

表的建立和查找

一、 表的概念

COBOL 语言中的表大体相当与其它高级语言中的数组，但也不完全相同。

把具有相同属性的数据项按一定的逻辑顺序组织在一起，成为一个整体的数据组织，用一个统一的名字来代表它们，这就是“表”。

构成表的各数据项称为表元素。

在建立一个“表”以后，只要指出表名和序号（即指出表中的第几个元素）或表中的相对地址（相对于表中的第一个元素的字节地址），就可以唯一的确定一个表元素。序号称为“下标”，相对地址称为“位标”，“下标”和“位标”统称为“出现号”。

二、 表的建立 (SAMPLE9-1)

表的建立就是定义一个表。表名应在数据部中说明

例： 01 STUDENT-RECORD

02 NAMES PIC X(20)

02 COURSE OCCURS 5 TIMES PIC 9 (3) .

COURSE 就是一个表，这个表包含 5 个类型和地位相同的数据项。

表元素可以是初等项，也可以是组合项。

引用表元素是只需一个下标的，是一维表，需要两个下标的则是二维表。

可以直接从数据部的描述看出：如果在数据项描述体中有一个 OCCURS 子句，而在它的上属数据项的描述体中没有 OCCURS 子句，则它是一维表，反之，则为二维表。当然还可以定义三维表。

OCCURS 子句的最简单的格式为：

OCCURS 整数 TIMES

说明：（1）OCCURS 后面只能为整数。

（2）OCCURS 子句不能出现在 77 层，因为 77 层是独立的数据项，不和其它数据发生组合关系

（3）OCCURS 子句不能用于 01 层。“表”只能定义在记录内部，可以在 01 层下面增加一层（例如 02 层），把表定义在这一层上。

（4）如果 OCCURS 子句描述的是初等项，则重复出现的是初等项；如果描述的是组合项，则重复出现的是组合项。

（5）在 COBOL 中表元素可以是组合项。而且在建立一个多维表的同时，也建立了一维（或一、二维）表。这和其他语言的数组是不同的。

（6）只有当 OCCURS 所说明的数据是初等项时，才能在数据项的描述中使用 PIC 子句。

（7）不能用 VALUE 子句对表赋初值。

（8）多维表的元素在内存中是以行序排列的。

三、可变长表

可变长表的一般格式：

OCCURS 整数 1 TO 整数 2 TIMES DEPENDING ON 数据名 1

数据名 1 可以在表所在的记录中描述。也可以不在本记录中描述。但不能是表的一部分。

四、表元素的引用

“表”必须先在数据部定义，才能在过程部引用。

各个表元素是按一定规律在内存中顺序存放的，因此，在引用时必须指出表名和这个元素在表中的位置。

注意：(1) 如果已说明 B 是一个表，则不能直接引用表名 B 而不加下标。

例： MOVE B TO A. 是错误的。

(2) 如果表元素是组合项，则引用他下属的项（可以是初等项或组合项），也必须用下标指明它是属于哪一个表元素的。

(3) 如果表元素是组合项，可以用它对下属的数据项进行限制。

(4) 下标只能是整常数或具有整数值的数据名也可以是表达式。如果下标是数据名，则此数据名可以在数据部的文件节或工作单元节中定义，也可在表所在的记录中定义，但不能属于“表”的一部分。

(5) 标准 COBOL 规定，下标不能是带下标的数据名，即不能是表元素。

(6) 带下标的数据名又称下标数据名。它是标识符的一种形式。

五、给表元素赋初值 (SAMPLE9-2)

在工作单元节中给表元素赋初值的办法有两种：

1. 对包括所有表元素的整个表赋初值。这是可以对表的描述体上面一层的数据项赋一个初值即可。

例： 01 TABLE VALUE IS ZERO
03 A OCCURS 20 TIMES PIC 9(3)

2. 联合使用 OCCURS 和 REDEFINES 子句给表元素赋初值，它的步骤是：

(1) 先在工作单元节中定义一个组合项，它站的内存大小和需赋值的表一样。在该组合项中定义若干个数据项，数据项的描述和表元素的描述相同。

(2) 然后对这些项分别用 VALUE 子句赋值。

(3) 把这个组合项重定义成一个表。

例： 01 UNIT-PRICE-TABLE
03 FILLER PIC X(10) VALUE '0100001350'
03 FILLER PIC X(10) VALUE '0200001420'
01 UNIT-PRICE-TABLE-R REDEFINES UNIT-PRICE-TABLE
02 TABLE OCCURS 2 TIMES
03 PROD-CODE PIC X(4)
03 PROD-PRICE PIC X(6)

