บทที่ 4

ขั้นตอนการออกแบบฐานข้อมูล

ขั้นตอนการออกแบบฐานข้อมูล (Phase of database design) (Mannila and Raiha, 1992)

1. วิเคราะห์ความต้องการของผู้ใช้ระบบ (Requirement analysis)

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

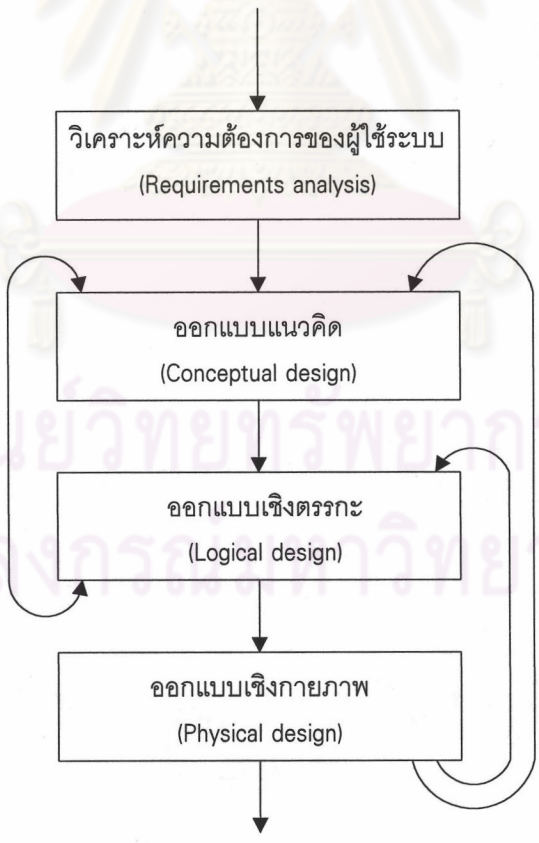
2. ออกแบบแนวคิด (Conceptual design) ผลของขั้นตอนนี้เป็นการอธิบายโครงสร้างของข้อมูลว่าควรมีข้อมูลอะไรบ้างในฐานข้อมูลซึ่งเป็นโครงสร้างในระดับบน (High-level Term) เรียกว่า เค้าร่างฐานข้อมูล (Database schema)

ในขั้นตอนนี้ต้องศึกษาแนวคิดต่างๆ รวมทั้งแนวคิดของผู้ใช้ระบบแต่ละท่าน เพื่อหาแนวคิดที่จำเป็นซึ่งได้จากคำอธิบายขั้นตอนการปฏิบัติงานที่ได้จากการวิเคราะห์ความต้องการในขั้นตอนแรก เพราะเป้าหมายที่จำเป็นของการออกแบบฐานข้อมูลคือ การเก็บข้อเท็จจริงที่ไม่ซ้ำซ้อน ในขั้นตอนนี้อาจใช้แบบจำลองข้อมูล (Data model) เพื่อใช้อธิบายแนวคิดต่าง ๆ และการทำทำวิจัยครั้งนี้ก็เลือกใช้แบบจำลองข้อมูลเชิงตรรกะ (Logical data model) มาจำลองแนวคิด เพราะเป็นแบบจำลองที่เข้าใจง่าย ไม่อิงกับโปรแกรมประยุกต์ หรือเทคโนโลยีใด ๆ สามารถขยายขอบเขตงานโดยมีผลกระทบต่องานที่มีอยู่แล้วน้อยมากหรืออาจไม่มีเลย ใช้แสดงโครงสร้างและการใช้ข้อมูลได้สอดคล้องกับแนวทางทางธุรกิจ

3. ออกแบบเชิงตรรกะ (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 ในขั้นตอนนี้จะต้องทราบถึงรูปแบบต่าง ๆ ในมุมมองของผู้ใช้ระบบหรือคิวรี (Query) ความถี่ในการเรียกใช้งาน หรือความถี่ในการประมวลผล เพื่อใช้ปรับโครงสร้างให้เหมาะสมกับการใช้งาน เพราะผลการทำงาน (Performance) ขึ้นอยู่กับโครงสร้างของตารางหรือแฟ้มข้อมูล ปริมาณข้อมูล และความถี่ของการใช้งาน ดังนั้นบางระบบอาจต้องใช้เวลาและต้องทำการประเมินผลอย่างสม่ำเสมอ เพราะปริมาณข้อมูลน้อยอาจมองไม่เห็นผลการทำงานที่ถูกต้อง เมื่อมีการเปลี่ยนแปลงจำนวนข้อมูลมากขึ้น จึงจะเห็นผลการทำงานที่ถูกต้อง ซึ่งอาจทำให้ต้องปรับโครงสร้างฐานข้อมูลใหม่

