บทที่ 3

แนวคิดและทฤษฎี

3.1 ระบบสารสนเทศ (Information System)

สารสนเทศ (Information) หมายถึง ข้อมูลที่ได้รับการประมวลผลให้อยู่ในรูปแบบที่มีความหมายต่อผู้รับและมีทั้งคุณค่าอันแท้จริงหรือที่คาดการณ์ว่าจะมีค่าสำหรับการดำเนินงานหรือการตัดสินใจในปัจจุบันหรืออนาคต

ระบบสารสนเทศ (Information System) เป็นระบบที่ประกอบด้วยคน เครื่องคอมพิวเตอร์ และอุปกรณ์ต่าง ๆ ที่ทำงานประสานกันเพื่อจัดทำสารสนเทศสนับสนุนการปฏิบัติงาน การจัดการ และการตัดสินใจในหน่วยงาน โดยทั่วไปอาจกล่าวได้ว่าหน่วยงานต่าง ๆ ล้วนแต่มีระบบสารสนเทศอยู่แล้ว ถ้าผู้บริหารไม่มีข้อมูลและข่าวสารหน่วยงานก็จะไม่สามารถเจริญก้าวหน้าได้ ระบบที่มีอยู่ทั่วไปเป็นระบบเอกสาร ทำงานล่าช้า และมักจะไม่สามารถให้บริการสารสนเทศได้ทันกับความต้องการของผู้ใช้และผู้บริหาร ดังนั้นจึงมีการจำกัดความหมายให้แคบลงไปว่าระบบสารสนเทศเป็นระบบที่ใช้คอมพิวเตอร์เก็บบันทึกและประมวลผลข้อมูลเป็นสารสนเทศ

ลักษณะสารสนเทศที่ดี อาจจำแนกได้เป็น 5 ลักษณะ ดังนี้

3.1.1 **เป็นปัจจุบัน** (Current) ข้อมูลอาจมีการปรับเปลี่ยนไปได้เรื่อย ๆ ตามกาลเวลา ข้อมูลที่ตรงตามความเป็นจริงในปัจจุบันจะมีค่ามากกว่าข้อมูลที่เป็นอดีตไปแล้ว ระบบสารสนเทศที่ดีต้องสามารถยืดหยุ่นให้มีการปรับเปลี่ยนค่าให้เป็นปัจจุบัน และหรือคงค่าเก็บไว้เพื่อประโยชน์การใช้งานต่าง ๆ กัน

3.1.2 **ทันเวลา** (Timely) สารสนเทศมีคุณค่าทางเวลาเข้ามาเกี่ยวข้อง ถ้าไม่ได้สารสนเทศในเวลาที่ต้องการอาจเกิดการสูญเสียโอกาสของการดำเนินงาน ระบบสารสนเทศที่มีประสิทธิภาพ คือ ระบบที่จะต้องจัดสรรให้ได้สารสนเทศเมื่อผู้ใช้ต้องการ

3.1.3 **มีค่าเที่ยงตรง** (Relevant) ผู้ใช้ต้องการสารสนเทศที่ตรงกับงานที่รับผิดชอบ ถ้าผู้ใช้ได้ข้อมูลที่ไม่สมบูรณ์หรือมีรายละเอียดปลีกย่อยเกินไปผู้ใช้ก็จะทำงานในส่วนของตนได้ไม่เต็ม

ที่ การที่ผู้ใช้แต่ละคนได้รับสารสนเทศที่ตรงกับความต้องการของตนเองมากเท่าใด ระบบสารสนเทศนั้นก็จะถูกจัดว่าเป็นระบบที่มีประสิทธิภาพยิ่งขึ้น

3.1.4 **มีความคงที่** (Consistent) ในหลาย ๆ กรณีสารสนเทศเองก่อให้เกิดความขัดแย้ง ข้อมูลจัดเก็บในหลาย ๆ ที่อาจไม่ตรงกัน วิธีการประมวลผลที่ต่างกันอาจทำให้เกิดความคลาดเคลื่อนขึ้นในผลลัพธ์ที่ได้ จุดมุ่งหมายหลักของระบบสารสนเทศข้อหนึ่ง คือ พยายามทำให้เกิดข้อขัดแย้งน้อยที่สุด ข้อมูลมีความคงที่มากที่สุดเท่าที่จะทำได้

3.1.5 **เสนอรูปแบบที่มีประโยชน์** (Presented in usable form) ระบบสารสนเทศที่มีประสิทธิภาพ คือ ระบบที่มีความยืดหยุ่นในการนำเสนอสารสนเทศให้กับผู้ที่ต้องการใช้สารสนเทศนั้น ๆ

3.2 วงจรการพัฒนาระบบ (System Development Life Cycle : SDLC)

วงจรการพัฒนาระบบ (System Development Life Cycle : SDLC) คือ กระบวนการในการพัฒนาระบบซึ่งประกอบด้วยขั้นตอนต่าง ๆ ในการดำเนินการวิเคราะห์และออกแบบระบบ ขั้นตอนต่าง ๆ ของวงจรการพัฒนาระบบแบ่งออกเป็น 7 ขั้นตอน คือ

3.2.1 **กำหนดปัญหา โอกาส และวัตถุประสงค์ของการพัฒนาระบบ** (Identifying Problems, Opportunities, and Objectives)

เป็นขั้นตอนแรกของการพัฒนาระบบ โดยนักวิเคราะห์ระบบจะต้องทำการกำหนดปัญหาที่เกิดขึ้น โอกาสของการพัฒนาระบบ และวัตถุประสงค์ในการพัฒนาระบบขั้นตอนนี้เป็นจุดสำคัญต่อความสำเร็จของส่วนที่เหลือของโครงการ การกำหนดปัญหาที่ผิดพลาดจะทำให้เสียเวลาในภายหลัง

