บทที่ 4

ระบบฐานข้อมูล

ระบบฐานข้อมูล (Whittington, 1988) ประกอบด้วย ฐานข้อมูล (Database) และระบบจัดการฐานข้อมูล (Database Management System) (อ้างถึงในมณฑา พิเชฐสกุล, 2538)

ฐานข้อมูล (Database)

นิยามของคำว่า ฐานข้อมูล (Database) มีได้หลายรูปแบบ ขึ้นอยู่กับหลักเกณฑ์ที่ใช้ในการพิจารณา ซึ่งพอสรุปได้ดังนี้ (Whittington, 1988)

1. พิจารณาจากภาพรวม ฐานข้อมูลแบ่งเป็น

1.1 ฐานข้อมูลเชิงกายภาพ (Physically) คือ บิต (Bit) จำนวนมากที่บันทึกไว้บนสื่อที่เป็นหน่วยเก็บ (Storage Media)

1.2 ฐานข้อมูลเชิงความหมาย (Semantically) คือ ตัวแทนของการบรรยายถึงสิ่งทั้งมวล (Universe of Discourse)

2. พิจารณาจากบทบาทหน้าที่ของผู้ที่เกี่ยวข้องกับขบวนการ การพัฒนาระบบ

2.1 ผู้จัดการองค์กร

ฐานข้อมูล คือ ทรัพยากรที่มีคุณค่าและราคาแพงจึงต้องจัดการอย่างมีประสิทธิภาพ และใช้ประโยชน์ได้อย่างเต็มที่

2.2 นักเขียนโปรแกรม

ฐานข้อมูล คือ ที่ที่รวบรวมประเภทของระเบียน (Record Type) และ เส้นทางชี้นำ (Navigation Paths)

2.3 ผู้ใช้ระบบ

ฐานข้อมูล คือ แหล่งของสารสนเทศ (Source of Information)

3. พิจารณาจากด้านเครื่องจักร (Machine-Eye)

ฐานข้อมูลเป็นที่รวมของแฟ้มข้อมูล บางแฟ้มข้อมูลจะเก็บข้อมูลดำเนินการ (Operational Data) บางแฟ้มข้อมูลเก็บสารสนเทศระบบ (System Information) รวมถึงเค้าร่างของฐานข้อมูล (Database Schema) และดรรชนี เพื่อการเข้าถึงข้อมูลอย่างมีประสิทธิภาพ

ข้อมูลดำเนินการ (Engles. 1974 อ้างถึงใน Date. 1986) แตกต่างจากข้อมูลเข้า (Input Data) ข้อมูลออก (Output Data) หรือข้อมูลประเภทอื่น เช่น ลำดับงาน (Work Sequence) ผลลัพธ์ชั่วคราว หรือ สารสนเทศชั่วครู่ (Transient Information) แต่หมายถึง เอนทิตีพื้นฐาน (Basic Entities)

ข้อมูลนำเข้า หมายถึง ข้อมูลที่บันทึกเข้าระบบครั้งแรก ผ่านทางแผงแป้นอักขระ ตัวกราดตรวจ (Scanner) หรืออุปกรณ์นำเข้าอื่น ๆ และผ่านขั้นตอนการเปลี่ยนแปลงจึงได้ข้อมูลดำเนินการ

ข้อมูลออก หมายถึง ข่าวสาร (Message) และผลที่ได้จากการนำข้อมูลดำเนินการผ่านขบวนการต่าง ๆ และนำมาแสดงทางจอภาพหรือรายงาน

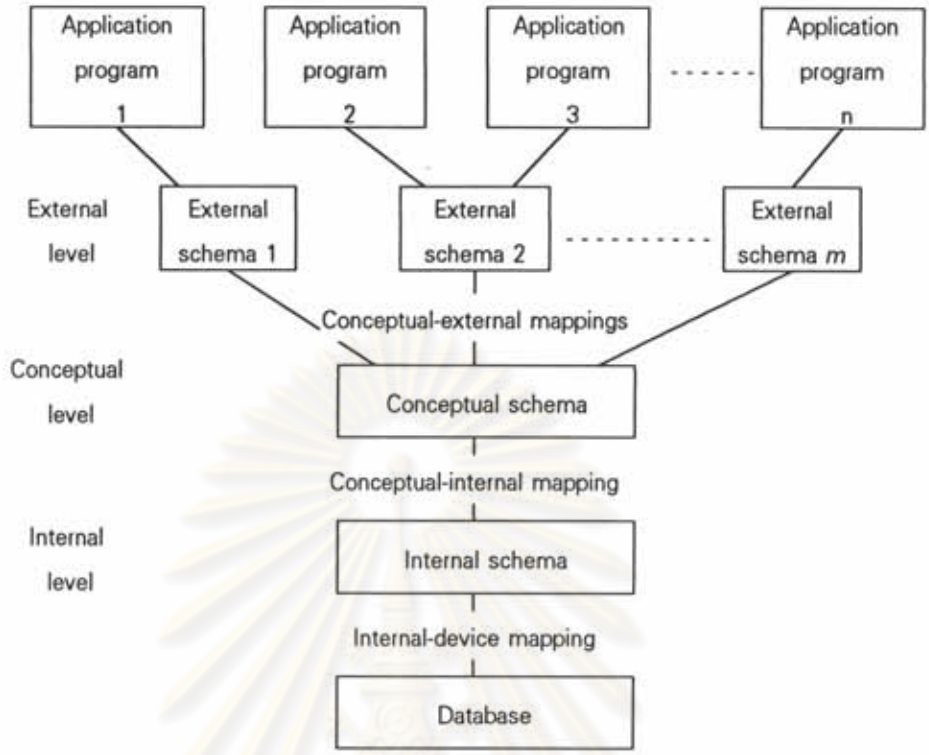
อย่างไรก็ตามนิยามของคำว่า ฐานข้อมูลที่นิยมใช้กันมากคือ คลังข้อมูลที่เบ็ดเสร็จ (Integrated) และอธิบายตัวมันเองได้ (Self-Describing)

เบ็ดเสร็จ หมายถึง ฐานข้อมูลที่มีความครบถ้วน (Wholeness) สามารถรักษาให้มีความสอดคล้องกัน (Consistent) เมื่อมีการอ้างถึงซึ่งกันและกัน (Cross-Reference) และนำมาใช้ประโยชน์ได้โดยการใช้โปรแกรมประยุกต์

<u>สถาปัตยกรรมฐานข้อมูล</u>

The ANSI/SPARC Study Group on Database Management Systems (ANSI, 1975) ได้นำเสนอหลักการของฐานข้อมูล ซึ่งได้ใช้เป็นแนวทางในการกำหนดกรอบงาน (Framework) ของระบบจัดการฐานข้อมูล ในขณะเดียวกันหลักการดังกล่าวก็สามารถใช้เป็นข้อมูลในการเปรียบเทียบ และประเมินคุณสมบัติของระบบจัดการฐานข้อมูลทั่ว ๆ ไปของแต่ละบริษัท หรือแต่ละยี่ห้อว่ามีความแตกต่าง หรือเหมือนกันอย่างไร

ในรายงานดังกล่าวได้บรรยายและเน้นถึงระบบจัดการฐานข้อมูลในลักษณะของความสัมพันธ์ระหว่างบทบาทของคน หน้าที่การประมวลผล (Processing Functions) และการไหลไปของข้อมูล (Information Flow) และเสนอลักษณะสำคัญของการย่อข้อมูล (Data Abstraction) เป็น 3 ระดับคือ คือ เค้าร่างภายใน (Internal Schema) เค้าร่างมโนภาพ (Conceptual Schema) และเค้าร่างภายนอก (External Schema) ดังแสดงในรูป



ระบบการจัดการฐานข้อมูลแบบ 3 ระดับ (Whittington, 1988)

1. เค้าร่างภายใน (Internal Schema) อธิบายการจัดการข้อมูลในระดับต่ำ แต่มิใช่การเก็บข้อมูลระดับกายภาพ เพราะมิได้อธิบายถึงการเก็บข้อมูลเป็นทรงกระบอก (Cylinder) หรือเป็นวง (Track) (Date, 1986) ลักษณะการเก็บข้อมูลกายภาพจะต้องพึ่งพิงโครงสร้างของหน่วยเก็บ เพราะหน่วยเก็บแต่ละชนิดจะมีโครงสร้างแตกต่างกันไป เค้าร่างภายในเป็นการอธิบายถึงข้อมูลในระดับที่อยู่เหนือจากกายภาพ และไม่พึ่งพิงโครงสร้างของหน่วยเก็บ โดยจะอธิบายถึงประเภทระเบียน (Record) เขตข้อมูล (Field) มีอะไรบ้าง เก็บอย่างไร ใช้อะไรเป็นดรรชนี มีการเรียงลำดับกายภาพ (Physical Sequence) อย่างไร ความยาวของระเบียน เค้าร่างภายในเป็นเพียงมุมมองภายในของลักษณะการเก็บข้อมูลในฐานข้อมูล และเพื่อเป็นการเชื่อมโยงระหว่างเค้าร่างภายในกับการเก็บข้อมูลกายภาพที่ขึ้นอยู่กับโครงสร้างของฮาร์ดแวร์ (Hardware) นั้นจะต้องใช้วิธีแปลงส่ง (Map) ภาษาที่ใช้อธิบายลักษณะข้อมูลในระดับนี้เรียกว่า Internal Data Definition Language หรือ Internal-DDL เป็นภาษาของอุปกรณ์เฉพาะ (Device Dependent Language)

