บทที่ 3

แนวคิดและทฤษฎี

เพื่อให้บรรลุถึงวัตถุประสงค์ของการวิจัยดังได้กล่าวไว้ในบทที่ 1 ผู้วิจัยจึงได้ทำการศึกษาทฤษฎีต่าง ๆ ที่เกี่ยวข้อง เพื่อนำมาใช้ในระบบงาน ดังมีรายละเอียดต่อไปนี้

3.1 ระบบสารสนเทศเพื่อการจัดการ (Management Information System) [5]

ระบบสารสนเทศเพื่อการจัดการ หมายถึง ระบบที่มีการรวมอุปกรณ์ หรือเครื่องมือต่าง ๆ เพื่อสร้างข่าวสารข้อมูล ช่วยสนับสนุนการทำงาน (Operation) การบริหาร (Management) และการตัดสินใจในองค์กร (Decision Making) โครงสร้างของระบบสารสนเทศเพื่อการจัดการแบ่งได้เป็น 3 ระดับ ดังนี้

3.1.1 สารสนเทศด้านกลยุทธ์ (Strategic Information) เป็นการคาดการณ์ในอนาคต ซึ่งมีความไม่แน่นอน จะเกี่ยวข้องกับการวางแผนทางด้านนโยบายต่าง ๆ ขององค์กร ได้แก่ งานของผู้บริหารระดับสูง เช่น การวางแผนปรับปรุงหลักสูตรการศึกษาของนิสิต เป็นต้น

3.1.2 สารสนเทศด้านการจัดการ (Management Information) เป็นสารสนเทศเพื่อการจัดการในระดับกลาง โดยผู้ใช้สารสนเทศในระดับนี้คือ ผู้ที่จะนำเอาแผนงานจากผู้บริหารสูงมาขยายให้ชัดเจน และส่งเสริมนโยบายต่าง ๆ ขององค์กร

3.1.3 สารสนเทศด้านการปฏิบัติการ (Operation Information) เป็นสารสนเทศที่ใช้ในระยะสั้น จะถูกใช้ในการดำเนินงานต่าง ๆ ขององค์กร

3.2 ระบบฐานข้อมูล (Database System) [6]

ระบบฐานข้อมูล หมายถึง การรวบรวมข้อมูลที่มีความสัมพันธ์กัน และอาจอยู่ต่างที่กันให้เสมือนอยู่ร่วมกัน เพื่อให้สามารถรับใช้งานที่มีวัตถุประสงค์แตกต่างกันของหน่วยงานต่าง ๆ โดยที่ผู้ใช้ฐานข้อมูลไม่ได้รับรู้ข้อมูลทั้งหมดในฐานข้อมูล แต่รับรู้เฉพาะส่วนที่เกี่ยวข้องกับงานของตน ดังนั้นผู้ใช้แต่ละคนจะรับรู้ข้อมูลที่แตกต่างกันจากฐานข้อมูลเดียวกัน ระบบฐานข้อมูลมีรายละเอียดที่ต้องศึกษาดังนี้

3.2.1 ชนิดของระบบฐานข้อมูล

ระบบฐานข้อมูลแบ่งตามลักษณะการใช้งานออกเป็น 3 ชนิด คือ

3.2.1.1 ฐานข้อมูลเชิงลำดับชั้น (Hierarchical Database)

เป็นฐานข้อมูลรุ่นแรก ปัจจุบันไม่นิยมใช้ มีโครงสร้างในการเก็บข้อมูลเป็นแบบต้นไม้ (Tree) คือจะแบ่งข้อมูลเป็นลำดับชั้นต่ำลงมา แต่ละตัวจะเรียกว่า โหนด (Node) โหนดตัวแรกสุดเรียกว่า รูท (Root) โหนดถัดไปที่มีการแตกย่อยจะเรียกโหนดตัวบนว่า โหนดแม่ (Parent Node) และเรียกโหนดตัวล่างว่า โหนดลูก (Child Node) โดยมีหลักการว่า โหนดแม่ชี้ไปยังโหนดลูก การเข้าถึงข้อมูลจะต้องผ่านโหนดแม่ลงไปจนถึงโหนดลูก ดังนั้นผู้ออกแบบจะต้องกำหนดความสัมพันธ์ระหว่างโหนดแม่และโหนดลูกตั้งแต่เริ่มสร้างฐานข้อมูล ข้อดีของระบบฐานข้อมูลแบบนี้คือความสัมพันธ์ของข้อมูลไม่ซับซ้อน ข้อเสียคือนำไปใช้งานยาก เพราะมีข้อจำกัดที่ว่าโหนดลูกจะมีความสัมพันธ์กับโหนดแม่ได้เพียง 1 ความสัมพันธ์เท่านั้น

3.2.1.2 ฐานข้อมูลเชิงเครือข่าย (Network Database)

เป็นฐานข้อมูลที่ถูกนำมาใช้ในเวลาใกล้เคียงกับระบบฐานข้อมูลเชิงลำดับชั้น ข้อเสียของระบบฐานข้อมูลเชิงลำดับชั้นนั้นถูกแก้ด้วยระบบฐานข้อมูลเชิงเครือข่าย กล่าวคือ 1 เมมเบอร์ (Member) จะมีความสัมพันธ์กับโอนเนอร์ (Owner) ได้มากกว่า 1 ข้อดีคือเหมาะสมกับระบบงานที่มีความสัมพันธ์ของข้อมูลเป็นจำนวนมาก ข้อเสียคือต้องกำหนดความสัมพันธ์ตั้งแต่เริ่มสร้างฐานข้อมูล และไม่สามารถเปลี่ยนแปลงได้ในระหว่างการทำงาน

3.2.1.3 ฐานข้อมูลเชิงสัมพันธ์ (Relational Database)

เป็นระบบฐานข้อมูลที่เก็บข้อมูลในรูปของตารางสองมิติ คือ ด้านแถว (Row) และด้านคอลัมภ์ (Column) ข้อมูลที่เก็บอยู่ในตารางเหล่านี้จะมีความสัมพันธ์ระหว่างกัน โดยความสัมพันธ์ระหว่างตารางเกิดจากคอลัมภ์ที่เหมือนกัน ข้อดีคือการเปลี่ยนแปลงความสัมพันธ์ทำได้ง่าย และสามารถนำไปประยุกต์เป็นระบบฐานข้อมูลเชิงลำดับชั้น หรือเชิงเครือข่ายได้ ข้อเสียถ้าข้อมูลมีความสัมพันธ์กันเป็นจำนวนมาก จะทำให้การค้นหาข้อมูลช้ากว่าแบบอื่น

3.2.2 ข้อดีของระบบฐานข้อมูล

การเก็บรวบรวมข้อมูลจำนวนมากเข้าด้วยกัน และนำมาพัฒนาเป็นระบบฐานข้อมูล มีผลดีคือ

3.2.2.1 ลดปัญหาความซ้ำซ้อนของข้อมูล (Reduce Data Redundancy)

3.2.2.2 ลดปัญหาความขัดแย้งของข้อมูล (Reduce Data Inconsistency)

3.2.2.3 สามารถใช้ข้อมูลร่วมกันได้ (Share of Data)

3.2.2.4 สามารถควบคุมความเป็นมาตรฐานของข้อมูล (Standard of Data)

3.2.2.5 มีระบบรักษาความปลอดภัย (Security Restriction)

3.2.2.6 สามารถควบคุมความคงสภาพของข้อมูลได้ (Data Integrity)

3.2.2.7 ข้อมูลมีความเป็นอิสระ (Data Independence)

3.2.3 สถาปัตยกรรมฐานข้อมูล (Database Architecture)

ได้มีการกำหนดมาตรฐานขึ้นโดย ANSI (American National Standard Institute) ในปี พ.ศ. 2518 เรียกกันว่า สถาปัตยกรรม 3 ระดับ (Three-Level Architecture) ดังรูปที่ 3.1 สถาปัตยกรรมฐานข้อมูลใช้ในการอธิบายถึงโครงสร้างฐานข้อมูลโดยรวม และแสดงให้เห็นถึงความแตกต่างระหว่างฐานข้อมูลเชิงกายภาพกับสิ่งที่ผู้ใช้มองเห็น สถาปัตยกรรม 3 ระดับดังกล่าว ได้แก่

3.2.3.1 แบบแผนฐานข้อมูลภายนอก (External Schemas)

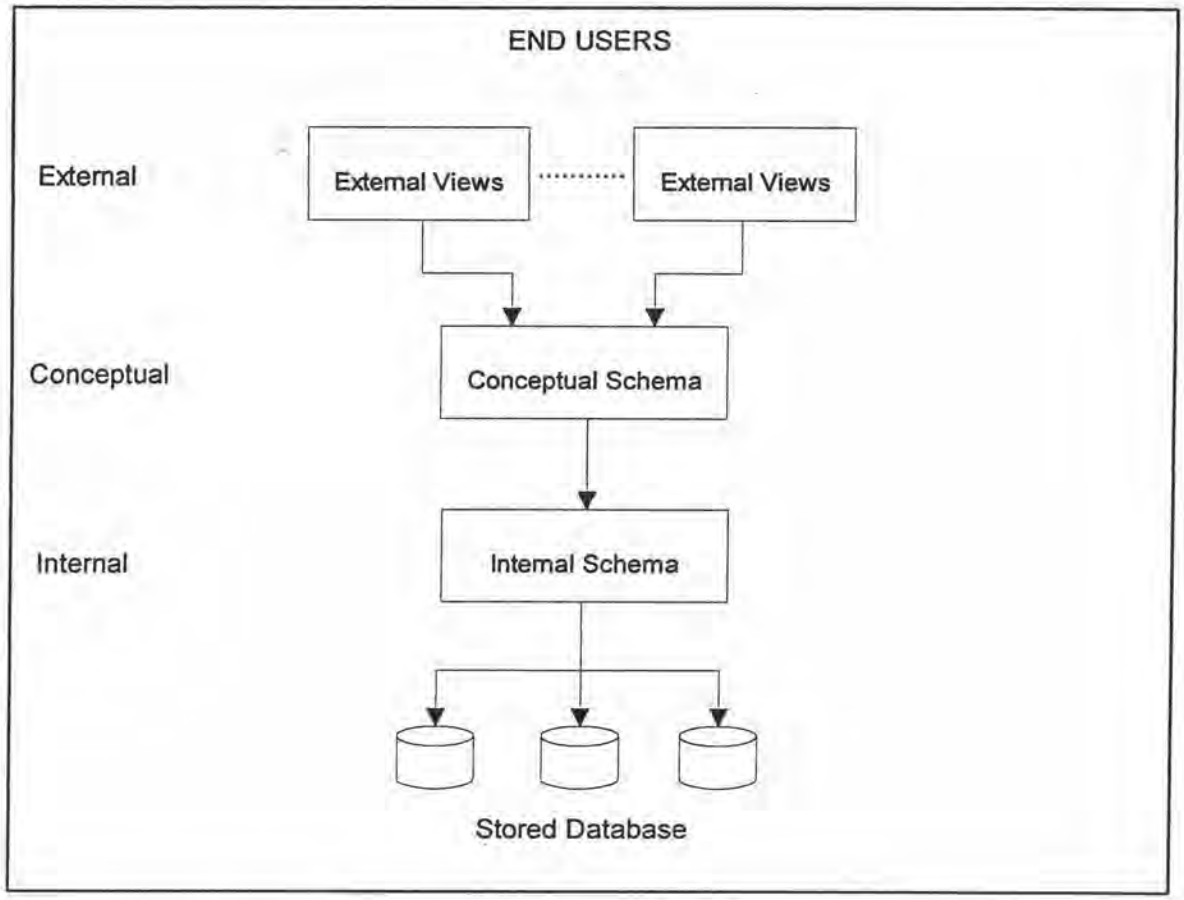
เป็นระดับที่ใกล้กับผู้ใช้มากที่สุดอธิบายถึงวิว (View) ที่ผู้ใช้สนใจ ข้อมูลที่เก็บจริงอาจมีมากกว่าที่ผู้ใช้มองเห็น และข้อมูลเดียวกัน ผู้ใช้อาจมองเห็นแตกต่างกัน เช่น ข้อมูลวันที่ ผู้ใช้คนหนึ่งอาจมองเห็นเป็น วัน/เดือน/ปี ผู้ใช้อีกคนมองเห็นเป็น เดือน/วัน/ปี เป็นต้น