ในการทำงานแต่ละขั้นตอนควรมีการตรวจสอบผลป้อนกลับ (Feedback) เพื่อให้แน่ใจว่าผลที่ได้ในแต่ละขั้นตอนถูกต้องตามที่ผู้ออกแบบต้องการ ดังรูปที่ 4.1



รูปที่ 4.1 ขั้นตอนการออกแบบฐานข้อมูล (Mannila and Raiha, 1992)

แบบจำลองข้อมูลเชิงตรรกะ (Logical data model) (Flaming, Halle, 1989)

การจำลองข้อมูลเชิงตรรกะเป็นเทคนิคที่ใช้แทนโครงสร้างสารสนเทศเชิงธุรกิจ 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 ทั้งนี้เพราะเทคนิคนี้ไม่สนใจว่าจะทำอย่างไร และเนื่องจากการใช้แผนภาพ (Diagram) และคำอธิบายประกอบแทนการใช้คำพรรณนา จึงเข้าใจง่ายแม้จะไม่มีความรู้ทางด้านคอมพิวเตอร์

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

การจำลองข้อมูลเชิงตรรกะนี้ได้ใช้แนวคิดมาจาก การจัดข้อมูลโดยระบบจัดการฐานข้อมูลที่ใช้สถาปัตยกรรม 3 ระดับของ ANSI/X3/SPARC ที่ได้กล่าวไว้ในบทที่ 3

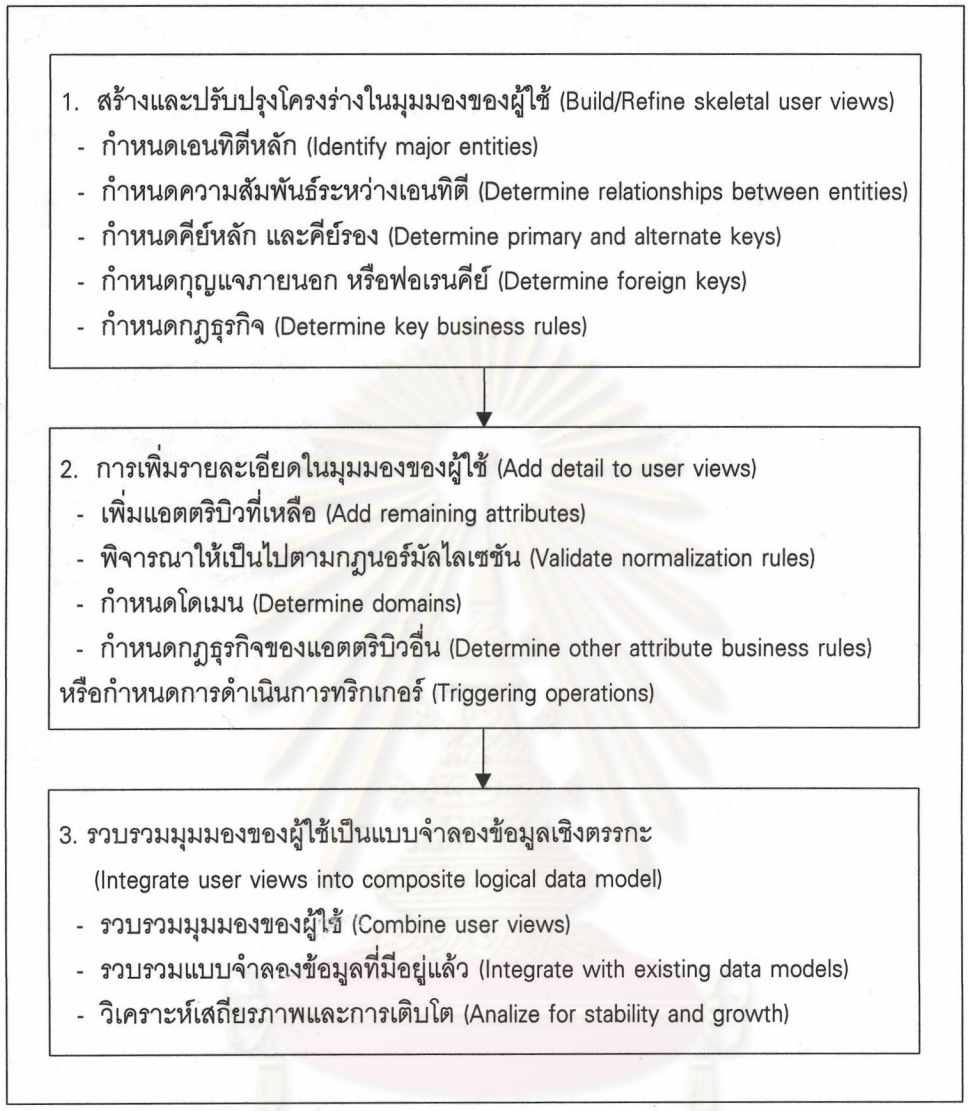
การสร้างแบบจำลองข้อมูลเชิงตรรกะ (Fleming, Halle, 1989)