โดยทั่วไปการเขียนโปรแกรมประยุกต์จะไม่ได้รับอนุญาตให้จัดการข้อมูลในระดับเค้าร่างภายในนี้ นอกจากโปรแกรมอรรถประโยชน์ เพราะโปรแกรมประยุกต์ที่เข้ามาจัดการเค้า

ร่างภายในนี้จะไม่ผ่านระบบรักษาความปลอดภัยทำให้เกิดความเสี่ยงต่อบูรณภาพของข้อมูล (Data Integrity) แต่บางครั้งเพื่อผลการกระทำ (Performance) ที่ดี หรือเพื่อให้ได้ฟังก์ชั่นบางอย่างตามต้องการ นักเขียนโปรแกรมจำเป็นต้องเขียนโปรแกรมจัดการถึงระดับเค้าร่างภายในนี้

2. เค้าร่างมโนภาพ (Conceptual Schema) หรือการมองภาพรวมของข้อมูลธุรกิจทั้งหมด (Fleming and Halle, 1989) ภาพที่เห็นในระดับนี้จะเป็นตัวแทนของเนื้อหาทั้งหมดของฐานข้อมูล ซึ่งเป็นการมองข้อมูลที่แท้จริงที่ควรจะเป็น โดยไม่สนใจว่าจะใช้ภาษาอะไรในการจัดการหรือใช้ฮาร์ดแวร์อะไรในการเก็บข้อมูล ผู้บริหารฐานข้อมูลที่เรียกว่า Database Administrator หรือ DBA ทำหน้าที่รวบรวม (Integrate) ความต้องการของผู้ใช้ในองค์กรทั้งหมด และกำหนดเนื้อหาของฐานข้อมูลควรจะเก็บข้อมูลอะไรบ้าง อย่างไร และสร้างเป็นเค้าร่างมโนภาพ โดยไม่สนใจโครงสร้างของหน่วยเก็บ หรือกลยุทธ์ในการเข้าถึงข้อมูล สิ่งที่จะกำหนดในเค้าร่างมโนภาพมีดังนี้

2.1 กำหนดลักษณะของข้อมูล (Data Definition) อธิบายประเภทข้อมูล ความยาวของเขตข้อมูลที่เล็กที่สุดในฐานข้อมูล ในฐานข้อมูลนั้นจะประกอบด้วยกลุ่มข้อมูล ข้อมูลแต่ละกลุ่มเรียกว่า รีเลชัน (Relations) หรือเอนทิตี (Entities) หรือเซทของเอนทิตี (Entity Set) และ 1 เอนทิตีประกอบด้วย ชิ้นข้อมูล (Data Items) เรียกว่า แอตตริบิว (Attributes) หรือเขตข้อมูล แต่โดยทั่วไปจะใช้คำว่า เอนทิตี (Entity) แทนคำว่า เซทของเอนทิตี เค้าร่างมโนภาพจะกำหนดว่าฐานข้อมูลประกอบด้วย เอนทิตีอะไรบ้าง (Entity Type) แต่ละเอนทิตีประกอบด้วยระเบียนที่มีความยาวเท่าใด มีแอตตริบิว (Attribute) อะไรบ้าง ประเภทและความยาวของแต่ละแอตตริบิว เป็นการอธิบายถึงระเบียนเชิงตรรก รวมถึงการกำหนดการตรวจสอบความปลอดภัย และบูรณภาพของข้อมูล จะใช้ข้อมูลอย่างไร การไหลของข้อมูล การตรวจสอบและควบคุมข้อมูลแต่ละจุด

2.2 ความสัมพันธ์ของข้อมูล (Data Relationship) กำหนดการเชื่อมโยงของข้อมูลระหว่างเอนทิตีที่มีความสัมพันธ์กัน

ข้อกำหนดในเค้าร่างมโนภาพนี้จะเป็นลักษณะของระเบียนเชิงตรรก (Logical Records) ไม่มีข้อมูลเกี่ยวข้องกับหน่วยเก็บ (Storage) การกำหนดลักษณะเค้าร่างมโนภาพจะใช้ภาษาที่เรียกว่า Conceptual Data Definition Language หรือ Conceptual-DDL ซึ่งจะไม่พึ่งพิงต่อโครงสร้างของหน่วยเก็บเช่นกัน เค้าร่างมโนภาพนี้นอกจากจะทำหน้าที่ควบคุมลักษณะข้อมูลทั้งหมดขององค์กรแล้ว ยังทำหน้าที่เป็นตัวเชื่อมทางอ้อมระหว่างเค้าร่างภายในและเค้าร่างภายนอก และการเชื่อมโยงระหว่างเค้าร่างมโนภาพและเค้าร่างภายในจะใช้วิธีการแปลงส่ง เรียกว่าการแปลงส่งเค้าร่างมโนภาพและเค้าร่างภายใน (Conceptual/Internal Mapping) เพื่อบอกให้ทราบว่า

จะเก็บระเบียนและแอตตริบิวอย่างไรในเค้าร่างภายใน แต่การแปลงส่งนี้จะกระทำโดยระบบซึ่งผู้ใช้ระบบจะมองไม่เห็น หรือไม่จำเป็นต้องเรียนรู้วิธีการ

3. เค้าร่างภายนอก (External Schema) เค้าร่างภายนอกนี้เป็นการมองข้อมูลของผู้ใช้แต่ละคนซึ่งอาจเป็นผู้ใช้ขั้นสุดท้าย (End User) ต้องการใช้ข้อมูลในลักษณะเชื่อมตรง (On-Line) หรือเป็นนักเขียนโปรแกรมจะใช้ข้อมูลในแต่ละโปรแกรมเพื่อประมวลผลตามต้องการ เค้าร่างภายนอกจะมองภาพข้อมูลจากฐานข้อมูลเป็นส่วน ๆ อธิบายมุมมองของผู้ใช้แต่ละคน ภาพของข้อมูลที่ผู้ใช้แต่ละคนเห็นนั้นอาจไม่เหมือนกัน ผู้ใช้หนึ่งคนอาจกำหนดเค้าร่างภายนอกหนึ่งชุดหรือหลายชุด ทำนองเดียวกันเค้าร่างภายนอกหนึ่งชุดมีผู้ใช้ได้หลายคน และเค้าร่างภายนอกแต่ละชุดอาจเหมือนหรือไม่เหมือนกับข้อมูลที่เก็บไว้จริงในฐานข้อมูล แต่จะมีการบำรุงรักษาให้มีความสอดคล้องกับเค้าร่างมโนภาพ กล่าวคือเค้าร่างภายนอกนี้จะเป็นการบอกลักษณะข้อมูลที่ใช้ในโปรแกรมประยุกต์แต่ละโปรแกรมตามความต้องการของผู้ใช้แต่ละคน โดยอธิบายให้ทราบว่าเค้าร่างภายนอกแต่ละชุดนั้นประกอบด้วยเอนทิตีอะไรบ้างจากฐานข้อมูล ไม่ใช่ทุกเอนทิตี แอตตริบิวอะไรบ้าง หรือทุกแอตตริบิวของระเบียน เค้าร่างแต่ละชุดจึงเป็นเพียงส่วนหนึ่งของเค้าร่างมโนภาพ บางครั้งจึงเรียกเค้าร่างภายนอกแต่ละชุดว่า เค้าร่างย่อย (Subschema) หรือเซทย่อย (Subset) ดังรูปที่ 3.2 ประกอบด้วยเค้าร่างภายนอก M ชุด แต่ละชุดเป็นส่วนหนึ่งของเค้าร่างมโนภาพ

การจัดการเค้าร่างภายนอกจะใช้ภาษาสองลักษณะคือ ภาษาที่ใช้อธิบายถึงฐานข้อมูลในระดับเค้าร่างภายนอกเรียกว่า External Data Definition Language หรือ External-DDL โดยทั่วไปจะเรียกว่า DDL และภาษาที่ใช้จัดการหรือประมวลผลข้อมูลเหล่านั้นเรียกว่า Data Manipulation Language หรือ DML และเพื่อให้เค้าร่างภายนอกเชื่อมโยงกับเค้าร่างมโนภาพจึงต้องมีการแปลงส่งซึ่งเรียกว่าการแปลงส่งเค้าร่างภายนอกและเค้าร่างมโนภาพ (External/Conceptual Mapping)

การทำงานของระบบฐานข้อมูลที่ใช้วิธีแปลงส่ง (Mapping) ระหว่างเค้าร่างมโนภาพและเค้าร่างภายใน และระหว่างเค้าร่างภายนอกและเค้าร่างมโนภาพ เมื่อมีการเปลี่ยนแปลงลักษณะการเก็บข้อมูลทางกายภาพ จะต้องมีการปรับปรุงการแปลงส่งเค้าร่างมโนภาพและเค้าร่างภายใน (Conceptual/Internal Mapping) แต่ไม่ต้องแก้ไขในส่วนของเค้าร่างมโนภาพ ซึ่งเรียกคุณสมบัตินี้ว่า ความไม่พึ่งพิงข้อมูล (Data Independence) หรือความไม่พึ่งพิงหน่วยเก็บข้อมูล (Data-Storage Independence) (Whittinton, 1988)

กรณีการประยุกต์ใช้ข้อมูลของผู้ใช้ระบบมีการเปลี่ยนแปลง เช่น มีการเพิ่มโปรแกรมประยุกต์ การแก้ไขนั้นเพียงเพิ่มเค้าร่างภายนอก และการแปลงส่งเค้าร่างภายนอกและเค้าร่างมโนภาพ หรือต้องการปรับปรุงโปรแกรมประยุกต์ที่มีอยู่ก็เพียงปรับเฉพาะการแปลงส่งเฉพาะส่วนที่ปรับปรุงเท่านั้น ไม่มีผลกระทบต่อเค้าร่างภายนอกอื่น และไม่จำเป็นต้องแก้ไขเค้าร่างมโนภาพหรือเค้าร่างภายใน คุณลักษณะเช่นนี้เรียกว่า ความไม่พึ่งพิงโปรแกรมประยุกต์ (Data-Application Independent) (Whittington, 1988)

ระบบฐานข้อมูลที่มีโครงสร้างตามแนวทางที่ ANSI/SPARC กำหนดไว้ และนิยมใช้เป็นแบบจำลองข้อมูลในการพัฒนาระบบจัดการฐานข้อมูลคือฐานข้อมูลแบบสัมพันธ์ (Relational Database) ฐานข้อมูลชนิดนี้จะมองข้อมูลในลักษณะของตาราง (Table) 2 มิติ แต่ละตารางจะแยกจากกัน ไม่เป็นลำดับชั้น หรือโครงสร้างที่ซับซ้อน การค้นคืนข้อมูลจึงสามารถเข้าถึงแต่ละตารางได้โดยตรงโดยไม่ต้องกำหนดเส้นทาง (Path) ไว้ล่วงหน้า และสามารถเชื่อมโยงตาราง 2 ตาราง ด้วยเขตข้อมูลที่มีความสัมพันธ์กัน ในระบบฐานข้อมูลแบบสัมพันธ์มีส่วนประกอบต่าง ๆ และคำศัพท์ (Terminology) ที่ใช้เรียกกันดังนี้ (Tsai, 1988)

1. รีเลชัน (Relation) หมายถึง ตาราง ซึ่งเป็นที่ที่เก็บข้อมูลที่มีความสัมพันธ์กัน ตารางประกอบด้วยรายการ (Occurrences) และแต่ละรายการจะประกอบด้วยเขตข้อมูล (Field)

2. แอตตริบิว (Attribute) แต่ละสดมภ์ของรีเลชันจะเรียกว่า แอตตริบิว หรือเขตข้อมูล (Field) นั่นเอง

3. โดเมน (Domain) คือ เซทของข้อมูลที่แต่ละแอตตริบิวสามารถมีค่าได้ เช่น โดเมนของแอตตริบิว ฤดู (Season) คือ ฤดูใบไม้ร่วง ฤดูร้อน ฤดูใบไม้ผลิ ฤดูฝน เป็นต้น

4. ทัพเพิล (Tuple) คือแถว (Row) ของตาราง (Table) แต่ละทัพเพิลจะแทนข้อมูล 1 รายการ (Record Occurrence) และ 1 รีเลชัน จะประกอบด้วยเซทของทัพเพิล

5. กุญแจหลัก หรือคีย์หลัก (Primary Key) และแคนดิเดทคีย์ (Candidate Key) เขตข้อมูลที่สามารถกำหนดข้อมูลของทัพเพิลได้ชัดเจน หรือยูนีค (Uniquely Identify) เรียกว่า แคนดิเดทคีย์ (Candidate Key) แต่ละรีเลชันมีได้หลายแคนดิเดทคีย์ แต่หนึ่งในหลายแคนดิเดทคีย์จะถูกกำหนดให้ทำหน้าที่เป็นคีย์หลักซึ่งอาจประกอบด้วยหนึ่งหรือหลายแอตตริบิว

6. กุญแจภายนอก หรือฟอเรนคีย์ (Foreign Key) เขตข้อมูลที่มีค่าตรงกับค่าของคีย์หลักในอีกรีเลชัน

7. แฟ้มระนาบ (Flat File) คือแฟ้มข้อมูลที่แต่ละแถวจะประกอบด้วยแอตตริบิวที่แต่ละแอตตริบิวจะมีค่าเพียง 1 ค่า การสร้างรีเลชันของฐานข้อมูลควรจะมีลักษณะของแฟ้มระนาบ

ดังตัวอย่างรูปที่ 3.3 แฟ้ม PROJECT เป็นแฟ้มระนาบ แต่ละ PROJECT ID จะมีแอตตริบิว PROJECT MANAGER ได้เพียง 1 ค่า และตัวอย่างในรูปที่ 3.4 แฟ้ม EMPLOYEE ไม่เป็นแฟ้มระนาบ เพราะแอตตริบิว CHILDREN มีค่ามากกว่า 1 ค่า

PROJECT File

PROJECT ID	PROJECT NAME	PROJECT MANAGER
P001	LENDING	E.LEE
P002	FUNDING	H.COHEN
P003	SECURITY	G.TURNER

รูปที่ 3.3 ตัวอย่างตารางที่มีลักษณะเป็นแฟ้มระนาบ

EMPLOYEE File

EMP-NO	EMP-NAME	CHILDREN
E105	E.LEE	EDWARD, CINDY
E106	H.COHEN	DAN
E107	G.TURNER	MARY, BOB, GREG
E108	R.CHARON	ALBERT

ตัวอย่างตารางที่ไม่ใช่แฟ้มระนาบ เพราะ CHILDREN มีค่าได้มากกว่า 1 ค่า

ฐานข้อมูลแบบสัมพันธ์ซึ่งมีโครงสร้าง 3 ระดับดังที่กล่าวมาแล้ว จึงมีข้อดีดังนี้

1. เป็นโครงสร้างที่เข้าใจง่าย ยืดหยุ่น สามารถปรับปรุงได้ง่าย
2. สามารถเข้าถึงข้อมูลแต่ละตารางได้โดยตรง ไม่ยุ่งยาก
3. มีลักษณะของความไม่พึ่งพิงข้อมูล และความไม่พึ่งพิงโปรแกรมประยุกต์
4. สามารถทำงานแบบข้อคำถามที่มิได้เตรียมล่วงหน้า (Ad-Hoc Queries) ได้อย่างมีประสิทธิภาพมากกว่างานผลิต (Production)

ระบบจัดการฐานข้อมูล (Database Management System - DBMS)

เพื่อให้การทำงานระบบฐานข้อมูลเป็นไปตามสถาปัตยกรรมดังที่กล่าวมาแล้วข้างต้น จะต้องมีตัวจัดการฐานข้อมูลที่สอดคล้องกับโครงสร้างนั้น ระบบฐานข้อมูลจึงมีส่วนประกอบที่สำคัญอีกส่วนคือ ระบบจัดการฐานข้อมูล (DBMS)

ระบบจัดการฐานข้อมูลประกอบด้วยซอฟต์แวร์รูทีน (Software Routine) จำนวนมากที่มีความสัมพันธ์กัน โดยแต่ละรูทีนจะทำงานเฉพาะอย่าง (Tsai. 1988) แต่หน้าที่สำคัญของระบบจัดการฐานข้อมูล (Whittington. 1988) คือ

1. การกำหนดรายละเอียดข้อมูล (Data Definition) การสร้าง และการปรับปรุงโครงสร้างของฐานข้อมูล ตลอดจนทำหน้าที่บรรจุข้อมูลดำเนินการเข้าสู่ฐานข้อมูล

2. การจัดการข้อมูล (Data Manipulation) ทำการปรับปรุง (Update) หรือค้นคืน (Retrieval) ข้อมูลจากฐานข้อมูลทันทีพร้อม ๆ กันหลายทาง และตอบสนองตามความต้องการของผู้ใช้ โดยมิให้เกิดความขัดแย้ง

3. การควบคุมข้อมูล (Data Control) กำหนดและควบคุมการใช้ข้อมูลให้เป็นไปตามเอกสิทธิ (Privileges) เพื่อรักษาความปลอดภัย (Security) และบูรณภาพของข้อมูล (Data Integrity)และป้องกันมิให้เกิดความเสียหายต่อข้อมูล รวมถึงการสำรองข้อมูล และสามารถกู้กลับ (Recovery) เมื่อระบบเกิดปัญหาด้วย

บูรณภาพของข้อมูล (Data Integrity) (Fleming and Halle, 1989) หมายถึง กฎธุรกิจ (Business Rules) ที่เป็นข้อกำหนดเกี่ยวกับค่าของข้อมูล หรือความสัมพันธ์ของข้อมูลควรเป็นอย่างไร บูรณภาพของข้อมูลสามารถแบ่งเป็น 3 อย่างคือ

3.1 บูรณภาพเอนทิตี (Entity Integrity) กฎนี้กล่าวไว้ว่า สดมภ์หรือแอตตริบิวที่เป็นส่วนหนึ่งของคีย์หลัก (Primary Key - คีย์หลักอาจประกอบด้วยหนึ่งหรือหลายแอตตริบิว) จะมีค่าว่าง (Null Value) หรือค่าที่ไม่รู้จักไม่ได้ ทั้งนี้เพราะคีย์หลักนี้จะเป็นตัวชี้เฉพาะข้อมูลแต่ละแถวหรือระเบียน จึงไม่ควรมีค่าว่างหรือค่าที่ไม่รู้จัก

3.2 บูรณภาพอ้างอิง (Referential Integrity) ข้อกำหนดเกี่ยวกับฟอเรนคีย์ (Foreign Key)

ฟอเรนคีย์ หมายถึง แอตตริบิวหรือกลุ่มของแอตตริบิวของตารางหนึ่ง มีค่าตรงกับคีย์หลักของอีกตารางหนึ่ง เนื่องจากบูรณภาพอ้างอิงเป็นการเชื่อมโยงกันระหว่าง 2 ตาราง เพื่อใช้อ้างอิงข้อมูลที่มีความสัมพันธ์กัน ดังนั้นบูรณภาพอ้างอิงจึงกล่าวไว้ว่า ตารางใดที่มีฟอเรนคีย์ ค่าทุกค่าของฟอเรนคีย์ไม่ว่าจะเป็นค่าว่างหรือไม่จะต้องมีค่าตรงกับคีย์หลักของอีก

ตารางหนึ่ง การเพิ่ม การปรับปรุง หรือลบข้อมูลของทั้ง 2 ตารางย่อมมีผลกระทบต่อกัน แต่จะมีผลกระทบอย่างไรขึ้นอยู่กับข้อมูลทางธุรกิจของแต่ละฐานข้อมูล

3.3 บูรณภาพโดเมน (Domain Integrity) โดเมน คือ ค่าที่เป็นไปได้ของแต่ละแอตตริบิว ไม่ว่าแอตตริบิวนั้นจะเป็นคีย์หลัก ฟอเรนคีย์ หรือแอตตริบิวอื่นที่ไม่ใช่คีย์ในตาราง ผู้ใช้สามารถกำหนดค่าที่เป็นไปได้ของแต่ละแอตตริบิวไว้ เพื่อใช้ตรวจสอบข้อมูลที่จะบรรจุในแต่ละแอตตริบิว

4. รวบรวมสถิติการใช้งานของระบบ เพื่อให้สามารถควบคุมการทำงานของระบบ และปรับปรุงผลการดำเนินการ (Performance) ให้ดีขึ้นตามที่คาดหวังไว้

5. ใช้ภาษารุ่นที่ 4 (Fourth-Generation Languages) เช่น แบบฟอร์ม (Forms) โปรแกรมก่อกำเนิดข้อมูลนำเข้า (Input-Program Generators) โปรแกรมก่อกำเนิดรายงาน (Report-Program Generators) เป็นต้น เพื่อเพิ่มความสามารถในการประยุกต์ใช้งาน

6. มีพจนานุกรมข้อมูล (Data Dictionary) เพื่อใช้ช่วยพัฒนาระบบฐานข้อมูล

7. สามารถเชื่อมโยงกับระบบประมวลผลคำ (Word Processing) แผ่นตารางทำการ (Spreadsheet) และระบบงานอื่น ๆ

หน้าที่ของระบบจัดการฐานข้อมูล 3 ประการหลังนี้ เป็นส่วนที่ระบบในปัจจุบันได้พยายามจัดเตรียมให้มีขึ้น เพื่อให้ระบบสามารถเชื่อมโยงกับระบบงานอื่น ๆ และสามารถทำงานได้สมบูรณ์มากยิ่งขึ้น

<u>ขั้นตอนการออกแบบฐานข้อมูล</u> (Phase of Database Design) (Mannila and Raiha, 1992)

1. วิเคราะห์ความต้องการของผู้ใช้ระบบ (Requirement Analysis)

ผู้เชี่ยวชาญการออกแบบฐานข้อมูลจะต้องรวบรวมความต้องการของผู้ใช้ระบบจากหน่วยงานต่างๆ และวิเคราะห์ความต้องการและปัญหาของระบบและของผู้ใช้ในปัจจุบัน รวมทั้งความต้องการขยายงานในอนาคต เพื่อให้แน่ใจว่าฐานข้อมูลเก็บข้อมูลที่จำเป็นในการประยุกต์ใช้งานอย่างครบถ้วน ผลที่ได้จากขั้นตอนนี้คือ ทราบการกระทำ (Actions) ในระบบ และการปฏิบัติการ (Operations) ที่ใช้ในฐานข้อมูลนี้

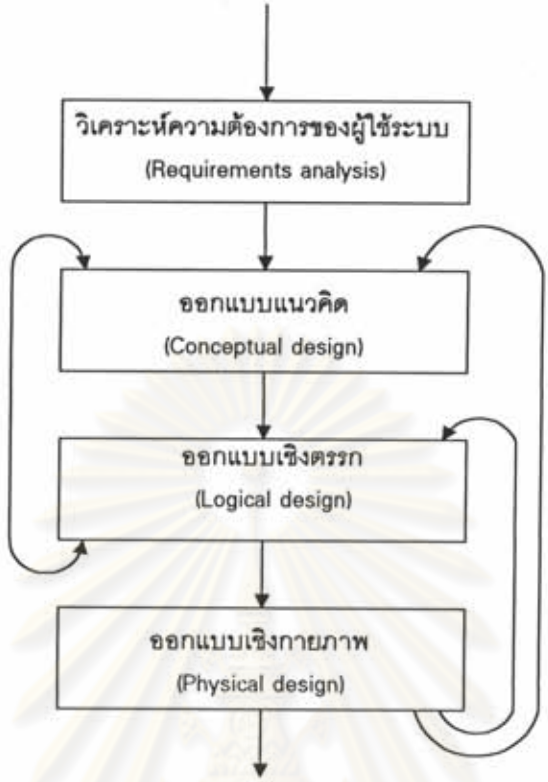
2. ออกแบบแนวคิด (Conceptual Design) ผลของขั้นตอนนี้เป็นการอธิบายโครงสร้างของข้อมูลว่าควรมีข้อมูลอะไรบ้างในฐานข้อมูลซึ่งเป็นโครงสร้างในระดับบน (High-Level Term) เรียกว่า เค้าร่างฐานข้อมูล (Database Schema)

ในขั้นตอนนี้ต้องศึกษาแนวคิดต่างๆ รวมทั้งแนวคิดของผู้ใช้ระบบแต่ละท่าน เพื่อหาแนวคิดที่จำเป็นซึ่งได้จากคำอธิบายขั้นตอนการปฏิบัติงานที่ได้จากการวิเคราะห์ความต้องการในขั้นตอนแรก เพราะเป้าหมายที่จำเป็นของการออกแบบฐานข้อมูลคือ การเก็บข้อเท็จจริงที่ไม่ซ้ำซ้อน ในขั้นตอนนี้อาจใช้แบบจำลองข้อมูล (Data Model) เพื่อใช้อธิบายแนวคิดต่าง ๆ และการทำทำวิจัยครั้งนี้ก็เลือกใช้แบบจำลองข้อมูลเชิงตรรก (Logical Data Model) มาจำลองแนวคิด เพราะเป็นแบบจำลองที่เข้าใจง่าย ไม่อิงกับโปรแกรมประยุกต์ หรือเทคโนโลยีใด ๆ สามารถขยายขอบเขตงานโดยมีผลกระทบต่องานที่มีอยู่แล้วน้อยมากหรืออาจไม่มีเลย ใช้แสดงโครงสร้างและการใช้ข้อมูลได้สอดคล้องกับแนวทางทางธุรกิจ

3. ออกแบบเชิงตรรก (Logical Design) เพื่อให้ได้ข้อมูลที่ไม่ซ้ำซ้อน (Redundancy) และผลจากการที่เก็บข้อมูลไม่ซ้ำซ้อนจึงเป็นการลดความผิดพลาดในการปรับปรุงข้อมูล ในขั้นตอนนี้จะใช้กฎนอร์มัลไลเซชัน (Normalization Rules) และดีนอร์มัลไลเซชัน (Denormalization) ผลที่ได้จากขั้นตอนนี้คือ คำอธิบายโครงสร้างที่จะใช้เก็บข้อมูลในฐานข้อมูล รวมทั้งการกำหนดประเภท ความยาว และค่า (Values) ของข้อมูล

4. ออกแบบเชิงกายภาพ (Physical Design) สร้างตารางให้มีโครงสร้างตามที่ได้กำหนดไว้ในขั้นตอนที่แล้ว กล่าวคือขั้นตอนนี้เป็นการเปลี่ยน (Transform) แบบจำลองข้อมูลที่ได้จากขั้นตอนที่ 3 ให้มาอยู่ในรูปแบบทางกายภาพ ในขั้นตอนนี้จะต้องทราบถึงรูปแบบต่างๆ ในมุมมองของผู้ใช้ระบบหรือคิวรี (Query) ความถี่ในการเรียกใช้งาน หรือความถี่ในการประมวลผล เพื่อใช้ปรับโครงสร้างให้เหมาะสมกับการใช้งาน เพราะผลการทำงาน (Performance) ขึ้นอยู่กับโครงสร้างของตารางหรือแฟ้มข้อมูล ปริมาณข้อมูล และความถี่ของการใช้งาน ดังนั้นบางระบบอาจต้องใช้เวลาและต้องทำการประเมินผลอย่างสม่ำเสมอ เพราะปริมาณข้อมูลน้อยอาจมองไม่เห็นผลการทำงานที่ถูกต้อง เมื่อมีการเปลี่ยนแปลงจำนวนข้อมูลมากขึ้น จึงจะเห็นผลการทำงานที่ถูกต้อง ซึ่งอาจทำให้ต้องปรับโครงสร้างฐานข้อมูลใหม่

ในการทำงานแต่ละขั้นตอนควรมีการตรวจสอบผลป้อนกลับ (Feedback) เพื่อให้แน่ใจว่าผลที่ได้ในแต่ละขั้นตอนถูกต้องตามที่ผู้ออกแบบต้องการ ดังรูป



ขั้นตอนการออกแบบฐานข้อมูล (Mannila and Raiha, 1992)

<u>แบบจำลองข้อมูลเชิงตรรก</u> (Logical Data Model) (Flaming and Halle, 1989)

การจำลองข้อมูลเชิงตรรกเป็นเทคนิคที่ใช้แทนโครงสร้างสารสนเทศเชิงธุรกิจ และกฎเกณฑ์ต่างๆ ที่จะใช้ในกระบวนการออกแบบฐานข้อมูลได้อย่างชัดเจน และยังสามารถใช้เป็นเอกสารที่บันทึกข้อเท็จจริงของข้อมูลที่มีอยู่ในระบบ โดยไม่สนใจว่าใครจะเป็นผู้ใช้ และจะใช้อย่างไร หรือเป็นระบบที่ใช้คอมพิวเตอร์หรือไม่ ทั้งนี้เพราะระเบียบวิธี (Methodology) นี้มีคุณลักษณะดังนี้

1. ใช้ข้อมูลขับ (Data-Driven) หลักการคือ สนใจข้อมูลที่ต้องการใช้ และวิธีจัดการ (Manipulate) กับข้อมูลในระบบ (Mannila, Raiha, 1992)

2. ทำให้เข้าใจความต้องการสารสนเทศเชิงธุรกิจ (Business Information Requirements)

3. ใช้เป็นสื่อในการติดต่อระหว่างผู้ออกแบบ ผู้พัฒนา และผู้ใช้ระบบ เพื่อให้ทุกฝ่ายมีความเข้าใจตรงกัน และแน่ใจว่าได้รวบรวมข้อมูลที่ครบถ้วนถูกต้องตามต้องการ ทั้งนี้เพราะ

เทคนิคนี้ไม่สนใจว่าจะทำอย่างไร และเนื่องจากการใช้แผนภาพ (Diagram) และคำอธิบายประกอบแทนการใช้คำพรรณนา จึงเข้าใจง่ายแม้จะไม่มีความรู้ทางด้านคอมพิวเตอร์

4. เทคนิคนี้เป็นพื้นฐานการออกแบบฐานข้อมูลให้มีความถูกต้อง (Correct) มีความสอดคล้อง (Consistent) ใช้ร่วมกัน (Sharable) และยืดหยุ่น (Flexible) และสามารถใช้กับเทคโนโลยีฐานข้อมูลทุกชนิด

การจำลองข้อมูลเชิงตรรกนี้ได้ใช้แนวคิดมาจากการจัดข้อมูลโดยระบบจัดการฐานข้อมูลที่ใช้สถาปัตยกรรม 3 ระดับของ ANSI/X3/SPARC ที่ได้กล่าวไว้แล้วข้างต้น

<u>การสร้างแบบจำลองข้อมูลเชิงตรรก</u> (Fleming and Halle, 1989)

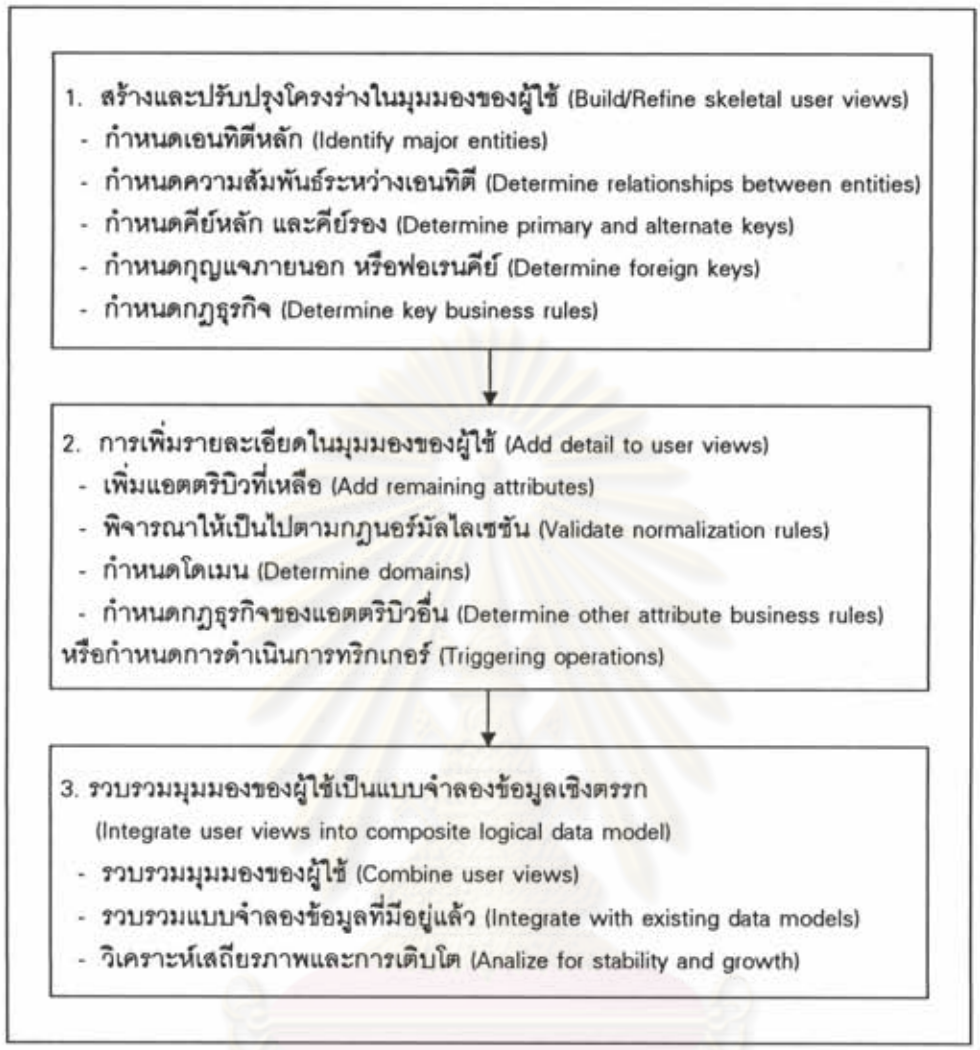
การสร้างแบบจำลองข้อมูลเชิงตรรกมี 3 ขั้นตอนหลักๆ ดังรูปที่ 3.7 ดังนี้

1. สร้างและปรับปรุงโครงร่างในมุมมองของผู้ใช้ (Build/Refine Skeletal User Views)

