บทที่ 3

ทฤษฎีฐานข้อมูล

ฐานข้อมูล (Database)

ฐานข้อมูล หมายถึง การนำข้อมูลในองค์กรที่มีความเกี่ยวข้องกันมารวมกันไว้อย่างมีระบบในที่เดียวกัน โดยผู้ใช้แต่ละคนจะมองข้อมูลในฐานคนละแง่มุม ตามจุดประสงค์ของการประยุกต์ใช้งาน โดยอาศัยระบบจัดการฐานข้อมูล (Database Management System) เป็นตัวควบคุม วัตถุประสงค์หลักของระบบจัดการฐานข้อมูล คือ การจัดหามุมมองให้แก่ผู้ใช้ โดยผู้ใช้ไม่จำเป็นต้องรู้รายละเอียดโครงสร้าง ลักษณะการจัดเก็บข้อมูลโดยแท้จริงว่าเป็นอย่างไร โดยระบบจะซ่อนรายละเอียดต่าง ๆ ของข้อมูลเหล่านั้นว่าเก็บและบำรุงรักษาอย่างไร ฐานข้อมูลปัจจุบันมีอยู่ 3 ชนิดได้แก่

1. โมเดลเชิงความสัมพันธ์ (Relational Model) ผู้ใช้ทั่ว ๆ ไปจะเห็นภาพโมเดลนี้ในลักษณะการเก็บข้อมูลในรูปตาราง โครงสร้างข้อมูลประกอบด้วยกลุ่มข้อมูลย่อยที่มีความสัมพันธ์ระหว่างกลุ่มข้อมูลด้วยค่าย่อยในบางสดมภ์ของแต่ละความสัมพันธ์ หรือตารางนั้น ๆ เมื่อเปรียบเทียบกับแฟ้มข้อมูลแบบลำดับ โมเดลเชิงสัมพันธ์มีแถว เปรียบเสมือนระเบียนของแฟ้มข้อมูล และมีสดมภ์ เปรียบเสมือนเขตข้อมูล ของแฟ้มข้อมูลนั่นเอง

2. โมเดลเชิงเครือข่าย (Network Model) ผู้ใช้จะมองโมเดลนี้ในลักษณะของการรวบรวมระเบียน และความสัมพันธ์ระหว่างระเบียน โครงสร้างข้อมูลคล้ายกับโครงสร้างแบบต้นไม้ แต่ความสัมพันธ์ยืดหยุ่นได้โดยที่ข้อมูลแต่ละระดับเกิดจากข้อมูลระเบียนแม่ไม่จำกัดจำนวน ทำให้โครงสร้างข้อมูลมีความซับซ้อน การเก็บข้อมูลเป็นลักษณะการเชื่อมโยงหรือตัวชี้ จากระเบียนแม่ เชื่อมโยงไปหาระเบียนลูก ความแตกต่างที่เห็น ได้แสดงความสัมพันธ์เอาไว้ นั่นก็คือ ระเบียนที่สัมพันธ์กันต้องมีค่าในเขตข้อมูลใดเขตข้อมูลหนึ่งเหมือนกัน สำหรับความสัมพันธ์ ในโมเดลเชิงเครือข่ายนี้จะเป็นไปอย่างเห็นได้ชัด การค้นหาข้อมูลจะทำให้โดยเชื่อมโยงความสัมพันธ์ ดังนั้น จึงไม่ต้องเก็บข้อมูลที่เป็นตัวเชื่อมความสัมพันธ์ไว้ในระเบียนของข้อมูลเหมือนโมเดลเชิงสัมพันธ์

3. โมเดลเชิงลำดับขั้น (Hierarchical Model) โมเดลแบบลำดับขั้นนี้มีโครงสร้างแบบต้นไม้ มีความสัมพันธ์ระหว่างข้อมูลคล้ายลักษณะการแตกกิ่งก้านของต้นไม้ โดยเริ่มจากรากเดียวเป็นหลัก แล้วแตกกิ่งก้านออก มีตั้งแต่ 1 กิ่งขึ้นไป จากนั้นกิ่งย่อยแต่ละกิ่งก็อาจแตกกิ่งต่อออกไปเรื่อย ๆ ข้อมูลที่อยู่ระดับบนสุดเรียกว่า ราก (Root) กิ่งที่เป็นต้นตอก่อนที่จะแตกกิ่งย่อย แต่ละจุดเรียกว่า พาเรนท์ (Parent) ซึ่งเปรียบเสมือนเป็นพ่อแม่ และเรียกกิ่งย่อยที่แตกแขนงออกไปว่า ไชล์ด (Child) ซึ่งเปรียบเสมือนลูก

จากโมเดลฐานข้อมูลทั้ง 3 ชนิด ที่กล่าวมาข้างต้นนี้ ปัจจุบันโมเดลที่ได้รับความนิยมมากคือ โมเดลเชิงความสัมพันธ์ ทั้งนี้ เพราะในท้องตลาดส่วนใหญ่จะมีฐานข้อมูลเชิงความสัมพันธ์ ซึ่งได้รับการพิสูจน์แล้วว่า เป็นระบบที่มีประสิทธิภาพสูง มีความยืดหยุ่น สามารถปรับเปลี่ยนรายงานตามความต้องการของผู้ใช้ได้ง่าย และมีลักษณะการจัดเก็บข้อมูลในรูปตาราง ทำให้การเปรียบเทียบข้อมูลได้ชัดเจน และผู้ใช้มีความคุ้นเคยกับรูปแบบตารางทำให้เกิดประโยชน์ ดังนี้

(1) สร้างความเข้าใจได้ง่ายในมุมมองของผู้ใช้

(2) โมเดลนี้มีเครื่องมือ ที่ช่วยให้ผู้ใช้สามารถปฏิบัติการ ภายใต้ความยุ่งยากของจัดทำข้อมูลด้วยคำสั่งง่าย ๆ ได้

(3) การจัดเก็บข้อมูลแบบกายภาพ มีความแตกต่างจากข้อมูลแบบตรรกโดยสิ้นเชิงนับว่าเป็นโมเดลที่สอดคล้องกับหลักการของฐานข้อมูล โดยที่ผู้ใช้ไม่จำเป็นต้องรับรู้ในรายละเอียดของการเก็บข้อมูลภายใน

ระบบจัดการฐานข้อมูล

ระบบจัดการฐานข้อมูล คือ ระบบที่มีหน้าที่ควบคุม ดูแลการสร้างและเรียกใช้ฐานข้อมูล โดยที่ผู้ใช้ไม่จำเป็นต้องทราบเกี่ยวกับรายละเอียดในโครงสร้างของฐานข้อมูล ระบบจัดการฐานข้อมูลมีหน้าที่ ดังนี้

1. การสร้างและแก้ไขโครงสร้างของฐานข้อมูล รวมทั้งบรรจุข้อมูลในการทำงาน
2. การเข้าถึงเนื้อหาในฐานข้อมูล เพื่อการแก้ไขหรือเรียกดูข้อมูลได้พร้อมกัน
3. กำหนดคำจำกัดความ และข้อบังคับเกี่ยวกับความต้องการในการรักษาความปลอดภัย การให้สิทธิ์ผู้ใช้ และการป้องกันความเสียหาย ตลอดจนการทำสำรองข้อมูล
4. รวบรวมสถิติที่เกี่ยวข้องกับการใช้ฐานข้อมูล
5. เฝ้าคุมระบบ และเปลี่ยนแปลงแก้ไขระบบ เพื่อให้ทำงานได้ดีขึ้น
6. มีพจนานุกรมข้อมูล เกี่ยวกับฐานข้อมูล เช่น โครงสร้าง ชนิด รูปแบบ และข้อจำกัดของข้อมูล เป็นต้น

สถาปัตยกรรมมาตรฐานของฐานข้อมูล

ได้มีการกำหนดมาตรฐานขึ้นโดย ANSI (American National Standards Institute) ในปี 1975 เรียกว่า สถาปัตยกรรมสามระดับ (Three_Level Architecture) การอธิบายรายละเอียดของระดับต่าง ๆ จะไม่เกี่ยวข้องกับข้อมูลที่เก็บในฐานข้อมูล แต่จะอธิบายโครงสร้างของฐานข้อมูลโดยรวม เพื่อให้สามารถแยกได้ ก็เนื่องจากการที่ผู้ใช้แต่ละคนอาจมองข้อมูลต่างกัน ดังนั้น ถ้ามีการเปลี่ยนแปลงโครงสร้างภายในก็ไม่ควรให้กระทบกับมุมมองของผู้ใช้ สถาปัตยกรรมสามระดับประกอบด้วย

1. สคีมาระดับภายนอก (External Schemas Level) จะเป็นระดับที่ใกล้กับผู้ใช้มากที่สุด เป็นสิ่งที่ผู้ใช้คิดเกี่ยวกับข้อมูลจะอธิบายถึงวิว (View) ที่ผู้ใช้สนใจ ข้อมูลที่เก็บจริงอาจมีมากกว่าที่ผู้ใช้ต้องการ และข้อมูลตัวเดียวกันผู้ใช้อาจมองไม่เหมือนกัน เช่น ข้อมูลที่เกี่ยวกับวันที่ ผู้ใช้คนหนึ่งอาจมองเป็น วัน เดือน ปี แต่อีกคนหนึ่งอาจจะมองเป็น เดือน วัน ปี นอกจากนั้นสิ่งที่ผู้ใช้มองเห็นอาจไม่ได้เก็บจริงในฐานข้อมูลแต่ได้จากการคำนวณออกมา ส่วนนี้เองจะถูกแปลโดยระบบจัดการฐานข้อมูลและเก็บไว้ในพจนานุกรมข้อมูล

2. สคีมาระดับเชิงมโนภาพ (Conceptual Schemas Level) จะเป็นตัวที่ใช้เชื่อมระหว่างสคีมาระดับภายนอกกับสคีมาระดับภายใน อธิบายฐานข้อมูลในรายละเอียดโดยภาพรวมทั้งหมด เพื่อเชื่อมกับสิ่งที่ผู้ใช้มอง รูปแบบข้อมูลที่มีความสัมพันธ์ เงื่อนไขต่างๆ รวมทั้งความมั่นคงและความถูกต้องของข้อมูลจะถูกเก็บไว้ด้วย แต่จะไม่ลงลึกถึงการเก็บในระบบ

3. สคีมาระดับภายใน (Internal Schemas Level) เป็นระดับการเก็บข้อมูลจริง อธิบายฐานข้อมูลในการเก็บทางกายภาพจริง ๆ โดยที่มุมมองของระบบจัดการฐานข้อมูลใช้โครงสร้างข้อมูล และการจัดองค์กรแฟ้มข้อมูล โดยจะทำงานร่วมกับระบบปฏิบัติการในการเก็บข้อมูลไว้ที่หน่วยเก็บสำรอง

นอกจากนั้น สถาปัตยกรรมทั้งสามดังกล่าวยังมีความเป็นอิสระของข้อมูล คือ สคีมาระดับภายนอกจะมีรูปแบบที่คงตัวไม่เปลี่ยนแปลง แม้ว่าจะมีการเปลี่ยนสคีมาระดับเชิงมโนภาพ เช่น มีการเพิ่มรูปแบบข้อมูลใหม่ หรือมีความสัมพันธ์ใหม่ ๆ เพิ่มขึ้น เรียกว่า ความไม่พึ่งพิงทางตรรก (Logical Data Independence) และสคีมาระดับเชิงมโนภาพก็เช่นกัน จะมีรูปแบบที่คงตัวแม้จะมีการเปลี่ยนแปลงทางกายภาพ เช่น เปลี่ยนวิธีการเข้าถึงข้อมูล หรือลำดับของข้อมูลที่เก็บอยู่จริงไม่เปลี่ยนไป เรียกว่า ความไม่พึ่งพิงทางกายภาพ (Physical Data Independence) อย่างไรก็ตาม โดยทั่วไปแล้วระบบจัดการฐานข้อมูลไม่ได้แบ่งสถาปัตยกรรม

ทั้ง 3 ระดับออกจากกันโดยเด็ดขาด ในการทำงานส่วนมาก จะให้ผู้ใช้ระบุถึงระบบภายนอกเท่านั้น แล้วระบบจัดการฐานข้อมูลจะเป็นตัวเปลี่ยนการร้องขอ หรือการระบุนั้น ๆ ให้เป็นระดับมโนภาพ แล้วจึงประมวลผลข้อมูลออกมาให้

วงจรการพัฒนาฐานข้อมูล

ช่วงเวลาระหว่างการออกแบบ สร้าง และใช้ จนถูกแทนที่ด้วยระบบใหม่ที่สมบูรณ์มักจะต้องใช้เวลานาน ประกอบด้วยขั้นตอนต่าง ๆ ดังนี้

1. ข้อกำหนดระบบ เป็นการศึกษาความต้องการของผู้ใช้ระบบ พิจารณาทำความเข้าใจกับระบบงานปัจจุบัน โดยศึกษาขั้นตอนการปฏิบัติงาน รวมถึงซอฟต์แวร์ที่ใช้ เพื่อกำหนดรูปแบบรายงานต่าง ๆ แฟ้มข้อมูลหรือตารางข้อมูล

2. การออกแบบ จะประกอบไปด้วยการออกแบบทางตรรก และการออกแบบทางกายภาพ การออกแบบที่ดีจะต้องเป็นระบบที่ง่ายต่อการเปลี่ยนแปลงหรือขยายตัวของระบบในอนาคต โดยจะทำเป็นโมเดลที่วาดเป็นผังไดอะแกรมของระบบจากหลาย ๆ มุมมอง หรือทำต้นแบบ

3. การติดตั้งระบบ เป็นกระบวนการสร้างข้อกำหนดของฐานข้อมูล ลงในฐานข้อมูลผ่านระบบจัดการฐานข้อมูลที่เลือก

4. การบรรจุหรือการแปลงข้อมูล

5. การทดสอบและตรวจสอบความถูกต้อง

6. การปฏิบัติการ เป็นขั้นตอนการปฏิบัติการตามที่ได้กำหนดไว้ รวมทั้งการสำรองข้อมูลและฟื้นคืนสภาพข้อมูลได้ เมื่อมีข้อขัดข้องเกิดขึ้น

7. การติดตามและการบำรุงรักษา ดูแล การเจริญเติบโตและขยายตัวที่จะเกิดขึ้นกับข้อมูลและซอฟต์แวร์

ขั้นตอนการออกแบบ ติดตั้ง และนำข้อมูลเข้าเป็นจุดสำคัญที่สุดในวงจร ขั้นตอนต่างๆ เหล่านี้ บางครั้งก็แยกกันโดยเด็ดขาด บางครั้งก็อาจจะทำคู่ขนานได้

กระบวนการออกแบบฐานข้อมูล

การออกแบบฐานข้อมูล จะประกอบด้วยขั้นตอนต่าง ๆ คือ

1. รวบรวมข้อมูล เพื่อกำหนดเนื้อหารายละเอียดข้อมูล จัดเตรียมโครงสร้างของฐานข้อมูล

2. ทำความเข้าใจกับข้อมูลและการประยุกต์ใช้งาน เพื่อวาดรูปแบบระบบฐานข้อมูล แสดงการแยกแยะ จัดข้อมูลไว้หลาย ๆ ตาราง จัดกลุ่มข้อมูลที่จะเชื่อมโยงความสัมพันธ์

3. จัดรูปแบบของข้อมูลขั้นต้น เพื่อวางโครงร่างของระบบคร่าว ๆ ว่า ภายในระบบจะมีการส่งผ่านข้อมูล และแสดงผลลัพธ์ในลักษณะใด

4. สร้างระบบที่ต้องการใช้งาน หลังจากได้รูปแบบระบบขั้นต้นแล้ว จะต้องศึกษาอย่างละเอียดว่าจะจัดการอย่างไรตามลำดับขั้นตอน เพื่อเป็นแนวทางสำหรับผู้สร้างระบบ

ความมั่นคงของฐานข้อมูล

ฐานข้อมูล เป็นที่เก็บข้อมูลที่มีค่าจำนวนมาก ดังนั้นจึงต้องมีการรักษาความมั่นคงของข้อมูลเหล่านั้น ซึ่งในการออกแบบ ผู้ออกแบบจะต้องพิจารณาในหัวข้อต่อไปนี้

1. วัตถุประสงค์ของการรักษาความมั่นคงของข้อมูล

1.1 ป้องกันและขัดขวางไม่ให้ผู้ไม่มีสิทธิเข้าไปดึง แก้ไข เปลี่ยนแปลงข้อมูล หรือปรับโครงสร้างของฐานข้อมูล

1.2 กีดขวางไม่ให้ผู้มีสิทธิขยายวงในการสืบค้น หรือทำความเข้าใจกับข้อมูลในส่วนที่เขาไม่มีสิทธิ์

1.3 ค่าใช้จ่ายในการเสี่ยงของผู้ที่บุกรุกสูงกว่าประโยชน์ที่จะได้รับ

2. สาเหตุของการบุกรุกมีได้ทั้งอุบัติเหตุและโดยจงใจ ผู้บุกรุกนั้นเป็นได้ทั้งบุคคลภายในและภายนอกหน่วยงาน การบุกรุกแบ่งได้ ดังนี้

2.1 การบุกรุกโดยอุบัติเหตุ

(1) ในการกำหนดสิทธิการใช้ข้อมูลในฐานของข้อจำกัดระบบจัดการฐานข้อมูลหรือระบบปฏิบัติการ ทำให้ผู้ใช้ทั่วไปมีสิทธิดึงข้อมูลได้โดยไม่ตั้งใจ

(2) การส่งข่าวสารถึงผู้ใช้ผิดคน ทำให้ผู้ใช้ไม่มีสิทธิรู้ข้อมูล

(3) ความผิดพลาดของระบบสื่อสารมีผลทำให้ผู้ใช้เข้าไปในเครือข่ายของผู้อื่น

(4) ความผิดพลาดของระบบปฏิบัติการทำให้เกิดการบันทึกซ้ำ หรือส่งแฟ้มข้อมูลผิดให้กับผู้ใช้คนอื่น

2.2 การบุกรุกโดยจงใจ

(1) การลอบต่อสายในการสื่อสาร

(2) การใช้สัญญาณอิเลคโทรนิคดึงสัญญาณจากเครื่องอื่น

(3) แอบดูข้อมูลจากหน้าจอ หรือสิ่งพิมพ์ของผู้ใช้อื่นที่มีสิทธิ

(4) การเข้าดึงข้อมูลจากผู้มีสิทธิในระดับต่ำกว่า

(5) การขโมยสื่อที่ใช้บันทึกข้อมูล เช่น เทป จานแม่เหล็กที่บันทึกข้อมูลไว้ ออกจากห้องคอมพิวเตอร์

3. การรักษาความมั่นคง แบ่งได้ 2 วิธี คือ

3.1 การรักษาความมั่นคงทางกายภาพได้แก่ การติดตั้งระบบรักษาความมั่นคงต่าง ๆ ก่อนที่ผู้บุกรุกจะเข้ามาเปิดเครื่องได้ เช่น การมียามรักษาการณ์ มีกุญแจล็อกเครื่องให้เซ็นชื่อหรือพิมพ์ลายนิ้วมือ ซึ่งสิ่งเหล่านี้จะอยู่นอกเหนือขอบเขตของผู้ออกแบบ แต่ผู้ออกแบบก็ควรแนะนำไว้

3.2 การรักษาความมั่นคงทางซอฟต์แวร์ โดยการสร้างระบบนี้ไว้ในโปรแกรมดังต่อไปนี้

(1) การให้อำนาจ คือ การตรวจสอบหลักฐานผู้ใช้ว่า เป็นตัวจริงที่มีสิทธิมาร้องขอข้อมูลหรือไม่ จะติดตั้งการรักษาความมั่นคงไว้ที่ระดับระบบปฏิบัติงาน วิธีที่นิยมมากที่สุด คือ ให้ใส่รหัสผ่าน แต่ถึงอย่างไรก็ตามก็ไม่ปลอดภัยนัก บางแห่งอาจจะมีการเพิ่มเติมให้ใช้บัตร หรือกุญแจเฉพาะเมื่อจะเข้าสู่ระบบ ในระบบจัดการฐานข้อมูล ส่วนมากจะมีระบบรักษาความมั่นคงนี้ให้สำหรับผู้ใช้แต่ละคนเพื่อกำหนดผู้มีสิทธิใช้ข้อมูลของตนได้เอง เรียกว่าภาษาในการให้อำนาจ เช่น ในเอสคิวแอล มีคำสั่ง Grant เป็นต้น ซึ่งเป็นการให้อำนาจในการกระทำกับตารางนั้นกับผู้ใช้อื่น ๆ

(2) การควบคุมการเข้าถึง คือ การทำให้แน่ใจว่าข้อมูลหรือสิ่งต่างๆ ในระบบจะถูกเข้าถึงได้เฉพาะทิศทางที่ต้องผ่านการตรวจสอบการให้อำนาจเท่านั้น ซึ่งวิธีที่จะทำได้เมื่อมีการติดตั้งการใช้อำนาจแล้วเท่านั้น เราอาจทำโดยสร้างเป็นตารางการเข้าถึงเพื่อให้ง่ายต่อการควบคุมโดยแต่ละสดมภ์แทนเป้าหมายในระบบ เช่น ชื่อตาราง หรือวิว ส่วนของแถว ต่าง ๆ แทนรหัสประจำตัวของผู้ใช้ หรือกลุ่มผู้ใช้ และคำในแต่ละช่องที่สัมพันธ์กันก็คือ อำนาจในการเข้าถึงที่กำหนดให้ผู้ใช้ สำหรับตาราง หรือ วิวนั้น ๆ ได้แก่ อำนาจการลบ เพิ่ม เปลี่ยนแปลง อ่านข้อมูล

(3) วิว การสร้างวิวในระบบจัดการฐานข้อมูล โดยทั่วไปจะมีคำสั่งที่ใช้ในการสร้างวิวจากตารางที่มีในระบบ เพื่อให้ได้ข้อมูลเฉพาะในส่วนที่ผู้ใช้จะมีสิทธิโดยผู้จัดการฐานข้อมูลจะเป็นคนทำและให้สิทธิในการเข้าถึงวิวนั้น ๆ ให้แก่ผู้ใช้อื่น ๆ

โมเดลข้อมูลเชิงตรรก (Fleming, 1989)

เป็นวิธีการแสดงความต้องการสารสนเทศในระบบเชิงธุรกิจให้เป็นแผนภาพ ในขั้นตอนการวิเคราะห์และออกแบบระบบฐานข้อมูล เน้นตัวข้อมูลที่มีอยู่จริง โดยไม่คำนึงถึงรายละเอียดในการติดตั้ง ความต้องการพิเศษอื่นในด้านการใช้งาน และความเร็วในการสืบค้นข้อมูล หรืออาจกล่าวอีกนัยหนึ่งว่า โมเดลข้อมูลเชิงตรรกเป็นการสร้างโครงสร้างวิวของผู้ใช้ (Skeletal User View) จะแสดงข้อมูลในขอบเขตที่ผู้ออกแบบสนใจโดยมีสิ่งที่ต้องกำหนดเป็นพื้นฐานได้แก่เอนทิตี, รีเลชันชิป และแอตตริบิว

วิวของผู้ใช้ คือ โมเดลหรือการแทนความต้องการในด้านผู้ใช้แต่ละคนเทียบเท่ากับการปฏิบัติงานหนึ่งอย่าง เช่น การพิมพ์ใบส่งสินค้าให้ลูกค้ารายหนึ่ง และในที่สุดมีการรวมวิวของผู้ใช้เข้าด้วยกัน เรียกว่า โมเดลข้อมูลเชิงตรรกแบบเบ็ดเสร็จ (Integrated Logical Data Model)

เอนทิตี คือสิ่งที่มีอยู่จริง จับต้องได้ หรือเป็นจินตนาภาพที่แสดงความเป็นหนึ่งเดียวเมื่อกล่าวถึง วิว ทุกคนเข้าใจตรงกัน เช่น สินค้า, วัตถุดิบ, แผนกต่าง ๆ, การสั่งซื้อ

รีเลชันชิป คือ ความสัมพันธ์ซึ่งเป็นลักษณะการเกี่ยวพันกันระหว่างเอนทิตีหนึ่งกับตัวมันเอง หรือเอนทิตีอื่น อาจเป็นความสัมพันธ์ที่มากกว่า 2 เอนทิตีก็ได้ เช่น แผนกจัดซื้อทำการสั่งซื้อสินค้าหรือวัตถุดิบ

แอตตริบิว คือ กลุ่มของค่าความจริงใด ๆ ที่เป็นรายละเอียดของเอนทิตีซึ่งแสดงลักษณะและคุณสมบัติของเอนทิตี ทำให้เข้าใจเอนทิตีได้ลึกซึ้งยิ่งขึ้น และเป็นสิ่งที่ไม่สามารถแตกย่อยลงไปได้อีก โดยไม่เสียความหมายไป เช่น รหัสสินค้า, สถานที่เก็บ, ชื่อสินค้า, ราคา นอกจากนั้นยังมีการระบุด้วยว่าแอตตริบิวใดเป็นคีย์ กำหนดกฎข้อบังคับต่าง ๆ ของเอนทิตีและรีเลชันชิป

คุณลักษณะของโมเดลข้อมูลเชิงตรรก

1. แสดงได้ด้วยแผนภาพ (Graphical Diagrams) ไม่ว่าจะเป็นเทคนิคโมเดลข้อมูลแบบใดก็ตาม จะมีภาษาและรูปแบบทางกราฟฟิคโดยเฉพาะ เพื่อใช้แสดงรายละเอียดข้อมูลทั้งกลุ่มใหญ่และรายละเอียดส่วนย่อย ซึ่งทำให้ง่ายต่อการแปลความ เช่น ใช้วงกลมหรือสี่เหลี่ยมแทนเอนทิตี ใช้เส้นโค้ง หรือเส้นตรงแทนรีเลชันชิป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

2. แสดงชัดเจนถึงความหมายของข้อมูล (Explicit Representation of Semantic) มีทางเลือกในการแสดงความหมายของข้อมูล เราอาจใช้สัญญลักษณ์ที่ต่างกันจำนวนมากบ้างน้อยบ้างเพื่อแสดง แต่จุดสำคัญคือ แผนภาพที่ออกมาควรง่าย ไม่ซับซ้อนและเห็นความหมายของข้อมูลชัดเจน โดยเฉพาะอย่างยิ่งสัญญลักษณ์หนึ่ง ๆ มีหลายความหมาย

3. แสดงรายละเอียดในระดับที่เหมาะสม (Appropriate Level of Detail) กล่าวคือ โมเดลระดับตรรกจะมีรายละเอียดที่เพียงพอที่จะชี้จุดที่ทำให้เกิดความแตกต่างระหว่างชนิดของรีเลชันชิปและข้อบังคับต่าง ๆ แต่จะน้อยกว่าโมเดลทางกายภาพ

4. ไม่พึ่งพิงกับระบบจัดการฐานข้อมูลแบบใดแบบหนึ่ง (DBMS Independence) โมเดลที่ได้จากการออกแบบใช้ได้กับฐานข้อมูลหลายประเภท ได้แก่ เชิงความสัมพันธ์, เชิงลำดับขั้น, และเชิงเครือข่าย

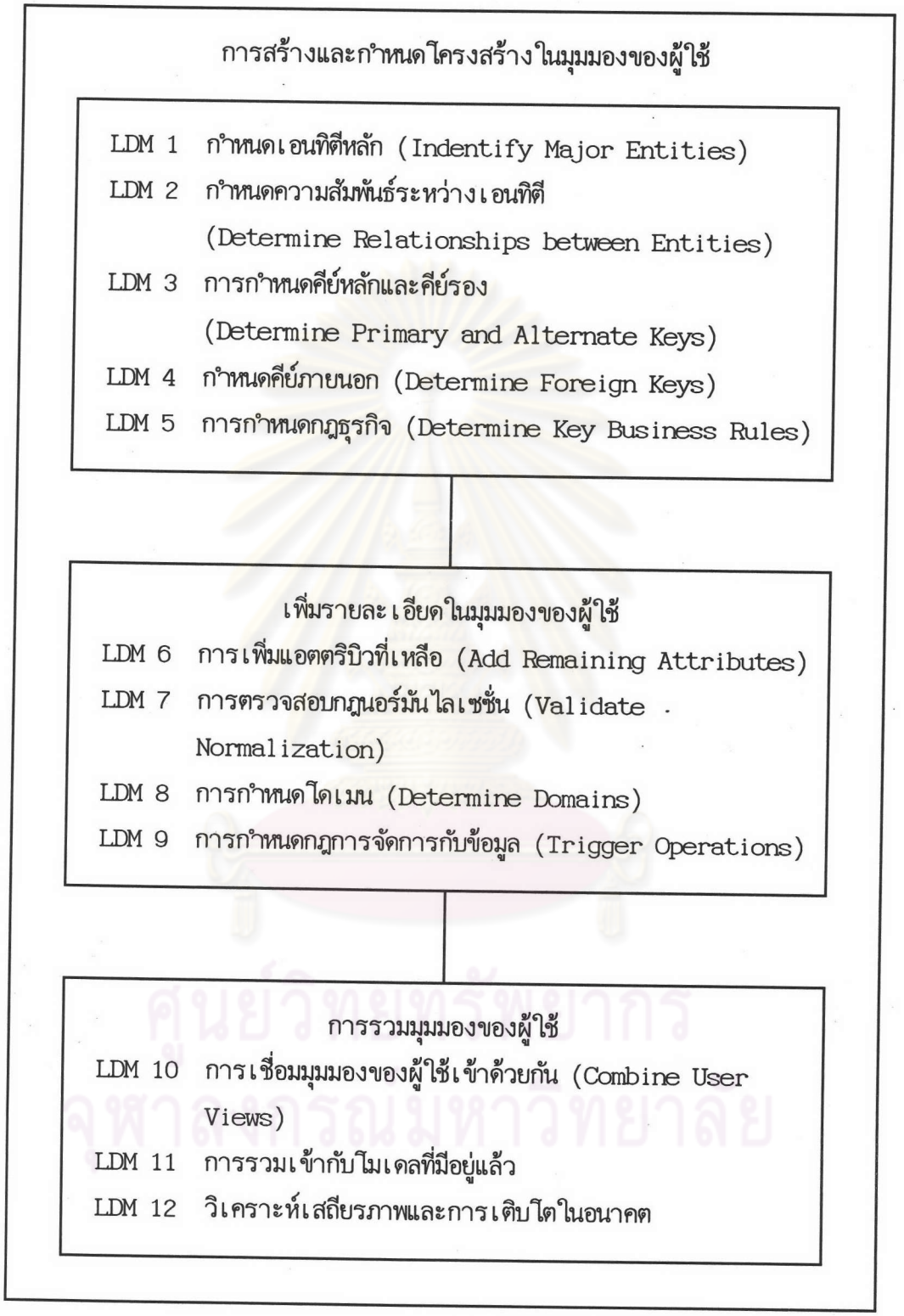
5. ง่ายต่อการศึกษาและใช้งาน ในที่นี้จะต้องง่ายเพียงพอสำหรับผู้ใช้ทุกประเภท จะทำความเข้าใจและนำไปใช้ได้

ขั้นตอนการออกแบบโมเดลข้อมูลเชิงตรรก

ในการออกโมเดลข้อมูลเชิงตรรก มีด้วยกันหลายขั้นตอน แต่ใน 5 ขั้นตอนแรกจะเป็นการออกแบบทางด้านโครงสร้างพื้นฐานของโมเดล ได้แก่ พวกเอนทิตี รีเลชันชิป คีย์หลัก คีย์สำรอง คีย์ภายนอก กฎเกณฑ์พื้นฐาน จากนั้นจึงเริ่มเพิ่มรายละเอียดในระดับผู้ใช้วิว และรวบรวมรายละเอียดเหล่านั้นเข้าด้วยกัน จึงได้เป็นโมเดลข้อมูลเชิงตรรกที่สมบูรณ์ สรุปดังรูปที่ 3.1

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ไมเดลข้อมูลเชิงตรรก



รูปที่ 3.1 ขั้นตอนการออกแบบไมเดลข้อมูลเชิงตรรก

ขั้นตอนที่ 1 การกำหนด เอนทิตี หลัก

โดยกำหนดชื่อและความหมายลงในพจนานุกรมข้อมูล และเขียนลงโมเดลข้อมูลด้วยการตั้งชื่อ ซึ่งไม่ควรเกิน 20 ตัวอักษร

การกำหนดเอนทิตี ต้องอาศัยความร่วมมือของผู้ที่เข้าใจระบบที่เราออกแบบ เพื่อคัดเลือกสิ่งที่ถูกต้อง มีความสำคัญและเหมาะสมที่สุดมาเป็นเอนทิตี วิธีการอย่างคร่าว ๆ ก็คือให้การพิจารณาข้อมูลทั้งหมดที่มี และจัดกลุ่มของข้อมูลโดยดูจากค่าและความหมาย ถ้าสามารถรวมกลุ่มกันได้ก็ให้รวมเข้าไว้ในเอนทิตีเดียวกัน นอกจากนั้นยังสามารถแยกเอนทิตีย่อยลงไปได้อีก โดยเรียกว่า ซับไทป์ (Subtype) ของอีกเอนทิตีหนึ่ง เช่น ถ้าบอกว่า A เป็นซับไทป์ของ B จะสังเกตได้ดังนี้

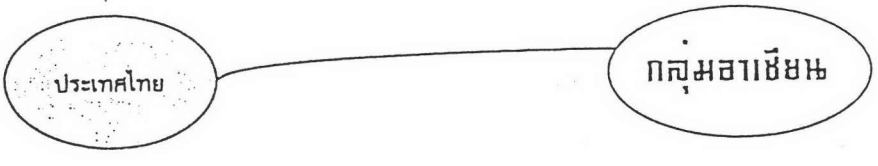
1. เอนทิตี A และเอนทิตี B ต้องแทนสิ่งเดียวกัน
2. เอนทิตี A จะประกอบด้วยสิ่งต่าง ๆ ใน B และส่วนที่เป็นของตัวมันเองด้วย
3. ทุก ๆ เอนทิตี A จะปรากฏเพียงหนึ่งใน B (1 : 1)

ขั้นตอนที่ 2 การกำหนดความสัมพันธ์ระหว่างเอนทิตี

ส่วนประกอบที่สำคัญของโมเดลเชิงตรรก คือ เอนทิตีและรีเลชันชิปจะทำได้โดยกำหนดชื่อ ความหมาย ทิศทาง และขนาดอัตราส่วนที่เกิดรีเลชันชิปนั้น ๆ พร้อมทั้งบันทึกลงในพจนานุกรมข้อมูลด้วย สำหรับชื่อก็ไม่ควรเกิน 20 ตัวอักษรเช่นเดียวกับชื่อของเอนทิตี สำหรับความหมาย ถ้าเป็นภาษาอังกฤษก็ให้ใช้คำกิริยาในรูปแบบปัจจุบัน (Present Tense)

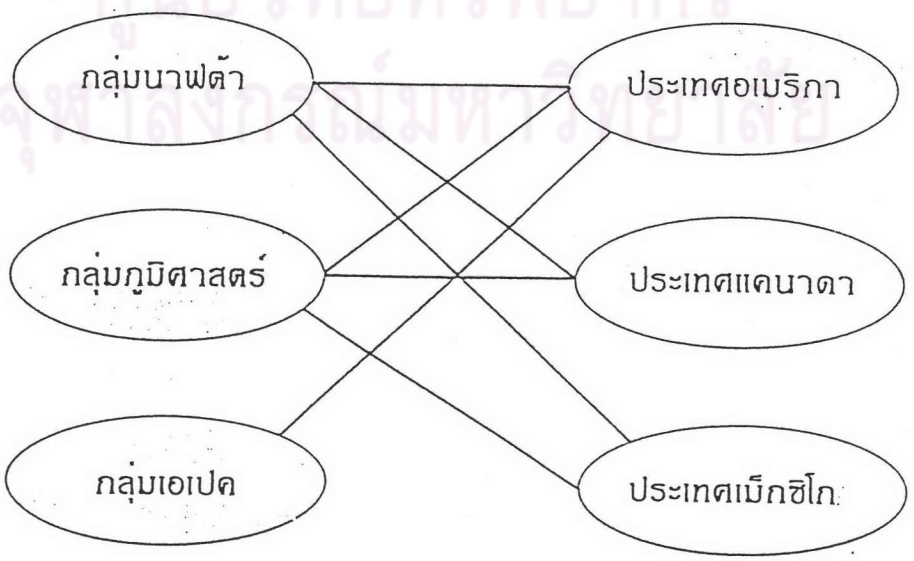
การกำหนดความสัมพันธ์ระหว่างเอนทิตีนั้น ต้องคำนึงถึงทิศทางของการกำหนดความความสัมพันธ์ด้วยว่า เป็นความสัมพันธ์ขึ้นไปจากเอนทิตีใด ไปเอนทิตีใด โดยใช้ลูกศรแทนทิศทางของความสัมพันธ์ และจะต้องมีเอนทิตีหนึ่งเป็นเอนทิตีหลัก เราเรียกเอนทิตีหลักนี้ว่าเอนทิตีพ่อแม่ (Parent Entity) และเรียกเอนทิตีที่ตามมาว่า เอนทิตีลูก (Child Entity) และหัวลูกศรจะชี้ไปทางเอนทิตีลูกเสมอ โดยวิธีนี้เราสามารถแบ่งความสัมพันธ์ออกเป็น 3 ประเภท โดยขึ้นอยู่กับอัตราส่วนของระเบียน ในเอนทิตีพ่อแม่และเอนทิตีลูก คือ

1. ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (one-to-one, 1 : 1 Relationaship) เป็นความสัมพันธ์ที่เมื่อพิจารณาระเบียนใดระเบียนหนึ่งในเอนทิตีหนึ่ง จะมีความสัมพันธ์กับระเบียนในอีกเอนทิตีหนึ่งเพียงระเบียนเดียวเท่านั้น



2. ความสัมพันธ์แบบหนึ่งต่อหลาย (one-to-many, 1 : N Relationship) ความสัมพันธ์จะเกิดขึ้นได้ตั้งแต่ไม่มีเลยจนถึงหลาย ๆ ครั้ง โดยแต่ละระเบียนของเอนทิตีลูกจะสัมพันธ์กับเอนทิตีพ่อแม่ได้หนึ่งระเบียนเท่านั้น แต่หนึ่งระเบียนในเอนทิตีพ่อแม่สัมพันธ์กับเอนทิตีลูกได้หลายระเบียน

3. ความสัมพันธ์แบบหลายต่อหลาย (many-to-many, M:N Relationship) เป็นความสัมพันธ์ที่เกิดขึ้นได้ตั้งแต่ไม่มีเลยจนถึงหลาย ๆ ครั้งในสองทิศทาง กล่าวคือ ระเบียนใดระเบียนหนึ่งในเอนทิตีลูก จะมีความสัมพันธ์กับระเบียนในเอนทิตีพ่อแม่ได้หลายระเบียน และในทางกลับกันระเบียนใดระเบียนหนึ่งในเอนทิตีพ่อแม่ ก็สัมพันธ์กับระเบียนในเอนทิตีลูกได้หลายระเบียนเช่นกัน



หลังจากที่เราสามารถแบ่งกลุ่มความสัมพันธ์ ระหว่างเอนทิตีได้เรียบร้อยแล้ว จะพบว่าความสัมพันธ์แบบหลายต่อหลายเป็นสิ่งที่เราต้องสนใจมากที่สุด เพราะเป็นตัวทำให้การสร้างฐานข้อมูลเชิงตรรกมีความยุ่งยากซับซ้อนมากขึ้น โดยปกติฐานข้อมูลโดยทั่วไปจะไม่สามารถจัดการกับความสัมพันธ์แบบกลุ่มนี้ได้ ดังนั้น เราจึงควรกระจายความสัมพันธ์แบบหลายต่อหลายมาเป็นความสัมพันธ์แบบหนึ่งต่อหลายสองความสัมพันธ์ โดยกำหนดเอนทิตีใหม่ขึ้นมาอีกเอนทิตีหนึ่งตัวให้มีความสัมพันธ์กับเอนทิตีเดิมแบบหนึ่งต่อหลาย

จากลักษณะของความสัมพันธ์แบบหนึ่งต่อหนึ่ง ถ้าเอนทิตีลูกหลาย ๆ เอนทิตีเป็นส่วนประกอบของเอนทิตีพ่อแม่เพียงเอนทิตีเดียว และเอนทิตีเหล่านั้นมีความหมายในเรื่องเดียวกัน สามารถรวมความสัมพันธ์เหล่านั้นเป็นความสัมพันธ์เดียวกัน เรียกว่า ซุปเปอร์ไทป์และซับไทป์ ซึ่งเป็นความสัมพันธ์อีกแบบหนึ่ง ที่แสดงถึงความสัมพันธ์ระหว่างเอนทิตีพ่อแม่กับชุดของเอนทิตีลูกที่มีความหมายในเรื่องเดียวกัน

ขั้นตอนที่ 3 การกำหนดคีย์หลักและคีย์รอง

หลังจากที่ได้กำหนดเอนทิตีต่าง ๆ แล้ว ขั้นตอนต่อไปของการสร้างโมเดลเชิงตรรก คือ การเพิ่มข้อมูลที่เรียกว่า แอตตริบิวลงในทุก ๆ เอนทิตี

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

ขั้นตอนที่ 4 การกำหนดคีย์ภายนอก

เมื่อกำหนดคีย์หลักและคีย์รองได้แล้ว ให้กำหนดคีย์ภายนอกสำหรับเอนทิตีที่มีความสัมพันธ์กัน คีย์ภายนอก คือ แอตตริบิวในเอนทิตีระดับลูกที่แทนคีย์หลักของเอนทิตีพ่อแม่เพื่อใช้ในการอ้างถึงระเบียนในเอนทิตีพ่อแม่ และแสดงถึงความสัมพันธ์ระหว่างเอนทิตีต่าง ๆ ซึ่งอาจให้คีย์หลักเป็นคีย์ภายนอกด้วยก็ได้ แอตตริบิวที่เป็นคีย์ภายนอกมีความสำคัญ ดังนี้

ความสำคัญของคีย์ภายนอก คือ

1. ทำให้เกิดกฎธุรกิจ (Business Rule) ในเอนทิตีต่าง ๆ

2. แม้จะดูเหมือนว่าการกำหนดคีย์ภายนอกเป็นการทำให้ซ้ำซ้อน แต่จริง ๆ แล้ว คีย์ภายนอกจะเป็นตัวช่วยในการออกแบบ เพื่อให้เราสามารถตรวจสอบได้ว่า เอนทิตีไหนเป็นเอนทิตีลูกและอันไหนเป็นเอนทิตีพ่อแม่

3. ทำให้ขั้นตอนการออกแบบฐานข้อมูลง่ายขึ้น

ข้อควรระวัง

ไม่ควรจะกำหนดคีย์รองให้เป็นคีย์ภายนอก เพราะจะทำให้ซับซ้อนและอาจมีค่าเป็นนัล (Null) ได้ ซึ่งคีย์รองไม่ควรเป็นเช่นนั้น

ขั้นตอนที่ 5 การกำหนดกฎการจัดการกับข้อมูล

เป็นขั้นตอนที่ทำขึ้นเพื่อความสมบูรณ์ของข้อมูล และความถูกต้องตรงกันของค่าข้อมูล กฎการจัดการกับข้อมูลแบ่งได้เป็น 3 ประเภท คือ

1. กฎธุรกิจของคีย์ (Key Business Rules) เป็นการกำหนดกฎเพื่อความสมบูรณ์ของความสัมพันธ์ กฎต่าง ๆ จะมีผลจากการจัดการต่าง ๆ บนความสัมพันธ์ อันได้แก่ การเพิ่ม การลบ และการแก้ไขข้อมูล เช่น กำหนดผลที่จะเกิดขึ้นจากการเปลี่ยนค่าของคีย์หลัก

2. โดเมน (Domain) คือ การกำหนดกฎเพื่อความสมบูรณ์ถูกต้องของแอตตริบิว หมายถึง การกำหนดข้อบังคับและค่าที่เป็นไปได้สำหรับแอตตริบิวทั้งที่เป็นคีย์และไม่ใช่คีย์

3. กฎการจัดการ (Triggering Operation) เป็นการกำหนดผลกระทบ จากการเพิ่ม, ลบ, แก้ไข หรือดึงข้อมูลที่เกิดกับเอนทิตีอื่น หรือ แอตตริบิวอื่นภายในเอนทิตีเดียวกัน

สำหรับในขั้นตอนนี้จะพิจารณาเฉพาะกฎธุรกิจของคีย์ก่อน ซึ่งการกำหนดกฎในการเพิ่มข้อมูลนั้นมักจะเป็นผลเมื่อเราทำการเพิ่มข้อมูลในเอนทิตีลูก สำหรับกฎในการแก้ไขและลบข้อมูลนั้นจะเป็นผลเมื่อกระทำกับเอนทิตีพ่อแม่

ขั้นตอนที่ 6 การเพิ่มแอตตริบิวที่ไม่ใช่คีย์ เพื่อเป็นรายละเอียดของเอนทิตี

ให้ทำการกำหนดแอตตริบิวอื่น ๆ ในเอนทิตี โดยการตั้งชื่อและเพิ่มในโมเดลข้อมูลเชิงตรรกพร้อมทั้งบันทึกในพจนานุกรมข้อมูลด้วย

จากขั้นตอนการทำโมเดลข้อมูลเชิงตรรกที่ผ่านมาแอตตริบิวที่เรารู้จัก ก็คือ คีย์หลัก คีย์รอง และคีย์ภายนอก ขั้นนี้ก็จะเป็นการกำหนดแอตตริบิวอื่น ๆ ที่ไม่ใช่คีย์เพิ่มเข้าไป โดยที่แอตตริบิวแต่ละตัวที่จะเพิ่มนั้นจะต้องขึ้นกับทั้งหมดของคีย์ของเอนทิตีนั้น ไม่ใช่ขึ้นกับบางส่วนของ

คีย์ โดยต้องขยายในเอนทิตีพ่อแม่ ไม่ใช่ขยายในเอนทิตีที่มีคีย์นั้นเป็นคีย์ภายนอก นอกจากนั้น ถ้าเกิดแอตตริบิวดังกล่าวขึ้นกับคีย์หลักทั้งหมด แต่มีค่ามากกว่าหนึ่งค่า (Multivalued) ให้แตกออกเป็นอีกเอนทิตีที่มีความสัมพันธ์กับเอนทิตีเดิมแบบหนึ่งต่อหลาย (1:N) พร้อมทั้งตั้งชื่อและระบุรายละเอียดอื่น ๆ ใส่ในโมเดลข้อมูลเชิงตรรกและใส่ลงในพจนานุกรมข้อมูล

สำหรับแอตตริบิวบางตัว เมื่อพิจารณาแล้วพบว่าจะขยายความสัมพันธ์ ก็ให้แตกความสัมพันธ์นั้นออกเป็นเอนทิตีใหม่ ที่มีความสัมพันธ์กับเอนทิตีเดิมเป็นแบบหนึ่งต่อหลาย (1:N) สิ่งที่ควรหลีกเลี่ยง คือ การใช้รหัสแทนแอตตริบิว แต่ถ้าจำเป็นก็ให้กำหนดรหัสเป็นอิสระ คือ ไม่รวมความหมายมากกว่าหนึ่งอย่างไว้ในรหัสเดียวกัน ในกรณีที่แอตตริบิวสามารถหาได้จากสูตรหรือคำนวณได้จากแอตตริบิวอื่น จะเรียกว่า ดิไรฟว์แอตตริบิว (Derived Attribute) ให้ระบุตัวดี (d) ในโมเดลข้อมูลด้วย ในกรณีที่ เป็นตัวบ่งชี้ (Flag) ให้ใส่ไว้เพื่อระบุซับไทป์-ซูเปอร์ไทป์ให้ระบุตัวเอส (s) ในโมเดลเช่นกัน

นอกจากการกำหนดแอตตริบิวแล้ว ยังให้พิจารณาเชื่อมเอนทิตีที่มีลักษณะดังต่อไปนี้เข้าด้วยกัน คือ

1. เอนทิตีที่มีคีย์หลักเหมือนกัน และให้ความหมายอันเดียวกัน
2. เอนทิตีที่เป็นซับไทป์ และมีแอตตริบิวทุกตัวเหมือนกัน
3. เอนทิตีที่ต่างก็เป็นซับไทป์-ซูเปอร์ไทป์ ซึ่งซับไทป์เป็นตัวขยายซูเปอร์ไทป์
4. เอนทิตีที่ไม่มีแอตตริบิวอื่นที่ไม่ใช่คีย์ให้ยุบรวมกัน

ขั้นตอนที่ 7 การตรวจสอบวิวของผู้ใช้ด้วยกฎของนอร์มัลไลเซชั่น

ให้ทำการตรวจสอบเอนทิตีต่าง ๆ ให้อยู่ในกฎนอร์มัลไลเซชั่น ซึ่งประกอบด้วย 1NF, 2NF, 3NF, 4NF, 5NF

ประโยชน์ที่จะได้รับจากการนอร์มัลไลเซชั่น คือ

1. ลดที่ว่างที่ต้องใช้ในการเก็บข้อมูล
2. ลดความผิดพลาด ความไม่ตรงกับข้อมูลในฐานข้อมูล
3. ลดการเกิดอะนอร์มัลไลเซชั่นของการลบและแก้ไขข้อมูล
4. เพิ่มความคงทนแก่โครงสร้างฐานข้อมูล

ในกรณีที่ได้เอนทิตีที่เป็นนอร์มัลไลเซชั่นที่สมบูรณ์แล้ว สิ่งที่ต้องระวังคือ ไม่แตกเอนทิตีนั้นย่อยลงไปอีก

ขั้นตอนที่ 8 พิจารณาขอบเขตค่าโดเมนของแอตตริบิว ให้กำหนดโดเมนของแอตตริบิวทุกตัวในเอนทิตีแล้วบันทึกในพจนานุกรมข้อมูล โดเมน คือ กลุ่มค่าที่ถูกต้องเป็นไปได้สำหรับแอตตริบิวแต่ละตัว ดังนี้

1. ชนิดของข้อมูล (Data Type) เช่น จำนวนเต็ม วันที่ ตัวอักษร ทศนิยม
2. ความยาว (Length) เช่น 4 หลัก 30 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

ได้มีการกำหนดกฎไว้เป็นพิเศษสำหรับโดเมนของแอตตริบิวบางพวก ต่อไปนี้

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

ขั้นตอนที่ 9 พิจารณากฎทริกเกอร์ดำเนินการ หรือกฎธุรกิจของแอตตริบิว

เป็นกฎที่มีเพื่อตรวจสอบความถูกต้อง และควบคุมผลกระทบเมื่อเราทำการสืบค้นเพิ่ม, แก้ไข,ลบ เราจะพิจารณาผลรวมนั้นที่เกิดกับเอนทิตีอื่นและแอตตริบิวอื่นภายในเอนทิตีที่เรากระทำด้วย เมื่อกำหนดกฎการจัดการต่าง ๆ แล้วให้เก็บลงในพจนานุกรมข้อมูล ซึ่งประกอบด้วยส่วนต่าง ๆ ดังนี้

1. กฎของผู้ใช้ (User Rule) เป็นข้อความอธิบายกำหนดการดำเนินตามความต้องการของผู้ใช้
2. เหตุการณ์ของทริกเกอร์ดำเนินการ (Event) ได้แก่ เพิ่ม (Insert) แก้ไข (Update) ลบ (Delete) หรือสืบค้น (Select)

3. ส่วนประกอบของเหตุการณ์(Object Of Event) ได้แก่ ชื่อของเอนทิตีหรือแอตตริบิวที่มีการเปลี่ยนแปลงหรือถูกสืบค้นในเหตุการณ์นั้น ๆ

4. เงื่อนไขภายใต้ทริกเกอร์ดำเนินการ (Condition)

การกำหนดกฎการจัดการเพื่อประโยชน์ ดังนี้

(1) ความสมบูรณ์ ถูกต้อง และเป็นอันหนึ่งอันเดียวกันของค่าของแอตตริบิว

(2) กำหนดกฎเกณฑ์จัดการสำหรับทุกแอตตริบิว ที่เป็นตัวต้นของดิไรฟว์แอต-ตริบิว เช่น เมื่อมีการแก้ไขค่าของแอตตริบิวต้นกำเนิด ต้องมีทริกเกอร์ดำเนินการในการแก้ไขแอตตริบิวดิไรฟ์วด้วย

(3) กำหนดกฎเกณฑ์การจัดการสำหรับซับไทป์-ซูเปอร์ไทป์ โดยถ้าซับไทป์ถูกลบ ต้องลบซูเปอร์ไทป์ด้วย

(4) กำหนดกฎการจัดการโดยกำหนดด้วยเวลา เช่น ถ้าระเบียนข้อมูลนี้มีอายุเกิน 1 ปี ให้ทำการลบทิ้ง

(5) การกระทำ (Action) เช่น ไม่ยอมรับเหตุการณ์ หรือทริกเกอร์ที่เกี่ยวกับเหตุการณ์นั้น

ขั้นตอนต่าง ๆ ที่ผ่านมาเราได้ทำการพัฒนาจนได้มุมมองของผู้ใช้ที่มีรายละเอียดสมบูรณ์ ขั้นตอนต่อจากนี้จะเป็นการปรับแต่งมุมมองต่าง ๆ เข้าด้วยกัน เพื่อขจัดส่วนที่ซ้ำซ้อนและแก้ปัญหาความไม่ตรงกันของข้อมูล ซึ่งเป็นขั้นตอนที่ต้องใช้ทั้งประสบการณ์และทักษะในการวิเคราะห์มากและนับว่าเป็นส่วนที่สำคัญที่สุด

ขั้นตอนที่ 10 การเชื่อมมุมมองของผู้ใช้เข้าด้วยกันเป็นแบบเบ็ดเสร็จ

พิจารณาส่วนของวิวของผู้ใช้ที่คาบเกี่ยวกัน พยายามลดความเหลือเฟือ ตัดความซ้ำซ้อน และแก้ปัญหาความไม่ถูกต้องตรงกันของข้อมูล โดยอาจมีการเพิ่มความสัมพันธ์ หรือกฎทางธุรกิจใหม่ ๆ ได้ด้วย

สิ่งที่ควรพิจารณาในการรวมเอนทิตี

1. การรวมเอนทิตีที่มีคีย์หลักตัวเดียวกันและค่าที่เป็นไปได้ของคีย์หลักเหมือนกัน จะต้องใช้เอนทิตีใหม่ที่มีแอตตริบิวรวมของสองเอนทิตีเดิม

2. ถ้าเอนทิตีสองอันมีคีย์หลักเดียวกัน แต่ค่าที่เป็นไปได้ของคีย์หลักนั้นเป็นซับเซต (Subset) ของกัน เราจะรวมได้ในรูปแบบของซับไทป์-ซูเปอร์ไทป์ โดยต้องตัดแอตตริบิวที่มีแล้วในเอนทิตีซูเปอร์ไทป์ออกจากเอนทิตีที่เป็นซับไทป์

3. ถ้าเอนทิตีสองอันมีคีย์หลักเดียวกัน แต่มีผลไปกำหนดแอตตริบิวที่ต่างกัน บางตัวให้กำหนดซูเปอร์ไทป์ขึ้นมาอันหนึ่งสัมพันธ์กับของเอนทิตีเดิม

4. การเชื่อมเอนทิตีสองตัวที่คีย์หลักของตัวหนึ่งเป็นคีย์รองของอีกตัว จะได้เอนทิตีใหม่ที่มีคีย์หลักตามเอนทิตีตัวหนึ่ง ส่วนคีย์หลักของเอนทิตีอีกตัวจะกลายเป็นคีย์รองไป และมีแอตตริบิวรวมระหว่างสองเอนทิตีเดิม แล้วตัดแอตตริบิวที่ซ้ำซ้อนออกเสีย และต้องกำหนดกฎธุรกิจให้ด้วยตามข้อบังคับเก่า เช่น ในกรณีคีย์หลักเดิมที่กลายมาเป็นคีย์รองในเอนทิตีอื่นที่ไม่ได้เกี่ยวข้อง

5. การรวมเอนทิตีใด ๆ ต้องไม่มีผลไปเปลี่ยนแปลงเอนทิตีอื่น ๆ ที่ไม่ได้เกี่ยวข้อง

สิ่งที่ควรพิจารณาในการรวมความสัมพันธ์

1. ให้รวมความสัมพันธ์ระหว่างเอนทิตีที่ให้ความหมายเหมือนกันเข้าด้วยกัน โดยถ้าผลทำให้เกิดเป็นความสัมพันธ์แบบหลายต่อหลาย จะต้องทำการแตกเอนทิตีให้เป็นความสัมพันธ์แบบหนึ่งต่อหลายกลุ่ม

2. การรวมความสัมพันธ์ใด ๆ ต้องไม่กระทบกับความสัมพันธ์อื่น ๆ ที่ไม่ต้องการการเปลี่ยนแปลง นอกจากจะพิจารณาแล้วว่า ควรตัดออกเนื่องจากซ้ำซ้อน หรืออาจเพิ่มขึ้นใหม่เพื่อความเหมาะสม

3. จากการรวมเอนทีตีที่หลักเป็นคีย์รองของเอนทิตีอีกตัว ให้ตรวจสอบคีย์ภายนอกของเอนทิตีอื่น ๆ ที่อ้างมาถึงว่า ได้อ้างถึงคีย์หลักหรือคีย์รองของเอนทิตีใหม่ที่ได้จากการรวมนั้น ถ้าอ้างถึงคีย์รองต้องทำการเปลี่ยนให้เป็นคีย์หลัก

4. เมื่อรวมมุมมองแล้วให้กำหนดกฎธุรกิจของคีย์สำหรับความสัมพันธ์ใหม่ด้วย

สิ่งที่ควรพิจารณาในการรวมแอตตริบิว

1. ให้ทำการรวมแอตตริบิว ที่มีความหมายเหมือนกันภายในเอนทิตีเดียวกัน และรวมค่าที่เป็นไปได้ รวมถึงกฎการจัดการเข้าด้วยกัน และให้พิจารณาค่าที่เป็นไปได้ของแอตตริบิวอื่นด้วยว่าเปลี่ยนไปหรือไม่

2. เมื่อรวมเอนทิตีแล้ว ให้พิจารณาตัดแอตตริบิวที่เป็นดิไรฟว์หรือตัวบ่งชี้ (Flag) ที่ไม่จำเป็นทิ้งเสีย

3. หลังจากได้รวม ตัด หรือเพิ่มความสัมพันธ์แล้วให้ทำการนอร์มัลไลซ์อีกครั้งเพื่อตัดสิ่งที่ซ้ำซ้อนออกเสีย

ขั้นตอนที่ 11 การรวมรวมโมเดลข้อมูลที่มีอยู่เข้าด้วยกัน

จะเป็นขั้นที่เกี่ยวข้องกับเชิงมโนภาพ โดยจะรวมโมเดลข้อมูลเชิงตรรกที่ได้ กับของที่มีอยู่เดิม ให้พัฒนาโมเดลใหม่ควบคู่กับการพิจารณากฎเกณฑ์ข้อบังคับของเดิม โดยอาจมีการใช้เอนทิตีหรือความสัมพันธ์ร่วมกับของเดิม และมีการกำหนดเอนทิตีขึ้นมาใหม่ด้วย

ขั้นตอนที่ 12 การวิเคราะห์ความมีเสถียรภาพและการเติบโตในอนาคต

การออกแบบโมเดลที่ทำผ่านจะพิจารณาข้อมูลที่เห็นได้ในปัจจุบันเป็นส่วนมาก สำหรับขั้นนี้ให้พิจารณาถึงสิ่งที่อาจเกิดขึ้นหรือเป็นไปได้ในอนาคต ในกรณีดังต่อไปนี้

(1) อาจมีเอนทิตี หรือความสัมพันธ์ใหม่เกิดขึ้น ทำให้ต้องเพิ่มคีย์ภายนอกในเอนทิตีของเดิม

(2) ความสัมพันธ์แบบหนึ่งต่อหลาย อาจกลายเป็นความสัมพันธ์แบบหลายต่อหลายได้

(3) คีย์หลัก อาจเปลี่ยนแปลงได้เนื่องจากของเดิมไม่เป็นหนึ่งเดียว

กรณีดังกล่าวข้างต้นเมื่อเราพิจารณาแล้วอาจทำการดัดแปลงโมเดลไว้เพื่อรองรับหรือจดบันทึกเก็บไว้ก่อน

หลักการนอร์มัลไลเซชั่น (Normalization)

นอร์มัลไลเซชั่น เป็นกระบวนการเพื่อพัฒนาการเชื่อมต่อของข้อมูลเพื่อตอบคำถามที่ว่าการออกแบบฐานข้อมูลทั้งทางตรรกและทางกายภาพที่ได้ออกมาใช้ได้หรือยัง และควรทำความเข้าใจคำศัพท์ต่าง ๆ ที่เกี่ยวข้อง ดังต่อไปนี้

1. ความสัมพันธ์ คือ กลุ่มของ 1 ทูเบิล (Tuple) (เทียบได้กับระเบียนข้อมูล) โดยมากจะแสดงในรูปแบบของตาราง แต่ละแถวของตาราง คือ 1 ทูเบิลนั่นเอง ซึ่งแต่ละทูเบิลจะมีค่าต่าง ๆ บรรจุอยู่

2. คีย์ คือ แอตตริบิวตั้งแต่ 1 ตัวขึ้นไป ซึ่งจะช่วยระบุ ค้นหา ทูเบิลที่ต้องการในความสัมพันธ์ คีย์ตัวหนึ่งจะระบุถึงทูเบิลหนึ่งเท่านั้น คีย์ประกอบด้วย

(1) คีย์หลัก (Primary Key) คือ แอตตริบิวซึ่งทุกแอตตริบิวในความสัมพันธ์ต้องขึ้นกับคีย์หลัก (Dependence) หรือ กลุ่มของแอตตริบิวที่เป็นหนึ่งเดียว (Unique) และทุกทูเบิลที่จะเพิ่มเข้าในความสัมพันธ์อย่างน้อยแอตตริบิวนี้ต้องมีค่า

(2) คีย์ประกอบ (Compound Key) คือ คีย์ที่ประกอบด้วยแอตตริบิวมากกว่า 1 ตัวขึ้นไป

(3) คีย์รอง (Alternate Key) แอตตริบิวที่มีคุณสมบัติเหมือนคีย์หลักทุกประการ แต่ไม่ได้รับเลือกเป็น Key หลัก

(4) คีย์เป็นได้ (Candidate Key) คือ แอตตริบิวที่มีคุณสมบัติที่สามารถเป็นคีย์หลักได้

การนอร์มัลไลเซชั่น แบ่งออกเป็นหลายระดับ ดังต่อไปนี้

นอร์มัลไลเซชั่น ระดับที่ 1 (First Normal Form : 1NF) เป็นการขจัดแอตตริบิว หรือกลุ่มแอตตริบิวที่ซ้ำกันไปอยู่ในเอนทิตีลูก เมื่อแต่ละรายการในเอนทิตีไม่มีค่าของแอตตริบิวหรือค่าของกลุ่มแอตตริบิวที่ซ้ำกัน

นอร์มัลไลเซชั่น ระดับที่ 2 (Second Normal Form : 2NF) เป็นการขจัดแอตตริบิวที่ไม่ขึ้นกับคีย์หลักออกไป เพื่อให้แอตตริบิวอื่นทั้งหมดขึ้นตรงกับส่วนที่เป็นคีย์หลักทั้งหมดเท่านั้น

นอร์มัลไลเซชั่น ระดับที่ 3 (Third Normal Form : 3NF) เป็นการขจัดแอตตริบิวที่ไม่เป็นคีย์ที่ขึ้นตรงกับแอตตริบิวอื่นที่ไม่ใช่คีย์หลักออกไป เพื่อให้แอตตริบิวที่ไม่ใช่คีย์หลักต้องขึ้นตรงกับทั้งส่วนที่เป็นคีย์หลัก และไม่ขึ้นกับแอตตริบิวอื่นที่ไม่ใช่คีย์หลัก

นอร์มัลไลเซชั่น ระดับที่ 4 (Fourth Normal Form : 4NF) เป็นการทำให้ได้รูปแบบของความสัมพันธ์นอร์มัล ระดับที่ 3 และไม่มีการขึ้นแก่กันแบบหลายค่าอยู่ในความสัมพันธ์

นอร์มัลไลเซชั่น ระดับที่ 5 (Fifth Normal Form : 5NF) คือ ความสัมพันธ์ที่ไม่สามารถนำไปสร้างความสัมพันธ์ใหม่ ทำได้โดยการเชื่อมระหว่างสองความสัมพันธ์ด้วยคีย์ต่างกันได้ โดยปกติมักจะเป็นปัญหา สำหรับคีย์หลักที่เป็นคีย์ประกอบ เป็นขั้นที่พิจารณาได้ยาก จะเกิดระเบียนข้อมูลใหม่ที่ไม่จริงขึ้นมา เมื่อนำเอนทิตีมารวมกัน

หมายเหตุ ในการพัฒนาโมเดลข้อมูลเชิงตรรกสามารถเข้าสู่ชนิดของนอร์มัลไลเซชั่นในระดับที่ 3 ส่วนในระดับถัดไปต้องพิจารณาถึงลักษณะของข้อมูลว่า จำเป็นต้องดำเนินการเพิ่มเติมหรือไม่ ซึ่งมีหลักการซับซ้อนขึ้น

การออกแบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database Design : RDD)

เป็นขั้นตอนการแปลงจากโมเดลข้อมูลเชิงตรรก เข้าสู่ระบบฐานข้อมูลเชิงสัมพันธ์ (Transaction Process) อย่างมีเสถียรภาพภายใต้ระบบจัดการฐานข้อมูลที่ติดตั้งบนเครื่องคอมพิวเตอร์ เป็นการปรับให้เข้ากับหน้าที่ใช้สอยเฉพาะ และตามความต้องการในเรื่องประสิทธิภาพการทำงาน ประกอบกัน 2 ส่วน คือ

I16758341

1. โครงสร้างข้อมูล บอกที่อยู่ของโครงสร้างข้อมูล ได้แก่ เอนทิตี แอตตริบิว และความสัมพันธ์ ด้วยการระบุตารางความสัมพันธ์และสดมภ์ ดังนี้

1.1 ตารางความสัมพันธ์เทียบเท่ากับเอนทิตี และสดมภ์เทียบเท่ากับแอตตริบิวในโมเดลข้อมูลเชิงตรรก โดยคำนึงว่าแต่ละตารางและสดมภ์ที่พบในระบบจัดการฐานข้อมูลเชิงสัมพันธ์ ต้องไม่มีความขัดแย้งกัน

สดมภ์ คือ หน่วยหนึ่งของข้อมูลที่สามารถกำหนดตำแหน่งที่อยู่ได้ อาจจะเรียกอีกอย่างหนึ่งว่า "เขตข้อมูล"

ตาราง คือ กลุ่มของสดมภ์ โดยสดมภ์ทุกตัวในตารางเป็นตัวพรรณาถึงลักษณะคุณสมบัติของตารางนั้น ๆ

1.2 บางกรณีอาจมีการรวมตารางเข้าเป็นตารางเดียว หรือแบ่งตารางออกเป็นหลายตารางเพื่อประโยชน์ในด้านการทำงานรวดเร็วขึ้น

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

ความบูรณภาพของข้อมูลเป็นการรักษาคุณภาพ ความน่าเชื่อถือของข้อมูลในฐานข้อมูลของระบบสารสนเทศ และป้องกันมิให้ข้อมูลเกิดความผิดพลาด ถ้าพิจารณาจากระบบจัดการฐานข้อมูล จะพบว่าปัจจุบันไม่ได้เอื้ออำนวยในการควบคุมโดยตรง กล่าวคือ ภาษาเอสคิวแอล (SQL Language) ซึ่งเป็นภาษาในยุคที่ 4 (4^{th} Generation Language, 4GL) เป็นคำสั่งแบบไม่ใช่วิธีดำเนินการ (Nonprocedural Syntax) ใช้ในการสร้าง, เก็บ, เปลี่ยนแปลง สืบค้น และดูแลรักษา สารสนเทศในฐานข้อมูล ไม่อาจระบุคำสั่งเพื่อให้เกิดบูรณภาพของข้อมูลได้โดยตรง จึงจำเป็นต้องกำหนดกฎเพื่อความบูรณภาพของข้อมูล โดยแบ่งออกเป็น 3 ประเภท คือ

