บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

ในการศึกษาเรื่อง "ระบบสารสนเทศงานทะเบียน สถาบันราชภัฏเชียงใหม่" ประกอบด้วยรายละเอียดของเอกสารและงานวิจัยที่เกี่ยวข้อง ดังนี้

2.1 ลักษณะทั่วไปขององค์กร

2.2 ทฤษฎีที่ใช้ในการวิจัย

2.3 งานวิจัยที่เกี่ยวข้อง

2.1 ลักษณะทั่วไปขององค์กร

สถาบันราชภัฏเชียงใหม่ เป็นสถานศึกษาในสังกัดกระทรวงศึกษาธิการ โดยมีการแบ่งส่วนราชการในสถาบันราชภัฏเชียงใหม่ (2544 : 9) ดังนี้

1. คณะครุศาสตร์
2. คณะมนุษยศาสตร์และสังคมศาสตร์
3. คณะวิทยาศาสตร์และเทคโนโลยี
4. คณะวิทยาการจัดการ
5. คณะเทคโนโลยีการเกษตร
6. สำนักงานอธิการบดี
7. สำนักวิจัยและบริการวิชาการ
8. สำนักศิลปวัฒนธรรม
9. สำนักกิจการนักศึกษา
10. สำนักวางแผนและพัฒนา
11. สำนักส่งเสริมวิชาการ

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

จากการศึกษาข้อมูลจากสถาบันราชภัฏเชียงใหม่ สามารถสรุปรายละเอียดเกี่ยวกับข้อมูลหลักสูตร ระดับการศึกษา ประเภทนักศึกษา ได้ดังนี้

1. หลักสูตรที่เปิดสอน มี 4 สาขาวิชา คือ

1.1 สาขาวิชาการศึกษา จำนวน 32 โปรแกรมวิชา

1.2 สาขาวิชาศิลปศาสตร์ จำนวน 22 โปรแกรมวิชา

1.3 สาขาวิชาวิทยาศาสตร์ จำนวน 22 โปรแกรมวิชา

1.4 สาขาวิชาการบริหารธุรกิจ จำนวน 4 โปรแกรมวิชา

2. ระดับการศึกษามี 5 ระดับ คือ อนุปริญญา ปริญญาตรี ปริญญาตรี 2 ปีหลังปริญญาโท และประกาศนียบัตรวิชาชีพครู

3. ประเภทนักศึกษา มีทั้งหมด 9 ประเภท คือ

3.1 นักศึกษาภาคปกติ

3.2 นักศึกษาโครงการการจัดการศึกษาเพื่อปวงชน (กศ.ปช.) ภาคเสาร์-อาทิตย์

3.3 นักศึกษาโครงการการจัดการศึกษาเพื่อปวงชน (กศ.ปช.) ภาคจันทร์-ศุกร์

3.4 นักศึกษาโครงการการจัดการศึกษาเพื่อปวงชน (กศ.ปช.) กศ.น. แม่ฮ่องสอน และกลุ่มสนใจ

3.5 นักศึกษาโครงการการจัดการศึกษาเพื่อปวงชน (กศ.ปช.) พระธรรมจาริก

3.6 นักศึกษาต่างชาติ

3.7 นักศึกษาประกาศนียบัตรวิชาชีพครู

3.8 นักศึกษาปริญญาโท

3.9 นักศึกษาโครงการขยายโอกาสทางการศึกษา

2.1.1 ลักษณะระบบงาน

ข้อมูลนักศึกษาสถาบันราชภัฏเชียงใหม่ อยู่ภายใต้การบริหารงานของสำนักส่งเสริมวิชาการ ซึ่งมีหน้าที่และรับผิดชอบข้อมูลของนักศึกษาทั้งสถาบัน ตั้งแต่นักศึกษาเริ่มเข้าเรียน ในสถาบัน จนกระทั่งสำเร็จการศึกษา โดยสามารถแบ่งภาระงานหลักๆ ได้ดังนี้

2.1.1.1 จัดการข้อมูลนักศึกษาใหม่

- รับรายงานตัวเข้าเป็นนักศึกษา (ข้อมูลประวัติส่วนตัว)
- กำหนดเลขรหัสประจำตัวนักศึกษา
- กำหนดเลขรหัสหมู่เรียน ชื่อหมู่เรียน
- จัดการอาจารย์ที่ปรึกษาประจำหมู่เรียน

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

- พิมพ์รายงานต่างๆ เช่น ข้อมูลนักศึกษาจำแนกตามหมู่เรียนเพื่อมอบให้อาจารย์ที่ปรึกษา ข้อมูลนักศึกษาจำแนกตามคณะ ข้อมูลนักศึกษาจำแนกตามสาขาวิชา เป็นต้น

2.1.1.2 การลงทะเบียน

- จัดเตรียมปฏิทินการศึกษา
- เตรียมข้อมูลวิชาเรียนที่เปิดสอนก่อนการลงทะเบียน
- รับลงทะเบียนเรียนปกติ เพิ่ม ถอน ยกเลิก วิชาเรียน
- รับลงทะเบียนการ โอน ยกเว้น วิชาเรียน
- ตรวจสอบการลงทะเบียนของนักศึกษา
- พิมพ์รายงานเกี่ยวกับการลงทะเบียน เช่น เอกสารการลงทะเบียนเรียน (ลงทะเบียนเพิ่มวิชา ลงทะเบียนถอนวิชา ลงทะเบียนยกเลิกวิชา ลงทะเบียนโอนวิชา ลงทะเบียนยกเว้นวิชา) รายงานรายชื่อนักศึกษาที่ลงทะเบียนแต่ละหมู่เรียน รายงานสรุป การลงทะเบียนเรียน เป็นต้น

2.1.1.3 ประเมินผลการศึกษา

- บันทึกผลการศึกษา
- ประมวลผลการศึกษาของนักศึกษาแต่ละคน
- พิมพ์รายงานต่างๆ เช่น ใบแจ้งผลการศึกษาของแต่ละภาคเรียน ใบรายงานผลการเรียน (Transcript) รายงานข้อมูลนักศึกษาที่สำเร็จการศึกษาเพื่อขออนุมัติสำเร็จการศึกษา รายงานสรุปจำนวนนักศึกษาที่สำเร็จการศึกษา เป็นต้น

2.1.2 ขั้นตอนการดำเนินงานของแต่ละระบบงาน

2.1.2.1 จัดการข้อมูลนักศึกษาใหม่ การจัดการข้อมูลนักศึกษาใหม่ มีขั้นตอนในการดำเนินการ ดังนี้

(1) รับรายงานตัวผู้ที่สอบคัดเลือกได้ เพื่อลงทะเบียนเป็นนักศึกษาของสถาบัน

(2) จัดการข้อมูลหลักของนักศึกษา คือ กำหนดเลขรหัสประจำตัวนักศึกษา กำหนดเลขรหัสหมู่เรียน ชื่อหมู่เรียน จัดการอาจารย์ที่ปรึกษาประจำหมู่เรียน

(3) บันทึกข้อมูลประวัตินักศึกษาเข้าระบบฐานข้อมูล และข้อมูลหลักของนักศึกษา ตามข้อ (2)

(4) ตรวจสอบความถูกต้องของข้อมูลที่ได้บันทึก เพิ่มเติม แก้ไข และลบข้อมูลที่ผิดพลาด

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

2.1.2.2 การลงทะเบียน

(1) จัดการข้อมูลหลักก่อนการลงทะเบียน ได้แก่ รายวิชาที่จัดการเรียนการสอน อาจารย์ผู้สอน ตารางเรียน และปฏิทินการศึกษา

(2) นักศึกษารับเอกสารการลงทะเบียน และรายการจัดการเรียนการสอนที่สำนักส่งเสริมวิชาการ

(3) พบอาจารย์ที่ปรึกษาเพื่อแนะนำข้อมูล และรายวิชาเพื่อลงทะเบียนเรียน และอาจารย์ที่ปรึกษาลงนามในเอกสารการลงทะเบียน

(4) นักศึกษานำเอกสารการลงทะเบียน ยื่นให้สำนักส่งเสริมวิชาการ

(5) สำนักส่งเสริมวิชาการรับลงทะเบียน บันทึกข้อมูลการลงทะเบียน และพิมพ์เอกสารการรับลงทะเบียน

(6) นักศึกษานำเอกสารการรับลงทะเบียน ไปชำระเงินที่ธนาคาร โดยเจ้าหน้าที่ของธนาคารประทับตราและลงนามในเอกสารการรับลงทะเบียน และนำเอกสารดังกล่าวส่งคืนสำนักส่งเสริมวิชาการ เพื่อเก็บไว้เป็นหลักฐาน

2.1.2.3 ประเมินผลการศึกษา

(1) ฝ่ายทะเบียนและประเมินผล พิมพ์รายชื่อนักศึกษาให้อาจารย์ผู้สอนเพื่อเป็นหลักฐานการเข้าสอบ และส่งเกรดหลังการประเมินผลการศึกษาแล้ว

(2) อาจารย์ผู้สอน ส่งผลการเรียนของนักศึกษาตามขั้นตอน คือ โปรแกรมวิชา คณะวิชา และสำนักส่งเสริมวิชาการ ตามลำดับ

(3) ฝ่ายทะเบียนและประเมินผล บันทึกข้อมูลผลการเรียนของนักศึกษาแต่ละคน จำแนกตามรายวิชา หมู่เรียน และตรวจสอบความถูกต้อง

(4) ประเมินผลค่าระดับคะแนนเฉลี่ยประจำปีการศึกษา ค่าระดับคะแนนเฉลี่ยสะสม ตามข้อบังคับสภาประจำสถาบันราชภัฏเชียงใหม่ ว่าด้วยการประเมินผลการศึกษาระดับอนุปริญญาและปริญญาตรี พ.ศ. 2541 และประกาศผลการศึกษา พร้อมพิมพ์ใบแจ้งผลการเรียนให้นักศึกษาแต่ละคน

(5) พิมพ์รายชื่อนักศึกษาที่สำเร็จการศึกษาเพื่อขออนุมัติสำเร็จการศึกษาจากสภาประจำสถาบัน

(6) ประกาศรายชื่อที่สำเร็จการศึกษา และพิมพ์ใบรายงานผลการเรียนให้นักศึกษาที่สำเร็จการศึกษา

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

2.2 ทฤษฎีที่ใช้ในการศึกษา

ในการศึกษา เรื่อง “ระบบสารสนเทศงานทะเบียน สถาบันราชภัฏเชียงใหม่” โดยใช้การจัดการรูปแบบของ Client/Server ประกอบด้วยรายละเอียดของเอกสาร ทฤษฎี และงานวิจัยที่เกี่ยวข้อง ดังนี้

2.2.1 ข้อมูลและสารสนเทศ

2.2.2 การวิเคราะห์และออกแบบระบบ

2.2.3 ระบบฐานข้อมูล

2.2.4 การออกแบบฐานข้อมูล

2.2.5 ระบบ Client/Server

2.2.1 ข้อมูลและสารสนเทศ

2.2.1.1 ความหมายของข้อมูล

จรณิต แก้วกังวาล (2538 : 10) ให้ความหมายว่า ข้อมูล (Data) คือ ข้อเท็จจริงขั้นต้น ซึ่งอาจเรียกว่าเป็นวัตถุดิบของสารสนเทศ (information) เมื่อข้อมูลถูกนำมาประมวล (เรียงลำดับ แยกประเภท เชื่อมโยง คำนวณ หรือสรุปผล) และจัดให้อยู่ในรูปแบบที่ใช้ประโยชน์ได้

เกษมชาติ ทองธา (2540 : 118) ให้ความหมายว่า ข้อมูล หมายถึง กลุ่มอักขระที่เมื่อนำมารวมกันแล้วมีความหมายอย่างใดอย่างหนึ่ง หรือหมายถึงคำที่ใช้อธิบาย หรือข้อความที่กล่าวถึงสิ่งใดสิ่งหนึ่ง ข้อความนั้นอาจเป็นตัวเลข หรือสัญลักษณ์ ซึ่งสามารถนำไปประมวลผลด้วยเครื่องคอมพิวเตอร์ได้

2.2.1.2 ความหมายของสารสนเทศ

จรณิต แก้วกังวาล (2538 : 10) ได้ให้ความหมายว่า สารสนเทศ (Information) หมายถึง ข้อมูลที่ถูกนำมาประมวล (เรียงลำดับ แยกประเภท เชื่อมโยง คำนวณ หรือสรุปผล)

วาสนา สุขกระสานติ (2540 : 22) ได้ให้ความหมายว่า สารสนเทศ (Information) หมายถึง ข่าวสารที่ได้จากการนำ ข้อมูลดิบ (raw data) มาคำนวณทางสถิติหรือประมวลผลอย่างใดอย่างหนึ่ง ซึ่งข่าวสารที่ได้ออกมานั้นจะอยู่ในรูปที่สามารถนำไปใช้งานได้ทันที

ศรีไพร ศักดิ์รุ่งพงศากุล (2544 : 17) ได้ให้ความหมายว่า สารสนเทศ หมายถึง สิ่งที่ได้จากการประมวลผลแล้ว ซึ่งในบางครั้งสารสนเทศอาจจะเป็นข้อมูลเพื่อการประมวลผลให้ได้ข้อสนเทศอีอย่างหนึ่งก็ได้ เช่น คะแนนสอบของนักศึกษาเป็นข้อมูล เมื่อผ่านกระบวนการตัดเกรด

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

จะได้เกรดเป็นสารสนเทศ และเมื่อนำเกรดนักศึกษาไปคำนวณหาค่าเฉลี่ย เกรดของนักศึกษาจะเป็นข้อมูล และสารสนเทศที่ได้ คือ เกรดเฉลี่ย (GPA)

สานิตย์ กายาผาด (2542 : 2) คำว่าสารสนเทศ ซึ่งมีความหมายกว้าง ๆ ว่า ข้อมูลข่าวสารความรู้ต่าง ๆที่มีการบันทึกอย่างเป็นระบบ ตามหลักวิชาการ เพื่อนำมาเผยแพร่และใช้งานต่าง ๆ ทุกสาขา

ลารี่ ลอง (2543 : 4) สารสนเทศ นั้นหมายถึงสิ่งที่ทำให้เกิดความรู้เข้าใจตรงกันระหว่างผู้ให้และผู้รับ

สาทิพย์ ธรรมชีวีวงศ์ (2544 : 2) ได้ให้ความหมาย สารสนเทศ หมายถึงข้อความรู้ที่ประมวลได้จากข้อมูลต่าง ๆ ที่เกี่ยวข้องในเรื่องนั้น จนได้ข้อสรุปที่เป็นประโยชน์และทำให้เกิดความรู้ความเข้าใจที่ตรงกันระหว่างผู้ให้และผู้รับ

2.2.1.3 ลักษณะของสารสนเทศที่ดี

จรณิต แก้วกังวาล (2538 : 11) ได้กล่าวถึง ลักษณะสารสนเทศที่ดี ซึ่งจำแนกได้เป็น 5 ลักษณะ ดังนี้

1. เป็นปัจจุบัน (Current) ข้อมูลอาจมีการปรับเปลี่ยนไปได้เรื่อยตามกาลเวลา ระบบสารสนเทศที่ดีจะต้องสามารถยืดหยุ่น มีการปรับเปลี่ยนค่าให้เป็นปัจจุบัน และ/หรือ คงค่าเก่าเก็บไว้เพื่อประโยชน์การใช้งานต่าง ๆ กัน

2. ทันเวลา (Timely) สารสนเทศที่มีคุณค่าทางเวลา ถ้าไม่ได้สารสนเทศในเวลาที่ต้องการอาจจะเกิดการสูญเสียโอกาส ระบบสารสนเทศที่มีประสิทธิภาพ คือ ระบบที่จะต้องจัดสรรให้ได้สารสนเทศเมื่อผู้ใช้ต้องการในเวลาที่ต้องการ

3. มีค่าเที่ยงตรง (Relevant) ผู้ใช้ต้องการสารสนเทศที่ตรงกับงานของเขา ถ้าผู้ใช้ได้ข้อมูลที่ไม่สมบูรณ์ หรือมีรายละเอียดปลีกย่อยมากเกินไป ผู้ใช้ก็จะทำงานในส่วนของตนได้ไม่เต็มที่ ยิ่งสารสนเทศที่ได้รับ ตรงตามความต้องการของผู้ใช้แต่ละคนมากเท่าใด ระบบสารสนเทศนั้นก็จะถูกจัดว่าเป็นระบบที่มีประสิทธิภาพมากยิ่งขึ้นเท่านั้น

4. มีความคงที่ (Consistent) ในหลายกรณี สารสนเทศเองก่อให้เกิดความขัดแย้ง ข้อมูลในหลายที่อาจไม่ตรงกัน วิธีการประมวลผลที่ต่างกัน อาจทำให้เกิดความคลาดเคลื่อนในผลลัพธ์ที่ได้ จุดมุ่งหมายหลักของระบบสารสนเทศข้อหนึ่ง คือ พยายามทำให้เกิดความขัดแย้งน้อยที่สุด ข้อมูลมีความคงที่มากที่สุดเท่าที่จะทำได้

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

5. นำเสนอรูปแบบที่มีประโยชน์ (Presented in Usable Form) ถึงแม้ระบบจะมีลักษณะทั้ง 4 ประการข้างต้น แต่ถ้านำเสนอผลลัพธ์ในรูปแบบที่ผู้ใช้นำไปใช้ประโยชน์ไม่ได้ ระบบดังกล่าวก็จะมีค่าน้อยเต็มที ระบบสารสนเทศที่มีประสิทธิภาพ คือ ระบบที่มีความยืดหยุ่นในการนำเสนอสารสนเทศให้กับผู้ที่ต้องการใช้สารสนเทศนั้นๆ

2.2.2 การวิเคราะห์และการออกแบบระบบ (System Analysis and Design)

ลักขณา พฤกษากร (2536 : 65) ได้ให้ความหมาย การวิเคราะห์และออกแบบระบบ คือ วิธีการที่ใช้ในการสร้างระบบสารสนเทศขึ้นมาใหม่ ในธุรกิจใดธุรกิจหนึ่ง หรือระบบย่อยของธุรกิจ นอกจากการสร้างระบบสารสนเทศใหม่แล้ว การวิเคราะห์ระบบ ยังช่วยในการแก้ไขระบบสารสนเทศเดิมที่มีอยู่แล้วให้ดีขึ้นด้วยก็ได้ การวิเคราะห์และออกแบบระบบก็คือ การหาความต้องการ (Requirements) ของระบบสารสนเทศว่า คืออะไร หรือต้องการเพิ่มเติมอะไรเข้าไปในระบบ และการออกแบบก็คือ การนำเอาความต้องการของระบบ มาเป็นแบบแผน หรือเรียกว่าพิมพ์เขียวในการสร้างระบบสารสนเทศนั้น ให้ใช้งานได้จริง ตัวอย่างระบบสารสนเทศ เช่น ระบบงานขาย ความต้องการของระบบก็คือ สามารถติดตามยอดขายได้เป็นระยะ เพื่อฝ่ายบริหารสามารถปรับปรุงการขายได้ทันท่วงที

2.2.2.1 ทฤษฎีการวิเคราะห์และออกแบบโครงสร้าง (Structure System Analysis Design)

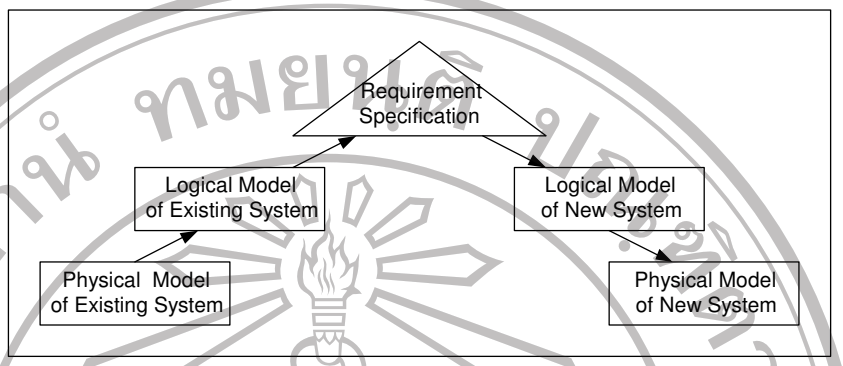
- **จากระบบปัจจุบัน สู่ระบบใหม่ (Existing system to New system)**

เมสินี นาคมณี (2545 : 3) ได้กล่าวถึง สิ่งที่ควรเริ่มต้นในการวิเคราะห์และออกแบบระบบ คือ ทำความเข้าใจหรือวิเคราะห์ระบบงานเดิม เพื่อให้ทราบถึงปัญหา และสามารถนำระบบงานใหม่มาใช้ในการแก้ไขปัญหาได้ โดยทั้งนี้ ระบบงานเดิมที่นำคอมพิวเตอร์ไปใช้แก้ปัญหา อาจเป็นระบบงานที่ใช้ระบบคอมพิวเตอร์ที่ไม่สามารถรองรับการทำงานหรือเป็นระบบงานเดิมในลักษณะระบบมือก็ได้ โดยเมื่อนำมาวิเคราะห์และออกแบบ สิ่งที่ผู้ศึกษาควรทำความเข้าใจไว้เสมอ คือ การวิเคราะห์และออกแบบระบบคอมพิวเตอร์ มิใช่การนำคอมพิวเตอร์ไปแทนการทำงานของคนเท่านั้น หากแต่สามารถวิเคราะห์และออกแบบระบบ เพื่อปรับปรุงให้ได้กระบวนการทำงานที่มีประสิทธิภาพด้วย

หลักการวิเคราะห์แบบ วิเคราะห์โครงสร้าง (Structure Analysis) มักจะแทนการปรับเปลี่ยนจากระบบปัจจุบันไปยังระบบใหม่ ตามลักษณะการเปลี่ยนจากทางภายภาพของระบบปัจจุบันไปยัง

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

ระบบตรรกะของระบบปัจจุบัน จากนั้น จึงปรับเปลี่ยนจากระบบตรรกะของระบบปัจจุบันไปสู่ระบบใหม่ และเชื่อมโยงไปถึงระบบกายภาพของระบบใหม่ ดังนี้



รูปที่ 2.1 แสดงการวิเคราะห์โมเดลกระบวนการ

(ที่มา : Jerry FitzGerald & Ardra F.FitzGerald (1987 : 57))

- ผังการทำงานเชิงกายภาพของระบบปัจจุบัน (Physical Model of Existing System)

จะทำการรวบรวมข้อมูลจากการทำงานจริง และทำการจำลองการทำงานเดิมตามลักษณะทางกายภาพ เพื่อแสดงข้อดีและข้อบกพร่องของระบบ ทำให้ทราบการทำงานของกระบวนการต่าง ๆ เพื่อให้เกิดแนวคิดในการสร้างระบบใหม่ที่มีโครงสร้างที่ดีขึ้น รวมทั้งช่วยในการสื่อสารระหว่างตัวผู้ศึกษา กับผู้ที่เกี่ยวข้องในระบบงานเดิม

- ผังการทำงานเชิงตรรกะของระบบปัจจุบัน (Logical Model of Existing System)

ทำหน้าที่แสดงข้อมูลหรือฟังก์ชั่นในการทำงานของระบบปัจจุบัน โดยไม่คำนึงถึงลักษณะทางกายภาพของระบบงาน

- ผังการทำงานเชิงตรรกะของระบบใหม่ (Logical Model of New System)

ผู้ศึกษาต้องทำการวิเคราะห์แนวคิดของระบบงานใหม่ หรือฟังก์ชั่นในการทำงานเพื่อให้ระบบงานบรรลุวัตถุประสงค์ และเกิดแนวคิดของระบบการทำงานใหม่ จากการนำข้อบกพร่องในระบบงานปัจจุบันมาทำการปรับปรุงแก้ไข โดยไม่จำเป็นต้องแสดงลักษณะทางกายภาพของระบบงานใหม่

- ผังการทำงานเชิงกายภาพของระบบใหม่ (Physical Model of New System)

ทำหน้าที่แสดงผลลัพธ์ของระบบงานที่จะได้จากแนวคิดและการวิเคราะห์และออกแบบระบบงานใหม่ แสดงลักษณะการทำงาน อุปกรณ์ที่เกี่ยวข้อง เพื่อให้สามารถจำลองลักษณะของระบบงานใหม่ที่จะเกิดขึ้นภายหลังจากการพัฒนาระบบงาน

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

- **ผังกิจกรรมการปฏิบัติงาน (Activity Diagram)**

เมสินี นาคมณี (2545 : 9) ได้กล่าวถึง ผังกิจกรรมการปฏิบัติงานว่า ทำหน้าที่ สื่อถึงขั้นตอนของระบบงาน พนักงาน และอุปกรณ์ที่เกี่ยวข้องในแต่ละขั้นตอน ซึ่งผู้ศึกษาสามารถนำมาใช้ในการวิเคราะห์ข้อดีและข้อเสีย หรือปัญหาของระบบงาน เพื่อนำมาแก้ไขได้

นอกจากนั้น เมสินี นาคมณี (2545 : 9-11) ยังได้ให้รายละเอียดของผังระบบ ผังการแยกฟังก์ชั่นงานย่อย และผังการไหลของข้อมูลตามลำดับดังนี้

