บทที่ 2
หลักการและทฤษฎี

ปัจจุบันการนำคอมพิวเตอร์มาใช้ในการจัดการเกี่ยวกับฐานข้อมูล (Database) ได้รับความนิยมมาก โดยเฉพาะอย่างยิ่งในองค์กรที่มีขนาดใหญ่ ทั้งนี้เนื่องจากการจัดการสามารถทำได้รวดเร็วและถูกต้องแม่นยำ ทำให้มีประสิทธิภาพโดยรวมในการดำเนินการขององค์กรสูงขึ้น ด้วยระบบฐานข้อมูล (Database System) กล่าวคือ การจัดเก็บข้อมูลอย่างเป็นระบบ ซึ่งผู้ใช้สามารถเรียกใช้ข้อมูลดังกล่าว ได้ในลักษณะต่างๆ เช่น การเพิ่มข้อมูล (Add Data) การแทรกข้อมูล (Insert Data) การเรียกใช้ข้อมูล (Retrieve Data) การแก้ไขและลบข้อมูล (Update & Delete Data) ตลอดจนการเคลื่อนย้ายข้อมูล (Move Data) ไปตามกำหนด

2.1 ระบบแฟ้มข้อมูล

จากบทบาทของคอมพิวเตอร์ ที่เข้ามามีอิทธิพลต่อการดำเนินงานภายในองค์กร ได้ส่งผลให้การจัดเก็บข้อมูล ในแฟ้มข้อมูล มีการใช้งานกันอย่างแพร่หลายมากยิ่งขึ้น จากเดิมที่มีเพียง 2 หรือ 3 ได้เพิ่มจำนวนขึ้นเป็น 10 ถึง 20 แฟ้มข้อมูล ดังนั้น จึงต้องมีการเข้ามาควบคุมโครงสร้าง และการใช้งานแฟ้มข้อมูลต่างๆ ให้มีความเหมาะสมต่อการใช้งานมากขึ้น และรวบรวมแฟ้มข้อมูลเหล่านี้ เข้าเป็นระบบที่เรียกว่า "ระบบแฟ้มข้อมูล (File System)"

การใช้งานระบบแฟ้มข้อมูล จะต้องอาศัยโปรแกรมเมอร์พัฒนาโปรแกรมเพื่ออ่านข้อมูลจากแฟ้มข้อมูลต่างๆ ขึ้นมาประมวลผล เพื่อให้ได้ผลลัพธ์ตามที่ผู้ใช้ต้องการ ภาษาคอมพิวเตอร์ที่ใช้ในการพัฒนาโปรแกรม โดยทั่วไปจะ ได้แก่ ภาษาคอมพิวเตอร์ในยุคที่ 3 (Third-generation Language) เช่น ภาษา COBOL, FORTRAN, BASIC ฯลฯ เป็นต้น แต่ภาษาคอมพิวเตอร์เหล่านี้มีข้อจำกัดในการเรียกใช้ข้อมูลจากแฟ้มข้อมูล เนื่องจากภาษาคอมพิวเตอร์เหล่านี้จะอ้างถึงข้อมูลในแฟ้มข้อมูลตามโครงสร้างทางกายภาพของข้อมูลในแฟ้มข้อมูลนั้น

ด้วยเหตุนี้จึงส่งผลให้การพัฒนาแต่ละโปรแกรมขึ้นใช้งานกับแฟ้มข้อมูลต่างๆ มีความซับซ้อน และต้องใช้เวลาค่อนข้างมาก รวมทั้งทำให้แต่ละโปรแกรมถูกผูกติดอยู่กับแฟ้มข้อมูลต่างๆ ดังนั้นเมื่อต้องมีการเปลี่ยนแปลงโครงสร้างของแฟ้มข้อมูลใด แฟ้มข้อมูลหนึ่ง จึงต้องแก้ไขโปรแกรมต่างๆที่มีการเรียกใช้ข้อมูลจากแฟ้มข้อมูลนั้นตามไปด้วย ส่งผลให้ค่าใช้จ่ายในการบำรุงรักษาค่อนข้างสูง โดยเฉพาะอย่างยิ่งในกรณีที่ต้องมีการแก้ไขโปรแกรม จำนวนมาก

2.2 ปัญหาที่เกิดจากระบบแฟ้มข้อมูล

ด้วยการจัดการแฟ้มข้อมูลอย่างเป็นเอกเทศ และกระจัดกระจายของระบบแฟ้มข้อมูล เมื่อพิจารณาดูแล้วจะพบว่าระบบแฟ้มข้อมูลต่างๆ มีความปลอดภัย และความคล่องตัวสูง เนื่องจากมีขนาดเล็ก และแยกเก็บภายในแต่ละหน่วยงาน แต่ในขณะเดียวกัน ได้ก่อให้เกิดปัญหาต่างๆ ขึ้นเช่นเดียวกัน ดังจะกล่าวต่อไปนี้

2.2.1 Data Redundancy เป็นปัญหาที่เกิดขึ้นจาก มีการจัดเก็บข้อมูลที่ซ้ำซ้อนกัน กล่าวคือ ข้อมูลชุดเดียวกันถูกจัดเก็บอยู่ใน 2 แฟ้มข้อมูลหรือมากกว่า ซึ่งการซ้ำซ้อนของข้อมูลในลักษณะนี้ จะส่งผลให้เสียพื้นที่ในการจัดเก็บไป

2.2.2 Data Inconsistency เป็นปัญหาที่เป็นผลกระทบมาจากการจัดเก็บข้อมูลที่มีความซ้ำซ้อน เนื่องจากมีข้อมูลชุดเดียวกันจัดเก็บอยู่หลายแฟ้มข้อมูล อาจทำให้เกิดข้อมูลชุดเดียวกันที่มีค่าต่างกันในต่างแฟ้มข้อมูล ได้ส่งผลทำให้ไม่ทราบว่า ข้อมูลชุดใดเป็นชุดที่ถูกต้อง

2.2.3 Data Anomaly เป็นปัญหาที่เป็นผลกระทบมาจากการจัดเก็บข้อมูลที่มีความซ้ำซ้อนอีก ลักษณะหนึ่ง เนื่องจากการที่มีข้อมูลชุดเดียวกัน ถูกจัดเก็บอยู่หลายแฟ้มข้อมูล อาจส่งผลให้ความสัมพันธ์ระหว่างข้อมูลในแฟ้มต่างๆ สูญเสียไป ในกรณีที่มีการเพิ่ม ลบ หรือเปลี่ยนแปลงค่าของข้อมูลชุดเดียวกันในแฟ้มข้อมูลต่างๆ ที่สัมพันธ์กันไม่ครบถ้วน ซึ่งการสูญเสียความสัมพันธ์ระหว่างแฟ้มข้อมูลนี้ เกิดขึ้นได้ 3 ลักษณะ ดังนี้

2.2.3.1 Modification Anomaly เป็นการเปลี่ยนแปลงค่าของข้อมูลในแฟ้มข้อมูลต่างๆ ที่สัมพันธ์กันไม่ครบถ้วน

2.2.3.2 Insertion Anomaly เป็นการกำหนดข้อมูลเพิ่มเติมให้กับแฟ้มข้อมูลต่างๆ ที่สัมพันธ์กันไม่ครบถ้วน

2.2.3.3 Deletion Anomy เป็นการลบข้อมูลจากแฟ้มข้อมูลต่างๆ ที่สัมพันธ์กันไม่ครบถ้วน

2.3 ระบบฐานข้อมูล

ฐานข้อมูล (Database) คือ การนำเอาข้อมูลต่างๆที่มีความสัมพันธ์กันมาเก็บรวบรวมกันไว้ภายในฐานข้อมูลเดียว และสามารถใช้ข้อมูลร่วมกันได้

ระบบฐานข้อมูล (Database System) คือ การจัดเก็บข้อมูลอย่างเป็นระบบซึ่งผู้ใช้สามารถเรียกใช้ข้อมูลดังกล่าวได้ในลักษณะต่างๆเช่น การเพิ่มข้อมูล (Add Data) การเรียกใช้ข้อมูล (Retrieve Data) การแก้ไขและลบข้อมูล (Update & Delete Data) ตลอดจนการเคลื่อนย้ายข้อมูล (Move Data) ไปตามกำหนด

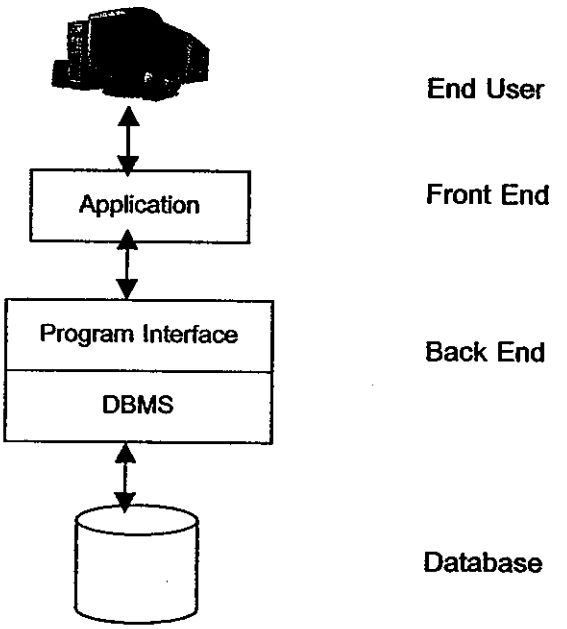
หรือระบบฐานข้อมูล อาจหมายถึง การจัดทำฐานข้อมูลขึ้นเพื่อสนับสนุนการดำเนินงานหรือกิจกรรม อย่างใดอย่างหนึ่ง

2.4 โครงสร้างของระบบฐานข้อมูล (Structure of database)

ระบบฐานข้อมูลสามารถแบ่งออก ตามลักษณะโครงสร้าง ซึ่งประกอบไปด้วยโครงสร้างหลัก 2 ส่วน ได้แก่ ส่วน Front End และ Back End

2.4.1 Front End เป็นโปรแกรมประยุกต์ (Application) ที่อาจจะสร้างจากภาษาต่างๆ ส่วนนี้โดยปกติ จะรองรับการทำงานของผู้ใช้ (End User) เพื่อทำหน้าที่ติดต่อกับระบบ

2.4.2 Back End เป็นส่วนที่ทำหน้าในการจัดการกับระบบฐานข้อมูลทั้งหมด ในแง่ของการจัดเก็บ และเรียกใช้ข้อมูลจากแหล่งข้อมูลจริง ได้แก่ การปฏิบัติการต่างๆกับข้อมูล, การจัดทำ Backup, การควบคุมความถูกต้องในการใช้ข้อมูลพร้อมกัน รวมไปถึงการควบคุมความปลอดภัยของระบบ เป็นต้น



รูปที่ 2.1 โครงสร้างของระบบฐานข้อมูล

2.5 องค์ประกอบของระบบฐานข้อมูล

องค์ประกอบของระบบฐานข้อมูลมีดังต่อไปนี้

2.5.1 ข้อมูล(Data) เนื่องจากฐานข้อมูลเป็นการจัดเก็บรวบรวมข้อมูลให้มีลักษณะเป็นศูนย์กลางข้อมูลอย่างเป็นระบบ ในกรณีที่มีผู้ใช้ร่วมกันหลายคน(Multi-User) ข้อมูลจะต้อง

สามารถเรียกใช้ร่วมกันได้ ซึ่งในทางปฏิบัติผู้ใช้จะมองภาพของข้อมูล ที่แตกต่างกันไปตามระดับของการออกแบบระบบ

2.5.2 ฮาร์ดแวร์ (Hardware) ในส่วนของ Hardware ที่เกี่ยวข้องกับระบบ จะพิจารณาถึงส่วนประกอบที่สำคัญสองประการ ส่วนแรกคือ สื่อในการเก็บข้อมูล (Secondary Storage) ได้แก่ การเก็บข้อมูลด้วย Magnetic Disk รวมไปถึงการติดต่อระหว่างอุปกรณ์ที่เกี่ยวข้อง เช่น I/O Device ต่างๆส่วนที่สองจะเกี่ยวข้องกับความเร็วในการทำงานของ โปรเซสเซอร์และเมโมรี ซึ่งจะขึ้นอยู่กับขนาดของข้อมูลในระบบและจำนวนของข้อมูลในระบบและจำนวนของผู้ใช้เป็นตัวกำหนด

โดยอุปกรณ์ทางคอมพิวเตอร์ที่มีส่วนเกี่ยวข้องกับระบบฐานข้อมูล จะประกอบด้วย 2 ส่วนหลักๆ ดังต่อไปนี้

2.5.2.1 หน่วยความจำสำรอง (Secondary Storage) เนื่องจากเป็นอุปกรณ์ทางคอมพิวเตอร์ที่ใช้จัดเก็บข้อมูลของฐานข้อมูล ดังนั้นสิ่งที่ต้องคำนึงถึงสำหรับอุปกรณ์ในส่วนนี้จึงได้แก่ ความจุของหน่วยความจำสำรองที่นำมาใช้ในการจัดเก็บข้อมูลของฐานข้อมูลนั้น

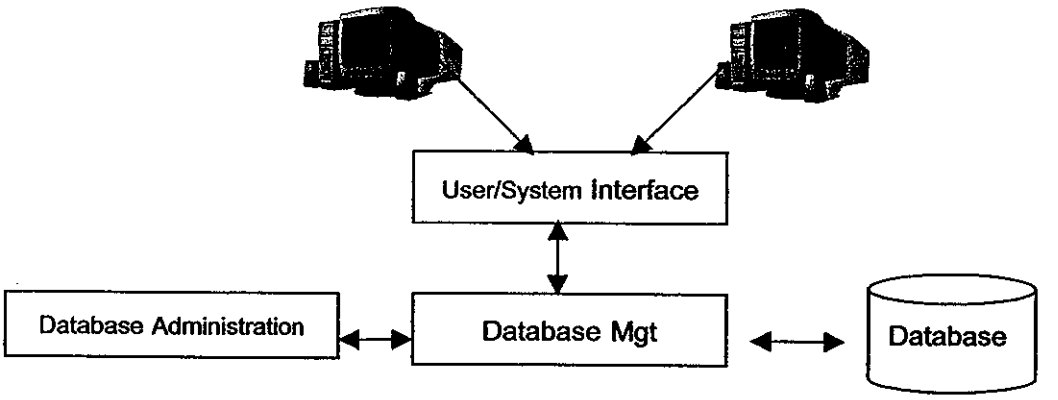
2.5.2.2 หน่วยประมวลผล และหน่วยความจำหลัก เนื่องจากเป็นอุปกรณ์ที่จะต้องทำงานร่วมกัน เพื่อนำข้อมูลจากฐานข้อมูลขึ้นมาประมวลผลตามคำสั่งที่กำหนด ดังนั้นสิ่งที่ต้องคำนึงถึงสำหรับอุปกรณ์ในส่วนนี้ จึงได้แก่ ความเร็วของหน่วยประมวลผล และขนาดของหน่วยความจำหลักของเครื่องคอมพิวเตอร์ที่นำมาใช้ประมวลผลร่วมกับฐานข้อมูลนั้น

2.5.3 ผู้ใช้งานระบบฐานข้อมูล (User) ในระบบฐานข้อมูลจะมีผู้ที่เรียกใช้ข้อมูลจากระบบฐานข้อมูลมาใช้งาน สามารถแบ่งออกเป็น 3 กลุ่ม ได้ดังนี้

2.5.3.1 Application Programmer เป็นบุคลากรที่ทำหน้าที่เขียนโปรแกรมประยุกต์ใช้งาน พัฒนาโปรแกรม (Application Program) เพื่อจัดเก็บและเรียกใช้งานข้อมูลจากระบบฐานข้อมูลมาประมวลผล โดยโปรแกรมที่พัฒนาขึ้นส่วนใหญ่ มักจะใช้ร่วมกับคำสั่งในกลุ่ม Data Manipulation Language (DML) ของ Query Language เพื่อเรียกใช้ข้อมูลจากฐานข้อมูล

2.5.3.2 End User เป็นบุคลากรที่ทำการใช้ข้อมูลจากระบบ ซึ่งโดยปกติจะทำงานใน 3 ลักษณะ คือ การอ่านค่า (Read Only), การเพิ่มหรือการลบข้อมูล (Add/Delete) และการแก้ไขข้อมูล (Modify Data) เป็นต้น

2.5.3.3 DBA (Database Administrator) เป็นบุคลากรที่ทำหน้าที่เป็นผู้ควบคุม และบริหารงานของระบบฐานข้อมูลทั้งหมด นั้นคือจะเป็นผู้ที่ต้องตัดสินใจว่าข้อมูลใด ที่จะรวบรวมเข้าสู่ระบบรวมไปถึงเป็นผู้กำหนดกฎเกณฑ์ที่ใช้ภายในระบบ เช่น วิธีการในการจัดเก็บข้อมูล การเรียกใช้ข้อมูลตลอดจนการกำหนดการรักษาความปลอดภัยในระบบ เป็นต้น



รูปที่ 2.2 การติดต่อกับระบบฐานข้อมูลของบุคคลากร

2.5.4 ซอฟต์แวร์ (Software) ทำหน้าที่เป็นสื่อกลางระหว่าง ผู้ใช้และข้อมูลที่ถูกจัดเก็บในสื่อต่างๆ Software ในส่วนนี้ เรียกว่า Database Management System (DBMS) นั่นคือ ความต้องการใช้ข้อมูลจากผู้ใช้จะถูกจัดการโดย DBMS เพื่อที่จะทำงานในลักษณะต่างๆ ไม่ว่าจะเป็นการเรียกข้อมูลการจัดทำรายงานและการปรับเปลี่ยนหรือแก้ไขในรูปแบบต่างๆ

2.6 Data Independence

ในการเรียกใช้ข้อมูลที่จัดเก็บอยู่ในระบบแฟ้มข้อมูล จะต้องอาศัยโปรแกรมที่เขียนขึ้น เพื่อเรียกใช้ข้อมูล ในแฟ้มข้อมูลนั้นโดยเฉพาะ ผู้ใช้จะต้องให้ โปรแกรมเมอร์จัดทำโปรแกรม เพื่ออ่านข้อมูลจากแฟ้มข้อมูล และพิมพ์รายงานที่แสดงเฉพาะข้อมูลที่ตรงตามเงื่อนไขที่กำหนด ดังนั้น เมื่อมีการเปลี่ยนแปลงโครงสร้างทางกายภาพของแฟ้มข้อมูลใดแฟ้มข้อมูลหนึ่ง จึงส่งผลให้โปรแกรมต่างๆ ที่เรียกใช้ข้อมูลในแฟ้มข้อมูลนั้น ต้องมีการเปลี่ยนแปลงตามไปด้วย

ภายในระบบฐานข้อมูล ไม่สามารถที่จะยอมให้ความเป็นอิสระ ระหว่างข้อมูล และโปรแกรมเกิดขึ้นได้ เนื่องจาก 2 สาเหตุหลักๆ ดังนี้

1. เนื่องจากในฐานข้อมูล จะต้องไม่ปรากฏข้อมูลที่ซ้ำซ้อนกันเกิดขึ้น แต่ในแง่ความเป็นจริงแล้ว ข้อมูลที่ผู้ใช้ต้องการถึงแม้จะเป็นข้อมูลเดียวกัน ก็อาจต้องการรูปแบบของข้อมูลที่ต่างกันได้

2. เนื่องจาก DBA มีสิทธ์ที่จะเปลี่ยนแปลงโครงสร้างของข้อมูลภายในฐานข้อมูลได้ ดังนั้นจึงจำเป็นที่จะต้องทำให้ข้อมูลภายในฐานข้อมูล เป็นอิสระจากโปรแกรมต่างๆ ที่เรียกใช้ เพื่อที่จะทำให้ DBA สามารถเปลี่ยนแปลงโครงสร้างของข้อมูลภายในฐานข้อมูลได้ โดยไม่ส่งผลกระทบกับโปรแกรมต่างๆ

ในการกำหนดให้ข้อมูลเป็นอิสระจากโปรแกรมที่เรียกใช้ จะแบงออกเป็น 2 ระดับ ดังนี้

2.6.1 ระดับ Physical

เป็นระดับที่โครงสร้างทางกายภาพของข้อมูลเป็นอิสระจากโปรแกรมที่ใช้ เช่น สามารถเปลี่ยนแปลง โครงสร้างของ Index File ได้โดยไม่ต้องแก้ไขโปรแกรมที่เรียกใช้ข้อมูลนั้น

2.6.2 ระดับ Logical

เป็นระดับที่ความสัมพันธ์ระหว่างข้อมูลในส่วนต่างๆ ภายในฐานข้อมูลเป็นอิสระจากโปรแกรมที่เรียกใช้ เช่น สามารถแยกบาง Field ออกไปเป็นแฟ้มข้อมูลใหม่ ได้โดยไม่ต้องแก้ไขโปรแกรมที่เรียกใช้ข้อมูลนั้น

เมื่อพิจารณาความเป็นอิสระ ของข้อมูลที่มีต่อ โปรแกรมที่เรียกใช้ในทั้ง 2 ระดับจะสังเกตเห็นว่า การกำหนดให้เป็นอิสระจากโปรแกรมที่เรียกใช้ในระดับ Logical จะกระทำได้ยากกว่าในระดับ Physical เนื่องจากความสัมพันธ์ระหว่างข้อมูลในส่วนต่างๆ ภายในแฟ้มข้อมูลเดียวกัน จะมีความเกี่ยวพันกับโปรแกรมมากกว่าโครงสร้างทางกายภาพของข้อมูล

2.7 Database Management System (DBMS)

เป็นโปรแกรมที่ทำหน้าที่เป็นตัวกลางในการติดต่อระหว่างผู้ใช้กับฐานข้อมูล เพื่อการจัดการและควบคุมความถูกต้อง ความซ้ำซ้อน และความสัมพันธ์ระหว่างข้อมูลต่างๆภายในฐานข้อมูล ซึ่งต่างจาระบบแฟ้มข้อมูลที่หน้าที่เหล่านี้จะเป็นหน้าที่ของโปรแกรมเมอร์ ในการติดต่อกับข้อมูลในฐานข้อมูลไม่ว่าจะด้วยการใช้คำสั่งในกลุ่มคำสั่ง DML หรือ DDL หรือจะด้วยโปรแกรมต่างๆ ทุกคำสั่งที่ใช้กระทำกับข้อมูล จะถูกโปรแกรม Database Management System นำมาแปล (Compile) เป็นการกระทำ (Operation) ต่างๆ ภายใต้คำสั่งนั้นๆ เพื่อนำไปกระทำกับตัวข้อมูลภายในฐานข้อมูลต่อไปสำหรับส่วนการทำงานต่างๆภายในโปรแกรม Database Management System ที่ทำหน้าที่ในการแปลคำสั่งไปเป็นการกระทำต่างๆ ที่จะกระทำกับข้อมูลนั้น ประกอบด้วยส่วนการทำงานต่างๆ ดังนี้

2.7.1 Database Manager เป็นส่วนที่ทำหน้าที่กำหนดการกระทำต่างๆ ให้กับส่วน File Manager เพื่อไปกระทำกับข้อมูลที่เก็บอยู่ในฐานข้อมูล (File Manager เป็นส่วนที่ทำหน้าที่บริหาร หละจัดการกับข้อมูลที่เก็บอยู่ในฐานข้อมูลในระดับกายภาพ)

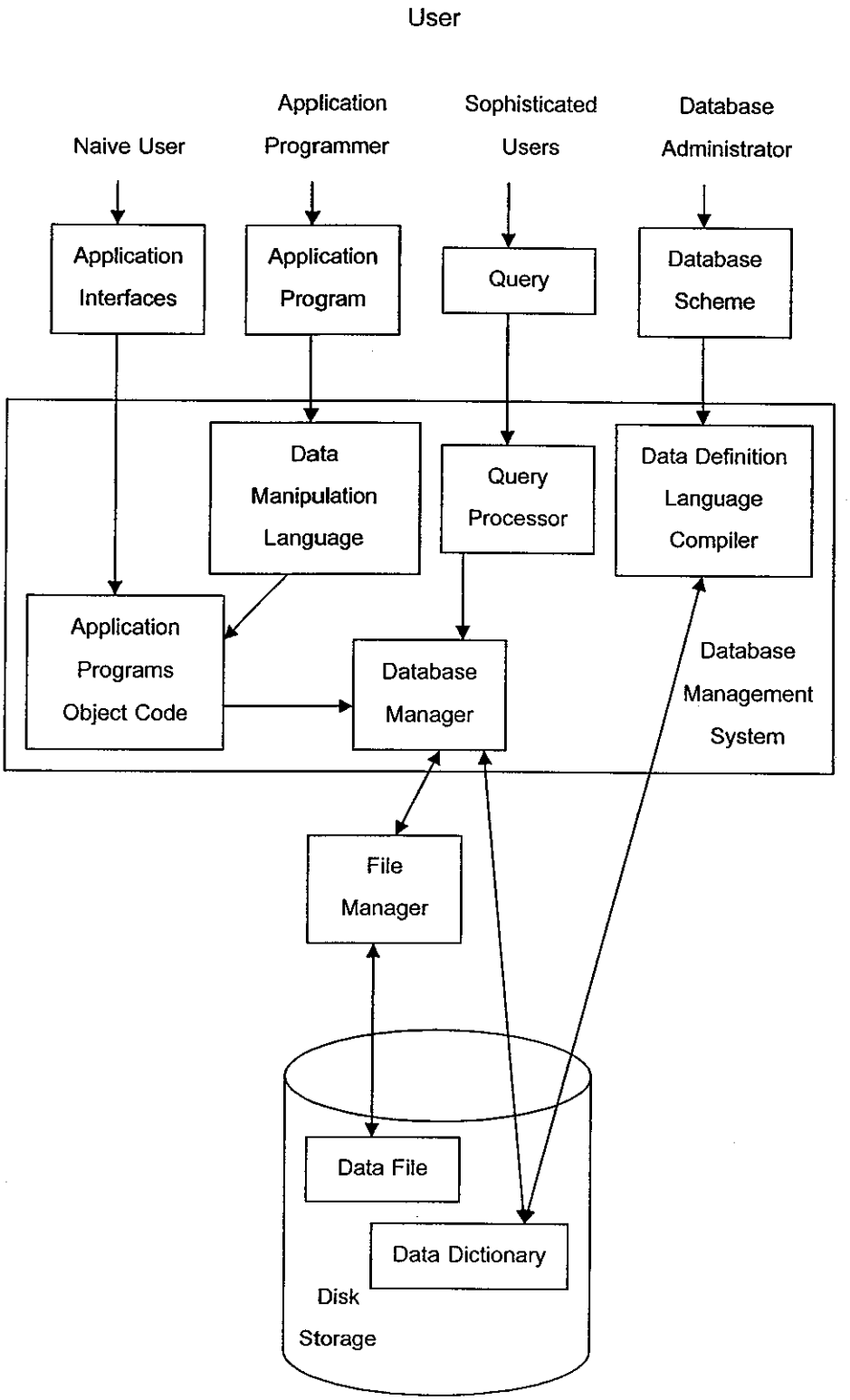
2.7.2 Query Processor เป็นส่วนที่ทำน้าที่แปลงประโยคคำสั่งของ Query Language ให้อยู่ในรูปแบบของคำสั่งที่ Database Manager เข้าใจ

2.7.3 Data Manipulation Language Precompiler เป็นส่วนที่ทำหน้าที่แปล (Compile) ประโยคคำสั่งของกลุ่มคำสั่ง DML ให้อยู่ในรูปที่ส่วน Application Programs Object Code จะนำไปเข้ารหัสเพื่อส่งต่อไปยังส่วน Database manager ในการแปลประโยคคำสั่งของกลุ่มคำสั่ง DML ของส่วน Data Manipulation Language Precompiler นี้จะต้องทำงานร่วมกับส่วน Query Processor

2.7.4 Data Definition Language Precompiler เป็นส่วนที่ทำหน้าที่แปล (Compile) ประโยคคำสั่งของกลุ่มคำสั่ง DDL ให้อยู่ในรูปแบบของ Metadata ที่เก็บอยู่ในส่วน Data Dictionary ของฐานข้อมูล (Metadata ได้แก่ รายละเอียดที่บอกถึงโครงสร้างต่างๆของข้อมูล)

2.7.5 Application Programs Object Code เป็นส่วนที่ทำหน้าที่แปลงคำสั่งต่างๆ ของโปรแกรม รวมทั้งคำสั่งในกลุ่มคำสั่ง DML ที่ส่งต่อมาจากส่วน Data Manipulation Language Precompiler ให้อยู่ในรูปของ Object Code ที่จะส่งต่อไปให้ Database Manager เพื่อกระทำกับข้อมูลในฐานข้อมูล

ทั้ง 5 ส่วนของโปรแกรม DBMS สามารถแสดงด้วยแผนภาพ ได้ดังรูป



รูปที่ 2.3 แสดงส่วนของโปรแกรม DBMS

โปรแกรม DBMS นี้ได้ถูกพัฒนาขึ้นมาเพื่อแก้ปัญหาทางด้าน Database Independence ที่ไม่มีในระบบแฟ้มข้อมูล ดังนั้นจึงมีความเป็นอิสระจากทั้งตัว Hardware และตัวข้อมูลภายในฐานข้อมูล กล่าวคือ โปรแกรม DBMS จะมีการทำงานที่ไม่ขึ้นกับรูปแบบ (Platform) ของตัว Hardware ที่นำมาใช้กับระบบฐานข้อมูล รวมทั้งมีรูปแบบในการอ้างถึงข้อมูลที่ไม่ขึ้นอยู่กับโครงสร้างทางกายภาพของข้อมูล ด้วยการใช้ Query Language ในการติดต่อกับข้อมูลในฐานข้อมูลแทนคำสั่งของภาษาคอมพิวเตอร์ในยุคที่ 3 ส่งผลให้ผู้ใช้สมารถเรียกใช้ข้อมูลจากฐานข้อมูล โดยไม่จำเป็นต้องทราบถึงประเภทของข้อมูล หรือขนาดของข้อมูลนั้น หรือสามารถกำหนดลำดับที่ของ Field ในการแสดงผลได้โดยไม่ต้องคำนึงถึงลำดับที่แท้จริงของ Field นั้น

2.8 หน้าที่ของ Database Management System (DBMS)

สำหรับหน้าที่ของโปรแกรม Database Management System (DBMS) มีดังนี้

2.8.1 ทำหน้าที่แปลงคำสั่งที่ใช้จัดการกับข้อมูลภายในฐานข้อมูล ให้อยู่ในรูปที่ฐานข้อมูลเข้าใจ

2.8.2 ทำหน้าที่ในการนำสั่งต่างๆ ซึ่งได้รับการแปลแล้ว ไปสั่งให้ฐานข้อมูลทำงาน เช่น การเรียกใช้ข้อมูล (Retrieve Data) การจัดเก็บข้อมูล (Update Data) การลบข้อมูล (Delete Data) การเพิ่มข้อมูล (Add Data) เป็นต้น

2.8.3 ทำหน้าที่ป้องกันความเสียหายที่เกิดขึ้นกับข้อมูลภายในฐานข้อมูล โดยจะคอยตรวจสอบว่าคำสั่งใดที่สามารถทำงานได้ และคำสั่งใดที่ไม่สามารถทำงานได้

2.8.4 ทำหน้าที่รักษาความสัมพันธ์ของข้อมูลภายในฐานข้อมูลให้มีความถูกต้องอยู่เสมอ

2.8.5 ทำหน้าที่เก็บรายละเอียดต่างๆที่เกี่ยวข้องกับข้อมูลภายในฐานข้อมูลไว้ใน Data Dictionary ซึ่งรายละเอียดเหล่านี้จึงมักจะถูกเรียกว่า "ข้อมูลของข้อมูล (Metadata)"

2.8.6 ทำหน้าที่ควบคุมให้ฐานข้อมูล ทำงานได้อย่างถูกต้องและมีประสิทธิภาพ

2.9 Data Dictionary and Manager

ทุกฐานข้อมูลจะต้องมีส่วนที่ใช้เก็บข้อมูลในลักษณะ Metadata ซึ่งเป็นข้อมูลที่บอกถึงรายละเอียดของตัวข้อมูลที่เก็บอยู่ในฐานข้อมูล เช่นโครงสร้างฐานข้อมูล โครงสร้างของ Table โครงสร้างของ Index กฎที่ใช้ควบคุมความถูกต้องของข้อมูล (Integrity Rule) กฎที่ใช้ในการรักษาความปลอดภัยให้กับข้อมูล (Security Rule) ฯลฯ ข้อมูลเหล่านี้จัดเป็นข้อมูลที่มีความจำเป็นต่อโปรแกรม Database Management System (DBMS) ในการตัดสินใจที่จะดำเนินการใดๆ ในฐานข้อมูล เช่น ข้อมูลที่เกี่ยวข้องกับกฎการรักษาความปลอดภัยให้กับข้อมูล จะถูกนำมาใช้ในการพิจารณาให้สิทธิกับผู้ใช้ในการใช้งานฐานข้อมูล เป็นต้น สำหรับส่วนที่ใช้จัดเก็บข้อมูลในลักษณะของ Metadata นี้ ได้แก่ Data Dictionary หรือ Catalog

สำหรับ File Manager เป็นส่วนที่ทำหน้าที่บริหารและจัดการกับข้อมูลที่เก็บอยู่ในฐานข้อมูลในระดับกายภาพ (Physical Level)

2.10 ผลกระทบของการประมวลผลด้วยระบบฐานข้อมูล

2.10.1 ข้อดีของการประมวลผลด้วยระบบฐานข้อมูล

2.10.1.1 ลดความซ้ำซ้อนของข้อมูล (Minimal Data Redundancy) การจัดเก็บข้อมูลในลักษณะที่เป็นแฟ้มข้อมูล อาจทำให้ข้อมูลที่เป็นประเภทเดียวกัน ถูกเก็บไว้หลายๆแห่ง ทำให้เกิดความซ้ำซ้อนของข้อมูลขึ้นได้ ดังนั้นการนำข้อมูลมารวมเก็บไว้ในระบบฐานข้อมูลจะช่วยลดปัญหาการซ้ำซ้อนของข้อมูลได้

2.10.1.2 หลีกเลี่ยงความขัดแย้งของข้อมูลได้ (Consistency of Data) การจัดเก็บข้อมูลในลักษณะเป็นแฟ้มข้อมูล โดยที่ข้อมูลเป็นเรื่องเดียวกัน อาจมีอยู่ในหลายแฟ้ม ซึ่งก็ให้เกิดความขัดแย้งของข้อมูลขึ้นได้ ทั้งนี้อาจเนื่องมาจากการแก้ไขข้อมูลที่แฟ้มแห่งหนึ่ง แต่มิได้แก้ไขข้อมูลเรื่องเดียวกันที่อยู่ในไฟล์อื่นทำให้ข้อมูลนั้นๆ แตกต่างกันได้

2.10.1.3 จำกัดความผิดพลาดในการป้อนข้อมูลให้น้อยที่สุด (Data Integrity) บางครั้งความผิดพลาดของข้อมูล อาจเกิดขึ้นจากการป้อนข้อมูลที่ไม่ถูกต้องเข้าสู่ระบบ ดังนั้นในระบบจัดการฐานข้อมูล จึงจำเป็นที่จะต้องกำหนดกฎเกณฑ์ในการรับข้อมูลจากการป้อนของผู้ใช้ เพื่อรักษาความถูกต้องของข้อมูลให้มากที่สุดเท่าที่จะทำได้

2.10.1.4 สามารถใช้ข้อมูลร่วมกันได้ (Sharing of Data) เนื่องจากระบบฐานข้อมูลเป็นการจัดเก็บข้อมูลไว้ในที่เดียวกัน เมื่อผู้ใช้ต้องการเรียกใช้ข้อมูลจากแฟ้มที่แตกต่างกัน ก็จะสามารถทำได้โดยง่าย

2.10.1.5 สามารถกำหนดความเป็นมาตรฐานเดียวกันได้ (Enforcement of Standard) การเก็บข้อมูลไว้ด้วยกันจะสามารถกำหนด และควบคุมความมีมาตรฐานของข้อมูลให้เป็นไปในทิศทางเดียวกันได้ ดังนั้นจึงทำให้ระบบเกิดความเชื่อมั่นมากยิ่งขึ้น

2.10.1.6 สามารถกำหนดระบบความปลอดภัยของข้อมูลได้ (Security and Privacy Control) เนื่องจากระบบจะทำการกำหนดระดับของผู้ใช้แต่ละคน ตามลำดับความสำคัญของผู้ใช้ ดังนั้นจึงสามารถที่จะควบคุมและดูแลความปลอดภัยของข้อมูลภายในระบบได้ดียิ่งขึ้น

2.10.1.7 ข้อมูลมีความเป็นอิสระ (Data Independence) ระบบฐานข้อมูลจะทำหน้าที่เชื่อมโยงกับโปรแกรมประยุกต์ ที่ทำงานกับข้อมูลโดยตรง โดยการแก้ไขข้อมูลนั้นจะแก้ไขในส่วน นั้นเท่านั้น ส่วนโปรแกรมอื่นจะเป็นอิสระต่อการเปลี่ยนแปลง

2.10.1.8 สามารถตอบสนองต่อความต้องการการใช้ข้อมูลในหลายรูปแบบ

2.10.2 ข้อเสียของการประมวลผลด้วยระบบฐานข้อมูล

2.10.2.1 ขั้นตอนการออกแบบและการบำรุงรักษามีต้นทุนสูง เนื่องจากระบบต้องใช้ผู้เชี่ยวชาญเฉพาะในการออกแบบระบบ ไม่ว่าจะเป็นทางด้าน Hardware และ Software รวมไปถึงราคาอุปกรณ์ที่ใช้มีราคาค่อนข้างสูง

2.10.2.2 ระบบมีความซับซ้อนจำเป็นต้องมีผู้ดูแลระบบที่ถูกฝึกมาอย่างดี เพื่อรองรับสถานการณ์ที่ผิดพลาดอันอาจจะเกิดขึ้นได้

2.10.2.3 การเสี่ยงต่อการหยุดชะงักของระบบ เนื่องจากข้อมูลอาจถูกจัดเก็บแบบรวมศูนย์ (Centralized Database System) ความล้มเหลวของการทำงานบางส่วน อาจทำให้ระบบฐานข้อมูลโดยรวมหยุดชะงักการทำงานได้

2.11 นิยามพื้นฐานในระบบฐานข้อมูล

Database มีคุณลักษณะที่สำคัญ 2 ประการ คือ ประการแรกจะต้องเป็นกลุ่มข้อมูลที่รวมเป็นหนึ่งเดียว (Integrated) และข้อมูลนี้จะถูกผู้ใช้สามารถเรียกใช้ร่วมกันได้

2.11.1 Data คือ ความเป็นจริง (Fact) ที่เกี่ยวกับบุคคล สถานที่ เหตุการณ์ หรือสิ่งของต่างๆ ซึ่งสามารถนับจำนวนได้

2.11.2 Information คือ ข้อมูลที่ถูกจัดรวบรวมให้อยู่ในรูปที่สามารถ จะนำไปประกอบการตัดสินใจ อย่างใดอย่างหนึ่งได้

2.11.3 Entity คือ สิ่งใดสิ่งหนึ่ง ได้แก่ ชื่อของบุคคล สถานที่ สิ่งของหรือการกระทำ ที่ต้องการจัดเก็บข้อมูลนั้นไว้ Entity ที่ใช้สำหรับแสดงความสัมพันธ์กัน ระหว่างข้อมูลในระบบ เช่น ลูกค้า ผู้ว่าจ้าง เป็นต้น สัญลักษณ์ที่ใช้แทน Entity คือ รูปสี่เหลี่ยมพื้นผ้า

2.11.4 Attribute คือ รายละเอียดของข้อมูลใน Entity หนึ่งๆ ที่ใช้แสดงลักษณะและคุณสมบัติของ Entity ที่ถูกอ้างถึง เช่น Attribute ของ นิสิต ได้แก่ เลขประจำตัวนิสิต ชั้นปี คณะ ภาควิชา เป็นต้น ค่า Attribute คือค่าที่เก็บอยู่ใน Entity นั่นเองสัญลักษณ์ที่ใช้เขียนแทน คือ รูปวงรี

2.11.5 Entity Set คือ Entity หลายๆตัวที่มีค่า Attribute เหมือนกัน และสามารถนำมารวมกันในรูปของTableเพื่อสะดวกในการเข้าถึงข้อมูลกลุ่มดังกล่าว เช่น Entity Set ของกลุ่มยาที่มีชื่อสามัญทางยาเหมือนกัน เป็นต้น

2.11.6 Field เกิดจากการรวมตัวของข้อมูลที่เล็กที่สุดภายในคอมพิวเตอร์ เรียกว่า บิต (Bit) นำมาประกอบกันจะได้ข้อมูล เรียกว่า ไบต์ (Byte) หรือตัวอักขระ (Character) หากนำอักขระมาประกอบกันเป็นกลุ่ม ก็จะได้ข้อมูลที่ขยายตัวออกเป็นรูปแบบใหม่ที่เรียกว่า Field และอาจกล่าวได้ว่า ส่วนของ Field ก็จะได้แก่ Attribute นั่นเอง

2.11.7 Record เกิดจากการรวมตัวของ Field หรือ Attribute ที่แสดงคุณสมบัติของ Entity ตัวใดตัวหนึ่ง

2.11.8 File คือ กลุ่มของ Record ชนิดเดียวกัน ที่ถูกนำมารวมกันเป็นหมวดหมู่ ข้อมูลที่อยู่ภายในไฟล์จะสามารถมองได้เป็น อาร์เรย์ 2 มิติ นั่นคือ ในรูปของแถวซึ่งแสดงถึงจำนวน Record และคอลัมน์ ซึ่งแทนค่าของ Attribute แต่ละตัวนั่นเอง

2.11.9 Association คือ สัญลักษณ์แสดงความสัมพันธ์กัน ระหว่าง Entity ซึ่งจะเกิดขึ้นได้กับ Entity ตั้งแต่ 2 ตัวขึ้นไป

2.12 การทำนอมัลไลซ์ (Normallization)

การนอมัลไลซ์นับว่าเป็นขั้นตอนสำคัญมากในการออกแบบตารางซึ่งตามทฤฎีการออกแบบฐานข้อมูลจะแบ่งนอมัลไลซ์เป็น 5 ระดับ แต่พบที่ใช้งานส่วนใหญ่จะใช้งานเพียงแค่ 3 ระดับก็เพียงพอแล้ว เพราะจะมีตารางบางรูปแบบเท่านั้นที่ต้องการนอมัลไลซ์ถึง 5 ระดับ

2.12.1 นอมัลไลซ์ระดับที่ 1 (First Nomal Form)

เป็นการทำให้ทุกๆ เซลล์ในตารางเก็บค่าไว้เพียงค่าเดียว ตัวอย่างเช่น

รหัสของยา	ชื่อสามัญทางยา	ชื่อทางการค้า	จำนวน
1	Paracetamol	Sara Tiffy	30 55

ดังนั้นเราจึงต้องทำให้ Repeating Group หมดไปโดยการเติมข้อมูลลงไปในตารางให้สมบูรณ์ดังนี้

รหัสของยา	ชื่อสามัญทางยา	ชื่อทางการค้า	จำนวน
1	Paracetamol	Sara	30
1	Paracetamol	Tiffy	55

เมื่อทำการกำหนดค่าดังกล่าวทำให้ฟิลด์รหัสยาไม่สามารถจะกำหนดให้ฟิลด์หลักได้จึงต้องมีการนำเอาฟิลด์ชื่อทางการค้ามาประกอบด้วย ทำให้ตารางต้องมีการใช้งานคีย์ร่วม

2.12.2 นอมัลไลซ์ระดับที่ 2 (Second Normal Form)

เป็นการพิจารณาว่าฟิลด์อื่นๆ ที่ไม่ได้กำหนดเป็นคีย์หลักนั้น ต้องขึ้นอยู่กับคีย์หลักเพียงอย่างเดียว ดังนั้นถ้าคีย์หลักมีฟิลด์เดียวก็ถือว่าผ่านการนอมัลไลซ์ระดับที่ 2 โดยอัตโนมัติ

2.12.3 นอมัลไลซ์ระดับที่ 3 (Third Normal Form)

เป็นการพิจารณาว่าฟิลด์อื่นๆ นอกเหนือจากจะต้องขึ้นอยู่กับคีย์หลักทั้งหมด (Second Normal Form) แล้วยังจะต้องไม่ขึ้นต่อกันอีกด้วย

2.12.4 นอมัลไลซ์แบบ Boyce-Codd หรือ BCNF

ในบางครั้งการนอมัลไลซ์ระดับที่ 3 อาจพบกับผิดกับความผิดปกติของข้อมูล (ซึ่งไม่พบบ่อยนัก) ซึ่งจะเกิดขึ้นเมื่อมีการแก้ไข หรือเพิ่ม/ลบข้อมูล ซึ่งเกิดจากการไม่ระบุ Candidate Key ให้ครบในฟิลด์ที่ถือว่าเป็น Determinant ซึ่งตามศัพท์ของการออกแบบฐานข้อมูลเราจะได้ยินคำว่า ต้องไม่มีขั้นตอนเชิงฟังก์ชั่น (Functional Dependencies)

2.12.5 นอมัลไลซ์ระดับที่ 4 (Fourt Normal Form)

สำหรับการนอมัลไลซ์ระดับที่ 4 นั้นเกิดขึ้นน้อยมาก ซึ่งการที่จะผ่านการนอมัลไลซ์ระดับที่ 4 นั้น ต้องผ่าน BCNF และต้องไม่มีการขึ้นต่อกันเชิงกลุ่ม

2.12.6 นอมัลไลซ์ระดับที่ 5 (Fifth Normal Form)

นอมัลไลซ์ระดับที่ 5 นั้นเราจะพบได้น้อยมากๆ ซึ่งการที่จะผ่านการนอมัลไลซ์ระดับที่ 5 นั้น จะต้องผ่าน 4NF และต้องไม่มีการขึ้นต่อกันเชิงร่วม

ในทางปฏิบัตินั้นเราจะพบ BCNF, 4NF และ 5NF น้อยมาก ทำให้กานอมัลไลซ์ข้อมูลส่วนใหญ่มักจะทำเพียงแค่ 3NF ก็สารมารถได้ข้อมูลในฐานข้อมูลที่ถูกต้องแล้ว ซึ่งรายละเอียดของการนอมัลไลซ์ตั้งแต่ BCNF ขึ้นไปนั้น เป็นเรื่องที่ต้องศึกษาทฤษฎีการออกแบบฐานข้อมูลโดยละเอียด ทำให้ไม่สามารถนำเสนอส่วนนี้ได้อย่างดี จึงไม่ขออธิบาย

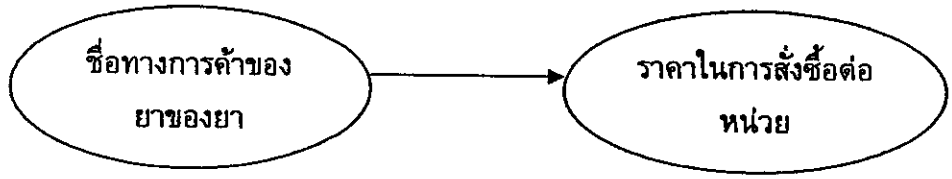
2.13 Type of Entity Association

ค่าของ Entity แต่ละตัวจะถูกเก็บอยู่ในรูปของไฟล์ ในขณะเดียวกัน ค่าของ Attribute ก็จะได้แก่ ค่า Field นั่นเอง ส่วนความสัมพันธ์ ระหว่าง Entity ได้แก่ ความสัมพันธ์ระหว่างไฟล์ที่ถูกนำเสนอในรูปของการกำหนดค่าของ Field ในไฟล์หนึ่ง เพื่อแสดงความสัมพันธ์ไปยังอีกไฟล์หนึ่งนั่นเอง

ความสัมพันธ์ระหว่าง Entity สามารถเขียนแทนได้ด้วย สัญลักษณ์ หัวลูกศร แบ่งชนิดของความสัมพันธ์ออกได้เป็น 3 ลักษณะ ดังต่อไปนี้

2.13.1 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One to One Relationships)

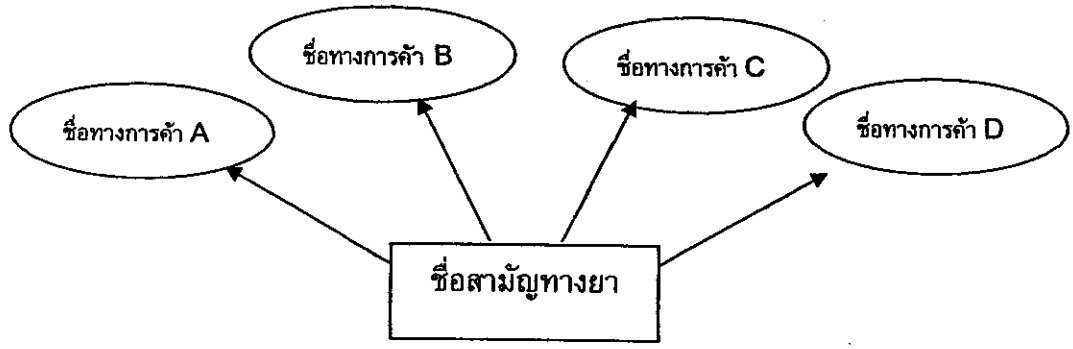
หมายถึง ในช่วงระยะเวลาที่กำหนด ค่าของ Entity A มีความสัมพันธ์กับค่าของ Entity B เพียงค่าเดียวเท่านั้น นั่นคือ หากทราบว่าค่าของ Entity A ก็สามารถหาค่าของ Entity B ได้ด้วย เช่น ในกรณีของชื่อทางการค้าของยาจาก Entity A จะอ้างถึงค่าของ Entity B ได้เพียงค่าเดียวเท่านั้น คือที่ราคาในการสั่งซื้อต่อหน่วยของยานั้น



รูปที่ 2.4 แสดงตัวอย่าง Entity ความสัมพันธ์แบบหนึ่งต่อหนึ่ง

2.13.2 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One to Many Relationships)

หมายถึง ในช่วงระยะเวลาที่กำหนด ค่าของ Entity A จะมีความสัมพันธ์กับค่าของ Entity B ได้มากกว่า 1 เท่า เช่นชื่อสามัญทางยา 1 ตัว อาจมีชื่อทางการค้าหลายตัวได้ เป็นต้น ความสัมพันธ์ในลักษณะนี้จะเกิดขึ้นเป็นส่วนใหญ่ในระบบ



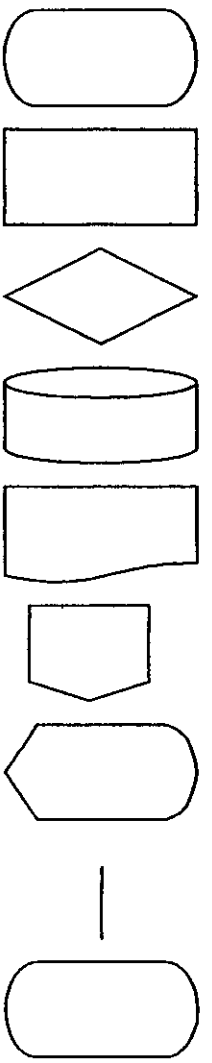
รูปที่ 2.5 แสดงตัวอย่าง Entity ความสัมพันธ์แบบหนึ่งต่อกลุ่ม

2.13.3 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many to Many Relationships)

หมายถึง ในช่วงระยะเวลาที่กำหนด ทั้งค่าของ Entity A มีความสัมพันธ์กับค่าของ Entity B ได้มากกว่า 1 ค่า

2.14 แผนภาพการทำงานของโปรแกรม

Flow Chart หรือแผนภาพจะเป็นเครื่องมือที่โปรแกรมเมอร์ใช้ในการเปลี่ยน Algorithm ความคิดหรือความต้องการของผู้ใช้ ให้อยู่ในรูปของแผนภาพการทำงานของโปรแกรม โดยทั่วไป Flow Chart จะมีลักษณะที่ไม่ขึ้นกับภาษาคอมพิวเตอร์ใดๆ ทำให้เราสามารถใช้ Flow Chart เป็นเสมือนเครื่องมือสื่อสารระหว่างโปรแกรมเมอร์ หรือระหว่างโปรแกรมเมอร์กับผู้ใช้ ว่าแผนงานหรือการประมวลผลของโปรแกรมจะมีลักษณะขั้นตอนตามนี้ นอกจากนั้น Flow Chart ยังเป็นเสมือนแผนภาพโดยรวมของโปรแกรม ที่เราสามารถนำไปแปลงให้เป็นภาษาคอมพิวเตอร์ ก็ได้ การเขียน Flow Chart จะประกอบด้วยสัญลักษณ์หลักๆ ดังต่อไปนี้



ความหมาย

เครื่องหมายเริ่มต้นการทำงาน

Process ของการประมวลผลข้อมูล

Decision การตัดสินใจ

ติดต่อดึงข้อมูลจากฐานข้อมูล

พิมพ์ข้อมูลออกทางเครื่องพิมพ์

จุดต่ออยู่ต่างหน้ากัน

แสดงผลข้อมูลออกทางจอภาพ

เส้นแสดงการเชื่อมต่อทางเดินของการประมวลผล

เครื่องหมายจบการทำงาน

2.15 Data Flow Diagram (DFD)

กรรมวิธีการวิเคราะห์ระบบอย่างมีโครงสร้างนั้น วิธีหนึ่งที่นิยมใช้ในทางปฏิบัติ คือ การมองภาพรวมในรูปแบบการไหลของข้อมูล (Data Flow) โดยที่วิธีนี้จะช่วยให้นักวิเคราะห์สามารถแบ่งระบบเป็นระบบย่อยได้ง่ายขึ้นและสามารถตรวจสอบได้สะดวก

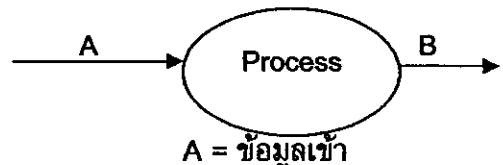
การนำเสนอระบบแบบการไหลของข้อมูลนั้นจะใช้สัญลักษณ์แทนการบรรยายการทำงานของระบบ ซึ่งลักษณะที่จะใช้จะเป็นรูปวงกลม สี่เหลี่ยมจัตุรัส สี่เหลี่ยมผืนผ้าปลายเปิด เส้นโค้ง ลูกศร โดยนำสัญลักษณ์เหล่านี้มาเชื่อมต่อ การแสดงการต่อเนื่องของข้อมูลและการประมวลผล

2.15.1 สัญลักษณ์ Data Flow Diagram (DFD) ในแผนภาพของ DFD จะประกอบด้วยสัญลักษณ์ต่างๆ ดังต่อไปนี้

2.15.1.1 ลูกศร ใช้แทนการไหลของข้อมูลพร้อมกับชื่อของข้อมูลนั้นๆ จะต้องกำกับไว้ด้วย

รูปที่ 2.6 การแทนกระแสข้อมูลเป็นลูกศร

2.15.1.2 รูปวงกลม ใช้แทนกริยาการกระทำต่อข้อมูลที่ไหลเข้ามา โดยไม่คำนึงถึงว่าจะเป็นการกระทำ โดยคนหรือคอมพิวเตอร์ก็ตาม จะได้มาซึ่งผลลัพธ์ที่จะไหลออกจากวงกลม ภายในวงกลมจะระบุคำสั้นๆ ที่จะใช้แทนการกระทำต่อข้อมูล



A = ข้อมูลเข้า

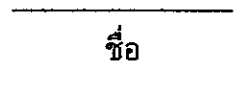
B = ข้อมูลออก

รูปที่ 2.7 การแทนการไหลข้อข้อมูล

2.15.1.3 รูปสี่เหลี่ยม ใช้แทนนามที่อยู่ภายนอกระบบซึ่งเป็นการกำเนิดของข้อมูล หรือการสิ้นสุดของข้อมูล โดยมีชื่ออยู่ภายในสี่เหลี่ยมนั้น

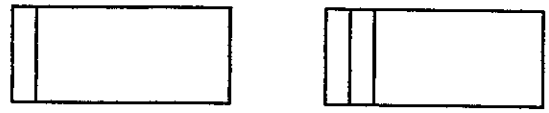
รูปที่ 2.8 การแทนนามที่อยู่นอกระบบ

2.15.1.4 รูปสี่เหลี่ยมผืนผ้าปลายเปิด เป็นตัวแทนของแหล่งเก็บข้อมูลหรือแฟ้มข้อมูลเสมือนเป็นตัวพักหรือช่วงขาดของการไหลของข้อมูลเพื่อนำไปเก็บเท่านั้น การกำหนดชื่อของแหล่งเก็บข้อมูลต้องอยู่ในสี่เหลี่ยม



รูปที่ 2.9 การแทนแหล่งเก็บข้อมูล

2.15.1.5 สัญลักษณ์เพิ่มเติม จะใช้เติมลงในสัญลักษณ์ที่กล่าวมาข้างต้นเพื่อแสดงความเป็นสิ่งเดียวกัน แต่จะถูกกล่าวหลายๆครั้งในแผนภาพ



รูปที่ 2.10 การแทนสัญลักษณ์เพิ่มเติม

2.15.2 ลำดับชั้นใน Data Flow Diagram

ในการเขียน DFD นักวิเคราะห์ระบบจะต้องมองระบบจากภาพรวมก่อนจากนั้นมองลึกเข้าสู่รายละเอียดข้างในของระบบยิ่งมองลึกมากเท่าใด ก็ยิ่งเห็นรายละเอียดของระบบย่อยได้มากขึ้นเท่านั้น

DFD ระดับที่ 0

ให้คือว่าระบบทั้งระบบเป็น Process หรือวงกลมหนึ่งวง มีลูกศรแทน Input และ Output ตามที่จำเป็น

DFD ระดับที่ 1

ให้แตกวงกลมที่ลำดับ 0 ออกเป็นวงกลมย่อย 2-5 วงตามความเหมาะสม

DFD ระดับที่ 2

ให้แตกวงกลมที่ลำดับ 1 ออกเป็นวงกลมย่อยลงไปอีกเท่าที่จะทำได้

DFD ระดับที่ 3

ถ้าจำเป็นก็ต้องตรวจดูว่า วงกลมใดในภาพลำดับที่ 2 ยังมีความซับซ้อนที่จำเป็นต้องแตกย่อยก็ต้องแตกย่อย ก็ต้องสร้างภาพประกอบด้วยวงกลมย่อยแทนวงกลมนั้นให้ได้รายละเอียดสุดท้าย

2.15.3 ประโยชน์ของการใช้ Data Flow Diagram

2.15.3.1 DFD ช่วยให้นักวิเคราะห์ระบบสามารถ

- สรุปข้อมูลที่เกี่ยวกับระบบ
- เข้าใจจึงปัญหาสำคัญของระบบและระบุส่วนของการทำงานที่ซ้ำซ้อน
- เข้าใจถึงความสัมพันธ์ระหว่างส่วนต่าง ๆ ของระบบและการประกอบกันเป็นระบบ
- พัฒนาระบบได้อย่างมีประสิทธิภาพ

2.15.3.2 DFD เป็นเอกสารร่วมที่ช่วยให้นักวิเคราะห์ระบบและผู้ใช้สามารถเข้าใจระบบและตรวจสอบความถูกต้องได้สองฝ่าย

2.15.3.3 ในการตรวจสอบเรื่องเวลาที่ใช้ในแต่ละขบวนการนั้น นักวิเคราะห์สามารถใช้ DFD เป็นเครื่องมือ

2.16 Entity Relationship Model

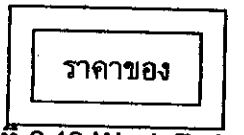
Entity Relationship Model หรือที่นิยมเรียกกันสั้นๆว่า ER Modelนี้ เป็น Data Model รูปแบบหนึ่งที่นิยมใช้ในการนำเสนอรายละเอียดต่างๆ ที่เกี่ยวข้องกับข้อมูลในฐานข้อมูลที่ออกแบบ ในการออกแบบฐานข้อมูลขึ้นใช้ในงานในระบบสารสนเทศ เนื่องจาก ER Model เป็นแบบจำลองที่มีรูปภาพที่ใช้แทนโครงสร้างทางด้าน Abstraction ต่างๆได้เป็นอย่างดี ซึ่งประกอบด้วยสัญลักษณ์ต่างๆ ต่อไปนี้

2.16.1 Entity เป็นรูปภาพที่ใช้แทน Class ของสิ่งของต่างๆ ซึ่งอาจเป็นทั้งสิ่งที่จับต้องได้ เช่น ห้องเรียน หรือไม่สามารถจับต้องได้ เช่น เวลาเรียน ซึ่งใน ER Model จะแบ่ง Entity ออกเป็น 2 ประเภท ดังนี้

2.16.1.1 Regular Entity (Strong Entity) ได้แก่ Entity ที่ประกอบด้วยสมาชิกที่มีคุณสมบัติ ซึ่งบ่งบอกถึงเอกลักษณ์ของแต่ละสมาชิกนั้น รูปภาพที่ใช้แทน Entity ได้แก่ รูปสี่เหลี่ยมผืนผ้า โดยมีชื่อของ Entity นั้นอยู่ภายใน

รูปที่ 2.11 Regular Entity

2.16.1.2 Weak Entity สมาชิกของ Entity ชนิดนี้ เป็นสมาชิกที่ไม่สามารถมีคุณสมบัติ ซึ่งบ่งบอกถึงเอกลักษณ์ของแต่ละสมาชิกนั้นได้ จะต้องอาศัยคุณสมบัติใดคุณสมบัติหนึ่งของ Regular Entity มาประกอบกับคุณสมบัติของตัวมันเอง รูปภาพที่ใช้แทน Entity ได้แก่ รูปสี่เหลี่ยมผืนผ้า 2 รูป ซ้อนกัน โดยมีชื่อของ Entity นั้นอยู่ภายใน



รูปที่ 2.12 Weak Entity

2.16.1.3 Property หรือ Attribute เป็นข้อมูลที่แสดงลักษณะและคุณสมบัติของ Entity จะสามารถแบ่งย่อยได้ดังนี้

- Simple Property ได้แก่ Property ที่ค่าภายใน Property นั้นไม่สามารถแบ่งย่อยได้อีก เช่น รหัสหมายเลขครุภัณฑ์ ชื่ออุปกรณ์ รูปภาพที่ใช้แทน Property ชนิดนี้ ได้แก่ วงรีที่มีเส้นเชื่อมต่อไปยัง Entity ที่เป็นเจ้าของ Property นั้น กำกับอยู่หลังวงกลม

- Composite Property ได้แก่ Property ที่ค่าภายใน Property นั้นสามารถแยกเป็น Property ย่อยได้อีก รูปภาพที่ใช้แทน Property ประเภทนี้ จะใช้วงกลมที่ต่อเชื่อมกับวงรี ของ Simple Property ที่เป็นเจ้าของ Composite Property นั้นแทน

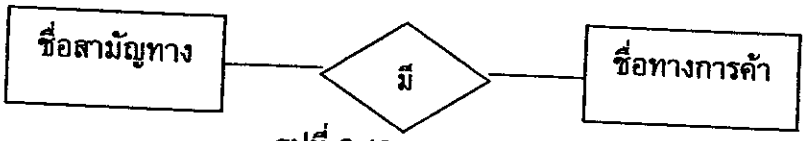
- Key เป็น Property ได้แก่ Property ที่มีค่าในแต่ละสมาชิกไม่ซ้ำกันเลย ซึ่งถูกนำมาใช้กำหนดความเป็นเอกลักษณ์ ให้กับแต่ละสมาชิกใน Entity รูปภาพที่ใช้แทน Key จะใช้วงกลมสีดำที่มีเส้นเชื่อมต่อไปยัง Entity ที่เป็นเจ้าของ Property นั้น โดยมีชื่อของ Property นั้นกำกับอยู่หลังวงกลม

- Single Value เป็น Property ที่มีข้อมูลภายใต้ค่าของ Property หนึ่งเพียงค่าเดียว เช่น Property "Equip_ID" ซึ่งเป็นอุปกรณ์ แต่ละชิ้นจะมีหมายเลขครุภัณฑ์ ได้เพียงค่าเดียวจะมีรหัสประจำตัวเพียงค่าเดียว

- Multi-valued Property เป็น Property ที่มีค่าของข้อมูลได้หลายค่า ภายใต้ค่าของ Property ใด Property หนึ่ง เช่น Property "Room" ที่ใช้ระบุวิชาเรียนในแต่ละห้อง ซึ่งแต่ละห้องจะมีวิชาที่ใช้ห้องเรียนนั้นได้หลายวิชา โดยรูปภาพที่ใช้แทน Property ประเภทนี้จะใช้วงกลมที่มีเส้นเชื่อมต่อไปยัง Entity ที่เป็นเจ้าของ Property โดยมีชื่อของ Property นั้นพร้อมกับ Mapping Cardinality กำกับอยู่ด้านหลัง

- Derived Property เป็น Property ที่ค่าของข้อมูลได้มาจากการนำค่าของ Property อื่นๆมาทำการคำนวณ ซึ่งค่าของ Property ประเภทนี้ จะต้องเปลี่ยนแปลงทุกครั้งเมื่อมีการเปลี่ยนแปลงค่าของ Property ที่ถูกนำมาคำนวณ เช่น Property Tot_Sal ซึ่งได้มาจากผลรวมของค่าใน Property "Income" ของ Entity "Employee" ซึ่งเป็นเงินเดือนที่พนักงานแต่ละคนได้รับในแต่ละเดือนสำหรับรูปภาพที่ใช้แทน Property ประเภทนี้ จะใช้วงกลมที่มีเส้นเชื่อมต่อไปยัง Entity ที่เป็นเจ้าของ Property โดยมีชื่อของ Property นั้นพร้อมกับคำว่า "Derived Data" กำกับอยู่ด้านหลัง

2.16.2 Relationship โดยทั่วไป จะกำหนดขึ้นจาก Entity ที่มี Property ร่วมกับสมาชิกของ Relationship จึงเกิดจากการจับคู่กันระหว่างสมาชิกของ Entity ที่มารวมกันภายใต้ Relationship นั้นโดย Relationship ที่สร้างขึ้นนี้ จะแทนความสัมพันธ์ใดความสัมพันธ์หนึ่งระหว่างสมาชิกของ Entity ที่มารวมกันภายใต้ Relationship นั้น ดังนั้นในการตั้งชื่อให้ Relationship จึงควรที่จะตั้งชื่อที่แสดงถึงความสัมพันธ์นั้นๆ สำหรับรูปภาพที่ใช้แทน Relationship ได้แก่ รูปสี่เหลี่ยมข้าวหลามตัดที่มีชื่อของ Relationship นั้น อยู่ภายในรูปภาพ Relationship จะไม่สามารถปรากฏอยู่เดี่ยวๆได้ จะต้องปรากฏคู่กับ Entity เสมอ เช่น



รูปที่ 2.13 Relationship

สำหรับ Relationship ที่ใช้กับ Weak Entity ก็จะใช้รูปภาพเดียวกันกับรูปภาพของ Relationship โดยทั่วไป

Relationship ระหว่าง Entity ใดๆ ไม่จำเป็นที่จะต้องมีเพียง Relationship เดียว ถ้าความสัมพันธ์ระหว่างสมาชิกใน Entity เหล่านั้นมีมากกว่า 1 ความสัมพันธ์ สมาชิกใน Entity ที่เกี่ยวข้องกับ Relationship จะถูกเรียกว่า Participant ซึ่งจำนวนของ Participant นี้จะถูกนำไปใช้ในการกำหนดประเภทของ Relationship ดังนี้

2.16.2.1 One-to-One Relationship เป็น Relationship ที่แต่ละ Participant ของ Entity หนึ่ง จะมีความสัมพันธ์กับอีก Participant ของ Entity หนึ่งได้เพียง Participant เดียว เช่นกรณีที่บุคคล 1 คน สามารถจองห้องได้เพียง 1 ห้อง

2.16.2.2 One-to-Many Relationship เป็น Relationship ที่แต่ละ Participant ของ Entity หนึ่ง จะมีความสัมพันธ์กับอีก Participant ของอีก Entity หนึ่ง มากกว่า 1 Participant เช่นกรณีที่บุคคล 1 คน สามารถจองห้องได้หลายห้อง

2.16.2.3 Many-to-Many Relationship เป็น Relationship ที่ Participant มากกว่า 1 Participant ของ Entity หนึ่ง จะมีความสัมพันธ์กับอีก Participant ของ Entity หนึ่งมากกว่า 1 Participant เช่นกรณีที่บุคคล 1 คน สามารถจองห้องได้หลายห้อง และแต่ละห้องก็สามารถมีบุคคลมาจองได้มากกว่า 1 คน

แผนภาพ ER Model ที่ดีนั้นจะต้องสามารถอธิบายโครงสร้างของข้อมูลได้อย่างดี และครบถ้วน รวมทั้งต้องมีรูปแบบที่ง่ายต่อความเข้าใจ มีความชัดเจน และต้องไม่ซับซ้อนด้วย

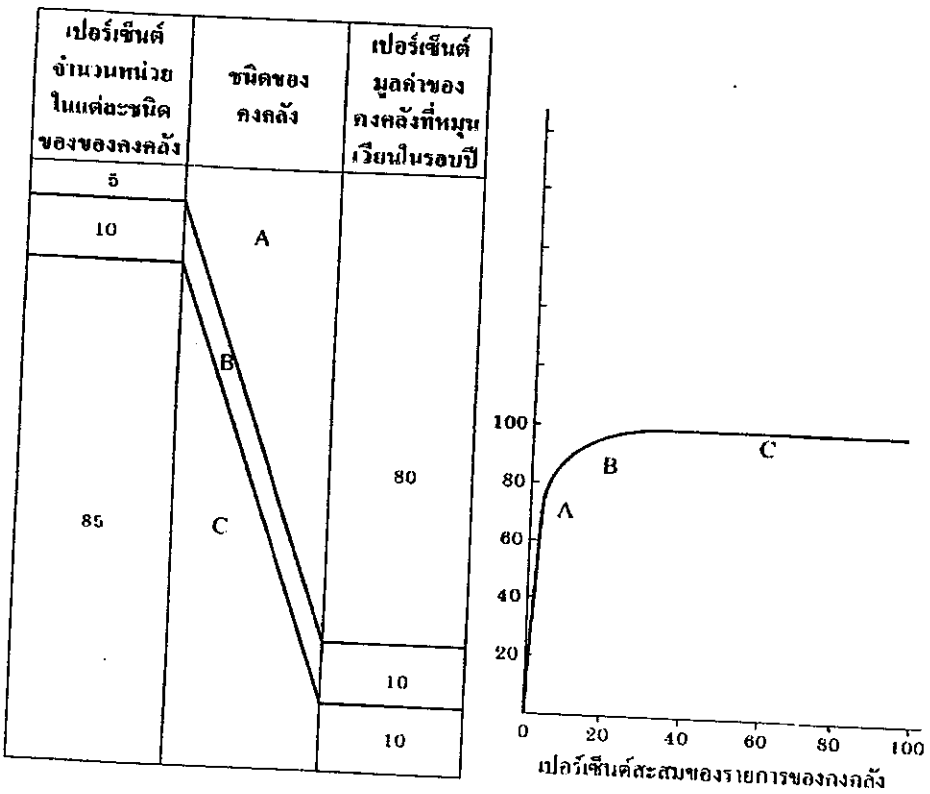
2.17 หลักการและทฤษฎีเกี่ยวกับการจัดเก็บของคงคลัง

2.17.1 การวิเคราะห์ความสำคัญของของคงคลังแบบ ABC

การควบคุมของคงคลังเป็นงานที่ทำขึ้นเพื่อให้ค่าใช้จ่ายที่เกิดขึ้นจากการจัดให้มีของคงคลังต่ำที่สุด แต่อย่างไรก็ตาม บริษัทมักมีของคงคลังมากมายหลายชนิดไม่ว่าจะเป็นวัตถุดิบ ชิ้นส่วนประกอบหรือสินค้าสำเร็จรูป ตลอดจนของใช้สำนักงาน ถ้าเราให้ความสนใจควบคุมของเล่านี้ทั้งหมดในคลังอย่างใกล้ชิดก็จะทำให้สิ้นเปลืองค่าใช้จ่ายและเสียเวลามาก ของบางชนิดถึงแม้ว่าจะมีปริมาณการใช้มาก แต่ถ้าราคาต่ำมากๆ เช่น ของจำพวกสกรูและนอต การให้ความสนใจอย่างใกล้ชิดในการควบคุมของคงคลังชนิดนี้ก็จะไม่คุ้มกับส่วนที่ประหยัดได้ แต่ของบางอย่างถึงแม้เราจะมีจำนวนการใช้น้อย คิดเป็นจำนวนแค่ 5-10% ของจำนวนของ

ทั้งหมด แต่มูลค่าการใช้อาจจะสูงถึง 80% ของมูลค่าทั้งหมด ดังนั้น นอกเหนือจากส่วนที่เป็นนโยบายของฝ่ายบริษัทแล้ว การควบคุมพัสดุคงคลังควรพิจารณาถึงความเหมาะสมของชนิดของคงคลังด้วย ทางที่ดีที่สุดควรจะจำแนกประเภทของคงคลังออกเป็นชนิดที่มีความสำคัญมากและที่มีความสำคัญรองลงไป วิธีการจำแนกชนิดของของคงคลังที่รู้จักกันทั่วไปคือ วิธี ABC ซึ่งเป็นเทคนิคที่มีหลักการจำแนกของคงคลังตามจำนวนเงินของของคงคลังที่หมุนเวียนในคลังในรอบปี

ความหมายของการจำแนกของคงคลังตามระบบ ABC จะแบ่งของคงคลังออกเป็น 3 ชนิดคือ ชนิด A เป็นจำนวนเงินที่หมุนเวียนในคลังในรอบปีมีมูลค่าสูงที่สุด ชนิด B มีมูลค่าสูงปานกลาง และชนิด C มีมูลค่าต่ำที่สุด เหตุผลที่ต้องจำแนกของคงคลังในลักษณะนี้คือ การจำแนกเพื่อกำหนดความสำคัญมากน้อยของของคงคลัง ถ้าเขียนกราฟระหว่างค่าใช้จ่ายและคำนวณชนิดของคงคลังจะได้ดังรูปที่ 2.14 ซึ่งเรียกว่า Pareto Curve ในรูปแสดงว่ามีของคงคลังประมาณ 5 % ของจำนวนหน่วยทั้งหมดที่มีมูลค่าสูงถึง 80% ของมูลค่าของคงคลังทั้งหมด จึงถือว่า มีความสำคัญสูงมาก จึงจักให้ของคงคลังชนิดนี้อยู่ในประเภท A ส่วนที่เหลือที่มีความสำคัญน้อยลงไปก็จะจัดแบ่งให้เป็นประเภท B และC ตามลำดับ



รูปที่ 2.14 แสดงการจำแนกชนิดของคงคลังโดย ABC เทคนิค

ป QA
76.6
.ร155ก
2547

4840129

สำนักหอสมุ
15 ก.ค. 25

จำนวนเปอร์เซ็นต์ที่เราใช้ในการจำแนกของคงคลังแต่ละประเภทควรจะเป็นเท่าไรนั้น ย่อมขึ้นอยู่กับสภาพการณ์ของการมีของคงคลัง ของคงคลังประเภท A มักจะมีราคาสูง การตั้งเกณฑ์ราคาไว้ระดับหนึ่งจะช่วยให้แบ่งประเภทได้ง่ายขึ้น แต่ช่วงที่จะใช้ เป็นชนิด B มักจะกำหนดได้ยาก อย่างไรก็ตาม แต่ละบริษัทก็มักจะมีวิธีและแนวทางที่เป็นของตนเอง Magee และ Boodman ได้ให้หลักในการกำหนดประเภทความสำคัญดังนี้

ประเภท A มีของคงคลังประมาณ %5 ถึง 10% ของของคงคลังทั้งหมดที่มีมูลค่าสูงสุด

ประเภท B มีของคงคลังประมาณ 20% ถึง 30% ซึ่งมีมูลค่ารองลงมา

ประเภท C คือปริมาณของคงคลังทั้งหมดที่เหลือซึ่งคิดเป็นต้นทุนเพียงเล็กน้อยของต้นทุนทั้งหมด

แนวความคิดในการนำเอาระบบ ABC เทคนิคไปใช้ในเรื่องเกี่ยวกับของคงคลัง พอจะแยกออกเป็นหัวข้อได้ดังนี้

2.17.2 ระดับการควบคุม

ประเภท A ต้องมีการควบคุมปริมาณและการสั่งของอย่างใกล้ชิดเข้มงวด การสั่งและการใช้ของจะต้องมีบันทึกรายการให้เป็นไปอย่างสมบูรณ์และถูกต้อง มีผู้ควบคุมและตรวจสอบอยู่เสมอๆ

ประเภท B มีการควบคุมตามปรกติ กล่าวคือมีการตรวจสอบของคงคลังเป็นระยะๆ เช่นทุก 3 เดือน เป็นต้น บันทึกและศึกษาดูว่ามีการเปลี่ยนแปลงมากน้อยเพียงใด

ประเภท C การควบคุมไม่ต้องเข้มงวดเป็นไปอย่างง่ายๆ ไม่จำเป็นต้องมีการจดบันทึกรายการแต่ควรมีการตรวจนับเป็นครั้งแรก ของในกลุ่มนี้ควรมีของจำนวนมากและสั่งซื้อครั้งละมากๆ เพื่อป้องกันการขาดแคลน

2.17.3 ระดับการสั่งการ

ประเภท A ต้องมีการสั่งอย่างระมัดระวังในเรื่องการกำหนดของการสั่งซื้อและจุดสั่งซื้อที่แน่นอน ต้องมีการตรวจสอบอยู่เสมอ เพื่อลดจำนวนของเท่าที่จะเป็นไปได้ หรือเพื่อป้องกันการขาดแคลนของคงคลัง

ประเภท B โดยทั่วไปขนาดของการสั่งซื้อและจุดสั่งซื้อ จะวิเคราะห์โดยใช้สูตร EOQ มีการตรวจสอบทุกงวด 3-4 เดือน หรือเมื่อเกิดการเปลี่ยนแปลงอย่างมาก

ประเภท C การสั่งซื้อครั้งละมากๆ โดยไม่จำเป็นต้องคำนวณหา EOQ หรือสั่งสินค้าไว้ใช้ตลอด 1 ปี แม้ว่าสินค้าจะมีเหลืออยู่เป็นจำนวนมาก

2.18 โปรแกรม Microsoft Access

เป็นโปรแกรมที่ใช้จัดการกับฐานข้อมูล ซึ่งจะช่วยให้เราจัดการกับข้อมูลปริมาณมากๆ ได้อย่างง่ายดาย ทั้งในแง่การจัดเก็บข้อมูล, การค้นหาข้อมูล, การจัดทำรายงานข้อมูล และการสำรองข้อมูลเก็บไว้

โปรแกรม Microsoft Access นั้นเป็นโปรแกรมประเภทที่เรียกว่า ระบบจัดการฐานข้อมูล (DBMS) ซึ่งโปรแกรมประเภทนี้ที่มักรู้จักกันก็เช่น dBase, FoxPro, Oracle, SQL, Server, DB2 ซึ่งมีขนาดและขอบเขตการใช้งานที่แตกต่างกัน ซึ่งการใช้งานโปรแกรม Microsoft Access นั้นจะขออธิบายอย่างคร่าวๆ ใน ภาคผนวก ก.

2.19 โปรแกรม Visual Basic

Visual Basic เป็นโปรแกรมที่ใช้สร้างโปรแกรมประยุกต์ สำหรับระบบปฏิบัติการ Windows

Visual เป็นส่วนที่หมายถึงเมธอด (Method) ในการติดต่อแบบ graphical user interface (GUI) ซึ่งการสร้างทำได้โดยการเพิ่มอ๊อบเจค ลงบนฟอร์มที่ทำหน้าที่ติดต่อกับผู้ใช้ผ่านจอภาพ

Basic เป็นส่วนที่หมายถึงภาษา BASIC (Beginners ALL Purpose Symbolic Instruction Code) โดยVisual Basic ได้เปลี่ยนแปลงจากภาษา BASIC ดั้งเดิม ด้วยการเพิ่มประโยคคำสั่ง ฟังก์ชัน และคีย์เวิร์ด ที่เกี่ยวข้องโดยตรงกับ GUI

2.19.1 แนวคิดของ Visual Basic

โปรแกรมประยุกต์ Visual Basic เป็นการพัฒนาในสภาพแวดล้อมของ windows ซึ่งแนวคิดพื้นฐานในการทำงานของระบบ Windows ที่สำคัญมี 3 ประการ คือ window, events และ ข่าวสาร (message)

โปรแกรมประยุกต์ Visual Basic มีการทำงานแบบ Event-Driven ที่เป็นการประมวลผลตามคำสั่งในแต่ละส่วนเพื่อตอบสนองต่อ event ซึ่ง event เหล่านี้สามารถเปลี่ยนโดยการทำงานของผู้ใช้ ข่าวสารของระบบหรือโปรแกรมประยุกต์อื่น หรือภายในโปรแกรมเดียวกัน ลำดับการทำงานของ event จะจัดลำดับโดยจากการประมวลคำสั่ง

2.19.2 โปรแกรมประยุกต์ Visual Basic

การสร้างโปรแกรมประยุกต์ Visual Basic ประกอบด้วยขั้นตอนหลัก 3 ขั้นตอน คือ

2.19.2.1 การสร้างอินเตอร์เฟซ โดยมีฟอร์มเป็นอ๊อบเจคพื้นฐานและเป็นที่วางตัว control สำหรับการติดต่อกับผู้ใช้

2.19.2.2 ตั้งค่าคุณสมบัติ เป็นการกำหนดพฤติกรรมและการทำงานให้กับอ๊อบเจคต่างๆ

2.19.2.3 การเขียนคำสั่ง เป็นการควบคุมการประมวลผลผ่าน procedure ที่กำหนด

2.19.3 โครงสร้างของ Project

Project ที่สร้างขึ้น สามารถประกอบด้วยไฟล์และอ๊อบเจค ดังนี้

2.19.3.1 Form Modules (.frm) สามารถเก็บข้อความของฟอร์ม ตัว control ที่มีการกำหนดค่าคุณสมบัติ และเก็บระดับการประกาศค่าระดับฟอร์มของค่าคงที่, ตัวแปร และ procedure ภายนอก, event procedure และ procedure ทั่วไป

2.19.3.2 Class Modules (.cls) คล้ายกับ form module แต่แตกต่างที่จะไม่มีการรบกวนจากผู้ใช้ทั่วไป ใน class module สามารถสร้างอ๊อบเจคของผู้พัฒนาโปรแกรมรวมถึงคำสั่ง เมธอด และคุณสมบัติ

2.19.3.3 Standard Modules (.bas) สามารถเก็บการประกาศค่า public และระดับโมดูล ของค่าคงที่ประเภทตัวแปร Procedure ภายนอก และ Public Procedure

2.19.3.4 Resource File (.res) เก็บไฟล์บิตแม็บ, ข้อความ และข้อมูลอื่นที่สามารถเปลี่ยนแปลงได้โดยไม่ต้องแก้ไขคำสั่ง เช่น ผู้พัฒนาโปรแกรมวางแผนการสร้างโปรแกรมประยุกต์เป็นภาษาอื่น จะสามารถเก็บข้อความที่ติดต่อกับผู้ใช้ และไฟล์บิตแม็บใน resource file แต่ project สามารถมี resource file เพียง 1 ไฟล์ต่อ project

2.19.3.5 ActiveX Documents (.dob) คล้ายกับฟอร์ม แสดงด้วย internet browser โดย Visual Basic ชุด Professional และ Enterprise สามารถสร้าง ActiveX document

2.19.3.6 User Control และ Property Page Modules โดย user control (.ctl) และ property page (.pag) เป็นโมดูลที่คล้ายกับฟอร์ม แต่ใช้ในการสร้างตัว control ประเภท ActiveX และ property page ที่ใช้งานร่วมกัน สำหรับการแสดงคุณสมบัติของการออกแบบ Visual Basic ชุด Professional และ Enterprise สามารถสร้าง ActiveX control ได้

2.19.3.7 Components หมายถึงไฟล์ หรือโมดูลที่เพิ่มเข้าไปใน project ซึ่ง components หลายประเภทที่สามารถเพิ่มเข้าไปใน project

2.19.3.8 ActiveX Control (.ocx) เป็นตัว control เพิ่มเติมที่สามารถเพิ่มเข้าไปใน toolbox และฟอร์ม เมื่อมีการติดตั้ง Visual Basic ไฟล์ที่เก็บตัว control ของ Visual Basic จะได้รับการคัดเลือกไปยังไดเรคทอรีร่วม (Windows/System) นอกจากนี้ตัว control ประเภท ActiveX มีแหล่งที่กว้างขวาง และผู้พัฒนาโปรแกรมสามารถสร้างตัว control ของตัวเองได้ในชุด Professional และ Enterprise

2.19.3.9 Insertable Objects หมายถึง component เช่น ไฟล์ Excel ที่สามารถสร้างขึ้นเพื่อเป็นส่วนหนึ่งของโปรแกรมในรูปแบบ Integrated Solutions โดยรูปแบบนี้สามารถเก็บข้อมูลที่มีฟอร์แมตที่แตกต่างกัน เช่น ไฟล์กระดาษทำการ (เช่น Excel) บิตแม็บ และข้อความ ซึ่งมาจากโปรแกรมประยุกต์ที่แตกต่างกัน

2.19.3.10 References เป็นการเพิ่มตัว control ประเภท ActiveX ภายนอกมาใช้ในโปรแกรมประยุกต์ การอ้างถึงทำได้โดยไดอะล็อกของ Reference ซึ่งเข้าถึงด้วย เมนู Reference ในเมนู Project

2.19.3.11 ActiveX Designer เป็นเครื่องมือในการออกแบบ class ที่ต้องการอ็อบเจค โดยการออกแบบการติดต่อสำหรับฟอร์มที่เป็น designer เริ่มต้น ส่วน designer เพิ่มเติมมีให้จากแหล่งอื่น

2.19.3.12 Standard Control เป็นตัว control มาตรฐานที่ Visual Basic มีไว้ให้เช่น ปุ่มคำสั่ง combo box โดยจะอยู่ใน toolbox

รายละเอียดการใช้งานอย่างคร่าวๆอธิบายไว้ที่ ภาคผนวก ข.

2.20 งานวิจัยที่เกี่ยวข้องและเอกสารประกอบการศึกษา

จากการศึกษาผลการวิจัยที่ผ่านมา พบว่ามีผลงานวิจัยที่เกี่ยวข้องกับการจัดระบบฐานข้อมูลมากมาย ทั้งเรื่องฐานข้อมูลอาคาร ข้อมูลบุคคลากร และการประยุกต์ใช้ต่างๆมากมาย โดยผู้จัดทำการทำวิจัยครั้งนี้ได้รวบรวมรายละเอียดบางส่วนเกี่ยวกับงานวิจัยที่ผ่านมา โดยมีรายละเอียดการวิจัยดังต่อไปนี้

นายจักราวุธ แก้วสุวรรณ์ และ นายปรีชา การะสิน ปีการศึกษา 2545 ได้จัดทำการวิจัยเกี่ยวกับ ระบบตรวจสอบการเข้าใช้ห้องคอมพิวเตอร์ (Room Access System) โดยโปรแกรมที่ใช้เขียนข้อมูลในครั้งนี้ คือ Microsoft Visual Basic 6.0 โดยมีวัตถุประสงค์ที่จะศึกษาและพัฒนาในเรื่องของระบบตรวจสอบการเข้าใช้ห้องคอมพิวเตอร์ เพื่อนำมาเก็บข้อมูลบุคคลที่เข้าใช้ห้องคอมพิวเตอร์ และความสะดวกสบายของผู้ใช้ห้องคอมพิวเตอร์ โดยโปรแกรมที่จัดทำขึ้นนำไปใช้งานในส่วนของห้องคอมพิวเตอร์ของคณะวิศวกรรมศาสตร์ ความสามารถของโปรแกรมกล่าวคือ โปรแกรมจะทำการบันทึก แก้ไขเปลี่ยนแปลง และลบค่าข้อมูลของบุคคล ตลอดจนสามารถควบคุมเวลาการใช้งาน รวมถึงการสร้างรายงานของข้อมูลที่จัดเก็บเอาไว้ได้ โดยการเรียกดูข้อมูลจะต้องเข้าไปในส่วนของการควบคุม

นางสาวนัฐกร เอมะพลานนท์ และ นางสาวปาริชาติ กุลจู ปีการศึกษา 2544 ได้จัดทำการวิจัยเกี่ยวกับ ระบบฐานข้อมูลจัดการอาคารสารสนเทศสถานที่ (Building Information System) โดยมีวัตถุประสงค์จัดทำขึ้นเพื่อ อำนวยความสะดวกให้กับผู้ใช้ในการค้นหาข้อมูลและกำหนดจุดประ- สงค์ การใช้ห้องต่างๆภายในอาคารเรียนคณะวิศวกรรมศาสตร์ โดยตัวโปรแกรมได้พัฒนาขึ้นมาจากโปรแกรมภาษา Java โดยโปรแกรมฐานข้อมูลการจัดการสารสนเทศอาคารสถานที่นี้มีความสามารถของการประมวลผลข้อมูลดังต่อไปนี้กล่าวคือ สามารถเก็บรวบรวม ค้นหา แก้ไข ตรวจสอบ และแสดงผลเกี่ยวกับ ข้อมูลอุปกรณ์ของอาคาร, ข้อมูลกิจกรรมต่างๆที่เกิดขึ้นในแต่ละห้อง ได้

นายกฤษดา ขันกสิกรรม และ นายอุทิศ คำหนู ปีการศึกษา 2542 ได้จัดทำการวิจัยเกี่ยวกับระบบฐานข้อมูลห้องสมุดคณะวิศวกรรมศาสตร์ (Library's database of Faculty of Engineering) โดยมีวัตถุประสงค์ที่จะศึกษาระบบการทำงานของ ระบบฐานข้อมูลแบบ Stand alone และมีการเขียนโปรแกรมติดต่อกันฐานข้อมูลโดยใช้ Microsoft Visual Basic 6.0 และการเขียน Web Application ติดต่อกันฐานข้อมูลโดยใช้ ASP (Active Sever Page) โดยนำมาประยุกต์ใช้กับระบบฐานข้อมูลห้องสมุดคณะวิศวกรรมศาสตร์ ซึ่งมีผลการดำเนินงานดังต่อไปนี้ โปรแกรมการจัดการระบบฐานข้อมูลห้องสมุดคณะวิศวกรรมศาสตร์นั้น สามารถรองรับการทำงาน ในด้านต่างๆที่มีความจำเป็นต่อการให้บริการของห้องสมุดได้ดีในระดับหนึ่ง โดยโปรแกรมสามารถทำงานในระบบ Stand alone Database ดังนั้นโปรแกรมนี้จึงสามารถที่จะ

นำไปใช้ได้กับห้องสมุดขนาดเล็กและขนาดกลาง หรือห้องสมุดที่มีจุดบริการหลายๆที่ ที่ห่างไกล อีกทั้งยังให้บริการค้นหาหนังสือ ในห้องสมุด ผ่านระบบเครือข่ายคอมพิวเตอร์ได้อีกด้วย

นางสาววันดี เทียนบุตร และ นางสาวอัจฉรี สุริยกุล ณ อยุธยา ได้จัดทำการวิจัยเกี่ยวกับระบบฐานข้อมูลสถานที่ฝึกงานของนิสิตภาควิชาวิศวกรรมอุตสาหการ มหาวิทยาลัยนเรศวร โดยจัดทำฐานข้อมูล และฟอร์มของ Access สำหรับการทำงาน เพื่อให้มีการจัดเก็บข้อมูลอย่างเป็นระเบียบ มีการออกเอกสาร การค้นหา และข้อมูลที่ทันสมัยตลอดเวลา อีกทั้งยังเชื่อมข้อมูลผ่านเครือข่ายอินทราเน็ต (Intranet) จากผลการดำเนินงานวิจัยสามารถทำการรับข้อมูลระหว่างนิสิตกับ ผู้ดำเนินงานฝ่ายเอกสารของภาควิชา ด้านข้อมูลสถานที่ฝึกงาน, การลงทะเบียนจองสถานที่ฝึกงาน และเอกสารที่เกี่ยวข้องกับการฝึกงานได้

นายณัฐพล หอมกลิ่น และ นายธวัชชัย บัววัฒน์ ปีการศึกษา 2543 ได้ทำการพัฒนาและปรับปรุงระบบคงคลังอะไหล่ในกรณีศึกษา บริษัทคาร์กิลล์สยาม จำกัด จังหวัดพิษณุโลก โดยทำการศึกษาระบบที่มีอยู่เดิมแล้วพัฒนาระบบนั้นให้มีประสิทธิภาพและเหมาะสมทั้งในด้าน การจัดเก็บ การเบิกจ่าย รวมถึงความสะดวกในการตรวจนับอะไหล่ที่คงเหลืออยู่ ในการพัฒนาระบบนี้ได้มีการนำระบบต่างๆ เข้ามาใช้ในการกำหนดรหัส การออกแบบตำแหน่งของอะไหล่ วิธีเบิกจ่าย การตรวจนับ รวมไปถึงวิธีการกำหนดจุดสั่งซื้อและจุดสต๊อกเพื่อความปลอดภัย (Safety Stock)

นายจักรภพ เกษสุวรรณ และนางสาวสุชาสิณี วรรณศิลป์ ปีการศึกษา 2540ได้ทำการศึกษา การพัฒนาและปรับปรุงระบบควบคุมวัสดุคงคลังของภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร ให้มีประสิทธิภาพในด้านการตรวจรับ การจัดเก็บ การเบิกจ่าย ตรวจสอบ และทำให้เกิดความสะดวกในการดำเนินงานด้านต่างๆ มากขึ้น โดยพัฒนาระบบควบคุมวัสดุคงคลังใหม่ รวมทั้งการนำคอมพิวเตอร์มาใช้ในการบริหารควบคุม โดยการเขียนโปรแกรม Micro Excel ภายใต้ Microsoft Excel

นายดานุภพ หนูเทพ และ นายวิทยา ยิ้มสาระ ปีการศึกษา 2544 ได้ทำการศึกษาพัฒนาและปรับปรุงคงคลังอะไหล่ที่ใช้ในการซ่อมบำรุงเครื่องจักรที่ใช้ในโรงงาน กรณีศึกษาโรงงานผลิตภัณฑ์นมสุโขทัย จังหวัดสุโขทัย เพื่อเพิ่มประสิทธิภาพให้เหมาะสมทั้งในด้านการจัดเก็บ รวมถึงความสะดวกในการตรวจนับ โดยมีการนำคอมพิวเตอร์มาใช้ในการปรับปรุงและพัฒนาระบบอะไหล่คงคลัง