โอกาสของการพัฒนาระบบ (Opportunities) เป็นสถานการณ์ที่นักวิเคราะห์ระบบเชื่อว่าจะสามารถปรับปรุงให้ดีขึ้นได้ด้วยการใช้ระบบสารสนเทศที่มีคอมพิวเตอร์ การใช้โอกาสเพื่อพัฒนาระบบนี้อาจจะทำให้ธุรกิจได้เปรียบในการแข่งขันหรือเป็นการกำหนดมาตรฐานอุตสาหกรรม

การกำหนดวัตถุประสงค์ (Objectives) ก็เป็นส่วนสำคัญของขั้นตอนนี้ นักวิเคราะห์ระบบจะต้องมองออกว่าองค์กรนั้นกำลังพยายามทำอะไร ดังนั้นนักวิเคราะห์ระบบควร

จะสามารถมองเห็นทิศทางบางอย่างของระบบสารสนเทศประยุกต์ที่สามารถช่วยให้องค์กรบรรลุวัตถุประสงค์จากการกำหนดปัญหาหรือโอกาส

3.2.2 **การกำหนดความต้องการสารสนเทศ** (Detemining Information Requirements) ในขั้นตอนต่อมานักวิเคราะห์ระบบจะต้องทำการวิเคราะห์ความต้องการสารสนเทศของผู้ใช้เฉพาะที่เกี่ยวข้อง มีเครื่องมือหลาย ๆ อย่างที่ถูกใช้เพื่อกำหนดความต้องการสารสนเทศขององค์กร รวมถึงการสุ่มตัวอย่างและการสำรวจข้อมูล การสัมภาษณ์ การออกแบบสอบถาม การสังเกตลักษณะการตัดสินใจและสภาพแวดล้อมของที่ทำงานและการทำต้นแบบ

3.2.3 **การวิเคราะห์ความต้องการของระบบ** (Analyzing System Needs) ในขั้นตอนนี้นักวิเคราะห์ระบบจะต้องทำการวิเคราะห์ความต้องการของระบบ มีเครื่องมือและเทคนิคพิเศษในการช่วยนักวิเคราะห์ระบบเพื่อกำหนดความต้องการหลายอย่าง ซึ่งสิ่งเหล่านี้รวมถึงการใช้แผนผังการไหลของข้อมูล (Data Flow Diagram) ในระหว่างขั้นตอนนี้นักวิเคราะห์ระบบจะต้องทำการวิเคราะห์โครงสร้างการตัดสินใจ ซึ่งในการตัดสินใจนั้นประกอบด้วยเงื่อนไขทางเลือกในเงื่อนไข การปฏิบัติและกฎในการปฏิบัติที่ถูกกำหนดขึ้น มี 3 วิธี สำหรับการวิเคราะห์โครงสร้างการตัดสินใจ คือ Structured English, Decision Tables and Decision Trees

3.2.4 **การออกแบบระบบ** (Designing The Recommended System) ในขั้นตอนนี้นักวิเคราะห์ระบบจะใช้ข้อมูลที่ได้เก็บรวบรวมมาเพื่อใช้ในการออกแบบระบบ นักวิเคราะห์ระบบจะต้องออกแบบการปฏิบัติการนำข้อมูลเข้าซึ่งทำให้ข้อมูลเข้าสู่ระบบได้อย่างถูกต้อง การออกแบบข้อมูลนำเข้าของระบบด้วยเทคนิคของการออกแบบรูปแบบและออกแบบหน้าจอรับข้อมูลที่ดี

ส่วนหนึ่งของขั้นตอนนี้คือ การออกแบบส่วนที่ติดต่อกับผู้ใช้ (User Interface) ส่วนที่ติดต่อกับผู้ใช้ หมายถึง ผู้ใช้จะติดต่อกับระบบด้วยอะไร เช่น แป้นพิมพ์ (key board) การใช้รายการเลือก (on-screen menus) การใช้เมาส์ (use of a mouse) และวิธีอื่น ๆ การออกแบบขั้นตอนนี้รวมถึงการออกแบบแฟ้มข้อมูลหรือฐานข้อมูลที่เป็นข้อมูลที่จำเป็นต้องใช้ในการตัดสินใจขององค์กร

3.2.5 **การพัฒนาระบบและการทำเอกสาร** (Developing And Documenting Documenting Software) ขั้นตอนที่ 5 ของการพัฒนาระบบนั้นเกี่ยวกับการออกแบบและการเขียนโปรแกรม เทคนิคในการออกแบบโครงสร้างการเขียนโปรแกรมและการจัดทำเอกสาร ได้แก่ วิธี HIPO, Flowchart, Nassi-Shneiderman charts เป็นต้น

3.2.6 **การทดสอบและการบำรุงรักษาระบบ** (Testing And Maintaining System) ก่อนที่ระบบสารสนเทศจะถูกนำไปใช้จริงจะต้องถูกทดสอบก่อน การทดสอบจะทำโดยนักเขียน

โปรแกรมโดยลำพังหรือทดสอบโดยนักวิเคราะห์ระบบร่วมกับนักเขียนโปรแกรมก็ได้ โดยการทดสอบนั้นจะใช้ข้อมูลจากระบบที่ทำอยู่ในปัจจุบัน

3.2.7 **การดำเนินการและการประเมินผลระบบ** (Implementing And Evaluating The System) ขั้นตอนสุดท้ายของการพัฒนาระบบ นักวิเคราะห์ระบบจะดำเนินการติดตั้งระบบ การฝึกอบรมการใช้งานในระบบให้กับผู้ใช้ระบบ การวางแผนในการเปลี่ยนจากระบบเก่าไปสู่การใช้งานของระบบใหม่ การประเมินผลระบบอาจทำในระหว่างขั้นตอนการพัฒนาระบบก็ได้ เพื่อให้ระบบสามารถดำเนินการได้อย่างมีประสิทธิภาพที่สุด

3.3 **แผนภาพการเคลื่อนไหวของข้อมูล** (Data Flow Diagram : DFD)

การวิจัยครั้งนี้ผู้วิจัยได้เลือกใช้แผนภาพการเคลื่อนไหวของข้อมูลเป็นเครื่องมือในการวิเคราะห์และออกแบบระบบงานแบบโครงสร้าง โดยใช้สัญลักษณ์ทั้ง 4 แบบ ดังแสดงในตารางที่ 3.1 มาประกอบเข้าด้วยกันอย่างมีกฎเกณฑ์ เพื่อแสดงการเคลื่อนไหวของข้อมูล แผนภาพการเคลื่อนไหวของข้อมูลมีข้อดีหลายประการดังนี้

3.3.1 ให้ความอิสระต่อนักวิเคราะห์ระบบที่จะทำการออกแบบระบบงาน โดยไม่ต้องผูกติดกับวิธีการติดตั้งระบบงานและข้อจำกัดต่าง ๆ ของระบบมากเกินไป

3.3.2 ช่วยให้ผู้วิเคราะห์เข้าใจในระบบงานและความสัมพันธ์ระหว่างระบบงานย่อย ๆ ทั้งยังสามารถแบ่งแยกขอบเขตของระบบงานออกจากสภาวะแวดล้อมได้อย่างชัดเจน

3.3.3 ใช้เป็นเครื่องมือในการสื่อสารระหว่างผู้วิเคราะห์กับผู้ใช้ระบบงาน ทำให้ผู้ใช้มองเห็นภาพการทำงานของระบบได้อย่างชัดเจน จึงสามารถพบข้อผิดพลาดที่อาจเกิดขึ้นหรือให้ข้อคิดแนะนำต่าง ๆ เพื่อให้ผู้วิเคราะห์ระบบสามารถปรับปรุงแก้ไขได้ดียิ่งขึ้น

3.3.4 ใช้เป็นเอกสารประกอบระบบงาน

ในการจัดทำแผนภาพการเคลื่อนไหวของข้อมูลมีขั้นตอนและวิธีการดังนี้

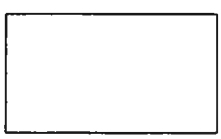
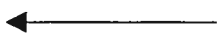
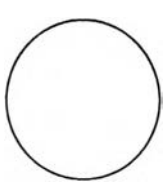
- เขียนแผนภาพโดยใช้หลักการเขียนจากบนลงล่าง (Top-Down Approach) หรือจากใหญ่ไปเล็ก โดยผู้วิเคราะห์จะต้องรู้ว่าหน่วยงานภายนอกที่เกี่ยวข้องกับระบบทั้งหมดมีอะไรบ้าง ข้อมูลจะเคลื่อนไปในระบบและเคลื่อนออกจากระบบอย่างไร จะต้องมีแฟ้มข้อมูลอะไรบ้าง ซึ่งสิ่งเหล่านี้จะทำให้ผู้วิเคราะห์ระบบสามารถกำหนดขอบเขตของระบบงานได้

- เขียนแผนภาพการเคลื่อนไหวของข้อมูลระดับ 0 (Context Diagram) และทบทวนว่าได้ครอบคลุมระบบงานทั้งหมดหรือไม่

- เพิ่มเติมรายละเอียดของแต่ละกระบวนการ โดยการแบ่งออกเป็นกระบวนการย่อย ๆ รวมทั้งเพิ่มเติมข้อยกเว้นต่าง ๆ

- ปรับปรุงแผนภาพ และกำหนดชื่อหรือคำอธิบายให้กับสัญลักษณ์ต่าง ๆ เพื่อให้ได้ความหมายที่ชัดเจนครบถ้วนและเข้าใจได้ง่าย