การสร้างแบบจำลองข้อมูลเชิงตรรกะมี 3 ขั้นตอนหลักๆ ดังรูปที่ 4.2 ดังนี้

1. สร้างและปรับปรุงโครงร่างในมุมมองของผู้ใช้ (Build/Refine skeletal user views)

2. การเพิ่มรายละเอียดในมุมมองของผู้ใช้ (Add detail to user views)

3. รวบรวมมุมมองของผู้ใช้เป็นแบบจำลองข้อมูลเชิงตรรกะ (Integrate user views into composite logical data model)



รูปที่ 4.2 ขั้นตอนการสร้างแบบจำลองข้อมูลเชิงตรรกะ

ในขั้นตอนที่ 1 เป็นการออกแบบแนวคิดซึ่งจะได้โครงสร้างระดับบนซึ่งเป็นเค้าร่างฐานข้อมูล ส่วนขั้นตอนที่ 2 และ 3 เป็นการออกแบบเชิงตรรกะที่ผู้ออกแบบต้องพิจารณาข้อมูลให้อยู่ในแบบฟอร์มปกติ (Normal form) และเป็นไปตามข้อกำหนดทางธุรกิจ

สร้างและปรับปรุงโครงร่างในมุมมองของผู้ใช้

เทคนิคการสร้างแบบจำลองเชิงตรรกะเริ่มจากการสร้างโครงร่างในมุมมองของผู้ใช้ (Skeletal user view) มุมมองของผู้ใช้ 1 คน หมายถึง 1 ฟังก์ชันธุรกิจ (Business function) โครงร่าง

ในมุมมองของผู้ใช้จะแสดงถึงข้อมูลที่ต้องใช้ในฟังก์ชันธุรกิจเท่านั้นจะไม่มีรายละเอียด ข้อกำหนด หรือค่าของข้อมูลแต่ละตัว คำศัพท์ (Terminology) ที่ใช้ในการสร้างโครงร่างในมุมมองของผู้ใช้คือ

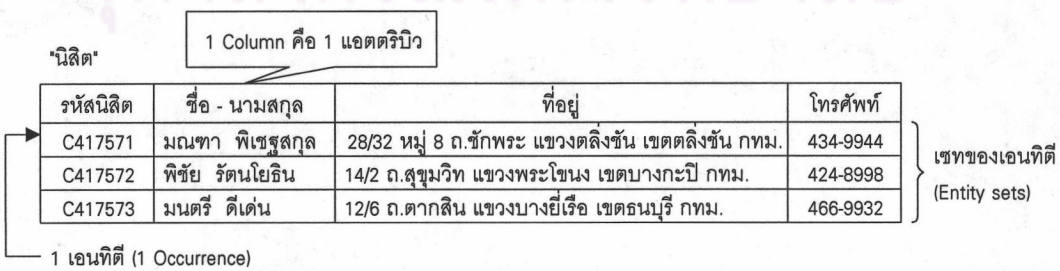
1. เอนทิตี (Entity) หมายถึงระเบียน
2. ความสัมพันธ์ (Relationships)
3. กุญแจหลัก หรือคีย์หลัก (Primary keys)
4. กุญแจสำรอง หรือคีย์รอง(Alternate keys)
5. กุญแจภายนอก หรือฟอเรนคีย์ (Foreign keys)
6. กฎทางธุรกิจ (Key business rules)