六、用位标法引用表元素

1. 位标的概念：“位标”是代表一个表元素在表中的相对位置。

2. 位标名的指定方法：作为位标的数据名要专门加以指定。在数据部中定义一个表时所用的 OCCURS 子句中要加上“INDEXED BY 位标名”短语来指定。

例： 01 A
02 TABLE OCCURS 5 TIMES INDEXED BY I.

这个位标不需要在任何其它地方对它进行数据描述，它是一种特殊类型的数据，专用于表元素的访问。

说明：(1) 凡以后需要用位标来引用表元素的，都需要在建立表时用“INDEXED BY ”短语来指定位标的名字。

(2) 由于位标是专用于引用表元素的特殊数据项，不能进行算术运算。也不能由输入语句来改变它的值，也不能将它的值直接打印或显示出来。也不用在工作单元节或文件节中定义。

(4) 一维或多维表可以按需要指定多个位标，引用时只能在该维中使用。

例： 01 A
02 B OCCURS 5 TIMES INDEXED BY B1,B2

04 C OCCURS 4 TIMES INDEXED BY C1, C2

05 D OCCURS 4 TIMES

INDEXED BY D1, D2, D3

B(B1), B(B2), C(B1, C1), C(B2, C1), D(B1, C1, D2), D(B2, C2, D3) 是合法的。

C(C1, D2), B(C1) 是不合法的。

- (5) 当需要把位标的值转存到另一数据项中时，该数据项需在数据部中定义，并用 USAGE 子句说明。

例： 77 A USAGE IS INDEX.

3. SET (设置) 语句

由于位标是特殊的数据类型，只能用 SET 语句对其赋值

例： SET I TO 10

意思是“将位标 I 置于第十个表元数第一个字节的相对位置上”，是将位标地址送过去。

设置语句的一般形式

$$\text{SET} \left\{ \begin{array}{l} \text{标识符 1 [, 标识符 2]...} \\ \text{位标 1 [, 位标 2]} \end{array} \right\} \text{ TO } \left\{ \begin{array}{l} \text{标识符 3} \\ \text{位标 3} \\ \text{整数} \end{array} \right\}$$

注意：不能用 SET 语句将一整数送到位标数据项中或将一般数值数据项或整数送到数值数据项中。

设置语句的格式二

$$\text{SET 位标 1 [, 位标 2]} \left\{ \begin{array}{l} \text{UP BY} \\ \text{DOWN BY} \end{array} \right\} \left\{ \begin{array}{l} \text{标识符} \\ \text{整数} \end{array} \right\}$$

例： SET I UP BY 2

SET I DOWN BY J

4. 使用位标引用表元素的方法

- (1) 使用位标有两种方法：一是在表明后的括号中直接写上位标，如 A(A1)；二是在括号中使位标增加或减少一个整数，如 A(A1 + 1)。
- (2) 位标所代表的“出现号”不能超过相应维的长度。如表 A 有十元素，则 A(A1 + 11)，其中 A1 当前指向第一个元素，则 A(A1 + 11) 执行第 12 个元素，就是错的。
- (3) 一个表无论是否指定位标，都可以使用下标。但不允许位标与下标混用，允许位标与常数混用。
- (4) 位标可以在 IF 语句中参加比较，即在参加比较时系统会自动将位标转换成相应的序号。

八、表的检索 (SAMPLE9-3)

1. 用于顺序检索的 SEARCH 语句

在无序表中检索数据只能一个一个的顺序检索，这叫顺序检索法。

SEARCH 语句的一般格式之一

$$\text{SEARCH 表名} \left[\text{VARYING} \left\{ \begin{array}{l} \text{位标名 1} \\ \text{标识符 2} \end{array} \right\} \right] \left[\text{AT END 强制语句 1} \right]$$
$$\left\{ \begin{array}{l} \text{强制语句 2} \end{array} \right\} \left[\left\{ \begin{array}{l} \text{强制语句 3} \end{array} \right\} \right]$$

WHEN 条件 1

NEXT SENTENCE

WHEN 条件 2

NEXT SENTENCE

注意：(1) 该语句顺序检索表中的各元素，当未找到满足 WHEN 条件的记录则转入 NEXT SENTENCE。SEARCH 语句检索的表必须用位标法引用。在 WHEN 条件中必须出现位标法引用的表元素。

例：SET I TO 1

SEARCHING.

