บทที่ 3

ทฤษฎี

ระบบฐานข้อมูล (Database System)

ฐานข้อมูลคือที่เก็บรวบรวมกลุ่มข้อมูลของระบบไว้ด้วยกัน เพื่อให้บุคคลและงานประยุกต์ต่าง ๆ สามารถมาดำเนินการอย่างใดอย่างหนึ่ง (ดึง, เพิ่ม, ลบ, แก้ไข) กับข้อมูลนั้น ๆ ได้ มีเรื่องต่าง ๆ ที่ควรรู้ดังนี้ (Date, 1986)

1. สถาปัตยกรรมของฐานข้อมูล (Database Architecture)

ได้มีการกำหนดมาตรฐานขึ้นโดย ANSI (American National Standards Institute) ในปี 1975 เรียกว่าสถาปัตยกรรมสามระดับ (Three-level Architecture) การอธิบายรายละเอียดของระดับต่าง ๆ จะไม่เกี่ยวข้องกับข้อมูลที่เก็บในฐานข้อมูล แต่จะอธิบายโครงสร้างของฐานข้อมูลโดยรวมเพื่อให้สามารถแยกได้นั้นก็เนื่องมาจากการที่ผู้ใช้แต่ละคนอาจมองข้อมูลตัวเดียวกันต่างกัน ดังนั้นถ้ามีการเปลี่ยนแปลงโครงสร้างภายในก็ไม่ควรให้กระทบกับมุมมองของผู้ใช้ สถาปัตยกรรมสามระดับประกอบด้วย

1.1 สคีมาระดับภายนอก (External Schemas Level) จะเป็นระดับที่ใกล้กับผู้ใช้มากที่สุด เป็นสิ่งที่ผู้ใช้คิดเกี่ยวกับข้อมูลจะอธิบายถึงวิว (View) ที่ผู้ใช้สนใจ ข้อมูลที่เก็บจริงอาจมีมากกว่าที่ผู้ใช้ต้องการและข้อมูลตัวเดียวกันผู้ใช้อาจมองไม่เหมือนกัน เช่น ข้อมูลวันที่ (ผู้ใช้คนหนึ่งอาจมองเป็น วัน/เดือน/ปี อีกคนมองเป็น เดือน/วัน/ปี ก็ได้) นอกจากนั้นสิ่งที่ผู้ใช้มองเห็นอาจไม่ได้เก็บจริงในเครื่องแต่ได้จากการคำนวณออกมาส่วนนี้เองจะถูกแปลโดยระบบจัดการฐานข้อมูลเก็บไว้ในพจนานุกรมข้อมูล

1.2 สคีมาระดับเชิงมโนภาพ (Conceptual Schemas Level) จะเป็นตัวที่ใช้เชื่อมระหว่างสคีมาระดับภายนอกกับสคีมาระดับภายใน อธิบายฐานข้อมูลในรายละเอียดโดยรวมทั้งหมด เพื่อเชื่อมกับสิ่งที่ผู้ใช้มอง รูปแบบข้อมูลความสัมพันธ์ เงื่อนไขต่าง ๆ รวมถึงความมั่นคง และความถูกต้องของข้อมูลจะถูกเก็บไว้ด้วย แต่จะไม่ลงลึกถึงการเก็บในเครื่อง

1.3 สคีมาระดับภายใน (Internal Schemas Level) พิจารณาการจัดการระบบการเก็บข้อมูลจริง อธิบายฐานข้อมูลในการเก็บทางกายภาพจริง ๆ มองข้อมูล โดยมุมมองของระบบจัดการฐานข้อมูล ใช้โครงสร้างข้อมูล (Data Structure) และการจัดระเบียนแฟ้ม (File Organization) ในการอธิบาย และจะทำงานร่วมกับระบบปฏิบัติการ (Operating System) ในการเก็บข้อมูลลงที่หน่วยเก็บสำรอง (Secondary Storage)

นอกจากนั้นสถาปัตยกรรมทั้งสามระดับดังกล่าวยังมีความเป็นอิสระของข้อมูลที่เห็นได้คือสคีมาระดับภายนอกจะมีรูปแบบที่คงที่ไม่เปลี่ยนแปลงไป แม้จะมีการเปลี่ยนสคีมาระดับเชิงมโนภาพ เช่น มีการเพิ่มรูปแบบข้อมูลใหม่ หรือมีความสัมพันธ์ใหม่ ๆ เพิ่มขึ้น เรียกว่าความไม่พึ่งพิงทางตรรกะ (Logical Data Independence) และสคีมาระดับเชิงมโนภาพก็เช่นกันจะมีรูปแบบที่คงที่แม้จะมีการเปลี่ยนแปลงทางกายภาพ เช่น เปลี่ยนวิธีเข้าถึงข้อมูล (Access Method) หรือลำดับของข้อมูลที่เก็บอยู่จริงเปลี่ยนไป เรียกว่า ความไม่พึ่งพิงทางกายภาพ (Physical Data Independence) อย่างไรก็ตามโดยทั่วไปแล้วระบบจัดการฐานข้อมูลไม่ได้แบ่งสถาปัตยกรรมทั้ง 3 ระดับจากกันโดยเด็ดขาด ในการทำงานส่วนมากจะให้ผู้ใช้ระบุถึงระดับภายนอกเท่านั้น แล้วระบบจัดการฐานข้อมูลจะเป็นตัวเปลี่ยนการร้องขอ หรือการระบุนั้น ๆ ให้เป็นระดับมโนภาพหรือภายในเองแล้วจึงประมวลผลข้อมูลออกมาให้

2. วงจรระบบฐานข้อมูล (Database System Life Cycle)

คือช่วงเวลาระหว่างการออกแบบ สร้าง และใช้จนถูกแทนที่ด้วยระบบใหม่ที่สมบูรณ์มักจะต้องกินเวลาหลายปี ประกอบด้วยขั้นตอนต่าง ๆ ดังนี้

2.1 ข้อกำหนดระบบ (System Definition) การศึกษาระบบงานปัจจุบัน รายงาน, งานประยุกต์ต่าง ๆ รวมถึงซอฟต์แวร์ที่ใช้ การกำหนดขอบเขต และจุดมุ่งหมายของระบบฐานข้อมูลอันใหม่

2.2 การออกแบบ (Design) จะประกอบไปด้วยการออกแบบทางตรรกะและการออกแบบทางกายภาพ การออกแบบที่ดีจะต้องได้ระบบที่ง่ายต่อการเปลี่ยนแปลงหรือขยายตัวของระบบในอนาคต ผลที่ได้สุดท้ายจากขั้นนี้คือ การออกแบบทางตรรกะและทางกายภาพที่พัฒนาบนระบบจัดการฐานข้อมูลที่ได้เลือกแล้ว

2.3 การติดตั้งระบบ (Implementation) เป็นกระบวนการบรรจุข้อกำหนดของฐานข้อมูลทั้ง 3 ระบบลงในฐานข้อมูลผ่านระบบจัดการฐานข้อมูลที่เลือกแล้วนั่นเอง

2.4 การบรรจุหรือการเปลี่ยนแปลงข้อมูล(Loading or Data Conversion) เพื่อนำข้อมูลเข้าสู่ระบบใหม่ ซึ่งจะนำเข้าโดยตรงหรืออาจแปลงแฟ้มข้อมูลจากระบบเดิมก่อนก็ได้

2.5 การแปลงงานประยุกต์ (Application Conversion) การดัดแปลงงานประยุกต์ที่มีอยู่เดิมให้เข้าสู่ระบบใหม่ และสร้างงานประยุกต์ใหม่ขึ้นมาด้วย

2.6 การทดสอบและตรวจสอบความถูกต้อง (Testing and Validation) ของระบบใหม่

2.7 การปฏิบัติการ (Operation) เป็นขั้นตอนการรวมงานประยุกต์ทุกงานเข้าด้วยกัน แล้วจัดการเกี่ยวกับสิทธิและขอบเขตการทำงานของผู้ใช้แต่ละคน กำหนดใครจะเป็นคนทำการกู้ข้อมูลและทำแฟ้มข้อมูลสำรอง

2.8 การเฝ้าคุม และบำรุงรักษา (Monitoring and Maintenance) ดูแลการเติบโตและขยายตัวที่จะเกิดขึ้นกับข้อมูลและซอฟต์แวร์

ขั้นตอนการออกแบบ ติดตั้งและนำข้อมูลเข้าเป็นจุดสำคัญที่สุดในวงจร ขั้นตอนต่าง ๆ เหล่านี้ ในบางครั้งก็แยกกันโดยเด็ดขาด บางครั้งก็อาจทับซ้อนกันได้ ในที่นี้จะอธิบายรายละเอียดเฉพาะการออกแบบเท่านั้น

3. การออกแบบฐานข้อมูล

กระบวนการในการออกแบบทางตรรกะและทางกายภาพของฐานข้อมูลจะประกอบด้วยงาน 2 ส่วน คือ การออกแบบในส่วนของเนื้อหาข้อมูล และ โครงสร้างของฐานข้อมูล (Content and Structure of Database) อีกส่วนคือ การออกแบบในเรื่องการประมวลผลฐานข้อมูลและซอฟต์แวร์ของงานประยุกต์ (Database Processing and Software Application) การออกแบบงานทั้งสองส่วนนั้นมักจะทำไปด้วยกันแยกจากกันได้ยากมากเพราะในบางครั้งเราจะสามารถระบุเนื้อหาข้อมูลได้จากการศึกษางานประยุกต์ หรืออาจกำหนดงานประยุกต์โดยอ้างถึงโครงสร้างของฐานข้อมูล

3.1 จุดประสงค์ของการออกแบบ

3.1.1 เตรียมข้อมูลที่จำเป็น เพื่อสนองตอบตามความต้องการของผู้ใช้และงานประยุกต์ที่เข้ามา

3.1.2 เตรียมโครงสร้างข้อมูลที่เป็นธรรมชาติ และง่ายต่อการเข้าใจมากที่สุด

3.1.3 เพื่อรองรับต่อความต้องการในการประมวลผล จัดเก็บข่าวสารอย่างมีประสิทธิภาพ และใช้เวลาในการตอบสนองที่เหมาะสม

3.2 การออกแบบฐานข้อมูลแบ่งงานออกได้เป็นหลายช่วง และไม่จำเป็นต้องทำโดยเรียงลำดับ อาจทำหลายช่วงควบคู่กันไปก็ได้ ประกอบด้วย 6 ช่วงด้วยกันคือ

3.2.1 การเก็บรวบรวมและวิเคราะห์(Requirements Collection and Analysis)

3.2.2 การออกแบบฐานข้อมูลเชิงมโนภาพ (Conceptual Database Design)

3.2.3 การเลือกระบบจัดการฐานข้อมูล (Choice of a DBMS)

3.2.4 การแปลงให้อยู่ในโมเดลของระบบจัดการฐานข้อมูล (Data Model Mapping หรือบางทีเรียกว่า Logical Database Design)

3.2.5 การออกแบบฐานข้อมูลทางกายภาพ (Physical Database Design)

3.2.6 การติดตั้ง และนำฐานข้อมูลมาใช้จริง (Database Implementation)

4. ความมั่นคงของฐานข้อมูล (Database Security)

ฐานข้อมูลนั้นก็เป็นที่ทราบกันดีอยู่แล้วว่า เป็นที่เก็บข้อมูลที่มีค่าจำนวนมาก ดังนั้นจึงต้องมีการรักษาความมั่นคงของข้อมูลเหล่านั้น ซึ่งในการออกแบบผู้ออกแบบจะต้องพิจารณาในเรื่องเหล่านี้ด้วย

4.1 วัตถุประสงค์ของการรักษาความมั่นคงของข้อมูล

4.1.1 ป้องกันและขัดขวางไม่ให้ผู้ไม่มีสิทธิเข้าไปดึง,แก้ไข,เปลี่ยนแปลงข้อมูล หรือ ปรับโครงสร้างของฐานข้อมูล

4.1.2 กีดขวางไม่ให้ผู้มีสิทธิขยายวงในการสืบค้น หรือทำความเข้าใจกับข้อมูลในส่วนที่เขาไม่มีสิทธิ

4.1.3 ทำให้เงินลงทุนและอัตราการเสี่ยงของผู้ที่บุกรุกสูงกว่าประโยชน์ที่จะได้

4.2 สาเหตุของการบุกรุกมีได้ทั้งอุบัติเหตุ และโดยตั้งใจ ผู้บุกรุกนั้นเป็นได้ทั้งบุคคลภายในและภายนอกหน่วยงาน การบุกรุกแบ่งได้ดังนี้

4.2.1 การบุกรุกโดยอุบัติเหตุ

4.2.1.1 เกิดจากข้อจำกัดของการกำหนดสิทธิในการใช้ข้อมูลของระบบจัดการฐานข้อมูลหรือระบบปฏิบัติการทำให้ผู้ใช้ระดับเดียวกับผู้มีสิทธิดึงข้อมูลนั้นโดยไม่ตั้งใจ

4.2.1.2 การส่งข่าวสารถึงผู้ใช้ผิดคน ทำให้ผู้ไม่มีสิทธิรู้ข้อมูล

4.2.1.3 ความผิดพลาดของระบบสื่อสารมีผลไปเชื่อมผู้ใช้เข้าไปในเครือข่ายของผู้อื่น

4.2.1.4 ความผิดพลาดของระบบปฏิบัติการ ทำให้เกิดการเขียนทับหรือส่งแฟ้มข้อมูลผิดให้กับผู้ใช้คนอื่น

4.2.2 การบุกรุกโดยตั้งใจ

4.2.2.1 การลอบต่อสายในการสื่อสาร

4.2.2.2 การใช้สัญญาณอิเลคโทรนิคดึงสัญญาณจากเครื่องอื่น

4.2.2.3 แอบดูข้อมูลจากหน้าจอหรือสิ่งพิมพ์ของผู้ใช้อื่นที่มีสิทธิ

4.2.2.4 การเข้าถึงข้อมูลจากผู้มีสิทธิในระดับต่ำกว่า

4.2.2.5 การขโมยสิ่งที่ใช้บันทึกข้อมูล เช่น เทปออกจากห้องคอมพิวเตอร์

4.3 การรักษาความมั่นคงแบ่งได้เป็น 2 ทาง คือ

4.3.1 การรักษาความมั่นคงทางกายภาพ ได้แก่ การติดตั้งระบบรักษาความมั่นคงต่าง ๆ ก่อนที่ผู้บุกรุกจะเข้ามาเปิดเครื่องได้ เช่น การมียามรักษาการณ์, มีกุญแจล็อคเครื่อง, ให้เซ็นชื่อหรือ พิมพ์ลายนิ้วมือซึ่งสิ่งเหล่านี้จะอยู่นอกเหนือขอบเขตของผู้ออกแบบ แต่ผู้ออกแบบก็ควรได้แนะนำไว้

4.3.2 การรักษาความมั่นคงทางซอฟต์แวร์ การทำไว้ในโปรแกรม เช่น

4.3.2.1 การให้อำนาจ (Authorization) คือการตรวจสอบหลักฐานผู้ใช้ว่าเป็นตัวจริงที่มีสิทธิมาร้องขอข้อมูลหรือไม่ มักจะติดตั้งการรักษาความมั่นคงไว้ที่ระดับระบบปฏิบัติการวิธีที่นิยมมากที่สุดคือให้ใส่รหัสผ่าน (Password) แต่ถึงอย่างไรก็ไม่ปลอดภัยนัก บางแห่งอาจมีการเพิ่มเติมให้ใช้บัตร หรือกุญแจเฉพาะเมื่อจะเข้าสู่ระบบ (Login) ในระบบจัดการฐานข้อมูลส่วนมากจะมีระบบการรักษาความมั่นคงนี้ให้สำหรับผู้ใช้แต่ละคน เพื่อกำหนดผู้มีสิทธิใช้ข้อมูลของตนได้เอง เรียกว่า ภาษาในการให้อำนาจ (Authorization Language) เช่น ในเอสคิวแอล (SQL) มีคำสั่ง grant เป็นต้น ซึ่งเป็นการให้อำนาจในการกระทำกับตารางนั้นกับผู้ใช้อื่น ๆ

4.3.2.2 การควบคุมการเข้าถึง (Access Control) คือการทำให้แน่ใจว่าข้อมูลหรือสิ่งต่าง ๆ ในระบบจะถูกเข้าถึงได้เฉพาะทิศทางที่ต้องผ่านการตรวจสอบการให้อำนาจเท่านั้นซึ่งวิธีนี้จะทำได้เมื่อมีการติดตั้งการให้อำนาจแล้วเท่านั้น เราอาจทำโดยสร้างเป็นตารางการเข้าถึง (Access Control Matrix) เพื่อให้ง่ายต่อการควบคุมโดยแต่ละ

