บทที่ 3

ทฤษฎีที่สำคัญ

3.1 ระบบฐานข้อมูล

วัตถุประสงค์หนึ่งของการพัฒนาซอฟต์แวร์ต้นแบบเพื่อควบคุมพัสดุต่างประเทศสำหรับงานก่อสร้างโรงไฟฟ้า ก็เพื่อจัดเก็บข้อมูลที่เกี่ยวกับพัสดุต่างประเทศที่ใช้ในการก่อสร้างโรงไฟฟ้าของทุกหน่วยงานที่เกี่ยวข้องให้เป็นมาตรฐานเดียวกัน เพื่อให้สามารถควบคุมความถูกต้องและลดความซ้ำซ้อนของข้อมูลได้ เพื่อบรรลุถึงเป้าหมายดังกล่าว จึงใช้หลักการออกแบบฐานข้อมูลซึ่งเน้นการจัดการกับข้อมูล (Data Oriented) มาใช้แทนการออกแบบแฟ้มข้อมูล ซึ่งเน้นวิธีประมวลผลข้อมูล (Process Oriented) ดังจะได้กล่าวถึงขั้นตอนการสร้างฐานข้อมูลต่อไป

การจัดเก็บข้อมูลอย่างมีระเบียบแบบแผน และวิธีการนำข้อมูลส่วนที่ต้องการออกมาใช้ให้ทันต่อเหตุการณ์ จะยังประโยชน์แก่ผู้นำมาใช้ได้อย่างสูง ไม่ว่าจะเป็นการนำมาใช้ในการวางแผนหรือตัดสินใจด้านงบประมาณ การลงทุนและอื่น ๆ ความยากลำบากในการจัดเก็บข้อมูลอย่างมีระเบียบแบบแผน และวิธีการนำข้อมูลส่วนที่ต้องการออกมาใช้ให้ทันต่อเหตุการณ์ เมื่อนำระบบคอมพิวเตอร์เข้ามาใช้ในการประมวลผลข้อมูล ทำให้ระบบการจัดเก็บข้อมูลเป็นไปอย่างสะดวกขึ้น แต่การควบคุมความปลอดภัยถูกต้องของข้อมูล ตลอดจนการเรียกใช้ข้อมูล ยังอยู่ภายใต้ความสามารถของนักเขียนโปรแกรม สิ่งที่ตามมาคือต้องใช้เจ้าหน้าที่และเวลาในการสร้างโปรแกรม เพื่อประยุกต์ใช้งานตามความต้องการเป็นจำนวนมาก และนี่คือจุดกำเนิดของฐานข้อมูล เพื่อทำหน้าที่ควบคุมและจัดการความถูกต้องและประสิทธิภาพของการบริหารข้อมูล

การเก็บข้อมูลจำนวนมากไว้ แล้วสามารถนำออกมาใช้ได้อย่างมีประสิทธิภาพ ต้องอาศัยฐานข้อมูลที่ดี แต่ทว่าขั้นตอนในการสร้างข้อมูลที่ดีนั้นต้องอาศัยเทคนิคต่าง ๆ รวมถึงประสบการณ์ด้วย การออกแบบฐานข้อมูลเป็นเรื่องยากและซับซ้อน เพราะต้องออกแบบระบบข้อมูลที่ครอบคลุมทั้งหน่วยงาน แต่ก็ต้องสามารถป้องกันให้มีแต่ผู้ที่มีหน้าที่รับผิดชอบ ให้เรียกใช้เฉพาะข้อมูลส่วนของตนได้เท่านั้น จะอ่านและเรียกใช้ข้อมูลของผู้อื่นไม่ได้ ทั้ง ๆ ที่ข้อมูลเหล่านี้อยู่ในฐานข้อมูลเดียวกัน งานส่วนที่ยุ่งยากที่สุดของการออกแบบก็คือ การทำความเข้าใจข้อมูลที่ผู้ใช้ต้องการใช้ในงานต่าง ๆ และการแทนข้อมูลเหล่านี้ในฐานข้อมูล

3.1.1 ระบบฐานข้อมูล (Database System)

ระบบฐานข้อมูล หมายถึงการรวบรวมข้อมูลที่มีความสัมพันธ์กัน และอาจอยู่ในต่างที่กันให้เสมือนอยู่ร่วมกัน โดยกำหนดรูปแบบที่แน่นอนเพื่อให้สามารถรับใช้งานที่มีวัตถุประสงค์

แตกต่างกันของหน่วยงานต่าง ๆ โดยที่ผู้ใช้ฐานข้อมูลไม่ได้รับรู้ข้อมูลทั้งหมดในฐานข้อมูล แต่รับรู้เฉพาะส่วนที่เกี่ยวข้องกับงานของตนเท่านั้น ฉะนั้นผู้ใช้แต่ละคนจะรับรู้ข้อมูลที่แตกต่างกันจากฐานข้อมูลเดียวกัน เช่น ผู้ใช้ที่อยู่ฝ่ายการเงิน จะรู้ชื่อและเงินเดือนของพนักงาน ในขณะที่ผู้บริหารรับรู้ข้อมูลเกี่ยวกับผลงานของพนักงาน ทำนองเดียวกับคนตาบอดคลำช้าง คนที่คลำถูกหางช้าง คิดว่าช้างมีลักษณะเหมือนเชือก ในขณะที่คนที่คลำถูกลำตัว คิดว่าช้างเหมือนฝาผนัง เป็นต้น ฉะนั้นผู้ออกแบบช้าง (ฐานข้อมูล) จะต้องแน่ใจว่า สิ่งที่ออกแบบนั้นสามารถตอบสนองทัศนะที่แตกต่างกันของผู้ใช้ได้

3.1.2 แบบจำลองข้อมูล (Data Modeling)

การกำหนดแบบจำลองข้อมูล คือ การอธิบายโครงสร้างของฐานข้อมูล ได้แก่ ชนิดข้อมูล (Data Type), ความสัมพันธ์ (Relation) และข้อกำหนดของข้อมูลนั้น แบ่งเป็น 3 ระดับ คือ

3.1.2.1 **ระดับบน (High Level)** หรือ แบบจำลองเชิงแนวคิด (Conceptual Data Model) เป็นโครงสร้างข้อมูลในระดับที่ผู้ใช้งานมองเห็น หรือเกี่ยวข้องกับผู้ใช้งานโดยตรง ไม่เกี่ยวกับวิธีการจัดเก็บข้อมูล

3.1.2.2 **ระดับระเบียน (Record Base)** หรือ แบบจำลองข้อมูลเชิงการใช้งาน (Implementation Data Model) เป็นโครงสร้างในระดับการใช้งาน โดยผู้ใช้สามารถเข้าใจได้แต่ไม่สัมพันธ์กับการจัดเก็บข้อมูล

3.1.2.3 **ระดับล่าง (Low Level)** หรือ แบบจำลองข้อมูลเชิงกายภาพ (Physical Data Model) เป็นโครงสร้างข้อมูลที่ถูกเก็บภายในสื่อต่าง ๆ ของคอมพิวเตอร์

3.2 เหตุผลในการพัฒนาระบบฐานข้อมูล

ข้อมูลจำนวนมากมายที่ต้องการรวบรวมไว้ที่เดียวกันนั้น สมควรที่จะพัฒนาเป็นฐานข้อมูลเป็นอย่างยิ่ง เนื่องจากมีข้อดีที่เห็นได้ชัดหลายประการดังนี้

3.2.1 สามารถลดความซ้ำซ้อนของข้อมูล เนื่องจากการเก็บข้อมูลในรูปของฐานข้อมูลจะรวมเอาข้อมูลมาเก็บไว้ที่เดียวกัน ทำให้สามารถลดปัญหาความซ้ำซ้อนของข้อมูลได้ แต่ในบางกรณีก็จำเป็นจะต้องมีความซ้ำซ้อนของข้อมูลในฐานข้อมูลบ้าง อาจด้วยเหตุผลที่เกี่ยวกับประสิทธิภาพ (Performance) หรืออะไรก็ตาม ในกรณีเช่นนี้ผู้บริหารฐานข้อมูลจะต้องบอกให้ระบบจัดการฐานข้อมูลรับรู้เพื่อที่จะตรวจสอบและแก้ไขข้อมูลที่ซ้ำซ้อนกันนี้ให้สอดคล้องกันเสมอ

3.2.2 สามารถแก้ปัญหาการที่ข้อมูลไม่สอดคล้องกัน คือ เมื่อปัญหาความซ้ำซ้อนลดลง หรือหากมีแต่ระบบจัดการฐานข้อมูลได้รับรู้ ปัญหาที่ว่าข้อมูลไม่สอดคล้องกันก็จะไม่เกิดขึ้น

3.2.3 สามารถควบคุมความถูกต้องของข้อมูล (Integrity) นอกจากความสอดคล้องกันของข้อมูลในกรณีที่มีความซ้ำซ้อนแล้ว ความถูกต้องของข้อมูลที่จะต้องคำนึงถึงมีมากมาย หากในระบบฐานข้อมูลนั้นใช้ระบบจัดการฐานข้อมูลที่มีความสามารถ ผู้บริหารฐานข้อมูลก็จะสามารถกำหนดกฎเกณฑ์ให้ระบบจัดการฐานข้อมูล เพื่อใช้ในการตรวจสอบและแก้ไขข้อมูลให้ถูกต้องได้ เช่น อาจจะกำหนดได้ว่าอายุของพนักงานในบริษัทจะมีต่ำกว่า 18 ปี และ จะไม่เกิน 60 ปี ดังนั้นหากมีการป้อนข้อมูลที่แสดงอายุเป็น 70 ปี ระบบจัดการฐานข้อมูลก็จะไม่ยอม หรืออาจกำหนดได้ว่า ชื่อของพนักงานที่จะปรากฏอยู่ในฐานข้อมูล จะต้องเป็นชื่อที่ปรากฏในตารางแสดงประวัติพนักงานเท่านั้น ซึ่งเป็นตารางที่เก็บประวัติพนักงานที่บริษัทว่าจ้างอยู่ในปัจจุบัน (ตัวอย่างนี้เป็นกรณีของฐานข้อมูลเชิงสัมพันธ์) หรืออาจกำหนดได้ว่า หากพนักงานคนใดถูกลบออกจากตารางแสดงประวัติ ก็ให้ลบส่วนอื่น ๆ ที่เกี่ยวข้องกับพนักงานคนนั้นออกไปจากฐานข้อมูลด้วย เป็นต้น

3.2.4 สามารถสร้างหรือกำหนดระบบความปลอดภัยได้ จากตัวอย่างที่แสดง จะเห็นได้ว่าผู้บริหารฐานข้อมูลเป็นผู้ที่สามารถกำหนดได้ว่า จะให้ผู้ใช้คนใดเห็นข้อมูลในส่วนใดได้บ้าง โดยการสร้างวิวให้ผู้ใช้ เฉพาะในส่วนที่เขาจะมีสิทธิรับรู้เท่านั้น นอกจากนี้หากระบบจัดการฐานข้อมูลที่ใช้เป็นตัวที่มีความสามารถมากพอ ก็จะสามารถกำหนดละเอียดลงไปได้อีกว่า สำหรับผู้ใช้ต่าง ๆ ที่ใช้วิวเดียวกันนี้ ผู้ใช้คนใดบ้างที่จะมีสิทธิในการแก้ไขข้อมูล หรืออาจจะมีสิทธิได้แค่เรียกดูข้อมูลได้เพียงอย่างเดียว และในระบบจัดการฐานข้อมูลที่มีระบบความปลอดภัยที่ดี จะให้ผู้ใช้หรือโปรแกรมเข้าถึงข้อมูลได้ ก็ต่อเมื่อติดต่อผ่านระบบจัดการฐานข้อมูลเท่านั้น คือเมื่อผู้ใช้ป้อนข้อมูลเข้ามา ระบบจัดการฐานข้อมูลก็จะต้องทำการเข้ารหัสข้อมูลก่อนที่จะนำไปเก็บภายในฐานข้อมูล เมื่อมีการเรียกใช้ข้อมูล จึงทำการถอดรหัสแล้วค่อยนำเสนอผู้ใช้งาน

3.2.5 เกิดความอิสระของข้อมูล คือเมื่อมีการเปลี่ยนแปลงโครงสร้างของข้อมูลในระดับล่าง จะต้องไม่มีผลกระทบต่อโครงสร้างข้อมูลในระดับบน ซึ่งความอิสระของข้อมูลนี้ รวมไปถึงในกรณีที่สามารถกำหนดได้ว่า ข้อมูลที่ปรากฏในระดับต่าง ๆ กัน สามารถมีรูปแบบที่ต่างกันได้ตามความต้องการ เช่น ในระดับหลักการ เงินเดือนของพนักงานอาจอยู่ในรูปของเลขฐานสอง แต่เมื่อสร้างวิวให้กับผู้ใช้คนหนึ่ง เงินเดือนที่ปรากฏอาจจะอยู่ในรูปของเลขฐานสิบ และในวิวของผู้ใช้อีกคน อาจเห็นเงินเดือนในรูปของเลขฐานสอง ซึ่งการแปลงข้อมูลนี้ก็จะเป็นหน้าที่ของระบบจัดการฐานข้อมูล โดยมีผู้บริหารฐานข้อมูลเป็นผู้กำหนด

3.2.6 ข้อมูลสามารถใช้ร่วมกันได้ ระบบงานทุกระบบที่มีความสัมพันธ์กันจะใช้ฐานข้อมูลชุดเดียวกันได้ รวมไปถึงการที่ต้องพัฒนาระบบใหม่ขึ้นมา ซึ่งทำให้พัฒนาเร็วยิ่งขึ้น

3.2.7 ระบบงานมีมาตราฐานมากขึ้น เมื่อมีฐานข้อมูลชุดเดียวกัน ระบบงานก็จะถูกควบคุมไปโดยอัตโนมัติด้วยลักษณะของการใช้ซอฟท์แวร์ในการพัฒนาระบบ และการจัดเก็บข้อมูลในสื่อ

3.2.8 ลดความขัดแย้งของระบบงาน เนื่องจากการออกแบบข้อมูลเป็นไปในแนวเดียวกันทั้งองค์กร ด้วยการศึกษาจากความต้องการของทุกส่วนงานในการออกแบบเบื้องต้น จึงทำให้ระบบงานขององค์กรเดียวกันมีความสัมพันธ์ไปในแนวเดียวกัน

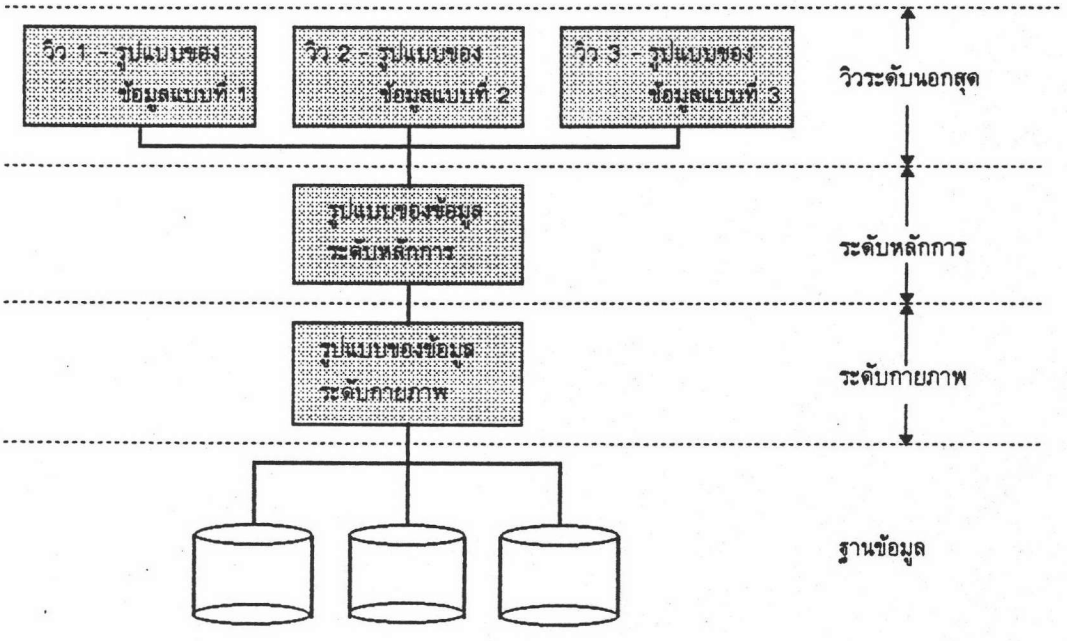
3.3 สถาปัตยกรรมของระบบฐานข้อมูล (Database Architecture)

การใช้ฐานข้อมูลจะช่วยให้ผู้ใช้งานข้อมูลได้สะดวก และไม่ต้องคำนึงถึงลักษณะการจัดเก็บ และการจัดการ เนื่องจากระบบฐานข้อมูลได้เก็บรายละเอียด วิธีการจัดเก็บ ตลอดจนการจัดการกับฐานข้อมูล นอกจากนี้ยังสามารถกำหนดขอบเขตการใช้ฐานข้อมูล ตามความสำคัญของงาน และผู้ใช้งานได้ ลักษณะรูปแบบของข้อมูลที่สอดคล้องกับแบบจำลองที่สร้างขึ้นนั้น จะเรียกว่าสคีมา (Schema) แบ่งเป็น 3 ระดับ คือ

3.3.1 **ระดับกายภาพ (Internal or Physical Level)** เป็นระดับที่ต่ำที่สุด ได้แก่ ระดับการจัดเก็บข้อมูลในสื่อบันทึกข้อมูล ถูกกำหนดโดยระบบบริหารฐานข้อมูล

3.3.2 **ระดับหลักการ (Conceptual Level or Schema)** เป็นระดับที่อยู่ตรงกลางระหว่างระดับในสุดและระดับนอกสุด เป็นการกำหนดรูปแบบของฐานข้อมูล อันได้แก่ ชนิด ลักษณะ และความสัมพันธ์เชิงตรรกของทั้งหมดของฐานข้อมูล ภาพที่เห็นในระดับนี้เรียกว่าภาพระดับหลักการ (Conceptual View) นอกจากนี้ยังกำหนดขอบเขตของข้อมูลที่ผู้ใช้ในระดับนอกสุดควรจะเห็น ภาพระดับหลักการนี้กำหนดโดยผู้บริหารฐานข้อมูล

3.3.3 **ระดับนอกสุด (External Level or Subshema)** เป็นระดับที่ใกล้กับผู้ใช้มากที่สุด คือในระดับนี้ภาพของฐานข้อมูลที่ผู้ใช้เห็นจะเป็นเพียงส่วนที่เกี่ยวข้องกับผู้ใช้เท่านั้น ภาพที่เห็นนี้บางครั้งก็เรียกว่า **ภาพนอกสุด (External View)** ผู้ใช้จะสามารถเรียกค้นหรือแก้ไข (หากทำได้) ก็เฉพาะในส่วนที่มองเห็นนี้เท่านั้น ถ้าภาพระดับหลักการถือเป็นภาพรวมของฐานข้อมูล ภาพนอกสุดก็จัดเป็นภาพย่อยของฐานข้อมูล เช่น ถ้าแผนที่ถนนหนทางทั้งหมดในกรุงเทพเป็นภาพรวมแล้ว การวาดเฉพาะเส้นทางระหว่างบ้านกับที่ทำงานก็คือภาพย่อย



รูปที่ 3.1 แสดงสถาปัตยกรรมของระบบฐานข้อมูล

3.4 ส่วนประกอบของระบบฐานข้อมูล มี 4 ประการ คือ

3.4.1 ข้อมูล

3.4.2 ฮาร์ดแวร์ (Hardware)

3.4.3 ซอฟต์แวร์ (Software) โดยมีระบบจัดการฐานข้อมูล (Database Management System : DBMS) เป็นซอฟต์แวร์ที่ทำหน้าที่จัดการกับฐานข้อมูลตามที่ผู้ใช้ต้องการ

3.4.4 ผู้ใช้ข้อมูล (User) แบ่งออกเป็น

3.4.4.1 โปรแกรมเมอร์งานประยุกต์ (Application Programmer)

3.4.4.2 ผู้ใช้งานจริง (End User)

3.4.4.3 ผู้บริหารฐานข้อมูล (Database Administrator : DBA) มีหน้าที่คอยดูแลและจัดวางโครงสร้างต่าง ๆ เพื่อที่จะให้ระบบฐานข้อมูลทำงานได้

3.5 การเลือกระบบจัดการฐานข้อมูล

3.5.1 ระบบการจัดการฐานข้อมูล (Database Management System) หรือเรียกย่อ ๆ ว่า DBMS นั้น ประกอบด้วย การรวมข้อมูลเข้าด้วยกัน และชุดคำสั่งที่จะมาดำเนินการกับข้อมูลนั้น ซึ่งระบบการจัดการฐานข้อมูล จะอำนวยความสะดวกประสิทธิภาพในการเรียกใช้และการจัดเก็บข้อมูลลงในฐานข้อมูล หน้าที่ของระบบจัดการฐานข้อมูลคือ

3.5.1.1 กำหนดลักษณะขององค์ประกอบของฐานข้อมูล ได้แก่การกำหนดองค์ประกอบของระเบียน การกำหนดชนิดของระเบียน รวมถึงการกำหนดความสัมพันธ์ต่าง ๆ

3.5.1.2 การกำหนดโครงสร้างการจัดเก็บฐานข้อมูล ได้แก่ การกำหนดโครงสร้างของสื่อบันทึกข้อมูล วิธีในการเข้าถึงข้อมูล การดึงข้อมูลมาใช้งาน การแก้ไข และการสำรองระบบข้อมูล

3.5.1.3 กำหนดการเข้าถึงข้อมูลในฐานข้อมูลด้วยภาษาทางคอมพิวเตอร์

3.5.1.4 จะต้องมีระบบการควบคุมได้ เช่น การสำรองข้อมูล การกู้ระบบเมื่อมีความเสียหายเกิดขึ้น รวมทั้งการควบคุมภาวะการดำเนินการที่พร้อมกันของระบบ

3.5.2 การเลือกพิจารณาระบบจัดการฐานข้อมูล

การเลือกนี้จะต้องคำนึงถึงทางด้านเทคนิคและด้านเศรษฐกิจ ตลอดจนถึงโครงสร้างในการเก็บข้อมูล ทิศทางการเข้าถึงข้อมูลที่ฐานข้อมูลชนิดนั้นจะทำได้ ภาษาระดับสูงที่ใช้ได้

3.5.2.1 การพิจารณาทางด้านเทคนิค โดยขอกล่าวจากฐานข้อมูลแต่ละประเภทดังนี้

3.5.2.1.1 แบบเชิงลำดับชั้น (Hierarchical Model)

เป็นฐานข้อมูลในรุ่นแรกจะเก็บความสัมพันธ์ของข้อมูลเป็นแบบต้นไม้ (Tree) โดยมีหลักการว่า ข้อมูลจากแม่ (Parent) ชี้ไปยังข้อมูลในลูก (Child) ผู้ออกแบบจะต้องกำหนดแม่และลูก ตั้งแต่การสร้างว่าข้อมูลอะไรเกี่ยวข้องกัน และมีข้อจำกัดว่าใน 1 ลูก จะเกี่ยวข้องกับ 1 แม่เท่านั้น ฐานข้อมูลแบบนี้ ได้แก่ IMS/VS

ข้อดี : ความสัมพันธ์เป็นแบบง่ายๆ ทำให้เข้าถึงข้อมูลได้ง่าย

ข้อเสีย : ถ้าเป็นงานที่มีความสัมพันธ์กันเป็นจำนวนมาก (M:N) จะไม่เหมาะสมกับโครงสร้างแบบต้นไม้ และถ้าต้องการเปลี่ยนความสัมพันธ์ของข้อมูลภายหลังจากใช้งานไปแล้วทำได้ยาก

3.5.2.1.2 แบบข่ายงาน (Network Model)

ข้อเสียของแบบเชิงลำดับชั้น ได้ถูกแก้ด้วยแบบข่ายงาน คือ 1 ลูก จะเกี่ยวข้องกับแม่มากกว่า 1 ได้ แต่ก็ยังต้องกำหนดความสัมพันธ์ตั้งแต่เริ่มสร้างฐานข้อมูล ไม่สามารถเปลี่ยนแปลงในระหว่างการทำงานได้ ฐานข้อมูลแบบนี้ได้แก่ DBTG

ข้อดี : เหมาะสมกับระบบที่มีความสัมพันธ์ของข้อมูลเป็นจำนวนมาก (M:N)

ข้อเสีย : ค่อนข้างซับซ้อน โปรแกรมเมอร์ที่เขียนคำสั่งเพื่อดึงข้อมูลที่ต้องการจะต้องเข้าใจและทราบตำแหน่งของข้อมูล จึงยุ่งยากกว่าแบบอื่น

3.5.2.1.3 แบบเชิงสัมพันธ์ (Relational Model)

เกิดขึ้นในช่วงหลัง เป็นฐานข้อมูลที่ได้รับการพัฒนาโดยที่ความสัมพันธ์ของข้อมูล สามารถกำหนดขึ้นมาเมื่อใดก็ได้ที่ผู้ใช้งานต้องการ มีลักษณะคล้ายคลึงกับการเก็บเป็นระเบียน และ เขตข้อมูล โดยเรียกระเบียนว่าทูเปิล (Tuple) และเรียกเขตข้อมูลว่า แอททริบิวต์ (Attribute) ทำให้เข้าใจง่าย ฐานข้อมูลแบบนี้ได้แก่ System R, DB2, SQL/DS, Ingres, Oracle

ข้อดี : การออกแบบ และการเปลี่ยนแปลงความสัมพันธ์ทำได้ง่าย และอาจประยุกต์เป็นแบบเชิงลำดับชั้นหรือข่ายงานก็ได้

ข้อเสีย : ถ้าข้อมูลมีความสัมพันธ์กันเป็นจำนวนมากจะทำให้การค้นหาข้อมูลช้ากว่าแบบอื่น

3.5.2.2 การพิจารณาทางด้านเศรษฐกิจ เป็นการพิจารณาถึงค่าใช้จ่ายที่เกี่ยวข้องเมื่อเลือกระบบจัดการฐานข้อมูลชนิดนั้น ค่าใช้จ่ายต่าง ๆ ได้แก่

3.5.2.2.1 ราคาของซอฟต์แวร์ การเลือกต้องพิจารณาหน้าจอ วิธีของการดึงข้อมูล การสำรองข้อมูล ประกอบการพิจารณาราคา

3.5.2.2.2 ค่าบำรุงรักษาระบบ

3.5.2.2.3 ราคาของฮาร์ดแวร์ ต้องพิจารณาร่วมกับประเภทของเทอร์มินอล หน่วยความจำหลัก หน่วยความจำสำรองที่จะได้มา

3.5.2.2.4 ค่าใช้จ่ายในการสร้างระบบใหม่ และการเปลี่ยนจากระบบเก่าเป็นระบบใหม่

3.5.2.2.5 ค่าจ้างบุคลากร รวมถึงค่าฝึกหัดต่าง ๆ

3.5.3 ภาษาฐานข้อมูล

โดยทั่วไปแล้วระบบจัดการฐานข้อมูลซึ่งเป็นสิ่งที่คั่นกลางระหว่างฐานข้อมูลและผู้ใช้ จะมีภาษาสำหรับสร้างฐานข้อมูลสำหรับให้ผู้ใช้เรียกค้นข้อมูล และจัดการด้านต่าง ๆ ในทางทฤษฎีแล้วภาษาฐานข้อมูลนี้จะประกอบด้วย 2 ส่วน คือ

3.5.3.1 ส่วนนิยามข้อมูล (Data Definition Language : DDL)

เป็นส่วนที่ใช้ในการสร้างฐานข้อมูลทั้ง 3 ระดับ หรือหากมองเป็น

ระบบฐานข้อมูลเชิงสัมพันธ์ ภาษานิยามข้อมูลนี้จะใช้สำหรับสร้างตารางหลัก (Base Table) วิว (View) และบอกถึงลักษณะการเก็บข้อมูลจริง นอกจากนี้ภาษานิยามข้อมูลยังกำหนดกฎเกณฑ์เพื่อให้ระบบจัดการฐานข้อมูลใช้สำหรับตรวจสอบความถูกต้องของข้อมูลอีกด้วย

3.5.3.2 ส่วนจัดการข้อมูล (Data Manipulation Language : DML)

เป็นส่วนที่ใช้ในการค้นหาข้อมูล (Retrieve) เพิ่มข้อมูล (Insert) แก้ไขข้อมูล (Update) และลบข้อมูล (Delete) ดังนั้นการใช้ฐานข้อมูลในระดับของผู้ใช้นั้น ก็สามารถกระทำได้โดยการใช้ส่วนนี้นั่นเอง โดยทั่วไปแล้วแบ่งได้เป็น 2 ประเภทคือ

3.5.3.2.1 แบบต้องกำหนดวิธีการ (Procedural Language) แบบนี้เป็นหน้าที่ของผู้ใช้ที่จะต้องระบุสิ่งที่ต้องการ และวิธีการให้ได้มาซึ่งผลลัพท์อันนั้นด้วย

3.5.3.2.2 แบบไม่ต้องกำหนดวิธีการ (Nonprocedural Language) แบบนี้ผู้ใช้งานเพียงแต่ระบุสิ่งที่ต้องการ โดยไม่ต้องบอกว่าทำอย่างไร

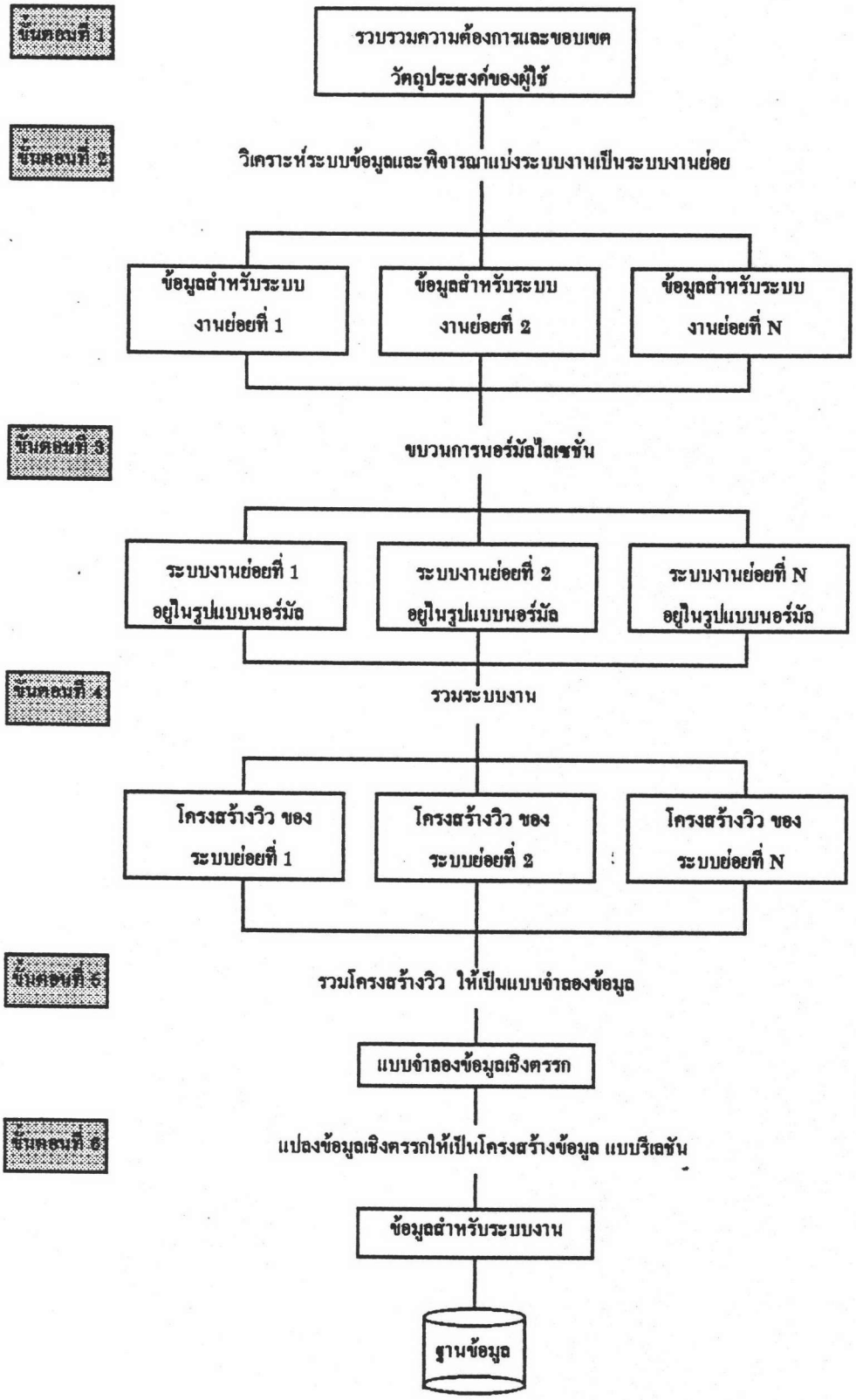
โดยทั่วไปแล้วภาษาเอสคิวแอล (Structured Query Language หรือคำย่อว่า SQL) ซึ่งเป็นภาษาที่ใช้ในการติดต่อกับระบบจัดการฐานข้อมูล นอกจากนี้จะประกอบด้วยส่วนจัดการและนิยามข้อมูลแล้วยังมีส่วนควบคุมข้อมูล (Data Control Language หรือคำย่อว่า DCL) เพื่อใช้ควบคุมและป้องกันการเข้าถึงข้อมูลที่เป็นส่วนตัวด้วย

3.6 ขั้นตอนการออกแบบฐานข้อมูล

การออกแบบฐานข้อมูลของระบบงานใดระบบหนึ่งเพื่อใช้งาน ประกอบด้วยหลักการออกแบบใหญ่ ๆ 2 หลัก คือ

3.6.1 การออกแบบฐานข้อมูลแบบตรรก (Logical Database Design)

เป็นการรวบรวมความต้องการของผู้ใช้ ข้อมูลทั้งหมดและนำมาออกแบบโดยคำนึงถึงความสัมพันธ์ข้อมูลเป็นหลัก ไม่สนใจถึงอุปกรณ์ฮาร์ดแวร์ โปรแกรมระบบ หรือ โปรแกรมการจัดการฐานข้อมูล ดังนั้นการออกแบบฐานข้อมูลแบบตรรกจะเป็นอิสระจากอุปกรณ์ (Hardware Independent) ผลที่ได้จากการออกแบบจะเป็นแบบจำลองข้อมูลเชิงตรรก (Logical Data Modeling หรือคำย่อว่า LDM) ของระบบงาน ซึ่งแสดงขั้นตอนได้ดังนี้



รูปที่ 3.2 ขั้นตอนการออกแบบฐานข้อมูล

จากรูปที่ 3.2 แสดงให้ทราบถึงขั้นตอนการออกแบบฐานข้อมูล โดยการออกแบบฐานข้อมูลแบบตรรก จะครอบคลุมขั้นตอนที่ 1 ถึง ขั้นตอนที่ 6

3.6.1.1 ขั้นตอนที่ 1 เป็นการรวบรวมความต้องการ (Requirement) ของผู้ใช้ซึ่งความต้องการเหล่านี้จะรวมถึงความต้องการในปัจจุบันของระบบงาน และความต้องการในอนาคตของระบบงานที่มีโอกาสเกิดขึ้น ผู้รวบรวมข้อมูล (Data Administrator Modeler) จะรวบรวมข้อมูลปัจจุบันและข้อมูลอนาคตที่คาดคะเนไว้ เช่น ข้อมูลที่เจ้าหน้าที่ตรวจสอบคอมพิวเตอร์ (Computer Auditor) จำเป็นต้องใช้ในการตรวจสอบการทำงานของโปรแกรมระบบงานต่าง ๆ ฯลฯ ซึ่งข้อมูลที่รวบรวมจะอยู่ในลักษณะที่แบ่งย่อยไม่ได้ (Atomic Data)

3.6.1.2 ขั้นตอนที่ 2 เป็นการรวบรวมคุณสมบัติข้อมูลของฐานข้อมูล และทำการวิเคราะห์ระบบข้อมูล โดยหลังจากวิเคราะห์แล้ว อาจแบ่งระบบงานเป็นระบบงานย่อย คุณสมบัติของข้อมูลที่รวบรวมจะประกอบด้วย

3.6.1.2.1 ความหมายของข้อมูล (Data Meaning)

3.6.1.2.2 ค่าความจริงของข้อมูล (Data Facts)

ในขั้นตอนนี้จะพบจำนวนของข้อมูลมีมากสำหรับระบบงาน เพื่อขจัดปัญหาในการที่จะบริหารข้อมูลที่มีมากเกินไปที่จะพบในขั้นตอนต่อไป จึงมีการแบ่งระบบงานออกเป็นระบบงานย่อย โดยรวมข้อมูลของงานประยุกต์ (Application) ที่ใช้เข้าอยู่ในระบบงานย่อยเดียวกัน แต่ก็มีข้อมูลบางส่วนที่ใช้ร่วมกัน สำหรับระบบงานย่อยหลาย ๆ ระบบงานย่อยซึ่งจะมีการขจัดในขั้นตอนรวมแบบจำลองระบบงานย่อยเป็นแบบจำลองระบบงานภายหลัง ในการวิเคราะห์ข้อมูลและแสดงคุณสมบัติข้อมูล จะใช้วิธีเทคนิคแบบจำลองข้อมูล (Data Modeling Technique) ผลจากการวิเคราะห์จะได้โมเดลของระบบงานย่อยของแต่ละระบบงานย่อยอย่างคร่าว ๆ เท่านั้น

3.6.1.3 ขั้นตอนที่ 3 เป็นการพิจารณาข้อมูลในแต่ละระบบงานย่อยเพื่อให้ได้เอนทิตีและความสัมพันธ์ระหว่างเอนทิตี มีการนำขบวนการนอร์มัลไลเซชันมาลดความซ้ำซ้อนข้อมูล และจัดข้อมูลให้อยู่ในรูปแบบนอร์มัล

3.6.1.4 ขั้นตอนที่ 4 เป็นการสร้างแบบจำลองข้อมูลสำหรับแต่ละระบบงานย่อยซึ่งแบบจำลองข้อมูลนั้นเกิดจากการนำเอนทิตีของแต่ละระบบงานย่อยมาเชื่อมโยงเข้าด้วยกัน

3.6.1.5 ขั้นตอนที่ 5 เป็นการนำแบบจำลองข้อมูลของแต่ละระบบงานย่อยมารวมกันเป็นแบบจำลองข้อมูลเชิงตรรกของระบบงานต้องการ

3.6.1.5.1 รวมความสัมพันธ์ หรือ ยุบความสัมพันธ์ในหลาย ๆ ความสัมพันธ์ที่ใช้แสดงเอนทิตีให้เป็นหนึ่งความสัมพันธ์

3.6.1.5.2 เชื่อมเอนทิตีในแต่ละระบบงานย่อย ให้เป็นแบบจำลองข้อมูลเชิงตรรกของระบบงาน

3.6.1.6 ขั้นตอนที่ 6 เป็นการนำแบบจำลองข้อมูลเชิงตรรกของระบบงานรวม มาดัดแปลงเพื่อให้เหมาะสมกับระบบการจัดการฐานข้อมูลที่จะนำมาทำงานร่วมกัน ผลจากขั้นตอนนี้จะได้แบบจำลองข้อมูลเชิงกายภาพแบบรีเลชันนัล

3.6.2 การออกแบบฐานข้อมูลแบบกายภาพ (Physical Database Design)

การออกแบบฐานข้อมูลแบบกายภาพอาศัยรายละเอียด คือแบบจำลองข้อมูลเชิงตรรก ซึ่งเกิดจากหลักการแรกเป็นพื้นฐานในการดำเนินงานต่อไป โดยจะมีขบวนการแปล (Translation Process) จากแบบจำลองข้อมูลเชิงตรรกเป็นแบบจำลองข้อมูลเชิงกายภาพ

รูป 3.2 แสดงให้ทราบถึงขั้นตอนการออกแบบฐานข้อมูล โดยการออกแบบฐานข้อมูลแบบกายภาพ จะครอบคลุมขั้นตอนที่ 6 ซึ่งในบทนี้จะกล่าวถึงการออกแบบฐานข้อมูลกายภาพแบบรีเลชันนัล

การออกแบบฐานข้อมูลโดยพิจารณาจากทิศทางการออกแบบ สามารถแบ่งออกได้ 2 วิธี คือ

3.6.2.1 การออกแบบจากระดับบนไปสู่ระดับล่าง (Top-Down Design)

เป็นการวิเคราะห์ข้อมูลของทั้งองค์กร โดยพิจารณาความหมายของข้อมูลในระดับบน ไม่เน้นการประยุกต์ใช้งาน ทำให้ง่ายต่อการรวมแบบจำลองข้อมูลแบบเบ็ดเสร็จ เรียกลักษณะนี้ว่าเป็นการทำแบบจำลองของสารสนเทศ (Information Modeling)

3.6.2.2 การออกแบบจากระดับล่างไปสู่ระดับบน (Bottom-Up Design)

เป็นการวิเคราะห์ข้อมูลของระบบงาน โดยวิเคราะห์ความต้องการของผู้ใช้แต่ละคนหรือกลุ่ม และจำลองวิวของผู้ใช้แยกกัน จากนั้นจึงนำมารวมกันเป็นฐานข้อมูลของทั้งระบบ ซึ่งวิธีนี้ใช้บ่อยในองค์กรที่ระบบงานเดิมทีการประยุกต์ใช้งานอยู่แล้ว ทำให้แบบจำลองข้อมูลที่ได้มีขอบเขตชัดเจน และถูกต้องกับความเป็นจริงมาก แต่ประสบปัญหาเกี่ยวกับจุดเชื่อมโยงระหว่างวิวของผู้ใช้ เพื่อให้เป็นแบบจำลองข้อมูลแบบเบ็ดเสร็จ ทำให้โมเดลข้อมูลที่ได้ไม่สามารถแสดงกิจกรรมทางธุรกิจทั้งหมดขององค์กรอย่างสมบูรณ์ เรียกลักษณะนี้ว่าเป็นการรวบรวมวิว (View Integration)

การออกแบบฐานข้อมูลทางกายภาพ มีสิ่งที่ต้องคำนึงถึงดังนี้

1. เวลาตอบสนอง (Response Time) คือช่วงเวลาตั้งแต่การส่งงานเข้าไปถึงเมื่อได้รับผลลัพท์ที่ต้องการออกมา

2. การใช้ที่ว่าง (Space Utilization) จำนวนที่ว่างของหน่วยความจำที่ถูกใช้โดยแฟ้มข้อมูลของฐานข้อมูลและโครงสร้างทิศทางการเข้าถึงข้อมูล

3. งานที่ได้ออกมา (Transaction Throughput) จะคิดเป็นค่าเฉลี่ย คำนวณจากจำนวนงานที่สามารถประมวลผลได้โดยระบบจัดการฐานข้อมูลต่อหนึ่งหน่วยเวลา

3.7 การควบคุมความปลอดภัย (Security Control)

การควบคุมความปลอดภัย หมายถึงการป้องกันไม่ให้ทรัพยากรในระบบคอมพิวเตอร์ รวมทั้งข้อมูลเกิดความเสียหายขึ้นอาจเกิดจากเจตนาหรือไม่ก็ตาม การควบคุมความปลอดภัย เป็นการเพิ่มความมั่นใจว่าระบบมีมาตรฐาน และประสิทธิภาพสูง ด้วยการป้องกันสาเหตุที่ก่อให้เกิดความเสียหายกับอุปกรณ์คอมพิวเตอร์ ซอฟต์แวร์ต่าง ๆ ประกอบด้วยเทคนิค 2 เทคนิคดังนี้

3.7.1 เทคนิคความปลอดภัยทางกายภาพ (Physical Security Techniques)

เป็นการพิจารณาถึงความปลอดภัยของอุปกรณ์ ที่ตั้งอุปกรณ์ ฯลฯ เพื่อไม่ให้ความเสียหายเกิดขึ้น ด้วยการคำนึงถึง

3.7.1.1 การควบคุมการเข้าถึงอุปกรณ์ต่าง ๆ

เป็นพื้นฐานของระบบความปลอดภัย มีหลักการว่าความเสียหายจะมีโอกาสเกิดขึ้นน้อยมาก เมื่อมีการตรวจสอบอย่างเข้มงวดในการเข้าถึงอุปกรณ์ เช่น การอนุญาตเฉพาะบุคลากรที่ทำงานเกี่ยวข้องกับคอมพิวเตอร์เท่านั้นสามารถเข้า-ออกหน่วยงานคอมพิวเตอร์ได้ ด้วยการตรวจสอบบัตรอนุญาตเข้า-ออก เป็นต้น

3.7.1.2 สถานที่ตั้ง

ตำแหน่งที่ตั้งของหน่วยงานคอมพิวเตอร์ มีความสำคัญในการวางแผนความปลอดภัย เช่น พิจารณา

3.7.1.2.1 สถานที่ตั้งระบบคอมพิวเตอร์ แยกเป็นอิสระจากสถานที่ทำงานของหน่วยงานอื่น ๆ

3.7.1.2.1 เส้นทางและที่ตั้งของสายไฟ สายสื่อสารต่าง ๆ ที่ใช้

3.7.1.2.3 สถานที่เก็บสื่อกลางอุปกรณ์ที่ใช้ในการสำรองข้อมูล

3.7.1.3 การป้องกันอุปกรณ์ (Physical Protection)

เป็นการพิจารณาถึงการป้องกันอุปกรณ์อื่น ๆ นอกจากที่กล่าวมาแล้ว เช่น ระบบดับเพลิงต่าง ๆ ในกรณีเกิดอุบัติเหตุไฟไหม้ ระบบไฟสำรองในกรณีไฟดับ หรือมาตรการต่าง ๆ ที่เกิดจากภัยพิบัติทางธรรมชาติ

3.7.2 เทคนิคดำเนินการความปลอดภัย (Procedural Security Technique)

เป็นการพิจารณาในลักษณะการใช้งานประยุกต์ต่าง ๆ โดยคำนึงถึง

3.7.2.1 การกำหนดรหัสผ่านแก่ผู้มีสิทธิ์ และเข้าถึงข้อมูลได้เฉพาะส่วนที่มีสิทธิ์เท่านั้น

3.7.2.2 การควบคุมการเข้าถึงรายละเอียดของเอนทิตี รีเลชันชิป และแอททริบิวต์ ซึ่งในส่วนนี้ภาษารุ่นที่ 4 จะมีคำสั่งสร้างวิว (View) เป็นส่วนช่วยให้การเข้าถึงระดับ

ของคอลัมน์และแถวของตารางได้ อีกทั้งกำหนดการทำงานต่าง ๆ เช่น สามารถเลือกดู (Select) แก้ไข (Update) เพิ่ม (Insert) ภายในตารางนั้นได้

เนื่องจากงานควบคุมพัสดุต่างประเทศเพื่องานก่อสร้างโรงไฟฟ้า เป็นลักษณะงานที่ข้อมูลมีความสัมพันธ์กับหลายหน่วยงาน อาจมีการเปลี่ยนแปลงความสัมพันธ์ของข้อมูลเพื่อให้การใช้งานเป็นไปตามความต้องการของผู้ใช้งานในภายหลัง ดังนั้นเพื่อให้การแก้ไข ปรับปรุง และบำรุงรักษาข้อมูลเป็นไปโดยง่าย สามารถพัฒนาโปรแกรมได้โดยไม่มีผลกระทบต่อข้อมูล ผู้ใช้ทั่วไปที่ไม่ใช่โปรแกรมเมอร์ สามารถเรียกดูข้อมูลได้รวดเร็ว เพราะมีภาษาระดับง่ายสำหรับผู้ใช้โดยเฉพาะ อีกทั้งการใช้ข้อมูลก็ไม่ได้ต้องการผลตอบสนองในทันที จากเหตุผลดังกล่าวจึงเลือกออกแบบและพัฒนางานบนฐานข้อมูลเชิงสัมพันธ์

ขั้นตอนการออกแบบระบบฐานข้อมูล สำหรับควบคุมพัสดุต่างประเทศเพื่องานก่อสร้างโรงไฟฟ้า ที่สำคัญมี 2 ประการ คือ

1. แบบจำลองข้อมูลเชิงตรรก (Logical Data Modeling : LDM)
2. ฐานข้อมูลแบบเชิงสัมพันธ์ (Relational Database Design : RDD)

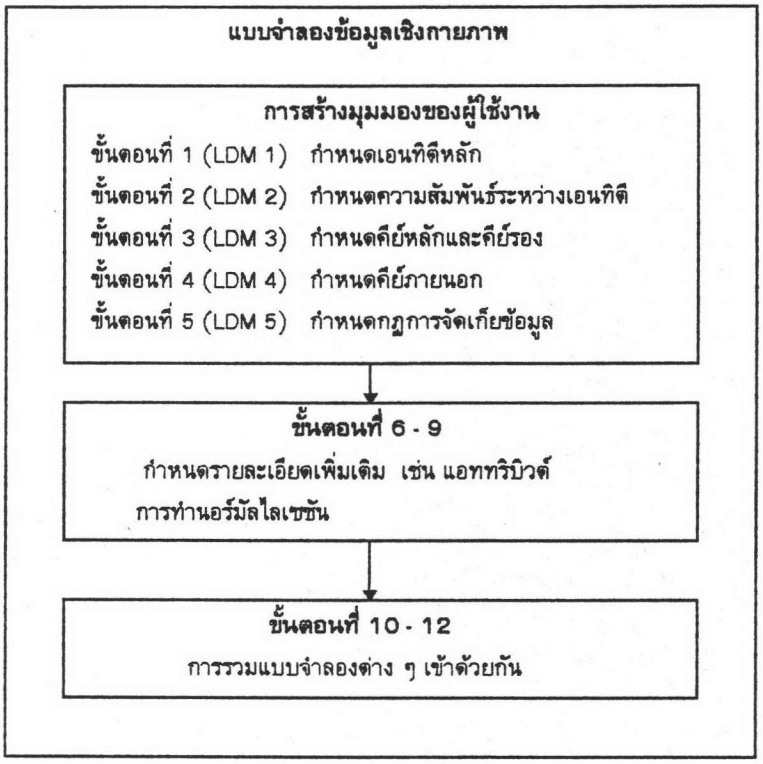
1. แบบจำลองข้อมูลเชิงตรรก (Logical Data Modeling)

เป็นโมเดลที่ใช้ในขั้นการออกแบบฐานข้อมูลเชิงมโนภาพ โดยแทนโครงสร้างและลักษณะของข้อมูลที่ต้องการ โดยเป็นลักษณะข้อมูลที่มีอยู่จริง ให้อยู่ในรูปแบบที่พร้อมที่จะสร้างเป็นฐานข้อมูลแบบอื่น ๆ ต่อไป แบบจำลองข้อมูลเชิงตรรกนี้จะทำให้ได้ฐานข้อมูลที่ถูกต้อง ไม่ขัดแย้งกัน สามารถใช้ข้อมูลร่วมกันได้ และสามารถปรับปรุงแก้ไขข้อมูลได้ง่าย การออกแบบในขั้นนี้จะต้องมีการกำหนดในสิ่งที่เป็นพื้นฐานได้แก่

เอนทิตี (Entity) คือ สิ่งที่มีอยู่จริงทางกายภาพหรือเป็นจินตภาพที่มีความหมายเป็นอย่างเดียว เมื่อกล่าวแล้วทุกคนเข้าใจเป็นที่ตรงกัน เช่น บริษัทประกันภัย ฝ่ายจัดหา ฝ่ายพัสดุ พนักงาน โครงการก่อสร้าง เป็นต้น

ความสัมพันธ์ (Relationship) หมายถึง ความสัมพันธ์ระหว่างเอนทิตีหนึ่งกับเอนทิตีอื่น โดยอาจมีความสัมพันธ์กันมากกว่า 1 เอนทิตีได้ เช่น โครงการก่อสร้างกับสัญญามีความสัมพันธ์กันโดยที่โครงการก่อสร้างโรงไฟฟ้า 1 โครงการ สามารถมีสัญญาการสั่งซื้อพัสดุมากกว่า 1 สัญญาได้ เป็นต้น

แอททริบิวต์ (Attribute) คือ คุณลักษณะที่เป็นรายละเอียด หรือกลุ่มความจริงที่เกี่ยวข้องเพื่ออธิบายเอนทิตี เช่น บริษัทจะประกอบด้วยชื่อบริษัท ที่ตั้ง เป็นต้น



รูปที่ 3.3 แสดงขั้นตอนการสร้างแบบจำลองข้อมูลเชิงตรรก

ขั้นตอนในการสร้างแบบจำลองข้อมูลเชิงตรรก มีดังต่อไปนี้

1.1 LDM 1 : กำหนดเอนทิตีหลัก

เอนทิตี เปรียบเสมือนกับเป็นคำนามอันได้แก่ บุคคล สถานที่ หรือ สิ่งของ ซึ่งอาจเป็นสิ่งที่มีตัวตน เช่น โรงไฟฟ้า บริษัทประกันภัย หรือเป็นนามธรรม เช่น อัตราแลกเปลี่ยนเงินตรา เป็นต้น ขั้นตอนนี้เป็นการกำหนดเอนทิตีหลัก ผู้ออกแบบจะต้องศึกษาระบบให้เข้าใจ เพื่อสามารถกำหนดเอนทิตีที่เกี่ยวข้องกันในระบบหนึ่งได้

1.2 LDM 2 : กำหนดความสัมพันธ์ระหว่างเอนทิตี

ความสัมพันธ์ คือ ลักษณะของการเกี่ยวพันระหว่างเอนทิตี โดยส่วนความสัมพันธ์สามารถกำหนดโดยประกอบด้วยทิศทางความสัมพันธ์ (Direction) และ สัดส่วนความสัมพันธ์ (Cardinality Ratio) ของสมาชิกในเอนทิตีแม่ และเอนทิตีลูก ความสัมพันธ์มีอยู่ 3 แบบคือ

1.2.1 ความสัมพันธ์ที่มีอยู่จริง (Existence Relationship) เช่น ที่ผู้เข้าประกวดราคาโครงการก่อสร้างโรงไฟฟ้า ดังนั้นความสัมพันธ์ระหว่างเอนทิตีผู้เข้าประกวดราคาและเอนทิตีโครงการก่อสร้างโรงไฟฟ้า คือ การเสนอราคา เป็นความสัมพันธ์ที่มีอยู่จริง

1.2.2 ความสัมพันธ์ตามหน้าที่ (Functional Relationship) เช่น คณะกรรมการที่ได้รับมอบหมายให้ทำหน้าที่คำนวณหาราคากลางของงานประกวดราคา ดังนั้นความสัมพันธ์ระหว่างเอนทิตีคณะกรรมการที่ได้รับมอบหมาย และเอนทิตีงานประกวดราคา คือการคำนวณหาราคากลางเป็นความสัมพันธ์ตามหน้าที่

1.2.3 ความสัมพันธ์ตามเหตุการณ์ที่เกิดขึ้น (Event Relationship) เช่น เมื่อโครงการก่อสร้างโรงไฟฟ้าใดได้รับการอนุมัติ ก็จะมีการดำเนินการจัดซื้ออุปกรณ์ ดังนั้นความสัมพันธ์ระหว่างเอนทิตีโครงการที่ได้รับอนุมัติและเอนทิตีอุปกรณ์ คือการจัดซื้อ เป็นความสัมพันธ์ตามเหตุการณ์ที่เกิดขึ้น

ทิศทางความสัมพันธ์ จะถูกกำหนดโดยใช้ลูกศรชี้จากเอนทิตีแม่ไปยังเอนทิตีลูกเสมอ โดยอาศัยสัดส่วนความสัมพันธ์ สามารถแบ่งความสัมพันธ์ได้ 3 ลักษณะดังนี้

1. ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (1 : 1 หรือ One-to-One Relationship) แต่ละค่าของเอนทิตีแม่มีความสัมพันธ์กับเอนทิตีลูกได้อย่างมากที่สุด 1 ค่า เช่น ในการประกวดราคาแต่ละครั้งจะมีผู้ชนะการประกวดราคาเพียงบริษัทเดียวเท่านั้น
หรือ การประกวดราคา <-----> ผู้ชนะการประกวดราคา

2. ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (1 : N หรือ One-to-Many Relationship) แต่ละค่าของเอนทิตีแม่มีความสัมพันธ์กับเอนทิตีลูกได้หลายค่า แต่ในทางกลับกันแต่ละค่าของเอนทิตีลูกมีความสัมพันธ์กับเอนทิตีแม่ได้เพียงค่าเดียว เช่น ผู้เข้าประกวดราคาโครงการจะมีสิทธิ์ชนะการประกวดราคาได้หลายโครงการ แต่การประกวดราคาโครงการหนึ่งจะมีผู้ชนะการประกวดราคาได้เพียงรายเดียวเท่านั้น
หรือ ผู้ชนะการประกวดราคา <---->> โครงการ

3. ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (M : N หรือ Many-to-Many Relationship) แต่ละค่าของเอนทิตีแม่มีความสัมพันธ์กับเอนทิตีลูกได้หลายค่า และในทางกลับกันแต่ละค่าของเอนทิตีลูกมีความสัมพันธ์กับเอนทิตีแม่ได้หลายค่า เช่น งานก่อสร้างโรงไฟฟ้าหนึ่งจะมีการจัดซื้ออุปกรณ์หลายประเภท และอุปกรณ์แต่ละประเภทจะถูกซื้อนำไปใช้ในหลายงานได้
หรือ งานก่อสร้างโรงไฟฟ้า <<------>> อุปกรณ์

1.3 LDM 3 : กำหนดคีย์หลักและคีย์รอง (Primary and Alternate Key)

ขั้นตอนนี้เป็นการเพิ่มข้อมูลที่เรียกว่า แอททริบิวต์ ลงในทุกเอนทิตีโดยที่แอททริบิวต์ก็คือข้อมูลที่แสดงลักษณะและคุณสมบัติของเอนทิตี เช่นแอททริบิวต์ของพนักงาน ก็อาจจะได้แก่ เลขประจำตัว ชื่อ สังกัด ฯลฯ เปรียบได้กับฟิลด์ (Field) ของแฟ้มข้อมูลนั่นเอง

คีย์หลัก จะประกอบด้วยค่าของแอททริบิวต์ 1 ตัว หรือมากกว่า 1 ตัวก็ได้ ที่สามารถใช้เป็นตัวเจาะจงบอกว่าต้องการอ้างถึงทูเปิลใด รหัสของโครงการ เป็นต้น

คีย์รอง จะมีคุณสมบัติเหมือนคีย์หลักทุกประการ แต่ไม่ได้ถูกเลือกให้เป็นคีย์หลัก

ทุกเอนทิตีจะต้องมีคีย์หลักเสมอ และมีคุณสมบัติที่ต่างจากคีย์ตัวอื่นคือ มีค่าเป็นค่าว่าง (Null Character) ไม่ได้ และในเอนทิตีหนึ่ง ๆ อาจจะกำหนดให้มีคีย์รองได้ ซึ่งคีย์รองก็มีคุณสมบัติในการอ้างอิงถึงรายการต่าง ๆ ในเอนทิตีได้เหมือนคีย์หลัก แต่จะต้องเป็นคนละชุดกับคีย์หลัก สำหรับรายการใดที่แบ่งเป็นซับไทป์ (Subtype) และซูเปอร์ไทป์ (Supertype) คีย์หลักของเอนทิตีที่เป็นซับไทป์นั้น จะต้องเหมือนกับคีย์หลักของซูเปอร์ไทป์เอนทิตี

1.4 LDM 4 : กำหนดคีย์นอก (Foreign Keys)

คีย์นอกคือแอททริบิวต์ของเอนทิตีหนึ่งที่ซ้ำกับแอททริบิวต์ในอีกเอนทิตีหนึ่ง เป็นตัวทำให้เกิดความสัมพันธ์ ระหว่างเอนทิตี และใช้เป็นส่วนช่วยควบคุมความถูกต้องของข้อมูลด้วย โดยคีย์นอกจะอยู่ในเอนทิตีลูก และมีค่าเท่ากับคีย์หลักของเอนทิตีแม่ ความสำคัญของคีย์นอกคือ

1.4.1 ทำให้เกิดกฎธุรกิจ (Business Rule) กับเอนทิตีต่าง ๆ

1.4.2 เป็นประโยชน์ในด้านการทำให้สามารถมองเห็นได้ว่าเอนทิตีใดเป็นเอนทิตีลูก และเอนทิตีใดเป็นเอนทิตีแม่

1.4.3 ทำให้การออกแบบฐานข้อมูลง่ายขึ้น

1.5 LDM 5 : กำหนดกฎเกณฑ์การจัดการข้อมูล (Key Business Rules)

เป็นสิ่งที่ใช้ตรวจสอบเพื่อให้แน่ใจในความถูกต้อง และความสอดคล้องกันของข้อมูล มี 3 เรื่องดังนี้

1.5.1 โดเมน (Domain)

1.5.2 กฎการเพิ่มและลบข้อมูล (Insert and Delete Rule)

1.5.3 ทริกเกอร์ดำเนินการ (Trigger Operation)

1.5.1 โดเมน หมายถึงกรอบของค่าต่าง ๆ ที่เป็นไปได้ของแอททริบิวต์ เช่น กรอบของแอททริบิวต์วันที่ จะกินความเฉพาะค่าวันที่ที่เป็นไปได้ เช่น จำนวนวันอยู่ระหว่าง 1 ถึง 31 มีเดือนอยู่ระหว่าง 1 ถึง 12 หรือโดเมน ของชื่อต้องเป็นตัวอักษรเท่านั้น เป็นต้น

1.5.2 กฎการเพิ่มและการลบข้อมูล

1.5.2.1 กฎการเพิ่มข้อมูล เป็นการกำหนดเงื่อนไขที่ต้องตรวจสอบสำหรับเอนทิตีแม่ ในการเพิ่มคีย์นอกในเอนทิตีลูก โดยทั่วไปมี 6 ลักษณะ คือ

1.5.2.1.1 การเพิ่มแบบขึ้นต่อกัน (Dependent) อนุญาติให้เพิ่มข้อมูลในเอนทิตีลูกได้ถ้ามีข้อมูลที่สัมพันธ์กันในเอนทิตีแม่

1.5.2.1.2 การเพิ่มแบบอัตโนมัติ (Automatic) ถ้าตรวจสอบพบว่าไม่มีข้อมูลที่สัมพันธ์กันในเอนทิตีแม่ จะเพิ่มค่าข้อมูลในเอนทิตีแม่ให้เองทันที

1.5.2.1.3 เปลี่ยนเป็นค่าว่าง (Nullify) ถ้าตรวจสอบพบว่าไม่มีข้อมูลที่สัมพันธ์กันในเอนทิตีแม่จะเปลี่ยนค่าคีย์นอกเป็นค่าว่าง โดยที่ค่าว่าง ก็คือค่าที่ไม่ทราบแน่ชัด หรือค่าที่ไม่เหมาะสม กล่าวอีกนัยหนึ่งก็คือค่าว่างจะเป็นค่าที่ไม่อยู่ในกรอบของโดเมนนั่นเอง

1.5.2.1.4 เปลี่ยนเป็นค่าที่กำหนด (Default) กรณีที่ไม่มีข้อมูลในเอนทิตีแม่จะเปลี่ยนค่าคีย์นอกเป็นค่าที่กำหนดไว้

1.5.2.1.5 แบบมีเงื่อนไข (Customized) จะเพิ่มข้อมูลในเอนทิตีลูกได้ก็ต่อเมื่อเงื่อนไขที่กำหนดไว้ถูกตรวจสอบว่าเป็นจริง

1.5.2.1.6 ไม่ตรวจสอบ (No Effect) สามารถเพิ่มข้อมูลในเอนทิตีลูกได้โดย ไม่ต้องตรวจสอบจากเอนทิตีแม่ว่ามีค่าตรงกับเอนทิตีแม่หรือไม่

1.5.2.2 กฎการลบข้อมูล เป็นการกำหนดเงื่อนไขในการลบคีย์หลักในเอนทิตีแม่ที่ถูกอ้างอิงถึงโดยคีย์นอกในเอนทิตีลูก โดยทั่วไปมี 6 ลักษณะ คือ

1.5.2.2.1 การลบแบบมีข้อจำกัด (Restricted) จะลบข้อมูลในเอนทิตีแม่ได้ ก็ต่อเมื่อไม่มีข้อมูลที่สัมพันธ์กันในเอนทิตีลูกเท่านั้น

1.5.2.2.2 กระทำแบบเป็นทอด ๆ (Cascade) จะลบข้อมูลในเอนทิตีแม่และจะลบข้อมูลที่สัมพันธ์กันในเอนทิตีลูกออกด้วย

1.5.2.2.3 เปลี่ยนเป็นค่าว่าง ในการลบข้อมูลจะทำการเปลี่ยนข้อมูลของคีย์นอกที่อ้างอิงถึงให้เป็นค่าว่าง

1.5.2.2.4 เปลี่ยนเป็นค่าที่กำหนด การลบข้อมูล จะทำการเปลี่ยนข้อมูลของคีย์นอกที่อ้างอิงให้เป็นค่าที่กำหนดไว้

1.5.2.2.5 แบบมีเงื่อนไข การลบข้อมูลจะกระทำได้ก็ต่อเมื่อเป็นไปตามเงื่อนไขที่กำหนดไว้

1.5.2.2.6 ไม่ตรวจสอบการลบข้อมูล โดยไม่ต้องทำการตรวจสอบเงื่อนไขใด ๆ ทั้งสิ้น

1.5.3 ทริกเกอร์ดำเนินการ เป็นข้อกำหนดถึงผลกระทบที่ต้องดำเนินการกับเอนทิตีอื่น หรือแอททริบิวต์อื่นในเอนทิตีเดียวกัน เพื่อมีการเพิ่ม ลบ แก้ไข หรือเรียกใช้ข้อมูลต่อเอนทิตีอื่น ๆ หรือแอททริบิวต์อื่นในเอนทิตีเดียวกัน

1.6 LDM 6 : เพิ่มแอททริบิวต์อื่น ๆ ลงในเอนทิตี

หลังจากกำหนดคีย์หลัก คีย์รอง และคีย์นอกให้กับแต่ละเอนทิตีแล้ว ต่อไปก็จะเป็นการกำหนดแอททริบิวต์อื่น ๆ ที่เหลือ (Nonkey Attribute) ให้กับเอนทิตี โดยมีข้อแม้ว่าต้องเป็นแอททริบิวต์ที่ขึ้นกับคีย์หลักเท่านั้น และยังคงทำให้คีย์หลักมีค่าไม่ซ้ำกันด้วย (Unique)

ในกรณีที่แอททริบิวต์ใด สามารถหาได้จากสูตร หรือคำนวณได้จากแอททริบิวต์อื่น จะเรียกว่าเป็น ดีไรฟ์แอททริบิวต์ (Derived Attribute) ให้ระบุตัว 'd' ในโมเดลข้อมูลด้วย กรณีเป็นแฟล็กใส่ไว้เพื่อระบุซับไทด์ - ซุปเปอร์ไทด์ ให้ระบุตัว ' s ' ในโมเดลเช่นกัน

1.7 LDM 7 : ตรวจสอบความถูกต้องของมุมมองผู้ใช้โดยใช้ทฤษฎีนอร์มัลไลเซชัน (Validate User Views Through Normalization Rules)

ข้อดีของการทำนอร์มัลไลเซชัน

1.7.1 ลดเนื้อที่ที่ใช้เก็บข้อมูล

1.7.2 ลดปัญหาเรื่องความขัดแย้งของข้อมูล

1.7.3 ลดความซ้ำซ้อนของข้อมูล

นอร์มัลไลเซชันเป็นทฤษฎีที่ใช้วิเคราะห์และจัดโครงสร้างข้อมูลใหม่ เพื่อลดความซ้ำซ้อนของโครงสร้างข้อมูล และได้โครงสร้างที่มีเสถียรภาพ โดยการออกแบบระบบให้อยู่ในรูปนอร์มัล (Normal Form) ก่อนที่จะกล่าวถึงรูปนอร์มัล ขอกล่าวถึงนิยามที่เกี่ยวข้องดังต่อไปนี้

1. **ตัวเลือก (Determinant)** คือแอททริบิวต์หรือกลุ่มของแอททริบิวต์ใด ๆ ที่สามารถเลือก (Determine) แอททริบิวต์ตัวอื่น ๆ ได้

2. **ฟังก์ชันการขึ้นต่อกัน (Functional Dependency)**

นิยาม ถ้าให้ B และ A เป็นแอททริบิวต์ กล่าวได้ว่า B มีฟังก์ชันการขึ้นอยู่กับค่าของ A ก็ต่อเมื่อ สามารถใช้ A เป็นตัวเลือกค่าของ B ได้เพียงหนึ่งค่าเสมอ

สัญลักษณ์ที่ใช้แทน B มีฟังก์ชันการขึ้นอยู่กับ A คือ A -> B

3. **การขึ้นต่อกันอย่างสมบูรณ์ (Full Functional Dependency)**

นิยาม ถ้าให้ B และ A เป็นแอททริบิวต์ กล่าวได้ว่า B มีฟังก์ชันการขึ้นอยู่กับค่าของ A อย่างสมบูรณ์ ก็ต่อเมื่อ B มีฟังก์ชันการขึ้นอยู่กับค่าของ A แต่ไม่มีคุณสมบัติฟังก์ชันการขึ้นอยู่กับแอททริบิวต์อื่นที่เป็นส่วนประกอบของ A

4. **การขึ้นต่อกันเชิงกลุ่ม (Multivalued Dependency)**

นิยาม ในเอนทิตีที่ประกอบด้วยแอททริบิวต์ A B และ C กล่าวได้ว่าการขึ้นต่อกันเชิงกลุ่มระหว่าง B และ A โดยที่ B ขึ้นต่อ A คือค่าของ A หนึ่งค่าจะอิงกับกลุ่มของ B โดยการขึ้นต่อกันนี้จะเป็นอิสระกับค่าของ C

หลักการนอร์มัลไลเซชันโดยสังเขป คือการจัดโครงสร้างข้อมูลให้อยู่ในรูปนอร์มัล ซึ่งประกอบด้วยระดับต่าง ๆ ดังต่อไปนี้

1. **รูปแบบนอร์มัลระดับที่ 1 (First Normal Form : 1NF)**

นิยาม เอนทิตีจะอยู่ในรูปนอร์มัลระดับที่ 1 ถ้าไม่มีกลุ่มของแอททริบิวต์ที่มีค่าซ้ำกัน กล่าวคือ การไม่มีรายการใด ๆ ของเอนทิตีที่ขึ้นต่อคีย์หลักมากกว่า 1 รายการ

2. **รูปแบบนอร์มัลระดับที่ 2 (Second Normal Form : 2NF)**

นิยาม เอนทิตีจะอยู่ในรูปนอร์มัลระดับที่ 2 เมื่อเอนทิตีดังกล่าวอยู่ในรูปนอร์มัลระดับที่ 1 และไม่มีนันคีย์แอททริบิวต์ตัวใด ขึ้นกับส่วนใดส่วนหนึ่งของคีย์

3. **รูปแบบนอร์มัลระดับที่ 3 (Third Normal Form : 3NF)**

นิยาม เอนทิตีจะอยู่ในรูปนอร์มัลระดับที่ 3 เมื่อเอนทิตีดังกล่าวอยู่ในรูปนอร์มัลระดับที่ 2 และนำแอททริบิวต์อื่นที่ขึ้นกับนันคีย์แอททริบิวต์ออก

4. **รูปนอร์มัลแบบบอยซ์และคอดค์ (Boyce/Codd Normal Form : BCNF)**

นิยาม เอนทิตีจะอยู่ในรูปนอร์มัลแบบบอยซ์และคอดค์ ก็ต่อเมื่อเอนทิตีดังกล่าวอยู่ในรูปนอร์มัลระดับที่ 3 และตัวเลือกทุกตัวต้องเป็นแคนดิเดคคีย์เท่านั้น

5. **รูปแบบนอร์มัลระดับที่ 4 (Fourth Normal Form : 4NF)**

นิยาม เอนทิตีจะอยู่ในรูปนอร์มัลระดับที่ 4 ก็ต่อเมื่อเอนทิตีดังกล่าวอยู่ในรูปแบบบอยซ์และคอดค์ และเป็นความสัมพันธ์ที่ไม่ขึ้นต่อกันเชิงกลุ่ม

6. รูปแบบนอร์มัลระดับที่ 5 **(Fifth Normal Form : 5NF)**

นิยาม เอนทิตีจะอยู่ในรูปนอร์มัลระดับที่ 5 ก็ต่อเมื่อเอนทิตีดังกล่าวอยู่ในรูปนอร์มัลระดับที่ 4 และไม่สามารถแยกเอนทิตีดังกล่าวออกได้อีก

สำหรับการออกแบบจำลองข้อมูลเชิงตรรก โดยการประยุกต์ใช้ทฤษฎีนี้ ถึงแม้ว่าจะได้โครงสร้างที่ถูกขจัดความซ้ำซ้อนและมีเสถียรภาพ แต่เมื่อแปลงอยู่ในรูปฐานข้อมูลเชิงกายภาพ อาจมีปัญหาในเรื่องประสิทธิภาพในการประมวลผล จึงอาจต้องมีการปรับให้อยู่ในรูปแบบที่เหมาะสมในภายหลัง นอกจากนี้การที่จะให้แน่ใจว่าระบบที่ได้ถูกต้องตามความต้องการของผู้ใช้ การใช้ทฤษฎีนอร์มัลไลเซชันควรกระทำภายหลังจากการออกแบบผ่านขั้นตอน 1 (LDM 1) ถึงขั้นตอนที่ 6 (LDM 6) มาก่อน

ปกติการจัดโครงสร้างข้อมูลให้อยู่ในรูปนอร์มัลระดับที่ 3 ก็เพียงพอในการขจัดปัญหาความซ้ำซ้อนและความขัดแย้งกันของข้อมูล ส่วนการจัดให้อยู่ในรูปนอร์มัลระดับสูงขึ้นต้องพิจารณาถึงความจำเป็นของข้อมูล

1.8 LDM 8 : กำหนดโดเมนของแอททริบิวต์

เป็นขั้นตอนการกำหนดโมนของแอททริบิวต์ใด ๆ พร้อมบันทึกรายละเอียดในพจนานุกรมข้อมูล (Data Dictionary)

1.8.1 รายละเอียดที่บันทึกในพจนานุกรม ประกอบด้วยรายละเอียดดังนี้

1.8.1.1 ชนิดของข้อมูล (Data Type) เช่น จำนวนเต็ม ทศนิยม ตัวอักษร วันที่

1.8.1.2 รูปแบบข้อมูล (Format) เช่น dd/mm/yy (วันที่) nnn-nnnn (เบอร์โทรศัพท์)

1.8.1.3 ค่าที่ยอมรับได้ (Range) เช่น อายุพนักงานจะเป็นเลขจำนวนเต็ม 2 หลักเท่านั้น

1.8.1.4 ความหมาย (Meaning) คือการอธิบายความหมายของแอททริบิวต์นั้นว่าคืออะไร

1.8.1.5 ความยาว (Length) เช่น 7 หลัก 20 ตัวอักษร

1.8.1.6 มีค่าซ้ำกันได้หรือไม่ (Uniqueness)

1.8.1.7 เป็นค่าว่างได้หรือไม่ (Null Support)

1.8.2 กฎพิเศษสำหรับโดเมนของแอททริบิวต์บางชนิด ได้มีการกำหนดไว้เป็นพิเศษสำหรับโดเมนของแอททริบิวต์บางพวกต่อไปนี้

1.8.2.1 แอททริบิวต์ที่เป็นคีย์หลัก ซึ่งจะเป็นเพียงแอททริบิวต์เดียวหรือแอททริบิวต์ประกอบ จะต้องมีความเป็นหนึ่งเดียวและเป็นค่าว่างไม่ได้ แต่แอททริบิวต์ย่อย ๆ แต่ละตัวที่ประกอบเป็นคีย์หลัก ไม่จำเป็นต้องเป็นหนึ่งเดียวแต่รวมแล้วต้องมีความเป็นหนึ่ง

1.8.2.2 แอททริบิวต์ที่เป็นคีย์รอง มีกฎเหมือนกับคีย์หลัก (ต้องมีค่าไม่ซ้ำกัน) แต่เป็นค่าว่างได้

1.8.2.3 คีย์นอกต้องสอดคล้องกับโดเมนของคีย์หลักในเอนทิตีแม่

1.8.2.4 ดีไรฟ์แอททริบิวต์ ต้องมีค่าอยู่ในช่วงของผลลัพท์ที่ได้จากการผ่านอัลกอริทึมของแอททริบิวต์ที่เป็นที่มา และมีชนิดของข้อมูลแบบเดียวกันด้วย

1.8.2.5 แอททริบิวต์ที่เป็นคีย์หลักของซับไทป์เอนทิตี จะต้องมีค่าเป็นสับเซท (Subset) ของคีย์หลักของซูเปอร์ไทป์เอนทิตี

1.9 LDM 9 : กำหนดทริกเกอร์ดำเนินการ

เป็นขั้นตอนการกำหนดกฎเพื่อตรวจสอบความถูกต้อง และการควบคุมผลที่จะเกิดจากการกระทำต่าง ๆ เช่น การเพิ่ม การแก้ไข การลบ รวมทั้งการค้นหารายการ โดยที่จะพิจารณาถึงผลกระทบที่จะเกิดขึ้นกับเอนทิตีอื่น หรือ แอททริบิวต์อื่นภายในเอนทิตีเดียวกัน เมื่อกำหนดเรียบร้อยแล้วให้บันทึกลงในพจนานุกรมข้อมูล

1.10 LDM 10 : รวมมุมมองของผู้ใช้เข้าด้วยกันเพื่อลดความซ้ำซ้อน

ความไม่สอดคล้องของข้อมูล และเพิ่มความสัมพันธ์ระหว่างมุมมองของผู้ใช้ ซึ่งประกอบด้วยการรวมเอนทิตี การรวมความสัมพันธ์ และการรวมแอททริบิวต์

1.10.1 ข้อควรพิจารณาในการรวมเอนทิตี

1.10.1.1 การรวมเอนทิตีที่มีคีย์หลักตัวเดียวกัน และค่าที่เป็นไปได้ของคีย์หลักเหมือนกัน จะต้องได้เอนทิตีใหม่ที่มีแอททริบิวต์รวมของเอนทิตีเดิม

1.10.1.2 ถ้าเอนทิตีสองอันมีคีย์หลักเดียวกัน แต่ค่าที่เป็นไปได้ของคีย์หลักนั้นเป็นสับเซตของกัน จะรวมได้ในรูปของซูเปอร์ไทป์ซับไทป์ โดยต้องตัดทำการแอททริบิวต์ที่มีแล้วในซูเปอร์ไทป์เอนทิตีออกจากซับไทป์เอนทิตี

1.10.1.3 ถ้าเอนทิตีสองอันมีคีย์หลักอันเดียวกัน แต่มีผลไปกำหนดแอททริบิวต์ที่ต่างกันบางตัว ให้กำหนดซูเปอร์ไทป์ขึ้นมาอันหนึ่งให้สัมพันธ์กับสองเอนทิตีเดิม

1.10.1.4 การเชื่อมเอนทิตีสองตัว ที่มีคีย์หลักของตัวหนึ่งเป็นคีย์รองของอีกตัวหนึ่ง จะได้เอนทิตีใหม่ที่มีคีย์หลักตามเอนทิตีตัวหนึ่ง ส่วนคีย์หลักของเอนทิตีอีกตัวจะกลายเป็นคีย์รองไป และมีแอททริบิวต์รวมระหว่างสองเอนทิตีเดิม แล้วตัดแอททริบิวต์ที่ซ้ำซ้อนออกเสีย และต้องกฎเกณฑ์การแก้ไขให้ด้วยตามข้อบังคับเก่า เช่น ในกรณีคีย์หลักเดิมที่กลายมาเป็นคีย์รองในเอนทิตีใหม่ก็ต้องยังคงห้ามเป็นค่าว่างด้วย

1.10.1.5 การรวมเอนทิตีใด ๆ ก็ตาม ต้องไม่มีผลไปเปลี่ยนแปลงเอนทิตีอื่นที่ไม่เกี่ยวข้อง

1.10.2 ข้อควรพิจารณาในการรวมรีเลชันชิป

1.10.2.1 ให้รวมรีเลชันชิประหว่างเอนทิตี ที่ให้ความหมายเหมือนกันเข้าด้วยกัน โดยถ้าผลทำให้เกิดเป็นรีเลชันชิปแบบ M : N จะต้องการทำการแตกให้เป็นรีเลชันชิปแบบ 1 : N สองอัน

1.10.2.2 ในการรวมรีเลชันชิปใด ๆ ต้องไม่ไปกระทบกับรีเลชันชิปอื่นที่ไม่ต้องการการเปลี่ยนแปลง นอกจากจะพิจารณาแล้วว่าควรตัดออกเนื่องจากความซ้ำซ้อน หรือ อาจเพิ่มขึ้นใหม่เพื่อความเหมาะสม

1.10.2.3 จากการรวมเอนทิตีที่มีคีย์หลัก เป็นคีย์รองของเอนทิตีอีกตัวให้ตรวจสอบคีย์นอกของเอนทิตีอื่น ๆ ที่อ้างมาถึงว่าได้อ้างถึงคีย์หลักหรือคีย์รองของเอนทิตีใหม่ที่ได้จากการรวมนั้น ถ้าอ้างถึงคีย์รองต้องทำการเปลี่ยนให้เป็นคีย์หลัก

1.10.2.4 เมื่อรวมมุมมองต่าง ๆ แล้ว ให้กำหนดกฎเกณฑ์การแก้ไขคีย์สำหรับรีเลชันชิปใหม่ด้วย

1.10.3 ข้อควรพิจารณาในการรวมแอททริบิวต์

1.10.3.1 ให้ทำการรวมแอททริบิวต์ที่มีความหมายเหมือนกันภายในเอนทิตีเดียวกัน และรวมค่าที่เป็นไปได้รวมถึงกฎการจัดการเข้าด้วยกันด้วย โดยพิจารณาค่าที่เป็นไปได้ของแอททริบิวต์อื่นด้วยว่าเปลี่ยนไปหรือไม่

1.10.3.2 เมื่อรวมเอนทิตีแล้ว ให้พิจารณาตัดแอททริบิวต์ที่เป็นคีไรพ์ หรือ แฟล็กที่ไม่จำเป็นทิ้ง

1.10.3.3 หลังจากได้รวม ตัด หรือ เพิ่มรีเลชันชิปแล้วให้ทำการนอร์มัลไลซ์อีกครั้งหนึ่ง เพื่อตัดสิ่งที่ซ้ำซ้อนออกเสีย

1.11 LDM 11 : รวมโมเดลที่ได้เข้ากับโมเดลที่มีอยู่แล้ว

การรวบรวมแบบจำลองที่มีอยู่เข้าด้วยกัน เป็นขั้นตอนที่เกี่ยวข้องกับส่วนประกอบของโครงสร้างข้อมูล ในระดับแบบจำลองเชิงแนวคิดโดยรวมแบบจำลองข้อมูลเชิงกตตรกกับแบบจำลองข้อมูลที่มีอยู่แล้ว พร้อมทั้งการพิจารณาการสร้างแบบจำลองใหม่ไปพร้อมกับการพิจารณากฎเกณฑ์ข้อบังคับเดิม ซึ่งอาจมีการใช้เอนทิตีหรือรีเลชั่นชิปรวมกับขอบเดิม หรืออาจมีการกำหนดเอนทิตีใหม่ด้วย

1.12 LDM 12 : วิเคราะห์ความเสถียรภาพและการเติบโตในอนาคต

การออกแบบโมเดลในขั้นตอนที่ผ่านมาจะพิจารณาข้อมูลที่เห็นได้ในปัจจุบันเป็นส่วนมาก ในขั้นสุดท้ายนี้จะพิจารณาสิ่งที่อาจเกิดขึ้นหรือเป็นไปได้ในอนาคต แล้วทำการดัดแปลงโมเดลไว้รองรับ หรือจดบันทึกไว้เพื่อพิจารณาในภายหลัง และให้สามารถรองรับกับการเปลี่ยนแปลง

2. ฐานข้อมูลเชิงสัมพันธ์ (Relational Database)

เป็นหลักเกณฑ์ที่ใช้ในการแปลงโมเดลข้อมูลเชิงตรรก เป็นโมเดลข้อมูลเชิงกายภาพเพื่อให้สะดวกในการใช้งานจริงบนฐานข้อมูลเชิงสัมพันธ์ ซึ่งประกอบด้วยโครงสร้างข้อมูล (Data Structure) และความถูกต้องของข้อมูล (Data Integrity)

2.1 โครงสร้างของฐานข้อมูลเชิงสัมพันธ์ (Data Structure) หมายถึง ส่วนที่อธิบายถึงโครงสร้างข้อมูล อันประกอบไปด้วย เอนทิตี แอททริบิวต์ และ ความสัมพันธ์ของแต่ละเอนทิตี การสร้างตารางความสัมพันธ์เทียบได้กับเอนทิตี และการกำหนดสดมภ์ซึ่งเทียบเท่ากับแอททริบิวต์ ในแบบจำลองข้อมูลเชิงตรรก (Logical Data Modeling) การสร้างตารางความสัมพันธ์นี้อาจจะมีการรวมตารางเข้าด้วยกัน หรือ แยกตารางออกจากกัน เพื่อประโยชน์ในแง่การทำงานให้รวดเร็ว ลักษณะโครงสร้างข้อมูล ประกอบด้วย

2.1.1 สดมภ์ (Column) คือ คุณสมบัติของข้อมูลเมื่อเปรียบเทียบกับแบบจำลองข้อมูลเชิงตรรก ก็คือแอททริบิวต์ในเอนทิตีนั่นเอง

2.1.2 แถว (Row) คือ ค่าของข้อมูลในตาราง

2.1.3 ตารางความสัมพันธ์ (Table) เป็นตาราง 2 มิติ ที่

2.1.3.1 ต้องไม่มีคอลัมน์ที่ซ้ำกันในตาราง

2.1.3.2 ค่าของข้อมูลในคอลัมน์จะต้องมีคุณสมบัติเหมือนกัน

2.1.3.3 ข้อมูลในตาราง จะต้องไม่มีข้อมูลแถวใดที่ซ้ำกัน

2.1.3.4 ลำดับของคอลัมน์ไม่มีความสำคัญ

2.1.3.5 ลำดับข้อมูลแต่ละแถว ไม่มีความสำคัญ

2.1.3.6 ชื่อคอลัมน์ในตารางต้องไม่ซ้ำกัน

2.2 ความถูกต้องของข้อมูล (Data Integrity) คือ การรักษาให้ข้อมูลมีความเชื่อถือ มีความถูกต้อง และป้องกันมิให้เกิดความเสียหายกับข้อมูล ความเป็นบูรณภาพของข้อมูลมีกฎเกณฑ์ความถูกต้องดังนี้ เมื่อกล่าวถึงความถูกต้องของข้อมูล โดยทั่วไปแล้วจะกินความหมายถึงกฎเกณฑ์ความถูกต้องอยู่ 3 กรณีคือ

2.2.1 ความถูกต้องของเอนทิตี (Entity Integrity Rules) ซึ่งกล่าวว่า ' **แอททริบิวต์ทุกตัวที่เป็นส่วนของคีย์หลัก จะไม่อนุญาตให้มีค่าว่าง** ' เป็นกฎสำหรับการเพิ่ม ปรับปรุง และลบข้อมูลในตาราง

2.2.2 ความถูกต้องของการอ้างอิง (Referential Integrity Rules) ซึ่งกล่าวว่า ' ถ้าตารางความสัมพันธ์ R2 ซึ่งมี FK (Foreign Key) เป็นคีย์นอกที่อ้างอิงถึงคีย์หลัก PK (Primary Key) ในตารางความสัมพันธ์ R1 สำหรับทุก ๆ ค่าของ FK ใน R2 จะต้อง

2.2.2.1 มีค่าเท่ากับ PK ในแถวใดแถวหนึ่งในตารางความของสัมพันธ์ R1

2.2.2.2 มีค่าของแอททริบิวต์ทุกตัวใน FK เป็นค่าว่าง

ความสำคัญของกฎข้อนี้คือ เมื่อมีการอ้างอิงจากความสัมพันธ์ที่ 1 ไปยังความสัมพันธ์ที่ 2 ต้องมีค่าเสมอ

2.2.3 ความถูกต้องของโดเมน (Domain Integrity Rules) หมายถึงกฎการรักษาความถูกต้องของคอลัมน์ทุกคอลัมน์ในตาราง รวมทั้งคีย์หลัก คีย์นอกและคอลัมน์ที่ไม่ใช่คีย์ ซึ่งกฎนี้ประกอบด้วย

2.2.3.1 ชนิดของข้อมูล

2.2.3.2 รูปแบบของข้อมูล

2.2.3.3 ความยาวของข้อมูล

2.2.3.4 ช่วงค่าของข้อมูล

2.2.3.5 ค่าที่กำหนดไว้

2.2.3.6 มีค่าซ้ำกันหรือค่าว่างได้หรือไม่

เวลาที่พูดถึงกฎเกณฑ์ความถูกต้องดังกล่าว เราต้องการที่จะประกันความถูกต้องนี้ในทุก ๆ สถาวะ และทุก ๆ ขณะของฐานข้อมูล นั่นก็หมายความว่าเมื่อมีการแก้ไขข้อมูลในระบบเกิดขึ้นเมื่อไร ก็เป็นหน้าที่ของระบบจัดการฐานข้อมูลที่จะต้องตรวจสอบอยู่เสมอว่า

