



### บทที่ 3

#### การออกแบบระบบ (System Design)

ระบบการจัดการข้อมูล (Data Management System หรือ DMSYS)<sup>1</sup> เป็นโปรแกรมสำเร็จรูป ที่ใช้ในการสร้างแฟ้มข้อมูล บันทึกข้อมูล เปลี่ยนแปลงแก้ไขข้อมูลและโครงสร้าง พิมพ์ข้อมูล และจัดโครงสร้างของข้อมูลใหม่ (reorganization) โดยผู้ใช้ติดต่อกับโปรแกรมนี้ด้วยคำสั่ง ซึ่งออกแบบสำหรับระบบนี้โดยเฉพาะ ทำให้ผู้ใช้ได้รับประโยชน์จากการใช้โปรแกรมดังต่อไปนี้

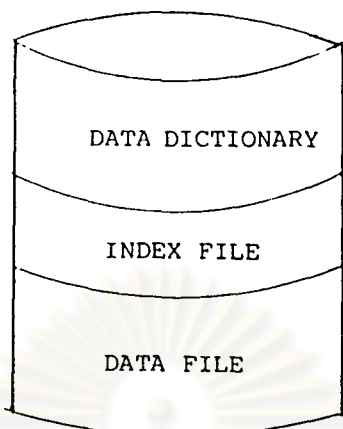
1. สร้างแฟ้มข้อมูล บันทึกข้อมูล และจัดการกับข้อมูลในแฟ้มข้อมูลต่าง ๆ ได้ โดยไม่ต้องเขียนโปรแกรม
2. สามารถเปลี่ยนแปลงโครงสร้างของแฟ้มข้อมูล เช่น การเพิ่มหรือลบเขตข้อมูลในแฟ้มข้อมูล โดยไม่ต้องสร้างแฟ้มข้อมูลใหม่
3. สามารถเข้าถึงข้อมูลได้ ทั้งแบบเรียงลำดับ (sequential) และแบบโดยตรง (Direct)

#### 3.1 โครงสร้างของแฟ้มข้อมูลในดีเอ็มซิส

ข้อมูลของ ดีเอ็มซิส จัดเก็บอยู่ในแฟ้มข้อมูลแบบ วีแซม เคเอสซีเอส ซึ่งแบ่งออกเป็น 3 แฟ้มข้อมูล ดังแสดงในรูปที่ 3.1 พื้นที่ของแฟ้มข้อมูลทั้ง 3 นี้ จะบรรยายไว้ใน วีแซม แคตาล็อก (VSAM Catalog) โดยโปรแกรมบริการของวีแซม ชื่อ "IDCAMS"

3.1.1 ปทานุกรมข้อมูล (Data Dictionary) เป็นแฟ้มข้อมูลที่ใช้บันทึกลักษณะโครงสร้างของแฟ้มข้อมูลต่าง ๆ ได้แก่ ลักษณะของเขตข้อมูล โครงสร้างของแฟ้มข้อมูล และค่าสถิติที่จำเป็นบางอย่าง เช่น จำนวนระเบียบข้อมูลภายในแฟ้มข้อมูล จำนวนระเบียบข้อมูลที่ลบออกจากแฟ้มข้อมูล เป็นต้น ปทานุกรมข้อมูลประกอบด้วยระเบียบข้อมูล 4 ประเภท (record type) คือ

<sup>1</sup> DMSYS เป็นชื่อย่อของระบบการจัดการข้อมูล ที่ใช้ชื่อนี้เพราะไม่ต้องการให้ไปเข้ากับ DMS หรือ Data Management System ของระบบ IBM



รูปที่ 3.1 แสดงแฟ้มข้อมูลของ คีเอ็มซิส

3.1.1.1 ระเบียบข้อมูล เอ (record Type A or RTA) เป็น ระเบียบข้อมูลแรกที่ใช้เก็บข้อมูลในการควบคุมเกี่ยวกับจำนวนแฟ้มข้อมูล และจำนวนเขตข้อมูล ทั้งหมดที่สร้างขึ้น ประกอบด้วยเขตข้อมูลต่าง ๆ ดังแสดงในรูปที่ 3.2

T	R	NFI	NFD
---	---	-----	-----

รูปที่ 3.2 แสดงโครงสร้างของระเบียบข้อมูล เอ

- 1) T เป็นรหัสประเภทของระเบียบข้อมูล มีค่าเป็น 'A'
- 2) R สำรอง 4 ไบต์ มีค่าเป็น '0000'
- 3) NFI เป็นจำนวนของแฟ้มข้อมูล ที่สร้างขึ้นในระบบ
- 4) NFD เป็นจำนวนเขตข้อมูล

ในการเริ่มต้นระบบใหม่ ระเบียบข้อมูลนี้จะบันทึกค่า 'A000000010001'

### 3.1.1.2 ระเบียบข้อมูล บี (Record Type B or RTB)

เป็นระเบียบที่ใช้บันทึกลักษณะของเขตข้อมูล ที่บันทึกอยู่ในแฟ้มข้อมูล เป็นระเบียบข้อมูลที่มีความยาวคงที่ ประกอบด้วยเขตข้อมูลต่าง ๆ ดังแสดงในรูปที่ 3.3

T	FDN	D	FD NAME	FDL	TY
---	-----	---	---------	-----	----

รูปที่ 3.3 แสดงโครงสร้างของระเบียบข้อมูล บี

- 1) T เป็นรหัสประเภทของระเบียบข้อมูล มีค่าเป็น 'B'
- 2) FDN เป็นหมายเลขของเขตข้อมูล ซึ่งควบคุมด้วย NFD จากระเบียบข้อมูลแรก มีค่าตั้งแต่ 0001, 0002, ... 9999
- 3) D เป็นไบต์ที่แสดงสถานะของระเบียบเขตข้อมูล มี 2 สถานะ คือ
  - ' ' (ว่าง) แสดงว่ายังใช้ระเบียบนี้อยู่
  - '\*' แสดงว่าระเบียบนี้ถูกลบออกไปแล้ว
- 4) DF NAME เป็นชื่อของเขตข้อมูล มีความยาวไม่เกิน 8 ไบต์ ถ้าความยาวไม่ถึง 8 ไบต์ จะเติมว่างข้างหลัง
- 5) FDL เป็นความยาวของเขตข้อมูล
- 6) TY เป็นชนิดของข้อมูลในเขตข้อมูลนั้น ซึ่งมีค่าได้ดังนี้
  - 'N' หมายถึง ค่าตัวเลข
  - 'C' หมายถึง ตัวอักษร

### 3.1.1.3 ระเบียบข้อมูล ซี (Record Type C or RTC) เป็น

ระเบียบข้อมูลที่ใช้บันทึกข้อมูลที่ใช้ในการควบคุมแฟ้มข้อมูลต่าง ๆ ซึ่งผู้ใช้สร้างขึ้น ประกอบด้วยเขตข้อมูลต่าง ๆ ดังแสดงในรูปที่ 3.4

T	FID	D	FILE NAME	KF 1	KF 2	KF 3	RES	NFD	RL	NR	NDR
---	-----	---	-----------	------	------	------	-----	-----	----	----	-----

### รูปที่ 3.4 แสดงโครงสร้างของระเบียบข้อมูล ซี

- 1) T เป็นรหัสประเภทของระเบียบข้อมูล มีค่าเป็น 'C'
- 2) FID เป็นหมายเลขประจำแฟ้มข้อมูล
- 3) D เป็นไบต์แสดงถึงสถานะของแฟ้มข้อมูล โดยมีได้

2 สถานะ คือ

- ' ' (ว่าง) แสดงว่ายังใช้แฟ้มข้อมูลนี้อยู่
- '\*' แสดงว่าแฟ้มข้อมูลนี้ถูกลบออกไปแล้ว

- 4) FILE NAME เป็นชื่อของแฟ้มข้อมูล
- 5) KF1 - KF3 เป็นหมายเลขเขตข้อมูลที่ถูกระบุให้ใช้เป็น

คีย์ ของแฟ้มข้อมูลนี้ ถ้าไม่มีการกำหนดคีย์ เขตข้อมูลจะเป็นว่างหรือมีค่าเป็น '40' เลขฐานสิบหก

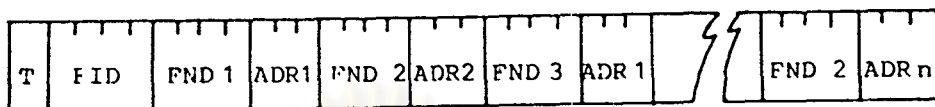
- 6) RES สำรอง ว่างไว้ 4 ไบต์
- 7) NFD เป็นจำนวนของเขตข้อมูลที่ประกอบกันเป็นระเบียบ

ข้อมูลของแฟ้มข้อมูล

- 8) RL เป็นความยาวของระเบียบข้อมูล
- 9) NR เป็นจำนวนระเบียบข้อมูลที่บันทึกในแฟ้มข้อมูล
- 10) NDR เป็นจำนวนระเบียบข้อมูลที่ถูกลบออกจากแฟ้มข้อมูล

#### 3.1.1.4 ระเบียบข้อมูล ดี (Record Type D or RTD) ใช้เก็บ

โครงสร้างของแฟ้มข้อมูล เป็นระเบียบข้อมูลที่มีความยาวไม่คงที่ ขึ้นอยู่กับจำนวนเขตข้อมูลในแต่ละแฟ้มข้อมูล ประกอบด้วยเขตข้อมูลต่าง ๆ ดังแสดงในรูปที่ 3.5



รูปที่ 3.5 แสดงโครงสร้างของระเบียบข้อมูล คือ

- 1) T เป็นรหัสประเภทของระเบียบข้อมูล มีค่าเป็น 'D'
- 2) FID เป็นหมายเลขแฟ้มข้อมูล
- 3) FND 1 และ ADR 2 เป็นหมายเลขเขตข้อมูล และตำแหน่งของเขตข้อมูลในระเบียบข้อมูล ตามลำดับ
- 4) FND 2 และ ADR 2 เป็นหมายเลขและตำแหน่งของเขตข้อมูลที่ 2 ในระเบียบข้อมูล
- 5) FND n และ ADR n เป็นหมายเลขและตำแหน่งของเขตข้อมูลที่ n ในระเบียบข้อมูล

ระเบียบข้อมูลทั้ง 4 ประเภท บันทึกอยู่ในแฟ้มข้อมูลปทานุกรม ซึ่งเก็บรายละเอียดของข้อมูลทั้งหมดที่บันทึกอยู่ในแฟ้มข้อมูล ปทานุกรมข้อมูลเป็นแฟ้มข้อมูล วีแซม เคเอสดีเอส ข้อมูลจะถูกบันทึกแบบเรียงลำดับตามคีย์ ซึ่งกำหนดให้อยู่ใน ไบต์ที่ 1-5 ของระเบียบข้อมูลทั้ง 4 ประเภท

3.1.2- แฟ้มข้อมูล (Data File) เป็นส่วนที่ใช้เก็บข้อมูลจริง ระเบียบข้อมูลของแฟ้มต่าง ๆ ตามที่บรรยายไว้ในปทานุกรมข้อมูล จะถูกนำมาบันทึกไว้ในแฟ้มข้อมูลนี้ทั้งสิ้น โครงสร้างของระเบียบข้อมูลประกอบด้วย 2 ส่วน คือ ส่วนนำ(prefix) และข้อมูล(data) ในส่วนนำประกอบด้วย 3 เขตข้อมูล ดังแสดงในรูปที่ 3.6 คือ

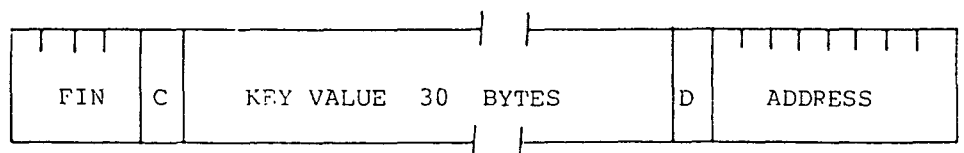


รูปที่ 3.6 แสดงโครงสร้างของระเบียบข้อมูลในแฟ้มข้อมูล

- 1) FIN เป็นหมายเลขแฟ้มข้อมูล
- 2) RN เป็นหมายเลขระเบียบข้อมูลภายในแฟ้มข้อมูล
- 3) D เป็นสถานะของระเบียบข้อมูล มี 2 สถานะ คือ
  - ' ' หมายถึง ยังมีระเบียบข้อมูลอยู่ในแฟ้มข้อมูล
  - '\*' หมายถึง ได้ลบระเบียบข้อมูลนี้ออกจากแฟ้มข้อมูลแล้ว

จากเขตข้อมูลที่ 1 และที่ 2 ทำให้ทราบว่าระเบียบข้อมูลนั้น มาจากแฟ้มข้อมูลใด และเป็นระเบียบที่เท่าไร เมื่อรวม 2 เขตข้อมูลนี้เข้าด้วยกันแล้วจะมีค่าไม่ซ้ำกันเลย ในทุกระเบียบข้อมูลภายในแฟ้มข้อมูลนี้ และวีแชน จะนำเอา 2 เขตข้อมูลนี้เป็นคีย์ ไปสร้างเป็นดัชนีของ วีแชน ข้อมูลจะถูกบันทึกเรียงลำดับตามคีย์

3.1.3 แฟ้มดัชนี (Index File) เป็นแฟ้มข้อมูลที่เกิดจากการที่ผู้ใช้กำหนดให้เขตข้อมูลใด ๆ ตั้งแต่ 1-3 เขตข้อมูลมารวมกันเป็นคีย์ของระเบียบข้อมูล โดยที่ผลรวมของความยาวของเขตข้อมูลยาวไม่เกิน 30 ไบต์ ดีเอ็มซิส จะนำเอาคีย์ที่ผู้ใช้กำหนด และที่อยู่(Address) ของระเบียบข้อมูลนั้นไปสร้างแฟ้มดัชนีขึ้น เพื่อใช้ในการค้นหาระเบียบข้อมูลในภายหลัง โครงสร้างของระเบียบข้อมูลในแฟ้มดัชนี ประกอบด้วยเขตข้อมูลต่าง ๆ ดังแสดงในรูปที่ 3.7



รูปที่ 3.7 แสดงโครงสร้างของระเบียบข้อมูลในแฟ้มดัชนี

- 1) FIN เป็นหมายเลขแฟ้มข้อมูล
- 2) C เป็นรหัสแสดงว่าเป็นคีย์ของระเบียบข้อมูล ซึ่งมีค่า = 1
- 3) KEY VALVE เป็นค่าของข้อมูลที่เป็นคีย์ทั้ง 3 เขตข้อมูลต่อกัน

ถ้าความยาวไม่ถึง 30 ไบต์ จะเติมว่างไว้ข้างหลังจนครบ

- 4) D เป็นสถานะของระเบียบข้อมูลในแฟ้มคีย์นี้ มีได้ 2

สถานะ คือ

- ' ' (ว่าง) หมายถึงยังมีระเบียบข้อมูลอยู่

- '\*' หมายถึงลบระเบียบข้อมูลนั้นออกแล้ว

5) ADDRESS เป็นที่อยู่ของระเบียบข้อมูลในแฟ้มข้อมูล ให้มาจากคีย์ของระเบียบข้อมูลในแฟ้มข้อมูล (ไบต์ที่ 1-8 ของระเบียบ จากข้อ 3.1.2) ซึ่งระบบสร้างขึ้นในขณะที่มีการบันทึกข้อมูล

ในคีย์ของแต่ละแฟ้มข้อมูล จะมีระเบียบแรกใช้สำหรับเป็นตัวบอกจุดเริ่มต้นของแฟ้มข้อมูล มีโครงสร้างเหมือนกับรูปที่ 3.8 ต่างกันเฉพาะในเขตข้อมูล KEY VALVE ซึ่งจะมีค่าเป็น 'A...' แทน จำนวนอักษร A จะเท่ากับความยาวของคีย์ของแฟ้มข้อมูลนั้น

### 3.2 ส่วนประกอบของระบบ

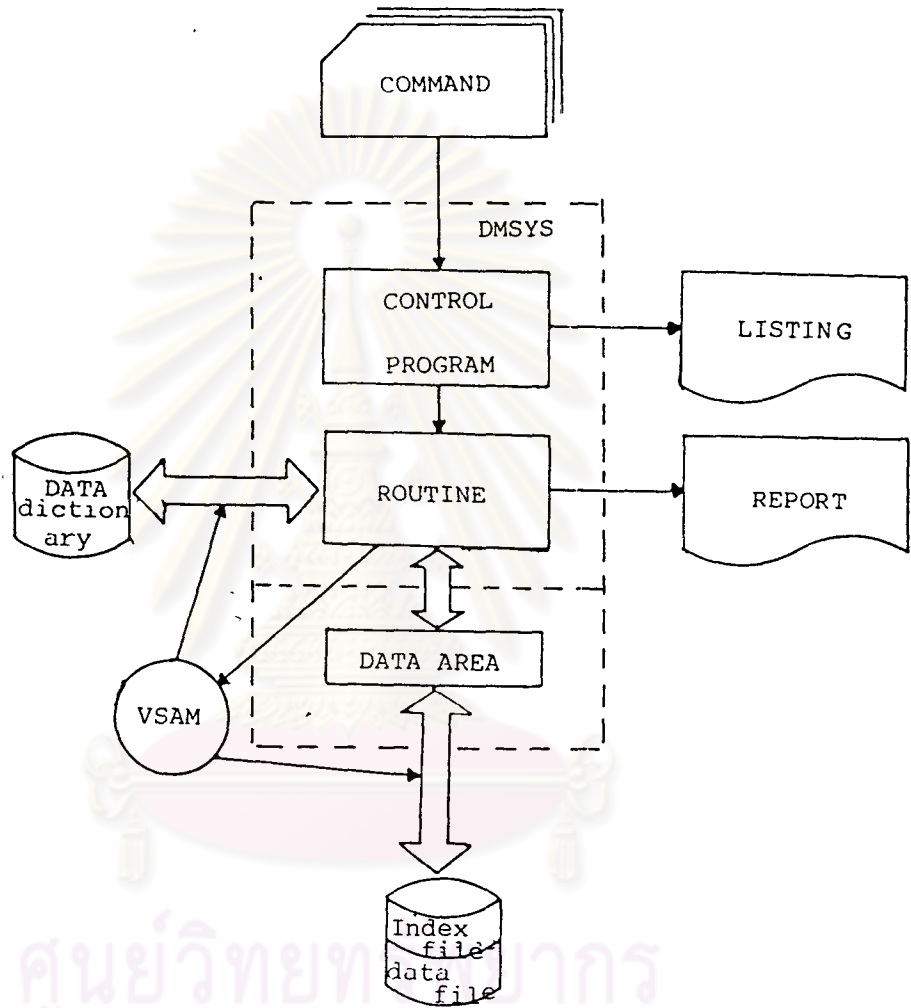
คือเอ็มซิส ประกอบด้วยส่วนต่าง ๆ ดังแสดงในรูปที่ 3.8

3.2.1 คำสั่ง (Command) เป็นภาษาซึ่งใช้ในการติดต่อสั่งการ ให้ระบบทำงานตามต้องการ ซึ่งจะได้กล่าวถึงโดยละเอียดในภายหลัง

3.2.2 คือเอ็มซิส เป็นโปรแกรม ซึ่งทำงานตามคำสั่ง แบ่งออกเป็น 3 ส่วนใหญ่ ๆ คือ

3.2.2.1 โปรแกรมควบคุม (control program) ทำหน้าที่อ่านคำสั่ง

ตรวจสอบความถูกต้อง แปลคำสั่ง และจ่ายงานให้โปรแกรมย่อยต่าง ๆ ทำงาน และหยุดการทำงาน ของระบบ พร้อมกับพิมพ์คำสั่งออกมาตามที่ใช้เขียน



รูปที่ 3.8 แสดงความสัมพันธ์ขององค์ประกอบต่าง ๆ ของระบบ



3.2.2.2 โปรแกรมทำงาน(working routine) ประกอบด้วย โปรแกรมย่อยต่าง ๆ ดังนี้

- 1) โปรแกรมย่อยบรรยายลักษณะของข้อมูล โปรแกรมนี้ทำหน้าที่ บันทึกลักษณะของข้อมูลลงในปทานุกรมข้อมูล
- 2) โปรแกรมบันทึกข้อมูล ทำหน้าที่บันทึกข้อมูลลงในแฟ้มข้อมูล และแฟ้มดัชนี ตามข้อกำหนดในปทานุกรมข้อมูล
- 3) โปรแกรมทำข้อมูลให้ทันสมัย(update routine) ทำหน้าที่ค้นหาระเบียบข้อมูลที่ต้องการ และเปลี่ยนแปลงข้อมูลในเขตข้อมูล ลบระเบียบข้อมูล เปลี่ยนโครงสร้างแฟ้มข้อมูล เปลี่ยนชื่อเขตข้อมูล ชื่อแฟ้มข้อมูล
- 4) โปรแกรมแสดงผล ทำหน้าที่พิมพ์ระเบียบข้อมูลหรือบางเขตข้อมูล พิมพ์โครงสร้างของแฟ้มข้อมูล และพิมพ์ข่าวสารแจ้งผลการปฏิบัติงานของโปรแกรม
- 5) โปรแกรมจัดโครงสร้างของระบบข้อมูลใหม่ (reorganization routine) ทำหน้าที่จัดโครงสร้างของข้อมูล ซึ่งมีการเปลี่ยนแปลงขึ้นใหม่ เพื่อลบขยะ(garbage) ออกจากระบบข้อมูล

3.2.2.3 พื้นที่ข้อมูล (data area) เป็นพื้นที่ซึ่งใช้สำหรับเก็บข้อมูล ที่อ่านมาจากแฟ้มข้อมูลต่าง ๆ เพื่อให้โปรแกรมประมวลผลข้อมูลตามคำสั่งต่อไป และเป็นพื้นที่ซึ่งใช้เก็บข้อมูล เพื่อบันทึกลงในแฟ้มข้อมูล

3.2.3 วีแซม เป็นโปรแกรมที่ใช้ในการเข้าถึงข้อมูลของระบบเครื่อง ไอบีเอ็ม เมื่อมีการอ่านหรือการบันทึกข้อมูลเกิดขึ้น โปรแกรม ดีเอ็มซิส จะส่งให้ วีแซม จัดการอ่านหรือบันทึกให้

### 3.3 การทำงานของ ดีเอ็มซิส

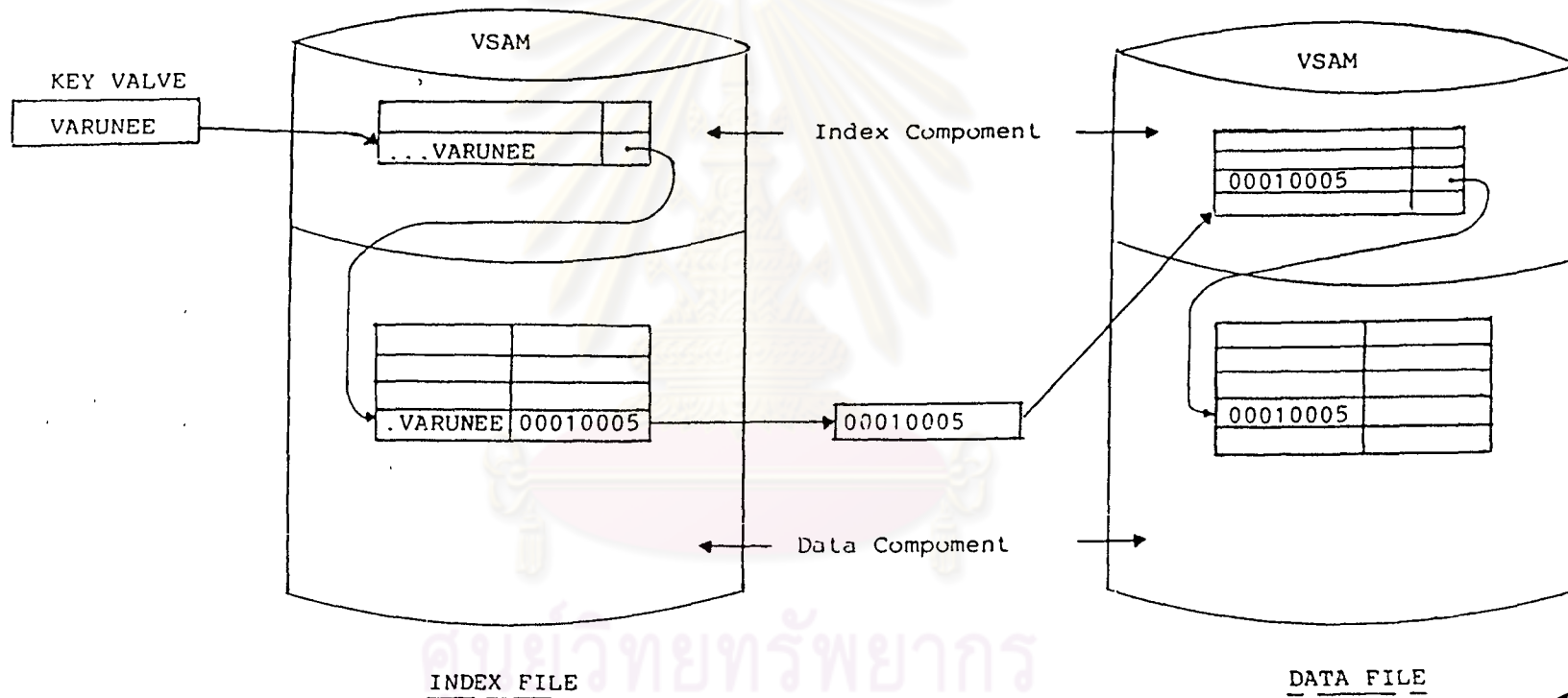
3.3.1 การบรรยายลักษณะข้อมูล ก่อนบันทึกข้อมูลลงในแฟ้มข้อมูลต่าง ๆ จะต้องมีการกำหนดรายละเอียดของข้อมูล และบันทึกไว้ในปทานุกรมข้อมูลก่อนเสมอ ซึ่งประกอบด้วย การบรรยายส่วนต่าง ๆ ดังนี้

3.3.1.1 บรรยายเขตข้อมูล เป็นการบอกรายละเอียดของเขตข้อมูลทั้งหมด ที่จะนำมารวมกันเป็นระเบียบข้อมูลของแฟ้มข้อมูลต่าง ๆ รายละเอียดของเขตข้อมูลนี้ ระบบจะนำไปสร้างระเบียบข้อมูล ตามรูปที่ 3.3 และบันทึกในปทานุกรมข้อมูล

3.3.1.2 บรรยายแฟ้มข้อมูล เป็นการบอกลักษณะโครงสร้างของแฟ้มข้อมูล ว่าประกอบด้วยเขตข้อมูลอะไรบ้าง โดยการกำหนดชื่อแฟ้มข้อมูล และชื่อของเขตข้อมูล ที่ประกอบกันเป็นระเบียบข้อมูล จากนั้นระบบจะจัดการคำนวณหาจุดเริ่มต้นของเขตข้อมูลแต่ละเขต ในระเบียบข้อมูล ความยาวของระเบียบข้อมูล และนำเอาข้อมูลเหล่านี้มาสร้างเป็นระเบียบข้อมูล ซี และ ดี ตามรูปที่ 3.4 และรูปที่ 3.5 ตามลำดับ นำไปบันทึกในปทานุกรมข้อมูลต่อไป

3.3.2 การบันทึกข้อมูล ดีเอ็มซีส จะถูกข้อกำหนดของแฟ้มข้อมูลจากปทานุกรมข้อมูล และสร้างส่วนนำของระเบียบข้อมูล ซึ่งประกอบด้วยหมายเลขแฟ้มข้อมูล และหมายเลขระเบียบข้อมูล เอาส่วนนำไปต่อกับข้อมูลที่อ่านเข้ามาเพื่อบันทึกเป็นระเบียบข้อมูล ตามรูปที่ 3.6 และส่งไปให้ วีแซม บันทึกในแฟ้มข้อมูล วีแซม จะนำระเบียบข้อมูลนี้ไปบันทึกในส่วนข้อมูล และเอาส่วนนำและที่อยู่ของระเบียบข้อมูลไปบันทึกในส่วนดัชนีของวีแซม ในการบันทึกข้อมูล เรา กำหนดให้หมายเลขแฟ้มข้อมูล และหมายเลขระเบียบข้อมูล เป็นคีย์ของ วีแซม ดังนั้นข้อมูลจะถูกบันทึกเรียงตามลำดับคีย์นี้

3.3.3 การสร้างดัชนีของระบบ เราสามารถที่จะกำหนดให้เขตข้อมูลใด ๆ ในแฟ้มข้อมูล ตั้งแต่ 1-3 เขตข้อมูล รวมกันเป็นคีย์ของระเบียบข้อมูลได้ โดยมีความยาวของเขตข้อมูลที่เป็นคีย์ รวมกันไม่เกิน 30 ไบต์ ระบบจะนำเอาหมายเลขเขตข้อมูลที่ถูกกำหนดให้เป็นคีย์ ไปบันทึกในระเบียบข้อมูล ซี ของปทานุกรมข้อมูล จากนั้นจะนำเอาค่าของคีย์ ในแต่ละระเบียบข้อมูล พร้อมทั้งที่อยู่ของระเบียบ (ซึ่งในที่นี้ หมายถึง หมายเลขระเบียบข้อมูล) นำไปสร้างเป็นระเบียบข้อมูลใหม่ ตามรูปที่ 3.8 และนำเอาระเบียบข้อมูลนี้ให้ วีแซม ไปบันทึกลงในแฟ้มดัชนี (ตามข้อ 3.1.3) ซึ่งในทำนองเดียวกัน วีแซม จะนำเอาระเบียบข้อมูลดังกล่าวไปบันทึกในส่วนข้อมูลของแฟ้มดัชนี และเอา 35 ไบต์แรกของระเบียบข้อมูล (ซึ่งถูกกำหนดให้เป็นคีย์ของ วีแซม) และที่อยู่ของระเบียบข้อมูลนี้ในแฟ้มดัชนีไปบันทึกในส่วนดัชนีของ วีแซม



รูปที่ 3.9 แสดงขั้นตอนการอ่านระเบียบข้อมูล โดยใช้ดัชนีของ คีเอ็มซิส



จะเห็นได้ว่าทั้งเพิ่มข้อมูล และเพิ่มดัชนีของระบบ เราสามารถเข้าถึงข้อมูลแบบ โดยตรง(direct) ได้ ถ้าเราทราบ คีย์ ของแต่ละเพิ่มข้อมูล ซึ่งทำให้ ดีเอ็มซิส สามารถทราบ ที่อยู่ของระเบียบข้อมูลแรกที้นที่ในเพิ่มข้อมูลได้ โดยเพียงแต่ทราบชื่อเพิ่มข้อมูล และนำเอาชื่อนี้ ไปค้นหาหมายเลขเพิ่มข้อมูลในปทานุกรมข้อมูล และนำไปต่อกับหมายเลขระเบียบข้อมูลคือ '0001' ก็จะได้ คีย์ของวีแชมสามารถอ่านระเบียบแรกได้ และในทำนองเดียวกันถ้าส่งค่าของเขตข้อมูลที่ เป็นคีย์ (ตามที่บรรยายไว้ในปทานุกรมข้อมูล) ไปให้ ดีเอ็มซิส ดีเอ็มซิส ก็สามารถค้นหาระเบียบ ข้อมูลที่ต้องการได้ ดังแสดงในรูปที่ 3.9 กล่าวคือ ดีเอ็มซิส จะส่งคีย์('VARUNEE')ไปให้วีแชม อ่านระเบียบในเพิ่มดัชนี ซึ่งจะได้ที่อยู่ของระเบียบข้อมูล '00010005' ในเพิ่มข้อมูล ซึ่งเป็นคีย์ ของระเบียบข้อมูลในเพิ่มข้อมูลนั่นเอง ดีเอ็มเอส จะส่งค่าคีย์ดังกล่าวไปให้ วีแชม อีกครั้งหนึ่ง เพื่อที่จะอ่านระเบียบข้อมูลที่ต้องการมาให้ ดีเอ็มซิส

### 3.3.4 การทำข้อมูลให้ทันสมัย (update data)

3.3.4.1 การเปลี่ยนชื่อเขตข้อมูล และเพิ่มข้อมูล เป็นการแก้ไข ข้อมูลเฉพาะในปทานุกรมข้อมูล

3.3.4.2 การเพิ่มข้อมูลบางเขตข้อมูลเข้าไปในระเบียบข้อมูล เป็นการเพิ่มเขตข้อมูล ซึ่งไม่ได้มีการกำหนดไว้ในเพิ่มข้อมูลนั้นมาก่อน วิธีการทำงานแบ่งเป็น 2 ขั้นตอน คือ

1) ระบบจะนำเอาหมายเลขเขตข้อมูลนั้นไปเพิ่มใน โครงสร้างของเพิ่มข้อมูล ในปทานุกรมข้อมูล โดยนำไปต่อท้ายระเบียบข้อมูลเดิม

2) นำข้อมูลไปต่อท้ายระเบียบข้อมูลเดิมในเพิ่มข้อมูล

3.3.4.3 การลบข้อมูล เป็นการทำความเครื่องหมายไว้เฉย ๆ โดยที่ ข้อมูลไม่ได้ถูกลบออกไปจริง ๆ แบ่งออกเป็น 3 อย่าง คือ

1) ลบเพิ่มข้อมูล เป็นการใส่เครื่องหมายดอกจัน('\*') ลงในไบต์ D ของระเบียบข้อมูลในปทานุกรมข้อมูล (ดูรูปที่ 3.4 ประกอบ)

2) ลบระเบียบข้อมูล เป็นการใส่เครื่องหมายดอกจัน ( '\*' ) ลงในไบต์ D ของระเบียบข้อมูลในแฟ้มข้อมูล ( รูปที่ 3.6 ประกอบ )

3) การลบเขตข้อมูล เป็นการลบเขตข้อมูลออกจากโครงสร้างของแฟ้มข้อมูล โดยระบบจะไปเปลี่ยนค่าของหมายเลขเขตข้อมูล ในระเบียบข้อมูล ที่ ในปทานุกรมข้อมูลเป็น '\*\*\*\*'

3.3.4.4 การแทนที่ระเบียบข้อมูล (replace record) เป็นการนำเอาระเบียบข้อมูลใหม่ไปบันทึกแทนที่ระเบียบข้อมูลเดิมในแฟ้มข้อมูล ถ้าหากค่าของเขตข้อมูลที่เป็นคีย์ของระเบียบข้อมูลใหม่ แตกต่างจากคีย์ของระเบียบข้อมูลเดิม ระบบจะไปลบคีย์เดิมในแฟ้มดัชนี และนำเอาคีย์ใหม่บันทึกลงไปแทน

3.3.4.5 การเปลี่ยนแปลงค่าของเขตข้อมูล (change field value) เป็นการเปลี่ยนแปลงค่าของระเบียบข้อมูลบางเขตข้อมูล และนำเอาระเบียบข้อมูลนั้นบันทึกลงไปในแฟ้มข้อมูลตามเดิม ในการเปลี่ยนแปลงนี้ หากเขตข้อมูลที่เป็นคีย์เปลี่ยนจะต้องแก้ไขในแฟ้มดัชนีเหมือนข้อ 3.3.3.4 ด้วย

ในการเข้าถึงระเบียบข้อมูลต่าง ๆ ในแฟ้มข้อมูล เพื่อทำข้อมูลให้ทันสมัยจะต้องมีการกำหนดตำแหน่งของระเบียบข้อมูลนั้น ๆ ก่อน โดยการบอกค่าคีย์ของระเบียบข้อมูล หรือของเขตข้อมูลใด เขตข้อมูลหนึ่ง ให้ระบบทราบ ระบบจะนำค่านั้นไปค้นหาระเบียบข้อมูลที่ต้องการ และจะบันทึกตำแหน่งของระเบียบข้อมูลนั้นไว้ในพื้นที่การทำงานของโปรแกรม เมื่อมีคำสั่งให้เปลี่ยนแปลงค่าของข้อมูล ระบบจะไปดูตำแหน่งของระเบียบข้อมูล และไปอ่านข้อมูลระเบียบที่ต้องการขึ้นมา เพื่อทำการเปลี่ยนแปลงค่าและบันทึกระเบียบข้อมูลนั้นกลับไปในตำแหน่ง เดิมต่อไป

3.3.5 การพิมพ์ข้อมูล เป็นการพิมพ์ข้อมูลซึ่งบันทึกอยู่ในแฟ้มข้อมูลหรือโครงสร้างของข้อมูลพื้นที่ในระบบ

3.3.5.1 การพิมพ์ข้อมูลทั้งหมดในแฟ้มข้อมูล จากแฟ้มข้อมูลที่ได้ระบุชื่อไว้

3.3.5.2 การพิมพ์ข้อมูลเฉพาะระเบียบข้อมูลที่กำหนดให้ โดยการกำหนดค่าของคีย์หรือค่าของเขตข้อมูลใดเขตข้อมูลหนึ่ง

3.3.5.3 การพิมพ์โครงสร้างของแฟ้มข้อมูล เป็นการนำเอาโครงสร้างของแฟ้มข้อมูลจากปทานุกรมข้อมูลมาพิมพ์

3.3.6 การจัดโครงสร้างของข้อมูลใหม่ (reorganization) เมื่อใช้ข้อมูลมาระยะหนึ่ง ข้อมูลมีการเปลี่ยนแปลง มีการเพิ่ม การลบ การขยาย และลด โครงสร้างของระเบียบข้อมูล ข้อมูลในระบบจึงมีส่วนหนึ่งที่เป็นขยะ จึงควรมีการจัดโครงสร้างใหม่ โดยลบส่วนที่เป็นขยะออกจากระบบ และจัดโครงสร้างของระเบียบข้อมูลใหม่ โดยขยับเขตข้อมูลที่ห่างกัน เนื่องจากการลบข้อมูลให้เข้ามาชิดกัน และบันทึกข้อมูลเฉพาะส่วนที่ยังไม่ไคลบ ลงในแฟ้มข้อมูลใหม่ จากนั้นจึงใช้คำสั่ง DELETE ของเอเอ็มเอส ลบแฟ้มข้อมูลปทานุกรมข้อมูล แฟ้มข้อมูล และแฟ้มดัชนี (ดูภาคผนวก ค. ประกอบ) แล้วสร้างแฟ้มข้อมูลทั้งสามขึ้นมาใหม่ ด้วยคำสั่ง DEFINE ของ เอเอ็มเอส (ดูภาคผนวก ข. ประกอบ) จากนั้นจึง โหลดข้อมูลจากแฟ้มข้อที่ สร้างขึ้นใหม่ กลับมาบันทึก ในปทานุกรม และแฟ้มข้อมูลอีกครั้งหนึ่ง ด้วยคำสั่ง REPRO (ดูภาคผนวก ง.)

### 3.4 กฎการเขียนคำสั่ง และสัญลักษณ์ที่ใช้ในการเขียน.

#### 3.4.1 กฎการเขียนคำสั่ง

1. เขียนคำสั่งลงบนบัตร เริ่มที่สัณคม์(Colume) ใดก็ได้ ตั้งแต่ สัณคม์ที่ 1-80 และการเริ่มต้นคำสั่งจะต้องขึ้นบัตรใหม่เสมอ
2. จบคำสั่งด้วยการเว้นว่าง(blank) หรือเครื่องหมายอัฒภาค(';')
3. คอ่บรหัทคัวยเครื่องหมายคิงค์ ('-')
4. ใช้เครื่องหมายลูกน้ำ(,) ช่องว่าง(blank) หรือเครื่องหมายวงเล็บ('(' หรือ ')') คั่นระหว่างคำหรือคำต่าง ๆ ในคำสั่ง
5. ถ้าเขียนคำหรือคำ ไม่จบในบรรทัดให้ใช้เครื่องหมายบวก('+') แสดงว่าคำหรือคำนั้นยังมีต่อในบรรทัดต่อไป

<sup>1</sup> IBM, VSAM-Using Access Method Service(OS/VS , OS/MVS),



### 3.4.2 สัญลักษณ์ที่ใช้ในการเขียนรายงาน

1. คำที่เขียนด้วยอักษรพิมพ์ใหญ่ แสดงว่าจะต้องสะกดตามนั้น ส่วนคำที่เขียนด้วยตัวเขียนเป็นชื่อหรือคำที่ใช้เป็นผู้กำหนดขึ้น
2. คำที่เขียนภายในเครื่องหมายก้ามปู (•[ ]•) เป็นคำที่มีหรือไม่มี(option) ในคำสั่งก็ได้
3. คำที่เขียนอยู่ในเครื่องหมายวงเล็บปีกกา (•[ ]•) แสดงว่าให้เลือกใช้อย่างใดอย่างหนึ่งเพียงอย่างเดียว

### 3.4.3 การตั้งชื่อแฟ้มข้อมูล และชื่อเขตข้อมูล

การตั้งชื่อให้ตัวแรกเป็นตัวอักษร และความยาวของชื่อไม่เกิน 8 อักขร ถ้ายาวเกิน 8 อักขร ระบบจะตัดเหลือเพียง 8 อักขร ถ้าความยาวไม่ถึง 8 อักขร ระบบจะเติมว่าง(blank) ให้ข้างหลังจนครบ 8 นอกจากนี้ห้ามมีเครื่องหมายต่าง ๆ ตามข้อ 3.4.1 ปรากฏอยู่ภายในชื่อ

### 3.5 คำสั่ง (Command) และรูปแบบการเขียน

3.5.1 คำสั่ง INIT ใช้ในการเริ่มระบบเป็นครั้งแรก และใช้ครั้งเดียวเท่านั้น มีรูปแบบการเขียนดังนี้

INIT

3.5.2 คำสั่ง DEFINE ใช้ในการกำหนดลักษณะข้อมูล แบ่งออกเป็น 3 อย่าง ตามพารามิเตอร์(Parameter) ที่ตามหลังคำสั่ง ดังนี้

- การกำหนดลักษณะของเขตข้อมูล มีรูปแบบการเขียนดังนี้

DEFINE FIELD -

field-name, length, type-  
:  
field-name, length, type;

- การกำหนดลักษณะของแฟ้มข้อมูล มีรูปแบบการเขียนดังนี้

```
DEFINE FILE file-name -
```

```
(field-name 1, ..., field-name n) ;
```

- การกำหนดคีย์ของแฟ้มข้อมูล มีรูปแบบการเขียนดังนี้

```
DEFINE KEY file-name -
```

```
(field-name 1 [, field-name 2 , field-name 3 ]),
```

เมื่อ length คือความยาวของเขตข้อมูล และ type เป็นรูปแบบของข้อมูล ซึ่งมีค่าและความหมายดังนี้

N = ตัวเลข (numeric)

C = ตัวอักษร (character)

### 3.5.3 คำสั่ง WRITE ใช้สำหรับบันทึกข้อมูล มีรูปแบบการเขียนดังนี้

```
WRITE file-name -
```

```
DATA
```

```
DATA FILE
```

```
END DATA ;
```

ข้อมูลใน data file เริ่มจากสคมภ์ที่ 1 เรียงต่อกันไปจนถึงสคมภ์ที่ 80 และถ้าระเบียบข้อมูลยาวกว่า 80 ไบต์ ให้ต่อในบรรทัดถัดไปได้ และการเริ่มระเบียบข้อมูลใหม่ จะต้องเริ่มที่สคมภ์ที่ 1 เสมอ สำหรับ DATA และ END DATA เป็นจุดเริ่มต้นและสุดท้ายของข้อมูล ตามลำดับ

3.5.4 คำสั่ง LOCATE ใช้สำหรับค้นหาที่อยู่ของระเบียบข้อมูล ซึ่งจะต้องกำหนดค่าของคีย์(key-value) ในพารามิเตอร์ KEY หรือกำหนดคีย์หมายเลขระเบียบข้อมูล(record no.) ในพารามิเตอร์ RECNO หรือกำหนดชื่อเขตข้อมูล(field-name) และค่า(Value)ในพารามิเตอร์ FIELD อย่างใดอย่างหนึ่ง ดังรูปแบบการเขียนต่อไปนี้



```

LOCATE    file-name -
{
  RECNO    record-no.;
  KEY      (key-value);
  FIELD    (field-name, value);
}

```

3.5.5 คำสั่ง CHANGE เป็นคำสั่งที่ใช้ในการเปลี่ยนค่าข้อมูล ในเขตข้อมูลของระเบียบข้อมูล ซึ่งกำหนดโดยคำสั่ง LOCATE มีรูปแบบการเขียนดังนี้

```
CHANGE    field-name, value ;
```

เมื่อ value คือค่าของข้อมูลที่ต้องการเปลี่ยน

3.5.6 คำสั่ง REPLACE เป็นคำสั่งที่ใช้ในการแก้ไขระเบียบข้อมูล โดยเอาระเบียบข้อมูลใหม่บันทึกทับระเบียบข้อมูลเดิม ตามตำแหน่งที่กำหนดโดยคำสั่ง LOCATE มีรูปแบบการเขียนดังนี้

```

REPLACE
DATA      }
           } data record
END DATA }

```

3.5.7 คำสั่ง DELETE ใช้ในการลบแฟ้มข้อมูล ระเบียบข้อมูล หรือเขตข้อมูล การลบระเบียบข้อมูลจะต้องกำหนดระเบียบข้อมูลด้วย คำสั่ง LOCATE ก่อน สำหรับการลบเขตข้อมูลเป็นการเปลี่ยนแปลงโครงสร้างของระเบียบข้อมูล ถ้าไม่มีชื่อแฟ้มข้อมูล (file-name) หลังพารามิเตอร์ FIELD แสดงว่า ลบเขตข้อมูลนี้ออกจากระบบ โดยจะลบออกจากทุกแฟ้มข้อมูล และลบออกจากปทานุกรมข้อมูลด้วย แต่ถ้ามีชื่อแฟ้มข้อมูลจะลบเฉพาะในแฟ้มข้อมูลที่ระบุชื่อเท่านั้น มีรูปแบบการเขียนดังนี้

```

DELETE {
  FILE      file-name;
  RECORD ;
  FIELD (field-name [, file-name]);
}

```

3.5.8 คำสั่ง ADD ใช้สำหรับเพิ่มเขตข้อมูลเข้าไปในโครงสร้างของระเบียบข้อมูล ซึ่งบรรยายอยู่ในโปรแกรมข้อมูล และเพิ่มค่าของเขตข้อมูลเข้าไปในแฟ้มข้อมูล มีรูปแบบการเขียนดังนี้

```

ADD      field-name, file-name
DATA    ]
        ]-field-value
END DATA ;

```

เมื่อ field-value คือข้อมูลที่นำไปเพิ่มในแฟ้มข้อมูล

3.5.9 คำสั่ง RENAME ใช้สำหรับเปลี่ยนชื่อแฟ้มข้อมูลหรือเขตข้อมูล มีรูปแบบการเขียนดังนี้

```

RENAME {
  FILE (old-name, new-name);
  FIELD (old-name, new-name);
}

```

3.5.10 คำสั่ง DISPLAY ใช้สำหรับพิมพ์โครงสร้างของแฟ้มข้อมูล มีรูปแบบการเขียนดังนี้

```

DISPLAY [file-name,...] ;

```

ถ้าไม่ระบุชื่อแฟ้มข้อมูลจะพิมพ์โครงสร้างของแฟ้มข้อมูลทั้งหมด

3.5.11 คำสั่ง TYPE ใช้ในการพิมพ์ข้อมูลทุกระเบียนข้อมูล จากแฟ้มข้อมูลที่กำหนด มีรูปแบบการเขียนดังนี้

```

TYPE file-name ;

```

3.5.12 คำสั่ง LIST ใช้ในการพิมพ์เฉพาะระเบียบข้อมูลเดียว ตามที่ LOCATE  
ไว้ในขณะนั้น มีรูปแบบการเขียนดังนี้

LIST

3.5.13 คำสั่ง REORGZ ใช้สำหรับจัดโครงสร้างใหม่ มีรูปแบบการเขียน  
ดังนี้

REORGZ ;



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย