

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

2.1 ความหมายของระบบฐานข้อมูล

ศิริช กัญจนนุช และวิชาญ หงษ์บิน (2542, หน้า 3-25) ฐานข้อมูล (Database) หมายถึง กลุ่มของข้อมูล (Data Group) ที่ถูกรวบรวมเข้าไว้ด้วยกัน โดยครอบคลุมรายละเอียดต่าง ๆ เช่น ในสำนักงานก็จะรวบรวมข้อมูลตั้งแต่หมายเลขโทรศัพท์ของผู้ที่ติดต่อจนถึงการเก็บเอกสาร ทุกอย่างของสำนักงาน ซึ่งข้อมูลจะมีส่วนที่สัมพันธ์กันและเป็นที่ต้องการออกมาใช้ให้เป็นประโยชน์ ข้อมูลนั้นอาจจะเกี่ยวกับบุคคล สิ่งของ สถานที่ หรือเหตุการณ์ใด ๆ ที่เราสนใจศึกษา ซึ่งข้อมูล (Data) อาจจะได้มาจากการสังเกต การนับหรือการวัด และข้อมูลอาจเป็นได้ทั้งตัวเลขหรือเป็นข้อความก็ได้ ที่สำคัญคือข้อมูลจะต้องเป็นสิ่งที่มีความจริง รายละเอียดของข้อมูลต่างๆ ต้องนำมาเชื่อมโยงสัมพันธ์กันให้ตรงตามต้องการเพื่อสะดวกในการค้นหาและกรอกข้อมูลเพิ่มเติม

ปรีชา อัสวเดชาบุตร และเสาวรส ใหญ่สว่าง (2531) ได้กล่าวถึงฐานข้อมูลว่าเป็นหัวใจสำคัญของการประมวลผลข้อมูลในองค์กร ข้อมูลที่หน่วยงานต่างๆในองค์กรต้องการใช้จะถูกเก็บไว้ในที่เดียวกัน ข้อมูลเหล่านี้จะสัมพันธ์กัน แต่จะไม่มีข้อมูลที่ซ้ำซ้อนกัน สามารถตอบสนองความต้องการของผู้ใช้ได้ โดยใช้ระบบจัดการฐานข้อมูล (Database Management System; DBMS) ช่วยในการจัดเก็บข้อมูล แก้ไขข้อมูล เพิ่มเติมข้อมูล และค้นหาข้อมูล ซึ่งอำนวยความสะดวก แก่ผู้ใช้

จรนิต แก้วกัจจาวล (2536) ให้ความหมายของฐานข้อมูลไว้ว่าเป็นการรวบรวมข้อมูลที่สัมพันธ์กัน และกำหนดรูปแบบการจัดเก็บอย่างเป็นระบบ การจัดเก็บเป็นฐานข้อมูลมักจะจัดเก็บไว้ที่หน่วยศูนย์กลาง เพื่อให้ผู้ใช้หลายๆ หน่วยงานในองค์กร สามารถเรียกใช้ข้อมูลที่จัดเก็บไว้ได้ตามความต้องการของแต่ละหน่วยงาน ซึ่งอาจถูกเรียกใช้เสมอ เป็นข้อมูลที่ใช้เป็นประจำ

วาสนา สุขกระสานติ (2541) กล่าวว่า ระบบฐานข้อมูล หมายถึง ระบบสำหรับสร้างแฟ้มข้อมูลต่างๆ เก็บไว้ในสื่ออิเล็กทรอนิกส์ โดยมีเครื่องมือในการอำนวยความสะดวกเกี่ยวกับการจัดการแฟ้มข้อมูล เช่น การเพิ่ม หรือแก้ไขข้อมูลที่จัดเก็บอยู่ หรือสามารถเรียกแฟ้มข้อมูลนั้นขึ้นมาแสดงโดยกำหนดเงื่อนไขให้เลือกข้อมูลมาแสดง

2.2 องค์ประกอบของระบบฐานข้อมูล

องค์ประกอบของระบบฐานข้อมูลในที่นี้ หมายถึง โครงสร้างสารสนเทศ (Information) ที่ประกอบด้วย Entity หลาย ๆ ตัว ซึ่งบรรดา Entity เหล่านี้จะต้องมีความสัมพันธ์กัน

2.2.1 Entity

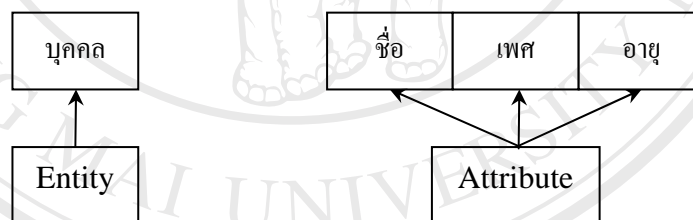
หมายถึง สิ่งที่มีอยู่จริง อันได้แก่ บุคคล สถานที่ และสิ่งของ ซึ่งตัวอย่าง Entity ในระบบฐานข้อมูลของโรงเรียนอาจได้แก่ อาจารย์ แผนกวิชา ประวัติการทำงาน หรือถ้าเป็น Entity ของบริษัทก็อาจจะได้แก่ พนักงานขาย ลูกค้า การสั่งซื้อ และประเภท สินค้า เป็นต้น

2.2.2 Attribute

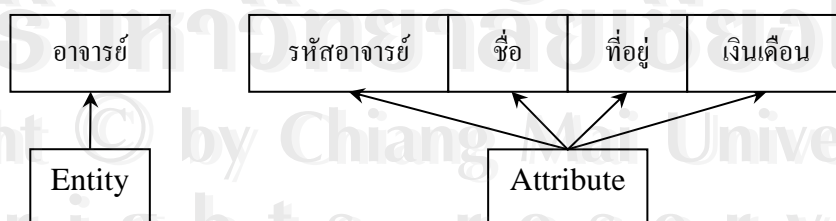
คือ ส่วนที่แสดงลักษณะและคุณสมบัติของข้อมูล เช่น

1. Attribute ของบุคคล ได้แก่ ชื่อ เพศ อายุ ฯลฯ
2. Attribute ของอาจารย์ ได้แก่ รหัสอาจารย์ ชื่อ ที่อยู่ เงินเดือน ฯลฯ
3. Attribute ของประเภทสินค้า ได้แก่ รหัสสินค้า สี ราคา ฯลฯ

ซึ่งพอจะแจกแจงได้ดังนี้



รูป 2.1 แสดง Entity ของบุคคล



รูป 2.2 แสดง Entity ของอาจารย์

จึงเปรียบได้ว่า Entity นั้นเหมือนกับแฟ้มข้อมูล ส่วน Attribute นั้นเหมือนกับเขตข้อมูล

2.2.3 เขตข้อมูลหรือฟิลด์ (Field)

หมายถึง ตัวอักษร (Character) แต่ละตัวที่ถูกนำมาประกอบกันเป็นกลุ่มคำที่มีความหมายขึ้น เช่น กลุ่มตัวอักษรที่ประกอบกันเป็นชื่อหรือนามสกุล กลุ่มตัวเลขที่ประกอบกันเป็นรหัสประจำตัว, ราคา โดยเรียกกลุ่มข้อมูลนี้ว่า เขตข้อมูล เช่น เขตข้อมูลชื่อ เขตข้อมูลนามสกุล เขตข้อมูลรหัสประจำตัว เป็นต้น

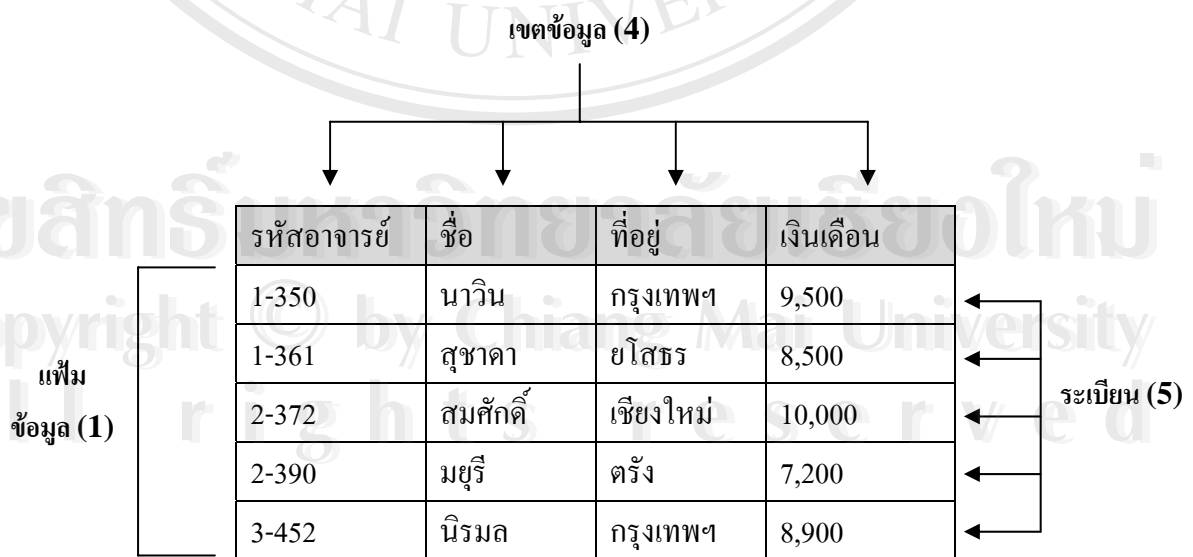
2.2.4 ตัวอักษร

หมายถึงข้อมูลพื้นฐานที่เล็กที่สุดภายในแฟ้มข้อมูล ซึ่งก็คือ บิต (Bit : Binary Digit) บิตเป็นหน่วยข้อมูลพื้นฐานที่เก็บอยู่ในหน่วยความจำภายในคอมพิวเตอร์ โดยจะแทนด้วยตัวเลข 1 ตัว ได้แก่ 1 หรือ 0 อย่างไม่อย่างหนึ่ง เรียกตัวเลข 1 หรือ 0 นี้ว่า เป็นบิต 1 บิต

ข้อมูลซึ่งได้แก่ ตัวอักษรแต่ละตัวเช่น A, B,Z, 0, 1, 2,, 9 และสัญลักษณ์พิเศษอื่น ๆ เช่น \$, &, +, -, *, / ฯลฯ เมื่อจะถูกนำไปเก็บไว้ในคอมพิวเตอร์จะต้องถูกแปลงให้อยู่ในรูปของบิตหลายบิตที่มาประกอบกัน โดยตัวอักษร 1 ตัวจะแทนด้วยบิต 7 หรือ 8 บิต ซึ่งจะเรียก ได้อีกอย่างว่า ไบต์ (Byte) ตัวอย่างเช่น ตัวอักษร A เมื่อเก็บอยู่ในคอมพิวเตอร์จะเก็บเป็น 1000001 ตัวอักษร B จะเก็บเป็น 100010 เป็นต้น

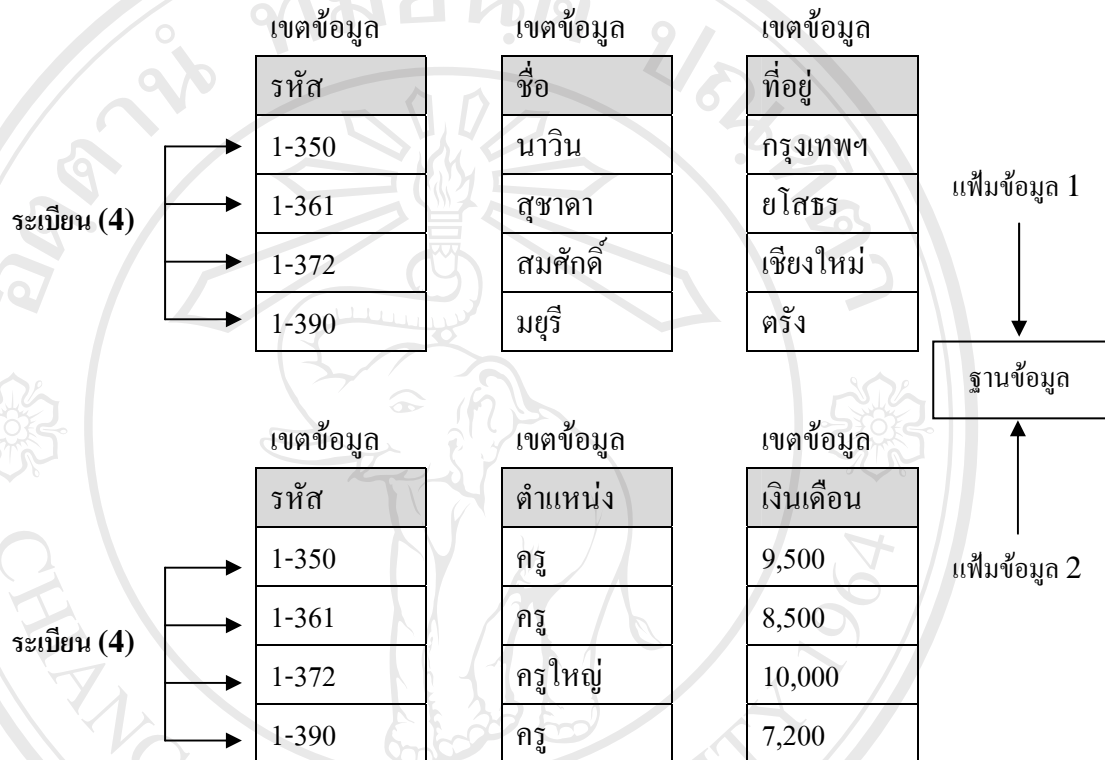
2.2.5 ระเบียบหรือเรคคอร์ด (Record)

