บทที่ 3

แนวคิด และทฤษฎี

ในองค์กรที่มีการจัดการกับข้อมูลในรูปแบบแฟ้มข้อมูลแบบคอนเวนชันนัลนั้นจะมีโปรแกรมประยุกต์แต่ละโปรแกรมจัดการกับแฟ้มข้อมูลของตัวเอง ดังนั้นข้อมูลในองค์กรเหล่านี้ก็ไม่จำเป็นที่จะต้องใช้ร่วมกัน แต่บางครั้งโปรแกรมที่ต่างกันก็อาจต้องการใช้ข้อมูลชุดเดียวกันแต่อาจจะอยู่ในรูปแบบที่ไม่เหมือนกัน ดังนั้นผลก็คือ จะมีข้อมูลชุดเดียวกันอยู่ในหลายแฟ้มข้อมูล ซึ่งจะก่อให้เกิดปัญหาต่างๆ เช่น

1. เกิดความซ้ำซ้อนของข้อมูล (Data Redundancy) ซึ่งนำไปสู่การเปลืองเนื้อที่จัดเก็บข้อมูล และเปลืองค่าใช้จ่าย

2. ข้อมูลมีความไม่คงที่ (Data Inconsistency) นั้นคือว่ามีข้อมูลชุดเดียวกันอยู่หลายที่ (หลายแฟ้มข้อมูล) ดังนั้นอาจเกิดการเปลี่ยนแปลงที่ไม่เหมือนกัน เช่น เมื่อเปลี่ยนแปลงข้อมูลในแฟ้มหนึ่ง แต่ไม่เปลี่ยนแปลงในอีกแฟ้มหนึ่ง ทั้งๆ ที่เป็นข้อมูลเดียวกัน

3. ข้อมูลอยู่กันอย่างกระจัดกระจาย (Data Isolation) ดังนั้นอาจมีรูปแบบของแฟ้มข้อมูล (Format) ที่แตกต่างกัน จึงยากแก่การเขียนโปรแกรมประยุกต์ขึ้นมาสำหรับสืบค้น (Retrieve) ข้อมูลที่เหมาะสมได้

4. ความปลอดภัยของข้อมูล (Data Security) เนื่องมาจากว่าในระบบแฟ้มข้อมูลแบบคอนเวนชันนัลนั้นข้อมูลอยู่อย่างกระจัดกระจาย และไม่มีการกำหนดสิทธิในการเข้าถึงข้อมูล

5. ข้อมูลไม่มีบูรณภาพ (Integrity) ในระบบแฟ้มข้อมูลแบบคอนเวนชันนัลนั้นเมื่อต้องการเพิ่มเงื่อนไขต่างๆ เงื่อนไขนั้นก็จะต้องถูกเพิ่มเข้าไปในแฟ้มข้อมูลทุกๆ แฟ้ม ถ้าเพิ่มไม่ครบแล้วข้อมูลก็จะไม่มีบูรณภาพ

จุดประสงค์ของวิธีการฐานข้อมูล

การนำวิธีการของฐานข้อมูลมาใช้จัดการแหล่งข้อมูลขององค์กรนั้นมีจุดประสงค์ดังนี้

- ป้องกันคุณค่าของข้อมูล เพราะมีการควบคุมข้อมูลให้มีบูรณภาพ
- มีการประมวลผลข้อมูลให้ตอบสนองต่อความต้องการสารสนเทศใหม่ๆ ที่องค์กรต้องการได้ บางระบบจัดการฐานข้อมูลได้รวมเทคนิคของโครงสร้างข้อมูล (Data Structure) ที่ยืดหยุ่น เพื่อสามารถทำให้จัดการกับสิ่งที่เปลี่ยนแปลงได้อย่างสะดวกดังนั้น ผู้ออกแบบจะมีอิสระ

จากความจำเป็นที่จะต้องรู้รายละเอียดของโครงสร้างต่างๆ หรือสื่อที่ใช้เก็บข้อมูล

วิธีการของฐานข้อมูลใช้เทคนิคของโมเดลข้อมูลเชิงตรรก ที่อนุญาตให้แหล่งข้อมูลถูกกำหนดเป็นอิสระจากการติดตั้ง (Implement) จริงๆของแหล่งข้อมูลนั้น วิธีการของฐานข้อมูลจะเน้นถึงการจัดการของความหมายของข้อมูลมากกว่า เพื่อให้องค์กรมีระบบการจัดการ การตัดสินใจที่ดีขึ้น และเพื่อช่วยลดค่าใช้จ่ายที่เกี่ยวกับการปรับปรุงประสิทธิภาพ และการเปลี่ยนแปลงของการพัฒนาเทคโนโลยีทางคอมพิวเตอร์ ระบบจัดการฐานข้อมูลสามารถจัดการให้ผู้ใช้ได้รับอิสระจากรายละเอียดของสื่อการในการเก็บข้อมูล ซึ่งสามารถถูกเปลี่ยนแปลงให้มีประสิทธิภาพเพิ่มขึ้นได้โดยปราศจากการเปลี่ยนแปลงแหล่งข้อมูล หรือวิธีการที่ใช้เข้าถึงข้อมูลนั้นๆ

<u>โครงสร้างของฐานข้อมูล</u>

ระบบจัดการฐานข้อมูลเป็นที่รวบรวมของแฟ้มข้อมูล และเซตของโปรแกรมต่างๆ ที่อนุญาตให้ผู้ใช้หลายคนเข้าถึง และเปลี่ยนแปลงแฟ้มข้อมูลเหล่านั้น จุดประสงค์หลักของระบบฐานข้อมูลก็คือจัดหามุมมองของข้อมูลให้กับผู้ใช้นั้นคือ ระบบจะซ่อนรายละเอียดต่างๆ ที่ว่าข้อมูลเหล่านั้นถูกเก็บ และถูกบำรุงรักษาไว้อย่างไร อย่างไรก็ตามเพื่อที่จะให้ได้ระบบที่มีประโยชน์มากที่สุด ข้อมูลจะต้องถูกสืบค้นได้อย่างมีประสิทธิภาพมากที่สุด

การพิจารณาถึงประสิทธิภาพนี้ นำไปสู่การออกแบบโครงสร้างข้อมูลที่สลับซับซ้อนสำหรับเป็นตัวแทนของข้อมูลในฐานข้อมูล อย่างไรก็ตามระบบฐานข้อมูลโดยทั่วไปจะถูกใช้โดยบุคคลที่ไม่ได้มีความรู้ทางคอมพิวเตอร์เท่าไรนัก ดังนั้นความยุ่งยากจึงต้องถูกแฝงไว้จากผู้ที่จะใช้ระบบฐานข้อมูล ซึ่งจะทำให้ประสบผลสำเร็จได้โดยการกำหนดระดับต่างๆ ของฐานข้อมูลได้ 3 ระดับ เพื่อที่จะแยกทางที่ฐานข้อมูลถูกแทนที่ทางกายภาพจากทางที่ผู้ใช้คิด หรือพิจารณาฐานข้อมูลนั้น โดยมีเหตุผลต่างๆ ในการแบ่งระดับฐานข้อมูลดังนี้

ก. ผู้ใช้ต่างๆ กันต้องการมองข้อมูลชุดเดียวกันต่างมุมมองกัน

ข. วิธีการที่ผู้ใช้ต้องการดูข้อมูลนั้นต่างกันออกไป

ค. ผู้ใช้ไม่ควรที่จะยุ่งเกี่ยวกับความยุ่งยากซับซ้อนของโครงสร้างที่เก็บฐานข้อมูล

ง. เพื่อให้ผู้บริหารฐานข้อมูล (Database Administrator) เปลี่ยนแปลงโครงสร้างทางกายภาพของฐานข้อมูลโดยปราศจากการกระทบต่อผู้ใช้

จ. เพื่อให้ผู้บริหารฐานข้อมูลเปลี่ยนแปลงข้อมูล และโครงสร้างของแฟ้มข้อมูลโดยปราศจากการกระทบโครงสร้างทางกายภาพทั้งหมด หรือ มุมมองของผู้ใช้ (User View) ทั้งหมด

ฉ. โครงสร้างของฐานข้อมูลจะไม่ถูกกระทบเมื่อมีการเปลี่ยนแปลงอุปกรณ์ต่างๆ เช่น อุปกรณ์ที่เก็บข้อมูล เป็นต้น

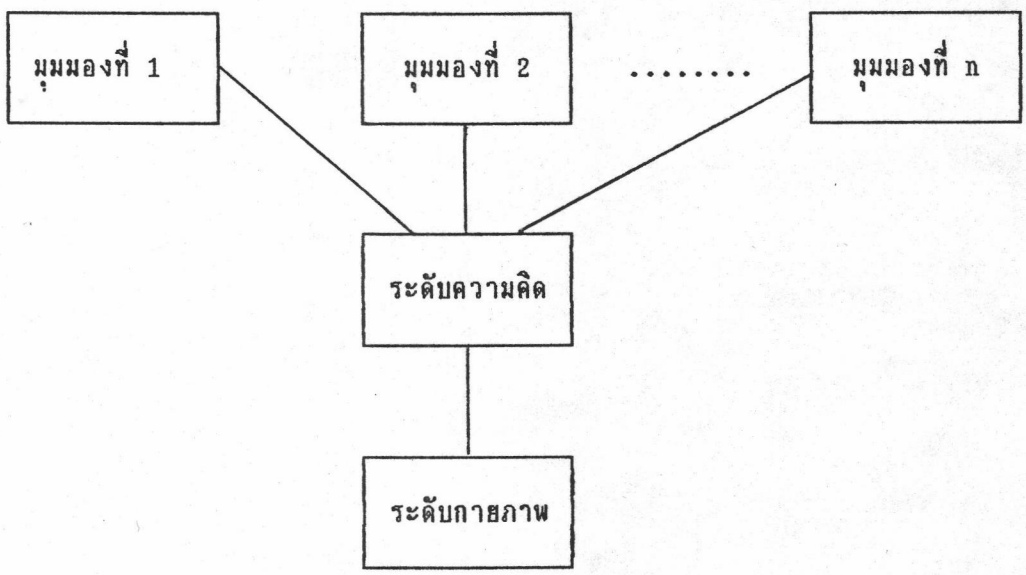
ระดับของฐานข้อมูลทั้ง 3 ระดับได้แก่

1. ระดับภายใน (Internal หรือ Physical Level) เป็นระดับที่ต่ำที่สุดที่ใช้อธิบายว่าข้อมูลจะถูกเก็บจริงๆ ได้อย่างไร ในระดับนี้เป็นโครงสร้างข้อมูลที่ซับซ้อน

2. ระดับความคิด (Conceptual Level) เป็นระดับที่ใช้อธิบายว่าข้อมูลอะไรที่จะถูกเก็บอยู่จริงในฐานข้อมูล และความสัมพันธ์ระหว่างข้อมูลที่มีอยู่ ถึงแม้ว่าโครงสร้างของระดับความคิดนี้จะเกี่ยวพันกับโครงสร้างที่ซับซ้อนในระดับภายในก็ตาม ผู้ใช้ในระดับความคิดไม่จำเป็นที่จะต้อง หรือคำนึงในเรื่องนี้ ระดับความคิดนี้จะถูกใช้โดยผู้บริหารฐานข้อมูลผู้ซึ่งต้องตัดสินใจว่าสารสนเทศใดจะถูกเก็บลงในฐานข้อมูล

3. ระดับภายนอก (External Level หรือ View Level) เป็นระดับที่สูงสุดในการอธิบายส่วนใดส่วนหนึ่งของฐานข้อมูล ผู้ใช้หลายๆ คนของระบบฐานข้อมูลจะไม่เกี่ยวข้องกับสารสนเทศทุกๆ ส่วนแต่จะเกี่ยวข้องในบางส่วนของฐานข้อมูลเท่านั้น เพื่อที่จะให้การติดต่อกับผู้ใช้และระบบง่ายขึ้นระดับภายนอกนี้จึงถูกกำหนดขึ้น ซึ่งจะประกอบไปด้วยหลายๆ มุมมองของผู้ใช้สำหรับฐานข้อมูลเดียวกัน

ความสัมพันธ์ทั้ง 3 ระดับนั้นจะเห็นได้ดังรูปที่ 3.1



รูปที่ 3.1 แสดงการแบ่งระดับโครงสร้างของฐานข้อมูล

<u>ขั้นตอนการออกแบบฐานข้อมูล</u>

ในการวิเคราะห์ระบบนั้นจะพบว่า เมื่อมีการเปลี่ยนแปลงใหม่ๆ เกิดขึ้นมาระบบเก่าจะต้องถูกแทนที่ แต่ในสภาพแวดล้อมของฐานข้อมูลนั้นฐานข้อมูลจะถูกออกแบบในทางที่เกี่ยวข้องและเกี่ยวพันกับการเปลี่ยนแปลงสารสนเทศต่างๆ จะเกิดขึ้นในอนาคตขององค์กรด้วย การออก

แบบฐานข้อมูลนี้จะเป็นไปได้เมื่อ ผู้ออกแบบพัฒนาโมเดลข้อมูลเชิงตรรกสำหรับองค์กรให้มีคุณสมบัติดังต่อไปนี้

ก. โมเดลเป็นกระจกสะท้อนถึงการปฏิบัติต่างๆ ขององค์กร

ข. ยืดหยุ่นเพียงพอที่สามารถให้มีการเปลี่ยนแปลงสารสนเทศใหม่ๆ ขึ้นมาได้

ค. สนับสนุนหลายๆ มุมมองของผู้ใช้

ง. เป็นอิสระต่อการติดตั้งทางกายภาพ (Physical Implementation)

จ. ไม่ขึ้นกับโมเดลใดโมเดลหนึ่งที่ถูกใช้โดยระบบจัดการฐานข้อมูลใดโดยเฉพาะ

การออกแบบฐานข้อมูลที่ดีจะป้องกันแหล่งข้อมูลให้ใช้สารสนเทศต่างๆ ได้ทั้งในปัจจุบันและในอนาคต ถ้าระบบเป็นอิสระต่อการติดตั้งทางกายภาพจริงๆ แล้วระบบสามารถถูกเปลี่ยนไปใช้ฮาร์ดแวร์ใหม่ หรือใช้ระบบจัดการฐานข้อมูลใหม่ก็ได้ซึ่งไม่สามารถทำให้โมเดลของข้อมูลเชิงตรรกเปลี่ยนแปลงเลย

ขั้นตอนในการออกแบบฐานข้อมูลนั้นเป็นวิธีการออกแบบจากบนลงล่าง (Top-down Design) ซึ่งเริ่มโดยพิจารณาความต้องการต่างๆ อย่างกว้างๆ แล้วค่อยลงรายละเอียดต่างๆ การออกแบบฐานข้อมูลนั้นมีขั้นตอนต่างๆ ดังต่อไปนี้

1. วิเคราะห์สภาพแวดล้อมของผู้ใช้ ขั้นตอนแรกในการออกแบบฐานข้อมูลถูกกำหนดโดยสภาพแวดล้อมของผู้ใช้ปัจจุบัน ผู้ออกแบบควรจะศึกษาทุกๆ งานประยุกต์ (Applications) ที่ใช้อยู่ในปัจจุบัน กำหนด อินพุต และ เอาต์พุต ตรวจสอบทุกๆ รายงานที่เป็นของระบบปัจจุบัน และสัมภาษณ์ผู้ใช้ว่าเขาใช้ระบบได้อย่างไร หลังจากที่เข้าใจระบบปัจจุบันแล้วผู้ออกแบบควรจะถามความต้องการของผู้ใช้ว่าในระบบใหม่นั้นเขาต้องการอะไรบ้าง ผู้ออกแบบไม่ควรพิจารณาแต่เพียงความต้องการในปัจจุบันเท่านั้น แต่ควรจะคำนึงถึงความต้องการในอนาคตด้วย ผลลัพธ์ของการวิเคราะห์ก็คือ โมเดลของสภาพแวดล้อมของผู้ใช้ และความต้องการผู้ใช้

2. พัฒนาโมเดลข้อมูลเชิงตรรกจากสภาพแวดล้อมของผู้ใช้ ผู้ออกแบบควรพัฒนารายละเอียดของโมเดลข้อมูลเชิงตรรกโดยการกำหนด เอนติตี แอตตริบิวต์ และรีเลชันชิป นอกจากนี้ผู้ออกแบบควรพิจารณาว่าฐานข้อมูลนี้ถูกใช้ได้อย่างไร ควรกำหนดประเภทต่างๆ ของงานประยุกต์ รายการเปลี่ยนแปลง (Transaction) ชนิดของการเข้าถึง ปริมาณของรายการเปลี่ยนแปลง ปริมาณของข้อมูล ความถี่ในการเข้าถึง และจำนวนข้อมูล นอกจากนี้แล้ว เงื่อนไขต่างๆ ที่ควรกำหนดได้แก่ งบประมาณ และ ประสิทธิภาพ

3. เลือกระบบจัดการฐานข้อมูล ผู้ออกแบบควรเลือกระบบจัดการฐานข้อมูลที่เข้าได้กับ ฮาร์ดแวร์ และซอฟต์แวร์ที่มีอยู่

4. ผู้ออกแบบต้องแปลง (Map) โมเดลข้อมูลเชิงตรรก ให้เข้ากับโครงสร้างของระบบจัดการฐานข้อมูลที่เลือกใช้

5. ผู้ออกแบบวางรูปแบบของโครงสร้างของข้อมูล (Physical Model) ที่สนับสนุนโดย ระบบจัดการฐานข้อมูล ฮาร์ดแวร์ และซอฟต์แวร์ที่มีอยู่

6. ผู้ออกแบบประมาณประสิทธิภาพ ของทุกๆ งานประยุกต์ และรายการเปลี่ยนแปลงโดยพิจารณาปริมาณข้อมูล โดยการทำต้นแบบซึ่งเป็นส่วนหนึ่งของการ ติดตั้งฐานข้อมูลจนกระทั่งมุมมองของผู้ใช้ได้ถูกตรวจสอบ และวัดประสิทธิภาพ

7. ทำการปรับปรุงต่างๆ เพื่อให้ประสิทธิภาพ ดีขึ้นเช่นการปรับปรุงเปลี่ยนแปลงโครงสร้างทางกายภาพ เป็นต้น

8. ถ้าผลการประเมินเป็นที่น่าพอใจก็ให้ติดตั้งโมเดลข้อมูลทางกายภาพ เพื่อให้ฐานข้อมูลเริ่มใช้งานได้

<u>โมเดลข้อมูลเชิงตรรก</u>

โมเดลข้อมูลเชิงตรรกเป็นเทคนิคที่ใช้เป็นอินพุต ในการออกแบบฐานข้อมูลโดยการแทนโครงสร้าง และสารสนเทศต่างๆ ขององค์กร โดยที่เทคนิคนี้จะมองข้อมูลว่าเป็นแหล่งที่มีค่าในการทำธุรกิจองค์กร ปรัชญาของโมเดลข้อมูลเชิงตรรกก็คือ การวิเคราะห์และบันทึกความจริงที่ว่าข้อมูลขององค์กรมีอยู่จริง และเป็นอิสระว่าข้อมูลเหล่านั้นจะถูกเข้าถึงได้อย่างไร โดยใคร และใช้คอมพิวเตอร์หรือไม่ก็ได้ เช่น ลูกค้าสั่งสินค้า ลูกค้าจ่ายเงิน ความจริงเหล่านี้ที่เกิดขึ้นในธุรกิจขององค์กรซึ่งสามารถแทนลงในโมเดลข้อมูลเชิงตรรกได้ ถึงแม้ว่าจะไม่ได้ติดตั้งความจริงที่เกิดขึ้นในธุรกิจขององค์กรลงในฐานข้อมูลได้ครบก็ตาม ก็จะต้องกำหนดความจริงเหล่านี้ลงในโมเดลข้อมูลเชิงตรรกด้วย

คุณสมบัติของวิธีการของโมเดลข้อมูลเชิงตรรก

1. เป็นวิธีการที่ทำความเข้าใจความต้องการสารสนเทศต่างๆ ขององค์กรได้ดี

2. ช่วยส่งเสริมการสื่อสาร ความเข้าใจ ระหว่างผู้ออกแบบ ผู้พัฒนา และผู้ใช้ตลอดระยะเวลาในการออกแบบระบบ

3. เป็นวิธีการที่รวบรวมพื้นฐานในการออกแบบโดยให้ข้อมูลในองค์กรมีความถูกต้องคงที่ สามารถใช้ร่วมกัน และยืดหยุ่นได้ ไม่ว่าจะใช้เทคโนโลยีของฐานข้อมูลใดๆ ก็ตาม

ประโยชน์ของวิธีการของโมเดลข้อมูลเชิงตรรก

1. ง่ายเพราะแสดงด้วยรูปภาพที่ไม่ซับซ้อน

2. แสดงความจริงของธุรกิจในองค์กรที่แจ่มชัด และไม่กำกวม

โครงสร้างที่สำคัญของโมเดลข้อมูลเชิงตรรกประกอบไปด้วย เอนติตี และรีเลชันชิป เอนติตี หมายถึง คน สถานที่ สิ่งของ หรือ แนวคิดที่ต้องการบันทึก เช่น ผู้เช่า อาคารพาณิชย์ เป็นต้น รีเลชันชิป หมายถึง ความสัมพันธ์ระหว่าง 2 เอนติตี เช่น ผู้เช่าเช่าอาคารพาณิชย์ อาคารพาณิชย์ถูกเช่าโดยผู้เช่า เป็นต้น ในโมเดลข้อมูลเชิงตรรกแทน เอนติตีด้วยรูปสี่เหลี่ยม และแทนรีเลชันชิปด้วยลูกศร จะต้องกำหนดชื่อของเอนติตี เช่น ชื่อเอนติตีว่า ผู้เช่า และคำอธิบาย เอนติตี นอกจากนี้ที่จะต้องเก็บรายละเอียดของผู้เช่าแต่ละคน โดยมีรายละเอียดต่างๆ

เช่น ชื่อ ที่อยู่ เบอร์โทรศัพท์ เป็นต้น เรียกรายละเอียดต่างๆ เหล่านี้ว่าเป็น แอตตริบิวต์ โดยที่ แอตตริบิวต์หมายถึง ความจริง หรือ สารสนเทศที่แยกออกไม่ได้แล้วในการอธิบายเอนติตี ซึ่งเป็นหน่วยที่เล็กที่สุดของสารสนเทศที่ต้องการอ้างถึงในขณะใดขณะหนึ่ง

นอกจากนี้แล้วต้องบันทึกความหมาย หรือคำจำกัดความของแอตตริบิวต์ลงในพจนานุกรมข้อมูลด้วยเพราะการเป็นโมเดลข้อมูลเชิงตรรกที่สมบูรณ์นั้น ไม่ได้ประกอบไปด้วยแผนภาพเท่านั้น แต่จะต้องเก็บข้อกำหนดต่างๆ ลงในพจนานุกรมข้อมูลด้วย พจนานุกรมข้อมูลก็คือคู่มือ หรือที่ที่เป็นที่เก็บสารสนเทศเกี่ยวกับโปรแกรมประยุกต์ ฐานข้อมูล โมเดลข้อมูลเชิงตรรก ผู้ใช้ และอำนาจในการเข้าถึงข้อมูล (Access Authorizations)

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

ในขั้นตอนแรกของการสร้างโมเดลข้อมูลเชิงตรรกจำเป็นต้องพิจารณาการสร้าง โครงสร้างจากมุมมองของผู้ใช้ (Skeletal User View) โดยที่มุมมองของผู้ใช้หมายถึง โมเดลที่เป็นตัวแทนความต้องการจากผู้ใช้คนใดคนหนึ่ง หรือหน้าที่ส่วนใดส่วนหนึ่งของระบบ

โครงสร้างจากมุมมองของผู้ใช้ประกอบไปด้วย ความต้องการต่างๆ ของระบบนั่นคือจะมีเทอมต่างๆ ได้แก่ เอนติตี รีเลชันชิป คีย์หลัก (Primary Key) คีย์รอง (Alternate Key) และ คีย์ของกฎธุรกิจ (Key Business Rule) โครงสร้างจากมุมมองของผู้ใช้เป็นประโยชน์อย่างมากในการเป็นอินพุตของการออกแบบฐานข้อมูล รวมทั้งการออกแบบต้นแบบของฐานข้อมูล

ต้นแบบ หมายถึง การทดลองสร้างบางส่วนของการติดตั้งในที่นี้หมายถึง การติดตั้งฐานข้อมูล การออกแบบต้นแบบนี้มีจุดประสงค์เพื่อทดสอบว่าระบบมีความต้องการเพียงพอหรือไม่

ขั้นตอนในการออกแบบโมเดลข้อมูลเชิงตรรก

1. กำหนดเอนติตีหลัก

เอนติตี หมายถึง คน สถานที่ สิ่งของ หรือ ความคิดที่ต้องการบันทึกสารสนเทศต่างๆ เอนติตีแบ่งออกได้เป็น เอนติตีที่เป็นซับไทป์ ซึ่งมีข้อกำหนดและคุณสมบัติต่างๆ เพิ่มขึ้นมากกว่าเอนติตีที่เป็นซูเปอร์ไทป์ ซึ่งทั้งเอนติตีที่เป็นซับไทป์ และเอนติตีที่เป็นซูเปอร์ไทป์แทนสิ่งเดียวกันเช่น ลูกจ้างเป็นเอนติตีที่เป็นซูเปอร์ไทป์ โดยมีวิศวกร ผู้จัดการ เสมียน เป็นเอนติตีที่เป็นซับไทป์เป็นต้น เราสามารถกำหนดซับไทป์ได้หลายๆ ซับไทป์ สำหรับหนึ่งซูเปอร์ไทป์ โดยทั่วไปเอนติตี ก เป็นซับไทป์ของเอนติตี ข และเอนติตี ข เป็น ซูเปอร์ไทป์ของเอนติตี ก ก็ต่อเมื่อ

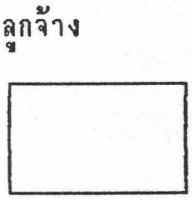
1.1 ก และ ข แทนสิ่งเดียวกัน

1.2 ก มีคุณสมบัติเดียวกับ ข นอกจากนี้ยังมีคุณสมบัติที่พิเศษต่างไปจาก ข

1.3 ทุกๆ ค่าของ ก จะมีอยู่ในค่าๆ หนึ่งของ ข แต่ในทางกลับกันคือ ทุกๆ ค่าของ ข ไม่จำเป็นต้องอยู่ในค่าๆ หนึ่งของ ก

เมื่อกำหนดเอนติตีหลักได้แล้วต้องกำหนดชื่อ แผนภาพ และความหมายของเอนติตี แล้วบันทึกลงในพจนานุกรมข้อมูล

แผนภาพโมเดลข้อมูลเชิงตรรก เป็นแผนภาพที่ใช้แทนโมเดลข้อมูลเชิงตรรก ซึ่งประกอบไปด้วยสารสนเทศ และความสัมพันธ์ต่างๆ เอนติตีเป็นสิ่งแรกที่ถูกวาดลงในแผนภาพ เอนติตีถูกแทนด้วยรูปสี่เหลี่ยมโดยจะเขียนชื่อของเอนติตีที่มุมบนด้านซ้ายของสี่เหลี่ยมนั้นดังรูปที่ 3.2



รูปที่ 3.2 แสดงการแทนเอนติตีด้วยภาพ

2. กำหนดรีเลชันชิประหว่างเอนติตี

รีเลชันชิป หมายถึง ความจริงที่เกี่ยวข้องกับความสัมพันธ์ระหว่างเอนติตี 2 เอนติตี มีรีเลชันชิปอยู่สามแบบ คือ

ก. รีเลชันชิปที่มีอยู่จริง (Existence Relationships) เช่น ลูกจ้างมีลูก

ข. รีเลชันชิปตามหน้าที่ (Functional Relationships) เช่น ครูสอนนักเรียน

ค. รีเลชันชิปตามเหตุการณ์ (Event Relationships) เช่น ลูกค้าสั่งสินค้า

เมื่อกำหนดรีเลชันชิปแล้วให้กำหนดชื่อ แผนภาพ และบันทึกลงในพจนานุกรมข้อมูล รีเลชันชิปมีทิศทางโดยจะอยู่ระหว่างเอนติตี 2 เอนติตีที่เกี่ยวข้องกันโดยมีทิศทางจาก เอนติตีหนึ่งซึ่งเรียกว่าเอนติตีแม่ (Parent Entity) ไปยังอีกเอนติตีซึ่งเรียกว่าเอนติตีลูก (Child Entity)

รีเลชันชิปแบ่งออกเป็นชนิดต่างๆ ตามสัดส่วนความสัมพันธ์(Cardinality Ratio) ได้ สามแบบคือ

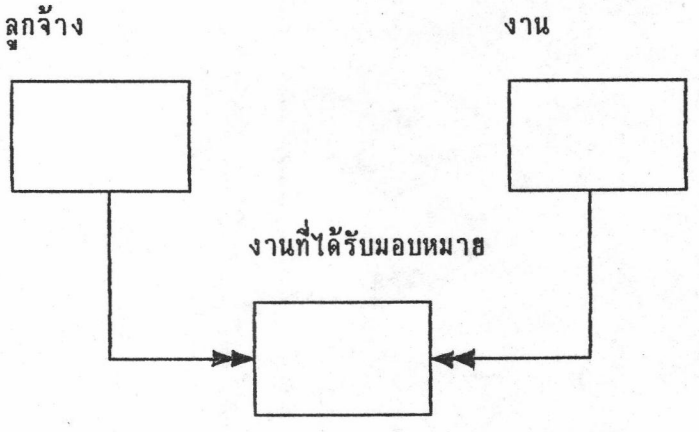
ก. แบบหนึ่งต่อหนึ่ง (One to one หรือ 1:1 Relationship) แต่ละค่าของเอนติตีแม่มีความสัมพันธ์กับเอนติตีลูกได้อย่างมากที่สุดหนึ่งค่า

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

ค. แบบหลายต่อหลาย (Many to many หรือ M:N Relationship) แต่ละค่าของเอนติตีแม่มีความสัมพันธ์กับเอนติตีลูกได้หลายค่า และในทางกลับกันแต่ละค่าของเอนติตีลูกมีความสัมพันธ์กับเอนติตีแม่ได้หลายค่า

ในโมเดลข้อมูลเชิงตรรกกำหนดรีเลชันชิประหว่างเอนติตีได้โดยวาดเส้นระหว่างรูปสี่เหลี่ยมที่ใช้แทนเอนติตีแม่ และเอนติตีลูก ใส่หัวลูกศรหนึ่งลูกชี้ไปในทิศทางของความสัมพันธ์ที่เป็นจริงที่เอนติตีลูก เพื่อแทนความสัมพันธ์แบบหนึ่งต่อหนึ่งใส่หัวลูกศร 2 ลูกแทนความสัมพันธ์แบบหนึ่งต่อหลาย มีข้อยกเว้นสำหรับความสัมพันธ์แบบหลายต่อหลาย คือการแทนความสัมพันธ์แบบนี้ค่อนข้างจะยุ่งยากในวิธีการของโมเดลข้อมูลเชิงตรรก ตลอดจนการออกแบบฐานข้อมูล ดังนั้นวิธีการที่ทำให้ง่ายขึ้นได้โดยแปลงความสัมพันธ์แบบหลายต่อหลายหนึ่งความสัมพันธ์มาเป็นความสัมพันธ์แบบหนึ่งต่อหลาย 2 ความสัมพันธ์

เช่น จากความสัมพันธ์แบบหลายต่อหลาย คือ ลูกจ้างคนหนึ่งทำงานหลายงาน และ งานงานหนึ่งทำได้โดยลูกจ้างหลายคน จากเดิมมีเอนติตีคือ ลูกจ้างและงาน ดังนั้นวิธีการแปลงต้องสร้างเอนติตีขึ้นมาใหม่ คืองานที่ได้รับมอบหมาย โดยมีความสัมพันธ์ดังรูปที่ 3.3



รูปที่ 3.3 แสดงความสัมพันธ์ของ 2 เอนติตีที่มีความสัมพันธ์แบบหลายต่อหลาย

ถ้ามีรีเลชันชิปที่เกิดขึ้นระหว่าง 3 เอนติตีขึ้นไปจะต้องพิจารณาใหม่ให้มีรีเลชันชิประหว่าง 2 เอนติตีเท่านั้น นอกจากนี้ให้กำหนดรีเลชันชิประหว่างเอนติตีที่เป็นซับไทป์ และเอนติตีที่เป็นซูเปอร์ไทป์

3. กำหนดคีย์หลัก และคีย์รอง

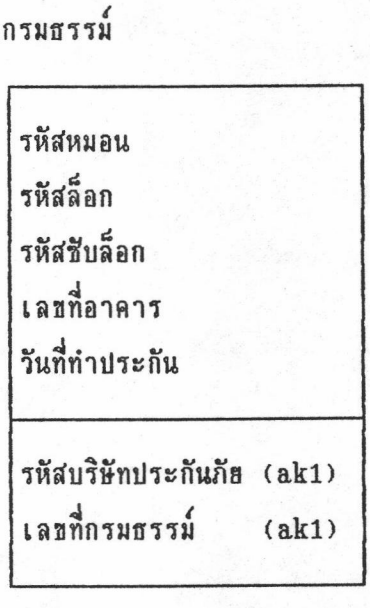
ขั้นตอนนี้เป็นการกำหนดรายละเอียดต่างๆ ของเอนติตีโดยการกำหนดแอตตริบิวต์ โดยที่แอตตริบิวต์ หมายถึง หน่วยที่เล็กที่สุดของข้อมูลที่มีความเกี่ยวข้องกับเอนติตีซึ่งเป็นความจริงที่ไม่สามารถแบ่งเป็นส่วนเล็กๆ ออกไปกว่านี้ได้แล้วในการใช้อธิบายเอนติตีได้แก่ เลขประจำตัวลูกจ้าง ชื่อลูกจ้าง เป็นต้น

แอตตริบิวต์แรกที่ต้องกำหนดลงในโมเดลข้อมูลเชิงตรรกก็คือ คีย์หลัก และคีย์รอง แอตตริบิวต์ใดๆ หรือเซตของแอตตริบิวต์ใดๆ ที่ใช้ระบุค่าใดค่าหนึ่งของเอนติตีได้เป็นหนึ่งค่า (Unique) จะเรียกแอตตริบิวต์นั้นหรือเซตของแอตตริบิวนั้นว่าเป็น แคนดิเดทคีย์ (Candidate Key) แต้ถ้า แคนดิเดทคีย์ประกอบไปด้วยแอตตริบิวต์มากกว่าหนึ่งตัว คือเป็นเซตของแอตตริบิวต์ จะเรียกว่าเป็น คีย์ประกอบ (Composite Key)

ให้เลือกคีย์หลักมาหนึ่งค่าสำหรับเอนติตีแต่ละเอนติตีโดยที่แคนดิเดทคีย์ใดๆ ที่ไม่ถูกเลือกมาเป็นคีย์หลัก จะเรียกว่าเป็นคีย์รอง (Alternate Key)

สำหรับคีย์หลักของเอนติตีที่เป็นซับไทป์นั้น จะต้องเหมือนกับคีย์หลักของเอนติตีที่เป็นซูเปอร์ไทป์ เมื่อกำหนดคีย์หลัก และคีย์รองของเอนติตีได้แล้ว ให้บันทึก ชื่อ และความหมายลงในพจนานุกรมข้อมูล ในการตั้งชื่อแอตตริบิวต์ควรให้มีมาตรฐานในการตั้งชื่อ (รวมทั้งการตั้งชื่อเอนติตี และรีเลชันชิปด้วย)

ในแผนภาพโมเดลข้อมูลเชิงตรรกนั้นต้องเขียนชื่อของคีย์หลักให้อยู่ในรูปสี่เหลี่ยมที่ใช้แทนเอนติตีโดยให้อยู่เหนือเส้นนอน และเขียนชื่อแอตตริบิวต์ที่เป็นคีย์รองอยู่ใต้เส้นนอนแล้วตามด้วยสัญญลักษณ์ akn หลังชื่อ แอตตริบิวต์โดยที่ n คือ แอตตริบิวต์ตัวที่ n ของเอนติตีนั้นดังรูปที่ 3.4

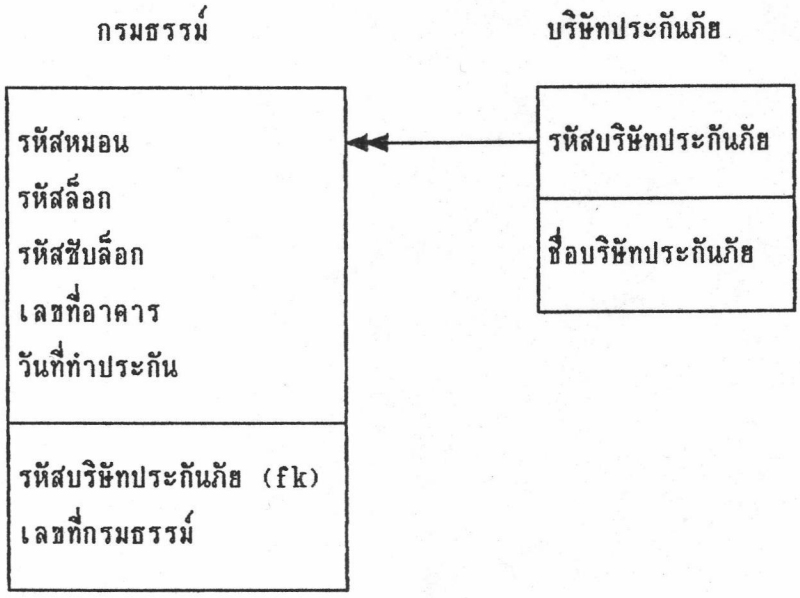


รูปที่ 3.4 แสดงเอนติตีกรมธรรม์ซึ่งแสดงให้เห็นถึงคีย์หลัก และคีย์รอง

4. กำหนดฟอร์เรนจ์คีย์

ฟอร์เรนจ์คีย์ (Foreign Key)เป็นแอตตริบิวต์ หรือเซตของแอตตริบิวต์ที่มีความสัมพันธ์ที่กำหนดโดยเอนติตีแม่ โดยที่ฟอร์เรนจ์คีย์จะถูกกำหนดไว้ที่เอนติตีลูก ซึ่งเทียบเท่ากับเมื่อเป็นคีย์หลักในเอนติตีแม่ คำว่า ฟอร์เรนจ์ (Foreign) บ่งชัดว่าแอตตริบิวต์นั้นอ้างถึง เป็นของหรือมีจุดกำเนิดมาจากเอนติตีอื่น เช่น พิจารณาจากรูปที่ 3.5

จากรูปที่ 3.5 นี้จะเห็นได้ชัดว่าเอนติตีแม่ คือบริษัทประกันภัย เอนติตีลูกคือกรมธรรม์ โดยมีรหัสบริษัทประกันภัย เป็นฟอร์เรนจ์คีย์ในเอนติตีลูก และเป็นคีย์หลักในเอนติตีแม่



รูปที่ 3.5 แสดงความสัมพันธ์ระหว่าง 2 เอนติตีโดยแสดงให้เห็นถึงแอตตริบิวต์ที่เป็นฟอร์เรนจ์คีย์

ต้องกำหนดฟอร์เรนจ์คีย์ให้กับแต่ละรีเลชันชิปในโมเดลข้อมูลเชิงตรรก การกำหนดฟอร์เรนจ์คีย์นี้สำคัญมาก โดยเฉพาะอย่างยิ่งเมื่อสร้างฐานข้อมูลแบบรีเลชันนัล เพราะว่าฟอร์เรนจ์คีย์จะเก็บคุณสมบัติในการอ้างถึงระหว่างตารางรีเลชันนัล นอกจากนี้แล้วให้บันทึกชื่อความหมาย ความสัมพันธ์ที่เกี่ยวข้องกันลงในพจนานุกรมข้อมูล

5. กำหนดคีย์ของกฎธุรกิจ (Key Business Rules)

การกำหนดข้อมูลให้มีความเป็นบูรณภาพนั้นเป็นส่วนหนึ่งของโมเดลข้อมูลเชิงตรรก เพื่อทำให้เกิดความมั่นใจว่าข้อมูลมีความถูกต้อง และมีความคงที่

วิธีการกำหนดกฎธุรกิจ (Business Rule) ของเอนติตีมีอยู่ 3 ประเภทคือ

ก. คีย์ของกฎธุรกิจ (Key Business Rules) จะเกี่ยวข้องกับความเป็นบูรณภาพของรีเลชันชิปโดยที่กฎเหล่านี้ครอบคลุมผลของคำสั่งการแทรก (Insert) การลบ (Delete) การปรับปรุง (Update) ที่กระทำต่อรีเลชันชิป เช่น จะมีผลกระทบอย่างไรบ้างเมื่อมีการเปลี่ยนแปลงของคีย์หลัก หรือฟอร์เรนจ์คีย์

ข. โดเมน (Domains) จะพิจารณาถึงความเป็นบูรณภาพของแอตตริบิวต์โดยทั่วๆ ไป และเงื่อนไขของค่าต่างๆ ที่เป็นไปได้ของแอตตริบิวต์

ค. ทริกเกอร์ดำเนินการ (Triggering Operations) จะครอบคลุมผลลัพธ์ของคำสั่งการแทรก การลบ การปรับปรุง และการดึงข้อมูลที่มีผลต่อเอนติตีต่างๆ หรือ

แอตตริบิวต์ต่างๆ ในเอนติตีเดียวกัน

การกำหนดคีย์ของกฎธุรกิจมีขั้นตอนดังต่อไปนี้

5.1 กำหนดกฎการแทรกให้กับแต่ละรีเลชันชิป

กฎการแทรก (Insert Rule) หมายถึง การกำหนดเงื่อนไขที่เป็นไปได้ในการแทรกเอนติตีลูก (หรือการปรับปรุงฟอร์เรนจ์คีย์ของเอนติตีลูก)

กฎการแทรกแบ่งออกเป็น 6 ประเภทคือ

5.1.1 ขึ้นต่อกัน (Dependent) อนุญาตให้แทรกเอนติตีลูกได้เมื่อมีค่าตรงกับเอนติตีแม่ที่มีอยู่

5.1.2 อัตโนมัติ (Automatic) อนุญาตให้แทรกเอนติตีลูกได้เมื่อมีค่าตรงกับเอนติตีแม่ที่มีอยู่ แต่เมื่อค่าในเอนติตีแม่ไม่มีก็ให้สร้างค่าในเอนติตีแม่นั้นขึ้นมาแล้วค่อยแทรกลงในเอนติตีลูก

5.1.3 กำหนดให้มีค่าเป็นนัล (Nullify) อนุญาตให้แทรกเอนติตีลูกได้เมื่อมีค่าตรงกับเอนติตีแม่ที่มีอยู่ แต่เมื่อค่าในเอนติตีแม่ไม่มีให้กำหนดฟอร์เรนจ์คีย์ในลูกให้เป็นนัล

5.1.4 กำหนดให้มีค่าโดยปริยาย (Default) อนุญาตให้แทรกเอนติตีลูกได้เมื่อมีค่าตรงกับเอนติตีแม่ที่มีอยู่ แต่เมื่อค่าในเอนติตีแม่ไม่มีให้กำหนดฟอร์เรนจ์คีย์ในลูกให้มีค่าโดยปริยาย

5.1.5 กำหนดให้มีค่าตามต้องการ (Customized) อนุญาตให้แทรกเอนติตีลูกได้เมื่อมีค่าตรงกับเอนติตีแม่ที่มีอยู่ แต่เมื่อค่าในเอนติตีแม่ไม่มีให้กำหนดฟอร์เรนจ์คีย์ในลูกให้มีค่าตามเงื่อนไขที่กำหนดไว้ตามต้องการ

5.1.6 ไม่มีผลกระทบ (No Effect) กำหนดให้มีการแทรกเอนติตีลูกได้โดยไม่ต้องตรวจสอบจากเอนติตีแม่ว่ามีค่าตรงกับเอนติตีแม่หรือไม่

5.2 กำหนดกฎการลบให้กับแต่ละรีเลชันชิป

กฎการลบ (Delete Rule) หมายถึง การกำหนดเงื่อนไขที่เป็นไปได้ว่าเมื่อมีการลบเอนติตีแม่ (หรือปรับปรุงคีย์หลักของเอนติตีแม่) แล้วจะเกิดอะไรขึ้นกับเอนติตีลูก

กฎการลบแบ่งออกเป็น 6 ประเภทคือ

5.2.1 มีข้อจำกัด (Restrict) ลบเอนติตีแม่ได้เมื่อไม่มีเอนติตีลูกที่ตรงกับเอนติตีแม่

5.2.2 ต่อเนื่อง (Cascade) เมื่อจะลบเอนติตีแม่ให้ดูว่ามีเอนติตีลูกที่ตรงกันหรือไม่ ถ้ามีให้ลบเอนติตีลูกนั้นด้วย

5.2.3 กำหนดให้มีค่าเป็นนัล (Nullify) เมื่อมีเอนติตีลูกที่ตรงกับเอนติตีแม่ที่ต้องการลบ กำหนดฟอร์เรนจ์คีย์ในเอนติตีลูกให้มีค่าเป็นนัล

5.2.4 กำหนดให้มีค่าโดยปริยาย (Default) กำหนดฟอร์เรนจ์คีย์ในเอนติตีลูกให้มีค่าโดยปริยายเมื่อต้องการลบเอนติตีแม่ที่ตรงกับเอนติตีลูก

5.2.5 กำหนดให้มีค่าตามต้องการ (Customized) กำหนดฟอร์เรนจ์คีย์ในเอนติตีลูกให้มีค่าตามเงื่อนไขที่กำหนดไว้ตามความต้องการ

5.2.6 ไม่มีผลกระทบ (No Effect) ลบเอนติตีแม่ได้โดยไม่ต้องตรวจสอบว่ามีเอนติตีลูกที่ตรงกันหรือไม่

การกำหนดคีย์ของกฎธุรกิจนั้นพยายามหลีกเหลี่ยงการใช้กฎการแทรก หรือกฎการลบประเภทกำหนดให้มีค่าเป็นนัล ควรจะใช้กฎประเภทกำหนดให้มีค่าโดยปริยายแทนนอกจากนั้นห้ามกำหนดกฎประเภทกำหนดให้มีค่าเป็นนัล เมื่อฟอร์เรนจ์คีย์เป็นส่วนหนึ่งของคีย์หลักในเอนติตีลูก และควรกำหนดกฎการแทรกสำหรับรีเลชันชิประหว่างเอนติตีที่เป็นซับไทป์ กับเอนติตีที่เป็นซูเปอร์ไทป์ให้เป็นกฎประเภทอัตโนมัติ และสำหรับกฎการลบให้เป็นกฎประเภทต่อเนื่อง

6. กำหนดแอตตริบิวต์ต่างๆ

ในขั้นตอนนี้ให้กำหนดแอตตริบิวต์ต่างๆ ที่ไม่เป็นคีย์ แต่เพื่อเป็นรายละเอียดในการอธิบายเอนติตีได้อย่างแจ่มชัดโดยการกำหนดแอตตริบิวต์ที่ไม่เป็นคีย์มีวิธีการดังต่อไปนี้

6.1 กำหนดแอตตริบิวต์ให้กับเอนติตี โดยที่แอตตริบิวต์เหล่านั้นต้องขึ้นกับคีย์หลักของเอนติตีนั้นๆ ด้วย เช่น ชื่อผู้เช่า (ในเอนติตีผู้เช่า)

6.2 กำหนดแอตตริบิวต์ที่ไม่ได้เป็นคีย์ให้มีมากที่สุดเท่าที่เป็นได้

6.3 ถ้าแอตตริบิวต์ใดแอตตริบิวหนึ่งที่ขึ้นกับคีย์หลักแต่แอตตริบิวต์นั้นมีค่าหลายค่า ให้กำหนดแอตตริบิวต์นั้นใหม่โดยกำหนดให้เป็นเอนติตีลูก ถ้าแอตตริบิวนั้นมีค่าเป็นหนึ่งให้แอตตริบิวต์นั้นเป็นคีย์หลักของเอนติตีลูกที่กำหนดขึ้นมาใหม่ แต่ถ้าไม่ให้นำแอตตริบิวต์นั้นประกอบกับคีย์หลักของเอนติตีเดิมมาเป็นคีย์หลักของเอนติตีลูกดังกล่าว

6.4 กำหนดชื่อของแอตตริบิวต์ ความหมาย ความยาว ชนิด เอนติตีที่เกี่ยวข้อง และระบุว่าเป็นคีย์หลัก คีย์รอง หรือฟอร์เรนจ์คีย์โดยบันทึกลงในพจนานุกรมข้อมูล

6.5 ถ้ามีแอตตริบิวต์ที่ดูเหมือนว่าจะอธิบายรีเลชันชิปให้สร้างรีเลชันชิปนั้นเป็นเอนติตีใหม่ในฐานะที่เป็นเอนติตีลูกของ 2 เอนติตีเดิม

6.6 หลีกเลี่ยงการแทนแอตตริบิวต์ในรูปของการเข้ารหัส (Encoded Attribute) ถ้ารหัสนั้นไม่ได้กำหนดให้มีความหมายต่อระบบ การหลีกเลี่ยงการเข้ารหัสของแอตตริบิวต์นั้นมีเหตุผลดังนี้คือ

- รหัสทำให้เกิดการไม่สื่อความหมายกับผู้ใช้ ซึ่งขัดกับจุดประสงค์ของการสร้างโมเดลข้อมูลเชิงตรรก ที่ว่าต้องทำให้มุมมองของผู้ใช้กระจ่างแจ่มชัด
- ทำให้ต้องมีเอนติตีใหม่ขึ้นมาในโมเดลข้อมูลเชิงตรรกนั้นก็คือ ตารางที่ต้องใช้อธิบายรหัสเหล่านั้นต้องกลายมาเป็นเอนติตีแม่ของทุกเอนติตีที่มีรหัสเหล่านั้นปรากฏอยู่
- การใช้รหัสทำให้ผู้ใช้ปลายทางสอบถามข้อมูลได้ลำบาก

6.7 ไม่ควรกำหนดแอตตริบิวต์ที่เป็นแฟล็กไว้ในโมเดลข้อมูลเชิงตรรก

6.8 ถ้าต้องแทนแอตตริบิวต์ให้อยู่ในรูปการเข้ารหัสแล้ว ต้องทำให้รหัสเหล่านั้นมีค่าที่เป็นอิสระจากกัน (Mutually Independent) พิจารณาตามตารางที่ 3.1 จะเห็นว่ามีรหัสที่มีค่าที่ไม่เป็นอิสระจากกัน ดังนั้นจึงต้องจัดรหัสเสียใหม่ให้มีตารางรหัส 2 ตารางดังตารางที่ 3.2

รหัส	ความหมาย
WE	ลูกจ้างที่เป็นวิศวกรรับเงินรายสัปดาห์
WM	ลูกจ้างที่เป็นผู้จัดการรับเงินรายสัปดาห์
DE	ลูกจ้างที่เป็นวิศวกรรับเงินรายวัน
DM	ลูกจ้างที่เป็นผู้จัดการรับเงินรายวัน

ตารางที่ 3.1 แสดงแอตตริบิวต์ในรูปของการเข้ารหัส และความหมายของรหัสโดยที่รหัสนั้นมีค่าที่ไม่เป็นอิสระจากกัน

รหัส	ความหมาย
W	รับเงินรายสัปดาห์
D	รับเงินรายเดือน

รหัส	ความหมาย
E	ลูกจ้างที่เป็นวิศวกร
M	ลูกจ้างที่เป็นผู้จัดการ

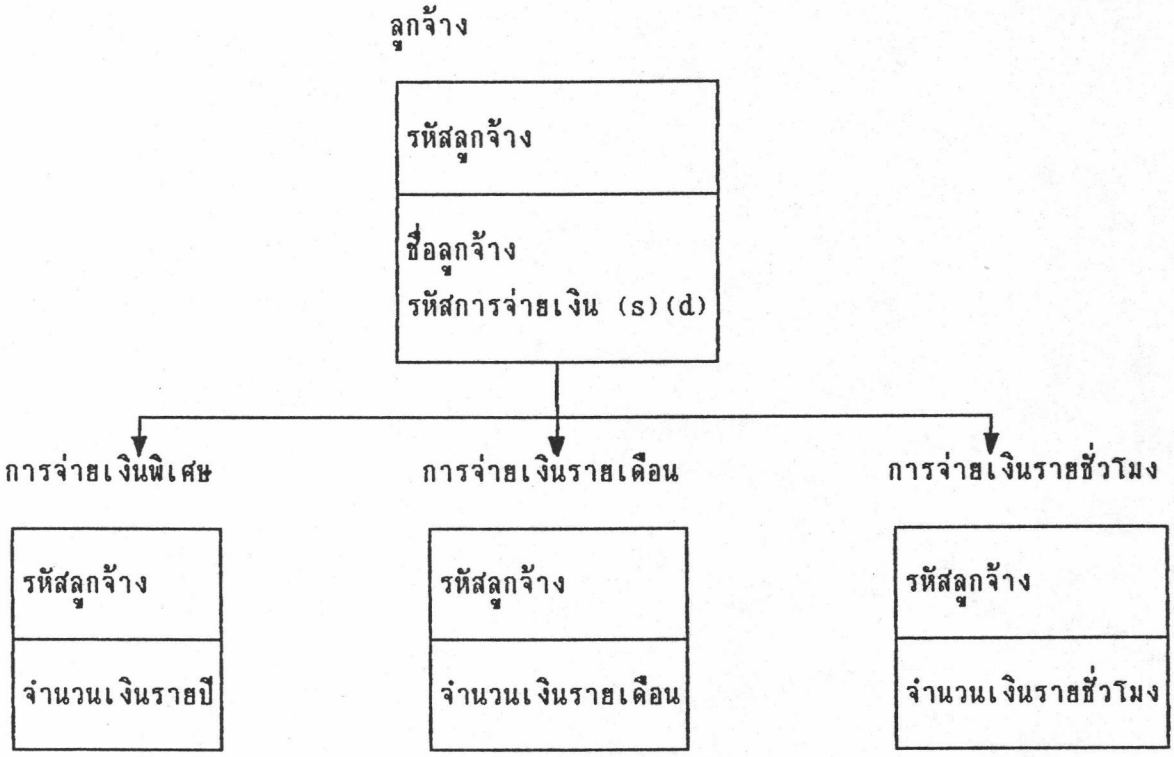
ตารางที่ 3.2 แสดงแอตตริบิวต์ในรูปของการเข้ารหัส และความหมายของรหัสโดยที่รหัสนั้นมีค่าที่เป็นอิสระจากกัน

6.9 แอตตริบิวต์ต่างๆ ที่สามารถหาค่าได้ภายในโมเดลข้อมูลเชิงตรรก ให้ระบุว่าเป็นดิไรฟ์วแอตตริบิวต์ (Derived Attribute)โดยใส่ตัวอักษร d ไว้ที่ท้ายชื่อของแอตตริบิวต์นั้นในแผนภาพของโมเดลข้อมูลเชิงตรรก ในการคำนวณหาดิไรฟ์วแอตตริบิวต์อาจต้องใช้ อัลกอริทึมหรือสูตรในการคำนวณ ดังนั้นจึงต้องบันทึกสูตรในการคำนวณเหล่านี้ลงในพจนานุกรมข้อมูลด้วย

6.10 กำหนดสัญญลักษณ์พิเศษให้กับแอตตริบิวต์ที่ระบุซับไทป์ (Subtype Identifier) โดยใส่ตัวอักษร s ไว้ที่ท้ายชื่อแอตตริบิวต์ในแผนภาพของโมเดลข้อมูลเชิงตรรก

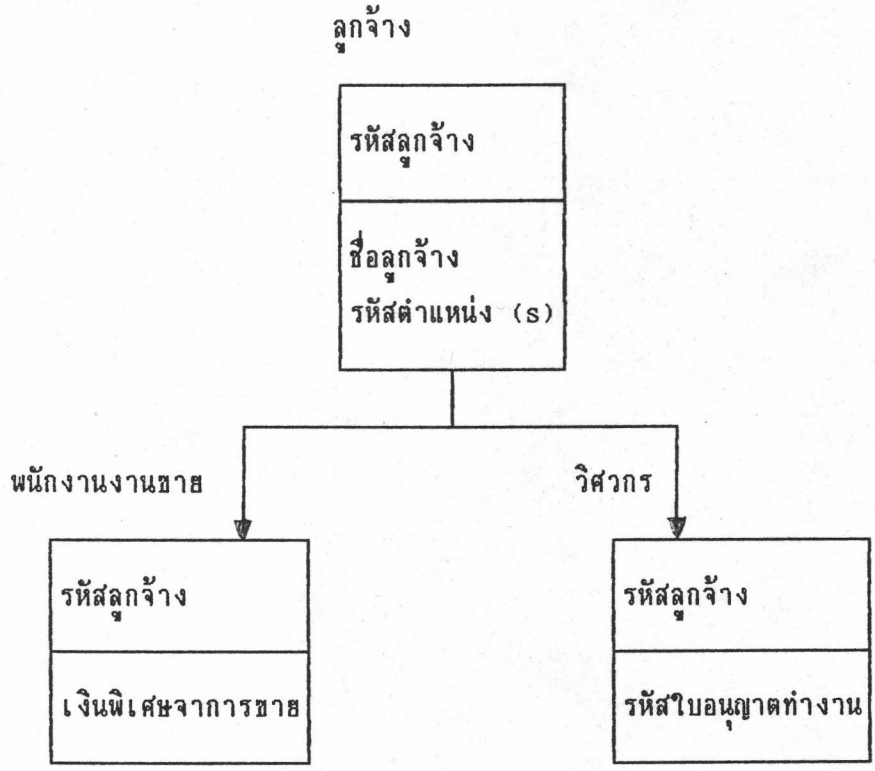
ให้พิจารณาตัวอย่างดังรูปที่ 3.6 ในที่นี้มีรหัสการจ่ายเงินในเอนติตีลูกจ้างเป็นแอตตริบิวต์ที่ระบุซับไทป์ โดยมีเอนติตีจ่ายเงินพิเศษ เอนติตีจ่ายเงินรายเดือน และเอนติตีจ่าย

เงินรายชั่วโมงเป็นเอนติตีที่เป็นซับไทป์ในที่นี้รหัสการจ่ายเงินต้องระบุด้วยว่าเป็นดิไรฟว์แอตตริบิวต์ ถ้าการจ่ายเงินให้แก่ลูกจ้างมีแค่ 3 ประเภท คือ การจ่ายเงินพิเศษ การจ่ายเงินรายเดือน และการจ่ายเงินรายชั่วโมงเท่านั้น ดังนั้นรหัสการจ่ายเงินก็จะเกิดการซ้ำซ้อนขึ้นเพราะสามารถหาได้จากรีเลชันชิประหว่างเอนติตีที่เป็นซับไทป์ และเอนติตีที่เป็นซูเปอร์ไทป์



รูปที่ 3.6 แสดงเอนติตีที่เป็นซูเปอร์ไทป์-ซับไทป์โดยต้องระบุตัวระบุซับไทป์ว่าเป็นดิไรฟว์แอตตริบิวต์

พิจารณาจากรูปที่ 3.7 และตารางที่ 3.3 แล้วจะเห็นว่า รหัสตำแหน่งบางครั้งจะเป็นตัวระบุซับไทป์แต่ในบางครั้งจะไม่เป็นดังนั้น รหัสตำแหน่งไม่ทำให้เกิดความซ้ำซ้อนและไม่ต้องระบุว่าเป็นดิไรฟว์แอตตริบิวต์แต่ต้องระบุว่าเป็นตัวระบุซับไทป์



รูปที่ 3.7 แสดงเอนติตีที่เป็นซูเปอร์ไทป์-ซับไทป์โดยไม่ต้องระบุตัวระบุซับไทป์ว่าเป็นดิไรฟ์ แอตตริบิวต์

ลูกจ้าง

รหัสลูกจ้าง	ชื่อลูกจ้าง	รหัสตำแหน่ง
23312	สมศักดิ์	ผู้จัดการ
12462	สุขุม	วิศวกร
18164	สมศรี	เสมียน
19816	ชูพงษ์	พนักงานขาย

ตารางที่ 3.3 แสดงให้เห็นข้อมูลของเอนติตีลูกจ้างที่แสดงให้เห็นถึงรหัสตำแหน่ง

6.11 ให้ใส่แอตตริบิวต์ที่ใช้ร่วมกันระหว่างเอนติตีที่เป็นซูเปอร์ไทป์ และเอนติตีที่เป็นซับไทป์ไว้ที่เอนติตีที่เป็นซูเปอร์ไทป์

6.12 ให้รวมเอนติตีที่มีคีย์หลักเหมือนกันเอาไว้ด้วยกัน แต่เอนติตีที่จะนำมารวมได้นั้นจะต้องแทนสิ่งๆ เดียวกัน

6.13 รวมเอนติตีที่เป็นซับไทป์ที่มีแอตตริบิวต์ และรีเลชันชิปที่เหมือนกันเอาไว้

เป็นเอนติตีเดียวกัน

6.14 ให้รวมเอนติตีที่ไม่มีแอตตริบิวต์ที่ไม่เป็นคีย์ (Nonkey Attribute)ไว้ที่เอนติตีลูก

7. พิจารณาให้อยู่ในกฎนอร์มัลไลเซชัน

นอร์มัลไลเซชันเป็นทฤษฎีของการวิเคราะห์ และแยกโครงสร้างของข้อมูลให้เป็นเซตของความสัมพันธ์ที่มีคุณสมบัติตามต้องการ นอร์มัลไลเซชันไม่มีผลทำให้สารสนเทศต่างๆ หายไป หรือทำให้เกิดสารสนเทศใหม่ๆ ที่ไม่จริงขึ้นมา โมเดลข้อมูลที่ผ่านขบวนการนอร์มัลไลเซชันแล้วนั้นจะเป็นโมเดลข้อมูลเชิงตรรกที่ดี ซึ่งทำให้การออกแบบฐานข้อมูลเป็นไปได้อย่างง่าย ถูกต้อง คงที่ ไม่ซ้ำซ้อน และมีเสถียรภาพ

ประโยชน์ของนอร์มัลไลเซชันมีดังต่อไปนี้

ก. ลดเนื้อที่ที่ใช้เก็บข้อมูล

ข. ลดการเกิดความไม่คงที่ภายในฐานข้อมูล

ค. ลดความเป็นไปได้ในการปรับปรุง และลบข้อมูลที่ผิด

ง. เพิ่มเสถียรภาพของโครงสร้างข้อมูล

ขั้นตอนการนอร์มัลไลเซชันประกอบไปด้วยขั้นตอนต่างๆ ดังต่อไปนี้

7.1 นอร์มัลไลขั้นที่ 1 (First Normal Form หรือ 1NF) จะจัดการรวบรวมข้อมูลใหม่ไม่ให้มีข้อมูลที่ซ้ำๆ กัน (Repeating Group) โดยพิจารณาเอนติตีที่มีแอตตริบิวต์ที่มีค่าหลายค่า (Multivalued Attributes) ให้มาเป็นเอนติตีลูกของเอนติตีเดิม

ลูกจ้าง

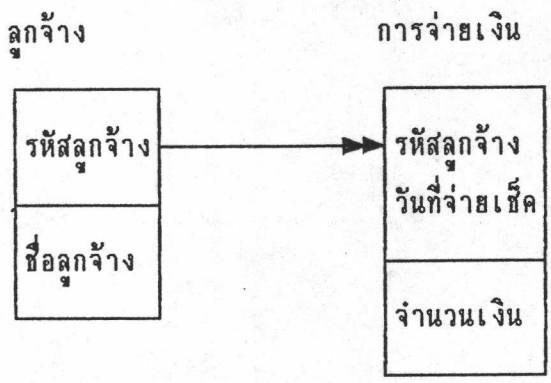
รหัสลูกจ้าง	ชื่อลูกจ้าง	วันที่จ่ายเช็ค	จำนวนเงิน	วันที่จ่ายเช็ค	จำนวนเงิน	วันที่จ่ายเช็ค	จำนวนเงิน
23312	สมศักดิ์	02/09/33	2000	09/09/33	2000	16/09/33	2500
12462	สุขุม	02/09/33	1650	09/09/33	1680	16/09/33	1930
18164	สมศรี	02/09/33	3500	09/09/33	3500	16/09/33	3500

ตารางที่ 3.4 แสดงข้อมูลของเอนติตีลูกจ้าง

ตัวอย่างตารางที่ 3.4 เป็นเอนติตีที่ไม่อยู่ใน 1NF คือเอนติตีลูกจ้างซึ่งประกอบไปด้วยวันที่ และจำนวนเงินที่จ่ายเช็ค หลายวัน และหลายจำนวน

จะเห็นได้ว่าแอตตริบิวต์ที่มีค่าหลายค่าได้แก่ วันที่จ่ายเช็ค และจำนวนเงิน

ดังนั้นเพื่อให้เอนติตีลูกจ้างอยู่ใน 1NF โดยการกำจัดแอตตริบิวต์ที่มีค่าหลายค่า และสร้างเอนติตีใหม่ขึ้นมาชื่อว่า เอนติตีการจ่ายเงิน โดยจะมีแผนภาพดังรูปที่ 3.8 โดยมีตัวอย่างข้อมูลดังตารางที่ 3.5