- **ผังระบบ (Context Flow Diagram)**

แสดงลักษณะของระบบงาน รวมทั้งเป็นตัวกำหนดขอบเขตการศึกษา โดยแสดงถึงขอบเขตและผู้เกี่ยวข้องในระบบงาน ทั้งนี้โดยสิ่งที่อยู่ภายนอกระบบ ผู้ศึกษาไม่จำเป็นต้องทำการวิเคราะห์ เพราะอยู่นอกเหนือการควบคุม

- **ผังการแยกฟังก์ชั่นงานย่อย (Decomposition Diagram)**

การแยกฟังก์ชั่นงานย่อย เปรียบเสมือนการเชื่อมระหว่างผังระบบงานและผังการไหลของข้อมูล ทั้งนี้ในผังระบบงาน จะแสดงลักษณะเพียงระบบโดยรวมและเอนทิตี้หรือบุคลากรภายนอกที่เข้ามาเกี่ยวข้อง แต่เพื่อจะแยกย่อยไปถึงผังการไหลของข้อมูลนั้น จำเป็นต้องทำการวิเคราะห์ขั้นตอนย่อยหรือฟังก์ชั่นย่อยในการทำงานของแต่ละระบบงานนั้น ๆ เพื่อเป็นแนวทางในการจัดทำผังการไหลของข้อมูลต่อไป

- **ผังการไหลของข้อมูล (Data Flow Diagram : DFD)**

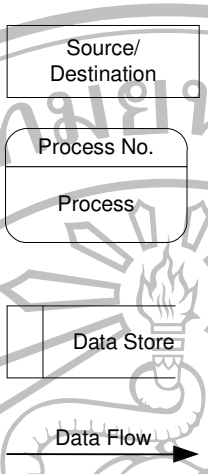
ผังการไหลของระบบ แสดงการไหลของข้อมูลภายในระบบ โดยสามารถแบ่งย่อยออกเป็นระดับ (Level) ต่าง ๆ โดยผังระดับบนต่อเนื่องจากผังของระบบ คือ ผังการไหลของระบบระดับ 0 (Data Flow Diagram : DFD Level 0) ซึ่งทำหน้าที่เป็นผังแม่ เพื่อแยกย่อยลงไปในระดับผังลูกในแต่ละระดับอีกครั้ง กล่าวคือ ผังการไหลของระบบระดับ 1,2,…,N (Data Flow Diagram : DFD Level 1,2,...,N)

1. รูปแบบแผนภาพกระแสข้อมูล (Data Flows Model)

การออกแบบระบบงาน ในวงจรการพัฒนาระบบ นิยมใช้แผนภาพกระแสข้อมูล เนื่องจากเป็นแบบจำลองที่ประกอบด้วยรูปภาพ ที่สามารถแสดงถึงส่วนประกอบของฟังก์ชัน การทำงานต่างๆ ของระบบ งานสารสนเทศ ซึ่งประกอบด้วย ส่วนประมวลผล ส่วนจัดเก็บข้อมูล ทิศทางการไหลของข้อมูลระหว่างส่วนประมวลผลต่างๆ รวมทั้งบุคคลและสิ่งของต่างๆ ที่กระทำกับส่วนประมวลผลเหล่านั้น โดยแผนภาพกระแสข้อมูลที่ใช้จะเป็นรูปแบบของ Gene - Sarson

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

ลักขณา พฤกษากร (2536 : 60) แสดงสัญลักษณ์ของแผนภาพกระแสข้อมูลดังรูปที่ 2.2



แหล่งข้อมูลที่อยู่ภายนอกระบบ

กระบวนการทำงาน

แหล่งเก็บข้อมูล

ทิศทางของกระแสข้อมูล

รูปที่ 2.2 แสดงสัญลักษณ์แผนภาพกระแสข้อมูล

- **ความหมาย สัญลักษณ์แผนภาพกระแสข้อมูล**

1. รูปสี่เหลี่ยมจัตุรัส หมายถึง หน่วยงานภายนอก (external entity) ซึ่งจะทำหน้าที่เป็นตัวให้หรือรับข้อมูลจากระบบ นั่นก็คือ หน่วยงานภายนอกจะเป็นได้ทั้งจุดกำเนิด หรือจุดปลายทางของข้อมูล

2. ลูกศร จะแสดงทิศทางการเคลื่อนที่ของข้อมูลจากจุดหนึ่งไปยังอีกจุดหนึ่ง โดยทิศทางที่ข้อมูลเคลื่อนที่ไปจะไปในทิศทางเดียวกับหัวลูกศรเสมอ

3. รูปสี่เหลี่ยมผืนผ้ามุมมน จะหมายถึงขั้นตอนหรือกระบวนการหนึ่งในระบบงาน ซึ่งขั้นตอนเหล่านี้มักจะทำให้ลักษณะของข้อมูลได้เปลี่ยนแปลงไป

4. รูปสี่เหลี่ยมผืนผ้าปลายเปิด จะหมายถึงแฟ้มข้อมูลซึ่งอาจจะถูกเก็บอยู่ในที่ไหนก็ได้ เช่น ในแผ่นดิสก์ เทป CD-ROM หรือออปติคัลดิสก์ เป็นต้น แฟ้มข้อมูลใน DFDs จะมีความหมายเพียงเป็นตัวที่ใช้เก็บข้อมูล และพร้อมที่จะส่งข้อมูลให้เมื่อระบบต้องการเท่านั้น

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

2. กฎการเขียนผังการไหลของข้อมูล

- **กระบวนการ (Process)**

คือ กระบวนการที่ต้องทำในระบบ โดยจะเป็นกริยา (Verb) จำนวนโปรเซส มีอยู่ระหว่าง 2-7 โปรเซส หรือในบางตำราได้กำหนดจำนวนโปรเซสควรอยู่ในระหว่าง 7 บวกลบด้วย 2

- **แหล่งข้อมูล (Data Store)**

- ข้อมูลต้องไหลเข้า หรือออก โดยกระบวนการ ไม่สามารถไหลไปยังแหล่งข้อมูลหรือเอ็นทิตี้ภายนอกโดยตรงได้

- ชื่อเป็น นาม (Noun) โดยแสดงชื่อแหล่งข้อมูล

- **เอนทิตี้ภายนอก (Extend Entity/Source/Sink)**

- ข้อมูลไม่สามารถไหลจากเอนทิตี้ภายนอกไปยังอีกเอนทิตี้หนึ่งได้

- ชื่อเป็นคำนาม (Noun) โดยแสดงชื่อเอนทิตี้ภายนอกที่เกี่ยวข้อง

- **การไหลของข้อมูล (Data Flow)**

- ข้อมูลสามารถไหลในทิศทางเดียวหรือสองทิศทางระหว่างแหล่งข้อมูลได้

- การเพิ่มลูก (Fork) คือ การไหลของข้อมูลเดียวกันไปยังเอ็นทิตี้ภายนอก 2 เอ็นทิตี้หรือมากกว่าได้

- ข้อมูลไม่สามารถไหลย้อนกลับไปยังกระบวนการเดิมที่ไหลมาได้

- การไหลเข้าของข้อมูลไปยังแหล่งข้อมูล หมายถึงการปรับปรุงข้อมูล (Update)

- การไหลออกของข้อมูลไปยังกระบวนการ หมายถึงการเรียกใช้ (Retrieve/Use)

- ชื่อเป็นคำนาม (Noun) โดยแสดงข้อมูลที่ไหล ทั้งนี้ไม่ควรแสดงชื่อเป็นลักษณะเอกสาร เช่น ใบสมัคร เนื่องจากอาจต้องการใช้ข้อมูลเฉพาะเช่น ชื่อ นามสกุล แต่ทั้งนี้สามารถยกเว้นได้ในกรณีที่ไม่ต้องการแสดงรายละเอียดย่อย

2.2.3 ระบบฐานข้อมูล (Database System)

2.2.3.1 ความหมายของฐานข้อมูล

พลพฐู ปียวรรณ (2539 : 45) ให้ความหมายว่า ฐานข้อมูล คือ แฟ้มข้อมูลที่ถูกรวบรวมและเก็บรักษาไว้ในที่เดียวกัน เพื่อขจัดปัญหาความซ้ำซ้อนของข้อมูลที่กระจัดกระจายอยู่ในหน่วยงานย่อยของกิจการ และสามารถให้บริการทางข้อมูลกับผู้ใช้หลาย ๆ คน

ยุพิน ไทยรัตนานนท์ (2540 : 176) ให้ความหมายว่า ฐานข้อมูล คือ การรวบรวมข้อมูล ที่มีความสัมพันธ์กันและอาจอยู่ต่างที่กัน ให้เสมือนอยู่ร่วมกัน เพื่อให้สามารถรับใช้งานที่มีวัตถุประสงค์แตกต่างกันของหน่วยงานต่างๆ โดยผู้ใช้ฐานข้อมูลไม่ได้รับรู้ข้อมูลทั้งหมดในฐานข้อมูล แต่รับรู้เฉพาะส่วนที่เกี่ยวข้องกับงานของตนเท่านั้น

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

ศิริลักษณ์ โรจนกิจอำนวย (2542:11) ให้ความหมายว่า ฐานข้อมูล ประกอบด้วยรายละเอียดของข้อมูลที่เกี่ยวข้องกัน ซึ่งนำมาใช้ในงานด้านต่างๆ ข้อมูลเหล่านี้จะมีการจัดเก็บไว้อย่างมีระบบ เพื่อประโยชน์ในการจัดการและเรียกใช้ข้อมูลได้อย่างมีประสิทธิภาพ

2.2.3.2 วัตถุประสงค์ของการใช้ฐานข้อมูล

ยุพิน ไทยรัตนานนท์ (2540 : 177) ได้กล่าวถึงวัตถุประสงค์ของการใช้ฐานข้อมูล ดังนี้

1. เพิ่มความเร็วในการพัฒนาโปรแกรม โปรแกรมเมอร์ไม่ต้องสนใจเกี่ยวกับการเก็บข้อมูลทางกายภาพ

2. ลดค่าใช้จ่ายในการบำรุงรักษาโปรแกรม ไม่มีปัญหาการแปลงผันข้อมูล เมื่อระบบขยายตัว

3. อำนวยความสะดวกให้ผู้ใช้ทั่วไปที่ไม่ใช่โปรแกรมเมอร์ สามารถเรียกดูข้อมูลได้อย่างมีประสิทธิภาพ เพราะมีภาษาระดับง่ายสำหรับผู้ใช้โดยเฉพาะ

4. สามารถใช้ข้อมูลร่วมกันได้ (Integration of Data) และสามารถจับกลุ่มข้อมูลได้หลายรูปแบบ

5. ควบคุมข้อมูลได้ง่ายขึ้น ไม่ว่าจะเป็นด้านความถูกต้องของข้อมูล หรือการกำหนดขอบเขตสิทธิของผู้ใช้ข้อมูล

2.2.3.3 องค์ประกอบของระบบฐานข้อมูล

ศิริลักษณ์ โรจนกิจอำนวย (2542 : 12-15) ได้จัดแบ่งองค์ประกอบของระบบฐานข้อมูลออกเป็น 5 ประเภท คือ

1. ฮาร์ดแวร์ (Hardware) หมายถึง เครื่องคอมพิวเตอร์ และอุปกรณ์ต่อพ่วงต่างๆ

2. โปรแกรม (Software) หมายถึง โปรแกรมที่ทำหน้าที่ควบคุมฮาร์ดแวร์ โปรแกรมที่ทำหน้าที่ควบคุมดูแลการสร้าง การเรียกใช้ข้อมูล การจัดทำรายงาน การปรับเปลี่ยนแก้ไขโครงสร้าง หรืออาจกล่าวอีกนัยหนึ่งได้ว่า ระบบจัดการฐานข้อมูล (Database Management System : DBMS) คือ โปรแกรมที่ทำหน้าที่ในการจัดการฐานข้อมูล โดยจะเป็นสื่อกลางระหว่างผู้ใช้ และโปรแกรมประยุกต์ต่าง ๆ ที่มีอยู่ในระบบฐานข้อมูล ซึ่งมีหน้าที่สำคัญ ดังนี้

- ช่วยกำหนดและเก็บโครงสร้างฐานข้อมูล (Define and Store Database Structure)
- การบรรจุข้อมูลจากฐานข้อมูล (Load Database)
- เก็บและดูแลรักษาข้อมูล (Store and Maintain Data
- ประสานงานกับระบบปฏิบัติการ (Operating System)
- ช่วยควบคุมความปลอดภัย (Security Control)

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

- การทำข้อมูลสำรองและการกู้ (Backup and Recovery)
- ควบคุมการใช้ข้อมูลพร้อมกัน (Concurrency Control)
- ควบคุมความบูรณภาพของข้อมูล (Integrity Control)
- จัดทำพจนานุกรมข้อมูล (Data Dictionary)

3. ข้อมูล (Data) หมายถึง ข้อมูลที่เป็น ตัวอักษร รูปภาพ เสียง ซึ่งการจัดเก็บในลักษณะฐานข้อมูล จะมีการเก็บรวบรวมข้อมูล ให้เป็นศูนย์กลางข้อมูลอย่างมีระบบ สามารถเรียกใช้ร่วมกันได้ ผู้ใช้จะมองภาพในลักษณะข้อมูลที่แตกต่างกัน ผู้ใช้บางคนอาจมองภาพของข้อมูลที่ถูกจัดเก็บไว้ในสื่อเก็บข้อมูลจริง (Physical Level) บางคนอาจมองภาพของข้อมูลจากการใช้งานของผู้ใช้ (External Level)

4. บุคลากร (People) ในระบบฐานข้อมูล มีบุคลากรที่เกี่ยวข้อง ดังนี้

- ผู้ใช้ทั่วไป (User)
- พนักงานปฏิบัติการ (Operator)
- นักวิเคราะห์ระบบ (System Analyst)
- ผู้เขียนโปรแกรมประยุกต์ใช้งาน (Programmer)
- ผู้บริหารฐานข้อมูล (Data Administrator : DBA)

5. **ขั้นตอนการปฏิบัติงาน (Procedures)** หมายถึง การจัดทำเอกสารที่ระบุขั้นตอนการทำงานของหน้าที่งานต่าง ๆ ในระบบฐานข้อมูล ทั้งในสภาวะปกติ และในสภาวะที่ระบบอาจเกิดปัญหา (Failure) ซึ่งจะเป็นขั้นตอนการปฏิบัติงานสำหรับบุคลากรในทุกระดับขององค์กร

2.2.3.4 รูปแบบของฐานข้อมูล

ศิริลักษณ์ โรจนกิจอำนวย (2542:26) จัดแบ่งรูปแบบของฐานข้อมูลออกเป็น 4 ประเภท คือ

1. ฐานข้อมูลแบบข่ายงาน (Network Database)
2. ฐานข้อมูลแบบลำดับขั้น (Hierarchical Database)
3. ฐานข้อมูลเชิงสัมพันธ์ (Relational Database)
4. ฐานข้อมูลเชิงวัตถุ (Object - Oriented Database)

ในการศึกษาครั้งนี้ ผู้วิจัยได้พัฒนาโปรแกรมโดยใช้รูปแบบฐานข้อมูลแบบฐานข้อมูลเชิงสัมพันธ์ ซึ่ง ศิริลักษณ์ โรจนกิจอำนวย (2542:26-31) ได้กล่าวถึงฐานข้อมูลเชิงสัมพันธ์ ไว้ดังนี้

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

ฐานข้อมูลเชิงสัมพันธ์ เป็นการจัดเก็บข้อมูลแบบตารางที่มีลักษณะเป็น 2 มิติ คือ แถว (Row) และคอลัมน์ (Column) ที่ประกอบด้วยแอททริบิวต์ที่แสดง คุณสมบัติของรีเลชั่นหนึ่งๆ โดยรีเลชั่นต่างๆ ได้ผ่านกระบวนการทำรีเลชั่นให้เป็นบรรทัดฐาน (Normalized) เพื่อลดความซ้ำซ้อน และเพื่อให้การจัดการฐานข้อมูลเป็นไปอย่างมีประสิทธิภาพ และเป็นฐานข้อมูลซึ่งให้ภาพของข้อมูลในระดับภายนอก (External Level หรือ View) และระดับแนวคิด (Conceptual Level)

ข้อดีของฐานข้อมูลเชิงสัมพันธ์

- ฐานข้อมูลเชิงสัมพันธ์เป็นกลุ่มของรีเลชั่น หรือตารางที่ข้อมูลถูกจัดเก็บเป็นแถวและคอลัมน์ ซึ่งทำให้ผู้ใช้เห็นภาพของข้อมูลได้ง่าย
- ผู้ใช้ไม่ต้องรู้ว่าข้อมูลถูกจัดเก็บจริงอย่างไร รวมถึงวิธีการเรียกใช้ข้อมูล (Access Approach)
- ภาษาที่ใช้ในการเรียกดูข้อมูล เป็นลักษณะคล้ายภาษาอังกฤษ และไม่จำเป็นต้องเขียนเป็นลำดับขั้นตอน (Procedural)
- การเรียกใช้หรือเชื่อมโยงข้อมูลทำได้ง่าย โดยใช้โอเปอเรเตอร์ทางคณิตศาสตร์
- การลำดับความสัมพันธ์ของเรคคอร์ดลูกจะลำดับจากซ้ายไปขวา

2.2.3.5 สถาปัตยกรรมของฐานข้อมูล

ศิริลักษณ์ โรจนกิจอำนวย (2542:34-37) จัดแบ่งสถาปัตยกรรมของฐานข้อมูลเป็น 3 ระดับ คือ

1. ระดับภายนอกหรือวิว เป็นระดับข้อมูลที่ประกอบด้วยภาพผู้ใช้แต่ละคนมองข้อมูล (View) เค้าร่างของข้อมูลระดับนี้เกิดจากภาพและความต้องการข้อมูลของผู้ใช้
2. ระดับแนวคิด ประกอบด้วยเค้าร่างที่อธิบายถึงฐานข้อมูลรวมว่ามีเอนทิตี้ โครงสร้างของข้อมูล ความสัมพันธ์ของข้อมูล กฎเกณฑ์และข้อจำกัดต่าง ๆ ข้อมูลในระดับนี้เป็นข้อมูลที่ผ่านการวิเคราะห์และออกแบบ โดย ผู้บริหารฐานข้อมูล (DBA) หรือนักวิเคราะห์และออกแบบระบบฐานข้อมูล เป็นระดับของข้อมูลที่ออกแบบเพื่อให้ผู้ใช้ข้อมูลในระดับภายนอกสามารถเรียกใช้ข้อมูลได้ ซึ่งผู้ใช้ทั่วไปในระดับภายนอกอาจจะต้องการใช้ข้อมูลที่แตกต่างกัน
3. ระดับภายใน (Internal หรือ Physical Level) ประกอบด้วยเค้าร่างที่จัดเก็บข้อมูลจริงๆ ว่ามีโครงสร้างการจัดเก็บรูปแบบใด รวมถึงวิธีการเข้าถึงข้อมูลต่างๆ ในฐานข้อมูล เพื่อดึงข้อมูลที่ต้องการ

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

2.2.4 การออกแบบฐานข้อมูล

2.2.4.1 วงจรการพัฒนาระบบ

อำไพ พรประเสริฐ (2537 : 18-30) กล่าวถึงขั้นตอนการพัฒนาระบบ ประกอบด้วย 7 ขั้นตอน คือ

1. เข้าใจปัญหา การศึกษาระบบปัจจุบัน และตระหนักถึงปัญหาของระบบ

2. ศึกษาความเป็นไปได้ รวบรวมข้อมูล การกำหนดว่าปัญหาของระบบ คืออะไร ความต้องการของระบบมีอะไรบ้าง ค่าใช้จ่าย และตัดสินใจว่าการพัฒนาสร้างระบบสารสนเทศหรือการแก้ไขระบบสารสนเทศเดิมมีความเป็นไปได้หรือไม่

3. วิเคราะห์ คือ ศึกษาระบบงานเดิม กำหนดความต้องการของระบบใหม่ โดยใช้เทคนิคการเก็บรวบรวมข้อมูล พจนานุกรมข้อมูล (Data dictionary) แผนภาพการกระแสข้อมูล (Data Flow Diagram) ข้อมูลเฉพาะการประมวลผล (Process Specification) รูปแบบข้อมูล (Data Model) รูปแบบระบบ (System Model) แบบทดลองผังงานระบบ (System Flowcharts)

4. ออกแบบ การออกแบบระบบใหม่ เพื่อให้สอดคล้องกับความต้องการ ของผู้ใช้และฝ่ายบริหาร โดยนักวิเคราะห์ระบบตัดสินใจเลือกคอมพิวเตอร์ฮาร์ดแวร์ และซอฟต์แวร์ ความปลอดภัยของระบบ โดยใช้เครื่องมือ พจนานุกรมข้อมูล แผนภาพการกระแส ข้อมูล ข้อมูลเฉพาะการประมวลผล รูปแบบข้อมูล รูปแบบระบบ ผังงานระบบ ผังงานโครงสร้าง (Structure Charts) ผังงาน HIPO (HIPO Charts) แบบฟอร์มข้อมูลนำเข้า (Input) และรายงาน (Output)

5. สร้าง หรือพัฒนาระบบ ขั้นตอนการเขียน ทดสอบโปรแกรม และเตรียมคู่มือการใช้และฝึกอบรม

6. การปรับเปลี่ยน การป้อนข้อมูล และเริ่มใช้ระบบใหม่ ภายใต้การดูแลของนักวิเคราะห์ระบบ โดยใช้งานควบคู่ไปกับระบบเก่า

7. บำรุงรักษา การแก้ไขโปรแกรมหลังการใช้งานแล้ว หรือการเพิ่มเติมสิ่งที่ต้องการ

2.2.4.2 วงจรชีวิตของการพัฒนาระบบฐานข้อมูล (Database Life Cycle)

วงจรชีวิตของการพัฒนาระบบฐานข้อมูล หรือ DBLC เป็นขั้นตอนที่กำหนดขึ้นเพื่อใช้เป็นแนวทางในการพัฒนาระบบฐานข้อมูลขึ้นใช้งาน

กิตติ ภักดีวัฒนะกุล และ จำลอง ครูอุตสาหะ (2541 : 97-98) กล่าวถึงวงจรชีวิตของการพัฒนาระบบฐานข้อมูล ประกอบด้วยขั้นตอนต่างๆ ดังนี้

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

1. Database Initial Study คือ ขั้นตอนที่ผู้พัฒนาระบบฐานข้อมูล จะต้องวิเคราะห์ความต้องการต่างๆ ของผู้ใช้ เพื่อกำหนดจุดมุ่งหมาย ปัญหา ขอบเขตและกฎระเบียบต่างๆ ของระบบฐานข้อมูลที่จะพัฒนาใช้

2. Database Design คือ ขั้นตอนที่ผู้พัฒนาระบบ นำรายละเอียดต่างๆ จากขั้นตอนการวิเคราะห์ในขั้นตอนแรก มาใช้เป็นแนวทางในการออกแบบฐานข้อมูล

3. Implementation and Loading คือ ขั้นตอนที่นำโครงร่างต่าง ๆ ของระบบฐานข้อมูลที่ได้จากการออกแบบในขั้นตอน Database Design มาสร้างเป็นตัวฐานข้อมูลที่จะใช้กับข้อมูลจริง รวมทั้งทำการแปลงข้อมูลของระบบงานเดิม ให้สามารถนำมาใช้ในระบบฐานข้อมูลที่พัฒนาขึ้นใหม่ ในกรณีที่ระบบเดิมมีการใช้คอมพิวเตอร์ในการประมวลผล

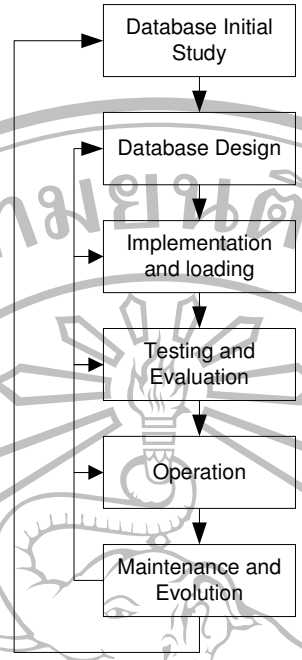
4. Testing and Evaluation คือ ขั้นตอนของการทดสอบระบบฐานข้อมูลที่พัฒนาขึ้น เพื่อหาข้อผิดพลาดต่าง ๆ รวมทั้งทำการประเมินความสามารถของระบบฐานข้อมูลนั้น เพื่อนำไปใช้เป็นแนวทางในการปรับปรุงให้ระบบฐานข้อมูลที่พัฒนาขึ้นนั้น สามารถรองรับความต้องการของผู้ใช้ในด้านต่าง ๆ ได้อย่างถูกต้อง และครบถ้วน