การสร้างและปรับปรุงโครงร่างในมุมมองของผู้ใช้ประกอบด้วย 5 ขั้นตอนคือ

1. กำหนดเอนทิตีหลัก (Identify major entities) ซึ่งมีแนวทางดังนี้

1.1 จดหรือบันทึกชิ้นข้อมูล (Data item) ที่ผู้ใช้แต่ละคนบอก เช่น รหัสนิสิต ชื่อและที่อยู่นิสิต หัวข้อวิทยานิพนธ์ภาษาไทย หัวข้อวิทยานิพนธ์ภาษาอังกฤษ สถานะโครงร่างวิทยานิพนธ์ วันที่เสนอวิทยานิพนธ์ เป็นต้น

1.2 จัดหมวดหมู่ชิ้นข้อมูลทั้งหมดเป็นเรื่องๆ เช่น รหัสนิสิต ชื่อนิสิต ที่อยู่นิสิต รวมเป็น 1 กลุ่ม และกำหนดชื่อของข้อมูลแต่ละกลุ่ม ข้อมูลแต่ละกลุ่มนี้เรียกว่า เซทของเอนทิตี (Entity sets) หรือประเภทเอนทิตี (Entity types) หนึ่งเซทของเอนทิตีประกอบด้วยหลายเอนทิตี และหนึ่งเอนทิตีจะมีนิยามและรายละเอียดของข้อมูลหรือแอตตริบิวเหมือนกันทุกเอนทิตี เช่น รหัสนิสิต ชื่อและที่อยู่นิสิต จัดเป็นกลุ่มเดียวกันและกำหนดให้ชื่อ "นิสิต" รายละเอียดนิสิต 1 คนเรียกว่า 1 เอนทิตี ข้อมูลใน "นิสิต" จะมีรายละเอียดของนิสิตหลายคน หรือหลายเอนทิตี ซึ่งเรียกว่า เซทของเอนทิตี และต่อไปนี้จะใช้คำว่า เอนทิตี แทน เซทของเอนทิตี ดังตัวอย่างในรูปที่ 4.3



รูปที่ 4.3 ตัวอย่างเซทของเอนทิตีชื่อ "นิสิต" ซึ่งในที่นี้ประกอบด้วย 3 เอนทิตี และ 4 แอตตริบิว

I16658759

ในการสร้างโครงร่างครั้งแรกนี้อาจพิจารณาแยกเอนทิตีเป็น ซับไทป์ (Subtype) เช่น เอนทิตี "นิสิต" อาจแยกเป็น "นิสิตที่กำลังศึกษาอยู่" และ "นิสิตที่สำเร็จการศึกษา" วาดภาพสี่เหลี่ยมแทนเอนทิตี พร้อมทั้งเขียนชื่อเอนทิตี และจำนวนรายการที่คาดว่าจะมีในเอนทิตีนั้น ดังรูปที่ 4.4

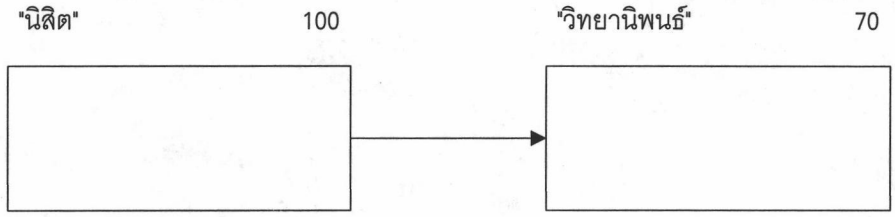
รูปที่ 4.4 แสดงการแยกเอนทิตี "นิสิต" ออกเป็น ซับไทป์ คือ เอนทิตี "นิสิตที่กำลังศึกษา" และเอนทิตี "นิสิตที่สำเร็จการศึกษา" พร้อมตัวเลขแสดงจำนวนข้อมูล (Occurrences) ในแต่ละเอนทิตี

