

## Notions avancées : Tableaux

### 1 Principes

Une table (ou tableau) est une collection ordonnée d'objets du même type ; tout objet de la table peut être sélectionné indépendamment des autres en citant ses "coordonnées".

Exemple : Table des intitulés des jours de la semaine LUNDI MARDI MERCREDI JEUDI VENDREDI SAMEDI DIMANCHE. L'intitulé du 3e jour est MERCREDI.

Une table est de **dimension**  $n$ ,  $n > 0$  ; l'exemple précédent montre une table de 7 chaînes de caractères et de dimension 1.

- ◆ Le langage COBOL comporte des clauses permettant de décrire des tables de **dimension**  $\leq 3$  (7) ; les objets d'une table peuvent être de type élémentaire ou structuré.

Exemple : Tarif des consommations ; c'est une table de dimension 1, composée d'objets du type "consommation" défini par :

- . nom-consommation (chaîne de 15 caractères)
- . prix-consommation (réel)

- ◆ Le langage COBOL inclut les manipulation de tables, c'est-à-dire qu'il permet de sélectionner des objets appartenant à des tables. Deux techniques peuvent être utilisées :
  - L'indilage : consiste à préciser dans l'ordre les valeurs des différents sélecteurs sous forme d'indices; les indices étant du type entier, leurs valeurs peuvent être calculées à l'aide des instructions arithmétiques classiques.
  - L'indexation consiste à préciser dans l'ordre les valeurs des différents sélecteurs sous forme de pointeurs appelés index ; la valeur d'un index correspond en fait à une forme d'adressage de la mémoire centrale ; elle ne peut donc être calculée par les instructions arithmétiques classiques et nécessite l'emploi d'instructions spécifiques. Par ailleurs, un index est rattaché à un sélecteur d'une table et ne peut être utilisé à d'autres fins.

On peut cependant définir une variable index en DATA DIVISION, par la clause :

```
[USAGE IS] INDEX
```

**Toute autre clause est interdite.**

Ex. : 77 IND1 USAGE INDEX.

Ceci peut être utile pour sauvegarder des valeurs index.

La **sélection** d'un objet d'une table s'exprime par : nom de l'objet (liste d'expressions d'indices ou d'index). Le nombre d'expressions d'indice ou d'index correspond à la dimension de la table.

Une **expression d'indices ou d'index** est de la forme :

$$\text{nd-1} \left( \begin{array}{l} \text{entier-1} \\ \text{nd-2} \quad [ \{ \pm \} \text{entier-2} ] \\ \text{index-1} \quad [ \{ \pm \} \text{entier-3} ] \end{array} \right) \dots$$

Exemples :

A (3)  
 B (I, 5)  
 B (I, J)      ou I, J est du type entier  
 B (IND + 1, 5)    IND est du type index

Remarques :

- 1) Les virgules séparant les expressions d'indices ou d'index sont facultatives et peuvent être remplacées par des espaces.
- 2) Un objet d'une table étant nécessairement à occurrences multiples, il est indispensable de préciser l'occurrence de l'objet sur laquelle porte le traitement ; cette sélection doit être répétée pour chaque instruction faisant référence à cet objet.

## 2 Définition d'une table - OCCURS

Format 1 :      OCCURS ent-2 TIMES

$$\left\{ \begin{array}{l} \underline{\text{ASCENDING}} \\ \underline{\text{DESCENDING}} \end{array} \right\} \text{KEY IS nd2 [nd3] ...}$$

[INDEXED BY index1 [index2] ... ] ...

Format 2 :      OCCURS ent-1 TO ent-2 TIMES DEPENDING ON nd1

$$\left\{ \begin{array}{l} \underline{\text{ASCENDING}} \\ \underline{\text{DESCENDING}} \end{array} \right\} \text{KEY IS nd2 [nd3] ...}$$

[INDEXED BY index1 [index2] ... ] ...

- ◆ ent-1 et ent-2 sont des entiers désignant respectivement le nombre minimum et le nombre maximum d'objets dans la table (ent-1 < ent-2)
- ◆ nd1 doit décrire un entier positif ; elle contient le nombre exact d'occurrences de l'objet dans la table courante (nd1 ≥ 1) et sera placée dans la partie fixe de l'article ou hors de celui-ci.
- ◆ nd2, nd3, ... doivent être des rubriques subordonnées à celle contenant OCCURS ; cependant, nd2 peut désigner cette rubrique (si elle est seule) ; aucune des rubriques citées ne peut contenir OCCURS

- ◆ INDEXED est obligatoire si l'objet (ou ses rubriques subordonnées) doit être référencé par indexation ; index1, index2, ... ne peuvent être définis ailleurs dans le programme.
- ◆ une rubrique contenant le format 2 ne peut être suivie dans l'enregistrement que par des rubriques subordonnées.
- ◆ OCCURS ne peut être spécifié aux niveaux 01, 77, 66 ou 88
- ◆ nd1 ne peut pas être subordonnée à la rubrique sujet de OCCURS
- ◆ La locution DEPENDING, autorisant la gestion de tables de longueur variable (i.e. ayant un nombre variable d'éléments) permet d'optimiser l'occupation des supports externes, et de limiter les recherches en table quand celle-ci n'est pas totalement remplie dans tous les articles.

Exemple :

```

01 ABONNE.
   05 NOM PIC X(20)
   05 NB-OUV PIC 99.
   05 OUVRAGE OCCURS 1 TO 20 DEPENDING ON NB-OUV.
       10 CODOUV PIC 9(5).
       10 DATEMP PIC 9(6).

```

La locution ASCENDING (ou DESCENDING) KEY permet de préciser les rubriques de l'entrée servant de critères de classement dans le cas d'une table ordonnée. Elle est indispensable pour une recherche dichotomique (SEARCH ALL). L'ordre de classement est déterminé selon les règles de comparaison classique ; les critères sont à citer dans l'ordre d'importance décroissante (i.e. majeur en tête)

Exemple :

**Table des élèves d'une classe**, chaque élève ayant 10 résultats sous forme (note, coefficient)

```

01 TABLE-ELEVES.
   02 ELEVE OCCURS 25.
       03 NOM-ELEVE PIC X(20).
       03 RESULTAT-ELEVE OCCURS 10.
           04 NOTE-ELEVE PIC 99V9.
           04 COEFF-ELEVE PIC 9.

```

Cette description nous donne :

- au niveau 2, une table de 25 "élèves"
- au niveau 3, une table de dimension 2, de 25 fois 10 résultats

Ainsi :

ELEVE (3) réfère l'ensemble des informations concernant le 3ème élève.

NOM-ELEVE (3) désigne le nom du 3ème élève.

RESULTAT-ELEVE (3, 6) désigne le 6ème résultat du 3ème élève

COEFF-ELEVE (3, 6) désigne le coefficient du 6ème résultat du 3ème élève.

### Utilisation conjointe de la clause OCCURS et des autres clauses :

- Aucune rubrique rattachée à un objet appartenant à une table ne peut être initialisée par une clause VALUE en ANS74. En ANS85, chaque occurrence est initialisée par la valeur spécifiée.
- La clause REDEFINES est autorisée :
  - pour redéfinir la structure d'un élément d'une table.
  - lorsque la table est incluse dans la redéfinition.

Il est par contre **interdit de redéfinir** une rubrique dont la structure comporte une table.

Exemples :

```

1) 01 A.
    02 TABLE1.
        03 ELEMENT PIC X OCCURS 10.
    02 B REDEFINES TABL PIC X(10).
        est interdit

2) 01 A.
    02 B PIC X(10).
    02 TABLE2 REDEFINES B.
        03 ELEMENT PIC X OCCURS 10.
        est autorisé
    
```

```

3) 01 A.
    02 ELEMENT OCCURS 10.
        03 A1 PIC XX.
        03 A2 REDEFINES A1.
            04 E1 PIC 9.
            04 E2 PIC 9.
        est autorisé
    
```

## 3. SET

Elle permet d'initialiser, d'incrémenter ou de décrémenter une variable index.

Format 1 : SET { nd-1 [nd-2]... } TO index-3 { nd-3 }  
 { index-1 [index-2].. } { ent-1 }

Format 2 : SET index-4 [index-5]... { UP BY } { nd-4 }  
 { DOWN BY } { ent-2 }

- ◆ L'instruction SET...TO permet d'initialiser une suite d'index à une valeur exprimée sous forme d'entier (nd-3 ou ent-1) ou d'index (index-3) ; inversement, elle permet d'initialiser une suite d'indices (nd-1, nd-2,...) à une valeur exprimée sous forme d'index (index-3).

Attention : Il est interdit d'initialiser un indice, par l'instruction SET, à une valeur exprimée sous forme d'un entier ; une telle affectation s'exprime par une instruction MOVE (cf. chap. 3 §4.8)

Exemple : Si I et J sont du type entier et IND du type index, on peut écrire SET I J TO IND :  
initialisation des indices I et J au rang correspondant à la valeur de l'index IND SET IND TO 3 :  
Positionnement de l'index IND sur la 3e occurrence dans la table à laquelle il est rattaché.

- ◆ L'instruction SET...UP BY sert à incrémenter un ou plusieurs index d'un déplacement correspondant à n occurrences, n étant exprimé par nd-4 ou ent-2.

Exemple: SET IND UP BY 2

L'instruction SET...DOWN BY sert à décrémenter des index

Exemple : SET IND DOWN BY 3

Remarques :

- 1) Le format 2 de l'instruction SET ne peut être utilisé pour incrémenter ou décrémenter des indices ; une telle opération s'exprimerait par les instruction ADD ou SUBTRACT.
- 2) Dans les tests de relation, il est possible de faire référence à des index ; leurs valeurs sont "remplacées" par des valeurs entières correspondant aux rangs des occurrences sur lesquelles ils pointent ; on est ainsi ramené à des comparaisons numériques.

## 4. Recherche en table - SEARCH

L'instruction SEARCH est utilisée pour rechercher dans une table un élément qui satisfait à une condition donnée.

### 4.1. Recherche séquentielle

$$\text{SEARCH } \text{élément} \left( \text{VARYING} \left\{ \begin{array}{l} \text{rubrique} \\ \text{index} \end{array} \right\} \right) \quad [\text{AT } \text{END} \text{ phrase impérative}]$$

$$\left\{ \begin{array}{l} \text{WHEN } \text{condition-1} \\ \text{NEXT } \text{SENTENCE} \end{array} \right\} \left\{ \begin{array}{l} \text{phrase-impérative} \\ \text{NEXT } \text{SENTENCE} \end{array} \right\} \dots$$

[END-SEARCH]

- ◆ L'élément de table cité après SEARCH ne doit être ni indicé ni indexé.

- ◆ Si une rubrique sert à la progression, elle doit être numérique entière (indice) ou d'usage INDEX.

La recherche débutera à partir de l'élément pointé par le contenu de l'index (ou de la rubrique) si ce dernier est inférieur au rang maximum autorisé. Les conditions sont évaluées dans l'ordre cité, et la recherche se terminera dès que l'une d'entre elles sera satisfaite ; l'index point alors l'entrée pour lequel la condition est remplie. Si la recherche a été vaine (fin de table atteinte sans satisfaire aucune condition), la locution AT END sera validée et la phrase impérative correspondante exécutée ; à défaut le contrôle passe à l'instruction suivant SEARCH.

- ◆ A la fin de l'exécution d'une phrase impérative ne comportant pas GO TO, le contrôle passe à l'instruction suivante.
- ◆ Si VARYING n'est pas spécifié, l'index utilisé par défaut est le premier cité dans la locution INDEXED BY de la définition de l'élément (idem s'il s'agit d'une rubrique après VARYING) ; les autres ne sont pas touchés.

N.B. : L'élément cité dans SEARCH peut être subordonné à une rubrique contenant OCCURS (cas de table à 2 ou 3 dimensions).

Exemple : SET IND1 TO 1  
 SEARCH ELEMENT WHEN RUB (IND1) = VALEUR PERFORM TRAIT  
 WHEN RUB (IND1) = SPACE GO TO ERREUR

## 4.2. Recherche dichotomique

SEARCH ALL élément [AT END phrase-impérative]

WHEN condition { phrase-impérative }  
 { NEXT SENTENCE }

[END-SEARCH]

où la condition est de la forme :

$$\text{critère-1} \left\{ \begin{array}{l} = \\ \underline{\text{EQUAL}} \text{ TO} \end{array} \right\} \left\{ \begin{array}{l} \text{rubrique-1} \\ \text{exp-arith-1} \end{array} \right\} \left[ \underline{\text{AND}} \right. \text{ critère-2} \left. \left\{ \begin{array}{l} = \\ \underline{\text{EQUAL}} \text{ TO} \end{array} \right\} \left\{ \begin{array}{l} \text{rubrique-2} \\ \text{exp-arith-2} \end{array} \right\} \right] \dots$$

- ◆ Les critères doivent être cités dans la locution KEY et indexés avec le premier index ; ce peut être également un nom-condition associé à une valeur unique du critère.
- ◆ Les rubriques spécifiées ne doivent pas être des critères de l'élément concerné, ni être indexés par le premier index de celui-ci ; il en va de même pour les opérandes des expressions arithmétiques. On peut utiliser des littéraux.
- ◆ Lorsqu'un critère est utilisé, tous ceux le **précédant** dans la locution KEY doivent être également référencés.
- ◆ Le contenu de l'index sera **indéterminé** en cas de recherche infructueuse (critères non trouvés ou non discriminatoires).

Exemple :

```
.  
. 05 RESID OCCURS 1000 ASCENDING DEPT COMU INDEXED BY IND IND1.  
   10 DEPT PIC 99.  
   10 COMU PIC 999.  
   10 RUE PIC X(30).  
.  .  
.  .  
.  .  
      SEARCH ALL RESID WHEN DEPT (IND) = 67 AND COMU (IND) = 200  
          PERFORM STRBG-OUEST.
```