ตารางที่ 3.1 สัญลักษณ์ในแผนภาพการเคลื่อนไหวของข้อมูลและความหมาย

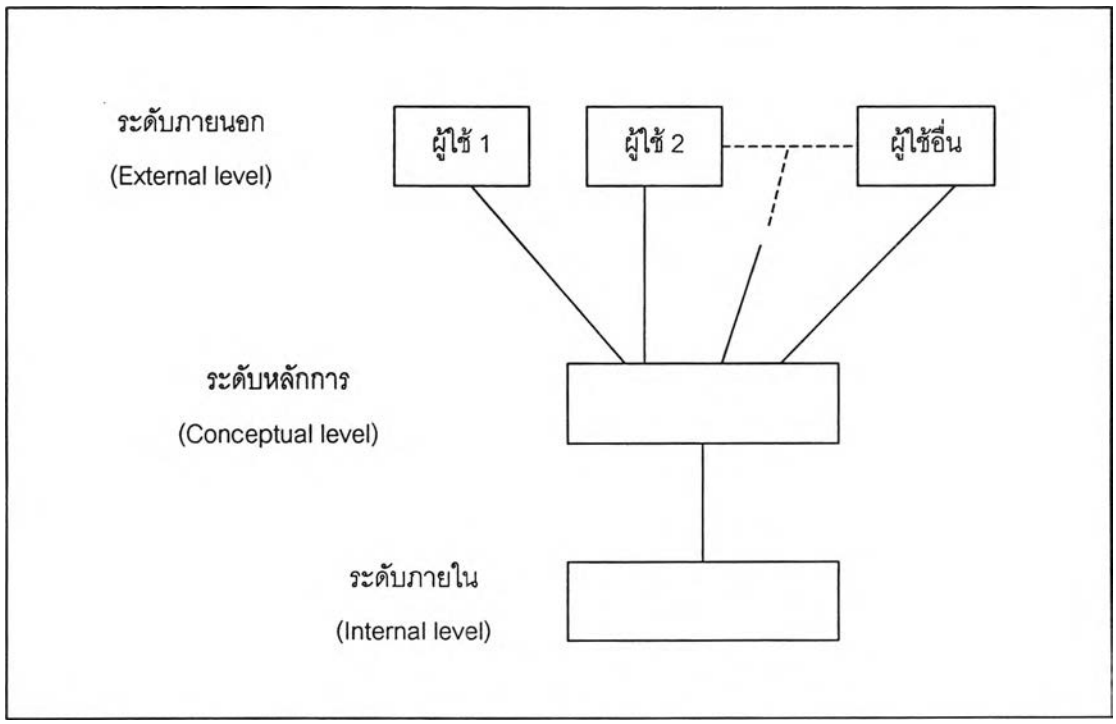
สัญลักษณ์	ความหมาย
	แหล่งรับ-ส่งข้อมูล (External Entity)
	ทิศทางการเคลื่อนไหวของข้อมูล (Flow of Data)
	กระบวนการ (Process)
	ที่เก็บข้อมูล (Data Store)

3.4 ระบบฐานข้อมูล (Database System)

ฐานข้อมูล (Database) คือ ศูนย์กลางของข้อมูลที่หมายถึง การปันส่วนข้อมูลกับผู้ใช้หลาย ๆ รายด้วยโปรแกรมประยุกต์ (Application program) ที่หลากหลาย รวมทั้งการสร้าง การปรับปรุงฐานข้อมูลและการสร้างรายงาน ฐานข้อมูลประกอบด้วยการรวบรวมข้อมูลที่คงอยู่ (Persistent data) ที่ใช้ด้วยระบบประยุกต์ที่กิจการกำหนดให้

ระบบฐานข้อมูล (Database system) คือ ระบบเก็บบันทึกด้วยคอมพิวเตอร์ เป็นระบบคอมพิวเตอร์ที่มีจุดประสงค์โดยรวมคือ การบำรุงรักษาสารสนเทศและทำให้สารสนเทศนั้นใช้ได้เมื่อต้องการ สารสนเทศที่เกี่ยวข้องสามารถเป็นอะไรก็ได้ที่เชื่อว่ามีความหมายต่อรายบุคคลหรือองค์กรซึ่งระบบจะทำหน้าที่ให้บริการ กล่าวได้อีกอย่างว่า ระบบฐานข้อมูล คือ สิ่งจำเป็นในการช่วยให้กระบวนการทั่วไปในการดำเนินธุรกิจของรายบุคคลนั้นหรือองค์นั้นได้สารสนเทศที่ต้องการ

3.4.1 สถาปัตยกรรมของระบบฐานข้อมูล (An Architecture for a Database System)



รูปที่ 3.1 ภาพสถาปัตยกรรมระบบฐานข้อมูล

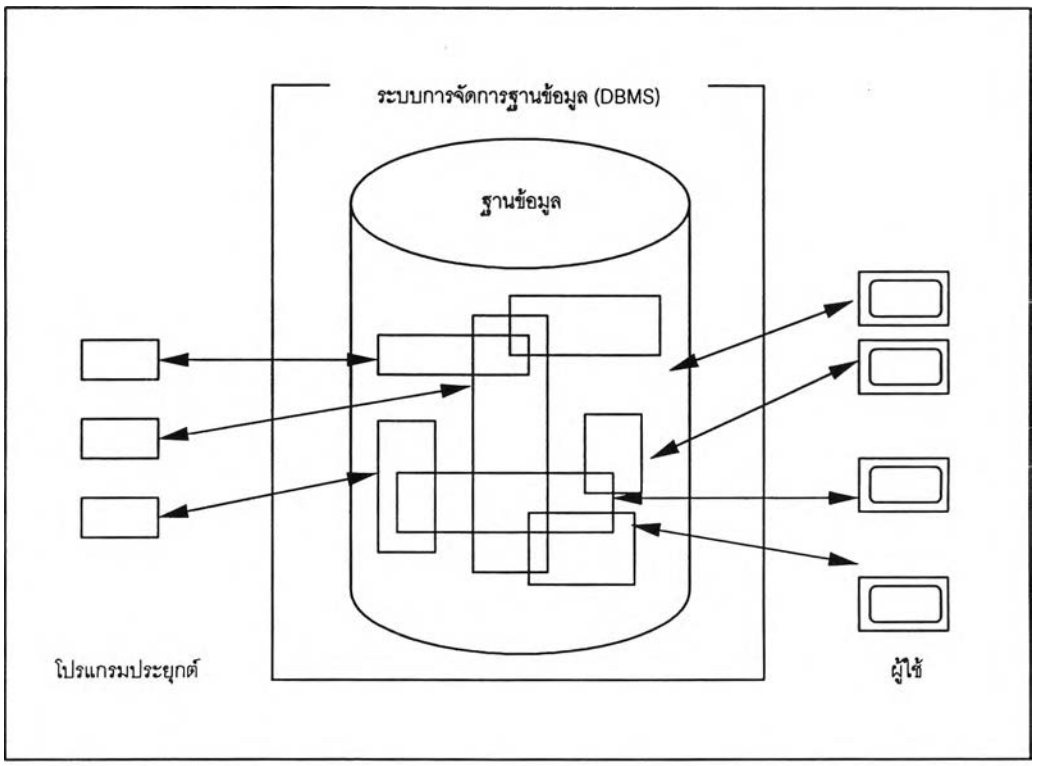
สถาปัตยกรรมระบบฐานข้อมูล แบ่งออกเป็น 3 ระดับ ดังรูปที่ 3.1 ซึ่งประกอบด้วย

3.4.1.1 ระดับภายนอก (External level) คือ ระดับของผู้ใช้รายบุคคล เป็นส่วนที่อยู่ใกล้ผู้ใช้มากที่สุด ระดับของข้อมูลจะมองเห็นจากการใช้งานของผู้ใช้แต่ละคน

3.4.1.2 ระดับหลักการ (Conceptual level) คือ ระดับที่อยู่ถัดขึ้นมา ได้แก่ระดับของการมองเอนทิตี โครงสร้างข้อมูล ความสัมพันธ์ระหว่างเอนทิตีทั้งหมด กฎเกณฑ์ต่าง ๆ เกี่ยวกับข้อมูลและผู้ที่มีสิทธิใช้ฐานข้อมูล เป็นต้น ข้อมูลในระดับนี้ผู้บริหารฐานข้อมูล (DBA) จะเป็นคนรับผิดชอบดูแลเกี่ยวกับการออกแบบและควบคุมการใช้ฐานข้อมูล

3.4.1.3 ระดับภายใน (Internal level) เป็นระดับล่างสุดของฐานข้อมูล ในระดับนี้เป็นระดับของการจัดเก็บข้อมูลจริง ๆ เกี่ยวข้องกับโครงสร้างของข้อมูล การเข้าถึงข้อมูล การกำหนดว่ามีข้อมูลดัชนีอะไรบ้าง เป็นต้น

3.4.2 **ระบบจัดการฐานข้อมูล** (Database Management System : DBMS)



รูปที่ 3.2 ภาพระบบการจัดการฐานข้อมูล (DBMS)

ระบบจัดการฐานข้อมูล คือ ซอฟต์แวร์ที่ออกแบบขึ้นมา เพื่ออำนวยความสะดวกให้ผู้ใช้สามารถเข้าถึงข้อมูลในฐานข้อมูล ระบบจัดการฐานข้อมูล ทำหน้าที่ควบคุมดูแลการสร้างและการเรียกใช้ฐานข้อมูล ซึ่งเป็นส่วนที่มีหน้าที่อำนวยความสะดวกในการทำงานต่าง ๆ เช่น การสร้างและการแก้ไขโครงสร้างของฐานข้อมูล การจัดเก็บดูแล และการเข้าถึงฐานข้อมูล การกำหนดคุณลักษณะ ตลอดทั้งการป้องกันความเสียหาย และการรักษาความปลอดภัยของข้อมูล เป็นต้น วัตถุประสงค์หลักของระบบจัดการฐานข้อมูล คือ การจัดหามุมมองของข้อมูลให้กับผู้ใช้ เพื่อให้สามารถใช้ข้อมูลได้อย่างมีประสิทธิภาพมากที่สุด

3.4.3 ประโยชน์การประมวลผลด้วยฐานข้อมูล

3.4.3.1 ลดความซ้ำซ้อนของข้อมูล (redundancy can be reduced) การประมวลผลโดยใช้ไฟล์ธรรมดานั้นจำเป็นที่ผู้ใช้แต่ละกลุ่มจะต้องมีไฟล์ส่วนตัว ดังนั้นจึงทำให้เกิดข้อมูลชนิดเดียวกันถูกเก็บไว้หลาย ๆ แห่ง หรือที่เรียกว่าความซ้ำซ้อน การนำข้อมูลทั้งหมดมาเก็บไว้ที่เดียวกันในฐานข้อมูลเป็นการลดความซ้ำซ้อนลงได้

3.4.3.2 สามารถหลีกเลี่ยงความขัดแย้งของข้อมูลได้ในระดับหนึ่ง (inconsistency can be avoided…to some extent) การจัดเก็บข้อมูลชุดเดียวกันไว้หลาย ๆ แห่งอาจจะก่อให้เกิดปัญหา กรณีการแก้ไขข้อมูลชุดเดียวกันทำไม่เหมือนกันในทุก ๆ แห่งทำให้เกิดปัญหาว่า ข้อมูลชุดเดียวกันอาจมีค่าในแต่ละแห่งไม่ตรงกัน ในระบบฐานข้อมูลจะสามารถจัดการควมคุมดูแลได้

3.4.3.3 สามารถใช้ข้อมูลร่วมกันได้ (the data can be shared) การใช้ข้อมูลร่วมกันได้นี้ไม่ได้จำกัดอยู่เฉพาะโปรแกรมที่ใช้ข้อมูลอยู่ในปัจจุบันเท่านั้น แต่รวมความถึงโปรแกรมประยุกต์ที่จะพัฒนาขึ้นมาใหม่ด้วย ที่สามารถจะใช้ข้อมูลที่มีอยู่ได้เลยโดยไม่จำเป็นต้องเพิ่มข้อมูลเข้าไปในระบบอีกกรณีที่ฐานข้อมูลมีข้อมูลที่ต้องการครบถ้วน

3.4.3.4 สามารถควบคุมความมาตรฐานได้ (standards can be enforced) จากการที่นำข้อมูลมารวมกันไว้ในฐานข้อมูล ทำให้ผู้ที่มีหน้าที่ควบคุมดูแลการใช้ระบบฐานข้อมูลสามารถกำหนดมาตรฐานของข้อมูลได้ เช่น ให้ใช้หน่วยมาตรการวัดที่เหมือนกัน รูปแบบการเขียนวันที่ให้เหมือนกัน เป็นต้น ซึ่งการที่ข้อมูลใช้มาตรฐานเดียวกันทำให้การแลกเปลี่ยนข้อมูลระหว่างระบบเป็นไปอย่างสะดวกและถูกต้อง

3.4.3.5 สามารถจัดหาระบบความปลอดภัยที่รัดกุมได้ (security restrictions can be applied) ระบบความปลอดภัยในที่นี้ หมายถึง การป้องกันไม่ให้ผู้ใช้ที่ไม่มีสิทธิมาใช้ข้อมูลในระบบได้ โดยผู้บริหารฐานข้อมูล (DBA) สามารถกำหนดสิทธิการใช้ให้แก่ผู้ใช้คนใดก็ได้

ตามความเหมาะสม และผู้ใช้แต่ละคนอาจจะใช้ข้อมูลได้ในระดับที่ต่างกันหรือวิวที่ต่างกัน โดยถ้าผู้บริหารฐานข้อมูลไม่ได้รวมข้อมูลใดไว้ในวิวของผู้ใช้แล้ว ผู้ใช้คนนั้นก็จะไม่มีสิทธิเรียกใช้ข้อมูลส่วนนั้น

3.4.3.6 สามารถควบคุมความคงสภาพของข้อมูลได้ (integrity can be maintained) เช่น ข้อมูลอายุพนักงาน 30 ปี แต่พิมพ์เป็น 300 ปี ข้อมูลในลักษณะเช่นนี้ผู้ออกแบบระบบฐานข้อมูลสามารถใส่กฎเกณฑ์เพื่อควบคุมความคงสภาพไว้ เช่น อาจจะใส่กฎว่าค่าของอายุจะต้องเป็นตัวเลขระหว่าง 16 ถึง 60 ปี เป็นต้น ดังนั้นเมื่อมีการใส่ข้อมูลระบบจัดการฐานข้อมูลก็จะควบคุมดูแลให้การใส่ข้อมูลถูกต้องตามกฎเกณฑ์

3.4.3.7 สามารถสร้างสมดุลในความขัดแย้งของความต้องการได้ (conflicting requirements can be balanced) การที่ผู้ใช้ทั้งหมดขององค์กรใช้ข้อมูลจากฐานข้อมูลร่วมกัน ทำให้ผู้บริหารฐานข้อมูลทราบถึงความต้องการและความสำคัญของผู้ใช้งานทั้งหมด จึงสามารถกำหนดโครงสร้างของฐานข้อมูลเพื่อให้บริการที่ดีที่สุดได้ เช่น เลือกเก็บข้อมูลที่ต้องใช้บ่อยไว้ในสื่อข้อมูลที่มีความเร็วเป็นพิเศษ เป็นต้น เป็นการสร้างสมดุลของความต้องการไม่ให้เกิดความขัดแย้งในหมู่ผู้ใช้

3.4.3.8 เกิดความเป็นอิสระของข้อมูล (data independence) ลักษณะของข้อมูลที่ไม่เป็นอิสระคือ ข้อมูลที่ถูกนำมาประยุกต์ใช้ยังมีความผูกพันอยู่กับวิธีการจัดเก็บและเรียกใช้ข้อมูล ในลักษณะการเขียนโปรแกรมประยุกต์บางประเภทอาจจำเป็นต้องใส่เทคนิคการจัดเก็บและเรียกใช้ข้อมูลไว้ในตัวโปรแกรมด้วย ดังนั้นถ้าต้องมีการเปลี่ยนแปลงวิธีการจัดเก็บหรือการเรียกใช้ข้อมูลแล้ว ผู้ใช้ก็จำเป็นที่ต้องสร้างวิธีการประยุกต์ขึ้นมาใหม่ ซึ่งเป็นความไม่สะดวกอย่างยิ่ง ทำให้การปรับปรุงโครงสร้างของข้อมูลทำได้ลำบาก ดังนั้นการใช้ระบบฐานข้อมูลจะทำให้เกิดความเป็นอิสระระหว่างการจัดเก็บข้อมูลและการประยุกต์ใช้ ทำให้ใช้งานได้อย่างมีประสิทธิภาพมากยิ่งขึ้น

3.4.4 **การออกแบบฐานข้อมูลเชิงสัมพันธ์** (Relational Database Design)

เป็นขั้นตอนการแปลงจากแบบจำลองข้อมูลเชิงตรรกเข้าสู่ระบบฐานข้อมูลเชิงสัมพันธ์อย่างมีเสถียรภาพภายใต้ระบบจัดการฐานข้อมูลที่ติดตั้งบนเครื่องคอมพิวเตอร์นั้น ๆ การออกแบบฐานข้อมูลเชิงสัมพันธ์ ประกอบด้วย 2 ขั้นตอนดังนี้

3.4.4.1 การออกแบบแบบจำลองข้อมูลเชิงตรรก (Logical Data Model: LDM)

การออกแบบแบบจำลองข้อมูลเชิงตรรก คือ การนำเสนอโครงสร้างของข้อมูลและกฎระเบียบขององค์กร (Business rule) อย่างชัดเจน จากการรวบรวมความต้องการของผู้ใช้ โดยคำนึงถึงความสัมพันธ์ระหว่างกันของข้อมูลเป็นหลักโดยไม่คำนึงถึงอุปกรณ์ฮาร์ดแวร์ โปรแกรมประยุกต์ โปรแกรมระบบ หรือโปรแกรมจัดการฐานข้อมูล การออกแบบในขั้นตอนนี้จะต้องมีการกำหนดสิ่งที่เป็นพื้นฐานได้แก่

- เอนทิตี (Entity) คือสิ่งต่าง ๆ ในระบบ อาจเป็นสิ่งที่มีอยู่จริงในทางกายภาพหรือสิ่งที่มีอยู่จริงในทางความคิด

- แอตทริบิวต์ (Attribute) คือข้อมูลที่แสดงลักษณะและคุณสมบัติของเอนทิตี

- รีเลชันชิป (Relationship) คือความสัมพันธ์ระหว่างเอนทิตี

ขั้นตอนการออกแบบแบบจำลองข้อมูลเชิงตรรก มีขั้นตอนดังนี้

1) การสร้างโครงร่างมุมมองของผู้ใช้ (Build skeletal user views)

ขั้นตอน LDM 1 กำหนดเอนทิตีหลัก

ขั้นตอน LDM 2 กำหนดความสัมพันธ์ระหว่างเอนทิตี

2) กำหนดคีย์ให้กับมุมมองของผู้ใช้ (Add keys to users views)

ขั้นตอน LDM 3 กำหนดคีย์หลักและคีย์รอง

ขั้นตอน LDM 4 กำหนดคีย์นอก

ขั้นตอน LDM 5 กำหนดคีย์ของธุรกิจ

3) การเพิ่มรายละเอียดให้กับมุมมองของผู้ใช้ (Add detail to user views)

ขั้นตอน LDM 6 การเพิ่มแอตทริบิวต์ที่เหลือ

4) การตรวจสอบกฎนอร์มัลไลเซชัน (Validate user views through normalization)

ขั้นตอน LDM 7 การตรวจสอบกฎนอร์มัลไลเซชัน

5) กำหนดแอตทริบิวต์ที่เป็นกฎธุรกิจเพิ่ม (Detemine additional attribute business rules)

ขั้นตอน LDM 8 กำหนดโดเมน

ขั้นตอน LDM 9 กำหนดทริกเกอร์การดำเนินการ

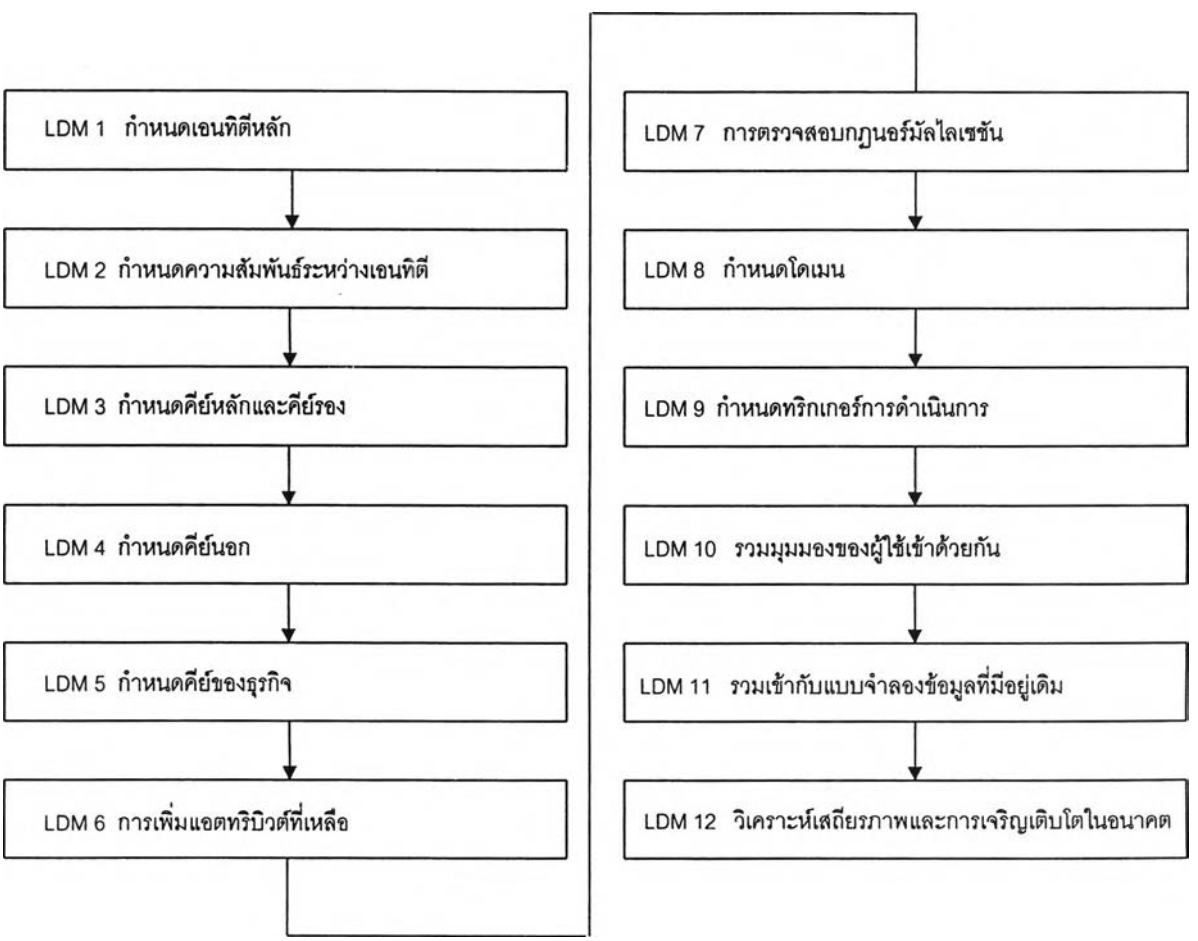
6) การรวบรวมมุมมองของผู้ใช้ (Integrate user views)

ขั้นตอน LDM 10 รวมมุมมองของผู้ใช้เข้าด้วยกัน

ขั้นตอน LDM 11 รวมเข้ากับแบบจำลองข้อมูลที่มีอยู่เดิม

ขั้นตอน LDM 12 วิเคราะห์เสถียรภาพและการเจริญเติบโตในอนาคต

ขั้นตอนการออกแบบแบบจำลองข้อมูลเชิงตรรกแสดงได้ดังรูปที่ 3.3



รูปที่ 3.3 ขั้นตอนการออกแบบแบบจำลองข้อมูลเชิงตรรก

3.4.4.2 การออกแบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database Design: RDD)

เป็นการนำเอาแบบจำลองข้อมูลเชิงตรรกที่ได้มาเปลี่ยนให้อยู่ในรูปแบบของฐานข้อมูลเชิงสัมพันธ์ โดยให้หลักการออกแบบฐานข้อมูลเชิงสัมพันธ์มี 2 ขั้นตอนดังนี้

1) การแปลโครงสร้างข้อมูลเชิงตรรก (Translate the logical data structure)

ขั้นตอน RDD 1 กำหนดตารางความสัมพันธ์

ขั้นตอน RDD 2 กำหนดสดมภ์

ขั้นตอน RDD 3 ปรับโครงสร้างของข้อมูลให้เข้ากับระบบจัดการฐานข้อมูล

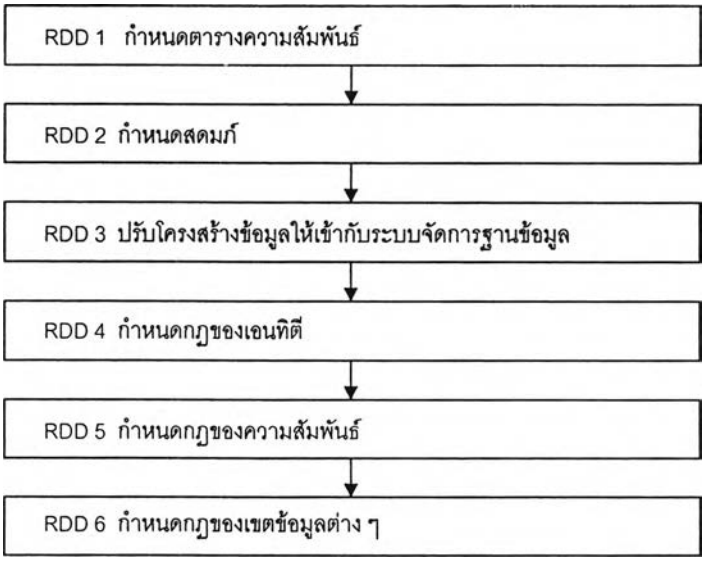
2) การแปลกฎบูรณภาพของข้อมูล (Translate the logical data integrity)

ขั้นตอน RDD 4 การออกแบบกฎธุรกิจที่เกี่ยวข้องกับเอนทิตี

ขั้นตอน RDD 5 การออกแบบกฎธุรกิจที่เกี่ยวข้องกับความสัมพันธ์

ขั้นตอน RDD 6 การออกแบบกฎธุรกิจเพิ่มเติมที่เกี่ยวกับแอตทริบิวต์

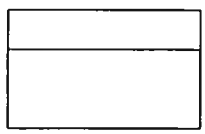
ขั้นตอนการออกแบบฐานข้อมูลเชิงสัมพันธ์แสดงได้ดังรูปที่ 3.4



รูปที่ 3.4 ขั้นตอนการออกแบบฐานข้อมูลเชิงสัมพันธ์

สัญลักษณ์ต่าง ๆ และความหมายที่ใช้ในแผนภาพแบบจำลองข้อมูลมีรายละเอียดดังแสดงในตารางที่ 3.2

ตารางที่ 3.2 สัญลักษณ์ต่าง ๆ ในแผนภาพแบบจำลองข้อมูล

สัญลักษณ์	ความหมาย
	เอนทิตีของระบบ
	เอนทิตีที่แสดงมาก่อนแล้วในแบบจำลองที่ระบุชื่อ
	ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (1:1)
	ความสัมพันธ์แบบหนึ่งต่อหลาย (1:M)
	ความสัมพันธ์แบบหลายต่อหลาย (M:M)
ชื่อแอตทริบิวต์ (PK)	แอตทริบิวต์ที่เป็นคีย์หลัก (Primary Key)
ชื่อแอตทริบิวต์ (AK)	แอตทริบิวต์ที่เป็นคีย์ที่เป็นคีย์รอง
ชื่อแอตทริบิวต์ (FK)	(Secondary Key)
ชื่อแอตทริบิวต์ (D)	แอตทริบิวต์ที่เป็นดิไรฟ์แอตทริบิวต์
	(Derived Attribute)