2. กำหนดความสัมพันธ์ระหว่างเอนทิตี (Determine relationships between entities)

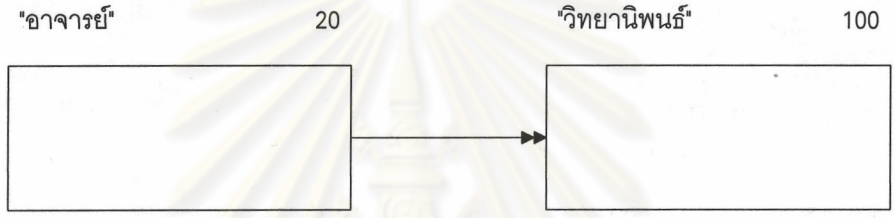
กำหนดชื่อ ทิศทาง ความหมาย และอัตราส่วนความสัมพันธ์ระหว่าง 2 เอนทิตีที่มีความสัมพันธ์กัน โดยจะแสดงความสัมพันธ์จากเอนทิตีหลักซึ่งเรียกว่า เอนทิตีพ่อแม่ (Parent entity) ไปหาเอนทิตีรองซึ่งเรียกว่า เอนทิตีลูก (Child entity) แทนความสัมพันธ์ และทิศทางด้วยลูกศร หัวลูกศรจะชี้ไปที่เอนทิตีลูก อัตราส่วนที่กำหนดในขั้นตอนนี้จะช่วยตรวจสอบจำนวนรายการของเอนทิตีที่กำหนดไว้ในขั้นต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

2.1 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One-to-one หรือ 1:1 relationship) เมื่อพิจารณาระเบียนใดระเบียนหนึ่งจากเอนทิตีหนึ่ง จะมีความสัมพันธ์กับระเบียนในอีกเอนทิตีเพียงหนึ่งระเบียนเท่านั้น และในทางกลับกัน เช่น นิสิต 1 คนจะทำวิทยานิพนธ์เพียง 1 หัวข้อ และหัวข้อวิทยานิพนธ์ 1 เรื่องจะมีนิสิตทำได้เพียง 1 คนเท่านั้น ดังรูปที่ 4.5

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 อาจารย์ 1 คน เป็นที่ปรึกษาหลักให้วิทยานิพนธ์ได้หลายหัวข้อ แต่วิทยานิพนธ์ 1 หัวข้อมีอาจารย์ที่ปรึกษาหลักได้เพียง 1 คน ดังรูปที่ 4.6

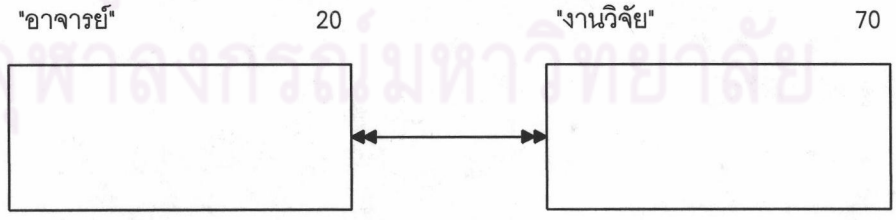


รูปที่ 4.5 แสดงความสัมพันธ์ระหว่างเอนทิตีแบบหนึ่งต่อหนึ่ง
(One-to-one หรือ 1 : 1 relationship)



รูปที่ 4.6 แสดงความสัมพันธ์ระหว่างเอนทิตีแบบหนึ่งต่อหลาย
(One-to-many หรือ 1 : N relationship)

