

บทที่ 11

โครงสร้างข้อมูลและระบบฐานข้อมูล

โครงสร้างของระบบนี้

11.1 ลำดับชั้นข้อมูล

11.1.1 ข้อมูลพื้นฐานแบบตรรกะและแบบกายภาพ

11.2 โครงสร้างการจัดเก็บเพิ่มข้อมูล วิธีการเข้าถึงข้อมูล และการประมวลผล

11.3 การจัดโครงสร้างเพิ่มข้อมูลแบบเรียงลำดับและการประมวลผลการจัดโครงสร้างเพิ่มข้อมูลแบบเรียงลำดับ

11.4 การประมวลฐานข้อมูล (DATABASE PROCESSING)

11.4.1 หลักการของฐานข้อมูล (Database Concept)

11.4.2 ระบบประมวลผลฐานข้อมูล (DBMS:Database Management System)

11.4.3 โครงสร้างฐานข้อมูล (Database Structures)

11.4.4 การพัฒนาฐานข้อมูล (Developing a Database)

11.5 ประเภทของฐานข้อมูล (Type of Database)

11.6 ฐานข้อมูลแบบกระจาย (Distributed Database)

11.1 ลำดับชั้นข้อมูล

ลองจินตนาการคุณภาพของข้อมูลหรือสารสนเทศที่เรานำมาใช้งานถูกจัดเก็บอย่างไม่เป็นระเบียบ จะทำให้เรามีความยุ่งยากในการนำข้อมูลขึ้นมาใช้สักเพียงใด ดังนั้นข้อมูลหรือสารสนเทศที่ถูกจัดเก็บในระบบคอมพิวเตอร์จึงควรมีการจัดเก็บให้เป็นระเบียบและมีโครงสร้างที่สามารถนำมาใช้หรือแก้ไขได้ง่าย ในบทนี้จะกล่าวถึงวิธีการจัดเก็บข้อมูล วิธีการประมวลผลข้อมูลในรูปแบบต่างๆ

เพื่ออำนวยความสะดวกและการทำงานของผู้ใช้เราได้จัดแบ่งโครงสร้างของการจัดเก็บข้อมูลออกเป็นระดับชั้น ดังนี้คือ ตัวอักษร (characters) รายการข้อมูล (fields) ระเบียบ (record) และแฟ้มข้อมูล (data file) ซึ่งการจัดแบ่งนี้เราสามารถเทียบได้กับการจัดแบ่งข้อในจดหมายเป็นชั้นย่อยซึ่งประกอบด้วย คำ ประโยค และย่อหน้า

ตัวอักษร (Character)

ตัวอักษร (character) คือระดับย่อยขั้นพื้นฐานของโครงสร้างข้อมูลที่เราจัดแบ่ง ซึ่งประกอบด้วยข้อมูลเดี่ยวๆของตัวอักษร (alphabetic) ตัวเลข (numeric) หรือ สัญลักษณ์ต่างๆ ซึ่งการจัดแบ่งนี้อาจมีการโต้แย้งว่าระดับขั้นพื้นฐานของข้อมูลที่มีมากกว่านี้คือ บิต (bit) หรือ ไบต์ (byte) เนื่องจากส่วนย่อยเหล่านี้ถูกใช้เป็นระดับขั้นข้อมูลพื้นฐานที่ใช้ในระบบคอมพิวเตอร์ แต่ในมุมมองของผู้ใช้แล้วตัวอักษรเป็นระดับขั้นย่อยพื้นฐานที่สุดที่ผู้ใช้สามารถที่จะมองเห็นและจัดการกับข้อมูลได้

รายการข้อมูล (Field)

รายการข้อมูล คือ ระดับขั้นในการมองข้อมูลที่สูงกว่าตัวอักษร ซึ่งประกอบด้วยกลุ่มของข้อมูลของตัวอักษร เช่น กลุ่มตัวอักษรซึ่งแทนความหมายของชื่อคน หรือกลุ่มของตัวเลขซึ่งแทนความหมายของจำนวนเงินที่ซื้อสินค้า เป็นต้น ระดับขั้นนี้บางครั้งเรียกว่า *item* หรือ *คำ (word)* ข้อมูลของรายการข้อมูลนี้จะแสดงคุณลักษณะของข้อมูล (*attribute*) ของวัตถุ เช่น คน สถานที่หรือเหตุการณ์ต่างๆ เช่น ฟิล์มของอายุคนซึ่งแสดงถึงคุณสมบัติส่วนตัวของแต่ละคน

ระเบียบ (Record)

คือข้อมูลของฟิล์มซึ่งมีความสัมพันธ์กันถูกนำมารวบรวมไว้ด้วยกัน ดังนั้นเราสามารถกล่าวได้ว่าระเบียบแสดงถึงการรวบรวมคุณลักษณะ (*attribute*) ที่อธิบายคุณสมบัติของวัตถุหรือสิ่งของ เช่น ระเบียบข้อมูลเงินเดือนของพนักงานซึ่งประกอบด้วย ชื่อพนักงาน เลขประกันสังคม และอัตราเงินเดือน ระเบียบแบ่งออกเป็น 2 ชนิด คือ Fixed-length record ซึ่งมีขนาดของรายการในแต่ละ

ระเบียบมีขนาดแน่นอนเหมือนกัน และ Variable-length record ซึ่งมีขนาดของรายการข้อมูลที่มีความยาวไม่แน่นอน

เพิ่มข้อมูล (File)

การรวมกลุ่มกันของระเบียบที่มีความสัมพันธ์เป็นเรื่องเดียวกันเรียกว่า เพิ่มข้อมูล (data file) หรือ data set ดังนั้นเพิ่มข้อมูลเงินเดือนของพนักงานจึงประกอบด้วยระเบียบเงินเดือนทั้งหมดภายในบริษัท การจัดประเภทเพิ่มข้อมูลจะแบ่งตามประเภทของงานที่นำไปใช้ เช่น เพิ่มข้อมูลเงินเดือน หรือ เพิ่มข้อมูลบัญชีรายรับ นอกจากนี้แล้วเพิ่มข้อมูลยังถูกจัดแบ่งตามลักษณะการประมวลผล ได้แก่ **transaction file** ซึ่งเพิ่มข้อมูลประเภทนี้จะถูกนำไปประมวลผลในช่วงเวลาที่กำหนด โดยจะนำไปปรับปรุงกับเพิ่มข้อมูลหลัก (**master file**) ซึ่งข้อมูลที่เก็บในเพิ่มข้อมูลนี้เป็นข้อมูลมาก่อนข้างดาว เพิ่มข้อมูลทั้งสองเป็นสิ่งที่ใช้เก็บข้อมูลเพื่อนำไปใช้ในการประมวลผล **เพิ่มคำสั่งคอมพิวเตอร์ (programe file)** เป็นเพิ่มซึ่งบรรจุคำสั่งคอมพิวเตอร์เพื่อสั่งให้คอมพิวเตอร์ให้ทำงาน

ฐานข้อมูล (Database)

การประมวลผลข้อมูลด้วยระบบคอมพิวเตอร์ในยุคแรกๆนั้นมีความยุ่งยากอย่างมากในเรื่องของการจัดการกับข้อมูลในระดับต่างๆ เพราะเมื่อผู้ใช้ต้องการข้อมูลจะต้องทราบหมายเลขระเบียบ ตำแหน่งของรายการข้อมูลและเพิ่มข้อมูลที่จัดเก็บ อีกทั้งข้อมูลที่ถูกจัดเก็บอาจมีความซ้ำซ้อนกัน เช่น การจัดเก็บข้อมูลเพื่อใช้เป็นสารสนเทศในการวางแผนการตลาด อาจมีข้อมูลเดียวกับในเรื่องของการขายสินค้า ในการประมวลผลสมัยใหม่ได้เพิ่มระดับชั้นการมองข้อมูล เพื่อง่ายต่อการจัดการข้อมูล ซึ่งได้พัฒนามาเป็นระบบฐานข้อมูล (database) ที่มีใช้กันในปัจจุบันนี้

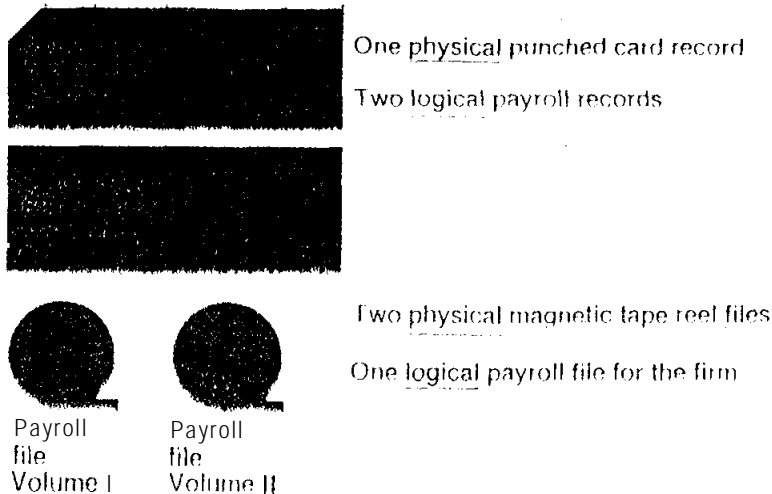
ฐานข้อมูล คือ การจัดเก็บเพิ่มข้อมูลต่างๆที่มีความสัมพันธ์กัน ซึ่งข้อมูลที่ถูกจัดเก็บนี้จะไม่มีการซ้ำซ้อนทางตรรก (logical) เช่น ฐานข้อมูลของบุคลากรจะประกอบไปด้วยเพิ่มข้อมูลต่างๆกัน เช่น เพิ่มข้อมูลพนักงาน เพิ่มเงินเดือน ส่วนคำว่า data bank จะหมายถึงการรวบรวมเพิ่มข้อมูลต่างๆเข้าไว้ด้วยกัน

11.1.1 ข้อมูลพื้นฐานแบบตรรกและแบบกายภาพ (Logical and Physical Data Elements)

ข้อมูลพื้นฐานแบบกายภาพ (Physical Data Element) เป็นหน่วยของข้อมูลซึ่งมีความสัมพันธ์กับชนิดของสื่อที่ใช้ในการจัดเก็บข้อมูล เช่น บัตรเจาะรู เทปแม่เหล็ก หรือจานแม่เหล็กที่ใช้ในการจัดเก็บข้อมูล ตัวอย่างข้อมูลแบบกายภาพ เช่น ข้อมูลในบัตรเจาะรูหนึ่งใบหมายถึงระเบียบทางกายภาพหนึ่งระเบียบ (physical record) หรือข้อมูลในเทปแม่เหล็กหนึ่งม้วนหมายถึงเพิ่มข้อมูลทางกายภาพ (physical file)

ข้อมูลพื้นฐานแบบตรรกะ (Logical Data Element) เป็นหน่วยของข้อมูลซึ่งไม่ขึ้นกับชนิดของอุปกรณ์ที่ใช้ในการจัดเก็บ เช่น เทปแม่เหล็กหลายตัวอาจใช้ในการจัดเก็บแฟ้มข้อมูลทางตรรกะเพียงหนึ่งแฟ้มข้อมูลเท่านั้น

รูปที่ 11.1 แสดงตัวอย่างความสัมพันธ์ระหว่างการจัดเก็บข้อมูลทางกายภาพและทางตรรกะ โดยอัตราเจาะรูหนึ่งโหลประกอบด้วยข้อมูลเงินเดือนของพนักงานสองคน และแฟ้มข้อมูลเงินเดือนจัดเก็บในเทปแม่เหล็กสองตัว



รูปที่ 11.1 ความสัมพันธ์ระหว่างข้อมูลแบบตรรกะและแบบกายภาพ

11.2 โครงสร้างการจัดเก็บแฟ้มข้อมูล วิธีการเข้าถึงข้อมูล และการประมวลผล

(File Organization Access and Processing)

อุปกรณ์ที่ใช้ในการจัดเก็บแฟ้มข้อมูลมีหลายชนิด ดังนั้นจึงต้องมีการจัดโครงสร้างการจัดเก็บเพื่อให้สอดคล้องกับการเข้าถึงข้อมูล (access) อันได้แก่ การจัดเก็บข้อมูล (store) การหาตำแหน่งของข้อมูล (locate) การดึงข้อมูล (retrieve) เพื่อการประมวลผล ในบทนี้จะกล่าวถึงหลักการอย่างคร่าวๆ ในเรื่องของการจัดโครงสร้างแฟ้มข้อมูลที่สำคัญๆ ตลอดจนวิธีการเข้าถึงแฟ้มข้อมูลและการประมวลผลแฟ้มข้อมูล

ในบทที่ 5 เราได้กล่าวถึงชนิดของอุปกรณ์ที่ใช้ในการจัดเก็บข้อมูลไปแล้ว 2 ชนิด คือ sequential access storage device เช่น เทปแม่เหล็ก (magnetic tape) และ direct access storage device (DASD) เช่น จานแม่เหล็ก (magnetic disk) ในบทนี้เราจะกล่าวถึง โครงสร้างแฟ้มข้อมูล วิธีการเข้าถึง และการประมวลผล เมื่อข้อมูลถูกจัดเก็บในอุปกรณ์ต่างๆเหล่านี้

การจัดโครงสร้างการจัดเก็บข้อมูลบนอุปกรณ์ชนิดต่างๆสามารถทำได้ 2 วิธี คือ โครงสร้างแฟ้มข้อมูลแบบเรียงลำดับ (sequential file organization) และ โครงสร้างแฟ้มข้อมูลแบบสุ่ม (random file organization) ซึ่งบางครั้งถูกเรียกว่า วิธีการโครงสร้างแบบโดยตรง (direct) หรือ แบบไม่เรียงลำดับ (nonsequential)

ก่อนกล่าวถึงเรื่องวิธีการจัดโครงสร้างแฟ้มข้อมูล การเข้าถึงข้อมูล และการประมวลผลแฟ้มข้อมูลแบบต่างๆ จะขออธิบายนิยามของคำศัพท์ต่างๆเพื่อช่วยในการเข้าใจ

Key

ระเบียบต่างๆในแฟ้มข้อมูลจะมี identification field หรือ key เพื่อใช้ในการระบุระเบียบเพื่อใช้ในการค้นหาหรือการจัดเรียงระเบียบในแฟ้มข้อมูล เช่น แฟ้มเงินเดือนเราสามารถใส่ เลขประกันสังคมเป็น key เพื่อใช้ระบุระเบียบของพนักงานแต่ละคนในแฟ้ม

Pointer

ข้อมูลในระเบียบอาจมีรายการข้อมูลที่ทำหน้าที่เก็บตำแหน่งของระเบียบข้อมูลที่มีความสัมพันธ์กันแต่ถูกจัดเก็บคนละแฟ้มกัน ซึ่งเรียกว่า pointer field เช่น รายการข้อมูลของเงินเดือนพนักงานมีการจัดเก็บตำแหน่งข้อมูลของพนักงานอีกคนหนึ่งที่ทำงานโครงการเดียวกัน

Index

คือ รายการของ key และตำแหน่งของระเบียบของ key นั้น ซึ่งช่วยให้เราสามารถค้นหาข้อมูลบนแฟ้มข้อมูลได้รวมเร็วขึ้น ตัวอย่าง Index ของแฟ้มเงินเดือนอาจประกอบด้วย เลขประกันสังคมซึ่งเป็น key และตำแหน่งที่จัดเก็บข้อมูลของระเบียบของ key นั้น

Directory

คือ รายการของ ชื่อ และคุณลักษณะอื่นๆของแฟ้มข้อมูลและแฟ้มคำสั่งที่มีอยู่บนอุปกรณ์ที่จัดเก็บข้อมูลบางอย่าง เช่น floppy disk บนเครื่องไมโครคอมพิวเตอร์

11.3 การจัดโครงสร้างเพิ่มข้อมูลแบบเรียงลำดับและการประมวลผลการจัดโครงสร้าง

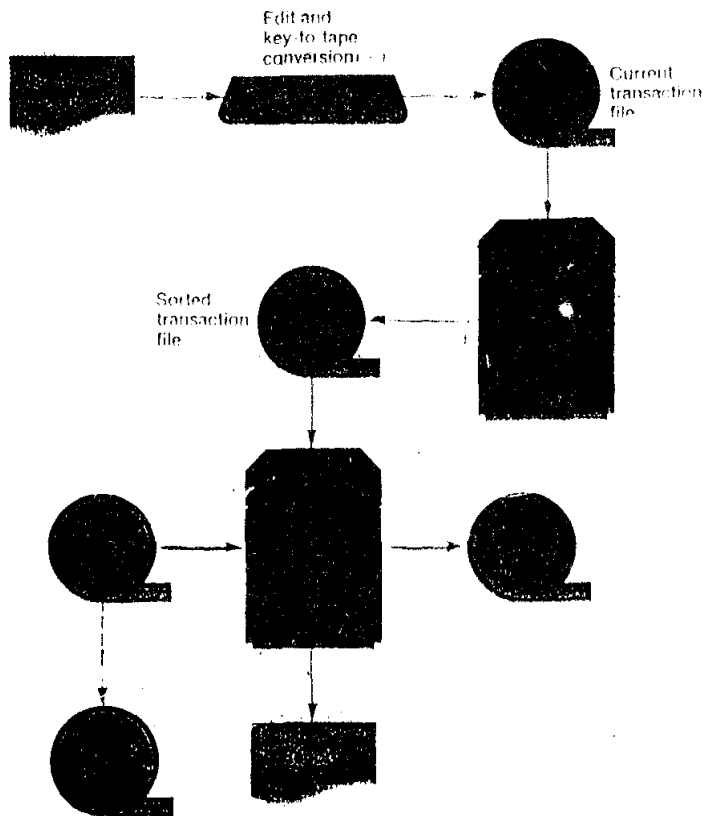
เพิ่มข้อมูลแบบเรียงลำดับ

การจัดโครงสร้างลักษณะนี้เป็นวิธีที่ง่ายและมีประสิทธิภาพสูงเมื่อนำไปใช้กับงานประมวลผลกับข้อมูลที่มีขนาดใหญ่และเป็นแบบกำหนดช่วงเวลาประมวลผล ลักษณะโครงสร้างเพิ่มข้อมูลแบบเรียงลำดับข้อมูลของระเบียนต่างๆที่ถูกจัดเก็บทางกายภาพจะเป็นแบบเรียงลำดับ โดยการจัดเรียงลำดับของระเบียนต่างๆในเพิ่มข้อมูลจะใช้ key ของระเบียนเป็นตัวจัดเรียง เช่น การจัดเก็บระเบียนของเพิ่มเงินเดือนจะถูกจัดเรียงตามลำดับของเลขประกันสังคมหรือตามชื่อของพนักงาน การประมวลผลกับโครงสร้างเพิ่มข้อมูลลักษณะนี้ข้อมูลของระเบียนทั้งหมดจะต้องถูกจัดเรียงลำดับข้อมูลก่อน

การประมวลผลเพิ่มข้อมูลที่ถูกเข้าถึงแบบเรียงลำดับ

(Sequential Access File Processing)

เมื่อเพิ่มข้อมูลที่ถูกจัดเก็บแบบเรียงลำดับถูกจัดเก็บในอุปกรณ์ เช่น เทปแม่เหล็ก ซึ่งเป็นอุปกรณ์ที่มีการเข้าถึงข้อมูลแบบเรียงลำดับแล้ววิธีเข้าถึงข้อมูลเพื่อประมวลผลจะต้องทำในลักษณะเรียงลำดับด้วย ซึ่งการประมวลผลลักษณะนี้เรียกว่า sequential access file processing



รูปที่ 11-2 แสดงตัวอย่างการประมวลผลแบบ sequential access file processing โดยข้อมูลเข้ามาจาก source documents เช่น ใบสั่งซื้อ จะถูกบันทึกลงเทปแม่เหล็กด้วยอุปกรณ์ key-to-tape ซึ่งเรียกข้อมูลที่ถูกจัดเก็บนี้เรียกว่า transaction file จากนั้นนำแฟ้มข้อมูลนี้ไปทำการจัดเรียงในลักษณะเช่นเดียวกับแฟ้มข้อมูลหลัก (master file) จากนั้นนำข้อมูล Transaction file นี้ไปทำการปรับปรุงกับแฟ้มข้อมูลหลัก

การจัดโครงสร้างแฟ้มข้อมูลแบบสุ่ม (Random File Organization)

การจัดโครงสร้างลักษณะนี้มี 3 แบบคือ direct, relative และ index file organization การจัดเก็บวิธีนี้มีแฟ้มข้อมูล 2 แฟ้ม คือ แฟ้มข้อมูลหลัก (master file) โดยระเบียบของข้อมูลที่ถูกจัดเก็บในแฟ้มนี้จะไม่มีการเรียงลำดับ และแฟ้มดัชนี (index, pointer, keys) ซึ่งประกอบด้วยค่า key และค่าตำแหน่ง (address) ของระเบียบของแฟ้มข้อมูลหลัก การเข้าถึงข้อมูลของโครงสร้างชนิดนี้ด้วยกันหลายวิธี ดังที่กล่าวข้างต่อไป

การเข้าถึงแฟ้มข้อมูลแบบสุ่ม

Key Transformation Access Method หรือ Key transformation

วิธีนี้ใช้อัลกอริทึมแบบ randomizing หรือ hashing algorithm ในการแปลงค่า key เป็นตำแหน่งที่ใช้ในการจัดเก็บ ซึ่งเราเรียกกระบวนการนี้ว่า key transformation

บางครั้งผลลัพธ์ที่ได้จากอัลกอริทึมนี้อาจได้ค่าตำแหน่งที่ตรงกันแม้ว่าค่า key จะคนละค่ากัน ซึ่งเราเรียกเหตุการณ์นี้ว่า การชนกัน (Collision) และเราเรียกค่า key แบบนี้ว่า synonyms

Index Access Method

เป็นอีกวิธีที่ใช้ในการเก็บข้อมูลและค้นหาข้อมูลแบบสุ่ม วิธีนี้มีแฟ้มข้อมูล 2 แฟ้ม คือ แฟ้มข้อมูลหลัก ซึ่งใช้เก็บข้อมูลแบบปกติ และแฟ้มข้อมูล Index ซึ่งข้อมูลในระเบียบของแฟ้มนี้ประกอบด้วย 2 ส่วน คือ ค่า key และตำแหน่งที่อยู่ของระเบียบในแฟ้มข้อมูลหลัก โดยวิธีนี้เมื่อมีการเพิ่มข้อมูลจะมีการบันทึกข้อมูลลงใน 2 ส่วนคือ แฟ้มข้อมูลหลัก และแฟ้ม Index

Record Key (employee number)	Record Address
28541	101
35879	102
47853	103
50917	104

รูปที่ 11.3 แสดงตัวอย่างของ Index

Index Sequential Access method (ISAM)

วิธีนี้จะเรียงข้อมูลจะถูกจัดเก็บอย่างเรียงลำดับในสื่อบันทึกข้อมูลที่สามารถเข้าถึงข้อมูลได้โดยตรง เช่น จานแม่เหล็ก โดยการเรียงลำดับนี้จะเรียงตามค่า key ของข้อมูล แต่อย่างไรก็ตามการจัดเก็บวิธีนี้ยังมีการใช้เพิ่มข้อมูล Index ในการจัดเก็บค่า key และตำแหน่งของระเบียบในเพิ่มข้อมูลเช่นกัน

วิธีการของ ISAM นี้ได้รวมข้อดีของการจัดโครงสร้างเพิ่มข้อมูลแบบลำดับและแบบสุ่มซึ่งโครงสร้างเพิ่มข้อมูลแบบนี้เหมาะสำหรับงานที่มีปริมาณข้อมูลมากและมีการประมวลผลแบบเรียงลำดับ เช่น ระบบประมวลผลข้อมูลแบบ Batch และถ้าบางระเบียบมีความต้องการที่จะถูกประมวลผลอย่างรวดเร็วก็สามารถทำได้โดยการใช้เพิ่ม Index ในการเข้าถึงข้อมูล

อย่างไรก็ตามแม้ว่า ISAM จะมีข้อดีแต่ก็ยังคงมีข้อเสียเช่นกันกล่าวคือ ค่าใช้จ่ายในการสร้าง การจัดเก็บข้อมูล และการบำรุงรักษาสูง และเวลาที่ใช้การเข้าถึงข้อมูลช้ากว่าการเข้าถึงข้อมูลแบบโดยตรง (Direct Organization) เพราะในการประมวลผลข้อมูลของเพิ่มข้อมูล Index ไม่ได้ถูกนำเข้าสู่หน่วยความจำหลัก แต่จะถูกจัดเก็บหน่วยความจำสำรอง

List Access Method

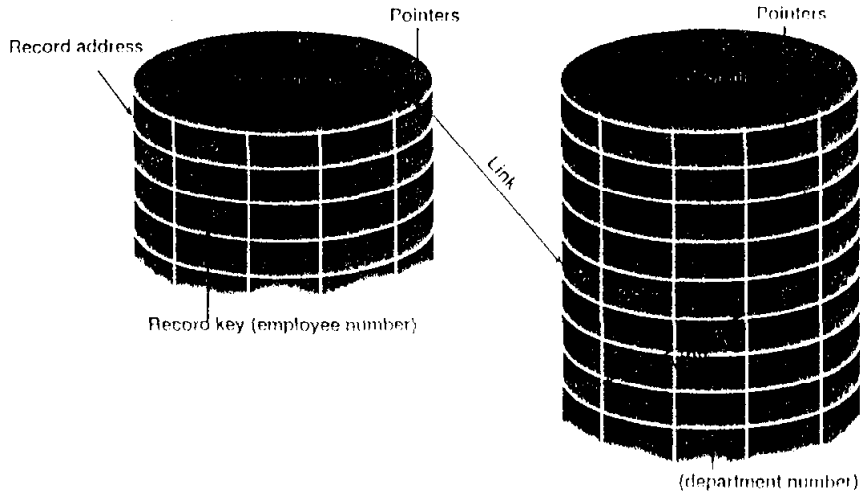
วิธีนี้มีการใช้ pointer ในการค้นหาระเบียบที่มีความสัมพันธ์กันซึ่งถูกจัดเก็บในสื่อแบบไม่เรียงลำดับ โครงสร้างข้อมูลของระเบียบที่ใช้นี้แบ่งเป็นส่วนคือ ข้อมูล (data) และ pointer ที่เก็บตำแหน่งของระเบียบที่มีความสัมพันธ์กับข้อมูล ซึ่งเรียกโครงสร้างของการจัดเก็บนี้ว่า List หรือ Link List ซึ่งโครงสร้างนี้ทำหน้าที่แสดงความสัมพันธ์ของข้อมูล ดังนั้นการค้นหาข้อมูลที่มีความสัมพันธ์กับระเบียบต่างๆจะใช้ pointer ในการค้นหา

รูปที่ 11.4 แสดงตัวอย่าง link ของระเบียบ ของเพิ่มข้อมูลบุคคล ซึ่งมีความสัมพันธ์กับเพิ่มข้อมูล payroll

โดยสรุปแล้วโครงสร้างเพิ่มข้อมูลแบบ List ช่วยให้การค้นหาข้อมูล ซึ่งมีความสัมพันธ์กันได้ง่าย ส่วนข้อเสียของโครงสร้างลักษณะนี้ คือ การที่ค้นหาข้อมูลจะช้าถ้ามีปริมาณข้อมูลที่มีความสัมพันธ์กันมาก

Inverted File Access method

วิธีจะมีเพิ่ม Index ซึ่งเรียกเพิ่ม Inverted ในการค้นหา (บางครั้งเรียกว่า Inverted Index หรือ Inverted directory) เพิ่ม Inverted จะเก็บตำแหน่งของระเบียบซึ่งมีความสัมพันธ์กัน รูปที่ 11.5 แสดงตัวอย่างเพิ่ม Inverted ที่เก็บตำแหน่งข้อมูลของอายุพนักงานที่อยู่ในช่วง 18-25 ,26-30 และ 31-35 ปี ข้อดีสามารถค้นหาข้อมูลที่มีความสัมพันธ์แบบ Multiple relationship ได้ ส่วนข้อเสียคือ ค่าใช้จ่ายในการสร้าง การจัดเก็บ การบำรุงรักษาสูง



รูปที่ 11.4 ตัวอย่างการเชื่อมกันของ linked ระเบียบของแฟ้มข้อมูลบุคคลกับ payroll

Portion of Personnel File			Inverted File by Age	
Record Address	Employee Number	Age	Age	Record Address
101	28541	43	18-25	104,
102	35879	77	26-35	102, 103.
103	47053	37	36-45	101, .
104	509 17	74		

รูปที่ 11.5 ตัวอย่างของ Inverted file

11.4 การประมวลฐานข้อมูล (DATABASE PROCESSING)

สมมุติว่าประธานบริษัทที่คุณทำงานอยู่มีความต้องการจะดูข้อมูลของพนักงาน เช่น เงินเดือน ประวัติการศึกษา การทำงาน ซึ่งข้อมูลเหล่านี้ถูกจัดเก็บคนละระบบกัน รูปที่ 11.6 แสดงโปรแกรมและเพิ่มข้อมูลเพื่อให้ได้ข้อมูลที่ต้องการ จากรูปที่จะสังเกตเห็นว่าการที่จะได้ข้อมูลในการตอบคำถามของผู้ประธานบริษัทจะต้องใช้โปรแกรมในการประมวลถึง 3 โปรแกรม ซึ่งเราเรียกการประมวลลักษณะนี้ว่า ระบบประมวลผลเพิ่มข้อมูล (file processing system) ซึ่งมีความยุ่งยากมาก และมีปัญหามากมายที่เกิดจากประมวลผลลักษณะนี้ได้แก่

Information Requested	File	Application Program
Employee salary	Payroll file	Payroll program
Educational background	Employee skills file	Skills inventory program
Salary increases and promotions	Personnel action file	Personnel action program

รูปที่ 11.6 แสดงตัวอย่างของเพิ่มข้อมูลและโปรแกรมสำหรับการประมวลพนักงาน

ความซ้ำซ้อนของข้อมูล (Data Redundacy)

จากตัวอย่างข้างต้นจะเห็นว่าโปรแกรมประยุกต์ต่างๆ ใช้เพิ่มข้อมูลของตัวเองในการประมวลผล ซึ่งข้อมูลเหล่านี้อาจมีการซ้ำกัน เช่น ชื่อ ที่อยู่ การจัดเก็บข้อมูลต้องมีความระมัดระวังในเรื่องของข้อมูลที่ไม่ตรงกัน และการทำงานเกี่ยวกับเพิ่มข้อมูลเช่น การเพิ่ม การลบ จะต้องทำทั้ง 3 เพิ่มข้อมูล

การไม่รวมกันของข้อมูล (Unintegrated Data)

การจัดเก็บข้อมูลคนละเพิ่มกันนั้นมีความยุ่งยากที่จะนำข้อมูลนั้นมาแสดงให้กับผู้ใช้ที่ดูข้อมูล ดังนั้นในการนำข้อมูลมาใช้ร่วมกันต้องมีการเขียนโปรแกรมเพื่อมาจัดการเป็นพิเศษ ซึ่งทำให้เสียค่าใช้จ่ายและเวลาในการพัฒนาโปรแกรม

การโปรแกรมและความไม่อิสระของข้อมูล (Program/Data Dependence)

การประมวลผลข้อมูลด้วยคอมพิวเตอร์ โปรแกรมแต่ละตัวจะมีเพิ่มข้อมูลเป็นของตัวเอง ดังในการใช้ข้อมูลรวมกันอาจทำไม่ได้ เพราะรูปแบบการจัดเก็บของเพิ่มข้อมูลในแต่ละโปรแกรมอาจไม่เหมือนกัน ดังนั้นถ้าหากต้องการใช้ข้อมูลรวมกันต้องมีการเขียนโปรแกรมในการแปลงรูปแบบการจัดเก็บเพื่อให้สามารถใช้กับอีกโปรแกรมหนึ่งได้

11.4 .1 หลักการของฐานข้อมูล (Database Concept)

จากปัญหาในระบบประมวลผลเพิ่มข้อมูลจึงได้เกิดแนวความคิดในการพัฒนา**ฐานข้อมูล (Database)** และการประมวลฐานข้อมูล (Database Processing)

“ฐานข้อมูล” : การรวบรวมระเบียบซึ่งไม่ซ้ำซ้อนทางตรรกะหรือเพิ่มข้อมูลที่รวบรวมระเบียบต่างๆ ในเพิ่มข้อมูลอย่างอิสระไม่ขึ้นอยู่กับโปรแกรม ด้วยเหตุนี้เองโปรแกรมต่างๆสามารถที่จะเข้าถึงข้อมูลต่างๆ ได้โดยไม่ต้องคำนึงถึงสื่อที่ใช้ในการจัดเก็บข้อมูล

“การประมวลผลฐานข้อมูล” : เป็นรูปแบบการประมวลผลสารสนเทศสมัยใหม่ ซึ่งรูปแบบการจัดการกับข้อมูลแบบ database orientation สำหรับการจัดเก็บ (storage) และการประมวลผล (processing) ข้อมูล

การทำงานกับฐานข้อมูลมีซอฟต์แวร์ที่ทำหน้าที่ในการดูแล จัดการกับฐานข้อมูล การอำนวยความสะดวกในการติดต่อกับผู้ใช้ เรียกว่า Database management system (DBMS)

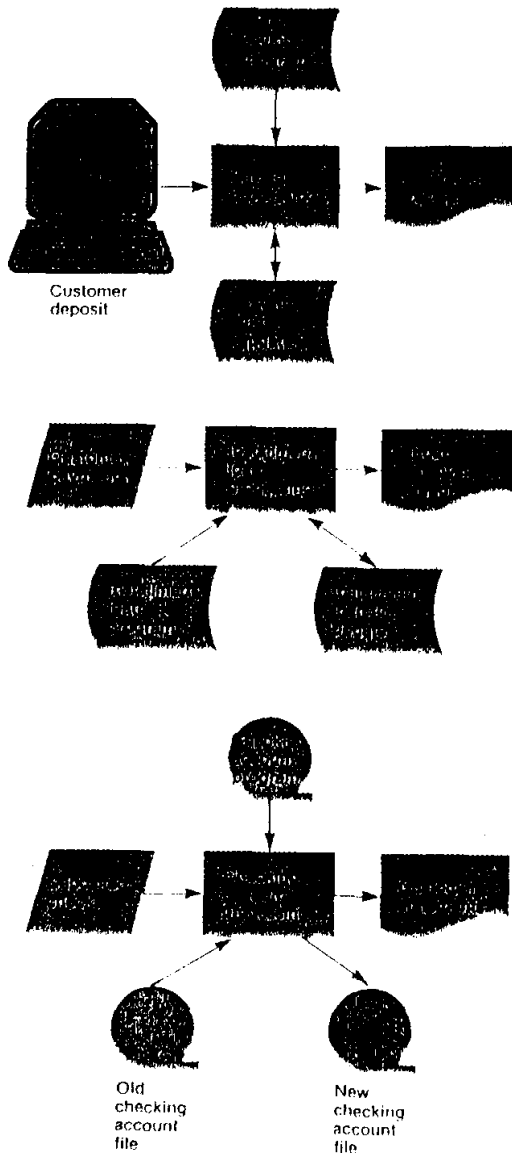
การจัดเก็บฐานข้อมูล (Database Storage)

ข้อมูลที่ถูกใช้ในการประมวลผลในโปรแกรมต่างๆจะถูกรวบรวมไว้เป็นฐานข้อมูลเดียวกัน แทนที่จะเก็บแยกไว้คนเพิ่มข้อมูลกันสำหรับการประมวลผลในแต่ละโปรแกรมประยุกต์ ตัวอย่างเช่น ระเบียบลูกค้าและข้อมูลต่างๆที่ใช้ในโปรแกรมของระบบต่างๆในธนาคาร เช่น ระบบเช็ค ระบบ ATM ระบบเครดิต จะถูกจัดเก็บในฐานข้อมูลลูกค้า

การประมวลฐานข้อมูล (Data Processing)

การประมวลผลด้วยรูปแบบการจัดการกับข้อมูลแบบ database orientation ประกอบด้วย 3 กิจกรรมหลัก

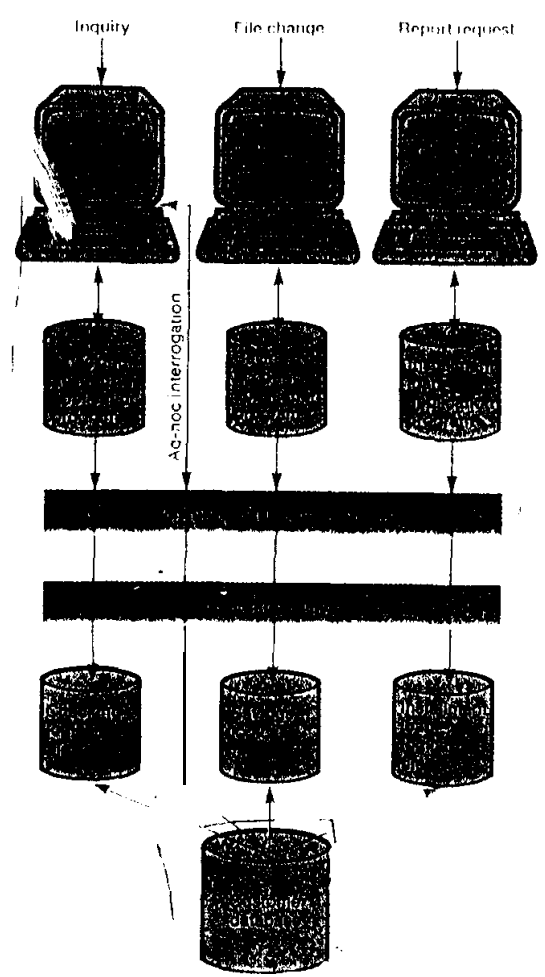
1. การปรับปรุงและการบำรุงรักษาฐานข้อมูล
2. เตรียมสารสนเทศที่จำเป็นให้กับแต่ละโปรแกรมที่ต้องใช้ข้อมูลรวมกันในฐานข้อมูล
3. เตรียมเครื่องมือในการสอบถามข้อมูลให้กับผู้ใช้



รูปที่ 11.7

รูปที่ 11.7 ตัวอย่างของระบบการประมวลผลเพิ่มข้อมูลของธนาคาร

และ

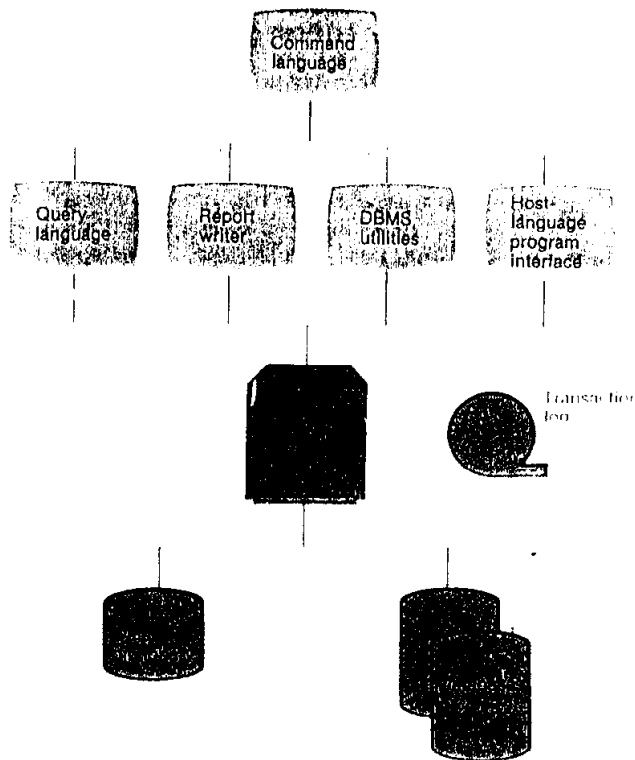


รูปที่ 11.8

รูปที่ 11.8 ตัวอย่างระบบฐานข้อมูลของธนาคาร

11.4.2 ระบบประมวลผลฐานข้อมูล (DBMS:Database Management System)

คือ ซอฟต์แวร์ทำหน้าที่ดูแลฐานข้อมูลซึ่งในระบบของ DBMS จะประกอบด้วยกลุ่มของโปรแกรมที่หน้าที่ในการสร้าง (create) บำรุงรักษา (Maintanance) การติดต่อระหว่างผู้ใช้ รูปที่ 11.9 แสดงส่วนประกอบต่างๆของ DBMS



รูปที่ 11.9 ส่วนประกอบของระบบฐานข้อมูล

ระบบ DBMS ช่วยให้ผู้ใช้สามารถทำงานต่างๆ ได้ดังนี้

- การสร้างฐานข้อมูล (Database creation) การกำหนดองค์ประกอบของข้อมูล ความสัมพันธ์ และโครงสร้างของข้อมูลที่ใช้ในการสร้างฐานข้อมูล
- การซักถามฐานข้อมูล (Data interrogation) การเข้าถึงข้อมูลในฐานข้อมูลที่ช่วยสนับสนุนการประมวลผลสารสนเทศ เช่น การสอบถามข้อมูล การสร้างรายงาน
- การบำรุงรักษาฐานข้อมูล (Data maintenance) การเพิ่ม การลบ การปรับปรุง การตรวจสอบความถูกต้องของข้อมูล การป้องกันข้อมูลในฐานข้อมูล

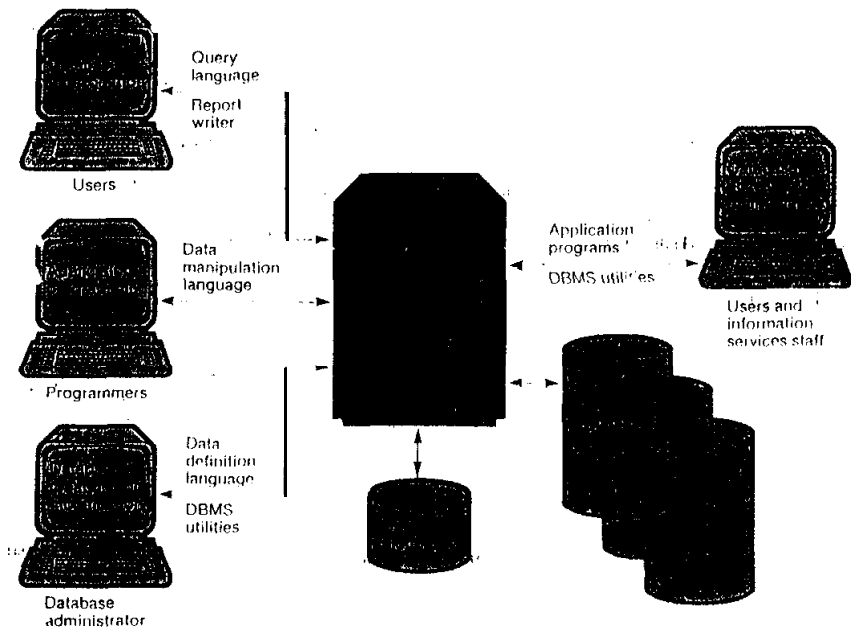
ระบบจัดการฐานข้อมูลทำหน้าที่ควบคุมการใช้ข้อมูลต่างๆในระบบทั้งหมด โดยการดำเนินงานของซอฟต์แวร์จะทำงานร่วมกับโปรแกรมควบคุมการจัดการข้อมูลของระบบปฏิบัติการ ซึ่งเป็นส่วนแรกที่ติดต่อกับหน่วยนำข้อมูลเข้า/ออก และจัดเก็บทางกายภาพในระหว่างประมวลผล

บางครั้งระบบคอมพิวเตอร์อาจมีการใช้ bank-end processor หรือ database machine ซึ่งเป็นเครื่องคอมพิวเตอร์ชนิดพิเศษที่นำมาใช้กับโปรแกรม DBMS รูปที่ 11.10 แสดงการใช้ลักษณะที่สำคัญของ DBMS

1. ผู้ใช้สอบถามข้อมูลโดยใช้ query language หรือ report writer ซึ่งมีลักษณะคล้ายคลึงกับภาษาอังกฤษ การสอบถามนี้ผู้ใช้จะได้รับผลลัพธ์ทันทีที่จอภาพ โดยที่ผู้ใช้ไม่จำเป็นต้องเขียนโปรแกรมในการสอบถามข้อมูล รูปที่แสดงการสอบถามและผลลัพธ์

2. DBMS เตรียมภาษา data manipulation language (DML) ที่ช่วยให้ผู้พัฒนาโปรแกรมสามารถติดต่อกับฐานข้อมูลได้ง่าย โดยการเขียนโปรแกรมผู้พัฒนาเพียงแต่แทรกภาษา DML เข้าไปในภาษาโปรแกรมปกติ รูปที่ 11.11 แสดงการใช้ภาษา DML กับภาษา COBOL

3. บุคคลดูแลฐานข้อมูล คือ database administrator (DBA) การจัดการกับฐานข้อมูลนี้ใช้ภาษา data definition language (DDL) ซึ่งใช้ในการกำหนดองค์ประกอบของข้อมูล กำหนดความสัมพันธ์ กำหนดโครงสร้างของฐานข้อมูลด้วย โดย DBMS จะมีการจัดเก็บข้อมูลเกี่ยวกับการจัดการข้อมูลที่ DBA กำหนดไว้ในแฟ้มข้อมูลพิเศษคือ data dictionary รูปที่ 11.12 แสดงตัวอย่างของ data dictionary รูปที่ 11.13 แสดงตัวอย่าง DBMS ที่มีใช้ในปัจจุบัน



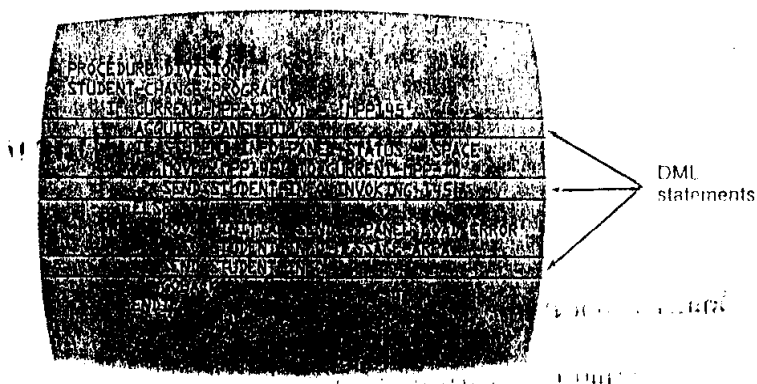
รูปที่ 11.10 แสดงการทำงานของ DBMS ที่สำคัญ

11.4.3 โครงสร้างฐานข้อมูล (Database Structures)

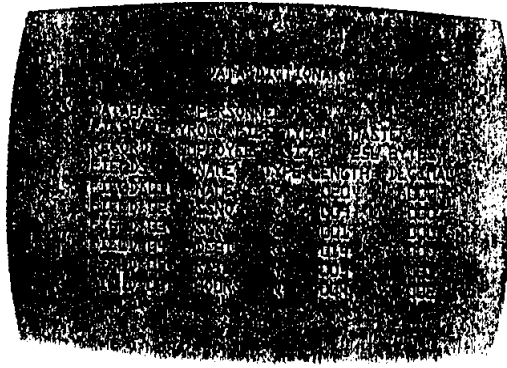
จากที่กล่าวข้างต้นฐานข้อมูลคือการรวบรวมข้อมูลของระเบียบและเพิ่มข้อมูลที่มีความสัมพันธ์กันสลับซับซ้อน ซึ่งความสัมพันธ์ระหว่างข้อมูลต่างๆในฐานข้อมูลสามารถแสดงได้ด้วยโครงสร้างข้อมูลทางตรรกะ (logical data structure) หรือตัวแบบ (model) ของฐานข้อมูล ตัวแบบฐานของที่ใช้ในการออกแบบฐานข้อมูลมี 3 ตัวแบบ คือ hierarchical ,network และ relational รูปที่ 11.14 แสดงตัวอย่างการแสดงความสัมพันธ์ของฐานข้อมูลของทั้ง 3 ตัวแบบนี้

โครงสร้างแบบลำดับชั้น (Hierarchical Structure)

ตัวแบบนี้แสดงความสัมพันธ์ทางตรรกะระหว่างข้อมูลหรือระเบียบในรูปแบบ “ลำดับชั้น (hierarchy)” หรือ ต้นไม้ (tree) รูปที่ 11.14 (รูป hierarchical structure) แสดงตัวอย่างการแสดงความสัมพันธ์ระหว่างข้อมูลของแผนก (department) ด้วยโครงสร้างแบบลำดับชั้น จากรูปเรียกที่ส่วนที่อยู่บนสุดว่า root และเรียกเส้นที่เชื่อมข้อมูลในแต่ละชั้นว่า กิ่ง (branches) ซึ่งทำหน้าที่แสดงความระหว่งข้อมูลแต่ละลำดับชั้น จากตัวอย่างรูปที่ 11.4 แสดงความสัมพันธ์ระหว่างข้อมูลของโครงการและพนักงานดังนี้ โครงการ A มีพนักงานที่ทำ 2 คน คือ Employee-1 และ Employee-2 ในทางกลับกันพนักงาน Employee-1 และ Employee-2 ก็ทำงานกับโครงการ A เพียงโครงการเดียว โดยความสัมพันธ์ลักษณะนี้เป็นแบบ one-to-many ซึ่งเป็นลักษณะความสัมพันธ์ที่โครงสร้างนี้แสดงได้ การเข้าถึงข้อมูลของโครงสร้างแบบนี้จะเริ่มจาก root แล้วไต่ลงตามกิ่งลงจนกระทั่งพบข้อมูล



รูปที่ 11.1 I การใช้ Data manipulation language



รูปที่ 11.12 ตัวอย่างของ data dictionary

Microcomputer DBMS

Condor III
dBase II and III
Knowledgeman
R.base 4000 and 6000
Revelation

Mainframe DBMS

ADABAS
DATACOM DR
IDMS, IDMS-R
IMS, SQL/DS, DB2
Model 204
RAMIS II
System 7000
TIS, TOTAL

Supplier

Condor
Ashton Tate
Micro-Data Base Systems
Microrim
Cosmos

Supplier

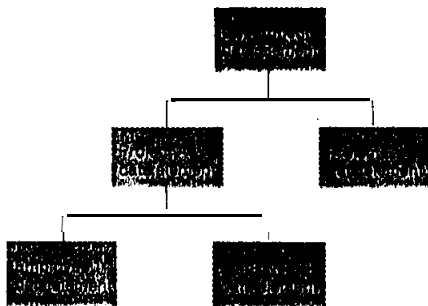
Software AG
Applied Data Research
Cullinet
IBM
Computer Corp. of America
Mathematica
INTEL
CINCOM

รูปที่ 11.13 แสดงตัวอย่างรายการผลิตภัณฑ์ของ DBMS

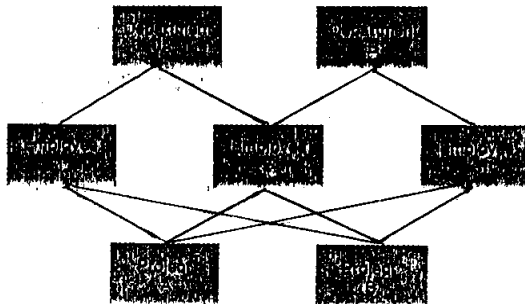
โครงสร้างเครือข่าย (Network Structure)

โครงสร้างชนิดนี้สามารถแสดงความสัมพันธ์ระหว่างข้อมูลหรือระเบียบทางตรรกะได้ สลับซับซ้อน โดยความสัมพันธ์ที่แสดงด้วยโครงสร้างนี้เป็นแบบ many-to-many รูปที่ 11.14 (Network Structure) แสดงตัวอย่างความสัมพันธ์ของข้อมูลระหว่างพนักงานกับโครงการ จากรูปจะสังเกตเห็นว่าพนักงาน Employee1 และ Employee3 ทำงานให้กับโครงการ A และ B ในทางกลับกันความสัมพันธ์ของโครงการ A และ B มีพนักงาน Employee1 กับ Employee2 ทำทั้งสองโครงการ การเข้าถึงข้อมูลสามารถทำ ณ ที่ตำแหน่งใดก็ได้ข้อมูลเนื่องจากข้อมูลของแต่ละระเบียบมีความสัมพันธ์ทั่วถึงกันหมด บางครั้งเรียกโครงสร้างนี้ว่า CODASYL model

HIERARCHICAL STRUCTURE



NETWORK STRUCTURE



RELATIONAL STRUCTURE

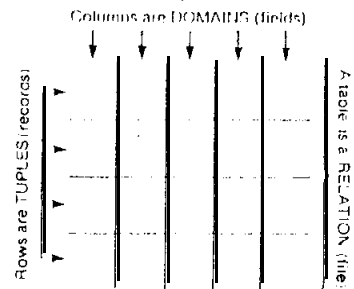
Departmental records

Dept No	D Name	D Loc	D MGR
Dept A			
Dept B			
Dept C			

Employee records

Emp No	E Name	E Title	F Salary	Dept No
Emp 1				Dept A
Emp 2				Dept B
Emp 3				Dept A
Emp 4				Dept B
Emp 5				Dept C
Emp 6				Dept B

General form of the relational database structure



รูปที่ 11.14 โครงสร้างพื้นฐานของฐานข้อมูล

โครงสร้างแบบเชิงสัมพันธ์ (Relational Structure)

ตัวแบบนี้เป็นที่นิยมใช้กันมากในปัจจุบัน โครงสร้างนี้แสดงความสัมพันธ์ระหว่างข้อมูลทางตรรกะด้วยรูปแบบของ “ตาราง (table)” โดยที่ตารางนี้เทียบเท่ากับแฟ้มข้อมูลซึ่งเรียกว่า “relations” และแต่ละแถว (row) ในตารางเรียกว่า “tuples” ซึ่งเทียบเท่ากับระเบียบในแฟ้มข้อมูล ส่วนสดมภ์ (column) ในตารางเทียบเท่ากับเขตข้อมูล (field) ซึ่งเรียกว่า domains รูปที่ 11.14 (relation) แสดงตัวอย่างการแสดงความสัมพันธ์ของข้อมูลแบบเชิงสัมพันธ์

11.4.4 การพัฒนาฐานข้อมูล (Developing a Database)

การพัฒนางานด้านฐานข้อมูลซึ่งมีขนาดใหญ่ด้วยวิธี Network และ Hierarchical เป็นงานที่ค่อนข้างสลับซับซ้อนกว่าวิธีการ relational ในทางปฏิบัติแล้วงานด้านการพัฒนาฐานข้อมูลจะมีบุคคลที่ทำหน้าที่นี้โดยตรงคือ Database Administrator (DBA) ซึ่งนอกจากทำงานด้านนี้แล้ว DBA จะทำงานร่วมกับนักวิเคราะห์ระบบ โปรแกรมเมอร์และผู้ใช้ระบบในการออกแบบ และสร้างฐานข้อมูล รูปที่ 11.15 แสดงขั้นตอนการออกแบบฐานข้อมูลทางตรรกะ

โปรแกรมจัดการฐานข้อมูล (DBMS) ได้แบ่งระดับชั้นของการมองข้อมูลในฐานข้อมูลออกเป็น 2 ระดับชั้น คือ ระดับชั้นการมองของผู้ใช้/ผู้ออกแบบ และระดับชั้นของระบบจัดการฐานข้อมูล

Logical View เป็นระดับชั้นการมองของผู้ออกแบบฐานข้อมูล ซึ่งระดับชั้นนี้แสดงให้เห็นถึงเฉพาะความสัมพันธ์ระหว่างข้อมูลโดยไม่คำนึงถึงวิธีการในการจัดเก็บข้อมูล ดังนั้นเมื่อผู้ออกแบบออกแบบฐานข้อมูลจะพิจารณาเฉพาะความสัมพันธ์ของข้อมูลเพียงอย่างเดียวเท่านั้น ซึ่งช่วยให้การออกแบบง่ายขึ้นเพราะผู้ออกแบบไม่ต้องคำนึงวิธีการจัดเก็บของข้อมูล โดยปล่อยให้เป็นที่หน้าที่ของโปรแกรมจัดการฐานข้อมูล ส่วนระดับชั้นการมองของผู้ใช้จะกล่าวในหัวข้อ Subschema

ระดับที่สองคือ Physical View บางครั้งเรียกว่า internal view เป็นระดับที่แสดงให้เห็นถึงวิธีการจัดเก็บข้อมูลต่างๆจัดเก็บด้วยวิธีใด index file หรือ inverted file ซึ่งระดับชั้นนี้โปรแกรมระบบจัดการฐานข้อมูลจะทำหน้าที่ดูแล

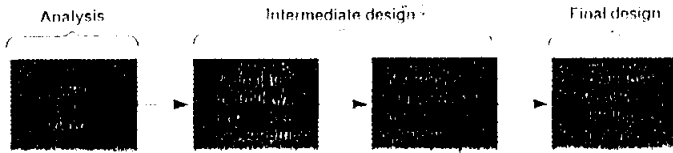
Schema

ผลลัพธ์ที่ได้จากการกำหนดความสัมพันธ์ของข้อมูลในระดับ logical view ทั้งหมดของฐานข้อมูล เรียกว่า schema

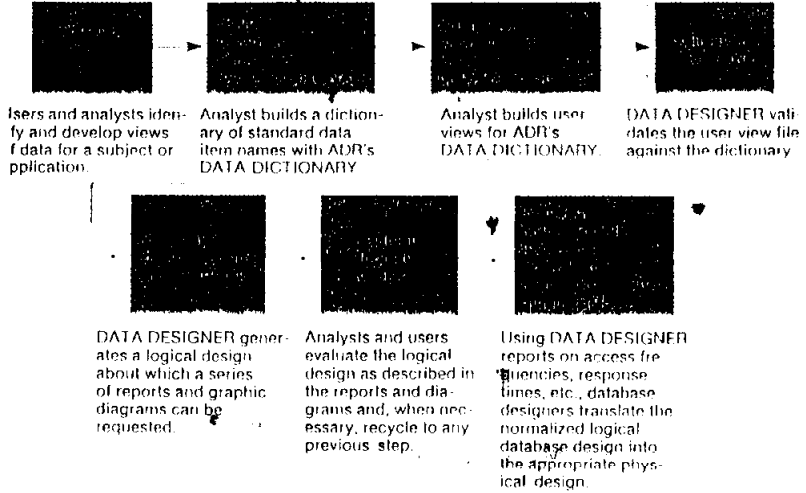
Subschema

การออกแบบฐานข้อมูลเมื่อมีการกำหนด schema ของฐานข้อมูลแล้ว DBA จึงจะกำหนด Subschema ให้กับโปรแกรมประยุกต์แต่ละตัวที่ใช้ฐานข้อมูล ซึ่ง Subschema จะแสดงความสัมพันธ์ของข้อมูลบางส่วน เฉพาะที่ใช้ในโปรแกรมประยุกต์เท่านั้น ตัวอย่างเช่น โปรแกรมทำเช็คของธนาคารซึ่งต้องใช้ข้อมูลของลูกค้าบางส่วนและเพิ่มข้อมูลที่เกี่ยวข้องเท่านั้น ดังนั้นจึงมีการกำหนด Subschema ซึ่งประกอบด้วยข้อมูลลูกค้าที่ใช้เช็คและข้อมูลที่เกี่ยวข้องให้กับโปรแกรมเท่านั้น การกำหนด Subschema ถือเป็นสร้างชั้นระดับการมองฐานข้อมูลอีกระดับหนึ่งโดยเป็นระดับชั้นการมองข้อมูลของผู้ใช้ซึ่งเรียกว่า User View

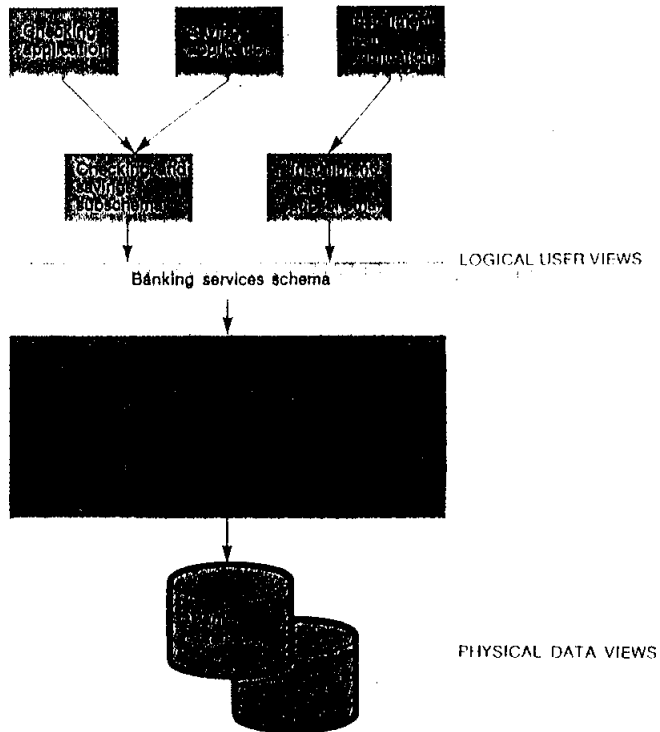
A. STEPS OF LOGICAL DATABASE DESIGN



3. COMPUTER ASSISTED DATABASE DESIGN USING DATA DICTIONARY AND DATABASE DESIGN SOFTWARE PACKAGES FROM APPLIED DATA RESEARCH (ADR)



รูปที่ 11.15 แสดงขั้นตอนในการออกแบบฐานข้อมูล



รูปที่ 11.16 แสดงความสัมพันธ์ของ subschema และ schema ในระบบฐานข้อมูล

11.5 ประเภทของฐานข้อมูล (Type of Database)

งานด้านการประมวลผลข้อมูลจริง ๆ นี้ต้องมีการใช้ฐานข้อมูลหลายประเภทร่วมกันในการทำงาน นอกจากนี้แล้วการประมวลผลข้อมูลจะอยู่ในรูปของฐานข้อมูลแบบกระจาย (distributed database) ตัวอย่างเช่น ระบบฐานข้อมูลของธนาคารต่างๆซึ่งมีหลายสาขา ดังนั้นการจัดเก็บข้อมูลอาจให้แต่ละสาขาเก็บข้อมูลของตัวเอง เป็นต้น การประมวลผลฐานข้อมูลได้แบ่งประเภทของฐานข้อมูลดังนี้

Common Operational Database ฐานข้อมูลชนิดนี้การเก็บรายละเอียดต่างๆของข้อมูลจะถูกสร้างโดยพนักงานที่ทำงานในองค์กร บางครั้งเรียกว่า transaction database หรือ production database

Common User Database ฐานชนิดนี้จะจัดเก็บเฉพาะข้อมูลและสารสนเทศที่ถูกคัดเลือกแล้วจากเจ้าหน้าที่และฐานข้อมูลจากภายนอกองค์กร ซึ่งสิ่งที่ถูกจัดเก็บทั้งหมดจะประกอบด้วยข้อมูลสรุป สารสนเทศที่จำเป็นของผู้จัดการและผู้ใช้ บางครั้งเรียกฐานข้อมูลประเภทนี้ว่า information database หรือ management database โดยปกติแล้วใช้ในระบบ Decision Support System (DSS)

Distributed Local Database ฐานข้อมูลชนิดนี้ประกอบด้วยฐานข้อมูลของสำนักงานหรือสาขาต่างๆซึ่งสามารถใช้งานร่วมกันได้ โดยฐานข้อมูลในแต่ละแห่งอาจเป็นแบบ Operation หรือ Common User Data ก็ได้

Personal User Database ฐานข้อมูลนี้ประกอบเพิ่มข้อมูลต่างๆซึ่งถูกพัฒนาขึ้นเพื่อใช้เป็นข้อมูลของตนเอง ฐานข้อมูลประเภทนี้อาจถูกสร้างมาจากโปรแกรมพวก spreadsheet หรือ โปรแกรมจัดการฐานข้อมูลบนเครื่องไมโครคอมพิวเตอร์

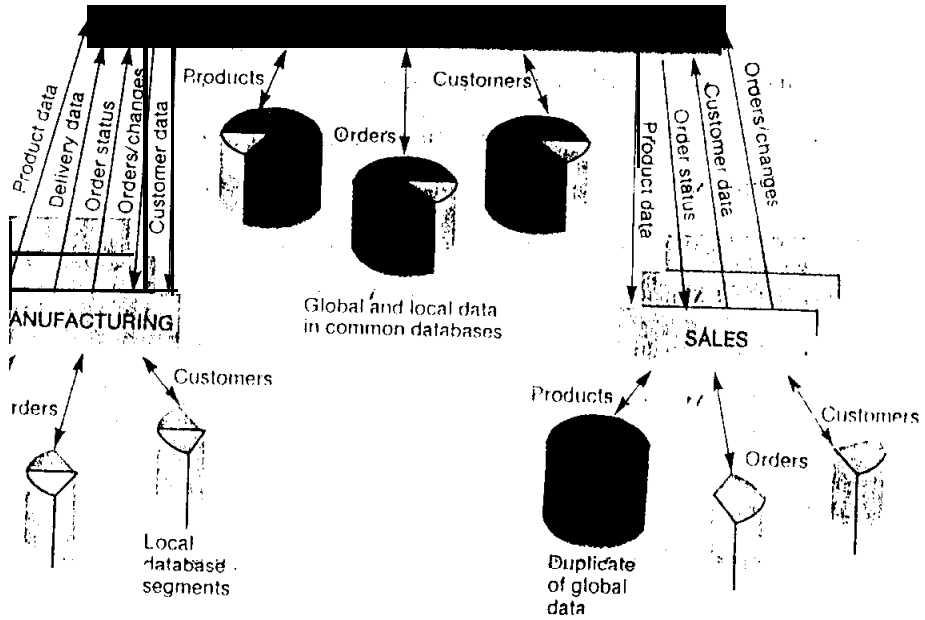
External Database ฐานข้อมูลประเภทประกอบด้วยฐานข้อมูลจากภายนอกองค์กรซึ่งเราสามารถเข้าถึงข้อมูลได้ และนอกจากนี้ฐานข้อมูลของเราก็อาจถูกเข้าถึงได้จากผู้อื่นได้ด้วย

11.6 ฐานข้อมูลแบบกระจาย (Distributed Database)

หลักการของฐานข้อมูลแบบกระจายจะถูกนำมาใช้ในงานประมวลผลแบบกระจาย (distributed processing system) การจัดเก็บข้อมูลของฐานข้อมูลประเภทนี้จะมีการกระจายข้อมูลไปยังแผนกต่างๆที่ใช้ข้อมูลในการประมวลผลซึ่งเรียกว่า local data และข้อมูลที่มีการใช้ร่วมกันระหว่างแผนกซึ่งเรียกว่า global data จะถูกจัดเก็บไว้ที่ศูนย์กลาง หรือใช้วิธีการสำเนาข้อมูลไปยังแต่ละแผนกที่ต้องการใช้ข้อมูลนี้

การประมวลผลโดยการใช้ฐานข้อมูลแบบกระจายมีทั้งข้อดีและข้อเสีย ตัวอย่างเช่น สมมุติว่าเราใช้วิธีการจัดเก็บข้อมูลแบบ global data เป็นแบบศูนย์กลาง ซึ่งจะสังเกตเห็นได้ว่าทุกๆสถานที่ที่ต้องการใช้ข้อมูลต้องมาเข้าถึงข้อมูลที่คอมพิวเตอร์ที่เป็นศูนย์กลางทั้งหมด ซึ่งจะทำให้เกิดปัญหาใน

เรื่องของความเร็วในการเข้าถึงข้อมูล ปัญหานี้อาจแก้ได้ด้วยวิธีการสำเนาข้อมูลแล้วกระจายไปยังแผนกต่างๆที่ต้องการใช้ข้อมูล แต่ก็อาจทำให้เกิดปัญหาได้ในเรื่องของความสอดคล้องของข้อมูล ซึ่งอาจไม่เหมือนกัน รูปที่ 11.17 แสดงระบบการประมวลผลฐานข้อมูลแบบกระจายศูนย์



รูปที่ 11.17 แสดงระบบการประมวลผลฐานข้อมูลแบบกระจายศูนย์