

บทที่ 2

หลักการและทฤษฎี

2.1 เครื่องมือช่วยในการแก้ปัญหา

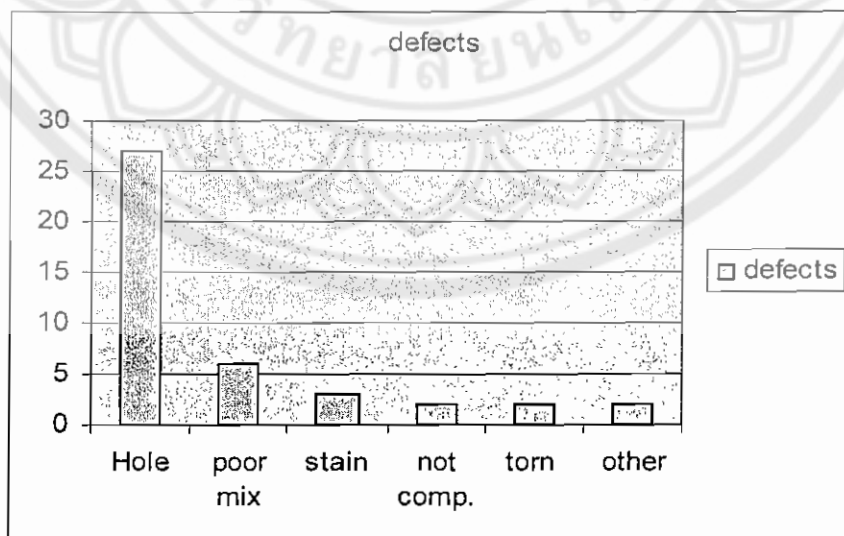
2.1.1 กราฟแท่ง

เป็นกราฟที่แสดงถึงความสูงแทนขนาดของข้อมูล มักใช้ข้อมูลที่แยกจากกัน ใช้เมื่อมีข้อมูลมากกว่าหรือเท่ากับ 2 ข้อมูล โดยใช้ในการเปรียบเทียบที่พื้นที่ของกราฟ หรือความยาวของเส้นกราฟไม่เหมาะสมที่จะใช้สำหรับการดูแนวโน้มในระยะยาวแต่เหมาะสำหรับใช้เปรียบเทียบข้อมูลในแต่ละช่วงเวลา

2.1.1.1 ลักษณะของกราฟแท่งนั้นก็คือการใช้เปรียบเทียบกับข้อมูลในช่วงระยะเวลาใดระยะเวลาหนึ่ง โดยการดูความยาวของแท่งกราฟ และที่สำคัญข้อมูลบนกราฟแต่ละแท่งจะต้องมีความอิสระต่อกัน ซึ่งจะขอสรุปการใช้งานกราฟแท่งไว้ดังต่อไปนี้

- 1) สามารถใช้กราฟแท่งได้ทั้งชนิดตั้งและชนิดนอนแต่นิยมใช้ในแนวตั้ง
- 2) การวาดกราฟนั้นจะต้องดูองค์ประกอบรวมของกราฟให้มีความสมดุลกันไม่มีรูปแบบตายตัว แต่สิ่งที่ต้องคำนึงถึงก็คือ สเกลของกราฟ ขนาดกราฟ และความกว้างของกราฟ

2.1.1.2 หน้าทีของกราฟแท่ง คือ ใช้ในการเปรียบเทียบข้อมูลที่เป็นอิสระต่อกัน ดังรูปที่ 2.1



รูปที่ 2.1 แสดงตัวอย่างกราฟแท่ง

(ที่มา : กราฟแท่งจากสื่อประกอบการสอนของพีณา จันทะแก้ว หน้าที 47)

2.1.2 ใบตรวจสอบ (Check Sheet)

คือใบรายการที่เราใช้อ้างอิงสำหรับการตรวจสอบเปรียบเทียบระบบงานที่เราได้กระทำจริงกับระบบงานที่กำหนดไว้ว่าเป็นอย่างไร ใบตรวจสอบนี้ สามารถใช้สำหรับการเก็บข้อมูล ซึ่งจะช่วยให้ได้ข้อมูลที่ถูกต้องมากยิ่งขึ้น ดังรูปที่ 2.2 และ รูปที่ 2.3

อาการเสีย	ความถี่
ปก PVC ไม่เรียบ	
เครื่องหยุด ค้าง ไม่ทำงาน	
ย้าทองแดงไม่เรียบ	

รูปที่ 2.2 แสดงตัวอย่างใบตรวจสอบ

(ที่มา : ใบตรวจสอบจากหนังสือการควบคุมคุณภาพของ เกษม พิพัฒน์ปัญญาคุณ)

Customers in Party	Count
1	
2	
3	
4	
5	
6	
>6	

รูปที่ 2.3 แสดงตัวอย่างใบตรวจสอบ

(ที่มา : ใบตรวจสอบจาก Mark, M. Davis, Nicholas, J. Aquilano, and Richard, B.

Chase, Fundamentals of Operations Management, 2003: 250.)






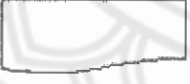


การออกแบบใบตรวจสอบ

เราสามารถแบ่งรายละเอียดขั้นตอนในการออกแบบและจัดทำใบตรวจสอบได้ดังนี้

- 1) กำหนดเป้าหมายในการตรวจสอบ เพื่อเราจะได้ออกแบบใบตรวจสอบให้สามารถใช้ประโยชน์ได้อย่างเต็มที่ เราต้องทราบถึงจุดประสงค์ที่แท้จริงว่าเราจะนำใบตรวจสอบมาใช้ตรวจสอบข้อมูลอะไรบ้าง และข้อมูลที่เราได้จากการตรวจสอบนั้นจะเอาไปใช้ทำอะไร เราอาจนำเทคนิคการตั้งคำถามมาช่วยในขั้นตอนนี้ก็ได้ เพื่อให้สามารถหาคำตอบได้ง่ายขึ้น เช่น "ปัญหาคืออะไร" "ข้อมูลอะไรบ้างที่เรา ต้องใช้ในการวิเคราะห์ปัญหา" "ใครจะเป็นคนใช้ข้อมูลนี้" "ใครมีหน้าที่ในการเก็บ ข้อมูล" ฯลฯ
- 2) กำหนดรูปแบบฟอร์มสำหรับการเก็บรวบรวมข้อมูล โดยเราจะกำหนดให้หัวข้อของข้อมูลที่เราต้องการทำการตรวจสอบอยู่ทางด้านซ้ายมือของกระดาษ ทางขวามือจะเว้นว่างไว้ให้ผู้ตรวจสอบกรอกรายละเอียดได้สำหรับเวลาในการตรวจสอบและสถานที่ในการตรวจสอบควรเอาไว้บนหัวกระดาษ
- 3) จัดเก็บข้อมูลที่ต้องการนำมาใช้ในการตรวจสอบ ผู้ทำการรวบรวมข้อมูลจะต้อง ทำการบันทึกข้อมูลที่เกิดขึ้นจริง
- 4) รวบรวมข้อมูลที่เก็บได้ทั้งหมดและนำไปวิเคราะห์

2.2 ผังงานโปรแกรม(Program Flowchart)

การเขียนผังโปรแกรมจะประกอบไปด้วยการใช้สัญลักษณ์มาตรฐานต่างๆที่เรียกว่า สัญลักษณ์ ANSI(American National Standards Institute) ในการสร้างผังงาน ดังตัวอย่างที่แสดงในรูปต่อไปนี้ ดังรูปที่ 2.4

	จุดเริ่มต้น / สิ้นสุดของโปรแกรม
	ลูกศรแสดงทิศทางการทำงานของโปรแกรมและการไหลของข้อมูล
	ใช้แสดงคำสั่งในการประมวลผลหรือการกำหนดค่าข้อมูลให้กับตัวแปร
	แสดงการอ่านข้อมูลจากหน่วยเก็บข้อมูลสำรองเข้าสู่หน่วยความจำหลักภายในเครื่องหรือการแสดงผลลัพธ์จากการประมวลผลออกมา
	การตรวจสอบเงื่อนไขเพื่อตัดสินใจโดยจะมีเส้นออกจากรูปเพื่อแสดงทิศทางการทำงานต่อไป เงื่อนไขเป็นจริงหรือเป็นเท็จ
	แสดงผลหรือรายงานที่ถูกสร้างออกมา
	แสดงจุดเชื่อมต่อของผังงานภายในหรือเป็นที่บรรจบของเส้นหลายเส้นที่มาจากหลายทิศทางเพื่อจะไปสู่การทำงานอย่างใดอย่างหนึ่งที่เหมือนกัน
	การขึ้นหน้าใหม่ ในกรณีที่ผังงานมีความยาวเกินกว่าที่จะแสดงพอในหนึ่งหน้า

รูปที่ 2.4 รูปแสดง สัญลักษณ์ในการเขียนผังงานโปรแกรม (Program Flowchart)
(ที่มา : กราฟแท่งจากสื่อประกอบการสอนของพินา จันทะแก้ว หน้าที่ 47)

2.3 แนวทางในการจัดทำคู่มือการใช้งาน(User Manual)