ยังคงมีความถูกต้องตามกกเกณฑ์อยู่ และถ้าการแก้ไขข้อมูลครั้งใด จะยังผลให้ผิดกฎเกณฑ์ความถูกต้องไปแล้วก็ขึ้นอยู่กับการออกแบบของระบบจัดการฐานข้อมูลแต่ละตัวว่าจะแก้ไขสถานการณ์อย่างไร

วิธีการแก้ไขสถานการณ์แบ่งได้เป็น 3 วิธีคือ

วิธีที่ 1 : ระบบจะไม่ยอมรับการเปลี่ยนแปลงแก้ไขข้อมูล ที่กำให้เกิดปัญหานั้น โดยอาจจะแสดงข้อความออกมาบอกผู้ใช้

วิธีที่ 2 : ระบบจะอนุญาตให้การแก้ไขนั้น ๆ เกิดขึ้นได้ โดยระบบจะทำการปรับให้ผลลัพท์มีความคงสภาพโดยอัตโนมัติ

วิธีที่ 3 : ระบบจะอนุญาตให้ผู้ใช้หรือผู้บริหารฐานข้อมูล สามารถตัดสินใจได้ในแต่ละคราวว่าจะยอมหรือไม่ยอมให้มีการแก้ไขข้อมูล

2.3 ขั้นตอนการออกแบบฐานข้อมูลเชิงสัมพันธ์ มีขั้นตอนสำคัญดังนี้

2.3.1 RDD 1 : สร้างตารางความสัมพันธ์จากเอนทิตี โดยตารางแต่ละตารางจะได้จาก 1 เอนทิตีเท่านั้น

2.3.2 RDD 2 : ระบุสดมภ์ในตาราง โดยกำหนดให้เอนทิตีแต่ละตัว คือ 1 สดมภ์

2.3.3 RDD 3 : ปรับปรุงโครงสร้างของตารางความสัมพันธ์ให้เหมาะกับสภาพแวดล้อมของระบบจัดการฐานข้อมูล รวมถึงประสิทธิภาพในการทำงานด้วย

2.3.4 RDD 4 : ออกแบบกฎธุรกิจที่เกี่ยวข้องกับตารางความสัมพันธ์

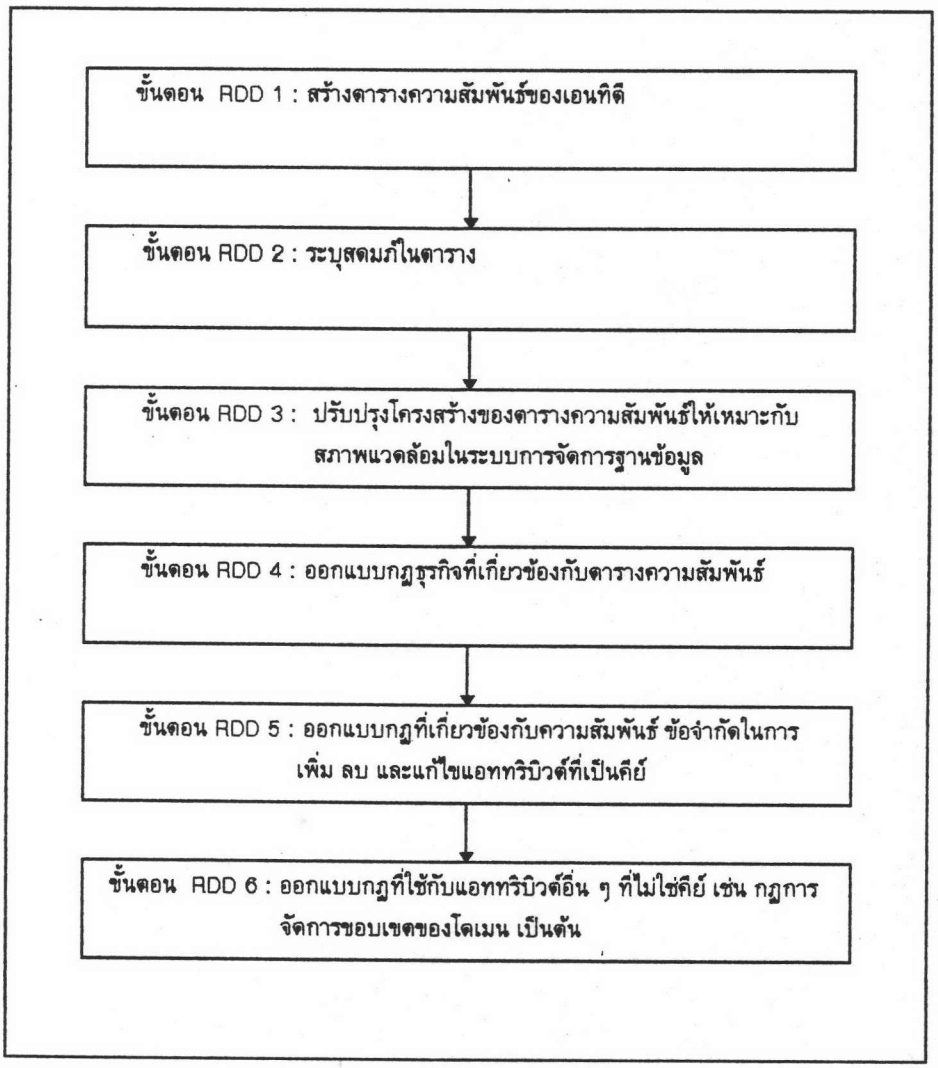
2.3.4.1 กำหนดคุณสมบัติทางตรรกของคีย์หลัก เช่น การแสดงความเป็นหนึ่งเดียว (Uniqueness) การไม่ยอมให้คีย์หลักมีค่าว่าง (null)

2.3.4.2 กำหนดคุณสมบัติทางตรรกของคีย์รอง เช่น การแสดงความเป็นหนึ่งเดียว เป็นต้น

2.3.5 RDD 5 : ออกแบบกฎที่เกี่ยวข้องกับความสัมพันธ์ข้อจำกัดในการเพิ่ม ลบ และแก้ไขแอททริบิวต์ที่เป็นคีย์ในลักษณะให้มีความสอดคล้องกัน

2.3.6 RDD 6 : ออกแบบกฎที่ใช้กับแอททริบิวอื่น ๆ ที่ไม่ใช่คีย์ ได้แก่ การกำหนดกฎการจัดการ การกำหนดขอบเขตของโดเมน ชนิดข้อมูล รูปแบบข้อมูล ความเป็นเอกลักษณ์ กำหนดค่าปริยายหรือการยอมให้มีค่าว่างได้หรือไม่

ขั้นตอนการเปลี่ยนแปลงแบบจำลองข้อมูลเชิงตรรก ให้เป็นโครงสร้างฐานข้อมูลเชิงสัมพันธ์สามารถแสดงได้ดังรูป 3.4



รูปที่ 3.4 แสดงกระบวนการเปลี่ยนแบบจำลองข้อมูลเชิงตรรกเป็นโครงสร้างฐานข้อมูลเชิงสัมพันธ์

2.4 ลักษณะของฐานข้อมูลเชิงสัมพันธ์ที่สมบูรณ์

นักคณิตศาสตร์ชื่อ อี เอฟ คอดด์ (E.F. Codd) ได้เสนอรากฐานของระบบฐานข้อมูลเชิงสัมพันธ์ว่า คือการรวบรวมตารางแบบต่าง ๆ ที่สามารถกระทำการใด ๆ บนคณิตศาสตร์ของตรรกได้ และตารางนี้จะต้องเป็นไปตามข้อบัญญัติ (Fidelity Rules) ดังนี้

2.4.1 **กฎข้อที่ 0 (Foundation Rule)** ระบบใด ๆ จะประกาศว่าเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์จะต้องจัดการกับฐานข้อมูลทั้งหมด ด้วยความสามารถในลักษณะที่เป็นตาราง

2.4.2 **กฎข้อที่ 1 (Information Rule)** ข้อมูลทั้งหมดในฐานข้อมูลเชิงสัมพันธ์จะต้องถูกแสดงในระดับตรรกะด้วยวิธีการเดียวเท่านั้น โดยลักษณะของตารางต่าง ๆ เช่น ประกอบด้วย ชื่อตาราง ชื่อคอลัมน์ ชนิดของข้อมูลในแต่ละคอลัมน์ เป็นต้น

2.4.3 **กฎข้อที่ 2 (Guaranteed Access Rule)** ในแต่ละและทุก ๆ ค่าของข้อมูลในฐานข้อมูลเชิงสัมพันธ์ จะต้องรับประกันได้ว่าผู้ใช้จะสามารถเข้าถึงข้อมูลทุกตัวในทุก ๆ ตารางได้โดยรวมกันของชื่อตาราง ค่าคีย์หลัก และชื่อคอลัมน์

2.4.4 **กฎข้อที่ 3 (Missing Information Rule)** การแยกแยะข้อมูลที่ยังไม่พร้อมหรือยังไม่มีค่านั้น จะใช้ค่าว่างเป็นตัวบ่ง เพื่อให้เกิดความแตกต่างระหว่างสตริงที่ว่างหรือสตริงของค่าแบลงค์ หรือค่า 0

2.4.5 **กฎข้อที่ 4 (System Catalog Rule)** โครงสร้างแคตตาลอกของระบบจะต้องมีสภาพของฐานข้อมูลเชิงสัมพันธ์ ดังนั้น ผู้ใช้จะสามารถเรียกดู หรือ แก้ไขได้ด้วยภาษาเดียวกันกับการเรียกดูข้อมูลของระบบ

2.4.6 **กฎข้อที่ 5 (Comprehensive Language Rule)** ไม่ว่าจะมีภาษาอยู่กี่ภาษาหรือโหมดในการติดต่ออยู่กี่โหมดที่ใช้ได้ แต่อย่างน้อยที่สุดจะมี 1 ภาษาที่สามารถนิยามโครงสร้างของระบบข้อมูล นิยามวิว เปลี่ยนแปลงแก้ไขข้อมูล ทั้งแบบการเขียนโปรแกรมหรือใช้คำสั่งแบบมีการประมวลผลโดยทันที ควบคุมความถูกต้องของข้อมูลและให้สิทธิการใช้งาน

2.4.7 **กฎข้อที่ 6 (View Updatability Rule)** ระบบจัดการฐานข้อมูลจะต้องมีการแก้ไขข้อมูลผ่านทางวิวได้โดยสามารถตัดสินใจได้ว่า จะให้ผู้ใช้เพิ่มหรือลบแถวในตาราง หรือแก้ไขคอลัมน์ใด ๆ โดยผ่านทางวิวได้เลยหรือไม่

2.4.8 **กฎข้อที่ 7 (Set Level Update Rule)** ระบบจัดการฐานข้อมูลควรจะมีภาษาที่สามารถให้ผู้ใช้สามารถเพิ่ม ลด และแก้ไขข้อมูลในหลาย ๆ แถว หรือ หลาย ๆ คอลัมน์ได้

2.4.9 **กฎข้อที่ 8 (Physical Data Independence Rule)** ระบบจัดการฐานข้อมูลจะต้องมีความเป็นอิสระของข้อมูลในระดับกายภาพ คือไม่ขึ้นกับฮาร์ดแวร์ หรือวิธีจัดการกับข้อมูล (Access Method)

2.4.10 **กฎข้อที่ 9 (Logical Data Independence Rule)** ระบบจัดการฐานข้อมูลจะต้องมีความเป็นอิสระของข้อมูลในระดับตรรก ไม่มีผลต่อการเปลี่ยนแปลงตารางหลัก

2.4.11 **กฎข้อที่ 10 (Integrity Independence Rule)** ระบบจัดการฐานข้อมูลจะต้องมีความเป็นอิสระในการจัดเก็บฐานข้อมูลเพื่อให้คงความถูกต้อง โดยไม่ขึ้นกับโปรแกรม

2.4.12 **กฎข้อที่ 11 (Distribution Independence Rule)** ระบบจัดการฐานข้อมูลจะต้องไม่ขึ้นต่อการกระจายของข้อมูล คือผู้ใช้ไม่ต้องสนใจว่าข้อมูลจะอยู่ในส่วนใด หรือคอมพิวเตอร์เครื่องใดที่เชื่อมโยงกันเป็นระบบ

2.4.13 กฎข้อที่ 12 **(Nonsubversion Rule)** ระบบจัดการฐานข้อมูลจะต้องไม่ยอมให้ภาษาอื่นใด สามารถทำลายกฎต่าง ๆ โดยเฉพาะอย่างยิ่งในกฎที่เกี่ยวกับความถูกต้องของระบบ หรือไม่ว่าจะเป็นเครื่องมือใด ๆ ก็ตาม