บทที่ 2

แนวเหตุผลและทฤษฎีสำคัญ

2.1 การออกแบบระบบพจนานุกรมข้อมูล

การออกแบบระบบพจนานุกรมข้อมูลเพื่อใช้เป็นแหล่งบันทึกข้อมูลจากการวิเคราะห์ฐานข้อมูล เป็นเครื่องมือในการระบุและอธิบายลักษณะสมบัติของข้อมูลทั้งหมด ซึ่งปรากฏในโมเดลข้อมูล การจัดองค์กรของข้อมูลอาศัยหลักการของระบบฐานข้อมูลแบบรีเลชันนัล (Relational Database System) ในช่วงการออกแบบฐานข้อมูล เพื่อใช้เป็นเครื่องมือช่วยในการพัฒนาระบบฐานข้อมูลต่อไป

สารสนเทศที่ได้จากพจนานุกรมข้อมูลเป็นข้อมูลที่เก็บบันทึกอยู่ในฐานข้อมูล ภายใต้การจัดการของระบบจัดการฐานข้อมูล (DBMS) เรียกลักษณะเช่นนี้ว่าเป็นระบบพจนานุกรมข้อมูลที่ขึ้นกับการทำงานของระบบการจัดการฐานข้อมูล (Dependent Data Dictionary System) (Atre, 1988) การพัฒนาระบบพจนานุกรมข้อมูลที่กล่าวถึงนี้อาจมีฐานข้อมูลอยู่ 2 ชุดที่มีเนื้อหาซ้ำซ้อนกันอยู่ คือฐานข้อมูลของระบบพจนานุกรมข้อมูลที่เกิดขึ้นเองจากระบบการจัดการฐานข้อมูล และฐานข้อมูลของระบบพจนานุกรมข้อมูลที่เกิดเนื่องจากการพัฒนา ซึ่งจำเป็นต้องมีการดูแลและควบคุมให้ฐานข้อมูลทั้ง 2 ชุดมีเนื้อหาสอดคล้องกัน และไม่พบความเหลื่อมล้ำของข้อมูลเมื่อเกิดการเปลี่ยนแปลงไม่ว่าในที่ใดที่หนึ่งขึ้น เรียกระบบนี้ว่าระบบพจนานุกรมแบบแพชชิฟ (Passive Data Dictionaries System) (Brathwaite, 1985)

คุณสมบัติที่สำคัญของระบบพจนานุกรมข้อมูล

1. เป็นตัวตรวจสอบความถูกต้องของข้อมูล (validation check), ตรวจสอบความไม่ขัดแย้งกัน (consistency check), ตรวจสอบความเหลือเฟือ (redundancy check), ทำให้ข้อมูลภายในระบบมีความเป็นบูรณภาพ (integrity), มีระบบควบคุมความปลอดภัย (security control) และมีมาตรฐาน (standard) ในการกำหนดรายละเอียดต่างๆ ของข้อมูลที่ทำการบันทึกในระบบนี้ (Van Duyn,1982)

2. มีลักษณะเป็นศูนย์รวมข้อมูลที่เกี่ยวข้องกับข้อมูลของหน่วยงานต่างๆ เรียกลักษณะนี้ว่าข้อมูลแบบเมทา (meta-data) (Marti,1983)

3. สามารถนำไปอ้างอิงกับระบบฐานข้อมูลแบบรีเลชันนัล ใช้แสดงแทนโครงสร้างของสารสนเทศ และพัฒนาโมเดลข้อมูลเชิงมโนภาพในระดับเฉพาะบริเวณและระดับส่วนกลาง (Local and global conceptual data models) ในช่วงการออกแบบและวิเคราะห์ข้อมูล เป็นการรวบรวมข้อมูลที่ผ่านการวิเคราะห์ตามหน้าที่ (functional analysis) โดยผ่านขบวนการนอร์มัลไลเซชัน (normalization) แล้ว (Maddison,1985)

4. มีการควบคุมและจัดการเมื่อมีการอ้างอิงถึงข้อมูลส่วนอื่น ต้องสามารถตรวจสอบค่าที่อ้างอิงว่ามีอยู่จริงและเป็นค่าที่ถูกต้อง เรียกลักษณะนี้ว่า เกณฑ์ในการอ้างอิงให้สอดคล้อง (referential constraints) (Marti,1983)

5. มีลักษณะเป็นออนไลน์ (online) เมื่อผู้ใช้มีความต้องการสืบค้นสารสนเทศหรือต้องการบำรุงรักษาระบบพจนานุกรมข้อมูลในเชิงโต้ตอบ (interactive) ได้แก่

5.1 การป้อนข้อมูลเข้า (online entry)
5.2 การปรับปรุงแฟ้มข้อมูล (online update)
5.3 การตรวจค้นหาข้อมูล (online retrieval)

วัตถุประสงค์หลักของระบบพจนานุกรมข้อมูล

1. ทำให้เกิดความเข้าใจโครงสร้างข้อมูลที่ได้ทำการออกแบบไว้และใช้ตรวจสอบความถูกต้องด้วย
2. ลดความเหลื่อมเพื่อให้น้อยที่สุด
3. ควบคุมการกำหนดข้อมูลไว้ใช้ในระบบฐานข้อมูลให้มีมาตรฐานเดียวกัน
4. เป็นศูนย์กลางควบคุมดูแลข้อกำหนดของข้อมูลให้เกิดบูรณภาพเสมอ ถึงแม้ว่าจะมีการเปลี่ยนแปลงเกิดขึ้นภายหลัง
5. สามารถจัดทำรายงานเฉพาะกิจเกี่ยวกับพจนานุกรมข้อมูลได้ตามความต้องการของผู้ใช้ เพื่อเป็นรายงานไว้ใช้อ้างอิง

การออกแบบและพัฒนาระบบพจนานุกรมข้อมูลอาศัย หลักการใหญ่ ๆ 2 ข้อ คือ

1. การทำโมเดลข้อมูลเชิงตรรก (Logical Data Modeling : LDM)
2. การออกแบบฐานข้อมูลแบบรีเลชันนัล (Relational Database Design: RDD)

ทั้ง 2 หลักการนี้สามารถเชื่อมโยงกันได้ กล่าวคือการออกแบบฐานข้อมูลแบบรีเลชันนัลอาศัยรายละเอียดจากหลักการแรกเป็นพื้นฐานในการดำเนินงานต่อไป โดยผ่านขั้นตอนการแปลจากโมเดลข้อมูลเชิงตรรกเป็นการออกแบบฐานข้อมูลแบบรีเลชันนัล (Translation Process)

ในการเชื่อมโยงนี้เพื่อให้เกิดความไม่ขัดแย้งกัน ต้องมีกลไกในการย้ายสารสนเทศจากระดับเชิงตรรกสู่ระดับกายภาพ ในขั้นตอนพัฒนาลงสู่ระบบการจัดการฐานข้อมูล (Fleming,1989)

2.2 การทำโมเดลข้อมูลเชิงตรรก

เป็นวิธีการแสดงความต้องการสารสนเทศในระบบธุรกิจให้เป็นแผนภาพ ในขั้นตอนการวิเคราะห์และออกแบบระบบฐานข้อมูล เน้นตัวข้อมูลที่มีอยู่จริงโดยไม่คำนึงถึงรายละเอียดในการติดตั้ง, ความต้องการพิเศษอื่นในแง่การใช้งาน และความเร็วในการสืบค้นข้อมูล หรืออาจกล่าวอีกนัยหนึ่งว่าโมเดลข้อมูลเชิงตรรกเป็นการสร้างโครงร่างวิวของผู้ใช้ (skeletal user view)

การสร้างโมเดลข้อมูลเชิงตรรกอาจกระทำได้ 2 ลักษณะคือ

1. จากระดับล่างไปสู่ระดับบน (bottom-up) เป็นการพิจารณาข้อมูลจากรายละเอียดความต้องการของผู้ใช้เฉพาะบริเวณ มักพบบ่อยในช่วงการวิเคราะห์ข้อมูลหรือจากระบบงานเดิมที่มีการประยุกต์ใช้งานอยู่แล้ว ทำให้ได้โมเดลข้อมูลที่มีขอบเขตชัดเจน แต่มักประสบปัญหาเกี่ยวกับการรวมโมเดลข้อมูลแบบเบ็ดเสร็จคือไม่สามารถหาจุดเชื่อมระหว่างกันได้ ทำให้ไม่สามารถสะท้อนถึงกิจกรรมทางธุรกิจทั้งหมดขององค์กร เรียกลักษณะนี้ว่าเป็นการรวบรวมวิว
(View Integration)
2. จากระดับบนไปสู่ระดับล่าง (top-down) เป็นการพิจารณาความหมายของข้อมูลทางธุรกิจ เริ่มจากเอนติตีและรีเลชันชิปที่มีบทบาทต่อกิจกรรมทั้งองค์กรในระดับบน ไม่เน้นการประยุกต์ใช้งาน ทำให้สะดวกต่อการรวมโมเดลข้อมูลแบบ

เบ็ดเสร็จ ก่อนที่จะพัฒนารายละเอียดเฉพาะบริเวณ ทำให้ไม่เกิดความขัดแย้งของข้อมูล เรียกลักษณะนี้ว่าเป็นการทำโมเดลของสารสนเทศ (Information Modeling)

การสร้างโมเดลข้อมูลอาจยึดตามหลักการในลักษณะใดก็ได้จาก 2 ลักษณะข้างต้น หรืออาจใช้หลักการทั้ง 2 ผสมผสานกันก็ได้ตามความเหมาะสม

วิวของผู้ใช้ (user view) หมายถึงโมเดลหรือการแทนความต้องการในแง่ผู้ใช้แต่ละคนเทียบเท่ากับการปฏิบัติงานหนึ่งอย่าง เช่นการพิมพ์ใบส่งสินค้าให้ลูกค้ารายหนึ่ง และในที่สุดมีการรวมวิวของผู้ใช้เข้าด้วยกันเป็นส่วนเดียวกัน เรียกว่า โมเดลข้อมูลเชิงตรรกแบบเบ็ดเสร็จ (integrated logical data model)

ลักษณะที่สำคัญของการทำโมเดลข้อมูลเชิงตรรก คือ

1. เป็นการจำลองแผนภาพเพื่อแสดงว่าระบบธุรกิจใดๆ มีการจัดการสารสนเทศและนำมาใช้อย่างไรบ้าง โดยไม่คำนึงถึงความต้องการในแง่ปฏิบัติการหรือเทคนิคใดๆ หัวใจก็คือเป็นการมองว่าข้อมูลคือแหล่งที่มาสำหรับการจัดองค์กรในธุรกิจ (data-driven approach/ technology-independent) ทำให้การออกแบบฐานข้อมูลมีเสถียรภาพและยืดหยุ่นต่อการนำไปพัฒนาระบบฐานข้อมูล เนื่องจากเป็นการเน้นที่ความสัมพันธ์ของข้อมูลเท่านั้น จึงสามารถนำโมเดลที่ได้นี้ไปทำการพัฒนาและติดตั้งด้วยเทคนิคใดก็ได้ ซึ่งยังคงสามารถปฏิบัติงานในเชิงธุรกิจตามความต้องการของผู้ใช้ได้ครบถ้วน

2. เป็นขั้นตอนในการทำเอกสารที่สมบูรณ์แบบเพื่อใช้บันทึกความต้องการต่าง ๆ ทั้งในลักษณะแผนภาพและข้อความอธิบายประกอบ ช่วยให้เข้าใจถึงความต้องการสารสนเทศและความสัมพันธ์ในเชิงธุรกิจได้ชัดเจน และรัดกุมกว่าการอธิบายโดยใช้ข้อความบรรยายแต่เพียงอย่างเดียว

3. เป็นสื่อในการติดต่อระหว่างผู้ออกแบบโมเดล ผู้พัฒนาระบบฐานข้อมูล และผู้ใช้ในระหว่างขั้นตอนการออกแบบได้ดีและมีประสิทธิภาพ เนื่องจากผู้เกี่ยวข้องทุกคนสามารถตรวจสอบความถูกต้องของข้อมูลได้สะดวก

4. เป็นพื้นฐานในการออกแบบ ทำให้เกิดความถูกต้อง (correct), ความไม่ขัดแย้งกัน (consistent) และสามารถใช้ร่วมกันได้ (sharable) ทำให้เกิดฐานข้อมูลที่ยืดหยุ่นได้ไม่ว่าจะเลือกใช้เทคนิคระบบการจัดการฐานข้อมูลแบบใด ๆ ภายใต้ข้อจำกัดหรือความต้องการพิเศษอื่นๆ ที่เกิดขึ้น โดยไม่มีผลกระทบต่อระบบเดิมที่มีอยู่

วิธีการพัฒนาระบบโดยพิจารณาข้อมูล (Data-Driven Systems Development Methodology) มีลักษณะดังนี้

1. อธิบายชนิดของข้อมูลที่มีความสัมพันธ์กันในแง่สภาพแวดล้อมของระบบฐานข้อมูล จากแหล่งที่มาของข้อมูล

2. มีความเป็นอิสระจากความต้องการของผู้ใช้ และในด้านการพัฒนาเพื่อนำไปใช้ (implementation) เช่น รายละเอียดในการเก็บข้อมูล, ระบบปฏิบัติการ ซึ่งเป็นการทำงานในระดับกายภาพ

3. ไม่ขึ้นกับประเภทของระบบการจัดการฐานข้อมูลใดๆ ที่จะนำมาใช้ เพราะอาศัยหลักการของโมเดลข้อมูล

วงจรโครงงานพัฒนาระบบแบบพิจารณาข้อมูล (Data-Driven Project Lifecyle) มีวัตถุประสงค์เพื่อติดตั้งระบบฐานข้อมูลและซอฟต์แวร์ ภายใต้มโนภาพของแบบแผนฐานข้อมูลแบบเบ็ดเสร็จ (integrated conceptual schema) ซึ่งมีความแตกต่างกับวงจรโครงงานพัฒนาระบบแบบสัญนิยม (Conventional Systems Development Lifecyle) ดังนี้

วงจรโครงงานพัฒนาระบบแบบสัญนิยม	วงจรโครงงานพัฒนาระบบแบบพิจารณาข้อมูล
- เป็นการสร้างฐานข้อมูลและแฟ้มข้อมูลแยกกัน	- เป็นการรวมฐานข้อมูลต่างๆ เข้าด้วยกัน
- การพัฒนาต้องทราบความต้องการอย่างสมบูรณ์และถูกต้องก่อนเสมอ โดยทำการสำรวจรายละเอียดต่างๆ จากผู้ใช้ก่อน	- การพัฒนายึดการทำต้นแบบ (prototyping) สามารถทดลองทำต้นแบบควบคู่กันไปได้ในช่วงการพัฒนา เนื่องจากเน้นที่ความสัมพันธ์ของข้อมูล สามารถค้นหาความต้องการของผู้ใช้ โดยผู้ใช้ต้องคอยแนะนำให้คำปรึกษาตลอด

วงจรโครงงานพัฒนาระบบแบบสัญนิยม	วงจรโครงงานพัฒนาระบบแบบพิจารณาข้อมูล
- ไม่ยืดหยุ่นเมื่อมีการเปลี่ยนแปลง หรือถ้าทำการสำรวจไม่รัดกุม อาจทำให้การพัฒนาไม่ตรงตามความต้องการของผู้ใช้ ทำให้สิ้นเปลืองค่าใช้จ่ายในการแก้ปัญหาที่ตามมา	- มีความยืดหยุ่น สามารถทดลองพัฒนาก่อนการออกแบบเพื่อใช้งานจริง เพื่อดูการทำงานว่าสามารถกระทำได้หรือไม่ ถ้าไม่ตรงตามความต้องการของผู้ใช้ ก็สามารถแก้ไขโครงสร้างของฐานข้อมูล ในระดับกายภาพได้ โดยที่โมเดลข้อมูลในระดับตรรกยังคงอยู่ เกิดผลกระทบกับของเดิมน้อย และประหยัดค่าใช้จ่ายในการพัฒนา

ขั้นตอนของวงจรโครงงานพัฒนาระบบแบบพิจารณาข้อมูล ประกอบด้วย 6 ขั้นตอนคือ

1. สำรวจ (survey)
   - เพื่อหาขอบเขตและหมายกำหนดการของโครงงาน
   - รวบรวมและวิเคราะห์สารสนเทศเกี่ยวกับข้อมูลภายในขอบข่ายของโครงงาน
   - พัฒนาขอบเขตโมเดลข้อมูลเชิงตรรกในระดับสูง และระบุถึงการเปลี่ยนแปลงตามระยะเวลาต่างๆ

2. ออกแบบ (design)
   - พัฒนาโมเดลข้อมูลเชิงตรรกโดยละเอียดจากสภาพแวดล้อมใหม่
   - วางแผนขั้นตอนการทำต้นแบบ
   - วาดภาพแสดงขั้นตอนการเกิดรายการเปลี่ยนแปลงต่างๆ ของฐานข้อมูล และบันทึกรายละเอียดของผลกระทบที่จะเกิดกับข้อมูล

3. ทำต้นแบบ (prototype)
   - ออกแบบโครงสร้างของฐานข้อมูลในระดับกายภาพ สำหรับการทำต้นแบบในระยะเริ่มต้น
   - พัฒนาต้นแบบให้สมบูรณ์ตามที่ได้ออกแบบไว้
   - แก้ไขและตรวจสอบให้ถูกต้องตามวิวของผู้ใช้ เพิ่มส่วนของความต้องการที่มีการเปลี่ยนแปลงใหม่ๆ ขึ้น
   - ย้ายต้นแบบเข้าสู่สถานะการทำงาน (working status)

4. ทำการรวมเข้าด้วยกัน (integration)
   - รวบรวมโมเดลข้อมูลเชิงตรรกต้นแบบเข้าด้วยกัน เนื่องจากมิได้ออกแบบแยกกันเป็นเอกเทศหรือเป็นฐานข้อมูลที่แยกกัน ซึ่งแต่ละส่วนคือมโนภาพของแบบแผนฐานข้อมูล (conceptual schema) มีข้อมูลภายใต้ขอบเขตที่สัมพันธ์กัน โดยมีการเปลี่ยนแปลงและมีการขยายตัวอยู่เสมอ
   - ปรับจุดเชื่อมต่อระหว่างโมเดลข้อมูลเชิงตรรกให้เป็นส่วนร่วมระหว่างกัน (interschema mappings)

5. ปรับแต่ง (tuning)
   - ปรับแต่งโครงสร้างของฐานข้อมูลให้เข้ากับซอฟต์แวร์และฮาร์ดแวร์ เพื่อให้การทำงานมีประสิทธิภาพที่สุด

6. ทบทวน (review)
   - ควบคุมคุณภาพของผลงานจากโครงงาน
   - ย้ายต้นแบบสู่สภาวะการทำงานจริง (production status) (Mary,1987)

แผนภาพโมเดลข้อมูลเชิงตรรกประกอบด้วย

1. เอนติตี (entity) แสดงแทนด้วยรูปสี่เหลี่ยม หมายถึง คน สถานที่ สิ่งของ หรือ นามธรรมใดๆ เป็นตัวเก็บบันทึกสารสนเทศ นั่นคือข้อเท็จจริงต่าง ๆ ภายในระบบธุรกิจที่ทำการวิเคราะห์ ใช้ข้อความภาษาอังกฤษระบุชื่อเอนติตีบริเวณเหนือรูปสี่เหลี่ยม

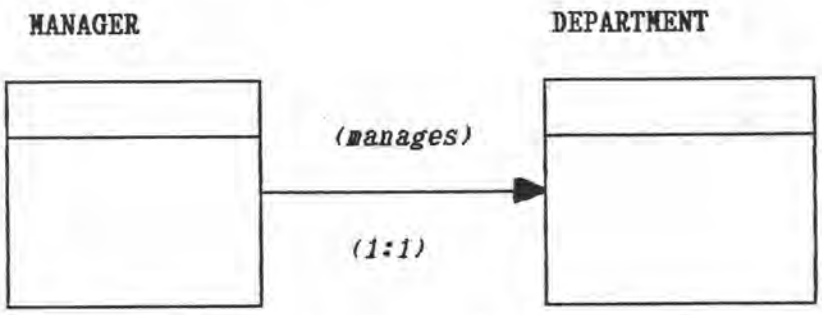
2. รีเลชันชิป (relationship) แสดงแทนด้วยเส้นตรงที่เชื่อมโยงระหว่างสี่เหลี่ยมทั้งสอง หมายถึง ความสัมพันธ์ต่อกันระหว่างเอนติตีทั้งสองแบบไบนารีรีเลชันชิป (binary relationship) ระบุชื่อรีเลชันชิปโดยใช้กริยาวลีอธิบายความหมายที่ปรากฏในแผนภาพให้เกิดความเข้าใจได้ชัดเจนยิ่งขึ้นบริเวณเหนือและใต้เส้นตรง บริเวณปลายเส้นตรงด้านใดด้านหนึ่งเป็นทิศทางของลูกศร มีความหมายคือการโยงความสัมพันธ์จากเอนติตีแม่ (parent entity) ไปยังเอนติตีลูก (child entity) ด้วยหัวลูกศร จำนวนหัวลูกศรแทนความหมายประเภทของรีเลชันชิป จำนวนที่ปรากฏของเอนติตีทั้งสองของรีเลชันชิปใด ๆ ใช้ค่าสัดส่วนคาร์ดินัลลิตี (cardinality ratio) เป็นตัวระบุ ใช้สัญลักษณ์ (n:m) แทนสัดส่วน

คาร์ดินัลลิตี เมื่อค่า n คือ จำนวนที่ปรากฏของเอนติตีแม่ และค่า m คือจำนวนที่ปรากฏของเอนติตีลูก เช่น รีเลชันชิประหว่างใบส่งสินค้ากับรายการแจ้งราคาสินค้ามีสัดส่วนคาร์ดินัลลิตีเป็น (1:2) หมายถึงใบส่งสินค้า 1 ใบประกอบด้วยรายการแจ้งราคาสินค้า 2 รายการ เป็นต้น

ประเภทของรีเลชันชิป แบ่งเป็น 3 ประเภท ได้แก่

2.1 รีเลชันชิประหว่างเอนติตีทั้งสองแบบหนึ่งต่อหนึ่ง (one-to-one relationship) คือ ภายใต้รีเลชันชิปนี้จำนวนที่ปรากฏของเอนติตีแม่และเอนติตีลูกต้องมีค่าเป็น 1 เท่านั้น สัญลักษณ์แสดงแทนด้วยจำนวนหัวลูกศร 1 หัว ใช้ (1:1) ระบุค่าสัดส่วนคาร์ดินัลลิตี

ตัวอย่างรูปที่ 2.1 ผู้จัดการ (MANAGER) เป็นเอนติตีแม่ และแผนก (DEPARTMENT) เป็นเอนติตีลูก มีรีเลชันชิปชื่อ manages หมายถึงผู้จัดการแต่ละคนบริหารงานได้เพียง 1 แผนก และแต่ละแผนกมีผู้จัดการได้เพียง 1 คนเท่านั้น



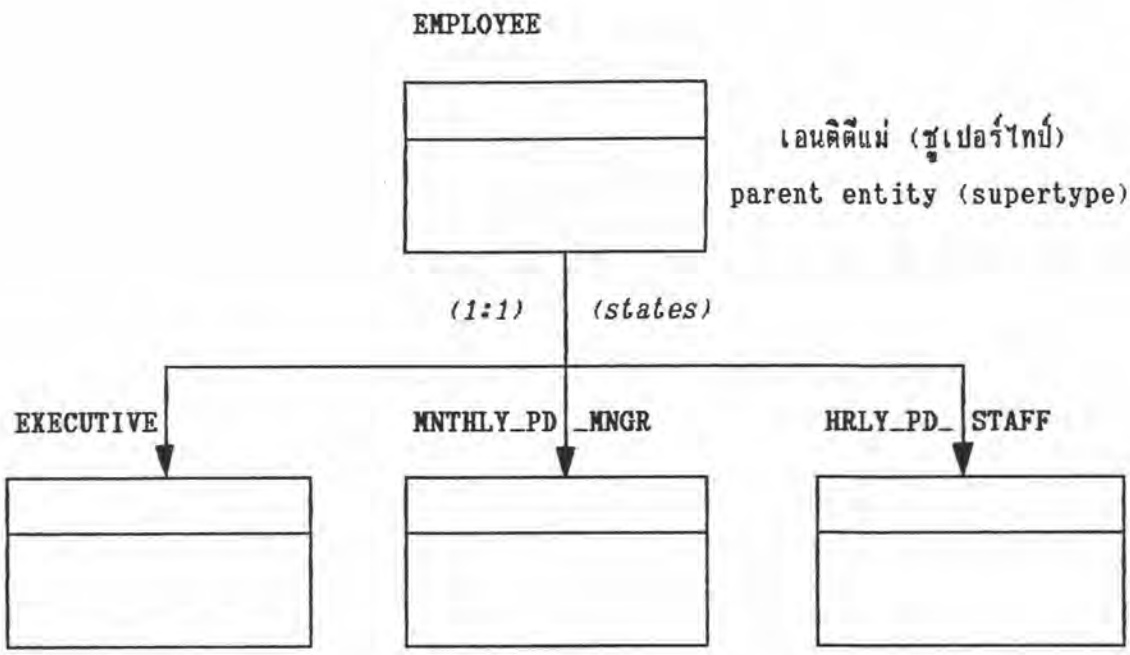
one-to-one relationship

รูปที่ 2.1 แสดงรีเลชันชิประหว่างเอนติตีทั้งสองแบบหนึ่งต่อหนึ่ง

<u>ตัวอย่างพิเศษของรีเลชันชิปประเภทหนึ่งต่อหนึ่ง</u>

เอนติตีทั้งสองอาจมีรีเลชันชิประหว่างซับไทป์กับซูเปอร์ไทป์แบบหนึ่งต่อหนึ่ง (subtype-to-supertype relationship) ซับไทป์ คือเอนติตีลูกที่เป็นเซ็ทย่อยของซูเปอร์ไทป์หรือเอนติตีแม่ อาจเรียกซับไทป์ว่าประเภท (category) มีคุณสมบัติคือเป็นประเภทใดประเภทหนึ่งเท่านั้น และไม่เคยปรากฏในประเภทอื่นพร้อมกัน (mutually exclusive) ใช้ระบุรายละเอียดเพิ่มเติมนอกเหนือจากที่ระบุในซูเปอร์ไทป์

ตัวอย่างรูปที่ 2.2 ลูกจ้าง (EMPLOYEE) เป็นเอนติตีแม่หรือซูเปอร์ไทป์ประกอบด้วยแอตตริบิว (attribute) ที่ระบุสมบัติโดยส่วนรวมของเอนติตีนี้ มีเอนติตีลูกหรือซับไทป์ซึ่งประกอบด้วยแอตตริบิวที่ระบุคุณสมบัติเพิ่มเติมจากซูเปอร์ไทป์เฉพาะประเภทนั้นๆ เป็นเอกเทศจากกัน ในที่นี้คือลูกจ้างแต่ละคนมีสถานะ (states) ตามระดับตำแหน่งได้ 3 ระดับ คือ ผู้จัดการระดับบริหาร (EXECUTIVE), ผู้จัดการควบคุมการผลิต (MNTHLY_PD_MNGR) และ คนงานรายชั่วโมง (HRLY_PD_STAFF)



เอนติตีลูก (ซับไทป์) child entities (subtypes)

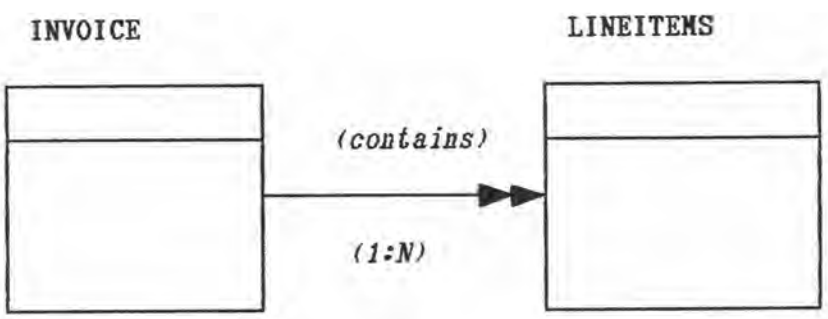
(subtype-to-supertype relationship)

รูปที่ 2.2 แสดงรีเลชันชิประหว่างซับไทป์กับซูเปอร์ไทป์แบบหนึ่งต่อหนึ่ง

การระบุซับไทป์และซูเปอร์ไทป์ด้วยคีย์หลักตัวเดียวกัน เนื่องจากเป็นการระบุถึงสิ่งเดียวกัน คีย์หลักในซับไทป์ (เอนติตีลูก) ยังมีคุณสมบัติเป็นฟอร์เรนจ์คีย์ที่อ้างอิงถึงคีย์หลักของซูเปอร์ไทป์ (เอนติตีแม่)

2.2 รีเลชันชิประหว่างเอนติตีทั้งสองแบบหนึ่งต่อหลาย (one-to-many relationship) คือ ภายใต้รีเลชันชิปนี้จำนวนที่ปรากฏของเอนติตีแม่อาจมีค่าเป็น 0,1 หรือมากกว่าก็ได้ ส่วนจำนวนที่ปรากฏของเอนติตีลูกมีค่าเป็น 1 เท่านั้น สัญญลักษณ์แสดงด้วยหัวลูกศรคู่หันทิศทางเดียวกันและอยู่ต่อท้ายติดกัน ใช้ (1:N) ระบุสัดส่วนคาร์ดินัลลิตี

ตัวอย่างรูปที่ 2.3 ใบส่งสินค้า (INVOICE) เป็นเอนติตีแม่ และรายการแจ้งราคาสินค้า (LINEITEM) เป็นเอนติตีลูก มีรีเลชันชิปชื่อ contains หมายถึง ใบส่งสินค้า 1 ใบประกอบด้วยรายการแจ้งราคาสินค้าหลายรายการ และรายการแจ้งราคาสินค้าแต่ละรายการจะปรากฏในใบส่งสินค้าได้เพียงใบเดียวและครั้งเดียวเท่านั้น



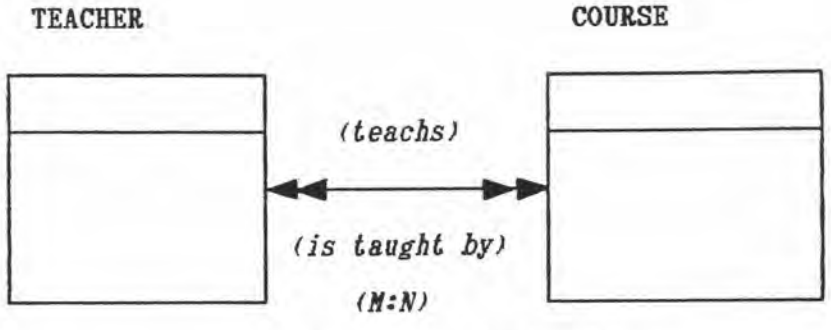
one-to-many relationship

รูปที่ 2.3 แสดงรีเลชันชิประหว่างเอนติตีทั้งสองแบบหนึ่งต่อหลาย

2.3 รีเลชันชิประหว่างเอนติตีทั้งสองแบบหลายต่อหลาย (many-to-many relationship) คือ ภายใต้รีเลชันชิปนี้จำนวนที่ปรากฏของเอนติตีแม่และเอนติตีลูกอาจมีค่าเป็น 0, 1 หรือมากกว่าก็ได้ ใช้ (M:N) ระบุสัดส่วนคาร์ดินัลลิตี

ตัวอย่างรูปที่ 2.4 ถ้าพิจารณาถึงทิศทางหัวลูกศรคู่ เมื่อหันหัวไปทางขวา มีรีเลชันชิปชื่อ teachs อาจารย์ (TEACHER) เป็นเอนติตีแม่ และรายวิชา (COURSE) เป็นเอนติตีลูก หมายถึงอาจารย์แต่ละคนมีการสอนได้หลายรายวิชา

ถ้าพิจารณาถึงทิศทางหัวลูกศรคู่ เมื่อหันหัวไปทางซ้าย มีรีเลชันชิปชื่อ is taught by รายวิชาเป็นเอนติตีแม่ และอาจารย์เป็นเอนติตีลูก หมายถึง แต่ละรายวิชาอาจมีอาจารย์หลายคนสอน

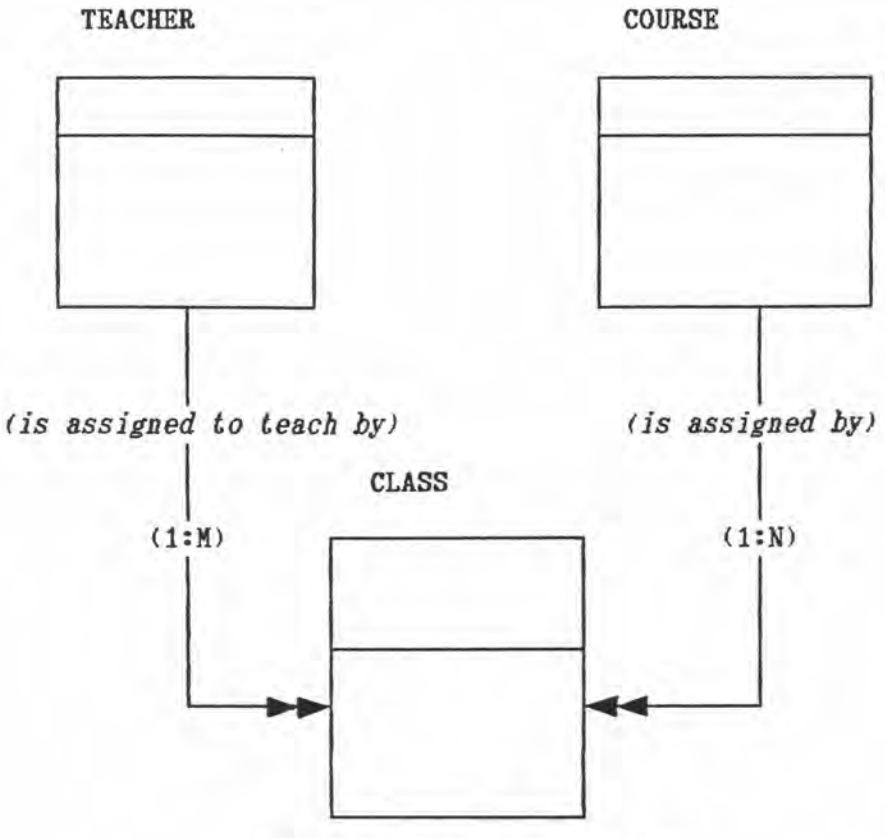


รูปที่ 2.4 แสดงรีเลชันชิประหว่างเอนติตีทั้งสองแบบหลายต่อหลาย

แต่เนื่องจากรีเลชันชิปประเภทนี้ มีความซับซ้อนมากในขั้นตอนการทำโมเดลข้อมูลเชิงตรรก จึงได้มีการแก้ไขโดยแยกเป็นรีเลชันชิปแบบหนึ่งต่อหลายได้ 2 รีเลชันชิป โดยเอนติตีทั้ง 2 นี้มีรีเลชันชิปกับเอนติตีที่ระบุขึ้นใหม่

ตัวอย่างรูปที่ 2.5 มีการระบุเอนติตีใหม่คือชั้นเรียน (CLASS) มีรีเลชันชิปแรกชื่อ is assigned to teach by อาจารย์เป็นเอนติตีแม่ และชั้นเรียนเป็นเอนติตีลูก หมายถึง ภายในชั้นเรียนกำหนดให้แต่ละอาจารย์สอนได้หลายรายวิชา ใช้ (1:M) ระบุสัดส่วนคาร์ดินัลลิตี และรีเลชันชิปหลังชื่อ is assigned by รายวิชาเป็นเอนติตีแม่ และ ชั้นเรียนเป็นเอนติตีลูก หมายถึง ภายในชั้นเรียนกำหนดให้แต่ละรายวิชามีอาจารย์สอนได้หลายคน ใช้ (1:N) ระบุสัดส่วนคาร์ดินัลลิตี

017649



รูปที่ 2.5 แสดงรีเลชันชิประหว่างเอนติตีทั้งสองแบบหนึ่งต่อหลาย 2 รีเลชันชิป

3. แอตตริบิว (attribute) เป็นข้อความที่ปรากฏอยู่ภายในสี่เหลี่ยมแต่ละบรรทัด หมายถึงตัวพรรณาลักษณะต่าง ๆ ของเอนติตี อาจเรียกว่าองค์ประกอบข้อมูล (data element), ชิ้นข้อมูล (data item), หรือ เขตข้อมูล (data field) ก็ได้

ประเภทของแอตตริบิวแบ่งเป็น 3 ประเภท ดังนี้

3.1 แอตตริบิวประเภทคีย์หลัก (primary key attribute) คือแอตตริบิวหรือกลุ่มแอตตริบิวที่ระบุถึงแต่ละรายการในเอนติตีใด ๆ โดยค่าที่ระบุมีผลทำให้รายการนั้นเป็นเอกลักษณ์ไม่ซ้ำกับรายการอื่น ใช้ข้อความภาษาอังกฤษภายในรูปสี่เหลี่ยมเป็นชื่อของคีย์หลัก อยู่เหนือเส้นตรงแนวนอนในช่องบน ส่วนแอตตริบิวอื่นนอกจากนี้จะปรากฏอยู่บริเวณส่วนล่างเส้นแนวนอนลงมา

3.2 แอตตริบิวประเภทคีย์รอง (alternate key attribute) คือ แอตตริบิวหรือกลุ่มแอตตริบิวอื่นที่ใช้ระบุถึงแต่ละรายการในเอนติตี เช่นเดียวกับคีย์หลักแต่ไม่เป็นคีย์หลัก ใช้สัญลักษณ์ AKn ระบุไว้หลังข้อความภายในรูปสี่เหลี่ยม โดยค่าของ n เป็นเลขจำนวนเต็ม มีค่าตั้งแต่ 1,2,3,...,n หมายถึงคีย์รองตัวที่ n

3.3 แอตตริบิวประเภทฟอร์เรนจ์คีย์ (foreign key attribute) คือ แอตตริบิวหรือกลุ่มแอตตริบิวที่อยู่ในเอนติตีลูก และมีค่าสอดคล้องกับค่าของคีย์หลักในเอนติตีแม่ที่มีรีเลชันชิปกันอยู่ และอาจไม่มีค่าได้ (null) ใช้สัญลักษณ์ FKn ระบุไว้หลังข้อความภายในรูปสี่เหลี่ยม โดยค่าของ n เป็นเลขจำนานเต็ม มีค่าตั้งแต่ 1,2,3,...,n หมายถึงฟอร์เรนจ์คีย์ตัวที่ n

แต่ในกรณีที่ฟอร์เรนจ์คีย์เป็นส่วนหนึ่งของคีย์หลักในเอนติตีลูก ควรหลีกเหลี่ยงการไม่มีค่าในฟอร์เรนจ์คีย์ เพราะจะทำลายคุณสมบัติความเป็นเอกลักษณ์ของคีย์หลัก

3.4 แอตตริบิวประเภทที่ไม่ใช่คีย์ (non key attribute) คือแอตตริบิวที่ไม่มีคุณสมบัติตาม 3 ข้อที่กล่าวมาแล้วข้างต้น

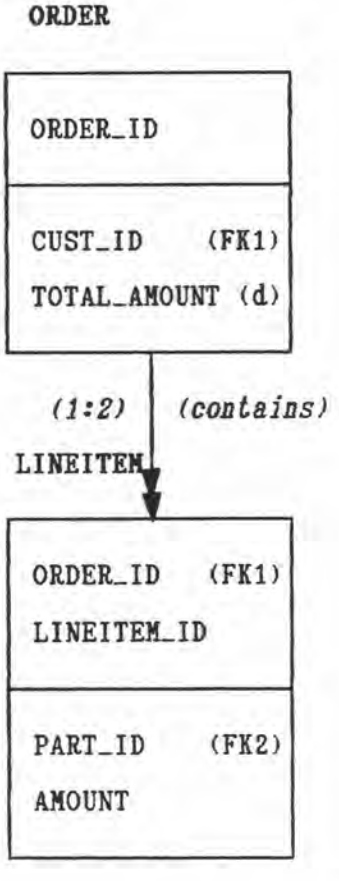
นอกจากนี้ยังมีประเภทแอตตริบิวพิเศษอื่น ได้แก่

- แอตตริบิวประเภทดิไรฟ์ (derived attributes) ควรระบุอัลกอริทึมของแอตตริบิวนี้ โดยอ้างถึงแอตตริบิวอื่นที่เป็นแหล่งกำเนิดค่าในพจนานุกรมข้อมูลในระดับตรรก และใช้สัญลักษณ์ d ระบุหลังข้อความภาษาอังกฤษซึ่งเป็นชื่อของแอตตริบิวในโมเดลข้อมูลเชิงตรรกส่วนในระดับกายภาพไม่มีการสร้างคอลัมน์นี้ เพื่อเป็นการประหยัดเนื้อที่ในการเก็บข้อมูล และลดภาระการแก้ไขเมื่อมีการเปลี่ยนแปลงแอตตริบิวต้นกำเนิดที่แอตตริบิวนี้อ้างถึงในการคำนวณ ทำให้ไม่เกิดความขัดแย้งของข้อมูล

ตัวอย่างรูปที่ 2.6 มีแอตตริบิวประเภทดิไรฟ์ชื่อ TOTAL_AMOUNT เป็นยอดจำนวนเงินของใบสั่งสินค้าใช้ขยายเอนติตีแม่คือ ใบสั่งสินค้า (ORDER) มีรีเลชันชิปชื่อ contains กับเอนติตีลูกคือ รายการสั่งสินค้า (LINEITEM) แบบหนึ่งต่อหลาย มีค่าสัดส่วนคาร์ดินัลลิตีเท่ากับ (1:2) ใช้สัญลักษณ์ d ระบุหลังชื่อแอตตริบิวประเภทนี้ในแผนภาพโมเดลข้อมูลเชิงตรรก ค่าของ

TOTAL_AMOUNT ได้มาจากผลรวมของ AMOUNT จากเอนติตีรายการสั่งสินค้า เมื่อค่ารหัสใบสั่งสินค้าในเอนติตีใบสั่งสินค้าและเอนติตีรายการสั่งสินค้ามีค่าตรงกัน อาจเขียนเป็นค่าสั่ง SQL ได้ดังนี้

```
SQL>  SELECT SUM(LINE_ITEM.AMOUNT) TOTAL_AMOUNT
      FROM   LINEITEM,ORDER
      WHERE  ORDER.ORDER_ID = LINEITEM.ORDER_ID;
```



รูปที่ 2.6 แสดงตัวอย่างของแอตตริบิวประเภทดิไรฟ์ชื่อ TOTAL_AMOUNT

การพัฒนาโมเดลข้อมูลเชิงตรรกที่สมบูรณ์ไม่เพียงแต่จะปรากฏอยู่ในแผนภาพเท่านั้น ยังต้องระบุไว้ในระบบพจนานุกรมข้อมูลด้วย โดยจะต้องมีคุณสมบัติคือมีความเป็นบูรณภาพ, ไม่มีความขัดแย้งกัน, ลดความเหลื่อเฟือให้น้อยที่สุด อาศัยหลักการดังนี้

1. กฎคีย์ธุรกิจ (key business rule) ควบคุมความสัมพันธ์ระหว่างคีย์หลักและฟอร์เรนจ์คีย์ที่อ้างอิงถึงกัน โดยระบุชนิดเหตุการณ์ การลบ/แก้ไขค่าคีย์หลัก หรือ การเพิ่ม/แก้ไขฟอร์เรนจ์คีย์ ภายใต้การตรวจสอบเงื่อนไขการมีอยู่จริงของค่าคีย์หลักหรือค่าฟอร์เรนจ์คีย์ที่เกี่ยวข้อง ถ้าพบว่าเงื่อนไขไม่ตรงตามที่กำหนดจะไม่ยอมรับการดำเนินการนั้น

2. กฎโดเมนธุรกิจ (domain business rule) ควบคุมชนิดและช่วงของค่าที่ยอมรับได้ของแอตตริบิวให้ถูกต้อง

3. ทริกเกอร์ดำเนินการ (trigger operation) ควบคุมผลกระทบเมื่อมีการเพิ่ม, เปลี่ยนแปลงหรือยกเลิกเอนติตีหรือแอตตริบิวใด ๆ

การสร้างโครงวิวของผู้ใช้ เป็นการสำรวจดูกิจกรรมหรือหน้าที่ของธุรกิจ ณ.เวลาใดๆ โดยสร้างแผนภาพจากความต้องการสารสนเทศของแต่ละกิจกรรมคือวิวของผู้ใช้ และนำมาเชื่อมต่อเข้าด้วยกันได้แผนภาพรวม นอกจากนี้ต้องมีการระบุขอบเขตวิวของผู้ใช้ เนื่องจากอาจมีผลกระทบต่อกิจกรรมโดยส่วนรวมในระดับบนได้ การรวบรวมแผนภาพโมเดลข้อมูลเชิงตรรกที่สมบูรณ์ต้องคำนึงถึงจำนวนและขอบเขตวิวของผู้ใช้ทั้งหมดอย่างละเอียดก่อน โดยยึดขั้นตอนของ LDM$_n$ ซึ่งย่อมาจาก Logical Data Modeling และ n หมายถึงขั้นตอนที่ ประกอบด้วย 12 ขั้นตอนดังนี้

ขั้นตอน LDM 1 กำหนดเอนติตีหลักที่หน่วยงานต่างๆ จำเป็นต้องอ้างถึงก่อน

ขั้นตอน LDM 2 พิจารณารีเลชันชิประหว่างเอนติตีหลักเหล่านี้ ทำการตั้งชื่อเอนติตีไว้ในระบบพจนานุกรมข้อมูล นำเอนติตีและรีเลชันชิปบรรจุลงแผนภาพโมเดลข้อมูล แทนเอนติตีด้วยสัญลักษณ์สี่เหลี่ยม ระบุชื่อเอนติตีเหนือสี่เหลี่ยมด้วยตัวอักษรภาษาอังกฤษ และแทนรีเลชันชิปด้วยกริยาวลีภาษาอังกฤษ เพื่อทำให้เกิดความเข้าใจถึงความต้องการสารสนเทศเบื้องต้นก่อน

ขั้นตอน LDM 3 พิจารณาแอตตริบิวประเภทคีย์หลัก, คีย์รอง เพื่อระบุคุณสมบัติของเอนติตี

ขั้นตอน LDM 4 พิจารณาแอตตริบิวประเภทฟอร์เรนจ์คีย์ เพื่อระบุคุณสมบัติของรีเลชันชิป

ขั้นตอน LDM 5 พิจารณากฎคีย์ธุรกิจคือข้อเท็จจริงเกี่ยวกับความหมายของข้อมูล ใช้ควบคุมผลกระทบที่เกิดจากการเพิ่ม, ลบ, หรือแก้ไขค่าของคีย์หลักและฟอร์เรนจ์คีย์ เป็นการกำหนดข้อจำกัดบังคับการมีอยู่ให้ถูกต้องและสอดคล้องกัน เรียกว่า existence constraints ผู้ใช้ต้องระบุกฎการเพิ่มและลบให้สอดคล้องกับวิวทางธุรกิจ ดังนี้

- กฎการเพิ่ม (Insert Rule) ควบคุมเงื่อนไขการเพิ่มรายการ หรือ แก้ไขค่าฟอร์เรนจ์คีย์ในเอนติตีลูก ต้องคำนึงถึงค่าฟอร์เรนจ์คีย์ที่สอดคล้องกับค่าคีย์หลักในเอนติตีแม่

  ตัวอย่างกฎการเพิ่มในกรณีรีเลชันชิปแบบซูเปอร์ไทป์-ซับไทป์ เป็นไปได้ 2 แบบ

  1. ยอมให้เพิ่มรายการในเอนติตีลูกได้ ถ้าค่าคีย์หลักไม่ตรงกับค่าคีย์หลักในเอนติตีแม่ ต้องทำการเพิ่มรายการในเอนติตีแม่ด้วยค่าคีย์หลักเดียวกันนี้ก่อนเสมอแบบ Automate Insert

  2. ยอมให้เพิ่มรายการในเอนติตีลูกได้ ก็ต่อเมื่อมีค่าคีย์หลักตรงกับค่าคีย์หลักของเอนติตีแม่เท่านั้น แบบ Dependent Insert

- กฎการลบ (Delete Rule) ควบคุมเงื่อนไขการลบรายการ หรือ แก้ไขค่าคีย์หลักในเอนติตีแม่ ต้องคำนึงถึงผลกระทบต่อค่าฟอร์เรนจ์คีย์ที่เอนติตีลูกอ้างถึง

  ตัวอย่างกฎการลบในกรณีรีเลชันชิปแบบซูเปอร์ไทป์-ซับไทป์ เป็นไปได้ 2 แบบ

  1. ยอมให้ลบรายการในเอนติตีแม่ ก็ต่อเมื่อมีการลบทุกรายการในเอนติตีลูกที่มีค่าคีย์หลักตรงกับค่าคีย์หลักของเอนติตีแม่ด้วยแบบ Cascade Delete

  2. ยอมให้ลบรายการในเอนติตีแม่เมื่อไม่มีการอ้างอิงจากเอนติตีลูกแบบ Restrict Delete

ขั้นตอน LDM 6 เพิ่มแอตตริบิวอื่นที่ไม่ใช่คีย์ เพื่อเป็นรายละเอียดของเอนติตี

ขั้นตอน LDM 7 ตรวจสอบวิวของผู้ใช้โดยอาศัยหลักการนอร์มัลไลเซชัน ควรเริ่มทำขั้นตอนนี้เมื่อผ่านขั้นตอน LDM 1 ถึง LDM 6 มาก่อน

นอร์มัลไลเซชัน เป็นทฤษฎีในการวิเคราะห์และแยกโครงสร้างของข้อมูลเป็นส่วนๆ ให้ได้กลุ่มของโครงสร้างข้อมูลใหม่ที่มีเสถียรภาพ และได้โมเดลข้อมูลเชิงตรรกที่เหมาะสม ปราศจากความซ้ำซ้อนของข้อมูล แต่อย่างไรก็ตามการใช้หลักการนี้เมื่อถึงขั้นตอนการออกแบบฐานข้อมูลระดับกายภาพอาจมีบางส่วนไม่เหมาะสม เนื่องจากไม่สามารถตอบสนองความต้องการในแง่ประสิทธิภาพในการทำงาน, ความรวดเร็วในการประมวลผล ฯลฯ ทั้งนี้จะต้องมีการดัดแปลงคืนรูปการนอร์มัลไลเซชันตามความเหมาะสมก็ได้ นอร์มัลไลเซชันมีขั้นตอนพอสังเขปดังนี้

ระดับที่ 1 (First normal form : 1NF) เป็นการขจัดแอตตริบิวหรือกลุ่มแอตตริบิวที่ซ้ำกันไปอยู่ในเอนติตีลูก เพื่อแต่ละรายการในเอนติตีไม่มีค่าของแอตตริบิวหรือค่าของกลุ่มแอตตริบิวที่ซ้ำกัน

ระดับที่ 2 (Second normal form : 2NF) เป็นการขจัดแอตตริบิวที่ไม่ขึ้นกับทั้งส่วนของคีย์หลักออกไป เพื่อให้แอตตริบิวอื่นทั้งหมดขึ้นตรงกับส่วนที่เป็นคีย์หลักทั้งหมดเท่านั้น

ระดับที่ 3 (Third normal form : 3NF) เป็นการขจัดแอตตริบิวที่ไม่เป็นคีย์ที่ขึ้นตรงกับแอตตริบิวอื่นที่ไม่ใช่คีย์หลักออกไป เพื่อให้แอตตริบิวที่ไม่ใช่คีย์หลักต้องขึ้นตรงกับทั้งส่วนที่เป็นคีย์หลักและไม่ขึ้นกับแอตตริบิวอื่นที่ไม่ใช่คีย์หลัก

ระดับของ (Boyce/Codd normal form : BCNF)

ระดับที่ 4 (Fourth normal form : 4NF)

ระดับที่ 5 (Fifth normal form : 5NF)

<u>หมายเหตุ</u> ในการพัฒนาโมเดลข้อมูลเชิงตรรก สามารถเข้าสู่ขั้นตอนนอร์มัลไลเซชันในระดับที่ 3 ได้อยู่แล้ว ส่วนในระดับถัดไปต้องพิจารณาถึงลักษณะของข้อมูลว่าจำเป็นต้องดำเนินการเพิ่มเติมหรือไม่ ซึ่งมีหลักการซับซ้อนขึ้น

ขั้นตอน LDM 8 พิจารณาขอบเขตค่าโดเมนของแอตตริบิวที่ถูกต้อง เช่น ชนิดของข้อมูล (data type), ความยาว (length), รูปแบบ (format), ความเป็นเอกลักษณ์ (uniqueness), ต้องมีค่าหรือไม่ (null support), ค่าโดยปริยาย (default values), และคำอธิบาย เป็นต้น

ขั้นตอน LDM 9 พิจารณากฎทริกเกอร์ดำเนินการหรือกฎธุรกิจของแอตตริบิว (attribute business rule) ควบคุมผลกระทบที่เกิดจากการสืบค้น, เพิ่ม, แก้ไข และ ลบ เอนติตีและแอตตริบิวที่มีต่อเอนติตีอื่นหรือแอตตริบิวอื่นภายใต้เอนติตีเดียวกัน เพื่อควบคุมบูรณภาพของข้อมูลและความไม่ขัดแย้งของค่าแอตตริบิว

การบันทึกทริกเกอร์ดำเนินการลงในพจนานุกรมข้อมูล ประกอบไปด้วยส่วนต่างๆ ดังนี้

- กฎของผู้ใช้ (user rule) เป็นข้อความอธิบายข้อกำหนดการดำเนินงานตามความต้องการของผู้ใช้
- เหตุการณ์ของทริกเกอร์ดำเนินการ (event) ได้แก่ เพิ่ม (INSERT), แก้ไข (UPDATE), ลบ (DELETE) หรือ สืบค้น (SELECT)
- ส่วนประกอบของเหตุการณ์ (object of event) ได้แก่ ชื่อของเอนติตี หรือแอตตริบิวที่มีการเปลี่ยนแปลงหรือถูกสืบค้นในเหตุการณ์นั้นๆ
- เงื่อนไขภายใต้ทริกเกอร์ดำเนินการ (condition)
- การกระทำ (action) เช่น ไม่ยอมรับเหตุการณ์หรือทริกเกอร์ที่เกี่ยวกับเหตุการณ์นั้น

การระบุทริกเกอร์ดำเนินการควรมีการบันทึกตามกรณีที่เกิดขึ้น ดังนี้คือ

- สำหรับทุกแอตตริบิวที่เป็นแหล่งกำเนิดของแอตตริบิวประเภทดิไรฟ์ เช่น เมื่อมีการแก้ไขค่าของแอตตริบิวต้นกำเนิด ต้องมีทริกเกอร์ดำเนินการในการแก้ไขแอตตริบิวดิไรฟ์ด้วย
- ควรระบุกฎการเพิ่มรายการในเอนติตีลูกหรือซับไทป์ ในลักษณะ Automatic Insert หรือ Dependent Insert
- ควรระบุกฎการลบรายการในเอนติตีแม่หรือซูเปอร์ไทป์ ในลักษณะ Cascade Delete

- เมื่อมีการรักษาความปลอดภัยของข้อมูลในลักษณะการสืบค้น ทั้งในระดับเอนติตี และแอตตริบิว เช่น กรณีทำการบันทึกเหตุการณ์ว่ามีใครทำการสืบค้นข้อมูลส่วนนี้บ้าง เมื่อเวลาเท่าไร

ขั้นตอน LDM 10 รวบรวมวิวของผู้ใช้เข้าด้วยกันเป็นแบบเบ็ดเสร็จ (integrate user views) พิจารณาส่วนของวิวของผู้ใช้ที่คาบเกี่ยวกัน พยายามลดความเหลื่อมล้ำ และแก้ปัญหาความซ้ำซ้อนที่เกิดข้ามวิวของผู้ใช้ มีขั้นตอนย่อย 3 ขั้นตอน คือ

1. รวบรวมเอนติตีและกฎธุรกิจที่เกี่ยวข้องเข้าด้วยกัน พิจารณาคีย์หลักและคีย์รอง
2. รวบรวมรีเลชันชิปและกฎธุรกิจที่เกี่ยวข้องเข้าด้วยกัน พิจารณากฎการเพิ่ม/ลบ
3. รวบรวมแอตตริบิวและกฎธุรกิจที่เกี่ยวข้องเข้าด้วยกัน พิจารณาโดเมน และทริกเกอร์ดำเนินการ

ขั้นตอน LDM 11 รวบรวมโมเดลข้อมูลที่มีอยู่เข้าด้วยกัน ระบุการเชื่อมโยงระหว่างกัน การปฏิบัติงานภายใต้โมเดลข้อมูลเฉพาะบริเวณ การมีแบบแผนฐานข้อมูลเชิงมโนภาพทางธุรกิจ ทำให้ได้โมเดลข้อมูลหลายภาพ

ขั้นตอน LDM 12 พิจารณาความเปลี่ยนแปลงที่อาจมีในอนาคต ว่ามีผลกระทบต่อโมเดลของข้อมูลในปัจจุบันหรือไม่

2.3 <u>การออกแบบฐานข้อมูลแบบรีเลชันนัล</u> (Relational Database Design: RDD)

เป็นขั้นตอนการแปลงจากโมเดลข้อมูลเชิงตรรกเข้าสู่ระบบฐานข้อมูลแบบรีเลชันนัล (Translation Process) อย่างมีเสถียรภาพ ภายใต้ระบบการจัดการฐานข้อมูลที่ติดตั้งบนเครื่องคอมพิวเตอร์ เป็นการปรับให้เข้ากับหน้าที่ใช้สอยเฉพาะและตามความต้องการในเรื่องประสิทธิภาพการทำงาน ประกอบด้วย 2 ส่วน คือ

2.3.1 โครงสร้างข้อมูล (data structure) บอกที่อยู่ของโครงสร้างข้อมูล ได้แก่ เอนติตี, แอตตริบิว ด้วยการระบุตารางรีเลชันนัลและคอลัมน์ดังนี้

- ตารางรีเลชันนัลเทียบเท่ากับเอนติตีและคอลัมน์เทียบเท่ากับแอตตริบิวในโมเดล-ข้อมูลเชิงตรรก โดยต้องคำนึงว่าแต่ละตารางและคอลัมน์ที่พบในระบบการจัดการฐานข้อมูลแบบรีเลชันนัลต้องไม่มีความขัดแย้งกัน

คอลัมน์ คือ หน่วยหนึ่งของข้อมูลที่สามารถกำหนดตำแหน่งที่อยู่ได้ อาจเรียกอีกอย่างว่าเขตข้อมูล

ตาราง คือ กลุ่มของคอลัมน์ โดยคอลัมน์ทุกตัวในตารางเป็นตัวพรรณาถึงลักษณะสมบัติของตารางนั้นๆ

- บางกรณีอาจมีการรวมตารางเข้าเป็นตารางเดียว หรือ แบ่งตารางออกเป็นหลายตาราง เพื่อประโยชน์ในแง่การทำงานรวดเร็วขึ้น

2.3.2 บูรณภาพข้อมูล (data integrity) จากแผนภาพโมเดลข้อมูลเชิงตรรกสามารถควบคุมบูรณภาพข้อมูลด้วยกฎธุรกิจ ซึ่งได้แก่พารามิเตอร์ระบุตารางรีเลชันนัล เช่น การสร้างเพื่อระบุความเป็นเอกลักษณ์ (unique index), คำสั่งมาโคร, โปรแกรม ฯลฯ

บูรณภาพของข้อมูล (data integrity) เป็นการรักษาคุณภาพ ความน่าเชื่อถือของข้อมูลในฐานข้อมูลของระบบสารสนเทศ และป้องกันมิให้ข้อมูลเกิดความผิดพลาด ถ้าพิจารณาจากระบบจัดการฐานข้อมูลพบว่าปัจจุบันไม่ได้เอื้ออำนวยในการควบคุมทางด้านนี้โดยตรง กล่าวคือ ภาษาเอสคิวแอล (SQL Language) ซึ่งเป็นภาษาในยุคที่ 4 (4th Generation Language, 4GL) เป็นคำสั่งแบบไม่ใช่วิธีดำเนินการ (nonprocedural syntax) ใช้ในการสร้าง, เก็บ, เปลี่ยนแปลง, สืบค้น และดูแลรักษาสารสนเทศในฐานข้อมูล ไม่อาจระบุคำสั่งเพื่อให้เกิดบูรณภาพของข้อมูลได้โดยตรง จึงจำเป็นต้องกำหนดกฎเพื่อคงความเป็นบูรณภาพของข้อมูล แบ่งเป็น 3 ประเภทดังนี้

2.3.2.1 กฎบูรณภาพของเอนติตี (Entity integrity rule) กำหนดให้ส่วนประกอบของคีย์หลักเป็นตัวระบุความเป็นเอกลักษณ์ภายในแต่ละทูเพิล (tuple) ของตารางรีเลชันใดๆ จะต้องมีค่าเสมอ ในขั้นตอนการเพิ่ม, ปรับปรุง หรือลบ จำเป็นต้องคงไว้ซึ่งคุณสมบัติของคีย์หลัก (หมายเหตุ ทูเพิล คือ แต่ละรายการในตาราง)

ในคำสั่ง CREATE TABLE ต้องระบุ NOT NULL ท้ายคอลัมน์ (column) หรือกลุ่มของคอลัมน์ที่ต้องการให้เป็นคีย์หลัก

และในคำสั่ง CREATE UNIQUE INDEX เพื่อให้ระบบจัดการฐานข้อมูลตรวจสอบความเป็นเอกลักษณ์ของแต่ละทูเพิลตามค่าในคีย์หลัก

2.3.2.2 กฎบูรณภาพในการอ้างอิง (Referential integrity rule) เป็นการหาแหล่งที่มาของฟอร์เรนจ์คีย์ ถ้าตารางรีเลชันใดมีฟอร์เรนจ์คีย์ปรากฏอยู่ ทุกค่าของฟอร์เรนจ์คีย์อาจเป็นไปได้ 2 กรณี คือ

- ไม่มีค่า (null value)
- มีค่าที่สอดคล้องกับค่าคีย์หลักในตารางรีเลชันอื่น

ดังนั้นต้องมีการระบุถึงการอ้างอิงระหว่างตารางรีเลชันด้วยกัน เพื่อตรวจสอบค่าของฟอร์เรนจ์คีย์ว่าตรงกับค่าที่มีอยู่จริงในคีย์หลักหรือไม่ ในการดำเนินการเพิ่ม, ปรับปรุง หรือลบ อาจกล่าวอีกนัยหนึ่งว่าเป็นการควบคุมบูรณภาพของฟอร์เรนจ์คีย์ก็ได้

2.3.2.2 กฎบูรณภาพของโดเมน (Domain integrity rule) เป็นการควบคุมความถูกต้องสำหรับทุกๆ คอลัมน์ที่อยู่ในตารางรีเลชัน ได้แก่

- ชนิดของข้อมูล (data type) เช่น เป็นตัวอักษร, วันที่, ตัวเลขแบบจำนวนเต็ม หรือ ทศนิยม เป็นต้น
- ความยาวของข้อมูล (length)
- ช่วงของค่าข้อมูล (range of data)
- ค่าโดยปริยาย (default value)
- ความเป็นเอกลักษณ์ (uniqueness)
- ต้องมีค่าหรือไม่ (nullability) เป็นต้น

ขั้นตอนการออกแบบฐานข้อมูลแบบรีเลชันนัล มีดังนี้

RDD1 ระบุตารางรีเลชันนัลจากเอนติตี

RDD2 ระบุคอลัมน์ภายในตารางรีเลชันนัลสำหรับแต่ละแอตตริบิวที่ขยายคุณสมบัติของเอนติตี

RDD3 ดัดแปลงโครงสร้างให้เหมาะสมกับสภาพแวดล้อมของระบบจัดการฐานข้อมูลแบบรีเลชันนัลและปรับในเรื่องประสิทธิภาพการทำงานด้วย

RDD4 ออกแบบกฎธุรกิจที่เกี่ยวข้องกับเอนติตี

- บังคับสมบัติทางตรรกของคีย์หลักในการติดตั้งระบบแบบรีเลชันนัล เช่น ความเป็นเอกลักษณ์ ไม่อนุญาตให้ค่าในคีย์หลักไม่มีค่า เป็นต้น

- บังคับสมบัติทางตรรกของคีย์รองในการติดตั้งระบบแบบรีเลชันนัล เช่น ความเป็นเอกลักษณ์ เป็นต้น

RDD5 ออกแบบกฎธุรกิจที่เกี่ยวข้องกับรีเลชันชิป ข้อจำกัดในการเพิ่ม,ลบ และแก้ไขแอตตริบิวที่เป็นคีย์ในลักษณะการอ้างอิงให้สอดคล้อง

RDD6 ออกแบบกฎธุรกิจเพิ่มเติมเกี่ยวกับแอตตริบิว ได้แก่ ทริกเกอร์ดำเนินการ, การควบคุมค่าโดเมน, ชนิดข้อมูล, รูปแบบข้อมูล, ค่าที่ยอมรับ (ขอบเขต, ความต่อเนื่องของข้อมูล), ความเป็นเอกลักษณ์, อนุญาตให้มีค่าได้หรือไม่ และค่าปริยาย (Fleming,1989)

การปรับปรุงโมเดลข้อมูลใหม่เมื่อมีสารสนเทศใหม่เพิ่มเติม อาจทำให้เกิดการแบ่งแยก (splitting) หรือการยุบรวม (merging) ของเอนติตีขึ้น โดยพิจารณาว่าแอตตริบิวเดิมควรจะถูกย้ายจากเอนติตีเดิมไปสู่เอนติตีใหม่หรือไม่ จะต้องคำนึงถึงสภาพความเป็นจริงเช่น

- แอตตริบิวมีความสัมพันธ์กับแอตตริบิวอื่นอย่างไร
- การสร้างเอนติตีใหม่สามารถแก้ปัญหาการสืบค้นข้อมูลได้หรือไม่
- ข้อเท็จจริงของแอตตริบิวอ้างอิงถึงแอตตริบิวอื่นหรือไม่
- เอนติตีใหม่มีความสัมพันธ์กับเอนติตีเดิมและกับเอนติตีอื่นๆ ในโมเดลอย่างไร

การจัดกลุ่มในระดับเอนติตี (Entity Clustering)

ตั้งอยู่บนพื้นฐานของขอบเขตข้อมูล หมายความว่าข้อมูลทั้งหมดที่จำเป็นต้องการใช้ภายในหน่วยงาน โดยแต่ละหน่วยงานมีหน้าที่รับผิดชอบในการจัดกลุ่มเอนติตี เนื่องจากเป็นเจ้าของข้อมูล โครงสร้างข้อมูลภายในองค์กรที่ปรากฏอยู่จะเป็นตัวบ่งชี้ว่าเอนติตีนี้ควรจะปรากฏในหน่วยงานใดบ้าง การจัดกลุ่มอาจเป็นกลุ่มย่อยของกลุ่มใหญ่รวมกันหลายๆ กลุ่มก็ได้ ขึ้นกับระดับความละเอียดที่ต้องการแสดง เช่น ในระดับบนโมเดลข้อมูลที่ได้จากการจัดกลุ่มมักแสดงเฉพาะเอนติตีหลักที่มีส่วนเชื่อมโยงกับเอนติตีอื่น เพื่อแสดงความสัมพันธ์ของข้อมูลทุกหน่วยงานภายในองค์กรโดยสังเขป ส่วนในระดับล่างลงมาอาจแสดงกลุ่มเอนติตีที่สนใจเฉพาะหน่วยงานเท่านั้น ในกรณีที่หน่วยงานใดมีข้อมูลที่ซับซ้อนหรือมากมายก็อาจจัดกลุ่มย่อยในระดับล่างลงมาอีกได้

การจัดกลุ่มเอนติตีอาจเกิดได้ในระหว่างการออกแบบฐานข้อมูลทั้ง 2 ขั้นตอน ได้แก่ ขั้นตอนเชิงตรรก และ ขั้นตอนเชิงกายภาพ

เนื่องจากโมเดลข้อมูลประกอบด้วยเอนติตีมากมาย การจัดกลุ่มของเอนติตีในขั้นตอนเชิงตรรก ในแต่ละเอนติตีมีการจัดกลุ่มของแอตตริบิวเพื่อขยายคุณสมบัติต่างๆ ขึ้นกับธรรมชาติของข้อมูลที่ปรากฏอยู่และโครงสร้างข้อมูล แต่ขั้นตอนเชิงกายภาพไม่เป็นเช่นนั้น เป็นการจัดกลุ่มเอนติตีเพื่อตอบสนองวิวของผู้ใช้ โดยต้องคำนึงถึงการดำเนินการและความต้องการในการเข้าถึงฐานข้อมูล

การจัดกลุ่มในระดับเอนติตีต้องพิจารณาถึง

- ขอบเขตของข้อมูลที่เกี่ยวข้อง
- โครงสร้างข้อมูลที่ปรากฏอยู่
- เป็นการแสดงแทนวิวของผู้ใช้เฉพาะบริเวณ โดยปรากฏเฉพาะกลุ่มของเอนติตีที่สนใจเท่านั้น

- การสอบถามข้อมูลว่าครอบคลุมถึงเอนติตีตัวใดบ้าง
- การประมวลผลที่ต้องการจากผู้ใช้ (Brathwaite,1985)

<u>ความปลอดภัยของข้อมูล</u>

- เป็นการป้องกันการเข้าถึงข้อมูล, การเพิ่ม, การแก้ไข, การยกเลิกข้อมูลที่เก็บอยู่ในระบบคอมพิวเตอร์
- ป้องกันการบริการข้อมูลที่ผิดวัตถุประสงค์
- ป้องกันระบบจากอันตรายทางกายภาพ

อาจแบ่งเป็น 2 อย่าง คือ

1. ทางกายภาพ เป็นการป้องกันการเข้าถึงเครื่องคอมพิวเตอร์, ระบบอุปกรณ์รอบนอก, และสิ่งแวดล้อม เช่น กำหนดกฎให้บุคคลผู้มีหน้าที่เกี่ยวข้องเท่านั้นที่ได้รับอนุญาตให้เข้ามาในบริเวณที่ติดตั้งระบบคอมพิวเตอร์ เป็นต้น

2. ความปลอดภัยของข้อมูล เป็นการป้องกันการเข้าถึงข้อมูล โดยกำหนดรหัสผ่านแก่ผู้มีสิทธิ์ และกำหนดให้ผู้ใช้สามารถเข้าถึงข้อมูลได้เฉพาะส่วนที่ตนมีสิทธิ์เท่านั้น อาศัยหลักการควบคุมการเข้าถึง (access control) นอกจากนี้ในกรณีมีผู้ใช้หลายคน (multi-user) ทำการสืบค้นสารสนเทศหรือแก้ไขข้อมูลพร้อมกันต้องมีระบบควบคุมแบบการประมวลผลจวบกัน (concurrent processing)

การระบุและควบคุมความปลอดภัยและความเป็นส่วนตัวของข้อมูล (security/privacy) สำหรับฐานข้อมูลแบบเมทาขึ้นเป็นระบบย่อยเพื่ออำนวยความสะดวกสำหรับด้านนี้ โดยมีการอ้างถึงประเภทความปลอดภัยและความเป็นส่วนตัวเช่นในระดับของ เอนติตี รีเลชันชิป และแอตตริบิว เป็นต้น

ในระดับเอนติตีหรือแอตตริบิว มีการป้องกันโดยระบุว่าผู้ใดสามารถเข้าถึงเอนติตีหรือแอตตริบิวใดบ้าง และสามารถใช้คำสั่งในการจัดการข้อมูลอะไรได้บ้าง เช่น เพิ่ม,แก้ไข,ลบ, หรือ สืบค้น เป็นต้น โดยจำกัดสิทธิ์อาศัยระบบการจัดการฐานข้อมูลที่เอื้ออำนวยให้

ปัจจัยในการตัดสินใจให้มีการเข้าถึงข้อมูล (access decisions) ได้แก่

- ความไหวตัวต่อเหตุการณ์ (even-sensitive) คือ ป้องกันการเข้าถึงข้อมูลจากผู้ใช้ในช่วงเวลาใดบ้าง เช่น อนุญาตให้ผู้ใช้สามารถใช้ได้ในช่วงเวลาทำงาน คือ 7:00 น.ถึง 15:00 น. ถ้านอกเหนือเวลาเหล่านี้ระบบจะไม่อนุญาตให้ผู้ใช้เข้ามาปฏิบัติการได้

- ความไหวตัวต่อค่าของข้อมูล (value-sensitive) คือ ขึ้นกับค่าของข้อมูลขณะนั้น เช่น การป้องกันไม่ให้ผู้ใช้อ่านประวัติพนักงานที่มีเงินเดือนมากกว่า 20,000 บาท เป็นต้น

- ความไหวตัวต่อสภาพแบบพลวัต (dynamic-state-sensitive) ของระบบการจัดการฐานข้อมูล

ในทางปฏิบัติอาจใช้วิธี

- สร้างวิว (VIEW) จากระบบการจัดการฐานข้อมูล ให้แตกต่างกันตามวัตถุประสงค์เพื่อป้องกันมิให้ผู้ใช้เข้าถึงข้อมูลตามเงื่อนไขใดบ้าง

- การดัดแปลงการสอบถาม (query modification) ระบุความปลอดภัยให้ผู้ใช้เข้าถึงสารสนเทศตามเงื่อนไข

- การสร้างวิถีการเข้าถึง (access path) โดยอนุญาต (GRANT) ให้มีการเข้าถึงข้อมูลตามที่ผู้ใช้แต่ละคนมีสิทธิ์เท่านั้น ซึ่งอาจกำหนดเป็นกลุ่มผู้ใช้ โดยต้องทราบถึงขอบเขตของผู้ใช้ในแต่ละกลุ่มอย่างชัดเจนเสียก่อน