บทที่3 ระเบียน

3.1 บทนิยาม

3.2 ระเบียนใน COBOL และ Pascal

แบบฝึกหัด

IT 204 61

บทที่ 3
ระเบียน
(Record)

ระเบียน คือ โครงสร้างข้อมูลอย่างง่ายที่มีความสำคัญ
ระเบียน คือ ส่วนประกอบพื้นฐานของแฟ้มและฐานข้อมูล
(Records are the basic components of files and of databases.)

3.1 บทนิยาม (Definition)

ระเบียน หมายถึงกลุ่มจำกัดแบบอันดับของสมาชิกซึ่งมีแบบชนิดข้อมูลไม่เหมือนกันที่ถือราวกับว่าเป็นหนึ่งหน่วย
(A **record** is a finite, ordered collection of possibly heterogeneous elements that are treated as a unit.)
ระเบียนแตกต่างจากแถวลำดับตรงที่สมาชิกทั้งหมดในแถวลำดับต้องมีโครงสร้างเหมือนกัน แต่สมาชิกส่วนประกอบของระเบียนอาจมีโครงสร้างข้อมูลแตกต่างกันบางครั้งระเบียนอาจเรียกว่า โครงสร้าง (a structure) สมาชิกของระเบียนเรียกว่า **เขตข้อมูล** (fields)
เขตข้อมูล คือเนื้อที่เฉพาะของระเบียนใช้สำหรับชนิดสารสนเทศโดยเฉพาะ (A field is a specified area of a record used for a particular kind of information.)

การประกอบกันเป็นระเบียน (Forming Records)

หนึ่งหน่วยของสารสนเทศได้มาจากอย่างน้อยที่สุดความหมายของมันจากบางสิ่งซึ่งเป็นความสัมพันธ์ของมันกับสารสนเทศอื่น
โครงสร้างข้อมูลระเบียนทำให้เซตของสมาชิกของสารสนเทศซึ่งเกี่ยวข้องกันเชิงตรรกะสามารถเป็นกลุ่มได้อย่างชัดแจ้ง
ตัวอย่าง กลุ่มข้อมูลซึ่งประกอบขึ้นเป็นระเบียนของสารสนเทศของพนักงาน

JOB – TITLE	EMP – NO	PAY – RATE
ANALYST	123456789	15.93

รูป 3-1 ระเบียนตัวอย่าง

ตัวอย่าง

สมาชิกแต่ละตัวของระเบียนอาจเป็นโครงสร้างข้อมูลผลประกอบ เช่น เป็นอีกหนึ่งระเบียนหรือแถวลำดับ ตัวอย่างเช่น ระเบียนของพนักงานประกอบด้วยเขตข้อมูล NAME และ OFFICE – ADDR ในรูป 3-2

JOB - TITLE	EMP - NO	PAY-RATE	NAME			OFFICE -ADDR	
ANALYST	123456789	15.93	LAST	FIRST	INIT	BLDG	ROOM
			MCCALL	JOE	N	CSC	403

รูป 3 -2 ระเบียนตัวอย่าง

สมาชิกของระเบียนอาจเป็นแถวลำดับของ PROJECT – CODE

JOB-TITLE	EMP-NO	PAY-RATE	NAME			OFFICE- ADDR		PROJECT- CODE				
ANALYST	123456789	15.93	LAST	FIRST	INIT	BLDG	ROOM	18	40	41	50	53
			MCCAL	JOE	N	CSC	403					

รูป 3-3 ระเบียนตัวอย่างประกอบด้วยแถวลำดับ

เขตข้อมูลซึ่งไม่มีเขตข้อมูลย่อยเรียกว่า **ข้อมูลเบื้องต้น**

(A field with no subordinate fields is called and **elementary item.**)

เขตข้อมูลที่มีข้อมูลย่อยเรียกว่า **ข้อมูลกลุ่ม**

(A field with subordinate field is called **group item.**)

จากตัวอย่างข้างต้น NAME และ OFFICE-ADDR เป็นข้อมูลกลุ่ม

โครงสร้างข้อมูลระเบียน EMPLOYEE ประกอบด้วยเขตข้อมูลชื่อ JOB-TITLE, EMP-NO, PAY-RATE,NAME, OFFICE-ADDR, และ PROJECT-CODE

แต่ละเขตข้อมูลมีโครงสร้างข้อมูลของตนเอง ตัวแปร EMPLOYEE มีค่ามากมายตลอดเวลา

การกำหนดคีย์ (Identifying Keys)

โดยปกติหนึ่งระเลียนกำหนดโดยชื่อหนึ่งชื่อดังตัวอย่างข้างต้น การเกิดของระเบียน IMPLOYEE ของพนักงานระบุโดยค่าในเขตข้อมูล EMP-NO

เขตข้อมูลที่กำหนดถึงระเบียนเรียกว่า key field (The identifying field of a record is called it's **key**

คีย์ของระเบียนอาจเป็นข้อมูลเบื้องต้นหรือข้อมูลกลุ่มก็ได้

ตัวอย่างเช่น ระเบียน COURSE ของชั้นเรียนหนึ่งอาจกำหนดโดยค่าของเขตข้อมูล DEPARTMENT , NUMBER และ SECTION

แฟ้ม (File)

กลุ่มของระเบียนที่เกี่ยวข้องกันเชิงตรรกะซึ่งถือว่าเป็นหนึ่งหน่วยเรียกว่า **แฟ้ม**

(A collection of logically related record occurrences that are treated as a unit is called a **file**.)

โดยปกติ ระเบียนทั้งหมดของแฟ้มมีรูปแบบระเบียนเพียงอย่างเดียวแต่บางครั้งบนหนึ่งแฟ้มอาจมีรูปแบบระเบียนหลายรูปแบบได้

ตัวอย่างเช่น ระเบียน EMPLOYEE อาจตามด้วยระเบียนชนิด PROJECT-RESPONSIBILITY หลายระเบียนแต่ละชุดมีรายละเอียดเกี่ยวกับความพยายามของพนักงานบนโครงการเฉพาะเรื่อง การปฏิบัติการที่ถูกต้องบนระเบียนได้แก่การปฏิบัติการซึ่งนิยามบนเขตข้อมูลของระเบียนขึ้นอยู่กับโครงสร้างข้อมูลของมัน

3.2 ระเบียนใน COBOL และ Pascal (Records in COBOL and Pascal)

ขณะนี้ให้พิจารณาสิ่งอำนวยความสะดวกที่ใช้ในการประกาศตัวแปรระเบียนใน COBOL และ Pascal

COBOL

ภาษา COBOL ใช้ระบบของ level – numbers เพื่อแสดงถึงโครงสร้างของระเบียน การอธิบายระเบียนเริ่มต้นด้วยชื่อระเบียนซึ่งต้องอยู่ที่ระดับ 01 เสมอ หลังจากนั้นข้อกำหนดรายละเอียดสำหรับเขตข้อมูลต่าง ๆ ประกอบกันเป็นระเบียน

ตัวอย่าง

```
01 EMPLOYEE.
     02 JOB-TITLE      PICTURE    X(7).
     02 EMP-NO         PICTURE    X(9).
     02  EMP-RATE      PICTURE    X(2)V9(2).
```

```
02  NAME.
        03    L-NAME              PICTURE     X(12).
        03    F-NAME              PICTURE     X(18).
        03    INIT                PICTURE     X.
02  OFFICE-ADDR.
        03    BLDG                PICTURE     X(3).
        03    ROOM                PICTURE     9(3).
02  PROJECT-CODES OCCURS 2 TIMES PICTURE 9(2).
02  TELEPHONE- NO  OCCURS 2 TIMES.
        03    AREA-CODE           PICTURE     9(3).
        03    LOCAL               PICTURE     9(7).
```

การอธิบายข้อมูลแต่ละตัวให้นำหน้าด้วย level-number ซึ่งเอามาจากเซต {01,02,03,.. ..,49}

การจัดกลุ่มของ level-numbers เหล่านี้อธิบายโครงสร้างเชิงลำดับชั้นของเขตข้อมูลต่าง ๆ ในระเบียน

เมื่อตัวเลข level – number เพิ่มขึ้นจาก 01 ไปเป็น 02 ระหว่าง EMPLOYEE และ JOB-TITLE และจาก 02 ไป 03 ระหว่าง NAME กับ L- NAME ระหว่าง OFFICE –ADDR กับ BLDG และระหว่าง TELEPHONE-NO กับ AREA-CODE แสดงถึงการแบ่งย่อยของข้อมูลกลุ่มไปเป็นเขตข้อมูลที่เป็นองค์ประกอบ

ข้อมูลกลุ่ม EMPLOYEE ประกอบด้วย JOB-TITLE, EMP-NO, PAY-RATE, NAME, OFFICE-ADDR, PROJECT-CODES และ TELEPHONE-NO

ข้อมูลกลุ่ม TELOPHON-NO เป็นแถวลำดับมีสมาชิกสองตัวแต่ละตัวประกอบด้วย AREA-CODE และ LOCAL โปรดสังเกตว่า ข้อมูลเบื้องต้น (elementary items) จะต้องมี PICTURE clauses เสมอ

ลำดับชั้นที่ลึกมากขึ้นใน COBOL การใช้ level-number ต้องเป็นเลขมีค่ามากขึ้น

ตัวอย่าง

```
02 TELEPHON-NO OCCURS 2 TIMES.
    03  AREA-CODE  PICTURE      9(3).
    03  LOGICAL.
        04  EXCHANGE       PICTURE      9(3).
        04  NUMB           PICTURE      9(4).
```

ภาษา PL/1 ใช้ level-numbers ในลักษณะเดียวกับที่ COBOL ทำยกเว้น PL/1 ใช้ level 1 แทน level 01 และ level 2 แทน level 02 เช่นนี้เรื่อยไป

Pascal

ภาษา Pascal ใช้สิ่งอำนวยความสะดวกการจัดกลุ่มไม่ใช่ข้อตกลง level –number เพื่อนิยามโครงสร้างระเบียน ระเบียนแต่ละชุดแยกจากกันโดยแต่ละชิดอยู่ใน **record** และ **end** เพื่อแสดงการจัดกลุ่มของสมาชิกข้อมูล

```
Type    name = record
                    Lname: packed array[1..12] of char;
                    Fname: packed array [1..8] of char;
                    Init : char;
               end.
        Addr = record
                    Bldg: packed array[1..3] of char;
                    Room: integer;
              end;
        Localphone = record
                    Exchange: integer;
                    Numb: integer;
                   end;
        Phoneno =  record
                    Areacode: integer;
                    Local: localphone
                  end;
        Employeerec = record
                    Jobtitle: packed array[1..7] of char;
                    Empno: packed array[1..9] of char;
                    Payrate: real;
```

```
        Empname: name;
        Officeaddr: addr;
        Projectcodes: array[1..6] of integer;
        Telephoneno: array[1..2] of phoneno;
    end;
```

ตัวอย่างนี้แสดงถึงบทนิยามของ new data type คือ name, addr, localphone, phoneno และ employses

ข้อความสั่ง **type** อาจตามด้วยบทนิยามของตัวแปรด้วยโครงสร้างซึ่งนิยามมาแล้ว เช่น

```
var employee : employeesec;
```

Qualified Names

การอ้างถึงสมาชิกตัวใดตัวหนึ่งของโครงสร้างระเบียนต้องใช้ชื่อตัวแปรที่กำหนดเป็นได้อย่างเดียว(unique) ถ้าชื่อเขตข้อมูลไม่เป็นหนึ่งอย่าง จำเป็นต้องมีคุณสมบัติบางอย่างนำมาใช้เพื่อทำให้ชื่อนั้นเป็นได้อย่างเดียว

ตัวอย่างเช่น จงพิจารณาโครงสร้างระเบียน COBOL ข้างล่างนี้

```
01  STUDENT.
    . . .
    02 BIRTH- DATE.
        03 MM        PICTURE      99.
        03 DD        PICTURE      99.
        03 YY        PICTURE      99.
    . . .
    02 UNDERGRAD.
        03    MATRIC-DATE.
              04 MM        PICTURE      99.
              04 DD        PICTURE      99.
              04 YY        PICTURE      99.
        03    DEGREE-DATE.
              04 MM        PICTURE      99.
              04 DD        PICTURE      99.
```

```
                04 YY          PICTURE      99.
    . . .
02  GRAD.
    03  MATRIC-DATE.
        04 MM          PICTURE      99.
        04 DD          PICTURE      99.
        04 YY          PICTURE      99.
    03  DEGREE-DATE.
        04 MM          PICTURE      99.
        04 DD          PICTURE      99.
        04 YY          PICTURE      99.
    . . .
```

ในที่นี้ data name YY ไม่เป็นได้อย่างเดียว (not unique) แต่สามารถมีคุณสมบัติเพื่อแสดงว่าเป็น YY ตัวไหนในโครงสร้างที่ต้องการอ้างถึง ตัวอย่างเช่น

YY OF BRITH-DATE

แต่ YY OF MATRIC-EATE ไม่เป็นได้อย่างเดียวเพราะฉะนั้นจึงต้องมีคุณสมบัติเพิ่มอีก เช่น

YY OF MATRIC-DATE OF UNDERGRAD

หรือ YY OF MARTIC-DATE OF GRAD

ชื่อตัวแปรทีมีคุณสมบัติเต็ม (A fully qualified variable name) ต้องอ้างย้อนกลับไปจนถึงระดับ 01

ตัวอย่างเช่น

YY OF MATRIC-DATE OF GRAD OF STUDENT

ภาษา Pascal ใช้หลักเกณฑ์แตกต่างเล็กน้อยสำหรับการให้คุณสมบัติของชื่อตัวแปรจากตัวอย่างข้างต้น Pascal เขียนดังนี้

birthdate.yy

undergrad.matricdata.yy

grad.matricdate.yy

student.grad.matricdate.yy

ภาษา FORTRAN ไม่มีสิ่งอำนวยความสะดวกสำหรับการประกาศของตัวแปรระเบียน (record variables) ดังนั้นข้อมูลเบื้องต้นแต่ละตัวที่ประกอบเป็นระเบียนจึงต้องให้นิยามแยกกันการจัดกลุ่มเพื่อประกอบเป็นระเบียนจึงเป็นเพียงโดยนัย ตรรกะของโปรแกรมหรือเอกสารกำกับโปรแกรม (documentation) จึงควรจะแสดงว่ากลุ่มของเขตข้อมูลต่าง ๆ นั้นที่จริงประกอบเป็นระเบียน

แบบฝึกหัด

1. อะไรคือความแตกต่างระหว่างแถวลำดับและระเบียน แถวลำดับเป็นส่วนหนึ่งของระเบียนได้หรือไม่, ระเบียนเป็นส่วนหนึ่งของแถวลำดับได้หรือไม่
2. ทำไมโปรแกรมเมอร์จึงแบ่งหนึ่งเขตข้อมูลให้เป็นชิ้นส่วนประกอบเล็ก ๆ
3. โปรแกรมสองโปรแกรมเข้าถึงระเบียนของแฟ้มเดียวกัน โปรแกรมหนึ่งถือว่า phoneno เป็นข้อมูลเบื้องต้น ในขณะที่อีกโปรแกรมหนึ่งถือว่า phoneno เป็นข้อมูลกลุ่ม ทำไมจึงขัดแย้งกัน
4. จากเนื้อหา (contents) ในบทนี้ทำไมภาษา FORTRAN จึงไม่ถือว่าเป็นภาษาโปรแกรมเชิงธุรกิจ (business-oriented programming language)