5. Operation คือ ขั้นตอนที่นำเอาระบบฐานข้อมูลที่พัฒนาขึ้นเสร็จเรียบร้อยแล้วไปใช้งานจริง

6. Maintain and Evolution คือ ขั้นตอนที่เกิดขึ้นระหว่างการใช้งานระบบฐานข้อมูลจริง เพื่อบำรุงรักษาให้ระบบฐานข้อมูลทำงานได้อย่างมีประสิทธิภาพ รวมทั้งเป็นขั้นตอนการแก้ไข และปรับปรุงระบบฐานข้อมูล ในกรณีที่มีการเพิ่ม หรือเปลี่ยนแปลงความต้องการของผู้ใช้ ที่ส่งผลกระทบต่อระบบฐานข้อมูล

ซึ่งทั้ง 6 ขั้นตอน สามารถแสดงด้วยแผนภาพ ดังรูป 2.3

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved



รูปที่ 2.3 แสดงวงจรชีวิตของการพัฒนาระบบฐานข้อมูล
(ที่มา : กิตติ ภักดีวัฒนะกุล และ จำลอง ครูอุตสาหะ (2541 : 98))

2.2.4.3 โครงร่างของฐานข้อมูล

ในการออกแบบฐานข้อมูล เมื่อได้ทำการศึกษาระบบงานที่จะออกแบบแล้ว ขั้นตอนต่อไป คือ การออกแบบโครงร่างของข้อมูลในระดับแนวคิด (Conceptual Schema Design) เพื่อกำหนดโครงสร้างของฐานข้อมูล ความหมาย (Semantics) ความสัมพันธ์และข้อจำกัดต่างๆ ของข้อมูลในระบบ

ในการออกแบบเค้าร่างของฐานข้อมูลในระดับแนวคิด สามารถใช้เครื่องมือร่วมในการทำงาน ซึ่งวิธีที่นิยมใช้อย่างแพร่หลายวิธีหนึ่ง คือ Entity Relationship Model (E-R Model)

- **แบบจำลองความสัมพันธ์ระหว่างข้อมูล**

แบบจำลองความสัมพันธ์ระหว่างข้อมูล เป็นเครื่องมือในการจำลองความสัมพันธ์ระหว่างข้อมูลที่คิดค้นโดย อี. เอฟ คอดด์ (E.F. Codd)

ศิริลักษณ์ โรจนกิจอำนวย (2542:141) กล่าวถึงขั้นตอน การออกแบบฐานข้อมูล โดยใช้แบบจำลองความสัมพันธ์ระหว่างข้อมูล ประกอบด้วยขั้นตอนดังนี้

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

1. กำหนดเอนทิตี้ที่เกี่ยวข้อง เมื่อได้ทำการศึกษารายละเอียดของระบบงานที่จะออกแบบ จะต้องกำหนดเอนทิตี้ที่เกี่ยวข้องว่ามีอะไรบ้าง เอนทิตี้เหล่านี้ เป็นเอนทิตี้ที่มีรายละเอียดย่อยของตัวเอง ที่สามารถจัดกลุ่มเป็น Supertype หรือ Subtype หรือไม่

2. พิจารณาความสัมพันธ์ระหว่างเอนทิตี้ หลังจากระบุเอนทิตี้แล้ว จะต้องกำหนดความสัมพันธ์ระหว่างเอนทิตี้ ว่าแต่ละเอนทิตี้มีความสัมพันธ์กันอย่างไร เพื่อเป็นประโยชน์ในการกำหนดแอททริบิวต์ที่จะใช้ในการอ้างอิงกันระหว่างเอนทิตี้

3. กำหนดคีย์หลัก คีย์รอง คีย์นอกและแอททริบิวต์ที่เกี่ยวข้อง หลังจากที่กำหนดเอนทิตี้ และความสัมพันธ์ระหว่างเอนทิตี้แล้ว ก็จะทำการระบุแอททริบิวต์ที่จะเป็นคีย์ ในแต่ละแอททริบิวต์ โดยเฉพาะคีย์หลัก (Primary Key) และคีย์นอก (Foreign Key) เพื่อใช้อ้างอิงถึงแอททริบิวต์ที่เป็นคีย์หลักในอีกเอนทิตี้หนึ่ง

4. การปรับเค้าร่างข้อมูลให้อยู่ในรูปแบบบรรทัดฐาน การปรับเค้าร่างข้อมูลให้อยู่ในรูปแบบบรรทัดฐาน (Normalization) เพื่อให้โครงสร้างข้อมูลไม่มีความซ้ำซ้อน ถูกต้องและเชื่อถือได้ ซึ่งโดยทั่วไปจะทำถึงรูปบรรทัดฐานขั้นที่ 3 เป็นอย่างน้อย ยกเว้น บางกรณีที่เข้าเงื่อนไขที่จะต้องปรับปรุงให้ถึงรูปแบบบรรทัดฐานบอยส์และคอดด์ หรือรูปแบบบรรทัดฐานขั้นที่ 4 และ 5

5. พิจารณาถึงลักษณะและขอบเขตของข้อมูลของแต่ละแอททริบิวต์ รวมถึง**ข้อจำกัด**หรือกฎเกณฑ์ที่มีผลต่อการจัดเก็บข้อมูล ขั้นตอนนี้ เป็นการพิจารณากฎเกณฑ์ต่างๆ ที่ศึกษาจากระบบงาน ว่าควรจะเก็บข้อมูลในลักษณะใด ขอบเขต และข้อจำกัด เพื่อเป็นการควบคุมความถูกต้องและเชื่อถือได้ของข้อมูล

สัญลักษณ์ที่ใช้ในแบบจำลองความสัมพันธ์ระหว่างข้อมูล

ศิริลักษณ์ โรจนกิจอำนวย (2542 : 141) กล่าวถึงสัญลักษณ์ และความหมายของสัญลักษณ์ที่ใช้ในแบบจำลองความสัมพันธ์ระหว่างข้อมูล ประกอบด้วยสัญลักษณ์ดังต่อไปนี้

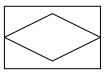
เอนทิตี้

เอนทิตี้ชนิดอ่อนแอ

ประเภทของความสัมพันธ์ระหว่างเอนทิตี้

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

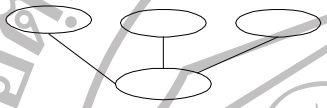
รูปที่ 2.4 แสดงสัญลักษณ์แบบจำลองความสัมพันธ์ระหว่างข้อมูล (ที่มา : ศิริลักษณ์ โรจนกิจ-อำนวย (2542 : 141))



เป็น Composite Entity หรือ Gerund ที่จะแปลงความสัมพันธ์ของเอนทิตี้แบบ M:N ให้เป็น 1:N

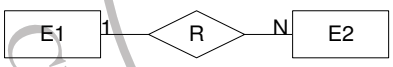
แอททริบิวต์

แอททริบิวต์ที่เป็นคีย์หลัก

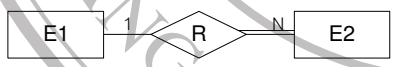


แอททริบิวต์ที่เป็นคีย์ผสม

แอททริบิวต์ที่แปลค่ามา (Derived Attribute)



ความสัมพันธ์ของข้อมูลระหว่างสองเอนทิตี้ (Cardinality Ratio)



ความสัมพันธ์ของเอนทิตี้ E2 ที่มีต่อ E1 แบบ Total Participation

รูปที่ 2.4 (ต่อ)

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

- **ความสัมพันธ์ระหว่างเอนทิตี้ (Relationship)** หมายถึง การนำเอาเอนทิตี้ มารวมกันแบบ Aggregation Abstraction สมาชิกของ Relationship จึงเกิดจากการจับคู่กันระหว่างสมาชิกของเอนทิตี้ ที่มารวมกันภายใต้ Relationship นั้น

กิตติ ภักดีวัฒนะกุล และจำลอง ครูอุตสาหะ (2541 : 33 - 38) กล่าวถึงความสัมพันธ์ระหว่างเอนทิตี้หนึ่งไปยังอีกเอนทิตี้หนึ่ง (Cardinality Ratio) แบ่งเป็น 3 ประเภท คือ

1. ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One-to-one Relationship) เป็นการแสดงความสัมพันธ์ของข้อมูลของเอนทิตี้หนึ่งว่า มีความสัมพันธ์กับข้อมูลอย่างมาก หนึ่งข้อมูลกับอีกเอนทิตี้หนึ่ง ในลักษณะที่เป็นหนึ่งต่อหนึ่ง

2. ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One-to-many Relationship) เป็นการแสดงความสัมพันธ์ของข้อมูลของเอนทิตี้หนึ่ง ว่ามีความสัมพันธ์กับข้อมูลหลายข้อมูลกับอีกเอนทิตี้หนึ่ง

3. ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many-to-many Relationship) เป็นการแสดงความสัมพันธ์ของข้อมูลของสองเอนทิตี้ในลักษณะแบบกลุ่มต่อกลุ่มความสัมพันธ์ระหว่างข้อมูลของสองเอนทิตี้เป็นแบบกลุ่มต่อกลุ่ม เป็นเรื่องที่ค่อนข้างจะยุ่งยากในการออกแบบฐานข้อมูลแบบเชิงสัมพันธ์ ซึ่งอาจมีปัญหาในด้านการซ้ำซ้อนและการปรับปรุงแก้ไขข้อมูล โดยทั่วไปจะแก้ปัญหาดังกล่าว โดยสร้างเอนทิตี้ใหม่ขึ้นมา เรียกว่า Gerund (Composite Entity หรือ Intersection Entity หรือ Synthetic Entity) เพื่อเป็นเอนทิตี้ที่เชื่อม ความสัมพันธ์กับสองเอนทิตี้เดิม โดยมีวัตถุประสงค์เพื่อปรับความสัมพันธ์ให้อยู่ในรูปหนึ่งต่อกลุ่ม

- **การทำรีเลชั่นให้อยู่ในรูปแบบบรรทัดฐาน**

แนวคิดในการทำรีเลชั่นให้อยู่ในรูปแบบบรรทัดฐาน (Normalization Process) คิดค้นโดย อี.เอฟ.คอดด์ (E.F.Codd) เป็นกระบวนการที่นำเค้าร่างของรีเลชั่นมาทำให้อยู่ในรูปแบบ ที่เป็นบรรทัดฐาน (Normal Form) เพื่อให้แน่ใจว่าการออกแบบเค้าร่างของรีเลชั่น เป็นการออกแบบที่เหมาะสม

ศิริลักษณ์ โรจนกิจอำนวย (2542 : 116-117) กล่าวถึงวัตถุประสงค์ของการทำรีเลชั่นให้เป็นบรรทัดฐาน คือ

1. เพื่อลดเนื้อที่ในการจัดเก็บข้อมูล การทำให้เป็นบรรทัดฐานเป็นการลดความซ้ำซ้อนของข้อมูลในรีเลชั่น ซึ่งทำให้ลดเนื้อที่ในการจัดเก็บข้อมูลได้

2. เพื่อลดปัญหาที่ข้อมูลไม่ถูกต้อง (Inconsistency) เนื่องจากข้อมูลในรีเลชั่น หนึ่งจะมีข้อมูลไม่ซ้ำกัน เมื่อมีการปรับปรุงข้อมูลก็จะปรับปรุงทูเพิลนั้นๆ ครั้งเดียว ไม่ต้องปรับปรุงหลายแห่ง โอกาสที่จะเกิดความผิดพลาดในการปรับปรุงไม่ครบถ้วนก็จะไม่เกิดขึ้น

3. เป็นการลดปัญหาที่เกิดจากการเพิ่ม ปรับปรุงและลบข้อมูล (Insert, Update and Delete Anomalies) ช่วยแก้ปัญหาที่อาจจะเกิดขึ้นจากการปรับปรุงข้อมูลไม่ครบ หรือข้อมูลหายไปจากฐานข้อมูลหรือการเพิ่มข้อมูล

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

- **รูปแบบบรรทัดฐาน**

ศิริลักษณ์ โรจนกิจอำนวย (2542:117-134) กล่าวถึงรูปแบบบรรทัดฐาน ที่ใช้ในการกำหนดแอททริบิวต์ที่เหมาะสมในรีเลชั่น ดังนี้ คือ

- **รูปแบบบรรทัดฐานขั้นที่ 1 (First Normal Form : 1NF)**

รีเลชั่นหนึ่งๆ จะอยู่ในรูปแบบบรรทัดฐานขั้นที่ 1 ก็ต่อเมื่อ ค่าของแอททริบิวต์ต่างๆ ในแต่ละทูเพิลจะมีค่าของข้อมูลเพียงค่าเดียว

- **รูปแบบบรรทัดฐานขั้นที่ 2 (Second Normal Form : 2NF)**

รีเลชั่นหนึ่ง ๆ จะอยู่ในรูแบบบรรทัดฐานขั้นที่ 2 ก็ต่อเมื่อ รีเลชั่นนั้นๆ อยู่ในรูปแบบบรรทัดฐานขั้นที่ 1 และมีคุณสมบัติอีกประการหนึ่งคือ แอททริบิวต์ทุกแอททริบิวต์ที่ไม่ได้เป็นคีย์หลัก จะต้องมีความสัมพันธ์ระหว่างค่าของแอททริบิวต์แบบฟังก์ชั่นกับคีย์หลัก (Fully Functional Dependency) กล่าวอีกนัยหนึ่งคือค่าของแอททริบิวต์ที่ไม่ได้เป็นคีย์หลักจะสามารถระบุค่าโดยแอททริบิวต์ที่เป็นคีย์หลัก หรือโดยแอททริบิวต์ทั้งหมดที่ประกอบกันเป็นคีย์หลักในกรณีที่คีย์หลักเป็นคีย์ผสม (ไม่มี Partial Dependency เกิดขึ้น)

- **รูปแบบบรรทัดฐานขั้นที่ 3 (Third Normal Form : 3NF)**

รีเลชั่นนั้น ๆ อยู่ในรูปแบบบรรทัดฐานขั้นที่ 2 และมีคุณสมบัติอีกประการหนึ่ง คือ แอททริบิวต์ที่ไม่ได้เป็นคีย์หลักไม่มีคุณสมบัติในการกำหนดค่าของแอททริบิวต์อื่นที่ไม่ใช่คีย์หลัก (ไม่มี Transitive Dependency เกิดขึ้น)

- **รูปแบบบรรทัดฐานของบอยส์และคอดด์ (Boyce/Codd Normal Form : BCNF)**

รีเลชั่นหนึ่งๆ จะอยู่ในรูปแบบบรรทัดฐานของบอยส์ และคอดด์ ก็ต่อเมื่อรีเลชั่นนั้นๆ อยู่ในรูปแบบบรรทัดฐานขั้นที่ 3 และไม่มีแอททริบิวต์อื่นในรีเลชั่นที่สามารถระบุค่าของแอททริบิวต์ที่เป็นคีย์หลักหรือส่วนหนึ่งส่วนใดของคีย์หลักในกรณีที่คีย์หลักเป็นคีย์ผสม

- **รูปแบบบรรทัดฐานขั้นที่ 4 (Fourth Normal Form : 4NF)**

รีเลชั่นจะอยู่ในรูปแบบบรรทัดฐานขั้นที่ 4 ก็ต่อเมื่อ รีเลชั่นนั้นๆ อยู่ในรูปแบบ BCNF และเป็นรีเลชั่นที่ไม่มีความสัมพันธ์ในการระบุค่าของแอททริบิวต์แบบหลายค่าโดยที่แอททริบิวต์ที่ถูกระบุค่าหลายค่าเหล่านี้ไม่มีความสัมพันธ์กัน (Independently Multivalued Dependency)

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

- **รูปแบบบรรทัดฐานขั้นที่ 5 (Fifth Normal Form : 5NF)**

รีเลชั่นหนึ่งๆ จะอยู่ในรูปแบบบรรทัดฐานขั้นที่ 5 หรือที่เรียกว่า Project-Join Normal Form (PJ/NF) ก็ต่อเมื่อ รีเลชั่นนั้นอยู่ในรูปแบบบรรทัดฐานขั้นที่ 4 และไม่มี Symmetric Constraint กล่าวคือ หากมีการแตกรีเลชั่นออกเป็นรีเลชั่นย่อย (Projection) และเมื่อทำการเชื่อมโยงรีเลชั่นย่อยทั้งหมดจะไม่ก่อให้เกิดข้อมูลใหม่ที่เหมือนรีเลชั่นเดิม (Spurious Topple)

ศิริลักษณ์ โรจนกิจอำนวย (2542:134-135) กล่าวถึงประเด็นที่ควรคำนึงถึง ในการทำให้เป็นรูปแบบบรรทัดฐาน คือ

1. การแตกรีเลชั่นมากเกินไป (Over normalization)

วัตถุประสงค์หนึ่งของการทำให้เป็นรูปแบบบรรทัดฐานก็คือ เพื่อลดปัญหาในด้านความซ้ำซ้อนของข้อมูล และลดปัญหาในเรื่องการเพิ่ม การปรับปรุง หรือลบข้อมูล โดยทั่วไปแล้วการออกแบบในระดับแนวคิด ผู้ออกแบบจะพยายามวิเคราะห์รีเลชั่นให้อยู่ในรูปแบบขั้นที่ 3 แต่ถ้ามีกรณีของปัญหาที่จำเป็นต้องทำต่อไปถึงรูปแบบบรรทัดฐานของบอยส์และคอดด์ หรือขั้นที่ 4 หรือขั้นที่ 5 แต่ไม่ควรแตกรีเลชั่นให้มากเกินความจำเป็น (Over normalization) เพราะการแตกรีเลชั่นออกเป็นรีเลชั่นย่อยมากเกินไป มีผลต่อประสิทธิภาพในการทำงานของฐานข้อมูลนั้นๆ เช่น ในการค้นหาข้อมูลจะใช้เวลามาก เป็นต้น

2. การดีนอร์มอลไลเซชั่น (Denormalization)

ในกรณีที่บางรีเลชั่นถูกออกแบบโดยการไม่ทำให้อยู่ในรูปแบบบรรทัดฐานที่เป็นไปตามกฎเกณฑ์ที่กำหนดไว้ เช่น รีเลชั่นนั้นๆ ควรจะปรับให้อยู่ในรูปแบบบรรทัดฐานขั้นที่สาม แต่หยุดอยู่เพียงรูปแบบบรรทัดฐานขั้นที่ 2 เป็นต้น ทั้งนี้ อาจเป็นเพราะเหตุผลในเรื่องของประสิทธิภาพในการเรียกดูหรือค้นหาข้อมูล และยอมให้เกิดความซ้ำซ้อนของข้อมูลได้ ด้วยเหตุที่การดีนอร์มอลไลเซชั่นอาจก่อให้เกิดปัญหาความซ้ำซ้อนของข้อมูลเกิดขึ้นได้ จึงควรมีการระบุสาเหตุ และวิธีการในการปรับปรุงข้อมูลในโปรแกรมประยุกต์ใช้งาน เพื่อป้องกันไม่ให้เกิดปัญหาข้อมูลไม่ถูกต้อง อีกประเด็นซึ่งอาจใช้เป็นแนวทางว่าจะยอมให้มีการดีนอร์มอลไลเซชั่นหรือไม่ก็คือ ถ้าข้อมูลในรีเลชั่นนั้นๆ ส่วนใหญ่จะเป็นการเรียกดูข้อมูลมากกว่า การเพิ่ม ปรับปรุง หรือลบข้อมูล ก็อาจจะดีนอร์มอลไลเซชั่นได้ถ้าคิดว่าการออกแบบลักษณะนี้จะเพิ่มประสิทธิภาพในการทำงานของฐานข้อมูล และไม่มีปัญหาด้านความไม่ถูกต้องของข้อมูลที่ซ้ำซ้อนกันได้

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

2.2.5 ระบบ Client/Server

สุชาย ธนวเสถียร และนรินทร์ อัครพิเชษฐ์ (2543 : 17) กล่าวว่า ระบบ Client - Server คือ สถาปัตยกรรมซอฟท์แวร์ที่ระบบซอฟท์แวร์ได้รับการออกแบบให้แยกออกเป็น 2 ส่วน ส่วนแรกเรียกว่า ส่วน Client และอีกส่วนเรียกว่า ส่วน Server ซอฟท์แวร์ส่วน Client ต้องสื่อสาร ติดต่อกับส่วน Server โดยที่ส่วนซอฟท์แวร์ Client จะขอใช้ข้อมูลจากซอฟท์แวร์ ส่วน Server ซอฟท์แวร์ส่วน Server จะตอบสนองโดยการดึงข้อมูลจากฐานข้อมูล แล้วส่งไปยังส่วน Client เพื่อการประมวลผลต่อไป

ประชา ตระการศิลป์ (2543 : 17-22) กล่าวถึง องค์ประกอบของ Client/Server วัตถุประสงค์ เป้าหมาย และหน้าที่ของ Client/Server ไว้ดังนี้

2.2.5.1 องค์ประกอบของ Client/Server

องค์ประกอบของการพัฒนาระบบงานประยุกต์ (Application Software Development) ในสถาปัตยกรรม แบบ Client/Server ประกอบด้วยกัน 3 องค์ประกอบ คือ

1. ไคลเอนต์ (Client) เรียกว่า ตัวลูก คือ เครื่องคอมพิวเตอร์ (พีซี) ที่ทำหน้าที่เป็นผู้รับ – ส่งข้อมูล ข่าวสาร และคำสั่งจากผู้ใช้ระบบงานไปให้แก่ Server (ตัวแม่) เพื่ออ่านข้อมูล ประมวลผลและส่งกลับมาให้ผู้ใช้

2. เซิร์ฟเวอร์ (Server) เรียกว่า ตัวแม่ คือ เครื่องคอมพิวเตอร์ (พีซี หรือ พีซีขนาดใหญ่) ที่ทำหน้าที่เป็นผู้รับ – ส่งข้อมูลข่าวสาร คำสั่งจาก Client เพื่ออ่านข้อมูลประมวลผล และส่งกลับมาให้ Client ซึ่ง Server 1 ตัวอาจจะมี Client ที่ต่อเชื่อมอยู่ในระบบงานได้หลายตัว และในแต่ละเครือข่ายอาจจะมี Server กี่ตัวก็ได้ตามความเหมาะสมของแต่ละระบบงาน

3. ระบบงานเครือข่าย (Network) คือ ระบบงานที่ประกอบไปด้วยอุปกรณ์ฮาร์ดแวร์ และซอฟต์แวร์เพื่อเป็นทางเดินให้กับข้อมูล ข่าวสาร คำสั่งโปรแกรมที่มีการรับ - ส่ง ระหว่าง Client กับ Server ที่ต่อเชื่อมโยงกัน

2.2.5.2 วัตถุประสงค์ของสถาปัตยกรรมแบบ Client/Server

- ลดงบประมาณในการลงทุน (Cost Saving)
- เพิ่มผลผลิตในการประมวลผล (Increased Productivity)
- สามารถขยายระบบงานได้ (Flexibility)
- สามารถใช้ทรัพยากรที่มีอยู่อย่างมีประสิทธิภาพ (Resource Utilization)

2.2.5.3 เป้าหมายของการพัฒนาระบบงานแบบ Client/Server

เป้าหมายของการพัฒนาระบบงานแบบ Client/Server คือ พยายามกำหนด หรือออกแบบให้ผู้ใช้ระบบงาน (End User/Client) รับผิดชอบงานในส่วนของงานทางด้านการประมวลผล และคำสั่งโปรแกรมต่าง ๆ โดยสามารถที่จะควบคุม สั่งการประมวลผล และการใช้ทรัพยากรที่มีอยู่ทั้งหมดในเครือข่ายได้อย่างมีประสิทธิภาพโดยไม่ต้องคำนึงถึงอุปกรณ์หรือแพลตฟอร์มที่แตกต่างกัน

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

2.2.5.4 การแบ่งหน้าที่ของ Client/Server

หน้าที่การทำงานการประมวลผลงานของเทคนิคแบบ Client/Server สามารถแบ่งออกเป็น 2 ส่วนคือ

1. Application Tasks

งานในส่วนของระบบงานประยุกต์นั้น สามารถแบ่งออกได้อีก 6 งาน คือ

(1) User Interface หมายถึง งานของโปรแกรมประยุกต์ (Application Program) ในส่วนที่ผู้ใช้งาน (End-User) เรียกใช้ข้อมูล เช่น คำสั่งโปรแกรมที่ผู้ใช้สั่งให้เครื่องคอมพิวเตอร์รับคำสั่ง เป็นต้น

(2) Presentation Logic หมายถึง การแสดงผลลัพธ์ที่เกิดขึ้นบนจอภาพ จากการที่ผู้ใช้งานบันทึกคำสั่ง ให้เครื่องคอมพิวเตอร์ทำงาน

(3) Application logic หมายถึง การประมวลผลที่เกิดขึ้นจากการที่ผู้ใช้งานบันทึกคำสั่งให้เครื่องคอมพิวเตอร์ทำงานตามโปรแกรมที่ได้กำหนด

(4) Data Requests and Results Acceptance หมายถึง ส่วนของงานที่จะแสดงให้ผู้ใช้ระบบงานรับทราบว่า เครื่องคอมพิวเตอร์ได้รับทราบคำสั่ง หรือได้แสดงผลลัพธ์ของการทำงานแล้ว

(5) Data Integrity หมายถึง ส่วนของโปรแกรม ที่ทำหน้าที่ตรวจสอบความเป็นไปได้ของข้อมูล (Validation) ความปลอดภัย และความครบถ้วนสมบูรณ์ของข้อมูล

(6) Physical Data Management หมายถึงโปรแกรมที่ทำหน้าที่ในการปรับปรุง แก้ไข อ่าน ลบทิ้ง เพิ่มเติม หรือ จัดการเก็บข้อมูลทางด้านกายภาพ

โดยทั่วไปการออกแบบระบบงานแบบ Client/Server จะกำหนดให้เครื่องคอมพิวเตอร์แม่ข่าย (Server) ทำหน้าที่งานในส่วนของการบริหารจัดการข้อมูลการบริหารจัดการเครือข่าย ส่วนอื่นที่เหลือทั้งหมดจะให้เป็นหน้าที่ของเครื่องคอมพิวเตอร์ลูกข่าย เพื่อที่จะได้ประสิทธิภาพของการทำงานที่สูงสุด โดย Client จะรับงานการประมวลผลข้อมูล ดังนั้น ถ้ามีการเพิ่มขยายเครือข่าย หรือ Client มากขึ้น งานที่เพิ่มขึ้นมา จะอยู่ที่ Client เกือบทั้งหมด โดยที่ Server จะมีงานเพิ่มเพียงคำสั่งโปรแกรมจาก Client ที่เพิ่มขึ้นมาเท่านั้น

ในการออกแบบโปรแกรมประยุกต์ หน้าที่ของ Client ในการประมวลผลนั้น ควรจะต้องกำหนดให้ Client ทำหน้าที่ตรวจสอบความเป็นไปได้ของข้อมูลที่ผู้ใช้บันทึกเข้ามาตรวจสอบผิดพลาดของข้อมูลเพื่อป้องกันกลั่นกรองไม่ให้ Client ส่งข้อมูลที่ผิดๆ ไปให้ Server ทำงานซึ่งมีผลทำให้ลดภาวะงานของ Server และปริมาณงานบนเครือข่ายลง มีผลถึงประสิทธิภาพที่สูงขึ้นด้วย

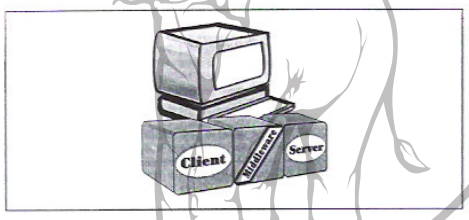
ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

2. Rightsizing

คือ การวิเคราะห์ และออกแบบให้ระบบงานสามารถแบ่งหน้าที่ การทำงานการประมวลผลงานออกเป็นส่วนๆ และให้แต่ละส่วนของงานนั้นทำการประมวลผลที่ Client หรือ Server ที่เหมาะสมได้ครบถ้วนอย่างอิสระโดยเป็นหน้าที่ภารกิจของนักวิเคราะห์และออกแบบระบบงานฯ (Systems Analyst/System Designer) ที่จะต้องใช้ความรู้ และประสบการณ์เพื่อพิจารณาว่างานของแต่ละโปรแกรมจะมีความเหมาะสมมากที่สุดในการประมวลผลที่ Server หรือที่ Client เพื่อให้เกิด และประสิทธิภาพประสิทธิผลสูงสุด

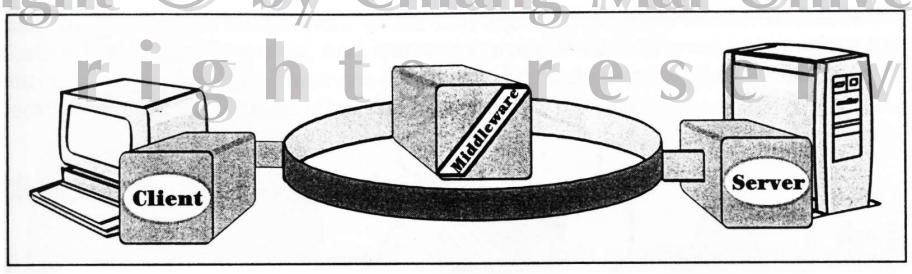
2.2.5.5 รูปแบบของ Client/Server ที่ใช้งานจะมีอยู่ 4 ชนิดด้วยกัน คือ

1. Stand alone Client/Server การทำงานแบบนี้ ผู้ให้บริการหรือ Server จะอยู่บนเครื่องเดียวกับผู้ขอใช้บริการหรือ Client ทำให้มีความเร็วในการติดต่อสื่อสารระหว่าง ผู้ให้บริการและผู้ใช้บริการสูงมาก แต่ประสิทธิภาพในการประมวลผลระบบฐานข้อมูลจะลดลงบ้าง ระบบนี้เรียกอีกอย่างว่า Tiny Client/Server



รูปที่ 2.5 แสดงรูปแบบของ Client/Server แบบ Stand alone Client/Server

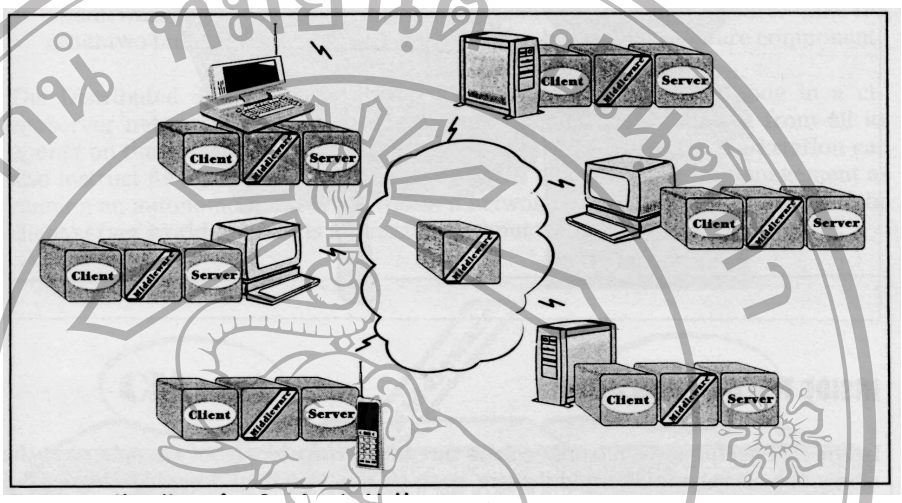
2. Department Client/Server หรือ LAN based single server การทำงาน แบบนี้จะมีผู้ให้บริการเกี่ยวกับฐานข้อมูล แอพพลิเคชั่น ฯลฯ อยู่บนเครื่อง Server และผู้ขอใช้บริการทั้งหลาย จะอยู่บนเครื่อง Client โดยจะเชื่อมต่อกันด้วยระบบท้องถิ่น (LAN) และมีมิดเดิลแวร์ (Middleware) เป็นตัวกลางที่ทำงานอยู่ระหว่าง Client และ Server ดังรูปที่ 2.6 การติดต่อสื่อสารกันระหว่างผู้ให้บริการและผู้ขอใช้บริการจะช้ากว่าแบบ stand alone เพราะจะต้องติดต่อผ่านระบบเครือข่าย ยิ่งถ้ามีผู้ขอใช้บริการเข้ามาดึงข้อมูลกันครั้งละมากๆ หลายๆ เครื่อง ประสิทธิภาพจะลดลงอย่างเห็นได้ชัด วิธีเพิ่มประสิทธิภาพก็คือ การเพิ่มเครื่อง Server ขึ้นในระบบ



รูปที่ 2.6 แสดงรูปแบบของ Client/Server แบบ Department Client/Server

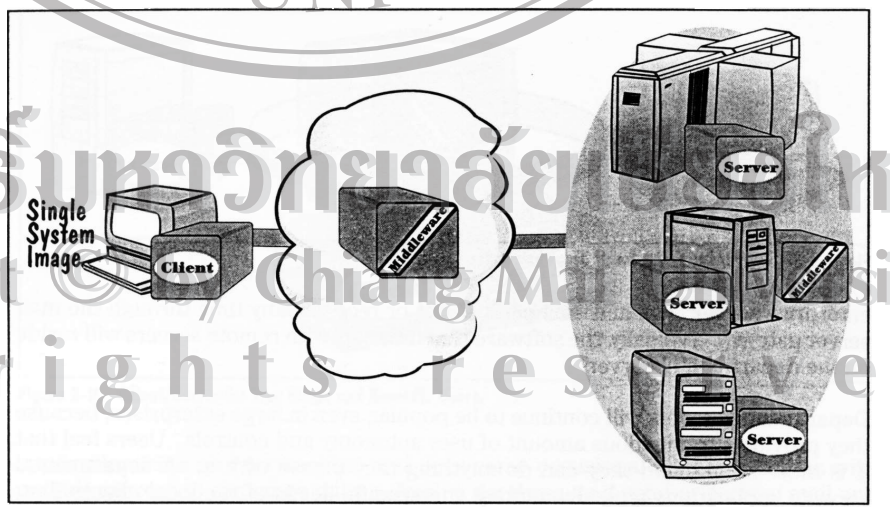
ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

3. Workgroups Client/Server การทำงานแบบ workgroups นี้ จะเป็นกลุ่มของ Server ที่หลากหลายแพลตฟอร์ม หลายผู้ผลิต มีความแตกต่างกันของ Server แต่ทั้งหมดนี้ จะเชื่อมต่อกันทางระบบเครือข่าย LAN และ WAN และใช้เป็น Middleware มาตรฐานในการทำงาน



รูปที่ 2.7 แสดงรูปแบบของ Workgroups Client/Server

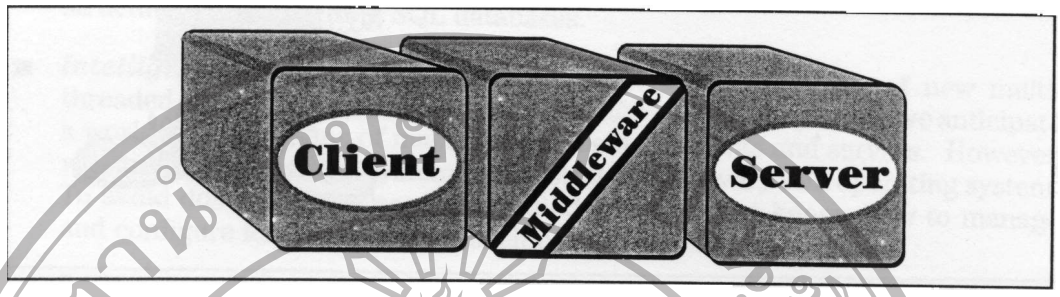
4. Enterprise Client/Server การทำงานแบบ Enterprise หรือระดับองค์กรจะทำให้มีการเชื่อมโยงเครื่อง Server หรือ Host ต่าง แพลตฟอร์ม เข้าด้วยกัน ทำให้มีการใช้ทรัพยากรบนระบบได้อย่างมีประสิทธิภาพสูงสุด โดยที่ Client สามารถที่จะเลือกใช้ทรัพยากร ฐานข้อมูลจาก Server เครื่องใดก็ได้ผ่านทางมิดเดิลแวร์ ดังรูป



รูปที่ 2.8 แสดงรูปแบบของ Enterprise Client/Server

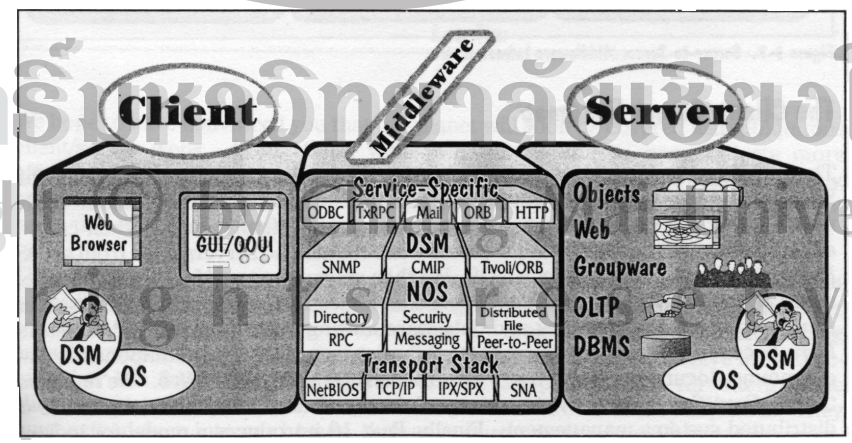
ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

2.2.5.6 โครงสร้างพื้นฐานของ Client/Server



รูปที่ 2.9 แสดงโครงสร้างพื้นฐานของ Client/Server

จากการพัฒนาของระบบปฏิบัติการเน็ตเวิร์ก ทั้งทางค่ายไมโครซอฟท์ คือ Windows NT Server และ โนเวลคือ Netware ทำให้การทำงานบนบนระบบเน็ตเวิร์กเป็นไปอย่างสะดวกรวดเร็วและกว้างขวางขึ้น โดยเฉพาะโปรโตคอลแบบ Ethernet ที่ใช้ในปัจจุบัน ซึ่งเหมาะกับระบบ Client/Server อีกทั้งระบบปฏิบัติการดอส ที่รันอยู่บนเครื่อง Client ก็มีความสามารถ ทางเน็ตเวิร์กและการทำงานแบบมัลติเทรด Windows NT Server เป็นระบบปฏิบัติการ 32 บิต ที่ทำงานแบบพรีเอมพ์ทีพ มัลติทาสก์กิ้ง (Preemptive Multitasking) และมัลติเทรดดิ้ง(Multithreading) ซึ่งจะแยกงานของแอพพลิเคชั่นแต่ละตัวออกจากกัน ทำให้ไม่มีการรบกวนการทำงานระหว่างกัน เพราะโมเดลในการออกแบบ Windows NT Server ก็เป็นสถาปัตยกรรม Client/Server อยู่แล้ว จากรูปที่ 2.9 จะเห็นว่าโครงสร้างพื้นฐานของ Client/Server ถูกแบ่งออกเป็น 3 ส่วน คือ Client, Middleware และ Server



รูปที่ 2.10 แสดงการทำงานของระบบ Client/Server

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

- **Client** เป็นส่วนที่จะรันแอพพลิเคชั่นบน Client โดยจะใช้ระบบ GUI (Graphical User Interface)หรือ OOUI (Object Oriented User Interface) หรือ DSM (Distributed System Management) เป็นการติดต่อกับ User ผ่านกราฟฟิกส์ซึ่งทำงานแบบเชิงวัตถุ (Object)

- **Middleware** เป็นส่วนที่ทำงานอยู่ระหว่าง Client/Server เป็นเสมือนสะพานเชื่อมต่อการทำงาน สามารถแบ่งออกเป็น 4 แบบ ดังนี้คือ

1. ***Service Specific*** หรือการบริการโดยเฉพาะ จะขึ้นอยู่กับการใช้แอพพลิเคชั่นในการทำงาน เช่น แอพพลิเคชั่นของอ๊อบเจ็กต์แบบกระจาย จะใช้มิดเดิลแวร์ ORB (Oject Reguest Broker) แอพพลิเคชั่นกรุ๊ปแวร์ จะใช้มิดเดิลแวร์ Mail และ TP monitor จะใช้มิดเดิลแวร์ TxRPC (Transectional Remote Procedure call) ส่วนระบบฐานข้อมูล SQL จะใช้ ODBC (Open Database Connectivity) DRDA (Distribute Relationnal Database Architecture) ของ IBM,RDA (Remote Database Access),Oracle Glue,CLI (Call-level Interface)

2. ***DSM (Distributed System Management)*** จะรันบนทุกโหนดของระบบเน็ตเวิร์กที่เป็น Client/Server จะมีมิดเดิลแวร์ SNMP (Simple Network Management Protocol),CMIP (Common Management Information Protocol) และ DME

3. ***NOS (Network Operating System)*** เป็นระบบปฏิบัติการเน็ตเวิร์ก ซึ่งให้บริการทั่วๆไป โดยจะมีทั้ง Directory Service,Naming Service,Security/Authentication Service, Messaging Service,Distributed file,PRC,Peer to Peer ฯลฯ ระบบปฏิบัติการเหล่านี้ เช่น Windows NT server,Netware,Banyan Vines,OSF DCE

- NOS จะช่วยให้การใช้ชื่อ (Namespace) เพียงชื่อเดียวสามารถเข้าถึงทรัพยากรต่าง ๆ บนระบบเน็ตเวิร์กรวมได้

- NOS จะทำให้ผู้ใช้งาน (User) ไม่ต้องรับรู้เกี่ยวกับความผิดพลาดที่เกิดขึ้น เช่น การรับ - ส่งข้อมูลผิดพลาด ระบบเน็ตเวิร์กมีปัญหา หรือมีการเคลื่อนย้ายทรัพยากรจาก ไดเรกทอรีไปยังเซิร์ฟเวอร์ NOS จะแก้ไขและอับเดทข้อมูลต่าง ๆ ให้เป็นหนึ่งเดียวทั้งระบบ

- NOS ทำให้สามารถใช้รหัสเพียง 1 ชุด เข้าสู่ระบบเน็ตเวิร์กจากเครื่องใดที่ไหนก็ได้ โดยจะใช้ระบบรักษาความปลอดภัยแบบ DCE (Distributed Computing Environment) ในการตรวจสอบ

- NOS จะมีระบบไดเรกทอรีแบบ Global Directory ซึ่งจะนำคนแอพพิเคชั่น โปรแกรม สิ่งต่างๆ เข้ามาทำงานร่วมกัน ทำให้ไม่ต้องขึ้นกับสถานที่ สามารถเปลี่ยนสถานที่ในการเข้าใช้ทรัพยากรได้

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

- NOS จะจัดการในเรื่อง Distributed time ให้ทั้งระบบคือจะมีการซิงโครไนซ์ในเรื่องเวลาระหว่างเซิร์ฟเวอร์และไคลเอ็นต์ทุกตัว

- NOS จะจัดการในเรื่อง Distributed Security อย่างต่ำจะอยู่ในระดับ C2 ซึ่งจะต้องมีการแสดงตน (Authentication) มีการเข้ารหัสผ่าน (Encrypt) ใช้มาตรฐาน Kerboros และแอพพิเคชั่นเซิร์ฟเวอร์จะมีอำนาจ (Athorization) ในการใช้ ACLs (Access Control Lists) เพื่อควบคุมการเข้าใช้ทรัพยากรจาก User

- NOS สามารถจะใช้ มิดเดิลแวร์ MOM (Message Oriented Middleware) ในการช่วยจัดคิว ข้อความ (Message queue) เพื่อให้ทั้งไคลเอ็นต์ และเซิร์ฟเวอร์ยังคงทำงานได้อย่างต่อเนื่องแม้จะมีปัญหาทางระบบสื่อสาร ลักษณะนี้อาจเรียกว่า Loosely-Coupled queue based และอีกรูปแบบ คือการใช้ RPCs (Remote Procedure Calls) ซึ่ง NOS เหล่านี้คือ OSF/DCF, ONC/SUN, Netware 4.xx/Novell

4. ***Transport stack*** เป็นบริการพื้นฐานในการสื่อสารระหว่าง Client และ Server บนระบบ LAN และ WAN โปรโตคอลหลัก ๆ ในส่วนของ Transport stack มีอยู่ 4 ตัวด้วยกันคือ NetBIOS, TCP/IP, IPX/SPX และ SNA

- NetBIOS เป็นโปรโตคอลที่ออกแบบโดย บริษัท ไอบีเอ็มฯ ให้ใช้งานเครือข่ายขนาดเล็ก ต่อมาพัฒนาเป็น NetBEUI (NetBIOS Extended User Interface) สามารถจะใช้งานกับระบบเครือข่าย ที่มีเครื่องคอมพิวเตอร์ตั้งแต่ 20 - 200 เครื่อง ไม่สามารถใช้งานกับเครือข่ายที่มี ขนาดใหญ่ได้ และไม่สามารถค้นหาเส้นทางได้ จะเห็นว่าใช้กับงานเวิร์กกรุ๊ปเช่น Windows for workgroups หรือ Microsoft LAN Manager โปรโตคอล NetBIOS จะทำงานอยู่ในชั้นของ Session Layer ตามมาตรฐาน OSI-7 Layer

- TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นโปรโตคอล ที่ใช้งานบนระบบ UNIX พัฒนาขึ้นในปี 2512 โดยกระทรวงกลาโหมของสหรัฐอเมริกามีเครือข่าย ชื่อ ARPANET (Advanced Research Project Agency Network) สำหรับใช้งานกับเครือข่ายขนาดใหญ่อย่าง WANs มีความสามารถในการค้นหาเส้นทาง และมีความยืดหยุ่นในการทำงานสูง

- IPX/SPX (Internetwork Packet Exchange/Sequenced Packet Exchange) เป็นโปรโตคอลหลักของระบบปฏิบัติการเน็ตเวิร์ก Netware มีความฉลาดในการทำงานกว่า NetBIOS คือสามารถค้นหาเส้นทางได้ ทำให้โปรโตคอล IPX/SPX สามารถจะทำงานบนระบบ LAN และ WAN ได้ (แต่การทำงานบนระบบ WAN เช่น อินเตอร์เน็ตยังสู้โปรโตคอล TCP/IP ไม่ได้)

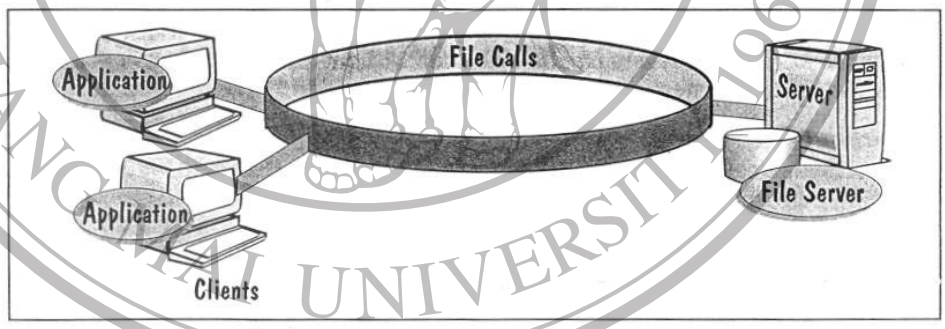
- SNA (System Network Architecture) เป็นโปรโตคอลที่ออกแบบโดย บริษัท ไอบีเอ็มฯ เพื่อใช้งานบนระบบเครือข่ายเครื่องเมนเฟรมของไอบีเอ็ม

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

- **Server** เป็นส่วนหนึ่งจะรับแอพพิเคชั่นในการจัดการทรัพยากรต่างๆ สำหรับระบบ Client/Server ซึ่งสามารถแบ่งออกได้ 6 แบบด้วยกัน คือ

1. ระบบแฟ้มข้อมูล (File)
2. ระบบฐานข้อมูล SQL (DBMS)
3. ระบบจัดการทรานแซคชั่น (TP monitor)
4. ระบบกรุ๊ปแวร์ (Groupware)
5. ระบบอ๊อบเจ็กต์แบบกระจาย (Distributed objects)
6. ระบบเครือข่ายใยแมงมุม (Web)

1. ระบบแฟ้มข้อมูล (File Server) ในส่วนของ File Server ฝั่ง Client (ชนิดของ PC) จะทำการร้องขอแฟ้มข้อมูล ผ่านระบบเครือข่ายไปยัง File Server (ตามรูป) ระบบนี้เป็นรูปแบบเดิมที่สุดในการต่อเชื่อมระบบเครือข่ายและการให้บริการข้อมูล File Server จะเป็นผู้ที่แบ่งปัน กระจายแฟ้มข้อมูลผ่านเครือข่ายระบบนี้จะจำเป็นในการกระจายเอกสารที่รวมไว้ รวมทั้งรูปภาพ ภาพลายเส้น และข้อมูลขนาดใหญ่อื่นๆ

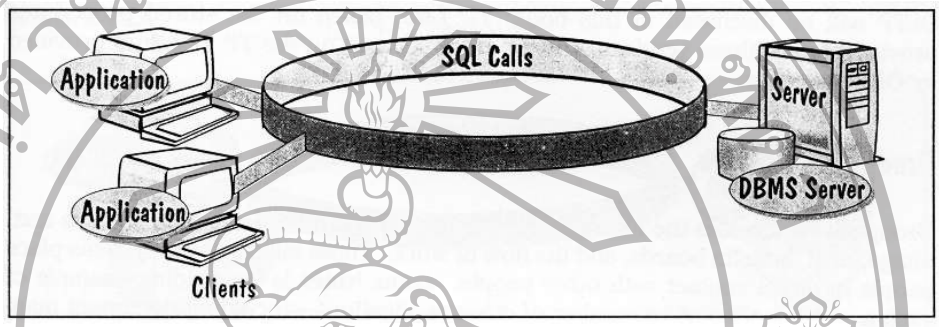


รูปที่ 2.11 แสดงระบบแฟ้มข้อมูล File Server

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

2. ระบบฐานข้อมูล (Database Server) การประมวลผลฐานข้อมูลในระบบ Client/Server จะมีสองส่วนคือ แบ็คเอ็นต์ (Back end) และฟร๊อนเอ็นต์ (Front end) ซึ่งแบ็คเอ็นต์ จะเป็นส่วนของ Server ซึ่งจะมีระบบฐานข้อมูล SQL เช่น Microsoft SQL Sever ทำหน้าที่เก็บข้อมูล จัดเรียงลำดับ ค้นหา เรียกใช้ ป้องกันข้อมูล ฯลฯ ส่วนฟร๊อนเอ็นต์ จะเป็นส่วนของ Client มีหน้าที่คอยจัดเตรียมแสดงผลข้อมูลซึ่งมีเครื่องมือในการสร้างฟร๊อนเอ็นต์ บน Windows เช่น Visual Basic, Power Builder, SQL Windows, Delphi ฯลฯ ด้วยเหตุที่ SQL เป็นภาษาในเชิงสอบถาม อธิบายแบบมีโครงสร้าง คำสั่งที่ใช้ก็อ่านเข้าใจง่าย โปรเซสการทำงานเริ่มจาก User ส่งคำสั่งเข้าไปเพื่อขอใช้บริการ ผู้ให้บริการรับคำสั่งมาทำการประมวลผลเสร็จแล้วส่งผลลัพท์กลับไปให้ User ผู้ขอใช้

บริการ จะเห็นว่ามีเพียงผลลัพธ์ที่ User ต้องการเท่านั้นถูกส่งออกไปบนระบบเน็ตเวิร์กเป็นการลด Traffic ของระบบลงด้วย มีส่วนของชุดคำสั่ง ภาษา SQL เรียกว่า Stored procedure สำหรับจัดการข้อมูล นอกจากนี้ยังมี Triggers, Rules, Views และ Scroll cursor จะช่วยทำให้ข้อมูลมีความถูกต้องและสมบูรณ์ที่สุด



รูปที่ 2.12 แสดงระบบฐานข้อมูล Database Server

3. ระบบจัดการทรานแซคชั่น (Transaction Server) การทำงานบนเครื่องเมนเฟรม ทั้งระบบ จะมีความสลับซับซ้อนของโปรแกรมมาก จะต้องมี TP monitor (Transaction Processing monitor) ซึ่งเป็นระบบติดตามการประมวลผลทรานแซคชั่นอยู่ในระบบด้วย ซอฟต์แวร์ประเภท TP monitor ที่นิยมใช้กันคือ CICS (IBM), Tuxedo(BEA) และ Encina สำหรับระบบ Client/Server การนำ TP Monitor มาใช้งานนับว่ามีประโยชน์ และเพิ่มประสิทธิภาพในการประมวลผลแบบกระจาย (Distributed processing)

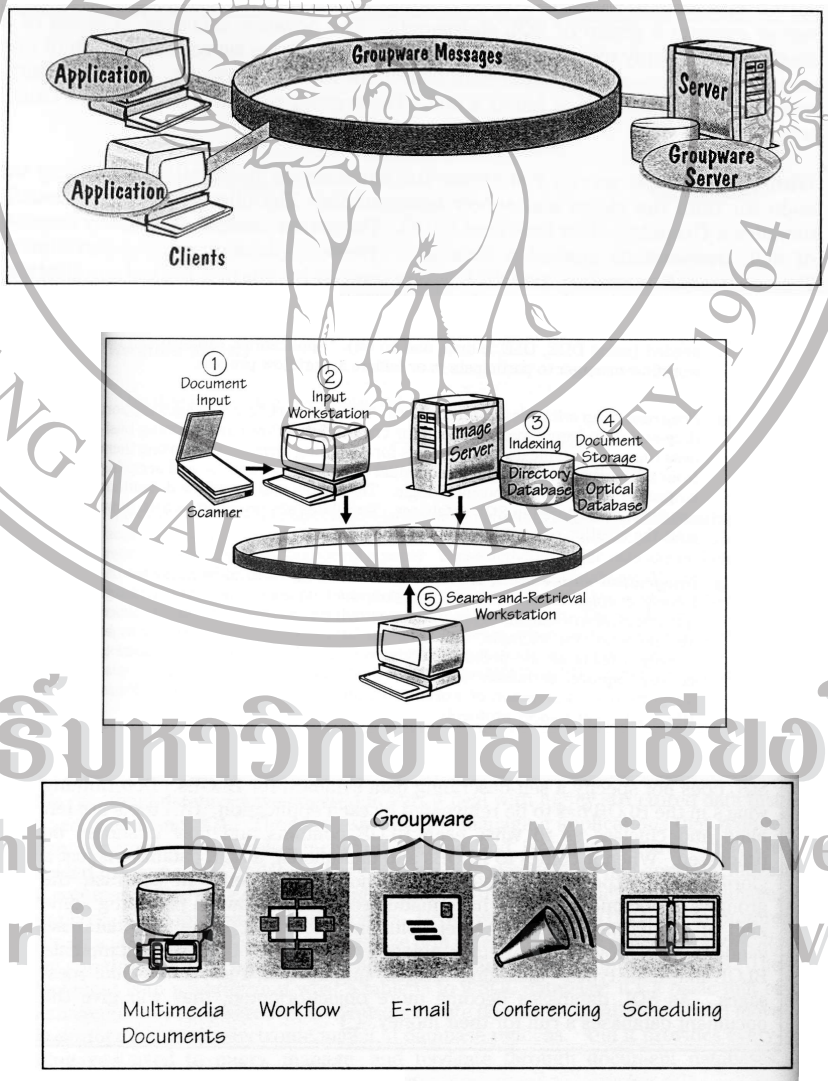
ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

รูปที่ 2.13 แสดงระบบจัดการทรานแซคชั่น Transaction Server

TP monitor จะจัดการกับทรานแซคชั่น โดยการจัดเส้นทางเดินในระบบให้จาก Client ไปยัง Server ตัวใดตัวหนึ่งบนระบบ แล้วกลับมาที่เดิม ถ้าไม่ประสบความสำเร็จ ก็จะเริ่มต้นทำงานใหม่ งานที่ TP monitor ทำคือ การจัดการทรัพยากร และ User Request จัดการเรื่องของ Two phase

commit เก็บ Log ของทรานแซคชั่นไว้ จะเห็นว่า TP monitor มีส่วนในการจัดการควบคุมทรานแซคชั่นและทรัพยากรในระบบด้วย สำหรับระบบ NOS ของ Windows NT Server ทาง บริษัท ไมโครซอฟท์ ได้ออกแบบ MTS (Microsoft Transaction Server) ให้ทำงานร่วมกับฐานข้อมูล Microsoft SQL Server อย่างมีประสิทธิภาพ นอกจากนี้เรายังสามารถจะใช้ซอฟต์แวร์ TP monitor ของเมนเฟรมเช่น Tuxedo ให้มาทำงานบนแพลตฟอร์ม Windows NT Server ได้อีกด้วย

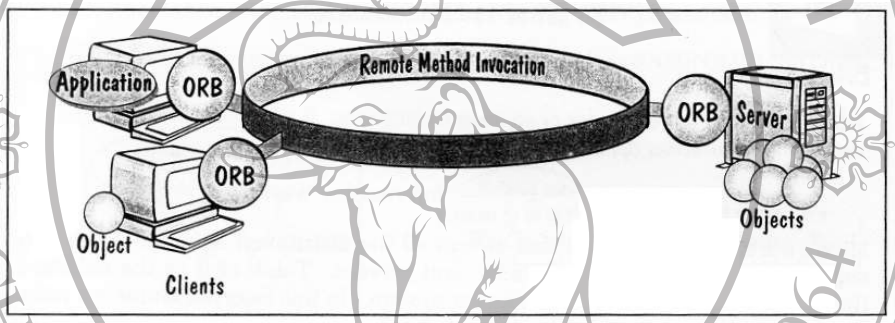
4. ระบบกรุ๊ปแวร์ (Groupware Servers) กรุ๊ปแวร์ เป็นเทคโนโลยีบนระบบแมนเฟรมซึ่งมีหลายส่วนทำงานร่วมกัน เช่น ระบบ E-mail ระบบเวิร์กโฟลว์ ระบบจัดการเอกสารแบบมัลติมีเดีย ระบบจัดการรูปภาพ ระบบจัดตารางเวลา ระบบการประชุม แต่สามารถนำมาใช้งานกับระบบ Client/Server ได้ ซอฟต์แวร์ประเภทนี้คือ Lotus Notes, DCA OpenMind, Image Plus/2



ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

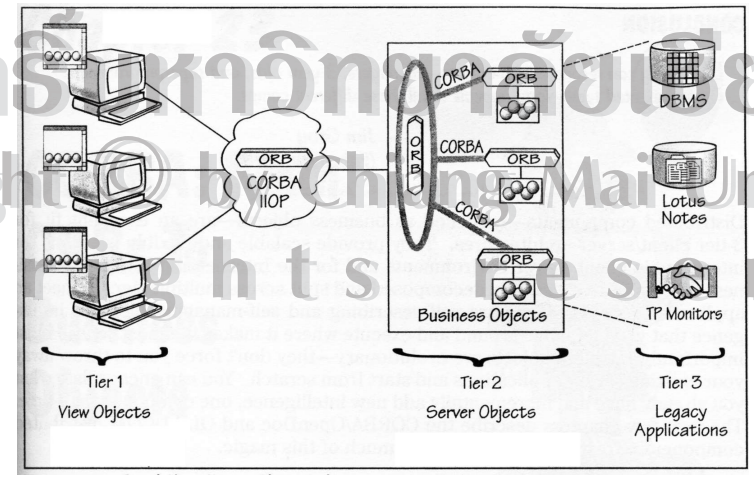
รูปที่ 2.14 แสดงระบบกรุ๊ปแวร์ Groupware Servers

5. ระบบอ๊อบเจ็กต์แบบกระจาย (Object Server) เทคโนโลยีอ๊อบเจ็กต์แบบกระจาย (Distributed Object) ช่วยให้ระบบ Client/Server แบบนี้นำหน้าระบบ Client/Server ทั้ง 3 แบบ (ระบบฐานข้อมูล, SQL,ระบบ TP monitor และระบบกรุ๊ปแวร์) ด้วยการรวบรวมขั้นตอนในการทำงานและข้อมูลไว้ในอ๊อบเจ็กต์ ซึ่งมันจะเดินทางไปบนระบบเน็ตเวิร์ก (ได้ทุกหนทุกแห่ง) และยังทำงานแบบมัลติแพลตฟอร์มอีกด้วยเช่น AIX, SUN-solaris, HP-UX, IBM-MVS,Digital UNIX Open VMS,Linux ฯลฯ กลุ่ม OMG(Object Management Group) ได้ออกแบบอ๊อบเจ็กต์สถาปัตยกรรม CORBA(Common Object Reguest Broker Architecture) ให้ใช้งานนานร่วม 8 ปี พร้อมทั้งมีบริษัทที่ใช้งาน CORBA อยู่ร่วม 600 บริษัท การทำงานของ CORBA แสดงดังรูปที่ 2.15



รูปที่ 2.15 แสดงระบบอ๊อบเจ็กต์แบบกระจาย Object Server

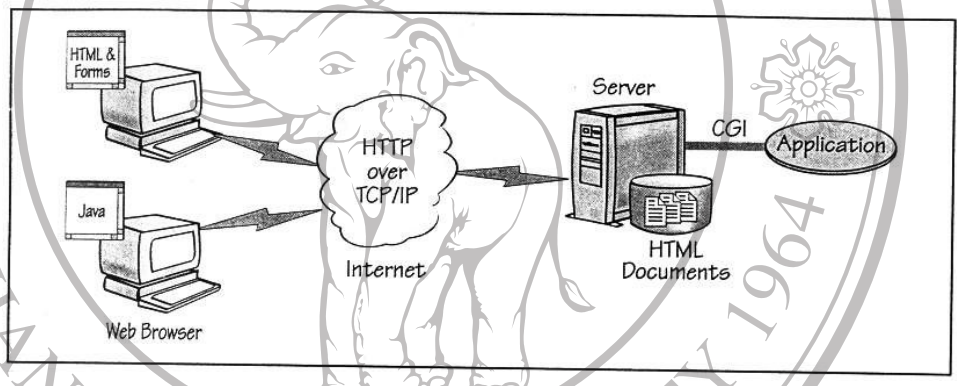
คู่แข่งสำคัญอย่าง บริษัท ไมโครซอฟท์ ส่ง DCOM (Distributed Common Object Model) ออกมาทำตลาด โดยเริ่มจากแพลตฟอร์ม บน Windows 95 และ Windows NT นอกจากนี้ กำลังจะพอร์ตข้ามไปยัง NVS, UNIX, Solaris, AIX, SCO UNIX, HP-UX, Linux,Open VMS



รูปที่ 2.16 แสดง 3-Tiered Client/Server, Object – Style.

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

6. ระบบ โครงข่ายใยแมงมุม (Web Servers) World Wide Web เป็นรูปแบบของ Client/Server Application แบบแรกในโลกปัจจุบัน รูปแบบใหม่ของ Client/Server นี้ประกอบไปด้วย แบบผอม สะดวกเคลื่อนย้าย เป็นเอกภาพ ฝั่ง client ที่พูดถึง super fat server ในรูปแบบที่ง่ายที่สุด Web Server จะคืนเอกสารเมื่อฝั่ง client ร้องขอโดยใช้ชื่อ (ตามรูปที่ 2.17) ทั้งฝั่ง client และฝั่ง server ติดต่อกันโดยใช้ RPC ซึ่งเป็นข้อตกลง (Protocol) ที่เรียกว่า HTTP ข้อตกลงนี้กำหนดโดยส่วนของคำสั่งง่ายๆ ตัวแปร (parameter) จะถูกส่งผ่านเหมือน string ที่ซึ่งไม่มีข้อกำหนดของชนิดของข้อมูล Web กำลังจะเตรียมถูกขยายในรูปแบบ interactive ของคอมพิวเตอร์รูปแบบของ Client/Server แต่อย่างไรก็ตาม Web และ distributed objects ก็ยังเป็นอะไรที่เริ่มและมาพร้อมกัน Java เป็นที่ประจักษ์แรกสำหรับ Object Web แบบใหม่



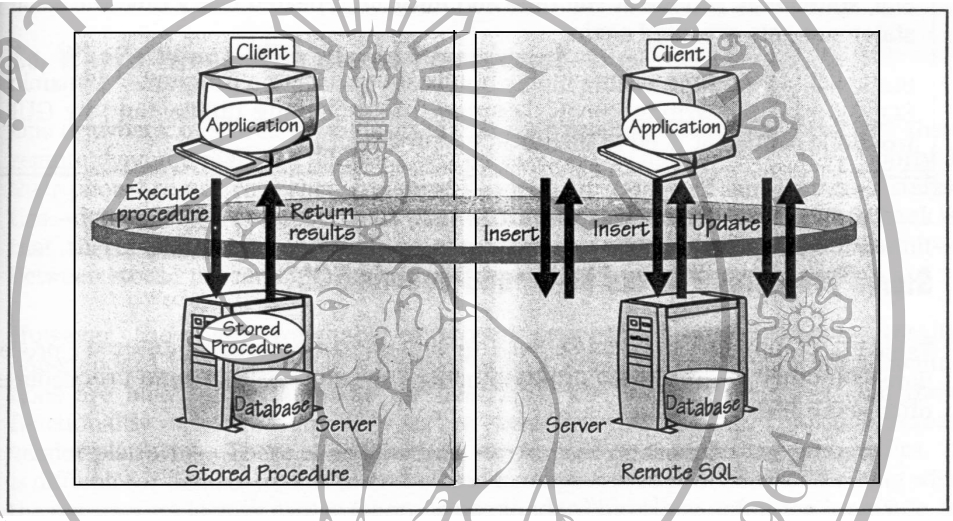
รูปที่ 2.17 แสดงระบบ โครงข่ายใยแมงมุม Web Servers

2.2.5.7 ฐานข้อมูลระบบ Client/Server

เมื่อกล่าวถึง Client/Server ก็จะนึกถึงเรื่องของการประมวลผลแบบกระจาย ที่ช่วยให้ระบบงานภายในองค์กรมีประสิทธิภาพขึ้น แล้วระบบฐานข้อมูลที่ใช้งานอยู่เป็นแบบใด จากอดีตที่เครื่องไมโครคอมพิวเตอร์ทำงานแบบ Stand-Alone ไม่ยุ่งเกี่ยวกับใคร จนมีการสร้างระบบเน็ตเวิร์ก LAN ขึ้นมาเพื่อเชื่อมโยง-สื่อสารกันระหว่างเครื่องคอมพิวเตอร์ มีการแบ่งกันใช้ทรัพยากร ไม่ว่าจะเป็นไฟล์ เครื่องพิมพ์ ฯลฯ สิ่งเหล่านี้จะเกิดขึ้นและทำงานบนโปรโตคอล IPX/SPX ของ Netware ซึ่งเป็นโปรโตคอลที่มีความสามารถในการทำงานและค้นหาเส้นทางได้ โปรแกรมฐานข้อมูลจึงพากันพัฒนาระบบของตน ให้ทำงานบนโปรโตคอล IPX/SPX ซึ่งจะเป็นการออกแบบฐานข้อมูลแบบกระจาย (Distributed Database) ที่รันบนเซิร์ฟเวอร์ระดับ Low-end ที่ใช้ไมโครโปรเซสเซอร์ของ Intel (80X86,Pentium) Alpha (R4X00) Motorola(68X00) ส่วนเครื่องในระบบ Mid-rang และ High-end จะใช้โปรโตคอล TCP/IP ที่มีความสามารถในการเชื่อมต่อ

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

ระบบเน็ตเวิร์ก LAN และ WAN ซึ่งทำให้การสื่อสารข้ามทวีปเป็นเรื่องง่าย ระบบฐานข้อมูลมีการพัฒนาในรูปแบบ DBMS (Database Management System) RDBMS (Relational DBMS) และ DDBMS(Distributed DBMS) ในปัจจุบันเครื่องเซิร์ฟเวอร์ Low-end สามารถจะรันบนโปรโตคอล TCP/IP ได้อย่างมีประสิทธิภาพ ที่เห็นได้ชัดเจนคือการเป็นเว็บเซิร์ฟเวอร์



รูปที่ 2.18 แสดงฐานข้อมูลระบบ Client/Server

จากรูปที่ 2.18 จะเห็นว่าเป็นการใช้คำสั่งภาษา SQL ซึ่งจะช่วยลดปัญหาการจราจรบนระบบเน็ตเวิร์กลงได้ (เพราะในรูปแบบของไฟล์เซิร์ฟเวอร์ เมื่อเครื่องไคลเอ็นต์ต้องการไฟล์ข้อมูล ไฟล์เซิร์ฟเวอร์ก็จะส่งข้อมูลจำนวนมากออกมาบนระบบเน็ตเวิร์ก เพื่อส่งต่อให้ไคลเอ็นต์ จะเป็นการเพิ่ม Traffic ให้กับระบบ) ฉะนั้นฐานข้อมูล SQL จึงเหมาะสมที่สุด ระบบฐานข้อมูล SQL ที่ทำงานอยู่บนเครื่องเซิร์ฟเวอร์ทั้ง 3 ระดับ บนระบบ Client/Server มีดังนี้

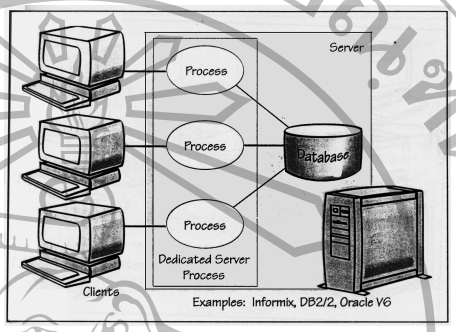
- ORACLE
- SYBASE
- Informix
- IBM DB2/2
- Gupta SQLBASE
- WATCOM SQL
- Microsoft SQL Server

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

- **สถาปัตยกรรม SQL Database Server**

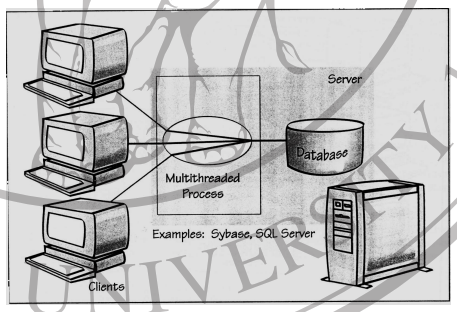
ระบบฐานข้อมูล SQL ที่ทำงานอยู่บนระบบ Client/Server มีอยู่หลายผลิตภัณฑ์ แต่ละผลิตภัณฑ์จะมีรูปแบบการทำงานต่างกัน สถาปัตยกรรมของ SQL Database Server จะมีอยู่ 3 แบบด้วยกันคือ

1. Process-per-Client Architectures



รูปที่ 2.19 แสดง Process per Client Database Server Architectures.

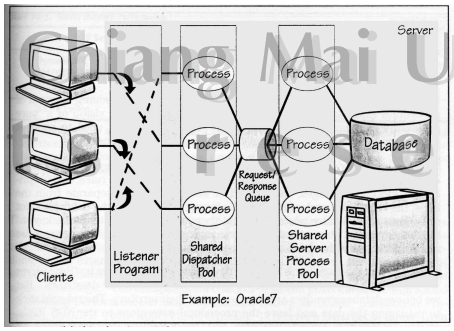
2. Multithreaded Architectures



รูปที่ 2.20 แสดง Multithreaded Database Server Architectures.

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

3. Hybrid Architectures

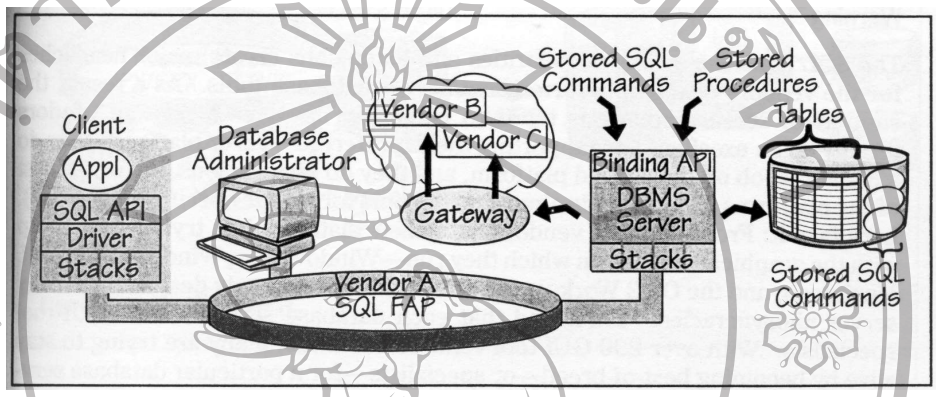


รูปที่ 2.21 แสดง Hybrid Database Server Architectures.

- **SQL มิดเดิลแวร์**

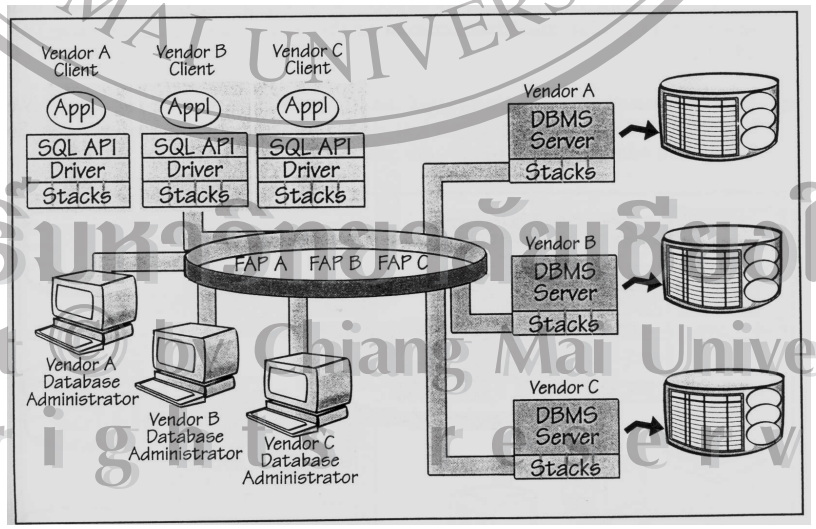
โปรแกรมของ SQL มิดเดิลแวร์ จะเป็นตัวกลางระหว่างไคลเอ็นต์และเซิร์ฟเวอร์คอยจัดการให้ไคลเอ็นต์ สามารถเข้าไปใช้ฐานข้อมูลในเซิร์ฟเวอร์ได้หลากหลาย ผู้ผลิต (Multivendor) มิดเดิลแวร์จะทำงานเกี่ยวกับ API บนไคลเอ็นต์ รูปแบบของมิดเดิลแวร์จะมีอยู่ 2 แบบด้วยกัน คือ

(1) Single Vendor Option



รูปที่ 2.22 แสดงรูปแบบของมิดเดิลแวร์ แบบ Single Vendor Option

(2) Multi vendor option

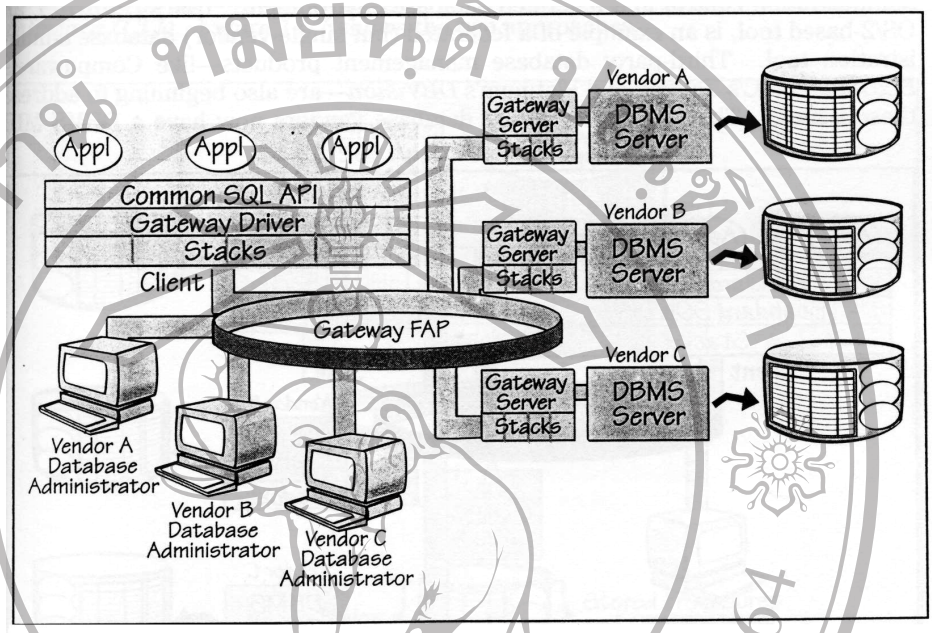


รูปที่ 2.23 แสดงรูปแบบของมิดเดิลแวร์ แบบ Multi vendor option

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved

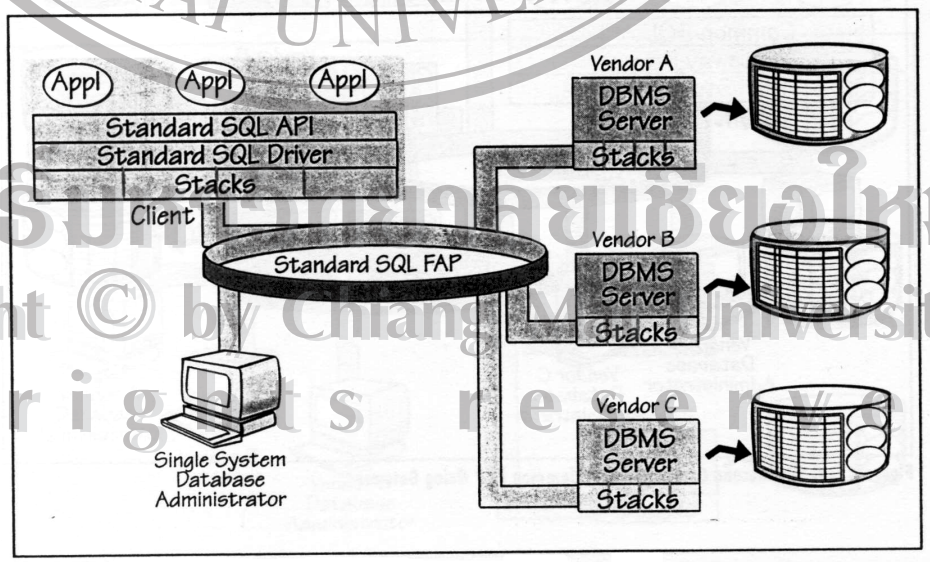
มิดเดิลแวร์โซลูชั่น หรือทางออกของมิดเดิลแวร์ในการจัดการกับฐานข้อมูล มีอยู่ 3 แบบ ด้วยกันคือ

(1) Common SQL Interface



รูปที่ 2.24 แสดง Common SQL Interface

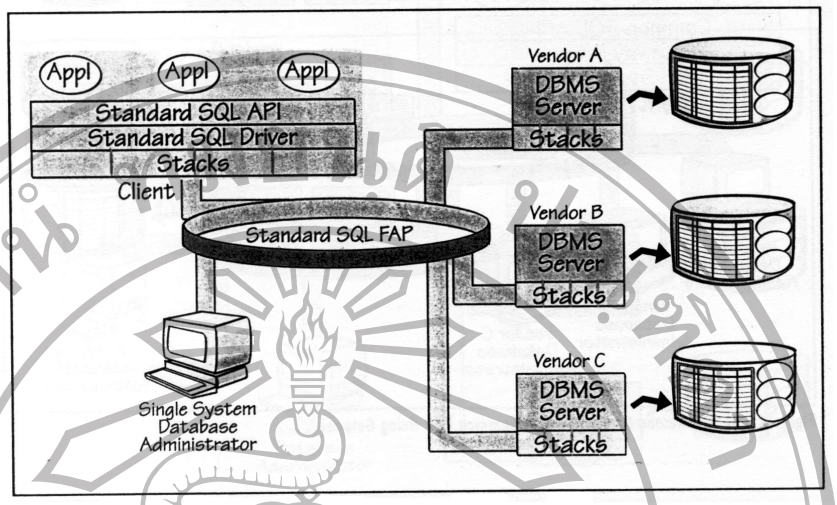
(2) Open SQL Gateway



รูปที่ 2.25 แสดง Open SQL Gateway

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

(3) Federated Database Administrator

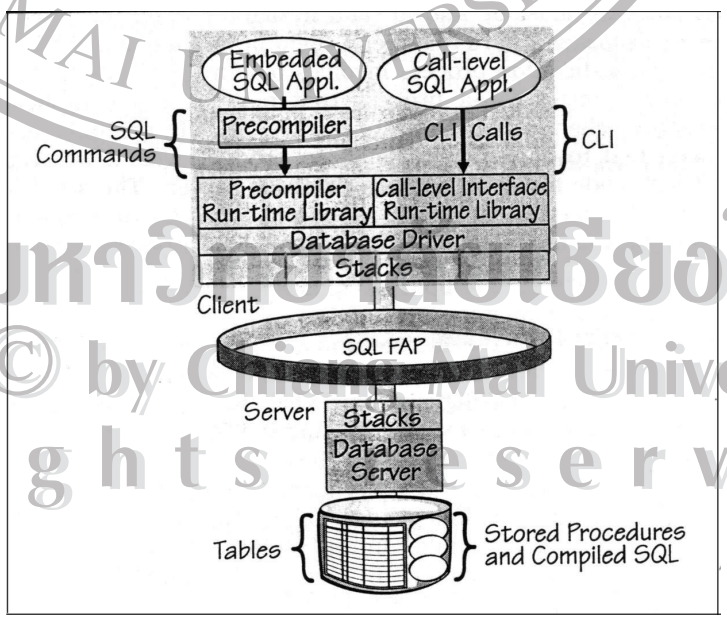


รูปที่ 2.26 แสดง Federated Database Administrator

- **SQL Application Program Interfaces**

SQL เป็นภาษาสอบถามเชิงโครงสร้างที่มีความใกล้เคียงภาษาเขียนของมนุษย์มากจึงมีการนำรูปแบบของภาษา SQL ไปใช้งานกับภาษาโปรแกรมมิ่ง ซึ่งจะมีอยู่ 2 รูปแบบ คือ

- Embedded SQL (ISO standard)
- SQL CLI (Cell-Level Interface)

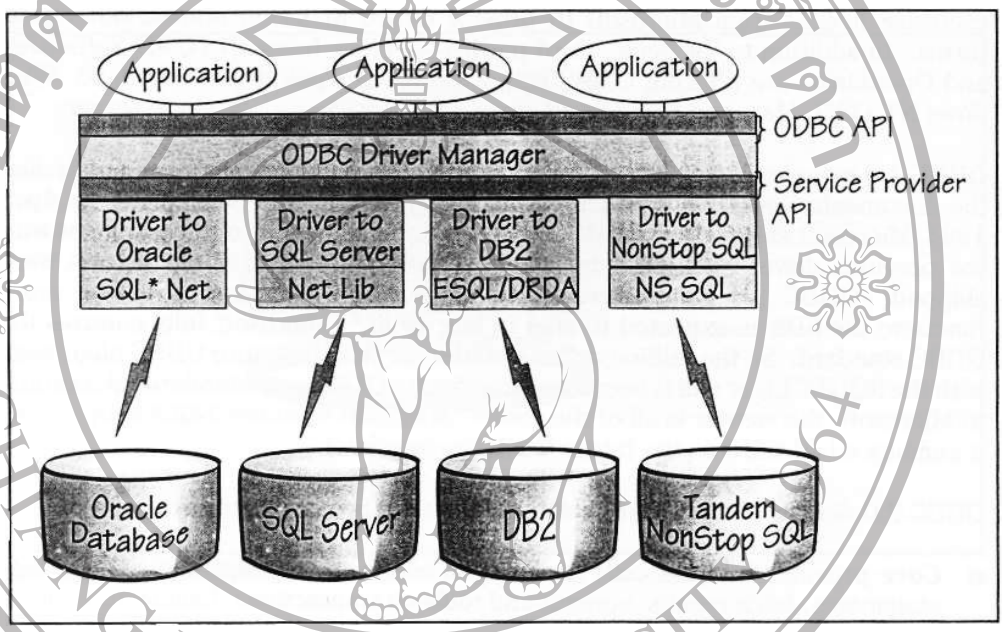


รูปที่ 2.27 แสดง SQL APIs Come in Two Styles : CLI and ESQL.

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

Embedded SQL หรือ ESQL เป็นมาตรฐาน ISO SQL-92 ที่กำหนดรูปแบบการฝังสเตทเมนต์ SQL ลงไปในภาษา C, PASCAL, COBOL, FORTRAN, PL/1, ADA ฯลฯ ส่วน SQL CLI เป็นมาตรฐานของหน่วยงาน SAG (SQL Access Group) ที่กำหนดฟังก์ชั่น ในการเรียกใช้ภาษา SQL ได้โดยตรงจากโปรแกรมแอพพลิเคชั่น จึงไม่มีการฝัง (embedded) ส่วนของภาษา SQL ลงไป

- **ไดร์ฟเวอร์ฐานข้อมูล ODBC (Open Database Connectivity)**

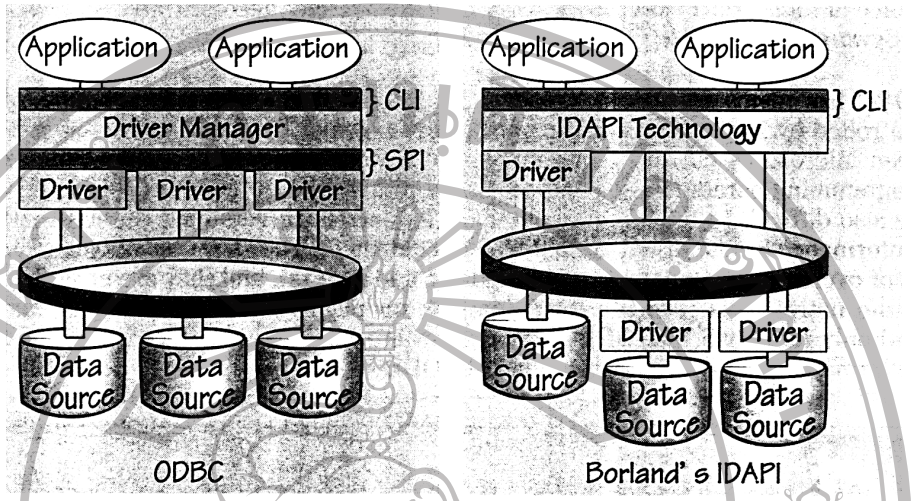


รูปที่ 2.28 แสดง ไดร์ฟเวอร์ฐานข้อมูล ODBC (Open Database Connectivity)

ODBC นี้ เป็นผลิตภัณฑ์ของไมโครซอฟท์ เป็น API ที่มีหน้าที่ในการเข้าถึงข้อมูลจากฐานข้อมูลที่ต่างกันได้ โดยจะใช้แอพพลิเคชั่นตัวเดียวกัน แต่จะใช้ไดร์ฟเวอร์ต่างกัน (ทำให้ไม่ต้องไปแก้ไขแอพพลิเคชั่น) จากรูปที่ 2.28 จะเห็นว่า แอพพลิเคชั่นจะทำงานผ่าน ODBC API และ ODBC DRIVER MANAGER ซึ่งเป็น DLL (Dynamic Link Library) มีหน้าที่ตรวจสอบความถูกต้องในการเรียกใช้ข้อมูล กำหนดการเรียกใช้ฟังก์ชั่น ODBC ให้ไดร์ฟเวอร์ทั่วไป และแมบข้อมูลกับไดร์ฟเวอร์นามสกุล DLL ทั้งหลาย จากนั้นก็จะเชื่อมต่อผ่านมายังส่วนของ Service Provider Interface (SPI) แล้วก่อนจะเข้าสู่ฐานข้อมูลต่าง ๆ ในส่วน Back-end ก็จะมีไดร์ฟเวอร์ของแต่ละผลิตภัณฑ์มารองรับ (ไมโครซอฟท์ได้นำเทคโนโลยีการเข้าถึงข้อมูลตัวใหม่เรียกว่า UDA (Universal Data Access) โดยจะเป็นการรวมเอา ODBC, ADO และ OLE DB เข้าด้วยกัน ทำให้สามารถใช้งานได้ทั้งฐานข้อมูลเชิงสัมพันธ์และฐานข้อมูลทั่วไป

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

- **IDAPI (Integrated Database Application Program Interface)**



รูปที่ 2.29 แสดง Drivers : ODBC Versus IDAPI

ประมาณปี 2536 บริษัท Borland, Novell, IBM และ WordPerfect ได้ร่วมกันร่างมาตรฐานไลบรารี API ของตนเองที่เรียกว่า IDAPI และมีการรวมฟังก์ชั่นมาตรฐานของ CLI จำนวน 25 ฟังก์ชั่นเข้ากับฟังก์ชั่นของ IDAPI อีก 81 ฟังก์ชั่น เทคโนโลยีของ IDAPI เป็นที่ยอมรับของบริษัทผลิตฐานข้อมูลค่ายใหญ่ ๆ เช่น Gupta, Informix, Sybase, Oracle ฯลฯ

จากรูปที่ 2.29 จะเห็นว่าเทคโนโลยี IDAPI มี Object Layer เป็นตัวจัดการสภาพแวดล้อมต่างๆ จัดสรรทรัพยากร จัดการของแอพพลิเคชั่น ฯลฯ นอกจากนี้ยังมีชั้นของ Service Layer ที่เป็นส่วนเสริมประกอบด้วย 3 โมดูล คือ

- Operating System Module จะมีฟังก์ชั่น I/O ฟังก์ชั่นจัดการหน่วยความจำของแต่ละระบบปฏิบัติการ
- Language Module เป็นตัวสนับสนุนให้สามารถใช้งานได้หลายภาษา
- Buffer Management เป็นตัวจัดการเก็บข้อมูลแบบ BLOB (Binary Large Object) ของไฟล์โปรแกรม Word และรูปที่ถูกสแกน

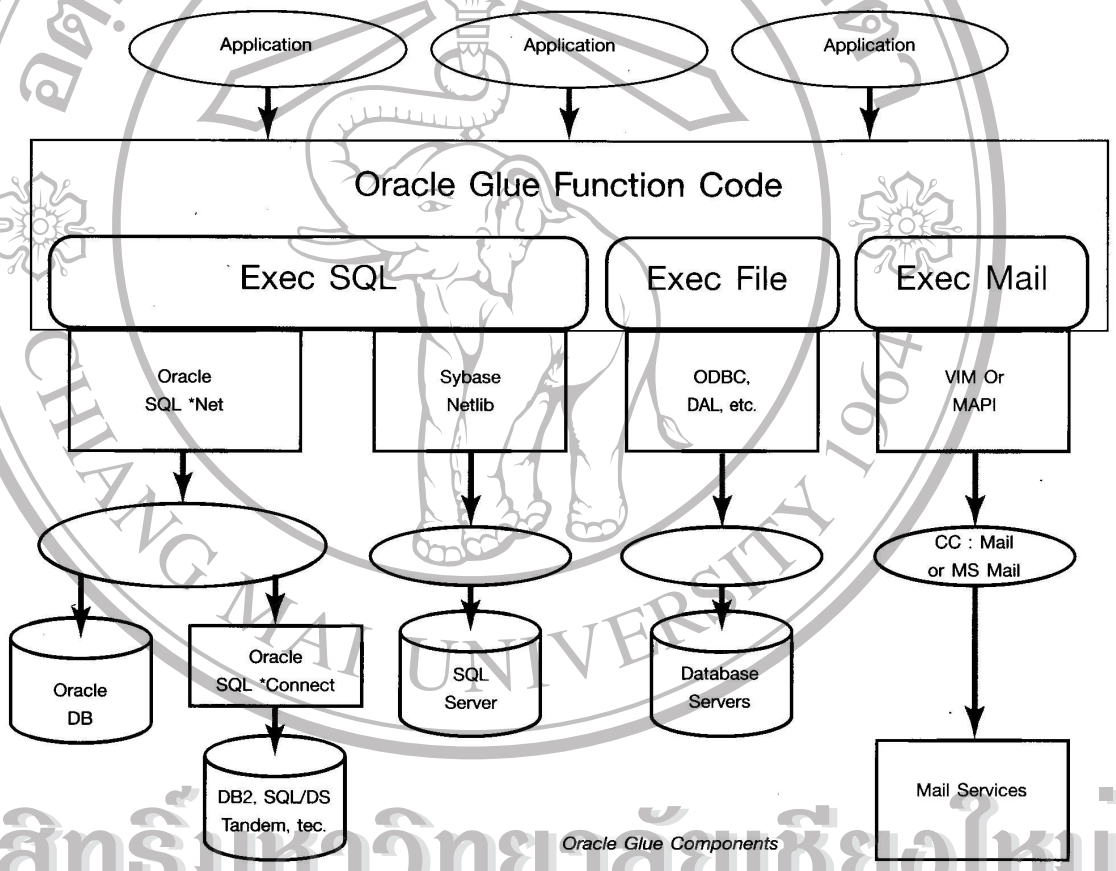
ส่วน SQL Driver นั้นจะเป็นไดร์ฟเวอร์สำหรับฐานข้อมูลประเภท SQL เช่น Netware SQL, Interbase, Oracle และ ODBC

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

- **Oracle Glue**

Oracle Glue เป็นผลิตภัณฑ์ของบริษัท Oracle ซึ่งสามารถจะทำงานร่วมกับ ODBC, IDAPI, Sybase DB LIB, Paradox, APPle's DAL และ Email

Glue API มีฟังก์ชันการทำงานอยู่ 4 ตัวคือ Exec SQL, Exec file, Exec mail และ Exec link ดังรูปที่ 2.30

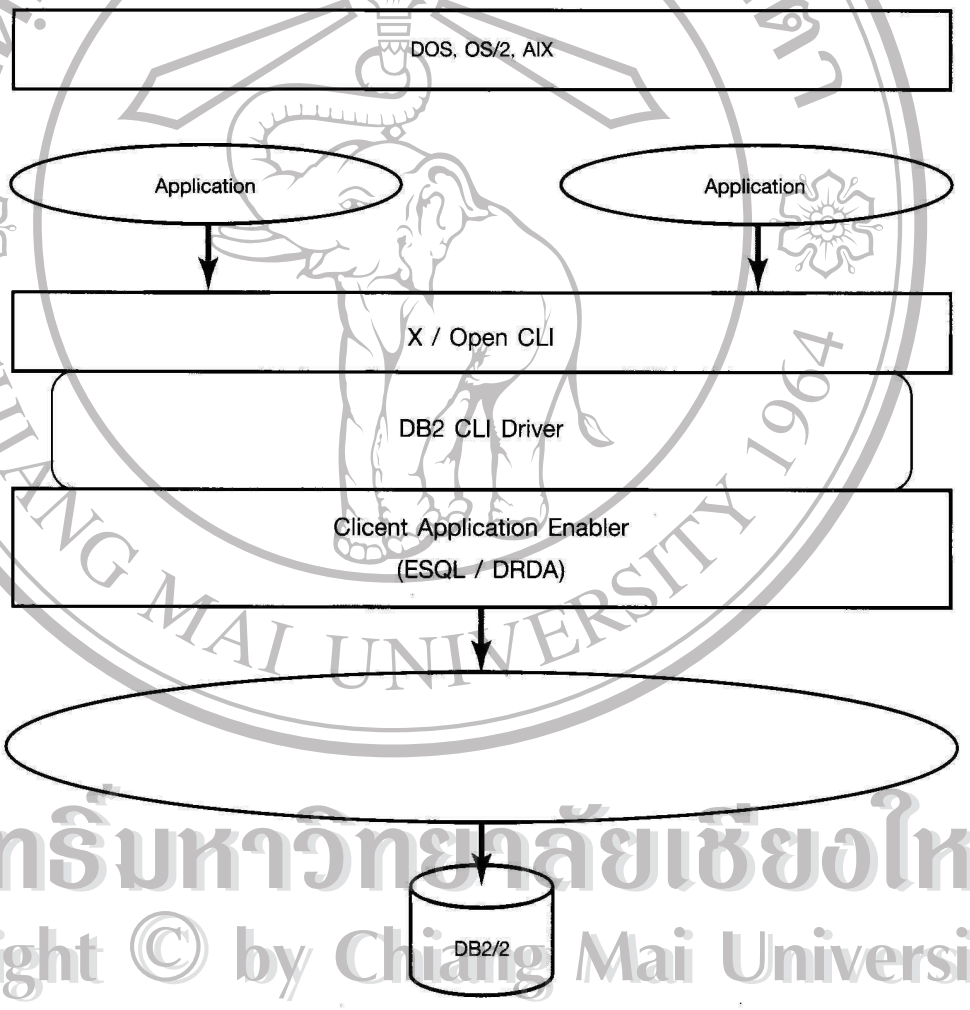


ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

รูปที่ 2.30 แสดงฟังก์ชันการทำงานของ Glue API

- **IBI EDA/SQL (Enterprise Data Access/SQL)**

IBI หรือ Information Builder, Inc. จึงเข้าถึงระบบฐานข้อมูลโดยใช้ระบบเกตเวย์ (Gateways) ทำให้สามารถเข้าถึงฐานข้อมูลเชิงสัมพันธ์ (RDBMS) ได้มากกว่า 50 ตัว (ระบบ SQL Gateways ดังรูป) IBI EDA/SQL ยังเป็นส่วนประกอบของ IBM Information Warehouse DDS ส่วนประกอบของ EDA/SQL มี API/SQL, EDA/Extenders, EDA/Link, EDA/Server, EDA/DATA Drivers



แสดง DB2 CLI Driver

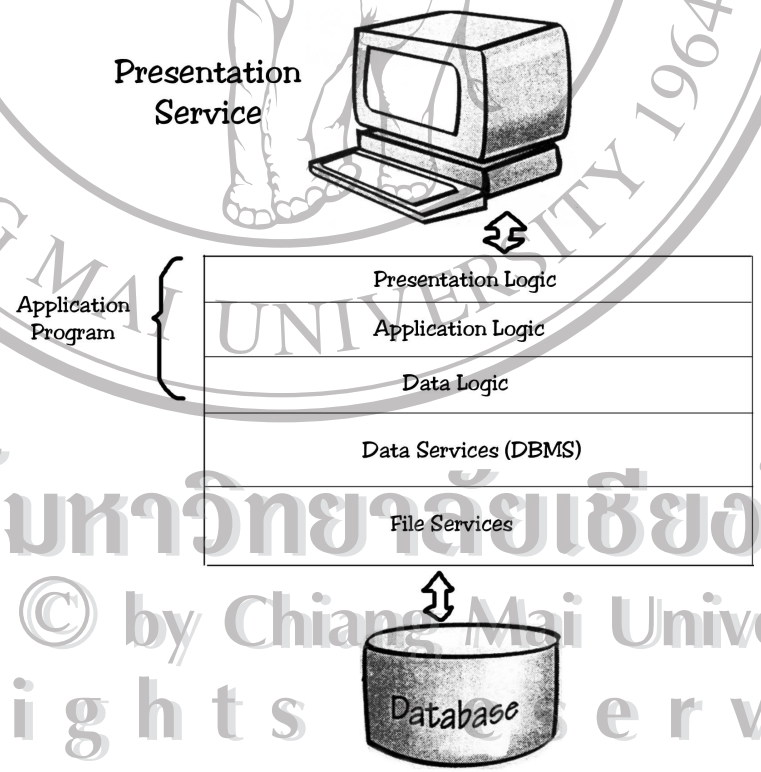
รูปที่ 2.31 แสดง IBI EDA/SQL (Enterprise Data Access/SQL)

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

2.2.5.8 สถาปัตยกรรม N Tier, 2 Tier, 3 Tier Client/Server

เทคโนโลยี Client/Server ทำให้มีการแบ่งงานกันทำระหว่างเครื่องเซิร์ฟเวอร์และเวิร์กสเตชั่น แอพพลิเคชั่น หรือโปรแกรมประยุกต์ในระบบ Client/Server ก็จะทำงานในลักษณะการประมวลผลแบบกระจาย (Distributed Processing) เช่นกัน แอพพลิเคชั่นจะมีส่วนประกอบในการทำงานอยู่ 6 ส่วนด้วยกัน คือ

1. Presentation Services
2. Presentation Logic
3. Application Logic
4. Data Logic
5. Data services
6. File Services



ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

รูปที่ 2.32 แสดงส่วนประกอบในการทำงานของ Application

1. Presentation Services (ส่วนบริการแสดงผล) เป็นส่วนในการนำเสนอและแสดงผลข้อมูลที่รับมาจากผู้ใช้ทางคีย์บอร์ด แล้วแสดงผลออกมาทางจอภาพแบบตัวอักษร (Text mode) หรือรับคำสั่งผ่านเมาส์บน GUI เพื่อเสนอในแบบกราฟฟิกส์

2. Presentation Logic (ส่วนบริการแสดงผลแบบตรรก) เป็นส่วนดูแลในการติดต่อระหว่างผู้ใช้กับแอพพลิเคชั่น การรับอินพุตจากคีย์บอร์ด การคลิ๊กปุ่มบนเมาส์ การเลื่อนเคอร์เซอร์ขึ้น - ลง การเลือกเมนูคำสั่ง ฯลฯ

3. Application Logic (ส่วนจัดการแอพพลิเคชั่นแบบตรรก) เป็นส่วนหนึ่ง ในการจัดการโปรเซสการทำงานของแอพพลิเคชั่นแบบตรรก การกำหนดวงรอบในการทำงาน - การคำนวณฯลฯ ดังนั้น ภาษาที่ใช้จะต้องเป็นโพซีเดอรัล (Procedural Language) ซึ่งสามารถทำงานแบบอัลกอริธึม (Algorithm) ในการกำหนดเงื่อนไขต่างๆ เช่น If…then…else การตรวจความถูกต้องของข้อมูล ฯลฯ เช่น ภาษา COBOL, PASCAL, C

4. Data Logic (ส่วนจัดการข้อมูลแบบตรรก) เป็นส่วนจัดการข้อมูลต่างๆ ทั้งการค้นหา (Search) ปรับปรุง (Update) เพิ่มเติม (Insert) การลบ (Delete) ระบบฐานข้อมูลเชิงสัมพันธ์ (RDBMS) จะใช้ภาษา SQL ในการจัดการกับข้อมูลเหล่านี้ด้วยคำสั่งที่เข้าใจง่ายและเป็นธรรมชาติ

5. Data Services (ส่วนบริการข้อมูล) เป็นส่วนบริการข้อมูลโดยมี DBMS (Database Management System) เป็นตัวจัดการค้นหา จัดระเบียบข้อมูล ฯลฯ บนระบบฐานข้อมูลเชิงสัมพันธ์จะมีลักษณะเป็นตาราง (Table) ทำให้ผู้ใช้มีความสะดวกและง่ายในการใช้งาน

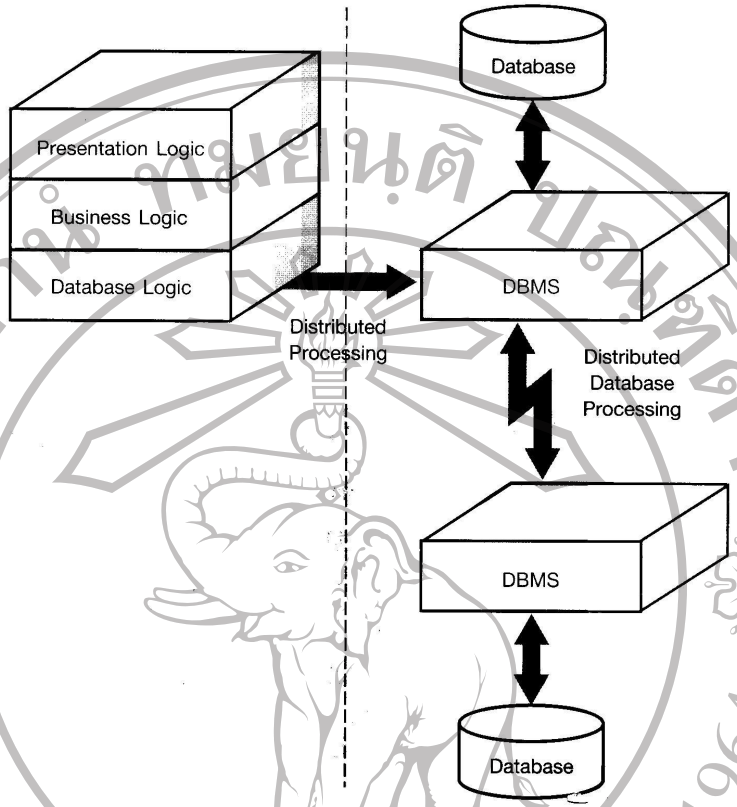
6. File Servics (ส่วนบริการไฟล์ข้อมูล) เป็นส่วนในการบริการและจัดการไฟล์ข้อมูลในระดับกายภาพ ส่วนนี้จะอยู่ใกล้กับฐานข้อมูลมากที่สุด

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

- **สถาปัตยกรรมแบบกระจาย (Distributed Architectures)**

การทำงานในแบบกระจายเพื่อประสิทธิภาพของการประมวลผลบนระบบเน็ตเวิร์ก LAN และ WAN สามารถแบ่งออกเป็น 2 รูปแบบ คือ

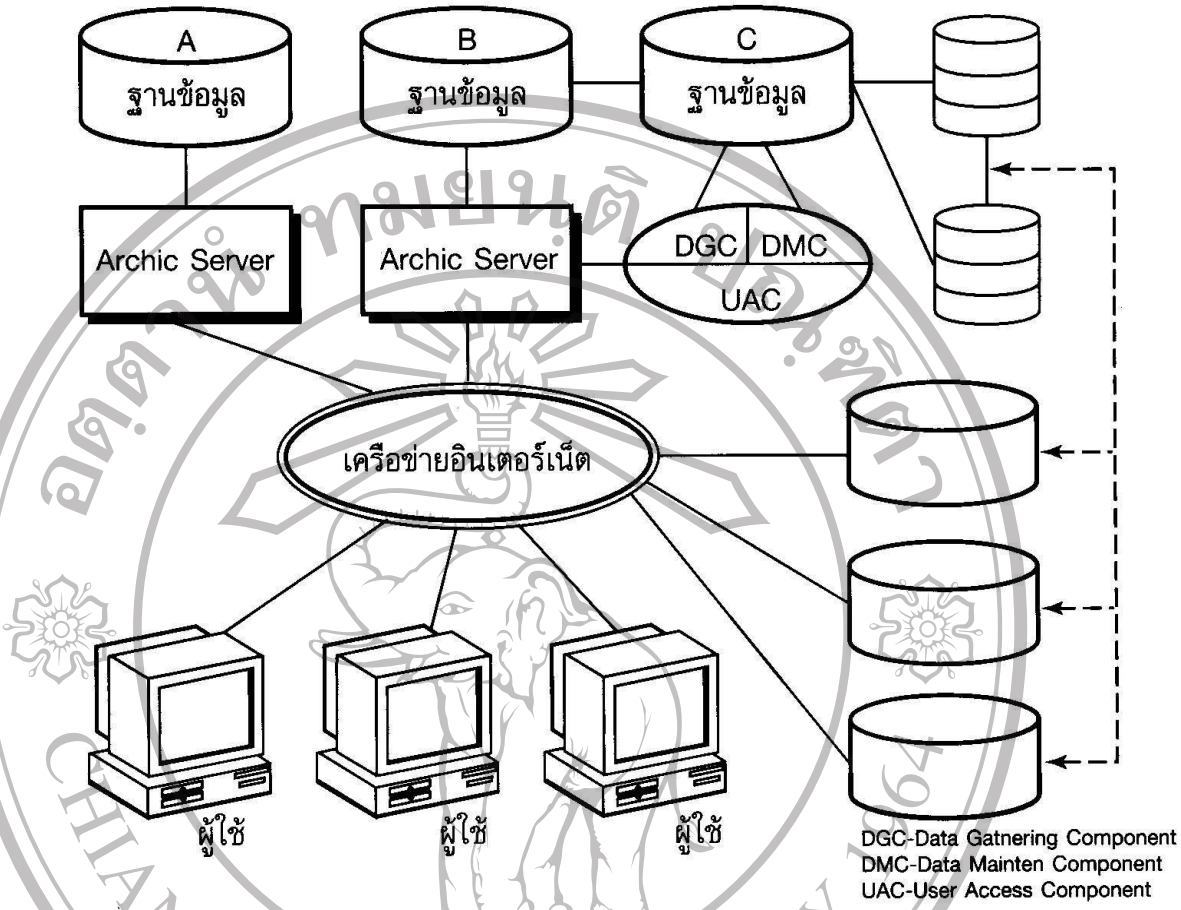
- การประมวลผลแบบกระจาย (Distributed Processing)
- ฐานข้อมูลแบบกระจาย (Distributed Database)



รูปที่ 2.33 แสดงการทำงานของสถาปัตยกรรมแบบกระจาย

จากรูปที่ 2.33 จะเป็นว่าในการประมวลผลแบบกระจาย สามารถจะแยกกันทำงานได้ โดยที่ทำการประมวลผลแอพพลิเคชั่นธุรกิจ (Business Application) บนเซิร์ฟเวอร์ A และประมวลผลทางด้านฐานข้อมูลบนเซิร์ฟเวอร์ B ทำให้งานไม่โหลดอยู่ที่เครื่องใดเครื่องหนึ่งสำหรับฐานข้อมูลแบบกระจายนั้นจะเป็นการแยกเอาข้อมูลไปเก็บไว้ในที่ต่างๆ บนระบบเน็ตเวิร์ก ซึ่งอาจจะมีข้อมูลหลากหลาย สถาปัตยกรรมในการเชื่อมโยงฐานข้อมูลแบบกระจายจะต้องช่วยให้ แต่ละฐานข้อมูลทำงานร่วมกันได้ เครือข่ายอินเตอร์เน็ตเป็นการจัดการฐานข้อมูลแบบกระจาย ที่มีประสิทธิภาพในปัจจุบัน ดังรูป จะเห็นว่าไม่ว่าฐานข้อมูลจะกระจายอยู่ส่วนใดบนเน็ตเวิร์กหรือเครือข่ายอินเตอร์เน็ต ก็ไม่มีผลกระทบในการเรียกใช้งานจากผู้ใช้ (คือผู้ใช้งานไม่จำเป็นต้องรู้ว่าข้อมูลที่ต้องการ อยู่ที่ใด แต่สามารถเรียกใช้งานได้ทันที)

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved



รูปที่ 2.34 แสดงโครงสร้างการเชื่อมโยงฐานข้อมูลแบบกระจาย

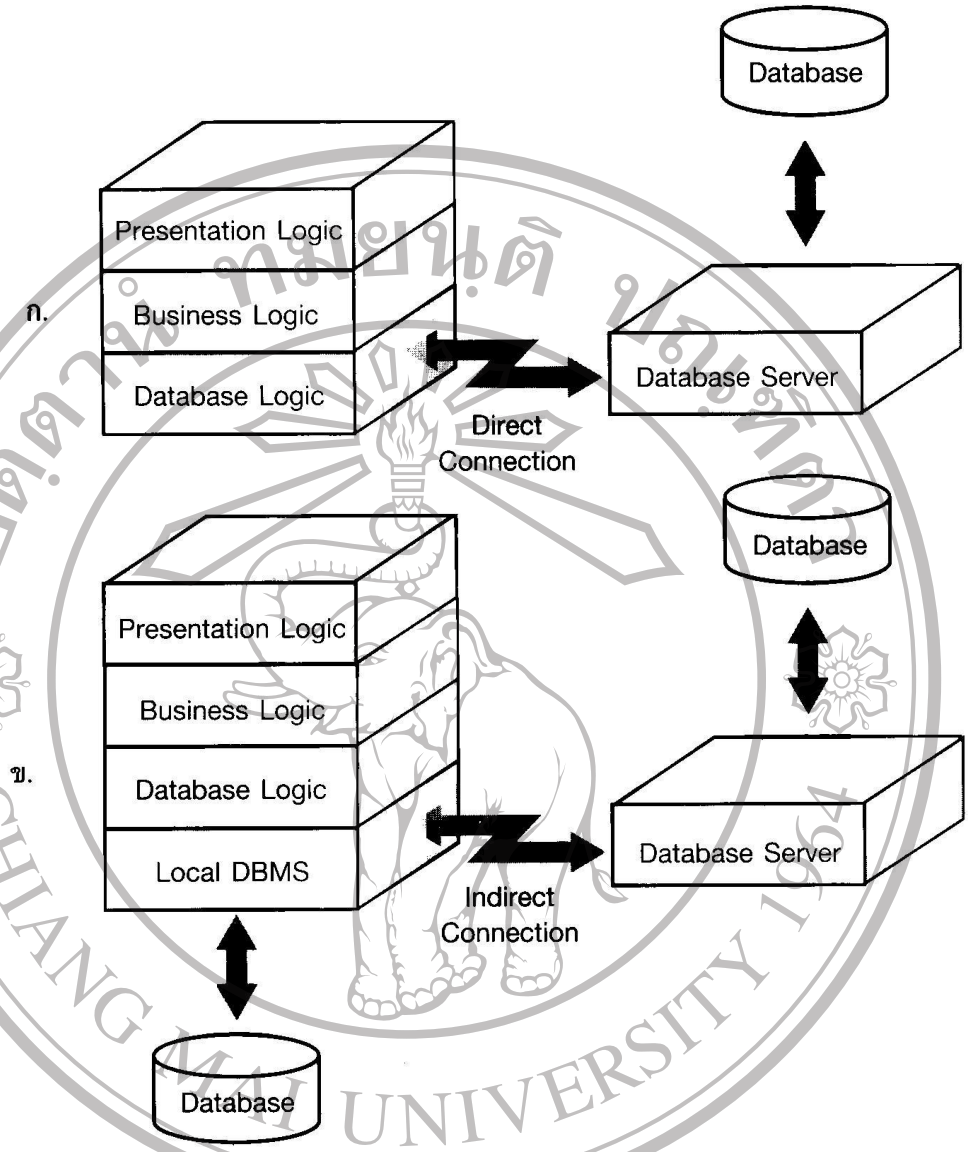
- **การติดต่อสื่อสาร**

ในการติดต่อสื่อสารและเชื่อมโยงกันระหว่างไคลเอ็นต์แอพพลิเคชั่น กับเซิร์ฟเวอร์ ฐานข้อมูลจะมีอยู่ 2 แบบด้วยกัน คือ

- การเชื่อมต่อโดยตรง (Direct Connection)
- การเชื่อมต่อทางอ้อม (Indirect Connection)

การเชื่อมต่อโดยตรง คือการที่ไคลเอ็นต์แอพพลิเคชั่น สามารถจะติดต่อสื่อสารกับฐานข้อมูลที่อยู่ห่างไกลโดยตรง ซึ่งจะใช้ส่วนจัดการข้อมูลแบบตรรก (Database logic) เป็นตัวจัดการ ดังรูปที่ 2.35 ในการเชื่อมต่อระหว่างไคลเอ็นต์แอพพลิเคชั่นกับ เซิร์ฟเวอร์ ฐานข้อมูลที่อยู่ห่างไกลจะใช้ Gateway ในการทำงาน

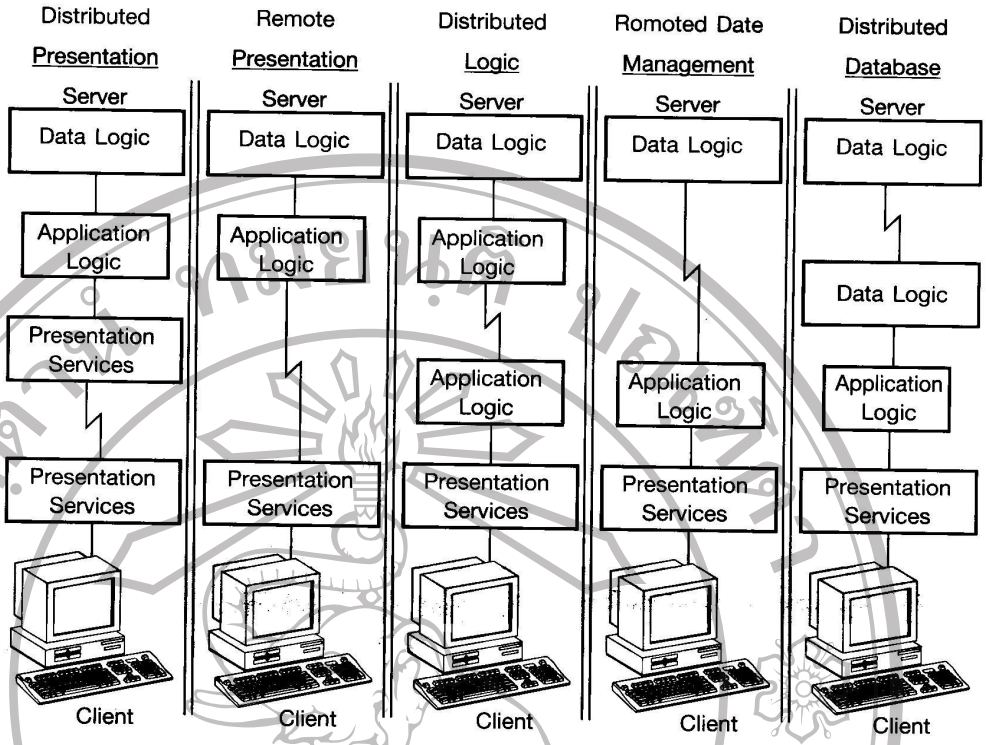
ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved



รูปที่ 2.35 แสดงการติดต่อสื่อสารแบบ Direct และ Indirect Connection.

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

การเชื่อมต่อทางอ้อม คือการที่ไคลเอ็นต์แอพพลิเคชั่น ไม่สามารถจะติดต่อสื่อสารกับเซิร์ฟเวอร์ฐานข้อมูลที่อยู่ห่างไกลได้โดยตรง แต่จะต้องติดต่อผ่านผู้ให้บริการฐานข้อมูลท้องถิ่น (Local DBMS) หรือในบริเวณเน็ตเวิร์กวงเดียวกันเท่านั้น ดังรูป ข. เช่นมีระบบเซิร์ฟเวอร์ฐานข้อมูล A รันอยู่บนเครื่องเซิร์ฟเวอร์ในระบบเน็ตเวิร์ก (LAN) ที่มีเครื่องไคลเอ็นต์อยู่ 50 เครื่อง เมื่อใดที่แอพพลิเคชั่นบนไคลเอ็นต์ต้องการใช้ข้อมูลบนฐานข้อมูลที่อยู่ห่างไกล เซิร์ฟเวอร์ฐานข้อมูล A จะทำการเชื่อมต่อให้ ทำให้แอพพลิเคชั่นสามารถจะเรียกใช้ข้อมูลที่อยู่ในบริเวณเดียวกันและห่างไกลได้



รูปที่ 2.36 แสดงรูปแบบการทำงานแบบกระจาย ทั้ง 5 ชนิด

รูปแบบการทำงานแบบกระจายทั้ง 5 ชนิด มีรายละเอียดดังนี้

1. Distributed Presentation เป็นการกระจายการแสดงผล โดยที่เซิร์ฟเวอร์จะทำการคำนวณและจัดการกับข้อมูลทั้งหมด แล้วส่งผลลัพธ์มาให้ส่วน Presentation Services เพื่อแสดงผล

2. Remote Presentation การทำงานของแอพพลิเคชั่นโลจิก การคำนวณและการจัดการกับข้อมูลจะอยู่ที่เซิร์ฟเวอร์ทั้งหมด แต่ส่วน Presentation Service จะอยู่เฉพาะที่ไคลเอ็นต์เท่านั้น

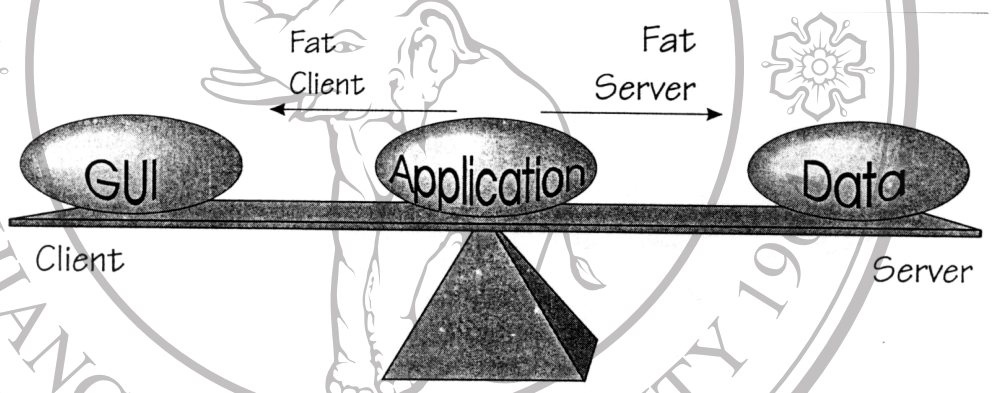
3. Distributed Logic เป็นการกระจายการทำงานของส่วนแอพพลิเคชั่นให้อยู่บนเซิร์ฟเวอร์และไคลเอ็นต์ การคำนวณบางส่วนจะถูกส่งมาให้ไคลเอ็นต์ช่วยทำ แต่การคำนวณที่ซับซ้อนทางเซิร์ฟเวอร์จะเป็นผู้ทำเอง ลักษณะนี้จะมีการทำงานแบบ Front-end และ Back-end

4. Remote Data Management เป็นการแยกการทำงานของแอพพลิเคชั่นโลจิก ให้มาอยู่ที่ไคลเอ็นต์ ส่วนเซิร์ฟเวอร์จะมีหน้าที่จัดการกับข้อมูลเท่านั้น การทำงานแบบนี้ไคลเอ็นต์จะต้องมีประสิทธิภาพในการทำงานพอสมควร

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

5. Distributed Database เป็นการทำงานที่มีประสิทธิภาพมาก เพราะจะกระจายการจัดการข้อมูล (Data Logic) ให้อยู่ทั้งไคลเอ็นต์และเซิร์ฟเวอร์ การจัดการข้อมูลบางส่วน และการทำงานของแอพพลิเคชั่นทั้งหมดจะอยู่ที่ไคลเอ็นต์ แต่การคำนวณ-การค้นหาข้อมูลที่ซับซ้อนจะทำบนเซิร์ฟเวอร์และที่สำคัญเครื่องไคลเอ็นต์จะต้องมีประสิทธิภาพในการทำงานสูงกว่าทั้ง 4 แบบที่กล่าวมา

- **FAT SERVERS OR FAT CLIENTS**



รูปที่ 2.37 แสดง FAT SERVERS OR FAT CLIENTS?

ในส่วนของ Client/Server สามารถแบ่งได้โดยลักษณะของการให้บริการ Client/Server Application สามารถแบ่งออกด้วยการให้บริการการกระจายของ Application (Distributed Application) ระหว่างฝั่งของ client และฝั่งของ server (ตามรูปที่ 2.37).

สำหรับ Fat Server Model จะใช้ในกรณีที่เน้นภาระงานความรับผิดชอบที่อยู่ที่ Server ส่วน Fat Client Model ก็จะเป็นในแนวกลับกัน ก็คือ ภาระงานจะอยู่ที่ Client

สำหรับรูปแบบของ Groupware, Transaction และ Web Servers จะจัดอยู่ในรูปแบบ Fat server ส่วน database และ File Servers จะจัดอยู่ในรูปของ Fat client และ Distributed Objects สามารถเป็นไปได้ทั้ง 2 รูปแบบ

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

Fat Clients เป็น รูปแบบดั้งเดิมของการทำงานแบบ Client/Server ภาพรวมของ application ทำงานบนฝั่ง client เท่ากัน ทั้ง file servers และ database server model โดย client รู้ว่าข้อมูลจะถูกจัดการและเก็บไว้อย่างไร ที่ฝั่งของ server Fat client ถูกใช้เพื่องานช่วยประกอบการตัดสินใจ และโปรแกรมส่วนบุคคล เพราะรูปแบบนี้จะ ยืดหยุ่น (flexibility) และ เหมาะสม โอกาส (opportunities) สำหรับการสร้าง front-end tools และ end-users สามารถสร้างโปรแกรมประยุกต์ของตนได้

Fat Server โปรแกรมสำเร็จรูปจะถูกจัดการอย่างง่าย และกระจายบนเครือข่าย เพราะส่วนมากของ code จะทำงานบน server Fat server พยายามที่จะใช้เครือข่ายในการแลกเปลี่ยนข้อมูลให้น้อยที่สุด โดยจัดการสร้างชั้นการบริการต่างหาก เช่น Transaction และ Object servers จะทำการสร้างมาครอบฐานข้อมูลอีกครั้ง แทนที่จะส่งออกข้อมูลที่ไม่ได้กลั่นกรองล้วนๆ ในรูปแบบนี้ จะส่งเป็น procedure (หรือ เป็นรูป method ในแบบ Object-oriented terminology) ที่ซึ่งกระทำบนข้อมูลนั้น ๆ Client ในรูปของ Fat server จะให้ GUI และ การติดต่อกับ server ผ่านกระบวนการทำงานระยะไกล

ทั้ง 2 รูปแบบของ Client/Server จะถูกใช้งานเฉพาะตามรายละเอียดที่กล่าวมา และมีหลายกรณีรูปแบบแต่ละแบบก็จะรองรับซึ่งกันและกัน. และจะไม่ได้เฉพาะของใครของมันในหนึ่ง Application ยกตัวอย่างเช่นโปรแกรมประยุกต์เกี่ยวกับรูปภาพโดย Groupware สามารถกำหนดรูปแบบรวมทั้งหมดเป็นหนึ่งเดียวบน (all-in-one)server จะรวมระหว่าง แฟ้มข้อมูล (File) ฐานข้อมูล (Database) แฟ้มชั่วคราว (Transaction) และ Object Services Fat servers จะถูกใช้สำหรับพันธกิจวิกฤต การนำเสนอพื้นที่อัตราเจริญเติบโตของ Personal Computer ที่เป็นฐานของ Client/Server

- **2-TIER VS 3-TIER**

ผู้เชี่ยวชาญทางด้าน Client/Server มักจะใช้คำว่า 2-tier และ 3-tier Client/Server แทนคำว่า Fat clients และ Fat servers ซึ่งก็มาจากพื้นฐานแนวคิดอันเดียวกัน เพียงแต่สนใจว่า ทำอย่างไรจึงสามารถแยกส่วนของ Client/Server Application ในหน่วยของหน้าที่ที่จะทำงานเป็น client หรือรูปแบบของ server ส่วนมากหน้าที่หลักจะแบ่งเป็นส่วนของ user interface, business logic และ shared data มีอยู่หลายสาเหตุที่สามารถเป็นไปได้ของรูปแบบโครงสร้างแบบ multi-tier ขึ้นอยู่กับว่าเราจะแบ่งในส่วนของ application และ middleware ที่ใช้ในการต่อเชื่อมระหว่าง tier

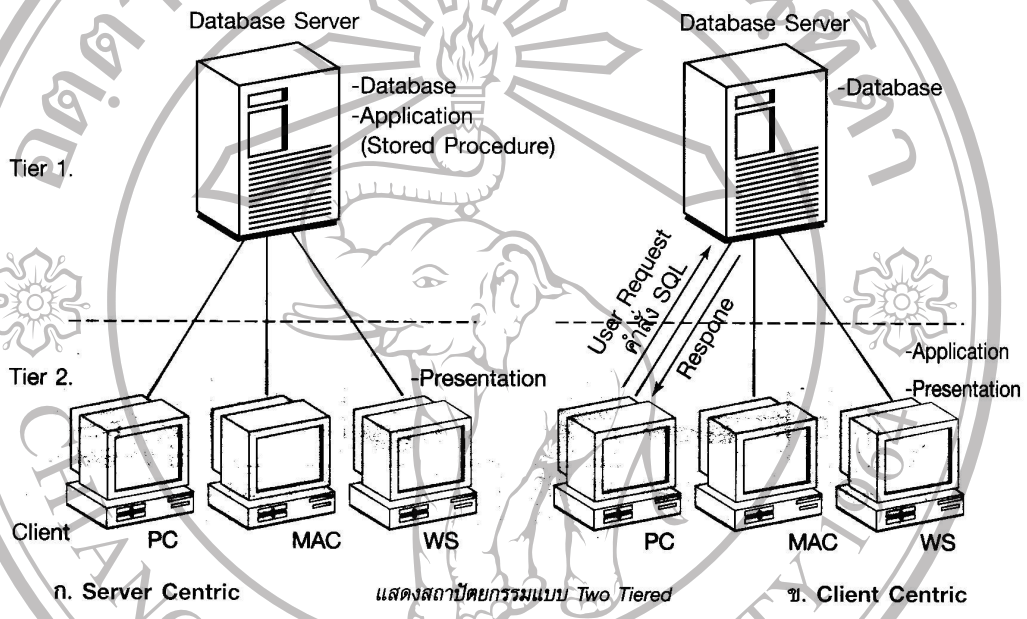
รูปแบบระบบ Client/Server แบบ 2-tier, application logic อาจจะอยู่ที่ฝั่งของ client (user interface) หรือจะอยู่กับระบบฐานข้อมูลบน server (หรือทั้งคู่) รูปแบบระบบ Client/Server แบบ 3-tier application

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

- **สถาปัตยกรรม N-Tier**

Client/Server จะมีการแยกส่วนของ แอพพลิเคชั่นฐานข้อมูลและส่วนแสดงผลออกจากกัน เพื่อประสิทธิภาพในการทำงานจึงมีสถาปัตยกรรม N-Tier ซึ่งมีอยู่ 2 แบบด้วยกัน คือ

- สถาปัตยกรรม Two Tiered
- สถาปัตยกรรม Three Tiered



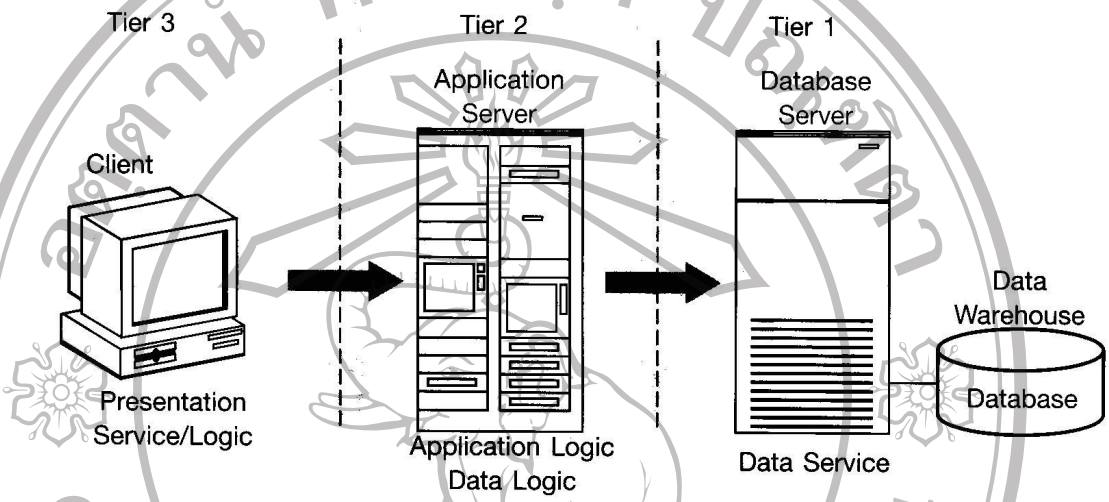
รูปที่ 2.38 แสดงสถาปัตยกรรมแบบ Two Tiered

การทำงานในสถาปัตยกรรม Two Tiered จะเป็นการแยกข้อมูลออกจากส่วนจัดการแอพพลิเคชั่นและส่วนแสดงผล จากรูป ก. เป็นแบบ Server Centric จะเห็นว่าฐานข้อมูลและแอพพลิเคชั่นอยู่บนเซิร์ฟเวอร์ฐานข้อมูลใน Tier 1 เครื่องไคลเอ็นต์จะจัดการในส่วนที่ต้องติดต่อกับผู้ใช้ในแบบ Text/Graphics Mode ใน Tier 2

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

Client Centric นั้นจะมีเฉพาะฐานข้อมูลอยู่บนเซิร์ฟเวอร์ใน Tier 1 เท่านั้นและไคลเอ็นต์จะมีส่วนจัดการแอพพลิเคชั่น ส่วนติดต่อกับผู้ใช้ อยู่ใน Tier 2 จากรูปที่ ข. เมื่อไคลเอ็นต์ต้องการข้อมูลก็จะร้องขอไปยังเซิร์ฟเวอร์ในรูปแบบภาษา SQL ลักษณะนี้จะช่วยแบ่งเบาภาระการทำงานให้เซิร์ฟเวอร์แต่ถ้ามีการปรับปรุงแก้ไขแอพพลิเคชั่นก็จะต้องตามแก้ไขที่ไคลเอ็นต์ทุกเครื่องเช่นกัน โดยปรกติของการทำงานควรจะนำส่วนที่เปลี่ยนแปลงบ่อยที่สุดไปไว้ยังเซิร์ฟเวอร์ (Host) เมื่อมีการ

แก้ไขก็สามารถทำที่เครื่องเซิร์ฟเวอร์เพียงเครื่องเดียว เพื่อช่วยลด Traffic บนระบบเน็ตเวิร์ก ในการทำงานกับ Two Tiered ทำให้มีเครื่องมือประเภท RAD (Rapid Application Development) ในการพัฒนาขึ้นเช่น Power Builder, Forest & Trees,

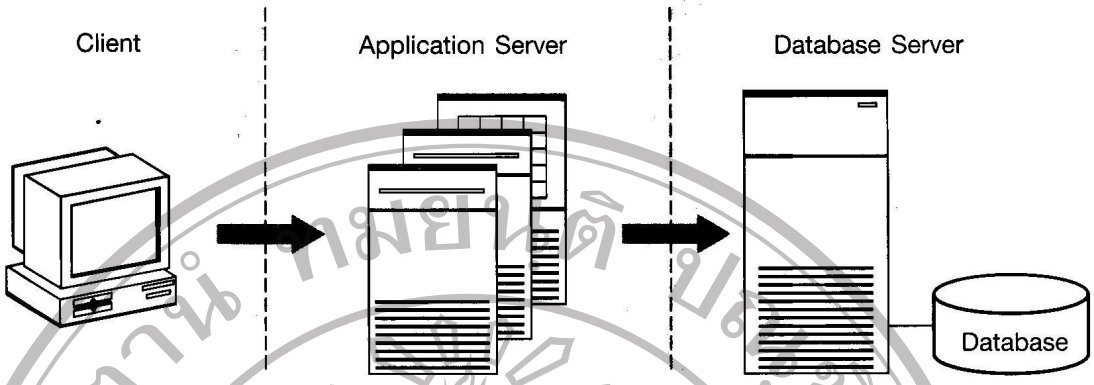


รูปที่ 2.39 แสดงสถาปัตยกรรมแบบ Three Tiered

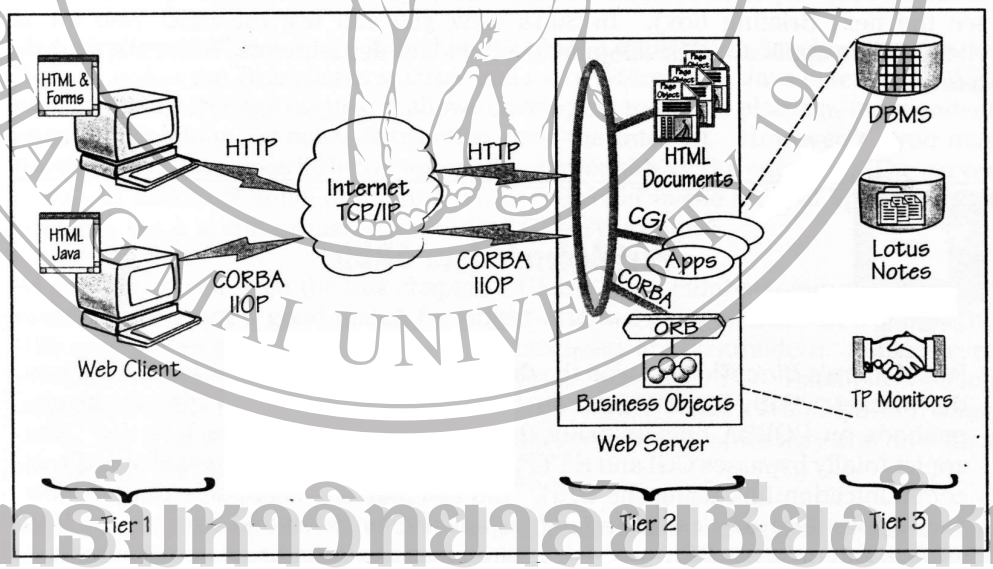
- **SQL Windows**

การทำงานในสถาปัตยกรรม Three Tiered นี้จะเป็นการแยกส่วนนำเสนอ (Presentation Service/logic) ส่วนแอพพลิเคชั่นโลจิก (Business logic) และข้อมูลออกจากกันอย่างชัดเจน จากรูปจะเห็นว่าส่วนนำเสนอ-แสดงผลจะอยู่ใน Tier 3 บนเครื่องไคลเอ็นต์มีหน้าที่จัดการในส่วนที่ต้องติดต่อกับผู้ใช้เท่านั้น ส่วนแอพพลิเคชั่นจะอยู่ใน Tier 2 บนเครื่องแอพพลิเคชั่นเซิร์ฟเวอร์ ทำหน้าที่ประมวลผลเกี่ยวกับ แอพพลิเคชั่นโปรแกรมต่าง ๆ และเชื่อมต่อกับตัวจัดการ DBMS บนเครื่องดาต้าเบสเซิร์ฟเวอร์ สุดท้ายเป็นส่วนของฐานข้อมูล อยู่ใน Tier 1 บนเครื่องดาต้าเบสเซิร์ฟเวอร์ มีหน้าที่ให้บริการข้อมูลสถาปัตยกรรม Three Tiered นี้มีความยืดหยุ่นสูงสามารถเพิ่มขยายได้ในอนาคต ถ้าที่การเปลี่ยนแปลงแก้ไขแอพพลิเคชั่นโปรแกรมก็สามารถแก้ไขเฉพาะใน Tier 2 บนแอพพลิเคชั่นเซิร์ฟเวอร์ และถ้ามีการเพิ่มจำนวนผู้ใช้มากขึ้น ก็ไม่ต้องไปปรับเปลี่ยนดาต้าเบสเซิร์ฟเวอร์ เพียงแต่เพิ่มเซิร์ฟเวอร์เข้าไปใน Tier 2 เท่านั้น

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

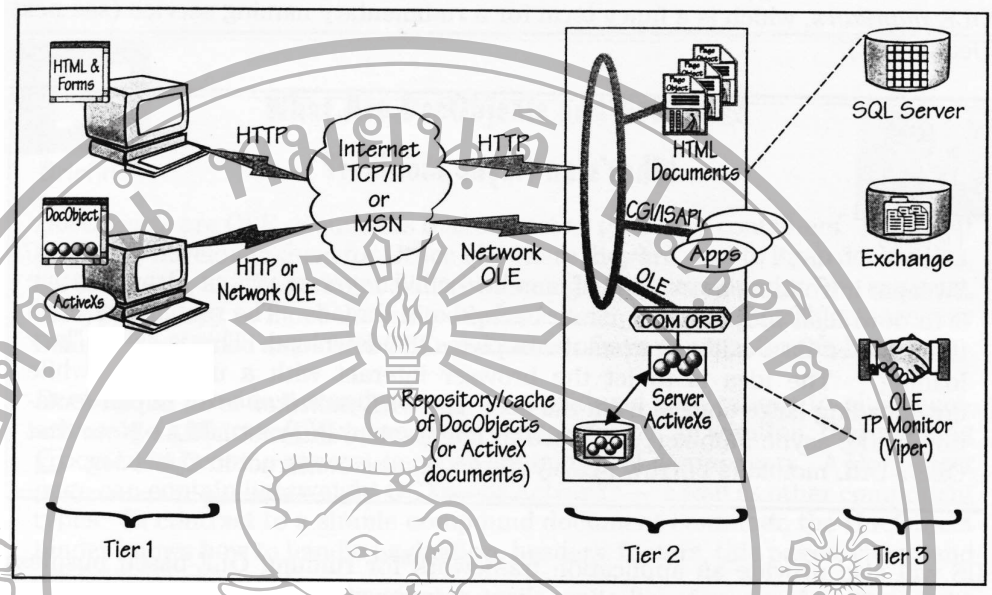


รูปที่ 2.40 แสดงสถาปัตยกรรมแบบ N - Tiered

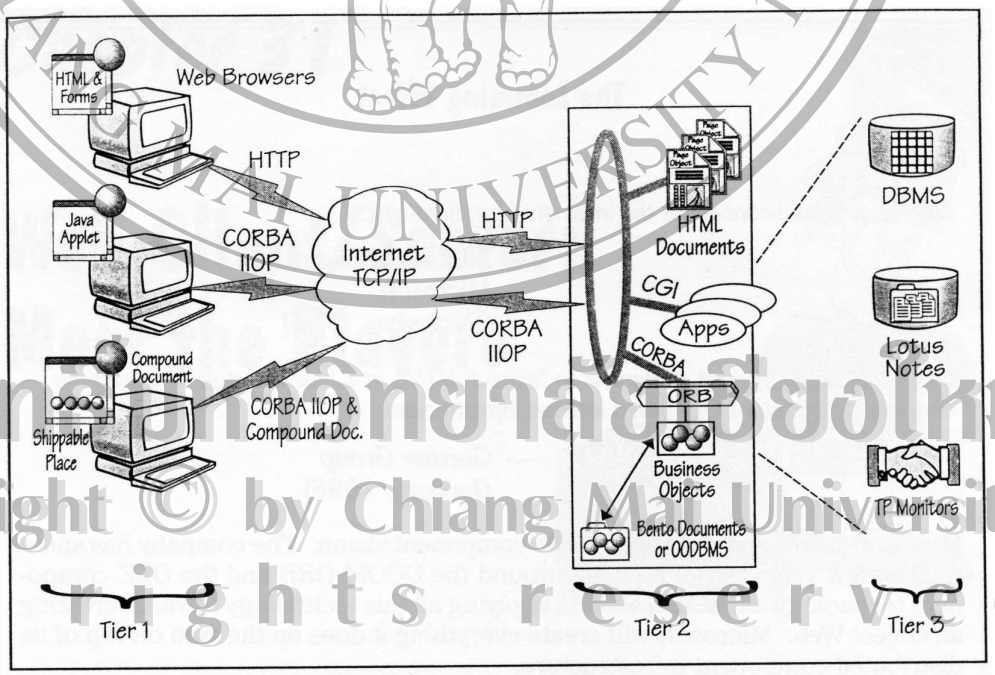


รูปที่ 2.41 แสดงรูปแบบการเชื่อมต่อของ Java Client และ CORBA ORBS แบบ 3 Tier

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved



รูปที่ 2.42 แสดงรูปแบบการเชื่อมต่อของ Microsoft Object Web แบบ 3 Tier



รูปที่ 2.43 แสดงรูปแบบการเชื่อมต่อของ CORBA/Cyberdog Object Web แบบ 3 Tier

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

2.3 งานวิจัยที่เกี่ยวข้อง

จากการศึกษาข้อมูลการวิจัยเรื่องที่เกี่ยวข้อง ผู้วิจัยพบงานที่ตรงกับงานค้นคว้าของผู้วิจัย ดังนี้

ดุษฎี ประเสริฐธิติพงษ์ (2542) ได้ศึกษาค้นคว้าแบบอิสระเชิงวิทยานิพนธ์เรื่อง “ระบบสารสนเทศด้านนักศึกษา สำหรับสำนักทะเบียนและประมวลผล มหาวิทยาลัยเชียงใหม่” งานค้นคว้านี้มีวัตถุประสงค์คือ เพื่อพัฒนาระบบสารสนเทศด้านนักศึกษา สำหรับสำนักทะเบียนและประมวลผล มหาวิทยาลัยเชียงใหม่ เพื่อศึกษาทฤษฎีพื้นฐานด้านระบบสารสนเทศ รวมถึงทฤษฎีการพัฒนาระบบเชิงวิศวกรรมซอฟแวร์ และนำมาประยุกต์ใช้กับระบบงานจริง ในงานค้นคว้านี้ ได้ระบบสารสนเทศด้านนักศึกษา สำหรับสำนักทะเบียนและประมวลผล มหาวิทยาลัยเชียงใหม่ ที่สามารถทำการนำเสนอข้อมูลสารสนเทศในมุมมองหลายมิติ โดยสามารถทำการแสดงผลใน 3 รูปแบบ อันได้แก่ การแสดงผลรายงานในรูปตารางทางจอภาพ การแสดงผลรายงานในรูปตารางทางเครื่องพิมพ์ และการแสดงผลแผนภูมิทางจอภาพ ระบบได้พัฒนาให้ทำงานภายใต้ระบบเครือข่าย โดยมีเครื่องแม่ข่ายสารสนเทศทำหน้าที่จัดเก็บฐานข้อมูลแบบเชิงสัมพันธ์ ผู้ใช้สามารถเข้ามาใช้งานเครือข่ายได้โดยผ่านทางเครื่องลูกข่าย เพื่อเรียกใช้ข้อมูลสารสนเทศที่จัดเก็บที่เครื่องแม่ข่ายเพื่อการแสดงผลสารสนเทศ

กมล รุ่งสอาด (2546) ได้ศึกษาค้นคว้าแบบอิสระเชิงวิทยานิพนธ์เรื่อง “การพัฒนาระบบการจัดการฐานข้อมูลเพื่อรายงานผลการเรียนและการลงทะเบียนของนักศึกษา มหาวิทยาลัยพายัพ” งานค้นคว้านี้มีวัตถุประสงค์คือ เพื่อเป็นแหล่งรวบรวมข้อมูลผลการเรียนและการลงทะเบียนของนักศึกษา ในการเผยแพร่ให้กับนักศึกษา อาจารย์และผู้บริหารของมหาวิทยาลัยพายัพผ่านระบบเครือข่ายอินเทอร์เน็ต และใช้เป็นระบบสารสนเทศเพื่อการวางแผนการเรียนของนักศึกษา จากผลการศึกษา การพัฒนาระบบนี้ ทำให้สามารถตอบสนองความต้องการข้อมูลสารสนเทศของผู้ใช้ได้เป็นอย่างดี และระบบยังสามารถลดภาระงานด้านบันทึกผลการเรียนของนักศึกษาให้กับเจ้าหน้าที่สำนักทะเบียนและบริการการศึกษาได้อีกด้วย

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved

บุญยรัตน์ ศรีอาวัชนาการ (2546) ได้ศึกษาค้นคว้าแบบอิสระเชิงวิทยานิพนธ์เรื่อง “การพัฒนาระบบสารสนเทศเพื่อจัดการข้อมูลการพัฒนาบุคลากรงานผลิตและพัฒนาบุคลากรโรงพยาบาลลำปาง” งานค้นคว้านี้มีวัตถุประสงค์คือ เพื่อสร้างโปรแกรมระบบสารสนเทศเพื่อจัดการข้อมูลการพัฒนาบุคลากร โรงพยาบาลลำปาง โดยใช้ระบบคอมพิวเตอร์ช่วยในการประมวลผล และถูกพัฒนาขึ้นโดยใช้โปรแกรมภาษาวิชวลเบสิค 6.0 ในการเขียนโปรแกรมและใช้โปรแกรมไมโครซอฟท์ แอสเซส 97 ในการเก็บฐานข้อมูล จากการศึกษาพบว่าการประเมินผลการ

ใช้งานเห็นด้วยในระดับดีในเรื่องโปรแกรมระบบฐานข้อมูล มีความเหมาะสมในการออกแบบหน้าจอ ความสะดวกในการบันทึกข้อมูล มีคำอธิบายในการใช้งานและครอบคลุมงานที่ปฏิบัติ

อรรถพร จอมใจธิปี (2546) ได้ศึกษาค้นคว้าแบบอิสระเชิงวิทยานิพนธ์เรื่อง “การพัฒนาระบบบุคลากรสำหรับฝ่ายปฏิบัติการภาคเหนือ การไฟฟ้าฝ่ายผลิตแห่งประเทศไทย”งานค้นคว้านี้มีวัตถุประสงค์คือ เพื่อพัฒนาระบบบุคลากรสำหรับฝ่ายปฏิบัติการภาคเหนือ การไฟฟ้าฝ่ายผลิต และเพื่อเพิ่มประสิทธิภาพระบบบุคลากรฝ่ายปฏิบัติการภาคเหนือ การไฟฟ้าฝ่ายผลิตแห่งประเทศไทย จากการศึกษาพบว่าโปรแกรมการพัฒนาระบบบุคลากรสำหรับฝ่ายปฏิบัติการภาคเหนือที่พัฒนาขึ้นนี้ ง่ายต่อการใช้งานสำหรับพนักงาน ช่วยลดขั้นตอนการปฏิบัติงาน สามารถสืบค้นข้อมูลจากฐานข้อมูลได้อย่างรวดเร็วและสารสนเทศที่ได้รับมีความถูกต้องและทันสมัย

จากการศึกษาข้อมูลเอกสารและงานวิจัยที่เกี่ยวข้องตามลำดับข้างต้น ผู้วิจัยได้ศึกษาทฤษฎี รวมทั้งกระบวนการในการพัฒนาและออกแบบระบบสารสนเทศ รวมถึงขั้นตอนในการดำเนินงานการพัฒนาระบบสารสนเทศ และสามารถนำไปสู่การพัฒนาระบบสารสนเทศงานทะเบียน สถาบันราชภัฏเชียงใหม่ได้อย่างเป็นรูปธรรม

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright © by Chiang Mai University
All rights reserved