2. การเพิ่มรายละเอียดในมุมมองของผู้ใช้ (Add Detail To User Views)

3. รวบรวมมุมมองของผู้ใช้เป็นแบบจำลองข้อมูลเชิงตรรก (Integrate User Views Into Composite Logical Data Model)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



ขั้นตอนการสร้างแบบจำลองข้อมูลเชิงตรรก

ในขั้นตอนที่ 1 เป็นการออกแบบแนวคิดซึ่งจะได้โครงสร้างระดับบนซึ่งเป็นเค้าร่างฐานข้อมูล ส่วนขั้นตอนที่ 2 และ 3 เป็นการออกแบบเชิงตรรกที่ผู้ออกแบบต้องพิจารณาข้อมูลให้อยู่ในแบบฟอร์มปกติ (Normal Form) และเป็นไปตามข้อกำหนดทางธุรกิจ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สร้างและปรับปรุงโครงร่างในมุมมองของผู้ใช้

เทคนิคการสร้างแบบจำลองเชิงตรรกเริ่มจากการสร้างโครงร่างในมุมมองของผู้ใช้ (Skeletal User View) มุมมองของผู้ใช้ 1 คน หมายถึง 1 ฟังก์ชั่นธุรกิจ (Business Function) โครงร่างในมุมมองของผู้ใช้จะแสดงถึงข้อมูลที่ต้องใช้ในฟังก์ชั่นธุรกิจเท่านั้นจะไม่มีรายละเอียด ข้อ

กำหนด หรือค่าของข้อมูลแต่ละตัว คำศัพท์ (Terminology) ที่ใช้ในการสร้างโครงร่างในมุมมองของผู้ใช้คือ

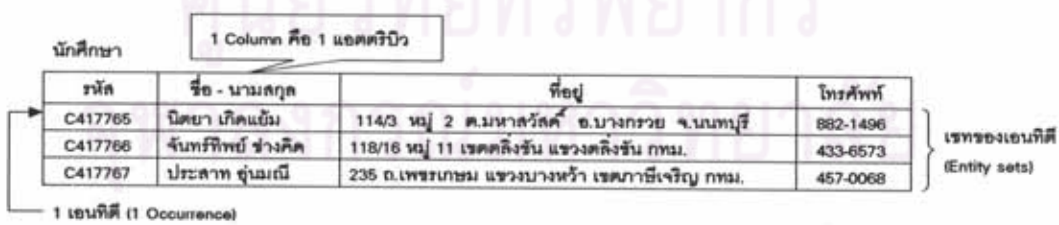
1. เอนทิตี (Entity) หมายถึงระเบียน
2. ความสัมพันธ์ (Relationships)
3. กุญแจหลัก หรือคีย์หลัก (Primary Keys)
4. กุญแจสำรอง หรือคีย์รอง (Alternate Keys)
5. กุญแจภายนอก หรือฟอเรนคีย์ (Foreign Keys)
6. กฎทางธุรกิจ (Key Business Rules)

การสร้างและปรับปรุงโครงร่างในมุมมองของผู้ใช้ประกอบด้วย 5 ขั้นตอนคือ

1. กำหนดเอนทิตีหลัก (Identify Major Entities) ซึ่งมีแนวทางดังนี้

1.1 จดหรือบันทึกชิ้นข้อมูล (Data Item) ที่ผู้ใช้แต่ละคนบอก เช่น รหัสนักศึกษา ชื่อ ที่อยู่ และหมายเลขโทรศัพท์ของนักศึกษา เป็นต้น

1.2 จัดหมวดหมู่ชิ้นข้อมูลทั้งหมดเป็นเรื่องๆ เช่น รหัสนักศึกษา ชื่อนักศึกษา ที่อยู่นักศึกษา หมายเลขโทรศัพท์นักศึกษา รวมเป็น 1 กลุ่ม และกำหนดชื่อของข้อมูลแต่ละกลุ่ม ข้อมูลแต่ละกลุ่มนี้เรียกว่า เซทของเอนทิตี (Entity Sets) หรือประเภทเอนทิตี (Entity Types) หนึ่งเซทของเอนทิตีประกอบด้วยหลายเอนทิตี และหนึ่งเอนทิตีจะมีนิยามและรายละเอียดของข้อมูลหรือแอตตริบิวเหมือนกันทุกเอนทิตี เช่น รหัสนักศึกษา ชื่อ ที่อยู่ และหมายเลขโทรศัพท์นักศึกษา จัดเป็นกลุ่มเดียวกันและกำหนดให้ชื่อ "นักศึกษา" รายละเอียดนักศึกษา 1 คนเรียกว่า 1 เอนทิตี ข้อมูลใน "นักศึกษา" จะมีรายละเอียดของนักศึกษาหลายคน หรือหลายเอนทิตี ซึ่งเรียกว่า เซทของเอนทิตี และต่อไปนี้จะใช้คำว่า เอนทิตี แทน เซทของเอนทิตี ดังตัวอย่างในรูป



ตัวอย่างเซทของเอนทิตีชื่อ "นักศึกษา" ซึ่งในที่นี้ประกอบด้วย 3 เอนทิตี และ 4 แอตตริบิว

ในการสร้างโครงร่างครั้งแรกนี้อาจพิจารณาแยกเอนทิตีเป็น ซับไทป์ (Subtype) เช่น เอนทิตี "นักศึกษา" อาจแยกเป็น "นักศึกษาที่กำลังศึกษาอยู่" และ "นักศึกษาที่สำเร็จการศึกษา"

วาดภาพสี่เหลี่ยมแทนเอนทิตี พร้อมทั้งเขียนชื่อเอนทิตี และจำนวนรายการที่คาดว่าจะมีในเอนทิตีนั้น ดังรูป

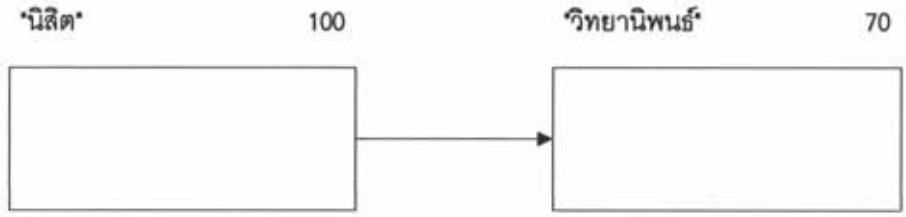
แสดงการแยกเอนทิตี 'นักศึกษา' ออกเป็น ซับไทป์ คือ เอนทิตี 'นักศึกษาที่กำลังศึกษา' และเอนทิตี 'นักศึกษาที่สำเร็จการศึกษา' พร้อมตัวเลขแสดงจำนวนข้อมูล (Occurrences) ในแต่ละเอนทิตี

2. ความสัมพันธ์ (Relationships)

กำหนดชื่อ ทิศทาง ความหมาย และอัตราส่วนความสัมพันธ์ระหว่าง 2 เอนทิตีที่มีความสัมพันธ์กัน โดยจะแสดงความสัมพันธ์จากเอนทิตีหลักซึ่งเรียกว่า เอนทิตีพ่อแม่ (Parent Entity) ไปหาเอนทิตีรองซึ่งเรียกว่า เอนทิตีลูก (Child Entity) แทนความสัมพันธ์ และทิศทางด้วยลูกศร หัวลูกศรจะชี้ไปที่เอนทิตีลูก อัตราส่วนที่กำหนดในขั้นตอนนี้จะช่วยตรวจสอบจำนวนรายการของเอนทิตีที่กำหนดไว้ในขั้นตอนที่ 1 ได้ จากอัตราส่วนความสัมพันธ์ระหว่างเอนทิตีทำให้สามารถแบ่งความสัมพันธ์ได้เป็น 3 ประเภทคือ

2.1 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One-To-One หรือ 1:1 Relationship) เมื่อพิจารณาระเบียนใดระเบียนหนึ่งจากเอนทิตีหนึ่ง จะมีความสัมพันธ์กับระเบียนในอีกเอนทิตีเพียงหนึ่งระเบียนเท่านั้น และในทางกลับกัน เช่น นักศึกษา 1 คนมีรหัสสมาชิกห้องสมุดเพียง 1 รหัส คอมพิวเตอร์ และรหัสคอมพิวเตอร์ 1 รหัสจะมีนักศึกษาเป็นสมาชิกรหัสนั้นได้เพียง 1 คนเท่านั้น

