ่งใช้สำหรับแสดงสิ่งต่างๆที่มีอยู่ในยูนิตนั้น เช่น ออบเจ็ค ตัวแปร ค่าคงที่ และชื่อของยูนิตต่างๆที่ถูกอ้างอิงถึงในยูนิต ลักษณะการแสดงรายละเอียดจะอยู่ในรูปแบบของโครงสร้างต้นไม้ (Tree View)

2.8.10.10.2 Code editor โค้ดเอดิเตอร์เป็นส่วนที่ใช้เขียนคำสั่ง โดยมีแท็บของชื่อไฟล์ที่ใช้เก็บยูนิตนั้นสำหรับเลือกยูนิตที่ต้องการ

2.8.10.11 คอนโทรลต่างๆที่ใช้ในการทำงานกับฐานข้อมูล ใน Delphi 4 สามารถทำการเชื่อมต่อกับฐานข้อมูลได้โดยใช้คอมโพเนนต์หลักดังนี้คือ

2.8.10.11.1 คอมโพเนนต์เทเบิล (Table) เป็นดาต้าเซ็ทคอมโพเนนต์สำหรับเข้าถึงข้อมูลในเทเบิลที่เก็บข้อมูลครั้งละหนึ่งเทเบิลโดยการใช้ Borland Database Engine (BDE) ในการเข้าถึงข้อมูลทุกเรคคอร์ดและฟิลด์ในเทเบิล โดยสามารถใช้งานกับฐานข้อมูลแบบ Paradox, dBASE, Access, FoxPro และฐานข้อมูลที่สามารถเรียกผ่าน ODBC ได้ หรือเชื่อมต่อกับฐานข้อมูล SQL ที่อยู่บนเซอร์เวอร์ที่อยู่ห่างไกลออกไปเช่น InterBase, Oracle, Sybase, MS-SQL Server, Informix หรือ DB2 คอมโพเนนต์เทเบิลสามารถทำงานเป็นซับเซตของเรคคอร์ดในเทเบิลฐานข้อมูลโดยการใช้ช่วงข้อมูลและฟิลเตอร์ได้อีกด้วย ในขณะที่ออกแบบเราสามารถสร้าง (create), ลบ (delete), ปรับปรุงข้อมูล (update) หรือเปลี่ยนชื่อเทเบิลฐาน

ข้อมูลที่กำลังเชื่อมต่อโดยเทเบิลอยู่ได้โดยการคลีกขวาบนคอมโพเนนต์เทเบิลเพื่อเรียกใช้งานป็อบอัพเมนูต่อไป

2.8.10.11.2 คอมโพเนนต์คิวรี่ (Query) เป็นดาต้าเซ็ทคอมโพเนนต์สำหรับเข้าถึงข้อมูลทีละหนึ่งหรือหลายๆเทเบิลในฐานข้อมูลโดยการใช้ชุดคำสั้ง SQL คอมโพเนนต์คิวรี่สามารถใช้งานกับเซอร์เวอร์ระยะไกลต่างเช่น Sybase, SQL Server, Oracle, Informix, DB2, และ InterBase หรือกับเทเบิลที่อยู่ในเครื่องเดียวกันเช่น Paradox, dBASE, Access, and FoxPro และฐานข้อมูลที่ใช้งานผ่าน ODBC ได้ คอมโพเนนต์คิวรี่เป็นคอมโพเนนต์ที่สามารถใช้งานได้อย่างมีประโยชน์มากเพราะสามารถเข้าถึงข้อมูลได้มากกว่าหนึ่งเทเบิลในครั้งหนึ่งโดยใช้คำสั่ง "join" ใน คำสั่ง SQL ได้ อีกทั้งยังสามารถเข้าถึงข้อมูลในแถวและคอลัมย์ของทุกเทเบิลที่เชื่อมต่ออยู่ได้โดยอัติโนมัติอีกด้วย

2.8.10.11.3 คอมโพเนนต์ดาต้าซอร์ส (Datasource) เป็นคอมโพเนนต์สำหรับเชื่อมต่อระหว่างคอมโพเนนต์ดาต้าเซ็ตต่างๆเช่นเทเบิล หรือคิวรี่ กับคอมโพเนนต์จำพวกดาต้าคอลโทรลต่างๆบนฟอร์มเพื่อแสดงข้อมูล,ค้นหาข้อมูลและแก้ไขข้อมูลจากฐานข้อมูลที่ถูกเชื่อมต่อโดยคอมโพเนนต์ดาต้าเซ็ต

2.8.10.11.4 คอมโพเนนต์ดาต้าคอลโทรล เป็นคอมโพเนนต์สำหรับดึงข้อมูลจากฐานข้อมูลมากระทำกรรมวิธีต่างๆเช่น แสดงผลโดยการใช้คอมโพเนนต์ดีบีเท็กบ็อก (DBTextBox) หรือ ดีบีกริด (DBGrid) เป็นต้น

2.8.11 BDE (Borland Database Engine)

BDE เป็นเครื่องมือที่ถูกออกแบบมาเพื่อช่วยในการพัฒนาแอปพลิเคชั่นบนวินส์ดดว์สำหรับภาษา C,C++ และ Delphi สามารถใช้กับฐานข้อมูลได้หลายประเภทเช่น dBase,Paradox, Microsoft Access,Oracle,Sybase,Interbase และ Informix เป็นต้น

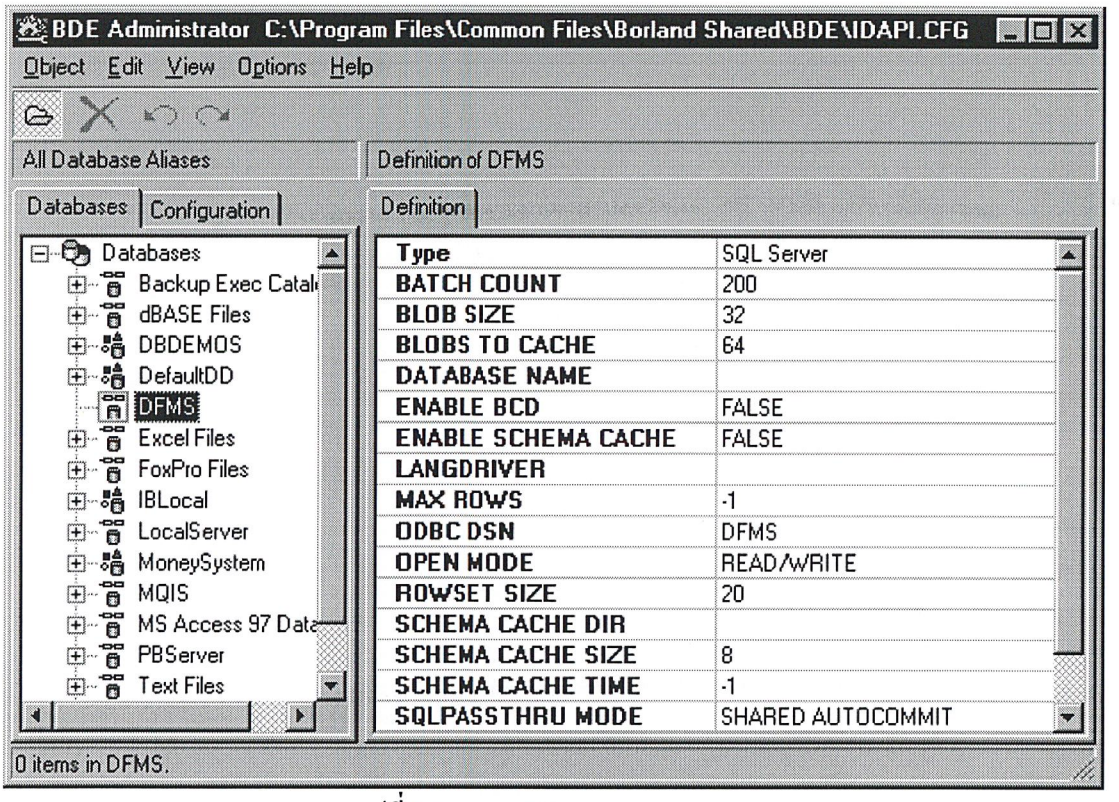
2.8.11.1 ประโยชน์ของ BDE

- ใช้ API เดียวกันในการเข้าถึงฐานข้อมูลทุกประเภท เช่น dBase,Paradox, Microsoft Access,Oracle,Sybase,Interbase และ Informix เป็นต้น ทำให้วิธีการพัฒนาโปรแกรมไม่ขึ้นอยู่กับชนิดของฐานข้อมูล
- สามารถใช้กับการพัฒนาแอปพลิเคชั่นแบบ Client/Server โดยเข้าถึงข้อมูลได้ทั้ง local database และ ที่อยู่บนเซอร์เวอร์
- สามารถติดต่อและเข้าถึงแหล่งข้อมูลประเภทต่างๆได้โดยไม่จำเป็นต้องทำการอิมพอร์ต
- ทุกแอปพลิเคชั่นสามารถใช้ BDE ร่วมกันได้ ซึ่งในกรณีที่มีแอปพลิเคชั่นหลายตัวเชื่อมต่อกับฐานข้อมูลหลายประเภทก็ใช้ BDE ตัวเดียวร่วมกันได้

2.8.11.2 การใช้งาน BDE

เนื่องจาก BDE เป็นเครื่องมือที่ใช้สำหรับเชื่อมต่อกับฐานข้อมูลทุกประเภท เราจึงสามารถกำหนด driver และคุณสมบัติต่างๆของแต่ละฐานข้อมูลที่จะใช้งาน เช่นประเภทของฐานข้อมูล ไดเร็คทอรี่ หรือ

เซิร์ฟเวอร์ที่ใช้เก็บฐานข้อมูล เป็นต้น คุณสมบัติเหล่านี้สามารถกำหนดโดยใช้โปรแกรม BDE Administrator ซึ่งเรียกใช้ได้จากเมนูในกลุ่มเดียวกันกับ Delphi โดยภายในวินโดว์ของ BDE Administrator แบ่งเป็น 2 ส่วนคือ



รูปที่ 2.33 แสดง BDE Administrator

- กรอบด้านซ้ายแสดงข้อมูลแบบ Tree View ซึ่งมี 2 แท็บดังนี้

  databases สำหรับกำหนดเอเลียส(Alias) คือชื่อที่ใช้อ้างอิงไปยังฐานข้อมูล ซึ่งแอปพลิเคชั่นที่สร้างขึ้นจะใช้ชื่อนี้ในการอ้างอิงถึงข้อมูล

  Configuration สำหรับกำหนดคุณสมบัติของ driver ที่ใช้ในการติดต่อกับฐานข้อมูลและตัว BDE เอง โดยปกติ Delphi จะมี driver มาให้เราใช้งานได้จำนวนหนึ่ง ซึ่งเลือกติดตั้งได้ขณะที่ติดตั้ง Delphi และจะแสดงเป็นรายการอยู่ในนี้

- กรอบด้านขวามีเพียงแท็บเดียวคือ Definition สำหรับแสดงรายละเอียดของแต่ละรายการที่เลือกจากกรอบด้านซ้าย

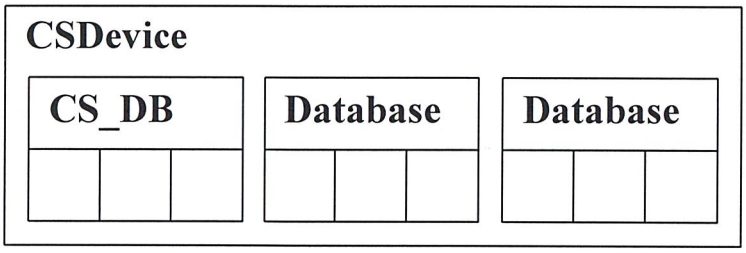
2.9 ความรู้เกี่ยวกับเอสคิวแอลเซิร์ฟเวอร์ (SQL Server)

เอสคิวแอลเซิร์ฟเวอร์ดาต้าเบส (SQL Server Database) ทำการเข้าถึงข้อมูลโดยผ่านโอดีบีซีไดร์ฟเวอร์ เนื่องจากนิยมใช้โอดีบีซีกันอย่างแพร่หลาย โครงสร้างของเอสคิวแอลเซิร์ฟเวอร์ มีลักษณะแตกต่างจากมัลติยูสเซอร์เดสทอปดาต้าเบส (MultiUser Desktop Database) ตัวอื่นๆ เนื่องจากภายในไม่มีเครื่องมือ

ในการสร้างฟรอนท์เอนด์ (Front End) มากนัก และโครงสร้างของการเก็บข้อมูลต่างกัน โดยเอสคิวแอลเซิร์ฟเวอร์ใช้โครงสร้างการเก็บข้อมูลได้หลายแบบ ทำให้สามารถรองรับความต้องการของผู้ใช้ และเป็นที่นิยมใช้กันอย่างแพร่หลาย

2.9.1 โครงสร้างข้อมูลในเอสคิวแอลเซิร์ฟเวอร์

เอสคิวแอลเซิร์ฟเวอร์ใช้วิธีกำหนดพื้นที่ดิสค์ (Allocate disk) ตามแบบฉบับของตนเองเพื่อให้มีประสิทธิภาพสูงสุด



รูปที่ 2.34 แสดงโครงสร้างข้อมูลของเอสคิวแอลเซิร์ฟเวอร์

ดีไวซ์ (Device) เป็นส่วนของการเก็บข้อมูลที่ต่อเนื่องกันบนฮาร์ดิสค์ของเซิร์ฟเวอร์ ใช้สำหรับเก็บดาต้าเบส (Database) และทรานสแอคชั่นล็อก (Transaction Log) ซึ่งดาต้าเบสจะไม่สามารถสร้างขึ้นได้จนกว่าจะมีการสร้างและกำหนดดีไวซ์ก่อนเพื่อสำหรับเก็บดาต้าเบส ดาต้าเบสสามารถเก็บแยกให้อยู่ในดีไวซ์ที่ต่างกันได้

การกำหนดขนาดของดีไวซ์เป็นหน้าที่ของแอดมินิสเทรเตอร์ (Administrator) การทำให้ระบบมีประสิทธิภาพสูงสุด สามารถทำได้โดยการเพิ่มหรือลดขนาดของดาต้าเบสที่เก็บอยู่ในไฟล์พาร์ติชัน (File partition) โดยไม่อ้างอิงกับระบบไฟล์ของโอเอส (OS) เมื่อมีการสร้างดีไวซ์จะมีการระบุชนิดของดีไวซ์และมีการทำดาต้าเบสแบคอัพ (Database Backup)

ในฐานข้อมูลการทำการเก็บตารางต่างๆ โดยใช้เอสคิวแอลออบเจคต์เมเนเจอร์ (SQL Object Manager) หรืออาจใช้จากคำสั่งเอสคิวแอลโดยตรง ฐานข้อมูลจะถูกทำการกำหนดลงในดีไวซ์ ซึ่งใน 1 ดีไวซ์สามารถเก็บได้หลายตาราง การเชื่อมต่อโดยใช้โอดีบีซีจะกำหนดให้ทำการแอคเซสฐานข้อมูลได้ครั้งละ 1 ฐานข้อมูลเท่านั้น ดังนั้นตารางที่มีความสัมพันธ์กันควรที่จะเก็บไว้ในฐานข้อมูลเดียวกัน

การสรสร้างทรานแอคชั่นล็อกจะทำการสร้างในดีไวซ์เดียวกันกับฐานข้อมูล ทำหน้าที่เป็นบัฟเฟอร์ (Buffer) เก็บทรานสแอคชั่นต่างๆ ที่ทำงานกับฐานข้อมูล เพื่อประโยชน์ในการแก้ไขภายหลังโดยมีหลักการคือ ก่อนที่จะมีการเปลี่ยนแปลงข้อมูล จะต้องทำการบันทึกทรานสแอคชั่นล็อกก่อน และเปลี่ยนแปลงแก้ไขข้อมูลได้ก็ต่อเมื่อเก็บทรานสแอคชั่นเรียบร้อย

2.9.2 ซิสเต็ม – แอดมินิสเตเตอร์ : เอสเอ (System Administrator : SA) และดาต้าเบสโอนเนอร์ : ดีบีโอ (Database Owner : DBO)

ทั้งเอสเอและดีบีโอเป็นบุคคลที่มีอำนาจในการเปลี่ยนแปลงโครงสร้างของฐานข้อมูลได้มากกว่าผู้ใช้อื่นๆ ทั่วไป โดยเอสเอทำหน้าที่สร้างฐานข้อมูลใหม่กำหนดและให้สิทธิต่างๆ และควบคุมการทำงานในระบบทั้งหมดส่วนดีบีโอหมายถึงผู้เป็นเจ้าของฐานข้อมูลมีสิทธิ์ในการให้สิทธิ์เข้าถึงข้อมูลแก่ผู้ใช้อื่น ๆ

2.9.3 ซิสเต็มเทเบิ้ล (System Table)

เอสคิวแอลเซิร์ฟเวอร์ทำการสร้างตาราง เพื่อจัดระเบียบของผู้ใช้ที่ทำการสร้างตารางและคอยควบคุมจัดการ ซึ่งเรียกว่าซิสเต็มคาต้าล็อก (System Catalog) เก็บอยู่ในมาสเตอร์ – ดาต้าเบส (Master Database) มีทั้งหมด 13 ตาราง

SysChaSet	ชุดตัวอักษรและการจัดลำดับของอักษรที่ใช้ใน SQL Server
SysConfingures	ค่าของ User Parameter ซึ่งจะส่งผลเมื่อกำหนดคำสั่ง Reconfigure
SysCurconfig	ค่าของ User Parameter ขณะ Session หรือ Query จนกระทั่งปัจจุบัน
SysDatabase	ข้อมูลของ Database ต่างๆ บนเซิร์ฟเวอร์
SysDevice	ชุดของดีไวซ์ต่างๆ ในระบบ
SysLanguage	เรคอร์ดของภาษาต่างๆ ที่ SQL Server สามารถเข้าใจได้ ซึ่งค่า Default คือ US English
SysMessage	เรคอร์ดที่ทำการเก็บ Error Message ที่ Return จาก SQL Server
SysLock	User Account และ Password ทั้งหมด
SysProcess	ข้อมูลเกี่ยวกับ Server Process
SysMotelLogins	เรคอร์ดของ Remote User ที่ทำการเรียก Stored Procedure
SysServed	เรคอร์ดสำหรับแต่ละ Remote SQL Server สามารถทำการร้องขอและเรียกใช้ Stored Procedure ได้
SysUsage	ข้อมูลเกี่ยวกับ Disk Allocation

ตารางที่ 2.1 ตารางแสดงการเก็บข้อมูลของซิสเต็มเทเบิ้ลต่างๆ

นอกจากนี้ในแต่ละฐานข้อมูลยังเก็ฐตาราง อีก 13 ตารางเรียกว่า ดาต้าเบสคาตาล็อก (Database Catalog) หรือดาต้าดิคชันนารี (Data Dictionary)

SysAlternative	ใช้สำหรับ Alternate User ที่ไม่ได้เก็บใน SysUser Table
SysColumn	เรคอร์ดสำหรับคอลัมน์ต่างๆ ในทุกๆ Table และ View
SysConnect	Comment ต่างๆ ของ View, Rules, Trigger, Default และ Stored Procedure
SysDepend	Cross Reference Data
SysIndex	เรคอร์ดสำหรับ Cluster Index และ UnCluster Index
SysKey	เรคอร์ดสำหรับ Key ต่างๆ ในฐานข้อมูลทั้ง Primary key, Foreign key และ Common key
SysLog	Transaction Log ของ Database
SysObject	เรคอร์ดสำหรับเก็บ Database Object แต่ละชนิด
SysProcedure	เรคอร์ดสำหรับแต่ละ View, Default, Rules, Trigger และ Stored Procedure
SysProtect	User Permmission Information รวมถึง Grant และ Revoke
SysSegment	เรคอร์ดของแต่ละ Segment
SysType	ตารางสำหรับ Data type ต่างๆ
SysUser	เรคอร์ดสำหรับ User ที่อนุญาตให้ Access Database

ตารางที่ 2.2 ตารางแสดงการเก็บข้อมูลของดาต้าดิคชั่นนารี

2.9.4 ชนิดข้อมูล (Data Type)

เอสคิวแอลเซิร์ฟเวอร์ให้ใช้งานข้อมูลที่ตรงกันกับชนิดข้อมูลที่ในเอสคิวแอลเซิร์ฟเวอร์ และชนิดข้อมูลที่ผู้ใช้กำหนดขึ้นเอง การเลือกใช้ชนิดข้อมูลให้ตรงกับฟิลด์หรือคอลัมน์เป็นสิ่งสำคัญ

ชื่อ (Name)	**ชนิดข้อมูล (Data Type)**	**ขนาด (Size)**
Char (n)	Character	N (nondynamic)
Varchar (n)	Character	Size of data, Maximum Specified by n

ตารางที่ 2.3 ตารางแสดงชนิดข้อมูลในเอสคิวแอลเซิร์ฟเวอร์

ชื่อ (Name)	ชนิดข้อมูล (Data Type)	ขนาด (Size)
Int	Integer	4 byte
Smallint	Integer	2 byte
Tinyint	Integer	1 byte
Float	Floating – point	8 byte
Real	Floating – point	4 byte
Binary (n)	Binary	N (nondynamic)
Varbinary (n)	Binary	Size of data, Maximum specified by n
Bit	Other	1 bit
Money	Money	8 byte
Smallmoney	Money	4 byte
Datetime	Date and time	8 byte .. 4 bytes for date and 4 bytes for time
Smalldatetime	Date and time	4 bytes .. 2 bytes for date and 2 bytes for time
Text	Character	
Image	Other	
Timestamp	Other	
Sysname	Other	Not null

ตารางที่ 2.3 ตารางแสดงชนิดข้อมูลในเอสคิวแอลเซิร์ฟเวอร์ (ต่อ)

2.9.5 วิว (View)

ชนิดของตารางแบบเสมือน คือไม่มีตารางนั้นอยู่จริงในฐานข้อมูล แต่สามารถทำงานแบบตารางได้ เช่น คิวรี (Query) อัพเดท (Update) โดยวิวในเอสคิวแอลจะมีลักษณะเหมือนกับเรคอร์ดเซตในภาษาเดลไฟ ซึ่งมีพื้นฐานจากตารางในฐานข้อมูล วิวมีประโยชน์ในการสร้างคิวรีมาก เพราะในแอพพลิเคชันจะได้ไม่จำเป็นที่จะต้องทำการคิวรี หรือจอยน์ (Join) ทุกครั้งที่มีการเรียกใช้ เพราะได้ทำไว้ก่อนแล้ว

2.9.6 เอสคิวแอลอินเด็กซ์ (SQL Index)

ในเอสคิวแอลเซิร์ฟเวอร์ยอมให้มีการสร้างอินเด็กซ์ในฟิลด์ข้อมูลโดยใช้โครงสร้างแบบบี-ทรี (B – Tree) ในการเก็บอินเด็กซ์ และใน 1 ตารางสามารถมีได้มากกว่า 1 อินเด็กซ์ และในแต่ละอินเด็กซ์อาจเป็นแบบมัลติเทียร์ (Multitier) เอสคิวแอลเซิร์ฟเวอร์จะใช้อินเด็กซ์โดยอัตโนมัติเมื่อทำการคิวรี และเพื่อเป็น

การหลีกเลี่ยงปัญหาการสร้างอินเด็กซ์มากเกินไป ซึ่งจะทำให้ประสิทธิภาพของฐานข้อมูลลดลงได้ ดังนั้น ทุกครั้งที่มีการอินเสิร์ท (Inser) หรือ อัพเดท (Update) เรคอร์ดก็ต้องมีการอัพเดทอินเด็กซ์ด้วย

2.9.7 ทรานสแอคชั่น – เอสคิวแอล (Transaction SQL) สตอร์โพรซีเดอร์ (Store Procedure) และทริกเกอร์ (Trigger)

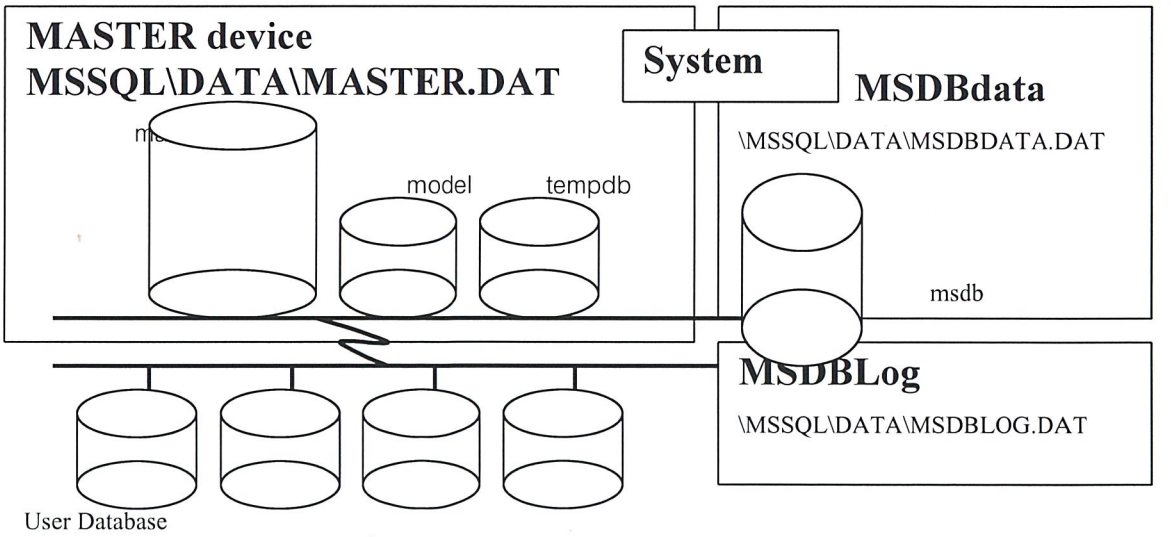
เอสคิวแอลเซิร์ฟเวอร์ยอมให้มีการเก็บโค้ดในส่วนเซิร์ฟเวอร์ได้ ซึ่งโค้ดเหล่านี้เขียนอยู่ในรูปทรานสแอคชั่น – เอสคิวแอล และเก็บอยู่ในรูปแบบที่ยังไม่ทำการคอมไพล์ในดาต้าดิคชันนารีของเซิร์ฟเวอร์

ทริกเกอร์เป็นสตอร์โพรซีเดอร์ที่ประมวลผลโดยอัตโนมัติเมื่อตรงกับเงื่อนไขที่กำหนด โดยทำการตรวจสอบเงื่อนไขเหล่านั้นทุกครั้งที่ทำการอินเสิรท์ อัพเดท และดีลีท เรคคอร์ด

2.9.8 สถาปัตยกรรมระบบฐานข้อมูลของ SQL Server

ระบบฐานข้อมูล Microsoft SQL Server จะมีสถาปตยกรรมฐานข้อมูลอยู่สองแบบ คือ System Database (ประกอบด้วย master, model, tempdb, msdb) และ User - Database SQL Server จะใช้ System Database ในการปฏิบัติและการจัดระบบ ผู้บริหารสามารถจะจัดการกับ System Database และเรียกดูข้อมูลใน System Table ซึ่งจะเก็บรายละเอียดฐานข้อมูล ออบเจ็กต์ต่างๆ อินฟอร์เมชั่นของผู้ใช้ ฯลฯ

Microsoft SQL Server จะติดตั้งระบบฐานข้อมูล 4 ระบบ คือ master, model, tempdb และ msdb ระบบ Stored Procedure จะใช้คิวรีข่าวสารเกี่ยวกับข้อมูลที่อยู่ในตาราง ซึ่งตารางจะเก็บบันทึกข่าวสารอินฟอร์เมชั่นเกี่ยวกับ SQL Server ทั้งหมดและข้อมูลของ User



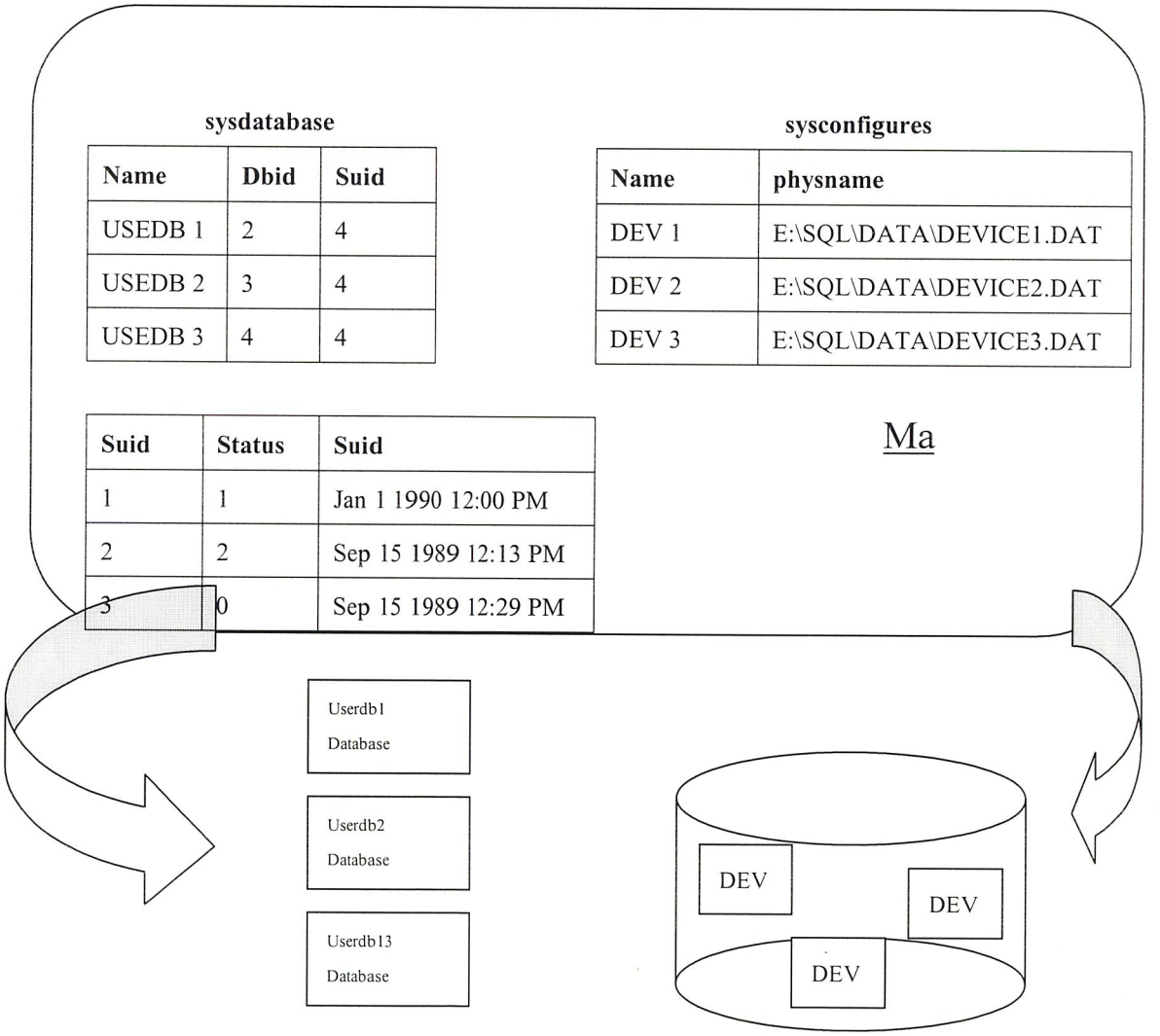
รูปที่ 2.35 รูปแสดงโครงสร้างฐานข้อมูลของ SQL Server

2.9.9 ฐานข้อมูลของระบบ SQL Server

จากรูปจะเห็นว่ามีดีไวซ์ของฐานข้อมูลอยู่ 3 ตัว คือ MASTER, MSDBDATA และ MSDBLOG โดยที่ใน MASTER devices จะมีฐานข้อมูล master , tempdb และ pubs อยู่ภายใน ส่วนฐานข้อมูล msdb จะอยู่ใน MSDBDATA devices และ transaction log ของ msdb จะเก็บอยู่ใน MSDBLOG

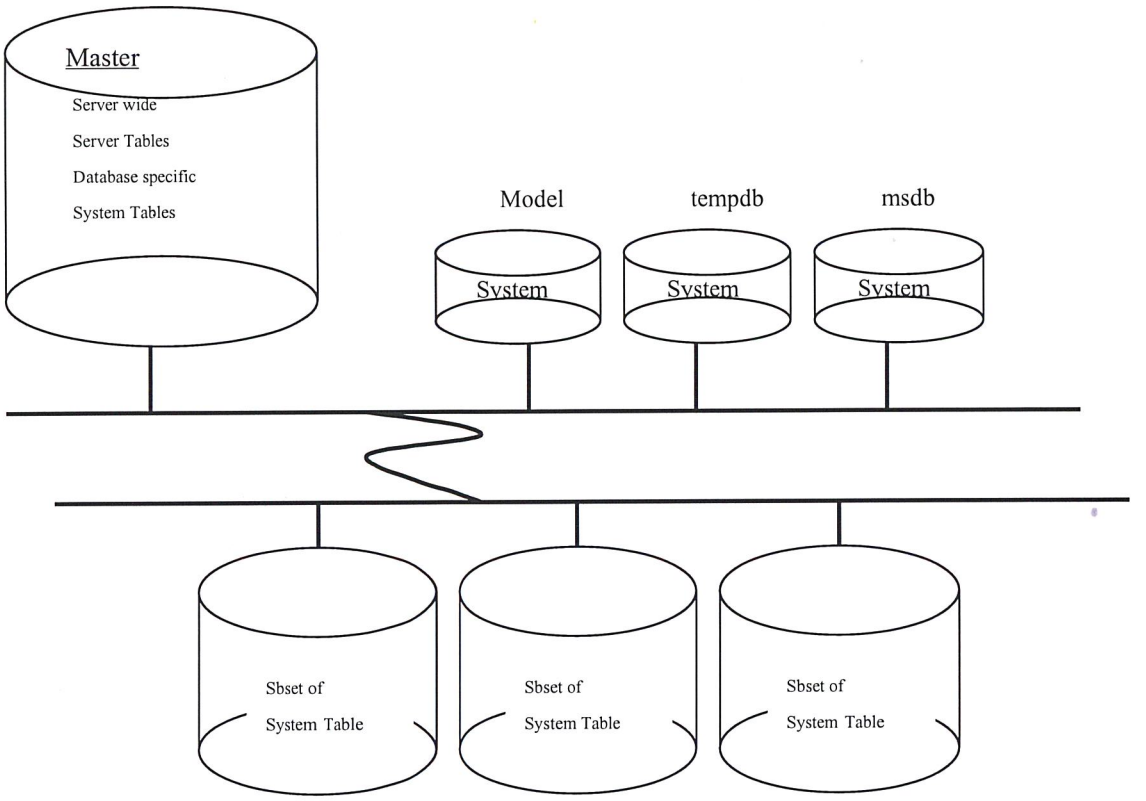
2.9.10.1 Master Database

Master Database จะบรรจุตาราง (System Tables) ซึ่งเก็บอินฟอร์เมชั่น เกี่ยวกับ SQL Server ทั้งหมดและข้อมูลเกี่ยวกับ User ตารางประกอบด้วยไดเรกทอรีข้อมูล หรือระบบแคททาล็อกของ SQL Server Master Database จะควบคุมฐานข้อมูลของ User และปฏิบัติการกับ SQL Server นอกจากนี้ยังเก็บ User account, System error message, active locks, database – backup device, system – stored procedure ฯลฯ (จะใช้พื้นที่ 25 เมกะไบต์ (ดีฟอลท์) สำหรับ master database)



รูปที่ 2.36 รูปแสดงโครงสร้างของ master database

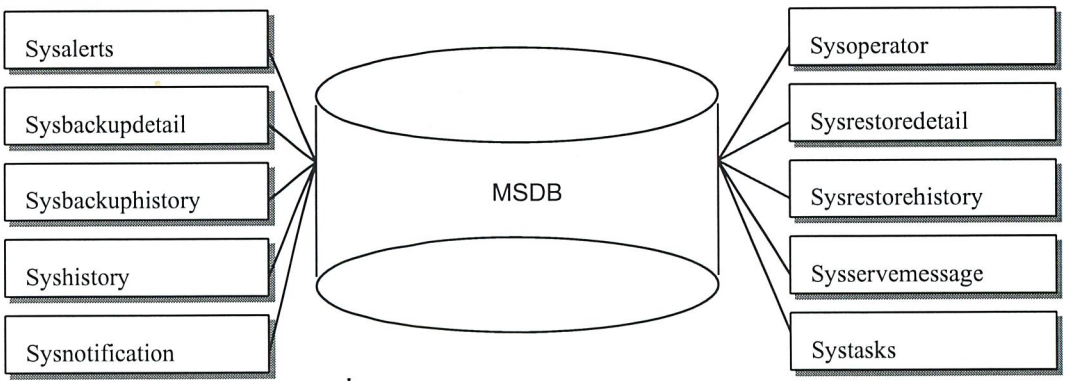
2.9.9.2 System Database



รูปที่ 2.37รูปแสดงลักษณะของ System Tables

2.9.9.3 Mdb Database

Mdb database จะสนับสนุนบริการของ SQL Excutive Task scheduling และ Alerts management ฐานข้อมูลตัวนี้จะใช้ในการแก้ปัญหาเกี่ยวกับ scheduled task mdb database จะถูกสร้างขึ้นอัตโนมัติโดยโปรแกรม setup (จะใช้พื้นที่ 6 เมกะไบต์ สำหรับ mdb Database และ 2 เมกะไบต์สำหรับ Transaction log)



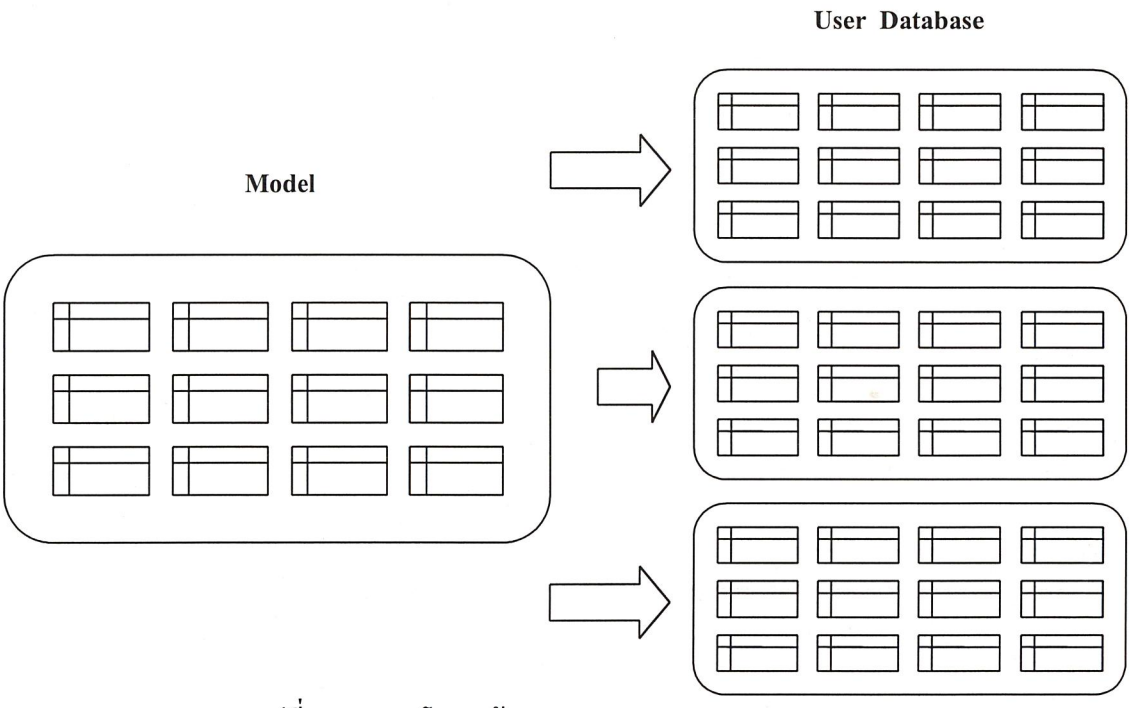
รูปที่ 2.38 รูปแสดงโครงสร้างของ mdb

System Tables	หน้าที่
Sysalerts	เก็บอินฟอร์เมชั่นเกี่ยวกับ User ทุกคนกำหนดการเตือน SQL Executive engine จะตรวจเช็ค เพื่อดูว่ามีการเข้ามาในตารางนี้ ก็จะร่วมบันทึกเหตุการณ์ลงใน Event Viewer ของ Windows NT และตัดสินใจในการเตือน (Alert)
Sysbackupdetail	รายการผลสรุปของ device ที่ใช้ในการแบ็คอัพ
Sysbackuphistory	รายการผลสรุปของทุกการทำแบ็คอัพ
Sysoperators	เก็บอินฟอร์เมชั่นเกี่ยวกับการจัดการ ดำเนินการทั้งหมดเช่น work schedules e-mail names และ pager number
Systask	เก็บอินฟอร์เมชันเกี่ยวกับ User ทั้งหมด กำหนดงาน SQL Executive engine จะคอยตรวจดูว่ามีการเข้ามาในตารางนี้เพื่อกำหนดงาน (Task)
Syshistory	เก็บอินฟอร์เมชั่นในอดีเกี่ยวกับ alert และ Task ที่ได้ทำไปแล้ว (ทั้งสำเร็จและไม่สำเร็จ) เก็บหลักฐานในการส่ง e mail จำนวนครั้งในการพยายามสำหรับงาน (task) วันเวลาในการทำงาน ฯลฯ
Sysnotifications	เก็บ alert ที่เกิดขึ้นและวิธีการที่ใช้ส่ง(e – mail หรือ pager)
Sysrestoredetail	รายการผลสรุปของ device ที่ใช้ในการโหลด (Restore)
Sysrestorehistory	รายการผลสรุปของการโหลด (restore) ทุกครั้ง

ตารางที่ 2.4 ตารางแสดง System Tables ของ mdb Database

2.9.9.4 Model Database

Model Database จะบรรจุตาราง (System Table) ฐานข้อมูลของ User ทุกคน ทุกครั้งที่มีการสร้างฐานข้อมูล SQL Server จะใช้ model Database เป็นทีมเพลทเราสามารถจะปรับเปลี่ยน เพื่อ customize โครงสร้างของฐานข้อมูลที่สร้างขึ้นใหม่ และสิ่งที่ได้ทำบน model จะให้ผลกับทุกฐานข้อมูลใหม่ รูปแบบในการเปลี่ยนแปลงบน model Database จะรวมถึงการเซตสิทธิในการเข้าถึงอย่างแน่นอน การเปลี่ยนแปลงจะรวมถึงการเพิ่ม User กำหนดชนิดข้อมูล rule defaluts หรือ stored procedure และ database configuration option ก็สามารถจะเซตใน model Database ได้เช่นกัน (ใช้พื้นที่ 1 เมกะไบต์ สำหรับ model Database)

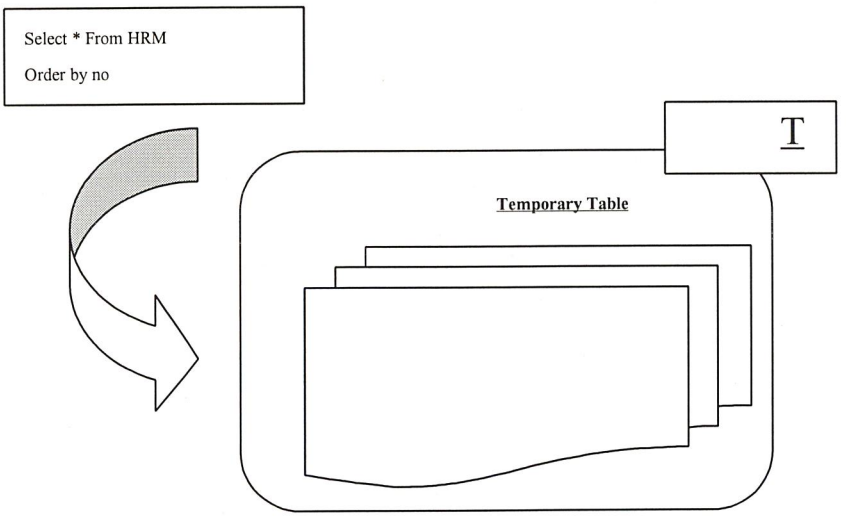


รูปที่ 2.39 แสดงโครงสร้างของ model Database

2.9.9.5 Tempdb Database

Tempdb database จะจัดเตรียมที่เก็บสำหรับตารางชั่วคราว และงานชั่วคราวที่ต้องเก็บ (storage) tempdb Database จะทำการแชร์พื้นที่ว่างในการทำงาน ซึ่งจะใช้โดยฐานข้อมูลทั้งหมดบน SQL Server ตารางชั่วคราวทั้งหมดจะเก็บอยู่ใน tempdb โดยจะไม่คำนึงถึงฐานข้อมูลที่ user กำลังใช้อยู่ และเมื่อใดก็ตามที่ User ไม่ได้เชื่อมต่อกับ SQL Server แล้ว User 's tempdb ที่อยู่ภายจะถูกเคลีย์ด้วย และถ้าระบบเกิดหยุดทำงาน (Stop) สิ่งที่อยู่ภายใน tempdb ทั้งหมดจะถูกเคลีย์ทันที tempdb Database ยังใช้ในการจัดเรียงอินฟอร์มเมชั่นได้ เช่น เมื่อ query จัดการ (execute) บนตารางที่ปราศจาก index ก็จะมีการสร้าง index ชั่วคราวขึ้นใน tempdb บางระบบ stored procedure จะใช้ tempdb Queries จะจัดเรียงหรือกรองอินฟอร์เมชั่นก็จะใช้ Tempdb ในการเก็บผลลัพธ์ระดับกลาง

Tempdb database สามารถจะอยู่ในหน่วยความจำ RAM โดยการเซต SQL Server Configura Option : tempdb in RAM (MB) ซึ่งจะมีผลกระทบครอบคลุมถึง Server Performance เพราะ queries จะจัดเรียงโดยใช้ tempdb ในหน่วยความจำ RAM จัดการอย่างรวดเร็วกว่าบนฮาร์ดิสก์ ตัวเลือกนี้ไม่ควรใช้เว้นแต่ว่าเซิร์ฟเวอร์จะมีหน่วยความจำ Ram อย่างน้อย 64 เมกะไบต์ (จะใช้พื้นที่สำหรับ tempdb Database เพียง 2 เมกะไบต์เป็นค่าดีฟอลท์ สามารถปรับเปลี่ยนได้โดย SA)



รูปที่ 2.40 รูปแสดงโครงสร้างของ tempdb Database

2.9.10 คุณสมบัติใหม่ๆที่เพิ่มเข้ามาใน SQL Server เวอร์ชั่น 7.0

2.9.10.1 ความสะดวกในการใช้งาน

สามารถทำงานบนระบบปฏิบัติการ Windows ได้ตั้งแต่ Windows 95 บนเครื่อง Laptop จนถึง Windows NT Server for Enterprise เพื่อให้ง่ายแก่การใช้งาน จึงมีการใช้ DatabaseName_Data.mdfและ DatabaseName_Log.ldf แทนการใช้ DeviceName.dat โดยวิธีนี้ จะมีข้อแตกต่างจากการใช้DAT files ดังนี้ MDF/LDF files จะมีการเพิ่ม และลดขนาดเอง โดยอัตโนมัติ เมื่อมีการเพิ่มและลบข้อมูล ในฐานข้อมูล และ MDF/LDF files จะเก็บข้อมูลได้เพียง 1 ฐานข้อมูลMicrosoft Management Console (MMC) ถูกนำมาใช้แทน Enterprise Managerมีการเพิ่มความสามารถ ในส่วนของการดูแลระบบฐานข้อมูล (Automatic Configuration andTuning) รวมถึง Wizard ต่างๆที่เพิ่มขึ้นมา รวมถึงการทำการติดตั้งระบบ โดยไม่ต้องมีผู้ดูแล (FullUnattended Installation) มีการเพิ่มความสามารถ ในการกำหนด Roles เพื่อให้ผู้ดูแลระบบฐานข้อมูลสามารถกำหนดกลุ่มของผู้ใช้ นอกเหนือจาก การกำหนดกลุ่มของ Windows NT (Group)เพื่อช่วยให้สะดวกในการใช้งาน และแบ่งเบาภาระการทำงาน ของผู้ดูแลระบบ

2.9.10.2 ขีดความสามารถ และการขยาย และลดขนาดของฐานข้อมูล

มีการเพิ่มขีดความสามารถในส่วนของ การสอบถามข้อมูล (Advanced Query Processing)โดยมีการนำเทคนิคของการใช้ Hash and sort merge joins และ Multiple indicesเพื่อช่วยเพิ่มความรวดเร็ว ในการสอบถาม รวมถึง การสอบถามข้อมูลต่างสถานที่ และต่างรูปแบบ(Heterogeneous Query) เช่น การสอบถามข้อมูล โดยที่ข้อมูลบางส่วน อยู่ที่ Remote Siteและบางส่วนอยู่ที่ Local Site รวมถึงการที่ข้อมูลบางส่วน เป็นชนิดของฐานข้อมูลในรูปแบบอื่น เช่น ORACLE หรือ Excel File สามารถ LOCK ข้อมูลได้ตั้งแต่ระดับ TABLE, PAGE จนถึงระดับ ROW (Row-level Locking)โครงสร้างการเก็บข้อมูล ได้มีการออกแบบใหม่ เพื่อให้มีประสิทธิภาพสูงขึ้น โดยที่ขนาดของ Page Size ได้มีการ เพิ่มขึ้นจาก 2K เป็น 8K ทำให้ขนาดของ Character สามารถเพิ่ม จาก 255 Bytes เป็น 8000 Bytes และ Column ต่อ 1 Table จาก 250 เป็น 1,024 Cols /Table จึงเป็นสาเหตุให้ SQL Server 7.0 สามารถรองรับข้อมูล ที่มีขนาดใหญ่มากขึ้น จาก 1 TB เป็น 1,048,516 TB โดยทั้งยังเพิ่มความสามารถในการเพิ่ม หรือลดขนาดของฐานข้อมูล ได้โดยอัตโนมัติ

2.9.10.3 การกระจายข้อมูล

SQL AGENT ถูกนำมาใช้แทน SQL EXECUTIVE สนับสนุนการทำงาน ในสภาวะแวดล้อมที่มีการ UPDATE ข้อมูลไปยังหลาย Server (Multiserver Administration) รวมถึง การสนับสนุนการทำงานบน Internet โดยมีการกำหนด Anonymous Subscriber และเพิ่มความสะดวก ในการทำงานด้วย New Web Assistant มีการเพิ่มความสามารถ ในการเปลี่ยนแปลงรูปแบบของข้อมูล (Data Transformation Service, DTS) โดยการเพิ่มความสามารถของ OLE DB ให้กับ BCP จึงทำให้เพิ่มความรวดเร็ว และความสามารถในการ IMPORT / EXPORT ฐานข้อมูลต่างประเภท

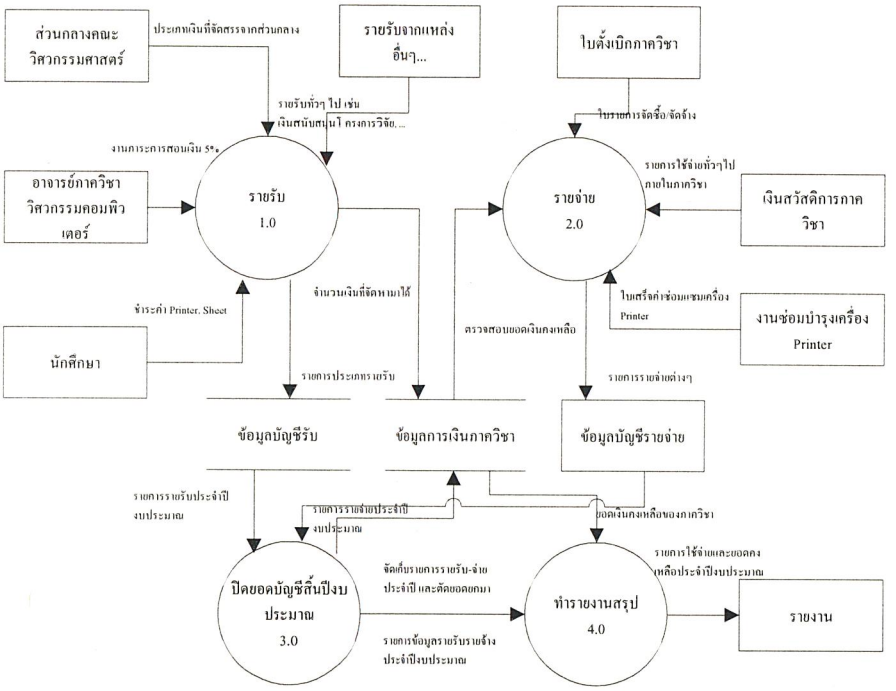
2.9.10.4 สนับสนุนการพัฒนาระบบฐานข้อมูล

รองรับการเชื่อมต่อกับฐานข้อมูล โดยใช้ OLE DB, ODBC, และ JDBC เพิ่มความสามารถในการเชื่อมต่อระหว่างฐานข้อมูล โดยใช้ ODBC 3.0 Driver for SQL Server ภายใน Driver ใหม่สำหรับ SQL Server นี้ ได้มีการ เพิ่ม Replication APIs และ BCP APIs เพื่อในการทำ Bulk Copy รวมถึง การสนับสนุนการทำงาน และ ชนิดข้อมูลใหม่ ที่มีใน MS SQL Server 7.0 และยังสนับสนุนการเข้ารหัสของ Password เพื่อใช้ใน การรักษาความปลอดภัยของระบบ SQL Server 7.0 สนับสนุนการใช้ UNICODE Datatype จึงสามารถเก็บข้อมูล ได้หลายภาษา (Multi-language Data) Microsoft SQL Server 7.0 จึง เป็นระบบฐานข้อมูลที่น่าสนใจ ไม่ว่าจะเป็น Internet/ Intranet Applications หรือฐานข้อมูล สำหรับข้อมูลธุรกิจ โดยเฉพาะสำหรับข้อมูลสินค้าคงคลัง ซึ่งมีขนาดข้อมูลที่ใหญ่ และต้องการเครื่องมือช่วย ในการวิเคราะห์ข้อมูล (Online Analytical Processing – OLAP) Microsoft พร้อมแล้ว สำหรับการเผชิญหน้า กับระบบฐานข้อมูลขนาดใหญ่จากค่ายต่างๆ

บทที่ 3
การออกแบบโปรแกรม

3.1 ระบบงานด้านการเงินภาควิชาวิศวกรรมคอมพิวเตอร์

ระบบงานด้านการเงินของภาควิชาวิศวกรรมคอมพิวเตอร์ ประกอบไปด้วยบัญชีรายรับ – รายจ่ายที่มาจากแหล่งต่างๆ ที่แตกต่างกัน ซึ่งจากเดิมนั้นในการทำบัญชีต่างๆ ของรายรับและรายจ่ายนั้น ทำโดยเจ้าหน้าที่ธุรการภาควิชา จะเป็นผู้ลงบัญชีแยกตามประเภทว่าเป็นบัญชีรายรับหรือรายจ่าย ซึ่งจัดทำโดยการบันทึกลงสมุดบัญชีทุกครั้งที่มีรายการเข้ามา (ซึ่งในโครงงานเราทำการสร้างฐานข้อมูลเพื่อใช้ในการจัดเก็บรายการต่างๆ ให้มีประสิทธิภาพ) ซึ่งสามารถแยกบัญชีรายรับ – รายจ่าย ตามประเภทที่จัดเก็บโดยสามารถแทนได้ด้วยดาต้าโฟลไดอะแกรมดังรูป

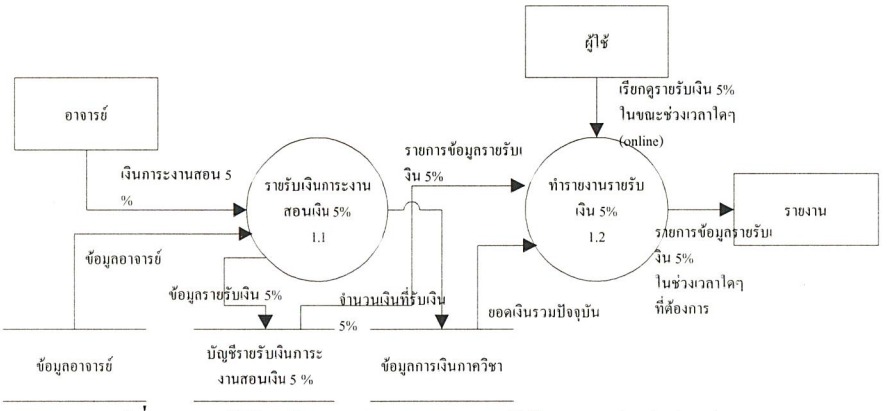


รูปที่ 3.1 ดาต้าโฟลไดอะแกรมของระบบงานทั้งหมด

จากดาต้าโพลไดอะแกรมหลักเราสามารถแยกระบบงานย่อยลงไปอีกได้ดังนี้

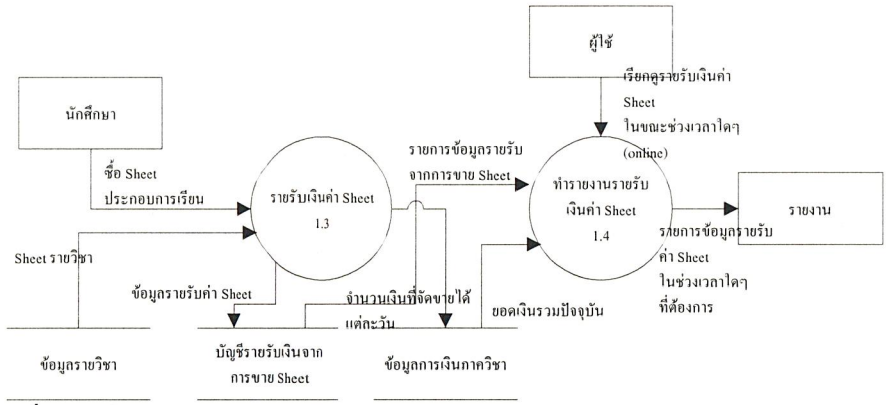
3.1.1 บัญชีรายรับ ประกอบไปด้วยรายรับจากแหล่งต่างๆ แบ่งได้ตามประเภทดังนี้

3.1.1.1 รายได้จากเงินภาระการสอนของอาจารย์เงิน 5 เปอร์เซ็นต์ ซึ่งเงินส่วนนี้จะจัดเก็บได้ในทุกๆ เทอมการศึกษาโดยหนึ่งเทอมจะจัดเก็บได้ครั้งเดียว เจ้าหน้าที่ธุรการจะทำการคำนวณภาระการสอนของอาจารย์แต่ละคนเพื่อหายอดเงินนำมาลงบัญชี โดยคำนวณจากภาระการสอนของอาจารย์ที่ทำการสอนภาคพิเศษหรือภาคสมทบ ซึ่งเงินส่วนนี้จะถูกนำมาลงในบัญชีเงินรายได้ 5 % เพื่อนำไปใช้เป็นสวัสดิการภายในภาควิชา และเมื่อครบเทอมการศึกษาในแต่ละปีจะมีการตัดยอดเงินงบประมาณในส่วนนี้ โดยถ้ายังมียอดเงินคงเหลือก็จะยกไปเป็นยอดยกมาในเทอมต่อไป ซึ่งเราสามารถเขียนแทนด้วยดาต้าโฟลไดอะแกรมดังรูป



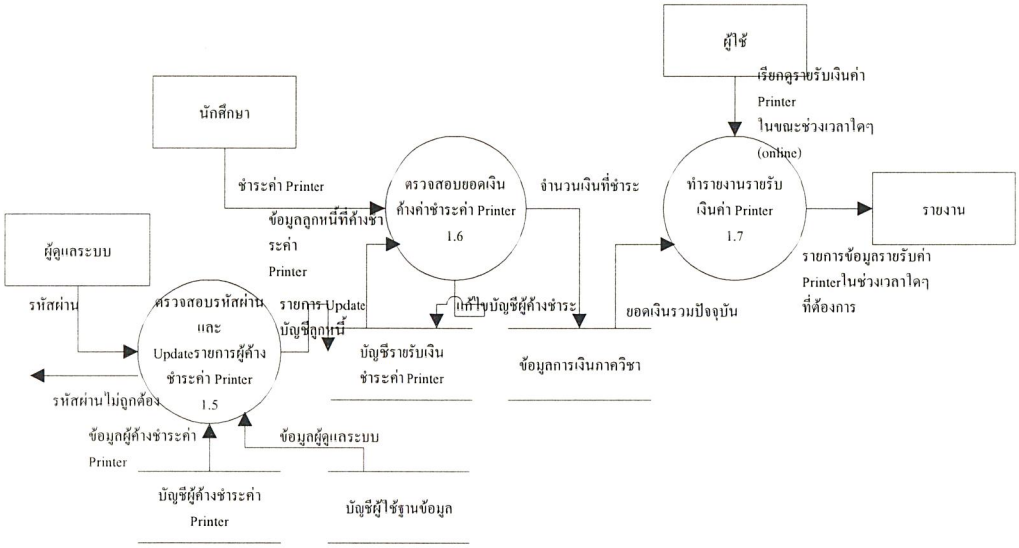
รูปที่ 3.2 ดาต้าโฟลไดอะแกรมแสดงรายได้เงิน 5 เปอร์เซ็นต์

3.1.1.2 รายได้จากการขายเอกสารประกอบการเรียน ภาควิชาจะทำการถ่ายเอกสารประกอบการเรียนแล้วจัดวางไว้บริเวณเคาร์เตอร์หน้าห้องธุรการภาคแล้วให้นักศึกษาที่ต้องการซื้อนำเงินมาใส่ไว้ในกระป๋องที่เตรียมไว้ให้ โดยเงินในส่วนนี้จะจัดเก็บได้ทุกๆ วันโดยเมื่อถึงเวลาปิดห้องธุรการเจ้าหน้าที่ธุรการจำทำการนับยอดเงินที่ได้จากการขายแล้วทำการบันทึกลงในบัญชีรายได้จากการจัดขายเอกสาร ซึ่งเราสามารถเขียนแทนด้วยดาต้าโฟลไดอะแกรมดังรูป



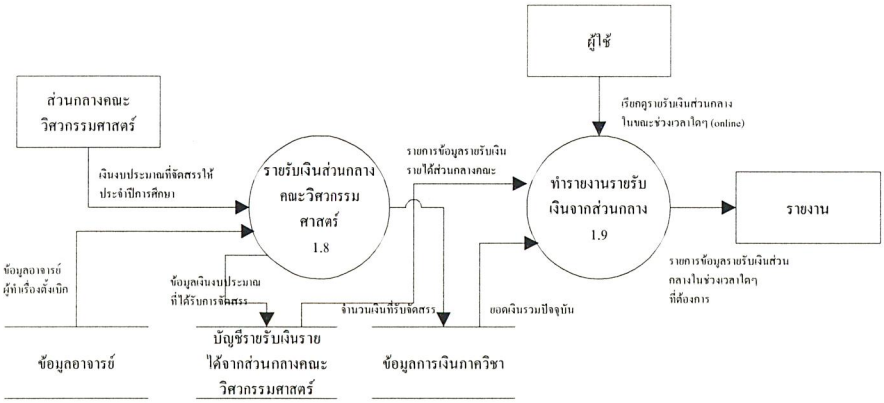
รูปที่ 3.3 ดาต้าโฟลไดอะแกรมแสดงรายได้จากการขายเอกสารประกอบการเรียน

3.1.1.3 รายได้จากการชำระค่าปริ้นเตอร์ ภาควิชาได้จัดเตรียมเครื่อง Laser Printer ไว้ให้บริการแก่นักศึกษาประจำภาควิชาคอมพิวเตอร์โดยนักศึกษาที่ทำการพิมพ์งานจากเครื่อง Printer ใน Compnet Server จะมีบัญชีลูกหนี้ที่ทำการสั่งพิมพ์จัดเก็บอยู่ในฐานข้อมูลใน Compnet โดยในทุกๆ เทอมจะมีเจ้าหน้าที่จากห้อง ISAG จะนำรายการค้างชำระค่า Printer มาให้แก่เจ้าหน้าที่ธุรการเพื่อทำการจัดเก็บเงิน ซึ่งเงินในส่วนนี้จะได้มาเป็นช่วงๆ ตามแต่นักศึกษาคนใดจะมาชำระเงิน แต่ข้อมูลผู้ค้างชำระจะมีการอัพเดททุกๆ เทอม ซึ่งเราสามารถเขียนแทนด้วยดาต้าโฟลไดอะแกรมดังรูป



รูปที่ 3.4 ดาต้าโฟลไดอะแกรมแสดงรายได้เงินค่าปริ้นเตอร์

3.1.1.4 รายได้จากเงินงบประมาณจากส่วนกลางคณะวิศวกรรมศาสตร์ เงินรายได้ส่วนนี้เป็นเงินรายได้หลักของภาควิชาที่จะได้รับการจัดสรรมาทุกๆ ปีการศึกษาและจะบันทึกไว้ในบัญชีรายรับจากส่วนกลางคณะ ซึ่งแบ่งออกได้เป็น 3 ประเภท คือ เงินงบประมาณปกติ, เงินงบประมาณเร่งรัด และเงินเทียบโอน ซึ่งเงินในส่วนนี้ภาควิชาจะนำมาจัดสรรแบ่งออกเป็นหมวดงานย่อยๆ เพื่อนำไปใช้จ่ายตามวัตถุประสงค์ต่างๆ ตามแต่งานในภาควิชา ซึ่งเราสามารถเขียนแทนด้วยดาต้าโฟลไดอะแกรมดังรูป

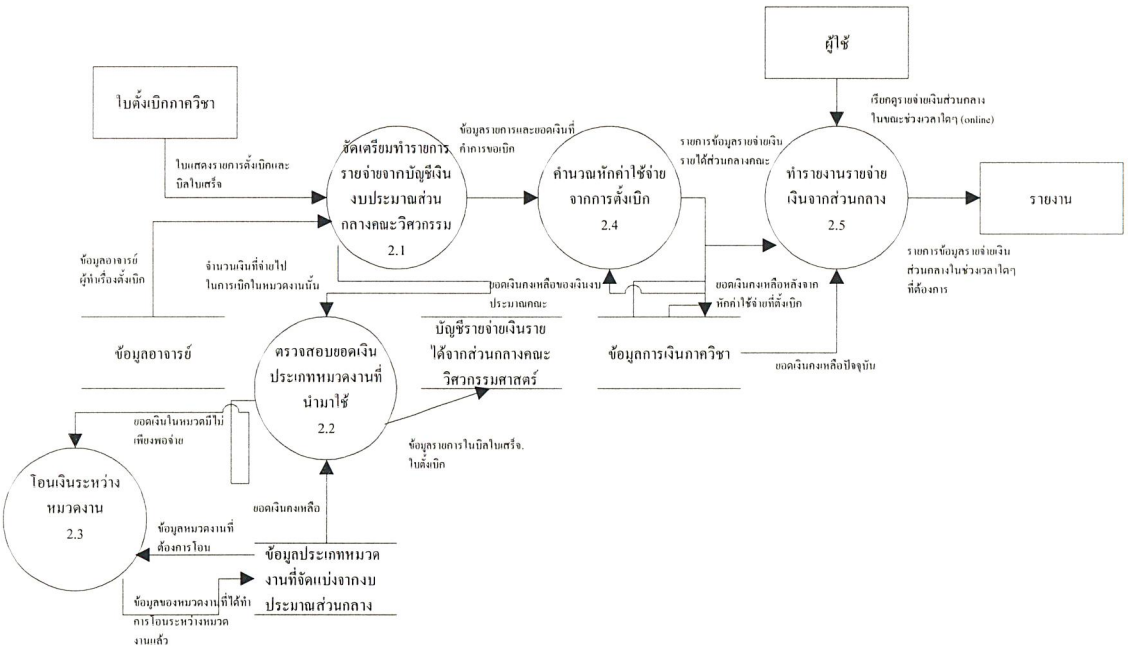


รูปที่ 3.5 ดาต้าโฟลไดอะแกรมแสดงรายได้จากเงินงบประมาณจากส่วนกลางคณะวิศวกรรมศาสตร์

3.1.2 บัญชีรายจ่าย ประกอบไปด้วยรายจ่ายจากแหล่งต่างๆ แบ่งได้ตามประเภทดังนี้

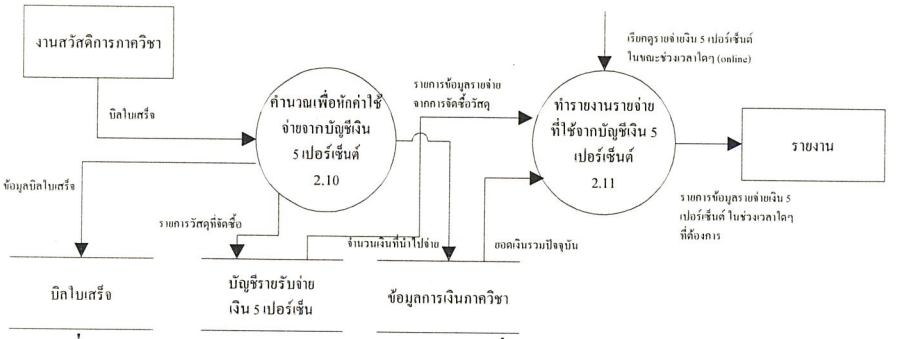
3.1.2.1 รายจ่ายที่เกี่ยวกับงานจัดซื้อ/จัดจ้าง ซึ่งรายจ่ายในส่วนนี้เป็นรายจ่ายหลักที่เกิดจากการเรียน การสอนในภาควิชา เช่น การจัดซื้ออุปกรณ์มาใช้ในการทดลอง, การจัดซื้ออุปกรณ์เพื่อสนับสนุนการทำโครงงานของนักศึกษา ฯลฯ ซึ่งการขอเบิกเงินในส่วนของรายจ่ายประเภทนี้

อาจารย์จะทำเรื่องเขียนใบตั้งเบิกภาควิชาเพื่อนำมาขอเบิกเงินที่ได้ทำการจัดซื้ออุปกรณ์ไป (หรืออาจจะเป็นนักศึกษาโดยนำบิลใบเสร็จมาให้อาจารย์ผู้ควบคุมเพื่อทำเรื่องขอเบิกเงิน) ซึ่งในความเป็นจริงแล้วรายจ่ายจากเงินในส่วนนี้จะนำเงินในส่วนของบัญชีรายได้จากส่วนกลางคณะวิศวกรรมมาใช้ แต่เนื่องด้วยทางคณะไม่อนุมัติให้มีการใช้จ่ายเงินในส่วนที่จัดสรรนี้ได้ในกระการอื่นที่ไม่ได้เกี่ยวกับการเรียนการสอน (เช่น ซื้อเครื่องคอมพิวเตอร์, ซื้อโต๊ะ, ซื้อกระดาษฯลฯ) ดังนั้นจึงจำเป็นต้องมีการแก้ไขบิลใบเสร็จให้เป็นรายการอย่างอื่นเพื่อสามารถที่จะเสนอกลับไปให้แก่คณะได้เมื่อครบช่วงงบประมาณประจำเป็น (เช่น ทำการแก้ไขรายการเป็นอุปกรณ์ที่ใช้ในการทำแลป หรือทำโครงงานแทนเช่น IC, Transiter, Resister ฯลฯ) ซึ่งรายการที่แก้ไขจะต้องสามารถทำการตรวจสอบได้ว่างานที่นำไปใช้จริงเป็นอะไร และสามารถเรียกดูได้อย่างถูกต้อง ซึ่งเราสามารถเขียนแทนด้วยดาต้าโฟลไดอะแกรมดังรูป



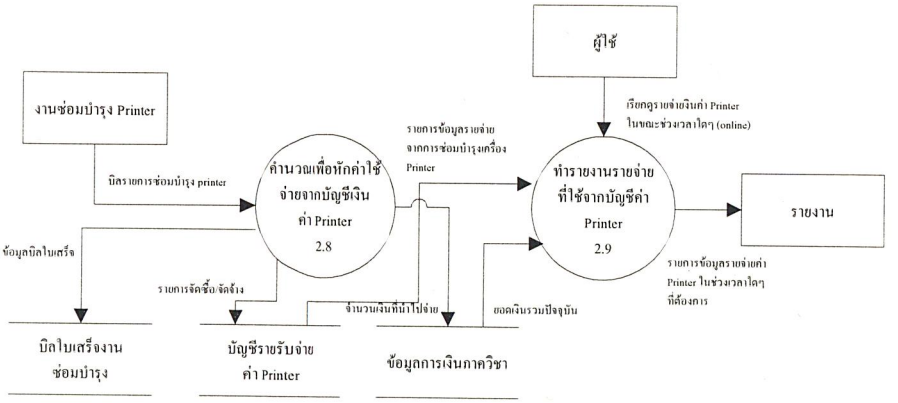
รูปที่ 3.6 ดาต้าโฟลไดอะแกรมแสดงรายจ่ายที่เกี่ยวกับงานจัดซื้อ/จัดจ้าง

3.1.2.2 รายจ่ายเกี่ยวกับสวัสดิการภายในภาควิชา เช่น ค่าใช้จ่ายในการจัดเลี้ยงสังสรรร, ค่าใช้จ่ายในการทำกิจกรรมของนักศึกษาเลี้ยงน้องภาค, ค่าเงินพิเศษเจ้าหน้าที่ธุรการ, ค่าใช้จ่ายในการทำบุญภาควิชา ฯลฯ ซึ่งนำเงินจากบัญชีรายรับเงิน 5 % มาใช้ ซึ่งเราสามารถเขียนแทนด้วยดาต้าโฟลไดอะแกรมดังรูป



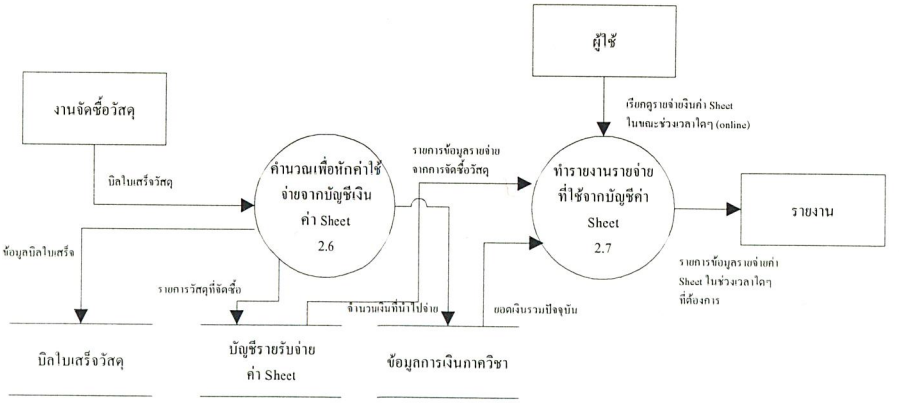
รูปที่ 3.7 ดาต้าโฟลไดอะแกรมแสดงรายจ่ายเกี่ยวกับสวัสดิการภายในภาควิชา

3.1.2.3 รายจ่ายที่เกี่ยวกับงานซ่อมบำรุงเครื่องปริ้นเตอร์ เช่น การซ่อมแซมเครื่อง Printer เมื่อเกิดการขัดข้องโดยจะนำเงินในส่วนของบัญชีเงินรายได้ที่การจัดเก็บเงินค่าค้างชำระค่า Printer ของนักศึกษามาใช้ ซึ่งเราสามารถเขียนแทนด้วยดาต้าโฟลไดอะแกรมดังรูป



รูปที่ 3.8 ดาต้าโฟลไดอะแกรมแสดงรายจ่ายที่เกี่ยวกับงานซ่อมบำรุงเครื่องปริ้นเตอร์

3.1.2.4 รายจ่ายที่เกี่ยวกับงานเอกสาร เช่น การซื้อกระดาษสำหรับพิมพ์งาน โดยจะนำเงินในส่วนของบัญชีเงินรายได้ที่การจัดเก็บเงินค่าขายเอกสารให้แก่นักศึกษามาใช้ ซึ่งเราสามารถเขียนแทนด้วยดาต้าโฟลไดอะแกรมดังรูป

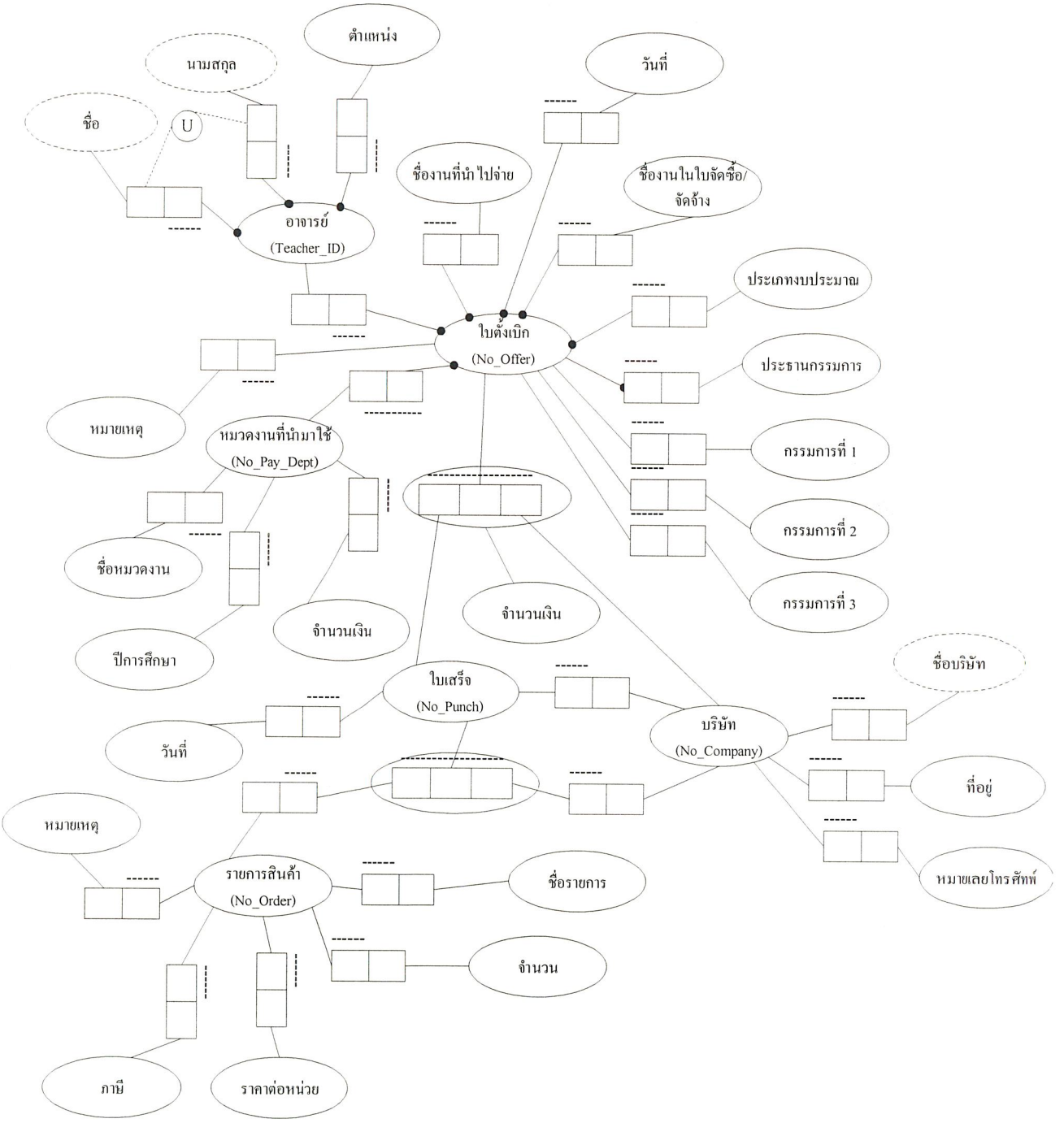


รูปที่ 3.9 ดาต้าโฟลไดอะแกรมแสดงรายจ่ายที่เกี่ยวกับงานเอกสาร

นอกจากรายการรายรับ – รายจ่ายต่างๆ แล้วยังมีส่วนของการปิดยอดเงินงบประมาณประจำปีการศึกษา ซึ่งจะทำการทำสรุปยอดรายรับ – รายจ่ายในช่วงปีงบประมาณเพื่อนำเสนอ และคำนวณหายอดเงินคงเหลือในแต่ละบัญชีเพื่อนำเสนอ ในส่วนของเงิน 5 % เมื่อทำการปิดยอดปีงบประมาณจะนำเงินในส่วนที่เหลือที่ใช้ไม่หมดนำมาเป็นยอดยกมาเพื่อใช้ในเทอมต่อไปเรื่อยๆ และในส่วนของรายงานผู้ใช้สามารถทำการเรียกดูเพื่อตรวจสอบรายรับ – รายจ่ายที่สนใจได้โดยทันที ซึ่งจะมีการทำรายงานสรุปทุกๆ สิ้นปีงบประมาณ

3.2 การออกแบบฐานข้อมูล

การออกแบบโปรแกรมให้ทำงานได้ตามหน้าที่ใช้งานของระบบ เริ่มต้นที่การออกแบบระบบฐานข้อมูลขึ้นมาเพื่อรองรับการใช้งาน โดยการออกแบบฐานข้อมูลใช้วิธีของไนแอมซึ่งเมื่อออกแบบ แล้วจะได้ออกมาดังรูป



รูปที่ 3.10 แสดงแผนภาพไนแอม แสดงข้อมูลและความสัมพันธ์ของข้อมูลของงานโดยรวม

จากแผนภาพไนแอมเราสามารถทำการแปลงเป็นเทเบิลฐานข้อมูลต่างๆได้ดังนี้

3.2.1 เทเบิลเก็บข้อมูลยอดค้างชำระค่าปริ้นเตอร์ (BalancePrinter Table) เป็นเทเบิลที่ใช้สำหรับเก็บข้อมูลยอดค้างชำระเงินค่าการใช้งานปริ้นเตอร์ของผู้ที่มีชื่ออยู่ในทะเบียนผู้ใช้งานเครือข่ายของภาควิชา โดยมีฟิลด์ข้อมูลต่างๆดังตาราง

ชื่อฟิลด์ข้อมูล	ชนิดข้อมูล	ขนาด (ไบต์)
Student_Code	VarChar	10
Balance_Money	Float	8

ตารางที่ 3.1 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลยอดค้างชำระค่าปริ้นเตอร์

3.2.2 เทเบิลเก็บข้อมูลบริษัท (Company Table) เป็นเทเบิลสำหรับเก็บข้อมูลของบริษัทต่างๆที่ภาควิชาติดต่อในการจัดซื้อ โดยมีฟิลด์ข้อมูลต่างๆดังตาราง

ชื่อฟิลด์ข้อมูล	ชนิดข้อมูล	ขนาด (ไบต์)
No_Company	Integer	4
Name	Varchar	40
Address	Varchar	80
Phone	Varchar	15

ตารางที่ 3.2 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลบริษัท

3.2.3 เทเบิลเก็บข้อมูลประเภทเงินงบประมาณ (Department_Payment Table) เป็นเทเบิลสำหรับเก็บข้อมูลประเภทเงินงบประมาณของภาควิชา โดยมีฟิลด์ต่างๆดังตาราง

ชื่อฟิลด์ข้อมูล	ชนิดข้อมูล	ขนาด (ไบต์)
No_Pay_Dept	Integer	4
Years	Integer	4
Total	Float	8

ตารางที่ 3.3 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลประเภทเงินงบประมาณ

3.2.4 เทเบิลเก็บข้อมูลชื่อประเภทเงินงบประมาณ (Department_Payment_Name Table) เป็นเทเบิลสำหรับเก็บข้อมูลชื่อประเภทของเงินงบประมาณภาควิชาโดยมีความสัมพันธ์กับเทเบิลเก็บข้อมูลประเภทเงินงบประมาณ โดยมีฟิลด์ต่างๆดังตาราง

ชื่อฟิลด์ข้อมูล	ชนิดข้อมูล	ขนาด (ไบต์)
No_Pay_Dept	Integer	4
Name_Payment	VarChar	20

ตารางที่ 3.4 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลชื่อประเภทเงินงบประมาณ

3.2.5 เทเบิลเก็บข้อมูลการใช้เงินงบประมาณในใบตั้งเบิก(Department_Payment_To_Offer Table) เป็นเทเบิลสำหรับเก็บข้อมูลการใช้งบประมาณที่จัดแบ่งไว้โดยการตั้งเบิก

ชื่อฟิลด์ข้อมูล	ชนิดข้อมูล	ขนาด (ไบต์)
No_Pay_Dept	Integer	4
No_Offer	Integer	4

ตารางที่ 3.5 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลการใช้เงินงบประมาณในใบตั้งเบิก

3.2.6 เทเบิลเก็บข้อมูลใบตั้งเบิก (Offer Table) เป็นเทเบิลสำหรับเก็บข้อมูลใบตั้งเบิกที่ใช้เบิกจากเงินงบประมาณของภาควิชาที่ถูกแบ่งไว้

ชื่อฟิลด์ข้อมูล	ชนิดข้อมูล	ขนาด (ไบต์)
No_Offer	Integer	4
Offer_Name	Varchar	45
Kind_Of_Budget	Varchar	45
Project_Name_Fact	Varchar	40
Project_Name_Punch	Varchar	40
Dat_	Datetime	8
Head_Committee	Varchar	45
Commitee1	Varchar	45
Commitee2	Varchar	45
Commitee3	Varchar	45
Commant	Varchar	60

ตารางที่ 3.6 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลใบตั้งเบิก

3.2.7 เทเบิลเก็บข้อมูลใบเสร็จต่างๆในใบตั้งเบิก (Offer_To_Punch Table) ใช้สำหรับเก็บข้อมูลเลขที่ใบเสร็จที่ใช้ในการตั้งเบิกงบประมาณ

ชื่อฟิลด์ข้อมูล	ชนิดข้อมูล	ขนาด (ไบต์)
No_Offer	Integer	4
No_Punch	Varchar	15
Total	Float	8

ตารางที่ 3.7 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลใบเสร็จต่างๆในใบตั้งเบิก

3.2.8 เทเบิลเก็บข้อมูลใบเสร็จ (Order_Punch Table) ใช้สำหรับเก็บข้อมูลต่างๆในใบเสร็จที่ใช้ในการตั้งเบิก

ชื่อฟิลด์ข้อมูล	ชนิดข้อมูล	ขนาด (ไบต์)
No_Order	Integer	4
List_Name	Varchar	60
Qty	Integer	4
Price	Float	8
Commant	Varchar	100
Vat	Varchar	5

ตารางที่ 3.8 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลใบเสร็จ

3.2.9 เทเบิลเก็บข้อมูลรายจ่ายจากเงินค่าชีต (Payment_Of_Sheet Table) ใช้วำหรับเก็บข้อมูลรายจ่ายที่นำเงินค่าชีตไปใช้

ชื่อฟิลด์ข้อมูล	ชนิดข้อมูล	ขนาด (ไบต์)
Dat_	Datetime	8
Pay_for	Varchar	40
Total	Float	8

ตารางที่ 3.9 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลรายจ่ายจากเงินค่าชีต

3.2.10 เทเบิลเก็บข้อมูลรายจ่ายจากเงินค่าปริ้นเตอร์ (Payment_Of_Printer Table) ใช้สำหรับเก็บข้อมูลรายจ่ายที่นำเงินค่าปริ้นเตอร์ไปใช้

ชื่อฟิลด์ข้อมูล	ชนิดข้อมูล	ขนาด (ไบต์)
Dat_	Datetime	8
Pay_for	Varchar	40
Total	Float	8

ตารางที่ 3.10 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลรายจ่ายจากเงินค่าชีต

3.2.11 เทเบิลเก็บข้อมูลรายจ่ายจากเงิน 5 เปอร์เซ็นต์ (Payment_Of_The_Work Table) ใช้สำหรับเก็บข้อมูลรายจ่ายที่นำเงิน 5 เปอร์เซ็นต์ไปใช้

ชื่อฟิลด์ข้อมูล	ชนิดข้อมูล	ขนาด (ไบต์)
Dat_	Datetime	8
Pay_for	Varchar	40
Total	Float	8

ตารางที่ 3.11 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลรายจ่ายจากเงิน 5 เปอร์เซ็นต์

3.2.12 เทเบิลเก็บข้อมูลใบเสร็จของบริษัทต่างๆ (Punch Table) ใช้สำหรับเก็บข้อมูลเลขที่ใบเสร็จของบริษัทต่างๆ

ชื่อฟิลด์ข้อมูล	ชนิดข้อมูล	ขนาด (ไบต์)
Dat_	Datetime	8
No_Punch	Varchar	15
No_Company	Integer	4

ตารางที่ 3.12 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลใบเสร็จของบริษัทต่างๆ

3.2.13 เทเบิลเก็บข้อมูลใบเสร็จของบริษัทต่างๆและรายการต่างๆในใบเสร็จ (Punch_To_Order Table) ใช้สำหรับเก็บข้อมูลเลขที่ใบเสร็จและรายการต่างๆในใบเสร็จว่าเป็นของบริษัทใด

ชื่อฟิลด์ข้อมูล	ชนิดข้อมูล	ขนาด (ไบต์)
No_Punch	Varchar	15
No_Order	Integer	4
No_Company	Integer	4

ตารางที่ 3.13 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลใบเสร็จของบริษัทต่างๆและรายการต่างๆในใบเสร็จ

3.2.14 เทเบิลเก็บข้อมูลรายรับอื่นๆ (Receive_Else Table) ใช้สำหรับเก็บข้อมูลรายรับจากรายการอื่นๆ ที่ไม่ได้ระบุไว้

ชื่อฟิลด์ข้อมูล	ชนิดข้อมูล	ขนาด (ไบต์)
Dat_	Datetime	8
Receive_from	Varchar	40
Total	Float	8
Comment	Varchar	40

ตารางที่ 3.14 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลรายรับอื่นๆ

3.2.15 เทเบิลเก็บข้อมูลรายรับเงินงบประมาณภาควิชา (Receive_Faculity Table) ใช้สำหรับเก็บข้อมูลรายรับจากเงินงบประมาณที่แบ่งมาจากคณะวิศวกรรมศาสตร์

ชื่อฟิลด์ข้อมูล	ชนิดข้อมูล	ขนาด (ไบต์)
Dat_	Datetime	8
Years	Float	8
Kind	Varchar	15
Total	Float	8
Commant	Varchar	40

ตารางที่ 3.15 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลรายรับเงินงบประมาณภาควิชา

3.2.16 เทเบิลเก็บข้อมูลรายรับเงิน 5 เปอร์เซ็นต์ (Receive_Money5 Table) ใช้สำหรับเก็บข้อมูลรายรับเงิน 5 เปอร์เซ็นต์จากค่าสอนของอาจารย์ในภาควิชา

ชื่อฟิลด์ข้อมูล	ชนิดข้อมูล	ขนาด (ไบต์)
Dat_	Datetime	8
Receive_from	Varchar	40
Total	Float	8
Commant	Varchar	40

ตารางที่ 3.16 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลรายรับเงิน 5 เปอร์เซ็นต์

3.2.17 เทเบิลเก็บข้อมูลรายรับเงินค่าปริ้นเตอร์ (Receive_Printer Table) ใช้สำหรับเก็บข้อมูลรายรับเงิน จากการใช้งานปริ้นเตอร์ภายในภาควิชา

ชื่อฟิลด์ข้อมูล	ชนิดข้อมูล	ขนาด (ไบต์)
Dat_	Datetime	8
Receive_from	Varchar	15
Total	Float	8

ตารางที่ 3.17 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลรายรับเงินค่าปริ้นเตอร์

3.2.18 เทเบิลเก็บข้อมูลรายรับเงินค่าชีต (Receive_Sheet Table) ใช้สำหรับเก็บข้อมูลรายรับเงิน จากการขายชีตภายในภาควิชา

ชื่อฟิลด์ข้อมูล	ชนิดข้อมูล	ขนาด (ไบต์)
Dat_	Datetime	8
Total	Float	8

ตารางที่ 3.18 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลรายรับเงินค่าชีต

3.2.19 เทเบิลเก็บข้อมูลอาจารย์ (Teacher Table) ใช้สำหรับเก็บข้อมูลของอาจารย์ภายในภาควิชา

ชื่อฟิลด์ข้อมูล	ชนิดข้อมูล	ขนาด (ไบต์)
ID_Teacher	Integer	4
Name	Varchar	15
Surname	Varchar	15
Position	Varchar	30

ตารางที่ 3.19 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลอาจารย์

3.2.20 เทเบิลเก็บข้อมูลการโอนย้ายเงินระหว่างงบประมาณ (TransferMoney Table) ใช้สำหรับเก็บข้อมูลการโอนย้ายเงินระหว่างงบประมาณที่จัดสรรไว้

ชื่อฟิลด์ข้อมูล	ชนิดข้อมูล	ขนาด (ไบต์)
NO	Integer	4
Dat_	Datetime	8
Years	Integer	4
Money_Kind1	Varchar	60
Money_Kind2	Varchar	60
TransferMoney	Float	8

ตารางที่ 3.20 แสดงฟิลด์ต่างๆในเทเบิลเก็บข้อมูลการโอนย้ายเงินระหว่างงบประมาณ

3.3 การหาผลสรุปของข้อมูลจากเทเบิลต่างๆ

3.3.1 ยอดรวมของเงินรายได้จากงบประมาณภาควิชา

สามารถหาได้โดยการนำค่าในฟิลด์ข้อมูล "Total" จากเทเบิลเก็บข้อมูลรายรับเงินงบประมาณภาควิชา (Receive_Faculity Table) ทั้งหมดมารวมกัน โดยใช้คำสั่ง SQL ดังนี้

```
SELECT SUM(ToTal)
 FROM Receive_Faculity
WHERE (Years = m_EduYears)
```

โดยที่ค่าในตัวแปร m_EduYears เป็นปีที่ต้องการทราบจำนวนเงินรายได้

3.3.2 ยอดรวมของเงินรายได้จากการเก็บเงิน 5 เปอร์เซ็นต์

ยอดรวมของเงินรายได้ 5 เปอร์เซ็นต์สามารถมาได้จากสองแหล่งคือ เก็บจากยอดเงิน 5 เปอร์เซ็นต์จากค่าการสอนของอาจารย์ภายในภาควิชา และยอดเงินจากรายได้อื่นๆภายในภาควิชา ดังนั้น ยอดรวมของเงิน 5 เปอร์เซ็นต์จึงสามารถหาได้โดยการนำค่าในฟิลด์ข้อมูล "Total" จากเทเบิลเก็บข้อมูลรายรับเงิน 5 เปอร์เซ็นต์ (Receive_Money5 Table) ทั้งหมดมารวมกับค่าผลรวมของค่าในฟิลด์ข้อมูล "Total" จากเทเบิลเก็บข้อมูลรายรับอื่นๆ (Receive_Else Table) โดยใช้คำสั่ง SQL ดังนี้

- คำสั่งหายอดรวมจากเทเบิลเก็บข้อมูลรายรับเงิน 5 เปอร์เซ็นต์ (Receive_Money5 Table)

```
SELECT SUM( Total )
FROM Receive_Money5
```

-เทเบิลเก็บข้อมูลรายรับอื่นๆ (Receive_Else Table)

```
SELECT SUM( Total )
 FROM Receive_Else
```

3.3.3 ยอดรวมของเงินรายได้จากการเก็บเงินค่าปริ้นเตอร์

สามารถหาได้โดยการนำค่าในฟิลด์ข้อมูล "Total" จากเทเบิลเก็บข้อมูลรายรับเงินค่าปริ้นเตอร์ (Receive_Printer Table) ทั้งหมดมารวมกัน โดยใช้คำสั่ง SQL ดังนี้

```
SELECT SUM( Total )
FROM Receive_Printer
```

3.3.4 ยอดรวมของเงินรายได้จากการเก็บเงินค่าชีต

สามารถหาได้โดยการนำค่าในฟิลด์ข้อมูล "Total" จากเทเบิลเก็บข้อมูลรายรับเงินค่าชีต (Receive_Sheet Table) ทั้งหมดมารวมกัน โดยใช้คำสั่ง SQL ดังนี้

```
SELECT SUM( Total )
FROM Receive_Sheet
```

3.3.5 ยอดรวมของเงินจ่ายที่ใช้โดยเงินค่าชีต

สามารถหาได้โดยการนำค่าในฟิลด์ข้อมูล "Total" จากเทเบิลเก็บข้อมูลรายจ่ายเงินค่าชีต (Payment_Of_Sheet Table) ทั้งหมดมารวมกัน โดยใช้คำสั่ง SQL ดังนี้

```
SELECT SUM( Total )
FROM Payment_Of_Sheet
```

3.3.6 ยอดรวมของเงินจ่ายที่ใช้โดยเงินค่าปริ้นเตอร์

สามารถหาได้โดยการนำค่าในฟิลด์ข้อมูล "Total" จากเทเบิลเก็บข้อมูลรายจ่ายเงินค่าปริ้นเตอร์ (Payment_Of_Payment Table) ทั้งหมดมารวมกัน โดยใช้คำสั่ง SQL ดังนี้

```
SELECT SUM( Total )
FROM Payment_Of_Printer
```

3.3.7 ยอดรวมของเงินจ่ายที่ใช้โดยเงิน 5 เปอร์เซ็นต์

สามารถหาได้โดยการนำค่าในฟิลด์ข้อมูล "Total" จากเทเบิลเก็บข้อมูลรายจ่ายจากเงิน 5 เปอร์เซ็นต์ (Payment_Of_The_Work Table) ทั้งหมดมารวมกัน โดยใช้คำสั่ง SQL ดังนี้

```
SELECT SUM( Total )
FROM Payment_Of_The_Work
```

3.3.8 ยอดรวมของเงินจ่ายที่ใช้โดยเงินงบประมาณภาควิชา

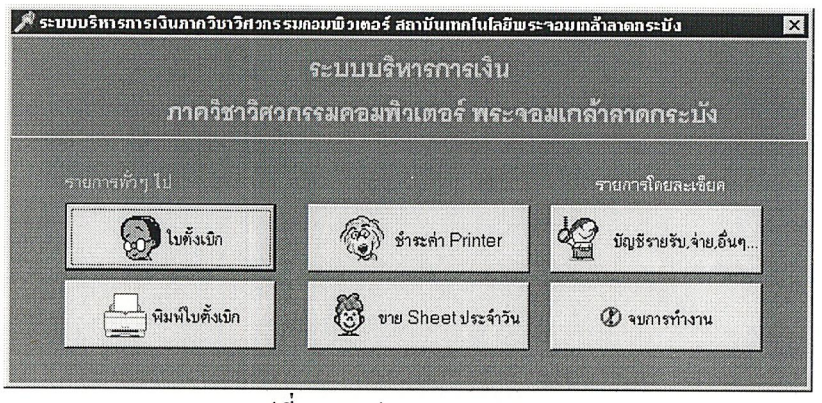
เงินงบประมาณภาควิชาจะถูกแบ่งออกเป็นงบประมาณย่อยๆอีกครั้งเพื่อนำไปใช้ในงานประเภทต่างๆเช่น ใช้เป็นงบประมาณสำหรับอุปกรณ์ห้องแลป งบประมาณสำหรับโปรเจ็คของนักศึกษา เป็นต้น เมื่อแบ่งออกเป็นเงินก้อนย่อยๆแล้ว ก็สามารถหารายจ่ายได้โดยการนำค่าในฟิลด์ข้อมูล "Total" จากเทเบิลเก็บข้อมูลใบเสร็จต่างๆในใบตั้งเบิก (Offer_To_Punch Table) ทั้งหมดมารวมกัน โดยใช้คำสั่ง SQL ดังนี้

```
SELECT SUM( Total )
FROM Offer_To_Punch
```

3.4 การออกแบบส่วนติดต่อกับผู้ใช้งานโปรแกรม

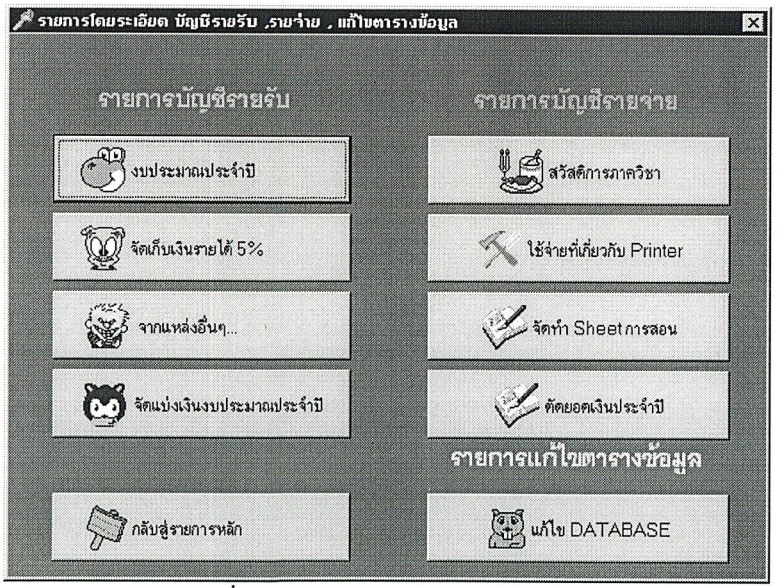
จากระบบงานบัญชีของภาควิชาวิศวกรรมคอมพิวเตอร์ดังที่ได้กล่าวมาแล้ว รวมทั้งฐานข้อมูลที่ได้ออกแบบไว้ ลำดับต่อไปคือการออกแบบหน้าจอเพื่อให้ผู้ใช้ป้อนข้อมูลเข้าสู่ฐานข้อมูลให้ตรงตามข้อกำหนดต่างๆของระบบงาน โดยหน้าจอต่างๆของโปรแกรมจะเป็นดังนี้

3.4.1 เมนูหลัก เมนูหลักเป็นหน้าจอแรกเมื่อทำการเรียกโปรแกรมขึ้นมา โดยในเมนูหลักจะประกอบไปด้วยตัวเลือกที่ใช้งานบ่อยๆเช่น ใบตั้งเบิกและการพิมพ์ใบตั้งเบิก เป็นต้น โดยมีลักษณะดังรูป



รูปที่ 3.11 รูปแสดงเมนูหลัก

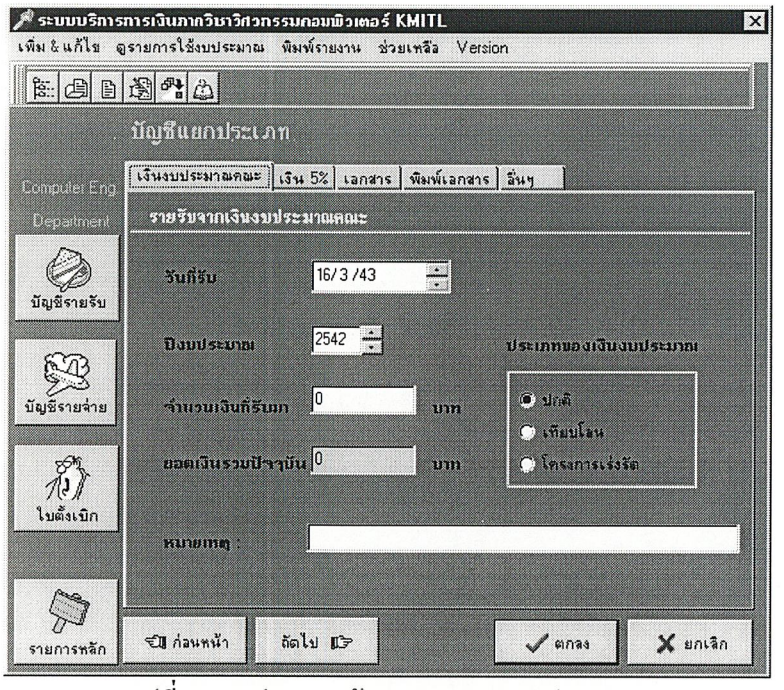
3.4.2 เมนูย่อย เมนูย่อยจะเป็นแหล่งรวมของรายการการกระทำทั้งหมดของโปรแกรมเช่น การแก้ไขข้อมูลในฐานข้อมูล การตัดยอดเงิน รวมทั้งรายการที่ไม่ค่อยใช้งานบ่อยนักเช่น รายรับเงินงบประมาณภาควิชา เป็นต้น โดยลักษณะของเมนูย่อยจะเป็นดังรูป



รูปที่ 3.12 รูปแสดงเมนูเมนูย่อย

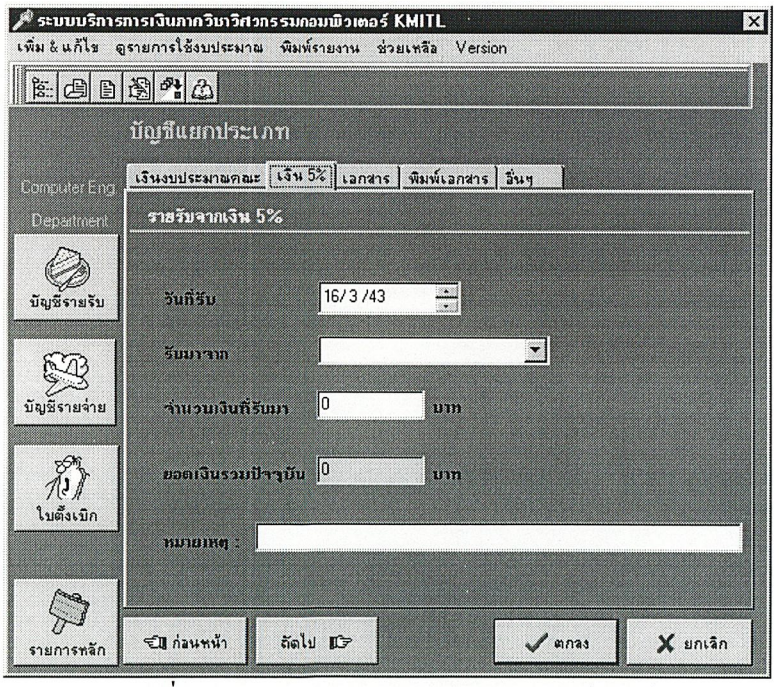
3.4.3 หน้าจอรายรับ หน้าจอรายรับเป็นหน้าจอสำหรับกรอกข้อมูลรายรับประเภทต่างๆของภาควิชาวิศวกรรมคอมพิวเตอร์ ซึ่งประกอบไปด้วยรายการต่างๆดังนี้

3.4.3.1 รายรับเงินงบประมาณคณะ เป็นหน้าจอสำหรับกรอกรายได้ของเงินงบประมาณที่ได้จากคณะวิศวกรรมศาสตร์ มีลักษณะดังรูป



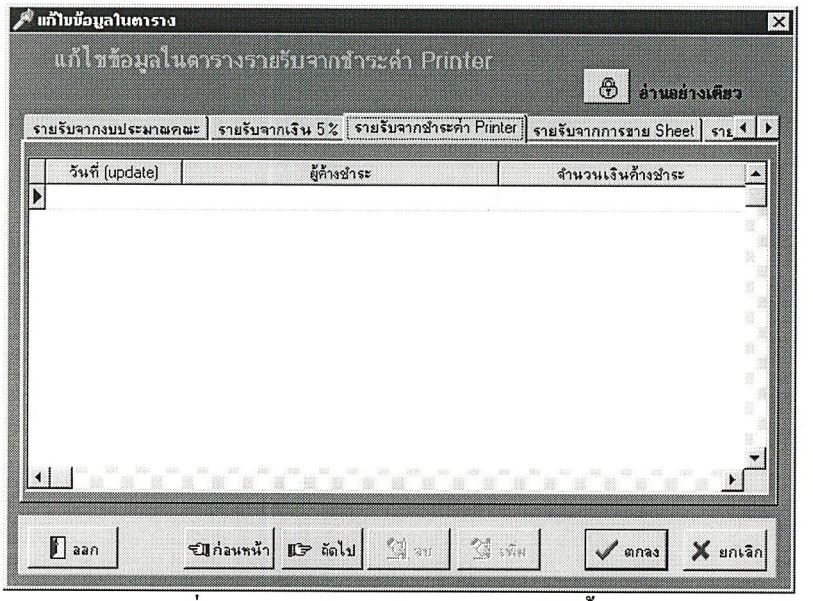
รูปที่ 3.13 รูปแสดงหน้าจอรายรับเงินงบประมาณคณะ

3.4.3.2 รายรับเงิน 5 เปอร์เซ็นต์ เป็นหน้าจอสำหรับกรอกรายได้ของเงิน 5 เปอร์เซ็นต์จากค่าสอนของอาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ มีลักษณะดังรูป



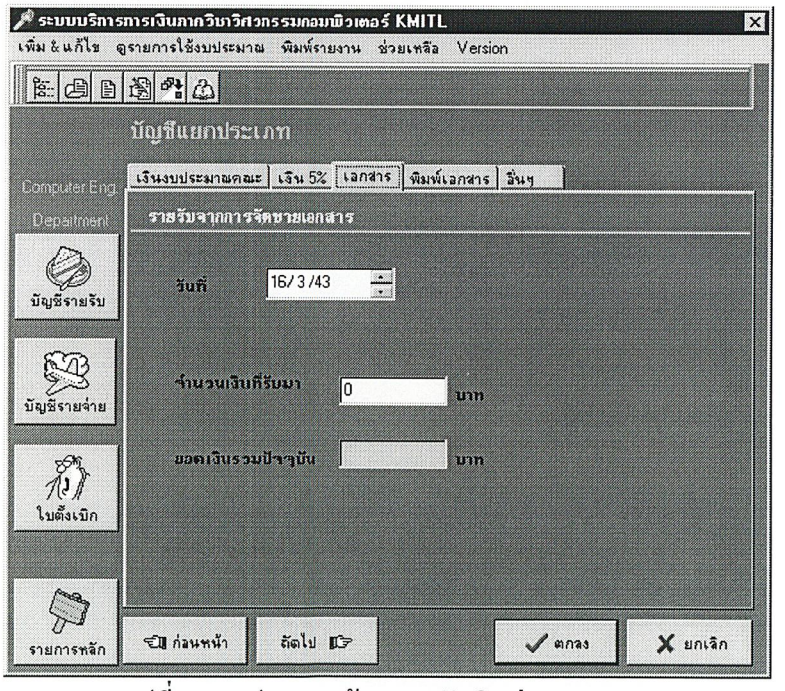
รูปที่ 3.14 รูปแสดงหน้าจอรายรับเงิน 5 เปอร์เซ็นต์

3.4.3.3 รายรับเงินค่าใช้งานปริ้นเตอร์ เป็นหน้าจอสำหรับกรอกรายได้จากการเก็บค่าบริการการใช้ปริ้นเตอร์ในการพิมพ์งานภายในภาควิชาวิศวกรรมคอมพิวเตอร์ มีลักษณะดังรูป



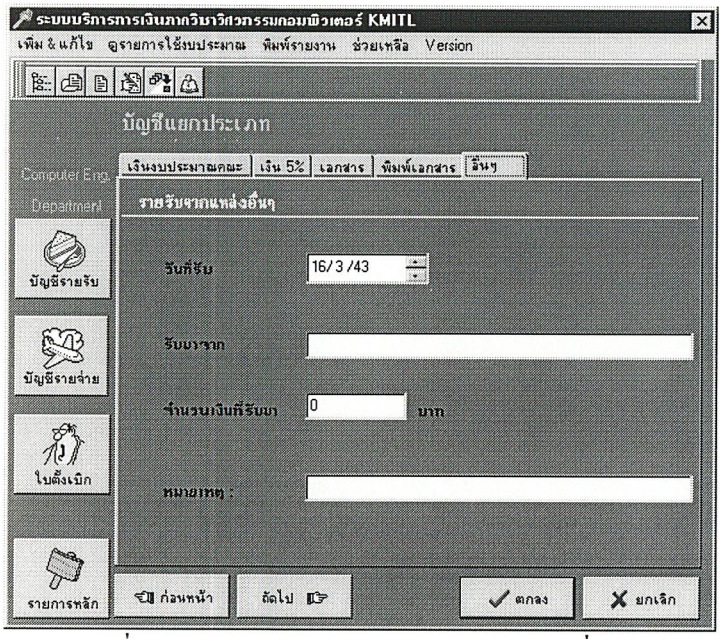
รูปที่ 3.15 รูปแสดงหน้าจอรายรับเงินค่าปริ้นเตอร์

3.4.3.4 รายรับเงินค่าขายเอกสาร เป็นหน้าจอสำหรับกรอกรายได้ของเงินรายได้ที่มาจากการขายเอกสารประกอบการสอนของภาควิชาวิศวกรรมคอมพิวเตอร์ มีลักษณะดังรูป



รูปที่ 3.16 รูปแสดงหน้าจอรายรับเงินค่าขายเอกสาร

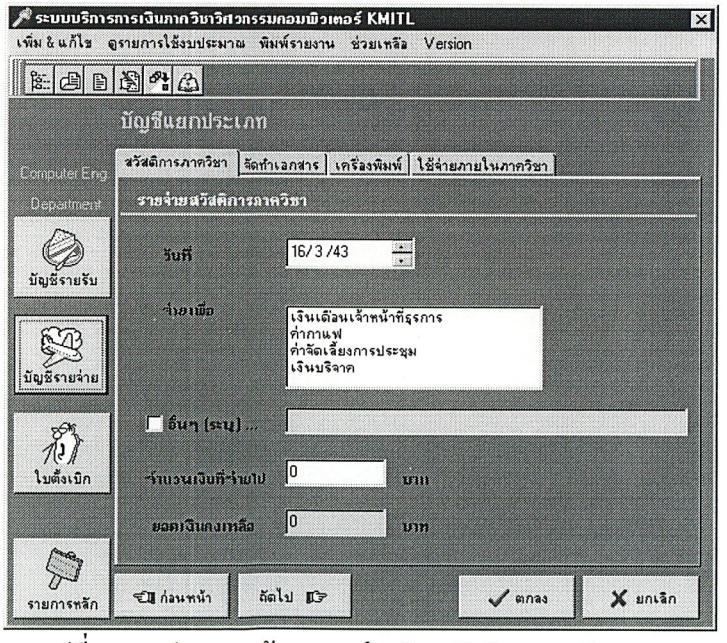
3.4.3.5 รายรับเงินจากรายได้อื่นๆ เป็นหน้าจอสำหรับกรอกรายได้ของเงินรายได้ที่มาจากแหล่งอื่นๆนอกจากที่กล่าวมาแล้ว เช่น รายได้จากการจัดสัมมนา เป็นต้นมีลักษณะดังรูป



รูปที่ 3.17 รูปแสดงหน้าจอรายรับเงินจากรายได้อื่นๆ

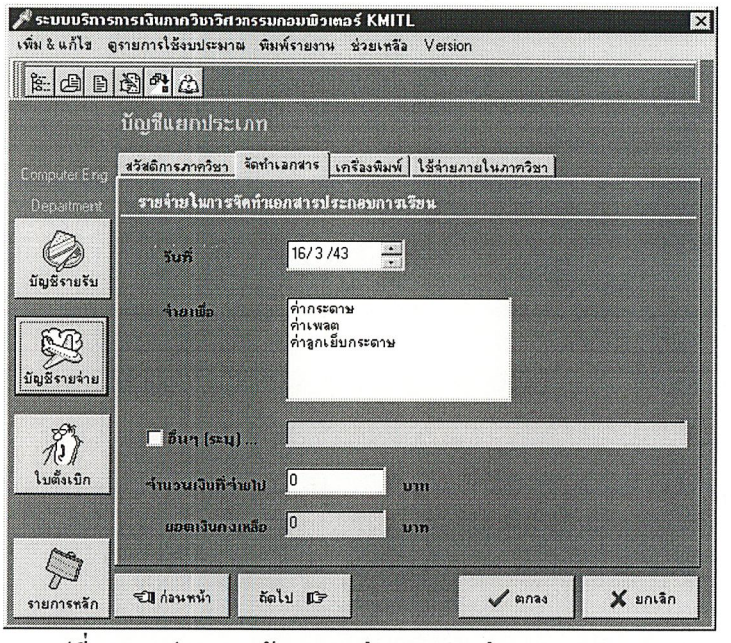
3.4.4 หน้าจอรายจ่าย หน้าจอรายรับเป็นหน้าจอสำหรับกรอกข้อมูลรายจ่ายประเภทต่างๆของภาควิชาวิศวกรรมคอมพิวเตอร์ ซึ่งประกอบไปด้วยรายการต่างๆดังนี้

3.4.4.1 รายจ่ายสวัสดิการภาควิชา เป็นหน้าจอสำหรับกรอกรายจ่ายจำพวกสวัสดิการของพนักงานภายในภาควิชาวิศวกรรมคอมพิวเตอร์ มีลักษณะดังรูป



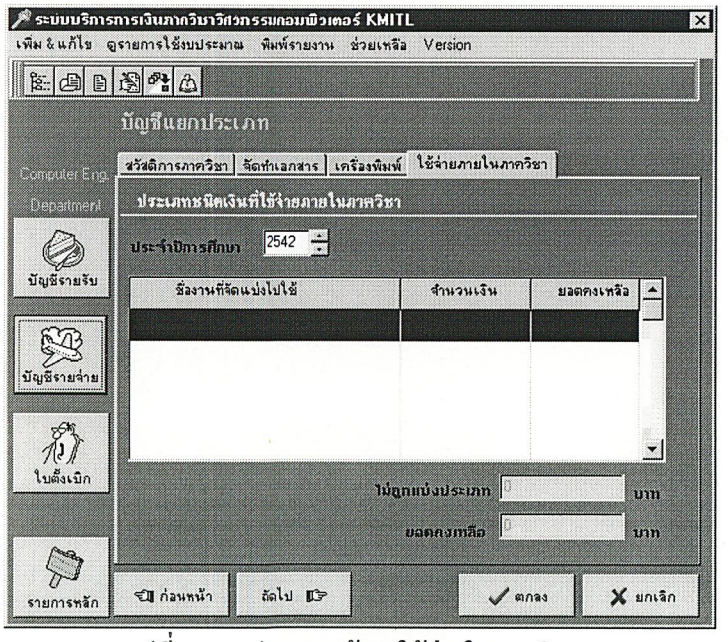
รูปที่ 3.18 รูปแสดงหน้าจอรายจ่ายเงินสวัสดิการภาควิชา

3.4.4.2 รายจ่ายจากเงินค่าขายเอกสาร เป็นหน้าจอสำหรับกรอกรายจ่ายที่นำเงินจากรายได้ที่มาจากการขายเอกสารประกอบการสอนของภาควิชาวิศวกรรมคอมพิวเตอร์ ไปใช้ในการจัดซื้อ มีลักษณะดังรูป



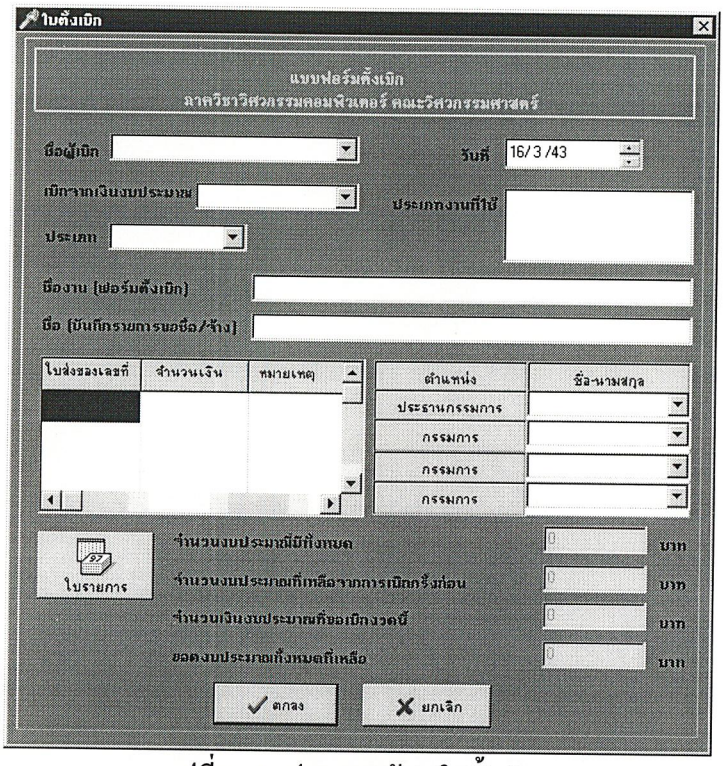
รูปที่ 3.19 รูปแสดงหน้าจอรายจ่ายจากเงินค่าขายเอกสาร

3.4.4.3 ใช้จ่ายในภาควิชา เป็นหน้าจอที่แสดงจำนวนเงินงบประมาณที่ถูกจัดสรรไว้ภายในปีงบประมาณนั้นๆสำหรับใช้ในการตั้งเบิกเพื่อนำไปใช้ในกิจกรรมต่างๆ มีลักษณะดังรูป



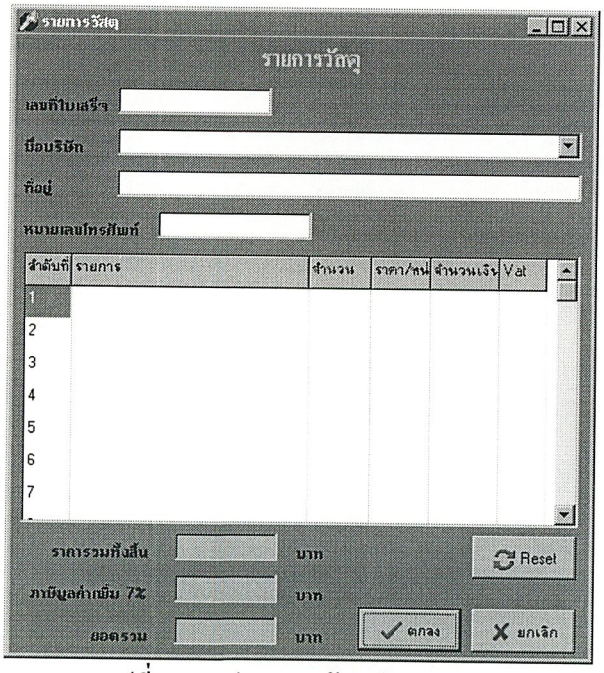
รูปที่ 3.20 รูปแสดงหน้าจอใช้จ่ายในภาควิชา

3.4.4.4 หน้าจอใบตั้งเบิก เป็นหน้าจอสำหรับใช้กรอกข้อมูลในใบตั้งเบิกสำหรับเบิกงบประมาณไปใช้ในกิจกรรมต่างๆซึ่งมีลักษณะดังรูป



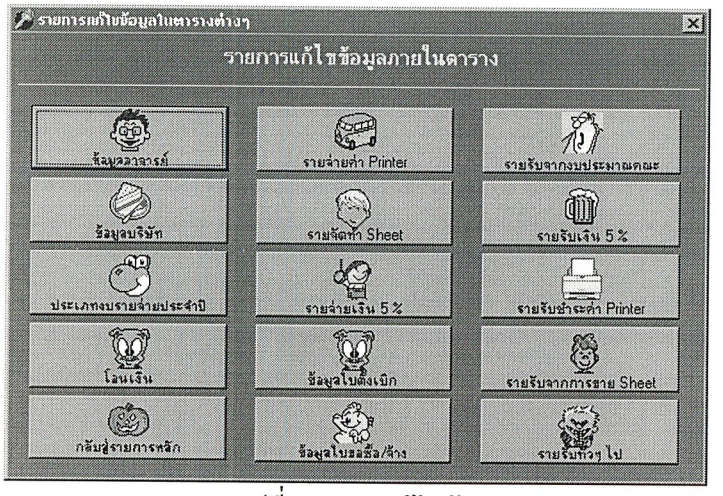
รูปที่ 3.21 รูปแสดงหน้าจอใบตั้งเบิก

3.4.4.5 หน้าจอใบรายการ เป็นหน้าจอสำหรับใช้กรอกข้อมูลในใบรายการสิ่งของที่จะทำการจัดซื้อสำหรับใบตั้งเบิก ซึ่งมีลักษณะดังรูป



รูปที่ 3.22 รูปแสดงหน้าจอใบรายการ

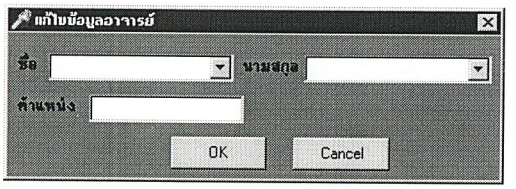
3.4.5 หน้าจอแก้ไขข้อมูล เมื่อมีการใส่ข้อมูลต่างๆเข้าไปภายในโปรแกรมก็ต้องมีการแก้ไขข้อมูลนั้นได้เช่นกัน ดังนั้นจึงต้องมีหน้าจอสำหรับให้ผู้ใช้ทำการแก้ไขข้อมูลต่างๆตามที่ต้องการ โดยสามารถเลือกได้ว่าจะแก้ไขข้อมูลในส่วนใดจากเมนูดังรูป



รูปที่ 3.23 เมนูแก้ไขข้อมูล

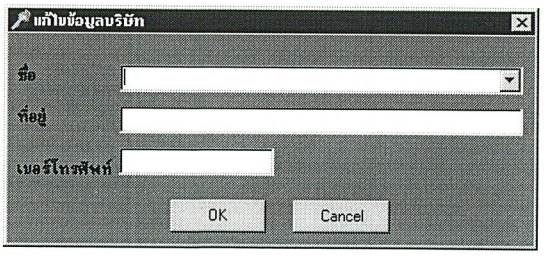
จากเมนูจะแบ่งเป็นการแก้ไขข้อมูลประเภทต่างๆดังนี้

3.4.5.1 หน้าจอแก้ไขข้อมูลอาจารย์ ใช้สำหรับแก้ไขข้อมูลเกี่ยวกับอาจารย์ภายในภาควิชาวิศวกรรมคอมพิวเตอร์ โดยจะมีฟิลด์ที่สามารถแก้ไขได้คือ ชื่อ-นามสกุล และตำแหน่ง ดังรูป



รูปที่ 3.24 รูปแสดงหน้าจอแก้ไขข้อมูลอาจารย์

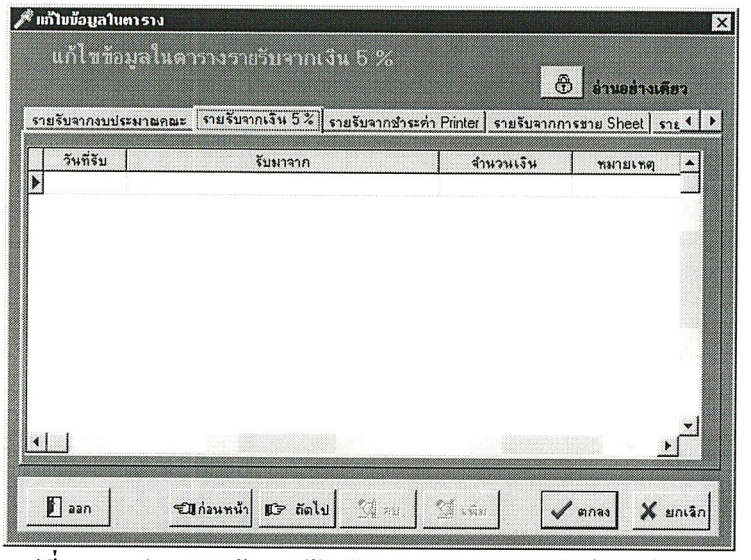
3.4.5.2 หน้าจอแก้ไขข้อมูลบริษัท ใช้สำหรับแก้ไขข้อมูลเกี่ยวกับบริษัทต่างๆที่ได้ทำการจัดซื้อสิ่งของในใบรายการสั่งซื้อ โดยจะมีฟิลด์ที่สามารถแก้ไขได้คือ ชื่อบริษัท-ที่อยู่ และเบอร์โทรศัพท์ ดังรูป



รูปที่ 3.25 รูปแสดงหน้าจอแก้ไขข้อมูลบริษัท

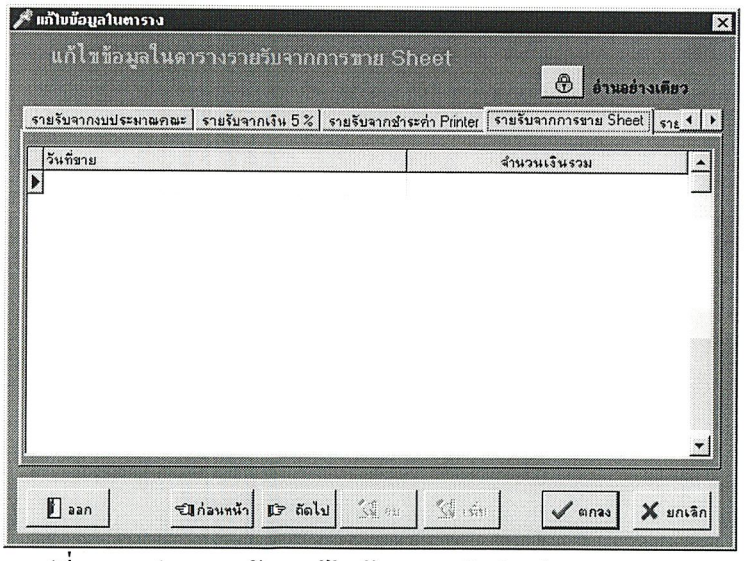
3.4.5.3 หน้าจอแก้ไขข้อมูลรายรับ ใช้สำหรับแก้ไขข้อมูลเกี่ยวกับรายรับต่างๆของภาควิชาวิศวกรรมคอมพิวเตอร์ โดยสามารถแบ่งย่อยออกได้เป็น

3.4.5.3.1 แก้ไขรายรับเงิน 5 เปอร์เซ็นต์ ใช้สำหรับแก้ไขข้อมูลรายรับเงิน 5 เปอร์เซ็นต์ ซึ่งมีฟิลด์ที่สามารถแก้ไขได้คือ วันที่รับ,รับมาจาก,จำนวนเงิน และหมายเหตุ ดังรูป



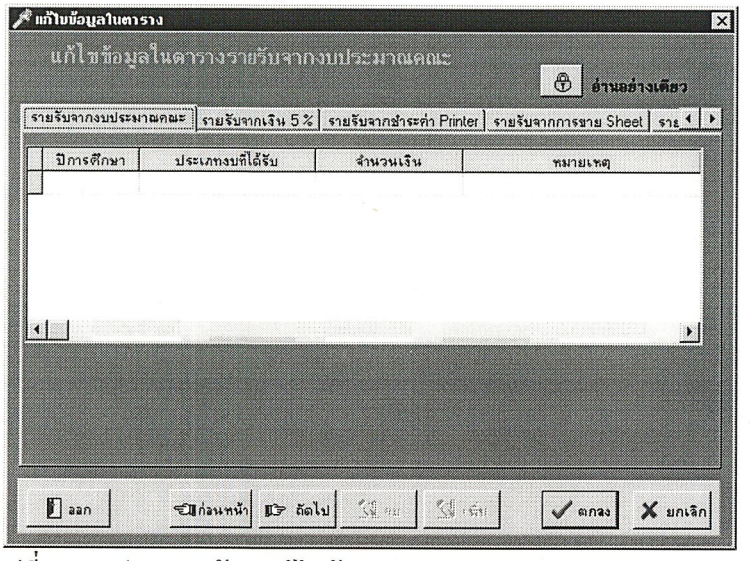
รูปที่ 3.26 รูปแสดงหน้าจอแก้ไขข้อมูลรายรับเงิน 5 เปอร์เซ็นต์

3.4.5.3.2 แก้ไขรายรับเงินค่าขายเอกสาร ใช้สำหรับแก้ไขข้อมูลรายรับเงินที่ได้จากการขายเอกสาร ซึ่งมีฟิลด์ที่สามารถแก้ไขได้คือ วันที่รับ และจำนวนเงินดังรูป



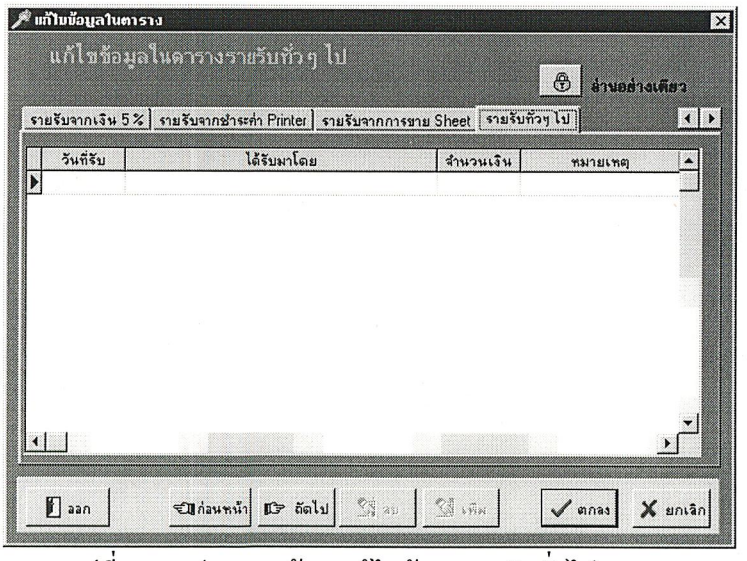
รูปที่ 3.27 รูปแสดงหน้าจอแก้ไขข้อมูลรายรับเงินค่าขายเอกสาร

3.4.5.3.3 แก้ไขรายรับเงินงบประมาณคณะ ใช้สำหรับแก้ไขข้อมูลรายรับเงินที่ได้จากงบประมาณคณะวิศวกรรมศาสตร์ ซึ่งมีฟิลด์ที่สามารถแก้ไขได้ดังรูป



รูปที่ 3.28 รูปแสดงหน้าจอแก้ไขข้อมูลรายรับเงินงบประมาณคณะ

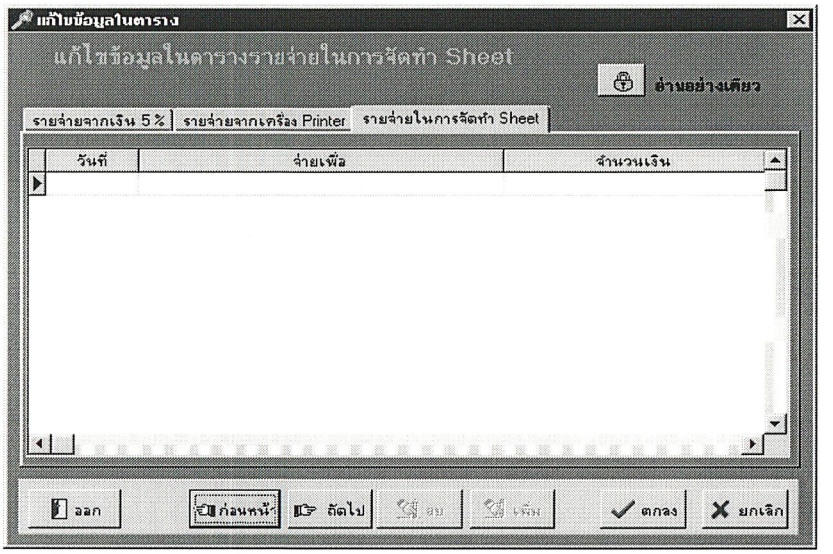
3.4.5.3.4 แก้ไขรายรับทั่วไป ใช้สำหรับแก้ไขข้อมูลรายรับทั่วๆไป ซึ่งมีฟิลด์ที่สามารถแก้ไขได้ดังรูป



รูปที่ 3.29 รูปแสดงหน้าจอแก้ไขข้อมูลรายรับทั่วไป

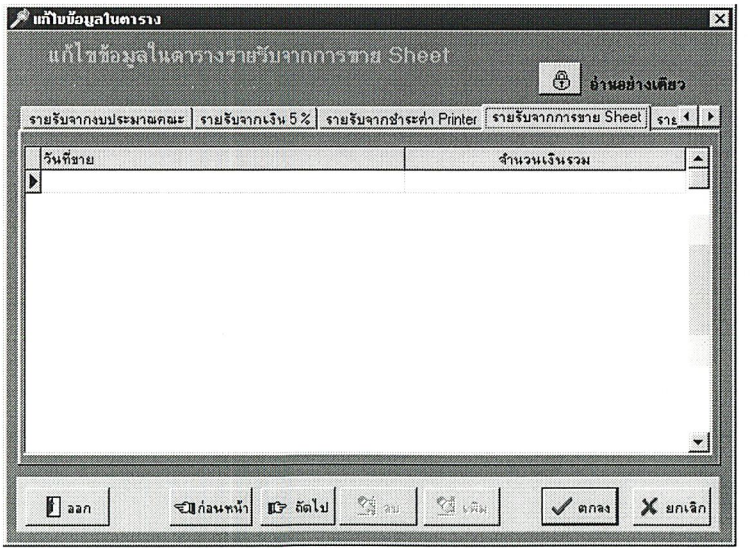
3.4.5.4 หน้าจอแก้ไขข้อมูลรายจ่าย ใช้สำหรับแก้ไขข้อมูลเกี่ยวกับรายจ่ายต่างๆของภาควิชาวิศวกรรมคอมพิวเตอร์ โดยสามารถแบ่งย่อยออกได้เป็น

3.4.5.4.1 แก้ไขรายจ่ายเงิน 5 เปอร์เซ็นต์ ใช้สำหรับแก้ไขข้อมูลรายจ่ายที่นำเงินจากรายรับเงิน 5 เปอร์เซ็นต์มาใช้ ซึ่งมีฟิลด์ที่สามารถแก้ไขได้ ดังรูป



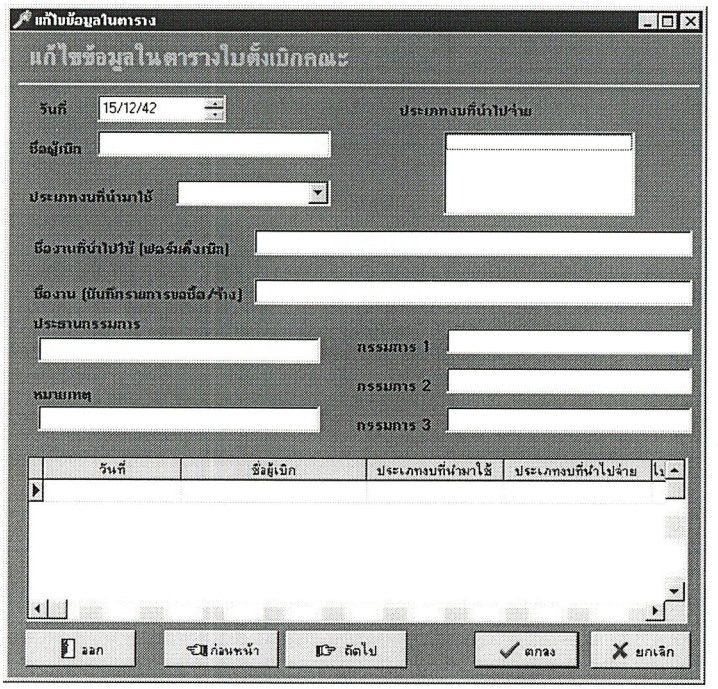
รูปที่ 3.30 รูปแสดงหน้าจอแก้ไขข้อมูลรายจ่ายเงิน 5 เปอร์เซ็นต์

3.4.5.4.2 แก้ไขรายจ่ายเงินค่าขายเอกสาร ใช้สำหรับแก้ไขข้อมูลรายจ่ายซึ่งนำเงินจากรายรับเงินที่ได้จากการขายเอกสารมาใช้ ซึ่งมีฟิลด์ที่สามารถแก้ไขได้ดังรูป



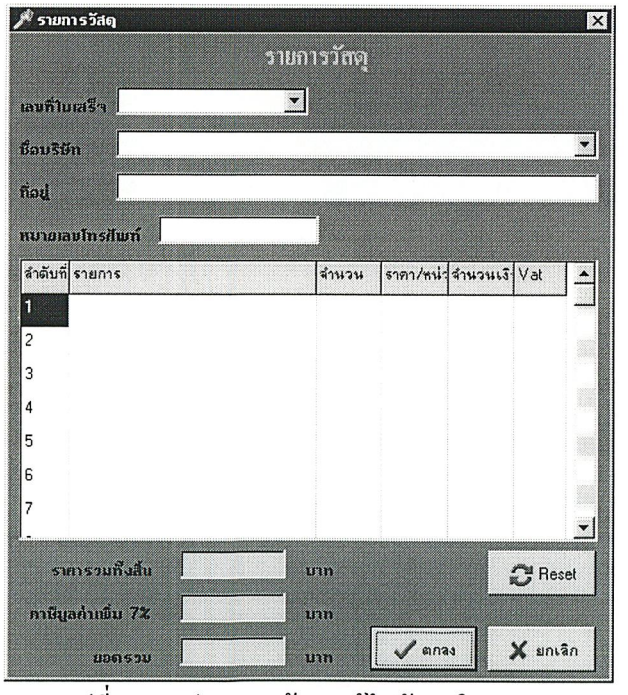
รูปที่ 3.31 รูปแสดงหน้าจอแก้ไขข้อมูลรายจ่ายเงินค่าขายเอกสาร

3.4.5.4.3 แก้ไขข้อมูลใบตั้งเบิก ใช้สำหรับแก้ไขข้อมูลในใบตั้งเบิกซึ่งใช้สำหรับเบิกเงินงบประมาณของภาควิชาไปใช้ ซึ่งมีฟิลด์ที่สามารถแก้ไขได้ดังรูป



รูปที่ 3.32 รูปแสดงหน้าจอแก้ไขข้อมูลใบตั้งเบิก

3.4.5.4.4 แก้ไขใบรายการ ใช้สำหรับแก้ไขข้อมูลในใบรายการสิ่งของที่จะทำการตั้งเบิกงบประมาณสำหรับจัดซื้อ ซึ่งมีฟิลด์ที่สามารถแก้ไขได้ดังรูป



รูปที่ 3.33 รูปแสดงหน้าจอแก้ไขข้อมูลใบรายการ

3.5 การออกแบบรายงานต่างๆที่ใช้ในโปรแกรม

3.5.1 รายงานรายรับเงิน 5 เปอร์เซ็น เป็นรายงานสำหรับใช้แสดงข้อมูลรายรับจากเงิน 5 เปอร์เซ็นของรายได้จากการสอนของอาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ โดยจะมีการแสดงวันที่รับเงินรับเงินเข้ามาเป็นวันที่เท่าใด รับมาจากอาจารย์ท่านใด เป็นจำนวนเงินเท่าใด และหมายเหตุ

รายรับเงิน 5 เปอร์เซ็นต์

ว/ด/ป	รับมาจาก	จำนวนเงิน	หมายเหตุ
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx	
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx	
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx	
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx	
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx	
	รวมเป็นเงิน	xx,xxx บาท	

รูปที่ 3.34 แสดงรายงานรายรับเงิน 5 เปอร์เซ็นต์

3.5.2 รายงานรายรับเงินค่าปริ้นเตอร์ เป็นรายงานสำหรับใช้แสดงข้อมูลรายรับจากเงินค่าปริ้นเตอร์ตามจำนวนการใช้งานของผู้ใช้ปริ้นเอต์ในภาควิชาวิศวกรรมคอมพิวเตอร์ โดยจะมีการแสดงวันที่รับเงินรับเงินเข้ามาเป็นวันที่เท่าใด รับมาจากใคร เป็นจำนวนเงินเท่าใด

รายรับเงินค่าปริ้นเตอร์

ว/ด/ป	รับมาจาก	จำนวนเงิน
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
	รวมเป็นเงิน	xx,xxx บาท

รูปที่ 3.35 แสดงรายงานรายรับเงินค่าปริ้นเตอร์

3.5.3 รายงานรายรับเงินค่าชีต เป็นรายงานสำหรับใช้แสดงข้อมูลรายรับจากเงินรายได้จากการขายชีตในภาควิชาวิชาวิศวกรรมคอมพิวเตอร์ โดยจะมีการแสดงวันที่รับเงินรับเงินเข้ามาเป็นวันที่เท่าใด รับมา เป็นจำนวนเงินเท่าใด

รายรับเงินค่าชีต

ว/ด/ป	จำนวนเงิน
xx/xx/xx	xx,xxx
xx/xx/xx	xx,xxx
xx/xx/xx	xx,xxx
xx/xx/xx	xx,xxx
xx/xx/xx	xx,xxx
รวมเป็นเงิน	xx,xxx บาท

รูปที่ 3.36 แสดงรายรับเงินค่าชีต

3.5.4 รายงานรายจ่ายจากเงิน 5 เปอร์เซ็น เป็นรายงานสำหรับใช้แสดงข้อมูลรายจ่ายที่นำเงินจากเงินรายได้เงิน 5 เปอร์เซ็นต์มาใช้ โดยจะมีการแสดงวันที่รับเงินรับเงินเข้ามาเป็นวันที่เท่าใด ใช้เพื่ออะไร เป็นจำนวนเงินเท่าใด

รายจ่ายเงิน 5 เปอร์เซ็น

ว/ด/ป	ใช้เพื่อ	จำนวนเงิน
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
รวมเป็นเงิน		xx,xxx บาท

รูปที่ 3.37 แสดงรายงานรายจ่ายเงิน 5 เปอร์เซ็นต์

3.5.5 รายงานรายจ่ายเงินค่าปริ้นเตอร์ เป็นรายงานสำหรับใช้แสดงข้อมูลรายจ่ายที่นำเงินจากเงินรายได้ค่าปริ้นเตอร์มาใช้ โดยจะมีการแสดงวันที่รับเงินรับเงินเข้ามาเป็นวันที่เท่าใด ใช้เพื่ออะไร เป็นจำนวนเงินเท่าใด

รายจ่ายเงินค่าปริ้นเตอร์

ว/ด/ป	ใช้เพื่อ	จำนวนเงิน
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
	รวมเป็นเงิน	xx,xxx บาท

รูปที่ 3.38 แสดงรายงานรายจ่ายเงินค่าปริ้นเตอร์

3.5.6 รายงานรายจ่ายเงินค่าชีต เป็นรายงานสำหรับใช้แสดงข้อมูลรายจ่ายที่นำเงินจากเงินรายได้จากการขายเอกสารมาใช้ โดยจะมีการแสดงวันที่รับเงินรับเงินเข้ามาเป็นวันที่เท่าใด ใช้เพื่ออะไร เป็นจำนวนเงินเท่าใด

รายจ่ายเงินค่าชีต

ว/ด/ป	ใช้เพื่อ	จำนวนเงิน
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
xx/xx/xx	xxxxxxx xxxxxxx	xx,xxx
	รวมเป็นเงิน	xx,xxx บาท

รูปที่ 3.39 แสดงรายงานรายจ่ายค่าชีต

3.5.7 รายงานรายจ่ายจากเงินงบประมาณคณะวิชา

เงินงบประมาณคณะวิชาจะถูกนำไปใช้ได้นั้นต้องมีการเขียนใบตั้งเบิกขึ้นมาเพื่อขอเบิกเงินงบประมาณไปใช้ โดยภายในใบตั้งเบิกจะมีการเขียนรายการต่างๆที่ต้องการนำเงินไปใช้ รวมทั้งรายชื่อคณะกรรมการ ที่เป็นผู้พิจารณาการเบิกจ่ายเงินงบประมาณ ซึ่งหลังจากที่ใส่ข้อมูลใบตั้งเบิกลงในโปรแกรมแล้ว โปรแกรมจะทำการสร้างรายงานขึ้นมา 4 ฉบับ ประกอบไปด้วย

3.5.7.1 ใบส่งของ เป็นรายงานแสดงรายการสิ่งของที่บริษัทผู้ขายส่งมาให้กับภาควิชาวิศวกรรมคอมพิวเตอร์ ซึ่งข้อมูลในรายงานประกอบไปด้วย รายการสิ่งของ จำนวนเงิน รายชื่อเจ้าหน้าพัสดุและเจ้าหน้าที่การเงินผู้รับของดังรูป

แบบฟอร์มใบส่งของ

ภาควิชา/แผนก วิศวกรรมคอมพิวเตอร์ ผู้ส่ง ธนา หงษ์ยุวรรณ วันที่

ชื่อบริษัท ห้างหุ้นส่วน ร้านค้า	ใบส่งของ	จำนวนเงิน	หมายเหตุ
ซีเกทเทคโนโลยี	001/2543	8375	

เจ้าหน้าที่พัสดุผู้รับ เจ้าหน้าที่การเงินผู้รับ

วันที่ วันที่

รูปที่ 3.40 รูปแสดงรายงานใบส่งของ

3.5.7.2 รายงานใบบันทึกรายการขอซื้อขอจ้าง เป็นรายงานสำหรับบันทึกรายการสิ่งของต่างๆที่ภาควิชาทำการจัดซื้อซึ่งข้อมูลในรายงานประกอบไปด้วย รายการสิ่งของ ,จำนวนที่สั่งซื้อ, จำนวนเงินต่อหน่วย และรายชื่อคณะกรรมการดังรูป

บันทึกรายการขอซื้อ/ขอจ้าง

ส่วนราชการ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ที่ วันที่

เรื่อง ขออนุมัติ ซื้อ / จ้าง ประจำปีงบประมาณ โดยวิธีตกลงราคา

เรียน คณบดี คณะวิศวกรรมศาสตร์

ด้วย ธนา หงษ์สุวรรณ ตำแหน่ง

ขอรายงานเสนอเพื่อขอจัดซื้อ/จ้าง วัสดุ

ให้แก่ เนื่องจากมีความจำเป็น Computer project

ตามรายการและราคาดังนี้

รายการและขนาด	จำนวน	ราคา	ราคาซื้อ/จ้าง	เคยซื้อ/จ้าง
Hard Disk	1	5000		
Floppy Disk	5	500		

รวม 2 รายการ เป็นเงิน 7500 บาท ภาษีมูลค่าเพิ่ม 525 บาท รวมเป็นเงินทั้งสิ้น 8025 บาท

กำหนดเวลาที่ต้องการใช้ วัน ตามรายละเอียดข้างบนนี้จะดำเนินการจัดซื้อ/จัดจ้าง โดยวิธีตกลงราคา ตามข้อ 19 แห่งระเบียบสำนักนายกรัฐมนตรีว่าด้วยการพัสดุ พศ.2535 (แก้ไขฉบับที่ 2 พศ. 2538) โดยขอเบิกจ่ายจากเงิน งบประมาณ พร้อมขอแต่งตั้ง คณะกรรมการตรวจรับพัสดุ

1. นางสาวนวพร วรรณวิมลศรี ประธานกรรมการ
2. นายชม ลิ้มชำนาญ กรรมการ
3. กรรมการ

จึงเรียนมาเพื่อโปรดพิจารณาอนุมัติ ให้ซื้อ/จ้าง ตามวิธีตกลงราคา พร้อมทั้งแต่งตั้งกรรมการตรวจรับพัสดุข้างต้นต่อไปด้วย

ลงชื่อ จำหน้าที่พัสดุหรือผู้รายงาน

( )

ลงชื่อ หัวหน้าพัสดุ

( )

อนุมัติให้ซื้อ/จ้าง โดยวิธีตกลงราคา และอนุมัติแต่งตั้งกรรมการดังกล่าวข้างต้น

ลงชื่อ ผู้อนุมัติ

( )

คณบดี คณะวิศวกรรมศาสตร์

รูปที่ 3.41 รูปแสดงรายงานใบบันทึกรายการขอซื้อขอจ้าง

3.5.7.3 รายงานใบเบิกวัสดุ เป็นรายงานสำหรับบันทึกรายการสิ่งของต่างๆที่ทำการเบิกไปเพื่อทำกิจกรรมบางประเภทเช่น ใช้ในการทำโปรเจ็ค หรือใช้ในการทำแลปเป็นต้น ข้อมูลในรายงานประกอบไปด้วย รายการสิ่งของ ,จำนวนที่สั่งซื้อ,จำนวนเงินต่อหน่วย และรายชื่อผู้เบิกของดังรูป

ใบเบิกวัสดุ

เลขที่ 1

วันที่

รายจ่ายของตามรายการต่อไปนี้ให้แก่ ธนา หงษ์สุวรรณ

สำหรับงาน Computer project ภาควิชา วิศวกรรมคอมพิวเตอร์

รายการ	จำนวน	หน่วยละ	จำนวนเงิน
Floppy Disk	5	500	2500
Hard Disk	1	5000	5000

ผู้เบิก ผู้จ่าย

อนุญาติ ผู้รับของ

วันที่

รูปที่ 3.42 รูปแสดงรายงานใบเบิกวัสดุ

3.5.7.4 รายงานใบสำคัญเพื่อคืนเงินทดลองจ่าย เป็นรายงานสำหรับบันทึกรายการใบเสร็จรับเงินของบริษัทต่างๆเพื่อใช้สำหรับเบิกเงินในการจัดซื้อ ข้อมูลในรายงานประกอบไปด้วย รายชื่อบริษัท , เลขที่ใบเสร็จ,จำนวนเงิน และรายชื่อผู้รับของของดังรูป

แบบฟอร์มส่งใบสำคัญเพื่อคืนเงินยืมทดลองจ่าย

ชื่อผู้ยืม ซีเกทเทคโนโลยี ภาควิชา วิศวกรรมคอมพิวเตอร์

เงินยืมทดลองจ่าย วันที่ เป็นเงิน 8375 บาท

ชื่อบริษัท,ห้างหุ้นส่วน,ร้านค้า	ใบเสร็จรับเงินเล่มที่	จำนวนเงิน(บาท)	หมายเหตุ
ซีเกทเทคโนโลยี	001/2543	8375	

คืนเงินสด บาท

รวมเป็นเงิน บาท

เจ้าหน้าที่พัสดุผู้รับ เจ้าหน้าที่การเงินผู้รับ

วันที่ วันที่

ใบรับใบสำคัญเพื่อคืนเงินยืมทดลองจ่าย

ใบสำคัญของอาจารย์ รวมเป็นเงิน บาท

เจ้าหน้าที่พัสดุผู้รับ วันที่

รูปที่ 3.43 รูปแสดงรายงานใบสำคัญเพื่อคืนเงินทดลองจ่าย

บทที่ 4
บทสรุปและวิจารณ์

4.1 ข้อได้เปรียบของระบบเดิม

4.1.1 ความยืดหยุ่นสูง เนื่องจากงานเดิมใช้มนุษย์ทำงานทั้งหมด ดังนั้นจึงสามารถใช้ความสามารถด้าน การตัดสินใจของมนุษย์ในการปรับให้เข้ากับสถานการณ์ที่ต่างๆ กันได้และมนุษย์ยังสามารถคิดแก้ปัญหาที่ซับซ้อนและไม่มีกฎเกณฑ์ตายตัวได้ดี

4.2 ข้อเสียเปรียบของระบบเดิม

4.2.1 การยึดติดกับความสามารถของบุคคลมากทำให้การทำงานไม่มีมาตราฐานเดียวกันจะขึ้นอยู่กับความสามารถของบุคคลที่ปฏิบัติงานมากกว่า อาทิ เช่น ความสามารถในการตัดสินใจที่ถูกต้องรวดเร็ว, ความละเอียดรอบของในการทำงาน, ความรวดเร็วในการรวบรวมข้อมูลเพื่อนำเสนอ ฯลฯ

4.2.2 ข้อจำกัดของมนุษย์ อาทิ เช่น ต้องการการพักผ่อน, มนุษย์มีช่วงเวลาที่สามารถทำงานติดต่อกันได้จำกัดข้อจำกัดของการใช้ความจำในสิ่งต่างๆ ที่มีปริมาณมาก และหลากหลาย ความเร็วในการทำงาน การรักษาประสิทธิภาพในการทำงานให้คงที่ตลอดการทำงานเป็นเวลานาน

4.2.3 ข้อจำกัดของรูปแบบการทำงาน เนื่องจากในการทำเรื่องใบตั้งเบิกรายการที่ได้นำไปใช้จริงและรายการที่จะต้องเขียนในใบจัดซื้อจัดจ้างนั้นไม่ตรงกัน และรายการสินค้าและบิลที่นำไปใช้นั้นจะต้องมีการแก้ไขซึ่งเป็นกระบวนการที่ค่อนข้างยุ่งยาก ทำให้เกิดความสับสนได้ง่ายหากขาดการจัดการวางแผนการจัดเก็บข้อมูลที่ไม่ดีเพียงพอ อาจทำให้นำเสนอต่อหน่วยงานผิดพลาดได้ ซึ่งจะส่งผลกระทบโดยตรงแก่ภาควิชา

4.2.4 ระบบการคำนวณและการจัดเก็บข้อมูลค่อนข้างมาก เสียเวลาในการเรียกค้นหาข้อมูลที่ต้องการ และก่อให้เกิดความผิดพลาดได้ง่าย

4.3 ข้อดีของระบบใหม่

4.3.1 ช่วยแบ่งเบาภาระหน้าที่เจ้าหน้าที่ได้มากขึ้น สะดวกต่อการใช้งานในการจัดเก็บรายการรายรับ – รายจ่ายประจำวัน ลดความผิดพลาดเนื่องมาจากการงานเอกสาร

4.3.2 มีการจัดเก็บข้อมูลได้อย่างมีประสิทธิภาพ

4.3.3 สนับสนุนระบบสารสนเทศภาควิชา สามารถใช้ฐานข้อมูลเดียวกันได้ ถ้าต้องการพัฒนาระบบงานใหม่ๆ เพื่อใช้ในภาควิชา

4.3.4 การจัดทำเอกสาร แบบฟอร์ม เช่น ใบตั้งเบิก, ใบจัดซื้อ/จัดจ้าง, ใบส่งของ ฯลฯ ทำได้อย่างรวดเร็วและเป็นอัตโนมัติ อำนวยความสะดวกให้แก่เจ้าหน้าที่ซึ่งไม่ต้องเสียเวลาในค้นหาข้อมูลรายการต่างๆ ที่ได้ขอทำเรื่องขอเบิก

4.3.5 เป็นการนำเอาระบบเทคโนโลยีทางด้านสารสนเทศมาใช้ได้อย่างมีประสิทธิภาพ

4.4 ข้อเสียของระบบใหม่

4.4.1 เนื่องจากช่วงเวลาที่ใช้ในการทดสอบมีระยะสั้นอาจไม่เพียงพอในการที่จะพบข้อผิดพลาดในระบบ ซึ่งอาจจะมีผลทำให้งานบ้างอย่างผิดพลาดได้ ซึ่งได้พยายามแก้ไขให้เหลือน้อยที่สุดเท่าที่เวลาจะอำนวย

4.4.2 จำเป็นต้องใช้ผู้มีความรู้ในด้านฐานข้อมูลเพื่อเป็นผู้ดูแลระบบฐานข้อมูล เนื่องจากเป็นส่วนสำคัญที่สุดในระบบงานนี้ การแก้ไขฐานข้อมูลในฐานข้อมูลจะทำโดยผู้ดูแลระบบฐานข้อมูลเท่านั้น ผู้ใช้ทั่วๆ ไปไม่ควรมีสิทธิในการเข้าขึ้นฐานข้อมูลในระดับสิทธิที่มีการแก้ไขข้อมูลได้

บรรณานุกรม

[1] C.J.Date , "An Introduction to Database System" , Addison-Wesley Publishing Company, 854 p. , 1990.

[2] Ian Sommerville , "Software Engineering" , fifth edition , Addison-Wesley Publishing Company, 742 p. , 1995.