2.3.1 แนวทางที่ใช้จัดทำคู่มือการใช้งาน มีหลักการดังนี้

2.3.1.1 เนื้อหาส่วนใหญ่ของคู่มือการใช้งาน จะต้องเป็นการบรรยายขั้นตอนเพื่อให้ผู้ใช้ติดตามและดำเนินการตามขั้นตอนนั้นได้จนบรรลุวัตถุประสงค์ของการใช้งานในครั้งหนึ่งๆ ดังนั้นการบรรยายส่วนใหญ่จึงไม่ใช่การอธิบายทีละหัวข้อ แต่เป็นการแบ่งหัวเรื่องให้ตรงตามความต้องการทำงานของผู้ใช้

2.3.1.2 ควรมีการแบ่งส่วนของเนื้อหาในคู่มือเป็นส่วน ๆ ต่างกันโดยชัดเจนสามารถเปิดค้นหาเนื้อหาเฉพาะที่เกี่ยวข้องได้สะดวก

2.3.1.3 ในการแสดงผลการวิเคราะห์หรือการแสดงผลงานที่มีความซับซ้อน สามารถสร้างเป็นหัวข้อเพื่ออธิบายวิธีการทำความเข้าใจกับผลการวิเคราะห์หรือรายงานนั้นได้ในคู่มือการใช้งาน

2.3.2 ส่วนหลักของคู่มือการใช้งาน

ส่วนนี้จะมีการแบ่งออกเป็นบทๆ ได้หลายบท แต่ละบทอาจมีหลายหัวเรื่องย่อย ขึ้นกับการแบ่งหัวข้อของวิธีการใช้งานในแต่ละหัวเรื่องย่อย จะต้องครบถ้วนในตัวเอง คือ ผู้ใช้สามารถเปิดคู่มือมาถึงหัวเรื่องย่อยหนึ่งๆ และปฏิบัติตามขั้นตอนตั้งแต่ต้นจนจบ ที่อธิบายไว้ก็จะสามารถสำเร็จได้ตามวัตถุประสงค์ มีข้อควรปฏิบัติดังนี้

2.3.2.1 การกำหนดชื่อหัวเรื่องย่อยจะต้องใช้หลักการว่าจบในตัวเองตามความต้องการของผู้ใช้ในครั้งหนึ่งๆ

2.3.2.2 หัวเรื่องย่อยที่ไม่ควรขาดไป ควรจะต้องถูกอธิบายในลำดับต้นๆ ของคู่มือ

2.3.2.3 ในหัวเรื่องย่อยที่มีความคล้ายกันในเชิงปฏิบัติ สามารถรวมอธิบายในหัวเรื่องย่อยเดียวกันได้ โดยอธิบายวิธีการโดยละเอียดเพียง 1 ตัวอย่าง และแสดงเพียงตัวอย่าง (ของข้อมูล หรือ รายงาน) ที่ต่างกันไปประกอบ

2.3.2.4 ในกรณีที่มีการอ้างอิงถึงส่วนใดจากภาพหรือรายงานที่ใช้ประกอบการอธิบายขั้นตอนการใช้งาน ควรใช้สัญลักษณ์ หรือตัวอักษร (font) ที่แตกต่างจากคำบรรยายปกติ เพื่อให้ผู้ใช้สามารถมองหาส่วนที่อ้างถึงได้ง่ายขึ้นจากภาพประกอบนั้น

2.3.2.5 เมื่ออธิบายขั้นตอนเสร็จ อาจมี "ข้อควรระวัง" ปิดท้าย เพื่อเป็นการเตือนผู้ใช้ไม่ให้กระทำการผิดพลาดและอาจมี "แนวทางแก้ไขข้อผิดพลาด" ที่ช่วยนำทางผู้ใช้กลับมายังระบบปกติได้

2.3.2.6 ในการอ้างอิงระหว่างหัวข้อย่อยสามารถทำได้ การอ้างอิงนี้มีข้อควรระวังคืออย่าให้มีการอ้างอิงหลายชั้น

2.4 ฐานข้อมูลและระบบจัดการฐานข้อมูล

ฐานข้อมูล (database) หมายถึง กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ โดยมีความสัมพันธ์ซึ่งกันและกัน โดยไม่ได้บังคับว่าข้อมูลทั้งหมดนี้จะต้องเก็บไว้ในแฟ้มข้อมูลเดียวกันหรือแยกเก็บหลาย ๆ แฟ้มข้อมูล นั่นก็คือการเก็บข้อมูลในฐานข้อมูลนั้นเราอาจจะเก็บทั้งฐานข้อมูล โดยใช้แฟ้มข้อมูลเพียงแฟ้มข้อมูลเดียวกันได้หรือจะเก็บไว้ในหลายๆแฟ้มข้อมูลที่สำคัญคือจะต้องสร้างความสัมพันธ์ระหว่างระเบียบและเรียกใช้ความสัมพันธ์นั้นได้ มีการกำจัดความซ้ำซ้อนของข้อมูลออกและเก็บแฟ้มข้อมูลเหล่านั้นไว้ที่ศูนย์กลาง เพื่อที่จะนำข้อมูลเหล่านั้นมาใช้ร่วมกัน

ฐานข้อมูลที่รู้จักกันในปัจจุบันมีด้วยกัน 4 ประเภท คือ ฐานข้อมูลแบบเครือข่าย(hierarchical model) ฐานข้อมูลแบบลำดับชั้น (hierarchy model) ฐานข้อมูลแบบสัมพันธ์ (relational model) และฐานข้อมูลแบบจำลองเชิงวัตถุ (object oriented model) ฐานข้อมูลมีการนำเสนอต่อเนื่องมาเป็นระยะๆ ฐานข้อมูลที่ประสบความสำเร็จและมีผู้ใช้ในวงการธุรกิจเริ่มจากฐานข้อมูลแบบแรกคือ ฐานข้อมูลแบบลำดับชั้นหรือไฮราคี ถัดมาคือฐานข้อมูลแบบเครือข่าย และฐานข้อมูลที่มีการใช้กันมากที่สุดในปัจจุบัน คือ ฐานข้อมูลแบบสัมพันธ์ และปัจจุบันนี้มีฐานข้อมูลสมัยใหม่เกิดขึ้นคือ ฐานข้อมูลแบบออบเจกต์ นั่นคือฐานข้อมูลนั้นมีการนำเสนอข้อมูลและความสัมพันธ์ระหว่างข้อมูลที่แตกต่างกันไป ซึ่งการนำเสนอในฐานข้อมูลแบบสัมพันธ์หรือรูปแบบตารางเป็นที่นิยมใช้กันแพร่หลายมากที่สุด

2.4.1 ฐานข้อมูลแบบลำดับชั้น (hierarchical data model)

คิดขึ้นโดยบริษัทไอบีเอ็มเป็นฐานข้อมูลที่นำเสนอข้อมูลและความสัมพันธ์ระหว่างข้อมูลในรูปแบบของ โครงสร้างต้นไม้ (tree structure) เป็นโครงสร้างลักษณะคล้ายต้นไม้เป็นลำดับชั้น ซึ่งแตกออกเป็นกิ่งก้านสาขา ผู้ที่คิดค้นฐานข้อมูลแบบนี้คือ North American Rockwell เพื่อต้องการให้เป็นฐานข้อมูลที่สามารถกำจัดความซ้ำซ้อน (Data Redundancy) โดยใช้แนวความคิดของโปรแกรมที่ชื่อว่า Generalized Update Access Method(GUAM)

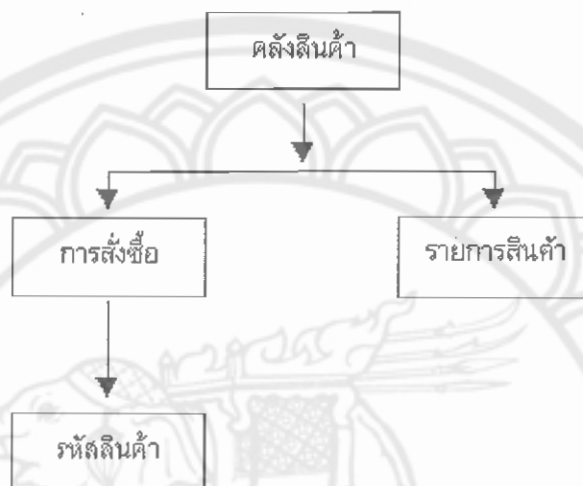
2.4.1.1 โครงสร้างของฐานข้อมูลแบบลำดับชั้น

1) ลักษณะโครงสร้างระบบฐานข้อมูลที่ให้รูปแบบนี้จะมีโครงสร้างของข้อมูลเป็นลักษณะ

ความสัมพันธ์แบบพ่อลูก คือ พ่อ (parent) 1 คนมีลูก (child) ได้หลายคน แต่ลูกมีพ่อได้คนเดียว (นั่นคือเป็นความสัมพันธ์แบบ 1 ต่อ n) หรือแบบพ่อคนเดียวมีลูก 1 คน (นั่นคือเป็นความสัมพันธ์แบบ 1 ต่อ 1) ซึ่งจัดแยกออกเป็นลำดับชั้น โดยในระดับชั้นที่ 1 จะมีเพียงแฟ้มข้อมูลเดียวนั่นคือมีพ่อคนเดียว ในระดับที่ 2 จะมีที่แฟ้มข้อมูลก็ได้ ในทำนองเดียวกันระดับ 2

ก็จะมีความสัมพันธ์กับระดับ 3 เหมือนกับ ระดับ 1 กับระดับ 2 โดยในโครงสร้างข้อมูลแบบลำดับชั้นแต่ละกรอบจะมีตัวชี้ (pointers) หรือ หัวลูกศรวิ่งเข้าหาได้ไม่เกิน 1 หัว จากตัวอย่างดังนี่

ดังรูปที่ 2.5



รูปที่ 2.5 แสดงโครงสร้างฐานข้อมูลแบบลำดับชั้น

(ที่มา: <http://www.elecnec.chandra.ac.th/learn/nh/qc/chspter/qc7tool.html>)

จากฐานข้อมูลแบบลำดับชั้นจะมีปัญหาถ้าความสัมพันธ์ของข้อมูลเป็นแบบลูกมีพ่อได้หลายคน จะใช้โครงสร้างฐานข้อมูลแบบลำดับชั้นไม่ได้ เช่น ความสัมพันธ์ระหว่างลูกจ้างกับงานที่ทำ งานชิ้นหนึ่งอาจทำโดยลูกจ้างหลายคนได้ โครงสร้างฐานข้อมูลแบบลำดับชั้น จะไม่สามารถออกแบบลักษณะข้อมูลแบบนี้ได้ปัญหาเช่นนี้ทำให้ไม่ค่อยมีผู้นิยมใช้ฐานข้อมูลแบบลำดับชั้น เนื่องจากความสัมพันธ์ของข้อมูลที่จะเก็บไว้ในฐานข้อมูลเป็นแบบพ่อ-ลูกเท่านั้น

2) กฎควบคุมความถูกต้อง คือ เรคอร์ดพ่อสามารถมีเรคอร์ดลูกได้หลายเรคอร์ดแต่เรคอร์ดลูกแต่ละเรคอร์ดจะมีเรคอร์ดพ่อได้เพียงเรคอร์ดเดียวเท่านั้นจากรูปแบบฐานข้อมูลแบบลำดับชั้นจะมีปัญหา ถ้าความสัมพันธ์ของข้อมูลเป็นแบบเรคอร์ดลูก 1 เรคอร์ดมีพ่อได้หลายเรคอร์ด จะใช้โครงสร้างฐานข้อมูลแบบลำดับชั้นไม่ได้ เช่นความสัมพันธ์ระหว่างลูกจ้างกับงานที่ทำงานชิ้นหนึ่งอาจทำโดยลูกจ้างหลายคนได้โครงสร้างฐานข้อมูลแบบลำดับชั้นจะไม่สามารถออกแบบลักษณะข้อมูลแบบนี้ได้ ปัญหาเช่นนี้ทำให้ไม่มีผู้นิยมใช้

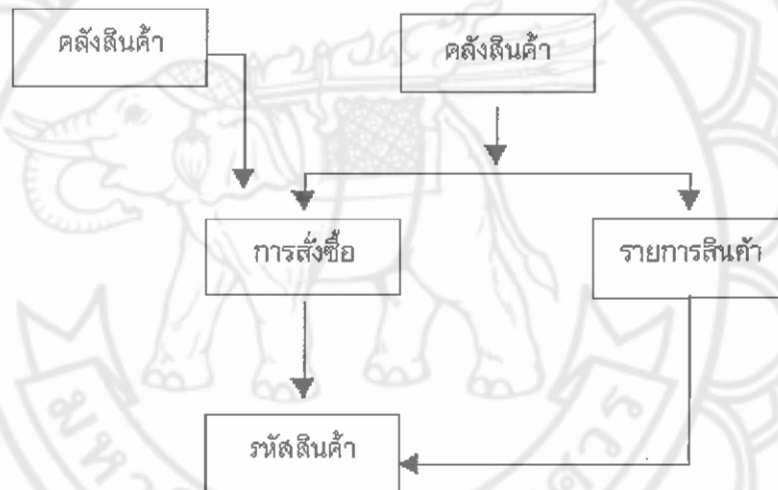
3) การจัดการ รูปของโครงสร้างต้นใช้ภาษาปฏิบัติการที่ละเรคอร์ด ตัวอย่างของภาษาปฏิบัติการของฐานข้อมูลนี้ได้แก่ IMS/VS

4) ลักษณะเด่นและข้อจำกัดของการจัดการฐานข้อมูลแบบลำดับชั้น

2.4.2 ฐานข้อมูลแบบ Network Data Model

2.4.2.1 โครงสร้างของฐานข้อมูลแบบเครือข่าย

ลักษณะโครงสร้างระบบฐานข้อมูลแบบเครือข่ายจะมีโครงสร้างของข้อมูล แต่ละเพิ่มข้อมูลมีความสัมพันธ์คล้ายร่างแหโดยมีลักษณะโครงสร้างคล้ายกับโครงสร้างแบบลำดับชั้น มีข้อแตกต่างที่ว่าโครงสร้างแบบเครือข่ายสามารถยินยอมให้ระดับชั้นที่อยู่เหนือกว่าจะมีได้หลายเพิ่มข้อมูลถึงแม้ว่าระดับชั้นถัดลงมาจะมีเพียงเพิ่มข้อมูลเดียวเปรียบเสมือนความสัมพันธ์แบบลูกจ้างกับงานที่ทำ โดยงานชิ้นหนึ่งอาจทำโดยลูกจ้างหลายคน (m ต่อ n) ดังรูปที่ 2.6



รูปที่ 2.6 แสดงโครงสร้างของฐานข้อมูลแบบเครือข่าย

(ที่มา: <http://www.elecnnet.chandra.ac.th/learn/nh/qc/chapter/qc7tool.html>)

เพิ่มข้อมูลการสั่งซื้อจะถูกเชื่อมโยงกับเพิ่มข้อมูลลูกค้าเพิ่มขึ้นอีก 1 เพิ่มข้อมูล ทำให้เพิ่มข้อมูลการสั่งซื้อเปรียบเสมือนงาน 1 ชิ้น จะถูกทำโดยลูกจ้าง มากกว่า 1 คนหรือในเพิ่มข้อมูลรหัสสินค้าจะมีเพิ่มข้อมูลการสั่งซื้อและเพิ่มข้อมูลรายการสินค้าเป็นพ้องซึ่งการออกแบบลักษณะของฐานข้อมูลแบบเครือข่ายจะทำให้สะดวกในการค้นหา มากกว่าลักษณะฐานข้อมูลแบบลำดับชั้น เพราะไม่ต้องไปเริ่มค้นหาตั้งแต่ข้อมูลต้นกำเนิดโดยทางเดียว ข้อมูลแต่ละกลุ่มจะเชื่อมโยงกันโดยตัวชี้ ลักษณะฐานข้อมูลนี้จะคล้ายกับลักษณะฐานข้อมูลแบบลำดับชั้น นั้นจะมีข้อแตกต่างกันตรงที่ในลักษณะฐานข้อมูลแบบเครือข่ายนี้สามารถมีต้นกำเนิดของข้อมูลได้มากกว่า 1

2.4.2.2 กฎการควบคุม

โครงสร้างแบบเครือข่ายสามารถยินยอมให้ระดับชั้นที่อยู่เหนือกว่าจะมีได้หลายเพิ่มข้อมูลถึงแม้ว่าระดับชั้นถัดลงมาจะมีเพียงเพิ่มข้อมูลเดียว โดยเรคอร์ดที่อยู่เหนือกว่ามีความสัมพันธ์กับเรคอร์ดที่อยู่ระดับล่างได้มากกว่า 1 เรคอร์ด โดยแต่ละเรคอร์ดสัมพันธ์กันด้วยลิงค์ (links) ฐานข้อมูลแบบเครือข่ายจะทำให้สะดวกในการค้นหามากกว่าฐานข้อมูลแบบลำดับชั้น เพราะไม่ต้องไปเริ่มค้นหาตั้งแต่ข้อมูลต้นกำเนิดโดยทางเดียว ข้อมูลแต่ละกลุ่มจะเชื่อมโยงกันโดยตัวชี้

2.4.2.3 ภาษาการจัดการ

ใช้ภาษาปฏิบัติการที่ละเรคอร์ดจัดการกับฐานข้อมูลตัวอย่างระบบปฏิบัติการรูปแบบภาษาของระบบจัดการฐานข้อมูลแบบเครือข่าย ได้แก่ IDMS

2.4.2.4 ลักษณะเด่นและข้อจำกัดระบบโครงสร้างแบบเครือข่าย

1) ลักษณะเด่น

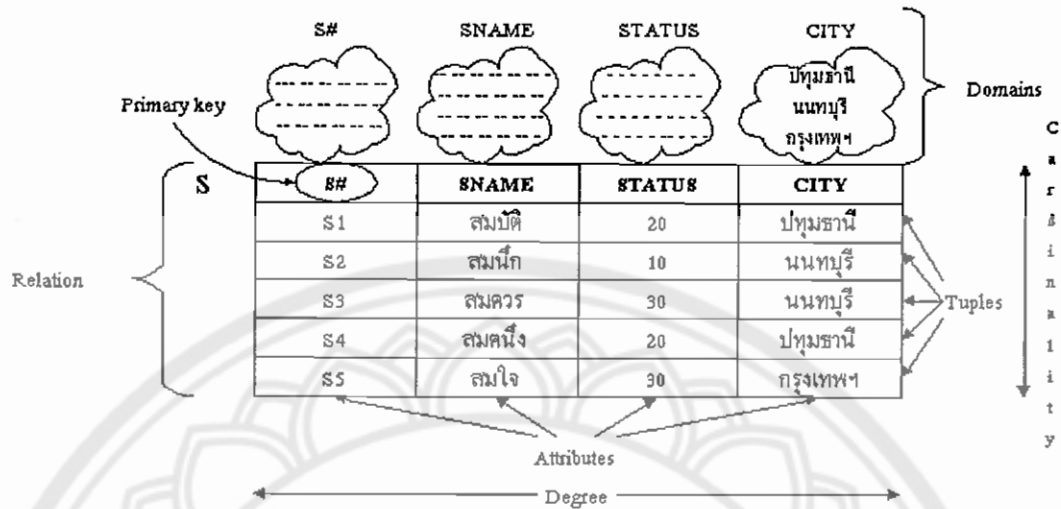
เหมาะสำหรับงานที่เพิ่มข้อมูลมีความสัมพันธ์แบบเครือข่าย มีโอกาสเกิดความซ้ำซ้อนของข้อมูลน้อยกว่าโครงสร้างแบบลำดับชั้น การค้นหาข้อมูลมีเงื่อนไขได้มากและกว้างกว่าโครงสร้างแบบลำดับชั้น

2) ข้อจำกัด

โครงสร้างแบบเครือข่ายเป็นโครงสร้างที่ง่ายไม่ซับซ้อนจึงทำให้ป้องกันความลับของข้อมูลได้ยากมีค่าใช้จ่ายและสิ้นเปลืองพื้นที่ในหน่วยความจำเพราะจะเสียพื้นที่ในอุปกรณ์เก็บข้อมูลสำหรับตัวบ่งชี้มาก ถ้าความสัมพันธ์ของระเบียบประเภทต่างๆ เกิน 3 ประเภท จะทำให้การออกแบบโครงสร้างแบบเครือข่ายยุ่งยากซับซ้อน นอกจากฐานข้อมูลที่กล่าวไปแล้วในปัจจุบันยังมี ฐานข้อมูลแบบออบเจกต์ (object-oriented data model) ที่ได้มีการคิดค้นและพัฒนาขึ้นในปัจจุบัน โดยฐานข้อมูลแบบนี้จะถูกนำเสนอข้อมูลและความสัมพันธ์ระหว่างข้อมูลในรูปออบเจกต์ (object) สำหรับฐานข้อมูลแบบออบเจกต์นั้นหน่วยงานธุรกิจในปัจจุบันนี้ยังไม่มีการใช้ ยังต้องมีการค้นคว้าและวิจัยต่อไปเพื่อที่จะสร้างฐานข้อมูลแบบออบเจกต์

2.4.3 ฐานข้อมูลแบบ Relational Data Model

ฐานข้อมูลแบบสัมพันธ์ (Relational Model) แบ่งออกเป็น 3 ส่วนคือ โครงสร้างข้อมูล (data structure) การควบคุมความถูกต้องให้กับข้อมูล (data integrity) และการจัดการกับข้อมูล (data manipulation) ดังรูปที่ 2.7



รูปที่ 2.7 แสดงโครงสร้างฐานข้อมูลแบบสัมพันธ์

(ที่มา: <http://www.elecnet.chandra.ac.th/learn/nh/qc/chspter/qc7tool.html>)

2.4.3.1 โครงสร้างข้อมูล (Data Structure)

โครงสร้างของฐานข้อมูลแบบสัมพันธ์เป็นการนำเสนอข้อมูลและความสัมพันธ์ระหว่างข้อมูลในรูป รีเลชัน (relation) Relation จะถูกเรียกอีกอย่างหนึ่งว่าตาราง (Table) โครงสร้างของ Relation ประกอบด้วย

1) แถว (Row) ของข้อมูล (body) แถวข้อมูล 1 แถว (body) หมายถึงข้อมูล 1 รายการ ซึ่งแต่ละแถวของ Relation เรียกว่า "Tuple" Tuple คือ แถวข้อมูลในตาราง โดยแต่ละแถวของข้อมูลจะประกอบไปด้วยหลาย Attribute หรือคอลัมน์ของข้อมูล จำนวนแถวข้อมูลในตารางเราเรียกว่า Cardinality และจำนวน Attributes ทั้งหมดในตารางเราเรียกว่า Degree

2) สดมภ์ (Column) แต่ละสดมภ์ของ Relation ได้แก่คุณลักษณะของข้อมูลในแต่ละแถวซึ่งเราเรียกว่า "Attribute" เช่น ตัวอย่าง Relation S สำหรับเก็บรายละเอียดของ suppliers ประกอบด้วย รหัส ชื่อ สถานะ และเมือง ซึ่ง Relation ดังกล่าวจะประกอบไปด้วย 5 tuples โดยแต่ละ tuples ประกอบไปด้วย 4 attribute โดยภายในคอลัมน์จะประกอบด้วย โดเมน (Domain) เป็นการกำหนดขอบเขตค่าข้อมูลและชนิดข้อมูลของแต่ละ Attribute ที่เป็นไปได้ทั้งหมด เช่นจากรูป domain ของ S# คือกลุ่มของรหัส suppliers ทั้งหมด ได้แก่ S1, S2, S3, S4 และ S5 และ กำหนดว่า STATUS ของตาราง S จะต้องีค่าเป็น 10, 20 และ 30 เท่านั้น หรือ S# จะต้องมีค่าเป็น S1, S2, S3, S4 และ S5 เท่านั้น และ CITY ของ suppliers ทั้งหมดจะต้องเป็น ปทุมธานี นนทบุรี และกรุงเทพฯเท่านั้น เป็นต้น

คีย์หลัก (Primary key) เป็น attribute หรือกลุ่มของ attribute ที่บ่งบอกว่า ข้อมูลจะต้องไม่ซ้ำกันในแต่ละแถวข้อมูลของตาราง

2.4.3.2 การควบคุมความถูกต้องให้กับข้อมูล(data integrity)

ในฐานข้อมูลแบบสัมพันธ์ มีกฎการควบคุมความถูกต้องของข้อมูลดังนี้

1) Tuple จะมีข้อมูลไม่ซ้ำกัน เนื่องจาก Relation ในโครงสร้างข้อมูลแบบสัมพันธ์อยู่ในรูปแบบของเซตทางคณิตศาสตร์ โดยภายในเซตจะต้องประกอบไปด้วยสมาชิกที่ไม่ซ้ำกัน ดังนั้น Relation R ใดๆ ต้องมี Attribute ใด attribute หนึ่งที่ทำให้แต่ละ tuple ใน relation มีข้อมูลไม่ซ้ำกัน เช่น Relation CUSTOMER เป็น relation ที่เก็บประวัติของลูกค้าบัญชีเงินกู้ของธนาคาร และประกอบไปด้วย เลขที่บัตรประชาชน ชื่อ นามสกุล และ ที่อยู่ของลูกค้า ซึ่งจะเห็นว่า ข้อมูลลูกค้ามีชื่อและนามสกุลซ้ำกัน คือปรากฏลูกค้าชื่อ สมบัติ นามสกุล มิมาพันธ์ จำนวน 2 tuple แต่มี attribute เลขที่บัตรประชาชน (CITIZEN_ID) เป็นสิ่งที่บ่งบอกให้รู้ว่าลูกค้าทั้งสองคนเป็นคนละคนกัน

2) Tuple ไม่มีลำดับจากบนลงล่าง เนื่องจาก body Relation ในโครงสร้างข้อมูลแบบสัมพันธ์อยู่ในรูปแบบของเซตทางคณิตศาสตร์โดยภายในเซตจะต้องประกอบไปด้วยสมาชิกที่ไม่มีลำดับ

3) Attribute ไม่มีลำดับจากซ้ายไปขวา เนื่องจาก heading ของ Relation ในโครงสร้างข้อมูลแบบสัมพันธ์อยู่ในรูปแบบของเซตทางคณิตศาสตร์ โดยภายในเซตจะต้องประกอบไปด้วยสมาชิกที่ไม่มีลำดับ

4) ในค่าของ Attribute ทุกค่าจะต้องเป็น atomicity เนื่องจาก Domain มีเฉพาะค่าที่เป็น atomic เท่านั้น ดังนั้นทุก Attribute ในแต่ละตำแหน่งของ tuple ใน relation จะมีค่าเพียงค่าเดียว จะไม่มีรายการของข้อมูล (repeating group หรือ list of value หรือ array) ใน relation นั่นก็หมายถึง Relation ต้องผ่านขั้นตอนการเพิ่มประสิทธิภาพของข้อมูล (Normalization) ตัวอย่างเช่น Relation SP ก่อนทำการเพิ่มประสิทธิภาพของข้อมูล ค่าของข้อมูลในแต่ละ Tuple จะมีลักษณะของรายการข้อมูล (repeating group) หลังจากทำการเพิ่มประสิทธิภาพของข้อมูล ค่าข้อมูลในแต่ละ Tuple ไม่มีรายการข้อมูลอีก

5) กำหนด Domain ให้กับข้อมูลจะมีข้อกำหนดว่าค่าที่กำหนดให้กับข้อมูล จะต้องเป็นค่า Scalar นั่นคือจะต้องเป็นค่าข้อมูลที่มีความหมายและเป็นหน่วยเล็กที่สุด ไม่สามารถแบ่งแยกออกไปได้อีก เช่น เลขที่เงินกู้ 014100001 สามารถแบ่งออกเป็น รหัสสาขา รหัสปีเงินกู้ และลำดับบัญชีเงินกู้ เป็นต้น หรือที่อยู่ สามารถแบ่งย่อยออกไปเป็น เลขที่บ้าน ตำบล อำเภอ

จังหวัดฯฯ ดังนั้น เลขที่เงินกู้ และที่อยู่ ไม่มีค่าเป็น Scalar เพราะยังสามารถแบ่งย่อยออกไปได้อีก แต่รหัส supplier (S#) มีค่าเป็น S1 ซึ่งแบ่งย่อยอีกไม่ได้เพราะฉะนั้น S1 มีค่าเป็น Scalar ค่าข้อมูลที่เกิดขึ้นในฐานข้อมูลที่มีค่าเป็น Scalar นี้เราจะเรียกว่ามีคุณลักษณะของ atomicity ข้อมูลที่สามารถนำมากำหนด Domain ได้ จะต้องเป็นข้อมูลที่เป็นอิสระจากข้อมูลอื่น เช่น relation S ,P และ SP ดังรูปข้างล่างนี้ รหัส ชื่อ สถานะ และเมืองของ supplier เป็นข้อมูลที่เป็นอิสระจากข้อมูลอื่น ในทำนองเดียวกับ รหัส ชื่อ สี น้ำหนัก และเมืองของ relation part เป็นข้อมูลที่เป็นอิสระจากข้อมูลอื่น แต่ relation SP รหัสตัวแทนจำหน่าย (S#) ค่าที่กำหนดขึ้นมาเป็นไปตามรหัสตัวแทนจำหน่าย(S#) ของ relation S และรหัสวัสดุ(P#) ค่าที่กำหนดขึ้นมาเป็นไปตามรหัสวัสดุ(P#) ของ relation P ดังนั้นทั้งสอง attribute จึงไม่สามารถนำมากำหนด domain ของ relation SP ได้

6) ข้อมูลจะต้องเป็นข้อมูลชนิดเดียวกัน เช่น ถ้าค่าข้อมูลของ COLOR ใน relation P มีค่าที่เป็นไปได้คือ "สีฟ้า น้ำทะเล" "สีฟ้าอ่อน" หรือ "น้ำตาล" แต่ถ้าเรากำหนดให้ค่าของ COLOR สามารถเป็น 1 สำหรับสีฟ้า น้ำทะเล หรือ 2 สำหรับสีฟ้าอ่อน และ 3 สำหรับสีน้ำตาล โดยค่าข้อมูลของ attribute COLOR สามารถบันทึกค่าข้อมูลได้ทั้งสองแบบแล้ว เราไม่สามารถกำหนด Domain ได้เพราะค่าข้อมูลมีชนิดเป็นได้ทั้ง string และ เลขจำนวนเต็ม ในเวลาเดียวกันไม่ได้

2.4.3.3 การจัดการข้อมูล

ภาษาฐานข้อมูล (structured query language, SQL) เป็นภาษาที่ใช้กันอย่างแพร่หลายมีลักษณะคล้ายกับภาษาอังกฤษ ใช้ในการปฏิบัติงานและควบคุมฐานข้อมูล ในภาษาฐานข้อมูลจะมีคำสั่งตั้งนี้การสร้างตาราง สามารถทำได้ด้วยคำสั่ง CREATE TABLE คำสั่งสอบถามข้อมูลพื้นฐานเป็นการสอบถามข้อมูลหรือ query คำสั่งการป้อนข้อมูลเพิ่มลงสู่ตารางเราสามารถปรับปรุงเปลี่ยนแปลงแก้ไขข้อมูลที่มีอยู่แล้วในตารางได้ด้วยคำสั่ง UPDATE นอกจากการแก้ไขข้อมูลด้วยคำสั่งของ DELETE เราสามารถสอบถามข้อมูลครั้งหนึ่งจากตารางได้มากกว่าหนึ่งตารางโดยใช้โครงสร้างของ SQL ที่เรียกว่า joining คำสั่งใน SQL สามารถกำหนดความปลอดภัยในการควบคุมการเข้าถึงข้อมูลในฐานข้อมูลได้ โดยคำสั่ง GRANT เป็นการกำหนดสิทธิมอบอำนาจให้สามารถเข้าถึงข้อมูล REVOKE เป็นการเรียกสิทธิอำนาจคืนจากการกำหนดมอบสิทธิด้วยคำสั่ง GRANT

ฐานข้อมูลแบบความสัมพันธ์นี้เป็นรูปแบบที่ถูกพัฒนาขึ้นมาภายหลังและเป็นที่นิยมใช้กันสำหรับการออกแบบฐานข้อมูลในปัจจุบัน โปรแกรมสำเร็จทางด้านฐานข้อมูลก็ใช้รูปแบบนี้เช่นกัน

2.4.3.4 ลักษณะเด่นและข้อจำกัดของการจัดการฐานข้อมูลแบบสัมพันธ์

ลักษณะเด่น

- 1) เหมาะกับงานที่เลือกดูข้อมูลแบบมีเงื่อนไขหลายคีย์ฟิลด์ข้อมูล
- 2) ป้องกันข้อมูลถูกทำลายหรือแก้ไขได้ดี เนื่องจากโครงสร้างแบบสัมพันธ์นี้ ผู้ใช้จะไม่ทราบว่าการเก็บข้อมูลในฐานข้อมูลอย่างแท้จริงเป็นอย่างไร จึงสามารถป้องกันข้อมูลถูกทำลายหรือถูกแก้ไขได้ดี
- 3) การเลือกดูข้อมูลจะทำได้ง่ายขึ้น มีความซับซ้อนของข้อมูลระหว่างแฟ้มต่างๆ น้อยมาก อาจมีการฝึกฝนเพียงเล็กน้อยก็สามารถใช้ทำงานได้

ข้อจำกัด

- 1) มีการแก้ไขปรับปรุงแฟ้มข้อมูลได้ยากเพราะผู้ใช้จะไม่ทราบการเก็บข้อมูลในฐานข้อมูลอย่างแท้จริงเป็นอย่างไร
- 2) มีค่าใช้จ่ายของระบบสูงมากเพราะเมื่อมีการประมวลผลคือการอ่าน เพิ่มเติมปรับปรุงหรือยกเลิกระบบจะต้องทำการสร้างตารางขึ้นมาใหม่ทั้งที่ในแฟ้มข้อมูลที่แท้จริงอาจจะมีการเปลี่ยนแปลงเพียงเล็กน้อย แต่ต้องมาปรับแต่งตารางใหม่ให้ผู้ใช้แฟ้มข้อมูลนั้นถูกใช้ในรูปของตารางที่ดูง่ายสำหรับผู้ใช้

สรุปได้ว่าฐานข้อมูลอาจมีการใช้ฐานข้อมูลต่างกัน แต่ในองค์กรส่วนใหญ่ นิยมใช้ฐานข้อมูลแบบสัมพันธ์มากที่สุด ส่วนฐานข้อมูลแบบลำดับชั้นและฐานข้อมูลแบบเครือข่าย ปัจจุบันนี้ไม่ได้รับความนิยมแล้ว แต่ยังมีใช้อยู่ในองค์กรขนาดใหญ่ที่มีเครื่องคอมพิวเตอร์เมนเฟรม เทรนด์การยังมีการใช้แบบจำลองข้อมูลแบบลำดับชั้นอยู่ ส่วนฐานข้อมูลแบบอบเจกต์เป็นฐานข้อมูลสมัยใหม่ซึ่งปัจจุบันยังไม่มีผู้ใช้แพร่หลายนักเมื่อเทียบกับฐานข้อมูลแบบสัมพันธ์ ซึ่งยังต้องมีการค้นคว้าและวิจัยต่อไป

2.5 การจัดทำแบบสอบถาม

แบบสอบถามคือ เอกสารที่นักวิเคราะห์ระบบสร้างขึ้นโดยมีมีวัตถุประสงค์เพื่อเก็บรวบรวมข้อมูลต่างๆ ที่มีความจะเป็นต่อการวิเคราะห์ระบบ โดยส่งไปให้ผู้เกี่ยวข้องตอบกลับมา การใช้แบบสอบถามนักวิเคราะห์ระบบสามารถที่จะสร้างคำถามในแบบสอบถามเกี่ยวข้องกับเรื่องที่ศึกษาด้านต่างๆ เช่น ทักษะคน ทักษะการปฏิบัติงานและคุณสมบัติ ที่มีความสำคัญกับองค์กร ผู้ซึ่งอาจมีผลกระทบต่อระบบทั้งในปัจจุบันและอนาคต ผลที่ได้รับจากการใช้แบบสอบถามสามารถนำมาใช้เป็นข้อมูลเพื่อการวิเคราะห์ต่อไป

2.5.1 วางแผนการใช้แบบสอบถาม

2.5.1.1 ควรจะพิจารณาใช้แบบสอบถามถ้าคนที่คุณต้องการสอบถามนั้นมีกระจายอยู่ตามที่ต่างๆ เช่น ตามสาขาย่อยของบริษัท

2.5.1.2 ควรพิจารณาใช้แบบสอบถามถ้ามีคนที่เกี่ยวข้องกับโครงการระบบจำนวนมาก และต้องการรู้สัดส่วนของกลุ่มคน

2.5.1.3 ควรเริ่มพิจารณาใช้แบบสอบถามถ้าต้องการนำแบบสอบถามนั้นไปใช้เพื่อการศึกษาและใช้เป็นเกณฑ์มาตรฐานทางความคิดในการกำหนดทิศทางของโครงการระบบ

2.5.1.4 ควรเริ่มพิจารณาใช้แบบสอบถามถ้าต้องการสืบค้นปัญหาที่เกิดขึ้นกับระบบในปัจจุบัน และดูว่าต้องการข้อมูลเพิ่มเติมส่วนใดเพื่อใช้ในการสัมภาษณ์ต่อไป

2.5.2 การเขียนคำถาม (Writing Questions)

2.5.2.1 การใช้คำถามปลายเปิด

จะเป็นการตอบคำถามโดยให้ผู้ตอบมีอิสระในการตอบ ซึ่งถ้าไม่มีการกำหนดขอบเขตของ คำตอบแล้ว จะไม่สามารถเก็บข้อมูลได้ถูกต้องเป็นไปตามที่ต้องการ ในการตั้งคำถามแบบเปิดจะต้องตั้งคำถามให้แคบเพียงพอที่คำตอบที่มีทิศทางเฉพาะคำถามปลายเปิดนี้จะมีประโยชน์ในสถานการณ์ที่เป็นการสำรวจ วิจัย เช่น ต้องถามคำถามเกี่ยวกับทัศนคติ

2.5.2.2 การใช้คำถามแบบปิด

คำถามปิด จะเป็นคำถามที่กำหนดคำตอบให้ตอบ การใช้คำถามปิดควรจะใช้เมื่อนักวิเคราะห์สามารถที่จะกำหนดรายการคำตอบได้อย่างชัดเจน คำถามแบบปิดทำให้ง่ายต่อตอบ เพราะมีทางเลือกของคำตอบ และง่ายต่อการแปลความหมายของข้อมูล ที่ได้รับการตอบ

2.5.2.3 ภาษาที่ใช้ในแบบสอบถาม

ภาษาที่ใช้ในการตั้งคำถามในแบบสอบถาม เพื่อเป็นการตรวจสอบให้มั่นใจว่าภาษาที่ใช้ในแบบสอบถามเหมาะสมต่อผู้ตอบนักวิเคราะห์ควรจะลองทำคำถามจากกลุ่มตัวอย่างเพื่อทดสอบกลุ่มย่อยและขอคำแนะนำในการใช้ภาษาหรือคำศัพท์ที่ใช้จากผู้เชี่ยวชาญขอคำแนะนำการเลือกภาษาที่ใช้ในแบบสอบถามดังนี้

- 1) ใช้ภาษาที่ตอบสนองได้ดี ใช้คำที่เข้าใจง่าย
- 2) หลีกเลี่ยงการใช้คำถามที่จะเป็นคำเฉพาะให้มากที่สุด เนื่องจากอาจเป็นคำที่มีความหมายไม่ชัดเจน

- 3) ใช้คำถามที่สั้น กระชับ ได้ใจความ

- 4) ไม่ใช้คำหยาบคาย

5) หลีกเลียงคำที่มีความเอนเอียงต่างๆ ในคำถาม หรือคำถามที่มีเป้าหมายที่ต้องการคำตอบแบบใดแบบหนึ่ง

6) คำถามนั้นจะต้องมีเป้าหมายที่ตอบสนองได้ตรงกับความต้องการอยากรู้คำถามที่ตั้งขึ้นมานั้นต้องแน่ใจว่าเป็นเทคนิคที่ถูกต้องก่อนที่จะใช้

2.5.2.4 การใช้สเกลในแบบสอบถาม

สเกล หมายถึงการกำหนดตัวเลขหรือสัญลักษณ์อื่นที่ใช้แสดงคุณสมบัติ หรือลักษณะเพื่อวัตถุประสงค์ในการวัดลักษณะของข้อมูลสเกลจะถูกกำหนดเองตามเป้าหมายที่ต้องการ เช่น สเกลของปรอท ต้องการวัดอุณหภูมิ เข็มไมล์รถยนต์ วัดความเร็ว เข็มตาชั่งกิโลกรัม บอคน้ำหนัก ส่วนสเกลของแบบสอบถาม คือ รูปแบบทางเลือกคำถามในแบบสอบถาม

2.5.3 ความถูกต้องและความน่าเชื่อถือของคำตอบ

2.5.3.1 ความถูกต้องของคำตอบคือการใช้แบบสอบถามนั้นวัดว่าจะได้ตามเป้าหมายในการนำมาวิเคราะห์หรือไม่ ตัวอย่าง ถ้าแบบสอบถามถูกใช้เพื่อตัดสินใจว่าองค์กรนั้นพร้อมที่จะเปลี่ยนมาใช้คอมพิวเตอร์หรือไม่ก็ต้องตั้งคำถามให้ถูกต้องตามที่ต้องการเพื่อนำคำตอบนั้นมาตัดสินใจ

2.5.3.2 ความน่าเชื่อถือของคำตอบ คือการแสดงให้เห็นว่า ถ้าแบบสอบถามนั้นใช้มาครั้งหนึ่งแล้วนำมาใช้อีกภายใต้เงื่อนไขเดียวกันและเหตุผลเดียวกันแล้วยังสามารถใช้เป็นเครื่องมือวัดได้ผลเหมือนเดิม

2.5.4 การออกแบบและการจัดการแบบสอบถาม

2.5.4.1 รูปแบบของแบบสอบถาม (Questionnaire format)

- 1) ให้เว้นช่องกว้าง ให้แบบสอบถามดูสะอาด น่าตอบ
- 2) เว้นช่องว่างในการที่จะตอบคำถามให้พอเพียงกับการตอบในกรณีถ้าเป็นคำถามเปิด และต้องมีสัก 3-5 บรรทัดเพื่อให้ผู้ตอบแสดงความเห็นได้สะดวก
- 3) ถ้าเป็นคำถามที่ต้องวงคำตอบ ซึ่งค่อนข้างจะคาดคะเนยาก ซึ่งทำให้เป็นการยากในการรวบรวม รวมคำตอบด้วย หรือกรณีเป็นแบบเช็คบล็อกร ให้ใส่วงเล็บและหรือช่องในการตอบ

4) การจัดรูปแบบคำถามให้เป็นไปตามเป้าหมาย ถ้าใช้เครื่องอ่านฟอร์ม ก็ต้องออกแบบให้เป็นไปตามแบบของเครื่องอ่านนั้น

5) การออกแบบสอบถามให้มีรูปแบบสอดคล้องกัน เช่น ในแต่ละปัญหาจะมีกรอบสีเน้นที่คำตอบเหมือนกันหมด เนื่องจากจะทำให้ผู้ตอบตอบได้เร็ว และผิดพลาดน้อย

2.5.4.2 การเรียงคำถาม

- 1) คำถามที่สำคัญในการตอบสนองควรเป็นคำถามแรก
- 2) กลุ่มของหัวข้อคำถามต่าง ๆ ควรเหมือนกัน หรือสอดคล้องกัน
- 3) ให้มีคำถามที่มีข้อโต้แย้งนำไปสู่ปัญหาน้อยที่สุด

2.6 ความรู้เบื้องต้นเกี่ยวกับ Visual Basic (VB)

โปรแกรม Visual Basic (VB) เป็นโปรแกรมสำหรับพัฒนาโปรแกรมประยุกต์ที่กำลังเป็นที่นิยมใช้อยู่ในปัจจุบัน โปรแกรม Visual Basic เป็นโปรแกรมที่ได้เปลี่ยนรูปแบบการเขียนโปรแกรมใหม่ โดยมีชุดคำสั่งมาสนับสนุนการทำงาน มีเครื่องมือต่าง ๆ ที่เรียกกันว่า คอนโทรล (Controls) ไว้สำหรับช่วยในการออกแบบ (Toolbars) เป็นแถบสัญลักษณ์ที่ใช้สำหรับเข้าถึงชุดคำสั่งของ Visual Basic ได้ทันที โดยจะนำคำสั่งที่ถูกใช้งานบ่อย ๆ มาแสดง

2.6.1 Toolbars สามารถแบ่งออกได้เป็น 4 กลุ่มใหญ่ ๆ คือ

2.6.1.1 Standard Toolbars เป็น툴บาร์มาตรฐานประกอบด้วยคำสั่งที่เกี่ยวข้องกับการจัดการ Project

2.6.1.2 Edit Toolbars เป็น툴บาร์ที่ประกอบไปด้วยคำสั่งที่ใช้สำหรับช่วยในการเขียนโค้ดใน code editor

2.6.1.3 Debug Toolbars เป็น툴บาร์ที่ประกอบไปด้วยคำสั่งที่ใช้สำหรับตรวจสอบการทำงานการประมวลผลโปรแกรม

2.6.1.4 Form Editor Toolbars เป็น툴บาร์ที่ประกอบไปด้วยคำสั่งที่ใช้สำหรับช่วยในการปรับขนาด, ย้าย, เปลี่ยนตำแหน่งคอนโทรลต่างๆ ที่อยู่บนฟอร์ม

2.6.2 Form Designer

เป็นส่วนที่ใช้ออกแบบการแสดงผลส่วนที่ใช้ติดต่อกับผู้ใช้ ฟอร์มเป็นออบเจกต์แรกที่ถูกเตรียมไว้ให้ใช้งาน คอนโทรลทุกตัวที่ต้องการใช้งานจะต้องนำไปบรรจุไว้ในฟอร์ม นำคอนโทรลมาประกอบกันขึ้นเป็นโปรแกรมประยุกต์ ทุกครั้งที่เปิด Visual Basic ขึ้นมา หรือ สร้าง Project ใหม่ จะมีฟอร์มว่าง 1 ฟอร์มถูกสร้างเตรียมไว้เสมอ

2.6.3 Project Explorer

ใช้สำหรับบริหารและจัดการโปรเจกต์ โดยจะแสดงองค์ประกอบของแต่ละโปรเจกต์แบบโครงสร้างต้นไม้ (tree-view) ตัวโปรเจกต์จะหมายถึงโปรแกรมประยุกต์ซึ่งจะอยู่ส่วนบนสุด ถัดมาจะแสดงส่วนประกอบต่างๆ ของโปรเจกต์นั้นๆ ว่าประกอบด้วยอะไรบ้าง เช่น ฟอร์มโมดูล

รายงาน เป็นต้น ถ้ามี 2 โปรเจกต์ขึ้นไปก็จะแสดงแยกออกเป็นส่วนตัวหากอีกโปรเจกต์ ถ้าต้องการใช้งานส่วนใด ของโปรเจกต์ไหนก็สามารถคลิกเลือกได้ทันที

2.6.4 Properties Window

หน้าต่างคุณสมบัติเป็นส่วนที่ใช้กำหนดคุณสมบัติของออบเจกต์ที่ถูกเลือก(active) หรือได้รับความสนใจ (focus) อยู่ขณะนั้นซึ่งสามารถที่จะปรับเปลี่ยนค่าต่างๆ ของคอลโทรลเพื่อให้เกิดความเหมาะสมและตรงกับความต้องการใช้งานได้ทันที

ในหน้าต่างคุณสมบัติ จะประกอบไปด้วยแท็บ 2 แท็บ คือ

- 1) แท็บ Alphabetic เป็นแท็บที่แสดงรายการคุณสมบัติ เรียงตามตัวอักษรในภาษาอังกฤษ
- 2) แท็บ Categorized เป็นแท็บที่แสดงรายการคุณสมบัติ โดยการจัดกลุ่มของคุณสมบัติที่มีหน้าที่คล้ายกัน หรือมีความสัมพันธ์กัน

หน้าต่าง Form Layout เป็นส่วนที่แสดงให้เห็นตำแหน่งของฟอร์ม และสามารถกำหนดตำแหน่งของฟอร์ม ที่ปรากฏบนจอภาพในขณะประมวลผลได้ โดยการเคลื่อนย้ายฟอร์มจำลอง ที่อยู่ในจอภาพจำลองด้วยการ drag เมาส์ ไปยังตำแหน่งที่คุณต้องการ โดยจะมีผลในขณะประมวลผลเท่านั้น

2.6.5 ฟอร์ม

ฟอร์ม (Form) คือ หน้าต่างที่ใช้สำหรับแสดงผล โดยจะมี ActiveX Controls ต่างๆ บรรจุอยู่ภายใน มีหน้าที่สำหรับติดต่อกับผู้ใช้งาน โดย Form ก็ถือว่าเป็นออบเจกต์ด้วย

1) พร็อพเพอร์ตี้ที่สำคัญของ Form ดังตารางที่ 2.1

ตารางที่ 2.1 แสดงพร็อพเพอร์ตี้ที่สำคัญของ Form

พร็อพเพอร์ตี้	คุณสมบัติ
Name	ใช้สำหรับกำหนดชื่อ
BackColor	ใช้สำหรับกำหนดสีพื้น
BorderStyle	ใช้สำหรับกำหนดรูปแบบของเส้นขอบ
Caption	ใช้สำหรับกำหนดข้อความบน Title Bar ของ Form
ControlBox	ใช้สำหรับกำหนดให้มีปุ่มควบคุมของ Form
Enabled	ใช้สำหรับกำหนดให้ Form สามารถใช้งานได้หรือไม่
Font	ใช้สำหรับกำหนดตัวอักษรของข้อความอุปกรณ์ต่าง ๆ
ForeColor	ใช้สำหรับกำหนดสีตัวอักษรของข้อความอุปกรณ์ต่าง ๆ
MaxButton	ใช้สำหรับกำหนดให้มีปุ่มขยายขนาดของ Form
MDI Child	ใช้สำหรับกำหนดให้ Form มีคุณสมบัติเป็น Form ย่อยของ MDI Form
MinButton	ใช้สำหรับกำหนดให้มีปุ่มย่อขนาดของ Form
Moveable	ใช้สำหรับกำหนดให้ Form สามารถย้ายตำแหน่งได้หรือไม่
Picture	ใช้สำหรับกำหนดรูปบน Form
ShowInTaskbar	ใช้สำหรับกำหนดให้มีไอคอนแสดงบน Taskbar
StartPosition	ใช้สำหรับกำหนดตำแหน่งการแสดงผล Form บนจอภาพ
Visible	ใช้สำหรับกำหนดให้ซ่อนหรือแสดง Form
WindowState	ใช้สำหรับกำหนดขนาดของ Form เมื่อมีการทำงาน

2) เมธอดที่สำคัญของ Form ดังตารางที่ 2.2

ตารางที่ 2.2 แสดงเมธอดที่สำคัญของ Form

เมธอด	คุณสมบัติ
Hide	เป็นการทำงานที่สั่งให้ซ่อน Form
Line	เป็นการทำงานที่สั่งให้วาดเส้นลงบน Form
Move	เป็นการทำงานที่สั่งให้ Form ย้ายตำแหน่งไปยังตำแหน่งที่กำหนด
Print	เป็นการทำงานที่สั่งให้พิมพ์ Form ออกทางเครื่องพิมพ์
Show	เป็นการทำงานที่สั่งให้แสดง Form
Unload	เป็นการทำงานที่สั่งให้ยกเลิกการใช้งานของ Form

3) อีเวนต์ที่สำคัญของ Form ดังตารางที่ 2.3

ตารางที่ 2.3 แสดงอีเวนต์ที่สำคัญของ Form

อีเวนต์	คุณสมบัติ
Activate	จะเกิดขึ้นเมื่อเลือกใช้งาน Form กรณีที่มีการเปิด Form หลาย ๆ Form พร้อมกัน
Initialize	จะเกิดขึ้นเมื่อ Form ถูกโหลดเข้าไปในหน่วยความจำ
Load	จะเกิดขึ้นเมื่อ Form แสดงผลหลังจากที่ถูกโหลดเข้าไปในหน่วยความจำ
QueryUnload	จะเกิดขึ้นเมื่อมีการปิด Form
Terminate	จะเกิดขึ้นเมื่อ Form ถูกลบออกจากหน่วยความจำ
Unload	จะเกิดขึ้นเมื่อ Form ถูกยกเลิกการใช้งาน

4) ActiveX Control

ActiveX Control คือเครื่องมือที่ Visual Basic ได้จัดเตรียมไว้ให้นักพัฒนาโปรแกรมเพื่อใช้ในการพัฒนาโปรแกรมประยุกต์ให้เกิดความง่ายและรวดเร็วในการเขียนโปรแกรม

5) Label : แถบอักษร

แถบอักษร หรือ แถบข้อความใช้เพื่อแสดงข้อความ เมื่อแสดงผลจะไม่สามารถพิมพ์แก้ไขข้อมูลได้ นอกจากจะเขียนชุดคำสั่งกำหนดให้มีการเปลี่ยนแปลง

6) พื้นฐานการเขียนโค้ด

มี 2 วิธีที่สามารถเรียก editor ขึ้นมาใช้งานคือ

- ดับเบิลคลิกที่ตัวคอนโทรลนั้น ๆ
- คลิกที่คอนโทรลนั้น ให้อยู่ในสภาพใช้งาน (active) หรือได้รับความสนใจ

(focus) แล้วกด F7

7) การใช้งาน Editor

Editor ถือได้ว่าเป็นส่วนที่มีความสำคัญมากอีกส่วนหนึ่งในบรรดาเครื่องมือที่ Visual Basic มี เพราะใช้สำหรับเขียนโค้ดให้โปรแกรมประยุกต์ทำงานได้ เครื่องมือตัวนี้ต้องใช้งานมากที่สุด ในขบวนการพัฒนาโปรแกรมประยุกต์ด้วย Visual Basic การศึกษาสภาพแวดล้อมของ Editor จึงมีความสำคัญเป็นอย่างยิ่ง สามารถแยกส่วนต่างๆ ของ Editor ออกได้เป็น 3 ส่วนดังนี้

- ส่วน Object List Box มีหน้าที่แสดงชื่อคอนโทรลหรืออ็อบเจกต์ที่ถูกนำมาใช้งาน
- ส่วน Event List Box มีหน้าที่แสดงเหตุการณ์ (Event) ของคอนโทรลที่ถูกเลือกใน Object List Box
- ส่วนการเขียนโค้ด เมื่อเลือกคอนโทรลใน Object List Box และเลือกเหตุการณ์ใน Event List Box แล้ว Visual Basic จะสร้างโพรซีเจอร์ (Procedure) ให้อัตโนมัติ

2.6.6 ความสามารถพิเศษของ Editor

ในการใช้งาน Editor เมื่อพิมพ์ชื่อคอนโทรลแล้วพิมพ์ Editor จะแสดง ToolTip ที่เป็นรายการพรีอเพอร์ตี้หรือรายการเมธอดที่คอนโทรลนั้นสนับสนุนอยู่ขึ้นมาทันที ช่วยให้ไม่ต้องจำว่าคอนโทรลนี้มีพรีอเพอร์ตี้หรือมีเมธอดอะไรบ้าง รวมถึงป้องกันไม่ให้พิมพ์ผิดอีกด้วย และถ้ามีการเรียกใช้งานฟังก์ชันมาตรฐานต่างๆ ToolTip ก็ จะแสดงรูปแบบไวยากรณ์ของฟังก์ชันนั้นๆ ให้ทันทีเช่นกัน ความสามารถของ Editor อีกอย่างก็คือสามารถตรวจสอบไวยากรณ์ (Syntax) ตามโครงสร้างของภาษา Visual Basic ได้อีกด้วย โดยขณะที่พิมพ์โค้ดเข้าไปเมื่อกด Enter จบบรรทัด

Visual Basic จะทำงานตรวจสอบไวยากรณ์ทันที ถ้ามีข้อผิดพลาด ในการใช้งานไวยากรณ์เกิดขึ้น Visual Basic จะแสดงข้อความช่วยเหลือที่เกี่ยวข้องกับความผิดพลาดนั้นๆ ขึ้นมาทันที

2.6.7 การใช้ Message Box

1) Message Box เป็นเครื่องมือที่ใช้โต้ตอบกับผู้ใช้ โดยจะแสดงข้อมูลเพียงอย่างเดียวแล้วให้ผู้ใช้ Click ปุ่มเลือกในกรณีที่ต้องการให้ผู้ใช้เลือกตอบ รูปแบบการใช้งานคือ MsgBox Prompt[,Buttons][,Title]

2) Prompt คือข้อความที่ต้องการแสดงใน Message Box ในกรณีที่ต้องการ แสดงข้อมูลหลายบรรทัดทำได้โดยเชื่อมกับ chr (13)

3) Buttons คือส่วนที่ใช้กำหนดการแสดงผลปุ่มและกำหนดรูปไอคอนบน Message Box

4) Title คือส่วนของข้อความที่ต้องการแสดงบนแถบด้านบนของ Message Box การกำหนดปุ่มและไอคอนของปุ่มสามารถทำได้โดยการระบุค่าคงที่ของแต่ละอย่างเชื่อมด้วยเครื่องหมาย + ซึ่งรายละเอียดของค่าคงที่สามารถใช้งานได้มีดังนี้

- กลุ่มที่ใช้สำหรับกำหนดปุ่มที่จะแสดงใน Message Box ค่าคงที่มีรายละเอียด ดังแสดงในตารางที่ 2.4

ตารางที่ 2.4 แสดงกลุ่มที่ใช้สำหรับกำหนดปุ่มที่จะแสดงใน Message Box

กลุ่ม	ผลการแสดงผลปุ่มใน Message Box
VbOKOnly	แสดงผลปุ่ม OK ปุ่มเดียว
VbOKCancel	แสดงผลปุ่ม OK และ Cancel
VbYesNo	แสดงผลปุ่ม Yes และ No
VbYesNoCancel	แสดงผลปุ่ม Yes No และ Cancel
VbAbortRetryIgnore	แสดงผลปุ่ม Abort Retry และ Ignore
VbRetryCancel	แสดงผลปุ่ม Retry และ Cancel

- กลุ่มที่ใช้สำหรับกำหนดไอคอนที่จะแสดงใน Message Box ค่าคงที่มีรายละเอียด ดังแสดงในตารางที่ 2.5



ตารางที่ 2.5 แสดงกลุ่มที่ใช้สำหรับกำหนดไอคอนที่จะแสดงใน Message Box

กลุ่ม	ผลการแสดงไอคอน
VbCritical	แสดงไอคอน Critical Message
VbExclamation	แสดงไอคอน Warning Message
VbInformation	แสดงไอคอน Information Message
VbQuestion	แสดงไอคอน Question Message

๗
T๗
153
ก125ก
255๗

- กลุ่มที่ใช้สำหรับกำหนดปุ่มเริ่มต้นค่าคงที่มีรายละเอียดดังแสดงในตารางที่

2.6

ตารางที่ 2.6 แสดงกลุ่มที่ใช้สำหรับกำหนดปุ่มเริ่มต้นค่าคงที่

กลุ่ม	ผลการกำหนด
VbDefaultButton1	กำหนดให้ปุ่มแรกเป็นปุ่มเริ่มต้น
VbDefaultButton2	กำหนดให้ปุ่มที่ 2 เป็นปุ่มเริ่มต้น
VbDefaultButton3	กำหนดให้ปุ่มที่ 3 เป็นปุ่มเริ่มต้น
VbDefaultButton4	กำหนดให้ปุ่มที่ 4 เป็นปุ่มเริ่มต้น

2.6.8 การใช้ Input Box

Input Box เป็นเครื่องมือที่รับข้อมูลโดยให้ผู้ใช้ป้อนข้อมูลลงไป แล้วเก็บข้อมูลนั้นไว้ในตัวแปร

- 1) รูปแบบการใช้งาน คือ Input Box(Prompt[,Title][,Default])
- 2) Prompt คือข้อความที่ต้องการแสดงใน Input Box ในกรณีที่ต้องการ แสดงข้อมูลหลายบรรทัดทำได้โดยเชื่อมกับ chr (13)
- 3) Title คือส่วนของข้อความที่ต้องการแสดงบนแถบด้านบนของ Input Box
- 4) Default คือค่าที่กำหนดให้กรณีที่ไม่มีการป้อนข้อมูลใน Input Box

2.6.9 การสร้างเมนู

ในการพัฒนาโปรแกรมประยุกต์ 1 โปรแกรมจะประกอบไปด้วยฟอร์มจำนวนมาก ในการเรียกใช้งานฟอร์มแต่ละฟอร์มจำเป็นต้องมีเมนูมาช่วยจัดหมวดหมู่ของฟอร์ม เพื่อให้ง่ายต่อการใช้งาน การสร้างเมนูใน Visual Basic สามารถทำได้โดยเรียกใช้คำสั่งสำหรับสร้างเมนูโดยกดปุ่มขวาบนฟอร์มที่ต้องการสร้างเมนู เลือกคำสั่ง Menu Editor

2.7 หลักการทำคู่มือและแนวทางการนำคู่มือไปใช้

รูปแบบของข้อมูลเกี่ยวกับหลักการทำคู่มือและแนวทางการนำคู่มือไปใช้จะแสดงอยู่ในรูปแบบของเอกสาร โดยมีข้อมูลโดยสรุป ดังนี้

2.7.1 หลักการและแนวทางที่ใช้ในการจัดทำคู่มือการใช้งาน ซึ่งประกอบด้วยหลักการคร่าวๆ ดังนี้

1) เนื้อหาส่วนใหญ่ของคู่มือการใช้งานจะต้องเป็นการบรรยายขั้นตอนเพื่อให้ผู้ใช้ติดตามและดำเนินการตามขั้นตอนนั้นได้จนบรรลุวัตถุประสงค์ของการใช้งานในครั้งหนึ่งๆ ดังนั้นการบรรยายส่วนใหญ่จึงไม่ใช่การอธิบายที่ละเอียดแต่เป็นการแบ่งหัวเรื่องให้ตรงตามความต้องการทำงานของผู้ใช้

2) ควรมีการแบ่งส่วนของเนื้อหาในคู่มือตามผู้ใช้เป็นส่วนๆ หมายถึงว่าผู้ใช้ที่มีหน้าที่ที่ต่างกันโดยชัดเจนสามารถเปิดค้นหาเนื้อหาเฉพาะที่เกี่ยวข้องกับตนได้ใน (section) หนึ่งๆ เช่น ถ้าระบบมีการแยกการใช้งานผู้ดูแลระบบ (Admin) ผู้บริหารและผู้ใช้ทั่วไปอย่างชัดเจนก็ควรจัดทำคู่มือเป็น สำหรับ ผู้ดูแลระบบสำหรับผู้บริหารและสำหรับผู้ใช้ทั่วไป เป็น 3 ส่วน (section) เป็นต้น

3) ในกรณีที่มีรูปภาพหน้าจอประกอบและมีการอ้างถึงส่วนใดส่วนหนึ่งในภาพหน้านั้น ควรใช้ตัวอักษรในรูปแบบ (font) ต่างจากตัวอักษรที่ใช้ปกติ

4) ในการแสดงผลการวิเคราะห์หรือการแสดงผลงานที่มีความซับซ้อนสามารถสร้างเป็นหัวข้อเพื่ออธิบายวิธีการทำความเข้าใจกับผลการวิเคราะห์หรือรายงานนั้นได้ในคู่มือการใช้งาน

2.7.2 โครงสร้างของคู่มือการใช้งาน

1) การแนะนำระบบคือการแนะนำข้อมูลคร่าวๆ ของคู่มือบางส่วนว่ามีความเกี่ยวข้องกับอะไร หรือข้อมูลนี้มีจุดประสงค์ในการจัดทำขึ้นเพื่ออะไร

2) ส่วนหลักของคู่มือการใช้งานคือส่วนที่แสดงรายละเอียดที่ผู้จัดทำคู่มือต้องการแสดงข้อมูลให้ผู้ใช้ได้เข้าใจเกี่ยวกับหลักการหรือข้อมูลต่างๆ ซึ่งผู้ใช้สามารถเปิดคู่มือและปฏิบัติตามขั้นตอนที่ระบุไว้ในคู่มือ และสามารถสำเร็จได้ตามวัตถุประสงค์ในการใช้ในแต่ละครั้ง

3) ส่วนปิดท้ายคือ บทสุดท้ายในแต่ละส่วน (section) ของคู่มือการใช้งาน สามารถใช้เพื่อให้ข้อมูลแก่ผู้ใช้ ในเรื่องที่ยังไม่ครอบคลุมใน 2 ส่วนต้น เช่น การใช้งานส่วนช่วยเหลือ เป็นต้น