รูปที่ 3.8 แสดงเอนติตีที่อยู่ใน 1NF

ลูกจ้าง

รหัสลูกจ้าง	ชื่อลูกจ้าง
23312	สมศักดิ์
12462	สุขุม
18164	สมศรี

การจ่ายเงิน

รหัสลูกจ้าง	วันที่จ่ายเช็ค	จำนวนเงิน
23312	02/09/33	2000
23312	09/09/33	2000
23312	16/09/33	2500
12462	02/09/33	1650
12462	09/09/33	1680
12462	16/09/33	1930
18164	02/09/33	3500
18164	09/09/33	3500
18164	16/09/33	3500

ตารางที่ 3.5 แสดงข้อมูลของเอนติตีตามรูปที่ 3.8

จุดประสงค์ของการออกแบบโมเดลข้อมูลเชิงตรรกโดยการทำนอร์มัลไลซ์ชั้นที่1 นี้มีเหตุผลสำคัญ 3 ประการคือ

ก. ทำให้โมเดลข้อมูลเชิงตรรกง่ายขึ้น (ทำให้โครงสร้างของข้อมูลง่ายขึ้น)

ข. ทำให้สามารถใช้เทคนิคของนอร์มัลไลเซชันในขั้นต่อไปได้

ค. ทำให้มีความสามารถในการแปลง โมเดลข้อมูลเชิงตรรกไปเป็นโครง

สร้างของฐานข้อมูลแบบใดๆ ก็ได้โดยไม่สูญเสียหน้าที่ต่างๆ ของโครงสร้างของฐานข้อมูลเหล่านั้น

7.2 นอร์มัลไลซ์ชั้นที่2 (Second Normal Form หรือ 2NF) ทุกๆ แอตตริบิวต์ที่อยู่ภายในเอนติตีเดียวกันต้องขึ้นอยู่กับคีย์หลักเท่านั้น และเอนติตีนี้ต้องอยู่ใน 1NF ด้วย ดังนั้นถ้าต้องการให้เอนติตีต่างๆ อยู่ใน 2NF จะต้องนำเอาแอตตริบิวต์ที่ไม่ขึ้นกับส่วนของคีย์หลักทั้งหมดออก(นั่นคือนำแอตตริบิวต์ที่ขึ้นอยู่กับส่วนใดส่วนหนึ่งของคีย์หลักออก) พิจารณาตารางที่ 3.6 แสดงข้อมูลของเอนติตีรายการสั่งซื้อสินค้า ซึ่งมีเลขที่ใบสั่งซื้อ และครั้งที่เป็นคีย์หลัก

รายการสั่งซื้อสินค้า

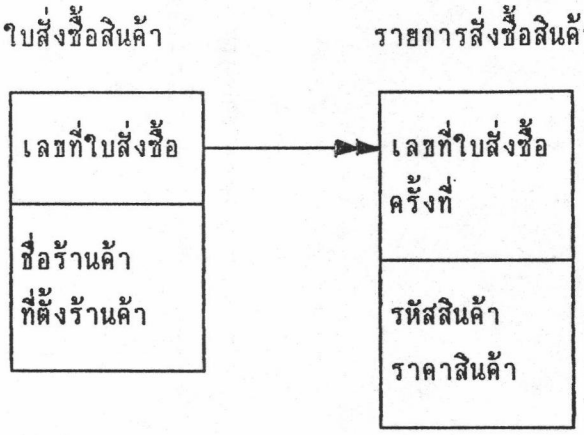
เลขที่ใบสั่งซื้อ	ครั้งที่	ชื่อร้านค้า	ที่ตั้งร้านค้า	รหัสสินค้า	ราคาสินค้า
124	1	รุ่งเรือง	สามย่าน	145	50
124	2	รุ่งเรือง	สามย่าน	83	63
138	1	สินไท	สยาม	9762	119
138	2	สินไท	สยาม	19	253
138	3	สินไท	สยาม	81	18
259	1	พาณิชย์	สุขุมวิท	781	30
259	2	พาณิชย์	สุขุมวิท	119	45

ตารางที่ 3.6 แสดงให้เห็นว่าเอนติตีรายการสั่งซื้อสินค้าไม่อยู่ใน 2NF

จากตารางข้างต้นนี้จะเห็นว่าทุกๆ ชื่อร้านค้า และที่ตั้งร้านค้าที่เหมือนกันจะขึ้นอยู่กับเลขที่ใบสั่งซื้อ และไม่ขึ้นอยู่กับคีย์หลัก (แต่จะขึ้นอยู่กับส่วนใดส่วนหนึ่งของคีย์หลัก) ดังนั้นให้นำเอาแอตตริบิวต์เหล่านี้ออกไป (อยู่ในเอนติตีใหม่หรือเอนติตีเดิมที่มีอยู่แล้ว) ซึ่งในตัวอย่างนี้จะเป็นดังรูปที่ 3.9 และมีข้อมูลดังตารางที่ 3.7

7.3 นอร์มัลไลซ์ชั้นที่3 (Third Normal Form 3NF) ให้กำจัดแอตตริบิวต์ที่ไม่เพียงขึ้นกับคีย์หลักเท่านั้นแต่ยังขึ้นกับแอตตริบิวต์อื่นที่ไม่เป็นคีย์ ดังนั้นถ้าต้องการให้เอนติตีต่างๆ อยู่ใน 3NF จะต้องเป็นเอนติตีที่อยู่ใน 2NF และนำแอตตริบิวต์อื่นที่ขึ้นอยู่กับแอตตริบิวต์อื่นที่ไม่เป็นคีย์ออก

จากตารางใบสั่งซื้อสินค้า จะเห็นได้ว่า ที่ตั้งร้านค้าจะขึ้นอยู่กับชื่อร้านค้า ถึงแม้ว่าที่ตั้งร้านค้าจะขึ้นอยู่กับเลขที่ใบสั่งซื้อก็ตาม ดังนั้นตามวิธีการของ 3NF จึงนำเอาที่ตั้งร้านค้าออกมาอยู่ในเอนติตีร้านค้าดังรูปที่ 3.10



รูปที่ 3.9 แสดงให้เห็นว่าเอนติตีอยู่ใน 2NF

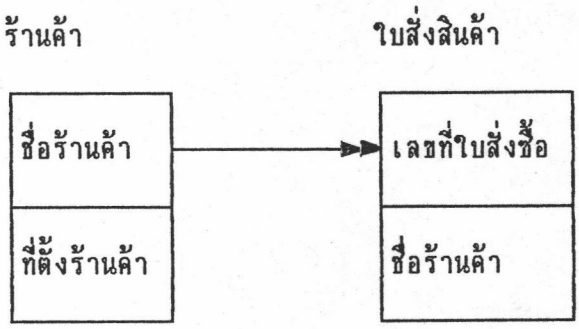
ใบสั่งซื้อสินค้า

เลขที่ใบสั่งซื้อ	ชื่อร้านค้า	ที่ตั้งร้านค้า
124	รุ่งเรือง	สามย่าน
138	สินไท	สยาม
259	พาณิชย์	สุขุมวิท

รายการสั่งซื้อสินค้า

เลขที่ใบสั่งซื้อ	ครั้งที่	รหัสสินค้า	ราคาสินค้า
124	1	145	50
124	2	83	63
138	1	9762	119
138	2	19	253
138	3	81	18
259	1	19	30
259	2	81	45

ตารางที่ 3.7 แสดงข้อมูลตามรูปที่ 3.9



รูปที่ 3.10 แสดงเอนติตีที่อยู่ใน 3NF

7.4 บอยซ์/คอดด์นอร์มัลไล (Boyce/codd Normal Form BCNF) เอนติตีใดๆ จะอยู่ใน BCNF ก็ต่อเมื่อทุกแอตตริบิวต์ในเอนติตีนั้นถูกกำหนดโดยทุกส่วนของแคนดิเดทคีย์ (คีย์หลัก และคีย์รอง) ไม่ใช่ขึ้นอยู่กับส่วนใดส่วนหนึ่งของแคนดิเดทคีย์

7.5 นอร์มัลไลขั้นที่4 (Fourth Normal form 4NF)

7.6 นอร์มัลไลขั้นที่5 (Fifth Normal form 5NF)

8. กำหนดโดเมน

โดเมน หมายถึง เซตของค่าต่างๆ ที่เป็นไปได้ของแอตตริบิวต์ใดแอตตริบิวต์หนึ่ง การกำหนดโดเมนให้กับแต่ละแอตตริบิวต์มีวิธีการดังต่อไปนี้

8.1 กำหนดโดเมน หรือเซตของคุณสมบัติของโดเมนให้กับแต่ละแอตตริบิวต์โดยที่คุณสมบัติของโดเมนที่ควรกำหนดได้แก่

8.1.1 ชนิดของข้อมูล (Data Type) กำหนดว่าเป็นข้อมูลชนิดใดเช่น จำนวนเต็ม (Integer) ทศนิยม (Decimal) หรือ อักขระ(Character)

8.1.2 ความยาว (Length)

8.1.3 รูปแบบ (Format)

8.1.4 พิสัย (Range)

8.1.5 ความหมาย (Meaning)

8.1.6 ค่าที่ยอมได้ (Allowable Values)

8.1.7 ความเป็นหนึ่ง (Uniqueness)

8.1.8 มีค่าเป็นนัลได้หรือไม่

8.1.9 มีค่าโดยปริยายหรือไม่

คุณสมบัติบางประการของโดเมน จะเป็นอิสระจากเอนติตีที่แอตตริบิวต์นั้นปรากฏอยู่คุณสมบัติเหล่านี้ได้แก่ ชนิดของข้อมูล และความยาว ควรจะเหมือนกันตลอดในโมเดลข้อมูลเชิงตรรก แต่ก็มีคุณสมบัติบางประการของโดเมนที่ขึ้นอยู่กับเอนติตีที่แอตตริบิวต์เหล่านั้นปรากฏอยู่ได้แก่คุณสมบัติความเป็นหนึ่งเช่นเมื่อแอตตริบิวต์หนึ่งอยู่ในเอนติตีที่มีแอตตริบิวต์นี้เป็นคีย์หลักดังนั้น

จึงต้องมีคุณสมบัติความเป็นหนึ่ง แต่เมื่อไปปรากฏอยู่ที่เอนติตีอื่นแอตตริบิวต์นี้ไม่จำเป็นต้องมีคุณสมบัติความเป็นหนึ่งก็ได้

8.2 บันทึกโดเมน หรือเซตของคุณสมบัติต่างๆ ของโดเมนของแต่ละแอตตริบิวต์ลงในพจนานุกรมข้อมูล

8.3 กำหนดโดเมนของคีย์หลักโดยให้พิจารณาตามกฎต่างๆ ดังนี้

8.3.1 คีย์หลักต้องมีค่าเป็นหนึ่ง

8.3.2 แอตตริบิวต์ที่เป็นส่วนประกอบของคีย์หลักไม่ต้องมีค่าความเป็นหนึ่ง

8.3.3 คีย์หลัก หรือส่วนประกอบของคีย์หลักมีค่าเป็นนัลไม่ได้

8.3.4 คีย์หลัก และส่วนประกอบของคีย์หลักต้องรับค่าโดยปริยายได้ แต่ต้องคงไว้ซึ่งความเป็นหนึ่ง

8.4 กำหนดโดเมนของคีย์รองโดยให้พิจารณาตามกฎต่างๆ ดังนี้

8.4.1 คีย์รองต้องมีค่าเป็นหนึ่ง

8.4.2 แอตตริบิวต์ที่เป็นส่วนประกอบของคีย์รองไม่ต้องมีค่าเป็นหนึ่ง

8.4.3 คีย์รอง และส่วนประกอบของคีย์รองอาจมีค่าเป็นนัลได้

8.4.4 คีย์รอง และส่วนประกอบของคีย์รองต้องรับค่าโดยปริยายได้ แต่ต้องคงไว้ซึ่งความเป็นหนึ่ง (คุณสมบัติของคีย์รองต่างจากคีย์หลัก ตรงที่ว่า คีย์รองมีค่าเป็นนัลได้)

8.5 กำหนดโดเมนของ ฟอร์เรนจ์คีย์ โดยให้พิจารณาตามกฎต่างๆ ดังนี้

8.5.1 ชนิดของข้อมูล ความยาว และรูปแบบของฟอร์เรนจ์คีย์ จะต้องเหมือนกับชนิดของข้อมูล ความยาว และรูปแบบ เมื่อเป็นคีย์หลักในเอนติตีแม่

8.5.2 คุณสมบัติความเป็นหนึ่งของฟอร์เรนจ์คีย์จะขึ้นอยู่กับชนิดของความสัมพันธ์ถ้าความสัมพันธ์เป็นแบบหนึ่งต่อหนึ่ง ฟอร์เรนจ์คีย์จะต้องมีคุณสมบัติความเป็นหนึ่งแต่ถ้าความสัมพันธ์เป็นแบบหนึ่งต่อหลาย ฟอร์เรนจ์คีย์จะไม่มีคุณสมบัติความเป็นหนึ่ง

8.6 กำหนดโดเมนของดิไรฟ์แอตตริบิวต์ โดยให้พิจารณาตามกฎต่างๆ ดังนี้

8.6.1 ให้กำหนดอัลกอริทึมของดิไรฟ์แอตตริบิวต์ ไว้ที่คุณสมบัติของโดเมนตรงค่าที่ยอมได้

8.6.2 ชนิดของข้อมูลของดิไรฟ์แอตตริบิวต์จะต้องมีชนิดเดียวกับแอตตริบิวต์ที่เป็นแหล่งที่มาของดิไรฟ์แอตตริบิวต์นี้ มิฉะนั้นอาจถูกกำหนดโดยอัลกอริทึม

8.6.3 ความหมายของดิไรฟ์แอตตริบิวต์กำหนดได้จากอัลกอริทึม และจากความหมายของแอตตริบิวต์ที่เป็นแหล่งที่มาของดิไรฟ์แอตตริบิวต์

8.7 กำหนดโดเมนของคีย์หลักของเอนติตีที่เป็นซับไทป์ โดยโดเมนนี้จะต้องเป็นซับเซตของโดเมนของคีย์หลักของเอนติตีที่เป็นซูเปอร์ไทป์ซึ่งจะมีคุณสมบัติดังต่อไปนี้

8.7.1 ชนิดของข้อมูล ความยาว และรูปแบบ จะต้องเหมือนกับคีย์หลักของเอนติตีที่เป็นซูเปอร์ไทป์

8.7.2 ค่าที่ยอมได้จะต้องขึ้นกับตัวระบุชับไทป์ ว่าในโมเดลข้อมูลเชิงตรรก มีตัวระบุชับไทป์อยู่หรือไม่

8.7.3 ความหมายจะต้องเหมือนกับคีย์หลักของซูเปอร์ไทป์ แต่ขึ้นอยู่กับตัวระบุชับไทป์

8.7.4 จะต้องมีคุณสมบัติความเป็นหนึ่ง

8.7.5 ส่วนประกอบของคีย์หลักไม่ต้องมีคุณสมบัติความเป็นหนึ่ง

8.7.6 ห้ามมีค่าเป็นนัล

8.7.7 สามารถกำหนดค่าเป็นค่าโดยปริยายได้

นอกจากกำหนดโดเมนให้กับแอตตริบิวต์ที่เป็นคีย์ที่กล่าวมาข้างต้นแล้วนั้น จะต้องกำหนดโดเมนให้กับแอตตริบิวต์ต่างๆ ที่ปรากฏอยู่ในโมเดลข้อมูลเชิงตรรก

9. กำหนดทริกเกอร์ดำเนินการ

ทริกเกอร์ดำเนินการเป็นกฎที่ควบคุมการเป็นไปได้ของการใช้คำสั่งในการแทรก ลบ ปรับปรุง และดึงข้อมูล รวมทั้งผลลัพธ์ของการทำคำสั่งเหล่านี้ที่มีผลกระทบ กับเอนติตีอื่นๆ หรือ แอตตริบิวต์อื่นๆ ในเอนติตีเดียวกัน

การกำหนดทริกเกอร์ดำเนินการนี้ จะทำให้ค่าต่างๆ ของแอตตริบิวต์มีความเป็นบูรณภาพ และมีความคงที่โดยการกำหนดทริกเกอร์ดำเนินการนี้ อาจส่งผลกระทบต่อเงื่อนไขต่างๆ ดังนี้

- เงื่อนไขที่เกี่ยวพันกับแอตตริบิวต์หลายตัวในหลายเอนติตี หรือในหลายค่าของเอนติตี
- เงื่อนไขที่เกี่ยวพันกับแอตตริบิวต์ภายในเอนติตีเดียวกันตั้งแต่ 2แอตตริบิวต์ขึ้นไป
- เงื่อนไขที่เกี่ยวพันกับแอตตริบิวต์เดียว เอนติตีเดียว หรือ พารามิเตอร์ใด พารามิเตอร์หนึ่ง

ทริกเกอร์ดำเนินการมีส่วนประกอบทางตรรกภาพที่สำคัญ 2 ส่วนคือ

ก. เหตุการณ์และเงื่อนไขที่ทำให้เกิดคำสั่งต่างๆ ขึ้น (The Trigger)

ข. การกระทำ และเงื่อนไขเมื่อเกิดเหตุการณ์ (The Operation)

ให้บันทึกทุกๆ ทริกเกอร์ดำเนินการลงในพจนานุกรมข้อมูลโดยสิ่งที่ควรบันทึกมีดังต่อไปนี้

9.1 เหตุการณ์เริ่มต้นที่ทำให้ทริกเกอร์ดำเนินการนั้นทำงาน เช่น การแทรก การลบ หรือการดึงข้อมูล

9.2 สิ่งต่างๆ ที่เกี่ยวข้องเมื่อเหตุการณ์นั้นเกิดขึ้น เช่นบันทึกชื่อของเอนติตี หรือแอตตริบิวต์ที่จะถูกเปลี่ยนแปลง หรือถูกเข้าถึง

9.3 เงื่อนไขต่างๆ ที่จะทำให้เหตุการณ์ดังกล่าวเกิดขึ้น

9.4 ผลลัพธ์ หรือ การกระทำเมื่อเหตุการณ์นั้นเกิดขึ้น

นอกจากกำหนดทริกเกอร์ดำเนินการให้แก่ แอตตริบิวต์ต่างๆ แล้วต้องกำหนดให้กับแอตตริบิวต์ที่เป็นแหล่งที่มาของดิไรฟ์แอตตริบิวต์ในกรณีที่เป็นซับไทป์ให้กำหนดทริกเกอร์ดำเนินการ เช่น เมื่อค่าต่างๆ ในเอนติตีที่เป็นซับไทป์ถูกลบ ค่าที่ตรงกับค่าที่ถูกลบในเอนติตีที่เป็นซับไทป์จะต้องถูกลบในเอนติตีที่เป็นซูเปอร์ไทป์ด้วย

10. รวบรวมมุมของผู้ใช้ทั้งหมดเข้าด้วยกัน

ขั้นตอนต่างๆ ที่ได้กล่าวมาแล้วตั้งแต่ขั้นตอนที่1 ถึงขั้นตอนที่9 นั้นเป็นการพัฒนารายละเอียดความต้องการสารสนเทศของผู้ใช้แต่ละคน กลุ่มของผู้ใช้ หรือหน้าที่ส่วนใดส่วนหนึ่งของระบบเท่านั้น ดังนั้นจะพบว่าในระบบเมื่อรวบรวมความต้องการต่างๆ จากผู้ใช้หลายคนหรือหลายมุมมองแล้วนั้น อาจทำให้เกิดความต้องการที่เลื่อมล้ำ หรือ ซ้ำซ้อนกันเกิดขึ้นก็ได้

จุดประสงค์ในการรวบรวมมุมมองของผู้ใช้

ก. เพื่อเป็นตัวแทนของมุมมองต่างๆ ได้อย่างแม่นยำ

ข. เพื่อกำจัดความซ้ำซ้อน

ค. เพื่อแก้ปัญหาการไม่คงที่ที่มีขึ้นระหว่างมุมมองต่างๆ

ง. เพื่อเพิ่มรีเลชันชิปใหม่ๆ ที่เกิดขึ้นมาระหว่างมุมมองต่างๆ

การรวบรวมมุมมองต่างๆ เข้าด้วยกันนั้นจะต้องทำตามขั้นตอนต่างๆ ดังต่อไปนี้

ก. รวบรวมเอนติตี และกฎธุรกิจต่างๆ ที่เกี่ยวข้องกัน

ข. รวบรวมรีเลชันชิป และกฎธุรกิจต่างๆ ที่เกี่ยวข้องกัน

ค. รวบรวมแอตตริบิวต์ และกฎธุรกิจต่างๆ ที่เกี่ยวข้องกัน

วิธีการในการรวบรวมมุมมองของผู้ใช้นั้นมีวิธีการดังต่อไปนี้

10.1 ขณะที่รวบรวมมุมมองของผู้ใช้นั้นให้นำเอนติตีที่มีคีย์หลักเหมือนกัน และมีโดเมนของคีย์หลักที่เทียบเท่ากันมาไว้รวมกัน โดยให้นำแอตตริบิวต์ของเอนติตีที่นำมารวมกันเอาไว้ด้วยกันโดยกำจัดแอตตริบิวต์ที่ซ้ำออก

จะต้องไม่รวมเอนติตีที่เป็นซับไทป์ และเอนติตีที่เป็ซูเปอร์ไทป์ในมุมมองของผู้ใช้เดียวกันเอาไว้ด้วยกันถึงแม้ว่าจะมีคีย์หลักเหมือนกันก็ตาม แต่ในทางกลับกันอาจจะรวมเอนติตีที่เป็นซับไทป์ และเอนติตีที่เป็นซูเปอร์ไทป์ที่อยู่ต่างๆ มุมมองของผู้ใช้เข้าไว้ด้วยกันก็ได้

10.2 ให้ตั้งรีเลชันชิประหว่างเอนติตีที่เป็นซับไทป์ และเอนติตีที่เป็นซูเปอร์ไทป์ขึ้นมาระหว่างเอนติตีที่มีคีย์หลักเหมือนกัน โดยที่โดเมนของคีย์หลักของเอนติตีที่จะมาเป็นซับไทป์นั้นเป็นซับเซตของโดเมนของคีย์หลักของเอนติตีที่จะมาเป็นซูเปอร์ไทป์ และให้กำจัดแอตตริบิวต์ในเอนติตีที่เป็นซับไทป์ออก ถ้าพบว่ามีอยู่แล้วในเอนติตีที่เป็นซูเปอร์ไทป์

10.3 ให้ตั้งเอนติตีที่เป็นซูเปอร์ไทป์ขึ้นมาเมื่อพบว่ามี 2 เอนติตีที่มีคีย์หลักเหมือนกันและโดเมนของคีย์หลักมีค่าเท่าเทียมกัน (ถึงแม้ว่าอาจมีเงื่อนไขของค่าที่ยอมได้ต่างกันก็ตาม)

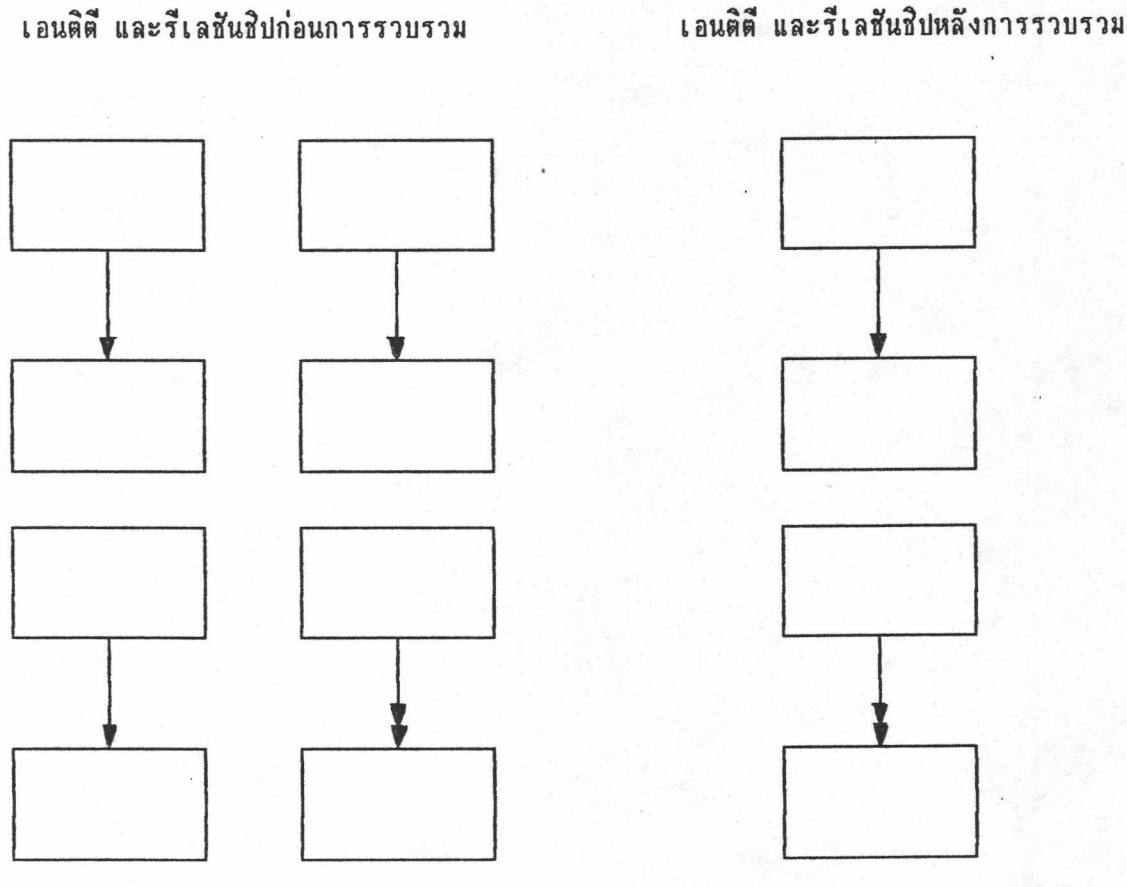
i 20372024

10.4 รวมเอนติตีที่มีคีย์หลักที่เป็นแคนดิเดทคีย์ ของอีกเอนติตีหนึ่ง เอาไว้เป็นเอนติตีเดียวกัน โดยให้เลือกเอาว่าจะเอาแอตตริบิวต์ตัวใดเป็นคีย์หลัก และในขณะเดียวกันให้กำจัด แอตตริบิวต์ที่เหมือนกันออกด้วย

10.5 เอนติตีใดๆ ที่มีคีย์หลักต่างๆ กัน ซึ่งไม่สามารถรวมกับเอนติตีอื่นได้นั้น ให้คงเอนติตีเหล่านั้นอยู่อย่างเดิมโดยไม่ต้องเปลี่ยนแปลงใดๆ ทั้งสิ้น

10.6 ให้คงกฎธุรกิจที่เกี่ยวกับแคนดิเดทคีย์เอาไว้แต่ในบางกรณีอาจยกเว้นเช่น จากเดิมเคยเป็นคีย์หลัก แต่เมื่อรวบรวมมุมมองของผู้ใช้แล้วได้กลายมาเป็นคีย์รอง ดังนั้น แอตตริบิวต์นี้สามารถมีค่าเป็นนัลได้ เป็นต้น

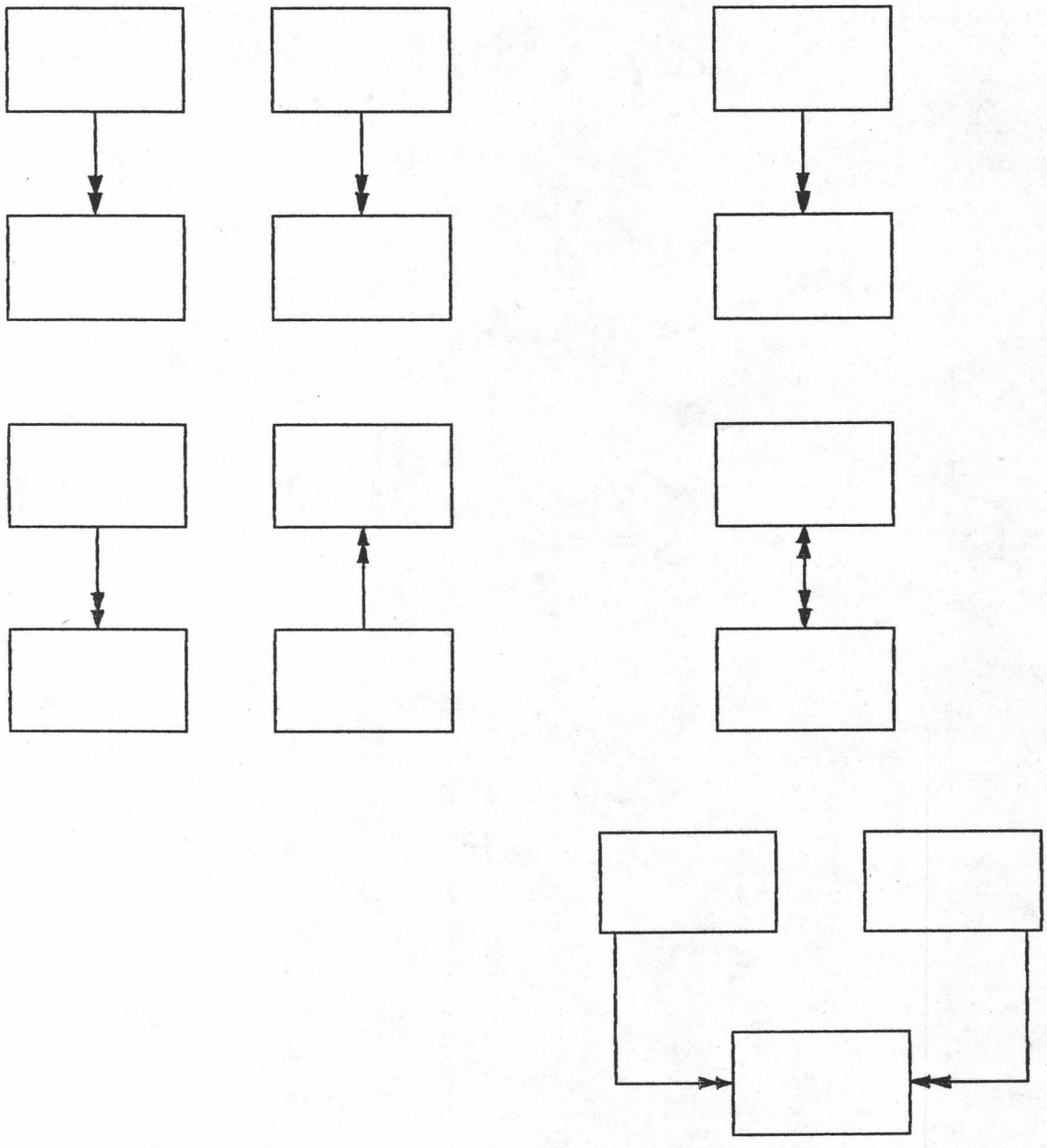
10.7 เมื่อรวบรวมเอนติตีต่างๆ เข้าด้วยกันนั้นรีเลชันชิปก็จะถูกรวบรวมเข้าด้วยกันถ้ารีเลชันชิประหว่างเอนติตีเหล่านั้นยังคงมีความหมายเหมือนเดิม แต่สัดส่วนความสัมพันธ์อาจเปลี่ยนไปจากเดิม โดยพิจารณาจากรูปที่ 3.11 และรูปที่ 3.12



รูปที่ 3.11 แสดงเอนติตี และรีเลชันชิป ก่อน และหลังการรวบรวม

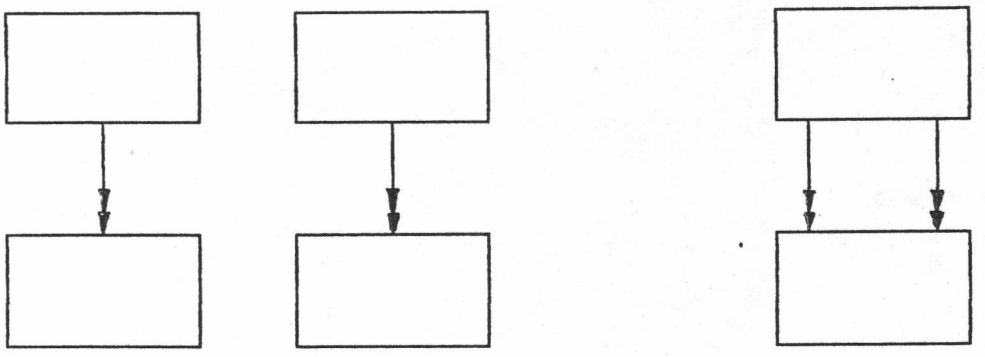
เอนติตี และรีเลชันชิปก่อนการรวบรวม

เอนติตี และรีเลชันชิปหลังการรวบรวม



รูปที่ 3.12 แสดงเอนติตี และรีเลชันชิป ก่อน และหลังการรวบรวม (ต่อ)

แต่ถ้าความสัมพันธ์ในแต่ละมุมมองของผู้ใช้ไม่เหมือนกันก็จะไม่รวมรีเลชันชิปเข้าด้วยกันพิจารณารูปที่ 3.13



รูปที่ 3.13 แสดงเอนติตี และรีเลชันชิป ก่อนและหลังการรวบรวมโดยที่มุมมองของผู้ใช้ไม่เหมือนกัน

10.8 ให้คงรีเลชันชิปที่มีความหมายแตกต่างๆ กันไว้อย่างเดิม และให้กำจัดรีเลชันชิปที่ซ้ำซ้อนออกไป

10.9 ให้พิจารณาว่ามีรีเลชันชิปใดขาดหายไปบ้าง ถ้าพบว่ามีก็ให้เพิ่มรีเลชันชิปขึ้นมา

10.10 พิจารณา ฟอร์เรนจ์คีย์ของมุมมองของผู้ใช้ที่รวบรวมแล้วให้ถูกต้อง

10.11 ให้รวมเอาคีย์ของกฎธุรกิจที่กำหนดไว้ในมุมมองของผู้ใช้เดิม ที่ยังไม่ได้ถูกรวบรวมเอาไว้ด้วย และเพิ่มคีย์ของกฎธุรกิจสำหรับรีเลชันชิปที่เพิ่มขึ้นมาใหม่ด้วย

10.12 รวมแอตตริบิวต์ที่มีความหมายเดียวกันที่อยู่ที่เอนติตีเดียวกันเข้าไว้ด้วยกัน โดยให้มีโดเมน และทริกเกอร์ดำเนินการที่เข้ากันได้

10.13 ให้พิจารณาแอตตริบิวต์ต่างๆ ในมุมมองของผู้ใช้ที่รวบรวมแล้วว่าเป็นดิไรฟว์แอตตริบิวต์ หรือไม่ ในบางครั้งจะพบว่าเมื่อรวบรวมมุมมองของผู้ใช้แล้วแอตตริบิวต์ใดๆ จาก มุมมองของผู้ใช้อาจหาได้จาก แอตตริบิวต์อื่นใน มุมมองของผู้ใช้อีกมุมมองหนึ่ง

10.14 หลังจากที่รวบรวมเอนติตีเข้าด้วยกันแล้ว กำจัดหรือเพิ่มรีเลชันชิปให้เหมาะสมแล้วนั้น ให้ทำการนอร์มัลไลเซชันอีกครั้งหนึ่งเพื่อขจัดความซ้ำซ้อน

10.15 ให้พิจารณาแอตตริบิวต์ต่างๆ อีกครั้งหนึ่งโดยพิจารณาที่โดเมนว่า สามารถมีค่าเป็นนัล ได้หรือไม่เพื่อที่พิจารณาให้กำหนดเอนติตีที่เป็นซับไทป์ขึ้นมาเพื่อลดการเป็นนัลนั้น

11. รวบรวมโมเดลข้อมูลที่มีอยู่แล้ว

ขั้นตอนนี้เป็นการรวบรวมโมเดลข้อมูลเชิงตรรกอันใหม่กับของเดิมที่มีอยู่แล้ว ก็จะเหมือนกับเป็นการรวบรวมมุมมองของผู้ใช้ 2 คนเข้าด้วยกัน แต่ถ้าในระบบไม่มีโมเดลข้อมูลเชิงตรรกอยู่ อาจจะพบว่าเกิดความไม่คงที่ขึ้น เช่นชื่อของข้อมูล และความหมายของแอตตริบิวต์ในเอนติตีเดียวกัน ดังนั้นจุดประสงค์ที่ต้องการรวบรวมนั้นไม่จำเป็นที่จะต้องจัดการกับความแตกต่างที่มีอยู่ให้เข้ากันได้ เพียงแต่เข้าใจ และบันทึกความสัมพันธ์ รวมทั้งความไม่คงที่ต่างๆ ที่เกิดขึ้นระหว่าง โมเดลข้อมูลเชิงตรรก กับ ระบบเดิม

12. พิจารณาเสถียรภาพ และการเติบโต

ขั้นตอนต่างๆ ที่ได้กล่าวมาแล้วพิจารณาแค่ความต้องการของระบบในปัจจุบันเท่านั้น แต่ตามความเป็นจริงแล้วควรพิจารณาสิ่งที่ควรจะเกิดขึ้นต่อไปด้วย เช่นจะมีการเปลี่ยนแปลงในโมเดลของข้อมูลเชิงตรรกอย่างไรถ้าในระบบมีการดำเนินการต่างๆ เปลี่ยนไป หรือมุมมองต่างๆ ของผู้ใช้เปลี่ยนไป เช่น

- จะมีเอนติตี หรือรีเลชันชิปใหม่เพิ่มขึ้นแล้วไปกระทบต่อเอนติตีเดิมที่มีอยู่โดยการเพิ่ม ฟอร์เรนจ์คีย์ใหม่ลงไปหรือไม่
- จะมีสัดส่วนของความสัมพันธ์เปลี่ยนแปลงไปหรือไม่ เช่นจากเดิมความสัมพันธ์แบบ 1:M มาเป็นความสัมพันธ์แบบ M:N
- จะมีคีย์หลักเปลี่ยนไปหรือไม่ เนื่องจากต้องการให้มีความเป็นหนึ่งซึ่งจะมีผลกระทบต่อฟอร์เรนจ์คีย์ในเอนติตีที่สัมพันธ์กัน
- จะมีเอนติตีที่เป็นซับไทป์เพิ่มขึ้นหรือไม่
- จะมีแอตตริบิวต์ใหม่เพิ่มขึ้น หรือแอตตริบิวต์เดิมที่มีอยู่แล้วจะถูกกำจัดไปหรือไม่
- จะมีกฎธุรกิจเปลี่ยนแปลงไปหรือไม่ เช่น กฎการเพิ่ม กฎการลบ โดเมน หรือ ทริกเกอร์ดำเนินการ
- ขนาดของเอนติตี(ปริมาณข้อมูลในเอนติตี) เพิ่มขึ้น หรือ ลดลงหรือไม่

เสถียรภาพ และการเติบโตของโมเดลข้อมูลเชิงตรรกขึ้นอยู่กับความสามารถในการคาดคะเนถึงการเปลี่ยนแปลงความต้องการที่จะมีขึ้นมาใหม่