



**โครงการวิจัยขยายผลการพัฒนาระบบสนับสนุนการดำเนินงานปรับเปลี่ยน
ระบบการบริหารการเงิน**

เล่มที่ 12/ 13

ระบบบริหารผู้ใช้งานระบบ : คู่มือการใช้งานระบบ

โดย

ผศ.ดร.เหรียญ บุญดีสกุลโชค และคณะ

โครงการวิจัยเลขที่ 63G-IE-2545

ทุนงบประมาณแผ่นดิน ปี 2545

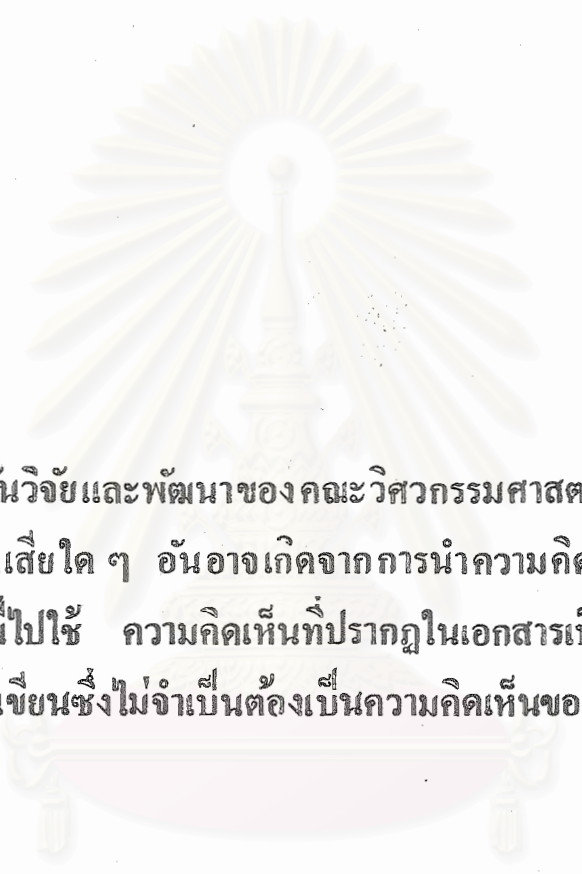
สถาบันวิจัยและพัฒนา
จุฬาลงกรณ์มหาวิทยาลัย

คณะวิศวกรรมศาสตร์

จุฬาลงกรณ์มหาวิทยาลัย

กรุงเทพฯ

พฤศจิกายน 2546



สถาบันวิจัยและพัฒนาของ คณะวิศวกรรมศาสตร์ไม่รับผิดชอบ
ต่อผลเสียใด ๆ อันอาจเกิดจากการนำความคิดเห็นในเอกสาร
ฉบับนี้ไปใช้ ความคิดเห็นที่ปรากฏในเอกสารเป็นความคิดเห็น
ของผู้เขียนซึ่งไม่จำเป็นต้องเป็นความคิดเห็นของสถาบันฯ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

โครงการวิจัยขยายผลการพัฒนาระบบสนับสนุนการดำเนินงานปรับเปลี่ยน
ระบบการบริหารการเงิน

เล่มที่ 12/ 13

ระบบบริหารผู้ใช้งานระบบ : คู่มือการใช้งานระบบ

โดย

เหรียญ บุญดีสกุลโชค D.Eng. (AIT) และคณะ

โครงการวิจัยเลขที่ 63G-IE-2545
ทุนงบประมาณแผ่นดิน ปี 2545

สถาบันวิจัยการ
จุฬาลงกรณ์มหาวิทยาลัย

คณะวิศวกรรมศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย
กรุงเทพฯ
พฤศจิกายน 2546

ภาคผนวก ก



ขั้นตอนในการ Implement



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

เลขหมู่ ^{จ.ร.} ๐๕15
เลขทะเบียน ๐11954
วัน,เดือน,ปี 22 มี.ช. 47 ^{๓.12}

ขั้นตอนในการ Implement

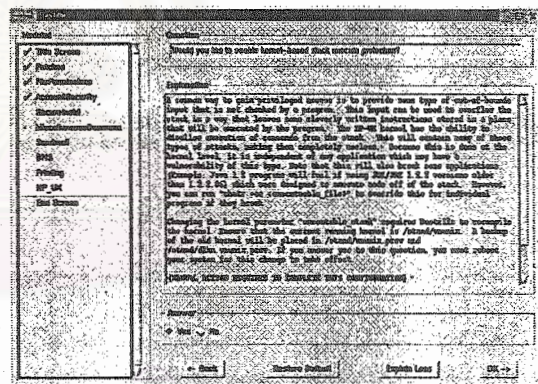
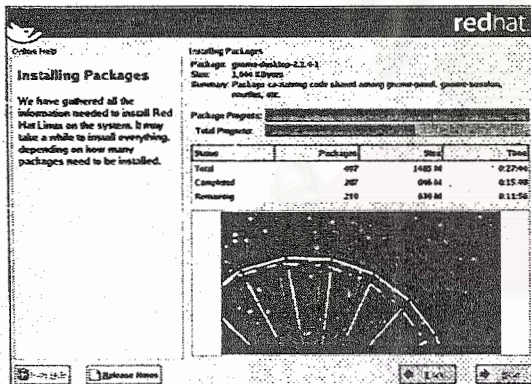
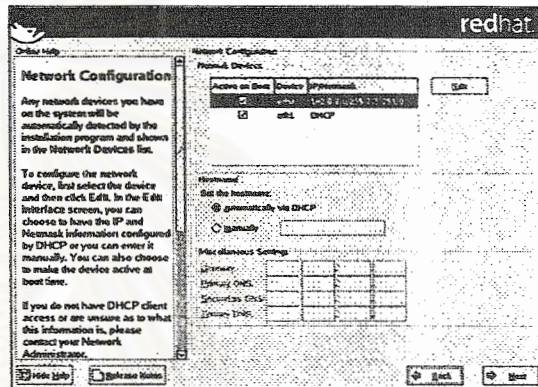
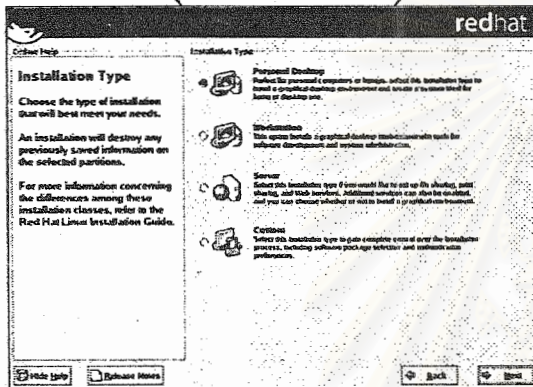
แบ่งออกเป็น 2 ลักษณะคือ

1. การทำงานที่เกี่ยวข้องกับเครื่องแม่ข่าย (Server) ที่จะทำหน้าที่เป็น Database Server และ Web Server
2. การทำงานที่เกี่ยวข้องกับเครื่องลูกข่าย (Client)

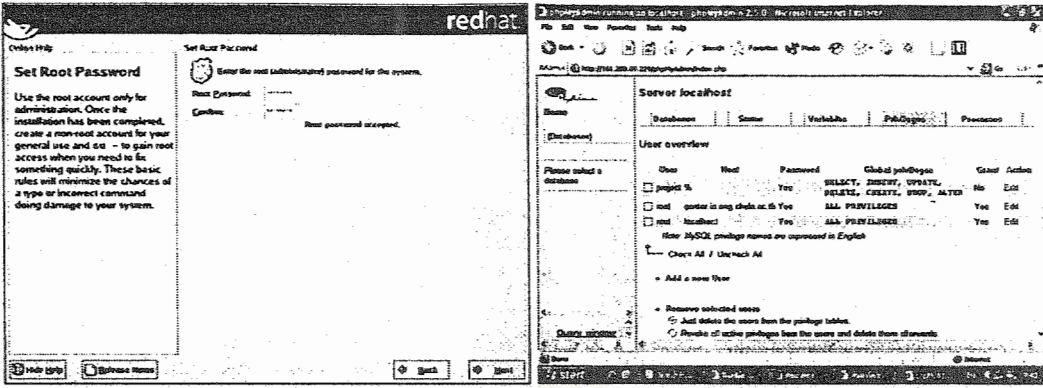
โดยมีรายละเอียดและขั้นตอนในการทำงานต่างๆ ดังต่อไปนี้

1. การทำงานที่เกี่ยวข้องกับเครื่องแม่ข่าย (Server)

- ได้ทำการลง OS Linux9 (Shrike) ตลอดจนค่า IP Address, Netmask, Gateway และติดตั้ง Firewall (Bastille Firewall)

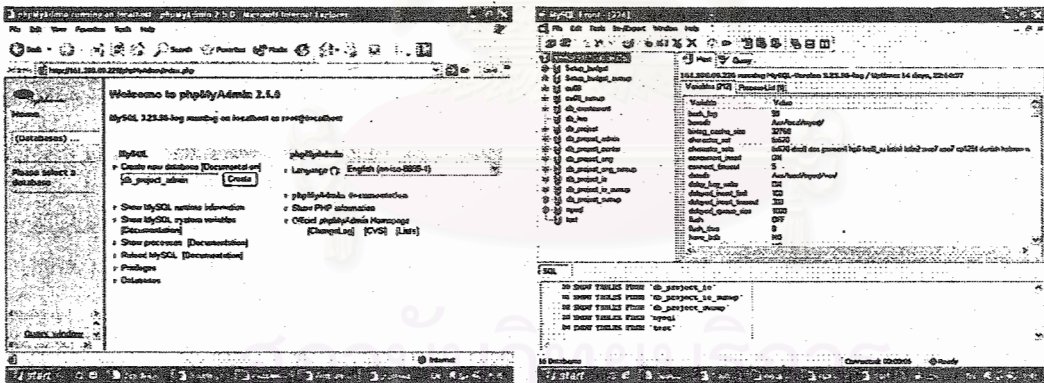


- ทำการลง ฐานข้อมูล (MySQL), Web Server (Apache), PHP, ตัวบริหารฐานข้อมูลบน Webbased Application (phpMyAdmin)
- ได้ทำการสร้าง User & Password ต่างๆ อันได้แก่
 - o User & Password ของ Linux (ทั้ง root และ username อื่นๆ ของ linux)
 - o User & Password & Privileges ของ MySQL (MySQL Username/ Password)



- นำโครงสร้างของฐานข้อมูลต้นแบบ (3 ฐานข้อมูล) ไปทำการสร้างฐานข้อมูลเพื่อรองรับข้อมูลในแต่ละโมดูลที่จะทำการกรอกต่อไป และ ฐานข้อมูล 3 ตัวได้แก่
 - o ฐานข้อมูลสำหรับการ Login เข้าตัว chulaup สำหรับ Admin Module (db_project_admin)
 - o ฐานข้อมูลสำหรับเก็บข้อมูลปัจจุบันของ Asset, Budgetary, KPI Module (db_project_md)
 - o ฐานข้อมูลสำหรับเก็บข้อมูลในลักษณะรวมภาพย่อยไปหากภาพใหญ่ สำหรับ Budgetary Module (db_project_md_sumup)

โดยผ่านทาง phpMyAdmin หรือถ้าติดตั้งโปรแกรมต่างๆ บน client เสร็จเรียบร้อยแล้ว สามารถใช้ MySQL-Front บน client เข้ามาสร้างฐานข้อมูลต้นแบบทั้งสามบนเครื่อง Server ได้



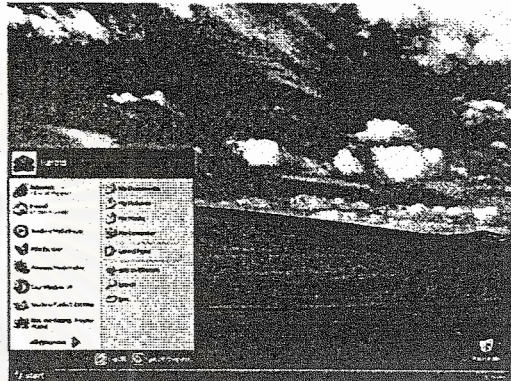
จุฬาลงกรณ์มหาวิทยาลัย

2. การทำงานที่เกี่ยวข้องกับเครื่องลูกข่าย (Client)

การติดตั้งเพื่อใช้งาน ChulaUp ในเครื่องลูกข่ายมีขั้นตอนในการลงโปรแกรมต่างๆ โดยแบ่งออกเป็นขั้นตอน โดยรวม และขั้นตอนในแต่ละโมดูลดังนี้

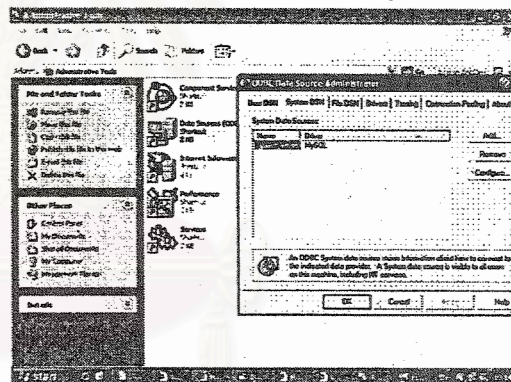
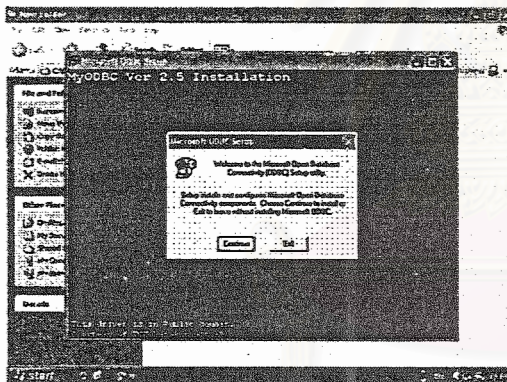
ขั้นตอนโดยรวม

- ทำการลง OS WindowsXP

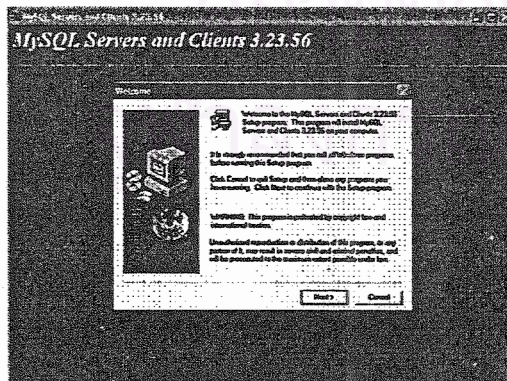


- ทำการลง MicroSoft OfficeXP

- ทำการลง myODBC (เป็น driver สำหรับเครื่องลูกข่ายที่จะทำการติดต่อกับฐานข้อมูล MySQL)



- ทำการลง ChulaUp Package



- ทำการลง MySQL for Windows ซึ่งเป็นคนละตัวที่ลงกับเครื่องแม่ข่าย เนื่องจาก มีข้อมูลบางลักษณะของ Budgetary และ Asset Module ที่ต้องการเก็บพักไว้ก่อนที่เครื่องลูกข่าย ก่อนที่จะไปอัปเดต (update) ข้อมูลบนฐานข้อมูลบนเครื่องแม่ข่ายต่อไป

- ทำการติดตั้งฐานข้อมูลต้นแบบ (project_local)
- เข้าโปรแกรม ChulaUp โดยจะต้องกรอก ค่าในการเชื่อมต่อกับฐานข้อมูลสำหรับการเข้า ChulaUp ในครั้งแรก เพื่อใช้สำหรับสร้าง DSN Configuration สำหรับ ฐานข้อมูลสำหรับการ Login อัน ได้แก่
 - IP Address ของ Database Server
 - ค่า Username ของฐานข้อมูลในการติดต่อกับฐานข้อมูล (MySQL)
 - ค่า Password ของ MySQL Username
 - ทำการป้อนค่า Password ของ Root ของ ChulaUp
- จากนั้นทำการ Loginเข้า ChulaUp โดยใช้ Username 'root' และ Password เป็นค่าที่ตั้งไว้ก่อนหน้านี้
- ทำการสร้าง Username & Password ที่จะป็นเจ้าของฐานข้อมูล ที่กำหนดผ่านทาง ChulaUP
- ทำการสร้าง DSN Configuration ต่างๆ สำหรับในการเชื่อมต่อ ไปยังฐานข้อมูลที่เก็บข้อมูลปัจจุบัน (db_project_md) ของ Asset + Document System, Budgetary, KPI Module อัน ได้แก่
 - DSN Detail
 - DSN Owner หรือการระบุผู้ที่จะมาเป็นเจ้าของฐานข้อมูลปัจจุบัน (db_project_md)
- ทำการ Logoff จาก root และทำการ Login ใหม่ในฐานะเจ้าของฐานข้อมูล และจะได้หน้าเมนู ขึ้นมาให้เลือกเพื่อการทำงานในแต่ละ โมดูลที่เลือกต่อไป

ขั้นตอนในแต่ละโมดูล

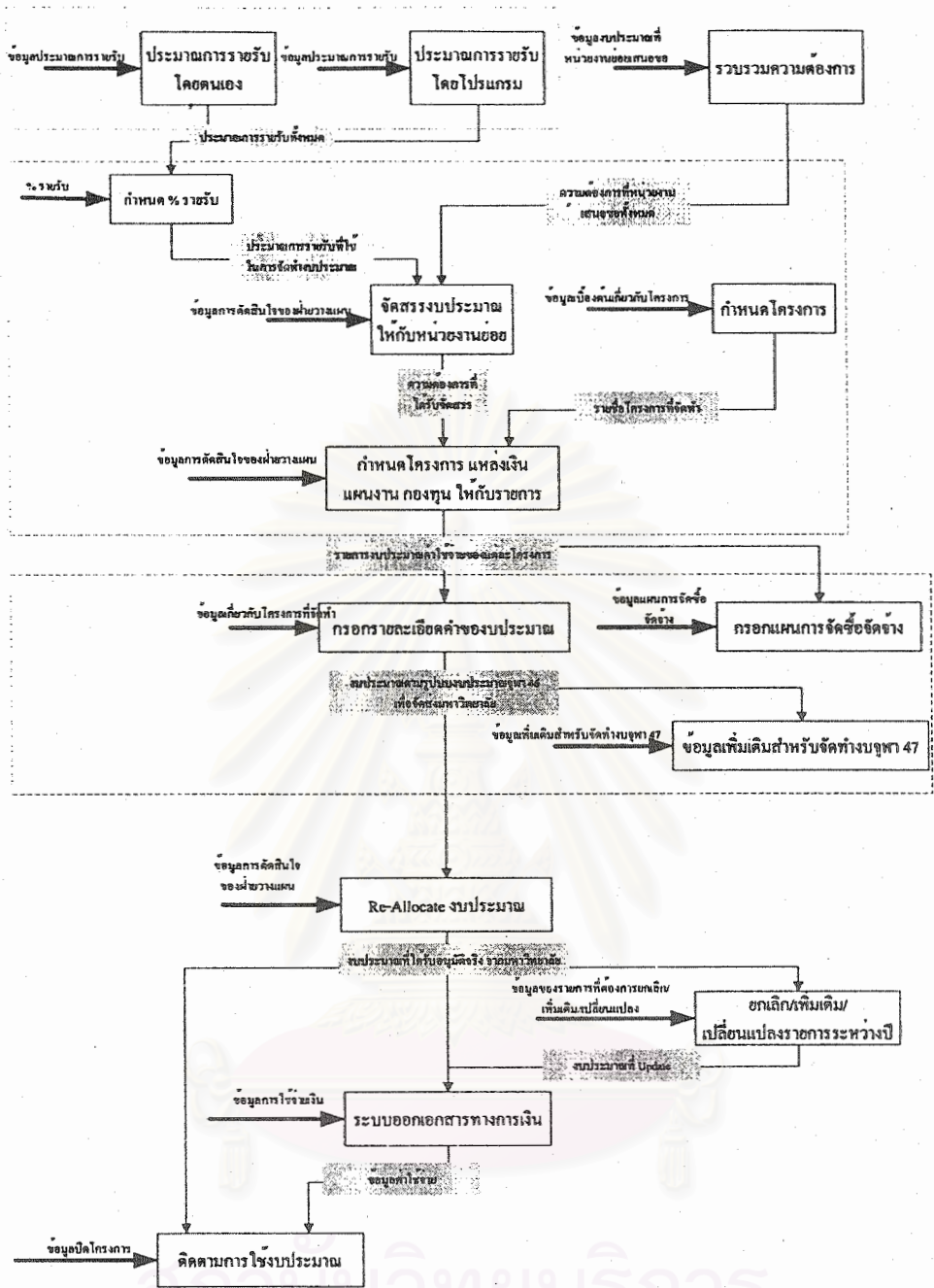
- Admin Module

- รวบรวมข้อมูลคั้งแบบฟอร์มการเก็บข้อมูล ดังในเอกสารอ้างอิง
- Login เข้า ChulaUp ในฐานะเจ้าของฐานข้อมูล
- ทำการเพิ่มรายชื่อบุคลากรตลอดจนรายละเอียดอื่นๆ ที่จำเป็นเข้าระบบ สำหรับหน่วยงานนั้นๆ
- ทำการเพิ่มรายชื่อ username & password สำหรับบุคลากรในการ login เข้า ChulaUP สำหรับหน่วยงานนั้นๆ
- ทำการกำหนดกลุ่ม และจำแนก username ลงแต่ละกลุ่มนั้น
- ทำการกำหนดสิทธิในการใช้งานในแต่ละกลุ่ม และในแต่ละ โมดูล
- ทำการสร้าง DSN-Organization Relationship
- เมื่อเสร็จขั้นตอนดังกล่าวทั้งหมดแล้ว ผู้ใช้งานคนอื่นๆ ที่ได้รับการสร้างขึ้นมา จะสามารถทำการ login และเข้ามาใช้งาน ChulaUP ได้ตามปกติ

- Asset Module

- รวบรวมข้อมูลคั้งแบบฟอร์มการเก็บข้อมูล ดังในเอกสารอ้างอิง
- ทำการติดตั้ง Printer Barcode โดยการลง Driver สำหรับเครื่องคอมพิวเตอร์ที่จะทำ Asset Module
- ทำการติดตั้ง Scanner โดยการลง Driver และ โปรแกรมที่ช่วยในการ scan รูปภาพ เอกสาร
- ทำการติดตั้ง Barcode Reader & Receiver ตามคู่มือ พร้อมกับทดสอบการใช้งาน
- ทำการกรอกค่าเริ่มต้นเกี่ยว
 - เครื่องพิมพ์เอกสาร
 - เครื่องพิมพ์ Barcode
 - ข้อมูลในการทำ FTP (File Transfer Protocol)
 - ข้อมูลเกี่ยวกับสถานที่จัดเก็บครุภัณฑ์
- เมื่อได้ค่าข้อมูลเบื้องต้นเป็นที่เรียบร้อยแล้ว จึงเริ่มเข้าสู่กระบวนการ ตรวจสอบ (ทั้งระบบเก่า และระบบใหม่), การทำการแก้ไขทะเบียนครุภัณฑ์, การยืมคืน, การโอนย้าย, การซ่อมแซม, การตรวจนับ ตลอดจนการจำหน่ายออกไป

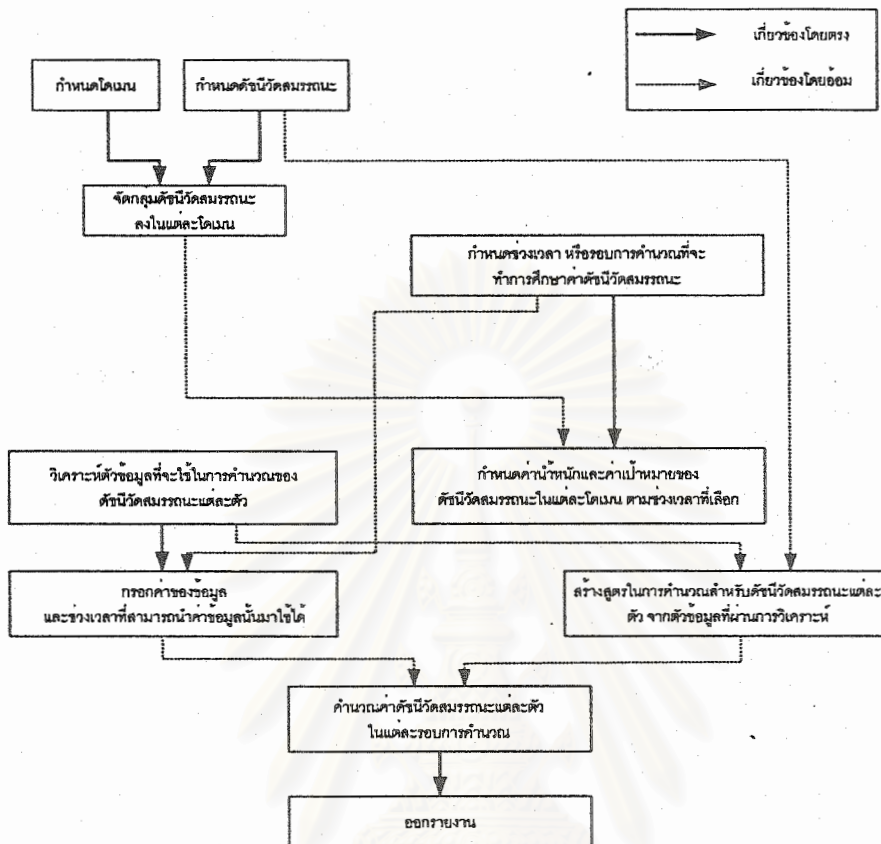
- Budgetary Module



KPI Module

- ทำการกรอกข้อมูล โดเมน
- ทำการกรอกข้อมูลดัชนีวัดสมรรถนะ
- ทำการ จัดกลุ่มดัชนีวัดสมรรถนะลงในแต่ละ โดเมน
- ทำการกำหนดช่วงเวลาที่ จะทำการศึกษาและวิเคราะห์ค่าดัชนีวัดสมรรถนะ
- ทำการกำหนดค่าน้ำหนักของดัชนีวัดสมรรถนะ ในแต่ละ โดเมน ในช่วงเวลาที่กำหนด
- ทำการค่าเป้าหมายของดัชนีวัดสมรรถนะ ในแต่ละ โดเมน ในช่วงเวลาที่เลือก
- ทำการกำหนดสูตรของดัชนีวัดสมรรถนะ
- ทำการวิเคราะห์สูตรของดัชนีวัดสมรรถนะเพื่อหาตัวข้อมูลทั้งหมดที่เกี่ยวข้องกับการคำนวณของสูตรต่างๆ เหล่านั้น

- ทำการกรอกค่าของตัวข้อมูลในช่วงเวลาที่เลือก
- ทำการคำนวณค่าดัชนีชี้วัดเพื่อแสดงผลในรูปแบบของเรดาร์ชาร์ตต่อไป



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข

นิตยภัตที่ใ้ในระบบ Chula UP



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1. IDEFO Integration Definition for Function Modeling
2. KPI Key Performance Indicators หรือ "ดัชนีวัดสมรรถภาพ" คือ "หัวข้อหรือรายการเฉพาะที่สำคัญๆ ที่เป็นตัวแทนอย่างเพียงพอต่อการแสดงออกหรือบ่งบอกหรือวัด ซึ่งความพึงพอใจหรือความคาดหวังของลูกค้า ที่กำหนดเลือกขึ้นมาด้วยความเต็มใจของผู้ผลิตผลงานให้เป็นเครื่องมือแสดงความมุ่งมั่นตั้งใจอย่างเป็นรูปธรรมที่จะทำให้ลูกค้า (ซึ่งอาจเป็นลูกค้าภายนอกผู้ซื้อสินค้าและบริการไปใช้ประโยชน์ หรืออาจเป็นกระบวนการถัดไปที่นำผลลัพธ์ ไปจัดการต่อ) ได้รับความพึงพอใจในผลงานที่ผลิตขึ้นนั้น เพื่อเปิดโอกาสให้ สามารถเก็บรวบรวมข้อมูลข้อเท็จจริงได้ เพื่อก่อให้เกิดการวัดค่าตามสูตรที่กำหนด เพื่อแสดงระดับหรือสถานะภาพที่เป็นจริงของความพึงพอใจของลูกค้า เพื่อประโยชน์ในการปรับปรุงต่อไปโครงสร้างแผนงาน ภาพรวมโครงสร้างแผนงานทั้งหมดของมหาวิทยาลัย ในระดับแผนงาน/งาน/กลุ่มกิจกรรม/กิจกรรมย่อย ที่กำหนดไว้ตามเอกสารโครงสร้างแผนงานปี 2545
3. ABM Activity Based Management

1. งบประมาณแผ่นดิน งบประมาณแผ่นดิน
2. งบเงินนอก งบประมาณเงินนอก หรือ งบประมาณเงินรายได้
3. โครงสร้างแผนงาน ภาพรวมโครงสร้างแผนงานทั้งหมดของมหาวิทยาลัย ในระดับ
แผนงาน/งาน/กลุ่มกิจกรรม/กิจกรรมย่อย ที่กำหนดไว้ตามเอกสารโครงสร้างแผนงานปี
2545
4. กองทุน กองทุนที่สภามหาวิทยาลัยกำหนดขึ้นตาม ซึ่งมีอยู่ด้วยกัน 9 กองทุน
5. ปีงบประมาณ ปีงบประมาณหนึ่งๆ จะมีระยะเวลา 1 ปี
6. งบประมาณ งบประมาณจุฬาลงกรณ์
7. Log-Frame ค่าของงบประมาณจุฬาลงกรณ์
8. โครงการ โครงการที่จัดทำขึ้นตามรูปแบบของงบประมาณจุฬาลงกรณ์
9. ABC ต้นทุนฐานกิจกรรม (Activity-Based Costing)
10. กิจกรรม การกระทำที่เปลี่ยนทรัพยากรของกิจการออกมาเป็นผลได้
11. Activity Driver ตัวผลักดันกิจกรรม

1. เจ้าหน้าที่ครุภัณฑ์ เจ้าหน้าที่ที่ทำหน้าที่ในการดูแลครุภัณฑ์ และทำหน้าที่ในการบันทึกและจัดเก็บประวัติครุภัณฑ์
2. ผู้รับผิดชอบครุภัณฑ์ ผู้ที่ทำหน้าที่ในการดูแลครุภัณฑ์เพื่อให้ครุภัณฑ์นั้นมีความพร้อมในการใช้งานตลอดเวลา และมีหน้าที่ในการติดตามครุภัณฑ์ในกรณีที่ครุภัณฑ์นั้นทำการตรวจนับแล้วไม่พบ
3. ผู้ครอบครองครุภัณฑ์ ผู้ที่มีสิทธิ์ในการครอบครองครุภัณฑ์นั้น โดยปกติแล้วจะเป็นคนเดียวกับผู้รับผิดชอบครุภัณฑ์ เว้นแต่ครุภัณฑ์นั้นได้ถูกยืม
4. สถานที่จัดเก็บครุภัณฑ์ สถานที่ที่ครุภัณฑ์ตั้งอยู่ หรือเป็นสถานที่ในการจัดเก็บครุภัณฑ์นั้น



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1. โดเมน (Domain) เกณฑ์หรือกลุ่มที่ใช้ในการจัดกลุ่มดัชนีชี้วัด
2. ดัชนีชี้วัด (Performance Indicator: PI) ดัชนีที่ใช้ในการวัดหรือประเมินว่าผลการดำเนินงานขององค์กรหรือหน่วยงานเป็นอย่างไร
3. ดัชนีชี้วัดหลัก (Key Performance Indicator: KPI) ดัชนีชี้วัด (PI) ที่มีความสำคัญที่ส่งผลกระทบต่อสมรรถนะขององค์กรหรือหน่วยงาน
4. Radar Chart เป็นกราฟแสดงความสัมพันธ์ระหว่างค่าเป้าหมาย (Target) และค่าจริง (Actual) สำหรับสะท้อนสมรรถนะการดำเนินงานภายใต้โดเมนต่างๆ
5. ส่วนการติดตั้ง (Setup) เป็นส่วนสำหรับกำหนดหรือติดตั้งข้อมูลบางรายการในช่วงเริ่มต้นให้กับโปรแกรม ซึ่งข้อมูลเหล่านี้เป็นข้อมูลที่มีความจำเป็นต่อการทำงานของโปรแกรม
6. ส่วนการดำเนินงาน (Operation) เป็นส่วนสำหรับให้ทำงานตามปกติของโปรแกรม
7. ส่วนสรุปรายงาน เป็นส่วนที่ใช้สำหรับดูรายงานสรุปผลการคำนวณดัชนีชี้วัด
8. การกำหนดแหล่งข้อมูล เป็นส่วนกำหนดแหล่งข้อมูลที่จำเป็น เพื่อนำไปใช้ในการคำนวณค่าดัชนีชี้วัด ซึ่งประกอบไปด้วย
 - 8.1 การระบุรายการเอกสาร เป็นการระบุรายละเอียดของรายการเอกสารซึ่งจะให้ข้อมูลนำเข้าระบบสำหรับนำไปใช้คำนวณดัชนีชี้วัด
 - 8.2 การระบุรายการข้อมูล เป็นการระบุรายการข้อมูลที่ต้องการในแต่ละเอกสารและระบบสนับสนุนเพื่อนำไปใช้คำนวณดัชนีชี้วัด
9. การกำหนดโดเมนและดัชนีชี้วัด เป็นส่วนของการกำหนดรายการโดเมนสำหรับใช้ในการสะท้อนสมรรถนะในด้านต่างๆของภาควิชา และการกำหนดรายการดัชนีชี้วัดสมรรถนะการดำเนินงาน ซึ่งประกอบไปด้วย
 - 9.1 การกำหนดโดเมน เป็นการกำหนดมุมมองสำหรับสำหรับสะท้อนสมรรถนะในด้านต่างๆของภาควิชา
 - 9.2 การกำหนดดัชนีชี้วัด เป็นการกำหนดรายการดัชนีชี้วัดสำหรับวัดผลสำเร็จของการดำเนินกิจกรรม
10. การจัดกลุ่มดัชนีชี้วัดเข้าตามโดเมน เป็นการจัดกลุ่มรายการดัชนีชี้วัดเข้าตามโดเมนที่มีความสัมพันธ์กับดัชนีชี้วัดนั้นๆ สำหรับสะท้อนสมรรถนะการดำเนินงานในด้านต่างๆ โดยอาศัยการให้ค่าระดับความสัมพันธ์ของดัชนีชี้วัดเชื่อมโยงกับโดเมนในรูปแบบของน้ำหนักความสัมพันธ์
11. การคำนวณดัชนีชี้วัด เป็นส่วนของการคำนวณค่าของดัชนีชี้วัด

12. การออกรายงาน เป็นการสรุปผลของการดำเนินงานดัชนีชี้วัดในแต่ละปี ออกมาในรูปแบบของรายงาน ซึ่งประกอบไปด้วย
- 12.1 รายงานดัชนีชี้วัดหลัก (KPI) เป็นการสรุปผลรายการดัชนีชี้วัดหลัก (KPI) ที่ใช้สะท้อนถึงผลการดำเนินงานโดยรวมในด้านต่างๆของทางภาควิชา ออกมาในรูปแบบของรายงานจำแนกตามโดเมน สำหรับให้ผู้บริหารทำการพิจารณาและประเมินผลการดำเนินงานในด้านต่างๆของภาควิชาในภาพรวมได้
 - 12.2 รายงานดัชนีชี้วัด (PI) เป็นการสรุปผลรายการดัชนีชี้วัด (PI) ที่ใช้สำหรับวัดผลการดำเนินกิจกรรมในด้านต่างๆของทางภาควิชาออกมาในรูปแบบของรายงานจำแนกตามโดเมน สำหรับให้ผู้บริหารทำการติดตามผลการดำเนินกิจกรรมต่างๆของทางภาควิชา ว่าสามารถดำเนินการได้บรรลุผลสำเร็จตามวัตถุประสงค์ที่ต้องการมากน้อยเพียงใด
 - 12.3 รายงาน Radar Chart เป็นรายงานที่ใช้สะท้อนถึงผลการดำเนินงานโดยรวมในด้านต่างๆของทางภาควิชา ออกมาในรูปแบบของกราฟแสดงความสัมพันธ์ระหว่างค่าเป้าหมาย (Target) และค่าจริง (Actual) ของผลการดำเนินงานในแต่ละโดเมน
13. ระบบสนับสนุน เป็นระบบสนับสนุนการดำเนินงานของภาควิชาวิศวกรรมอุตสาหการ ประกอบด้วย ระบบการวางแผนและติดตามการใช้งบประมาณ(Budgetary Module) ระบบการคำนวณต้นทุนฐานกิจกรรม (ABC Module) และระบบการบริหารสินทรัพย์ (Asset Module)

สารบัญ

Generic Network Configuration.....	1
ศัพท์เบื้องต้นเกี่ยวกับระบบ Network	1
IP Address.....	2
Network Class	3
IP Routing	5
อุปกรณ์ที่ใช้ในเน็ตเวิร์ค	5
หลักการพื้นฐานของ IP Routing.....	6
Subnet Addressing	9
Subnet Mask.....	11
Client/ Server Model	13
องค์ประกอบของ Client/ Server.....	13
เทคโนโลยี Client/ Server ในปัจจุบัน.....	13
ข้อดีของการพัฒนาระบบงานแบบ Client/ Server	14
จุดอ่อน (ข้อด้อย) ของระบบงานแบบ Client/ Server.....	14
Server	15
ความหมาย	15
ประเภทของ Server	15
ขนาดของ Server.....	15
มาตรฐานประสิทธิภาพ	16
การแบ่งประเภทของ Server	16
คุณสมบัติของ Server.....	18
ระบบการประมวลผลแบบ On-Line	21
(On-Line Transaction Processing System)	21
ความหมาย	21
รูปแบบการประมวลผลงานแบบ On-Line	21
On-Line Interactive System.....	21
Database Concept	24
ระบบฐานข้อมูล.....	24
การจัดเก็บข้อมูลในรูปแบบแฟ้มข้อมูล.....	24
การจัดเก็บข้อมูลในรูปแบบฐานข้อมูล	24
ข้อเสียของการจัดเก็บข้อมูลในรูปแบบฐานข้อมูล	25

ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System)	25
คำศัพท์ที่เกี่ยวข้อง	25
ประเภทของคีย์	26
กฎที่ใช้กับฐานข้อมูลเชิงสัมพันธ์	27
ภาษาทางด้านฐานข้อมูล (Query Language).....	28
ขั้นตอนการทำการออกแบบฐานข้อมูล	30
1. ทำความเข้าใจระบบงานให้ชัดเจน	30
2. ร่างความสัมพันธ์คร่าว ๆ โดยแบ่งออกเป็นกลุ่ม ๆ	30
3. กำหนดความสัมพันธ์และคีย์หลักของข้อมูล	31
รูปแบบบรรทัดฐาน (Normalization)	32
วัตถุประสงค์ของการทำให้เป็นบรรทัดฐาน มีดังนี้	32
ข้อมูลที่ซ้ำซ้อนกันนี้จะก่อให้เกิดปัญหา	32
หลักการทำให้ Normalization	32

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

รูปที่ 1 Network Class	3
รูปที่ 3 การสื่อสารในเน็ตเวิร์กที่ต่อร่วมกัน (Shared Network)	7
รูปที่ 4 การสื่อสารระหว่าง 2 เน็ตเวิร์ก	7
รูปที่ 6 แสดงการจัดแบ่ง IP Address ของคลาส ออกเป็นเน็ตเวิร์กย่อยด้วยวิธี Subnetting โดยแบ่งพื้นที่ส่วนที่เป็นของ hostid เดิมออกเป็น 2 ส่วน โดยเป็นของ subnetid ขนาด 8 บิต และ hostid ใหม่ที่มีขนาดเล็กลงเหลือเพียง 8 บิต	10
รูปที่ 7 Symmetric Multiprocessing	19
รูปที่ 8 Functional Multiprocessing	19
รูปที่ 9 On-Line Interactive System	21
รูปที่ 10 Off-Line Transaction Processing System	22
รูปที่ 11 Batch Processing System	23
รูปที่ 12 On-Line Real Time Interactive System	23
รูปที่ 13 แสดงให้เห็นการแบ่งกลุ่มข้อมูล	30

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก ค

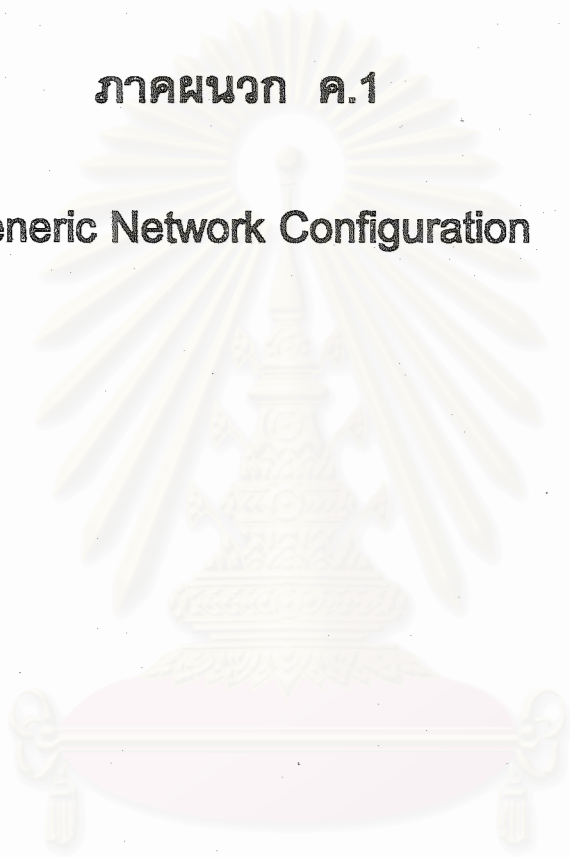
ระบบคอมพิวเตอร์และเครือข่าย



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค.1

Generic Network Configuration



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Generic Network Configuration

ศัพท์เบื้องต้นเกี่ยวกับระบบ Network

ระบบเครือข่าย หรือ Network หมายถึงการที่เรานำคอมพิวเตอร์ ตั้งแต่ 2 เครื่องขึ้นไป มาเชื่อมต่อกัน เพื่อที่จะให้คอมพิวเตอร์ เหล่านี้ สามารถใช้ทรัพยากร ร่วมกันได้ เช่นใช้ Hard disk หรือ CD-ROM drive ร่วมกันได้ เพื่อเป็นการลดรายจ่าย ในการจัดซื้อ หรือเพื่อส่งข่าวสารหรือข้อมูลถึงผู้ใช้งานอื่นๆ ในเครือข่ายเป็นต้น ระบบเครือข่าย แบ่งเป็น 2 ลักษณะใหญ่ๆ คือเครือข่าย LAN และ WAN

LAN มาจาก Local Area Network คือเครือข่ายคอมพิวเตอร์ระยะใกล้ เช่นในโรงเรียน หรือสำนักงาน ที่ได้ที่หนึ่ง การที่คอมพิวเตอร์แต่ละเครื่อง จะสามารถเชื่อมโยงถึงกันได้ ต้องมีอุปกรณ์สำคัญคือ LAN Card หรือ Network Interface Card (NIC) และสายที่ใช้ในการเชื่อมต่อ ซึ่งมีหลายชนิดเช่น Coaxial Cable, สาย Unshielded Twisted Pair (UTP) หรือสาย Fiber Optic ขึ้นอยู่กับ ชนิดของเครือข่ายว่าต้องการใช้งาน ใกล้-ไกลเท่าใด

WAN มาจาก Wide Area Network คือการเชื่อมต่อระบบ LAN ตั้งแต่ 2 ระบบขึ้นไปเข้าด้วยกัน อาจจะอยู่บริเวณเดียวกัน หรืออยู่ไกลกันมากก็ได้ ซึ่งการเชื่อมต่อนี้ สามารถทำผ่านสื่อต่างๆ ได้หลายชนิด เช่นสายทองแดงที่เป็น lease line (สายโทรศัพท์ธรรมดา) ผ่านไมโครเวฟ หรือผ่านดาวเทียมเป็นต้น อุปกรณ์ที่ทำให้การเชื่อมต่อสำเร็จได้คือ Router ซึ่งทำหน้าที่เป็น ประตูเชื่อมต่อ หรือ Gateway

การเชื่อมโยงคอมพิวเตอร์ หลายๆเครือข่ายเข้าด้วยกัน ถ้าเป็นระบบที่แตกต่างกัน ก็ไม่สามารถใช้งาน หรือติดต่อกันได้ ดังนั้นจึงจำเป็นต้องมีมาตรฐาน ในการติดต่อสื่อสารกัน เรียกว่า Protocol และในระบบ Internet ได้ใช้ TCP/IP (Transmission Control Protocol/ Internet Protocol) เป็นมาตรฐานในการติดต่อสื่อสารกัน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

IP Address

Internet Protocol Address ประกอบด้วยตัวเลขฐานสอง 4 bytes (32 bits) เวลาใช้เพื่อให้ง่ายต่อการเข้าใจ จะเขียนอยู่ในรูป เลขฐานสิบ ขึ้นด้วยจุด "." เมื่อแปลงเป็นเลขฐานสิบแล้ว แต่ละชุดจะมีตัวเลขระหว่าง 0-255 เช่น 192.168.1.1 เป็นต้น เรียกว่า IP Address

Internet Protocol Networks คือตัวเลข IP Addresses ที่เรียงต่อกันไป ซึ่งจะมีความหมายดังนี้ Address ทั้งหมดที่อยู่ภายใน network เดียวกันและมีตัวเลขเหมือนกัน เรียกว่า "network portion" ตัวเลขที่เหลือจะเรียกว่า "host portion" ตัวเลขที่ใช้ร่วมกันสำหรับทุก addresses ภายในเน็ตเวิร์ค เรียกว่า "netmask" และ netmask นี้ก็ใช้เป็นตัวกำหนดว่า addresses ไດอยู่ภายใน network หรือ addresses ไດไม่อยู่

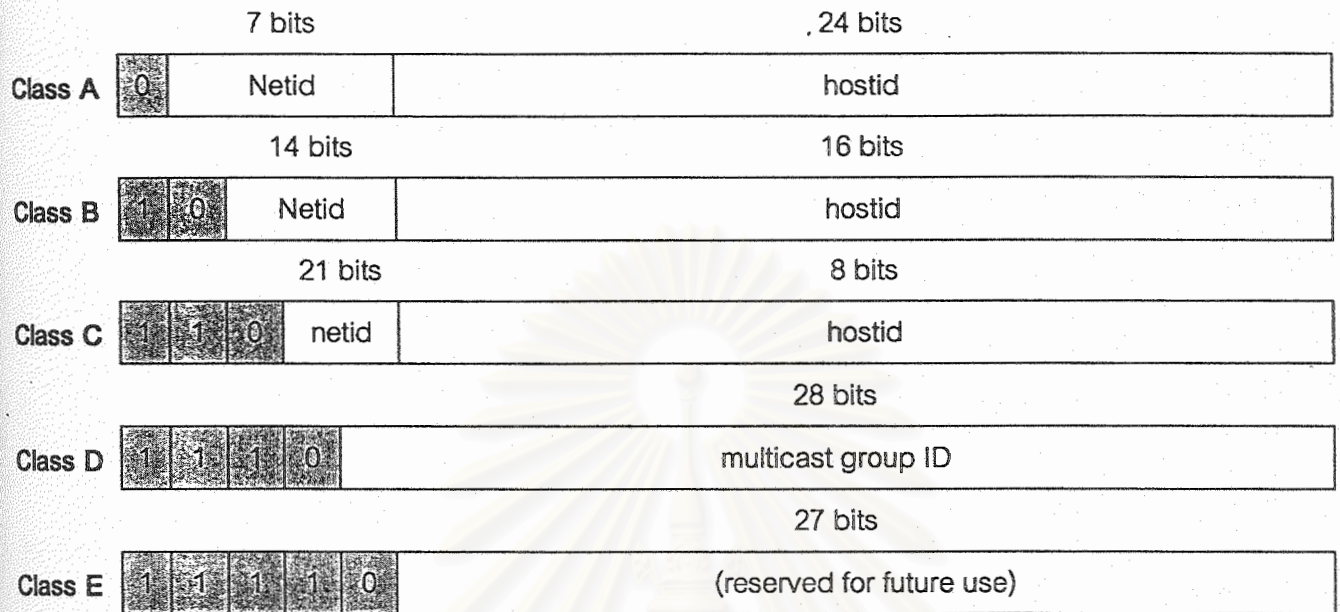
Network Portion	192.168.1.	ภายใน network นี้ ทุก address จะขึ้นต้นด้วยเลขนี้เหมือนกันหมด
Host Portion	0.1	แต่ละ host ก็จะมีเลข host portion เป็นของตัวเองไม่ซ้ำกัน
Host Address	192.168.1.10	Network Portion + Host Portion
Network Mask (netmask)	255.255.255.0	ทุก addresses ภายใน network เดียวกัน ต้องใช้ค่าเดียวกัน และหมายเลขนี้ ก็จะแสดงให้รู้ว่า ภายใน network นี้มี IPs รวมกันทั้งหมดกี่หมายเลข ในที่นี้ก็จะ มี 256 IPs
Network Address	192.168.1.0	
Broadcast Address	192.168.1.255	

ตารางที่ 1 อธิบายค่าต่างๆ ใน IP Address

จุฬาลงกรณ์มหาวิทยาลัย

Network Class

การแบ่ง IP Address ออกเป็นหลายระดับ ความจริงก็แบ่งตามความใหญ่ ของเครือข่ายนั่นเอง



รูปที่ 1 Network Class

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Network Class	Range
A	0.0.0.0 - 127.255.255.255
B	128.0.0.0 - 191.255.255.255
C	192.0.0.0 - 223.255.255.255
D	224.0.0.0 - 239.255.255.255
E	240.0.0.0 - 247.255.255.255

ตารางที่ 2 Network Class

Class A Network

ระบบเครือข่ายคลาส A มีแอดเดรสเริ่มต้นตั้งแต่ 0 จนถึง 127 (Network Portion) ตัวเลขที่เหลืออีกสามส่วน เป็น host portion จะเห็นว่าเป็นระบบเครือข่ายที่ใหญ่มาก สามารถมีอุปกรณ์ (router, host ฯ) ต่างๆ ไม่ซ้ำกันได้ถึง 16 ล้านตัว ระบบเครือข่ายคลาส A ได้ถูกกำหนดให้กับบริษัทต่าง จนหมดไปนานแล้ว

Class B Network

มีแอดเดรสเริ่มตั้งแต่ 128 จนถึง 192 มีระบบเครือข่ายคลาส B อยู่ 16,000 เครือข่าย ในแต่ละเครือข่าย จะมีแอดเดรสที่ไม่ซ้ำกันอยู่ 64,000 แอดเดรส และก็ถูกกำหนดให้กับองค์กร หรือบริษัทใหญ่ๆ หมดแล้วเช่นกัน เหตุที่เครือข่ายนี้ไม่ใหญ่เท่า เครือข่ายคลาส A เนื่องจากถูกกำหนดโดย netmask

Class C Network

มีแอดเดรสเริ่มตั้งแต่ 192 จนถึง 223 ซึ่งมีอยู่ประมาณ 2 ล้านเครือข่าย แต่ละเครือข่ายมีอุปกรณ์ได้สูงสุด 254 ตัว

Class D Network (Multicast)

หรืออาจเรียกได้ว่าเป็น เครือข่ายคลาส D มีแอดเดรสเริ่มต้นตั้งแต่ 224 จนถึง 239 สำหรับใช้งานในลักษณะเหมือนกับ วิทยูหรือโทรทัศน์ คือส่งออกไปอย่างเดียว ใครจะรับก็ไปดักรับเอา ส่วนแอดเดรส 240 ถึง 247 เป็นของระบบเครือข่ายคลาส E ซึ่งในส่วนนี้เก็บไว้ใช้ในอนาคต

IP Routing

IP Routing เป็นกระบวนการค้นหาเส้นทางในการส่งผ่านข้อมูลจากต้นทางไปยังที่หมายปลายทางโดยผ่านการส่งต่อของอุปกรณ์ IP ที่อยู่ในเน็ตเวิร์คซึ่งจะช่วยกันทำหน้าที่ส่งต่อข้อมูลไปจนกว่าข้อมูลจะถึงปลายทาง กลไกสำคัญที่ทำให้ IP เป็นโปรโตคอลสำหรับขนส่งข้อมูลไปยังทุกๆ ที่ในโลกบนอินเทอร์เน็ตที่ดีที่สุดในขณะนี้คือการที่ IP มีกระบวนการ IP Routing นี้เอง สิ่งที่น่าสนใจที่สุดของ IP Routing คือการที่ต้นทางและปลายทางของการสื่อสารนั้นในบางโอกาสต่างก็อยู่กันแสนไกล การสื่อสารข้อมูลแต่ละครั้ง ข้อมูลจะต้องเดินทางผ่านโครงข่ายอันสลับซับซ้อนมากมาย แต่ในที่สุดข้อมูลก็สามารถส่งถึงกันได้ในเวลาอันรวดเร็วเป็นที่น่าอัศจรรย์ โครงข่ายอินเทอร์เน็ตคงไม่อาจเกิดขึ้นได้หากไม่มีโปรโตคอล IP ที่ช่วยขนส่งข้อมูลไปบนเครือข่ายอย่างมีประสิทธิภาพ การเข้าใจถึงกระบวนการ IP Routing จะช่วยให้เราเข้าใจคุณสมบัติของอินเทอร์เน็ตได้เป็นอย่างดี กระบวนการ IP Routing นี้ได้ถูกออกแบบมาอย่างชาญฉลาดและรัดกุมพอสมควรในอันที่จะให้บรรลุภารกิจในการขนส่งข้อมูล หลักการพื้นฐานของ IP Routing เริ่มต้นด้วยข้อกำหนดที่เรียบเรียงดังนี้

อุปกรณ์ที่ใช้ในเน็ตเวิร์ค

Host โฮสต์เป็นอุปกรณ์ที่ทำหน้าที่ให้กำเนิดข้อมูลในกรณีเป็นผู้ส่ง หรือทำหน้าที่รับข้อมูลไปใช้งานในกรณีเป็นผู้รับ การสื่อสารข้อมูลใดๆ จะต้องเป็นการสื่อสารจากโฮสต์ไปยังโฮสต์เสมอ

Router เราเตอร์เป็นอุปกรณ์สำคัญอย่างยิ่งสำหรับ IP ที่จะทำให้การขนส่งข้อมูลเป็นไปอย่างสมบูรณ์ เราเตอร์ทำหน้าที่ในการส่งผ่านข้อมูลจากเน็ตเวิร์คหนึ่งไปยังอีกเน็ตเวิร์คหนึ่ง ตำแหน่งของเราเตอร์จะอยู่ในจุดที่เชื่อมต่อระหว่างสองเน็ตเวิร์คเข้าด้วยกัน ด้วยข้อกำหนดของ IP ข้อมูลจะส่งไปถึงกันโดยตรงข้ามเน็ตเวิร์คได้ จะต้องอาศัยเราเตอร์เป็นผู้ทำหน้าที่ส่งผ่านข้อมูลไปให้ ดังนั้นเน็ตเวิร์คของ IP ถึงแม้ไม่ได้ต่อกันในทางกายภาพ แต่ก็สามารถสื่อสารกันได้โดยอาศัยเราเตอร์เป็นตัวเชื่อมประสานเข้าด้วยกัน

Network เน็ตเวิร์คในที่นี้จะกล่าวถึงเฉพาะเน็ตเวิร์คในความหมายของ IP (IP Network) เท่านั้น ไม่รวมถึงเน็ตเวิร์คประเภทอื่น ใน IP นั้นจะมีการระบุหมายเลขประจำโฮสต์โดยใช้ IP Address เพื่อระบุตำแหน่งของต้นทาง และปลายทาง โดยใน IP Address นั้นนอกจากจะระบุตำแหน่งของโฮสต์แล้ว ยังใช้ระบุตำแหน่งของเน็ตเวิร์คที่โฮสต์นั้นเชื่อมต่อด้วย ทั้งในโปรโตคอล IP มีกระบวนการที่จะแยกหมายเลขประจำตัวของโฮสต์และของเน็ตเวิร์คออกจากกัน เพื่อให้อุปกรณ์ทั้งหลายสามารถพิจารณาได้ว่าทันทีว่าจะส่งข้อมูลที่ได้รับมานั้นไปในทิศทางใด



หลักการพื้นฐานของ IP Routing

IP Routing โดยใช้ Default Router

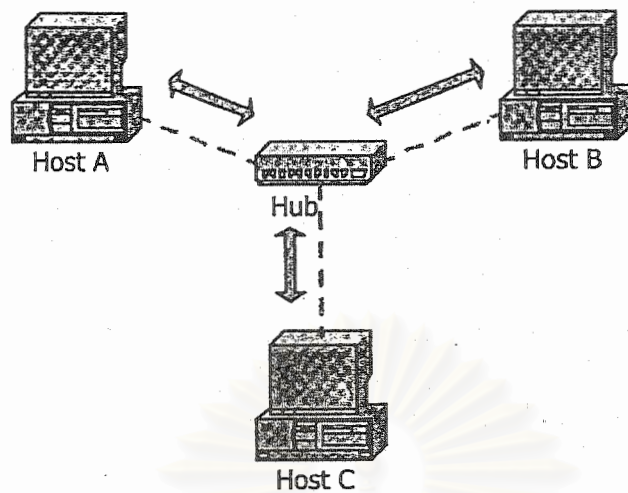
กระบวนการ IP Routing เริ่มต้นด้วยหลักพื้นฐานที่ไม่สลับซับซ้อนและเข้าใจได้ไม่ยากคือ



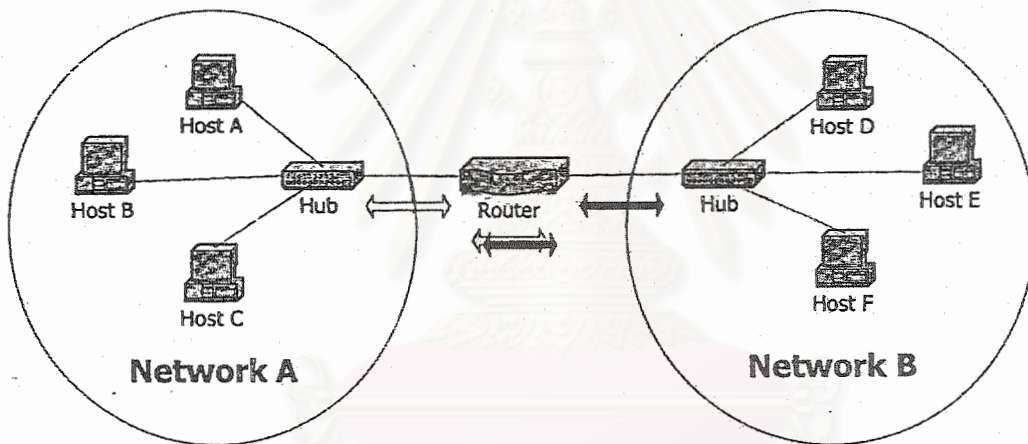
รูปที่ 2 การสื่อสารในการเชื่อมต่อแบบจุดต่อจุด

1. ถ้าโฮสต์ต้นทางและปลายทางต่อถึงกันโดยตรงเช่นการเชื่อมต่อแบบจุดต่อจุด ข้อมูลจะถูกส่งไปยังโฮสต์ปลายทางโดยตรง
2. ถ้าโฮสต์ต้นทางและปลายทางต่อเชื่อมร่วมอยู่ในเน็ตเวิร์คเดียวกัน เช่น อีเทอร์เน็ตหรือโทเคนริง ดังแสดงในภาพที่ 4.3 ข้อมูลจะถูกส่งไปยังโฮสต์ปลายทางโดยตรง
3. ถ้าไม่เป็นไปตามข้อที่ 1 และ 2 ข้อมูลจะถูกส่งไปยังดีฟอลต์เราเตอร์เพื่อทำการส่งต่อข้อมูลไปยังปลายทางต่อไป
4. เมื่อเราเตอร์ได้รับ ข้อมูลจากข้อ 3 แล้ว ตรวจสอบดู หากพบว่าโฮสต์ปลายทางต่อร่วมอยู่บนเน็ตเวิร์คเดียวกันกับเราเตอร์ ให้ทำการส่งข้อมูลไปที่โฮสต์นั้น หากไม่ได้ต่อร่วมกันก็ส่งข้อมูลไปที่เราเตอร์ตัวต่อไป และกลับไปขั้นตอนในข้อที่ 2 ใหม่ จนกว่าข้อมูลจะเดินทางถึงปลายทางหรือหมดเวลาในการส่ง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



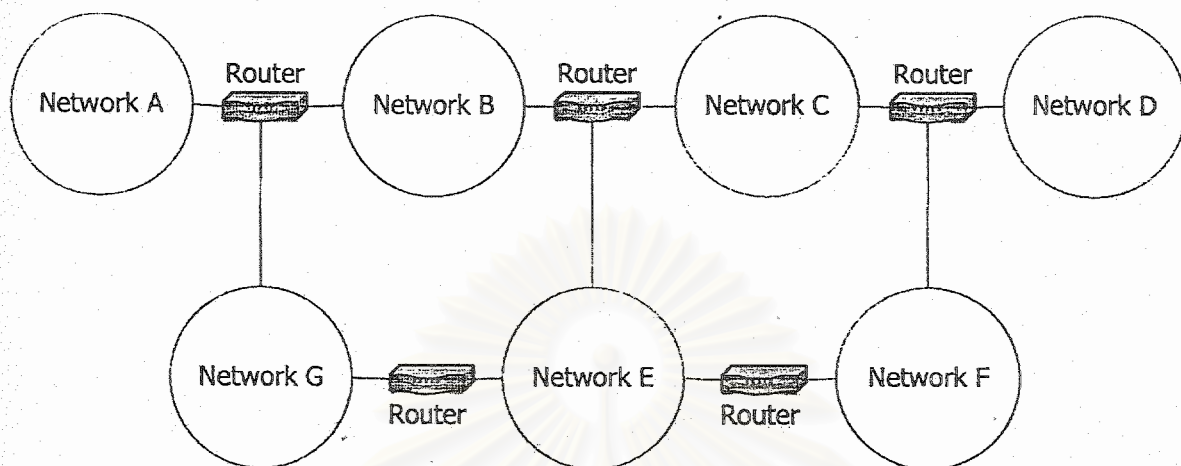
รูปที่ 3 การสื่อสารในเน็ตเวิร์กที่ต่อรวมกัน (Shared Network)



รูปที่ 4 การสื่อสารระหว่าง 2 เน็ตเวิร์ก

หากเน็ตเวิร์กมีเพียง 2 เน็ตเวิร์กเหมือนในรูปที่ 4 มีเพียงแค่อุปกรณ์เราเตอร์ก็คงจะเพียงพอและการทำงานในการส่งข้อมูลข้ามเน็ตเวิร์กก็คงจะไม่ยุ่งยากมากนัก และคงเป็นไปตามขั้นตอนข้างต้น หากสังเกตเห็นว่าตัวเราเตอร์เองนั้นจะมีเน็ตเวิร์กที่ต้องติดต่อกับ 2 ฝั่ง คือ network A และ network B ซึ่งมีทิศทางการเคลื่อนที่ของข้อมูลเพียงเส้นทางเดียว มีเราเตอร์เพียงตัวเดียว ไม่ว่าปลายทางของข้อมูลจะไปที่ไหนหากมีข้อมูลอยู่ในเน็ตเวิร์กเดียวกันแล้ว ข้อมูลทั้งหมดก็ต้องส่งผ่านเราเตอร์อยู่ดี โดยไม่ต้องทำการวิเคราะห์ใดๆ การที่ข้อมูลถูกส่งข้ามเน็ตเวิร์ก 1 ครั้ง เรียกว่า 1 ฮอป (hop) เปรียบเสมือนระยะในการเดินทางของข้อมูลจากภาพตัวอย่างข้อมูลเดินทางจากโฮสต์ต้นทางเพียง 1 ฮอปก็ถึงโฮสต์ปลายทาง การส่งต่อข้อมูลโดยเราเตอร์ก็มีเพียงส่งไปและส่งกลับระหว่าง network A กับ network B เท่านั้น แต่หาระยะทางถึงโฮสต์ปลายทางจะต้องเดินทางมากกว่า 1 ฮอปแล้ว เราเตอร์ก็จะทำงานยุ่งยาก ซับซ้อนขึ้นเพราะมีเน็ตเวิร์กอื่นๆ

ที่ไม่ได้เชื่อมต่อโดยตรงและต้องส่งข้อมูลผ่านเราเตอร์หลายตัว และมีหลายเส้นทางที่ข้อมูลสามารถเดินทางไปได้ ดังนั้นการส่งข้อมูลของเราเตอร์จึงเป็นปัจจัยสำคัญในการกำหนดประสิทธิภาพของ IP Routing



รูปที่ 5 การเชื่อมต่อกันของหลายเน็ตเวิร์ก

การเดินทางของข้อมูลโดยกระบวนการ IP Routing นั้นทำงานอยู่บนพื้นฐานของการส่งข้อมูลที่ละฮอป (hop-by-hop) คือเราเตอร์เองจะทำงานโดยรู้จักเฉพาะเน็ตเวิร์กที่ต่ออยู่กับตัวเองเท่านั้น หากโฮสต์ปลายทางมิได้อยู่ในเน็ตเวิร์กที่ต่อเชื่อมอยู่ ก็จะทำการส่งข้อมูลต่อไปอีกฮอปให้แก่เราเตอร์ตัวต่อไปส่งต่อ และถือว่าหมดหน้าที่ต่อข้อมูลนั้นแล้ว เพราะการส่งต่อข้อมูลไปเรียบร้อยแล้ว ส่วนจะถึงปลายทางหรือไม่ นั้นเป็นอีกเรื่องหนึ่ง และเราเตอร์ตัวอื่นๆ ที่อยู่ระหว่างทางก็เช่นกัน ก็จะส่งต่อข้อมูลไปเรื่อยๆ เช่นนั้นที่ละฮอปทีละฮอปจนกว่าจะถึงปลายทางหรือหมดเวลา

เพื่อให้กระบวนการ IP Routing ดำเนินไปอย่างมีประสิทธิภาพจึงมีการเพิ่มความสามารถของเราเตอร์ให้มากขึ้นกล่าวคือ ในกรณีที่โฮสต์ปลายทางมิได้อยู่ในเน็ตเวิร์กที่ต่ออยู่กับตัวเองนั้นแทนที่จะทำการส่งต่อข้อมูลไปยังดีฟอลต์เราเตอร์ทั้งหมด ก็ให้เราเตอร์ทำการพิจารณาเน็ตเวิร์กปลายทางว่าอยู่ที่ใดแล้วจึงทำการส่งต่อข้อมูลนั้นไปยังเราเตอร์ที่อยู่ใกล้กับเน็ตเวิร์กมากที่สุด (ใช้จำนวนฮอปในการส่งข้อมูลน้อยที่สุด) เพื่อการนี้จึงจำเป็นต้องมีข้อมูลให้แก่ตัวเราเตอร์ว่าเน็ตเวิร์กใดควรจะส่งข้อมูลไปยังเราเตอร์ใด ข้อมูลเหล่านี้จะเก็บอยู่ในเรตติ้งเทเบิล (Routing Table) ซึ่งจะประกอบด้วยข้อมูลดังนี้

Destination IP Address: หมายถึงแอดเดรสของโฮสต์หรือเน็ตเวิร์กปลายทาง

IP Address of a next-hop router: หมายถึง IP Address ของเราเตอร์ตัวอื่นที่ต่อโดยตรงอยู่บนเน็ตเวิร์กเดียวกัน

Flags: จะเป็นข้อมูลส่วนที่ขยายความเพิ่มเติมของ Destination IP Address และ Next-hop router

Interface: หมายถึงอินเตอร์เฟซของเราเตอร์ที่จะต้องใช้เพื่อการส่งข้อมูลออกไป

เมื่อเราเตอร์มีเราต์ติ้งเทเบิลแล้ว กระบวนการในการที่จะส่งข้อมูลจากเราเตอร์ตัวหนึ่งไปยังตัวต่อไป จะต้องนำข้อมูลในเราต์ติ้งเทเบิลไปร่วมพิจารณาด้วยดังนี้

- ค้นหาข้อมูลในเราต์ติ้งเทเบิลเพื่อหา IP Address ที่ตรงกันพอดีกับ IP Address ของโฮสต์ปลายทางของข้อมูล หากพบให้ส่งไปยังแอดเดรสที่ระบุอยู่ในฟิลด์ของ Next-hop router ทันที หากไม่พบข้อมูลให้ทำต่อในข้อถัดไป
- ค้นหาในเราต์ติ้งเทเบิลเพื่อหาเน็ตเวิร์คแอดเดรสที่ตรงกับเน็ตเวิร์คแอดเดรสของโฮสต์ปลายทาง หากพบให้ส่งไปยังแอดเดรสที่ระบุอยู่ในฟิลด์ของ Next-hop router ทันที หากไม่พบข้อมูลให้ทำต่อในข้อถัดไป
- ค้นหาในเราต์ติ้งเทเบิลเพื่อหาข้อมูลรายการที่ระบุไว้ว่า "Default" และส่งต่อไปยังแอดเดรสที่ระบุอยู่ในฟิลด์ของ Next-hop router

กระบวนการที่กล่าวมานี้จะทำให้การส่งต่อข้อมูลเป็นไปในทิศทางที่เหมาะสมและมีประสิทธิภาพที่สุด โดยอาศัยข้อมูลที่กำหนดไว้ก่อนแล้วของเราเตอร์ ซึ่งหากไม่พบข้อมูลที่ตรงกับโฮสต์หรือเน็ตเวิร์คเลยในเราต์ติ้งเทเบิลแล้ว ข้อมูลก็จะถูกส่งไปยังดีฟอลต์เราเตอร์เสมอ ดังนั้น หากเราเตอร์แต่ละตัวต่างก็มีเราต์ติ้งเทเบิลที่ถูกต้องแล้ว ข้อมูลก็จะถูกส่งไปเรื่อยๆ จนถึงปลายทางในที่สุด อาจจะมีล่าช้าหรือไม่มีประสิทธิภาพบ้างแต่ก็พอใช้งานได้

จะเห็นได้ว่าเราเตอร์ไม่จำเป็นต้องเก็บเราต์ติ้งเทเบิลของทุกๆ เน็ตเวิร์คหรือทุกๆ โฮสต์ไว้ทั้งหมด แต่จะอาศัยการกระจายกันของข้อมูลในเราเตอร์ทุกๆ ตัวบนเน็ตเวิร์คแต่ละตัวก็ส่งต่อข้อมูลให้ถูกต้อง ข้อมูลก็จะสามารถเดินทางถึงปลายทางได้ ด้วยเหตุนี้เองทำให้เน็ตเวิร์คสามารถขยายเพิ่มเติมออกไปได้เรื่อยๆ โดยไม่จำกัดและไม่ต้องทำการแก้ไขโครงสร้างของเน็ตเวิร์คเดิม อินเทอร์เน็ตจึงแผ่ขยายครอบคลุมโลกได้อย่างรวดเร็ว

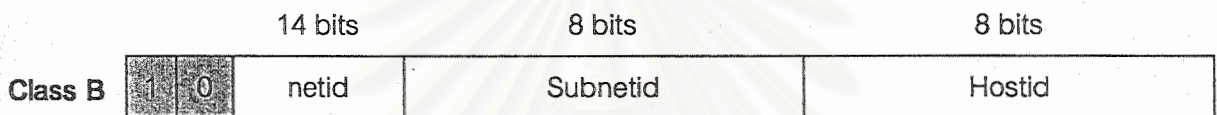
Subnet Addressing

ในตอนเริ่มต้นใช้โปรโตคอล TCP/IP นั้นการแบ่ง IP Address ออกเป็นแอดเดรสของเน็ตเวิร์ค (netid) และแอดเดรสของโฮสต์ (hostid) เป็นไปตามกติกาที่ระบุของแต่ละคลาส ต่อมีผู้เสนอให้มีการการแบ่งเน็ตเวิร์คย่อยภายในแต่ละ netid เพิ่มขึ้นอีก เพื่อจะได้ใช้งาน IP Address ได้อย่างมีประสิทธิภาพที่สุด

เนื่องในคลาส A และ คลาส B นั้นมีการจัดสรรส่วนที่เป็น hostid ในแต่ละเน็ตเวิร์คเป็นจำนวนมาก คือในเน็ตเวิร์คคลาส A แต่ละเน็ตเวิร์คนั้นสามารถมีจำนวนโฮสต์ได้มากถึง $2^{24}-2=16,777,214$ โฮสต์ และสำหรับในเน็ตเวิร์คของคลาส C นั้นสามารถมีจำนวนโฮสต์ได้สูงสุดถึง $2^{16}-2=65,534$ โฮสต์ ซึ่งการที่

จะนำ IP Address มาใช้อย่างทั่วถึงนั้นมีโอกาสเป็นไปได้ยากมาก ทั้งคลาส A และ คลาส B เพราะมีโอกาสน้อยมากที่จะมีเน็ตเวิร์คใดในโลกที่มีจำนวนโฮสต์มากมายขนาดนั้นอยู่ภายในเน็ตเวิร์คเดียว ดังนั้น IP Address ที่จัดสรรให้ไปในแต่ละเน็ตเวิร์คของคลาสเหล่านั้นจึงถูกใช้ไม่หมดและไม่สามารถนำไปใช้ประโยชน์ที่อื่นได้

การทำ subnet คือการแบ่งเน็ตเวิร์คย่อยภายในเน็ตเวิร์คเพื่อให้แต่ละเน็ตเวิร์คมีขนาดที่เหมาะสมกับปริมาณโฮสต์ที่มีอยู่ โดยใช้หลักการเดียวกับการนำ IP Address มาแยกออกเป็น Host id และ Network id คือแทนที่จะให้ค่า Host id เป็นค่าอิสระตั้งแต่ 1 จนถึงค่าสูงสุด ก็ทำการจัดกลุ่มของ host id เหล่านั้น ออกเป็นกลุ่มของเน็ตเวิร์คย่อย คือ นำค่าในส่วนที่เป็น host id เดิมมาแยกออกเป็นสองส่วนคือ subnet id และเป็น host id ใหม่ ซึ่งจะทำให้สามารถจัดสรรการใช้งาน IP Address ได้อย่างเหมาะสมกับการมีอยู่จริงของโฮสต์ในแต่ละเน็ตเวิร์ค



รูปที่ 6 แสดงการจัดแบ่ง IP Address ของคลาส ออกเป็นเน็ตเวิร์คย่อยด้วยวิธี Subnetting โดยแบ่งพื้นที่ส่วนที่เป็นของ hostid เดิมออกเป็น 2 ส่วน โดยเป็นของ subnetid ขนาด 8 บิต และ hostid ใหม่ที่มีขนาดเล็กลงเหลือเพียง 8 บิต

Class B	จำนวนเน็ตเวิร์คที่มีได้	จำนวนโฮสต์สูงสุดในแต่ละเน็ตเวิร์ค
เดิม	16,382	65,532
หลังการ subnet	4,161,028	254

ตารางที่ 3 ผลกระทบต่อจำนวนของและขนาดของเน็ตเวิร์คเมื่อถูก subnet

ผลที่ได้คือขนาดของเน็ตเวิร์คจะเล็กลงและมีจำนวนมากขึ้น จากเดิม 16,382 เน็ตเวิร์คก็เพิ่มเป็น 4,161,028 ($16,382 * 254$) และในขณะเดียวกันจำนวนโฮสต์ในแต่ละเน็ตเวิร์คจะลดลงจาก 65,532 เหลือเพียง 254 โฮสต์ อย่างไรก็ตามการ subnet นั้นไม่จำเป็นต้องมีขนาดของ subnet id คงที่ตายตัวเสมอไป ผู้บริหารระบบสามารถปรับเปลี่ยนได้ตามความเหมาะสมของการใช้งาน เช่นอาจจะมีจำนวนเน็ตเวิร์คน้อยลง และจำนวนโฮสต์มากขึ้นสามารถกระทำได้โดยการแบ่งขนาดของ subnet id และ host id ใหม่ตามต้องการ

นอกจากการที่สามารถใช้ IP Address ได้อย่างมีประสิทธิภาพแล้ว ข้อดีอีกอย่างหนึ่งของการ subnet คือช่วยให้ประสิทธิภาพการสื่อสารดีขึ้นด้วย กล่าวคือ กระบวนการของ TCP/IP บางประเภทมีการใช้การสื่อสารแบบบรอดคาสต์ (broadcast) เพื่อทำการสื่อสารกระจายไปยังทุกๆ โฮสต์ที่อยู่ในเน็ตเวิร์ควง

เดียวกัน ดังนั้นหากเป็นเน็ตเวิร์คคลาส A ซึ่งมีโฮสต์ได้ถึง 16 ล้านโฮสต์แล้ว การสื่อสารด้วยวิธีบรอดคาสต์แต่ละครั้งจะเป็นการกระจายข้อมูลไปยังเครื่องอื่นๆ จำนวนมาก และใช้แบนด์วิดท์มากมายมหาศาลทั่วไปทั้งเน็ตเวิร์ค ส่งผลกระทบต่อการสื่อสารตามปกติอย่างยิ่ง และหากเน็ตเวิร์คประเภทนี้ถูกโจมตีแล้วก็มีโอกาสมากที่การสื่อสารภายในเน็ตเวิร์คจะเป็นอัมพาตได้อย่างรวดเร็ว ผู้ออกแบบเน็ตเวิร์คส่วนใหญ่จึงมักจะหลีกเลี่ยงการออกแบบให้เน็ตเวิร์คมีขนาดใหญ่เกินไป เนื่องจากควบคุมได้ยุ่งยากและมีประสิทธิภาพต่ำ วิธีการ subnet จึงเป็นส่วนที่ถูกนำมาใช้ในการออกแบบเสมอ

โดยทั่วแล้วมักจะพบเห็นการ subnet สำหรับ IP Address ในคลาส B เสียเป็นส่วนใหญ่ เนื่องจากคลาส B มีผู้ใช้งานกันแพร่หลาย ส่วนคลาส A จะพบได้ไม่บ่อยนัก เพราะมีผู้ได้รับจัดสรรไม่มาก ส่วนคลาส C ก็อาจจะพอมีผู้ทำ subnet อยู่บ้าง แต่เนื่องจากคลาส C มีขนาดของเน็ตเวิร์คไม่ใหญ่อยู่แล้ว จึงสามารถแบ่งย่อยออกไปได้อีกไม่มากนัก อย่างไรก็ตาม IP Address ทุกคลาสล้วนแต่สามารถถูกนำมา subnet ได้ทั้งสิ้น

Subnet Mask

หากกล่าวถึงการ subnet แล้วย่อมจะต้องกล่าวถึง subnet mask ด้วย การที่มีเฉพาะ IP Address เพียงอย่างเดียว นั้นกรณีที่เป็นการกำหนด netid และ hostid ตามที่ระบุในคลาสต่างๆ นั้นเราก็สามารถทราบค่าทั้งสองได้ไม่ยากนัก โดยพิจารณาว่า IP Address ดังกล่าวอยู่ในช่วงใด อยู่ในคลาสใด หลักจากนั้นก็ สามารถแยก netid และ hostid ได้จากการเปรียบเทียบกับมาตรฐานของคลาสเหล่านั้น

แต่เมื่อมีการแบ่งเน็ตเวิร์คย่อยโดยการ subnet แล้ว ย่อมไม่สามารถใช้วิธีการข้างต้นเพื่อหา netid และ hostid ได้อีกต่อไป เนื่องจาก subnet นั้นสามารถกำหนดได้โดยผู้ออกแบบเน็ตเวิร์คเอง และมีได้มีข้อบังคับแต่อย่างใด ดังนั้นจึงจำเป็นต้องมีการระบุค่าใดค่าหนึ่งไว้เพื่อให้สามารถนำมาใช้ในการหาค่าจาก IP Address ได้ว่าเป็น hostid, netid และ subnet id และค่านี้ก็คือ subnet mask นั่นเอง

Subnet mask เป็นตัวเลขขนาด 32 บิตเท่ากับ IP Address ทำหน้าที่ระบุหมายเลขของ host id และ net id + subnet id ของโฮสต์นั้น การกำหนดค่าของ subnet mask จะอยู่ในรูปแบบเดียวกับ IP Address คือทำการแบ่ง subnet mask ออกเป็นเลข 8 บิตจำนวน 4 ชุด และแยกแต่ละชุดออกจากกันด้วยจุด (.)

ตัวอย่างของ Subnet mask เช่น

FF.FF.FF.00 (hex)

255.255.255.0 (dec)

ค่า subnet mask นี้จะต้องกำหนดไว้บนทุกโฮสต์คู่กันกับค่า IP Address เสมอ เนื่องจากโปรโตคอล IP จำเป็นต้องใช้ค่านี้ไปคำนวณค่า netid ซึ่งจะเป็นอย่างยิ่งในกระบวนการ IP Routing อย่างไรก็ตามค่า

subnet mask นี้จะไม่ถูกส่งไปด้วย โฮสต์ทั้งหนายสามารถนำค่า subnet mask มาทำการทางคณิตศาสตร์กับ IP Address ก็จะสามารถหาค่า host id, subnet id, netid ออกมาได้โดยวิธีดังนี้

$$\text{Netid} + \text{subnet id} = [\text{IP address}] \text{ AND } [\text{Subnet Mask}]$$

$$\text{Hostid} = [\text{IP Address}] \text{ AND } (\text{NOT } ([\text{Subnet Mask}])))$$

ตัวอย่าง

$$\text{IP Address} = 192.168.15.20 \quad \text{Subnet Mask} = 255.255.255.0$$

$$\text{Netid} = 192.168.15.20 \text{ AND } 255.255.255.0 = 192.168.15.0$$

$$\text{Hostid} = 192.168.15.20 \text{ AND } 0.0.0.255 = 0.0.0.20$$

หรือพูดง่าย ๆ ได้ว่า "Netid ก็คือ IP Address ส่วนที่ตรงกับบิตของ Subnet mask ที่มีค่าเป็น 1 ส่วน hostid คือ IP Address ส่วนที่ตรงกับ Subnet mask ที่มีค่าเป็น 0" นั่นเอง

ดังนั้นพึงระลึกเสมอว่านอกจากการกำหนด IP Address ที่ถูกต้องแล้ว การกำหนดค่า Subnet Mask ก็มีผลต่อ IP Routing เช่นเดียวกัน การกำหนดค่า Subnet Mask ผิดพลาดย่อมจะส่งผลให้การสื่อสารข้อมูลของ IP ไม่สามารถจะกระทำได้เช่นกัน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Client/ Server Model

องค์ประกอบของ Client/ Server

- Client

มักจะเรียกว่า ตัวลูก คือ เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นผู้รับ-ส่งข้อมูลข่าวสาร และคำสั่งจากผู้ใช้ระบบงานไปให้แก่ Server (ตัวแม่) เพื่ออ่านข้อมูลประมวลผลและส่งกลับมาให้ผู้ใช้
- Server

มักจะเรียกว่า ตัวแม่ คือ เครื่องคอมพิวเตอร์ ที่ทำหน้าที่เป็นผู้รับ-ส่งข้อมูลข่าวสาร คำสั่งจาก Client เพื่ออ่านข้อมูลผล และส่งกลับมาให้ Client ซึ่ง Server 1 ตัวอาจจะมี Client ที่ต่อเชื่อมอยู่ในระบบงานได้หลายตัว และในแต่ละเครือข่ายอาจจะมี Server ก็ตัวก็ได้ตามความเหมาะสมของแต่ละระบบงาน
- Networking

คือ ระบบงานที่ประกอบไปด้วยอุปกรณ์ ฮาร์ดแวร์ และซอฟต์แวร์เพื่อเป็นทางเดินให้กับข้อมูลข่าวสาร คำสั่ง โปรแกรมที่มีการรับ-ส่งระหว่าง Client กับ Server ที่ต่อเชื่อมโยงกัน

เทคโนโลยี Client/ Server ในปัจจุบัน

การพัฒนาแบบระบบงานแบบ Client/ Server คือ ระบบงานที่มีการจัดแบ่งหน้าที่การทำงาน การประมวลผลของแต่ละงานให้เครื่องคอมพิวเตอร์ (Client หรือ Server) ที่มีความเหมาะสมมากที่สุดทำการประมวลผล เพื่อให้เกิดประสิทธิภาพในการทำงานสูงสุด เช่น

- Client ควรจะทำงานเกี่ยวกับระบบการรับ-แสดงผลทางจอภาพ
- Server ควรจะทำงานทางด้าน Database Management & Storage เป็นต้น

โดยทั่วไปการออกแบบระบบงานแบบ Client/ Server มักจะออกแบบให้เครื่องแม่ข่ายและลูกข่ายทำงานในลักษณะดังกล่าวข้างต้น ดังนั้นในอนาคตไม่ว่าจะมีการเพิ่มขยายเครือข่าย หรือ Client มากแค่ไหนก็ตาม งานที่เพิ่มขึ้นจะอยู่ที่ Client เกือบทั้งหมด โดยที่ Server จะมีงานเพิ่มเพียงคำสั่งโปรแกรมจาก Client ที่เพิ่มขึ้นมาเท่านั้น

ในการออกแบบโปรแกรมประยุกต์ หน้าที่ของ Client ในการประมวลผลนั้นควรจะต้องกำหนดให้ Client ทำหน้าที่ตรวจสอบความเป็นไปได้ของข้อมูลที่ผู้ใช้บันทึกเข้ามา ตรวจสอบผิดพลาดของข้อมูล เพื่อ

ป้องกัน กลั่นกรองไม่ให้ Client ส่งข้อมูลที่ผิดๆ ไปให้ Server ทำงาน ซึ่งมีผลทำให้ลดภาระงานของ Server และปริมาณงานบนเครือข่ายลดลง ยังมีผลถึงประสิทธิภาพที่สูงขึ้นด้วย

ข้อดีของการพัฒนาระบบงานแบบ Client/ Server มีดังนี้

1. ประหยัดงบประมาณในการลงทุน
2. เพิ่ม/ ก่อให้เกิดประสิทธิผลสูง
3. มีความยืดหยุ่น และสามารถขยายขีดความสามารถ/ ประสิทธิภาพได้
4. ก่อให้เกิดการใช้ทรัพยากรที่มีอยู่ให้เกิดประโยชน์สูงสุด
5. สามารถบริหาร ควบคุมจากส่วนกลาง
6. มีคุณสมบัติที่เป็นระบบงานแบบเปิด
 - สามารถทำงานภายใต้ Multiple Environments คือ สามารถทำงานประมวลผลงานภายใต้ Platform และ Environment ที่หลากหลาย
 - สามารถใช้ Database Application ที่หลากหลายได้ ผู้ใช้สามารถใช้ Application Software โปรแกรมที่คุ้นเคยเพื่อเข้าถึงฐานข้อมูลได้ ผู้พัฒนาระบบงานสามารถเลือกใช้ Front-end/ Programming Language/ Development Tools ได้ตามความเหมาะสม ความชำนาญ
 - สามารถทำงาน ประมวลผลงานข้ามระหว่างระบบงานฯ ต่างๆ ได้ และยังสามารถโยกย้ายระบบงานจากแพลตฟอร์มหนึ่งไปอีกแพลตฟอร์มหนึ่งได้

จุดอ่อน (ข้อด้อย) ของระบบงานแบบ Client/ Server

1. งบประมาณการลงทุนที่ค่อนข้าง มีผู้ที่พยายามพัฒนาระบบงานคอมพิวเตอร์แบบ Client/ Server คาดหมายงบประมาณในการลงทุนที่ผิดพลาดไป ซึ่งรายการงบประมาณการลงทุนที่ควรจะเป็นสรุปได้เป็น 7 รายการดังนี้
 - Hardware and Equipment
 - System Software and Utility
 - Network and Data Communication
 - Commercial DBMS and Development Tools
 - Application Software Development and Consulting Services
 - Training and Education
 - Maintenance
2. ค่าใช้จ่ายในการฝึกอบรม การรับบริการ การบำรุงรักษา และการบริหารจัดการสูง
3. ผู้ให้บริการมีหลากหลาย

การพัฒนาระบบงานแบบ Client/ Server จำเป็นต้องอาศัยองค์ประกอบทางด้าน Hardware, System

Software, Utility, DBMS, Development Tools, Network and Data Communication ที่ผู้ขายแต่ละด้านมีความสามารถ ความเชี่ยวชาญและเหมาะสมแตกต่างกันไป ซึ่งมีผลทำให้ผู้พัฒนาระบบงานฯ บางคนไม่สามารถคัดเลือก หรือตัดสินใจ เลือกใช้องค์ประกอบเหล่านั้นได้อย่างเหมาะสม และอาจมีผลกระทบไปถึงความสำเร็จ หรือ ล้มเหลวของโครงการฯ ได้

4. ขาดเครื่องมือในการบริหารจัดการระบบงานแบบกระจายศูนย์คอมพิวเตอร์
5. ขาดมาตรฐาน ข้อจำกัดในข้อ 3 มีผลทำให้ขาดมาตรฐานของการประมวลผลร่วมกันข้าม Platform ไม่สามารถทำงานอย่างมีประสิทธิภาพเท่าที่ควรจะเป็น
6. เทคโนโลยีและเทคนิคยังขาดความสมบูรณ์
7. ขาดโปรแกรมช่วยในการปรับเปลี่ยน/ โยกย้ายระบบงาน
การโยกย้าย ปรับเปลี่ยน ระบบงานเดิมที่มีลักษณะเป็น text based, proprietary system หรือ ระบบงานเดิม ไปเป็นระบบงานแบบใหม่ที่เป็นแบบ Client/ Server ใช้เทคนิคของ Graphical User Interfaces (GUI) นั้น ยังขาดเครื่องมือที่จะช่วยในการโยกย้ายระบบงานฯ ทำให้เป็นอุปสรรค และข้อจำกัดของการปรับเปลี่ยนระบบงานเดิมที่พัฒนาและใช้งานมานานๆ แล้ว

Server

ความหมาย

Server หมายถึง เครื่องคอมพิวเตอร์ทุกๆ ขนาด ทุกๆ ชนิด ที่ทำหน้าที่ตามที่เรากำหนดได้หลากหลาย ไม่ว่าจะเป็น File Sharing, Applications Processing, บริหารจัดการเกี่ยวกับข้อมูล บริหารจัดการเกี่ยวกับระบบงานเครื่องข่ายและการสื่อสารโทรคมนาคม ประมวลผลตามคำสั่งโปรแกรม ค้นหา/ เรียกข้อมูลจากระบบเครื่องคอมพิวเตอร์อื่นๆ ในเครือข่าย รับ-ส่งข้อมูลไปให้ระบบเครื่องคอมพิวเตอร์อื่นๆ ในเครือข่าย และยังสามารถทำหน้าที่อื่นๆ ตามที่จะกำหนดได้อีกด้วย

ประเภทของ Server

Server สามารถแบ่งออกได้หลายประเภทตามที่จะกำหนดหน้าที่ คุณสมบัติให้เหมาะสมกับระบบงานที่จะใช้ เช่น File Server, Data Server, Database Server, Compute Server, Communication Server, Application Server เป็นต้น

ขนาดของ Server

ขนาดของ Server มีได้ตั้งแต่ PC ขนาดเล็กไปจนถึง High Power Workstation ขนาดใหญ่มาก (RAM อาจจะมีตั้งแต่ 4 MB ขึ้นไป ตามแต่ผู้ผลิตฮาร์ดแวร์)

มาตรฐานประสิทธิภาพ

ได้มีหน่วยงานองค์กร สมาคม ที่รวมกลุ่มกันกำหนดมาตรฐานทางด้านประสิทธิภาพของอุปกรณ์ Client และ Server เช่น Transaction Processing Counsel (TPC) เป็นสมาคมของผู้ผลิตฮาร์ดแวร์และซอฟต์แวร์ ก่อตั้งเมื่อปี 1988 เพื่อร่วมกันกำหนดมาตรฐานของการประมวลผลฐานข้อมูล โดยสมาคมฯ นี้ ได้กำหนดมาตรฐานได้ 3 กลุ่มระดับคือ

- TPC-A

ใช้หลักการคำนวณประสิทธิภาพของการประมวลผล จำนวนรายการ (Transaction) ที่ถูกบันทึกเข้ามาจากหลายจอภาพ (Terminal/ PC-Client) ที่สามารถทำงานได้ภายในเวลา 1 วินาที เป็นตัววัดประสิทธิภาพ

- TPC-B

เป็นมาตรฐานของการประมวลผลของระบบงานที่เน้นที่ Database Engine และประสิทธิภาพของการประมวลผลแบบ Batch File Processing

- TPC-C

เป็นมาตรฐานของการทดสอบประสิทธิภาพของระบบงานแบบ Client/ Server ในสิ่งแวดล้อมหลากหลายรูปแบบของฮาร์ดแวร์และซอฟต์แวร์ แต่จะทำงานประเภท OLTP (On-Line Transaction Processing)

ในปัจจุบัน มาตรฐานหลักเกณฑ์ของ TPC ได้กลายเป็นมาตรฐานที่เปรียบเทียบอ้างอิงถึงประสิทธิภาพการทำงาน การประมวลผลงานของเครื่องคอมพิวเตอร์ ที่เป็นที่ยอมรับและใช้กันมากดังจะเห็นได้จากการเข้าร่วมสัมมนา หรืออ่านเอกสารข้อมูลทางด้านเทคนิคของเครื่องคอมพิวเตอร์ยี่ห้อต่างๆ รุ่นต่างๆ ของผู้ผลิตฯ เราจะพบข้อมูลอ้างอิงถึงประสิทธิภาพการทำงานการประมวลผลงานของเครื่องคอมพิวเตอร์เหล่านั้น โดยอาศัยเกณฑ์ของ TPC เป็นมาตรฐานอ้างอิงอยู่บ่อยครั้ง

การแบ่งประเภทของ Server

ในปัจจุบันมีนักคอมพิวเตอร์ และผู้ให้ระบบงานคอมพิวเตอร์จำนวนมาก มีความสับสน ในการแบ่งประเภทของคอมพิวเตอร์ Server ที่ขายกันอยู่ในท้องตลาด ทั้งนี้อาจเนื่องจากการโฆษณาประชาสัมพันธ์ของผู้ผลิตฯ หรือความเจริญก้าวหน้าอย่างรวดเร็วของเทคโนโลยีคอมพิวเตอร์และสารสนเทศได้ จึงทำให้มีการเรียกชื่อ Server ตามชื่อทางการค้า ตามชื่อทางด้านการทำงานและหน้าที่ที่หลากหลายออกไป ซึ่งจะขอแบ่งประเภทของ Server ตามลักษณะของการประมวลผลข้อมูลเป็น 6 ประเภทคือ

1 File Server

คือเครื่องคอมพิวเตอร์ Server ที่ทำหน้าที่จัดการด้าน application และ data files โดยจะรับ-ส่งข้อมูลไปให้ Client ตามที่กำหนด มีลักษณะการทำงานเป็น I/O Oriented เป็นส่วนใหญ่ ในการทำงานการประมวลผลงาน File Server มักจะอ่านข้อมูลจากฐานข้อมูลที่มีอยู่ทั้งหมด และส่งผ่านเข้าไปในระบบงานเครือข่าย เพื่อไปที่ Client ที่ต้องการ จากนั้น Client จะทำหน้าที่ประมวลผลงานของข้อมูลตามคำสั่งโปรแกรมที่ถูกกำหนดไว้

File Server ควรจะมีอุปกรณ์สำหรับเชื่อมโยงเครือข่าย มีฮาร์ดดิสก์ที่มีความจุปริมาณมากๆ และทำงานได้รวดเร็ว ในการทำงานรูปแบบนี้ อาจมีโอกาที่จะเกิดปัญหาในระบบของ file ได้ เช่น File Locking คือการที่ Client ตัวใดตัวหนึ่งเรียกและประมวลผลข้อมูลจากแฟ้มข้อมูลใดแฟ้มข้อมูลหนึ่งที่อยู่ใน Server จะมีผลทำให้ Client ตัวอื่นๆ ไม่สามารถเรียก และประมวลผลข้อมูลจากแฟ้มข้อมูลนั้นๆ จนกว่า Client ตัวแรกจะประมวลผลเสร็จ

2 Application Server

คือ เครื่องคอมพิวเตอร์ Server ที่ทำหน้าที่เสมือนเป็น host โดยอาจสามารถนำเอา application software จาก host เดิม มาใช้ในสิ่งแวดล้อมของ application Server ได้เลย บางครั้งอาจเรียกว่า host-based Client/ Server และเราอาจพัฒนาระบบงานฯ ให้มีการใช้เทคนิคของ GUI (Graphical User Interface) เพิ่มเติมที่ Client ได้

3 Data Server

คือ เครื่องคอมพิวเตอร์ Server ที่ทำหน้าที่จัดเก็บและจัดการเกี่ยวกับข้อมูล (Data Oriented) โดยเฉพาะเป็นเสมือนคลังข้อมูลขนาดใหญ่ โดยทั่วไปจะต่อเชื่อมโยงกับ Compute Server และมักจะถูกกำหนดไม่ให้ประมวลผลทางด้านโปรแกรมใดๆ (Application Logic Processing) ยกเว้น การประมวลผลสถานะเบื้องต้นของตัวข้อมูลเท่านั้น เช่น Data Validation เป็นต้น Data Server จำเป็นต้องมีโพรเซสเซอร์ที่มีประสิทธิภาพสูง หน่วยความจำมากๆ และ Disk Storage ที่มีความจุมากๆ นอกจากนี้ Data Server 1 ตัว อาจจะต้องเชื่อมเพื่อให้บริการข้อมูลกับ Compute Server ได้ หลายตัวขึ้นอยู่กับ Application Software Environment ที่ออกแบบไว้

4 Compute Server

คือ เครื่องคอมพิวเตอร์ Server ที่ทำหน้าที่รับคำร้องขอข้อมูลจาก client ส่งต่อไปที่ Data Server และรับข้อมูลจาก Data Server (อาจจะมีการประมวลผลบางส่วน) และส่งกลับไปที่ client ตัว Compute server ควรจะต้องมีโพรเซสเซอร์ที่มีประสิทธิภาพสูง RAM มากๆ แต่ Disk Storage ไม่จำเป็นต้องมีความจุมากนัก

5 Database Server

คือ เครื่องคอมพิวเตอร์ Server ที่ทำหน้าที่รับคำร้องขอข้อมูลจาก client ค้นหาข้อมูล และส่งกลับไปให้ client โดยที่โปรแกรม (Application Program Logic) จะอยู่ที่ client (มีลักษณะการทำงานเหมือนกับรวม Data Server กับ Compute Server เข้าด้วยกัน) ลักษณะการทำงานแบบนี้เป็นการประมวลผลที่อาจเรียกได้ว่าเป็นรูปแบบที่แท้จริงของ Client/ Server System Development

Database Server ควรจะต้องออกแบบให้ส่งข้อมูลเฉพาะที่ client ต้องการเท่านั้น ไม่ส่งทั้งหมดของ File/ Table ซึ่งทำให้เพิ่มประสิทธิภาพของระบบงานเครือข่าย การติดต่อสื่อสาร (Networking and Data Communication)

การออกแบบ/ เลือก Database Server นี้ เราควรจะต้องพิจารณาขนาดของข้อมูล, Response Time ที่คาดหวัง จำนวนผู้ใช้งาน และรูปแบบของ Networking ที่ใช้ ทั้งนี้เพื่อให้เกิดประสิทธิภาพสูงสุด

6 Communication Server

คือ เครื่องคอมพิวเตอร์ Server ที่ทำหน้าที่ในการบริหารจัดการ Gateways เพื่อทำการติดต่อสื่อสารกับ LAN ที่มีลักษณะแตกต่างกัน ระบบเครือข่ายที่แตกต่างกัน รวมทั้งเครื่องข่ายที่ใช้ Midrange/ Mainframe Computer เพื่อสามารถติดต่อสื่อสาร แลกเปลี่ยนข้อมูลข่าวสารระหว่างกันได้อย่างมีประสิทธิภาพ

โดย Communication Server นี้จะต้องมี Multiple Slots โพรเซสเซอร์ความเร็วสูง ทำการแปล Networking Protocols ที่หลากหลายในเครือข่าย

คุณสมบัติของ Server

ถึงแม้ว่าเครื่องคอมพิวเตอร์ แทบทุกรุ่น แทบทุกขนาด หรือแทบทุกยี่ห้อจะสามารถนำมาเป็นเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Server ในระบบงานเครือข่ายได้ก็ตาม แต่นักวิเคราะห์และออกแบบระบบงาน Client/ Server ควรจะต้องเข้าใจ และพิจารณาคูณลักษณะ คุณสมบัติของเครื่องคอมพิวเตอร์เหล่านั้น ได้ตามความเหมาะสมของงาน (Application Software) โดยการพิจารณาจากคุณสมบัติ เช่น

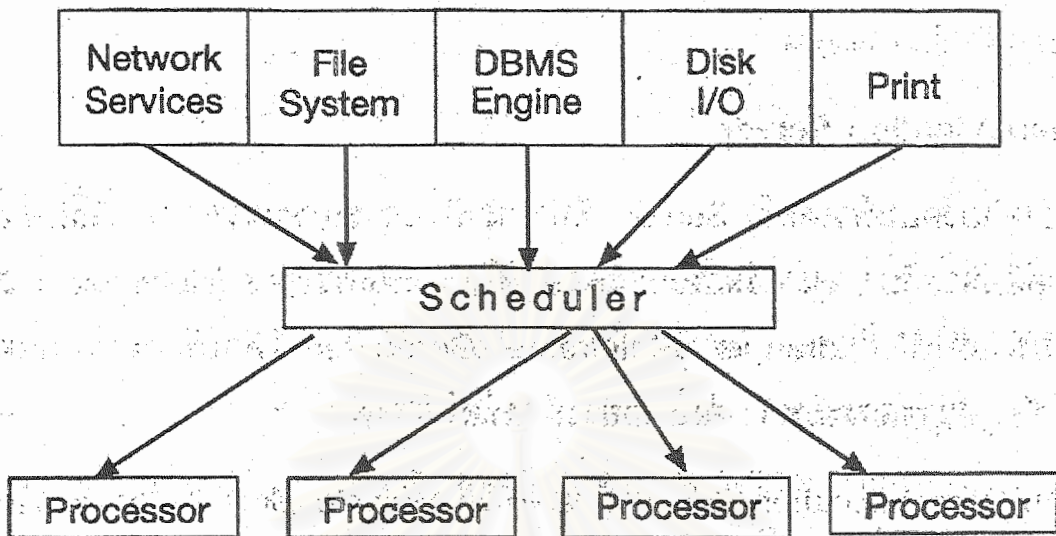
Multiprocessing

คือระบบเครื่องคอมพิวเตอร์ที่ออกแบบให้สามารถติดตั้ง Multiple Processor ได้ ซึ่งจะทำให้ server มีประสิทธิภาพของการประมวลผลสูงขึ้น มีลักษณะการทำงาน 2 แบบคือ

- Symmetric Multiprocessing

คือการประมวลผลของงานใดๆ ที่สามารถจะส่งไปประมวลผลที่โพรเซสเซอร์ใดก็ได้ที่มีความพร้อมมากกว่า โดยจะต้องมีโปรแกรมจาก Network Operating System Server คอยควบคุม สั่งการดังรูปที่ 7 เช่น Banyan's VINES and The Santa Cruz Operation's SCO UNIX เป็นต้น (บริษัท Microsoft Inc. ประกาศว่า WindowsNT มีลักษณะการทำงานเป็นแบบ Symmetric

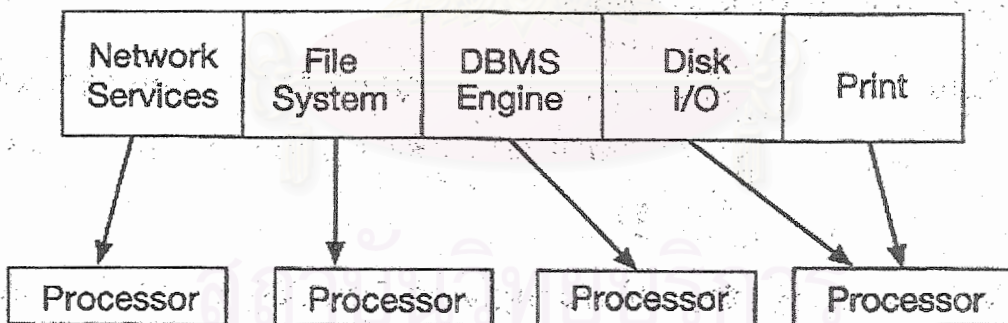
Multiprocessing นี้ด้วย เมื่อมกราคม 2540 บริษัท Novell Inc. ก็ประกาศว่า Netware มีการทำงานลักษณะนี้เช่นกัน)



รูปที่ 7 Symmetric Multiprocessing

- Functional Multiprocessing

คือการประมวลผลของงานที่ถูกกำหนดให้ส่งไปประมวลผลที่ Processor ตัวใดตัวหนึ่งอย่างถาวรแน่นอน ดังรูปที่ 8 เช่น Microsoft's LAN Manager เป็นต้น



รูปที่ 8 Functional Multiprocessing

6.1 Multithreading

Thread คือ หน่วยของงานที่เล็กที่สุดที่ระบบงานสามารถควบคุมให้ทำการประมวลผลได้ ในการทำงานแบบ Multithreading แต่ละ Process จะถูกแบ่งออกมาเป็นงานอิสระเล็กๆ ที่สามารถประมวลผลได้ภายในตัวเอง

6.2 Disk Arrays

คือ การใช้เทคนิคของการต่อเชื่อม Disk Storage (ฮาร์ดดิสก์หลายๆ ตัว) ให้สามารถทำงานในรูปแบบของ Fault Tolerant (Fault Tolerant มักจะหมายถึงระบบงานที่สามารถทำงานประมวลผลงานให้แก่ผู้ใช้ได้ตลอด ถึงแม้ว่าอุปกรณ์บางส่วนจะเสียหายไปก็ตาม โดยยึดหลักของการมีอุปกรณ์ที่สำคัญสำรองไว้อย่างน้อย 1 ชุด) ได้อย่างมีประสิทธิภาพ ราคาไม่แพง และสามารถทำงานได้อย่างรวดเร็ว ซึ่งเรียกว่า RAID (Redundant Arrays of Inexpensive Disk หรือ Redundant Arrays of Independent Disks) โดยใช้เทคนิคให้ข้อมูล 1 ชุดแตกตัวออกเป็นข้อมูลเล็กๆ หลายๆ ชุด บันทึกลงใน Disk Storage หลายๆ ตัว (ซึ่งทำให้บันทึกข้อมูลได้เร็ว) ในกรณีที่ Disk Storage ตัวใดตัวหนึ่งเสียไป Disk Storage ตัวที่เหลืออยู่ก็สามารถที่จะกู้กลับคืนข้อมูล หรือ สร้าง (Reconstruct) ข้อมูลทั้งชุดขึ้นมาใหม่ได้ (ทำให้ข้อมูลมีความปลอดภัยสูง)