SEARCH SHOP-PRODUCTION (表名)

AT END GO TO FINISH

WHEN QTY(I) < 500 DISPLAY NUM(I), QTY(I)

WHEN GRADE(I) = 'D' DISPLAY NUM(I)

SET I UP BY 1

GO TO SEARCHING.

(2) 如果不带 VARYING 可选项，则检索时使用定义表时的第一个位标，其它位标值不变；如果带 VARYING 可选项，则使用指定的位标，其它位标值不变；如果使用其它表的位表，则仍使用定义表时的第一个位标，两个位标值同时增长；当 VARYING 可选项指定的普通的数据项时，仍使用第一个位标，数据型和位标同时增长。

2. 用于有序表的 SEARCH 语句

(1) 折半检索法，最多检索次数位以 2 为底表长的对数。

(2) ASCENDING KEY 和 DESCENDING KEY

说明升降序的一般格式为：

$$\left\{ \begin{array}{l} \text{ASCENDING} \\ \text{DESCENDING} \end{array} \right\} \text{KEY IS 数据名 1 [, 数据名 2]...}$$

用折半法检索的表，表元素都已经按照一项（多项）严格按照升序（降序）排列好的。

例：01 BOOK

02 BOOK-TABLE OCCURS 100 TIMES

ASCENDING KEY IS NAME

INDEXED BY I

03 NAME PIC X(10)

03 CLASS PIC X(2)

03 CODE PIC X(4)

说明：在定义表时，计算机不会自动将表中的记录按照升序排列，而是在录入时就应严格按照升序录入，也就是说，折半查找的对象是已经排好序的表。

(3) SEARCH 语句格式二

SEARCH 语句的一般格式二

SEARCH ALL 表名 [AT END 强制语句 1] WHEN 条件 1 $\left\{ \begin{array}{l} \text{强制语句 2} \\ \text{NEXT SENTENCE} \end{array} \right\}$

例：SEARCH ALL BOOK-TABLE

```

AT END DISPLAY 'CAN NOT FIND THE NAME'
WHEN NAME(I) = 'COMPUTER BOOKS'
    DISPLAY NAME(I), NUM(I).

```

说明：(1) “条件” 中值允许使用关系运算符 “EQUAL TO” 或 “=”

(2) 作为比较条件的一方，必须是 KEY 中的表元素，并且必须在其后的括号中指定在 “INDEXED BY” 字句中出现的位标。

(3) 系统会自动找到中间位置的元素，不必用 SET 语句指定。

九、用 PERFORM 语句对表进行检索 (SAMPLE9-4)

利用 PERFORM 可以实现多重循环的功能，在多维表中，按照检索条件，对表中的元素进行逐一的顺序检索。

例：WORKING-STORAGE SECTION

```

77 SCHOOL-IX PIC 9(2)
77 CLASS-IX PIC 9(2)
77 STUDENT-IX PIC 9(2)
01 TABLE
    02 SCHOOL OCCURS 10
    03 CLASS OCCURS 5
    04 STUDENT OCCURS 40
        05 NAME PIC X(20)
        05 NUM PIC 9(5)
        05 ADDR PIC X(20)
        05 AGE PIC 9(3)

```

PROCEDURE DIVISION.

...

```

PERFORM SELECTION THRU SELECTION-END
    VARYING SCHOOL-IX FROM 1 BY 1
        UNTIL SCHOOL-IX > 10
    AFTER CLASS-IX FROM 1 BY 1
        UNTIL CLASS-IX > 5
    AFTER STUDENT-IX FROM 1 BY 1
        UNTIL STUDENT-IX > 40

```

...

SELECTION.

```

IF NAME(SCHOOL-IX, CLASS-IX, STUDENT-IX) = 'ZHANG MING'
    AND AGE(SCHOOL-IX, CLASS-IX, STUDENT-IX) > 20
    DISPLAY STUDENT(SCHOOL-IX, CLASS-IX, STUDENT-IX).
SELECTION-END . EXIT.

```