

VisualAge Pacbase



DBD RELATIONNELLES SQL

Version 3.5



VisualAge Pacbase



DBD RELATIONNELLES SQL

Version 3.5

Note

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Notices», à la page v.

Vous pouvez consulter ou télécharger la documentation de VisualAge Pacbase, régulièrement mise à jour, à partir de :

<http://www.ibm.com/support/docview.wss?rs=37&context=SSEP67&uid=swg27005478>

La section Catalogue dans la page d'accueil de la Documentation vous permet d'identifier la dernière édition disponible du présent document.

Première édition (Septembre 2006)

La présente édition s'applique à :

- VisualAge Pacbase Version 3.5

Vous pouvez nous adresser tout commentaire sur ce document (en indiquant sa référence) via le site Web de notre Support Technique à l'adresse suivante : <http://www.ibm.com/software/awdtools/vapacbase/support.html> ou en nous adressant un courrier à :

IBM Paris Laboratory
1, place Jean-Baptiste Clément
93881 Noisy-le-Grand, France.

IBM pourra disposer comme elle l'entendra des informations contenues dans vos commentaires, sans aucune obligation de sa part.

© Copyright International Business Machines Corporation 1983,2006. All rights reserved.

Table des matières

Notices	v	Chapitre 7. Exemples d'éléments de génération	85
Marques	vii	DB2	85
Chapitre 1. Introduction	1	SQL SERVER	91
Présentation du module DBD	1	DATACOM/DB	95
Principes de description	1	SQL/DS	101
Chapitre 2. Utilisation du module avec SQL	3	NONSTOP SQL	106
Introduction	3	ORACLE (V5,V6 et V7)	110
Utilisation des entités VisualAge Pacbase	3	RDMS	129
Chapitre 3. Colonnes : Rubriques	7	INTEREL RDBC	134
Définition d'une colonne (E.....)	7	INTEREL RFM	139
Description d'une colonne (E.....D)	21	DB2/2 ET DB2/6000	143
Chapitre 4. Tables et Vues : Segments	31	SYBASE	147
Définition d'une Table / Vue (S....)	31	Chapitre 8. Génération d'un Bloc	153
Description d'une Table / Vue (S....CE)	36	Génération du DDL (B.....GN)	153
Complément de description d'une Vue (S....DBE)	45	Mise à jour du catalogue	156
Chapitre 5. Accès SQL	49	Chapitre 9. Mode d'accès	159
Aide à la prise en compte des accès SQL	49	TP	159
Personnalisation des accès SQL	50	Edition génération	162
Introduction	50	Chapitre 10. Exemples	165
Description	50	Ecrans communs	165
Prise en compte	57	DB2	172
Exemples	60	SQL SERVER	180
Chapitre 6. Bases de Données : Blocs	65	DBD/2	185
Définition d'une Base de Données (B.....)	65	DATACOM/DB	190
Description d'une Base de Données (B.....DR)	71	NONSTOP SQL	198
Composition de la clé/d'un Alter Table (-DRnnnK)	76	ORACLE (<V6)	203
Eléments et options de génération	79	ORACLE V7	211
Procédures SQL	84	RDMS	221
		SQL/DS	229
		INTEREL RDBC	237
		INTEREL RFM	242
		SYBASE	247

Notices

Ce document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM. Cela ne signifie pas qu'IBM ait l'intention de les annoncer dans tous les pays où la compagnie est présente. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante : IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk NY 10504-1785, U.S.A.

Les détenteurs de licences du présent produit souhaitant obtenir des informations sur celui-ci à des fins : (i) d'échange d'informations entre des programmes développés indépendamment et d'autres programmes (y compris celui-ci) et (ii) d'utilisation mutuelle des informations ainsi échangées doivent s'adresser à : IBM Paris Laboratory, SMC Department, 1 place J.B.Clément, 93881 Noisy-Le-Grand Cedex, France. De telles informations peuvent être mises à la disposition du Client et seront soumises aux termes et conditions appropriés, y compris dans certains cas au paiement d'une redevance.

IBM peut modifier ce document, le produit qu'il décrit ou les deux.

Marques

IBM est une marque d'International Business Machines Corporation, Inc. AIX, AS/400, CICS, CICS/MVS, CICS/VSE, COBOL/2, DB2, IMS, MQSeries, OS/2, VisualAge Pacbase, RACF, RS/6000, SQL/DS et VisualAge sont des marques d'International Business Machines Corporation, Inc. dans certains pays.

Java et toutes les marques et logos incluant Java sont des marques de Sun Microsystems, Inc. dans certains pays.

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation dans certains pays.

UNIX est une marque enregistrée aux Etats-Unis et/ou dans d'autres pays et utilisée avec l'autorisation exclusive de la société X/Open Company Limited.

D'autres sociétés peuvent être propriétaires des autres marques, noms de produits ou logos qui pourraient apparaître dans ce document.

Chapitre 1. Introduction

Présentation du module DBD

Le Module DESCRIPTION DE BASES DE DONNEES (DBD) assure la génération automatique des descriptions de bases de données adaptée aux SGBD utilisés à partir des descriptions de Segments et de Liens effectuées en phase d'analyse.

Il traite les catégories de bases de données suivantes :

- Les bases de données relationnelles,
- Les bases de données en réseau (type CODASYL),
- Les bases de données hiérarchiques (type DL/1),
- Les bases de données type fichier physique AS400 et DDL TANDEM,
- Les bases de données TurboImage,
- Les bases de données DMSII.

Ces catégories font chacune l'objet d'un Manuel de Référence propre.

DBD RELATIONNELLES SQL

Ce module ne peut être utilisé qu'en liaison avec le module DICTIONNAIRE : les données définies dans le Dictionnaire (qu'on utilise ou non le module METHODE) peuvent être réutilisées sans modification pour la génération et la description de la base de données.

Ces informations sont décrites à l'aide d'un langage de description de données indépendant du système de gestion de base de données utilisé, ce qui permet d'obtenir des générations différentes à partir d'une même source.

Principes de description

La description des entités et des lignes gérées par VisualAge Pacbase est en général composée de deux parties :

- Une introduction expliquant le but et les caractéristiques générales de l'entité ou de la ligne.
- Une description exhaustive, pour chaque ligne, des zones d'entrée dans les écrans.

Pour la description de ces entrées en batch, reportez-vous au manuel des "Procédures du Développeur".

Dans la description, chaque zone est repérée par un numéro d'ordre correspondant à l'ordre de tabulation sur l'écran.

NOTE : Si vous utilisez la Station de Travail VisualAge Pacbase, il est recommandé de consulter le guide de "l'Interface Utilisateur Station de Travail" dans lequel sont documentées les fenêtres de la Station.

REMARQUES :

Chaque type de Bloc Base de Données possède une description spécifique. Certaines descriptions correspondent à un même bordereau batch.

Une même Rubrique peut donc avoir des significations différentes ou ne pas être utilisée, en fonction du type de Bloc décrit.

Chapitre 2. Utilisation du module avec SQL

Introduction

Le Module DBD SQL permet d'utiliser le Système pour décrire des bases de données pour les SGBD utilisant le langage SQL : il permet la génération du langage de description (DDL) des Blocs Bases de Données et de leur différents objets (ESPACES, TABLES, VUES et INDEX) en batch ou en TP.

Le Manuel Description de Bases de Données SQL n'est pas un manuel de formation aux techniques de gestion des bases de données.

Une connaissance préalable des bases de données relationnelles et du module DICTIONNAIRE est indispensable.

Le but de ce manuel est de vous guider au cours de la description d'une base de données relationnelle dans le Dictionnaire.

Utilisation des entités VisualAge Pacbase

ROLE DU DICTIONNAIRE

Le rôle du Dictionnaire est de gérer une description logique des diverses vues externes à délivrer aux programmes. Pour cela, il utilise les entités suivantes :

- les Rubriques,
- les Segments,
- les Blocs Base de Données,
- les Lignes de Commentaires, d'Éléments et Options de Génération associées aux Segments et aux Blocs Base de Données, ainsi que les lignes de Génération de l'aide en ligne associées aux Segments.
- les Formats-Guides.

CORRESPONDANCE TERMINOLOGIQUE

RAPPEL :

Une base de données SQL est un ensemble d'environnements physiques (TABLESPACE, DBSPACE, SPACE, STORAGE-AREA, etc... appelés ESPACES dans ce manuel) comprenant des TABLES. Une TABLE est un ensemble de COLONNES.

Un Bloc Base de Données est un ensemble de Segments, chaque Segment appelant lui-même des Rubriques.

A partir de ce parallélisme, il est possible d'établir le tableau suivant :

OBJET SQL	EQUIVALENT
Base	Bloc Base de Données de type Qn
Espace	Ligne de description de B.D. de type P
Table	Segment appelé sur une ligne de type T
Vue	Segment appelé sur une ligne de type V
Colonne	Rubrique
Index	Rubrique(s) du Segment appelé sur une ligne de type I (écran -DR)
Primary Key	Rubrique(s) du Segment appelé sur une ligne de type K (écran -DR)
Foreign Key	Rubrique(s) du Segment appelé sur une ligne de type J (écran -DR)
Package	Rubrique(s) du Segment appelé sur une ligne de type C (écran -DR)
Fonction	Rubrique(s) du Segment appelé sur une ligne de type E (écran -DR)
Procédure	Rubrique(s) du Segment appelé sur une ligne de type Q (écran -DR)
Trigger ou Rule	Rubrique(s) du Segment appelé sur une ligne de type R (écran -DR)

GENERATION D'UNE BASE DE DONNEES

Une base de données est générée à partir d'un Bloc Base de Données.

En reprenant toutes les informations dont il dispose au niveau du Dictionnaire (informations de niveau logique) et en fonction du type de Bloc, le module assure alors au niveau Bloc la génération des lignes de langage de description de données (DDL) correspondant au type du Bloc Base de Données.

Pour chaque ligne de description des Tables, Vues et Index, le Système génère le DDL en fonction des données que vous lui fournissez à l'aide des lignes de description de Segments et des lignes de définition de Rubriques.

Vous pouvez remplacer ou compléter ces lignes générées à l'aide des lignes d'Options de Génération (-GO), d'Eléments de Génération (-GG), de Commentaires (-GC).

MISE EN OEUVRE

Les ordres SQL sont générés à partir de la description du Bloc (B.....DR), où sont appelés des Segments (qui contiennent la description des Tables et des Vues).

Les Rubriques appelées dans le Segment seront éventuellement reprises dans les Index.

Cet écran permet également la définition des Espaces dont la description est effectuée sur des lignes d'Eléments de génération (B.....DRnnnGG).

Notes : nnn correspond au numéro de ligne de l'écran de description.

Les espaces ne sont pas pris en compte pour INTEREL RDBC, INTEREL RFM, NON STOP SQL, DB2/2, DB2/6000, SYBASE et SQL SERVER.

A partir de l'écran de description sont obtenus :

- Un écran de saisie et de consultation du contenu de l'Index, des Primary et Foreign keys, ou des Colonnes modifiées, ajoutées ou supprimées d'une Table (B.....DRnnnK).
- Un écran de saisie des informations nécessaires à la génération des ordres SQL (telles que la sélection des Colonnes à ajouter à une Table), (B.....GN, B.....GNnnn, ou après saisie d'un Q dans la zone code Action de l'écran de description).
- A partir de ce dernier écran, visualisation des ordres SQL générés (touche ENTREE sur l'écran B.....GN).

Sur les écrans de Commentaires, Eléments et Options de Génération, des lignes guides et des lignes virtuelles vous aident dans la saisie des informations; l'appel de Format-Guides facilite également cette saisie.

Chapitre 3. Colonnes : Rubriques

Définition d'une colonne (E.....)

Une Colonne étant assimilable à une Rubrique, elle est définie à l'aide d'une ligne de définition de Rubrique, obtenue par le choix :

CH: E.....

LIGNES EVENTUELLEMENT ASSOCIEES

Commentaires (obtenus par le choix CH: -GC).

NOM DE LA COLONNE

Il est possible d'indiquer un nom de Colonne lors de l'appel de la Rubrique dans un Segment : c'est ce nom qui est retenu en priorité (voir le chapitre "Table ou Vue", sous-chapitre "Description d'un Segment"); ce nom peut également être indiqué sur une ligne de description de la Rubrique de type 'R' (voir le sous-chapitre "Description d'une Rubrique").

Sinon, le code de la Rubrique sera utilisé.

FORMATS

C'est à partir du format interne que le système génère le type et la longueur de la donnée.

Les tableaux suivants décrivent, par type de base de données, les formats générés à partir des formats renseignés sur les lignes de définition des Rubriques. Les Rubriques définies avec d'autres formats ne sont pas prises en compte à la génération.

Une CHAR(n) est générée pour les formats date du Système (M, G, D, I, E, S, C, T, TS) qui ne génèrent pas un format date SQL.

Si vous saisissez un format date non-standard sur le -D de la Rubrique (sur une ligne de type E), un format DATE est généré pour ORACLE, SYBASE et SQL SERVER.

DB2, SQL/DS

FORMAT SQL	FORMAT	USAGE	REMARQUES
CHAR(n)	X(n) avec n<255	D	

FORMAT SQL	FORMAT	USAGE	REMARQUES
VARCHAR(n)	X(n) avec n<255	D	Voir ci-après
LONG VARCHAR	X(n) avec n>254	D	Voir ci-après
SMALLINT	S9(n) avec n<5	C	
INTEGER	S9(n) avec n>4	C	
DECIMAL(n+m,m)	S9(n)V9(m)	3,1	Usage 1 pour DB2 uniquement
FLOAT	Pas de format	W	
DATE	M, G	D	
TIME	T	D	
TIMESTAMP	TS	D	
REAL	Pas de Format	F	DB2 uniquement

DB2/2, DB2/6000

FORMAT SQL	FORMAT	USAGE	REMARQUES
CHAR(n)	X(n)	D	
VARCHAR(n)	X(n) avec n<4001	D	Voir ci-après
LONG VARCHAR	X(n) avec n>4000	D	Voir ci-après
SMALLINT	S9(n) avec n<5	C, ou G	
INTEGER	S9(n) avec n>4	C, ou G	
DECIMAL(n+m,m)	S9(n)V9(m)	3,1	
FLOAT	Pas de format	W	
DATE	M, G	D	
TIME	T	D	
TIMESTAMP	TS	D	

ORACLE < V7

FORMAT SQL	FORMAT	USAGE	REMARQUES
CHAR(n)	X(n)	D	V5 : n<241 V6 : n<2001
LONG VARCHAR	X(n)	D	n<65536 voir ci-après
INTEGER	S9(n)	C,J,P,5,6	Longueur<=4
NUMBER(n+m,m)	S9(n)V9(m)	T,Q,3,8,9	
FLOAT	Pas de format	W	

FORMAT SQL	FORMAT	USAGE	REMARQUES
DATE	D,I,E,S,C,M,G,T	D	

RDMS 1100

FORMAT SQL	FORMAT	USAGE
CHAR(n)	X(n)	D
DECIMAL (n+m+1,m)	S9(n)V9(m)	D,C,3,H

A PARTIR DE LA VERSION 5RA4

FORMAT SQL	FORMAT	USAGE
CHAR (n)	X(n)	D
DECIMAL (n+m+1,m)	S9(n)V9(m)	3, C
NUMERIC (n+m,m)	S9(n)V9(m) 9(n)V9(m)	H ou D
FLOAT	Pas de format	F COMP-1
DOUBLE PRECISION	Pas de format	W COMP-2

ORACLE V7

FORMAT SQL	FORMAT	USAGE	REMARQUES
CHAR(n)	X(n)	D	n<2000 et indic = 'C'
VARCHAR2(n)	X(n)	D	n<2000
VARCHAR(n)	X(n)	D	n<2000 et indic='V'ou'W'
LONG VARCHAR	X(n)	D	n<65536 et indic = 'L'
INTEGER	S9(n)	C,J,P,5,6	
NUMBER(n+m,m)	S9(n)V9(m)	T,Q,3,8,9	
FLOAT	Pas de format	W	
DATE	D, I, E, S, C, M, G, T	D	

DATACOM/DB

FORMAT SQL	FORMAT	USAGE	REMARQUES
CHAR (n)	X(n)	D	avec n<32768
NUMERIC(n+m,m)	S9(n)V9(m)	D	

FORMAT SQL	FORMAT	USAGE	REMARQUES
DECIMAL(n+m,m)	S9(n)V9(m)	3	
INTEGER	S9(n) 4<n<9	C	
SMALLINT	S9(n) n<5	C	
DOUBLE PRECISION	Pas de Format	W	
DATE	M	D	
TIME	T	D	
TIMESTAMP	TS	D	

NON STOP SQL

FORMAT SQL	FORMAT	USAGE	REMARQUES
CHAR(n)	X(n)	D	n<4075
VARCHAR(n)	X(n)	D	V dans -CE
DECIMAL(n+m,n)	S9(n)V9(m)	D,1	
DECIMAL(n+m,n) unsigned	9(n)V9(m)		
NUMERIC(n+m,n)	S9(n)V9(m)	C	
NUMERIC(n+m,n) unsigned	9(n)V9(m)		
SMALLINT	S9(n)	C	n<5
SMALLINT unsigned	9(n)		
INTEGER	S9(n)	C	n<10
INTEGER unsigned	9(n)		
LARGEINT	S9(n)	C	n<19
DATE	M	D	version C30
TIME	T	D	version C30
TIMESTAMP	TS	D	version C30

INTEREL RDBC

FORMAT SQL	FORMAT	USAGE	REMARQUES
CHAR(n)	X(n)	D	
VARCHAR(n)	X(n) avec n<32000	D	V dans -CE voir ci-après

FORMAT SQL	FORMAT	USAGE	REMARQUES
LONG VARCHAR	X(n) avec n=32000	D	V dans -CE voir ci-après
SMALLINT	Pas de format	0 (zéro)	
INTEGER	Pas de format	J	
DECIMAL(n+m,m)	S9(n)V9(m)	D,1	
REAL	Pas de format	F	
DATE	E,M	D	
FLOAT	Pas de format	W	

INTEREL RFM

FORMAT SQL	FORMAT	USAGE	REMARQUES
CHAR(n)	X(n)	D	n<4096
SMALLINT	S9(n)	0 (zéro)	n<5
INTEGER(n)	S9(n)	J	4<n<10
DECIMAL(n+m,m)	S9(n)V9(m)	D	
FLOAT	Pas de format	W	
REAL	Pas de format	F	
DATETIME			
YEAR TO DAY	G	D	format de 8 caractères
TIME	T	D	
TIMESTAMP	TS	D	

SYBASE

FORMAT SQL	FORMAT	USAGE	REMARQUES
CHAR(n)	X(n)	D	n<255
VARCHAR(n)	X(n)	D	n<255
SMALLINT	S9(n) ou 9(n)	C, G	n<5
INTEGER	S9(n) ou 9(n)	C, G	n<9
NUMERIC (m,n)	S9(m-n)V9(n) ou 9(m-n)V9(n)	D,3,1,X	
DECIMAL(n,m)	S9(n)V9(m)	C	
FLOAT	Pas de format	W	

FORMAT SQL	FORMAT	USAGE	REMARQUES
REAL	Pas de format	F	
DATETIME	T,M,S,E,I,G	D	

SQL SERVER

FORMAT SQL	FORMAT	USAGE	REMARQUES
CHAR(n)	X(n)	D	n<255
VARCHAR(n)	X(n)	D	n<255
SMALLINT	S9(n) ou 9(n)	C, G	n<5
INTEGER	S9(n) ou 9(n)	C, G	n<9
NUMERIC (m,n)	S9(m-n)V9(n)	D,3,1,X	
	ou 9(m-n)V9(n)		
DECIMAL(n,m)	S9(n)V9(m)	C	
FLOAT	Pas de format	W	
REAL	Pas de format	F	
DATETIME	T,M,S,E,I,G	D	

COLONNES VARIABLES

Pour générer une Colonne variable d'une Table, vous devez définir une Rubrique de format alphanumérique X(n). Sur la ligne d'appel de celle-ci dans le Segment Table (-CE), vous indiquez qu'il s'agit d'un champ de longueur variable en positionnant un 'V' dans la zone INDICATIF. Le Système génère alors :

VARCHAR(n).

Note : Ce type de colonne est généré uniquement, pour DB2, SQL/DS, INTEREL RDBC, NONSTOP SQL, DB2/2, DB2/6000, SYBASE et SQL SERVER.

Pour DB2 et SQL/DS, si n est supérieur à 254 le Système génère alors :

LONG VARCHAR.

Pour DB2/2 et DB2/6000, si n est supérieur à 4000 le système génère alors :

LONG VARCHAR.

Les Host variables correspondantes générées par les modules Pacbench, Dialogue et Batch sont traitées comme des chaînes de caractères de longueur variable.

RDMS 1100, DATACOM/DB :

Il n'est pas possible de générer des Colonnes variables.

DB2, SQL/DS, DB2/2 et DB2/6000 :

- La valeur W génère une VARCHAR.
- La valeur L génère une LONG VARCHAR.

ORACLE V5 et V6 :

- La valeur V génère uniquement une VARCHAR.
- La valeur L génère une LONG VARCHAR.
- La valeur W génère une VARCHAR.

ORACLE V7 :

- La valeur V ou W génère une VARCHAR(n).
- Un champ de longueur <2000 et 'Espace' dans l'indicatif génèrent une VARCHAR2(n).
- La valeur L génère une LONG VARCHAR.
- La valeur C génère une CHAR.

COLONNES OBLIGATOIRES

Sur la ligne d'appel de la Rubrique dans le Segment Table (-CE), vous indiquez si vous voulez générer une Colonne obligatoire (NOT NULL), ou obligatoire avec valeurs par défaut.

Dans la zone PRESENCE EN CREATION de la description du Segment Table (-CE), vous précisez :

0 -> NOT NULL
P -> NOT NULL WITH DEFAULT (pour DB2,
DATACOM/DB, DB2/2 et DB2/6000)
DEFAULT (pour INTEREL RDBC)
DEFAULT SYSTEM (pour NONSTOP SQL C30)

SQL/DS, ORACLE, NONSTOP SQL, SYBASE :

Il n'est pas possible de générer une clause (NOT NULL WITH) DEFAULT. La valeur 'P' n'est pas prise en compte.

RDMS 1100 :

Aucun type de Colonne obligatoire ne peut être généré. Les valeurs O et P sont ignorées.

SQL SERVER :

Vous saisissez dans la zone PRESENCE EN CREATION :

- O -> NOT NULL
- P -> NOT NULL DEFAULT SPACE (nn) pour un champ alphanumérique.
- > NOT NULL DEFAULT 0 pour un champ numérique.

FICHE DE LA RUBRIQUE 1 REFCLI

NOM.....: 2 REFERENCE CLIENT

TYPE.....: 3 R

FORMAT D'ENTREE....: 4 X(30)

FORMAT INTERNE.....: 5 X(30)

FORMAT D'EDITION...: 7 X(30)

USAGE : 6 D

0: 8

LONGUEUR : 30

LONGUEUR : 30

LONGUEUR : 30

MOTS CLES ASSOCIES.: 9

RUBRIQUE MERE.....: 10

NO DE SESSION.....: 0067

BIBLIOTHEQUE : BMS

BLOCAGE :

0: C1 CH: E refcli

ACTION:

NUM	CLASSE	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
1	6	VALEUR	REPLISSAGE
1	6		CODE RUBRIQUE (OBLIGATOIRE)
			Code alphanumérique qui repère la Rubrique indépendamment de toute appartenance à une Structure de Données, à un Etat ou à un Ecran.
			Il est donc inutile d'inclure un code Segment, Etat ou Ecran dans le code Rubrique : ce code sera automatiquement ajouté lors de la génération.
			Certains codes Rubriques ont un emploi réservé dans les Structures de Données, les Etats ou les Ecrans. Leur définition dans le Dictionnaire est donc interdite :
			. ENPR, GRPR, ERUT, FILLER, SUITE pour les Structures de Données,
			. LIGNE pour la Structure '00' d'un Etat,
			. SAUT pour la Structure '00' d'un Etat. Se reporter à la colonne STRUCTURE de l'écran Structures de l'Etat (-CE),
			. LSKP est l'équivalent du code rubrique SAUT et est utilisé dans la version anglaise du Système.
			Options des modules Dialogue et Pacbench C/S (Clients TUI) :

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		. LIERR pour les messages d'erreurs,
		. ERMSG est l'équivalent du code Rubrique LIERR et est utilisé dans la version anglaise du Système,
		. PFKEY est utilisé dans les Ecrans pour représenter les touches fonction programmables.
		. *PASWD (IMS seulement) est utilisé pour les mots de passe d'un écran.
		Le code des Rubriques fournies par IBM commence par un point ".". N'utilisez donc pas de codes qui commencent par un point pour définir vos Rubriques.
		Pour de plus amples informations, voir la zone CODE RUBRIQUE OU CODE DE L'ECRAN A APPELER de l'écran Description d'un Ecran (O.....CE).
2	36	NOM DE LA RUBRIQUE (OBL. EN CREATION)
		Ce nom doit être aussi explicite que possible. Il apparaît dans la documentation lors de chaque utilisation de la Rubrique et en particulier dans la Documentation Utilisateur.
		Il permet aussi la constitution automatique de Mots- Clés dans les conditions détaillées dans le "Guide de l'interface utilisateur mode caractère", chapitre "Recherche d'instances", sous-chapitre "Recherche par Mot-Clé".
		Il convient également de penser à la possibilité d'obtenir la liste des Rubriques classées par nom.
		NB : pour IMS, utiliser des majuscules.
3	1	TYPE (PROPRIETE, RUBRIQUE OU ALIAS)
		'P'
		Propriété : information élémentaire définie au niveau conceptuel.
		Remarque : le format est facultatif.
		'R'
		Rubrique : information élémentaire définie au niveau du Dictionnaire (valeur par défaut).
		Module DBD : caractéristique SOCRATE/CLIO, donnée élémentaire CODASYL, colonne Relationnelle.
		'A'
		Rubrique de type ALIAS : ce type est positionné de façon automatique lors d'un passage "Reprise de l'Existant".
		'L'
		Rubrique de type Large Object
4	10	FORMAT D'ENTREE
		Non utilisé pour les bases de données.

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
5	10	Format interne
		C'est le format normalement utilisé pour les fichiers de l'application (fichiers permanents, base de données, fichiers temporaires) et pour les zones de saisie des écrans ; le format est automatiquement reporté dans la description des segments de données ; le choix définitif est fait lors de l'appel des Segments dans les Programmes ; pour les Programmes batch, vous pouvez sélectionner le type du format sur l'écran Appel de Structures de Données du Programme (-CD) ; il est également reporté, avec transformation, dans la description des écrans (zones de saisie) (Voir les Manuels Dialogue et Pacbench C/S (Clients TUI) ; il doit être codé comme une picture Cobol, sans caractères d'édition ; la zone 'Usage interne' concerne ce format seul
		Formats symboliques pour les dates :
	D	Date sans siècle (JJMMAA ou MMJJAA)
	I	Date sans siècle (AAMMJJ)
	E	Date sans siècle (JJ/MM/AA ou MM/JJ/AA)
	S	Date avec siècle (SSAAMMJJ)
	C	Date avec siècle (JJMMSSAA ou MMJJSSAA)
	M	Date avec siècle (JJ/MM/SSAA ou MM/JJ/SSAA)
	G	Format grégorien (SSAA-MM-JJ)
	T	Format TIME (HH:MM:SS)
	TS	Format TIMESTAMP
		Module Méthode : ce format peut être omis pour une propriété
		Pour les formats incluant un séparateur (E, G, M, T), vous pouvez indiquer, à la suite du caractère représentant le format, un caractère représentant un séparateur différent de celui prévu par défaut dans le format. Par exemple, vous pouvez indiquer "G/" dans cette zone pour générer un format grégorien SSAA/MM/JJ.
		Pour l'utilisation des formats avec les différents types de blocs bases de données, se reporter aux tableaux récapitulatifs du chapitre "Colonnes : Rubriques" dans le Manuel DBD relationnelles SQL.
6	1	USAGE INTERNE
		Correspond à la clause USAGE du COBOL.

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
	'D'	DISPLAY (option par défaut), tout matériel.
		L'usage DISPLAY est l'option obligatoire pour les Rubriques représentant des dates, ainsi que pour les Rubriques servant à générer des caractéristiques SOCRATE/CLIO de type numérique entier borné.
	'C'	COMPUTATIONAL type IBM ou équivalent; COMPUTATIONAL-4 IBM 38; COMPUTATIONAL-4 IBM 3-15D, COMPUTATIONAL-6 ICL 2900; BINARY type IBM et variante Cobol II. L'usage C est l'option obligatoire pour les Rubriques servant à générer des caractéristiques SOCRATE/CLIO de type numérique entier borné binaire.
	'R'	COMPUTATIONAL SYNCHRONIZED RIGHT, type IBM ou équivalents. Cet usage est recommandé, de préférence au précédent, lorsque les données binaires sont cadrées sur des adresses paires, car les instructions COBOL les concernant sont plus performantes.
	'B'	COMPUTATIONAL-1 ICL 1900. BINARY-1 UNISYS 1100 associé au format 1(n).
	'S'	COMPUTATIONAL SYNCHRONIZED RIGHT ICL 1900.
	'N'	COMPUTATIONAL UNISYS-A. COMP-4 cadré sur un demi-octet. En cas de longueur impaire, le complément n'est pas généré; vous devez donc l'ajouter.
	'P'	COMPUTATIONAL-1 BULL 66, 6000 et DPS8.
	'L'	COMPUTATIONAL-1 SYNCHRONIZED RIGHT ICL 1900.
	'Q'	COMPUTATIONAL BULL 66, 6000 et DPS8.
	'F'	COMPUTATIONAL-1 type IBM. COMPUTATIONAL-9 BULL DPS7. COMPUTATIONAL-11 BULL 66 et DPS8. DBD Relationnelles : virgule flottante simple précision.
	'T'	COMPUTATIONAL-3 PACKED SYNC BULL 66 et DPS8.
	'X'	DISPLAY SIGN IS TRAILING SEPARATE CHARACTER.
	'G'	COMPUTATIONAL SYNCHRONIZED RIGHT ICL 2900 AND COMPUTATIONAL-5 MICROFOCUS.
	'7'	COMPUTATIONAL-5 ICL 2900.
	'K'	COMPUTATIONAL CDC. COMPUTATIONAL UNISYS 1100 (COBOL 85)
	'M'	COMPUTATIONAL-1 CDC.
	'N'	COMPUTATIONAL UNISYS-A
	'O'	COMPUTATIONAL-4 UNISYS 1100

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
	'U'	COMPUTATIONAL-1 UNISYS 1100.
	'W'	COMPUTATIONAL-2 UNISYS 1100. COMPUTATIONAL-12 BULL 66 et DPS8. DBD Relationnelles : virgule flottante double précision.
	'H'	COMPUTATIONAL UNISYS 1100. BINARY UNISYS 1100 (COBOL 85)
	'8'	COMPUTATIONAL BULL 66 COBOL 74 et DPS8.
	'9'	COMPUTATIONAL-3 BULL 66 COBOL 74 DPS7 et DPS8.
	'J'	COMPUTATIONAL-6 BULL 66 COBOL 74 DPS7 et DPS8. REAL UNISYS-A.
	'Y'	DB-KEY BULL 66 DM4 et DPS8. POINTER IBM et MICROFOCUS.
	'I'	DISPLAY-1 UNISYS 1100.
	'5'	COMPUTATIONAL-1 BULL 64 66 MINI-6 COBOL 74 DPS7 DPS8.
	'6'	COMPUTATIONAL-2 BULL 64 66 MINI-6 COBOL 74 DPS7 DPS8.
	'3'	COMPUTATIONAL-3 type IBM ou équivalent. COMPUTATIONAL BULL 64 MINI-6 DPS7 PACKED-DECIMAL UNISYS 1100 (COBOL 85)
		DBD SOCRATE/CLIO : Pour une Rubrique servant à générer une caractéristique SOCRATE/CLIO de type numérique entier définie sous la forme décimale Nv0, l'usage doit obligatoirement être 3.
	'0'	COMPUTATIONAL-7 BULL 66 et DPS8.
	'1'	DISPLAY-1 NCR (décimal étendu signé sur l'octet). DISPLAY SIGN LEADING SEPARATE - DPS8, IBM, UNISYS 1100, TANDEM, DPS7.
	'4'	DISPLAY-2 NCR (décimal condensé non signé).
	'2'	DISPLAY-2 BULL = DISPLAY, la comparaison de zones de ce type se fait suivant la "commercial collating sequence" et non suivant la séquence standard BULL.
	'Z'	Option réservée au batch : permet de faire générer une clause 'BLANK WHEN ZERO', cette option n'étant utilisable qu'avec un format d'édition.
		Module METHODE : ce format peut être omis pour une Propriété.
7	27	FORMAT D'EDITION

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		Non utilisé pour les bases de données.
8	1	GENERATION CLAUSE BLANK WHEN ZERO
		Cette zone n'est pas utilisée lors de la définition d'une Rubrique servant à générer une donnée élémentaire CODASYL ou une colonne relationnelle.
9	55	MOTS CLES ASSOCIES
		Cette zone permet la saisie de mots-clés explicites, le système créant automatiquement des mots-clés implicites à partir du nom des entités.
		Les différents mots-clés doivent être séparés par au moins un blanc.
		La longueur maximum d'un mot-clé est de 13 caractères. Ceux-ci doivent être alphanumériques. Les caractères = et * sont ignorés.
		Majuscules et minuscules sont équivalentes, mais attention aux minuscules accentuées (e et E sont équivalents, mais pas é et E).
		NOTE : Les caractères accentués et les caractères spéciaux peuvent toutefois être déclarés équivalents à une valeur interne pour faciliter la recherche des instances par mots-clés (Administrator workbench, menu "Fenêtre", choix "Browser des Paramètres", onglet "Caractères spéciaux").
		Il est possible d'affecter dix mots-clés explicites au maximum par entité. Pour plus de détails, voir le chapitre "Recherche d'instances", dans la partie dédiée aux mots-clés dans le "Guide de l'Interface Mode Caractère".
10	6	CODE RUBRIQUE MERE
		Permet de définir des ensembles de Rubriques ayant des codes différents, mais des caractéristiques identiques.
		Si une Rubrique mère est indiquée, la Rubrique définie prend par défaut les caractéristiques de la mère, qui peuvent être modifiées au niveau de la Rubrique fille.
		La Rubrique mère doit avoir été préalablement définie.

Description d'une colonne (E.....D)

Une Colonne étant assimilée à une Rubrique, elle est décrite de la même façon qu'une Rubrique. L'écran de description d'une Rubrique est obtenu par le choix :

CH: E.....D

INDICATION DU NOM DE COLONNE

Par défaut, le nom de la Colonne est le code de 6 caractères de la Rubrique. Vous pouvez cependant indiquer un nom de 18 caractères dans l'écran de description (-D) de la Rubrique, sur une ligne de type 'R'.

Si aucun 'ALIAS' n'est indiqué au niveau de la Colonne lors de l'appel de la Rubrique dans le Segment décrivant la Table, c'est ce libellé qui sera utilisé.

S'il n'est pas indiqué non plus, le code de la Rubrique est alors utilisé.

DESCRIPTIONS PREREQUISES

La Rubrique doit avoir été préalablement définie.

NUM	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
LONG	VALEUR	REPLISSAGE
	blanc	Ligne de description complémentaire ; les lignes de ce type permettent de saisir un texte décrivant la rubrique ainsi que ses valeurs possibles avec leurs significations
	'D'	Valeur par défaut de la rubrique ; l'une des valeurs indiquées peut être repérée comme étant la valeur par défaut ; elle peut alors être attribuée comme valeur initiale dans les descriptifs de segments
		Types spéciaux (Modules Dialogue, Pacbench C/S et Pactables)
	'P'	Exemple de valeur de la rubrique ; la valeur est saisie dans la zone 'Description complémentaire' ; elle est utilisée lors de la simulation d'un Ecran
	'L'	Libellé court de la rubrique (18 caractères maximum) ; il est saisi dans la zone 'Description complémentaire' ; le délimiteur peut être paramétré (il est saisi dans le premier caractère de la zone 'valeur' ; si aucun délimiteur n'est saisi, le caractère 'E' est pris par défaut)
	'C'	Libellé en-tête de colonne ; il est saisi dans la zone 'Description complémentaire' de une à trois lors de son utilisation ; un délimiteur indique un changement de ligne ; au moins un délimiteur est obligatoire ; le délimiteur est saisi dans le premier caractère de la zone 'Valeur' ; si aucun délimiteur n'est saisi, le caractère / est pris par défaut ; la longueur effective du libellé lors de son utilisation est celle de la plus longue des lignes ; il doit être obligatoirement borné par au moins un délimiteur (18 caractères maximum y compris les délimiteurs)
	'F'	Définition du format conversationnel de la Rubrique ; si aucun format conversationnel n'est défini, le système prend par défaut le format interne étendu ou le format d'édition selon la nature de la Rubrique dans l'écran ; en cas de définition de format conversationnel pour une Rubrique de type date, celui-ci doit être précisé dans la zone VALEUR, le système affichant son décodage dans la zone décodage dans la zone 'Description complémentaire' ; dans les autres cas, on précise le format dans la zone DESCRIPTION COMPLEMENTAIRE ; ce format est saisi sous la forme d'un format d'édition ; le format interne correspondant est automatiquement calculé ; pour les Rubriques numériques, la clause 'Blank when zero' s'obtient en entrant un 'Z' après le format (exemple : 9(4) Z)

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			Traitements automatiques (modules Dialogue et Pacbench C/S (uniquement Clients TUI) (voir les manuels "Dialogue" et "Pacbench C/S - Services applicatifs et Clients TUI"). il est possible de déclarer les valeurs prises par le code opération et le code mouvement en fonction de la valeur d'une rubrique sur la description de la rubrique, sur des lignes de type O ou I ; la valeur de la rubrique est indiquée dans la zone 'Valeur de la rubrique, la valeur du code opération ou mouvement dans la zone 'Type d'action' ; il est également possible de déclarer ces valeurs directement sur la description de l'écran.
		'O'	Déclaration des valeurs de code opération
		'I'	Déclaration des valeurs de code mouvement
			Bases de données relationnelles
		'R'	Valeur permettant de faire générer le libellé relationnel de la Rubrique sur 18 caractères, indiqué dans la zone 'Description complémentaire' ; le libellé relationnel défini au niveau d'une rubrique mère n'est pas repris pour les rubriques filles lors de la génération ; dans le cas de TurboImage, cette zone permet de générer un nom d'item différent du code rubrique ; dans ce cas, seuls les 16 premiers caractères sont reconnus.
		'E'	Valeur permettant de saisir des formats date non standard dans la zone 'Description complémentaire' ; sur la définition de la rubrique, le format doit être X(n) avec n < 28 (ou n < 15 pour une base de données ORACLE pour la gestion automatique des dates dans Dialogue et Pacbench C/S) ; ce format est pris en compte dans la génération SQL (pour générer le type de données Date pour Oracle, Sybase et SQL Server, et Datetime pour Nonstop SQL), dans la génération Dialogue et Pacbench C/S (pour les accès SQL par exemple en générant les fonctions Tochar et Todate pour Oracle) ; aucun contrôle de date non standard n'est effectué dans les programmes générés ; seules les dates standard (types C,D,E,G,I,M,S) sont contrôlées ; l'opérateur de date (AD) ne peut pas non plus s'appliquer à ce format non standard ; un contrôle syntaxique est effectué à la saisie mais uniquement sur les éléments eux-mêmes du format et non sur leur assemblage (ex : MD sera rejeté mais MMMMMM et YY-DD/MM seront acceptés)
			Rubrique issue de Reverse engineering
		'S'	Qualifie une rubrique issue de Reverse engineering ; le(s) nom(s) COBOL de la donnée Reverse associée figurent dans la zone 'Description complémentaire'

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			Reprise de l'existant
		'A'	Rubrique de type ALIAS ; ce type est positionné de façon automatique lors d'un passage "Reprise de l'existant"
			Rubrique utilisée dans une Base de Données Adabas. Permet de saisir les valeurs associées pour alimenter la génération du Format-Buffer.
			Liste des classes Turboimage
		'T'	Valeurs de la liste des classes Turboimage
5	1		TYPE D'ACTION, SAUT, SUITE
			Cette rubrique offre plusieurs possibilités:
			Saut de ligne ou de page, suite de plage de valeur, type de code action.
			SAUT
			Le saut est pris en compte uniquement pour l'édition de documentation utilisateur.
		blanc ou 1	Nouvelle ligne.
		2	1 ligne blanche + 1 nouvelle ligne.
		3 à 9	de 2 à 8 lignes blanches + 1 nouvelle ligne.
		'*'	Edition des Manuels (entité 'U'): Saut de page.
			SUITE
		'+'	Repère la suite d'une plage de valeurs au cas où elle ne peut être indiquée sur une seule ligne.
			ACTION (module DIALOGUE et Client TUI de Pacbench C/S)
			Associé à la zone TYPE DE LIGNE.
			Trois catégories de valeurs suivant le type de ligne sélectionné :
			Valeurs associées au type de ligne 'T':
		'C'	Création.
		'M'	Modification.
		'A'	Suppression.
		'X'	Type de mouvement '4' (mise à jour implicite).
			Valeurs associées au type de ligne 'O':
		'A'	Affichage
		'P'	Même page

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		'M'	Mise à jour
		'E'	Abandon de la conversation
		'S'	Suite
		'O'	Appel d'un autre écran
6	10		VALEUR DE LA RUBRIQUE
			Permet de décrire les valeurs possibles du contenu de la Rubrique. Elles peuvent faire l'objet d'un contrôle de validité automatique, dans les programmes ou les écrans.
			. Elles doivent être décrites sous forme de littéraux numériques ou alphanumériques (entre quotes pour ces derniers);
			. Un intervalle se décrit par deux valeurs situées entre parenthèses et séparées par au moins un espace. Une parenthèse inversée (fermante à gauche ou ouvrante à droite) indique que la valeur ne fait pas partie de l'intervalle.
			EXEMPLES :
			('E' 'Z') pour un intervalle de E inclus à Z inclus
)0 100(pour un intervalle de 0 exclu à 100 exclu
			. Si la signification associée à une valeur comporte plusieurs lignes, la valeur ne doit figurer que sur la première,
			. Les valeurs attribuées à une Rubrique mère sont reprises pour chacune de ses Rubriques filles.
			MODULES DIALOGUE et PACBENCH C/S
			- Génération des contrôles de numéricité sur des Rubriques de format ALPHANUMERIQUE :
		*9	Contrôle de numéricité de type COBOL (IF NOT NUMERIC).
		*B	Contrôle de numéricité de type COBOL, après remplacement des blancs à gauche par des zéros.
		*Z	Contrôle de numéricité de type COBOL, après remplacement de tous les blancs par des zéros.
		*A	Contrôle de classe alphabétique.
		*L	Contrôle de classe alphabétique qui n'accepte que les minuscules.
		*U	Contrôle de classe alphabétique qui n'accepte que les majuscules.

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		Dans tous les cas, le Système affiche la désignation du contrôle dans la zone SIGNIFICATION.
		- Pour un libellé en-tête de colonne (zone TYPE DE LIGNE égale à C), un caractère quelconque saisi dans cette zone et cadré à gauche représente le délimiteur des lignes du libellé. Si aucun caractère n'a été saisi, la valeur par défaut du délimiteur est /.
		- Pour le libellé court de la Rubrique (zone TYPE DE LIGNE égale à L), un caractère quelconque saisi dans cette zone et cadré à gauche représente le délimiteur du libellé. Si aucun caractère n'a été saisi, la valeur par défaut du délimiteur est £.
		- Format conversationnel des Rubriques DATE (zone TYPE DE LIGNE égale à F) :
	'T'	Date sans siècle (picture x(6)) en AAMMJJ.
	'S'	Date avec siècle (picture x(8)) en SSAAMMJJ.
	'D'	Date sans siècle (picture x(6)) en MMJJAA ou JJMMAA selon le format date indiqué pour la Bibliothèque.
	'C'	Date avec siècle (picture x(8)) en MMJJSSAA ou JJMMSSAA selon le format date indiqué pour la Bibliothèque.
	'G'	Date avec siècle (picture x(10)) en SSAA-MM-JJ, format grégorien.
	'E'	Date sans siècle (picture x(8)) en MM/JJ/AA ou JJ/MM/AA.
	'M'	Date avec siècle (picture x(10)) en MM/JJ/SSAA ou JJ/MM/SSAA.
	'T'	TIME.
	'TS'	TIMESTAMP.
		Pour l'utilisation des formats avec les différents types de Blocs Bases de Données se reporter aux tableaux récapitulatifs du chapitre "Colonnes : Rubriques" du Manuel DBD RELATIONNELLES SQL.
		Pour le code Mouvement ou le code Opération (zone TYPE DE LIGNE égale I ou O) : Valeur de saisie possible, associée à la valeur interne indiquée dans TYPE D'ACTION.
7	54	DESCRIPTION COMPLEMENTAIRE
		Le contenu de cette zone dépend du TYPE DE LIGNE :
		Pour les valeurs BLANC, D, O ou I : Saisie d'un commentaire (facultatif).

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		Pour les valeurs L, C ou P : Saisie d'un libellé (avec ses bornes) ou d'une valeur de présentation.
		Pour la valeur A : Saisie du code du Segment dans lequel la Rubrique mère est appelée.
		Pour la valeur R : Saisie du nom de la colonne relationnelle.
		Pour la valeur E : Saisie d'un format date non standard décrit avec un ou plusieurs des éléments suivants :
		. YY : année (YYYY avec le siècle)
		. MM : mois
		. MON : trois premiers caractères du mois
		. DD : jour
		. HH : heure 00 à 23 sauf pour SQL Oracle :
		heure 00 à 12
		. HHAM ou HHPM : heure 00 à 12 + indicateur am/pm
		. HH24 : heure 00 à 23 pour SQL Oracle
		. MI : minute
		. SS : seconde
		. FF : fraction de seconde (millième de seconde)
		. les délimiteurs / . : - blanc
		Pour plus de détails, consultez la documentation de votre SGBD. Pour NONSTOP SQL : Saisie du champ de départ et du champ de fin.
		Pour la valeur F (pour les Rubriques qui ne sont pas des dates) : Saisie du format d'édition (utilisant la syntaxe COBOL standard). NOTE : pour générer une clause BLANK WHEN ZERO avec des Rubriques numériques, le format doit être suivi par un blanc puis un 'Z'. EXEMPLE : 9(4) Z
		Pour la valeur T : Saisie de la liste des classes TurboImage selon la terminologie TurboImage.

NUM	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
LONG	VALEUR	REPLISSAGE
	\$OFF \$ON	Lorsque la Description de la Rubrique est imprimée dans un Document avec l'option d'édition EO, la commande \$OFF cadrée à gauche permet d'exclure les lignes de Description suivantes. Pour mettre fin à cette exclusion, il suffit de créer une ligne comportant la commande \$ON cadrée à gauche, après la dernière ligne à exclure. ATTENTION : Cette exclusion n'est pas effective lorsque la Description fait partie de l'aide en ligne générée. Seules les lignes portant les commandes \$OFF et \$ON sont exclues. Pour plus de détails sur l'option EO et ces deux commandes, consultez le Manuel de "Gestionnaire de documentation personnalisée".
		DBD SOCRATE/CLIO
	'-NC-'	Caractéristique non codée.
		Nombre maximum de valeurs contenues dans une caractéristique de type liste de valeurs.
		Pour préciser explicitement les bornes d'une caractéristique de type numérique entier borné ainsi que l'écart maximum.
		EXEMPLE : BORNE = DE 01 A 99 (100)

Chapitre 4. Tables et Vues : Segments

Définition d'une Table / Vue (S....)

Une Table ou une Vue étant assimilable à un Segment, on les définit à l'aide d'une ligne de définition de Segment, obtenue par le choix :

CH: S....

C'est lors de son appel dans le Bloc Base de Données, que le Segment sera défini comme servant à générer une Table ou une Vue.

Il est possible d'indiquer sur cet écran le nom externe de la Table dans la zone CODE STRUCTURE. Ce nom sera utilisé si aucun nom externe n'est indiqué pour la Table dans l'écran de description du Bloc.

LIGNES EVENTUELLEMENT ASSOCIEES

Commentaires (choix CH: -GC).

Eléments de Génération (choix CH: -GG).

Options de Génération (choix CH: -GO).

Génération aide en Ligne (choix CH: -GE).

DESCRIPTION PREREQUISE

La Structure de Données dont dépend le Segment doit avoir été définie.

NOM D'UNE TABLE OU D'UNE VUE : PREFIXE

Le nom d'une Table ou d'une Vue est, par ordre de priorité décroissant :

- Le nom sur 27 positions précisé sur la ligne d'appel du Segment Table ou Vue dans le Bloc. Voir le sous-chapitre "Description d'un Bloc".
- Le nom sur 8 positions précisé dans l'écran de définition du Segment, dans la zone CODE STRUCTURE.
- Si aucun nom externe n'est précisé, le nom sera le code du Segment.

MODE DE PREFIXAGE (DB2, SQL/DS, DB2/2 et DB2/6000)

La présence d'un point dans le nom conditionnera le mode de préfixage :

- Si le nom est de la forme PREFIXE.NOM, il sera considéré tel quel par le Système et le SGBD, c'est-à-dire explicitement préfixé par l'utilisateur.
- Si le nom ne contient ni point ni préfixe, il sera généré par le Système avec un préfixe égal au code de l'utilisateur (8 positions). Il sera considéré tel quel par le SGBD, c'est-à-dire explicitement préfixé par le Système.
Pour DB2 seulement, il est possible de remplacer le code utilisateur VA Pac sur l'écran -GN en renseignant la zone PREFIXE.
- Si le nom est de la forme .NOM, il sera généré sans le point (NOM), de façon à ce que le préfixage soit assuré par le SGBD.

Dans ce dernier cas, le préfixe attribué dépendra du mode de mise à jour du catalogue :

- Dans l'environnement TSO, par l'application SQL interactive (SPUFI), le LOGON TSO est pris comme préfixe. Dans le cas de la mise à jour batch, le préfixe est un paramètre dans le JOB.
- Dans l'environnement CICS ou IMS, par l'application interactive (-GN), le préfixe dépend de la procédure de SIGNON utilisée et de l'emploi, ou pas, d'un outil de sécurité. Deux exemples :
 - sous CICS avec SIGNON CSSN, le préfixe est l'OPID associé au userid dans la DFHSNT.
 - sous IMS avec RACF et saisie du LOGON, le préfixe est le LOGON IMS.

Remarque : Vous pouvez modifier le préfixage en indiquant l'option PREFIX = NO sur l'écran "Options de génération" d'un bloc (B.....GO). Ainsi, un nom sans point ni préfixe sera généré tel quel, et un nom de la forme .NOM sera généré avec un préfixe correspondant au code utilisateur.

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.195
                                1 2
FICHE DU SEGMENT                DZ05

NOM DU SEGMENT.....: 3 FOURNITURE
NOMBRE DE POSTES.....: 4
OCCURRENCES DU SEGMENT....: 5

VALEUR DU CODE STRUCTURE...: 6
NOM DU CODE MOUVEMENT.....: 7
CONTROLE PRESENCE.....: CR:          MO:          AN:
                                M4:          M5:          M6:

MOTS CLES ASSOCIES.: 8

NO DE SESSION.....: 0067          BIBLIOTHEQUE : BMS          BLOCAGE :

O: C1 CH: S dz05                                ACTION:
-----

```

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			CODE COMPLET DU SEGMENT
1	2		CODE STRUCTURE DE DONNEES (OBLIGATOIRE)
			Code alphanumérique.
			Il s'agit d'un code logique propre à la Base et donc indépendant des noms utilisés dans les Blocs Bases de Données et dans les Programmes.
2	2		CODE ENREGISTREMENT (OBLIGATOIRE)
			Le premier caractère doit être un chiffre, le second peut être un chiffre ou une lettre mais, dans ce dernier cas, seulement si le premier caractère est différent de zéro.
		00	Désigne la partie commune aux enregistrements d'un fichier, obligatoirement en tête de ceux-ci (valeur par défaut).
			Les indicatifs, le code Structure et la clé des fichiers indexés doivent obligatoirement appartenir à la partie commune.
			Un fichier ne comporte pas obligatoirement de partie commune.

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		Si un fichier ne comporte qu'un seul type d'enregistrement (fichier mono-structure ayant une description unique), le code Enregistrement sera également '00'.
		Avec Pactables, cette valeur n'est pas autorisée.
	01-99	Désigne un enregistrement spécifique. Les Rubriques de la partie commune ne doivent pas être redécrites dans chaque partie spécifique. Par contre, la même Rubrique peut figurer dans la partie commune et dans un ou plusieurs enregistrements (sauf pour des fichiers utilisés comme tables).
3	36	NOM DU SEGMENT (OBL. EN CREATION)
		Ce nom doit être le plus explicite possible car il permet la constitution automatique de Mots Clés dans les conditions détaillées dans le "Guide de l'interface utilisateur mode caractère", chapitre "Recherche d'instances", sous-chapitre "Recherche par Mot-Clé".
4	4	NOMBRE DE POSTES
		ZONE NUMERIQUE PURE LANGAGE BATCH :
		C'est le nombre de postes réservés lorsque le Segment est stocké en mémoire (utilisation T ou X ou bien type de description 3 ou 4). Pour les fichiers en utilisation T ou X, la valeur par défaut au moment de la génération sera 100.
		Pactables : Cette valeur est documentaire.
		PACBENCH C/S :
		La valeur saisie dans cette zone indique la capacité de lecture ou mise à jour itérative d'un serveur appelant cette Vue Logique. Cette capacité s'exprime par un nombre maximum de répétitions. La Vue Logique peut alors être utilisée en tant que structure répétée.
		NOTE : L'utilisation d'une Vue Logique dans une maquette Fiche n'exclut pas son utilisation dans une maquette Liste. Il est donc fortement recommandé de systématiquement renseigner cette zone. De plus, la valeur saisie doit être suffisamment grande pour limiter les échanges entre client et serveur.
5	9	NOMBRE D'OCCURRENCES DU SEGMENT
		ZONE NUMERIQUE PURE
		Module LANGAGE BATCH : Nombre moyen estimé d'occurrences du Segment dans une base de données ou un fichier classique.

NUM	LON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			Module METHODE : Cette zone est utilisée pour le calcul d'activité sur le record ou le set utilisant le Segment (TP uniquement).
			Module DBD : Cette zone est utilisée pour indiquer le nombre d'applications d'une entité dans un Bloc SOCRATE/CLIO.
6	10		CODE STRUCTURE
			Pour définir un record CODASYL, cette zone n'est pas utilisée.
			Pour une Table ou une Vue relationnelle, il est possible d'y indiquer le nom externe entre quotes (').
7	36		CODE MOUVEMENT
			Pour définir un record CODASYL, une Table ou une Vue relationnelle, cette zone n'est pas utilisée.
8	55		MOTS CLES ASSOCIES
			Cette zone permet la saisie de mots-clés explicites, le système créant automatiquement des mots-clés implicites à partir du nom des entités.
			Les différents mots-clés doivent être séparés par au moins un blanc.
			La longueur maximum d'un mot-clé est de 13 caractères. Ceux-ci doivent être alphanumériques. Les caractères = et * sont ignorés.
			Majuscules et minuscules sont équivalentes, mais attention aux minuscules accentuées (e et E sont équivalents, mais pas é et E).
			NOTE : Les caractères accentués et les caractères spéciaux peuvent toutefois être déclarés équivalents à une valeur interne pour faciliter la recherche des instances par mots-clés (Administrator workbench, menu "Fenêtre", choix "Browser des Paramètres", onglet "Caractères spéciaux").
			Il est possible d'affecter dix mots-clés explicites au maximum par entité. Pour plus de détails, voir le chapitre "Recherche d'instances", dans la partie dédiée aux mots-clés dans le "Guide de l'Interface Mode Caractère".

Description d'une Table / Vue (S....CE)

Une Table ou une Vue étant assimilée à un Segment, elles sont décrites comme un Segment. L'écran de description d'un Segment est obtenu par le choix :

CH: S....CE

Cette description consiste à faire la liste des Colonnes composant la Table ou regroupées dans la Vue, c'est-à-dire à appeler les Rubriques dans le Segment.

Il est également possible, pour le DBD relationnel, d'appeler des Segments dans le Segment.

Note : Il est vivement recommandé de dédier un Segment à un seul type d'utilisation future.

LIGNES EVENTUELLEMENT ASSOCIEES

Commentaires associés à chaque ligne (choix CH: -CEnnnGC).

Éléments de génération associés à chaque ligne (choix CH: -CEnnnGG).

Libellés d'erreur, aide en ligne associés à chaque ligne (choix CH: -CEnnnGE).

où 'nnn' est le numéro de ligne.

DESCRIPTIONS PREREQUISES

La Table ou Vue et les Colonnes appelées doivent avoir été préalablement définies.

REMARQUES

Les Rubriques groupes ou redéfinies ne sont pas prises en compte lors de la génération.

Une Rubrique ne peut être utilisée qu'une fois par Table ou Vue, même si on lui attribue des noms de Colonne différents.

DEFINITION D'UNE VUE

Une Vue peut être définie de deux manières :

- par appel de Tables ou de Vues (Segment appelant des Segments),
- par appel de Colonnes par l'intermédiaire de l'écran '-DBE' décrit dans le sous-chapitre "Complément de Description de Vue (-DBE)".

Les lignes complémentaires de Vue '-DBE' sont matérialisées sur l'écran '-CE' : par une '*' dans le code Action et la mention -DBE.

PRIORITES DE GENERATION

Les noms de Colonnes peuvent être indiqués de quatre façons, qui sont prises en compte selon l'ordre de priorité suivant :

1. Déclarer le nom sur une ligne de l'écran "Eléments de génération" (-GG), associée à la ligne d'appel de la Table ou de la Vue dans le Bloc. Cette ligne doit avoir un numéro compris entre les numéros de ligne du repère de début et de fin d'insertion, après la ligne virtuelle appropriée (se référer au chapitre "Base de Données : Blocs" sous-chapitre "Eléments et options de génération".)
2. Déclarer le nom sur une ligne de description de la Vue ou de la Table à laquelle appartient la Colonne, en indiquant dans la zone MAJ/TABLE la valeur 'A*' suivie du nouveau nom (sur huit caractères au maximum).

EXEMPLE :

A NLG CORUB	MAJ/TABLE
100 COLNE1	A*COLONNE1

Ici, "COLONNE1" sera généré au lieu de 'COLNE1'.

1. Déclarer le nom dans la description de la Rubrique correspondant à la colonne, sur une ligne de type 'R'.
2. Déclarer le code de la Rubrique sur 6 caractères.

```

-----
                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.195
                  1 2
DESCRIPTION DU SEGMENT : DZ05 FOURNITURE
3 4   5       7       8 9   10 11 12 13 14       15       16
A NLG : CORUB FORM.INT. U OCC GR I CMS456 CRNS VALEUR/FCT MAJ/TABLE DOC BIBLI
100 : COCARA                                0                                0067
110 : NUCOD                                  P                                S 0      0067
120 : FOURNI                                V                                S 0      0067
200 : NUCLIE                                S 0                                0067
210 : DATE                                  0067
220 : RELEA                                0067
230 : REFCLI                                V                                S 0      0067
240 : RUE                                    V                                S 0      0067
250 : COPOS                                S 0                                0067
255 : VILLE                                V                                S 0      0067
260 : CORESP                                V                                S 0      0067
270 : REMISE                                0067
280 : MATE                                    V                                A*MATERIEL 0067
310 : PRIX1                                0067
330 : HEURE                                0067
340 : PRECIS                                S 0                                0067
:
: LIBELLE : 6
*** FIN ***
0: C1 CH: -CE
-----

```

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		CODE COMPLET DU SEGMENT
1	2	CODE STRUCTURE DE DONNEES (OBLIGATOIRE)
		Code alphanumérique.
		Il s'agit d'un code logique propre à la Base et donc indépendant des noms utilisés dans les Blocs Bases de Données et dans les Programmes.
2	2	CODE ENREGISTREMENT (OBLIGATOIRE)
		Le premier caractère doit être un chiffre, le second peut être un chiffre ou une lettre mais, dans ce dernier cas, seulement si le premier caractère est différent de zéro.
	00	Désigne la partie commune aux enregistrements d'un fichier, obligatoirement en tête de ceux-ci (valeur par défaut).
		Les indicatifs, le code Structure et la clé des fichiers indexés doivent obligatoirement appartenir à la partie commune.
		Un fichier ne comporte pas obligatoirement de partie commune.

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
			Si un fichier ne comporte qu'un seul type d'enregistrement (fichier mono-structure ayant une description unique), le code Enregistrement sera également '00'.
			Avec Pactables, cette valeur n'est pas autorisée.
		01-99	Désigne un enregistrement spécifique. Les Rubriques de la partie commune ne doivent pas être redécrites dans chaque partie spécifique. Par contre, la même Rubrique peut figurer dans la partie commune et dans un ou plusieurs enregistrements (sauf pour des fichiers utilisés comme tables).
3	1		CODE ACTION (OBLIGATOIRE)
		'C'	Création de la ligne
		'M'	Modification de la ligne
		'A' ou 'D'	Annulation de la ligne
		'T'	Transfert de la ligne
		'B'	Annulation multiple
		'G'	Transfert d'un groupe de lignes
		'?'	Demande de documentation
		'E' ou '-'	Inhibition de la mise à jour implicite sur la ligne
		'X'	Mise à jour implicite sans transformation majuscule/ minuscule
4	3		Numéro de ligne
			Numérique. Il est recommandé de commencer par la ligne 100 et de numéroter les lignes de 20 en 20 pour permettre des insertions ultérieures.
5	6		CODE RUBRIQUE
			RUBRIQUE ELEMENTAIRE DEFINIE DANS LE DICTIONNAIRE
			Elle prend obligatoirement les caractéristiques définies au niveau du DICTIONNAIRE. Si elle est utilisée comme groupe son format dépend de son contenu. Si le groupe sert d'indicatif son contenu doit être compatible avec le format du DICTIONNAIRE.
			RUBRIQUE NON DEFINIE DANS LE DICTIONNAIRE
			Les caractéristiques sont alors obligatoirement indiquées au niveau du Segment.
			CODES RESERVES OU INTERDITS

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		Le code Rubrique 'SUITE' est interdit, il est utilisé par le Système lors de la génération des Programmes.
	FILLER	Rubrique qu'on n'envisage pas d'utiliser (zones de cadrage).
		OPTIONS DU MODULE LANGAGE BATCH
	ENPR	Permet de stocker dans un fichier mouvement le vecteur des erreurs sur les Rubriques. Sa longueur doit être égale à $1 + n$, n étant le nombre de Rubriques élémentaires du fichier ou du Segment suivant l'option de vecteur erreur choisie.
	GRPR	Permet de stocker le vecteur des erreurs sur les enregistrements. Sa longueur doit être égale au nombre d'enregistrements plus un.
	ERUT	Permet de stocker le vecteur des erreurs utilisateur.
		Ces trois dernières Rubriques sont utilisées comme vecteurs erreurs dans les fichiers mouvements uniquement. Plus généralement elles peuvent être utilisées comme zones groupes facultatives, car elles apparaissent seulement sur option dans les programmes générés.
		APPEL D'AGREGATS DE DONNEES
		On peut indiquer un code Segment, un code Objet ou un code Relation (les Objets et les Relations sont décrits à l'aide du Module METHODE).
		La zone GROUPE a dans ce cas une codification particulière pour indiquer un appel. La structure appelée est insérée dans la description du Segment à l'endroit de l'appel, il n'est possible ni de la modifier ni de la compléter. Toutes les informations appartenant à la structure appelée sont prises en compte. L'appel peut être fait à l'intérieur d'un groupe éventuellement répété.
		A l'affichage, en O: C2, le code Action indique le niveau d'imbrication (affichage indiqué jusqu'à 4 niveaux d'imbrication).
		Le nombre de niveaux d'imbrication autorisé diffère suivant les générateurs. Dans les cas de génération de data et d'utilisation PAF, le nombre de niveaux d'imbrication est limité à 4.
		LIGNES SUITE

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			Il est possible de créer des lignes suite, notamment si plusieurs contrôles doivent être effectués sur une Rubrique. Dans ce cas, ne pas renseigner le code Rubrique. Il suffit que le numéro de ligne suive celui comportant le code Rubrique.
6	18		NOM DE LA RUBRIQUE
			Il est inutile de le renseigner pour une Rubrique définie dans le Dictionnaire car il prend obligatoirement la valeur indiquée dans celui-ci.
			Il est facultatif pour une Rubrique FILLER ou un appel d'agrégat de données.
			Remarques concernant la saisie en TP de Rubriques non déclarées dans le Dictionnaire :
			Cette zone ne peut pas être utilisée lors de la saisie pour plus d'une Rubrique à la fois. En effet, une seule zone est disponible sur cet Ecran, que ce soit à la saisie ou à l'affichage.
			En création ou en modification, il faut donc indiquer le numéro de ligne à deux endroits :
			. sur la ligne courante de saisie, servant à indiquer les diverses caractéristiques de la Rubrique, notamment son code et son format,
			. sur la ligne de saisie du nom (18 car. max.), située sur la partie inférieure de l'Ecran, pour relier cette information au reste de la Rubrique.
			Pour visualiser format et libellé, il faut utiliser l'option 'O: C2'.
			Si plusieurs Rubriques non déclarées dans le Dictionnaire ont été saisies, seul le nom de la première Rubrique apparaîtra à l'affichage, si le choix renseigné est 'S...CE'.
			Pour voir apparaître le nom de la Rubrique CORUB, numéro de ligne 130, par exemple, il faut utiliser le choix 'SsssCE130', ce qui a pour effet d'afficher les Rubriques du Segment '....' à partir de la Rubrique appelée sur la ligne 130.
7	10		FORMAT INTERNE DE LA RUBRIQUE
			Ce format est obligatoire seulement dans les cas suivants :
			- Rubrique élémentaire non définie dans le Dictionnaire, créée au niveau du Segment (format COBOL),

NUM	LON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			- Rubrique groupe clé ou appartenant à une clé ; sa longueur doit être égale à la somme des longueurs des Rubriques élémentaires qui la composent.
			- Rubrique FILLER.
			C'est le format interne, défini de la même façon que sur la fiche de définition d'une Rubrique. Les formats d'entrée et d'édition seront identiques, mais en usage DISPLAY.
8	1		USAGE INTERNE
			Il doit être renseigné pour une Rubrique non définie dans le Dictionnaire comportant un format. Valeur par défaut : 'D' pour DISPLAY.
			Pour les valeurs possibles, voir la zone USAGE INTERNE de la Définition d'une Rubrique.
9	3		NOMBRE DE REPETITIONS
			ZONE NUMERIQUE PURE
			Clause OCCURS, au niveau d'une Rubrique élémentaire ou d'un groupe (3 niveaux au maximum).
			Peut être transformée en clause OCCURS DEPENDING ON en mettant dans la zone complémentaire '**' suivis des codes enregistrement et rubrique du compteur.
			SOCRATE/CLIO
			Avec le Module DBD SOCRATE/CLIO, cette zone indique le nombre de lignes maximum d'une caractéristique de type texte.
			Remarque : les restrictions COBOL portant sur la clause OCCURS s'appliquent.
10	2		NOMBRE DE RUBRIQUES DANS UN GROUPE
			ZONE PSEUDO-NUMERIQUE
		'1-99'	Un groupe se définit par le nombre de Rubriques ELEMENTAIRES contenues (l'appel de segment correspond à une rubrique élémentaire).
			Les groupes peuvent contenir jusqu'à 99 Rubriques élémentaires.
			Les groupes à 9 niveaux sont autorisés. Ils ne peuvent pas se chevaucher.
			Il est possible d'utiliser cette zone pour appeler des entités METHODE ou des Segments déjà définis.

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
			Leurs codes seront indiqués dans la zone CODE RUBRIQUE (appel d'agrégats de données).
		'*M'	Appel d'Objet ou de Relation.
		'**'	Appel de Segments.
			SOCRATE/CLIO
			Cette zone sert à décrire une caractéristique de type Bloc en indiquant le nombre de caractéristiques qu'elle contient.
			DBD SQL
		'**'	Appel de Segment dans une Vue.
11	1		CLE D'ACCES OU INDICATIF DE TRI
			Cette zone n'est pas utilisée lors de la description d'un record CODASYL.
			Pour les Tables et Vues relationnelles :
		BLANC	Colonne de format fixe (valeur par défaut),
		'V'	Colonne de format variable.
		'W'	Pour DB2, SQL/DS, ORACLE, DB2/2 et DB2/6000, génération d'une VARCHAR quelle que soit sa longueur.
		'L'	Pour DB2, SQL/DS, ORACLE, DB2/2 et DB2/6000, génération d'une LONG VARCHAR.
		'C'	Pour Oracle V7, génération d'une CHAR.
12	6		INDICATEUR DE PRESENCE
			Seule la première position de la zone est utilisée pour qualifier la présence d'une Colonne dans une Table.
		Blanc	Colonne facultative (valeur par défaut).
			La Colonne est facultative en mise à jour et ne sera donc pas toujours renseignée en consultation. Dans un programme, pour tester sa présence, l'emploi d'un indicateur variable est obligatoire. On peut ajouter une telle colonne dans une Table par la commande ALTER.
		'O'	Colonne NOT NULL.
			La Colonne est obligatoire en mise à jour et sera toujours renseignée en consultation. Dans un programme, l'emploi d'un indicateur variable, pour tester sa présence, n'est donc pas nécessaire. On ne peut pas ajouter une colonne NOT NULL dans une Table par la commande ALTER.
			Cette valeur n'est pas prise en compte pour RDMS 1100.

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
		'P'	Colonne NOT NULL WITH DEFAULT (pour DB2, DB2/2, DB2/6000 et DATACOM/DB), ou DEFAULT (pour INTEREL RDBC et RFM), DEFAULT SYSTEM (pour NONSTOP SQL C30).
			La Colonne n'est pas obligatoire en mise à jour mais sera initialisée à zéro ou space par le SGBD, donc toujours renseignée en consultation. Dans un programme l'emploi d'un indicateur variable n'est pas nécessaire. On ne peut pas ajouter une Colonne (NOT NULL WITH) DEFAULT dans une Table par la commande ALTER.
			Cette valeur n'est pas prise en compte pour RDMS 1100 et pour NONSTOP SQL (Version C10).
13	4		SELECTION SCHEMA
		S	Indique la sélection du schéma.
			Les rubriques groupe et élémentaires doivent appartenir au même sous-schéma. Ce sous-schéma doit être indiqué sur toutes les lignes.
14	10		TOP DE PRESENCE
		O	Indique le top de présence dans un schéma.
15	10		MISE A JOUR / TABLE
			Le libellé relationnel d'une Colonne peut être saisi dans cette zone en y indiquant la valeur 'A*' cadrée à gauche, suivie du nom externe de la Colonne (sur huit caractères au maximum).
			Ce sera le nom pris en compte lors de la génération, sauf si un autre nom est précisé sur les lignes de Commentaires associées à la ligne d'appel de la Table/Vue dans le Bloc.
16	1		PRESENCE DE COMMENTAIRES
			Cette zone existe en TP uniquement. Elle n'est pas saisissable.
			Le caractère '*' indique qu'à cette ligne est affecté un commentaire, un élément de génération ou un message d'erreur.
			Accès à la ligne nnn : -CEnnn
			Accès au Commentaire de la ligne nnn : -CEnnnGC
			Accès à l'élément de génération de la ligne nnn : -CEnnnGG
			Accès au message d'erreur de la ligne nnn : -CEnnnGE

Complément de description d'une Vue (S....DBE)

Pour les Segments utilisés en tant que Vues, vous disposez d'un écran supplémentaire vous permettant d'indiquer l'origine (Table ou Vue) des Colonnes composant chaque Vue. Ceci n'est valable que si la zone TYPE DE CLE de l'écran -DR est à blanc. Cet écran s'obtient grâce au choix :

CH: -DBE

Les lignes provenant de l'écran de description du Segment (-CE) s'affichent. Elles sont repérées par un astérisque (*) dans le code action. Ces lignes ne peuvent pas être modifiées sur cet écran.

Les lignes de description complémentaire (-DBE) sont des lignes SUITE des lignes -CE. C'est-à-dire qu'une ligne -DBE se rapporte à la ligne de description de Segment qui la précède numériquement. Les lignes -DBE s'imbriquent donc numériquement dans les lignes de description du Segment (-CE).

EXEMPLE :

Soit l'écran suivant de description d'une Vue :

```
A NLG : CORUB
  080 : COLNE1
  100 : COLNE2
  120 : COLNE3
  140 : COLNE4
O: C1 CH: -CE
```

Pour créer une ligne -DBE se rapportant à la colonne COLNE2 (ligne 100), vous devez lui affecter un numéro compris entre 100 et 120.

Après cette création, l'écran -DBE se présentera ainsi :

```
A NLG : CORUB                                TAB/RUBR
* 080 : COLNE1
* 100 : COLNE2
  107 :                                       XX01CORUBA
* 120 : COLNE3
* 140 : COLNE4
O: C1 CH: -DBE
```

D'après l'exemple ci-dessus, la colonne COLNE2 provient de la colonne CORUBA de la table XX01.

Remarque : Dans la partie droite de l'écran -DBE, le code indiqué est le code du couple Segment-Rubrique, même si on lui a donné un nom de colonne différent par l'intermédiaire de A*, au niveau de la description du Segment.

Dans le cas d'un appel de Segment dans un Segment-Vue, le Système peut déduire certaines informations de manière automatique dans le but d'alléger la saisie, par exemple :

- Segment-Vue VV10 sans Rubrique mais appelant XX10
- Segment-Table XX10 avec CORUB

Le Système déduira que la colonne CORUB de la Vue VV10 a pour origine la colonne CORUB de la table XX10 (sans saisie sur l'écran -DBE).

UTILISATION DES SOUS-SCHEMAS

Il est possible de sélectionner les Rubriques participant à la Vue en indiquant :

- Dans l'écran '-DR', dans la zone TYPE DE CLE un numéro de sous-schéma de 0 à 9 ou '*'. La valeur '*' signifie que toutes les Rubriques du Segment (y compris les Rubriques de Segments appelées dans le Segment) appartiennent à la Vue.
- Dans le cas d'une sélection par un numéro de 0 à 9 dans la zone TYPE DE CLE : dans l'écran '-CE' du Segment, indiquer sur les rubriques devant être sélectionnées, dans la zone TYPE DE CONTROLE un 'S' (indication de sous-schéma) et un 'O' dans la zone VALEUR/FCT à la n^{ième} Colonne (n correspondant au numéro de sous-schéma indiqué sur l'écran -DR, O équivalant à 10).

Dans le cas de l'utilisation de sous-schémas les lignes -DBE éventuellement présentes sont ignorées.

UTILISATION DES APPELS DE SEGMENTS

- Si la Rubrique TYPE DE CLE est à blanc, c'est une Vue décrite par des lignes -DBE. Il est alors possible d'appeler un Segment qui sera considéré comme une Table et toutes les Rubriques de ce Segment seront sélectionnées sans obligation de rentrer des lignes -DBE.
ATTENTION : il ne peut y avoir qu'un niveau d'appel de Segment.
- Si la rubrique TYPE DE CLE est renseignée il s'agit d'une Vue décrite par sélection de sous-schémas ou de la totalité de la Table. En cas d'appel sur le premier niveau celui-ci est considéré comme une Table. Les Segments appelés dans ce premier Segment verront leurs Rubriques référencées à cette Table, quel que soit leur niveau d'imbrication.

VARIANTE CICS/VSAM		*PDMB.DDDD.BMS.241
1 2		
DESCRIPTION COMPL. VUE : DZ05 FOURNITURE		
3 4		5
A NLG :		TAB/RUBR. DOC BIBLI
* 100 : COCARA		0067
* 110 : NUCOD		0067
* 120 : FOURNI		0067
125 :		FF10FOURNI 0234
* 200 : NUCLIE		0067
205 :		CC20NUCLIE 0234
* 210 : DATE		0067
* 220 : RELEA		0067
* 230 : REFCLI		0067
235 :		CC20RAISOC 0234
* 240 : RUE		0067
245 :		CC30RUE 0234
* 250 : COPOS		0067
251 :		CC30COPOS 0234
* 255 : VILLE		0067
256 :		CC30VILLE 0234
* 260 : CORESP		0067
: LIBELLE :		
O: C1 CH: Sdz05 DBE		

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		CODE COMPLET DU SEGMENT
1	2	CODE STRUCTURE DE DONNEES (OBLIGATOIRE)
		Code alphanumérique.
		Il s'agit d'un code logique propre à la Base et donc indépendant des noms utilisés dans les Blocs Bases de Données et dans les Programmes.
2	2	CODE ENREGISTREMENT (OBLIGATOIRE)
		Le premier caractère doit être un chiffre, le second peut être un chiffre ou une lettre mais, dans ce dernier cas, seulement si le premier caractère est différent de zéro.
	00	Désigne la partie commune aux enregistrements d'un fichier, obligatoirement en tête de ceux-ci (valeur par défaut).
		Les indicatifs, le code Structure et la clé des fichiers indexés doivent obligatoirement appartenir à la partie commune.
		Un fichier ne comporte pas obligatoirement de partie commune.

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REPLISSAGE
			Si un fichier ne comporte qu'un seul type d'enregistrement (fichier mono-structure ayant une description unique), le code Enregistrement sera également '00'.
			Avec Pactables, cette valeur n'est pas autorisée.
		01-99	Désigne un enregistrement spécifique. Les Rubriques de la partie commune ne doivent pas être redécrites dans chaque partie spécifique. Par contre, la même Rubrique peut figurer dans la partie commune et dans un ou plusieurs enregistrements (sauf pour des fichiers utilisés comme tables).
3	1		CODE ACTION (OBLIGATOIRE)
		'C'	Création de la ligne
		'M'	Modification de la ligne
		'A' ou 'D'	Annulation de la ligne
		'T'	Transfert de la ligne
		'B'	Annulation multiple
		'G'	Transfert d'un groupe de lignes
		'?'	Demande de documentation
		'E' ou '-'	Inhibition de la mise à jour implicite sur la ligne
		'X'	Mise à jour implicite sans transformation majuscule/ minuscule
4	3		Numéro de ligne
			Numérique. Il est recommandé de commencer par la ligne 100 et de numéroter les lignes de 20 en 20 pour permettre des insertions ultérieures.
5	10		ORIGINE DE LA COLONNE
			Cette zone sert à indiquer de quelle Table ou Vue provient la Rubrique appelée, et son code dans le Segment d'origine, sous la forme ddsseeeee.
			En batch, utiliser la zone MAJ/TABLE.

Chapitre 5. Accès SQL

Aide à la prise en compte des accès SQL

Les Modules Pacbench C/S Dialogue et Batch vous offrent la possibilité d'écrire des accès SQL très simplement sans avoir à coder en spécifique l'intégralité des accès aux blocs base de données (ouverture, fermeture, prise en compte des mises à jours...). Le généré s'adapte automatiquement au type de bloc.

REGLES DE SAISIE

La saisie doit s'effectuer dans les traitements spécifiques (-P). Vous devez saisir un des opérateurs présentés ci-dessous dans la zone OPERATEUR et le code du Bloc dans la zone OPERANDE.

La syntaxe est la suivante pour un ordre :

```
CONNECT (ou équivalent) : SCC ccccc d
DISCONNECT (ou équivalent) : SDC ccccc d r
COMMIT : SCO ccccc d
ROLLBACK : SRO ccccc d
WHENEVER : SWH instruction
cccccc : code VA Pac du bloc (6 caractères)
d : valeur '2' si base distribuée (ex : Oracle, Sybase)
r : valeur 'R' pour choisir le DISCONNECT avec ROLLBACK.
```

Les indicateurs d et r peuvent être inversés.

Chaque ordre peut être complété sur des lignes suite (sans opérateur). Sur ces lignes suite, vous pouvez indiquer des options disponibles avec certains SGBD (option FORCE dans un ordre COMMIT pour Oracle par exemple).

Ex. de codification d'un CONNECT au bloc COBLOC Oracle:

```
OPE OPERANDE
SCC COBLOC 2
```

REGLES DE GENERATION

Si dans les Appels de Segments (-CS pour les modules Pacbench C/S et Dialogue ou -CD pour le module Batch), vous déclarez une organisation SQL (zone ORGANISATION) et donc un code Bloc (zone NOM EXTERNE), cette organisation est prioritaire par rapport au type de Bloc indiqué sur la Définition du Bloc.

Si le bloc figure dans les Appels de Segments comme étant distribué, les ordres liés à ce bloc seront générés 'distribués'.

Les ordres inconnus d'un SQL sont ignorés.

La génération du END-EXEC est automatique et pour le générateur batch, le END-EXEC est toujours suivi d'un point.

Personnalisation des accès SQL

Introduction

Attention : vous devez parfaitement maîtriser la syntaxe SQL pour personnaliser des accès SQL dans VA Pac.

INTRODUCTION

Vous pouvez :

- ajouter ou remplacer des clauses particulières dans l'accès généré en standard. Pour cela, vous pouvez modifier l'accès standard ou associer l'accès standard à un accès non-standard.
- créer un nouvel accès SQL en codant un accès non-standard,
- prendre en compte différents critères d'extraction correspondant à des méthodes d'extraction.

La prise en compte des accès standard est automatique ou quasiment automatique alors que celle des accès non-standard doit être demandée en spécifique.

C'est donc en fonction de la réutilisabilité de la personnalisation dans les Ecrans et Programmes que vous pouvez choisir entre personnaliser un accès standard ou non-standard.

VA Pac vous permet de spécifier des accès SQL personnalisés dans l'écran "Eléments de Génération" d'un Segment (CH: S...GG) et de les prendre en compte ou de les ignorer dans les entités qui utilisent ce Segment. En effet, vous pouvez les prendre en compte dans un Programme mais pas dans un autre, ou ne prendre que certains en compte... Vous avez donc une très grande latitude dans leur utilisation.

Un accès personnalisé peut être, par exemple, ne lire dans la base que les clients ayant passé des commandes d'un montant supérieur à 50 euros.

Description

Vous saisissez ces accès sur l'écran 'Eléments de Génération' (CH : S...GG).

Sur la première ligne, vous indiquez l'accès que vous voulez personnaliser et sur les lignes suivantes, vous indiquez la ou les partie(s) de l'accès que vous voulez modifier ou ajouter.

La capacité de stockage des accès SQL n'est plus limitée à 1000 lignes sur les -GG des Segments. La zone "NLG" est désormais alphanumérique. La saisie de lettres permet de créer des lignes supplémentaires.

PREMIERE LIGNE

Dans la zone TYPE DE LIGNE, saisissez la valeur 'G'.

En option C2, la zone DESCRIPTION contient deux zones tabulables :

- dans la première zone, saisissez 'SQL' pour indiquer que vous voulez personnaliser un accès SQL.
- dans la deuxième zone, indiquez le type d'accès à personnaliser. Le type d'accès est standard ou non standard.

Accès standard

Les types d'accès standard, qui seront automatiquement pris en compte dans les Ecrans et Programmes sont :

R : Select
RU : Select for Update
RA : Select (module Pacbench C/S)
W : Insert
RW : Update
D : Delete
P : Open Cursor
RN : Fetch
DC : Declare Cursor
CL : Close Cursor
UC : Update Cursor (module Pacbench C/S)

Remarque : l'accès standard DC (Declare Cursor) n'est pas géré pour les accès SQL où la génération du DC se fait en Working.

Exemple de personnalisation de l'accès standard en sélection :

T DESCRIPTION
G SQL R

Accès non-standard

Un accès non-standard doit être explicitement demandé dans l'Ecran ou le Programme. Il est codé sur un ou deux caractères de votre choix. Cet accès peut être :

- combiné à un accès standard si vous souhaitez ne modifier qu'une partie de l'accès standard et laisser au générateur le soin de gérer la syntaxe. Dans ce cas, vous devez d'abord saisir le code de l'accès non-standard puis, après un blanc, le code de l'accès standard.
- utilisé seul si vous souhaitez introduire un accès entièrement nouveau. Dans ce cas, l'accès doit être entièrement décrit et appelé en spécifique. La syntaxe (ponctuation, mot-clé...) doit être entièrement gérée.

Exemple de personnalisation d'un accès non-standard en sélection :

```
T DESCRIPTION
G SQL      R1 RA
```

Exemple d'introduction d'un accès non-standard pour compter le nombre d'enregistrements dans la table des trajets :

```
T DESCRIPTION
G SQL      CP
```

Méthodes d'extraction

Une méthode d'extraction est un accès en sélection sur un serveur. Elle est donc spécifique du Module Pacbench C/S.

Quatre ordres d'accès sont nécessaires : Declare Cursor, Open, Fetch, Close.

Vous pouvez donc :

- décrire partiellement un seul accès personnalisé, en saisissant deux fois le type d'accès 'EX'.
- ou décrire entièrement les quatre accès, en combinant vos accès personnalisés à chacun des quatre accès (DC, P, RN et CL).

Exemple de déclaration de la méthode d'extraction NOMMETHODE :

```
T DESCRIPTION
G SQL      EX EX NOMMETHODE
```

Exemple de personnalisation des quatre accès de la méthode d'extraction NOMMETHODE :

```
T DESCRIPTION
G SQL      D1 DC NOMMETHODE
G SQL      O1 P  NOMMETHODE
G SQL      F1 RN NOMMETHODE
G SQL      C1 CL NOMMETHODE
```

Vous devez saisir le nom de la méthode en majuscules, à partir de la colonne 7 sur 10 caractères maximum.

LIGNES SUIVANTES

L'accès SQL est décrit par un ensemble de clauses (ordre d'accès, where, order...). Chaque clause est décrite par un mot-clé et des données. A partir de la deuxième ligne, vous saisissez donc ces mots-clés, qui s'appliquent indifféremment à la personnalisation des accès standard et non-standard.

Dans la zone TYPE DE LIGNE, saisissez la valeur 'G'.

Dans la première partie de la zone DESCRIPTION, saisissez le mot-clé et dans la deuxième partie, saisissez les paramètres nécessaires à la mise en oeuvre de ce mot-clé.

Les données correspondantes remplaceront celles qui auraient été générées. Si elles doivent être ajoutées en complément, saisissez 'ADD' devant ou derrière chaque mot-clé.

Un mot-clé qui nécessite plusieurs lignes ne doit pas être répété sur chaque ligne.

MOT-CLE	PARAMETRES
ACCESS	Accès SQL
FROM	Liste des tables
ORDER	Liste de colonnes
COLUMN	Liste des colonnes et host-variables
WHERE	Critères de sélection
PARAM	Paramètres complémentaires

Voir les paragraphes suivants pour le détail de chacun de ces mots-clés.

Voir aussi la fin de ce sous-chapitre et du chapitre pour des exemples d'utilisation de ces mots-clés.

Le format généré des Rubriques date et des Rubriques variables utilisées en paramètres de ces mots-clés (ex : saisie de la colonne DATJOU (date du jour) comme paramètre de l'ordre ORDER) dépend du type de Bloc.

Règles générales de ponctuation

Le délimiteur correspondant est automatiquement généré à la fin de chaque ligne. Lorsqu'un nom de colonne et/ou de host-variable ne peuvent être saisis sur une ligne, vous indiquez le type 'V' pour cette ligne, et saisissez la suite sur la ligne suivante (reprise du type 'G').

Dans ce cas, et si vous indiquez un nom de colonne ET de host-variable (sous forme :- ou non), la host-variable doit être saisie sur une ligne différente du nom de la colonne.

Par contre, si vous codez une host-variable ET son indicateur sous la forme :- (sans code Segment), ils doivent être codés sur la même ligne.

Les paramètres associés au mot-clé COLUMN ont des contraintes supplémentaires concernant la ponctuation, détaillées dans le paragraphe "COLUMN" ci-dessous.

1. ACCESS

Permet de réécrire la clause d'accès (SELECT, UPDATE, DELETE...)

2. COLUMN

Ce mot-clé est suivi de la liste des colonnes.

Ces colonnes sont codées soit sous forme SQL-code Rubrique (dans ce cas elles sont remplacées à la génération par leur nom relationnel), soit directement sous leur nom relationnel.

Chaque colonne peut être suivie (séparée par au moins un blanc) du code de la host variable (et indicateur), codée soit sous la forme ':-code Rubrique' (à la génération, le code Segment Ecran défini dans le -CS sera inséré devant le :-code Rubrique), soit sous son nom de host variable.

Si la host variable n'est pas codée, pour les ordres où elle est nécessaire, elle sera générée avec le code Segment Ecran et le code Rubrique indiqué dans la colonne (sous forme SQL-code Rubrique, sinon il est obligatoire d'indiquer le code de la host variable).

Particularités de ponctuation :

- Si le nom de colonne et la host-variable ne peuvent tenir sur une ligne, vous pouvez utiliser une (ou plusieurs) ligne(s) de type 'V' mais uniquement pour le nom de la host-variable; le nom de colonne ne peut être saisi que sur une seule ligne.
- Pour l'ordre UPDATE ('RW') du mot clé COLUMN, l'ensemble formé de la host-variable et de son indicateur doit toujours être codé sur une seule ligne quelque soit le format de leur saisie.

3. FROM

Ce mot clé est suivi d'une liste de tables. A la génération, FROM est transformé en INTO ou reste à FROM selon le type d'accès.

4. WHERE

Permet de préciser le critère de sélection. Les noms colonnes peuvent être codés sous la forme 'SQL-code Rubrique' et les host variables et indicateurs sous la forme ':-code Rubrique'.

5. ORDER

Permet de préciser l'Order By.

Même principe que WHERE pour coder le nom des colonnes.

6. PARAM

Permet de coder des paramètres supplémentaires comme COUNT, HAVING... Le mot clé ADD n'est pas nécessaire.

Même principe que WHERE pour coder les noms de colonnes.

EXEMPLES :

Exemple de personnalisation d'un accès non-standard en sélection pour sélectionner un numéro de trajet supérieur ou égal au numéro demandé. Vous modifiez seulement la clause WHERE mais vous ne voulez pas modifier directement l'accès standard car cette sélection n'est valable que dans quelques Ecrans. Vous saisissez donc :

```
T DESCRIPTION
G SQL      RB RA
G WHERE    SQL-NOTJ >= :-NOTJ
```

Exemple d'ajout d'un accès non-standard pour compter le nombre d'enregistrements dans la table des trajets. Ici, il s'agit bien d'un accès non prévu en standard. Vous devez donc décrire un nouvel accès de la façon suivante :

```
T DESCRIPTION
G SQL      CP
G ACCESS   SELECT COUNT (*)
G          INTO :WWCA-COMPTEUR
G          FROM PTB0.TRAJET
```

La génération des mots clés ne se fait que pour les accès standards ou assimilés à un accès standard (CP R). Dans le cas de l'accès CP (non suivi par un type d'accès standard) il faut tout écrire sur la ligne 'G' y compris le mot clé.

Méthodes d'extraction

Si une méthode d'extraction est précisée dans la demande de service de sélection, la sélection de la liste d'occurrences doit s'effectuer selon un ou plusieurs critères.

Vous pouvez:

- décrire partiellement un seul accès personnalisé 'EX' avec uniquement les clauses nécessaires :

```
T DESCRIPTION
G SQL    EX EX NOMMETHODE
G WHERE  ...
G ORDER  ...
```

- ou décrire entièrement les quatre accès personnalisés, en précisant pour chacun les différentes clauses :

```
T DESCRIPTION
G SQL    D1 DC NOMMETHODE
description...
G SQL    O1 P  NOMMETHODE
description...
G SQL    F1 RN NOMMETHODE
description...
G SQL    C1 CL NOMMETHODE
description...
```

Voir aussi les informations complémentaires dans la Documentation du développeur, série Module eBusiness, "Applications Pacbench C/S - Services Applicatifs", chapitre "Composant Applicatif", sous-chapitre "Insertion de traitements spécifiques", "Insertion par rapport au niveau "Accès physiques"", partie "Accès SQL personnalisés".

OPTION

L'option UPPER=NO, introduite sur une ligne de type 'O' dans l'écran "Options de génération" (-GO), inhibe la conversion minuscules-majuscules des valeurs saisies entre deux délimiteurs de littéral (la valeur du littéral est définie au niveau de la Bibliothèque).

RESTRICTIONS

La syntaxe des ordres SQL pour les blocs RDMS (organisation 'R') n'est pas implémentée.

La règle de préfixage n'est pas appliquée. Le nom de la table est conservé, et le point situé en tête est supprimé, s'il existe.

ATTENTION: Dans les recouvrements programme-macro et macro-macro, les lignes générées à partir des écrans d'Éléments de Génération du Segment (S...GG) ne peuvent pas être éliminées.

Prise en compte

MODULES PACBENCH C/S ET DIALOGUE

PRESENTATION

Prérequis Module Dialogue : les accès SQL personnalisés sont pris en compte si l'option PACSQL est codée dans l'écran de compléments -O du Dialogue.

Tous les accès sont générés dans la fonction F80.

. Accès standard personnalisés

Les accès standard personnalisés sont automatiquement générés avec les compléments de personnalisation saisis dans l'écran S...GG du Segment.

En revanche, si vous souhaitez ignorer la personnalisation, vous devez le spécifier dans l'écran O.....P.

. Accès non-standard

Pour prendre en compte les accès non-standard, vous devez spécifiquement les appeler dans l'écran O.....P.

Remarque : Les accès standard sont ignorés pour les organisations R (SQL RDBMS) et 2 (DB2 sans accès généré) mais vous pouvez spécifier la prise en compte des accès non-standard.

SAISIE DANS LES -P

Vous demandez la prise en compte (ou la non-prise en compte) de la personnalisation des accès dans une sous-fonction de la fonction 80.

Vous devez saisir *C dans la zone "Type de condition" et le code Segment dans la zone "Condition d'exécution".

- Pour ignorer la personnalisation d'un accès standard, vous devez faire suivre le code Segment du code de l'accès standard.
- Pour prendre en compte un accès non standard, vous avez trois possibilités:
 - Si vous saisissez le code de l'accès standard avant le code de l'accès non-standard, la personnalisation de l'accès non-standard sera prise en compte et sera générée dans les traitements standard de la fonction 80.

- Si vous ne saisissez que le code de l'accès non-standard (sans indiquer d'accès standard), la personnalisation de l'accès non-standard sera prise en compte et générée dans une sous-fonction spécifique de la fonction 80. Elle constituera donc une fonction utilisateur inconnue du générateur et vous devrez donc la décrire et la gérer entièrement en spécifique.
- Si vous saisissez le code de l'accès non-standard, suivi du signe '=' et du code de l'accès standard, la personnalisation de l'accès non-standard sera prise en compte et complétée par les traitements générés automatiquement pour l'accès standard en fonction 80.

Exemples de saisie :

```
NVTY  CONDITION
10*C  SS00 R    l'accès standard personnalisé en sélection
                saisi dans S SS00 GG sera ignoré
10*C  SS00 R R1 l'accès non standard en sélection saisi
                dans S SS00 GG sera pris en compte et le
                traitement automatique de la lecture du
                Segment SS00 sera complété par les lignes
                correspondant à l'accès en sélection R1.
10*C  SS00 R1   l'accès non standard en sélection saisi
                dans S SS00 GG sera pris en compte mais les
                lignes correspondant à R1 seront générées
                sous la forme d'une fonction spécifique
                F80-SS00-R1.
10*C  SS00 R1=R l'accès non standard en sélection saisi
                dans S SS00 GG sera pris en compte et
                complété par les traitements générés
                automatiquement pour l'accès standard
                en sélection.
```

- Méthodes d'extraction

Si vous avez décrit un seul accès personnalisé 'EX', vous devez demander explicitement cet accès. Vous créez donc une sous-fonction de la fonction 80 et saisissez *C dans la zone "type de condition" et le code Segment dans la zone "Condition d'exécution". Vous devez ensuite saisir EX puis le nom de la méthode en majuscules.

Exemple d'appel d'une méthode d'extraction :

```
NVTY  CONDITION
10*C  SS00 EX NOMMETHODE
```

Si vous avez décrit quatre accès personnalisés différents (DC, P, RN, CL), vous devez appeler ces quatre accès en fonction 80 (même saisie que précédemment, sauf que 'EX' est remplacé par les accès personnalisés D1', 'P1', 'R1' et 'C1').

Exemple d'appel des quatre accès personnalisés :

```
NVTY  CONDITION
10*C  SS00 D1 NOMMETHODE
10*C  SS00 01 NOMMETHODE
10*C  SS00 F1 NOMMETHODE
10*C  SS00 C1 NOMMETHODE
```

Voir aussi le Manuel "Applications Pacbench C/S - Services Applicatifs", chapitre "Composant Applicatif".

MODULE BATCH

Les accès personnalisés ne peuvent être pris en compte par le générateur batch que si le -CD du Programme contient le code du bloc dans la zone NOM EXTERNE et l'organisation dans la zone ORGANISATION.

Vous devez appeler explicitement chaque accès personnalisé que vous voulez voir généré dans votre Programme. Dans l'écran de traitement spécifique (P.....P), saisissez 'SQL' dans la zone OPERATEUR, et dans la zone OPERANDE, saisissez :

- le code Segment dans le Programme
- le code Segment dans la Bibliothèque s'il est différent du code Segment dans le Programme,
- le type d'accès :
 - Si vous faites suivre l'accès standard de l'accès non-standard, la personnalisation de l'accès non-standard sera prise en compte et complètera l'accès standard.
 - Si en revanche vous n'indiquez que l'accès non-standard sans indiquer d'accès standard, la personnalisation de l'accès non-standard sera prise en compte. Elle constituera une fonction utilisateur inconnue du générateur et vous devrez donc la décrire et la gérer entièrement en spécifique.

La personnalisation d'un accès standard en sélection à partir du Segment SS00 (code Segment dans Programme : SGPG), sera donc spécifiée comme suit :

```
OPE  OPERANDE
SQL  SGPG SS00 R
```

La personnalisation d'un accès non-standard en sélection à partir du Segment SS00 (code Segment dans Programme : SGPG), sera donc spécifiée comme suit :

```
OPE  OPERANDE
SQL  SGPG SS00 R R1
```

Exemples

EXEMPLE D'ACCES PERSONNALISES

Dans l'exemple suivant, deux accès standard ont été personnalisés: l'accès en sélection et l'accès en mise à jour.


```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.241
ELEMENTS GENERATION DU SEGMENT      CN10

A NLG : T DESCRIPTION                                                    BIBLI
 100 : G SQL                    EX EX NOMCLIENT                        *DCC
 110 : G WHERE                  CUSTOMER_NUMBER >= :7-WW00-NUCLIE      *DCC
 120 : G                        AND NOMCLI >= :7-WW00-NOMCLI          *DCC
 125 : G                        ORDER BY CUSTOMER_NUMBER              *DCC
 200 : G SQL                    EX EX VILLE                            *DCC
 205 : G WHERE                  VILLE = 'PARIS'                        *DCC
 210 : G                        ORDER BY CUSTOMER_ NUMBER            *DCC
   :
   :
   :
   :
   :
   :
   :
   :
   :
   :
   :
0: C1 CH: SCN10 GG
-----

```

VARIANTE CICS/VSAM *PDMB.DDDD.BMS.241
TRAITEMENTS ECRAN 0 CLCLNT Serveur Client FONCTION: 80

A	SS	NLG	OPE	OPERANDE	NVTY	CONDITION
DD		N		PHYSICAL ACCESS FOR NOMCLIENT	10*C	CN10 EX NOMCLIENT
EE		N		PHYSICAL ACCESS FOR VILLE	10*C	CN10 EX VILLE

0: C1 CH: OCLCNT P

Chapitre 6. Bases de Données : Blocs

Définition d'une Base de Données (B.....)

Une base de données est définie par l'intermédiaire d'un Bloc Base de Données, obtenu par le choix :

CH: B.....

Le Bloc est défini par un code, un nom en clair et un type particulier :

Q2 pour les blocs DB2,

Q3 pour les blocs SQL SERVER,

QB pour les blocs DB2/2 et DB2/6000

QC pour les blocs DATACOM/DB,

QN pour les blocs NONSTOP SQL

QO pour les blocs ORACLE (versions antérieures à la V6),

QP pour les blocs ORACLE (à partir de la Version 6),

QR pour les blocs RDMS 1100,

QS pour les blocs SQL/DS,

QT pour les blocs INTEREL RDBC,

QU pour les blocs INTEREL RFM,

QY pour les blocs SYBASE.

Il est également possible de saisir un nom externe associé à la base de données.

Dans la mesure où l'écran de définition d'un Bloc est commun à tous les types de Blocs, la zone NOM EXTERNE DU SCHEMA, qui n'est pas utilisée, n'est pas réaffichée après la mise à jour du type de Bloc.

LIGNES EVENTUELLEMENT ASSOCIEES

"Eléments de génération" (-GG).

Les informations physiques nécessaires à la génération de la Base sont saisies sur les lignes "Eléments de génération" (-GG) associées au Bloc, de façon à compléter les informations logiques indiquées sur la Définition du Bloc.

"Options de génération" (-GO)

Sur cet écran, vous indiquez des options comme le mode de préfixage, la génération des COMMIT...

"Commentaires" (-GC)

Sur cet écran, vous saisissez des commentaires sur le Bloc ou sur les objets qu'il appelle.

VARIANTE CICS/VSAM *PDMB.DDDD.BMS.195

1

FICHE DU BLOC BASE DE DONNEES Q2BLOC

NOM.....:2 EXEMPLE BASE DB2

TYPE DE BLOC.....:3 Q2 D.B.2 SQL

VERSION.....:4

NOM EXTERNE.....: EXTQ2DB2 5

CARTES DE CONTROLES.... AVANT:6 APRES:7

MOTS CLES ASSOCIES.: 8

NO DE SESSION.....: 0067 BIBLIOTHEQUE : BMS BLOCAGE :

O: C1 CH: B q2bloc ACTION:

NUM	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
LONG	VALEUR	REMPLISSAGE
1	6	CODE BLOC BASE DE DONNEE (OBLIGATOIRE)
		1 à 6 caractères alphanumériques.
2	36	NOM DU BLOC (OBL. EN CREATION)
		Ce nom permet la constitution automatique de mots-clés et doit donc être aussi explicite que possible.
3	2	TYPE DU BLOC (OBL. EN CREATION)
		Pour les Bases de Données HIERARCHIQUES ou CODASYL, il n'est pas nécessaire, dès la création du Bloc, de préciser le type définitif. Le classement en réseau ou hiérarchique est suffisant à ce niveau. Ce n'est qu'au moment de la génération éventuelle du DDL qu'un type ayant une signification physique est requis.
	'SE'	Ensemble de sets (Bloc en réseau).
	'AR'	Arborescence (Bloc hiérarchique).
		BASE DE DONNEES HIERARCHIQUE - IMS/DL1
	'DP'	DBD physique.
	'DR'	DBD physique réduit (seulement les indicatifs).
	'DL'	DBD logique.

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
	'PC'	PCB.
	'IP'	Index primaire.
	'IS'	Index secondaire.
	'PS'	PSB (à affecter à la création et non modifiable par la suite).
		BASES DE DONNEES RELATIONNELLES
	'Q2'	DB2 SQL
	'Q3'	SQL SERVER
	'QB'	DB2/2, DB2/6000
	'QC'	DATAKOM/DB
	'QN'	NONSTOP SQL
	'QO'	ORACLE (versions antérieures à la V6)
	'QP'	ORACLE (à partir de la Version 6)
	'QR'	RDMS
	'QS'	SQL/DS
	'QT'	INTEREL RDBC
	'QU'	INTEREL RFM
	'QY'	SYBASE
	'DB'	DB2 (utiliser de préférence le type Q2).
		BASES DE DONNEES RESEAU (CODASYL)
		.DM4 (CII-HB H66 ou DPS8) :
	'M1'	Schéma (DDL) génération des champs élémentaires,
	'M4'	Schéma (DDL) génération des champs de premier niveau uniquement,
	'M2'	Schéma (DMCL),
	'M3'	Sous-schéma.
		.IDS2 (CII-HB H64 ou DPS7) :
	'I1'	Schéma (DDL),
	'I2'	Schéma (DMCL),
	'I3'	Sous-schéma (SDDL).
		.IDMS :
	'D0'	Schéma (DDL) (Release 10.0),
	'D1'	Schéma (DDL),
	'D2'	Schéma (DMCL),

NUM	LN	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		'D3'	Sous-schéma,
		'D4'	Sous-schéma (Release 5.7).
			.DMS (UNISYS 1100) :
		'S1'	Schéma (DDL),
		'S3'	Sous-schéma.
			BASE DE DONNEES SOCRATE/CLIO
		'SO'	Structure SOCRATE/CLIO
		'SS'	Sous-structure SOCRATE/CLIO
		'SA'	Ajout de structure ou de format
			DDL TANDEM
		'TD'	TANDEM.
			FICHER PHYSIQUE AS400
		'PF'	Fichier physique AS400 (IBM 38).
		'LF'	Logical File AS400 (IBM 38).
			Base de données DMSII
		'20'	Base DMSII (DASDL)
4	4		NUMERO DE VERSION
			Numéro de version du système de Base de Données
		3000	Nonstop SQL version C30
		5000	RDMS 1100 version 5RA4
		7000	ORACLE version V7
		blanc	Autres systèmes, toutes versions
5	8		NOM EXTERNE DU BLOC
			Nécessaire au moment de la génération.
			C'est le nom physique du module de langage de description de données généré par le Système.
			Un choix spécifique (LEB) permet d'obtenir la liste des Blocs triés par nom externe.
			Dans le cas de TurboImage, seuls les 6 premiers caractères sont pris en compte.
6	8		NOM EXTERNE DU SCHEMA
			Cette zone n'est utilisée que pour les Blocs de type SE (Ensemble de sets) et pour les Blocs CODASYL. Dans les autres cas elle n'est pas affichée.

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		Ce nom est nécessaire lors de la génération si le Bloc est un sous-schéma ou un DMCL.
		Il n'est pas nécessaire dans le cas d'un schéma.
		C'est le nom physique du schéma auquel se rattache le Bloc considéré.
7	1	OPTION LIGNES AVANT BLOC
		Nécessaire au moment de la génération.
		Code option du jeu de lignes de contrôle à insérer avant le Bloc généré.
8	1	OPTION LIGNES APRES BLOC
		Nécessaire au moment de la génération.
		Code Option du jeu de lignes de contrôle à insérer après le Bloc généré.
9	55	MOTS CLES ASSOCIES
		Cette zone permet la saisie de mots-clés explicites, le système créant automatiquement des mots-clés implicites à partir du nom des entités.
		Les différents mots-clés doivent être séparés par au moins un blanc.
		La longueur maximum d'un mot-clé est de 13 caractères. Ceux-ci doivent être alphanumériques. Les caractères = et * sont ignorés.
		Majuscules et minuscules sont équivalentes, mais attention aux minuscules accentuées (e et E sont équivalents, mais pas é et E).
		NOTE : Les caractères accentués et les caractères spéciaux peuvent toutefois être déclarés équivalents à une valeur interne pour faciliter la recherche des instances par mots-clés (Administrator workbench, menu "Fenêtre", choix "Browser des Paramètres", onglet "Caractères spéciaux").
		Il est possible d'affecter dix mots-clés explicites au maximum par entité. Pour plus de détails, voir le chapitre "Recherche d'instances", dans la partie dédiée aux mots-clés dans le "Guide de l'Interface Mode Caractère".

Description d'une Base de Données (B.....DR)

La description du Bloc est obtenue par le choix :

CH: B.....DR

Cette description permet de donner la liste des objets qui composent la base.

Sept types de lignes sont pris en compte :

- . Espace ----> P
- . Table ----> T
- . Vue ----> V
- . Index ----> I
- . Alter Table ----> A
- . Primary Key ----> K
- . Foreign Key ----> J

BLOCS / OBJETS	P	T	V	I	A	K	J
Q2	O	O	O	O	O	O	O
Q3	-	O	O	O	O	O	O
QC	O	O	O	O	O	O	O
QN	-	O	O	O	O	-	-
QO	O	O	O	O	O	-	-
QP	O	O	O	O	O	O	O
QR	O	O	O	O	O	-	-
QB	-	O	O	O	O	O	O
QS	O	O	O	O	O	O	O
QT	-	O	O	O	O	-	-
QU	-	O	O	O	Oa	-	-
QY	-	O	O	O	O	O	O
DB	O	O	O	O	O	O	O

Note :

a. ajout de colonne uniquement

Les objets appartiennent à l'espace qui les précède.

Un Alter Table doit être précédé de la ligne appelant la Table qu'il modifie.

Un Index doit être précédé de la ligne appelant la Table à laquelle il se rapporte.

Il n'est pas possible de créer un Index sur une Vue; l'ordre CREATE INDEX généré sera rejeté lors de la mise à jour du catalogue.

Pour RDMS, la Primary key n'est pas générée par une ligne I ; sa génération est incluse dans celle de la Table qui le précède (ligne T).

Il n'est pas possible de modifier le type d'une ligne INDEX, PRIMARY KEY ou FOREIGN KEY : il est nécessaire d'annuler la ligne puis de la recréer avec un autre type.

De même, le code Table indiqué sur une ligne INDEX, ALTER ou PRIMARY KEY ne peut être modifié.

Certains objets SQL sont exclusivement pris en compte dans Oracle V7, SYBASE et SQL SERVER :

- . Package ----> C (Paquet de fonction et de procedures;
Un 'B' dans le type de clef différen-
cie le Package BODY de sa définition)
- . Fonction ----> E
- . Procédure ----> Q
- . Trigger ----> R pour Oracle V7, Sybase et SQL Server,

ELEMENTS DE GENERATION

L'écran de Description du Bloc (-DR) contient les informations logiques nécessaires à la génération des instructions DDL. Le "Type de record SQL" provoque l'intégration des lignes virtuelles générées par le système. Ces lignes sont affichées et mises à jour sur l'écran "Eléments de Génération" (-GG) associé aux lignes de Description du Bloc.

Remarque concernant les descriptifs de bases de données :

Sur les lignes de type I, la présence d'un + indique que des clés ont été spécifiées pour cet appel (-K).

La présence d'un astérisque (*) en fin de ligne indique que des éléments de génération sont associés à cette ligne (-DRnnnGG).

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
4	1	Type de record SQL
	'P'	Espace (sauf pour Interel RDBC, Interel RFM, Nonstop SQL, Sybase et SQL Server)
	'T'	Table
	'V'	Vue
	'I'	Index
	'A'	Alter table : mise à jour des colonnes
	'K'	Primary Key.
		RDMS 1100 : prise en compte lors de la génération de la table qui la précède.
		DB2, Datacom/DB, SQL/DS, Oracle V6 V7, DB2/2, DB2/6000, Sybase et SQL Server : prise en compte lors de la génération par un ordre Alter table.
	'J'	Foreign key
		DB2, Datacom/DB, SQL/DS, Oracle V6 et V7, Sybase et SQL Server : prise en compte lors de la génération par un ordre Alter table
	'C'	Package (uniquement Oracle V7)
	'E'	Fonction (uniquement Oracle V7)
	'Q'	Procédure (Oracle V7, Sybase, SQL Server).
	'R'	Oracle V7, Sybase et SQL Server : Trigger
5	27	NOM EXTERNE DE L'OBJET
		C'est le nom manipulé par l'utilisateur.
		Il est obligatoire pour un Espace Table (type P).
		Il est interdit pour une Primary Key (type K, DB2, DB2/2, DB2/6000 ou DATACOM/DB).
		Pour tous les autres types, le nom externe peut être défini à plusieurs niveaux.
		A la génération, le nom externe est généré par priorité dans l'ordre suivant :
		- nom indiqué à ce niveau (-DR),
		- sinon on prend le nom indiqué dans le code structure du Segment définissant l'objet concerné,
		- sinon on prend le code du Segment définissant l'objet concerné.

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
		Pour une Foreign Key (type J) il faut deux codes séparés, le nom de la contrainte (8 caractères maxi.) et le code Segment de la Table référencée.
6	4	CODE TABLE, VUE
		Sur une ligne de TYPE T, V ou A : code du Segment représentant l'ensemble des colonnes de la Table ou de la Vue.
		Sur une ligne de TYPE I, K ou J : code du Segment supportant la clé.
		Sur une ligne de type P : cette zone ne doit pas être renseignée.
7	1	TYPE DE CLE
	U	Sur une ligne de type I : génère l'ordre UNIQUE.
	0-9	Sur une ligne de type V : Vue constituée de la sélection des Rubriques de ce sous-schéma dans le Segment. La valeur 0 correspond au sous-schéma 10.
	*	Toutes les Rubriques du Segment sont incluses dans la Vue.
	R C S	Sur une ligne de type J : Restrict (valeur par défaut pour les bases de type DB2 uniquement). Cascade (DB2, SQL/DS, DB2/2, DB2/6000 et ORACLE V7 uniquement). Set null (DB2, SQL/DS, DB2/2 et DB2/6000 uniquement)
	B BLANC	Sur une ligne de type C (ORACLE V7) : indique le type de package. Package BODY. Package normal.
	A B	Sur une ligne de type R : indique le déclenchement du trigger ou de la rule : Après (After). Avant (Before).
8	1	TYPE DE MOUVEMENT GENERE
		Cette zone permet de générer les ordres CREATE, ALTER et DROP.
	C	Création : Valeur attribuée par défaut à la création de la ligne; génère un ordre CREATE. Seule valeur possible pour un Alter Table (ligne de type A).
	M	Modification (sauf pour SQL/400 et INGRES SQL) : Ce choix est possible sur les lignes de type Table uniquement, et pour tous les types de Bases sauf SQL/400 et INGRES SQL. Le choix Modification génère un ordre ALTER.
	A	Annulation : Génère un ordre DROP. Pour un type de ligne J (Foreign Key) et K (Primary Key), génère un DROP PRIMARY KEY et un DROP FOREIGN KEY dans un ordre ALTER TABLE.

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
	'BLANC'	Pas de génération (-GN); pas de génération par la procédure GPRT en option C2.

Composition de la clé/d'un Alter Table (-DRnnnK)

La composition de la Clé (ou Index) de la Table, ou des colonnes d'un Alter Table est précisée sur l'écran obtenu par le choix :

CH: B bbbbbb DR nnn K

où nnn est le numéro d'une ligne de description du Bloc de type I, K, A ou J.

Cet écran peut également être obtenu en positionnant le curseur sur la ligne de description et en actionnant la touche fonction appropriée (en standard, PF9).

Composition d'un Index (I) :

Les Rubriques composant le Segment utilisé comme Index sont alors affichées : il est possible de les sélectionner en indiquant leur rang dans l'Index et l'ordre dans lequel elles sont triées (ascendant ou descendant).

Le rang est indiqué dans la zone RANG par une séquence de nombres de 1 à 16. La séquence doit commencer par 1 (qui détermine le critère majeur de tri) et ne doit pas comporter de trou.

Pour remettre un rang à blanc, il faut y saisir un caractère non numérique différent de blanc.

La zone ORDRE permet de préciser l'ordre du tri, qui peut être ascendant (A) ou descendant (D). Si aucune valeur n'est renseignée, aucun ordre ne sera précisé dans le DDL et l'ordre par défaut du SGBD sera utilisé.

Une fenêtre située à la droite de l'écran contient le récapitulatif des Rubriques composant la Clé, avec leur ordre de tri.

Colonnes d'un Alter Table (A) :

Les Rubriques du Segment appelé pour l'Alter Table sont affichées : il est possible de les sélectionner en indiquant un ordre de génération.

Le rang est indiqué dans la zone RANG par une séquence de nombres de 1 à 16. La séquence doit commencer par 1 et ne doit pas comporter de trou.

Pour remettre un rang à blanc, il faut y saisir un caractère non numérique différent de blanc.

La zone ORDRE permet de préciser le type de la modification :

blanc (ADD) : Ajout d'une colonne (valeur par défaut)

A ou D (DROP) : Annulation d'une colonne

M (MODIFY) : Modification d'une colonne

BASES DE DONNEES	blanc	M	A ou D
DB2 et DB2/6000	X		
ORACLE V7	X	X	
DATACOM	*	*	*
NONSTOP	X		
RDMS	*	*	*
SQL/DS	X		
INTEREL RDBC	X		X
INTEREL RFM	X		

'*' indique que les trois types de modification peuvent être utilisés simultanément.

'X' indique que les trois types de modification doivent être utilisés séparément.

NOTE : Composition d'une Primary Key ou Foreign Key (K, J) :

Les Rubriques du Segment appelé sont affichées : la sélection s'effectue par l'indication du numéro de rang en face des Rubriques retenues.

Le numéro de rang peut aller de 1 à 16, il ne doit pas comporter de trou. La zone ORDRE est ignorée.

REMARQUE

Si une Rubrique utilisée comme clé se trouve dans un appel de Segment, il faut d'abord l'annuler en tant que clé avant de l'annuler dans le Segment.

VARIANTE		CICS/VSAM		*PDMB.DDDD.BMS.195	
SAISIE DES RUBRIQUES 1	INDZ10	Q2BLOC	EXEMPLE BASE DB2		210
		DZ10			
		2	3		
		RANG	ORDRE		
COCARA				COLONNE	BIBLI
NUCOM				FOURNP	0067
FOURNP		02	A	NUCOM	0067
QTMLI		01		QTMLI	0067
QTMCO		03			0067
INFOR					0067
*** FIN ***					
0: C1 CH: -DR210K					

NUMLON	CLASSE VALEUR	SIGNIFICATION DES RUBRIQUES ET MODE DE REMPLISSAGE
1	6	CODE BLOC BASE DE DONNEE
		1 à 6 caractères alphanumériques.
2	2	RANG
	1-16	Cette zone permet de sélectionner les colonnes constituant la Primary Key, Foreign Key ou l'Index de la Table en indiquant leur rang; elle permet également d'indiquer les colonnes prises en compte pour la modification d'une Table en indiquant l'ordre de génération des ordres Alter.
		La sélection se fait par une séquence de nombres de 1 à 16; le critère majeur de tri est 1. La séquence ne doit pas comporter de trou.
		NOTE : pour remettre cette zone à blanc, il faut y saisir un caractère non numérique différent de blanc.
3	1	ORDRE DE TRI OU TYPE DE MODIFICATION
		Sélection des Colonnes de la Clé : -----
	'BLANC'	Valeur par défaut : pas de génération d'ordre de tri.
	'A'	Ascendant.
	'D'	Descendant.

NUM	CLASSE	SIGNIFICATION DES RUBRIQUES ET MODE DE
LN	VALEUR	REPLISSAGE
		Modification d'une Table : -----
	'BLANC'	Valeur par défaut : ajout de la Colonne.
	'A ou D'	Annulation de la Colonne.
	'M'	Modification de la Colonne.

Éléments et options de génération

L'écran de définition d'un Bloc permet au système de connaître les informations de niveau logique nécessaires à sa génération en langage source.

Les informations de niveau physique sont indiquées dans l'écran "Éléments de génération" du Bloc (B.....GG).

L'écran de description du Bloc permet au Système de connaître les informations de niveau logique nécessaires à la génération du DDL des Espaces, Tables, Vues et Index qui y sont décrits.

Les informations de niveau physique sont indiquées sur la ligne d'"Éléments de génération" affectée à chaque ligne de description du Bloc (B.....DRnnnGG).

La saisie des informations est facilitée par l'appel de lignes virtuelles, qui varient selon le type de ligne.

Les lignes d'éléments de génération à prendre en compte lors de la génération doivent être repérées par un type de ligne G.

Note : Les lignes virtuelles sont prises en compte uniquement pour la génération des ordres CREATE et des Alter Table générés par l'intermédiaire de l'écran -DR à partir de lignes de type A (et non des mouvements de type M).

GESTION DES OPTIONS

Pour indiquer des options, créez une ligne de type 'O' sur l'écran "Options de génération" du Bloc (B.....GO). Sur cette ligne, vous pouvez indiquer les options suivantes :

- Prise en compte de la ponctuation

Avec cette option (COMMA=YES), la ponctuation sera automatiquement générée en fin de ligne pour tout ajout ou modification des caractéristiques d'une colonne dans les "Éléments de Génération" (-GG) associés à la Table/Vue appelée dans le Bloc.

- Pas de prise en compte de la ponctuation

C'est l'option par défaut pour tous les types de Blocs SQL sauf DB2 SQL, SQL/DS et ORACLE V5.

Avec cette option (COMMA=NO), vous devez saisir vous-même la ponctuation quand vous ajoutez ou modifiez les caractéristiques d'une colonne dans les "Éléments de Génération" (-GG) associés à la Table/Vue appelée dans le Bloc.

- Pas de génération des Constantes

Indiquez cette option (COMMENT=NO) pour ne pas générer les Constantes VA Pac en commentaires lors de la génération du Bloc.

- Génération d'ordres COMMIT

Indiquez cette option (COMMIT=YES) pour qu'un ordre COMMIT soit généré après chaque annulation ("DROP") d'objet SQL. Dans un même flot de commandes, vous pourrez donc annuler puis recréer le même objet.

- Modification du préfixage standard

Cette option (PREFIX=NO) vous permet de ne pas utiliser le préfixage prévu en standard sur le nom externe des Tables et Vues.

Pour des informations sur le préfixage standard, reportez-vous au chapitre 'Tables et Vues : Segments', sous-chapitre 'Définition d'une Table/Vue'.

Si vous indiquez cette option, un nom externe sans point ni préfixe sera généré tel quel, et un nom de la forme '.NOM' sera généré avec un préfixe correspondant au code de l'utilisateur.

- Positionnement du code retour à '06'

Cette option (RETCOD=06) permet de renvoyer un code retour '06' au lieu de '08' quand des erreurs sont détectées lors de la génération du Bloc. Ainsi, la chaîne de génération ne s'arrête pas et passe à l'étape suivante.

- (Non) Génération des champs date, time et timestamp

Cette option (TIME=YES ou TIME=NO) permet de (ne pas) prendre en compte les champs Date (format 'M'), Time (format 'T') et Timestamp (format 'TS') dans les Blocs Tandem NonStop dont le numéro de version est à blanc ("Autres Systèmes, toutes versions").

- Pas de transformation des minuscules en majuscules

Indiquez cette option (UPPER=NO) pour empêcher la conversion automatique des minuscules en majuscules lors de la génération du Bloc.

- Description complète des Tables DB2

Cette option (DESCR=ALL) vous permet de générer la description complète des Segments qui composent les Tables DB2.

Toutes les Rubriques appelées dans les Segments (c'est à dire les Rubriques groupes, les redefines et les occurs) seront générées dans la Working Storage Section. Cette option est aussi obligatoire pour décomposer les dates générées en zones élémentaires pour les Segments DB2 (l'option BREAKDATE=YES doit être positionnée dans les "Options de Génération" (-GO) de la Bibliothèque, du Dialogue, du Programme, ou de l'Application eBusiness).

UTILISATION DES LIGNES VIRTUELLES

Les lignes virtuelles sont appelées automatiquement dans les écrans d'Eléments de génération, afin de faciliter la saisie des informations physiques.

Ces lignes sont repérées par un astérisque dans la zone CODE ACTION. Il est possible de les modifier.

Il existe deux types de lignes virtuelles :

- Lignes *VIRT :

Ces lignes sont repérées par la mention *VIRT dans la zone BIBLI. Elles sont prises en compte à la génération.

- Lignes *GUID :

Les lignes repérées par la mention *GUID dans la zone BIBLI. doivent être créées pour être prises en compte lors de la génération : lorsque la ligne doit être complétée, il suffit d'effacer l'astérisque du code action et de renseigner la ligne; si aucune information n'est à ajouter, il faut utiliser un code action explicite (C ou X).

Vous pouvez créer ou modifier les lignes donnant lieu à la génération.

Il est nécessaire d'effacer l'astérisque du code action pour que ces actions soient prises en compte.

Si le numéro de ligne est modifié, la mise à jour ne sera pas effectuée sur la ligne virtuelle elle-même mais sera, soit une création, soit une modification de la ligne ayant le numéro saisi.

Insertion de lignes :

```
A NLG : T DESCRIPTION
```

```
* 100 : G CREATE TABLE (NOM) IN NOM-DATABASE
      170 : G Ces deux lignes vont venir s'insérer dans la des-
      180 : G cription générée
* 200 : G          --> DEBUT D'INSERTION DES COLONNES <--
* 300 : G          --> FIN D'INSERTION DES COLONNES <--
```

Un point-virgule est généré avant la ligne insérée si elle débute en colonne 1.

MODIFICATION DES CARACTERISTIQUES D'UNE COLONNE

Plages d'insertion :

Vous devez utiliser des numéros de ligne compris entre ceux des repères de début et de fin d'insertion.

Pour modifier une colonne, saisissez <CORUB > sur la première ligne, puis la nouvelle description sur les lignes suivantes (en commençant en colonne 2).

Pour compléter les informations générées, saisissez, après <CORUB >, un espace et le mot clé 'ADD'. A la suite, saisissez les informations désirées. Vous devez saisir la ponctuation de fin sur la ligne de déclaration de la colonne car celle-ci ne sera plus générée.

Si vous ne voulez pas qu'apparaisse cette colonne, ne créez aucune ligne derrière la première ligne.

Six caractères sont obligatoires entre les deux délimiteurs.

Exemple : Modification du format

```
A NLG : T DESCRIPTION
* 100 : G CREATE TABLE (NOM) IN NOM-DATABASE
* 200 : G      ---> DEBUT D'INSERTION DES COLONNES <---
  210 : G <LIBFO >
  215 : G      LIBFO  FLOAT
* 300 : G      ---> FIN D'INSERTION DES COLONNES <---
* 400 : G      IN DATABASE-TABLESPACE
```

Notes : Les colonnes ne peuvent être supprimées de cette façon.

Vous devez gérer la virgule de séparation des colonnes ainsi que la parenthèse ouvrante (fermante) si la première (dernière) colonne est modifiée.

Pour DB2, la virgule de séparation est générée à la fin de chaque ligne. Pour ne pas la générer, il faut coder un '&' à la fin du dernier caractère non blanc de chaque ligne.

SOURCE D'UNE COLONNE DANS UNE VUE

Plages d'insertion :

Vous devez utiliser des numéros de ligne compris entre ceux des repères de début et de fin d'insertion.

Pour modifier la colonne source d'une colonne dans une vue, créez une ligne de type G de la forme <FFNN-CORUB > où FFNN est le code segment de la source et CORUB le nouveau code Rubrique.

EXEMPLE :

```
A NLG : T DESCRIPTION
* 100 : G CREATE VIEW (NOM DE LA VUE)
* 200 : G ---> DEBUT D'INSERTION DES COLONNES <---
  210 : G <LIBFO >
  215 : G DO10LIBFO FLOAT
* 400 : G ---> FIN D'INSERTION DES COLONNES <---
* 450 : G AS SELECT ALL
* 500 : G ---> DEBUT D'INSERTION DES COLONNES SOURCES <---
  510 : G <FFNN-CORUB >
  520 : G 'X'
* 600 : G ---> FIN D'INSERTION DES COLONNES SOURCES <---
* 700 : G FROM (NOM DES TABLES SOURCES)
```

Vous devez alors gérer la présence du code de la table source (après la clause FROM).

INSERTION DE LIBELLES ET DE COMMENTAIRES

Lors de la génération, les libellés et les commentaires des Tables et des Colonnes sont, par défaut, les codes et les noms indiqués sur les Définitions. Cependant, vous pouvez les remplacer par des libellés et commentaires. que vous saisissez sur les lignes de commentaires de la Table ou de la Vue.

. Insertion de libellés

Vous ne pouvez saisir des libellés que dans les blocs DB2, et SQL/DS sur les -GC des lignes T et V.

Les libellés sont limités à 30 caractères. Vous devez laisser le type de la ligne à blanc, puis saisir en colonne 1 la commande '+LAB', suivie du libellé dans le cas d'une Table ou Vue, ou suivie de <CORUB > et du libellé dans le cas d'une Colonne.

Si vous avez modifié plusieurs fois le même objet sur le -GC, seule la dernière ligne de modification sera prise en compte.

. Insertion de commentaires

Vous ne pouvez saisir des commentaires que dans les blocs DB2, SQL/DS, ORACLE, et NONSTOP/SQL, sur les lignes -GC des lignes T, V et I (uniquement pour NONSTOP/SQL).

Vous devez laisser le type de la ligne à blanc et saisir en colonne 1 la commande '+COM', suivie du commentaire dans le cas d'une Table, Vue ou Index, ou suivie de <CORUB > et du commentaire dans le cas d'une Colonne.

Vous pouvez entrer le commentaire sur plusieurs lignes. A la génération, une longueur maximum de 254 caractères, variable selon les SQL utilisés, sera prise en compte.

EXEMPLE :

```
A NLG : T COMMENTAIRE
  900 : +LAB NOUVEAU LIBELLE DE LA TABLE
  910 : +LAB <CORUB > NOUVEAU LIBELLE DE LA COLONNE
  920 : +COM NOUVEAU COMMENTAIRE DE LA TABLE
  930 : +COM <CORUB > NOUVEAU COMMENTAIRE DE LA COLONNE
```

Procédures SQL

PROCEDURE SQL: NOUVEAUX CHAINAGES

En décrivant les objets Package (C), Function (E), Procedure (Q) et Trigger ou Rule (R) dans les -GG associés à la ligne objet, vous pouvez obtenir des chaînages automatiques en se servant du tiret ('-'). Ce caractère est assimilé à un signe logique et est interdit dans les noms.

Vous pouvez avoir besoin:

- . de données locales, préfixées par PLQ-
- . de données colonne, préfixées par SQL-

Un chaînage n'est réalisable que sur une rubrique de 6 caractères. Des noms plus longs peuvent cependant être utilisés après un '_'. Le chaînage porte alors sur la première partie.

Exemple: PLQ-RUBRI_COMPLEMENT

Chapitre 7. Exemples d'éléments de génération

DB2

ELEMENTS DE GENERATION : DB2

Ce sous-chapitre présente les lignes d'éléments de génération d'un Bloc DB2.


```

-----
                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.195
ELEMENTS GENERATION DE BLOC B.D.    Q2BLOC                                210

A NLG : T DESCRIPTION                                                    BIBLI
* 150 : G CREATE (UNIQUE) INDEX (NOM)                                    *VIRT
* 200 : G ON (NOM DE LA TABLE)                                          *VIRT
* 300 :      ---> DEBUT D'INSERTION POUR LES COLONNES <---              *VIRT
* 400 :      ---> FIN D'INSERTION POUR LES COLONNES <---              *VIRT
* 500 : G USING STOGROUP _____                                       *GUID
* 550 : G PRIQTY      ---                                               *GUID
* 600 : G SECQTY      ---                                               *GUID
* 650 : G ERASE      NO                                               *GUID
* 700 : G SUBPAGES    4                                               *GUID
* 750 : G BUFFERPOOL _____                                       *GUID
* 800 : G CLOSE      YES                                              *GUID
* 900 : G DSETPASS   _____                                       *GUID
:
:
:
:
:
:
:
O: C1 CH: -DR210 GG
-----

```



```

-----
                                VARIANTE CICS/VSAM                                *PDMB.D801.BMS.258
ELEMENTS GENERATION DE BLOC B.D.    Q3BLOC                                110

A NLG : T DESCRIPTION                                                    BIBLI
* 150 : G ALTER TABLE (NOM)                                           *VIRT
* 180 : G ADD                                                            *VIRT
* 190 : G CONSTRAINT (NOM DE LA CONTRAINTE)                            *VIRT
* 195 : G PRIMARY KEY                                                    *VIRT
* 195 : G NONCLUSTERED                                                  *GUID
* 200 : ----> DEBUT D'INSERTION POUR LES COLONNES <----                *VIRT
* 400 : ----> FIN D'INSERTION POUR LES COLONNES <----                  *VIRT
* 500 : G WITH FILLFACTOR = _                                           *GUID
* 500 : G ON _____                                                 *GUID
:
:
:
:
:
:
:
:
:
:
*** FIN ***
O: C1 CH: -DR110GG
-----

```

DATAKOM/DB

ELEMENTS DE GENERATION : DATAKOM/DB

Ce sous-chapitre présente les lignes d'éléments de génération d'un Bloc DATAKOM/DB.


```

-----
                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.203
ELEMENTS GENERATION DE BLOC B.D.      QNBLOC                                130

A NLG : T DESCRIPTION                                                    BIBLI
* 150 : G CREATE VIEW (NOM DE LA VUE)                                    *VIRT
* 200 :      ---> DEBUT D'INSERTION POUR LES COLONNES <---            *VIRT
* 400 :      ---> FIN D'INSERTION POUR LES COLONNES <---              *VIRT
* 450 : G AS SELECT ALL                                                  *VIRT
* 500 :      ---> DEBUT D'INSERTION DES COLONNES SOURCES <---        *VIRT
* 700 :      ---> FIN D'INSERTION DES COLONNES SOURCES <---          *VIRT
* 800 : G FROM (NOM DES TABLES SOURCES)                                *VIRT
* 850 : G WHERE                                                           *GUID
* 900 : G FOR PROTECTION                                                  *GUID
* 950 : G CATALOG (CATALOG-NAME)                                         *GUID
* 960 : G SECURE                                                          *GUID
* 970 : G WITH CHECK OPTION                                              *GUID
      :
      :
      :
      :
      :
      :
0: C1 CH: -DR130GG
-----

```


A NLG : T	DESCRIPTION	BIBLI
* 150 : G	CREATE TABLE (NOM)	*VIRT
* 200 :	----> DEBUT D'INSERTION POUR LES COLONNES <---	*VIRT
* 400 :	----> FIN D'INSERTION POUR LES COLONNES<---	*VIRT
* 500 : G	TABLESPACE (NOM DU TABLESPACE)	*VIRT
* 600 : G	PCTFREE _____ PCTUSED _____	*GUID
* 650 : G	INITRANS _____ MAXTRANS _____	*GUID
* 700 : G	STORAGE (INITIAL _____ NEXT _____	*GUID
* 710 : G	MINEXTENTS _____ MAXEXTENTS _____	*GUID
* 720 : G	PCTINCREASE _____)	*GUID
* 800 : G	CLUSTER _____	*GUID
	:	
	:	
	:	
	:	
	:	
	:	
	:	
	:	

O: C1 CH: -DR200GG

	DESCRIPTION	BIBLI
A NLG :	T DESCRIPTION	BIBLI
* 150 :	G CREATE TABLE (NOM)	*VIRT
* 200 :	---> DEBUT D'INSERTION POUR LES COLONNES <---	*VIRT
* 400 :	---> FIN D'INSERTION POUR LES COLONNES<---	*VIRT
* 500 :	G TABLESPACE (NOM DU TABLESPACE)	*VIRT
* 600 :	G PCTFREE PCTUSED	*GUID
* 650 :	G INITRANS MAXTRANS	*GUID
* 700 :	G STORAGE (INITIAL NEXT	*GUID
* 710 :	G MINEXTENTS MAXEXTENTS	*GUID
* 720 :	G PCTINCREASE)	*GUID
* 800 :	G CLUSTER	*GUID
* 810 :	---> NOUVEAUX PARAMETRES (V7) <---	*VIRT
* 820 :	G DISABLE CONSTRAINT	*GUID
:		
:		
:		
:		
:		
:		

O: C1 CH: -DR200GG

A NLG : T DESCRIPTION	BIBLI
* 150 : G CREATE VIEW (NOM DE LA VUE)	*VIRT
* 200 : ----> DEBUT D'INSERTION POUR LES COLONNES <---	*VIRT
* 400 : ----> FIN D'INSERTION POUR LES COLONNES <---	*VIRT
* 450 : G AS SELECT	*VIRT
* 500 : ----> DEBUT D'INSERTION DES COLONNES SOURCES <---	*VIRT
* 700 : ----> FIN D'INSERTION DES COLONNES SOURCES <---	*VIRT
* 800 : G FROM (NOM DES TABLES SOURCES)	*VIRT
* 850 : G WHERE	*GUID

```

A NLG : T DESCRIPTION BIBLI
* 150 : G CREATE (UNIQUE) INDEX (NOM) *VIRT
* 200 : G ON (NOM DE LA TABLE) *VIRT
* 300 : ---> DEBUT D'INSERTION POUR LES COLONNES <--- *VIRT
* 400 : ---> FIN D'INSERTION POUR LES COLONNES <--- *VIRT
* 500 : G CLUSTER _____ *GUID
* 550 : G INITRANS _____ MAXTRANS _____ *GUID
* 600 : G TABLESPACE (NOM DU TABLESPACE) *GUID
* 700 : G STORAGE ( INITIAL _____ NEXT _____ *GUID
* 710 : G MINEXTENTS _____ MAXEXTENTS _____ *GUID
* 720 : G PCTINCREASE _____ ) *GUID
* 800 : G PCTFREE _____ *GUID
* 850 : G NOSORT *GUID
:
:
:
:
:
:

```

0: C1 CH: -DR210GG


```

CENTRALE POUR DOCUMENTATION *PDLB.NDOC.DCC.196
ELEMENTS GENERATION DE BLOC B.D. V7BLOC 630
A NLG : T DESCRIPTION BIBLI
* 010 : G CREATE OR REPLACE PROCEDURE (NOM DE LA PROCEDURE) *VIRT
* 020 : ---> INSERTION DES ARGUMENTS <--- *VIRT
* 100 : G AS *GUID
* 150 : G BEGIN *GUID
* 990 : G END *GUID
:
:
:
:
:
:
:
:
:
:
:
:
:
:
:
:
O: C1 CH: -DR630G

```

```

-----
CENTRALE POUR DOCUMENTATION                                *PDLB.NDOC.DCC.196
ELEMENTS GENERATION DE BLOC B.D.    V7BLOC                    650

A NLG : T DESCRIPTION                                        BIBLI
* 010 : G CREATE TRIGGER (NOM DU TRIGGER)                  *VIRT
* 030 : G DELETE OR INSERT OR UPDATE                       *GUID
* 040 : G ON (NOM DE LA TABLE)                           *VIRT
* 050 : G REFERENCING OLD AS _____ NEW AS _____   *GUID
* 060 : G FOR EACH ROW                                     *GUID
* 070 : G WHEN                                             *GUID
* 100 : G DECLARE                                          *GUID
* 150 : G BEGIN                                            *GUID
* 990 : G END                                              *GUID
:
:
:
:
:
:
:
:
:
:
*** FIN ***
O: C1 CH: -DR650GG
-----

```

RDMS

ELEMENTS DE GENERATION : RDMS

Ce sous-chapitre présente les lignes d'éléments de génération d'un Bloc RDMS.


```
-----
                           VARIANTE CICS/VSAM                *PDMB.DDDD.BMS.203
ELEMENTS GENERATION DE BLOC B.D.   QRBLOC                                100
A NLG : T DESCRIPTION                                                     BIBLI
* 150 : G CREATE PERMANENT TABLE (NOM)                                  *VIRT
* 200 : G       IN                  (NOM SCHEMA.NOM STORAGE-AREA)       *VIRT
* 250 :          ---> DEBUT D'INSERTION POUR LES COLONNES<---          *VIRT
* 450 :          ---> FIN D'INSERTION POUR LES COLONNES<---          *VIRT
* 600 : G       PRIMARY KEY      (NOM)           IS (NOM COLONNE)       *VIRT
:
:
:
:
:
:
:
:
:
:
:
:
O: C1 CH: -DR100GG
-----
```

```

-----
VARIANTE CICS/VSAM *PDMB.DDDD.BMS.203
ELEMENTS GENERATION DE BLOC B.D. QRBLOC 130
A NLG : T DESCRIPTION BIBLI
* 150 : G CREATE VIEW (NOM DE LA VUE) *VIRT
* 200 : ---> DEBUT D'INSERTION POUR LES COLONNES <--- *VIRT
* 400 : ---> FIN D'INSERTION POUR LES COLONNES <--- *VIRT
* 450 : G AS SELECT ALL *VIRT
* 500 : ---> DEBUT D'INSERTION DES COLONNES SOURCES <--- *VIRT
* 700 : ---> FIN D'INSERTION DES COLONNES SOURCES <--- *VIRT
* 800 : G FROM (NOM DES TABLES SOURCES) *VIRT
* 850 : G WHERE *GUID
* 900 : G GROUP BY *GUID
* 950 : G HAVING *GUID
:
:
:
:
:
:
:
:
O: C1 CH: -DR130GG
-----

```



```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.196
ELEMENTS GENERATION DE BLOC B.D.      QTBL0C                                350

A NLG : T DESCRIPTION                                                    BIBLI
* 150 : G CREATE VIEW (NOM DE LA VUE)                                    *VIRT
* 200 :      ----> DEBUT D'INSERTION POUR LES COLONNES <----          *VIRT
* 400 :      ----> FIN D'INSERTION POUR LES COLONNES <----          *VIRT
* 450 : G AS                                                            *VIRT
* 460 : G LOCKING TABLE _____ FOR EXCLUSIVE MODE                *GUID
* 480 : G SELECT                                                        *VIRT
* 500 :      ----> DEBUT D'INSERTION DES COLONNES SOURCES <----      *VIRT
* 700 :      ----> FIN D'INSERTION DES COLONNES SOURCES <----      *VIRT
* 800 : G FROM (NOM DES TABLES SOURCES)                               *VIRT
* 850 : G WHERE                                                         *GUID
:
:
:
:
:
:
:
:
:
:
:
O: C1 CH: -DR350 GG
-----

```



```

-----
                                 VARIANTE CICS/VSAM                *PDMB.D801.BMS.258
ELEMENTS GENERATION DE BLOC B.D.    QTBLOC                               700

A NLG : T DESCRIPTION                                             BIBLI
* 150 : G ALTER TABLE (NOM)                                     *VIRT
* 200 :      ----> DEBUT D'INSERTION POUR LES COLONNES <---    *VIRT
* 400 :      ----> FIN D'INSERTION POUR LES COLONNES<---      *VIRT
:
:
:
:
:
:
:
:
:
:
:
:
:
:
:
*** FIN ***
O: C1 CH: -DR700GG
-----

```

INTEREL RFM

ELEMENTS DE GENERATION : INTEREL RFM

Ce sous-chapitre présente les lignes d'éléments de génération d'un Bloc INTEREL RFM.


```

-----
                                 VARIANTE CICS/VSAM                               *PDMB.DDDD.BMS.196
ELEMENTS GENERATION DE BLOC B.D.      QUBLOC                                     350

A NLG : T DESCRIPTION                                                             BIBLI
* 150 : G CREATE VIEW (NOM DE LA VUE)                                           *VIRT
* 200 :      ----> DEBUT D'INSERTION POUR LES COLONNES <---                   *VIRT
* 400 :      ----> FIN D'INSERTION POUR LES COLONNES <---                     *VIRT
* 450 : G AS                                                                     *VIRT
* 480 : G SELECT                                                                  *VIRT
* 500 :      ----> DEBUT D'INSERTION DES COLONNES SOURCES <---                *VIRT
* 700 :      ----> FIN D'INSERTION DES COLONNES SOURCES <---                  *VIRT
* 800 : G FROM (NOM DES TABLES SOURCES)                                         *VIRT
* 850 : G WHERE                                                                    *GUID
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
O: C1 CH: -DR350 GG
-----

```



```

-----
VARIANTE CICS/VSAM *PDMB.DDDD.BMS.196
ELEMENTS GENERATION DE BLOC B.D. QBBLOC 350

A NLG : T DESCRIPTION BIBLI
* 150 : G CREATE VIEW (NOM DE LA VUE) *VIRT
* 200 : ----> DEBUT D'INSERTION POUR LES COLONNES <--- *VIRT
* 400 : ----> FIN D'INSERTION POUR LES COLONNES <--- *VIRT
* 450 : G AS SELECT *VIRT
* 500 : ----> DEBUT D'INSERTION DES COLONNES SOURCES <--- *VIRT
* 700 : ----> FIN D'INSERTION DES COLONNES SOURCES <--- *VIRT
* 800 : G FROM (NOM DES TABLES SOURCES) *VIRT
* 850 : G WHERE *GUID
:
:
:
:
:
:
:
:
:
:
:
O: C1 CH: -DR350GG
-----

```



```

-----
                                VARIANTE CICS/VSAM                               *PDMB.DDDD.BMS.196
ELEMENTS GENERATION DE BLOC B.D.      QYBLOC                                       130

A NLG : T DESCRIPTION                                                                BIBLI
* 150 : G CREATE VIEW (NOM DE LA VUE)                                              *VIRT
* 200 :      ----> DEBUT D'INSERTION POUR LES COLONNES <---                       *VIRT
* 400 :      ----> FIN D'INSERTION POUR LES COLONNES <---                        *VIRT
* 450 : G AS SELECT                                                                *VIRT
* 500 :      ----> DEBUT D'INSERTION DES COLONNES SOURCES <---                  *VIRT
* 700 :      ----> FIN D'INSERTION DES COLONNES SOURCES <---                   *VIRT
* 800 : G FROM (NOM DES TABLES SOURCES)                                          *VIRT
* 850 : G WHERE                                                                    *GUID
* 900 : G WITH CHECK OPTION                                                       *GUID
:
:
:
:
:
:
:
:
:
:
:
:
:
O: C1 CH: -DR130GG
-----

```

```

-----
                VARIANTE CICS/VSAM                          *PDMB.DDDD.BMS.196
ELEMENTS GENERATION DE BLOC B.D.    QYBLOC                          210

A NLG : T DESCRIPTION                                          BIBLI
* 150 : G   CREATE (UNIQUE) INDEX (NOM)                       *VIRT
* 200 : G     ON (NOM DE LA TABLE)                           *VIRT
* 300 :      ---> DEBUT D'INSERTION POUR LES COLONNES <---   *VIRT
* 400 :      ---> FIN D'INSERTION POUR LES COLONNES <---     *VIRT
* 420 : G   WITH FILLFACTOR = _                               *GUID
* 440 : G     IGNORE_DUP_KEY                                   *GUID
* 450 : G     SORTED_DATA                                     *GUID
* 460 : G     IGNORE_DUP_ROW / ALLOW_DUP_ROW                 *GUID
* 500 : G     ON _____                                    *GUID
:
:
:
:
:
:
:
:
:
:
O: C1 CH: -DR210GG
-----

```

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.D801.BMS.258
ELEMENTS GENERATION DE BLOC B.D.    QYBLOC                                110

A NLG : T DESCRIPTION                                                    BIBLI
* 150 : G ALTER TABLE (NOM)                                           *VIRT
* 180 : G      ADD                                                       *VIRT
* 190 : G      CONSTRAINT (NOM DE LA CONTRAINTE)                       *VIRT
* 195 : G      PRIMARY KEY                                              *VIRT
* 195 : G      NONCLUSTERED                                             *GUID
* 200 :      ----> DEBUT D'INSERTION POUR LES COLONNES <----         *VIRT
* 400 :      ----> FIN D'INSERTION POUR LES COLONNES <----           *VIRT
* 500 : G      WITH FILLFACTOR = _                                       *GUID
* 500 : G      ON _____                                             *GUID
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
      :
*** FIN ***
O: C1 CH: -DR110GG
-----

```

Chapitre 8. Génération d'un Bloc

Génération du DDL (B.....GN)

La génération du DDL est faite à partir des informations logiques indiquées par l'intermédiaire :

- des Rubriques et Segments qui définissent Tables, Vues et Colonnes,
- de la définition et de la description des Blocs qui définissent Bases de données, Espaces et Index,

et des informations physiques précisées sur les lignes d'éléments de génération affectées au Bloc et à ses lignes de description.

Les ordres SQL sont générés et visualisés en TP à partir de l'écran obtenu par les choix :

B.....GN

pour la génération de la Base de données, ou

B.....GN nnn

pour la génération du DDL de l'objet appelé sur la ligne 'nnn' de description du Bloc.

L'écran de génération d'un objet est également obtenu en indiquant, sur l'écran de description du Bloc, la valeur 'Q' dans la zone CODE ACTION de la ligne décrivant l'objet et en transmettant.

Un premier écran demande de compléter le TYPE et le NOM EXTERNE (pour les Bases de données et Espaces) ou le CODE SEGMENT (Tables, Vues ou Index).

Ces zones sont automatiquement renseignées lorsque l'écran est obtenu par le choix -GNnnn ou le code action 'Q'; le code action 'Q' n'est pas autorisé lorsque la zone TYPE DE MOUVEMENT GENERE est à blanc.

Le type d'ordre indiqué sur la description du Bloc est également indiqué.

La touche 'ENTREE' permet alors de déclencher la génération du DDL, ainsi que sa visualisation.

Il est possible de passer directement de l'écran -GN d'un objet à celui d'un autre objet en renseignant :

- pour les Espaces :
le TYPE et le NOM EXTERNE
- pour les Tables, Vues et Index :
le TYPE et le CODE SEGMENT

Note : La commande -GN ne génère les objets SQL PACKAGE (C), FONCTION (E), PROCEDURE (Q), TRIGGER ou RULE (R) qu'en procédure batch.

Lors de la procédure de génération :

- les données locales ne sont pas prises en compte.
- pour les données colonne, le code Rubrique est remplacé par son nom relationnel.

GENERATION DES ORDRES 'ALTER TABLE'

Les ordres ALTER TABLE sont générés pour les lignes I et J. Pour modifier une Table, il est nécessaire de modifier sa description, c'est-à-dire la description du Segment correspondant.

La génération de la commande ALTER sur l'écran B.....GN se fait alors en deux phases :

L'écran une fois renseigné, après transmission, la description est affichée. Il faut alors sélectionner les Colonnes pour lesquelles le DDL doit être généré en entrant la valeur 'C' dans la zone SELECTION.

Pour ORACLE, la valeur 'M' peut également être utilisée.

Les ordres ALTER TABLE apparaissent de trois façons :

- ALTER TABLE sur les lignes I et J pour les Primary et Foreign Keys.
- ALTER TABLE sur les lignes de type A pour mise à jour des Colonnes, dans ce cas indiquer les Colonnes par -DR...K.
- ALTER TABLE sur les lignes de type T avec le code mouvement M, dans ce cas l'étape GN se fait en 2 phases.

VARIANTE CICS/VSAM *PDMB.DDDD.BMS.202
GENERATION DES ORDRES SQL QCBLOC EXEMPLE DATACOM/DB

SEGMENT : DZ10 NOM EXTERNE :
TYPE : A
MAJ CATALOGUE O/N :

ALTER TABLE DZ10

O: C1 CH:

SAISIE DES RUBRIQUES	VARIANTE	CICS/VSAM	QCBLOC	EXEMPLE	DB	*PDMB.DDDD.BMS.202	700
			DZ10				
			RANG	ORDRE		COLONNE	BIBLI
COCARA						QTMLI A	0067
NUCOM			03			FOURNP M	0067
FOURNP			02	M		NUCOM	0067
QTMLI			01	A		QTMCO D	0067
QTMCO			04	D		INFOR	0067
INFOR			05				0067
*** FIN ***							
0: C1 CH:							

Mise à jour du catalogue

Le DDL généré en TP peut être utilisé pour mettre à jour directement le catalogue de la base de données.

Ceci permet de prendre en compte systématiquement les modifications effectuées dans les descriptions des bases de données.

Après transmission, le message suivant s'affiche sur l'écran de génération du DDL :

MISE A JOUR DU CATALOGUE O/N :

La mise à jour en TP du catalogue se demande en entrant la valeur O après ce message.

La mise en oeuvre du programme de mise à jour du catalogue est décrite dans les Manuels d'Exploitation.

DB2 : AUTORISATION DE MISE A JOUR DU CATALOGUE

Les deux principes suivants font que l'utilisateur doit posséder certaines autorisations :

- Le créateur d'une Table en est le propriétaire, son nom préfixe le nom de la TABLE.
- Un utilisateur peut manipuler une Table ne lui appartenant pas, si son propriétaire lui en a donné la possibilité.

Les Tables peuvent être préfixées de trois façons :

- a) avec un préfixe explicite,
- b) sans préfixe et le préfixage avec le USERID,
- c) sans préfixe et le préfixage avec le USERID de la base.

Deux modes permettent de générer les Tables et de mettre à jour le catalogue de la base de données.

Premier mode - la Génération Batch (GPRT) :

Dans les cas a et b, le LOGON TSO qui est utilisé pour mettre à jour le catalogue doit être autorisé, par le préfixe et par le USERID à travailler sur leurs Tables.

Dans le cas c, le LOGON TSO sera le préfixe.

Second mode - la Génération TP :

La mise à jour en temps réel du catalogue, depuis la transaction (CICS ou IMS/DC), impose une certaine discipline quant à l'organisation et l'attribution des autorisations. En effet, la règle est la suivante :

L'utilisateur est identifié différemment selon qu'il utilise CICS ou IMS/DC et qu'un outil de sécurité est employé ou non (se référer au sous-chapitre "Definition d'une Table ou d'une Vue"). Si la Table est préfixée (cas a et b), l'utilisateur identifié par la base de données doit être autorisé par le propriétaire de cette Table, c'est-à-dire le préfixe. Si la Table manipulée n'est pas préfixée (cas c), l'identifiant de l'utilisateur est utilisé comme préfixe.

Chapitre 9. Mode d'accès

TP

LISTE DES BLOCS BASES DE DONNEES

LCBaaaaaa

Liste des Blocs B.D. par code (à partir du Bloc 'aaaaaa').

LNBaaaaaa

Liste des Blocs B.D. par nom (à partir du Bloc de nom 'aaaaaa') (choix sensible aux majuscules/minuscules).

LTBaabbbbb

Liste des Blocs B.D. par type (à partir du type 'aa' et du Bloc 'bbbbbb').

LEBaaccccccc

Liste des Blocs B.D. par nom externe (à partir du type 'aa' et du nom externe 'ccccccc').

DESCRIPTION DU BLOC 'aaaaaa'

Baaaaaa (MAJ)

Définition du Bloc 'aaaaaa'.

BaaaaaaCR (MAJ)

Saisie des instances reliées au Bloc 'aaaaaa' par une Relation Utilisateur.

BaaaaaaGCbbb (MAJ)

Commentaires du Bloc 'aaaaaa' (à partir de la ligne 'bbb').

BaaaaaaGGbbb (MAJ)

Eléments de génération du Bloc 'aaaaaa' (à partir de la ligne 'bbb').

BaaaaaaGObbb (MAJ)

Options de génération du Bloc 'aaaaaa' (à partir de la ligne 'bbb').

BaaaaaaATbbbbbb

Texte associé au Bloc 'aaaaaa' (à partir du Texte 'bbbbbb').

BaaaaaaX

Utilisations du Bloc 'aaaaaa'.

BaaaaaaXBbbbbbb

Utilisation du Bloc 'aaaaaa' dans les PSB (à partir du PSB 'bbbbbb').

BaaaaaaXObbbbbbb

Utilisation du Bloc 'aaaaaa' dans les Ecrans (à partir de l'Écran 'bbbbbb').

BaaaaaaXObbbbbbbCSdddd

Utilisation du Bloc 'aaaaaa' dans les Ecrans à partir de l'Appel de Segments dans l'Écran 'bbbbbb' (à partir de la catégorie 'c' et du segment 'dddd'). Note : "c" est égal à & pour la catégorie en-tête.

BaaaaaaXObbbbbbbWccnnn

Utilisation du Bloc 'aaaaaa' dans les Ecrans à partir du descriptif des zones de travail de l'écran 'bbbbbb' (à partir de l'emplacement 'cc' et du numéro de ligne 'nnn').

BaaaaaaXQbbbbbb

Liste des entités reliées au Bloc 'aaaaaa' par Relation Utilisateur (à partir de la Relation 'aaaaaa').

BaaaaaaXVbbbbbb

Utilisation du Bloc 'aaaaaa' dans les Rapports (à partir du Rapport 'bbbbbb').

BaaaaaaXPbbbbbb

Utilisation du Bloc 'aaaaaa' dans les Programmes (à partir du Programme 'bbbbbb').

BaaaaaaXPbbbbbbWccnnn

Utilisation du Bloc 'aaaaa' dans les Programmes à partir du descriptif des zones de travail du programme 'bbbbbb' (à partir de l'emplacement 'cc' et du numéro de ligne 'nnn').

LISTES DES OBJETS RELATIONNELS-SQL

LTStdss

Liste des Objets Relationnels par type et par code (à partir du type 't' et du code 'ddss').

LESteeeeeeeeeee

Liste des Objets Relationnels par type et par nom externe (à partir du type 't' et du nom 'eeeeeeeeeee').

Note :

Si le nom externe est indiqué sur la définition du segment, il n'apparaît pas dans la liste.

DESCRIPTION DU BLOC RELATIONNEL-SQL 'aaaaa'

BaaaaaDRbbb (MAJ)

Description du Bloc 'aaaaa' de type relationnel (à partir de la ligne 'bbb').

BaaaaaDRbbbGCccc (MAJ)

Commentaires de la ligne 'bbb' du Bloc relationnel 'aaaaaa' (à partir de la ligne de commentaires 'ccc').

BaaaaaDRbbbGGccc (MAJ)

Éléments de génération de la ligne 'bbb' du Bloc relationnel 'aaaaaa' (à partir de la ligne d'éléments de génération 'ccc').

BaaaaaDRbbbK (MAJ)

Constitution de la clé du Bloc 'aaaaaa' sur la ligne numéro 'bbb'.

BaaaaaGN (MAJ)

Génération des ordres SQL pour le Bloc 'aaaaaa'.

BaaaaaGNnnn (MAJ)

Génération des ordres SQL pour la ligne 'nnn' de description du bloc 'aaaaa'.

NB : 'Baaaaa' peut être remplacé par '-' après la première transaction de type 'Baaaaa'.

Tous les éléments entre parenthèses sont facultatifs.

EaaaaaXKbbbb

Utilisation de la Rubrique dans les clés SQL (à partir du Segment 'bbbb').

LISTE DES CONTRAINTES D'INTEGRITE RELATIONNELLES

SbbbbCNaaaaa

Liste des contraintes d'intégrité du Segment 'bbbb' à partir du Bloc 'a' ou 'aaa'.

Edition génération

Listes :

LCB : Liste des Blocs Bases de Données classés par code :

Option C1 : sans mots-clés imprimés,

Option C2 : avec mots-clés imprimés.

LEB : Liste des Blocs Bases de Données classés par nom externe, sans mots-clés imprimés.

LTB : Liste des Blocs Bases de Données classés par type :

Options C1 : sans mots-clés imprimés,

Options C2 : avec mots-clés imprimés.

LKB : Liste des Blocs Bases de Données par mot-clé.

Une zone de sélection (SEL:) sur la commande permet de choisir les mots-clés implicites ('L') ou explicites ('M'). Sinon on obtient les deux. Les mots-clés sont précisés dans la zone suite.

LTS : Liste des objets SQL classés par code.

LES : Liste des objets SQL classés par nom externe.

Descriptions :

DTB : Définition, description, documentation, mots-clés et chaînages du Bloc dont le code est indiqué dans la zone CODE ENTITE. Si cette zone n'est pas renseignée, description de tous les blocs du type indiqué dans la zone TYPE, ou tous les blocs si on n'indique aucun type.

- Option C1 : sans la documentation ventilée;
- Option C2 : avec la documentation ventilée.

GENERATION :

GSQ : Génération du DDL pour le Bloc dont le code est indiqué dans la zone CODE ENTITE.

- Option C1 : Génération du DDL en mode création (CREATE DATABASE et CREATE de tous les objets du Bloc);
- Option C2 : Génération du DDL, suivant le type de génération demandé sur la description du bloc (-DR)

NOTE : Les seules commandes ALTER générées par la procédure GPRT sont les modifications indiquées sur l'écran de description (-DR) et détaillées dans l'écran Composition de la Clé ou d'un Alter Table (-DRnnnK).

- Option C3 : Génération de la commande LABEL pour les bases DB2, SQL/DS et SQL400.
- Option C4 : Génération de la commande COMMENT pour les bases DB2, SQL/DS, SQL400, ORACLE et NONSTOP/SQL.

La date, l'heure et le numéro de session apparaîtront en première ligne du généré.

FLS : Contrôle du flot. Permet d'indiquer un environnement de génération (Module ENVIRONNEMENT DE PRODUCTION) et de modifier ou compléter l'appel des cartes de contrôle indiqué sur la définition du Bloc.

Chapitre 10. Exemples

Ecrans communs

Ce chapitre présente des exemples de Blocs générés.

Pour tous les Blocs, vous trouverez un exemple de génération d'une Table, d'un Index et d'une Vue.

Pour les Blocs qui l'acceptent, vous trouverez aussi un exemple de génération d'un Espace,

Ce sous-chapitre présente les écrans communs à tous les types de Blocs : les demandes de génération d'un Espace, d'un Index et d'une Vue ainsi que la description des Segments utilisés pour les définir. Comme vous le verrez dans les sous-chapitres spécifiques aux types de Blocs, tous les blocs contiennent, dans leur description, trois lignes identiques (ligne 100 de création d'une Table, ligne 210 de création d'un Index et ligne 350 de création d'une Vue).

Ce sous-chapitre présente donc, dans l'ordre :

- la demande de génération d'une Table à partir du Segment DZ05, présent sur la ligne 100 de Description du Bloc,
- la description du Segment DZ05,
- la demande de génération d'un Index à partir du Segment DZ10, présent sur la ligne 210 de Description du Bloc,
- la description du Segment DZ10,
- la composition de l'Index,
- la demande de génération d'une Vue à partir du Segment DZ09, présent sur la ligne 350 de Description du Bloc,
- la description du Segment DZ09.

Les résultats de ces demandes de génération sont en revanche différents selon les types de Blocs. Il sont donc présentés dans les sous-chapitres spécifiques à chaque type de Bloc.


```

-----
                VARIANTE CICS/VSAM                      *PDMB.DDDD.BMS.197
GENERATION DES ORDRES SQL

SEGMENT   : DZ10   NOM EXTERNE   : INDZ10
TYPE      : I                               PREFIXE :
          CREATION INDEX

*** FIN ***
0: C1 CH: -GN210
-----

```


SAISIE DES RUBRIQUES		VARIANTE CICS/VSAM	*PDMB.DDDD.BMS.197	
INDZ10		DZ10	210	
	RANG	ORDRE	COLONNE	BIBLI
COCARA			FOURNP	0067
NUCOM	02	A	NUCOM	0067
FOURNP	01		QTMLI	0067
QTMLI	03			0067
QTMCO				0067
INFOR				0067

*** FIN ***
0: C1 CH: -DR210K

VARIANTE CICS/VSAM *PDMB.DDDD.BMS.197
GENERATION DES ORDRES SQL

SEGMENT : DZ09 NOM EXTERNE : VUDZ09DBE
TYPE : V PREFIXE :
CREATION VUE

*** FIN ***
O: C1 CH: -GN350

```

-----
                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.201
DESCRIPTION DU SEGMENT : DZ09 FOURNI. INTER

A NLG : CORUB FORM.INT U OCC GR I CMS456 CRNS VALEUR/FCT MAJ/TABLE DOC BIBLI
  110 : FO10                                **                                0067
1 090 : CLEFO                                D                                0067
1 100 : FOURNI X(3) D U S 0 0067
1 110 : MATE X(8) D S 000 000000 0067
1 120 : RELEA X(3) D 1 LIVRABLE 0067
1 130 : LANGU X D 0067
1 140 : QTMAS S9(4) C 2 0067
1 150 : QTMAM S9(4) C 0067
1 160 : LIBFO X(20) D 3 0067
1 200 : DATE M D S 0 0067
1 210 : HEURE T D S 0 0067
1 999 : FILLER XX D 0067
  120 : COMMEN X D 0067
* 130 : (-DBE)
:
:
:
: LIBELLE :

0: C2 CH: S dz09 CE
-----

```

DB2

EXEMPLE DB2

Ce sous-chapitre présente les écrans spécifiques d'un Bloc Base de Données de type DB2 (code 'Q2BLOC').

Vous trouverez, dans l'ordre :

- la description du bloc,
- la demande de génération d'un Espace, à partir de la ligne 080 de description du bloc,
- le résultat de la génération de cet Espace,
- le résultat de la génération d'une Table, à partir de la ligne 100 de description du bloc,
- le résultat de la génération d'un Index, à partir de la ligne 210 de description du bloc,
- le résultat de la génération d'une Vue, à partir de la ligne 350 de description du bloc.

Les demandes de génération et la description des Segments à partir desquels s'est effectuée la génération sont présentées dans le sous-chapitre 'Ecrans Communs' de ce chapitre.

VARIANTE CICS/VSAM		*PDMB.D801.BMS.258			
DESCRIP. BLOC RELATIONNEL Q2BLOC EXEMPLE DB2					
A NLG	: T NOM EXTERNE	CODE TABLE	TY	CODE	BIBLI
:	:	VUE	CLE	GEN	:
080	: P ESP1-TABLE-Q2BLOC			C	0067
100	: T DODZ05	DZ05		C	0067
110	: K	DZ05		C	0048
130	: V VUDZ05S3	DZ05	3	C	0058
200	: T DODZ10	DZ10		C	0067
210	: I INDZ10	DZ10		C	0048
220	: J CEXISTF DZ05	DZ10	C	C	0048
300	: T	F010		C	0067
350	: V VUDZ09DBE	DZ09		C	0067
360	: V VUDZ09S4	DZ09	4	C	0048
370	: A	F010		C	0256
:	:				
:	:				
:	:				
:	:				
:	:				
:	:				
:	:				
*** FIN ***					
0: C1 CH: BQ2BLOC DR					

VARIANTE CICS/VSAM *PDMB.DDDD.BMS.197
GENERATION DES ORDRES SQL Q2BLOC EXEMPLE BASE DB2

SEGMENT : NOM EXTERNE : ESP1-TABLE-Q2BLOC
TYPE : P
CREATION ESPACE TABLE

*** FIN ***
O: C1 CH: -GN080


```

VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.197
GENERATION DES ORDRES SQL  Q2BLOC EXEMPLE BASE DB2

SEGMENT   : DZ05  NOM EXTERNE   : DODZ05
TYPE      : T
          MAJ CATALOGUE O/N :

CREATE TABLE PDCL.DODZ05
(COCARA    CHAR          (00001)    NOT NULL,
NUCOD     SMALLINT      NOT NULL WITH DEFAULT,
FOURNI    CHAR          (00003),
NUCLIE    CHAR          (00008),
VILLE    VARCHAR      (00020),
CORESP    LONG VARCHAR,
REMISE    DECIMAL      (00006, 02),
MATERIEL  VARCHAR      (00008),
DATED     CHAR          (00010),
HEURE     TIME,
PRECIS    TIMESTAMP)
  IN EXTQ2DB2.ESP1-TABLE-Q2BLOC
;

*** FIN ***
O: C1 CH: -GN100

```

```

                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.197
GENERATION DES ORDRES SQL  Q2BLOC EXEMPLE BASE DB2

SEGMENT   : DZ10   NOM EXTERNE   : INDZ10
TYPE      : I
          MAJ CATALOGUE O/N :

CREATE INDEX PDCL.INDZ10
ON PDCL.DODZ10
(FOURNP
  NUCOM          ASC ,
  LIVRABLE       )
;

*** FIN ***
0: C1 CH: -GN210

```

```

                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.197
GENERATION DES ORDRES SQL      Q2BLOC EXEMPLE BASE DB2

SEGMENT      : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE         : V
              MAJ CATALOGUE O/N :

CREATE VIEW PDCL.VUDZ09DBE
(CLEFO      ,
FOURNI     ,
MATE       ,
RELEA     ,
LANGU     ,
QTMAS     ,
QTMAM     ,
LIBFO     ,
DATE      ,
HEURE     ,
FILLER    ,
COMMEN    )
AS SELECT
PDCL.FOUR.CLEFO
APPUYER SUR ENTREE POUR CONTINUER
O: C1 CH: -GN350

```

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.197
GENERATION DES ORDRES SQL  Q2BLOC EXEMPLE BASE DB2

SEGMENT   : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE      : V
          MAJ CATALOGUE O/N :

PDCL.FOUR.FOURNI
PDCL.FOUR.MATE
PDCL.FOUR.RELEA
PDCL.FOUR.LANGU
PDCL.FOUR.QTMAS
PDCL.FOUR.QTMAM
PDCL.FOUR.LIBFO
PDCL.FOUR.DATE
PDCL.FOUR.HEURE
PDCL.FOUR.FILLER
PDCL.DODZ05.COCARA
FROM PDCL.FOUR
     PDCL.DODZ05
;

*** FIN ***
O: C1 CH:
-----

```

SQL SERVER

EXEMPLE SQL SERVER

Ce sous-chapitre présente les écrans spécifiques d'un Bloc Base de Données de type SQL SERVER (code 'Q3BLOC').

Vous trouverez, dans l'ordre :

- la description du bloc,
- le résultat de la génération d'une Table, à partir de la ligne 100 de description du bloc,
- le résultat de la génération d'un Index, à partir de la ligne 210 de description du bloc,
- le résultat de la génération d'une Vue, à partir de la ligne 350 de description du bloc.

Les demandes de génération et la description des Segments à partir desquels s'est effectuée la génération sont présentées dans le sous-chapitre 'Ecrans Communs' de ce chapitre.


```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  Q3BLOC EXEMPLE SQL SERVER

SEGMENT   : DZ05   NOM EXTERNE   : DODZ05
TYPE      : T
          MAJ CATALOGUE O/N :

CREATE TABLE DODZ05
(COCARA   CHAR          (00001)   NOT NULL,
 NUCOD    SMALLINT      NULL,
 FOURNI   CHAR          (00003)   NULL,
 NUCLIE   CHAR          (00008)   NULL,
 *** CORESP RUBRIQUE DE LONGUEUR > 00255 NON PRISE EN COMPTE ***
 VILLE    VARCHAR       (00020)   NULL,
 REMISE   NUMERIC       (00006,02) NULL,
 MATERIEL VARCHAR       (00008)   NULL,
 DATED    DATETIME      NULL,
 HEURE    DATETIME      NULL,
 PRECIS   CHAR          (00026)   NULL)
;

*** FIN ***
O: C1 CH: -GN100
-----

```

VARIANTE CICS/VSAM *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL Q3BLOC EXEMPLE SQL SERVER

SEGMENT : DZ10 NOM EXTERNE : INDZ10
TYPE : I
MAJ CATALOGUE O/N :

```
CREATE INDEX INDZ10
ON DODZ10
(FOURNP          ,
NUCOM           ASC ,
LIVRABLE        )
;
```

*** FIN ***
O: C1 CH: -GNZ10

```

-----
                                VARIANTE CICS/VSAM                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  Q3BLOC EXEMPLE SQL SERVER

SEGMENT   : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE      : V
          MAJ CATALOGUE O/N :

CREATE VIEW VUDZ09DBE
(CLEFO
FOURNI
MATE
RELEA
LANGU
QTMAS
QTMAM
LIBFO
DATE
HEURE
FILLER
COMMEN
)
AS SELECT
FOUR.CLEFO
APPUYER SUR ENTREE POUR CONTINUER
O: C1 CH:
-----

```

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL   Q3BLOC EXEMPLE SQL SERVER

SEGMENT   : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE      : V
          MAJ CATALOGUE O/N :

FOUR.FOURNI
FOUR.MATE
FOUR.RELEA
FOUR.LANGU
FOUR.QTMAS
FOUR.QTMAM
FOUR.LIBFO
FOUR.DATE
FOUR.HEURE
FOUR.FILLER
DODZ05.COCARA
FROM FOUR
DODZ05
;

*** FIN ***
O: C1 CH:
-----

```

DBD/2

EXEMPLE DB2/2

Ce sous-chapitre présente les écrans spécifiques d'un Bloc Base de Données de type DB2/2 (code 'QBBLOC').

Vous trouverez, dans l'ordre :

- la description du bloc,
- le résultat de la génération d'une Table, à partir de la ligne 100 de description du bloc,
- le résultat de la génération d'un Index, à partir de la ligne 210 de description du bloc,
- le résultat de la génération d'une Vue, à partir de la ligne 350 de description du bloc.

Les demandes de génération et la description des Segments à partir desquels s'est effectuée la génération sont présentées dans le sous-chapitre 'Ecrans Communs' de ce chapitre.

VARIANTE CICS/VSAM		*PDMB.D801.BMS.259			
DESCRIPT. BLOC RELATIONNEL QBBLOC EXEMPLE DB2/2					
A NLG	: T NOM EXTERNE	CODE TABLE	TY	CODE	BIBLI
:	:	VUE	CLE	GEN	:
100	: T DODZ05	DZ05		C	0067
110	: K	DZ05		C	0058
130	: V VUDZ05S3	DZ05	3	C	0067
200	: T DODZ10	DZ10		C	0048
210	: I INDZ10	DZ10		C	0067
220	: J CEXISTF DZ05	DZ10	C	C	0067
300	: T	F010		C	0048
350	: V VUDZ09DBE	DZ09		C	+ 0219
360	: V VUDZ09S4	DZ09	4	C	
370	: A	F010		C	+
:	:				
:	:				
:	:				
:	:				
:	:				
*** FIN ***					
O: C1 CH: BQBBLOC DR					

```

                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QBBLOC EXEMPLE DB2/2

SEGMENT   : DZ05   NOM EXTERNE   : DODZ05
TYPE      : T
          MAJ CATALOGUE O/N :

CREATE TABLE PDCL.DODZ05
(COCARA    CHAR          (00001)    NOT NULL,
 NUCOD     SMALLINT      NOT NULL WITH DEFAULT,
 FOURNI    CHAR          (00003),
 NUCLIE    CHAR          (00008),
 VILLE     VARCHAR       (00020),
 CORESP    VARCHAR       (00256),
 REMISE    DECIMAL       (00006, 02),
 MATERIEL  VARCHAR       (00008),
 DATED     CHAR          (00010),
 HEURE     TIME,
 PRECIS    TIMESTAMP)
;

*** FIN ***
O: C1 CH: -GN100

```

```

                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QBBLOC EXEMPLE DB2/2

SEGMENT   : DZ10   NOM EXTERNE   : INDZ10
TYPE      : I
          MAJ CATALOGUE O/N :

CREATE INDEX PDCL.INDZ10
ON PDCL.DODZ10
(FOURNP
 NUCOM          ASC ,
 LIVRABLE      )
;

*** FIN ***
0: C1 CH: -GN210

```

```

                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL   QBBLOC EXEMPLE DB2/2

SEGMENT   : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE      : V
          MAJ CATALOGUE O/N :

CREATE VIEW PDCL.VUDZ09DBE
(CLEFO      ,
FOURNI     ,
MATE       ,
RELEA     ,
LANGU     ,
QTMAS     ,
QTMAM     ,
LIBFO     ,
DATE      ,
HEURE     ,
FILLER    ,
COMMEN    )
AS SELECT
PDCL.FOUR.CLEFO
APPUYER SUR ENTREE POUR CONTINUER
O: C1 CH: -GN350

```

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QBBLOC EXEMPLE DB2/2

SEGMENT   : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE      : V
          MAJ CATALOGUE O/N :

PDCL.FOUR.FOURNI
PDCL.FOUR.MATE
PDCL.FOUR.RELEA
PDCL.FOUR.LANGU
PDCL.FOUR.QTMAS
PDCL.FOUR.QTMAM
PDCL.FOUR.LIBFO
PDCL.FOUR.DATE
PDCL.FOUR.HEURE
PDCL.FOUR.FILLER
PDCL.DODZ05.COCARA
FROM PDCL.FOUR
      PDCL.DODZ05
;

*** FIN ***
O: C1 CH:
-----

```

DATAKOM/DB

EXEMPLE DATAKOM/DB

Ce sous-chapitre présente les écrans spécifiques d'un Bloc Base de Données de type DATAKOM/DB (code 'QCBLOC').

Vous trouverez, dans l'ordre :

- la description du bloc,
- la demande de génération d'un Espace, à partir de la ligne 080 de description du bloc,
- le résultat de la génération de cet Espace,
- le résultat de la génération d'une Table, à partir de la ligne 100 de description du bloc,
- le résultat de la génération d'un Index, à partir de la ligne 210 de description du bloc,
- le résultat de la génération d'une Vue, à partir de la ligne 350 de description du bloc.

Les demandes de génération et la description des Segments à partir desquels s'est effectuée la génération sont présentées dans le sous-chapitre 'Ecrans Communs' de ce chapitre.

VARIANTE CICS/VSAM					*PDMB.D801.BMS.259
DESCRIPT.	BLOC	RELATIONNEL	QCBLOC	EXEMPLE	DATA/DB
A NLG	:	T	NOM	EXTERNE	
	:				
080	:	P	ESP1-TABLE-QCBLOC		
100	:	T	DODZ05	DZ05	
130	:	V	VUDZ05S3	DZ05	3
200	:	T	DODZ10	DZ10	
210	:	I	INDZ10	DZ10	
220	:	J	CEXISTF DZ05	DZ10	
300	:	T		F010	
350	:	V	VUDZ09DBE	DZ09	
360	:	V	VUDZ09S4	DZ09	4
	:				
	:				
	:				
	:				
	:				
*** FIN ***					
O: C1 CH: BQCBLOC DR					

VARIANTE CICS/VSAM *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL QCBLOC EXEMPLE DATACOM/DB

SEGMENT : NOM EXTERNE : ESP1-TABLE-QCBLOC
TYPE : P
CREATION SCHEMA

*** FIN ***
O: C1 CH: -GN080

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198|
GENERATION DES ORDRES SQL   QCBLOC EXEMPLE DATACOM/DB
SEGMENT   :                   NOM EXTERNE   : ESP1-TABLE-QCBLOC
TYPE      : P
          MAJ CATALOGUE O/N :
CREATE SCHEMA AUTHORIZATION ESP1-TABLE-QCBLOC
;

*** FIN ***
0: C1 CH:
-----

```

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QCBLOC EXEMPLE DATCOM/DB

SEGMENT   : DZ05   NOM EXTERNE   : DODZ05
TYPE      : T
          MAJ CATALOGUE O/N :

CREATE TABLE DODZ05
(COCARA   CHAR          (00001)      NOT NULL,
NUCOD    SMALLINT      NOT NULL WITH DEFAULT,
FOURNI   CHAR          (00003),
NUCLIE   CHAR          (00008),
VILLE   CHAR          (00020),
CORESP   CHAR          (00256),
REMISE   DECIMAL      (00006,02),
MATERIEL CHAR          (00008),
DATED    CHAR          (00010),
HEURE    TIME,
PRECIS   TIMESTAMP,
;

*** FIN ***
O: C1 CH: -GN100
-----

```

```
-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QCBLOC EXEMPLE DATACOM/DB

SEGMENT   : DZ10   NOM EXTERNE   : INDZ10
TYPE      : I
          MAJ CATALOGUE O/N :

CREATE INDEX INDZ10
ON DODZ10
(FOURNP           ,
 NUCOM            ,
 LIVRABLE         )
;

*** FIN ***
0: C1 CH: -GN210
-----
```

```

                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL      QCBLOC EXEMPLE DATA/DB

SEGMENT      : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE         : V
              MAJ CATALOGUE O/N :

CREATE VIEW VUDZ09DBE
(CLEFO      ,
FOURNI     ,
MATE       ,
RELEA     ,
LANGU     ,
QTMAS     ,
QTMAM     ,
LIBFO     ,
DATE      ,
HEURE     ,
FILLER    ,
COMMEN    )
AS SELECT
FOUR.CLEFO
APPUYER SUR ENTREE POUR CONTINUER
O: C1 CH: -GN350

```

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QCBLOC EXEMPLE DATACOM/DB

SEGMENT   : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE      : V
          MAJ CATALOGUE O/N :

FOUR.FOURNI
FOUR.MATE
FOUR.RELEA
FOUR.LANGU
FOUR.QTMAS
FOUR.QTMAM
FOUR.LIBFO
FOUR.DATE
FOUR.HEURE
FOUR.FILLER
DODZ05.COCARA
FROM FOUR
   DODZ05
;

*** FIN ***
O: C1 CH:
-----

```

NONSTOP SQL

EXEMPLE NONSTOP SQL

Ce sous-chapitre présente les écrans spécifiques d'un Bloc Base de Données de type NONSTOP SQL (code 'QNBLOC').

Vous trouverez, dans l'ordre :

- la description du bloc,
- le résultat de la génération d'une Table, à partir de la ligne 100 de description du bloc,
- le résultat de la génération d'un Index, à partir de la ligne 210 de description du bloc,
- le résultat de la génération d'une Vue, à partir de la ligne 350 de description du bloc.

Les demandes de génération et la description des Segments à partir desquels s'est effectuée la génération sont présentées dans le sous-chapitre 'Ecrans Communs' de ce chapitre.


```

-----
                                VARIANTE CICS/VSAM                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QNBLOC EXEMPLE NONSTOP SQL

SEGMENT   : DZ05   NOM EXTERNE   : DODZ05
TYPE      : T
          MAJ CATALOGUE O/N :

CREATE TABLE DZ05
(COCARA          CHAR          (00001)          NOT NULL,
 NUCOD           SMALLINT      SIGNED          DEFAULT SYSTEM,
 FOURNI          CHAR          (00003),
 NUCLIE          CHAR          (00008),
 VILLE           VARCHAR      (00020),
 *** REMISE USAGE ERRONE ***
 CORESP          VARCHAR      (00256),
 MATERIEL        VARCHAR      (00008),
 DATED           DATETIME YEAR TO HOUR,
 HEURE           TIME,
 PRECIS          TIMESTAMP)
;

*** FIN ***
O: C1 CH: -GN100
-----

```

VARIANTE CICS/VSAM *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL QNBLOC EXEMPLE NONSTOP SQL

SEGMENT : DZ10 NOM EXTERNE : INDZ10
TYPE : I
MAJ CATALOGUE O/N :

```
CREATE INDEX INDZ10
ON DODZ10
(FOURNP
NUCOM          ASC ,
LIVRABLE       )
;
```

*** FIN ***
O: C1 CH: -GNZ10

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QNBLOC EXEMPLE NONSTOP SQL

SEGMENT   : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE      : V
          MAJ CATALOGUE O/N :

CREATE VIEW VUDZ09DBE
(CLEFO
FOURNI
MATE
RELEA
LANGU
QTMAS
QTMAM
LIBFO
DATE
HEURE
FILLER
COMMEN
)
AS SELECT
FOUR.CLEFO
APPUYER SUR ENTREE POUR CONTINUER
O: C1 CH: -GN350
-----

```

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QNBLOC EXEMPLE NONSTOP SQL

SEGMENT   : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE      : V
          MAJ CATALOGUE O/N :

FOUR.FOURNI                                     ,
FOUR.MATE                                       ,
FOUR.RELEA                                     ,
FOUR.LANGU                                     ,
FOUR.QTMAS                                     ,
FOUR.QTMAM                                     ,
FOUR.LIBFO                                     ,
FOUR.DATE                                      ,
FOUR.HEURE                                    ,
FOUR.FILLER                                   ,
DODZ05.COCARA
FROM FOUR                                     ,
   DODZ05
;

*** FIN ***
O: C1 CH:
-----

```

ORACLE (<V6)

EXEMPLE ORACLE

Ce sous-chapitre présente les écrans spécifiques d'un Bloc Base de Données de type Oracle (code 'QOBLOC').

Vous trouverez, dans l'ordre :

- la description du bloc,
- la demande de génération d'un Espace-Table, à partir de la ligne 080 de description du bloc,
- le résultat de la génération de cet Espace-Table,
- le résultat de la génération d'une Table, à partir de la ligne 100 de description du bloc,
- le résultat de la génération d'un Index, à partir de la ligne 210 de description du bloc,
- le résultat de la génération d'une Vue, à partir de la ligne 350 de description du bloc.

Les demandes de génération et la description des Segments à partir desquels s'est effectuée la génération sont présentées dans le sous-chapitre 'Ecrans Communs' de ce chapitre.

VARIANTE CICS/VSAM *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL QOBLOC EXEMPLE ORACLE

SEGMENT : NOM EXTERNE : ESP1-TABLE-QOBLOC
TYPE : P
CREATION D'UN SPACE

*** FIN ***
0: C1 CH: -GN080

VARIANTE CICS/VSAM *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL QOBLOC EXEMPLE ORACLE

SEGMENT : NOM EXTERNE : ESP1-TABLE-QOBLOC
TYPE : P
MAJ CATALOGUE O/N :

CREATE SPACE ESP1-TABLE-QOBLOC
;

*** FIN ***
O: C1 CH:

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QOBLOC EXEMPLE ORACLE

SEGMENT   : DZ05   NOM EXTERNE   : DODZ05
TYPE      : T
          MAJ CATALOGUE O/N :

CREATE TABLE DODZ05
(COCARA   CHAR          (00001)   NOT NULL,
 NUCOD    INTEGER,
 FOURNI   CHAR          (00003),
 NUCLIE   CHAR          (00008),
 VILLE    VARCHAR       (00020),
 CORESP   LONG VARCHAR,
 REMISE   NUMBER        (00006,02),
 MATERIEL VARCHAR       (00008),
 DATED    DATE,
 HEURE    DATE,
 PRECIS   CHAR          (00026))
;

*** FIN ***
O: C1 CH: -GN100
-----

```

```
-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QOBLOC EXEMPLE ORACLE

SEGMENT   : DZ10  NOM EXTERNE   : INDZ10
TYPE      : I
          MAJ CATALOGUE O/N :

CREATE INDEX INDZ10
ON DODZ10
(FOURNP
NUCOM          ASC ,
LIVRABLE      )
;

*** FIN ***
O: C1 CH: -GNZ10
-----
```

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QOBLOC EXEMPLE ORACLE

SEGMENT   : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE      : V
          MAJ CATALOGUE O/N :

CREATE VIEW VUDZ09DBE
(CLEFO
FOURNI
MATE
RELEA
LANGU
QTMAS
QTMAM
LIBFO
DATE
HEURE
FILLER
COMMEN
)
AS SELECT
FOUR.CLEFO
APPUYER SUR ENTREE POUR CONTINUER
O: C1 CH: -GN350
-----

```

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL      QOBLOC EXEMPLE ORACLE

SEGMENT      : DZ09  NOM EXTERNE   : VUDZ09DBE
TYPE         : V
              MAJ CATALOGUE O/N :

FOUR.FOURNI
FOUR.MATE
FOUR.RELEA
FOUR.LANGU
FOUR.QTMAS
FOUR.QTMAM
FOUR.LIBFO
FOUR.DATE
FOUR.HEURE
FOUR.FILLER
DODZ05.COCARA
FROM FOUR
DODZ05
;

*** FIN ***
O: C1 CH:
-----

```

ORACLE V7

EXEMPLE ORACLE V7

Ce sous-chapitre présente les écrans spécifiques d'un Bloc Base de Données de type ORACLE V7 (code 'QP Bloc').

Vous trouverez, dans l'ordre :

- la description du bloc,
- la demande de génération d'un Espace, à partir de la ligne 080 de description du bloc,
- le résultat de la génération de cet Espace,
- le résultat de la génération d'une Table, à partir de la ligne 100 de description du bloc,
- le résultat de la génération d'un Index, à partir de la ligne 210 de description du bloc,
- le résultat de la génération d'une Vue, à partir de la ligne 350 de description du bloc.
- la demande de modification de la Table,
- le résultat de la modification de la Table.

Les demandes de génération et la description des Segments à partir desquels s'est effectuée la génération sont présentées dans le sous-chapitre 'Ecrans Communs' de ce chapitre.

```

-----
                VARIANTE CICS/VSAM                                *PDMB.D801.BMS.259
DESCRIPT. BLOC RELATIONNEL  QPBLOC EXEMPLE ORACLE V7

A NLG : T NOM EXTERNE                CODE TABLE      TY CODE          BIBLI
      :                               VUE              CLE GEN
080 : P ESP1-TABLE-QPBLOC                C              C              0067
100 : T DODZ05                            DZ05             C              0058
130 : V VUDZ05S3                          DZ05             3 C              0067
200 : T DODZ10                            DZ10             C              0048
210 : I INDZ10                            DZ10             C              0067
300 : T                                    F010             C              0067
350 : V VUDZ09DBE                        DZ09             C              0048
360 : V VUDZ09S4                          DZ09             4 C              0213
510 : A ADDBTABLE                        DZ05             C              0219
700 : T EXTERNAL-PI00                    PI00             C              + 0219
805 : K                                    PI00             C              0219
810 : E FONCTION                          C              C              0219
      :
      :
      :
      :
*** FIN ***
O: C1 CH: BQPLOC DR
-----

```

```
-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QPBLOC EXEMPLE ORACLE V7

SEGMENT   :          NOM EXTERNE   : ESP1-TABLE-QPBLOC
TYPE      : P
          CREATION ESPACE TABLE

*** FIN ***
0: C1 CH: -GN080
-----
```

```
-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QPBLOC EXEMPLE ORACLE V7

SEGMENT   :          NOM EXTERNE   : ESP1-TABLE-QPBLOC
TYPE      : P
          MAJ CATALOGUE O/N :

CREATE TABLESPACE ESP1-TABLE-QPBLOC
;

*** FIN ***
O: C1 CH:
```

```

-----
                                VARIANTE CICS/VSAM                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QPBLOC EXEMPLE ORACLE V7

SEGMENT   : DZ05   NOM EXTERNE   : DODZ05
TYPE      : T
          MAJ CATALOGUE O/N :

CREATE TABLE DODZ05
(COCARA   VARCHAR2   (00001)   NOT NULL,
 NUCOD    INTEGER,
 FOURNI   VARCHAR2   (00003),
 NUCLIE   VARCHAR2   (00008),
 VILLE    VARCHAR    (00020),
 CORESP   VARCHAR    (00256),
 REMISE   NUMBER     (00006,02),
 MATERIEL VARCHAR    (00008),
 DATED    DATE,
 HEURE    DATE,
 PRECIS   VARCHAR2   (00026))
TABLESPACE ESP1-TABLE-QPBLOC
;

*** FIN ***
O: C1 CH: -GN100
-----

```

VARIANTE CICS/VSAM *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL QPBLOC EXEMPLE ORACLE V7

SEGMENT : DZ10 NOM EXTERNE : INDZ10
TYPE : I
MAJ CATALOGUE O/N :

```
CREATE INDEX INDZ10
ON DODZ10
(FOURNP          ,
NUCOM           ASC ,
LIVRABLE        )
;
```

*** FIN ***
O: C1 CH: -GN210

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QPBLOC EXEMPLE ORACLE V7

SEGMENT   : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE      : V
          MAJ CATALOGUE O/N :

CREATE VIEW VUDZ09DBE
(CLEFO
FOURNI
MATE
RELEA
LANGU
QTMAS
QTMAM
LIBFO
DATE
HEURE
FILLER
COMMEN
)
AS SELECT
FOUR.CLEFO
APPUYER SUR ENTREE POUR CONTINUER
O: C1 CH: -GN350
-----

```

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QPBLOC EXEMPLE ORACLE V7

SEGMENT   : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE      : V
          MAJ CATALOGUE O/N :

FOUR.FOURNI
FOUR.MATE
FOUR.RELEA
FOUR.LANGU
FOUR.QTMAS
FOUR.QTMAM
FOUR.LIBFO
FOUR.DATE
FOUR.HEURE
FOUR.FILLER
DODZ05.COCARA
FROM FOUR
   DODZ05
;
*** FIN ***
O: C1 CH:
-----

```

VARIANTE CICS/VSAM *PDMB.D801.BMS.259
GENERATION DES ORDRES SQL QPBLOC EXEMPLE ORACLE V7

SEGMENT : DZ05 NOM EXTERNE :
TYPE : A
MODIFICATION TABLE

*** FIN ***
0: C1 CH: -GN510

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.D801.BMS.259
GENERATION DES ORDRES SQL  QPBLOC EXEMPLE ORACLE V7

SEGMENT   : DZ05   NOM EXTERNE   :
TYPE      : A
          MAJ CATALOGUE O/N :

ALTER TABLE DZ05

O: C1 CH:
-----

```

RDMS

EXEMPLE RDMS

Ce sous-chapitre présente les écrans spécifiques d'un Bloc Base de Données de type RDMS (code 'QRBLOC').

Vous trouverez, dans l'ordre :

- la description du bloc,
- la demande de génération d'un Espace, à partir de la ligne 080 de description du bloc,
- le résultat de la génération de cet Espace,
- le résultat de la génération d'une Table, à partir de la ligne 100 de description du bloc,
- le résultat de la génération d'un Index, à partir de la ligne 210 de description du bloc,
- le résultat de la génération d'une Vue, à partir de la ligne 350 de description du bloc.

Les demandes de génération et la description des Segments à partir desquels s'est effectuée la génération sont présentées dans le sous-chapitre 'Ecrans Communs' de ce chapitre.


```
-----  
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198  
GENERATION DES ORDRES SQL   QRBLOC EXEMPLE RDMS
```

```
SEGMENT   :          NOM EXTERNE   : ESP1-TABLE-QRBLOC  
TYPE      : P  
          CREATION D'UN STORAGE-AREA
```

```
*** FIN ***  
0: C1 CH: -GN080
```

VARIANTE CICS/VSAM *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL QRBLOC EXEMPLE RDMS

SEGMENT : NOM EXTERNE : ESP1-TABLE-QRBLOC
TYPE : P
MAJ CATALOGUE O/N :

CREATE STORAGE-AREA ESP1-TABLE-QRBLOC
FOR SCHEMA EXTQRBLC
;

*** FIN ***
O: C1 CH:

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QRBLOC EXEMPLE RDMS

SEGMENT   : DZ05   NOM EXTERNE   : DODZ05
TYPE      : T
          MAJ CATALOGUE O/N :

CREATE TABLE DZ05
IN EXTQRBLC.ESP1-TABLE-QRBLOC
COLUMNS ARE COCARA      : CHARACTER (00001)  NOT NULL ,
                   NUCOD      : DECIMAL (00004)      ,
                   FOURNI     : CHARACTER (00003)      ,
                   NUCLIE     : CHARACTER (00008)      ,

                   VILLE      : CHARACTER (00020)      ,
                   CORESP     : CHARACTER (00256)      ,
                   REMISE     : DECIMAL (00007.02)     ,
                   MATERIEL   : CHARACTER (00008)      ,
                   DATED      : CHARACTER (00010)      ,
                   HEURE      : CHARACTER (00008)      ,
                   PRECIS     : CHARACTER (00026)

;

*** FIN ***
O: C1 CH: -GN100
-----

```

```
-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QRBLOC EXEMPLE RDMS

SEGMENT   : DZ10  NOM EXTERNE   : INDZ10
TYPE      : I
          MAJ CATALOGUE O/N :

CREATE INDEX INDZ10
ON DODZ10
(FOURNP
NUCOM          ASC ,
LIVRABLE      )
;

*** FIN ***
O: C1 CH: -GN210
-----
```

```

                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QRBLOC EXEMPLE RDMS

SEGMENT   : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE      : V
          MAJ CATALOGUE O/N :

CREATE VIEW VUDZ09DBE
(CLEFO
FOURNI
MATE
RELEA
LANGU
QTMAS
QTMAM
LIBFO
DATE
HEURE
FILLER
COMMEN
)
AS SELECT
FOUR.CLEFO
APPUYER SUR ENTREE POUR CONTINUER
0: C1 CH: -GN350

```

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL   QRBLOC EXEMPLE RDMS

SEGMENT   : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE      : V
          MAJ CATALOGUE O/N :

FOUR.FOURNI
FOUR.MATE
FOUR.RELEA
FOUR.LANGU
FOUR.QTMAS
FOUR.QTMAM
FOUR.LIBFO
FOUR.DATE
FOUR.HEURE
FOUR.FILLER
DODZ05.COCARA
FROM FOUR
DODZ05
;

*** FIN ***
O: C1 CH:
-----

```

SQL/DS

EXEMPLE SQL/DS

Ce sous-chapitre présente les écrans spécifiques d'un Bloc Base de Données de type SQL/DS (code 'QSBLOC').

Vous trouverez, dans l'ordre :

- la description du bloc,
- la demande de génération d'un Espace, à partir de la ligne 080 de description du bloc,
- le résultat de la génération de cet Espace,
- le résultat de la génération d'une Table, à partir de la ligne 100 de description du bloc,
- le résultat de la génération d'un Index, à partir de la ligne 210 de description du bloc,
- le résultat de la génération d'une Vue, à partir de la ligne 350 de description du bloc.

Les demandes de génération et la description des Segments à partir desquels s'est effectuée la génération sont présentées dans le sous-chapitre 'Ecrans Communs' de ce chapitre.

VARIANTE CICS/VSAM				*PDMB.D801.BMS.259		
DESCRPT.	BLOC	RELATIONNEL	QSBLOC	EXEMPLE	SQL/DS	
A NLG	:	T NOM EXTERNE		CODE TABLE	TY CODE	BIBLI
	:			VUE	CLE GEN	
080	:	P ESP1-TABLE-QSBLOC			C	0067
100	:	T DODZ05		DZ05	C	0058
110	:	K		DZ05	C	0067
130	:	V VUDZ05S3		DZ05	3 C	0067
200	:	T DODZ10		DZ10	C	0067
210	:	I INDZ10		DZ10	C +	0048
220	:	J CEXISTF DZ05		DZ10	C C	0219
300	:	T		F010	C	0219
350	:	V VUDZ09DBE		DZ09	C	0219
360	:	V VUDZ09S4		DZ09	4 C	
700	:	T EXTERNAL-PI00		PI00	C	
805	:	K		PI00	C +	
	:					
	:					
	:					
*** FIN ***						
O: C1 CH: BQSBLOC DR						

VARIANTE CICS/VSAM *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL QSBLOC EXEMPLE SQL/DS

SEGMENT : NOM EXTERNE : ESP1-TABLE-QSBLOC
TYPE : P
ALLOCATION D'UN DBSPACE

*** FIN ***
0: C1 CH: -GN080

```
-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QSBLOC EXEMPLE SQL/DS

SEGMENT   :          NOM EXTERNE   : ESP1-TABLE-QSBLOC
TYPE      : P
          MAJ CATALOGUE O/N :

ACQUIRE PUBLIC DBSPACE NAMED ESP1-TABLE-QSBLOC
;

*** FIN ***
O: C1 CH:
-----
```

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QSBLOC EXEMPLE SQL/DS

SEGMENT   : DZ05   NOM EXTERNE   : DODZ05
TYPE      : T
          MAJ CATALOGUE O/N :

CREATE TABLE PDCL.DODZ05
(COCARA    CHAR          (00001)    NOT NULL,
 NUCOD     SMALLINT,
 FOURNI    CHAR          (00003),
 NUCLIE    CHAR          (00008),
 VILLE     VARCHAR       (00020),
 CORESP    LONG VARCHAR,
 REMISE    DECIMAL       (00006, 02),
 MATERIEL  VARCHAR       (00008),
 DATED     CHAR          (00010),
 HEURE     TIME,
 PRECIS    TIMESTAMP)
          IN EXQSBLOC.ESP1-TABLE-QSBLOC
;

*** FIN ***
O: C1 CH: -GN100
-----

```

```

                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QSBLOC EXEMPLE SQL/DS

SEGMENT   : DZ10   NOM EXTERNE   : INDZ10
TYPE      : I
          MAJ CATALOGUE O/N :

CREATE INDEX PDCL.INDZ10
ON PDCL.DODZ10
(FOURNP
NUCOM          ASC ,
LIVRABLE      )
;

*** FIN ***
O: C1 CH: -GN210

```

```

                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QSBLOC EXEMPLE SQL/DS

SEGMENT   : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE      : V
          MAJ CATALOGUE O/N :

CREATE VIEW PDCL.VUDZ09DBE
(CLEFO
FOURNI
MATE
RELEA
LANGU
QTMAS
QTMAM
LIBFO
DATE
HEURE
FILLER
COMMEN
)
AS SELECT
PDCL.FOUR.CLEFO
APPUYER SUR ENTREE POUR CONTINUER
0: C1 CH: -GN350

```

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL   QSBLOC EXEMPLE SQL/DS

SEGMENT   : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE      : V
          MAJ CATALOGUE O/N :

PDCL.FOUR.FOURNI
PDCL.FOUR.MATE
PDCL.FOUR.RELEA
PDCL.FOUR.LANGU
PDCL.FOUR.QTMAS
PDCL.FOUR.QTMAM
PDCL.FOUR.LIBFO
PDCL.FOUR.DATE
PDCL.FOUR.HEURE
PDCL.FOUR.FILLER
PDCL.DODZ05.COCARA
FROM PDCL.FOUR
     PDCL.DODZ05
;

*** FIN ***
O: C1 CH:
-----

```

INTEREL RDBC

EXEMPLE INTEREL RDBC

Ce sous-chapitre présente les écrans spécifiques d'un Bloc Base de Données de type Interel RDBC (code 'QTBLOC').

Vous trouverez, dans l'ordre :

- la description du bloc,
- le résultat de la génération d'une Table, à partir de la ligne 100 de description du bloc,
- le résultat de la génération d'un Index, à partir de la ligne 210 de description du bloc,
- le résultat de la génération d'une Vue, à partir de la ligne 350 de description du bloc.

Les demandes de génération et la description des Segments à partir desquels s'est effectuée la génération sont présentées dans le sous-chapitre 'Ecrans Communs' de ce chapitre.


```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QTBL0C EXEMPLE INTEREL RDBC

SEGMENT   : DZ05   NOM EXTERNE   : DODZ05
TYPE      : T
          MAJ CATALOGUE O/N :

CREATE TABLE DODZ05
/* NUCOD  USAGE ERRONE                                     */
(COCARA      CHAR      (00001)   NOT NULL,
 FOURNI      CHAR      (00003),
 NUCLIE      CHAR      (00008),
 VILLE      VARCHAR   (00020),
/* REMISE USAGE ERRONE                                     */
 CORESP      VARCHAR   (00256),
 MATERIEL    VARCHAR   (00008),
 DATED      CHAR      (00010),
 HEURE      CHAR      (00008),
 PRECIS     CHAR      (00026)
;

*** FIN ***
O: C1 CH: -GN100
-----

```

```

                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QTBLOC EXEMPLE INTEREL RDBC

SEGMENT   : DZ10   NOM EXTERNE   : INDZ10
TYPE      : I
          MAJ CATALOGUE O/N :

CREATE INDEX INDZ10
ON DODZ10
(FOURNP
  NUCOM          ASC ,
  LIVRABLE       )
;

*** FIN ***
0: C1 CH: -GN210

```

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL   QTBL0C EXEMPLE INTEREL RDBC

SEGMENT   : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE      : V
          MAJ CATALOGUE O/N :

CREATE VIEW VUDZ09DBE
(CLEFO
 FOURNI
 MATE
 RELEA
 /* QTMAS  USAGE ERRONE
 /* QTMAM  USAGE ERRONE
 LANGU
 LIBFO
 DATE
 HEURE
 FILLER
 COMMEN
 )
AS
SELECT
APPUYER SUR ENTREE POUR CONTINUER
O: C1 CH: -GN350
-----

```

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QTBLOC EXEMPLE INTEREL RDBC
SEGMENT   : DZ09   NOM EXTERNE   : VUDZ09DBE

TYPE      : V
           MAJ CATALOGUE O/N :

FOUR.CLEFO           ,
FOUR.FOURNI          ,
FOUR.MATE            ,
FOUR.RELEA          ,
FOUR.LANGU           ,
FOUR.LIBFO           ,
FOUR.DATE            ,
FOUR.HEURE           ,
FOUR.FILLER          ,
DODZ05.COCARA        ,
FROM FOUR            ,
                   DODZ05
;

*** FIN ***
O: C1 CH:
-----

```

INTEREL RFM

EXEMPLE INTEREL RFM

Ce sous-chapitre présente les écrans spécifiques d'un Bloc Base de Données de type Interel RFM (code 'QUBLOC').

Vous trouverez, dans l'ordre :

- la description du bloc,
- le résultat de la génération d'une Table, à partir de la ligne 100 de description du bloc,
- le résultat de la génération d'un Index, à partir de la ligne 210 de description du bloc,
- le résultat de la génération d'une Vue, à partir de la ligne 350 de description du bloc.

Les demandes de génération et la description des Segments à partir desquels s'est effectuée la génération sont présentées dans le sous-chapitre 'Ecrans Communs' de ce chapitre.

```

-----
                VARIANTE CICS/VSAM                                *PDMB.D801.BMS.259
DESCRIPT. BLOC RELATIONNEL  QUBLOC EXEMPLE INTEREL RFM

A NLG : T NOM EXTERNE                CODE TABLE      TY CODE          BIBLI
      :                               VUE             CLE GEN
100 : T DODZ05                        DZ05                C                0067
200 : T DODZ11                        DZ11                C                0058
210 : I INDZ10                        DZ10                C                0067
300 : T                                F011                C                0067
350 : V VUDZ09DBE                     DZ09                C                0067
360 : V VUDZ08S4                       DZ08                4 C              0048
800 : T EXTERNAL-PI00                 PI00                C                0219
      :
      :
      :
      :
      :
      :
      :
      :
      :
*** FIN ***
O: C1 CH: BQUBLOC DR
-----

```

```

-----
                                VARIANTE CICS/VSAM                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QUBLOC EXEMPLE INTEREL RFM

SEGMENT   : DZ05   NOM EXTERNE   : DODZ05
TYPE      : T
          MAJ CATALOGUE O/N :

CREATE TABLE DODZ05
-- NUCOD  USAGE ERRONE
(COCARA      CHAR      (00001)    NOT NULL,
 FOURNI      CHAR      (00003),
 NUCLIE      CHAR      (00008),
 VILLE       CHAR      (00020),
-- REMISE USAGE ERRONE
CORESP      CHAR      (00256),
MATERIEL    CHAR      (00008),
DATED       CHAR      (00010),
HEURE       CHAR      (00008),
PRECIS      CHAR      (00026))
;

*** FIN ***
O: C1 CH: -GN100
-----

```

```
-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QUBLOC EXEMPLE INTEREL RFM

SEGMENT   : DZ10  NOM EXTERNE   : INDZ10
TYPE      : I
          MAJ CATALOGUE O/N :

CREATE INDEX INDZ10
ON DODZ10
(FOURNP
NUCOM          ASC ,
LIVRABLE      )
;

*** FIN ***
O: C1 CH: -GN210
-----
```

```

-----
                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QUBLOC EXEMPLE INTEREL RFM

SEGMENT   : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE      : V
          MAJ CATALOGUE O/N :

CREATE VIEW VUDZ09DBE
(CLEFO
 FOURNI
 MATE
 RELEA
-- QTMAS  USAGE ERRONE
-- QTMAM  USAGE ERRONE
 LANGU
 LIBFO
 DATE
 HEURE
 FILLER
 COMMEN
)
AS
SELECT
APPUYER SUR ENTREE POUR CONTINUER
O: C1 CH: -GN350
-----

```

```

-----
                                VARIANTE CICS/VSAM                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QUBLOC EXEMPLE INTEREL RFM

SEGMENT   : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE      : V
          MAJ CATALOGUE O/N :

FOUR.CLEFO                                ,
FOUR.FOURNI                                ,
FOUR.MATE                                  ,
FOUR.RELEA                                ,
FOUR.LANGU                                ,
FOUR.LIBFO                                ,
FOUR.DATE                                  ,
FOUR.HEURE                                 ,
FOUR.FILLER                               ,
DODZ05.COCARA
FROM FOUR                                  ,
      DODZ05
;

*** FIN ***
O: C1 CH:
-----

```

SYBASE

EXEMPLE SYBASE

Ce sous-chapitre présente les écrans spécifiques d'un Bloc Base de Données de type Sybase (code 'QYBLOC').

Vous trouverez, dans l'ordre :

- la description du bloc,
- le résultat de la génération d'une Table, à partir de la ligne 100 de description du bloc,
- le résultat de la génération d'un Index, à partir de la ligne 210 de description du bloc,
- le résultat de la génération d'une Vue, à partir de la ligne 350 de description du bloc.

Les demandes de génération et la description des Segments à partir desquels s'est effectuée la génération sont présentées dans le sous-chapitre 'Ecrans Communs' de ce chapitre.


```

                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QYBLOC EXEMPLE SYBASE

SEGMENT   : DZ05  NOM EXTERNE   : DODZ05
TYPE      : T
          MAJ CATALOGUE O/N :

CREATE TABLE DODZ05
(COCARA    CHAR          (00001)    NOT NULL,
 NUCOD     SMALLINT      NULL,
 FOURNI    CHAR          (00003)    NULL,
 NUCLIE    CHAR          (00008)    NULL,
 /* CORESP RUBRIQUE DE LONGUEUR > 00255 NON PRISE EN COMPTE */
 VILLE     VARCHAR       (00020)    NULL,
 REMISE    NUMERIC       (00006,02) NULL,
 MATERIEL  VARCHAR       (00008)    NULL,
 DATED     DATETIME      NULL,
 HEURE     DATETIME      NULL,
 PRECIS    CHAR          (00026)    NULL)
;

*** FIN ***
O: C1 CH: -GN100

```

```
-----
                                VARIANTE CICS/VSAM                *PDMB.DDDD.BMS.198|
GENERATION DES ORDRES SQL  QYBLOC EXEMPLE SYBASE

SEGMENT   : DZ10   NOM EXTERNE   : INDZ10
TYPE      : I
          MAJ CATALOGUE O/N :

CREATE INDEX INDZ10
ON DODZ10
(FOURNP
  NUCOM          ASC ,
  LIVRABLE      )
;

*** FIN ***
0: C1 CH: -GN210
-----
```

```

                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QYBLOC EXEMPLE SYBASE

SEGMENT   : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE      : V
          MAJ CATALOGUE O/N :

CREATE VIEW VUDZ09DBE
(CLEFO
FOURNI
MATE
RELEA
LANGU
QTMAS
QTMAM
LIBFO
DATE
HEURE
FILLER
COMMEN
)
AS SELECT
FOUR.CLEFO
APPUYER SUR ENTREE POUR CONTINUER
O: C1 CH: -GN350

```

```

                                VARIANTE CICS/VSAM                                *PDMB.DDDD.BMS.198
GENERATION DES ORDRES SQL  QYBLOC EXEMPLE SYBASE

SEGMENT   : DZ09   NOM EXTERNE   : VUDZ09DBE
TYPE      : V
          MAJ CATALOGUE O/N :

FOUR.FOURNI
FOUR.MATE
FOUR.RELEA
FOUR.LANGU
FOUR.QTMAS
FOUR.QTMAM
FOUR.LIBFO
FOUR.DATE
FOUR.HEURE
FOUR.FILLER
DODZ05.COCARA
FROM FOUR
      DODZ05
;

*** FIN ***
O: C1 CH:

```




Référence : DDDSQ000351F - 7420

Imprimé en France