2.2 ความสัมพันธ์แบบหนึ่งต่อหลาย (One-To-Many, หรือ 1:N Relationship) เอนทิตีด้านที่มีอัตราส่วน 1 เรียกว่าเอนทิตีพ่อแม่ ด้านที่มีอัตราส่วนมากกว่า 1 (N) เรียกว่า เอนทิตีลูก เอนทิตีพ่อแม่ 1 ระเบียน มีความสัมพันธ์กับเอนทิตีลูกมากกว่า 1 ระเบียน และในทางกลับกัน เอนทิตีลูกหลายระเบียน อาจไม่มีความสัมพันธ์กับเอนทิตีพ่อแม่ หรือมีเพียง 1 ระเบียน เช่น รหัสประเภทของสมาชิกห้องสมุด 1 รหัส (อาจารย์, เจ้าหน้าที่,นักศึกษา) สามารถมีสมาชิกห้องสมุดได้หลายคน แต่สมาชิกห้องสมุด 1 คนมีรหัสประเภทของสมาชิกห้องสมุดได้เพียง 1 รหัสเท่านั้น ดังรูป



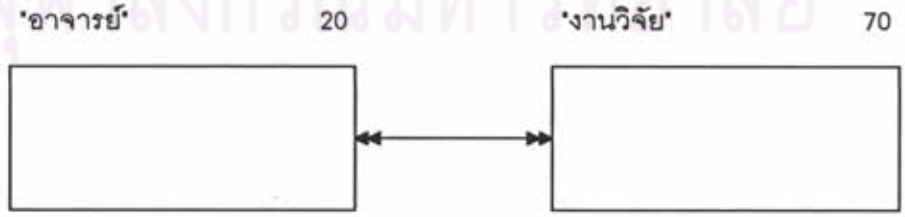
รูปแสดงความสัมพันธ์ระหว่างเอนทิตีแบบหนึ่งต่อหนึ่ง
(One-to-one หรือ 1 : 1 relationship)

รูปแสดงความสัมพันธ์ระหว่างเอนทิตีแบบหนึ่งต่อหลาย

(One-to-many หรือ 1 : N relationalship)

2.3 ความสัมพันธ์แบบหลายต่อหลาย (Many-to-many, หรือ M:N relationship) ความสัมพันธ์ประเภทนี้เมื่อพิจารณา 2 ทิศทางจะมีอัตราส่วนมากกว่า 1 ทั้งสองเอนทิตี เช่น งานวิจัยของอาจารย์ อาจารย์ 1 คนสามารถทำงานวิจัยได้หลายเรื่อง และงานวิจัย 1 เรื่องอาจมีอาจารย์ที่ทำวิจัย 1 คน หรือมากกว่า 1 คน ดังรูป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.7 แสดงความสัมพันธ์ระหว่างเอนทิตีแบบหลายต่อหลาย
(Many-to-many หรือ M : N relationship)

ความสัมพันธ์ประเภทนี้จะมีความซับซ้อนต่อการออกแบบฐานข้อมูล และไม่มีระบบฐานข้อมูลใดในปัจจุบันรองรับได้ จึงต้องเปลี่ยนความสัมพันธ์นี้ให้อยู่ในรูปความสัมพันธ์แบบหนึ่งต่อหลาย โดยการเพิ่มเอนทิตีขึ้น 1 เอนทิตี เพื่อใช้เป็นตัวกลาง ดังตัวอย่างข้างต้น ให้เพิ่มเอนทิตี "อาจารย์และงานวิจัย" และสร้างความสัมพันธ์นี้กับเอนทิตีเดิม เป็นแบบหนึ่งต่อหลาย ดังแสดงใน

รูปที่ 4.8 แสดงตัวอย่างการเปลี่ยนความสัมพันธ์ระหว่างเอนทิตีแบบหลายต่อหลาย เป็นแบบหนึ่งต่อหลาย

3. กำหนดคีย์หลัก และคีย์รอง (Determine primary and alternate keys)

ใส่รายละเอียด หรือแอตตริบิว (Attribute) ให้แต่ละเอนทิตี พร้อมตั้งชื่อให้แต่ละแอตตริบิว แอตตริบิวคือหน่วยที่เล็กที่สุดของเอนทิตีที่มีความหมาย หรือที่เรารู้จักทั่วไปคือ เขตข้อมูล (field) นั่นเอง เลือกแอตตริบิว หรือกลุ่มแอตตริบิวกำหนดให้เป็นคีย์หลักเพื่อใช้อ้างอิงถึงแต่ละระเบียน คีย์หลักนี้จะมีค่าว่าง (Null value) ไม่ได้ และกำหนดคีย์รองเพื่อใช้ในการสืบค้นข้อมูลหรือสร้างความสัมพันธ์กับเอนทิตีอื่น

การกำหนดคีย์หลักเป็นการกำหนดกฎธุรกิจที่จะทำให้เกิดบูรณภาพเอนทิตี (Entity integrity) ดังได้กล่าวไว้แล้วในบทที่ 3 คือ

3.1 ข้อมูลทุกระเบียนของเอนทิตีจะต้องมีคีย์หลัก

3.2 คีย์หลักของแต่ละระเบียนจะซ้ำกันไม่ได้ คีย์หลักต้องเป็นหนึ่งหรือยูนีค(Unique)

4. กำหนดกุญแจภายนอก หรือฟอเรนคีย์ (Determine foreign keys)

ฟอเรนคีย์คือ แอตตริบิวในเอนทิตีหนึ่งที่มีค่าตรงกับคีย์หลักในอีกเอนทิตีหนึ่งซึ่งจะเป็นเอนทิตีพ่อแม่ ส่วนเอนทิตีที่มีฟอเรนคีย์จะเป็นเอนทิตีลูก

5. กำหนดกฎธุรกิจ (Determine key business rules)

ในขั้นตอนนี้จะเป็นการกำหนดบูรณภาพของข้อมูล (Data integrity) เพื่อให้ข้อมูลมีความถูกต้อง และสอดคล้องกับความเป็นจริงทางธุรกิจ บูรณภาพของข้อมูลจะมีได้จากการกำหนดกฎธุรกิจ 3 ประเภทนี้คือ

5.1 กฎธุรกิจหลัก (Key business rules) เป็นข้อกำหนด การเพิ่ม (Insert) การลบ (Delete) หรือการปรับปรุง (Update) ข้อมูลของ 2 เอนทิตีที่สัมพันธ์กัน กฎข้อนี้จะช่วยตรวจสอบความสัมพันธ์ระหว่าง 2 เอนทิตีซึ่งเชื่อมโยงกันด้วยฟอเรนคีย์ เพื่อให้ข้อมูลของทั้ง 2 เอนทิตีมีความสอดคล้องกัน

5.2 โดเมน (Domain) การกำหนดค่าที่เป็นไปได้ของแต่ละแอตตริบิว

5.3 การดำเนินการทริกเกอร์ (Triggering operations) การกำหนดผลกระทบที่เกิดจากการเพิ่ม ลบ หรือค้นคืนข้อมูลในเอนทิตีอื่น หรือแอตตริบิวอื่นในเอนทิตีเดียวกัน

จาก 5 ขั้นตอนข้างต้น ผู้ออกแบบจะได้โครงร่างในมุมมองของผู้ใช้ ซึ่งเพียงพอที่จะนำไปสร้างเป็นต้นแบบ (Prototype) แต่โครงร่างนี้อาจไม่สมบูรณ์ ผู้ออกแบบจะต้องพิจารณาในรายละเอียดในขั้นตอนต่อไป

การเพิ่มรายละเอียดในมุมมองของผู้ใช้

การเพิ่มรายละเอียดในมุมมองของผู้ใช้มี 4 ขั้นตอนคือ

1. เพิ่มแอตตริบิวที่เหลือ (Add remaining attributes)

ในขั้นตอนที่ผ่านมาจะเป็นการกำหนดข้อมูลหลักของเอนทิตี คือแอตตริบิวที่ใช้เป็นคีย์หลัก คีย์รอง และฟอเรนคีย์ เพื่อกำหนดกฎธุรกิจให้เป็นไปตามความเป็นจริงทางธุรกิจ ในขั้นตอนนี้จะเป็นการพิจารณาข้อมูลอื่น ๆ ที่ไม่ใช่คีย์ ซึ่งเป็นข้อมูลที่จำเป็นในแต่ละเอนทิตี ข้อมูลใดจะใส่เป็นแอตตริบิวของเอนทิตีใดให้พิจารณาว่า ค่าของแอตตริบิวนั้นขึ้นอยู่กับค่าของแอตตริบิวที่เป็นคีย์หลักทั้งหมด (คีย์หลักอาจประกอบด้วยหนึ่งหรือหลายแอตตริบิว) นอกจากนี้ถ้าแอตตริบิวใดขึ้นอยู่กับคีย์หลักทั้งหมดแล้ว แต่แอตตริบิวนั้นมีค่าได้มากกว่า 1 ค่า (Multivalued) คือไม่ใช่แฟ้มระนาบ (Flat file) ให้สร้างเอนทิตีใหม่สำหรับแอตตริบิวนั้น

