COBOL

avec exemples de programme

Auteur : Maurice Grossel

Table des matières

aaaa

Structure

COBOL est un langage de programmation créé en 1959 (officiellement le 18 septembre 1959).

Son nom est l'acronyme de **CO**mmon **B**usiness **O**riented **L**anguage qui révèle sa vocation originelle : être un langage commun pour la programmation d'applications de gestion.

Le cobol est un langage structuré physiquement et logiquement :

Structure physique

Au début de l'informatique, les programmes étaient transcrits sur des cartes perforées de 80 colonnes, une ligne par carte. Pour des besoins évidents, certaines colonnes étaient réservées au tri :

- les colonnes 1 à 6 étaient réservées à la numérotation des cartes à l'intérieur d'un programme,
- La colonne 7 est :
  - à blanc (non remplie) : il s'agit d'une ligne cobol
  - * (astérisque) il s'agit d'un commentaire
  - (tiret) c'est la suite d'un libellé
- les colonnes 8 à 11 : marge A (début d'un titre)
- les colonnes 12 à 72 : marge B (suite éventuel du titre, instructions)
- les colonnes 73 à 80 étaient destinées au nom du programme.

Structure logique

Un programme cobol	(program)
– possède 4 divisions	(division)
– chaque division peut être découpée en sections	(section)
– les sections en paragraphes,	(praragraph)
– les paragrapahes en phrases	(sentence)
– les phrases en instructions	(statement)
– les instructions en mots (verbe, nom, nombre)	(verb, word, number)
– les mots ont des caractères	(character)

- A à Z,
- de 0 à 9,
- parenthèses () tiret - virgule , point . signe inférieur < supérieur > égal = quote (apostrophe) ' double quote (guillemet à l'anglaise) " dollar $ euro € livre £

Les noms

Un nom sert à nommer les variables et les paragraphes

C'est une suite de 1 à 30 caractères :

- lettre majuscule,
- chiffre,
- et – (tiret)

Il ne peut pas commencer ou se terminer par un tiret ni contenir 2 tirets consécutifs.

Il ne peut pas comporter de blanc.

Il est précédé d'au moins un blanc ou d'une parenthèse ouverte. et suivi d'au moins un blanc, ou d'une parenthèse droite, ou d'un signe de ponctuation.

Les noms de variables ne doivent pas être ambigus. Si deux zones portent le même nom, il faut spécifier sa zone de groupe par IN ou OF. IN et OF sont strictement identique. Exemple :

NOM OF CLIENT

NOM OF FOURNISSEUR

Les mots réservés

Certains mots sont réservés. Pour plus de lisibilité, l'initiale est en majuscule et le reste en minuscules.

A : Accept, Access, Add, Advancing, After, All, Alphabet, Alphabetic, Alphabetic-Lower, Alphabetic-upper, Alphanumeric, Alpha-edited, Also, Alter, Alternate, And, Any, Are, Areas, Ascending, Assign, At, Author

B : Before, Binary, Blank, Block, Bottom, By

C : Call, Cancel, Cd, Cf, Ch, Character, Characters, Class, Clock-units, Close, Cobol, Code, Code-set, Collating, Column, Comma, Common, Communication, Comp, Computational, Compute, Configuration, Contains, Content, Continue, Control, Controls, Converting, Copy, Corr, Corresponding, Count, Currency, Current-date

D: Data, Date, Date-Compiled, Date-Written, Day, Day-Of-Week, De, Debug-Contents,

Debug-Item, Denug-Line, Debug-Name, Debug-Sub-1, Debug-Sub-2, Debug-Sub-3, Debugging, Decimal-Point, Declaratives, Delete, Delimited, Delimiter, Depending, Descending, Destination, Detail, Disable, Display, Divide, Division, Down, Duplicates, Dynamic

E: Egi, Else, Emi, Enable, End, End-Add, End-Call, End-Compute, End-Delete, End-Divide,

End-Evaluate, End-If, End-Multiply, End-Of-Page, End-Perform, End-Read, End-Receive, End-Return, End-Rewrite, End-Search, End-Start, End-String, End-Substract, End-Unstring, End-Write, Enter, Environment, Eop , Equal, Error, Esi, Evaluate, Every, Exception, Exit, Extend, External

F: False, Fd, File, File-Control, Filler, Final, First, Footing, For, From, Function

G: Generate, Giving, Global , Go, Greater, Group

H : Heading, High-Value, High-Values

I: I-O, I-O-Control, Identification, If, In, Index, Indexed, Indicate, Initial, Initialize, Initiate, Input, Input-Output, Inspect, Installation, Into, Invalid, Is, Just, Justified

K: Key

L: Label, Last, Leading, Left, Length, Less, Limit, Limits, Linage, Linage-Counter, Line, Line-Counter, Lines, Linkage, Lock, Low-Value, Low-Values

M: Memory, Merge, Message, Mode, Modules, Move, Multiple, Multiply

N: Native, Negative, Next, No, Not, Number, Numeric, Numeric-Edited

O: Object-Computer, Occurs, Of, Off, Omitted, On, Open, Optional, Or, Order, Organization, Other, Output, Overflow

P: Packed-Decimal, Padding, Page, Page-Counter, Perform, Pf, Ph, Pic, Picture, Plus, Pointer, Position, Positive, Printing, Procedure, Procedures, Proceed, Program, Program-Id, Purge

Q: Queue, Quote, Quotes

R: Random, Rd, Read, Receive, Record, Records, Redefines, Reel, Reference, References, Relative, Release, Remainder, Removal, Renames, Replace, Replacing, Report, Reporting, Reports, Rerun, Reserve, Reset, Return, Return-code, Reversed, Rewind, Rewrite, Rf, Rh, Right, Rounded, Run

S: Same, Screen, Sd, Search, Section, Security, Segment, Segment-Limit, Select, Send, Sentence, Separate, Sequence, Sequential, Set, Sign, Size, Sort, Sort-Merge, Source, Source-Computer, Space, Spaces, Special-Names, Standard, Standard-1, Standard-2, Start, Status, Stop, String, Sub-Queue-1, Sub-Queue-2, Sub-Queue-3, Subtract, Sum, Suppress, Symbolic, Sync, Synchronized

T: Table, Tally, Tallying, Tape, Terminal, Terminate, Test, Text, Than, Then, Through, Thru, Time, Time-of-day, Times, To, Top, Trailing, True, Type

U: Unit, Unstring, Until, Use

W: Words

CURRENT-DATE : désigne sur 8 octets la date du jour sous la forme MM/JJ/AA. Elle ne peut pas être modifiée par le programmeur.

DATE : désigne la date du jour sous la forme PIC 9(6) YYMMJJ (année, mois, jour).

DAY : désigne la date du jour sous la forme PIC 9(5) YYQQQ (année, jour de l'année). Par exemple 22048 désigne le 17 février 2022.

RETURN-CODE : désigne une zone binaire de 2 octets PIC

TALLY : c'est un compteur non signé de 5 chiffres que le développeur n'a pas besoin de décrire.

TIME : désigne l'heure sous la forme PIC 9(8) HHMMSSC (hure, minute, seconde, centième de seconde). Par exemple 010203001 heure 2 minutes 3 secondes 0 centième de seconde.

TIME-OF-DAY : désigne sous forme de PIC X(6) l'heure sous la forme HHMMSS. Elle ne peut pas être modifiée par le programmeur.

Les littéraux

Un littéral est une constante incluse entre quotes ou doubles quotes, qui contient au maximum 120 caractères.

S'il contient une apostrophe, il la doubler.

S'il est trop long pour tenir sur une ligne, on continue sur la ligne suivante avec un tiret en colonne 7 et en commençant par une apostrophe. Dans ce cas la première ligne ne se termine pas par une apostrophe.

Exemple : 05 PHRASE PIC X(6) VALUE 'il l''a".

Remarque : les 2 quotes successives à l'intérieur du littéral ne compte que pour 1 caratère.

Les littéraux en virgule flottante

Les nombres en virgule flottante s'écrivent : (+ ou -)mantisse(+ ou -)exposant.

La mantisse comporte 1 à 16 chiffres et est suivi par E, puis par l'exposant sur 1 ou 2 chiffres.

Exemple : -0,123456 avec 10 à la puissance 50 s'écrit -0,123456E+50

Les constantes figuratives

Les constantes figuratives sont des littéraux désignés par un nom réservé :

ZERO, ZEROS, ZEROES : désigne la valeur zéro. La variable est remplie de caractère zéros. Toutes ces écritures sont identiques, le compilateur ne contrôle pas le singulier et le pluriel.

SPACE, SPACES : désigne un ou des blancs, c'est à dire en hexadécimal EBCDIC 40, ou en ASCII 32

HIGH-VALUE : désigne que tous les caractères de la variable ont la valeur la plus haute en binaire (1), donc en hexadécimal, en EBCDIC comme en ASCII, c'est FF

LOW-VALUE : désigne que tous les caractères de la variable ont la valeur la plus basse en binaire (0), donc en hexadécimal, en EBCDIC comme en ASCII, c'est 00

ALL représente un répétition d'un littéral. Exemple : ALL 'AZ12' indique que la variable est une suite des 4 caractères AZ12, avec une éventuelle troncature à droite. Si la variable a un format X(6), sa valeur est AZ12AZ.

Description des données

Les données sont décrites dans la DATA DIVISION.

La définition d'une donnée comprend le niveau (voir plus bas), son nom, son format, son usage, sa valeur.

donnée alphabétique

Son image est PIC A.

Exemples : NOM PIC AAAAAAAAAA ou NOM PIC A(10)

La variable NOM comporte 10 caractères exclusivement alphabétiques.

Ce format est très peu usité.

donnée alphanumérique

Son image est PIC X

La donnée alphanumérique peut contenir n'importe quel caractère codé en EBCDIC, ou ASCCII, même s'il n'est pas affichable.

Elle est toujours cadrée à droite.

Quand on copie une donnée dans une donnée d'une taille plus petite, celle-ci ne reçoit que le début.

Quand on copie une donnée dans une donnée d'une taille plus grande, celle-ci la reçoit et la complète à droite par des blancs.

donnée numérique

Son image est réprésentée par les symboles 9 S V P

Par défaut, le point décimal sépare les entiers des décimales.

On peut le remplace par une virgule en ajoutant dans le programe :

```
SPECIAL-NAMES.
        DECIMAL-POINT IS COMMA.
```

Le 9(n) : n indique le nombre de chiffres de la variable, mais peut être différent du nombre d'octets. Voir plus bas COMP-3

Le S : il indique que la variable peut être négative. Si le S est absent et que le résultat du calcul est négatif, la variable contient la valeur absolue.

Le V: il indique la place de la virgule virtuelle. Le compilateur en tient compte ppiur ses calculs, mais elle n'occupe pas d'octet ni de demi octet. Quand le nombre est un entier, le V est facultatif.

Lors d'une copie, le cadrage se fait en se positionnant sur l'emplacement de la virgule virtuelle.

La copie d'un nombre dans un autre avec plus d'entiers génère des zéros à gauche.

La copie d'un nombre dans un autre avec plus de décimales génère des zéros à droite.

Exemple : copie de ZA dans ZB et ZC

```
ZA PIC  99V9  VALUE 12,3
ZB PIC  99V99 après copie : 12,3
ZC PIC  999V9 après copie : 012,3
```

Le P : il indique un zéro virtuel pour lequel aucune position de la mémoire n'est réservée. Il est très peu utilisé.

Exemple : TOTAL PIC 999PPPV La variable totale s'exprime en milliers, mais n'occupe que 3 octets.

USAGE

Mot réservé facultatif.

- display, ou étendu. Il correspond à un caractère par octet.

USAGE IS DISPLAY : le IS est facultatif. L'expression est prise par défaut.

- COMP ou COMPUTATIONAL. Il correspond à une donnée numérique codée en binaire.

Une donée binaire doit être signée. Exemple : S9(3) COMP.

Pour des raisons de frontières imposées par l'épaisseur de la mémoire, les longueurs des zones binaires sont normalisées sur ou 2, ou 4 ou 8 octets.

Exemple : 01 MONTANT PIC S999 COMP-3 VALUE -2.

- PICTURE, ou en abrégé PIC, introduit le format (alphanumérique ou numérique, et la taille).

2 octets: valeurs maximum : 32.767. Correspond à S9(n) où n vaut 4 au maximum, exemple S9(4),

4 octets: valeurs maximum : 32.147.483.647. Correspond à S9(n) où n vaut entre 5 et 9, exemple S9(5)V9(4),

8 octets: valeurs maximum : ???? Correspond à S9(n) où n vaut entre 10 et 18, exemple S9(16)V9(2).

- COMP-1 ou COMPUTENTIONAL-1

Pour les données numériques en virgue flottante sur 1 mots, soit 4 octets ;

- COMP-2 ou COMPUTENTIONAL-2

Pour les données numériques en virgue flottante sur 2 mots, soit 8 octets ;

- COMP-3 ou COMPUTENTIONAL-

Appelé encore "packé".

Cet usage permet de mettre deux chiffres par octets.

Par exemple le nombre PIC 9(3) COMP-3 qui vaut 123 est écrit /12/F3 / sur 2 octets

Le nombre PIC S9(3) COMP-3 qui vaut+123 est écrit /12/3C/ sur 2 octets

Le nombre PIC S9(3) COMP-3 qui vaut-123 est écrit /12/3D / sur 2 octets

C comme Crédit, D comme Débit.

Le nombre PIC 9(3) DISPLAY qui vaut+123 est écrit /F1/F2/C3 / sur 3 octets.

VALUE

VALUE sert à initialiser une valeur dans la DATA DIVISION, uniquement dans la WORKING-STORAGE SECTION, et uniquement pour les données élémentaires, mais pas pour des données redéfinies.

On ne peut pas mettre une VALUE sur une zone qui redéfinit, ni sur aucune de ses zones.

Exemples :

```
01 CONSTANTE-DE-NEPER PIC 9V9(5) VALUE 2,71828.
01 REMARQUE-NEPER PIC X(100) VALUE 'ELLE EST TRES RARE EN GESTION'.
01 NULLE PIC 999 VALUE ZERO.
01 TEXTE PIC XXXXX  VALUE SPACE.
```

BLANK WHEN ZERO

Si on ne veut pas éditer les zéros d'une zone qui est à zéro on utilise la clause BLANK WHEN ZERO.

Exemples :

```
01 CONSTANTE-DE-NEPER PIC 9V9(5) BLANK WHEN ZERO.
```

L'avantage de la clause BLANK WHEN ZERO est de supprimer la virgule quand le nombre est nul et qu'il est défini avec des décimales.

SIGN

On peut coder le signe sur un octet précédent ou suivant le nombre.

Exemples:

NOMBRE PIC S999 **SIGN LEADING** SEPARATE. Exemple : +123 sur 4 octets

NOMBRE PIC S999 **SIGN TRAILING** SEPARATE. Exemple : 123+ sur 4 octets

NOMBRE PIC S999 **SIGN LEADING** . Exemple : +123 sur 3 octets. Le signe est sur le premier demi octet.

NOMBRE PIC S999 **SIGN TRAILING** . Exemple : 123+ sur 3 octets. Le signe est sur le dernier demi octet, comme d'habitude.

Les niveaux

Les données peut regrouper plusieurs données, qui peuvent elles-mêmes regroupées d'autres données., et ainsi de suite.

La donnée qui n'en regroupe aucune autre est appelée "donnée élémentaire" et doit avoir une PICTURE. Les autres, "données de groupe" et ne doivent pas avoir de PICTURE.

Lors de la définition, chaque donnée est précédée d'un niveau, suivie éventuellement d'une PICTURE et d'une VALUE, et obligatoirement d'un point.

La donnée qui ne fait pas partie d'une autre est soit d'un niveau 01, soit d'un niveau 77. Pour être d'un niveau 77, elle ne doit contenir aucune donnée.

La numérotation va de 01 à 49 au maximum, et peut sauter des numéros. En général, on va de 01 à 05, puis de 5 en 5 pour permettre l'insertion d'un nouveau niveau.

Exemple :

```
77     IDENTIFIE    PIC 9(8).
01     IDENTIFIANT  PIC 9(8).
01     IDENTITE1.
         05 NOM            PIC X(30).
         05 PRENOM  PIC X(30).
```

Une zone de groupe peut contenir une zone de groupe.

```
01     IDENTITE2.
         05 NOM            PIC X(30).
         05 PRENOMS .
                    10 PRENOM1    PIC X(30).
                    10 PRENOM2    PIC X(30).
```

Cobol calcule automatiquement la longueur des zones de groupe.

Ici IDENTITE1 fait 30 + 30 = 60 caractères.

IDENTITE2 fait 30 + 30 + 30 = 90 caractères.

PRENOMS fait 30 + 30 = 60 caractères.

Règles :

- chaque définition se termine par un point

- le "01" doit être écrit en marge A

- les autres nombres ("02" à "49") doivent être écrit en marge B

- toute donnée élémentaire doit avoir une PICTURE.

- l'USAGE (PICTURE nnn) doit être déclaré pour chaque donnée élémentaire, ou pour leur donnée de groupe.

- une donnée de groupe est considérée comme ayant un format en X(n)

Avant les compilateurs alignaient les données de niveau 01 sur des frontières de double mot.

SYNC???

...

REDEFINES

Ce mot permet de découper autrement des données, quel que soit leur niveau. Il faut que la zone qui redéfinit et celle qui est redéfinie soient du même niveau et qu'il n'y en pas d'autres entre elles du même niveau, sauf si elles redéfinissent toutes la même zone.

On ne peut pas mettre une VALUE sur une zone qui redéfinit, ni sur aucune de ses zones.

```
01     ADRESSE.
         05 RUE PIC X(40).
         05 CODE-POSTAL PIC X(5).
         05 CODE-POSTAL-NUM     REDEFINES  CODE-POSTAL  PIC 9(5).
         05 VILLE PIC X(35).
01     WS-GROUPE.
         05 RUE PIC X(40).
         05 WS-A.
            10 WS-B PIC X(40).
         05 WS-B     REDEFINES   WS-A  PIC X(40).
         05 WS-C     REDEFINES   WS-A  PIC X(40).
            10 WS-D PIC X(20).
            10 WS-E PIC X(20).
         05 SUITE PIC X(99).
```

JUSTIFIED RIGHT ou JUST

Cette clause permet de cadrer à gauche des remplissages de données, en les complétant éventuellement de blancs à gauche, ou en les tronquant.

Elle ne s'applique qu'aux zones élémentaires alphabétiques ou alphanumériques.

Les tables

Les tables en Cobol sont des ensembles répétitifs de données de format et de PICTURE identiques, qu'on appelle d'habitude "tableau" dans le autres langages.

La clause OCCURS est suivi d'un nombre qui indique le nombre de répétitions. Elle ne peut pas apparaître aux niveaux 01, 66, 77, 88 dans la DATA DIVISION.

Une donnée définie avec un OCCURS est dite de "premier niveau". Si elle contient une donnée qui a aussi un OCCURS, on parle de deuxième niveau. Il peut y avoir 3 niveaux.

Attention à ne pasconfondre le niveau de description dans la DATA DIVISION(01à 49) avec le niveau des tables qui correspond au nombre d'indices

Les éléments sont gérés par un indice ou un index.

Indice : c'est un nombre ou une donnée numérique. C'est le numéro de poste de la donnée. Il va de 1 au nombre maximum de postes de la table.

Exemple :
La table des mois.

```
01 WS-TAB-MOIS.
       05 WS-POSTE OCCURS 12.
                     10 WS-MOIS   PIC X(10).
```

Pour désigner le mois, on utilise un indice WS-MOIS (5) ou WS-MOIS (WS-I)si WS-I est numérique (mais pas numérique d'édition).
Le poste peut avoir plusieurs variables

```
01 WS-TAB-MOIS.
       05 WS-POSTE OCCURS 12.
                     10 WS-MOIS   PIC X(10).
                     10 WS-CA  PIC S9(5)V99  COMP-3.
```

Une zone peut également être découpée en table. On a des postes à deux niveaux

```
01 WS-TAB-MOIS.
       05 WS-POSTE OCCURS 12.
                     10 WS-MOIS   PIC X(10).
                     10 WS-VENTES.
                           15 WS-CA                      PIC S9(5)V99  COMP-3.
                           15 WS-NB-CLIENTS  PIC S9(5)V99  COMP-3.
```

On utilisera WS-MOIS (WS-I) et WS-NB-CLIENTS (WS-I WS-J).

Une table gère le stock de 500 produits. Pour chaque produit, on suit le nombre des ventes et du chiffres d'affaires chaque mois.
Le produit 150 est appelé T-PRODUIT(150).
Le nombre de ventes du cinquième mois de l'année pour ce produit est T-NB-VENTE(150 5).

```
01 T-TABLE-GEREE-PAR-INDICE.
       05 T-STOCK OCCURS 500.
             05 T-PRODUIT  PIC X(6).
       05 T-MOIS           OCCURS 12.
                    10 T-NB-VENTE              PIC S9(9) COMP-3.
                    10 T-CA-VENTE       PIC S9(9)V99 COMP-3.
```

Si on a défini une donnée, par exemple W-INDICE, on peut mettre 150 dans cette donnée et T-PRODUIT(150) devient T-PRODUIT(W-INDICE).

Index : c'est un nombre binaire dont in ne connait pas la valeur et qui est calculé différemment des autres données numériques. Il doit être attaché à une table lors de sa création (12 noms d'index maximum par niveau)

```
01 TABLE-GEREE-PAR-INDEX.
       05 ELEMENT1 OCCURS 2             INDEXED BY    IDX1 IDX2.
             10 ELEMENT2 OCCURS 3       INDEXED BY    IDX3 IDX4 IDX5.
                              15 RUBRIQUE1          PIC X(4).
                              15 RUBRIQUE2          PIC X(2).
```

Dans la pratique, on utilise un seul index pour chaque OCCURS.

On utilisera le verbe SET pour :
- POSITIONNER un index ou plusieurs index
exemple : SET IDX1 TO 1 , signifie positionner l'index IDX1 sur le 1er poste de la table.
Pour copier un index dans un autre index, on utilise SET
exemple : SET IDX2 TO IDX1, cette instruction positionne IDX2 sur l'occurrence pointée par IDX1
- INCREMENTER ou DECREMENTER un index exemple :
SET IDX UP BY 1, incrémente d'une occurrence

SET IDX DOWN BY 3, décrémente de 3 occurrences
- pour SAUVEGARDER la valeur d'un index ; dans ce cas il faut préalablement définir une zone en WORKING-STORAGE SECTION
Par exemple : 01 IDXSAUV USAGE IS INDEX.
Dans ce cas il n'est pas précisé de clause PICTURE : c'est le compilateur Cobol qui se charge de réserver la zone (généralement 4 octets binaires).
Dans la PROCEDURE DIVISION, on pourra si l'on a besoin de sauvegarder le contenu d'un index IDX par exemple, écrire l'instruction : SET IDXSAUV TO IDX.
Il est toutefois préférable d'attacher à une table plusieurs index, et de se servir de l'un d'eux comme index de sauvegarde.
Avec une table indexée, on peut utilisée le verbe SEARCH.

PICTURE d'édition.

Les nombres ont besoin d'un codage pour plus de lisibilité : transformer la virgle virtuelle en virgule réelle (ouen point pour les Américains), grouper les nombres par 3, éliminer les zéros non significatifs de gauche, ajouter le signes + ou -, ou CR ou DB pour crédit ou débit, accoler un signe de devise.

Pour ce faire, l'édition va utiliser en plus des A, X, 9, V et P les signes Z, *, B, O, +, -, CR, DB, $, £ et la virgule ou le point suivant qu'on est américain ou français.
Z : indique que le zéro non significatif sera remplacé par un blanc.
Exemple 000102 avec PIC Z(6) est imprimé bbb102, avec b qui représente un blanc (= un espace).

* : indique que le zéro non significatif sera remplacé par un astérisque.
Exemple 000102 avec PIC Z(6) est imprimé ***102. Cette mesure était utilisée pour éviter les falsification et l'ajout d'un chiffre.

B : indique qu'un blanc est inséré.
Exemple 113102 avec PIC ZZZBZZ9 est imprimé 113 102.

O : ???

+ : indique que le nombre positif est affecté d'un + et le nombre négatif d'un -.
Exemple 123 positif PIC +999 est imprimé +123. PIC 999+ est imprimé 123+
et 123 négatif PIC -999 est imprimé -123. PIC 999- est imprimé 123-

- : indique que le nombre négatif est affecté d'un blanc et le nombre négatif d'un -.
Exemple 123 positif PIC -999 est imNom-donnéeprimé blanc123. PIC 999- est imprimé 123blanc
et 123 négatif PIC -999 est imprimé -123. PIC 999- est imprimé 123-

CR et DB suivent les mêmes règles PIC+ et PIC - qu'ils remplacent.

$ et £ suivent les mêmes règles que * qu'ils remplacent.

Les niveaux spéciaux

66 : Nom-donnéee-1 RENAMES Nom-donnéee-2
ou Nom-donnéee-1 RENAMES Nom-donnéee-2 THRU Nom-donnéee-3

Le niveau 66 est écrit en marge A, comme le niveau 01. Il permet de donner un nouveau nom à une zone de groupe ou à une zone élémentaire.

L'écriture avec THRU permet de regrouper plusieurs nom de données sous une nouvelle appellation.

77 : Ce niveau n'est utilisable qu'en WORKING-STORAGE SECTION pour une zone éle=émentaire indépendante de toute structure.
Le niveau 77 est écrit en marge A, comme le niveau 01.

88 : nom-condition VALUE littéral.
Ou nom-condition VALUE littéral-1 THRU littéral-2.

Le niveau 88 sert à définir des noms correspondants à des valeurs différentes de la donnée.
Le niveau 88 est écrit en marge B.
Exemple :

```
10 ETAT-CIVIL PIC X.
        88 MONSIEUR  VALUE 1.
        88 MADAME           VALUE 2.
```

Exemple : 10 AGE PIC X.

```
 88 MINEURS          VALUE 1 THRU 17.
 88 JEUNES           VALUE 18 THRU 25.
 88 ADULTES          VALUE 26 THRU 70.
 88 VIEUX            VALUE 70 THRU 150.
```

En PROCEDURE DIVISION, on peut au choix écrire :

```
 IF  ETAT-CIVIL = '1'  ... ou IF    MONSIEUR ...
```