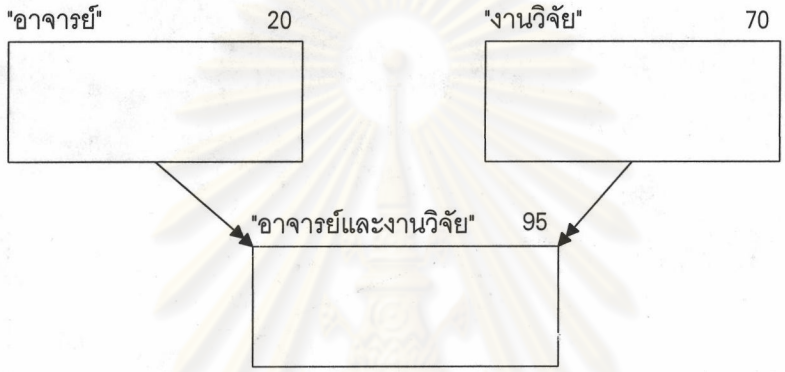
2.3 ความสัมพันธ์แบบหลายต่อหลาย (Many-to-many, หรือ M:N relationship) ความสัมพันธ์ประเภทนี้เมื่อพิจารณา 2 ทิศทางจะมีอัตราส่วนมากกว่า 1 ทั้งสองเอนทิตี เช่น งานวิจัยของอาจารย์ อาจารย์ 1 คนสามารถทำงานวิจัยได้หลายเรื่อง และงานวิจัย 1 เรื่องอาจมีอาจารย์ที่ทำวิจัย 1 คน หรือมากกว่า 1 คน ดังรูปที่ 4.7



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

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ความสัมพันธ์ประเภทนี้จะมีความซับซ้อนต่อการออกแบบฐานข้อมูล และไม่มีระบบฐานข้อมูลใดในปัจจุบันรองรับได้ จึงต้องเปลี่ยนความสัมพันธ์นี้ให้อยู่ในรูปความสัมพันธ์แบบหนึ่งต่อหลาย โดยการเพิ่มเอนทิตีขึ้น 1 เอนทิตี เพื่อใช้เป็นตัวกลาง ดังตัวอย่างข้างต้น ให้เพิ่มเอนทิตี "อาจารย์และงานวิจัย" และสร้างความสัมพันธ์นี้กับเอนทิตีเดิม เป็นแบบหนึ่งต่อหลาย ดังแสดงในรูปที่ 4.8



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 คีย์หลักนี้จะมีค่างว่าง (Null value) ไม่ได้ และกำหนดคีย์รองเพื่อใช้ในการสืบค้นข้อมูล หรือสร้างความสัมพันธ์กับเอนทิตีอื่น

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

ในบางกรณีผู้ออกแบบอาจพบว่าข้อมูลบางชนิดสามารถกำหนดเป็นแอตตริบิวหนึ่ง หรือกำหนดเป็นเอนทิตีหนึ่งก็ได้ ถ้าพบว่าข้อมูลนั้นอาจมีรายละเอียดประกอบก็ควรจะสร้างเป็นเอนทิตี ถ้าไม่มีรายละเอียดอย่างอื่นเพิ่มเติมก็จัดเป็นแอตตริบิวหนึ่งในเอนทิตีที่เกี่ยวข้อง เช่น สี ถ้าต้อง

กำหนดสูตรการผสมสีก็ควรแยกสีเป็นเอนทิตี 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)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

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)

ขั้นตอนนี้จะเป็นการวบรวมมุมมองของผู้ใช้ทั้งหมดเข้าด้วยกัน 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 ผู้วิเคราะห์ควรพิจารณาเพิ่มเติมว่าแบบจำลองที่ทำอยู่นี้จะต้องทำการเปลี่ยนแปลงอะไรบ้าง ถ้าการดำเนินการทางธุรกิจเปลี่ยน หรือมุมมองของผู้ใช้เปลี่ยนไปจากเดิม ผู้วิเคราะห์อาจพบว่าการเปลี่ยนแปลงแบบจำลองบางอย่างสามารถทำการขยายได้ในอนาคตได้ หรือถ้าจะทำการเปลี่ยนแปลงหรือ

ปรับปรุงแบบจำลองข้อมูลทันทีเพื่อให้สามารถรองรับการเปลี่ยนแปลงดังกล่าวจะมีผลการะทบต่อการดำเนินงานแต่วันละวันในปัจจุบัน ผู้วิเคราะห์อาจเลือกที่จะทำการบันทึกการเปลี่ยนแปลงที่มีผลต่อแบบจำลองข้อมูล พร้อมทั้งเหตุผล และความเป็นไปได้ของการเปลี่ยนแปลงนั้น ๆ เพราะบันทึกดังกล่าวจะมีผลต่อการตัดสินใจในการออกแบบฐานข้อมูลทันทีในปัจจุบัน หรืออาจมีผลต่อการปรับปรุง หรือการออกแบบจำลองข้อมูลเชิงตรรกะในอนาคต

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย