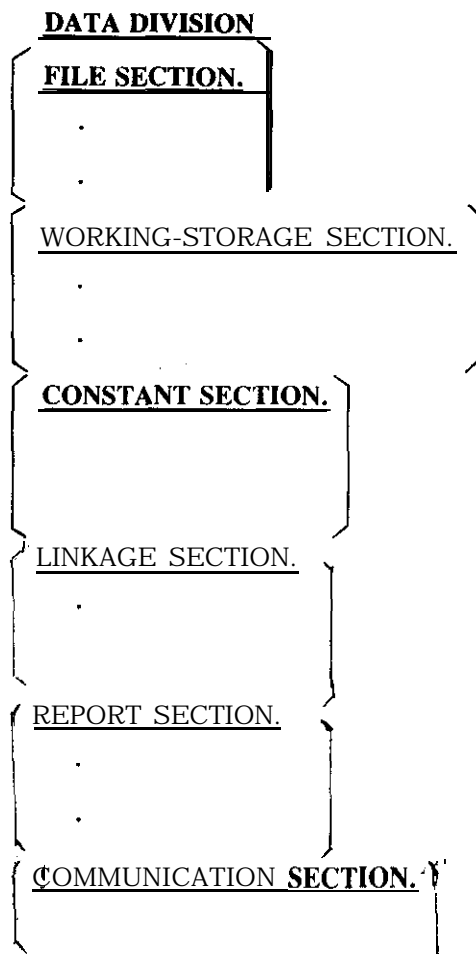


บทที่ 5

Data division

division นี้ มีไว้สำหรับให้โปรแกรมเมอร์ เขียนบรรยายรายละเอียดของข้อมูลที่จะนำไปประมวลผล กับ object program ว่ามีลักษณะเก็บอยู่ใน items ต่างๆ อย่างไรบ้าง item ในที่นี้ หมายถึงเนื้อที่ที่กำหนดตำแหน่งไว้แล้วในหน่วยความจำ และมูลค่าที่อยู่ในเนื้อที่นี้ก็คือมูลค่าของ item นั้นเอง ข้อมูลที่ถูกบันทึก ใน external device จะมีลักษณะเป็น external BCD format แต่ถ้าข้อมูลเก็บอยู่ในหน่วยความจำจะอยู่ในลักษณะของ internal BCD format โดยมีรูปแบบ ดังนี้



division นี้ แบ่งออกเป็น 6 sections ใน file section จะประกอบด้วย file description entries และ record description entries แต่ใน working-storage section และ constant section จะมีเฉพาะ record description entries และ data item description entry เท่านั้น

ถ้าในการประมวลผลนั้นไม่ต้องใช้แฟ้มข้อมูลส่วนที่เป็น file section ก็ไม่ต้องเขียนในทำนองเดียวกัน working-storage section และ constant section ถ้าส่วนไหนไม่จำเป็นต้องใช้ก็ไม่ต้องเขียนส่วนนั้น

สำหรับรายละเอียดของ Linkage section ให้ดูบทที่ 7 เรื่องโปรแกรมย่อย (Sub-programs) และ Report section ดูเรื่อง Report generation ส่วน Communication section ใช้สำหรับ access, process และ create messages สำหรับสื่อสารกับ local และ remote communication devices อย่างไรก็ตาม ทั้งสาม section เหล่านี้ไม่ค่อยได้ใช้ เพราะว่าฟังก์ชันเหล่านี้มีอยู่แล้วใน software packages ซึ่งจัดสรรโดยผู้ผลิตเครื่องคอมพิวเตอร์ จึงไม่ได้ให้รายละเอียดเรื่อง report section และ communication section ในหนังสือเล่มนี้

5.1.1 File description entry

ใช้สำหรับบรรยายลักษณะทาง physical ของแฟ้มข้อมูลแต่ละชุด ตัวอย่างเช่น recording density, block size และ labeling conventions ข้อเท็จจริงเหล่านี้จะทำให้ object program สามารถแปลความหมายหรือสร้างแฟ้มข้อมูลได้อย่างถูกต้อง โดยมีรูปแบบดังนี้

$$\begin{array}{l}
 \text{FD file-name-1} \left[\text{RECORDING MODE IS} \left\{ \begin{array}{l} \text{BINARY} \\ \text{DECIMAL} \end{array} \right\} \left\{ \begin{array}{l} \text{LOW} \\ \text{HIGH} \\ \text{HYPER} \end{array} \right\} \text{DENSITY} \right] \\
 \\
 \text{LABEL} \left\{ \begin{array}{l} \text{RECORDS} \\ \text{RECORD} \end{array} \right\} \left\{ \begin{array}{l} \text{ARE} \\ \text{IS} \end{array} \right\} \left\{ \begin{array}{l} \text{STANDARD VALUE OF} \\ \text{NON-STANDARD} \\ \text{OMITTED} \end{array} \right\} \left\{ \begin{array}{l} \text{ID} \\ \text{IDENTIFICATION} \end{array} \right\} \text{IS literal-} \\
 \\
 \left[\text{BLOCK CONTAINS integer-1} \left\{ \begin{array}{l} \text{RECORDS} \\ \text{CHARACTERS} \end{array} \right\} \right]
 \end{array}$$

$$\left\{ \begin{array}{l} \text{RECORD CONTAINS integer-2 CHARACTERS} \\ \text{DATA } \left\{ \begin{array}{l} \text{RECORDS} \\ \text{RECORD} \end{array} \right\} \left\{ \begin{array}{l} \text{ARE} \\ \text{IS} \end{array} \right\} \text{data-name-1 [data-name-2]} \end{array} \right\}$$

เพื่อความสะดวกในการอธิบายความหมาย จึงให้หมายเลข 1,2,3,4,5 สำหรับบรรทัดที่ 1, 2, 3, 4 และ 5 ใน file description entry ตามลำดับ

ทั้ง 5 หมายเลขนี้จะเรียงลำดับ อันไหนก่อน อันไหนหลังก็ได้ แต่เวลาเขียนต้อง แสดงให้เห็นว่าเป็นคนละส่วนกัน โดยใส่ space อย่างน้อยหนึ่งที ห้ามใช้เครื่องหมายกำกับ บรรทัดตอนระหว่างข้อความเมื่อจบแต่ละ entry ให้ใส่หนึ่ง period

FD เป็น level indicator ต้องเขียนที่ มาร์จิ้น A (คอลัมน์ 8-11) เป็นอันดับแรกใน file description entry

file-name-1 เป็นชื่อเดียวกับที่นิยาม (defined) มาแล้วใน SELECT entry ใน environment division และจะนำไปใช้อีกในคำสั่ง open และ close ต่อไป และถ้าเป็น input file ชื่อนี้จะปรากฏ ในคำสั่ง read ด้วย

หมายเลข 1 เรียกว่า recording mode clause ใช้กับเทปไฟล์เท่านั้น ถ้าไม่เขียน clause นี้ หมายความว่า recording mode เป็น decimal และ high density

ข้อมูลที่บันทึกในเทป (tape) ลักษณะ binary code เวลาประมวลผล (processed) จะ ทำได้รวดเร็วกว่าข้อมูลที่บันทึกเทปในลักษณะ decimal code

density หมายถึง ความหนาแน่นของเนื้อเทป แบ่งออกเป็น 3 ชนิด คือ

1) 200 bpi หรือเรียกว่า low density

bpi ย่อมาจาก bytes หรือ characters per inch

หมายความว่าเทปชนิดนี้ในเนื้อเทป ความยาวหนึ่งนิ้ว สามารถบันทึกข้อมูลได้ 200 ตัว

2) 556 bpi เรียกว่า high density เทปชนิดนี้ในเนื้อเทปความยาวหนึ่งนิ้ว บันทึกข้อมูล ได้ 556 ตัว

3) 800 bpi เรียกว่า hyper density เทปชนิดนี้ในเนื้อเทปความยาวหนึ่งนิ้ว บันทึก ข้อมูลได้ 800 ตัว

หมายเลข 2 เรียกว่า label record clause, clause นี้ต้องเขียนเสมอ label record อาจจะเป็นเรคคอร์ดแรกหรือเรคคอร์ดสุดท้ายในแฟ้มข้อมูลก็ได้

OMITTED หมายความว่าแฟ้มข้อมูลนั้นไม่มี label record

NON-STANDARD หมายความว่าแฟ้มข้อมูลนั้นมี label record แต่ไม่ต้องตรวจสอบ

ในขณะที่มีการประมวลผล

STANDARD หมายความว่า แฟ้มข้อมูลนั้นมี label record ประกอบด้วยตัวอักขระ 80 ตัวให้ตรวจสอบในขณะที่มีการประมวลผล literal-1 เป็น non-numeric literal ประกอบด้วยตัวอักขระไม่เกิน 14 ตัว แต่ถ้าน้อยกว่า 14 ตัว เครื่องจะเติม blank ทางซ้ายมือให้จนครบ 14 ตัว

หมายเลข 3 เรียกว่า block contains clause

มีไว้สำหรับบอกขนาดของ block (physical record) ถ้าไม่เขียน clause นี้ หมายความว่าหนึ่ง block ประกอบด้วยหนึ่ง logical record

ตัวอย่าง ข้อมูลบันทึกในบัตร

บัตร, 1 block เท่ากับ 1 logical record = 80 characters

พรินเตอร์, 1 block เท่ากับ 1 logical record = 136 characters

ถ้าเลือกคำ RECORDS ลักษณะของข้อมูลจะเป็น fixed size logical records

ถ้าเลือกคำ CHARACTERS ลักษณะของข้อมูลจะเป็น variable size logical records

หมายเลข 4 clause นี้เป็นตัวบอกว่าในหนึ่ง logical record นั้น มีกี่ตัวอักขระ ถ้าแฟ้มข้อมูลนั้นบันทึกในบัตรหรือกระดาษต่อเนื่องไม่ต้องเขียน เพราะทราบอยู่แล้ว

หมายเลข 5 เรียกว่า data records clause

data-name-1, data-name-2,.....ชื่อเหล่านี้เป็นชื่อเรคคอร์ดในแฟ้มข้อมูลนั้น ๆ จะเรียงลำดับชื่อเรคคอร์ดอันไหนก่อน อันไหนหลังก็ได้ และชื่อพวกนี้จะถูกนำไปบรรยายรายละเอียดอีกครั้งหนึ่ง โดยใช้เลขบอกระดับ 01 ใน record description entry ต่อไป

ตัวอย่าง file description entry

FD CARD-FILE LABEL RECORD IS OMITTED DATA RECORD IS CARD-REC.

FD PRINT-FILE LABEL RECORD IS OMITTED DATA RECORD IS PRINT-REC.

FD TAPE-FILE RECORDING MODE IS DECIMAL HIGH DENSITY LABEL RECORD

IS STANDARD VALUE OF ID IS "RU 1000-201" BLOCK CONTAINS 20 RECORDS
RECORD CONTAINS 80 CHARACTERS DATA RECORD IS TAPE-REC.

FD PERSONNEL-FILE LABEL RECORD IS OMITTED DATA RECORDS ARE
PERSONNEL-SALARY, PERSONNEL-HOURLY.

01 PERSONNEL-SALARY.

...

01 PERSONNEL-HOURLY.

5.1.2 Record description entry

ใช้สำหรับบรรยายลักษณะของข้อมูลแต่ละตัวที่เก็บอยู่ภายในหน่วยความจำของคอมพิวเตอร์ว่ามีลักษณะอย่างไร (data description entry) โดยที่ item แต่ละตัวต้องบรรยายแยกกันเป็นกนละบรรทัด โดยมีรูปแบบดังนี้

level number { data-name } [REDEFINES..] [OCCURS...] [PICTURE.....]
 { FILLER }
 [edit clauses..] [VALUE...] [SIZE...] [CLASS...] [USAGE...]
 ...

หมายเหตุ ถ้ามี REDEFINES clause ต้องตามหลัง data-name ทั้งนี้ ส่วน clause อื่น ๆ จะเรียงลำดับกันอย่างไรก็ได้

เลขบอกระดับ (level number) ใน record description entry ทุกชุดจะต้องมีเลขบอกระดับซึ่งประกอบด้วยเลขสองหลักเสมอ จะเขียนที่คอลัมน์ 8 หรือทางขวามือของคอลัมน์ 8 ก็ได้ ในกรณีที่เขียนเลขหลักเดียว (เลข 1-9) เราจะใส่เลข 0 หรือ blank ข้างหน้า เพื่อให้เป็นสองหลักได้, data-name และ level number ต้องเขียนในบรรทัดเดียวกัน แยกกันด้วย space อย่างน้อยหนึ่งตัว

(สำหรับกฎเกณฑ์ในการตั้งชื่อ data-name นั้นได้กล่าวมาแล้วในบทที่ 3)

คำว่า FILLER ใช้กับข้อมูลเดี่ยวหรือข้อมูลกลุ่มก็ได้ หมายถึง ข้อมูลส่วนนี้ไม่มีการอ้างถึงในโปรแกรม จึงไม่จำเป็นต้องตั้งชื่อ

โดยปกติแล้วใน logical record หนึ่งๆ อาจแบ่งออกได้เป็นหลายระดับ (levels) เช่น เป็น group item เป็น elementary item.

ข้อมูลเดี่ยว (elementary item) เป็น item ที่มีระดับต่ำที่สุด หมายถึง item ที่ไม่แบ่งแยกต่อไปอีก

ข้อมูลกลุ่ม (group item) เป็น collection ของข้อมูลเดี่ยวหรือข้อมูลกลุ่มอื่น ๆ ซึ่งจะแบ่งย่อยต่อไปได้อีกเป็นระดับต่ำกว่ามันมากที่สุดได้ถึง 49 ระดับ

ตัวอย่าง รายละเอียดของข้อมูลในบัตร มีดังนี้

data-name-1

data-name 2	3			data-name 7
	4	5	6	

จะเห็นว่า

item ที่เป็นข้อมูลเดี่ยว ได้แก่ data-name-2, data-name-4, data-name-5, data-name-6, และ data-name-7

item ที่เป็นข้อมูลกลุ่ม ได้แก่ data-name-1 และ data-name-3 ในที่นี้ data-name-1 เป็น item ที่มีระดับสูงสุด

data-name-2, data-name-3, และ data-name-7 มีระดับเท่ากัน แต่มีระดับต่ำกว่า data-name-1

data-name-4, data-name-5, data-name-6 อยู่ในระดับเดียวกัน และเป็น items ที่มีระดับต่ำสุด

กฎเกณฑ์ในการกำหนดเลขบอกระดับ (level numbers) ให้กับ item แต่ละตัว (data description entry)

1) เลขบอกระดับ 01 กำหนดให้กับ item ที่เป็นชื่อเรคคอร์ดเท่านั้น
ตัวอย่าง

	↓	คอลัมน์ 8
	01	data-name-1.
หรือ	01	PERSONNEL-RECORD.
หรือ	01	PERSONNEL-RECORD PICTURE X(80).

2) ถ้า item ตัวหนึ่งประกอบด้วย items อื่น ๆ ที่มีระดับต่ำกว่า item ที่มีระดับต่ำกว่า ต้องเขียนทีหลัง ในอีกบรรทัดหนึ่ง

และเลขบอกระดับของ item ที่มีระดับต่ำกว่าต้องเป็นเลขมีค่ามากกว่าเลขบอกระดับของ item ที่มีระดับสูงกว่าตัวมัน

ตัวอย่าง

คอลัมน์ 8	↓	คอลัมน์ 12	↓	คอลัมน์ 16
		02		DATE-HIRED.
				03 MO-HIRED PICTURE 99.
				03 DA-HIRED PICTURE 99.
				03 YR-HIRED PICTURE 99.
		02		OVERTIME-PERCENT PICTURE V999.

3) items ต่าง ๆ ที่อยู่ในระดับเดียวกัน เลขบอกระดับต้องเหมือนกัน .

4) item ที่อยู่ในระดับสูงกว่าข้อมูลเดี่ยวที่ผ่านมา ต้องให้เลขบอกระดับเหมือนกับเลขบอกระดับของข้อมูลกลุ่มที่ผ่านมา

5) เลขบอกระดับที่เป็นตัวเลขสูงกว่าหรือต่ำกว่า ไม่จำเป็นต้องเขียนเรียงมูลค่าตัวเลขตามลำดับ เพียงแต่ให้มีความหมายบอกความสัมพันธ์ในระดับที่ถูกต้องก็พอ

จากตัวอย่างเดิมเมื่อกำหนดเลขบอกระดับให้กับแต่ละ item จะอยู่ในรูปต่อไปนี้

01 data-name-1

03 data-name-2

03 data-name-3

เลขบอกระดับ 77 และ 01 ใน Working-storage section และ Linkage section ส่วนของ data-name ต้องเป็นชื่อที่ unique

05 data-name-4

05 data-name-5

05 data-name-6

03 data-name-7

6) เลขบอกระดับ 88 ใช้กับข้อมีเงื่อนไข (condition-name) ใน data division เท่านั้น ข้อมีเงื่อนไข หมายถึงชื่อ ที่กำหนดให้กับ item ซึ่งมีค่าทราบแน่นอนไว้แล้ว ตั้งแต่หนึ่งค่าขึ้นไป วิธีการตั้งข้อมีเงื่อนไขใช้กฎเกณฑ์เดียวกับการตั้งชื่อทั่วไป

โดยมีรูปแบบดังนี้

88 condition-name { VALUE IS } literal-1 [THRU literal-2] literal-3
 VALUES ARE
 [THRU literal-4].....

ข้อความข้างต้นนี้เรียกว่า condition-name entry ใช้บรรยายลักษณะของ data item ซึ่งเป็นข้อมูลเดียว และต้องอยู่ใน file section หรือ working-storage section เท่านั้น

ตัวอย่าง ในบัตรประวัติ (master card) ของนักศึกษา ข้อมูลในคอลัมน์ 10 แสดงสถานภาพสมรส ดังนี้

- ถ้าเป็นเลข 0 หมายถึง นักศึกษาชาย แต่งงานแล้ว
 - ถ้าเป็นเลข 1 หมายถึง นักศึกษาชาย โสด
 - ถ้าเป็นเลข 2 หมายถึง นักศึกษาหญิง แต่งงานแล้ว
 - ถ้าเป็นเลข 3 หมายถึง นักศึกษาหญิง โสด
- เราจะเขียน data description entries ได้ดังนี้

03 SEX PICTURE 9.

88 MALE-M VALUE IS 0.

88 MALE-S VALUE IS 1.

88 FEMALE-M VALUE IS 2.

88 FEMALE-.SVALUE IS 3.

เมื่อ MALE-M, MALE-S, FEMALE-M, FEMALE-S เป็นข้อมีเงื่อนไข

ตัวอย่าง 2

03 STD-CODE PICTURE 9(2).

88 YEAR VALUES ARE 14 THRU 20.

ในที่นี้ YEAR เป็นชื่อมีเงื่อนไข

7) เลขบอกระดับ 77 กำหนดให้กับ data-name ซึ่งเป็นข้อมูลอิสระ (independent item) หมายถึง item ที่ไม่สัมพันธ์กับข้อมูลอื่นใน working-storage section เท่านั้น และใน entry เดียวกันนี้จะมี OCCURS clause อยู่ด้วยไม่ได้

ตัวอย่าง

WORKING-STORAGE SECTION.

77 SUM-SCORES PICTURE 999 VALUE ZEROS.

77 AVERAGE PICTURE 999V99 VALUE ZEROS.

8) เลขบอกระดับ 66 ใช้กับ RENAMES clause เท่านั้น (ดูรายละเอียดหน้า 82)

PICTURE clause

ใช้สำหรับนิยาม (define) ลักษณะของ item ซึ่งเตรียมเนื้อที่เก็บในหน่วยความจำแบ่งออกเป็น 4 ชนิดคือ alphabetic, numeric, alphanumeric และ report item โดยมีรูปแบบดังนี้

$\left\{ \begin{array}{l} \text{PIC} \\ \text{PICTURE} \end{array} \right\}$ IS any-allowable-combination-of-characters-and-symbols

1. **Alphabetic item** จะมี picture เป็นตัวอักษร A, A แต่ละตัว หมายถึง มูลค่าในแต่ละตำแหน่งในฟิลด์นั้นเป็นตัวอักษรตัวใดตัวหนึ่ง (A ถึง Z) หรือเป็น space ก็ได้

ตัวอย่าง

03 LITERAL-1 PICTURE A(14).

03 FIELD-1 PICTURE AAAAA.

03 FIELD-2 PICTURE A(28).

2. **Alphanumeric item** จะมี picture เป็นตัวอักษร X, หมายถึง แต่ละตำแหน่งใน item นั้น ประกอบด้วยตัวอักษรตัวใดตัวหนึ่ง (A ถึง Z) หรือ space หรือตัวเลข (0 ถึง 9) หรือสัญลักษณ์พิเศษ เช่น period, comma, หรือ dollar sign เป็นต้น

ตัวอย่าง

03 STUDENT-NAME PICTURE X(25).

03 GRADE PICTURE X.

3. **Numeric item** จะมี picture ที่ประกอบด้วย combination ของตัวอักษรต่อไปนี้ 9, V, S และ P โดยที่ตัวอักษรหนึ่งตัว แทนตำแหน่งใน data item หนึ่งตำแหน่ง

9 ตำแหน่งของตัวอักษรใน item นั้นเป็นตัวเลข (0-9)

V บอกตำแหน่งของจุดทศนิยม แต่ไม่นับ V รวมอยู่ในขนาดของ item นั้น

ตัวอย่าง ข้อมูลของเรามีฟิลด์ percent เป็นเลขทศนิยม 3 หลักเสมอ กำหนด picture ดังนี้

03 PERCENT-FIELD PICTURE V999.

ในบัตรข้อมูลจริง ๆ ไม่ต้องเจาะจุดทศนิยม เพราะเครื่องจะนับให้เอง ตามตัวอย่าง ฟิลด์นี้จะใช้เนื้อที่สามคอลัมน์

ตัวอย่าง

แสดงข้อมูลในหน่วยความจำ

03 FIELD-3 PICTURE 99V99.

3	8	5	0
---	---	---	---

03 FIELD-4 PICTURE 9(5)V99.

		2	1	0	0	0
--	--	---	---	---	---	---

ถ้าเราพิมพ์ค่าของ FIELD-3 หรือ FIELD-4 ออกมาทางพรินเตอร์ (printer) จะไม่ปรากฏจุดทศนิยมให้เห็นกัน ผลลัพธ์คือค่าของ FIELD-3 = 3850 และค่าของ FIELD-4 = 0021000

S บอกตำแหน่งของเครื่องหมายบวกหรือลบหนึ่งตัว กฎเกณฑ์การใช้เหมือนกับ V (assumed decimal position) ตรงที่ไม่ต้องแทนหนึ่งตำแหน่งในบัตรข้อมูล กล่าวคือเราไม่นับ S รวมอยู่ในขนาดของ item นั้น แต่ S ต้องเป็นตัวอักษรตัวซ้ายมือสุดใน picture clause

ตัวอย่าง

03 A PICTURE S9(2).

01 NUMBER PICTURE S9(8).

ตัวอย่างหลัง ลักษณะในบัตรข้อมูลจะเป็นดังนี้

00000648

หมายถึง NUMBER = -648

ในกรณีที่มีข้อมูลมีค่าเป็นลบ การส่งข้อมูลเข้าเครื่องคอมพิวเตอร์โดยทางเครื่องอ่านบัตร (cardreader) ให้เจาะเครื่องหมาย-ทับตัวอักษรตัวขวามือสุดในฟิลด์ ตัวอย่างนี้พิมพ์เครื่องหมาย--ทับเลข 8 ในคอลัมน์ 8

ตัวอย่าง

03 FIELD-5 PICTURE S9(5)V99.

03 FIELD-6 PICTURE S999.

ในกรณีที่ข้อมูลมีค่าเป็นลบ ข้อมูลอยู่ในบัตรหรือส่งผ่านทางคีย์บอร์ดของเทอร์มินัล เราใส่ข้อมูลลบโดยตรวจสอบมูลค่าลบของเลขทั้งสิบตัวดังนี้ ตัวอย่างเช่น มูลค่าลบคือ -0, -1, -2, -3, ..., -9 ให้แทนด้วยมูลค่าต่อไปนี้ตามลำดับ I, J, K, L, ..., R ส่วน -100, -344 และ -189 ให้แทนด้วยมูลค่า 10I, 34M และ 18R ตามลำดับ

P บอกตำแหน่งของจุดทศนิยมซึ่งไม่ได้เกิดจริงใน data item เมื่อ P อาจเป็นตัวอักษรระ ช้ายมือหรือขวามือของ item ก็ได้ P หนึ่งตัวหมายถึงเพิ่มเลข 0 หนึ่งตัวและตามด้วยจุด ทศนิยมซ้ายมือ หรือขวามือของ P ตัวสุดท้ายตามตำแหน่งที่ P อยู่ และไม่นับ P รวมในขนาด ของ item นั้น

ตัวอย่าง คำสั่ง MOVE 25 TO A. ถ้ากำหนด picture ของ A ตามตารางข้างล่างนี้เป็นเครื่อง execute จะได้ค่าต่าง ๆ กันดังนี้

PICTURE	มูลค่า A
YPPPP	250000.
PPPPY	.000025
9VPP9	2.005
VPPP99	.00025
SVP999	+ .0025

4. **Report item** หมายถึง item ที่มีการบรรณาธิกรณ (editing) ด้วยตัวอักษรต่าง ๆ ให้อยู่ในรูปสะดวกสบายในการอ่าน หรือให้อยู่ในรูปแบบสวยงามอย่างที่โปรแกรมเมอร์ต้องการ ก่อนจะพิมพ์ออกมาเป็นผลลัพธ์ขั้นสุดท้ายจากพรินเตอร์

การอีดิต (edit) เกิดขึ้นเมื่อมีการเคลื่อนย้ายข้อมูลจากเนื้อที่หนึ่งในหน่วยความจำไป ยังเนื้อที่อื่นในหน่วยความจำ โดยมีข้อจำกัดต่าง ๆ ดังนี้

1) report item หนึ่งจะรับได้เฉพาะข้อมูลที่เป็นตัวเลข (numeric data) เท่านั้น หลัง จากอีดิตแล้วเรียกว่า numerically edited item พุดได้อีกอย่างหนึ่งก็คือ source item ต้องเป็น ตัวเลขทั้งหมด

2) การเคลื่อนย้าย numeric character ไปยัง report item หนึ่ง ๆ จะทำได้มากที่สุด เพียง 12 ตัวเท่านั้น (maximum size of source item)

3) report item หนึ่ง ๆ จะกำหนดความยาวของฟิลด์ได้มากที่สุด 63 ตัวอักษร

Picture editing

ตัวอักษรที่ใช้บรรยายลักษณะของ report item ใน PICTURE clause ได้แก่

9 V S P \$ + - . , 0 B / CR DB Z *

ตัวอักษร 9, V, S และ P ใช้ในความหมายเหมือนกับที่ใช้ใน numeric item, ตัวอักษร ส่วนที่เหลือแบ่งออกเป็นสองกลุ่ม คือ

a) Insertion characters

ได้แก่ ตัวอักษรที่จะใส่เพิ่มใน report item และจะปรากฏใน edited data item ตรง ตำแหน่งที่มันอยู่ นั่นคือ ขนาดของ report item ต้องนับรวมตัวอักษรเหล่านี้ด้วย ได้แก่ \$ + - . , 0 B / CR DB

- \$ dollar sign หนึ่งตัวต้องเป็นตัวอักษรซ้ายมือสุดใน picture ของ report item เครื่องจะใส่ \$ หนึ่งตัวตรงตำแหน่งนั้น
- + อาจเป็นตัวอักษรซ้ายมือสุด หรือขวามือสุดใน picture ของ report item ก็ได้, ถ้า ข้อมูลมีค่าเป็นลบ มันจะใส่เครื่องหมายลบให้, ถ้าข้อมูลมีค่าเป็นบวก หรือไม่มีเครื่องหมาย มันจะใส่เครื่องหมายบวกให้
- อาจเป็นตัวอักษรซ้ายมือสุด หรือขวามือสุดใน picture ของ report item ก็ได้, ถ้า ข้อมูลมีค่าเป็นลบ มันจะใส่เครื่องหมายลบให้, ถ้าข้อมูลมีค่าเป็นบวก หรือไม่มีเครื่องหมาย มันจะใส่ blank ให้ตรงตำแหน่งของตัวอักษรตัวนั้น (decimal point) แทนจุดทศนิยมที่เป็นจริง ใน picture ของ report item และ item หนึ่งตัว จะมีจุดทศนิยมมากกว่าหนึ่งจุดไม่ได้ เครื่องจะใส่จุดทศนิยมหนึ่งตัว ตรงตำแหน่งนั้น
- ,
- เครื่องหมาย, (comma) จะปรากฏใน edited data item ตรงตำแหน่งที่กำหนดไว้ใน PICTURE clause ในฟิลด์หนึ่ง ๆ จะมีเครื่องหมาย comma ก็ตัวก็ได้
- 0 เลขศูนย์ จะปรากฏใน edited data item ตรงตำแหน่งที่กำหนดไว้ใน PICTURE clause
- B จะให้ blank ใน edited data item ตรงตำแหน่งที่กำหนดไว้ใน PICTURE clause
- / จะให้เครื่องหมาย slash ใน edited data item แต่เนื่องจากว่า โดยปกติ ไม่ถือว่า slash* เป็น insertion character, คอมไพเลอร์จะ process / ให้แต่ก็พิมพ์ error diagnostic มา ด้วย

* เฉพาะเมื่อประมวลผลกับเครื่องคอมพิวเตอร์ CDC

- CR** แทน credit ต้องเป็น symbol ตัวขวามือใน picture ของ report item เท่านั้น ถ้าข้อมูลมีค่าเป็นลบ มันจะได้ CR ให้ใน edited data item ทางขวามือสุดเช่นกัน ถ้าข้อมูลมีค่าบวกหรือไม่มีเครื่องหมาย มันจะ blank ไว้สองตัวทางขวามือสุด
- DB** แทน debit ต้องเป็น symbol ตัวขวามือสุด ใน picture ของ report item, ผลลัพธ์เหมือนกับ credit symbol แต่แทนที่ CR ด้วย DB

ตัวอย่าง 1

03 DATE PICTURE 9(6) VALUE 120385.

03 EDDATE PICTURE 99/99/99.

คำสั่ง MOVE DATE TO EDDATE. จะทำให้เนื้อที่ชื่อ .. EDDATE มีมูลค่าเป็น 12/03/85

ตัวอย่าง 2

03 EDNAME PICTURE XBXXB (IO).

คำสั่ง MOVE "RBSMITH" TO EDNAME.

จะให้ผลลัพธ์ดังนี้

R		B		S	M	I	T	H				
---	--	---	--	---	---	---	---	---	--	--	--	--

ตัวอย่าง 3

03 EDSUM PICTURE 9(4)000.

คำสั่ง MOVE 1365 TO EDSUM.

จะให้ผลลัพธ์ดังนี้

1	3	6	5	0	0	0
---	---	---	---	---	---	---

EDSUM

ตัวอย่าง 4

Source data	Editing Picture	Edited item
4 8	\$99	\$ 4 8
4 8 3 4	\$99.99	\$ 4 8 . 3 4
4 8 3 4	9,999	4 , 8 3 4
2 9 2	+999	+ 2 9 2
2 9 2	+999	- 2 9 2

<table border="1"><tr><td>2</td><td>9</td><td>Z</td></tr></table>	2	9	Z	999-	<table border="1"><tr><td>2</td><td>9</td><td>2</td><td>-</td></tr></table>	2	9	2	-							
2	9	Z														
2	9	2	-													
<table border="1"><tr><td>2</td><td>9</td><td>2</td></tr></table>	2	9	2	999-	<table border="1"><tr><td>2</td><td>9</td><td>2</td><td></td></tr></table>	2	9	2								
2	9	2														
2	9	2														
<table border="1"><tr><td>2</td><td>4</td><td>3</td><td>2</td><td>1</td></tr></table>	2	4	3	2	1	\$BB999.99	<table border="1"><tr><td>\$</td><td></td><td>2</td><td>4</td><td>3</td><td>.</td><td>2</td><td>1</td></tr></table>	\$		2	4	3	.	2	1	
2	4	3	2	1												
\$		2	4	3	.	2	1									
<table border="1"><tr><td>2</td><td>4</td><td>3</td><td>2</td><td>1</td></tr></table>	2	4	3	2	1	\$00999.99	<table border="1"><tr><td>\$</td><td>0</td><td>0</td><td>2</td><td>4</td><td>3</td><td>.</td><td>2</td><td>1</td></tr></table>	\$	0	0	2	4	3	.	2	1
2	4	3	2	1												
\$	0	0	2	4	3	.	2	1								
<table border="1"><tr><td>1</td><td>1</td><td>3</td><td>4</td></tr></table>	1	1	3	4	99.99CR	<table border="1"><tr><td>1</td><td>1</td><td>.</td><td>3</td><td>4</td><td>C</td><td>R</td></tr></table>	1	1	.	3	4	C	R			
1	1	3	4													
1	1	.	3	4	C	R										
<table border="1"><tr><td>1</td><td>1</td><td>3</td><td>4</td></tr></table>	1	1	3	4	99.99CR	<table border="1"><tr><td>1</td><td>1</td><td>.</td><td>3</td><td>4</td><td></td><td></td></tr></table>	1	1	.	3	4					
1	1	3	4													
1	1	.	3	4												
<table border="1"><tr><td>2</td><td>3</td><td>7</td><td>6</td></tr></table>	2	3	7	6	99.99DB	<table border="1"><tr><td>2</td><td>3</td><td>.</td><td>7</td><td>6</td><td>D</td><td>B</td></tr></table>	2	3	.	7	6	D	B			
2	3	7	6													
2	3	.	7	6	D	B										
<table border="1"><tr><td>2</td><td>3</td><td>7</td><td>6</td></tr></table>	2	3	7	6	99.99DB	<table border="1"><tr><td>2</td><td>3</td><td>.</td><td>7</td><td>6</td><td></td><td></td></tr></table>	2	3	.	7	6					
2	3	7	6													
2	3	.	7	6												

b) Replacement characters

เป็นตัวอักษรใน picture ของ report item ทำหน้าที่ตัดเลขศูนย์ที่อยู่หน้าตัวเลขแล้ว แทนด้วย blanks ใน edited data item, ตัวอักษรพวกนี้ได้แก่ Z * \$ + -

- Z ตัวอักษร Z หนึ่งตัวทางซ้ายมือของ report item picture หมายถึงให้ตัดเลขศูนย์ แต่ ละตัวที่อยู่ข้างหน้าตัวเลขทั้งหมด แล้วแทนด้วย blanks สำหรับ edited data item ซึ่งมี Z หลาย ๆ ตัวติดกัน (Z's) อาจมี insertion character ตัวใดตัวหนึ่งจากชุด \$ + หรือ - นำหน้าก็ได้ และอาจจะมี insertion characters เช่น . , 0 หรือ B อยู่ในระหว่างฟิลด์ นั้นก็ได้
- * ใช้เหมือนกับตัวอักษร Z แต่แทนที่เลขศูนย์ที่ตัดทิ้งไปด้วยเครื่องหมาย *
- \$ dollar sign หนึ่งตัวใช้แทนเลขศูนย์ที่ตัดทิ้งไปหนึ่งตัว แต่อาจแทนเลขศูนย์ได้มากกว่าหนึ่งตัวก็ได้ นั่นคือ ให้ \$ หนึ่งตัวนำหน้าตัวเลขของ edited data item ที่ไม่ถูกตัดทิ้งไปและเลขศูนย์อื่น ๆ แทนด้วย blank โดยวิธีนี้ \$ จะเป็น floating sign
- + ถ้า edited data item มีค่าเป็นบวก หรือไม่มีเครื่องหมายให้ใส่เครื่องหมาย + หน้าตัว อักษรตัวแรกที่ไม่ถูกตัดทิ้ง ถ้ามีค่าเป็นลบให้ใส่เครื่องหมาย - ใน edited data item ในตำแหน่งเดียวกันนี้ ดังนั้น + จึงเป็น floating sign
- ถ้า edited data item มีค่าเป็นลบ ให้ใส่เครื่องหมายลบ หน้าตัวอักษรตัวแรกที่ไม่ถูก ตัดทิ้ง, ถ้ามูลค่าของ edited data item เป็นบวก เครื่องจะใส่ blank ให้แทนตำแหน่งนี้ ดังนั้น - จึงเป็น floating sign

ตัวอย่าง 1

Source data	Edited clause	Edited Item
0 0 9 2 3	zz999	9 2 3
0 0 9 2 3	zzz99	9 2 3
0 0 9 2 3	\$***.99	\$ * * . 9 . 2 3
0 0 0 8 2 4	\$\$\$\$.99	\$ 8 . 2 4
0 0 5 2 6	-9.99	- 5 . 2 6
3 2 6 5	\$\$\$.99	\$ 3 2 . 6 5
0 0 5 2	Z(3).Z(3)	. 0 5 2
0 0	Z(3).Z(3)	
1 3 2	\$ZZZ.9	\$ 1 3 . 2
1 3 2	\$ZZZ.Z	\$ 1 3 . 2
0 0 0 1	\$\$,\$\$Z.ZZZ	\$. 0 0 1

ตัวอย่าง 2

Data to be Edited	Picture of Report Item	Edited Item
0 1 2 3 4 5	ZZZ,999.99	1 2 , 3 4 5 . 0 0
0 0 1 2 3 4	Z99,999.99	0 0 , 0 1 2 . 3 4
0 0 0 1 2 3	\$ZZZ,ZZ9.99	\$ 1 . 2 3
0 0 0 0 1 2	\$ZZZ,ZZZ.99	\$ 1 2
0 0 1 2 3 4	\$***,**9.99	\$ * * 1 , 2 3 4 . 0 0
1 2 3 4 5 6	\$***,***.99	\$ 1 2 3 , 4 5 6 . 0 0
1 2 3 4 5 6	\$***,***.99	\$ * * * * * 1 . 2 3
0 0 0 0 1 2	-ZZZ,ZZZ	- 1 2
1 2 3 4 5 6	\$ZZZ,ZZ9.99CR	\$ 1 2 3 , 4 5 6 . 0 0 C R
0 0 1 2 3 4	\$(4),\$\$9.99	\$ 1 2 3 . 4 0
0 0 0 0 0 0	\$(4),\$\$\$.99	\$. 0 0
0 0 0 0 1 2	---,---.99	- . 1 2

สรุปการเลือกใช้ picture clause

ตาราง : แสดงชนิดของตัวอักขระที่ใช้ใน picture clauses

ชนิดของตัวอักขระ	symbol	ใช้เป็น
Field definition characters	9	Numeric field
	A	Alphabetic field
	X	Alphanumeric field
Numeric field special characters	V	Assumed decimal point
	P	Decimal scaling
	S	Operational (arithmetic) sign included
Editing characters	\$	Dollar sign
	Z	Zero suppression
	*	Check protection
		Decimal point
		Comma
	+	Plus sign
		Minus sign
	DB	Debit
	CR	Credit
	B	Blank insertion
	()	Zero insertion
/	Stroke insertion	

ตาราง : แสดงการแบ่งกลุ่มของข้อมูล (The five categories of data)

1. Numeric items ใน picture clause จะประกอบด้วยตัวอักษรต่อไปนี้เท่านั้น
 9 V S และ P
2. Alphabetic items ใน picture clause จะประกอบด้วยตัวอักษร A เท่านั้น
3. Alphanumeric items ใน picture clause จะประกอบด้วยตัวอักษร A 9 และ X
 แต่จะเป็นตัว A ทั้งหมด หรือ เลข 9 ทั้งหมดไม่ได้ต้อง
 ประกอบด้วยทั้งตัว A และ 9 ปนกัน หรือตัวอักษร X เพียง
 อย่างเดียวได้
4. Numeric edited items ใน picture clause จะประกอบด้วย combination ของตัวอักษร
 ต่อไปนี้
 B P V Z 0 9 , . + - CR DB \$ และ /
5. Alphanumeric edited items ใน picture clause จะประกอบด้วย combinations ของตัวอักษร
 ต่อไปนี้
 A X 9 B 0 และ /

แบบฝึกหัด

1. จงบอกขนาด (size) ของฟิลด์ต่อไปนี้

- | | |
|-----------------|------------------|
| a) 99V99 | g) VPP99 |
| b) 9(3).9 | h) S99.99 |
| c) S999V9 | i) ZZZ000 |
| d) ZZ,ZZZ | j) 99199199 |
| e) +(3).99 | k) XBXBXB |
| f) \$***,**9.99 | l) 9,999.99CR |

2. จงบอกเหตุผลว่าทำไม editing picture ข้างล่างนี้จึงใช้ไม่ได้

- a) 9zz.99
- b) CR999.99
- c) xxx-xx-xxx
- d) 999-99-999
- e) ***,*9
- f) Z,ZZZ.Z9
- g) 9,9\$\$\$.99

3. จงบอกค่าของผลลัพธ์ หลังจาก edited แล้วในแต่ละข้อข้างล่างนี้

Picture	Value of data	Edited result
zzzz.zz	0000.00	
**** **	0000.00	
zzzz.99	0000.00	
****.99	0000.00	
zz99.99	00000.00	
Z,ZZZ.ZZ+	+123.456	
*,***.**+	-123.45	
,,* **t	+12345678.9	
\$Z,ZZZ,ZZZ.ZZ CR	t 12345.67	
\$B*,***,***.** BBDB	-12345.67	

4. **จงบอกค่าของ receiving item จากตารางข้างล่างนี้**

Source data	Picture ของ source data	Picture ของ receiving data	ค่าของ receiving data
001056	9(4)V9(2)	\$Z,ZZZ.99CR	
- 347652	S9(4)V9(2)	\$\$,\$\$\$\$.99BDB	
- 7637	S9(3)V9	ZZZ.9 +	
- 007	S9(2)V9	- ZZ.9	
-0100	S9(4)	- - 9	
0000	9(4)	+ + , + + 9	
1234563	9(5)V9(2)	\$*,***.99	
000000	9(4)V9(2)	\$* ***.**	
01205	9(5)	\$ZZ,ZZZ.00	
00002	9(5)	\$\$\$,\$\$\$\$.00	

5. **ข้อความต่อไปนี้ถูกหรือผิด**

เฉพาะข้อมูลเดี่ยวเท่านั้นที่ต้องมี picture clause

(The picture clause can be used only with elementary items)

6. **ในการสอบไล่ครั้งหนึ่งกำหนดหลักเกณฑ์การให้เกรดไว้ดังนี้**

คะแนน	เกรด
80-100%	A
65-79.9	B
50-64.9	C
ต่ำกว่า 50	F

จงเขียนชื่อมีเงื่อนไขใน data division เพื่อสร้างหลักเกณฑ์นี้

7. **กำหนดให้ข้อมูลเกี่ยวกับจำนวนปีที่ได้รับการศึกษาเก็บในเนื้อที่ชื่อ EDUCATION มีค่าเป็นตัวเลขสองหลัก หมายถึงจำนวนปีที่จบการศึกษาสูงสุด, ตัวเลขที่บันทึกมีความหมายดังนี้**

น้อยกว่า 12	หมายถึง เรียนต่ำกว่าระดับมัธยมปลาย (high-school)
เท่ากับ 12	หมายถึง จบการศึกษาระดับมัธยมปลาย (high-school)
ระหว่าง 13-15	หมายถึง เรียนต่ำกว่าระดับวิทยาลัย (collage)

เท่ากับ 16 หมายถึง จบการศึกษาระดับวิทยาลัย (collage)
 มากกว่า 16 แต่ไม่เกิน 20 ปี หมายถึง จบการศึกษาระดับปริญญาตรี (graduate)
 ถ้าเป็นตัวเลขอื่นนอกเหนือจากนี้ แสดงว่ามีข้อผิดพลาดในการบันทึกข้อมูล
 ใช้นิยาม (define) เนื้อที่ในหน่วยความจำของเครื่องคอมพิวเตอร์ เพื่อให้มีความหมาย
 ดังกล่าวข้างต้นในลักษณะชื่อมีเงื่อนไข (condition-name)

8. จาก record description entries ข้างล่างนี้ picture ของแต่ละ item ใน OUTPUT-LINE มี
 editing symbol +, -, CR และ DB ซึ่งสัญลักษณ์เหล่านี้เครื่องคอมพิวเตอร์จะพิมพ์เครื่อง-
 หมายออกมาด้วย

01 NUMBER PICTURE S9(4).

01 OUTPUT-LINE.

02 A PICTURE +9(4).

02 FILLER PICTURE XX VALUE SPACES.

02 B PICTURE 9(4)+.

02 FILLER PICTURE XX VALUE SPACES.

02 C PICTURE -9(4).

02 FILLER PICTURE XX VALUE SPACES.

02 D PICTURE 9(4)-.

02 FILLER PICTURE XX VALUE SPACES.

02 E PICTURE Y(4) CR.

02 FILLER PICTURE XX VALUE SPACES.

02 F PICTURE 9(4)DB.

a) สมมติ NUMBER มีค่าเท่ากับ -276 แล้วให้เครื่องคอมพิวเตอร์ execute คำสั่ง
 MOVE NUMBER TO A, B, C, D, E, F.
 DISPLAY OUTPUT-LINE.

เครื่องจะพิมพ์ผลลัพธ์อะไร ?

b) ถ้า NUMBER มีค่าเท่ากับ +276 หลังจาก execute สองคำสั่ง เช่นเดียวกับข้อ (a)
 เครื่องจะพิมพ์ผลลัพธ์อะไร ?

เฉลยข้อ 2

- a) Z's จะปรากฏทางตำแหน่งขวามือของ picture 9 ไม่ได้
- b) CR จะต้องเป็น symbol ทางขวามือสุดของ item เท่านั้น
- c) จะมี hyphen ใน editing clause ไม่ได้
- d) จะมี hyphen ใน editing clause ไม่ได้
- e) *s จะอยู่ในตำแหน่งทางขวามือของจุดทศนิยมไม่ได้ เว้นไว้แต่ว่า ทุกตำแหน่งใน picture นั้นจะเป็น * ทั้งหมด เช่น *** **
- f) Z's จะอยู่ในตำแหน่งทางขวามือของจุดทศนิยมไม่ได้ เว้นไว้แต่ว่าทุกตำแหน่ง ใน picture นั้นเป็น Z ทั้งหมด
- g) \$ จะอยู่ในตำแหน่งทางขวามือของ picture 9 ไม่ได้

REDEFINES clause

ใช้สำหรับกำหนดให้ items สองตัว ใช้เนื้อที่ที่เดียวกันในหน่วยความจำ โดยมีรูปแบบดังนี้

level number data-name-1 REDEFINES data-name-2

(หมายเหตุ เราได้ฟิลด์ใหม่ชื่อ data-name-1) โดยมีข้อจำกัดดังนี้

- 1) เลขบอกระดับของ data-name-1 กับเลขบอกระดับของ data-name-2 ต้องเหมือนกัน และต้องไม่ใช่เลข 66 หรือ เลข 88
- 2) data-name-1 ต้องนิยาม (defines) หลัง data-name-2
- 3) data-name-1 อาจจะเป็นข้อมูลกลุ่ม หรือข้อมูลเดี่ยวก็ได้
- 4) data-name-1 ต้องมีขนาดเท่ากับ data-name-2
- 5) REDEFINES clause จะนำมาใช้กับเลขบอกระดับ 01 ใน file section ไม่ได้
- 6) record description entry ของ data-name-2 จะมี OCCURS clause อยู่ด้วยไม่ได้

ตัวอย่าง 1

03 A PICTURE 9(4).

03 B REDEFINES A PICTURE 9(4).

หมายความว่า เรามี data-name ชื่อ A สมมติมีค่าเท่ากับ 1234 หลังจากกำหนด B ขึ้นมาโดยใช้ REDEFINES clause, item B จะมีมูลค่าเท่ากับ 1234 เช่นกัน (ในที่นี้ A ต้องมีเลขบอกระดับเหมือนกับเลขบอกระดับของ B และทั้งสองฟิลด์ต้องมีขนาดเท่ากันด้วย)

1	2	3	4
---	---	---	---

ตัวอย่าง 2

01 CARD-REC.

03 YR-CARD.

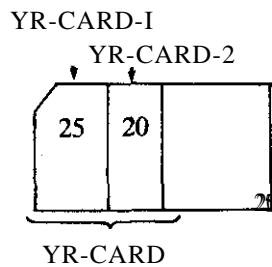
05 YR-CARD-I PICTURE 9(2).

05 YR-CARD-2 PICTURE 9(2).

03 YR-CARD-COPY REDEFINES YR-CARD PICTURE 9(4).

03 FILLER PICTURE X(76).

แสดงด้วยรูปดังนี้



เรคคอร์ดนี้ชื่อ CARD-REC.

ในที่นี้ YR-CARD เป็นข้อมูลกลุ่ม มีค่าเท่ากับ 2520

YR-CARD-COPY เป็นข้อมูลเดี่ยว ในที่นี้จะมามีค่าเท่ากับ 2520 เช่นกัน

ตัวอย่าง 3

02 HOURLY-RATE PICTURE 99V9999.

02 WEEK-SALARY REDEFINES HOURLY-RATE PICTURE 9999V99.

ถ้าเนื้อหาในหน่วยความจำของเครื่องคอมพิวเตอร์เก็บเลข

0	2	0	0	0	0
---	---	---	---	---	---

เวลาเรียก data-name มาใช้ HOURLY-RATE จะมีค่าเท่ากับ 02.0000 แต่ WEEK-SALARY จะมีค่าเท่ากับ 0200.00

ตัวอย่าง 4 ถ้าให้ A มีค่าเท่ากับ 1234 แล้วเรานิยาม data description entries ข้างล่างนี้

02 A PICTURE 9999.

02 B REDEFINES A PICTURE 9V999.

02 C REDEFINES B PICTURE 99V99.

หลังจาก execute คำสั่ง ADD A, B, C GIVING D.

ให้ D มี picture เท่ากับ 9(4)V9(2) จึงบอกค่าภายใน receiving item D (คำตอบ 1247.57)

OCCURS clause

ใช้ในกรณีที่ต้องการสร้างตารางในหน่วยความจำ เพื่อเก็บข้อมูลที่มีจำนวนมากๆ แต่มีข้อจำกัดว่ากำหนดได้เพียงสามมิติ (dimensions) เท่านั้น โดยมีรูปแบบดังนี้

OCCURS [integer-1 **TO**] integer-2 **TIMES** [**DEPENDING** ON data-name-1]

integer-1 แทนจำนวนครั้งต่ำสุดที่เกิด item นั้น

integer-2 แทนจำนวนครั้งสูงสุดที่เกิด item นั้น

integer-1 อาจเป็นศูนย์ได้ หมายความว่า ไม่มี item นั้น ถ้า integer-1 มีค่าเท่ากับ 1 หมายความว่า มี item นั้นและมีเพียงค่าเดียว

data-name-1 ต้องเป็นข้อมูลเดี่ยว และมีเพียงค่าเดียวเท่านั้น (unigue) จะมี subscripted หรือใช้ usage clause COMPUTATIONAL-1 ไม่ได้ ต้องเป็นเลขจำนวนเต็มบวก มีค่าระหว่าง integer-1 ถึง integer-2

หมายเหตุ

ข้อมูลอิสระซึ่งใช้เลขบอกระดับ 77 ใน working-storage section หรือใน constant section และ item ที่ใช้เลขบอกระดับ 01 หรือ เลขบอกระดับ 88 ใน file section จะนิยามด้วย OCCURS clause ไม่ได้

ตัวอย่าง ข้อมูลหนึ่งมิติ (1 dimensional array)

WORKING-STORAGE SECTION.

01 PRODUCT-NUMBER-TABLE.

02 PRODUCT-NUMBER PICTURE 9(5) OCCURS 5 TIMES

สมมติว่าเรามี item อยู่ทั้งหมด 5 ตัวคือ

Product number table

00105

00237

00238

00425

10004

ในที่นี้ PRODUCT-NUMBER (1) มีค่าเท่ากับ 00105
 PRODUCT-NUMBER (2) มีค่าเท่ากับ 00237

PRODUCT-NUMBER (5) มีค่าเท่ากับ 10004

ตัวอย่าง

อะเรย์หนึ่งคอลัมน์ข้างล่างนี้ชื่อ ACCOUNTS

100

150

75

750

เราเตรียมเนื้อที่เก็บข้อมูลใน working-storage section ดังนี้

01 ACCOUNTS.

02 BALANCE PICTURE 999 OCCURS 5 TIMES.

ตัวอย่าง

01 TIME-RECORD.

05 ID-NUMBER PICTURE 9(10).

05 H-WORKED OCCURS 7 TIMES PICTURE 9(5).

ให้นักศึกษาวาดรูปแสดงรายละเอียดของระเบียบชื่อ TIME-RECORD

ตัวอย่าง ข้อมูลสองมิติ (2 dimensional array)

เครื่องเก็บข้อมูลที่ละแถว (by rows)

Months

	1	2	3	4	5	6
1	100	175	300	450	500	750
2	150	150	100	75	100	150
3	75	100	125	150	175	200
4	200	250	300	500	750	900
5	750	700	650	400	500	450

เราเตรียมเนื้อที่เก็บข้อมูลใน working-storage section ได้ดังนี้

01 ACCOUNTS.

02 EXPENSE OCCURS 5 TIMES.

03 BALANCE PICTURE 999 OCCURS 6 TIMES.

ในที่นี้ BALANCE (11) = 100, BALANCE (13) = **300**

BALANCE (3 1) = 75, BALANCE (5 6) = 450

ตัวอย่าง Mixed arrays

ตาราง : Account Numbers และ Balances

Account Number	Months					
	1	2	3	4	5	6
602	100	175	300	450	500	750
610	150	150	100	75	100	150
612	75	100	125	150	175	200
620	200	250	300	500	750	900
625	750	500	650	400	500	450

เตรียมเนื้อที่เก็บข้อมูลดังนี้

แบบที่ 1 (ให้นักศึกษาวาดรูปแสดงรายละเอียดของเนื้อที่ ชื่อ ACCOUNTS)

01 ACCOUNTS.

02 ACCT-NUMBER PICTURE 999 OCCURS 5 TIMES.

02 EXPENSE OCCURS 5 TIMES.

03 BALANCE PICTURE 999 OCCURS 6 TIMES.

ในที่นี้

ACCT-NUMBER (1) = 602

ACCT-NUMBER (2) = 610

ACCT-NUMBER (5) = 625

ส่วน BALANCE (1 1) = **100**

BALANCE (12) = 175

BALANCE (16) = 750

แบบที่ 2 (ให้นักศึกษาวาดรูปแสดงรายละเอียดของเนื้อหาหน่วยความจำ ชื่อ ACCOUNT-SET)

01 ACCOUNT-SET.

02 ACCOUNT OCCURS 5 TIMES.

03 ACC-NUMBER PICTURE 999.

03 BALANCE PICTURE 999 OCCURS 6 TIMES.

RENAMES clause

clause นี้ทำให้โปรแกรมเมอร์สามารถจัดกลุ่มใหม่ให้กับข้อมูลเดี่ยว (regrouping elementary data item) ลักษณะคล้าย REDEFINES clause ยกเว้นกลุ่มใหม่นี้ ประกอบด้วยหลาย item รวมกัน, RENAMES clause ต้องใช้เลขบอกระดับ 66 เสมอ โดยมีรูปแบบดังนี้

66 data-name-1 **RENAMES** data-name-2 [**THRU** data-name-3]

ตัวอย่าง 1 จงวาดรูปแสดงรายละเอียดของ record description entries และการ renames เนื้อที่ข้างล่างนี้

01 OUT-REC.

02 FIELD-X.

03 SUMMARY-GROUP-X.

04 A PICTURE X.

04 B PICTURE X.

04 C PICTURE X.

02 FIELD-Y.

03 SUMMARY-GROUP-Y.

04 A PICTURE X.

04 B PICTURE X.

04 C PICTURE X.

02 FIELD-Z.

03 SUMMARY-GROUP-Z.

04 A PICTURE X.

04 B PICTURE X.

04 C PICTURE X.

- 66 SUM-X RENAMES FIELD-X.
- 66 SUM-XY RENAMES FIELD-X THRU FIELD-Y.
- 66 SUM-XYZ RENAMES FIELD-X THRU FIELD-Z.

ตัวอย่าง 2

01 TAXRECORD.

02 SOCIAL-NUMBER PICTURE 9(9).

02 NAME.

03 FIRST-NAME PICTURE X(10).

03 INITIAL PICTURE XX.

03 LAST-NAME PICTURE X(15).

02 TOTAL-YEAR-TO-DATE.

03 GROSS-PAY PICTURE 9(8)V99.

03 NET-PAY PICTURE 9(8)V99.

03 FED-PAY PICTURE 9(6)V99.

03 STATE-TAX PICTURE 9(4)V99.

66 LAST-GROSS RENAMES LAST-NAME THRU NET-PAY ,

การจัดกลุ่มข้อมูลใหม่ โดยใช้ RENAMES clause ตามตัวอย่างข้างต้น แสดงโดยรูปข้างล่างนี้ที่เป็นเส้นปะ

SOCIAL-NUMBER	NAME			TOTAL-YEAR-TO-DATE			
	FIRST-NAME	INITIAL	LAST-NAME	GROSS-PAY	NET-PAY	FED-PAY	STATE-PAY

LAST-GROSS

ในที่นี้ LAST-GROSS เป็นเนื้อที่หนึ่งซึ่งประกอบด้วยฟิลด์ LAST-NAME, GROSS-PAY และ NET-PAY ซึ่งเรานำไปใช้ได้และถือว่าเป็นข้อมูลกลุ่ม ซึ่งเราจะทำเช่นนี้ไม่ได้เลย ถ้าไม่มี RENAMES clause ใช้

VALUE clause

clause นี้ใช้สำหรับนิยามค่าแรกของ working-storage item หรือค่าของชื่อมีเงื่อนไข

2 รูปแบบคือ

(1) VALUE IS literal

(2) $\left\{ \begin{array}{l} \text{VALUE IS} \\ \text{VALUES ARE} \end{array} \right\}$ literal-1 [THRU literal-2] [literal-3 [THRU literali]]..

ในที่นี้ literal อาจจะเป็น figurative constant ได้

กฎการใช้ VALUE clause จะแตกต่างกันไปแล้วแต่ว่ามันอยู่ใน section ไหนของ Data division

1. ใน File section และ Linkage section, VALUE clause ใช้ได้เฉพาะใน condition-name entries เท่านั้น

2. ใน Working-storage section, VALUE clause ใช้ใน condition-name entries และอาจเอาไปใช้กำหนดค่าแรก (initial value) ของข้อมูลใด ๆ ก็ได้ และทำให้เนื้อที่นั้นมีค่าก่อนการ execute ของ object program

ใน data description entry ถ้าไม่ได้ใช้ VALUE clause ค่าแรกนั้นจะไม่ทราบค่า (unpredictable)

ใน data description entry ที่มี OCCURS clause หรือ entry ซึ่งเป็นส่วนย่อยของ entry ที่มี OCCURS clause จะมี VALUE clause ไม่ได้

ใน data description entry ที่มี REDEFINES clause หรือ entry ซึ่งเป็นส่วนย่อยของ entry ที่มี REDEFINES clause จะมี VALUE clause ไม่ได้

ถ้าใช้ VALUE clause ใน entry ซึ่งเป็นข้อมูลกลุ่ม (group item) literal ในที่นี้ ต้องเป็น figurative constant หรือ non-numeric literal และ ในส่วนย่อยของข้อมูลกลุ่มนี้ จะมี VALUE clause ไม่ได้

รูปแบบที่ 1 VALUE clause ต้องไม่ขัดแย้งกับ clauses อื่น ๆ ใน data description ของ item ตัวนี้

รูปแบบที่ 2 VALUE clause ใช้บรรยายชื่อมีเงื่อนไข ซึ่งมีเลขบอกระดับ 88

สำหรับข้อมูลกลุ่มซึ่งมี item ส่วนย่อยบรรยายด้วย JUSTIFIED, หรือ USAGE clause จะมี VALUE clause ไม่ได้

มีกฎต่าง ๆ ดังนี้

1. ถ้า item นั้นเป็น numeric, literals ทั้งหมดใน VALUE clause ต้องเป็น numeric literals ด้วย ถ้า literal นั้น defines ค่าของ working-storage item, literal จะถูกจัดเก็บในหน่วยความจำตามกฎของการย้ายตัวเลข (numeric move)

2. ถ้า item นั้น เป็น alphabetic หรือ alphanumeric (ทั้งข้อมูลเดี่ยวหรือข้อมูลกลุ่ม), literals ทั้งหมด ใน VALUE clause ต้องเป็น non-numeric literals ด้วย

3. numeric literals ทุกตัวใน VALUE clause ของแต่ละ item จะต้องมีค่าอยู่ในช่วงของค่า (range of values) ที่กำหนดโดย PICTURE clause ของ item ตัวนั้น

ตัวอย่าง

```
03 TEMP1 PICTURE 99PPP.
```

ถ้ากำหนด VALUE clause ให้กับ TEMP1, literal ต้องมีค่า ในช่วง 1000 ถึง 09000 หรือ zero

ตัวอย่าง

```
03 TEMP2 PICTURE PPP99.
```

ถ้ากำหนด VALUE clause ให้กับ TEMP2, literal ต้องมีค่าในช่วง .00000 ถึง .00099

4. หน้าทีของตัวอักษรที่ใช้เพื่อการ edit ใน PICTURE clause จะถูก ignored ในการกำหนดค่าแรกของ item อย่างไรก็ตาม ตัวอักษรที่ใช้เพื่อการ edit จะนับรวมอยู่ในขนาดของ item นั้น

ตัวอย่าง 1

```
01 A.
```

```
03 B PICTURE X(3) VALUE "YES".
```

```
03 C PICTURE X(2) VALUE SPACES.
```

ตัวอย่าง 2

```
77 SUM-SCORE PICTURE 9(3) VALUE ZEROS.
```

```
77 AVERAGE PICTURE 9(3)V9(2) VALUE ZEROS.
```

ตัวอย่าง 3

02 ITEM-A PICTURE 999V99 VALUE 123.45.

02 ITEM-B PICTURE A(24) VALUE "ACCUMULATED DEPRECIATION",

02 ITEM-C PICTURE X(10) VALUE "AMOUNT-S".

02 TAX-RATE PICTURE V99 VALUE 0.03.