ระบบการประมวลผลแบบ On-Line

(On-Line Transaction Processing System)

ความหมาย

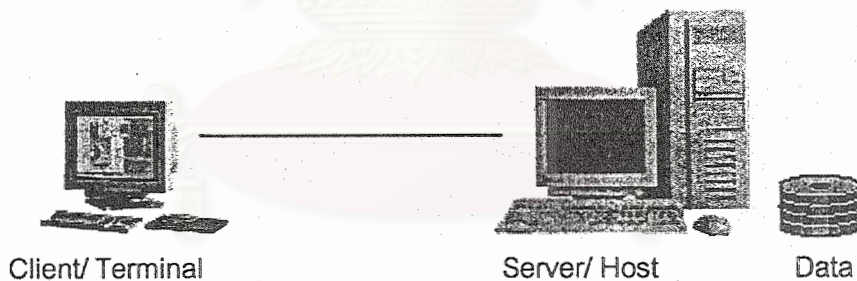
คือ ระบบการทำงานการประมวลผลงาน รับ-ส่งแลกเปลี่ยนข้อมูลข่าวสารระหว่างเครื่องคอมพิวเตอร์ (Client/ Terminal กับ Server/ Host) ในการประมวลผลทางธุรกรรมหนึ่ง

รูปแบบการประมวลผลงานแบบ On-Line

มีผู้ใช้คำว่า "On-Line" ในการออกแบบระบบงานคอมพิวเตอร์มากมาย บางท่านพูดสั้นๆ ว่า "ระบบงาน On-Line" บางท่านพูดโดยใช้ศัพท์ที่ใกล้เคียงกัน ซึ่งก่อให้เกิดความสับสนขึ้นในการสื่อความหมาย ความเข้าใจกันในรูปแบบของระบบงานประยุกต์ทางด้านคอมพิวเตอร์แก่ผู้อื่นๆ ที่เกี่ยวข้องจำนวนมาก ในที่นี้แบ่งรูปแบบการประมวลผลงานประยุกต์ออกเป็น 4 รูปแบบดังนี้

On-Line Interactive System

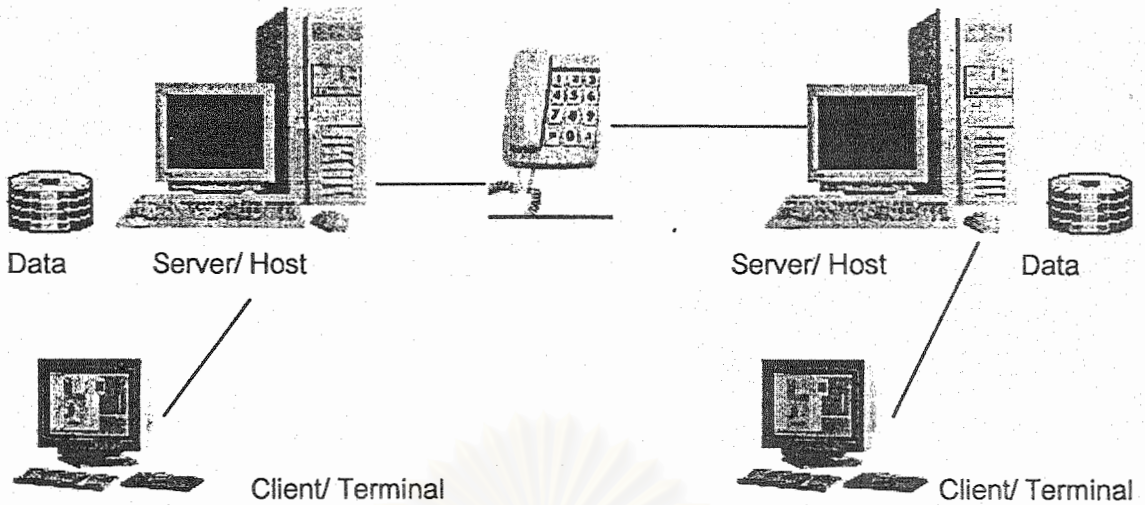
คือ ระบบงานที่ทำการประมวลผลได้ตอบทันทีทันใดระหว่าง Client/ Terminal กับ Server/ Host (ส่วนใหญ่จะเน้นที่เป็นภายในเครือข่ายเดียวกัน) ดังรูปที่ 9



รูปที่ 9 On-Line Interactive System

6.2.1 Off-Line Transaction Processing System

คือ ระบบงานคอมพิวเตอร์ที่ทำการบันทึกข้อมูลที่ Client/ Terminal กับ Server/ Host ที่เครือข่ายใดเครือข่ายหนึ่งก่อน แล้วจึงส่งผ่านข้อมูลเหล่านั้นไป Update ที่ Server/ Host อีกเครือข่ายหนึ่ง ดังรูปที่ 10 หรือบางตำราอาจเรียกว่า Batch File Transfer

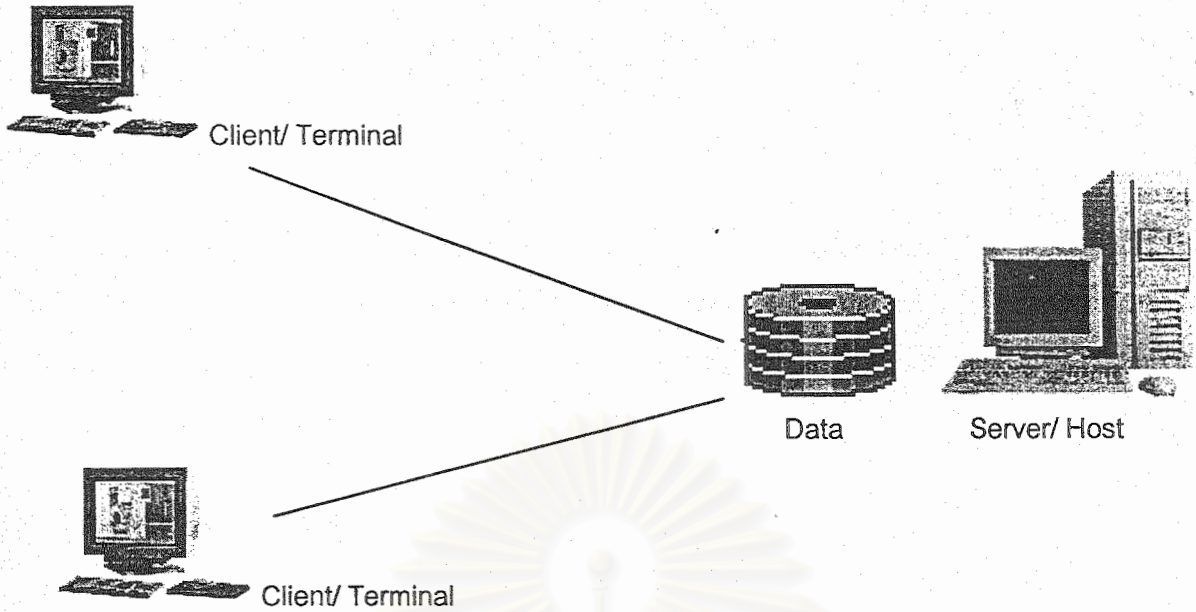


รูปที่ 10 Off-Line Transaction Processing System

เช่นระบบงานที่มีการแลกเปลี่ยนข้อมูลระหว่างห้องที่ที่อยู่ห่างไกลออกไป และสามารถทำงานอย่างอิสระได้ภายในเครือข่ายตัวเอง แต่จำเป็นต้องปรับปรุงข้อมูลกับเครือข่ายอื่นๆ เป็นต้น การลงทุนเครือข่ายในลักษณะนี้ ค่อนข้างถูก และเหมาะสมกับระบบงานที่ไม่จำเป็นต้องปรับปรุงข้อมูลทันทีทันใด โดยระบบงานฯ สามารถทำงานได้ครบกระบวนการในระบบคอมพิวเตอร์ของตนเอง และจะใช้การเชื่อมต่อเครือข่ายเป็นการชั่วคราวกับหน่วยงานอื่นที่เกี่ยวข้องเพื่อรับ-ส่ง แลกเปลี่ยนข้อมูล ข่าวดูสารระหว่างกัน เช่น ในหน่วยงานของราชการไทยจะมีหน่วยงานระดับอำเภอ จังหวัด เขต/ ภูมิภาค กรมฯ และกระทรวงฯ นักพัฒนาระบบงานฯ ต้องพิจารณาลักษณะของระบบงานฯ ว่ามีความจำเป็นต้องรับ-ส่ง ข้อมูลทันทีทันใดหรือไม่ อาจออกแบบให้ระบบงานฯ ทำงานในลักษณะ Standalone ที่อำเภอ และค่อยทำ Batch File Transfer ไปที่จังหวัด และจังหวัดอาจทำ Batch File Transfer ไปที่เขต/ ภูมิภาคอีกทอดหนึ่ง

6.2.2 Batch Processing System

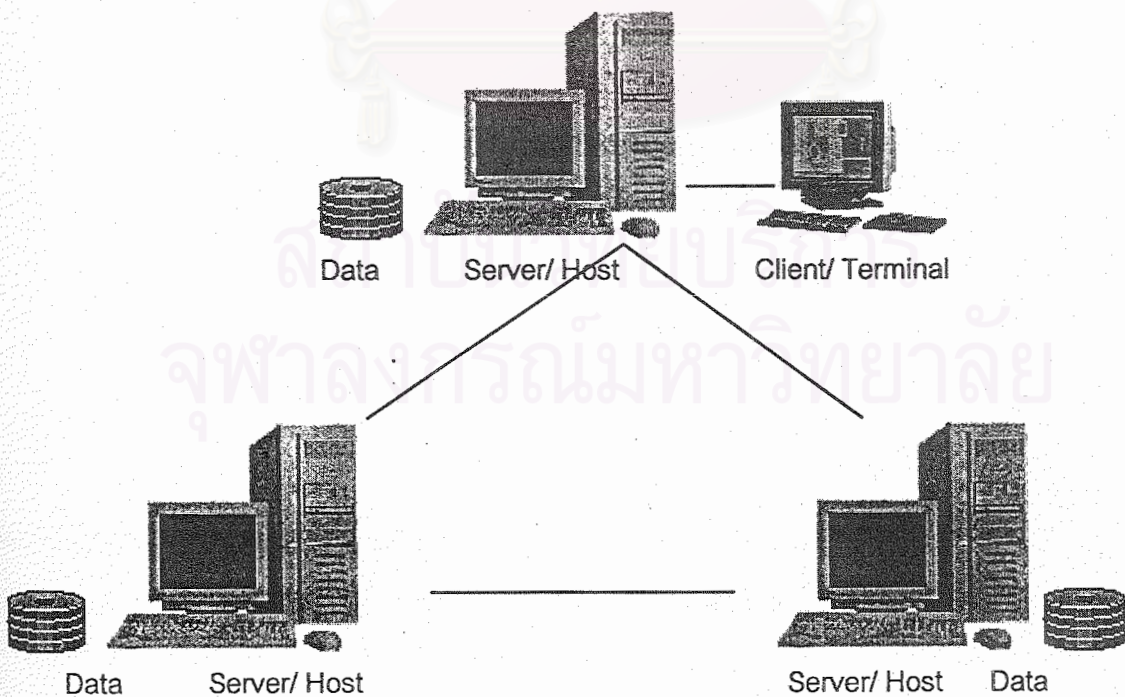
คือ ระบบงานที่บันทึกข้อมูล/ รายการ (Transaction) ไว้จำนวนหนึ่งก่อน แล้วจึงส่งผลข้อมูลเหล่านี้ไป Update ที่ระบบฐานข้อมูลอีกครั้งหนึ่ง (ส่วนใหญ่จะเป็นภายในเครือข่ายเดียวกัน ภายในเครื่องคอมพิวเตอร์เครื่องเดียวกัน หรือเป็นระบบงานเดียวกัน) ดังรูปที่ 11 เช่น ระบบงานบัญชี ที่อาจมีขั้นตอนการตรวจสอบข้อมูลให้ถูกต้อง หรือ Balance กันก่อน แล้วจึงจะ Update/ Post บัญชี ภายหลัง เป็นต้น



รูปที่ 11 Batch Processing System

6.2.3 On-Line Real Time Interactive System

คือ ระบบงานคอมพิวเตอร์ที่มีการประมวลผล และปรับปรุงข้อมูลทันทีทันใดให้แล้วเสร็จสมบูรณ์ที่ระบบฐานข้อมูล ไม่ว่ารายการข้อมูลนั้นจะถูกบันทึกที่ใดก็ตามในเครือข่ายเดียวกัน หรือต่างเครือข่ายก็ตาม ดังรูปที่ 12 เช่น ระบบงานเงินฝาก-ถอน แบบ ATM (Automatic Telling Machine) เป็นต้น



รูปที่ 12 On-Line Real Time Interactive System

Database Concept

ระบบฐานข้อมูล

เนื่องจากข้อมูลข่าวสารเป็นสิ่งที่มีความสำคัญต่อการดำเนินธุรกิจ ดังนั้นเทคนิคในการจัดเก็บข้อมูลจึงเป็นสิ่งสำคัญต่อการพัฒนาระบบ การออกแบบวิธีการเก็บข้อมูลอย่างมีประสิทธิภาพ จะทำให้การพัฒนาโปรแกรมง่ายและมีความรวดเร็วในการนำข้อมูลออกมาแสดงผล ซึ่งในปัจจุบันวิธีการจัดเก็บข้อมูลมี 2 แบบคือ

การจัดเก็บข้อมูลในรูปแบบแฟ้มข้อมูล

การจัดเก็บข้อมูลในรูปแบบนี้เป็นรูปแบบที่ใช้กันมานาน ตั้งแต่การพัฒนาระบบในยุคแรกๆ การจัดเก็บข้อมูลในลักษณะนี้มีข้อดี คือ สะดวก ใช้ทรัพยากรของระบบน้อย และไม่มีความซับซ้อนในการจัดเก็บ แต่มีปัญหาในการใช้งาน คือ

- ความซ้ำซ้อนของข้อมูล (Data Redundancy) - ในการจัดเก็บข้อมูลแบบแฟ้มข้อมูลอาจมีความจำเป็นที่จะต้องเก็บข้อมูลเดียวกันในหลายๆ แฟ้มข้อมูล ผู้ใช้แต่ละคนกำหนดข้อมูลตามที่ตนเองต้องการ
- ความไม่สอดคล้องกันของข้อมูล (Data Inconsistency) - ในกรณีที่มีการเก็บข้อมูลซ้ำซ้อนกัน ในหลายๆ แฟ้มข้อมูล การเปลี่ยนแปลงข้อมูลอาจทำให้ข้อมูลในแต่ละแฟ้มข้อมูลไม่เหมือนกันได้
- การใช้ข้อมูลร่วมกัน (Data Sharing) - การเก็บข้อมูลในรูปแบบแฟ้มข้อมูลจะไม่สามารถใช้ข้อมูลร่วมกันได้ ทุกคนที่ต้องการใช้ข้อมูลต้องมีสำเนา (copy) ของตนเอง
- การขึ้นต่อกันของข้อมูล (Data Dependency) - การจัดเก็บข้อมูลจะขึ้นกับโปรแกรมที่จะมา access ดังนั้นเมื่อเปลี่ยนโปรแกรมที่จะมา access ลักษณะข้อมูลก็จะต้องเปลี่ยนแปลงตาม
- ความปลอดภัยของข้อมูล (Security) - การจัดเก็บในรูปแบบแฟ้มข้อมูลจะมีความปลอดภัยของข้อมูลต่ำ ไม่สามารถสร้าง view สำหรับผู้ใช้แต่ละคนเพื่อให้เลือกมองเพียงบางส่วนได้ ดังนั้นถ้าจำเป็นต้องมีการใช้ข้อมูลบางส่วนในแฟ้มข้อมูลนั้น แต่ไม่ต้องการให้เห็นข้อมูลในส่วนที่เหลือก็ต้องสร้างแฟ้มข้อมูลขึ้นมาใหม่เพื่อความปลอดภัย

การจัดเก็บข้อมูลในรูปแบบฐานข้อมูล

ฐานข้อมูล คือ ข้อมูลที่มีความสัมพันธ์กันถูกเก็บรวบรวมไว้เพื่อใช้โดย application ซึ่งข้อมูลที่จัดเก็บอาจเป็นตัวอักษร ภาพ หรือเสียง โดยมี Database Management System (DBMS) ทำหน้าที่ในการจัดการและติดต่อกับฐานข้อมูล โดยมีจุดประสงค์หลักเพื่อทำให้เกิดความสะดวกและควมมีประสิทธิภาพในการเรียกใช้และจัดเก็บข้อมูลลงในฐานข้อมูล

ข้อดีของการจัดเก็บข้อมูลในรูปแบบฐานข้อมูล

- ลดความซ้ำซ้อนของการจัดเก็บข้อมูล
- ลดความไม่สอดคล้องกันของข้อมูล
- สามารถใช้ข้อมูลร่วมกันได้
- มีมาตรฐานในการจัดเก็บข้อมูล
- สามารถค้นหาข้อมูลได้ง่าย
- มีความปลอดภัยสูง สามารถสร้าง view ให้กับผู้ใช้แต่ละคนให้เข้าถึงเฉพาะข้อมูลที่เขาจำเป็นต้องใช้เท่านั้น
- การจัดเก็บข้อมูลมีความอิสระจากโปรแกรมที่จะมา access

ข้อเสียของการจัดเก็บข้อมูลในรูปแบบฐานข้อมูล

- ใช้ทรัพยากรของระบบมาก ทั้งฮาร์ดแวร์และซอฟต์แวร์
- ค่าใช้จ่ายสูง
- มีความซับซ้อน เพราะต้องคำนึงถึงหลายแง่ในการใช้ข้อมูล
- ความเร็วในการเข้าถึงข้อมูลขึ้นอยู่กับวิธีการออกแบบ ถ้าออกแบบฐานข้อมูลอย่างไม่มีประสิทธิภาพ การเข้าถึงข้อมูลจะใช้เวลานาน

ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System)

ฐานข้อมูลเชิงสัมพันธ์ เป็นฐานข้อมูลรูปแบบหนึ่งที่ยอมรับในปัจจุบัน เนื่องจากเป็นฐานข้อมูลที่เข้าใจง่าย ผู้ใช้ทั่วไปสามารถมองภาพของข้อมูลที่ถูกเก็บได้ง่ายเพราะข้อมูลถูกจัดเก็บในลักษณะของตารางสองมิติ คือเป็นแถว (Row) และคอลัมน์ (Column) เป็นรูปแบบฐานข้อมูลที่ไม่ซับซ้อนและมีเครื่องมือช่วยในการเรียกใช้ข้อมูลโดยใช้คำสั่งง่ายๆ เช่น SQL เป็นต้น ด้วยเหตุนี้ ระบบจัดการฐานข้อมูล (Database Management Systems: DBMS) ที่นิยมใช้กันอย่างแพร่หลายในปัจจุบันก็เป็นระบบที่สนับสนุนฐานข้อมูลประเภทนี้ค่อนข้างมาก เช่น DB2, Oracle, Sybase, Ingress เป็นต้น

คำศัพท์ที่เกี่ยวข้อง

- ตาราง (Table) หรือ ความสัมพันธ์ (Relation) เป็นคำที่ใช้แทนเรื่องใดเรื่องหนึ่ง เช่น ตารางข้อมูลพนักงาน, ตารางข้อมูลแผนก, ตารางข้อมูลโครงการ, ตารางข้อมูลงานที่รับมอบหมาย โดยตารางมีคุณสมบัติดังต่อไปนี้
 - ไม่มีแถวที่ซ้ำกันใน 1 relation
 - แถวไม่จำเป็นต้องเรียงลำดับกัน
 - คอลัมน์ไม่จำเป็นต้องเรียงลำดับกัน

➤ ค่าของคอลัมน์ใดๆ จะเป็นค่าที่แยกไม่ได้

- คอลัมน์ (Column) หรือแอททริบิวต์ (Attribute) คือรายละเอียดของข้อมูลในแต่ละตารางที่แสดงให้เห็นว่าตารางหนึ่งๆ ประกอบไปด้วยรายละเอียดอะไรบ้าง
- แถว (Row) หรือ ทูเพิล (Tuple) คือ ค่าข้อมูลแต่ละแถวในแต่ละตาราง
- โดเมน (Domain) คือ ขอบเขตหรือค่าที่ควรจะเป็นของข้อมูลในคอลัมน์หนึ่งๆ
- คีย์ แบ่งออกเป็น คีย์หลัก และ คีย์นอก ดังรายละเอียดในหัวข้อ 2.2
- เอนทิตี (entity) คือการรวบรวมสิ่งที่เกี่ยวข้องมาเก็บเป็นหมวดหมู่เดียวกันอย่างเช่นในการเก็บประวัตินักศึกษา และสถานที่ศึกษา ก็จะประกอบไปด้วย 2 entity คือ นักศึกษา กับ สถานที่ศึกษา จนมีบางคนสรุปว่า entity คือค่านามในประโยคนั้นเอง

ประเภทของคีย์

เค้าร่างของตาราง (Table/ Relation Schema) ประกอบด้วยรายละเอียดข้อมูลของคอลัมน์ต่างๆ ซึ่งคอลัมน์ใดคอลัมน์หนึ่งในตารางมีโอกาสที่จะมีคุณสมบัติเป็นคีย์ โดยคีย์นั้นๆ สามารถใช้ในการแสดงหรือบ่งบอกถึงค่าของข้อมูลที่เหลือในแต่ละแถว หรือใช้ในการเชื่อมโยงข้อมูลกับข้อมูลกับอีกตารางหนึ่ง ประเภทของคีย์ที่จะกล่าวในที่นี้ แบ่งออกเป็น 2 ประเภทหลัก คือ

- คีย์หลัก (Primary Key)

เป็น Attribute ที่มีคุณสมบัติของข้อมูลที่เป็นค่าเอกลักษณ์หรือมีค่าที่ไม่ซ้ำกัน คุณสมบัติดังกล่าวจะสามารถระบุว่าคุณข้อมูลนั้นเป็นข้อมูลของ Tuple ใด เช่น รหัสพนักงาน เลขที่ 3001 สามารถระบุได้ว่าเป็นของพนักงานชื่อ ดวงพร Attribute ที่มีคุณสมบัติเป็น คีย์หลักอาจประกอบด้วยหลาย Attribute รวมกันเรียกว่า Composite Key (คีย์ผสม)นอกจากนี้ ใน Relation หนึ่งๆ อาจมี Attribute ที่มีคุณสมบัติเป็นคีย์หลักได้มากกว่า หนึ่ง Attribute เรียก Attribute เหล่านี้ว่า Candidate Key (คีย์คู่แข่ง)ถ้า Attribute หนึ่งถูกกำหนดให้เป็นคีย์หลัก อีก Attribute หนึ่งที่มีคุณสมบัติเป็นคีย์หลัก แต่ไม่ได้ถูกเลือกให้เป็นคีย์หลักจะเรียกว่า คีย์สำรอง (Alternate Key)

- คีย์นอก (Foreign Key)

เป็น Attribute ใน Relation หนึ่งที่ใช้อ้างอิงถึง Attribute เดียวกันในอีก Relation หนึ่ง โดยที่ Attribute นี้มีคุณสมบัติ เป็นคีย์หลักใน Relation ที่ถูกอ้างอิง การมี Attribute นี้ปรากฏอยู่ในRelation ทั้งสองก็เพื่อประโยชน์ ในการเชื่อมโยงข้อมูล

กฎที่ใช้กับฐานข้อมูลเชิงสัมพันธ์

เนื่องจากตารางต่างๆ ในฐานข้อมูลเชิงสัมพันธ์จะอ้างอิงความสัมพันธ์โดยใช้คีย์ กฎที่เกี่ยวข้องกับคีย์ในฐานข้อมูลเชิงสัมพันธ์มีดังต่อไปนี้

- **กฎความบูรณภาพของเอนทิตี (The Entity Integrity Rule)**

กฎนี้ระบุไว้ว่าแอททริบิวต์ใดที่เป็นคีย์หลัก ข้อมูลในแอททริบิวต์นั้นจะเป็นค่าว่าง(Null) ไม่ได้ ความหมายของการเป็นค่าว่างไม่ได้(Not Null) หมายความว่า ข้อมูลของแอททริบิวต์ที่เป็นคีย์หลักจะไม่ทราบค่าที่แน่นอนหรือไม่มีค่าไม่ได้

- **กฎความบูรณภาพของการอ้างอิง (The Referential Integrity Rule)**

การอ้างอิงข้อมูลระหว่างรีเลชันในฐานข้อมูลเชิงสัมพันธ์จะใช้คีย์นอกของรีเลชันหนึ่งไปตรวจสอบกับค่าของแอททริบิวต์ที่เป็นคีย์หลักของอีกรีเลชันหนึ่งเพื่อเรียกดูข้อมูลอื่นๆที่เกี่ยวข้องหรือค่าของคีย์นอกจะต้องอ้างอิงให้ตรงกับค่าของคีย์หลักได้จึงจะสามารถเชื่อมโยงข้อมูลระหว่างสองรีเลชันได้สำหรับคีย์นอกจะมีค่าว่างได้หรือไม่ขึ้นอยู่กับกฎเกณฑ์การออกแบบฐานข้อมูล เช่น ในกรณีที่รีเลชันพนักงานมี Depno เป็นคีย์นอกอาจจะถูกระบุว่าต้องทราบค่าแต่ในกรณีพนักงานทดลองงานอาจยังไม่มีค่า Depno เพราะยังไม่ได้ถูกบรรจุในกรณีที่มีการลบหรือแก้ไขข้อมูลของแอททริบิวต์ที่เป็นคีย์หลักซึ่งมีคีย์นอก จากอีกรีเลชันหนึ่งอ้างอิงถึง จะทำการลบหรือแก้ไขข้อมูลได้หรือไม่ ขึ้นอยู่กับการออกแบบฐานข้อมูล ว่าได้ระบุให้แอททริบิวต์มีคุณสมบัติอย่างไร ซึ่งมีโอกาสเป็นไปได้ 4 ทางเลือก

- **การลบหรือแก้ไขข้อมูลแบบมีข้อจำกัด (Restrict)** การลบหรือแก้ไขข้อมูลจะกระทำไม่ได้ เมื่อข้อมูลของคีย์หลักในรีเลชันหนึ่งไม่มีข้อมูลที่ถูกต้องอ้างอิง โดยคีย์นอกของอีกรีเลชันหนึ่งเช่น รหัสแผนก Depno ในรีเลชันDep จะถูกแก้ไขหรือลบทิ้งต่อเมื่อไม่มีพนักงานคนใดสังกัดอยู่ในแผนกนั้น
- **การลบหรือแก้ไขข้อมูลแบบต่อเรียง (Cascade)** การลบหรือการแก้ไขข้อมูล จะทำแบบเป็นลูกโซ่ คือ หากมีการแก้ไขหรือลบข้อมูลของคีย์หลักในรีเลชันหนึ่งระบบจะทำการลบหรือแก้ไขข้อมูลของคีย์นอกในอีกรีเลชันหนึ่งที่อ้างอิงถึงข้อมูลของคีย์หลักที่ถูกลบให้ด้วย เช่น ในกรณีที่ยกเลิกแผนก 9 ใน Entity แผนก ข้อมูลของพนักงานที่อยู่แผนก 10 ใน Entity พนักงานจะถูกลบออกไปด้วย

- การลบหรือแก้ไขข้อมูลโดยเปลี่ยนเป็นค่าว่าง (Nullify) การลบหรือแก้ไขข้อมูลจะทำได้เมื่อมีการเปลี่ยนค่าของคีย์นอกในข้อมูลที่ถูกอ้างอิงให้เป็นค่าว่างเสียก่อน เช่น พนักงานที่อยู่ในแผนกที่ 9 จะถูกเปลี่ยนค่าเป็นค่าว่างก่อนหลังจากนั้น การลบข้อมูลของแผนกที่มีรหัส 9 จะถูกลบทิ้งหรือแก้ไขทันที ภายใน Entity แผนก
- การลบหรือแก้ไขข้อมูลแบบใช้ค่าโดยปริยาย (Default) การลบหรือแก้ไขข้อมูลของคีย์หลัก สามารถทำได้โดยถ้าหากมีคีย์นอกที่อ้างอิงถึงคีย์หลักที่ถูกลบหรือแก้ไข ก็จะทำการปรับค่าของคีย์นอกนั้นโดยปริยาย (Default Value) ที่ถูกกำหนดขึ้นเช่น ในกรณีที่ยกเลิกแผนก 9 ใน Entity แผนก ข้อมูลของพนักงานที่อยู่แผนก 9 ใน Entity พนักงานจะถูกเปลี่ยนค่าเป็น 00 ซึ่งเป็นค่าโดยปริยาย ที่หมายความว่า ไม่ได้สังกัดแผนกใด

ภาษาทางด้านฐานข้อมูล (Query Language) ผลิตภัณฑ์ทางด้านฐานข้อมูลที่มีโครงสร้างข้อมูลแบบ Relation (Rela-Lation Model) จำเป็นอย่างยิ่งที่จะต้องมีภาษาทางด้านฐานข้อมูล เช่น ภาษา SQL (Structure Query Language) ภาษา QBE (Qurey by Exaample) และภาษา Quel ฯลฯ เป็นต้น ภาษาเหล่านี้ได้ถูกพัฒนาขึ้นจากแนวคิดที่ต่างกัน เช่น ภาษา QBE ซึ่งถูกพัฒนาขึ้นจากแนวคิดของ Relational Calculus ส่วนภาษา Quel ถูกพัฒนาขึ้นจากแนวคิดของ Tuple Relational Calculus และ Relational Algebra เป็นหลักแต่อย่างไรก็ตาม ภาษาที่ได้รับความนิยมมากที่สุดคือภาษา "SQL"

Structured Qurey Language ได้ เริ่มต้นพัฒนาครั้งแรกโดย San Jose Research Laboratory (ปัจจุบันเปลี่ยนชื่อเป็น Almaden Res-earch Center) ของบริษัท IBM โดยมีชื่อแรกเริ่มว่า "Sequal" ซึ่งเป็นงานวิจัยในโครงการ R ในต้นทศวรรษ 1970 และได้ถูกนำมาใช้เป็นต้นแบบของภาษา SQL ของผลิตภัณฑ์ทางด้านฐานข้อมูลจำนวนมาก แต่อย่างไรก็ตาม ภาษา SQL ของแต่ละผลิตภัณฑ์ยังคงมีข้อแตกต่างกันในรายละเอียดทางด้านการใช้งาน ดังนั้นในปี ค.ศ.1986 ทางด้าน American National Standards Institute (ANSI) จึงได้กำหนดมาตรฐานของ SQL ขึ้นรวมทั้งบริษัท IBM ได้กำหนดมาตรฐานของตัวเองขึ้นมาเช่นเดียวกันโดยมีชื่อว่า Systems Application Architecture Database Interface (SAA-SQL) ซึ่งต่อมาทั้ง 2 มาตรฐานนี้ได้เป็นมาตรฐานในการผลิตภาษา SQL ของแต่ละบริษัท ส่วนคำสั่งต่างๆ ของภาษา SQL สามารถแบ่งตามลักษณะการใช้งานออกเป็น 3 กลุ่ม ดังนี้

DDL (Data Definition Language)

เป็นกลุ่มคำสั่งที่ใช้สำหรับสร้างฐานข้อมูล หรือใช้กำหนดโครงสร้างให้กับ Relation ภายในฐานข้อมูลเช่น การเพิ่ม เปลี่ยนแปลง ลบ Attribute ของ Relation ฯลฯ เป็นต้น

DML (Data Manipulation Language)

เป็นกลุ่มคำสั่งที่พัฒนาขึ้นตามแนวคิดของ Relational Algebra และ Record Relational Calculus โดยประกอบด้วยคำสั่งที่ใช้สำหรับ เพิ่ม ลบ หรือเปลี่ยนแปลงข้อมูลในฐานข้อมูล

Data Query Language

เป็นกลุ่มคำสั่ง DML ประเภทหนึ่ง ที่ใช้ในการเลือกข้อมูลจาก Relation ขึ้นมาแสดงผล ตามรูปแบบที่ต้องการ



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ขั้นตอนการทำการออกแบบฐานข้อมูล

1. ทำความเข้าใจระบบงานให้ชัดเจน

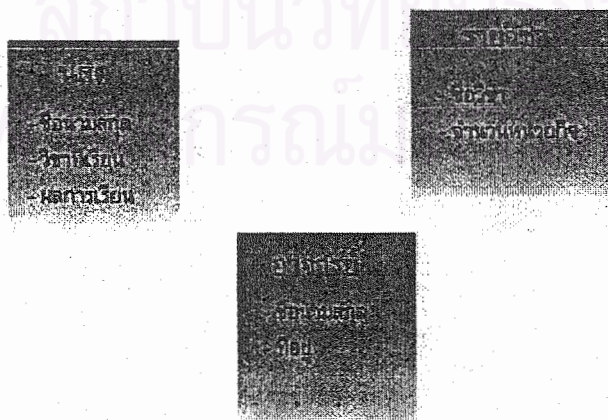
ขั้นตอนที่สำคัญมากที่สุดขั้นตอนหนึ่ง ที่จะต้องมองให้ชัดเจนทั้งในภาพรวมและภาพย่อย ทั้งของกลุ่มข้อมูลที่มีและไม่มีความสัมพันธ์

2. ร่างความสัมพันธ์คร่าว ๆ โดยแบ่งออกเป็นกลุ่ม ๆ

การกำหนด Entity ของเรื่องที่เราศึกษาออกเป็นกลุ่ม ตัวอย่างเช่น ถ้าเราต้องการทำการออกแบบฐานข้อมูลของฝ่ายทะเบียนนิสิต โดยมีข้อมูลดังต่อไปนี้

- ชื่อนามสกุลนิสิต
- ภาควิชาของนิสิตแต่ละคน
- อาจารย์ที่ปรึกษา
- ชื่อวิชา
- ที่อยู่นิสิต
- เบอร์โทรศัพท์อาจารย์ที่ปรึกษา
- จำนวนหน่วยกิตของนิสิต
- ผลการเรียนของนิสิตแต่ละวิชา

จากข้อมูลทั้งหมดที่กล่าวมาจะสามารถแบ่งออกเป็นกลุ่มได้ดังต่อไปนี้



รูปที่ 13 แสดงให้เห็นการแบ่งกลุ่มข้อมูล

โดยแต่ละกลุ่มข้อมูลเราจะมีศัพท์เฉพาะเรียกว่า Entity นั้นเอง

3. กำหนดความสัมพันธ์และคีย์หลักของข้อมูล

หลังจากที่ทำการกำหนด Entity ได้แล้วลำดับต่อมาให้ทำการใส่ฟิลด์ที่ต้องการให้ครบ โดยแยกในแต่ละกลุ่ม แล้วทำการกำหนดความสัมพันธ์ ระหว่างกลุ่ม อย่างเช่น ความสัมพันธ์ระหว่างอาจารย์กับรายวิชาคือ หนึ่งวิชามีอาจารย์สอนได้หนึ่งคน แต่อาจารย์หนึ่งคนสอนได้หลายวิชา ดังนั้นจึงจำเป็นต้องเพิ่มฟิลด์ ชื่ออาจารย์ไว้ใน Entity รายวิชาด้วย และวิชาที่เรียนใน Entity นิสิต ก็也将มีความสัมพันธ์กับชื่อวิชาใน Entity รายวิชา และทำการกำหนดคีย์หลักให้กับแต่ละ Entity โดยคีย์หลักจะเป็นข้อมูลที่เมื่อเราพูดถึงฟิลด์นั้นแล้วเราจะทราบทันทีว่าหมายถึงข้อมูลในแถวไหน ดังนั้นถ้าพิจารณาจาก Entity นิสิต เราจะพบว่า ชื่อและนามสกุลสามารถเป็นคีย์หลักของ Entity นี้ได้ก็จริง แต่บางครั้งก็มีโอกาสที่จะมีชื่อหรือนามสกุลซ้ำกันได้เช่นกัน และโดยปกติทั่วไปแล้ว เมื่อเราได้มีโอกาสไปเขียนโปรแกรมเกี่ยวกับฐานข้อมูลแล้วเกิดต้องมีการเรียกข้อมูลหลาย ๆ ตารางพร้อมกันแล้ว การเลือกคีย์หลักเป็นฟิลด์ที่มีลักษณะเหมือน Description จะไม่ค่อยดีนัก เพราะจะทำให้ประสิทธิภาพในการค้นหาข้อมูล และการเชื่อมต่อระหว่างข้อมูลไม่ดี ดังนั้นจึงมักจะมีการกำหนดรหัสไว้ในแต่ละตาราง ตัวอย่างเช่น Entity นิสิตก็จะมีรหัสนิสิตเป็นคีย์หลัก Entity รายวิชา ก็จะมีรหัสวิชาเป็นคีย์หลัก และ Entity อาจารย์ก็จะมีรหัสอาจารย์เป็นคีย์หลักในกรณีที่อาจารย์หนึ่งคนสอนหนึ่งวิชา เป็นต้น

ประเภทของความสัมพันธ์จะสามารถแบ่งออกได้เป็น 3 ประเภท ดังต่อไปนี้คือ

- ความสัมพันธ์แบบ One-to-One หมายถึงว่าจากข้อมูลจากตาราง A มีความเกี่ยวข้องกับตาราง B แค่แถวเดียว
- ความสัมพันธ์แบบ Many-to-One หรือ One-to-Many หมายถึงว่าข้อมูลจากตาราง A มีความเกี่ยวข้องกับตาราง B หลายแถว หรือ ข้อมูลจากตาราง B หลายแถวมีความเกี่ยวข้องกับตาราง A แถวเดียว
- ความสัมพันธ์แบบ Many-to-Many หมายถึงว่าข้อมูลจากตาราง A หลายแถวเกี่ยวข้องกับข้อมูลจากตาราง B หลายแถว ซึ่งความสัมพันธ์แบบนี้จะมีปัญหาที่สุด ดังนั้นจึงควรหลีกเลี่ยง

รูปแบบบรรทัดฐาน (Normalization)

แนวคิดในการทำรีเลชันให้อยู่ในรูปแบบบรรทัดฐานถูกคิดค้นโดย อี.เอฟ.คอตต์เป็นกระบวนการที่นำเค้าร่างของรีเลชันมาทำให้อยู่ในรูปแบบบรรทัดฐาน(normal form)

วัตถุประสงค์ของการทำให้เป็นบรรทัดฐาน มีดังนี้

1. เพื่อลดเนื้อหาในการเก็บข้อมูลการทำให้เป็นบรรทัดฐานเป็นการลดความซ้ำซ้อนของข้อมูลในรีเลชัน
2. เพื่อลดปัญหาที่ข้อมูลไม่ถูกต้อง(Inconsistency)เนื่องจากข้อมูลในรีเลชันหนึ่งจะมีข้อมูลไม่ซ้ำกันเมื่อมีการปรับปรุงข้อมูลก็จะปรับปรุงทุกฟิลด์นั้นๆครั้งเดียวไม่ต้องปรับปรุงหลายแห่ง
3. เป็นการลดปัญหาที่เกิดจากการเพิ่ม ปรับปรุงและลบข้อมูล

ข้อมูลที่ซ้ำซ้อนกันนี้จะก่อให้เกิดปัญหาน้อยอย่างน้อย 2 ประการ คือ

1. ปัญหาความผิดพลาดของข้อมูล เช่น การที่ลูกค้าเปลี่ยนชื่อหรือในกรณีที่บริษัทมีการเปลี่ยนแปลงรายละเอียดสินค้า เราจะต้องทำการแก้ไขข้อมูลให้ครบทุกเรคคอร์ดในตาราง มิฉะนั้นข้อมูลในบางเรคคอร์ดจะเกิดความผิดพลาดได้
2. เปลืองเนื้อที่ในการจัดเก็บข้อมูลเพราะจะต้องจัดเก็บข้อมูลเดียวกันไว้ในหลาย ๆ เรคคอร์ด

หลักการทำ Normalization

หลักการทำ Normalization สิ่งสำคัญคือ "การลดความซ้ำซ้อนและโอกาสที่จะเกิดความผิดพลาดกับข้อมูลได้" ซึ่ง การที่จะทำให้บรรลุจุดประสงค์ดังกล่าวจะต้องมีเกณฑ์และขั้นตอนในการวิเคราะห์ข้อมูลโดยทั่วไปเราต้องรู้ก่อนว่าแต่ละตารางมี field ไตบ้างสามารถปงชี้หรือค้นหาข้อมูลได้ เช่น เมื่อทราบรหัสลูกค้า จะ ทำให้สามารถค้นหา ชื่อ,นามสกุล,ที่อยู่ ฯลฯ ได้ สำหรับเกณฑ์เหล่านี้เราจะเรียกว่า "Functional Dependency" (FD) ใช้สัญลักษณ์ \rightarrow แทนการกำหนดค่าระหว่าง field คุณสมบัติที่สำคัญอีกประการของการทำ Normalization คือ เมื่อตารางใดจัดอยู่ใน Normal Form ไต แล้วจะต้องมีคุณสมบัติของ Normal Form ที่ต่ำกว่าเสมอ เช่น ถ้าตารางใดเป็น 3N จะต้องมิตุสมบัติของ 1N และ 2N อยู่ด้วย ในที่นี้จะไม่กล่าวถึง ระดับที่เหลืออยู่คือ BCNF, 4NF, และ 5NF

1NF(First Normal Form)

ตารางใดจะถือว่าอยู่ใน 1 NF หรือไม่จะพิจารณาทุก Fields ในตารางว่าจะต้อง "ไม่มี Fields ใด Fields หนึ่งที่ลักษณะเป็น Multivalued (Field เดียวแต่เก็บหลายๆ ค่าไว้ด้วยกัน) เช่นตารางที่ 4

Car Table			
CID	Brand	Model	Color
1ข-2776	Benz	E220	ขาว, เหลือง
9ค-8412	Misubishi	Lacer	แดง, ขาว,
2ก-5266	Toyota	Corolla	เหลือง, เทา

ตารางที่ 4 แสดงตารางตัวอย่างที่มี Multivalued

จากตัวอย่างดังกล่าวจะเห็นได้ว่า Field "Color" จะเป็นลักษณะของ Multivalued ซึ่งให้แก้ไขโดยแบ่ง Field ดังกล่าวออกเป็นอีกตาราง และตั้งเขา Primary Key ของตารางนั้นมาด้วยดังตารางที่ 5

Car Table		
CID	Brand	Model
1ข-2776	Benz	E220
9ค-8412	Misubishi	Lacer
2ก-5266	Toyota	Corolla

Color Table	
CID	Color
1ข-2776	ขาว
1ข-2776	เหลือง
9ค-8412	แดง
2ก-5266	เทา
9ค-8412	ขาว

ตารางที่ 5 แสดงการแบ่งตาราง ที่ 4 ออกเป็นสองตารางย่อย ที่ไม่มี Multivalued

2NF(Second Normal Form)

การทำ 2 NF เน้นการวิเคราะห์ฟิลด์ที่เป็น Primary Key โดยปกติแล้ว Primary Key ของตารางหนึ่งๆ อาจประกอบด้วย Field เพียง Field เดียวหรือหลาย Fields รวมอยู่ด้วยกันก็ได้โดยมีหลักว่า "หากมี

ตารางใดที่ประกอบด้วย Field เพียง Field เดียวจะถือว่าตารางนั้นอยู่ใน 2NF" ส่วนตารางใดที่มีหลาย

ตารางรายการขาย (InvoiceDetail Table)											
InvNo	InvDt	Pid	Pname	Brand	Model	Qu	Cid	Title	Fname	Lname	Tel
101	10/7/00	M011	หม้อหุงข้าว	Hitachi	CX98	20	C01	นาย สุรสิทธิ์	รักล้น		522-1622
101	10/7/00	F002	ตู้เย็น	Whirlpool	AB77	5	C01	นาย สุรสิทธิ์	รักล้น		522-1622
101	10/7/00	F003	ตู้เย็น	National	CC87	10	C01	นาย สุรสิทธิ์	รักล้น		522-1622
102	11/7/00	F002	ตู้เย็น	Whirlpool	AB77	15	C03	นาย สมศักดิ์	มักรวย		121-1111
102	11/7/00	F005	ตู้เย็น	Whirlpool	CC87	30	C03	นาย สมศักดิ์	มักรวย		121-1111
103	11/7/00	M011	หม้อหุงข้าว	Hitachi	CX98	2	C01	นาย สุรสิทธิ์	ทองมาก		252-1111
104	12/7/00	F002	ตู้เย็น	Whirlpool	AB77	5	C02	นส. การุณ	ทองสุข		254-1545

Fields รวมกันเป็น Primary Key ให้แยกเอา Fields นั้นไปสร้างตารางใหม่ ดังตารางที่ 6
ตารางที่ 6 ตารางตัวอย่างภายหลังจากการทำ 2NF

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

3NF (Third Normal Form)

หลักการทำให้ 3NF คือ การที่จะต้องไม่มีฟิลด์ใดในตาราง นอกจาก Primary Key ที่สามารถไปกำหนดฟิลด์อื่น ดังตัวอย่างในขั้นตอน 2NF จะเห็นได้ว่ารหัสลูกค้า (Cid) ไม่ได้เป็น Primary Key ของตารางแต่สามารถไปกำหนด ชื่อ,นามสกุล,ค่านำหน้า และเบอร์โทรศัพท์ ดังนั้นตารางรายการการขาย (Invoice Detail Table) จึงไม่อยู่ในรูป 3NF

วิธีแก้ไข คือ แยกเอาฟิลด์ต่างๆ ที่ถูกกำหนดโดยฟิลด์ที่ไม่ใช่ Primary Key นั้นออกมาอีกตารางหนึ่งและตั้งเอาฟิลด์เป็นตัวกำหนดนั้นมาเป็นส่วนร่วมในตารางใหม่นี้ด้วย ดังนั้นฟิลด์ Title, Fname, Lname, Tel, Cid จึงถูกแยกออกมาเป็นตารางลูกค้า (Customer Table) ดังตารางที่ 7

ตารางรายการขาย (Invoice Detail Table)

Pid	Pname	Brand	Model	Price
M011	หม้อหุงข้าว	Hitachi	CX98	2500
F002	ตู้เย็น	Whirlpool	AB77	8000
F003	ตู้เย็น	National	CC87	6800

ตารางใบกำกับสินค้า (Invoice Table)

InvNo	InvDt	Cid
101	10/7/00	C01
102	11/7/00	C03
103	11/7/00	C01

ตารางรายการขาย (Invoice Detail Table)

InvNo	Pid	Qu
101	M011	20
101	F002	5
101	F003	10
102	F002	15
102	F005	30
103	M011	2

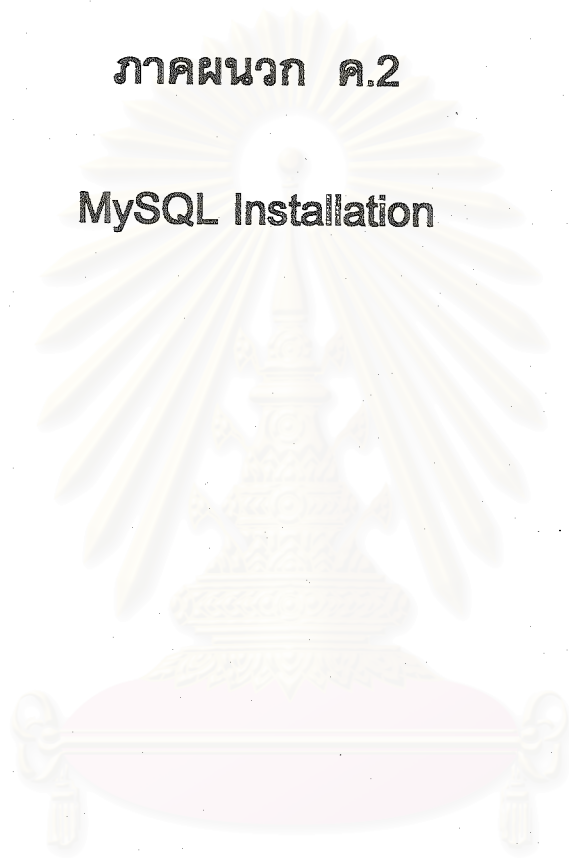
ตารางรายการขาย (Invoice Detail Table)

Cid	Title	Fname	Lname	Tel
C01	นาย	สุรสิทธิ์	รักสัน	522-1622
C03	นาย	สมศักดิ์	มังกรวย	435-7207
C01	นาย	สุรสิทธิ์	ทองมาก	252-1111
C02	นส	การณ	ทองสุข	254-1545

ตารางที่ 7 ตารางตัวอย่างภายหลังจากการทำ 3NF

ภาคผนวก ค.2

MySQL Installation



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

1.	บทที่ 1 Database Management System.....	1
1.1.	ความหมายของระบบฐานข้อมูล.....	1
1.2.	ข้อดีในการใช้โปรแกรมระบบจัดการฐานข้อมูล	1
1.3.	Relational Database	1
2.	บทที่ 2 ระบบฐานข้อมูลแบบ Client/Server.....	2
2.1.	ข้อดี ของระบบฐานข้อมูลแบบ Client/Server	2
2.2.	ข้อเสียของระบบ Client/Server.....	3
3.	บทที่ 3 เกี่ยวกับ MySQL.....	4
4.	บทที่ 4 ระบบการจัดการกับสิทธิการเข้าถึงข้อมูล.....	5
4.1	ความรู้พื้นฐานเกี่ยวกับ user-name และ รหัสผ่าน ใน MySQL.....	5
4.2	ความรู้พื้นฐานเกี่ยวกับการ connect MySQL server	5
4.3	ความรู้พื้นฐานเกี่ยวกับ Privileges system	5
4.4	ความรู้พื้นฐานเกี่ยวกับ Privileges table.....	6
4.5	ระบบรักษาความปลอดภัยทั่วไป.....	7
4.6	การป้องกัน MySQL ให้ปลอดภัยจาก crackers	8
4.7	เมื่อสิทธิของผู้ใช้ถูกเปลี่ยนแปลง จะมีผลต่อผู้ใช้เมื่อใด ?.....	8
4.8	การ set up สิทธิการใช้ MySQL เริ่มต้น	9
4.9	การเพิ่มผู้ใช้ และ การจัดการกับสิทธิของผู้ใช้.....	9
4.10	วิธีการกำหนด passwords.....	11
5.	บทที่ 5 Language Reference.....	13
5.1.	วิธีการเขียนสตริง และ ตัวเลข.....	13
5.1.1.	สตริง	13
5.1.2.	ตัวเลข	14
5.1.3.	เลขฐานสิบหก.....	14
5.1.4.	NULL value	14
5.1.5.	Database, table, index, column, alias names	15
5.1.6.	Case sensitivity in names.....	15
5.2.	ชนิดของข้อมูลคอลัมน์.....	16
5.2.1.	ข้อมูลชนิดตัวเลข	16
5.1.2.	ข้อมูลชนิดวันที่ และ เวลา	17
5.1.3.	ข้อมูลชนิดสตริง.....	17
5.3.	Functions for use in SELECT and WHERE clauses.....	18
5.1.1.	ฟังก์ชันเกี่ยวกับสตริง.....	33
5.1.2.	ฟังก์ชันเกี่ยวกับวันที่ และ เวลา.....	38
5.4.	CREATE DATABASE syntax.....	41
5.5.	DROP DATABASE syntax	42

5.6.	CREATE TABLE syntax.....	42
5.7.	ALTER TABLE syntax.....	44
5.8.	DROP TABLE syntax.....	46
5.9.	DELETE syntax.....	46
5.10.	SELECT syntax.....	46
5.11.	JOIN syntax.....	48
5.12.	INSERT syntax.....	49
5.13.	REPLACE syntax.....	50
5.14.	LOAD DATA INFILE syntax.....	51
5.15.	UPDATE syntax.....	52
5.16.	USE syntax.....	52
5.17.	SHOW syntax.....	53
5.18.	DESCRIBE syntax.....	53
5.19.	GRANT และ REVOKE syntax.....	54
5.20.	CREATE INDEX syntax.....	55
5.21.	DROP INDEX syntax.....	55
5.22.	COMMENT syntax.....	56
5.23.	MySQL reserved words.....	56
6.	บทที่ 6 MyODBC.....	59
6.1	การ install MyODBC บน Windows.....	59
6.2	การสร้าง DSN ของ ODBC ได้หลาย DSN เพื่อใช้ใน application ต่างกัน.....	60
6.3	การใช้ Microsoft Access เชื่อมต่อกับ MySQL.....	61
6.3.2.	import/ export table.....	61
7.	บทที่ 7 วิธีใช้งาน MySQL เบื้องต้น.....	62
7.1	การ connect และ disconnect MySQL server.....	62
7.2	การพิมพ์คำสั่งเพื่อดึงข้อมูลจาก database (Entering Queries).....	62
7.3	ตัวอย่างของการ query.....	64
7.3.1	การสร้างตาราง.....	64
7.3.2	การสร้าง และ ใช้งาน database.....	64
7.4	การ Load ข้อมูลจากไฟล์ ลงไปใน ตาราง.....	65
7.5	การดึงข้อมูลออกมาจากราย.....	66
7.5.1	เลือกข้อมูลทั้งตาราง.....	66
7.5.2	เลือกข้อมูลเฉพาะบางแถว (บาง records).....	66
7.5.3	เลือกข้อมูลเฉพาะบางคอลัมน์.....	67
7.6	การใช้ Pattern matching.....	68
7.7	การนับจำนวนแถวของข้อมูลที่ถูก select.....	69

1. บทที่ 1 Database Management System

1.1. ความหมายของระบบฐานข้อมูล

ระบบฐานข้อมูลคือ โครงสร้างสารสนเทศ (Information) ที่ประกอบด้วยข้อมูลต่างๆ เช่น ข้อมูลของลูกค้าของบริษัท ข้อมูลรายการสินค้า เป็นต้น ซึ่งข้อมูลเหล่านี้สามารถรวมเป็นหัวข้อที่สัมพันธ์กันได้ ระบบฐานข้อมูลได้สร้างวิธีการสำหรับการรวบรวมรายการ ข้อมูลที่เกี่ยวข้องกับงานนั้นๆ เข้าด้วยกัน สร้างวิธีสำหรับการเก็บและบำรุงรักษาข้อมูลเหล่านั้น

ระบบฐานข้อมูลประกอบด้วยส่วนประกอบ 2 ส่วนคือ

1.ระบบจัดการฐานข้อมูล (Database Management System (DBMS)) เป็นโปรแกรมที่ใช้เพื่อจัดระเบียบและบำรุงรักษารายการของข้อมูลเหล่านี้

2.แอปพลิเคชันฐานข้อมูล (Database Application) เป็นโปรแกรมที่ช่วยให้เราดู และแก้ไขข้อมูลที่ถูกเก็บอยู่ใน DBMS

โดยปกติทั้ง DBMS และ Database Application จะทำงานอยู่บนเครื่องเดียวกัน ส่วนมากทั้งสองส่วนจะถูกรวมอยู่ในโปรแกรมเดียวกัน แต่ขณะนี้ความสนใจส่วนมากได้มุ่งไปที่เทคโนโลยีในการปฏิบัติ DBMS คือเทคโนโลยี Client/Server นั่นเอง

1.2. ข้อดีในการใช้โปรแกรมระบบจัดการฐานข้อมูล

- (1) ลดความซ้ำซ้อนของฐานข้อมูล
- (2) ลดความผิดพลาดของข้อมูล ในกรณีที่เรามีการจัดเก็บข้อมูลไว้หลายแห่ง การเปลี่ยนแปลงแก้ไขข้อมูลจะเกิดปัญหาว่าข้อมูลไม่เหมือนกัน ทำให้การทำงานผิดพลาดและเกิดความเสียหายได้ ซึ่งระบบฐานข้อมูลสามารถลดปัญหาเหล่านี้ได้อย่างดี
- (3) สามารถใช้ข้อมูลร่วมกันได้ ทำให้ประหยัดค่าใช้จ่ายและลดความผิดพลาด
- (4) ควบคุมความเป็นมาตรฐานของข้อมูลได้ การจัดเก็บข้อมูลที่มีศูนย์กลางที่เดียวทำให้สามารถควบคุมรูปแบบของข้อมูลได้ ทำให้ข้อมูลเป็นอันหนึ่งอันเดียวกัน ง่ายต่อการดูแลรักษา
- (5) ความปลอดภัยในการเก็บรักษาข้อมูล เนื่องจากการเก็บข้อมูลไว้ที่ศูนย์กลางข้อมูลทีเดียวจึงทำให้สามารถควบคุมและจัดสรรระดับของผู้ใช้ที่เข้ามาขอใช้ข้อมูลได้ เช่นการกำหนดรหัสผ่านให้การเข้ามาใช้ข้อมูล แบ่งระดับความปลอดภัยของผู้เข้ามาใช้ข้อมูล
- (6) สามารถควบคุมความถูกต้องของข้อมูลได้โดยง่าย

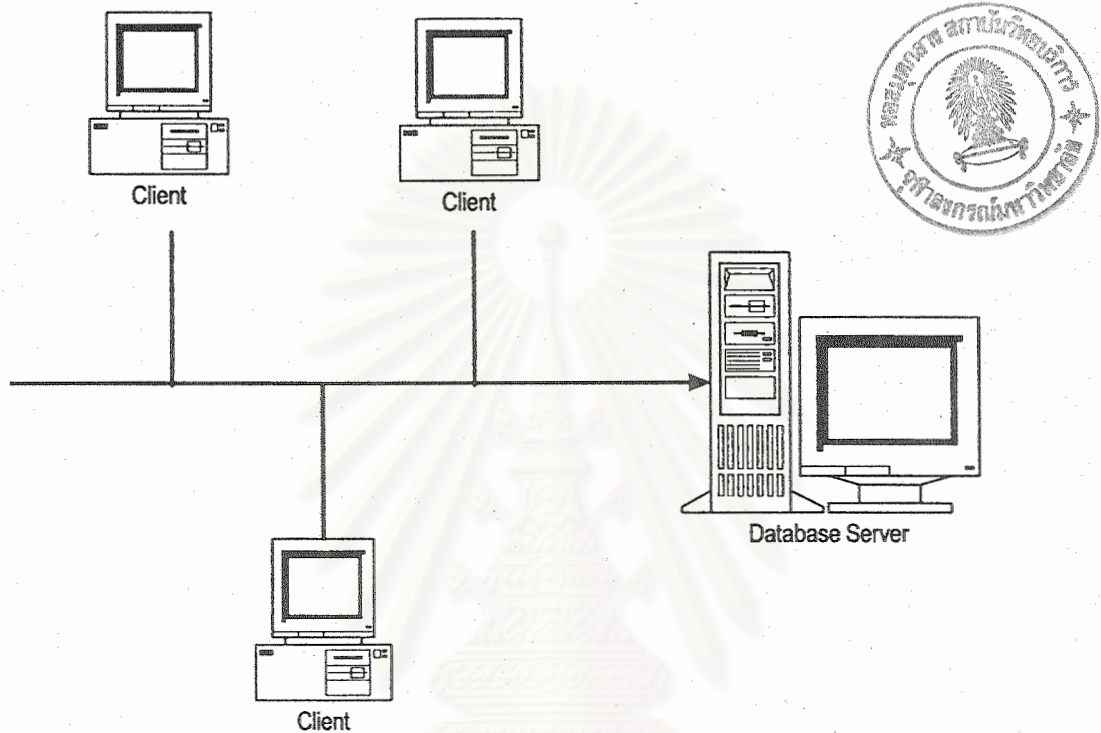
1.3. Relational Database

เป็น DBMS ประเภทหนึ่งที่นิยมใช้ในปัจจุบัน ในระบบ RDM ข้อมูลจะถูกจัดระเบียบเป็นเซตในทางคณิตศาสตร์ในโครงสร้างของตาราง พิลด์ข้อมูลแต่ละตัวจะเป็นคอลัมน์ในตาราง และแต่ละเรคอร์ดจะกลายเป็นแถวในตาราง

Relation Database มีข้อดีที่สำคัญคือ ความยืดหยุ่นที่สมบูรณ์ในการกล่าวถึงความสัมพันธ์ระหว่างรายการข้อมูลที่แตกต่างกัน การตัดสินใจหลักๆของผู้ออกแบบฐานข้อมูลคือการกำหนดตาราง Relation Database ถือว่าเป็นฐานข้อมูลแบบที่นำไปสู่การพัฒนาในระบบ Client/Server ซึ่งจะกล่าวถึงในหัวข้อต่อไป

2. บทที่ 2 ระบบฐานข้อมูลแบบ Client/Server

ระบบ Client/Server นี้เป็นการแยกการทำงานของ Front-end กับ Back-end ออกจากกันโดยผู้ใช้สามารถที่จะทำการทำงานกับฐานข้อมูลได้โดยไม่ต้องทำงานอยู่ที่เครื่องที่ทำหน้าที่เก็บฐานข้อมูลอยู่นั้นเราจะเรียกว่า Back-end ในการนำเสนอระบบนี้อย่างไรนั้นขึ้นกับ Platform ที่ Font-end กับ Back-end ทำงานอยู่ และระดับการจัดการที่ถูกแบ่งออกเป็น 2 ส่วน



รูปที่ 2.1 แสดงโครงสร้างของระบบ Client/Server Database

2.1. ข้อดี ของระบบฐานข้อมูลแบบ Client/Server

- (1) การแบ่งแยกการจัดการระหว่างระบบของ Client และ Database Server ออกจากกันโดยการจัดการฐานข้อมูลจะถูกทำที่ส่วนของ Back-end ส่วน DBMS จะถูกจัดการอยู่ที่ Server ทำให้สามารถทำการขยายการใช้งานของเครื่องคอมพิวเตอร์ที่มีขนาดเล็กหรือช้าเกินกว่าที่จะสามารถทำการ Run DBMS ที่ซับซ้อนบนเครื่องนั้น
- (2) ช่วยลดโหลด(Load)ให้กับระบบ Network ที่เชื่อมต่อได้ด้วยแทนที่จะต้องทำการส่งข้อมูลทั้งหมดไปและกลับผ่านทางสายแลน(LAN) ไปยังเครื่องที่ทำกรติดต่อเข้ามาทำให้การจราจรบนสายส่งลดน้อยลงเหลือเพียงแต่การส่ง Query มาจาก Client มายังตัวที่เป็นฐานข้อมูล เพื่อทำงานอย่างใดอย่างหนึ่ง เมื่อ Server ได้รับ Query ที่ส่งมากก็จะทำการคำนวณและส่งผลลัพธ์กลับไปยัง Client เพียงเท่านั้น

- (3) การทำงานของโปรแกรมไม่ขึ้นกับเครื่องที่ทำงาน ผู้ใช้จะไม่ถูกจำกัดบนเครื่องระบบใดระบบหนึ่งเท่านั้น สามารถนำเอาโปรแกรมไปทำงานบนเครื่องคอมพิวเตอร์เครื่องไหนก็ได้และสามารถทำงานได้กับระบบปฏิบัติการได้หลายตัวไม่ว่าจะเป็น MS-Windows, IBM OS/2, MS/PC-Dos เป็นต้นนอกจากนี้ Client และ Server ก็ไม่จำเป็นต้องใช้ฐานข้อมูลเดียวกัน
- (4) การรักษาความปลอดภัยของข้อมูล ในปัจจุบันระบบ Database Server ส่วนมากทำงานบน DBMS ที่ใช้การจัดการแบบ Relation DBMS เมื่อผู้ใช้ต้องการที่จะทำการแก้ไขค่าของข้อมูล จะต้องทำการ Log In เข้ามาใช้งานที่ Server ก่อนทำให้ข้อมูลไม่กระจัดกระจาย และมีความถูกต้องอยู่เสมอ
- (5) การปกป้องข้อมูล ที่เครื่องที่ทำหน้าที่ในการเก็บข้อมูล (Server) บางครั้งอาจมีการเข้ารหัสที่เก็บไฟล์โดยข้อมูลถูกเข้ารหัสเพื่อป้องกันการดูจากภายนอก DBMS ได้

2.2. ข้อเสียของระบบ Client/Server

- (1) การเพิ่มค่าใช้จ่ายในการจัดการและเตรียมบุคลากรที่จะทำหน้าที่บำรุงรักษา Database Server เมื่อจำนวนผู้ใช้มีมากขึ้นหรือเมื่อฐานข้อมูลมีขนาดใหญ่มากขึ้น นอกจากนี้การฝึกฝนผู้ควบคุมระบบยังเป็นค่าใช้จ่ายที่เพิ่มขึ้นในตอนเริ่มต้นด้วย เนื่องจากคนที่เข้ามาทำงานอาจจะไม่คุ้นเคยกับระบบที่ทำอยู่
- (2) การเพิ่มค่าใช้จ่ายทางด้าน Hardware ก็เพิ่มขึ้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

3. บทที่ 3 เกี่ยวกับ MySQL

MySQL เป็น SQL (Structured Queries Language) database server ซึ่ง SQL เป็น database language ที่นิยมมากภาษาหนึ่ง เป็น database server ขนาดเล็ก ซึ่งเหมาะกับ applications ที่มีขนาดเล็ก และ ปานกลาง อีกทั้งยังสนับสนุน standard SQL (ANSI) MySQL ถูกเขียนขึ้นมาใช้ได้หลาย platform ของคอมพิวเตอร์ ทั้ง Unix และ Windows

ในโลก MySQL เป็นการสร้าง client/server ที่ประกอบด้วย server daemon 'mysqld' และ client programs/libraries ที่แตกต่างกัน ความสามารถที่สำคัญของ MySQL คือ ความเร็ว และ ความทนทาน (robustness) MySQL ถูกสร้างเป็นกลุ่มของ routine ที่ใช้สำหรับตอบสนองการใช้งานซึ่งในปัจจุบัน MySQL ยังคงทำการพัฒนาอย่างต่อเนื่อง คุณสามารถดูรายละเอียดเพิ่มเติมของ MySQL ได้จาก web site หลักของ MySQL <http://www.mysql.com>

การติดตั้ง MySQL ในปัจจุบัน นั้นทำได้ง่ายมาก โดยเฉพาะ หากคุณใช้ Mandrake 7 เมื่อคุณ install mandrake 7 MySQL จะถูกติดตั้งมาให้เรียบร้อยแล้ว

หากต้องการติดตั้ง MySQL binary version ให้พิมพ์คำสั่ง ดังนี้

```
shell> gunzip < mysql-VERSION-OS.tar.gz | tar xvf -
shell> ln -s mysql-VERSION-OS mysql
shell> cd mysql
shell> scripts/mysql_install_db
shell> bin/safe_mysqld &
```

หากต้องการติดตั้ง MySQL แบบ RPM ต้องใช้ file ดังนี้

- MySQL-VERSION.i386.rpm สำหรับโปรแกรม MySQL server.
- MySQL-client-VERSION.i386.rpm สำหรับโปรแกรม MySQL client

วิธีการติดตั้งโปรแกรม ให้พิมพ์ดังนี้

```
shell> rpm -i MySQL-VERSION.i386.rpm MySQL-client-VERSION.i386.rpm
```

หากต้องการติดตั้งเพียงแค่โปรแกรม client ให้พิมพ์

```
shell> rpm -i MySQL-client-VERSION.i386.rpm
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4. บทที่ 4 ระบบการจัดการกับสิทธิการเข้าถึงข้อมูล

4.1 ความรู้พื้นฐานเกี่ยวกับ user-name และ รหัสผ่าน ใน MySQL

- User-name ใน MySQL ไม่ใช่ user-name ที่ใช้ในระบบ UNIX แต่โดยปกติเมื่อมีการ connect ไปยัง MySQL server แล้ว user-name บน UNIX จะถูกใช้เป็นค่า user-name เริ่มต้น โดยผู้ใช้สามารถระบุ user-name ของ MySQL โดยใช้ option `-u username` ได้

User-name ใน MySQL มีความยาวได้สูงสุด 16 ตัวอักษร

- รหัสผ่านของ user ใน MySQL ไม่ใช่รหัสผ่านของ user ที่ใช้ในระบบ UNIX

4.2 ความรู้พื้นฐานเกี่ยวกับการ connect MySQL server

MySQL client สามารถ connect ไปยัง MySQL server โดยใช้คำสั่งดังนี้

```
Shell> mysql [-h hostname] [-u username] [-pyourpassword]
Shell> mysql [--host=hostname] [--user=username] [--password=yourpassword]
```

โดยค่ามาตรฐานสำหรับ client มีดังนี้

หากไม่มีการระบุ option `-h hostname` มาตรฐานจะเป็น localhost

หากไม่มีการระบุ option `-u username` มาตรฐานจะเป็น UNIX user-name

การใช้ option `-p` โดยไม่ระบุ password MySQL client จะให้ผู้ใช้ป้อน password ดังนี้

```
Shell> mysql -u username -p
Enter password: *****
```

ในระบบ UNIX ผู้ใช้สามารถระบุค่าเริ่มต้นในการ connect MySQL server ได้โดยระบุภายในไฟล์ `.my.cnf` ใน home directory เช่น

```
[client]
host=hostname
user=username
password=yourpassword
```

เมื่อมีการระบุ password ในไฟล์ดังกล่าว ควรกำหนดสิทธิในการเข้าถึงไฟล์ให้เป็น 400 หรือ 600

4.3 ความรู้พื้นฐานเกี่ยวกับ Privileges system

หน้าที่หลักของ privileges system คือการตรวจสอบสิทธิของ user ในการ connect MySQL server, การใช้ query พื้นฐานเช่น select, insert, update และ delete นอกจากนี้ privileges system ยังใช้ในการจัดการสิทธิของ anonymous user, การเพิ่มสิทธิในการใช้ function พิเศษ เช่น LOAD DATA INFILE และการบริหารระบบ MySQL

4.4 ความรู้พื้นฐานเกี่ยวกับ Privileges table

Privileges table ประกอบไปด้วย 3 table หลักๆ คือ

(1) host table ใช้ในการกำหนดสิทธิ์ว่า host ใดมีสิทธิ์ทำอะไรกับ database ใดบ้าง โดยมี field ดังนี้

- Host – สามารถ connect จาก client ใดได้บ้าง
- Db – database ใดที่สามารถใช้งานได้
- Select_priv – สามารถใช้ query SELECT ได้หรือไม่ (Y/N)
- Insert_priv – สามารถใช้ query INSERT ได้หรือไม่ (Y/N)
- Update_priv – สามารถใช้ query UPDATE ได้หรือไม่ (Y/N)
- Delete_priv – สามารถใช้ query DELETE ได้หรือไม่ (Y/N)
- Create_priv – สามารถใช้ query CREATE ได้หรือไม่ (Y/N)
- Drop_priv – สามารถใช้ query DROP ได้หรือไม่ (Y/N)

เช่น client host ชื่อ cpe, database ที่ต้องการใช้งานชื่อ student จะใช้คำสั่งดังนี้เพื่อให้ client จาก cpe สามารถใช้งาน student ได้

```
mysql> insert into
-> host(host,db,Select_priv,Insert_priv,Update_priv,
-> Delete_priv,Create_priv,Drop_priv)
-> values('localhost','student','Y', 'Y', 'Y', 'Y', 'Y', 'Y');

mysql> insert into
-> host(host,db,Select_priv,Insert_priv,Update_priv,
-> Delete_priv,Create_priv,Drop_priv)
values('cpe','student','Y', 'Y', 'Y', 'Y', 'Y', 'Y');
```

โดยรูปแบบ field host ที่ใช้ได้คือ localhost, hostname, เลข IP, ตัวอักษรที่ใช้ wildcard เช่น cp% หมายถึงชื่อ host ที่ขึ้นต้นด้วย cp เป็นต้น

(2) user table ใช้ในการกำหนดสิทธิ์ว่า user ใด สามารถใช้งาน client ได้จาก host ใด และสามารถทำอะไรได้บ้าง โดยมี field ดังนี้

- Host – สามารถ connect จาก client ใดได้บ้าง
- User – username ใดสามารถ connect ได้
- Select_priv – สามารถใช้ query SELECT ได้หรือไม่ (Y/N)
- Insert_priv – สามารถใช้ query INSERT ได้หรือไม่ (Y/N)
- Update_priv – สามารถใช้ query UPDATE ได้หรือไม่ (Y/N)
- Delete_priv – สามารถใช้ query DELETE ได้หรือไม่ (Y/N)
- Create_priv – สามารถใช้ query CREATE ได้หรือไม่ (Y/N)
- Drop_priv – สามารถใช้ query DROP ได้หรือไม่ (Y/N)
- Reload_priv – สามารถใช้ query RELOAD ได้หรือไม่ (Y/N)

Shutdown_priv – สามารถสั่ง shutdown MySQL server ได้หรือไม่ (Y/N)
Process_priv – สามารถสั่ง list คำสั่งที่กำลังประมวลผลบน server ได้หรือไม่ (Y/N)
File_priv – user สามารถเขียนไฟล์ลงบน server ได้หรือไม่ (Y/N)

(3) db table ใช้ในการกำหนดสิทธิ์ว่า database ใด สามารถใช้งานโดย user ใด จาก client ใด และสามารถทำอะไรได้บ้าง โดยมี field ดังนี้

- Host – สามารถ connect จาก client ใดได้บ้าง
- Db – database ใดที่สามารถใช้งานได้
- User – username ใดสามารถ connect ได้
- Select_priv – สามารถใช้ query SELECT ได้หรือไม่ (Y/N)
- Insert_priv – สามารถใช้ query INSERT ได้หรือไม่ (Y/N)
- Update_priv – สามารถใช้ query UPDATE ได้หรือไม่ (Y/N)
- Delete_priv – สามารถใช้ query DELETE ได้หรือไม่ (Y/N)
- Create_priv – สามารถใช้ query CREATE ได้หรือไม่ (Y/N)
- Drop_priv – สามารถใช้ query DROP ได้หรือไม่ (Y/N)

4.5 ระบบรักษาความปลอดภัยทั่วไป

ระบบรักษาความปลอดภัยใน MySQL จะสามารถกำหนด หรือปรับเปลี่ยน ได้จาก Access Control Lists (ACL-s), SSL แต่สิ่งที่สำคัญที่สุดก็คือการใช้งาน MySQL อย่างปลอดภัย เมื่อ MySQL

หลักการการใช้งาน MySQL ให้ปลอดภัย

- (1) ให้ความสนใจกับ ACL-s การ GRANT หรือ REVOKE ควรทำโดยระมัดระวัง ไม่ควร GRANT สิทธิใด ๆ ให้แก่ผู้ใช้โดยไม่จำเป็น ควรตั้งรหัสผ่านสำหรับการเข้าสู่ระบบโดย account root เสมอ การตรวจสอบสิทธิ์ต่าง ๆ ของผู้ใช้สามารถทำได้โดยใช้คำสั่ง SHOW GRANTS
- (2) การเก็บรหัสผ่านควรมีการเข้ารหัสเสมอ ทำได้โดยใช้ MD5() หรือ hashing function
- (3) รหัสผ่านไม่ควรมีความสำคัญที่มีอยู่ในพจนานุกรมภาษาอังกฤษ เช่น xfish98
- (4) MySQL server ควรทำงานอยู่หลังระบบ firewall โดยป้องกัน port 3306 ของ MySQL ให้มีการเข้าถึงจาก client เท่าที่จำเป็น
- (5) การพัฒนาโปรแกรมประยุกต์ซึ่งติดต่อกับ MySQL ควรมีการตรวจสอบความถูกต้องเสมอ การตรวจสอบที่จำเป็นสำหรับแต่ละประเภทของโปรแกรมประยุกต์มีดังนี้
 - (5.1) โปรแกรมประยุกต์ซึ่งทำงานผ่าน www
 - ทดสอบการป้อนค่า %22 ("), %27(')
 - ทดสอบการป้อนค่าที่เป็นตัวอักษร, space, สัญลักษณ์ต่าง ๆ ลงใน field ที่เป็นตัวเลข
 - ทดสอบขนาดของ data ก่อนส่งไปประมวลผลยัง MySQL server
 - (5.2) โปรแกรมประยุกต์ซึ่งมีการใช้งาน PHP3
 - ตรวจสอบการใช้ function addslashes()
 - (5.3) โปรแกรมประยุกต์ซึ่งมีการใช้งาน MySQL C API
 - ตรวจสอบการเรียกใช้ API mysql_escape()
 - (5.4) โปรแกรมประยุกต์ซึ่งมีการใช้งาน MySQL++

- ตรวจสอบการใช้ escape และ quote ในการ query
- (6) การส่งข้อมูลผ่านระบบเน็ตเวิร์กควรมีการเข้ารหัสโดยใช้ SSL

4.6 การป้องกัน MySQL ให้ปลอดภัยจาก crackers

- (1) ตั้งรหัสผ่านสำหรับผู้ใช้งานทุกคนเสมอ โดยการตั้งหรือเปลี่ยนรหัสผ่านสามารถทำได้ดังนี้

```
Shell> mysql -u username mysql
Mysql> UPDATE user SET Password=PASSWORD('new_password') WHERE
user='username';
Mysql> FLUSH PRIVILEGES;
```

- (2) ไม่ควรสั่งให้ MySQL daemon ทำงาน โดยเป็น process ของ root
- (3) กำหนดสิทธิ์ Script ไฟล์ mysql.server ให้สามารถอ่านได้เฉพาะ root เท่านั้น เมื่อมีการกำหนดรหัสผ่านของ UNIX root ใน script ดังกล่าว
- (4) กำหนดสิทธิ์ในการอ่านเขียน directory ที่ใช้เก็บ database ให้แก่ UNIX user ที่เป็นเจ้าของ process MySQL daemon เท่านั้น
- (5) ไม่ควรกำหนดสิทธิ์ process ให้แก่ user ทุกคน
- (6) ไม่ควรกำหนดสิทธิ์ file ให้แก่ user ทุกคน

4.7 เมื่อสิทธิของผู้ใช้ถูกเปลี่ยนแปลง จะมีผลต่อผู้ใช้เมื่อใด ?

เมื่อ mysqld start , ข้อมูลใน grant table จะถูกอ่านเก็บไว้ในหน่วยความจำ เพื่อนำไปใช้งาน การแก้ไขข้อมูลใน grant tables โดยใช้คำสั่ง GRANT, REVOKE หรือ SET PASSWORD จะทำให้สิทธิที่ถูกแก้ไขมีผลใช้ได้ทันที

ถ้าคุณแก้ไข grant tables โดยการใช้คำสั่ง INSERT หรือ UPDATE คุณจำเป็นต้องใช้คำสั่ง FLUSH PRIVILEGES หรือ run mysqladmin flush-privileges เพื่อบอกให้ server reload grant tables ไมเช่นนั้นสิทธิต่างๆ ที่คุณแก้ไขจะไม่มีผลใดๆ จนกว่าคุณจะ restart server

เมื่อ server รับรู้ว่ามี grant tables ถูกเปลี่ยนแปลง, client ที่ติดต่อกับ server ในขณะนั้นๆ จะได้รับผลกระทบดังนี้

- สิทธิการใช้งาน Table และ column จะมีผลต่อ client เมื่อ client request ครั้งต่อไป
 - สิทธิการใช้งาน Database จะมีผลต่อ client เมื่อ client เรียกใช้คำสั่ง USE db_name ในครั้งต่อไป
- ส่วนการเปลี่ยนแปลงสิทธิอื่นๆ นอกเหนือจากนี้จะมีผลต่อ client เมื่อ client connect มายัง server ครั้งต่อไป

4.8 การ set up สิทธิการใช้ MySQL เริ่มต้น

หลังจากติดตั้ง MySQL, คุณสามารถ set up access privileges เริ่มต้นของ MySQL ได้โดยการ run script/mysql_install_db ซึ่ง script/mysql_install_db จะทำการ start up mysqld server จากนั้นก็จะกำหนดค่าเริ่มต้นของ grant table ดังนี้ :

- The MySQL root user จะเป็น superuser ซึ่งสามารถทำทุกอย่างได้ และการติดต่อกับ server จะต้องทำจาก localhost เท่านั้น Note: password เริ่มต้นของ root จะถูกกำหนดเป็นว empty นั่นคือทุกๆ คนสามารถ connect มายัง server โดยใช้ user root แบบไม่ต้องใช้ password ได้
- An anonymous user ถูกสร้างให้สามารถทำอะไรก็ได้กับ databases ที่ชื่อ 'test' หรือ ขึ้นต้นด้วย 'test_' การติดต่อกับ server ต้องทำจาก local host เท่านั้น
- สิทธิอื่น เช่น user ธรรมดาไม่สามารถ ใช้ mysqladmin shutdown หรือ mysqladmin processlist ได้

เมื่อคุณ install MySQL เสร็จ สิ่งแรกที่คุณควรทำคือการทำ password สำหรับ root user โดยใช้วิธีการดังนี้

```
shell> mysql -u root mysql
mysql> UPDATE user SET Password=PASSWORD('new_password')
      WHERE user='root';
mysql> FLUSH PRIVILEGES;
```

ใน MySQL 3.22, คุณสามารถใช้คำสั่ง SET PASSWORD ได้ดังนี้

```
shell> mysql -u root mysql
mysql> SET PASSWORD FOR root=PASSWORD('new_password');
```

อีกวิธีในการ set password ของ root คือ

```
shell> mysqladmin -u root password new_password
```

4.9 การเพิ่มผู้ใช้ และ การจัดการกับสิทธิของผู้ใช้

คุณสามารถเพิ่ม user ได้โดยใช้ 2 วิธี คือ

- (1) ใช้ GRANT statement
- (2) เปลี่ยนแปลง MySQL grant tables โดยตรง

ซึ่งวิธีที่สมควรใช้มากกว่า คือ วิธีที่ (1) เพราะเป็นวิธีที่รัดกุม และ เกิดข้อผิดพลาดได้น้อยกว่า

ตัวอย่างการ add new user โดยใช้วิธีที่ (1) ซึ่งผู้ที่สามารถ add new user ได้ จะต้อง connect เป็น root ก่อน, root user จะต้อง insert สิทธิ สำหรับ mysql database และ reload administrative privilege คุณสามารถเพิ่มผู้ใช้ได้โดยการใช้คำสั่ง GRANT ดังนี้ :

```
shell> mysql --user=root mysql
mysql> GRANT ALL PRIVILEGES ON *.* TO ooh@localhost
      IDENTIFIED BY 'something' WITH GRANT OPTION;
mysql> GRANT ALL PRIVILEGES ON *.* TO ooh@"%"
      IDENTIFIED BY 'something' WITH GRANT OPTION;
mysql> GRANT RELOAD,PROCESS ON *.* TO admin@localhost;
mysql> GRANT USAGE ON *.* TO dummy@localhost;
```


คำสั่ง GRANT เหล่านี้เป็นการสร้าง new users ขึ้นมาใหม่ 3 users

ooh

เป็น full superuser ซึ่งสามารถ connect มายัง server จากที่ไหนก็ได้ แต่ต้องใช้ password 'something' ในการติดต่อกับ server

admin

เป็น user ที่สามารถ connect จาก local host โดยไม่ใช้ password และสามารถใช้สิทธิในการ reload และ process MySQL ได้ นั่นคือ admin สามารถ run คำสั่ง mysqladmin reload, mysqladmin refresh และ mysql flush* ได้ แต่ไม่สามารถใช้ Database ไหนได้ ซึ่งมันสามารถถูก grant ได้ภายหลังโดยใช้ GRANT statement

dummy

user สามารถ connect โดยไม่ใช้ password แต่ต้องจาก local host เท่านั้น สิทธิทั่วไปถูก set ให้เป็น 'N'

คุณสามารถ add ข้อมูลของ user เหล่านี้ได้โดยตรง โดยใช้ INSERT statements และ บอกให้ server reload grant tables

```
shell> mysql --user=root mysql
mysql> INSERT INTO user VALUES('localhost','ooh',PASSWORD('something'),
'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y')
mysql> INSERT INTO user VALUES('?', 'ooh', PASSWORD('something'),
'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y')
mysql> INSERT INTO user SET Host='localhost',User='admin',
Reload_priv='Y', Process_priv='Y';
mysql> INSERT INTO user (Host,User,Password)
VALUES('localhost','dummy','');
mysql> FLUSH PRIVILEGES;
```

ตัวอย่างต่อไปนี้เป็นกรเพิ่ม user ชื่อ "custom" ซึ่งสามารถ connect เข้า server จาก host localhost, server.domain และ cpe.cmu. user นี้ ต้องการเข้าถึง (access) เพียง database stuinfo จาก localhost, database lecturerinfo จาก cpe.cmu และ database transcript จากทั้ง 3 host โดยใช้ password 'manager'

การ setup สิทธิต่างๆ ของ user โดยการให้ GRANT statements , ทำได้ดังนี้ :

```
shell> mysql --user=root mysql
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP
ON stuinfo.*
TO custom@localhost
IDENTIFIED BY 'manager';
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP
ON lecturerinfo.*
TO custom@cpe.cmu
IDENTIFIED BY 'manager';
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP
ON transcript.*
TO custom@'?
IDENTIFIED BY 'manager';
```

หรือจะ set up สิทธิของ user จากการแก้ไขที่ grant table โดยตรงก็ได้ ดังนี้ :

```
shell> mysql --user=root mysql

mysql> INSERT INTO user (Host,User,Password)
VALUES('localhost','custom',PASSWORD('manager'));
mysql> INSERT INTO user (Host,User,Password)
VALUES('server.domain','custom',PASSWORD('manager'));
mysql> INSERT INTO user (Host,User,Password)
VALUES('cpe.cmu','custom',PASSWORD('manager'));
mysql> INSERT INTO db
(Host,Db,User,Select_priv,Insert_priv,Update_priv,Delete_priv,
Create_priv,Drop_priv)
VALUES
('localhost','stuinfo','custom','Y','Y','Y','Y','Y','Y');
mysql> INSERT INTO db
(Host,Db,User,Select_priv,Insert_priv,Update_priv,Delete_priv,
Create_priv,Drop_priv)
VALUES
('whitehouse.gov','lecturerinfo','custom','Y','Y','Y','Y','Y','Y');
mysql> INSERT INTO db
(Host,Db,User,Select_priv,Insert_priv,Update_priv,Delete_priv,
Create_priv,Drop_priv)
VALUES('%','transcript','custom','Y','Y','Y','Y','Y','Y');
mysql> FLUSH PRIVILEGES;
```

คำสั่ง INSERT 3 คำสั่งแรก เป็นการ add ข้อมูลลงใน user table เพื่ออนุญาตให้ user custom สามารถ connect มายัง server โดยมาจาก host ที่แตกต่างกันได้ ด้วย password 'stupid' แต่ยังไม่ให้สิทธิต่างๆ แก่ custom

คำสั่ง INSERT อีก 2 คำสั่งต่อมา เป็นการ add ข้อมูลลงใน db table เพื่อให้สิทธิต่างๆ แก่ custom สำหรับ database ชื่อ bankaccount, expenses and customer โดยการ access เข้ามาใช้ database กระทำได้จาก host ที่เจาะจงไว้เท่านั้น ซึ่งวิธีนี้จะต้องทำการ reload grant table ให้กับ server ใหม่ โดยการใช้คำสั่ง FLUSH PRIVILEGES ในบรรทัดสุดท้าย เพื่อให้สิทธิต่างๆ สามารถใช้ได้เลย

ถ้าคุณต้องการเจาะจง user ในการ access จากเครื่องใดๆ ก็ได้ใน domain เดียวกัน คุณสามารถใช้คำสั่ง GRANT ดังนี้ :

```
mysql> GRANT ...
      ON *.*
      TO myusername@%.mydomainname.com"
      IDENTIFIED BY 'mypassword';
mysql> INSERT INTO user VALUES ('%.mydomainname.com', 'myusername',
      PASSWORD('mypassword'),...);
mysql> FLUSH PRIVILEGES;
```

4.10 วิธีการกำหนด passwords

จากตัวอย่างในหัวข้อที่ผ่านมาจะเห็นได้ว่าเมื่อคุณ INSERT หรือ UPDATE password คุณต้องใช้ function PASSWORD() เพื่อเข้ารหัส password เนื่องจากการทำงานเช่นนี้ ทำให้ password ถูกเก็บลง table ในแบบที่ถูกเข้ารหัส เพื่อความปลอดภัยของข้อมูล

หากคุณลืมใช้ function PASSWORD() , password ก็จะถูกเก็บในรูปแบบของตัวหนังสือ (plaintext)

```
shell> mysql -u root mysql
mysql> INSERT INTO user (Host,User,Password)
VALUES ('%', 'jeffrey', 'biscuit');
mysql> FLUSH PRIVILEGES;
```

จากคำสั่งนี้ ผลลัพธ์ที่ได้คือ คำว่า 'biscuit' จะถูกเก็บเป็น password ใน user table เมื่อ user jeffrey พยายาม connect มายัง server โดยใช้ password นี้, mysql client จะ encrypt password ที่ส่งเข้ามาโดยใช้ function PASSWORD() และส่งผลลัพธ์กลับไปให้ server และ server ก็จะทำการเปรียบเทียบค่าใน user table ซึ่งเป็นค่า 'biscuit' กับ password ที่ถูกส่งมาแบบเข้ารหัสซึ่งไม่ใช่ค่า 'biscuit' ผลลัพธ์ของการเปรียบเทียบก็ผิด ดังนั้น server จึงไม่ยอมให้ user login เข้ามาได้

```
shell> mysql -u jeffrey -pbiscuit test
Access denied
```

วิธีการใช้ INSERT statement ที่ถูกต้อง ควรเขียนดังนี้ :

```
mysql> INSERT INTO user (Host,User,Password)
VALUES ('%', 'jeffrey', PASSWORD('biscuit'));
```

หรือคุณจะใช้ PASSWORD() สำหรับ SET PASSWORD statements ก็ได้ ดังนี้ :

```
mysql> SET PASSWORD FOR jeffrey@"%" = PASSWORD('biscuit');
```

ถ้าคุณ set password โดยใช้ GRANT ... IDENTIFIED BY statement หรือ ใช้ คำสั่ง mysqladmin password ก็ไม่จำเป็นต้องใช้ function PASSWORD() เนื่องจากมันจะทำการเข้ารหัสให้โดยอัตโนมัติ ดังนี้ :

```
mysql> GRANT USAGE ON *.* TO jeffrey@"%" IDENTIFIED BY 'biscuit';
```

หรือ ใช้ mysqladmin

```
shell> mysqladmin -u jeffrey password biscuit
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

5. บทที่ 5 Language Reference

5.1. วิธีการเขียนสตริง และ ตัวเลข

5.1.1. สตริง

สตริง คือ ตัวอักษรที่นำมาเขียนเรียงกัน ซึ่งจะถูกล้อมด้วยเครื่องหมาย single quote (") หรือ double quote (") ก็ได้ เช่น

```
'a string'  
"another string"
```

สำหรับสัญลักษณ์พิเศษบางอย่างที่ใช้ใน string ที่เรียกว่า escape character นั้น ต้องใช้เครื่องหมาย backslash (\) ซึ่ง escape character ใน MySQL นั้น มีดังต่อไปนี้

Escape character	Meaning
\0	NULL character
\n	newline character
\t	Tab character
\r	Carriage return character
\b	Backspace character
\'	Single quote character
\"	Double quote character
\\	Backslash character
\%	Wildcard character
_	Wildcard character

วิธีการที่จะเขียน string ที่มี quotes รวมอยู่ด้วย มีหลายวิธี ดังนี้

- เครื่องหมาย ' ใน string ที่อยู่ในเครื่องหมาย " สามารถเขียนด้วย "
- เครื่องหมาย " ใน string ที่อยู่ในเครื่องหมาย "" สามารถเขียนด้วย ""
- คุณสามารถใช้เครื่องหมาย \ อยู่ข้างหน้า quote character ก็ได้
- เครื่องหมาย ' ใน string ที่อยู่ในเครื่องหมาย "" สามารถเขียนได้เลย
- เครื่องหมาย " ใน string ที่อยู่ในเครื่องหมาย " ก็สามารถเขียนได้เลย

```
mysql> SELECT 'hello', "hello", ""hello"", 'hel'lo', '\hello';
+-----+-----+-----+-----+-----+
| hello | "hello" | ""hello"" | hel'lo | '\hello |
+-----+-----+-----+-----+-----+

mysql> SELECT "hello", "'hello'", "'hello'", "hel""lo", "\"hello\"";
+-----+-----+-----+-----+-----+
| hello | 'hello' | 'hello' | hel"lo | "hello |
+-----+-----+-----+-----+-----+

mysql> SELECT "This\nIs\nFour\nlines";
+-----+
| This
Is
Four
lines |
+-----+
```

5.1.2. ตัวเลข

integer number คือ กลุ่มของตัวเลขที่เรียงต่อกัน ส่วน float number นั้นจะมีการใช้เครื่องหมาย . เพื่อทำการแยกส่วนทศนิยมออก โดยทั้ง integer และ float numbers นั้นสามารถมีค่าเป็นลบได้ กำหนดโดยใช้เครื่องหมาย - นำหน้า ดังตัวอย่างต่อไปนี้

```
1221
0
-32
294.42
-32032.6809e+10
148.00
```

จะเห็นว่าเลขจำนวนเต็มนั้นสามารถอ้างอยู่ในรูปของ float number ได้เช่นกัน

5.1.3. เลขฐานสิบหก

MySQL นั้นสามารถอ้างถึงตัวเลขฐานสิบหกได้ เมื่อ MySQL ทำการประมวลผลแล้วจะสามารถแปลงเลขฐานสิบหกเป็นเลขฐานสิบ หรือแปลงเป็นอักขระก็ได้ โดยเลขฐานสิบหกสองตัวติดกันจะถูกแปลงเป็นอักขระหนึ่งตัว ดังตัวอย่างต่อไปนี้

```
mysql> SELECT 0xa+0
-> 10
mysql> select 0x5061756c;
-> Paul
```

5.1.4. NULL value

NULL values หมายถึง ไม่มีค่า ซึ่งแตกต่างจากค่าศูนย์ในระบบตัวเลข หรือ empty string ในข้อมูลประเภท string โดย NULL สามารถอ้างได้โดยใช้เครื่องหมาย backslash นำหน้า ดังนี้ \N

5.1.5. Database, table, index, column, alias names

ในการตั้งชื่อ database, table, index, column และ alias นั้น มีกฎในการตั้งชื่อ ดังตารางต่อไปนี้

Identifier	max length	Allowed characters
Database	64	ใช้อักขระใดก็ได้ที่อนุญาตให้ใช้ในการตั้ง 'directory name ยกเว้นเครื่องหมาย /
Table	64	ใช้อักขระใดก็ได้ที่อนุญาตให้ใช้ในการตั้ง file name ยกเว้นเครื่องหมาย / หรือ .
Column	64	ใช้อักขระใดก็ได้
Alias	255	ใช้อักขระใดก็ได้

หมายเหตุ หากว่าชื่อของ database หรือ table หรือ column หรือ alias นั้นไปตรงกับชื่อคำสั่งเฉพาะของ MySQL แล้ว จะสามารถอ้างถึงได้โดยต้องใส่เครื่องหมาย '' ครอบชื่อนั้นๆไว้เสมอ ดังตัวอย่างต่อไปนี้

```
SELECT * from `select` where `select`.id > 100;
```

สำหรับการอ้างถึง columns นั้นสามารถทำได้ตามกฎเกณฑ์ดังในตารางต่อไปนี้

Column reference	Meaning
col_name	เป็นการอ้างถึง Column col_name ใน table ที่กำลังถูกใช้อยู่
tbl_name.col_name	อ้างถึง Column col_name จาก table tbl_name ของ database ที่กำลังเรียกใช้อยู่
db_name.tbl_name.col_name	อ้างถึง Column col_name จาก table tbl_name ของ database db_name. สามารถอ้างแบบนี้ได้เฉพาะ MySQL 3.22 หรือรุ่นที่ใหม่กว่า
`column_name`	เป็นการอ้างถึง column ที่มีชื่อเหมือนกับคำสั่งเฉพาะของ MySQL

เราไม่จำเป็นต้องอ้างถึง tbl_name หรือ db_name.tbl_name นอกเสียจากว่าเกิดกรณีที่ทำให้เกิดความสับสนขึ้นมาได้ ยกตัวอย่างเช่น table t1 และ t2 นั้นมี column ชื่อ c เหมือนกัน ดังนั้นในการอ้างถึงในการทำ query นั้นเราทำการเลือกใช้ทั้งสอง table แล้วทำให้เกิดความสับสน หากทำการอ้างถึง column c เนื่องจากไม่ได้มี column c เพียงอันเดียว ดังนั้นเวลาอ้างถึง column c นั้นจำเป็นต้องอ้างว่าเป็น t1.c หรือ t2.c ในทำนองเดียวกัน ถ้าหากมี table t ใน database db1 และ db2 เราต้องทำการอ้างถึง columns ในทั้งสอง table นั้นก็จำเป็นต้องทำการอ้างชื่อ database ด้วย ดังเช่น db1.t.col_name หรือ db2.t.col_name

5.1.6. Case sensitivity in names

ใน MySQL นั้น databases และ tables จะมีลักษณะเช่นเดียวกับ directory และ file ใน directory นั้น ดังนั้นในการตั้งชื่อ databases และ tables นั้น สำหรับระบบปฏิบัติการ unix นั้น อักขระตัวพิมพ์ใหญ่และตัวพิมพ์เล็กจะมีความแตกต่างกัน ถือเป็นคนละตัวกัน แต่ในระบบปฏิบัติการ Win32 นั้นอักขระตัวพิมพ์ใหญ่หรือเล็กนั้นเหมือนกัน

หมายเหตุ ถึงแม้ว่าชื่อของ database และ table นั้นจะไม่เกิดปัญหาในการอ้างชื่อด้วยตัวพิมพ์เล็กหรือตัวพิมพ์ใหญ่ก็ตาม เราไม่ควรที่จะอ้างถึง database และ table โดยใช้ทั้งตัวพิมพ์เล็กและตัวพิมพ์ใหญ่พร้อมๆกันในการทำ query ครั้งหนึ่งๆ จากตัวอย่างต่อไปนี้เราไม่สามารถอ้างถึง table ด้วยชื่อ my_table และ MY_TABLE ในการทำ query อันเดียวกันได้


```
mysql> SELECT * FROM my_table WHERE MY_TABLE.col=1;
```

ส่วน Column names นั้นจะสามารถอ้างถึงได้โดยต้องระบุถึงชื่อของ column ให้ถูกต้อง โดยอักษรตัวพิมพ์เล็กและตัวพิมพ์ใหญ่นั้นมีความแตกต่างกัน ไม่ว่าจะเป็นระบบ unix หรือ Win32

การทำ alias บน table ก็ต้องอ้างถึงด้วยตัวพิมพ์เล็กหรือตัวพิมพ์ใหญ่อย่างใดอย่างหนึ่ง เท่านั้นในการทำ query ครั้งหนึ่ง ดังตัวอย่างต่อไปนี้ จะเห็นได้ว่าไม่สามารถอ้างถึง table เดียวกัน ด้วยชื่อ A และ a ได้

```
mysql> SELECT col_name FROM tbl_name AS a  
WHERE a.col_name = 1 OR A.col_name = 2;
```

แต่การทำ alias บน column นั้นจะไม่เกิดปัญหาถึงแม้ว่าเราจะอ้างด้วยทั้งตัวพิมพ์เล็กหรือตัวพิมพ์ใหญ่

5.2. ชนิดของข้อมูลคอลัมน์

Column ที่สามารถมีได้ใน MySQL แต่ละชนิดต้องการเนื้อที่ในการเก็บข้อมูลต่างกัน ดังนี้

5.2.1. ข้อมูลชนิดตัวเลข

Column type	Storage required
TINYINT	1 byte
SMALLINT	2 bytes
MEDIUMINT	3 bytes
INT	4 bytes
INTEGER	4 bytes
BIGINT	8 bytes
FLOAT(X)	4 if $X \leq 24$ or 8 if $25 \leq X \leq 53$
FLOAT	4 bytes
DOUBLE	8 bytes
DOUBLE PRECISION	8 bytes
REAL	8 bytes
DECIMAL(M,D)	M bytes (D+2, if $M < D$)
NUMERIC(M,D)	M bytes (D+2, if $M < D$)

5.2.2. ข้อมูลชนิดวันที่ และ เวลา

Column type	Storage required
DATE	3 bytes
DATETIME	8 bytes
TIMESTAMP	4 bytes
TIME	3 bytes
YEAR	1 byte

หากไม่มีข้อมูลใน column ชนิด date และ time (Zero value) ค่าที่เก็บลงใน column แสดงดังตารางต่อไปนี้

Column type	"Zero" value
DATETIME	'0000-00-00 00:00:00'
DATE	'0000-00-00'
TIMESTAMP	0000000000000000 (length depends on display size)
TIME	'00:00:00'
YEAR	0000

5.2.3. ข้อมูลชนิดสตริง

Column type	Storage required
CHAR(M)	M bytes, $1 \leq M \leq 255$
VARCHAR(M)	L+1 bytes, where $L \leq M$ and $1 \leq M \leq 255$
TINYBLOB, TINYTEXT	L+1 bytes, where $L < 2^8$
BLOB, TEXT	L+2 bytes, where $L < 2^{16}$
MEDIUMBLOB, MEDIUMTEXT	L+3 bytes, where $L < 2^{24}$
LOBLOB, LONGTEXT	L+4 bytes, where $L < 2^{32}$
ENUM('value1','value2',...)	1 or 2 bytes, depending on the number of enumeration values (65535 values maximum)
SET('value1','value2',...)	1, 2, 3, 4 or 8 bytes, depending on the number of set members (64 members maximum)

5.3. Functions for use in SELECT and WHERE clauses

select_expression หรือ where_definition ในประโยค SQL สามารถใส่นิพจน์ใดที่ใช้ฟังก์ชันตั้งที่จะอธิบายต่อไปได้ นิพจน์ที่มี NULL จะได้ค่าที่เป็น NULL เสมอ เว้นแต่ว่าในนิพจน์นั้นจะมีฟังก์ชันและตัวกระทำตัวอื่นอยู่ในนั้น (หมายเหตุ ห้ามมีช่องว่างหรือวงเล็บในชื่อฟังก์ชัน)

ตัวอย่าง

```
mysql> select MOD(29,9);  
-> 2
```

5.3.1 ฟังก์ชันการรวมกลุ่ม

(...) วงเล็บ ใช้ในการจัดลำดับของการประมวลผลในนิพจน์ ตัวอย่างเช่น

```
mysql> select 1+2*3;  
-> 7  
mysql> select (1+2)*3;  
-> 9
```

5.3.2 ตัวกระทำทางคณิตศาสตร์ทั่วไป

ประกอบด้วยตัวกระทำทางคณิตศาสตร์ที่ใช้กันบ่อยๆ เช่น +, -, *, /

ตัวอย่าง

+ : การบวก

```
mysql> select 3+5;  
-> 8
```

- : การลบ

```
mysql> select 3-5;  
-> -2
```

* : การคูณ

```
mysql> select 3*5;  
-> 15  
mysql> select 18014398509481984*18014398509481984.0;  
-> 324518553658426726783156020576256.0  
mysql> select 18014398509481984*18014398509481984;  
-> 0
```

ผลลัพธ์ของนิพจน์สุดท้ายไม่ถูกต้องเพราะว่าผลลัพธ์ของการคูณเลขจำนวนเต็มเกินค่าในช่วงการคำนวณของ BIGINT 64 บิต

/ : การหาร

```
mysql> select 3/5;  
-> 0.60
```

การหารด้วยค่าศูนย์ ผลลัพธ์จะได้เป็นค่า NULL

```
mysql> select 102/(1-1);  
-> NULL
```

5.3.3 Bit functions

MySQL ใช้ BIGINT 64 บิตในการกระทำแบบบิต ดังนั้นตัวกระทำจึงมีขนาดใหญ่สุดได้ 64 บิต
ตัวกระทำทางบิต ได้แก่

| : Bitwise OR

```
mysql> select 29 | 15;  
-> 31
```

& : Bitwise AND

```
mysql> select 29 & 15;  
-> 13
```

<< : การเลื่อนตัวเลขจำนวนเต็มไปทางซ้าย

```
mysql> select 1 << 2  
-> 4
```

>> : การเลื่อนตัวเลขจำนวนเต็มไปทางซ้าย

```
mysql> select 4 >> 2  
-> 1
```

- : เปลี่ยนค่าบิตทั้งหมดให้เป็นค่าตรงกันข้าม

```
mysql> select 5 & -1  
-> 4
```

BIT_COUNT(N)

ส่งกลับจำนวนบิตที่ใช้ในการนับเลขจำนวน N

```
mysql> select BIT_COUNT(29);  
-> 4
```

5.3.4 Logical operations

ฟังก์ชันตรรกทั้งหมดจะส่งค่ากลับเป็น 1 (จริง) หรือ 0 (เท็จ) เท่านั้น

ฟังก์ชันทางตรรก ได้แก่

NOT : !

NOT จะส่งค่า 1 กลับถ้าอาร์กิวเมนต์เป็น 0 และส่ง 0 กลับถ้าอาร์กิวเมนต์เป็นหนึ่ง ถ้าเป็น NOT NULL จะส่งกลับค่า NULL เช่น

```
mysql> select NOT 1;  
-> 0  
mysql> select NOT NULL;  
-> NULL  
mysql> select ! (1+1);  
-> 0  
mysql> select ! 1+1;  
-> 1
```

OR : ||

OR จะส่งค่า 0 ถ้ามีอาร์กิวเมนต์ที่เป็น 1 หรือ NULL มิฉะนั้นจะส่งค่า 1 ตัวอย่างเช่น

```
mysql> select 1 || 0;  
-> 1  
mysql> select 0 || 0;  
-> 0  
mysql> select 1 || NULL;  
-> 1
```

AND : &&

AND จะส่งค่า 0 ถ้ามีอาร์กิวเมนต์ที่เป็น 0 หรือ NULL มิฉะนั้นจะส่งค่า 1 ตัวอย่างเช่น

```
mysql> select 1 && NULL;
```

```
-> 0
mysql> select 1 && 0;
-> 0
```

5.3.5 Comparison operators

ผลการเปรียบเทียบได้ค่าเป็น 1 หรือ 0 หรือ NULL เท่านั้น ฟังก์ชันเหล่านี้ใช้ได้กับตัวเลขและตัวอักษร ถ้าเป็นตัวอักษรมันจะถูกแปลงให้เป็นตัวเลขและจากตัวเลขเป็นตัวอักษรถ้าจำเป็น

MySQL จะทำการเปรียบเทียบโดยอาศัยกฎต่อไปนี้

- ถ้าอาร์กิวเมนต์หนึ่งตัวหรือมากกว่าเป็น NULL ผลลัพธ์จะออกมาเป็น NULL ยกเว้นตัวกระทำ <=>
- ถ้าอาร์กิวเมนต์ทั้งสองตัวในการเปรียบเทียบเป็นตัวอักษร มันก็จะถูกเปรียบเทียบเป็นตัวอักษร
- ถ้าอาร์กิวเมนต์ทั้งสองตัวเป็นตัวเลข ก็จะเปรียบเทียบเป็นตัวเลข
- ค่าเลขฐานสิบหกจะถูกทำเป็นเลขฐานสองของตัวอักษรในกรณีที่ไม่ได้ทำการเปรียบเทียบกับตัวเลข
- ถ้าอาร์กิวเมนต์ตัวหนึ่งเป็นคอลัมน์ TIMESTAMP หรือ DATETIME และอาร์กิวเมนต์ตัวอื่นเป็นค่าคงที่ ค่าคงที่จะถูกแปลงเป็น timestamp ก่อนที่จะทำการเปรียบเทียบ

ตัวอย่าง

```
mysql> SELECT 1 > '6x';
-> 0
mysql> SELECT 7 > '6x';
-> 1
mysql> SELECT 0 > 'x6';
-> 0
mysql> SELECT 0 = 'x6';
-> 1
```

ตัวกระทำเปรียบเทียบ ได้แก่

= : Equal

```
mysql> select 1 = 0;
-> 0
mysql> select '0' = 0;
-> 1
mysql> select '0.0' = 0;
```



```
-> 1
mysql> select '0.01' = 0;
-> 0
mysql> select '.01' = 0.01;
-> 1
```

<> : != : Not equal

```
mysql> select '.01' <> '0.01';
-> 1
mysql> select .01 <> '0.01';
-> 0
mysql> select 'zapp' <> 'zappp';
-> 1
```

<= : Less than or equal

```
mysql> select 0.1 <= 2;
-> 1
```

< : Less than

```
mysql> select 2 <= 2;
-> 1
```

>= : Greater than or equal

```
mysql> select 2 >= 2;
-> 1
```

> : Greater than

```
mysql> select 2 > 2;
-> 0
```

<=> : Null safe equal

```
mysql> select 1 <=> 1, NULL <=> NULL, 1 <=> NULL;
-> 1 1 0
```

IS NULL และ IS NOT NULL

ใช้ตรวจสอบว่าค่าเป็น NULL หรือ NOT NULL

```
mysql> select 1 IS NULL, 0 IS NULL, NULL IS NULL;
-> 0 0 1

mysql> select 1 IS NOT NULL, 0 IS NOT NULL, NULL IS NOT NULL;
-> 1 1 0
```

expr BETWEEN min AND max

ถ้า expr มีค่ามากกว่าหรือเท่ากับค่า min และ expr มีค่าน้อยกว่าหรือเท่ากับค่า max BETWEEN จะส่งค่า 1 มิฉะนั้นเป็น 0

การเปรียบเทียบแบบนี้มีค่าเท่ากับนิพจน์ (min <= expr AND expr <= max)

```
mysql> select 1 BETWEEN 2 AND 3;
-> 0

mysql> select 'b' BETWEEN 'a' AND 'c';
-> 1

mysql> select 2 BETWEEN 2 AND '3';
-> 1

mysql> select 2 BETWEEN 2 AND 'x-3';
-> 0
```

expr IN (value,...)

การเปรียบเทียบนี้จะส่งค่า 1 ถ้า expr คือค่าใดๆ ในรายการที่อยู่ข้างหลังคำว่า IN มิฉะนั้นก็จะส่งค่า 0 ตัวอย่าง

```
mysql> select 2 IN (0,3,5,'wefwf');
-> 0

mysql> select 'wefwf' IN (0,3,5,'wefwf');
-> 1
```

expr NOT IN (value,...)

มีค่าเท่ากับ NOT (expr IN (value,...)).

ISNULL(expr)

ถ้า expr เป็น NULL, ISNULL() จะส่งค่า 1, มิฉะนั้นจะส่งค่า 0.

```
mysql> select ISNULL(1+1);  
-> 0  
mysql> select ISNULL(1/0);  
-> 1
```

COALESCE(list)

ส่งค่าของตัวแรกในวงเล็บที่ไม่ใช่ NULL ตัวอย่าง

```
mysql> select COALESCE(NULL,1);  
-> 1  
mysql> select COALESCE(NULL,NULL,NULL);  
-> NULL
```

INTERVAL(N,N1,N2,N3,...)

ส่งกลับค่า 0 ถ้า $N < N1$, และ 1 ถ้า $N < N2$ อาร์กิวเมนต์ทุกตัวจะถูกมองเป็นตัวเลขจำนวนเต็ม และมีข้อกำหนดว่า $N1 < N2 < N3 < \dots < Nn$ เพื่อให้ฟังก์ชันนี้ทำงานได้อย่างถูกต้อง ตัวอย่าง

```
mysql> select INTERVAL(23, 1, 15, 17, 30, 44, 200);  
-> 3  
mysql> select INTERVAL(10, 1, 10, 100, 1000);  
-> 2  
mysql> select INTERVAL(22, 23, 30, 44, 200);  
-> 0
```

5.3.6 การเปรียบเทียบสตริง

โดยทั่วไปแล้ว การเปรียบเทียบสตริงจะแบ่งแยกตัวใหญ่ตัวเล็ก

การเปรียบเทียบสตริงจะได้ค่า 0(เท็จ) หรือ 1 (จริง) โดยการใช้ LIKE ในการเปรียบเทียบ คุณ
สามารถใช้อักขระ ต่อไปในรูปแบบการเปรียบเทียบได้

%	แทนตัวอักษรตั้งแต่ศูนย์ตัวขึ้นไป
_	แทนอักขระตัวเดียว

ตัวอย่าง

```
mysql> select 'David!' LIKE 'David_';
-> 1
mysql> select 'David!' LIKE '%D%v%';
-> 1
```

ถ้าต้องการเปรียบเทียบอักขระตัวเดียวด้วยการระบุอักขระ ให้ใช้ “\” นำหน้าอักขระที่ต้องการจะเปรียบเทียบในรูปแบบ

\%	แทนอักขระ % หนึ่งตัว
_	แทนอักขระ _ หนึ่งตัว

ตัวอย่าง

```
mysql> select 'David!' LIKE 'David\%';
-> 0
mysql> select 'David\_!' LIKE 'David\%';
-> 1
```

ในการกำหนดอักขระ ESCAPE ตัวอื่น ให้ใช้ประโยค ESCAPE ในการกำหนด เช่น

```
mysql> select 'David\_!' LIKE 'David|\_ ESCAPE '|';
-> 1
```

expr NOT LIKE pat [ESCAPE 'escape-char'] มีความหมายเช่นเดียวกับประโยค

NOT (expr LIKE pat [ESCAPE 'escape-char']).

expr REGEXP pat

expr RLIKE pat

ทั้งสี่ประโยคข้างต้นนี้ ใช้ในการเปรียบเทียบนิพจน์สตริง expr กับรูปแบบ pat RLIKE เป็นคำเหมือนของ REGEXP ใช้ได้กับ mSQL ตัวอย่าง

```
mysql> select 'Monty!' REGEXP 'm%y%%';
-> 0
mysql> select 'Monty!' REGEXP '!.*';
```

```
-> 1
mysql> select 'new\n*line' REGEXP 'new\\.*\\*line';
-> 1
mysql> select "a" REGEXP "A", "a" REGEXP BINARY "A";
-> 1 0
```

expr NOT REGEXP pat

expr NOT RLIKE pat

มีความหมายเหมือนกับประโยค NOT (expr REGEXP pat).

STRCMP(expr1,expr2)

STRCMP()

จะให้ค่า 0 ถ้าสตริงเหมือนกัน ให้ค่า -1 ถ้าอาร์กิวเมนต์ตัวแรกมีค่าน้อยกว่าตัวที่สอง มิฉะนั้นได้ค่า 1 ตัวอย่าง

```
mysql> select STRCMP('text', 'text2');
-> -1
mysql> select STRCMP('text2', 'text');
-> 1
mysql> select STRCMP('text', 'text');
-> 0
```

5.3.7 Cast operators

BINARY

ตัวกระทำ BINARY ใช้ในการกำหนดสตริงที่ต่อท้ายให้เป็นเลขฐานสองของสตริง วิธีช่วยให้การเปรียบเทียบคอลัมน์เป็นแบบ case sensitive ตัวอย่าง

```
mysql> select "a" = "A";
-> 1
mysql> select BINARY "a" = "A";
-> 0
```

ชั้นควบคุมทิศทางการทำงาน

IFNULL(expr1,expr2)

ถ้า expr1 ไม่ใช่ NULL, IFNULL() จะให้ค่า expr1, มิฉะนั้นจะให้ค่า expr2. IFNULL() จะให้ค่า ตัวเลขหรือสตริง ขึ้นอยู่กับว่าใช้แบบไหน ตัวอย่าง

```
mysql> select IFNULL(1,0);
-> 1
mysql> select IFNULL(0,10);
-> 0
mysql> select IFNULL(1/0,10);
-> 10
mysql> select IFNULL(1/0,'yes');
-> 'yes'
```

IF(expr1,expr2,expr3)

ถ้า expr1 เป็นจริง (TRUE) โดยที่ (expr1 <> 0 and expr1 <> NULL) แล้ว IF() จะให้ค่า expr2, มิฉะนั้นจะให้ค่า expr3. IF() จะให้ค่าตัวเลขหรือสตริงขึ้นอยู่กับว่าค่าไหนถูกใช้ ตัวอย่าง

```
mysql> select IF(1>2,2,3);
-> 3
mysql> select IF(1<2,'yes','no');
-> 'yes'
mysql> select IF(strcmp('test','test1'),'yes','no');
-> 'no'
```

expr1 ถูกคำนวณโดยใช้เป็นเลขจำนวนเต็ม ซึ่งหมายความว่าถ้าคุณใช้เป็น floating-point หรือ สตริงคุณก็สามารถใช้ฟังก์ชันนี้ด้วยการเปรียบเทียบได้ ตัวอย่าง

```
mysql> select IF(0.1,1,0);
-> 0
mysql> select IF(0.1<>0,1,0);
-> 1
```


5.3.9 ฟังก์ชันทางคณิตศาสตร์

ฟังก์ชันทางคณิตศาสตร์ทั้งหมดจะให้ค่าเป็น NULL หากเกิดข้อผิดพลาดขึ้น

- : เครื่องหมายลบ เปลี่ยนค่าเลขลบให้เป็นบวก และเปลี่ยนค่าบวกให้เป็นลบ ตัวอย่าง

```
mysql> select - 2;  
-> -2
```

ABS(X)

ให้ค่าสัมบูรณ์ของ X ตัวอย่างเช่น

```
mysql> select ABS(2);  
-> 2  
mysql> select ABS(-32);  
-> 32
```

SIGN(X)

ให้ค่าบวกหรือลบของอาร์กิวเมนต์ ขึ้นอยู่กับว่าค่านั้นเป็นลบหรือบวก ตัวอย่าง

```
mysql> select SIGN(-32);  
-> -1  
mysql> select SIGN(0);  
-> 0  
mysql> select SIGN(234);  
-> 1
```

MOD(N,M) : %

ให้ค่าเศษจากการหาร N ด้วย M ตัวอย่าง เช่น

```
mysql> select MOD(234, 10);  
-> 4  
mysql> select 253 % 7;  
-> 1  
mysql> select MOD(29,9);  
-> 2
```

FLOOR(X)

ให้ค่าเลขจำนวนเต็มที่ใหญ่ที่สุดที่ไม่มากกว่า X ตัวอย่างเช่น

```
mysql> select FLOOR(1.23);  
-> 1  
mysql> select FLOOR(-1.23);  
-> -2
```

CEILING(X)

ให้ค่าเลขจำนวนเต็มที่เล็กที่สุดที่ไม่น้อยกว่า X ตัวอย่างเช่น

```
mysql> select CEILING(1.23);  
-> 2  
mysql> select CEILING(-1.23);  
-> -1
```

ROUND(X)

ให้ค่าเลขจำนวนเต็มที่เอา X มาปัดเศษแล้ว ตัวอย่างเช่น

```
mysql> select ROUND(-1.23);  
-> -1  
mysql> select ROUND(-1.58);  
-> -2  
mysql> select ROUND(1.58);  
-> 2
```

ROUND(X,D)

ให้ค่าเลข X ที่มีจำนวนทศนิยมเท่ากับ D ตำแหน่ง ตัวอย่างเช่น

```
mysql> select ROUND(1.298, 1);  
-> 1.3  
mysql> select ROUND(1.298, 0);  
-> 1
```

EXP(X)

ให้ค่าเลขชี้โพเนนเชียลของ X เช่น

```
mysql> select EXP(2);  
-> 7.389056  
mysql> select EXP(-2);  
-> 0.135335
```

LOG(X)

ให้ค่าลอการิทึมของ X ตัวอย่างเช่น

```
mysql> select LOG(2);  
-> 0.693147  
mysql> select LOG(-2);  
-> NULL
```

LOG10(X)

ให้ค่าลอการิทึมฐาน 10 ของ X ตัวอย่างเช่น

```
mysql> select LOG10(2);  
-> 0.301030  
mysql> select LOG10(100);  
-> 2.000000  
mysql> select LOG10(-100);  
-> NULL
```

POW(X,Y)

POWER(X,Y)

ให้ค่าของ X ยกกำลัง Y ตัวอย่างเช่น

```
mysql> select POW(2,2);  
-> 4.000000  
mysql> select POW(2,-2);  
-> 0.250000
```


SQRT(X)

ให้ค่าสแควร์ทของ X ตัวอย่างเช่น

```
mysql> select SQRT(4);  
-> 2.000000  
mysql> select SQRT(20);  
-> 4.472136
```

PI()

ให้ค่าของ PI ตัวอย่างเช่น

```
mysql> select PI();  
-> 3.141593
```

COS(X)

ให้ค่าโคซายของ X, เมื่อ X มีหน่วยเป็นเรเดียน ตัวอย่าง

```
mysql> select COS(PI());  
-> -1.000000
```

SIN(X)

ให้ค่าซายของ X, เมื่อ X มีหน่วยเป็นเรเดียน ตัวอย่าง

```
mysql> select SIN(PI());  
-> 0.000000
```

TAN(X)

ให้ค่าแทนของ X, เมื่อ X มีหน่วยเป็นเรเดียน ตัวอย่าง

```
mysql> select TAN(PI()+1);  
-> 1.557408
```

ACOS(X)

ให้ค่าอาร์คโคซของ X ตัวอย่าง

```
mysql> select ACOS(1);  
-> 0.000000  
mysql> select ACOS(1.0001);  
-> NULL  
mysql> select ACOS(0);  
-> 1.570796
```

ASIN(X)

ให้ค่าอาร์คซายของ X ตัวอย่างเช่น

```
mysql> select ASIN(0.2);  
-> 0.201358  
mysql> select ASIN('foo');  
-> 0.000000
```

ATAN(X)

ให้ค่าอาร์คแทนเจนต์ของ X ตัวอย่าง

```
mysql> select ATAN(2);  
-> 1.107149  
mysql> select ATAN(-2);  
-> -1.107149
```

ATAN2(X,Y)

ให้ค่าอาร์คแทนเจนต์ของตัวแปร X และ Y ตัวอย่างเช่น

```
mysql> select ATAN(-2,2);  
-> -0.785398  
mysql> select ATAN(PI(),0);  
-> 1.570796
```

COT(X)

ให้ค่าโคแทนเจนต์ของ X ตัวอย่างเช่น

```
mysql> select COT(12);  
-> -1.57267341  
mysql> select COT(0);  
-> NULL
```

RAND()

RAND(N)

ให้ค่าเลข floating-point สุ่มในช่วงของ 0 ถึง 1.0 ค่า N ถ้าใส่จะเป็น seed value ตัวอย่างเช่น

```
mysql> select RAND();  
-> 0.5925  
mysql> select RAND(20);  
-> 0.1811  
mysql> select RAND(20);  
-> 0.1811  
mysql> select RAND();  
-> 0.2079  
mysql> select RAND();  
-> 0.7888
```

5.3.1. ฟังก์ชันเกี่ยวกับสตริง

ASCII(str)

Return รหัส ASCII ของตัวอักษรที่อยู่ซ้ายสุดของ string str

Return NULL ถ้า str เป็น empty string

```
mysql> select ASCII('2');  
-> 50  
mysql> select ASCII(2);  
-> 50  
mysql> select ASCII('dx');  
-> 100
```


BIN(N)

Return ค่า Binary ของ N ซึ่ง N สามารถมีค่าได้เท่ากับ BIGINT

Returns NULL ถ้า N เป็น NULL

```
mysql> select BIN(12);  
-> '1100'
```

OCT(N)

Return ค่า Octal ของ N ซึ่ง N สามารถมีค่าได้เท่ากับ BIGINT

Return NULL ถ้า N เป็น NULL

```
mysql> select OCT(12);  
-> '14'
```

HEX(N)

Return ค่า Hexadecimal ของ N ซึ่ง N สามารถมีค่าได้เท่ากับ BIGINT

Return NULL ถ้า N เป็น NULL

```
mysql> select HEX(255);  
-> 'FF'
```

CHAR(N,...)

Return string ของรหัส ASCII N

```
mysql> select CHAR(77,121,83,81,'76');  
-> 'MySQL'  
mysql> select CHAR(77,77.3,'77.3');  
-> 'MMM'
```

LENGTH(str)

Return ความยาวของ string str

```
mysql> select LENGTH('text');  
-> 4
```

LOCATE(substr,str)

POSITION(substr IN str)

Return ตำแหน่งของ ตัวอักษรตัวแรกของ substring substr ใน string str.

Return 0 ถ้า substr ไม่อยู่ใน str

```
mysql> select LOCATE('bar', 'foobarbar');
-> 4
mysql> select LOCATE('xbar', 'foobar');
-> 0
```

LOCATE(substr,str,pos)

Return ตำแหน่งของ ตัวอักษรตัวแรกของ substring substr ใน string str โดยเริ่มค้นหาตั้งแต่ตำแหน่ง pos

Return 0 ถ้า substr ไม่อยู่ใน str

```
mysql> select LOCATE('bar', 'foobarbar',5);
-> 7
```

INSTR(str,substr)

เหมือนกับ LOCATE เพียงแต่สลับที่ argument

Return ตำแหน่งของ ตัวอักษรตัวแรกของ substring substr ใน string str

Return 0 ถ้า substr ไม่อยู่ใน str

```
mysql> select INSTR('foobarbar', 'bar');
-> 4
mysql> select INSTR('xbar', 'foobar');
-> 0
```

LPAD(str,len,padstr)

Return string str ซึ่งจะเติมตัวอักษรทางซ้ายของ str ด้วยเครื่องหมาย padstr จน string มีความยาวเท่ากับ len ตัวอักษร

```
mysql> select LPAD('hi',4,'?');
-> '??hi'
```

RPAD(str,len,padstr)

Return string str ซึ่งจะเติมตัวอักษรทางขวาของ str ด้วยเครื่องหมาย padstr จน string มีความยาวเท่ากับ len ตัวอักษร

```
mysql> select RPAD('hi',5,'?');  
-> 'hi???'
```

LEFT(str,len)

Return ตัวอักษรที่อยู่ด้านซ้ายสุดของ string str จำนวน len ตัวอักษร

```
mysql> select LEFT('foobarbar', 5);  
-> 'fooba'
```

RIGHT(str,len)

Return ตัวอักษรที่อยู่ด้านขวาสุดของ string str จำนวน len ตัวอักษร

```
mysql> select RIGHT('foobarbar', 4);  
-> 'rbar'
```

SUBSTRING(str,pos,len)

SUBSTRING(str FROM pos FOR len)

MID(str,pos,len)

Return string ที่มีความยาวเท่ากับ len ตัวอักษร จาก str เริ่มจากตำแหน่ง pos

```
mysql> select SUBSTRING('Quadratically',5,6);  
-> 'ratica'
```

SUBSTRING(str,pos)

SUBSTRING(str FROM pos)

Returns substring จาก str โดยเริ่มตั้งแต่ตำแหน่งที่ pos

```
mysql> select SUBSTRING('Quadratically',5);  
-> 'ratically'  
mysql> select SUBSTRING('foobarbar' FROM 4);  
-> 'barbar'
```


LTRIM(str)

Returns str ที่ไม่มีช่องว่างอยู่ด้านหน้า str

```
mysql> select LTRIM(' barbar');  
-> 'barbar'
```

RTRIM(str)

Returns str ที่ไม่มีช่องว่างอยู่ด้านหลัง str

```
mysql> select RTRIM('barbar ');  
-> 'barbar'
```

TRIM([BOTH | LEADING | TRAILING] [remstr] FROM) str)

Return str ที่ถูกตัดตัวอักษร remstr ที่อยู่ด้านหน้า(LEADING) หรือ ด้านหลัง (TRAILING) หรือทั้ง 2 ด้านของ str ออกไปแล้ว

```
mysql> select TRIM(' bar ');  
-> 'bar'  
mysql> select TRIM(LEADING 'x' FROM 'xxxbarxxx');  
-> 'barxxx'  
mysql> select TRIM(BOTH 'x' FROM 'xxxbarxxx');  
-> 'bar'  
mysql> select TRIM(TRAILING 'xyz' FROM 'barxyz');  
-> 'barx'
```

LCASE(str)

LOWER(str)

Return str ที่เป็นตัวอักษรพิมพ์เล็กทั้งหมด

```
mysql> select LCASE('QUADRATICALLY');  
-> 'quadratically'
```

UCASE(str)

UPPER(str)

Return str ที่เป็นตัวอักษรพิมพ์เล็กทั้งหมด

```
mysql> select UCASE('Hej');  
-> 'HEJ'
```

5.3.2. ฟังก์ชันเกี่ยวกับวันที่ และ เวลา

DAYOFMONTH(date)

Return วันที่ใน 1 เดือน (1 ถึง 31)

```
mysql> select DAYOFMONTH('1998-02-03');  
-> 3
```

DAYOFYEAR(date)

Return วันที่ ใน 1 ปี (1 ถึง 366)

```
mysql> select DAYOFYEAR('1998-02-03');  
-> 34
```

MONTH(date)

Return เดือนที่ใน 1 ปี (1 ถึง 12)

```
mysql> select MONTH('1998-02-03');  
-> 2
```

DAYNAME(date)

Return ชื่อวัน ใน 1 อาทิตย์

```
mysql> select DAYNAME('1998-02-05');  
-> 'Thursday'
```

MONTHNAME(date)

Return ชื่อเดือน

```
mysql> select MONTHNAME('1998-02-05');  
-> 'February'
```

YEAR(date)

Return ปีสำหรับวันที่ ซึ่งอยู่ในช่วง 1000-9999

```
mysql> select YEAR('98-02-03');  
-> 1998
```

HOUR(time)

Return ชั่วโมง จาก 0 ถึง 23

```
mysql> select HOUR('10:05:03');  
-> 10
```

MINUTE(time)

Return นาที จาก 0 ถึง 59

```
mysql> select MINUTE('98-02-03 10:05:03');  
-> 5
```

SECOND(time)

Return วินาที จาก 0 ถึง 59

```
mysql> select SECOND('10:05:03');  
-> 3
```

DATE_FORMAT(date,format)

รูปแบบของวันที่ที่ MySQL สามารถแสดงได้ ซึ่งใช้ string ในตารางข้างล่างเป็นตัวกำหนดรูปแบบ

%M	Month name (January..December)
%W	Weekday name (Sunday..Saturday)
%D	Day of the month with english suffix (1st, 2nd, 3rd, etc.)
%Y	Year, numeric, 4 digits
%y	Year, numeric, 2 digits
%X	Year for the week where Sunday is the first day of the week, numeric, 4 digits, used with '%V'
%x	Year for the week, where Monday is the first day of the week, numeric, 4 digits, used with '%v'
%a	Abbreviated weekday name (Sun..Sat)
%d	Day of the month, numeric (00..31)
%e	Day of the month, numeric (0..31)
%m	Month, numeric (01..12)

%c	Month, numeric (1..12)
%b	Abbreviated month name (Jan..Dec)
%j	Day of year (001..366)
%H	Hour (00..23)
%k	Hour (0..23)
%h	Hour (01..12)
%l	Hour (01..12)
%l	Hour (1..12)
%i	Minutes, numeric (00..59)
%r	Time, 12-hour (hh:mm:ss [AP]M)
%T	Time, 24-hour (hh:mm:ss)
%S	Seconds (00..59)
%s	Seconds (00..59)
%p	AM or PM
%w	Day of the week (0=Sunday..6=Saturday)
%U	Week (0..53), where Sunday is the first day of the week
%u	Week (0..53), where Monday is the first day of the week
%V	Week (1..53), where Sunday is the first day of the week. Used with '%X'
%v	Week (1..53), where Monday is the first day of the week. Used with '%x'
%%	A literal `%'.

ตัวอย่าง

```
mysql> select DATE_FORMAT('1997-10-04 22:23:00', '%W %M %Y');
-> 'Saturday October 1997'
mysql> select DATE_FORMAT('1997-10-04 22:23:00', '%H:%i:%s');
-> '22:23:00'
mysql> select DATE_FORMAT('1997-10-04 22:23:00',
        '%D %y %a %d %m %b %j');
-> '4th 97 Sat 04 10 Oct 277'
mysql> select DATE_FORMAT('1997-10-04 22:23:00',
        '%H %k %l %r %T %S %w');
-> '22 22 10 10:23:00 PM 22:23:00 00 6'
mysql> select DATE_FORMAT('1999-01-01', '%X %V');
-> '1998 52'
```

TIME_FORMAT(time,format)

ใช้เหมือนกับ DATE_FORMAT แต่ใช้กับเวลา

CURDATE()

CURRENT_DATE

Return วันที่ปัจจุบัน

```
mysql> select CURDATE();  
-> '2000-02-05'
```

CURTIME()

CURRENT_TIME

Return เวลาปัจจุบัน

```
mysql> select CURTIME();  
-> '23:50:26'
```

```
mysql> select CURTIME() + 0;  
-> 235026
```

NOW()

SYSDATE()

CURRENT_TIMESTAMP

Return ทั้งวันที่ และ เวลา ในรูปแบบ 'YYYY-MM-DD HH:MM:SS' หรือ YYYYMMDDHHMMSS

```
mysql> select NOW();  
-> '2000-02-05 23:50:26'  
mysql> select NOW() + 0;  
-> 20000205235026
```

5.4. CREATE DATABASE syntax

รูปแบบของคำสั่ง CREATE DATABASE มีดังนี้

```
CREATE DATABASE db_name
```

สำหรับ MySQL , database จะถูกสร้างเป็น directory อยู่ภายใต้ data directory ของ MySQL ซึ่ง file ใน directory ก็คือ table ใน database นั้นเอง.

5.5. DROP DATABASE syntax

รูปแบบคำสั่ง DROP DATABASE มีดังนี้

```
DROP DATABASE [IF EXISTS] db_name
```

Drop database เป็นการลบทุกๆ ตารางใน database และ ลบ database ด้วย คำที่ส่งกลับมาจากคำสั่ง drop database คือ จำนวน file ที่ถูกลบจาก database directory ซึ่งโดยปกติแล้ว จำนวน file ที่ถูกลบจะเท่ากับ 3 เท่าของจำนวนตาราง นั่นคือ 1 ตาราง จะประกอบด้วย file นามสกุล .MYD , .MYI และ .frm

5.6. CREATE TABLE syntax

รูปแบบคำสั่งของการสร้างตารางด้วยคำสั่ง CREATE TABLE มีดังนี้

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name [(create_definition,...)]  
[table_options] [select_statement]
```

รูปแบบของ create_definition ได้แก่

```
col_name type [NOT NULL | NULL] [DEFAULT default_value] [AUTO_INCREMENT]  
[PRIMARY KEY] [reference_definition]  
หรือ PRIMARY KEY (index_col_name,...)  
หรือ KEY [index_name] (index_col_name,...)  
หรือ INDEX [index_name] (index_col_name,...)  
หรือ UNIQUE [INDEX] [index_name] (index_col_name,...)  
หรือ [CONSTRAINT symbol] FOREIGN KEY index_name (index_col_name,...)  
[reference_definition]  
หรือ CHECK (expr)
```

รูปแบบของ type ได้แก่

```
TINYINT[(length)] [UNSIGNED] [ZEROFILL]  
หรือ SMALLINT[(length)] [UNSIGNED] [ZEROFILL]  
หรือ MEDIUMINT[(length)] [UNSIGNED] [ZEROFILL]  
หรือ INT[(length)] [UNSIGNED] [ZEROFILL]  
หรือ INTEGER[(length)] [UNSIGNED] [ZEROFILL]  
หรือ BIGINT[(length)] [UNSIGNED] [ZEROFILL]  
หรือ REAL[(length,decimals)] [UNSIGNED] [ZEROFILL]  
หรือ DOUBLE[(length,decimals)] [UNSIGNED] [ZEROFILL]  
หรือ FLOAT[(length,decimals)] [UNSIGNED] [ZEROFILL]  
หรือ DECIMAL(length,decimals) [UNSIGNED] [ZEROFILL]  
หรือ NUMERIC(length,decimals) [UNSIGNED] [ZEROFILL]  
หรือ CHAR(length) [BINARY]  
หรือ VARCHAR(length) [BINARY]
```


หรือ DATE
 หรือ TIME
 หรือ TIMESTAMP
 หรือ DATETIME
 หรือ TINYBLOB
 หรือ BLOB
 หรือ MEDIUMBLOB
 หรือ LONGBLOB
 หรือ TINYTEXT
 หรือ TEXT
 หรือ MEDIUMTEXT
 หรือ LONGTEXT
 หรือ ENUM(value1,value2,value3,...)
 หรือ SET(value1,value2,value3,...)

รูปแบบของ index_col_name:

col_name [(length)]

รูปแบบของ reference_definition:

REFERENCES tbl_name [(index_col_name,...)]
 [MATCH FULL | MATCH PARTIAL]
 [ON DELETE reference_option]
 [ON UPDATE reference_option]

รูปแบบ reference_option:

RESTRICT | CASCADE | SET NULL | NO ACTION | SET DEFAULT

รูปแบบของ table_options:

TYPE = {ISAM | MYISAM | HEAP}
 หรือ AUTO_INCREMENT = #
 หรือ AVG_ROW_LENGTH = #
 หรือ CHECKSUM = {0 | 1}
 หรือ COMMENT = "string"
 หรือ MAX_ROWS = #
 หรือ MIN_ROWS = #
 หรือ PACK_KEYS = {0 | 1}
 หรือ PASSWORD = "string"
 หรือ DELAY_KEY_WRITE = {0 | 1}
 หรือ ROW_FORMAT= { default | dynamic | static | compressed }

รูปแบบของ select_statement:

```
[IGNORE | REPLACE] SELECT ... (Some legal select statement)
```

CREATE TABLE ใช้ในการสร้างตารางตามชื่อที่กำหนดไว้ในฐานข้อมูลที่กำลังเปิดอยู่ กฎการตั้งชื่อตารางอนุญาตให้ตั้งชื่อที่มีความยาวได้ไม่เกิน 64 ตัวอักษร และห้ามใช้อักขระ "/" หรือ "." ในชื่อตาราง อาจเกิดข้อความแสดงข้อผิดพลาดได้ถ้าตารางอ้างอิงไม่ได้อยู่ในฐานข้อมูลปัจจุบัน

ใน MySQL 3.22 หรือหลังจากนั้น ชื่อตารางสามารถถูกกำหนดเป็น db_name.tbl.name วิธีนี้ใช้ได้ทั้งฐานข้อมูลที่เปิดอยู่และที่ยังไม่ได้เปิด ใน MySQL 3.23 คุณสามารถใช้ Keyword ชื่อตารางตอนสร้างตารางได้ ตารางชื่อตารางจะถูกกลบโดยอัตโนมัติถ้าการเชื่อมต่อมีปัญหาและในการเชื่อมต่อแต่ละครั้งชื่อถูกใช้เฉพาะในการเชื่อมต่อครั้งนั้นๆ หมายความว่า การเชื่อมต่อสองครั้งที่แตกต่างกันจะสามารถใช้ชื่อตารางชื่อตารางที่เหมือนกันได้โดยไม่เกิดการซ้ำซ้อน คือตารางที่มีอยู่จะถูกซ่อนไว้จนกว่าตารางชื่อตารางจะถูกกลบ ดังนั้นตารางชื่อตารางจึงสามารถมีชื่อซ้ำกับตารางที่มีอยู่แล้วได้

ใน MySQL 3.23 หรือหลังจากนั้น คุณสามารถใช้ keyword IF NOT EXISTS ในการกำจัดข้อผิดพลาดถ้าหากว่ามีตารางนั้นอยู่ สังเกตว่าไม่มีกรตรวจสอบว่าโครงสร้างตารางเหมือนกันหรือไม่ แต่ละตาราง tbl_name ถูกแทนด้วยบางไฟล์ในไดเรกทอรีฐานข้อมูล ตัวอย่าง ตารางชนิด MyISAM จะมี

ไฟล์	จุดมุ่งหมาย
tbl_name.frm	Table definition (form) file
tbl_name.MYD	Data file
tbl_name.MYI	Index file

สำหรับข้อมูลเพิ่มเติมในเรื่องคุณสมบัติของคอลัมน์ชนิดต่างๆ กัน อ่านได้ในหัวข้อ Column types

5.7. ALTER TABLE syntax

รูปแบบคำสั่งของ ALTER TABLE เป็นดังนี้

```
ALTER [IGNORE] TABLE tbl_name alter_spec [, alter_spec ...]
```

รูปแบบของ alter_specification:

```
ADD [COLUMN] create_definition [FIRST | AFTER column_name ]
หรือ ADD INDEX [index_name] (index_col_name,...)
หรือ ADD PRIMARY KEY (index_col_name,...)
หรือ ADD UNIQUE [index_name] (index_col_name,...)
หรือ ALTER [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}
หรือ CHANGE [COLUMN] old_col_name create_definition
```

หรือ MODIFY [COLUMN] create_definition
 หรือ DROP [COLUMN] col_name
 หรือ DROP PRIMARY KEY
 หรือ DROP INDEX index_name
 หรือ RENAME [AS] new_tbl_name
 หรือ table_options

ALTER TABLE ใช้ในการเปลี่ยนแปลงโครงสร้างของตารางที่มีอยู่ ตัวอย่างเช่น คุณสามารถจะเพิ่มหรือลบคอลัมน์ สร้างหรือทำลายดรรชนี เปลี่ยนชนิดของคอลัมน์ที่มีอยู่แล้ว หรือเปลี่ยนชื่อคอลัมน์หรือแม้แต่ชื่อตารางเอง คุณสามารถเปลี่ยนคอมเมนต์ของตารางและชนิดของตารางได้ด้วย

ถ้าคุณใช้ ALTER TABLE ในการเปลี่ยนข้อกำหนดของคอลัมน์แต่ DESCRIBE tbl_name กำหนดไว้ว่าคอลัมน์ของคุณต้องไม่ถูกเปลี่ยนแปลงใดๆ ก็เป็นไปได้ที่ MySQL จะไม่เปลี่ยนแปลงคอลัมน์ให้ตามคำสั่งของคุณ ยกตัวอย่าง ถ้าคุณพยายามจะเปลี่ยนตัวแปรแบบ VARCHAR ไปเป็น CHAR MySQL จะยังคงใช้ VARCHAR ถ้าตารางมีคอลัมน์ที่ความยาวสามารถแปรผันได้

ALTER TABLE ทำงานโดยการสร้างชุดจำลองชั่วคราวของตารางต้นแบบ การเปลี่ยนแปลงจะถูกทำบนชุดจำลองนี้ จากนั้นตารางต้นแบบจะถูกลบและเปลี่ยนชื่อตารางใหม่ให้เหมือนอันเก่า ดังนั้นการปรับปรุงใดๆ จะถูกประยุกต์โดยตรงกับตารางใหม่นี้โดยไร้ข้อผิดพลาดใดๆ ทั้งสิ้น ขณะที่ ALTER TABLE กำลังทำงานตารางต้นแบบจะถูกอ่านโดยไคลเอนท์อื่นๆ การปรับปรุงและการเขียนลงตารางจะถูกหยุดไว้จนกว่าตารางใหม่จะพร้อมใช้

- ในการใช้ ALTER TABLE คุณจำเป็นต้องใช้ select, insert, delete, update, create และ drop กับตาราง
- IGNORE เป็นส่วนขยายของ MySQL บน ANSI SQL92 ถูกใช้ในการควบคุมว่า ALTER TABLE จะทำงานได้อย่างไร ถ้ามีการซ้ำซ้อนของคีย์หลักในตารางใหม่ ถ้าไม่มีการระบุ IGNORE ตารางจำลองจะถูกยกเลิกและการเปลี่ยนแปลงใดๆ จะสูญหาย แต่ถ้ามีการระบุ IGNORE แถวที่มีการซ้ำซ้อนบนคีย์หลักจะถูกเลือกเฉพาะแถวแรก ส่วนแถวอื่นถูกลบ
- คุณสามารถใช้ประโยค ADD, ALTER, DROP และ CHANGE ได้หลายประโยคในประโยค ALTER TABLE ประโยคเดียว
- MODIFY เป็นส่วนขยายของ ORACLE ที่ใช้กับ ALTER TABLE
- คุณสามารถเปลี่ยนชื่อคอลัมน์โดยใช้ประโยค CHANGE old_col_name create_definition ในการทำเช่นนี้ต้องกำหนดชื่อคอลัมน์เก่าและใหม่และชนิดของคอลัมน์ปัจจุบันไว้ด้วย ตัวอย่างเมื่อต้องการเปลี่ยนชื่อคอลัมน์จาก a เป็น b

```
mysql> ALTER TABLE t1 CHANGE a b INTEGER;
```

 เมื่อต้องการเปลี่ยนชนิดคอลัมน์แต่คงชื่อเดิม

```
mysql> ALTER TABLE t1 CHANGE a b INTEGER;
```

 ใน MySQL 3.22.16a คุณสามารถใช้ MODIFY ในการเปลี่ยนชนิดคอลัมน์โดยไม่ต้องเปลี่ยนชื่อ

```
mysql> ALTER TABLE t1 MODIFY b BIGINT NOT NULL;
```


subnet mask นี้จะไม่ถูกส่งไปด้วย โฮสต์ทั้งหยาสามารถนำค่า subnet mask มาทำการทางคณิตศาสตร์กับ IP Address ก็จะสามารถหาค่า host id, subnet id, netid ออกมาได้โดยวิธีดังนี้

Netid + subnet id = [IP address] AND [Subnet Mask]

Hostid = [IP Address] AND (NOT ([Subnet Mask]))

ตัวอย่าง

IP Address = 192.168.15.20 Subnet Mask = 255.255.255.0

Netid = 192.168.15.20 AND 255.255.255.0 = 162.168.15.0

Hostid = 192.168.15.20 AND 0.0.0.255 = 0.0.0.20

หรือพูดง่ายๆ ได้ว่า "Netid ก็คือ IP Address ส่วนที่ตรงกับบิตของ Subnet mask ที่มีค่าเป็น 1 ส่วน hostid คือ IP Address ส่วนที่ตรงกับ Subnet mask ที่มีค่าเป็น 0" นั่นเอง

ดังนั้นพึงระลึกเสมอว่านอกจากการกำหนด IP Address ที่ถูกต้องแล้ว การกำหนดค่า Subnet Mask ก็มีผลต่อ IP Routing เช่นเดียวกัน การกำหนดค่า Subnet Mask ผิดพลาดย่อมจะส่งผลให้การสื่อสารข้อมูลของ IP ไม่สามารถจะกระทำได้เช่นกัน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

- ใน MySQL 3.22 หรือหลังจากนั้น คุณสามารถใช้ FIRST หรือ ADD ... AFTER col_name ในการเพิ่มคอลัมน์ ณ ตำแหน่งที่ต้องการภายในตาราง

5.8. DROP TABLE syntax

รูปแบบของคำสั่ง DROP TABLE เป็นดังนี้

```
DROP TABLE [IF EXISTS] tbl_name [, tbl_name,...]
```

DROP TABLE ใช้ในการลบตารางได้มากกว่าหนึ่ง ข้อมูลทั้งหมดในตารางและข้อกำหนดต่างๆในตารางจะถูกลบ ดังนั้นจึงควรระวังเมื่อใช้คำสั่งนี้

ใน MySQL 3.22 หรือหลังจากนั้น คุณสามารถใช้ keyword IF EXISTS เพื่อเป็นการป้องกันข้อผิดพลาดอันจะเกิดจากกรณีที่ไม่มีตารางนั้นอยู่จริง

5.9. DELETE syntax

รูปแบบคำสั่ง DELETE มีดังนี้

```
DELETE [LOW_PRIORITY] FROM tbl_name [WHERE where_definition] [LIMIT rows]
```

DELETE จะทำการลบแถวจาก tbl_name ที่กำหนดไว้ในเงื่อนไข where_condition และส่งกลับจำนวนเรคคอร์ดที่ถูกลบ ถ้าคุณใช้คำสั่ง DELETE โดยไม่มีประโยค WHERE แถวทั้งหมดจะถูกลบ

DELETE deletes rows from tbl_name that satisfy the condition given by where_definition, and returns the number of records deleted.

5.10. SELECT syntax

รูปแบบของคำสั่ง SELECT มีดังนี้

```
SELECT [STRAIGHT_JOIN] [SQL_SMALL_RESULT] [SQL_BIG_RESULT] [HIGH_PRIORITY]
      [DISTINCT | DISTINCTROW | ALL]
      select_expression,...
      [INTO {OUTFILE | DUMPFILE} 'file_name' export_options]
      [FROM table_references
        [WHERE where_definition]
        [GROUP BY col_name,...]
        [HAVING where_definition]
        [ORDER BY {unsigned_integer | col_name | formula} [ASC | DESC] ,... ]
        [LIMIT [offset,] rows]
```

```
[PROCEDURE procedure_name]
```

SELECT ใช้ในการค้นหาแถวข้อมูลจากตารางที่ต้องการ select_expression ใช้ในการกำหนดคอลัมน์ที่ต้องการจะค้นหา SELECT อาจจะใช้ในการค้นหาแถวโดยไม่ต้องอ้างถึงตารางใดเลยก็ได้ ตัวอย่าง

```
mysql> SELECT 1 + 1;  
-> 2
```

keyword ทั้งหมดที่ใช้ต้องเรียงตามลำดับตามรูปแบบที่ให้ด้านบน ตัวอย่างเช่น ประโยค HAVING ต้องตามหลังประโยค GROUP BY และต้องมาก่อนประโยค ORDER BY

- นิพจน์ SELECT สามารถใช้คำย่อ AS ได้ โดยใช้กับชื่อคอลัมน์และใช้กับประโยค ORDER BY หรือ HAVING ได้ ตัวอย่างเช่น

```
mysql> select concat(last_name,', 'first_name) AS full_name from mytable ORDER BY  
full_name;
```

```
mysql> select t1.name, t2.salary from employee AS t1, info AS t2  
where t1.name = t2.name;
```

```
mysql> select t1.name, t2.salary from employee t1, info t2  
where t1.name = t2.name;
```

- ประโยค FROM table_reference ใช้กำหนดว่าต้องการใช้ตารางใดในการค้นข้อมูล ถ้าเรียกมากกว่าหนึ่งตาราง เรียกว่าเป็นการทำ Join รายละเอียดดูวิธีการใช้ Join
- การเลือกคอลัมน์ที่ต้องการให้แสดงออกมาอาจทำได้โดยใช้ประโยค ORDER BY โดยใช้ชื่อคอลัมน์ ชื่อย่อของคอลัมน์หรือตำแหน่งคอลัมน์ ตำแหน่งคอลัมน์เริ่มจาก 1 ตัวอย่างเช่น

```
mysql> select college, region, seed from tournament  
ORDER BY region, seed;
```

```
mysql> select college, region AS r, seed AS s from tournament  
ORDER BY r, s;
```

```
mysql> select college, region, seed from tournament  
ORDER BY 2, 3;
```


ในการเรียงลำดับในทางตรงข้าม ให้เพิ่ม keyword DESC (descending) ข้างหลังชื่อคอลัมน์ในประโยค ORDER BY ถ้าไม่กำหนดการเรียงจะเป็นแบบ ASC (ascending) คือเรียงจากน้อยไปมาก

- ประโยค HAVING สามารถในการอ้างถึงคอลัมน์หรือชื่อย่อใน select_expression ตัวอย่างเช่น

```
mysql> select col_name from tbl_name HAVING col_name > 0;
```

Write this instead:

```
mysql> select col_name from tbl_name WHERE col_name > 0;
```

In MySQL 3.22.5 or later, you can also write queries like this:

```
mysql> select user,max(salary) from users  
group by user HAVING max(salary)>10;
```

- ประโยค SECKT ... INTO OUTFILE 'file_name' เป็นการเลือกแถวข้อมูลเพื่อเก็บลงไฟล์ ไฟล์นี้จะถูกสร้างที่เครื่องเซิร์ฟเวอร์ และต้องเป็นไฟล์ที่ไม่เคยมีมาก่อน

5.11. JOIN syntax

รูปแบบของ Join ที่ใช้ในประโยค SELECT มีดังนี้

```
table_reference, table_reference  
table_reference [CROSS] JOIN table_reference  
table_reference INNER JOIN table_reference  
table_reference STRAIGHT_JOIN table_reference  
table_reference LEFT [OUTER] JOIN table_reference ON conditional_expr  
table_reference LEFT [OUTER] JOIN table_reference USING (column_list)  
table_reference NATURAL LEFT [OUTER] JOIN table_reference  
{ oj table_reference LEFT OUTER JOIN table_reference ON conditional_expr }
```

รูปแบบของ LEFT OUTER JOIN ดังแสดงข้างต้นเป็นแบบที่ใช้กับ ODBC ได้เท่านั้น

- ตารางที่อ้างถึงอาจจะเป็นชื่อย่อ โดยใช้ tbl_name AS alias_name หรือ tbl_name alias_name. ก็ได้ ตัวอย่างเช่น

```
mysql> select t1.name, t2.salary from employee AS t1, info AS t2  
where t1.name = t2.name;
```

- INNER JOIN และ , (จุลภาค) มีความหมายเดียวกัน คือ เป็นการทำให้ JOIN แบบเต็มระหว่างตารางที่ใช้ปกติแล้ว จะต้องกำหนดว่าตารางจะถูกเชื่อมอย่างไรเงื่อนไขในประโยค WHERE

- ถ้าไม่มีตารางตามที่ระบุใน LEFT JOIN แถวข้อมูลที่คอลัมน์ทั้งหมดเป็น NULL จะถูกใช้เป็นตารางที่ต้องการ ซึ่งคุณสามารถใช้ในการหาเรคคอร์ดในตารางที่ไม่มีอยู่ในตารางอื่นได้ ตัวอย่างเช่น

```
mysql> select table1.* from table1
LEFT JOIN table2 ON table1.id=table2.id
where table2.id is NULL;
```

ตัวอย่างนี้ ค้นหาแถวข้อมูลทั้งหมดในตาราง table1 ที่มีค่า id ไม่มีในตาราง table2 โดยสมมติฐานที่ว่า table2 ไม่มีข้อมูล id ที่เป็น NULL

- ประโยค USING (column-list) ใช้ในการให้ชื่อคอลัมน์ที่ต้องมีอยู่ในทั้งสองตาราง เช่น

```
A LEFT JOIN B USING (C1,C2,C3,...)
```

ประโยคข้างต้นมีความหมายเดียวกับประโยคข้างล่างนี้

```
A.C1=B.C1 AND A.C2=B.C2 AND A.C3=B.C3,...
```

- NATURAL LEFT JOIN เป็นการ join สองตารางเหมือนกับ LEFT JOIN ที่ใช้ USING
- STRAIGHT_JOIN เหมือนกับการใช้ JOIN ยกเว้นแต่ว่า ตารางทางด้านซ้ายจะถูกอ่านก่อนเสมอ

ตัวอย่าง

```
mysql> select * from table1,table2 where table1.id=table2.id;
mysql> select * from table1 LEFT JOIN table2 ON table1.id=table2.id;
mysql> select * from table1 LEFT JOIN table2 USING (id);
mysql> select * from table1 LEFT JOIN table2 ON table1.id=table2.id
LEFT JOIN table3 ON table2.id=table3.id;
```

5.12. INSERT syntax

รูปแบบการใช้คำสั่ง INSERT มีดังนี้

```
INSERT [LOW_PRIORITY | DELAYED] [IGNORE]
[INTO] tbl_name [(col_name,...)]
VALUES (expression,...),(...),...
or INSERT [LOW_PRIORITY | DELAYED] [IGNORE]
[INTO] tbl_name [(col_name,...)]
SELECT ...
or INSERT [LOW_PRIORITY | DELAYED] [IGNORE]
[INTO] tbl_name
```

```
SET col_name=expression, col_name=expression, ...
```

INSERT ใช้ในการเพิ่มแถวข้อมูลลงไปในตารางที่มีอยู่แล้ว รูปแบบ INSERT... VALUES ใช้ในการเพิ่มแถวข้อมูลโดยการอิงกับค่าที่ระบุในประโยค ส่วนรูปแบบ INSERT ... SELECT ใช้ในการเพิ่มแถวข้อมูลด้วยข้อมูลที่มาจกตารางอื่น (เลือกได้มากกว่าหนึ่ง)

ในนิพจน์อาจอ้างอิงถึงคอลัมน์ใดๆที่ถูกกำหนดไว้แล้วในเรคคอร์ดข้อมูล ตัวอย่างเช่น

```
mysql> INSERT INTO tbl_name (col1,col2) VALUES(15,col1*2);
```

ถ้าคุณกำหนด keyword LOW_PRIORITY การทำตามคำสั่ง INSERT จะถูกทำภายหลังจนกว่าจะไม่มีลูกข่ายใดกำลังอ่านข้อมูลจากตาราง ในกรณีนี้ลูกข่ายจะต้องรอนกว่าประโยค INSERT จะถูกทำงานเสร็จ ซึ่งอาจจะใช้เวลานานถ้าหากตารางกำลังถูกใช้งานอย่างหนัก ซึ่งจะตรงข้ามกับประโยค INSERT DELAYED ที่ยอมให้ลูกข่ายทำงานต่อจนกระทั่งเสร็จ

ถ้า keyword IGNORE ถูกกำหนดให้ประโยค INSERT พร้อมกับแถวข้อมูลหลายแถว แถวที่ซ้ำกับคีย์หลักจะถูกเพิกเฉยและไม่ถูกเพิ่มเข้าไปในตาราง แต่ถ้าไม่กำหนด keyword IGNORE การ INSERT จะถูกยกเลิกทันทีที่พบว่ามีความซ้ำกันของคีย์หลัก

5.13. REPLACE syntax

รูปแบบการใช้ REPLACE มีดังนี้

```
REPLACE [LOW_PRIORITY | DELAYED]
  [INTO] tbl_name [(col_name,...)]
  VALUES (expression,...)
or REPLACE [LOW_PRIORITY | DELAYED]
  [INTO] tbl_name [(col_name,...)]
  SELECT ...
or REPLACE [LOW_PRIORITY | DELAYED]
  [INTO] tbl_name
  SET col_name=expression, col_name=expression,...
```

REPLACE ใช้เหมือนกับ INSERT ต่างกันเพียงแต่ถ้าที่เรคคอร์ดใดในตารางมีค่าคีย์หลักเหมือนกับเรคคอร์ดใหม่ที่ต้องการจะเพิ่ม เรคคอร์ดเก่าจะถูกลบทิ้งก่อนที่จะใส่เรคคอร์ดใหม่เข้าไป

5.14. LOAD DATA INFILE syntax

รูปแบบการใช้ LOAD DATA INFILE มีดังนี้

```
LOAD DATA [LOW_PRIORITY] [LOCAL] INFILE 'file_name.txt' [REPLACE | IGNORE]
INTO TABLE tbl_name
[FIELDS
  [TERMINATED BY '\t']
  [OPTIONALLY] ENCLOSED BY '"'
  [ESCAPED BY '\\' ]
[LINES TERMINATED BY '\n']
[IGNORE number LINES]
[(col_name,...)]
```

LOCAL หมายถึง file ที่ถูกอ่านถูกเก็บไว้ที่ client ซึ่งจะทำให้เครื่องทำงานช้าลง เนื่องจาก server จะต้องไปอ่าน file จากเครื่อง client ถ้าไม่มีคำว่า LOCAL , file จะต้องถูกเก็บไว้ที่ server และ การอ่าน text files ที่อยู่บน server file จะต้องอยู่ใน database directory หรือเป็น file ที่ user ทุกคนสามารถเปิดอ่านได้ นั่นคือถ้าคุณต้องการอ่าน file จาก server คุณจะต้องกำหนด file privilege บน server host.

ถ้า LOW_PRIORITY ถูกกำหนด หมายถึงการทำคำสั่ง LOAD DATA จะถูกเลื่อนออกไปจนกระทั่งไม่มี clients เครื่องใดอ่านข้อมูลจากตาราง

การหาค่าแห่งของ file บน server นั้น server จะต้องทราบ pathname ของ file เพื่อหาค่าแห่งของ file โดยใช้กฎดังนี้ :

- ถ้า pathname เป็นแบบ absolute path, server จะใช้ pathname ในการค้นหา file
- ถ้า pathname เป็นแบบ relative pathname, server จะค้นหา file นั้นโดยอ้างอิง จาก directory ของ data ใน server
- ถ้า file name ไม่ได้กำหนด path มาด้วย, server จะค้นหา file จาก database directory ของ database ที่ใช้ปัจจุบัน

IGNORE number LINES ใช้สำหรับทำให้ MySQL ไม่สนใจข้อมูล number บรรทัดของส่วนหัวข้อ file เช่น ข้อมูลบรรทัดที่เป็นชื่อของ column เป็นต้น

```
mysql> LOAD DATA INFILE "/tmp/file_name" into table test IGNORE 1 LINES;
```

5.15. UPDATE syntax

รูปแบบการใช้ UPDATE มีดังนี้

```
UPDATE [LOW_PRIORITY] tbl_name SET col_name1=expr1,col_name2=expr2,...  
[WHERE where_definition] [LIMIT #]
```

UPDATE ใช้ในการปรับปรุงคอลัมน์ในตารางที่มีอยู่ด้วยค่าใหม่ ประโยค SET ใช้กำหนดว่าคอลัมน์ใดจะถูกเปลี่ยนแปลงและค่าใดที่จะถูกใช้ ประโยค WHERE ใช้กำหนดว่าแถวใดที่จะถูกปรับปรุง ถ้าไม่กำหนดแล้วทุกๆ แถวจะถูกปรับปรุงหมด

ถ้า LOW_PRIORITY ถูกกำหนด การทำงานตามคำสั่ง UPDATE จะถูกเลื่อนออกไปจนกระทั่งไม่มีลูกข่ายใดกำลังอ่านข้อมูลจากตาราง

ตัวอย่าง

```
mysql> UPDATE persondata SET age=age+1;
```

คอลัมน์ age ในตาราง persondata ถูกเพิ่มค่าขึ้นหนึ่ง จากค่าเดิมของมัน

5.16. USE syntax

รูปแบบการใช้ USE มีดังนี้

```
USE db_name
```

ประโยค USE db_name ใช้ในการบอก MySQL ว่าจะให้ใช้ฐานข้อมูลชื่อ db_name เป็นฐานข้อมูลหลักในการทำแบบสอบถาม ฐานข้อมูลที่กำหนดนี้จะถูกใช้ไปจนกระทั่งสิ้นสุดการทำงาน หรือจนกว่าคำสั่ง USE จะถูกเรียกใช้อีกครั้ง

ตัวอย่าง

```
mysql> USE db1;  
mysql> SELECT count(*) FROM mytable; # selects from db1.mytable  
mysql> USE db2;  
mysql> SELECT count(*) FROM mytable; # selects from db2.mytable
```

ในการทำให้ฐานข้อมูลหนึ่งเป็นฐานข้อมูลหลักในความหมายของคำสั่ง USE ไม่ได้หมายความว่าท่านไม่สามารถใช้ฐานข้อมูลอื่นได้ เพียงแต่เป็นการกำหนดฐานข้อมูลหลักในการทำงานเท่านั้น ตัวอย่างข้างล่างจะแสดงให้เห็นว่าสามารถเข้าถึงตารางจากฐานข้อมูลคนละตัวได้

```
mysql> USE db1;
mysql> SELECT author_name,editor_name FROM author,db2.editor
WHERE author.editor_id = db2.editor.editor_id;
```

5.17. SHOW syntax

รูปแบบการใช้ SHOW มีดังนี้

```
SHOW DATABASES [LIKE wild]
or SHOW TABLES [FROM db_name] [LIKE wild]
or SHOW COLUMNS FROM tbl_name [FROM db_name] [LIKE wild]
or SHOW TABLE STATUS
or SHOW STATUS
```

wild หมายถึง wildcard characters คือ '%' และ '_'

SHOW ใช้ดูข้อมูลเกี่ยวกับ database, table, columns และ server สำหรับการอ้างอิงชื่อตารางนั้น คุณสามารถอ้างอิงโดยใช้ db_name.tbl_name แทน tbl_name FROM db_name syntax. ดังตัวอย่าง 2 บรรทัดนี้ ที่มีความหมายเหมือนกัน

```
mysql> SHOW COLUMNS FROM mytable FROM mydb;
mysql> SHOW COLUMNS FROM mydb.mytable;
```

- SHOW DATABASES แสดงรายชื่อของ database ทั้งหมดที่มีใน MySQL server host
- SHOW TABLES แสดงรายชื่อของ table ทั้งหมดที่มีใน database ที่กำหนด โดยคุณสามารถใช้คำสั่ง mysqlshow db_name เพื่อแสดงรายชื่อของ table ทั้งหมดใน db_name นั้นๆ ได้เช่นกัน
- SHOW COLUMNS แสดงรายชื่อของ column ทั้งหมดใน table ที่กำหนด
- SHOW TABLE STATUS แสดงข้อมูลของแต่ละ table ซึ่งคุณสามารถใช้คำสั่ง mysqlshow -status db_name ได้เช่นกัน
- SHOW STATUS แสดงข้อมูลเกี่ยวกับสถานะการทำงานของ server

5.18. DESCRIBE syntax

รูปแบบการใช้ DESCRIBE มีดังนี้

```
{DESCRIBE | DESC} tbl_name {col_name | wild}
```

DESCRIBE ใช้แสดงข้อมูลเกี่ยวกับ columns ของ table tbl_name โดยที่ col_name อาจเป็นตัวอักษร หรือ '%' และ '_' (wildcard character)

5.19. GRANT และ REVOKE syntax

```
GRANT priv_type [(column_list)] [, priv_type [(column_list)] ...]
  ON {tbl_name | * | *.* | db_name.*}
  TO user_name [IDENTIFIED BY 'password']
  [, user_name [IDENTIFIED BY 'password'] ...]
  [WITH GRANT OPTION]

REVOKE priv_type [(column_list)] [, priv_type [(column_list)] ...]
  ON {tbl_name | * | *.* | db_name.*}
  FROM user_name [, user_name ...]
```

คำสั่ง GRANT and REVOKE อนุญาตให้ admin ให้สิทธิ และ เพิกถอนสิทธิให้กับ MySQL users ซึ่ง ระดับของสิทธิในการใช้งาน MySQL มีทั้งหมด 4 ระดับ ดังนี้ :

- (1) **Global level** เป็นสิทธิที่ใช้กับทุก database บน server ซึ่งข้อมูลเกี่ยวกับสิทธิระดับทั่วไปเก็บไว้ใน mysql.user table
- (2) **Database level** เป็นสิทธิเกี่ยวกับทุกตารางใน database ที่กำหนดให้, ข้อมูลเกี่ยวกับสิทธิระดับ database จะถูกเก็บไว้ใน mysql.db และ mysql.host table
- (3) **Table level** เป็นสิทธิเกี่ยวกับทุกๆ column ใน table ที่กำหนดให้ ข้อมูลเกี่ยวกับสิทธิระดับ table จะถูกเก็บไว้ใน mysql.tables_priv table
- (4) **Column level** เป็นสิทธิเกี่ยวกับ column แต่ละ column ใน table ที่กำหนดให้ ซึ่งจะถูเก็บไว้ใน mysql.columns table

ชนิดของสิทธิที่สามารถใช้ได้คำสั่ง GRANT และ REVOKE คือ :

ALL PRIVILEGES	FILE	RELOAD
ALTER	INDEX	SELECT
CREATE	INSERT	SHUTDOWN
DELETE	PROCESS	UPDATE
DROP	REFERENCES	USAGE

ALL : มีสิทธิครบทั้งหมด

USAGE : ไม่มีสิทธิใดๆ เลย

ชนิดของสิทธิที่ใช้สำหรับ table คือ SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, GRANT, INDEX and ALTER.

ชนิดของสิทธิที่ใช้สำหรับ column คือ SELECT, INSERT and UPDATE.

คุณสามารถกำหนด global privileges โดยใช้ ON *.* syntax และสามารถกำหนด database privileges โดยใช้ ON db_name.* syntax.

เพื่อให้การกำหนดสิทธิ์ของ user สะดวกขึ้น, MySQL สนับสนุนการกำหนดค่าของ user_name ในรูปแบบของ user@host ซึ่งถ้าคุณต้องการระบุ user string ที่มีตัวอักษรพิเศษเช่น '-' หรือ host string ที่มีตัวอักษรพิเศษ หรือ wildcard character(เช่น '%') คุณสามารถใช้เครื่องหมาย quote ที่ชื่อ user และ host ได้ เช่น 'test-user'@'test-hostname' เป็นต้น

คุณสามารถใช้เครื่องหมาย wildcards ใน host name ได้ เช่น user@"%.cmu.ac.th" หมายถึงผู้ใช้ชื่อ user สำหรับทุกๆ host ใน cmu.ac.th domain และ user@"203.146.57.%" หมายถึงผู้ใช้ชื่อ user สำหรับทุก host ใน class C subnet 144.155.166

5.20. CREATE INDEX syntax

รูปแบบการใช้คำสั่ง CREATE INDEX มีดังนี้

```
CREATE [UNIQUE] INDEX index_name ON tbl_name (col_name[(length)],... )
```

คำสั่งนี้นิยมใช้ร่วมกับคำสั่ง ALTER TABLE ซึ่งโดยทั่วไปแล้ว คุณสามารถสร้าง index ให้กับตารางข้อมูลได้ในตอนแรกที่ตารางถูกสร้างโดยการใช้คำสั่ง CREATE TABLE. คำสั่ง CREATE INDEX อนุญาตให้คุณเพิ่ม index ให้กับตาราง

สำหรับ column ที่มีชนิดข้อมูลเป็น CHAR และ VARCHAR , เราสามารถสร้าง index ให้กับข้อมูลทั้ง 2 ชนิดโดยใช้เพียงส่วนหนึ่งของ columns ได้ เช่น หากต้องการให้สร้าง index โดยใช้ตัวอักษร 10 ตัวแรกของ column ชื่อ name ให้ใช้คำสั่งดังนี้ :

```
mysql> CREATE INDEX part_of_name ON customer (name(10));
```

5.21. DROP INDEX syntax

รูปแบบการใช้คำสั่ง DROP INDEX มีดังนี้

```
DROP INDEX index_name ON tbl_name
```

DROP INDEX drops the index named index_name from the table tbl_name.

คำสั่งนี้ใช้ร่วมกับ ALTER TABLE เพื่อลบ index ออกจาก column ที่ต้องการ

5.22. COMMENT syntax

MySQL อนุญาตให้ใส่สัญลักษณ์

หรือ - เพื่อใช้เริ่มต้นเขียน comment จนถึงสิ้นสุดบรรทัด

/* ใช้เริ่มต้นเขียน comment จนถึงสิ้นสุดที่สัญลักษณ์ */

เช่น :

```
mysql> select 1+1; # This comment continues to the end of line
mysql> select 1+1; - This comment continues to the end of line
mysql> select 1 /* this is an in-line comment */ + 1;
mysql> select 1+
/*
this is a
multiple-line comment
*/
1;
```

5.23. MySQL reserved words

MySQL มีคำที่สงวนไว้ ซึ่งคุณไม่สามารถนำคำเหล่านี้ไปตั้งเป็นชื่อตารางหรือ column ได้ ซึ่งคำดังกล่าว ได้แก่ คำที่เป็นชนิดของข้อมูลใน MySQL , คำที่เป็นชื่อ function ที่มีให้ใช้ใน MySQL (เช่น TIMESTAMP, GROUP) เป็นต้น คุณสามารถใช้ชื่อ function ตั้งชื่อตารางหรือ column ได้เหมือนกัน แต่เวลาเรียกใช้ function นั้นๆ ต้องไม่ให้มีช่องว่างระหว่างชื่อ function กับ เครื่องหมายวงเล็บเปิดเด็ดขาด

คำต่างๆ ในตารางด้านล่าง เป็นคำที่สงวนสำหรับ MySQL ซึ่งส่วนใหญ่ก็ไม่สามารถใช้ตั้งชื่อตารางหรือ column ได้ตามมาตรฐานของ ANSI SQL92 ด้วยเช่นกัน

action	add	aggregate	all
alter	after	and	as
asc	avg	avg_row_length	auto_increment
between	bigint	bit	binary
blob	bool	both	by
cascade	case	char	character
change	check	checksum	column
columns	comment	constraint	create
cross	current_date	current_time	current_timestamp
data	database	databases	date
datetime	day	day_hour	day_minute
day_second	dayofmonth	dayofweek	dayofyear

dec	decimal	default	delayed
delay_key_write	delete	desc	describe
distinct	distinctrow	double	drop
end	else	escape	escaped
enclosed	enum	explain	exists
fields	file	first	float
float4	float8	flush	foreign
from	for	full	function
global	grant	grants	group
having	heap	high_priority	hour
hour_minute	hour_second	hosts	identified
ignore	in	index	infile
inner	insert	insert_id	int
integer	interval	int1	int2
int3	int4	int8	into
if	is	isam	join
key	keys	kill	last_insert_id
leading	left	length	like
lines	limit	load	local
lock	logs	long	longblob
longtext	low_priority	max	max_rows
match	mediumblob	mediumtext	mediumint
middleint	min_rows	minute	minute_second
modify	month	monthname	myisam
natural	numeric	no	not
null	on	optimize	option
optionally	or	order	outer
outfile	pack_keys	partial	password
precision	primary	procedure	process
processlist	privileges	read	real
references	reload	regexp	rename
replace	restrict	returns	revoke
rlike	row	rows	second
select	set	show	shutdown
smallint	soname	sql_big_tables	sql_big_selects
sql_low_priority_updates	sql_log_off	sql_log_update	sql_select_limit
sql_small_result	sql_big_result	sql_warnings	straight_join
starting	status	string	table

tables	temporary	terminated	text
then	time	timestamp	tinyblob
tinytext	tinyint	trailing	to
type	use	using	unique
unlock	unsigned	update	usage
values	varchar	variables	varying
varbinary	with	write	when
where	year	year_month	zerofill

สัญลักษณ์จากตารางด้านบนบางตัวไม่สามารถใช้ตั้งเป็นชื่อตารางหรือคอลัมน์ได้ ตามมาตรฐานของ ANSI SQL แต่สำหรับ MySQL สามารถใช้ได้ สัญลักษณ์เหล่านั้นได้แก่

- ACTION
- BIT
- DATE
- ENUM
- NO
- TEXT
- TIME
- TIMESTAMP



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

- User: ใส่ชื่อ user เพื่อ login MySQL server
- Password: ใส่รหัสผ่าน
- Port: port ที่ต้องการติดต่อด้วย ไม่จำเป็นต้องใส่ (default = 3306)

(5) กดปุ่ม OK จากนั้นจะเห็น Data Sources Windows อีกครั้ง ให้กดปุ่ม close

TDX mysql Driver default configuration

This is in public domain and comes with NO WARRANTY of any kind
Enter a database and options for connect

Windows DSN name: cpe petshop

MySQL host (name or IP): 10.10.60.14

MySQL database name: petshop

User: seller

Password: [masked]

Port (if not 3306): 3306

SQL command on connect:

Options that affects the behaviour of MyODBC

<input type="checkbox"/> Don't optimize column width	<input type="checkbox"/> Pad CHAR to full length
<input type="checkbox"/> Return matching rows	<input type="checkbox"/> Return table names in SQLDescribeCol
<input type="checkbox"/> Trace MyODBC	<input type="checkbox"/> Use compressed protocol
<input type="checkbox"/> Allow BIG results	<input type="checkbox"/> Ignore space after function names
<input type="checkbox"/> Don't prompt on connect	<input type="checkbox"/> Force use of named pipes
<input type="checkbox"/> Simulate ODBC 1.0	<input type="checkbox"/> Change BIGINT columns to INT
<input type="checkbox"/> Ignore # in # table	<input type="checkbox"/> No catalog [exp]
<input type="checkbox"/> Use manager cursors [exp]	<input type="checkbox"/> Read options from C:\my.cnf
<input type="checkbox"/> No extended fetch [exp]	<input type="checkbox"/> Safety (Check this if you have problems)

OK Cancel

รูปที่ (3) วิธีการใส่ข้อมูลลงไปใน Configuration

6.2 การสร้าง DSN ของ ODBC ได้หลาย DSN เพื่อใช้ใน application ต่างกัน

- (1) เปิด Control Panel
- (2) Double click ที่ไอคอน ODBC Data Sources 32 bits
- (3) Click ที่ tab User DSN
- (4) Click ที่ปุ่ม Add
- (5) เลือก MySQL ในหน้าจอ Create New Data Source จากนั้น click ปุ่ม finish
- (6) คุณจะเห็นหน้าต่าง TCX MySQL Driver default configuration ดังรูปที่ (3) ซึ่งคุณสามารถ config ค่าต่างๆ ได้

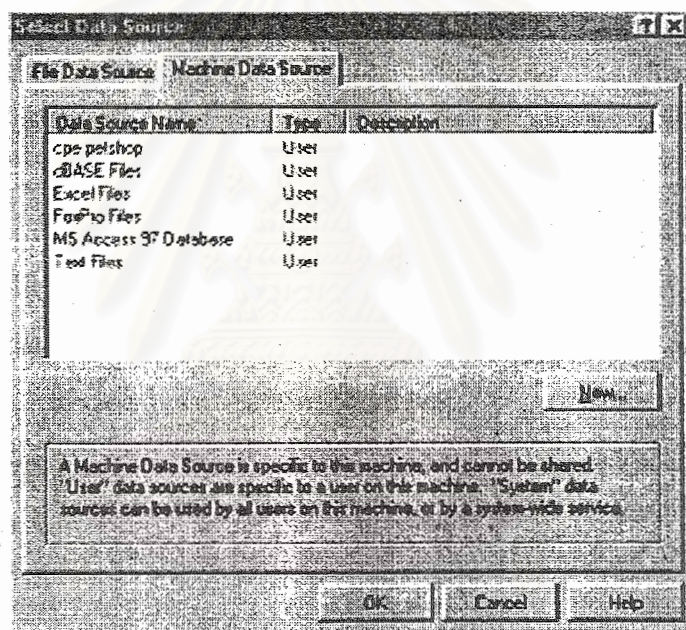
6.3 การใช้ Microsoft Access เชื่อมต่อกับ MySQL

วิธีการเชื่อมโยงข้อมูลกันระหว่าง Access กับ MySQL จะใช้ ODBC เป็นตัวเชื่อม ซึ่งแบ่งได้เป็น 2 กรณีคือ

6.3.1 การ Link table

คุณสามารถดึงข้อมูลจาก MySQL มาดูใน access ได้ ข้อมูลที่อยู่ใน MySQL จะแสดงให้คุณดูในรูปแบบของข้อมูลใน Access ด้วยวิธีการดังนี้

- (1) สร้าง DSN ที่สำหรับ database ของ MySQL ที่คุณต้องการติดต่อด้วย
- (2) ใช้ Microsoft office เปิด database ที่ต้องการจะ link table มาจาก MySQL
- (3) click ที่ file -> get external data -> Link Table...
- (4) เลือก File of type เป็น ODBC database
- (5) เลือก tab Machine Data Source
- (6) คุณจะเห็น ODBC ที่มีให้เลือก ดังรูปที่ (4) ให้เลือก DSN ที่คุณต้องการ
- (7) เลือก table ที่ต้องการ Link เข้ามาใน Access



รูปที่ (4) Select Data Source Window

6.3.2. import/ export table

การ import/export นี้ เปรียบเสมือนการคัดลอกข้อมูลระหว่าง Microsoft Access กับ MySQL ซึ่งวิธีการเรียกใช้ ODBC ก็เหมือนกับวิธีการ Link table

- Import -> เป็นการดึงข้อมูลจาก MySQL มาเก็บไว้ใน Microsoft Access
- Export -> เป็นการส่งข้อมูลจาก Microsoft Access ไปเก็บไว้ใน MySQL

7. บทที่ 7 วิธีใช้งาน MySQL เบื้องต้น

7.1 การ connect และ disconnect MySQL server

การ connect ไปยัง server จะต้องใช้ MySQL user name และ password ถ้า server run อยู่บนเครื่องอื่นคนต้องบอก host name ของเครื่อง server ด้วย

```
shell> mysql -h host -u user -p
Enter password: *****
```

เครื่องหมาย ***** จะปรากฏแทนที่ password ของคุณ เมื่อคุณป้อน password หากคุณสามารถ login เข้ามาได้ คุณจะเห็น mysql> prompt ซึ่ง หมายถึง MySQL พร้อมทั้งจะรับคำสั่งจากคุณ

```
shell> mysql -h host -u user -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 459 to server version: 3.22.20a-log

Type 'help' for help.

mysql>
```

หลังจากคุณสามารถ connect ไปยัง MySQL server ได้แล้ว คุณสามารถ disconnect ได้โดยพิมพ์คำว่า quit หรือกดปุ่ม Control-D

```
mysql> QUIT
Bye
```

7.2 การพิมพ์คำสั่งเพื่อดึงข้อมูลจาก database (Entering Queries)

ในขั้นต้น ให้พิมพ์คำสั่งเพื่อ ถาม version ของ MySQL server และ วันที่ปัจจุบันของเครื่อง ตามตัวอย่างข้างล่างแล้วกดปุ่ม Enter

```
mysql> SELECT VERSION(), CURRENT_DATE;
+-----+-----+
| version() | CURRENT_DATE |
+-----+-----+
| 3.22.20a-log | 1999-03-19 |
+-----+-----+
1 row in set (0.01 sec)
mysql>
```

การ query ข้อมูลข้างต้นแสดงให้เห็นวิธีการพิมพ์คำสั่งให้กับ MySQL ดังนี้ :

- ทุกๆ คำสั่งที่ประกอบด้วย SQL statement จะต้องปิดท้ายด้วยเครื่องหมาย semicolon (;)
- เมื่อคุณพิมพ์คำสั่งแล้วกด Enter, MySQL จะส่งคำสั่งไปยัง server เพื่อประมวลผลและจากนั้นก็แสดงผลพร้อมกลับมาให้ที่หน้าจอ พร้อมทั้งแสดง mysql> อีกครั้งเพื่อรอรับคำสั่งครั้งใหม่
- mysql แสดงผลลัพธ์ของการ query ในรูปแบบของตาราง แถวแรกของตารางคือชื่อของ column หรือ expression ที่คุณป้อนเข้าไป ส่วนแถวถัดไปจะเป็นผลลัพธ์ที่ได้จากการ query

- mysql จะแสดงจำนวนแถว (records) ที่ได้จากการ query และ ใช้เวลาในการ query นานเท่าใด ซึ่งจะทำให้เราทราบถึงประสิทธิภาพของ server ได้อย่างคร่าวๆ

Keywords ที่ใช้อาจพิมพ์โดยใช้ตัวเล็กหรือตัวใหญ่ก็ได้ เช่น :

```
mysql> SELECT VERSION(), CURRENT_DATE;
mysql> select version(), current_date;
mysql> SeLeCt vErSiOn(), current_DATE;
```

คุณสามารถใช้ mysql ในการคำนวณก็ได้ เช่น :

```
mysql> SELECT SIN(PI()/4), (4+1)*5;
+-----+-----+
| SIN(PI()/4) | (4+1)*5 |
+-----+-----+
| 0.707107 | 25 |
+-----+-----+
```

คำสั่งที่ป้อนเข้าไปใน mysql> prompt ไม่จำเป็นต้องอยู่บรรทัดเดียวกันทั้งหมด เนื่องจาก mysql จะพิจารณาว่า ทุกคำสั่งต้องสิ้นสุดด้วยเครื่องหมาย semicolon(;) เช่น

```
mysql> SELECT
-> USER()
-> ,
-> CURRENT_DATE;
+-----+-----+
| USER() | CURRENT_DATE |
+-----+-----+
| joesmith@localhost | 1999-03-18 |
+-----+-----+
```

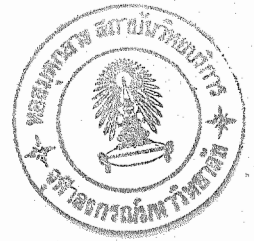
จากตัวอย่างข้างต้น จะเห็นว่า prompt ของ mysql เปลี่ยนจาก mysql> เป็น -> เมื่อคุณกดปุ่ม enter โดยที่ยังไม่ใส่เครื่องหมาย semicolon หากคุณต้องการยกเลิกคำสั่งที่คุณพิมพ์ไปแล้ว คุณสามารถยกเลิกได้โดยการพิมพ์ \c

```
mysql> SELECT
-> USER()
-> \c
mysql>
```

เมื่อคุณยกเลิกคำสั่งแล้ว prompt -> ก็จะกลับมาเป็น mysql> เพื่อรอรับคำสั่งใหม่ต่อไป

Prompt	ความหมาย
mysql>	พร้อมรับคำสั่งต่อไป
->	รอรับคำสั่งบรรทัดต่อไป
>	รอรับคำสั่งบรรทัดต่อไป ซึ่งจะต้องมีเครื่องหมาย single quote ในคำสั่งด้วย
>>	รอรับคำสั่งบรรทัดต่อไป ซึ่งจะต้องมีเครื่องหมาย double quote ในคำสั่งด้วย

ตารางแสดงรูปแบบของ prompt ที่คุณสามารถพบได้ใน MySQL



7.3 ตัวอย่างของการ query

7.3.1 การสร้างตาราง

ให้ทดลองสร้างตารางชื่อ shop เพื่อเก็บราคาของหนังสือ สำหรับแต่ละตัวแทนจำหน่าย

```
CREATE TABLE shop (  
  article INT(4) UNSIGNED ZEROFILL DEFAULT '0000' NOT NULL,  
  dealer CHAR(20) DEFAULT '' NOT NULL,  
  price DOUBLE(16,2) DEFAULT '0.00' NOT NULL,  
  PRIMARY KEY(article, dealer));  
  
INSERT INTO shop VALUES  
(1, 'A', 3.45), (1, 'B', 3.99), (2, 'A', 10.99), (3, 'B', 1.45), (3, 'C', 1.69),  
(3, 'D', 1.25), (4, 'D', 19.95);  
Okay, so the example data is:  
SELECT * FROM shop
```

article	dealer	price
0001	A	3.45
0001	B	3.99
0002	A	10.99
0003	B	1.45
0003	C	1.69
0003	D	1.25
0004	D	19.95

7.3.2 การสร้าง และ ใช้งาน database

(1) การสร้าง database

ถ้า administrator สร้าง database ให้คุณในตอนที set up permission, คุณสามารถใช้มันได้เลย หากยัง คุณต้องสร้าง database ขึ้นมาเอง ดังนี้:

```
mysql> CREATE DATABASE student;
```

สำหรับการใช้งานบน Linux, ชื่อของ database เป็นแบบ case sensitive นั่นคือ database student ไม่เหมือนกับ Student และ ไม่เหมือนกับ STUDENT ซึ่งการตั้งชื่อ table ก็ใช้กฎเดียวกันด้วย การสร้าง database จะสร้างขึ้นมามีครั้งแรกเพียงครั้งเดียวเท่านั้น แต่การใช้งานแต่คุณต้องเลือก database ก่อนที่จะใช้งานทุกครั้ง

(2) การใช้งาน database

```
mysql> USE student  
Database changed
```

อีกวิธีคือ ระบุ database ที่ต้องการใช้งาน ตอน login ใช้ MySQL Server เลย

```
shell> mysql -h host -u user -p database_name  
Enter password: *****
```

7.4 การ Load ข้อมูลจากไฟล์ ลงไปใน ตาราง

สมมติว่าเราสร้าง database ชื่อ petshop และสร้าง table ชื่อ pet โดยใช้ syntax ดังนี้

```
mysql> CREATE DATABASE petshop;
mysql> CREATE TABLE pet (name VARCHAR(20), owner VARCHAR(20),
-> species VARCHAR(20), sex CHAR(1), birth DATE, death DATE);
```

หลังจากมี database และ สร้าง table ได้แล้ว เราจำเป็นต้องเรียนคำสั่ง LOAD DATA และ INSERT เพื่อนำข้อมูลเข้าไปเก็บใน table สมมติว่าคุณสร้าง table pet โดยมีรายละเอียดของตารางและข้อมูล ดังนี้

name	owner	species	sex	birth	death
Fluffy	Harold	cat	f	1993-02-04	
Claws	Gwen	cat	m	1994-03-17	
Buffy	Harold	dog	f	1989-05-13	
Fang	Benny	dog	m	1990-08-27	
Bowser	Diane	dog	m	1998-08-31	1995-07-29
Chirpy	Gwen	bird	f	1998-09-11	
Whistler	Gwen	bird		1997-12-09	
Slim	Benny	snake	m	1996-04-29	

ถ้าคุณมีแค่ table ว่างๆ วิธีการง่ายๆ ที่จะ load ข้อมูลเข้า table คือ การสร้าง text file ซึ่งประกอบด้วยแถวสำหรับสัตว์แต่ละชนิด จากนั้นทำการ load ข้อมูลใน file เข้าไปไว้ใน table เช่น สร้าง file ชื่อ pet.txt ที่มี 1 record ต่อ 1 บรรทัด ซึ่งค่าแต่ละค่าแยกจากการด้วยเครื่องหมาย tab เรียงตามลำดับใน create table statement สำหรับ column ใดๆ ที่ไม่มีข้อมูลอยู่ ให้ใส่เครื่องหมาย \N เช่น

```
Whistler          Gwen          bird          \N          1997-12-09          \N
```

การ load text file ชื่อ pet.txt ไปเก็บใน pet table ใช้คำสั่ง

```
mysql> LOAD DATA LOCAL INFILE "pet.txt" INTO TABLE pet;
```

ถ้าคุณต้องการเพิ่ม record เข้าไปด้วยการ INSERT ให้ใช้คำสั่ง

```
mysql> INSERT INTO pet
-> VALUES ('Puffball', 'Diane', 'hamster', 'f', '1999-03-30', NULL);
```

โดยที่ค่าที่จะ insert เข้าไป จะต้องเรียงตาม column เหมือนกัน CREATE TABLE statement ค่าใดไม่มี ให้ใช้ NULL เดิมแทน

7.5 การดึงข้อมูลออกมาจากราย

คำสั่ง SELECT ถูกใช้ในการดึงข้อมูลออกมาจากราย รูปแบบทั่วไปของคำสั่ง select เขียนได้ดังนี้

```
SELECT what_to_select
FROM which_table
WHERE conditions_to_satisfy
```

what_to_select บอกว่า select ข้อมูลอะไร อาจเป็นชื่อ column หรือ * ซึ่งหมายถึงทุก
column ก็ได้ which_table บอกว่า จะ select ข้อมูลจากรายไหน
The WHERE clause เป็นตัวเลือก ไม่จำเป็นต้องพิมพ์
conditions_to_satisfy เป็นตัวกำหนดเงื่อนไขของข้อมูลที่จะดึงจากราย

7.5.1 เลือกข้อมูลทั้งตาราง

วิธีการ select ข้อมูลทั้งหมดจากราย

```
mysql> SELECT * FROM pet;
```

name	owner	species	sex	birth	death
Fluffy	Harold	cat	f	1993-02-04	NULL
Claws	Gwen	cat	m	1994-03-17	NULL
Buffy	Harold	dog	f	1989-05-13	NULL
Fang	Benny	dog	m	1990-08-27	NULL
Bowser	Diane	dog	m	1989-08-31	1995-07-29
Chirpy	Gwen	bird	f	1998-09-11	NULL
Whistler	Gwen	bird	NULL	1997-12-09	NULL
Slim	Benny	snake	m	1996-04-29	NULL
Puffball	Diane	hamster	f	1999-03-30	NULL

7.5.2 เลือกข้อมูลเฉพาะบางแถว (บาง records)

มีการใส่เงื่อนไขลงใน WHERE clause เช่น เลือกข้อมูลของสัตว์เลี้ยงที่ชื่อ "Bowser" เท่านั้น

```
mysql> SELECT * FROM pet WHERE name = "Bowser";
```

name	owner	species	sex	birth	death
Bowser	Diane	dog	m	1989-08-31	1995-07-29

เลือกข้อมูลของสัตว์เลี้ยงที่เกิดหลังจากปี 1999

```
mysql> SELECT * FROM pet WHERE birth >= "1998-1-1";
```

name	owner	species	sex	birth	death
Chirpy	Gwen	bird	f	1998-09-11	NULL
Puffball	Diane	hamster	f	1999-03-30	NULL

คุณสามารถใส่เงื่อนไขหลายๆ เงื่อนไขรวมกันได้ โดยใช้คำเชื่อมเช่น AND , OR

```
mysql> SELECT * FROM pet WHERE species = "dog" AND sex = "f";
```

name	owner	species	sex	birth	death
Buffy	Harold	dog	f	1989-05-13	NULL

```
mysql> SELECT * FROM pet WHERE species = "snake" OR species = "bird";
```

name	owner	species	sex	birth	death
Chirpy	Gwen	bird	f	1998-09-11	NULL
Whistler	Gwen	bird	NULL	1997-12-09	NULL
Slim	Benny	snake	m	1996-04-29	NULL

```
mysql> SELECT * FROM pet WHERE (species = "cat" AND sex = "m")  
-> OR (species = "dog" AND sex = "f");
```

name	owner	species	sex	birth	death
Claws	Gwen	cat	m	1994-03-17	NULL
Buffy	Harold	dog	f	1989-05-13	NULL

7.5.3 เลือกข้อมูลเฉพาะบางคอลัมน์

คุณสามารถเลือกข้อมูลออกมาเฉพาะ columns ได้ โดยการระบุชื่อ columns ใน SELECT statement

```
mysql> SELECT name, birth FROM pet;
```

name	birth
Fluffy	1993-02-04
Claws	1994-03-17
Buffy	1989-05-13
Fang	1990-08-27
Bowser	1989-08-31
Chirpy	1998-09-11
Whistler	1997-12-09
Slim	1996-04-29
Puffball	1999-03-30

ต้องการหาชื่อเจ้าของสัตว์เลี้ยง

```
mysql> SELECT owner FROM pet;
```

```
+-----+
| owner |
+-----+
| Harold |
| Gwen  |
| Harold |
| Benny  |
| Diane  |
| Gwen   |
| Gwen   |
| Benny  |
| Diane  |
+-----+
```

ต้องการหาชื่อเจ้าของสัตว์เลี้ยงโดยไม่ต้องการใช้แสดงชื่อซ้ำกันใน record ให้ใช้คำสั่ง DISTINCT:

```
mysql> SELECT DISTINCT owner FROM pet;
```

```
+-----+
| owner |
+-----+
| Benny  |
| Diane  |
| Gwen   |
| Harold |
+-----+
```

7.6 การใช้ Pattern matching

SQL pattern matching ใช้ '_' แทนที่ตัวอักษรใดๆ 1 ตัว และ '*' แทนที่ตัวอักษรใดๆ ก็ได้ สำหรับ MySQL, SQL patterns เป็นการ match แบบ case insensitive เมื่อคุณใช้ SQL patterns ให้ใช้ คำว่า LIKE หรือ NOT LIKE ในการเปรียบเทียบ ตัวอย่างเช่น

หาชื่อที่ขึ้นต้นด้วย 'b':

```
mysql> SELECT * FROM pet WHERE name LIKE "b%";
```

```
+-----+-----+-----+-----+-----+-----+
| name | owner | species | sex | birth | death |
+-----+-----+-----+-----+-----+-----+
| Buffy | Harold | dog      | f   | 1989-05-13 | NULL |
| Bowser | Diane | dog      | m   | 1989-08-31 | 1995-07-29 |
+-----+-----+-----+-----+-----+-----+
```


หาชื่อที่ลงท้ายด้วย 'fy' :

```
mysql> SELECT * FROM pet WHERE name LIKE "%fy";
```

name	owner	species	sex	birth	death
Fluffy	Harold	cat	f	1993-02-04	NULL
Buffy	Harold	dog	f	1989-05-13	NULL

หาชื่อที่มี 'w' อยู่ตำแหน่งใดก็ได้ ในชื่อ :

```
mysql> SELECT * FROM pet WHERE name LIKE "%w%";
```

name	owner	species	sex	birth	death
Claws	Gwen	cat	m	1994-03-17	NULL
Bowser	Diane	dog	m	1989-08-31	1995-07-29
Whistler	Gwen	bird	NULL	1997-12-09	NULL

หาชื่อที่ประกอบด้วยตัวอักษรอะไรก็ได้ จำนวน 5 ตัวอักษร ให้ใช้สัญลักษณ์ '_' :

```
mysql> SELECT * FROM pet WHERE name LIKE "_____";
```

name	owner	species	sex	birth	death
Claws	Gwen	cat	m	1994-03-17	NULL
Buffy	Harold	dog	f	1989-05-13	NULL

7.7 การนับจำนวนแถวของข้อมูลที่ถูก select

ใช้ function COUNT() ในการนับจำนวนแถวที่ query ได้ โดยใช้ร่วมกับ SELECT statement:

```
mysql> SELECT COUNT(*) FROM pet;
```

COUNT(*)
9

ถ้าต้องการ query ชื่อเจ้าของ และ จำนวนสัตว์เลี้ยงที่แต่ละคนมีอยู่:

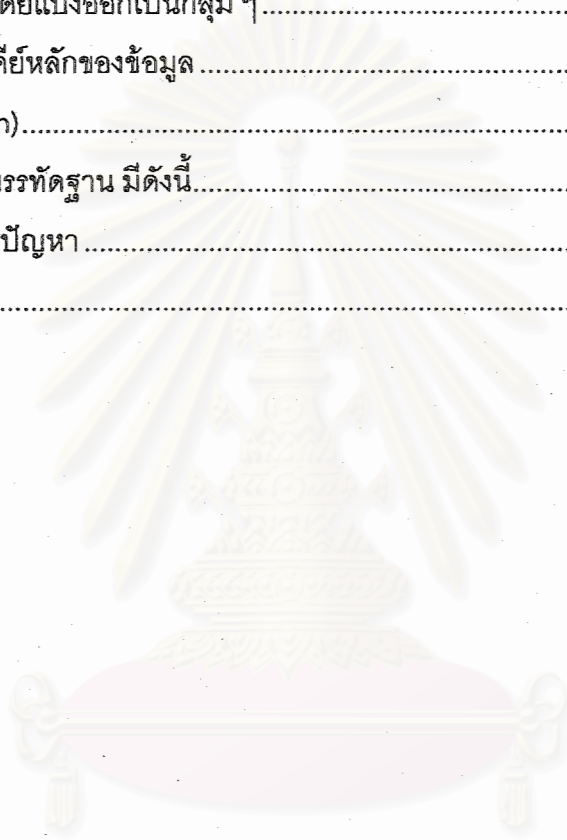
```
mysql> SELECT owner, COUNT(*) FROM pet GROUP BY owner;
```

owner	COUNT(*)
Benny	2
Diane	2
Gwen	3
Harold	2

สารบัญ

Generic Network Configuration.....	1
ศัพท์เบื้องต้นเกี่ยวกับระบบ Network.....	1
IP Address.....	2
Network Class.....	3
IP Routing.....	5
อุปกรณ์ที่ใช้ในเน็ตเวิร์ค.....	5
หลักการพื้นฐานของ IP Routing.....	6
Subnet Addressing.....	9
Subnet Mask.....	11
Client/ Server Model.....	13
องค์ประกอบของ Client/ Server.....	13
เทคโนโลยี Client/ Server ในปัจจุบัน.....	13
ข้อดีของการพัฒนาระบบงานแบบ Client/ Server.....	14
จุดอ่อน (ข้อด้อย) ของระบบงานแบบ Client/ Server.....	14
Server.....	15
ความหมาย.....	15
ประเภทของ Server.....	15
ขนาดของ Server.....	15
มาตรฐานประสิทธิภาพ.....	16
การแบ่งประเภทของ Server.....	16
คุณสมบัติของ Server.....	18
ระบบการประมวลผลแบบ On-Line.....	21
(On-Line Transaction Processing System).....	21
ความหมาย.....	21
รูปแบบการประมวลผลงานแบบ On-Line.....	21
On-Line Interactive System.....	21
Database Concept.....	24
ระบบฐานข้อมูล.....	24
การจัดเก็บข้อมูลในรูปแบบแฟ้มข้อมูล.....	24
การจัดเก็บข้อมูลในรูปแบบฐานข้อมูล.....	24
ข้อเสียของการจัดเก็บข้อมูลในรูปแบบฐานข้อมูล.....	25

ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System)	25
คำศัพท์ที่เกี่ยวข้อง	25
ประเภทของคีย์.....	26
กฎที่ใช้กับฐานข้อมูลเชิงสัมพันธ์.....	27
ภาษาทางด้านฐานข้อมูล (Query Language)	28
ขั้นตอนการทำการออกแบบฐานข้อมูล	30
1. ทำความเข้าใจระบบงานให้ชัดเจน.....	30
2. ร่างความสัมพันธ์คร่าว ๆ โดยแบ่งออกเป็นกลุ่ม ๆ.....	30
3. กำหนดความสัมพันธ์และคีย์หลักของข้อมูล	31
รูปแบบบรรทัดฐาน (Normalization).....	32
วัตถุประสงค์ของการทำให้เป็นบรรทัดฐาน มีดังนี้.....	32
ข้อมูลที่ซ้ำซ้อนกันนี้จะก่อให้เกิดปัญหา	32
หลักการทำให้ Normalization	32



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

รูปที่ 1 Network Class.....	3
รูปที่ 3 การสื่อสารในเน็ตเวิร์กที่ต่อร่วมกัน (Shared Network)	7
รูปที่ 4 การสื่อสารระหว่าง 2 เน็ตเวิร์ก.....	7
รูปที่ 6 แสดงการจัดแบ่ง IP Address ของคลาส ออกเป็นเน็ตเวิร์กย่อยด้วยวิธี Subnetting โดยแบ่งพื้นที่ส่วนที่เป็นของ hostid เดิมออกเป็น 2 ส่วน โดยเป็นของ subnetid ขนาด 8 บิต และ hostid ใหม่ที่มีขนาดเล็กลงเหลือเพียง 8 บิต	10
รูปที่ 7 Symmetric Multiprocessing	19
รูปที่ 8 Functional Multiprocessing	19
รูปที่ 9 On-Line Interactive System	21
รูปที่ 10 Off-Line Transaction Processing System.....	22
รูปที่ 11 Batch Processing System	23
รูปที่ 12 On-Line Real Time Interactive System.....	23
รูปที่ 13 แสดงให้เห็นการแบ่งกลุ่มข้อมูล.....	30

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Generic Network Configuration

ศัพท์เบื้องต้นเกี่ยวกับระบบ Network

ระบบเครือข่าย หรือ Network หมายถึงการที่เรานำคอมพิวเตอร์ ตั้งแต่ 2 เครื่องขึ้นไป มาเชื่อมต่อกัน เพื่อให้คอมพิวเตอร์ เหล่านี้ สามารถใช้ทรัพยากร ร่วมกันได้ เช่นใช้ Hard disk หรือ CD-ROM drive ร่วมกันได้ เพื่อเป็นการลดรายจ่าย ในการจัดซื้อ หรือเพื่อส่งข่าวสารหรือข้อมูลถึงผู้ใช้งานอื่นๆ ในเครือข่ายเป็นต้น ระบบเครือข่าย แบ่งเป็น 2 ลักษณะใหญ่ๆ คือเครือข่าย LAN และ WAN

LAN มาจาก Local Area Network คือเครือข่ายคอมพิวเตอร์ระยะใกล้ เช่นในโรงเรียน หรือสำนักงาน ที่ใดที่หนึ่ง การที่คอมพิวเตอร์แต่ละเครื่อง จะสามารถเชื่อมโยงถึงกันได้ ต้องมีอุปกรณ์สำคัญคือ LAN Card หรือ Network Interface Card (NIC) และสายที่ใช้ในการเชื่อมต่อ ซึ่งมีหลายชนิดเช่น Coaxial Cable, สาย Unshielded Twisted Pair (UTP) หรือสาย Fiber Optic ขึ้นอยู่กับ ชนิดของเครือข่ายว่าต้องการใช้งาน ใกล้-ไกลเท่าใด

WAN มาจาก Wide Area Network คือการเชื่อมต่อระบบ LAN ตั้งแต่ 2 ระบบขึ้นไปเข้าด้วยกัน อาจจะอยู่บริเวณเดียวกัน หรืออยู่ไกลกันมากก็ได้ ซึ่งการเชื่อมต่อนี้ สามารถทำผ่านสื่อต่างๆ ได้หลายชนิด เช่นสายทองแดงที่เป็น lease line (สายโทรศัพท์ธรรมดา) ผ่านไมโครเวฟ หรือผ่านดาวเทียมเป็นต้น อุปกรณ์ที่ทำให้การเชื่อมต่อสำเร็จได้คือ Router ซึ่งทำหน้าที่เป็น ประตูเชื่อมต่อ หรือ Gateway

การเชื่อมโยงคอมพิวเตอร์ หลายๆเครื่องเข้าด้วยกัน ถ้าเป็นระบบที่แตกต่างกัน ก็ไม่สามารถใช้งาน หรือติดต่อกันได้ ดังนั้นจึงจำเป็นต้องมีมาตรฐาน ในการติดต่อสื่อสารกัน เรียกว่า Protocol และในระบบ Internet ได้ใช้ TCP/IP (Transmission Control Protocol/ Internet Protocol) เป็นมาตรฐานในการติดต่อสื่อสารกัน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

IP Address

Internet Protocol Address ประกอบด้วยตัวเลขฐานสอง 4 bytes (32 bits) เวลาใช้เพื่อให้ง่ายต่อการเข้าใจ จะเขียนอยู่ในรูป เลขฐานสิบ ชั้นด้วยจุด "." เมื่อแปลงเป็นเลขฐานสิบแล้ว แต่ละชุดจะมีตัวเลขระหว่าง 0-255 เช่น 192.168.1.1 เป็นต้น เรียกย่อๆว่า IP Address

Internet Protocol Networks คือตัวเลข IP Addresses ที่เรียงต่อกันไป ซึ่งจะมีความหมายดังนี้ Address ทั้งหมดที่อยู่ภายใน network เดียวกันและมีตัวเลขเหมือนกัน เรียกว่า "network portion" ตัวเลขที่เหลือจะเรียกว่า "host portion" ตัวเลขที่ใช้ร่วมกันสำหรับทุก addresses ภายในเน็ตเวิร์ค เรียกว่า "netmask" และ netmask นี้ก็ใช้เป็นตัวกำหนดว่า addresses ใดอยู่ภายใน network หรือ addresses ใดไม่อยู่

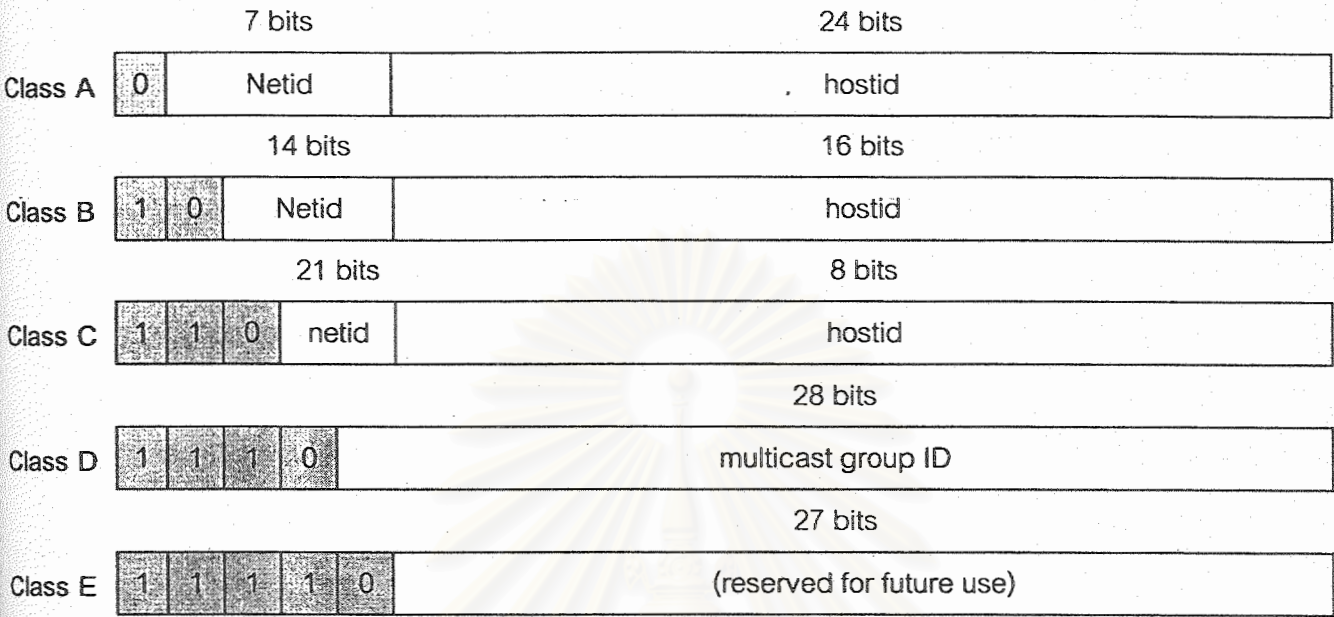
Network Portion	192.168.1.	ภายใน network นี้ ทุก address จะขึ้นต้นด้วยเลขนี้เหมือนกันหมด
Host Portion	0.1	แต่ละ host ก็จะมีเลข host portion เป็นของตัวเองไม่ซ้ำกัน
Host Address	192.168.1.10	Network Portion + Host Portion
Network Mask (netmask)	255.255.255.0	ทุก addresses ภายใน network เดียวกัน ต้องใช้ค่าเดียวกัน และหมายเลขนี้ ก็จะแสดงให้รู้ว่า ภายใน network นี้มี IPs รวมกันทั้งหมดกี่หมายเลข ในที่นี้ก็จะ มี 256 IPs
Network Address	192.168.1.0	
Broadcast Address	192.168.1.255	

ตารางที่ 1 อธิบายค่าต่างๆ ใน IP Address

สภามหาวิทยาลัย
จุฬาลงกรณ์มหาวิทยาลัย

Network Class

การแบ่ง IP Address ออกเป็นหลายระดับ ความจริงก็แบ่งตามความใหญ่ ของเครือข่ายนั่นเอง



รูปที่ 1 Network Class

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Network Class	Range
A	0.0.0.0 - 127.255.255.255
B	128.0.0.0 - 191.255.255.255
C	192.0.0.0 - 223.255.255.255
D	224.0.0.0 - 239.255.255.255
E	240.0.0.0 - 247.255.255.255

ตารางที่ 2 Network Class

Class A Network

ระบบเครือข่ายคลาส A มีแอดเดรสเริ่มต้นตั้งแต่ 0 จนถึง 127 (Network Portion) ตัวเลขที่เหลืออีกสามส่วน เป็น host portion จะเห็นว่าเป็นระบบเครือข่ายที่ใหญ่มาก สามารถมีอุปกรณ์ (router, host ฯ) ต่างๆ ไม่ซ้ำกันได้ถึง 16 ล้านตัว ระบบเครือข่ายคลาส A ได้ถูกกำหนดให้กับบริษัทต่าง จนหมดไปนานแล้ว

Class B Network

มีแอดเดรสเริ่มตั้งแต่ 128 จนถึง 192 มีระบบเครือข่ายคลาส B อยู่ 16,000 เครือข่าย ในแต่ละเครือข่าย จะมีแอดเดรสที่ไม่ซ้ำกันอยู่ 64,000 แอดเดรส และก็ถูกกำหนดให้กับองค์กร หรือบริษัทใหญ่ๆ หมดแล้วเช่นกัน เหตุที่เครือข่ายนี้ไม่ใหญ่เท่า เครือข่ายคลาส A เนื่องจากถูกกำหนดโดย netmask

Class C Network

มีแอดเดรสเริ่มตั้งแต่ 192 จนถึง 223 ซึ่งมีอยู่ประมาณ 2 ล้านเครือข่าย แต่ละเครือข่ายมีอุปกรณ์ได้สูงสุด 254 ตัว

Class D Network (Multicast)

หรืออาจเรียกได้ว่าเป็น เครือข่ายคลาส D มีแอดเดรสเริ่มต้นตั้งแต่ 224 จนถึง 239 สำหรับใช้งานในลักษณะเหมือนกับ วิทยุหรือโทรทัศน์ คือส่งออกไปอย่างเดียว ใครจะรับก็ไปดักรับเอา ส่วนแอดเดรส 240 ถึง 247 เป็นของระบบเครือข่ายคลาส E ซึ่งในส่วนนี้เก็บไว้ใช้ในอนาคต

IP Routing

IP Routing เป็นกระบวนการค้นหาเส้นทางในการส่งผ่านข้อมูลจากต้นทางไปยังที่หมายปลายทางโดยผ่านการส่งต่อของอุปกรณ์ IP ที่อยู่ในเน็ตเวิร์คซึ่งจะช่วยกันทำหน้าที่ส่งต่อข้อมูลไปจนกว่าข้อมูลจะถึงปลายทาง กลไกสำคัญที่ทำให้ IP เป็นโปรโตคอลสำหรับขนส่งข้อมูลไปยังทุกๆ ที่ในโลกบนอินเทอร์เน็ตที่ดีที่สุดขณะนี้คือการที่ IP มีกระบวนการ IP Routing นี้เอง สิ่งที่น่าสนใจที่สุดของ IP Routing คือการที่ต้นทางและปลายทางของการสื่อสารนั้นในบางโอกาสต่างก็อยู่บนเส้นไกลม การสื่อสารข้อมูลแต่ละครั้ง ข้อมูลจะต้องเดินทางผ่านโครงข่ายอันสลับซับซ้อนมากมาย แต่ในที่สุดข้อมูลก็สามารถส่งถึงกันได้ในเวลาอันรวดเร็วเป็นที่น่าอัศจรรย์ โครงข่ายอินเทอร์เน็ตคงไม่อาจเกิดขึ้นได้หากไม่มีโปรโตคอล IP ที่ช่วยขนส่งข้อมูลไปบนเครือข่ายอย่างมีประสิทธิภาพ การเข้าใจถึงกระบวนการ IP Routing จะช่วยให้เราเข้าใจคุณสมบัติของอินเทอร์เน็ตได้เป็นอย่างดี กระบวนการ IP Routing นี้ได้ถูกออกแบบมาอย่างชาญฉลาดและรัดกุมพอสมควรในอันที่จะให้บรรลุภารกิจในการขนส่งข้อมูล หลักการพื้นฐานของ IP Routing เริ่มต้นด้วยข้อกำหนดที่เรียบง่ายดังนี้

อุปกรณ์ที่ใช้ในเน็ตเวิร์ค

Host โฮสต์เป็นอุปกรณ์ที่ทำหน้าที่ให้กำเนิดข้อมูลในกรณีเป็นผู้ส่ง หรือทำหน้าที่รับข้อมูลไปใช้งานในกรณีเป็นผู้รับ การสื่อสารข้อมูลใดๆ จะต้องเป็นการสื่อสารจากโฮสต์ไปยังโฮสต์เสมอ

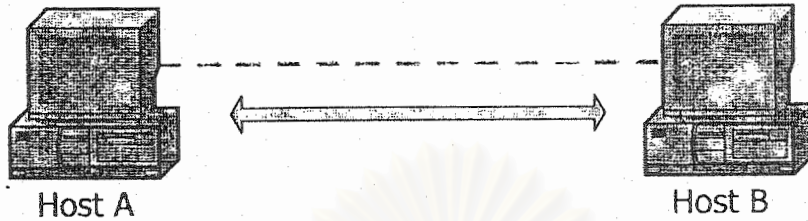
Router เราเตอร์เป็นอุปกรณ์สำคัญอย่างยิ่งสำหรับ IP ที่จะทำให้การขนส่งข้อมูลเป็นไปอย่างสมบูรณ์ เราเตอร์ทำหน้าที่ในการส่งผ่านข้อมูลจากเน็ตเวิร์คหนึ่งไปยังอีกเน็ตเวิร์คหนึ่ง ตำแหน่งของเราเตอร์จะอยู่ในจุดที่เชื่อมต่อระหว่างสองเน็ตเวิร์คเข้าด้วยกัน ด้วยข้อกำหนดของ IP ข้อมูลจะส่งไปถึงกันโดยตรงข้ามเน็ตเวิร์คได้ จะต้องอาศัยเราเตอร์เป็นผู้ทำหน้าที่ส่งผ่านข้อมูลไปให้ ดังนั้นเน็ตเวิร์คของ IP ถึงแม้ไม่ได้ต่อกันในทางกายภาพ แต่ก็สามารถสื่อสารกันได้โดยอาศัยเราเตอร์เป็นตัวเชื่อมต่อประสานเข้าด้วยกัน

Network เน็ตเวิร์คในที่นี้จะกล่าวถึงเฉพาะเน็ตเวิร์คในความหมายของ IP (IP Network) เท่านั้น ไม่รวมถึงเน็ตเวิร์คประเภทอื่น ใน IP นั้นจะมีการระบุหมายเลขประจำโฮสต์โดยใช้ IP Address เพื่อระบุตำแหน่งของต้นทาง และปลายทาง โดยใน IP Address นั้นนอกจากจะระบุตำแหน่งของโฮสต์แล้ว ยังใช้ระบุตำแหน่งของเน็ตเวิร์คที่โฮสต์นั้นเชื่อมต่อด้วย ทั้งในโปรโตคอล IP มีกระบวนการที่จะแยกหมายเลขประจำตัวของโฮสต์และของเน็ตเวิร์คออกจากกัน เพื่อให้อุปกรณ์ทั้งหลายสามารถพิจารณาได้ว่าทันทีว่าจะส่งข้อมูลที่ได้รับมานั้นไปในทิศทางใด

หลักการพื้นฐานของ IP Routing

IP Routing โดยใช้ Default Router

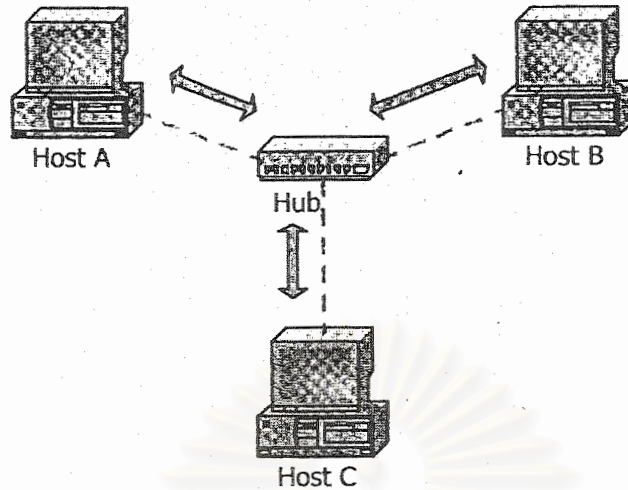
กระบวนการ IP Routing เริ่มต้นด้วยหลักพื้นฐานที่ไม่สลับซับซ้อนและเข้าใจได้ไม่ยากคือ



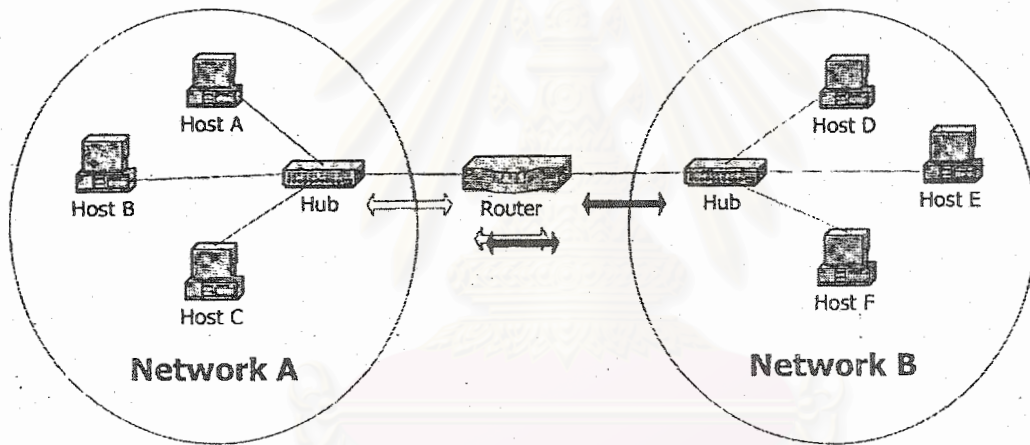
รูปที่ 2 การสื่อสารในการเชื่อมต่อแบบจุดต่อจุด

1. ถ้าโฮสต์ต้นทางและปลายทางต่อถึงกันโดยตรงเช่นการเชื่อมต่อแบบจุดต่อจุด ข้อมูลจะถูกส่งไปยังโฮสต์ปลายทางโดยตรง
2. ถ้าโฮสต์ต้นทางและปลายทางต่อเชื่อมร่วมอยู่ในเน็ตเวิร์กเดียวกัน เช่น อีเทอร์เน็ตหรือโทเคนริง ดังแสดงในภาพที่ 4.3 ข้อมูลจะถูกส่งไปยังโฮสต์ปลายทางโดยตรง
3. ถ้าไม่เป็นไปตามข้อที่ 1 และ 2 ข้อมูลจะถูกส่งไปยังดีฟอลต์เราเตอร์ เพื่อทำการส่งต่อข้อมูลไปยังปลายทางต่อไป
4. เมื่อเราเตอร์ได้รับ ข้อมูลจากข้อ 3 แล้ว ตรวจสอบดู หากพบว่าโฮสต์ปลายทางต่อร่วมอยู่บนเน็ตเวิร์กเดียวกันกับเราเตอร์ ให้ทำการส่งข้อมูลไปที่โฮสต์นั้น หากไม่ได้ต่อร่วมกันก็ส่งข้อมูลไปที่เราเตอร์ตัวต่อไป และกลับไปที่ยันตอนในข้อที่ 2 ใหม่ จนกว่าข้อมูลจะเดินทางถึงปลายทางหรือหมดเวลาในการส่ง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



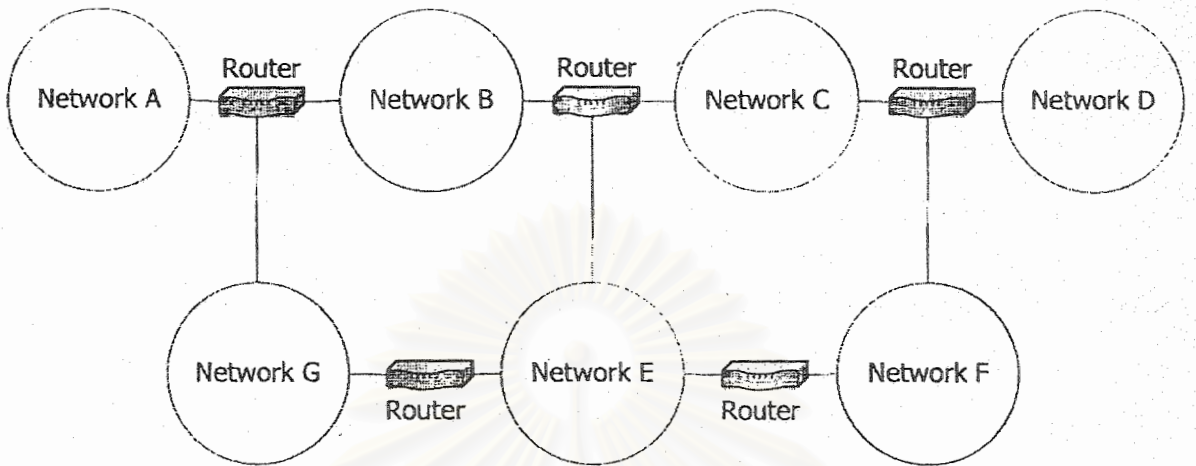
รูปที่ 3 การสื่อสารในเน็ตเวิร์กที่ต่อร่วมกัน (Shared Network)



รูปที่ 4 การสื่อสารระหว่าง 2 เน็ตเวิร์ก

หากเน็ตเวิร์กมีเพียง 2 เน็ตเวิร์กเหมือนในรูปที่ 4 มีเพียงแค้ดีฟอลต์เราเตอร์ก็คงจะเพียงพอและการทำงานในการส่งข้อมูลข้ามเน็ตเวิร์กก็คงจะไม่ยุ่งยากมากนัก และคงเป็นไปตามขั้นตอนข้างต้น หากสังเกตเห็นว่าตัวเราเตอร์เองนั้นจะมีเน็ตเวิร์กที่ต้องติดต่อด้วย 2 ฝั่ง คือ network A และ network B ซึ่งมีทิศทางการเคลื่อนที่ของข้อมูลเพียงเส้นทางเดียว มีเราเตอร์เพียงตัวเดียว ไม่ว่าจะปลายทางของข้อมูลจะไปที่ไหนหากมิใช่อยู่ในเน็ตเวิร์กเดียวกันแล้ว ข้อมูลทั้งหมดก็ต้องส่งผ่านเราเตอร์อยู่ดี โดยไม่ต้องทำการวิเคราะห์ใดๆ การที่ข้อมูลถูกส่งข้ามเน็ตเวิร์ก 1 ครั้ง เรียกว่า 1 ฮอป (hop) เปรียบเสมือนระยะในการเดินทางของข้อมูลจากภาพตัวอย่างข้อมูลเดินทางจากโฮสต์ต้นทางเพียง 1 ฮอปก็ถึงโฮสต์ปลายทาง การส่งต่อข้อมูลโดยเราเตอร์ก็มีเพียงส่งไปและส่งกลับระหว่าง network A กับ network B เท่านั้น แต่หาระยะทางถึงโฮสต์ปลายทางจะต้องเดินทางมากกว่า 1 ฮอปแล้ว เราเตอร์ก็จะทำงานยุ่งยาก ซับซ้อนขึ้นเพราะมีเน็ตเวิร์กอื่นๆ

ที่ไม่ได้เชื่อมต่อโดยตรงและต้องส่งข้อมูลผ่านเราเตอร์หลายตัว และมีหลายเส้นทางที่ข้อมูลสามารถเดินทางไปได้ ดังนั้นการส่งข้อมูลของเราเตอร์จึงเป็นปัจจัยสำคัญในการกำหนดประสิทธิภาพของ IP Routing



รูปที่ 5 การเชื่อมต่อกันของหลายเน็ตเวิร์ก

การเดินทางของข้อมูลโดยกระบวนการ IP Routing นั้นทำงานอยู่บนพื้นฐานของการส่งข้อมูลที่ละฮอป (hop-by-hop) คือเราเตอร์เองจะทำงานโดยรู้จักเฉพาะเน็ตเวิร์กที่ต่ออยู่กับตัวเองเท่านั้น หากโฮสต์ปลายทางมิได้อยู่ในเน็ตเวิร์กที่ต่อเชื่อมอยู่ ก็จะทำการส่งข้อมูลต่อไปอีกฮอปให้แก่เราเตอร์ตัวต่อไปส่งต่อ และถือว่าหมดหน้าที่ต่อข้อมูลนั้นแล้ว เพราะการส่งต่อข้อมูลไปเรียบร้อยแล้ว ส่วนจะถึงปลายทางหรือไม่ นั้นเป็นอีกเรื่องหนึ่ง และเราเตอร์ตัวอื่นๆ ที่อยู่ระหว่างทางก็เช่นกัน ก็จะส่งต่อข้อมูลไปเรื่อยๆ เช่นนั้นที่ละฮอปที่ละฮอปจนกว่าจะถึงปลายทางหรือหมดเวลา

เพื่อให้กระบวนการ IP Routing ดำเนินไปอย่างมีประสิทธิภาพจึงมีการเพิ่มความสามารถของเราเตอร์ให้มากขึ้นกล่าวคือ ในกรณีที่โฮสต์ปลายทางมิได้อยู่ในเน็ตเวิร์กที่ต่ออยู่กับตัวเองนั้นแทนที่จะทำการส่งต่อข้อมูลไปยังดีฟอลต์เราเตอร์ทั้งหมด ก็ให้เราเตอร์ทำการพิจารณาเน็ตเวิร์กปลายทางว่าอยู่ที่ใดแล้วจึงทำการส่งต่อข้อมูลนั้นไปยังเราเตอร์ที่อยู่ใกล้กับเน็ตเวิร์กมากที่สุด (ใช้จำนวนฮอปในการส่งข้อมูลน้อยที่สุด) เพื่อการนี้จึงจำเป็นต้องมีข้อมูลให้แก่ตัวเราเตอร์ว่าเน็ตเวิร์กใดควรจะส่งข้อมูลไปยังเราเตอร์ใด ข้อมูลเหล่านี้จะเก็บอยู่ในเรดดีตติ้งเทเบิล (Routing Table) ซึ่งจะประกอบด้วยข้อมูลดังนี้

Destination IP Address: หมายถึงแอดเดรสของโฮสต์หรือเน็ตเวิร์กปลายทาง

IP Address of a next-hop router: หมายถึง IP Address ของเราเตอร์ตัวอื่นที่ต่อโดยตรงอยู่บนเน็ตเวิร์กเดียวกัน

Flags: จะเป็นข้อมูลส่วนที่ขยายความเพิ่มเติมของ Destination IP Address และ Next-hop router

Interface: หมายถึงอินเตอร์เฟซของเราเตอร์ที่จะต้องใช้เพื่อการส่งข้อมูลออกไป

เมื่อเราเตอร์มีเราต์ติ้งเทเบิลแล้ว กระบวนการในการที่จะส่งข้อมูลจากเราเตอร์ตัวหนึ่งไปยังตัวต่อไป จะต้องนำข้อมูลในเราต์ติ้งเทเบิลไปร่วมพิจารณาด้วยดังนี้

- ค้นหาข้อมูลในเราต์ติ้งเทเบิลเพื่อหา IP Address ที่ตรงกันพอดีกับ IP Address ของโฮสต์ปลายทางของข้อมูล หากพบให้ส่งไปยังแอดเดรสที่ระบุอยู่ในฟิลด์ของ Next-hop router ทันที หากไม่พบข้อมูลให้ทำต่อในข้อถัดไป
- ค้นหาในเราต์ติ้งเทเบิลเพื่อหาเน็ตเวิร์คแอดเดรสที่ตรงกับเน็ตเวิร์คแอดเดรสของโฮสต์ปลายทาง หากพบให้ส่งไปยังแอดเดรสที่ระบุอยู่ในฟิลด์ของ Next-hop router ทันที หากไม่พบข้อมูลให้ทำต่อในข้อถัดไป
- ค้นหาในเราต์ติ้งเทเบิลเพื่อหาข้อมูลรายการที่ระบุว่า "Default" และส่งต่อไปยังแอดเดรสที่ระบุอยู่ในฟิลด์ของ Next-hop router

กระบวนการที่กล่าวมานี้จะทำให้การส่งต่อข้อมูลเป็นไปในทิศทางที่เหมาะสมและมีประสิทธิภาพที่สุด โดยอาศัยข้อมูลที่กำหนดไว้ก่อนแล้วของเราเตอร์ ซึ่งหากไม่พบข้อมูลที่ตรงกับโฮสต์หรือเน็ตเวิร์คเลยในเราต์ติ้งเทเบิลแล้ว ข้อมูลก็จะถูกส่งไปยังดีฟอลต์เราเตอร์เสมอ ดังนั้น หากเราเตอร์แต่ละตัวต่างก็มีเราต์ติ้งเทเบิลที่ถูกต้องแล้ว ข้อมูลก็จะถูกส่งไปเรื่อยๆ จนถึงปลายทางในที่สุด อาจจะล่าช้าหรือไม่มีประสิทธิภาพบ้างแต่ก็พอใช้งานได้

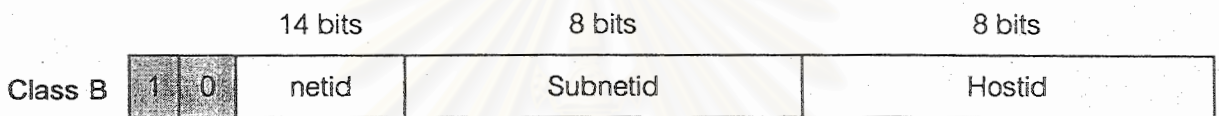
จะเห็นได้ว่าเราเตอร์ไม่จำเป็นต้องเก็บเราต์ติ้งเทเบิลของทุกๆ เน็ตเวิร์คหรือทุกๆ โฮสต์ไว้ทั้งหมด แต่จะอาศัยการกระจายกันของข้อมูลในเราเตอร์ทุกๆ ตัวบนเน็ตเวิร์คแต่ละตัวก็ส่งต่อข้อมูลให้ถูกต้อง ข้อมูลก็จะสามารถเดินทางถึงปลายทางได้ ด้วยเหตุนี้เองทำให้เน็ตเวิร์คสามารถขยายเพิ่มเติมออกไปได้เรื่อยๆ โดยไม่จำกัดและไม่ต้องทำการแก้ไขโครงสร้างของเน็ตเวิร์คเดิม อินเทอร์เน็ตจึงแผ่ขยายครอบคลุมโลกได้อย่างรวดเร็ว

Subnet Addressing

ในตอนเริ่มต้นใช้โปรโตคอล TCP/IP นั้นการแบ่ง IP Address ออกเป็นแอดเดรสของเน็ตเวิร์ค (netid) และแอดเดรสของโฮสต์ (hostid) เป็นไปตามกติกาที่ระบุของแต่ละคลาส ต่อมีผู้เสนอให้มีการการแบ่งเน็ตเวิร์คย่อยภายในแต่ละ netid เพิ่มขึ้นอีก เพื่อจะได้ใช้งาน IP Address ได้อย่างมีประสิทธิภาพที่สุด เนื่องในคลาส A และ คลาส B นั้นมีการจัดสรรส่วนที่เป็น hostid ในแต่ละเน็ตเวิร์คเป็นจำนวนมาก คือในเน็ตเวิร์คคลาส A แต่ละเน็ตเวิร์คนั้นสามารถมีจำนวนโฮสต์ได้มากถึง $2^{24}-2=16,777,214$ โฮสต์ และสำหรับในแต่ละเน็ตเวิร์คของคลาส C นั้นสามารถมีจำนวนโฮสต์ได้สูงสุดถึง $2^{16}-2=65,534$ โฮสต์ ซึ่งการที่

จะนำ IP Address มาใช้อย่างทั่วถึงนั้นมีโอกาสเป็นไปได้ยากมาก ทั้งคลาส A และ คลาส B เพราะมีโอกาสน้อยมากที่จะมีเน็ตเวิร์คใดในโลกที่มีจำนวนโฮสต์มากมายขนาดนั้นอยู่ในเน็ตเวิร์คเดียว ดังนั้น IP Address ที่จัดสรรให้ไปในแต่ละเน็ตเวิร์คของคลาสเหล่านั้นจึงถูกใช้ไม่หมดและไม่สามารถนำไปใช้ประโยชน์ที่อื่นได้

การทำ subnet คือการแบ่งเน็ตเวิร์คย่อยภายในเน็ตเวิร์คเพื่อให้แต่ละเน็ตเวิร์คมีขนาดที่เหมาะสมกับปริมาณโฮสต์ที่มีอยู่ โดยใช้หลักการเดียวกันกับการนำ IP Address มาแยกออกเป็น Host id และ Network id คือแทนที่จะให้ค่า Host id เป็นค่าอิสระตั้งแต่ 1 จนถึงค่าสูงสุด ก็ทำการจัดกลุ่มของ host id เหล่านั้นออกเป็นกลุ่มของเน็ตเวิร์คย่อย คือ นำค่าในส่วนที่เป็น host id เดิมมาแยกออกเป็นสองส่วนคือ subnet id และเป็น host id ใหม่ ซึ่งจะทำให้สามารถจัดสรรการใช้งาน IP Address ได้อย่างเหมาะสมกับการมีอยู่จริงของโฮสต์ในแต่ละเน็ตเวิร์ค



รูปที่ 6 แสดงการจัดแบ่ง IP Address ของคลาส ออกเป็นเน็ตเวิร์คย่อยด้วยวิธี Subnetting โดยแบ่งพื้นที่ส่วนที่เป็นของ hostid เดิมออกเป็น 2 ส่วน โดยเป็นของ subnetid ขนาด 8 บิต และ hostid ใหม่ที่มีขนาดเล็กลงเหลือเพียง 8 บิต

Class B	จำนวนเน็ตเวิร์คที่มีได้	จำนวนโฮสต์สูงสุดในแต่ละเน็ตเวิร์ค
เดิม	16,382	65,532
หลังการ subnet	4,161,028	254

ตารางที่ 3 ผลกระทบต่อจำนวนของและขนาดของเน็ตเวิร์คเมื่อถูก subnet

ผลที่ได้คือขนาดของเน็ตเวิร์คจะเล็กลงและมีจำนวนมากขึ้น จากเดิม 16,382 เน็ตเวิร์คก็เพิ่มเป็น 4,161,028 ($16,382 * 254$) และในขณะเดียวกันจำนวนโฮสต์ในแต่ละเน็ตเวิร์คจะลดลงจาก 65,532 เหลือเพียง 254 โฮสต์ อย่างไรก็ตามการ subnet นั้นไม่จำเป็นต้องมีขนาดของ subnet id คงที่ตายตัวเสมอไป ผู้บริหารระบบสามารถปรับเปลี่ยนได้ตามความเหมาะสมของการใช้งาน เช่นอาจจะมีจำนวนเน็ตเวิร์คน้อยลง และจำนวนโฮสต์มากขึ้นสามารถกระทำได้โดยการแบ่งขนาดของ subnet id และ host id ใหม่ตามต้องการ

นอกจากการที่สามารถใช้ IP Address ได้อย่างมีประสิทธิภาพแล้ว ข้อดีอีกอย่างหนึ่งของการ subnet คือช่วยให้ประสิทธิภาพการสื่อสารดีขึ้นด้วย กล่าวคือ กระบวนการของ TCP/IP บางประเภทมีการใช้การสื่อสารแบบบรอดคาสต์ (broadcast) เพื่อทำการสื่อสารกระจายไปยังทุกๆ โฮสต์ที่อยู่ในเน็ตเวิร์ควง

เดียวกัน ดังนั้นหากเป็นเน็ตเวิร์คคลาส A ซึ่งมีโฮสต์ได้ถึง 16 ล้านโฮสต์แล้ว การสื่อสารด้วยวิธีบรอดคาสต์แต่ละครั้งจะเป็นการกระจายข้อมูลไปยังเครื่องอื่นๆ จำนวนมาก และใช้แบนด์วิดท์มากมายมหาศาลทั่วไปทั้งเน็ตเวิร์ค ส่งผลกระทบต่อการสื่อสารตามปกติอย่างยิ่ง และหากเน็ตเวิร์คประเภทนี้ถูกโจมตีแล้วก็มีโอกาสมากที่การสื่อสารภายในเน็ตเวิร์คจะเป็นอัมพาตได้อย่างรวดเร็ว ผู้ออกแบบเน็ตเวิร์คส่วนใหญ่จึงมักจะหลีกเลี่ยงการออกแบบให้เน็ตเวิร์คมีขนาดใหญ่เกินไป เนื่องจากควบคุมได้ยุ่งยากและมีประสิทธิภาพต่ำ วิธีการ subnet จึงเป็นส่วนที่ถูกนำมาใช้ในการออกแบบเสมอ

โดยทั่วแล้วมักจะพบเห็นการ subnet สำหรับ IP Address ในคลาส B เสียเป็นส่วนใหญ่ เนื่องจากคลาส B มีผู้ใช้งานกันแพร่หลาย ส่วนคลาส A จะพบได้ไม่บ่อยนัก เพราะมีผู้ได้รับจัดสรรไม่มาก ส่วนคลาส C ก็อาจจะพบบ้างบ้าง แต่เนื่องจากคลาส C มีขนาดของเน็ตเวิร์คไม่ใหญ่อยู่แล้ว จึงสามารถแบ่งย่อยออกไปได้อีกไม่มากนัก อย่างไรก็ตาม IP Address ทุกคลาสล้วนแต่สามารถถูกนำมา subnet ได้ทั้งสิ้น

Subnet Mask

หากกล่าวถึงการ subnet แล้วย่อมจะต้องกล่าวถึง subnet mask ด้วย การที่มีเฉพาะ IP Address เพียงอย่างเดียวนั้น กรณีที่เป็นการกำหนด netid และ hostid ตามที่ระบุในคลาสต่างๆ นั้นเราก็สามารถทราบได้ว่าค่าทั้งสองได้ไม่ยากนัก โดยพิจารณาว่า IP Address ดังกล่าวอยู่ในช่วงใด อยู่ในคลาสใด หลักจากนั้นก็ สามารถแยก netid และ hostid ได้จากการเปรียบเทียบกับมาตรฐานของคลาสเหล่านั้น

แต่เมื่อมีการแบ่งเน็ตเวิร์คย่อยโดยการ subnet แล้ว ย่อมไม่สามารถใช้วิธีการข้างต้นเพื่อหา netid และ hostid ได้อีกต่อไป เนื่องจาก subnet นั้นสามารถกำหนดได้โดยผู้ออกแบบเน็ตเวิร์คเอง และมีได้มีข้อบังคับแต่อย่างใด ดังนั้นจึงจำเป็นต้องมีการระบุค่าใดค่าหนึ่งไว้เพื่อให้สามารถนำมาใช้ในการหาค่าจาก IP Address ได้ว่าเป็น hostid, netid และ subnet id และค่านั้นก็คือ subnet mask นั่นเอง

Subnet mask เป็นตัวเลขขนาด 32 บิตเท่ากับ IP Address ทำหน้าที่ระบุหมายเลขของ host id และ net id + subnet id ของโฮสต์นั้น การกำหนดค่าของ subnet mask จะอยู่ในรูปแบบเดียวกับ IP Address คือทำการแบ่ง subnet mask ออกเป็นเลข 8 บิตจำนวน 4 ชุด และแยกแต่ละชุดออกจากกันด้วยจุด (.)

ตัวอย่างของ Subnet mask เช่น

FF.FF.FF.00 (hex)

255.255.255.0 (dec)

ค่า subnet mask นี้จะต้องกำหนดไว้บนทุกโฮสต์คู่กันกับค่า IP Address เสมอ เนื่องจากโปรโตคอล IP จำเป็นต้องใช้ค่านี้ไปคำนวณค่า netid ซึ่งจะเป็นอย่างยิ่งในกระบวนการ IP Routing อย่างไรก็ตามค่า

subnet mask นี้จะไม่ถูกส่งไปด้วย โฮสต์ทั้งห้ายสามารถนำค่า subnet mask มาทำการทางคณิตศาสตร์กับ IP Address ก็จะสามารถหาค่า host id, subnet id, netid ออกมาได้โดยวิธีดังนี้

$$\text{Netid} + \text{subnet id} = [\text{IP address}] \text{ AND } [\text{Subnet Mask}]$$

$$\text{Hostid} = [\text{IP Address}] \text{ AND } (\text{NOT } ([\text{Subnet Mask}]))$$

ตัวอย่าง

$$\text{IP Address} = 192.168.15.20 \quad \text{Subnet Mask} = 255.255.255.0$$

$$\text{Netid} = 192.168.15.20 \text{ AND } 255.255.255.0 = 162.168.15.0$$

$$\text{Hostid} = 192.168.15.20 \text{ AND } 0.0.0.255 = 0.0.0.20$$

หรือพูดง่าย ๆ ได้ว่า "Netid ก็คือ IP Address ส่วนที่ตรงกับบิตของ Subnet mask ที่มีค่าเป็น 1 ส่วน hostid คือ IP Address ส่วนที่ตรงกับ Subnet mask ที่มีค่าเป็น 0" นั่นเอง

ดังนั้นจึงจะสังเกตเห็นว่านอกจากการกำหนด IP Address ที่ถูกต้องแล้ว การกำหนดค่า Subnet Mask ก็มีผลต่อ IP Routing เช่นเดียวกัน การกำหนดค่า Subnet Mask ผิดพลาดย่อมจะส่งผลให้การสื่อสารข้อมูลของ IP ไม่สามารถจะกระทำได้เช่นกัน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Quick Command Directory for DOS, Windows/NT & UNIX

	DOS, NT	UNIX
File Handling Commands		
List the files in a directory	dir	ls
Copy a file	copy, xcopy	cp
Delete a file	del (erase)	rm
Rename or move a file	rename, move	mv
Display a file on the screen	type	cat
Display a filetype with pauses between screen fills	fnmore	more, less
Print a file	print	lp
Edit or create a text file	edit, edlin	pico
Backup your files	backup	tar, cpio
Compare two files	comp, fc	diff
Find text in a file	find	grep
Sort a file	sort	sort
Retrieve a deleted file	undelete	
Allow others to see/use a file(s)	Disk: Share as...(1)	chmod
Directory Handling Commands		
	DOS, NT	UNIX
Change directories	chdir (cd)	cd
Display current directories	chdir (cd)	pwd
Make a new directory	mkdir (md)	mkdir
Remove a directory	rmdir (rd)	rmdir
Utilities		
	DOS, NT	UNIX
Change password	Change Password (2)	chpw
Check remaining disk space	chkdisk	quota -v
Clear screen	cls	clear
Close a Window	exit	exit
Check system date	date, time	date
Log off of the computer	File: Logoff(2)	exit
Format a new disk(ette)	format	initialize
File Transfer Protocol	ftp	ftp
Display online manual	help	man, man utcc
Send or receive mail	mail(1)	pine
Check amount of memory available	mem	vmstat
Set terminal type	mode	TERM=type
Recall previous commands	press the [up arrow]	history
Define an alias		alias
Capture a session of commands /responses		script
Send a file		
Set terminal features		stty
Running Applications and Commands		
	DOS, NT	UNIX
Automate a series of commands	fn .bat	.profile, .kshrc
Run a system command interactively	cmd	cmd
Run a system command as a subprocess	start, cmd low(2)	cmd &

Run a system command in batch	start, cmd low(2)	batch
Set search path to locate commands	path	PATH=
Show which programs are executing	[Ctrl][Esc](3)	ps
Stop an executing program	End Task(3)	kill
Creating and Running Programs	DOS, NT	UNIX
Compile a C program	Depends on version	cc
Compile a FORTRAN program	Depends on version	f77
Link or load object modules		ld
Run a compiled program in the background		a.out &
Run a compiled program interactively		a.out
Submit a command at a specific time		at, cron

Usage Notes:

- If a command has two names, they are separated by parentheses. Example: del (erase).
- If two commands perform similar functions, they are separated by a comma. Example: edit.
- Items separated by colons represent menu choices. For example, "File: Share as..." means choose "Share as..." from the "File" menu.
- Keyboard keys are enclosed in square brackets. For example, "Press [up arrow]" means press the key with the "up arrow" symbol.
- cmd represents any system command.
- fn represents any filename.
- Any item left blank means that the system does not have that capability.

DOS, Windows for Workgroups, Windows/NT Notes:

- File sharing, logoff, passwords under DOS relate only to Windows/NT.
- Logoff and changing passwords relate only to Windows/NT. They are preceded with [Ctrl] - [Alt] - [Delete].
- Commands that relate to programs which are executing are preceded with [Ctrl] - [Esc].

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Client/ Server Model

องค์ประกอบของ Client/ Server

- Client
มักจะเรียกว่า ตัวลูก คือ เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นผู้รับ-ส่งข้อมูลข่าวสาร และคำสั่งจากผู้ใช้ระบบงานไปให้แก่ Server (ตัวแม่) เพื่ออ่านข้อมูลประมวลผลและส่งกลับมาให้ผู้ใช้
- Server
มักจะเรียกว่า ตัวแม่ คือ เครื่องคอมพิวเตอร์ ที่ทำหน้าที่เป็นผู้รับ-ส่งข้อมูลข่าวสาร คำสั่งจาก Client เพื่ออ่านข้อมูลผล และส่งกลับมาให้ Client ซึ่ง Server 1 ตัวอาจจะมี Client ที่ต่อเชื่อมอยู่ในระบบงานได้หลายตัว และในแต่ละเครือข่ายอาจจะมี Server ที่ตัวก็ได้ตามความเหมาะสมของแต่ละระบบงาน
- Networking
คือ ระบบงานที่ประกอบไปด้วยอุปกรณ์ ฮาร์ดแวร์ และซอฟต์แวร์เพื่อเป็นทางเดินให้กับข้อมูลข่าวสาร คำสั่ง โปรแกรมที่มีการรับ-ส่งระหว่าง Client กับ Server ที่ต่อเชื่อมโยงกัน

เทคโนโลยี Client/ Server ในปัจจุบัน

การพัฒนาแบบ Client/ Server คือ ระบบงานที่มีการจัดแบ่งหน้าที่การทำงาน การประมวลผลของแต่ละงานให้เครื่องคอมพิวเตอร์ (Client หรือ Server) ที่มีความเหมาะสมมากที่สุดทำการประมวลผล เพื่อให้เกิดประสิทธิภาพในการทำงานสูงสุด เช่น

- Client ควรจะทำงานเกี่ยวกับระบบการรับ-แสดงผลทางจอภาพ
- Server ควรจะทำงานทางด้าน Database Management & Storage เป็นต้น

โดยทั่วไปการออกแบบระบบงานแบบ Client/ Server มักจะออกแบบให้เครื่องแม่ข่ายและลูกข่ายทำงานในลักษณะดังกล่าวข้างต้น ดังนั้นในอนาคตไม่ว่าจะมีการเพิ่มขยายเครือข่าย หรือ Client มากแค่ไหนก็ตาม งานที่เพิ่มขึ้นจะอยู่ที่ Client เกือบทั้งหมด โดยที่ Server จะมีงานเพิ่มเพียงคำสั่งโปรแกรมจาก Client ที่เพิ่มขึ้นมาเท่านั้น

ในการออกแบบโปรแกรมประยุกต์ หน้าที่ของ Client ในการประมวลผลนั้นควรจะต้องกำหนดให้ Client ทำหน้าที่ตรวจสอบความเป็นไปได้ของข้อมูลที่ผู้ใช้บันทึกเข้ามา ตรวจสอบผิดพลาดของข้อมูล เพื่อ

ป้องกัน กลั่นกรองไม่ให้ Client ส่งข้อมูลที่ผิดๆ ไปให้ Server ทำงาน ซึ่งมีผลทำให้ลดภาระงานของ Server และปริมาณงานบนเครือข่ายลดลง ยังมีผลถึงประสิทธิภาพที่สูงขึ้นด้วย

ข้อดีของการพัฒนาระบบงานแบบ Client/ Server มีดังนี้

1. ประหยัดงบประมาณในการลงทุน
2. เพิ่ม/ ก่อให้เกิดประสิทธิผลสูง
3. มีความยืดหยุ่น และสามารถขยายขีดความสามารถ/ ประสิทธิภาพได้
4. ก่อให้เกิดการใช้ทรัพยากรที่มีอยู่ให้เกิดประโยชน์สูงสุด
5. สามารถบริหาร ควบคุมจากส่วนกลาง
6. มีคุณสมบัติที่เป็นระบบงานแบบเปิด
 - สามารถทำงานภายใต้ Multiple Environments คือ สามารถทำงานประมวลผลงานภายใต้ Platform และ Environment ที่หลากหลาย
 - สามารถใช้ Database Application ที่หลากหลายได้ ผู้ใช้สามารถใช้ Application Software โปรแกรมที่คุ้นเคยเพื่อเข้าถึงฐานข้อมูลได้ ผู้พัฒนาระบบงานสามารถเลือกใช้ Front-end/ Programming Language/ Development Tools ได้ตามความเหมาะสม ความชำนาญ
 - สามารถทำงาน ประมวลผลงานข้ามระหว่างระบบงานฯ ต่างๆ ได้ และยังสามารถโยกย้ายระบบงานจากแพลตฟอร์มหนึ่งไปอีกแพลตฟอร์มหนึ่งได้

จุดอ่อน (ข้อด้อย) ของระบบงานแบบ Client/ Server

1. งบประมาณการลงทุนที่ซ่อนเร้น มีผู้ที่พยายามพัฒนาระบบงานคอมพิวเตอร์แบบ Client/ Server คาดหมายงบประมาณในการลงทุนที่ผิดพลาดไป ซึ่งรายการงบประมาณการลงทุนที่ควรจะเป็นสรุปได้เป็น 7 รายการดังนี้
 - Hardware and Equipment
 - System Software and Utility
 - Network and Data Communication
 - Commercial DBMS and Development Tools
 - Application Software Development and Consulting Services
 - Training and Education
 - Maintenance
 2. ค่าใช้จ่ายในการฝึกอบรม การรับบริการ การบำรุงรักษา และการบริหารจัดการสูง
 3. ผู้ให้บริการมีหลากหลาย
- การพัฒนาระบบงานแบบ Client/ Server จำเป็นต้องอาศัยองค์ประกอบทางด้าน Hardware, System

Software, Utility, DBMS, Development Tools, Network and Data Communication ที่ผู้ขายแต่ละด้านมีความสามารถ ความเชี่ยวชาญและเหมาะสมแตกต่างกันไป ซึ่งมีผลทำให้ผู้พัฒนาระบบงานฯ บางคนไม่สามารถคัดเลือก หรือตัดสินใจ เลือกใช้องค์ประกอบเหล่านั้นได้อย่างเหมาะสม และอาจมีผลกระทบไปถึงความสำเร็จ หรือ ล้มเหลวของโครงการฯ ได้

4. ขาดเครื่องมือในการบริหารจัดการระบบงานแบบกระจายศูนย์คอมพิวเตอร์
 5. ขาดมาตรฐาน ข้อจำกัดในข้อ 3 มีผลทำให้ขาดมาตรฐานของการประมวลผลร่วมกันข้าม Platform ไม่สามารถทำงานอย่างมีประสิทธิภาพเท่าที่ควรจะเป็น
 6. เทคโนโลยีและเทคนิคยังขาดความสมบูรณ์
 7. ขาดโปรแกรมช่วยในการปรับเปลี่ยน/โยกย้ายระบบงาน
- การโยกย้าย ปรับเปลี่ยน ระบบงานเดิมที่มีลักษณะเป็น text based, proprietary system หรือ ระบบงานเดิม ไปเป็นระบบงานแบบใหม่ที่เป็นแบบ Client/ Server ใช้เทคนิคของ Graphical User Interfaces (GUI) นั้น ยังขาดเครื่องมือที่จะช่วยในการโยกย้ายระบบงานฯ ทำให้เป็นอุปสรรค และข้อจำกัดของการปรับเปลี่ยนระบบงานเดิมที่พัฒนาและใช้งานมานานๆ แล้ว

Server

ความหมาย

Server หมายถึง เครื่องคอมพิวเตอร์ทุกๆ ขนาด ทุกๆ ชนิด ที่ทำหน้าที่ตามที่เรากำหนดได้หลากหลาย ไม่ว่าจะ File Sharing, Applications Processing, บริหารจัดการเกี่ยวกับข้อมูล บริหารจัดการเกี่ยวกับระบบงานเครือข่ายและการสื่อสารโทรคมนาคม ประมวลผลตามคำสั่งโปรแกรม ค้นหา/ เรียกข้อมูลจากระบบเครื่องคอมพิวเตอร์อื่นๆ ในเครือข่าย รับ-ส่งข้อมูลไปให้ระบบเครื่องคอมพิวเตอร์อื่นๆ ในเครือข่าย และยังสามารถทำหน้าที่อื่นๆ ตามที่จะกำหนดได้อีกด้วย

ประเภทของ Server

Server สามารถแบ่งออกได้หลายประเภทตามที่จะกำหนดหน้าที่ คุณสมบัติให้เหมาะสมกับระบบงานที่จะใช้ เช่น File Server, Data Server, Database Server, Compute Server, Communication Server, Application Server เป็นต้น

ขนาดของ Server

ขนาดของ Server มีได้ตั้งแต่ PC ขนาดเล็กไปจนถึง High Power Workstation ขนาดใหญ่มาก (RAM อาจจะตั้งแต่ 4 MB ขึ้นไป ตามแต่ผู้ผลิตฮาร์ดแวร์)

มาตรฐานประสิทธิภาพ

ได้มีหน่วยงานองค์กร สมาคม ที่รวมกลุ่มกันกำหนดมาตรฐานทางด้านประสิทธิภาพของอุปกรณ์ Client และ Server เช่น Transaction Processing Counsel (TPC) เป็นสมาคมของผู้ผลิตฮาร์ดแวร์และซอฟต์แวร์ ก่อตั้งเมื่อปี 1988 เพื่อร่วมกันกำหนดมาตรฐานของการประมวลผลฐานข้อมูล โดยสมาคมฯ นี้ ได้กำหนดมาตรฐานได้ 3 กลุ่มระดับคือ

- TPC-A

ใช้หลักการคำนวณประสิทธิภาพของการประมวลผล จำนวนรายการ (Transaction) ที่ถูกบันทึกเข้ามาจากหลายจอภาพ (Terminal/ PC-Client) ที่สามารถทำงานได้ภายในเวลา 1 วินาที เป็นตัววัดประสิทธิภาพ

- TPC-B

เป็นมาตรฐานของการประมวลผลของระบบงานที่เน้นที่ Database Engine และประสิทธิภาพของการประมวลผลแบบ Batch File Processing

- TPC-C

เป็นมาตรฐานของการทดสอบประสิทธิภาพของระบบงานแบบ Client/ Server ในสิ่งแวดล้อมหลากหลายรูปแบบของฮาร์ดแวร์และซอฟต์แวร์ แต่จะทำงานประเภท OLTP (On-Line Transaction Processing)

ในปัจจุบัน มาตรฐานหลักเกณฑ์ของ TPC ได้กลายเป็นมาตรฐานที่เปรียบเทียบอ้างอิงถึงประสิทธิภาพการทำงาน การประมวลผลงานของเครื่องคอมพิวเตอร์ ที่เป็นที่ยอมรับและใช้กันมากดังจะเห็นได้จากการเข้าร่วมสัมมนา หรืออ่านเอกสารข้อมูลทางด้านเทคนิคของเครื่องคอมพิวเตอร์ยี่ห้อต่างๆ รุ่นต่างๆ ของผู้ผลิตฯ เราจะพบข้อมูลอ้างอิงถึงประสิทธิภาพการทำงานการประมวลผลงานของเครื่องคอมพิวเตอร์เหล่านั้น โดยอาศัยเกณฑ์ของ TPC เป็นมาตรฐานอ้างอิงอยู่บ่อยครั้ง

การแบ่งประเภทของ Server

ในปัจจุบันมีนักคอมพิวเตอร์ และผู้ใช้ระบบงานคอมพิวเตอร์จำนวนมาก มีความสับสน ในการแบ่งประเภทของคอมพิวเตอร์ Server ที่ขายกันอยู่ในท้องตลาด ทั้งนี้อาจเนื่องจากการโฆษณาประชาสัมพันธ์ของผู้ผลิตฯ หรือความเจริญก้าวหน้าที่รวดเร็วมากของเทคโนโลยีคอมพิวเตอร์และสารสนเทศได้ จึงทำให้มีการเรียกชื่อ Server ตามชื่อทางการค้า ตามชื่อทางด้านการทำงานและหน้าที่ที่หลากหลายออกไป ซึ่งจะขอแบ่งประเภทของ Server ตามลักษณะของการประมวลผลข้อมูลเป็น 6 ประเภทคือ

1 File Server

คือเครื่องคอมพิวเตอร์ Server ที่ทำหน้าที่จัดการด้าน application และ data files โดยจะรับ-ส่งข้อมูลไปให้ Client ตามที่กำหนด มีลักษณะการทำงานเป็น I/O Oriented เป็นส่วนใหญ่ ในการทำงานการประมวลผลงาน File Server มักจะอ่านข้อมูลจากฐานข้อมูลที่มีอยู่ทั้งหมด และส่งผ่านเข้าไปในระบบงานเครือข่าย เพื่อไปที่ Client ที่ต้องการ จากนั้น Client จะทำหน้าที่ประมวลผลงานของข้อมูลตามคำสั่งโปรแกรมที่ถูกกำหนดไว้

File Server ควรจะมีอุปกรณ์สำหรับเชื่อมโยงเครือข่าย มีฮาร์ดดิสก์ที่มีความจุปริมาณมากๆ และ ทำงานได้รวดเร็ว ในการทำงานรูปแบบนี้ อาจมีโอกาที่จะเกิดปัญหาในระบบของ file ได้ เช่น File Locking คือการที่ Client ตัวใดตัวหนึ่งเรียกและประมวลผลข้อมูลจากแฟ้มข้อมูลใดแฟ้มข้อมูลหนึ่งที่อยู่ใน Server จะมีผลทำให้ Client ตัวอื่นๆ ไม่สามารถเรียก และประมวลผลข้อมูลจากแฟ้มข้อมูลนั้นๆ จนกว่า Client ตัวแรกจะประมวลผลเสร็จ

2 Application Server

คือ เครื่องคอมพิวเตอร์ Server ที่ทำหน้าที่เสมือนเป็น host โดยอาจสามารถนำเอา application software จาก host เดิม มาใช้ในสิ่งแวดล้อมของ application Server ได้เลย บางครั้งอาจเรียกว่า host-based Client/ Server และเราอาจพัฒนาระบบงานฯ ให้มีการใช้เทคนิคของ GUI (Graphical User Interface) เพิ่มเติมที่ Client ได้

3 Data Server

คือ เครื่องคอมพิวเตอร์ Server ที่ทำหน้าที่จัดเก็บและจัดการเกี่ยวกับข้อมูล (Data Oriented) โดยเฉพาะเป็นเสมือนคลังข้อมูลขนาดใหญ่ โดยทั่วไปจะต่อเชื่อมโยงกับ Compute Server และมักจะถูกกำหนดไม่ให้ประมวลผลทางด้านโปรแกรมใดๆ (Application Logic Processing) ยกเว้น การประมวลผลสถานะเบื้องต้นของตัวข้อมูลเท่านั้น เช่น Data Validation เป็นต้น Data Server จำเป็นต้องมีโพรเซสเซอร์ที่มีประสิทธิภาพสูง หน่วยความจำมากๆ และ Disk Storage ที่มีความจุมากๆ นอกจากนี้ Data Server 1 ตัว อาจจะต่อเชื่อมเพื่อให้บริการข้อมูลกับ Compute Server ได้ หลายตัวขึ้นอยู่กับการออกแบบ Application Software Environment ที่ออกแบบไว้

4 Compute Server

คือ เครื่องคอมพิวเตอร์ Server ที่ทำหน้าที่รับคำร้องขอข้อมูลจาก client ส่งต่อไปที่ Data Server และรับข้อมูลจาก Data Server (อาจจะมีการประมวลผลบางส่วน) และส่งกลับไปที่ client ตัว Compute server ควรจะต้องมีโพรเซสเซอร์ที่มีประสิทธิภาพสูง RAM มากๆ แต่ Disk Storage ไม่จำเป็นต้องมีความจุมากนัก

5 Database Server

คือ เครื่องคอมพิวเตอร์ Server ที่ทำหน้าที่รับคำร้องขอข้อมูลจาก client ค้นหาข้อมูล และส่งกลับไปให้ client โดยที่โปรแกรม (Application Program Logic) จะอยู่ที่ client (มีลักษณะการทำงานเหมือนกับรวม Data Server กับ Compute Server เข้าด้วยกัน) ลักษณะการทำงานแบบนี้เป็นการประมวลผลที่อาจเรียกได้ว่าเป็นรูปแบบที่แท้จริงของ Client/ Server System Development

Database Server ควรจะต้องออกแบบให้ส่งข้อมูลเฉพาะที่ client ต้องการเท่านั้น ไม่ส่งทั้งหมดของ File/ Table ซึ่งทำให้เพิ่มประสิทธิภาพของระบบงานเครือข่าย การติดต่อสื่อสาร (Networking and Data Communication)

การออกแบบ/ เลือก Database Server นี้ เราควรจะต้องพิจารณาขนาดของข้อมูล, Response Time ที่คาดหวัง จำนวนผู้ใช้งาน และรูปแบบของ Networking ที่ใช้ ทั้งนี้เพื่อให้เกิดประสิทธิภาพสูงสุด

6 Communication Server

คือ เครื่องคอมพิวเตอร์ Server ที่ทำหน้าที่ในการบริหารจัดการ Gateways เพื่อทำการติดต่อสื่อสารกับ LAN ที่มีลักษณะแตกต่างกัน ระบบเครือข่ายที่แตกต่างกัน รวมทั้งเครื่องข่ายที่ใช้ Midrange/ Mainframe Computer เพื่อสามารถติดต่อสื่อสาร แลกเปลี่ยนข้อมูลข่าวสารระหว่างกันได้อย่างมีประสิทธิภาพ

โดย Communication Server นี้จะต้องมี Multiple Slots โพรเซสเซอร์ความเร็วสูง ทำการแปล Networking Protocols ที่หลากหลายในเครือข่าย

คุณสมบัติของ Server

ถึงแม้ว่าเครื่องคอมพิวเตอร์ แทบทุกรุ่น แทบทุกขนาด หรือแทบทุกยี่ห้อจะสามารถนำมาเป็นเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Server ในระบบงานเครือข่ายได้ก็ตาม แต่นักวิเคราะห์และออกแบบระบบงาน Client/ Server ควรจะต้องเข้าใจ และพิจารณาคุณลักษณะ คุณสมบัติของเครื่องคอมพิวเตอร์เหล่านั้น ได้ตามความเหมาะสมของงาน (Application Software) โดยการพิจารณาจากคุณสมบัติ เช่น

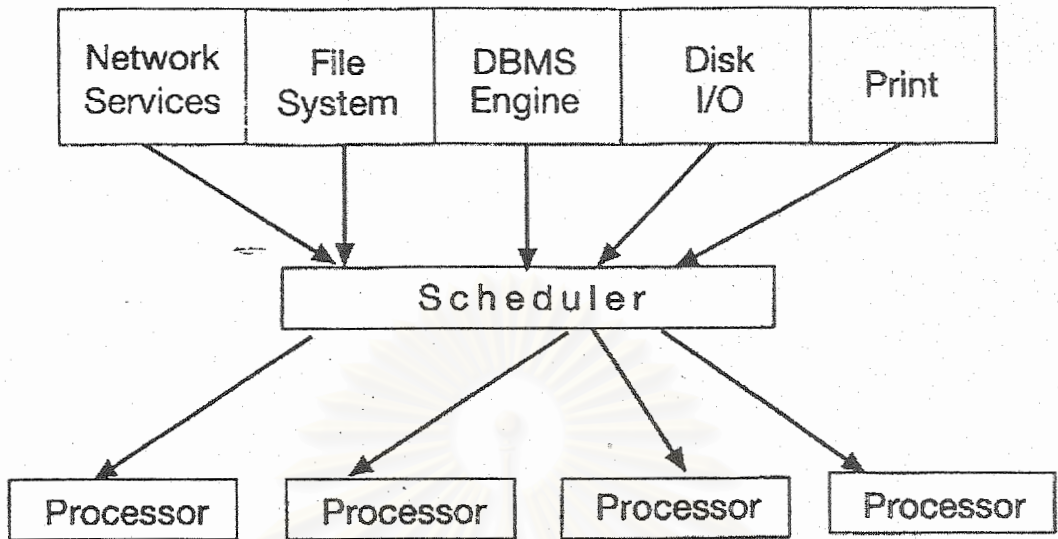
Multiprocessing

คือระบบเครื่องคอมพิวเตอร์ที่ออกแบบให้สามารถติดตั้ง Multiple Processor ได้ ซึ่งจะทำให้ server มีประสิทธิภาพของการประมวลผลสูงขึ้น มีลักษณะการทำงาน 2 แบบคือ

- Symmetric Multiprocessing

คือการประมวลผลของงานใดๆ ที่สามารถจะส่งไปประมวลผลที่โพรเซสเซอร์ใดก็ได้ที่มีความพร้อมมากกว่า โดยจะต้องมีโปรแกรมจาก Network Operating System Server คอยควบคุม สั่งการดังรูปที่ 7 เช่น Banyan's VINES and The Santa Cruz Operation's SCO UNIX เป็นต้น (บริษัท Microsoft Inc. ประกาศว่า WindowsNT มีลักษณะการทำงานเป็นแบบ Symmetric

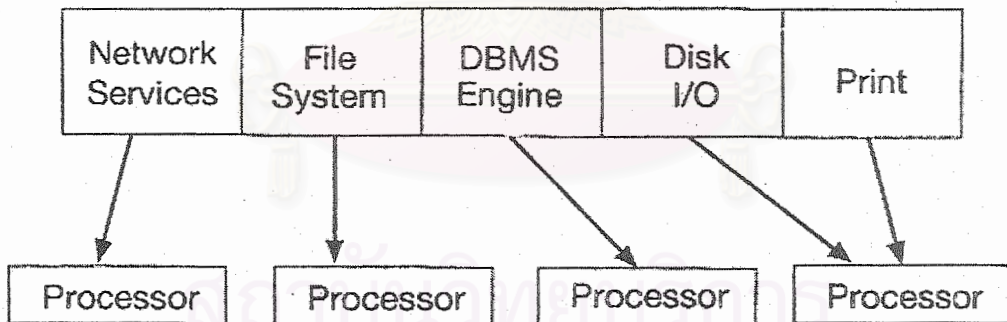
Multiprocessing นี้ด้วย เมื่อมกราคม 2540 บริษัท Novell Inc. ก็ประกาศว่า Netware มีการทำงานลักษณะนี้เช่นกัน)



รูปที่ 7 Symmetric Multiprocessing

- Functional Multiprocessing

คือการประมวลผลของงานที่ถูกกำหนดให้ส่งไปประมวลผลที่ Processor ตัวใดตัวหนึ่งอย่างถาวรแน่นอน ดังรูปที่ 8 เช่น Microsoft's LAN Manager เป็นต้น



รูปที่ 8 Functional Multiprocessing

6.1 Multithreading

Thread คือ หน่วยของงานที่เล็กที่สุดที่ระบบงานสามารถควบคุมให้ทำการประมวลผลได้ ในการทำงานแบบ Multithreading แต่ละ Process จะถูกแบ่งออกมาเป็นงานอิสระเล็กๆ ที่สามารถประมวลผลได้ภายในตัวเอง

6.2 Disk Arrays

คือ การใช้เทคนิคของการต่อเชื่อม Disk Storage (ฮาร์ดดิสก์หลายๆ ตัว) ให้สามารถทำงานในรูปแบบของ Fault Tolerant (Fault Tolerant มักจะหมายถึงระบบงานที่สามารถทำงานประมวลผลงานให้แก่ผู้ใช้ได้ตลอด ถึงแม้ว่าอุปกรณ์บางส่วนจะเสียหายไปก็ตาม โดยยึดหลักของการมีอุปกรณ์ที่สำคัญสำรองไว้อย่างน้อย 1 ชุด) ได้อย่างมีประสิทธิภาพ ราคาไม่แพง และสามารถทำงานได้อย่างรวดเร็ว ซึ่งเรียกว่า RAID (Redundant Arrays of Inexpensive Disk หรือ Redundant Arrays of Independent Disks) โดยใช้เทคนิคให้ข้อมูล 1 ชุดแตกตัวออกเป็นข้อมูลเล็กๆ หลายๆ ชุด บันทึกลงใน Disk Storage หลายๆ ตัว (ซึ่งทำให้บันทึกข้อมูลได้เร็ว) ในกรณีที่ Disk Storage ตัวใดตัวหนึ่งเสียไป Disk Storage ตัวที่เหลืออยู่ก็สามารถที่จะกู้กลับคืนข้อมูล หรือ สร้าง (Reconstruct) ข้อมูลทั้งชุดขึ้นมาใหม่ได้ (ทำให้ข้อมูลมีความปลอดภัยสูง)

ระบบการประมวลผลแบบ On-Line

(On-Line Transaction Processing System)

ความหมาย

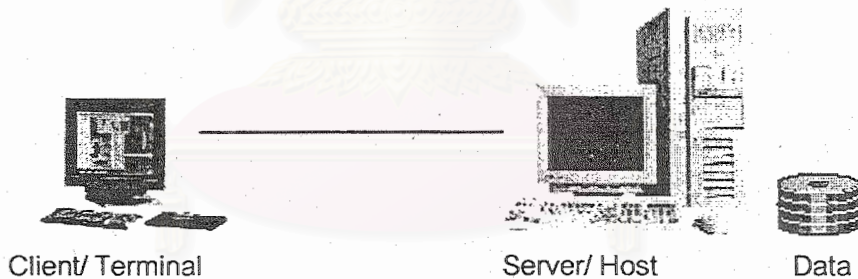
คือ ระบบการทำงานการประมวลผลงาน รับ-ส่งแลกเปลี่ยนข้อมูลข่าวสารระหว่างเครื่องคอมพิวเตอร์ (Client/ Terminal กับ Server/ Host) ในการประมวลผลทางธุรกรรมหนึ่ง

รูปแบบการประมวลผลงานแบบ On-Line

มีผู้ใช้คำว่า "On-Line" ในการออกแบบระบบงานคอมพิวเตอร์มากมาย บางท่านพูดสั้นๆ ว่า "ระบบงาน On-Line" บางท่านพูดโดยใช้ศัพท์ที่ใกล้เคียงกัน ซึ่งก่อให้เกิดความสับสนขึ้นในการสื่อความหมาย ความเข้าใจกันในรูปแบบของระบบงานประยุกต์ทางด้านคอมพิวเตอร์แก่ผู้อื่นๆ ที่เกี่ยวข้องจำนวนมาก ในที่นี้แบ่งรูปแบบการประมวลผลงานประยุกต์ออกเป็น 4 รูปแบบดังนี้

On-Line Interactive System

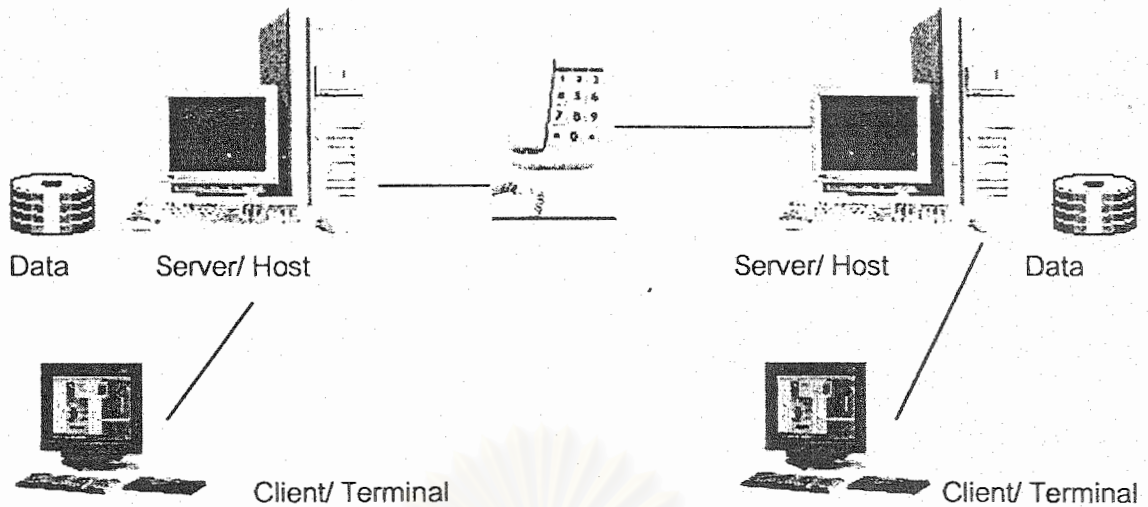
คือ ระบบงานที่ทำการประมวลผลโต้ตอบทันทีทันใดระหว่าง Client/ Terminal กับ Server/ Host (ส่วนใหญ่จะเน้นที่เป็นภายในเครือข่ายเดียวกัน) ดังรูปที่ 9



รูปที่ 9 On-Line Interactive System

6.2.1 Off-Line Transaction Processing System

คือ ระบบงานคอมพิวเตอร์ที่ทำการบันทึกข้อมูลที่ Client/ Terminal กับ Server/ Host ที่เครือข่ายใดเครือข่ายหนึ่งก่อน แล้วจึงส่งผ่านข้อมูลเหล่านั้นไป Update ที่ Server/ Host อีกเครือข่ายหนึ่ง ดังรูปที่ 10 หรือบางตำราอาจเรียกว่า Batch File Transfer

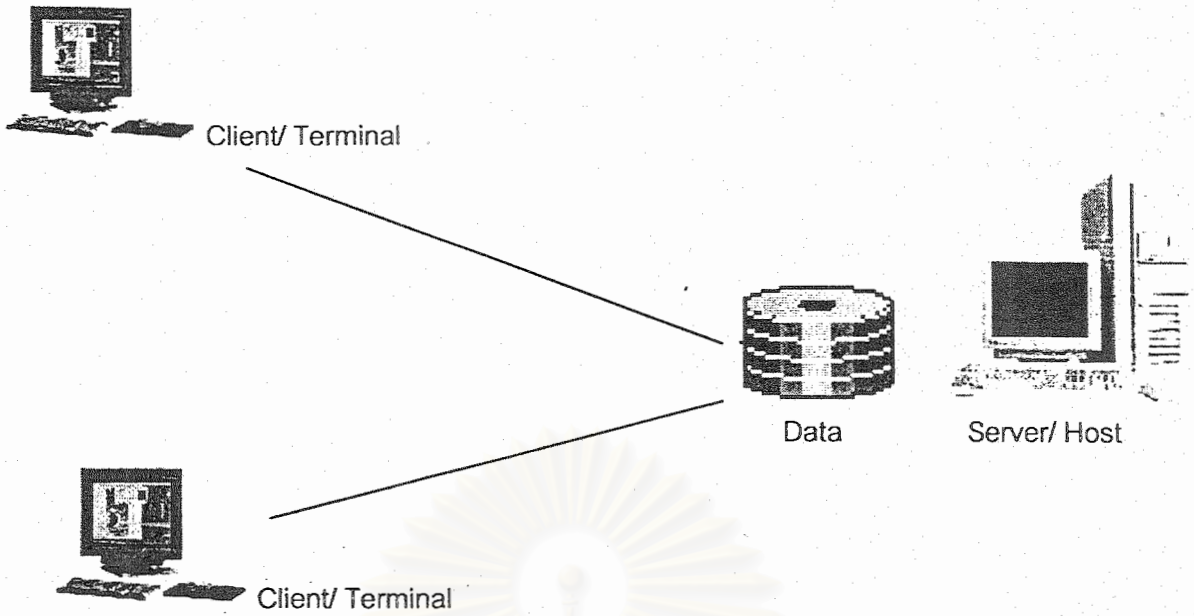


รูปที่ 10 Off-Line Transaction Processing System

เช่นระบบงานที่มีการแลกเปลี่ยนข้อมูลระหว่างห้องที่ที่อยู่ห่างไกลออกไป และสามารถทำงานอย่างอิสระได้ภายในเครือข่ายตัวเอง แต่จำเป็นต้องปรับปรุงข้อมูลกับเครือข่ายอื่นๆ เป็นต้น การลงทุนเครือข่ายในลักษณะนี้ ค่อนข้างถูก และเหมาะสมกับระบบงานที่ไม่จำเป็นต้องปรับปรุงข้อมูลทันทีทันใด โดยระบบงานฯ สามารถทำงานได้ครบกระบวนการในระบบคอมพิวเตอร์ของตัวเอง และจะให้การเชื่อมต่อเครือข่ายเป็นการชั่วคราวกับหน่วยงานอื่นที่เกี่ยวข้องเพื่อรับ-ส่ง แลกเปลี่ยนข้อมูล ข่าวสารระหว่างกัน เช่น ในหน่วยงานของราชการไทยจะมีหน่วยงานระดับอำเภอ จังหวัด เขต/ ภูมิภาค กรมฯ และกระทรวงฯ นักพัฒนาระบบงานฯ ต้องพิจารณาลักษณะของระบบงานฯ ว่ามีความจำเป็นต้องรับ-ส่ง ข้อมูลทันทีทันใดหรือไม่ อาจออกแบบให้ระบบงานฯ ทำงานในลักษณะ Standalone ที่อำเภอ และค่อยทำ Batch File Transfer ไปที่จังหวัด และจังหวัดอาจทำ Batch File Transfer ไปที่เขต/ ภูมิภาคอีกทอดหนึ่ง

6.2.2 Batch Processing System

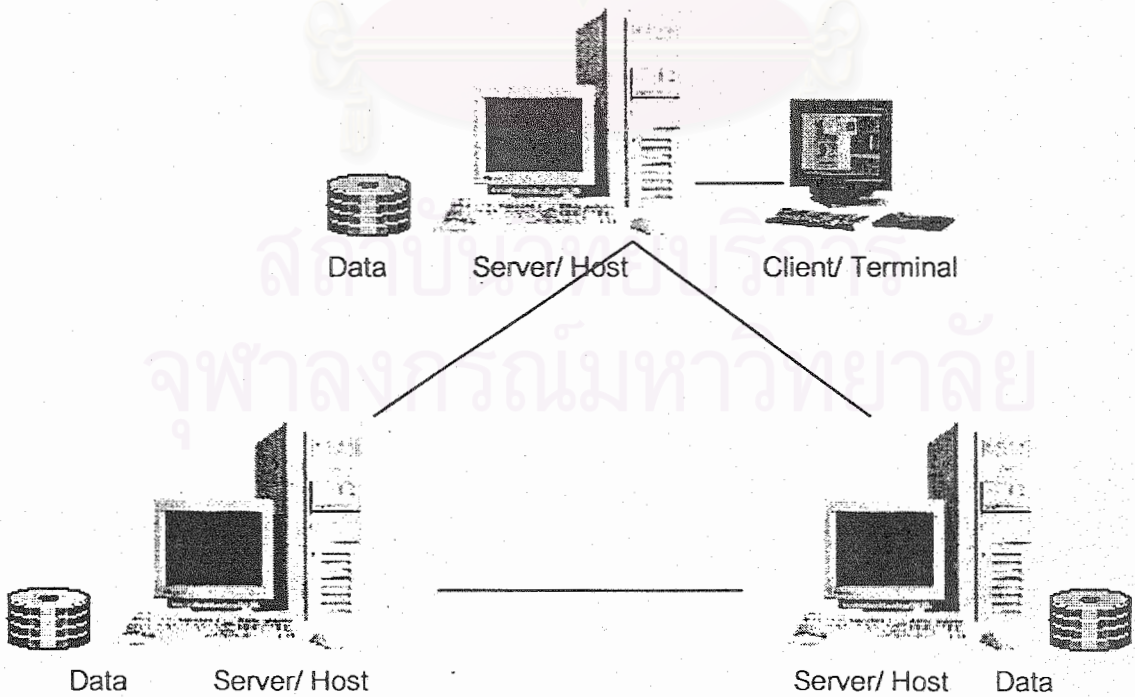
คือ ระบบงานที่บันทึกข้อมูล/ รายการ (Transaction) ไว้จำนวนหนึ่งก่อน แล้วจึงส่งผลข้อมูลเหล่านี้ไป Update ที่ระบบฐานข้อมูลอีกครั้งหนึ่ง (ส่วนใหญ่จะเป็นภายในเครือข่ายเดียวกัน ภายในเครื่องคอมพิวเตอร์เครื่องเดียวกัน หรือเป็นระบบงานเดียวกัน) ดังรูปที่ 11 เช่น ระบบงานบัญชี ที่อาจมีขั้นตอนการตรวจสอบข้อมูลให้ถูกต้อง หรือ Balance กันก่อน แล้วจึงจะ Update/ Post บัญชี ภายหลัง เป็นต้น



รูปที่ 11 Batch Processing System

6.2.3 On-Line Real Time Interactive System

คือ ระบบงานคอมพิวเตอร์ที่มีการประมวลผล และปรับปรุงข้อมูลทันทีทันใดให้แล้วเสร็จสมบูรณ์ที่ระบบฐานข้อมูล ไม่ว่ารายการข้อมูลนั้นจะถูกบันทึกที่ใดก็ตามในเครือข่ายเดียวกัน หรือต่างเครือข่ายก็ตาม ดังรูปที่ 12 เช่น ระบบงานเงินฝาก-ถอน แบบ ATM (Automatic Telling Machine) เป็นต้น



รูปที่ 12 On-Line Real Time Interactive System

Database Concept

ระบบฐานข้อมูล

เนื่องจากข้อมูลข่าวสารเป็นสิ่งที่มีความสำคัญต่อการดำเนินธุรกิจ ดังนั้นเทคนิคในการจัดเก็บข้อมูลจึงเป็นสิ่งสำคัญต่อการพัฒนาระบบ การออกแบบวิธีการเก็บข้อมูลอย่างมีประสิทธิภาพ จะทำให้การพัฒนาโปรแกรมง่ายและมีความรวดเร็วในการนำข้อมูลออกมาแสดงผล ซึ่งในปัจจุบันวิธีการจัดเก็บข้อมูลมี 2 แบบคือ

การจัดเก็บข้อมูลในรูปแบบแฟ้มข้อมูล

การจัดเก็บข้อมูลในรูปแบบนี้เป็นรูปแบบที่ใช้กันมานาน ตั้งแต่การพัฒนาระบบในยุคแรกๆ การจัดเก็บข้อมูลในลักษณะนี้มีข้อดี คือ สะดวก ใช้ทรัพยากรของระบบน้อย และไม่มีความซับซ้อนในการจัดเก็บ แต่มีปัญหาในการใช้งาน คือ

- ความซ้ำซ้อนของข้อมูล (Data Redundancy) - ในการจัดเก็บข้อมูลแบบแฟ้มข้อมูลอาจมีความจำเป็นที่จะต้องเก็บข้อมูลเดียวกันในหลายๆ แฟ้มข้อมูล ผู้ใช้แต่ละคนกำหนดข้อมูลตามที่ตนเองต้องการ
- ความไม่สอดคล้องกันของข้อมูล (Data Inconsistency) - ในกรณีที่มีการเก็บข้อมูลซ้ำซ้อนกันในหลายๆ แฟ้มข้อมูล การเปลี่ยนแปลงข้อมูลอาจทำให้ข้อมูลในแต่ละแฟ้มข้อมูลไม่เหมือนกันได้
- การใช้ข้อมูลร่วมกัน (Data Sharing) - การเก็บข้อมูลในรูปแบบแฟ้มข้อมูลจะไม่สามารถใช้ข้อมูลร่วมกันได้ ทุกคนที่ต้องการใช้ข้อมูลต้องมีสำเนา (copy) ของตนเอง
- การขึ้นต่อกันของข้อมูล (Data Dependency) - การจัดเก็บข้อมูลจะขึ้นกับโปรแกรมที่จะมา access ดังนั้นเมื่อเปลี่ยนโปรแกรมที่จะมา access ลักษณะข้อมูลก็ต้องเปลี่ยนแปลงตาม
- ความปลอดภัยของข้อมูล (Security) - การจัดเก็บในรูปแบบแฟ้มข้อมูลจะมีความปลอดภัยของข้อมูลต่ำ ไม่สามารถสร้าง view สำหรับผู้ใช้แต่ละคนเพื่อให้เลือกมองเพียงบางส่วนได้ ดังนั้นถ้าจำเป็นต้องมีการใช้ข้อมูลบางส่วนในแฟ้มข้อมูลนั้น แต่ไม่ต้องการให้เห็นข้อมูลในส่วนที่เหลือก็ต้องสร้างแฟ้มข้อมูลขึ้นมาใหม่เพื่อความปลอดภัย

การจัดเก็บข้อมูลในรูปแบบฐานข้อมูล

ฐานข้อมูล คือ ข้อมูลที่มีความสัมพันธ์กันถูกเก็บรวบรวมไว้เพื่อใช้โดย application ซึ่งข้อมูลที่จัดเก็บอาจเป็นตัวอักษร ภาพ หรือเสียง โดยมี Database Management System (DBMS) ทำหน้าที่ในการจัดการและติดต่อกับฐานข้อมูล โดยมีจุดประสงค์หลักเพื่อทำให้เกิดความสะดวกและควมมีประสิทธิภาพในการเรียกใช้และจัดเก็บข้อมูลลงในฐานข้อมูล

ข้อดีของการจัดเก็บข้อมูลในรูปแบบฐานข้อมูล

- ลดความซ้ำซ้อนของการจัดเก็บข้อมูล
- ลดความไม่สอดคล้องกันของข้อมูล
- สามารถใช้ข้อมูลร่วมกันได้
- มีมาตรฐานในการจัดเก็บข้อมูล
- สามารถค้นหาข้อมูลได้ง่าย
- มีความปลอดภัยสูง สามารถสร้าง view ให้กับผู้ใช้แต่ละคนให้เข้าถึงเฉพาะข้อมูลที่เขาจำเป็นต้องใช้เท่านั้น
- การจัดเก็บข้อมูลมีความอิสระจากโปรแกรมที่จะมา access

ข้อเสียของการจัดเก็บข้อมูลในรูปแบบฐานข้อมูล

- ใช้ทรัพยากรของระบบมาก ทั้งฮาร์ดแวร์และซอฟต์แวร์
- ค่าใช้จ่ายสูง
- มีความซับซ้อน เพราะต้องคำนึงถึงหลายแง่ในการใช้ข้อมูล
- ความเร็วในการเข้าถึงข้อมูลขึ้นอยู่กับวิธีการออกแบบ ถ้าออกแบบฐานข้อมูลอย่างไม่มีประสิทธิภาพ การเข้าถึงข้อมูลจะใช้เวลาานาน

ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System)

ฐานข้อมูลเชิงสัมพันธ์ เป็นฐานข้อมูลรูปแบบหนึ่งที่ยอมรับในปัจจุบัน เนื่องจากเป็นฐานข้อมูลที่เข้าใจง่าย ผู้ใช้ทั่วไปสามารถมองภาพของข้อมูลที่ถูกเก็บได้ง่ายเพราะข้อมูลถูกจัดเก็บในลักษณะของตารางสองมิติ คือเป็นแถว (Row) และคอลัมน์ (Column) เป็นรูปแบบฐานข้อมูลที่ไม่ซับซ้อนและมีเครื่องมือช่วยในการเรียกใช้ข้อมูลโดยใช้คำสั่งง่ายๆ เช่น SQL เป็นต้น ด้วยเหตุนี้ ระบบจัดการฐานข้อมูล (Database Management Systems: DBMS) ที่นิยมใช้กันอย่างแพร่หลายในปัจจุบันก็เป็นระบบที่สนับสนุนฐานข้อมูลประเภทนี้ค่อนข้างมาก เช่น DB2, Oracle, Sybase, Ingress เป็นต้น

คำศัพท์ที่เกี่ยวข้อง

- ตาราง (Table) หรือ ความสัมพันธ์ (Relation) เป็นคำที่ใช้แทนเรื่องใดเรื่องหนึ่ง เช่น ตารางข้อมูลพนักงาน, ตารางข้อมูลแผนก, ตารางข้อมูลโครงการ, ตารางข้อมูลงานที่รับมอบหมาย โดยตารางมีคุณสมบัติดังต่อไปนี้
 - ไม่มีแถวที่ซ้ำกันใน 1 relation
 - แถวไม่จำเป็นต้องเรียงลำดับกัน
 - คอลัมน์ไม่จำเป็นต้องเรียงลำดับกัน

➤ ค่าของคอลัมน์ใด ๆ จะเป็นค่าที่แยกไม่ได้

- คอลัมน์ (Column) หรือแอททริบิวต์ (Attribute) คือรายละเอียดของข้อมูลในแต่ละตารางที่แสดงให้เห็นว่าตารางหนึ่งๆ ประกอบไปด้วยรายละเอียดอะไรบ้าง
- แถว (Row) หรือ ทูเพิล (Tuple) คือ ค่าข้อมูลแต่ละแถวในแต่ละตาราง
- โดเมน (Domain) คือ ขอบเขตหรือค่าที่ควรจะเป็นของข้อมูลในคอลัมน์หนึ่งๆ
- คีย์ แบ่งออกเป็น คีย์หลัก และ คีย์นอก ดังรายละเอียดในหัวข้อ 2.2
- เอนทิตี (entity) คือการรวบรวมสิ่งที่เกี่ยวข้องมาเก็บเป็นหมวดหมู่เดียวกันอย่างเช่นในการเก็บประวัตินักศึกษา และสถานที่ศึกษา ก็จะประกอบไปด้วย 2 entity คือ นักศึกษา กับ สถานที่ศึกษา จนมีบางคนสรุปว่า entity คือค่านามในประโยคนั้นเอง

ประเภทของคีย์

เค้าร่างของตาราง (Table/ Relation Schema) ประกอบด้วยรายละเอียดข้อมูลของคอลัมน์ต่างๆ ซึ่งคอลัมน์ใดคอลัมน์หนึ่งในตารางมีโอกาสที่จะมีคุณสมบัติเป็นคีย์ โดยคีย์นั้นๆ สามารถใช้ในการแสดงหรือบ่งบอกถึงค่าของข้อมูลที่เหลือในแต่ละแถว หรือใช้ในการเชื่อมโยงข้อมูลกับข้อมูลกับอีกตารางหนึ่ง ประเภทของคีย์ที่จะกล่าวในที่นี้ แบ่งออกเป็น 2 ประเภทหลัก คือ

- คีย์หลัก (Primary Key)

เป็น Attribute ที่มีคุณสมบัติของข้อมูลที่เป็นค่าเอกลักษณ์หรือมีค่าที่ไม่ซ้ำกัน คุณสมบัติดังกล่าวจะสามารถระบุว่าข้อมูลนั้นเป็นข้อมูลของ Tuple ใด เช่น รหัสพนักงาน เลขที่ 3001 สามารถระบุได้ว่าเป็นของพนักงานชื่อ ดวงพร Attribute ที่มีคุณสมบัติเป็น คีย์หลักอาจประกอบด้วยหลาย Attribute รวมกันเรียกว่า Composite Key (คีย์ผสม) นอกจากนี้ ใน Relation หนึ่งๆ อาจมี Attribute ที่มีคุณสมบัติเป็นคีย์หลักได้มากกว่า หนึ่ง Attribute เรียก Attribute เหล่านี้ว่า Candidate Key (คีย์คู่แข่ง) ถ้า Attribute หนึ่งถูกกำหนดให้เป็นคีย์หลัก อีก Attribute หนึ่งที่มีคุณสมบัติเป็นคีย์หลัก แต่ไม่ได้ถูกเลือกให้เป็นคีย์หลักจะเรียกว่า คีย์สำรอง (Alternate Key)

- คีย์นอก (Foreign Key)

เป็น Attribute ใน Relation หนึ่งที่ใช้อ้างอิงถึง Attribute เดียวกันนี้ในอีก Relation หนึ่ง โดยที่ Attribute นี้มีคุณสมบัติ เป็นคีย์หลักใน Relation ที่ถูกอ้างอิง การมี Attribute นี้ปรากฏอยู่ใน Relation ทั้งสองก็เพื่อประโยชน์ ในการเชื่อมโยงข้อมูล

กฎที่ใช้กับฐานข้อมูลเชิงสัมพันธ์

เนื่องจากตารางต่างๆ ในฐานข้อมูลเชิงสัมพันธ์จะอ้างอิงความสัมพันธ์โดยใช้คีย์ กฎที่เกี่ยวข้องกับคีย์ในฐานข้อมูลเชิงสัมพันธ์มีดังต่อไปนี้

- กฎความบูรณาภาพของเอนทิตี (The Entity Integrity Rule)

กฎนี้ระบุว่าแอททริบิวต์ใดที่เป็นคีย์หลัก ข้อมูลในแอททริบิวต์นั้นจะเป็นค่าว่าง(Null) ไม่ได้ ความหมายของการเป็นค่าว่างไม่ได้(Not Null) หมายความว่า ข้อมูลของแอททริบิวต์ที่เป็นคีย์หลักจะไม่ทราบค่าที่แน่นอนหรือไม่มีค่าไม่ได้

- กฎความบูรณาภาพของการอ้างอิง (The Referential Integrity Rule)

การอ้างอิงข้อมูลระหว่างรีเลชันในฐานข้อมูลเชิงสัมพันธ์จะใช้คีย์นอกของรีเลชันหนึ่งไปตรวจสอบกับค่าของแอททริบิวต์ที่เป็นคีย์หลักของอีกรีเลชันหนึ่งเพื่อเรียกดูข้อมูลอื่นๆที่เกี่ยวข้องหรือค่าของคีย์นอกจะต้องอ้างอิงให้ตรงกับค่าของคีย์หลักได้จึงจะสามารถเชื่อมโยงข้อมูลระหว่างสองรีเลชันได้สำหรับคีย์นอกจะมีค่าว่างได้หรือไม่ขึ้นอยู่กับกฎเกณฑ์การออกแบบฐานข้อมูล เช่น ในกรณีที่รีเลชันพนักงานมี Depno เป็นคีย์นอกอาจจะถูกระบุว่าต้องทราบค่าแต่ในกรณีพนักงานทดลองงานอาจยังไม่มีค่า Depno เพราะยังไม่ได้ถูกรับรองในกรณีที่มีการลบหรือแก้ไขข้อมูลของแอททริบิวต์ที่เป็นคีย์หลักซึ่งมีคีย์นอก จากอีกรีเลชันหนึ่งอ้างอิงถึง จะทำการลบหรือแก้ไขข้อมูลได้หรือไม่ ขึ้นอยู่กับการออกแบบฐานข้อมูล ว่าได้ระบุให้แอททริบิวต์มีคุณสมบัติอย่างไร ซึ่งมีโอกาสเป็นไปได้ 4 ทางเลือก

- การลบหรือแก้ไขข้อมูลแบบมีข้อจำกัด (Restrict) การลบหรือแก้ไขข้อมูลจะกระทำไม่ได้ เมื่อข้อมูลของคีย์หลักในรีเลชันหนึ่งไม่มีข้อมูลที่ถูกอ้างอิง โดยคีย์นอกของอีกรีเลชันหนึ่งเช่น รหัสแผนก Depno ในรีเลชันDep จะถูกแก้ไขหรือลบทิ้งต่อเมื่อไม่มีพนักงานคนใดสังกัดอยู่ในแผนกนั้น
- การลบหรือแก้ไขข้อมูลแบบต่อเรียง (Cascade) การลบหรือการแก้ไขข้อมูล จะทำแบบเป็นลูกโซ่ คือ หากมีการแก้ไขหรือลบข้อมูลของคีย์หลักในรีเลชันหนึ่งระบบจะทำการลบหรือแก้ไขข้อมูลของคีย์นอกในอีกรีเลชันหนึ่งที่อ้างอิงถึงข้อมูลของคีย์หลักที่ถูกลบให้ด้วย เช่น ในกรณีที่ยกเลิกแผนก 9 ใน Entity แผนก ข้อมูลของพนักงานที่อยู่แผนก 10 ใน Entity พนักงานจะถูกลบออกไปด้วย

- การลบหรือแก้ไขข้อมูลโดยเปลี่ยนเป็นค่าว่าง (Nullify) การลบหรือแก้ไขข้อมูลจะทำได้เมื่อมีการเปลี่ยนค่าของคีย์นอกในข้อมูลที่ถูกอ้างอิงให้เป็นค่าว่างเสียก่อน เช่น พนักงานที่อยู่ในแผนกที่ 9 จะถูกเปลี่ยนค่าเป็นค่าว่างก่อนหลังจากนั้น การลบข้อมูลของแผนกที่มีรหัส 9 จะถูกลบทิ้งหรือแก้ไขทันที ภายใน Entity แผนก
- การลบหรือแก้ไขข้อมูลแบบใช้ค่าโดยปริยาย (Default) การลบหรือแก้ไขข้อมูลของคีย์หลัก สามารถทำได้โดยถ้าหากมีคีย์นอกที่อ้างอิงถึงคีย์หลักที่ถูกลบหรือแก้ไข ก็จะทำให้การปรับค่าของคีย์นอกนั้นโดยปริยาย (Default Value) ที่ถูกกำหนดขึ้นเช่น ในกรณีที่ยกเลิกแผนก 9 ใน Entity แผนก ข้อมูลของพนักงานที่อยู่ในแผนก 9 ใน Entity พนักงานจะถูกเปลี่ยนค่าเป็น 00 ซึ่งเป็นค่าโดยปริยาย ที่หมายความว่า ไม่ได้สังกัดแผนกใด

ภาษาทางด้านฐานข้อมูล (Query Language) ผลิตภัณฑ์ทางด้านฐานข้อมูลที่มีโครงสร้างข้อมูลแบบ Relation (Rela-Lation Model) จำเป็นอย่างยิ่งที่จะต้องมีการภาษาทางด้านฐานข้อมูล เช่น ภาษา SQL (Structure Query Language) ภาษา QBE (Qurey by Exaample) และภาษา Quel ฯลฯ เป็นต้น ภาษาเหล่านี้ได้ถูกพัฒนาขึ้นจากแนวคิดที่ต่างกัน เช่น ภาษา QBE ซึ่งถูกพัฒนาขึ้นจากแนวคิดของ Relational Calculus ส่วนภาษา Quel ถูกพัฒนาขึ้นจากแนวคิดของ Tuple Relational Calculus และ Relational Algebra เป็นหลักแต่อย่างไรก็ตาม ภาษาที่ได้รับความนิยมมากที่สุดคือภาษา "SQL"

Structured Qurey Language ได้ เริ่มต้นพัฒนาครั้งแรกโดย San Jose Research Laboratory (ปัจจุบันเปลี่ยนชื่อเป็น Almaden Res-earch Center) ของบริษัท IBM โดยมีชื่อแรกเริ่มว่า "Sequal" ซึ่งเป็นงานวิจัยในโครงการ R ในต้นทศวรรษ 1970 และได้ถูกนำมาใช้เป็นต้นแบบของภาษา SQL ของผลิตภัณฑ์ทางด้านฐานข้อมูลจำนวนมาก แต่อย่างไรก็ตาม ภาษา SQL ของแต่ละผลิตภัณฑ์ยังคงมีชื่อแตกต่างกันในรายละเอียดทางการใช้งาน ดังนั้นในปี ค.ศ.1986 ทางด้าน American National Standards Institute (ANSI) จึงได้กำหนดมาตรฐานของ SQL ขึ้นรวมทั้งบริษัท IBM ได้กำหนดมาตรฐานของตัวเองขึ้นมาเช่นเดียวกันโดยมีชื่อว่า Systems Application Architecture Database Interface (SAA-SQL) ซึ่งต่อมาทั้ง 2 มาตรฐานนี้ได้เป็นมาตรฐานในการผลิตภาษา SQL ของแต่ละบริษัท ส่วนคำสั่งต่างๆ ของภาษา SQL สามารถแบ่งตามลักษณะการใช้งานออกเป็น 3 กลุ่ม ดังนี้

DDL (Data Definition Language)

เป็นกลุ่มคำสั่งที่ใช้สำหรับสร้างฐานข้อมูล หรือใช้กำหนดโครงสร้างให้กับ Relation ภายในฐานข้อมูลเช่น การเพิ่ม เปลี่ยนแปลง ลบ Attribute ของ Relation ฯลฯ เป็นต้น

DML (Data Manipulation Language)

เป็นกลุ่มคำสั่งที่พัฒนาขึ้นตามแนวคิดของ Relational Algebra และ Record Relational Calculus โดยประกอบด้วยคำสั่งที่ใช้สำหรับ เพิ่ม ลบ หรือเปลี่ยนแปลงข้อมูลในฐานข้อมูล

Data Query Language

เป็นกลุ่มคำสั่ง DML ประเภทหนึ่ง ที่ใช้ในการเลือกข้อมูลจาก Relation ขึ้นมาแสดงผล ตามรูปแบบที่ต้องการ



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ขั้นตอนการทำการออกแบบฐานข้อมูล

1. ทำความเข้าใจระบบงานให้ชัดเจน

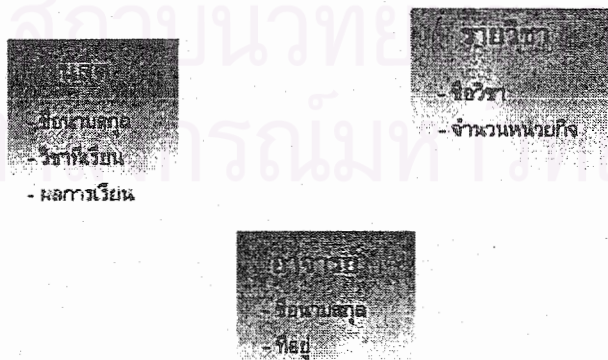
ขั้นตอนนี้สำคัญมากที่สุดขั้นตอนหนึ่ง ที่จะต้องมองให้ชัดเจนทั้งในภาพรวมและภาพย่อย ทั้งของกลุ่มข้อมูลที่มีและไม่มีความสัมพันธ์

2. ร่างความสัมพันธ์คร่าว ๆ โดยแบ่งออกเป็นกลุ่ม ๆ

การกำหนด Entity ของเรื่องที่เราศึกษาออกเป็นกลุ่ม ตัวอย่างเช่น ถ้าเราต้องการทำการออกแบบฐานข้อมูลของฝ่ายทะเบียนนิสิต โดยมีข้อมูลดังต่อไปนี้

- ชื่อนามสกุลนิสิต
- ภาควิชาของนิสิตแต่ละคน
- อาจารย์ที่ปรึกษา
- ชื่อวิชา
- ที่อยู่นิสิต
- เบอร์โทรศัพท์อาจารย์ที่ปรึกษา
- จำนวนหน่วยกิตของนิสิต
- ผลการเรียนของนิสิตแต่ละวิชา

จากข้อมูลทั้งหมดที่กล่าวมาจะสามารถแบ่งออกเป็นกลุ่มได้ดังต่อไปนี้



รูปที่ 13 แสดงให้เห็นการแบ่งกลุ่มข้อมูล

โดยแต่ละกลุ่มข้อมูลเราจะมีศัพท์เฉพาะเรียกว่า Entity นั้นเอง

3. กำหนดความสัมพันธ์และคีย์หลักของข้อมูล

หลังจากที่ทำการกำหนด Entity ได้แล้วลำดับต่อมาให้ทำการใส่ฟิลด์ที่ต้องการให้ครบ โดยแยกในแต่ละกลุ่ม แล้วทำการกำหนดความสัมพันธ์ ระหว่างกลุ่ม อย่างเช่น ความสัมพันธ์ระหว่างอาจารย์กับรายวิชาคือ หนึ่งวิชามีอาจารย์สอนได้หนึ่งคน แต่อาจารย์หนึ่งคนสอนได้หลายวิชา ดังนั้นจึงจำเป็นต้องเพิ่มฟิลด์ ชื่ออาจารย์ไว้ใน Entity รายวิชาด้วย และวิชาที่เรียนใน Entity นิสิต ก็也将มีความสัมพันธ์กับชื่อวิชาใน Entity รายวิชา และทำการกำหนดคีย์หลักให้กับแต่ละ Entity โดยคีย์หลักจะเป็นข้อมูลที่เมื่อเราพูดถึงฟิลด์นั้นแล้วเราจะทราบทันทีว่าหมายถึงข้อมูลในแถวไหน ดังนั้นถ้าพิจารณาจาก Entity นิสิต เราจะพบว่า ชื่อและนามสกุลสามารถเป็นคีย์หลักของ Entity นี้ได้ก็จริง แต่บางครั้งก็มีโอกาสที่จะมีชื่อหรือนามสกุลซ้ำกันได้เช่นกัน และโดยปกติทั่วไปแล้ว เมื่อเราได้มีโอกาสไปเขียนโปรแกรมเกี่ยวกับฐานข้อมูลแล้วเกิดต้องมีการเรียกข้อมูลหลาย ๆ ตารางพร้อมกันแล้ว การเลือกคีย์หลักเป็นฟิลด์ที่มีลักษณะเหมือน Description จะไม่ค่อยดีนัก เพราะจะทำให้ประสิทธิภาพในการค้นหาข้อมูล และการเชื่อมต่อระหว่างข้อมูลไม่ดี ดังนั้นจึงมักจะมีการกำหนดรหัสไว้ในแต่ละตาราง ตัวอย่างเช่น Entity นิสิตก็จะมีรหัสนิสิตเป็นคีย์หลัก Entity รายวิชา ก็จะมีรหัสวิชาเป็นคีย์หลัก และ Entity อาจารย์ก็จะมีรหัสอาจารย์เป็นคีย์หลักในกรณีที่อาจารย์หนึ่งคนสอนหนึ่งวิชา เป็นต้น

ประเภทของความสัมพันธ์จะสามารถแบ่งออกได้เป็น 3 ประเภท ดังต่อไปนี้คือ

- ความสัมพันธ์แบบ One-to-One หมายถึงว่าจากข้อมูลจากตาราง A มีความเกี่ยวข้องกับตาราง B แค่แถวเดียว
- ความสัมพันธ์แบบ Many-to-One หรือ One-to-Many หมายถึงว่าข้อมูลจากตาราง A มีความเกี่ยวข้องกับตาราง B หลายแถว หรือ ข้อมูลจากตาราง B หลายแถวมีความเกี่ยวข้องกับตาราง A แถวเดียว
- ความสัมพันธ์แบบ Many-to-Many หมายถึงว่าข้อมูลจากตาราง A หลายแถวเกี่ยวข้องกับข้อมูลจากตาราง B หลายแถว ซึ่งความสัมพันธ์แบบนี้จะมีปัญหาที่สุด ดังนั้นจึงควรหลีกเลี่ยง

รูปแบบบรรทัดฐาน (Normalization)

แนวคิดในการทำรีเลชันให้อยู่ในรูปแบบบรรทัดฐานถูกคิดค้นโดย อี.เอฟ.คอตต์เป็นกระบวนการที่นำเค้าร่างของรีเลชันมาทำให้อยู่ในรูปแบบบรรทัดฐาน(normal form)

วัตถุประสงค์ของการทำให้เป็นบรรทัดฐาน มีดังนี้

1. เพื่อลดเนื้อหาในการเก็บข้อมูลการทำให้เป็นบรรทัดฐานเป็นการลดความซ้ำซ้อนของข้อมูลในรีเลชัน
2. เพื่อลดปัญหาที่ข้อมูลไม่ถูกต้อง(Inconsistency)เนื่องจากข้อมูลในรีเลชันหนึ่งจะมีข้อมูลไม่ซ้ำกันเมื่อมีการปรับปรุงข้อมูลก็จะปรับปรุงทุกฟิลด์นั้นๆครั้งเดียวไม่ต้องปรับปรุงหลายแห่ง
3. เป็นการลดปัญหาที่เกิดจากการเพิ่ม ปรับปรุงและลบข้อมูล

ข้อมูลที่ซ้ำซ้อนกันนี้จะก่อให้เกิดปัญหาขึ้นอย่างน้อย 2 ประการ คือ

1. ปัญหาความผิดพลาดของข้อมูล เช่น การที่ลูกค้าเปลี่ยนชื่อหรือในกรณีที่บริษัทมีการเปลี่ยนแปลงรายละเอียดสินค้า เราจะต้องทำการแก้ไขข้อมูลให้ครบทุกเรคคอร์ดในตาราง มิฉะนั้นข้อมูลในบางเรคคอร์ดจะเกิดความผิดพลาดได้
2. เปลืองเนื้อที่ในการจัดเก็บข้อมูลเพราะจะต้องจัดเก็บข้อมูลเดียวกันไว้ในหลาย ๆ เรคคอร์ด

หลักการทำ Normalization

หลักการทำ Normalization ที่สำคัญคือ "การลดความซ้ำซ้อนและโอกาสที่จะเกิดความผิดพลาดกับข้อมูลได้" ซึ่ง การที่จะทำให้บรรลุจุดประสงค์ดังกล่าวจะต้องมีเกณฑ์และขั้นตอนในการวิเคราะห์ข้อมูลโดยทั่วไปเราต้องรู้ก่อนว่าแต่ละตารางมี field ไหนบ้างสามารถบ่งชี้หรือค้นหาข้อมูลได้ เช่น เมื่อทราบรหัสลูกค้า จะ ทำให้สามารถค้นหา ชื่อ,นามสกุล,ที่อยู่ ฯลฯ ได้ สำหรับเกณฑ์เหล่านี้เราจะเรียกว่า "Functional Dependency" (FD) ให้สัญลักษณ์ \rightarrow แทนการกำหนดค่าระหว่าง field คุณสมบัติที่สำคัญอีกประการของการทำ Normalization คือ เมื่อตารางใดจัดอยู่ใน Normal Form ใด แล้วจะต้องมีคุณสมบัติของ Normal Form ที่ต่ำกว่าเสมอ เช่น ถ้าตารางใดเป็น 3NF จะต้องมีความสัมพันธ์ของ 1NF และ 2NF อยู่ด้วย ในที่นี้จะไม่กล่าวถึง ระดับที่เหลืออยู่คือ BCNF, 4NF, และ 5NF

1NF(First Normal Form)

ตารางใดจะถือว่าอยู่ใน 1 NF หรือไม่จะพิจารณาทุก Fields ในตารางว่าจะต้อง "ไม่มี Fields ใด Fields หนึ่งที่ลักษณะเป็น Multivalued (Field เดียวแต่เก็บหลายๆ ค่าไว้ด้วยกัน) เช่นตารางที่ 4

CID	Brand	Model	Color
1ข-2776	Benz	E220	ขาว, เหลือง
9ค-8412	Misubishi	Lacer	แดง, ขาว,
2ด-5266	Toyota	Corolla	เหลือง, เทา

ตารางที่ 4 แสดงตารางตัวอย่างที่มี Multivalued

จากตัวอย่างดังกล่าวจะเห็นได้ว่า Field "Color" จะเป็นลักษณะของ Multivalued ซึ่งให้แก้ไขโดยแบ่ง Field ดังกล่าวออกเป็นอีกตาราง และดึงเอา Primary Key ของตารางนั้นมาด้วยดังตารางที่ 5

CID	Brand	Model
1ข-2776	Benz	E220
9ค-8412	Misubishi	Lacer
2ด-5266	Toyota	Corolla

CID	Color
1ข-2776	ขาว
1ข-2776	เหลือง
9ค-8412	แดง
2ด-5266	เทา
9ค-8412	ขาว
2ด-5266	เหลือง

ตารางที่ 5 แสดงการแบ่งตาราง ที่ 4 ออกเป็นสองตารางย่อย ที่ไม่มี Multivalued

2NF(Second Normal Form)

การทำ 2 NF เน้นการวิเคราะห์ฟิลด์ที่เป็น Primary Key โดยปรกติแล้ว Primary Key ของตารางหนึ่งๆ อาจประกอบด้วย Field เพียง Field เดียวหรือหลาย Fields รวมอยู่ด้วยกันก็ได้โดยมีหลักว่า "หากมี

ตารางใดที่ประกอบด้วย Field เพียง Field เดียวจะถือว่าตารางนั้นอยู่ใน 2NF" ส่วนตารางใดที่มีหลาย

ตารางรายการขาย (InvoiceDetail Table)

InvNo	InvDt	Pid	Pname	Brand	Model	Qu	Cid	Title	Fname	Lname	Tel
101	10/7/00	M011	หม้อหุงข้าว	Hitachi	CX98	20	C01	นาย สุรสิทธิ์	รักล้น		522-1622
101	10/7/00	F002	ตู้เย็น	Whirlpool	AB77	5	C01	นาย สุรสิทธิ์	รักล้น		522-1622
101	10/7/00	F003	ตู้เย็น	National	CC87	10	C01	นาย สุรสิทธิ์	รักล้น		522-1622
102	11/7/00	F002	ตู้เย็น	Whirlpool	AB77	15	C03	นาย สมศักดิ์	มักรวย		121-1111
102	11/7/00	F005	ตู้เย็น	Whirlpool	CC87	30	C03	นาย สมศักดิ์	มักรวย		121-1111
103	11/7/00	M011	หม้อหุงข้าว	Hitachi	CX98	2	C01	นาย สุรสิทธิ์	ทองมาก		252-1111
104	12/7/00	F002	ตู้เย็น	Whirlpool	AB77	5	C02	นส. การุณ	ทองสุข		254-1545

Fields รวมกันเป็น Primary Key ให้แยกเอา Fields นั้นไปสร้างตารางใหม่ ดังตารางที่ 6

ตารางที่ 6 ตารางตัวอย่างภายหลังจากการทำ 2NF

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

3NF (Third Normal Form)

หลักการทำให้ 3NF คือ การที่จะต้องมีฟิลด์ใดในตาราง นอกจาก Primary Key ที่สามารถไปกำหนดฟิลด์อื่น ดังตัวอย่างในขั้นตอน 2NF จะเห็นได้ว่ารหัสลูกค้า (Cid) ไม่ได้เป็น Primary Key ของตารางแต่สามารถไปกำหนด ชื่อ,นามสกุล,ค่านำหน้า และเบอร์โทรศัพท์ ดังนั้นตารางรายการการขาย (Invoice Detail Table) จึงไม่อยู่ในรูป 3NF

วิธีแก้ไข คือ แยกเอาฟิลด์ต่างๆ ที่ถูกกำหนดโดยฟิลด์ที่ไม่ใช่ Primary Key นั้นออกมาอีกตารางหนึ่งและตั้งเอาฟิลด์เป็นตัวกำหนดนั้นมาเป็นส่วนร่วมในตารางใหม่ด้วย ดังนั้นฟิลด์ Title, Fname, Lname, Tel, Cid จึงถูกแยกออกมาเป็นตารางลูกค้า (Customer Table) ดังตารางที่ 7

ตารางรายการขาย (Invoice Detail Table)

Pid	Pname	Brand	Model	Price
M011	หม้อหุงข้าว	Hitachi	CX98	2500
F002	ตู้เย็น	Whirlpool	AB77	8000
F003	ตู้เย็น	National	CC87	6800

ตารางใบกำกับสินค้า (Invoice Table)

InvNo	InvDt	Cid
101	10/7/00	C01
102	11/7/00	C03
103	11/7/00	C01
104	12/7/00	C02

ตารางรายการขาย (Invoice Detail Table)

InvNo	Pid	Qu
101	M011	20
101	F002	5
101	F003	10
102	F002	15
102	F005	30
103	M011	2
104	F002	5

ตารางรายการขาย (Invoice Detail Table)

Cid	Title	Fname	Lname	Tel
C01	นาย	สุรสิทธิ์	รักล้น	522-1622
C03	นาย	สมศักดิ์	มักรอย	435-7207
C01	นาย	สุรสิทธิ์	ทองมาก	252-1111
C02	นส.	การุณ	ทองสุข	254-1545

ตารางที่ 7 ตารางตัวอย่างภายหลังจากการทำ 3NF

Quick Command Directory for DOS, Windows/NT & UNIX

	DOS, NT	UNIX
File Handling Commands		
List the files in a directory	dir	ls
Copy a file	copy, xcopy	cp
Delete a file	del (erase)	rm
Rename or move a file	rename, move	mv
Display a file on the screen	type	cat
Display a filetype with pauses between screen fills	fnmore	more, less
Print a file	print	lp
Edit or create a text file	edit, edlin	pico
Backup your files	backup	tar, cpio
Compare two files	comp, fc	diff
Find text in a file	find	grep
Sort a file	sort	sort
Retrieve a deleted file	undelete	
Allow others to see/use a file(s)	Disk: Share as...(1)	chmod
Directory Handling Commands		
	DOS, NT	UNIX
Change directories	chdir (cd)	cd
Display current directories	chdir (cd)	pwd
Make a new directory	mkdir (md)	mkdir
Remove a directory	rmdir (rd)	rmdir
Utilities		
	DOS, NT	UNIX
Change password	Change Password (2)	chpw
Check remaining disk space	chkdisk	quota -v
Clear screen	cls	clear
Close a Window	exit	exit
Check system date	date, time	date
Log off of the computer	File: Logoff(2)	exit
Format a new disk(ette)	format	initialize
File Transfer Protocol	ftp	ftp
Display online manual	help	man, man utcc
Send or receive mail	mail(1)	pine
Check amount of memory available	mem	vmstat
Set terminal type	mode	TERM=type
Recall previous commands	press the [up arrow]	history
Define an alias		alias
Capture a session of commands /responses		script
Send a file		
Set terminal features		stty
Running Applications and Commands		
	DOS, NT	UNIX
Automate a series of commands	fn .bat	.profile, .kshrc
Run a system command interactively	cmd	cmd
Run a system command as a subprocess	start, cmd low(2)	cmd &

Run a system command in batch	start, cmd low(2)	batch
Set search path to locate commands	path	PATH=
Show which programs are executing	[Ctrl][Esc](3)	ps
Stop an executing program	End Task(3)	kill
Creating and Running Programs	DOS, NT	UNIX
Compile a C program	Depends on version	cc
Compile a FORTRAN program	Depends on version	f77
Link or load object modules		ld
Run a compiled program in the background		a.out &
Run a compiled program interactively		a.out
Submit a command at a specific time		at, cron

Usage Notes:

- If a command has two names, they are separated by parentheses. Example: del (erase).
- If two commands perform similar functions, they are separated by a comma. Example: edit.
- Items separated by colons represent menu choices. For example, "File: Share as..." means choose "Share as..." from the "File" menu.
- Keyboard keys are enclosed in square brackets. For example, "Press [up arrow]" means press the key with the "up arrow" symbol.
- cmd represents any system command.
- fn represents any filename.
- Any item left blank means that the system does not have that capability.

DOS, Windows for Workgroups, Windows/NT Notes:

- File sharing, logoff, passwords under DOS relate only to Windows/NT.
- Logoff and changing passwords relate only to Windows/NT. They are preceded with [Ctrl] - [Alt] - [Delete].
- Commands that relate to programs which are executing are preceded with [Ctrl] - [Esc].