คือการนำข้อมูลในหลายเขตข้อมูลหรือฟิลด์มา รวมกันซึ่งมีลักษณะเป็นแถว ซึ่งสามารถจะนำระเบียบแต่ละระเบียบของข้อมูลชนิดเดียวกันมา รวมกันเป็นแฟ้มข้อมูลหรือไฟล์ (File) ได้ดังตัวอย่างต่อไปนี้



รูป 2.3 แสดงรายละเอียดของแฟ้มข้อมูล

ถ้า นำเพิ่มข้อมูลซึ่งประกอบด้วยเขตข้อมูลมารวมกัน และเพิ่มข้อมูลนั้นมีความสัมพันธ์ซึ่งกันและกัน เราจะเรียกว่า **ฐานข้อมูล**



รูป 2.4 แสดงความสัมพันธ์ของตาราง

2.2.6 พจนานุกรมข้อมูล (Data Dictionary) หรือเมตาดาต้า (Meta Data)

เป็นส่วนที่เก็บคำอธิบายเกี่ยวกับ โครงสร้างของฐานข้อมูล และมีหน้าที่เก็บรายละเอียดของข้อมูลภายในฐานข้อมูลอธิบายลักษณะของข้อมูล รวมถึงการระบุความสัมพันธ์ของข้อมูลด้วย

นอกจากนี้แล้ว พจนานุกรมข้อมูลยังทำหน้าที่สำคัญดังต่อไปนี้ด้วย

1. ควบคุมระบบรักษาความปลอดภัยของข้อมูล และความคงสภาพของข้อมูล (Data Security and Data Integrity)
2. ควบคุมเกี่ยวกับการใช้งานฐานข้อมูลพร้อมกัน (Concurrency Control)

ทั้งนี้ หากเปรียบฐานข้อมูลเป็นห้องสมุด หนังสือในเก็บอยู่ในห้องสมุดก็เปรียบได้กับข้อมูลที่เก็บอยู่ในฐานข้อมูล ซึ่งการเก็บบัญชีรายชื่อหนังสือก็เปรียบได้กับพจนานุกรมข้อมูล

คือบอกถึงรายละเอียดเกี่ยวกับหนังสือแต่ละเล่มนั้น โดยจะถูกเก็บและจะถูกเรียกใช้งานในระหว่างที่มีการประมวลผลฐานข้อมูล

2.3 ความสัมพันธ์และรูปแบบฐานข้อมูล

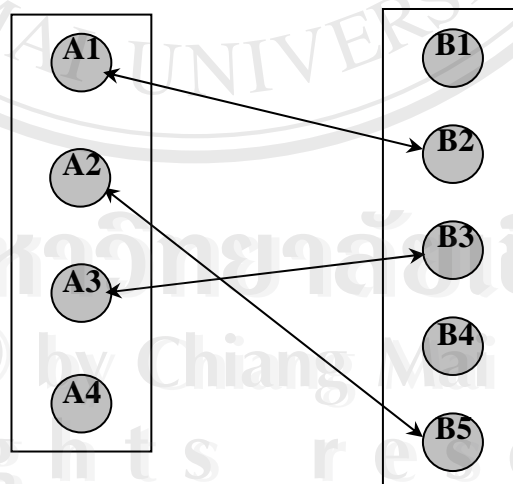
2.3.1 ความสัมพันธ์ (Relationship)

ในฐานข้อมูลหนึ่งๆ นั้น มักจะประกอบด้วยเพิ่มข้อมูลหลายเพิ่ม แต่ละเพิ่มเก็บข้อมูลแตกต่างกันออกไป จึงต้องมีการกำหนดความสัมพันธ์ระหว่างข้อมูลต่างๆ ที่ต้องการเก็บ โดยสามารถจำแนกชนิดของความสัมพันธ์นี้ได้ 3 ชนิด คือ

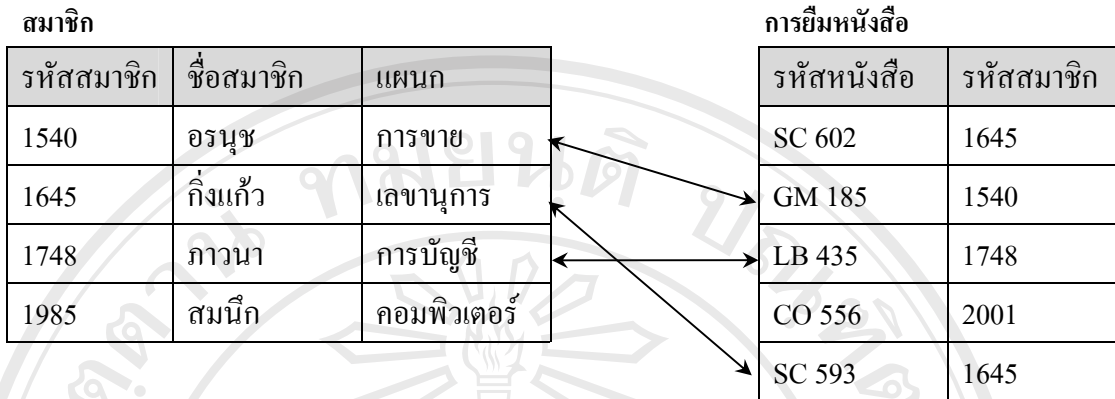
1. ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One-to-One Relationship)
2. ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One-to-Many Relationship)
3. ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many-to-Many Relationship)

1. ความสัมพันธ์แบบหนึ่งต่อหนึ่ง

เป็นลักษณะความสัมพันธ์ที่มีระเบียบเพียง 1 ระเบียบใน Entity A ที่มีความสัมพันธ์กับระเบียบเพียง 1 ระเบียบใน Entity B และในทางกลับกัน ระเบียบเพียง 1 ระเบียบใน Entity B จะมีความสัมพันธ์กับระเบียบเพียง 1 ระเบียบใน Entity A



รูป 2.5 แสดงความสัมพันธ์ในลักษณะหนึ่งต่อหนึ่ง

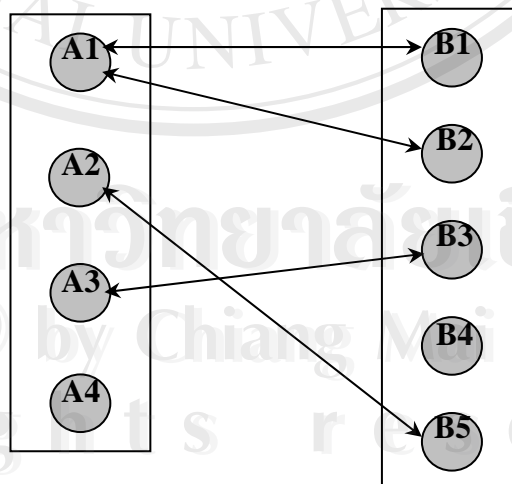


รูป 2.6 อธิบายความสัมพันธ์แบบหนึ่งต่อหนึ่งด้วยตารางความสัมพันธ์

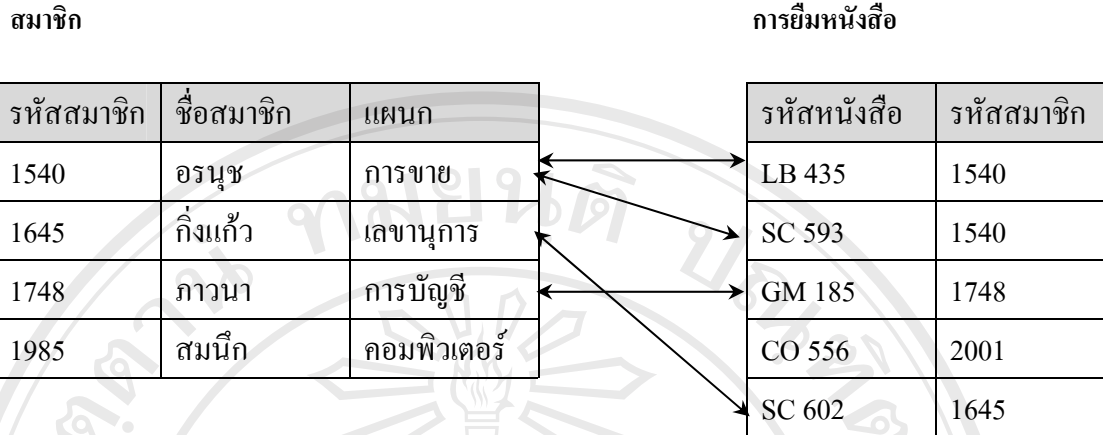
จากตัวอย่าง ความสัมพันธ์ดังกล่าว หมายถึง สมาชิกแต่ละคน สามารถที่จะยืมหนังสือ ได้เพียงคนละ 1 เล่มเท่านั้น และหนังสือเล่มดังกล่าวก็ไม่สามารถให้ใครยืมได้อีก เพราะมีเพียงเล่มเดียวในทันทีที่จะสังเกตได้ว่าทั้ง 2 Entity จะมีคีย์หรือข้อมูลหลักคือ รหัสสมาชิก

2. ความสัมพันธ์แบบหนึ่งต่อกลุ่ม

เป็นลักษณะความสัมพันธ์ที่ระเบียบหนึ่งระเบียบใน Entity A มีความสัมพันธ์กับระเบียบหลายระเบียบใน Entity B และในทางกลับกันหลายระเบียบใน Entity B จะมีความสัมพันธ์กับระเบียบเพียงหนึ่งระเบียบใน Entity A



รูป 2.7 แสดงความสัมพันธ์ในลักษณะหนึ่งต่อกลุ่ม

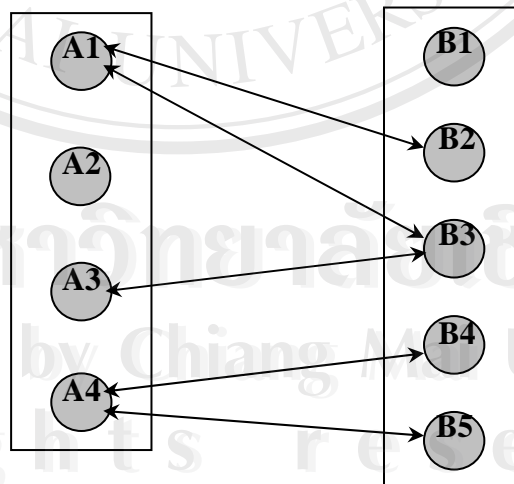


รูป 2.8 อธิบายความสัมพันธ์แบบหนึ่งต่อกลุ่มด้วยตารางความสัมพันธ์

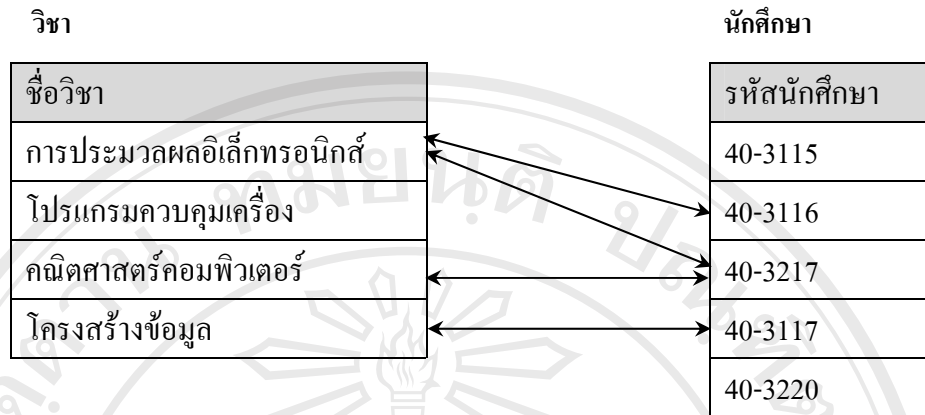
จากตัวอย่าง สมาชิกที่มีรหัสสมาชิก 1540 สามารถยืมหนังสือได้ 2 เล่ม คือ รหัสหนังสือ LB 435 และ SC 593 แต่หนังสือทั้ง 2 เล่ม จะถูกยืมด้วยสมาชิกได้เพียงคนเดียว

3. ความสัมพันธ์แบบกลุ่มต่อกลุ่ม

เป็นความสัมพันธ์ที่แต่ละระเบียนใน Entity A จะมีความสัมพันธ์กับระเบียนหลายระเบียนใน Entity B และในทางกลับกันแต่ละระเบียนใน Entity B จะมีความสัมพันธ์กับระเบียนที่อยู่ใน Entity A



รูป 2.9 แสดงความสัมพันธ์ในลักษณะกลุ่มต่อกลุ่ม



รูป 2.10 อธิบายความสัมพันธ์แบบกลุ่มต่อกลุ่มด้วยตารางความสัมพันธ์

จากตัวอย่าง ในแต่ละรายวิชา จะสามารถมีนักศึกษาเรียนได้มากกว่า 1 คน และในทางกลับกันนักศึกษาแต่ละคนก็สามารถเลือกเรียนวิชาได้มากกว่าหนึ่งรายวิชา

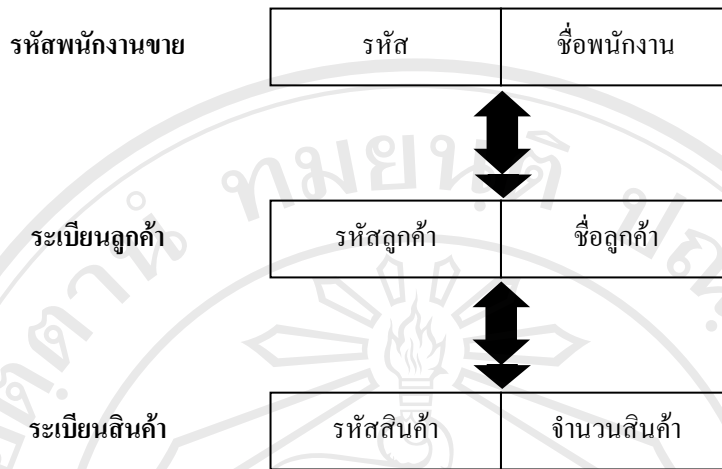
2.3.2 รูปแบบของฐานข้อมูล

โครงสร้างของข้อมูลโดยทั่วไปจะมี 3 แบบด้วยกัน คือ

1. ฐานข้อมูลแบบลำดับชั้น (Hierarchical Database)
2. ฐานข้อมูลแบบเครือข่าย (Network Database)
3. ฐานข้อมูลแบบเชิงสัมพันธ์ (Relational Database)

1. ฐานข้อมูลแบบลำดับชั้น

ลักษณะของโครงสร้างของฐานข้อมูลแบบลำดับชั้นนี้ จะมีลักษณะคล้ายต้นไม้ที่คว่ำหัวลงจึงอาจเรียกโครงสร้างข้อมูลแบบนี้ได้อีกแบบว่าเป็น โครงสร้างแบบต้นไม้ (Tree Structure) โดยจะมีระเบียบที่อยู่ด้านบนซึ่งจะเรียกว่าเป็น ระเบียบพ่อแม่ (Parent Record) ระเบียบในแถวถัดลงมาจะเรียกว่าระเบียบลูก (Child Record) ซึ่งระเบียบพ่อแม่จะสามารถมีระเบียบลูกได้มากกว่าหนึ่งระเบียบ แต่ระเบียบลูกแต่ละระเบียบจะมีพ่อแม่เพียงหนึ่งระเบียบเท่านั้น

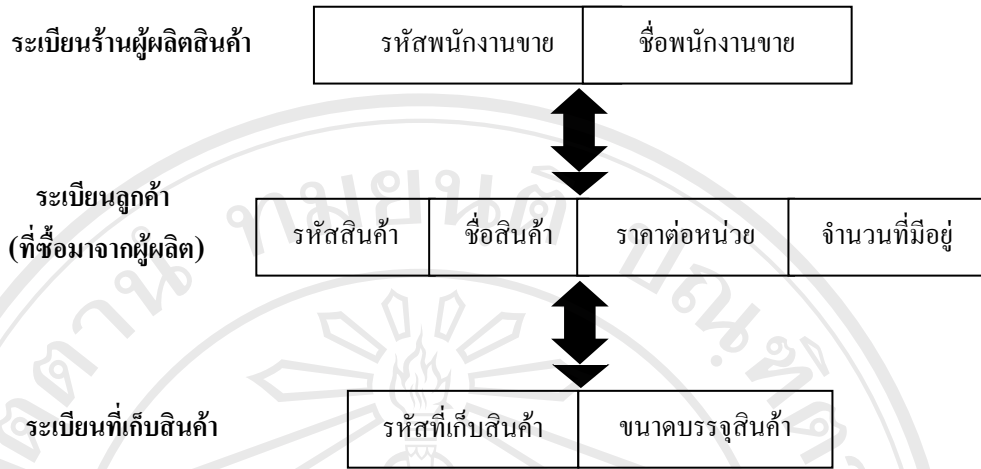


รูป 2.11 แสดงโครงสร้างฐานข้อมูลแบบลำดับชั้น

จากตัวข้างต้นจะเห็นว่า ลูกค้าแต่ละคนจะไม่สามารถได้รับบริการจากพนักงานขายมากกว่าหนึ่งคนได้ เนื่องจากลูกค้าแต่ละคนถือว่าเป็นระเบียบลูก ส่วนพนักงานขายจะถือว่าเป็นระเบียบพ่อแม่ของลูกค้า สินค้าแต่ละชนิดก็จะถูกซื้อ โดยลูกค้าเพียงคนเดียวเท่านั้น เนื่องจากสินค้าแต่ละชนิด จะเป็นระเบียบลูกของระเบียบลูกค้า เป็นต้น

2. ฐานข้อมูลแบบเครือข่าย

ข้อมูลภายในฐานข้อมูลแบบนี้ สามารถมีความสัมพันธ์กันแบบใดก็ได้ เช่น อาจเป็นแบบหนึ่งต่อหนึ่ง หรือหนึ่งต่อกลุ่ม ตัวอย่างของฐานข้อมูลแบบนี้ เช่น การสั่งซื้อสินค้าจากร้านผู้ผลิตสินค้า และการนำสินค้าไปเก็บในคลังสินค้า ซึ่งจะแสดงความสัมพันธ์ระหว่างระเบียบร้านผู้ผลิตสินค้าและระเบียบสินค้า และความสัมพันธ์ระหว่างระเบียบสินค้าและระเบียบที่เก็บสินค้าได้ โดยการใช้ลูกศรเชื่อมโยงเช่นกัน ดังรูป 2.12



รูป 2.12 แสดงโครงสร้างฐานข้อมูลแบบเครือข่าย

3. ฐานข้อมูลแบบเชิงสัมพันธ์

ฐานข้อมูลเชิงสัมพันธ์ จะมีโครงสร้างข้อมูลต่างจากฐานข้อมูลสองแบบแรก คือ ข้อมูลจะถูกเก็บอยู่ในรูปแบบของตาราง (Table) ซึ่งภายในตารางก็จะแบ่งออกเป็นแถว (Row) และคอลัมน์ (Column) แต่ละตารางจะมีจำนวนแถวได้หลายแถวและจำนวนคอลัมน์ได้หลายคอลัมน์ แถวแต่ละแถวจะสามารถเรียกได้อีกชื่อว่า ระเบียบหรือเรคคอร์ด (Record) คอลัมน์แต่ละคอลัมน์สามารถเรียกได้อีกชื่อว่า เขตข้อมูลหรือฟิลด์ (Field)

2.4 การพัฒนาฐานข้อมูล และระบบจัดการฐานข้อมูล

2.4.1 การพัฒนาฐานข้อมูล

ลูมิส (Loomis, 1987) กล่าวว่า การพัฒนาฐานข้อมูลที่จะนำขึ้นมาใช้ในองค์กร จะต้องมีการดำเนินการเป็นขั้นตอน ในการวางแผนพัฒนาออกแบบระบบของฐานข้อมูลควรใช้บุคลากรจากหลายๆ ฝ่ายร่วมมือกันเพื่อให้ฐานข้อมูลมีประสิทธิภาพ ซึ่งควรประกอบด้วยบุคคลดังต่อไปนี้

1. ผู้ใช้ (users)
2. ผู้บริหารข้อมูล (data administrators)
3. นักวิเคราะห์ (analysts)
4. ผู้เขียนโปรแกรม (programmers)

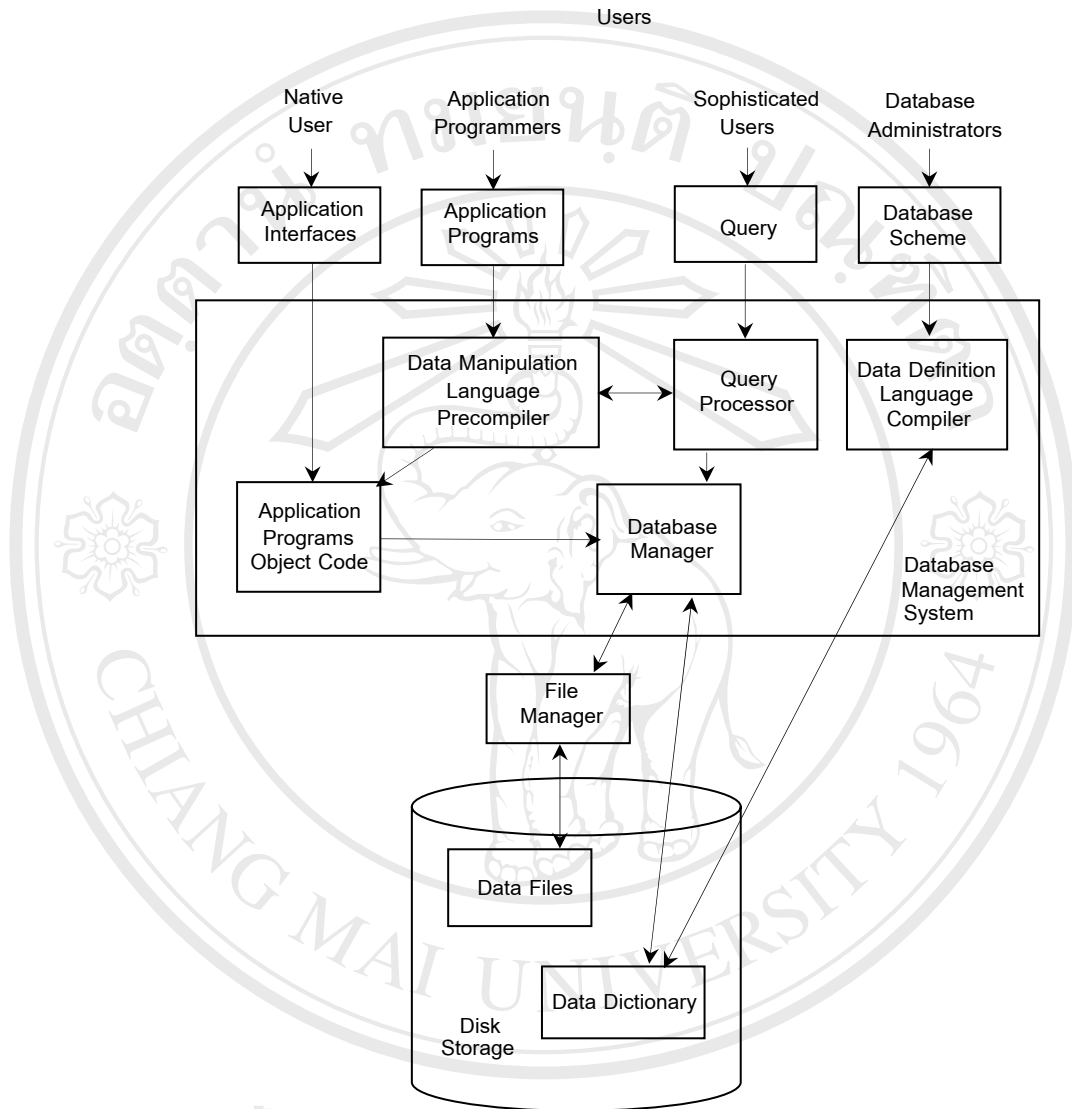
5. ผู้จัดการฐานข้อมูล (database administrators)

2.4.2 ระบบจัดการฐานข้อมูล (Database Management System ; DBMS)

กิตติ ภัคดีวัฒนะกุล และ จำลอง ครุอุตสาหะ (2544) ระบบจัดการฐานข้อมูล เป็นโปรแกรมที่ทำหน้าที่เป็นตัวกลางในการติดต่อระหว่างผู้ใช้กับฐานข้อมูล เพื่อจัดการและควบคุมความถูกต้อง ความซ้ำซ้อน และความสัมพันธ์ระหว่างข้อมูลต่างๆ ภายในฐานข้อมูล ซึ่งต่างจากระบบเพิ่มข้อมูลที่หน้าที่เหล่านี้จะเป็นหน้าที่ของโปรแกรมเมอร์ ในการติดต่อกับข้อมูลในฐานข้อมูลไม่ว่าจะด้วยการใช้คำสั่งในกลุ่มคำสั่ง DML (Data Manipulation Language) หรือกลุ่มคำสั่ง DDL (Data Definition Language) หรือจะด้วยโปรแกรมต่างๆ ทุกคำสั่งที่ใช้กระทำกับข้อมูลจะถูกโปรแกรม DBMS นำมาแปล (Compile) เป็นการกระทำ (Operation) ต่างๆ ภายใต้คำสั่งนั้นๆ เพื่อนำไปกระทำกับตัวข้อมูลภายในฐานข้อมูลต่อไป สำหรับส่วนการทำงานต่างๆ ภายในโปรแกรม DBMS ที่ทำหน้าที่ในการแปลคำสั่งไปเป็นการกระทำต่างๆ ที่จะกระทำกับข้อมูลนั้น ประกอบด้วยส่วนการทำงานต่างๆ ดังนี้

1. Database Manager เป็นส่วนที่ทำหน้าที่กำหนดการกระทำต่างๆ ให้กับส่วน File Manager เพื่อไปกระทำกับข้อมูลที่เก็บอยู่ในฐานข้อมูล (File Manager เป็นส่วนที่ทำหน้าที่บริหาร และจัดการกับข้อมูลที่เก็บอยู่ในฐานข้อมูลในระดับกายภาพ)
2. Query Processor เป็นส่วนที่ทำหน้าที่แปลงประโยคคำสั่งของ Query Language ให้อยู่ในรูปแบบของคำสั่งที่ Database Manager เข้าใจ
3. Data Manipulation Language Precompiler เป็นส่วนที่ทำหน้าที่แปล (Compile) ประโยคคำสั่งของกลุ่มคำสั่ง DML ให้อยู่ในรูปแบบที่ Application Programs Object Code จะนำไปเข้ารหัสเพื่อส่งไปส่วน Database Manager โดยทำงานร่วมกับส่วน Query Processor
4. Data Definition Language Precompiler เป็นส่วนที่ทำหน้าที่แปล (Compile) ประโยคคำสั่งของกลุ่มคำสั่ง DDL ให้อยู่ในรูปแบบของ MetaData ที่เก็บอยู่ในส่วน Data Dictionary ของฐานข้อมูล (MetaData ได้แก่ รายละเอียดที่บอกถึงโครงสร้างต่างๆ ของข้อมูล)
5. Application Programs Object Code เป็นส่วนที่ทำหน้าที่แปลงคำสั่งต่างๆ ของโปรแกรม รวมทั้งคำสั่งในกลุ่มคำสั่ง DML ที่ส่งต่อมาจากส่วน Data Manipulation Language Precompiler ให้อยู่ในรูปของ Object Code ที่จะส่งต่อไปให้ Database Manager เพื่อกระทำกับข้อมูลในฐานข้อมูล

ทั้ง 5 ส่วนของโปรแกรม DBMS สามารถแสดงด้วยแผนภาพได้ดังรูป



รูป 2.13 แสดงส่วนประกอบของโปรแกรม DBMS

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่

Copyright © by Chiang Mai University

โปรแกรม DBMS นี้ ได้ถูกพัฒนาขึ้นมาเพื่อแก้ปัญหาทางด้าน Data Independence ที่ ไม่มีในระบบแฟ้มข้อมูล ดังนั้นจึงมีความเป็นอิสระจากทั้งตัว Hardware และ ตัวข้อมูลภายใน ฐานข้อมูล กล่าวคือ โปรแกรม DBMS จะมีการทำงานที่ไม่ขึ้นกับรูปแบบ (Platform) ของตัว Hardware ที่นำมาใช้กับระบบฐานข้อมูล รวมทั้งมีรูปแบบในการอ้างอิงข้อมูลที่ไม่ขึ้นอยู่กับ โครงสร้างทางกายภาพของข้อมูล ด้วยการ ใช้ Query Language ในการติดต่อกับ ข้อมูลในฐานข้อมูลแทนคำสั่งของภาษาคอมพิวเตอร์ในยุคที่ 3 ส่งผลให้ผู้ใช้สามารถเรียกใช้ข้อมูล

จากฐานข้อมูล โดยไม่จำเป็นต้องทราบถึงประเภทของข้อมูลหรือขนาดของข้อมูลนั้น หรือสามารถกำหนดลำดับที่ของ Field ในการแสดงผลได้โดยไม่ต้องคำนึงถึงลำดับที่จริงของ Field นั้น

2.4.3 การเรียกใช้ การเพิ่มเติมหรือเปลี่ยนแปลงแก้ไขข้อมูลฐานข้อมูล

มีวิธีการต่างๆ ดังนี้

1. เชื่อมโยงกับภาษาการโปรแกรม (Programming Language Interfaces) นิยมใช้วิธีนี้ในการเขียนโปรแกรมที่ต้องการมีการเรียกใช้หรือแก้ไขค่าของข้อมูลในฐานข้อมูลตลอดจนการสร้างรายงานที่มีการคำนวณซับซ้อนอาจใช้ภาษาโคบอล (Cobol) ภาษาซี หรือภาษาในระดับสูงและสูงมากขึ้นในการเชื่อมต่อเข้ากับฐานข้อมูล

2. ภาษาในการจัดการข้อมูล (Query languages) เป็นภาษาที่ถูกออกแบบมาโดยเฉพาะให้ใช้กับฐานข้อมูลซึ่งนิยมใช้กันมากในปัจจุบัน เพราะใช้ง่ายและเรียกดูข้อมูลได้อย่างรวดเร็ว จัดเป็นภาษาในยุคที่สี่ ไม่ต้องมีการแปลภาษา หรือเชื่อมโยงก่อนใช้งาน

3. ตัวสร้างรายงาน (Report Generator) ถูกออกแบบมาให้สร้างรายงานที่ซับซ้อนและมีขนาดใหญ่มากได้อย่างรวดเร็ว

4. โปรแกรมอรรถประโยชน์ของระบบ (System utilities) เป็นโปรแกรมที่ถูกใช้งานโดยผู้จัดการระบบหรือที่นิยมเรียกว่า ผู้ดูแลระบบฐานข้อมูล โปรแกรมประเภทนี้นิยมใช้ในการเก็บสำรองฐานข้อมูล เรียกข้อมูลจากฐานข้อมูล หรือจัดเก็บข้อมูลไว้ในฐานข้อมูล รวมทั้งการเรียกคืนข้อมูลในกรณีที่ระบบมีปัญหา

2.4.4 คุณสมบัติของระบบจัดการฐานข้อมูล

ระบบการจัดการฐานข้อมูลที่ดีควรมีคุณสมบัติดังนี้ (วาสนา สุขกระสานติ, 2540)

1. ต้องมีการใช้งานทรัพยากรของคอมพิวเตอร์อย่างมีประสิทธิภาพ
2. ต้องมีความรวดเร็วในการตอบคำถาม ที่ผู้ใช้งานอยู่ในเกณฑ์ที่ยอมรับได้
3. ต้องมีความเข้ากันได้กับฮาร์ดแวร์ ซอฟต์แวร์ และข้อมูลที่มีใช้งานอยู่เดิมเพื่อลด ค่าใช้จ่ายในการเปลี่ยนแปลงให้เหลือน้อยที่สุด
4. ต้องสามารถเพิ่มหรือลบบันทึกของข้อมูลได้อย่างรวดเร็วและมีประสิทธิภาพ รวมทั้งต้องยืดหยุ่นพอที่จะจัดการกับการเปลี่ยนแปลงหรือเปลี่ยนรูปแบบของข้อมูลในฐานข้อมูล
5. ต้องให้ความสะดวกกับผู้ใช้ในการเรียกใช้งานฐานข้อมูล เช่น มีภาษาในการสอบถามข้อมูล (query language) รวมอยู่ด้วย
6. ต้องมีระบบรักษาความถูกต้องของข้อมูล โดยการสำรองข้อมูล รวมทั้งป้องกันผู้ใช้จากการทำงานผิดพลาดต่างๆ

7. ต้องมีระบบรักษาความลับของข้อมูลในฐานข้อมูลนั้น เช่น มีคุณสมบัติการตรวจสอบรหัสผ่าน และรหัสพิเศษในการเข้าไปใช้งาน

2.5 ประโยชน์จากการประมวลผลด้วยฐานข้อมูล

การประมวลผลด้วยระบบฐานข้อมูล มีประโยชน์ดังนี้ (ดวงแก้ว สวามิภักดิ์, 2540)

1. ลดความซ้ำซ้อนของข้อมูล ในกรณีที่มีข้อมูลชนิดเดียวกันถูกเก็บไว้หลาย ๆ แห่ง หรือที่เรียกว่าความซ้ำซ้อน การนำข้อมูลทั้งหมดมาเก็บไว้ที่เดียวกันในฐานข้อมูล จึงถือว่าเป็นการ “ลด” ความซ้ำซ้อนลงไปได้ ทั้งนี้ มิใช่หมายความว่าให้ขจัดข้อมูลออกไปเพื่อให้เหลือน้อยลง
2. สามารถหลีกเลี่ยงความขัดแย้งของข้อมูลได้ในระดับหนึ่ง เนื่องจากบางครั้งจะต้องมีการ แก้ไขข้อมูล จึงอาจจะก่อให้เกิดปัญหาในการแก้ไขคือ เมื่อเราแก้ไขข้อมูลที่เหมือนกัน แต่แก้ไขไม่หมดหรือแก้ไขไม่ครบทุกข้อมูลที่มีอยู่ในแต่ละแห่ง จึงทำให้ข้อมูลชุดเดียวกันอาจมีค่าในแต่ละแห่งไม่ตรงกัน ดังนั้นถ้าการใช้ระบบฐานข้อมูลทำให้เราสามารถลดความซ้ำซ้อนลงไปได้ ซึ่งถ้าใช้ระบบฐานข้อมูลเมื่อเกิดการแก้ไขข้อมูลขึ้นเมื่อใดก็จะต้องแก้ไขเหมือนกันครบทุกแห่ง
3. สามารถใช้ข้อมูลร่วมกันได้ การใช้ข้อมูลร่วมกันได้นี้ ไม่ได้จำกัดอยู่เฉพาะโปรแกรมที่ใช้ข้อมูลในปัจจุบันเท่านั้น แต่โปรแกรมประยุกต์ที่มีการพัฒนาขึ้นมาใหม่ก็สามารถที่จะใช้ข้อมูลที่มีอยู่ได้โดยไม่ต้องเพิ่มเติมข้อมูลเข้าไปในระบบอีก
4. สามารถควบคุมความเป็นมาตรฐานได้ เมื่อมีการนำข้อมูลมาเก็บรวบรวมกันไว้ในฐานข้อมูลเช่นนี้ ทำให้ผู้ที่มีหน้าที่ควบคุม ดูแลการใช้ระบบฐานข้อมูลสามารถกำหนดมาตรฐานของข้อมูลขึ้นมาได้ ทำให้การบริหารหรือ แลกเปลี่ยนข้อมูลระหว่างระบบเป็นไปอย่างสะดวกและถูกต้อง
5. สามารถจัดหาระบบความปลอดภัยที่รัดกุมได้ ระบบฐานข้อมูลสามารถกำหนดสิทธิการใช้งานให้แก่ผู้ใช้คนใดๆ ก็ได้ตามความเหมาะสม และผู้ใช้แต่ละคนก็อาจจะใช้ข้อมูลได้ในระดับที่ต่างกันหรือบุคคลิกนัยหนึ่งก็คือ ผู้ใช้แต่ละคนจะมองฐานข้อมูลด้วยวิสัยที่ต้องการ จึงทำให้มีความปลอดภัยในการใช้ข้อมูลร่วมกัน
6. สามารถควบคุมความคงสภาพของข้อมูลได้ โดยมีการกำหนดค่าต่างๆ หรือ จำกัดช่วงของข้อมูลไว้ เพื่อป้องกันการพิมพ์ข้อมูล ผิดพลาด เช่น กำหนดช่วงของข้อมูลในการ

กรอกหมายเลขโทรศัพท์ไว้ 7 ตัว เมื่อพิมพ์ครบ 7 ตัวแล้ว ก็กำหนดให้เลื่อนไปข้อมูลถัดไป ฉะนั้น ถ้าพิมพ์ไม่ครบโปรแกรมก็จะไม่เลื่อนให้หรือพิมพ์เกิน 7 ตัวก็ไม่ได้

7. สามารถสร้างสมดุลในความขัดแย้งของความต้องการได้ การที่ผู้ใช้ทั้งหมดขององค์กรใช้ข้อมูลจากฐานข้อมูลร่วมกัน ทำให้ทราบถึงความต้องการและความสำคัญของผู้ใช้งานทั้งหมด จึงสามารถกำหนดโครงสร้างของฐานข้อมูลเพื่อให้บริการที่ดีที่สุดได้ เช่น เลือกเก็บข้อมูลที่ต้องใช้บ่อยๆ ไว้ในสื่อข้อมูลที่มีความเร็วเป็นพิเศษ เป็นต้น เป็นการสร้างสมดุลของความ ต้องการไม่ให้เกิดความขัดแย้งในหมู่ผู้ใช้เพราะการออกแบบนั้นกระทำการบนแนวทางที่มุ่งจะให้ประโยชน์ส่วนรวมดีที่สุดแล้ว

8. เกิดความเป็นอิสระของข้อมูล โปรแกรมประยุกต์ที่เขียนขึ้นจะไม่ขึ้นกับโครงสร้างของตารางที่มีการเปลี่ยนแปลงไป เนื่องจากโครงสร้างของตารางต่าง ๆ และตัวข้อมูลในแต่ละตารางจะถูกเก็บอยู่ในฐานข้อมูลทั้งหมด โปรแกรมประยุกต์ไม่จำเป็นต้องเก็บโครงสร้างของตารางที่จะใช้ไว้ ซึ่งต่างกับระบบการประมวลผลเพิ่มข้อมูล ดังนั้นการเปลี่ยนแปลงโครงสร้างของตาราง เช่น การเปลี่ยนแปลงขนาดของเขต ข้อมูลในตารางใดภายในฐานข้อมูลก็ไม่จำเป็นต้องไปทำการแก้ไขโปรแกรมประยุกต์ที่มีการเรียกใช้เขตข้อมูลนั้น