3.2.3.2 แบบแผนฐานข้อมูลเชิงมโนภาพ (Conceptual Schemas)

เป็นส่วนที่เชื่อมระหว่างแบบแผนฐานข้อมูลภายนอกกับแบบแผนฐานข้อมูลภายใน และเป็นระดับที่กำหนดรูปแบบของข้อมูล ได้แก่ ชนิด ลักษณะ และความสัมพันธ์ของแบบจำลองข้อมูลเชิงตรรก นอกจากนี้ยังสามารถกำหนดขอบเขตการใช้ข้อมูลของกลุ่มผู้ใช้งานได้

3.2.3.3 แบบแผนฐานข้อมูลภายใน (Internal Schemas)

เป็นระดับที่อธิบายถึงการเก็บข้อมูลทางกายภาพ พิจารณาข้อมูลโดยการใช้โครงสร้างข้อมูล (Data Structure) และการจัดการแฟ้มข้อมูล (File Organization) จะทำงานร่วมกับระบบปฏิบัติการ (Operating System) เพื่อบันทึกข้อมูลลงบนสื่อบันทึกข้อมูล



รูปที่ 3.1 สถาปัตยกรรมของฐานข้อมูล

3.2.4 ภาษาฐานข้อมูล (Database Language)

โดยทั่วไประบบจัดการฐานข้อมูลเป็นสิ่งที่คั่นระหว่างฐานข้อมูลกับผู้ใช้ ซึ่งจะมีภาษาสำหรับฐานข้อมูลเพื่อให้ผู้ใช้เรียกค้นข้อมูล และจัดการด้านอื่น ๆ ในทางทฤษฎีภาษาฐานข้อมูลจะประกอบด้วย 2 ภาษา คือ

3.2.4.1 ภาษานิยามข้อมูล (Data Definition Language)

เป็นภาษาที่ใช้สร้างฐานข้อมูลทั้ง 3 ระดับดังได้กล่าวมาแล้ว นอกจากนี้ภาษานิยามข้อมูลยังสามารถกำหนดเงื่อนไขต่าง ๆ เพื่อให้ระบบจัดการฐานข้อมูลใช้ตรวจสอบความถูกต้องของข้อมูล

3.2.4.2 ภาษาจัดการข้อมูล (Data Manipulation Language)

เป็นภาษาที่ใช้ในการค้นหาข้อมูล (Retrieve) เพิ่มข้อมูล (Insert) แก้ไขข้อมูล (Update) และลบข้อมูล (Delete) การใช้ฐานข้อมูลในระดับของผู้ใช้สามารถกระทำได้โดยการใช้ภาษาจัดการข้อมูลนี้ โดยทั่วไปแล้วจะแบ่งได้เป็น 2 ประเภทคือ

3.2.4.2.1 ภาษากำหนดวิธีการ (Procedural Language) ผู้ใช้จะทำหน้าที่ระบุสิ่งที่ต้องการ และวิธีการ เพื่อให้ได้ผลที่ต้องการด้วยตนเอง

3.2.4.2.2 ภาษาไม่กำหนดวิธีการ (Nonprocedural Language) ผู้ใช้เพียงแต่ระบุสิ่งที่ต้องการ แต่ไม่ต้องบอกถึงวิธีการทำ

โดยทั่วไปแล้วภาษาเอสคิวแอล (Structured Query Language : SQL) ซึ่งเป็นภาษาที่ใช้ในการติดต่อกับระบบจัดการฐานข้อมูล นอกจากจะประกอบด้วยภาษานิยามข้อมูล และภาษาจัดการข้อมูลแล้ว ยังมีภาษาควบคุมข้อมูล (Data Control Language) เพื่อใช้ควบคุม และป้องกันการเข้าถึงข้อมูลของผู้ที่ไม่มีสิทธิ

3.2.5 วงจรระบบฐานข้อมูล (Database System Life Cycle)

การพัฒนาระบบฐานข้อมูลแบ่งการดำเนินงานเป็นหลายขั้นตอน แต่ละขั้นตอนมีรายละเอียดดังนี้

3.2.5.1 ข้อกำหนดระบบ (System Definition) เป็นการศึกษาระบบงานปัจจุบัน รายงาน งานประยุกต์ รวมถึงซอฟต์แวร์ที่ใช้ เพื่อกำหนดขอบเขต และจุดมุ่งหมายของระบบฐานข้อมูลใหม่

3.2.5.2 การออกแบบ (Design) ประกอบด้วยการออกแบบฐานข้อมูลเชิงตรรก และการออกแบบฐานข้อมูลเชิงกายภาพ (จะอธิบายรายละเอียดภายหลัง) การออกแบบที่ดีจะต้องได้ระบบที่ง่ายต่อการเปลี่ยนแปลง หรือขยายตัวของระบบในอนาคต

3.2.5.3 การติดตั้งระบบ (Implementation) เป็นกระบวนการนำข้อกำหนดของฐานข้อมูลทั้ง 3 ระดับบันทึกลงในฐานข้อมูล และรวมถึงการสร้างต้นแบบด้วย

3.2.5.4 การบรรจุ หรือการแปลงข้อมูล (Loading or Data Conversion) เป็นการนำข้อมูลเข้าสู่ระบบใหม่ โดยจะนำเข้าโดยตรงหรือทำการแปลงแฟ้มข้อมูลจากระบบเดิมเข้าสู่ระบบใหม่

3.2.5.5 การแปลงงานประยุกต์ (Application Conversion) การดัดแปลงงานประยุกต์ที่มีอยู่เดิมให้เข้าสู่ระบบใหม่ รวมถึงการสร้างงานประยุกต์ขึ้นใหม่

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3.2.5.6 การทดสอบและตรวจสอบความถูกต้อง (Testing and Validation) เป็นการทดสอบและตรวจสอบการทำงานของงานประยุกต์ รวมถึงข้อมูลว่ามีถูกต้องหรือไม่

3.2.5.7 การปฏิบัติการ (Operation) เป็นขั้นตอนรวมงานประยุกต์ทุกงานเข้าด้วยกัน และจัดการเกี่ยวกับสิทธิ รวมทั้งขอบเขตการทำงานของผู้ใช้แต่ละคน เช่น กำหนดให้ใครเป็นผู้ทำการกู้ข้อมูล และทำแฟ้มข้อมูลสำรอง เป็นต้น

3.2.5.8 การดูแลและบำรุงรักษา (Monitoring and Maintanance) ดูแลการเติบโต และการขยายตัวที่จะเกิดขึ้นกับข้อมูล และซอฟต์แวร์

3.2.6 การออกแบบฐานข้อมูล

การออกแบบฐานข้อมูลแบ่งงานออกได้เป็นหลายขั้นตอน ในแต่ละขั้นตอนบางครั้งสามารถทำควบคู่กันไป ซึ่งประกอบด้วย 6 ขั้นตอน คือ

3.2.6.1 การเก็บรวบรวมและวิเคราะห์ความต้องการ (Requirements Collection and Analysis) จะต้องพิจารณาถึงข้อมูลที่ใช้อยู่ในปัจจุบัน ข้อมูลที่ต้องการเพิ่มเติม รวมถึงข้อมูลที่จะเกิดขึ้นในอนาคต โดยทำการศึกษารายงาน ผังงาน โปรแกรมประยุกต์ สัมภาษณ์และสอบถาม ฯลฯ คำถามอย่างหนึ่งที่สำคัญในการสอบถามคือ ข้อมูลนั้น ๆ มีความจำเป็นต้องใช้ในระบบหรือไม่ และมีข้อมูลใดที่ยังขาด และต้องการเพิ่มเติม

3.2.6.2 การออกแบบฐานข้อมูลเชิงมโนภาพ (Conceptual Database Design) ผู้ออกแบบสามารถพัฒนารายละเอียดของข้อมูลที่ได้จากขั้นตอนที่ 1 โดยการใช้โมเดลข้อมูลเชิงตรรก (Logical Data Model) ซึ่งสามารถกำหนดเอนติตี (Entity) รีเลชันชิป (Relationship) และ แอตตริบิว (Attribute) ได้

3.2.6.3 การเลือกระบบจัดการฐานข้อมูล (Choice of a DBMS) ต้องพิจารณาชนิดของระบบฐานข้อมูล โดยศึกษาถึงโครงสร้างในการเก็บข้อมูล ทิศทางการเข้าถึงข้อมูลของฐานข้อมูล ภาษาระดับสูงที่ใช้ ฯลฯ

3.2.6.4 การแปลงในรูปของโมเดลระบบจัดการฐานข้อมูล (Data Model Mapping บางทีเรียก Logical Database Design) เป็นการแปลงแบบแผนฐานข้อมูลเชิงมโนภาพให้อยู่ในรูปโมเดลระบบจัดการฐานข้อมูลที่เลือกแล้ว โดยแปลงจากโมเดลที่ได้ในขั้นตอนที่ 2 โดยแบ่งออกเป็น 2 ขั้นตอน คือ

3.2.6.4.1 แปลงโดยเป็นอิสระจากชนิดของระบบจัดการฐาน-ข้อมูลคือแปลงให้อยู่ในรูปของระบบจัดการฐานข้อมูลที่เลือก แต่ไม่สนใจข้อจำกัดพิเศษของระบบจัดการฐานข้อมูลนั้น

3.2.6.4.2 ปรับปรุงให้อยู่ในรูปของระบบจัดการฐานข้อมูลชนิดที่เลือก

3.2.6.5 การออกแบบฐานข้อมูลเชิงกายภาพ (Physical Database Design) เป็นกระบวนการในการออกแบบโครงสร้างการเก็บข้อมูล และทิศทางการเข้าถึงข้อมูล สำหรับแฟ้มข้อมูลของฐานข้อมูล เพื่อให้ได้ฐานข้อมูลที่มีประสิทธิภาพมากที่สุด ระบบจัดการฐานข้อมูลมีระบบจัดการแฟ้มข้อมูล ทิศทางการเข้าถึงข้อมูล ดัชนีตัวชี้ ฯลฯ ที่แตกต่างกันไป สิ่งที่ต้องพิจารณาในการออกแบบฐานข้อมูลเชิงกายภาพคือ

3.2.6.5.1 เวลาในการตอบสนอง (Response Time) คือเวลาตั้งแต่การส่งงานเข้าประมวลผล จนได้รับผลลัพธ์ที่ต้องการ

3.2.6.5.2 การใช้ที่ว่าง (Space Utilization) จำนวนพื้นที่ว่างที่จะถูกใช้งานโดยแฟ้มข้อมูลของฐานข้อมูล และโครงสร้างทิศทางการเข้าถึงข้อมูล

3.2.6.5.3 ผลงานที่ได้ออกมา (Transaction Throughput) คิดเป็นค่าเฉลี่ยคำนวณจากจำนวนงานที่ประมวลผลได้โดยระบบจัดการฐานข้อมูลต่อหนึ่งหน่วยเวลา

3.2.6.6 การติดตั้งฐานข้อมูล (Database Implementation) หลังจากขั้นตอนการแปลงให้อยู่ในรูปของโมเดลระบบจัดการฐานข้อมูล และขั้นตอนการออกแบบฐานข้อมูลเชิงกายภาพ จะทำการติดตั้งฐานข้อมูลโดยใช้คำสั่งต่าง ๆ ของระบบจัดการฐานข้อมูลที่เลือกแล้ว เช่น การสร้างตาราง สร้างความสัมพันธ์ของข้อมูล เป็นต้น และนำข้อมูลเข้าสู่ฐานข้อมูลต่อไป

3.3 โมเดลข้อมูลเชิงตรรก (Logical Data Model)[7]

เป็นโมเดลที่ใช้ในขั้นการออกแบบฐานข้อมูลเชิงมโนภาพเพื่อแสดงถึงข้อมูลในขอบเขตที่ผู้ออกแบบสนใจ โดยมีศัพท์ที่จะต้องทำความเข้าใจ คือ

1) เอนติตี คือ สิ่งที่มีอยู่จริง จับต้องได้ หรือเป็นจินตภาพที่แสดงความเป็นหนึ่งเดียว เมื่อกล่าวถึงแล้วทุกคนเข้าใจตรงกัน เช่น นิสิต ชั้นเรียน เป็นต้น

2) รีเลชันชิป คือ ความสัมพันธ์ระหว่างเอนติตีใดเอนติตีหนึ่งกับตัวเอง หรือเอนติตีอื่น อาจเป็นความสัมพันธ์ที่มากกว่า 2 เอนติตีก็ได้ เช่น อาจารย์ที่ปรึกษาของนิสิต เป็นต้น

3) แอตตริบิว คือ กลุ่มของค่าความจริงใด ๆ ที่เป็นรายละเอียดของเอนติตี ทำให้เข้าใจเอนติตีได้ลึกซึ้งยิ่งขึ้น และเป็นสิ่งที่ไม่สามารถแตกย่อยลงไปได้อีก เช่น บ้านเลขที่นิสิต เป็นต้น

3.3.1 คุณสมบัติของโมเดลข้อมูลเชิงตรรก

คุณสมบัติของโมเดลข้อมูลเชิงตรรก จะต้องมีลักษณะดังต่อไปนี้

3.3.1.1 แสดงได้ด้วยแผนภาพ (Graphical Diagrams) ไม่ว่าจะเป็นเทคนิคโมเดลข้อมูลแบบใดจะมีภาษา และรูปภาพทางกราฟฟิกโดยเฉพาะ เพื่อใช้แสดงรายละเอียด ข้อมูลกลุ่มใหญ่ และรายละเอียดส่วนย่อย ซึ่งทำให้ง่ายต่อการแปลความ เช่น ใช้วงกลม หรือสี่เหลี่ยม แทนเอนติตี ใช้เส้นโค้ง หรือเส้นตรง แทนรีเลชันชิป เป็นต้น

3.3.1.2 การแสดงความหมายข้อมูลชัดเจน (Explicit Representation of Semantic) แผนภาพที่ได้ออกมาควรง่าย ไม่ซับซ้อน และเห็นความหมายของข้อมูลชัดเจนโดยเฉพาะอย่างยิ่งสัญลักษณ์หนึ่ง ๆ ไม่ควรมีหลายความหมาย

3.3.1.3 แสดงรายละเอียดในระดับที่เหมาะสม (Appropriate Level of Detail) คือโมเดลเชิงตรรกจะมีรายละเอียดพอที่เห็นความแตกต่างระหว่างชนิดของข้อมูลรีเลชันชิป และข้อบังคับต่าง ๆ แต่จะน้อยกว่าโมเดลเชิงกายภาพ

3.31.4 เป็นอิสระจากระบบจัดการฐานข้อมูล (DBMS Independence) คือ โมเดลที่ได้จากการออกแบบควรใช้ได้กับระบบฐานข้อมูลหลายชนิด ได้แก่ เชิงสัมพันธ์เชิงลำดับชั้น และเชิงเครือข่าย

3.3.1.5 ง่ายต่อการศึกษาและใช้งาน (Easy to Learn and Use) จะต้องมีความง่ายสำหรับผู้ใช้ทุกประเภทจะทำความเข้าใจและนำไปใช้ได้

3.3.2 ขั้นตอนการออกแบบโมเดลข้อมูลเชิงตรรก

การออกแบบโมเดลข้อมูลเชิงตรรกมีหลายขั้นตอนด้วยกัน จากรูปที่ 3.2 ใน 5 ขั้นตอนแรกเป็นการออกแบบทางด้านโครงสร้างพื้นฐานของโมเดล ได้แก่ เอนติตี รีเลชัน-ชิป คีย์หลัก คีย์รอง คีย์ภายนอก กฎเกณฑ์พื้นฐาน ขั้นตอนต่อไปเป็นการเพิ่มรายละเอียดใน

ระดับที่ผู้ใช้มองเห็น (User View) และรวมรายละเอียดเหล่านั้นเข้าด้วยกัน จึงจะได้เป็นโมเดลข้อมูลเชิงตรรกที่สมบูรณ์ ขั้นตอนต่าง ๆ มีรายละเอียดดังนี้

ขั้นตอนการออกแบบโมเดลข้อมูลเชิงตรรก

การสร้างและกำหนดโครงสร้างในมุมมองของผู้ใช้	
LDM1	การกำหนดเอนติตีหลัก (Identify Major Entities)
LDM2	กำหนดความสัมพันธ์ระหว่างเอนติตี (Determine Relationships between Entities)
LDM3	กำหนดคีย์หลักและคีย์รอง (Determine Primary and Alternate Keys)
LDM4	กำหนดคีย์ภายนอก (Determine Foreign Keys)
LDM5	กำหนดกฎธุรกิจ (Determine Key Business Rules)

เพิ่มรายละเอียดในมุมมองของผู้ใช้	
LDM6	การเพิ่มแอตตริบิวที่เหลือ (Add Remaining Attributes)
LDM7	การตรวจสอบกฎนอร์มัลไลเซชัน (Validate Normalization Rules)
LDM8	กำหนดโดเมน (Determine Domains)
LDM9	กำหนดกฎการจัดการกับข้อมูล (Trigger Operations)

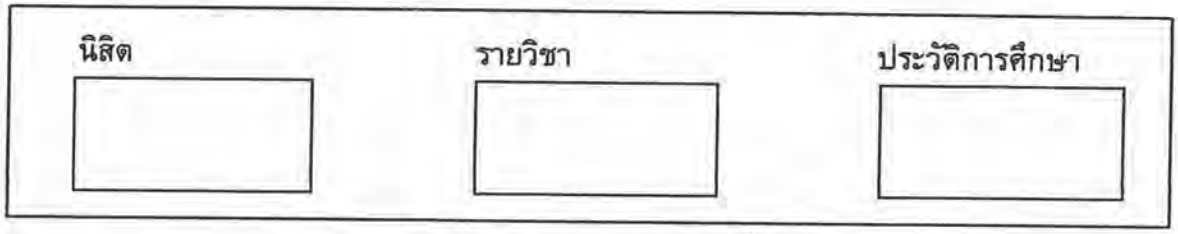
การรวมมุมมองของผู้ใช้	
LDM10	การเชื่อมมุมมองของผู้ใช้เข้าด้วยกัน(Combine User Views)
LDM11	การรวมเข้ากับโมเดลที่มีอยู่แล้ว
LDM12	วิเคราะห์เสถียรภาพและการเติบโตในอนาคต

รูปที่ 3.2 ขั้นตอนการออกแบบโมเดลข้อมูลเชิงตรรก

3.3.2.1 การกำหนดเอนติตีหลัก

เป็นการกำหนดชื่อ ความหมายของเอนติตีลงในพจนานุกรมข้อมูล และบันทึกลงโมเดลข้อมูล การตั้งชื่อไม่ควรเกิน 20 ตัวอักษร ดังแสดงในรูปที่ 3.3

การกำหนดเอนติตีนั้นเป็นงานที่ยาก จะต้องอาศัยความร่วมมือของผู้ที่เข้าใจระบบที่เราออกแบบเพื่อคัดเลือกสิ่งที่ถูกต้อง มีความสำคัญ และเหมาะสมที่สุดมาเป็นเอนติตี วิธีการอย่างคร่าว ๆ ในการกำหนดเอนติตีหลักคือ ให้พิจารณาข้อมูลทั้งหมดและจัดกลุ่มของข้อมูลโดยดูจากค่าและความหมาย ถ้าสามารถรวมกลุ่มกันได้ก็ให้รวมเข้าไว้ในเอนติตีเดียวกัน



รูปที่ 3.3 ตัวอย่างเอนติตีหลัก

3.3.2.2 การกำหนดรีเลชันชิประหว่างเอนติตี

กำหนดชื่อ ความหมาย ทิศทาง และขนาดอัตราส่วนที่เกิดรีเลชันชิปนั้น ๆ พร้อมทั้งบันทึกลงพจนานุกรมข้อมูล สำหรับชื่อก็เช่นเดียวกับชื่อเอนติตีคือ ไม่ควรเกิน 20 ตัวอักษร สำหรับความหมายถ้าเป็นภาษาอังกฤษให้ใช้คำกริยาในรูปแบบปัจจุบัน (Present Tense) รีเลชันชิปที่เกิดขึ้นเป็นไปได้ 3 อย่างด้วยกัน คือ

3.3.2.2.1 รีเลชันชิปที่เป็นเหตุการณ์ที่มีอยู่จริง เช่น ประวัตินิสิต เป็นต้น

3.3.2.2.2 รีเลชันชิปที่เป็นหน้าที่ เช่น นิสิตต้องเรียนรายวิชา เป็นต้น

3.3.2.2.3 รีเลชันชิปจากเหตุการณ์ เช่น นิสิตลงทะเบียนเรียน เป็นต้น

อัตราส่วนและทิศทางของรีเลชันชิป เป็นพื้นฐานในการแบ่งประเภทความสัมพันธ์ สำหรับรีเลชันชิประหว่าง 2 เอนติตี (Binary Relationship) จะมีด้วยกัน 3 แบบ คือ

1) หนึ่งต่อหนึ่ง (One-to-One, 1:1 Relationship)

รีเลชันชิปที่เกิดขึ้นนี้จะมีมากที่สุดแค่หนึ่งเท่านั้น เช่น คณะหนึ่งจะมีคณบดีเพียงคนเดียว และในทางกลับกันอาจารย์ท่านหนึ่งก็เป็นคณบดีของคณะได้คณะเดียวเท่านั้น ดังนั้นรีเลชันชิปแบบ

นี้ทำให้ต้องตัดสินใจว่าจะให้หัวลูกศรไปทางใด ซึ่งเมื่อกำหนดได้แล้วเอนติตีใดที่หัวลูกศรชี้เข้าหาจะเรียกว่า เอนติตีลูก (Child Entity) อีกเอนติตีหนึ่งจะเรียกว่าเอนติตีแม่ (Parent Entity) ใช้สัญลักษณ์ (P) ⟶ (C) แทน ดังรูปที่ 3.4

รูปที่ 3.4 รีเลชันชิปแบบหนึ่งต่อหนึ่ง

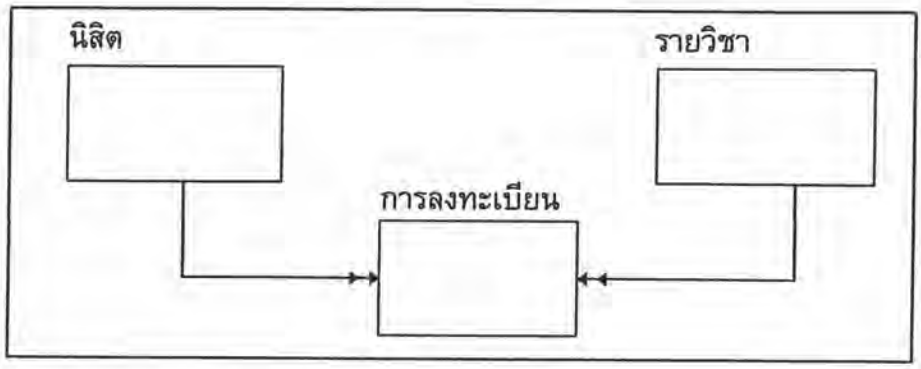
2) หนึ่งต่อหลาย (One-to-Many, 1:N Relationship)
รีเลชันชิปจะเกิดขึ้นได้ตั้งแต่ 0 ครั้ง จนถึงหลาย ๆ ครั้ง โดยแต่ละรีเลชันชิปของเอนติตีลูกจะสัมพันธ์กับเอนติตีแม่ได้หนึ่งค่าเท่านั้น แต่ค่าของเอนติตีแม่จะสัมพันธ์กับเอนติตีลูกได้หลายค่า ใช้สัญลักษณ์ (P) ⟶→ (C) แทน เช่น อาจารย์ท่านหนึ่งมีลูกศิษย์ในที่ปรึกษาได้หลายคน แต่นิสิตจะมีอาจารย์ที่ปรึกษาได้คนเดียวเท่านั้น ดังรูปที่ 3.5

รูปที่ 3.5 รีเลชันชิปแบบหนึ่งต่อหลาย

3) หลายต่อหลาย (Many-to-Many, M:N Relationship)
รีเลชันชิปที่เกิดขึ้นได้ตั้งแต่ 0 ครั้ง จนถึงหลาย ๆ ครั้งทั้งสองทิศทาง กล่าวคือแต่ละเอนติตีลูกจะสัมพันธ์กับเอนติตีแม่ได้หลายค่า และเอนติตีแม่ก็สัมพันธ์กับเอนติตีลูกได้หลายค่าเช่นกัน ใช้สัญลักษณ์ (P) ⟷→ (C) แทน เช่น นิสิตแต่ละคนเลือกรายวิชาเรียนได้หลายวิชา และรายวิชาแต่ละวิชาจะถูกเลือกเรียนโดยนิสิตได้หลายคน ดังรูปที่ 3.6

รูปที่ 3.6 รีเลชันชิปแบบหลายต่อหลาย

หลังจากแบ่งกลุ่มรีเลชันชิประหว่างเอนติตีแล้ว ความสัมพันธ์แบบหลายความสัมพันธ์ (M:N) เป็นสิ่งที่ต้องสนใจมากที่สุดเพราะเป็นสิ่งที่ยุ่งยาก โดยปกติฐานข้อมูลทั่วไปจะไม่สามารถจัดการกับรีเลชันชิปแบบนี้ได้ ดังนั้นจึงต้องแตกรีเลชันชิปออกมาเป็นรีเลชันชิปแบบ (1:N) สองรีเลชันชิป โดยการกำหนดเอนติตีใหม่ขึ้นมาอีกหนึ่งตัวให้มีรีเลชันชิปกับเอนติตีเดิมแบบ (1:N) เช่น รีเลชันชิประหว่างนิสิตกับรายวิชา แตกได้โดยเพิ่มเอนติตีการลงทะเบียนขึ้นมา ดังรูปที่ 3.7 จากนั้นให้บันทึกชื่อ ความหมาย รายละเอียดของเอนติตี และรีเลชันชิปที่เกิดขึ้นใหม่นี้ลงในพจนานุกรมข้อมูล

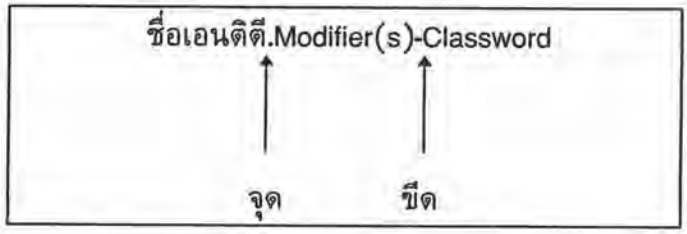


รูปที่ 3.7 รีเลชันชิปแบบ (M:N) ที่ถูกกำหนดขึ้นเป็นเอนติตีใหม่

3.3.2.3 การกำหนดคีย์หลัก และคีย์รอง

คีย์หลัก และคีย์รอง จะเป็นแอตตริบิวตัวแรกที่ถูกกำหนดในเอนติตี โดยเลือกจากแอตตริบิวที่สามารถใช้อ้างอิงถึงข้อมูลที่ต้องการได้ ซึ่งประกอบด้วยแอตตริบิว 1 ตัว หรือมากกว่า 1 ตัวก็ได้ และกำหนดคีย์รอง ซึ่งจะมีคุณสมบัติเหมือนคีย์หลักทุกประการ แต่ไม่ได้ถูกเลือกให้เป็นคีย์หลัก หลังจากนั้นให้ตั้งชื่อ และนำบันทึกไว้ในโมเดลข้อมูลเชิงตรรก พร้อมทั้งบันทึกลงในพจนานุกรมข้อมูลด้วย

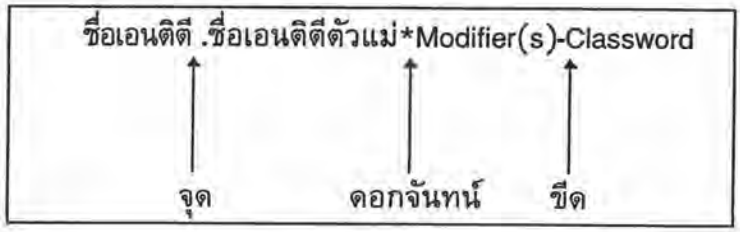
การตั้งชื่อควรกำหนดให้สั้น และง่าย อาจใช้ชื่อย่อได้ ควรหลีกเลี่ยงการตั้งชื่อแอตตริบิวของสองสิ่งที่ไม่เหมือนกันด้วยชื่อเดียวกัน รูปแบบการกำหนดอาจทำได้ดังรูปที่ 3.8 เช่น Employee.Last-Name



รูปที่ 3.8 การตั้งชื่อคีย์หลัก และคีย์รอง

3.3.2.4 การกำหนดคีย์ภายนอก

คีย์ภายนอก คือ แอตตริบิวของเอนติตีหนึ่งที่ซ้ำกับแอตตริบิวในอีกเอนติตีหนึ่งซึ่งเป็นแอตตริบิวที่ทำให้เกิดความสัมพันธ์ระหว่างเอนติตี และใช้เป็นส่วนควบคุมความถูกต้องของข้อมูล หลังจากกำหนดคีย์ภายนอกได้แล้วให้ตั้งชื่อ และนำบันทึกไว้ในโมเดลข้อมูลเชิงตรรก พร้อมทั้งบันทึกลงในพจนานุกรมข้อมูลด้วย รูปแบบการกำหนดอาจทำได้ดังรูปที่ 3.9 เช่น Student.Race*Code



รูปที่ 3.9 การตั้งชื่อคีย์ภายนอก

3.3.2.5 การกำหนดกฎธุรกิจ

เป็นการกำหนดกฎเกณฑ์ต่าง ๆ เพื่อตรวจสอบความถูกต้องและความสอดคล้องกันของข้อมูล ซึ่งแบ่งกฎธุรกิจแบ่งได้เป็น 3 ประเภทคือ

3.3.2.5.1 กฎธุรกิจของคีย์ (Key Business Rules) เป็นการกำหนดกฎเพื่อความสมบูรณ์ของรีเลชันชิป กฎต่าง ๆ จะมีผลต่อการจัดการบนรีเลชันชิป อันได้แก่ การเพิ่ม การลบ และการแก้ไขข้อมูล เช่น กำหนดผลของการเปลี่ยนค่าของคีย์หลัก เป็นต้น

3.3.2.5.2 โดเมน (Domain) เป็นการกำหนดกฎเพื่อความสมบูรณ์ของแอตตริบิวคือ การกำหนดข้อบังคับ และค่าที่เป็นไปได้สำหรับแอตตริบิว

3.3.2.5.3 กฎการจัดการ (Triggering Operation) เป็นการกำหนดผลกระทบจากการเพิ่ม ลบ แก้ไข หรือดึงข้อมูลที่เกิดกับเอนติตีอื่น หรือแอตตริบิวอื่นภายในเอนติตีเดียวกัน

สำหรับขั้นตอนนี้จะพิจารณาเฉพาะกฎธุรกิจของคีย์ การกำหนดกฎการเพิ่มข้อมูลมักจะมีผลเมื่อทำการเพิ่มข้อมูลในเอนติตีตัวลูก สำหรับกฎการแก้ไขและลบข้อมูลมักจะมีผลเมื่อกระทำกับเอนติตีตัวแม่ ดังมีรายละเอียดดังนี้

1) การกำหนดกฎการเพิ่มข้อมูล มี 6 รูปแบบด้วยกัน คือ

- แบบขึ้นต่อกัน (Dependent) จะอนุญาตให้เพิ่มข้อมูลในเอนติตีลูกเฉพาะที่มีความสัมพันธ์กับเอนติตีแม่เท่านั้น กล่าวคือค่าคีย์ภายนอกในเอนติตีลูกจะต้องเป็นค่าที่หาได้ในเอนติตีแม่

- แบบอัตโนมัติ (Automatic) ในการเพิ่มข้อมูลในเอนติตีลูก ถ้าหาค่าในเอนติตีแม่ที่ตรงกันไม่ได้ ให้เพิ่มเข้าในเอนติตีแม่ด้วยโดยอัตโนมัติ

- แบบกำหนดเป็นนัล (Nullify) เมื่อหาค่าในเอนติตีแม่ที่ตรงกันไม่ได้ให้กำหนดค่าคีย์ภายนอกในเอนติตีลูกเป็นนัล

- แบบมีค่ากำหนดไว้ (Default) มีการกำหนดค่าหนึ่งไว้ในกรณีหาค่าในเอนติตีแม่ที่ตรงกันไม่ได้ให้กำหนดคีย์ภายนอกเป็นค่าที่กำหนดไว้

- แบบมีธรรมเนียม (Customized) จะมีข้อกำหนดไว้ก่อน คือ จะยอมให้เพิ่มข้อมูลในเอนติตีลูกได้เมื่อตรวจสอบแล้วตรงกับข้อกำหนดนั้น ๆ ก่อน

- แบบไม่มีผลใด ๆ (No Effect) เพิ่มข้อมูลในเอนติตีลูกได้โดยไม่มีเงื่อนไขใด ๆ

2) การกำหนดกฎในการลบ หรือแก้ไขข้อมูล มี 6 รูปแบบคือ

- แบบมีข้อจำกัด จะอนุญาตให้ลบ หรือแก้ไขข้อมูลในเอนติตีแม่ได้ ถ้าในเอนติตีลูกไม่มีคีย์ภายนอกที่อ้างถึงเอนติตีแม่

- แบบต่อเนื่อง ถ้าในเอนติตีลูกมีการอ้างถึงเอนติตีแม่ การลบ หรือแก้ไข ให้ลบ หรือแก้ไขทั้งเอนติตีแม่ และเอนติตีลูก

- แบบกำหนดเป็นนัล ถ้าในเอนติตีลูกมีการอ้างถึงเอนติตีแม่ให้ เปลี่ยนค่าคีย์ภายนอกในเอนติตีลูกเป็นนัล แล้วจึงลบ หรือแก้ไขเอนติตีแม่

- แบบกำหนดค่า ถ้าในเอนติตีลูกมีการอ้างถึงเอนติตีแม่ให้เปลี่ยนค่าคีย์ภายนอกในเอนติตีลูกเป็นค่าที่กำหนดไว้

- แบบมีธรรมเนียม จะลบ หรือแก้ไขข้อมูลในเอนติตีแม่ได้ต่อเมื่อเงื่อนไขถูกต้อง

- แบบไม่มีผลใด ๆ สามารถลบหรือแก้ไขข้อมูลในเอนติตีแม่ โดยไม่มีการตรวจสอบ

3.3.2.6 การเพิ่มแอตตริบิวที่เหลือ

หลังจากกำหนดคีย์หลัก คีย์รอง และคีย์ภายนอกให้กับเอนติตีแล้ว ให้ทำการกำหนดแอตตริบิวอื่น ๆ ที่เหลือให้กับเอนติตี โดยมีเงื่อนไขคือจะต้องเป็นแอตตริบิวที่ขึ้นกับคีย์หลักเท่านั้น และยังคงทำให้คีย์หลักมีค่าไม่ซ้ำกัน หลังจากนั้นตั้งชื่อแอตตริบิวและบันทึกลงในโมเดลข้อมูลเชิงตรรก พร้อมทั้งบันทึกลงในพจนานุกรมข้อมูลด้วย

3.3.2.7 การตรวจสอบด้วยกฎของนอร์มัลไลเซชัน

ให้ทำการตรวจสอบเอนติตีต่าง ๆ ให้อยู่ในกฎนอร์มัลไลเซชัน (จะอธิบายรายละเอียดภายหลัง) ซึ่งประกอบด้วย 1NF, 2NF, 3NF, BCNF, 4NF, 5NF เพื่อประโยชน์ดังต่อไปนี้

3.3.2.7.1 ลดที่ว่างที่ต้องใช้ในการเก็บข้อมูล

3.3.2.7.2 ลดความผิดพลาด ความไม่ตรงกันของข้อมูลในฐานข้อมูล

3.3.2.7.3 ลดการเกิดอะนอร์มัลไลของการลบ และแก้ไขข้อมูล

3.3.2.7.4 เพิ่มความคงทนแก่โครงสร้างฐานข้อมูล

3.3.2.8 การกำหนดโดเมนของแอตตริบิว

โดเมน คือ กลุ่มค่าที่ถูกต้องเป็นไปได้สำหรับแอตตริบิวแต่ละแอตตริบิว โดยกำหนดโดเมนให้กับแอตตริบิวในเอนติตี พร้อมทั้งบันทึกในพจนานุกรมข้อมูล ซึ่งประกอบ ด้วยรายละเอียดดังนี้

3.3.2.8.1 ชนิดของข้อมูล (Data Type) เช่น จำนวนเต็ม วันที่ ตัวอักษร ทศนิยม เป็นต้น

3.3.2.8.2 ความยาว (Length) เช่น 5 หลัก 35 ตัวอักษร เป็นต้น

3.3.2.8.3 รูปแบบข้อมูล (Format) เช่น dd/mm/yy (วันที่) ccc-cccc (เบอร์โทรศัพท์) เป็นต้น

3.3.2.8.4 ค่าที่อนุญาต (Allowable Value) เช่น วันศุกร์ต้นเดือน เป็นต้น

3.3.2.8.5 ช่วงของข้อมูลหรือข้อกำหนดอื่น ๆ (Constraints, Rang) เช่น 2-50 วัน เป็นต้น

3.3.2.8.6 ความหมาย (Meaning) อธิบายความหมายของแอตตริบิวนั้นว่าคืออะไร

3.3.2.8.7 ความเป็นหนึ่งเดียว (Uniqueness) ต้องมีค่าเป็นหนึ่งเดียว

3.3.2.8.8 การเป็นนัล (Null Support) อนุญาตให้เป็นนัลได้หรือไม่

3.3.2.8.9 ค่าโดยปริยาย (Default Value) กำหนดให้มีค่าเป็นศูนย์

3.3.2.9 กำหนดกฎการจัดการกับข้อมูล

เป็นกฎเพื่อตรวจสอบความถูกต้องเมื่อเราทำการ เพิ่ม ลบ แก้ไข หรือดึงข้อมูล โดยจะพิจารณาผลของการกระทำที่เกิดกับเอนติตี และแอตตริบิวที่ถูกกระทำ เมื่อกำหนดกฎการจัดการต่าง ๆ แล้ว ให้บันทึกลงในพจนานุกรมข้อมูลโดยมีรูปแบบที่ประกอบด้วย เหตุการณ์ที่ทำ เช่น เพิ่ม ลบ ฯลฯ ในเอนติตี หรือแอตตริบิวที่ถูกกระทำ เงื่อนไขที่กำหนดไว้ การกระทำที่จะต้องเกิดสืบเนื่องจากเหตุการณ์นั้น เป็นต้น

3.3.2.10 การเชื่อมมุมมองผู้ใช้ทุกคนเข้าด้วยกัน

วัตถุประสงค์คือ แสดงมุมมองที่ละเอียดขึ้นด้วยโมเดลข้อมูลที่ซับซ้อน เพื่อลดความซ้ำซ้อน และความไม่สอดคล้องกันของข้อมูล โดยอาจมีการรวมเอนติตี รวมความสัมพันธ์ และรวมแอตตริบิว

3.3.2.10.1 ข้อควรพิจารณาในการรวมเอนติตี

1) การรวมเอนติตีที่มีคีย์หลักเหมือนกัน จะต้องได้เอนติตีใหม่ที่มีแอตตริบิวรวมของสองเอนติตีเดิม

2) ถ้าเอนติตีสองเอนติตีมีคีย์หลักเดียวกัน แต่ค่าที่เป็นไปได้ของคีย์หลักนั้นเป็นซับเซตของกันจะรวมได้ในรูปแบบของซับไทป์-ซูเปอร์ไทป์ โดยต้องตัดแอตตริบิวที่มีแล้วในเอนติตีซูเปอร์ไทป์ออกจากเอนติตีซับไทป์

3) ถ้าเอนติตีสองเอนติตีมีคีย์หลักเดียวกัน แต่มีผลไปกำหนดแอตตริบิวต่างกันบางแอตตริบิว ให้กำหนดซูเปอร์ไทป์ขึ้นมาอันหนึ่งสัมพันธ์กับสองเอนติตีเดิม

4) การเชื่อมเอนติตีสองเอนติตีที่คีย์หลักของเอนติตีหนึ่งเป็นคีย์รองของอีกเอนติตีหนึ่ง จะได้เอนติตีใหม่ที่มีคีย์หลักตามเอนติตีตัวที่หนึ่ง ส่วนคีย์หลักของเอนติตีอีกตัวจะกลายเป็นคีย์รอง และมีแอตตริบิวรวมระหว่างสองเอนติตีเดิม แล้วตัดแอตตริบิวที่ซ้ำซ้อนออก และต้องกำหนดกฎธุรกิจให้ด้วยตามข้อบังคับเก่า เช่น ในกรณีคีย์หลักเดิมที่กลายมาเป็นคีย์รองในเอนติตีใหม่ยังคงห้ามเป็นนัล

3.3.2.10.2 ข้อควรพิจารณาในการรวมรีเลชันชิป

1) ให้รวมรีเลชันชิประหว่างเอนติตีที่ให้ความหมายเหมือนกันเข้าด้วยกัน โดยถ้ามีผลทำให้เกิดรีเลชันชิปแบบ M:N จะต้องทำการแตกให้เป็นรีเลชันชิปแบบ (1:N) สองเอนติตี

2) การรวมรีเลชันชิปใด ๆ ต้องไม่กระทบกับรีเลชันชิปอื่นที่ไม่ต้องการเปลี่ยนแปลง นอกจากจะพิจารณาแล้วว่าควรตัดออกเนื่องจากความซ้ำซ้อน หรืออาจเพิ่มขึ้นใหม่เพื่อความเหมาะสม

3) การรวมเอนติตีที่มีคีย์หลักเป็นคีย์รองของเอนติตีอีกตัวให้ตรวจสอบคีย์ภายนอกของเอนติตีอื่น ๆ ที่อ้างมาถึงว่าได้อ้างถึงคีย์หลัก หรือคีย์รองของเอนติตีใหม่ที่ได้จากการรวมนั้น ถ้าอ้างถึงคีย์รองต้องทำการเปลี่ยนให้เป็นคีย์หลัก

4) เมื่อรวมมุมมองต่าง ๆ แล้ว ให้กำหนดกฎธุรกิจของคีย์สำหรับรีเลชันชิปใหม่ด้วย

3.3.2.10.3 ข้อควรพิจารณาในการรวมแอตตริบิว

1) ให้ทำการรวมแอตตริบิวที่มีความหมายเหมือนกันภายในเอนติตีเดียวกัน และรวมค่าที่เป็นไปได้ รวมถึงกฎการจัดการเข้าด้วยกันด้วย และพิจารณาค่าที่เป็นไปได้ของแอตตริบิวอื่นว่าเปลี่ยนไปหรือไม่

2) เมื่อรวมเอนติตีแล้วให้พิจารณาตัดแอตตริบิวที่เป็นดิไรฟว์ หรือแฟล็กซ์ที่ไม่จำเป็นออก

3) หลังจากได้รวม ตัด หรือเพิ่มรีเลชันชิป แล้วให้ทำการนอร์มัลไลซ์อีกครั้ง เพื่อตัดสิ่งที่ซ้ำซ้อนออกเสีย

3.3.2.11 การรวมโมเดลเข้ากับโมเดลข้อมูลที่มีอยู่เดิม

เป็นขั้นตอนที่เกี่ยวข้องกับแบบแผนเชิงมโนภาพ โดยจะรวมโมเดลข้อมูลเชิงตรรกที่ได้กับโมเดลข้อมูลที่มีอยู่เดิม และพัฒนาโมเดลใหม่ควบคู่ไปกับการพิจารณากฎเกณฑ์ข้อบังคับของเดิม โดยอาจใช้เอนติตีหรือรีเลชันชิปร่วมกับของเดิม หรือมีการกำหนดเอนติตีขึ้นใหม่

3.3.2.12 การวิเคราะห์ความมีเสถียรภาพและการเติบโตในอนาคต

การออกแบบโมเดลที่ทำผ่านมา จะพิจารณาข้อมูลที่เห็นได้ในปัจจุบันเป็นส่วนมาก สำหรับขั้นนี้ให้พิจารณาถึงสิ่งที่อาจเกิดขึ้น หรือเป็นได้ในอนาคต เช่น

3.3.2.12.1 อาจมีเอนติตีหรือรีเลชันชิปใหม่เกิดขึ้นทำให้ต้องเพิ่มคีย์ภายนอกในเอนติตีเดิม

3.3.2.12.2 รีเลชันชิปแบบ 1:N อาจเปลี่ยนเป็น M:N ได้

3.3.2.12.3 คีย์หลักอาจเปลี่ยนไปเนื่องจากไม่เป็นหนึ่งเดียว

3.4 หลักการนอร์มัลไลเซชัน (Normalization) [7]

นอร์มัลไลเซชัน เป็นกระบวนการเพื่อพัฒนาการเชื่อมต่อของข้อมูล และตอบคำถามว่า การออกแบบฐานข้อมูลทั้งเชิงตรรก และเชิงกายภาพใช้ได้หรือไม่ ก่อนจะอธิบายถึงวิธีทำนอร์มัลไลเซชัน เราควรทำความเข้าใจถึงคำศัพท์ต่าง ๆ ที่เกี่ยวข้อง ได้แก่

1) ความสัมพันธ์ (Relation) คือกลุ่มของทุเปิล (Tuple เทียบได้กับระเบียนข้อมูล) โดยมากแสดงในรูปแบบของตาราง แต่ละแถวของตารางคือ 1 ทุเปิล แต่ละทุเปิลจะมีค่าต่าง ๆ บรรจุอยู่ ดังแสดงในรูปที่ 3.10 ความสัมพันธ์จะประกอบด้วย ชื่อความสัมพันธ์ (เอนติตี) [1], ทุเปิล หรือโรล (rows) [2], แอตตริบิว [3], ค่าของแอตตริบิว [4], และคีย์ของความสัมพันธ์ [5]

2) คีย์ คือแอตตริบิวตั้งแต่ 1 ตัวขึ้นไป ทำหน้าที่ค้นหาทุเปิลที่ต้องการในความสัมพันธ์คีย์หนึ่งจะระบุถึงทุเปิลหนึ่งเท่านั้น ดังรูปที่ 3.10 คีย์ของความสัมพันธ์คือ รหัสนักศึกษา

(1) คีย์หลัก (Primary Key) คือแอตตริบิวซึ่งทุกแอตตริบิวในความสัมพันธ์ต้องขึ้นกับคีย์หลัก (Dependence) หรือกลุ่มของแอตตริบิวที่เป็นหนึ่งเดียว (Unique) และทุกทุเปิลที่จะเพิ่มเข้าไปในความสัมพันธ์อย่างน้อยแอตตริบิวนี้ต้องมีค่า

(2) คีย์ประกอบ (Compound Key) คือคีย์ที่ประกอบด้วยแอตตริบิวมากกว่า 1 ตัวขึ้นไป

(3) คีย์รอง (Alternate Key) คือแอตตริบิวที่มีคุณสมบัติเหมือนคีย์หลักทุกประการ แต่ไม่ได้รับเลือกเป็นคีย์หลัก

(4) คีย์เป็นได้ (Candidate Key) คือแอตตริบิวที่มีคุณสมบัติที่สามารถเป็นคีย์หลักได้

นิสิต [5]

[1]	รหัส	ชื่อ	ที่อยู่	วันเกิด	ภูมิลำเนา	เบอร์โทรศัพท์
[2]	111	สมใจ	บางเขน	2/05/08	เชียงใหม่	4534123 [4]
[2]	112	อุกฤษณ์	นนทบุรี	3/04/10	กรุงเทพฯ	5230555 [4]

[3]

รูปที่ 3.10 ความสัมพันธ์นิสิต

การนอร์มัลไลเซชันแบ่งออกเป็นหลายระดับ ได้แก่ แบบไม่นอร์มัลไลซ์, แบบนอร์มัลฟอร์มระดับที่ 1 (1NF), ..., จนถึงแบบนอร์มัลฟอร์มระดับที่ 5 (5NF) ซึ่งถ้าความสัมพันธ์ใดเป็นได้ถึง 5NF เรียกว่า เป็นการนอร์มัลไลซ์ที่สมบูรณ์ (Fully Normalized) และจะเป็นความ

สัมพันธ์ที่ง่ายต่อการทำงาน พื้นฐานความคิดของการทำนอร์มัลไลเซชันจะพิจารณาการขึ้นต่อกันของแอตตริบิว ดังนี้คือ

1) การขึ้นต่อกันทั้งหมด (Functional Dependency) ถ้าในความสัมพันธ์ R มีแอตตริบิว A และ B จะกล่าวว่าแอตตริบิว B เป็นการขึ้นต่อกันทั้งหมดกับแอตตริบิว A ก็ต่อเมื่อแต่ละค่าของ A ในความสัมพันธ์ R จะสัมพันธ์กับแอตตริบิว B เพียงหนึ่งค่าเท่านั้น และจะเรียก A ว่า ดีเทอร์มิแนน (Determinant) กล่าวคือทุกทุเปิลในความสัมพันธ์ R ที่มีค่าของ B ตรงกัน จะต้องมีค่าของ A ตรงกันด้วย เขียนแทนด้วย A ---> B และจะกล่าวว่า แอตตริบิว A ขึ้นต่อกันทั้งหมดกับคีย์ประกอบ [K1,K2] หรือเขียนแทนด้วย [K1,K2 ==> A]

เมื่อ K1,K2 ---> A และ K1 ---/---> A

และ K2 ---/---> A

2) การไม่ขึ้นตรงกับคีย์หลัก (Transitively Dependency) ถ้าในความสัมพันธ์ R มีคีย์หลักคือ K และแอตตริบิว A และ B จะกล่าวว่าแอตตริบิว B ไม่ขึ้นตรงกับคีย์หลักเมื่อ

K ---> A และ A ---/---> K และ A ---> B

3) การขึ้นต่อกันหลายค่า (Multivalued Dependency) ในความสัมพันธ์ R มีแอตตริบิว A, B, C เราจะกล่าวว่าแอตตริบิว B เป็นการขึ้นต่อกันหลายค่าบน A ก็ต่อเมื่อเซตของค่าของแอตตริบิว B ในความสัมพันธ์ R ที่มีความสัมพันธ์ตรงกับคู่ของ <A,C> นั้น จะไม่ขึ้นกับค่าของ C เขียนแทนด้วย A ---> B และจะมีลักษณะที่เห็นได้ดังนี้

(1) แอตตริบิว A ค่าหนึ่ง จะเป็นตัวกำหนดกลุ่มของค่าแอตตริบิว B กล่าวคือ เมื่อ 2 ทุเปิลในความสัมพันธ์ R มีค่า A เดียวกัน ไม่จำเป็นต้องมีค่า B เหมือนกัน แต่ค่าของ B จะต้องอยู่ในกลุ่มของค่า B ที่ถูกกำหนดโดย A

(2) การเปลี่ยนแปลงค่าใด ๆ ในแอตตริบิว C จะไม่มีผลกระทบต่อค่า B

(3) 2 ทุเปิลในความสัมพันธ์ R ที่มีค่า B เหมือนกัน ไม่จำเป็นต้องมีค่า A เดียวกัน

(4) ค่าของแอตตริบิว C 2 ค่าที่สัมพันธ์กับค่า A เดียวกัน จะต้องสัมพันธ์กับค่าของ B ในกลุ่มเดียวกัน และเป็นกลุ่มที่กำหนดโดยค่า A นั้น ๆ

3.4.1 ความสัมพันธ์ไม่นอร์มัลไลซ์ (Unnormalized Relations)

หมายถึงความสัมพันธ์ที่มีค่าของแอตตริบิวเป็นกลุ่ม เวลาจะเพิ่มทุเปิลหรือคอลัมภ์จะมีแอตตริบิวหลายค่า ดังเช่น เอนติตีนิสิตในรูปที่ 3.11

I 17409585

เอนดิตีนิสิต

รหัสนิสิต	ชื่อนิสิต	รหัสวิชาที่เรียน	ชื่อวิชา	รหัส อ. ที่ปรึกษา	ชื่อ อ. ที่ปรึกษา
1111111	สำรวย	171161,1711662, ...	แคล1, แคล2	1234567890112	อ. มยุรี

รูปที่ 3.11 ความสัมพันธ์แบบไม่นอร์มัลไลซ์

ข้อเสียของความสัมพันธ์แบบนี้ คือ ยากต่อการดูแล เพราะจำนวนค่าของแอตตริบิวในกลุ่มไม่คงที่ และมีความซ้ำซ้อนของข้อมูลเป็นจำนวนมาก ทำให้เกิดความผิดพลาดที่เกิดจากการเพิ่ม แก้ไข และลบข้อมูล

3.4.2 ความสัมพันธ์นอร์มัลฟอร์มระดับที่ 1 (First Normal Form : 1NF)

หมายถึงความสัมพันธ์ที่ทุกทูเปิลในความสัมพันธ์จะมีค่าของแอตตริบิวเป็นค่าเดียวโดด ๆ ไม่เป็นกลุ่ม หรืออาจไม่มีค่าเลยก็ได้ (เป็น Null) ยกเว้นแอตตริบิวที่เป็นคีย์หลัก หรือกล่าวอีกแบบหนึ่งได้ว่า แอตตริบิวที่ไม่ใช่คีย์จะต้องขึ้นต่อกันทั้งหมดกับคีย์ ดังในรูปที่ 3.12

วิธีการทำความสัมพันธ์ให้อยู่ใน 1NF จะต้องขจัดทุกทูเปิลที่มีแอตตริบิวเป็นกลุ่มออกไป (Flatten) ซึ่งจะทำให้การดูแลข้อมูลสะดวกและง่ายขึ้น แต่อย่างไรก็ตามนอร์มัลไลเซชันระดับนี้ก็ยังคงมีความซ้ำซ้อนของข้อมูล และอาจเกิดข้อผิดพลาดจากการกระทำต่าง ๆ ได้เช่นเดิม

เอนดิตีนิสิต

รหัสนิสิต	ชื่อนิสิต	รหัสวิชาที่เรียน	ชื่อวิชา	รหัส อ. ที่ปรึกษา	ชื่อ อ. ที่ปรึกษา
1111111	สำรวย	171161	แคล1	1234567890112	อ. มยุรี
1111111	สำรวย	171162	แคล2	1234567890112	อ.มยุรี
2222222	อาสา	171813	ฟิสิกส์	1234567890112	อ.มยุรี

รูปที่ 3.12 ความสัมพันธ์นอร์มัลฟอร์มระดับที่ 1

3.4.3 ความสัมพันธ์นอร์มัลฟอร์มระดับที่ 2 (Second Normal Form : 2NF)

หมายถึงความสัมพันธ์ที่อยู่ในระดับที่ 1 และทุกแอตตริบิวที่ไม่ใช่คีย์หลักต้องขึ้นตรงอย่างสมบูรณ์กับคีย์หลัก หรือกล่าวได้ว่านอร์มัลฟอร์มระดับที่ 2 จะต้องไม่มี

แอตตริบิวใดขึ้นกับบางส่วนของคีย์หลัก ดังในรูปที่ 3.13 (ก) และ (ข) จะพิจารณาได้โดยดูคีย์หลักที่เป็นคีย์ประกอบ ถ้าแอตตริบิวแต่ละอันในคีย์หลักมีความเกี่ยวข้องกันเป็นแบบหลายความสัมพันธ์ (M:N) จะพบว่าไม่อยู่ใน 2NF

เอนติตีนิสิต

รหัสนิสิต	ชื่อนิสิต	รหัส อ. ที่ปรึกษา	ชื่อ อ. ที่ปรึกษา
1111111	สำรวย	1234567890112	อ.มยุรี
2222222	อาสา	1234567890112	อ.มยุรี

(ก)

เอนติตีรายวิชา

รหัสวิชาที่เรียน	ชื่อวิชา
171161	แคล1
171162	แคล2
171813	ฟิสิกส์

(ข)

รูปที่ 3.13 ความสัมพันธ์นอร์มัลฟอร์มระดับที่ 2 (ก) ได้จากความสัมพันธ์ในรูปที่ 3.12 แล้วตัดแอตตริบิวรายวิชาออกเนื่องจากไม่ขึ้นกับรหัสนิสิต ได้ความสัมพันธ์ใหม่ดัง (ข)

วิธีทำให้อยู่ใน 2NF คือให้แยกเอนติตีที่มีแอตตริบิวที่ขึ้นกับส่วนหนึ่งของคีย์อยู่ออกเป็นเอนติตีใหม่ ซึ่งจะสามารถตัดปัญหาความซ้ำซ้อนในบางจุดลงได้ แต่ก็ยังคงมีความซ้ำซ้อนอยู่

3.4.4 ความสัมพันธ์นอร์มัลฟอร์มระดับที่ 3 (Third Normal Form : 3NF)

หมายถึงความสัมพันธ์ที่อยู่ในระดับที่ 2 และแอตตริบิวที่ไม่ใช่คีย์จะไม่มีการขึ้นต่อกัน กล่าวคือต้องไม่มีแอตตริบิวที่ไม่ขึ้นตรงกับคีย์หลัก จากเอนติตีนิสิตในรูปที่ 3.13 (ก) แยกใหม่ได้ดังรูปที่ 3.14

เอนติตีนิสิต

รหัสนิสิต	ชื่อนิสิต	รหัส อ. ที่ปรึกษา
1111111	สำรวย	1234567890112
2222222	อาสา	1234567890112

เอนติตีอาจารย์ที่ปรึกษา

รหัส อ. ที่ปรึกษา	ชื่อ อ. ที่ปรึกษา
1234567890112	อ.มยุรี

รูปที่ 3.14 ความสัมพันธ์นอร์มัลฟอร์มระดับที่ 3

3.4.5 บอยด์คอดนอร์มัลฟอร์ม (Boyce/Codd Normal Form : BCNF)

หมายถึงความสัมพันธ์ที่อยู่ใน 3NF และทุกแอตตริบิวที่เป็นดีเทอร์มิแนนจะต้องเป็นคีย์เป็นได้กล่าวคือ ทุกแอตตริบิวต้องขึ้นกับทั้งหมดของคีย์เป็นได้ไม่ใช่เพียงบางส่วน จะสังเกตได้ในเอนติตีที่มีคีย์เป็นได้เป็นคีย์ประกอบ และมีการทับซ้อนกันของคีย์นั้น ๆ ดังในรูป

ที่ 3.15 เอนติตีงานที่มอบหมาย จะเห็นว่าที่ปรึกษาคนหนึ่งจะรับผิดชอบงานเพียงงานเดียวเท่านั้น ดังนั้นจึงต้องจัดว่าแอตตริบิวชื่อที่ปรึกษาเป็นดีเทอร์มิแนนของรหัสงาน แต่ไม่ได้เป็นส่วนหนึ่งของคีย์เป็นได้ ดังนั้นจึงยังไม่อยู่ในบอยด์คอดนอร์มัลฟอร์มจึงต้องแตกออกเป็น 2 เอนติตี ดังรูปที่ 3.16 (ก) และ (ข)

เอนติตีงานที่มอบหมาย

รหัสงาน	ชื่อผู้ร่วมงาน	ชื่อที่ปรึกษา
101	สาคร	สมศรี
101	ดารณี	สมศรี
101	เจนจิรา	ธาณี
112	สาคร	เกรียงไกร

รูปที่ 3.15 เอนติตีที่ไม่อยู่ใน BCNF

เอนติตีสมาชิกผู้ทำงาน

ชื่อที่ปรึกษา	ชื่อผู้ร่วมงาน
สมศรี	สาคร
สมศรี	สาคร
ธาณี	เจนจิรา
เกรียงไกร	สาคร

(ก)

เอนติตีงานกับที่ปรึกษา

รหัสงาน	ชื่อผู้ร่วมงาน
101	สาคร
101	ดารณี
101	เจนจิรา
112	สาคร

(ข)

รูปที่ 3.16 ความสัมพันธ์แบบบอยด์คอดนอร์มัลฟอร์ม

3.4.6 ความสัมพันธ์นอร์มัลฟอร์มระดับที่ 4 (Fourth Normal Form : 4NF)

หมายถึงความสัมพันธ์ที่อยู่ใน 3NF และไม่มีการขึ้นต่อกันแบบหลายค่าอยู่ในความสัมพันธ์ ดังรูปที่ 3.17(ก) แอตตริบิวภาษาขึ้นต่อกันหลายค่ากับรหัสบุคลากร ทำให้ไม่จัดเป็น 4NF จะต้องทำการแยกเป็น 2 เอนติตี ดังรูปที่ 3.17 (ข) และ (ค)

เอนติตีบุคลากร

บุคลากร	ภาษา	งานวิจัย
1111	ไทย	002
1111	จีน	002
1111	ไทย	004
1111	จีน	004

(ก)

เอนติตีภาษาที่บุคลากรใช้ได้

บุคลากร	ภาษาที่ใช้
1111	ไทย
1111	จีน

(ข)

เอนติตีงานวิจัยของบุคลากร

บุคลากร	งานวิจัย
1111	002
1111	004

(ค)

รูปที่ 3.17 การทำให้อยู่ในรูปนอร์มัลฟอร์มระดับที่ 4

3.4.7 ความสัมพันธ์นอร์มัลฟอร์มระดับที่ 5 (Fifth Normal Form : 5NF)

หมายถึงความสัมพันธ์ที่ไม่สามารถนำไปสร้างความสัมพันธ์ใหม่ โดยการเชื่อมระหว่าง 2 ความสัมพันธ์ด้วยคีย์ต่างกันได้ โดยปกติมักจะเป็นปัญหาสำหรับคีย์หลักที่เป็นคีย์ประกอบ เป็นขั้นที่พิจารณาได้ยากและจะเกิดระเบียนข้อมูลใหม่ที่ไม่อยู่จริงขึ้นมา เมื่อนำเอนติตีมารวมกัน

3.5 ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) [7]

ดังได้กล่าวข้างต้นฐานข้อมูลเชิงสัมพันธ์เก็บข้อมูลในรูปตารางสองมิติ ซึ่งข้อมูลที่เก็บจะมีความสัมพันธ์กัน ในการวิจัยนี้ได้เลือกใช้ฐานข้อมูลเชิงสัมพันธ์ในการเก็บข้อมูลของระบบงาน เนื่องจากฐานข้อมูลเชิงสัมพันธ์มีข้อดีหลายประการคือ

1) ง่ายต่อการเข้าใจ มีรูปแบบที่ไม่ซับซ้อนมากนัก

2) สามารถที่จะค้นพบปัญหาที่เกิดขึ้นในการออกแบบฐานข้อมูลได้โดยง่าย และสะดวกในการแก้ไขข้อผิดพลาด

3) นิยมใช้กันมากเป็นเครื่องคอมพิวเตอร์หลายระดับ

ในที่นี้ขออธิบายโครงสร้างของฐานข้อมูลเชิงสัมพันธ์ และขั้นตอนการออกแบบฐานข้อมูลเชิงสัมพันธ์ รวมถึงลักษณะของฐานข้อมูลเชิงสัมพันธ์ที่สมบูรณ์ ดังมีรายละเอียดดังนี้

3.5.1 โครงสร้างของฐานข้อมูลเชิงสัมพันธ์ ประกอบด้วย

3.5.1.1 คอลัมภ์ คือ คุณสมบัติของข้อมูลเมื่อเปรียบเทียบกับโมเดลข้อมูลเชิงตรรกคือแอตตริบิว

3.5.1.2 แถว คือ ค่าของข้อมูลในตาราง

3.5.1.3 ตารางความสัมพันธ์ เป็นตาราง 2 มิติที่

3.5.1.3.1 ต้องไม่มีคอลัมภ์ที่ซ้ำกันในตาราง

3.5.1.3.2 ค่าของข้อมูลในคอลัมภ์ต้องมีคุณสมบัติเหมือนกัน

3.5.1.3.3 ข้อมูลในตารางจะต้องไม่มีข้อมูลแถวใดที่ซ้ำกัน

3.5.1.3.4 ลำดับของคอลัมภ์ไม่มีความสำคัญ

3.5.1.3.5 ลำดับของข้อมูลแต่ละแถวไม่มีความสำคัญ

3.5.1.3.6 ชื่อคอลัมภ์ในตารางต้องไม่ซ้ำกัน

3.5.2 ขั้นตอนการออกแบบฐานข้อมูลเชิงสัมพันธ์ มีขั้นตอนที่สำคัญดังนี้

RDD 1 เปลี่ยนเอนติตีของโมเดลข้อมูลเชิงตรรกให้อยู่ในรูปของตารางโดย 1 เอนติตี ต่อ 1 ตาราง

RDD 2 แปลง 1 แอตตริบิวของโมเดลข้อมูลเชิงตรรกเป็น 1 คอลัมภ์

RDD 3 ปรับโครงสร้างของโมเดลที่ได้จาก RDD 1 และ RDD 2 ให้เหมาะกับการติดตั้งระบบตามผลิตภัณฑ์ที่เลือกใช้

RDD 4 ออกแบบกฎการจัดการข้อมูลกับเอนติตี

RDD 5 ออกแบบกฎการจัดการข้อมูลกับความสัมพันธ์

RDD 6 ออกแบบกฎการจัดการข้อมูลกับแอตตริบิวที่เหลือ

3.5.3 ลักษณะของฐานข้อมูลเชิงสัมพันธ์ที่สมบูรณ์

นักคณิตศาสตร์ชื่อคอดค์ (E.F. Codd) ได้เสนอรากฐานของระบบฐานข้อมูลเชิงสัมพันธ์ว่า คือ การรวบรวมตารางแบบต่าง ๆ ที่สามารถกระทำการใด ๆ บน คณิตศาสตร์ของตรรกได้ และตารางนี้จะต้องเป็นไปตามข้อบัญญัติดังนี้ (Fidelity Rules)

กฎข้อที่ 0 (Foundation Rule) ระบบใด ๆ จะประกาศว่า เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์จะต้องจัดการกับฐานข้อมูลทั้งหมด ด้วยความสามารถในลักษณะที่เป็นตาราง

กฎข้อที่ 1 (Information Rule) ข้อมูลทั้งหมดในฐานข้อมูลเชิงสัมพันธ์จะต้องถูกแสดงในระดับตรรกด้วยวิธีการเดียวเท่านั้น โดยลักษณะของตารางต่าง ๆ เช่น ชื่อตาราง ชื่อคอลัมภ์ ชนิดของข้อมูลในแต่ละคอลัมภ์ เป็นต้น

กฎข้อที่ 2 (Guaranteed Access Rule) ในแต่ละ และทุก ๆ ค่าของข้อมูลในฐานข้อมูลเชิงสัมพันธ์จะต้องรับประกันได้ว่าผู้ใช้จะสามารถเข้าถึงข้อมูลทุกตัวในทุก ๆ ตารางได้โดยการรวมกันของชื่อตาราง ค่าคีย์หลัก และชื่อคอลัมภ์

กฎข้อที่ 3 (Missing Information Rule) การแยกแยะข้อมูลที่ยังไม่พร้อมหรือยังไม่มีค่านั้น จะใช้ค่านัลเป็นตัวกำหนด เพื่อให้เกิดความแตกต่างระหว่างสตริง (String) ที่ว่าง หรือสตริงของค่าแบลงค์ (Blank) หรือค่า 0

กฎข้อที่ 4 (System Catalog Rule) โครงสร้างแคตตาลอก (Catalog Structure) ของระบบจะต้องมีสภาพของฐานข้อมูลเชิงสัมพันธ์ ดังนั้นผู้ใช้จะสามารถสืบค้น หรือแก้ไขได้ด้วยภาษาเดียวกันกับการเรียกดูข้อมูลของระบบ

กฎข้อที่ 5 (Comprehensive Language Rule) ไม่ว่าจะมีภาษาอยู่กี่ภาษาหรือโหมด (Mode) ในการติดต่ออยู่กี่โหมดที่ใช้ได้ แต่อย่างน้อยที่สุดจะมี 1 ภาษา ที่สามารถนิยามโครงสร้างของระบบข้อมูล นิยามวิว เปลี่ยนแปลงแก้ไขข้อมูล ทั้งแบบการเขียนโปรแกรมหรือใช้คำสั่งแบบมีการประมวลผลโดยทันที ควบคุมความถูกต้องของข้อมูล และให้สิทธิการใช้งาน

กฎข้อที่ 6 (View Updatability Rule) ระบบจัดการฐานข้อมูลจะต้องมีการแก้ไขข้อมูลผ่านทางวิวได้ โดยสามารถตัดสินใจได้ว่า จะให้ผู้ใช้เพิ่ม หรือลบแถวในตาราง หรือแก้ไขคอลัมภ์ ใด ๆ โดยผ่านทางวิวได้เลยหรือไม่

กฎข้อที่ 7 (Set Level Update Rule) ระบบจัดการฐานข้อมูลควรมีภาษาที่สามารถให้ผู้ใช้สามารถเพิ่ม ลบ และแก้ไขข้อมูลในหลาย ๆ แถว หรือหลาย ๆ คอลัมภ์ได้

กฎข้อที่ 8 (Physical Data Independence Rule) ระบบจัดการฐานข้อมูลจะต้องมีความเป็นอิสระของข้อมูลในเชิงกายภาพ คือไม่ขึ้นกับฮาร์ดแวร์ หรือวิธีจัดการกับข้อมูล (Access Method)

กฎข้อที่ 9 (Logical Data Independence Rule) ระบบจัดการฐานข้อมูลจะต้องมีความเป็นอิสระของข้อมูลในระดับตรรก ไม่มีผลต่อการเปลี่ยนแปลงตารางหลัก

กฎข้อที่ 10 (Integrity Independence Rule) ระบบจัดการฐานข้อมูลจะต้องมีความเป็นอิสระในการจัดเก็บฐานข้อมูลเพื่อให้คงความถูกต้อง โดยไม่ขึ้นกับโปรแกรม

กฎข้อที่ 11 (Distribution Independence Rule) ระบบจัดการฐานข้อมูลจะต้องไม่ขึ้นต่อการกระจายของข้อมูลคือผู้ใช้ไม่ต้องสนใจว่าข้อมูลจะอยู่ในส่วนใด หรือคอมพิว-เตอร์เครื่องใดที่เชื่อมโยงกันเป็นระบบ

กฎข้อที่ 12 (Nonsubversion Rule) ระบบจัดการฐานข้อมูลจะต้องไม่ยอมให้ภาษาอื่นใดสามารถทำลายกฎต่าง ๆ โดยเฉพาะอย่างยิ่งในกฎที่เกี่ยวกับความถูกต้อง ของระบบหรือไม่ว่าจะเป็นเครื่องมือใด ๆ ก็ตาม

ในปัจจุบัน ยังไม่มีระบบจัดการฐานข้อมูลที่มีคุณสมบัติครบถ้วนตามกฎดังกล่าวข้างต้น ดังนั้นการตัดสินใจเลือกใช้ระบบจัดการฐานข้อมูลตัวใด จึงขึ้นกับลักษณะงานว่ามีความจำเป็นต้องปฏิบัติให้เป็นไปตามกฎใด และกฎข้อใดสามารถละเลยได้