2.1 กฎความบูรณภาพของเอนทิตี(Entity Integrity Rule) กำหนดให้ส่วนประกอบของคีย์หลักเป็นตัวระบุตามเป็นเอกลักษณ์ภายในแต่ละทูเบิลของตารางความสัมพันธ์ ซึ่งจะต้องมีค่าเสมอในการปรับปรุง หรือลบ จำเป็นต้องคงไว้ซึ่งคุณสมบัติของคีย์หลัก

2.2 กฎความบูรณภาพในการอ้างอิง (Referential Integrity Rule) เป็นการหาแหล่งที่มาของคีย์ภายนอก ถ้าตารางความสัมพันธ์ใดมีคีย์ภายนอกปรากฏอยู่ ทุกค่าของคีย์ภายนอกอาจเป็นไปได้ 2 กรณี ดังนี้

(1) ค่านัล

(2) มีค่าที่สอดคล้องกับค่าคีย์หลักในตารางความสัมพันธ์อื่น

ดังนั้นต้องมีการระบุถึงการอ้างอิงระหว่างตารางความสัมพันธ์ด้วยกัน เพื่อตรวจสอบค่าของคีย์ภายนอกว่าตรงกับค่าที่มีอยู่จริงในคีย์หลักหรือไม่ ในการดำเนินการเพิ่ม ปรับปรุงหรือลบ อาจกล่าวอีกนัยหนึ่งว่าเป็นการควบคุมความบูรณภาพของคีย์ภายนอกก็ได้

2.3 กฎความบูรณภาพของโดเมน (Domain Integrity Rule) เป็นการควบคุมความถูกต้องทุก ๆ สดมภ์ที่อยู่ในตารางความสัมพันธ์ ได้แก่

(1) ชนิดของข้อมูล เช่น ตัวเลข ตัวอักษร วันที่
(2) ความยาวของข้อมูล
(3) ช่วงของค่าของข้อมูล
(4) ค่าโดยปริยาย
(5) ความเป็นเอกลักษณ์
(6) ต้องมีค่าหรือไม่

ขั้นตอนการออกแบบฐานข้อมูลเชิงสัมพันธ์ มีดังนี้

RDD1 ระบุตารางความสัมพันธ์จากเอนทิตี การกำหนดตารางในการสร้างฐานข้อมูลเชิงสัมพันธ์จะแทนหนึ่งเอนทิตีจากโมเดล ข้อมูลเชิงตรรก ด้วยหนึ่งตารางเท่านั้น

RDD2 ระบุสดมภ์ในตารางความสัมพันธ์ สำหรับแต่ละแอตตริบิวที่ขยายคุณสมบัติของเอนทิตีในฐานข้อมูลเชิงสัมพันธ์นี้ จะเห็นว่าแถวในตารางหนึ่งมีความหมายเหมือนกับระเบียนของเอนทิตีที่ถูกแปลงไปเป็นตารางนั้น ในการกำหนดสดมภ์ของตารางพิมพ์จะใช้แอตตริบิวนั้น โดยแทนหนึ่งแอตตริบิวด้วยหนึ่งสดมภ์

RDD3 ดัดแปลงโครงสร้างให้เหมาะสมกับสภาพแวดล้อม ของระบบจัดการฐานข้อมูลเชิงสัมพันธ์ และปรับปรุงในเรื่องประสิทธิภาพการทำงานด้วย

RDD4 ออกแบบกฎธุรกิจที่เกี่ยวข้องกับเอนทิตี

1. บังคับคุณสมบัติทางตรรกของคีย์หลักในการติดตั้งระบบแบบความสัมพันธ์ เช่น ความเป็นเอกลักษณ์ ไม่อนุญาตให้ค่าในคีย์หลักไม่มีค่า เป็นต้น
2. บังคับคุณสมบัติทางตรรกของคีย์รองในการติดตั้งระบบแบบความสัมพันธ์ เช่น ความเป็นเอกลักษณ์ เป็นต้น

RDD5 ออกแบบกฎธุรกิจที่เกี่ยวข้องกับความสัมพันธ์ ข้อจำกัดในการเพิ่มลบและแก้ไขแอตตริบิวที่เป็นคีย์ในลักษณะการอ้างอิงให้สอดคล้อง

RDD6 ออกแบบกฎธุรกิจเพิ่มเติมเกี่ยวกับแอตตริบิว ได้แก่ ทริกเกอร์ดำเนินการ การควบคุมค่าโดเมน ชนิดของข้อมูล รูปแบบข้อมูล ค่าที่ยอมรับ (ขอบเขต ความต่อเนื่องของข้อมูล) ความเป็นเอกลักษณ์ อนุญาตให้มีค่าได้หรือไม่และค่าปริยาย การปรับปรุงโมเดลข้อมูลใหม่ เมื่อมีสารสนเทศใหม่เพิ่มเติม อาจทำให้เกิดการแบ่งแยก (Splitting) หรือการผสาน (Merging) ของเอนทิตีขึ้น โดยพิจารณาว่าแอตตริบิวเดิมควรจะถูกย้ายจากเอนทิตีเดิมไปสู่เอนทิตีใหม่หรือไม่ จะต้องคำนึงถึงสภาพความเป็นจริง เช่น

1. แอตตริบิว มีความสัมพันธ์กับแอตตริบิวอื่นอย่างไร
2. การสร้างเอนทิตีใหม่สามารถแก้ปัญหาการสืบค้นข้อมูลได้หรือไม่
3. ข้อเท็จจริงของแอตตริบิวอ้างอิงถึงแอตตริบิวอื่นหรือไม่
4. เอนทิตีใหม่สัมพันธ์กับเอนทิตีเดิมและกับเอนทิตีอื่น ๆ ในโมเดลอย่างไร

การจัดกลุ่มในระดับเอนทิตี (Entity Clustering)

ตั้งอยู่บนพื้นฐานของขอบเขตข้อมูล หมายความว่าข้อมูลทั้งหมดที่จำเป็นต้องการใช้ภายในหน่วยงาน โดยแต่ละหน่วยงานมีหน้าที่รับผิดชอบในการจัดกลุ่มเอนทิตี เนื่องจากเป็นเจ้าของข้อมูล โครงสร้างข้อมูลภายในองค์กรที่ปรากฏอยู่ จะเป็นตัวบ่งชี้ว่าเอนทิตีควรจะปรากฏในหน่วยงานใดบ้าง การจัดกลุ่มอาจเป็นกลุ่มย่อยของกลุ่มใหญ่รวมกันหลาย ๆ กลุ่มก็ได้ ทั้งนี้ ขึ้นกับระดับความละเอียดที่ต้องการแสดง เช่น ในระดับบนโมเดลข้อมูลที่ได้จากการจัดกลุ่ม มักแสดงเฉพาะเอนทิตีหลักที่มีความเชื่อมโยงกับเอนทีตีอื่น เพื่อแสดงความสัมพันธ์ของข้อมูลทุกหน่วยงานภายในองค์กรโดยสังเขป

ส่วนในระดับล่างลงมาอาจแสดงกลุ่มเอนทิตีว่าสนใจเฉพาะหน่วยงานเท่านั้น ในกรณีที่หน่วยงานใดมีข้อมูลที่ซับซ้อนหรือมากก็จัดกลุ่มย่อยในระดับล่างลงมาอีกได้ การจัดกลุ่มเอนทิตีอาจเกิดได้ในระหว่างการออกแบบฐานข้อมูลทั้ง 2 ขั้นตอน ได้แก่ ขั้นตอนเชิงตรรก และขั้นตอนเชิงกายภาพ

เนื่องจากโมเดลข้อมูลประกอบด้วยเอนทิตีมากมาย การจัดกลุ่มของเอนทิตีในขั้นตอนเชิงตรรกในแต่ละเอนทีตี สามารถจัดกลุ่มของแอตตริบิวเพื่อขยายคุณสมบัติต่าง ๆ ขึ้นกับธรรมชาติของข้อมูลที่ปรากฏอยู่และโครงสร้างข้อมูล แต่ขั้นตอนเชิงกายภาพไม่เป็นเช่นนั้น เป็นการจัดเข้ากลุ่มเอนทิตีเพื่อตอบสนองวิวของผู้ใช้ โดยต้องคำนึงถึงการดำเนินการความต้องการในการเข้าถึงฐานข้อมูล

โมเดลข้อมูลเชิงกายภาพ (Physical Data Model)

เป็นวิธีการเลือกโครงสร้างในการเก็บข้อมูล และทิศทางการเข้าถึงข้อมูล สำหรับแฟ้มข้อมูลของฐานข้อมูล เพื่อให้ได้มาซึ่งฐานข้อมูลที่มีประสิทธิภาพมากที่สุด แต่ละระบบจัดการฐานข้อมูลจะมีระบบจัดการแฟ้มข้อมูล ทิศทางการเข้าถึงข้อมูล ดัชนีมีตัวชี้ ฯลฯ ที่ต่างกันออกไป ซึ่งจะเป็นตัวจำกัดให้เลือกสิ่งที่เหมาะสมกับแฟ้มข้อมูล

สิ่งที่พิจารณาในการออกแบบเชิงกายภาพ คือ

1. เวลาในการตอบสนอง (Response Time) คือ ช่วงเวลาตั้งแต่การส่งงานเข้าไป กับเมื่อได้รับผลลัพธ์ที่ต้องการออกมา
2. การใช้ที่ว่าง (Space Utilization) จำนวนที่ว่างของหน่วยเก็บ จะถูกใช้โดยแฟ้มข้อมูลของฐานข้อมูล และโครงสร้างทิศทางการเข้าถึงข้อมูล
3. งานที่ได้ออกมา (Transaction Throghtput) จะคิดเป็นค่าเฉลี่ยคำนวณจากจำนวนงานที่สามารถประมวลผลได้ โดยระบบจัดการฐานข้อมูลต่อหนึ่งหน่วยเวลา

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย