บทที่ 3

แนวเหตุผล และ ทฤษฎีที่สำคัญ

การออกแบบฐานข้อมูล

การจัดการกับข้อมูลในสำนักการศึกษา ในสังกัดกรุงเทพมหานครในระบบเดิมมีการเก็บรวบรวมข้อมูลในรูปแบบแฟ้มข้อมูลแบบคอนเวนชันนัล (Conventional) ซึ่งมีโปรแกรมประยุกต์แต่ละโปรแกรมจัดการกับแฟ้มข้อมูลของตัวเอง ดังนั้นในองค์กรเหล่านี้ก็ไม่จำเป็นที่จะต้องใช้ข้อมูลร่วมกัน แต่ในบางครั้งมีความจำเป็นที่จะต้องใช้ข้อมูลชุดเดียวกันจากโปรแกรมประยุกต์ของแต่ละหน่วยงานที่แตกต่างกัน ก็อาจต้องได้รับข้อมูลที่อยู่ในรูปแบบที่ไม่เหมือนกัน ผลที่เกิดขึ้นก็คือจะมีข้อมูลชุดเดียวกันอยู่ในหลายแฟ้มข้อมูลของแต่ละหน่วยงาน ซึ่งจะก่อให้เกิดปัญหาต่างๆ เช่น

1. เก็บข้อมูลอย่างซ้ำซ้อน (Data Redundancy) ทำให้เปลืองเนื้อที่สื่อในการจัดเก็บข้อมูล เสียเวลาในการทำงานซ้ำซ้อนนั้น และสิ้นเปลืองค่าใช้จ่าย

2. ความไม่คงที่ในการปรับปรุงข้อมูลให้ตรงกัน (Data Inconsistency) ในการเก็บข้อมูลชุดเดียวกันอยู่หลายที่ หรือหลายแฟ้มข้อมูลนั้นอาจเกิดการเปลี่ยนแปลงที่ขัดแย้งกัน (Awad and Gotterer, 1992) เช่น หน่วยงานหนึ่งเปลี่ยนแปลงข้อมูลในแฟ้มหนึ่งแต่หน่วยงานอีกหน่วยงานหนึ่งไม่ได้เปลี่ยนแปลงข้อมูลในแฟ้มนั้น ทั้งๆที่เป็นข้อมูลเดียวกัน ผลก็คือมีความแตกต่างกันของข้อมูลชุดเดียวกัน

3. ไม่มีความคล่องตัวในการเข้าถึงข้อมูล (Difficult in accessing Data) การเรียกดูข้อมูลหรือการออกรายงานจะทำได้รวดเร็ว ถ้าข้อมูลมาจากแหล่งเดียวกัน หากข้อมูลที่ต้องการอยู่แยกกันไปแต่ละหน่วยงานเราไม่สามารถนำข้อมูลมาใช้พร้อมกันได้ด้วยคำสั่งง่าย ๆ

4. ข้อมูลอยู่กันอย่างกระจัดกระจาย (Data Isolation) การที่ข้อมูลอยู่กันอย่างกระจัดกระจายกันคนละแฟ้มนั้น อาจมีการจัดเก็บรูปแบบของแฟ้มข้อมูล (Format) ที่แตกต่างกัน จึงยากแก่การเขียนโปรแกรมประยุกต์ขึ้นมาสำหรับสืบค้น (Retrieve) และทำการประมวลผล ข้อมูลที่เหมาะสมได้

5. ระบบความปลอดภัยของข้อมูล (Data Security)ในระบบแฟ้มข้อมูลแบบคอนเวนชันนัลนั้นบางระบบอนุญาตให้ผู้ใช้ข้อมูลมากกว่า 1 คน เข้าใช้ได้ในขณะเดียวกัน และไม่มีการกำหนดสิทธิในการเข้าใช้ข้อมูล ทำให้ผู้ใช้ที่ไม่มีสิทธิในการ แก้ไขข้อมูลเข้าไปแก้ไขข้อมูลที่สำคัญต่าง ๆ ได้

6. ความไม่มีบูรณภาพของข้อมูล (Data Integrity)ในระบบแฟ้มข้อมูลแบบคอนเวนชันนัลนั้นเมื่อต้องการเพิ่มเงื่อนไขหรือข้อบังคับ (Constrain) ต่าง ๆเพื่อตรวจสอบความถูกต้องของข้อมูล เงื่อนไขหรือข้อบังคับนั้นจะต้องถูกเพิ่มเข้าไปในแฟ้มข้อมูลทุกๆแฟ้ม ถ้าเพิ่มไม่ครบแล้วข้อมูลก็จะไม่มีบูรณภาพ ดังนั้นถ้ามีการเปลี่ยนแปลงหรือมีข้อบังคับที่ซับซ้อนขึ้น ทำให้เราต้องแก้ไขโปรแกรมหลายๆโปรแกรม เนื่องจากเงื่อนไขหรือข้อบังคับเหล่านี้ไม่ได้ผูกติดไปกับข้อกำหนดของข้อมูล ทำให้เสียเวลาเข้าไปแก้ไขมากมาย

7. ขาดความเป็นอิสระของข้อมูล (Data Independency) ในระบบแฟ้มข้อมูลแบบคอนเวนชันนัลนั้น ข้อมูลจะไม่เป็นอิสระ จะผูกพันอยู่กับการจัดเก็บ และการเรียกใช้ข้อมูลเทคนิคในการเขียนโปรแกรมประยุกต์เพื่อใช้เฉพาะงาน ดังนั้นถ้าเกิดมีการเปลี่ยนแปลงวิธีจัดเก็บหรือการเรียกใช้ข้อมูลขึ้น ก็จะต้องเข้าไปแก้ไขโปรแกรมประยุกต์เหล่านั้นตามไปด้วย และถ้ามีการแก้ไขโครงสร้างข้อมูล ก็จะมีผลกระทบไปถึงโปรแกรมประยุกต์เหล่านั้นเช่นเดียวกัน ซึ่งเป็นการไม่สดวกอย่างยิ่งในการใช้งานและพัฒนาโปรแกรม

8. ขาดระบบการฟื้นสภาพของข้อมูล (Data Recoery System)ให้กลับคืนสู่สภาพปกติ เมื่อข้อมูลถูกทำลายหรือเสียหายไป แต่ละหน่วยงานซึ่งมีแฟ้มข้อมูลของตนเองอยู่ก็จะทำการแก้ไขซึ่งต่างคนต่างทำ ดังนั้นการบูรณะข้อมูลของตนให้กลับคืนสู่สภาพปกติจึงไม่มีมาตรฐานในการฟื้นสภาพของข้อมูล

คุณสมบัติที่สำคัญของระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database)

1. มีลักษณะเป็นศูนย์รวมข้อมูลที่สัมพันธ์กันของข้อมูลหน่วยงานต่างๆซึ่งเรียกข้อมูลลักษณะนี้ว่าข้อมูลแบบเมตา (meta-data) (Marti,1983)

2. ตรวจสอบความถูกต้องของข้อมูล (validation check), ตรวจสอบความซ้ำซ้อนของข้อมูล (redundancy check), ตรวจสอบความไม่ขัดแย้งกันของข้อมูล (consistency check), ทำให้ข้อมูลในระบบมีความเป็นบูรณภาพเสมอ (integrity data), ควบคุมความปลอดภัยของข้อมูล (security control) และมีมาตรฐาน (standard) ในข้อกำหนดพร้อมทั้งเงื่อนไขต่างๆของข้อมูลที่ทำการบันทึกลงสู่ระบบ (Van Duyn,1982)

3. มีการควบคุมการอ้างอิงให้สอดคล้องต้องกันเมื่อมีการอ้างอิงถึงข้อมูลส่วนอื่นตรวจสอบได้ว่ามีอยู่จริงและมีค่าถูกต้อง (referential constrains) (Matri,1983)

4. นำไปใช้กับระบบฐานข้อมูลแบบเชิงสัมพันธ์ ใช้แสดงโครงสร้างของสารสนเทศพัฒนาโมเดลเชิงมโนภาพในระดับหน่วยงานหรือบริหารในระดับส่วนกลางได้ (local and global conceptaul data models)

ในระยะเวลาที่ออกแบบและพัฒนาโปรแกรม ข้อมูลที่เก็บรวบรวมต้องวิเคราะห์โดยหน้าที่ (function analysis) และผ่านขบวนการทำให้เป็นบรรทัดฐาน (normalization) (Maddison,1985)

5. การทำงานเป็นแบบเชื่อมตรง (on-line) สามารถโต้ตอบการทำงาน กับผู้ใช้ได้ (interactive) ได้แก่การทำงานในด้าน

5.1 การป้อนข้อมูลเข้าสู่ระบบ (on-line entry)

5.2 การสืบค้นข้อมูลในระบบ (on-line retrieval)

5.3 การปรับปรุงข้อมูลให้ทันสมัย (on-line update)

5.4 การออกรายงานสารสนเทศ (on-line report)

<u>วัตถุประสงค์ของการมีฐานข้อมูล</u>

1. ทำการรวบรวมแฟ้มข้อมูลที่มีความสัมพันธ์ซึ่งกันและกันหลายแฟ้มข้อมูลเข้าไว้ด้วยกัน ซึ่งจะทำให้เกิดความเข้าใจถึงโครงสร้างของข้อมูลที่ได้ทำการออกแบบไว้และใช้ในการตรวจสอบความถูกต้องด้วย

2. ลดความซ้ำซ้อนของข้อมูล

3. เก็บข้อมูลไว้ที่ศูนย์กลางเพื่อควบคุมการใช้ฐานข้อมูลให้มีมาตรฐานเดียวกัน และควบคุมข้อกำหนดหรือข้อบังคับของข้อมูลให้เกิดบูรณภาพเสมอ ถึงแม้จะมีการเปลี่ยนแปลงเกิดขึ้นภายหลังก็ตาม

4. ผู้มีสิทธิเท่านั้นจึงจะสามารถใช้ข้อมูลได้ และใช้ได้เท่าที่ตนเองได้รับสิทธิ

5. สามารถจัดทำรายงานเฉพาะกิจเกี่ยวกับสารสนเทศได้ตามความต้องการของผู้ใช้ เพื่อใช้ในการอ้างอิงได้

การออกแบบและพัฒนาระบบฐานข้อมูลแบบเชิงสัมพันธ์ (Relational Database)

ในสภาพแวดล้อมของฐานข้อมูลนั้นฐานข้อมูลจะถูกออกแบบในทางที่เกี่ยวข้อง และเกี่ยวพันกับการเปลี่ยนแปลงสารสนเทศต่างๆ ที่จะเกิดขึ้นในอนาคตขององค์กรด้วย

การออกแบบฐานข้อมูลที่ดีจะทำให้แหล่งข้อมูลใช้สารสนเทศต่างๆ ได้ทั้งในปัจจุบันและในอนาคต โดยให้ระบบฐานข้อมูลเป็นอิสระต่อการติดตั้งทางกายภาพจริง ๆ และระบบฐานข้อมูลนั้นสามารถถูกเปลี่ยนไปใช้กับฮาร์ดแวร์ใหม่ ๆ ได้ หรือใช้กับระบบจัดการฐานข้อมูลใหม่ก็ได้ซึ่งจะไม่ทำให้แบบจำลองของข้อมูลเชิงตรรกเปลี่ยนแปลงเลย

ระบบฐานข้อมูลสำหรับองค์กรควรมีคุณสมบัติดังต่อไปนี้

1. สะท้อนถึงการปฏิบัติต่างๆ ขององค์กร
2. ยืดหยุ่นรับกับการเปลี่ยนแปลงสารสนเทศใหม่ๆได้
3. สนับสนุนการใช้หลายมุมมอง
4. ไม่ขึ้นกับระบบจัดการฐานข้อมูลใดโดยเฉพาะ
5. อิสระต่อการติดตั้งทางกายภาพ (Physical Imoplementation)

<u>การออกแบบและพัฒนาระบบฐานข้อมูล</u> อาศัยหลักการใหญ่ 3 ข้อ คือ

1. การรวบรวมความต้องการของผู้ใช้ (user view)ผู้ออกแบบควรจะศึกษาทุก ๆ งานที่ผู้ใช้ได้ใช้อยู่ในปัจจุบัน กำหนดข้อมูลเข้า (input) และข้อมูลที่นำออก (output) ตรวจสอบรายงานทุกรายงานที่เป็นระบบปัจจุบัน และสัมภาษณ์ผู้ใช้ ว่าเขาใช้ระบบอย่างไร เมื่อเข้าใจระบบงานปัจจุบันที่ใช้แล้ว ผู้ออกแบบและพัฒนา ควรจะรวบรวมความต้องการของผู้ใช้ว่า เมื่อมีระบบใหม่นั้นเขาต้องการอะไรบ้าง ผู้ออกแบบควรจะคำนึงถึงความต้องการของผู้ใช้ในอนาคตด้วย จากการรวบรวมในขั้นตอนนี้นำมาวิเคราะห์ได้ผลลัพธ์ของการวิเคราะห์ก็คือ โมเดลความต้องการ ของผู้ใช้ และสภาพแวดล้อมของผู้ใช้

2. การออกแบบจำลองข้อมูลเชิงตรรก (Logical Data Modeling : LDM) เป็นการสร้างโครงร่างวิวของผู้ใช้ (sketletal user view)จากความต้องการตามสภาพแวดล้อมของผู้ใช้ ซึ่งแสดงถึงความต้องการสารสนเทศของระบบงานออกมาให้เป็นแผนภาพ เน้นข้อมูลที่มีอยู่จริง ผู้ออกแบบควรพิจารณาว่าจะใช้ฐานข้อมูลนี้อย่างไร รวบรวมวิวต่างๆของงานประยุกต์ที่ผู้ใช้ต้องการ รวบรวมรายการเปลี่ยนแปลงของข้อมูล (Transaction) ปริมาณของรายการเปลี่ยนแปลงข้อมูลในแต่ละวัน เงื่อนไขพิเศษต่างๆในแง่ของการใช้งาน รายละเอียดต่าง ๆ ของ เอนติตี้ (entity) และ แอตตริบิวต์ (attribute) ทั้งนี้โดยไม่ต้องคำนึงถึงรายละเอียดในการติดตั้ง ความเร็วในการสืบค้นข้อมูล และการทำงานทางกายภาพของเครื่องคอมพิวเตอร์ (Physical Implementtation)

3. การออกแบบฐานข้อมูลแบบเชิงสัมพันธ์ (Relational Database Design : RDD) การออกแบบฐานข้อมูลแบบเชิงสัมพันธ์นั้นอาศัยรายละเอียดจากแบบจำลองข้อมูลเชิงตรรกในหลักการที่ 2 เป็นพื้นฐานในการทำงานต่อไปโดยการแปลง (Map) แบบจำลองข้อมูลเชิงตรรกให้เข้ากับโครงสร้างของระบบจัดการฐานข้อมูลแบบเชิงสัมพันธ์

การเชื่อมโยงจากระดับเชิงตรรกไปสู่ระดับกายภาพจะต้องไม่เกิดความขัดแย้งกัน จึงต้องมีกลไกย้ายสารสนเทศดังกล่าว (Fleming,1989)

<u>แบบจำลองข้อมูลเชิงตรรก (Flemming and Von Halle,1989)</u>

แบบจำลองข้อมูลเชิงตรรกเป็นเทคนิคที่ใช้แทนโครงสร้าง สารสนเทศต่าง ๆ และข้อบังคับขององค์กร เพื่อเป็นทางเข้าในการออกแบบฐานข้อมูล

หัวใจของแบบจำลองข้อมูลเชิงตรรกนี้จะมองข้อมูลว่าเป็นแหล่งทรัพยากรณ์ที่มีคุณค่าในการทำธุรกิจองค์กร ปรัชญาของแบบจำลองข้อมูลเชิงตรรกก็คือ ข้อมูลขององค์กรต้องมีอยู่จริงเป็นอิสระ และไม่สนใจว่าข้อมูลเหล่านั้นจะถูกเข้าถึงได้อย่างไร โดยใคร จะใช้คอมพิวเตอร์หรือไม่ก็ได้ เช่น ข้าราชการครูในสังกัดกรุงเทพมหานคร คุณวุฒิทางการศึกษาของข้าราชการ สถานที่อยู่ของข้าราชการ เป็นต้น

ความจริงที่เกิดขึ้นในธุรกิจเหล่านี้ขององค์กร เราสามารถแทนลงในแบบจำลองข้อมูลเชิงตรรกได้ ถึงแม้ว่าจะไม่ได้ใส่ความจริงที่เกิดขึ้นในธุรกิจขององค์กรลงในฐานข้อมูลได้ครบก็ตาม ก็จะต้องใส่ความจริงเหล่านี้ลงในแบบจำลองข้อมูลเชิงตรรกไว้ด้วยเพื่อบุคคลทั่วไปจะได้เข้าใจจุดประสงค์ในการทำธุรกิจนี้ได้อย่างถูกต้อง

ลักษณะที่สำคัญในการสร้างแบบจำลองข้อมูลเชิงตรรก

1. เป็นการสร้างเอกสารที่สมบูรณ์แบบที่ช่วยให้เข้าใจถึงความต้องการของสารสนเทศต่าง ๆ และความสัมพันธ์เชิงธุรกิจขององค์กรได้ดี เพราะบรรยายในลักษณะแผนภาพประกอบข้อความบรรยาย

2. เป็นสื่อช่วยส่งเสริมความเข้าใจ ระหว่างผู้ออกแบบพัฒนาโปรแกรม และผู้ใช้ตลอดระยะเวลาในขั้นตอนการออกแบบระบบ

3. เป็นการจำลองแผนภาพขององค์กรว่ามีการจัดการสารสนเทศอย่างไรโดยไม่คำนึงถึงวิธีปฏิบัติการในระดับกายภาพ และถือว่าข้อมูลขององค์กรคือแหล่งทรัพยากรสำหรับการจัดองค์กรธุรกิจ ซึ่งเป็นพื้นฐานของการออกแบบ ทำให้การออกแบบฐานข้อมูลมี เสถียรภาพ (Integrity) ไม่ขัดแย้งกัน (Conssistent) และมีความยืดหยุ่น สามารถใช้ร่วมกันได้ (Sharable) ไม่ว่าจะใช้เทคโนโลยีการจัดการฐานข้อมูลใด ๆ ก็ตาม

ประโยชน์ของแบบจำลองข้อมูลเชิงตรรก

1. เข้าใจง่ายเพราะแสดงด้วยรูปภาพที่ไม่ซับซ้อน

2. แสดงถึงความจริงของธุรกิจในองค์กรได้เด่นชัด

โครงสร้างที่สำคัญของแบบจำลองข้อมูลเชิงตรรก

ประกอบไปด้วย

1. เอนทิตี (entity) หมายถึง คน สัตว์ สิ่งของ สถานที่ นามธรรมใดๆ หรือแนวคิดที่ต้องการบันทึก เช่น อาจารย์ วิชาที่สอน นักศึกษา ผลการศึกษาของ นักศึกษาเป็นต้น ซึ่งเป็นข้อเท็จจริงต่าง ๆ ที่อยู่ในระบบธุรกิจที่ทำการวิเคราะห์นั่นเอง เขียนแผนภาพแทนเอนทิตีด้วยรูปสี่เหลี่ยม ระบุชื่อเป็นภาษาอังกฤษอยู่เหนือบริเวณรูปสี่เหลี่ยม

2. ความสัมพันธ์ หมายถึง ความสัมพันธ์ระหว่าง 2 เอนทิตี เป็นความสัมพันธ์แบบทวิภาค (binary relationship) เช่น เอนทิตีอาจารย์ผู้สอนประจำวิชาต่าง ๆ กับเอนทิตีนักศึกษาที่ลงทะเบียนเรียนวิชานั้น เป็นต้น เขียนแผนภาพแทนความสัมพันธ์ด้วยเส้นตรงที่ปลายเส้นตรงมีลูกศรแสดงทิศทางโยงความสัมพันธ์จากเอนทิตีแม่ (parent entity)ไปยังเอนทิตีลูก (child entity) จำนวนหัวลูกศรและทิศทางของลูกศรจะแทนความหมายและประเภทของความสัมพันธ์ ระบุชื่อของความสัมพันธ์ด้วยกริยาวลีที่อธิบายความหมายที่ปรากฏในแผนภาพให้ชัดเจนยิ่งขึ้นบริเวณเหนือและใต้เส้นลูกศรนั้น จำนวนเอนติตีที่ปรากฏในความสัมพันธ์ใด ๆ จะใช้ค่าสัดส่วนเป็นตัวระบุ เรียกว่า สัดส่วนคาร์ดินัลลิตี (cardinality ratio)ใช้สัญลักษณ์ (n:m) แทนสัดส่วนดังกล่าว เมื่อค่า n แทนจำนวนที่ปรากฏของเอนติตีแม่ และค่า m แทนจำนวนที่ปรากฏของเอนติตีลูก เช่น ความสัมพันธ์ระหว่างอาจารย์ผู้สอนกับรายวิชาที่สอนมีสัดส่วนคาร์ดินัล เป็น (1:2) หมายถึงอาจารย์ 1 ท่าน สอนวิชา 2 วิชา เป็นต้น

3. แอตตริบิวต์หมายถึง คำอธิบาย ข้อเท็จจริง หรือ สารสนเทศที่แยกออกไม่ได้แล้วใช้ในการอธิบายเอนทิตี แอตตริบิวต์เป็นลักษณะประจำหน่วยที่เล็กที่สุดของสารสนเทศที่ต้องการอ้างถึงในขณะใดขณะหนึ่ง

นอกจากนี้แล้วต้องบันทึกความหมาย หรือคำจำกัดความของแอตตริบิวต์ลงในพจนานุกรมข้อมูลด้วยเพราะการเป็นแบบจำลองข้อมูลเชิงตรรกที่สมบูรณ์นั้น ไม่ได้ประกอบไปด้วยแผนภาพเท่านั้น แต่จะต้องเก็บข้อกำหนดต่างๆ ลงในพจนานุกรมข้อมูลด้วย

พจนานุกรมข้อมูล ก็คือที่เก็บสารสนเทศเกี่ยวกับโปรแกรมประยุกต์ ฐานข้อมูลแบบจำลองข้อมูลเชิงตรรก ผู้ใช้ และอำนาจในการเข้าถึงข้อมูลของผู้ใช้ (Access Authorizations)

การออกแบบจำลองข้อมูลเชิงตรรก (Flemming and Von Halle, 1989)

ในขั้นตองแรกจำเป็นต้องพิจารณา โครงสร้างจากมุมมองของผู้ใช้ (Skeletal User View) โดยที่มุมมองของผู้ใช้หมายถึง รูปแบบต่าง ๆ ที่เป็นตัวแทนความต้องการจากผู้ใช้คนใดคนหนึ่ง หรือหน้าที่ส่วนใดส่วนหนึ่งของระบบ โครงสร้างจากมุมมองของผู้ใช้ประกอบไปด้วยความต้องการต่าง ๆ ของระบบนั่นคือจะมีเทอมต่างๆ ได้แก่ เอนทิตี ความสัมพันธ์ คีย์หลัก (Primary Key) คีย์รอง (Alternate Key) และคีย์ของกฎธุรกิจ (Key Business Rule) โครงสร้างจากมุมมองของผู้ใช้เป็นประโยชน์อย่างมากในการเป็นอินพุตของการออกแบบฐานข้อมูล รวมทั้งการออกแบบต้นแบบของฐานข้อมูล

ต้นแบบ หมายถึง การทดลองสร้างบางส่วนของการติดตั้งฐานข้อมูลการออกแบบต้นแบบนี้มีจุดประสงค์เพื่อทดสอบว่าระบบมีความต้องการเพียงพอหรือยัง

ขั้นตอนในการออกแบบจำลองข้อมูลเชิงตรรก

(Building a Logical Data Model)

1. กำหนดเอนทิตีหลัก (Identify Major Entities :LDM1)

เอนทิตี หมายถึง คน สถานที่ สิ่งของ หรือ ความคิดที่ต้องการบันทึก สารสนเทศต่างๆ เอนทิตีแบ่งออกได้เป็น เอนทิตีที่เป็นซับไทป์ และ เอนทิตีที่เป็นซูเปอร์ไทป์ ซึ่งเอนทิตีที่เป็นซับไทป์มีข้อกำหนดและคุณสมบัติต่าง ๆ เพิ่มขึ้นมากกว่า เอนทิตีที่เป็นซูเปอร์ไทป์ ซึ่งทั้งเอนทิตีเป็นซับไทป์ และเอนทิตีที่เป็นซูเปอร์ไทป์ จะแทนสิ่งเดียวกันก็ตาม เช่น ข้าราชการในสำนักการศึกษาสังกัดกรุงเทพมหานคร เป็นเอนทิตีที่เป็นซูเปอร์ไทป์ โดย มีข้าราชการครูเสมียน เป็นเอนทิตีที่เป็นซับไทป์ เป็นต้น ในหนึ่งซูเปอร์ไทป์เราสามารถกำหนดซับไทป์ได้หลาย ๆ ซับไทป์

โดยมีหลักการของการเป็น ซูเปอร์ไทป์ และ ซับไทป์ ดังนี้ เอนทิตี ก จะเป็นซับไทป์ของเอนทิตี ข และเอนทิตี ข จะเป็น ซูเปอร์ไทป์ของเอนทิตี ก ก็ต่อเมื่อ

1.1 ก และ ข แทนสิ่งเดียวกัน

1.2 ก มีคุณสมบัติเช่นเดียวกับ ข

1.3 ทุก ๆ ค่าของ ก จะมีอยู่ในค่า ๆ หนึ่งของ ข แต่ในทางกลับกันคือทุก ๆ ค่าของ ข ไม่จำเป็นต้องอยู่ในค่า ๆ หนึ่งของ ก

เมื่อกำหนดเอนทิตีหลักได้แล้วต้องกำหนดชื่อ แผนภาพ และความหมายของเอนทิตี แล้วบันทึกลงในพจนานุกรมข้อมูล

แผนภาพแบบจำลองข้อมูลเชิงตรรก ประกอบไปด้วยสารสนเทศ และความสัมพันธ์ต่าง ๆ เอนทิตีเป็นสิ่งแรกที่ถูกวาดลงในแผนภาพ เอนทิตีจะถูกแทนด้วยรูปสี่เหลี่ยมโดยจะเขียนชื่อของเอนทิตีเป็นภาษาอังกฤษตัวใหญ่ที่มุมบนด้านซ้ายของสี่เหลี่ยมนั้น ดังรูปที่ 3.1 ข้างล่างนี้

TEACHER

รูปที่ 3.1 แสดงการแทนเอนทิตีของครูด้วยภาพ

2. กำหนดความสัมพันธ์ระหว่างเอนทิตี (Determine Relationships Between Entities :LDM2)

ความสัมพันธ์ของข้อมูล หมายถึง ความจริงของความสัมพันธ์ระหว่างเอนทิตี 2 เอนทิตี ซึ่งมีความสัมพันธ์อยู่ 3 แบบ คือ

I 17498363

2.1 ความสัมพันธ์ที่มีอยู่จริง (Existence Relationships) เช่น ข้าราชการที่มีคู่สมรส

2.2 ความสัมพันธ์ตามหน้าที่ (Functional Relationships) เช่น ครูสอนนักเรียน

2.3 ความสัมพันธ์ตามเหตุการณ์ (Event Relationships) เช่น การเดินทางมาทำงานของข้าราชการ

เมื่อกำหนดความสัมพันธ์แล้วให้กำหนดชื่อ และแผนภาพ และบันทึกลงในพจนานุกรมข้อมูล แผนภาพของความสัมพันธ์เป็นเส้นตรงมีหัวเป็นลูกศรโดยจะอยู่ ระหว่างเอนทิตี 2 เอนทิตีที่เกี่ยวข้องกันโดยมีทิศทางจาก เอนทิตีหนึ่ง ซึ่งเรียกว่า เอนทิตีแม่ (Parent Entity) ไปยังอีกเอนทิตีหนึ่งซึ่งเรียกว่าเอนทิตีลูก (Child Entity) กำหนดชื่อของความสัมพันธ์เป็นภาษาอังกฤษตัวเล็กไว้เหนือเส้นตรงนั้น

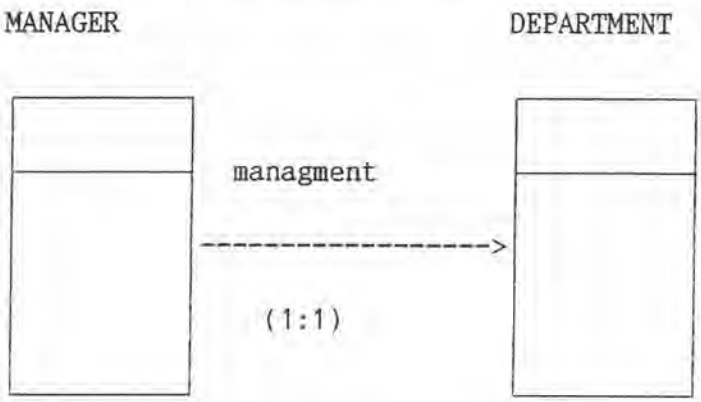
จำนวนของการสัมพันธ์ข้อมูลในแต่ละข้างระหว่างเอนทิตีแม่กับเอนทิตีลูก เมื่อนำมาเปรียบเทียบกันตามสัดส่วนความสัมพันธ์ เราเรียกว่าสัดส่วนนี้ว่าสัดส่วน คาร์ดินัลลิตี (Cardinulity Ratio) เขียนระบุไว้โดยใช้สัญลักษณ (n:m) เขียนไว้ใต้เส้นลูกศร เมื่อ n คือจำนวนที่ปรากฏของเอนทิตีแม่ และ m คือจำนวนที่ปรากฏของเอนทิตีลูก

ประเภทของความสัมพันธ์ มีอยู่ 3 แบบ คือ

1. แบบหนึ่งต่อหนึ่ง (One to one Relationship หรือ 1:1) แต่ละค่าของเอนทิตีแม่มีความสัมพันธ์กับเอนทิตีลูกได้อย่างมากที่สุดเพียงหนึ่งค่า เขียนสัญลักษณ์แสดงแทนด้วยหัวลูกศร 1 หัว และระบุค่าสัดส่วนคาร์ดินัลลิตีเป็น 1:1

ตัวอย่างเช่น โรงเรียนแห่งหนึ่งมีหลายหมวดวิชา (DEPARTMENT) ตามรูปในหน้าถัดไปนี้ มีอาจารย์หัวหน้าหมวดวิชา (MANAGER) เป็นเอนติตีแม่ และแผนก (DEPARTMENT) เป็น เอนทิตีลูก มีความสัมพันธ์ชื่อ managment หมายถึงต่ละแผนกจะมีอาจารย์หัวหน้าหมวดวิชา

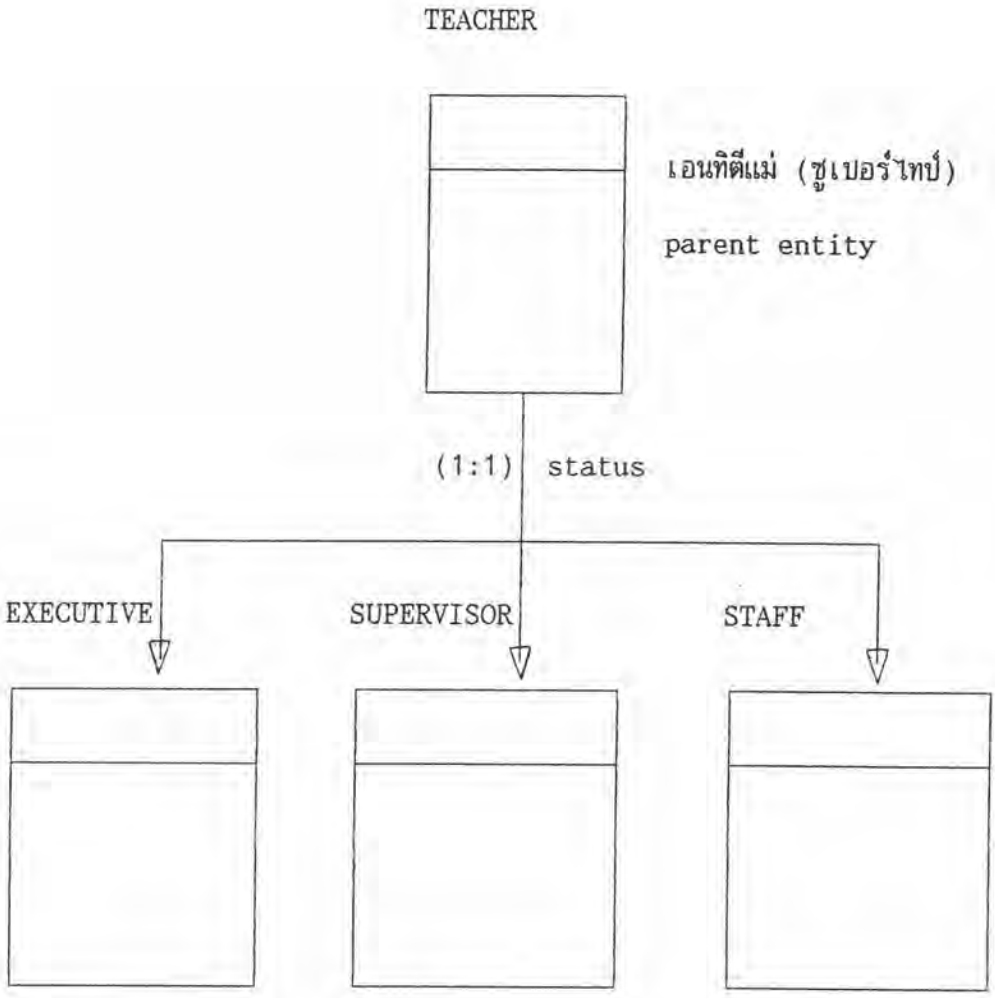
(MANAGER) บริหารงานได้เพียง 1 คน และอาจารย์ หัวหน้าหมวดวิชาแต่ละคนจะบริหารงานได้เพียง 1 หมวดวิชาเท่านั้น ดังรูปที่ 3.2 ข้างล่างนี้



รูปที่ 3.2 แสดงความสัมพันธ์ระหว่างเอนทิตีแบบหนึ่งต่อหนึ่ง

ตัวอย่างพิเศษของความสัมพันธ์ประเภทหนึ่งต่อหนึ่งของเอนทิตีที่เป็นซูเปอร์ไทป์กับเอนทิตีที่เป็นซับไทป์

ตัวอย่างเช่น เอนทิตีข้าราชการครู (TEACHER) ของสำนักการศึกษาในสังกัดกรุงเทพมหานคร เป็นเอนทิตีแม่ ประเภทซูเปอร์ไทป์ ประกอบด้วยแอตตริบิวต์ (attribute) ที่ระบุคุณสมบัติโดยส่วนรวมของเอนทิตีนี้ มีเอนทิตีลูกหรือซับไทป์ซึ่งประกอบด้วยแอตตริบิวต์ที่มีคุณสมบัติเพิ่มเติมไปจากเอนทิตีแม่หรือซูเปอร์ไทป์โดยเฉพาะประเภทนั้น ๆ โดยที่ข้าราชการครูแต่ละคนมี สถานะภาพ (Status)ในระดับตำแหน่งต่างกัน เช่น ระดับบริหาร (EXECUTIVE) ระดับหัวหน้าหมวดวิชา (SUPERVISOR) และ ระดับผู้สอน (STAFF) ดังรูปที่ 3.3 หน้าถัดไปนี้

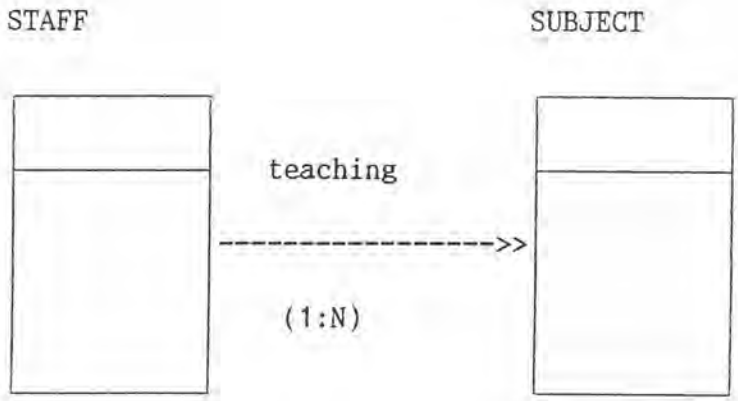


รูปที่ 3.3 แสดงความสัมพันธ์ระหว่างซูเปอร์ไทป์กับซับไทป์แบบหนึ่งต่อหนึ่ง

2. แบบหนึ่งต่อหลาย(One to many Relationship หรือ 1:N)

เอนทิตีแม่แต่ละค่าจะมีความสัมพันธ์กับเอนทิตีลูกได้หลายค่าแต่ในทางกลับกันแต่ละค่าของเอนทิตีลูกมีความสัมพันธ์กับเอนทิตีแม่ได้เพียงค่าเดียวเท่านั้น สัญญลักษณ์แสดงแทนด้วยหัวลูกศรคู่ติดกันหันทิศทางไปยังเอนทิตีลูก ระบุสัดส่วนคาร์ดินัลลิตีด้วย (1:N)

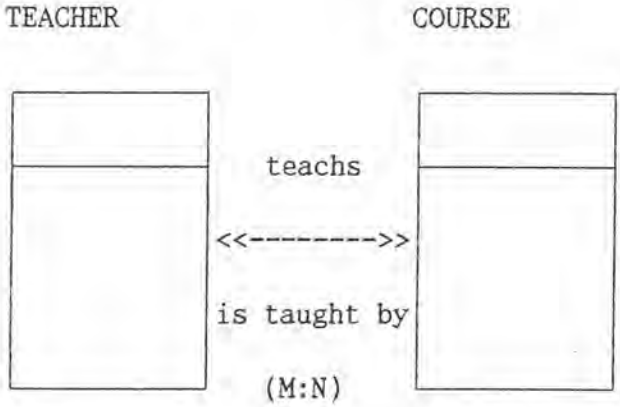
ตัวอย่างเช่น อาจารย์ผู้สอน (STAFF) เป็นเอนทิตีแม่ และรายวิชาที่สอน (SUBJECT) เป็นเอนทิตีลูก มีความสัมพันธ์ teaching หมายถึงอาจารย์ 1 ท่านสอนหลายวิชา ดังรูป 3.4 หน้าถัดไปนี้



รูปที่ 3.4 แสดงความสัมพันธ์ระหว่างเอนทิตีทั้งสองแบบหนึ่งต่อหลาย

3. แบบหลายต่อหลาย (Many to many หรือ M:N Relationship) แต่ละค่าของเอนทิตีแม่มีความสัมพันธ์กับเอนทิตีลูกได้หลายค่า และในทางกลับกันแต่ละค่าของเอนทิตีลูกมีความสัมพันธ์กับเอนทิตีแม่ได้หลายค่า ใช้สัญลักษณ์หัวลูกศรคู่หันไปเอนทิตีแม่ และเอนทิตีลูกทั้งสองทิศทาง ระบุสัดส่วนคาร์ดินัลลิตีเป็น (M:N)

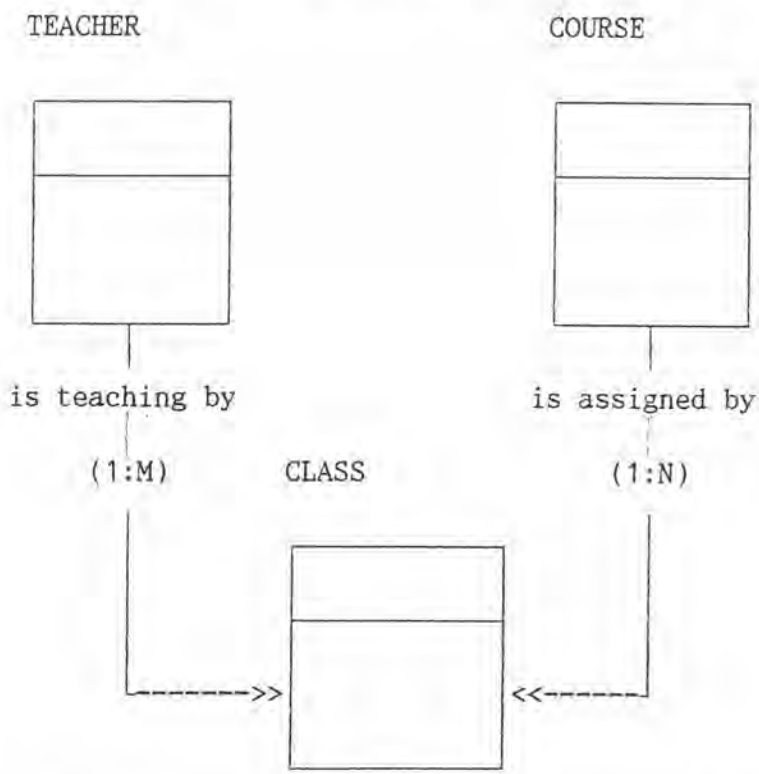
ตัวอย่างเช่น อาจารย์ (TEACHER) เป็นเอนทิตีแม่ และวิชา (COURSE)เป็นเอนทิตีลูก มีรีเลชันชิปชื่อ teachs หมายความถึง อาจารย์แต่ละคนมีการสอนได้หลายรายวิชา แต่ถ้าพิจารณาหัวลูกศรคู่ที่หันไปทางซ้ายมือ มีความสัมพันธ์ชื่อ is taugh by แล้วรายวิชา (COURSE) เป็นเอนทิตีแม่ และ อาจารย์ (TEACHER)เป็นเอนทิตีลูก หมายถึงแต่ละรายวิชามีอาจารย์สอนหลายคน ดังรูป 3.5 ข้างล่างนี้



รูปที่ 3.5 แสดงความสัมพันธ์ระหว่างเอนทิตีทั้งสองแบบหลายต่อหลาย

ความสัมพันธ์ข้อมูลประเภทหลายต่อหลายมีความซับซ้อนมาก คือการแทนความสัมพันธ์แบบนี้ค่อนข้างจะยุ่งยาก จึงแก้ไขทำให้ง่ายขึ้นได้โดยแปลงการสัมพันธ์ข้อมูลแบบหลายต่อหลายหนึ่งความสัมพันธ์ มาเป็นความสัมพันธ์แบบหนึ่งต่อหลาย 2 ความสัมพันธ์ โดยที่เอนทิตีทั้งสองนี้มีความสัมพันธ์กับเอนทิตีที่ระบุขึ้นมาใหม่เพื่อเชื่อมโยงเอนทิตีเดิม

ตัวอย่างเช่น ระบุชั้นเรียน (CLASS) เป็นเอนทิตีใหม่เพื่อใช้ในการเชื่อมโยงเอนทิตีทั้งสองของเดิมคือ อาจารย์ (TEACHER) และ รายวิชา (COURSE)มีความสัมพันธ์เกิดขึ้นใหม่ 2 ความสัมพันธ์ คือ is teaching by มีอาจารย์เป็นเอนทิตีแม่ และชั้นเรียน (CLASS)เป็นเอนทิตีลูก หมายถึงภายในชั้นเรียนอาจารย์ 1 คนสอนได้หลายรายวิชา ใช้ (1:M) ระบุสัดส่วนคาร์ดินัลลิตี และอีกความสัมพันธ์หนึ่งชื่อ is assigned by มีรายวิชาเป็นเอนทิตีแม่ และชั้นเรียนเป็นเอนทิตีลูก หมายถึงภายในชั้นเรียนกำหนดให้แต่ละรายวิชามีอาจารย์สอนได้หลายคน ใช้ (1:N) ระบุสัดส่วนคาร์ดินัลลิตี ดังรูปที่ 3.6 ข้างล่างนี้



รูปที่ 3.6 แสดงความสัมพันธ์ระหว่างเอนทิตีทั้งสองแบบหนึ่งต่อหลาย

3. กำหนดแอตตริบิวต์ประเภทคีย์หลัก และประเภทคีย์รอง (Determine Primary and Alternate keys :LDM3)

แอตตริบิวต์ หมายถึง ลักษณะประจำเป็นหน่วยที่เล็กที่สุดของข้อมูลที่มีความเกี่ยวข้องกับเอนทิตี ซึ่งเป็นความจริงที่ไม่สามารถแบ่งเป็นส่วนเล็กๆ ออกไปกว่านี้ได้แล้ว

ขั้นตอนนี้เป็นการกำหนดรายละเอียดต่างๆ ของเอนทิตีโดยการกำหนดแอตตริบิวต์เพื่ออธิบายรายละเอียด เช่น เลขประจำตัวของข้าราชการ ชื่อ เพศ เป็นต้น

ประเภทของแอตตริบิวต์ มี 4 ประเภทดังนี้

3.1 แอตตริบิวต์ประเภทคีย์หลัก (primary key attribute)

แอตตริบิวต์แรกที่ต้องกำหนดลงในแบบจำลองข้อมูลเชิงตรรกก็คือแอตตริบิวต์ประเภทคีย์หลัก คือ แอตตริบิวต์ หรือกลุ่มของแอตตริบิวต์ที่ระบุถึงแต่ละรายการในเอนติตีใด ๆ ซึ่งค่าที่ระบุนั้นมีผลทำให้รายการนั้นเป็นเอกลักษณ์ (Unique)ไม่ซ้ำกับรายการอื่น ใช้ข้อความภาษาอังกฤษเขียนเป็นชื่อของคีย์หลักอยู่เหนือเส้นตรงแนวนอนในช่องบนของรูปสี่เหลี่ยมที่แทนเอนทิตี ส่วนแอตตริบิวต์ อื่น ๆ นอกจากนี้จะเขียนไว้ใต้เส้นแนวนอนภายในสี่เหลี่ยมลงมา เช่น เอนทิตีที่ชื่อ TEACHER มี แอตตริบิวต์ชื่อ TEACHER_ID เป็นแอตตริบิวต์หลัก ดังรูปที่ 3.7 ข้างล่างนี้

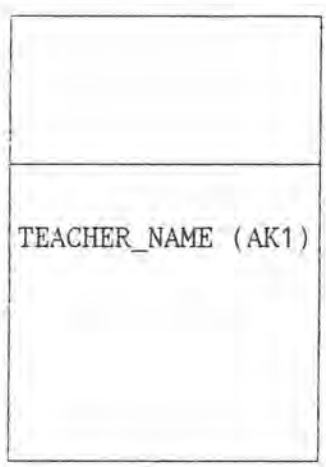
TEACHER

รูปที่ 3.7 แสดงการเขียนแอตตริบิวต์คีย์หลัก TEACHER_ID ในเอนทิตี TEACHER

สำหรับคีย์หลักของเอนทิตีที่เป็นซับไทป์นั้น จะต้องเหมือนกับคีย์หลักของเอนทิตีที่เป็นซูเปอร์ไทป์

3.2 แอตตริบิวต์ประเภทคีย์รอง (alternate key attribute)

แอตตริบิวต์หรือกลุ่มของแอตตริบิวต์อื่นที่ใช้ระบุถึงแต่ละรายการในเอนทิตีเช่นเดียวกับคีย์หลัก เราเรียกแอตตริบิวต์หลาย ๆ แอตตริบิวนี้ว่า แคนดิเดทคีย์ (Candidate Key) หรือคีย์คู่แข่ง แอตตริบิวต์ใด ๆ ที่ไม่ถูกเลือกมาเป็นคีย์หลัก จะเรียกว่าเป็นคีย์รอง (Alternate Key) จะใช้สัญญลักษณ์ AKn เขียนไว้หลังชื่อแอตตริบิวนั้นและเขียนไว้ใต้เส้นตรงแนวนอนภายในเอนทิตี โดยค่าของ n เป็นเลขจำนวนเต็มมีค่าตั้งแต่ 1,2,3,....n ซึ่งหมายถึงคีย์รองตัวที่ n ดังรูปที่ 3.8 ข้างล่างนี้



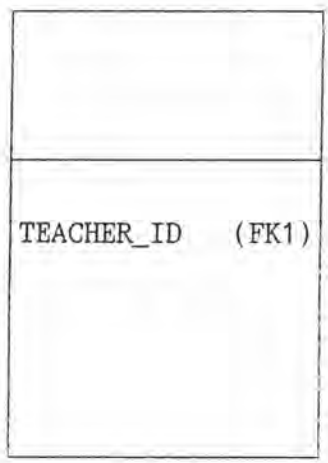
รูปที่ 3.8 แสดงการเขียนแอตตริบิวต์คีย์รอง TEACHER_NAME ในเอนทิตี TEACHER

เมื่อกำหนดคีย์หลัก และคีย์รองของเอนทิตีได้แล้ว ให้บันทึก ชื่อ และความหมายลงในพจนานุกรมข้อมูล ในการตั้งชื่อแอตตริบิวต์ควรให้มีมาตรฐานในการตั้งชื่อ ซึ่งรวมไปทั้งการตั้งชื่อเอนทิตี และความสัมพันธ์ด้วย

3.3 แอตตริบิวต์ประเภทฟอร์เรนจ์คีย์ (foreign Key attribute)

ฟอร์เรนจ์คีย์ (Foreign Key) เป็นแอตตริบิวต์ หรือเซตของแอตตริบิวต์ที่อยู่ในเอนทิตีลูกซึ่งมีค่าสอดคล้องกับค่าของคีย์หลักที่อยู่ในเอนทิตีแม่ ที่มีความสัมพันธ์กันอยู่ และอาจไม่มีค่าก็ได้ (null) ใช้สัญลักษณ์ FKn ระบุไว้หลังข้อความภายในรูปสี่เหลี่ยม โดยค่าของ n เป็นเลขจำนวนเต็ม มีค่าตั้งแต่ 1,2,3,.....,n n หมายถึงฟอร์เรนจ์คีย์ตัวที่ n ดังรูปที่ 3.9 ข้างล่างนี้

SUBJECT



รูปที่ 3.9 แสดงการเขียนแอตตริบิวต์ประเภทฟอร์เรนจ์คีย์ชื่อ SUBJECT_ID ในเอนทิตี TEACHER

จากรูปที่ 3.9 นี้จะเห็นได้ชัดว่าเอนทิตีแม่ คือ รายชื่ออาจารย์ (TEACHER) เอนทิตีลูกคือ วิชาที่เปิดสอน (SUBJECT)โดยมีรหัสของอาจารย์ผู้สอน เป็นฟอร์เรนจ์คีย์ในเอนติตีลูก และ เป็นคีย์หลักในเอนติตีแม่

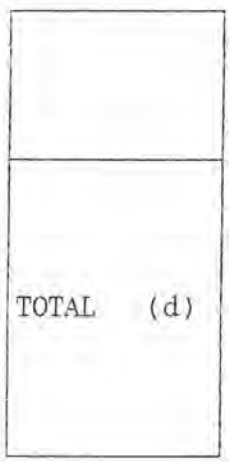
3.4 แอตตริบิวต์อื่น ๆ ที่ไม่ใช่คีย์ (non key attribute)

คือ แอตตริบิวที่ไม่มีคุณสมบัติตาม 3 ข้อดังกล่าวมาแล้วข้างต้น

แอตตริบิวต์พิเศษ คือ แอตตริบิวต์ที่ไม่เข้าพวกใน 4 ข้อข้างต้น ได้แก่

แอตตริบิวต์ประเภทดิไรฟว์ (derived attributes) แอตตริบิวต์นี้อ้างถึงในการคำนวณทำให้ไม่เกิดความขัดแย้งของข้อมูล ไม่มีการสร้างจริงในระดับกายภาพเพื่อประหยัดเนื้อที่ในการเก็บข้อมูล และลดภาระในการแก้ไขเมื่อมีการเปลี่ยนแปลงแอตตริบิวต์ต้นกำเนิด แอตตริบิวต์นี้อ้างถึงแอตตริบิวต์อื่นที่เป็นแหล่งกำเนิดค่าในพจนานุกรมข้อมูล ใช้สัญลักษณ์ d ระบุหลังข้อความที่เป็นชื่อในภาษาอังกฤษ ดังรูปที่ 3.10 ข้างล่างนี้

SALARY



รูปที่ 3.10 แสดงตัวอย่างของแอตตริบิวต์ประเภทดิไรฟว์ชื่อ TOTAL

ค่าของแอตตริบิวต์ประเภทดิไรฟว์ชื่อ TOTAL ได้มาจากผลรวมของเงินเดือนของข้าราชการครู เมื่อค่ารหัสข้าราชการครูในเอนทิตีข้าราชการครู (TEACHER) และ เอนทิตีเงินเดือน (SALARY) มีค่าตรงกันตรงกัน

4. กำหนดฟอร์เรนจ์คีย์ (Determine Foreign keys :LDM4)

การกำหนดฟอร์เรนจ์คีย์นี้สำคัญมาก โดยเฉพาะอย่างยิ่งเมื่อสร้างฐานข้อมูลแบบเชิงสัมพันธ์ เพราะว่าฟอร์เรนจ์คีย์จะเก็บคุณสมบัติในการอ้างถึงระหว่างตารางความสัมพันธ์ เมื่อกำหนดแล้วให้บันทึกชื่อ ความหมาย ความสัมนธ์ที่เกี่ยวข้องกันลงในพจนานุกรมข้อมูล

5. กำหนดคีย์ของกฎธุรกิจ (Determine Key Business Rules : LDM5)

การกำหนดข้อมูลให้มีความเป็นบูรณภาพ ไม่มีความขัดแย้งกัน ,ลดความเหลือเฟือให้น้อยที่สุด ข้อมูลมีความถูกต้อง และมีความคงที่ อาศัยหลักการกำหนดกฎเกณฑ์ต่าง ๆ คือ การกำหนดกฎธุรกิจ ดังนี้

วิธีการกำหนดกฎธุรกิจ (Business Rule) ของเอนติตีมีอยู่ 3 ประเภทคือ

1. กฎของคีย์ธุรกิจ (Key Business Rules) จะควบคุมความสัมพันธ์ระหว่างคีย์หลักและฟอร์เรนจ์คีย์ที่อ้างอิงถึงกัน โดยระบุเหตุการณ์ในเรื่องของคำสั่งการแทรก (Insert) การลบ (Delete) การปรับปรุง (Updatte) ที่กระทำต่อความสัมพันธ์ เช่น จะมีผลกระทบอย่างไรบ้างเมื่อมีการเปลี่ยนแปลงของคีย์หลัก หรือ ฟอร์เรนจ์คีย์

2. กฎของโดเมนธุรกิจ (Domains Business Rules) จะควบคุมชนิดและช่วงของค่าที่ยอมรับได้ของแอตตริบิวต์ให้ถูกต้อง

3. ทริกเกอร์ดำเนินการ (Triggering Operations) จะควบคุมผลกระทบเมื่อมีการเพิ่ม เปลี่ยนแปลง หรือ ยกเลิกแอตตริบิวต์ใด ๆ และการดึงข้อมูลที่มีผลต่อแอตตริบิวต์ต่าง ๆ ในเอนติตีเดียวกัน

ขั้นตอนของการกำหนดคีย์ของกฎธุรกิจ (Key Business Rules) มีดังต่อไปนี้

5.1 กำหนดกฎการแทรกให้กับแต่ละความสัมพันธ์

กฎการแทรก (Insert Rule) หมายถึง การกำหนดเงื่อนไข ควบคุมที่เป็นไปได้ในการแทรกหรือเพิ่มรายการของเอนทิตี หรือการปรับปรุงฟอร์เรนจ์คีย์ในเอนทิตีลูก ต้องคำนึงถึงค่าฟอร์เรนจ์คีย์ที่สอดคล้องกับค่าของคีย์หลักใน เอนทิตีแม่

กฎการแทรกแบ่งออกเป็น 6 ประเภทคือ

5.1.1 ขึ้นต่อกัน (Dependent) อนุญาตให้แทรกเอนทิตีลูกได้เมื่อคีย์หลักมีค่าตรงกับเอนทิตีแม่ที่มีอยู่

5.1.2 อัตโนมัติ (Automatic) อนุญาตให้แทรกเอนทิตีลูกได้เมื่อคีย์หลักมีค่าไม่ตรงกับคีย์หลักของเอนทิตีแม่ ก็ต่อเมื่อสร้างค่าคีย์หลักเดียวกันในเอนทิตีแม่นั้นขึ้นมาก่อนแล้วค่อยแทรกลงในเอนทิตีลูก

5.1.3 กำหนดให้มีค่าเป็นนัล (Nullify) อนุญาตให้แทรกเอนทิตีลูกได้เมื่อมีค่าคีย์หลักตรงกับเอนทิตีแม่ที่มีอยู่ แต่เมื่อค่าคีย์หลักในเอนทิตีแม่ไม่มีให้กำหนดฟอร์เรนจ์คีย์ในลูกให้เป็นนัล

5.1.4 กำหนดให้มีค่าโดยปริยาย (Default) อนุญาตให้แทรกเอนทิตีลูกได้เมื่อมีค่าคีย์หลักตรงกับเอนทิตีแม่ที่มีอยู่ แต่เมื่อค่าคีย์หลักในเอนทิตีแม่ไม่มีให้กำหนดฟอร์เรนจ์คีย์ในลูกให้มีค่าโดยปริยาย

5.1.5 กำหนดให้มีค่าตามต้องการ (Customized) อนุญาตให้แทรกเอนทิตีลูกได้เมื่อมีค่าคีย์หลักตรงกับเอนทิตีแม่ที่มีอยู่ แต่เมื่อค่าในเอนทิตีแม่ไม่มีให้กำหนดฟอร์เรนจ์คีย์ในลูกให้มีค่าตามเงื่อนไขที่กำหนดไว้ตามต้องการ

5.1.6 ไม่มีผลกระทบ (No Effect) กำหนดให้มีการแทรกเอนทิตีลูกได้โดยไม่ต้องตรวจสอบจากเอนทิตีแม่ว่ามีค่าคีย์หลักตรงกับเอนทิตีแม่หรือไม่

5.2 กำหนดกฎการลบให้กับความสัมพันธ์แต่ละความสัมพันธ์

กฎการลบ (Delete Rule) หมายถึง การกำหนดเงื่อนไขที่เป็นไปได้ว่าเมื่อมีการลบเอนทิตีแม่ หรือ ปรับปรุงคีย์หลักของเอนทิตีแม่ แล้วจะเกิดผลกระทบอะไรขึ้นกับเอนทิตีลูก

กฎการลบแบ่งออกเป็น 6 ประเภทคือ

5.2.1 มีข้อจำกัด (Restrict) ลบเอนติตีแม่ได้เมื่อไม่มีคีย์หลักของเอนทิตีลูกตรงกับคีย์หลักเอนทิตีแม่

5.2.2 ต่อเนื่อง (Cascade) เมื่อจะลบเอนทิตีแม่ให้ดูว่ามีคีย์หลักของเอนทิตีลูกที่ตรงกันกับคีย์หลักของเอนทิตีแม่หรือไม่ ถ้ามีให้ลบเอนทิตีลูกนั้นด้วย

5.2.3 กำหนดให้มีค่าเป็นนัล (Nullify) กำหนดฟอร์เรนจ์คีย์ในเอนทิตีลูกให้มีค่าเป็นนัล เมื่อมีคียหลักของเอนทิตีลูกตรงกับเอนทิตีแม่ที่ต้องการลบ

5.2.4 กำหนดให้มีค่าโดยปริยาย (Default) กำหนดฟอร์เรนจ์คีย์ในเอนทิตีลูกให้มีค่าโดยปริยาย เมื่อต้องการลบเอนทิตีแม่ที่มีคีย์หลักตรงกับเอนทิตีลูก

5.2.5 กำหนดให้มีค่าตามต้องการ (Customized) อนุญาตให้ลบเอนทิตีลูกได้เมื่อมีค่าคีย์หลักตรงกับเอนทิตีแม่ที่มีอยู่ แต่เมื่อคีย์หลักในเอนทิตีแม่ไม่มี ให้กำหนดฟอร์เรนจ์คีย์ในเอนทิตีลูกให้มีค่าตามเงื่อนไขที่กำหนดไว้ตามต้องการ

5.2.6 ไม่มีผลกระทบ (No Effect) กำหนดให้มีการลบเอนทิตีลูกได้โดยไม่ต้องตรวจสอบว่ามีค่าคีย์หลักตรงกับเอนทิตีแม่หรือไม่

การกำหนดคีย์ของธุรกิจควรหลีกเลี่ยงการใช้กฎการแทรก หรือกฎการลบประเภทที่กำหนดให้มีค่าเป็นนัล ควรใช้กฎที่เป็นค่าโดยปริยายแทน

ห้ามกำหนดกฎประเภทให้มีค่าเป็นนัล เมื่อฟอร์เรนจ์คีย์เป็นส่วนหนึ่งของคีย์หลักในเอนติตีลูก

ควรกำหนดกฎการแทรกระหว่างเอนทิตีที่เป็นซับไทป์ กับเอนทิตีที่เป็น ซูเบอร์ไทป์ให้เป็นประเภทอัตโนมัติ และสำหรับกฎการลบให้เป็นประเภทแบบต่อเนื่อง

6. เพิ่มแอตตริบิวต์อื่นที่ไม่ใช่คีย์ (Add Remainning Attributes : LDM6)

เพื่อเป็นรายละเอียดของเอนทิตีได้อย่างชัดเจน การกำหนดแอตตริบิวต์ที่ไม่ได้เป็นคีย์มีวิธีการดังต่อไปนี้

6.1 กำหนดแอตตริบิวต์ให้กับเอนทิตี โดยที่แอตตริบิวต์เหล่านั้นต้องขึ้นอยู่กับคีย์หลักของของเอนทิตีนั้น ๆ ด้วย เช่น ชื่อของสินค้า ขึ้นอยู่กับรหัสสินค้าซึ่งเป็นคีย์หลักในเอนทิตีรายการสินค้า (PRICE_LIST) เป็นต้น

6.2 กำหนดแอตตริบิวต์ให้กับเอนทิตีให้มากที่สุดเท่าที่จะมากได้

6.3 ถ้าแอตตริบิวต์ใดที่ขึ้นอยู่กับคีย์หลักมีค่าหลายค่า ให้กำหนดแอตตริบิวต์นั้นใหม่ให้เป็นเอนทิตีลูก ถ้าแอตตริบิวต์ใดในเอนทิตีลูกใหม่นั้นมีเพียงค่าเดียวให้นำแอตตริบิวต์นั้นเป็นคีย์หลักแต่ถ้าแอตตริบิวต์ในเอนทิตีลูกมีหลายค่าทุกตัวแอตตริบิวต์ ให้นำคีย์หลักของเอนทิตีเดิมมาเป็นคีย์หลักของเอนทิตีลูก

6.4 กำหนดชื่อแอตตริบิวต์ ความยาว ความหมาย ชนิด เอนทิตีที่เกี่ยวข้องและระบุว่าเป็นคีย์หลัก คีย์รอง หรือ ฟอร์เรนจ์คีย์ โดยบันทึกลงในพจนานุกรมข้อมูล

6.5 ถ้ามีแอตตริบิวต์ที่ดูเหมือนกับว่าจะอธิบายความสัมพันธ์ ให้สร้างความสัมพันธ์นั้นเป็นเอนทิตีใหม่ที่เป็นเอนทิตีลูกของเอนทิตีเดิมทั้งสอง

6.6 หลีกเลี่ยงการแทนแอตตริบิวต์ด้วยการเข้ารหัส (Encode Attribute) ถ้ารหัสนั้นไม่มีความหมายต่อระบบ

การหลีกเลี่ยงการเข้ารหัสของแอตตริบิวต์นั้นมีเหตุผลดังนี้คือ

6.6.1 รหัสไม่สื่อความหมายกับผู้ใช้ และเป็นการขัดวัตถุประสงค์ของการสร้าง แบบจำลองข้อมูลเชิงตรรก ที่ว่าต้องทำให้มุมมองของผู้ใช้ต้องกระจ่างชัดเจน

6.6.2 การใช้รหัสทำให้ผู้ใช้ปลายทางสอบถามข้อมูลได้ลำบาก

6.6.3 ทำให้ต้องมีเอนทิตีขึ้นมาใหม่เพื่ออธิบายรหัสเหล่านั้น และต้องกลายมาเป็นเอนทิตีแม่ของทุกเอนทิตีที่มีรหัสปรากฏอยู่

6.7 ไม่ควรกำหนดแอตตริบิวต์ที่เป็นตัวบ่งชี้ไว้ในแบบจำลองข้อมูลเชิงตรรก

6.8 ถ้าจำเป็นต้องแทนแอตตริบิวต์ให้อยู่ในรูปที่ต้องเข้ารหัสด้วยเหตุผลทางธุรกิจแล้ว ต้องทำให้รหัสเหล่านั้นมีค่าที่เป็นอิสระจากกัน (Mutually Independent)

พิจารณาตามตารางที่ 3.1 จะเห็นว่ามีรหัสที่มีค่าที่ไม่เป็นอิสระจากกันดังนั้นจึงต้องจัดรหัสเสียใหม่ให้มีตารางรหัส 2 ตารางดังตารางที่ 3.2

รหัส	ความหมาย
WM	ลูกจ้างที่เป็นผู้จัดการรับเงินรายสัปดาห์
WE	ลูกจ้างที่เป็นวิศวกรรับเงินรายสัปดาห์
MM	ลูกจ้างที่เป็นผู้จัดการรับเงินรายเดือน
ME	ลูกจ้างที่เป็นวิศวกรรับเงินรายเดือน

ตารางที่ 3.1 แสดงแอตตริบิวต์ในรูปของการเข้ารหัส และความหมายของรหัสโดยที่รหัสนั้นมีค่าที่ไม่เป็นอิสระจากกัน

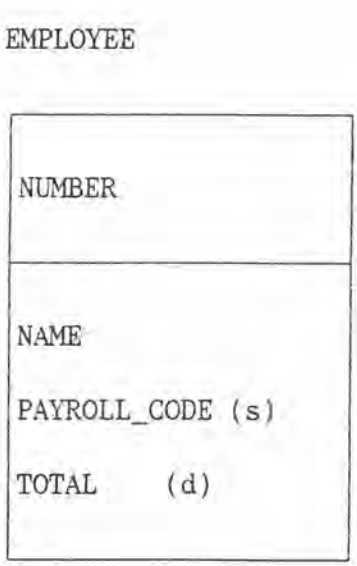
รหัส	ความหมาย
W	รับเงินรายสัปดาห์
M	รับเงินรายเดือน

รหัส	ความหมาย
M	ลูกจ้างที่เป็นผู้จัดการ
E	ลูกจ้างที่เป็นวิศวกร

ตารางที่ 3.2 แสดงแอตตริบิวต์ในรูปของการเข้ารหัส และความหมายของรหัสโดยที่รหัสนั้นมีค่าที่เป็นอิสระจากกัน

6.9 แอตตริบิวต์ต่าง ๆ ที่เป็นดิไรฟว์แอตตริบิวต์ คือแอตตริบิวต์ที่สามารถหาค่าได้ ในการคำนวณอาจต้องใช้สูตร หรืออัลกอริทึม จึงต้องบันทึกอัลกอริทึม หรือสูตรในการคำนวณลงในพจนานุกรมด้วย (ใส่ตัวอักษร d ไว้ที่ท้ายชื่อที่เป็นดิไรฟว์แอตตริบิวต์ในแผนภาพของแบบจำลองข้อมูลเชิงตรรก ดังรูปที่ 3.11 ข้างล่างนี้)

6.10 กำหนดสัญญลักษณ์พิเศษให้กับแอตตริบิวต์ที่เป็นซับไทป์ (Subtype Identifierd) โดยใส่ตัวอักษร s ไว้ที่ท้ายชื่อแอตตริบิวต์ในแผนภาพของแบบจำลองข้อมูลเชิงตรรก ดังรูปที่ 3.11 ข้างล่างนี้



รูปที่ 3.11 สัญญลักษณ์ของแอตตริบิวต์ที่เป็นเป็นซับไทป์ (s) และ
ดิไรฟว์แอตตริบิวต์ (d)

6.11 ให้ใส่แอตตริบิวต์ที่ใช้ร่วมกันระหว่างเอนทิตีที่เป็นซูเปอร์ไทป์ และเอนทิตีที่เป็นซับไทป์ไว้ที่เอนทิตีที่เป็นซูเปอร์ไทป์

6.12 ให้รวมเอนทิตีที่มีคีย์หลักเหมือนกันเอาไว้ด้วยกัน แต่เอนทิตีที่จะนำมารวมกันได้นั้นจะต้องแทนสิ่ง ๆ เดียวกัน

6.13 รวมเอนทิตีที่เป็นซับไทป์ที่มีแอตตริบิวต์ และความสัมพันธ์ที่เหมือนกันเอาไว้เป็นเอนทิตีเดียวกัน

6.14 ให้รวมเอนทิตีที่เป็นซูเปอร์ไทป์ และใช้เอนติตีที่เป็นซับไทป์ร่วมกันไว้ที่เอนทิตีซูเปอร์ไทป์รวม

6.15 ให้รวมเอนทิตีที่ไม่มีแอตตริบิวต์ใดเป็นคีย์เลย (Nonkey Attribute) ไว้ที่เอนติตีลูก

7. พิจารณาให้อยู่ในกฎการทำให้เป็นบรรทัดฐาน (Validate Normalization Rules : LDM7)

การทำให้เป็นบรรทัดฐานเป็นทฤษฎีของการวิเคราะห์ และแยกโครงสร้างของข้อมูลให้เป็นก่ลุม ๆ ของความัสมัพน์ธีที่มคุณสมับิตตาม้ตองการ การทำให้เป็นบรรัทดฐานไมทำให้สารสนเทศต่าง ๆ หายไป หรือทำให้เกิดสารสนเทศใหม่ ๆ ที่ไม่จริงขึ้นมา แบบจำลองข้อมูลที่ผ่านขบวนการการทำให้เป็นบรรทัดฐานแล้วนั้นจะเป็นแบบจำลองข้อมูลเชิงตรรกที่ดี ซึ่งทำให้การออกแบบฐานข้อมูลเป็นไปได้อย่างง่าย ถูกต้อง คงที่ ไม่ซ้ำซ้อน และมี เสถียรภาพ

ประโยชน์ของการทำให้เป็นบรรทัดฐาน (Normalization) มีดังต่อไปนี้

1. ลดเนื้อที่ที่ใช้เก็บข้อมูล
2. ลดการเกิดความไม่คงที่ภายในฐานข้อมูล
3. ลดความเป็นไปได้ในการปรับปรุง และลบข้อมูลที่ผิด
4. เพิ่มเสถียรภาพของโครงสร้างข้อมูล

การทำให้เป็นบรรทัดฐาน (Normalization) ประกอบไปด้วยขั้นตอนต่าง ๆ ดังต่อไปนี้

7.1 การทำให้เป็นบรรทัดฐานขั้นที่ 1 (First Normal Form หรือ 1NF)

ขจัดแอตตริบิวต์ที่ซ้ำ ๆ กันให้ไปอยู่ในเอนทิตีลูก เพื่อให้เอนทิตีแต่ละรายการไม่มีแอตตริบิวต์ หรือกลุ่มของแอตตริบิวต์ที่ซ้ำ ๆ กัน (Repeating Group)

จุดประสงค์ของการทำให้เป็นบรรทัดฐานขั้นที่ 1 นี้มีเหตุผลสำคัญ 3 ประการ คือ

1. ทำให้โครงสร้างของข้อมูลง่ายขึ้น

2. ทำให้สามารถใช้เทคนิคของการทำให้เป็นบรรทัดฐานในขั้นต่อไปได้

3. ทำให้มีความสามารถในการแปลงโครงสร้างของฐานข้อมูลแบบใด ๆ ก็ได้โดยไม่สูญเสียหน้าที่ต่าง ๆ ของโครงสร้างของฐานข้อมูลเหล่านั้น

7.2 การทำให้เป็นบรรทัดฐานขั้นที่ 2 (Second Normal Form หรือ 2NF)

ขจัดแอตตริบิวต์ของเอนทิตีที่อยู่ใน 1NF ที่ไม่ขึ้นกับทั้งส่วนของคีย์หลักทั้งหมดออกเพื่อให้แอตตริบิวต์ที่เหลืออยู่ขึ้นตรงกับส่วนที่เป็นคีย์หลักทั้งหมดเท่านั้น

7.3 การทำให้เป็นบรรทัดฐานขั้นที่ 3 (Third Normal Form หรือ 3NF)

ให้กำจัดแอตตริบิวต์ของเอนทิตีที่อยู่ใน 2NF ที่ไม่ใช่คีย์และขึ้นตรงกับแอตตริบิวอื่นที่ไม่ใช่คีย์หลักออกไป เพื่อให้แอตตริบิวที่ไม่ใช่คีย์หลักต้องขึ้นตรงทั้งส่วนต่อคีย์หลักเท่านั้นและไม่ขึ้นกับแอตตริบิวต์อื่นที่ไม่ใช่คีย์หลัก

7.4. บอยซ์/คอดด์นอร์มัลไล (Boyce/codd Normal Form BCNF) เอนทิตีใด ๆ จะอยู่ใน BCNF ก็ต่อเมื่อทุกแอตตริบิวต์นั้นถูกกำหนดโดยทุกส่วนของแคนดิเดทคีย์ (คีย์หลัก และคีย์รอง) ไม่ใช่ขึ้นอยู่กับส่วนใดส่วนหนึ่งของแคนดิเดทคีย์

7.5 การทำให้เป็นบรรทัดฐานขั้นที่ 4 (Fourth Normal form 4NF)

กำจัดแอตตริบิวต์ของเอนทิตีใน 3NF ที่ขึ้นอยู่กับค่าหลายค่าของคีย์หลักออกเป็นเอนทิตีใหม่อย่างน้อยเป็น 2 เอนทิตี

7.6 การทำให้เป็นบรรทัดฐานขั้นที่ 5 (Fifth Normal form 5NF)

เอนทิตีใด ๆ จะอยู่ใน 5NF ได้ต้องแยกเอนทิตีใน 4 NF ที่มีคีย์หลักหลาย ๆ คีย์ออกเป็นเอนทิตีใหม่หลาย ๆ เอนทิตี

8. กำหนดโดเมน (Determine Domains :LDM8)

โดเมน หมายถึง กลุ่มของช่วงค่าต่าง ๆ ที่เป็นไปได้ของแอตตริบิวต์ใดแอตตริบิวหนึ่งการกำหนดโดเมนให้กับแต่ละแอตตริบิวมีวิธีการดังต่อไปนี้

8.1 กำหนดโดเมน หรือกลุ่มของโดเมน หรือคุณสมบัติของโดเมนให้แต่ละแอตตริบิวต์ดังนี้

8.1.1 ชนิดของข้อมูล (Data Type) กำหนดว่าเป็นข้อมูลชนิดใด เช่นจำนวนเต็ม (Integer) ทศนิยม (Decimal) หรือ อักขระ (Character)

8.1.2 ความยาว (Length)

8.1.3 รูปแบบ (Format)

8.1.4 ความเป็นเอกลักษณ์ (Uniqueness)

8.1.5 ต้องมีมีค่าหรือไม่ (Null support)

8.1.6 มีค่าโดยปริยายหรือไม่ (Default values)

8.1.7 ค่าที่ยอมให้เป็นได้ (Allowable Values)

8.1.8 ความหมาย หรือ คำอธิบาย (Meaning)

8.1.9 ขอบเขต หรือ พิสัย (Range)

คุณสมบัติบางประการของโดเมน จะเป็นอิสระจากเอนทิตีที่แอตตริบิวต์นั้นปรากฏอยู่ คุณสมบัติหล่านี้ได้แก่ ชนิดของข้อมูล และความยาว ควรจะเหมือนกัน ตลอดในแบบจำลองข้อมูลเชิงตรรก แต่ก็มีคุณสมบัติบางประการของโดเมนที่ขึ้นอยู่กับเอนทิตีที่แอตตริบิวต์เหล่านั้นปรากฏอยู่ ได้แก่คุณสมบัติความเป็นเอกลักษณ์ เช่น เมื่อแอตตริบิวต์หนึ่งอยู่ในเอนทิตีที่มีแอตตริบิวต์นี้เป็นคีย์หลักดังนั้น จึงต้องมีคุณสมบัติความเป็นเอกลักษณ์ แต่เมื่อแอตตริบิวต์นี้ไปปรากฏอยู่ที่เอนทิตีอื่น แอตตริบิวต์นี้ไม่จำเป็นต้องมีคุณสมบัติความเป็นเอกลักษณ์ก็ได้

8.2 บันทึกโดเมน หรือกลุ่มของคุณสมบัติต่าง ๆ ของโดเมนของแต่ละแอตตริบิวต์ลงในพจนานุกรมข้อมูล

8.3 กำหนดโดเมนของคีย์หลัก ให้พิจารณาตามกฎต่าง ๆ ดังนี้

8.3.1 คีย์หลักต้องมีค่าเอกลักษณ์เป็นหนึ่งเดียว

8.3.2 แอตตริบิวต์ที่เป็นส่วนประกอบของคีย์หลักไม่ต้องมีค่าความเป็นเอกลักษณ์

8.3.3 คีย์หลัก หรือส่วนประกอบของคีย์หลักจะมีค่าเป็นนัลไม่ได้

8.3.4 คีย์หลัก และส่วนประกอบของคีย์หลักต้องรับค่าโดยปริยายได้แต่ต้องคงไว้ซึ่งความเป็นเอกลักษณ์

8.4 การกำหนดโดเมนของคีย์รองโดยให้พิจารณาตามกฎต่าง ๆ ดังนี้

8.4.1 คีย์รองต้องมีค่าความเป็นเอกลักษณ์

8.4.2 แอตตริบิวต์ที่เป็นส่วนประกอบของคีย์รองไม่ต้องมีค่าความเป็นเอกลักษณ์

8.4.3 คีย์รองและส่วนประกอบของคีย์รองอาจมีค่าเป็นนัลได้

8.4.4 คีย์รองและส่วนประกอบของคีย์รองต้องรับค่าโดยปริยายได้แต่ต้องคงไว้ซึ่งความเป็นเอกลักษณ์

คุณสมบัติของคีย์รองต่างจากคีย์หลักตรงที่ว่า คีย์รองมีค่าเป็นนัลได้

8.5 กำหนดโดเมนของฟอร์เรนจ์คีย์ ให้พิจารณาตามกฎต่าง ๆ ดังนี้

8.5.1 ชนิดของข้อมูล ความยาว และรูปแบบของฟอร์เรนจ์คีย์จะต้องเหมือนกับชนิดของข้อมูล ความยาว และรูปแบบเมื่อเป็นคีย์หลักในเอนทิตีแม่

8.5.2 คุณสมบัติความเป็นเอกลักษณ์ของฟอร์เรนจ์คีย์จะขึ้นอยู่กับชนิดของความสัมพันธ์ ถ้าความสัมพันธ์เป็นแบบหนึ่งต่อหนึ่ง ฟอร์เรนจ์คีย์จะต้องมีคุณสมบัติความเป็นเอกลักษณ์ แต่ถ้าความสัมพันธ์เป็นแบบหนึ่งต่อหลาย ฟอร์เรนจ์คีย์จะไม่มีคุณสมบัติความเป็นเอกลักษณ์

8.6 กำหนดโดเมนของดิไรฟ์แอตตริบิวต์ ให้พิจารณาตามกฎต่าง ๆ ดังนี้

8.6.1 ให้กำหนดอัลกอริทึมของดิไรฟ์แอตตริบิวต์ไว้ที่คุณสมบัติของโดเมน ตรงค่าที่ยอมให้เป็นได้

8.6.2 ชนิดของข้อมูลของดิไรฟ์แอตตริบิวต์จะต้องมีชนิดเดียวกันกับแอตตริบิวต์ ที่เป็นแหล่งที่มาของดิไรฟว์แอตตริบิวต์นี้ มิฉะนั้นอาจถูกกำหนดโดย อัลกอริทึม

8.6.3 ความหมายของดิไรฟว์แอตตริบิวกำหนดได้จาก อัลกอริทึม และจากความหมายของแอตตริบิวที่เป็นแหล่งที่มาของดิไรฟว์แอตตริบิวนั้น

8.7 กำหนดโดเมนของคีย์หลักของเอนทิตีที่เป็นซับไทป์ โดยที่โดเมนนี้จะต้องเป็นซับเซตของโดเมนหรือกลุ่มโดเมนที่เป็นคีย์หลักของเอนทิตีที่เป็นซูเปอร์ไทป์ซึ่งจะมีคุณสมบัติดังต่อไปนี้

8.7.1 ชนิดของข้อมูล ความยาว และรูปแบบ จะต้องเหมือนกับคีย์หลักของเอนทิตีที่เป็นซูเปอร์ไทป์

8.7.2 ค่าที่ยอมให้เป็นไปได้จะต้องขึ้นกับตัวระบุซับไทป์ ว่าในแบบจำลองข้อมูลเชิงตรรก มีตัวระบุซับไทป์ให้เป็นไปได้หรือไม่

8.7.3 ความหมายจะต้องเหมือนกับคีย์หลักของซูเปอร์ไทป์แต่ขึ้นอยู่กับตัวระบุซับไทป์

8.7.4 จะต้องมีคุณสมบัติความเป็นเอกลักษณ์

8.7.5 ส่วนประกอบของคีย์หลักไม่ต้องมีคุณสมบัติความเป็นเอกลักษณ์

8.7.6 ห้ามมีค่าเป็นนัล

8.7.7 สามารถกำหนดค่าให้เป็นค่าโดยปริยายได้

นอกจากกำหนดโดเมนให้กับแอตตริบิวต์ที่เป็นคีย์ดังที่กล่าวมาข้างต้นแล้วนั้น จะต้องกำหนดโดเมนให้กับแอตตริบิวอื่น ๆ ที่ปรากฏอยู่ในแบบจำลองข้อมูลเชิงตรรกอีกด้วยเช่นกัน

9. กำหนดทริกเกอร์ดำเนินการ (Triggering operations :LDM9)

ทริกเกอร์ดำเนินการเป็นกฎที่ควบคุมการเป็นไปได้ของการใช้คำสั่งในการแทรก การลบ การปรับปรุง และ การดึงข้อมูล รวมทั้งผลลัพธ์ของการทำคำสั่งเหล่านี้ที่มีผลกระทบกับเอนทิตีอื่น ๆ หรือ แอตตริบิวต์อื่น ๆ ในเอนทิตีเดียวกัน

การกำหนดทริกเกอร์ดำเนินการนี้ จะทำให้ค่าต่าง ๆ ของแอตตริบิวต์มีความเป็นบูรณภาพ และมีความคงที่

การกำหนดทริกเกอร์ดำเนินการนี้ อาจส่งผลกระทบต่อเงื่อนไขต่าง ๆ ดังนี้

1. เงื่อนไขที่เกี่ยวพันกับแอตตริบิวต์หลายตัวในหลายเอนทิตี
2. เงื่อนไขที่เกี่ยวพันกับแอตตริบิวต์ภายในเอนทิตีเดียวกันตั้งแต่ 2 แอตตริบิวต์ขึ้นไป
3. เงื่อนไขที่เกี่ยวพันกับแอตตริบิวต์เดียว ในเอนทิตีเดียว หรือ ตัวแปรเสริมตัวใดตัวหนึ่ง

ทริกเกอร์ดำเนินการมีส่วนประกอบทางตรรกภาพที่สำคัญ 2 ส่วน คือ

1. เหตุการณ์และเงื่อนไขที่ทำให้เกิดคำสั่งต่าง ๆ ขึ้น (The Trigger)
2. การกระทำ และเงื่อนไขเมื่อเกิดเหตุการณ์ขึ้นแล้ว (The Operation)

ให้บันทึกทุก ๆ ทริกเกอร์ดำเนินการลงในพจนานุกรมข้อมูลโดยสิ่งที่ควรบันทึกมีดังต่อไปนี้

9.1 เหตุการณ์เริ่มต้นที่ทำให้ทริกเกอร์ดำเนินการนั้นทำงาน เช่น การแทรก การลบ

9.2สิ่งต่าง ๆ ที่เกี่ยวข้องเมื่อเหตุการณ์นั้นเกิดขึ้นแล้ว เช่นบันทึกชื่อของเอนทิตี หรือแอตตริบิวที่จะถูกเปลี่ยนแปลง หรือถูกเข้าถึง

9.3 เงื่อนไขต่าง ๆ ที่จะทำให้เหตุการณ์ดังกล่าวเกิดขึ้น

9.4 ผลลัพธ์ หรือ การกระทำเมื่อเหตุการณ์นั้นเกิดขึ้นแล้ว

นอกจากกำหนดทริกเกอร์ดำเนินการให้แก่ แอตตริบิวต์ต่าง ๆ แล้วยังต้องกำหนดให้กับแอตตริบิวต์ที่เป็นแหล่งที่มาของดิไรฟว์แอตตริบิวต์อีกด้วย ในกรณีที่เป็นซับไทป์ ให้กำหนดทริกเกอร์ดำเนินการต่าง ๆ เช่น เมื่อค่าต่าง ๆ ในเอนทิตีที่เป็นซับไทป์ถูกลบ ค่าที่ตรงกับค่าที่ถูกลบในเอนทิตีที่เป็นซับไทป์จะต้องถูกลบในเอนทิตีที่เป็นซูเปอร์ไทป์ด้วย

10. **รวบรวมมุมมองของผู้ใช้ทั้งหมดเข้าด้วยกัน (Combind user views:LDM10)**

ขั้นตอนต่าง ๆ ที่ได้กล่าวมาแล้วตั้งแต่ขั้นตอนที่ 1 ถึงขั้นตอนที่ 9 นั้น เป็นการพัฒนารายละเอียดความต้องการสารสนเทศของ ผู้ใช้แต่ละคน กลุ่มของผู้ใช้ หรือ หน้าที่ส่วนใดส่วนหนึ่งของระบบเท่านั้น ดังนั้นจะพบว่าในระบบเมื่อรวบรวมความต้องการต่าง ๆ จากผู้ใช้หลายคน หรือหลายมุมมองแล้วนั้น อาจทำให้เกิดความต้องการที่เลื่อมล้ำ หรือ ซ้ำซ้อนกันเกิดขึ้นได้

จุดประสงค์ในการรวบรวมมุมมองของผู้ใช้

1. เพื่อเป็นตัวแทนของมุมมองต่าง ๆ ได้อย่างถูกต้อง
2. เพื่อขจัดความซ้ำซ้อน
3. เพื่อแก้ปัญหาความไม่คงที่ ที่มีขึ้นระหว่างมุมมองต่าง ๆ
4. เพื่อเพิ่มความสัมพันธ์ใหม่ ๆ ที่เกิดขึ้นมาระหว่างมุมมองต่าง ๆ

การรวบรวมมุมมองต่าง ๆ ของผู้ใช้เข้าด้วยกันนั้นจะต้องทำตามขั้นตอนต่าง ๆ 3 ขั้นตอนดังต่อไปนี้

1. รวบรวมเอนทิตี และกฎธุรกิจต่าง ๆ ที่เกี่ยวข้องเข้าด้วยกันโดยพิจารณาจากคีย์หลักและคีย์รอง

2. รวบรวมความสัมพันธ์ และกฎธุรกิจต่าง ๆ ที่เกี่ยวข้องเข้าด้วยกันโดยพิจารณาจากกฎการเพิ่มหรือการลบ

3. รวบรวมแอตตริบิวต์ และกฎธุรกิจต่าง ๆ ที่เกี่ยวข้องกันเข้าด้วยกัน โดยพิจารณาโดเมน และ ทริกเกอร์ดำเนินการ

วิธีการในการรวบรวมมุมมองของผู้ใช้นั้นมีวิธีการดังต่อไปนี้

10.1 ให้นำเอนทิตีที่มีคีย์หลักเหมือนกันและมีโดเมนของคีย์หลักที่เทียบเท่ากันมาไว้รวมกัน โดยให้นำแอตตริบิวต์ของเอนทิตีที่นำมารวมกันนั้นไว้ด้วยกันและขจัดแอตตริบิวต์ที่ซ้ำกันออก จะต้องไม่รวมเอนทิตีที่เป็นซับไทป์ และเอนทิตีที่เป็นซูเปอร์ไทป์ในมุมมองของผู้ใช้เดียวกันไว้ด้วยกัน ถึงแม้ว่าจะมีคีย์หลักเหมือนกันก็ตาม แต่ในทางกลับกันอาจจะรวมเอนทิตีที่เป็นซับไทป์และเอนทิตีที่เป็นซูเปอร์ไทป์ที่อยู่ต่าง ๆ มุมมองของผู้ใช้เข้าไว้ด้วยกันก็ได้

10.2 ให้ตั้งความสัมพันธ์ระหว่างเอนติตีที่เป็นซับไทป์ และเอนติตีที่เป็นซูเปอร์ไทป์ขึ้นมาระหว่างเอนติตีที่มีคีย์หลักเหมือนกัน โดยที่โดเมนของคีย์หลักของเอนติตีที่จะมาเป็นซับไทป์นั้นต้องเป็นซับเซตของโดเมนของคีย์หลักของเอนติตีที่เป็นซูเปอร์ไทป์ และขจัดแอตตริบิวต์ในเอนทิตีที่เป็นซับไทป์ออก ถ้าพบว่ามีอยู่แล้วในเอนทิตีที่เป็นซูเปอร์ไทป์

10.3 เมื่อพบว่ามี 2 เอนทิตีที่มีคีย์หลักเหมือนกันและโดเมนของคีย์หลักมีค่าเท่าเทียมกัน ให้ตั้งเอนทิตีนั้นเป็นซูเปอร์ไทป์

10.4 รวมเอนทิตีที่มีคีย์หลักที่เป็นแคนดิเดทคีย์ของอีกเอนทิตีหนึ่ง เอาไว้เป็นเอนทิตีเดียวกัน โดยให้เลือกว่าจะใช้แอตตริบิวต์ตัวใดเป็นคีย์หลักและในขณะเดียวกันให้ขจัดแอตตริบิวต์ที่ซ้ำกันออกด้วย

10.5 ให้คงเอนทิตีใด ๆ ที่มีคีย์หลักแตกต่างกัน ซึ่งไม่สามารถรวมกับเอนทิตีอื่น ๆ ได้เหล่านั้นให้อยู่อย่างเดิมโดยไม่ต้องเปลี่ยนแปลงใด ๆ ทั้งสิ้น

10.6 ให้คงกฎธุรกิจที่เกี่ยวกับแคนดิเดทคีย์เอาไว้ แต่ในบางกรณีอาจยกเว้น เช่น จากเดิมซึ่งเคยเป็นคีย์หลัก แต่เมื่อรวบรวมมุมมองของผู้ใช้แล้วได้กลายมาเป็นคีย์รอง ดังนั้นแอตตริบิวต์นี้สามารถมีค่าเป็นนัลได้ เป็นต้น

10.7 เมื่อรวบรวมเอนทิตีต่าง ๆ เข้าด้วยกันนั้น ความสัมพันธ์ก็จะถูกรวบรวมเข้าด้วยกันเมื่อความสัมพันธ์ระหว่างเอนทิตีเหล่านั้นยังคงมีความหมายเหมือนเดิม แต่สัดส่วนความสัมพันธ์อาจเปลี่ยนไปจากเดิม

แต่ถ้าความสัมพันธ์ในแต่ละมุมมองของผู้ใช้ไม่เหมือนกันก็จะไม่รวมความสัมพันธ์เหล่านั้นเข้าด้วยกัน

10.8 ให้คงความสัมพันธ์ที่มีความหมายแตกต่าง ๆ กันไว้อย่างเดิมและให้ขจัดความสัมพันธ์ที่ซ้ำซ้อนออกไป

10.9 เพิ่มความสัมพันธ์ใหม่ขึ้นมาแทนความสัมพันธ์ที่ขาดหายไป

10.10 พิจารณาฟอร์เรนจ์คีย์ที่รวบรวมไว้แล้วให้ถูกต้อง

10.11 ให้รวมเอาคีย์ของกฎธุรกิจที่กำหนดไว้ในมุมมองของผู้ใช้เดิมที่ยังไม่ได้ถูกรวบรวมเอาไว้ด้วย และเพิ่มคีย์ของกฎธุรกิจสำหรับความสัมพันธ์ที่เพิ่มขึ้นมาใหม่นั้นด้วย

10.12 รวมแอตตริบิวต์ที่มีความหมายเดียวกันที่อยู่ในเอนทิตีเดียวกันเข้าไว้ด้วยกัน โดยให้มีโดเมน และทริกเกอร์ดำเนินการที่เข้ากันได้

10.13 ให้พิจารณาแอตตริบิวต์ต่าง ๆ ในมุมมองของผู้ใช้ที่รวบรวมแล้วว่าเป็นดิไรฟว์แอตตริบิวต์ หรือไม่ และจะหาค่าผลลัพธ์ หรืออัลกอริทึมจากมุมมองของผู้ใช้ได้จากแอตตริบิวต์อื่นในมุมมองของผู้ใช้อีกมุมมองหนึ่ง

10.14 ให้ทำการให้เป็นบรรทัดฐานนอร์อีกครั้งหนึ่งหลังจากที่รวบรวมเอนติตีเข้าด้วยกันแล้ว ขจัดหรือเพิ่มความสัมพันธ์ให้เหมาะสมเพื่อขจัดความซ้ำซ้อน

10.15 ให้พิจารณาโดเมนของแอตตริบิวต์ต่าง ๆ อีกครั้งหนึ่งว่าสามารถมีค่าเป็นนัลได้หรือไม่ เพื่อที่จะสร้างเอนติตีใหม่ที่เป็นซับไทป์ขึ้นมาเพื่อลดค่าการเป็นนัลนั้น

11. รวบรวมแบบจำลองข้อมูลที่มีอยู่แล้วเข้าด้วยกัน (Intergrate With Existing Data Models :LDM11)

ขั้นตอนนี้จะระบุการเชื่อมโยงระหว่างกัน การปฏิบัติงานของแบบจำลองข้อมูลเฉพาะบริเวณ แบบแผนของฐานข้อมูลในเมโนภาพทางธุรกิจ ทำให้ได้แบบจำลองข้อมูลหลายรูปแบบ

จุดประสงค์ที่ต้องการรวบรวมนั้นไม่จำเป็นที่จะต้องจัดการกับความแตกต่างที่มีอยู่ให้เข้ากันได้ เพียงแต่เข้าใจ และ บันทึกความสัมพันธ์ รวมทั้งความไม่คงที่ต่าง ๆ ที่เกิดขึ้นระหว่าง แบบจำลองข้อมูลเชิงตรรกนั้น

12. วิเคราะห์ความมีเสถียรภาพและการเปลี่ยนแปลงที่อาจเกิดขึ้นในอนาคต (Analyze For Stability And Growth :LDM12)

ในขั้นตอนนี้จะพิจารณาสิ่งที่จะเกิดขึ้นในอนาคตต่อไปด้วย เช่นจะมีการเปลี่ยนแปลงในแบบจำลองข้อมูลเชิงตรรกอย่างไร ถ้าในระบบมีการดำเนินการต่าง ๆ เปลี่ยนแปลงไป หรือมุมมองต่าง ๆ ของผู้ใช้ถูกเปลี่ยนไป เช่น

1. จะต้องเพิ่มเอนทิตี หรือรีเลชันชิปขึ้นใหม่หรือไม่ แล้วจะไปกระทบต่อเอนทิตีเดิมที่มีอยู่ซึ่งอาจต้องเพิ่มฟอร์เรนจ์คีย์ใหม่ลงไปหรือไม่

2. จะต้องเปลี่ยนแปลงมีคีย์หลักหรือไม่ เนื่องจากต้องการให้มีความเอกลักษณ์ ซึ่งจะส่งผลกระทบต่อฟอร์เรนจ์คีย์ในเอนทิตีที่สัมพันธ์กัน

3. จะต้องเพิ่มเอนทิตีที่เป็นซับไทป์หรือไม่

4. จะต้องเพิ่มแอตตริบิวใหม่หรือไม่ หรือขจัดแอตตริบิวเดิมที่มีอยู่แล้ว

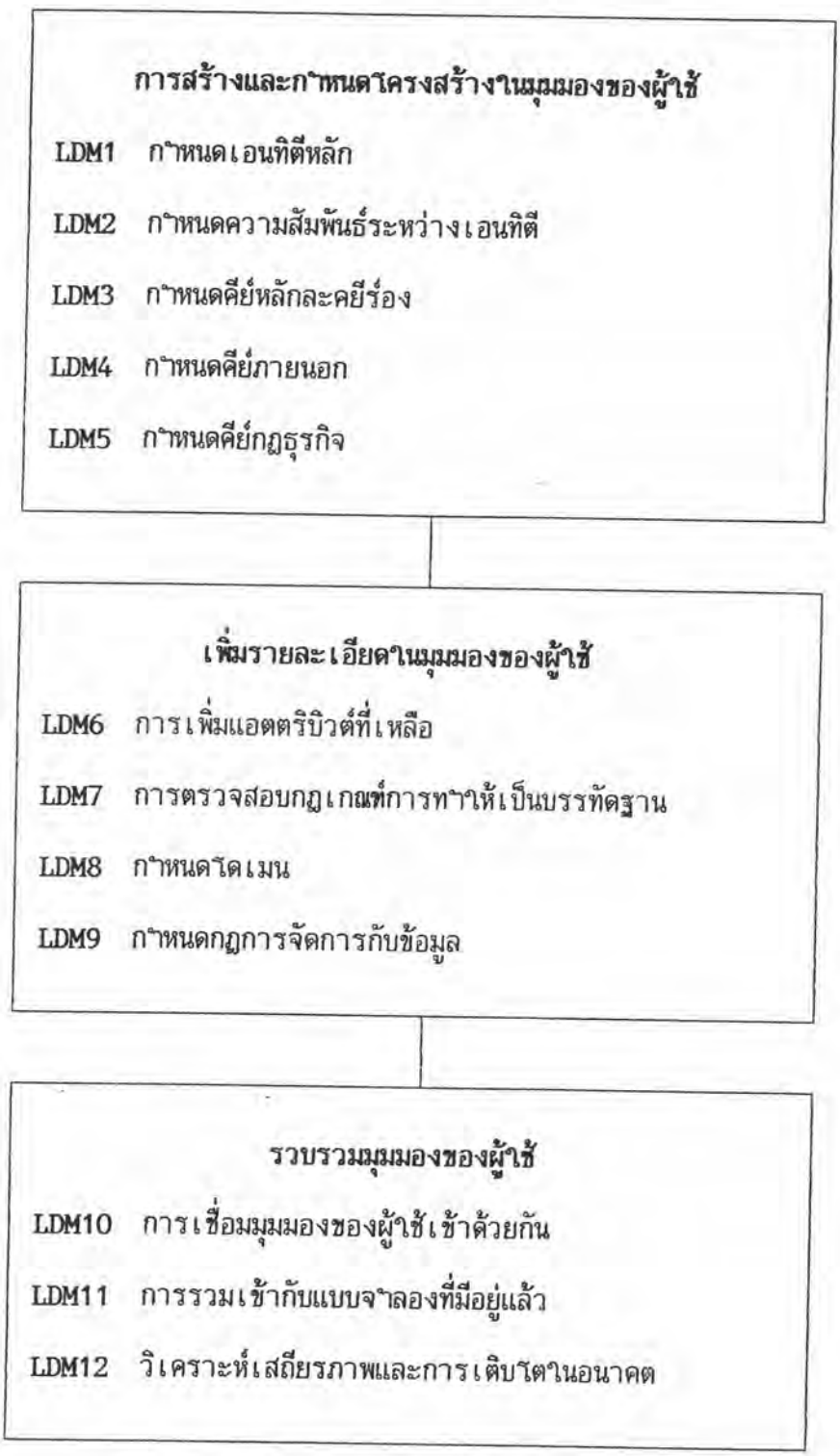
5. จะต้องเปลี่ยนแปลงขนาดของเอนทิตี หรือไม่

6. จะต้องเปลี่ยนแปลงสัดส่วนของความสัมพันธ์หรือไม่

7. จะต้องเปลี่ยนแปลงกฎธุรกิจหรือไม่ เช่น กฎการเพิ่ม กฎการลบ โดเมน หรือทริกเกอร์ดำเนินการ

ความมีเสถียรภาพ และการเจริญเติบโตของแบบจำลองข้อมูลเชิงตรรกนั้นขึ้นอยู่กับความสามารถของผู้พัฒนาระบบ เกี่ยวกับการคาดคะเนถึงการเปลี่ยนแปลงความต้องการต่าง ๆ ของผู้ใช้ในอนาคตที่จะเกิดมีขึ้นมาใหม่

ขั้นตอนการออกแบบจำลองข้อมูลเชิงตรรก



รูป 3.12 ขั้นตอนการออกแบบจำลองข้อมูลเชิงตรรก