ในบางกรณีผู้ออกแบบอาจพบว่าข้อมูลบางชนิดสามารถกำหนดเป็นแอตตริบิวหนึ่งหรือกำหนดเป็นเอนทิตีหนึ่งก็ได้ ถ้าพบว่าข้อมูลนั้นอาจมีรายละเอียดประกอบก็ควรจะสร้างเป็นเอนทิตี ถ้าไม่มีรายละเอียดอย่างอื่นเพิ่มเติมก็จัดเป็นแอตตริบิวหนึ่งในเอนทิตีที่เกี่ยวข้อง เช่น สี ถ้าต้องกำหนดสูตรการผสมสีก็ควรแยกสีเป็นเอนทิตี เพื่อสามารถเพิ่มรายละเอียด แต่ถ้าเพียงต้องการกำหนดว่าเป็นสีอะไรก็อาจจัดเป็นแอตตริบิวในเอนทิตี อย่างไรก็ตามการตัดสินใจเกี่ยวกับข้อมูลเช่นนี้ควรจะให้ผู้ใช้ระบบช่วยให้ข้อมูล

2. พิจารณาให้เป็นไปตามกฎนอร์มัลไลเซชัน (Validate normalization rules)

การเพิ่มแอตตริบิวในเอนทิตีในขั้นตอนที่แล้วจะใช้ความเข้าใจทั่ว ๆ ไป ในขั้นตอนนี้จะใช้กฎนอร์มัลไลเซชันช่วยในการพิจารณาข้อมูลอีกครั้ง กฎนอร์มัลไลเซชันจะช่วยทำให้ได้โครงสร้างข้อมูลที่มีความซ้ำซ้อน (Redundancy) น้อยที่สุด ทำให้ลดปัญหาการดูแลข้อมูลซึ่งจะทำให้มีความถูกต้อง (Correctness) สอดคล้องกัน (Consistency) และมีเสถียรภาพ (Stable) มากขึ้น

กฎนอร์มัลไลเซชันที่กำหนดโดย E.F.Codd มี 3 ระดับดังนี้

2.1 ฟอร์มที่ 1 (First normal form, 1NF) เมื่อพิจารณาข้อมูลแต่ละระเบียน (Each entity occurrence) ในเอนทิตี แต่ละแอตตริบิวของระเบียนนั้น ๆ ควรมีเพียง 1 ค่า ถ้าพบว่าบางแอตตริบิวมีค่าได้หลายค่า (Multivalues) ให้แยกเป็นอีกเอนทิตี

2.2 ฟอร์มที่ 2 (Second normal form, 2NF) จากฟอร์มที่ 1 พิจารณาข้อมูลของแอตตริบิวที่ไม่ใช่คีย์ ข้อมูลของแอตตริบิวตัวใดไม่ขึ้นอยู่กับข้อมูลในคีย์หลัก หรือขึ้นอยู่กับบางส่วนของคีย์หลัก ให้แยกแอตตริบิวนั้นเป็นอีกเอนทิตี

2.3 ฟอร์มที่ 3 (Third normal form, 3NF) จากฟอร์มที่ 2 ให้ขจัดแอตตริบิวที่มีข้อมูลที่ขึ้นอยู่กับแอตตริบิวอื่นที่ไม่ใช่คีย์หลัก

3. กำหนดโดเมน (Determine domains)

โดเมนในที่นี้หมายถึงค่าที่เป็นไปได้ของแต่ละแอตตริบิว ซึ่งประกอบด้วย

3.1 คำอธิบายความหมาย (Meaning) ของแอตตริบิวนั้น

3.2 ประเภทข้อมูล (Data type)

3.3 ความยาว (Length)

3.4 รูปแบบ (Format)

3.5 ช่วงค่าของข้อมูล (Range)

3.6 ค่าที่อนุญาต (Allowable values)

3.7 ความเป็นหนึ่งเดียว (Unique)

3.8 มีค่าว่างได้หรือไม่ (Null support)

3.9 ค่าโดยปริยาย (Default value)

4. กำหนดกฎธุรกิจของแอตตริบิวอื่น (Determine other attribute business rules) หรือกำหนดการดำเนินการทริกเกอร์ (Triggering operations)

การดำเนินการทริกเกอร์คือกฎธุรกิจหนึ่งดังได้กล่าวมาแล้วข้างต้น การดำเนินการทริกเกอร์คือ การดำเนินการอย่างใดอย่างหนึ่ง ได้แก่ การเพิ่ม ลบ ปรับปรุง หรือการค้นคืนข้อมูลเอนทิตีอื่น หรือแอตตริบิวอื่นในเอนทิตีเดียวกัน ภายใต้เหตุการณ์และเงื่อนไขที่กำหนด หรือเมื่อสภาวะการณ์บางอย่างเป็นจริง

รวบรวมมุมมองของผู้ใช้เป็นแบบจำลองข้อมูลเชิงตรรกะ

การรวบรวมมุมมองของผู้ใช้ประกอบด้วย 3 ขั้นตอนคือ

1. รวบรวมมุมมองของผู้ใช้ (Combine use views)

ขั้นตอนนี้จะเป็นการรวบรวมมุมมองของผู้ใช้ทั้งหมดเข้าด้วยกัน เพื่อให้แน่ใจว่าแบบจำลองข้อมูลเชิงตรรกะนี้ได้แสดงมุมมองของผู้ใช้ได้อย่างถูกต้องครบถ้วน ลดความซ้ำซ้อน แก้ไขความไม่สอดคล้องระหว่างมุมมองของผู้ใช้ และเพิ่มความสัมพันธ์ระหว่างมุมมองของผู้ใช้แต่ละแบบ พร้อมทั้งกฎธุรกิจ การรวมมุมมองของผู้ใช้สามารถทำได้ 3 ลักษณะ คือการรวมเอนทิตีพร้อมกฎธุรกิจที่เกี่ยวข้อง การรวมความสัมพันธ์พร้อมกฎธุรกิจที่เกี่ยวข้องเมื่อมีการเพิ่ม หรือลบข้อมูล และการรวมแอตตริบิวพร้อมกฎธุรกิจที่เกี่ยวข้อง

2. รวบรวมแบบจำลองข้อมูลที่มีอยู่แล้ว (Integrate with existing data models)

ในขั้นตอนนี้เป็นการรวมแบบจำลองข้อมูลเชิงตรรกะที่ได้ใหม่นี้กับแบบจำลองข้อมูลเชิงตรรกะที่มีอยู่แล้ว หรือเค้าร่างมโนภาพธุรกิจที่มีอยู่เดิม เพื่อให้เป็นเค้าร่างมโนภาพทางธุรกิจทั้งหมด (Business conceptual schema) การรวมกันของแบบจำลองข้อมูลกับเค้าร่างมโนภาพที่มีอยู่เดิมนี้อาจเกิดปัญหาการเข้ากันไม่ได้ หรือไม่สอดคล้องกันก็อาจต้องแก้ไขแบบจำลองเพื่อให้สามารถรวมกันได้ ถ้าไม่สามารถแก้ไขได้อาจจำเป็นต้องบันทึกปัญหานี้ไว้เพื่อทำการแก้ไขต่อไปในโครงการต่อไป

3. วิเคราะห์เสถียรภาพและการเติบโต (Analyze for stability and growth)

ขั้นตอนที่ผ่านมาเป็นการมองความต้องการใช้งานในปัจจุบันเท่านั้น เพื่อให้แบบจำลองข้อมูลเชิงตรรกะที่ได้มีเสถียรภาพ และสามารถทำการขยายต่อไปได้ ผู้วิเคราะห์ควร

พิจารณาเพิ่มเติมว่าแบบจำลองที่ทำอยู่นี้จะต้องทำการเปลี่ยนแปลงอะไรบ้าง ถ้าการดำเนินการทางธุรกิจเปลี่ยน หรือมุมมองของผู้ใช้เปลี่ยนไปจากเดิม ผู้วิเคราะห์อาจพบว่าการเปลี่ยนแปลงแบบจำลองบางอย่างสามารถทำการขยายได้ในอนาคตได้ หรือถ้าจะทำการเปลี่ยนแปลงหรือปรับปรุงแบบจำลองข้อมูลทันทีเพื่อให้สามารถรองรับการเปลี่ยนแปลงดังกล่าวจะมีผลการะทบต่อการดำเนินงานแต่วันละวันในปัจจุบัน ผู้วิเคราะห์อาจเลือกที่จะทำการบันทึกการเปลี่ยนแปลงที่มีผลต่อแบบจำลองข้อมูล พร้อมทั้งเหตุผล และความเป็นไปได้ของการเปลี่ยนแปลงนั้น ๆ เพราะบันทึกดังกล่าวจะมีผลต่อการตัดสินใจในการออกแบบฐานข้อมูลทันทีในปัจจุบัน หรืออาจมีผลต่อการปรับปรุง หรือการออกแบบจำลองข้อมูลเชิงตรรกะในอนาคต

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย