บทที่ 2
ระบบแฟ้มข้อมูล
(File System)

2.1 การจัดรูปแบบการใช้แฟ้มข้อมูล (File Organizations)

การจัดรูแบบการใช้แฟ้มข้อมูลแบ่งออกได้ 4 รูปแบบ คือ

1. **Serial File** (แฟ้มเรียง) คือ แฟ้มข้อมูลที่มีการจัดเก็บข้อมูลที่รับเข้ามาแบบมาก่อนเก็บก่อน มาหลัง เก็บหลัง จัดเก็บเรียงต่อๆ กันไป

	R7	R5	R1	R8	

- เหมาะกับการใช้เก็บข้อมูลที่มีจำนวนน้อย
- ใช้การค้นหาข้อมูลแบบ Sequential Search (การค้นหาตามลำดับ)

(*Sequential Search* คือ การค้นหาข้อมูลแบบลำดับจะต้องเริ่มค้นหาที่ Record แรกของแฟ้มข้อมูลเสมอ แล้วค้นหาเรื่อยไปจนพบ Record ที่เราต้องการ)

- มี Record Key (กุญแจหลักของระเบียน)โดย Record Key ของแฟ้มข้อมูลนี้ไม่ได้ใช้ประโยชน์ในการจัดเก็บ แต่ใช้ประโยชน์ในการค้นหา

(*Record Key* คือ Field ใด Field หนึ่ง หรือ กลุ่มของ Field ที่ใช้ในการระบุ Record ที่เราต้องการ สืบค้นแฟ้มข้อมูล)

ข้อดี จัดเก็บง่าย

ข้อเสีย ค้นหายาก เพราะในการค้นหาข้อมูลแต่ละครั้งจะต้องค้นหาจากต้น File เสมอ แล้วค้นหาไปเรื่อยๆ จนกว่าจะพบ ดังนั้น Access Time (เวลาในการค้นหาข้อมูลแต่ละตัว) จะไม่เท่ากัน โดย Access Time ของข้อมูลที่อยู่ต้น File จะน้อยกว่า Access Time ของข้อมูลที่อยู่ท้ายๆ ของ File

2. **แฟ้มข้อมูลทีละลำดับ (Sequential File)** คือ แฟ้มข้อมูลที่มีการจัดเก็บข้อมูลเรียงลำดับตาม Record Key แฟ้มข้อมูลชนิดนี้

	R3	R5	R7	R8	

- ข้อมูลที่จัดเก็บในแฟ้มข้อมูลจะต้องมีการเรียงลำดับ
- เหมาะกับข้อมูลที่มีขนาดใหญ่
- สามารถใช้การค้นหาข้อมูลได้ทั้งแบบ Sequential Search หรือ Binary Search (การค้นหาแบบทวิภาค)

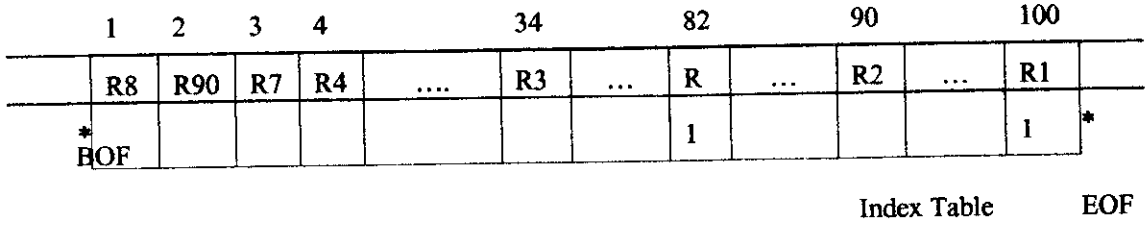
(*Binary Search* คือ การค้นหาข้อมูลที่ได้รับการเรียงลำดับเรียบร้อยแล้ว โดยจะค้นหาทีละครึ่ง แล้วตรวจสอบว่าข้อมูลที่ต้องการค้นหาอยู่นั้น อยู่ในครึ่งใด ครึ่งที่ไม่มีข้อมูลที่ต้องการจะถูกตัดออก ส่วนครึ่งที่มีก็จะถูกนำมาแบ่งครึ่งอีก แล้วทำการค้นหาเช่นนี้ไปเรื่อยๆ จนกว่าจะพบข้อมูลที่ต้องการ ซึ่งการค้นหาข้อมูลแบบนี้จะมีการค้นหาที่เร็วกว่าแบบ Sequential Search (การค้นหาตามลำดับ))

- Record Key ถูกใช้ประโยชน์ทั้งในการจัดเก็บและค้นหา

ข้อดี ค้นหาง่าย และ เร็วขึ้น เพราะใช้การค้นหาแบบ Binary ซึ่งทำให้ความแตกต่างของ Access Time ลดลง

ข้อเสีย จัดเก็บยาก เพราะต้องใช้กลไกในการเรียงลำดับข้อมูลที่จะจัดเก็บ

3. **แฟ้มข้อมูลดัชนี (Indexed – Sequential File)** คือ แฟ้มข้อมูลที่ได้นำเอาหลักการของ Serial และ Sequential File มาใช้ แฟ้มข้อมูลชนิดนี้เราจะต้องทราบจำนวนข้อมูล หรือ Record ทั้งหมดเสียก่อน และจะมีการสร้างตาราง Index (ดัชนี) ขึ้นมาซึ่งใช้เก็บข้อมูล 2 อย่าง คือ Record Key, Address โดยข้อมูลในตาราง Index จะมีการเรียงลำข้อมูลตาม Record Key การค้นหาข้อมูลจะทำการค้นหาในตาราง Index ซึ่งสามารถจะค้นหาได้ทั้งแบบ Sequential และ Binary ตาราง Index จะเป็นตารางที่ไม่กินเนื้อที่มาก



Record Key	Address
K - R8	1
K – R90	2
K – R7	3
K – R4	4
	:
K – R11	100

Record Key	Address
K – R1	82
K – R2	90
K – R3	34
K – R4	4
	:
K – R100	

ข้อดี จะเก็บและค้นหาข้อมูลง่าย เพราะได้รวมเอาข้อดีของ Serial และ Sequential File มาใช้

ข้อเสีย 1. ต้องหาเนื้อที่ในการสร้างตาราง Index ขึ้นมา

2. ต้องสร้างกลไกในการเรียงลำดับข้อมูลในตาราง Index เพื่อให้ง่ายในการค้นหาข้อมูลแบบ Binary

Serial , Sequential และ Indexed – Sequential File แฟ้มข้อมูลทั้ง 2 แบบนี้จะเป็นแฟ้มข้อมูลที่เหมาะกับการทำงานแบบ Batch (กลุ่ม)

(การทำงานแบบ Batch คือ การทำงานที่มีการรวบรวมงานไว้จำนวนหนึ่ง หรือในระยะเวลาหนึ่ง แล้วนำงานที่รวบรวมไว้นั้นไปประมวลผลในคราวเดียวกัน โดยงานที่จะนำมาใช้กับการทำงานประเภทนี้จะต้องเป็นงานที่ไม่ต้องการผลลัพธ์ในทันทีทันใด)

4. **แฟ้มข้อมูลเชิงสุ่ม (Random File (Direct File))** เริ่มขึ้นเมื่อมีการใช้ Magnetic Disk

ข้อดี *Magnetic Disk (จานแม่เหล็ก)* คือ มีการเข้าถึงข้อมูลแบบทั้ง Direct Access (เข้าถึงโดยตรง) และ Sequential Access (การเข้าถึงข้อมูลแบบ Direct Access จะดีกว่าแบบ Sequential Access) และเมื่อมีการใช้ Direct Access จึงมีการคิดค้น Random File โดยมีการคิดค้นสูตรคำนวณหา Address (เลขที่อยู่) ที่เร็วกว่า

ดังนั้น *Random File* ก็คือ แฟ้มข้อมูลที่นำเอาหลักการของ Indexed – Sequential File มาใช้ โดยจะมีการสร้างสูตรคำนวณเพื่อนำเอา Record Key มาคำนวณหา Address ซึ่ง

เรียกว่าเป็นการทำ Hashing Function (ฟังก์ชั่นแบบแฮช) แล้วจึงนำข้อมูลของ Record ดังกล่าวไปเก็บยัง Address ที่คำนวณได้

ข้อดี ของ Random File ที่เหนือกว่าแบบ Sequential คือ ไม่ว่าข้อมูลจะอยู่ ณ จุดใดบน Disk มันจะเข้าถึงได้โดยตรง ทำให้เวลาในการเข้าถึงข้อมูล (Access Time) แต่ละตัวไม่ต่างกันมากนั้น จึงทำให้เวลาในการทำงานที่เร็วมาก

ข้อเสีย คือ ถ้าการทำงานของ Hashing Function ไม่ดีพอ จะทำให้เกิดปัญหา Collision (การชนกัน) เกิดขึ้น (Collision คือ การนำเอา Record Key คนละ Key มาคำนวณหา Address แล้วได้ Address ที่เหมือนกัน หรือ เรียกได้อีกอย่างว่า การชนกันของ Address) ซึ่งมักพบว่า ยิ่งมีข้อมูลมาก ก็จะทำให้เกิด Collision มากขึ้นด้วย

การแก้ไข Collision เช่น เราอาจจะทำการสร้างเนื้อที่สำรองไว้สำหรับจัดเก็บ Record ที่เกิดการ Collision

<u>ตัวอย่าง</u> Record R7 คำนวณหา Address ได้ = 48 การเข้าถึงข้อมูลก็จะเข้าไปยัง Address 48 ต่อมา Record R92 ก็คำนวณหา Address ได้ = 48 เช่น กัน แต่ Address 48 ได้ใช้เก็บข้อมูลของ Record R7 แล้ว จึงต้องนำข้อมูลของ Record R92 ไปเก็บในเนื้อที่สำรองที่ จัดเตรียมไว้ โดยจะต้องมี Pointer (ตัวชี้) ของ Address ที่คำนวณได้ ทำหน้าที่ระบุว่า Record ที่คำนวณได้ Address ดังกล่าวที่ถูกเก็บอยู่ในเนื้อที่สำรองถูกเก็บไว้ ณ ที่ใด

แฟ้มข้อมูลชนิดนี้

- จะมีการค้นหาข้อมูลโดยใช้วิธี Hashing Function เช่นกัน โดยจะเรียกว่าเป็น Random Search (Direct Search)
- เหมาะกับการทำงานที่เป็นแบบ Interactive (เชิงโต้ตอบ) เท่านั้น

(การทำงานแบบ Interactive คือ การทำงานที่ผู้ใช้จะสื่อสารและโต้ตอบกับเครื่องคอมพิวเตอร์ได้โดยตรงผ่านทาง Terminal ดังนั้น งานที่จะใช้กับการทำงานแบบนี้จึงต้องเป็นงานที่ต้องการผลลัพธ์ที่ได้ออกมาอย่างรวดเร็วจนทำให้ผู้ใช้ไม่รู้สึกว่ามีการคอยเกิดขึ้น)

2.2 การจัดการข้อมูลในระบบแฟ้มข้อมูล (File Systems Data Management)

จะเกิดขึ้นจากการสร้างโปรแกรมประยุกต์ขึ้นมาใช้กับ File ข้อมูลนั้นๆ โดยปกติเกิดจากการใช้ภาษา 3rdGL (3rdGL คือภาษาโปรแกรมชั้นสูง (High-Level Language)มีลักษณะเป็น Procedural Oriented Language (ภาษาที่เชี่ยวชาญในการทำงาน) คือ เป็นภาษาที่เชี่ยวชาญในการทำงานโดยต้องระบุว่าอะไรคือสิ่งที่ต้องการ (What) และทำอย่างไรจึงจะได้สิ่งที่ต้องการนั้น (How) ซึ่งในการใช้ภาษา 3rdGL นี้ Programmer (นักเขียนชุดคำสั่ง) จะต้องรู้ว่าอะไรคือสิ่งที่ต้องการ และทำอย่างไรจึงจะได้ในสิ่งที่ตนต้องการนั้น) เมื่อนำเข้ามาใช้ในการพัฒนาระบบ File Programmer จะต้องทราบถึงโครงสร้างของข้อมูล เส้นทางการเข้าถึงข้อมูล รวมทั้งรูปแบบข้อมูลที่จัดเก็บจริงในหน่วยความจำด้วย ส่วน 4thGL หรือ ที่เรียกว่า Problem Oriented Language คือ ภาษาที่เชี่ยวชาญในปัญหาด้านใดด้านหนึ่ง ซึ่งในการใช้ 4thGL นี้ Programmer จะต้องรู้เพียงว่าอะไรคือสิ่งที่ต้องการ (What) เท่านั้น (ระบุ What เท่านั้น) ด้วยเหตุนี้ การจัดเก็บข้อมูล การค้นหาข้อมูล และการแก้ไขข้อมูล ใน 3rdGL จึงยุ่งยาก

การจัดการข้อมูล (Data Management) เกี่ยวกับการรวมข้อมูล (Data Collection), การจัดเก็บข้อมูล (Data Storage), การสืบค้นข้อมูล (Data Retrieval) ซึ่งทำให้เราสามารถเข้าถึง (Access) ข้อมูลได้อย่างมีประสิทธิภาพ และผลที่ตามมาก็คือทำให้เราได้ Information ที่ต้องการอย่างรวดเร็ว

การทำให้ Data Management มีประสิทธิภาพได้ก็โดยการใช้ Computer Database Systems เข้ามาช่วยซึ่งประกอบด้วย

- ฐานข้อมูล (Database :DB)
- ระบบการจัดการฐานข้อมูล (Database Management System :DBMS)

การใช้วิธีการจัดการฐานข้อมูลแทนที่การจัดการแฟ้มข้อมูลมีประเด็นที่สำคัญคือ

1. DBMS ทำให้การจัดการข้อมูลสะดวกและมีประสิทธิภาพขึ้น

CT 316 (S) 9

2. DBMS ประกอบด้วย Query Language (ภาษาสอบถาม) ซึ่งทำให้เกิดการสร้างคำตอบที่ต้องการแก่ผู้ใช้ในกรณีที่มี Ad Hoc Queries.(คำถามตามต้องการ)
3. DBMS ทำให้ผู้ใช้มีสภาพแวดล้อมในการใช้ข้อมูลที่ดีขึ้น เช่น เรียกใช้ข้อมูลได้ดี และสะดวกขึ้น รวมทั้งทำให้เกิด Information (สารสนเทศ) ได้เร็วขึ้น ซึ่งเหมาะกับการนำไปใช้ในการดำเนินธุรกิจ
4. การเข้าถึงข้อมูลที่ดีขึ้นในระบบ ฐานข้อมูล ทำให้การทำงานขององค์กรดีขึ้น โดยมีความรู้เกี่ยวกับการทำงานของส่วนต่างๆ ในองค์กรได้ดีขึ้น, สัมพันธ์กันขึ้น นั่นเอง
5. ทำให้ปัญหาเรื่อง Data Inconsistency (ความไม่มั่นคงของข้อมูล) น้อยลง

โดยปกติ Program ที่พัฒนาขึ้นมาเพื่อจัดการแฟ้มข้อมูล จะมีการทำงาน 5 อย่างด้วยกัน

1. การสร้างโครงสร้าง File (Create the File Structure)
2. ใส่ข้อมูลเข้าไปใน File (Add Data to the File)
3. การลบข้อมูลออกจากFile (Delete Data From the File)
4. การปรับเปลี่ยนข้อมูลใน File(Modify the Data in the File)
5. การแสดงข้อมูลที่อยู่ใน File ออกมาดู(List the File Contents)

ข้อเสีย ของ File System

1. ปัญหา Data Redundancy (การเกิดการซ้ำกันของข้อมูล) คือ การเกิดการซ้ำกันของข้อมูล เนื่องจากในการใช้งานแฟ้มข้อมูลนั้นหากในแต่ละหน่วยงานย่อยในองค์กร ต้องการใช้ข้อมูลชุดเดียวกันก็จะต้อง Copy ข้อมูลดังกล่าวนั้นไปเก็บไว้ใช้เป็นของตัวเอง ทำให้มีข้อมูลชุดเดียวกันซ้ำซ้อนกันหลายที่ และด้วยเหตุที่เกิดการซ้ำกันของข้อมูลนี้ก็จะทำให้เกิด
   - Data Inconsistency (ความไม่มั่นคงของข้อมูล) คือ ในการใช้งานข้อมูลที่ซ้ำซ้อนกันหลายที่ หากข้อมูลที่ถูกเก็บไว้ ณ ที่หนึ่งถูกเปลี่ยนแปลง แต่ข้อมูลชุดเดียวกันนี้ที่ถูกเก็บไว้ ณ ที่อื่นๆ ไม่ได้รับการเปลี่ยนแปลงตามไปด้วย ก็จะทำให้ข้อมูลที่ถูกเก็บไว้ในแต่ละที่นั้นไม่ถูกต้องตรงกัน ซึ่งทำให้เกิดความไม่มั่นคงของข้อมูล
   - Data Anormalies (ข้อมูลไม่ปกติ) คือ ในการที่ข้อมูลชุดเดียวกันที่ถูกเก็บไว้ในแต่ละที่ไม่ถูกต้องตรงกันนั้นบอกให้รู้ว่า ได้เกิดความไม่ปกติกับข้อมูลเกิดขึ้นซึ่งปัญหา Data Redundancy นี้ เป็นปัญหาที่สำคัญที่สุด ในระบบแฟ้มข้อมูล

2. การเขียนโปรแกรมยุ่งยากซับซ้อน เนื่องจาก
   - Programmer จะต้องทราบ Physical Data Structure (โครงสร้างข้อมูลทางกายภาพ) และ Logical Data Structure (โครงสร้างข้อมูลทางตรรก)
   - มีลักษณะของ Program Dependency (โปรแกรมที่น่าเชื่อถือ) คือในการสร้าง File แต่ละ File นั้น File 1 File จะเกิดขึ้นจากโปรแกรม 1 โปรแกรมเท่านั้น ถ้าหากต้องการสร้าง File ขึ้นมา 100 Files ก็ต้องสร้างโปรแกรมขึ้นมา 100 โปรแกรมด้วย
   - การเกิด Structural Dependency (โครงสร้างที่น่าเชื่อถือ) คือ การเปลี่ยนโครงสร้างของ Record เช่น เพิ่ม Field จะต้องไปเปลี่ยนในโปรแกรม ซึ่งจะมีผลต่อ File
   - การเกิด Data Dependency (ข้อมูลที่น่าเชื่อถือ) คือการเปลี่ยนโครงสร้างข้อมูลเช่น การเปลี่ยนข้อมูล ใน Field จะต้องไปเปลี่ยนในโปรแกรม ซึ่งจะมีผลต่อ File เช่นกัน
3. ไม่มีระบบป้องกันข้อมูล
4. ไม่มีการแบ่งบันทรัพยากรข้อมูล (not share)

จากข้อเสียทั้งหมดนี้ จึงได้นำระบบฐานข้อมูล (Database Systems) มาใช้แทนระบบแฟ้มข้อมูล (File Systems)