Chapitre 3
Déclarer et manipuler les données

1. Introduction

Du fait de l'environnement matériel sur lequel se trouve une grande partie des programmes COBOL aujourd'hui, l'environnement grand système IBM, un petit préambule sur les codes de caractères est indispensable.

1.1 Codes de caractère : EBCDIC

Après avoir remporté le concours organisé par le Bureau du recensement des États-Unis pour le recensement de 1890 avec une machine de statistiques à cartes perforées s'inspirant du système inventé par Jean-Baptiste Falcon en 1728 sur les métiers à tisser que Basile Bouchon avait rendu programmables à l'aide de rubans perforés en 1725 (invention reprise en 1801 par Joseph Marie Jacquard), Herman HOLLERITH fonde, en 1896, la Tabulating Machine Co. En 1924, après divers regroupements, la Tabulating Machine Co. devient IBM (*International Business Machines*) qui définit, en 1928, un standard de cartes perforées à 80 colonnes. Le code alors utilisé, baptisé code Hollerith, reste limité en nombre de caractères. En 1955, IBM met au point un code de caractères sur 6 bits qui offre 64 caractères (2^6) : le BCD (*Binary Coded Decimal*). Ce code sera utilisé sur la série des 700/7000 sur lesquels sera disponible le compilateur FORTRAN.

C'est en 1960 que débutent les travaux de l'American Standards Association (ASA) auxquels participe notamment Robert BEMER, pour la définition d'un code de caractères standard. Bien qu'IBM soit déjà en train de réfléchir à un codage sur 8 bits, définir un code sur 8 bits semble à l'ASA trop en avance par rapport aux technologies existantes et ils décident donc de définir un code sur 7 bits seulement n'offrant qu'un jeu de 128 (2^7) caractères.

Le 7 avril 1964, IBM annonce la mise sur le marché de sa série des 360. Ils utilisent une extension sur 8 bits du BCD, qui offre un jeu de 256 (2^8) caractères : l'EBCDIC (*Extended Binary-Coded Decimal Interchange Code*). IBM décide de faire de l'EBCDIC son code de caractères principal tout en annonçant que la série des 360 serait compatible avec l'ASCII en cours de définition.

Ce n'est qu'en 1968 qu'est officiellement publié le code ASCII (*American Standard Code for Information Interchange*) sous la référence ANSI X3.4-1968. Il est trop tard pour IBM pour faire retour arrière : ses machines resteront en EBCDIC. Leader sur le marché du mainframe durant les décennies suivantes, IBM continuera et continue d'utiliser son code EBCDIC. Cela a des conséquences directes sur la programmation puisque les caractères ASCII et EBCDIC ne sont pas séquencés de la même façon bien que, depuis, le standard ASCII ait été revu et possède maintenant de nouvelles versions codées sur 8 bits, à la base de beaucoup de nouvelles normes.

Voici un extrait de la table de conversion qui se retrouve dans sa totalité à la fin de l'ouvrage ainsi qu'en téléchargement :

Dec	Hex	ASCII	EBCDIC (297)	Dec	Hex	ASCII	EBCDIC (297)	Dec	Hex	ASCII	EBCDIC (297)	Dec	Hex	ASCII	EBCDIC (297)
32	20	blanc	blanc	52	34	4		129	81		a	198	C6		F
33	21	!		53	35	5		130	82		b	199	C7		G
34	22	"		54	36	6		131	83		c	200	C8		H
35	23	#		55	37	7		132	84		d	201	C9		I
36	24	$		56	38	8		133	85		e	202	CA		
37	25	%		57	39	9		134	86		f	203	CB		ô
38	26	&		64	40	@	blanc	135	87		g	204	CC		ö
39	27	'		65	41	A		136	88		h	226	E2		S
40	28	(		66	42	B	â	137	89		i	227	E3		T
41	29	)		67	43	C	ä	138	8A		«	228	E4		U
42	2A	*		68	44	D	@	139	8B		»	229	E5		V
43	2B	+		69	45	E	á	145	91		j	230	E6		W
44	2C	,		70	46	F	ã	146	92		k	231	E7		X
45	2D	-		71	47	G	å	147	93		l	232	E8		Y
46	2E	.		97	61	a	/	148	94		m	233	E9		Z
47	2F	/		98	62	b	Â	193	C1		A	240	F0		0
48	30	0		99	63	c	Ä	194	C2		B	241	F1		1
49	31	1		100	64	d	À	195	C3		C	242	F2		2
50	32	2		101	65	e	Á	196	C4		D	243	F3		3
51	33	3		102	66	f	Ã	197	C5		E	244	F4		4

Lorsque les données sont triées, elles ne sont pas dans le même ordre selon qu'elles sont codées en EBCDIC (environnement IBM) ou en ASCII (autre environnement). Le chiffre 0 (48 en décimal) en ASCII se trouve en séquence avant le A (65 en décimal) alors qu'en EBCDIC le 0 (240 en décimal) se trouve après le Z (233 en décimal).

C'est une caractéristique qu'il faut garder à l'esprit lorsqu'il y a des tests de supériorité ou d'infériorité à effectuer sur des zones pouvant contenir des chiffres et des lettres : en ASCII un libellé "Bonjour" est supérieur à n'importe quel nombre alors qu'en EBCDIC ce sont les nombres qui sont supérieurs au libellé "Bonjour".

ASCII et EBCDIC sont déclinés en plusieurs versions spécifiques – notamment – à chaque pays (chaque langue). On parle de *Coded Character Set Identifier* (CCSID) ou code page de caractères, ou charsets. À titre indicatif le CCSID 500 correspond à l'EBCDIC pour l'Europe de l'Ouest alors que le 297 est celui généralement utilisé en France. Il existe une nouvelle version 1147 incluant le signe de l'Euro. Ce ne sont maintenant plus des normes ASCII qui sont publiées, mais des normes ISO. Depuis 1991 le consortium international UNICODE a pour objectif d'assigner à chaque caractère un identifiant unique pour tous les CCSID utilisés de par le monde : vous trouverez des tables EBCDIC mentionnant cette équivalence. Par exemple, pour la lettre "a" minuscule le code UNICODE est le LA010000 (EBCDIC : hexa 81, ASCII : hexa 61) et pour le "A" majuscule le LA020000.

Il faut tenir compte de ce CCSID pour :

- envoyer des données vers un autre environnement,
- interroger des bases de données : elles peuvent ne pas avoir le même CCSID que le programme.

1.2 Déclaration des données

L'une des particularités du COBOL est que toute zone utilisée dans un programme – variable ou constante – doit être déclarée, c'est-à-dire décrite avec précision : nom, format (numérique ou alphanumérique...) et longueur.

Cette déclaration doit être faite en DATA DIVISION.

La DATA DIVISION est scindée en plusieurs sections sur le même principe que la PROCEDURE DIVISION, mais en DATA DIVISION les noms des sections sont des mots COBOL réservés et invariants.

Comme pour toute règle il y a cependant des exceptions : certaines zones ne nécessitent pas de déclaration car elles sont mises à disposition du développeur par COBOL :

- les constantes figuratives,
- les registres spéciaux.

Elles seront étudiées plus loin dans ce chapitre.

Le choix de la section pour effectuer la description dépend de son usage :

- FILE SECTION : pour les fichiers. Leur utilisation est détaillée dans le prochain chapitre.
- LINKAGE SECTION : pour les données reçues dans un sous-programme. C'est avec l'étude des sous-programmes dans le chapitre Techniques avancées que cette section est décrite.
- WORKING-STORAGE SECTION : toutes les autres zones doivent être déclarées dans cette section. C'est plus spécifiquement cette section qui est abordée dans ce chapitre.

■ Remarque

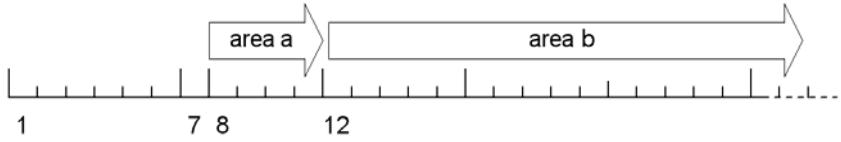
Toute zone COBOL est disponible pour l'ensemble du programme, où que soit codée sa déclaration.

Syntaxe

■ Remarque

Quelle que soit la section où se trouve la déclaration d'une donnée, elle répond toujours aux mêmes règles syntaxiques.

L'area utilisée dans les manuels COBOL est le regroupement des colonnes délimitées par les positions 8 et 12. La première plage, ou area A, commence en 8e position et va jusqu'à la 11e. La deuxième, ou area b, va de la 12e position à la 72e.



Ces areas permettent d'indiquer les plages de positions que doivent respecter certaines instructions.

De manière identique aux sections de la PROCEDURE DIVISION constituées de paragraphes, les sections de la DATA DIVISION sont constituées de niveaux principaux caractérisés par un numéro et un emplacement. Le numéro est un numéro de niveau hiérarchique :

Syntaxe :

```
----+----1----+----2----+----3----+----4----+----5----+----6---...
       N° Nom-zone  PIC picture .
```

■ Remarque

Le point en fin de ligne est obligatoire.

- N° de *niveau* hiérarchique : il est OBLIGATOIRE. Les niveaux **01** et **77** ne doivent commencer qu'en area a. Les autres numéros de niveau hiérarchique **02** à **49** peuvent par contre commencer indifféremment en area a ou b.
- Nom-zone : le nom de la zone doit être séparé de son numéro de niveau hiérarchique par au moins un espace. Il répond aux mêmes contraintes que tous les mots COBOL définis par le programmeur. Il peut faire jusqu'à 30 caractères parmi :
  - les chiffres de 0 à 9,
  - les lettres de "A" à "Z" et de "a" à "z",
  - et le tiret -
  - mais ne peut ni :
  - commencer par un chiffre ou un tiret,
  - se terminer par un tiret,
  - contenir de caractère accentué.

Tout comme pour les noms de paragraphes en PROCEDURE DIVISION, plusieurs zones peuvent avoir le même nom pourvu qu'elles appartiennent à des niveaux hiérarchiques différents.

Exemple :

```
01  ZONE-A.
        05  NOM                 PIC X(20).
        05  PRENOM              PIC X(10).

    01  ZONE-B.
        05  NOM                 PIC X(20).
        05  PRENOM              PIC X(10).
```

- `picture` : aussi appelée clause PICTURE. La clause PICTURE permet de définir un format de donnée.

D'autres clauses sont également disponibles : elles sont expliquées plus loin.

1.2.1 Clause PICTURE

Il s'agit de la contribution la plus importante du langage COMTRAN de Robert BEMER à la conception du COBOL. Elle permet de définir de façon très précise :
- le format d'une zone : numérique, alphanumérique...,
- sa longueur.

Optionnelle, elle est constituée :
- d'un caractère indiquant le format de la donnée : X pour une donnée alphanumérique, 9 pour une données numérique...,
- de la longueur de la donnée.

Exemples :

```
       01 ws-adr                PIC X(20).
```

`ws-adr` est une zone alphanumérique de 20 caractères pouvant par exemple contenir "`12 rue du Paradis`".

```
       01 ws-nb-pages           PIC 9(5).
```

`ws-nb-pages` est une zone numérique de 5 chiffres pouvant par exemple contenir `51163`.

```
       01 ws-no-page            PIC 999.
```

`ws-no-page` est une zone numérique de 3 chiffres pouvant par exemple contenir `461`.
- et d'un usage : ce point est abordé lors de la déclaration des zones numériques.

Il existe d'autres caractères pour d'autres formats, pas forcément tous très utiles, ainsi que des PICTURE spécifiques à la production d'états, ou PICTURE d'édition, indispensables pour la mise en forme des zones numériques.

La longueur de la zone peut être indiquée entre parenthèses, ou le caractère du format être dupliqué.