บทที่ 2

ระบบฐานข้อมูล

ประวัติความเป็นมาของระบบจัดการฐานข้อมูล

ความจริงแล้วนับเป็นเรื่องยากที่จะกล่าวลงไปอย่างแน่ชัดว่า ระบบฐานข้อมูลได้ถือกำเนิดขึ้นเมื่อใด แต่ก็มีเหตุผลที่น่าเชื่อถือได้ว่า ต้นกำเนิดของระบบฐานข้อมูลเกิดขึ้นจากโครงการอพอลโลของสหรัฐอเมริกา อันเป็นโครงการส่งมนุษย์อวกาศไปดวงจันทร์ซึ่งเกิดขึ้นเมื่อประมาณ 30 ปี ที่ผ่านมา และความสำเร็จในครั้งนั้นจะต้องมีการตระเตรียมงานที่มีความละเอียดรอบครอบสูงสุด ซึ่งแน่นอนว่าข้อมูลที่ใช้ในงานดังกล่าวจะต้องมีมากมาย และเบื้องหลังการจัดการระบบข้อมูลในโครงการนี้ก็เกิดจากการว่าจ้างบริษัทไอบีเอ็ม ให้พัฒนาระบบจัดการข้อมูลที่เรียกว่า GUAM(Generalized Update Access Method) ซึ่งถือได้ว่าเป็นต้นกำเนิดของระบบจัดการฐานข้อมูล และบริษัทไอบีเอ็มก็พบว่าการทำงานของระบบ GUAM ไม่เพียงแต่จะมีประโยชน์ในโครงการอวกาศเท่านั้น แต่ยังสามารถอำนวยประโยชน์ต่อการประมวลผลข้อมูลในงานธุรกิจอื่น ๆ ได้ ดังนั้นในอีกสองปีถัดมา ไอบีเอ็มจึงได้พัฒนาระบบจัดการฐานข้อมูลขึ้นมาใหม่เพื่อใช้ในงานธุรกิจทั่ว ๆ ไป อันได้แก่ระบบ DL/I(Data Language/I) จากนั้นก็ได้รับการพัฒนาต่อเนื่องจนในที่สุดก็ได้มาซึ่งระบบ IMS (Information Management System) ซึ่งยังคงใช้มาอยู่จนถึงปัจจุบัน

นอกจากบริษัทไอบีเอ็มแล้วก็ยังมีบริษัท GE(General Electric) ที่ทำการพัฒนาระบบฐานข้อมูล อันได้แก่ระบบที่ชื่อว่า IDS(Integrated Data Store) โดยมี Charles Bachman เป็นหัวหน้ากลุ่มผู้ทำการพัฒนาระบบ และก็ได้เริ่มใช้งานประมาณปี พศ.2509 และถือได้ว่าเป็นต้นกำเนิดของระบบโคดาซิวล์(CODASYL) ซึ่งเป็นโมเดลแบบเครือข่ายที่ยังมีการใช้งานจนถึงปัจจุบัน หลังจากนั้นไม่นานก็ได้มีการจัดตั้งคณะกรรมการขึ้นเพื่อร่างข้อกำหนดมาตรฐานของระบบโคดาซิวล์ขึ้นสำเร็จในปี พศ.2515

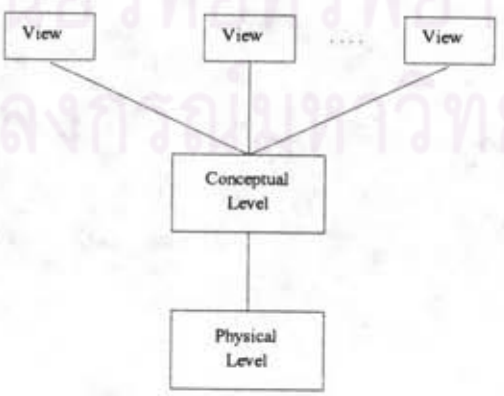
ในช่วงเวลาไล่เลี่ยกับที่ระบบโคดาซิวล์ฉบับมาตรฐานได้ถูกบัญญัติขึ้น ดร.คอดด์ (E.F. Codd) ก็ได้เสนอผลงานทางวิชาการ เกี่ยวกับโมเดลใหม่อีกโมเดลหนึ่งที่เรียกว่า "โมเดลเชิงสัมพันธ์" ซึ่งหลักทฤษฎีของ ดร.คอดด์ ได้รับความสนใจจากนักวิจัยและบริษัท

ผู้ค้าเครื่องคอมพิวเตอร์เป็นอย่างยิ่ง และเป็นผลงานที่มีความสำคัญยิ่งต่อวงการระบบฐานข้อมูลที่พัฒนาขึ้นตามหลักการและแนวความคิดของ ดร.คอดด์ ตัวอย่างเช่นระบบฐานข้อมูล R ซึ่งเป็นระบบที่พัฒนาขึ้นโดยบริษัทไอบีเอ็ม แต่กว่าที่ระบบฐานข้อมูลเชิงสัมพันธ์จะได้ก้าวเข้าสู่วงการธุรกิจก็ได้ใช้เวลาถึง 10ปี และระบบ R ก็เป็นเพียงระบบต้นแบบที่ยังคงใช้อยู่เฉพาะในห้องวิจัยเท่านั้น โดยบริษัทไอบีเอ็มได้สร้างระบบ DB2 ขึ้นมาแทนเพื่อนำออกสู่งานธุรกิจจริง

ในช่วง พศ.2525 เป็นต้นมา ถือได้ว่าเป็นยุคทองของระบบฐานข้อมูลที่ได้ก้าวเข้าสู่ตลาดคอมพิวเตอร์อย่างเต็มตัว จนกระทั่งในปัจจุบันก็ได้มีซอฟต์แวร์ที่เกิดมาเกื้อหนุนการทำงานของระบบจัดการฐานข้อมูลอีกมากมาย ไม่ว่าจะเป็นระบบพจนานุกรมข้อมูล ระบบช่วยในการออกแบบและสร้างข่ายงาน และภาษาสำหรับการเรียกค้น เป็นต้น และการเจริญเติบโตของระบบฐานข้อมูลก็คงไม่หยุดนิ่งลงอย่างแน่นอน (ดวงแก้ว สวามิภักดิ์, 2534)

ระบบฐานข้อมูล

ระบบฐานข้อมูล คือการนำข้อมูลในองค์กรที่มีความเกี่ยวข้องหรือสัมพันธ์กันมารวมไว้อย่างมีระบบในที่เดียวกัน โดยที่ผู้ใช้แต่ละคนจะมองข้อมูลในแง่มุมที่แตกต่างกันไปตามจุดประสงค์ของการประยุกต์ใช้งาน และผู้ใช้ไม่จำเป็นต้องสนใจว่าลักษณะการจัดเก็บข้อมูลที่แท้จริงเป็นอย่างไร เพียงแต่มุ่งอยู่ที่การเรียกใช้ข้อมูลให้เกิดประสิทธิภาพเท่านั้น ซึ่งการทำให้ผู้ใช้ไม่ต้องจัดการเกี่ยวกับการจัดเก็บข้อมูลที่แท้จริงนี้ เพราะว่าระบบฐานข้อมูลได้จัดแบ่งระดับของข้อมูลออกเป็น 3 ระดับ ดังนี้



Three level of data abstraction

รูปที่ 2.1 ลำดับชั้น 3 ระดับของระบบฐานข้อมูล

1. ระดับกายภาพ(Physical level) เป็นระดับต่ำที่สุดเพราะในระดับนี้จะทำการเก็บข้อมูลจริงลงในจานบันทึกข้อมูล(Diskette หรือ Hard Disk) โดยจะมีการกำหนดเป็นโครงสร้างข้อมูลขึ้น และใช้ส่วนจัดการแฟ้มข้อมูล(File Management)ในการติดต่อกับสื่อบันทึกข้อมูล อีกทั้งยังต้องทำงานร่วมกับระบบปฏิบัติการของระบบอีกด้วย

2. ระดับหลักการ(Conceptual level) จะเป็นการกำหนดเงื่อนไขต่าง ๆ สำหรับสร้างความสัมพันธ์ กฎเกณฑ์เพื่อควบคุมความถูกต้องของข้อมูล อีกทั้งผู้ที่จะมีสิทธิ์ใช้และรายละเอียดที่จำเป็นทั้งหมด ทั้งนี้จะเป็นส่วนเชื่อมระหว่างระดับกายภาพกับระดับมุมมองของผู้ใช้นั่นเอง

3. ระดับมุมมองของผู้ใช้(View level) เป็นระดับที่ผู้ใช้จะติดต่อกับระบบฐานข้อมูลได้ โดยจะอาศัยมุมมอง(View)ที่สร้างขึ้นมา ซึ่งผู้ใช้แต่ละคนสามารถมีมุมมองที่แตกต่างกันออกไปทั้ง ๆที่เป็นฐานข้อมูลตัวเดียวกันก็ได้ ทั้งนี้เพราะระบบฐานข้อมูลจะทำการเชื่อมข้อมูลเข้าไปในระดับหลักการจนเข้าไปถึงในระดับการภาพให้กับผู้ใช้เอง

การแบ่งลักษณะของข้อมูลออกเป็น 3 ระดับนี้ ก็เพื่อให้เกิดสภาพที่เหมาะสมที่ทำให้ผู้ใช้ไม่ต้องคอยพะวงกับรายละเอียดต่าง ๆในการเก็บข้อมูล และไม่จำเป็นต้องทราบเกี่ยวกับข้อมูลส่วนอื่น ๆ ที่ตนไม่ได้ใช้ซึ่งทำให้เกิดความอิสระที่ผู้ใช้ที่ทำการพัฒนาโปรแกรมไม่ต้องแก้ไขโปรแกรมทุกครั้งที่มีการเปลี่ยนแปลงโครงสร้างของฐานข้อมูล

โครงสร้างที่ได้จากการออกแบบฐานข้อมูลหนึ่ง ๆจะถูกใช้โดยระบบฐานข้อมูลเพื่อให้เกิดสภาพที่สอดคล้องกับการนิยามทั้งสามระดับของระบบฐานข้อมูลดังกล่าวข้างต้นโครงสร้างนี้จะถูกเรียกว่า "พจนานุกรมข้อมูล(Data Dictionary)" โดยพจนานุกรมข้อมูลจะสามารถสร้างขึ้นจากภาษาที่ใช้นิยามข้อมูล(Data definition language) จากนั้นจะถูกแปลเก็บไว้ในแฟ้มข้อมูลเพื่อให้ระบบจัดการฐานข้อมูลเรียกใช้ต่อไป
(Henry F.Korth and Abraham Silberschatz, 1991)

ฐานข้อมูลเชิงสัมพันธ์(Relational Database)

ระบบฐานข้อมูลเชิงสัมพันธ์ เป็นหนึ่งในระบบฐานข้อมูลที่มีอยู่หลายระบบในปัจจุบัน และเป็นรูปแบบที่ใช้กันอยู่มากที่สุด โดยที่ระบบฐานข้อมูลเชิงสัมพันธ์นี้จะมีลักษณะเป็นตารางความสัมพันธ์แบบสองมิติ คือจะมีส่วนของ แถว และสดมภ์ ซึ่งตารางความสัมพันธ์นี้จะต้องคุณสมบัติดังต่อไปนี้

1. แต่ละช่องของตารางจะบรรจุข้อมูลเพียงค่าเดียว
2. แต่ละสดมภ์มีความหมายแตกต่างกัน
3. การเรียงลำดับของสดมภ์ไม่ถือว่ามีความสำคัญ
4. ข้อมูลในแต่ละแถวจะต้องแตกต่างกัน
5. การเรียงลำดับแถวไม่ถือว่ามีความสำคัญ

และเมื่อรวบรวมตารางความสัมพันธ์ต่าง ๆที่มีความสัมพันธ์ระหว่างกันเหล่านี้ไว้ด้วยกันก็จะเกิดเป็น "ฐานข้อมูลเชิงสัมพันธ์" และจากความหมายในทางทฤษฎีของตารางความสัมพันธ์ เราก็สามารถโยงเข้ากับความเข้าใจในเรื่องของระบบแฟ้มข้อมูลที่คุ้นเคยได้โดยให้ความหมายดังต่อไปนี้

สดมภ์ หมายถึง เขตข้อมูล(Field)
แถว หมายถึง ระเบียน(Record)
ตาราง หมายถึง แฟ้มข้อมูล(File)

ส่วนของสดมภ์เองยังมีการกำหนดกฎเกณฑ์เพื่อใช้ควบคุมค่าที่จะเก็บในแต่ละสดมภ์ โดยการกำหนดกรอบค่าของข้อมูลที่จะยอมให้บันทึกลงในฐานข้อมูลได้ เรียกว่า "โดเมน" (domain) เช่น กำหนดให้สดมภ์ของรหัสพนักงานมีค่าระหว่าง 0001 ถึง 9999

ระหัสพนักงาน	ชื่อ-นามสกุล	วันเกิด
1111	สมชาย รักเจริญ	100405
1122	สมศรี แซ่ลี้	210909
1133	ทนงศักดิ์ มีทรัพย์หลาย	131006
1144	สมคิด งานดี	280203

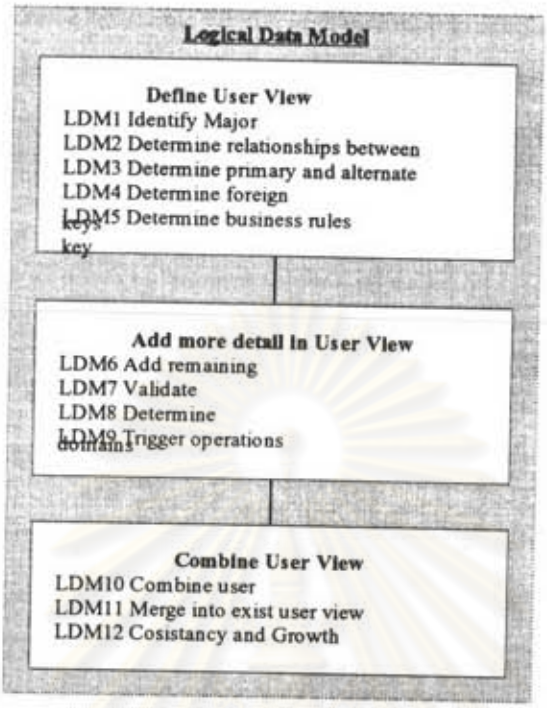
รูปที่ 2.2 ตารางพนักงานหลัก

นอกจากโดเมนแล้ว ยังมีการกำหนดอีกว่าในแต่ละแถวจะใช้สดมภ์ใดบ้าง เป็นตัวอ้างอิงในการเรียกใช้ โดยสดมภ์ที่ใช้อ้างอิงนี้อาจจะใช้หนึ่งหรือมากกว่าหนึ่งก็ได้ และเราจะเรียกส่วนของสดมภ์นี้ว่า "กุญแจหลัก(Primary key)" เช่นในตารางของพนักงานหลักเราสามารถใช้รหัสพนักงานเป็นกุญแจหลักได้ โดยถ้าเราระบุว่ารหัสพนักงานเท่ากับ 1122 จะหมายถึงข้อมูลในแถวที่ 2 ของตารางพนักงานหลักนั่นเอง

ในระบบฐานข้อมูลตารางความสัมพันธ์ต่าง ๆ ที่มีอยู่ นอกจากจะมีความสัมพันธ์ภายในตารางเองแล้วยังจะมีความสัมพันธ์กันระหว่างตารางเหล่านี้ด้วย ซึ่งการสร้างความสัมพันธ์ระหว่างตารางนี้ จะทำได้โดยใช้การจับคู่ระหว่างสดมภ์ที่เหมือนกันของตารางความสัมพันธ์ทั้งสองและจะเรียกสดมภ์คู่นี้ว่า "กุญแจนอก(Foreign key)" การสร้างความสัมพันธ์ระหว่างตารางในฐานข้อมูลจะทำได้สามชนิดคือ แบบหนึ่งต่อหนึ่ง แบบหนึ่งต่อกลุ่ม แบบกลุ่มต่อหนึ่ง และแบบกลุ่มต่อกลุ่ม

การออกแบบโมเดลข้อมูลเชิงตรรกะ

ในการออกแบบโมเดลเชิงตรรกะมีด้วยกันหลายขั้นตอน โดยจะเริ่มจากการออกแบบทางด้านโครงสร้างพื้นฐานก่อนได้แก่ ตารางความสัมพันธ์ ความสัมพันธ์ กุญแจหลัก กุญแจสำรอง กุญแจภายนอก และกฎเกณฑ์พื้นฐาน จากนั้นจึงเริ่มเพิ่มรายละเอียดในระดับที่มุมองของผู้ใช้(User View) แล้วรวมรายละเอียดเหล่านั้นเข้าด้วยกันในขั้นสุดท้าย จึงได้เป็นโมเดลข้อมูลเชิงตรรกะที่สมบูรณ์ ขั้นตอนต่าง ๆ สรุปได้ดัง รูปที่ 2.3



รูปที่ 2.3 ขั้นตอนการออกแบบโมเดลเชิงตรรก

1. การกำหนดตารางความสัมพันธ์

โดยกำหนดชื่อและความหมายลงในพจนานุกรมข้อมูล และเขียนลงโมเดลข้อมูลด้วยการตั้งชื่อไม่ควรเกิน 20 ตัวอักษร การกำหนดตารางความสัมพันธ์นั้นเป็นงานที่ยากและต้องอาศัยความร่วมมือของผู้ที่เข้าใจระบบที่เราออกแบบเพื่อคัดเลือกสิ่งที่ถูกต้องมีความสำคัญและเหมาะสมที่สุดมาเป็นตารางความสัมพันธ์ วิธีการอย่างคร่าว ๆ ก็คือให้พิจารณาข้อมูลทั้งหมดที่มีและจัดกลุ่มของข้อมูลโดยดูจากค่าและความหมาย ถ้าสามารถรวมกลุ่มกันได้ก็ให้รวมเข้าไว้ในตารางความสัมพันธ์เดียวกัน นอกจากนั้นเรายังสามารถแยกตารางความสัมพันธ์ย่อยลงไปได้อีก โดยเรียกว่าเป็นซับไทป์(Subtype)ของอีกตารางความสัมพันธ์หนึ่ง

2. การกำหนดความสัมพันธ์ระหว่างตารางความสัมพันธ์

ส่วนประกอบที่สำคัญของโมเดลเชิงตรรกะ คือตารางความสัมพันธ์และความสัมพันธ์จะทำได้โดยกำหนด ชื่อ ความหมาย ทิศทาง และขนาดอัตราส่วน ที่เกิดความสัมพันธ์นั้น ๆ พร้อมทั้งบันทึกลงในพจนานุกรมข้อมูลด้วยสำหรับชื่อก็ไม่ควรเกิน 20 ตัวอักษร เช่นเดียวกับชื่อของตารางความสัมพันธ์ สำหรับความหมายถ้าเป็นภาษาอังกฤษให้ใช้คำกริยาในรูปแบบปัจจุบัน(Present Tense)

การกำหนดความสัมพันธ์ระหว่างตารางความสัมพันธ์นั้น ต้องคำนึงถึงทิศทางของการกำหนดความสัมพันธ์นั้นด้วยว่าเป็นความสัมพันธ์จากตารางความสัมพันธ์ใดไปตารางความสัมพันธ์ใดโดยใช้ลูกศรแทนทิศทางของความสัมพันธ์ และจะต้องมีตารางความสัมพันธ์หนึ่งเป็นตารางความสัมพันธ์หลัก เราเรียกตารางความสัมพันธ์หลักนั้นว่า ตารางความสัมพันธ์ระดับพ่อแม่(Parent Entity) และเรียกตารางความสัมพันธ์ที่ตามมาว่าตารางความสัมพันธ์ระดับลูก (Child Entity) และหัวลูกศรจะชี้ไปทางตารางความสัมพันธ์ระดับลูกเสมอ โดยวิธีนี้เราสามารถแบ่งความสัมพันธ์ออกเป็น 3 ประเภท โดยขึ้นอยู่กับอัตราส่วนของระเบียน(Record)ในตารางความสัมพันธ์ระดับพ่อแม่และตารางความสัมพันธ์ระดับลูก คือ

2.1 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง(one-to-one) เป็นความสัมพันธ์ที่เมื่อพิจารณาระเบียนใดระเบียนหนึ่งในตารางความสัมพันธ์หนึ่ง จะมีความสัมพันธ์กับระเบียนในอีกตารางความสัมพันธ์หนึ่งเพียงระเบียนเดียวเท่านั้น

2.2 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม(one-to-many) ความสัมพันธ์จะเกิดขึ้นได้ตั้งแต่ 0 จนถึงหลาย ๆ ครั้ง โดยแต่ละระเบียนของตารางความสัมพันธ์ลูกจะสัมพันธ์กับตารางความสัมพันธ์แม่ได้หนึ่งระเบียนเท่านั้น แต่หนึ่งระเบียนในตารางความสัมพันธ์แม่จะสัมพันธ์กับตารางความสัมพันธ์ลูกได้หลายระเบียน

2.3 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม(many-to-many) เป็นความสัมพันธ์ที่เกิดขึ้นได้ตั้งแต่หนึ่งครั้งจนถึงหลายๆครั้งในทั้งสองทิศทาง กล่าวคือระเบียนใดระเบียนหนึ่งในตารางความสัมพันธ์ลูกจะมีความสัมพันธ์กับระเบียนในตารางความสัมพันธ์แม่ได้หลายระเบียน และในทางกลับกันระเบียนใดระเบียนหนึ่งในตารางความสัมพันธ์แม่ก็จะสัมพันธ์กับระเบียนในตารางความสัมพันธ์ลูกได้หลายระเบียนเช่นกัน

หลังจากที่เราสามารถแบ่งกลุ่มความสัมพันธ์ ระหว่างตารางความสัมพันธ์ได้เรียบร้อยแล้วจะพบว่าความสัมพันธ์แบบกลุ่มต่อกลุ่มเป็นสิ่งที่เราต้องสนใจมากที่สุด เพราะเป็นตัวทำให้การสร้างฐานข้อมูลเชิงตรรกะมีความยุ่งยากซับซ้อนมากขึ้น โดยปกติฐานข้อมูลโดยทั่วไปจะไม่สามารถจัดการกับความสัมพันธ์แบบกลุ่มนี้ได้ ดังนั้นเราจึงควรกระจายความสัมพันธ์แบบกลุ่มต่อกลุ่มมาเป็นความสัมพันธ์แบบหนึ่งต่อหลายสองความสัมพันธ์ โดยกำหนดตารางความสัมพันธ์ใหม่ขึ้นมาอีกหนึ่งตัว ให้มีความสัมพันธ์กับตารางความสัมพันธ์เดิมแบบหนึ่งต่อหลาย

จากลักษณะของความสัมพันธ์แบบหนึ่งต่อหนึ่ง ถ้าตารางความสัมพันธ์ระดับลูกหลาย ๆตารางความสัมพันธ์เป็นส่วนประกอบของตารางความสัมพันธ์ระดับพ่อแม่เพียงตัวเดียว และตารางความสัมพันธ์เหล่านั้นมีความหมายในเรื่องเดียว ก็สามารถรวมตารางความสัมพันธ์เหล่านั้นเป็นตารางความสัมพันธ์เดียวได้ เรียกว่าความสัมพันธ์ระดับบนและระดับล่าง (Supertypes and Subtypes) ซึ่งเป็นความสัมพันธ์อีกแบบหนึ่ง ที่แสดงถึงความสัมพันธ์ระหว่างตารางความสัมพันธ์ระดับพ่อแม่กับชุดของตารางความสัมพันธ์ระดับลูกที่มีความหมายในเรื่องเดียวกัน

3. การกำหนดกุญแจหลักและกุญแจรอง

หลังจากที่ได้กำหนดตารางความสัมพันธ์ต่าง ๆแล้วขั้นตอนต่อไปของการสร้างโมเดลข้อมูลทางตรรกะ คือ การเพิ่มข้อมูลที่เรียกว่า แอตตริบิวลงในทุก ๆ ตารางความสัมพันธ์

กุญแจ คือ ชุดของแอตตริบิวที่เล็กที่สุดที่ใช้อ้างถึงระเบียนต่าง ๆในตารางความสัมพันธ์ แอตตริบิวทำหน้าที่เป็นกุญแจจะมีค่าซ้ำกันไม่ได้(Unique) และทุก ๆตารางความสัมพันธ์จะต้องมีกุญแจหลักเสมอ กุญแจหลักมีคุณสมบัติพิเศษที่แตกต่างจากกุญแจอื่น ๆ คือ เป็นค่าว่าง(Null value)ไม่ได้ นอกจากนั้นในตารางความสัมพันธ์หนึ่ง ๆ อาจมีกุญแจรองเพื่อใช้ในการอ้างอิงถึงระเบียนต่าง ๆ ในตารางความสัมพันธ์ ซึ่งชุดของแอตตริบิวทำหน้าที่เป็นกุญแจรองจะต้องเป็นคนละชุดกับแอตตริบิวที่เป็นกุญแจหลักเสมอ และในตารางความสัมพันธ์หนึ่ง ๆ จะมีกุญแจรองหรือไม่ก็ได้

สิ่งสำคัญอีกประการหนึ่งคือตารางความสัมพันธ์ที่เป็นซับไทป์จะต้องมีกุญแจหลักอันเดียวกับตารางความสัมพันธ์ที่เป็นซุปเปอร์ไทป์ของมัน หลังจากกำหนดแล้วให้ตั้งชื่อระบุในโมเดลข้อมูลเชิงตรรกะพร้อมทั้งใส่ในพจนานุกรมข้อมูลด้วย การตั้งชื่อควรกำหนดสั้น ๆ ง่าย ๆ โดยอาจใช้ชื่อย่อก็ได้ และควรหลีกเลี่ยงการตั้งชื่อแอตตริบิวของสองสิ่งที่ไม่เหมือนกันด้วยชื่อเดียวกัน รูปแบบการกำหนดอาจทำได้ดังนี้

Entity name . Modifier(s) - Classword

4. การกำหนดกุญแจภายนอก

เมื่อกำหนดกุญแจหลักและกุญแจรองได้แล้ว ให้กำหนดกุญแจภายนอกสำหรับตารางความสัมพันธ์ที่มีความสัมพันธ์กันทุกอัน กุญแจภายนอกคือแอตตริบิวในตารางความสัมพันธ์ระดับลูกที่เป็นกุญแจหลักของตารางความสัมพันธ์ระดับพ่อแม่ เพื่อใช้ในการอ้างถึงระเบียนในตารางความสัมพันธ์ระดับพ่อแม่ และแสดงถึงความสัมพันธ์ระหว่างตาราง

ความสัมพันธ์ต่าง ๆ ซึ่งอาจให้กุญแจหลักเป็นกุญแจภายนอกด้วยก็ได้ กฎการตั้งชื่อแอตตริบิว ที่เป็นกุญแจภายนอกดังนี้

Entity name . Parent entity * Modifier(s) - Classword

ความสำคัญของกุญแจภายนอก คือ

4.1 ทำให้เกิดกฎธุรกิจ(Business Rules) ในตารางความสัมพันธ์ต่าง ๆ

4.2 ถึงแม้จะดูเหมือนว่าการกำหนดกุญภายนอกเป็นการทำให้ซ้ำซ้อนแต่จริง ๆ แล้วมันจะเป็นตัวช่วยในการออกแบบเพื่อให้เราสามารถตรวจสอบได้ว่าตารางความสัมพันธ์อันไหนเป็นตารางความสัมพันธ์ลูกและอันไหนเป็นตารางความสัมพันธ์แม่

4.3 ทำให้ขั้นตอนการออกแบบฐานข้อมูลง่ายขึ้น ข้อควรระวัง ไม่ควรจะกำหนดให้กุญแจรองเป็นกุญแจภายนอกด้วยเพราะจะทำให้ซับซ้อนและอาจเป็นค่าว่าง(Null)ได้ ซึ่งกุญแจรองไม่ควรเป็นเช่นนั้น

5. การกำหนดกฎการจัดการกับข้อมูล

เป็นขั้นตอนที่ทำขึ้นเพื่อความสมบูรณ์ของข้อมูล และความถูกต้องตรงกันของค่าของข้อมูล กฎการจัดการกับข้อมูลแบ่งได้เป็น 3 ประเภทคือ

5.1 กฎธุรกิจของกุญแจ(Key Business Rules) เป็นการกำหนดกฎเพื่อความสมบูรณ์ของความสัมพันธ์ กฎต่าง ๆ จะมีผลจากการจัดการต่าง ๆ บนความสัมพันธ์อันได้แก่ การเพิ่ม การลบ และการแก้ไขข้อมูล เช่นกำหนดผลที่จะเกิดขึ้นจากการเปลี่ยนค่าของกุญแจหลัก

5.2 โดเมน(Domain) คือการกำหนดกฎเพื่อความสมบูรณ์ถูกต้องของแอตตริบิว ซึ่งการกำหนดข้อบังคับและค่าที่เป็นไปได้สำหรับแอตตริบิว ทั้งที่เป็นกุญแจและไม่ใช่กุญแจ

5.3 กฎการจัดการ(Triggering Operation) เป็นการกำหนดผลกระทบจากการเพิ่ม การลบ การแก้ไข หรือดึงข้อมูล ที่เกิดกับตารางความสัมพันธ์อื่น หรือแอตตริบิวอื่นภายในตารางความสัมพันธ์เดียวกัน

สำหรับในขั้นตอนนี้จะพิจารณาเฉพาะกฎธุรกิจของกุญแจก่อน ซึ่งการกำหนดกฎในการเพิ่มข้อมูลนั้นมักจะเป็นผลเมื่อเราทำการเพิ่มข้อมูลในตารางความสัมพันธ์ตัวลูก สำหรับกฎในการแก้ไขและลบข้อมูลมันจะเป็นผลเมื่อกระทบกับตารางความสัมพันธ์ตัวแม่

6. การเพิ่มแอตตริบิวที่ไม่ใช่กุญแจเพื่อเป็นรายละเอียดของตารางความสัมพันธ์

ให้การกำหนดแอตตริบิวอื่น ๆ ในตารางความสัมพันธ์ทำได้โดยตั้งชื่อและเพิ่มในโมเดลข้อมูลเชิงตรรกะพร้อมทั้งบันทึกลงในพจนานุกรมข้อมูลด้วย จากขั้นตอนการทำโมเดลเชิงตรรกะที่ผ่านมาแอตตริบิวที่เรารู้จักก็คือกุญแจหลัก กุญแจรอง และกุญแจภายนอก ขั้นนี้ก็จะเป็นการกำหนดแอตตริบิวอื่น ๆ ที่ไม่ใช่กุญแจเพิ่มเข้าไป โดยแอตตริบิวแต่ละตัวที่จะเพิ่มนั้นจะต้องขึ้นกับกุญแจทั้งหมดของตารางความสัมพันธ์นั้น ไม่ใช่ขึ้นกับบางส่วนของกุญแจ โดยต้องขยายในตารางความสัมพันธ์ตัวแม่ ไม่ใช่ขยายในตารางความสัมพันธ์ที่มีกุญแจภายนอก นอกจากนั้นถ้าเกิดแอตตริบิวดังกล่าวขึ้นกับกุญแจหลักทั้งหมดแล้ว แต่มีค่ามากกว่าหนึ่งค่า (Multi-valued) ให้แตกออกเป็นอีกตารางความสัมพันธ์ที่มีความสัมพันธ์กับตารางความสัมพันธ์เดิมแบบหนึ่งต่อกลุ่ม จากนั้นให้บันทึกชื่อและรายละเอียดอื่น ๆ ใส่ในโมเดลและพจนานุกรมข้อมูล

การพิจารณาว่าสิ่งใดเป็นแอตตริบิวจริงหรือไม่นั้น บางครั้งยุ่งยากแต่มีข้อสังเกตคือเราจะถือว่าเป็นแอตตริบิวเมื่อไม่มีรายละเอียดปลีกย่อย สำหรับแอตตริบิวบางตัวเมื่อพิจารณาแล้วพบว่าจะขยายความสัมพันธ์ ก็ให้แตกความสัมพันธ์นั้นออกมาตั้งเป็นตารางความสัมพันธ์ใหม่ ที่มีความสัมพันธ์กับตารางความสัมพันธ์เดิมเป็นแบบหนึ่งต่อกลุ่ม สิ่งที่ควรหลีกเลี่ยงคือการใช้รหัสแทนแอตตริวบิว แต่ถ้าจำเป็นก็ให้กำหนดรหัสอย่างเป็นอิสระจากกันโดยไม่รวมความหมายมากกว่าหนึ่งอย่างไว้ในรหัสเดียวกัน ในกรณีที่แอตตริบิวใด ๆ สามารถหาได้จากสูตรหรือการคำนวณได้จากแอตตริบิวอื่นจะเรียกว่าเป็นดิไรฟ์วแอตตริบิว(Derived Attribute) ให้ระบุตัวดี(d)ในโมเดลข้อมูลด้วย ในกรณีที่ระบุ ซับไทป์-ซุปเปอร์ไทป์ ให้ระบุตัวเอส(s)ในโมเดลเช่นกัน

นอกจากการกำหนดแอตตริบิวแล้ว ยังให้พิจารณาเชื่อมตารางความสัมพันธ์ที่มีลักษณะดังต่อไปนี้เข้าด้วยกันคือ

6.1 มีกุญแจหลักเหมือนกันและให้ความหมายอันเดียวกัน

6.2 เป็นซับไทป์ด้วยกันและมีแอตตริบิวทุกตัวเหมือนกัน

6.3 เป็นซัป-ซุปเปอร์ไทป์ ซึ่งซัปไทป์เป็นตัวขยายซูเปอร์ไทป์

6.4 ไม่มีแอตตริบิวอื่นที่ไม่ใช่กุญแจให้ยุบรวมกับตารางความสัมพันธ์ที่มีความสัมพันธ์ต่อกันเสีย

7. การตรวจสอบมุมมองของผู้ใช้ด้วยกฎของนอร์มัลไลเซชัน

ให้ทำการตรวจสอบตารางความสัมพันธ์ต่าง ๆให้อยู่ในกฎของนอร์มัลไลเซชันซึ่งประกอบด้วย 1NF 2NF 3NF BCNF 4NF 5NF ประโยชน์ก็คือ

7.1 ลดที่ว่างที่ต้องใช้ในการการเก็บข้อมูล

7.2 ลดความผิดพลาดความไม่ตรงกันของข้อมูลในฐานข้อมูล

7.3 ลดการเกิดอะนอร์มัลไลของการลบและแก้ไขข้อมูล

ในกรณีได้ตารางความสัมพันธ์ที่เป็นนอมัลไลเซชันที่สมบูรณ์แล้ว สิ่งที่ต้องระวังคือไม่แตกตารางความสัมพันธ์นั้นย่อยลงไปอีก

8. พิจารณาขอบเขตค่าโดเมนของแอตตริบิว

8.1 กำหนดโดเมนของแอตตริบิวทุกตัวในตารางความสัมพันธ์ แล้วบันทึกในพจนานุกรมข้อมูลโดเมน คือกลุ่มค่าที่ถูกต้องเป็นไปได้สำหรับแอตตริวบิวแต่ละตัว อันได้แก่

8.1.1 ชนิดของข้อมูล(Data Type) เช่น จำนวนเต็ม วันที่ ตัวอักษร

8.1.2 ความยาว(Length) เช่น 5 หลัก, 35 ตัวอักษร

8.1.3 รูปแบบของข้อมูล(Format) เช่น dd/mm/yy (วันที่)

8.1.4 ค่าที่อนุญาติให้มีได้(Allowable value) เช่น เป็นได้เฉพาะวันศุกร์

8.1.5 ขอบเขตของข้อมูลหรือข้อกำหนดอื่น ๆ(Range, Constraints)

8.1.6 ความหมาย(Meaning) อธิบายความหมายของแอตตริบิว

8.1.7 ความเป็นหนึ่งเดียว(Uniqueness) ต้องมีค่าเป็นหนึ่งเดียว

8.1.8 ความเป็นค่าว่าง(Null support) อนุญาตให้เป็นค่าว่างได้หรือไม่

8.1.9 ค่าโดยปริยาย (Default value) เช่นกำหนดให้มีค่าเป็น 0

8.2 กฎพิเศษสำหรับโดเมนของแอตตริบิวบางพวก

8.2.1 แอตตริบิวของกุญแจหลักจะต้องมีความเป็นหนึ่งเดียว และห้ามไม่ให้เป็นค่าว่างในกรณีที่กุญแจหลักเป็น โดยที่แอตตริบิวแต่ละตัวที่มาประกอบกุญแจหลักไม่จำเป็นต้องมีค่าเป็นหนึ่งเดียว

8.2.2 แอตตริบิวของกุญแจรอง มีกฎเหมือนกุญแจหลัก เว้นแต่เป็นค่าว่างได้

8.2.3 กลุ่มของโดเมนที่กำหนดให้กับกุญแจภายนอก จะต้องเหมือนกับกลุ่มโดเมนของกุญแจหลักในตารางความสัมพันธ์แม่

8.2.4 แอตตริบิวที่เป็นกุญแจหลักของตารางความสัมพันธ์ซับไทป์ จะต้องมีค่าเป็นซับไทป์ของกุญแจหลักของตารางความสัมพันธ์ซูเปอร์ไทป์

9. พิจารณากฎธุรกิจของแอตตริบิว

9.1 กฎธุรกิจ เป็นกฎที่มีเพื่อตรวจสอบความถูกต้องและควบคุมผลกระทบเมื่อเราทำการสืบค้น เพิ่มเติม แก้ไข และลบ เราจะพิจารณาผลรวมทั้งที่เกิดกับตารางความสัมพันธ์อื่นและแอตตริบิวอื่น ภายในตารางความสัมพันธ์ที่เรากระทำด้วย เมื่อกำหนดกฎการจัดการต่าง ๆ แล้วให้เก็บลงในพจนานุกรมข้อมูล ซึ่งประกอบไปด้วยส่วนต่างๆดังนี้

9.1.1 กฎของผู้ใช้(User rule) เป็นข้อความอธิบายข้อกำหนดการดำเนินงานตามความต้องการของผู้ใช้

9.1.2 เหตุการณ์ของกระตุ้นดำเนินการ(Event) ได้แก่ การเพิ่ม(insert) การแก้ไข(update) การลบ(delete) หรือการสืบค้น(retrive)

9.1.3 ส่วนประกอบของเหตุการณ์(Object of event) ได้แก่ชื่อของตารางความสัมพันธ์หรือแอตตริบิวที่มีการเปลี่ยนแปลงหรือถูกสืบค้นในเหตุการณ์นั้นๆ

9.1.4 เงื่อนไขภายใต้กระตุ้นดำเนินการ(Condition)

9.1.5 การกระทำ(Action) เช่นไม่ยอมรับเหตุการณ์ หรือกระตุ้นที่เกี่ยวกับเหตุการณ์นั้น

9.2 เรากำหนดกฎการจัดการเพื่อ

9.2.1 ความสมบูรณ์ถูกต้องและเป็นอันหนึ่งอันเดียวกันของค่าของแอตตริบิว

9.2.2 กำหนดกฎการจัดการสำหรับทุกแอตตริบิวที่เป็นตัวต้นของดิไรฟว์แอตตริบิวเช่นเมื่อมีการแก้ไขค่าของแอตตริบิวต้นกำเนิดต้องมีกระตุ้นดำเนินการในการแก้ไขแอตตริบิวดีไรฟว์ด้วย

9.2.3 กำหนดกฎการจัดการสำหรับ ซับไทป์-ซุเปอร์ไทป์ โดยถ้าซับไทป์ถูกลบต้องลบซุปเปอร์ไทป์ด้วย

9.2.4 กำหนดกฎการจัดการโดยกำหนดด้วยเวลา เช่นถ้าระเบียบข้อมูลนี้มีอายุเกินหนึ่งปีให้ทำการลบทิ้ง

ขั้นตอนต่าง ๆ ที่ผ่านมา เราได้ทำการพัฒนาจนได้มุมมองของผู้ใช้ที่มีรายละเอียดสมบูรณ์ ขั้นตอนต่อจากนี้จะเป็นการปรับแต่งมุมมองต่าง ๆเข้าด้วยกัน เพื่อขจัดส่วนที่ซ้ำซ้อนและแก้ปัญหาความไม่ตรงกันของข้อมูลซึ่งเป็นขั้นตอนที่ต้องใช้ทั้งประสบการณ์และทักษะในการวิเคราะห์มาก และนับว่าเป็นส่วนที่สำคัญที่สุด

10. การเชื่อมมุมมองของผู้ใช้เข้าด้วยกันเป็นแบบเบ็ดเตล็ด

พิจารณาส่วนของวิวของผู้ใช้ที่คาบเกี่ยวกัน พยายามลดความเหลือเฟือ ตัดความซ้ำซ้อนและแก้ปัญหาความไม่ถูกต้องตรงกันของข้อมูล โดยอาจมีการเพิ่มความสัมพันธ์หรือกฎทางธุรกิจใหม่ ๆ ขึ้นด้วย

10.1 สิ่งที่ควรพิจารณาในการรวมตารางความสัมพันธ์

10.1.1 การรวมตารางความสัมพันธ์ที่มีกุญแจหลักตัวเดียวกัน และค่าที่เป็นไปได้ของกุญแจหลักเหมือนกัน จะต้องได้ตารางความสัมพันธ์ใหม่ที่แอตตริบิวรวมของสองตารางความสัมพันธ์เดิม

10.1.2 ถ้าตารางความสัมพันธ์สองอันมีกุญแจหลักเดียวกัน แต่ค่าที่เป็นไปได้ของกุญแจหลักนั้นเป็นซับเซตของกันและกัน เราจะรวมได้ในรูปแบบของ ซับไทป์-ซูเปอร์ไทป์ โดยต้องตัดแอตตริบิวที่มีแล้วในตารางความสัมพันธ์ซูเปอร์ไทป์ออกจากตารางความสัมพันธ์ที่เป็นซับไทป์

10.1.3 ถ้าตารางความสัมพันธ์สองอันมีกุญแจหลักเดียวกันแต่มีผลไปกำหนดแอตตริบิวที่ต่างกันบางตัว ให้กำหนดซูเปอร์ไทป์ขึ้นมาอันหนึ่งสัมพันธ์กันของตารางความสัมพันธ์เดิม

10.1.4 การเชื่อมตารางความสัมพันธ์สองตัวที่กุญแจหลักของตัวหนึ่งเป็นกุญแจรองของอีกตัวหนึ่ง จะได้ตารางความสัมพันธ์ใหม่ ที่มีกุญแจหลักตามตารางความสัมพันธ์ตัวหนึ่งส่วนกุญแจหลักของตารางความสัมพันธ์อีกตัวจะกลายเป็นกุญแจรองไป และมีแอตตริบิวรวมระหว่างสองตารางความสัมพันธ์เดิม แต่ตัดแอตตริบิวที่ซ้ำซ้อนออกเสีย และต้องกำหนดกฎธุรกิจให้ด้วยตามข้อบังคับเก่า เช่น ในกรณีกุญแจหลักเดิมที่กลายมาเป็นกุญแจรองในตารางความสัมพันธ์ใหม่ก็ต้องยังคงห้ามเป็นค่าว่าง

10.1.5 การรวมตารางความสัมพันธ์ใด ๆก็ตาม ต้องไม่มีผลไปเปลี่ยนแปลงตารางความสัมพันธ์อื่น ๆ ที่ไม่ได้เกี่ยวข้อง

10.2 สิ่งที่ควรพิจารณาในการรวมความสัมพันธ์

10.2.1 ให้รวมความสัมพันธ์ระหว่างตารางความสัมพันธ์ที่ให้ความหมายเหมือนกันเข้าด้วยกัน โดยถ้าผลทำให้เกิดเป็นความสัมพันธ์แบบกลุ่มต่อกลุ่ม จะต้องทำการแตกให้เป็นความสัมพันธ์แบบหนึ่งต่อกลุ่มสองอัน

10.2.2 การรวมความสัมพันธ์ใด ๆ ต้องไม่ไปกระทบกับความสัมพันธ์อื่นที่ไม่ต้องการเปลี่ยนแปลง นอกจากจะพิจารณาแล้วว่าควรตัดออกเนื่องจากซ้ำซ้อน หรืออาจเพิ่มขึ้นใหม่เพื่อความเหมาะสม

10.2.3 จากการรวมตารางความสัมพันธ์ที่มีกุญแจหลักเป็นกุญแจรองของตารางความสัมพันธ์อีกตัว ให้ตรวจสอบกุญแจภายนอกของตารางความสัมพันธ์อื่น ๆ ที่อ้างมาถึงว่าได้อ้างถึงกุญแจหลักหรือกุญแจรองของตารางความสัมพันธ์ใหม่ที่ได้จากการรวมนั้น ถ้าอ้างถึงกุญแจรองต้องทำการเปลี่ยนให้เป็นกุญแจหลัก

10.2.4 เมื่อรวมมุมมองแล้วให้กำหนดกฎธุรกิจของกุญแจสำหรับความสัมพันธ์ใหม่ด้วย

10.3 สิ่งที่ควรพิจารณาในการรวมแอตตริบิว

10.3.1 ให้ทำการรวมแอตตริบิว ที่มีความหมายเหมือนกันภายในตารางความสัมพันธ์เดียวกัน และรวมค่าที่เป็นไปได้รวมถึงกฎการจัดการเข้าด้วยกันด้วย และให้พิจารณาค่าที่เป็นไปได้ของแอตตริบิวอื่นได้ด้วยว่าเปลี่ยนไปหรือไม่

10.3.2 เมื่อรวมตารางความสัมพันธ์แล้ว ให้พิจารณาแอตตริบิวที่เป็นดิไรฟว์หรือตัวบ่งชี้(Flag)ที่ไม่จำเป็นทิ้งเสีย

10.3.3 หลังจากได้ รวม ตัด หรือเพิ่ม ความสัมพันธ์แล้ว ให้ทำการนอร์มัลไลซ์อีกครั้งเพื่อตัดสิ่งที่ซ้ำซ้อนออกเสีย

11. การรวบรวมโมเดลข้อมูลที่มีอยู่เข้าด้วยกัน

จะเป็นขั้นที่เกี่ยวข้องกับระดับมโนภาพ โดยจะรวมโมเดลข้อมูลเชิงตรรกะที่ได้กับของที่มีอยู่แล้วเดิม ให้พัฒนาโมเดลใหม่ควบคู่ไปกับการพิจารณากฎเกณฑ์ข้อบังคับของเดิม โดยอาจมีการใช้ตารางความสัมพันธ์ หรือความสัมพันธ์ร่วมกับของเดิม และมีการกำหนดตารางความสัมพันธ์ขึ้นมาใหม่ด้วย

12. การวิเคราะห์ความมีเสถียรภาพและการเติบโตในอนาคต

การออกแบบโมเดลที่ทำผ่านมาจะพิจารณาข้อมูลที่เห็นได้ในปัจจุบันเป็นส่วนมากสำหรับขั้นนี้ให้พิจารณาถึงสิ่งที่อาจเกิดขึ้น หรือเป็นไปได้ในอนาคตด้วย เช่น

12.1 อาจมีตารางความสัมพันธ์ หรือ ความสัมพันธ์ใหม่เกิดขึ้น ทำให้ต้องเพิ่มกุญแจภายนอกในตารางความสัมพันธ์ของเดิม

12.2 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม อาจกลายเป็นความสัมพันธ์แบบกลุ่มต่อกลุ่มได้

12.3 กุญแจหลักอาจเปลี่ยนไปเนื่องจากของเดิมไม่เป็นหนึ่งเดียวแล้วสิ่งเหล่านี้เมื่อเราพิจารณาแล้วอาจทำการดัดแปลงโมเดลไว้เพื่อรองรับหรือจบันทึกเก็บไว้ก่อนเฉยก็ได้

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

เป็นขั้นตอนการแปลงจากโมเดลข้อมูลเชิงตรรกะ เข้าสู่ระบบฐานข้อมูลเชิงสัมพันธ์ (Transaction Process) อย่างมีเสถียรภาพ ภายใต้ระบบจัดการฐานข้อมูลที่ติดตั้งบนเครื่องคอมพิวเตอร์ เป็นการปรับให้เข้ากับหน้าที่ใช้สอยเฉพาะ และตามความต้องการในเรื่องประสิทธิภาพการทำงาน ประกอบด้วยสองส่วนคือ

1. โครงสร้างข้อมูล (Data Structure)

บอกที่อยู่ของโครงสร้างข้อมูล ได้แก่ตารางความสัมพันธ์ แอตตริบิว และความสัมพันธ์ ด้วยการระบุตารางความสัมพันธ์และสดมภ์ ดังนี้

1.1 ตารางความสัมพันธ์เทียบเท่ากับตารางความสัมพันธ์ และสดมภ์เทียบเท่าแอตตริบิวในโมเดลข้อมูลเชิงตรรกะ โดยต้องคำนึงว่าแต่ละตารางและสดมภ์ที่พบในระบบจัดการฐานข้อมูลเชิงสัมพันธ์ต้องไม่มีความขัดแย้งกัน

สดมภ์ คือ หน่วยหนึ่งของข้อมูลที่สามารถกำหนดตำแหน่งที่อยู่ได้อาจเรียกอีกอย่างว่าเขตข้อมูล

ตาราง คือ กลุ่มของสดมภ์ โดยสดมภ์ทุกตัวในตารางเป็นตัวพรรณนาถึงลักษณะคุณสมบัติของตารางนั้นๆ

1.2 บางกรณีอาจมีการรวมตารางเข้าเป็นตารางเดียว หรือแบ่งตารางออกเป็นหลายตารางเพื่อประโยชน์ในแง่การทำงานที่รวดเร็วขึ้น

2. ความบูรณภาพของข้อมูล (Data Integrity)

จากแผนภาพโมเดลข้อมูลเชิงตรรกะสามารถควบคุมความบูรณภาพของข้อมูลด้วยกฎธุรกิจ ซึ่งได้แก่พารามิเตอร์ระบุตารางความสัมพันธ์ เช่น การสร้างเพื่อระบุความเป็นเอกลักษณ์ (Unique Index) เป็นต้น

ความบูรณภาพของข้อมูลเป็นการรักษาคุณภาพและความน่าเชื่อถือของข้อมูลในฐานข้อมูลของระบบสารสนเทศ และป้องกันมิให้ข้อมูลเกิดความผิดพลาด พิจารณาจากระบบจัดการฐานข้อมูลพบว่าปัจจุบันไม่ได้เอื้ออำนวยในการควบคุมทางด้านนี้โดยตรง กล่าวคือภาษาเอสคิวแอล(SQL Language) ซึ่งเป็นภาษาในยุคที่ 4 (4th Generation Language, 4GL) เป็นคำสั่งแบบไม่ใช่วิธีดำเนินการ(Nonprocedural Syntax) ใช้ในการสร้าง การเก็บ การเปลี่ยนแปลง การสืบค้น และการดูแลรักษา สารสนเทศในฐานข้อมูล ไม่อาจระบุคำสั่งเพื่อให้เกิดบูรณภาพของข้อมูลได้โดยตรง จึงจำเป็นต้องกำหนดกฎเพื่อความบูรณภาพของข้อมูลแบ่งเป็น 3 ประเภทดังนี้

2.1 กฎความบูรณภาพของตารางความสัมพันธ์ (Entity Integrity Rule) กำหนดให้ส่วนประกอบของกุญแจหลัก เป็นตัวระบุความเป็นเอกลักษณ์ภายในแต่ละระเบียนของตารางความสัมพันธ์ซึ่งจะต้องมีค่าเสมอ ในขั้นตอนการเพิ่ม ปรับปรุง หรือลบ จำเป็นต้องคงไว้ซึ่งคุณสมบัติของกุญแจหลัก

2.2 กฎความบูรณภาพในการอ้างอิง (Referential Integrity Rule) เป็นการหาแหล่งที่มาของกุญแจนอก ถ้าตารางความสัมพันธ์ใดมีกุญแจนอกปรากฏอยู่ทุกค่าของกุญแจนอกอาจเป็นไปได้ 2 กรณีคือ

2.2.1 ค่าว่าง(Null Value)

2.2.2 มีค่าที่สอดคล้องกับค่ากุญแจหลักในตารางความสัมพันธ์อื่น ดังนั้นต้องมีการระบุถึงการอ้างอิงระหว่างตารางความสัมพันธ์ด้วยกัน เพื่อตรวจสอบค่าของกุญแจนอก ว่าตรงกับค่าที่มีอยู่จริงในกุญแจหลักหรือไม่ ในการดำเนินการเพิ่ม ปรับปรุง หรือลบ อาจกล่าวอีกนัยหนึ่งว่าเป็นการควบคุมความบูรณภาพของกุญแจนอกก็ได้

2.3 กฎความบูรณภาพของโดเมน (Domain Integrity Rule) เป็นการควบคุมความถูกต้องสำหรับทุกๆสดมภ์ที่อยู่ในตารางความสัมพันธ์ 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

RDD1 ระบุตารางความสัมพันธ์จากตารางความสัมพันธ์

การกำหนดตาราง ในการสร้างฐานข้อมูลเชิงสัมพันธ์ จะแทนหนึ่งตารางความสัมพันธ์จากโมเดลข้อมูลเชิงตรรกะด้วยหนึ่งตารางเท่านั้น

RDD2 ระบุสดมภ์ในตารางความสัมพันธ์ สำหรับแต่ละแอตตริบิวที่ขยายคุณสมบัติของตารางสัมพันธ์ในฐานข้อมูลเชิงสัมพันธ์นี้ จะเห็นว่าแถวในตารางหนึ่งมีความหมายเหมือนกับระเบียนของตารางความสัมพันธ์ที่ถูกแปลงไปเป็นตารางนั้น ในการกำหนดสดมภ์ของตารางนั้นจะใช้แอตตริบิวในตารางความสัมพันธ์นั้นโดยแทนหนึ่งแอตตริบิวด้วยหนึ่งสดมภ์

RDD3 ดัดแปลงโครงสร้างให้เหมาะสมกับสภาพแวดล้อมของ ระบบจัดการฐานข้อมูลเชิงสัมพันธ์และปรับปรุงในเรื่องประสิทธิภาพการทำงานด้วย

RDD4 ออกแบบกฎธุรกิจที่เกี่ยวข้องกับตารางความสัมพันธ์

1. บังคับคุณสมบัติทางตรรกะของกุญแจหลัก ในการติดตั้งระบบแบบความสัมพันธ์ เช่น ความเป็นเอกลักษณ์ไม่อนุญาตให้ค่าในกุญแจหลักไม่มีค่า เป็นต้น

2. บังคับคุณสมบัติทางตรรกะของกุญแจรอง ในการติดตั้งระบบแบบความสัมพันธ์ เช่น ความเป็นเอกลักษณ์ เป็นต้น

RDD5 ออกแบบกฎธุรกิจที่เกี่ยวข้องกับความสัมพันธ์ ข้อจำกัดในการเพิ่ม ลบ และแก้ไขแอตตริบิวที่เป็นกุญแจในลักษณะการอ้างอิงให้สอดคล้อง

RDD6 ออกแบบกฎธุรกิจเพิ่มเติมเกี่ยวกับแอตตริบิว ได้แก่ ทริกเกอร์ดำเนินการ ควบคุมค่าโดเมน ชนิดของข้อมูล รูปแบบข้อมูล ค่าที่ยอมรับ(ขอบเขตความต่อเนื่องของข้อมูล) ความเป็นเอกลักษณ์ อนุญาตให้มีค่าได้หรือไม่และค่าปริยาย การปรับปรุงโมเดลข้อมูลใหม่เมื่อมีสารสนเทศใหม่เพิ่มเติม อาจทำให้เกิดการแบ่งแยก(Splitting) หรือการยุบรวม (Marging)ของตารางความสัมพันธ์ขึ้น โดยพิจารณาว่าแอตตริบิวเดิมควรจะถูกย้ายจากตารางความสัมพันธ์เดิมไปสู่ตารางความสัมพันธ์ใหม่หรือไม่ จะต้องคำนึงถึงสภาพความเป็นจริง เช่น

1. แอตตริบิวมีความสัมพันธ์กับแอตตริบิวอื่นอย่างไร
2. การสร้างตารางความสัมพันธ์ใหม่สามารถแก้ปัญหาการสืบค้นได้หรือไม่
3. ข้อเท็จจริงของแอตตริบิวอ้างอิงถึงแอตตริบิวอื่นหรือไม่
4. ตารางความสัมพันธ์ใหม่สัมพันธ์กับตารางความสัมพันธ์เดิม และตารางความสัมพันธ์อื่น ๆ ในโมเดลอย่างไร

<u>การจัดกลุ่มในระดับตารางความสัมพันธ์</u> (Entity Clustering)

ตั้งอยู่บนพื้นฐานของขอบเขตข้อมูล หมายความว่าข้อมูลทั้งหมดที่จำเป็นต้องการใช้ภายในหน่วยงาน โดยแต่ละหน่วยงานมีหน้าที่รับผิดชอบในการจัดกลุ่มตารางความสัมพันธ์ เนื่องจากเป็นเจ้าของข้อมูล โครงสร้างข้อมูลภายในองค์การที่ปรากฏอยู่จะเป็นตัวบ่งชี้ว่าตารางความสัมพันธ์นี้ควรจะปรากฏในหน่วยงานใดบ้าง การจัดกลุ่มอาจเป็นกลุ่มย่อยของกลุ่มใหญ่รวมกันหลาย ๆกลุ่มก็ได้ ขึ้นกับระดับความละเอียดที่ต้องการแสดง เช่น ในระดับบนโมเดลข้อมูลที่ได้จากการจัดกลุ่มมักแสดงเฉพาะตารางความสัมพันธ์หลักที่มีส่วนเชื่อมโยงกับตารางความสัมพันธ์อื่น โดยความสัมพันธ์ของข้อมูลทุกหน่วยงานภายในองค์กรโดยสังเขป ส่วนในระดับล่างลงมา อาจแสดงกลุ่มตารางความสัมพันธ์ที่สนใจ เฉพาะหน่วยงานเท่านั้นในกรณีที่หน่วยงานใดมีข้อมูลที่ซับซ้อนหรือมากมายก็อาจจัดกลุ่มย่อยในระดับล่างลงมาอีกได้ การจัดกลุ่มตารางความสัมพันธ์อาจเกิดได้ในระหว่างการออกแบบฐานข้อมูลทั้งสองขั้นตอน ได้แก่ ขั้นตอนเชิงตรรกะ และขั้นตอนเชิงกายภาพ

เนื่องจากโมเดลข้อมูลประกอบด้วยตารางความสัมพันธ์มากมาย การจัดกลุ่มของตารางความสัมพันธ์ในขั้นตอนเชิงตรรกะในแต่ละเอนทิตี้มีการจัดกลุ่มของแอตตริบิว เพื่อขยายคุณสมบัติต่าง ๆ ขึ้นกับธรรมชาติของข้อมูลที่ปรากฏอยู่และโครงสร้างข้อมูล แต่ขั้นตอนเชิงกายภาพไม่เป็นเช่นนั้น เป็นการจัดกลุ่มตารางความสัมพันธ์เพื่อตอบสนองวิวของผู้ใช้ โดยต้องคำนึงถึงการดำเนินการและความต้องการในการเข้าถึงฐานข้อมูล

การจัดกลุ่มในระดับตารางความสัมพันธ์ต้องพิจารณาถึง

1. ขอบเขตของข้อมูลที่เกี่ยวข้อง
2. โครงสร้างข้อมูลที่ปรากฏอยู่
3. เป็นการแสดงแทนวิวของผู้ใช้เฉพาะบริเวณ โดยปรากฏเฉพาะกลุ่มของตารางความสัมพันธ์ที่สนใจเท่านั้น
4. การสอบถามข้อมูลว่าครอบคลุมถึงตารางความสัมพันธ์ตัวใดบ้าง
5. การประมวลผลที่ต้องการจากผู้ใช้

<u>การนอร์มัลไลซ์</u> (Normalized)

วิธีการ normalization เป็นเทคนิคที่มีประโยชน์มากในการสร้างระบบฐานข้อมูล แต่เทคนิคนี้จะมีข้อด้อยอยู่บ้าง โดยเฉพาะด้านการจัดการระบบควบคุม I/O หากมีการวางแผนไม่ดีพอ ในกรณีที่ต้องการเรียกดูข้อมูลที่จัดเก็บแยกกันคนละแฟ้ม การจัดการด้านการอ่านและเขียนก็จะเกิดขึ้นบ่อยครั้ง ซึ่งจะมีผลกระทบต่อประสิทธิภาพของการประมวลผลของระบบไม่ว่าจะเป็นการทำงานแบบโต้ตอบ(Interactive) หรือการประมวลผลแบบยาวนาน(Batch) นั่นคือจะเสียเวลาในการทำงานนานขึ้น ผู้ออกแบบระบบจะต้องหาสมดุลที่เหมาะสมระหว่างการทำ normalization ของฐานข้อมูล กับการทำงานของระบบโดยรวม

เป้าหมายที่สำคัญของการใช้ระบบฐานข้อมูลเชิงสัมพันธ์ก็ คือการออกแบบความสัมพันธ์ เพื่อที่จะสามารถใช้ใช้ข้อมูลในระบบได้อย่างสะดวก และมีประสิทธิภาพโดยพยายามให้เกิดความซ้ำซ้อนในการเก็บข้อมูลน้อยที่สุด ซึ่งถือว่าเป็นพื้นฐานของการออกแบบฐานข้อมูลในการทำให้ฐานข้อมูลอยู่ในรูปของ "normal form" โดยอาศัยความรู้เกี่ยวกับฟังก์ชั่นการขึ้นต่อกัน(functional dependency) ซึ่งการทำนอร์มอลไลเซชั่นนั้นสามารถแบ่งออกได้หลายระดับ ดังนี้ (ดวงแก้ว สวามิภักดิ์, 2521)

1. <u>นอร์มอลไลเซชั่นระดับที่ 1</u> (First normal form)

เป็นการขจัดสดมภ์ให้ไปเป็นระเบียนใหม่ เพื่อให้แต่ละรายการในตารางความสัมพันธ์ไม่มีค่าในสดมภ์ที่ซ้ำกัน ซึ่งเป็นการปรับจากความสัมพันธ์ที่ไม่นอร์มัล ดังตัวอย่างแสดงดัง รูปที่ 2.4

Order No.	Order date	Item no.	Quantity
12489	020931	AX12	11
12491	020931	BT04	
		BZ66	1
12494	040931	CB03	4
12495	040931	CX11	
12498	050931	AZ52	2
		BA7	4
12500	050931	BT04	1
12504	050931	CZ81	2

รูปที่ 2.4 ตัวอย่าวความสัมพันธ์ที่ไม่นอร์มัล

Order No.	Order date	Item no.	Quantity
12489	020931	AX12	11
12491	020931	BT04	
12491	020931	BZ66	1
12494	040931	CB03	4
12495	040931	CX11	
12498	050931	AZ52	2
12498	050931	BA7	4
12500	050931	BT04	1
12504	050931	CZ81	2

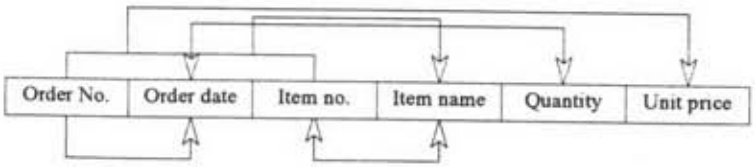
รูปที่ 2.5 ผลจากการนอร์มอลไลซ์ขั้นที่ 1

2. <u>นอร์มอลไลเซชั่นระดับที่ 2</u> (Second normal form)

เป็นการขจัดสดมภ์ที่ไม่ได้ขึ้นตรงกับสดมภ์ที่เป็นกุญแจหลักออกไป เพื่อให้ได้สดมภ์ทั้งหมดขึ้นตรงกับกุญแจหลักเท่านั้น วิธีการที่จะทำให้ความสัมพันธ์อยู่ในรูปของนอร์มอลไลซ์ขั้นที่ 2 กระทำได้โดยการสร้างความสัมพันธ์ขึ้นมาใหม่สำหรับกรณีที่การขึ้นต่อกันเป็นปัญหา โดยที่ใน รูปที่ 2.6 จะเป็นความสัมพันธ์ที่ผ่านการนอร์มอลไลซ์ขั้นที่หนึ่งมาแล้ว

Order No.	Order date	Item no	Item name	Quantity	Unit price
12489	020931	AX12	Micro wave	11	12000
12491	020931	BT04	Stove		1500
12491	020931	BZ66	Refigurator	1	8500
12494	040931	CB03	Motorcycle	4	25000
12495	040931	CX11	Foot ball		350
12498	050931	AZ52	Sket bord	2	1600
12498	050931	BA7	Ball	4	450
12500	050931	BT04	Stove	1	1700
12504	050931	CZ81	Base ball	2	1400

รูปที่ 2.6 ตัวอย่างความสัมพันธ์



รูปที่ 2.7 ผังแสดงการขึ้นต่อกัน

ดังนั้นเราจะสร้างความสัมพันธ์ใหม่อีก 2 ตัว โดยให้รหัสการสั่งเป็นกุญแจในความสัมพันธ์ที่ 1 และให้รหัสสินค้าเป็นกุญแจในอีกความสัมพันธ์หนึ่ง ผลที่ได้จากการทำนอร์มอลไลซ์ขั้นที่ 2 แสดงดัง รูปที่ 2.7

Order No.	Order date
12489	020931
12491	020931
12491	020931
12494	040931
12495	040931
12498	050931
12498	050931
12500	050931
12504	050931

Order table

Item no.	Item name
AX12	Micro wave
BT04	Stove
BZ66	Refigurator
CB03	Motorcycle
CX11	Foot ball
AZ52	Sket bord
BA74	Ball
BT04	Stove
CZ81	Base ball

Item tabl

Order No.	Item no.	Quantity	Unit price
12489	AX12	11	12000
12491	BT04	1	1500
12491	BZ66	1	8500
12494	CB03	4	25000
12495	CX11	2	35
12498	AZ52	2	1600
12498	BA74	4	45
12500	BT04	1	1700
12504	CZ81	2	1400

Order list table

รูปที่ 2.8 ผลจากการนอร์มอลไลซ์ขั้นที่ 2

3. นอร์มอลไลเซชั่นระดับที่ 3 (Third normal form)

เป็นการขจัดสดมภ์ที่เป็นกุญแจที่ไม่ได้ขึ้นตรงกับกุญแจหลักออกไป เพื่อไม่ให้มีสดมภ์ที่เป็นกุญแจที่ไม่ได้ขึ้นตรงต่อกุญแจหลัก