สดมภ์แทนเป้าหมายในระบบ เช่น ชื่อตารางหรือวิว ส่วนแถวต่าง ๆ แทนรหัสประจำตัวของผู้ใช้หรือกลุ่มผู้ใช้ และค่าในแต่ละช่องที่สัมพันธ์กัน ก็คืออำนาจในการเข้าถึงที่กำหนดให้ผู้ใช้สำหรับตาราง หรือวิวนั้น ๆ ได้แก่ อำนาจการลบ เพิ่ม เปลี่ยนแปลง อ่านข้อมูล

4.3.2.3 วิว (View) การสร้างวิวในระบบจัดการฐานข้อมูลโดยทั่วไปจะมีคำสั่งที่ใช้ในการสร้างวิวจากตารางที่มีในระบบ เพื่อให้ได้ข้อมูลเฉพาะในส่วนที่ผู้ใช้จะมีสิทธิ โดยผู้จัดการฐานข้อมูลจะเป็นคนทำและให้สิทธิในการเข้าถึงวิวนั้น ๆ ให้แก่ผู้ใช้อื่น ๆ

4.3.2.4 การเข้ารหัสลับ (Data Encryption) เป็นการเก็บข้อมูลที่จะใช้ระบบที่เรียกว่า Cipher System ที่ประกอบไปด้วยการเข้ารหัสโดยเราจะนำข้อมูลในรูปแบบปกติ (Plain Text) มาผ่านขั้นตอนวิธี (Algorithm) ในการเข้ารหัสลับซึ่งจะมีการกำหนดคีย์ในการเข้ารหัสลับด้วย จึงจะได้ข้อมูลในรูปแบบที่เรียกว่าไซเฟอร์เทกซ์ (Cipher Text) และเราจะเก็บข้อมูลไว้ในระบบในรูปแบบนี้ เวลาจะต้องใช้ก็จะนำไปผ่านขั้นตอนวิธีในการถอดรหัสซึ่งก็จะมีคีย์สำหรับการถอดรหัสเช่นกัน ดังนั้นถ้ามีผู้บุกรุกมาขโมยข้อมูลไปก็จะใช้ไม่ได้ทันทีต้องทำการถอดรหัสก่อนซึ่งยุ่งยากและอาจเสียเวลามาก ขั้นตอนวิธีในการเข้ารหัสที่เป็นที่นิยม และตั้งเป็นมาตรฐานไว้ คือ ดีอีเอส (DES : Data Encryption Standard) ซึ่งสามารถสร้างไซเฟอร์เทกซ์ได้แตกต่างกันมากกว่า 2^{56} แบบ

4.3.2.5 การลงบันทึกความมั่นคง (Security Logs) เป็นการบันทึกเกี่ยวกับผู้ที่ตั้งใจจะฝ่าฝืนเข้าสู่ระบบ โดยจะบันทึกข้อมูลเหล่านั้นไว้ในแฟ้มลงบันทึกหรือมีการส่งข่าวไปให้ผู้ควบคุมเครื่อง หรือผู้บริหารฐานข้อมูลทราบ

4.3.2.6 การทำหลักฐานการตรวจสอบ (Audit Trail) เป็นการเก็บข่าวสารเกี่ยวกับว่าใครมาดึงข้อมูล หรือมาใช้เครื่องเทอร์มินอลเมื่อเวลาใด

<u>ฐานข้อมูลแบบกระจาย</u>

ฐานข้อมูลแบบกระจาย (ดวงแก้ว สวามิภักดิ์, 2534) คือระบบฐานข้อมูลที่มีฐานข้อมูลเก็บอยู่ในระบบคอมพิวเตอร์หลาย ๆ เครื่องที่ติดตั้งอยู่ตามที่ต่าง ๆ ซึ่งเรียกว่า สถานี (SITES) หรือ จุดต่อกิ่ง (NODES) โดยที่ระบบคอมพิวเตอร์เหล่านี้มีการสื่อสารกันได้ และผู้ใช้ไม่จำเป็นต้องรับทราบว่าข้อมูลที่ตนต้องการนั้นจัดเก็บอยู่บนเครื่องใด กล่าวคือในทางกายภาพข้อมูลจะถูกเก็บกระจายอยู่ตามเครื่องต่าง ๆ ภายในระบบเครือข่ายคอมพิวเตอร์ แต่ในแง่มุมของผู้ใช้จะเสมือนหนึ่งว่ากำลังใช้ระบบฐานข้อมูลแบบรวมศูนย์อยู่ตามปกติ นอกจากนี้แต่ละสถานียังมีความสามารถในการดำเนินงาน และ ควบคุมระบบงานของตนเองได้อย่างอิสระอีกด้วย

1. ส่วนประกอบของระบบฐานข้อมูลแบบกระจาย สามารถแบ่งได้เป็นสองส่วน คือ (Tramer, 1991)

1.1 ส่วนของฮาร์ดแวร์ เป็นปัจจัยหลักที่เป็นตัวบอกถึงความแตกต่างระหว่างฐานข้อมูลแบบกระจาย และฐานข้อมูลแบบรวมศูนย์ คือ

1.1.1 ระบบฐานข้อมูลแบบกระจาย ประกอบด้วยเครื่องคอมพิวเตอร์มากกว่าหนึ่งเครื่อง หรือมากกว่าหนึ่งสถานี

1.1.2 สถานีต่าง ๆ เหล่านี้ จะถูกเชื่อมต่อเข้าด้วยกันโดยใช้เครือข่ายงานเฉพาะที่ (Local Area Network-LAN) หรือเครือข่ายงานบริเวณกว้าง (Wide Area Network-WAN) เพื่อให้สามารถส่งข้อมูลและคำสั่งระหว่างสถานีได้

1.2 ส่วนของซอฟต์แวร์ เป็นส่วนประกอบที่จำเป็นต่อการสร้างฐานข้อมูลแบบกระจาย มีดังนี้คือ

1.2.1 ส่วนการสื่อสารข้อมูล (Data Communication Component-DC) หมายถึง ซอฟต์แวร์สำหรับจัดการเกี่ยวกับระบบสื่อสารพื้นฐานต่าง ๆ ที่ใช้ในการส่งข้อมูล และคำสั่งระหว่างสถานี

1.2.2 ตัวจัดการฐานข้อมูลภายใน (Local Database Management Component-DBMS) หมายถึง ระบบจัดการฐานข้อมูลมาตรฐานนั่นเอง ซึ่งเป็นตัวควบคุมฐานข้อมูลภายในของแต่ละสถานี

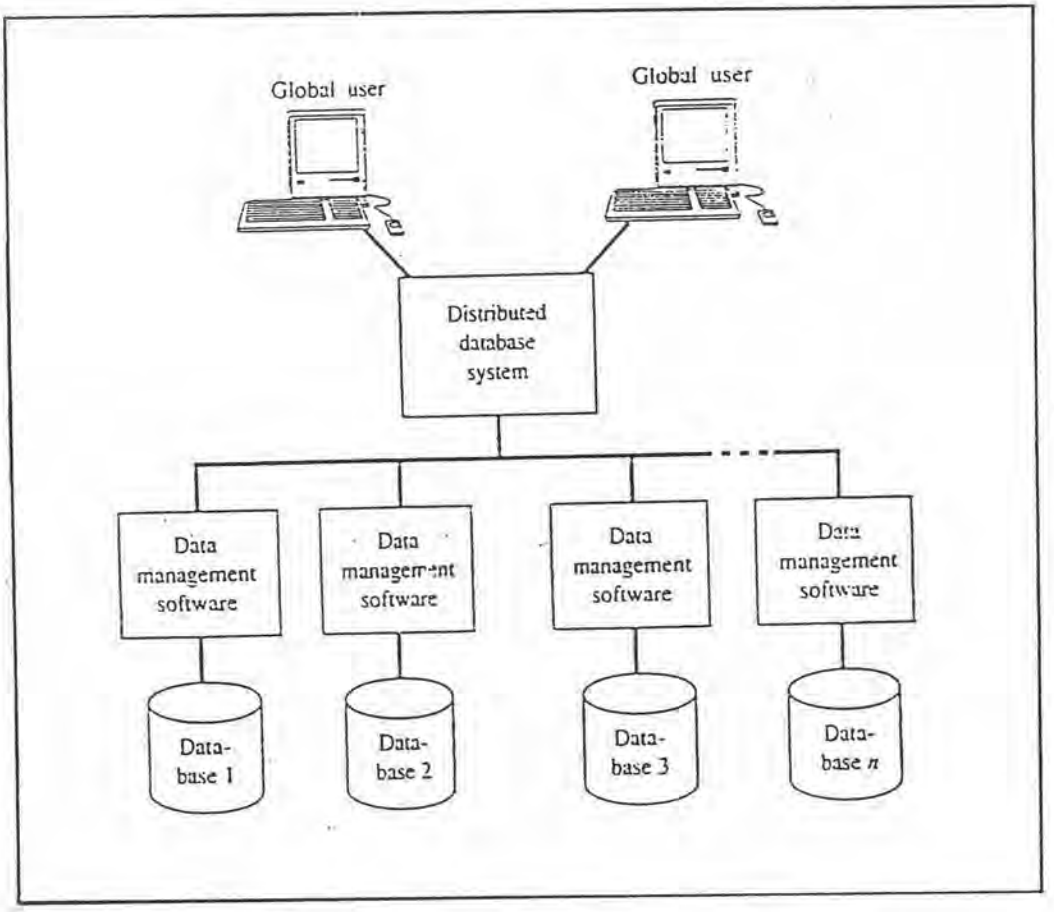
1.2.3 ส่วนพจนานุกรมข้อมูลส่วนรวม (Global Data Dictionary-DD) เป็นที่เก็บสารสนเทศของฐานข้อมูล ที่กระจายอยู่ตามสถานีต่าง ๆ ว่าข้อมูลเหล่านั้นถูกเก็บอยู่ที่สถานีใดบ้าง พร้อมทั้งรายละเอียดอื่น ๆ ของข้อมูลที่มีอยู่ในระบบ

1.2.4 ส่วนของฐานข้อมูลแบบกระจาย (Distributed Database Component-DDB) เป็นตัวจัดการเกี่ยวกับระบบทั้งหมด เช่น ตัวเชื่อมประสานกับผู้ใช้ (User Interface) โดยทำให้ผู้ใช้มองเห็นราวกับว่า ข้อมูลทั้งหมดถูกเก็บอยู่ที่สถานีของตนเพียงแห่งเดียวเพื่อประมวลผลการสอบถามที่เกิดขึ้น นอกจากนั้นยังทำหน้าที่ ในการควบคุมความถูกต้องตรงกันของข้อมูลชุดต่าง ๆ และการกู้ข้อมูล (Recovery) ด้วย

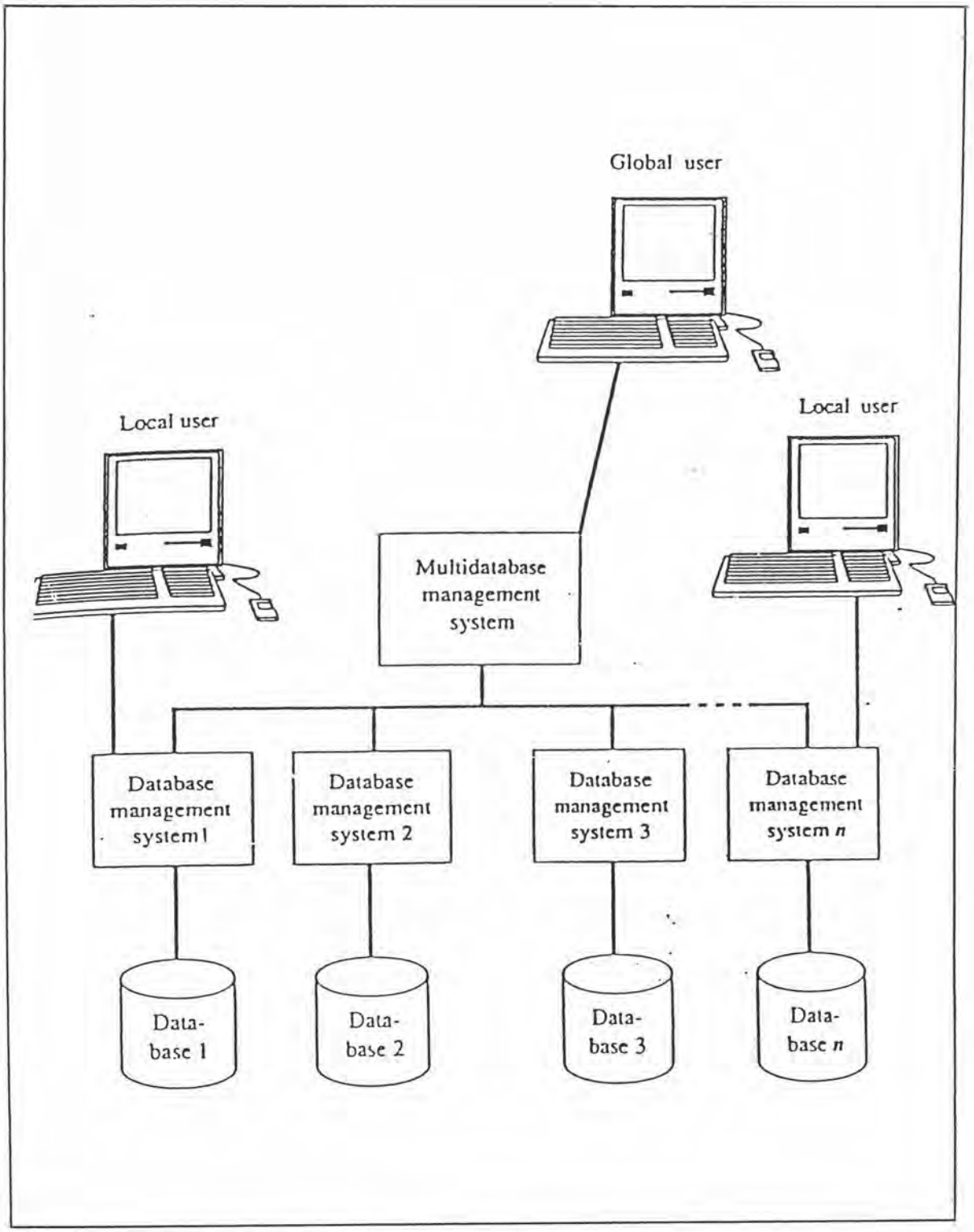
2. ประเภทของฐานข้อมูลแบบกระจาย อาจแบ่งอย่างกว้าง ๆ ได้สองประเภท คือ (David and Jane, 1992)

2.1 ชนิดที่มีความเหมือนกัน (Homogeneous) หมายถึงระบบที่เครื่องที่อยู่ตามสถานีต่าง ๆ ใช้ระบบจัดการฐานข้อมูลชนิดเดียวกัน ดังแสดงในรูปที่ 3.1

2.2 ชนิดที่ไม่มีความเหมือนกัน (Heterogeneous) หมายถึง ระบบที่เครื่องที่อยู่ตามสถานีต่าง ๆ ใช้ระบบจัดการฐานข้อมูลต่างชนิดกัน หรือ เรียกว่า "MULTIDATABASE MANAGEMENT SYSTEM" ซึ่งยิ่งมีความหลากหลายมากก็จะทำให้ระบบจัดการฐานข้อมูลต้องทำงานซับซ้อนมากขึ้นดังรูปที่ 3.2



รูปที่ 3.1 แสดงฐานข้อมูลแบบกระจายชนิดที่มีความเหมือนกัน



รูปที่ 3.2 แสดงฐานข้อมูลแบบกระจายชนิดที่ไม่มีความเหมือนกัน

3. สถาปัตยกรรมของระบบฐานข้อมูลแบบกระจาย (Distributed Database Architecture) (ปรีชา แสงอาสภวิริยะ, 2525)

3.1 วัตถุประสงค์ของการออกแบบฐานข้อมูลแบบกระจาย

3.1.1 วัตถุประสงค์ด้านฐานข้อมูล คือ การควบคุมและการใช้ข่าวสาร ซึ่งเป็นทรัพยากรขององค์การ เพื่อปรับปรุงความพร้อมของระบบ ซึ่งหมายถึงความสามารถในการเข้าถึงข้อมูล และการใช้ข้อมูลร่วมกันได้ของผู้ใช้และปรับปรุงความถูกต้องของข้อมูล ฐานข้อมูลที่นิยมใช้ คือ ฐานข้อมูลเชิงสัมพันธ์

3.1.2 วัตถุประสงค์เพื่อการสื่อสาร คือ ความพยายามลดขนาดและจำนวนของข่าวสาร และความยาวของเส้นทางซึ่งข่าวสารจะถูกส่งผ่านให้น้อยที่สุด

3.2 สถาปัตยกรรมของระบบฐานข้อมูลแบบกระจาย สามารถแบ่งการจัดองค์กรของข้อมูล (Data Organization) ออกเป็นระดับต่าง ๆ ดังแสดงในรูปที่ 3.3 (Tramer, 1991)

3.2.1 สคีมาระดับภายนอก (External Schema) คือ ส่วนของข้อมูลที่ผู้ใช้ หรือโปรแกรมประยุกต์มองเห็น

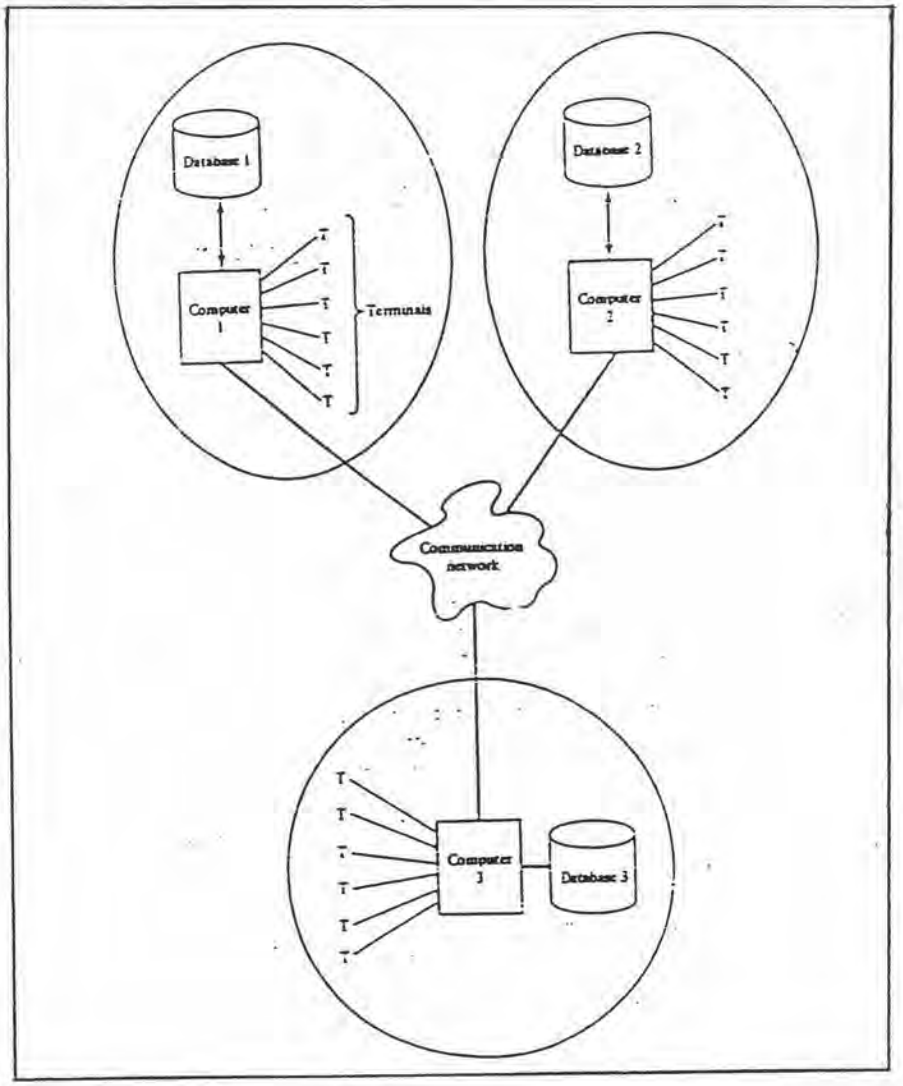
3.2.2 สคีมาเชิงมโนภาพระดับรวม (Global Conceptual Schema) คือส่วนเก็บรายละเอียดของข้อมูลทั้งหมด ในระบบฐานข้อมูลแบบกระจาย

3.2.3 สคีมาแบบแตกกระจาย (Fragmentation Schema) เนื่องจากสคีมาเชิงมโนภาพในระดับรวม สามารถถูกแบ่งออกเป็นส่วน ๆ โดยแต่ละส่วนจะไม่คาบเกี่ยวกัน ส่วนต่าง ๆ เหล่านี้จะถูกกระจายไปตามสถานีต่าง ๆ ในเครือข่าย ดังนั้นในระดับนี้จึงเป็นการแปลง (Map) จากสคีมาเชิงมโนภาพในระดับรวมให้เป็นส่วนเล็ก ๆ หลาย ๆ ส่วนนั่นเอง

3.2.4 สคีมาแบบระบุตามสถานที่จัดเก็บ (Allocation Schema) ระดับนี้ จะเป็นการเก็บรายละเอียดว่าส่วนของฐานข้อมูลต่าง ๆ ถูกเก็บอยู่ที่สถานีใดบ้าง

3.2.5 สคีมาเชิงมโนภาพระดับภายใน (Local Conceptual Schema) ระดับนี้เป็นการแปลงส่วนต่าง ๆ ของฐานข้อมูลที่แตก

กระจาย และถูกกำหนดสถานที่ในการจัดเก็บแล้ว ให้เป็นสิ่งซึ่ง ระบบจัดการข้อมูลภายในรู้จัก



รูปที่ 3.3 แสดงสถาปัตยกรรมของฐานข้อมูลแบบกระจาย

4. การออกแบบฐานข้อมูลแบบกระจาย (Distributed Database Design)

การออกแบบฐานข้อมูลแบบกระจาย หมายถึง การออกแบบสคีมาเชิงมโนภาพในระดับรวมและการออกแบบสคีมาเชิงมโนภาพระดับภายใน โดยกำหนดลักษณะการกระจายของฐานข้อมูลให้เหมาะสมกับการเรียกใช้

4.1 แนวทางหลักที่ใช้ในการออกแบบฐานข้อมูลแบบกระจาย มีอยู่สองแนวทาง ดังนี้

4.1.1 การออกแบบจากบนลงล่าง (Top-down Approach) วิธีนี้เริ่มจากการออกแบบสคีมาเชิงมโนภาพในระดับรวม แล้วจึงออกแบบลักษณะการกระจายข้อมูลเพื่อเป็นการกำหนดสคีมาเชิงมโนภาพระดับภายใน ต่อมาจึงทำการออกแบบฐานข้อมูลเชิงกายภาพ (Physical Database Design) ซึ่งเป็นการแปลงจากสคีมาเชิงมโนภาพภายในที่ได้ ไปยังอุปกรณ์ที่ใช้เก็บข้อมูลจริง วิธีนี้จะเหมาะกับการสร้างระบบฐานข้อมูลแบบกระจายขึ้นใหม่โดยที่ยังไม่เคยมีระบบฐานข้อมูลระบบใด ๆ มาก่อน

4.1.2 การออกแบบจากล่างขึ้นบน (Bottom-up Approach) วิธีนี้เหมาะสำหรับการพัฒนาฐานข้อมูลที่มีอยู่ให้เป็นฐานข้อมูลแบบกระจาย ซึ่งต้องรวบรวมระบบฐานข้อมูลหลาย ๆ ระบบที่มีอยู่เข้าเป็นระบบเดียวกัน นั่นคือเป็นการรวบรวมสคีมาเชิงมโนภาพระดับภายในของแต่ละระบบให้เป็นสคีมาเชิงมโนภาพในระดับรวม

4.2 โมเดลข้อมูล (Data Model)

การออกแบบโมเดลข้อมูลในระบบฐานข้อมูลแบบกระจายมีลักษณะเช่นเดียวกันกับการออกแบบโมเดลข้อมูลในระบบฐานข้อมูลแบบรวมศูนย์

4.3 การออกแบบการแตกกระจาย (Designing the Fragmentation) ได้แก่การจัดแยกข้อมูลออกเก็บตามสถานีต่าง ๆ เพื่อให้ได้ประโยชน์ใช้งานสูงสุด ซึ่งโดยทั่ว ๆ ไปก็จะเก็บไว้ยังสถานีที่เกิดการเรียกใช้ข้อมูลนั้น ๆ บ่อยที่สุดในที่นี้สมมุติโมเดลที่ใช้เป็นรีเลชันนัลโมเดลดังนั้นจึงสามารถแบ่งส่วนฐานข้อมูลแบบกระจายออกเป็นส่วนต่าง ๆ ตามลักษณะการใช้งานของภายในได้ดังนี้ คือ

4.3.1 การแตกกระจายตามแนวนอน(Horizontal Fragmentation) เป็นการแบ่งตารางความสัมพันธ์ระดับรวมออกเป็นส่วน ๆ เราสามารถกำหนดการแตกกระจายตามวิธีนี้ได้โดยทำการเลือก (Selection Operation) จากตารางความสัมพันธ์ระดับรวมดังตัวอย่างจากรูปที่ 3.4 สมมุติตารางความสัมพันธ์ระดับรวมของ JOB หมายถึง งานซึ่งจะถูกจำแนกตามงบประมาณออกเป็น 2 ส่วนดังแสดงในรูปที่ 3.5 คือ ตารางความสัมพันธ์ย่อย JOB_1 แสดงถึงงานที่ใช้งบประมาณน้อยกว่า $200,000 และตารางความสัมพันธ์ย่อย JOB_2 ซึ่งเป็นงานที่ต้องใช้งบประมาณมากกว่า $200,000 เป็นต้น

JOB

JOBNO	JOBNAME	BUDGET	LOCATION
J1	Instrument	150000	Montreal
J2	DB develop.	135000	New York
J3	CAD/CAM	250000	New York
J4	Maintenance	310000	Paris

รูปที่ 3.4 แสดงตารางความสัมพันธ์ระดับรวม

JOB_1

JOBNO	JOBNAME	BUDGET	LOCATION
J1	Instrument	150000	Montreal
J2	DB develop	135000	New York

JOB_2

JOBNO	JOBNAME	BUDGET	LOCATION
J3	CAD/CAM	250000	New York
J4	Maintenance	310000	Paris

รูปที่ 3.5 แสดงการแตกกระจายฐานข้อมูลตามแนวนอน

4.3.2 การแตกกระจายตามแนวตั้ง (Vertical Fragmentation) เป็นการแบ่งส่วนตารางความสัมพันธ์ระดับรวมตามแอตตริบิว โดยการโปรเจคชัน (Projection Operation) ดังแสดงในรูปที่ 3.6 คือตารางความสัมพันธ์ย่อย JOB_1 ประกอบด้วยข้อมูลเกี่ยวกับงบประมาณของงานต่าง ๆ ส่วนตารางความสัมพันธ์ย่อย JOB_2 ประกอบด้วยข้อมูลชื่อ และสถานที่ตั้งของงาน

JOB_1

JOBNO	BUDGET
J1	150000
J2	135000
J3	250000
J4	310000

JOB_2

JOBNO	JOBNAME	LOCATION
J1	Instrument	Montreal
J2	DB develop.	New York
J3	CAD/CAM	New York
J4	Maintenance	Paris

รูปที่ 3.6 แสดงการแตกกระจายฐานข้อมูลตามแนวตั้ง

4.3.3 การแตกกระจายแบบผสม (Hybrid Fragmentation) การแตกกระจายตามวิธีนี้เป็นการผสมระหว่างการแตกกระจายตามแนวนอนและแนวตั้ง ดังนั้นการแตกกระจายฐานข้อมูลตามวิธีนี้จึงต้องใช้ทั้งการเลือก และทำโปรเจคชันร่วมกัน จากตัวอย่างข้างต้นเราอาจจะทำโปรเจคชันเอาเฉพาะแอตตริบิวหมายเลขงานและงบประมาณของงาน ดังแสดงในรูปที่ 3.7 ซึ่งได้มาจากการเลือกเฉพาะงาน ซึ่งตั้งอยู่ที่เมืองนิวยอร์คเท่านั้น

JOB_3

JOBNO	BUDGET
J2	135000
J3	250000

รูปที่ 3.7 แสดงการแตกกระจายฐานข้อมูลแบบผสม

4.4 การกำหนดสถานที่เก็บข้อมูล (The Allocation of Fragments)

หลังจากทำการแตกกระจายฐานข้อมูลแล้ว ขั้นตอนต่อไปก็คือ การกระจายข้อมูลออกไปเก็บตามสถานีต่าง ๆ โดยเลือกเก็บส่วนต่าง ๆ ของฐานข้อมูลเหล่านั้น ในสถานีที่มีการใช้ข้อมูลบ่อยที่สุด การจัดเก็บฐานข้อมูลไปตามสถานีต่าง ๆ มี 4 แบบดังนี้

4.4.1 ฐานข้อมูลแบบรวมศูนย์ (Centralized Database) เป็นวิธีการเก็บฐานข้อมูลทั้งหมด และระบบจัดการฐานข้อมูลไว้ที่คอมพิวเตอร์ศูนย์กลาง แต่ผู้ใช้จะถูกกระจายออกไปตามจุดต่าง ๆ การเก็บแบบนี้จะเหมาะสมสำหรับเครือข่ายที่มีโครงสร้างแบบลำดับชั้น (Hierarchical Network Structure) ซึ่งจะมีงานส่งเข้ามาเพื่อทำการประมวลผลโดยใช้ข้อมูลในฐานข้อมูลแบบแบทช์ (Batch) สำหรับการประมวลผลในลักษณะการสอบถามข้อมูล ผลลัพธ์จะถูกส่งออกไปยังสถานีต่าง ๆ ตามความต้องการ

4.4.2 ฐานข้อมูลแบบเก็บบางส่วน (Partitioned Database) วิธีนี้เป็นการกระจายส่วนของฐานข้อมูลไปยังสถานีต่าง ๆ กัน โดยแต่ละส่วนที่แบ่งออกมาจะไม่มีข้อมูลที่ซ้ำกันสำหรับการแบ่งข้อมูลในระบบฐานข้อมูลแบบกระจาย อาจจะแบ่งข้อมูลตามการใช้งานของข้อมูลหรือแบ่งตามสภาพทางภูมิศาสตร์ที่ข้อมูลอยู่ก็ได้

4.4.3 ฐานข้อมูลแบบเก็บซ้ำ (Replicated Database) ส่วนหนึ่งหรือทั้งหมดของฐานข้อมูลจะถูกสำเนาไปไว้ที่สถานีตั้งแต่สองสถานีขึ้นไป ตัวอย่างที่เห็นชัดสำหรับการใช้ฐานข้อมูลชนิดนี้ ได้แก่ ธุรกิจธนาคารซึ่งแต่ละสถานีจะมีฐานข้อมูลที่มีชื่อบัญชีของลูกค้าของตัวเองก่อนที่จะเริ่มวันใหม่ ฐานข้อมูลเฉพาะที่ (Local Database) จะถูกส่งมาจากศูนย์กลางมายังสถานีต่าง ๆ ในระหว่างวันการทำงานต่าง ๆ จะกระทำลงบนฐานข้อมูลเฉพาะที่ของสถานีนั้น ๆ เมื่อสิ้นวันทำรายการ (Transaction) จะถูกรวม และส่งเข้าแก้ไขเปลี่ยนแปลงที่ฐานข้อมูลศูนย์กลาง

4.4.4 ฐานข้อมูลแบบผสม (Mixed Database) วิธีนี้แต่ละสถานีจะเก็บส่วนของฐานข้อมูลที่แตกต่างกันออกไปทั้งนี้ขึ้นอยู่กับรูปแบบการใช้ข้อมูล เช่น ข้อมูลส่วนใดที่ใช้เฉพาะสถานีนั้น ๆ ก็จะถูกเก็บแบบบางส่วน แต่หากมีข้อมูลที่ถูกเรียกใช้ร่วมกันโดยสถานีต่าง ๆ แล้ว ข้อมูลส่วนนั้นจะถูกเก็บแบบซ้ำ

สมมุติส่วนของฐานข้อมูลประกอบด้วย $F = \{F_1, F_2, \ldots, F_n\}$ และสถานีในเครือข่ายประกอบด้วย $S = \{S_1, S_2, \ldots, S_n\}$

ดังนั้นการจะเลือกวิธีการกำหนดสถานที่เก็บข้อมูลแบบใด จะต้องพิจารณาสิ่งต่อไปนี้

1. ต้นทุนต่ำสุด ต้นทุนในที่นี้หมายถึงต้นทุนในการเก็บ

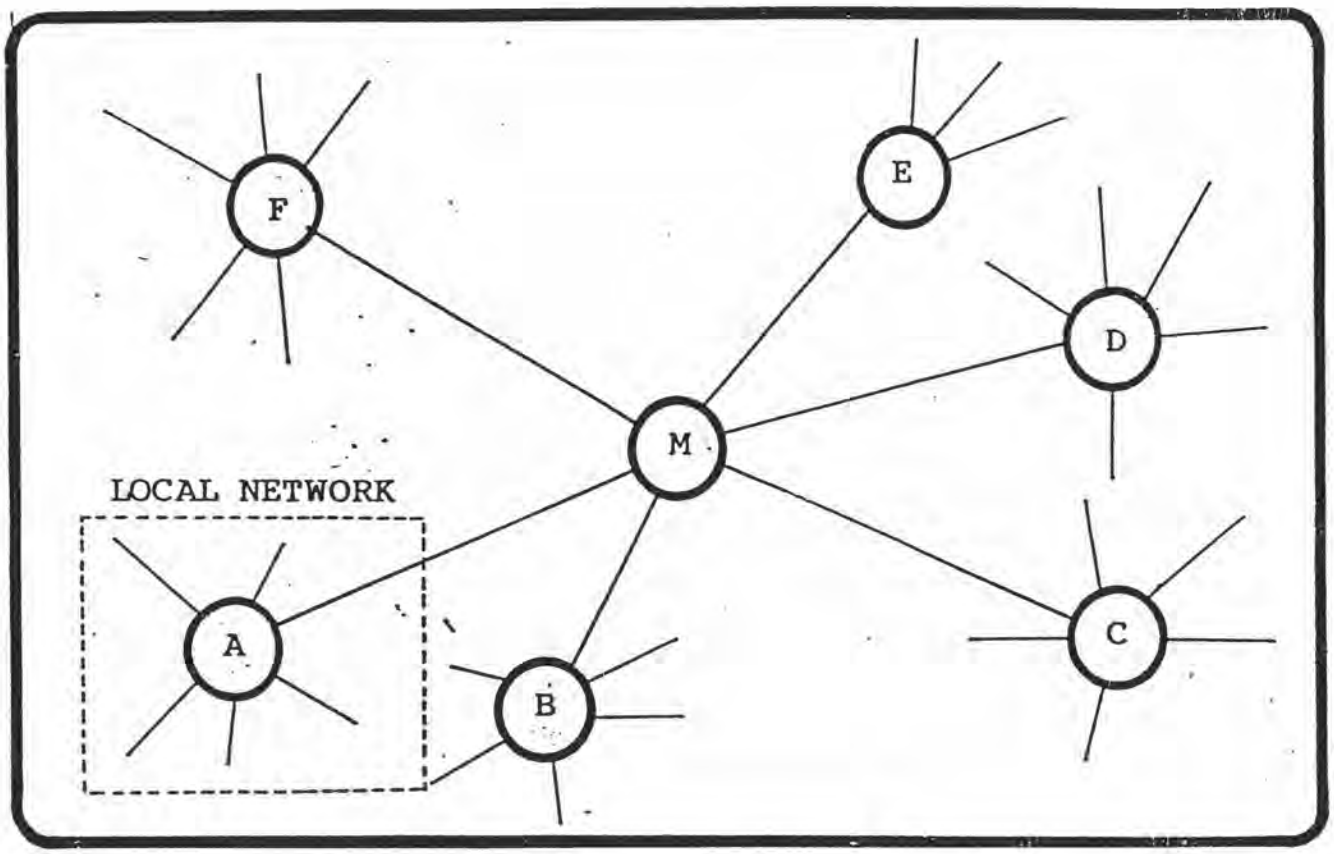
ส่วนของฐานข้อมูล F_i ไว้ที่สถานี S_j ต้นทุนในการสอบถามส่วนต่าง ๆ ของฐานข้อมูล F_i ซึ่งเก็บอยู่ที่สถานี S_j ต้นทุนในการแก้ไขส่วนต่าง ๆ ของฐานข้อมูล F_i ในทุก ๆ สถานีรวมทั้งต้นทุนในการสื่อสารข้อมูลระหว่างสถานีด้วย

2. ประสิทธิภาพสูงสุดนั่นคือ เวลาในการตอบสนองต้องน้อยที่สุด และผลลัพธ์ของระบบงาน (Throughput) ต้องมากที่สุดด้วย

5. ระบบเครือข่าย (Network) ระบบเครือข่ายแบ่งออกเป็นหลายประเภทตามลักษณะต่าง ๆ ดังนี้ (สุริยัน ศรีสวัสดิ์กุล, 2530)

5.1 การแบ่งตามโครงสร้างการเชื่อมต่อระหว่างสถานี (Topology) โดยทั่วไปแบ่งระบบเครือข่ายออกเป็น 4 รูปแบบ คือ

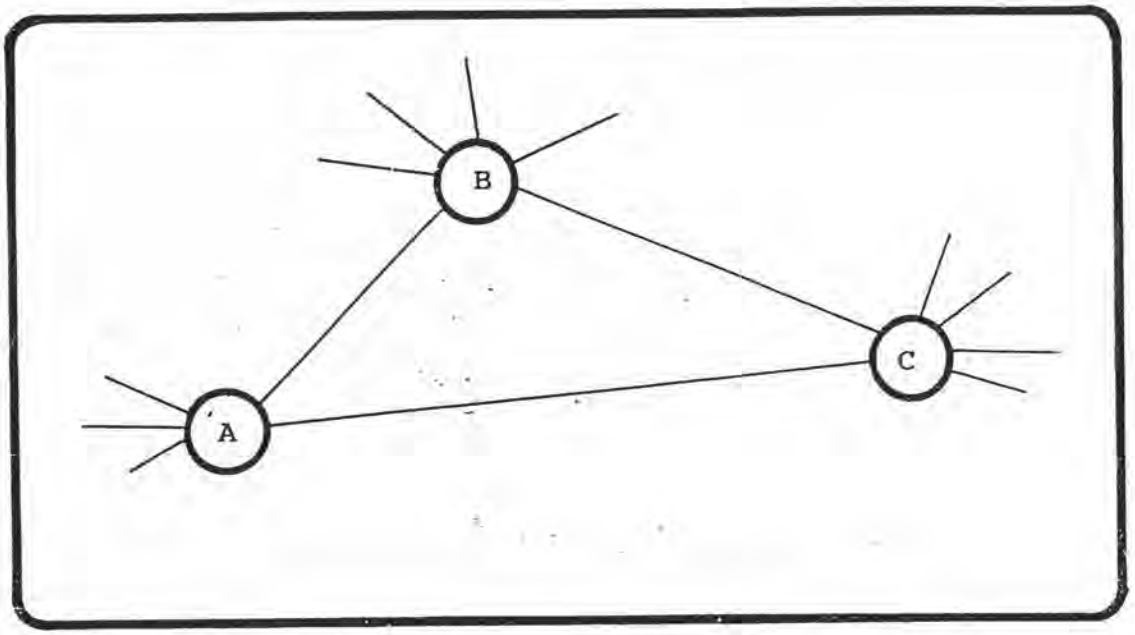
5.1.1 เครือข่ายรูปดาว (Star Topology) ซึ่งเป็นรูปแบบที่มีการใช้กันมากโดยเครือข่ายรูปแบบนี้อุปกรณ์รับส่งข้อมูลจะเชื่อมโยงกับศูนย์กลางแบบจุดต่อจุด (point-to-point) แต่ในรูปแบบการเชื่อมโยงแบบจุดต่อจุดนี้สายแต่ละเส้น อาจมีการต่อกับอุปกรณ์รับส่งข้อมูลอีกหลายเครื่องก็ได้ ซึ่งถือเป็น Multipoint Line เครือข่ายแบบดาวนี้ขึ้นตรงต่อความสมบูรณ์แบบของศูนย์กลางโดยตรง ในกรณีที่มีงานของระบบมาก ๆ เช่น การใช้ในกิจการของสายการบินอาจจะต้องมีอุปกรณ์สำรองบางส่วน หรือทั้งระบบเลยก็ได้เพื่อกันการเสียหายของการทำงานอันเกิดจากเครื่องเสียในกระบวนการประมวลผลข้อมูลด้วยคอมพิวเตอร์นั้น บางครั้งจะกระจายระบบคอมพิวเตอร์ไปตามจุดต่าง ๆ ในจุดที่เครื่องคอมพิวเตอร์เครื่องหนึ่งติดตั้งอยู่เครื่องนั้น ๆ ก็จะทำเครือข่ายสื่อสารภายในท้องถิ่นของงาน (Local Network) ซึ่งส่วนใหญ่ก็ใช้รูปแบบของเครือข่ายแบบดาวเช่นกัน จากนั้นคอมพิวเตอร์แต่ละเครื่องจะเชื่อมโยงกับเครื่องคอมพิวเตอร์ตัวหนึ่งซึ่งเลือกไว้เป็นศูนย์กลางในรูปแบบของเครือข่ายแบบดาวเช่นกัน ดังแสดงในรูปที่ 3.8 จะสังเกตุได้ว่าคอมพิวเตอร์ทุกเครื่องมีเครือข่ายแบบดาวเป็นของตนเองและการติดต่อกับคอมพิวเตอร์เครื่องอื่น ๆ ก็ถือเสมือนเป็นอุปกรณ์รับส่งข้อมูลทางไกล (Remote Terminal) สมมุติว่าเป็นอุปกรณ์รับส่งข้อมูลของอีกเมืองหนึ่งก็สามารถทำได้ โดยผ่านคอมพิวเตอร์ที่ทำหน้าที่เป็นศูนย์กลาง แต่ข้อเสียอยู่ที่ว่าถ้าเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นศูนย์กลางเกิดเสียหายขึ้นมาการติดต่อสื่อสารจะทำไม่ได้เลย



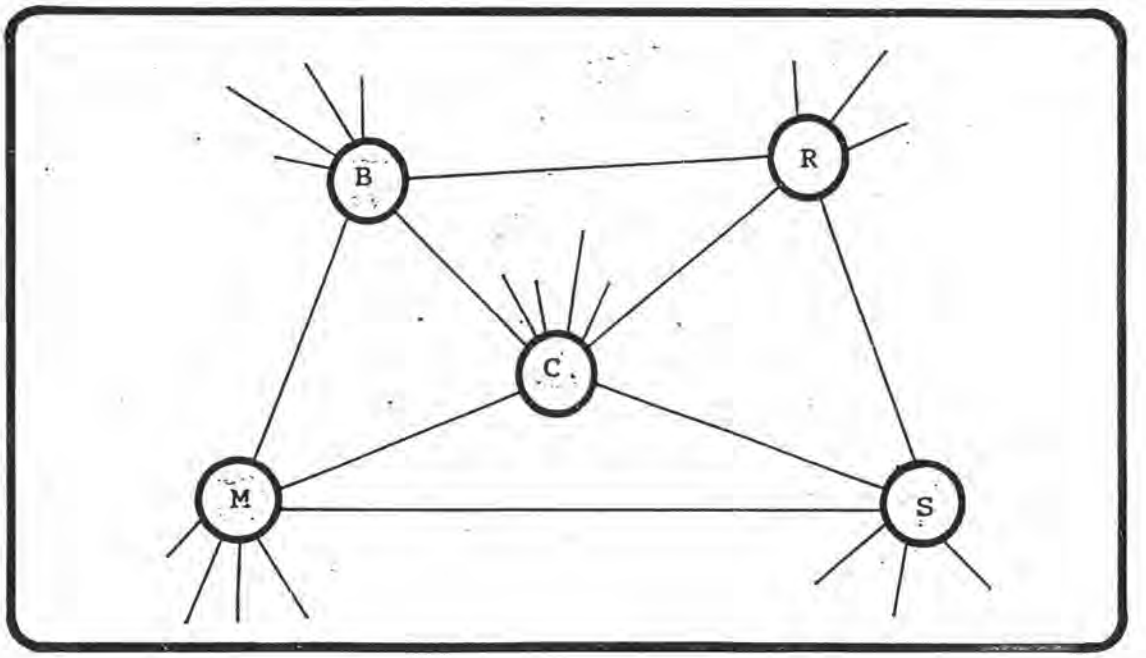
รูปที่ 3.8 แสดงการเชื่อมโยงคอมพิวเตอร์หลาย ๆ ระบบเข้าด้วยกันโดยใช้เครือข่ายแบบดาว และคอมพิวเตอร์แต่ละตัวยังทำหน้าที่เป็นศูนย์เครือข่ายประจำท้องถิ่นด้วย

5.1.2 เครือข่ายแบบวงแหวน (Ring Topology) จากข้อเสียของการจัดเครือข่ายแบบดาวที่กล่าวมาแล้วเราอาจปรับปรุงเครือข่ายเสียใหม่ โดยใช้รูปแบบวงแหวนดังแสดงในรูปที่ 3.9 ซึ่งจะเห็นได้ว่าที่เครื่องคอมพิวเตอร์แต่ละเครื่องควบคุมเครือข่ายแบบดาวของตนเองไว้แล้วแต่ละเครื่องจะเชื่อมโยงถึงกันในรูปวงแหวน ซึ่งช่วยให้เกิดช่องทางการสื่อสารระหว่างคอมพิวเตอร์เครื่องใด ๆ 2 เครื่อง สามารถติดต่อกันได้ 2 ทาง ในกรณีที่อีกช่องทางหนึ่งส่งผ่านไม่ได้ก็จะใช้อีกช่องทางหนึ่ง

5.1.3 เครือข่ายแบบตาข่าย (Meshed Topology) ในกรณีที่ต้องมีการส่งผ่านข้อมูลถึงกันในปริมาณที่สูง และเป็นการส่งผ่านทางไกลระหว่างเมืองต่าง ๆ เราอาจต้องใช้เครือข่ายแบบตาข่ายช่วย เพื่อให้คอมพิวเตอร์แต่ละเครื่องสามารถติดต่อถึงกันโดยตรงโดยไม่ต้องผ่านเครื่องอื่น หรือจำเป็นต้องผ่านก็จะลดจำนวนเครื่องที่ต้องผ่านไปดังรูปที่ 3.10 ถ้าเครื่องคอมพิวเตอร์ B ต้องการติดต่อกับเครื่องคอมพิวเตอร์ S สามารถเลือกใช้เส้นทางการสื่อสารได้มากขึ้นแม้ว่าจะต้องผ่านเครื่องคอมพิวเตอร์เครื่องอื่นก็ตาม การที่จะตัดสินใจเลือกรูปแบบของเครือข่ายแบบไหน จะต้องพิจารณาในด้านค่าใช้จ่ายในเรื่องของสายส่งสัญญาณลักษณะทางภูมิศาสตร์ที่เครือข่ายครอบคลุมไปถึง และปริมาณข้อมูลที่จะมีการส่งผ่านภายในเครือข่ายสื่อสาร ตัวอย่างเช่นในประเทศออสเตรเลียมีเมืองใหญ่ ๆ เพียงไม่กี่เมือง และแต่ละเมืองอยู่ห่างไกลกันนับพันกิโลเมตร ดังนั้นเครือข่ายสื่อสารที่ใช้กันมากในประเทศนี้จะเป็นในรูปแบบเครือข่ายแบบดาว ส่วนในประเทศอังกฤษและอเมริกามีเมืองขนาดใหญ่มากและปริมาณข้อมูลที่ต้องส่งผ่านสูง ดังนั้นเครือข่ายที่พบมากใน 2 ประเทศนี้ก็คือ เครือข่ายแบบวงแหวน และแบบตาข่าย



รูปที่ 3.9 เครือข่ายแบบวงแหวน

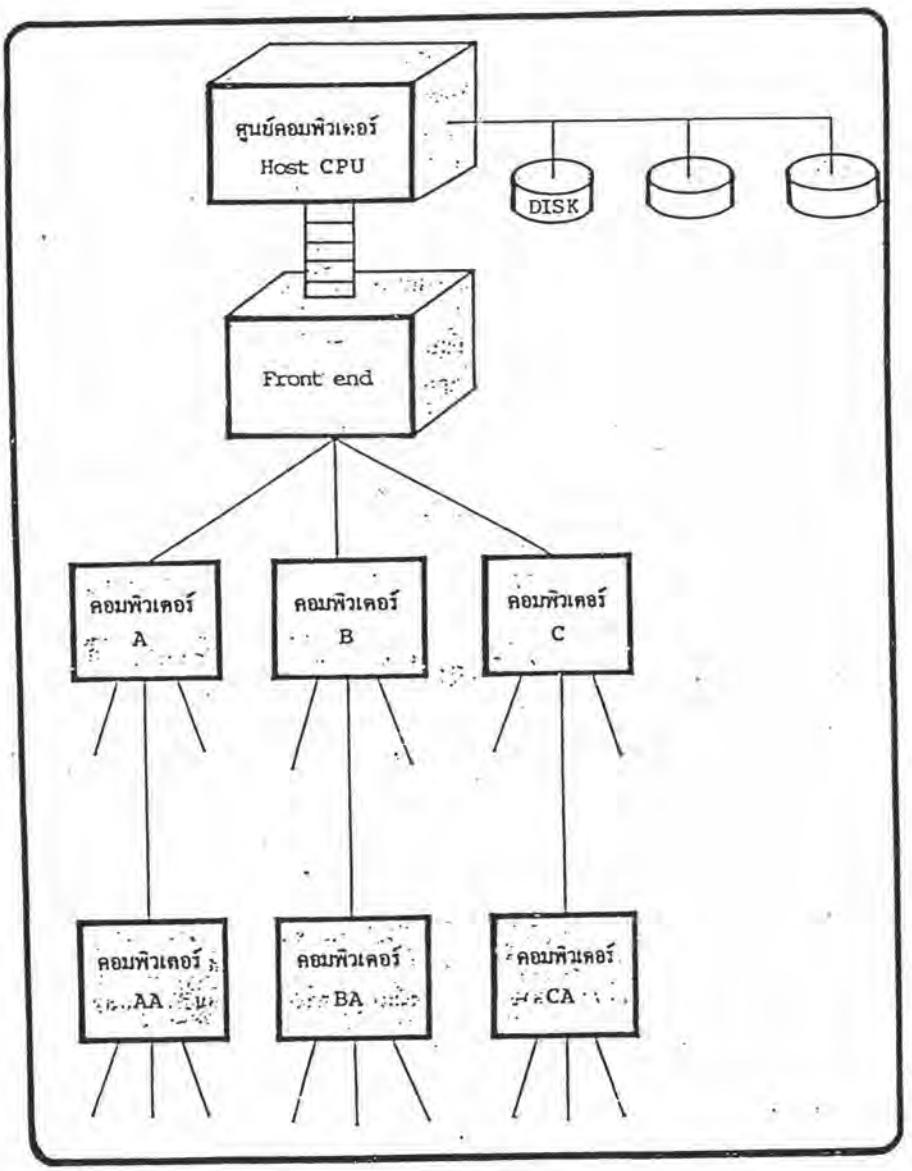


รูปที่ 3.10 เครือข่ายแบบตาข่าย

5.1.4 เครือข่ายแบบลำดับชั้น (Hierarchical Topology) เครือข่ายรูปแบบนี้จะมีลักษณะคล้ายรากต้นไม้ดังแสดงในรูปที่ 3.11 ในเครือข่ายแบบนี้จะประกอบไปด้วยเครื่องคอมพิวเตอร์ที่มีระดับต่าง ๆ กันอยู่หลายระดับ แล้วต่อเชื่อมโยงซึ่งกันและกันเป็นชั้น ๆ ดูแล้วคล้ายแผนผังโครงสร้างขององค์กรต่าง ๆ (Organization Chart) ถ้าพิจารณาจากแผนภาพต่าง ๆ ของเครือข่ายแบบดาว, แบบวงแหวน, และแบบตาข่ายเมื่อเครือข่ายเหล่านี้มีปริมาณมากขึ้นและต้องการความถูกต้องสูง เครือข่ายแบบนี้จะมีความสลับซับซ้อนมากยิ่งขึ้น ในเครือข่ายแบบดาว ถ้าเครื่องที่ทำหน้าที่เป็นศูนย์กลางเสีย การติดต่อซึ่งกันและกันก็จะทำไม่ได้แต่วิธีแก้ไขที่ใช้กันเป็นส่วนมากแล้วมักจะมีเครื่องสำรองอยู่ที่ศูนย์กลาง เมื่ออีกเครื่องเสียอีกเครื่องก็จะทำงานแทนทันที

ในเครือข่ายแบบวงแหวน ถ้ามีเครื่องใดในวงแหวนเสียไปการติดต่อจะทำไม่ได้เฉพาะอุปกรณ์รับส่งข้อมูลที่พ่วงติดอยู่กับเครื่องที่เสียเท่านั้น ส่วนอื่น ๆ จะยังคงสื่อสารกันได้เพราะยังมีเส้นทางอื่นไว้สำรอง ในรูปแบบเครือข่ายแบบตาข่ายก็เช่นกันจะเพิ่มความสะดวกในการสื่อสารมากขึ้นอีกเพราะมีเส้นทางสื่อสารให้เลือกใช้ได้มาก ในกรณีเส้นทางหนึ่งมีงานอยู่ก็จะไปเลือกใช้เส้นทางอื่นได้

ในรูปแบบดาว, วงแหวน, และตาข่าย จะมีความแตกต่างกันที่การใช้ขนาดความยาวของการสื่อสาร และความถูกต้องในการสื่อสารตามหลักของการออกแบบเครือข่ายสื่อสารจะพิจารณากันที่ความถูกต้อง ในการส่งผ่านข้อมูล, ประสิทธิภาพในการส่งผ่านข้อมูล, และค่าใช้จ่ายในการจัดตั้งเครือข่ายสื่อสาร ซึ่งอาจจะออกมาในรูปแบบเครือข่ายสื่อสารแบบผสม (Irregular) คือมีบางส่วนของข่ายสื่อสารใช้รูปแบบดาว, บางส่วนเป็นแบบวงแหวน หรือตาข่ายก็ได้ ซึ่งขึ้นอยู่กับการใช้งานที่เหมาะสมของเครือข่ายส่วนนั้น ๆ



รูปที่ 3.11 เครือข่ายสื่อสารแบบลำดับชั้น

5.2 การแบ่งตามรูปแบบการส่งข้อมูล วิธีนี้จะแบ่งระบบเครือข่ายดังนี้

5.2.1 เครือข่ายประเภทส่งข้อมูลแบบจุดต่อจุด (Point-to-Point Channel) โดยที่สถานีทุกคู่จะเชื่อมโยงถึงกันด้วยสายสื่อสาร (Link) และจะไม่ยอมให้ผู้อื่นใช้สายร่วมด้วย เช่น เครือข่ายรูปดาว เครือข่ายรูปวงแหวน และเครือข่ายแบบลำดับชั้น เป็นต้น

5.2.2 เครือข่ายประเภทส่งข้อมูลได้หลายจุด (Multi Point Network) ด้วยสาเหตุที่ค่าเช่าช่องทางการส่งผ่านข้อมูลต้องเสียค่าใช้จ่ายมาก ดังนั้นจึงทำให้ผู้ใช้ระบบการสื่อสารมีแนวความคิดที่ว่าทำอย่างไรจึงจะคุ้มค่ามากที่สุด และในที่สุดก็ออกมาในรูปที่ใช้ช่องทางการสื่อสารข้อมูลเพียงช่องทางเดียว แต่เชื่อมโยงอุปกรณ์รับส่งข้อมูลไว้หลาย ๆ จุดนั่นเอง เช่น เครือข่ายแบบบัส (Bus) และเครือข่ายที่ส่งข้อมูลโดยใช้ดาวเทียม เป็นต้น

5.3 แบ่งตามพื้นที่ทางภูมิศาสตร์ คือ ระบบเครือข่ายจะถูกแบ่งออกเป็น 2 ประเภทคือ เครือข่ายระยะใกล้ หรือเครือข่ายงานเฉพาะที่ (Local Area Network) และเครือข่ายงานบริเวณกว้าง (Wide Area Network) ข้อแตกต่างระหว่างเครือข่ายทั้งสองประเภทนี้คือ ระยะทางระหว่างเครื่องคอมพิวเตอร์ที่เชื่อมต่อเข้าด้วยกัน สำหรับเครือข่ายงานเฉพาะที่นั้นแต่ละเครื่องอาจตั้งอยู่ภายในอาคารเดียวกันหรือตั้งอยู่ภายในรัศมีหนึ่งกิโลเมตร ส่วนเครือข่ายงานบริเวณกว้าง เครื่องต่าง ๆ จะกระจายอยู่ตามสถานที่ต่าง ๆ เป็นระยะทางไกลได้ถึงสิบกิโลเมตร

<u>โมเดลข้อมูลเชิงตรรกะ (Logical Data Model)</u> (Fleming, 1989)

เป็นวิธีการแสดงความต้องการสารสนเทศในระบบธุรกิจให้เป็นแผนภาพ ในขั้นตอนการวิเคราะห์และออกแบบระบบฐานข้อมูล เน้นตัวข้อมูลที่มีอยู่จริงโดยไม่คำนึงถึงรายละเอียดในการติดตั้ง, ความต้องการพิเศษอื่นในแง่การใช้งาน และความเร็วในการสืบค้นข้อมูล หรืออาจกล่าวอีกนัยหนึ่งว่าโมเดลข้อมูลเชิงตรรกะเป็นการสร้างโครงร่างวิวของผู้ใช้ (Skeletal User View) จะแสดงข้อมูลในขอบเขตที่ผู้ออกแบบสนใจโดยมีสิ่งที่ต้องกำหนดเป็นพื้นฐานได้แก่ เอนทิตี, รีเลชันชิป, แอตตริบิว

วิวของผู้ใช้ คือ โมเดลหรือการแทนความต้องการในแง่ผู้ใช้แต่ละคนเทียบเท่ากับการปฏิบัติงานหนึ่งอย่าง เช่น การพิมพ์ใบส่งสินค้าให้ลูกค้ารายหนึ่ง และในที่สุดมีการรวมวิวของผู้ใช้เข้าด้วยกัน เรียกว่า โมเดลข้อมูลเชิงตรรกะแบบเบ็ดเสร็จ (Integrated Logical Data Model)

เอนทิตี คือ สิ่งที่มีอยู่จริง จับต้องได้ หรือเป็นจินตภาพที่แสดงความเป็นหนึ่งเดียว เมื่อกล่าวถึงแล้วทุกคนเข้าใจตรงกัน เช่น สินค้า, วัตถุดิบ, แผนกต่าง ๆ, การสั่งซื้อ

รีเลชันชิป คือ ความสัมพันธ์ซึ่งเป็นลักษณะการเกี่ยวพันกันระหว่างเอนทิตีหนึ่งกับตัวมันเองหรือเอนทิตีอื่น อาจเป็นความสัมพันธ์ที่มากกว่า 2 เอนทิตีก็ได้ เช่น แผนกจัดซื้อทำการสั่งซื้อสินค้าหรือวัตถุดิบ

แอตตริบิว คือ กลุ่มของค่าความจริงใด ๆ ที่เป็นรายละเอียดของเอนทิตีซึ่งแสดงลักษณะ และคุณสมบัติของเอนทิตี ทำให้เข้าใจเอนทิตีได้ลึกซึ้งยิ่งขึ้น และเป็นสิ่งที่ไม่สามารถแตกย่อยลงไปได้อีกโดยไม่เสียความหมายไป เช่น รหัสสินค้า, สถานที่เก็บ, ชื่อสินค้า, ราคา นอกจากนั้นยังมีการระบุด้วยว่าแอตตริบิวใดเป็นคีย์ กำหนดกฎข้อบังคับต่าง ๆ ของเอนทิตีและรีเลชันชิป

1. คุณลักษณะของโมเดลข้อมูลทางตรรกะ

1.1 แสดงได้ด้วยแผนภาพ (Graphical Diagrams) ไม่ว่าจะเป็นเทคนิคโมเดลข้อมูลแบบใดก็ตามจะมีภาษาและรูปภาพทางกราฟฟิกโดยเฉพาะ เพื่อใช้แสดงรายละเอียดข้อมูลทั้งกลุ่มใหญ่และรายละเอียดส่วนย่อย ซึ่งทำให้ง่ายต่อการแปลความ เช่น ใช้วงกลม หรือสี่เหลี่ยมแทนเอนตีตีใช้เส้นโค้งหรือเส้นตรงแทนรีเลชันชิป

1.2 แสดงชัดเจนถึงความหมายของข้อมูล (Explicit Representation of Semantic) มีทางเลือกในการแสดงความหมายของข้อมูล เราอาจใช้สัญญลักษณ์ที่ต่างกันจำนวนมากบ้างน้อยบ้างเพื่อแสดง แต่จุดสำคัญคือแผนภาพที่ได้ออกมาควรง่าย ไม่ซับซ้อน และเห็นความหมายของข้อมูลชัดเจน โดยเฉพาะอย่างยิ่งสัญญลักษณ์หนึ่ง ๆ ไม่ควรมีหลายความหมาย

1.3 แสดงรายละเอียดในระดับที่เหมาะสม (Appropriate Level of Detail) กล่าวคือ โมเดลระดับตรรกะจะมีรายละเอียดที่เพียงพอที่จะชี้จุดที่ทำให้เกิดความแตกต่างระหว่างชนิดของข้อมูลรีเลชันชิปและข้อบังคับต่าง ๆ แต่จะน้อยกว่าโมเดลทางกายภาพ

1.4 ไม่พึ่งพิงกับระบบจัดการฐานข้อมูลแบบใดแบบหนึ่ง(DBMS Independence) โมเดลที่ได้จากการออกแบบแล้วควรใช้ได้กับระบบฐานข้อมูลหลายแบบ ได้แก่ แบบความสัมพันธ์, แบบลำดับชั้น และแบบเครือข่าย

1.5 ง่ายต่อการศึกษาและใช้งาน ในที่นี้จะต้องง่ายเพียงพอสำหรับผู้ใช้ทุกประเภทจะทำความเข้าใจและนำไปใช้ได้

2. ขั้นตอนการออกแบบโมเดลข้อมูลเชิงตรรกะ

ในการออกแบบโมเดลข้อมูลเชิงตรรกะมีด้วยกันหลายขั้นตอนสำหรับใน 5 ขั้นตอนแรกจะเป็นการออกแบบทางด้านโครงสร้างพื้นฐานของโมเดล ได้แก่ พวกเอนทิตี, รีเลชันชิป, คีย์หลัก, คีย์สำรอง, คีย์ภายนอก, กฎเกณฑ์พื้นฐาน จากนั้นจึงเริ่มเพิ่มรายละเอียดในระดับที่ผู้ใช้มองเห็น (User View) และรวมรายละเอียดเหล่านั้นเข้าด้วยกัน จึงได้เป็นโมเดลข้อมูลเชิงตรรกะที่สมบูรณ์ ขั้นตอนต่าง ๆ สรุปได้ดังภาพที่ 3.12

ขั้นตอนที่ 1 การกำหนดเอนทิตีหลัก

โดยกำหนดชื่อและความหมายลงในพจนานุกรมข้อมูล และเขียนลงโมเดลข้อมูลด้วยการตั้งชื่อไม่ควรเกิน 20 ตัวอักษร

การกำหนดเอนทิตีนั้นเป็นงานที่ยาก และต้องอาศัยความร่วมมือของผู้ที่เข้าใจระบบที่เราออกแบบ เพื่อคัดเลือกสิ่งที่ถูกต้อง มีความสำคัญและเหมาะสมที่สุดมาเป็นเอนทิตี วิธีการอย่างคร่าว ๆ ก็คือให้พิจารณาข้อมูลทั้งหมดที่มี และจัดกลุ่มของข้อมูลโดยดูจากค่าและความหมาย ถ้าสามารถรวมกลุ่มกันได้ก็ให้รวมเข้าไว้ในเอนทิตีเดียวกัน นอกจากนั้นเรายังสามารถแยกเอนทิตีย่อยลงไปได้อีก โดยเรียกว่าเป็นซับไทป์ (Subtype) ของอีกเอนทิตีหนึ่ง เช่น ถ้าบอกว่า X เป็นซับไทป์ของ Y จะสังเกตุได้ดังนี้

1. เอนทิตี X และเอนทิตี Y ต้องแทนสิ่งเดียวกัน
2. X จะประกอบด้วยสิ่งต่าง ๆ ใน Y และส่วนที่เป็นของมันเองด้วย
3. ทุก ๆ X จะปรากฏเพียงหนึ่งใน Y (1:1)

ขั้นตอนที่ 2 การกำหนดรีเลชันชิประหว่างเอนทิตี

ส่วนประกอบที่สำคัญของโมเดลเชิงตรรกะ คือ เอนทิตีและรีเลชันชิปจะทำได้โดยกำหนดชื่อ ความหมาย ทิศทาง และขนาดอัตราส่วนที่เกิดรีเลชันชิปนั้น ๆ พร้อมทั้งบันทึกลงในพจนานุกรมข้อมูลด้วยสำหรับชื่อก็ไม่ควรเกิน 20 ตัวอักษรเช่นเดียวกับชื่อของเอนทิตี สำหรับความหมายถ้าเป็นภาษาอังกฤษให้ใช้คำกริยาในรูปแบบปัจจุบัน (Present Tense)

รูปที่ 3.12 ขั้นตอนการออกแบบโมเดลเชิงตรรกะ

การกำหนดรีเลชันชิประหว่างเอนทิตีนั้นต้องคำนึงถึงทิศทางของการกำหนดรีเลชันชิปนั้นด้วยว่าเป็นรีเลชันชิปจากเอนทิตีใดไปเอนทิตีใดโดยใช้ลูกศรแทนทิศทางของรีเลชันชิป และจะต้องมีเอนทิตีหนึ่งเป็นเอนทิตีหลัก เราเรียกเอนทิตีหลักนั้นว่า เอนทิตีระดับพ่อแม่ (Parent Entity) และเรียกเอนทิตีที่ตามมาว่าเอนทิตีระดับลูก (Child Entity) และหัวลูกศรจะชี้ไปทางเอนทิตีระดับลูกเสมอ โดยวิธีนี้เราสามารถแบ่งรีเลชันชิปออกเป็น 3 ประเภท โดยขึ้นอยู่กับอัตราส่วนของระเบียน (Record) ในเอนทิตีระดับพ่อแม่และเอนทิตีระดับลูก คือ

1. รีเลชันชิปแบบหนึ่งต่อหนึ่ง (one-to-one, 1:1 Relationship) เป็นรีเลชันชิปที่เมื่อพิจารณาระเบียนใดระเบียนหนึ่งในเอนทิตีหนึ่ง จะมีความสัมพันธ์กับระเบียนในอีกเอนทิตีหนึ่งเพียงระเบียนเดียวเท่านั้น

2. รีเลชันชิปแบบหนึ่งต่อหลาย (one-to-many, 1:N Relationship) รีเลชันชิปจะเกิดขึ้นได้ตั้งแต่ 0 จนถึงหลาย ๆ ครั้ง โดยแต่ละระเบียนของเอนทิตีลูกจะสัมพันธ์กับเอนทิตีแม่ได้หนึ่งระเบียนเท่านั้น แต่หนึ่งระเบียนในเอนทิตีแม่สัมพันธ์กับเอนทิตีลูกได้หลายระเบียน

3. รีเลชันชิปแบบหลายต่อหลาย (many-to-many, M:N Relationship) เป็นรีเลชันชิปที่เกิดขึ้นได้ตั้งแต่ 0 ครั้งจนถึงหลาย ๆ ครั้งในทั้งสองทิศทาง กล่าวคือระเบียนใดระเบียนหนึ่งในเอนทิตีลูกจะมีความสัมพันธ์กับระเบียนในเอนทิตีแม่ได้หลายระเบียนและในทางกลับกันระเบียนใดระเบียนหนึ่งในเอนทิตีแม่ก็สัมพันธ์กับระเบียนในเอนทิตีลูกได้หลายระเบียนเช่นกัน

หลังจากที่เราสามารถแบ่งกลุ่มรีเลชันชิประหว่างเอนทิตีได้เรียบร้อยแล้ว จะพบว่ารีเลชันชิปแบบหลายต่อหลายเป็นสิ่งที่เราต้องสนใจมากที่สุดเพราะเป็นตัวทำให้การสร้างฐานข้อมูลเชิงตรรกะมีความยุ่งยากซับซ้อนมากขึ้น โดยปกติฐานข้อมูลโดยทั่วไปจะไม่สามารถจัดการกับรีเลชันชิปแบบกลุ่มนี้ได้ ดังนั้นเราจึงควรกระจายรีเลชันชิปแบบหลายต่อหลายมาเป็นรีเลชันชิปแบบหนึ่งต่อหลายสองรีเลชันชิป โดยกำหนดเอนทิตีใหม่ขึ้นมาอีกเอนทิตีหนึ่งตัวให้มีรีเลชันชิปกับเอนทิตีเดิมแบบหนึ่งต่อหลาย

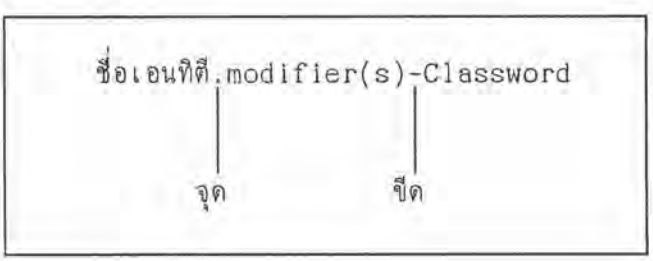
จากลักษณะของรีเลชันชิปแบบหนึ่งต่อหนึ่ง ถ้าเอนทิตีระดับลูกหลาย ๆ เอนทิตีเป็นส่วนประกอบของเอนทิตีระดับพ่อแม่เพียงเอนทิตีเดียว และเอนทิตีเหล่านั้นมีความหมายในเรื่องเดียวกันสามารถรวมรีเลชันชิปเหล่านั้นเป็นรีเลชันชิปเดียวได้ เรียกว่ารีเลชันชิประดับบนและระดับล่าง (Supertypes and Subtypes) ซึ่งเป็นรีเลชันชิปอีกแบบหนึ่งที่แสดงถึงรีเลชันชิประหว่างเอนทิตีระดับพ่อแม่กับชุดของเอนทิตีระดับลูกที่มีความหมายในเรื่องเดียวกัน

ขั้นตอนที่ 3 การกำหนดคีย์หลักและคีย์รอง

หลังจากที่ได้กำหนดเอนทิตีต่าง ๆ แล้ว ขั้นตอนต่อไปของการสร้างโมเดลข้อมูลทางตรรกะคือ การเพิ่มข้อมูลที่เรียกว่า แอตตริบิวลงในทุก ๆ เอนทิตี

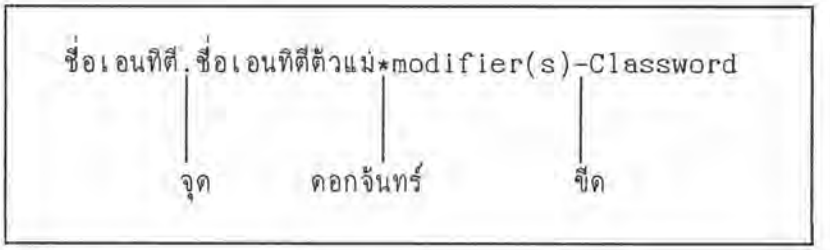
คีย์ คือ ชุดของแอตตริบิวที่เล็กที่สุดที่ใช้อ้างถึงระเบียนต่าง ๆ ในเอนทิตี แอตตริบิวที่แทนค่าคีย์จะมีค่าซ้ำกันไม่ได้ (Unique) ทุกเอนทิตีจะต้องมีคีย์หลักเสมอ คีย์หลักมีคุณสมบัติพิเศษต่างจากคีย์อื่น ๆ คือมีค่าเป็นนัล (Null Character) ไม่ได้ นอกจากนั้นในเอนทิตีหนึ่ง ๆ อาจมีคีย์รองเพื่อใช้ในการอ้างอิงถึงระเบียนต่าง ๆ ในเอนทิตีซึ่งชุดของแอตตริบิวที่แทนคีย์รองจะต้องเป็นคนละชุดกับแอตตริบิวที่แทนคีย์หลักเสมอ และในเอนทิตีหนึ่ง ๆ จะมีคีย์รองหรือไม่ก็ได้

สิ่งที่สำคัญอีกประการหนึ่ง คือ เอนทิตีที่เป็นซับไทป์จะต้องมีคีย์หลักอันเดียวกับเอนทิตีที่เป็นซุปเปอร์ไทป์ของมัน หลังจากกำหนดแล้วให้ตั้งชื่อระบุในโมเดลข้อมูลเชิงตรรกะพร้อมทั้งใส่ในพจนานุกรมข้อมูลด้วย การตั้งชื่อควรกำหนดสั้น ๆ ง่าย ๆ อาจใช้ชื่อย่อก็ได้ และควรหลีกเลี่ยงการตั้งชื่อแอตตริบิวของสองสิ่งที่ไม่เหมือนกันด้วยชื่อเดียวกัน รูปแบบการกำหนดอาจทำได้ดังนี้



ขั้นตอนที่ 4 การกำหนดคีย์ภายนอก

เมื่อกำหนดคีย์หลักและคีย์รองได้แล้วให้กำหนดคีย์ภายนอกสำหรับเอนทิตีที่มีรีเลชันชิปกันทุกอัน คีย์ภายนอก คือ แอตตริบิวในเอนทิตีระดับลูกที่แทนคีย์หลักของเอนทิตีระดับพ่อแม่ เพื่อใช้ในการอ้างถึงระเบียนในเอนทิตีระดับพ่อแม่และแสดงถึงรีเลชันชิประหว่างเอนทิตีต่าง ๆ ซึ่งอาจให้คีย์หลักเป็นคีย์ภายนอกด้วยก็ได้ มีกฎการตั้งชื่อแอตตริบิวที่เป็นคีย์ภายนอกดังนี้



ความสำคัญของคีย์ภายนอก คือ

1. ทำให้เกิดกฎธุรกิจ (Business Rules) ในเอนทิตีต่าง ๆ

2. ถึงแม้จะดูเหมือนว่าการกำหนดคีย์ภายนอกเป็นการทำให้ซ้ำซ้อน แต่จริง ๆ แล้วมันจะเป็นตัวช่วยในการออกแบบ เพื่อให้เราสามารถตรวจสอบได้ว่าเอนทิตีอันไหนเป็นเอนทิตีลูกและอันไหนเป็นเอนทิตีแม่

3. ทำให้ขั้นตอนการออกแบบฐานข้อมูลง่ายขึ้น

แต่ข้อควรระวังคือจะกำหนดให้คีย์รองเป็นคีย์ภายนอกด้วยนั้นไม่ควร เพราะจะทำให้ซับซ้อนและอาจมีค่าเป็นนัล (Null) ได้ ซึ่งคีย์รองไม่ควรเป็นเช่นนั้น

<u>ขั้นตอนที่ 5</u> การกำหนดกฎการจัดการกับข้อมูล

เป็นขั้นตอนที่ทำขึ้นเพื่อความสมบูรณ์ของข้อมูล และความถูกต้องตรงกันของค่าของข้อมูล กฎการจัดการกับข้อมูลแบ่งได้เป็น 3 ประเภทคือ

1. กฎธุรกิจของคีย์ (Key Business Rules) เป็นการกำหนดกฎเพื่อความสมบูรณ์ของรีเลชันชิป กฎต่าง ๆ จะมีผลจากการจัดการต่าง ๆ บนรีเลชันชิป อันได้แก่ การเพิ่ม, การลบและการแก้ไขข้อมูล เช่น กำหนดผลที่จะเกิดขึ้นจากการเปลี่ยนค่าของคีย์หลัก

2. โดเมน (Domain) คือการกำหนดกฎเพื่อความสมบูรณ์ถูกต้องของแอตตริบิว ก็คือการกำหนดข้อบังคับและค่าที่เป็นไปได้สำหรับแอตตริบิวทั้งที่เป็นคีย์และไม่ใช่คีย์

3. กฎการจัดการ (Triggering Operation) เป็นการกำหนดผลกระทบจากการเพิ่ม, ลบ, แก้ไขหรือดึงข้อมูลที่เกิดกับเอนทิตีอื่น หรือแอตตริบิวอื่นภายในเอนทิตีเดียวกัน

สำหรับในขั้นตอนนี้จะพิจารณาเฉพาะกฎธุรกิจของคีย์ก่อน ซึ่งการกำหนดกฎในการเพิ่มข้อมูลนั้นมักจะเป็นผลเมื่อเราทำการเพิ่มข้อมูลในเอนทิตีตัวลูก สำหรับกฎในการแก้ไขและลบข้อมูลมันจะเป็นผลเมื่อกระทำกับเอนทิตีตัวแม่

ขั้นตอนที่ 6 การเพิ่มแอตตริบิวที่ไม่ใช่คีย์ เพื่อเป็นรายละเอียดของเอนทิตี

ให้ทำการกำหนดแอตตริบิวอื่น ๆ ในเอนทิตี โดยตั้งชื่อและเพิ่มในโมเดลข้อมูลเชิงตรรกะพร้อมทั้งบันทึกลงในพจนานุกรมข้อมูลด้วย

จากขั้นตอนการทำโมเดลเชิงตรรกะที่ผ่านมาแอตตริบิวที่เรารู้จักก็คือคีย์หลัก, คีย์รอง, และคีย์ภายนอก ขั้นนี้ก็จะเป็นการกำหนดแอตตริบิวอื่น ๆ ที่ไม่ใช่คีย์เพิ่มเข้าไป โดยแอตตริบิวแต่ละตัวที่จะเพิ่มนั้นต้องขึ้นกับทั้งหมดของคีย์ของเอนทิตีนั้นไม่ใช่ขึ้นกับบางส่วนของคีย์ โดยต้องขยายในเอนทิตีตัวแม่ไม่ใช่ขยายในเอนทิตีที่มีคีย์นั้นเป็นคีย์ภายนอก นอกจากนั้นถ้าเกิดแอตตริบิวดังกล่าวขึ้นกับคีย์หลักทั้งหมดแล้วแต่มีค่ามากกว่าหนึ่งค่า (Multivalued) ให้แตกออกเป็นอีกเอนทิตีที่มีความสัมพันธ์กับเอนทิตีเดิมแบบหนึ่งต่อหลาย จากนั้นให้บันทึกชื่อและรายละเอียดอื่น ๆ ใส่ในโมเดลและพจนานุกรมข้อมูล

การพิจารณาว่าสิ่งใดเป็นแอตตริบิวจริงหรือไม่นั้น บางครั้งยุ่งยากแต่มีข้อสังเกตคือเราจะถือว่าเป็นแอตตริบิวเมื่อไม่มีรายละเอียดในปลีกย่อย สำหรับแอตตริบิวบางตัวเมื่อพิจารณาแล้วพบว่าจะขยายรีเลชันชิป ก็ให้แตกรีเลชันชิปนั้นออกมาตั้งเป็นเอนทิตีใหม่ที่มีรีเลชันชิปกับเอนทิตีเดิมเป็นแบบหนึ่งต่อหลาย สิ่งที่ควรหลีกเลี่ยงคือการใช้รหัสแทนแอตตริบิวแต่ถ้าจำเป็นก็ให้กำหนดรหัสอย่างเป็นอิสระจากกันคือ ไม่รวมความหมายมากกว่าหนึ่งอย่างไว้ในรหัสเดียวกัน ในกรณีที่แอตตริบิวใด ๆ สามารถหาได้จากสูตรหรือการคำนวณได้จากแอตตริบิวอื่น จะเรียกว่าเป็นดิไรฟ์แอตตริบิว (Derived Attribute) ให้ระบุตัวดี (d) ในโมเดลข้อมูลด้วย ในกรณีเป็นแฟล็กใส่ไว้เพื่อระบุซับไทป์-ซูเปอร์ไทป์ให้ระบุตัวเอส (s) ในโมเดลเช่นกัน

นอกจากการกำหนดแอตตริบิวแล้ว ยังให้พิจารณาเชื่อมเอนทิตีที่มีลักษณะดังต่อไปนี้เข้าด้วยกันคือ

1. เอนทิตีที่มีคีย์หลักเหมือนกันและให้ความหมายอันเดียวกัน
2. เอนทิตีที่เป็นซับไทป์ด้วยกันและมีแอตตริบิวทุกตัวเหมือนกัน
3. เอนทิตีที่เป็นซับไทป์-ซูเปอร์ไทป์ ซึ่งซับไทป์เป็นตัวขยายซูเปอร์ไทป์
4. เอนทิตีที่ไม่มีแอตตริบิวอื่นที่ไม่ใช่คีย์ให้ยุบรวมกับเอนทิตีที่มีรีเลชันชิปต่อกันเสีย

ขั้นตอนที่ 7 การตรวจสอบวิวของผู้ใช้ด้วยกฎของนอร์มัลไลเซชัน

ให้ทำการตรวจสอบเอนทิตีต่าง ๆ ให้อยู่ในกฎนอร์มัลไลเซชันซึ่งประกอบด้วย 1NF, 2NF, 3NF, BCNF, 4NF, 5NF ซึ่งจะได้กล่าวในหัวข้อต่อไป ประโยชน์ก็คือ

1. ลดที่ว่างที่ต้องใช้ในการเก็บข้อมูล
2. ลดความผิดพลาด ความไม่ตรงกันของข้อมูลในฐานข้อมูล

3. ลดการเกิดอะนอร์มัลไลของการลบและแก้ไขข้อมูล

4. เพิ่มความคงทนแก่โครงสร้างฐานข้อมูล

ในกรณีได้เอนทิตีที่เป็นนอมัลไลเซชันที่สมบูรณ์แล้ว สิ่งที่ต้องระวังคือไม่แตกเอนทิตีนั้นย่อยลงไปอีก

ขั้นตอนที่ 8 พิจารณาขอบเขตค่าโดเมนของแอตตริบิว

ให้กำหนดโดเมนของแอตตริบิวทุกตัวในเอนทิตีแล้วบันทึกในพจนานุกรมข้อมูล

โดเมน คือ กลุ่มค่าที่ถูกต้องเป็นไปได้สำหรับแอตตริบิวแต่ละตัว อันได้แก่

1. ชนิดของข้อมูล (Data Type) เช่น จำนวนเต็ม, วันที่, ตัวอักษร, ทศนิยม
2. ความยาว (Length) เช่น 5 หลัก, 35 ตัวอักษร
3. รูปแบบข้อมูล (Format) เช่น dd/mm/yy (วันที่)
4. ค่าที่อนุญาต (Allowable value) เช่น เป็นได้เฉพาะวันศุกร์ต้นเดือน
5. ช่วงของข้อมูลหรือข้อกำหนดอื่น ๆ (Range, Constraints)
6. ความหมาย (Meaning) อธิบายความหมายของแอตตริบิวนั้นว่าคืออะไร
7. ความเป็นหนึ่งเดียว (Uniqueness) ต้องมีค่าเป็นหนึ่งเดียว
8. ความเป็นนัล (Null support) อนุญาตให้เป็นนัลได้หรือไม่
9. ค่าโดยปริยาย (Default value) กำหนดให้มีค่าเป็น 0

ได้มีการกำหนดกฎไว้เป็นพิเศษสำหรับโดเมนของแอตตริบิวบางพวกต่อไปนี้

1. แอตตริบิวของคีย์หลัก จะต้องมีความเป็นหนึ่งเดียว และห้ามมีค่าเป็นนัล ในกรณีที่คีย์หลักเป็นคีย์ประกอบ แอตตริบิวแต่ละตัวที่มาประกอบคีย์หลักไม่จำเป็นต้องเป็นหนึ่งเดียว
2. แอตตริบิวของคีย์รอง มีกฎเหมือนคีย์หลัก เว้นแต่เป็นนัลได้
3. คีย์ภายนอก กลุ่มของโดเมนที่กำหนดให้จะต้องเหมือนกับกลุ่มโดเมนของคีย์หลักในเอนทิตีแม่
4. แอตตริบิวที่เป็นคีย์หลักของเอนทิตีซับไทป์ จะต้องมีค่าเป็นซับไทป์ของคีย์หลักของเอนทิตีซูเปอร์ไทป์

ขั้นตอนที่ 9 พิจารณากฎทริกเกอร์ดำเนินการ หรือกฎธุรกิจของแอตตริบิว

เป็นกฎที่มีเพื่อตรวจสอบความถูกต้องและควบคุมผลกระทบเมื่อเราทำการสืบค้น, เพิ่ม, แก้ไข, และลบ เราจะพิจารณาผลรวมทั้งที่เกิดกับเอนทิตีอื่นและแอตตริบิวอื่นภายในเอนทิตีที่

เรากระทำด้วย เมื่อกำหนดกฎการจัดการต่าง ๆ แล้วให้เก็บลงในพจนานุกรมข้อมูล ซึ่งประกอบไปด้วยส่วนต่าง ๆ ดังนี้

1. กฎของผู้ใช้ (User rule) เป็นข้อความอธิบายข้อกำหนดการดำเนินงานตามความต้องการของผู้ใช้
2. เหตุการณ์ของทริกเกอร์ดำเนินการ (Event) ได้แก่ เพิ่ม (INSERT), แก้ไข (UPDATE), ลบ (DELETE), หรือ สืบค้น (SELECT)
3. ส่วนประกอบของเหตุการณ์ (Object of event) ได้แก่ ชื่อของเอนทิตีหรือแอตตริบิวที่มีการเปลี่ยนแปลงหรือถูกสืบค้นในเหตุการณ์นั้น ๆ
4. เงื่อนไขภายใต้ทริกเกอร์ดำเนินการ (Condition)
5. การกระทำ (Action) เช่น ไม่ยอมรับเหตุการณ์ หรือทริกเกอร์ที่เกี่ยวกับเหตุการณ์นั้น

เรากำหนดกฎการจัดการเพื่อ

1. ความสมบูรณ์ ถูกต้องและเป็นอันหนึ่งอันเดียวกันของค่าของแอตตริบิว
2. กำหนดกฎการจัดการสำหรับทุกแอตตริบิว ที่เป็นตัวต้นของดิไรฟว์แอตตริบิว เช่น เมื่อมีการแก้ไขค่าของแอตตริบิวต้นกำเนิด ต้องมีทริกเกอร์ดำเนินการในการแก้ไขแอตตริบิวดิไรฟว์ด้วย
3. กำหนดกฎการจัดการสำหรับซับไทป์-ซูเปอร์ไทป์ โดยถ้าซับไทป์ถูกลบ ต้องลบซูเปอร์ไทป์ด้วย
4. กำหนดกฎการจัดการโดยกำหนดด้วยเวลา เช่น ถ้าระเบียนข้อมูลนี้มีอายุเกิน 1 ปีให้ทำการลบทิ้ง

ขั้นตอนต่าง ๆ ที่ผ่านมาเราได้ทำการพัฒนาจนได้มุมมองของผู้ใช้ที่มีรายละเอียดสมบูรณ์ ขั้นตอนต่อจากนี้จะเป็นการปรับแต่งมุมมองต่าง ๆ เข้าด้วยกัน เพื่อขจัดส่วนที่ซ้ำซ้อนและแก้ปัญหาความไม่ตรงกันของข้อมูล ซึ่งเป็นขั้นตอนที่ต้องใช้ทั้งประสบการณ์และทักษะในการวิเคราะห์มาก และนับว่าเป็นส่วนที่สำคัญที่สุด

<u>ขั้นตอนที่ 10</u> การเชื่อมมุมมองของผู้ใช้เข้าด้วยกันเป็นแบบเบ็ดเสร็จ

พิจารณาส่วนของวิวของผู้ใช้ที่คาบเกี่ยวกัน พยายามลดความเหลื่อมล้ำ ตัดความซ้ำซ้อน และแก้ปัญหาความไม่ถูกต้องตรงกันของข้อมูล โดยอาจมีการเพิ่มรีเลชันชิป หรือกฎทางธุรกิจใหม่ ๆ ขึ้นด้วย

สิ่งที่ควรพิจารณาในการรวมเอนทิตี

1. การรวมเอนทิตีที่มีคีย์หลักตัวเดียวกันและค่าที่เป็นไปได้ของคีย์หลักเหมือนกัน จะต้องได้เอนทิตีใหม่ที่มีแอตตริบิวรวมของสองเอนทิตีเดิม

2. ถ้าเอนทิตีสองอันมีคีย์หลักเดียวกัน แต่ค่าที่เป็นไปได้ของคีย์หลักนั้นเป็นซับเซตของกัน เราจะรวมได้ในรูปแบบของซับไทป์-ซูเปอร์ไทป์ โดยต้องตัดแอตตริบิวที่มีแล้วในเอนทิตีซูเปอร์ไทป์ออกจากเอนทิตีที่เป็นซับไทป์

3. ถ้าเอนทิตีสองอันมีคีย์หลักเดียวกัน แต่มีผลไปกำหนดแอตตริบิวที่ต่างกันบางตัวให้กำหนดซูเปอร์ไทป์ขึ้นมาอันหนึ่งสัมพันธ์กันสองเอนทิตีเดิม

4. การเชื่อมเอนทิตีสองตัวที่คีย์หลักของตัวหนึ่งเป็นคีย์รองของอีกตัว จะได้เอนทิตีใหม่ที่มีคีย์หลักตามเอนทิตีตัวหนึ่งส่วนคีย์หลักของเอนทิตีอีกตัวจะกลายเป็นคีย์รองไป และมีแอตตริบิวรวมระหว่างสองเอนทิตีเดิม แล้วตัดแอตตริบิวที่ซ้ำซ้อนออกเสีย และต้องกำหนดกฎธุรกิจให้ด้วยตามข้อบังคับเก่า เช่น ในกรณีคีย์หลักเดิมที่กลายมาเป็นคีย์สำรองในเอนทิตีใหม่ก็ต้องบังคงห้ามเป็นนัล

5. การรวมเอนทิตีใด ๆ ก็ตาม ต้องไม่มีผลไปเปลี่ยนแปลงเอนทิตีอื่นที่ไม่ได้เกี่ยวข้อง

สิ่งที่ควรพิจารณาในการรวมรีเลชันชิป

1. ให้รวมรีเลชันชิประหว่างเอนทิตีที่ให้ความหมายเหมือนกันเข้าด้วยกันโดยถ้าผลทำให้เกิดเป็นรีเลชันชิปแบบหลายต่อหลาย จะต้องทำการแตกให้เป็นรีเลชันชิปแบบหนึ่งต่อหลายสองอัน

2. การรวมรีเลชันชิปใด ๆ ต้องไม่ไปกระทบกับรีเลชันชิปอื่น ที่ไม่ต้องการการเปลี่ยนแปลง นอกจากจะพิจารณาแล้วว่าควรตัดออกเนื่องจากซ้ำซ้อน หรืออาจเพิ่มขึ้นใหม่เพื่อความเหมาะสม

3. จากการรวมเอนทิตีที่มีคีย์หลักเป็นคีย์รองของเอนทิตีอีกตัว ให้ตรวจสอบคีย์ภายนอกของเอนทิตีอื่น ๆ ที่อ้างมาถึงว่า ได้อ้างถึงคีย์หลักหรือคีย์รองของเอนทิตีใหม่ที่ได้จากการรวมนั้น ถ้าอ้างถึงคีย์รองต้องทำการเปลี่ยนให้เป็นคีย์หลัก

4. เมื่อรวมมุมมองแล้วให้กำหนดกฎธุรกิจของคีย์สำหรับรีเลชันชิปใหม่ด้วย

สิ่งที่ควรพิจารณาในการรวมแอตตริบิว

1. ให้ทำการรวมแอตตริบิวที่มีความหมายเหมือนกันภายในเอนทิตีเดียวกัน และรวมค่าที่เป็นไปได้ รวมถึงกฎการจัดการเข้าด้วยกันด้วย และให้พิจารณาค่าที่เป็นไปได้ของแอตตริบิวอื่นได้ด้วยว่าเปลี่ยนไปหรือไม่

2. เมื่อรวมเอนทิตีแล้วให้พิจารณาตัดแอตตริบิวที่เป็นดิไรฟว์หรือตัวบ่งชี้(Flag) ที่ไม่จำเป็นทิ้งเสีย

3. หลังจากได้รวม, ตัด, หรือเพิ่มรีเลชันชิปแล้ว ให้ทำการนอร์มัลไลซ์อีกครั้ง เพื่อตัดสิ่งที่ซ้ำซ้อนออกเสีย

ขั้นตอนที่ 11 การรวบรวมโมเดลข้อมูลที่มีอยู่เข้าด้วยกัน

จะเป็นขั้นที่เกี่ยวข้องกับสคีมาเชิงมโนภาพ โดยจะรวมโมเดลข้อมูลเชิงตรรกะที่ได้กับของที่มีอยู่แล้วเดิม ให้พัฒนาโมเดลใหม่ควบคู่ไปกับการพิจารณากฎเกณฑ์ข้อบังคับของเดิมโดยอาจมีการใช้เอนทิตี หรือรีเลชันชิปร่วมกับของเดิม และมีการกำหนดเอนทิตีขึ้นมาใหม่ด้วย

ขั้นตอนที่ 12 การวิเคราะห์ความมีเสถียรภาพและการเติบโตในอนาคต

การออกแบบโมเดลที่ทำผ่านมาจะพิจารณาข้อมูลที่เห็นได้ในปัจจุบันเป็นส่วนมาก สำหรับขั้นนี้ให้พิจารณาถึงสิ่งที่อาจเกิดขึ้น หรือเป็นไปได้ในอนาคตด้วย เช่น

1. อาจมีเอนทิตี หรือ รีเลชันชิปใหม่เกิดขึ้น ทำให้ต้องเพิ่มคีย์ภายนอกในเอนทิตีของเดิม
2. รีเลชันชิปแบบหนึ่งต่อหลาย อาจกลายเป็นรีเลชันชิปแบบหลายต่อหลายได้
3. คีย์หลักอาจเปลี่ยนไปเนื่องจากของเดิมไม่เป็นหนึ่งเดียวแล้ว

สิ่งเหล่านี้เมื่อเราพิจารณาแล้วอาจทำการดัดแปลงโมเดลไว้เพื่อรองรับ หรือจดบันทึกเก็บไว้ก่อนเฉย ๆ ก็ได้

หลักการนอร์มัลไลเซชัน (Normalization)

นอร์มัลไลเซชัน เป็นกระบวนการเพื่อพัฒนาการเชื่อมต่อของข้อมูลเพื่อตอบคำถามที่ว่าการออกแบบฐานข้อมูลทั้งทางตรรกะ และทางกายภาพที่ได้ออกมาใช้ได้หรือยัง ก่อนอื่นเราควรทำความเข้าใจคำศัพท์ต่าง ๆ ที่เกี่ยวข้อง ได้แก่

1. ความสัมพันธ์ (Relation) คือกลุ่มของทูเปิล (เทียบได้กับระเบียนข้อมูล) โดยมากจะแสดงในรูปแบบของตาราง แต่ละแถวของตารางก็คือ 1 ทูเปิลนั่นเอง ซึ่งแต่ละทูเปิลจะมีค่าต่าง ๆ บรรจุอยู่

2. คีย์ คือแอตตริบิวตั้งแต่ 1 ตัวขึ้นไป ซึ่งจะช่วยระบุ ค้นหาทูเปิลที่ต้องการในความสัมพันธ์ คีย์ตัวหนึ่งจะระบุถึงทูเปิลหนึ่งเท่านั้น คีย์ประกอบด้วย

2.1 คีย์หลัก (Primary Key) คือ แอตตริบิวซึ่งทุกแอตตริบิวในความสัมพันธ์ต้องขึ้นกับมัน (Dependence) หรือ กลุ่มของแอตตริบิวที่เป็นหนึ่งเดียว (Unique) และทุกทูเปิลที่จะเพิ่มเข้าในความสัมพันธ์อย่างน้อยแอตตริบิวนี้ต้องมีค่า

2.2 คีย์ประกอบ (Compound Key) คือ คีย์ที่ประกอบด้วยแอตตริบิวมากกว่า 1 ตัวขึ้นไป

2.3 คีย์รอง (Alternate Key) คือ แอตตริบิวที่มีคุณสมบัติเหมือนคีย์หลักทุกประการแต่ไม่ได้รับเลือกเป็นคีย์หลัก

2.4 คีย์เป็นได้ (Candidate Key) คือ แอตตริบิวที่มีคุณสมบัติที่สามารถเป็นคีย์หลักได้

พื้นฐานความคิดของการนอร์มัลไลเซชันจะพิจารณาการขึ้นต่อกันของแอตตริบิวดังนี้

1. การขึ้นต่อกันทั้งหมด (Functional Dependency) ถ้าในความสัมพันธ์ R มีแอตตริบิว A และ B จะกล่าวว่าแอตตริบิว B เป็นการขึ้นต่อกันทั้งหมดกับแอตตริบิว A ก็ต่อเมื่อแต่ละค่าของ A ในความสัมพันธ์ R จะสัมพันธ์กับแอตตริบิว B เพียงหนึ่งค่าเท่านั้นและเราจะเรียก A ว่าดีเทอร์มิแนน (Determinant) กล่าวคือทุกทูเปิลในความสัมพันธ์ R ที่มีค่าของ B ตรงกันจะต้องมีค่าของ A ตรงกันด้วย เขียนแทนด้วย A -----› B และจะกล่าวว่าแอตตริบิว A ขึ้นต่อกันทั้งหมดกับคีย์ประกอบ [K1,K2] หรือเขียนแทนด้วย [K1,K2 ==› A]

เมื่อ K1,K2 ----› A และ K1 --/--› A

และ K2 --/--› A

2. การไม่ขึ้นตรงกับคีย์หลัก (Transitively Dependency) ถ้าในความสัมพันธ์ R มีคีย์หลักคือ K และแอตตริบิว A และ B จะกล่าวว่าแอตตริบิว B ไม่ขึ้นตรงกับคีย์หลัก

เมื่อ K -------› A และ A ---/--› K

และ A ------› B

3. การขึ้นต่อกันหลายค่า (Multivalued Dependency) ในความสัมพันธ์ R มีแอตตริบิว A, B, C เราจะกล่าวว่า แอตตริบิว B เป็นการขึ้นต่อกันหลายค่าบน A ก็ต่อเมื่อเซทของค่าของแอตตริบิว B ในความสัมพันธ์ R ที่มีความสัมพันธ์ตรงกับคู่ของ ‹A,C› นั้น จะไม่ขึ้นกับค่าของ C เขียนแทนด้วย A -----›› B และจะมีลักษณะที่เห็นได้ดังนี้

3.1 แอตตริบิว A ค่าหนึ่ง จะเป็นตัวกำหนดกลุ่มของค่าแอตตริบิว B คือ เมื่อ 2 ทูเปิลในความสัมพันธ์ R มีค่า A เดียวกัน ไม่จำเป็นต้องมีค่า B เหมือนกัน แต่ค่าของ B จะต้องอยู่ในกลุ่มของค่า B ที่ถูกกำหนดโดย A

3.2 การเปลี่ยนแปลงค่าใดในแอตตริบิว C จะไม่มีผลกระทบต่อค่า B

3.3 สองทูเปิลในความสัมพันธ์ R ที่มีค่า B เหมือนกันไม่จำเป็นต้องมีค่า A เดียวกัน

3.4 ค่าของแอตตริบิว C สองค่าที่มีความสัมพันธ์กับค่า A เดียวกันจะต้องสัมพันธ์กับค่าของ B ในกลุ่มเดียวกันและเป็นกลุ่มที่ถูกกำหนดโดยค่า A นั้น ๆ

การนอร์มัลไลเซชันแบ่งออกได้เป็นหลายระดับ ได้แก่

นอร์มัลไลเซชันระดับที่ 1 (First normal form : 1NF) เป็นการขจัดแอตตริบิว หรือกลุ่มแอตตริบิวที่ซ้ำกันไปอยู่ในเอนทิตีลูก เพื่อแต่ละรายการในเอนทิตีไม่มีค่าของแอตตริบิวหรือค่าของกลุ่มแอตตริบิวที่ซ้ำกัน

นอร์มัลไลเซชันระดับที่ 2 (Second normal form : 2NF) เป็นการขจัดแอตตริบิวที่ไม่ขึ้นกับทั้งส่วนของคีย์หลักออกไป เพื่อให้แอตตริบิวอื่นทั้งหมดขึ้นตรงกับส่วนที่เป็นคีย์หลักทั้งหมดเท่านั้น

นอร์มัลไลเซชันระดับที่ 3 (Third normal form : 3NF) เป็นการขจัดแอตตริบิวที่ไม่เป็นคีย์ที่ขึ้นตรงกับแอตตริบิวอื่นที่ไม่ใช่คีย์หลักออกไป เพื่อให้แอตตริบิวที่ไม่ใช่คีย์หลักต้องขึ้นตรงกับทั้งส่วนที่เป็นคีย์หลัก และไม่ขึ้นกับแอตตริบิวอื่นที่ไม่ใช่คีย์หลัก

นอร์มัลไลเซชันระดับของ (Boyce/Codd normal form : BCNF)

นอร์มัลไลเซชันระดับที่ 4 (Fourth normal form : 4NF)

นอร์มัลไลเซชันระดับที่ 5 (Fifth normal form : 5NF)

<u>หมายเหตุ</u> ในการพัฒนาโมเดลข้อมูลเชิงตรรกะ สามารถเข้าสู่ขั้นตอนนอร์มัลไลเซชันในระดับที่ 3 ได้อยู่แล้ว ส่วนในระดับถัดไปต้องพิจารณาถึงลักษณะของข้อมูลว่าจำเป็นต้องดำเนินการเพิ่มเติมหรือไม่ ซึ่งมีหลักการซับซ้อนขึ้น

<u>การออกแบบฐานข้อมูลเชิงสัมพันธ์</u> (Relational Database Design: RDD)

เป็นขั้นตอนการแปลงจากโมเดลข้อมูลเชิงตรรกะ เข้าสู่ระบบฐานข้อมูลเชิงสัมพันธ์ (Transaction Process) อย่างมีเสถียรภาพภายใต้ระบบจัดการฐานข้อมูลที่ติดตั้งบนเครื่องคอมพิวเตอร์เป็นการปรับให้เข้ากับหน้าที่ใช้สอยเฉพาะและตามความต้องการในเรื่องประสิทธิภาพการทำงาน ประกอบด้วย 2 ส่วน คือ

1. โครงสร้างข้อมูล (Data Structure) บอกที่อยู่ของโครงสร้างข้อมูล ได้แก่ เอนทิตี, แอตตริบิว และความสัมพันธ์ ด้วยการระบุตารางความสัมพันธ์และสดมภ์ ดังนี้

1.1 ตารางความสัมพันธ์เทียบเท่ากับเอนทิตี และสดมภ์เทียบเท่ากับแอตตริบิว ในโมเดลข้อมูลเชิงตรรกะ โดยต้องคำนึงว่าแต่ละตารางและสดมภ์ที่พบในระบบจัดการฐานข้อมูลเชิงสัมพันธ์ต้องไม่มีความขัดแย้งกัน

สดมภ์ คือ หน่วยหนึ่งของข้อมูลที่สามารถกำหนดตำแหน่งที่อยู่ได้ อาจเรียกอีกอย่างว่าเขตข้อมูล

ตาราง คือ กลุ่มของสดมภ์ โดยสดมภ์ทุกตัวในตารางเป็นตัวพรรณาถึงลักษณะคุณสมบัติของตารางนั้น ๆ

1.2 บางกรณีอาจมีการรวมตารางเข้าเป็นตารางเดียวหรือแบ่งตารางออกเป็นหลายตารางเพื่อประโยชน์ในแง่การทำงานรวดเร็วขึ้น

2. ความบูรณภาพของข้อมูล (Data Integrity) จากแผนภาพโมเดลข้อมูลเชิงตรรกะสามารถควบคุมความบูรณภาพของข้อมูลด้วยกฎธุรกิจ ซึ่งได้แก่พารามิเตอร์ระบุตารางความสัมพันธ์ เช่น การสร้างเพื่อระบุความเป็นเอกลักษณ์ (Unique Index) เป็นต้น

ความบูรณภาพของข้อมูลเป็นการรักษาคุณภาพ ความน่าเชื่อถือของข้อมูลในฐานข้อมูลของระบบสารสนเทศ และป้องกันมิให้ข้อมูลเกิดความผิดพลาด ถ้าพิจารณาจากระบบจัดการฐานข้อมูลพบว่าปัจจุบันไม่ได้เอื้ออำนวยในการควบคุมทางด้านนี้โดยตรง กล่าวคือภาษาเอสคิวแอล (SQL Language) ซึ่งเป็นภาษาในยุคที่ 4 (4th Generation Language, 4GL) เป็นคำสั่งแบบไม่ใช่วิธีดำเนินการ (Nonprocedural Syntax) ใช้ในการสร้าง, เก็บ, เปลี่ยนแปลง, สืบค้น และดูแลรักษาสารสนเทศในฐานข้อมูล ไม่อาจระบุคำสั่งเพื่อให้เกิดบูรณภาพของข้อมูลได้โดยตรง จึงจำเป็นต้องกำหนดกฎเพื่อความบูรณภาพของข้อมูล แบ่งเป็น 3 ประเภทดังนี้

2.1 กฎความบูรณภาพของเอนทิตี (Entity Integrity Rule) กำหนดให้ส่วนประกอบของคีย์หลัก เป็นตัวระบุความเป็นเอกลักษณ์ภายในแต่ละทูเปิลของตารางความสัมพันธ์ ซึ่งจะต้องมีค่าเสมอ ในขั้นตอนการเพิ่ม,ปรับปรุง,หรือลบ จำเป็นต้องคงไว้ซึ่งคุณสมบัติของคีย์หลัก

2.2 กฎความบูรณภาพในการอ้างอิง (Referential Integrity Rule) เป็นการหาแหล่งที่มาของฟอร์เรนจ์คีย์ ถ้าตารางความสัมพันธ์ใดมีฟอร์เรนจ์คีย์ปรากฏอยู่ ทุกค่าของฟอร์เรนจ์คีย์อาจเป็นไปได้ 2 กรณี คือ

2.2.1 ค่านัล (Null Value)

2.2.2 มีค่าที่สอดคล้องกับค่าคีย์หลักในตารางความสัมพันธ์อื่น

ดังนั้นต้องมีการระบุถึงการอ้างอิงระหว่างตารางความสัมพันธ์ด้วยกัน เพื่อตรวจสอบค่าของฟอร์เรนจ์คีย์ว่าตรงกับค่าที่มีอยู่จริงในคีย์หลักหรือไม่ ในการดำเนินการเพิ่ม, ปรับปรุง หรือลบ อาจกล่าวอีกนัยหนึ่งว่าเป็นการควบคุมความบูรณภาพของฟอร์เรนจ์คีย์ก็ได้

2.3 กฎความบูรณภาพของโดเมน (Domain Integrity Rule) เป็นการควบคุมความถูกต้องสำหรับทุก ๆ สดมภ์ที่อยู่ในตารางความสัมพันธ์ ได้แก่

2.3.1 ชนิดของข้อมูล (Data Type) เช่น เป็นตัวอักษร,วันที่,ตัวเลข

2.3.2 ความยาวของข้อมูล (Length)

2.3.3 ช่วงของค่าของข้อมูล (Range of Data)

2.3.4 ค่าโดยปริยาย (Default Value)

2.3.5 ความเป็นเอกลักษณ์ (Uniqueness)

2.3.6 ต้องมีค่าหรือไม่ (Nullability)

ขั้นตอนการออกแบบฐานข้อมูลเชิงสัมพันธ์ มีดังนี้

RDD1 ระบุตารางความสัมพันธ์จากเอนทิตี

การกำหนดตาราง ในการสร้างฐานข้อมูลเชิงสัมพันธ์ จะแทนหนึ่งเอนทิตีจากโมเดลข้อมูลเชิงตรรกะด้วยหนึ่งตารางเท่านั้น

RDD2 ระบุสดมภ์ในตารางความสัมพันธ์ สำหรับแต่ละแอตตริบิวที่ขยายคุณสมบัติของเอนทิตีในฐานข้อมูลเชิงสัมพันธ์นี้ จะเห็นว่าแถวในตารางหนึ่งมีความหมายเหมือนกับระเบียนของเอนทิตีที่ถูกแปลงไปเป็นตารางนั้น ในการกำหนดสดมภ์ของตารางนั้นจะใช้แอตตริบิวในเอนทิตีนั้นโดยแทนหนึ่งแอตตริบิวด้วยหนึ่งสดมภ์

RDD3 ดัดแปลงโครงสร้างให้เหมาะสมกับสภาพแวดล้อม ของระบบจัดการฐานข้อมูลเชิงสัมพันธ์และปรับปรุงในเรื่องประสิทธิภาพการทำงานด้วย

RDD4 ออกแบบกฎธุรกิจที่เกี่ยวข้องกับเอนทิตี

1. บังคับคุณสมบัติทางตรรกะของคีย์หลัก ในการติดตั้งระบบแบบความสัมพันธ์ เช่น ความเป็นเอกลักษณ์ ไม่อนุญาตให้ค่าในคีย์หลักไม่มีค่า เป็นต้น
2. บังคับคุณสมบัติทางตรรกะของคีย์รอง ในการติดตั้งระบบแบบความสัมพันธ์ เช่น ความเป็นเอกลักษณ์ เป็นต้น

RDD5 ออกแบบกฎธุรกิจที่เกี่ยวข้องกับรีเลชันชิป ข้อจำกัดในการเพิ่ม, ลบ และแก้ไขแอตตริบิวที่เป็นคีย์ในลักษณะการอ้างอิงให้สอดคล้อง

RDD6 ออกแบบกฎธุรกิจเพิ่มเติมเกี่ยวกับแอตตริบิว ได้แก่ ทริกเกอร์ดำเนินการ, การควบคุมค่าโดเมน, ชนิดของข้อมูล, รูปแบบข้อมูล, ค่าที่ยอมรับ (ขอบขต, ความต่อเนื่องของข้อมูล), ความเป็นเอกลักษณ์, อนุญาตให้มีค่าได้หรือไม่ และค่าปริยาย

การปรับปรุงโมเดลข้อมูลใหม่เมื่อมีสารสนเทศใหม่เพิ่มเติม อาจทำให้เกิดการแบ่งแยก (Splitting) หรือการยุบรวม (Merging) ของเอนทิตีขึ้น โดยพิจารณาว่าแอตตริบิวเดิมควรจะถูกย้ายจากเอนทิตีเดิมไปสู่เอนทิตีใหม่หรือไม่ จะต้องคำนึงถึงสภาพความเป็นจริง เช่น

1. แอตตริบิวมีความสัมพันธ์กับแอตตริบิวอื่นอย่างไร
2. การสร้างเอนทิตีใหม่สามารถแก้ปัญหาการสืบค้นข้อมูลได้หรือไม่
3. ข้อเท็จจริงของแอตตริบิวอ้างอิงถึงแอตตริบิวอื่นหรือไม่
4. เอนทิตีใหม่สัมพันธ์กับเอนทิตีเดิมและกับเอนทิตีอื่น ๆ ในโมเดลอย่างไร

การจัดกลุ่มในระดับเอนทิตี (Entity Clustering)

ตั้งอยู่บนพื้นฐานของขอบเขตข้อมูล หมายความว่าข้อมูลทั้งหมดที่จำเป็นต้องการใช้ภายในหน่วยงาน โดยแต่ละหน่วยงานมีหน้าที่รับผิดชอบในการจัดกลุ่มเอนทิตี เนื่องจากเป็นเจ้าของข้อมูล โครงสร้างข้อมูลภายในองค์กรที่ปรากฏอยู่ จะเป็นตัวบ่งชี้ว่าเอนทิตีนี้ควรจะปรากฏในหน่วยงานใดบ้าง การจัดกลุ่มอาจเป็นกลุ่มย่อยของกลุ่มใหญ่รวมกันหลาย ๆ กลุ่มก็ได้ ขึ้นกับระดับความละเอียดที่ต้องการแสดง เช่น ในระดับบนโมเดลข้อมูลที่ได้จากการจัดกลุ่มมักแสดงเฉพาะเอนทิตีหลักที่มีส่วนเชื่อมโยงกับเอนทิตีอื่น เพื่อแสดงความสัมพันธ์ของข้อมูลทุกหน่วยงานภายในองค์กรโดยสังเขป ส่วนในระดับล่างลงมาอาจแสดงกลุ่มเอนทิตีที่สนใจเฉพาะหน่วยงานเท่านั้น ในกรณีที่หน่วยงานใดมีข้อมูลที่ซับซ้อนหรือมากมายก็อาจจัดกลุ่มย่อยในระดับล่างลงมาอีกได้ การจัดกลุ่มเอนทิตีอาจเกิดได้ในระหว่างการออกแบบฐานข้อมูลทั้ง 2 ขั้นตอน ได้แก่ขั้นตอนเชิงตรรกะ และ ขั้นตอนเชิงกายภาพ

เนื่องจากโมเดลข้อมูลประกอบด้วยเอนทิตีมากมาย การจัดกลุ่มของเอนทิตีในขั้นตอนเชิงตรรกะในแต่ละเอนทิตีมีการจัดกลุ่มของแอตตริบิว เพื่อขยายคุณสมบัติต่าง ๆ ขึ้นกับธรรมชาติของข้อมูลที่ปรากฏอยู่และโครงสร้างข้อมูล แต่ขั้นตอนเชิงกายภาพไม่เป็นเช่นนั้น เป็นการจัดกลุ่มเอนทิตีเพื่อตอบสนองวิวของผู้ใช้ โดยต้องคำนึงถึงการดำเนินการและความต้องการในการเข้าถึงฐานข้อมูล

การจัดกลุ่มในระดับเอนทิตีต้องพิจารณาถึง

1. ขอบเขตของข้อมูลที่เกี่ยวข้อง
2. โครงสร้างข้อมูลที่ปรากฏอยู่
3. เป็นการแสดงแทนวิวของผู้ใช้เฉพาะบริเวณ โดยปรากฏเฉพาะกลุ่มของเอนทิตีที่สนใจเท่านั้น
4. การสอบถามข้อมูลว่าครอบคลุมถึงเอนทิตีตัวใดบ้าง
5. การประมวลผลที่ต้องการจากผู้ใช้